Simple Layout Segmentation of Gray-Scale Document Images

A. Suvichakorn, S. Watcharabusaracum, and W. Sinthupinyo

National Electronics and Computer Technology Center,
112 Phahol Yothin Road, Klong Luang, Pathumthani 12120, Thailand

Abstract. A simple yet effective layout segmentation of document images is proposed in this paper. First, $n \times n$ blocks are roughly labeled as background, line, text, images, graphics or mixed class. For blocks in mixed class, they are split into 4 sub-blocks and the process repeats until no mixed class is found. By exploiting Savitzky-Golay derivative filter in the classification, the computation of features is kept to the minimum. Next, the boundaries of each object are refined. The experimental results yields a satisfactory results as a pre-process prior to OCR.

1 Introduction: Problems Stated

Like most Optical Character Recognition(OCR) systems, Thai OCR is suffered from compound documents, particularly in magazines or newspapers. For example, by doing connected component analysis for recognition, big objects such as images will cause memory run-out. In addition, lines or parts of graphic result in dropping of recognition accuracy. Thus, an effective layout segmentation is required as a pre-process to the OCR. The segmentation is also expected to duplicate the original document's layout to electronic form. On the other hand, the algorithm should not affect much in computation time of the system.

Therefore, a simple yet effective layout segmentation is proposed. The algorithm exploits texture-based multiscale segmentation and a set of post-processing rules to refine object boundaries. The computation is at a low level, by deriving classification's features from Savitzky-Golay filter[1], but the accuracy rate is *comparable* to those using DCT or wavelet decomposition[2] in recent techniques.

This paper is organized as follows. In section 2, the algorithm is described. Next, the experimental results and the conclusion of this study will be presented in sections 3 and 4 respectively.

2 Algorithm

The Classifier consists of two parts: *block classification* and *boundary refinement*. In block classification, each block is labled as background, line, text, images or graphics. Then, boundary refinement is applied to obtain accurate boundaries of each object. In this part, tables are also extracted from graphic objects using a set of rules.

D. Lopresti, J. Hu, and R. Kashi (Eds.): DAS 2002, LNCS 2423, pp. 245–248, 2002.

2.1 Block Classification

We starts the block classification with an initial block size of 64×64 pixels. In each block, 5 features are calculated to catagorize the block as background, text, lines, graphics or images. Blocks, which contains many objects and do not fall in any stated classes are assigned to an intermediate class. Such cases will be analyzed by splitting the block into 4 sub-blocks. The process continues until no intermediate class is found or the block size is 16×16 pixels. The smaller block size than this provides not enough information for classifying and may confuse the classifier what it actually contains.

Classification's Features. Since one requirement of the segmentation is minimum computation, the features we selected here are easy to compute but have high performance in classification. First, mean(μ) and standard deviation(σ) of intensity(I) are used to separate background and images from text and graphics.

Note here that text in non-white background may have μ and σ close to those of images. Moreover, such text could not be recognized correctly without binarization. To solve the problem, we calculate active pixels(α), using the method of adaptive thresholding in binarization, to find amounts of pixels that should be active or black when the binarization is applied.

$$\alpha = \sum_{blocksize} (I < \mu - k \cdot \sigma), \tag{1}$$

where k is a weighting factor. This is the easiest way to compute thresholds. The faster and more effective can be found in the most recent research [3].

Next, we will find more features to classify text and graphics. These two classes have similar global characteristics in sense of μ and σ. However, we can use *uniqueness* of character's pattern to separate them. Thus, we introduce two features derived from 1-D Savitzky-Golay filter, which is well-known for its low computation time. Here, we use the filter to find the second derivatives of average intensity, calculated in X ($I_{av,x}$) and Y ($I_{av,y}$) direction. This is different from typical derivative filter, such as the Sobel filter, because the Savitzky-Golay filter also performs noise filtering while it finds the derivatives. Therefore, we can adjust the parameters of the filter, e.g. degree and window's width(M), to gain the suitable *texture* characteristics like edges or frequency response of the image blocks.

The equation of the two parameters, namely D_x and D_y are described briefly below.

$$D_x = \sum_{blockwidth} a_2; \tag{2}$$

$$D_y = \sum_{blockheight} a_2; \tag{3}$$

where a_2 is the second coefficient of the Savitzky-Golay filter, which equals to its second derivative. Let $\boldsymbol{f}$ denote input vector $(f_{i-M}, \ldots, f_{i+M})$ at the position i in the I_{av} stream. Then, a_2 can be expressed symbollicaly as

$$a_2 = \{(\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\boldsymbol{f})\}_3 \tag{4}$$

The notation $\{\}_3$ denotes the third element of vector $\boldsymbol{a}$ and $\boldsymbol{f}$ is $(I_{av,x})$ and $(I_{av,y})$ in equations 2 and 3, respectively. $\mathbf{A}$ is a design matrix, which is known *a priori*. Hence, we can compute the coefficients $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ prior to the filtering operation. The computation of $\mathbf{A}$ and its relation to the filter's degree and width are described in [1].

Classifier's Performance. In the learning process, we supervised the system with the training sets that are manually labeled. The output of the process is a decision tree, which minimizes misclassification error of the labeled data set by the Baysian approach. The trees are designed especially for each block size.

Let K be the number of classes. The misclassification error is defined as:

$$Error = \sum_{i}^{K}\sum_{j}^{K} P(\text{assigned class} = i \text{ and true class} = j) \tag{5}$$

3 Boundary Refinement

Block information does not identify large objects such as headings or big pictures, etc. As a result, an image block containing parts of such objects will be classified incorrectly. Also, some times only parts of text occurs in a block and is labeled as graphics. Moreover, it is necessary to identify the image's or the textual column's coordinates to the OCR system. Consequently, the boundary refinement is applied to find the whole objects' boundaries.

- **The First Stride**: Region growing is applied to the label map in order to group blocks with same labels and change maybe-mislabeled blocks. For example, if a text block is enclosed by graphics, it is likely to be graphics. Besides, the graphic blocks close to images are possible to be edges or parts of images.
- **Textual Column Separation**: We calculate the best path that has minimum $\sum_{column} D_x$ less than a pre-defined threshold at 64×64 pixels block size. Next, we calculate the sum of the block's intensity in the path. The column is split at the highest (white) intensity column.
- **Closed Table Extraction**: Here, groups of vertical(high D_x, low D_y) and horizontal (high D_y, low D_x) lines with text within aresconsidered. The lines are tracked to find inclided angles of the tables and rotate the region. We judge whether it is a table by significant peaks of intensity sum in X and Y direction. The table structure can be recognized by the algorithm in [4].

4 Experimental Results

20 Scanned images of pages from magazines, text books and newspapers were used as training images and another 20 images were used as test images. The

images were 8-bit gray scale, 200 dpi and had size of 1500 × 2000 pixels for newspapers and 2200 × 1600 pixels for others. The algorithm was written in Borland Builder C++ 5.0 and run on PentiumII - 500 MHz. The computation time was approximately 2 seconds. The Savitzky-Golay filter has the second degree and 11 pixels width. The approches we compare with are described in [2].

Table 1. Comparision of average misclassification errors in percent

Part I Only	Part I& Part II	Sobel Filter	DCT Bit rate	Wavelet (Haar)
10.40	9.59	13.10	10.15	18.50

We propose the algorithm as a choice instead of those hard-computing approaches. The results shown are comparable with the recent techniques and are a reliable guidance to identify the positions of objects. When apply to Thai OCR, the memory runout is clearly solved. Moreover, the algorithm increases a few recognition rate, because it disposes those parts of graphic that disturb the recognition system. The coordinates of the object, using corner positions of blocks, can provide information to duplicate the document's layout. However, due to limitation in block separation, if text is close to other objects less than 16 pixels, some characters will be lost. Though the result is satisfactory for OCR, the research recently continues on appropriate utilizing the filter's parameters at different degrees or window width. Faster and more effective region refinement is also helpful.

5 Conclusion

A new choice for document segmentation has been proposed in this paper. The algorithm is simple yet effective as a pre-processing of OCR.

References

1. A. Savitzky and M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedure. Analytical Chemistry **36** (1964) 1627–1639
2. I. Keslassy, M. Kalman, D. Wang, and B. Girod, Classification of Compound Images Based on Transform Coefficient Likelihood. Proc. ICIP 2001 (2001)
3. In-Kwon Kim, Dong-Wook Jung and Rae-Hong Park Document image binarization based on topographic analysis using a water flow model. Pattern Recognition, **35** (1) (2002) 265–277
4. Sarin Watcharabusaracum and Wasin Sinthupinyo Unknown Table Image Recognition. Proc. SNLP - Orietal COCOSDA 2002 (2002) 201–204