A Method of Evaluating Table Segmentation Results Based on A Table Image Ground Truther

Yanhui Liang[1], Yizhou Wang[1], Eric Saund[2]

[1]School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[2] Palo Alto Research Center, Palo Alto, CA, USA
{yanhuiliang,Yizhou.Wang}@pku.edu.cn, saund@parc.com

***Abstract*—In this paper, we propose a novel method to evaluate table segmentation results based on a table image ground truther. In the ground-truthing process, we first extract connected components from a given table image and connect them into an atom graph with weighed edges. The edge weight is computed by taking the connected component size cohesion and their spatial distance into consideration. Then the ground truther semi-automatically decides the locations and spans of the row/column separators according to the projection profiles with human interaction. We evaluate a given table segmentation by computing the edit distance from its row and column separator setting to that of the ground truth. The edit distance is the sum of all the edit operation costs that correct the wrong row and column separators. Each edit operation cost is a function of the sum of the weights of the edges that the separator cuts through. The experimental results demonstrate that the proposed evaluation method is not only efficient, but also competent to reveal the quality of different segmentations.**

***Keywords*-Table Segmentation; Evaluation; Edit Distance; Ground Truther;**

I. Introduction

As a ubiquitous form in document images, tables play an important role in representation, transfer and comparison of structured information. With the sophistication of document analysis systems growing recently [1][2], more and more table processing algorithms and systems are introduced [3][4][5]. Thus, an efficient and accurate evaluation method is highly desired.

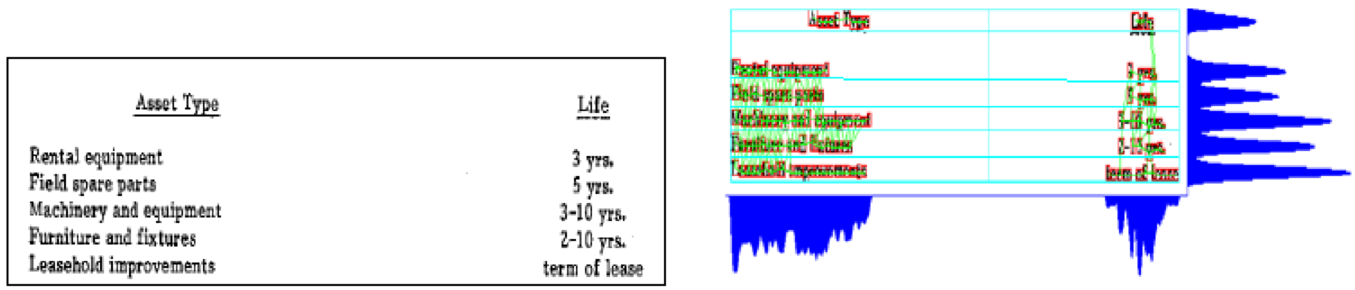
In Recent years, several evaluation techniques of table processing results have been proposed [6][7]. Hu et al. [6] presented an evaluation system that represents tables, both the processing result and ground truth, as directed acyclic attribute graphs. Then it posed a series of queries and compared the responses of the two graphs. The essential limitation of this method is that attributed graph matching is difficult and error-prone. It may lead to the worst-case exponential running time for its solution [8]. Also, its evaluation measure does not reflect the committed error types since its evaluation criterion is expressed in terms of the number of correct answers for all probes.

Since more and more algorithms, as discussed by Embley et al. [9], just aim at converting the physical structure of table images for editable Microsoft Excels or Word tables while concentrate little on tagging logical labels to give the semantic interpretation of tables, we introduce an efficient evaluation method that focuses on the physical structure analysis of tables. Based on a ground truther we developed, the proposed method adopts an edit distance as the quantitative evaluation. In the ground-truthing process, we first extract atoms (connected components) from a given table image and connected them into an atom graph with weighed edges. The weight, which indicates the binding force between atoms, is computed by taking the atoms size cohesion and their spatial distance into consideration. Then we decide the locations and spans of the row and column separators semi-automatically according to horizontal and vertical projection profiles and correct segmentation errors interactively. Each separator is assigned a weight which sums all the weights of the edges it cuts through. Next, we assess a given segmentation result by computing the edit distance from its row and column separator setting to that of the ground truth. We identify three error types of separators in the segmentation result, that is, missing separator, spurious separator and redundant separator. The cost function of error-corrected operation is defined by considering the weight of the edited separator. Finally, accumulating all the edit operation costs, the edit distance is returned as the evaluation result. The experimental results demonstrate that the proposed evaluation method is not only efficient, but also competent to reveal the quality of different segmentations.

The remainder of this paper is organized as follows. We describe our evaluation method in detail in Section II. In Section III, a ground-truthing system is introduced as the base of the evaluation algorithm. Section IV presents several examples of applying our evaluation method to table segmentation results. The conclusions and discussion are given in Section V.

II. The evaluation method

In this section, we present our approach to evaluate table segmentation results given the ground truth. Our method adopts an edit distance measure which can locate and recognize the error types appeared in the segmentation result as well as give the quantity measurement for each incorrect



(a) (b)

Figure 1. (a) The original table image. (b) The ground truth.

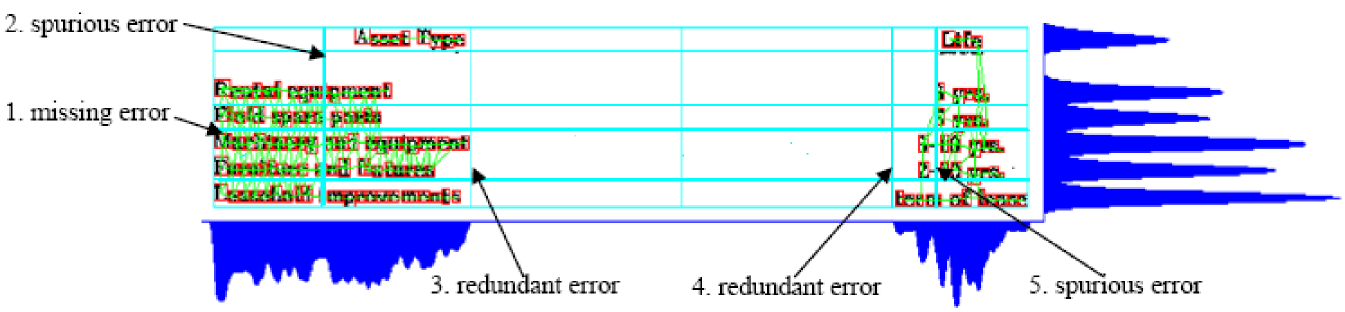


Figure 2. A segmentation result with various error types (errors are shown in format ”id.error type”). The result is generated according to the vertical and horizontal projection profiles of the table.

row/column separator. It returns the edit distance of the segmentation to the ground truth as the evaluation measure.

A. *Edit distance measure*

Given a table segmentation result, first we identify its row/column separators. To do this, we extract connected components from the table image and judge whether a connected component is a row separator or not according to the width and the ratio of the height to the width of its bounding box. If its width is smaller than a threshold (3 pixels in our algorithm) and the ratio is bigger than a threshold (10 pixels we set), we recognize it as a row separator. We locate the column separators in a similar way by exchanging the width and the height.

Next we match the given segmentation result with the ground truth by finding the correspondence between the detected separators and the ground-truth separators. If the location and span of a detected separator match completely with its corresponding ground-truth separator, we say they are perfectly corresponded. Otherwise, segmentation error may happen. We classify the segmentation errors of separators into three categories: missing separator, spurious separator and redundant separator. As depicted in Figure 2, missing separator error (1.missing error) happens when no separator in the segmentation result matches its counterpart in the ground truth. Spurious separator (2.spurious error and 5.spurious error) is the one who makes a wrong cut of the table structure. Redundant separator (3.redundant error and 4.reduntant error) is an extra separator that appears in one whitespace of the table image. By executing this matching procedure, we can identify all the errors in the segmentation result.

To compute the edit distance of the segmentation result to the ground truth, we define cost functions to charge the error correction operations. Then we obtain the edit distance by summing all edit operation costs. Suppose D is the edit distance and G is atom graph.s_m, M, s_s, S and s_r, R indicate a missing separator, the set of missing separators, a spurious separator, the set of spurious separators, a redundant separator, and the set of redundant separators, respectively. The cost functions are $c_m(G, s_m)$, the cost of correcting a missing separator; $c_s(G, s_s)$, the cost of correcting a spurious separator; $c_r(G, s_r)$, the cost of correcting a redundant separator. Then D can be formulated as:

$$D = \sum_{s_m \in M} c_m(G, s_m) + \sum_{s_s \in S} c_s(G, s_s) + \sum_{s_r \in R} c_r(G, s_r) \tag{1}$$

In most of the literature, the cost of each edit operation is always set to a constant. Though it is very simple and easy to implement, it cannot reflect the expense of correcting different error separators. In our method, we define the cost function by taking the edge-cutting weight of the edited separators into consideration. Thus we can formulate the cost functions as follows:

$$c_m(G, s_m) = \frac{\omega_{max} - \omega_{s_m}}{\omega_{max}} \quad (2)$$

$$c_s(G, s_s) = \frac{\omega_{s_s}}{\omega_{max}} \quad (3)$$

$$c_r(G, s_r) = \frac{\omega_{max} - \omega_{s_r}}{\omega_{max}} \quad (4)$$

where ω_{s_m}, ω_{s_s} and ω_{s_r} indicate the edge-cutting weights (see subsection 2.2 for details) of s_m,s_s and s_r, respectively. ω_{max} is set as the maximum edge-cutting weight of row lines of the table image if s_m, s_s and s_r are edited row separators. Otherwise, it is set as the maximum edge-cutting weight of column lines of the table image. We will describe these four items in detail in the following subsections.

For tables with simple grid structure, we execute the presented algorithm and return the edit distance as the evaluation result. Specially, for tables with nested structure, we evaluate the segmentation in a coarse-to-fine recursive manner. We first detect row and column separators that cut through the entire table and compute the costs of the involved edit operations. Then we go to the super-cells which contains a smaller table structure, to continue the evaluation process. We repeat the above procedures until all the super-cells are evaluated. We finally sum all the editing costs and return the edit distance.

Since the proposed method performs the evaluation by examining all the separators in both the given segmentation result and the ground truth, the complexity of our algorithm is $O(n^2)$, where n is the number of separators appeared in the segmentation result and the ground truth.

After executing our algorithm, we can identify exactly the error types of row/column separators in the segmentation result as well as count the number of each error type. Furthermore, the quantity measurement of each segmentation error is expressed by the cost it is charged when corrected. Thus we can not only give the global edit distance of the segmentation result to the ground truth but also tell the detail of the evaluation result. This provides more information about the performance of the segmentation result and makes the subsequent analysis of table more convenient and efficient.

B. Separator edge-cutting weight

To compute the edge-cutting weight of a row/column separator, we first detect the connected components (atoms) of the table image and build an atom graph out of them by applying a Voronoi-like algorithm [10]. The atom graph G is denoted as: $G =< V, E >$.

The vertices

$$V = \{a_i = ((x_i, y_i), h_i, w_i), i = 1, ..., N\} \quad (5)$$

in which (x_i, y_i) is the centroid coordinate of the atom a_i; h_i and w_i are a_i's bounding box height and width, respectively; N is the number of the atoms in the table image.

The neighborhood structure is specified by the edge set

$$E = (e_{ij} : a_i, a_j \in V) \quad (6)$$

where e_{ij} is the edge connecting atoms a_i and a_j.

Each edge is assigned to a weight, $w(e_{ij})$, which tells the binding force between the pair of neighboring atoms. It is determined by the following factors:

1. The spacial distance of the pair of atoms, δ_{ij}.

2. Table image projection profiles, π. If the bin height of the projection profile at the edge location is π_{ij} and the global maximum bin height is π_{max}, the weight due to this factor can be defined as π_{ij}/π_{max}.

3. Size cohesion of the pair of atoms, ε_{ij}, which is defined as $|h_i - h_j| \,/\, |w_i - w_j|$.

Thus, the edge weight can be expressed as

$$w(e_{ij}) = \lambda_0 \exp\{-\delta_{ij}\} + \lambda_1 \pi_{ij}/\pi_{max} + \lambda_2 \exp\{-\varepsilon_{ij}\} \quad (7)$$

$$\sum_k \lambda_k = 1, k = 0, 1, 2 \quad (8)$$

where λ_k is the weight to balance the different cues. It can be simply set to equal.

The cutting-edge weight of a row/column separator is computed based on the weight of the edges it cuts through. It can be computed as:

$$\omega_s = \sum_{e_{ij} \in E} w(e_{ij}) \quad (9)$$

where E is the set contains all the weighed edges that separator s cuts through.

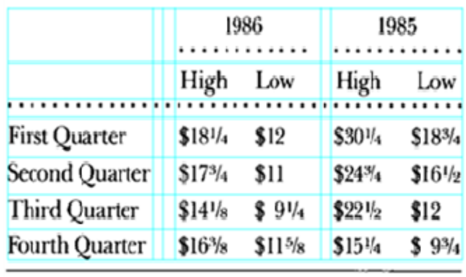
To find ω_{max}, we use the horizontal and vertical projection profile as a search clue. We observe that the line with maximum edge-cutting weight always locates nearby the site of the maximum bin height of the projection profiles. So we search for the maximum edge-cutting weight for the row/column at the locations of the peak of horizontal/ vertical projection profiles, half character horizontally or vertically away from the peak.

Thus, we can calculate the edge-cutting weight for each error separator shown in Figure 2 by setting each λ_k equally for every weighed edge that the separator cuts through. Table I shows the result.

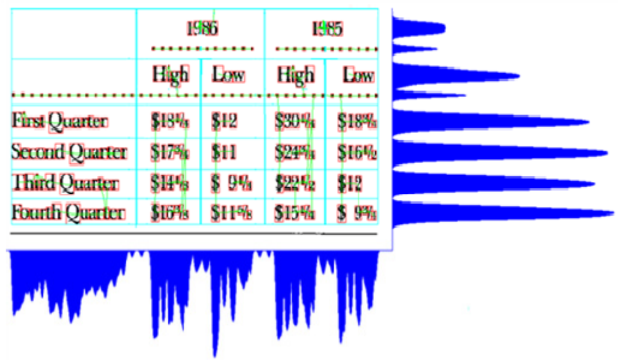
Table I
EDGE-CUTTING WEIGHT FOR EACH UNCORRECT SEPARATOR (SHOWN IN FIGURE 2).

Separator ID	1	2	3	4	5
Edge-cutting weight	4.539	3.231	0	0	2.914

We then locate lines with the maximum edge-cutting weight for row and column. The maximum edge-cutting



(a) The segmentation result.



(b) The ground truth.

Figure 3. Table segmentation result and its corresponding ground truth

weights for the row and column returned by our system are 5.414 and 4.036, respectively. Then we obtain that the edit distance of the segmentation result in Figure 2 from its ground truth in Figure 1(b) is 3.684.

III. Semi-automatic ground truther

To ground-truth the physical structure segmentation of tables, we have developed a semi-automatic ground truther. If there exist row and column separators in the table image, we directly adopt the detected separator as ground truth. Otherwise, the ground truther first decides the locations of the row/column separators automatically according to the horizontal and vertical projection profiles of table image then if any conspicuous segmentation errors happen, we correct them manually.

We adopt two stages to segment tables physical structure. The first is table image pre-processing step. In this stage, our ground truther first bianarizes the table image then applies standard technique to extract connected components and use a Voronoi-like algorithm [10] to connect the extracted atoms into an atom graph.

The next stage is structure segmentation. The table is segmented automatically based on the results of its horizontal and vertical projection profiles. We define the locations of the row/column separators as the valleys of table images horizontal-vertical projection profiles and compute the span of each separator by locating its starting and ending points. Then the conspicuous segmentation errors are corrected manually based on the interactive interface if necessary.

Figure 1 shows a table image and its physical structure segmentation result produced by our ground truther. In 1(b), the small red rectangles around characters are the bounding boxes of the atoms and the connections between them in green color are weighed edges of the atom graph. Projection profiles of this table image are shown diagrammatically as the dark blue graphs outside of the table. The light blue cut-lines locate the row and column separators of the table.

IV. Experimental results

To demonstrate the effectiveness of our method for evaluating table segmentation result, we build a dataset containing 360 tables from various document images. An example is shown in Figure 3.

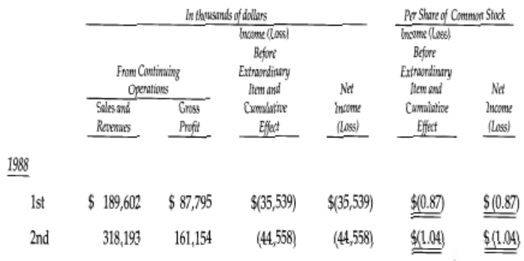
Figure 3(a) shows a segmentation result of a nested table which is generated by its horizontal and vertical projection profiles with a different threshold. We obtain its corresponding ground truth from our ground truther as shown in Figure 3(b). By executing the evaluation algorithm, one row separator and three column separators are found redundant and two column separators are missed. So we exercise an editing sequence to transform the segmentation result into the ground truth. The maximum edge-cutting weight of the row is 3.736 and 2.978 for the column. By summing the weights of edges the edited separator cuts through, We calculate the edge-cutting weights for each edited separator, i.e. 0, 0, 0, 0.278, 0 and 0.334, respectively. We then compute the edit distance 7.794.

From the edit distance and the edge-cutting weight of each edited separator we can see, the three redundant errors in the segmentation result cause bad splitting of the columns and lead to misunderstanding of the structure as well as the content of tables.

In Figure 4, we display other table segmentation results which are generated using the method as above. Then we calculate their corresponding edit distance under our evaluation method. The first segmentation result is much closer to the ground truth than the second one so its edit distance is 3.812, much smaller than the edit distance of the second segmentation result.

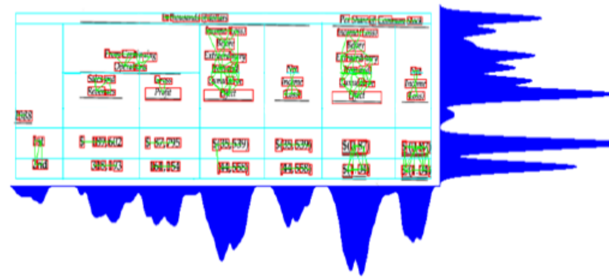
V. Conclusion

In this paper, we presented a method of evaluating table segmentation results based on a table image ground truther. Not only is the method able to recognize the errors types appeared in segmentations, but also it returns the edit

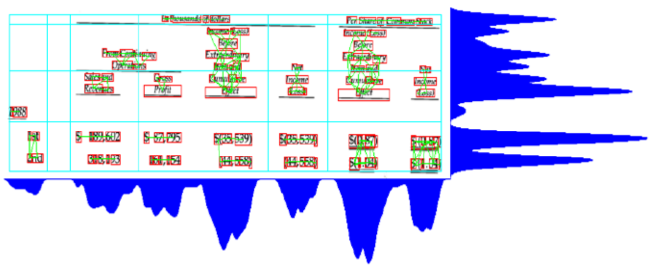


	In thousands of dollars				Per Share of Common Stock	
	From Continuing Operations					
	Sales and Revenues	Gross Profit	Income (Loss) Before Extraordinary Item and Cumulative Effect	Net Income (Loss)	Income (Loss) Before Extraordinary Item and Cumulative Effect	Net Income (Loss)
1988						
1st	$ 189,602	$ 87,795	$(35,539)	$(35,539)	$(0.87)	$ (0.87)
2nd	318,193	161,154	(44,558)	(44,558)	$(1.04)	$ (1.04)

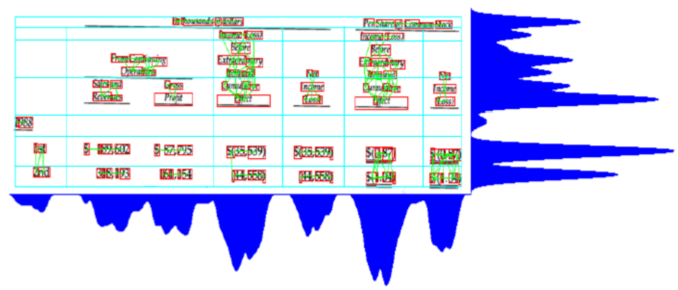
(a) The original table image.



(b) The ground truth.



(c) 1st parsing result with edit distance 3.812.



(d) 2nd parsing result with edit distance 8.473.

Figure 4. (a) An original table image. (b) The ground truth returned by our ground truther. (c) One segmentation result of the table image. (d) Another segmentation result.

distances as the evaluation of the quality of different segmentation results. We collect a dataset containing hundreds of table images to validate the performance of our evaluation method. The presented experimental results demonstrate its efficacy and accuracy.

There are few issues need further discussion. The first is the atom cohesion. By adopting advanced OCR engine, we can incorporating more features into the cohesion measure, e.g. font type, boldness, italics, digit vs. text.

The second is the recognition of redundant separators. If there are multiple separators in one wide whitespace of the table image, which one should be selected as the right separator? Now we choose the one which is closest to its corresponding separator in the ground truth. Besides, if the valley of the projection profiles is wide enough, should we edit the separator which is far away from its corresponding ground truth? Or just consider it as a right separator?

In the future, we will further study the above-mentioned issues. Besides, more rigorous studies of the correlation of our evaluation measures with humans perception of the quality of the table segmentation results are to be researched.

REFERENCES

[1] K.Itonori. Table structure recognition based on textblock arrangement and ruled line position. Proc. Second Intl Conf. Document Analysis and Recognition, pp.765-768, Tsukuba Science City, Japan, 1993.

[2] W.Kornfeld and J. Wattecamps. Automatically locating, extracting and analyzing tabular data. Proc. Twenty-first Intl ACM SIGIR Conf. Research and Development in Information Retrieval, pp.347-348, Melbourne, Australia, 1998.

[3] D. Lopresti and G. Nagy. Automated table processing: An (opinionated) survey. In Proceedings of the Third IAPR International Workshop on Graphics Recognition, pp.109-134, Jaipur, India, September 1999.

[4] C. Peterman, C. H. Chang, and H. Alam. A system for table understanding. In Proceedings of the Symposium on Document Image Understanding Technology, pp. 55-62, Annapolis, MD, 1997.

[5] K.Zuyev. Table image segmentation. Proc. Fourth Intl Conf. Document Analysis and Recognition, pp.705-708, Ulm, Germany, 1997.

[6] J. Hu, R. Kashi, D. Lopresti, G. Wilfong. Evaluating the performance of table processing algorithms, International Journal on Document Analysis and Recognition, 4(3): 140-153, 2002.

[7] F. Kboubi, A. H. Chabi, and M. B. Ahmed. Table recognition evaluation and combination method. In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR), pp. 1237C1241, Seoul, Korea, 2005.

[8] Steven S. Skiena. Geometric Probing. Science, Urbana, IL ,1988

[9] D.W. Embley, M. Hurst, D. Lopresti, and G. Nagy. Table processing paradigms: A research survey. International Journal of Document Analysis and Recognition, 8(2-3):66-86, June 2006.

[10] K. Kise, A. Sato, and M. Iwata, Segmentation of page images using the area Voronoi diagram, CVIU, 1998.