MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

A Fast And Robust Approach For Document Segmentation And Classification

Han Wang Stan Z Li S Ragupathi
School of Electrical & Electronic Engineering
Nanyang Technological University, Singapore 639798
e-mail: hw@ntuix.ntu.ac.sg

ABSTRACT

We report in this paper a robust implementation of document segmentation and classification. In this algorithm, top-down approach is adopted due to its fast processing speed and reliability. In addition, the approach accounts for every portion of the processed page and it is also capable of image display at different scale and allowing editing of classified blocks such as changing, merging and splitting of blocks. In classification stage, relevant attributes extracted out from the segmentation process are utilised. The algorithm was tested on a number of sample document images and the results were satisfactory.

1 Introduction

Document Image Processing (DIP) System can provide a mechanism for archiving large volumes of documents such as reports, books, legal or financial transaction. It provides an efficient way of implementing electronic document filing systems. The preprocessing steps of a DIP system is **segmentation** and **classification**. The scanned and digitised document page will be separated into various blocks of single data type and classified as one of few basic classes: texts, paragraphs, lines, graphs and photographs.

Most of the existing document segmentation and classification methods described in literature can be classified into three broad categories: top-down[3, 2], bottom-up[5, 1] and hybrid[4]. A top-down control strategy recursively segments large regions into smaller sub-regions. A bottom-up control strategy starts by grouping pixels of interest together and progressively merging it into larger regions. A hybrid control strategy is the combination of both top-down and bottom-up control strategies. Each of these three strategies exhibits its own strengths and weakness when applied to diverse documents.

A top-down control strategy typically starts at high level by hypothesizing a series of interpretations and attempts to verify each by searching through the nodes of a tree of implied hypotheses, finally the lowest nodes (leaf) are consulted for evidence. Typically a depth-first with fully backtracking search method is applied.

Using recursive X-Y cuts or recursive projection profile cuts is one of the method to split a document into a set of blocks. At each step of the recursive process, the projection profile is computed along both horizontal and vertical directions; a projection along a line parallel to, say the x-axis, is simply a sum of all the pixels' values along that line. Then sub-division along the two directions is accomplished by making cuts corresponding to deep valleys, with width larger than a predetermined threshold, in the projection profile. The application of cuts is based on the configuration of the pixels. Based on the observation that the printed pages are primarily made up of rectangular blocks, a page can be recursively cut into rectangular blocks. Thus the document is represented in the form of a tree of nested rectangular blocks. Each node in the tree corresponds to either a set of rectangles obtained by horizontal partitions (X-cuts) of the parent rectangle, or a set of rectangles obtained by vertical partitions (Y-cuts). The X-cut-sets and Y-cut-sets of horizontal and vertical partitioning alternate strictly, level by level. The first partitioning may be arbitrarily set to either horizontal or vertical direction. The result of the segmentation process is a tree which corresponds to the entire page.

The advantage of the top-down approach is that high speed can be achieved and the page is guaranteed to be completely accounted for. Since only rectangles are generated, identical processing steps are able to be applied at every level; where the elegant recursive programming comes into picture. The main limitation of the top-down approach is that tables, irregular layout documents and forms cannot be successfully segmented.

2 The segmentation algorithm

In this project, bi-level images are used as the input to the segmentation and classification algorithm. Bi-level images normally having black pixels as foreground (represented by logical '1') and white pixels as background (represented by logical '0'). Each bit rep-

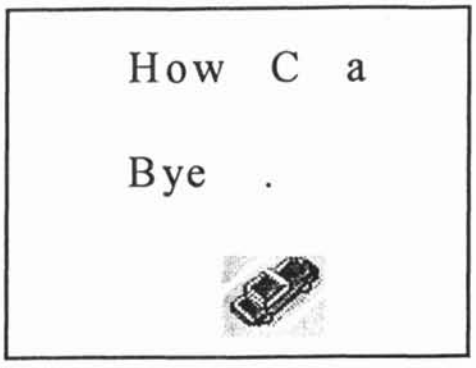


Figure 1: *Sample document to be segmented*

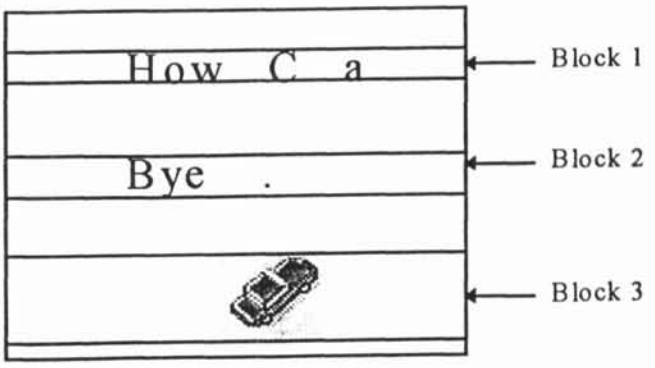


Figure 2: *Framing of document image after horizontal profiling*

resents one pixel image element and they are aligned with byte boundary. That is each row of the document images is represented by an integer number of data bytes. The size of the document image is specified by the number of rows and the number of data bytes within each row.

Segmentation serves to divide an image into various block sizes depend on the layout of the document. Each block can be text, graphs or photos and are used for later processing. Segmentation is achieved by horizontal and vertical profile of the image. After obtaining the image length and width, profiling is done horizontally for every row of the image. This will indicate those regions that contain black pixels. Then vertical profiling take place for every column of respective blocks for further partition. Each partitioned block then goes through horizontal profiling again to frame up individual character. Only characters in fields that required to be recognized are framed up.

The segmentation algorithm developed in this report uses the Top-Down strategy called Recursive X-Y Cut. The algorithm consists of following steps:

1. Scan through every row and column of the image to count the number of black pixels.
2. Perform horizontal and vertical profiling alternately to frame up separate characters. Each character in the same row is checked with certain threshold value to decide whether they are to be grouped together.
3. Some of the groups are merged together to form text line blocks.

Segmentation Process By Example
The segmentation algorithm will be applied to Fig 1 and the process is described as follows.

Initially, the horizontal profiling is carried out. After the horizontal profiling, the profile array will store

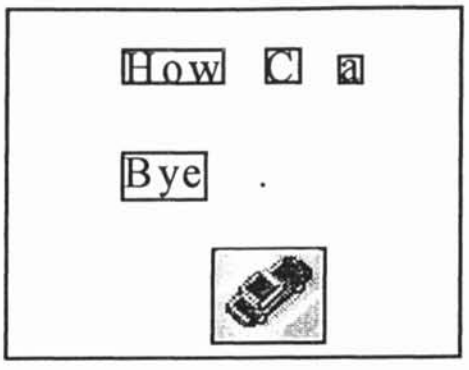
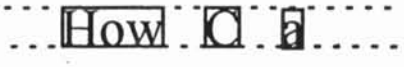


Figure 3: *Final result of segmentation of the sample document image*

the number of black pixel per line. Consecutive black pixel lines were framed up and followed by a routine to merge the blocks by comparing the gap in between the blocks. The result is shown in Fig 2.

Each block is passed to the algorithm for vertical profiling. As character 'H', 'o' and 'w' are closed enough, they are merged as a block. While the character 'C' and 'a' are group as individual blocks. The following shows the result for the first pass of vertical profiling:



Subsequently, all the three blocks: 'How', 'C', and 'a' are sent for horizontal profiling for the second time. Note that the block 'a' frame fit exactly to the character as shown below.

Figure 3 shows the final result of the segmentation process for the sample image.

3 Classification

After segmentation, a link list was formed with all the connected components. The size of these components

vary, some small, some very large. These connected components can be either text, photos, graphs and lines. The second stage of the algorithm is to classify each block in the list and if necessary, merge the blocks of the same type into large blocks. The following rules are applied for classification:

(1)Text line: sequence of adjacent character blocks of similar height, separated by inter-character spacing rule.
(2)Paragraph: sequence of blocks of text lines of same height, separated by spacing rule.
(3)Column: sequence of paragraph blocks of same width, separated by paragraph-cutting rule.
(4)Photo: large block with approximately equal frequency of '1' and '0' pixels.
(5)Drawing: large block with low frequency of '1' pixels.

The classification process is carried out in the following sequence.

1. Filter out small size blocks.
2. Classify each block into one of the five types. To simplify the classification process, reasonable assumptions are made, such as the minimum text font size, maximum font size, minimum size of the graphic block, etc.
3. Determine whether the document is horizontal or vertical layout. For mixed layout document, the main layout direction will be determined. For horizontal layout document, horizontal merging will be performed. Likewise for vertical layout document, vertical merging will be performed. For mixed layout, merging will be performed on both directions.
4. Merging will be performed in several steps.

For horizontal merge, the follow steps will be carried out

1. Sort all the blocks based on their Y values followed by their X values. During sorting, if the difference between the Y values of two blocks is less than a predetermined threshold, their order will be determined by their X values with the smaller X values being placed in front. Bubble sort algorithm is used.
2. Spilt a block list into two separate lists : one contains non-text blocks and another one contains only text blocks.
3. Merge same row small block text blocks into one bigger block. If the text blocks are in different columns, they will not be merged.
4. Merge different row text blocks into paragraph blocks. Only those text blocks having same width, with reasonable vertical line spacing between them and their overall length greater than a predetermined value will be merged. During this step, no overlapped condition will be checked.
5. Merge different paragraph blocks into bigger paragraph blocks. The blocks with their X values or X+W values being the same, or having small difference (less than maximum indentation) and no overlap existing will be merged.
6. Eliminate picture in picture blocks. This step will be executed only if step 5 is performed.
7. Combined two linked lists (non text and text) into one.

4 Experiments and discussion

Figure 4,5 and 6 show the sample document, the results of the segmentation and classification process. Fifty over documents had been scanned and tested by this algorithm. In general, the algorithm works fine for documents that aligned and scanned correctly. It is effective and robust. However, there are still room for improvements and enhancements, for example, to cater for large skew angle.

ACKNOWLEDGEMENTS–authors would like to thank Lim Chin Thiam and Lim Chong Min for their diligent work that brings the best out of the project.

References

[1] Wang D and S N Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics and Image Processing*, 47:327–352, Sept 1989.

[2] J Higanisho, H Fujisawa, Y Nakano, and M Ejiri. A knowledge-based segmentation method for document understanding. In *Proc. 8th ICPR*, pages 745–748, Paris, 1986.

[3] G Nagy, S Seth, and S D Stoddard. Document analysis with an expert system. In *Proc. Patern Recognition Practice*, volume II, pages 149–159, Amsterdam, 1985.

[4] M Okamoto and M Takahashi. A hybrid page segmetation method. In *Proc. 2nd Int'l Conf. on Document Analysis and Recognition*, pages 743–748, Tokyo, 1993.

[5] F M Wahl, K Y Wong, and R G Casey. Block segmentation and text extraction in mixed text/graphics images. *Computer Vision, Graphics and Image Processing*, 20:375–390, 1982.

Figure 4: *Original document image; This example shows that the algorithm can handle text other than English*

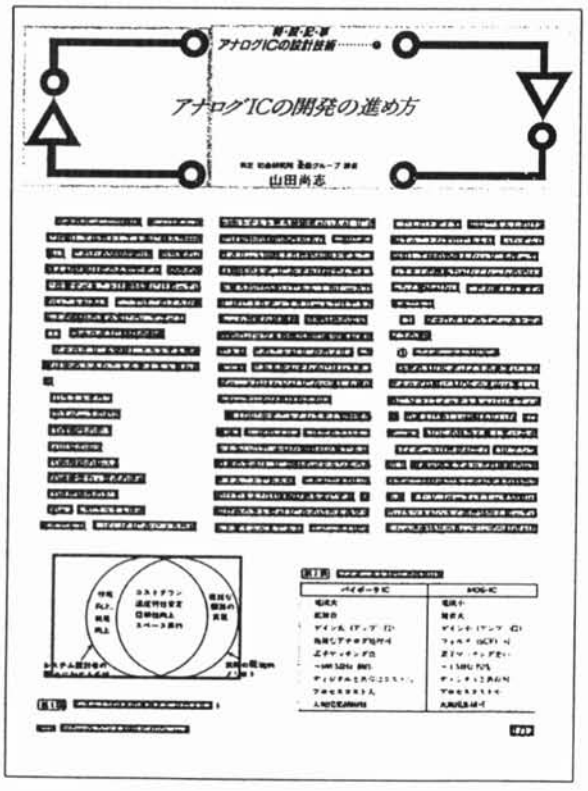


Figure 5: *Result of the segementation process*

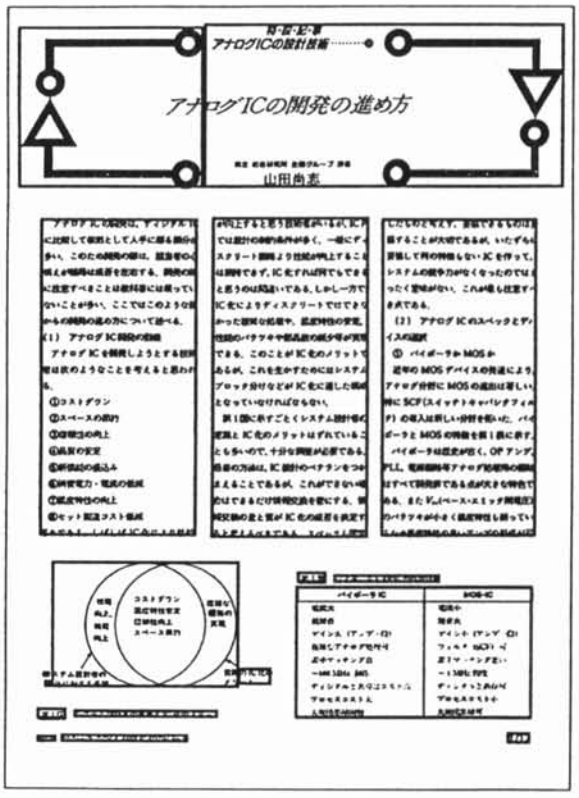


Figure 6: *Output of the classification algorithm*