



ELSEVIER

Image and Vision Computing 16 (1998) 879–896

image  
AND  
VISION  
COMPUTING

# Text identification for document image analysis using a neural network

C. Strouthopoulos, N. Papamarkos\*

*Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece*

Received 21 March 1997; accepted 12 November 1997

---

## Abstract

A new bottom-up method is described that clusters the content of a mixed type document into text or non-text areas. The proposed approach is based on a new set of features combined with a self-organized neural network classifier. The set of features corresponds to the contents and the relationship of  $3 \times 3$  masks, is selected by using a statistical reduction procedure, and provides texture information. Next, a Principal Components Analyzer (PCA) is applied, which results in a reduced number of ‘effective’ features. The final set of features is then utilized as input vector into a proper neural network to achieve the classification goal. The neural network classifier is based on a Kohonen Self Organized Feature Map (SOFM). Document blocks are classified as text, graphics, and halftones or to secondary subclasses corresponding to special cases of the primal classes. The proposed method can identify text regions included in graphics or even overlapped regions, that is, regions that cannot be separated with horizontal and vertical cuts. The performance of the method was extensively tested on a variety of documents with very promising results. © 1998 Elsevier Science B.V. All rights reserved.

**Keywords:** Block classification; Document segmentation; Page layout analysis; Neural network classifiers

---

## 1. Introduction

The use of paper documents is a common way for information transfer and communication. The conversion of paper documents in a proper electronic form is essential for its processing, understanding, archiving, and transmitting by computers. An important procedure in digital processing of documents is the Page Layout Analysis (PLA). The goal of the PLA is to discover the formatting of the text and, from that, to derive meaning associated with the positional and functional blocks in which the text is located [1–3]. As a result, it labels the parts of a document as text, halftones, and line drawings [4]. To achieve PLA the document must be first separated in meaningful blocks which include data having common and certain homogeneous attributes, for example blocks containing only text or only line drawings or halftone images. The PLA of mixed type documents is a prerequisite to facilitate later processes such as the recognition of text, vectorization of graphics, and compression of images. For example, when a document is to be processed by an optical character recognition (OCR) system it is necessary to separate text from halftones and line drawings, so that time will not be wasted

in attempting to interpret the graphics as text. Also, the improvement of the compression ratio, if we encode text and image areas of a document with different methods, is useful for its electronic archiving and transmission.

There are many techniques proposed for document segmentation. These techniques can be classified as either top-down (or model driven) [4–8], bottom-up (or data driven) [9–11], and hybrid [12–15]. In top-down methods, a document is first split into major regions (large components), and major regions into smaller subregions (more detailed components), and so on. Most of the top-down techniques are based on the run length smoothing (RLS) algorithm [5], also called constrained run length (CRL) algorithm [6], and the projection profile cuts [7]. The RLS algorithm imposes a smoothing on the binary form of the document using two predetermined parameters, one for the vertical and one for the horizontal smoothing of the blocks. For the block classification, additional parameters (defined in a heuristic way) are used leading, to the necessity to train the system with documents having similar fonts or other common morphological characteristics. It must be noted that the method is not robust since if the assumptions made to determine the heuristic parameters are not satisfied, the method will fail. Another disadvantage of this method is based on the assumption that pages consist of only rectangular and orthogonal blocks, which can be

---

\* Corresponding author. Tel.: +30 541 26478; fax: +30 541 79569; e-mail: papamark@voreas.ee.duth.gr

separated by horizontal and vertical cuts. So, for documents where halftones and graphics are intermixed or overlapped with text, the RLS based methods will fail.

Bottom-up methods involve grouping of pixels as connected components (marks) and merging these components into successively larger components. Kasturi and co-workers [9,10] proposed a method that starts by first finding the connected components of an image and then separating graphics from text using the relative frequency of occurrence of these components as a function of their areas. In the next step, an iterative procedure is used to improve the initial estimation by applying the Hough Transform to all connected components. The method performs well if the initial document conforms to certain requirements about the size of characters, the interline spacing, the character spacing, and the resolution. This method is quite complex and as the authors have reported, computationally expensive. The document segmentation technique proposed by O'Gorman [11] is based on the document spectrum (docstrum) for  $k$ -nearest-neighbor marks. In this approach, the first step is the determination of marks and their centroids. Next, the direction vectors (distances, angles) of the  $k$ -nearest-neighbor connections of each mark are accumulated in a histogram. This histogram has two major peaks: one corresponding to intercharacter spacing and the other corresponding to the document skew. Using these peaks, a complex process groups the characters in words and text lines. This method, including both skew estimation and layout analysis, is computationally expensive and ineffective in multi-size, short text strings, and in halftone blocks.

Hybrid techniques are those that use local and global strategies in combination with texture analysis approaches. Fan et al. [12] proposed a document analysis method that first performs an RLS operation followed by a stripe merging procedure for block segmentation. Next, text blocks are classified by utilizing its periodic behavior in the  $y$ -projection. Finally, a feature-based classification scheme is performed to classify blocks into predetermined categories. Unfortunately, this method fails in many cases. First non-horizontal text or small text blocks do not satisfy the periodic behavior in the  $y$ -projection. Second, the feature set corresponds only to the number of black pixels included in a  $3 \times 3$  mask and therefore the correct classification of graphics and halftones is not guaranteed. Using Gabor filters, which have been used earlier for the general problem of texture segmentation [13], Jain and Bhattacharjee [14] proposed a method for text segmentation of gray-level documents. The method works well with low-resolution documents and is robust to skew. However, as the authors have reported, because of the frequency domain usage, this method is time consuming, requiring about 2 min of workstation CPU time for a  $512 \times 512$  image. Also, Gabor filtering techniques do not guarantee that this is the best way for a given texture segmentation and classification task [15]. The recent page segmentation technique proposed by Jain and

Zhong [15] is based on a set of texture masks in combination with a neural network classifier. In this method, text and line drawing regions are first classified in the same class. Unfortunately, the following discrimination of these two classes, which is a crucial step for text identification, is achieved by an ineffective procedure which is based only on the size of the connected components in the blocks. Strouthopoulos et al. [16,17] proposed a method for identification of text only areas in documents. The first stage of this method is an effective bottom-up technique, which results in elongated orthogonal blocks using a mark extraction and a bounding box connection algorithm. Next, these rectangles are classified as text or non-text areas by using thirteen  $3 \times 3$  masks called document structural elements (DSEs). Usually, this method gives satisfactory segmentation results. However, due to the type of feature set and the classification scheme used, this method fails when the documents include elongated line drawings or some kind of halftones.

This paper discusses a research effort that is intended to develop a document analysis system which can automatically identify text regions by classifying text, graphics and halftone blocks embedded in a document. In this paper, we propose a new page segmentation method that uses spatial structural features in combination with a neural network classifier. The set of features in addition to the use of DSEs, combines the contents of neighbor pairs of  $3 \times 3$  masks and therefore provides two-dimensional texture information. Applying a feature selection technique 34 effective features only remain, which are then taken as input to a PCA. The result of this procedure is a reduced number of eight ‘effective’ features, which are then used by a neural network classifier. The PCA is trained with the generalized Hebbian algorithm (GHA) [18–21]. It transforms the input data vectors to achieve as much of the total variation as possible. The neural network classifier is a Kohonen SOFM, which has as input the transformed vectors, resulting from the first stage and as output a competition layer consisting of an  $8 \times 8$  grid.

The proposed document block classification method is distinguished from other similar techniques due to the following desirable properties:

1. the robust and fast extraction of the document marks and blocks,
  2. the independence of the type and the size of the characters and also, the type of halftones and graphics,
  3. it can identify new desirable classes of blocks, such as blocks having italic characters, characters of specific font size, and graphics with predefined texture characteristics,
  4. it can identify text regions included in graphics or even overlapped regions, that is, regions that cannot be separated with horizontal and vertical cuts (Manhattan layout [1]). In comparison with the previous method [16,17], the new approach provides:
1. the improvement of the feature set by using an additional

- set of features, which includes information about the two-dimensional texture of the blocks,
2. the application of a feature reduction technique that is not based only on a statistical selection scheme but on the use of the PCA method,
  3. the use of the SOFM that optimally identifies the block clusters.

The method was tested with many mixed type documents. For the lack of comparison, many of the test documents were taken from the UW Document Image Database [22]. The method was compared with the commercial products Recognita Plus and Anagnostis. In all the test cases, the page layout procedure of these products is not applicable on special document cases such as the documents used in our examples. This paper presents characteristic examples that cover special layout difficulties. The experimental results confirm the effectiveness of the proposed method.

## 2. Block extraction

A mark is a connected group of black pixels which have the following property: for every pair of pixels belonging to a mark there is always a path leading from one pixel to the other. After the extraction of the marks, every mark is included in a rectangular bounding box. For example, the application of this procedure to the document of Fig. 1 results in the bounding boxes of Fig. 2.

According to the heights of the bounding boxes, a histogram is formulated whose main peaks can be determined by using the hill-clustering algorithm [23,24]. The result of this clustering procedure is the determination of the histogram peaks, which correspond to the distribution of the most used character sizes of the document. For a typical document, there is often a global peak for the distribution of characters of the predominant size, and smaller peaks for the rest of the characters and the noise. Thus, the histogram takes a global maximum value for the boxes that bound the characters of the most common size. Consequently, each iteration of the block extraction algorithm corresponds to a histogram peak. The procedure is repeated peak by peak from the biggest to the smallest peak until no other significant peaks exist. For each iteration, corresponding to a histogram peak value  $H_{\max}$ , those boxes with heights  $h_i$  satisfying the following condition are accepted only:

$$\frac{H_{\max}}{2} \leq h_i \leq 2H_{\max}. \quad (1)$$

According to Ref. [25], each text string must contain type of similar size (a range of less than two to one in point size). Thus, it can be easily seen how in Eq. (1) the coefficients 1/2 and 2 have been selected. Similar coefficients are used in Ref. [9].

### 2.1. Extension and merging of the bounding boxes

After the construction and filtering of the bounding boxes, these are extended to chains of connected boxes. This procedure is important for many reasons but mainly for the extraction of text lines and the independence of small document skews. Fig. 3 shows the remaining boxes, after height filtering and after having them extended by adding two equal rectangular extensions to their left and right sides (see Fig. 4). Referring to this figure, the value of  $h_1$  is taken equal to the histogram  $H_{\max}$  in order to merge together the adjacent and extended boxes that are collinear and the distance between them is less than  $2H_{\max}$ . Obviously,  $H_{\max}$  approaches the local average character height [25]. The values  $h_2 = H_{\max}/2$  and  $h_3 = H_{\max}/4$  have been chosen so that the boxes of high characters (like h, l) or of characters with a tail (like p, q), can be joined with their adjacent boxes. Value  $h_3$  is enough for the connection of bounding boxes of text lines with small skew. A larger value of  $h_3$  produces a larger tolerance to skew. The value of  $h_3$  satisfies the conditions described in Ref. [25]. That is, the leading, or interline spacing between lines of text, must not be excessively small. It must be more than 25% of the average letter height.

The extension and merging procedure results in the joining between the bounding boxes of a text line and the creation of chains of boxes which correspond to the text lines. In practice, this procedure substitutes the complex and time-consuming use of the Hough Transform for collinearity and vicinity checking of the bounding boxes.

Considering the result of the previous step as a new binary image, we surround each mark (which is now composed of extended-connected boxes) with new rectangles. It is obvious that the rectangles, which enclose extended boxes of a text line, are elongated. On the contrary, boxes that are located excessively far from their adjacent boxes or boxes that constitute a small isolated group either do not create overlapping boxes or create chains of relatively small length. Fig. 5 shows the chains of the overlapping boxes and the new rectangles that surround the chains. We accept those new rectangles that have a base to height ratio greater than 3.5. The threshold value 3.5 corresponds to the case with only two connected characters.

After the above process is completed, the elongated blocks of the document, which consist of collinear and neighboring components of almost the same height, have been detected. The majority of these blocks correspond to text lines. However, it is possible to have horizontal dispositions of halftones or drawings, which compose such blocks. The application of a block classification technique will discriminate and classify the blocks corresponding only to text.

## 3. Feature extraction

For the block classification, we use two sets of DSE

(2-16)	$(N+1) \times (N+1)$	$3N^2 + 8N + 8$
(2-17)	$(N+1)(N+1) \times 1$	$N^3 + 3N^2 + 3N + 1$
(2-18)	$(N+1) \times 1$	$N^3 + 2N^2 + N$
(2-19)	$N(N+1) \times 1$	$N^3 + 3N^2 + 3N + 2$
(2-20)	$N(N+1) \times (N+1)$	$N^3 + 2N^2 + N$
(2-21)	$1 \times 1$	$N^2 + N$
(2-22)	$N(N+1) \times 1$	$N^2 + N$
total cost : $7N^3 + 21N^2 + 22N + 11$		

### 3 Simulation results

In this section, we present the results of the noise removing of the lena image. LMS, RLS and fast M-D RLS algorithms have been used in order to remove the noise added to an original image. The following scheme have been used :

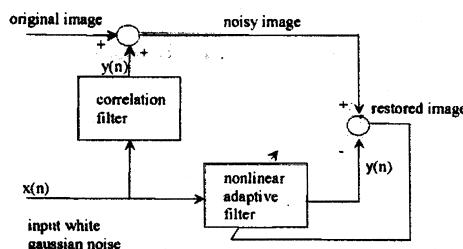


figure 3 : the adaptive noise canceller scheme

The correlation filter is a nonlinear filter given by

$$\begin{aligned} y(n) = & 0.6x(n) + 0.3x(n-1) + 0.001x^2(n) \\ & + 0.0005x(n)x(n-1) + 0.003x^2(n-1) \end{aligned}$$

The adaptive filter (or noise estimator) has been used in the experiment as linear or non linear adaptive filter. The corresponding results of the nonlinear case are reported in figures 4-8 :



figure 4 : original image

Fig. 1. Original binary document.

features. The first set is identical with the features used by Strouthopoulos et al. [17]. It is observed that this set of features does not represent the exact spatial information well in cases where there are large line drawings, especially lines, or some type of halftone in the blocks. For this reason an additional set of features is proposed which combines the



figure 5 : noisy image

SNR=4.59dB

figure 6 : nonlinear LMS

$\mu_1=0.02$   $\mu_2=0.01$

SNR=17.28dB



figure 7 : nonlinear RLS

$\alpha=100$ , SNR=21.2dB

figure 8 : fast M-D

nonlinear RLS

$E=1$ ,  $W=1$

SNR=22.54dB

### 4 Concluding remarks

We have presented an efficient algorithm for the adaptive Volterra second-order filter based on the M-D fast RLS algorithm. The computation complexity is of order  $O(N^3)$  multiplications per sample, which represents a substantial saving over direct implementation of the RLS algorithm. Further work should be done to achieve better computation cost, for example in our algorithm equation (2-22) has  $N(N+1)$  elements with  $N(N-1)/2$  zeros elements, we can think during implementation to avoid the computation of these elements.

### References

- [1] H. K. Baik and M. J. Mathews, 'Adaptive Lattice bilinear filters', *IEEE Trans. on SP*, vol 41, n°6, pp 2033-2046, 1993.
- [2] M. Bellanger, *Adaptive digital filters and signal analysis*, Marcel Dekker, Inc., 1987.

intrinsic information in neighboring DSEs. Next, we will describe in detail the two sets of features.

#### 3.1. First set of features

To express the spatial information of the blocks we use

[2-16]	$(N+1) \times (N+1)$	$3N^2 + 8N + 8$
[2-17]	$(N+1)(N+1) \times 1$	$N^3 + 3N^2 + 3N + 1$
[2-18]	$(N+1) \times 1$	$N^3 + 2N^2 + N$
[2-19]	$N(N+1) \times 1$	$N^3 + 3N^2 + 3N + 2$
[2-20]	$N(N+1) \times (N+1)$	$N^3 + 2N^2 + N$
[2-21]	$1 \times 1$	$N^2 + N$
[2-22]	$N(N+1) \times 1$	$N^2 + N$
total cost : $7N^3 + 21N^2 + 22N + 11$		

### 3 Simulation results

In this section, we present the results of the noise removing of the Lena image. LMS, RLS and fast M-D RLS algorithms have been used in order to remove the noise added to an original image. The following scheme have been used :

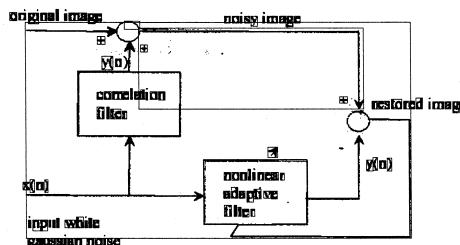


Figure 3 : the adaptive noise canceller scheme

The correlation filter is a nonlinear filter given by

$$\begin{aligned} y(n) = & 0.6x(n) + 0.3x(n-1) + 0.001x^2(n) \\ & + 0.0005x(n)x(n-1) + 0.003x^2(n-1) \end{aligned}$$

The adaptive filter (or noise estimator) has been used in the experiment as linear or non linear adaptive filter. The corresponding results of the nonlinear case are reported in figures 4-8 :



Figure 4 : original image



Figure 5 : noisy image  
SNR=4.59dB



Figure 6 : nonlinear LMS  
 $\mu_1=0.02$   $\mu_2=0.01$   
SNR=17.28dB



Figure 7 : nonlinear RLS  
 $\alpha=100$ , SNR=21.2dB  
figure 8 : fast M-D  
nonlinear RLS  
 $E=I$ ,  $W=I$   
SNR=22.54dB

### 4 Concluding remarks

We have presented an efficient algorithm for the adaptive Volterra second-order filter based on the M-D fast RLS algorithm. The computation complexity is of order  $O(N^3)$  multiplications per sample, which represents a substantial saving over direct implementation of the RLS algorithm. Further work should be done to achieve better computation cost, for example in our algorithm equation (2-22) has  $N(N+1)$  elements with  $N(N-1)/2$  zeros elements, we can think during implementation to avoid the computation of these elements.

### References

- [1] H. K. Bakh and M. J. Mathews, 'Adaptive Lattice bilinear filters', *IEEE Trans. on SP*, vol 41, n°6, pp 2033-2046, 1993.
- [2] M. Bellanger, *Adaptive Digital Filters and signal analysis*, Marcel Dekker, Inc., 1987.

Fig. 2. Bounding boxes after identification of marks.

features which can describe what type of  $3 \times 3$  masks are included in the blocks. The first set of features corresponds to 13 structural  $3 \times 3$  binary masks, which we call document structure elements. The order of pixels in the mask is as follows:

b <sub>8</sub>	b <sub>7</sub>	b <sub>6</sub>
b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>
b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>

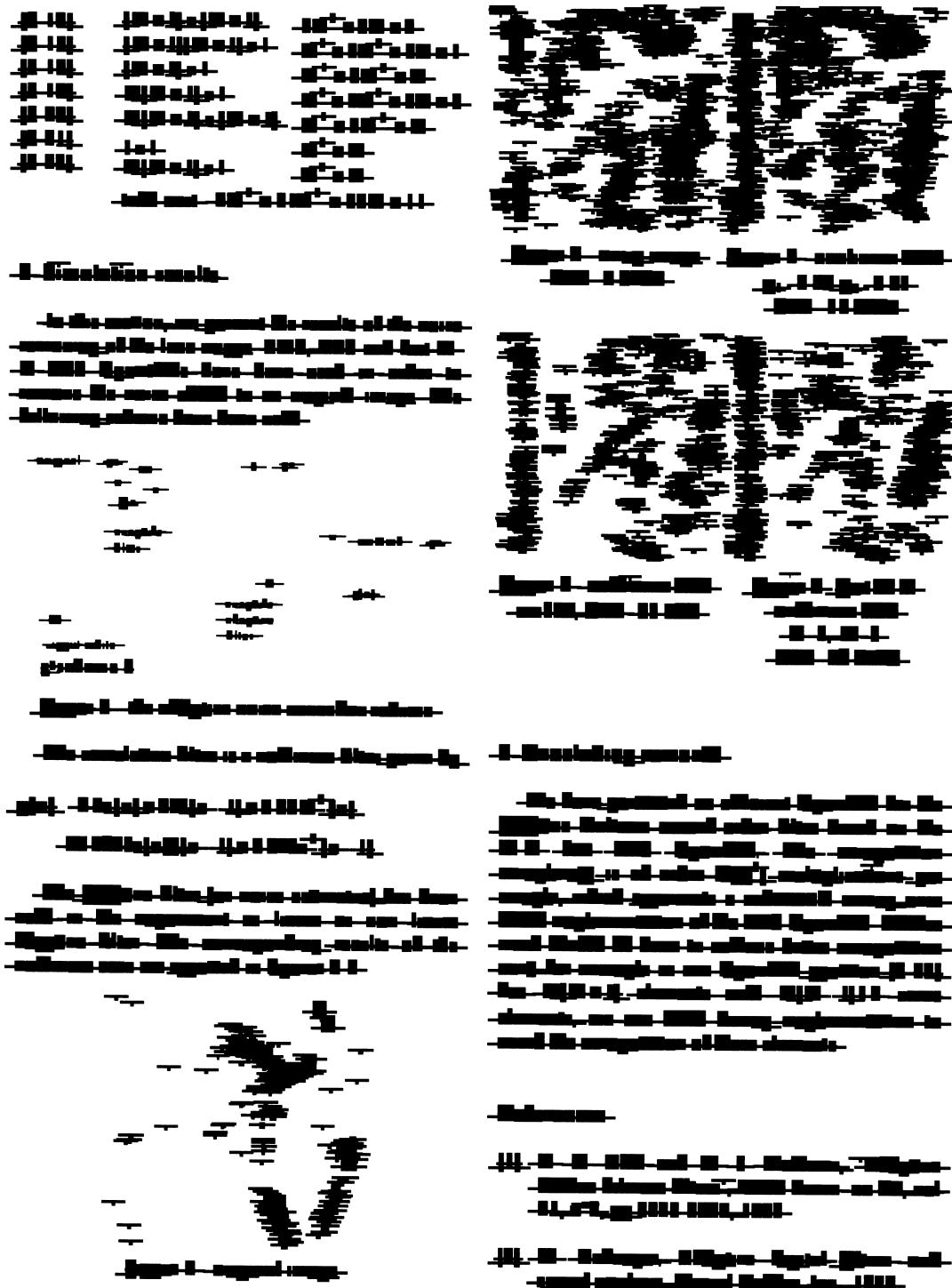


Fig. 3. Remaining boxes after their height filtering and extensions.

An integer

$$L = \sum_{i=0}^8 b_i 2^i$$

with  $b_i \in \{0,1\}$  is assigned to any DSE which is called the

document structure element characteristic number (DSECN). It is obvious that since  $L \in \{0,1,2,\dots,511\}$ , there are 512 different DSEs. For each block a histogram is formulated corresponding to the contribution of the DSEs (the 0 and 511 DSE are not considered because they correspond to pure background and object regions, respectively) in the block. For a rectangular block  $A$ , let  $K$  be the number of columns and  $J$  the number of rows.

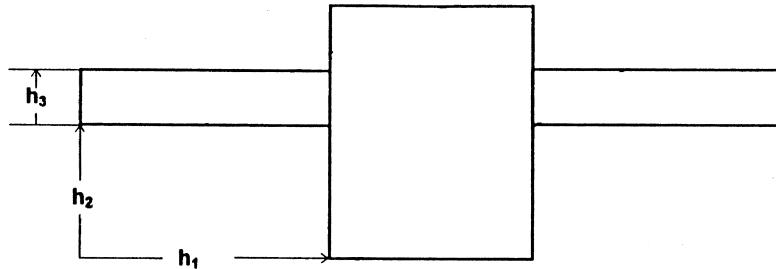


Fig. 4. Form of extension of a bounding box.

Obviously,  $A$  has  $(K - 2)(J - 2)$  DSEs which can formulate the histogram function and the probability density function of the DSEs:

$$H_1(l) = \frac{h_1(l)}{\sum_{l=1}^{510} h_1(l)}. \quad (2)$$

Fig. 6 shows the  $H_1(l)$  functions of the three typical blocks, i.e. blocks containing text, graphics, and halftone images.

The 510 values of  $H_1(l)$  can be taken as texture features. However, as we explain below, by the application of a feature selection technique, only 13 of them are finally selected as texture features.

### 3.2. Second set of features

The second set of features is used in order to overcome difficult block classification cases. From our experiments we have observed that when a block contains some kind of large line drawing or halftones then the use of the first set of features only results in wrong classification. This happens because the first set of features does not describe the morphology of graphics in the blocks well. To overcome these difficulties, additional features are used which are extracted by combining the information of pairs of neighboring DSEs. Specifically, for every DSE we examine, as a pair, each one of the eight neighboring DSEs. Doing this, a two-dimensional histogram  $h_2(l_1, l_2)$  is formulated that gives the contribution of each pair of DSEs in the block.

The two-dimensional histogram function  $h_2(l_1, l_2)$  of block  $A$  is obtained using the relation

$$h_2(l_1, l_2) = \begin{cases} h_2(l_1, l_2) + 1, & \text{if } l_1 = L_1(k, j) \text{ and } l_2 = L_2(k', j') \\ h_2(l_1, l_2), & \text{otherwise} \end{cases} \quad (3)$$

where

$$\begin{aligned} l_1 &\in \{1, 2, \dots, 510\}, \quad l_2 \in \{1, 2, \dots, 510\}, \\ k &\in \{4, \dots, K - 5\}, \quad j \in \{4, \dots, J - 5\} \end{aligned}$$

$$k' = k + a, \quad j' = j + b$$

$$(a, b) \in \{(-3, 0), (-3, -3), (0, -3), (3, -3), (3, 0), \\ \times (3, 3), (0, 3), (-3, 3)\}$$

$L_1(k, j)$  the characteristic number of a DSE in position  $j, k$  and  $L_2(k', j')$  the characteristic number of each neighboring DSE in position  $k', j'$ .

For this histogram, the probability density function  $H_2(l_1, l_2)$  is equal to

$$H_2(l_1, l_2) = \frac{h_2(l_1, l_2)}{\sum_{l_2} \sum_{l_1} h_2(l_1, l_2)}. \quad (4)$$

Fig. 7 shows the functions for the three typical blocks, i.e. blocks containing text, graphics, and halftone images.

Using the probability density function  $H_2(l_1, l_2)$  we can extract  $510^2$  features. However, as in the case of the first feature set, the feature selection procedure results in only 21 features. So, the total number of features extracted from the two feature sets is 34.

### 3.3. Feature selection

In order to select the feature space and to have only the most powerful of them, a feature selection procedure is necessary. In this approach, feature selection is based on stability, separability, and similarity criteria. The entire procedure is performed on a large number of test blocks extracted from different types of document and for each set of features separately.

The stability test tries to identify the features, which give significantly stable performance. We accept only the highly stable features having normalized standard deviation  $\sigma * < 0.01$ .

For the separability test, we compute for every feature  $l$  belonging to two classes  $i$  and  $j$  the separability factor  $S_{\text{avg}}^l$ .

For the separability test, for every feature  $l$  belonging to classes  $i$  and  $j$  the separability factor  $S_{\text{avg}}^l$  is computed:

$$S_{\text{avg}}^l = \frac{|(\mu_i^l)^2 - (\mu_j^l)^2|}{\sqrt{(\sigma_i^l)^2 + (\sigma_j^l)^2}} \quad (5)$$

where  $\mu_i^l, \mu_j^l$  are the mean values and  $\sigma_i^l, \sigma_j^l$  the standard deviation values for each class. A large separability means

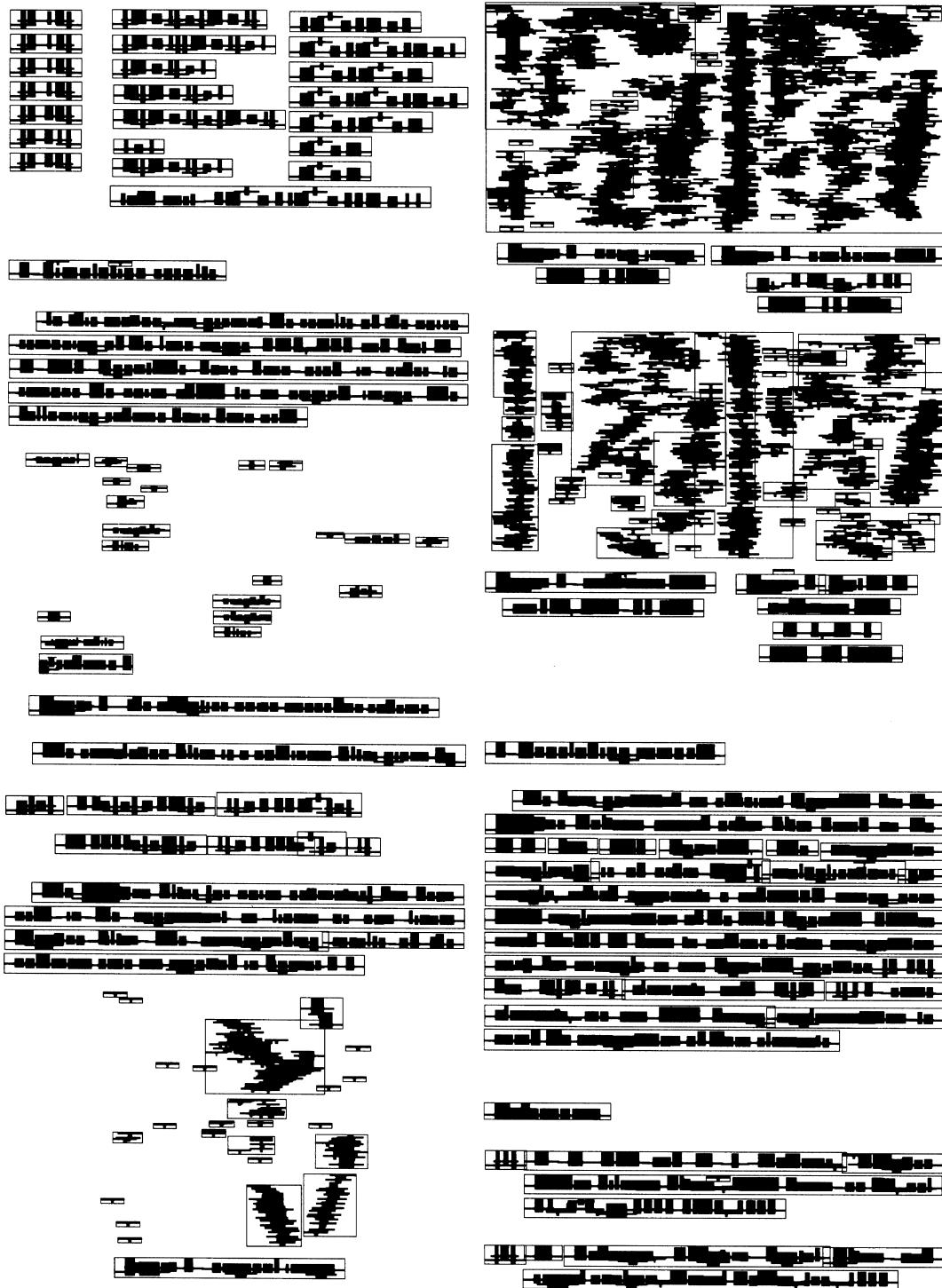


Fig. 5. Chains of overlapping boxes.

that the feature has a good distinguishability for the two classes. The separability factor has been computed for any feature and for all class pairs. A proper threshold value for the separability factor is 3. For the first feature set, the separability test procedure rejects 436 features.

The feature similarity analysis estimates for every two features  $l$  and  $m$ , belonging to the same class  $p$ , the

correlation factor

$$C_{lm}^p = \frac{\frac{1}{n_p} \sum_{i=1}^{n_p} (l_i - \mu_l) \cdot (m_i - \mu_m)}{\sigma_l \cdot \sigma_m} \quad (6)$$

where  $n_p$  is the number of elements of class  $p$ ,  $l_i$  and  $m_i$  the

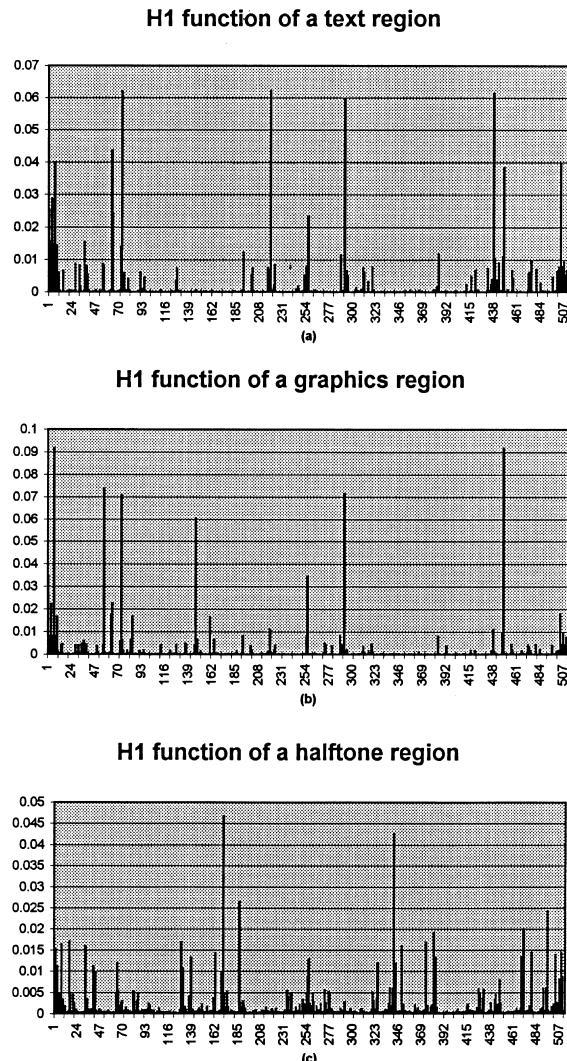
Fig. 6. Probability density functions  $H_1(l)$  of typical regions.

Table 1  
The first set of 13 DSE features

219	73	438	292	1
256	170	341	186	495
448	7	56		

feature values of element  $i$ , and  $\mu_l$ ,  $\mu_m$ ,  $\sigma_l$  and  $\sigma_m$  the means and standard deviations of the features in class  $p$ . Of the features that show high correlation,  $C_{\text{avg}} > 0.9$ , those with high separabilities are retained and the others deleted. We have found that 49 features can be rejected for the first feature set. Therefore, the application of the above feature selection process, for the first feature set, results in only 13 features corresponding to the DSEs given in Table 1. It is noted that the threshold values of the entire feature selection procedure are taken from the previous work of Driels and Nolan [26].

To speed up the selection process of the second feature set, first, the features having approximately near zero values are rejected. We apply this criterion because very low and similar feature values correspond to a low separability factor. Doing this, a total of 259 941 features are rejected and in the remaining features we apply the stability, separability, and similarity criteria in the same way as in the first set. The entire feature selection procedure leads to the selection of only 21 features shown in Table 2. Specifically, the application of stability, separability, and

## H2 function of a halftone region

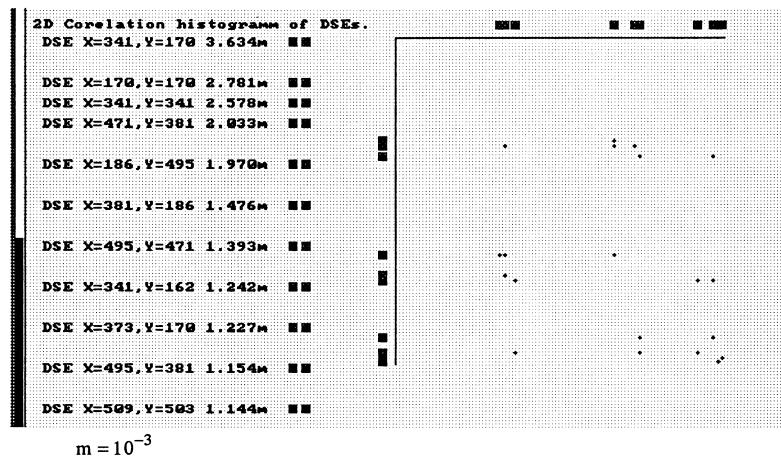
Fig. 7. Probability density functions  $H_2(l_1, l_2)$  of typical regions.

Table 2  
The second set of features

 292	 219	 438	 438	 292	 292	 219	 219
 73	 73	 292	 73	 448	 448	 7	 7
 63	 63	 56	 56	 341	 170	 170	 170
 341	 341	 381	 471	 495	 186	 495	 471
 381	 186	 495	 381	 510	 255	 510	 507
 447	 255						

similarity tests reject 14, 87, and 37 features, respectively. As it can be observed from Tables 1 and 2, these results are reasonable. For example, the main components of halftone blocks contain 170, 341, 186, 495, 381, and 471 DSEs. Thus, the final feature set consists of only 34 features, which after a linear normalization corresponds to a 34-dimension space.

In order to explain the contribution of the use of the second feature set, let us consider the two images of Fig. 8. For these images the non-zero values of the first feature set are shown in Table 3 and are exactly the same. On the contrary, because of the different values of the second feature set, these images can be clearly identified. The non-zero values of the second feature set are given in Table 4.

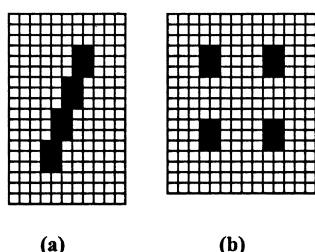


Fig. 8. Test images.

Using this feature set we can say that each block is represented by its corresponding vector in the feature space. However, in order to achieve better classification results we apply a PCA procedure, which drives a linear combination of the features to a self-organized neural network.

#### 4. Principal component analysis

The feature set selected is considered as inputs to a neural network PCA system. PCA provides a means by which to achieve such a transformation where the feature space accounts for as much of the total variation as possible. Specifically, using the PCA procedure the original feature space is transformed into another feature space that has exactly the same dimension as the original. However, the transformation is designed in such a way that the original feature set may be represented by a reduced number of ‘effective’ features and yet retains most of the intrinsic information content of the data. Therefore, using PCA we achieve not only the increasing of feature variation but also decreasing of feature space dimensionality. Karhunen–Loeve transformation (KLT) [18] is a well-known tool for PCA in multivariate analysis. A complete analysis of the

Table 3

First set of features		Values for Fig. 8(a)	Values for Fig. 8(b)
DSE 438		4	4
DSE 292		4	4
DSE 73		4	4
DSE 219		4	4
DSE 256		4	4
DSE 1		4	4

PCA method used in this paper is given in Refs. [19,20]. The form of PCA is shown in Fig. 10 as the first part of the entire neural network. Briefly, in this approach the input vector of the PCA is the 34 feature vector  $\mathbf{x}$  while the output is an 8-dimension vector  $\mathbf{y}$ , which is given by the relation:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \quad (7)$$

where the transformation matrix  $\mathbf{W}$  contains the neural network coefficients. It is noted that after training,  $\mathbf{W}$  will approach the matrix whose rows are the first  $M$  eigenvectors of  $\mathbf{C}$ , ordered by decreasing eigenvalues, where  $\mathbf{C}$  is the covariance matrix of the training set.

The selection of eight output neurons is explained as follows.

If  $M$  takes a value less than 34 a reduction of the  $y$ -dimension is achieved. To find the optimal value of  $M$  the mean square error  $\bar{\epsilon}^2(M)$  [19,20] is estimated for  $M = 1, 2, \dots, 33$  according the relations:

$$\bar{\epsilon}^2(M) = E[\epsilon^2] \quad (8)$$

$$\epsilon = \sum_{i=M}^{33} y_i \mathbf{w}_i \quad (9)$$

where  $y_i$  is the  $i$  output,  $\mathbf{w}_i$  the coefficients vector of the  $i$  neuron, and  $E[.]$  the statistical expectation operator which is computed for the whole training set. As we can see from Fig. 9,  $M = 8$  is the lower value in which  $\bar{\epsilon}^2(M)$  is practically zero and therefore it can be considered as optimal.

The neural network is trained by the Generalized Hebbian Algorithm (GHA). GHA is an unsupervised learning algorithm based upon a Hebbian learning rule. It calculates the eigenvectors of the covariance matrix  $\mathbf{C}$  of the training feature set and it has been proven that it converges with probability one. In contrast to the Karhunen–Loeve algorithm, this approach does not need to compute  $\mathbf{C}$

analytically, since the eigenvectors are derived directly from the data.

## 5. The neural network classifier

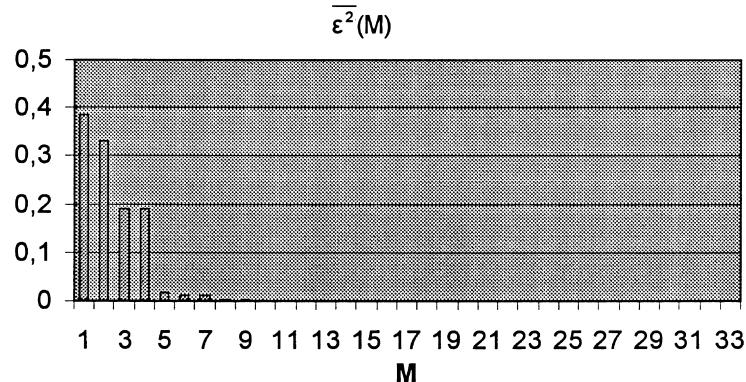
The neural network classifier is based on a Kohonen SOFM. The inputs of this neural net are the outputs of the PCA. The network combines its input layer with a competitive layer of neurons, and is trained by unsupervised learning. The two layers are fully interconnected since every input is connected to all of the neurons in the competitive layer. Typically, the competitive layer is organized as a two-dimensional square grid, and each neuron represents a class. Kohonen self-organization has as a result the representation of similar classes, by neighboring neurons on the competitive layer.

Fig. 10 gives the Kohonen SOFM topology as the second part of the entire neural network. In this approach the competition layer has an  $8 \times 8$  grid. Although the grid is two-dimensional, each neuron is labeled by an index  $k$  which takes values from the set  $\{0, 1, 2, \dots, 63\}$ .

After training, each neuron on the output layer represents a class of patterns. Patterns of large similarity are represented by the same neuron on the grid. Each neuron is labeled by the identity of the patterns that were classified on it. The labeling method of SOFM is based on the density-matching criterion [20]. More, specifically, due to the high discrimination ability of the input feature vector, the distributions of occurrences (votes) in the output neurons have small standard deviations. That is, the majority of occurrences in each neuron in the competition layer corresponds clearly to one of the three main block categories. Thus, after the training stage it is easy to label each neuron with the correct block class. The Kohonen

Table 4

Second set of features		Values for Fig 8(a)	Values for Fig 8(b)
DSEs 292-219		3	0
DSEs 292- 73		4	4

Fig. 9. The graph of function  $\overline{\varepsilon^2}(M)$ .

feature map organizes the neurons of the competitive layer in such a way that similarities among patterns are mapped into closeness relationships of the competitive layer grid. Also, the Kohonen SOFM provides advantages over classical pattern-recognition techniques because it utilizes the parallel architecture of a neural network and provides a graphical organization of pattern relationships [21].

The final segmentation results for the document of Fig. 1 are shown in Fig. 11. The document of Fig. 11 has a size of  $1700 \times 2312$  pixels and a resolution of 300 dpi. Text regions have been identified properly by the method and are shown as shadowed rectangles.

## 6. Experimental results

### 6.1. Neural network training experimental results

The training of the neural network classifier is based on a set of representative blocks derived from different document types having a resolution in the range 100–300 dpi. Every block was corresponded to any of the

three basic classes of text, graphics, and halftones. Text blocks of characters of different sizes and font types, graphics blocks with different thickness, and blocks of images displayed with different halftone techniques were included in the training set. First, the PCA neural network was trained for various numbers of neurons and finally eight neurons were selected. Next, the SOFM neural network classifier was trained. Fig. 12 illustrates the training results. Each neuron is painted according to the class it is represented by. It can be observed that neurons of similar patterns create uniform groups, which correspond to the three main classes.

The results of training were tested with two sets of experiments. The first set of 500 blocks was obtained from documents belonging to the training set, while the other set of 2500 blocks was obtained from documents out of the training set. Table 5 summarizes the classification results for text, graphics, and halftone classes.

Another significant result is the identification of block subclasses that are not obvious in the beginning. Specifically, there are some neurons, each of them represents exclusively one pattern type, which can be

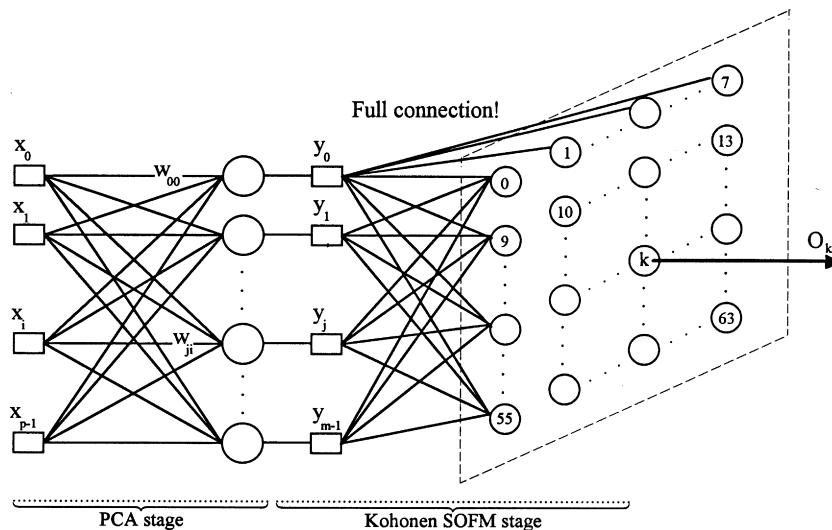


Fig. 10. The two stages of a neural network.

(2-16)	$(N+1) \times (N+1)$	$3N^2 + 8N + 8$
(2-17)	$(N+1)(N+1) \times 1$	$N^3 + 3N^2 + 3N + 1$
(2-18)	$(N+1) \times 1$	$N^3 + 2N^2 + N$
(2-19)	$N(N+1) \times 1$	$N^3 + 3N^2 + 3N + 2$
(2-20)	$N(N+1) \times (N+1)$	$N^3 + 2N^2 + N$
(2-21)	$1 \times 1$	$N^2 + N$
(2-22)	$N(N+1) \times 1$	$N^2 + N$
total cost : $7N^3 + 21N^2 + 22N + 11$		

### 3 Simulation results

In this section, we present the results of the noise removing of the lena image. LMS, RLS and fast M-D RLS algorithms have been used in order to remove the noise added to an original image. The following scheme have been used :

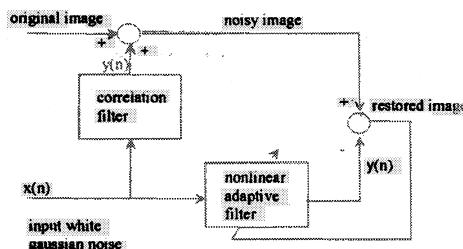


figure 3 : the adaptive noise canceller scheme

The correlation filter is a nonlinear filter given by

$$\begin{aligned} y(n) = & 0.6x(n) + 0.3x(n-1) + 0.001x^2(n) \\ & + 0.0005x(n)x(n-1) + 0.003x^2(n-1) \end{aligned}$$

The adaptive filter (or noise estimator) has been used in the experiment as linear or non linear adaptive filter. The corresponding results of the nonlinear case are reported in figures 4-8 :

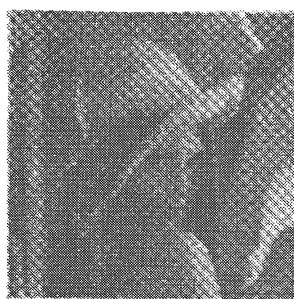


figure 4 : original image

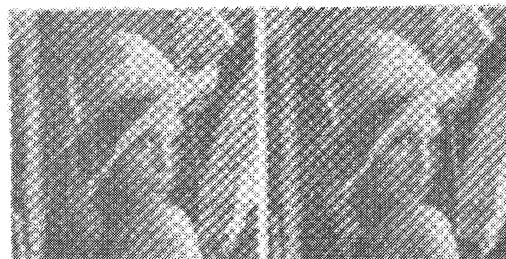


figure 5 : noisy image  
SNR=4.59dB  
 $\mu_1=0.02$   $\mu_2=0.01$   
SNR=17.28dB

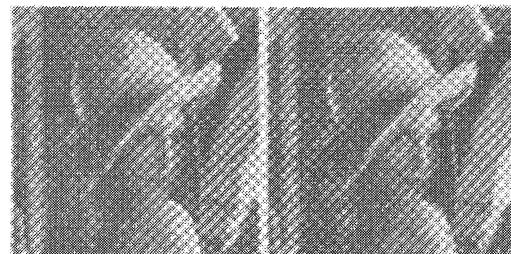


figure 6 : nonlinear LMS  
 $\alpha=0.02$   $\mu_1=0.01$   
SNR=17.28dB

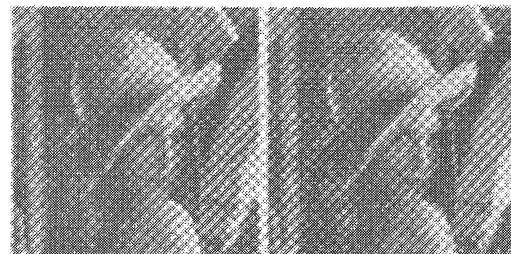


figure 7 : nonlinear RLS  
 $\alpha=100$ , SNR=21.2dB  
 $E=1$ ,  $W=1$   
SNR=22.54dB

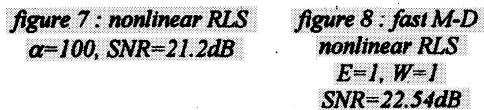


figure 8 : fast M-D nonlinear RLS  
 $E=1$ ,  $W=1$   
SNR=22.54dB

### 4 Concluding remarks

We have presented an efficient algorithm for the adaptive Volterra second-order filter based on the M-D fast RLS algorithm. The computation complexity is of order  $O(N^3)$  multiplications per sample, which represents a substantial saving over direct implementation of the RLS algorithm. Further work should be done to achieve better computation cost, for example in our algorithm equation (2-22) has  $N(N+1)$  elements with  $N(N-1)/2$  zeros elements, we can think during implementation to avoid the computation of these elements.

### References

- [1] H. K. Baik and M. J. Mathews, 'Adaptive Lattice bilinear filters', *IEEE Trans. on SP*, vol 41, n°6, pp 2033-2046, 1993.
- [2] M. Bellanger, *Adaptive digital filters and signal analysis*, Marcel Dekker, Inc., 1987.

Fig. 11. Final text identification results.

considered as a subclass of one of the basic classes. That is, each main class, which consists of a number of neurons, includes a number of subclasses. In the  $8 \times 8$  grid the competition layer is expected to have subclasses. The observation of subclasses is made after the test procedure by taking into account the type of classified blocks. Such neurons are mentioned below.

Neuron	Subclass
52	Text of Italic characters
3	Text of Arial font characters
12	Text of Roman font characters
13	Text of Sanserif font characters
42	Text of underline characters
59	White Noise Halftoning
51	Clustered-Dot Ordered Dither

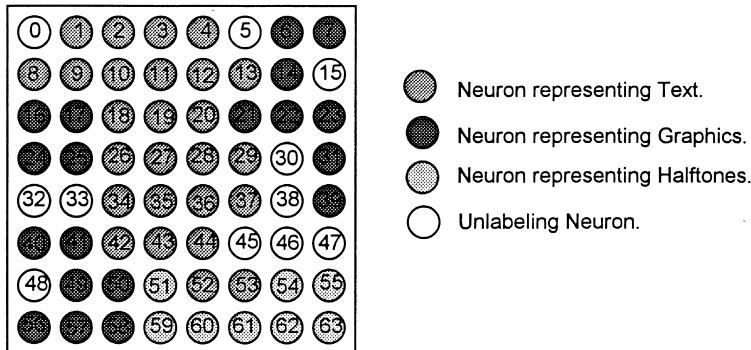


Fig. 12. Kohonen SOFM after training.

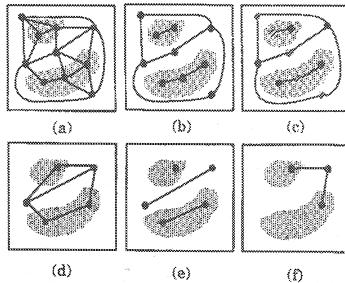


Fig. 2. Example of extracting connected components from a binary image. (a)  $G[l]$ . (b)  $G'[l]$ . (c) Survivor cells (filled circles) and the allocation of children (hatched circles) at level  $l$ . (d)  $G[l+1]$ . (e)  $G'[l+1]$ . (f)  $G[m]$  for  $m = l + 2$ , the root level.

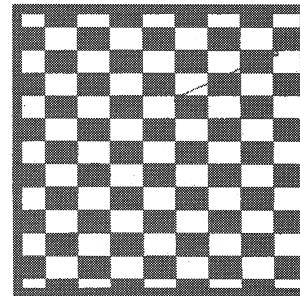


Fig. 3. The checkerboard image with the adjacency graph of the root level superposed.

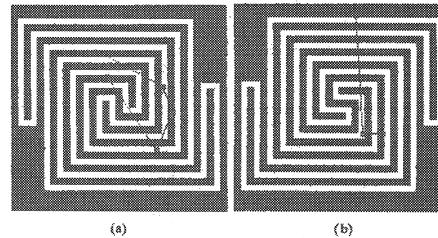


Fig. 4. The connectedness puzzle of Minsky and Papert. (a) Two white and three black bands. (b) One white and two black bands. The adjacency graph of the root level is superposed.

of the edge  $(c_0, c_1)$ . The case  $\lambda = 0$  is equivalent to removing the edge from  $E[l]$ . Let  $E'[l]$  be the set of edges having  $\lambda = 1$ , and let  $G'[l] = (V[l], E'[l])$ . The connected components in the labeled image are represented by connected components in the graph  $G'[l]$ , for all  $l \geq 0$ .

The subgraphs of  $G'$  are processed independently, each subgraph being recursively contracted into one vertex, the *root* of the connected component. The contraction process is based on the technique described in the previous section: first the survivor vertices are designated and then the nonsurvivor vertices are locally allocated to survivors. If a nonsurvivor has more than one survivor neighbor it chooses the one carrying the largest outcome of the random variable  $x$  from the last iteration of the decimation process. Because the neighbors are neighbors in  $G'$ , the survivors can only have children belonging to their own class. Thus from each connected component of the input image a pyramidal hierarchy of irregular tessellations is built in  $O(\log(\text{component\_size}))$  steps.

The different hierarchies may have different heights, but in  $\log[\max(\text{component\_size})]$  steps the entire image is reduced to roots. This situation is detected at the level  $m$  when  $E'[m]$  becomes empty. Evidently  $\text{component\_size}$  can differ from  $\text{image\_size}$ . For example, a connected linear pattern passing through every second row of the image has length  $N(N+1)/2$  pixels. Since the hierarchy is built over the pattern the number of levels depends on its intrinsic diameter.

At each level, the graph  $G[l]$  includes edges between cells that arise from different labels; it preserves the spatial relations among the connected components. At the root level,  $G[m]$  is the adjacency graph of the original labeled image; it has one vertex for each connected component and its edges represent the adjacencies between these components.

Fig. 2(a) shows an example of a graph  $G[l]$  superposed on the binary image from which it was derived. The induced graph  $G'[l]$  is shown in Fig. 2(b). Note that in  $G'[l]$  each connected component corresponds to a connected subgraph. The cells surviving level  $l$  and the allocation of the nonsurvivors are shown in Fig. 2(c). The graph  $G[l+1]$  of the next level is shown in Fig. 2(d) and the corresponding graph  $G'[l+1]$  in Fig. 2(e). Level  $l+2$  is the root level and its graph  $G[m]$  is shown in Fig. 2(f). It correctly represents the adjacency relations among the three connected components of the image: the background and the two blobs.

In Fig. 3 a *checkerboard* image and the adjacency graph of its root level are shown. The *checkerboard* is a “worst-case” image, the two connected components (both defined by the relation of eight-connectedness) being distributed across the entire input. Cibulskis and Dyer [10] employed a regular pyramid structure to segment this image. In their results the “white” component was allocated to one root at the apex, but the representation of the black squares had to be spread over several levels. The size of the image is  $64 \times 64$  and the two roots were obtained at the eighth level of the hierarchy. Recall that the height of the hierarchy depends on  $\text{component\_size}$ . Since random processes are involved in the construction of the irregular tessellations the location of the roots depends on the outcomes of local processes. Nevertheless, the same root level adjacency graph is always obtained at the top of the hierarchy.

The famous connectedness puzzle of Minsky and Papert [27, Fig. 5.1] can be solved by our technique in  $O(\log(\text{component\_size}))$  steps. The pattern in Fig. 4(a) contains three black and two white bands, while in the pattern in Fig. 4(b) the two white bands are connected, leaving only two black bands. The adjacency graphs obtained at the root level clearly show the different topologies.

The irregular tessellations that arise in the hierarchies defined by the connected components do not convey meaningful representations at intermediate levels. Let us define the *receptive field* of a cell on level  $l$  as the set of all the pixels at level 0 (input) associated with it. This field is always a connected set

Fig. 13. Results for a document from the UW database.

Table 5

Percentage of correct block classification

	Blocks of training set (%)	2500 blocks out of the training set (%)
Text	99.5	98.5
Graphics	99.8	98.6
Halftones	100.0	99.5

This is a promising advantage of the proposed block classification procedure because it gives additional valuable information for the blocks.

### 6.2. Experimental results of text block identification

The proposed method was extensively tested on a series of mixed type documents. In addition to our test documents, the method has also been applied to some documents obtained from the University of Washington database [22]. Also, the method is compared with the commercial products Recognita Plus, TextBridge and Anagnostis. In all the test cases, the page layout procedures of these products are not applicable on special document cases such as the documents used in our examples. The experimental results show clearly the improvement of the block classification. Blocks that were misclassified using the first feature set, now are classified correctly using the new feature set and the proposed neural network classifier. It is noted that the majority of the examples use documents with special PLA difficulties that are not covered by known methods.

Fig. 13 presents the results obtained by applying the

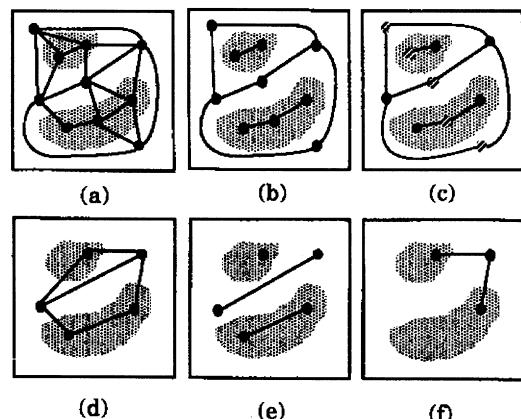


Fig. 14. Misclassified blocks.

proposed method to mixed type documents in which misclassification appeared in some blocks by applying the previous method. The document of Fig. 13 has a size of  $1689 \times 2200$  pixels and a resolution of 300 dpi and is taken from the University of Washington database. On the top-left corner of this document, we can observe the blocks, which were shown in Fig. 14. These blocks are classified as non-text while they were misclassified using the first feature set.

Fig. 15(a)–(c) shows selected mixed type documents with different types of segmentation difficulties. These documents were scanned by an HP ScanJet IIc scanner at 150 dpi resolution. All the documents in Fig. 15 contain text with fonts of different type, style and size, and graphics that cannot be separated with vertical and horizontal lines. In addition to normal text blocks, Fig. 15(c) also contains

	PC/MS-DOS	Windows	Macintosh
Entertainment	26.8%	142.2%	25.5%
Home education	37.3%	**	63.9%
Finance	16.8%	224.6%	30.8%
Word processors	-43.7%	65.0%	56.0%
Spreadsheets	-18.7%	64.7%	-5.4%
Databases	-0.2%	968.5%	20.1%
Integrated	-21.7%	158.2%	92.6%
Utilities	-25.6%	358.3%	46.3%
Presentation graphics	-37.9%	183.3%	39.7%

Source: the Software Publishers Association \*in North America

Fig. 15. Instances where segmentation is difficult.

## (b) English across the curriculum

**Biology: the muscles in your body**

**Use a dictionary to help you check the meanings of these words. Then use the words to complete the text.**

ankles  
arm  
body  
chest  
face  
neck  
stomach  
thigh  
toes

**DID YOU KNOW that**  
you have more than six hundred and twenty muscles in your **body**? What do all those muscles do?

The answer is: they do a lot! For example, the muscles in your (1) ..... help you to talk, smile and eat. You use about fourteen of them when you smile! You use your (2) ..... muscles to help you control the movements of your head.

In the upper part of your body, the muscles in your (3) ..... are important for breathing. Your (4) ..... muscles allow you to lift and carry objects. And your (5) ..... muscles help you to digest food.

In the lower part of your body, the (6) ..... muscles just below your waist help you to move your legs. The calf muscles at the back of your legs control your (7) ..... and your feet.

And did you know that you've got about twenty muscles in each foot? You've even got muscles in your two big (8) .....

**Match the instructions with the pictures.**

Sit on a chair and put your hand on the scales.

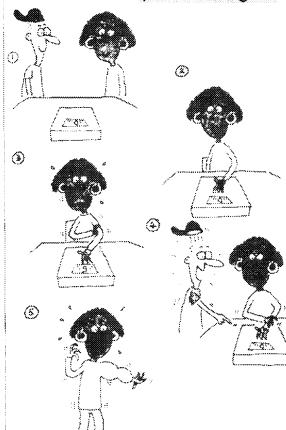
Read the result. How much pressure does your arm produce?

Put some bathroom scales on a table.

For bigger and stronger arms, do this experiment regularly.

Press your hand down on the scales and feel your triceps.

**How to test your strength**



??? Did you know? ???



The word 'muscle' comes from the Latin word for mouse. Bend your arm and look at the biceps. You can see something under your skin. It's like a mouse in a bag!

Fig. 15. (Continued)

tilted and handwritten text blocks. It can be observed that we have accurate text identification results for all tested documents.

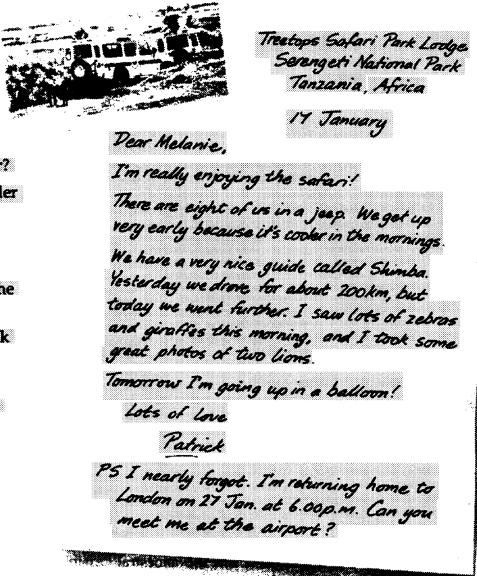
## 7. Conclusions

Segmentation is an important issue in the automated document analysis area. Text segmentation plays a significant role in document retrieval and storage systems. This paper proposes a new method that identifies the regions of a mixed type document in text or non-text areas. The method belongs to the bottom-up segmentation techniques and in its first stage is a block extraction procedure. Generally, the extracted blocks contain text, graphics, and halftones. To identify the type of each block, texture

features are extracted. These features are then processed by a PCA and classified by a neural network classifier. A feature selection scheme is employed in the definition of the final feature set, which results in only 34 effective features. The PCA optimally decreases the feature space dimensionality. The final stage consists of a Kohonen SOFM classifier, the competition layer of which consists of an  $8 \times 8$  grid. Although we are interested in identifying only the text blocks, the classification scheme classifies efficiently all the documents' blocks as text, graphics, and halftone blocks. The combination of the texture feature set with PCA and the neural network classifier confronts many segmentation difficulties. In comparison with other similar segmentation techniques, the proposed method works well even with documents that do not satisfy the Manhattan layout. It is independent of the size and type of characters

## (c) Skills work

- On safari**  
**Read the letter and answer the questions below.**
- 1 What is the address of the sender?
  - 2 What is the date of the letter?
  - 3 What greeting does the sender use?
  - 4 What is the message of the letter?
  - 5 How does the writer close the letter? (*Lots of ...*)
  - 6 Who is the letter from? Look at the signature.
  - 7 What is the postscript (PS) about? What did the writer forget to say?



**Discuss this advice with your teacher:**

- GOOD WRITING STRATEGIES**
- Make notes and write a list of words about the topic.
  - Arrange your notes into sentences and paragraphs.
  - Check your work.
  - Learn from your mistakes!

**3 Imagine you are on an unusual holiday. Write a letter to an English-speaking friend about it.**

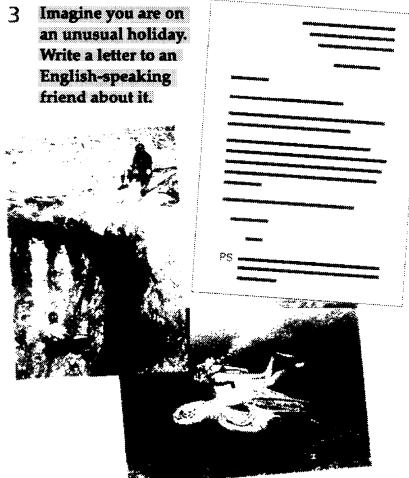


Fig. 15. (Continued)

and the position of the text in the document and it is insensitive to a small tilt. Also, the proposed method can recognize text areas even if these are included in graphics or halftone regions. The proposed segmentation method was extensively tested with many mixed type documents presenting significant difficulties for segmentation. Experimental results presented in this paper show the feasibility and the robustness of the method in segmenting of mixed type documents.

## References

- [1] L. O'Gorman, R. Kasturi, Document Image Analysis, IEEE Computer Society Press, Los Alamos, California, 1995.
- [2] H. Fujisawa, Y. Nakano, K. Kurino, Segmentation methods for character recognition: from segmentation to document structure analysis, Proceedings of IEEE (July 1992) 1079–1092.
- [3] J. Schurmann, N. Bartneck, T. Bayer, J. Franke, E. Mandler, M. Oberlander, Document analysis—from pixels to contents, Proceedings of IEEE (July 1992) 1101–1119.
- [4] T. Pavlidis, J. Zhou, Page segmentation and classification, Graphical Models and Image Processing 54 (1992) 484–496.
- [5] K.Y. Wong, R.G. Casey, M. Wahl, Document analysis system, IBM J. Res. Dev. 6 (1982) 647–656.
- [6] F.M. Wahl, K.Y. Wong, R.G. Casey, Block segmentation and text extraction in mixed text/image documents, Computer Graphics and Image Processing 20 (1989) 375–390.
- [7] D. Wang, S.N. Shihari, Classification of newspaper image blocks using texture analysis, Computer Vision Graphics and Image Processing 47 (1989) 327–352.
- [8] G. Nagy, S. Seth, M. Viswanathan, A prototype document image analysis system for technical journals, IEEE Comput. 25 (1992) 10–22.
- [9] L.A. Fletcher, R. Kasturi, A Robust algorithm for text string separation from mixed text/images, IEEE Trans. Pattern Anal. Machine Intell. 10 (1988) 910–918.
- [10] R. Kasturi, S.T. Bow, W. El-Masri, J. Shah, J.R. Gattiker, U.B. Mokate, A system for interpretation of line drawings, IEEE Trans. Pattern Anal. Machine Intell. 12 (1990) 978–991.

- [11] L. O'Gorman, The document spectrum for page layout analysis, *IEEE Trans. Pattern Anal. Machine Intell.* 15 (1993) 1162–1173.
- [12] K.-C. Fan, C.-H. Liu, Y.-K. Wang, Segmentation and classification of mixed text/graphics/image documents, *Pattern Recognition Letters* 15 (1994) 1201–1209.
- [13] F. Farrokhinia, Multi-channel filtering techniques for texture segmentation and surface quality inspection, Ph.D. thesis, Dept. of Electrical Engineering, Michigan State University, 1990.
- [14] A. Jain, S. Bhattacharjee, Text segmentation using Gabor filters for automatic document processing, *Machine Vision and Applications* 5 (1992) 169–184.
- [15] A.K. Jain, Yu Zhong, Page segmentation using texture analysis, *Pattern Recognition* 29 (1996) 743–770.
- [16] C. Strouthopoulos, N. Papamarkos, C. Chamzas, Identification of text-only areas in mixed type documents, *IEEE Workshop on Non-linear Signal and Image Processing* 1 (1995) 162–165.
- [17] C. Strouthopoulos, N. Papamarkos, C. Chamzas, Identification of text-only areas in mixed type documents, *Engineering Applications of Artificial Intelligence* 10 (4) (1997) 387–401.
- [18] A.S. Pandya, R.B. Macy, *Pattern Recognition with Neural Networks in C + +*, CRC Press and IEEE Press, 1995, pp. 195–211.
- [19] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks* 12 (1989) 459–473.
- [20] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan, New York, 1994.
- [21] J. Dayhoff, *Neural Network Architectures: An Introduction*, Van Nostrand Reinhold, New York, 1990.
- [22] UW: English Document Image Database, University of Washington, Seattle, 1993.
- [23] D.M. Tsai, Y.H. Chen, A fast histogram-clustering approach for multilevel thresholding, *Pattern Recognition Letters* 13 (1992) 245–252.
- [24] N. Papamarkos, B. Gatos, A new approach for multilevel threshold selection, *CVGIP: Graphical Models and Image Processing* 56 (1994) 357–370.
- [25] I. Witten, A. Moffat, T. Bell, *Managing Gigabytes*, Van Nostrand Reinhold, New York, 1994.
- [26] M. Driels, D. Nolan, Automatic defect classification of printed wiring board solder joints, *IEEE Trans. on Components, Hybrids and Manufactury Technology* 13 (1990) 331–340.