COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 70, No. 3, June, pp. 370–382, 1998
ARTICLE NO. IV980684

Segmentation of Page Images Using the Area Voronoi Diagram

Koichi Kise,* Akinori Sato, and Motoi Iwata

Department of Computer and Systems Sciences, College of Engineering, Osaka Prefecture University, Osaka, Japan

Received February 12, 1997; accepted December 15, 1997

This paper presents a method of page segmentation based on the approximated area Voronoi diagram. The characteristics of the proposed method are as follows: (1) The Voronoi diagram enables us to obtain the candidates of boundaries of document components from page images with non-Manhattan layout and a skew. (2) The candidates are utilized to estimate the intercharacter and interline gaps without the use of domain-specific parameters to select the boundaries. From the experimental results for 128 images with non-Manhattan layout and the skew of 0°∼45° as well as 98 images with Manhattan layout, we have confirmed that the method is effective for extraction of body text regions, and it is as efficient as other methods based on connected component analysis. © 1998 Academic Press

1. INTRODUCTION

Layout analysis is the process of identifying the layout structure by analyzing page images. This is often subdivided into two tasks: page segmentation and page classification. The task of page segmentation is to extract regions of document components such as text, figures, tables, and halftones. On the other hand, that of page classification is to identify the type of each extracted region. The performance of a page segmentation method is crucial for succeeding steps of document image understanding including page classification, since it provides fundamental data for these steps and thus dominates their performance.

Methods of page segmentation have been elaborated in pursuit of accuracy and efficiency within several classes of page layout. In general, they display better performance for more restricted classes.

Page layout can be broadly divided into two classes: overlapping and nonoverlapping. Pages with overlapping layout include a document component whose region overlaps with those of others, while in nonoverlapping layout, document components are separated by white space. We are concerned here with nonoverlapping layout.

An important subclass on nonoverlapping layout is *rectangular* layout. The layout is rectangular if all document components can be circumscribed by rectangles whose sides are parallel or perpendicular with one another. Most journal papers, some magazines, and some textbooks have this subclass of layout. Another important and broader subclass is *Manhattan* layout. The layout is Manhattan if the regions have boundaries consisting of line segments parallel or perpendicular with one another. The class of Manhattan layout, which includes rectangular layout as its subclass, covers most printed pages.

For pages within the class of Manhattan layout, a large number of methods have so far been proposed [1, 2]. Some, including the method by Baird *et al*. [3], accomplish high accuracy as well as efficiency by normalizing the skew of page images beforehand. In the case that page images have no skew, the task of segmentation is simplified to find boundaries consisting of vertical and horizontal line segments.

In recent years, however, the number of pages whose layout is beyond the class of Manhattan layout, i.e., pages including document components of arbitrary shape, seems to be growing as designers aim to make pages attractive. We refer to such layout as *non-Manhattan* layout. In the pages with non-Manhattan layout, tilted text regions are sometimes contained for highlighting from other upright areas. This makes it difficult to deskew page images. A reasonable way of analyzing such images is to deskew text regions individually after the extraction of document components. Therefore, development of algorithms which are capable of dealing with page images with non-Manhattan layout and a skew becomes one of the central problems of page segmentation.

Until now, some researchers have attempted to overcome the problem. Jain and Bhattacharjee have proposed a method based on texture analysis [4]. Connected component analysis is in common use for images with non-Manhattan layout as well as a skew [5–7]. In recent years, the analysis of background (the white area of a page) has been actively applied to such images [8–10].

In this paper, we propose a method of page segmentation for page images with non-Manhattan layout and a skew as well. Our method is also based on the connected component analysis but has the following characteristics. For page images with either non-Manhattan layout or a skew, boundaries of document components cannot be represented by vertical and horizontal line segments; line segments of arbitrary orientation and length need to be utilized. In order to obtain candidates of such line segments efficiently, we employ the *approximated area Voronoi diagram* [11, 12] which represents the neighborhood of connected components as polygons. Based on this representation, the process

* E-mail: kise@cs.osakafu-u.ac.jp.

1077-3142/98 $25.00

of page segmentation can be considered to be the selection of appropriate line segments as boundaries of document components. For this purpose, we utilize two characteristic features: the Euclidean distance and the area ratio calculated from a pair of connected components that are neighbors in the sense of the Voronoi diagram. Intercharacter and interline gaps are estimated from the Voronoi diagram to determine adaptively the thresholds of distance under the assumption that body text regions are dominant in a page.

In this paper, we also discuss the advantages and limitations of our method based on the experimental results for 128 images with non-Manhattan layout and skew angles from 0° to 45°, as well as 98 images with Manhattan layout. In addition, we evaluate our method in relation to other methods applicable to pages with non-Manhattan layout and a skew.

2. VORONOI DIAGRAMS

In the field of image processing, including document image analysis, Voronoi diagrams have sometimes been used as the representation of input images. Burge and Monagan have recently applied a Voronoi diagram to extract words and symbols from map images [13].

To begin, we describe a short review of the definitions of Voronoi diagrams. For further details, see [11, 12].

2.1. Point Voronoi Diagram

Let $P = \{p_1, \ldots, p_n\}$ be a set of points or *generators* in the two-dimensional plane, and $d(p, q)$ be the Euclidean distance between points p and q. A *Voronoi region* of a point p_i is the region given by

$$V(p_i) = \{p \mid d(p, p_i) \leq d(p, p_j), \forall j \neq i\}. \tag{1}$$

The *ordinary Voronoi diagram* generated from a set of points P is given as a set of Voronoi regions

$$V(P) = \{V(p_i), \ldots, V(p_n)\}. \tag{2}$$

In this paper, we call this the *point Voronoi diagram*.

The boundaries of Voronoi regions may consist of line segments, half lines, or infinite lines, which are called *Voronoi edges*. A *Voronoi point* indicates a point where Voronoi edges come in contact.

A variety of methods have been proposed for the construction of the point Voronoi diagram. It is known that the construction requires $O(n \log n)$ time, where n is the number of generators, but on the average the time is proportional to n.

2.2. Area Voronoi Diagram

The point Voronoi diagram has been generalized in many directions. We focus here on the generalization of generators from points to figures of arbitrary shape. Such a Voronoi diagram is sometimes called the *area Voronoi diagram*.

Let $G = \{g_1, \ldots, g_n\}$ be a set of nonoverlapping figures in the two-dimensional plane, and let $d(p, g_i)$ be the Euclidean distance between a point p and a figure g_i defined as

$$d(p, g_i) = \min_{q \in g_i} d(p, q), \tag{3}$$

where q is a point in g_i. Then the Voronoi region $V(g_i)$ and the area Voronoi diagram $V(G)$ are defined as

$$V(g_i) = \{p \mid d(p, g_i) \leq d(p, g_j), \forall j \neq i\} \tag{4}$$

$$V(G) = \{V(g_1), \ldots, V(g_n)\}. \tag{5}$$

Voronoi edges and Voronoi points are defined in a similar way for the point Voronoi diagram.

It is known that the area Voronoi diagram can be approximately constructed by applying an algorithm for the point Voronoi diagram in the following manner.

Step 1. Let $P_i = \{p_{i1}, \ldots, p_{im_i}\}$ be a set of points lying on the boundary of a figure g_i.

Step 2. Generate the point Voronoi diagram from the generators $P = P_1 \cup \ldots \cup P_n$.

Step 3. For all i, j, k, delete Voronoi edges generated from points p_{ij} and p_{ik}, i.e., Voronoi edges generated from points on the same figure.

Although a Voronoi edge of the area Voronoi diagram can be a complex curve, that of the approximated version consists of line segments, which are the approximation of the curve. In what follows, the approximated version is referred to as the area Voronoi diagram for simplicity.

2.3. Voronoi Edges in Document Images

We can construct the point Voronoi diagram from a document image by, for instance, using the centroids of connected components. This approach has been applied to document image processing; Ittner *et al.* have estimated text-line orientation using the minimum spanning tree [14], which is a subset of the Delaunay triangulation, i.e., the dual of the point Voronoi diagram.

However, the point Voronoi diagram is unsuitable for the task of page segmentation, since the approximation of each connected component as a single point is too imprecise to represent the structure of pages; pages generally contain various size and shape of connected components. Fortunately, we can also construct the area Voronoi diagram in a straightforward manner using the above algorithm.

Figure 1 demonstrates the steps of constructing the area Voronoi diagram. By applying labeling, border following, and sampling procedures to the document image in Fig. 1a, points on borders of connected components are obtained as in Fig. 1b. Using these points as the generators, the point Voronoi diagram and the area Voronoi diagram are constructed as shown in Figs. 1c and 1d, respectively.

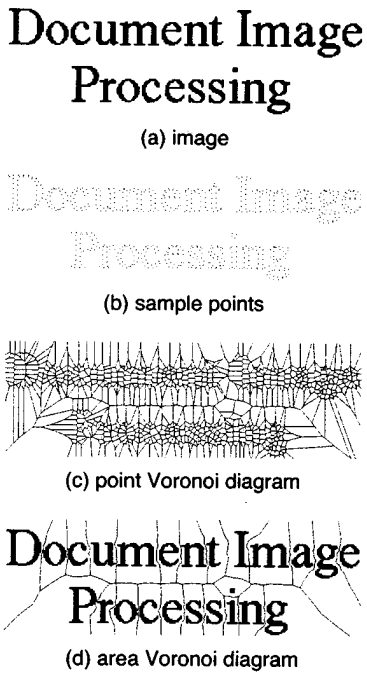


FIG. 1. Construction of the area Voronoi diagram.

3. SEGMENTATION AS SELECTION OF VORONOI EDGES

As shown in Fig. 1d, Voronoi edges lie between any adjacent connected components. In the case that a page has nonoverlapping layout, every document component is represented as a set of Voronoi regions which are adjacent with one another. In other words, Voronoi edges represent the structure of page layout as potential boundaries of document components. The process of page segmentation is, therefore, considered to be the selection of the Voronoi edges which represent the boundaries of document components. To this end, we need criteria for selecting appropriate Voronoi edges from the area Voronoi diagram.

3.1. Features for Selection

We attempt to select the appropriate Voronoi edges by deleting superfluous ones which are on the space between characters, words, text lines, etc. For this purpose, we employ two characteristic features: the minimum distance and the area ratio of connected components.

3.1.1. Minimum distance. In general, gaps between characters, words, and text lines are narrower than those between columns. Thus, the Voronoi edges in narrow spaces can be deleted as superfluous.

From this viewpoint, we employ the minimum distance defined as follows. Let $E = \{l_1, \ldots, l_m\}$ be a Voronoi edge between two connected components g_1 and g_2, where l_i is a line segment of E. Note that, for each line segment l_i, a pair of points p_i and q_i on the borders of g_1 and g_2, respectively, is associated as its generators. Then, the minimum distance $d(E)$ is defined by

$$d(E) = \min_{1 \le i \le m} d(p_i, q_i). \tag{6}$$

3.1.2. Area ratio. The minimum distance is effective for selecting Voronoi edges on thick white areas between columns. However, boundaries of document components do not necessarily have large values of d; document components that are visually distinguishable from text such as halftones and figures are sometimes laid out closer to text. Thus, the minimum distance alone is insufficient to extract all boundaries of document components.

Generally speaking, the area of figures and halftones is considerably larger than that of characters. On the other hand, characters have a little difference in the area. Thus, the area of a connected component can be used to overcome the above difficulty.

Let us consider again a Voronoi edge E which divides two connected components g_1 and g_2. If g_1 is close to g_2 with respect to the area, the Voronoi edge E should be deleted. To evaluate this point, we utilize the area ratio $a_r(E)$ defined by

$$a_r(E) = \frac{\max(a(g_1), a(g_2))}{\min(a(g_1), a(g_2))}, \tag{7}$$

where $a(g_i)$ represents the area (the number of pixels) of g_i $(i = 1, 2)$.

3.2. Evaluation of Voronoi Edges

Voronoi edges are evaluated using the above two features. If two connected components are characters in different columns, their area is not so different but they are distinguishable because of a wide gap between them (i.e., a column gap). If one is a character and the other is a halftone, they may be closely laid out but also distinguishable because of the difference of their area.

In order to represent the above relation between d and a_r, and to evaluate Voronoi edges in an efficient manner, we utilize the criteria

$$d(E)/T_{d1} < 1 \quad \text{or} \tag{8}$$

$$d(E)/T_{d2} + a_r(E)/T_a < 1, \tag{9}$$

where T_{d1}, T_{d2} $(T_{d1} < T_{d2})$, and T_a are thresholds. If a Voronoi edge satisfies at least one of these equations, it is deleted. Figure 2 illustrates the d–a_r space in which a Voronoi edge corresponds to a point; Voronoi edges in the shaded area are deleted. Equation (8) shows that Voronoi edges in too narrow spaces are deleted regardless of the area ratio. Equation (9) represents the above relation between d and a_r.

In order to use these criteria, it is necessary to determine the values of the thresholds.

The appropriate value of T_a is independent of resolution, but it is affected by the size of figures and halftones. We currently use

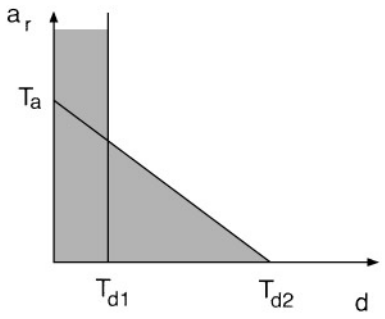


FIG. 2. Criteria for deleting superfluous Voronoi edges. A Voronoi edge corresponds to a point in the d–a_r space. The shaded portion indicates the area in which Voronoi edges are deleted as superfluous.

the fixed value of $T_a = 40$, which approximately corresponds to the largest area ratio between characters of the same font and size.

The appropriate values of T_{d1} and T_{d2} depend heavily on the resolution and the layout style. Thus, it is necessary to estimate the values adaptively from an input image. To this end, we utilize a frequency distribution of d. As shown in an example in Fig. 3, there are apparent two peaks P_1 and P_2 near the origin. It is considered that v_1 for P_1 corresponds to the intercharacter gap in body text regions, while v_2 for P_2 indicates the interline gap in body text regions.

The values v_1 and v_2 are found as follows. First, a raw frequency distribution $f_r(d)$ in which values of d are accumulated by the step of 1 [pixel] is smoothed using the local average,

$$f(d) = \frac{1}{2w+1} \sum_{i=d-w}^{d+w} f_r(i), \tag{10}$$

where $f(d)$ and w are a smoothed distribution and a window size, respectively (Fig. 3 is a distribution smoothed with $w = 2$). Then v_1 and v_2 are obtained as

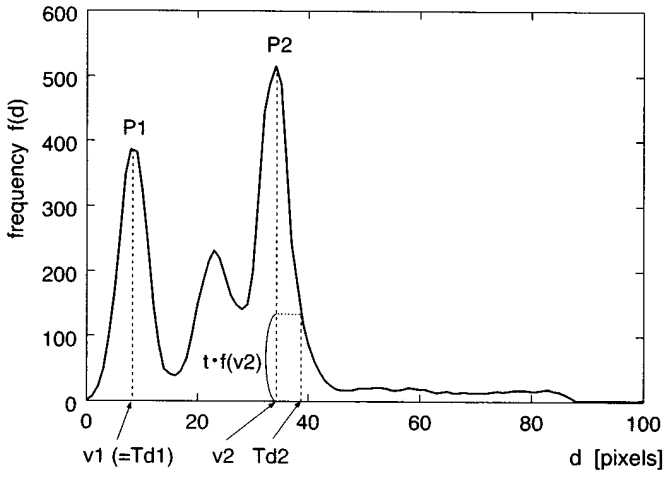


FIG. 3. A frequency distribution of the minimum distance d.

- $f(v_1)$ and $f(v_2)$ are the two largest peaks
- $v_1 < v_2$.

A peak is simply defined as $f(d)$, which fulfills $f(d-1) < f(d)$ and $f(d) > f(d+1)$.

Since most of the gaps between document components are estimated larger than the intercharacter gap in body text regions, we use v_1 as the value of T_{d1}. On the other hand, T_{d2} should be set to the value with which the Voronoi edges between text lines in a document component can be deleted. The value v_2 is inappropriate as T_{d2}, because some of such edges have larger values of d than v_2; we need to add a margin to v_2. Thus, T_{d2} is defined as

$$T_{d2} > v_2 \tag{11}$$

$$f(T_{d2}) = t \cdot f(v_2), \tag{12}$$

where t is a parameter for controlling the margin. Since $f(d)$ takes discrete values in actual cases, we apply linear interpolation to obtain T_{d2} which satisfies Eq. (12). If multiple values of T_{d2} are found, the smallest is used.

4. SEGMENTATION PROCEDURE

The segmentation procedure consists of three steps. The details of each step are described below using the example of Fig. 4a, which is tilted by 10° after scanning at 300 dpi from a magazine and includes nonrectangular document components (halftones and body text regions).

(1) Labeling. The first step is labeling to obtain 8-connected components from an input image. We utilize a fast labeling procedure based on border following [15], so that sample points on borders of connected components are simultaneously obtained while labeling. The sampling procedure preserves every Rth pixel until it terminates for each connected component, where R is a sampling parameter. In the labeling procedure, small connected components whose length of borders is less than or equal to N pixels are filtered out as noise.

(2) Generation of the area Voronoi diagram. The area Voronoi diagram is constructed using the sample points. We utilize the algorithm named the *plane sweep method* [12].

From the image in Fig. 4a, the area Voronoi diagram in Fig. 4b is generated. As shown in this figure, a large number of superfluous Voronoi edges, e.g., the edges lying between characters, words, and text lines, are included.

(3) Deletion of superfluous Voronoi edges. The final step is to delete the superfluous Voronoi edges to obtain boundaries of document components. After the smoothing of a frequency distribution, Voronoi edges satisfying Eqs. (8) and (9) are deleted. For the Voronoi edges in Fig. 4b, $T_{d1} = 8.0$ and $T_{d2} = 38.0$ are determined from Fig. 3 with $t = 0.34$, and used to select the Voronoi edges in Fig. 4c.

As shown in Fig. 4c, the deletion yields some Voronoi edges satisfying the following condition: at least one of their terminals

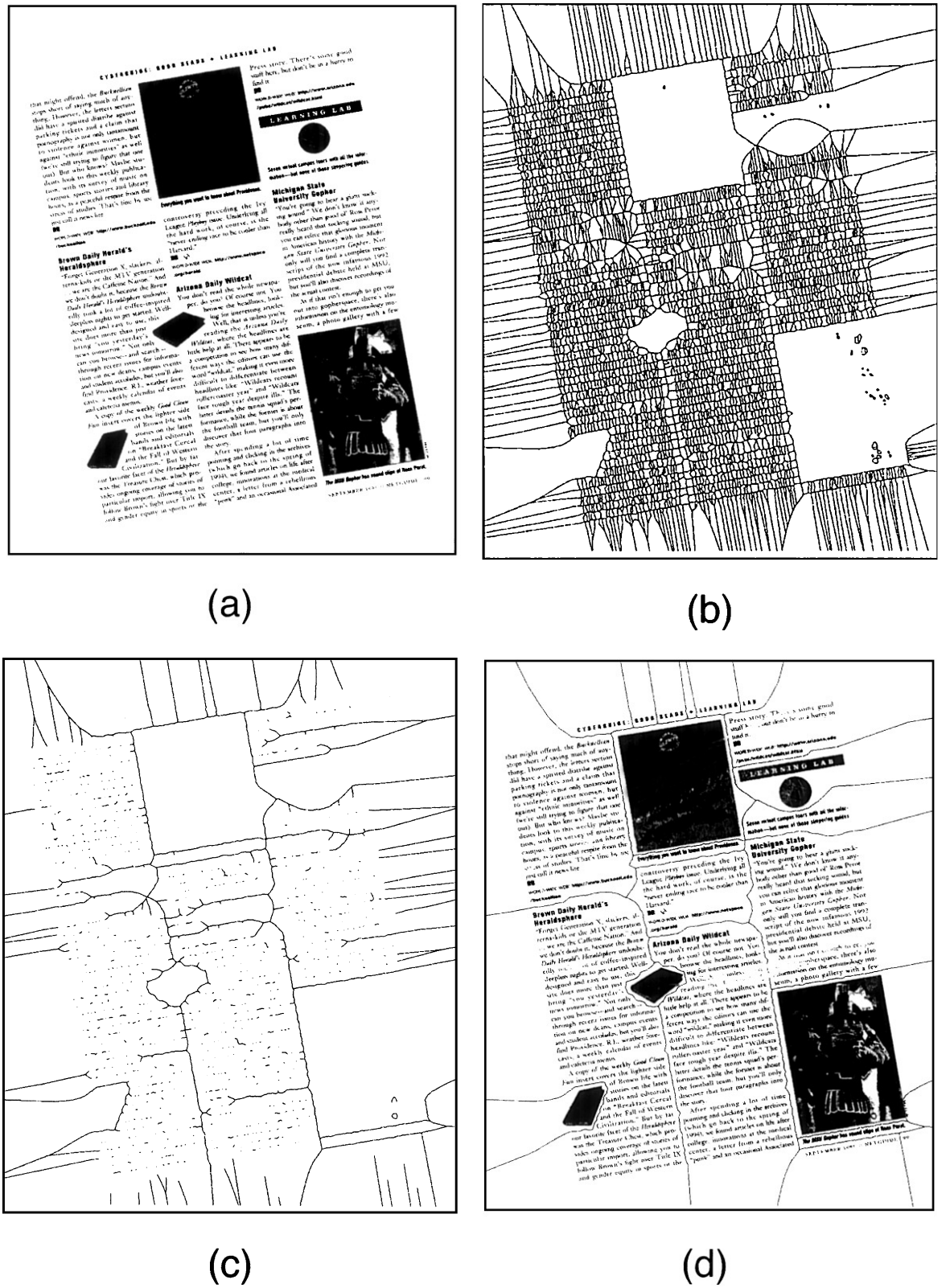


FIG. 4. Generation and deletion of Voronoi edges. From the input image (a) scanned with the resolution of 300 dpi, the area Voronoi diagram (b) is constructed. The Voronoi edges (c) are selected by the deletion using Eqs. (8) and (9) with $t = 0.34$. The final result (d) is obtained by deleting further Voronoi edges which violate the loop condition.

is neither shared by other Voronoi edges nor on the edge of the page image. Since boundaries of document components are loops consisting of either Voronoi edges or the edge of the page image, which we call *the loop condition*, such Voronoi edges are also deleted as superfluous. As a result, the Voronoi edges shown in Fig. 4d are obtained. For this example, all document components except the header and the lower left halftone are correctly segmented.

5. EXPERIMENTS

We implemented our method with C language and made experiments using a personal computer with a Pentium Pro 200 MHz CPU and 128 MB of real memory.

5.1. Data Sets

Table 1 shows the properties of three data sets used for the experiments. The data set nM-90 dpi was prepared by scanning 16 English pages with non-Manhattan layout at the resolution of 90 dpi. These pages had two-, three-, and four-column layout as listed in Table 1. The data set nM-300 dpi included the same pages but the resolution was 300 dpi. Since we applied no skew correction, the images in nM-90 dpi and nM-300 dpi had the skew of about $\pm 3^\circ$. In order to test the robustness against severe skews, these images were artificially tilted counterclockwise by angles of 10°, 30°, and 45°. Thus, the number of images in each data set was 64 (= 16 pages $\times$ 4 skew angles). On the occasion of rotating the images, we added white pixels on the borders of images to keep all original black pixels in the rotated images.

In order to test the applicability to pages with Manhattan layout, we also used images contained in the University of Washington database 1 (UW1): 98 images in which characters are not smeared by severe copy noises were selected from the images whose names begin with the letter A (scanned from the first generation copies). Since we did not introduce artificial skews for these images, the number of images is equal to the number of pages.

TABLE 1
Data Sets

	nM-90 dpi	nM-300 dpi	UW1
Resolution	90 dpi	300 dpi	300 dpi
Layout	non-Manhattan		Manhattan
Number of pages	16		98
1-column page	0 page		71 pages
2-column page	7 pages		24 pages
3-column page	7 pages		3 pages
4-column page	2 pages		0 page
Original skew	about $\pm 3^\circ$		$-2.12^\circ \sim 1.47^\circ$
Artificial skew	10°, 30°, 45°		no
Total number of images	64		98

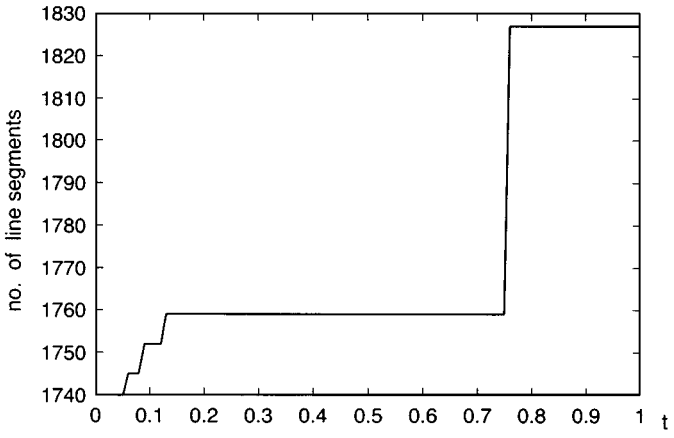


FIG. 5. Relation between t and the number of line segments in a final result for the image of Fig. 4a.

5.2. Parameters

First, the values of parameters were determined using half of the images in nM-90 dpi and nM-300 dpi, both of which included the same pages. Since the parameters N, R, and w depend on the resolution, different values were determined for each data set: $N = 4$, $R = 4$, $w = 0$ for nM-90 dpi, and $N = 13$, $R = 13$, $w = 2$ for nM-300 dpi. Also, the same value of $t = 0.34$ was used since it is independent of the resolution.

Note that the parameter t was not so sensitive for most of the images. Figure 5 illustrates the relation between t and the number of line segments in a segmentation result for Fig. 4a, which is one of the images used to determine the parameters. The result shown in Fig. 4d was equally obtained using $0.13 \leq t \leq 0.75$. Although Voronoi edges deleted by Eq. (9) varied according to t, the variation was removed by deleting Voronoi edges with the loop condition.

We did not take the images in UW1 into account for the determination of the parameter values. Since their resolution was also 300 dpi, we applied the same values for nM-300 dpi.

5.3. Criteria of Evaluation

In order to evaluate the results of segmentation, we used the criteria described below.

Document components can be classified into two types: text regions and nontext regions. Text regions can be subdivided into body text regions and the remainder which we call auxiliary text regions. The former consist of text in chapters and sections as well as footnotes, while the latter include titles of documents, chapters and sections, authors, captions, headers, and footers. A body text region indicates a block including a maximal number of text lines which have a consecutive reading order. Nontext regions consist of figures, tables, and halftones. For each type of document components, we consider that the results of segmentation are correct if they satisfy the following conditions.

Body text region. Segmentation is correct in this region if all of its paragraphs are not fragmented and not merged with different document components. We consider that a body region is correct even if it is fragmented between consecutive paragraphs. A body text region merged with a chapter or a section title is incorrect.

Auxiliary text region. Segmentation is correct in this region if all text lines are not fragmented and not merged with different document components. In headers and footers, multiple items such as a page number and a publishing date are often laid out apart. Their regions are regarded as correct even if these items are separated. Regions of chapter and section titles are correct even if their numbers are separated from text lines of titles.

Nontext region. Segmentation is correct in this region if it is not fragmented and not merged with different document components.

We did not take account of ruled lines, since solid ruled lines were often fragmented in the 90 dpi images due to the low resolution, and most dotted ruled lines in 300 dpi images were filtered out as noises.

Errors of segmentation can be divided into two categories: *fragmentation* and *over-merging*. The former is the case where a region of a single document component is erroneously divided into two or more regions, while the latter is the case where the regions of multiple document components are erroneously merged into one region. From the viewpoint of succeeding steps in document image understanding such as page classification, character recognition, and logical labeling, over-merged regions generally cause more serious problems than fragmented regions. Thus, it is a requirement for page segmentation methods to keep the frequency of over-merging low.

The results were evaluated for three categories of document components, i.e., body text regions, auxiliary text regions, and nontext regions, by using the rates of fragmentation and over-merging defined as follows. Let N, N_f, and N_o be the number of document components, the number of fragmented, and the number of over-merged document components in a category, respectively. The rates of fragmentation and over-merging are defined as N_f/N and N_o/N. In the case that both of fragmentation and over-merging occurred in a document component, it was counted as over-merging. The number of document components included in each data set is listed in Table 2. For nM-90 dpi and nM-300 dpi, document components were counted in 16 pages.

TABLE 2
The Number of Document Components

	nM-90 dpi	nM-300 dpi	UW1
Body text	59		220
Auxiliary text	58		371
Nontext	26		47

TABLE 3
Results of the Experiments

	Body text	Auxiliary text	Nontext
nM-90 dpi	0.8%/7.2%	34.9%/6.0%	58.7%/12.5%
nM-300 dpi	2.1%/0.4%	25.9%/0.4%	65.4%/1.9%
UW1	16.8%/5.5%	10.5%/2.7%	97.9%/2.1%

Note: Fragmentation rate/over-merging rate.

5.4. Results

Examples of the results for each data set are shown in Figs. 6a to 6c, respectively. Although the header in Fig. 6a and the figure in Fig. 6b were fragmented, other document components were correctly extracted.

The results for each data set are summarized in Table 3, where the rates of fragmentation and over-merging are shown at the left and the right of a slash. A noticeable point is that over-merging occurred less frequently than fragmentation except in the case of body text regions for nM-90 dpi.

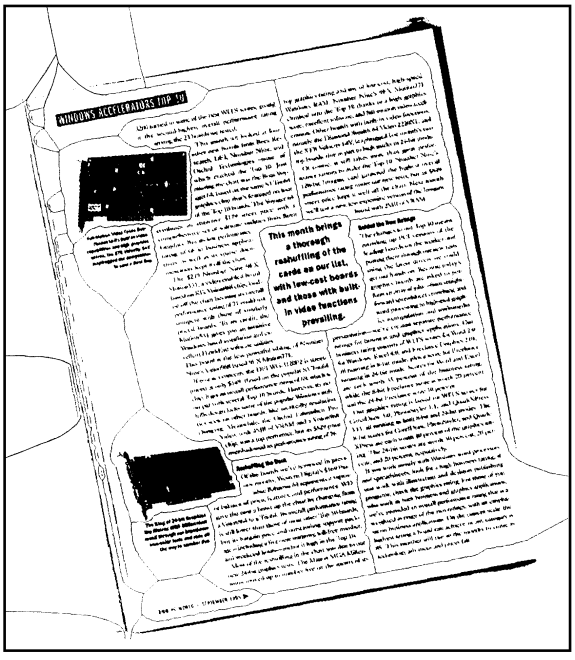
Most of the fragmentation of text regions occurred in auxiliary text regions. An example of the fragmentation is shown in Fig. 7a. Titles are often written in larger fonts and thus the gaps between words are wider than those between columns. As a result, the value of T_{d2}, which was suitable for body text regions, was too small to extract titles correctly. Headers and footers with wider interword gaps were also fragmented because of the same reason.

The fragmentation rate of body text regions for UW1 was higher than those for nM-90 dpi and nM-300 dpi. Most of the fragmentation was on equations and lists of references, both of which were not contained in nM-90 dpi and nM-300 dpi. Figure 7b shows an example. The fragmentation occurred in the case in which there exist wider interline gaps above and below of a text-line and wider intercharacter gaps caused by commas and periods.

For nontext regions, small parts surrounded by white space were fragmented in the case in which they were not merged with their body by Eq. (8), since the area ratio between a part and a body was too large to satisfy Eq. (9). Such small parts were blobs in halftones, words and phrases in figures, and items and columns in tables. An example is shown in Fig. 7c.

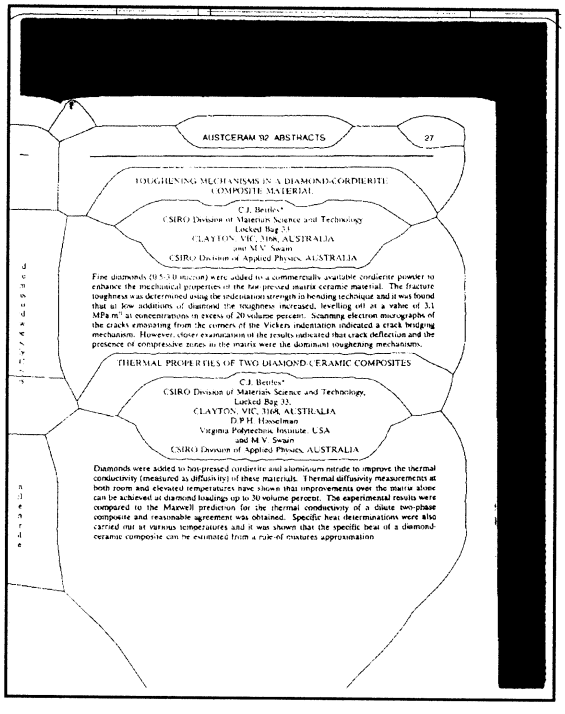
In comparison with fragmentation, over-merging occurred less frequently. The number of pages (and images) which included over-merged regions was 7 (20) in nM-90 dpi, 1 (2) in nM-300 dpi, and 11 (11) in UW1. Although over-merged regions were found on various pages in nM-90 dpi, most of them were eliminated by raising the resolution. Figure 7d illustrates an example of over-merged pages in nM-90 dpi; body text regions were erroneously merged due to the dots in the ruled line. However, the correct results were obtained for the same page scanned at 300 dpi.

Over-merging in UW1 occurred more frequently than in nM-300 dpi. However, there was a clear tendency for these errors;



(a)

(b)



(c)

FIG. 6. Segmentation results for images in (a) nM-90 dpi, (b) nM-300 dpi, and (c) UW1. (a) and (b) are tilted by 10°.

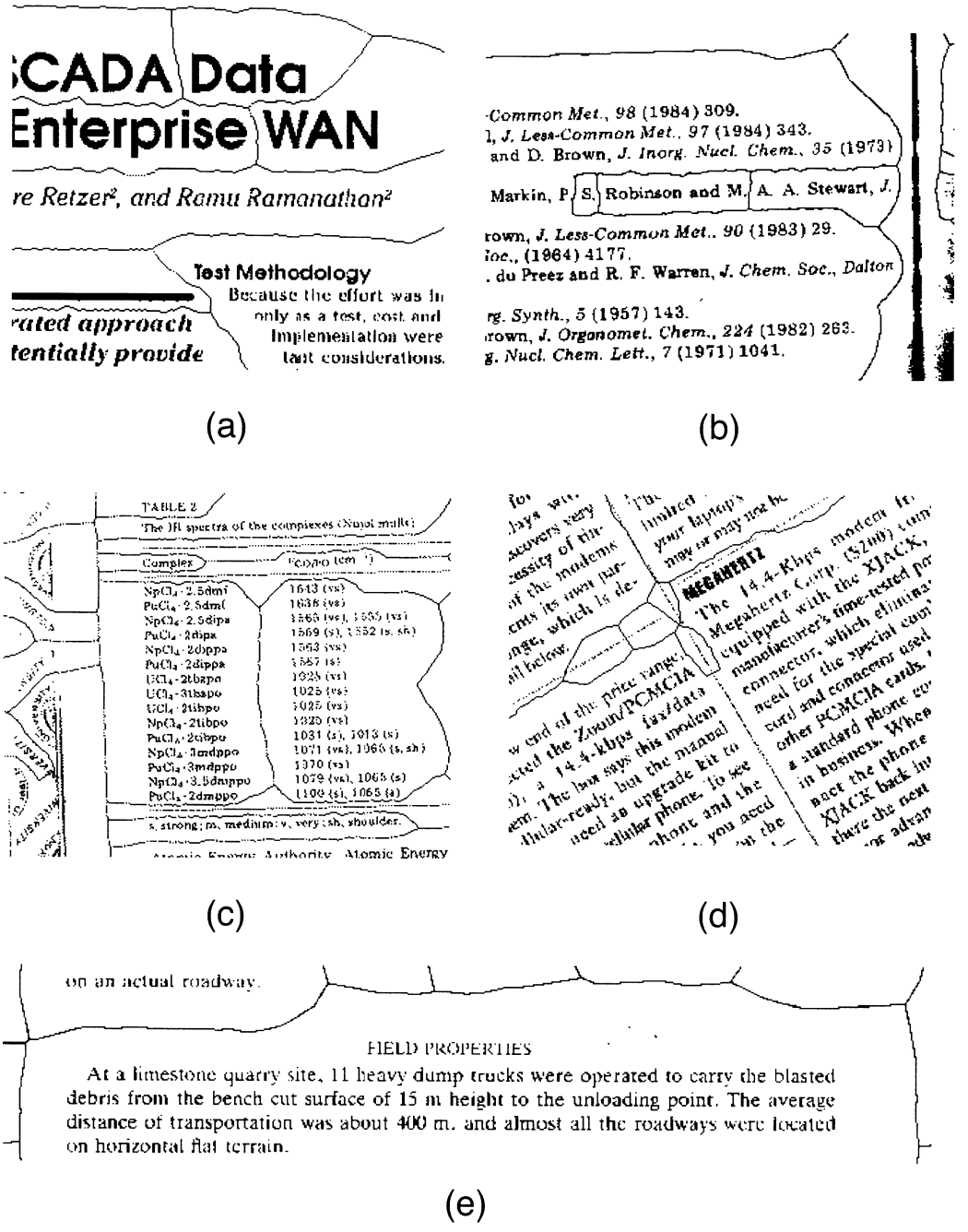


FIG. 7. Examples of segmentation errors. (a) The title is fragmented due to wide interword gaps. (b) Words are fragmented from the body text region (a list of references). (c) Columns of a table are isolated. (d) Body text regions are merged due to the dotted ruled line. (e) The section title is merged with the body text region. Parts (a) and (d) are in the results for nM-90 dpi, and the rest are in the results for UW1.

over-merging between a body text region and a footer was in four pages, and that between a body text region and a section title was in four different pages. Figure 7e shows an example. Section titles and footers were laid out close to body text regions in some pages in UW1. The rest were over-merging between a caption and a figure (one page), and between body text regions due to a stamped mark of a library (two pages).

Broadly speaking, although the results for nM-90 dpi were error-prone due to the low resolution, most of the errors were removed in nM-300 dpi. Since the parameters were determined using images in nM-90 dpi and nM-300 dpi, some errors in UW1 could be corrected by adjusting the parameters. However, most of them were caused by the lack of knowledge about the document components. For instance, errors in text regions could

TABLE 4
Average Properties of Images in the Data Sets

	nM-90 dpi	nM-300 dpi	UW1
Image size	1053×1149	3114×3554	2592×3300
Connected components	4.91×10^3	1.36×10^4	3.99×10^3
Sample points	2.49×10^4	5.42×10^4	2.60×10^4
Line segments in			
Point Voronoi	6.15×10^4	9.77×10^4	6.98×10^4
Area Voronoi	2.99×10^4	3.88×10^4	2.85×10^4
Result	1.88×10^3	1.72×10^3	2.44×10^3

be corrected using the knowledge about the layout of text lines (e.g., centering and indent).

It is also worth noting that the results were almost independent of the skew angles introduced in nM-90 dpi and nM-300 dpi. The number of document components differently segmented depending on skew angles was less than or equal to two for nM-300 dpi and three for nM-90 dpi. Figures 8a–8c illustrate the results for images in nM-300 dpi, which is the same page as in Fig. 4a but with different skews. For these images, most of the segmented regions are identical to those in Fig. 4d.

5.5. Processing Time

Finally, let us mention the efficiency of our method. Table 4 lists averages of various data for each data set. For nM-300 dpi, for example, 50,000 sample points were extracted from the average image including 10,000 connected components. A result of segmentation consisted of about 1700 line segments which were 4% of the line segments contained in the area Voronoi diagram.

For the images of the above properties, our method needed computation time shown in Table 5. In this table, "labeling," "Voronoi," and "deletion" correspond to the steps described in Section 4, while "others" indicates memory allocation, initialization, and file I/O.

The step of labeling dominated the computation time for all data sets. The step of constructing the area Voronoi diagram took the second longest time but it was about half or less amount of the time for labeling. As the resolution of images increased, the rate of the time consumed by the construction of the area Voronoi diagram decreased.

TABLE 5
Computation Time (in Seconds)

	nM-90 dpi	nM-300 dpi	UW1
Labeling	1.75 (59.7%)	5.17 (73.6%)	4.02 (74.9%)
Voronoi	1.07 (36.5%)	1.73 (24.6%)	1.23 (22.9%)
Deletion	0.08 (2.8%)	0.10 (1.4%)	0.08 (1.5%)
Others	0.03 (1.0%)	0.03 (0.4%)	0.04 (0.7%)
Total	2.93	7.03	5.37

Note. The ratio of time to the total is indicated in parentheses.

Figure 9 shows the relation between the number of sample points and the time for the construction of the area Voronoi diagram. For the images in nM-90 dpi and UW1, the time is almost proportional to the number of sample points as shown by the broken line in Fig. 9. Although there was no such regularity for nM-300 dpi, the time for nM-300 dpi was also on the broken line when we performed the experiments with different values of the parameter R; we have not experienced the case in which the time apparently exceeded the line in Fig. 9.

From these results, it can be said that our method is applicable to images that can be labeled in reasonable time.

6. DISCUSSION

From the results of the experiments, we have confirmed that our method is capable of dealing with document components of arbitrary shape regardless of layout and skew angles. The automated setting of thresholds enables us to extract body text regions with high accuracy.

Our method relies on neither the linearity of text lines nor the uniqueness of skew angles of text lines. This property enables us to extract text regions of arbitrary shape and skew angles, which could be contained in pages with non-Manhattan layout. Figure 10 exemplifies the ability of our method. The image in Fig. 10 was obtained by scanning at 300 dpi and processed using the same parameter values for nM-300 dpi.

On the other hand, our method has some limitations to be improved. One limitation is that it often fragments figures, tables, and halftones as well as titles with larger fonts, headers and footers with wider interword gaps. In addition, we assume that

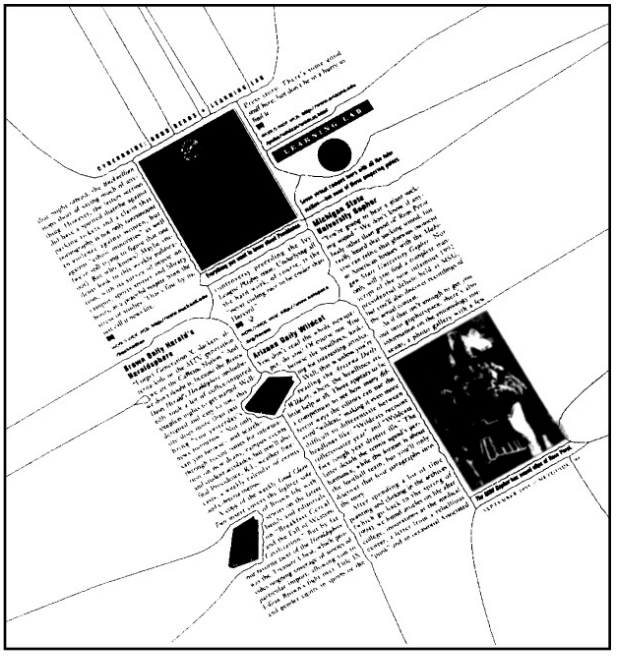
- body text regions are dominant on a page; i.e., most Voronoi edges are in body text regions,
- body text regions are uniform; i.e., the same intercharacter and interline gaps are used

to determine the values of thresholds from the frequency distribution. These assumptions pose another important limitation. For the segmentation of pages in which body text regions are not dominant, as well as various gaps are used, it is necessary to use a different way of the determination of thresholds. However, the area Voronoi diagram is still considered useful for such pages.

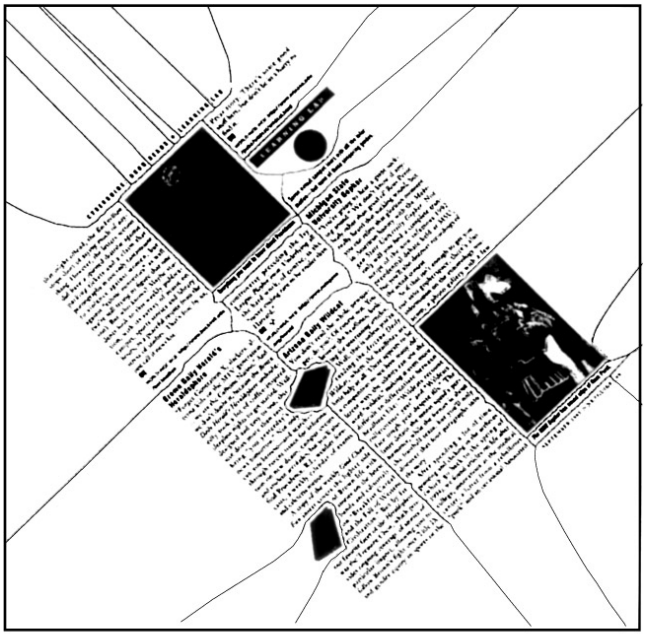
From the analysis of processing time, it is apparent that the use of the area Voronoi diagram causes no drawback in efficiency. Since the dominant step is labeling, our method is as efficient as other methods based on the analysis of connected components.

Another advantage is that the result of segmentation is represented by a small number of line segments. This requires much less memory than the representation by labeled images. For pages with the rectangular layout, rectangles play an important role in both representing and processing images. We consider that the line segments could play a similar role for pages with non-Manhattan layout.

(a)



(b)



(c)

FIG. 8. Segmentation results for the same page in Fig. 4a but with different skew angles. (a) The original image, (b) the image tilted by 30°, (c) the image tilted by 45°. The resolution is 300 dpi.

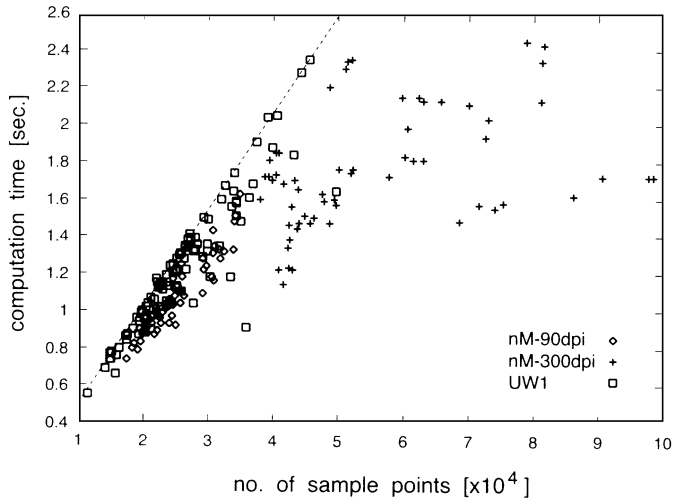


FIG. 9. Relation between the number of sample points and the computation time for constructing the area Voronoi diagram.

7. RELATED WORK

In this section, we compare our method with other methods which are also applicable for pages with non-Manhattan layout and a skew. From the viewpoint of objects to be analyzed, methods of page segmentation are classified into foreground analysis and background analysis. We describe the comparison with each of them below.

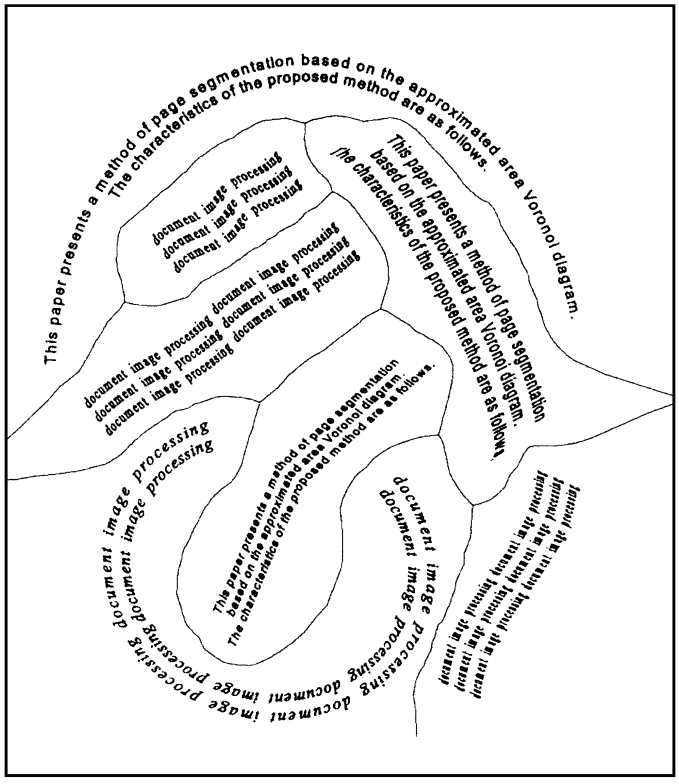


FIG. 10. A segmentation result for text regions in complex shapes. The resolution and the size of the image are 300 dpi and 2400 × 2790, respectively.

7.1. Foreground Analysis Methods

Foreground analysis is a category of methods which collects black pixels to form document components. In order to analyze pages with non-Manhattan layout and a skew, we need primitives that can be extracted from page images regardless of their layout or skew. In addition to pixels themselves, connected components meets this requirement.

In the case that connected components are used as primitives, the process of page segmentation consists of merging them into document components. The various notions of neighbors are exploited to define the criteria for merging.

Fletcher and Kasturi have utilized four nearest neighbors within a collinear group of connected components [5]. The nearest neighbors in the group are merged into words and phrases based on the distance. O'Gorman has focused on k-nearest neighbors to form blocks as well as text-lines from connected components [6]. He has used the parallelness and some other features in addition to the distance and orientation. The feature shared by these methods is that they rely on the use of restricted neighbors, i.e., four nearest neighbors within a collinear group and k-nearest neighbors with a predetermined value of k, as well as the assumption of linearity of text lines. In addition, connected components are regarded as points so that it is difficult to deal directly with connected components of a variety of shape and size.

These difficulties can be resolved by using the area Voronoi diagram [13]. In the area Voronoi diagram, neighbors are defined without the use of parameters and assumptions as the relation between connected components sharing a Voronoi edge. Another advantage is that it reflects the shape and size of connected components. Our method inherits these advantages which help us to estimate the intercharacter and interline gaps.

As stated in the previous section, our method is limited in the extraction of text regions with large font sizes. Some researchers have proposed methods capable of coping with this problem by using iterative processing. For example, Hönes and Lichter have proposed a relaxation-based method [7]. Although such methods consume a large amount of time in the iteration, they give better results. Since the step of deleting Voronoi edges requires considerably small amount of time in our method, we could consider the introduction of iterative processing into our method.

7.2. Background Analysis Methods

Background analysis is a category of methods which obtain document components by extracting their boundaries from page images. In recent years, this approach has attracted much attention, since many methods for non-Manhattan layout and tilted pages have been proposed [8–10].

The key issues of background analysis are (1) to define advantageous primitives for representing background so as to deal with non-Manhattan layout and tilted pages, and (2) to develop a filtering strategy to extract correct boundaries.

Pavlidis and Zhou have used right rectangles of a predetermined short height as primitives and filtered them by their width to extract column gaps [8]. The method by Antonacopoulos and Ritchings employ a similar approach [9]. Normand and Viard-Gaudin have discussed the advantage of the use of regular octagons as primitives to attain two-dimensional isotropy [10]. Kise *et al*. have proposed a different approach which uses connected white pixels obtained by thinning the background [16].

It can be considered that the elaboration of both primitives and filtering methods is to find pairs of connected components on the boundaries of document components. Since Voronoi edges generated by our method are on the background in ordinary cases, they can be approximately regarded as a representation of background. Voronoi edges indeed bear resemblance to the connected white pixels produced by thinning. However, Voronoi edges consist of not pixels but line segments, so that the amount of data to represent the boundaries is much less than those of the connected white pixels. In addition, the construction of the area Voronoi diagram is faster than the operation of thinning, because the area Voronoi diagram is generated based on *sampled* pixels on borders of connected components.

8. CONCLUSION

We have presented a method of page segmentation based on the approximated area Voronoi diagram. The characteristics of our method are as follows:

- The approximated area Voronoi diagram enables us to obtain the candidates of boundaries of document components regardless of layout and a skew.
- The intercharacter and interline gaps can be estimated from Voronoi edges without the use of domain specific parameters.
- Boundaries of document components are obtained by deleting superfluous Voronoi edges.

We use the simple criteria described by two features: the minimum distance and the area ratio. The thresholds are adaptively determined based on the estimated gaps.

From the experimental results for images with Manhattan and non-Manhattan layout as well as various skews, we have confirmed that our method is effective in extracting body text regions but has some limitations in extracting text regions with large fonts and wide interword gaps, as well as figures, tables, and halftones. Experimental results also indicate that our method is as efficient as other methods based on the analysis of connected components.

Future work is to overcome the limitations by, for example, using iterative processing such as split-and-merge with more fruitful features for selecting Voronoi edges.

REFERENCES

1. L. O' Gorman and R. Kasturi, *Document Image Analysis*, IEEE Computer Society Press, Los Alamitos, 1995.
2. R. M. Haralick, Document image understanding: Geometric and logical layout, in *Proceedings, Computer Vision and Pattern Recognition '94, Washington, 1994*, pp. 385–390.
3. H. S. Baird, S. E. Jones, and S. J. Fortune, Image segmentation by shape-directed covers, in *Proceedings, 10th Int'l Conf. on Pattern Recognition, Atlantic City, 1990*, pp. 820–825.
4. A. K. Jain and S. Bhattacharjee, Text segmentation using Gabor filters for automatic document processing, *Machine Vision Appl*. **5**, 1992, 169–184.
5. L. A. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images, *IEEE Trans. Pattern Anal. Machine Intell*. **10**, 1988, 910–918.
6. L. O'Gorman, The document spectrum for page layout analysis, *IEEE Trans. Pattern Anal. Machine Intell*. **15**, 1993, 1162–1173.
7. F. Hönes and J. Lichter, Layout extraction of mixed mode documents, *Machine Vision Appl*. **7**, 1994, 237–246.
8. T. Pavlidis and J. Zhou, Page segmentation and classification, *CVGIP: Graphical Models Image Process*. **54**, 1992, 484–494.
9. A. Antonacopoulos and R. T. Ritchings, Flexible page segmentation using the background, in *Proceedings, 12th Int'l Conf. on Pattern Recognition, Jerusalem, 1994*, pp. 339–344.
10. N. Normand and C. Viard-Gaudin, A background based adaptive page segmentation algorithm, in *Proceedings, 3rd Int'l Conf. on Document Analysis and Recognition, Montreal, 1995*, pp. 138–141.
11. K. Sugihara, Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams, *CVGIP: Graphical Models Image Process*. **55**, 1993, 522–531.
12. A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester, 1992.
13. M. Burge and G. Monagan, *Using the Voronoi Tessellation for Grouping Words and Multipart Symbols in Documents*, Technical Report No. 131–95, Johannes Kepler University, 1995.
14. D. J. Ittner and H. S. Baird, Language-free layout analysis, in *Proceedings, 2nd Int'l Conf. on Document Analysis and Recognition, Tsukuba, 1993*, pp. 336–340.
15. Y. Ishiyama, F. Kubo, H. Takahashi, and F. Tomita, Labeling board based on boundary tracking, *Trans. IEICE Jpn. (D-II)* **J78-D-II**, 1995, 69–75. [In Japanese]
16. K. Kise, O. Yanagida, and S. Takamatsu, Page segmentation based on thinning of background, in *Proceedings, 13th Int'l Conf. on Pattern Recognition, Vienna, 1996*, III, pp. 788–792.