Pergamon

PII:S0952-1976(97)00013-4

Engng Applic. Artif. Intell. Vol. 10, No. 4, pp. 403–412, 1997

0952-1976/97 $17.00+0.00

Contributed Paper

A Binary-Tree-Based OCR Technique for Machine-Printed Characters

B. GATOS
Democritus University of Thrace, Xanthi, Greece

N. PAPAMARKOS
Democritus University of Thrace, Xanthi, Greece

C. CHAMZAS
Democritus University of Thrace, Xanthi, Greece

(Received May 1995; in revised form February 1996)

This paper describes the structure of an optical character recognition (OCR) system for printed documents. This system is trained for Latin and Greek typewritten text, but it can be easily adapted to any typewritten character set. The proposed method is divided into two main stages. In the first stage suitable binary features are extracted, most of which are independent of the scaling and rotation of the characters. After that, a binary tree classification technique is used, and an optimal tree classifier is constructed. In the second stage, the characters at the end-nodes of the binary tree are classified by using a new template-matching technique. By setting a suitable threshold for the matching, a decision can be reached for the greatest part of the characters. For those characters that the binary tree cannot recognize with great confidence, a secondary minimum distance, classifier trained with the Zernike moments of the characters, is used. Experimental results show that the performance of the proposed OCR system is high, and the recognition rate can exceed 99.5%.

Keywords: Optical character recognition, feature extraction, matching, classification, binary features.

1. INTRODUCTION

Optical Character Recognition (OCR) is very important in office automation. By definition, OCR systems are used to translate human-readable characters to machine-readable codes. Up to now, several OCR techniques have been proposed, based on statistical, matching, transform and shape features (Stentiford, 1985; Lettera *et al.*, 1986; Kahan *et al.*, 1987; Persoon and Fu, 1986; Abdelazim and Hashish, 1989; Cash and Hatamian, 1987; Papamarkos *et al.*, 1994). Recently, integrated OCR systems have been proposed that take advantage of specific character-driven hardware implementations (Pereira and Bourbakis, 1993, 1995). An excellent survey of OCR is given by Impedovo *et al.* (1991), while IEEE (1992) describes the state of the art on OCR. Typically, an OCR system consists of the pre-processing, feature extraction and the classification stages. Therefore, in an OCR system, the selection of a set of sufficient and powerful features is very important. On the other hand, it is generally accepted that feature extraction is one of the most difficult and crucial problems in pattern recognition (Impedovo *et al.*, 1991). It is well known that a set of "good" features generally embodies some important characteristics such as:

(a) *Discrimination.* Features should take on significantly different values for characters that belong to different classes.
(b) *Reliability.* Features should take on similar values for characters that belong to the same class.
(c) *Independence.* Features should be uncorrelated with each other.

Correspondence should be sent to: Associate Prof. Nikos Papamarkos, Electric Circuits Analysis Laboratory, Department of Electrical & Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece [E-mail: papamark@voreas.ee.duth.gr].

(d) *Small feature space*. The number of features used should be small enough to make classification simple and fast.

Moreover, it is desirable that the features should satisfy some additional requirements, such as low computational cost and low complexity of the feature-extraction techniques. For these reasons, simple and powerful features cannot easily be found.

The common features that are used in OCR are continuous, i.e. features that result in continuous real values, and are therefore sensitive to noise and have a low discrimination ability. On the other hand, binary features possess some desirable characteristics, such as

- *High discrimination ability*. For example, the answer to the question as to whether a character has a hole in it or not is crucial to the effective classification of characters into classes.
- *Determination accuracy*. For example, there is no need to determine the exact positions of the end points. This property implies that the binary features have low sensitivity to noise.

The objective of this paper is to propose an OCR method, based on binary features, which combines a binary tree classifier and a template-matching technique. In view of the above-mentioned benefits, this OCR system has been developed by using thirty-four binary features, most of which are independent of scaling and rotation. These features correspond to:

- holes;
- end points, types of end points and tree points (junction points);
- black and white horizontal and vertical transitions;
- measurements of the perimeter;
- relative positions of the characters in the text line.

In order to use a binary tree classifier, the above features are transformed into binary form. For example, the number of holes in a character is described by two features. Then if a character has only one hole, it gives the following feature values:

- feature(5)=0, and
- feature(6)=1.

The classification technique employed here is based on the minimum entropy criterion, in combination with a binary decision-tree classifier. The binary decision tree is generated automatically using the entropy reduction technique (Casey and Nagy, 1984; Shlien, 1988). According to this approach, the structure of the binary tree is determined such that, at each node of the tree, the maximum reduction of the total entropy is obtained. The main advantage of this technique is that it is not necessary to assume the statistical independence of the binary features. Also, characters with multimodal distributions such as (a,o) are processed correctly in an automatic manner. Therefore, it can be claimed that the binary tree classifier defines the best path (the sequence of the necessary features) of the tree which should be followed for fast and accurate recognition of each one of the characters.

Due to the characters' shape similarities, it is possible that different characters, belonging to different classes, will result in the same final tree node. This means that those characters will remain unclassified, and therefore an additional procedure for the final classification is necessary. In order to overcome this problem, a new template-matching technique is employed at every final tree node. This technique is based on the calculation of the matching distance between the input unclassified characters and the known prototypes.

Due to the binary tree, some characters may end up at the wrong final tree nodes, at a great distance from all the trained prototypes of the nodes. Therefore, correct decisions may not be obtained for those characters. In order to overcome this problem, and to increase the entire recognition rate, only the unrecognized characters are sent to a subsidiary classifier. By using the minimum distance classifier trained with characters' Zernike moments, high recognition rates can be achieved at high speeds, due to the fact that the greater part of the characters are processed and classified by the binary tree, which is very fast.

Figure 1 illustrates the proposed OCR system. It can be divided in two main stages, the training and the recognition stage. At the training stage, the OCR system can easily be trained to be able to recognize new character sets. The result of the training stage is a new binary tree classifier, a new set of integers for the matching procedure and a new set of Zernike moments for the subsidiary minimum distance classifier.

The recognition stage has the following steps:

- Image pre-processing.
- Feature extraction (extraction of the necessary features according to the optimal path that arise from the binary tree classifier).
- Minimum matching distance classification.
- Minimum distance classification using Zernike moments, when this is necessary.

Experimental results given at the end of the paper show that the recognition rate achieved by the proposed OCR method is satisfactory.

2. PRE-PROCESSING

To apply the OCR procedure, a pre-processing scheme is necessary. The pre-processing operations include the following main procedures:

- An optimal threshold-selection method to obtain the binary form of the digitized document. This method is based on a histogram valley approximation by a rational function and the one-dimensional Golden minimization algorithm (Papamarkos *et al.*, 1988; Papamarkos and Gatos, 1994; Press *et al.*, 1986). According to this method the positions of the peaks of the histogram's hills are first approximately determined. Next, the valley (that is, the region between two peaks) is approximated by a real rational function according to the minimax criterion. Afterwards, in order to find the optimal threshold values for each sub-region, the Golden search minimization algorithm is applied. This technique is simple, and gives satisfactory binary images.

- Rotation of the document to remove some orientation and skewing variations. To do this, a method based on the cross-correlation between the pixels of vertical lines is used (Gatos *et al.*, 1995). Therefore, after this stage all the symbols have approximately the same horizontal orientation.
- Segmentation to separate the characters from the background. Assuming that the layouts of the documents can be partitioned into isolated blocks and characters, the segmentation technique proposed by E. Wahl *et al.* (1982)

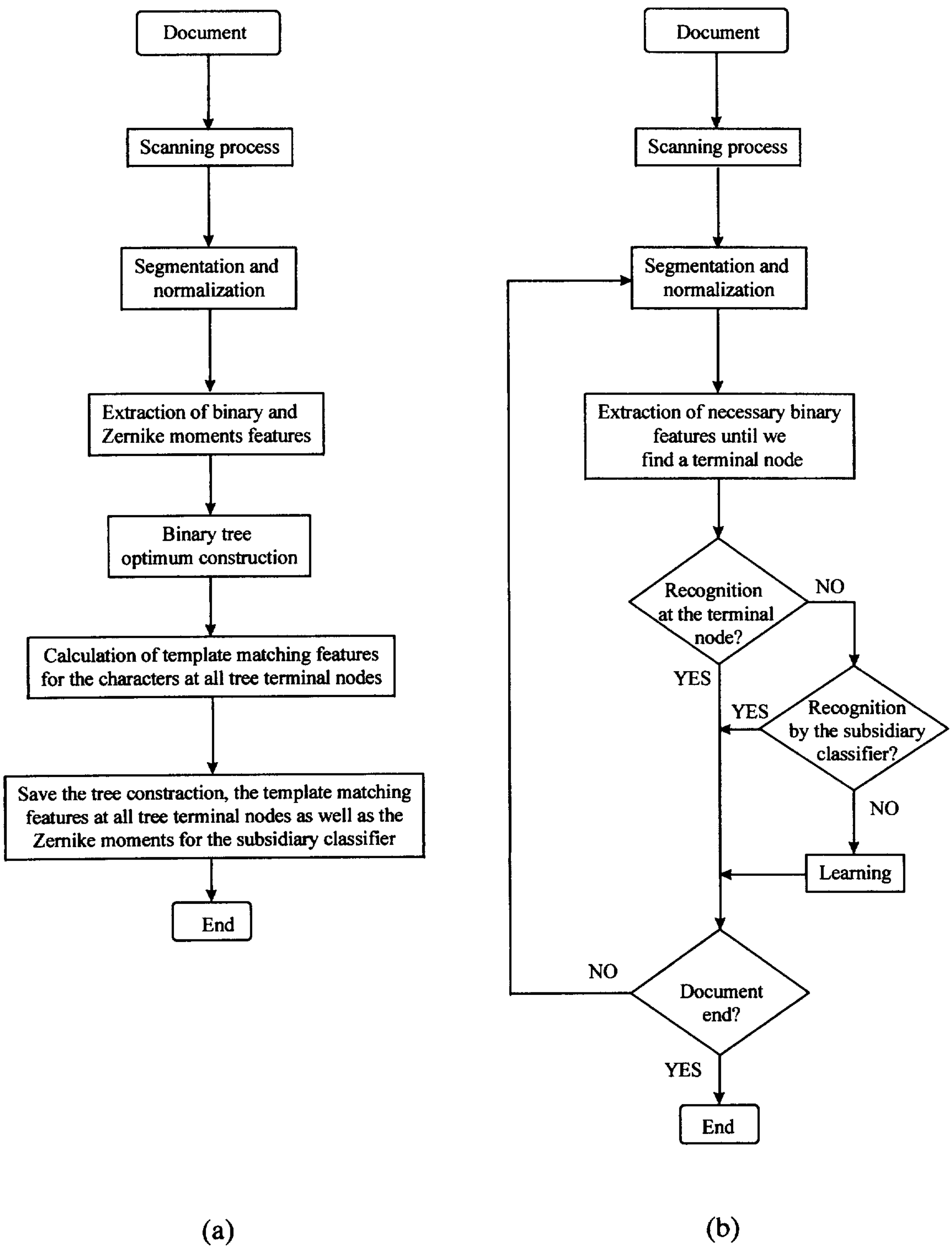

Fig. 1. Binary tree classifier: learning (a) and recognition (b).

is applied, in combination with techniques that permit the separation of Italic and touching characters.

- Normalization. A normalization technique is used to scale each character, and to fit it into a 16×16-pixel window so that the character touches the left-hand side and the bottom of its window. Obviously, the value of the scaling coefficient depends on the size of each character. This technique does not change the relative positions of the character's pixels, even if there are some noisy pixels.

3. FEATURE EXTRACTION

It is very difficult to define the ideal features for an OCR system. It is well known that a set of "good" features must have some important characteristics. The features used here correspond to the characteristics described in the introduction. Specifically, the following set of binary features is used.

3.1. Black and white transitions horizontally and vertically

As can be seen from Fig. 2, the middle horizontal and the middle vertical axes have been defined in the 16×16 normalized character window, as transition directions. Considering that the background pixels are white and the character pixels are black, white pixels are set in the boundaries of the normalized window. For each transition direction, the total number of transitions is calculated, from white to black and from black to white. If a character has N horizontal or vertical transitions, then it has $N/2$ contacts with the middle horizontal or middle vertical axes respectively. This is a crossing technique for feature extraction, and is concerned with topological characteristics (Impedovo *et al.*, 1991; Shridhar and Badreldin, 1984). According to the above, the extracted binary features are:

$$\text{feature}(1)=\begin{cases}1, & \text{if } x\_\text{contacts}<2\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

$$\text{feature}(2)=\begin{cases}1, & \text{if } x\_\text{contacts}=2\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

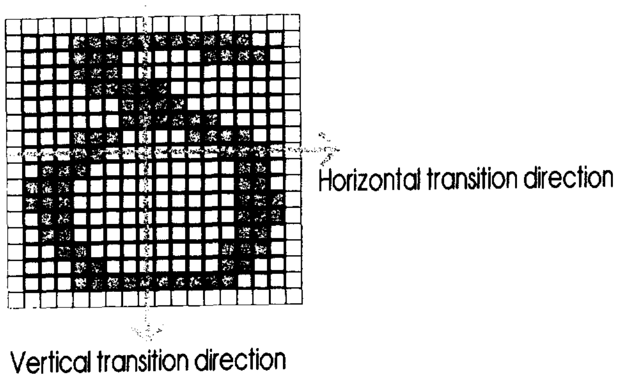

Fig. 2. The two transition directions.

$$\text{feature}(3)=\begin{cases}1, & \text{if } y\_\text{contacts}<2\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

$$\text{feature}(4)=\begin{cases}1, & \text{if } y\_\text{contacts}=2\\ 0, & \text{otherwise.}\end{cases} \tag{4}$$

For example, for the character "δ", it can be seen from Fig. 2 that it has four transitions in the x-direction and six transitions in the y-direction. Therefore, the character "δ" gives

feature(1) = 0, feature(2) = 1, feature(3) = 0 and feature(4) = 0.

3.2. Holes

By using the method described in (Pavlidis, 1992), the total number of holes is determined for each of the characters. The features defined by the holes correspond to the holes = 0 and holes = 1 cases. Specifically

$$\text{feature}(5)=\begin{cases}1, & \text{if the number of holes}=0\\ 0, & \text{otherwise}\end{cases} \tag{5}$$

$$\text{feature}(6)=\begin{cases}1, & \text{if the number of holes}=1\\ 0, & \text{otherwise}\end{cases} \tag{6}$$

For example, the character "δ" in Fig. 2 has one hole, and hence

feature(5) = 0 and feature(6) = 1.

3.3. End-points and tree-points

This set of features is commonly used in OCR. In order to find the end-points, first the skeleton of a character is calculated using the method described in (Gonzalez and Wintz, 1987). Next, the end-points are defined as those that correspond to the pixels that satisfy the following relation:

$$N[P_1]=1 \tag{7}$$

where $N[P_1]$ is the total number of pixels with value 1 neighbouring the central pixel in the 3×3 mask shown in Fig. 3.

At the same time, the tree-points (junction points) of the skeleton are determined. The tree-points are defined as the points that satisfy the relation

$$N[P_1] \geq 3. \tag{8}$$

In Fig. 4 it can be seen that the application of the above procedure to the character "δ" gives one end-point and one tree-point.

P3	P2	P9
P4	P1	P8
P5	P6	P7

Fig. 3. 3×3 mask.

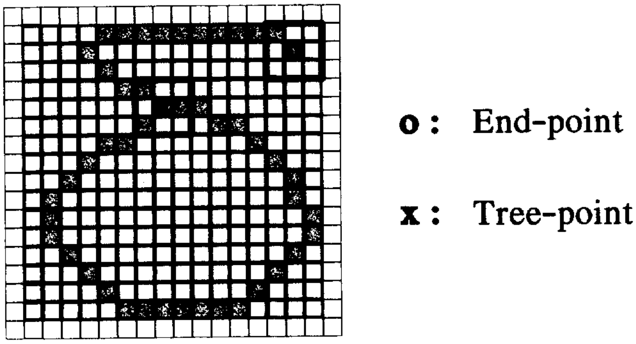

Fig. 4. Tree- and end-points for the character δ.

After having defined the end-points and the tree-points, binary features are extracted, considering the number of those points as well as their nearest neighbours from the 6 predominant positions (E_i^x,E_i^y), (T_i^x,T_i^y) $i=1\ldots6$, of the end-points and tree-points, respectively. Those positions are derived from a statistical technique that uses all the end-points and tree-points of the training set (see the Appendix). These features are defined by the following relations:

$$\text{feature}(7+n)=\begin{cases}1, & \text{if the number of end}-\text{points}=n\\ 0, & \text{otherwise}\end{cases}\tag{9}$$

for $n=0,1,2,3$

$$\text{feature}(10+n)=\begin{cases}1, & \text{if the number of tree}-\text{points}=n\\ 0, & \text{otherwise}\end{cases}\tag{10}$$

for $n=1,2,3$

$$\text{feature}(13+n)=\begin{cases}1, & \begin{array}{l}\text{if at least one end point is determined}\\ \text{to be nearest to }(E_n^x,E_n^y)\end{array}\\ 0, & \text{otherwise}\end{cases}\tag{11}$$

for $n=1,2,3,4,5,6$

$$\text{feature}(19+n)=\begin{cases}1, & \begin{array}{l}\text{if at least one tree point is determined}\\ \text{to be nearest to }(T_n^x,T_n^y)\end{array}\\ 0, & \text{otherwise}\end{cases}\tag{12}$$

for $n=1,2,3,4,5,6$.

3.4. Perimeter features

For each character, the external perimeter is calculated, and according to its length six sets of binary features are defined. The statistical method given in the Appendix is used to find the six predominant positions, P_i, $i=1\ldots6$, of the perimeter values. According to the perimeter value distance value from P_i, the following six binary features are defined:

$$\text{feature}(25+n)=\begin{cases}1, & \begin{array}{l}\text{if perimeter is determined}\\ \text{to be nearest } P_n\end{array}\\ 0, & \text{otherwise}\end{cases}\tag{13}$$

for $n=1,2,3,4,5,6$.

3.5. Relative position features

As can be seen in Fig. 5, by dividing the text line into three zones, one can search for a character's existence in each of them (Pavlidis, 1992) The binary features extracted in this case are

$$\text{feature}(31+n)=\begin{cases}1, & \begin{array}{l}\text{if a part of character}\\ \text{exists in zone } n\end{array}\\ 0, & \text{otherwise}\end{cases}\tag{14}$$

for $n=1,2,3$

Figure 6 shows the binary features of a character "α".

4. OPTIMAL CONSTRUCTION OF THE BINARY TREE

After the extraction of a set of binary features, the binary tree classifier must be built. For the optimal construction of the binary tree a criterion is needed in order to select, at every node, the feature with the maximum discrimination ability, plus a criterion in order to decide whether an end node has been reached, or if there is a need to extend it further (Shlien, 1988). The feature selected at each node is the one that maximizes the information for the character discrimination. The information obtained is defined by the reduction of the Shannon entropy, as calculated directly from the distributions of the various class patterns reaching that certain node.

If n_k is the number of patterns of the k class of the training characters in a certain node, then the estimated entropy is given by

$$E=-\sum_{k=1}^{K}\frac{n_k}{N}\log_2\left(\frac{n_k}{N}\right)\tag{15}$$

where $N=n_1+n_2+\ldots+n_k$ and K the number of classes.

If E_1 and E_2 are the calculated entropies of the two pattern groups M and N–M which are created based on a certain feature, then the information gained from this feature is:

$$I=E-\frac{ME_1}{N}-\frac{(N-M)E_2}{N}.\tag{16}$$

To find the optimum feature in a node the frequency distribution function $f(j,k)$ of the training patterns is calculated for the J possible features and for the K possible classes.

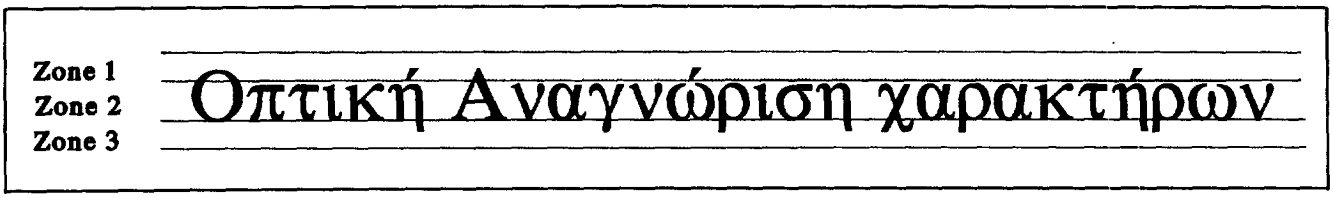

Fig. 5. The three zones.

If, at a certain node, the feature with the maximum discrimination ability divides the training patterns into class distributions m_k, $n_k - m_k$, then the node may be defined as a terminal node when:

$$P = \frac{{}^{n_1}C_{m_1}\,{}^{n_2}C_{M_2}\cdots{}^{n_k}C_{m_k}}{N_{C_M}} > 0.001 \tag{17}$$

where $N = n_1 + n_2 + \ldots + n_k$, $M = m_1 + m_2 + \ldots + m_k$ and ${}^{n}C_{m} = \frac{n!}{m!(m-n)!}$. P is defined as the probability of observing a distribution m_k from sampling the parent population with distribution n_k (Shlien, 1988).

Alternatively, an end-node can be defined when no feature possesses any discrimination ability at this node.

5. TEMPLATE MATCHING

In a terminal node of the binary tree, it is possible to have more than one class of characters. To classify a character that has ended up at this node into one of the node classes, a template-matching classifier is used. For the template matching, the information existing in eight line directions (horizontal, vertical and diagonal) of the character matrix, is selected from the characters, as shown in Fig. 7. By doing this, eight binary numbers are extracted for every character. For example, in Fig. 7 and for the first feature-extraction direction, the binary number: 0001100000010010 is extracted, which also corresponds to the integer 6152.

Define, in each terminal node, and for every class:

- n the number of pattern prototypes,
- λ_i^k the λ_i integer of the k prototype pattern, and
- μ_i the μ_i integer of the input unrecognized character whose distance from a certain class must be calculated.

Then the distance of the character from the particular class is:

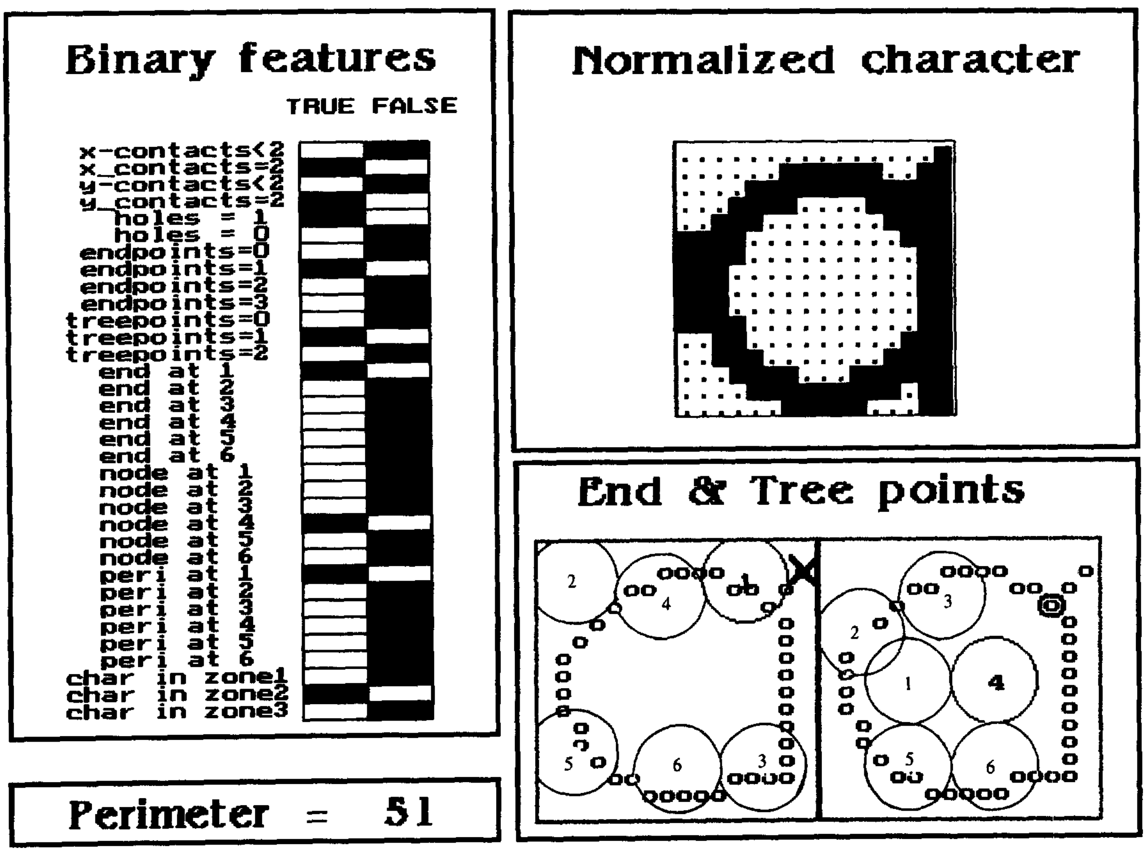

Fig. 6. Binary feature extraction.

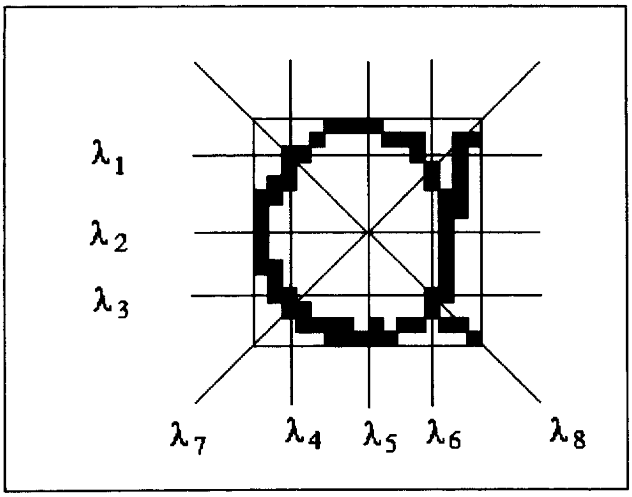

Fig. 7. The eight feature-extraction directions.

$$D=\frac{1}{n}\sum_{i=1}^{8}\sum_{j=1}^{16}\left|nM_{ij}-\sum_{k=1}^{n}L_{ij}^{k}\right| \qquad (18)$$

where M_{ij} is the jth bit of the integer μ_i and L_{ij}^k is the jth bit of the integer λ_i^k.

In order to improve the performance of the directional template-matching technique, the integers μ_i and λ_i^k are transformed, giving some weight to the 0 elements of their binary presentation according to their distance from the 1 elements, where the 1 elements get a bigger value. This is done to achieve a distortion independence of the template-matching technique. Specifically, if $j \leq q$ then the value of each element of μ_i and λ_{ij}^k is equal to $q+1-j$, where j is the distance from the nearest 1, and q a suitable auxiliary variable (usually q is taken equal to 2). For example, the integer

$$\underbrace{0\,0\,0\;.\;.\;.\;0}_{q}\;1\;\underbrace{0\;.\;.\;.\;0\,0\,0}_{q}$$

is generally transformed to the number sequence:

$$\mathbf{1\,2\,3 \ldots q(q+1)q \ldots 3\,2\,1.}$$

Additionally, with $q=2$, the integer

$$\mathbf{0100110100100010}$$

is transformed to the number sequence:

$$\mathbf{2322332322321232.}$$

Using the above transformation, the μ_i values are transformed into M_{ij} and the λ_i^k into L_{ij}^k. After these transformations, M_{ij} and L_{ij}^k have values from $[0 \ldots q+1]$, and the distance D is given from equation (18).

It is obvious that for the implementation of equation (18) it is not necessary to store all prototype features for every class at an end-node, but just the sum and the number of them. To obtain better matching results, at the expense of computational cost and storage requirements, one could choose that the matching procedure will take place, not with the average values, but with the values of every trained class pattern. So, equation (18) becomes:

$$D=\min\sum_{i=1}^{8}\sum_{j=1}^{16}|M_{ij}-L_{ij}^{k}|,\ k=1 \ldots n \qquad (19)$$

and D is now the distance of the input character from a prototype.

The input character is recognized, if its minimum Euclidean distance is less than a suitable threshold value. Otherwise, the character is directed to a subsidiary minimum-distance classifier, trained with Zernike moments.

Table 1. Values of patterns A1, A2 and B

A1	$\lambda_1^1=9224=0001110000111000$
A2	$\lambda_1^2=14\,384=0011100000110000$
B	$\mu_1=12\,312=0011000000011000$

As an example of the matching procedure, assume that a particular character class A consists of two prototypes, A1 and A2, shown in Table 1. For this example, it is assumed that only one feature-extraction direction is available, not eight. Consider now the unclassified character B, for which it is necessary to calculate its distance D from the above class A.

Obviously, to calculate D, the values of λ_1^1, λ_1^2 *and* μ_1 are needed. Using equation (18) the value of D is calculated as follows:

$$\lambda_1^1 \rightarrow 0\,0\,0\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,0\,0\,0$$

$$\lambda_1^2 \rightarrow 0\,0\,1\,1\,1\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0$$

$$\mu_1 \rightarrow 0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0$$

$$\mathbf{D}=0+0+1+0+2+1+0+0+0+0+2+0+1+0+0+0=7$$

Otherwise, using the feature transformation to include distortion independence, with $q=3$, λ_1^1, λ_1^2 and μ_1 to L_{1j}^1, L_{1j}^2 and M_{1j} must first be transformed. The final distance D is calculated as follows:

$$\lambda_1^1 \rightarrow 0\,1\,2\,3\,3\,3\,2\,1\,1\,2\,3\,3\,3\,2\,1\,0$$

$$\lambda_1^2 \rightarrow 1\,2\,3\,3\,3\,2\,1\,0\,1\,2\,3\,3\,2\,1\,0\,0$$

$$\mu_1 \rightarrow 1\,2\,3\,3\,2\,1\,0\,0\,0\,1\,2\,3\,3\,2\,1\,0$$

$$\mathbf{D}=1+1+1+0+2+3+3+1+2+2+2+0+1+1+1+0=21$$

6. SUBSIDIARY CLASSIFIER

According to the above analysis, it is possible that at some final tree nodes there may be characters belonging to a class other than the classes determined in the training stage of the binary tree classifier. In these cases the tree classifier may give a wrong result, but with a great minimum distance value D. It is clear that adjusting the threshold value will give different recognition rates. From Table 2 it can be seen how the recognition rate increases as

Table 2. Confidence threshold and recognition rate

Confidence threshold	Characters processed	Errors	Recognition rate
20	4850 of 5760	3	99.93%
25	5219 of 5760	9	99.82%
30	5361 of 5760	14	99.73%
40	5477 of 5760	32	99.41%
50	5531 of 5760	55	99.00%
60	5568 of 5760	82	98.52%

the confidence threshold of the minimum distance classifier decreases, with a simultaneous increase of the non-classified character set.

In order to increase the recognition rate of the entire OCR system, an additional classifier based on Zernike moments (Khotanzad and Hong, 1990) is used as a final recognition process. Specifically, the non-classified characters are directed to a subsidiary minimum-distance classifier which has as features the Zernike moments of the characters. These moments have some important characteristics, such as their rotation invariance. The system described here uses as features, 49 Zernike moments of order 12.

If $C(x,y)\in(0,1)$ for $x,y\in[-1,1]$ is the image of every character, then the Zernike moments of order n with repetition m are calculated by the equation:

$$Z_{nm}=\frac{n+1}{\pi}\sum_x\sum_y C(x,y)V^*_{nm}(p,\theta),\ x^2+y^2\leq 1 \qquad (20)$$

where

- $V_{nm}(p,\theta)=R_{nm}(p)\exp(jm\theta)$,
- p: the distance of point (x,y) from the image centre,
- θ: the angle of p with the x-axis in a clockwise direction.

7. EXPERIMENTAL RESULTS AND CONCLUSIONS

There is currently no OCR system that can read all documents satisfactorily (Bokser, 1992). The performance is satisfactory only under some conditions related to the size, type and number of the tested characters, and the quality of the documents. It is well known that to achieve an impressive performance, in commercial OCR products, millions of characters are used in the training sets. Additionally, the training sets that are used are taken from a variety of fonts, they vary in image quality, and they are suitable for the variation in the documents that the OCR system is expected to handle. Difficulties that are associated with insufficient features can be overcome by using a large number of prototypes, in combination with a suitable classifier and a syntactic pattern-recognition technique. Apart from the above difficulties, for a good performance, an OCR system must give sufficient solutions to a variety of many other problems associated with the pre-processing phase, such as the scanning quality and the segmentation. Most of the segmentation problems relate to touching and overlapping characters, and to the presence of images in the document. Non-conventional OCR systems, such as "anagnostis" (Pereira and Bourbakis, 1993, 1995) are character-driven systems, designed for a specific hardware implementation. These systems have learning capabilities, and can reach high recognition rates.

The OCR system described in this paper is not a commercial product, but it proposes some interesting ideas such as the use of only geometrical binary features in accordance with the binary tree classifier, in combination with the matching method and the Zernike-moment-trained classifier. Also, the OCR system can be easily extended to handle more binary features, and can be trained with a large number of desirable fonts. The OCR system was implemented in the C programming language and for IBM 80386 and IBM 80486 computers. The documents are scanned at a 300 dpi resolution by using an HP scanner. The scanned documents are taken from 300 dpi laser printers, as well as from matrix printers, and the sizes of the characters are limited to $8\sim20$ points. Here, some experimental results, derived from the proposed OCR system are presented. As a measurement of the performance, a training set of 1152 characters, belonging to the Arc, Arial and Courier fonts and printed on an HP laser printer, was used. The training set consisted of Greek and Latin characters and their size was 12 points. The test sets used were:

AAC set:	A large sample of characters of the same fonts as the training set (5760 characters).
FT set:	Characters of the same fonts as the training set, but the documents were photocopied before scanning (7680 characters).
PIN set:	Characters of the same fonts as the training set, but printed by a dot-matrix printer (Hundai Pinovia printer—3840 characters).
CW set:	Characters of different fonts from the training set (ChiWriter font—380 characters).

Table 3 summarizes the performances of the binary tree classifier, the minimum-distance Zernike-moment-trained classifier, and the combinational classifier. From this table one can observe that:

- —The collaboration of the two classifiers helps to achieve in all cases high recognition results.
- —The computation time is always considerably less when using both classifiers, compared with the time consumed using only the Zernike moment classifier.
- —The performance of the system tested on the same font is high (99.51% at AAC set).
- —The recognition rate still remains high for a noisy set (99.05% for the FT set) or a distorted set (95.52% for the PIN set and 90% for the CW set).

Table 3. Experimental results

Test set	Zernike moments	Binary Tree	Combinational classifier		
			Binary tree	Zernike moments	Total
AAC 5760 chars	RR=99.32% Errors=39 Time=3450 sec	RR=95.79% Errors=242 Time=116 sec	RR=99.82% Chars=5219 Errors=9 Time=104 sec	RR=96.48% Chars=541 Errors=19 Time=324 sec	RR=99.51% Chars=5760 Errors=28 Time=428 sec
FTC 7680 chars	RR=98.67% Errors=102 Time=4600 sec	RR=95.66% Errors=333 Time=153 sec	RR=99.90% Chars=5440 Errors=5 Time=108 sec	RR=96.96% Chars=2240 Errors=68 Time=1344 sec	RR=99.05% Chars=7680 Errors=73 Time=1452 sec
PIN 3840 chars	RR=90.54% Errors=363 Time=2300 sec	RR=99.32% Errors=39 Time=3450 sec	RR=98.82% Chars=3056 Errors=36 Time=61 sec	RR=82.65% Chars=784 Errors=136 Time=470 sec	RR=95.52% Chars=3840 Errors=172 Time=531 sec
CW 3840 chars	RR=81.35% Errors=716 Time=2300 sec	RR=82.44% Errors=674 Time=65 sec	RR=95.91% Chars=3252 Errors=133 Time=65 sec	RR=57.31% Chars=588 Errors=251 Time=352 sec	RR=90.00% Chars=3840 Errors=384 Time=417 sec

It is noted that, apart from the pre-processing difficulties, the main reason for unclassified characters or characters leading to wrong classes, is the significant morphological similarities between some Greek characters, such as (φ,ψ), (α,o,σ), (η,υ) that are not properly classified, because of their similar shapes. For this reason, in the near future, the authors will consider enriching the OCR system described here, with a syntactic pattern-recognition algorithm.

REFERENCES

Abdelazim, H. and Hashish, M. A. (1989) Automatic reading of bilingual typewritten text. *Proceeding of VLSI and Microelectronic Applications in Intelligent Peripherals and their Application Networks*, pp. 140–144.

Bokser, M. (1992) Omnidocument technologies: invited paper. *Proceedings of the IEEE*, **80**(7), 1066–1078.

Casey, R. and Nagy, G. (1984) Decision tree design using probabilistic model. *IEEE Transactions on Information Theory*, **30**, 93–99.

Cash, G. and Hatamian, M. (1987) Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, **39**, 291–310.

De Luca, P. and Gisotti, A. (1991) Printed character preclassification based on word structure. *Pattern Recognition*, **24**, 609–615.

Gatos, B., Papamarkos, N. and Chamzas, C.(1995) Skew detection of digitized documents using cross-correlation. Presented at *5th International Conf. on Advances in Communications & Control—Telecommunications/Signal Processing in the Multimedia Era, COMCON5*, Rethymnon, Greece.

Gonzalez, C. and Wintz, P. (1987) *Digital Image Processing*, 2nd ed. Addison-Wesley, Reading MA.

Proceeding of the IEEE (1992) Special Issue on Optical Character Recognition, July.

Impedovo, S., Ottaviano, L. and Occhinegro, S. (1991) Optical character recognition—A survey. *International Journal of Pattern Recognition and Artificial Intelligence*, **5**, 1–23.

Kahan, S., Pavlidis, T. and Baird, H. S. (1987) On the recognition of printed characters of any font and size. *IEEE Trans. Pattern Anal. Mach. Intell.*, **9**, 274–287.

Khotanzad, A. and Hong, Y. H. (1990) Invariant image recognition by Zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, **12**(5), 489–497.

Lettera, C., Maier, M., Maser, L. and Paoli, C. (1986) Character recognition in office automation. In *Advances in Image Processing and Pattern recognition*, eds V. Cappellini and R. Marconi, pp. 191–197. Elsevier, Amsterdam.

Papamarkos, N. and Gatos, B. (1994) A new approach for multilevel threshold selection. *Graphical Models and Image Processing*, **56**(5), 357–370.

Papamarkos, N., Vachtsevanos, G. and Mertzios, B. (1988) On the optimum approximation of real rational functions via linear programming. *Applied Mathematics and Computation*, **26**, 267–287.

Papamarkos, N., Spiliotis, I. and Zoumadakis, A. (1994) Character recognition by signature approximation. *International Journal of Pattern Recognition and Artificial Intelligence*, **8**(5), 1171–1187.

Pavlidis, T. (1992) *Algorithm for Graphics and Image Processing*. Computer Science Press, Rockville, MD.

Pereira, N. and Bourbakis, N. (1993) Recognition of handwritten characters using a character reduction methodology. *Proceedings of "WFHR"*, New York, 25–27 May.

Pereira, N. and Bourbakis, N. (1995) Design of a character driven text reading system. *Proceedings of "SPIE"*, San Jose, California, 6–9 February.

Persoon, E. and Fu, K. S. (1986) Shape discrimination using Fourier descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**, 388–397.

Press *et al.* (1986) *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press.

Shlien, A. (1988) Multifont character recognition for typeset documents. *International Journal of Pattern Recognition and Artificial Intelligence*, **2**, 603–620.

Shridhar, M. and Badreldin, A. (1984) High accuracy character recognition algorithm using Fourier and topological descriptors. *Pattern Recognition*, **17**(5), 515–524.

Stentiford, F. (1985) Automatic features design for optical character recognition using an evolutionary search. *IEEE Trans. Pattern Anal. Mach. Intell.*, **7**, 349–355.

Wahl, E., Wong, K. and Casey, R. (1982) Block segmentation and text extraction in mixed text/image documents. *Computer Vision Graphics and Image Processing*, **20**, 375–390.

APPENDIX A

To define the predominant position for the end-points, tree-points and perimeters the following statistical technique was used:

Say that (x_i,y_i), $i=1\ldots n$ are the co-ordinates of all n end-points of all trained patterns in a normalized window. Form an accumulator array C of size 16×16 constructed according to all end-points. Specifically, for each end-point, define a square region of size $(2\lambda+1)\times(2\lambda+1)$ centered at the end-point; then the accumulator array is modified as follows:

for $i=1$ to n

for $k=0$ to λ

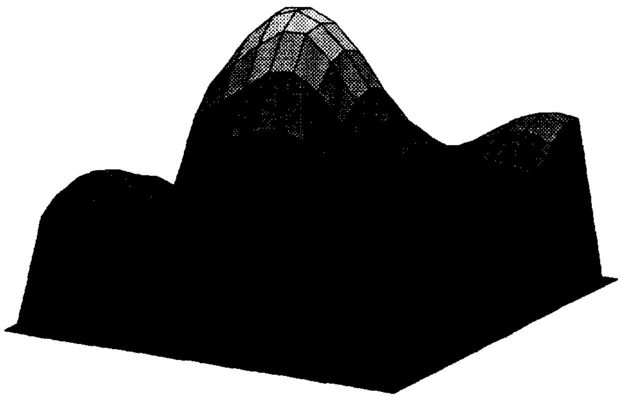

Fig. 8. Accumulator array of end-points.

for $x=x_i-k$ to x_i+k

for $y=y_i-k$ to y_i+k

$$C(x,y)=C(x,y)+\frac{1}{k+1}$$

where λ is an integer and the initial value of C is zero. An example of an accumulator array is presented in Fig. 8.

Next, it is desirable to calculate the predominant positions (E_i^x, E_i^y), $i=1\ldots6$, of the end-points. The most predominant position of end-points (E_1^x, E_1^y) corresponds to the global maximum of C. That is

If $C(x,y)=\max$ for $x=x\max \wedge y=y\max$ then $E_1^x, E_1^y = (x\max, y\max)$.

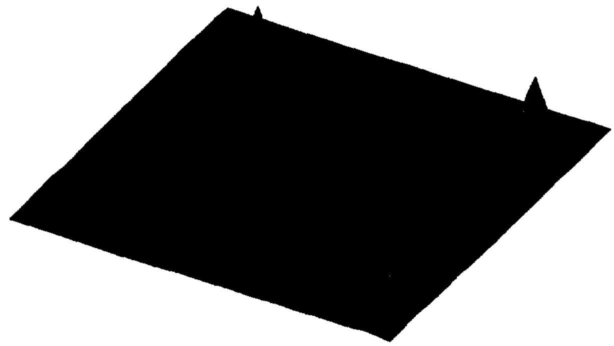

Fig. 9. Predominant positions of end-points.

The other five predominant positions of the end-points are calculated from the maxima of C after decreasing the area of the accumulator array near every maximum ($x$max,$y$max):

for $x=x\max-k$ to $x\max+k$

for $y=y\max-k$ to $y\max+k$

$$C(x,y)=\frac{C(x,y)}{(k+1)a}$$

where a is a parameter that depends on the distance between the desired predominant points.

Figure 9 shows the 6 predominant positions of all the end-points that have the accumulator array of Fig. 8. In the same way, the predominant positions of tree-points (T_i^x, T_i^y) and of parameters P_i (using a one-dimensional accumulation array) can be calculated. For the experiments, $\lambda=5$ and $a=0.8$ were used.

AUTHORS' BIOGRAPHIES

Nikos Papamarkos received a Diploma in Electrical Engineering in 1979 from the University of Thessaloniki, Greece and a Ph.D. degree in Electrical Engineering from the Democritus University of Thrace, Greece in 1986 where he is now an Associate Professor. His current interests are in the areas of digital signal processing, filter design, algorithms, image processing, pattern recognition and computer vision. Dr Nikos Papamarkos is a member of the IEEE, a member of the New York Academy of Sciences and a member of the Greek Technical Chamber.

Basilios Gatos received a Diploma in Electrical Engineering from the Democritus University of Thrace in 1992. At present, he is working on his Ph.D. degree in the Electrical & Computer Engineering Department of Democritus University of Thrace in the field of Optical Character Recognition. Since January 1993, he has been a Postgraduate Scholarship holder of the Institute of Informatics and Telecommunications, National Research Centre "Demokritos". His current interests include document analysis and image processing. B. Gatos has been a member of the Technical Chamber of Athens since 1992.