Document layout structure extraction using bounding boxes of different entities

text: title

J. Liang, J. Ha, R. Rogers, R. M. Haralick

Department of Electrical Engineering, Box 352500, University of Washington, Seattle, WA 98195

I. T. Phillips

Department of Computer Science, Seattle University, Seattle, WA 98112

Abstract

This paper presents an efficient and accurate technique for document page layout structure extraction and classification by analyzing the spatial configuration of the bounding boxes of different entities on a given image. The text, table, and nontext structures are detected on document images. The text-lines and words are extracted and the tabular structure is further decomposed into row and column items. Finally, the document layout hierarchy is produced from these extracted entities.

We develop a performance metric for the document layout analysis by finding the correspondences between detected bounding boxes and ground-truth. We evaluate our algorithms on 1600 images from the UW-III Document Image Database, and the quantitative performance measures in terms of the rates of correct, miss, false, merging, splitting, and spurious detections are reported. We describe a method for determining the optimcal algorithm tuning parameters given the ground-truth. The results show that the average performance of the algorithms is improved by 26.6% after the optimization process.

Key words: document layout structure, bounding box, performance, optimization.

1 Introduction

The goal of document understanding is to convert existing paper documents into a machine readable form. It involves extracting the geometric page layout, recognizing the text content through an Optical Character Recognition (OCR) system, determining the logical structure, and formatting the data and

After finding the search direction, we use the golden section algorithm to find the best scalar k.

4 Experimental results

In this section, we report the performance of our algorithms on the images from the UW-III Document Image Database. The parameters of each algorithm are decided from the optimization process described in Section 3.4. For the segmentation process, the weights we used for computing the performance is given in Table 1. For the classification process, we assign weight 0 when $d = g$

Table 1
Weights assigned to different spatial matching.

Correct	Miss	False	Merge	Split	Spurious
0	1	1	0.5	0.5	1

and assign 1 when $d \neq g$. Therefore, our performance criterion is the cost for converting the detected layout structure to the ground-truth.

For each algorithm, we start with their default parameters obtained by observing a small number of images, then we tune the parameters by minimizing the algorithm's cost on the images in the UW-III database, until the improvement of performance is less than a threshold. The algorithms' cost after the parameter tuning, compared to their performance using the default parameters (reported in [7]), is shown in Table 2. The average improvement is 26.6% with respect to the original performance.

Table 2
Illustrates the performance of algorithms before and after the optimization process.

	Page	Line 1	Line 2	Word	Block
original	0.143	0.022	0.015	0.015	0.141
optimized	0.104	0.017	0.006	0.012	0.137
improved by	27.3%	22.7%	60%	20%	2.8%

table: 1

The numbers and percentages of miss, false, correct, splitting, merging and spurious detections of each algorithm after the optimization process are presented in the following sections.

4.1 Performance of page segmentation

Table 3 illustrates the numbers and percentages of miss, false, correct, splitting, merging and spurious detections with respect to the ground-truth zones as well as the algorithm output. The cost for the page segmentation is 10.2%. Since the page segmentation finds the coarse homogeneous zones, we do not consider the merging of adjacent text within the same column as error.

Table 3
Performance of X-Y cut page segmentation with respect to the ground-truth and the algorithm output

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	24216	21019	462	2186	245	304
table: 2		(86.80%)	(1.91%)	(9.03%)	(1.01%)	(1.25%)
Detected	14848	11346	1883	710	592	317
		(76.41%)	(12.68%)	(4.78%)	(3.99%)	(2.14%)

This algorithm is restricted to bi-directional X-Y cuttable layouts. It is also sensitive to severe page skew. So deskew has to be done before applying the projection. To decide if applying a cut on a projection profile valley, a threshold on the width of valley is used. Instead of having a global value, the threshold might be adaptively determined by considering the width and depth of the projection profile valley, the size of nearby connected components, and the aspect ratio of current zone, etc.

4.2 Performance of text-line segmentation

The numbers and percentages of miss, false, correct, splitting, merging and spurious detections of two text-line extraction algorithms are shown in Table 4.

The cost of the first text-line extraction algorithm (projection profile cut) is 1.73%. The advantages of this algorithm are that it is very simple and fast, and it produces very low misdetection rate. But this algorithm is sensitive to skew (global or local), smear, warping, and noise. If the inter-text-line spacing is very small and the superscript, subscript, or the ascender and descender of adjacent lines overlap or touch with each other, the text-lines are usually merged.

The cost of the second text-line extraction algorithm (merging and splitting connected components) is 0.62%. The splitting procedure is able to split the touched connected components and warped or skewed text-lines, but it may

Table 4
Performance of text-line extraction algorithms: (a) projection profile cut; (b) merging and splitting of connected components.

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	105439	100471	124	4543	157	33
		(95.39%)	(0.12%)	(4.31%)	(0.15%)	(0.03%)
Detected	102494	100471	383	1504	4	132
		(98.03%)	(0.37%)	(1.47%)	(0.00%)	(0.13%)

(a)

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	105439	104250	148	657	335	49
		(98.87%)	(0.14%)	(0.62%)	(0.32%)	(0.05%)
Detected	105107	104250	390	287	1	179
		(99.19%)	(0.37%)	(0.27%)	(0.00%)	(0.17%)

(b)

also cause some splitting errors, i.e. the superscript and subscript are split from the text-line.

4.3 Performance of word extraction

Table 5 illustrates the numbers and percentages of miss, false, correct, splitting, merging and spurious detections with respect to the ground-truth words as well as the algorithm output. The performance of this algorithm is 1.15%.

Table 5
Performance of word extraction.

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	828201	810208	5192	11259	1327	215
		(97.83%)	(0.63%)	(1.35%)	(0.16%)	(0.03%)
Detected	828296	810208	13380	4333	192	183
		(97.82%)	(1.62%)	(0.52%)	(0.02%)	(0.02%)

This algorithm is robust for different layout and conditions since the word spacing is determined adaptively. Small skew and warping are tolerable. One of the limitations of this algorithm is that the text-line is required as input. It is sensitive to noise, italic font, in-line mathematical formula, underlined text, etc. We can find the base line and x-height first, and only use the portion of