PERGAMON

Engineering Applications of Artificial Intelligence 12 (1999) 119–138

Engineering Applications of
ARTIFICIAL
INTELLIGENCE

PLA using RLSA and a neural network

C. Strouthopoulos, N. Papamarkos *, C. Chamzas

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

Received 1 December 1997; accepted 1 October 1998

Abstract

This paper describes a new method for document page layout analysis. The proposed approach is based on the use of the run-length smoothing algorithm (RLSA) and a neural network block classified (NNBC). The RLSA is used locally and globally for the block segmentation by using optimal pre-estimated smoothing values. The NNBC is used in the classification steps of the method as a tool which classifies the blocks of the document into basic classes or subclasses. The NNBC consists of a principal component analyzer (PCA) and a self-organized feature map (SOFM). The input feature vector is a set of features corresponding to the contents and the relationships of 3×3 masks. This set is selected by using a statistical selection procedure, and provides textural information. In the final step, and after the application of a grouping procedure, the document blocks are classified as text frames and isolated text lines, graphics and halftones, or into secondary subclasses corresponding to special cases of the basic classes. The proposed method can identify blocks that cannot be separated with horizontal and vertical cuts, and gives very correct classification even on documents of bad scanning quality. The performance of the method has been extensively tested on a variety of documents. Several examples illustrate the strength and the effectiveness of the methodology.

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

1. Introduction

The use of paper documents is a common method of information transfer and communication. The conversion of paper documents into a proper electronic form is essential for their processing, understanding, archiving and transmission by computers. An important procedure in the 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 the meaning associated with the positional and functional blocks in which the text is located (Chauvet et al., 1992; Fujisawa et al., 1992; Schurmann et al., 1992; Witten et al., 1994).

Structural layout analysis (SLA) and functional layout analysis (FLA) are the two types of PLA. SLA (also called physical and geometric layout analysis) (O'Gorman and Kasturi, 1995) is performed to obtain a physical segmentation of a document into groups of document components. To achieve SLA, the document must be separated into meaningful blocks, which include data having common and certain homogeneous attributes. As a result, SLA labels the structural blocks of a document as text, halftones or line drawings. FLA (also called syntactic and logical layout analysis) (O'Gorman and Kasturi, 1995), uses domain-dependent information consisting of layout rules of a particular page to perform the labeling of the structural blocks, giving some indication of the function of the block. This functional labeling may also entail the splitting or merging of structural blocks. It is based on document-specific rules using features derived from the document, such as the relative and absolute positions of a block on a page, the relative and absolute positions of a field within a block, general typesetting rules of spacing, and the font size and style of the text. As an example, the result of functional labeling would indicate the title, author block, abstract, keywords, paragraphs of the text body, and so on.

The PLA of mixed-type documents is a prerequisite to facilitating later processes such as the recognition of

* Corresponding author. Tel.: +30-541-79585; fax: +30-541-79569; email: papamark@voreas.ee.duth.gr.

0952-1976/99/$ - see front matter
PII: S0952-1976(98)00053-0

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. Generally, document understanding, archiving, storage and acquisition systems require the PLA as a preprocessing step.

There are many techniques proposed for the PLA. Most of them are concerned with the structural page layout analysis, mainly the textual block identification. Due to the large variation of formats and some common image-processing problems, erroneous feature detection and imperfect text recognition results, most of the work undertaken in functional labeling has been restricted to a particular domain, e.g. postal recognition (O'Gorman and Kasturi, 1995).

A PLA method consists of two main steps. The first step is the segmentation of the document into visually distinct blocks, and the second step is the block classification. The proposed techniques for document segmentation can be classified as top-down (or model-driven) (Pavlidis and Zhou, 1992; Wong et al. 1982; Wahl et al., 1989; Wang and Shihari, 1989; Nagy et al., 1992), bottom-up (or data-driven) (Fletcher and Kasturi, 1988; Kasturi et al., 1990; O'Gorman, 1993), or hybrid (Fan et al., 1994; Farrokhinia, 1990; Jain and Bhattacharjee, 1992; Jain and Zhong, 1996). In top-down methods, a document is first split into major regions (large components), and major regions are split into smaller subregions (more-detailed components), and so on. Bottom-up methods involve the grouping of pixels as connected components (marks), and the merging of these components into successively larger regions. Hybrid techniques are those that use local and global strategies in combination with texture analysis approaches. In this category belong methods that use texture-analysis in the frequency domain, such as the Gabor filters.

A basic advantage of top-down methods is that they perform PLA fast. However, the top-down methods are applicable when the blocks are orthogonal, when their size is greater than or equal to the paragraph size, and when they can be separated with horizontal and vertical cuts. So, for documents where figures are intermixed or overlapped with the text, these methods may be inappropriate. The bottom-up techniques are more appropriate for these formats, with the disadvantage that they are usually more expensive in terms of computing time.

For document segmentation, most of the top-down techniques are based on the RLSA, and on the recursive projection profiles cuts technique (Wong et al., 1982; Wahl et al., 1989; Wang and Shihari, 1989). Most of the bottom-up methods start by extracting the document marks and including them in bounding boxes. Next, these boxes are connected into larger blocks, according to their relative frequency of occurrence, dimensions and positions. It must be noted that there is no single, general method that typifies all bottom-up techniques (O'Gorman and Kasturi, 1995). Therefore, RLSA can be used as a bottom-up technique if appropriate small values can be assigned to the smoothing parameters. Making use of this, Fan et al. (1993) detect the text lines of documents as stripes, and propose a procedure for stripe grouping into paragraph blocks.

Until now, two approaches have been proposed for PLA by Strouthopoulos et al. (1997) and Strouthopoulos and Papamarkos (1997). The first one (Strouthopoulos et al., 1997) proposes a method for the 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 bounding-box connection algorithm. Next, these rectangles are classified as either text or non-text areas by using thirteen 3×3 masks called document structural elements (DSE). This method gives satisfactory segmentation results for documents of good scanning quality. 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 kinds of halftones. The second approach (Strouthopoulos and Papamarkos, 1997) performs a PLA by using spatial structural features in combination with a neural-network classifier. The set of features consists of 34 features, which in addition of the use of DSE, combines the contents of neighbouring pairs of 3×3 masks. These features are next processed by a neural network that consists of a PCA and a Kohonen SOFM (Pandya and Macy, 1995; Sanger, 1989; Haykin, 1994; Dayhoff, 1990). The output of the SOFM is an 8×8 competition layer that clusters the block types. This method performs well if the document scanning quality is good. Unfortunately, for low-quality documents, the mark-extraction procedure fails, and this leads to an inappropriate segmentation.

To overcome the segmentation problems associated with documents of low quality, a new method is proposed here, which uses the RLSA as a bottom-up procedure in combination with an NNBC system. In low-quality documents, RLSA performs well, and can identify successfully blocks of handwritten text or halftones composed of multi-size and type dots. However, RLSA fails to separate nonorthogonal blocks that are intermixed with graphics or halftones. The reason for this drawback is the use of global smoothing values that are not

suitable for the local characteristics of the documents. In this approach, the RLSA is used as a bottom-up procedure with pre-estimated local smoothing values. Both the pre-estimated local smoothing values and the block classification are based on the iterative use of an NNBC, which has

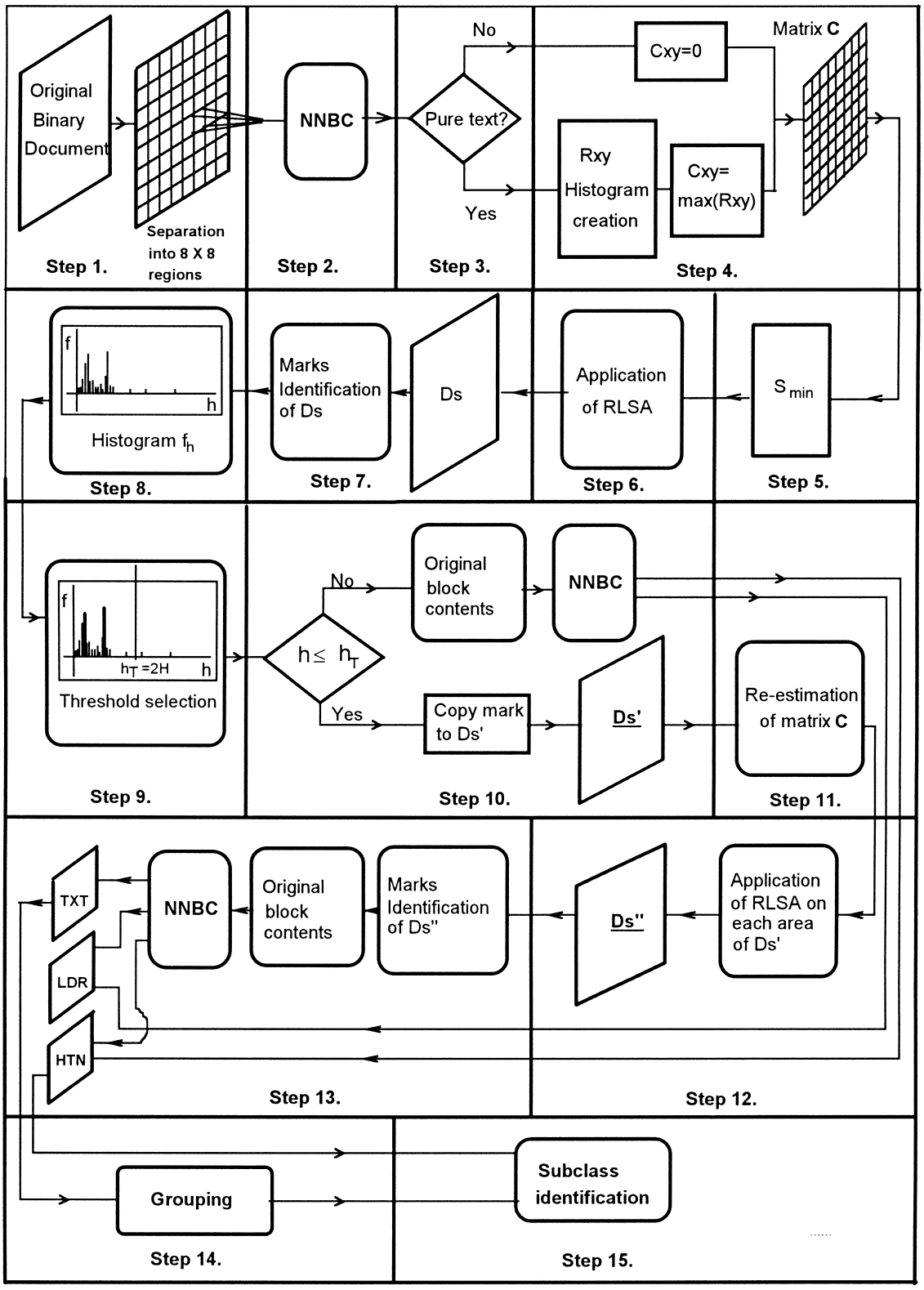


Fig. 1. The steps of the method.

as input the textural features described in Strouthopoulos and Papamarkos (1997). In addition, the proposed method has three advantages. The first is the identification of not only the test blocks, but also the halftones and graphics blocks. The second advantage is gained by the use of a grouping technique for the final PLA of the document. Finally, the method classifies the document blocks

(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)	1×1	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 :

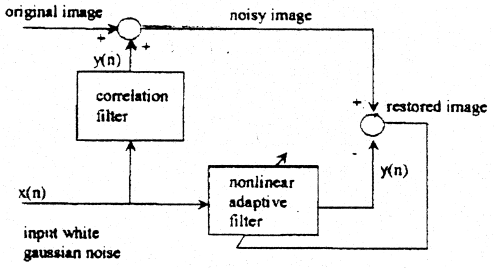


figure 3 : the adaptive noise cancellor scheme

The correlation filter is a nonlinear filter given by :

$$y(n) = 0.6x(n)+03x(n-1)+0.001x^2(n)$$
$$+0.0005x(n)x(n-1)+0.003x^2(n-1)$$

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=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. 2. The original test document.

not only into the basic classes, but also into secondary subclasses corresponding to the writing style, the font type and the halftoning techniques.

Briefly, the basic stages of the method are:

Separation of the document into 8×8 regions, identification of text regions using NNBC and, for each

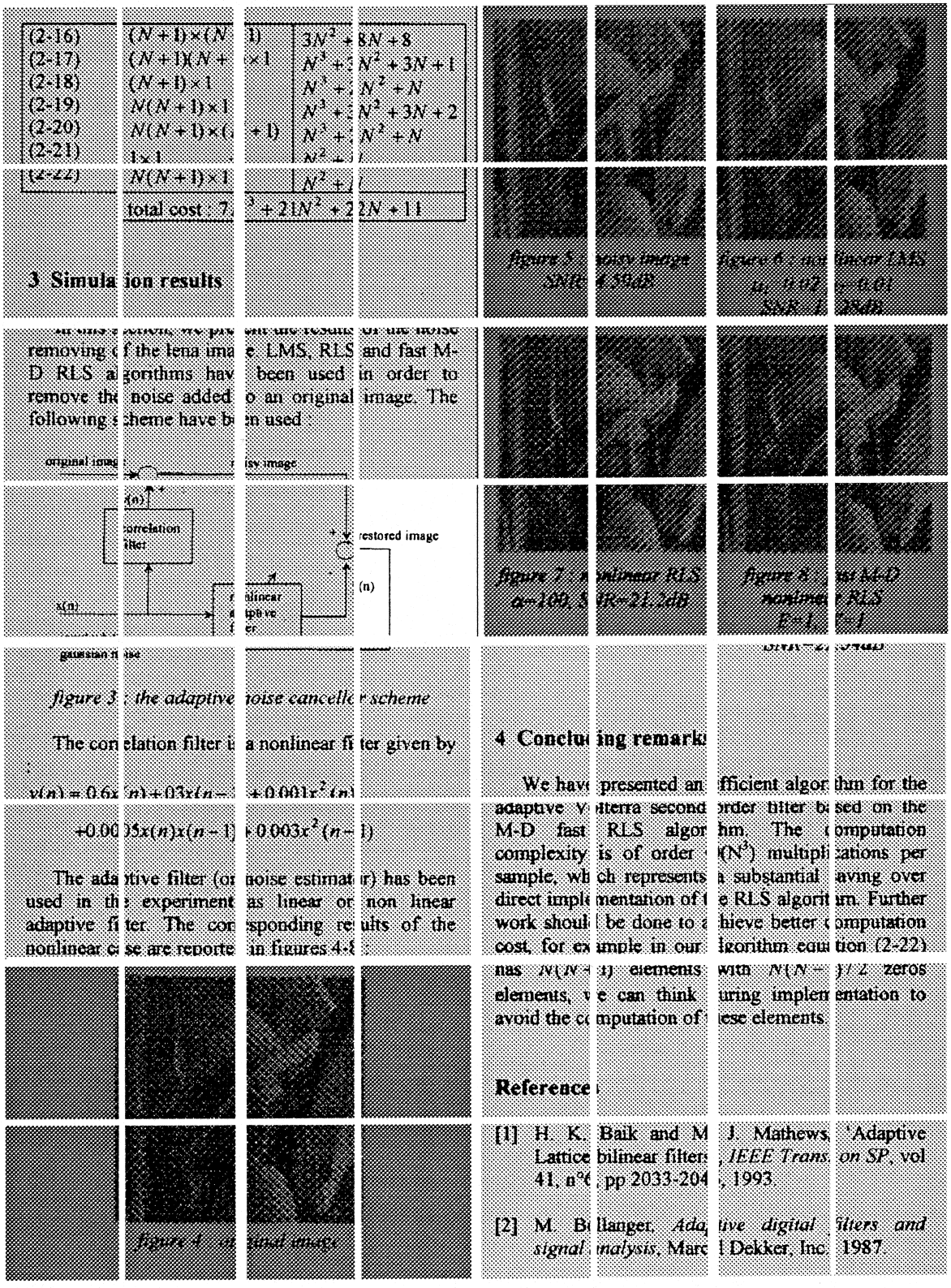


Fig. 3. The results obtained after Step 2.

region, the estimation of the local values of the horizontal mean character distance.

Global application of the RLSA, using as smoothing values the smallest of the mean character distances.

Identification of high marks.

Classification of high marks by using the NNBC.

Local application of RLSA in the remaining marks by using local smoothing values.

[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]	1×1	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 :

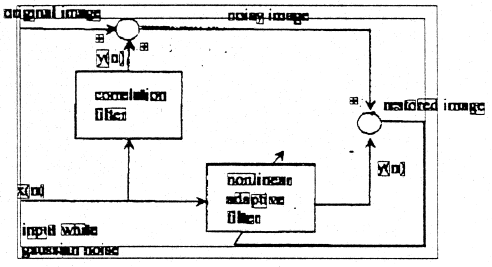


figure 3 : the adaptive noise cancellor scheme

The correlation filter is a nonlinear filter given by :

$$y[n] = 0.6x[n] + 0.3x[n-1] + 0.001x^2[n] + 0.0005x[n]x[n-1] + 0.0003x^2[n-1]$$

The adaptive filter (on noise estimation) 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=12.28dB

figure 7 : nonlinear RLS $\alpha=1.00$ 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.

Fig. 4. The marks of image D_s and the corresponding bounding boxes.

Classification of blocks and identification of text lines, halftones and graphics blocks by using the NNBC.

Grouping of text lines into text frames.

Classification of text frames and halftone blocks into secondary subclasses.

The method was tested with many documents. Many of the test documents are from the UW document image database (UW, 1993). This paper presents characteristic examples that cover special types of documents. The experimental results shown in this paper confirm the effectiveness of the proposed method.

2. Description of the method

The processed document must be in binary form. At the end of this procedure, three images are produced, named TXT, LDR and HTN, which contain text, graphics and halftones, respectively. Fig. 1 describes the steps of the method. In detail, the steps of the entire method are described next. The results of each step are referred to the document of Fig. 2.

Step 1. The prototype binary document is divided into $K \times K$ orthogonal regions. For documents with resolution greater than 100 dpi, an appropriate value for K is $K = 8$. Each region is named Axy, with x, $y \in \{0, 1, \ldots, K-1\}$.

Step 2. Each Axy region is examined by the NNBC and characterized as a possible region of text, graphics or halftones. An analytical description of the NNBC is given in a subsequent section. Fig. 3 shows the results obtained by the application of this procedure to the document of Fig. 2.

Step 3. It is possible, due to the arbitrary division of the document, for some text Axy regions to contain not only text. For this reason, each Axy is characterized as:

- a pure text region, when it contains text and does not have neighbouring graphics or halftone regions in the horizontal and vertical directions;
- a non-pure text region in any other cases.

Step 4: For every pure text region, the mean value C_{xy} of the horizontal character distances is estimated. To do this, the histogram R_{xy} of the horizontal white pixel runs is determined. The maximum histogram value corresponds to the most frequent white run between the pixels of the characters (O'Gorman and Kasturi, 1995). The value of C_{xy} corresponds to maximum R_{xy}. For nonpure text regions, C_{xy} cannot be determined in this way because the dots of halftones and the lines of graphics distort the histogram. For these regions, the C_{xy} values are temporarily set to zero. The final values of C_{xy} are estimated with an algorithm described in Step 11.

Step 5. The minimum value $S_{\min} = \text{minimum}\{C_{xy} / C_{xy} > 0\}$ is calculated. In order to merge the dots of halftone regions without merging the text with regions of other types, $S_{\min}$ is taken as an appropriate small value that is suitable for the global application of RLSA. In this example, the value of $S_{\min}$ has been calculated to be equal to 3.

Step 6. The RLSA is now applied with horizontal and vertical smoothing values equal to $S_{\min}$. The same value for horizontal and vertical smoothing is assigned because the regions of halftones have similar dot distances. The result of the RLSA is a new binary image, D_s, consisting of groups of connected pixels (marks). It is noted that mainly the adjacent dots of halftones, the broken lines of graphics and some characters of small size have been merged into marks. The large marks consist of dot groups from the halftone regions, or from graphic blocks. The small marks are created from smoothed characters of the document text regions. Therefore, the small marks are much greater in number than the large marks.

Step 7. The marks of image D_s are separated by the application of a bounding-box technique (Pratt, 1991), which surrounds each mark with a bounding box. Fig. 4 shows the marks of image D_s, and the corresponding bounding boxes. For weeding out marks that are clearly not textual, the following procedure is applied, as described in steps 8, 9 and 10.

Step 8. The histogram $f(h)$ of the bounding box heights is created. On the left of the histogram local peaks, appear due to the short and multitudinous boxes. Each peak corresponds to:

- the character marks of any different text type used in the document;
- short marks of graphics or halftones;
- some noise pixels.

The small values in the right of the histogram are due to the few and high bounding boxes. These values come from:

- large halftone regions;
- large graphical regions such as tables and picture borders.

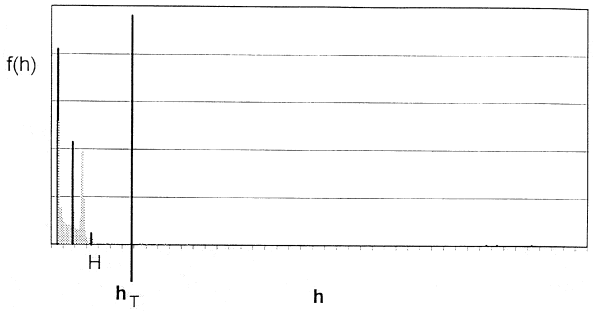
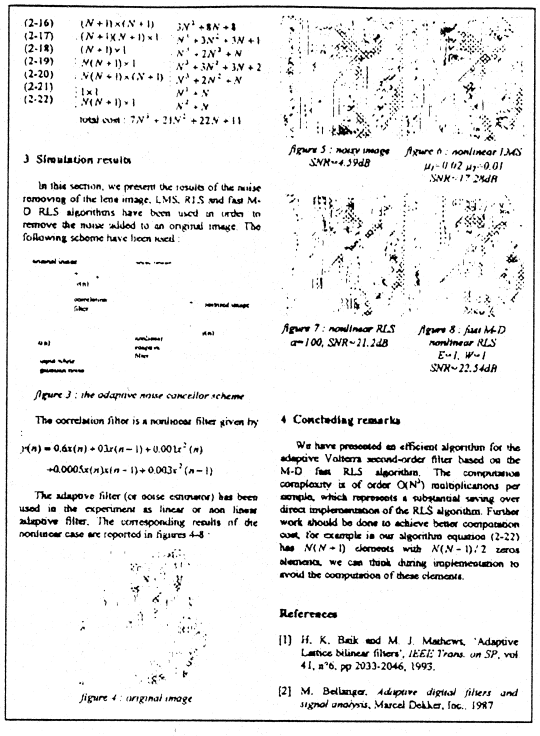


Fig. 5. The threshold value h_T.

Step 9. Each histogram value $f(h)$ is examined to see if it is the maximum peak in the range $[h/2, 2h]$. According to O'Gorman and Kasturi (1995), each text string must contain characters of similar size (a range less than two to one in point size). Thus, it can be easily seen how the coefficients 1/2 and 2 have been



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

$$y(n) = 0.6x(n) + 0.3x(n-1) + 0.001x^2(n) + 0.0005x(n)x(n-1) + 0.003x^2(n-1)$$

The adaptive filter (or noise canceller) 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=4.39dB

figure 6 : nonlinear LMS μ_1=0.02 μ_2=0.01 SNR=17.29dB

figure 7 : nonlinear RLS α=100, SNR=21.2dB

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

4 Concluding remarks

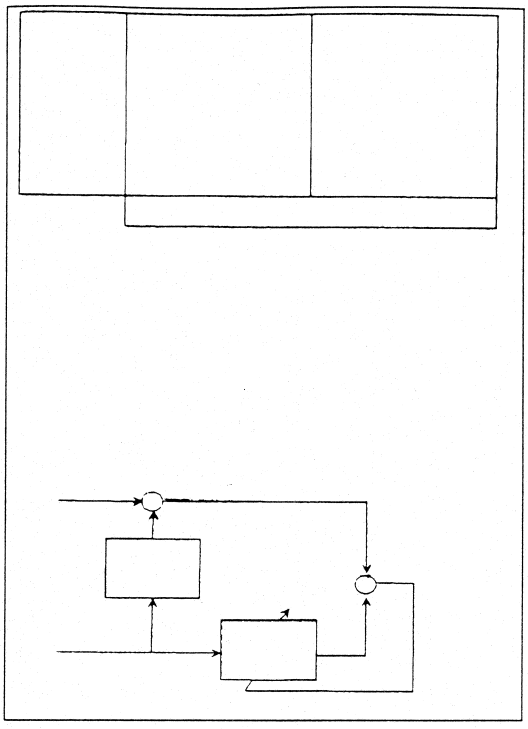
We have presented an efficient algorithm for the adaptive Volterra second-order filter based on the M-D fast RLS algorithm. The computation complexity is of order $O(N^2)$ multiplications per sample, which represents a substantial saving over direct implementation of the RLS algorithm. Further work should 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

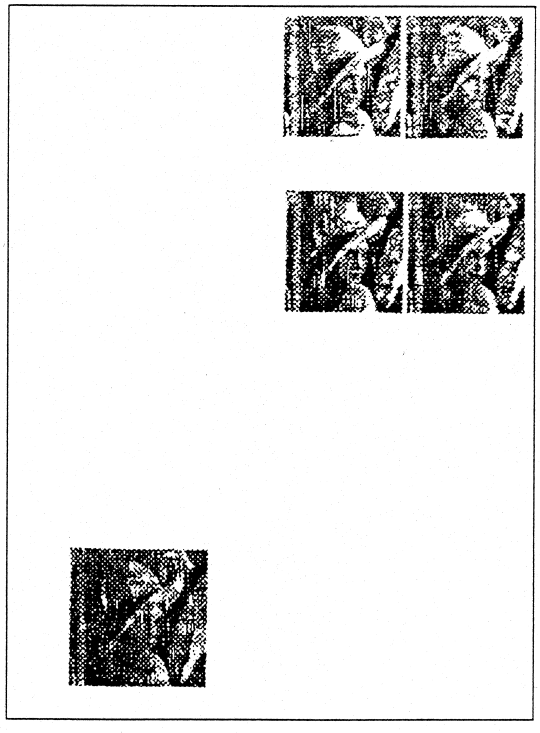
[1] H. K. Baik and M. J. Mathews, 'Adaptive Lattice bilinear filters', *IEEE Trans. on SP*, vol 41, n°6, pp 2033-2046, 1993.

[2] M. Bellanger, *Adaptive digital filters and signal analysis*, Marcel Dekker, Inc., 1987

(a)



(b)



(C)

Fig. 6. (a) The D_s' image. (b) LDR image. (c) The HTN image.

selected. Similar coefficients are used in Kasturi et al. (1990). The far-right histogram peak H corresponds to the highest document characters, or to graphics and halftone marks, or to a combination of these. The histogram values in the right side of H come mainly from graphics and halftone marks. A threshold value

$$h_T = 2H \tag{1}$$

is suitable for the discrimination of marks as short (if $h \leqslant h_T$) and high (if $h > h_T$). Fig. 5 shows the peaks and the threshold value for the document of Fig. 2.

Step 10. The content of the original document for every high mark examined by the NNBC is classified as halftone or graphics, and is stored to LDR or to HTN. The remaining short marks are driven to a new image D_s'. The decomposition of the original document into D_s', LDR and HTN images is shown in Fig. 6.

Step 11. After the above step, the segmentation of text lines is achieved by the use of horizontal run-length smoothing. Due to the different sizes of the characters, the horizontal RLSA is applied separately on each of the D_s' regions. For each region, the smoothing values are locally calculated from C_{xy} values. Specifically, as in the original image D_0, the image D_s' is divided into the same 8×8 orthogonal regions A_{xy}'. In each region, a suitable horizontal smoothing value S_{xy} is determined from the C_{xy} values. In the beginning, for each pure text region, $S_{xy} = C_{xy}$. Next for each nonpure text region, the horizontal smoothing value is calculated from the relation:

$$S_{xy} = \begin{cases} C_{x_0y} + \frac{C_{x_1y} - C_{x_0y}}{x_1 - x_0}(x - x_0), \forall x \in (x_0, x_1), x_0 \neq x_1 \\ C_{x_0y} \forall x \in [0, x_0), x_0 > 0 \\ C_{x_1y} \forall x \in (x_1, 7], x_1 < 7 \end{cases} \tag{2}$$

where A_{x_0y}, A_{x_1y} are the pure text regions nearest to A_{xy} with $x_0 \leqslant x_1$. In the case where there is no pure text region in a line, the values of S_{xy} are calculated by a similar procedure (linear interpolation) from the smoothing values in the vertical direction. Fig. 7 gives an example of S_{xy} determination. It is obvious that even only one pure text region is enough for the calculation of all the smoothing values. It is noted that S_{xy} gives a measure of the mean horizontal character distance in each region.

Step 12. The horizontal smoothing value must be large enough in order to accomplish the merging not only of characters, but also of the words of the text lines. For this reason, a multiple value of S_{xy} must be taken. It is found that a value of $8\,S_{xy}$ is suitable for the horizontal run-length smoothing of each region. The value of 8 for the smoothing coefficient is a result of the measurement tests for the inter-word distances. Image D_s'' contains the results of the RLSA application, and is shown in Fig. 8.

Step 13. The D_s'' image consists mainly of word marks. However, it is possible to have some short halftone and graphics marks. The content of the original document for each mark of D_s'' is examined by the NNBC, and is classified as text, graphics or halftones, and is stored in TXT, LDR and HTN images. After this step, the LDR and HTN images have ben completed with the rest of the components of the graphics and halftones of the original document.

Fig. 9 shows the decomposition results.

Step 14. The blocks of TXT images are grouped into text frames using a text block grouping procedure that will be described in a subsequent section.

Step 15. Classification of blocks into secondary subclasses is done according to the writing style, the character font type and the halftoning techniques. The identification of subclasses is based on the use of the NNBC in the blocks derived from Step 14.

0	4	5	0	0	2	0	0	C_{xy} values
4	4	5	4	3	2	2	2	S_{xy} values

Fig. 7. Example of S_{xy} determination.

3. The neural-network block classifier

A basic tool in this approach is the use of the NNBC, which classifies a document block into a proper class of text, graphics or halftones. The NNBC consists of a PCA and an SOFM in a cascade connection. The spatial texture features of the blocks used in (Strouthopoulos and Papamarkos, 1997) are taken as the input feature set of the NNBC. The output of this neural-network system is a 8×8 competition layer, giving the correct classification result for each block type. The form of the NNBC system is given in Fig. 10, and is described below.

3.1. First set of features

Features are used to express the spatial information of the blocks, and can describe what types of 3×3 masks are included in the blocks. The first set of features corresponds to 13 structural 3×3 binary masks, which are called document structure elements (DSE). The order of pixels in the mask is as follows:

$$\begin{matrix} b_8 & b_7 & b_6 \\ b_5 & b_4 & b_3 \\ b_2 & b_1 & b_0 \end{matrix}$$

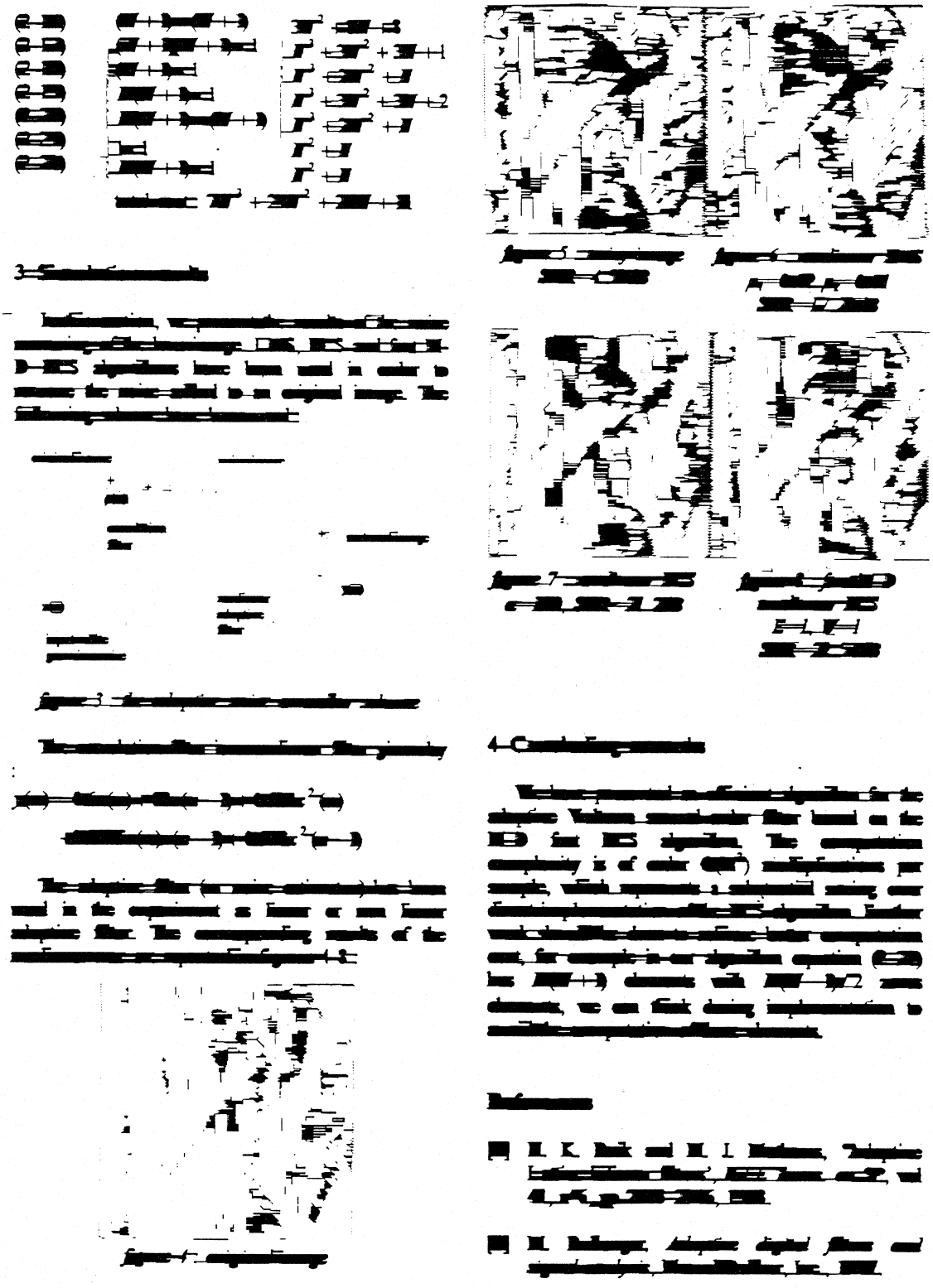


Fig. 8. The image D_s''.

An integer $L = \Sigma_{i=0}^{8} b_i 2^i$ with $b_i \in \{0, 1\}$ is assigned to any DSE, and is called the document structure element characteristic number. Since $L \in \{0, 1, 2, \ldots, 511\}$, 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

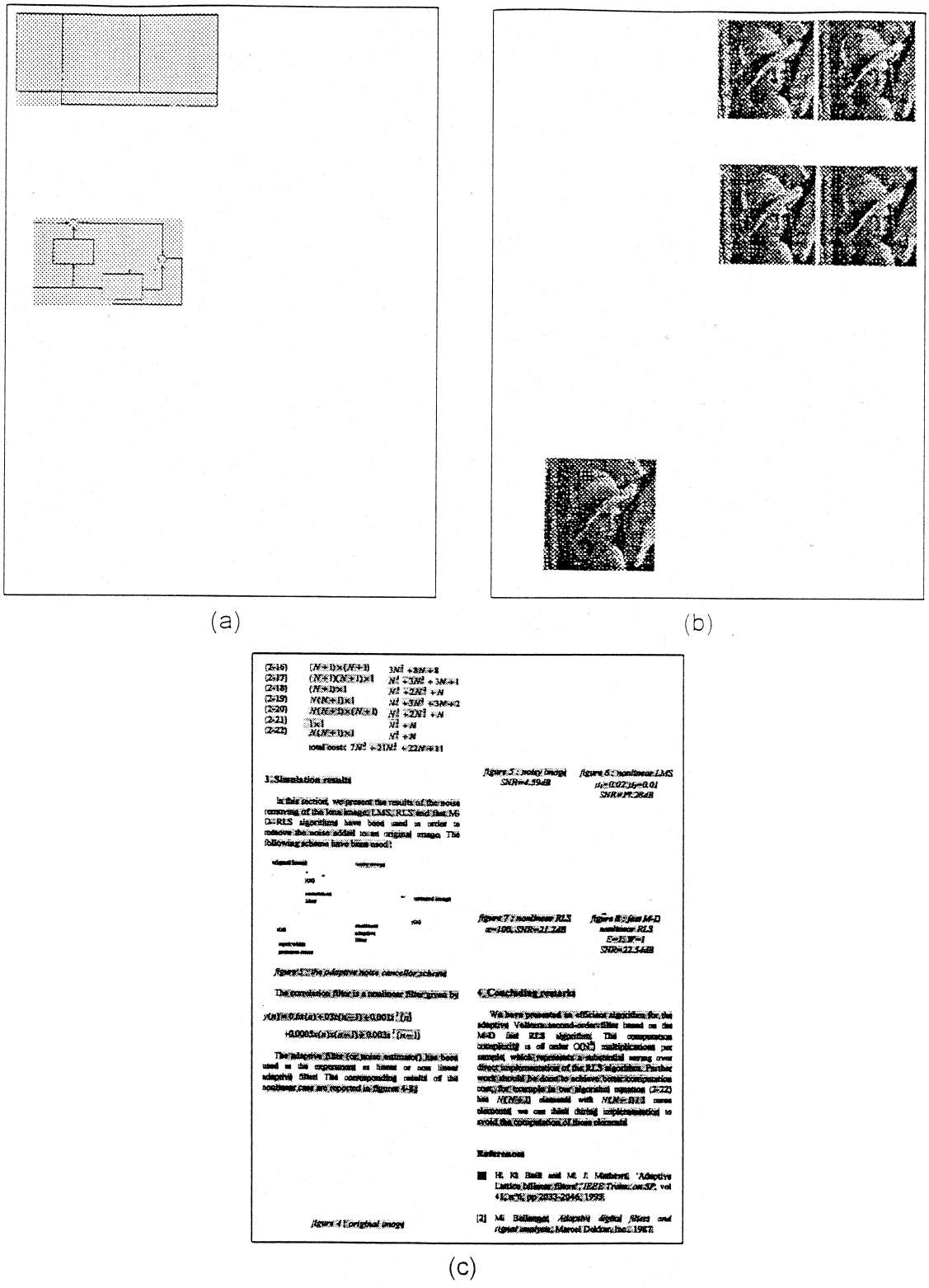


Fig. 9. (a) The LDR image. (b) The HTN image. (c) The TXT image.

because they correspond to pure background and object regions, respectively) in the block. If $h_1(l)$ is the histogram function then the probability density function $H_1(l)$ is given by the relation:

$$H_1(l) = \frac{h_1(l)}{\sum_{l=1}^{510}}, \quad l \in \{1, 2, \ldots, 510\} \tag{3}$$

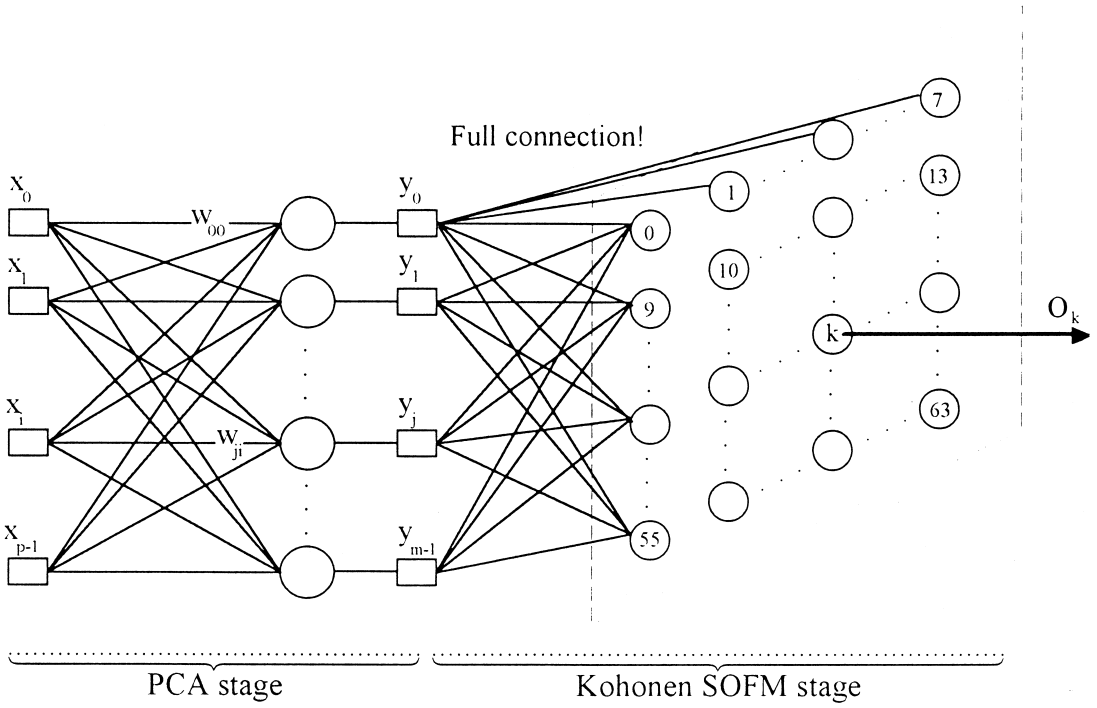


Fig. 10. The text frames.

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

3.2. The second set of features

The second set of features is used in order to overcome difficult block classification cases associated with some kinds of large line drawings of halftones. This happens because the first set of features does not describe well the morphology of the graphics in the blocks. To overcome these difficulties, additional features are used, which are extracted by combining the information of pairs of neighbor DSEs. Specifically, every DSE is examined in a pairing with each one of the eight neighbor DSEs. Doing this, a two-dimensional histogram is formulated that gives the contribution of each pair of DSEs in the block.

The 2-D histogram function $h_2(l_1, l_2)$ of a KXJ block A is obtained using the relation

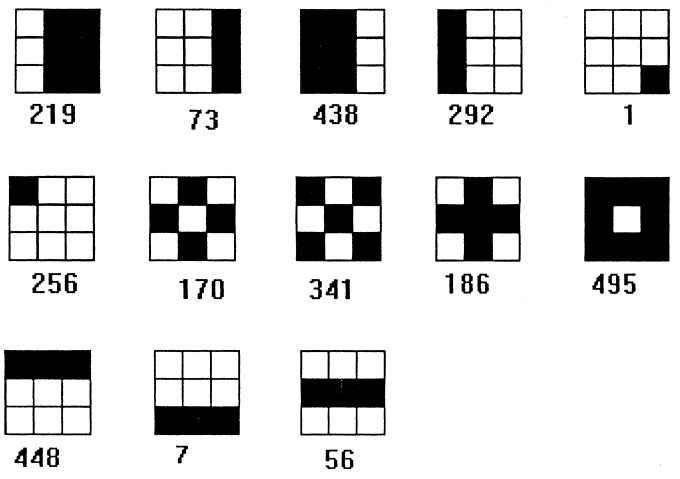
$$h_2(l_1, l_2) = \begin{cases} h_2 l_1, l_2) + 1, & \text{if} \quad l_1 = L_1(k, j) \quad \text{and} \quad l_2 = L_2(k', j') \\ h_2(l_1, l_2), \text{otherwise} \end{cases}$$

where $l_1 \in \{1,2,\ldots,510\}$, $l_2 \in \{1,2,\ldots,510\}$, $k \in \{4,5,\ldots,K-5\}$, $j \in \{4,5,\ldots,J-5\}$, $k' = k + a$, $j' = j + b$, (a, b) $\in \{(-3, 0), (-3, -3), (0, -3), (3, -3), (3, 0), (3, 3), (0, 3), (-3, 3)\}$, $L_1(k, j)$ is the characteristic number of a DSE in the position j, k and $L_2(k', j')$ is the characteristic number of each neighbor DSE in the position (k', j'). Using the probability density function, one 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

An important process in recognition systems is the selection of the smaller set of appropriate features from a much bigger set. Selection of 'good' features is critical to the performance of recognition and classification. In this approach, the selection is based on stability, separability and similarity criteria, and follows the procedure described in (Driels and Nolan, 1990). The entire procedure is performed on a large number of testing blocks, extracted from different types of documents. The application of the above feature-reduction process, for the first feature set, results in the only 13 features, corresponding to the DSEs given in Table 1, and to only 21 features for the second feature set shown in Table 2. Thus, the final feature set consists of only 34 texture features.

Table 1
The first set of 13 DSE features



3.4. The Principal Component Analyzer

The first part of the neural-network classifier is a PCA which achieves not only an increasing in feature variation but also a decrease in the dimensionality of the feature space. Briefly, in this approach the input vector of the PCA is the 34-feature vector $\mathbf{x}$ while the output is an eight-dimensional vector $\mathbf{y}$ which is given by the relation:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{5}$$

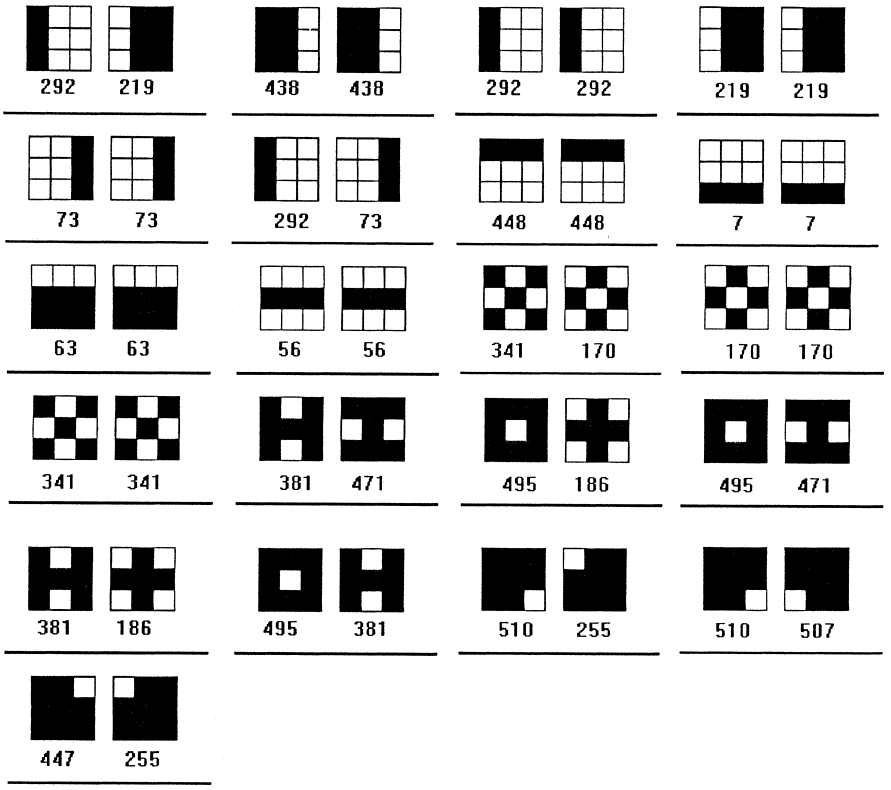
where the transformation matrix $\mathbf{W}$ contains the neural-network coefficients.

The neural network is trained by the generalized hebbian algorithm, which is an unsupervised learning algorithm. It calculates the eigenvectors of the covariance matrix $\mathbf{C}$ of the training feature set, and it has been proved to converge 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. A complete analysis of the PCA method used in this paper is given in Strouthopoulos and Papamarkos (1997).

3.5. The SOFM

The second part of the neural-network classifier is a Kohonen self organized feature map (SOFM). The

Table 2
The second set of features



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. In this approach the competition layer has an 8×8 grid.

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. Similar patterns are represented by neighbor neurons. 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. In the entire PLA method the NNBC is used initially in steps 2, 10 and 13. For each one of these steps the NNBC is trained separately with wide and representative sets of blocks for each step. The goal of the NNBC for these steps is analyzed above, and is associated with the identification of only the three basic classes of text, graphics and halftones.

In the final stage of the method, the NNBC is used to extract additional information that will be useful for FLA. This procedure is based on the nature of an NNBC, which can identify not only the basic block classes but also block subclasses that are not obvious in the beginning. Specifically, there are some neurons, each of them exclusively representing one pattern type, which can be considered as a subclass of one of the basic classes. That is, each basic class, which consists of a number of neurons, includes a number of subclasses that are mapped into the 8×8 grid competition layer. This result is important for many applications, such as the OCR techniques. From experience it has been found that the system can effectively identify subclasses of:

- different font types;
- different writing styles;
- white noise halftoning; and
- clustering-dot ordered dither.

4. Grouping of text lines

Image TXT consists of only text lines. Graphics and halftones are stored in LDR and HTN images. next, working only on TXT images, the text lines are grouped as text frames (O'Gorman and Kasturi, 1995). The grouping process is a merging procedure that uses the physical properties of text lines, specifically their height and position.

Each text line is bounded by a rectangle $R(l_R, r_R, t_R, b_R)$, defined by the coordinates (l_R, t_R), (r_R, t_R), (l_R, b_R) and (r_R, b_R) of its corners C_i^R, $I = 1, \ldots, 4$.

Let $U(l_U, r_U, t_U, b_U)$ and $V(l_V, r_V, t_V, b_V)$ be two other text line rectangles. The 16 distances between the corners of these rectangles are given by the relation

$$D_m = \mid C_i^U - C_j^V \mid, \quad \text{where } m = 4(i-1)+j \quad \text{and} \quad j = 1, \ldots, 4 \tag{6}$$

Also, the distances between the sides of the rectangles are defined as

If $l_U < l_V < r_U$ then $D_{17} = \mid t_U - b_V \mid$, $D_{18} = \mid b_U - t_V \mid$, otherwise D_{17} and D_{18} are not defined.

If $l_U < r_V < r_U$ then $D_{19} = \mid t_U - b_V \mid$, $D_{20} = \mid b_U - t_V \mid$, otherwise D_{19} and D_{20} are not defined.

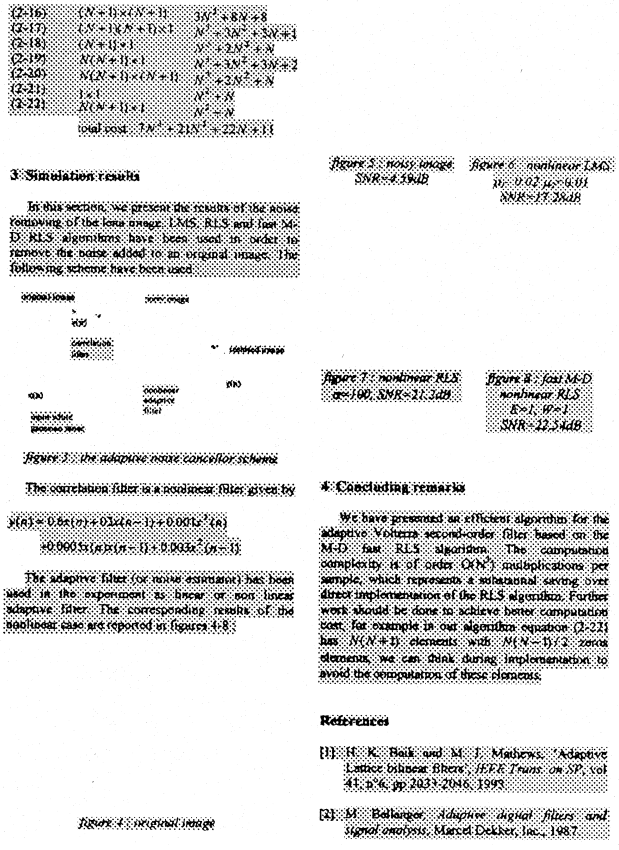


Fig. 11. The text frames.

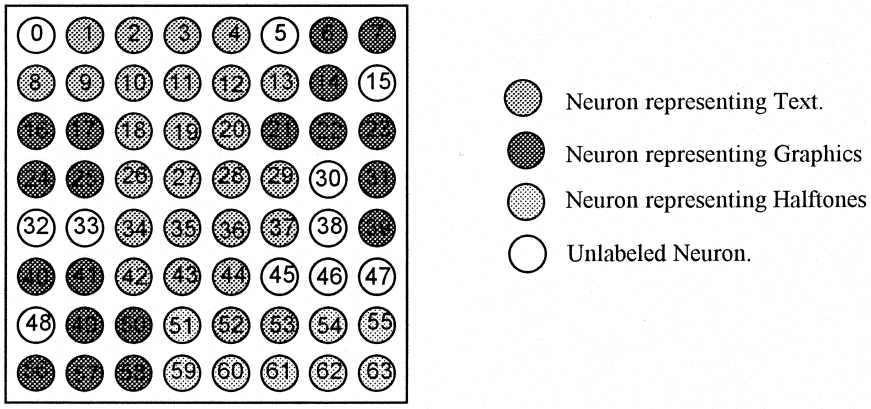


Fig. 12. Kohonen SOFM after training.

If $t_U < t_V < b_U$ then $D_{21} = | l_V - r_U |, D_{22} = | r_V - l_U |$, otherwise D_{21} and D_{22} are not defined.

If $t_U < b_V < b_U$ then $D_{23} = | l_V - r_U |, D_{24} = | r_V - l_U |$, otherwise D_{23} and D_{24} are not defined.

If $\Delta = \{D_m / D_m \in \mathcal{R}\}$ then the distance d between the rectangles U, V is defined as

$$\mathrm{d}(U, V) = \mathrm{minimum}(\Delta) \tag{7}$$

Now, the text line V is grouped with the text frame of text line U, if and only if the following conditions are satisfied:

$$0.5h_U \leq h_V \leq 2h_U \tag{8}$$

$$\mathrm{d}(U, V) \leq 2h_U \tag{9}$$

where h_U and h_V the heights of the rectangles. A complete analysis of this procedure is given in Appendix A. Fig. 11 shows the final text-block grouping result for the document of Fig. 2.

Table 3
Block recognition rate for Step 13

	Blocks from the training set	Blocks not in the training set
Text	99.5%	98.5%
Graphics	99.8%	98.6%
Halftones	100.0%	99.5%

5. Experimental results

5.1. NNBC training experimental results

The NNBC was trained with different training sets composed from the blocks which are processed by the NNBC in each step of the method. Thus, at the end of the training procedure, four NNBC have been constructed. Each of them is suitable for a specific step of the method. The training of the SOFM follows the training of the PCA. The blocks were derived from representative document types, having resolutions in the range 100–300 dpi. These documents contain text blocks of characters of different sizes and font types, graphic blocks with different thickness, and blocks of images displayed with different halftoning techniques. The evaluation of the NNBC was done by using two sets of experiments. The first set was obtained from documents belonging to the training set, while the other set was obtained from documents from outside the training set. As can be observed in Table 3, the block recognition rate of the NNBC was high for each step. The smallest recognition rates refer to Step 13, where the NNBC processes small blocks. The highest recognition rates are observed in the final and most important step, where the NNBC classifies the large blocks, not only

Table 4
Subclasses appearing in the competition layer

Neuron	Subclass
52, 53	Text of italics characters
2, 3, 4	Text of arial font characters
11, 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

into the basic classes but also into useful subclasses. 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 basic classes.

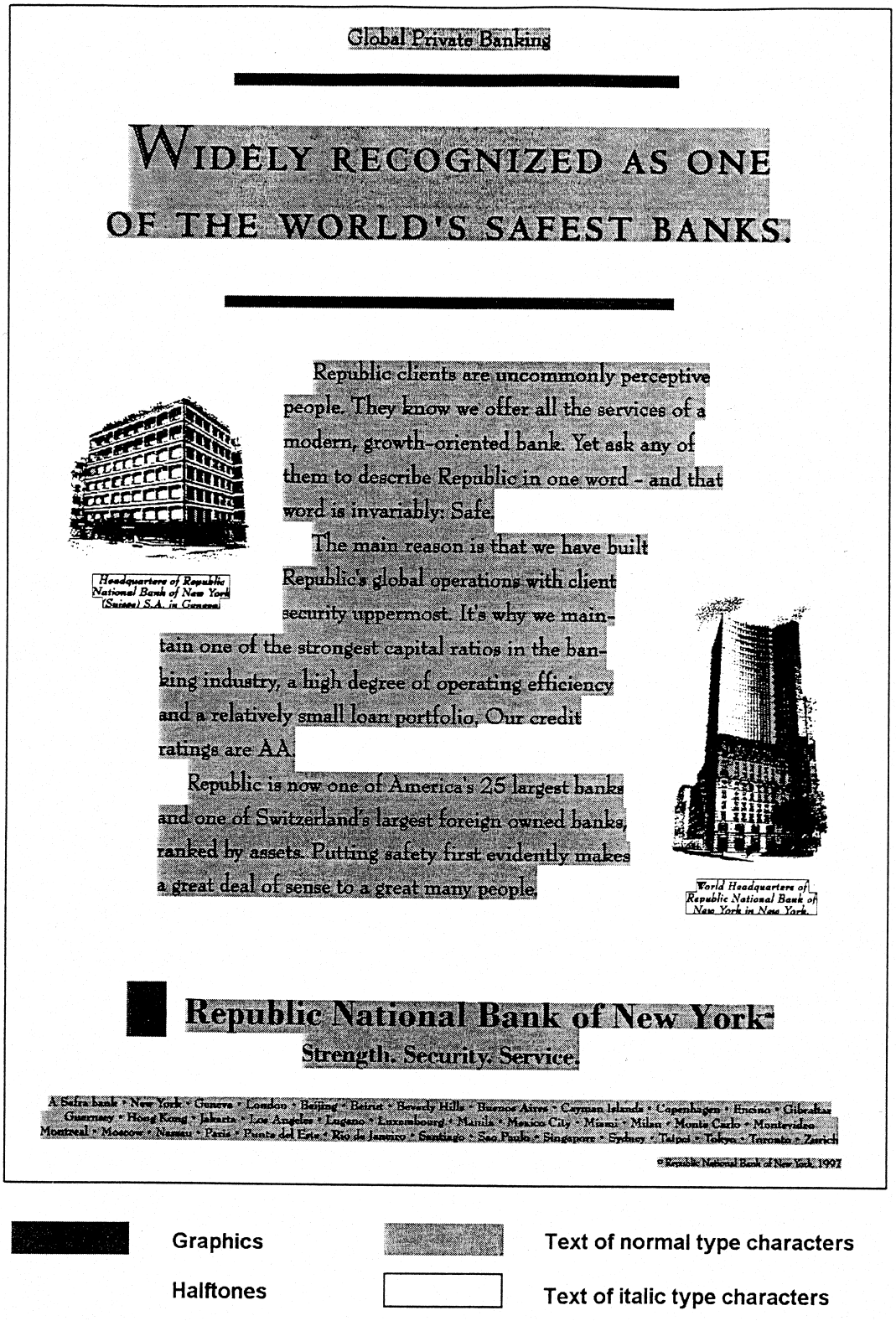


Fig. 13. Document of example 1.

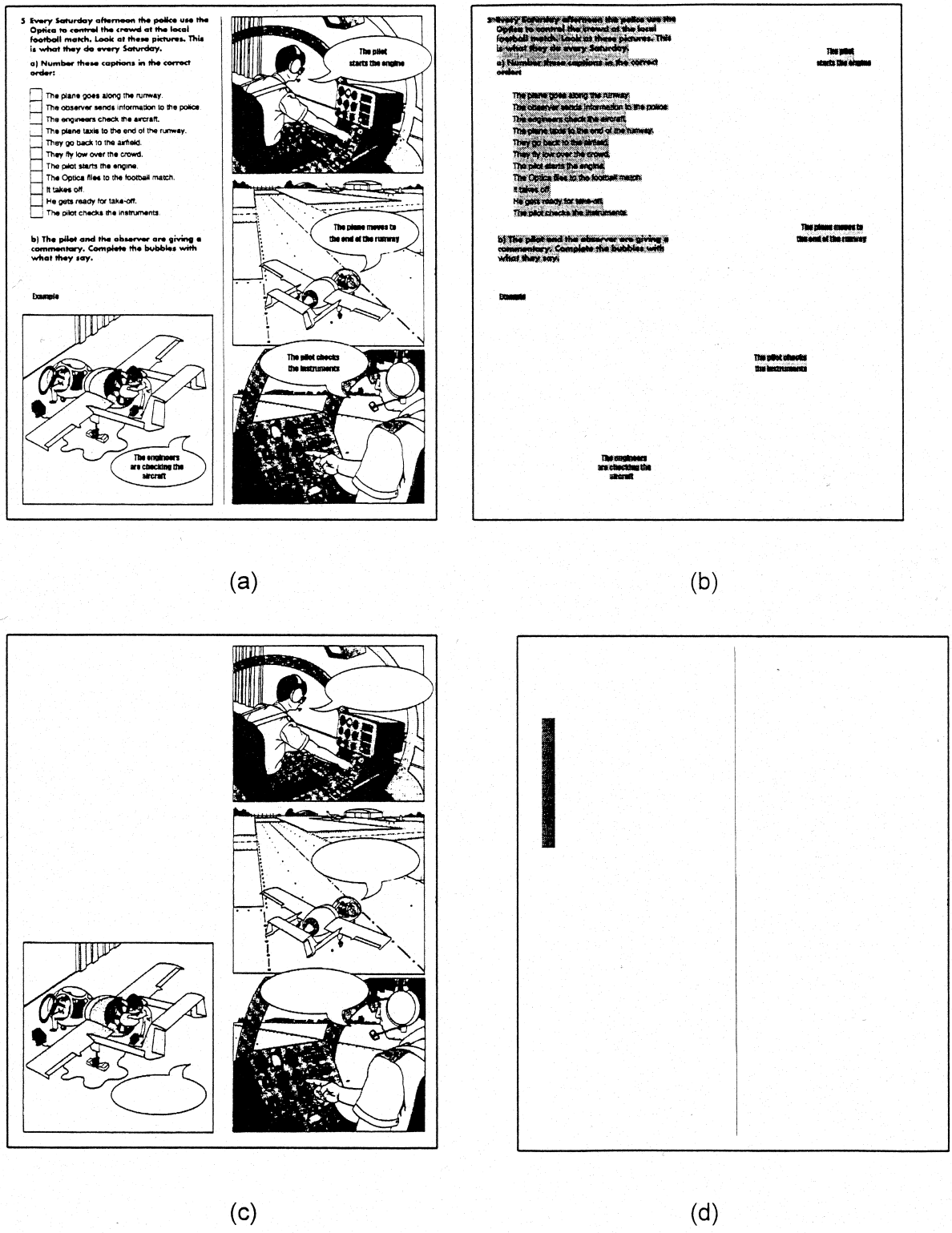


Fig. 14. (a) The original document. (b) Text frames of the document. (c) Halftone blocks of the document. (d) Graphics blocks of the document.

In the final use of the NNBC, subclasses can be identified. Specifically, in the competition layer there are some neurons, each of them representing one pattern type exclusively which can be considered as a subclass of one of the basic classes. The observation of subclasses is made after the test procedure by taking into account the type of the classified blocks. Such neurons are mentioned in Table 4.

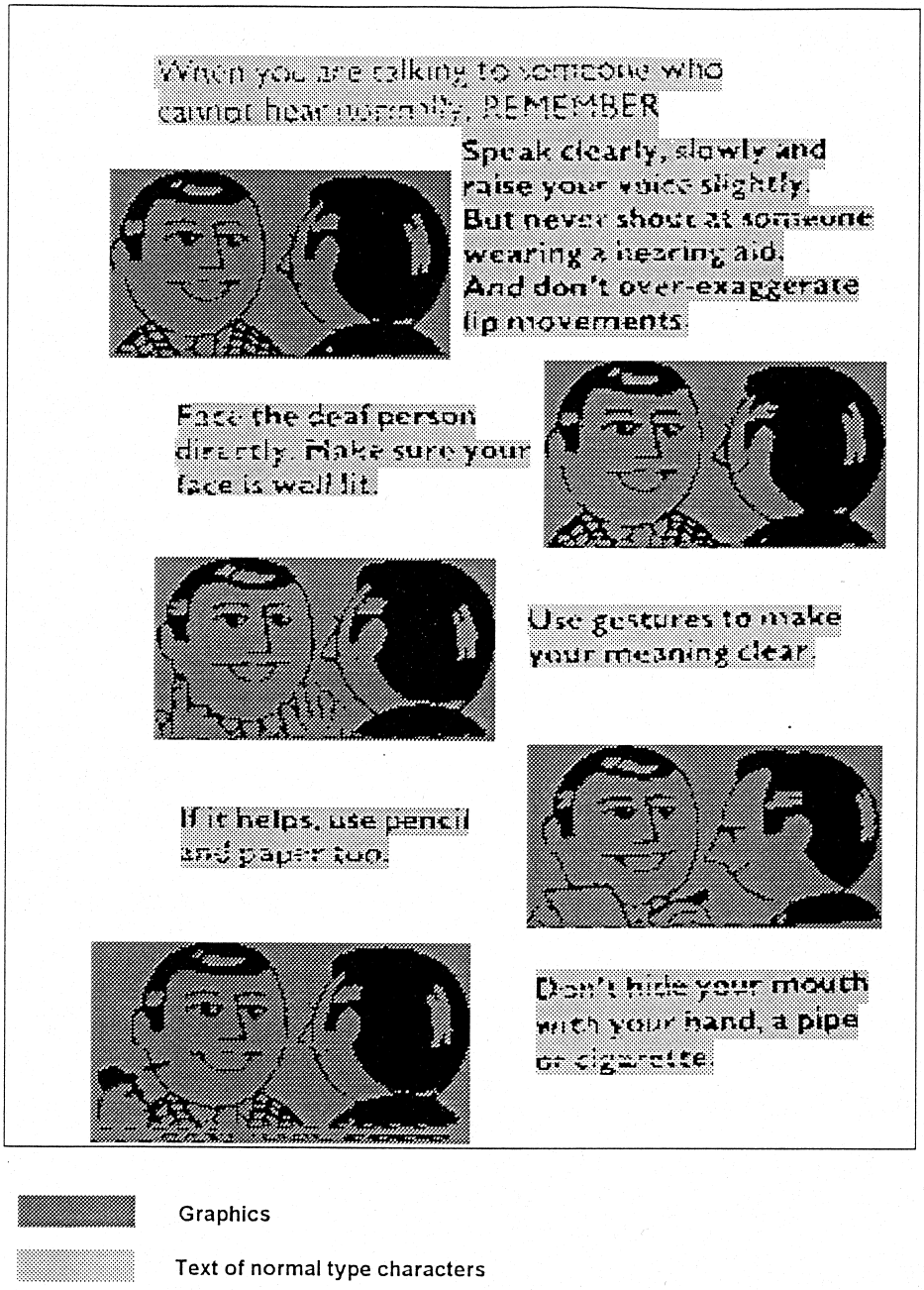


Fig. 15. Document of example 3.

5.2. Experimental PLA results

The proposed method was extensively tested on a variety of mixed-type documents. Due to the space limitation, only three additional examples are given here. The documents in Figs 13–15 are selected mixed-type documents with different types of segmentation difficulties. These documents were scanned at 200, 300 and 100 dpi resolution, respectively. The test documents contain text blocks of characters of different sizes and font types, graphics blocks, and blocks of halftones.

The halftones and the fonts of the first document do not belong to the training set. Figure captions consist of Italic type characters. The blocks of the document, including the Italic-type blocks, are identified correctly by the application of the proposed method.

In the document of Fig. 14, there are four pictures containing text blocks written with Arial type characters. This type of font is part of the entire font set used in the training procedure. The application of the method gave correct identification results. It is noted that, as was expected, the blocks of Arial type

characters are enabled by the neuron 3 of the competition layer.

In the final example, the proposed method is applied on a document of low scanning quality and resolution. Although this document has not been used in the training stage, the identification results were correct, as is shown in Fig. 15.

6. Conclusions

In this paper, a new method for PLA of mixed-type documents is proposed. The method is based on the use of RLSA and an NNBC. RLSA is used locally and globally to perform segmentation of document blocks. For better segmentation results, and to identify the document text lines, a pre-estimation technique is used, which gives the proper local smoothing values of the RLSA. Thus, the method combines the advantages of a bottom-up technique and the RLSA, giving correct results even with low-quality documents that do not satisfy the Manhattan layout (O'Gorman and Kasturi, 1995). The NNBC, which is used in four steps of the method, is a powerful procedure that can classify the document blocks efficiently. It consists of a PCA and an SOFM. The input of the NNBC is a set of textural features, and the output is an 8×8 grid competition layer. In the final use of NNBC the neurons of the competition layer correspond not only to the three basic classes of text, graphics and halftones, but also to subclasses. Thus, the method can identify subclasses of text font types and images of different halftoning techniques. This important result can be used in many applications, such as in OCR and document-retrieval systems.

For each step, the NNBC was trained separately, using a wide and representative set of blocks. The proposed method was extensively tested with many mixed-type documents, having significant difficulties for segmentation. many of these documents are from known databases. In all the test documents the method gave satisfactory results that show its feasibility and robustness. The entire technique is implemented in C++, and the average computing time for an A4 document is about 35 s in a Pentium-200 computer.

Appendix A

Let image TXT consist of N bounding boxes containing text lines that must be grouped into text frames. Each group is labeled by an integer. Assume also that each box b_n has height h_n and is characterized by an integer g_n, which is equal to the label of its group. When $g_n = 0$, it is considered that box b_n has not yet been grouped by the grouping procedure. When $g_n > 0$, it is considered that box b_n belongs to group g_n.

Initializations: $g_n = 0\ \forall n = 1, 2, \ldots, N$. Define the set $S = \{b_n / b_n \text{ with } g_n = 0\}$, and let $f = 1$ and k be the repetition counter.

Step 1. Set $k = 1$. The set $G_f^k = \{b_m / b_m$ the first element of $S\}$, $m \in \{1, 2, \ldots, N\}$ is defined. Set $g_m = f$ and for this reason $b_m \notin S$.

Step 2. Set $l = k + 1$ and the set G_f^l is defined by the relation

$$G_f^l = \{b_j / b_j \in S \quad \text{and} \quad b_m \in G_f^k : \quad D(b_m, b_j) < 2h_m \quad \text{and} \quad 0.5 < h_j < 2h_m\}$$

For each $b_j \in G_f^l$ we set $g_j = f$. For this reason $b_j \notin S$. Set $k = l$. Step 2 is repeated until $G_f^l = \emptyset$. The elements of the set $G_f = \cup_{l=1}^{k} G_f^l$ are the boxes of group f.

Step 3. The label f is increased by one and we continue from the step 1 until $S = \emptyset$.

References

Chauvet, P., Krahe, J., Taflin, E., Maitre, H., 1992. System for an intelligent office document analysis, recognition and description. Signal Processing 32, 161–190.

Dayhoff, J., 1990. Neural Network Architectures: an Introduction. Van Nostrand Reinhold, New York.

Driels, M., Nolan, D., 1990. Automatic defect classification of printed wiring board solder joints. IEEE Transactions on Components, Hybrids and Manufactory Technology 13, 331–340.

Fan, K.C., Liu, C.H., Wang, Y.K., 1994. Segmentation and classification of mixed text/graphics/image documents. Pattern Recognition letters 15, 1201–1209.

Farrokhinia, F., 1990. Multi-channel filtering techniques for texture segmentation and surface quality inspection. Ph.D. thesis, Department of Electrical Engineering, Michigan State University.

Fletcher, L.A., Kasturi, R., 1988. A Robust algorithm for text string separation from mixed text/graphics images. IEEE Transaction Pattern Analysis and Machine Intelligence 10, 910–918.

Fujisawa, H., Nakano, Y., Kurino, K., 1992. Segmentation methods for character recognition: from segmentation to document structure analysis. Proceedings of IEEE 80, 1079–1092.

Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan, New York.

Jain, A., Bhattacharjee, S., 1992. Text segmentation using Gabor filters for automatic document processing. Machine Vision and Applications 5, 169–184.

Jain, A., Zhong, Y., 1996. Page segmentation using texture analysis. Pattern Recognition 29, 743–770.

Kasturi, R., Bow, S., El-Masri, W., Shah, J., Gattiker, J.R., Mokate, U.B., 1990. A system for interpretation of line drawings. IEEE Transaction on Pattern Analysis Machine Intelligence 12, 978–991.

Nagy, G., Seth, S., Viswanathan, M., 1992. A prototype document image analysis system for technical journals. IEEE Transactions on Computers 25, 10–22.

O'Gorman, L., 1993. The document spectrum for page layout analysis. IEEE Transactions on Pattern Analysis Machine Intelligence 15, 1162–1173.

O'Gorman, L., Kasturi, R., 1995. Document Image Analysis. IEEE Computer Society Press.

Pandya, A., Macy, R., 1995. Pattern Recognition with Neural Networks in C + + . CRC press and IEEE Press, pp. 195–211.

Pavlidis, T., Zhou, J., 1992. Page segmentation and classification. Graphical models and Image Processing 54, 484–496.

Pratt, W.K., 1991. Digital Image Processing. 2nd ed.. Wiley, New York.

Sanger, T.D., 1989. Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Networks 12, 459–473.

Schurmann, J., Bartneck, N., Bayer, T., Franke, J., Mandler, E., Oberlander, M., 1992. Document analysis—from pixels to contents. Proceedings of IEEE 80, 1101–1119.

Strouthopoulos, C., Papamarkos, N., 1997. Text identification for document image analysis using a neural network. In: 13th International Conference on Digital Signal Processing—DSP97, Santorini, Greece, pp. 999–1002.

Strouthopoulos, C., Papamarkos, N., Chamzas, C., 1997. Identification of text-only areas in mixed type documents. Engineering Applications of Artificial Intelligence 10, 387–401.

UW, 1993. English Document Image Database. University of Washington, Seattle.

Wahl, F.M., Wong, K.Y., Casey, R.G., 1989. Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing 20, 375–390.

Wang, D., Shihari, S.N., 1989. Classification of newspaper image blocks using texture analysis. Computer Vision Graphics and Image processing 47, 327–352.

Witten, I., Moffat, A., Bell, T., 1994. Managing Gigabytes. Van Nostrand Reinhold, Amsterdam.

Wong, K., Casey, R.G., Wahl, M., 1982. Document analysis system. IBM Journal of Research and Development 6, 647–656.