MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

AN OCR SYSTEM FOR PRINTED DOCUMENTS

F. Lebourgeois, J.L. Henry, H. Emptoz

LIPSI Equipe de modélisation des Systèmes et Reconnaissance de Formes
Institut National des Sciences Appliquées de Lyon 69621 Villeurbanne France

ABSTRACT

This paper describes the general structure of a full automated document analysis system for printed documents. The system is based on a character preclassification stage which reduces the number of patterns to recognize and introduces a new contextual processing. This specific approach for multifont printed documents reading is based on pattern character redundancies. With the study of prototype pattern instead of characters, we can extract reliable contextual information on the entire text. This statistical information is used to check hypotheses given by the multifont recognition stage. We shall describe the system organization and particularly the preclassification stage, then we shall give some results on recent documents.

INTRODUCTION

The results which has were obtained during the last twenty years in O.C.R domain [1] underline the weaknesses of a reading system only based on character recognition. In spite of recent improvements in character recognition [2], the usual recognition rate for multifont characters cannot exceed 98 % without training. This poor performance can be improved by a training step with a test page. But this stage forces the user to type all characters from the training set; in the other hand , the recognition rate greatly depends on the document quality. Moreover a lot of recognition stages fail with broken or linked characters. In order to improve the recognition rate, many authors use a contextual processing restricted to a word spelling correction which is often placed after the recognition stage. At this place, the lexical checking cannot give complete satisfaction. Some research works use a word dictionary during the recognition step. But the general performance also depends on the dictionary size and the user vocabulary. These problems can be solved by extending the hypothesis to the entire text, with the study of every word which contains the same character pattern [3]

PROPOSITION

First, we propose to identify automatically different character patterns in the text and simultanously build character prototypes during a learning step. Secondly, we label all characters with correspondent prototype numbers in a "label sheet". This method reduces the number of characters to be recognized and authorizes the building of a set of training characters, as a training system does. These training characters can be recalled every time we need to read other documents of the same type. Moreover we introduce a new contextual processing using all the words which contain the same character pattern by analysing the "label sheet". The recognition stage only gives hypothesis on a reduced set of character prototypes which are checked by the word spelling correction stage using all the words within the text. Figure 1 resumes the main organization of the system. We can easily see the separation between the pattern identification during the training step and the characters recognition stage combined with the contextual processing. The communication between these two parts is represented by a prototype image database, the label sheet and a correspondence table. The prototype image database is used during the training and the recognition step. A cross reference table links prototypes with their bitmap images and their identification hypothesis computes by the recognition stages. For each prototype, the contextual stage checks in this table, the right hypothesis according to the maximum probability to find in the dictionary most of the words containing the character prototype. The confusion matrix allows the system to take into account confusion probabilities between prototypes. The cross reference table also stores for each prototype some typographical information (upcase letter, accentuation, presence of apostrophe in the neighbourhood) and statistical information (prototype frequencies in the text, histogram of the prototype position in the word...). Meanwhile most of this information is not yet used by the contextual stage.

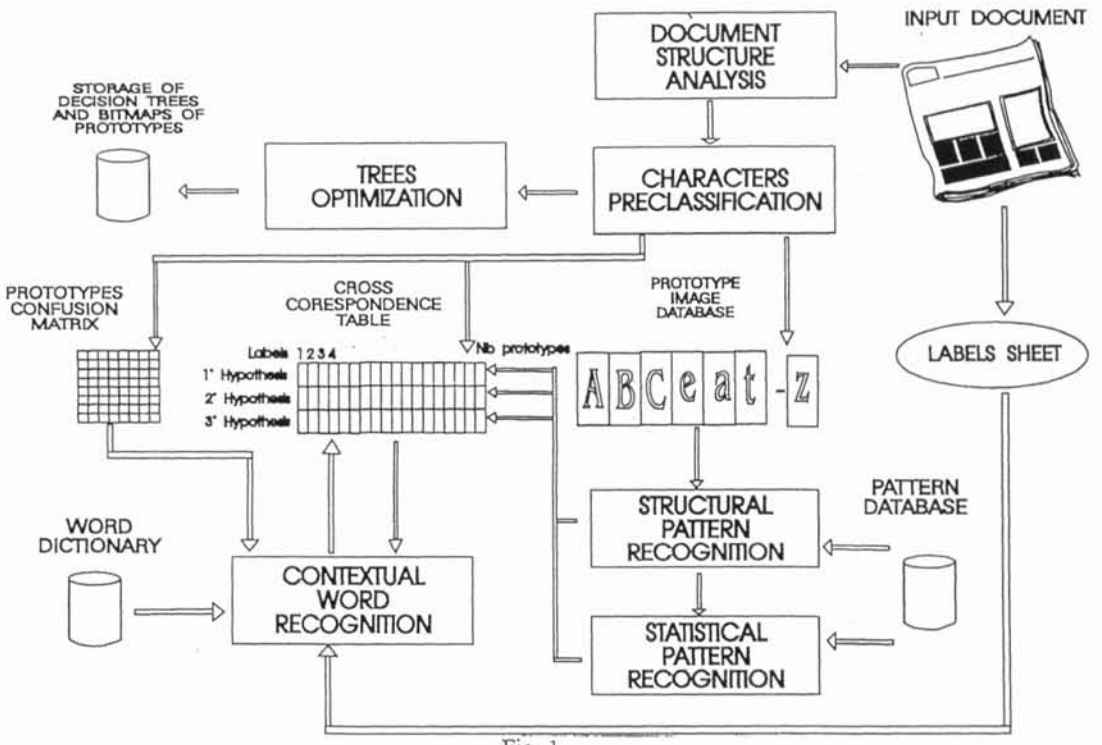

Fig. 1

SEGMENTATION STAGE

The segmentation stage uses a fast and efficient method for extracting graphics and text paragraphs with a usual bottom-up approach [4]. In order to speed up document structure analysis we propose a modified Run Length Smoothing Algorithm which merges all characters into line blocks and simultanously reduces the size of the input document. Created blocks are segmented either as text lines or graphic parts by a labelling stage. The text lines are merged into paragraphs according to typographic rules. This processing takes 8-10 seconds with a personal computer IBM AT 386/25 and 300 Dpi images. It also allows us to read string from tables or text within graphic parts. The segmentation of overlapped paragraphs by rectangular paragraphs is also possible with this algorithm.

CHARACTER PRECLASSIFICATION

This stage must automatically identify all different patterns present in the text. The preclassification stage must have a very low substitution rate (less than 0.5 %) and a high speed processing. So we use the two well known algorithms [5] : the template matching for its expensive computation saving and the matrix matching for its low error rate (Fig 2). The classifier uses binary decision trees built progressively during the text reading. In order to improve the classification performance, we build a decision tree for each character size.

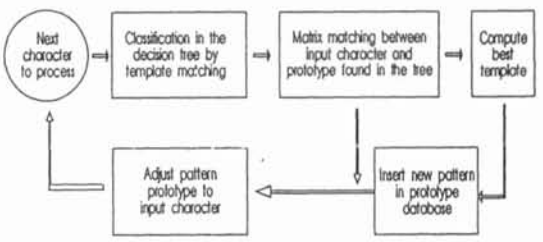

Fig. 2

Each node represents a particular test defined by a template position and the final node gives the right character prototype. The matrix matching stage verifies the correspondance between the two bitmap images.If there are too many differences, the stage looks for the best template which characterizes the main difference. This template is defined by the bigger and the most rectangular area in the XOR image (Fig 3). Considering only the size of the template, the system may build unreliable tests. The rise of the character prototypes depend on the document quality and the number of fonts used in the text. To reduce the prototype number increase, we adjust the pattern prototype to the input character with logical operations between the two bitmap images. The result is stored in the prototype image database, so that the prototype will fit better to the next input character. The character modeling becomes more efficient particularly in noisy environment. The decision threshold used by the matrix matching stage is computed from the input character size or defined by the appropriate tree.

This threshold varies between 2x2 pixels for 6 or 8 Point characters and nx8 or 8xn pixels (n<8) for 24 point characters and more. As the tree organization depends on the input character order, the trees are often unbalanced and the character prototypes are redundant. For example, with the following input character (u,n,o,e) the algorithm builds a better organized tree than the following sequence (n,e,o,u) does. In fact it is more difficult for the system to distinguish (e) from (nou) than to make a good separation between (n) and (oue) and finally between (o) and (e). (Fig 4) The problem is solved by a tree reorganization with a tree optimisation algorithm after each page reading.

Input character xor image character prototype (+) template found

Fig. 3

This simple method allows high speed identification (between 60 and 80 characters per second on a PC386/25) and a low error rate. The average rate of pattern redundancies found in a first page of any document is better than 95 % and this rate can reach 99 % for high quality document after the second page reading. A 95 % pattern redundances rate for a text of 5000 characters, means that the recognition stage has only 250 pattern prototypes to recognize with less than one misclassification error.

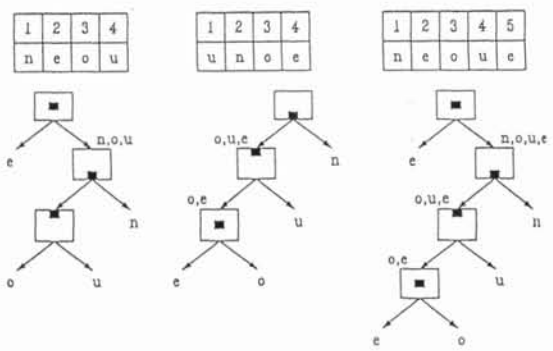

Fig. 4

CHARACTER PROTOTYPE RECOGNITION

The recognition stage combines a structural stage, using a simple topologic description and a statistical classifier based on usual geometric measures [6]. The first stage extracts structural features with a horizontal Line Adjacency Graph (Fig. 5). This very general description is almost independent from the police but the recognition process always fails when the character structure is damaged (broken or linked characters , fill hole ...). The statistical stage computes the features by histrogramming the character image projection along four directions, horizontal,vertical and the two diagonal directions. Each histogram divided into three parts gives three maximum values. For each subpart, we compute the maximum distance between the pattern and the border along the XY axis.

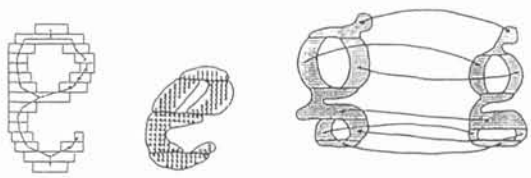

Fig. 5

The unknown pattern is represented by a vector $X=(x_1..x_{24})$. The statistical classifier writes the best answers in the cross reference table. Generally the statistical stage hardly improves the recognition results on noisy characters. The pattern library was built by an intensive learning over 60 usual fonts. The recognition rate is quite low for the first hypothesis (95 %). But with two hypotheses the recognition rate reaches 99%. These results lead us to consider the three main hypotheses given by both stages. So the contextual processing must check for each prototype in the cross reference table the right hypothesis.As we only recognize the character prototypes, the recognition speed does not matter.

CONTEXTUAL PROCESSING

The contextual processing mainly uses a lexical check with a word dictionary. The word correction is based on a Dictionary Viterbi Algorithm [7] , which uses transition and substitution probabilities between characters. The substitution probabilities have been computed from the dictionary. Moreover usual OCR confusion like (9,g)(0,o).. (l,1,t,j,i,I) has been introduced [8]. For each character prototype X we compute the recognition error probabilities $P(X=k)$ on the entire text where k depicts the character identification hypothesis found in the cross reference table. $P(X=k)$ stands for the error probability we made with the substitution of the character prototype X by the identification hypothesis k

$$P(X=k)=\frac{\text{Number of words containing the prototype } \boldsymbol{X} \text{ named } \boldsymbol{k} \text{ found in the dictionary}}{\text{Number of words containing the character's prototype } \boldsymbol{X}}$$

For each hypothesis k , we compute the maximum probability which gives the correct identification k of a prototype pattern X. If $\max_{k=1..m}[P(X=k)]<\varepsilon$ the system cannot give the right answer for three reasons

- The recognition stage has not given the good hypothesis
- There are not enough words to compute reliable probabilities
- there are not enough characters which have already been recognized

The third problem is solved by analysing first the more frequent prototypes in order to reduce the number of unknown prototypes. The program repeats the contextual analysis for each prototype until no more modification. The second fact underlines the limits of our system. Our contextual processing performance greatly depends on the document length. But if there is no pattern redundancies and if each prototype stand for only one character in the text then the contextual stage will work as usual by an analysis restricted to only one word.

CONCLUSION

In most cases, the final recognition rate exceed 99 % with usual printed documents such as newspapers,books or publications. But we cannot evaluate the results of each component of the reading system. In practice, we have noticed that a high substitution rate of the preclassification stage may damage the global results. It appears that the recognition stage is less important than the preclassification stage.We shall try some experiments with more hypotheses in the cross reference table, but we must take care of the excessive computation. This approach allows us to implement the software on a slow computer and to program more sophisticated algorithm for character pattern recognition. Meanwhile the presented system is more efficient with long text and it cannot really work with few words or sentences. So our application is better suited for massive documents analysis.

REFERENCES

[1] KAHAN, S. PALVIDIS, T.BIRD On the recognition of printed characters of any font and size IEEE Transactions on Pattern Analysis and machine Intelligence, March 1987, Vol 9,N°2 P.274-288

[2] S. BAIRD, R. FOSSEY; A 100-font classifier first International Conference on Document Analysis and Recognition, September 30-october 2,1991 P332-340

[3] F. LEBOURGEOIS,H. EMPTOZ. Organization of a reading system for multifont printed document. 8° congres de Reconnaissance des Formes et Intelligence Artificielle,25-29 Nov 1991.Vol 2, P713-718.

[4] F. LEBOURGEOIS, Z. BUBLINSKY, H.EMPTOZ, A Fast and efficient method for extracting text paragraphs and graphics from unconstrained Documents,11 th IAPR International Conference on Pattern Recognition, August 30-September 3,1992 P272-276

[5] BRICKMAN, ROSENBAUM, Word aotucorrelation redundancy match (WARM) Technology, nov 1982,IBM. J. Res. Dev.,vol 26, N° 6, P 681-686.

[6] P. CAMPIGLI,V. CAPPELLINI A reading system for printed documents , first International Conference on Document Analysis and Recognition, September 30-october 2,1991 P585-593

[7] R. SINHA , On partitioning a dictionary for visual text recognition,Pattern Recognition vol 23 N°5,1990,P 497-500.

[8] SINHA,BIRENDRA,PRASADA, Visual text recognition through contextual processing.Pattern Recognition,1988,Vol 21,N°5, P463-479.