MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

A High-Speed Document Image Classifier

Shao Lejun

School of Electrical and Electronics Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 2263

S. Ragupathi, and Teo Kiem Chua Edgar

Information Technology Institute
71 Science Park Drive, NCB Building, Singapore 0511

Abstract

In this paper, a high-speed document image classification algorithm is presented. The algorithm is based on the bottom-up strategy which can successfully segment and classify any type of sophisticated layout documents without the limitation of Manhattan rules. Special techniques are employed to overcome the slow speed problem facing most of the bottom-up algorithms. During segmentation process, the algorithm used a byte-based operation to establish a list of connected components within a scan line, and to merge touching connected components in adjacent scan lines. The texture features of each connected component are also obtained during the process. This greatly reduced the computation time spent on classification stage. For a typical A_4 size document image with 200DPI resolution, the average processing time on a Pentium based IBM-PC computer is 1.2 second.

1. Introduction

Document image classification is the first yet very important step in a document image processing and understanding (DTPU) system. Its task is to classify the document image into different types of regions: text regions, graphics regions, and image regions for later processing. [4] gives a broad overview of early work. Since then, various document processing systems have been proposed [1,2,3,5,6,7,8].

In spite of considerable research, most current commercial systems still rely on manual segmentation and classification [6]. It seems that the lack of practical system is mainly due to the following two reasons: the impractical assumptions about the layout features of a document image, and the slow processing speed. In this paper, a new document image classification algorithm is presented. The objective of our research is to try to develop a robust system to classify the document image at an acceptable speed.

This paper is organized as follows. Section 2 briefly describes some recent work on document image classification. Section 3 describes our new algorithm. Section 4 gives experimental results that demonstrate the high-speed and robustness features of the proposed algorithm. Finally, a few concluding remarks are made in Section 5.

2. Review of Recent Work

Currently, most of the document segmentation and classification systems fall into one of the following three categories: *top-down*, *bottom-up* and *hybrid*. A top-down control strategy recursively segments large regions into smaller subregions. Using recursive X-Y cuts or recursive projection profile cuts is one of the methods to split a document image into a set of blocks. Examples of such an approach can be found in [5,9]. This control strategy, if carefully designed, can achieve a very high processing speed. The algorithm proposed by [8] can run well under one second if a 486/33 IBM-PC computer is used for a A_4 size 200DPI document image. The main limitation of the top-down approach is that it can only apply to documents with the layout complying with the Manhattan rules. Because of this limitation, tables, diagrams, forms cannot be successfully segmented in top-down strategy.

In bottom-up method, the control strategy starts by grouping pixels at the lowest level of detail and then rises to higher levels until the whole page is completely assembled. Most recent algorithms employ this approach [1,2]. The advantage of the bottom-up approach is its accuracy and generality. Any sophisticated layout document can be processed. The main problem associated with bottom-up approach is the slow speed due to the excessive computation in the lower level, and the time spent on early decisions made based on evidence from the smallest samples which often lead to the unwanted rapid accumulation of the mistakes. The algorithm proposed by O' Gorman [2] spent an average of 65 seconds to process a 300DPI image of PAMI article page running on a

UNIX Sun Workstation. To speed up the process, some used reduced document image. To copy with this problem, some approaches process reduced images. In [3]'s algorithm, the original 300 DPI image is reduced by a factor of 8 and 3 in horizontal and vertical directions respectively. In [7]'s approach, the original 400 DPI image is reduced by 1/8 and 1/4.But this may introduce some inaccuracy of the segmentation.

3. Proposed Algorithm

There are two main objectives of our proposed algorithm:

1. The algorithm should be quite robust. There should be no limitation on the layout features of the document images and it should give accurate classification results.

2. The algorithm should be fairly fast. The goal is to limit the processing time to be around 1-2 second when it is running on a popular IBM PC or compatible machine for a A_4-size 200 DPI document image.

To achieve the objectives, we have developed a new algorithm. The algorithm is based on bottom-up strategy. It consists of the following steps:

step1. Establish a **black-run** linked list for each scan line of document image. A black-run is the consecutive black pixels in each scan line. We call each black-run as a **line segment** in our algorithm. Each line segment is specified by the scan line index and its starting and ending position in that scan line.

step2. Construct a **connected component** linked list from line segment. A connected component is a group of adjacent line segments which overlap.

step3. Extract major text blocks from the connected components and merge these blocks into text lines and paragraphs.

step4. Classify remained blocks such as graphs, photos, large text blocks, etc.

3.1 Creation of a black-run linked list within a scan line

Traditional algorithms of establishing line-segment are to scan the whole image and check the status of every individual pixel. This step is the most time-consuming operation in most of the bottom-up algorithms. For an A_4 size 200DPI document image, at least 3.52 million memory accesses are needed. To speed up the line segment extraction process, we have developed an optimized algorithm for it. Some important features of the algorithm are listed below:

1. Byte-based operation is used. That is, during the process, the algorithm only access and process individual image bytes, not individual image pixels.

2. Precomputed tables are constructed during initialization stage to avoid repeated computation. The tables include look-up table and mask tables.

3. State-machine is used to allow the processing of one image byte efficiently.

The basic idea of line segment extraction algorithm is to repeatedly read one byte of image in the same image buffer, and based on the contents (bit pattern) of the byte and the current state of the algorithm to act accordingly. One line segment may start and end within one data byte, and may continue cross several data bytes. Because one data byte can only have 256 different bit patterns, for each bit pattern, the bit position for the first black pixel and the bit position for the last black pixel is fixed. We can use a precomputed look-up table of size 256 to store this information, and use the bit pattern as an index to access this information. During the processing of one data byte, the algorithm may be in several different states. We have identified six states: **start, cont_1, cont_10, cont_20, end, end_start** and the status information is also recorded into the look-up table for quick reference. Other information such as the black and white dot distribution pattern, the number of black/white run, the maximum length of a continuous black/white run, etc. can also be derived from the look-up table entry. This information will be used in later block classification stage. Therefore, no extra processing is needed to classify each segmented block. Experience has shown that the new algorithm has speeded up the processing time by 10 to 20 times compared with traditional methods.

3.2 Establish connected components and identify skew angle

Each time, once a linked list of line segments is formed for a scan line, they will be merged to the connected components formed up to the previous scan line immediately. This feature of forming connected components will greatly reduce the memory demand since only line segments from two adjacent scan lines will be stored for the processing. During this process, the center coordinates of each connected component, the dimension and height to width ratio of the connected component, density of the black pixels in the component, $BWHL$ ratio (the ratio of the number of black-white alternative runs to the number of connected components formed during the merging process) *etc.* will also be recorded for skew detection and block classification purpose.

3.3 Separation of text components from non-text components and merge them into text lines

Normally, a document page may contain graphics drawings, photos, and text blocks of different font sizes, no matter what type of document we are dealing with. However, the majority part of a document page is always covered by the main text body of the same font size. The next step of the algorithm is to identify the major text blocks and separate these blocks from the rest of the blocks. This makes the remaining processing much easier.

Five heuristic rules are used for this purpose. The heuristic rules are derived from the observation that for all the white-runs (consecutive white pixels) in a scan-line, its lengths (white-length) are approximately equivalent to one another if it belongs to a text-line (a portion of row pixels across text) having the same font size and style. The same holds true for black-runs in a text-line. Once, text components are identified, text block growing techniques are used to further enlarge the text component into text lines and/or paragraphs.

3.4 Classify graphics, halftone and large text blocks

Classification of graphics, halftone, and large text blocks is the last step of the whole process. The texture features used to classify the remained blocks have been recorded during segmentation stage. This makes the classification process quite easy. The halftone blocks are usually filled with plenty of intermixing of black and white pixels to simulate gray scale pictures. In our algorithm, halftone blocks are identified by two parameters: the high percentage of the number of black pixels and the high occurrence of alternative run of black to white pixels in the block.

Graphics blocks are formed by line drawings such as tables, figures, *etc.*. They usually establish a low black to white pixel ratio and certain black and white alternative run pattern. Large text blocks can be separated from graphics blocks by using the following two features: 1). they have thicker strokes; and 2). they have approximately the same height, close to each other and in the same horizontal or vertical position.

4. Experimental Results

The algorithm presented in previous sections has been implemented in C language and run under Microsoft Window 3.1 environment. To examine the performance of the algorithm, more than 200 real document images have been used and the testings were carried out on a Petium/60 IBM-PC compatible computer. The results of eight sample images are given in the following two tables. Table 1 gives the dimensions (in pixels), the number of line segments, the number of connected components, and the number of classified blocks for each sample document image. Table 2 gives the time (in seconds) spent on segmentation and classification process. It could be found from the two tables that classification takes much less time. On the average, the total processing time for each image is about 1.2 second. Fig. 1 and Fig. 2 gives the segmentation results for some Japanese documents.

Filename	Width (pixels)	Height (pixels)	Number of line segment	Number of Connected Components	Number of Blocks
Sample 1	1,516	1,746	66,984	2,784	37
Sample 2	1,461	2,006	40,103	950	75
Sample 3	1,465	1,993	75,969	1,462	22
Sample 4	1,432	2,161	77,167	4,410	63
Sample 5	1,456	1,996	74,547	1,452	47
Sample 6	1,871	1,371	34,122	1,450	134
Sample 7	1,356	1,776	43,974	2,227	192
Sample 8	1,616	2,168	56,535	5,995	42

Table 1 Dimension, no. of line segment, connected component, blocks

Filename	Segmentation time (sec)	Classification time (sec)	Total time (sec)
Sample 1	0.82	0.33	1.15
Sample 2	0.66	0.06	0.71
Sample 3	0.88	0.11	0.99
Sample 4	1.21	0.27	1.48
Sample 5	0.93	0.11	1.04
Sample 6	0.55	0.17	0.71
Sample 7	0.77	0.82	1.59
Sample 8	0.71	0.22	0.93

Table 2 Time taken running under Intel's Pentium/60 CPU.

5. Conclusion

A new algorithm for document image segmentation and classification has been presented. It used bottom-up strategy. The algorithm is very robust. It can process very sophisticated layout of document in a very short period of time without the memory space tradeoff. The algorithm has been successfully implemented and run under Microsoft Window environment. More than 200 document images of different types have been tested with very good results.

References

[1] Yutaka Hata, Xianfeng He, *etc.*, *"Japanese document reader system"*, Proc. Int'l Conf. on Image Processing, September 1992, Singapore, pp.193-197.

[2] Lawrence O'Gorman, *"The document spectrum for page layout analysis"*, IEEE Trans. PAMI, Feb. 1994.

[3] F. Lebourgeois, Z. Bublinski, and H. Emptoz, *" A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents"*, Proc. 11th Int'l Conf. on Pattern Recognition, August 30-September 3 1992, The Netherlands, pp.272-276.

[4] M. Nadler, *"A survey of document segmentation and coding techniques"*, Computer Vision, Image and Graphics Processing, Vol. 28, 1984, pp.240-262.

[5] George Nagy, Sharad Seth and Mahesh Viswanathan, *" A prototype document image analysis system for technical journals"*, Special issue of IEEE Computer on Document image analysis systems, July 1992, pp.10-21.

[6] Theo Pavlidis and JiangYing Zhou, *"Page segmentation and classification"*, CVGIP: Graphical Models and Image Processing, Vol.54, No.6, November, 1992, pp.484-496.

[7] Takashi Saitoh, and Theo Pavlidis, *"Page segmentation without rectangle assumption"*, Proc. 11th Int'l Conf. on Pattern Recognition, August 30-September 3 1992, The Netherlands, pp.277-280.

[8] Shao Lejun, *"A Prototype Document Image Classification System for Japanese Documents"*, Proc. of 5th Int'l Symposium on IC Technology, Systems & Applications, September 15-17, 1993, pp.673-677, Singapore.

[9] Shao Lejun, Loh Kah Onn, and Hamdan B Othman, *"The Applications of clustering techniques in document image classification"*, 1992 Annual Meeting of the Classification Society of North America, East Lansing, USA, June 11-13, 1992.

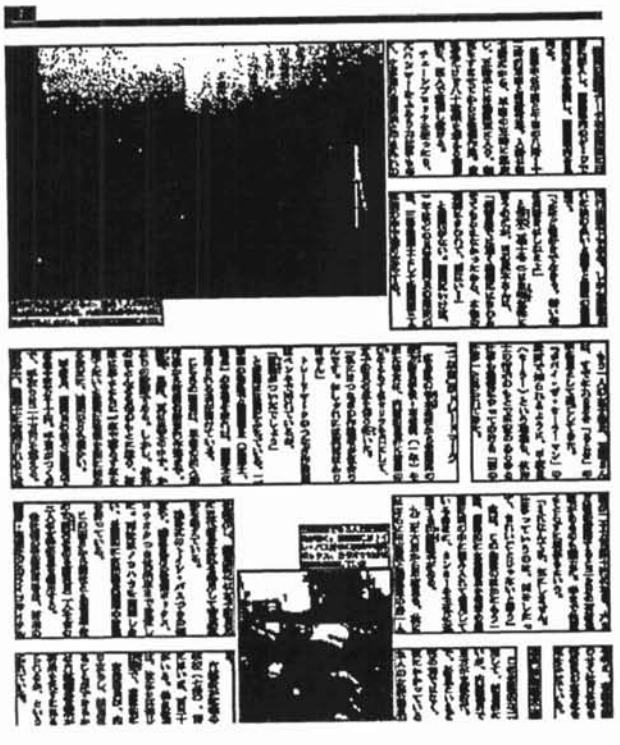

Fig. 1. Document having 0.5*mm* minimum between-block spacing

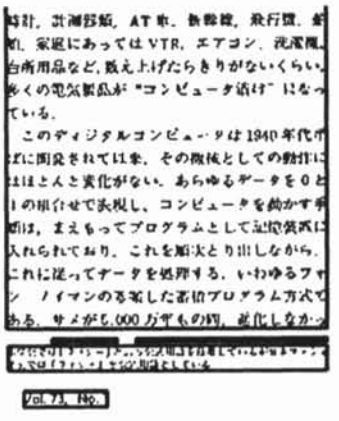

Fig. 2. Varying gaps between-character

Fig.3. Japanese characters with notes