

Automatic reading of cursive scripts using a reading model and perceptual concepts

The PERCEPTO system

M. Côté^{1,2,3,*}, E. Lecolinet¹, M. Cheriet², C.Y. Suen³

¹ ENST, SIG / CNRS URA 820, 46 rue Barrault, F-75634 Paris Cedex 13, France

² E.T.S, Université du Québec, 1100 rue Notre-Dame Ouest, Montréal, Québec H3C 1K3, Canada

³ CENPARMI, Concordia University, Suite GM-606, 1455 de Maisonneuve Ouest, Montréal, Québec H3G 1M8, Canada

Received June 29, 1997 / Revised August 13, 1997

Abstract. This paper presents a model for reading cursive scripts which has an architecture inspired by the behavior of human reading and perceptual concepts. The scope of this study is limited to offline recognition of isolated cursive words. First, this paper describes McClelland and Rumelhart's reading model, which formed the basis of the system. The method's behavior is presented, followed by the main original contributions of our model which are: the development of a new technique for baseline extraction, an architecture based on the chosen reading model (hierarchical, parallel, with local representation and interactive activation mechanism), the use of significant perceptual features in word recognition such as ascenders and descenders, the creation of a fuzzy position concept dealing with the uncertainty of the location of features and letters, and the adaptability of the model to words of different lengths and languages. After a description of our model, new results are presented.

Key words: Offline recognition – Cursive script recognition – Perception – Reading model – Activation

1 Introduction

The advent of the computer has deeply modified our way of interacting with our environment. Even though this machine is able to perform complex calculations and often exceeds human capacity, it remains limited in many other aspects. In fact, communication with a computer through a keyboard is not very natural, and requires much discipline. Unfortunately, even after 30 years of intensive research in this domain, a complete solution to the automatic reading of cursive script has not yet been found. On the other hand, handwriting recognition

can play an important role in future reading systems (Bartneck 1996), and consequently it is a worthy and challenging area of further investigation and research.

Since humans are able to read handwritten texts with apparent ease, it may seem appropriate to base an automatic handwriting reader on human reading models. What features are detected while reading? How do humans access information concerning the meaning of a word? Does perception of a word build up from the perception of its letters, or by its shapes? For many years, researchers in the fields of biology, neurophysiology, cognitive psychology, and linguistics have studied these questions (Taylor and Taylor 1983), and various reading models have resulted from their investigations. Even though these models are still in progress, and many theories defending different ideas are being debated, we believe that we can benefit from their observations for the offline recognition of isolated cursive words. In fact, we consider that features detected while reading play a key role in determining good features for the recognition. Also, we assume that the problem of mental lexical access can influence the architecture to implement the method.

Though models have been proposed to explain the process of mental lexicon access while reading, they mostly rely on printed texts. Few studies have been conducted on the mechanisms involved in the reading of handwriting. In (De Zuniga et al. 1991), the authors conclude that even though the reading of handwritten words differs from printed words at first glance, once cursive normalization has taken place, handwritten and printed words (in lowercase letters) seem to be subject to similar processes. For example, experimental studies suggest that people use word shape to help them recognize words, and this can be useful too in handwriting recognition since the overall shape of a handwritten word follows the same sequence of tall (ascenders: “l”, “t”), short (“o”, “r”, “a”), and projecting letters (descenders: “p”, “j”) as the shape of the word printed in lowercase. Thus, it seems that the underlying concepts

* Presently at INO 369, rue Franquet, Ste-Foy, Québec G1P 4N8, Canada; e-mail: mcote@ino.qc.ca

Correspondence to: M. Côté

of reading models can be adapted to the recognition of cursive scripts.

Dealing with offline cursive scripts does not simplify this adaptation; because cursive writing is fundamentally prone to ambiguity, offline cursive script recognition constitutes a complex problem. Often, letters in the words are poorly written if not missing. Consequently, perfect letter segmentation is impossible, and letter position is not known precisely. This paper presents a new method based on a human reading model, influenced by the work of McClelland and Rumelhart in perception. The scope of this study is limited to the offline recognition of isolated cursive words. Our model, already described in (Côté et al. 1995, 1996b, 1997), shares the following characteristics with that proposed by McClelland and Rumelhart (McClelland and Rumelhart 1981): network with local representation, parallel processing, and activation mechanism (top-down and bottom-up processes).

The organization of the paper is as follows: Sect. 2 briefly talks about contextual analysis methods, Sect. 3 introduces knowledge representation in neural networks, Sect. 4 explains why McClelland and Rumelhart’s reading model was chosen as the foundation of our system, and Sect. 5 describes our method. New results and interpretations are included in Sect. 6, followed by discussions and concluding remarks.

2 Contextual analysis

Cursive script recognition requires contextual analysis due to its large variability. There are two main types of contextual analysis: left-right methods and top-down methods.

In the left-right methods, the contextual analysis is done essentially by dynamic programming, editing distance or by HMM (Chen and Kundu 1993). All the possible combinations of segmentations are computed in order to find the best combination.

In the top-down methods, the parallel information processing is favored and relies on hypothesis generation and validation. Methods using a global strategy are generally of this type (Houle 1993). On the contrary, methods using an analytical strategy of segmentation are usually considered as bottom-up methods (Favata and Srihari 1992).

New methods are now combining top-down and bottom-up strategies in order to build more flexible segmentation techniques (implicit analytical segmentation methods) as in (Bozinovic 1989).

Regarding the way we integrate contextual information into the recognition process, we chose an approach which better fulfils the research results in psychology.

- According to Taylor and Taylor (1993, p. 194):

To recognize handwriting one needs no more than letter features in roughly the right places...

Hence, contextual analysis mainly allows us to benefit from an important redundancy of information. Consequently, the recognition of all the letters of a word is not necessary to identify it.

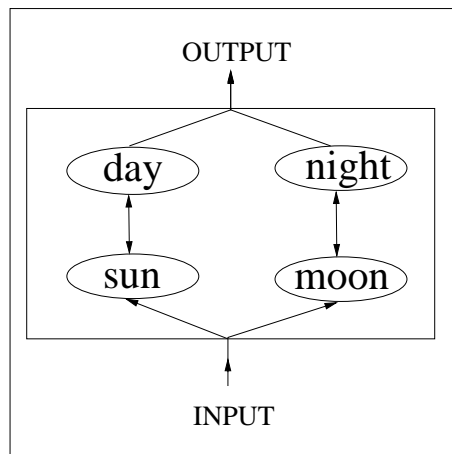


Fig. 1. Network with local representation

- They also mention that the reading of a word is not performed from the left to the right but outside-in.

This is why unlike HMM techniques, our method of contextual analysis is not left-right. It combines bottom-up and top-down approaches together using hypothesis generation and validation.

3 Knowledge representation in neural networks

The connectionist approach occupies an important place in pattern recognition. Several types of neural networks exist, which can be divided into different categories. More generally speaking there are two kinds of systems: *black-box* systems and *transparent* systems. The working mode of the latter are fully explainable while the working mode of the former is totally opaque. One characteristic that we would like to highlight is the way information is coded in the network. According to this point of view, there are two types of networks (Jodouin 1994): those with local representation, and with distributed representation.

- networks with local representation

In this type of network, each cell or neuron corresponds to a specific concept. When the neuron detects the presence of this concept it is active, otherwise it is inactive. This coding is used in networks with no learning phase because it is relatively easy to establish the connections between neurons according to the interpretation that we want to give them. In the example shown in Fig. 1, if we present ‘sun’ as input, the output of the network will always be ‘day’, because we have linked the concept ‘sun’ to the concept ‘day’.

This local coding is simple and very useful when the number of data to be represented is small, and when the data can be described with simple relations. The behavior of this type of network can be explained step by step. It thus falls into the category of “transparent” systems.

- networks with distributed representation

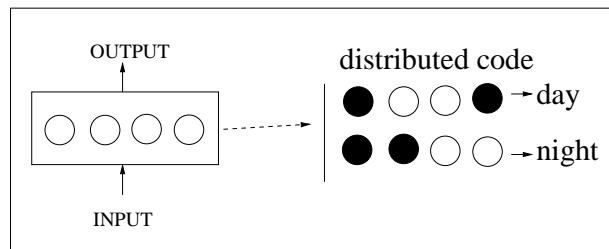


Fig. 2. Network with distributed representation

In this type of network, a concept is distributed over several neurons. A code is associated with each concept. This code has no meaning in itself, and cannot be directly explained as for local representation networks. This is why this type of network has been compared to a “black box”.

From several examples, the network develops an internal representation: it learns configurations or codes specific to each concept. The example in Fig. 2 shows the distributed version of the preceding example. The network learns to answer ‘day’ when we present ‘sun’ as input. It then represents the concept ‘day’ with the following distributed code: black, white, white, black.

At present, most of the networks used for pattern recognition are distributed ones. However, this type of network demands an intensive and time-consuming learning phase. It also needs a large data base. Moreover, it is difficult to explain the behavior of these networks step by step. In addition, it is very difficult to analyze the origin of the recognizer errors and to correct them to improve the system performance.

In the next section, we will show why we chose a local knowledge representation for our system.

4 Psychological aspects

One of the trends in cursive script recognition is to get inspiration from reading models (Higgins and Bramall 1996; Guillevic 1995; Côté et al. 1996a). We agree with Wesolkowski when he says:

Humans are the best cursive word recognizers; therefore, by studying our performance on this task we might be able to set preliminary performance goals for cursive script recognition systems. (Wesolkowski 1996, p. 270)

It has been suggested that reading handwriting in context requires no more features than the first letter and the word shape (Higgins and Bramall 1996). Obviously, the reading will be facilitated by the presence of additional information. Nevertheless, the underlying idea is derived from studies of reading, in order to build efficient automatic reading systems. Word recognition implies the process of visual information, and its representation at the linguistic level. Psychologists call lexical access the processes by which humans associate the image of a word

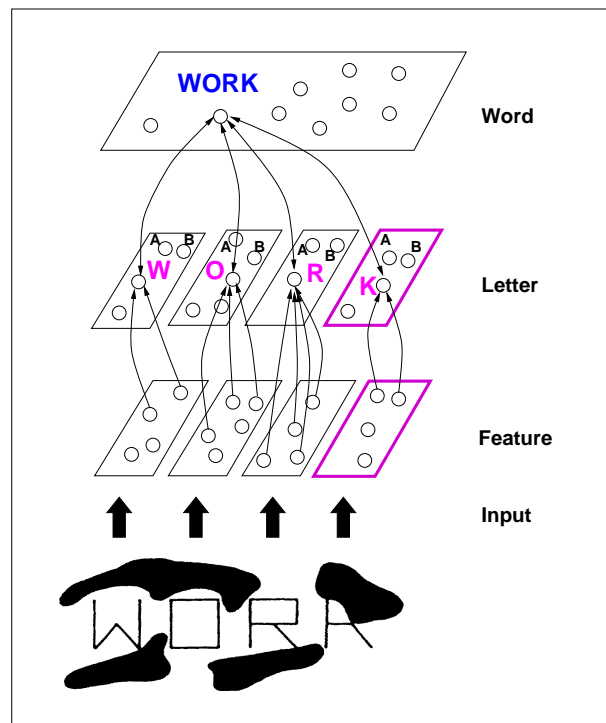


Fig. 3. Interactive Activation Model (McClelland and Rumelhart 1981)

with its meaning. Most lexical access models take into account the orthographic (the way the word is written) and the phonological aspects (the way the word is pronounced) of the word, because both of them are tightly bound. Several models of lexical access have been developed, but up to now there is no final explanation on this matter, and research is continuing (Taft 1991). The question is, which reading model is best suited for the task?

Jacobs and Grainger (Jacobs and Grainger 1994) have published an overview of word reading models, compared and evaluated according to their ability to reproduce the behaviors observed in humans. Three models emerged from the others. The first and the second are representatives of a traditional school, while the last is an adaptation of “brain-style” simulation models.

- *Verification Model* (Paap et al. 1982)

In this model, the visual inputs trigger the activation of some of the words in the lexicon. These activated words constitute a set of candidates. At the verification stage, this set of candidates is then sequentially checked against the sensory representation of the stimulus (as stored in visual memory), until a match is made.

- *Dual route* (Coltheart and Rastle 1994)

This model assumes two main routes for lexical access while reading: one for the words, and one for pseudowords (pronounceable non-words such as ‘REET’ or ‘MAVE’).

- *Interactive Activation Model* (McClelland and Rumelhart 1981)

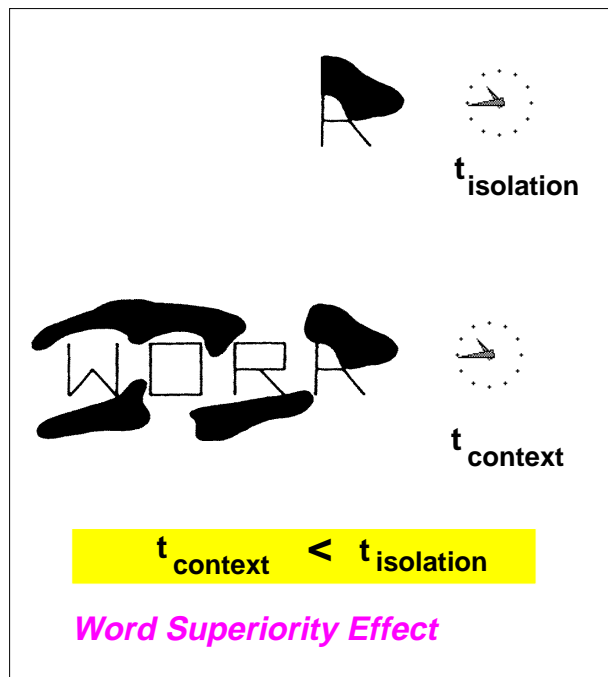


Fig. 4. Experimental observations. $t_{isolation}$ is the time needed for the recognition of a letter in isolation where exists some a priori ambiguity between possible letters such as “A, H, K, or R”; $t_{context}$ is the time needed for the recognition of the letter within a word. In this case, the letter in context is obviously ‘K’ because it corresponds to the word “work”

In this model, which is opposite to the Coltheart and Rastle model, only one route is used for lexical access. Words and pseudowords are processed in the same way.

The output of the interactive-activation model, contrary to Paap’s model, is a single word that has been isolated from a set of active candidates by means of an inhibitory mechanism working on competing units. The verification step is integral to the recognition process. (see Fig. 3)

Since the perceptual approach is our first motivation, we have chosen the McClelland and Rumelhart model, as explained in (Côté 1996a). In fact, this model has been especially designed to mimic the *Word Superiority Effect* (WSE) which is defined as the superiority of letter recognition within a context over the recognition of isolated letters. Figure 4 shows how this effect has been observed when subjects have been asked to recognize a letter either in isolation or in context (i.e. in an existing word).

This *Interactive Activation Model* has been modified many times subsequently. Its local representation has been changed to a distributed one (Seidenberg and McClelland 1989; Rumelhart and McClelland 1986). These ideas have given birth to the current generation of neural networks, which are able to learn new paradigms. However, when comparing both of McClelland’s models (the classical and the distributed one), Forster (Forster 1994) remains skeptical about the pretension of Parallel Distributed Models to explain cognitive functions in terms

of neurons. To him, a simulation is not an explanation. The demonstration that a network works does not give a theoretical explanation of the cognitive processes involved. He concludes:

I suggest that the type of connections needed are effectively equivalent to local representations... I also protest the trend to substitute simulations for theoretical explanations. (Forster 1994, p. 1303)

We are facing the “black box” problem here. Jacobs supports Forster’s affirmation:

evidence from behavioral and brain imaging studies supports a word recognition model closer in spirit to the interactive-activation model (McClelland and Rumelhart) than to more recent (distributed) models (Seidenberg and McClelland, 1989). (Jacobs and Grainger 1994, p. 1326)

Because we choose a local representation instead of a distributed one for our model, we are not in the main stream or current recognition methods, but we still value our personal slant on recognition: human perception. Hence in our network we can introduce a priori knowledge, and we can explain the behavior of the system step by step.

An important point is that we can benefit from this analytic architecture to integrate an implicit segmentation (Casey and Lecolinet 1996). Thus, this architecture gives us the opportunity to improve the segmentation by taking advantage of contextual information. One should note that this is not possible in a network with distributed representation, because it behaves as a classifier which corresponds to a holistic approach. Also, there is no learning phase by connection weight modifications. Another advantage of this knowledge representation is that we do not need an extensive database to train the system.

5 Proposed method

We will introduce the behavior of our method, and underline its main originalities. The different modules of the system are represented in Fig. 5, and will be described in this section.

5.1 Overview

Our method, developed for the offline recognition of isolated cursive words, models the contextual effects that were reported from studies in experimental psychology. We were particularly interested in the Interactive Activation Model proposed by McClelland and Rumelhart, because it is a lexical access model which mimics human perception, and because it accounts for the Word Superiority Effect. Even with inspiration from this reading model, it should be noted that our data and objectives are radically different from those of McClelland and

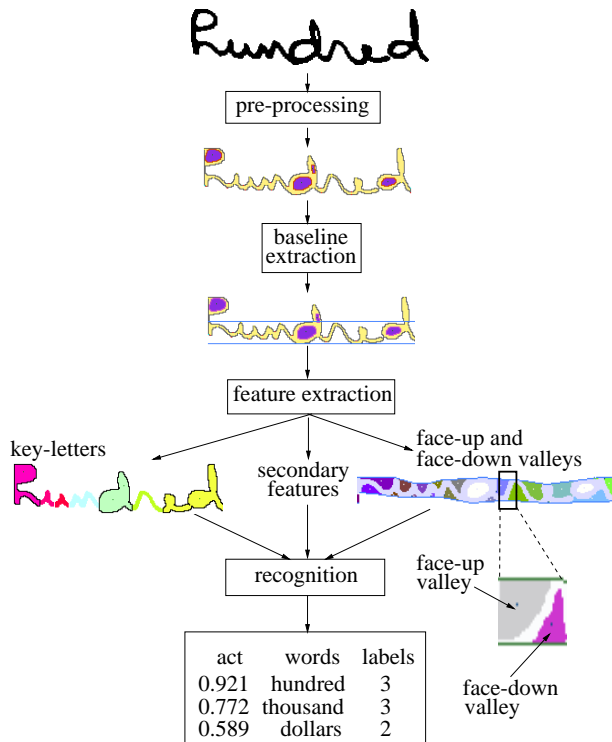


Fig. 5. System overview. In this example, the key-letters are 'h', 'd', 'e' and 'd'

Rumelhart. They use printed words to study human perception; we work with handwritten words. In their case, they build a reading model, whereas in our case, we want to recognize a cursive word.

Hence, our recognition method is based on some ideas presented in the McClelland and Rumelhart model: neural network with local knowledge representation, parallel processing of information, gradual propagation of activation between adjacent levels of cells, following several bottom-up and top-down processes. However, because of the variability of handwriting, we have included in this architecture some characteristics specific to cursive scripts: meaningful features such ascenders and descenders, relative position of letters, fuzzy matching technique, and contextual analysis.

Our system PERCEPTO for cursive word recognition has resulted from these new developments. It is mainly composed of four modules: pre-processing, baseline extraction, feature extraction, and recognition. Fig. 5 illustrates the propagation of information in the system. First of all, a scanned image of an isolated cursive word is given as input. Pre-processing is performed on this input image, including contour extraction, loop detection and identification of local minima. The local minima will be used later during the pre-segmentation of the image into zones, each containing a key-letter (cf., Sect. 5.2.2).

Once pre-processing is complete, the baselines of the word image are found in order to prepare for feature extraction (cf., Sect. 5.2). Three types of features are extracted: primary and secondary features, face-up and face-down valleys.

In the recognition module, a neural network with three layers of neurons identifies the input word from the features extracted through a succession of perceptual cycles (bottom-up and top-down processes) (cf., Sect. 5.5). A fuzzy matching technique identifies a correspondence between the zones in the input image (parts of the image which are recognized) and letters in each word of the lexicon (contextual knowledge) (cf., Sect. 5.7).

The output of the recognition module is a list of candidate words sorted by decreasing order of activation.

5.2 Feature extraction

As explained in the section above, three types of features are extracted in this method, as shown in Fig. 5: primary features, secondary features, and face-up and face-down valleys.

Primary features are used to detect key-letters (letters or part of letters containing an ascender, a descender or a loop in the body of the word) (cf., Sect. 5.2.2). Secondary features are conditional, because they are only detected when they are found in the presence of a prime feature (cf., Sect. 5.2.3). Face-up and face-down valleys are extracted from the background of the image (cf., Sect. 5.2.4). Primary and secondary features are used during the bottom-up processes of the recognition module (cf., Sect. 5.5.1), while face-up and face-down valleys are involved in the top-down processes only (cf., Sect. 5.5.2). Successful identification of these features relies on the baselines of the word image.

5.2.1 Baseline extraction. Baselines split a word into three regions: one for ascenders (region above the superior baseline), one for descenders (region below the inferior baseline), and one for the body of the word (region between the two baselines). We have developed a new method for baseline extraction using entropy, which is described in (Côté 1996c) and illustrated in Fig. 6. This method avoids the correction of the slant of the word, and consequently the baseline follows the word as it is written. We basically compute several histograms for different y projections, and calculate the entropy associated with each of them. We recall that entropy is a measurement of the compactness associated with the density of points given by (1).

$$E = - \sum_i p_i \log(p_i) \quad (1)$$

$$p_i = \frac{N_i}{N} \quad (2)$$

where N_i is the number of pixels with ordinate y_i in the histogram, and N is the total number of pixels in the contour of the word. The probability p_i gives the occurrence of ordinate y_i in the histogram ($\sum_i p_i = 1$).

Entropy E is maximum when all probabilities p_i are equal. It is minimum when all probabilities p_i , except one, are equal to zero. When a distribution is compact,

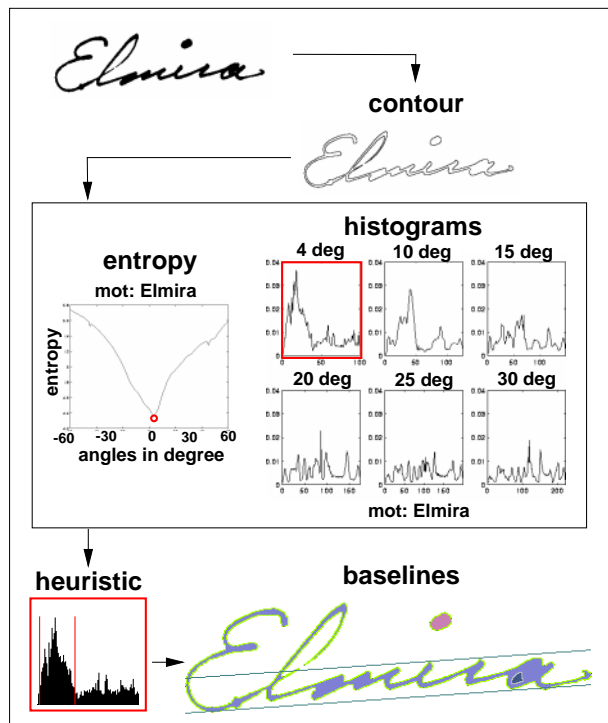


Fig. 6. Baseline extraction method

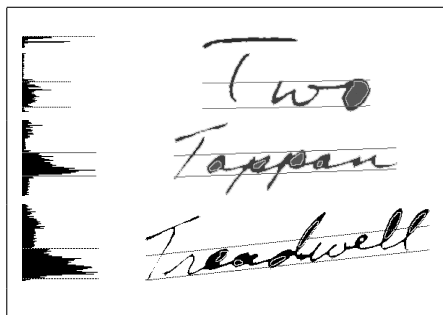


Fig. 7. Baseline extraction for the short word “Two”, for a word with descenders such as “Tappan” and for the slanted word “Treadwell”

its entropy is small. On the contrary, when a distribution is spread, its entropy is large. Consequently, the histogram having the lowest entropy will correspond to the writing direction. With this specific histogram, we then find the thresholds delimiting the body of the word following a simple heuristic. Examples of this baseline extraction technique are given in Fig. 7.

5.2.2 Primary features and key-letters. The primary features are ascenders, descenders, ascender-descenders, and loops within the body of the word. (see Fig. 8) The key-letters (Cheriet and Suen 1993) are the letters (or parts of letters) which are described by these primary features. As these features are considered to be robust (Bouma 1971), they will be marked as the anchor points of the recognition process (Houle et al. 1993). The goal here is not to segment the whole word in its letters but

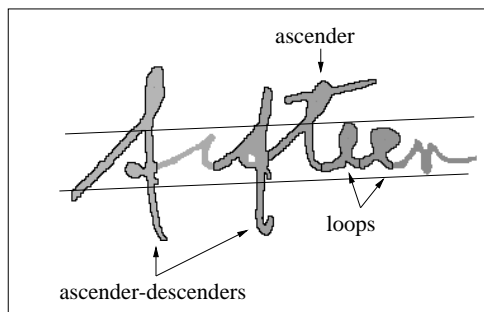


Fig. 8. Primary features. Key-letters have a black contour delimiting letters ‘f’, ‘f’, ‘t’, ‘e’ and ‘e’. Letters ‘i’ and ‘n’ are ignored because they are not described with primary features. They will be sought during the top-down processes

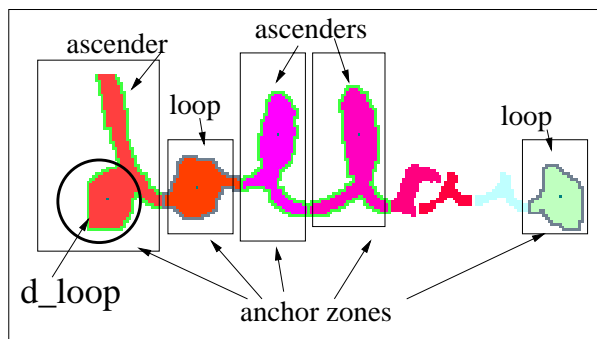


Fig. 9. Conditional features

rather to find the key-letters. First, local minima of the upper contour are found, in order to locate potential ligatures. Although more sophisticated algorithms could be used to find potential segmentation points, we just use here the local minima as first approximation.

Once these potential ligatures are detected, the word is pre-segmented into connected components. The key-letters are the connected components which overlap the ascender and/or descender regions. Connected components detected as loops in the body of the word are also identified as key-letters. Each key-letter determines the width and the location of a zone. These zones form the input to the recognition module during the bottom-up processes.

5.2.3 Secondary features or conditionals. Secondary features, such as b_loop, d_loop or t bars, have their presence attested only when they are found in combination with the primary features; for this reason they are called conditional features, as in (Lecolinet 1994). For example, the feature d_loop is a loop that will only be detected when an ascender is found on its right within the same zone. Tests show that detection of secondary features increases the recognition rate by 4% on the training set. (see Fig. 9)

5.2.4 Face-up and face-down valleys. Here, the background of the image is taken into account. (see Fig. 10) Face-up and face-down valleys are the connected com-

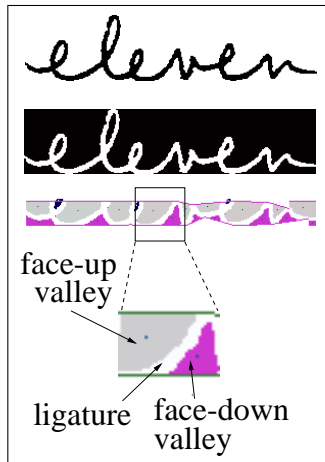


Fig. 10. Face-up and face-down valleys

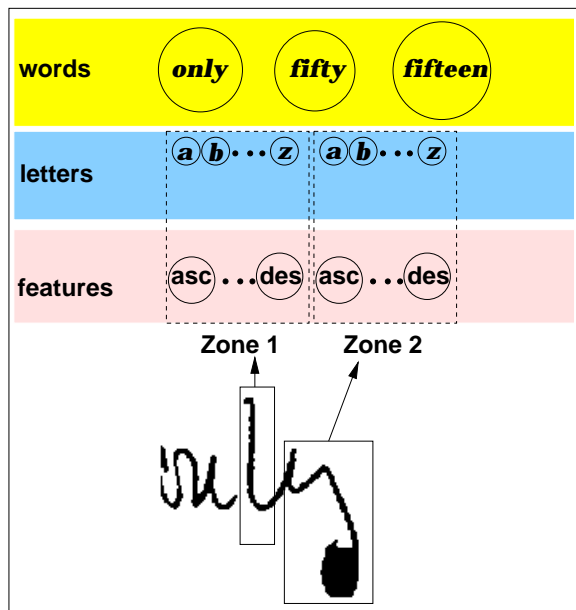


Fig. 11. Three level system

ponents of the background extracted between the lower and the upper contours of the word (Cheriet and Suen 1993). These features are less stable, but will be used in the hypothesis generation, validation and insertion process to find other clues leading to the identity of the target word (cf., Sect. 5.5.2).

5.3 Architecture of the recognition model

Our system is based on three levels of cells, hierarchically organized at feature, letter, and word levels as shown in Fig. 11. We assume that there is a cell for each word in the pre-defined lexicon, as well as for each letter and each feature associated with a given location in the image. Namely, there is a possibility of 32 words, 26 letters and 11 features.

There is a strong link between the number of zones in the image and the number of feature-cells in the first level of the system. The same observation also applies

to the number of letter-cells in the second level of the system.

Connections between adjacent levels are excitatory¹ and bi-directional, except between the feature and the letter level, where the connection is bottom-up only. There are no connections within the same level. This is why a word-cell may be connected to a letter-cell but not to a feature-cell. The cells are pre-linked according to a priori knowledge. Two lexicons link the adjacent levels of cells: a feature-letter lexicon and a letter-word lexicon. Hence, according to these lexicons, the word-cell “two” will be connected to the letter-cell ‘t’ but not the word-cell ‘f’. We call this letter-cell ‘t’, the neighbor of the word-cell “two”. Hence, the word-cell “two” has three neighbors: the letter-cells ‘t’, ‘w’ and ‘o’. For each word in the lexicon, there is a labelling array which links each of the word letters and each of the associated zones in the image.

5.4 Activation states of a cell

In this system, the cells are either active or passive, and their internal energy or *activation* has a value which varies between 0 and 1.

When a cell detects a stimulus, its activation increases and then it can influence its neighbors, which are the cells of the adjacent levels to which it is connected. Hence, the activation of a cell depends not only on its internal energy, but also on its neighbors’ activation. Figure 12 presents the different activation states of a cell which are described below. In this example, letter-cells ‘b’ and ‘f’ have been activated by the feature-cell “ascender”.

- **Deactivated cell:** letter-cell ‘b’ receives a maximal initial activation. This cell has no neighbors. Consequently, its activation cannot be supported and its internal activation decreases until it reaches the resting level (= 0).
- **Active cell:** letter-cell ‘f’ has two neighbors, which gradually increase its activation until the maximal activation value is reached.
- **Inactive cell:** letter-cell ‘o’ will stay minimal in activation because it cannot receive activation from its neighbors. Thus, its activation will stay minimal even

¹ As explained in (Côté et al. 1996a, p. 306), there is no inhibition in this system as in several psycho-physiological systems because of the particular nature of cursive script which means that we are dealing with noisy and unstable information. As in (Bozinovic and Srihari 1989), we *reward occurrence of events but not their absence*. Our particular way of using information is to delay the decision about the identity of the word, and keep all the current hints that will help the system to make a decision. This choice is made at the price of more confusion between similar words. It is a trade-off between refining the word selection, using reliable low level information, and accepting more candidates by being less selective, using unstable low level information. Possible solutions should not be eliminated too early in the decision process

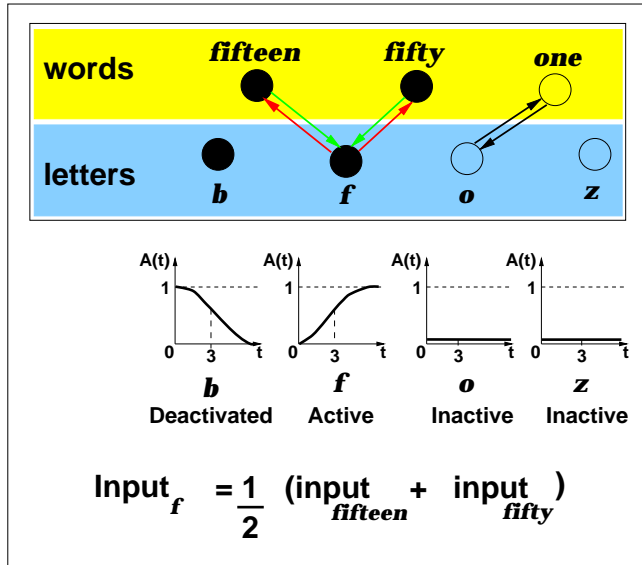


Fig. 12. Activation states of a cell. Each cell is represented by a circle which is black when the cell is active, or white when the cell is not

if it has a neighbor at the adjacent level because the activation of this neighbor is zero. An inactive cell cannot activate a neighbor.

Letter-cell ‘z’ is inactive. Because this cell has no neighbors, its activation will stay at its minimum value.

5.5 Perceptual cycles

Two complementary processes allow the transmission of information within the three levels of the system: bottom-up and top-down processes. A perceptual cycle has been completed when a bottom-up process is followed by a top-down process. During the bottom-up process, the information propagates from the lower (feature) level toward the higher (word) level, and vice versa in the top-down process. In the latter, an implicit segmentation of the unknown word is performed based on the contextual information given by the higher level. We describe each of the processes in the following sections.

5.5.1 Bottom-up process. From the offline image of an unknown handwritten word, meaningful features such as ascenders, descenders, ascender-descenders (as in letter ‘f’), and loops are first extracted. These form the anchors (or also the key-letters) of the image because they constitute the most stable part of the image and therefore, they will be processed in priority.

Since we have the order and relative (but not absolute) position of these key-letters in the image, we model the letter position by a zone which does not necessarily contain one letter. Following the feature extraction, a zone is created for each key-letter.

Extracted features are the input to the recognition system, which will use this information to initiate the

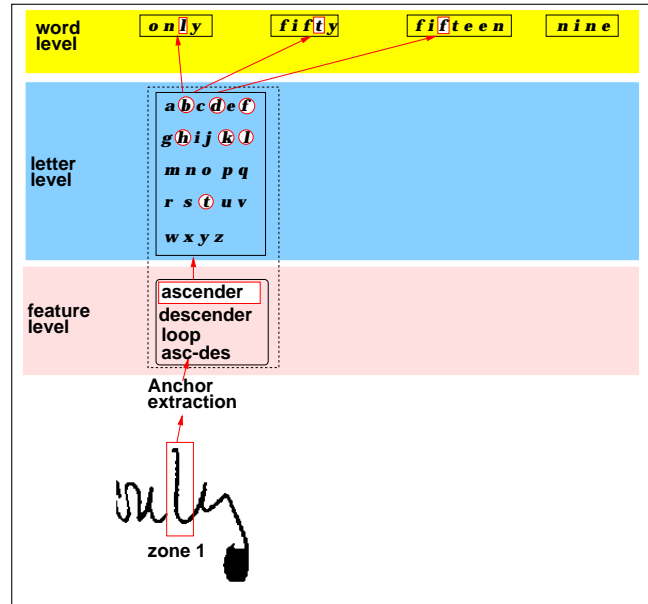


Fig. 13. Bottom-up process. For simplicity, we illustrate here only the processing of the “ascender” of the letter ‘l’. Activated cells are represented by a white background

bottom-up process: the corresponding feature-cells are activated. In the example shown in Fig. 13, for simplicity, we only illustrate the processing of the “ascender” of letter ‘l’. Usually, in this case, both the ascender of letter ‘l’ and the descender of letter ‘y’ are detected, and the bottom-up processes associated with each feature are realized in parallel for both zones. To simplify things, we are concentrating now on the description of the bottom-up process for zone 1. Hence, the detection of feature “ascender” in zone 1 triggers the activation of the feature-cell “ascender” in the region of the network corresponding to this zone. The activation of the feature-cell initiates the propagation of activation toward the adjacent levels. Hence, letter-cells corresponding to zone 1 and connected to this feature-cell “ascender” are also activated.

Following the same process, activated letter-cells trigger the word-cells they are related to. At the end of the bottom-up process, some word-cells are activated and some are not. In this example, the word-cells “only”, “fifty”, and “fifteen” are activated, but the word-cell “nine” is not.

Thus, the features initially detected in the image have initiated the activation of some word-cells which constitute some of the possible solutions in the word identification process.

5.5.2 Top-down process. During the top-down process, contextual information is taken into account in two manners, *feedback* and *insertion*.

During feedback, the propagation of activation spreads from the word-level to the letter-level. The word-cells stimulate the letter-cells, following the same network connections used in the bottom-up process. This feedback will increase the activation of letter-cells that

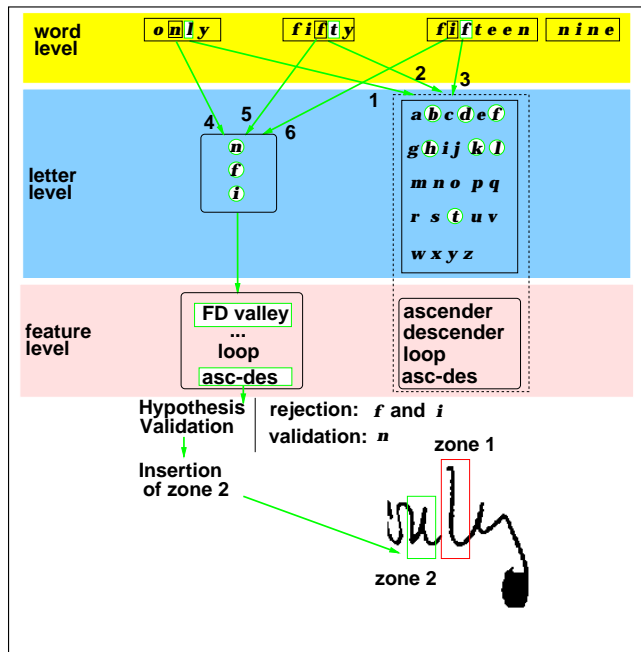


Fig. 14. Top-down process. Feedback: arrows 1, 2 and 3. Insertion: arrows 4, 5 and 6

best match the lexicon. In Fig. 14, the word-cells “only”, “fifty”, and “fifteen” will stimulate the letter-cells which have already contributed to their activation during the bottom-up process (these are the letter-cells ‘b’, ‘d’, ‘f’, ‘h’, ‘k’, ‘l’, and ‘t’ associated with zone 1).

The other process involved in the top-down process is insertion. More precisely, we are talking about the hypothesis generation, validation, and insertion process. Again, we have simplified the diagram shown in Fig. 14, which illustrates the insertion process. The idea is to use contextual information given by the lexicon in order to increase the chances of recognizing the word presented to the system.

The activated word-cells generate letter hypotheses which give some hints about the identity of the unknown letters present in the image. These hypotheses are then checked against the real image. If the features matching the letter hypotheses are present in the unknown word image, the hypotheses are validated and the corresponding cells are activated; if not, they are rejected.

For example, in Fig. 14, the word-cell “only” proposes the letter ‘n’, the word-cell “fifty” the letter ‘f’, and the word-cell “fifteen” the letter ‘i’. The hypotheses sought are ‘n’, ‘f’ and ‘i’ to the left of zone 1. In the image, to the left of this zone, we can find a “face-down valley” feature which validates the presence of letter ‘n’, but does not accept letters ‘f’ and ‘i’ as possibilities. A new zone, zone 2 in this example, is thus created and inserted on the left side of zone 1.

Following this example, we explain the above steps in more detail:

- **Generation:** the system builds a topological representation of the input word image based on information such as: mean width of a zone, beginning and end

of a zone. It also takes into account the estimation of the number of letters between the anchor zones based on the mean width of a zone. Once this topology has been established, the system computes the distance between this target topology and each of the labelled word in the lexicon. The words that are closer to this target topology are retained as word candidates to the hypotheses validation.

- **Validation:** for each word considered as a possibility, we try to validate the retained hypotheses with the input image. A letter in a word will be validated if its features can be found in the image. When a letter is validated, the score associated with the corresponding word is increased. The words which have the highest scores will participate in the insertion process.
- **Insertion:** for each candidate word and for each validated letter within this word, a zone is created and inserted at the appropriate location in the image. In each of these new zones, the feature-detectors corresponding to the features found in this zone are activated. In the next cycle, these features will also contribute to the propagation of activation among the three different levels of the system.

In conclusion, at the end of the top-down process, during feedback, the activation of letter-cells associated with each zone of the image is reinforced. During insertion, new zones are created and inserted.

5.5.3 Complete cycle and saturation. Because the activation increases gradually over time (cf., Sect. 5.6), the cells of the system need several perceptual cycles before they can reach an activation level high enough to decide on the identity of the unknown word.

The sequence of zones at the end of a perceptual cycle constitutes the input for the next cycle. Hence, at the end of a perceptual cycle, hypotheses are validated and the corresponding feature-cells are activated. These newly activated features are added to those features activated from the beginning. This is why, in Fig. 14, the zone 2 created and inserted beside zone 1 will trigger the activation of the feature-cell “face-down valley” in the region of the network associated with this zone. The input for the next cycle will be zone 1 with its feature “ascender”, and zone 2 with its feature “face-down valley”. The detection of these features initiates the activation of the feature-cell “ascender” and the feature-cell “face-down valley” in the regions of the network associated with zones 1 and 2 respectively.

After several perceptual cycles, usually between 6 and 14 cycles, the activation of a word-cell reaches its maximal value, meaning that the system has converged toward a solution. When this happens, we say that the system saturates (cf., Sect. 5.6). It is then possible to establish a list of candidate words sorted in decreasing order of activation. The words having the highest activation values among candidate words are selected as a recognition result array.

5.6 Activation

Now that we have described this system in general, we give a formal definition of activation and explain how the weights of the connections between cells are calculated. Each cell or unit has a momentary activation $A_i(t)$. The unit's activation varies between 0 and 1 in accordance with the following equation:

$$A_i(t + \Delta t) = (1 - \Theta)A_i(t) + E_i(t) \quad (3)$$

with: $A_i(t + \Delta t)$ the new value of the activation of unit i at time $(t + \Delta t)$.

Θ a constant for the unit's decay, set to 0.07.

$E_i(t)$ the effect on unit i at time t due to inputs from its neighbors.

The effect from the neighbors on unit i at time t , $E_i(t)$, is represented by:

$$E_i(t) = n_i(t)(M - A_i(t)) \quad (4)$$

with: $n_i(t)$ the total excitatory influences from the neighbors at time t on unit i .

M the maximum activation level of the unit, set to 1.

where the factor $n_i(t)$ is defined as:

$$n_i(t) = \sum_j \alpha_{ij} a_j(t) \quad (5)$$

with: $a_j(t)$ the activation of an active excitatory neighbor of the unit.

α_{ij} the associated weight constant.

The factor $(M - A_i(t))$ modulates the contribution of the neighbors $n_i(t)$, to keep the input to the unit from driving it beyond some maximum. As can be seen, when the activation of the unit has reached its maximum value (one), the effect of the input is reduced to zero. Thus, the activation is always bounded.

The weights of the connections are not learned. They adapt during the process according to the following formulas:

$$\alpha_{fl} = \frac{1}{NF}, \quad \alpha_{lw} = \mathcal{F}(\Delta)_{lw} \frac{1}{NZ}, \quad \alpha_{wl} = \frac{1}{NW}$$

where NF is the number of features found in the signal for letter L , $\mathcal{F}(\Delta)_{lw}$ is the position coefficient for letter L in word W (the definition follows), NZ is the number of zones found in the signal at time t , and NW is the number of words in the lexicon containing the letter L . Figure 15 shows a general diagram of the model's behavior.

5.7 Labelling and fuzzy matching

During the bottom-up process, we first try to match the activated letter with the related word (letter-word lexicon), and at the right position within the word. To do

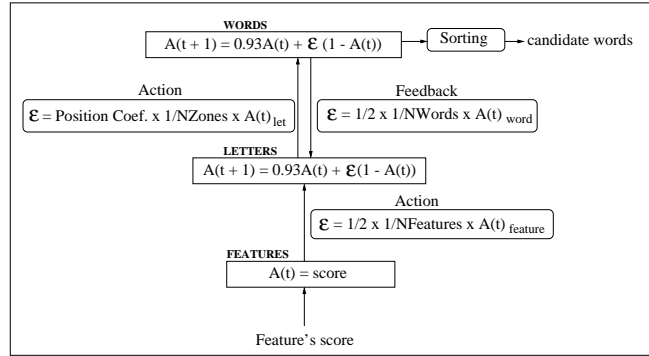


Fig. 15. Behavior of the model

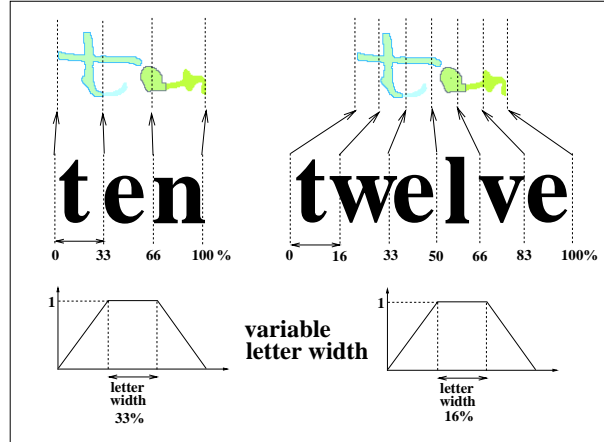


Fig. 16. Projection of the lexicon on the input image; variable letter width

so, a labelling technique has been developed which deals with the relative order of letters and the letter's fuzzy position in a word.

The estimation of the relative letter positions is a very important parameter for successful letter labelling of each word in the lexicon. During the labelling process, we always compare the position of the tested letter in the word with the corresponding position in the image. The letter position in the image is relative to the word image length.

Because the number of letters in the word image is difficult to evaluate, and because an approximate position of a letter in the word image is necessary for a good labelling, an estimation of a letter width is calculated. This estimation is obtained by projecting the lexicon on the input image. Hence during the matching between a zone in the image and a letter in a word of the lexicon, the width of a letter varies according to the number of letters in the considered word and is known a priori. Figure 16 illustrates the mapping used.

Since the letter position in the image is not known precisely, we introduce a position coefficient (real number between 0 and 1). When its value is 1, we consider that the letter in the word corresponds exactly to the letter in the image. On the contrary, a 0 value means that the letter in the word does not correspond at all to the letter in the image. Between these two limits, this

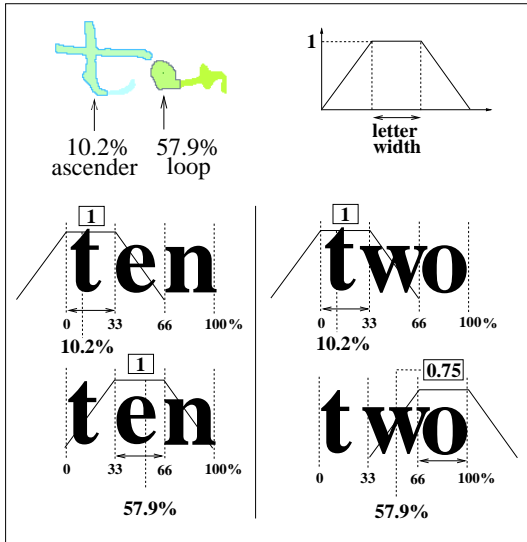


Fig. 17. Fuzzy matching. The position coefficient values are inside the boxes. For the letter ‘t’ in the lexicon words ‘ten’ and ‘two’ the position coefficient is evaluated to 1 in both cases. All the features detected in the image are attributed a relative position (%)

coefficient varies as a fuzzy function. One should note that this fuzzy function depends on the letter width of the image word. Since we try to match each word of the lexicon with the input image, the fuzzy function will vary according to the length of the lexicon word candidate for labelling. This flexible fuzzy function enables the compensation of any distortions that may appear in the image because its particular shape takes into account more than one letter.

Figure 17 shows how the fuzzy matching function is superimposed on each letter candidate for a match with a corresponding zone in the image, in order to obtain the position coefficient. In this example, an ascender and a loop have been found at relative positions of 10.2% and 57.9% respectively in the image of the word “ten”. The ascender at 10.2% is compared with the ‘t’ of the lexicon words “ten” and “two”. The loop at 57.9% is compared with letter ‘e’ of the word “ten” and letter ‘o’ of the word “two”. In this latter case, the position coefficient evaluated at a position of 57.9% has a value of 0.75, because the matching function decreases between positions 33% and 66% when it is centered on the letter ‘o’ associated with the loop.

6 Experiments and results

Two types of results will be discussed in this section: qualitative and quantitative. The former validates the behavior of the model, the latter validates the pattern recognition task performed by our system.

6.1 Qualitative results

After implementation of our method, results have been obtained on real images. Examples of the outputs pro-

act	words	labels	act	words	labels
0.923	fifteen	6	0.924	only	3
0.591	sixteen	4	0.512	forty	2
0.516	eighteen	4	0.512	fifty	2
0.512	fourteen	4	0.366	dollars	2
0.423	dollars	3	0.364	thirty	2
0.370	nineteen	3	0.364	twenty	2
0.363	thirteen	3	0.235	four	1

Fig. 18. Output lists of word candidates for the word images “fifteen”, and “only”. “act” is the activation of the word, “labels” is the number of zones matched with the word

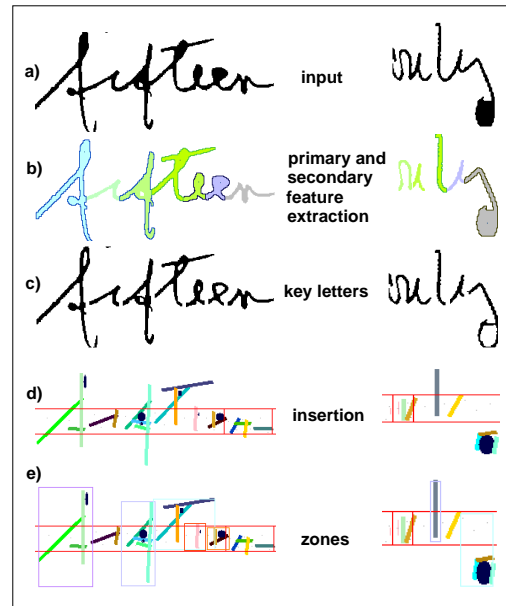


Fig. 19. Words “fifteen”, and “only” **a** before processing, **b** after feature extraction, **c** after key letter extraction, **d** zones created after insertion process, **e** zones created after the bottom-up and top-down processes

duced by our system are shown in Figs. 18 and 19, for the words “fifteen” and “only”. The value associated with each word in the output list is an activation value. The higher the value, the higher the likelihood that this word corresponds to the target word.

In the images of Fig. 19d,e, the segments are the representation of strokes and ascenders. The loops are represented by circles. In the images of Fig. 19e, each zone is represented by a box. As we can observe in Fig. 19b, the meaningful features such as ascenders, descenders and loops have been detected and validated.

The activation curves in Fig. 20 show the behavior of the most highly activated cells (not all the cells are represented here) involved in the recognition process of the words “fifteen” and “only” respectively.

In Fig. 20a, candidate words “fifteen” and “only” are more highly activated because they have more letters matched than their competitors. One should notice that “fifteen”, “sixteen”, “eighteen”, and “fourteen” belong to the same perception family, because they share the same global shape. This is true also for the words

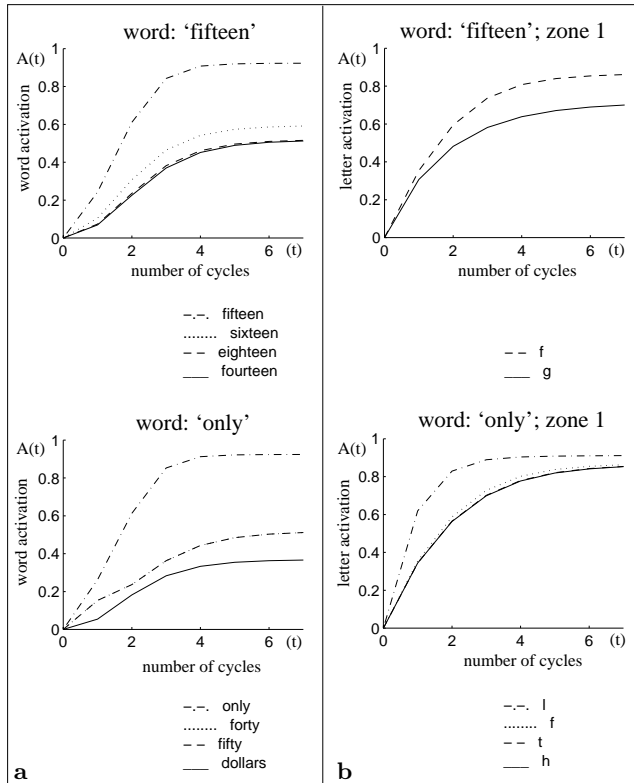


Fig. 20. **a** word activation curves; **b** letter activation curves; for “fifteen”, zone 1 corresponds to letter ‘f’, for “only”, zone 1 corresponds to letter ‘l’

“only”, “forty”, and “fifty”. Hence, the activation curves for the words “forty” and “fifty” are overlapped.

In Fig. 20b, letters ‘f’ and ‘g’ of zone 1, at the beginning of the word “fifteen”, and the letters ‘l’, ‘f’, ‘t’ and ‘h’ of zone 1, in the middle of the word “only”, are in competition, because they have a common feature, namely a descender in the first example, and an ascender in the second example respectively. Since few words in the lexicon begin with letter ‘g’ in the first case, letter ‘f’ receives more activation. In the second example, letters ‘l’, ‘f’, ‘t’, and ‘h’ are in competition, but the feedback of the word-cell ‘only’ to letter-cell ‘l’ enables this letter to win the competition. This fact illustrates very well the “Word Superiority Effect” described by our equation of activation. Indeed, when a cell is activated, it receives more stimuli from its neighbors at adjacent levels, and therefore its activation increases. Moreover, these results are coherent with our psycho-perceptual model because the errors made by the system mimic those observed in humans (confused words are part of a same perceptual family).



Fig. 21. Some word samples from the CENPARMI database

Table 1. Cursive word recognition results (% correct in top N choices)

N (%)	1	2	5	10
Training set	76.1	88	91.8	94
Testing set	73.6	81	89.4	92.7

6.2 Quantitative results

6.2.1 Database. At CENPARMI, where this research has been partly conducted, a large database has been built for a project aiming at the recognition of the legal amount of cheques (Lam et al. 1995). This database has been created from 2500 handwritten cheques written in English, and 1900 handwritten cheques written in French. The number of writers is estimated to be close to 800 for the English cheques, and close to 600 for the French cheques. In this database, 7837 English words and 7135 French words are available. Examples of some typical words found in the database are shown in Fig. 21.

We decided to experiment with this database, first because it was already available and can be a common basis for performance evaluation of different methods and second because of compatibility advantages if this method was to be integrated into a larger framework combining different experts (Suen et al. 1992).

6.2.2 Testing conditions and parameters. The word recognizer has been trained on a small set of 184 images, and tested on a set of 2929 images. None of these images contain capital letters, because the current version of the system is restricted to lowercase letters. The lexicon used for the tests includes 32 English cursive words with 3–9 letters.

Table 1 gives the results obtained for the recognition of isolated cursive words without hypothesis generation, validation or insertion (we will discuss why in Sect. 6.2.3).

As might be expected, the words described with a larger number of anchor features are more often properly recognized. This facilitates the detection of long words, and increases the confidence associated with the recognition of these words as shown in Table 2. On the contrary, short words are more difficult to recognize, espe-

Table 2. Cursive word recognition results per class of word length for the testing set (% correct in top N choices)

Word length	1	2	5	10
3	66.3	69.3	81.4	85.3
4	71.1	78.9	84.2	86.6
5	69.8	79.5	86	90.5
6	77.2	87.5	96	98.2
7	79.8	87.6	96	99.2
8	81.1	90.3	96.3	98.9
9	69	82.8	96.6	100

Table 3. Recognition results for words without extensions with the testing set

Words	1	2	5	10
One	51	57.6	60.9	64.1
Six	30.1	34.2	38.4	43.8
Nine	46.8	48.1	50.6	57
Seven	42.2	49.4	60.2	71.1
Totals	43.1	48	53.2	59.6

Table 4. Recognition results with the testing set with and without “no extension words”

N (%)	1	2	5	10
With	73.6	81	89.4	92.7
Without	77.4	85.1	94	96.9

cially words such as “one” and “nine”, because they do not contain any ascenders or descenders, as we can see in Table 3. These words are responsible for a decrease by 4% in the recognition rates, as we can observe in Table 4. This is why, in these cases, we should extract other features to improve the recognition rate.

When we analyse the experimental results, we observe that there are two main sources of errors. The first one is related to pre-segmentation problems, and does not occur frequently. The second one occurs more often, and is caused by a bad detection of the primary features. For both cases, the information given to the system is erroneous from the beginning. The propagation of this information during the bottom-up process leads to a bad labelling of the lexicon words. Moreover, during the feedback process, these words will contribute to the reinforcement of bad choices.

These facts suggest that we should add an independent labelling mechanism based on contextual information, in order to be able to reconsider the erroneous labelling proposed by the bottom-up process. The use of a rejection threshold may also be part of the solution.

We also realise that our primary feature detection algorithm is robust, even though these features are not always precise enough for difficult cases.

6.2.3 Insertion. As we can see in Table 5, hypothesis insertion does not necessarily increase the recognition rates for all words.

In fact, the insertion process will improve the recognition rates only when the number of zones matched for the target word is increased. Otherwise, a wrong word

Table 5. Results with and without insertion for the training set

N (%)	1	2	5	10
Without insertion	76.1	88	91.8	94
With insertion	69	83	89	94

will be suggested as solution. The following observations may explain this situation:

- hypothesis generation is suggested by the most highly activated word-cells.
When the target word is not in this group of words, the recognition may diverge because it is based on wrong decisions.
- estimation of the topology is erroneous.
The topology estimation depends on the evaluation of the number of letters between the anchor zones, which is calculated from the mean width of a key-letter. Since this measurement is not always accurate, the estimation may be erroneous.
- features looked for during the validation process are too vague.
Consequently, almost all hypotheses will then be accepted. This will lead to a bad hypothesis choice at the validation step, even though the target word is part of the possible word list.

Thus, the insertion process is not efficient to increase the recognition of words already badly recognized. However, it increases the discrimination of the words which have a good activation level.

7 Conclusion and perspectives

After an overview of our method, where we underlined the major contributions of our system and justified the choice of our architecture, we have tested the model on a training set of 184 images and a testing set of 2929 images of English cursive words from the CENPARMI database. The recognition rates obtained for the learning set are 76% (top choice) and 92% (top five), and for the testing set 74% (top choice) and 89% (top five). In real applications, such as mail sorting of cheque recognition, syntactic information can be used to disambiguate the recognition of the words when identifying the whole sentence. Thus, it is reasonable to consider several choices for the recognition of each word of the sentence. Moreover, baseline extraction is usually more accurate when considering sentences rather than isolated words. This is especially true in case of small words such as “one” that does not contain ascenders nor descenders. Finally, it can be noticed that the lexicon used for the tests contains words (essentially numbers) with quite a similar structure. In sentence recognition, the syntax would help to compensate for the ambiguity relative to this type of lexicon.

We have shown that the method is operational and that it has the expected behavior (cf., Figs. 18–20), i.e., that it behaves according to the perceptual concepts

studied. Errors made by the system are not incoherent but mimic in some manner those observed in humans because words which are confused are part of the same perceptual family. Considering that humans recognize words in sentences by using contextual analysis, it is reasonable to think that more than one possibility is evaluated during the recognition of isolated words.

Consequently, we have validated our psycho-perceptual approach for offline recognition of isolated cursive words. The main original issues of our model are summarized below:

1. An architecture based on a reading model: hierarchical, parallel, with local representation and interactive activation mechanism
2. Significant perceptual features in word recognition, such as ascenders and descenders
3. Fuzzy position concept, dealing with the location uncertainty of features and letters
4. Adaptability of the model to words of different lengths and from different languages.

Solutions are suggested below in order to further improve this method. Bad feature detection may be handled by improving the primary feature detection, and by using a labelling mechanism independent of the one resulting from the bottom-up process. Additional features could also improve the insertion module. Moreover, the reliability of the system could be improved if rejection is considered. Finally, a hybrid architecture could be envisaged. It would be interesting to combine a local knowledge representation with a distributed one. The distributed knowledge representation (neural nets with learning phase) would be used between the feature and letter levels in order to improve letter recognition, while the local knowledge representation would be used for contextual analysis between the word and letter levels (analytical approach).

Acknowledgements. We would like to thank Professor Claudie Faure and Ms. Christine Nadal for their assistance, and IRIS, the National Networks of Centres of Excellence of Canada, Natural Sciences and Engineering Research of Canada, the Ministry of Education of Quebec, the *Fond de développement académique du réseau du Québec*, and the *Centre de coopération inter-universitaire franco-québécoise* for their financial funding.

References

1. Bartneck N (1996) The role of handwriting recognition in future reading systems. In: Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition. University of Essex, UK, pp 147–176
2. Bouma H (1971) Visual recognition of isolated lower-case letters. *Vision Research* 11:459–474
3. Bozinovic RM, Srihari SH (1989) Off-line cursive script word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11:68–83
4. Casey RG, Lecolinet E (1996) A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(7):690–706
5. Chen MY, Kundu A (1993) An alternative to variable duration HMM in handwritten word recognition. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp 82–91
6. Cheriet M, Suen CY (1993) Extraction of key letters for cursive script recognition. *Pattern Recognition Letters* 11:1009–1017
7. Coltheart M, Rastle K (1994) Serial processing in reading aloud: Evidence for dual-route models of reading. *J. Experimental Psychology: Human Perception and Performance* 20:1197–1211
8. Côté M, Lecolinet E, Cheriet M, Suen CY (1995) Building a perception based model for reading cursive script. In: Proceedings Third ICDAR 95, Vol. II. Montréal, Canada, pp 898–901
9. Côté M, Lecolinet E, Cheriet M, Suen CY (1996a) Using reading models for cursive script recognition. In: Simner ML, Leedham CG, Thomassen AJWM (eds) *Handwriting and Drawing Research: Basic and Applied Issues*. IOS Press, Amsterdam, pp 299–313
10. Côté M, Lecolinet E, Cheriet M, Suen CY (1996b) Lecture automatique d'écriture cursive utilisant des concepts perceptuels. In: Actes de congrès de l'Association canadienne-française pour l'avancement de la science. Montréal, Canada
11. Côté M, Cheriet M, Lecolinet E, Suen CY (1996c) Détection des lignes de base de mots cursifs à l'aide de l'entropie. In: Actes de congrès de l'Association canadienne-française pour l'avancement de la science. Montréal, Canada
12. Côté M, Cheriet M, Lecolinet E, Suen CY (1997) Automatic reading of cursive scripts using human knowledge. In: Proceedings Fourth ICDAR 97. Ulm, Germany, pp 107–111
13. De Zuniga CM, Humphreys GW, Evett LJ (1991) Additive and interactive effects of repetition, degradation, and word frequency in reading of handwriting. In: Besner D, Humphreys GW (eds) *Basic processes in reading: Visual word recognition*. Lawrence Erlbaum, Hillsdale, NJ, pp 10–33
14. Favata JT, Srihari SN (1992) Recognition of general handwritten words using a hypothesis generation and reduction methodology. In: Proceedings of the USPS Advanced Technology Conference, pp 237–251
15. Forster KI (1994) Computational modeling and elementary process analysis in visual word recognition. *J. Experimental Psychology: Human Perception and Performance; Special Section: Modeling Visual Word Recognition* 20(6):1292–1310
16. Guillevic D (1995) Unconstrained Handwriting Recognition applied to the Processing of Bank Cheques. PhD thesis, Department of Computer Science, Concordia University, Montréal, Canada
17. Higgins C, Bramall P (1996) An on-line cursive script recognition system. In: Simner ML, Leedham CG, Thomassen AJWM (eds) *Handwriting and Drawing Research: Basic and Applied Issues*. IOS Press, Amsterdam, pp 285–298
18. Houle G, Radelar C, Resnick S, Bock P (1993) Handwritten word recognition using collective learning systems theory. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp 92–101

19. Jacobs AM, Grainger J (1994) Models of visual word recognition— sampling the state of the art. *J. Experimental Psychology: Human Perception and Performance; Special Section: Modeling Visual Word Recognition* 20, 6:1311–1334
20. Jodouin JF (1994) *Les réseaux de neurones*. Hermès, Paris, France
21. Lam L, Suen CY, Guillevic D, Strathy NW, Cheriet M, Liu K, Said JN (1995) Automatic processing of information on cheques. In: *Proceedings of the International Conference on Systems, Man and Cybernetics*. Vancouver, Canada, pp 2353–2358
22. Lecolinet E (1994) Cursive script recognition by backward matching. In: Faure C, Keuss P, Lorette G, Vinter A (eds) *Advances in Handwriting and Drawing: A Multidisciplinary Approach*. Europia, Paris, pp 117–135
23. McClelland JL, Rumelhart DE (1981) An interactive activation model of context effects in letter perception. *Psychological Review* 88:375–407
24. Paap K, Newsome SL, McDonald JE, Schvaneveldt RW (1982) An activation-verification model for letter and word recognition: the word superiority effect. *Psychological Review* 89:573–594
25. Rumelhart DE, McClelland JL (eds) (1986) *Parallel Distributed Processing; Explorations in the Microstructure of Cognition*, vol. 1: Foundations. The MIT Press, Cambridge
26. Seidenberg S, McClelland JL (1989) A distributed, developmental model of word recognition and naming. *Psychological Review* 96:523–4568
27. Suen CY, Nadal C, Legault R, Mai TA, Lam L (1992) Computer recognition of unconstrained handwritten numerals. *Proceedings of the IEEE* 80(7):1162–1179
28. Taft M (1991) *Reading and the Mental Lexicon*. Lawrence Erlbaum, Hillsdale, NJ
29. Taylor I, Taylor MM (1983) *The Psychology of Reading*, Ch. 9: Letter and Word Recognition. Academic Press, New York
30. Wesolkowski S (1996) Cursive script recognition: a survey. In: Simner ML, Leedham CG, Thomassen AJWM (eds) *Handwriting and Drawing Research: Basic and Applied Issues*. IOS Press, Amsterdam, pp 267–284

Myriam Côté received her B. Eng. degree from Université Laval, Québec, Canada in 1989. In 1992, she obtained a M.Sc. degree in optics and photonics from Université de Paris XI, France, and in 1993, a M.Sc. degree in artificial intelligence and pattern recognition from Université de Paris VI, France. In 1997, she received her Ph.D degree in Telecommunications from École Nationale Supérieure des Télécommunications, France. She is now working as a postdoctoral fellow at INO, Québec, Canada. Her research interests include pattern recognition, perception, optics and image synthesis.

Eric Lecolinet received his Ph.D degree in Computer Science from Université Pierre et Marie Curie, Paris, France in 1990. He has been working on OCR and Cursive Script Recognition at Matra, France from 1987 to 1990 and at the IBM Almaden Research Center, California from 1990 to 1992. He is currently an Associate Professor at the École Nationale Supérieure des Télécommunications (ENST), Paris, France and is a member of the associated research unit of CNRS, URA 820. His research interests include pattern recognition, artificial intelligence and human-computer interaction. Dr Lecolinet is a member of the IEEE and has published more than 30 papers on these subjects.

Mohamed Cheriet received his Ph.D. degree in Computer Science from Université de Paris 6 (France) in 1988. From 1988 to 1990, he worked as a Research Assistant at LAFORIA/CNRS laboratory. He then joined CENPARMI at Concordia University in Montreal, where he worked as a postdoctoral fellow for two years. He was appointed Assistant Professor in 1992 and Associate Professor in 1996 in the department of Automated Production Engineering at the École de Technologie Supérieure de l'Université du Québec in Montreal. His research focuses on image processing, pattern recognition, character recognition, text processing, handwritten documents analysis and recognition, and perception. Dr Cheriet is a member of IEEE and an active member at CENPARMI. He has published over 30 technical papers in the field.

Ching Y. Suen joined the Department of Computer Science of Concordia University, Montreal, in 1972. Presently he is the Director of CENPARMI, the Centre for Pattern Recognition and Machine Intelligence of Concordia. Prof. Suen is the author/editor of 11 books on subjects ranging from computer Vision and Shape recognition, Handwriting Recognition and Expert Systems, to Computational Analysis of Mandarin and Chinese. Dr. Suen is the author of more than 250 papers on these subjects. A fellow of the IEEE, IAPR, and the Royal Society of Canada, Dr. Suen is an Associate Editor of several journals related to Pattern Recognition. He is the Past President of the Canadian Image Processing and Pattern Recognition Society, and the Chinese Language Computer Society. He is Founder or Co-founder of several conferences, including Vision Interface, IWFHR, and ICDAR. Prof. Suen is the recipient of several awards for outstanding contributions to pattern recognition, expert systems, and computational linguistics.