Document Page Decomposition by the Bounding-Box Projection Technique

Jaekyu Ha & Robert M. Haralick

Dept. of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

Ihsin T. Phillips

Dept. of Computer Science
Seattle University
Seattle, WA 98122

Abstract

This paper describes a method for extracting words, textlines and text blocks by analyzing the spatial configuration of bounding boxes of connected components on a given document image. The basic idea is that connected components of black pixels can be used as computational units in document image analysis. In this paper, the problem of extracting words, textlines and text blocks is viewed as a clustering problem in the 2-dimensional discrete domain. Our main strategy is that profiling analysis is utilized to measure horizontal or vertical gaps of (groups of) components during the process of image segmentation. For this purpose, we compute the smallest rectangular box, called the bounding box, which circumscribes a connected component. Those boxes are projected horizontally and/or vertically, and local and global projection profiles are analyzed for word, textline and text-block segmentation. In the last step of segmentation, the document decomposition hierarchy is produced from these segmented objects.

1 Introduction

The printing process is the transformation of the logical hierarchy of a given document into the physical hierarchy. The process must follow the set of rules or protocols which prescribe the physical document layout requirements at the time of production. The requirements may include the font type, size and style for each symbol, the column format (including the number of columns and column width), the header, the footer and margin dimensions. Also, there are also intrinsic spacing protocols for the symbols and words as well as for textlines, text blocks and text columns. In almost all cases, spacings between symbols are much smaller than spacings between words within the same printed document. Similarly, spacings between textlines are smaller than spacings between text-blocks and/or text-columns. This tendency has been used as prior knowledge in most OCR and document image analysis algorithms.

This paper describes a technique for extracting words, textlines and text blocks by analyzing the spatial configuration of the bounding boxes of symbols in a given document page. In particular, the 'bounding boxes' of the connected-components of black pixels are used as the basis of such extractions.

The remainder of this paper is organized as follows: In Section 2, we describe the decomposition algorithm in a step-by-step manner. Section 3 discusses experiments on the UW English Document Image Database I. The concluding remarks are given in Section 4.

2 Text Zone Delineation

Now we describe the page decomposition algorithm in a step-by-step manner. We assume that the input document image has been correctly deskewed.

2.1 Bounding Boxes of Connected Components

Let I denote the input binary image. A connected component analysis algorithm [2] is applied to the foreground region of I to produce the set of connected components. Then, for each connected component, its associated bounding box – the smallest rectangular box which circumscribes the component – is calculated. A bounding box can be represented by giving the coordinates of the upper left and the lower right corners of the box.

Figure 1(a) shows a segment of an English document image (taken from the UW English Document Image Database I, page id "L006SYN.TIF") and Figure 1(b) shows the bounding boxes produced in this step. Note that, the number of bounding boxes are

0-8186-7128-9/95 $4.00 © 1995 IEEE

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

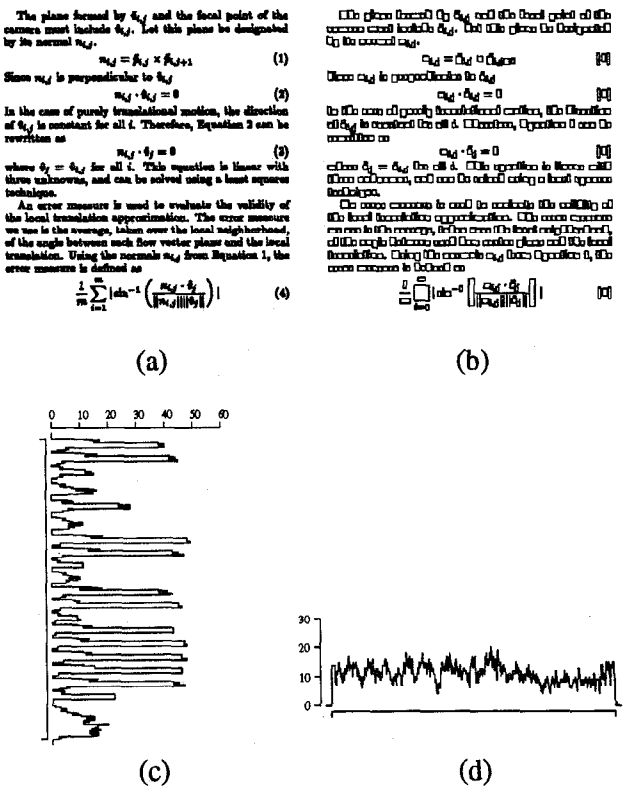

Figure 1: (a) an English document, (b) bounding boxes of connected components of black pixels, (c) horizontal projection profile, (d) vertical projection profile.

always larger than the number of symbols since multiple bounding boxes are produced for multi-component symbols. Our page decomposition scheme analyzes the spatial configuration of those bounding boxes of connected components to extract textlines, words, and paragraphs.

2.2 Projections of Bounding Boxes

Analysis of the spatial configuration of bounding boxes can be done by projecting those bounding boxes onto a straight line. Since paper documents are usually written in the horizontal or vertical direction, projections of bounding boxes onto the vertical and horizontal lines are of particular interest. While projecting bounding boxes onto the horizontal or vertical line, they will accumulate onto that line, which results in the projection profile. A projection profile is a frequency distribution of the projected bounding boxes on the projection line. The bounding box projection profiles provide important information about the number of bounding boxes aligned along the projection direction. Figure 1(c) and 1(d) shows the horizontal and vertical projection profiles of the bounding boxes in Figure 1(b).

2.3 Extraction of Textlines

In this step, the algorithm first determines the textline direction of the page by analyzing both horizontal and vertical projection profiles. Once the textline direction of the page is determined, the algorithm partitions the page bounding box into textline bounding boxes.

From Figure 1(c) and 1(d), it is easy to see that textlines are horizontally oriented: On the horizontal projection profile, there are distinct high peaks and deep valleys at somewhat regular intervals, whereas on the vertical projection profile, there is no such distinction. Since the bounding boxes are represented by the coordinates of two opposite end points, textlines are easily extracted and Figure 1(f) shows the result.

2.4 Extraction of Words

In this step, the algorithm groups the bounding boxes on each textline (produced from the last step) into bounding boxes of words.

The algorithm first computes the projection profiles within each of the textline bounding boxes. Figure 1(e) shows projection profiles within textlines. Next, the algorithm considers each of the projection profiles as a one-dimensional *gray-scale image*, and thresholds each of the images with threshold value 1 to produce a binary image. Note that, during the binarization, a symbol (or a broken symbol) with multiple bounding boxes may be merged into one, as well as, those adjacent symbols within the same textline whose bounding boxes are overlapping with each other. But this will not cause any problem in the result of our word extraction process, since our algorithm extracts words by merging bounding boxes based on the lateral proximity of neighboring boxes.

After such binarization, the algorithm performs a morphological closing operation on each of the binarized textline projection profiles with structuring element of appropriate size. The length of the structuring element is determined by analyzing the distribution of the run-lengths of 0's on the binarized textline projection profile. In general, such a run-length distribution is bi-modal. One mode corresponds to the inter-character spacings within words, and the other to the inter-word spacings. A threshold value can be chosen in the valley between the two dominant histogram modes. The two elegant techniques suggested

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

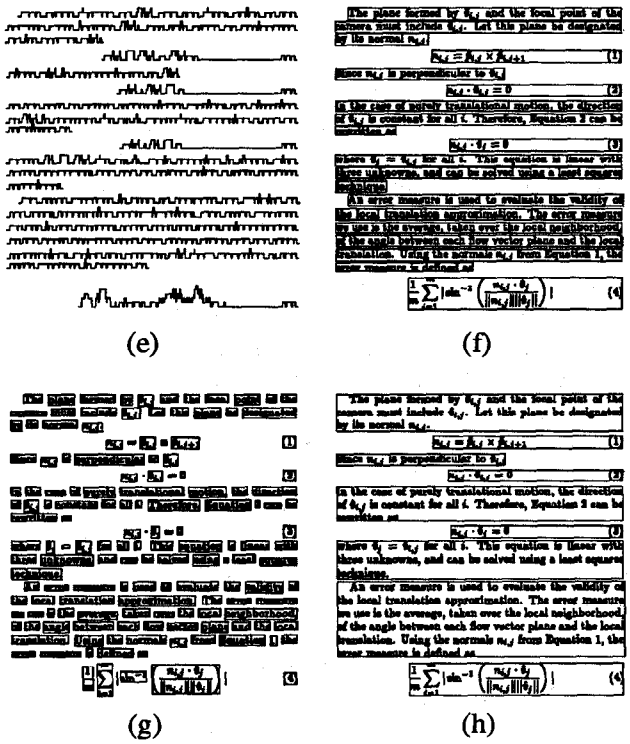

Figure 1 (continued): (e) vertical projection profiles, (f) textline bounding boxes, (g) word bounding boxes, (h) text-block bounding boxes.

by Otsu (1979) and Kittler & Illingworth (1986) can be applied to find an optimum threshold value. If such a threshold value is used as the length of the structuring element, the morphological closing operation closes spaces between symbols, but not the spaces between words. Figure 1(g) shows the results of this step.

2.5 Extraction of Paragraphs

In this step, the algorithm groups the bounding boxes of the extracted textlines into bounding boxes of text-blocks.

Though a paragraph is merely a unit in the logical hierarchy, its physical appearance is noticeable in a document image. There are four basic types of text line layout (justification) that are in common use: centered, flush left, flush right, and justified. Whatever justification is used in the preparation of a document, paragraphs are usually made either by changing the justification of the current text line or by putting more space between two text lines, one of which is from the previous paragraph and the other of which from the current one. In the former case, one might usually indent the first line of a paragraph of text.

Extraction of paragraphs should be primarily based on the above two basic paragraph-breaking methods. When a significant change in textline heights or in inter-textline spacings occurs, we might say that a new paragraph begins. The distributions of textline heights and inter-textline spacings together with the horizontal projection profile give us a clue for paragraph breaking occurrences. If there is a change in the textline justification, we might say that a new (type of) text block begins. Figure 1(h) shows text block bounding boxes for documents in Figure 1(a).

3 Experiments

We tested our word segmentation algorithm on 168 synthetic images in the UW English Document Image Database I, because the word-box groundtruth data can be easily generated from groundtruth files in the database.

3.1 Word Box Groundtruth Data

The UW English Document Image Database I is the first database to be constructed in a series of comprehensive CD-ROM document databases [Phillips et. al., 1993]. The database includes more than 1,000 distinct scientific journal article pages, as well as reproductions and degradations (through real-life copying) of these distinct article pages. The database is intended to serve as a comprehensive data set and to be used by researchers and developers in the area of OCR and document image analysis.

All 168 synthetic images are provided in the database. For each synthetic image, there is provided a so-called character groundtruth file in which every symbol is represented by its bounding box, font type, size and code. Another type of groundtruth data provided in the database is the zone-based groundtruth. It consists of the symbol strings which are contained in the text zone. This includes the standard ASCII characters as well as escape sequences which represent special symbols. The lines in the groundtruth data are broken at the same position of the string where the physical line is broken on the page.

For a particular synthetic image in the database, we can generate the word box groundtruth data in the following way. First, we locate all the text zones in the image from the corresponding zone attribute and zone box files. With such information, we can determine textlines from the corresponding character groundtruth file. The decision is made based on the simple rule: $\Delta x < -T_x$ and $\Delta y > T_y$ where

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE

	mismatched all	mismatched text	total
word seg.	227 (0.37%)	63	60936
groundtruth	318 (0.52%)	73	61027

Table 1: word segmentation result

Δx and Δy are difference of x- and y-coordinates of two consecutive bounding boxes listed in the character groundtruth file, and T_x and T_y are some positive integer values. Then we find correspondence of the symbols in the zone-based groundtruth file and the symbols associated with bounding boxes in the character groundtruth file. By grouping the bounding boxes which form a word, we can generate the word box groundtruth data for synthetic images in the database.

3.2 Evaluation of the Word Segmentation Algorithm

The output of the word segmentation algorithm is a set of word bounding boxes. To evaluate the performance, we need to compare the word box groundtruth data and the word bounding boxes produced by the word segmentation algorithm. Let $\mathcal{G} = \{G_1, G_2, \cdots, G_N\}$ denote the total of N groundtruth word bounding boxes and let $\mathcal{D} = \{D_1, D_2, \cdots, D_M\}$ denote the total of M detected word bounding boxes which are produced by the word segmentation algorithm. For our purpose, we simply compute $\mathcal{G} - \mathcal{D}$ and $\mathcal{D} - \mathcal{G}$.

Technical/scientific document images usually contain math expressions embedded in text lines. Our word segmentation algorithm will produce word bounding boxes based on the amount of space between consecutive symbols. Therefore, if a word bounding box produced by the algorithm contains a pure text word, it is the word bounding box in the true sense. However, if a word bounding box produced by the algorithm contains an inline math expression, we still call it a word bounding box because we are not concerned with the content of the box until symbol recognition is attempted.

Table 1 shows the number of elements in $\mathcal{G} - \mathcal{D}$ and $\mathcal{D} - \mathcal{G}$ versus the number of elements in $\mathcal{G}$ and $\mathcal{D}$. In the tables, "mismatched all" represents the number of mismatched word bounding boxes, and "mismatched text" represents the number of mismatched bounding boxes of pure text words. In the last column, the total number of bounding boxes are recorded.

4 Discussions

In this paper, we describe a new document page decomposition technique. The entire decomposition process is based on the analysis of the spatial configuration of bounding boxes of connected components. In our approach, connected components become the lowest level of the document hierarchy.

Correction of page skew is not of concern in this study. However, it is worth mentioning that performance of our decomposition method strongly depends on how much a document image is skewed. In fact, we may not be able to correctly extract text lines for about 0.5° skew of a letter-sized, single column, single spaced text document. Therefore, deskewing of the document image must precede the decomposition.

Our decomposition method has its own computational aspect: Once bounding boxes are obtained, the method does not refer to actual images. During the decomposition process, the method manipulates only bounding boxes. Hence, for a letter-sized document image at 300 dpi resolution, the number of computational units are reduced from $8.4 \times 10^6 (= 2550 \times 3300)$ pixels to at most a few thousands of bounding boxes. Our decomposition method [5][6] completes page decomposition within a few second and, thus, its superiority to the pixel-projection approach is obvious.

References

[1] Ihsin T. Phillips, Su Chen and R. M. Haralick, "English Document Database Standard," *Proc. ICDAR*, Japan, 1993.

[2] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision : Volume I*, Addison Wesley, 1992

[3] J. Kittler and J. Illingworth, "Minimum Error Thresholding," *Pattern Recognition*, Vol. 19, No. 1, pp.41–47, 1986

[4] Nobuyuki Otsu, "A Threshold Selection Method from Gray-level Histogramws," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-9, No. 1, pp. 62–66, January, 1979

[5] Jaekyu Ha, Ihsin T. Phillips and R. M. Haralick, "Recursive X–Y Cut using Bounding Boxes of Connected Components," ISL Report, Dept. Electrical Eng., University of Washington, 1994.

[6] J. Ha, I.T. Phillips and R.M. Haralick, "Document Image Decomposition using Bounding Boxes of Connected Components," ISL Report, Dept. Electrical Eng., University of Washington, 1994.

Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95)
0-8186-7128-9/95 $10.00 © 1995 IEEE