A HYBRID DOCUMENT SEGMENTATION METHOD FOR TURKISH NEWSPAPERS

by

M. Feridun AKTAŞ

BS. in E.C., İstanbul Technical University, 1994

Submitted to the Institute for graduate Studies in
Science and Engineering in partial fulfillment of
The requirements for the degree of
Master of Science
in
Electrical and Electronic Engineering

Boğaziçi University

1998

ACKNOWLEDGEMENTS

I am very thankful to Prof. Dr. Bülent Sankur who assumed the direction of my thesis, for his understanding, research assistance, and comments on my work.

I would like to thank the thesis examining comitee, Assoc. Prof. Dr. Yağmur Denizhan, Assoc. Prof. Dr. Lale Akarun and guest Dr. Valery V. Starovoitov for their careful examining and kindly guidance to assure the correctness of my thesis. I also would like to thank my research partner Ufuk Barış, for his great partnership on many step of the project.

I would like to thank all other those who contributed to the successful outcome of this thesis by their support, their advice or their friendly presence.

A HYBRID DOCUMENT SEGMENTATION METHOD FOR TURKISH NEWSPAPERS

ABSTRACT

Today most of the information is conveyed in the form of printed papers. The range of them varies from the newspapers to formal correspondence letters, from banking documents to envelopes etc. The evaluation of document processing systems made it possible to transfer this information from the printed materials to the electronic media. To transfer and archive this information some compression and conversion techniques are used. These techniques extract the document components and process them regarding the content type. Documents are mainly composed of text and image blocks, line and drawings.

This thesis is focused on the extraction of document image components for further processing. This operation is known as document analysis. Several document analysis techniques are reviewed and one of them, Recursive X - Y Cut, is modified and applied to the Turkish newspapers. This method recursively analyze the horizontal and vertical projection profile of documents and locate the most appropriate cut (horizontal or vertical) over the documents. The process recursively continues until the smallest desired blocks are found or not any appropriate cut place exists on the document. At the result, blocks that mostly contain single type of document component, are extracted. The blocks, that contains several type of document components, are fed to another segmentation algorithm.

TÜRKÇE GAZETELER İÇİN KARMA BİR BELGE BÖLÜTLEME METHODU

ÖZET

Günümüzde bilgilerin büyük bir çoğunluğu kağıtlara basılı olarak bulunmaktadır. Bu belgeler gazete sayfasından resmi yazışmalara, banka makbuzlarından mektup zarflarına kadar değişen bir yelpazede yer almaktadır. Belge işleme sistemlerindeki gelişmeler kağıtlara basılı olan bu bilgilerin elektronik ortamlara taşınmasına olanak vermiştir. Bu bilgileri elektronik ortama taşımak için bazı tanıma, sıkıştırma ve dönüştürme teknikleri kullanılmaktadır. Bu teknikler belgelerin bileşenlerini bulmakta ve onları içeriklerine göre farklı şekillerde işleme tabi tutmaktadır. Belgeler temelde yazı ve resim blokları, çizgiler ve çizimlerden oluşmaktadır.

Bu tez, belgelerin bileşenlerini, sonradan işlenmek amacıyla, bulma işlemi üzerinde yoğunlaşmaktadır. Yapılan bu işleme belge anlama adı verilir. Çeşitli belge anlama yöntemleri incelenmiş ve bunlardan biri olan Ardışıl Yatay – Dikey Kesmeler yöntemi iyileştirilip Türkçe gazeteler üzerinde uygulanmıştır. Bu yöntemde belgenin yatay ve düşey izdüşüm eğrileri incelenmekte ve belgenin uygun bir yerine (yatayda veya düşeyde) kesme yerleştirilmektedir. İşlem, ardışıl olarak istenen en küçük boyuttaki blok bulununcaya ya da kesme yerleştirilecek uygunlukta bir yer kalmayıncaya kadar devam etmektedir. Sonuçta tek tür belge bileşeni içeren bloklar elde edilmektedir. Birden fazla belge bileşeni içeren bloklar başka bir bölütleme algoritmasına gönderilmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

S_i	Text stripe$_i$
Xul_i	X axis value or up-left coordinate of text stripe$_i$
Xul_i	X axis value or low-right coordinate of text stripe$_i$
Yul_i	Y axis value for up-left coordinate of text stripe$_i$
Ylr_i	Y axis value for low-right coordinate of text stripe$_i$
P,Q	Column interval
LS_i	Line segment$_i$
$\mathbf{lp_f(i)}$	Lowpass filter output for function **f(i)**
V(i)	Valley at i^{th} position
$\mathbf{P_r(i)}$	Right peak of **V(i)**
$\mathbf{P_l(i)}$	Left peak of **V(i)**
$\mathbf{V_d(i)}$	Valley depth of **V(i)**
$\mathbf{V_s(i)}$	Valley steepness of **V(i)**
$\mathbf{D_{th}(i)}$	Threshold value for valley width of **V(i)**
$\mathbf{V_w(i)}$	Valley width of **V(i)**
$\mathbf{V_b(i)}$	Valley base of **V(i)**
$\mathbf{max_p}$	Maximum peak value
th_1	Threshold for block height
th_2	Threshold for block width
gn	Group number

1. INTRODUCTION

1.1. Document Processing

Printed and written documents are the most common medium of information transmission. The way that information conveyed by document varies in a great scale. From the newspapers to formal correspondence letters, from banking documents to engineering drawings and from envelope contents to technical journals etc. The acquisition of knowledge from such documents by an information system can involve an extensive amount of handcrafting. Such handcrafting is time consuming and severely limit the application of information systems. Actually it is a bottleneck of information systems. Therefore the need for automation of this process is very urgent. With the development of high-power computing systems several new document processing techniques have become feasible. The goal of these developments is to extract and understand information from document images more efficiently, more precisely and faster.

A document image is a visual representation of a printed page such as a newspaper, a journal article page, a facsimile cover page, a technical document, an office letter, an envelope etc. Typically it consists of blocks of text, i.e. letters, words and sentences that are interspersed with tables and figures. The figures can be symbolic icons, gray-level images, line drawings or maps. A digital document image is a two-dimensional representation of a document image obtained by optically scanning and digitizing a hardcopy document. A document processing system runs over this digital document image.

The way that a document processing system works is highly dependent on the type of handled document. Mainly, we can group these documents into three classes:

1. The ones with mixed image and text blocks, such as newspapers. The size and location of these blocks greatly differ from one issue to another. And the font

size of different text lines - such as titles, headers and paragraph lines - within a page also differ in great extent. These features of newspaper pages impose big restrictions on both segmentation and classification techniques. For example the system cannot extract paragraph blocks using a predefined location knowledge because they change in every issue. Systems for newspapers, therefore, should use global formatting knowledge for segmentation, such as white gaps between columns and ones between black pixels within different block types. Almost none of segmentation methods performs perfect segmentation on newspaper pages. The solution comes with hybrid systems that combine the complimentary features of different techniques.

2. Another type of document is envelope. In the envelope processing systems, the aim is to find the address regions and to extract the sender and recipient addresses from these regions. These systems highly increase the speed of mail distribution process. The address blocks are located almost in same position. Therefore extracting these regions is relatively simple. But the bottleneck for envelope processing systems arises in recognition phase. Addresses are mostly hand-written on envelopes. Therefore recognizing hand-writings is quite more complicated than printed characters. To solve this problem very sophisticated recognition techniques have been developed.

3. The third document type is form. Form processing systems are widely used to store and process the documents in business area. The basic property of forms is having a fixed block structure. Banking cheques and forms, company specific documents such as ordering forms are example of this document type. The picture blocks, like company logos, and text blocks are located in exactly the same position for a single type of form and they are mostly rectangular. Utilizing these properties of forms causes developing some techniques solely for forms. Form Definition Language (FDL) is an example to these systems. It works as a script that uses a priori known block relations and generates a coded language showing these relations.

A complete document processing system for above described documents is illustrated in Figure 1.1. The principal concepts of such system is as follows [1]:

- A concrete document is considered to have two structure: the geometric structure (or layout) and the logical structure.
- Document processing is divided into two phases: document analysis and document understanding.
- Extraction of the geometric structure from a document is defined as document analysis; mapping the geometric structure into a logical structure is defined as document understanding. Once the logical structure has been captured, artificial intelligence or other techniques can decode its meaning.
- But in some cases the boundary between the two phases just described is not clear. For example the logical structures of bank cheques may also be found during an analysis by knowledge rules.

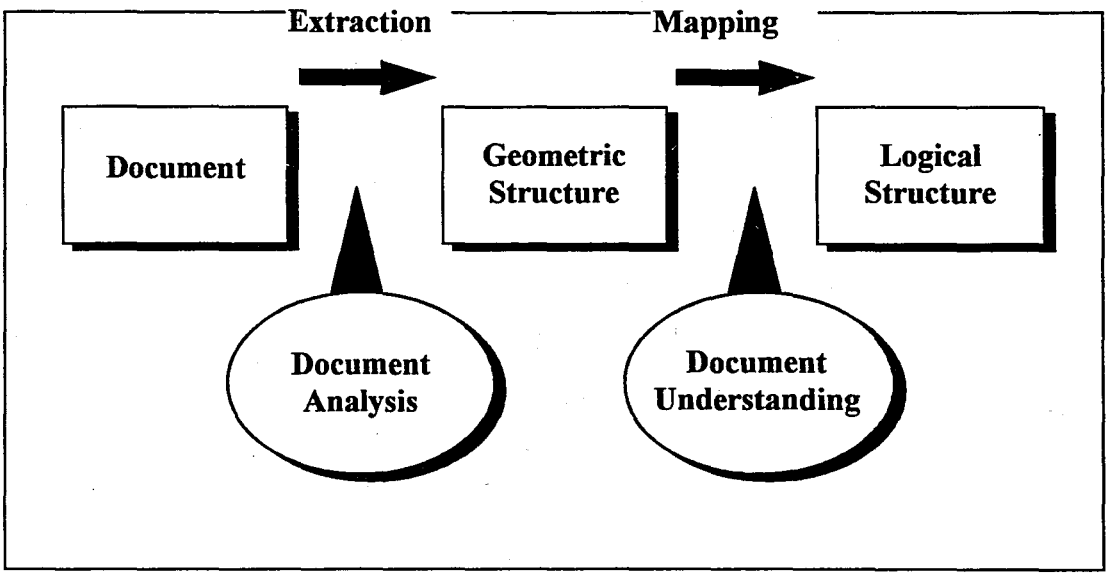


FIGURE 1.1. Basic model for document processing [1]

The methods used in this project will be used in the document analysis phase of a newspaper processing system. Therefore it uses the properties of newspapers, specifically Turkish newspapers.

1.2. Newspaper Analysis And Archiving System

The need for accessing and searching the older newspaper issues is important due to legal obligations, news research, and anthological reasons. The older issues before the computerization are still, archived in microfilms or volumes. But this method does not facilitate people requirements. On the other hand for press companies, transferring and accessing both their own and other companies' issues over an electronic media, is of vital importance. This brings some major advantages. The physical space requirement for older issues will be drastically decreased. Instead of big volumes of newspapers, some type of electronic media, such as compact discs (CD-ROM) or magnetic tapes, can be used. Using this facility, one can store hundreds of newspaper issues onto a single CD-ROM by the help of compression techniques. Because of the fact that, these media can be replicated easily, distributing such data also becomes faster and easier.

Another bottleneck for current archiving system is search restrictions. Since there is not an indexing system for newspaper, such as ones in the technical journals, searching an article or news in the older issues can just be done by eyes. This is quite slow and tiring for people. But, once the older issues have been transferred onto an electronic media, accessing and searching them will be easier via computer networks. Using internetworking technologies remote access to the newspaper archives can be achieved and by the help of well-developed search engines, faster and more precise search results can be obtained. with respect to visual search.

Designing a document processing system for newspapers is quite complex. A lot of difficulties emerging from the layout and content of the newspapers are imposed on such systems. The greatest bottleneck for newspaper processing systems is the page content i.e. photographs, text blocks, drawings and lines often exist within one page. Therefore an efficient system should classify each of these components successfully. Another problem is block locations, which vary in great scale even between the different issues of same newspaper. The location of blocks forms the page layout. The blocks within a page are not

always aligned and do not always constitute a regular column structure. Two samples for this case are illustrated in Figure 1.2.

Mustafa Kemal'i anlamak, aldatmak değil...

RECEP Bilginer'in söylediklerini dinlerken, yazdıklarını okurken, aklımıza Halim Yağcıoğlu'nun "Atatürk'ten Son Mektup" şiiri geldi...

Yağcıoğlu, o şiirinin sonunda şöyle der:

"Arayı kapatmak istiyorum uygar uluslarla,
Bilime, sanata varımız rezil dalkavuklukla.
Bu vatan, bu canım vatan sizden çalışmak ister,
Paydos övünmeye, paydos avunmaya yeter, yeter!

Mustafa Kemal'i anlamak, aldatmak değil,
Mustafa Kemal ülküsü sadece söz değil!"

★★★

BİR süreden beri, Ankara Devlet Tiyatrosu "Büyük Sahnesi"nde Recep Bilginer'in bir oyunu oynanıyor: "Savaştan Barışa, Aşıktan Kavgaya Mustafa Kemal"

Oyunda Atatürk'ün düşmanlarla, yobazlarla, hatta dava arkadaşlarıyla nasıl mücadele ettiği, dramı yanlarıyla çıkarabilme cesaretini gösterdiğim için.

HASAN PULUR

Gerçekten böyle bir piyes yazmak, cesaret işidir. Risklidir.

Ülkemizde Atatürk karşıtları var. Aşırı sağcılar, aşırı solcuları. Bir de Atatürk'ü artık geride kaldı, şimdi yeni bir dünya düzeni kuruluyor, diye umursamayanlar.

★★★

AMA ya Atatürkçüler, Atatürkçü düşünceye bağlı kalanlar, O'nun kurduğu demokratik, laik Cumhuriyet'e sahiplenler? Ülkenin ve rejimin tehlikeye düşme olasılığı karşısında düzlüğe çıkma olanağını, O'nun ilkelerinde arayanlar!

Onlar nasıl karşılayacaklar oyunu? Doğrusu şu ya; piyes, Atatürk Kültür Merkezi'nce kitaplaştırıldıktan ve Devlet Tiyatroları'nda provalara başlandıktan sonra, bu konuda gördüğüm umursamazlık, beni çok düşündürdü.

İşte bankalara gönderilen mektup

FIGURE 1.2. Examples of disturbed column structure.

This kind of layouts cannot be handled by a single segmentation method, which will be explained in Section 1.3, and often requires some hybrid techniques. Font sizing in text blocks, which vary from large headlines to little figure captions, is another problem. Grouping these characters into text blocks strictly depends on the letter size and the inter-character gap. Because, some threshold values and alignment issues are determined by these quantities. This fact should also been taken into consideration by any document processing method.

All these factors have caused the development of application specific document processing systems. The GARİLDİ (Gazete Arşivi ve İletişimi Dizgesi) project is developed specifically for the processing of Turkish newspapers. The documents are stored into a database by following the steps in Figure 1.3.

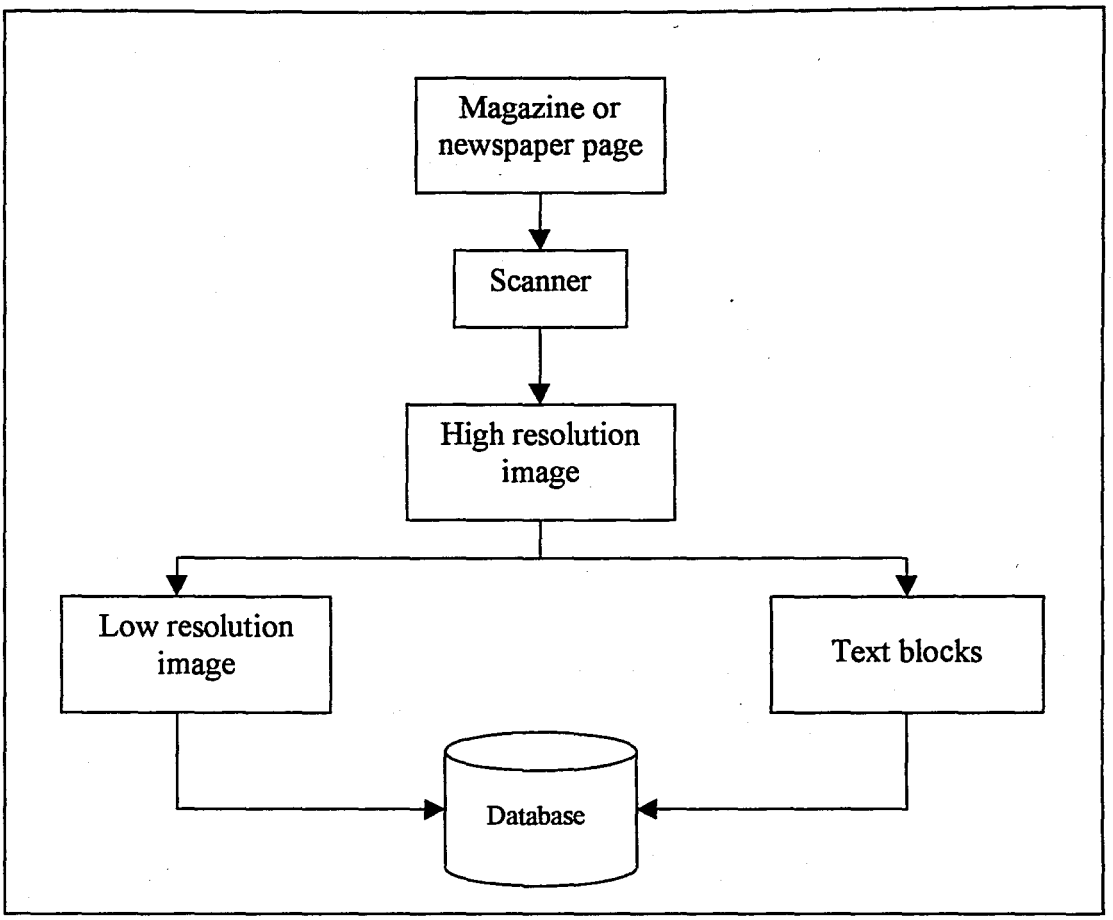


Figure 1.3. Document processing flow in GARILDI project

1.3. Document Analysis

Document analysis is the extraction of the constituting blocks of a page. This task can be divided into two phase [2]. Phase 1 consists of block segmentation where the document is decomposed into several rectangular blocks. Each block is a homogenous entity containing one of the following: text of a uniform font, a picture, a diagram, or a table. The result of the first phase is a set of blocks with relevant properties. A textual block is associated with its font type, style and size; a table might be associated with the number of columns and rows, etc. Phase 2 consists of block classification. The result of the

second phase is an assignment of labels (title, regular text, picture, table, drawing etc.) to all the blocks using properties of individual blocks from the first phase, as well as spatial layout rules. In our study a segmentation method for the first phase of document analysis has been developed.

The aim of the segmentation process is to divide the document image into regions that contain document constituents, that is specifically text, horizontal and vertical lines, graphics or halftone pictures. There are some restrictions in the segmentation process.

1. Regions should be mostly rectangular. Because most of the segmentation algorithms (such as RXYC, connected component method etc.) assumes that text, drawing or photograph regions are of rectangular shape. In fact printed documents such as scientific papers and newspapers are composed of rectangular regions. But this rule is generally violated in newspapers as given in Figure 1.2. On the other hand some techniques, mostly based on texture analysis, are developed to deal with non-rectangular blocks.
2. The scale of attempted separation between text and graphics must be chosen carefully. That is, a fine grained analysis can be deployed to separate textual annotations from graphical diagrams or text line and graph can be seen as a whole graphic block. If our aim is just to feed the text regions into the OCR process then locating text regions within the document would be sufficient. But if the analysed documents are CAD - CAM documents then more attention should be paid on graphics. Another concern related with separation is the size of text block, that is, whether text regions be constructed in line segments, paragraphs or collections of paragraphs within a text column.
3. Available prior information (such as document layout and font types and sizes) facilitates the process. It is faster and easier to deal with a document of fixed-sized font because some parameters such as thresholds in smearing algorithm would remain constant for whole document and no adaptive techniques would be required. In layout extraction process prior layout information of document is widely used especially in banking documents, cheques, invoices and such fixed-form documents.

To overcome these restrictions various approaches have been developed for ðage segmentation. These approaches can be categorized into three broad categories: top-down (model-driven), bottom-up (data-driven) and hybrid systems. The goal is the same : finding homogenous regions i.e. text, horizontal and vertical lines, photographs and drawings.

1.3.1. Top-Down Methods

A top-down segmentation method typically starts by hypothesizing a series of interpretations (e.g. that the page has a header above columns, there are white bands between blocks) at high level and attempts to verify each by a search of a tree or structure of implied hypotheses at lower levels of detail, finally going down until the lowest level is reached (e.g. characters, lines or text blocks) [3]. These methods divides documents into a set of blocks recursively until a threshold or minimum is reached. Most popular top-down segmentation technique known as the *recursive X-Y cut (RXYC).* The RXYC technique will be explained in Section 2.2.

1.3.2. Bottom-Up Methods

Bottom-up strategies start by merging evidence at the lowest level of detail (e.g., forming words from characters, lines from words) and then rise, merging words into lines, lines into columns, etc. until entire page is completely assembled. The merging process utilizes the specific features of each block type at each hierarchical level. For example character spacing at the beginning and then neighborhood between word blocks etc. These features are investigated in different studies and some segmentation methods are developed for specific document types and goals. Examples of this method includes run length smearing algorithm, connected component methods, line merging, neighborhood line density methods and others.

1.3.3. Hybrid Methods

Some documents cannot be segmented by neither bottom-up nor top-down methods. In these cases, hybrid methods are used. In a hybrid method a bottom-up method follows top-down method or vice versa. As stated in the article of Pavlidis and Zhou [3], H. Kida, O. Iwaki and K.Kawada, developed a method in 1986, which roughly segments an image by a sequence of horizontal and vertical projections, then uses connectivity analysis to complete the segmentation. Similarly, Akiyama and Masuda deployed a hybrid technique in 1985 which uses the skew-correction followed by the projection profile method, but in addition they attempt to cope with variable character size and line spacing.

1.REVIEW OF SEGMENTATION TECHNIQUES

The various segmentation techniques in the literature are comperatively reviewed in the sequel with a view to be used in newspaper documents.

1.1. The Run Length Smearing Algorithm

RLSA is one of the first proposed document segmentation techniques which was developed in 1985. The advantage of the RLS algorithm is that it can be deployed in any kind of document and no restrictions are imposed. RLSA itself can act as a preprocessing tool. Documents fed into the RLSA must be skew angle corrected and binarized.

Smearing is an operation that connects two nonadjacent black runs into one merged run if the distance between these runs is smaller than a threshold. In Wong's algorithm, the smearing operation is first applied to a document horizontally, row by row, and then vertically, column by column. The two intermediate results are then combined by a logical OR operation to generate the final result. Then connected component algorithm is applied to the final result to find the blocks. The algorithm is as follows: [4]

Consider a string $S=(a_1, ..., a_i, a_{i+1}, ..., a_{j-1}, a_j, ..., a_n)$ with two 1-runs $r_1=(a_1, ..., a_i)$, $r_3=(a_j, ..., a_n)$ and one 0-run $r_2=(a_{i+1}, ..., a_{j-1})$. The two 1-runs r_1 and r_3 are merged if the value of $j-i-1$ is smaller than t where $j-i-1$ is the distance between r_1 and r_3 and t is a selected threshold. The contents of the 0-run are accordingly changed from 0 to 1 if runs satisfy the smearing condition. An example of smearing algorithm with a threshold of four is illustrated in Figure 2.1.

before smearing:
11111111100000001111111100011
after smearing:
11111111100000001111111111111

Figure 2.1. Example of the smearing operation.

An image or graphic block is defined as the enclosing rectangular surrounding it. If it is a text block then the resulting block consists of several consecutive stripes with each stripe representing one line of text. But lines can also be merged into paragraph blocks if proper vertical threshold values are chosen for document under analysis.

An important consideration in RLSA is the threshold value. Depending on the content and layout of a document, the vertical and horizontal RLSA thresholds should result in proper blocks of text or image. The threshold values should preferably be made adaptive on the estimated font size. Another point is combining two intermediate images. Most document processing systems combine these intermediate images using logical AND operation. But in some cases logical OR operation may be employed. This choice depends on the required resultant image. If it is supposed to have smaller blocks, such as word or line segments, then AND operation should be chosen. But if bigger blocks, such as paragraph or text blocks, are necessarry for processing then OR operation should be deployed. The threshold values and the choice of logical operation type (AND or OR) are closely interdependent. In other words different combinations of threshold values and logical operation can result in similar outputs. For example a smaller threshold followed by an OR operation could nearly result in the same output with a bigger threshold followed by an AND operation.

If the document layout is simple, that is, if columns are in a regular format as in most of the technical journals, RLSA can perform well in locating all kind of blocks and fails if column structure is irregular. On the other hand for regular documents with proper threshold values and logical operation, RLSA can easily and quickly segment these documents. However, for Turkish newspapers, blocks have various sizes and locations - i.e. they are randomly located between text blocks -, and column formation such as line endings and continuity of gaps are not regular. Another difficulty arises from the font size

variety. If font size were not to vary within a whole document or at least were to vary within a few points than RLSA can merge characters and / or lines efficiently. But if headlines in a newspaper have a much bigger font size than the text body, then RLSA can not merge headline characters into one block. Special care must be taken for the large headline case.

2.2. The Recursive X - Y Cut

Dacheng Wang and Sargur N. Srihari proposed the Recursive X - Y Cut method (RXYC) in 1989 [5]. This well-known known top-down method has been applied to the segmentation of newspapers. Under the assumptions that newspapers are made of rectangular blocks, text, photograph, header and etc. These blocks are separated from each other by white gaps and these gaps are most proper places to locate segments in between.

The main tool in the RXYC method is the projection profile extraction, which is used in both horizontal and vertical directions. Projection profile of a binary document is simply the total number of black pixels in each horizontal - or vertical - scanning line. This profile shows the distribution of black pixels, in other words printed areas, through vertical or horizontal direction. Then a smoothing algorithm, such as lowpass filter, can be run over the projection profiles. Then a minimum-detection algorithm should be deployed to find out the valleys in horizontal or vertical directions, which correspond to column and paragraph gaps. Then a cut is performed corresponding to most dominant valley in the document. Also a check must be done on segments to prevent over segmentation, i.e. segmenting document down to the level of line segments.

RXYC can be applied on both original binary image or on the resulting image of RLSA. Mostly the latter case is used, because the results of RLSA has more evident and less noisy gaps between columns. One advantage of RXYC is that, at the end it can represent the document in the form of a tree with nested rectangular blocks.

Correspondingly if the structure of document is known a priori, then some threshold and checks can be imposed on RXYC algorithm to perform a better segmentation.

Wang *et al* [5], employed RXYC on the result of RLSA for newspaper images and they obtained quite good results. But their test images had proper column structure and homogenous rectangular blocks as seen in Figure 2.2. The Wall Street Journal is a typical case which has almost a fixed layout and not disturbed column structure for most of issues. Similarly most technical journals also have regular column format. In Turkish newspapers the first and last pages are typically the most difficult to handle, while the inner pages fit into the patter of regular documents.

RXYC should be deployed with the help of another method for newspaper segmentation since newspapers do not have regular column format.

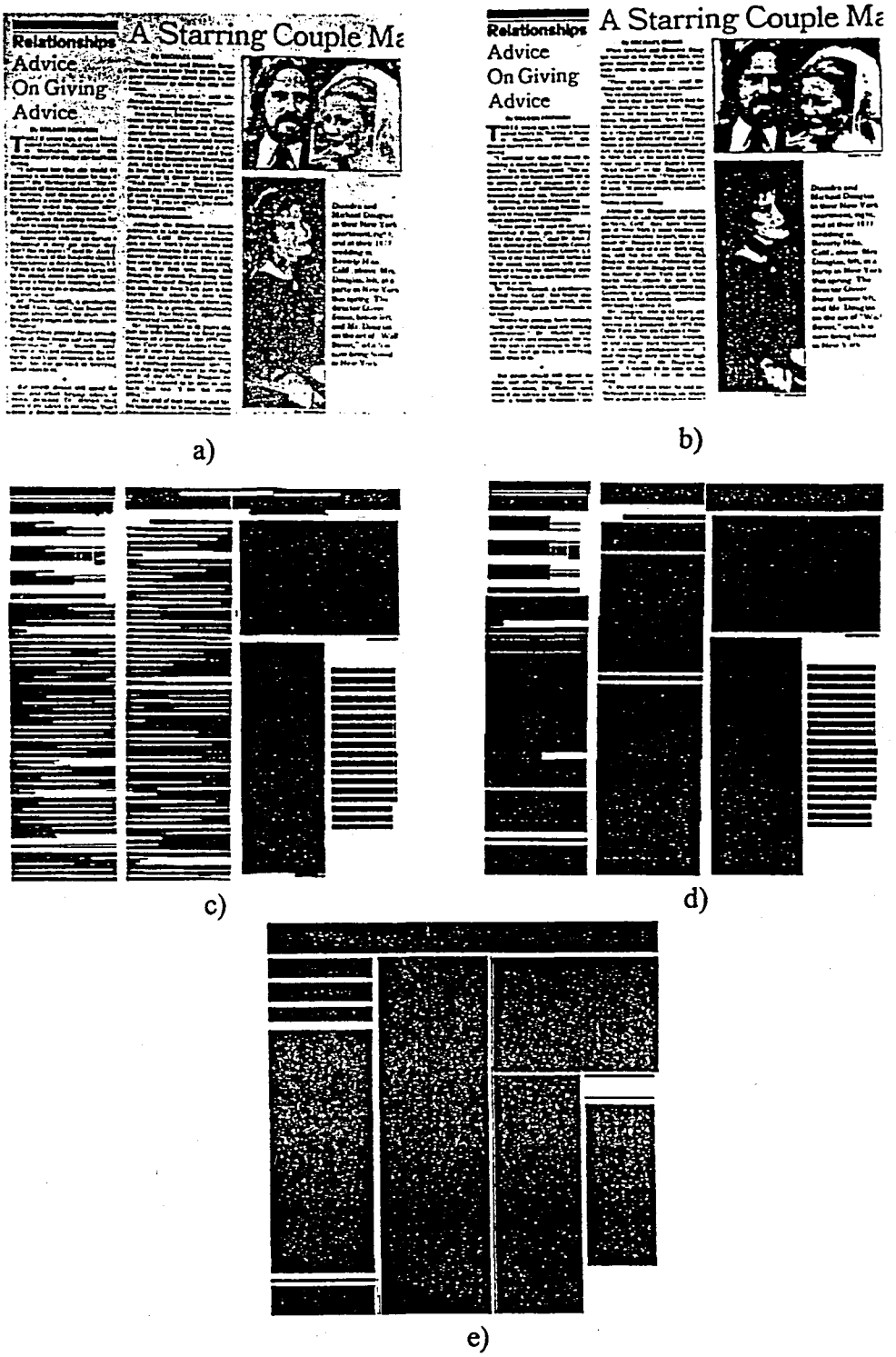


FIGURE 2.2. Steps of RXYC [5]. a) An example newspaper image. b) The binarized image. c) The block segmentation result by using RLSA. technique. d) The result of computing the bounding rectangular block for each connected component. e) The block segmentation result by using RXYC technique

2.3. The Stripe Merging Method

Stripe Merging was proposed by K.C. Fan, C.H. Liu, Y.K. Wang in 1994 [6]. In their method text blocks are segmented as paragraphs and not text lines. Because after segmentation process, storing a document in text stripes instead of text block definitely increases the storage demand. Another problem arises in document understanding, i.e. Optical Character Recognition (OCR) phase. Retrieving text stripes is also time consuming for OCR. To remedy these two problems Fan *et al* [6] developed the method of stripe merging.

Stripe merging is not a segmentation method itself. It is used as a post-processing technique to merge separate text stripes into text paragraphs using some rules. Before performing stripe merging algorithm, text stripes should be extracted. Fan *et al* assume that each text stripe can be extracted properly using RLSA. Two stripes are merged if they both belong to same text block or paragraph. The basic algorithm is as follows:

1. Each stripe S_i is represented by two coordinates. Upper-left coordinate (Xul_i, Yul_i) and the lower-right coordinate (Xlr_i, Ylr_i).
2. Each stripe S_i is compared with S_{i-1} and S_{i+1} regarding their representing coordinates.

Through the analysis of whole document, comparison of stripes S_i and S_{i+1} or S_i and S_{i-1} is done according to three criteria. If two stripes meet them, then they are merged into one. But the shortcoming of this method is that, it assumes that the paragraphs are in regular form as depicted in an example in Figure 2.3.

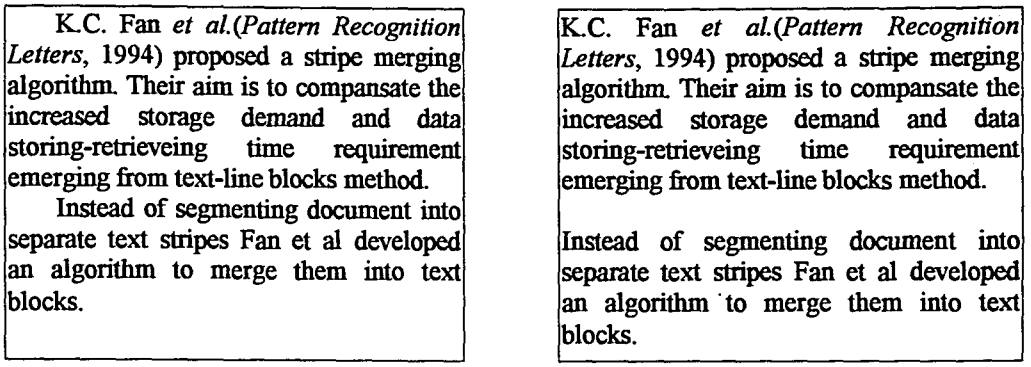


FIGURE 2.3. Tested paragraph formats in the article of Fan *et al* [6]
a) indent separated paragraphs, b) space separated paragraphs

If the block size of the resulting image is much larger than the preselected threshold then stripe merging process is abandoned, otherwise it is performed. Two text stripes are merged if both of the following two conditions are satisfied.

1. The projection of these two stripes in the vertical direction overlap. That corresponds to the fact that stripes within a text blocks are aligned in a way that they vertically occupy the same coordinates.
2. None of the below three cases occur. This condition is explicitly checked by the merging rules.

Case 1 : Since two paragraphs are usually delimited by wider spaces (see Figure 2.3), this property can be utilized to determine if S_i and S_{i-1} belong to same text block.
Stripes S_i and S_{i-1} do not belong to the same text block if the vertical distance between these two stripes is larger then a threshold.
$| Yul_i - Ylr_{i-1} | > threshold_1$ ----------> *S_i and S_{i-1} do not belong the same block.*

Case 2 : Since usually appear at the beginning of the first line in a new paragraph, it can be utilized to decide if S_i and S_{i-1} belong to same text block.

Stripes S_i and S_{i-1} do not belong to the same text block if an indent occurs at the head of stripe S_i.

$|Xul_i - Xul_{i-1}| > threshold_2$ ----------> S_i and S_{i-1} do not belong the same block.

Case 3 : Since short line stripes usually appear at the end of the last line in the current paragraph, it can be utilized to determine if two stripes S_i and S_{i+1} belong to same text block.

Stripes S_i and S_{i+1} do not belong to the same text block if indent occurs at the end of stripe S_i.

$|Xlr_i - Xlr_{i+1}| > threshold_3$ ----------> S_i and S_{i+1} do not belong the same block.

The formats tested above are not valid for every newspaper because the paragraph format is quite distributed as shown in Figure 2.4. The ends of text stripes are not aligned as one in the article. To remedy this problem to a certain extent, only beginnings of stripes and average stripe length might be compared.

K.C. Fan *et al.*(*Pattern Recognition Letters*, 1994) proposed a stripe merging algorithm. Their aim is to compensate the increased storage demand and data storing-retrieving time requirement emerging from text-line blocks method.

Instead of segmenting document into separate text stripes Fan et al developed an algorithm to merge them into text blocks.

FIGURE 2.4. Paragraph with not aligned line endings

Beside the line ending problem, another issue is the font size problem. One cannot be assured to get proper text stripes for every font size after RLSA and to merge these varying stripes within the same process. Fine tuning over threshold values, which vary according to the height and width of the text stripes, may solve this problem to a certain extent. But instead of doing this Fan *et al* [6] assumes these text stripes are extracted after RLSA and take such big text blocks out of merging process. A sample output for this method is illustrated in Figure 2.5.

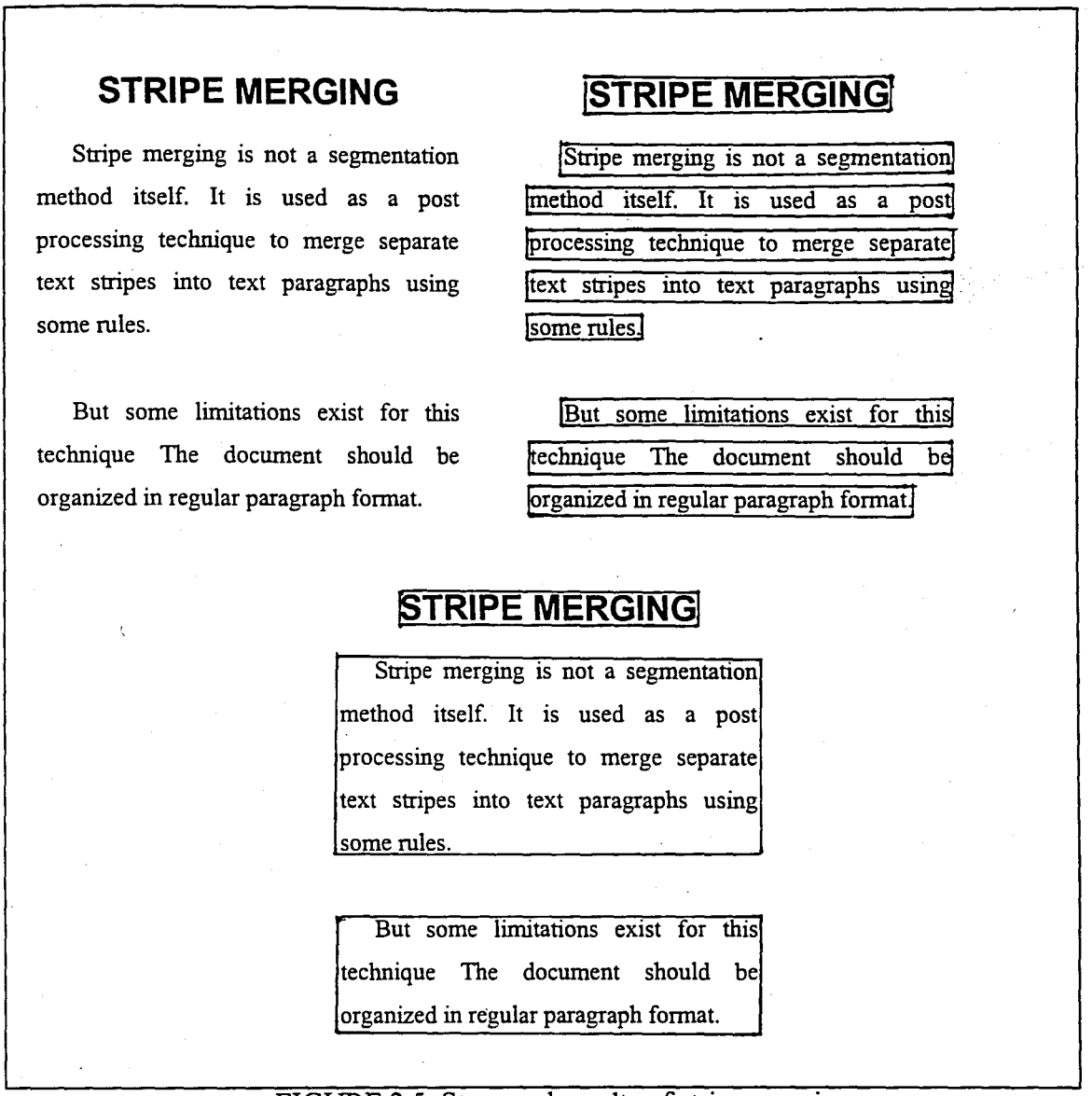


FIGURE 2.5. Steps and results of stripe merging.

2.4. White Streams

This segmentation method was developed by T. Pavlidis and J. Zhou in 1992 [3]. In this method instead of developing relations among nearby foreground (black) objects, as in most of the studies, they analyze the structure of background regions (white spaces). The basic assumption is that columns are subregions of the input page containing ideally a

unique type of data separated by white spaces which are wide enough to be distinguished from other spacing such as the white spacing between words. Their characteristic is that, they are continuous in vertical direction with respect to other white gaps such as ones between characters or words. They also assume that vertical white spaces of size larger than a threshold, which is calculated with respect to average text line height, separate column parts such as text blocks or paragraphs. This algorithm is aimed to find the largest column blocks. No restriction is applied on the layout of the page. Also they assume that text could have various font types and sizes within the page.

While extracting white spaces two error-prone case should be examined and avoided. The first one is the false spacing between ascenders and descenders within a column and second one is that across column. These faulty cases are illustrated in Figure 2.6. Former one may cause false starts i.e. the algorithm may conclude that there is a column gap between ascenders (or descenders) where as there is not in fact. The latter case may disturb the width of the stream i.e. a wider gap between inter column ascender (or descenders) may force the algorithm to estimate the column gap larger than it is. Both problems can be remedied if a vertical projection profile is used over a block of scan lines. Because vertical projection profile gives an idea about where columns are located, which in turn implies the column gaps. Therefore a vertical projection profile must first be obtained for the image. Prior to obtaining vertical projection, a horizontal smearing operation is applied to emphasize column structure and to remove sporadic noise spots.

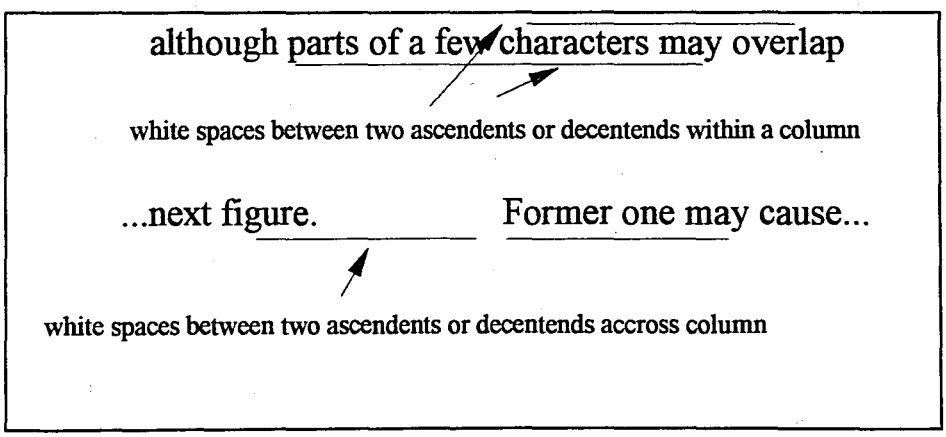


FIGURE 2.6. White spaces that do not correspond to the column spaces.

In this method on the vertical projection profile the following operations are used:

1. Form the vertical projection and find the white spaces that correspond to column gaps.
2. For each pair of column gaps, identify the entity between two gaps, which are called column interval. The start and end of the block of scanlines are considered to be (trivial) column gaps.
3. Except for the column intervals encountered in the first scanline block, compare the column intervals with those of the above and merge them if the rules below are satisfied.

Rules for Merging Column Intervals : Each column interval has four parameters : X_{left}, X_{right}, Y_{top}, Y_{bottom} . A parameter that belongs to interval P is shown as P$\Rightarrow X_{left}$ A new column interval P will be merged with column interval Q if following conditions are met.

1. Two intervals are very close in vertical direction

$$| P \Rightarrow Y_{top} - Q \Rightarrow Y_{bottom} | < threshold_1$$

2. One of the horizontal projection of the two column intervals contains the other.

$$P \Rightarrow X_{left} > Q \Rightarrow X_{left} - threshold_2 \quad \text{and} \quad P \Rightarrow X_{right} < Q \Rightarrow X_{right} + threshold_2$$

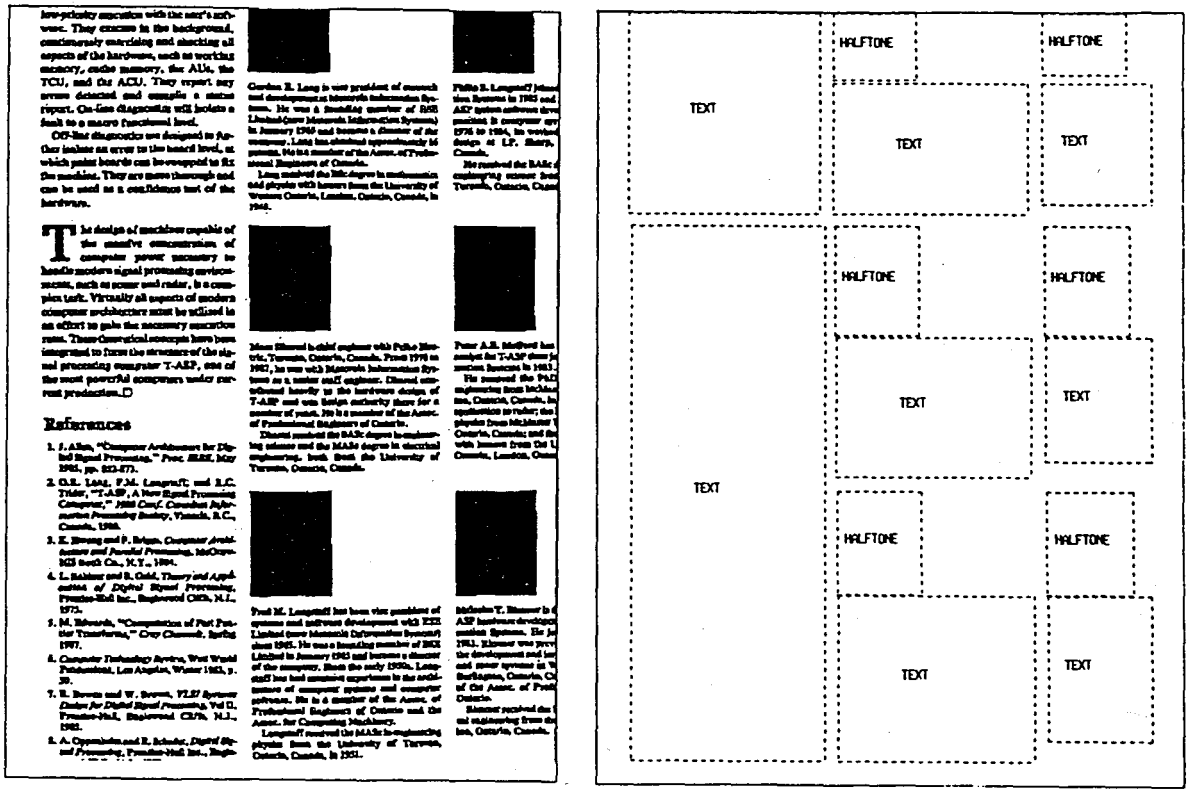
or

$$P \Rightarrow X_{left} < Q \Rightarrow X_{left} + threshold_2 \quad \text{and} \quad P \Rightarrow X_{right} > Q \Rightarrow X_{right} - threshold_2$$

3. The widths of the two intervals are approximately the same.

$$\frac{\min \{P \Rightarrow X_{right} - P \Rightarrow X_{left} \ , \ Q \Rightarrow X_{right} - Q \Rightarrow X_{left} \}}{\max \{P \Rightarrow X_{right} - P \Rightarrow X_{left} \ , \ Q \Rightarrow X_{right} - Q \Rightarrow X_{left} \}} < threshold_3$$

After forming blocks and merging them, if necessary conditions are met, they deployed a process to eliminate small regions, which are caused by printing defects etc. But to avoid removing some small blocks, such as those produced by isolated characters, very short text lines and very narrow blocks in vertical direction, the final segmentation is re-evaluated. If they are small in either horizontal or vertical direction and satisfy the first and the second rules, with another threshold values, than they are merged with the major blocks. The result of this method is given in Figure 2.7.



a) b)

FIGURE 2.7. An example for white space method [3]. a) An example document image and b) the result after applying the merging process.

This method is quite similar to the RXYC method in Section 2.2. Both method utilize the column gaps and use the projection profiles. But, the "white stream" method extracts the vertical projection profile from the total page and uses it through the whole processing whereas the RXYC recursively uses and extracts both the horizontal and the vertical projection profiles for page segments at each step. The valley in the profiles are main decision criteria for RXYC while they are just supports the merging process for the present method. It can be concluded that method by Pavlidis *et al* is slower and more complex than the RXYC method and does not have any advantage over it.

2.5. Line Growing

This method developed by A.A. Zlatoposky in 1994, [7] which is based on the assumptions that all the factors that perturb the global document layout - complicated block arrangement, image rotation, different font size and line position etc. - do not destroy the simple local layout.

After binarization, to locate text elements with rectangle bounding boxes, connected component detection algorithm is also used. Then line growing algorithm is applied over the image.

Line growing algorithm starts with the simplest objects - characters, signs - that form connected black regions in the binarized image. The input data for line growing algorithm are these regions bounded with rectangles. Then some neighborhood relations among these regions are used for growing operation.

Line segments (LS). growing : The algorithm starts with the leftmost object and look for its nearest right neighbor which is "near" in vertical direction. If such a neighbor is found it is included in the current line segment else a new LS is started.

The detection of line segments neighbors: A search for nearest neighbors of each LS is performed at this step. Each LS has one right and one left neighbor where there may be several upper and lower neighbors. This case is illustrated in the following Figure 2.8.

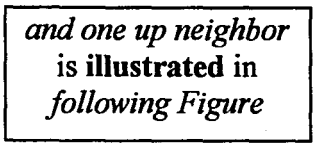


FIGURE 2.8 Example of line segments neighborhood

Assume that we have each word as a line segment after the smearing process. Here the word, i.e. line segment, **'illustrated'** has four upper neighbors and two lower neighbors (as shown in *italics*). Whereas, it has only one left neighbor and one right neighbor.

Horizontal neighbors : For the leftmost unprocessed LS one finds the nearest right neighbor which is not excessively shifted in the vertical direction.

Vertical neighbors : One finds all the nearest upper (lower) LSs that overlap with the current one in the vertical direction and are not "very far". Only the leftmost and rightmost upper (lower) neighbors are used later, so each LS has up to two upper (lower) neighbors. But this neighborhood relation is not mutual (LS_i can be vertical neighbor of LS_j but it does not mean that LS_j is a neighbor of LS_i.)

Line segment merging : To find and join LSs vertically, algorithm looks for "small" near vertical neighbors of each LSs. If we consider horizontal direction, the aim is to merge LSs in each column but do not merge LSs in different columns. That is why algorithm looks for LSs that form the left margins of text blocks. First, they determine the vertical neighbors that are in same leftmost position to find the beginning of lines. Starting with these LSs they search the near enough upper and lower neighbors. This search is done in both upwards and downwards direction

After LS merging process, LSs within the same vertical projection are joined to construct text blocks which corresponds to paragraphs. These paragraphs can also be merged to form bigger text blocks. The steps and result of algorithm is given in Figure 2.9.

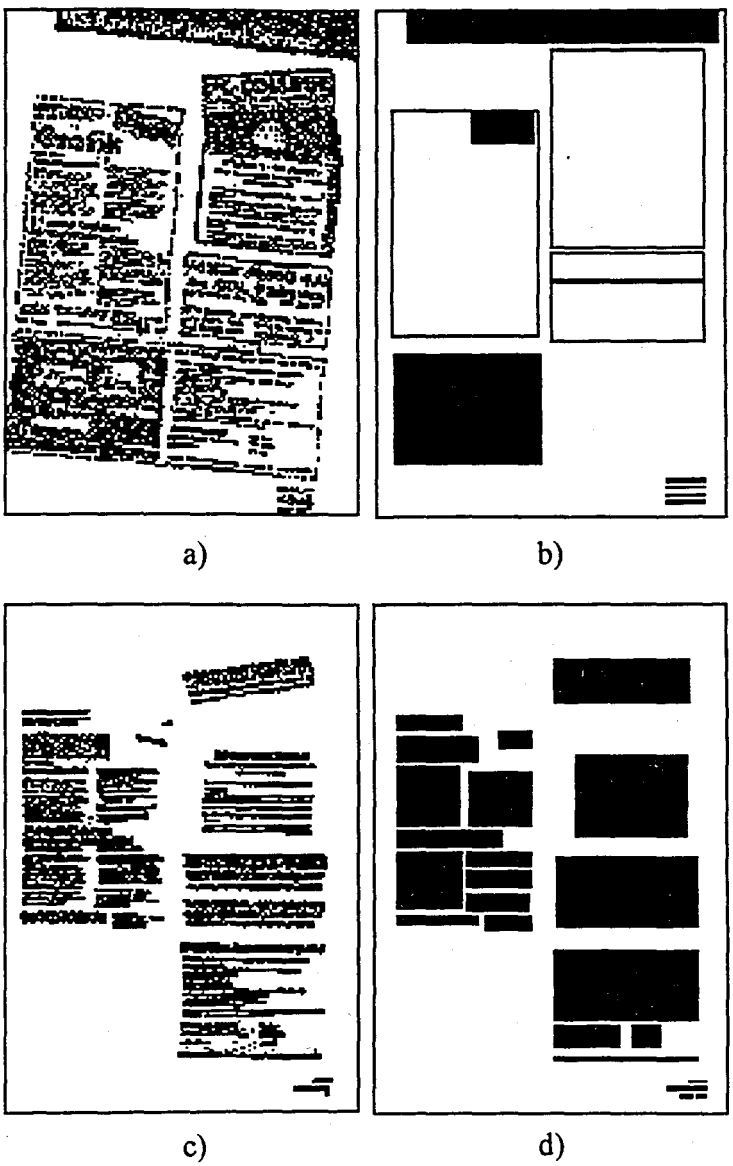


FIGURE 2.9. Steps of line growing. a) Input image. b) Detected graphic blocks, empty rectangles mark graphic blocs that can include text blocks. c) Detected text lines. d) Detected text blocks.

2.6. Texture Based Page Segmentation

A. K. Jain and Y. Zhong [8] proposed a method where the algorithm works directly on the gray level image. This method has no restriction and can be used for any kind of document, since they consider the texture properties of the image. Initially, they perform a segmentation method on gray level image and produce three types of texture segment namely background, image and text and line drawings. Then they group text regions and line drawing regions into one class.

Their basic assumption is that the image, background, text and line drawing regions have different textual features. Jain uses special purpose filters to increase the accuracy of document segmentation obtained via a multilayer neural network.

To speed up segmentation process, static or adaptive quantization on the input image is used to reduce the number of gray levels. Then texture discrimination masks are obtained. by training a neural network using sample data from these three classes within 7X7 windows. The three-layer perceptron model neural network is trained using the back propagation algorithm to obtain the twenty masks used for classification. The input layer corresponds to masks, weights from input layer to the hidden layer are the coefficients of the masks. The output layer corresponds to the each of the three classes.

At segmentation step, the neural network classifies the input pixels into one of the halftone regions, background regions and, text and line drawing regions. Later, to separate text regions from line drawings another segmentation process is performed. In this stage these regions are first thresholded and binarized. Then using connectivity features these regions are labelled as text region or line drawing.

The postprocessing operations involve the removal of salt and pepper effect, merging neighboring regions and placing bounding boxes around the classified regions. The steps of the segmentation method are shown in Figure 2.10.

This method is totally independent of the page layout. There is always a danger of misclassifying text regions as a halftone region. The greatest advantage of this method is that, it uses the same features for both segmentation and classification.

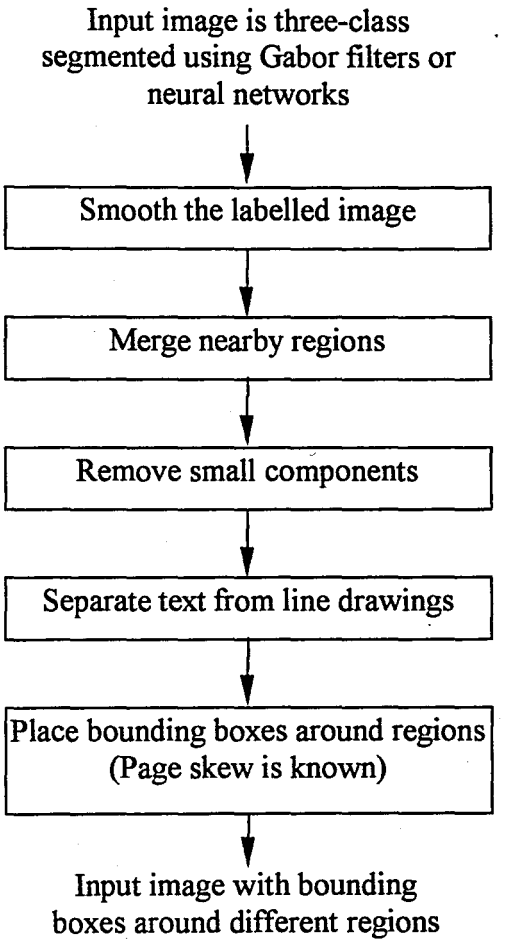


FIGURE 2.10. Steps of segmentation after classification

3. A HYBRID METHOD

Top-down methods have some assumptions on document structure. Documents are supposed to have a regular column structure and have wide gaps between columns, paragraphs and images. That is not always the case for Turkish newspapers.

On the other hands, bottom-up methods suffers from the mostly character sizes within a document. For example, classification of small blocks, such as letters and words, may produce faulty results. Some difficulties may also be encountered during merging of relevant blocks, such as merging two or more word segments into one line segment. Therefore a hybrid approach as a combination of a top-down and a bottom-up method, which are explained in Section 3.1 and Section 3.2, is deployed. This is infact one of the original contributions of our proposed method

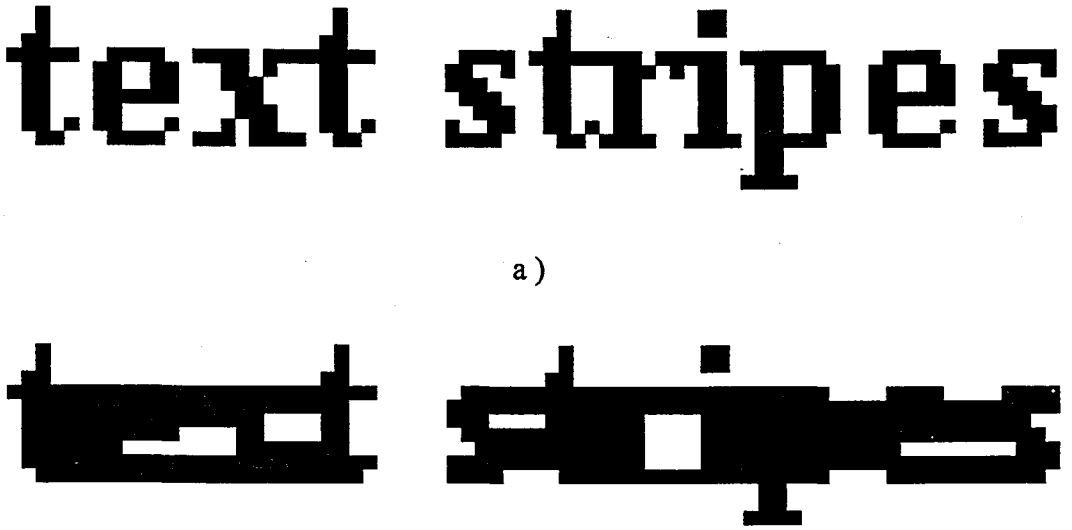
3.1. Modified Recursive X - Y Cut

In modified Recursive X – Y Cut (RXYC), a more efficient Run-Length Smearing method (RLS) is used as the bottom-up method. Then RXYC as the top-down method is deployed. After all another bottom-up method is introduced to execute over the resultant blocks of modified RXYC method, if necessary.

3.1.1. The Smearing Operation

The first operation on the bi-level image is smearing. The aim of this operation is to fill the white gaps which have a length less than a given threshold. Consequently the density of the black regions within a document is emphasized. The details of smearing

operation is given in Section 2.1. Here, a sample horizontal smearing operation with a threshold of three is given in Figure 3.1.



a)

b)

FIGURE 3.1. a) Sample text line, b) same line smeared with a threshold of three.

The bi-level image is first smeared vertically with a given threshold, and then the original image is smeared a second time in the horizontal direction. This mean that, adjacent background pixels in either direction within distances less than their respective thresholds are converted to foreground. As a result, two separate smeared images are obtained. These two images later might be combined using logical "AND" or "OR" operation. We have investigated both of them.

In the former one, if a pixel is black in both of the smeared images then, after "AND" operation, a black foreground label is assigned to that position in the resulting image.. To remove the residual small white gaps between line segments a vertical smearing operation with a smaller threshold is applied over the "AND"ed image. In the "OR" operation, if a pixel position is white in both of the smeared images than an "OR" operation assigns white pixel label to that position.

As can be seen from the results in Figure 3.3, the "AND" operation gives smaller and disconnected blocks which are almost similar to the original document. On the other

hand "OR" operation results in larger and more black regions with respect to "AND". Consequently, the "OR" combined image better suits to the RXYC method. The results for both operations are given in Figure 3.4.


HABERLER

PERŞEMBE

Skandal başladı, faili meçhul bitti

Bankalardaki ürkütücü tablo

Güneydoğu'da 13 gazeteci öldürüldü

'Susurluk bir olaylar zincirinden ibarettir'

FIGURE 3.2. Original newspaper image

FIGURE 3.3. AND combined image

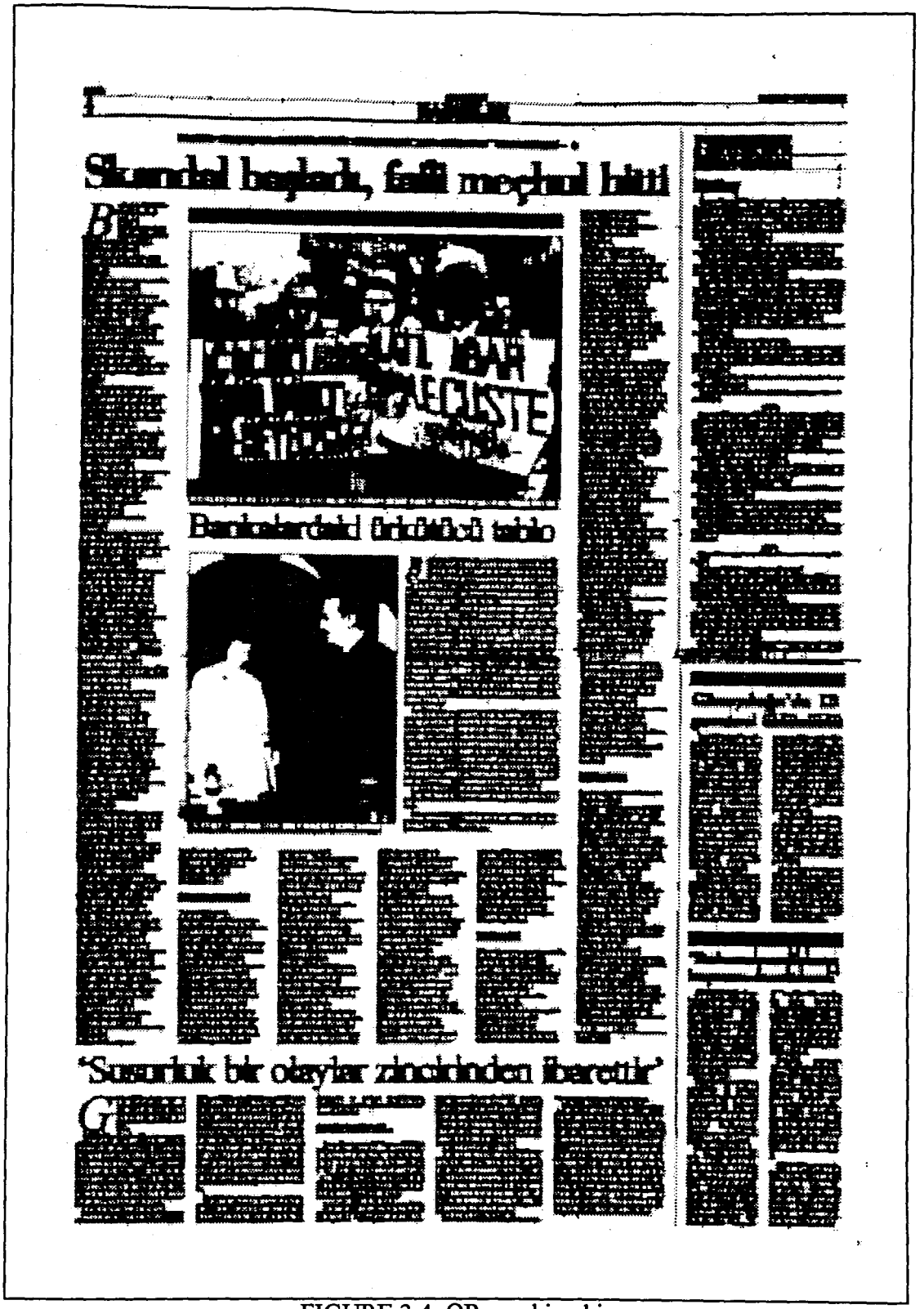


FIGURE 3.4. OR combined image

Some faulty results after either the horizontal or the vertical smearing operations may occur. For example a column gap between two vertical adjacent line segments might be merged. Therefore, the thresholds in both smearing operations should be chosen carefully. The thresholds in the horizontal smearing operation must be assigned to a value that is smaller than the inter-column gaps and larger than or equal to the inter-word gaps. After applying horizontal smearing operation, the resulting image is supposed to have characters merged into words and words merged into line segments. One must be careful in selecting the threshold value in that, if this threshold is too big than the adjacent text blocks in horizontal direction could be merged into a single block, which would compromise the whole page segmentation process. Typically information about character pitch size could aid in the selection of the horizontal threshold.

On the other hand vertical threshold is found considering the vertical gaps between line segments, line pitch and between text blocks and paragraphs. The vertical smearing operation should merge line segments and may merge paragraphs, however, text blocks should be kept separate.

After whole of the smearing operation, the bi-level image containing black labeled regions which hopefully correspond to text, image, graphic and line blocks is obtained. In Figure 3.5 through Figure 3.7., the images at each step of smearing operation is given for the sample newspaper image in Figure 3.2.

FIGURE 3.5. Vertically smeared image

FIGURE 3.6. Horizontally smeared image

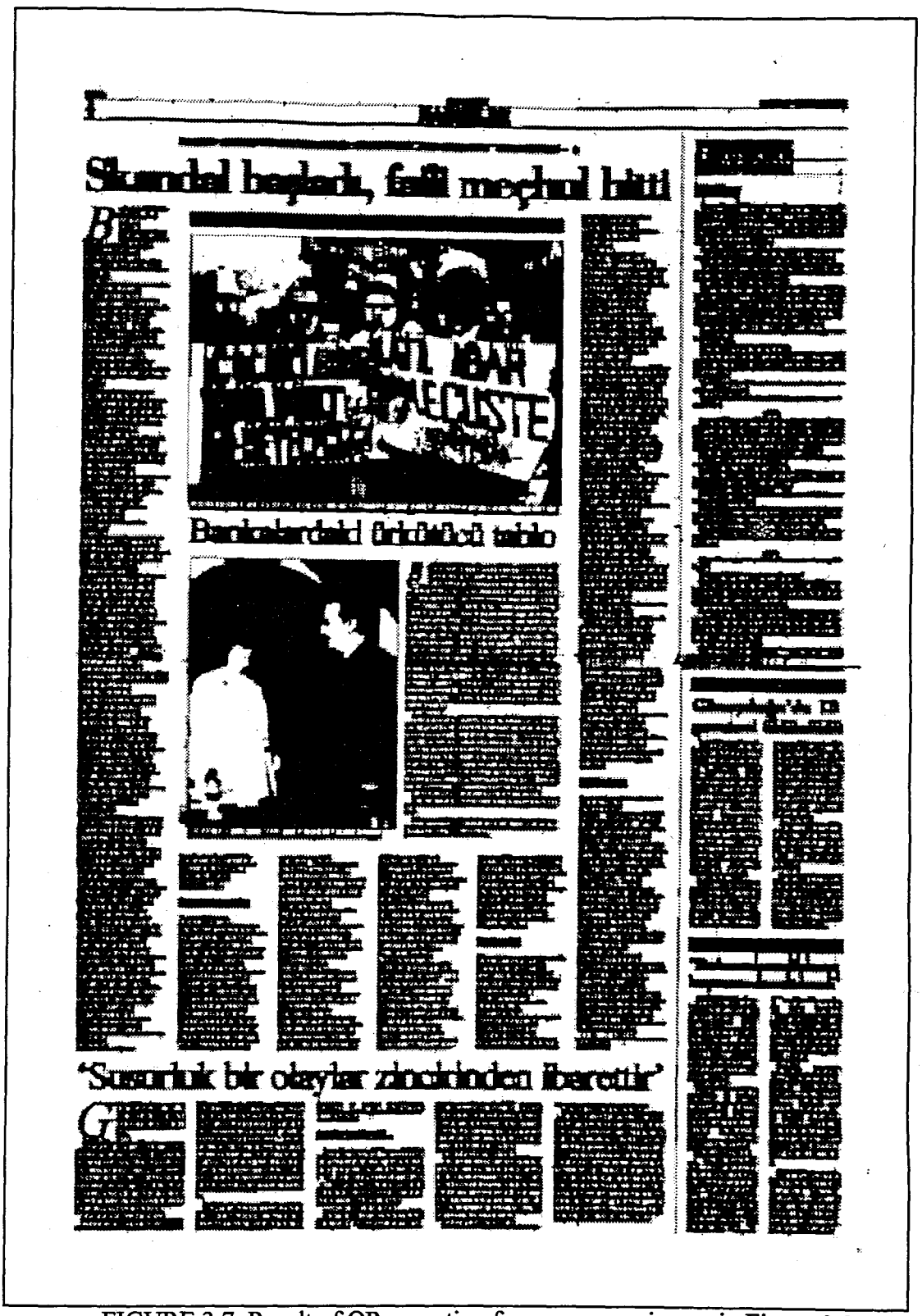


FIGURE 3.7. Result of OR operation for newspaper image in Figure 3.2.

3.1.2. Projection Profile Extraction

In the recursive X - Y cut method a newspaper image is always cut down into two subimages. At each step there is either a horizontal or a vertical cut placed over the image. The location and the direction of the cut - whether horizontal or vertical - are determined using the horizontal and vertical projection profiles of the bi-level image. Therefore the horizontal and vertical projection profile must be obtained at each step for each sub-image individually. These profiles are then evaluated according to some criteria - described in Section 3.1.3 - and a cut decision is given. The cut place corresponds to the most dominant gap in the image; or, in other word, to the most dominant valley in the projection profiles. The recursive cut process is explained in the following Section.

3.1.2.1. Obtaining Projection Profiles. The projection profile, which is simply the projection of pixels onto an axis, e.g. horizontal or vertical, is obtained by counting the number of black pixels in each row or column respectively. The projection profiles give information about the location of white gaps between columns and blocks, from which clues are derived about the location of text, image and graphical blocks also.

Either profile will consist of valleys and plateaus. The plateaus correspond to document objects like text blocks, photographs, columns etc. The valleys correspond to the gaps between objects such as text blocks, photographs, headlines (horizontally) and page setup columns (vertically). The deeper and wider the valley the longer and wider the gap between blocks is assumed to be. The plateaus, on the other hand correspond to the document objects. The width and height of the plateaus are proportional to the size of the blocks in the smeared image. As a result the variations of profiles provide evidences for the location of blocks and gaps. Horizontal and vertical projections of sample newspaper images are given in Figure 3.8.

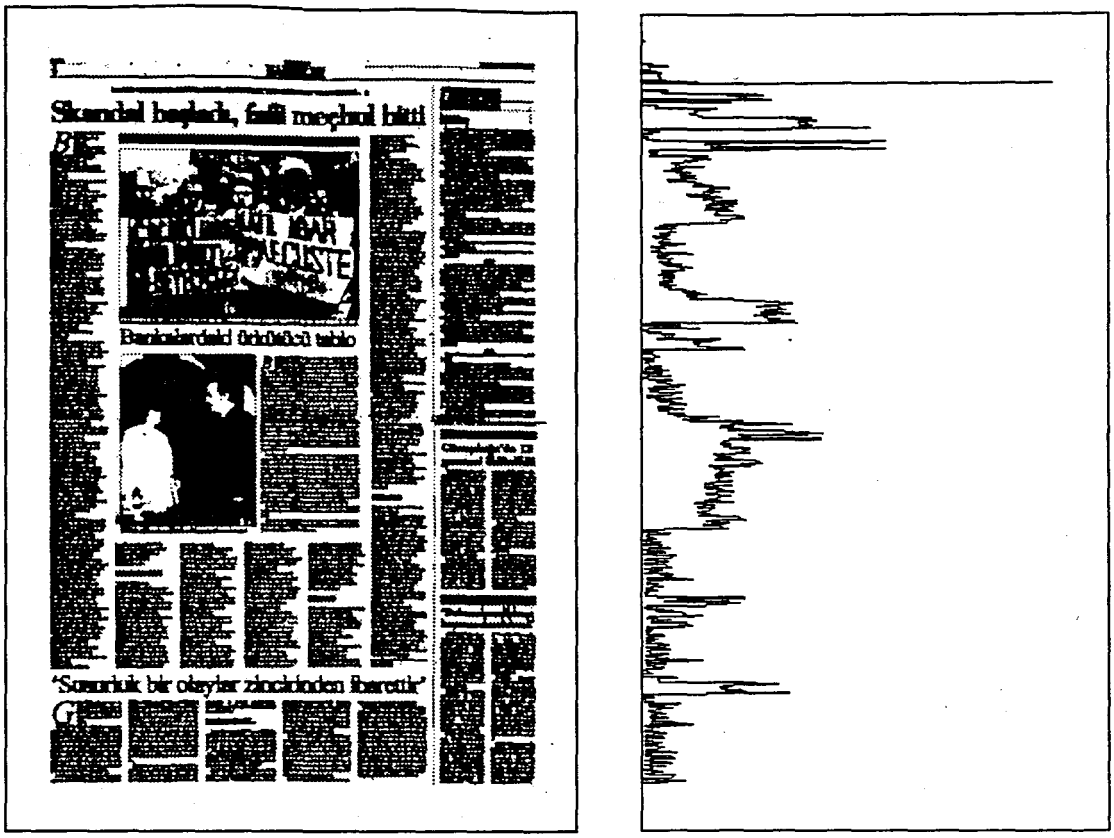


FIGURE 3.8. Sample page and its horizontal and vertical projection profiles

3.1.2.2. Pre-evaluation of Projection Profiles. Before analysing the horizontal and projection profiles to determine a cut place, a pre-evaluation operation should be carried out. The aim is to verify that the block will be analysed in both direction or in only one direction. This decision is based on the size of the block and depth of the valleys in both profiles. Evaluation of them is explained in Section 3.1.4. The algorithm is depicted in Figure 3.9.

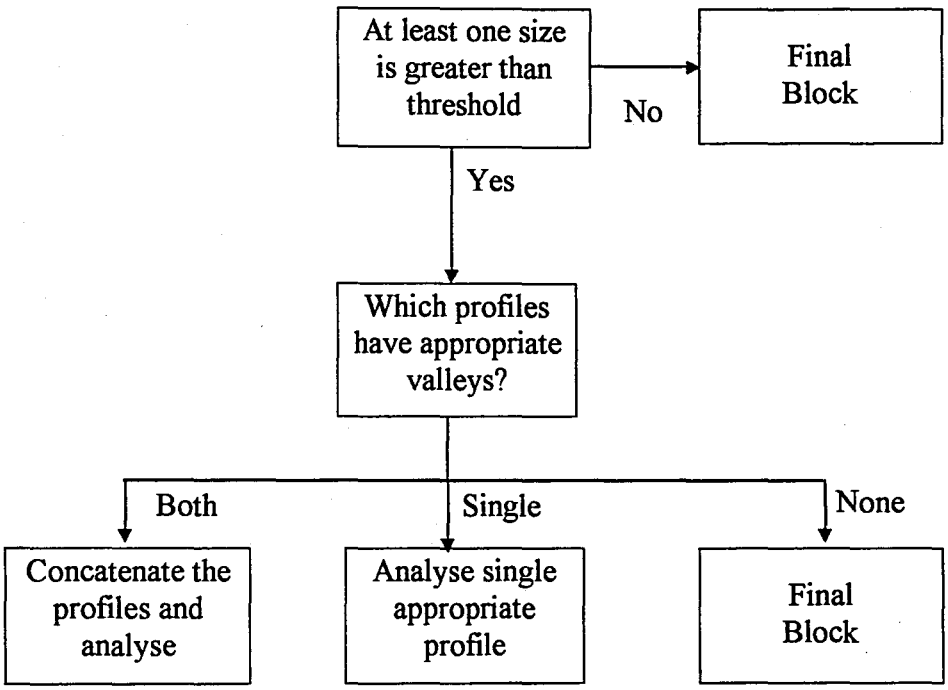


FIGURE 3.9. Flow of pre-evaluation process.

3.1.2.3. Concatenation of Projection Profiles. As pointed out before, at each step of the recursion, only one cut is placed in one of the directions of the image. Therefore, if concatenation decision is given in pre-evaluation step then analysing horizontal and vertical projections separately would be suboptimal and could cause segmentation errors. Since at least one valley at each profile can be labelled as "the most dominant valley", then there would be two candidates for a cut place. One should select the more relevant cut of the horizontal and vertical options, and this can only be achieved if the two profiles are observed on a commensurate basis. In other words the RXYC algorithm should analyse the horizontal and the vertical profiles after they are properly spliced. We have arbitrarily

selected to add the vertical projection profile after the horizontal one without any loss of generality. To denote the concatenation point in the projection profile the value of "zero" is inserted between the profiles. Afterwards, we let the decision algorithm to find the most dominant valley in the concatenated profile. An example of concatenation process is given in Figure 3.10.

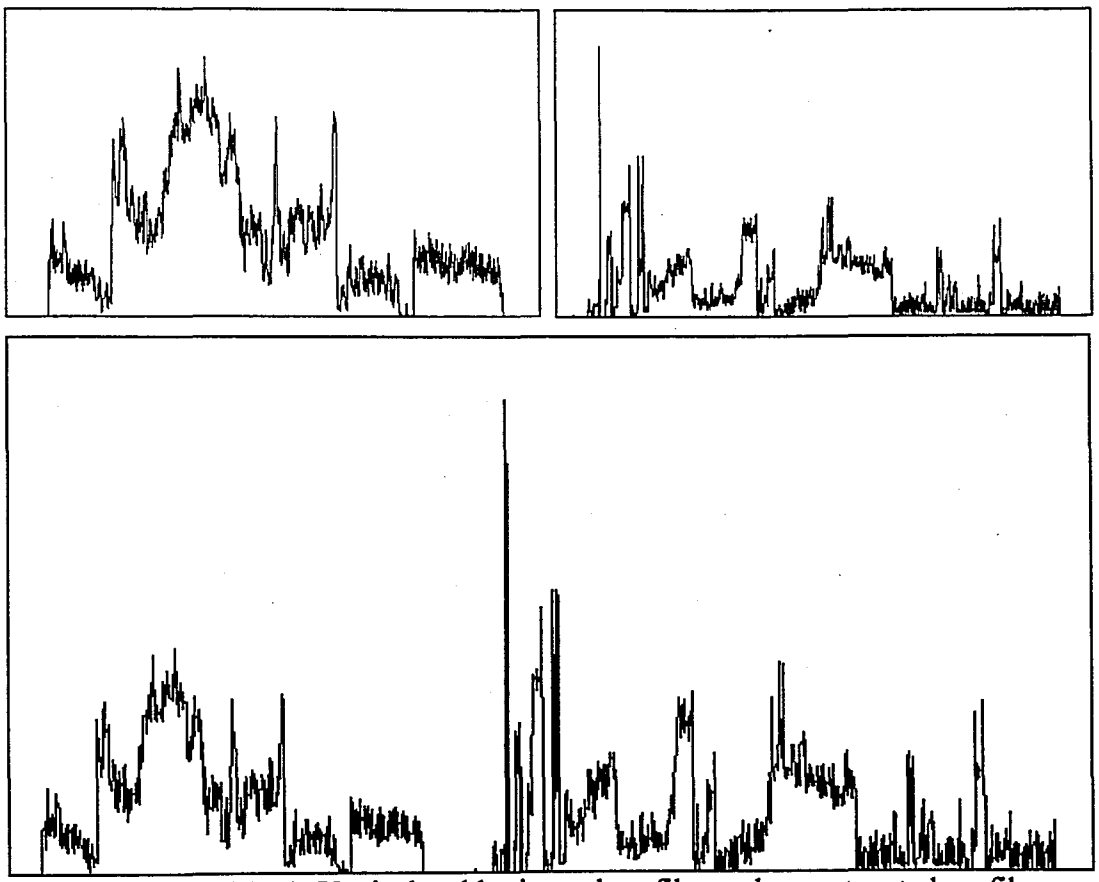


FIGURE 3.10. Vertical and horizontal profiles and concatenated profile.

If pre-evaluation algorithm does not result in the concatenation decision, then single profile, which is appropriate, should be analysed. The rest of whole analysing algorithm treats same to both the concatenated and single profiles. Except, since there is not a concatenation process scaling operation, which is explained in Section 3.1.2.4, makes no sense for the single profile case. But for the sake of generality, we call both types of profiles as concatenated.

3.1.2.4. Scaling of Profiles. The concatenated profile has two different segments which may differ significantly in scale from each other. The width and height of the analyzed subimages are almost never the same at any one step of the recursion. Recall that the horizontal and vertical projection profiles have heights strictly dependent on the width and height of the subimage. Thus if the of projection line is long, then the resulting sum will be large, and vice versa. If the horizontal and vertical projection profiles are not commensurate the segmentation decisions would be wrongly biased.

We would like the make the dominance of a valley independent of the size of the subimage. Consider the case of a longitudinal column, with an obvious horizontal cut. However in the projection profiles, the vertical direction will accumulate more points, so that its valley-peak difference will be larger than that of the horizontal profile. To compensate for this effect one should scale the horizontal and vertical profiles with respect to each other to give the same chance to both of the valleys.

We have investigated two scaling methods. The first takes into consideration the peak values while the second one is according to the areas of the profiles. However we have realized that area based scaling does not make sense since vertical and horizontal profiles would integrate to the same number. Recall that after all they are obtained by row wise or column wise summation over the same number of pixels. Consequently, the scaling operation is carried out using peak values. The peak for each profile is found, and the profile with the lower peak is scaled up in proportion to the higher peak.

$$\text{scaled\_profile} = \text{lower\_profile} \times \frac{\text{peak\_of\_higher\_profile}}{\text{peak\_of\_lower\_profile}} \qquad (3.1)$$

where lower (higher) - profile means the profile with the lower (higher) - peak value. After the scaling operation the valleys in each profile become comparable, in other words they have now similar chances of being the most dominant valley in the whole document. The Figure 3.11 shows the concatenated profiles before and after scaling operation for a block of differing width and height. Notice for example the salient vertical valley before concatenation becomes secondary after scaling and concatenation. Further adjustments of the same scaling operation is done after lowpassing of profiles.

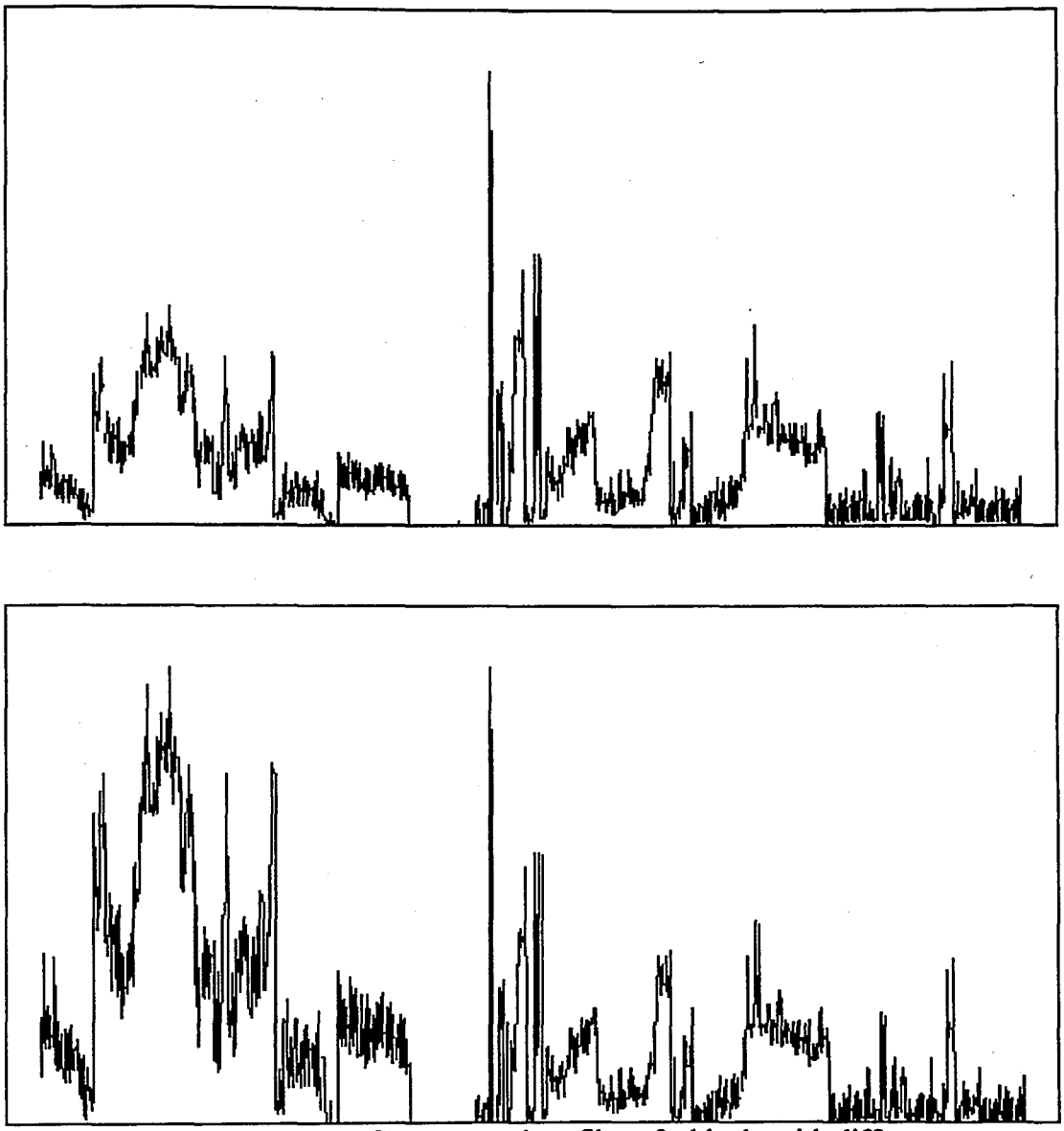
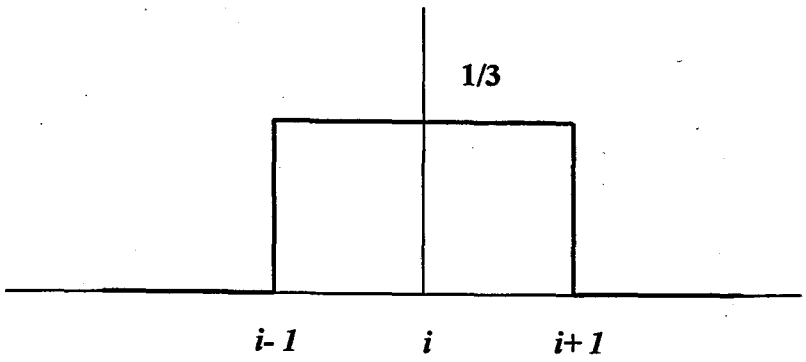
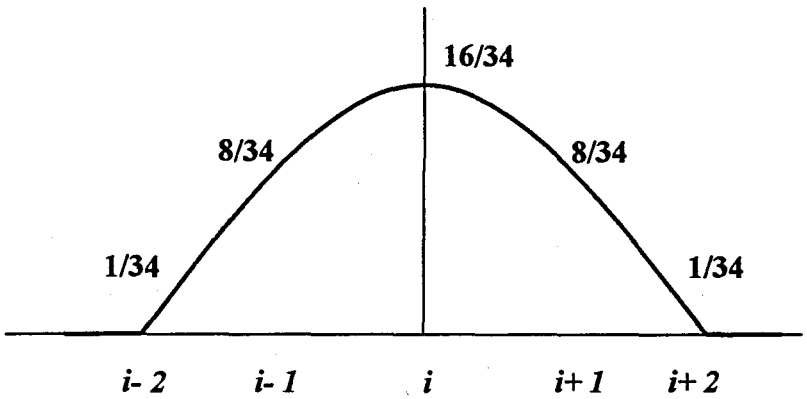


FIGURE 3.11. Concatenated profiles of a block, with different height and width, before and after scaling operation.

3.1.2.5. Lowpass Filtering of the Profiles. Due to the different font sizes within a page, irregular distribution of black pixels in a document image blocks and non-aligned text and image blocks cause several irrelevant valleys and plateaus of varying sizes along the profile. The small valleys and hills can be considered as pure noise. The recursive cut operation becomes more effective if these spurious details are eliminated first. To remove them a lowpass filtering operation is applied to the profiles. We have considered two lowpass filter types.



$$lp_f(i) = (f(i+1) + f(i) + f(i-1)) / 3 \quad (3.2)$$



$$lp_f(i) = (f(i+2) + 8f(i+1) + 16f(i) + 8f(i-1) + f(i-2)) / 34 \quad (3.3)$$

FIGURE 3.12. Filter shapes.

The first one is a three-coefficient box filter. It simply takes the average of current point and its neighbors at both sides. The second one has a Gaussian curve shape and has five coefficients. The impulse response and the shape of the filters are shown in Figure 3.12.

For computational speed, the box filter is chosen for the lowpass filtering operation. Observing the Figure 3.13 it can be concluded that this choice would not have much of an impact on the result.

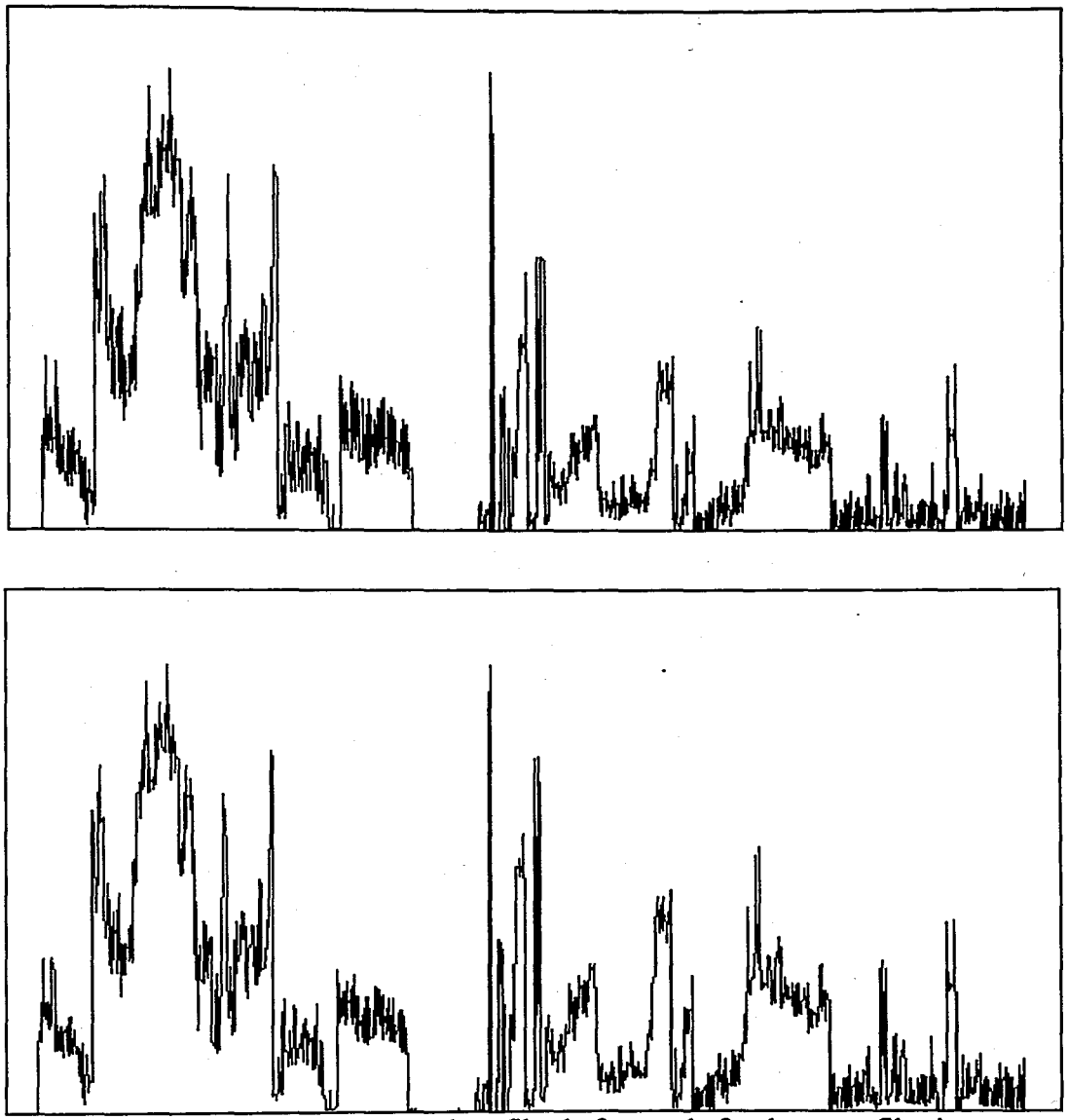


FIGURE 3.13. Concatenated profiles before and after lowpass filtering.

At this point, concatenated profile curve is ready to be evaluated and to find its most dominant valley. To do this, four generic features of a valley are calculated and each valley is evaluated with these features.

3.1.3. Locating a Cut Place

In document segmentation the blocks are separated from each other by white gaps. These gaps also determine the block boundaries. To find out each block individually these

boundaries should be extracted. In RXYC method a white gap, corresponding the boundaries of a block or more than one block, is found at each step. The cut place is the most dominant horizontal or vertical gap between blocks. To find out this valley, first, the following valley features are defined.

3.1.3.1. Features of a Valley. We have used four valley features, the first three being extracted from the lowpassed profile curves, while the fourth one is obtained from the unfiltered projection profile. These four features form a valley discriminator and depicted in Figure 3.14..

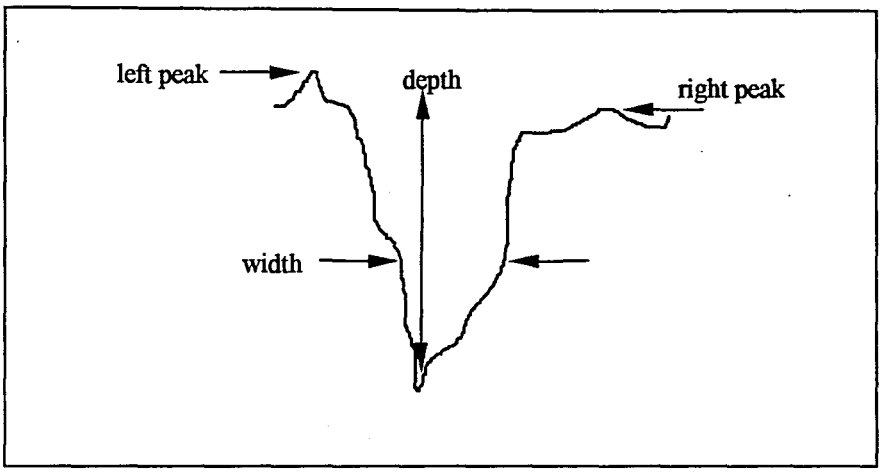


FIGURE 3.14. Valley and its features

Peaks : There are two peaks associated with each valley, one on each side. Two criteria of analysis use the peaks at each side for each value in the histogram. These peaks are used to qualify the depth of a valley. The right and left peaks are defined as the furthest point that is NOT smaller than all the previous one in the search order starting from the bottom of the valley and proceeding right or left. This is depicted in Figure 3.15.

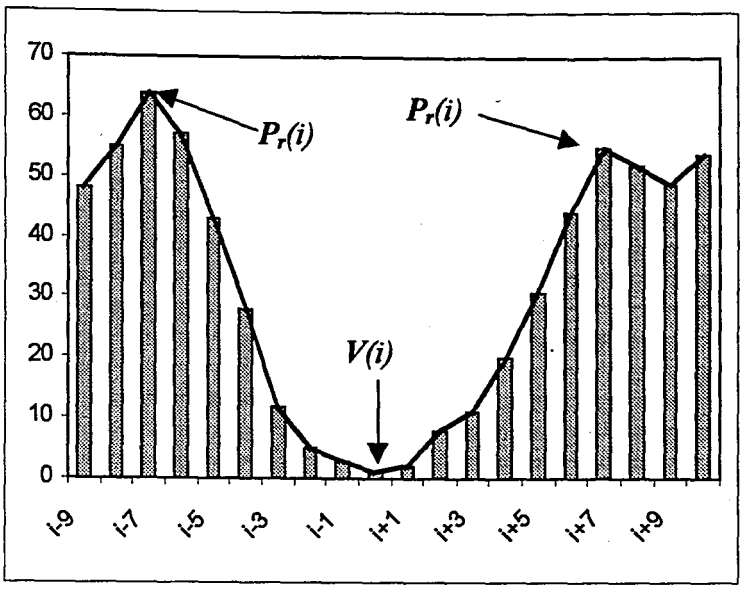


FIGURE 3.15. Peaks of a valley.

$$P_r(i) = \max\left\{V(j) \,|\, V(j) > V(j-1),\ j = i, i+1, \ldots, N\right\} \tag{3.4}$$

$$P_r(i) = \max\left\{V(j) \,|\, V(j) > V(j+1),\ j = i, i-1, \ldots, 0\right\} \tag{3.5}$$

3.1.3.2. First Feature - Valley Depth. Our aim is to place the cut into a position that corresponds to the deepest and widest valley. The first feature evaluates the depth of valleys. The depth of a valley is also proportional to the height of the peaks on both sides of that valley. Projection peaks imply edges of blocks of big sizes, while valley depths are directly proportional to the length of the inter-block gaps Based on these observations, the algorithm first finds out the peaks at each side and finds their mean value. This mean-peak value is divided by the valley value (in fact plus one; in order to avoid "division by zero"). As is illustrated in Figure 3.16, for the profile in Figure 3.13, the result of this step are larger the deeper the valleys.

$$V_d(i) = \frac{P_r(i) + P_l(i)}{V(i)} \tag{3.6}$$

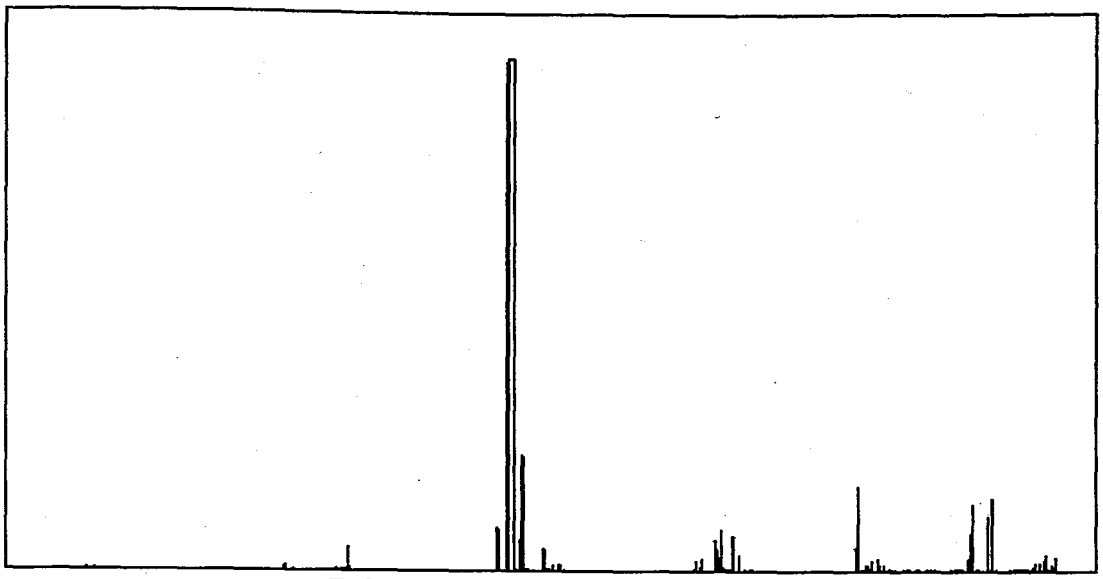


FIGURE 3.16. Result of the first feature

3.1.3.3. Second Feature - Valley Steepness. The second feature evaluates the steepness of valleys. A steep valley means fast roll down on both sides. This implies that the edges along a projection, are linearly arranged or it means a gap common to more than one collinear block. The steeper the projection the more suitable a cut place.

To find the steepness of a valley the second derivative of the projection curve is used. The second derivative tends to peak for valley bottoms with a sharper and narrower turn.. Two different methods for second derivative are evaluated; ordinary derivation and quadratic derivation methods. The formulae of the two methods are given in Equation 3.7 and Equation 3.8. and their application for a profile curve are given in Figure3.17. It is obvious that the quadratic derivation has smoother shape than the ordinary one. Therefore the quadratic derivation method is chosen for the calculation of the second feature.

$$f'(x) = \big(f(x+1) - f(x-1)\big)/2 \qquad \text{ordinary method} \tag{3.7}$$

$$f'(x) = \big(2f(x+2) + f(x+1) - f(x-1) - 2f(x-2)\big)/6 \qquad \text{quadratic method} \tag{3.8}$$

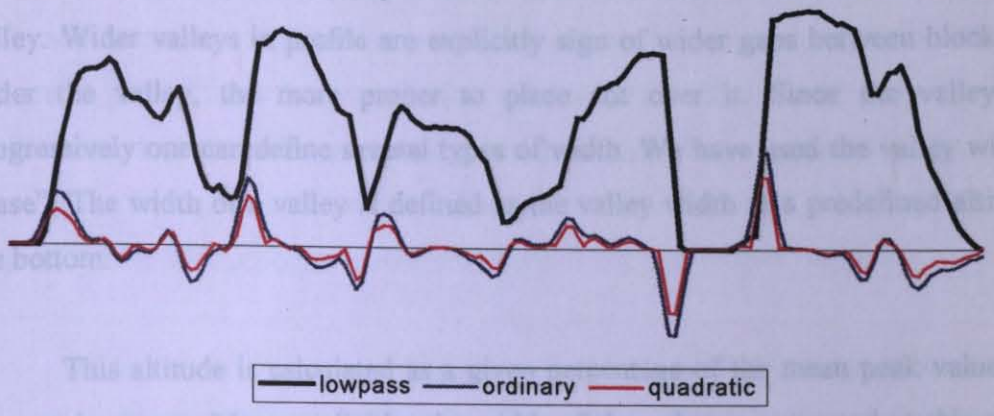


FIGURE 3.17. Derivation curves for quadratic and ordinary methods

As in the case of the first feature, to emphasize the steepness, the second derivative of the profile is divided by the profile value itself. To prevent a "division by zero" one is added to the profile values. Figure 3.18 shows results for second criterion.

$$V_s(i) = \frac{V''(i)}{V(i)} \qquad (3.9)$$

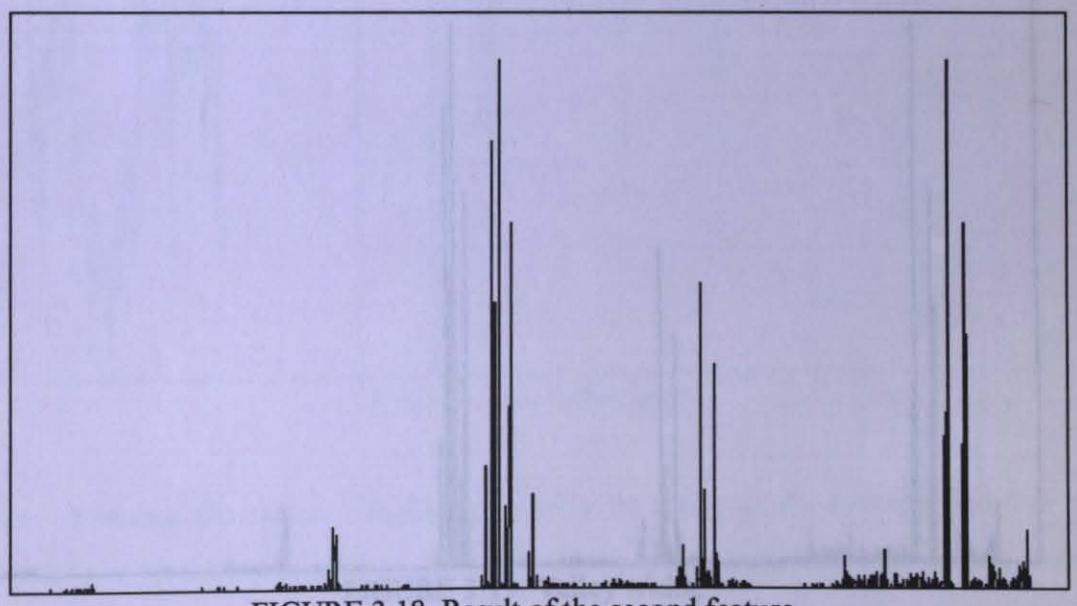


FIGURE 3.18. Result of the second feature

3.1.3.4. Third Feature - Valley Width. The third feature focuses on the width of the valley. Wider valleys in profile are explicitly sign of wider gaps between blocks. So, the wider the valley, the more proper to place cut over it. Since the valleys roll-off progressively one can define several types of width. We have used the valley width at the "base". The width of a valley is defined as the valley width at a predefined altitude from the bottom.

This altitude is calculated as a given percentage of the mean peak value found in the previous step. More explicitly, the width of the value is measured at this altitude by computing the number of neighborhood projection values in each direction, both left and right, which are smaller than threshold value $\mathbf{D}_{th}$. If the altitude percentage value, k, is chosen too big or too small most of valleys would fail to satisfy this criterion.. Adequate values of the valley base altitude is chosen as 30 percent of the mean peak value. The result for this criterion is depicted in Figure 3.19.

$$\mathbf{D}_{th}(i) = k * \left(\mathbf{P}_r(i) + \mathbf{P}_l(i)\right) \tag{3.10a}$$

$$\mathbf{V}_w(i) = S\left\{\mathbf{V}(j) : \mathbf{V}(j) \text{ is neighbor of } \mathbf{V}(i) \middle| \mathbf{V}(j) < \mathbf{D}_{th}\right\} \tag{3.10b}$$

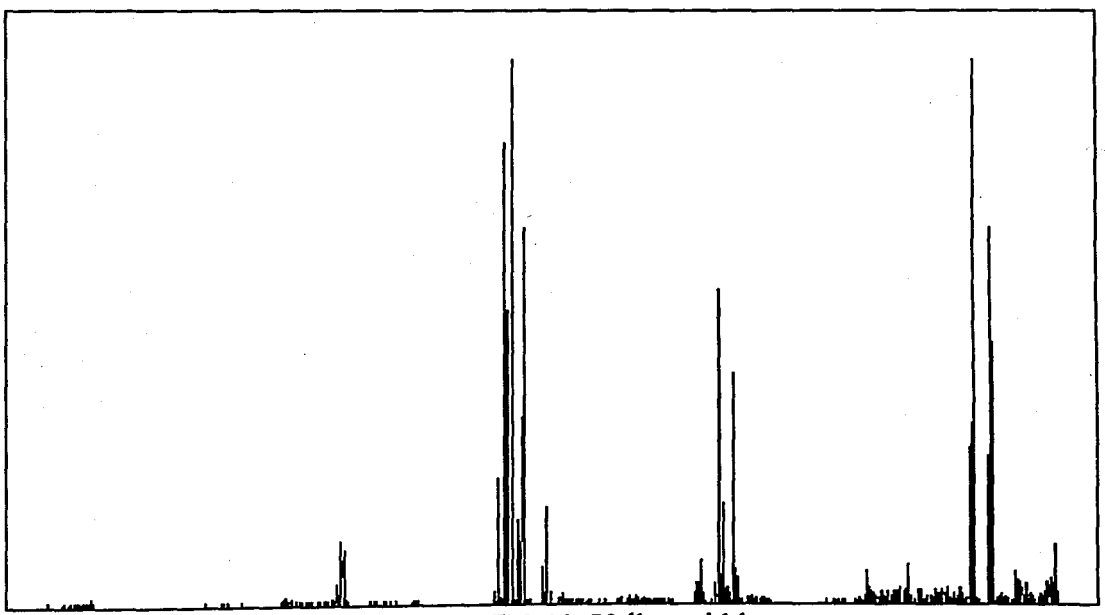


FIGURE 3.19. Valley width.

3.1.3.5. Fourth Feature – Valley Base. The last criterion used to locate cut over document image, checks the profile values in the sense of their closeness to zero. If there is no document element such as line, text, image or graphic along a projection line, then that projection profile value in that position will be "zero". The closer to zero a projection profile is the more it becomes a candidate for the cut place. By chance some zero points may appear in a projection profile, although there is not a real inter-block gap for this projection line. This problem is solved by combining four criteria results.

While the previous three features, lowpassed profiles are used, in this case the original profile itself is used. The fourth criterion, first considers the maximum value of the concatenated projection profile. Then, all profile values are subtracted from this maximum value. In a sense, the complement of the projection profile is found with respect to the maximum. The results of this complement one is given in Figure 3.20.

$$V_b(i) = max_p - V(i) \tag{3.11}$$

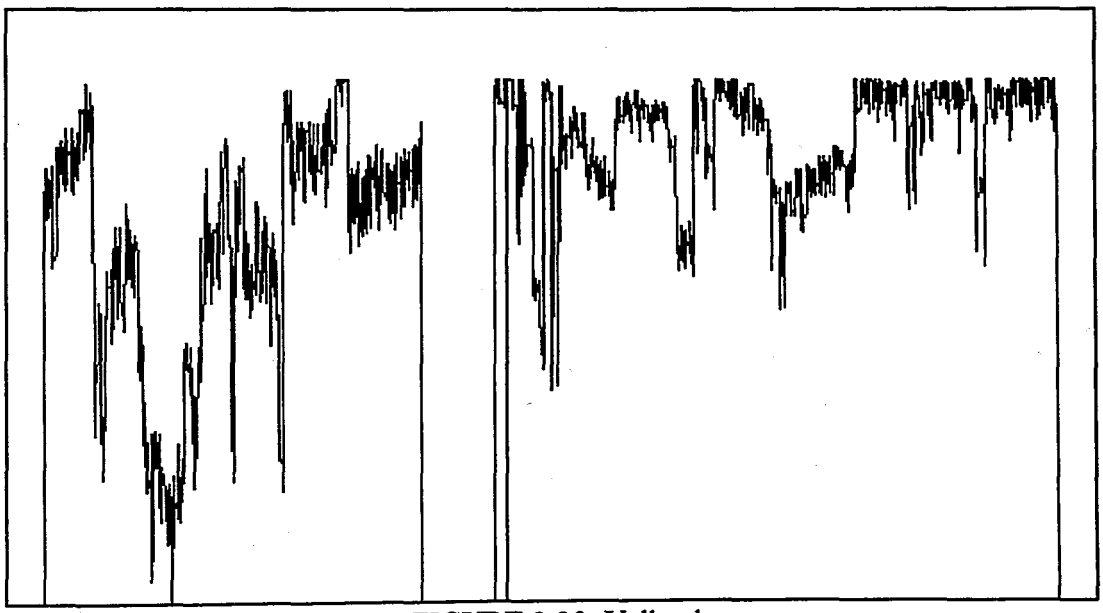


FIGURE 3.20. Valley base

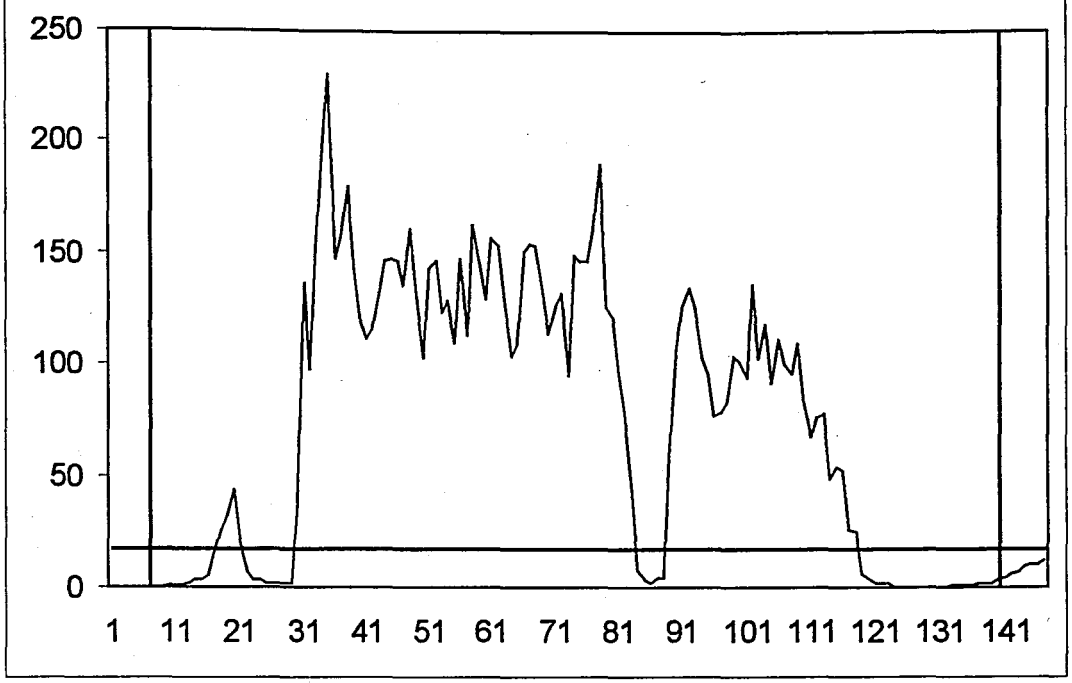
3.1.3.6. Pruning Operation. Special care must be given to the splicing point of the horizontal and vertical projection profiles, as well as to the beginning and ending slopes. The three points of the profiles may turn out to be the artificially dominant valley over the

others. Therefore one must avoid placing cuts to these three artificial valleys. For a single profile, only the beginning and ending slopes should be considered.

An algorithm is run to prune the portions of the profiles to be suppressed. Four criteria are evaluated individually for horizontal and vertical projection profile values and as a result suppression values are determined at the beginning and end portions, using following criteria.

1. If the profile is monotonously increasing in these extremity portions (beginning and end), the value where the rising trend ends is found.
2. If profile is monotonously decreasing in these portions (beginning and end), the value where the decrement ends is found.
3. For horizontal profiles, the 5% and 95% of width are set as suppression values and for vertical profiles, the 5% and 95% of height are set as suppression values.
4. The value where the profile values "first" exceeds the 10% of the peak of profile, which is currently evaluated, is set us suppression value.

A sample evaluation of above four criteria are depicted in Figure 3.21.



According to four criteria;

The horizontal bold line indicates the 10% percent of the peak value. The vertical bold line on the left indicates the 5% of the total points where that on the right corresponds to95%.

Evaluation of left side

1 – The increasing trend ends at the value **21.**
2 – Since there is not a decreasing trend this criterion has no contribution, **0.**
3 – Since there is 150 points, 5% of it corresponds to **7.**
4 – The peak is 230 and 10% of it is 23, the curve reaches this value at **17.**
The maximum of these four is suppression value and equals to **21.**

Evaluation of right side

1 – Since there is not a increasing trend this criterion has no contribution, **150.**
2 –. The decreasing trend ends at the value **123.**
3 – Since there is 150 points, 95% of it corresponds to **143.**
4 – The peak is 230 and 10% of it is 23, the curve reaches this value at **118.**
The minimum of these four is suppression value and equals to **118.**

As a result values between **0-21** and **118-150** will be set to zero.

FIGURE 3.21. Sample profile and results of three criteria.

At the left hand side of profiles the maximum of the above three criteria is set to left suppression value whereas at the right hand side the minimum of above three is set as right suppression value. All criteria values lying beyond these suppression points are set to zero. After this operation reliable profile portions for evaluation are produced.

3.1.3.7. Scaling Criteria Results. The last post-processing step over features is the scaling operation. Because of different mathematical methods used for each criteria. the numerical range of outputs are in quite different ranges. For example, the maximum value for the height feature might be "20", for the mean peak value might be "55", for width might be "12" and that for fourth criteria might be "680". If a valley has become the "best" according to criteria-1, 2 and 3 and the "second" for criteria-4, this does not suffice for that valley to be the "winner" (the "winner" is assigned according the weighted summation of the results of four criteria), although it is the best three out of four criteria. To solve this problem, every value for a criterion is scaled to a value between 0 and 100. This is done to give the same importance and chance to every feature.

3.1.3.8. Weighting and Obtaining Result. At the end of whole process, the results of each normalized feature for each candidate in the profile is summed up. The summation can be done straight forward or after a weighting process. Weighting is done to put criteria outputs in an importance order. The "winner" corresponds to the position that has the maximum value after the summation and weighting process. If the cutting position is at the left hand side of concatenation point (the point where horizontal and vertical histograms joins) cut is obtained from vertical projection profile and a vertical cut will be placed at the end - and vice versa. An example for four criteria and weighted result is given in Figure 3.22.

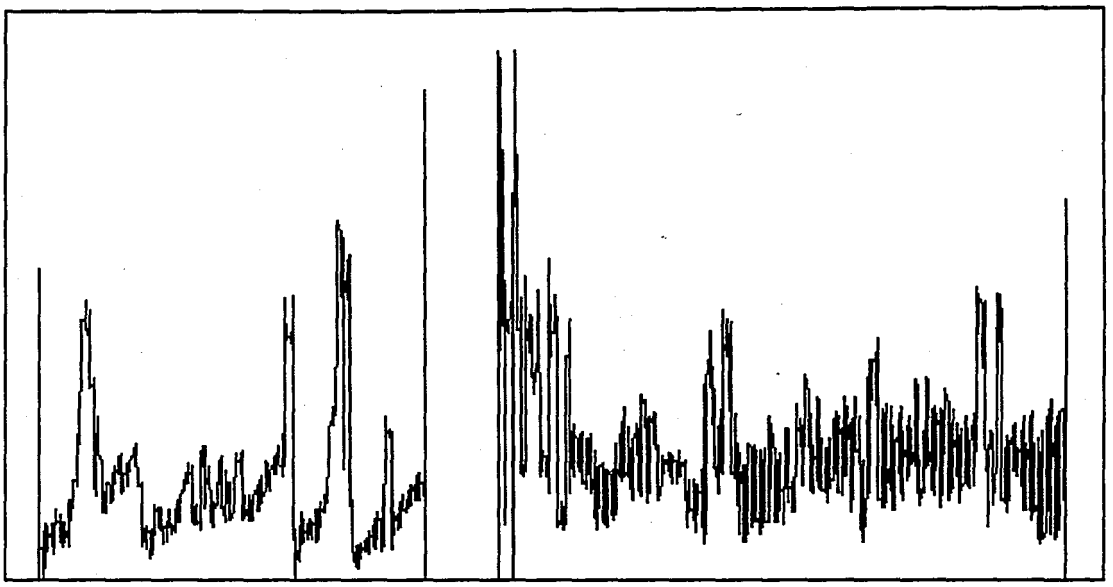


FIGURE 3.22. Weighted result for four criteria

3.1.4. Stopping Criteria

In RXYC method, the document image is segmented into two subimages recursively. Therefore, we have to define a "value" at which the recursive cuts should stop. If this value is reached, the recursion should be stopped and the coordinates of final block must be recorded to be utilized later in document understanding phase. Since the newspapers are processed in this study, the stopping values are defined specifically for this case. The aimed minimum blocks are the text blocks, not the words or characters. Therefore stopping values are defined depending upon the column size, headline font size and an approximate text block height. Another noteworthy point is that we have to evaluate horizontal and vertical stopping values independently for each block.

3.1.4.1. Stopping Values for Horizontal Cuts. At each step of recursive algorithm, we evaluate the blocks to see whether they are appropriate for horizontal cut or not. The evaluation is based on the height, width and the unfiltered horizontal histogram of the block. A threshold value corresponding to the 3% or 5% of block width is assigned. The points in horizontal histogram lower than this threshold value correspond to white bands - or a proper place for cut. If these white bands are wide enough then the block can be processed for horizontal cut. A question may arise for the threshold value. Why a

percentage value, instead of zero, is used ? Because of printing faults and scanner noise some black points may reside within the white bands and they must not affect the cut decision. To remedy this problem a value close to zero and proportional to the height of block should be chosen.

Another issue for horizontal stopping decision is the block height. To prevent over-segmentation and find homogenous blocks, the height of block is checked at each recursive step. If the block height is smaller than a threshold this block is not allowed to be cut horizontally. There are two choices: the ordinary text line height and the headline font size. A minimum text block includes at least 3 or 4 text lines. Therefore the block height should not go below that size. But some headlines have such big fonts that are almost 6 - 7 text line width. In this case we assign the horizontal cut stopping value to the headline font size. This threshold value, in units of pixel, depends on the resolution and can be changed.

3.1.4.2. Stopping Values for Vertical Cuts. As in the case for horizontal cut, same sort of evaluation is made for vertical cut at each step. The height, the width and the unfiltered vertical projection profile of block are used for evaluation. Depending on the height of block, the white bands are searched through the vertical projection profile using the same manner as for horizontal cut. If wide enough vertical valleys exist, than that block becomes proper for vertical cut.

The block width is also tested at each step. The basic data to check block width is obtained from the column structure of newspaper pages. Except the advertisement pages, the maximum number of column within a page is six. Therefore the one sixth of whole page width is assigned to the threshold value. The value, in units of pixel, may change depending on the document image resolution. The vertical cut is not permitted over the block, if its width is smaller than this threshold.

There is a strong inter-relation between horizontal and vertical cut evaluation. This phenomenon is slightly mentioned in Section 3.1.2.2. This relation can result in four possible outcomes. This case is illustrated in Table 3.1.

TABLE 3.1. Cross relation in cutting criteria.

		Valley Appropriateness			
		Ver & Hor not appr.	Only ver. appropriate	Only hor. appropriate	Both appropriate
Size	$h >= th_1$ & $w >= th_2$	0	2	3	1
	$h >= th_1$ & $w < th_2$	0	0	3	3
	$h < th_1$ & $w >= th_2$	0	2	0	2
	$h < th_1$ & $w < th_2$	0	0	0	0

0 = Final block; recursive algorithm stops.

1 = Concatenate both profiles and analyze it.

2 = Only horizontal projection profile will be analyzed.

3 = Only vertical projection profile will be analyzed.

3.1.5. Recursive Algorithm

The recursive algorithm uses the methods which are explained in Sections 3.1.2, 3.1.3 and 3.1.4 at each step of process. It evaluates the block i.e. document image to decide whether to stop algorithm or not. If not, then it finds the best cut place in either direction. After applying the proper cut, recursive algorithm produces two subimages. The cutting operation goes on until stopping values for the subimage are reached in both direction, that means neither a horizontal nor a vertical cut can be made over the subimage. At each step, the algorithm has two subimages in hand. If the stop decision is given then the block coordinates for two subimages are recorded. This recursive operation produces a tree structure of subimages as given in Figure 3.23. After the stop decision is given, algorithm goes one level up, shifts to the adjacent branch and again goes down over the new branch; until the stopping values are reached. The whole algorithm is depicted in Figure 3.24.

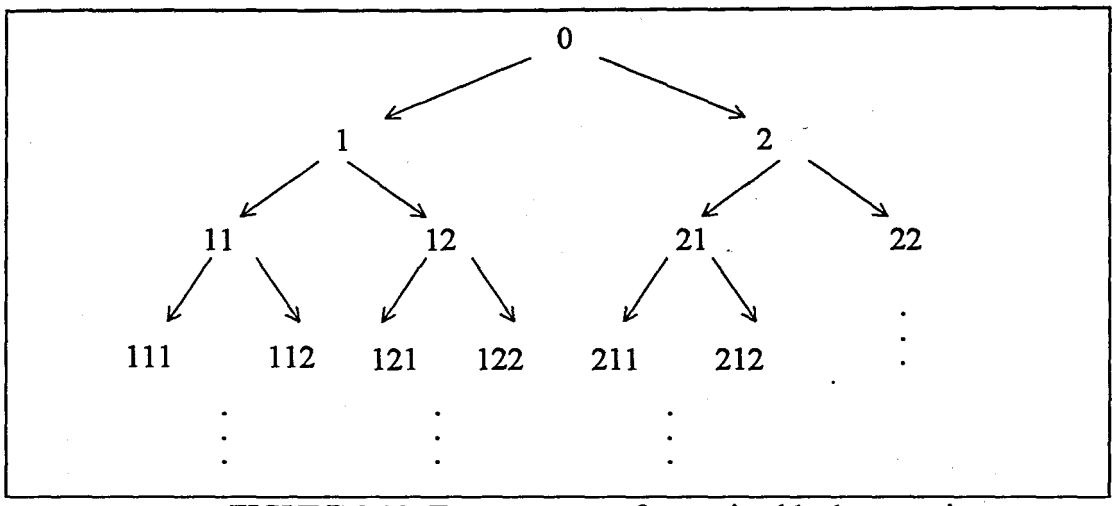


FIGURE 3.23. Tree structure of recursive block extraction.

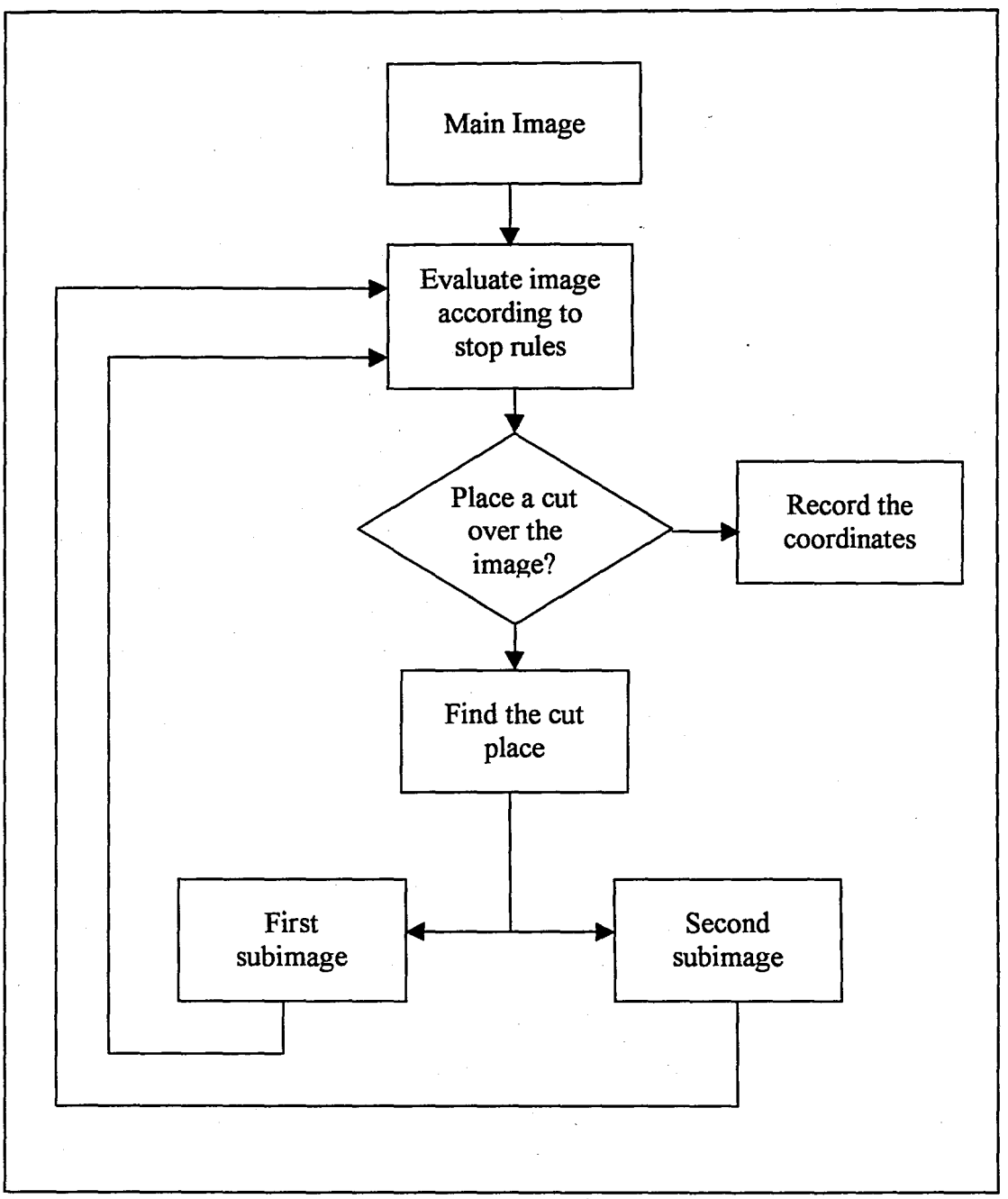


FIGURE 3.24. Algorithm flow chart

3.2. Bottom-Up Method

The blocks produced by RXYC are not unitype, i.e. do not always contain single type of document components, such as image, text block, header etc. If segmentation produces such blocks, that contains mixed type of component, then we have to reanalyze them. To understand whether a block unitype or multitype some features can be defined. These features could be size (width and height) of the block, aspect ratio, and the complexity of the block, which is defined as distribution of gray levels and the number of zero crossings within the block along the horizontal direction. After RXYC, all blocks can be labelled as unitype or multitype. If a block is multitype, then a bottom-up segmentation method should be deployed to segment different type of components. In this chapter such a bottom-up method is explained.

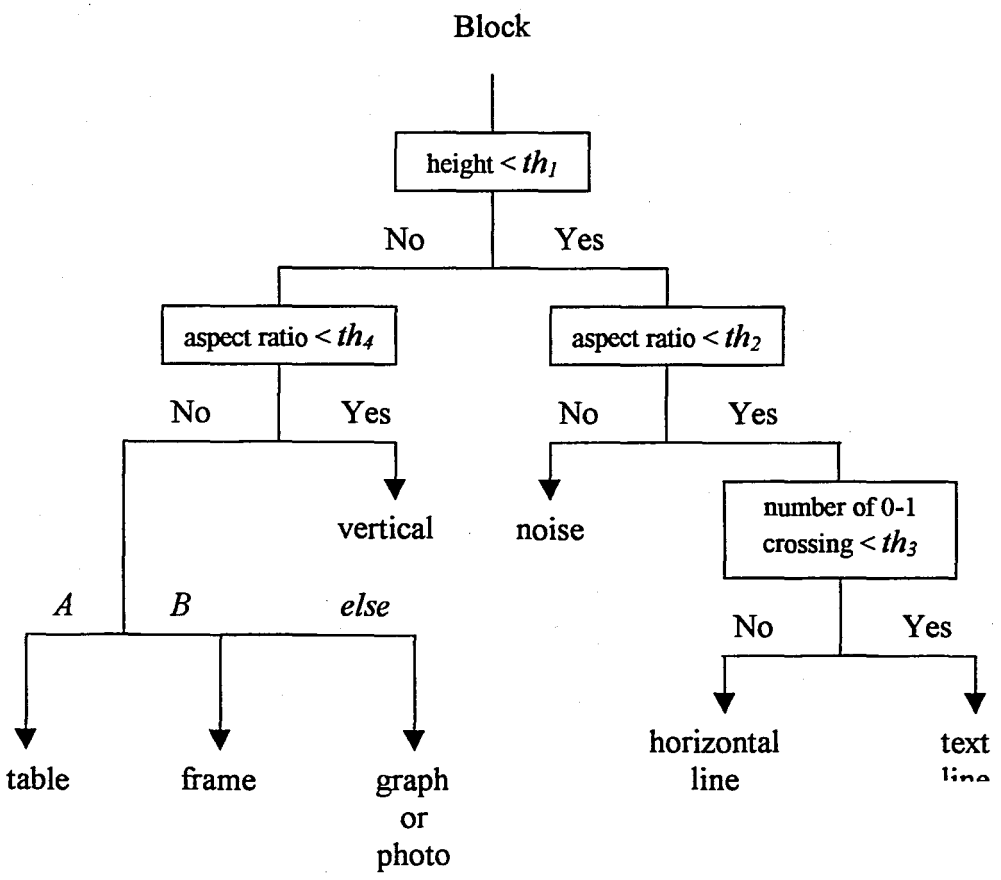
3.2.1. Segment Extraction

Initially to find out the black pixels which are closed to each other and can constitute a segment, a two step smearing is executed in the block. A horizontal smearing following a vertical smearing with different threshold values are executed. The two resultant images are then combined with logical "AND" operation. A second vertical smearing operation is performed on the resultant image to remove the small horizontal white stripes between line segments.

To find out the segments, the well-known connected component method is applied and meaningful blocks are produced. These blocks are generally the line segments within text blocks, the word segments within subtitles, the letters within headers, horizontal and vertical lines and image blocks etc. But this is not sufficient because the aim is to extract the text blocks, subtitles and headers completely. Therefore a merging operation should be executed. But we have to merge only segments of same type, so each segment should be labelled before merging operation.

3.2.2. Segment Classification

After extracting segments, they have to be labelled. Segments can be classified as text line, photograph, graphic, horizontal or vertical line or frame. The classification process is based on the physical properties of blocks. These are size of block (height and width), aspect ratio, horizontal and vertical projection profiles and frequency zero-one crossings. These properties utilize the both geometrical features and content of blocks. The threshold values for these properties may vary depending on the type of document. The classification process and the threshold values are given in Figure 3.25. The th_1 is threshold for height, th_2 and th_4 are for aspect ratio and th_3 is for zero-one crossing.



A : More than two peaks in projection profile
B : Two peaks in projection profile

FIGURE 3.25. Classification of blocks

This is not the final segmentation. The final one is done after merging blocks. Because, some blocks might be segmented incorrectly. Such as, dots ad commas and points in letters "İ", "j", "ü", "ö" and "ğ" can be classified as noise. To remedy this problem, these small blocks are labelled as text block.

3.2.3. Extracting Text Lines

Regarding the size of gaps between word segments with respect to the letter height, two neighbouring word segments that are closed to each other are merged into one. The merged word segments, then, constitutes the line segments.

Using the property that, line segments are left aligned we can constitute the columns. The boundaries of columns are extracted by taking the vertical projection profile of the left hand sides of the line segments. The local maximums in this profile correspond to the left hand sides of the columns. Following this process, the horizontally neighbouring line segments are merged to construct text stripes.

3.2.4. Text Block Extraction

The text stripes are then grouped to form the text blocks. A group number (**gn**) is assigned to each text stripes related with column position as in Equation 3.12.

$$\mathbf{gn} = \sum_{i}^{N} c[i] * 2^{(i-1)} \tag{3.12}$$

$c[i] = 1$ if text stripe resides in the i^{th} column
$c[i] = 0$ else
N total number of columns

After assigning a group number to each text stripe, the vertically neighbour ones with the same group number and close to each other within a threshold value are merged. As a result text blocks can be obtained. This operation is depicted in Figure 3.26. For the second time, a segment classification algorithm, as in Section 3.2.2, is executed over the blocks obtained from the merging operation. As a result, after these operations we can get the type and location of blocks within the page. The steps of this bottom – up method are shown in Figure 3.27 [9].

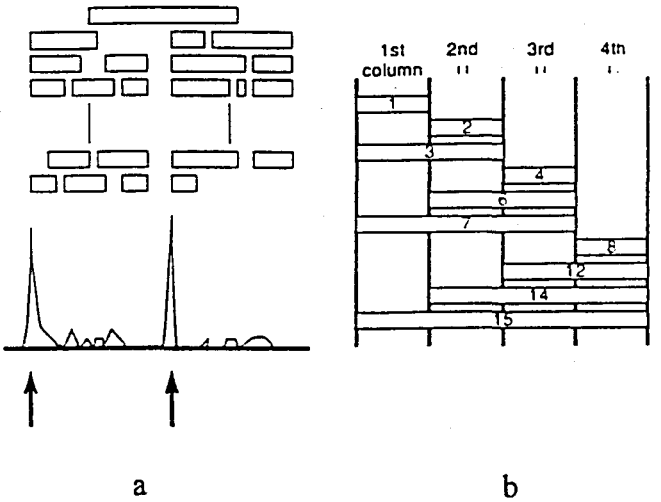
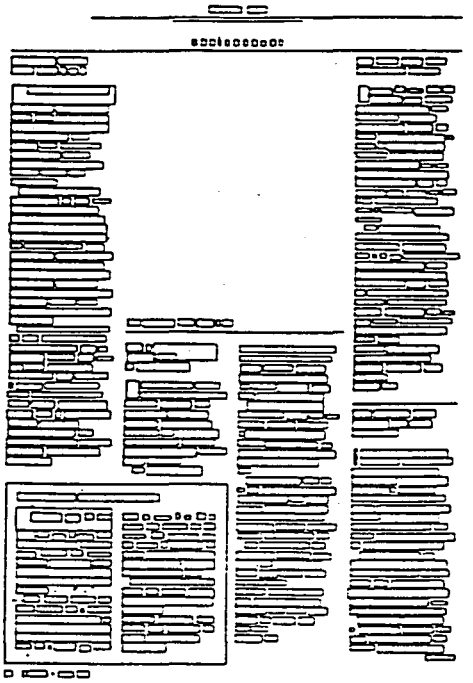
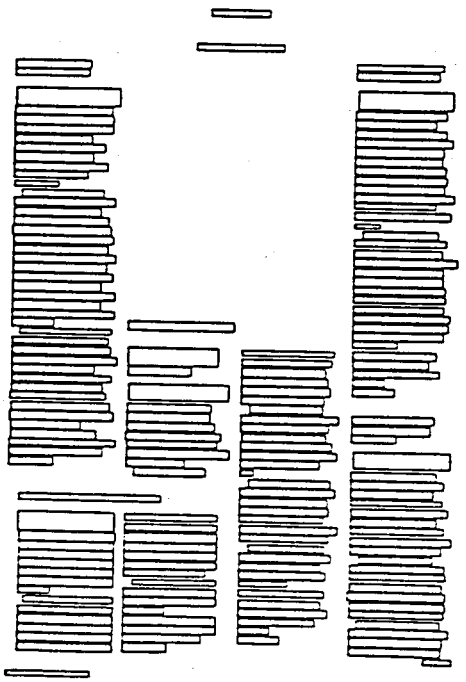


FIGURE 3.26. a) How to determine the column position.
b) Group number, depending on column number

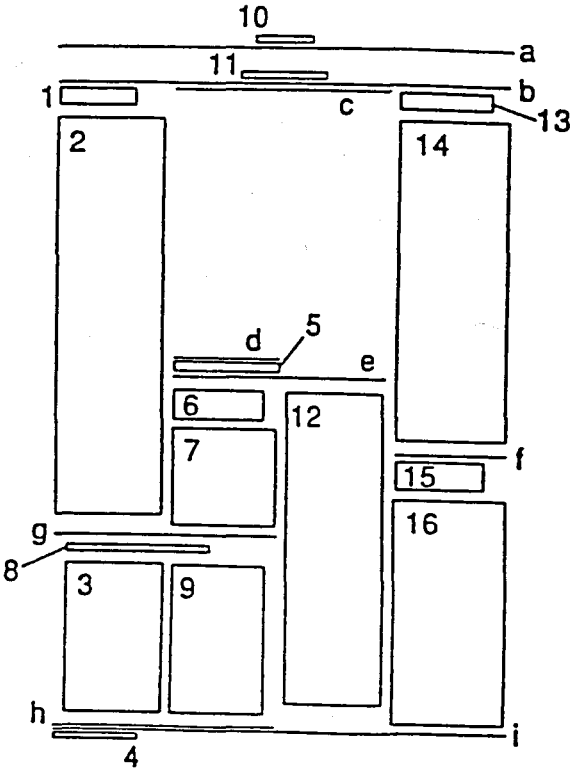
a)



b)



c)



d)

FIGURE 3.27. Block extraction [9]. a) Original document image. b) Segment extraction result. c) Text line extraction result. d) Block extraction result.

4. EXPERIMENTAL RESULTS

The performance of the hybrid RXYC method can be evaluated according to speed and result of segmentation process, i.e. number of extracted blocks and correctness of the cuts placed over document.

Compared to bottom-up methods, the process time is quite little for top-down methods. This generic feature is valid for our method. The process time, of course, vary greatly depending on the page layout. The smaller the number of blocks, that can be extracted by hybrid RXYC method, the less time required for the process. Also page size and thresholds in smearing operation definitely affect the process time. But within our test pages, the longest time was consumed by the test image in Figure 4.8.a and it took 15 seconds. This page is quite suitable for our segmentation process and almost all of the blocks within this test page are correctly extracted by the hybrid RXYC method. Although we do not have exact figures for the process time for the bottom-up method in Section 3.2, these figures are around one minute for a page of similar layout. The process time for all test images are within a range of eight to 15 seconds. If the page is not proper for the hybrid RXYC method, as in Figure 4.5.a, the length of process time gets smaller, but as a trade-off only a few number of blocks are extracted. Regarding all these issues, our method can be labelled as a fast segmentation process.

We also evaluated the horizontal and vertical smearing thresholds to find out most proper values. In the original thresholded image of 75 dpi, the number of black pixels is 21.9 percent of the number all of pixels. We applied several different thresholds for both smearing operation and find out the variation of number of black pixels in the smeared image. Naturally, as thresholds get bigger the percentage of black pixels within the whole page increases. This ratio for different threshold values is given in Table 4.1.

TABLE 4.1. The variation of black pixel percentage according to vertical and horizontal smearing thresholds.

		Vertical Smearing Threshold				
		4	8	12	16	20
Horizontal Smearing Threshold	4	33.4	41.0	44.8	48.5	52.1
	8	36.3	43.3	46.7	50.2	53.5
	12	37.9	44.5	47.7	51.0	54.2
	16	40.2	46.5	49.5	52.7	55.8
	20	42.3	48.2	51.2	54.3	57.3

We got bigger black blocks as threshold values increase and more proper page layouts are obtained for efficient segmentation. But, the following problems were observed. After the vertical threshold of eight the gaps between the text blocks are smeared out and no gaps were left for separating these text blocks. When the horizontal threshold value is bigger than 12 we can obtain the headings as one block. But this time almost all of the vertical lines and some of the column gaps were smeared to the adjacent blocks. On the other hand, when we chosen both thresholds too small, then some image blocks and text blocks were improperly segmented into two separate blocks. Regarding all these issues, we have chosen 10 for both horizontal and vertical thresholds and obtain sufficient results.

Another issue that can affect the segmentation result is valley appropriateness. As explained in Section 3.1.4.2, a block can be segmented in any direction if enough wide and deep valleys exist in the profile. The evaluation of valleys at this step depends on two parameters. The first one is the threshold value, taken as the percentage of height, for valleys in vertical projection profile and that of width for valleys in horizontal projection profile. This threshold used to test the depth of valleys. The number of sequential values less then this threshold is used to test the width of valleys. So we can use different values for these two parameters. We segmented the same

smeared image for the several combinations of these parameters and find out the number of blocks for each. The results are given in Table 4.2.

TABLE 4.2. The number of blocks after segmentation according to different valley depth and valley width thresholds.

		Width threshold as number of sequential values in profile less then depth threshold		
		4	7	10
Depth threshold as percentage of size (height - width)	3	56	51	11
	6	68	68	53
	10	76	75	68

When we choose the percentage value too small, algorithm searches for very "clear" valleys along the projection line. Because of printing and scanner errors and some blackened pixels after smearing operation there may not exist sufficiently "clear" valleys along any projection. But these artificial "dirts" may prevent algorithm to place a proper cut over the image. On the other hand, if it is chosen too big, than some inappropriate valleys can be evaluated as "proper valley" and faulty segmentation may occur.

When evaluating valleys' width, algorithm counts the number of sequential values less than depth threshold and if it exceeds a certain value, then block is labelled as appropriate and a cut can be placed over it. Same results for depth thresholds are obtained for different width thresholds. If width threshold is too big, then algorithm searches very wide valleys and since this is not satisfied for every valley, such as ones between text block and surrounding lines, the number of extracted blocks decreases. If it is chosen too small the number of extracted block does not increase drastically, only some faulty segmentation such as separating the "S" and "," in letter "Ş" may occur.

After testing results for different combination of width and depth threshold values we have chosen seven for width threshold and six for depth threshold.

The last issue for evaluation is the size of the block. During segmentation process we always compare the height and width of the block with a threshold value. The threshold value for size check is a proportion of the whole page size. Based on our observations for Turkish newspaper, there may be six to 10 columns within a page and the height of text blocks are around the one 10^{th} and one 15^{th} of page the page height. We tested the different values for both and obtained the results in Table 4.3.

TABLE 4.3. The number of blocks after segmentation according to different height and width thresholds for block size.

		Width threshold as a proportion of page width		
		1/6	1/8	1/10
Height threshold as a proportion of page width	1/10	68	86	93
	1/12	72	89	96
	1/15	75	96	103

As can be seen in Table 4.3, the variations in width threshold affect the number of extracted blocks more than that in height threshold. But when we chose 1/10 of page width as width threshold, the increase in the number of blocks are due to faulty segmentation or segmentation of horizontal lines into two or three line blocks. In order to avoid faulty segmentation we have chosen 1/6 ratio for width segmentation. Height threshold did not have such an impact on segmentation results and to extract the maximum number of blocks we have chosen 1/15 as height threshold ratio.

Here we tested six different Turkish newspaper pages. There are both first pages and inner pages of newspapers. The generic feature of first page is that it includes more images than inner pages and the font size shows great variety within the page.

Our input image is scanned newspaper images. The input image must be skew-corrected and thresholded. The examples through Figure 4.1 – 4.8 shows the original gray level newspaper page, block boundaries extracted by the RXYC and the corresponding segments on the newspaper page. The pages, that are proper for our RXYC method, are segmented correctly and results are satisfactory. For those which are not appropriate for RXYC, the algorithm stops at the expected blocks and does not cause faulty segmentation.


4 HABERLER

KUTLU SAVAŞ'IN HAZIRLADIĞI SUSURLUK RAPORUNUN TAM METNİ - 3

Skandal başladı, faili meçhul bitti

B

Bankalardaki ürkütücü tablo

B

PERŞEMBE

Güneşlioğlu'da 13 gazeteci çıkarıldı

'Susurluk bir olaylar zincirinden ibarettir'

G

FIGURE 4.1.a. Test image 1

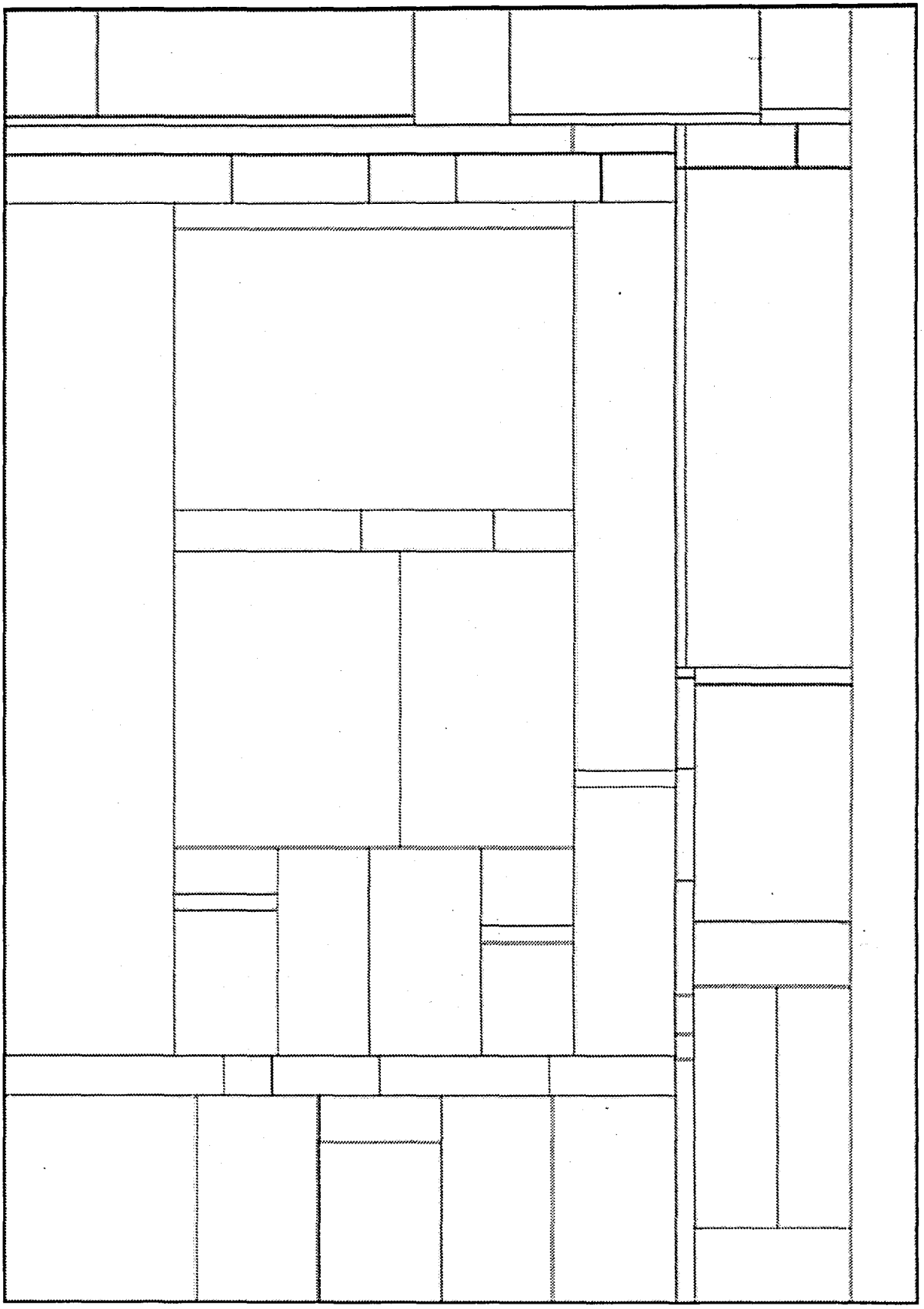


FIGURE 4.1.b. The extracted blocks from test image 1

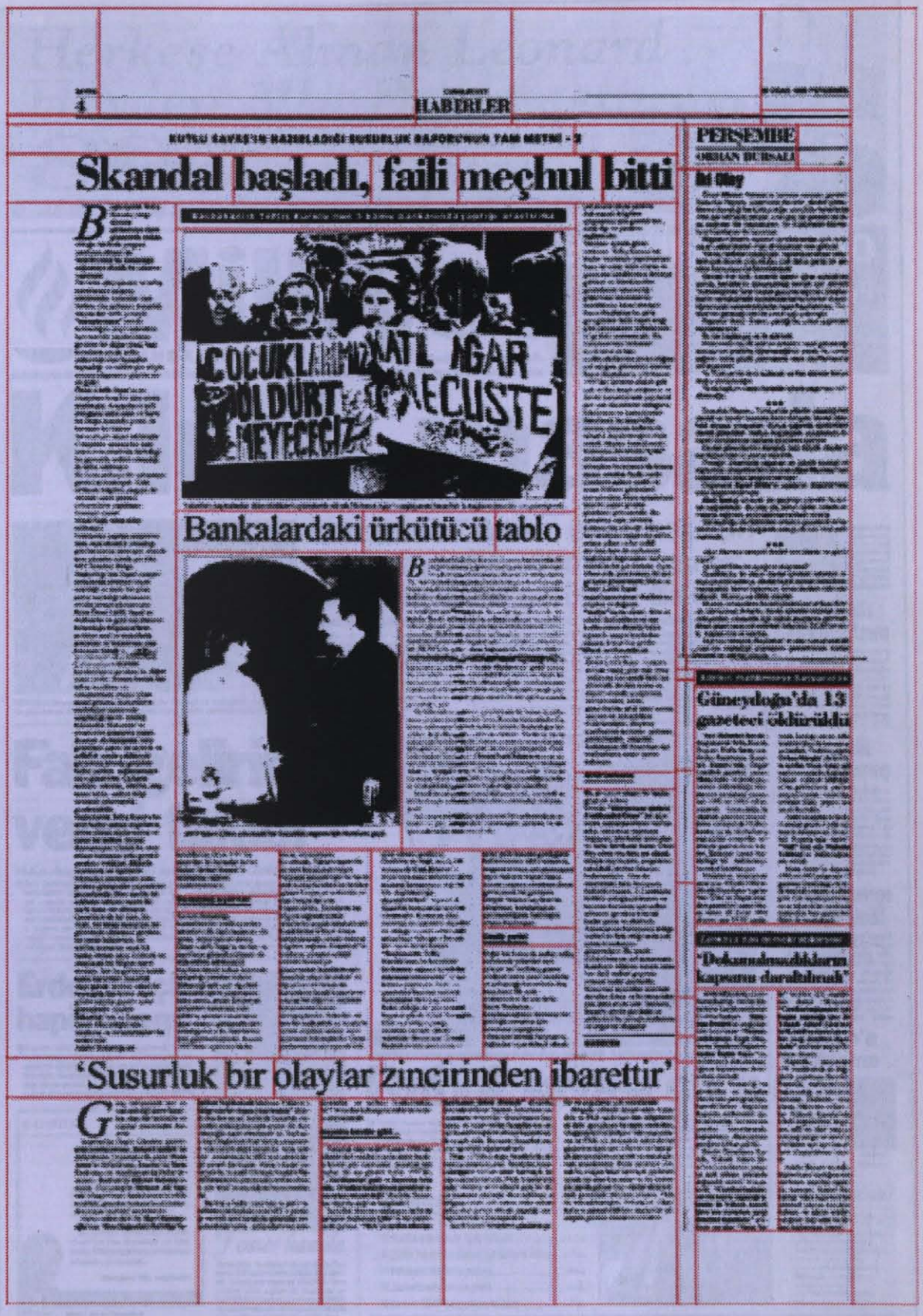


FIGURE 4.1.c. Location of blocks on the test image 1

FIGURE 4.2.a. Test image 2

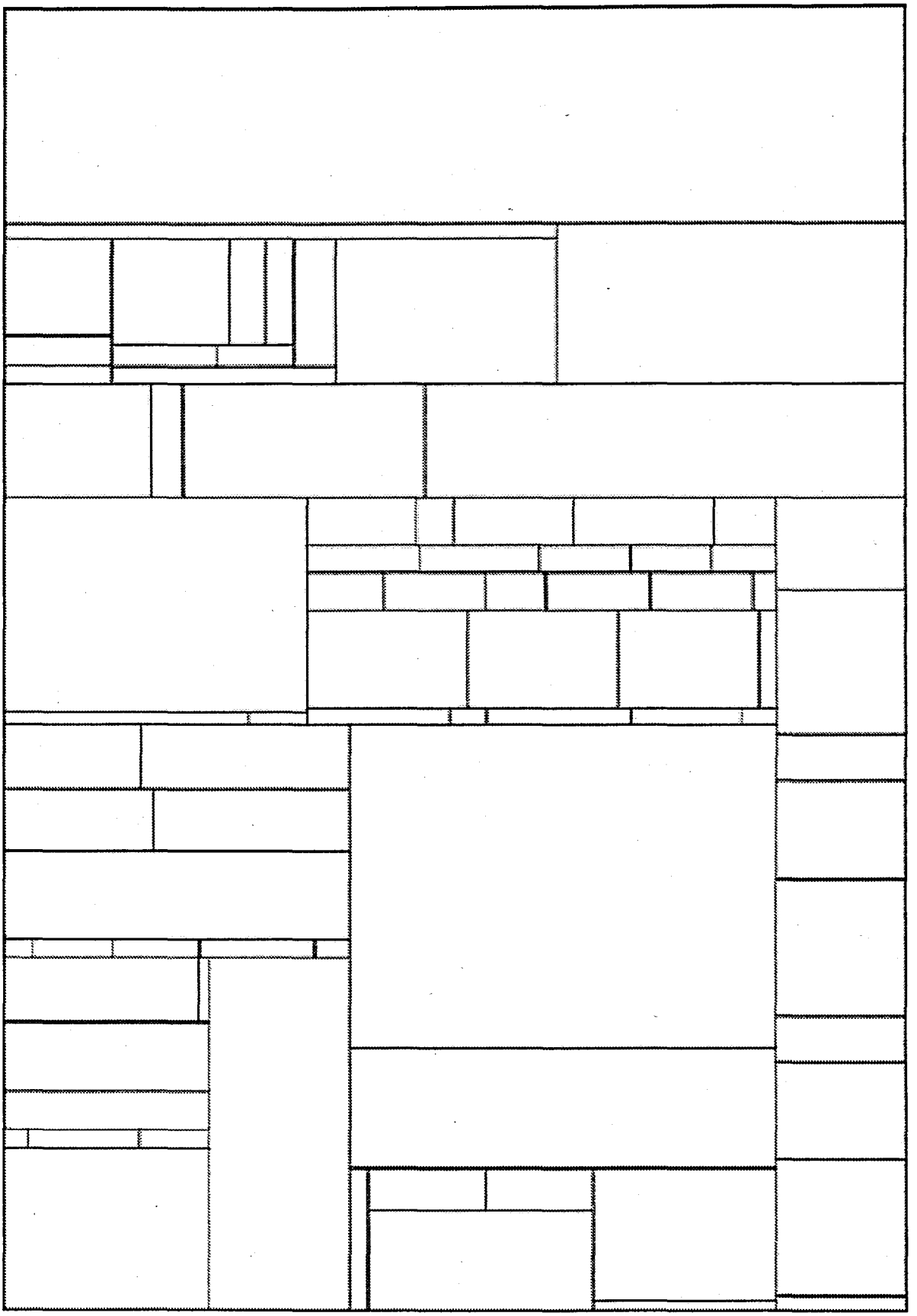


FIGURE 4.2.b. The extracted blocks from test image 2

FIGURE 4.2.c. Location of blocks on the test image 2

FIGURE 4.3.a. Test image 3

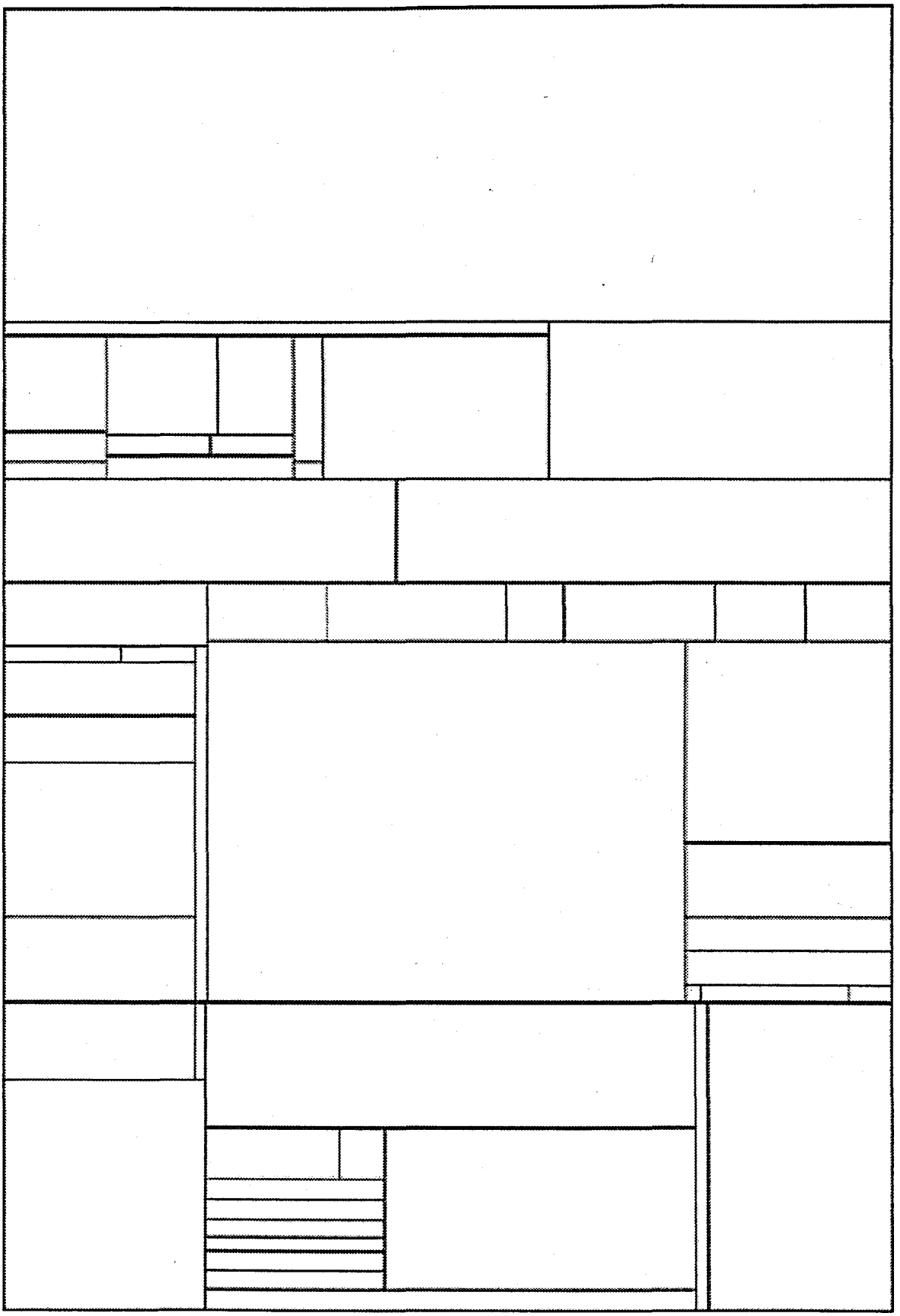


FIGURE 4.3.b. The extracted blocks from test image 3

FIGURE 4.3.c. Location of blocks on the test image 3

FIGURE 4.4.a. Test image 4

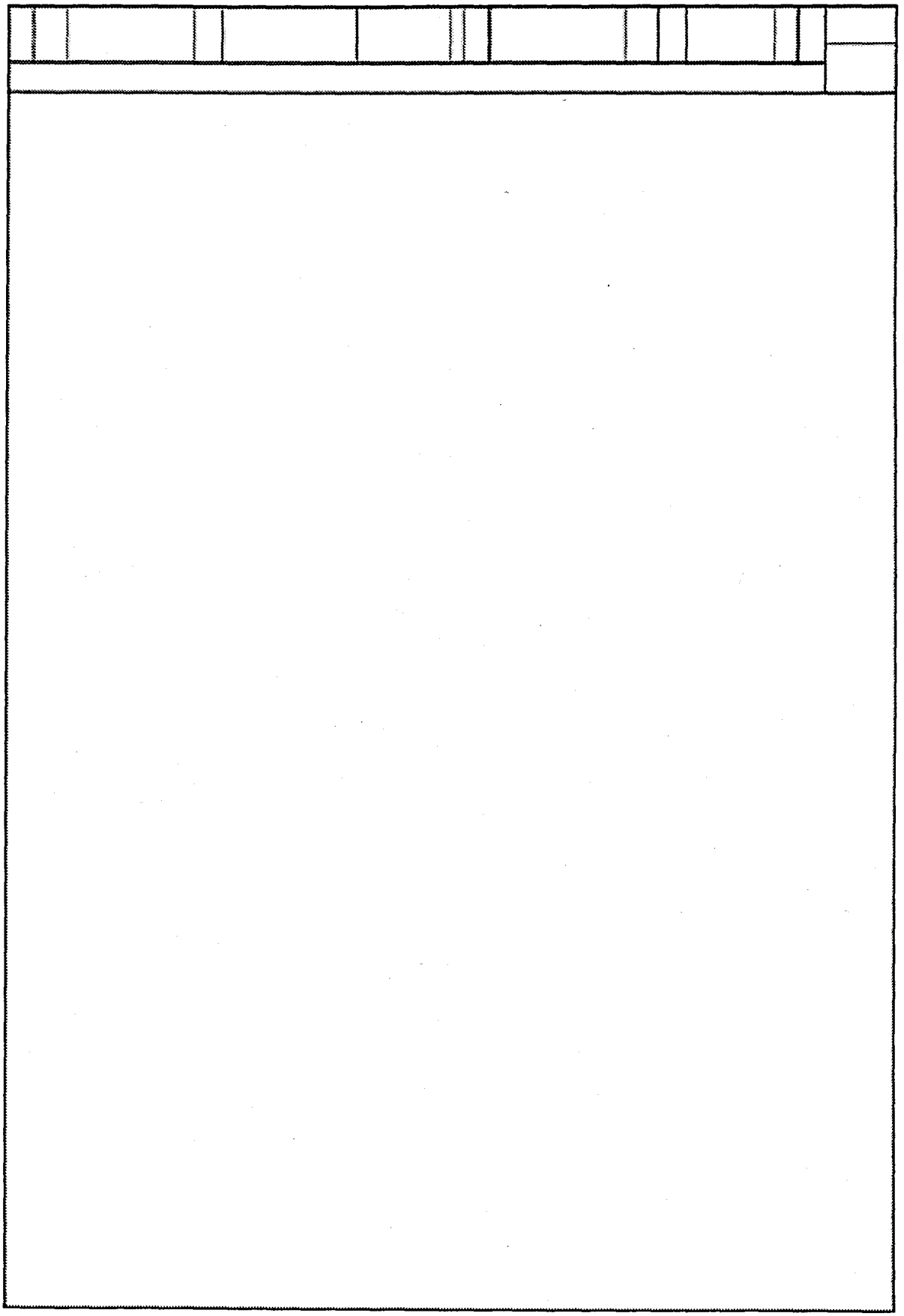


FIGURE 4.4.b. The extracted blocks from test image 4

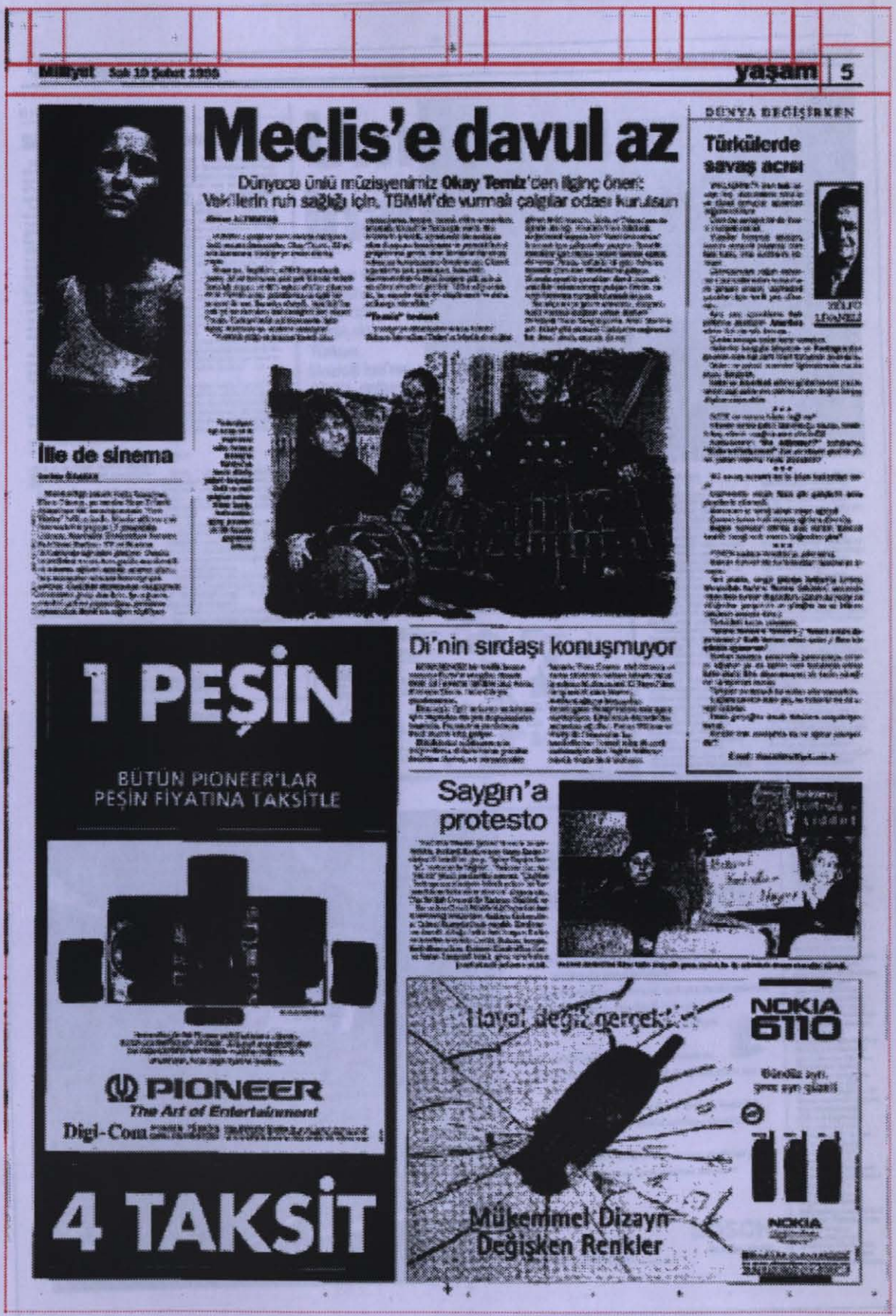


FIGURE 4.4.c. Location of blocks on the test image 4

FIGURE 4.5.a. Test image 5

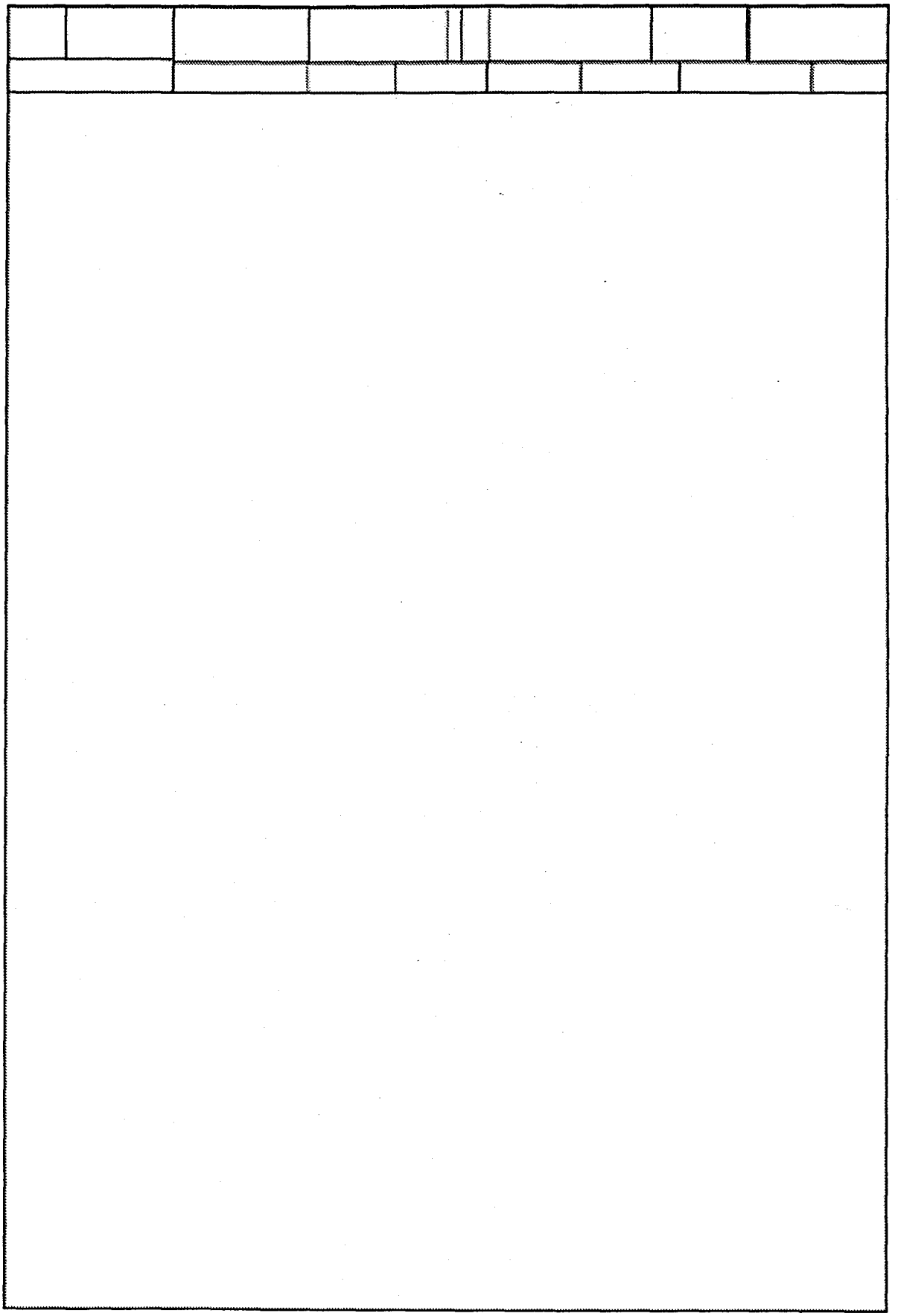


FIGURE 4.5.b. The extracted blocks from test image 5

FIGURE 4.5.c. Location of blocks on the test image 5

FIGURE 4.6.a. Test image 6

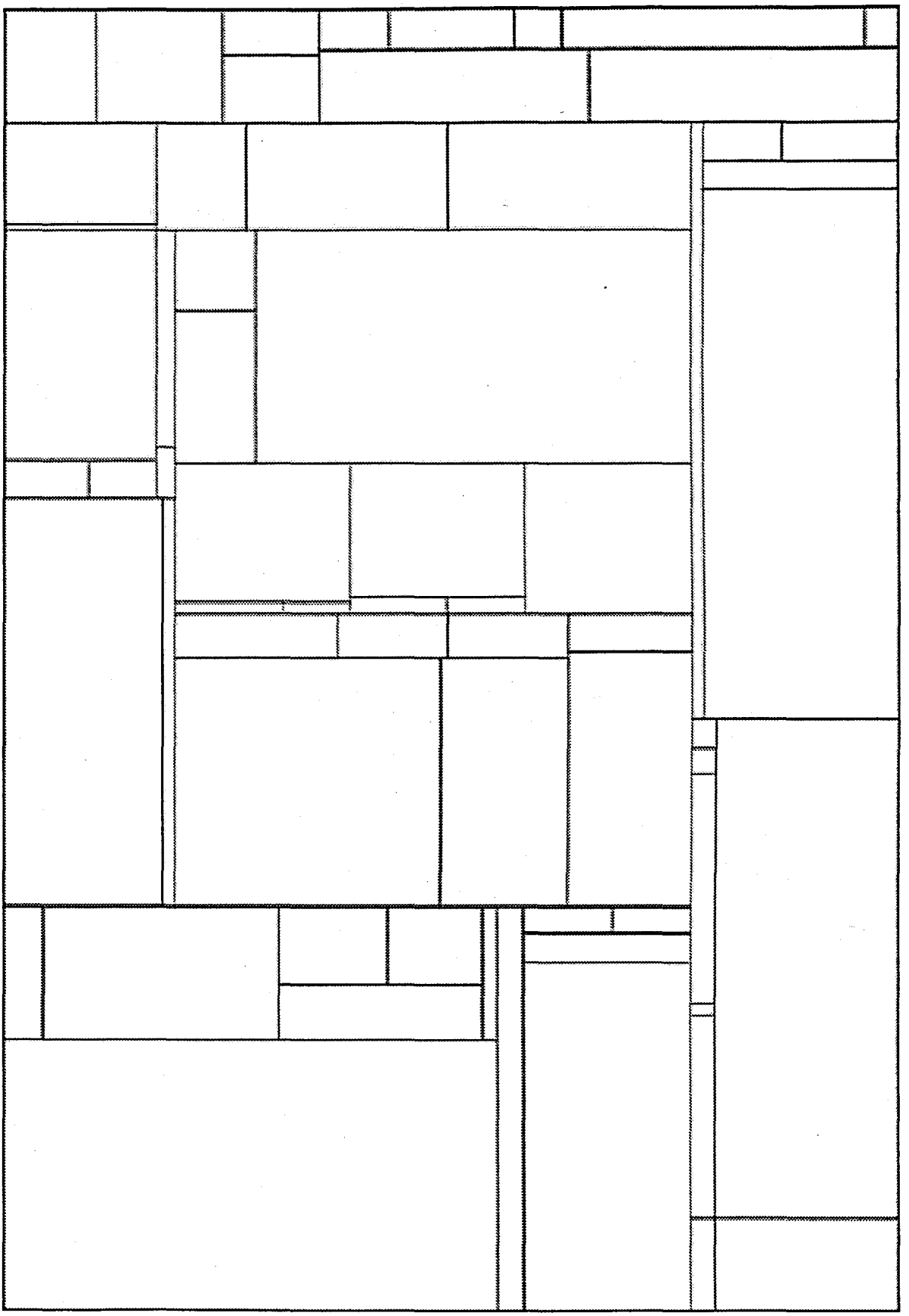


FIGURE 4.6.b. The extracted blocks from test image 6

FIGURE 4.6.c. Location of blocks on the test image 6

FIGURE 4.7.a. Test image 7

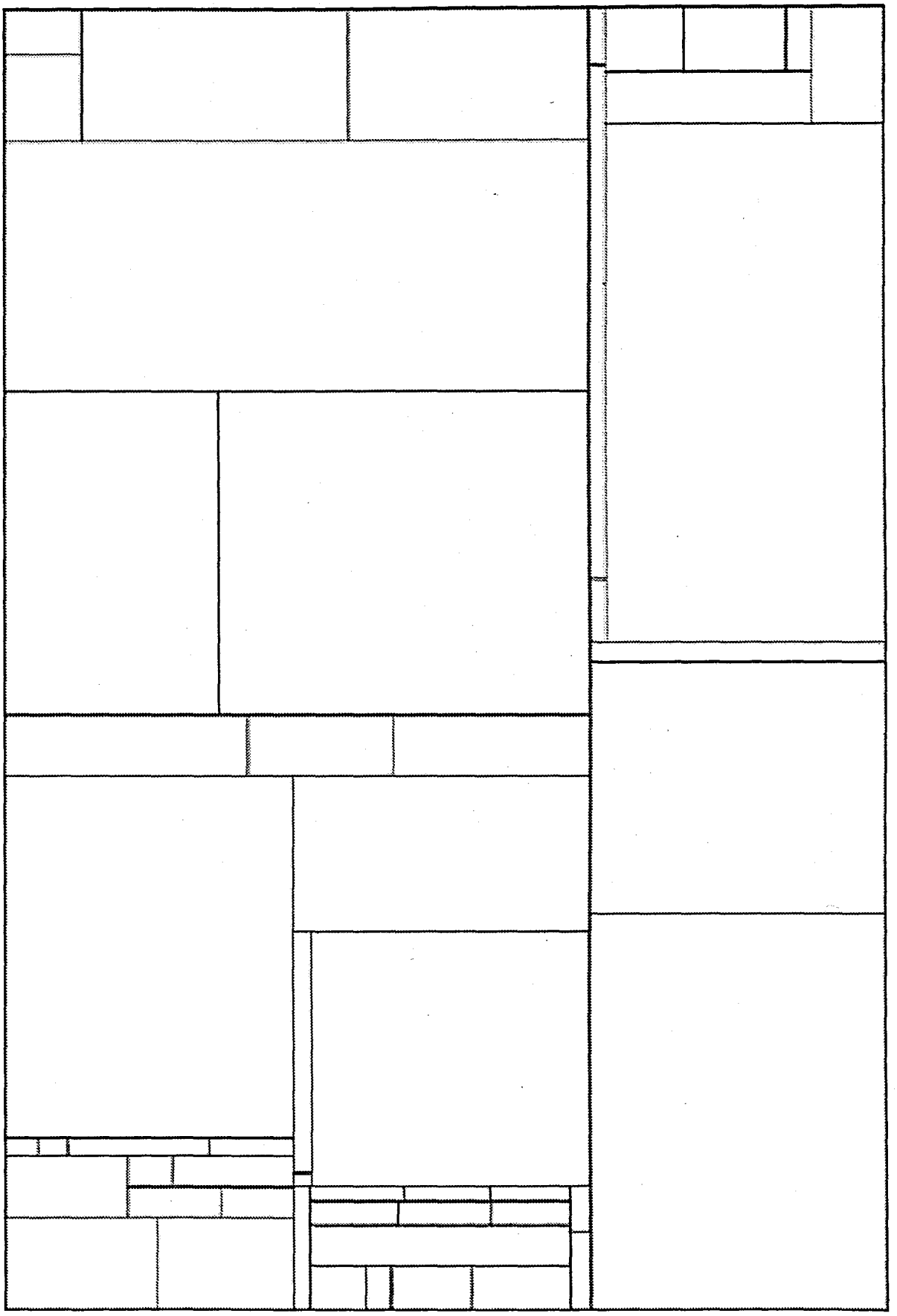


FIGURE 4.7.b. The extracted blocks from test image 7

FIGURE 4.7.c. Location of blocks on the test image 7

Milliyet Salı 10 Şubat 1998 7

ekonomi

ŞİRKET HABERLERİ

Gümüşpuryu tasarımda dünya birincisi

Kellogg Europe "Hill & Knowlton" ile çalışıyor

Osmanlı Bankası Kuzey Kıbrıs'ta

Gillette Sensor Excel yeni fiyatıyla

Dikom'a yeni genel müdür

Kamu, 'lira'yı gözden çıkardı

Hiperenflasyon tartışma konusu

'Sosyal patlama olabilir... Dikkat'

'İlk aylar yüksek olur'

Yılmaz: Hedeften vazgeçmedik

'Ekonomi yönetimi başarısız'

Küresel Türk kaplanı seçiliyor

Vergi tasarısı Komisyon'da

FIGURE 4.8.a. Test image 8

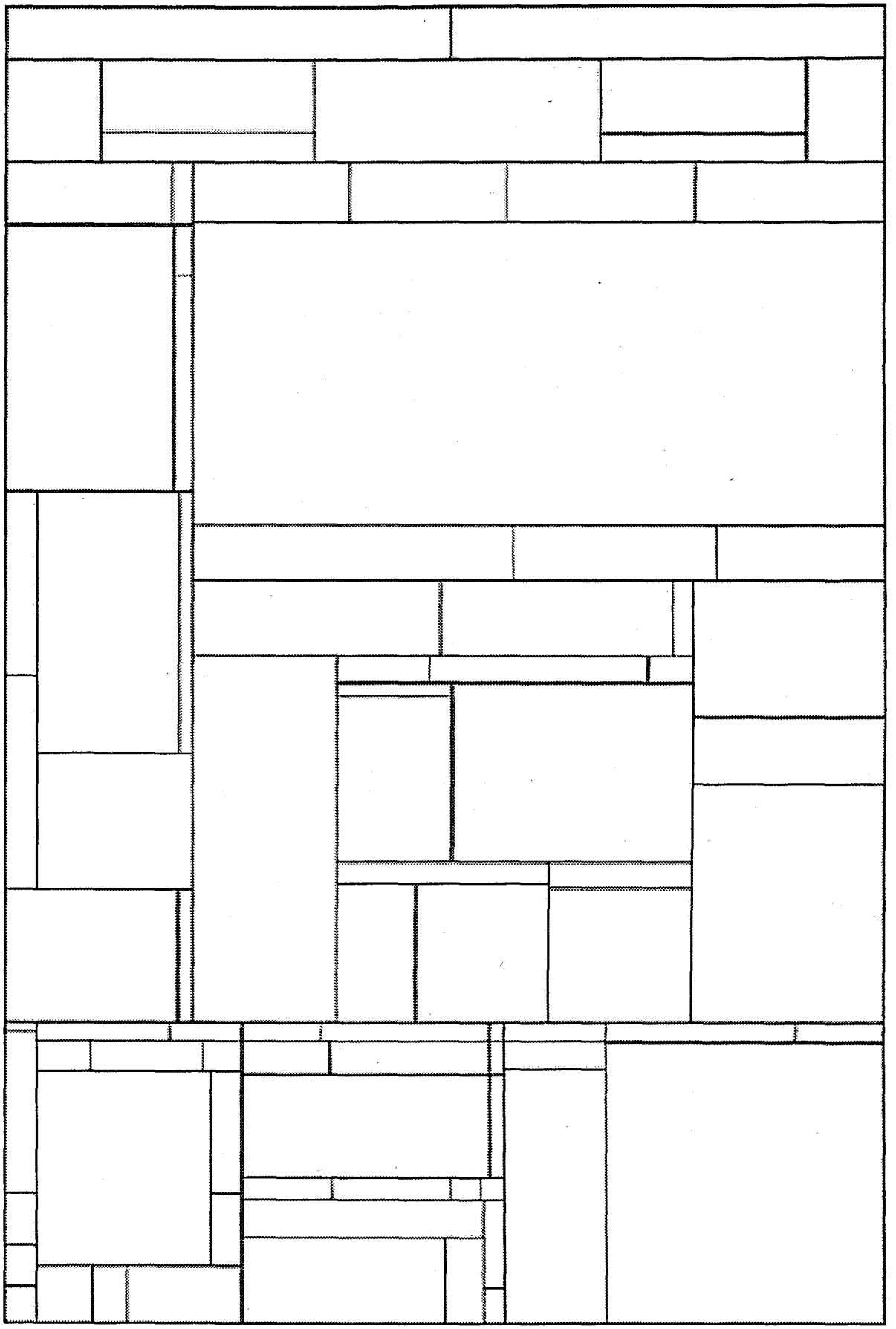


FIGURE 4.8.b. The extracted blocks from test image 8

FIGURE 4.8.c. Location of blocks on the test image 8

5. CONCLUSION AND THE FUTURE WORK

The modified RXYC method works efficiently on most of Turkish newspaper pages. These pages can be segmented down to desired level i.e. extraction of paragraph blocks either subtitles, lines and images. But some over segmentation cases may occur, such as dividing a single header down to word level even sometimes letter level; or segmenting a single line into two or more line pieces. A merging algorithm can be used to remedy this problem. Such merging algorithm can utilize the locations of header segments, which shows linearity emerging from the structure of RXYC method. At the same time, the gray level distribution, number of zero crossing and aspect ratio of over segmented blocks support the merging algorithm.

On the other hand some pages that have distributed column structure can not be segmented down the desired level; although small number of proper blocks are extracted from the document image. In such cases proposed bottom-up segmentation method can be utilized on the not segmented part of the document block. These two approaches, one from top-down and one from bottom-up direction support each other and may produce sufficient results.

In the sense of speed, the modified RXYC method produced very satisfactory results. Segmenting a first or an inner page of a newspaper never exceeded ten seconds. This result can be graded as quite well regarding to other techniques.

As a conclusion, if a well designed and conforming post-processing technique is deployed with the modified RXYC, successful segmentation of Turkish newspaper pages can be achieved in great extent. Adjusting some threshold values and processes within the modified RXYC method can make it possible to use it for the segmentation of other type of document.

REFERENCES

1. Tang Y.Y., S.W. Lee, and C.Y. Suen, "Automatic Document Processing: A Survey," *Pattern Recognition*, Vol. 29, No.12, pp.1931-1952, 1996

2. Srihari S.N., S.W. Lam, V. Govindaraju, R.K. Srihari, and J.J. Hull, "Document Image Understanding," *http://www.cedar.buffalo.edu/Publications/TechReps/Survey/Survey.html.*

3. Pavlidis, T., and J. Zhou, "Page Segmentation and Classification," *CVIP: Graphical Models and Image Processing*, Vol 54, No 6, pp 484-496, 1992.

4. Witten, I.H., A. Moffat, and T.C. Bell, *Managing Gigabytes*, NewYork, Van Nostrand Reinhold, 1994.

5. Wang, D., and S.N. Shriari, "Classification of Newspaper Image Blocks Using Texture Analysis", *Computer Vision, Graphics and Image Processing*, Vol 47, pp 327-352, 1989.

6. Fan K.C., C.H. Liu, and Y.K. Wang, "Segmentation and Classification of Mixed Type Text / Graphics / Image Documents," *Pattern Recognition Letters*, Vol. 15, pp 1201-1209, December 1994.

7. Zlatoposky, A.A., "Automated Document Segmentation," *Pattern Recognition Letters*, Vol 15, pp 699-704, July, 1994.

8. Jain, A.K., and Y. Zhong, "Page Segmentation Using Texture Analysis", *Pattern Recognition*, Vol 29, No 5, pp 743-770, 1996.

9. Tsujimoto S., and H Asada, "Major Components of a Complete Text Reading System," *Proceedings of The IEEE*, Vol. 80, No. 7, pp. 1133-1149, 1992.

REFERENCES NOT CITED

Çakır, F. *Dikdörtgen Blok Yaklşımı ile Belge Analizi*. Sinyal İşlemleri Uygulamaları, 1997.

Eickel, J. *Logical and Layout Structures of Documents*. Computer Physics Communications 61, North-Holland, 1990.

Fan, K.C., L.S. Wang, Y.K. Wang. *Page Segmentation and Identification for Intelligent Signal Processing*. Signal Processing 45, 1995.

Fung, H. T. "Efficient Segmentation and Compression of Scanned Document Images". Ph.D. Dissertation, University of Rochester, 1996.

Scürmann, J., N. Bartneck, T. Bayer, J. Franke, E. Mandler, and M. Oberlander. *Document Analysis – From Pixels to Content.* Proceedings of the IEEE, Vol.80, No.7, 1992.