

# **A Robust Analytical Approach for Handwritten Word Recognition**

G. Congedo, G. Dimauro, S. Impedovo, G. Pirlo, D. Sfregola

*Dipartimento di Informatica - Università di Bari  
Via Amendola 173 - 70126 Bari - Italy*

**Abstract.** This paper presents a system for handwritten word recognition based on an analytical approach. Through a fast segmentation procedure, basic strokes are detected and used to recognize the word by a bottom-up approach. The system has been tested on the worded amounts on bankchecks. The experimental results show the effectiveness of the approach.

## **1 Introduction**

Handwritten documents are generally massively used to transfer information in administrative and technical offices. In fact, from tax offices to banks, from insurance companies to health offices, and everywhere handwriting is used, handwriting recognition is required. In these environments a wide number of activities could receive a significant support by systems able to handle handwritten data forms [1,2].

Unfortunately automatic document analysis and processing involves a wide range of tasks, ranging from intelligent layout analysis to typewritten and handwritten character and word recognition, and for many of them the scientific community is still far to obtain conclusive solutions. Among the others, handwritten word recognition involves the integration of several different aspects in the field of pattern recognition as well as in the field of the exploitation of knowledge at several levels [3].

In this paper a new system for handwritten word recognition is presented. The system uses an analytical approach based on the consideration that each character consists of a sequence of basic strokes: humps, cusps, oriented lines, circles, semicircles, etc.; therefore each word can rightly be considered as a sequence of basic strokes [4]. Therefore the process of word recognition here proposed follows four main phases: preprocessing, segmentation of the word into basic strokes, feature extraction and stroke classification, and recognition of the word. The experimental results have been carried out testing the system on the basic words extracted from Italian bankchecks. The results point out the robustness of the approach and demonstrate the effectiveness of the design strategies used for this system. These strategies mainly consist in performing recognition at the last classification stages in a closed-loop procedure and using the complete a-priori lexical knowledge available in the specific domain of the application.

**ACKNOWLEDGEMENT:** This work was supported by MURST 60% - 1994, Italy.

## 2 Preprocessing

The preprocessing phase involves three successive steps: smoothing, word zones detection and noise removal. Smoothing is carried out by the traditional algorithms which work on the contour of the image [5]. Fig. 1 shows an example of handwritten word before and after smoothing.

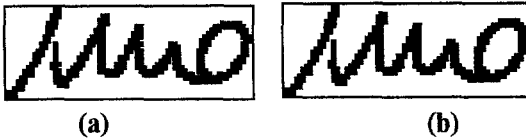


Fig. 1. Examples of word before (a) and after (b) smoothing.

Word zones identification is accomplished by detecting the Upper Basic Line (UBL) and the Lower Basic Line (LBL) by the analysis of the horizontal histogram [6]. This approach allows to detect three basic zones of the word: the "middle zone", the "upper zone" and the "lower zone". The "middle zone" is delimited by the UBL and the LBL and is always present. The "upper zone" is the part of the word above the UBL and indicates the existence of ascending pulls (like those of characters "l", "d", "t") or dots of the letters "i". The "lower zone" is the part of the word below the LBL and indicates the existence of descending pulls (like those of character "q"). Figure 2 shows the "middle zone", the "upper zone", and the "lower zone" detected by this approach on an input word. Noise removal is carried out by detecting noisy patterns like spurious points in the image. At this purpose, a connected component detector is used and isolated components of few pixels are removed in an automatic way [5].

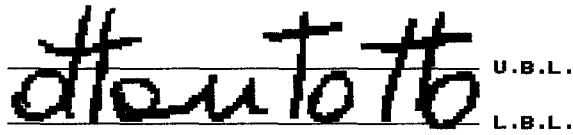


Fig. 2. Word zone detection.

## 3 Segmentation

The objective of the segmentation phase is to obtain a representation of the word as a sequence of basic strokes. A stroke is here defined as a pattern between two consecutive local minima in the upper contour of the word. Therefore, stroke detection starts from the detection of the local maxima and minima in the vertical direction of the word. Successively the cutting line  $r$  is defined according to the direction of the bisecting line of  $s$  and  $t$  as shown in figure 3, where:  $M$  is the local minimum candidate for splitting,  $P$  is the local maximum preceding  $M$  in the upper contour,  $Q$  is the local maximum following  $M$  in the upper contour,  $s$  is the straight line connecting  $P$  and  $M$ ,  $t$  the

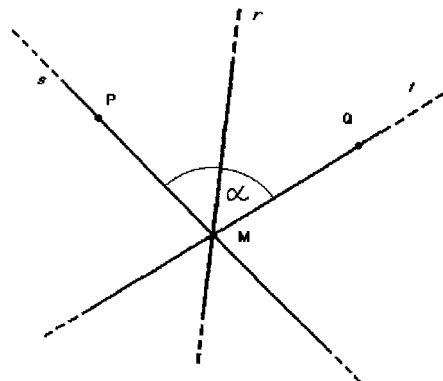


Fig. 3. Computation of the cutting angle.



Fig. 4. A segmented word.

width), and zone of the stroke. The positions are: *lower, middle- lower, middle, middle-upper, upper.*

straight line connecting Q and M. Fig. 4 shows a typical result of the segmentation procedure. For each stroke of the word some features like position, dimension (height and

#### 4 Feature extraction and stroke classification

After segmentation, each stroke is normalized in a matrix of 45x45 pixels. The matrix is then divided in nine zones of equal size as fig. 5 shows and for each zone a feature vector is detected. Specifically, for each pixel of the contour, the local slope of the contour is computed by the analysis of adjacent contour pixels. Finally, the number of contour pixels with the same slope is counted. Eight basic slopes are considered as fig. 6 shows.

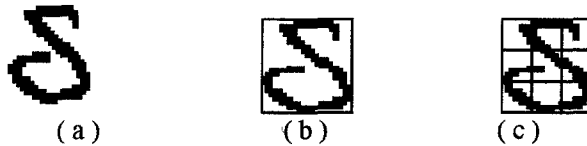


Fig. 5. (a) A stroke, (b) The stroke after normalization, (c) The nine zones of the normalization matrix.

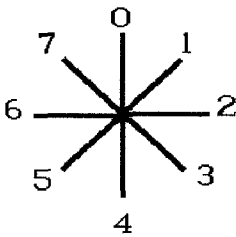


Fig. 6. Basic slopes.

Consequently, for each zone a feature vector of 8 elements is extracted and the feature vector  $v$  of 72-elements for the entire stroke is obtained by connecting the feature vectors of each zone.

In the learning phase, the stroke feature vectors are used, for the selection of the classes of basic strokes by a supervised classification process. For each class, the vectors of the averages and standard deviations of the feature of the elements in the class are computed. A further information stored for each class in the feature vector is the position of the strokes belonging to the class inside the word. Successively, a suitable grouping of the classes detected into groups of basic shapes (humps, cusps, circles, etc.) is carried out manually by a human operator. The use of the groups allows to simplify the word description code and to improve the recognition process.

In the recognition phase, the test word is splitted into strokes and each stroke is firstly classified. Stroke classification is based on the similarity measure provided by the following distance:

$$d(v, \mu^j, \sigma^j) = \frac{1}{72} \sum_{i=1}^{72} \frac{|v_i - \mu_i^j|}{\sigma_i^j},$$

where:  $v = (v_1, v_2, \dots, v_{72})$  is the feature vector of the stroke that must be classified,  $\mu^j = (\mu_1^j, \mu_2^j, \dots, \mu_{72}^j)$  is the mean feature vector and  $\sigma^j = (\sigma_1^j, \sigma_2^j, \dots, \sigma_{72}^j)$  is the vector of the standard deviations of the strokes of the  $j$ -th class.

According to the similarity measure and from the analysis of the position of the stroke, the classification process returns a ranked list of classes to which the stroke could belong. Successively, each class is replaced with the corresponding group and successive occurrences of the same group in the ranked list are removed.

## 5 Recognition

The recognizer works according to a simple iterative strategy: at each step of the recognition phase, the system checks if the sequence of groups under examination is a lawful character. At this purpose, a list of sequences of groups of strokes is associated to each character and for each different sequence of groups the zones of affiliation of the strokes (lower, middle-lower, middle, middle-upper, upper) are also stored.

When a character is identified, an entry in the trie structure of the data dictionary is gained and the recognition procedure continues for another lawful character until a complete word is identified. The data dictionary contains the description of each basic word. It is organized as a trie structure in which each node is a character and each sequence of characters obtained beginning from the root of

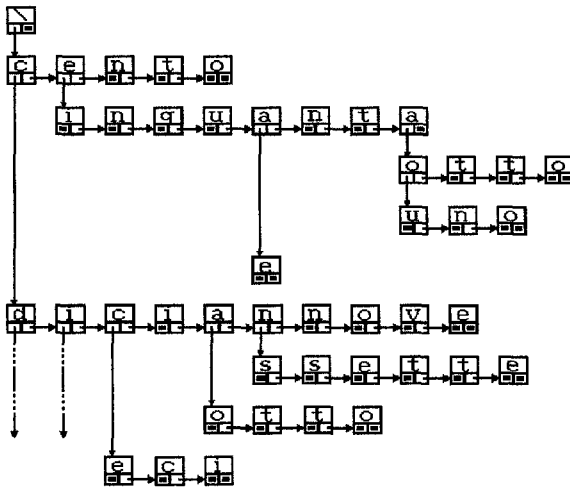


Fig. 7. The trie structure representing the data dictionary.

the trie structure to one of its terminal nodes represents a basic word. Figure 7 shows the structure of a small part of the data dictionary used in the system.

In some cases the code contains strokes useless to the aim of the recognition or it lacks of entire characters due to very poor writing quality. The recognizer recovers these conditions by cashing from the code a boundless number of

groups retained bindings or adding a finite number of character supposed to be lacking. Of course, this recovery procedure stresses the normal operation of the system and therefore this condition adds a penalty to the similarity index used for the final classification. Specifically, the recognition index used to select the candidate words is computed in the following way:

$$Ind = \frac{1}{n} \left( \sum_{j=1}^n pos_j + b\_penalty \cdot \sum_{i=1}^{nb} i + l\_penalty \cdot \sum_{k=1}^{nl} k \right)$$

where:  $n$  is the number of code groups used to compose the word,  $pos_i$  is the position of the selected group in the group list,  $nb$  is the number of bindings,  $b\_penalty$  is the penalty when a binding is found,  $nl$  is the number of characters not found in the code,  $l\_penalty$  is the penalty when a character is not found in the code.

## 6 Experimental Results

The system presented in this paper has been realized in TURBO C++ 3.00 for DOS and it actually runs on a PC - 80486DX/33Mhz. Data acquisition has been accomplished by a HP scanjet IIc with a resolution of 150 dpi.

It has been tested on the set of fifty basic words extracted from the worded amounts of italian bankchecks [7]: *uno, due, tre, quattro, cinque, sei, sette, otto, nove, dieci, undici, dodici, tredici, quattordici, quindici, sedici, diciassette, diciotto, diciannove, venti, ventuno, ventotto, trenta, trentuno, trentotto, quaranta, quarantuno, quarantotto, cinquanta, cinquantuno, cinquantotto, sessanta, sessantuno, sessantotto, settanta, settantuno, settantotto, ottanta, ottantuno, ottantotto, novanta, novantuno, novantotto, cento, mille, mila, unmilione, milioni, unmiliardo, miliardi*. The experimental results have been obtained using two suitably defined databases. The *reference database* has been used by a human operator for the definition of the basic strokes and of the sequences of strokes composing each character. It consists of 200 reference words written by four different writers. The *test database* consists of the 4050 words, written by 81 different writers.

A preliminary analysis of the accuracy of the system has been carried out with respect to the stroke classification. Column A of table 1 reports the

Position	A (in %)	B (in %)
1	79.1	87.5
2	9.3	7.7
3	4.8	2.1
4	2.3	1.2
5	1.4	0.3
more than 5	2.6	0.7
Error	0.5	0.5

Table 1.

performance in stroke classification. In 79.1% of the cases a stroke is correctly classified in the first position of the list. The probability to be in the first five positions of the list is 96.9%. Column B of table 1 reports the performance of the system in groups classification.

As column B shows, misclassification goes down when groups are considered. In this case the probability to be in the first position is 87.5%, while the probability to be in the first five positions of the list is about 98.8%.

The complete test about the performance of the system in recognizing basic words is reported in column A of table 2. The recognition in the first position has been achieved in 70.7% of the cases, in the first ten positions in 93.9% of the cases. The recognition error, due to the absence of the correct word in the list of hypotheses of the recognizer, is 5.4%. A manual segmentation procedure has been carried out on the word images in order to demonstrate how uncorrect segmentation can lead to misclassification.

Position	A (in %)	B (in %)
1	70.7	82.6
2	12.8	8.5
3	4.0	2.6
4	2.2	1.2
5	1.5	0.6
6	0.9	0.2
7	0.6	0.2
8	0.4	0.2
9	0.5	0.1
10	0.3	0.1
more than 10	0.7	0.2
Error	5.4	3.5

Table 2.

Column B of Table 2 reports the performance of the system when manual segmentation is performed.

Currently, the system performs word recognition with a speed of about 5 words/second when complete ranked lists are considered for stroke classification. Faster output can be enforced if reduced ranked lists are considered.

The system has been also tested on complete worded amounts written by

different writers also by using the syntactic rules of the italian worded amounts [7]. On a test database of 750 amounts, the system obtains a correct recognition the 60.8% of the cases when automatic segmentation is accomplished. When manual segmentation is carried out, the correct recognition reaches the 72.2% of the cases.

## 7 Conclusion

In this paper an analytical approach for handwritten word recognition is presented. The approach uses a simple scheme to identify shapes in the word image and a recursive procedure to verify each hypothesis generated at the lower level from the analysis of the basic shapes. The recognition results points out the effectiveness of the approach in domains with a reduced dictionary.

## References

1. K.Sakai, H.Asami and Y.Tanabe, "Advanced application systems for handwritten character recognition", in *From Pixels to Features III - Frontiers in Handwriting Recognition*, S. Impedovo and J.C.Simon eds., Elsevier Publ., pp.371-384, 1992.
2. R. Casey, D. Ferguson, K. Mohiuddin, E. Walach, "Intelligent Forms Processing System", *Machine Vision and Applications*, Vol. 5, N. 3, pp. 143-155, 1992.
3. L.J. Evett, C.J. Wells, F.G. Keenan, T. Rose, R.J. Whitrow, "Using linguistic information to aid handwriting recognition" in *From Pixels to Features III- Frontiers in Handwriting Recognition*, S. Impedovo, J.C. Simon, (eds.), Elsevier Publ., pp.339-348, 1992.
4. C.A. Higgins, D.M. Ford, "A new segmentation method for cursive script recognition", in *From Pixels to Features III- Frontiers in Handwriting Recognition*, S. Impedovo, J.C. Simon, (eds.), Elsevier Publ., pp. 75-86, 1992.
5. T.Pavlidis, *Algorithms for Graphics and Image Processing*, Berlin-Heidelberg, Springer-Verlag, 1982.
6. J. Hull, Tin Kam Ho, J. Favata, V. Govindaraju, S. Srihari, "Combination of Segmentation-based and Wholistic Handwritten Word Recognition Algorithms", in *From Pixels to Features III- Frontiers in Handwriting Recognition*, S. Impedovo, J.C. Simon, (eds.), Elsevier Publ., pp. 261-272, 1992.
7. G. Dimauro, M.R. Grattagliano, S. Impedovo, G. Pirlo, "A System for Bankchecks Processing", IEEE Computer Society Press, Vol. 4960-02, ICDAR '93, Tsukuba Science City -Japan, Oct. 1993, pp. 454-459.