Table Detection via Probability Optimization

Yalin Wang[1], Ihsin T. Phillips[2], and Robert M. Haralick[3]

[1] Dept. of Elect. Eng. Univ. of Washington
Seattle, WA 98195, US
ylwang@u.washington.edu
[2] Dept. of Comp. Science, Queens College, City Univ. of New York
Flushing, NY 11367, US
yun@image.cs.qc.edu
[3] The Graduate School, City Univ. Of New York
New York, NY 10016, US
haralick@gc.cuny.edu

Abstract. In this paper, we define the table detection problem as a probability optimization problem. We begin, as we do in our previous algorithm, finding and validating each detected table candidates. We proceed to compute a set of probability measurements for each of the table entities. The computation of the probability measurements takes into consideration tables, table text separators and table neighboring text blocks. Then, an iterative updating method is used to optimize the page segmentation probability to obtain the final result. This new algorithm shows a great improvement over our previous algorithm. The training and testing data set for the algorithm include $1,125$ document pages having 518 table entities and a total of $10,934$ cell entities. Compared with our previous work, it raised the accuracy rate to 95.67% from 90.32% and to 97.05% from 92.04%.

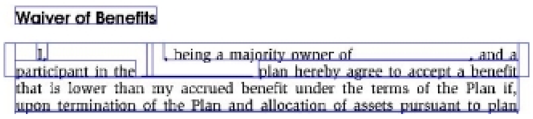
1 Introduction

With the large number of existing documents and the increasing speed in the production of multitude new documents, finding efficient methods to process these documents for their content retrieval and storage becomes critical. For the last three decades, the document image analysis researchers have successfully developed many outstanding methods for character recognition, page segmentation and understand of text-based documents. Most of these methods were not designed to handle documents containing complex objects, such as tables. Tables are compact and efficient for presenting relational information and most of the documents produced today contain various types of tables. Thus, table structure extraction is an important problem in the document layout analysis field. A few table detection algorithms have been published in the recent literature ([1]–[2]). However, the performance of these reported algorithms are not yet good enough for commercial usage.

Among the recently published table detection algorithms, some of them are either using a predefined table layout structures [3][4], or relying on complex heuristics for detecting tables ([5], [6], [7]). Klein et. al. [7] use a signal model to detect tables. Hu et. al. [2] describe an algorithm which detects tables based on computing an optimal partitioning of an input document into some number of tables. They use a dynamic programming technique to solve the optimization problem. Their algorithm works for both ASCII and image documents, and according to their global evaluation protocol, the algorithm yields a recall rate of 83% for 25 ASCII documents and 81% for 25 scanned images, and a precision rate of 91% and at 93%, respectively.

Our early table detection work [1] is based on a background analysis, has a coarse to fine hierarchy structure and is a probability based algorithm. It determines the table candidates by finding the large horizontal blank blocks [8] within the document and then statistically validates if the candidates are table entities. Due to the non-iterative nature of this algorithm, its accuracy was not high enough. Figure 1 shows two of failed examples of the earlier algorithm. Figure 1(a) is a false alarm example (b) is a misdetection example.

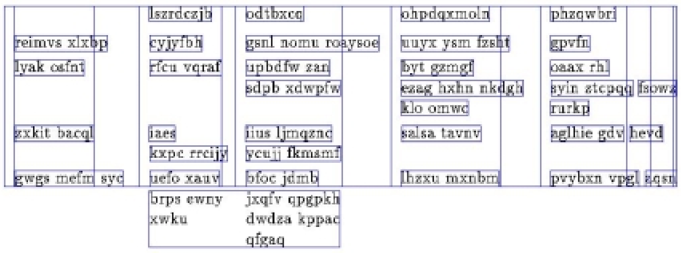
In this paper, we define the table detection problem as a probability optimization problem. We begin, as we did in our previous algorithm, finding and validating each detected table candidates. We proceed to compute a set of probability measurements for each of the table entities. The computation of the probability measurements takes into consideration tables, table text separators and table neighboring text blocks. Then, an iterative updating method is used to optimize the page segmentation probability to obtain the final result. This new algorithm shows a great improvement over our previous algorithm.



Waiver of Benefits

I, ______________ , being a majority owner of ______________ , and a participant in the ______________ plan hereby agree to accept a benefit that is lower than my accrued benefit under the terms of the Plan if, upon termination of the Plan and allocation of assets pursuant to plan

(a)



	lszrdczjb	odtbxcq	ohpdqxmoln	phzqwbri	
reimvs xlxbp	cyjyfbh	gsnl nomu roaysoe	uuyx ysm fzsht	gpvfn	
lyak osfnt	rfcu vqraf	upbdfw zan	byt gzmgf	oaax rhl	
		sdpb xdwpfw	ezag hxhn nkdgh	syin ztcpqq	fsowz
			klo omwc	rurkp	
zxkit bacql	iaes	iius ljmqznc	salsa tavnv	aglhie gdv	hevd
	kxpc rrcijy	ycujj fkmamf			
gwgs mefm syc	uefo xauv	bfoc jdmb	lhzxu mxnbm	pvybxn vpgl	zqsn
	brps ewny xwku	jxqfv qpgpkh dwdza kppac qfgaq			

(b)

Fig. 1. Examples of table detection research of our early research; (a) a false alarm example; (b) a misdetection example.

The remainder of the paper is organized as follows. We give the problem statement in Section 2. The probability estimation details are described in Section 3. We present our algorithm details in Section 4. The experimental results are reported in Section 5 and we conclude with our future directions in Section 6.

2 Table Detection Problem

Let $\mathcal{A}$ be a set of zone entities. Let $\mathcal{L}$ be a set of content labels, $\{$table, text-block$\}$. Function $f : \mathcal{A} \to \mathcal{L}$ assigns each element of $\mathcal{A}$ with a label. Function $V : \wp(\mathcal{A}) \to \Lambda$ computes measurements made on subset of $\mathcal{A}$, where Λ is the measurement space.

We define a probability of labeling and measurement function as

$$P(V(\tau) : \tau \in \mathcal{A}, f|\mathcal{A}) = P(V(\tau) : \tau \in \mathcal{A}, |f, \mathcal{A})P(f|\mathcal{A}) \tag{1}$$

By making the assumption of conditional independence that when the label $f_\tau, \tau \in \mathcal{A}$ is known, no knowledge of other labels will alter the probability of $V(\tau)$, we can decompose the probability in Equation 1 into

$$P(V(\tau) : \tau \in \mathcal{A}|f, \mathcal{A}) = \underbrace{\prod_{\tau\in\mathcal{A}} P(V(\tau)|f, \mathcal{A})}_{(a)} \underbrace{P(f|\mathcal{A})}_{(b)} \tag{2}$$

Expression (a) in Equation 2 can be computed by applying different measurement functions V_{TAB} and V_{TXT} according to f function values, table or text-block, where V_{TAB} is used for tables and V_{TXT} is used for text-blocks.

$$P(V(\tau) : \tau \in \mathcal{A}|f, \mathcal{A}) = \prod_{f_\tau = table}^{\tau\in\mathcal{A}} P(V_{TAB}(\tau)|f, \mathcal{A}) \prod_{f_\tau = text-block}^{\tau\in\mathcal{A}} P(V_{TXT}(\tau)|f, \mathcal{A})P(f|\mathcal{A}) \tag{3}$$

To compute expression (b) in Equation 2, we consider the discontinuity property between neighbors to two zone entities with different labels. Let $\mathcal{A} = \{A_1, A_2, \cdots, A_M\}$ be the set of document elements extracted from a document page. Each element $A_i \in \mathcal{A}$ is represented by a bounding box (x, y, w, h), where (x, y) is the coordinate of top-left corner, and w and h are the width and height of the bounding box respectively. The spatial relations between two adjacent boxes are shown in Figure 2.

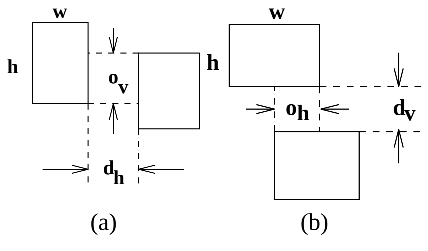


Fig. 2. Illustrates the spatial relations between two bounding boxes that are (a) horizontally adjacent (b) vertically adjacent.

For a pair of bounding boxes $a(x_a, y_a, w_a, h_a)$ and $b(x_b, y_b, w_b, h_b)$, the horizontal distance $d_h(a, b)$ and vertical distance $d_v(a, b)$ between them are defined as

$$d_h(a,b) = \begin{cases} x_b - x_a - w_a & \text{if } x_b > x_a + w_a \\ x_a - x_b - w_b & \text{if } x_a > x_b + w_b \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$d_v(a,b) = \begin{cases} y_b - y_a - h_a & \text{if } y_b > y_a + h_a \\ y_a - y_b - h_b & \text{if } y_a > y_b + h_b \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The horizontal overlap $o_h(a, b)$ and vertical overlap $o_v(a, b)$ between a and b are defined as

$$o_h(a,b) = \begin{cases} x_a + w_a - x_b & \text{if } x_b > x_a, x_b < x_a + w_a \\ x_b + w_b - x_a & \text{if } x_a > x_b, x_a < x_b + w_b \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$o_v(a,b) = \begin{cases} y_a + h_a - y_b & \text{if } y_b > y_a, y_b < y_a + h_a \\ y_b + h_b - y_a & \text{if } y_a > y_b, y_a < y_b + h_b \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Let $A_a = (x_a, y_a, w_a, h_a)$ and $A_b = (x_b, y_b, w_b, h_b)$ be two zone entities.

- We define A_b as a *right neighbor* of A_a if $A_b \neq A_a, x_b > x_a,$ and $o_v(a, b) > 0$. Let B_a be the set of right neighbors of A_a. Zone entities A_a and A_b are called *horizontally adjacent* if

$$A_b = \arg \min_{A_i \in B_a} (d_h(a,i) | x_i > x_a, o_v(a,i) > 0). \tag{8}$$

- We define A_b as a *lower neighbor* of A_a if $A_b \neq A_a, y_b > y_a,$ and $o_h(a, b) > 0$. Let B_a be the set of right neighbors of A_a. Zone entities A_a and A_b are called *vertically adjacent* if

$$A_b = \arg \min_{A_i \in B_a} (d_v(a,i) | y_i > y_a, o_h(a,i) > 0). \tag{9}$$

The neighbor set is defined as

$$\mathcal{N} = \{(v_a, v_b) | v_a \text{ and } v_b \text{ horizontally or vertically adjacent}, v_a \in \mathcal{V}, v_b \in \mathcal{V}\}$$

Assuming the conditional independence between each neighborhood relationship, expression (b) in Equation 2 can be computed as

$$P(f|\mathcal{A}) = \prod_{\{p,q\} \in \mathcal{N}} P_{\{p,q\}}(f_p, f_q | p, q) \tag{10}$$

where $P_{\{p,q\}}(f_p, f_q|p, q)$ has the property

$$P_{\{p,q\}}(f_p, f_q|p,q) = \begin{cases} P_{\{p,q\}}(f_p, f_q|p,q) & f_p \neq f_q \\ 0 & f_p = f_q \end{cases} \tag{11}$$

Equation 3 can be written as

$$P(V(\tau) : \tau \in \mathcal{A}|f, \mathcal{A}) =$$
$$\prod_{f_\tau = table}^{\tau \in \mathcal{A}} P(V_{TAB}(\tau)|f, \mathcal{A}) \prod_{f_\tau = text-block}^{\tau \in \mathcal{A}} P(V_{TXT}(\tau)|f, \mathcal{A}) \prod_{\{p,q\} \in \mathcal{N}} P(f_p, f_q|p, q) \tag{12}$$

The table detection problem can be formulated as follows: *Given initial set $\mathcal{A}^0$, find a new set $\mathcal{A}^s$ and a labeling function $f^s : A^s \to L$, that maximizes the probability:*

$$P(V(\tau) : \tau \in \mathcal{A}^f|f^s, \mathcal{A}^s) =$$
$$\prod_{f^s_\tau = table}^{\tau \in \mathcal{A}^s} P(V_{TAB}(\tau)|f^s, \mathcal{A}^s) \prod_{f^s_\tau = text-block}^{\tau \in \mathcal{A}^s} P(V_{TXT}(\tau)|f^s, \mathcal{A}^s) \prod_{\{p,q\} \in \mathcal{N}} P(f^s_p, f^s_q|p, q) \tag{13}$$

Our goal is to maximize the probability in Equation 12 by iteratively updating $\mathcal{A}^k$ and f^k. Our table detection system works as follows: we used our early research [1] to get preliminary table detection results. Then we systematically adjust the labeling to maximize the probability until no further improvement can be made.

3 Probability Estimation

3.1 Table and Text Separator Probability

Given a table, t and its vertically adjacent neighboring text block B, we compute the probability of the separator between them being a table and text separator as

$$P(f_t, f_B|t, B) = P(TableTextSeparator|o_h(t, B), d_v(t, B))$$

where the definitions of $d_v(t, B)$ and $o_h(t, B)$ can be found at Equation 5 and Equation 6.

3.2 Table Measurement Probability

To facilitate table detection, we applied our table decomposition algorithm [1] on each detected table. Based on the table decomposition results, three features are computed. These features are given below.

– Ratio of total large vertical blank block [8] and large horizontal blank block [8] areas over identified table area. Let t be an identified table and $\mathcal{B}$ be the set of large horizontal and vertical blank blocks and in it, $ra = \frac{\sum_{\beta \in \mathcal{B}} Area(\beta)}{Area(t)}$;
– Maximum difference of the cell baselines in a row. Denote the set of the cells in a row i as $\mathcal{RC}_i$, $\mathcal{RC}_i = \{c_{i,1}, c_{i,2}, ..., c_{i,i_m}\}$. Denote the set of $\mathcal{RC}_i$ as $\mathcal{RC}$, $\mathcal{RC} = \{\mathcal{RC}_i, i = 1, ..., m\}$, where m is the row number in the table. Let $baseline(c)$ be the y coordinate of the cell entity bottom line, $mc = \max_{\mathcal{RC}_i \in \mathcal{RC}} (\max_{c_{i,j} \in \mathcal{RC}_i} (baseline(c_{i,j})) - \min_{c_{i,j} \in \mathcal{RC}_i} (baseline(c_{i,j})))$;

– Accumulated difference of the justification in all columns. Denote the set of cells in a column, i, in the table $\mathcal{CC}_i = \{c_{i,1}, c_{i,2}, ..., c_{i,i_n}\}$. Denote the set of $\mathcal{CC}_i$ as $\mathcal{CC}$, $\mathcal{CC} = \{\mathcal{CC}_i, i = 1, ..., n\}$, where n is the column number in the table. Let $x_{i,j}, y_{i,j}, w_{i,j}, h_{i,j}$ represent the bounding box of the cell $c_{i,j} \in \mathcal{CC}_i$. We estimate the justification of a column, $i, i = 1, ..., n$, by computing the vertical projection of the left, center, and right edge of $c_{i,j}, j = 1, ..., i_n$,

$$\begin{aligned} C_{left}[i] &= max_{c_{i,j} \in \mathcal{CC}_i}(x_{i,j}) - min_{c_{i,j} \in \mathcal{CC}_i}(x_{i,j}) \\ C_{center}[i] &= max_{c_{i,j} \in \mathcal{CC}_i}(x_{i,j} + w_{i,j}/2) - min_{c_{i,j} \in \mathcal{CC}_i}(x_{i,j} + w_{i,j}/2) \\ C_{right}[i] &= max_{c_{i,j} \in \mathcal{CC}_i}(x_{i,j} + w_{i,j}) - min_{c_{i,j} \in \mathcal{CC}_i}(x_{i,j} + w_{i,j}) \\ J_i &= min\{C_{left}[i], C_{center}[i], C_{right}[i]\} \end{aligned}$$

The accumulated difference of the justification in all columns, mj, is computed as: $mj = \sum_{i=1}^{n} J_i$.

Finally, we can compute the table consistent probability for table t as

$$P(V_{TAB}(t)) = P(consistency(t)|ra(t), mc(t), mj(t))$$

3.3 Text Block Measurement Probability

A text block, in general, has a homogeneous inter-line spacing and an alignment type (such as left-justisfied, etc.) Given a detected text block B, we compute the probability that B has homogeneous inter-line spacing, and a text alignment type. We define leading as inter-line spacing. As in Liang et. al.' [9], we compute text block measurement probability as

$$P(V_{TXT}(B)) = P(V_{TXT}(B)|Leading(B), Alignment(B))$$

By making the assumption of conditional independence, we can rewrite the above equation as

$$P(V_{TXT}(B)) = P(V_{TXT}(B)|Leading(B))P(V_{TXT}(B)|Alignment(B))$$

Let $B = (l_1, ..., l_n)$ be an extracted block. $D_B = (d(1), d(2), ..., d(n-1))$ is a sequence of inter-line space distances, where $d(j)$ is the space distance between l_j and l_{j+1}. We compute the median and the maximum value of the elements of D_B. The probability is

$$P(V_{TXT}(B)|Leading(B)) = P(median(D_B), max(D_B)|Leading(B)) .$$

Given a text block B that consists of a group of text lines $B = (l_1, l_2, \cdots, l_n)$, we determine the text alignment of B by observing the alignment of the text line edges. Let e_{li} be the left edge of the text line l_i and let e_{ci} and e_{ri} be the center and right edges of the line box respectively. Let E_l be the left edges of text line 2 to n, such that $E_l = \{e_{li}|2 \leq i \leq n\}$. E_c is the center edges of text line 2 to $n-1$, and E_r is the right edges of text line 1 to $n-1$. We first estimate the median of E_l, then compute the absolute deviation D_l of the elements of E_l from its median,

$$D_l = \{d_i | d_i = |e_{li} - \text{median}(E_l)|, 2 \leq i \leq n\}.$$

Similarly, we estimate the absolute deviation of the center edges and right edges: D_c and D_r. Then, we compute the probability of B being left, center, right, or both justified by observing the mean absolute deviation of the left, center and right edges,

$$P(V_{TXT}(B)|Alignment(B)) = P(\text{mean}(D_l), \text{mean}(D_c), \text{mean}(D_r)|Alignment(B)). \tag{14}$$

4 Table Detection Algorithm

Figure 3 shows our algorithm diagram. Given a labeled page, first we estimate its segmentation probability. For each table, we consider several adjustments, which are to keep it as a table, to grow the table to include its upper and lower neighbors, to merge the table with its upper and lower neighbors and label it as text block. For each adjustment, we compute the new probability. We select the adjustment which produces the biggest improvement upon the initial page segmentation probability. This process is repeated until no improvement can be made. The details of the algorithm are described below.

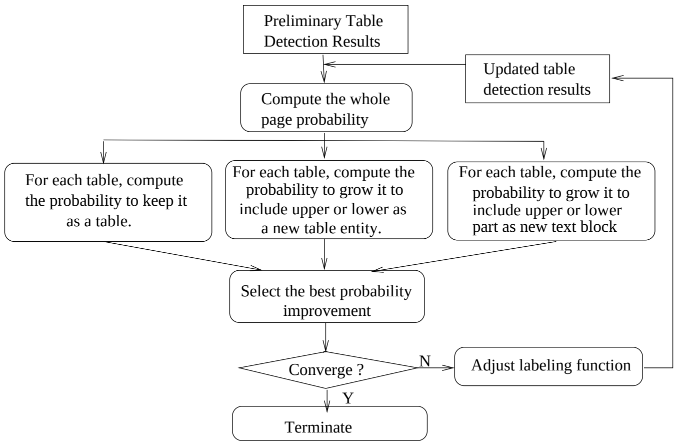

Preliminary Table Detection Results
Updated table detection results
Compute the whole page probability
For each table, compute the probability to keep it as a table.
For each table, compute the probability to grow it to include upper or lower as a new table entity.
For each table, compute the probability to grow it to include upper or lower part as new text block
Select the best probability improvement
Converge ?
N
Adjust labeling function
Y
Terminate

Fig. 3. Overview of the table detection algorithm

Algorithm 41 *Table Optimization*

1. The input data to the algorithm are our previous table detection [1] and text block segmentation results. They are a set of block entities, $\mathcal{A}^0$ and function $f^0 : \mathcal{A}^0 \rightarrow \mathcal{L}$;

2. Set $k = 0$;
3. For each hypothesized table, $i, i = 1, ..., N$, where N is the number of tables in $\mathcal{A}^k$. Compute the different probabilities under different adjustments.
   – Keep the table. Compute the probability $P_{(i,1)}$ following Equation 12;
   – Merge table i with its upper text neighbor and label it as a new table. Compute the new probability $P_{(i,2)}$ following Equation 12;
   – Merge table i with its upper text neighbor and label it as a new text block. Compute the new probability $P_{(i,3)}$ following Equation 12;
   – Merge table i with its lower text neighbor and label it as a new table. Compute the new probability $P_{(i,4)}$ following Equation 12;
   – Merge table i with its lower text neighbor and label it as a new text block. Compute the new probability $P_{(i,5)}$ following Equation 12.
4. Compute $P_{max} = max(P_{(i,j)}), i = 1, ..., N, j = 1, ..., 5$ and get its appropriate adjustment *action*.
5. If the *action* is to keep the table, then, return the labeling result $\mathcal{A}^k$ as $\mathcal{A}^s$ and stop the algorithm.
6. If the *action* is not to keep the table, then take the adjustment *action* and we get $\mathcal{A}^{k+1}$ and $f^{k+1} : \mathcal{A}^{k+1} \to \mathcal{L}$.
7. Set $k = k + 1$ and go back to 3.

5 Experimental Results

Our test data set has $1,125$ document pages [1]. All of them are machine printed, noise free data. Among them, 565 pages are real data from different business and law books. Another 560 pages are synthetic data generated using the method described in [1]. A hold-out cross validation experiment [10] was conducted on all the data with $N = 9$. Discrete lookup tables were used to represent the estimated joint and conditional probabilities used at each of the algorithm decision steps.

Suppose we are given two sets $\mathcal{G} = \{G_1, G_2, ..., G_M\}$ for ground-truthed foreground table related entities, e.g. cell entities, and $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ for detected table related entities. The algorithm performance evaluation can be done by solving the correspondence problem between the two sets. Performance metrics developed in [11] can be directly computed in each rectangular layout structure set. The performance evaluation was done on the cell level. The numbers and percentages of miss, false, correct, splitting, merging and spurious detections on real data set and on the whole data set are shown in Table 1 and Table 2, respectively. Compared with our early work [1], we improved the detection result rates from 90.32% to 95.67% and from 92.04% to 97.05% in the whole data set. For the real data set, we improved the detection result rates from 89.69% to 96.76% and from 93.12% to 93.86%.

Figure 4 shows a few table detection examples. Figure 4(a), (b) are the correct detection results of Figure 1(a), (b), respectively. In Figure 4(a), our algorithm grows the original table and include its lower neighbor and construct a new table entity. In Figure 4(b), our algorithm eliminates the originally detected table and merge it with its lower neighbor and construct a new text block entity. Figure 4(c), (d) illustrate some failed

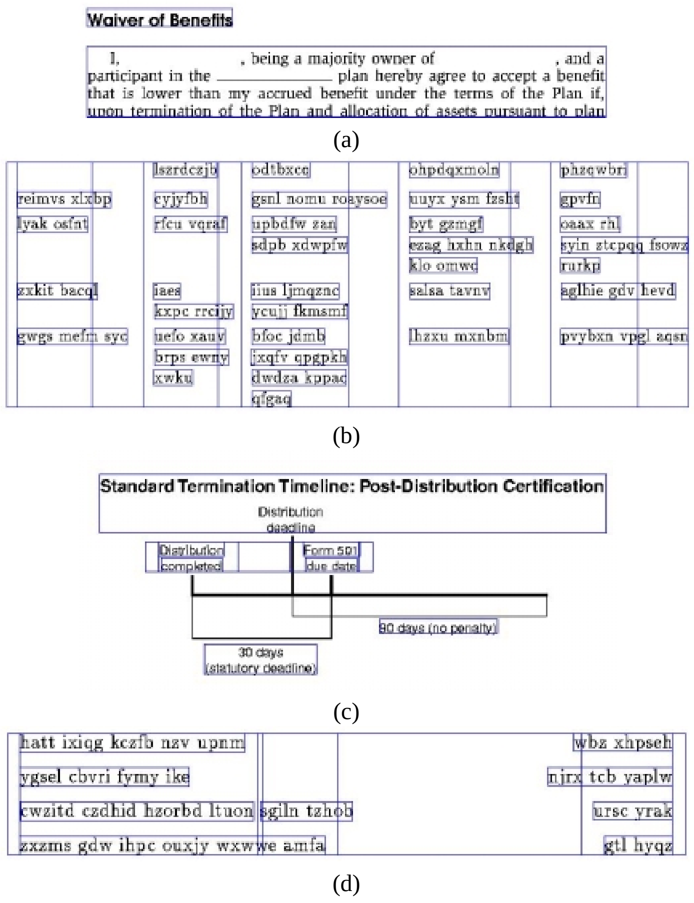

Waiver of Benefits
I, , being a majority owner of , and a participant in the plan hereby agree to accept a benefit that is lower than my accrued benefit under the terms of the Plan if, upon termination of the Plan and allocation of assets pursuant to plan
(a)
(b)
Standard Termination Timeline: Post-Distribution Certification
Distribution deadline
Distribution completed
Form 591 due date
90 days (no penalty)
30 days (statutory deadline)
(c)
(d)

Fig. 4. Illustrates the table detection results; (a), (b) Correct table detection results; (c), (d) failed table detection results.

examples. Figure 4(c) shows a false alarm example. Some texts in a figure are detected as a table entity. Figure 4(d) shows an error example where our table decomposition algorithm failed.

6 Conclusion and Future Work

In this paper, we formulated table detection problem in the whole page segmentation framework. We tried to improve table detection result by optimizing the whole page segmentation probability, including table entities, text block entities and the separators between them. We used iterative updating method to improve the probability. We implemented our algorithm and tested on a data set which includes $1,125$ document pages with $10,934$ table cell entities. Among them, 565 pages are real data from differ-

Table 1. Cell level performance of the table detection algorithm on real data set.

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	679	657 (96.76%)	3 (0.44%)	15 (2.21%)	4 (0.59%)	0 (0.00%)
Detected	700	657 (93.86%)	6 (0.86%)	7 (1.00%)	30 (4.29%)	0 (0.00%)

Table 2. Cell level performance of the table detection algorithm on the whole data set.

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	10934	10461 (95.67%)	132 (1.21%)	45 (0.41%)	296 (2.71%)	0 (0.00%)
Detected	10779	10461 (97.05%)	264 (2.45%)	18 (0.17%)	36 (0.33%)	0 (0.00%)

ent business and law books. Another 560 pages are synthetic data generated using the method described in [1]. The experimental results demonstrated the improvement of our algorithm.

As shown in Figure 4(d), our current table decomposition algorithm needs further refined, incorporating some additional information carrying features. Our formulation has the potential to be useful to other more general page segmentation problems, for example, to segment text, figure and images, etc. We will study this in the future.

References

1. Y. Wang, I. T. Phillips, and R. Haralick. Automatic table ground truth generation and a background-analysis-based table structure extraction method. In *Sixth International Conference on Document Analysis and Recognition(ICDAR01)*, pages 528–532, Seattle, WA, September 2001.
2. J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *SPIE Document Recognition and Retrieval VII*, pages 291–302, San Jose, California, January 2000.
3. E. Green and M. Krishnamoorthy. Model-based analysis of printed tables. In *Proceedings of the 3rd ICDAR*, pages 214–217, Canada, August 1995.
4. J. H. Shamilian, H. S. Baird, and T. L. Wood. A retargetable table reader. In *Proceedings of the 4th ICDAR*, pages 158–163, Germany, August 1997.
5. T. G. Kieninger. Table structure recognition based on robust block segmentation. *Document Recognition V.*, pages 22–32, January 1998.
6. T. Kieninger and A. Dengel. Applying the t-rec table recognition system to the business letter domain. In *Sixth International Conference on Document Analysis and Recognition(ICDAR01)*, pages 518–522, Seattle, WA, September 2001.
7. B. Klein, S. Gokkus, T. Kieninger, and A. Dengel. Three approaches to "industrial" table spotting. In *Sixth International Conference on Document Analysis and Recognition(ICDAR01)*, pages 513–517, Seattle, WA, September 2001.

8. Y. Wang, R. Haralick, and I. T. Phillips. Improvement of zone content classification by using background analysis. In *Fourth IAPR International Workshop on Document Analysis Systems. (DAS2000)*, Rio de Janeiro, Brazil, December 2000.
9. J. Liang, I. T. Phillips, and R. M. Haralick. Consistent partition and labeling of text blocks. *Journal of Pattern Analysis and Applications*, 3:196–208, 2000.
10. R. Haralick and L. Shapiro. *Computer and Robot Vision*, volume 1. Addison Wesley, 1997.
11. J. Liang. *Document Structure Analysis and Performance Evaluation*. Ph.D thesis, Univ. of Washington, Seattle, WA, 1999.