# A Survey On Off-Line Cursive Word Recognition

Alessandro Vinciarelli [a]

IDIAP–RR 00-43

[a]  IDIAP

# A Survey On Off-Line Cursive Word Recognition

Alessandro Vinciarelli

**Abstract.** This paper presents a survey on Off-Line Cursive Script Recognition. The approaches to the problem are described into detail. Each step of the process leading from raw data to the final result is analyzed.

This survey is divided into two parts, the first one dealing with the general aspects of Cursive Script Recognition, the second one focusing on the applications presented in the literature.

# 1   Introduction

Off-Line Cursive Script Recognition (CSR) is the transcription into an electronic format of cursive handwritten informations. The main development of the field took place in the last decade [1][2] and some commercial products, based on CSR, are yet running in real world applications [3][4].

This does not means that the problem is solved. The attention was mostly focused on the recognition of single words, while many informations of potential interest belong to continuous texts. Moreover, the recognition of cursive words is often supported by other information coming from the application environment. If enough information is available, the accuracy achieved is satisfying, otherwise no.

This made CSR technology effective only in few application domains, indeed postal address reading (where the recognition of the zip code plays an important role) and bank check legal amount recognition (where the courtesy amount, i.e. the amount written in digits, helps the recognition of the legal amount, i.e. the amount written in letters). Many issues are then still open and the problem of the general CSR is still far from being solved.

Several aspects of the recognition process are however independent of the application domain and can be considered in a general framework. For this reason, this survey is divided into two parts. The first one concerns the problems a CSR system must deal with. Each step of the processing is described into detail and the main techniques developed to perform it are shown. The second one focuses on applications presented in the literature and their performances.

The first part is composed of section 2, where the structure of a CSR system is outlined and the single processing step are described into detail, the second one of section 3, where the main application domains of CSR are illustrated. In the final section 4, some conclusions are drawn.

# 2   Structure of a CSR System

The basic structure of a CSR system is shown in figure 1, the only exception to such architecture is given by the human reading inspired systems (see section 2.7). Some of the tasks performed in the recognition process are independent of the approach (e.g the preprocessing), others are related to it and can be used to discriminate among different systems (e.g. the segmentation).

Usually, the raw data cannot be processed directly and the word images must be preprocessed in order to achieve a form suitable for the recognition, this is the aim of the *preprocessing*. The operations performed at its level depend on the data. The removal of background textures, rulers and similar elements is often needed when the word is extracted from forms or checks; a binarization is useful when the words are stored in gray level images. In general terms, the result of the preprocessing must be an image containing the word to be recognized without any other disturbing element.

The next step is the *normalization*. Slant and slope different than 0 (see figure 2) can be caused by acquisition and handwriting style, and their removal results in a word image invariant with respect to such factors. For this reason the step is called normalization.

The first two steps are independent of the recognition approach of the system. The same preprocessing and normalization algorithms can be shared by systems using different recognition approaches. This is no longer true for the *segmentation* that depends on whether the system uses Dynamic Programming or HMMs to perform the recognition. In the first case, the segmentation is said *explicit*, i.e. an attempt is made to isolate the single letters that are then separately recognized. In the second case, the segmentation is *implicit*, i.e. the word is fragmented into subletters and the only constraint to be respected is the oversegmentation: the word must be split at least in correspondence of the actual ligatures between characters. In other terms, each subunit of the word, must belong to only one character. In correspondence of the two alternatives, figure 1 shows two paths.

The *recognition* step uses the word fragments isolated by the segmentation to calculate, for each element of the lexicon, a score. The best scoring lexicon word is assumed as interpretation of the handwritten data. Before the recognition, the fragments are converted into vectors through a *feature extraction process*.
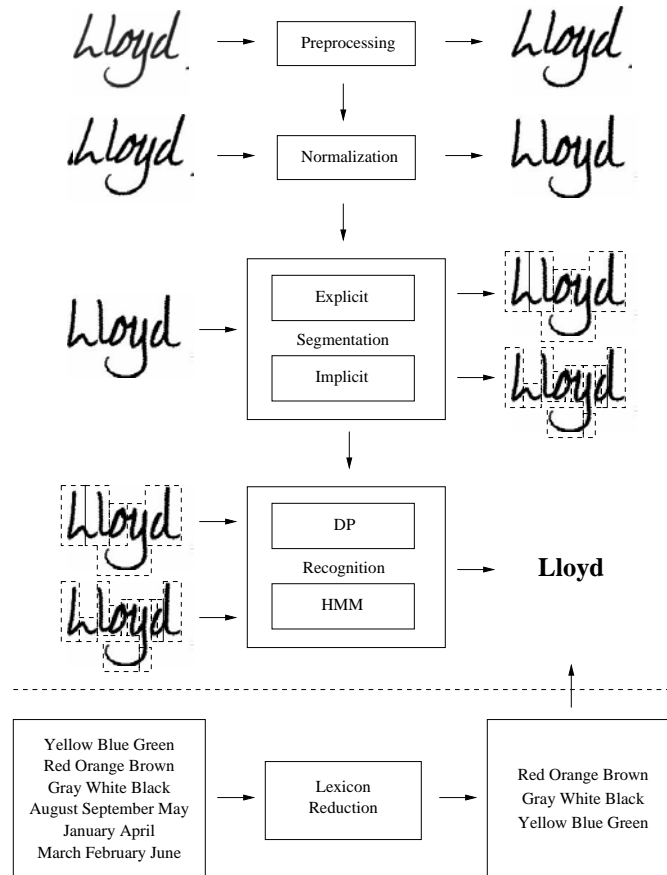
Figure 1: General Model of a CSR system.

A fundamental element in a CSR system is the *lexicon*, a list of the allowed interpretations of the handwritten data. Intuitively, by reducing the size of the lexicon, the accuracy of a CSR system can be improved since the probability of misclassification is reduced. A first, most important, limit to the lexicon size is given by the application environment (e.g., when recognizing legal amounts on bank checks, the only allowed transcriptions are numbers written in letters). A further lexicon reduction can then be achieved by analyzing the handwritten data itself and by extracting from the lexicon all the incompatible interpretations (if a handwritten word does not present ascenders or descenders, only transcriptions composed of letters without ascenders or descenders can be accepted).

In the following sections, each step of the processing will be described in detail.

## 2.1  Normalization

In an ideal model of handwriting, a word is supposed to be written horizontally and with ascenders and descenders aligned along the vertical direction. In real data, such conditions are rarely respected. *Slope* (the angle between the horizontal direction and the direction of the implicit line on which the word is aligned) and *slant* (the angle between the vertical direction and the direction of the strokes supposed to be vertical) are often different than 0 and must then be eliminated (see figure 2).

The normalized images are invariant with respect to the sources of slant and slope (acquisition and handwriting style) and this is helpful to the recognition process. In Dynamic Programming based systems, the removal of slant and slope makes the characters less variable in shape, then easier to
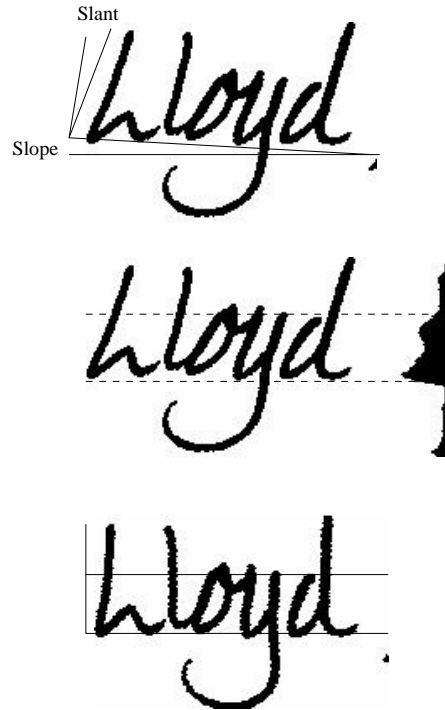
Figure 2: Preprocessing. The word, before the preprocessing, shows slope and slant different than 0. In the horizontal density histogram, the lines corresponding to the core region are evident. The long horizontal stroke in the lower part of the $y$ creates a high density area that can be erroneously assumed as core region. After the preprocessing, the word appears horizontal with ascenders and descenders aligned along the vertical axis.

be classified with pattern recognition techniques. Besides, the normalization creates segments where the handwritten data is piece-wise stationary, whose presence is a necessary assumption to the use of HMMs.
In the following subsections 2.1.1 and 2.1.2, methods for removing respectively slope and slant are described. Some attempt to use in the recognition the information lost after the normalization is described in section 2.1.3.

### 2.1.1  Slope correction and reference line finding

Most of the desloping techniques presented in the literature are inspired by the method proposed in [5]. This consists in giving a first, rough estimate of the *core region* (the region enclosing the character bodies), then in using the stroke minima closest to its lower limit to fit the ideal line on which the word is aligned. The image is rotated until such line is horizontal and the image is finally desloped (see figure 2).
The estimation of the core region, the fundamental step, is made by finding the lines with the highest horizontal density (number of foreground pixels per line). The core region lines are in fact expected to be more denses than the others. The horizontal density histogram is analyzed looking for features such as maxima and first derivative peaks, but these features are very sensitive to local characteristics and many heuristic rules are needed to find the actual core region lines [6].
Some alternative techniques were proposed in [7] and [8]. In such works, the density distribution is analyzed rather than the density histogram itself in order to make statistically negligible the influence of local strokes. The method presented in [7] is based on the entropy of the distribution (supposed to

be lower when the word is desloped), while the technique in [8] applies the Otsu Method [9] in order to find a threshold distinguishing between core region lines (above the threshold) and other lines. In [10], the image is rotated for each angle in a interval and the rotated image giving the highest peak of the first derivative of the horizontal density histogram is assumed as desloped. Long disturbing horizontal strokes are eliminated with Gabor filters [11].

Another important result of the desloping is the detection of the limits of the core region, called *upper* and *lower baseline*, that play an important role as reference lines.

### 2.1.2   Slant correction

Most of the methods for slant correction are also based on the technique proposed in [5]. This relies on the selection of near vertical strokes the slope of which is assumed as local slant estimate. The global slant value is obtained by averaging over all the local estimates. Techniques based on such idea can be found in [10][12][13][14], each work using a different method to select the strokes involved in the global slant estimation.

A method avoiding the selection of specific strokes can be found in [15], where all the points on the border are used to calculate the most represented directions.

A different approach was proposed in [8][16]: a measure of the "deslantedness" is performed over all the shear transforms of the word image corresponding to the angles in a reasonable interval. The transformed image giving the highest "deslantedness" value is the deslanted one. In [16], the measure is given by the Wigner-Ville Distribution [11] of the vertical density histogram. In [8], the measure is the sum of the square values of the vertical densities of the columns containing a continuous stroke.

### 2.1.3   Use of the writing style as a source of information

The normalization step, by eliminating characteristics introduced by the writing style, gives the handwritten words a standard form, but, in the meantime, destroys some information.

In some work, the possibility of using the writing style as a source of information helpful to the recognition process has been proposed. Several approaches were experimented to group into families the handwriting styles.

In [17], stroke width, number of strokes per unit length in the core region, core region position and the histogram of the quantized directions of generic strokes are used as features to characterize the writing style. Such features are selected because they are not related to any character in particular, so they do not depend on the word they are extracted from.

Fractal dimension [18] related measures have been proposed for the same purpose in [19][20]. The fractal dimension is shown to be a very stable parameter for a writer even in samples produced in different years.

The features proposed are efficient in grouping the styles in well defined families, but no results were presented in terms of recognition rate improvement.

## 2.2   The segmentation

The segmentation of an image is performed by connecting, or identifying maximal connected sets of pixels participating in the same spatial event [9]. In CSR terms, this means to isolate fragments in the handwritten word supposed to be the basic information units for the recognition process. As pointed out in section 2, the segmentation can be explicit or implicit depending on whether the isolated primitives are expected to be characters or not.

The explicit segmentation is a difficult and error prone process because of the Sayre's Paradox [21]: a letter cannot be segmented before having been recognized and cannot be recognized before having been segmented. Until now, no methods were developed able to segment handwritten words exactly into letters [22][23].

On the contrary, implicit segmentation is easy to achieve because the only constraint to be respected

is the oversegmentation (see section 2). The number of spurious cuts [1] needs not to be limited.

In principle, the segmentation is independent of the recognition technique, however the explicit segmentation is mostly performed in Dynamic Programming based systems, while implicit segmentation is used in architectures involving Hidden Markov Models. For this reason the segmentation was used elsewhere as a key component in distinguishing among different approaches [2]. This is, in our opinion, not completely correct because, if the choice of the segmentation was free, the implicit segmentation, easier to be performed, would be always preferred. The real problem is that, when applying the Dynamic Programming, the fragments extracted from the word are supposed to be characters and only small variations with respect to this condition can be tolerated. The segmentation must then be as explicit as possible and spurious ligatures must be minimized.

HMMs are not only able to work on a sequence of fragments not necessarily corresponding to letters, but can also face with variations and noise that can occur in the sequence itself. This allows the use of an implicit segmentation.

The actual key point seems then not to be in the segmentation, but rather in the recognition technique adopted.

## 2.3 Feature extraction

The features can be grouped into three classes depending on whether they are extracted from the whole word (high level features), the letters (medium level features) or subletters (low level features). In the next three subsections, each feature class is described in more detail.

### 2.3.1 Low level features

Low level features are extracted from letter fragments that have elementary shapes such as small lines, curved strokes, bars and similars. The features account, in general, for their position and simple geometric characteristics.

It is frequent to use features that describe the distribution of pixels with respect to reference lines: the percentages of the stroke edges in core, ascenders and descenders regions are used in [24][25], the distances of the foreground-background transitions from the median line of the core region are proposed in [26][27], the percentages of foreground pixels in core, ascenders and descenders region are applied in [28][29].

To have an overall description of the shape, features like curvature [24][25], center of mass [29], histogram of the strokes directions [14][29][30] are used.

In several works, the small strokes are considered deformations of the elements of a basic set of strokes and the deformation itself is used as feature. In [5][31][32], the set of basic elements is composed by different curve or linear strokes (e.g. curves up, down, left or right directed).

### 2.3.2 Medium level features

Systems based on explicit segmentation must face with the recogniton of cursive characters. The biggest problem is the variability of their shapes [33], then features performing an averaging over local regions are preferred. A normalized image of the character is used in [34]. Background-foreground transition distribution is used in [26]. In [35], bar features are proposed. In [36], the feature extraction is performed over trigrams and consists in a vectorization of the contour. The systems involving the character recognition must also cope with primitive aggregates that are not characters. To distinguish between actual letters and non-letters, a method is proposed in [26]: the presence of too many ascenders or descenders in a letter candidate is used as a rejection criterion.

---

[1]points where the word is split even if does not correspond to actual ligatures between characters.

### 2.3.3   High level features

Features such as loops, ascenders and descenders are often referred as high level features. Since they consist of the detection of structural elements, they do not depend on the writing style and are then stable with respect to cursive variability.
Together with loops, ascenders and descenders (the most used since they are easily detected), we also find junctions, stroke endpoints, t-bars and dots in the literature.
In some works [14][25][30], high level features are extracted from the word skeleton (a representation of the word that allows an easy detection and ordering of structure elements).
In some cases [7][12], mainly in applications involving small lexica (such as bank check reading), the high level features are used to give a rough representation of the word. This allows to discard part of the lexicon or to reject a result of the recognition process whose representation is not compatible with the detected one (see figure 6).

## 2.4   Lexicon reduction

The size of the lexicon is one of the most important parameters in a CSR system. As the number of allowed intepretations increases, the probability of misclassification becomes higher. For this reason several Lexicon Reduction Systems (LRS) were developed. In some cases they are based on other informations than the word to be recognized (e.g. the zip code, in postal applications, limits the number of allowed transcriptions of a handwritten town name). In other cases they use the handwritten word itself to discard some interpretations from the lexicon. In this section, the attention will be focused on this latter category of LRS.
In general terms, an LRS takes as input a lexicon and a handwritten word and gives as output a subset of the entries of the lexicon. An optimal trade-off must be found between the compression rate of the lexicon and the number of times that the correct transcription is not discarded.
The LRS are always based on a rough representation of the handwritten data that allows to rank (depending on a compatibility score) the lexicon entries or discard those which are incompatible with the data. In [36], a system based on the detection of trigrams is shown to put the correct intepretation in the top 200 positions of a rank generated using a 16200 word lexicon with an accuracy of 96%. This corresponds to a compression rate of 1.2%.
In [37], the handwritten word is converted into a string of characters representing structural elements (e.g. stroke extrema). For each entry of the lexicon, an ideal model (represented by a string) is given and the compatibility between lexicon words and handwritten data is measured with edit distance. Starting from a lexicon of around 21000 words, an average compression rate of ∼33% is achieved with an accuracy higher than 99%.
The LRS described in [38] relies on the concept of key characters. These are the letters that are most easily recognized and are used to find the most compatible entries of the lexicon. The average reduction rate achieved is 72.9% with an accuracy of 98.6%.

## 2.5   The data

The data can be considered not only as an input to a CSR system, but as an actual part of it. The data to be recognized is not a simple collection of words without any relationship between them, but a sample of the data produced in some human activity. The nature of such activity creates conditions that influence the solution of the recognition problem. Changing the data means changing the problem: depending on the data used, a lexicon will be present or not and, when present, its size will change. Depending on the number of writers, the style of each one of them will affect with different importance the algorithms and their results.
Several databases [14][39][40][41] are available and can be shared by different groups. Each one is related to some application, e.g the CEDAR database [39] is composed of postal material and allows the simulation of a postal plant. The use of the same data by many researchers allows an actual comparison of the results achieved. However the literature presents most often works showing results
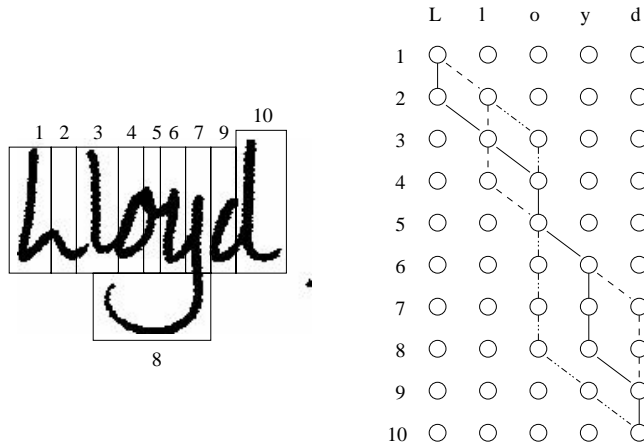
Figure 3: Recognition in systems based on segmentation and dynamic programming. The matching of the word with the sequence of primitives is found as the optimal path through the trellis.

obtained over data used only by their authors.

A solution to this problem has been proposed in [42], where the human performance is indicated as an absolute term of comparison. The same data used to test a system (or a representative subset of it) should be transcribed by a human reader. The performance of the human should be considered as the best result achievable over the data.

## 2.6   The recognition

The recognition consists in finding, among the words of the lexicon, the most compatible one with the handwritten data. The two main techniques for this task are Dynamic Programming and Hidden Markov Modeling that will be described in subsection 2.6.1 and 2.6.2 respectively.

These methods use the feature vectors extracted from the word fragments to calculate a score giving a measure of the matching of the handwritten data with a possible interpretation. The score can be a probability, a distance or a cost depending on the specific case. The best scoring word of the lexicon is assumed to be the correct interpretation of the data.

### 2.6.1   Dynamic programming

Dynamic programming [43][44] is often used after explicit segmentation to match a sequence of $M$ letters with a sequence of $N$ word fragments. This is defined in [45] as a synchronous sequential decison problem and its solution corresponds to finding the optimal cost path in the trellis of figure 3. After having matched the first $m < M$ letters with the first $i < N$ word fragments, the following step can reach any further primitive $n$ (where $n > i$ since the words are written from left to right). The optimal step, associating the first $m + 1$ letters with the first $n$ primitives, will be found as follows:

$$\phi_{m+1}(i,n) = min_l[\phi_m(i,l) + \zeta(l,n)] \tag{1}$$

where $\phi()$ represents the optimal cost and $\zeta()$ the cost to associate the $m+1^{th}$ letter to the aggregation composed by primitives $i+1, i+2, \ldots, n$. The equation 1 implies that any partial, consecutive sequence of moves of the optimal sequence from $i$ to $n$ must also be optimal, and that any intermediate point must be the optimal point linking the optimal partial sequences before and after that point. This is the way the Bellmann's optimality principle [2] is put into a functional equation.

---

[2] An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the final decision.
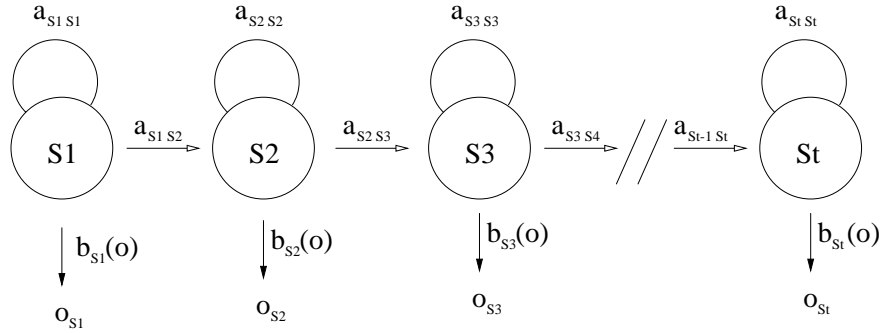
Figure 4: Hidden Markov Model. The sequence of the states is produced by the transition probabilities $a_{ij}$. At each state is associated an emission probability $b_j(\mathbf{o})$.

Usually a maximum number $k$ of primitives allowed to form a single character is set, then $n \in [i+1, i+k]$.

The cost is calculated in several ways: sometimes as a distance from letter prototypes (the optimal path will then be a minimum cost path), sometimes as a probability for the aggregations of being a certain letter (the optimal path will then be the maximum cost path and, in equation 1, $min$ will be replaced by $max$).

A problem in using DP was pointed out in [46][47]. The cost of the path is calculated by averaging over the costs of the single steps. The cost of a single step is in general related to the output of a statistical or neural recognizer trained at the character level. The performance of such recognizers is not tied to the word recognition performance. In other words, it can happen that even if the character recognizer is improved, the overall word recognizer performance is decreased. This is undesirable because makes very difficult to control the system performance. The proposed solution consists in using weights trained at the word level when calculating the cost of the path.

### 2.6.2 Hidden Markov Models

For a good introduction to Hidden Markov Models, see [45][48][49] and for their specific use in handwriting recognition see [50].

An HMM is a statistical model of observation sequences. These are supposed to be produced by a system that changes state at regular steps. The set of possible states has finite size $N$ and the transition from any state $i$ to any other state $j$ (eventually $i = j$) is governed by a stochastic process. The probability of being in state $s_t$ at step $t$ can be expressed as:

$$p(s_t) = p(s_t | s_{t-1}, s_{t-2}, \ldots, s_{t-d}) \tag{2}$$

but, in general, the assumption $d = 1$ is made so that the state at step $t$ depends only on the state at step $t - 1$. The model is then called a first order HMM. A second important assumption is that the transition probability does not depend on $t$ and is then stationary. The evolution dynamic of the system is entirely represented by the matrix $A = \{a_{ij}\} = \{p(s_t = j | s_{t-1} = i)\}$ (see figure 4).

To start the process, an initial state probability $\pi_i$ (probability of being in state $i$ at first step) must also be defined. The vector $\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$ represents the initial state probability distribution. The sequence of the states is assumed to be not observable (thus the name hidden), but gives rise to a second stochastic process characterized by the probabilities $b_j(\mathbf{o})$ to emit observation $\mathbf{o}$ while being in state $j$. The $b_j$ are probability density functions in the space of the observations and are continuous or discrete depending on the nature of the observations themselves. The set $B = \{b_1(\mathbf{o}), b_2(\mathbf{o}), \ldots, b_N(\mathbf{o})\}$ is the bridge between the hidden states and the observations.

A Hidden Markov Model $\lambda$ is represented by the set: $\{A, B, \pi\}$. The calculation of the parameters in
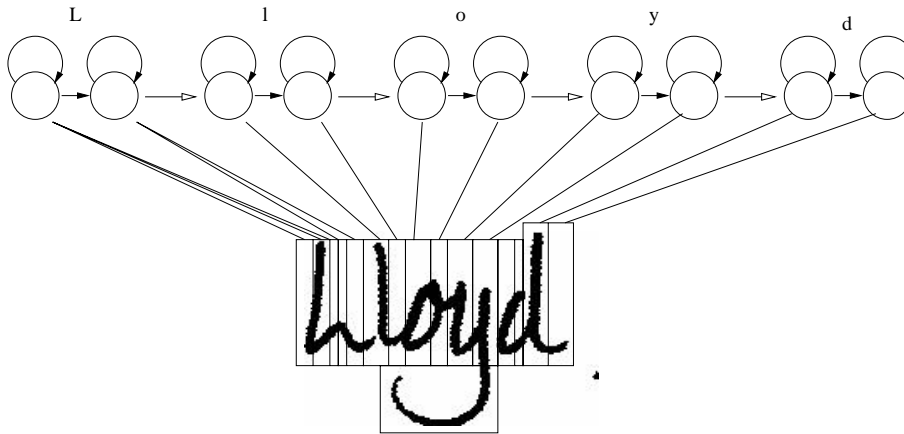
Figure 5: Recognition in systems based on HMMs. The statistical nature of HMMs allows them to cope with the cursive variability.

$A$,$B$ and $\pi$ (the training), is performed with the Baum-Welch Algorithm, a particular form of the EM algorithm [51][52][53][54][55].

Once a model $\lambda$ is trained, the probability $P(\mathbf{O}|\lambda)$ of an observation sequence $\mathbf{O}$ being produced by the model $\lambda$ is calculated usually with the Viterbi algorithm, which gives an acceptable approximation of the probability with a reduced computational effort [56][57].

When applied to the handwriting, the hidden states are the letters of a possible interpretation and the observations are the feature vectors extracted from the primitive segments (see figure 5). An HMM is built for each word[3] in the lexicon and the corresponding likelihood (probability of the data being produced by the model) with the handwritten data is calculated. The most likely interpretation is assumed to be the correct one.

## 2.7   Human Reading Inspired Systems

Several human reading models were elaborated by cognitive scientists[58][59] [60][61]. None of them is able to account for every aspect of the *lexical access*, i.e. the process of giving meaning to a handwritten word, but some of them explain effects that can be useful in developing a CSR system. In particular, the McClelland and Rumelhart model [60] shows how the *Word Superiority Effect* (the fact that a letter is easier to be recognized when embedded in a word than when isolated) emerges.

The Word Superiority Effect can be considered as the influence of the word context in single letter recognition. This has been proposed also in works not inspired by psychological models, but the influence of the context had to be modeled directly while in the McClelland and Rumelhart model it is an effect of the global architecture.

The model is formed by three levels of processing (corresponding to different levels of abstraction) working in parallel: feature level, letter level and word level. The levels interact in that the results obtained at a certain level are used by the process at another level. The interaction is due to inhibitory or excitatory connections between levels. The feature level tends to excite the letters corresponding to the features detected and to inhibe the others (e.g. a loop excitates the presence of the letter $o$, but inhibes the presence of the letter $s$). The letter level excitates the connections with the word level linking words that contain the letter detected. Viceversa the word level excitates the letters that are contained in excitated words, but were not detected at the letter level. The inhibitory connections work in a similar way and tend to decrease the excitation of words or letters that are not compatible

---

[3]The word HMMs are usually constructed by concatenating letter models. This allows to change the lexicon without changing the training set (training samples of the lexicon entries are not necessary) making the system more flexible.
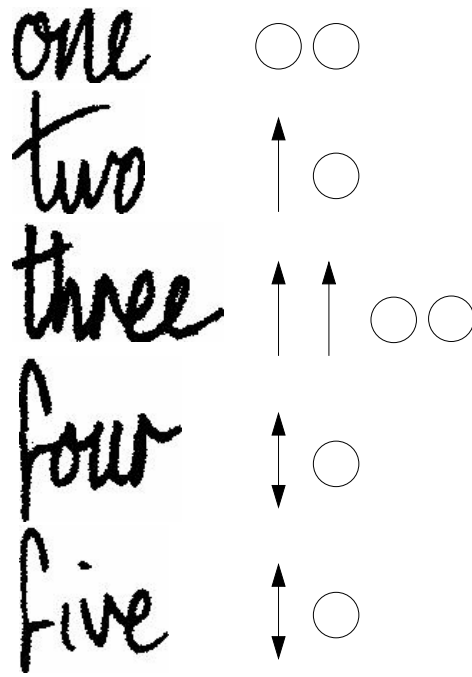
Figure 6: A representation of handwritten numbers between one and five in terms of global features. The representation allows to discriminate one, two and three, but there is an ambiguity between four and five.

with each other.

By using such model in CSR system, the influence of context is not modeled (e.g. as in [26]), but is an effect of the global architecture.

Several works were inspired by this model [7][36][62][63][64][65][66][67]. In general, they present a recognition cycle divided into two parts: a bottom up and a top down process. The bottom up process consists in extracting global features such as ascenders, descenders and loops. This kind of features is not expected to give the identity of a letter, but only to account for the global shape of the word. This representation is used to create a short list of candidates among the lexicon elements by selecting only the words with a compatible shape. (e.g. the presence of an ascender will allow to discard all the words of the lexicon that do not contain letters with ascenders, see figure 6). In some cases, the short list contains only one word and the recognition process is completed, in other cases several words are selected and the top down process is needed. This consists of a verification, i.e. of looking for the letters that compose each candidate of the short list. The global features, previously extracted, are used as anchor points and help in indicating where to look for a certain letter. The presence of a letter is verified by looking for local features referred some time as secondary or conditional features in the sense that they are used only in conjunction with the primary when these are not sufficient.

This approach do not need the recognition of all the letters of the word, in some cases (when the short list contains only one word) it is not necessary to recognize any letter. This technique is based more on the discrimination between the words of the lexicon than on their recognition. Because of this, it is effective only in applications that involve a small ($25 \sim 30$ words) and static lexicon.

| Authors | DB size | performance |
|---|---|---|
| Côté et al. [7] | 3113 | 74.0% |
| Guillevic et al. [66] | 7837 | 71.8% |
| Dodel et al. [65] | 1150 | 70% |
| Olivier et al. [68] | 6000 | 69.6% |
| Knerr et al. [29] | 189659 | 89.2% |
| Saon [69] | 100000 | 82.5% |
| Kaufmann et al. [70] | 13000 | 71.9% |

Table 1: Performances in bank check reading. For each work cited in the first column, the table reports the size of the database used in the experiments and the best performance (in terms of words correctly recognized) obtained. The lexicon size is between 25 and 30 depending on the language of the legal amounts processed. The double horizontal line separates the results obtained with human reading model inspired systems (upper part) from that obtained with HMM based systems (lower part).

# 3   Applications

In the next subsections, the main applications of CSR will be presented. Bank check reading is described in subsection 3.1, postal applications are illustrated in subsection 3.2 and generic content document recognition is presented in subsection 3.3.

## 3.1   Bank check reading

Many works about CSR are dedicated to the bank check legal amount recognition. The developed systems are good enough to be used in commercial products as described in [4], where a family of systems able to work on french, english and american checks is claimed to have a performance close to a human reader (rejecting 30-40% of the data when their score is too low).

The reading of legal amounts involves small lexicons (between 25 and 30 words) and each word to be recognized is produced by a different writer. An important advantage in bank check reading is the presence of the courtesy amount (the amount written in digits). This can be read reliably, but is not relevant from a legal point of view, so an automatic check processing system must read also the amount written in letters. On the other hand, the redundancy of information when reading both courtesy and legal amount can improve the performance of the system.

In [29][32][68], an implicit segmentation is applied, and the recognition is performed with an Hidden Markov Model.

A segmentation free approach is proposed in [71][72][73], where a sliding window is applied to the amount image and a recurrent neural network is used in conjunction with HMMs. The sliding window is also used in [70], where a correction mechanism activated by a mismatch between courtesy and legal amount is proposed.

In [69], the scaled images of the legal amount are considered as random field realizations and recognized in conjunction by HMM and Markov Random Fields [74].

An hybrid approach, using analytic features to segment the legal amount in sub words and then global features to recognize these latter, is proposed in [75].

Bank check reading, due to the small lexicon, is the only domain where the human reading inspired systems were found to be effective [7][65][66][67].

The results obtained in terms of percentage of words correctly recognized in several works are shown in table 1. The double horizontal line separates the results obtained with methods inspired by psychological models and those obtained with other techniques.

The best performance has been achieved in [29] with a system based on HMMs. The result is achieved

| Authors | DB size | performance |
|---|---|---|
| Chen et al. [46] | 996 | 90.8% |
| Gader et al. [26] | 1000 | 85.8% |
| Kim et al. [13] | 3000 | 88.2% |
| Mohamed et al. [27] | 10567 | 89.3% |
| Kundu et al. [79] | 3000 | 88.3% |
| Chen et al. [28] | 1583 | 72.3% |
| El Yacoubi et al. [12] | 19447 | 96.3% |

Table 2: Performances in postal applications. For each work cited in the first column, the database size and the performance (in terms of percentage of words correctly recognized) for a 100-word lexicon is reported. The double horizontal line separates the results obtained with HMM based systems (lower part) from that obtained with explicit segmentation and dynamic programming based systems (upper part).

by combining the output of several word classifiers, among them, the Markov recognizer alone has a performance of 85.1%.
In [4], the human performance is said to be around 99%. This rate can be achieved by current automatic readers only by discarding the more ambiguous samples.

## 3.2   Postal applications

Most of the works in the literature concern postal applications. The data involved in this domain are completely unconstrained, each word is written by a different writer, the words can be cursive, handprinted or a mix of the two styles. The lexicon depends, in general, on the output of a zip code recognizer. When the zip code is recognized, it is not even necessary to read further informations in the address. When there is unacceptable ambiguity on the last, last two or last three digits of the zip code, then it is necessary to read the town name and the lexicon will contain ten, hundred or towsand words respectively.
Several works are based on segmentation and dynamic programming [27][46][76][77]. In [27], the performance is improved by using, together with the segmentation based system, a segmentation free system based on HMM. The combination of two different approaches (lexicon free and lexicon directed) is also described in [77][78]. Techniques to calculate the score of a lexicon word, given the single character confidences, are proposed in [46] and [76].
A system based on HMM is presented in [28], where a modified Viterbi algorithm is described. In [12], after having performed an explicit segmentation, the system uses an HMM based technique to combine two feature sets: the first oriented to characters, the second to ligatures. The segmentation statistics (the probability of segmenting each letter into $n$ primitives) are taken into account during the recognition process in [13][80][81][79].
A Minimum Edit Distance modified to better represent the errors that can occur in a CSR is described in [82].
In [83][84][85][86], the possibility of reading handwritten lines is investigated to recognize different forms assumed by the same address (e.g. *Park Avenue* or *Park Av.*).
A direct comparison among the systems is impossible because the results are obtained over different databases. However all the databases consist of real world data obtained by digitizing handwritten addresses. The performances presented in the literature are collected in table 2, where the double line separates the segmentation based works (upper part) from the HMM based systems (lower part). The performances are reported for lexicon size 100.

| Authors | DB size | lexicon size | performance |
|---|---|---|---|
| Bozinovic et al. [5] | 130 | 710 | 77% |
| Bunke et al. [25] | 3000 | 150 | 99.3% |
| Senior et al. [14] | 4053 | 1334 | 92.8% |
| Marti et al. [87] | 50754 | 7719 | 60.0% |

Table 3: Performances in generic content document recognition. For each work cited in the first column, the DB size, the lexicon size and the performance (in terms of percentage of correctly recognized words) are collected.

## 3.3    Generic Content Recognition

In the previous sections, systems concerned with specific applications were presented. The application environment involving the system is a source of informations that have a strong influence on the recognition process (the zip code limits the lexicon size in postal applications, the courtesy amount helps in recognizing the legal amount in bank check reading).

In the works presented in this section, the recognition was performed over data that did not allow the use of any other information than the handwritten words themselves. At most, if the words belong to a text, the linguistic knowledge would be introduced.

The data used in the works related to this subfield of CSR is often created ad hoc by asking writers (in some case cooperatives) to produce samples. A possible application of similar works can be the development of personal notes transcription tools. The writer should be asked to respect some constraint in writing (e.g. to leave no spaces between the letters of the same word, to use some standard characters, etc.), but the simplification of the interface between man and machine would still be great. In [5][31] the words produced by few writers are recognized. Both works are based on explicit segmentation and use different level representations of the words that allow making hypotheses about the transcription and looking for its confirmation at the feature level. In [31], the confirmation is obtained as a degree of alignment of letter prototypes with the actual handwritten data. In [5] the confirmation is given by matching the sequence of expected events (e.g. loops, curve strokes of various shape, etc.) with the actual sequence detected in the handwritten word.

In [34][88], a word is segmented explicitly first and then an HMM is used to find the best match between the fragments (the similarity with character prototypes is used to calculate the probability of a fragment beeing a certain letter) and words in the lexicon.

In [24][25], the words written by cooperative writers are represented, after a skeletonization of the word, as a sequence of strokes organized in a graph. Each stroke is represented by a feature vector and their sequence is recognized by an HMM.

The first example of recognition of data extracted from a text (to our knowledge) is presented in [14][30]. The selection of the text is addressed by linguistic criteria, the text is extracted from a corpus supposed to be representative of the current english. This allows the use of linguistic knowledge in recognition.

The data is produced by a single writer, so that an adaptation to his/her writing style can play a role in improving the system performance. In [14][30], the words are skeletonized and then a uniform framing is performed. From each frame a feature vector is extracted and an HMM is used for the recognition. A recurrent neural network is used to calculate the emission probabilities of the HMM. This work contains also a model able to recognize words without lexicon. This is very important in view of the recognition of a generic text, since the extension of the lexicon to the whole dictionary of a language would result in too heavy computational effort.

The use of linguistic knowledge was shown to be effective in recognizing whole sentences rather than single words in [40][89][87][90]. The applied language models are based on statistic distributions of unigrams and bigrams [91]. The use of syntactical constraints (expressed by transition probabilities between different syntactical categories) was experimented in [92][93].

The results achieved in the works presented in this section are obtained over different data sets, not only because they contain different data, but also because they were created with different aims and criteria. While in the last sections 3.1 and 3.2 it was possible to find at least a common application environment, in this section, there are no similarities between different data sets. The performances of some systems dealing with generic texts is reported in table 3.

# 4    Conclusions

This work presented a survey on Off-Line Cursive Script Recognition. A description of the major approaches is given as well as an overview of the applications presented in the literature.
The importance of the information brought by the application environment is highlighted by showing that most of the works on CSR concern two specific tasks: bank check and postal address reading. In both cases, the handwritten data to be recognized is accompanied by other information (courtesy amount and zip code respectively) improving the performance of the system.
In such domains, the accuracy achieved allowed the use of the developed systems in real world applications. There are however many other practical problems involving handwritten informations that can take an advantage from the use of a CSR system. This leads the research efforts toward systems able to recognize data as generic as possible so that an eventual adaptation to a specific task can be quickly performed. The linguistic knowledge is investigated as a solution for the lack of other information than the handwritten data to be recognized.
Independently of specific data or applications, two approaches were shown to be mainly used: the first one is based on the explicit segmentation followed by Dynamic Programming, the second one relies on implicit segmentation and Hidden Markov Models. This latter technique was most frequently used in the last years. The strong theoretic framework provided by HMMs limits the amount of heuristic needed to improve the system performance.
The human reading inspired architectures were also described and their inability to deal with lexica bigger than $25 \sim 30$ words was shown to limit their use to bank check recognition.
A list of available databases was presented. The data is not just an input in a CSR system. The nature of the data has an influence on the recognition process. The use of data written by a single person allows an adaptation to a single handwriting style; a collection of words extracted form a text can be recognized by applying linguistic models; the application environment can limit the lexicon size. Furthermore the availability of databases that can be shared by several research groups helps in making comparisons between different techniques.
The performances of the systems presented in the literature were finally reported showing, in some cases, recognition rates sufficient for real world applications [3][4].
Off-Line CSR is a still evolving field. The important results related to specific application domains cannot be considered conclusive. The open issues to achieve a general Cursive Script Recognition are still many and important.

# References

[1]  R. Plamondon, S. Srihari, On line and off-line handwriting recognition: A comprehensive survey, IEEE Transactions on Pattern Analysis and Machine Intelligence .

[2]  T. Steinherz, E. Rivlin, N. Intrator, Off-line cursive script word recognition - a survey, International Journal of Document Analysis and Recognition 2 (2) (1999) 1–33.

[3] D. D'Amato, E. Kuebert, A. Lawson, Results from a performance evaluation of handwritten address recognition systems for the United States Postal Service, in: Proceedings of $7^{th}$ Int. workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 189–198.

[4] N. Gorski, V. Anisimov, E. Augustin, O. Baret, D. Price, J. Simon, A2iA check reader: A family of bank check recognition systems, in: Proceedings of $5^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Bangalore, 1999, pp. 523–526.

[5] R. M. Bozinovic, S. N. Srihari, Off-line cursive script word recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1) (1989) 69–83.

[6] M. Morita, J. Facon, F. Bortolozzi, S. Garnes, R. Sabourin, Mathematical morphology and weighted least squares to correct handwriting baseline skew, in: Proceedings of $5^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Bangalore, 1999, pp. 430–433.

[7] M. Cote', E. Lecolinet, M. Cheriet, C. Y. Suen, Automatic reading of cursive scripts using a reading model and perceptual concepts - the PERCEPTO system, International Journal of Document Analysis and Recognition 1 (1) (1998) 3–17.

[8] A. Vinciarelli, J. Luettin, Off-line cursive script recognition based on continuous density HMM, in: Proceedings of $7^{th}$ International Workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 493–498.

[9] R. M. Haralick, L. G. Shapiro, Computer and Robot Vision, Addison Wesley, USA, 1992.

[10] J. Cai, Z.-Q. Liu, Off-line unconstrained handwritten word recognition, International Journal of Pattern Recognition and Artificial Intellgence 14 (3) (2000) 259–280.

[11] S. Qian, D. Chen, Joint time-frequency analysis: method and application, Prentice Hall, 1996.

[12] A. El-Yacoubi, M. Gilloux, R. Sabourin, C. Suen, An HMM-based approach for off-line unconstrained handwritten word modeling and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (8) (1999) 752–760.

[13] G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real time application, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (4) (1997) 366–379.

[14] A. W. Senior, A. J. Robinson, An off-line cursive handwriting recognition system, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 309–321.

[15] Y. Ding, F. Kimura, Y. Miyake, M. Shridar, Slant estimation for handwritten words by directionally refined chain code, in: Proceedings of $7^{th}$ Int. workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 53–62.

[16] E. Kavallieratou, N. Fakotakis, G. Kokkinakis, A slant removal algorithm, Pattern Recognition 33 (7) (2000) 1261–1262.

[17] J. P. Crettez, A set of handwriting families: Style recognition, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 489–494.

[18] B. Mandelbrot, Fractals: chance, form and dimension, Freeman, San Francisco, 1977.

[19] V. Bouletreau, N. Vincent, N. Sabourin, H. Emptoz, Synthetic parameters for handwriting classification, in: Proceedings of $4^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Ulm, 1997, pp. 102–106.

[20] T. Freche, N. Vincent, Local and global approaches to achieve quantitative measurement of hand-writings, in: Proceedings of $5^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Bangalore, 1999, pp. 593–596.

[21] K. Sayre, Machine recognition of handwritten words: a project report, Pattern Recognition 5 (3) (1973) 213–228.

[22] G. Casey, E. Lecolinet, Strategies in character segmentation: A survey, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 1028–1033.

[23] Y. Lu, M. Shridar, Character segmentation in handwritten words - an overview, Pattern Recognition .

[24] H. Bunke, M. Roth, E. Schukat-Talamazzini, Off-line recognition of cursive script produced by a cooperative writer, in: Proceedings of International Conference on Pattern Recognition, 1994, pp. 383–386.

[25] H. Bunke, M. Roth, E. Schukat-Talamazzini, Off-line cursive handwriting recognition using Hidden Markov Models, Pattern Recognition 28 (9) (1995) 1399–1413.

[26] P. D. Gader, M. Mohamed, J. H. Chiang, Handwritten word recognition with character and inter-character neural networks, IEEE Transactions on Systems, Man and Cybernetics B 27 (1) (1997) 158–164.

[27] M. Mohamed, P. Gader, Handwritten word recognition using segmentation-free Hidden Markov Modeling and segmentation-based Dynamic Programming techniques, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (5) (1996) 548–554.

[28] 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 16 (5) (1994) 481–496.

[29] S. Knerr, E. Augustin, O. Baret, D. Price, Hidden Markov Model based word recognition and its application to legal amount reading on french checks, Computer Vision and Image Understanding 70 (3) (1998) 404–419.

[30] A. W. Senior, Off-line cursive handwriting recognition using recurrent neural network, Ph.D. thesis, University of Cambridge, Cambridge, UK (1994).

[31] S. Edelman, T. Flash, S. Ullman, Reading cursive handwriting by alignment of letter prototypes, International Journal of Computer Vision 5 (3) (1990) 303–331.

[32] T. Paquet, Y. Lecourtier, Recognition of handwritten sentences using a restricted lexicon, Pattern Recognition 26 (3) (1993) 391–407.

[33] F. Camastra, A. Vinciarelli, Isolated cursive character recognition based on neural nets, Kuenstliche Intelligenz 13 (2) (1999) 17–19.

[34] B. A. Yanikoglu, P. A. Sandon, Off-line cursive handwriting recognition using style parameters, Technical Report PCS-TR93-192 Dartmouth College .

[35] P. Gader, M. Mohamed, J. Chiang, Comparison of crisp and fuzzy character neural networks in handwritten word recognition, IEEE Transactions on Fuzzy Systems 3 (3) (1995) 357–364.

[36] C. Parisse, Global word shape processing in off-line recognition of handwriting, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (4) (1996) 460–464.

[37] S. Madhvanath, V. Kripasundar, Pruning large lexicons using generalized word shape descriptors, in: Proceedings of $4^{th}$ International Conference on Document Analysis and Recognition, Vol. 2, Ulm, 1997, pp. 552–555.

[38] M. Zimmermann, J. Mao, Lexicon reduction using key characters in cursive handwritten words, Pattern Recognition Letters 20 (1999) 1297–1304.

[39] J. J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (5) (1994) 550–554.

[40] U.-V. Marti, H. Bunke, A full english sentence database for off-line handwriting recognition, in: Proceedings of $5^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Bangalore, 1999, pp. 705–708.

[41] C. Viard-Gaudin, P. M. Lallican, S. Knerr, P. Binter, The IRESTE on/off (IRONOFF) dual handwriting database, in: Proceedings of $5^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Bangalore, 1999, pp. 455–458.

[42] C. Barriere, R. Plamondon, Human identification of letters in mixed script handwriting: An upper bound on recognition rates, IEEE Transactions on Systems, Man and Cybernetics B 28 (1) (1998) 78–82.

[43] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

[44] R. Bellman, S. Dreyfus, Applied Dynamic Programming, Princeton University Press, 1962.

[45] L. Rabiner, B. H. Huang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[46] W. Chen, P. Gader, H. Shi, Lexicon-driven handwritten word recognition using optimal linear combinations of order statistics, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1) (1999) 77–82.

[47] W. Cheng, P. Gader, Word level discriminative training for handwritten word recognition, in: Proceedings of $7^{th}$ Int. workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 393–402.

[48] Y. Bengio, Markovian models for sequential data, Neural Computing Surveys .

[49] L. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, in: A. Waibel, L. Kai-Fu (Eds.), Readings in Speech Recognition, Morgan Kaufmann, Palo Alto, CA, 1989, pp. 267–296.

[50] M. Gilloux, Hidden Markov Models in handwriting recognition, in: S. Impedovo (Ed.), Fundamentals in Handwriting Recognition, Springer, Berlin - Heidelberg, 1994, pp. 264–288.

[51] L. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state markov chains, Annals of Mathematical Statistics 37 (1966) 1554–1563.

[52] L. Baum, J. Egon, An inequality with applications to statistical estimation for probabilistic function of a markov process and to a model for ecology, Bulletin of American Metheorological Society 73 (1967) 360–363.

[53] L. Baum, G. Sell, Growth functions for transformation on manifolds, Pac. J. Math. 27 (2) (1968) 211–227.

[54] L. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains, Annals of Mathematical Statistics 41 (1) (1970) 164–171.

[55] L. Baum, An inequality and associated maximization technique in statistical estimation for prob-abilistic functions of markov processes, Inequalities 3 (1972) 1–8.

[56] G. Forney, The viterbi algorithm, Proceedings of IEEE 61 (3) (1973) 268–278.

[57] A. Viterbi, Error bounds for convolutional codes and an asimptotically optimal decoding algo-rithm, IEEE Transactions on Information Theory IT-13 (2) (1967) 260–269.

[58] M. Coltheart, K. Rastle, Serial processing in reading aloud: evidence for dual-route models in reading, Journal of Experimental Psychology 20 (1994) 1197–1211.

[59] A. Jacobs, J. Grainger, Models of visual word recognition - sampling the state of the art, Journal of Experimental Psychology 20 (6) (1994) 1311–1334.

[60] J. McClelland, D. Rumelhart, An interactive activation model of context effects in letter percep-tion, Psychological Review 88 (1981) 375–407.

[61] K. Paap, S. Newsome, J. McDonald, R. Schvaneveldt, An activation-verification model for letter and word recognition: the word superiority effect, Psychological Review 89 (1982) 573–594.

[62] M. Côté, E. Lecolinet, M. Cheriet, C. Suen, Building a perception based model for reading cursive script, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 898–901.

[63] M. Coté, M. Cheriet, E. Lecolinet, C. Suen, Automatic reading of cursive scripts using human knowledge, in: Proceedings of $4^{th}$ International Conference on Document Analysis and Recogni-tion, Vol. 1, Ulm, 1997, pp. 107–111.

[64] M. Coté, Utilisation d'un modéle d'accés lexical et de concepts perceptifs pour la reconnaissance d'images de mots cursifs, Ph.D. thesis, Ecole nationale suprieure des télécommunications de Paris, Paris, France (1997).

[65] J. P. Dodel, R. Shinghal, Symbolic/neural recognition of cursive amounts on bank cheques, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 15–18.

[66] D. Guillevic, C. Y. Suen, Cursive script recognition applied to the processing of bank cheques, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 11–14.

[67] D. Guillevic, Unconstrained handwriting recognition applied to the processing of bank cheques, Ph.D. thesis, Concordia University, Montreal, Canada (1995).

[68] C. Olivier, T. Paquet, M. Avila, Y. Lecourtier, Recognition of handwritten words using stochastic models, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 19–23.

[69] G. Saon, Cursive word recognition using a random field based Hidden Markov Model, Interna-tional Journal of Document Analysis and Recognition 1 (1) (1999) 199–208.

[70] G. Kaufmann, H. Bunke, Automated reading of cheque amounts, Pattern Analysis and Applica-tions 3 (2000) 132–141.

[71] D. Guillevic, C. Suen, Hmm word engine recognition, in: Proceedings of $4^{th}$ International Con-ference on Document Analysis and Recognition, Vol. 2, Ulm, 1997, pp. 544–547.

[72] U. Marti, G. Kaufmann, B. H., Cursive script recognition with time delay neural networks using learning hints, in: W. Gerstner, A. Gernoud, M. Hasler, J. Nicoud (Eds.), Artificial Neural Networks - ICANN97, Springer Verlag, 1997, pp. 973–979.

[73] W. Wang, A. Brakensiek, A. Kosmala, G. Rigoll, HMM based high accuracy off-line cursive handwriting recognition by a baseline detection error tolerant feature extraction approach, in: Proceedings of $7^{th}$ Int. workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 209–218.

[74] F. Cohen, Markov random fields for image modelling e analysis, in: Modelling and Applications of Stochastic Processes, Kluwer Academic Press, 1986, pp. 243–272.

[75] V. Di Lecce, A. Dimauro, Guerriero, S. Impedovo, G. Pirlo, A. Salzo, A new hybrid approach for legal amount recognition, in: Proceedings of $7^{th}$ Int. workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 199–208.

[76] J. T. Favata, General word recognition using approximate segment-string matching, in: Proceedings of $4^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Ulm, 1997, pp. 92–96.

[77] M. Shridar, G. Houle, K. F., Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms, in: Proceedings of $4^{th}$ International Conference on Document Analysis and Recognition, Vol. 2, Ulm, 1997, pp. 861–865.

[78] S. Srihari, Handwritten address interpretation: a task of many pattern recognition problems, International Journal of Pattern Recognition and Artificial Intelligence 14 (5) (2000) 663–674.

[79] A. Kundu, Y. He, M. Che, Alternatives to variable duration hmm in handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1275–1280.

[80] M. Chen, A. Kundu, An alternative to variable duration hmm in handwritten word recognition, in: Proceedings of International Workshop on Frontiers in Handwriting Recognition, 1993.

[81] G. Kim, V. Govindaraju, Handwritten word recognition for real time applications, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 24–27.

[82] G. Seni, V. Kripasundar, R. K. Srihari, Generalizing edit distance to incorporate domain information: Handwritten text recognition as a case study, Pattern Recognition .

[83] A. El Yacoubi, J. Bertille, G. M., Conjoined location and recognition of street names within a postal address delivery line, in: Proceedings of $3^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Montreal, 1995, pp. 1024–1027.

[84] S. Madhvanath, E. Kleinberg, V. Govindaraju, S. Srihari, The HOVER system for rapid holistic verification of off-line handwritten phrases, in: Proceedings of $4^{th}$ International Conference on Document Analysis and Recognition, Vol. 2, Ulm, 1997, pp. 855–859.

[85] S. Madhvanath, E. Kleinberg, V. Govindaraju, Holistic verification of handwritten phrases, IEEE Transactions on Pattern Analysis and Machine Intelligence .

[86] J. Park, V. Govindaraju, S. Srihari, Efficient word segmentation driven by unconstrained handwritten phrase recognition, in: Proceedings of $5^{th}$ International Conference on Document Analysis and Recognition, Vol. 1, Bangalore, 1999, pp. 605–608.

[87] U. Marti, H. Bunke, Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system, International Journal of Pattern Recognition and Artificial Intelligence .

[88] B. Yanikoglu, P. Sandon, Off line cursive handwriting recognition using neural networks, in: Proceedings of SPIE Conference on Applications of Artificial Neural Networks, 1993.

[89] U. Marti, H. Bunke, Towards general cursive script recognition, in: Proceedings of Sixth Int. workshop on Frontiers in Handwriting Recognition, Korea, 1998, pp. 379–388.

[90] U. Marti, H. Bunke, Handwritten sentence recognition, in: Proceedings of $15^{th}$ International Conference on Pattern Recognition, Vol. 3, Barcelona, 2000, pp. 467–470.

[91] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1997.

[92] R. Srihari, C. Baltus, Incorporating syntactic constraints in recognizing handwritten sentences, in: Proceedings of International Joint Conference on Artificial Intelligence, Chambery, 1993, pp. 1262–1267.

[93] R. Srihari, Use of lexical and syntactic techniques in recognizing handwritten text, in: Proceedings of ARPA workshop on Human Language Technology, Princeton, 1994, pp. 403–407.