COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 47, 327–352 (1989)

Classification of Newspaper Image Blocks Using Texture Analysis*

DACHENG WANG AND SARGUR N. SRIHARI

Department of Computer Science, State University of New York at Buffalo, Buffalo, New York 14260

Received October 27, 1988; revised January 27, 1989

An important step in the analysis of images of printed documents is the classification of segmented blocks into categories such as half-tone photographs, text with large letters, text with small letters, line drawings, etc. In this paper, a method to classify blocks segmented from newspaper images is described. It is assumed that homogeneous rectangular blocks are first segmented out of the image using methods such as run-length smoothing and recursive horizontal/vertical cuts. The classification approach is based on statistical textural features and feature space decision techniques. Two matrices, whose elements are frequency counts of black–white pair run lengths and black–white–black combination run lengths, are used to derive texture information. Three features are extracted from the matrices to determine a feature space in which block classification is accomplished using linear discriminant functions. Experimental results using different block segmentation results, different newspapers, and different image resolutions are given. Performance and speed with different image resolutions are indicated.

1. INTRODUCTION

Documents are the most common medium of information transmission in society. Understanding messages conveyed by printed documents such as newspapers is a common intelligent activity of humans. Making a computer perform the task of analyzing and understanding a newspaper image is a challenging one. The Document Image Understanding group at SUNY/Buffalo is involved in research whose long-term goal is to develop a newspaper understanding system [1].

Newspaper images are usually of a poorer image quality than most printed documents. A newspaper page is also a complex document containing several static visual information representation forms, such as textual, pictorial, and their many combinations. Structurally, a newspaper page consists of rectangular blocks of varying size and content. The basic content categories are: titles, paragraphs, line drawings, photographs, and mixed graphics and text. A newspaper understanding system needs to be able to accomplish the following tasks for a given newspaper page:

A. *Image Digitization and Binarization*

In this step the newspaper page is digitized into a grey-scale image using a CCD camera or other appropriate means. The first processing step is to convert the grey-scale image into a binary-valued image. This is an appropriate as well as necessary step. Appropriate, since documents are printed as dark dots on a light background or *vice versa*. Necessary, since the subsequent segmentation, classification, and character recognition steps require a binary image. The image can be

*This work was supported by the National Science Foundation Grant IRI-86-13361.

0734-189X/89 $3.00

binarized using a global or adaptive thresholding scheme. If color is to be handled then three grey-scale images corresponding to red, green, and blue filters are used.

B. *Block Segmentation*

This is the first step in determining the layout structure of a newspaper page from its binary image. At this stage the image is partitioned into several, usually rectangular, regions each of which is referred to as a block. The partitioning is based on local and global visual properties of regions. The segmentation process is usually much more straightforward than in the case of a natural scene due to the fact that documents are printed as rectangular blocks with linear spaces in between.

C. *Block Classification (or Labeling)*

A preliminary label is given to each block in this stage. Typically, classification is into a few basic classes: halftones (which usually correspond to photographs, and called as such because of the illusion of shades of grey in between black and white), text with large letters (with point size above 32 and usually correspond to headlines), text with medium-sized letters (with point size 14 to 32 and correspond to subtitles), text with small letters (with point size below 14 and correspond to the main body of the text), and graphics (which are predominantly line drawings with some text). In order to accomplish the classification it is necessary to first extract appropriate features from each block. The features are used to classify a block using either a feature space partitioning technique or by means of production rules.

D. *Analysis and Understanding*

Steps A to C may be referred to as primary image analysis, whose objective is to derive segmented and labeled physical blocks. The next, or secondary, stage of image analysis is to find out interrelated blocks and merge them into logical units such as articles with their titles, photographs with their captions, etc. Then for each unit, higher level analysis is attempted by techniques such as character and word recognition, natural language understanding, and scene analysis. These processes usually need to cooperate in order to derive a complete interpretation. The secondary stage of image analysis and higher level processes may be collectively referred to as document image understanding.

Each of the tasks A to D can take advantage of the knowledge of how newspapers are typically laid out [2]. This paper is primarily concerned with the third task, viz., block classification. The method described here uses texture analysis and unlike previous methods does not use block size as a feature. The technique utilizes two matrices whose elements are counts of black–white pair run lengths and black–white–black combination run lengths, respectively. Three features are extracted from the two matrices to create a feature space. The feature space is used to classify blocks into a set of predetermined document categories.

The paper is organized into six sections. Section 2 describes and compares two previously known block segmentation procedures. Section 3 is concerned with feature extraction for block classification; the first part surveys previous methods and the rest describes texture matrices and features. In Section 4, the feature space and decision surfaces are explained. Experimental results are discussed in Section 5. Section 6 describes the flow chart of a newspaper image recognition system based on this block classification method. The discussion of Section 7 contains a summary of the paper and some research problems in block segmentation and classification.

2. BLOCK SEGMENTATION

The newspaper page is first digitized as a grey-scale image and then binarized, i.e., converted into a binary-valued image, before doing block segmentation. The method of binarization is dependent on the quality of the image. If the background is uniform then global thresholding suffices, otherwise adaptive thresholding is necessary [3]. Figures 1a and 1b show a grey-scale image digitized at 200 ppi and its binary version obtained by using a global threshold.

We describe here two different block segmentation techniques that are applicable to binary document images. They are: RLSA (run length smoothing algorithm) of Wong, Casey, and Wahl [4], and RXYC (recursive X–Y cuts) of Nagy, Seth, and Stoddard [5]. The two block segmentation methods yield different block sizes.

RLSA Procedure

The run length smoothing algorithm (RLSA) operates on a binary image under which any two black pixels (1's), which are equal to or less than a certain threshold t apart, are merged into a continuous stream of dark pixels. White pixels (0's) are left unchanged. For example, if the input sequence was

0001100000110010001

and the value of $t = 3$, then the result of the RLSA on this sequence would be

1111100000111110001.

The RLSA is first applied row-by-row and then column-by-column, yielding two distinct bit maps. The two results are then combined by applying a logical AND to each pixel location. The resulting RLSA image contains a *smear* wherever printed material appears on the original image. The thresholds t_x and t_y in the two directions need not be the same. The segmentation is expected to yield blocks each of which should contain only one type of data (text, graphics, halftone, etc.). The result obtained by applying the RLSA to the binary image of Fig. 1b is shown in Fig. 1c.

From Fig. 1c it is clear that all of the blocks segmented are scattered except photographic blocks. In the text area, most blocks correspond to text lines, and in the title area blocks correspond to some unit of adjacent large letters. The RLSA smears those black pixels that are very closely located. In Fig. 1c it can be observed that some text lines are stuck together. The reason is that the original image was skewed when it was digitized.

For the purpose of the next classification stage, the block results of RLSA can be made neater. In this study we used an algorithm to compute the bounding rectangular block for each connected component, and to label these rectangles. Figure 1d shows the result of using this algorithm with the RLSA result shown in Fig. 1c.

RXYC Procedure

Using recursive X–Y cuts (or recursive projection profile cuts) is another way to decompose a document image into a set of blocks. At each step of the recursive process, the projection profile is computed along both horizontal and vertical directions; a projection along a line parallel to, say the x-axis, is simply a sum of all the pixel values along that line. Then subdivision along the two directions is accomplished by making cuts corresponding to deep valleys, with width larger than a predetermined threshold, in the projection profile.

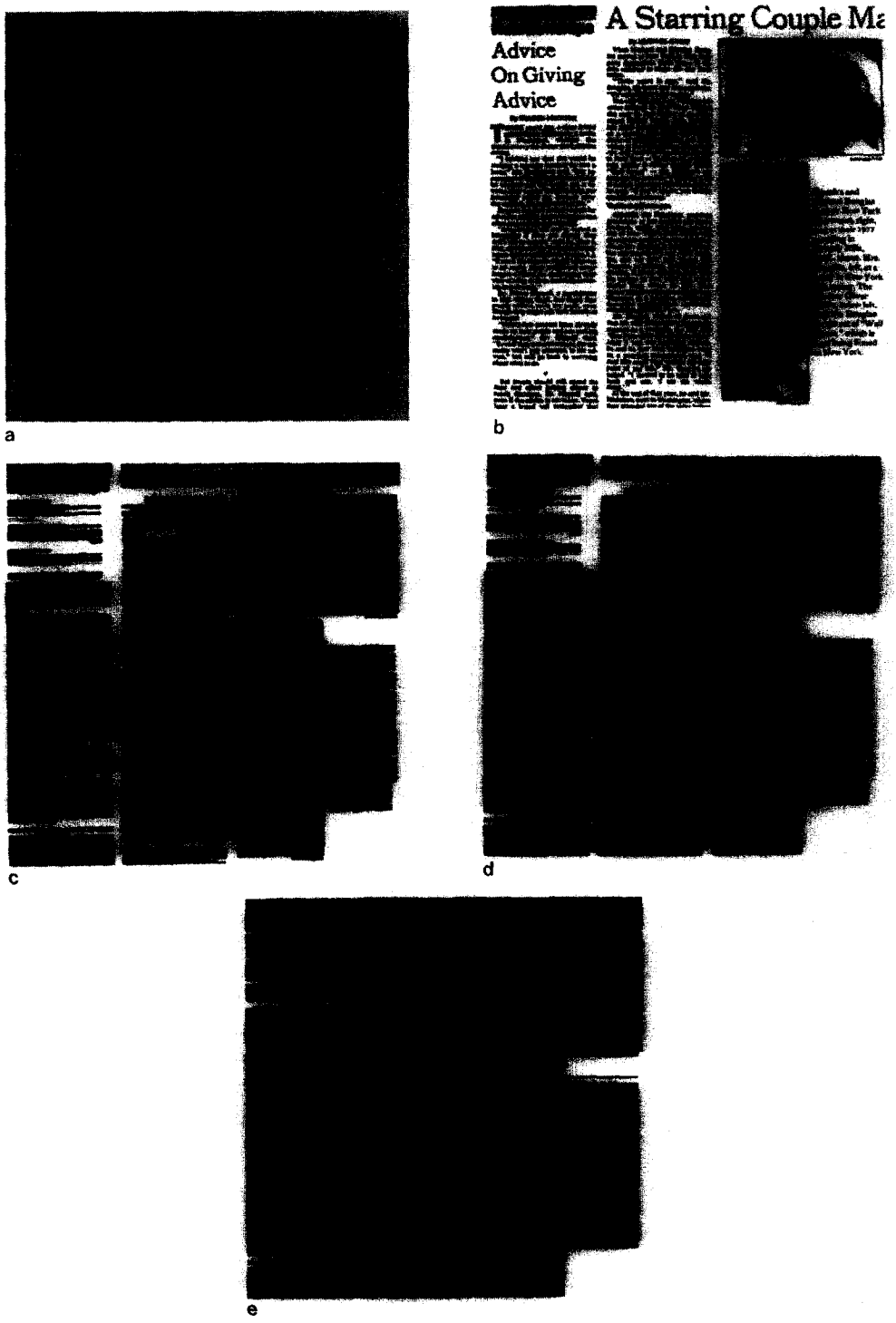

FIG. 1. (a) An example of a newspaper image with resolution of 200 ppi. (b) The binarized image. (c) The block segmentation result by using RLSA (run length smoothing algorithm) technique. (d) The result of computing the bounding rectangular block for each connected component. (e) The block segmentation result by using RXYC (recursive X–Y cuts) technique.

Based on the observation that printed pages are primarily made up of rectangular blocks, a page can be recursively cut into rectangular blocks. Thus the document is represented in the form of a tree of nested rectangular blocks. The application of cuts is based on the configuration of the pixels. A "local" peak detector is applied to horizontal and vertical "profiles" to detect local peaks (corresponding to thick black or white gaps) at which the cuts are placed; it is local in that the width is determined by the nesting level of the recursion, e.g., gaps between paragraphs are thicker than those between lines.

Although the RXYC procedure is to be applied to the original binary image, in our experimentation it was executed on the result of the RLSA. This was because from the result of the RLSA it was clear that the cuts can be easily placed at the zero intervals of the projection profile. Some prior knowledge about newspaper layout structure was used to control the depth of recursion of the RXYC procedure execution. For example, if the height of a rectangle is less than a threshold which is determined by the height of title areas, then this rectangle is not cut further. Because the interval between text lines is less than the interval between paragraphs, we can set appropriate interval thresholds for choosing cut places so that the paragraph of text can be bounded by one rectangle without subdivision. The result of RXYC executed on the image of Fig. 1d is shown in Fig. 1e.

Comparison of RLSA and RXYC Procedures

Based on experimental results, we can make a simple comparison between the two methods. For getting small blocks such that each block includes just a text line in the text area, RLSA is better than RXYC. This is because the RLSA smearing process is done within each text line, and not between text lines. So the blocks which contain just one text line can be obtained directly by using RLSA once. On the other hand, the RXYC cuts have to be done several times to make each text line into a block.

If large blocks corresponding to, say, paragraphs are needed, then RXYC is better than RLSA. A merging algorithm has to follow RLSA so as to merge blocks corresponding to single text lines into blocks corresponding to paragraphs.

A shortcoming of the RLSA method is that the resulting blocks may be not rectangular. If all blocks need to be rectangular, an algorithm for finding the bounding rectangle has to be applied after executing the RLSA procedure.

Both block segmentation methods yield poor results with skewed images. An approach to handle skewed images, based on using the Hough transform, is discussed in [6]. Baird [7] discusses a method of determining skew for a wide variety of page layouts.

For newspaper analysis, the newspaper images would be better separated into large enough blocks so that a text block can contain a whole paragraph. Also, more flexibility is needed for block segmentation to deal with the complex layout structure of a newspaper. Thus we prefer RXYC over RLSA.

3. BLOCK FEATURE EXTRACTION

3.1. *Previous Approaches*

A method for document block classification was proposed by Wong *et al.* [4]. The basic features used for classifying blocks are block height and block mean black pixel run length. A threshold is set with respect to the height of the block, and this

threshold is one parameter to classify text line blocks and graphic or halftone picture blocks. It exploits the fact that text lines have approximately a constant and small height.

More specifically, the following measurements are taken:

- total number of black pixels in the segmented RLSA image block (BC),
- minimum x–y coordinates of a block and its x–y lengths ($x_{\min}$, Δx, $y_{\min}$, Δy),
- total number of black pixels in the original image for the block (DC), and
- number of horizontal white–black transitions in the original image block (TC).

Classifying each block is done by computing several features for each block from the above measurements and then using a linear pattern classifier. The features computed are:

- the height of a block, $H = \Delta y$,
- its eccentricity, $E = \Delta x / \Delta y$,
- the ratio of the number of black pixels to the area of the surrounding rectangle, $S = BC/(\Delta x \times \Delta y)$, and
- the mean horizontal length of the black runs in the original data from the block, $R = DC/TC$.

A block is determined to be text if it is a textured stripe of mean height H_m and mean length of black run R_m. The distribution of values in the R–H plane derived from sample documents are observed to determine the discrimination function. Low R and H values represent regions containing text. To determine the threshold values of R and H that define the text region in the R–H plane, an adaptive method is used. This method estimates R_m, H_m and the standard deviation of R and H.

These values are then used to classify the various blocks as text, horizontal and vertical solid black lines, graphics, and halftone images by using the following pattern classification scheme that assumes linear separability:

- text: if $R < C_{21} \times R_m$ and $H < D_{22} \times H_m$,
- horizontal solid black lines: if $R > C_{21} \times R_m$ and $H < C_{22} \times H_m$,
- graphic and halftone images: if $E > 1/C_{23}$ and $H > C_{22} \times H_m$, and
- vertical solid black lines: if $E < 1/C_{23}$ and $H > C_{22} \times H_m$.

Although prior knowledge about the structural characteristics of a newspaper can be used for classifying blocks, in some cases these features will lead to classification errors. For example, the geometric characteristic that a text line has approximately a given constant height could be used for deciding a block to be a text line. But if the image was skewed while digitizing it, some text lines were linked together by the block segmentation procedure, then linked text lines may be classified into the graphic and halftone categories. When text lines are linked together by the RLSA procedure, say due to small line-to-line spacing, these linked text lines are classified as a graphic or halftone picture. This is a shortcoming of using block size as a

classification feature. This is the reason for our search for features other than geometric block size for classifying blocks.

3.2. *Textural Features*

An image region can be considered to possess a certain texture if it has some basic subpatterns which occur repeatedly according to some specified rules of arrangement. Many statistical approaches to texture analysis have been proposed. Each of these methods usually has two basic stages. First, a series of intermediate matrices are computed for the region. The elements of these matrices count the number of times a given event occurs in the image region. Then a set of features are computed from these intermediate matrices. The important thing is to define appropriate intermediate matrices.

The use of statistical texture analysis for discriminating between document image categories has been previously studied for the two-category case: print versus handwriting, formed characters versus dot-matrix characters, etc. [8]. In the present study, two matrices, viz., BW matrix and BWB matrix, are used to represent the textural characteristics of a newspaper image block.

Black–White Pair Run Length Matrix (BW Matrix)

Blocks of text are the most predominant components of documents. A text block (including titles) is composed of text lines, and each text line consists of a variety of characters. The basic components of characters are line segments which are nearly vertical or horizontal straight line segments and curved line segments. From this view point, the texture of a block of text is characterized by

(1) fundamental elements are line segments with different widths for different font sizes, and

(2) line segments assemble with certain density.

To represent these properties, the black–white pair run length matrix is defined as follows.

A black–white pair run is a set of consecutive black pixels followed by a set of consecutive white pixels as shown in Fig. 2. The length of the run is the number of pixels in the run. For simplification, various combinations of black–white pair runs with different proportions of length will be quantized into nine categories: $1, 2, \ldots, 9$. The category number represents the percentage (within an interval) of white part in a black–white pair run. The matrix element $p(i, j)$ specifies the number of times that the image contains a black–white pair run of length j, in the horizontal direction, consisting of white pixel runs having length as many as $10 * i$ percent of j. Table 1 gives an example of this matrix which is computed from a block of small

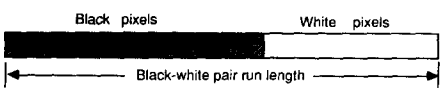

FIG. 2. A black–white pair run is defined as a set of consecutive black pixels followed by a set of consecutive white pixels.

TABLE 1

An Example of the Black–White Pair Run Length Matrix Obtained from a Block of Small Letters Chosen from the Image of Fig. 1b*

	Run length									
Categories	1	2	3	4	5	6	7	8	9	10
(1)	0	0	0	0	0	650	236	65	25	62
(2)	0	0	0	141	649	0	1171	227	101	100
(3)	0	0	35	0	0	1534	0	686	154	134
(4)	0	0	0	0	432	0	2191	0	261	171
(5)	0	25	0	109	0	688	962	1148	449	183
(6)	0	0	21	0	123	117	0	316	155	314
(7)	0	0	0	42	0	0	120	67	29	125
(8)	0	0	0	0	41	52	34	28	16	35
(9)	0	0	0	0	0	0	0	0	0	18

	Run length									
Categories	11	12	13	14	15	16	17	18	19	20
(1)	236	261	356	134	17	38	45	44	67	51
(2)	213	377	371	402	103	24	42	30	12	24
(3)	140	216	344	96	66	60	16	28	12	7
(4)	194	105	43	27	65	20	42	33	39	37
(5)	206	198	42	69	12	59	46	62	67	76
(6)	225	148	209	61	94	108	28	53	47	55
(7)	68	65	70	206	147	132	150	143	119	63
(8)	26	29	17	10	62	64	70	56	68	108
(9)	12	15	17	9	6	13	6	13	8	21

	Run length									
Categories	21	22	23	24	25	26	27	28	29	30
(1)	47	27	9	9	8	10	6	2	0	3
(2)	5	14	8	5	4	0	0	2	2	2
(3)	2	0	3	5	20	22	18	8	4	2
(4)	41	7	5	2	2	1	6	9	10	9
(5)	63	85	38	53	21	12	6	7	4	3
(6)	61	65	53	35	60	55	16	22	17	16
(7)	49	35	43	33	17	19	19	18	19	15
(8)	90	60	65	31	71	42	39	34	40	34
(9)	14	13	8	7	16	15	11	11	8	22

	Run length									
Categories	31	32	33	34	35	36	37	38	39	40
(1)	1	0	0	0	1	0	1	1	0	0
(2)	2	4	3	2	0	0	1	0	2	1
(3)	5	1	3	2	8	4	3	1	0	0
(4)	9	6	4	0	5	2	1	0	0	0
(5)	2	3	3	1	3	1	6	2	5	4
(6)	7	9	15	4	5	6	1	0	2	2
(7)	11	11	6	6	1	1	1	1	3	3
(8)	17	20	21	28	16	13	16	21	21	7
(9)	12	12	14	24	6	12	13	9	16	22

	Run length									
Categories	41	42	43	44	45	46	47	48	49	50
(1)	0	0	1	1	1	1	2	0	0	0
(2)	0	0	1	0	0	0	0	0	0	0
(3)	0	0	0	0	0	1	0	1	1	0
(4)	1	1	1	2	0	3	0	0	0	0
(5)	2	0	0	0	0	0	0	0	0	0
(6)	0	0	1	0	1	0	0	0	0	0
(7)	2	0	0	2	5	1	8	3	0	1
(8)	5	4	4	9	2	15	5	5	11	3
(9)	21	25	20	13	15	22	26	22	14	14

* This matrix has 389 columns. The columns 51 to 389 are omitted because most values are zero for them.

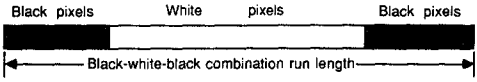

FIG. 3. A black–white–black combination run is defined as a pixel sequence in which two black pixels runs are separated by a white pixel run.

letter text chosen from the binary image of Fig. 1b; it corresponds to the second column from the left.

Black–White–Black Combination Run Length Matrix (BWB Matrix)

Blocks in which line drawings predominate are an important component category of newspapers. They include a variety of graphics, weather maps, cartoons, advertisements, etc. The content of line drawings is very complex and usually contains some printed text. The characteristic of such blocks, which we refer to as line drawings, is that there exist several large white spaces between black lines. Although there exist many white spaces between black strokes in case of letters, their size is obviously less than those of line drawings. The black–white–black combination run length matrix is designed to represent the distribution of white space between black lines.

A black–white–black combination run is defined as a pixel sequence in which two black pixel runs are separated by a white pixel run as shown in Fig. 3. The length of the run is defined as the number of white pixels in the white pixel run. The length of a black pixel run is fixed and assigned into one of three categories as shown in Table 2. Both black pixel runs should have approximately the same length and lie in the same category. The matrix element $p(i, j)$ is the number of times that the image contains a black–white–black combination run, in the horizontal direction, with white pixel run length j and black pixel runs with length lying in category i. So the BWB matrix should have three rows. Table 3 gives an example of this matrix which is obtained from the same block of text as for Table 1.

3.3. *Feature Definitions*

We derive two features, F_1 and F_2, from the BW matrix. They correspond to short run emphasis and long run emphasis, which were used by Galloway [9] for grey-level run length matrices.

TABLE 2
Arrangement of Categories for the Black–White–Black Combination Runs

Category	Assignment of run length range in pixels for both black pixel runs
1	1–4
2	5–8
3	9–12

TABLE 3

An Example of the Black–White–Black Combination Run Length Matrix Obtained from a Block of Small Letters Chosen from the Image of Fig. 1b*

	Run length									
Categories	1	2	3	4	5	6	7	8	9	10
(1)	48	174	253	176	69	37	18	22	14	10
(2)	71	82	65	43	33	12	8	2	3	6
(3)	33	45	55	13	3	2	3	1	5	2

	Run length									
Categories	11	12	13	14	15	16	17	18	19	20
(1)	18	18	5	11	12	11	12	12	10	6
(2)	4	10	3	7	2	2	7	3	4	4
(3)	10	3	2	1	1	0	0	5	1	0

	Run length									
Categories	21	22	23	24	25	26	27	28	29	30
(1)	10	7	8	2	5	5	4	6	4	3
(2)	1	0	0	1	1	0	1	0	0	2
(3)	0	0	3	1	0	0	0	0	0	0

	Run length									
Categories	31	32	33	34	35	36	37	38	39	40
(1)	0	4	4	3	4	3	6	7	4	2
(2)	0	0	0	1	0	1	0	0	0	0
(3)	0	0	0	0	0	0	0	0	0	0

	Run length									
Categories	41	42	43	44	45	46	47	48	49	50
(1)	3	1	4	4	3	2	1	1	0	1
(2)	0	0	0	0	0	1	0	0	0	0
(3)	0	0	0	0	0	0	0	0	0	0

	Run length									
Categories	51	52	53	54	55	56	57	58	59	60
(1)	0	0	0	2	3	0	3	0	3	1
(2)	0	0	0	0	0	0	0	0	0	1
(3)	0	0	0	0	0	0	0	0	0	0

	Run length									
Categories	61	62	63	64	65	66	67	68	69	70
(1)	1	1	1	1	4	1	0	1	0	0
(2)	0	1	0	0	0	1	1	0	0	0
(3)	0	0	0	0	0	0	0	0	0	0

	Run length									
Categories	71	72	73	74	75	76	77	78	79	80
(1)	2	0	0	1	2	0	2	0	1	0
(2)	0	0	0	0	0	0	0	0	0	0
(3)	0	0	0	0	0	0	0	0	0	0

	Run length									
Categories	81	82	83	84	85	86	87	88	89	90
(1)	1	0	1	0	1	1	2	0	0	0
(2)	0	0	0	0	0	0	0	0	0	0
(3)	0	0	0	0	0	0	0	0	0	0

*This matrix has 348 columns. The columns 91 to 348 are omitted because most values are zero for them.

(1) *Short Run Emphasis*

$$F_1 = \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \left(p(i, j)/j^2 \right) \Bigg/ \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} p(i, j), \tag{1}$$

where $p(i, j)$ is the (i, j)th entry in the given run length matrix, N_c is the number of different kinds of pixel runs, and N_r is the number of different run lengths that occurs. For the black–white pair run length matrix $N_c = 9$. The feature F_1 tends to emphasize short runs because the numerator of (1) is a summation of run length counts where each run length count is divided by the square of the run length. This feature is normalized by the denominator of (1) which is the total number of runs in the image.

It can be expected that the value of F_1 for small letters will be larger than the value of F_1 for large letters, because white spaces between strokes in small letters are smaller than those in large letters. Actually photograph blocks have the largest value of F_1, as seen in Section 4, because a photograph is composed of very small black dots with very small white spaces in between.

(2) *Long Run Emphasis*

$$F_2 = \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} j^2 p(i, j) \Bigg/ \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} p(i, j). \tag{2}$$

This feature should emphasize long runs. It can be expected that large letter blocks will have a large value of F_2 due to the same reason mentioned above.

A third feature, F_3, is derived from the BWB matrix for classifying graphics blocks.

(3) *Extra Long Run Emphasis*

$$F_3 = \sum_{j=T_1}^{N_r} j^2 \left(\sum_{i=1}^{N_c} p'(i, j) \right) \Bigg/ \sum_{j=T_1}^{N_r} \sum_{i=1}^{N_c} p'(i, j), \tag{3}$$

where

$$p'(i, j) = \begin{cases} p(i, j) & \text{if } p(i, j) > T_2 \\ 0 & \text{if } p(i, j) \le T_2 \end{cases}; \tag{4}$$

$p(i, j)$ is the (i, j)th entry in the given run length matrix. Threshold T_1 is set to delete short run lengths because only very long run lengths are needed to express the characteristics of graphics blocks. The threshold T_2 is for deleting the effect of small value of $p(i, j)$ because long run appears occasionally in letters blocks and photograph blocks and makes some small value of long run length. Thresholds T_1 and T_2 are determined by experimentation; in this study $T_1 = 50$, $T_2 = 15$.

4. BLOCK CLASSIFICATION

As mentioned in Section 3, the combination of various line segments with different width by certain density is a basic characteristic which can be used to describe the textural property of a block of text and a block of graphics. The goal of this study was to find features that would be able to measure these basic characteristics. The three features, F_1, F_2, and F_3, were measured for several sample blocks segmented from newspaper images.

Five kinds of blocks, corresponding to blocks of small letters, medium letters, large letters, graphics and halftones were collected and measured. In the 2-dimensional feature space defined by F_1 and F_2, the blocks corresponding to small letters, medium letters, large letters, and halftones are clustered together within each class and are well separated between classes. Graphics blocks are not well separated from

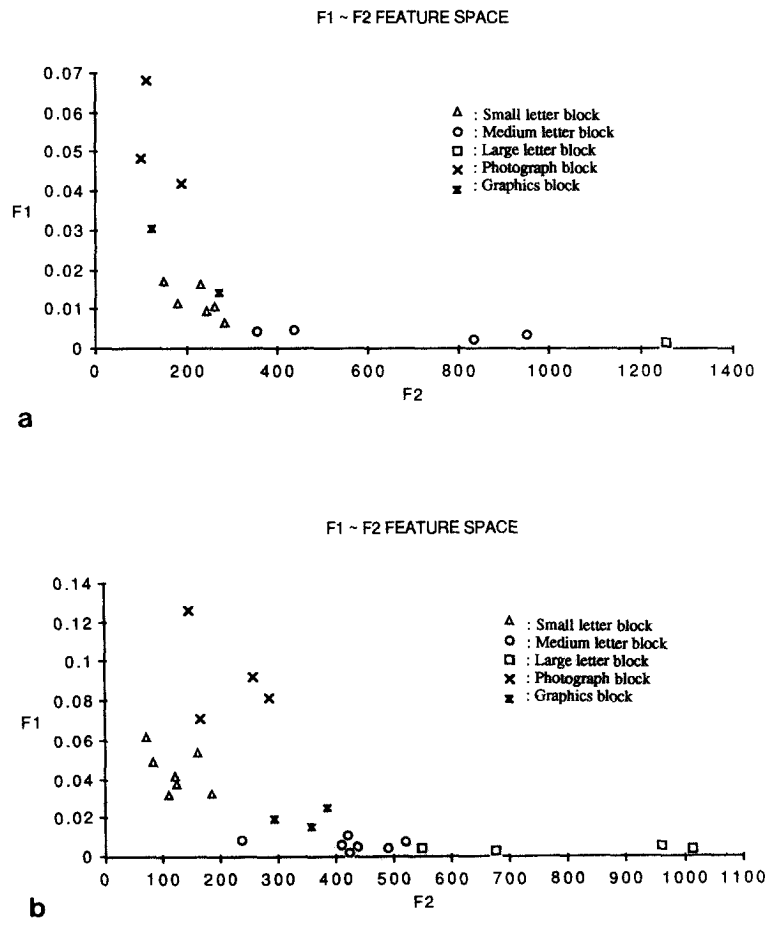

FIG. 4. An example of feature space determined by features F_1 and F_2 with samples obtained using resolution of (a) 200 ppi and (b) 100 ppi.

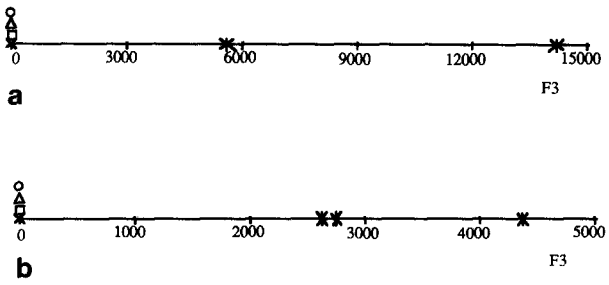

FIG. 5. An example of feature space determined by feature F_3 with samples obtained using resolution of (a) 200 ppi and (b) 100 ppi.

the other classes in the F_1–F_2 space. However, feature F_3 separates graphics blocks from the other four categories very well.

Figure 4 shows an example of the F_1–F_2 feature space created by results measured from five kinds of image blocks with resolution of 200 and 100 ppi. In the figure, the symbols, viz., triangle, circle, square, star, and cross symbols represent *patterns* (or feature vectors) corresponding to blocks containing small letters, medium letters, large letters, graphics, and halftones, respectively. It should be mentioned here that in computing these features, that parts of the BW matrix, which have entries with length of run larger than 50, were cut out. The reason for doing this is that the entries of this part of the matrix have almost zero value. Also the matrices of different blocks with different sizes have different numbers of columns, so this process could provide an identical comparison condition.

From Fig. 4 it can be seen that the patterns corresponding to halftones, small letters, medium letters, and large letters are clustered separately, but the patterns corresponding to graphics blocks are mixed up with small letter blocks. Thus the F_1 versus F_2 feature space is valuable for discriminating photographs, small letter blocks, medium letter blocks, and large letter blocks simultaneously, but cannot be used to separate graphics blocks.

Figure 5 shows an example of a 1-dimensional feature space determined by feature F_3, in which the symbols mean the same as in Fig. 4. In this feature space, blocks corresponding to small letters, medium letters, large letters, and halftones have zero value and graphics blocks have very large values. This fact gives us a method for separating graphics blocks from other kind of blocks.

We use a 3-dimensional feature space created by F_1, F_2, and F_3 to distinguish between the five kinds of blocks. It is clear from Fig. 5 that the features for graphics blocks and others are separated well enough. So we can simply set up a plane, which is parallel to the F_1–F_2 coordinate plane, as a decision surface to classify graphics blocks. This decision surface, called DS1, can be expressed as

$$w_3 F_3 + w_4 = 0, \tag{5}$$

where w_i's are weights. The projection of this decision surface on F_1–F_3 (or F_2–F_3) coordinate plane is a straight line as shown in Fig. 6. The feature F_3 for letters and

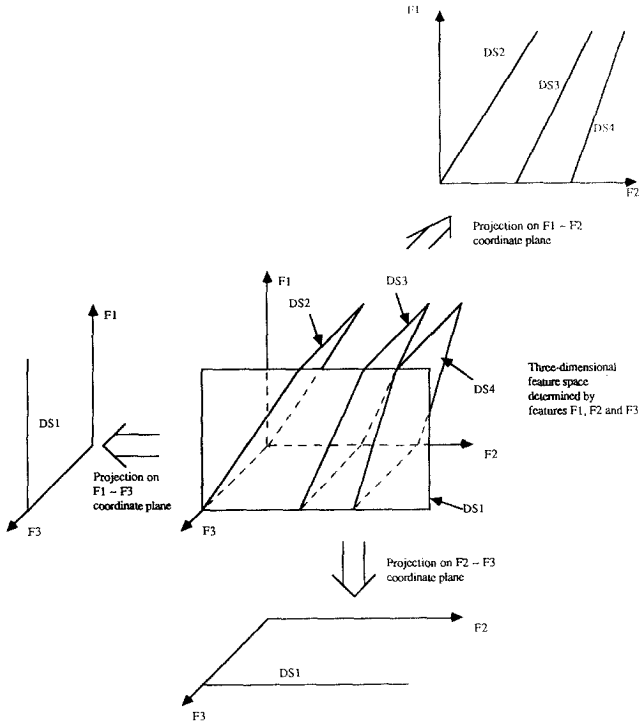

FIG. 6. The illustration of four decision surfaces DS1, DS2, DS3, and DS4 created in a 3-dimensional feature space determined by features F_1, F_2, and F_3.

photograph blocks takes zero value, but the features F_1 and F_2 for these kinds of blocks are linearly separable. For classifying small letters, medium letters, large letters, and halftones, three decision surfaces, called DS2, DS3, DS4, are defined. The three decision surfaces are all parallel to the F3 axis and are each defined as

$$w_1F_1 + w_2F_2 + w_5 = 0. \tag{6}$$

The projections of the three decision surfaces on F_1–F_2 coordinate plane are three straight lines as shown in Fig. 6.

Training Procedure

A fixed increment error correction procedure can be used (see [10]) to determine the weights w_1, w_2, and w_5 in Eq. (6) for each of the decision surfaces DS2, DS3, and DS4. Let the pattern with feature values F_1 and F_2 be represented by F, an augmented feature vector defined as

$$F = \begin{bmatrix} F_1 \\ F_2 \\ 1 \end{bmatrix}, \tag{7}$$

and W be a weight vector which is defined as

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_5 \end{bmatrix}. \tag{8}$$

For any F belonging to the sample set of photograph blocks, the product F^TW must be positive, i.e., $F^TW > 0$. If the pattern is classified incorrectly (i.e., $F^TW < 0$) or the result is undefined (i.e., $F^TW = 0$), then let the new weight vector be

$$W' = W + \alpha F, \tag{9}$$

where $\alpha > 0$ is the correction increment. On the other hand, for any F belonging to the samples of other categories, the product F^TW must be negative, i.e., $F^TW < 0$. If the classification result is in error (i.e., $F^TW > 0$) or undefined (i.e., $F^TW = 0$), then let

$$W' = W - \alpha F. \tag{10}$$

Block Classification Strategy

The block classification strategy is as follows. First check the feature vector location with decision surface DS1, i.e., to see whether feature F_3 is larger than zero. If the feature vector for a block is located higher than decision surface DS1, then it is classified as graphics. Otherwise it belongs to one of the other four classes; which is resolved by further testing. If the feature vector appears higher than decision surface DS2, then it is classified as a halftone. If the feature vector is located lower than decision surface DS2 but higher than decision surface DS3, it is classified as a small letter block. If the feature vector is located lower than decision surface DS3 but higher than decision surface DS4, it is classified as a medium letter block. Otherwise if the feature vector is located lower than decision surface DS4, it is classified as a large letter block.

5. EXPERIMENTAL RESULTS

The images used for our experimental study are taken from The New York Times, USA Today, and The Buffalo News. The original digitized images were obtained at two resolutions: 200 and 100 ppi, respectively.

There are five weights, $w_1, \ldots, w_5$, that need to be determined. For decision surface DS1, the weights w_3 and w_4 can be chosen manually because the features for graphics blocks and other blocks are separated well enough. In this study, the weights w_3 and w_4 were chosen as $w_3 = 1$, $w_4 = -1000$, thereby defining decision

TABLE 4

Training Samples Corresponding to Resolution of 200 ppi; Training Samples Consisted of 32 Small Letter Blocks, 16 Medium Letter Blocks, 1 Large Letter Block, and 3 Photographs

	F1	F2	F1	F2	F1	F2	F1	F2
Small letter blocks	0.0097	246.2	0.0124	238.6	0.0137	214.1	0.0093	262.5
	0.0166	232.0	0.0169	178.7	0.0167	163.5	0.0153	195.2
	0.0173	149.5	0.0109	264.8	0.0135	210.0	0.0156	167.7
	0.0172	163.2	0.0163	167.9	0.0137	219.0	0.0121	239.9
	0.0151	160.5	0.0169	161.7	0.0149	201.1	0.0155	166.8
	0.0149	166.6	0.0168	163.2	0.0175	151.5	0.0141	235.2
	0.0152	192.4	0.0129	224.1	0.0157	182.2	0.0148	192.8
	0.0118	181.2	0.0144	151.5	0.0069	285.4	0.0075	276.0
Medium letter blocks	0.0037	951.2	0.0046	687.7	0.0040	480.2	0.0047	437.7
	0.0045	354.9	0.0025	835.0	0.0022	762.6	0.0045	469.9
	0.0019	812.9	0.0037	687.3	0.0029	604.1	0.0043	456.4
	0.0016	968.8	0.0028	784.9	0.0049	386.1	0.0049	362.4
Large letter block	0.0017	1254.6						
Photograph blocks	0.0483	99.9	0.068	111.5	0.042	188.9		

surface function of DS1 by

$$F_3 - 1000 = 0.$$

The weights w_1, w_2, and w_5 are determined by a training procedure using several samples of features for different kinds of blocks which are shown in Tables 4 and 5. The entries in Table 4 correspond to four 200 ppi images including Fig. 1a and Fig. 10a, and the entries in Table 5 correspond to seven 100 ppi images that included Fig. 11a. The results of the training procedure for 200 and 100 ppi images are given in Tables 6 and 7, respectively. Figure 7 shows the projections of three decision surfaces on F_1–F_2 coordinate plane and the classification regions corresponding to four kinds of blocks.

An example of block classification result by using four decision surfaces obtained above for the image shown in Fig. 1a is shown in Fig. 8a, in which white blocks correspond to small letter blocks, grey blocks correspond to medium letter blocks, and black blocks correspond to photograph blocks. The output images including only one kind of component in each are shown in Fig. 8b–d, respectively.

In order to illustrate that the classification method presented here is not related to block size, the classification experiment was also executed on the block segmentation result obtained by using the RLSA technique shown in Fig. 1d. The decision surfaces used in this experiment are same as above. Figure 9 gives the block classification result, in which white, grey, dark grey, and black blocks correspond to small letters, medium letters, large letters, and halftones, respectively. Some small

TABLE 5

Training Samples Corresponding to Resolution of 100 ppi; Training Samples Consisted of 83 Small Letter Blocks, 40 Medium Letter Blocks, 15 Large Letter Blocks, and 10 Photographs

	F1	F2	F1	F2	F1	F2	F1	F2
Small letter blocks	0.032	111.0	0.030	168.8	0.036	126.7	0.031	207.4
	0.033	185.8	0.029	218.1	0.030	105.9	0.023	125.9
	0.049	81.8	0.040	121.0	0.045	102.2	0.029	225.2
	0.030	188.7	0.042	126.7	0.042	140.5	0.033	174.3
	0.051	105.6	0.036	122.7	0.045	97.2	0.022	219.6
	0.041	111.8	0.039	114.7	0.049	90.0	0.052	92.8
	0.043	110.0	0.046	108.4	0.048	86.1	0.038	123.7
	0.047	113.1	0.046	109.5	0.054	100.4	0.041	85.1
	0.047	97.6	0.040	127.4	0.044	111.4	0.051	112.9
	0.051	20.5	0.039	256.5	0.048	90.2	0.029	173.8
	0.041	122.2	0.036	150.3	0.033	158.5	0.035	146.9
	0.029	265.4	0.038	114.4	0.028	185.4	0.054	159.1
	0.040	116.4	0.034	268.5	0.035	122.2	0.041	103.3
	0.051	82.3	0.045	91.3	0.050	80.2	0.045	109.6
	0.022	188.0	0.036	115.3	0.048	99.1	0.041	131.4
	0.035	166.2	0.041	133.4	0.030	179.9	0.030	209.4
	0.028	149.2	0.039	126.4	0.011	207.0	0.050	97.5
	0.062	71.3	0.043	102.4	0.042	118.2	0.043	120.0
	0.054	85.8	0.042	121.1	0.044	108.4	0.039	119.2
	0.035	142.6	0.042	88.3	0.037	111.4	0.047	92.2
	0.037	132.4	0.050	93.6	0.039	123.0		
Medium letter blocks	0.0062	408.5	0.0045	486.3	0.0041	462.2	0.0038	453.5
	0.0050	490.5	0.0063	361.4	0.0068	373.5	0.0061	375.9
	0.0051	435.9	0.0070	270.0	0.0076	318.0	0.0054	402.8
	0.0051	356.3	0.0181	371.4	0.0074	363.7	0.0051	358.4
	0.0043	476.8	0.0054	428.9	0.0059	405.3	0.0057	384.2
	0.0086	242.8	0.0067	276.9	0.0053	380.1	0.0066	359.8
	0.0065	387.5	0.0085	236.5	0.0052	471.6	0.0024	424.6
	0.0109	421.2	0.0045	453.4	0.0066	422.7	0.0057	406.9
	0.0038	440.0	0.0046	471.2	0.0063	365.3	0.0078	521.3
	0.0042	469.4	0.0037	487.2	0.0069	535.2	0.0054	500.3
Large letter blocks	0.0054	959.8	0.0033	607.8	0.0044	547.6	0.0032	676.4
	0.0061	679.3	0.0036	629.3	0.0033	547.9	0.0035	554.2
	0.0045	565.9	0.0064	579.4	0.0049	659.0	0.0044	580.8
	0.0039	1013.8	0.0031	854.9	0.0042	549.9		
Photograph blocks	0.079	229.3	0.072	220.4	0.102	98.8	0.081	283.9
	0.126	146.9	0.140	136.8	0.092	256.2	0.095	89.2
	0.071	164.2	0.085	146.3				

TABLE 6

Decision Surface Parameters at 200 ppi

	Decision surface		
Correction increment	DS2 $\alpha = 0.00001$	DS3 $\alpha = 0.0001$	DS4 $\alpha = 0.000001$
Initial values of weights	$w_1 = 1.0$	$w_1 = 1.0$	$w_1 = 1.0$
	$w_2 = 0.0$	$w_2 = 0.0$	$w_2 = 0.0$
	$w_5 = 0.0$	$w_5 = 0.0$	$w_5 = 0.0$
Final values of weights	$w_1 = 1.000003$	$w_1 = 1.00102$	$w_1 = 1.000019$
	$w_2 = -0.000135$	$w_2 = -0.00022$	$w_2 = -0.000013$
	$w_5 = 0.00004$	$w_5 = 0.0666$	$w_5 = 0.011952$

TABLE 7
Decision Surface Parameters at 100 ppi

Correction increment	Decision surface DS2 $\alpha = 0.00001$	DS3 $\alpha = 0.0001$	DS4 $\alpha = 0.0001$
Initial values of weights	$w_1 = 1.0$	$w_1 = 1.0$	$w_1 = 1.0$
	$w_2 = 0.0$	$w_2 = 0.0$	$w_2 = 0.0$
	$w_5 = 0.0$	$w_5 = 0.01$	$w_5 = 0.0$
Final values of weights	$w_1 = 0.99899$	$w_1 = 1.000578$	$w_1 = 1.00053$
	$w_2 = -0.000006$	$w_2 = -0.000168$	$w_2 = -0.00015$
	$w_5 = -0.06252$	$w_5 = 0.02399$	$w_5 = 0.0776$

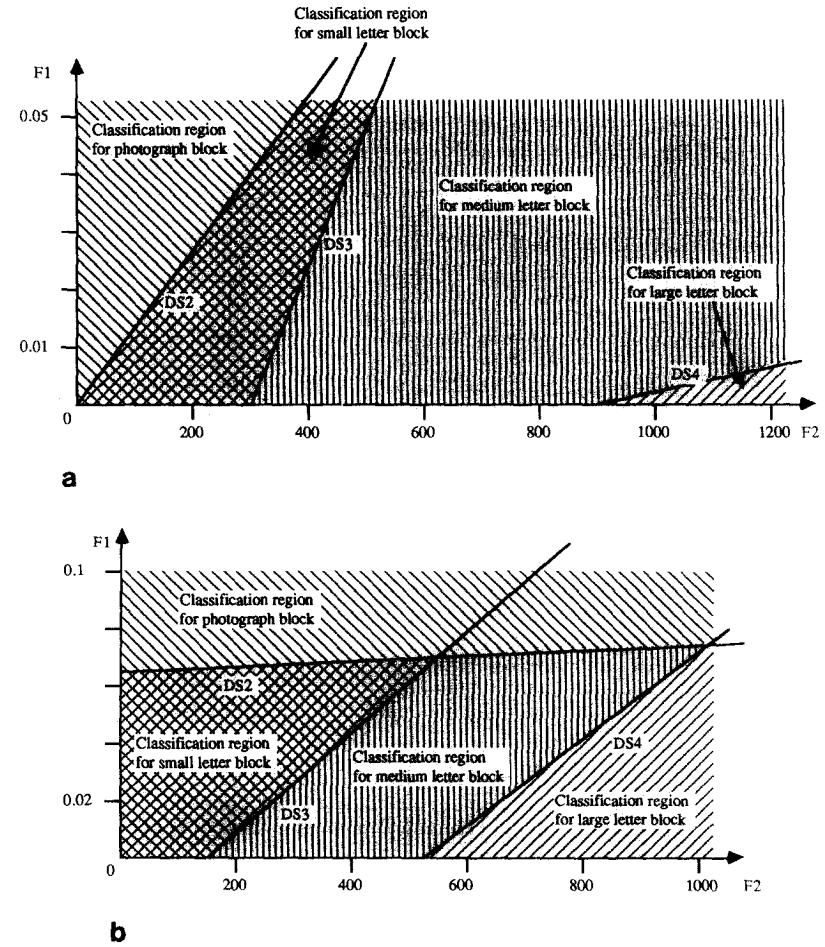

FIG. 7. The projections of three decision surfaces DS2, DS3, and DS4 on F_1–F_2 coordinate plane. Decision surfaces are determined with relation to the case of (a) 200 ppi resolution and (b) ppi resolution.

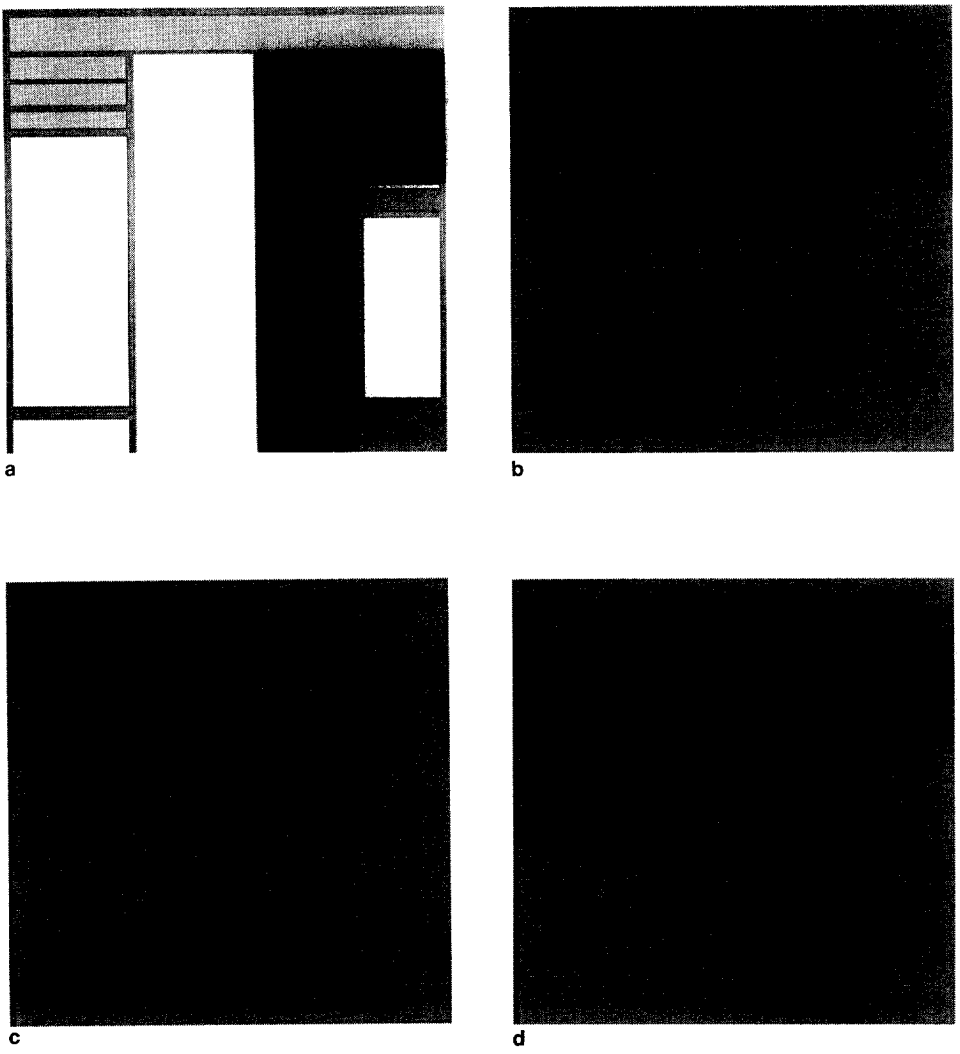

FIG. 8. (a) The block classification result by using feature space for the block image shown in Fig. 1e, (b) The text areas recognized. (c) The title areas recognized. (d) The photograph areas recognized.

empty rectangles illustrate blocks which were rejected by the recognition procedure. Actually no effective information appears in these rejected blocks. There were some errors corresponding to the two solid black lines, few short text lines, and part of medium letters in the original image.

Figures 10a and b give an example of a newspaper image which includes a line drawing block and its binary version. Figure 10c expresses the block segmentation result for Fig. 10b. Figure 10d shows a graphics classification result by using this approach.

Figure 11a shows a part of a newspaper image digitized at a resolution of 100 ppi. Figure 11b gives its binary version. The block segmentation result using the RXYC technique is shown in Fig. 11c. Figure 11d expresses the block classification result

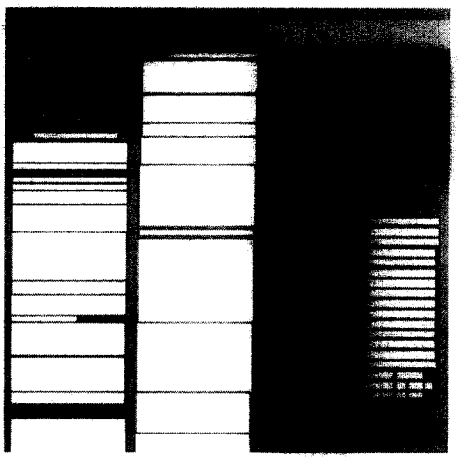

FIG. 9. The block classification result by using feature space for the block image shown in Fig. 1d.

by using the four decision surfaces explained above, in which white, grey, dark grey, and black correspond to blocks containing small letters, medium letters, large letters, and halftones, respectively. It illustrates that this approach is effective for images with a different resolution. Figure 12 gives another example of a block classification result for a whole page of newspaper at a resolution of 100 ppi. While the classification results of Figs. 1, 10, and 11 correspond to "testing on the training set," the classification result of Fig. 12 corresponds to a previously unseen test case.

The overall performance of the block classification method is given in Table 8. In the case of medium letter blocks, some classification errors were due to segmentation errors, i.e., while segmenting the whole newspaper page, medium letters were merged into small letter blocks and were classified as small letter blocks. Some large letter blocks were classified as medium letter blocks because small letters were mixed in these blocks. The overall correct recognition rate for 100 ppi resolution was 94% with an error rate of 6%, and for 200 ppi resolution was 100%. For the image of Fig. 1a, the block segmentation method yielded 21 blocks at 200 ppi as shown in Fig. 1e. For the same image at 100 ppi (leaving every other pixel out in the 200 ppi image), the block segmentation yielded only 12 blocks as shown in Fig. 13. At both resolutions all blocks were classified correctly except a few small blocks which consist of only noise. The high performance with these examples indicates that this approach leads to accurate classification.

The images were digitized using an EIKONIX 850 digitizing camera. The method was implemented on a SUN 3/60 computer. An example of the speed of recognition process including binarization, block segmentation, texture calculation, and block classification for the images of Figs. 1a and 13a is as follows. User times using 100 ppi (and 200 ppi) resolution were: 100 (407) s for binarization, 171 (568) s for block segmentation and 65 (351) s for texture calculation and block classification. System times were: 2.1 (8.3) s for binarization, 2.0 (7.2) s for block segmentation and 0.6 (2.3) s for texture calculation and block classification. The speed of block segmentation, texture calculation, and block classification for different newspaper images is quite different and depends on the number of blocks. In this example, 12

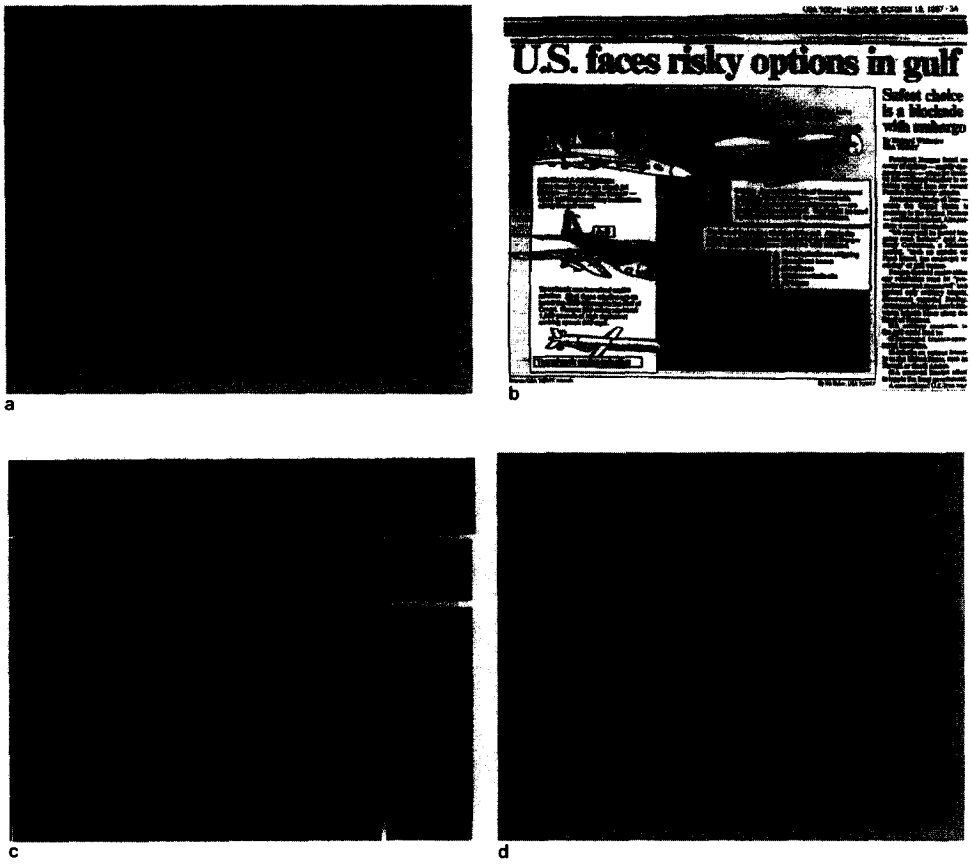

FIG. 10. (a) An example of a newspaper image with resolution of 200 ppi. (b) The binarized image. (c) The block segmentation result by using RXYC (recursive X–Y cuts) technique. (d) The line drawing block classification result.

and 21 blocks were segmented and classified with resolution 100 ppi and 200 ppi, respectively.

6. NEWSPAPER IMAGE RECOGNITION SYSTEM

A system to take as input a grey-scale newspaper image and produce as output segmented and labeled regions is as follows. Figure 14 gives a flow chart of a system that is based on the texture analysis method of block classification presented in this paper.

In Fig. 14 the thick arrow line indicates the flow of data during block classification, and the thin arrow line expresses the training procedure to obtain the feature space and decision surfaces from selected block samples. The newspaper is scanned and digitized at 100 or 200 ppi resolution depending on the performance–speed trade-off. The digitized image is binarized by adaptive thresholding to account for a nonuniform background. The RXYC (recursive X–Y cuts) technique is applied to this binary image to get the block segmentation result. For each block two matrices

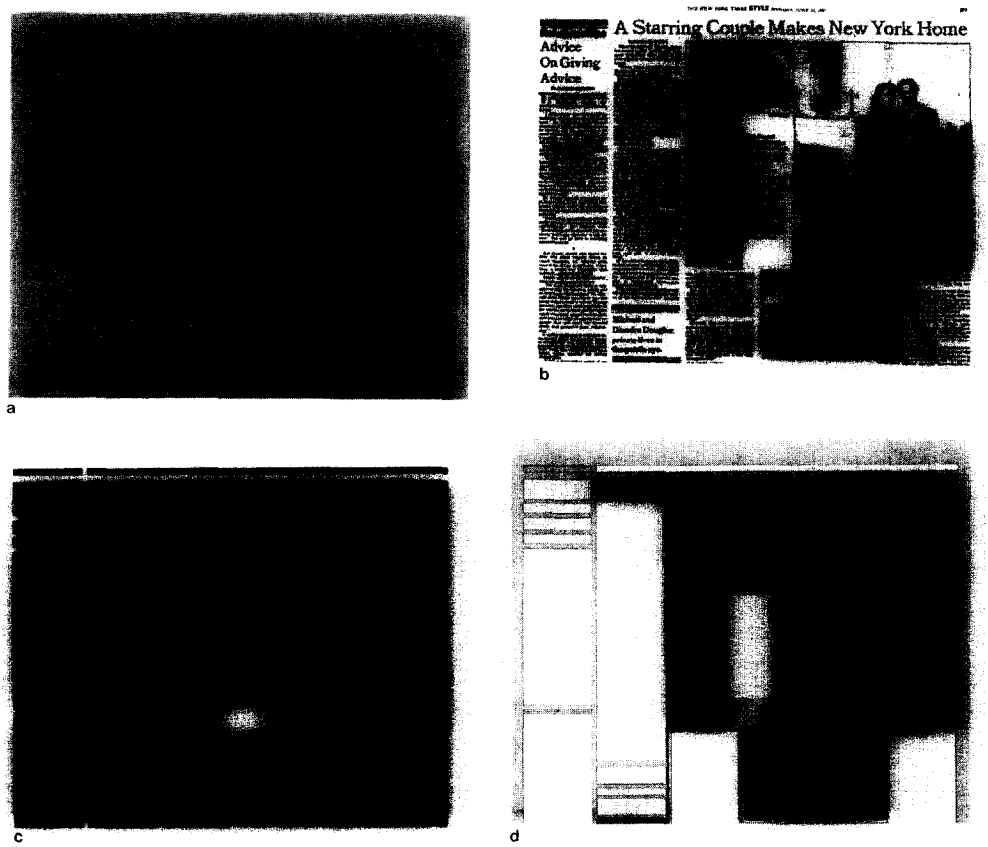

FIG. 11. (a) An example of a newspaper image with resolution of 100 ppi. (b) The binarized image. (c) The block segmentation result by using RXYC technique. (d) The block classification result.

are computed, and three features are extracted based on these matrices. The block is classified by checking with four decision surfaces in the feature space.

For obtaining a 3-dimensional feature space several block samples including halftones, graphics, small letter blocks, medium letter blocks, and large letter blocks are selected and processed to obtain features by using the procedures mentioned above. The four decision surfaces can be determined by the training procedure, in which classification errors are used to change the decision surfaces.

7. DISCUSSION

Our investigation shows the ability of texture analysis to classify blocks of newspaper images. Experimental results indicate that the method based on extracting three features from the **BW** and **BWB** texture matrices works very well. The features are good expressions of texture characteristics for different components of a newspaper.

The size of the text block segmented from the newspaper image should be big enough to contain more than a few letters so that the extracted features represent the image statistics accurately. Experimental results indicate that for a text block

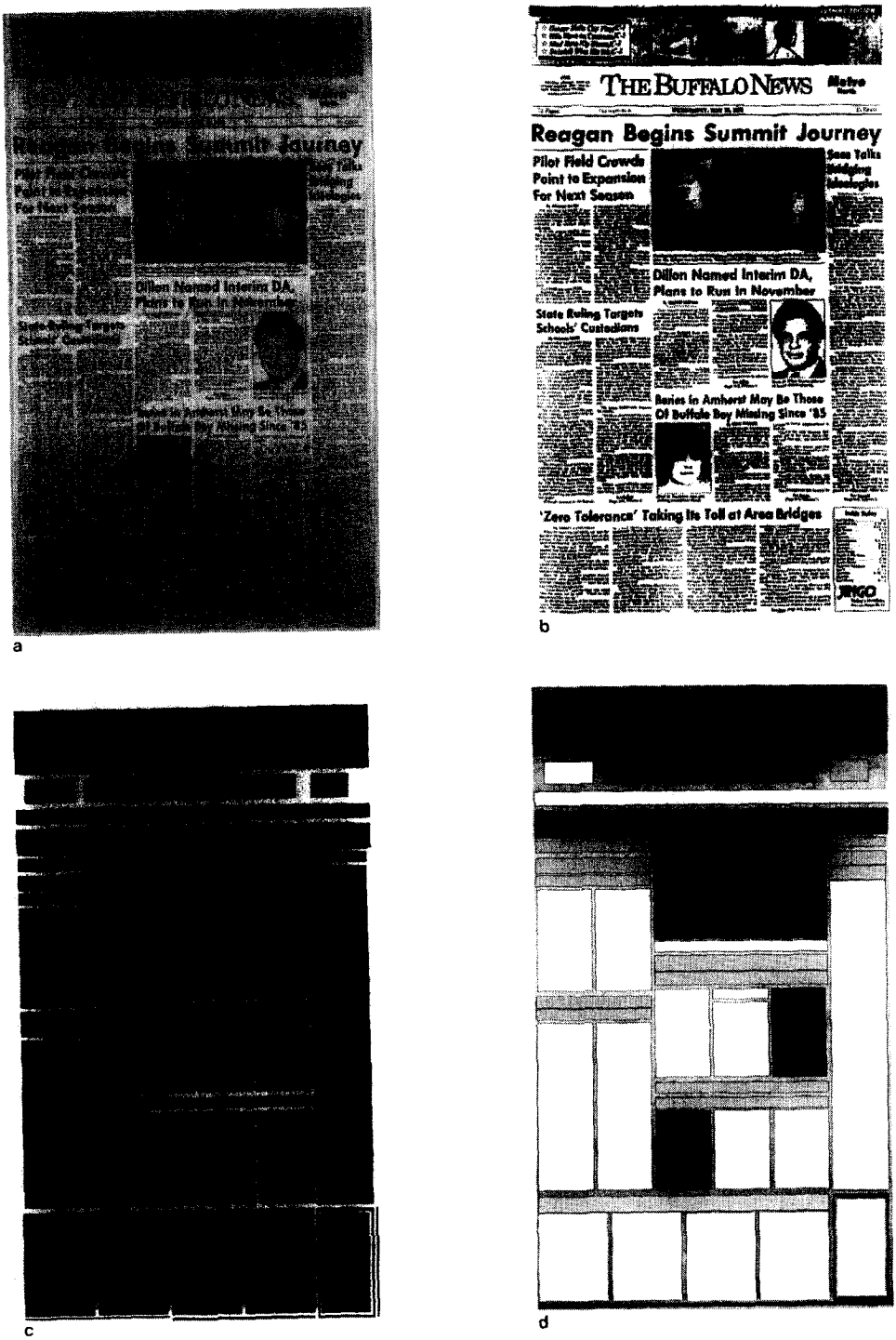

FIG. 12. Testing on an image not in training set: (a) the original image at resolution of 100 ppi; (b) the binarized image; (c) the block segmentation result by using RXYC technique; and (d) the block classification result.

TABLE 8
Performance of the Block Classification Method (100 ppi/200 ppi)

Categories of blocks (input)	Small letter	Medium letter	Large letter	Halftone	Graphics
Total number of test blocks	109/22	51/6	23/9	13/3	4/1
Block classification					
Small letter	106/22	3/0	0/0	0/0	0/0
Medium letter	0/0	48/6	5/0	0/0	0/0
Large letter	0/0	0/0	18/9	0/0	0/0
Halftone	3/0	0/0	0/0	13/3	0/0
Graphics	0/0	0/0	0/0	0/0	4/1
Performance percentage	97/100	94/100	78/100	100/100	100/100

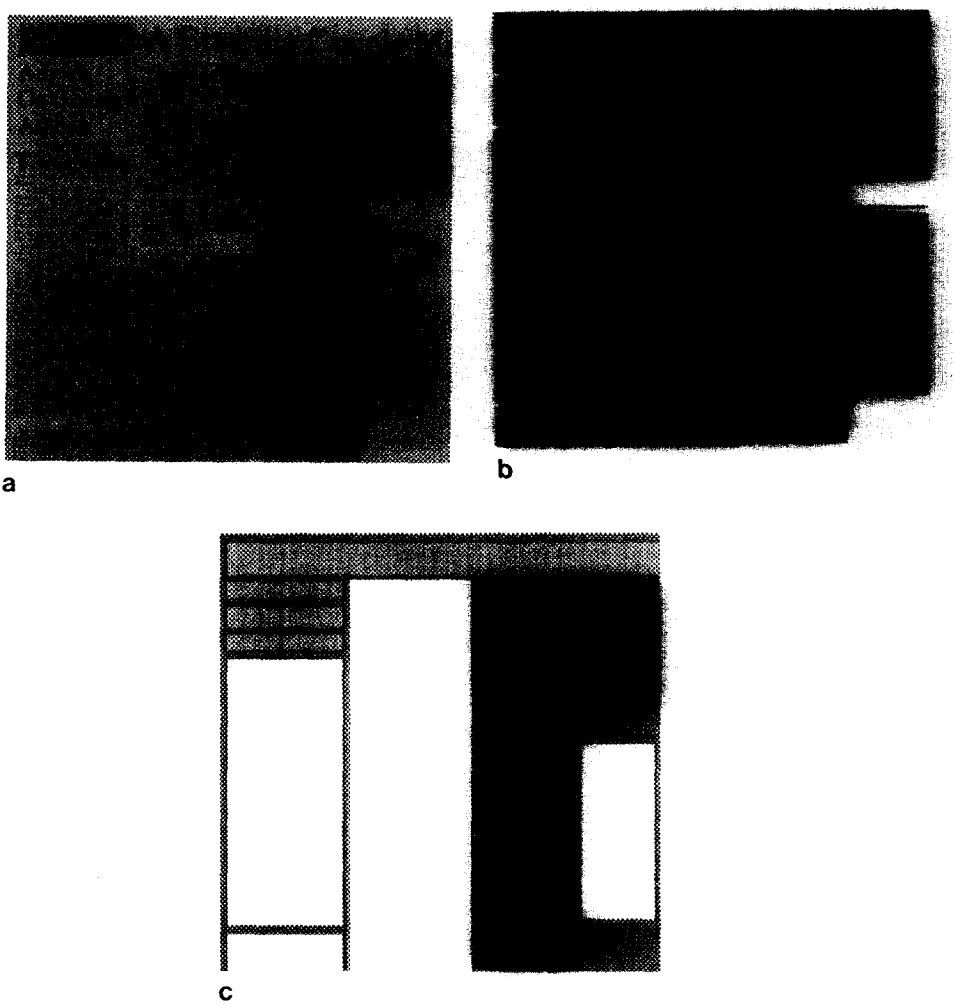

FIG. 13. Results with 100 ppi version of image shown in Fig. 1a: (a) original image; (b) block segmentation result; and (c) classified blocks.

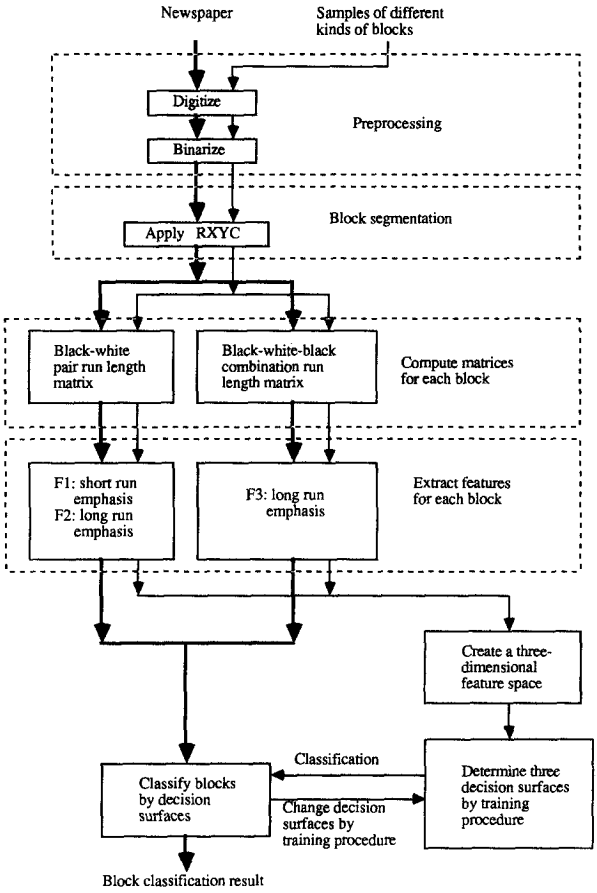

FIG. 14. Flow chart of a newspaper image recognition system.

containing only a single text line this method yields good classification results as shown in Fig. 9. The bigger the block is, the more accurate the features extracted are.

The features are not suitable for solid black lines. Blocks containing only a black line were classified incorrectly as halftones as shown in Figs. 9, 11d, and 12d. This drawback could be overcome by checking for solid black lines.

At 200 ppi classification performance is marginally better than at 100 ppi. However, segmentation at 200 ppi is better than at 100 ppi. The processing time at 100 ppi is four times faster than at 200 ppi, indicating linearity with the number of pixels.

ACKNOWLEDGMENTS

We are grateful to The Buffalo News, The New York Times, and USA Today for granting permission to reproduce portions of their newspaper pages.

REFERENCES

1. S. N. Srihari, Document image understanding, in *Proceedings, of IEEE Computer Society Fall Joint Computer Conference, Dallas, Nov. 1986*, pp. 87–96.
2. E. C. Arnold, *Modern Newspaper Design*, Harper & Row, New York, 1969.

3. P. W. Palumbo, P. Swaminathan, and S. N. Srihari, Document image binarization: Evaluation of algorithms, in *Proceedings, of SPIE Symposium on Applications of Digital Image Processing IX, 1986*, pp. 278–285.
4. K. Y. Wong, R. G. Casey, and F. M. Wahl, Document analysis system, *IBM J. Res. Develop.* **26**, No. 6, 1982, pp. 647–656.
5. G. Nagy, S. C. Seth, and S. D. Stoddard, Document analysis with an expert system, in *Proceedings, Pattern Recognition in Practice II, Amsterdam, June 19–21, 1985.*
6. S. N. Srihari and V. Govindaraju, *Analysis of Textural Images Using the Hough Transform*, Technical Report 88–08, Department of Computer Science, SUNY at Buffalo, April 1988; *Mach. Vision Appl.*, in press.
7. H. S. Baird, The skew angle of printed documents, in *Proceedings, SPSE 40th Conference and Symposium on Hybrid Imaging Systems, Rochester, New York, May 1987*, pp. 21–24.
8. S. N. Srihari, C.-H. Wang, P. W. Palumbo, and J. J. Hull, Recognizing address blocks on mail pieces: Specialized tools and problem-solving architecture, *AI Mag.* **8**, No. 4, 1987, 25–40.
9. M. M. Galloway, Texture analysis using gray level run lengths, *Comput. Graphics Image Process.* **4**, 1975, 172–179.
10. K. S. Fu, (Ed.), *Applications of Pattern Recognition*, CRC Press, Boca Raton, FL, 1982.