SEGMENTATION DE PAGES PAR COURANTS BLANCS

PAGE SEGMENTATION BY WHITE STREAMS

Theo Pavlidis[1] and Jiangying Zhou[2]

RÉSUMÉ -

Nous présentons une technique pour segmenter une page de document en colonnes et pour séparer les régions de texte des images tramées. Nous ne supposons pas que les colonnes sont forcément des rectangles droits; il est par conséquent possible de segmenter des pages inclinées. La méthode procède par recherche de plages blanches larges; les séquences de telles plages sur plusieurs lignes contiguës sont appelées des "courants blancs". Les régions entre ces courants blancs sont les colonnes. L'angle d'inclinaison est estimé au moment de la formation des colonnes. Le résultat est une partition de la page en colonnes séparées les unes des autres, l'angle d'inclinaison de chaque colonne étant connu. Aucune colonne n'est fragmentée par des coupures verticales intermittentes. La distinction entre texte et images tramées se fait par une technique de reconnaissance de formes utilisant entre autres la corrélation entre les lignes.

Mots clés :Segmentation de pages, séparation texte/images tramées, analyse de documents.

ABSTRACT -

We present a page segmentation technique for partitioning a document page into columns and separating text regions from halftone. The method does not assume that column blocks are upright rectangular regions, hence it can deal with severely tilted pages. Instead, it identifies wide white spaces on adjacent scanlines and forms sequences of such spaces called "white streams." Columns are then found as the regions between white streams. The tilt angle is also estimated during that process. The result is a partition of a page so that each column is isolated and is placed in accordance with the tilt angle. No column is fragmented by spurious vertical cuts. Text is separated from halftones by a pattern recognition technique using such features as correlation between scanlines.

Keywords : Page Segmentation, Halftone Separation, Document Analysis.

1) Computer Science Department, 2) Electrical Engineering Department; both SUNY at StonyBrook, NY11790

1. Introduction

Page segmentation and block classification are the early stages of an automatic document recognition system. We summarize here the current state of a page segmentation system that partitions printed pages of unconstrained multi-media documents into classified regions of text, halftone, graphics, etc.(See [1] for a complete account of the method).

Automatic document segmentation has been explored by many researchers. The challenge is to develop methods that could recognize text pages with complicated layout in a practical environment, especially documents where text is mixed with graphics and halftone images. Many of the current methods are based on the assumption that the printed pages are primarily made up of rectangular blocks[2,3]. This assumption is not valid when the input page is skewed or shear has been introduced during printing.

Published methods deal with tilt or shear by time consuming preprocess. Among them, Baird[4] suggests a preprocessing before column segmentation that first estimates skew angle and then employs a correction process to rotate each connected component. Another approach for handling skewed images, based on the Hough transformation, is discussed in Srihari[5]. For a detailed review of earlier work see [1].

We describe here a method which does not assume that column blocks are upright rectangular regions. In our method, we make no effort to find and correct the skew angle of the input page in advance, instead estimation of the skew angle is accomplished within the recognition and formation of column blocks. The result is a partition of a page so that each column is isolated and is placed in accordance with the tilt angle. No column is fragmented by spurious vertical cuts. Labels such as "*text*", "*halftone*", etc. are then assigned to each of the column blocks by a classification process.

2. Segmentation of Page into Columns

The task of page segmentation is to partition a document page into subregions (blocks). Each of which ideally contains only one type of data such as text, graphics, halftone, etc. Fragments due to horizontal cuts are tolerable.

2.1. Model and Assumptions

We model an *isolated* column as a subregion of the input page containing a unique type of data, surrounded by straight streams of white spaces. Up to this stage, no restriction is imposed upon the layout of the page. Non-textual graphics or halftone images could be present anywhere in the page. Text could have various fonts and their size may vary arbitrarily within the page, although they must lie within(usually loose) known bounds.

We assume that each text line consists of a set of character-images that share a straight *baseline*. Baselines are not required to be horizontal, but all the baselines within a column should be parallel. Columns in a page could have different skew angle(our skew estimation algorithm will give each column block an estimated angle respectively). Shear distortion, if not severe, is acceptable.

We further assume that columns are separated horizontally by white spaces which are wide enough to be distinguished from other spacing such as the white spacing between words, etc. In addition, such white spaces form a continuous stream, while white gaps between characters or words are fragmentary, lasting no more than the height of a textline in the vertical direction. Without any domain knowledge, we tacitly assume that a column is bounded vertically by blank spaces of sizes larger than the average size of the blank spaces that separate text lines. (e.g. twice as large as the nominal size of the largest text size)

2.2. Outline of The Method

In contrast with much of the earlier column recognition work, our method does not focus on the foreground (e.x. text, graphics, etc.) regions, but it analyzes the wide white spacing in a page. Of the published methods, Baird's[4] is the closet in spirit to ours. However, instead of developing a sequence of partial background covers from a time consuming exhausting searching process enumerating all maximal empty rectangles, our first step is to collect the white gaps in scanlines.

Simply picking up wide white gaps in a scanline as column gaps is unsatisfying in general case, due to several commonly occurring problems: (1) The white spacing between two descendants (ascendants) letters in English, for example, yields false column interval(Figure 1a). (2) Similar spaces existing between characters across columns may widen column intervals, resulting irregular column stream(Figure 1b)

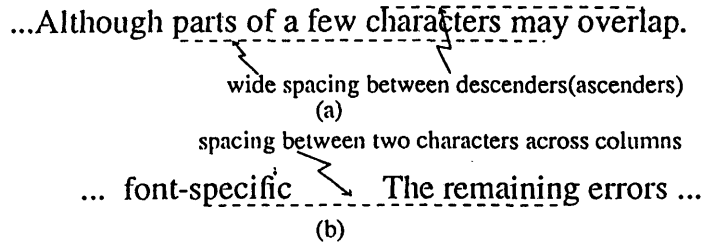

Figure 1: Examples of wide white spaces that do not correspond to column separators.

These problems are circumvented by using a *vertical projection profile* over a blocks of scanlines. Many prior workers have attempted to use projection profiles, especially for segmentation of pages into columns, columns into lines, and lines into characters. They usually take it over the whole page and therefore the results are not reliable because of skew. We take the projections only over a short height (typically less than 0.5") of scanlines so skew does not obscure column gaps.

Column intervals recognized from vertical projection profiles are merged and clustered. The process, from this point on, is geared to form column blocks as large as possible, while maintaining the assumption that each column block corresponds to a unique type of data valid. First, column blocks, which are converted from column intervals found in each scanline block, are merged. The merging is devised so as to smooth out the raggedness of column blocks due to printing defects subject to the constraint that it does not smear the shift of column blocks due to skew. Column blocks are then assembled by cluster analysis according to the alignment of their centers, skew angles are estimated meanwhile. After another merging process being applied to the skewed column blocks, column blocks are outlined. The overall algorithm is listed in table I

2.3. Recognition of Column Gaps

The vertical projection profile is a function $P(i)$ mapping a horizontal position to the number of black pixels in the vertical column at that position. Most of the white gaps between letters or words produce low but non-zero values of the vertical projection because such gaps last at most for a text-line height in the vertical direction and often overlap the black segment of characters below or above. It happens that $P(i)$ has zero value within some text or halftone regions. However, it is easy to distinguish them from column gaps by taking advantage of the fact that the run lengths of these white gaps are relatively short. Those sporadic white spots are further pruned by an embedded smoothing operation in computing vertical profile function, wherein any two black pixels (or white

Table I. Outline of the Segmentation Algorithm

1. Compute the vertical projection over group of scanlines, look for long white intervals from the projection.
2. From the long white intervals construct column blocks by the following steps:
   a. Convert column intervals into column blocks, merging small blocks into larger blocks.
   b. Cluster blocks into a set according to their alignments, estimate skew angle for each set.
   c. Outline column blocks.
3. Label each column block.

pixels), which are equal to or less than a certain threshold apart, are merged into a continuous stream of black (white) pixels. Thus, In practice, a zero value of $P(i)$ almost always corresponds to column intervals. This is true even the input page is tilted as much as 30^o, since we restrain the projection only over a limited number of scanlines.

Line drawing areas or tables may produce many white fragments in $P(x)$, causing ambiguous column segmentation. White fragments in light halftone image and complicated graphics area are even more difficult to characterize. While it is possible to modify the column recognition algorithm so that those fragments could be recognized and removed immediately, for the reason of economy, the solution is deferred to later stage, when structural and statistic information are available.

Based on the above observations, we identify column intervals as those pieces on $P(x)$ where $P(x) = 0$ and the run length of 0's is larger than a predefined threshold. Each block of scanlines yields a set of quadruples: *(Xleft, Xright, Ytop, Ybottom)*, where *Xleft* and *Xright* are the left and right sides of the column interval. *Ytop* and *Ybottom* are the top and bottom location of the section in a page.

2.4. Construction of the Column Blocks

In this stage we move from white column intervals to "*gray*" area (text, etc.) and we perform the clustering according to the latter. The column intervals recognized in each block of scanlines are clustered, defining the segmentation, one column to a block, in this stage. Skew angles are estimated during the clustering. The clustering process is done by a sequence of steps which are described in detail in [1].

We start the process by first linking the right side of each column interval block with the left side of its consecutive neighbor in the horizontal direction, forming a column block. The features of a column block are more revealing.

The main merging process is done simultaneously with the recognition process. Any new found column intervals in the current block of scanlines will be merged into an existing column interval if they satisfy the conditions:

(1) the two intervals are adjacent in the column direction

(2) one interval contain another in the column direction or

(3) one interval overlaps in most part with another

A refining process is thereafter employed to merge very narrow blocks such as those produced by isolated characters or very short text lines, and those blocks that are narrow in vertical direction, usually containing only a fragment of a single textline, into major blocks adjoined to them. The refining process is performed repeatedly until no change is possible.

The next step is to estimate the skew angle of the input image. This is done by first classifying the set of column blocks into subsets such that the centers of column blocks within each subset are

approximately aligned forming a straight line: $ax + b$. The skew angle, which is defined as $tg^{-1}(a)$, is obtained by a least-square line fitting algorithm. We then rule out aliasing skew due to irregularity of textlines in the light of majority principle. If the average estimated skew angle is less than a predefined value then we consider the input page is roughly upright, and no correction is applied. Otherwise, the left and right lines of each column blocks are rotated according to the skew angle.

After the clustering is finished, column blocks are once again submitted to a merging process to merge column blocks in the direction of the skew angle. The criterion used in guiding the merging process is: *If two blocks with close skew angles, touching each other in the vertical direction, have similar width and overlap each other mostly, then they are merged into one.* At this point all possible merges are explored and performed in accordance with the skew angle.

3. Classification of Column Blocks, Experiments and Discussion

Upon forming the column blocks, the next step is to classify each column block into text, halftone, etc. This leads us to investigate the statistics of the run length segments. The statistics of the run length segments has been explored by many authors[6,7]. Features such as *horizontal white-black transitions, mean length of black run, black-white pair run*, etc. are commonly used in identifying the contents of column blocks. We have been proposed using features such as the ratio of mean length of black intervals versus the mean length of white intervals, number of black intervals over a certain length, the total number of intervals. Another reliable classifier which is absent from the literature is offered by scanline correlation between two lines r distance apart. Such features were used successfully in our system as a pre-classification for eliminating large areas which definitely are not text. Both analytic and experimental results demonstrate that the scanline correlation could serve as a good measure in discrimination text areas from halftone areas(See [1] and [8] for details).

The method was tested on a series of documents scanned in by a Ricoh 100 scanner which produces a binary image of size 2512×3400. We have run experiments over pages from magazines, having multiple columns and complicated layout. We include here the examples(Figure 2-4) showing the column segmentation results of various kinds of text pages mixed with halftone pictures, graphics, table etc. as well as document pages tilted in various angles. The process requires about 10 seconds per page on a SUN 3/160 Workstation running C code which has not been optimized for speed.

Several factors may influence the result of column recognition. A potential obstacles to the application of this method may arise if by accident the scanned page is severely tilted. In this case, the vertical projection profile can fail to reveal the column gaps due to the unalignment of consecutive textline in vertical direction. This obstacle can be largely but not entirely overcome by restraining the size of scanlines block to be not larger than one or two text lines. Actually, the problem is not as serious as it seems to be because in practice the skew angle will not be arbitrarily large. A slew angle within $\pm 5^o$ from horizontal is more realistically anticipated. For a textline with height of 0.15" and the column's gap is 0.3", it will take about *700* scanlines with skew angle of $+5^o$ to overlap the whole column gap.

Another situation that may confuse the segmentation happens when a scanline block covers different type of layout regions, for example, part of a scanline block may contain single column text, while another part contains texts belonging to two columns. In this case the profile will miss the column gap. Once again, restricting the size of scanline block to be no larger than the size of a textline can alleviate the problem, but an unfortunate side effect is that such a size may yield a column partition that has too many fragments, thus jeopardizing column clustering.

Acknowledgment: This work was supported in part by a grant from Ricoh R&D of Japan. We want to thank Dr. Shunji Mori of Ricoh R&D for his encouragement of our work. The scanline correlation strategy for classification of text versus halftone was developed while T. Pavlidis was a consultant for Ricoh R&D.

Reference

[1] T. Pavlidis, J. Zhou, Page Segmentation and Classification, *Technical Report*, SUNY at StonyBrook

[2] F. Wahl, et al. Block Segmentation and Text Extraction in Mixed Text/Image Documents, *CGIP* **20**, pp 375-390, 1980.

[3] Nagy, G. et al. Document Analysis with an Expert System, *Proc. Pattern Recognition in Practice* **II**, pp. 19-21, June, 1985.

[4] H. Baird, et al. Image Segmentation by Shape-Directed Covers, *Proc.* **10***th ICPR*, Atlantic City, NJ, pp 820-825, June, 1990.

[5] S. Srishari, Document Image Understanding, *Proc.* **7***th ICPR*, Montreal, Canada, pp. 87-96,1984.

[6] K. Wong, et al. Document Analysis System, *IBM J. Res. Develop.* **26**, No. 6, 647-656, 1982

[7] D. Wang, et al. Classification of Newspaper Image Blocks Using Texture Analysis, *CVGIP* **47**, 327-352, 1989

[8] J. Zhou, T. Pavlidis, Discirmination Between Text and Halftone Areas in Document Pages, *Technical Report*, SUNY at StonyBrook

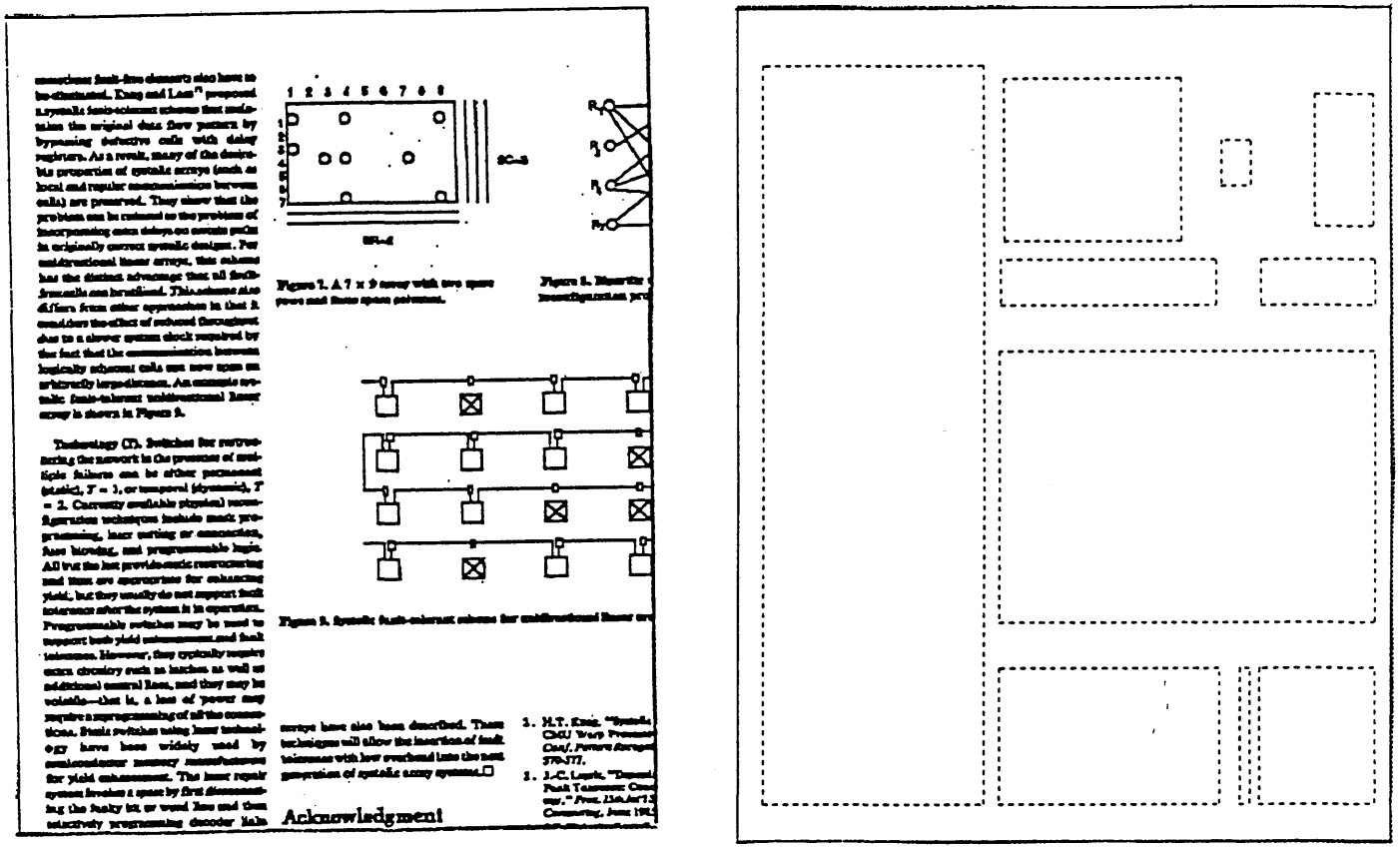

Figure 2: a. Input page with graphics. b. Segmentation result.

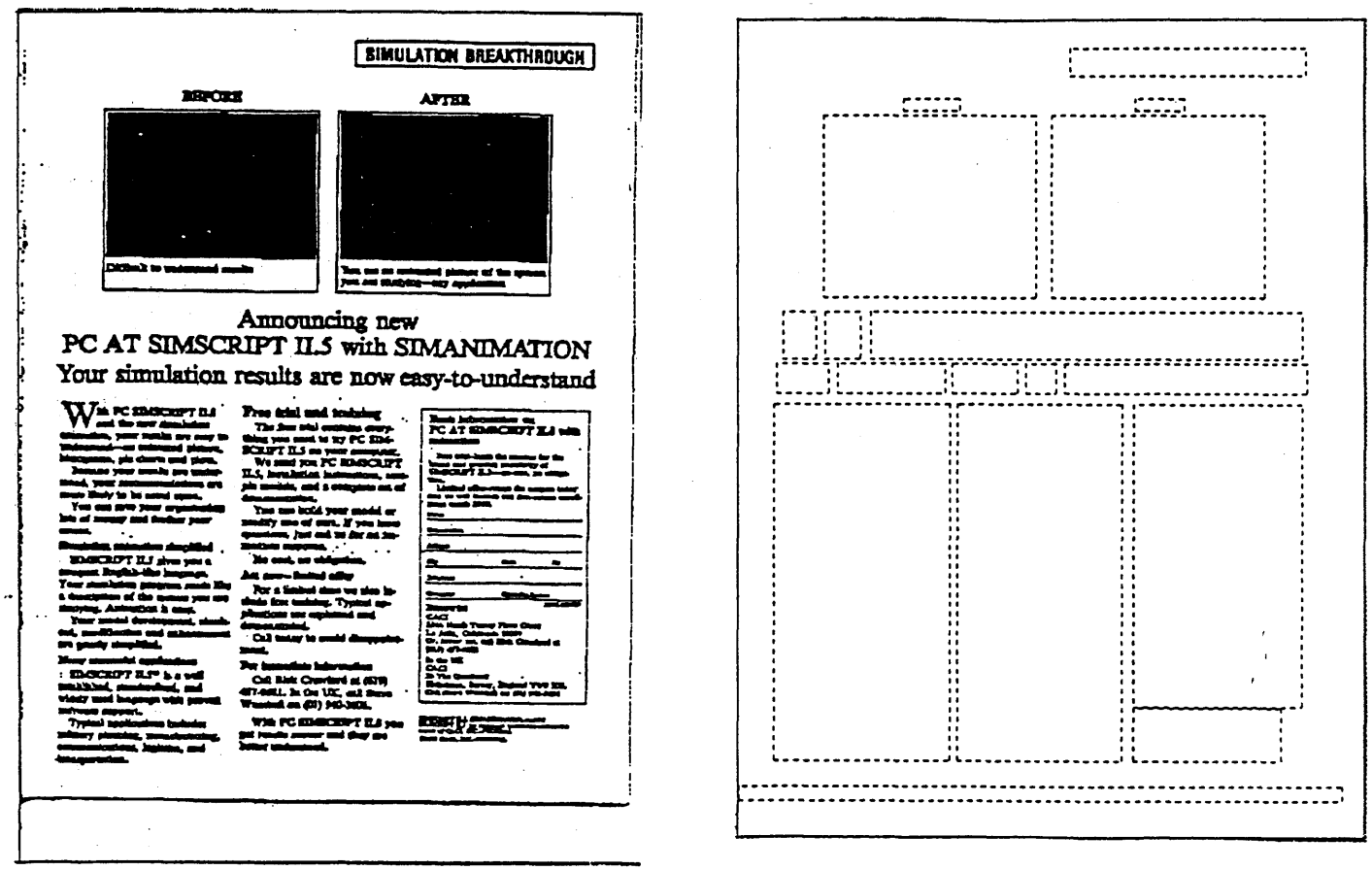

Figure 3: a. Input page with complicated layout. b. segmentation result.

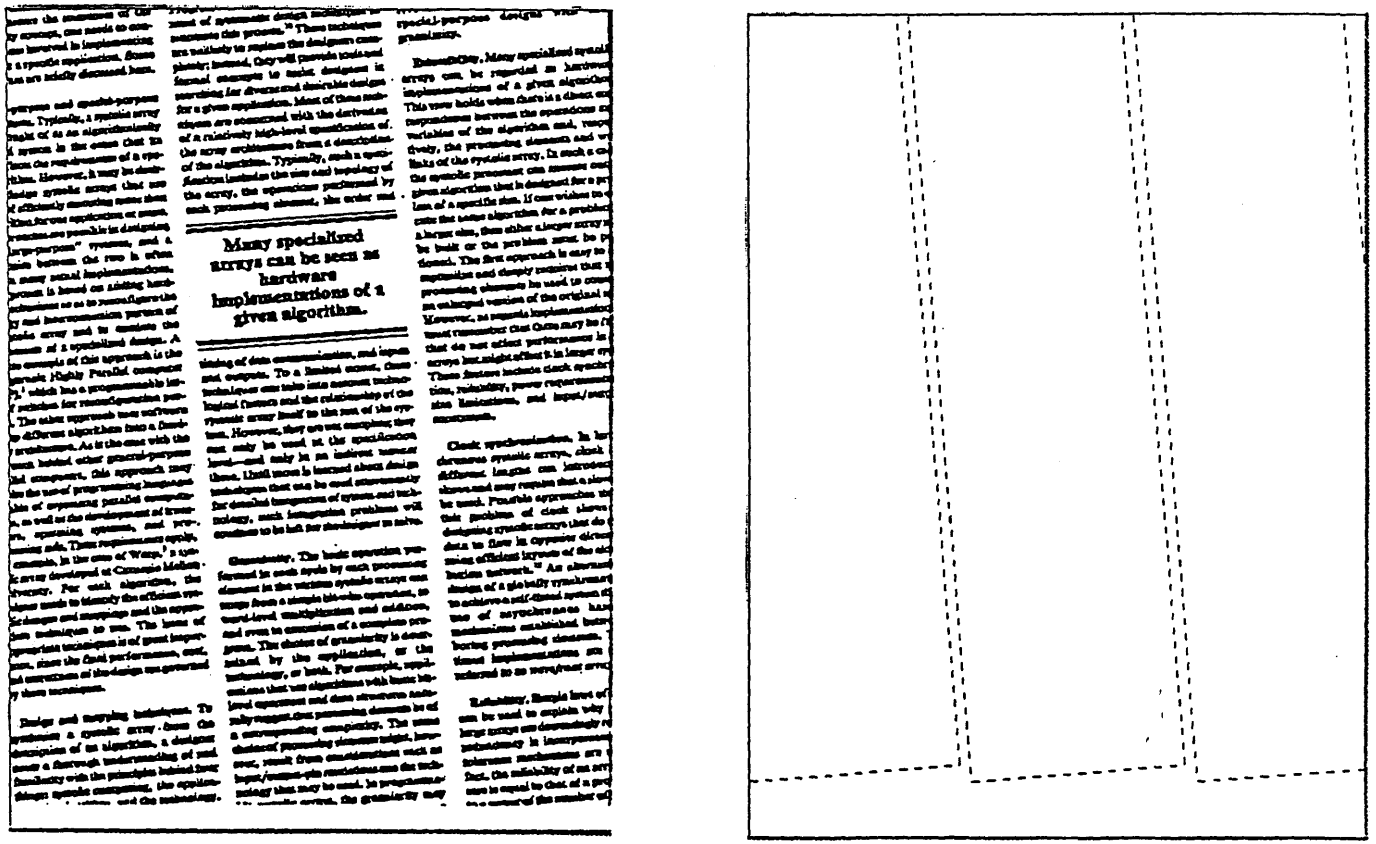

Figure 4: a. Input page with titled angle. b. segmentation result.