COMPUTER GRAPHICS AND IMAGE PROCESSING 20, 375–390 (1982)

NOTE

Block Segmentation and Text Extraction in Mixed Text/Image Documents

FRIEDRICH M. WAHL, KWAN Y. WONG, AND RICHARD G. CASEY

IBM Research Laboratory, San Jose, California 95193

Received January 18, 1982; revised February 4, 1982

The segmentation and classification of digitized printed documents into regions of text and images is a necessary first processing step in document analysis systems. It is shown that a constrained run length algorithm is well suited to partition most documents into areas of text lines, solid black lines, and rectangular boxes enclosing graphics and halftone images. During the processing these areas are labeled and meaningful features are calculated. By making use of the regular appearance of text lines as textured stripes, a linear adaptive classification scheme is constructed to discriminate text regions from others.

1. INTRODUCTION

Today, most information is saved, distributed, and presented on paper. For instance, paper is the primary medium for books, journals, and business correspondence. As it becomes cheaper and more practical to distribute information by electronic means, people need to convert information already stored on paper into a format suitable for the computers that handle this information. Data entry by operator keying is not only expensive but is also not capable of inputting documents with a mixture of text and images. We have started an effort to design and build an experimental Document Analysis System [1] that permits a user to extract appropriate information from printed documents. Components of the system will include image display and editing, automatic separation of a scanned document into regions of text and images, recognition of textual information, enhancement of images, and finally, analysis of the page layout structure to insert typographical instructions for reproduction of the document by typesetting systems. Some possible applications of the Document Analysis System are reediting existing manuals and books, converting submitted papers into a uniform format for journal publication, and efficient means for storing printed documents.

In this paper we deal with one of the components of the Document Analysis System, namely the block segmentation and the automatic classification of a digitized printed document into lines of text and regions of images, which include graphics, halftones, isolated horizontal and vertical lines. Some of the earlier approaches to this problem [2–4] require a knowledge of either the text character height and width or the line spacing in order to separate a document into text and nontext areas. Our proposed procedure uses some simple but powerful features of the separated blocks; a linear classifier, which can adapt itself to varying text character heights, is used to discriminate text and images. This method has been

0146-664X/82/120375-16$02.00/0

tested with a variety of printed documents from different sources. In this paper, we shall illustrate the performance using one fairly complex document as example.

2. BLOCK SEGMENTATION OF DIGITIZED DOCUMENTS

Block segmentation is a procedure that subdivides the area of a digitized document into subregions (blocks), each of which ideally contains only one type of data (text, graphic, halftone image, etc.). To achieve a certain adjacency of patterns belonging to the same type of data, the block segmentation derives an intermediate bitmap in which some white pixels have been changed into black pixels.

Consider, for instance, the document in Fig. 1a. It is composed of textlines, graphics, halftone images, and solid black lines. As can be seen in Fig. 1e, the block segmentation produces several compact black regions. These correspond to distinct data types on the original document, but are neither labeled nor classified up to this point. Note that characters imbedded in graphics or halftone images are considered to be parts of the surrounding data types, if no further segmentation is performed. In most cases, no problems with respect to subsequent processing arise from splitting one textline into two or more nonadjacent blocks. A block containing two different types of data will require additional, more sophisticated segmentation.

From Figs. 1a and 1e, it can be seen that the white background plays an important role in subdividing documents into meaningful blocks. A constrained run length algorithm (CRLA) has been proposed earlier [5, 6] to detect long vertical and horizontal white lines. Let us assume that white pixels are represented by binary 0's and black pixels by 1's. Within an arbitrary sequence of 0's and 1's the CRLA replaces 0's by 1's if the number of adjacent 0's is less than or equal to a certain predefined constraint C; for example, with $C = 2$ the sequence

$$0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,1\,0\,0\,1\,1\,0\,0\,0\,0$$

is mapped into the sequence

$$1\,1\,1\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0.$$

This one-dimensional bitstring operation is applied line-by-line as well as column-by-column to the two-dimensional input bitmap of the digitized document. The result of the horizontal CRLA applied to the document in Fig. 1a with $C_{hor} = 300$ is shown in Fig. 1b, while the vertical processing with $C_{ver} = 500$ is shown in Fig. 1c (the document in Fig. 1a consists of 2048×2400 pixels). These intermediate bitmaps are subsequently combined by a logical AND operation (Fig. 1d). This combination gives almost the desired final bitmap. However, some black regions representing textlines are interrupted by small gaps. Therefore, as a final processing step an additional nonlinear horizontal smoothing is performed by means of the CRLA (in this example with $C_{sm} = 30$; see Fig. 1e). This is especially necessary when documents with uniform character spacings are to be processed.

The choice of the parameters is rather uncritical. The values of C_{hor} and C_{ver} should be set to a number of pixels covering the length of long words, whereas C_{sm} should be set to cover a few character widths. The values used for the example in Fig. 1 led to fairly good results on a variety of documents from different origins. The one-dimensional bitstring manipulation CRLA can be applied in both the horizontal

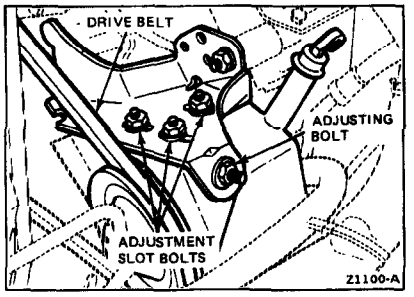

FIG. 23 – Adjustment Slot Bolts and Adjusting Bolt

A/C Idler Pulley Adjustment

In this adjustment, loosen the idler pulley pivot and adjusting bolts (Fig. 24). Then, adjust belt tension by installing a 3/8 to 1/2 inch adapter on the flex-handle. Insert the adapter into the pulley arm slot (Fig. 24). A long bar may also be used (Fig. 25).

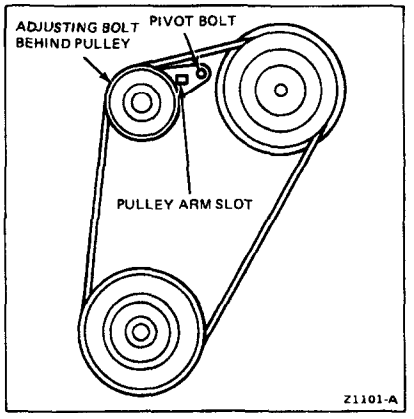

FIG. 24 – Idler Pulley Adjustment

CHECKING BELT TENSION

1. On some cars, it may be necessary to remove the fresh air pick-up tube to check belt tension. Remove pick-up tube as follows:

   a. Lift the release lever tab which is located on the air cleaner duct and disengage the tube from the duct (Fig. 26).

   b. Remove the front screw from the top of the tube, next to the radiator.

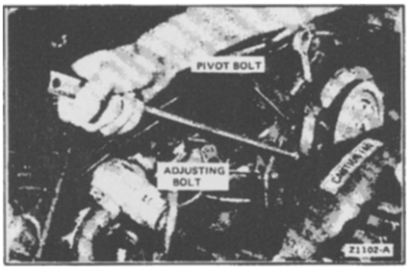

FIG. 25 – Applying Tension to Idler Pulley Arm

   c. Remove the rear screw located at the back of the tube which is mounted on the fender apron.

   d. Lift the tube from the back, making sure the bottom front latch clears the mounting hole (Fig. 26).

   e. For installation, reverse the above procedures. Make sure the bottom front latch is properly seated in the mounting hole.

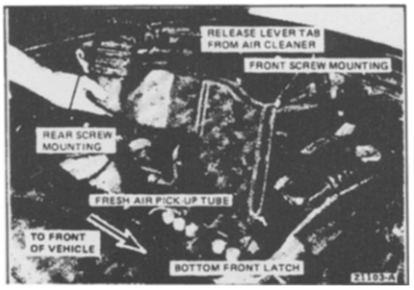

FIG. 26 – Removing and Installing Fresh Air Pick-Up Tube

2. Start engine and run the engine until it reaches normal operating temperature. Turn the engine off.

WARNING

The engine must not be running when checking or adjusting any drive belt.

FIGURE 1a

FIGURE 1b

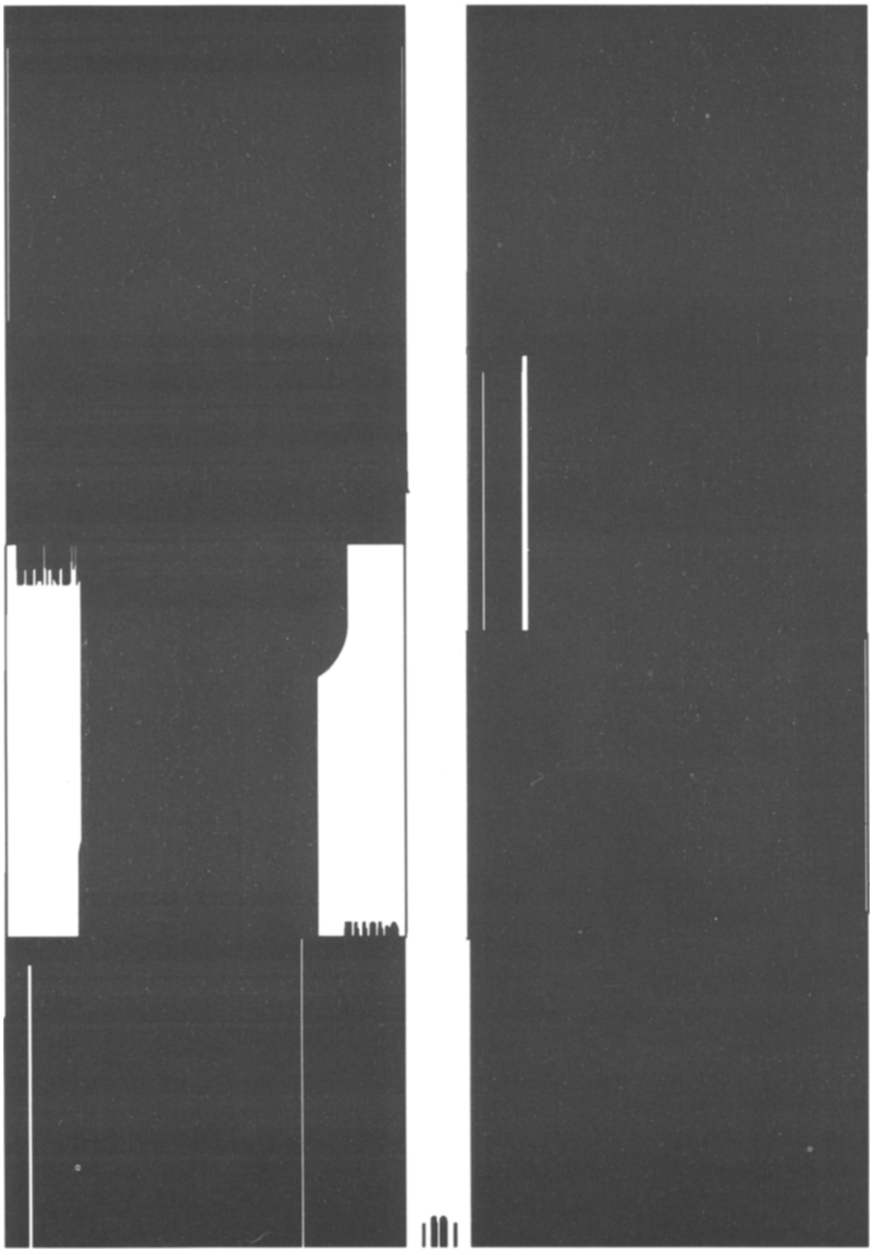

FIGURE 1c

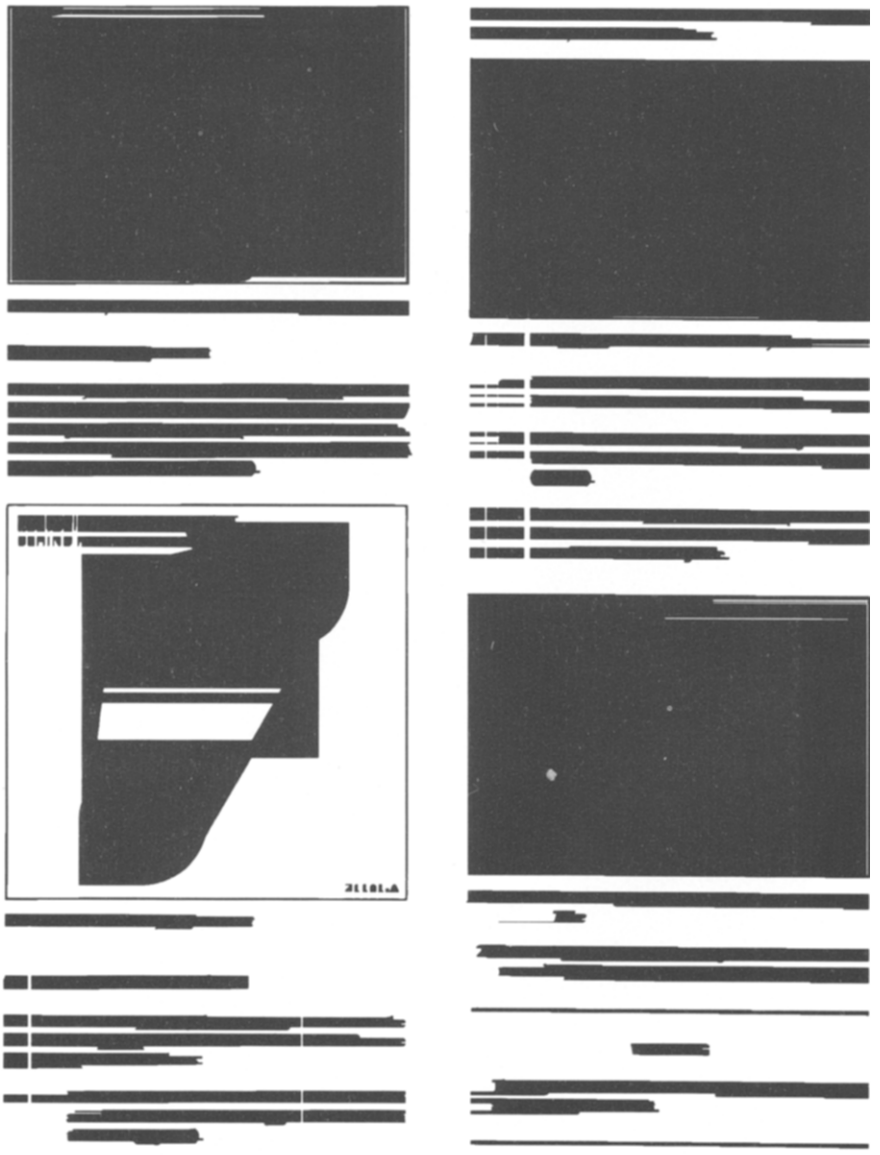

FIGURE 1d

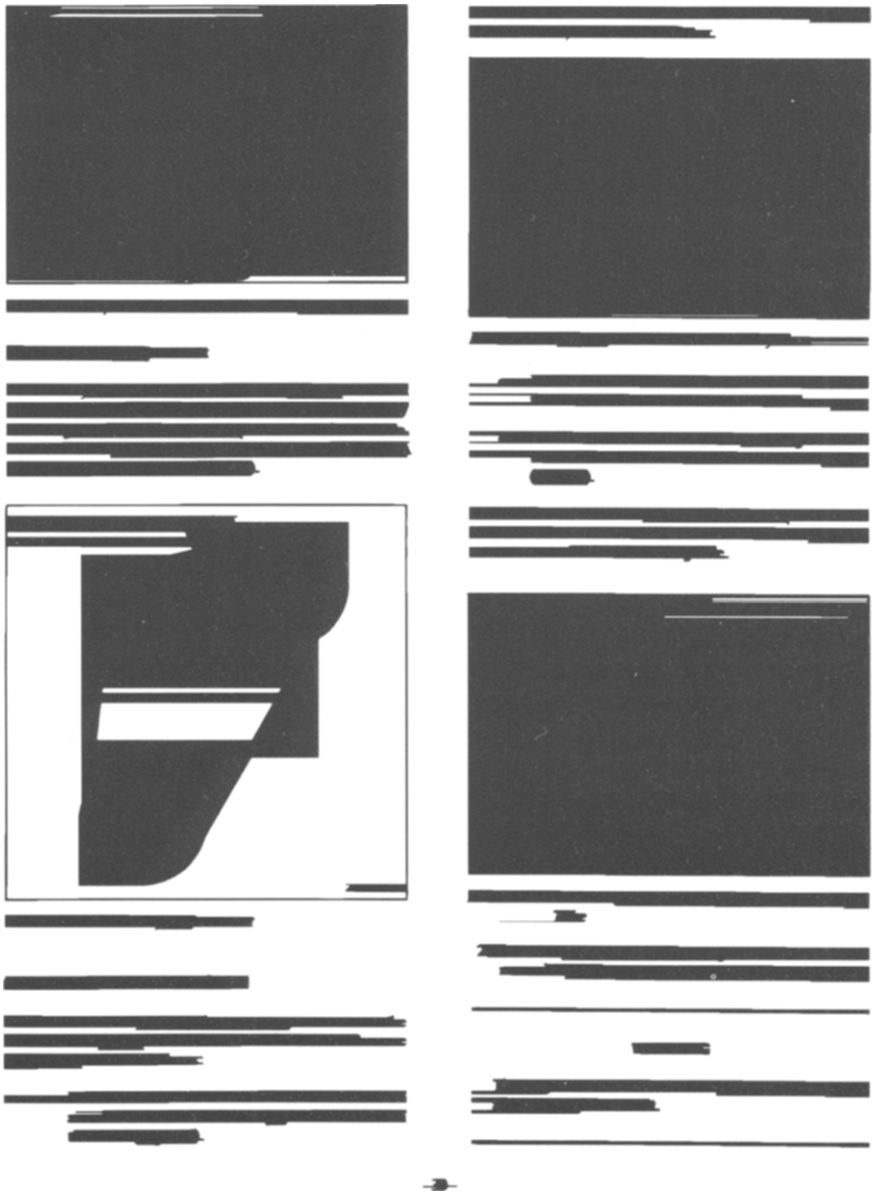

FIGURE 1e

FIG. 23 – Adjustment Slot Bolts and Adjusting Bolt

A/C Idler Pulley Adjustment

In this adjustment, loosen the idler pulley pivot and adjusting bolts (Fig. 24). Then, adjust belt tension by installing a 3/8 to 1/2 inch adapter on the flex-handle. Insert the adapter into the pulley arm slot (Fig. 24). A long bar may also be used (Fig. 25).

FIG. 24 – Idler Pulley Adjustment

CHECKING BELT TENSION

1. On some cars, it may be necessary to remove the fresh air pick-up tube to check belt tension. Remove pick-up tube as follows:

   a. Lift the release lever tab which is located on the air cleaner duct and disengage the tube from the duct (Fig. 26).

   b. Remove the front screw from the top of the tube, next to the radiator.

FIG. 25 – Applying Tension to Idler Pulley Arm

   c. Remove the rear screw located at the back of the tube which is mounted on the fender apron.

   d. Lift the tube from the back, making sure the bottom front latch clears the mounting hole (Fig. 26).

   e. For installation, reverse the above procedures. Make sure the bottom front latch is properly seated in the mounting hole.

FIG. 26 – Removing and Installing Fresh Air Pick-Up Tube

2. Start engine and run the engine until it reaches normal operating temperature. Turn the engine off.

WARNING

The engine must not be running when checking or adjusting any drive belt.

FIGURE 1f

and vertical direction very efficiently by means of a two-pass algorithm, which has been proposed in [7].

3. LABELING OF BLOCK SEGMENTS

To identify each block separately for subsequent feature extraction and classification, labels have to be assigned to different blocks.

This standard technique of digital image processing (see, e.g., [8]) is demonstrated with the binary pattern shown in Fig. 2. As the first line is scanned from left to right, the label 1 is assigned to the first black pixel. This label is "propagated" repeatedly, that is, subsequent adjacent black pixels are labeled with 1's and the first white pixel stops this "propagation." The next black pixel along the line is labeled with 2 and so are its adjacent neighbor black pixels. This is continued until the end of the first line is reached. For each black pixel in the second (and succeeding) lines, the neighborhood in the previously labeled line is examined along with the left neighborhood of the pixel. If a label has already been assigned to a pixel in the neighborhood, the same label is assigned to the current pixel. If a black pixel has no labeled neighborhood the next label not yet used is assigned to this pixel. This procedure is continued until the bottom line of the binary image is reached.

Upon completion of this algorithm, some adjacent black regions may be labeled differently. The existence of different adjacent labels can be detected while labeling, and can be used for updating the labels in a later step. As can be seen in the example of Fig. 2, labels 5, 6, and 7 are adjacent to label 2. Therefore these labels can be replaced by label 2, which yields four nonadjacent regions with labels ranging from 1 to 4. It will be shown next that it is not necessary to store the labeling results by means of integer maps. In our application, simultaneously with labeling, all information required for the subsequent processing can be extracted and stored in a table.

4. FEATURE EXTRACTION OF BLOCK SEGMENTS

To calculate features for the block classification, useful measurements can be performed simultaneously with labeling; for example, when assigning a label to a pixel, a counter for this particular label, designated BC, can be incremented by 1. When the labeling procedure is finished, the states of these counters represent the areas of the corresponding block segments. Similarly, coordinates of the surrounding rectangle of each block can be measured by a simple comparison of the coordinates x, y (indices) of the currently labeled pixel with four coordinate registers $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$, associated with each label. With each label assignment, the coordinate registers of the corresponding label are updated:

$$x_{\min} = \min(x, x_{\min}) \tag{1a}$$

$$x_{\max} = \max(x, x_{\max}) \tag{1b}$$

$$y_{\min} = \min(y, y_{\min}) \tag{1c}$$

$$y_{\max} = \max(y, y_{\max}). \tag{1d}$$

After the labeling procedure has finished, the final states of the coordinate registers provide information about position and size of each block segment within the document.

FIGURE 2a

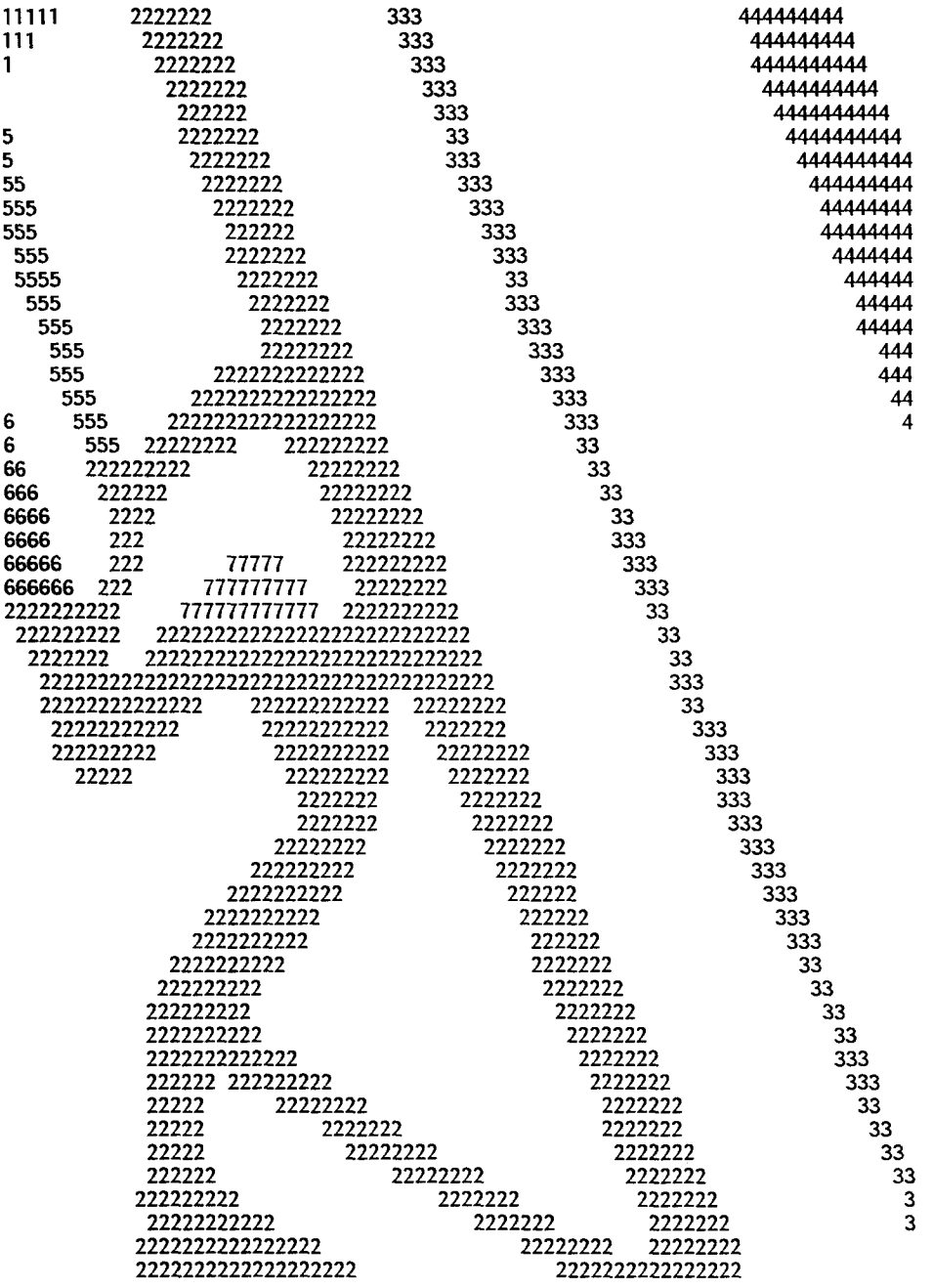

FIGURE 2b

Measurements applied directly to the original data are useful as well as measurements performed on the bitmap of the block segmentation result. For example, the number of black pixels of the original data within each block can be calculated by using an additional counter DC for each label. Whenever a label assignment occurs the associated *DC* is incremented by 1 if and only if the pixel of the original input bitmap at this location is black. Another useful quantity is the number of horizontal white–black transitions *TC* of the original data within each block segment, which can be counted in a similar way.

In Table 1 we show the result of the proposed measurements applied to the document in Fig. 1a and to its corresponding block segmentation result in Fig. 1e. In this table each block segment is characterized by one line. Labeling and measurements have been applied to the bitmap of Figs. 1a from bottom to top. Therefore, the first line in this table corresponds to the lowest pattern on the document (namely, the page number enclosed by dots). Note also that instead of the maximum x, y coordinates of the surrounding rectangles of the blocks, the x, y sizes Δx, Δy have been stored in this table. Furthermore, the measurements derived from adjacent regions have been merged simultaneously with updating labels; no additional measurements are necessary in order to do this.

Based on the measurements described above, the following features can be calculated:

1. the height of each block segment: $H = \Delta y$;

TABLE 1

Part of the Processing Results of the Document in Fig. 1.

BC	x_{min}	Δx	y_{min}	Δy	DC	TC	Class
702	995	68	2341	23	302	76	1
6089	1090	771	2266	13	5608	170	2
6387	307	265	2245	32	1142	396	1
15396	307	657	2208	31	3118	1005	1
9341	1090	366	2184	31	2469	580	1
16706	185	779	2171	24	3489	1070	1
19447	1090	771	2147	35	5181	1110	1
9244	185	385	2098	31	1780	528	1
3244	1401	152	2077	23	1587	303	1
18502	185	779	2060	32	4112	1183	1
17592	185	779	2024	30	3613	1018	1
5667	1091	770	2008	13	5394	72	2
12300	185	474	1947	28	4751	735	1
19318	1144	717	1923	35	4436	1155	1
19252	1101	760	1887	32	3981	1059	1
11101	188	483	1830	29	2713	756	1
1258	1144	171	1821	22	434	123	1
20200	1082	779	1782	34	4349	1229	1
276147	188	780	1037	766	47742	8738	3
418268	1084	780	1209	546	195328	59906	3
10935	1088	503	1117	30	2139	648	1
18370	1088	773	1080	31	3406	996	1
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Note. Columns 1–7: results of the measurements performed simultaneously with labeling; last column: text classification result.

2. the eccentricity of the surrounding rectangles of the blocks:

$$E = \Delta x/\Delta y; \tag{2}$$

3. the ratio of the number of block pixels to the area of the surrounding rectangle:

$$S = BC/(\Delta x\, \Delta y); \tag{3}$$

(which reflects, if S is close to 1, that a block segment has an approximate rectangular shape);

4. the mean horizontal length of black runs of the original data within each block:

$$R = DC/TC. \tag{4}$$

These features are used for the classification of block segments described next.

5. TEXT DISCRIMINATION

Text is the predominating data type in documents handled in a typical office environment. Because textlines can be considered as textured stripes of approximately constant height H and mean length of black runs R, the text blocks should cluster with respect to these features. This can be illustrated by plotting block segments in the RH plane. In Table 2 we show an example using the information in Table 1 which has been extracted from the document in Fig. 1 (note that the scale of this diagram is highly nonlinear). Each table entry is equal to the number of block segments in the corresponding range of R and H. Thus, such a plot can be considered as a two-dimensional histogram. It can be seen that the text lines of the document in Fig. 1 form a clustered population within the range $20 < H < 35$ and $2 < R < 8$. The two solid black lines in the lower right part of the original document are represented at high R and low H values in the RH plane, whereas the graphic and halftone images are represented at high values of H.

Of course, the mean values $\overline{H}$ and $\overline{R}$ for the text cluster may vary within a certain range for different types of documents depending on character size and character font. Furthermore, the standard deviations of the text cluster $\sigma(H)$ and $\sigma(R)$ may vary depending whether a document is printed in a single font or with multiple fonts and different character sizes. To this end it seems desirable to apply a discrimination scheme for text, which adjusts the decision boundaries within a certain range according to the properties of the predominant text on a particular document. A three-step, self-adjusting classifier for text is proposed.

Step 1

To estimate the means $\overline{H}, \overline{R}$ and the standard deviations $\sigma(H), \sigma(R)$ of a supposed text cluster, blocks satisfying the following intuitive constraints are selected:

$$H/R > C_1 \tag{5}$$

$$H < C_2 \tag{6}$$

$$E > C_3 \tag{7}$$

$$S > C_4. \tag{8}$$

TABLE 2
Table Histogram—Block Height versus Mean Length of Black Runs

Block Height H	Mean Run Length of Black Pixels R														
	1	2	3	4	6	8	10	20	30	40	60	80	100	300	1000
5															
10															
15										1		1			
20															
25			4		1										
30		1	7		1										
35		2	15	3											
40															
45															
50															
55															
60															
65															
70															
75															
80															
85															
90															
95															
100															
200															
300															
400															
500															
600			2		1										
700															
800					1										
900															
1000															
2000															
3000															
4000															

Blocks satisfying these constraints are highly likely to be text lines. For the document in Fig. 1, Table 3 shows the qualifying block data in the RH plane using the constraint parameters values $C_1 = 4$, $C_2 = 100$, $C_3 = 10$, and $C_4 = 0.5$. By comparing Table 3 with Table 2 it can be seen that some blocks—in this example graphics, halftone images, solid black lines, and short text lines—are neglected for the estimation of the text line cluster properties. In this example, 29 blocks of the original 40 blocks constitute the cluster; $\bar{H} = 30.6$, $\bar{R} = 3.6$, $\sigma(H) = 2.33$, and $\sigma(R) = 0.66$.

Step 2

Using the estimated text cluster properties $\bar{H}$, $\bar{R}$, $\sigma(H)$, and $\sigma(R)$ the following tests ascertain whether the data constitute a reasonable cluster at all:

$$\text{number } N \text{ of blocks in the cluster} > C_{11} \tag{9}$$

$$\text{ratio } N \text{ to total number of blocks} > C_{12} \tag{10}$$

$$\bar{R} < C_{13} \tag{11}$$

$$\bar{H} < C_{14} \tag{12}$$

$$\sigma(H) < C_{15} \tag{13}$$

$$\sigma(R) < C_{16} \tag{14}$$

$$\sigma(H)/\bar{H} < C_{17} \tag{15}$$

$$\sigma(R)/\bar{R} < C_{18}. \tag{16}$$

TABLE 3

Table Histogram—Block Height versus Mean Length of Black Runs

Block Height H	Mean Run Length of Black Pixels R														
	1	2	3	4	6	8	10	20	30	40	60	80	100	300	1000
5															
10															
15															
20															
25			2												
30		1	7		1										
35		1	14	3											
40															
45															
50															
55															
60															
65															
70															
75															
80															
85															
90															
95															
100															
200															
300															
400															
500															
600															
700															
800															
900															
1000															
2000															
3000															
4000															

Note. Table 3 is the same as Table 2, but with applied feature constraints.

If all these conditions are met by the properties of the hypothesized cluster, then it is decided that a text cluster exists and the block discrimination is performed in the next step. With the values $C_{11} = 10$, $C_{12} = 0.5$, $C_{13} = 8$, $C_{14} = 60$, $C_{15} = 5$, $C_{16} = 2$, $C_{17} = 0.5$, and $C_{18} = 0.5$, the cluster generated by the features of the document in Fig. 1 meets these conditions.

Step 3

A variable, linear, separable classification scheme is used to assign the following four classes to the blocks.

Class 1. Text:

$$R < C_{21} * \bar{R} \quad \text{and} \quad H < C_{22} * \bar{H}. \tag{17}$$

Class 2. Horizontal solid black lines:

$$R > C_{21} * \bar{R} \quad \text{and} \quad H < C_{22} * \bar{H}. \tag{18}$$

Class 3. Graphic and halftone images:

$$E > 1/C_{23} \quad \text{and} \quad H > C_{22} * \bar{H}. \tag{19}$$

Class 4. Vertical solid black lines:

$$E < 1/C_{23} \quad \text{and} \quad H > C_{22} * \bar{H}. \tag{20}$$

The classification result with $C_{21} = 3$, $C_{22} = 3$, and $C_{23} = 5$ for the document shown in Fig. 1 is shown in Table 1, last column. In Fig. 1f we show the document when regions classified as 2 and 3 are removed.

6. CONCLUSION

Because document analysis involves the processing of tremendous amounts of data (typically 5 million pixels/document), one objective of the image segmentation of the Document Analysis System [1] mentioned in Section 1 is to extract the usually predominant text regions of a document with a fairly low computational load. The output is a table that supplies this system with highly condensed information such as data types, block positions, and sizes. In this paper it has been shown that this requirement can be met by making use of the typical regular structure of text lines. The proposed method has been tested with a variety of printed documents from different origins with one common set of parameters. Misclassifications have been observed very rarely. They happen when several text lines are linked together by the block segmentation, due to small line-to-line spacings, and thus are discriminated as class 3 blocks.

In order to classify different data types within class 3, a second discrimination based on two recently proposed shape factors [9] is used in the Document Analysis System. Using the method outlined in this paper together with this higher sophisticated pattern analysis of [9] it is possible to perform a complete decomposition of most documents into regions of text, graphics, halftone images, and long horizontal and vertical solid black lines.

Although it has been shown that the text discrimination can be done efficiently with promising success, the method outlined in this paper should in the future be evaluated in terms of its statistical reliability. Moreover, the involved parameters, which up to now have been chosen empirically, should be optimized by statistical means.

REFERENCES

1. K. Y. Wong, R. G. Casey, and F. M. Wahl, Document Analysis System, *IBM J. Res. Develop.*, submitted.
2. G. Nagy, Preliminary investigation of techniques for automated reading of unformatted text, *Commun. ACM* **11**, 1968, 480–487.
3. E. G. Johnston, Printed text discrimination, *Computer Graphics and Image Processing* **3**, 1974, 83–89.
4. W. Scherl, F. Wahl, and H. Fuchsberger, Automatic separation of text, graphic and picture segments in printed material. In *Pattern Recognition in Practice* (E. S. Gelsema and L. N. Kanal, Eds., pp. 213–221, North-Holland, Amsterdam, 1980.
5. F. Wahl, L. Abele, and W. Scherl, Merkmale fuer die Segmentation von Dokumenten zur automatischen Textverarbeitung, Proceedings, 4th DAGM Symposium, Hamburg, FRG, Springer-Verlag, New York/Berlin, 1981.
6. L. Abele, F. Wahl, and W. Scherl, Procedures for an Automatic Segmentation of Text Graphic and Halftone Regions in Documents, Proceedings, 2nd Scandinavian Conference on Image Analysis, Helsinki, 1981.
7. F. M. Wahl and K. Y. Wong, An Efficient Method for Running a Constrained Run Length Algorithm (CRLA) in Vertical and Horizontal Direction on Binary Image Data, submitted Pat. Discl. IBM SA8-81-0310.
8. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, pp. 347–348, Academic Press, New York, 1976.
9. F. M. Wahl, A new distance mapping and its use for shape measurement on binary patterns, *IBM Res. Rep.* RJ3361, also submitted and accepted for publication in *Computer Graphics and Image Processing.*