Random Table and Its Ground Truth Automatic Generation: A Tool for Table Understanding Research

Yalin Wang† Ihsin T. Phillips‡ Robert Haralick†

† Department of Electrical Engineering
University of Washington Seattle, WA 98195 U.S.A.

‡ Department of Computer Science, Queens College
CUNY Flushing, NY 11367 U.S.A.

{ylwang@u.washington.edu}

Abstract

We developed a software tool to assist table understanding research. It can analyze any given table ground truth and generate documents that include similar table elements while have more variety on both table and non-table parts. Based on our novel content matching ground truthing idea, the table ground truth data for the generated table elements become available with little manual work. The validity of the proposed strategy was confirmed by our table detection algorithm development. We made this software package publicly available.

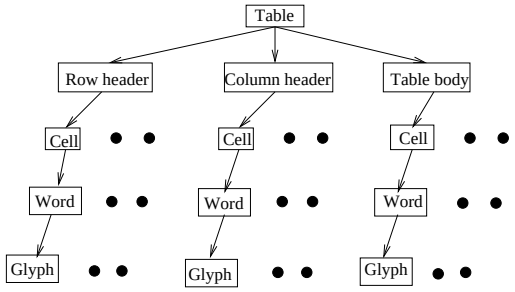


Figure 1. Illustrates a table hierarchy model

1 Introduction

Although there are constant interests on table understanding problem ([1]-[6]), there are no publicly available table ground truth data sets. In UW document image database III(UW CDROM III) [9], there are 215 marked table zones but no structure data for the them. Detailed Table structure information is required for a table detection system evaluation [7]. Clearly, UW CDROM III cannot be directly used to evaluate table detection systems.

Nonetheless, large data sets with ground truth are essential in assessing the performance of computer vision algorithms. Manually generating document ground truth proved to be very costly. However, studying synthetic data [8] at some research phase is a common practice in computer vision field. It has the advantage of extremely low cost, readily available ground truth information and more variety than the real images.

We developed a software package that can simulate any given table ground truth with additional controlled variety. To avoid the tedious manual ground truthing work, we made the table content unique in the given document image and therefore table structure can be determined by content matching. We demonstrated the feasibility of our algorithm on a real image data set and used the synthetic table data to aid our table detection research [7]. The software package is publicly available at [16]. Although it was designed for table understanding research, its potential usages include tabular data reconfiguration(e.g. from business style to technical style) and transformation.

The remainder of paper is organized as follows. First, we give our table ground truthing specification in Section 2. The detailed automatic table ground truth generation algorithm is described in Section 3. Our future work direction is given in Section 4. We put our table parameter set and non-table parameter set definitions in APPENDIX A and B.

2 Table Ground Truth Specification

We defined the table structure in a hierarchical structure, as shown in Figure 1. In the table ground truth, we need specify their hierarchical structure between table,

row/column header, table body and cell entities. For each cell, the following attributes have to be recorded.

- Starting/ending row, *sr* and *er*;
- Starting/ending column, *sc* and *ec*;
- Justification, *cj*. Its possible values are left, center, right and decimal.

Note that although we do not explicitly describe row and column structures, such information can be readily obtained by examining cell attributes. As explained in the next section, the table hierarchical structure and its cell attributes are automatically generated by our table ground truth generation tool.

3 Automatic Table Ground Truth Generation

Figure 2(a) shows the diagram of the system and Figure 2(b) an example of the automatic table ground truth generation results. The following parts describe the automatic table ground truth generation procedure.

3.1 Parameter Generator

This software is used to analyze a given table ground truth and non-table ground truth. Two kinds of parameter sets, $\mathcal{T}$ and $\mathcal{N}$, are designed. There are 12 table layout parameters in $\mathcal{T}$, e.g. column justification, spanning cell position, etc. There are 3 non-table layout parameters in $\mathcal{N}$. e.g. text column number, if there is marginal note, etc. Clearly, $\mathcal{T}$ is designed to add more variety to table instances and test the mis-detection performance of any table detection algorithm. $\mathcal{N}$ is designed to add more variety to non-table instances and test the false alarm performance of any table detection algorithm. Currently, the part which automatically estimates non-table parameters has not been implemented, so we enclose them in dashed lines in Figure 2(a). The table parameter set and non-table parameter set definitions can be found in APPENDIX A and B, respectively.

3.2 Table Latex File Generation Tool

This software randomly selects two parameter elements from sets $\mathcal{T}$ and $\mathcal{N}$. The resulting parameter for a page is a reasonable element in $\mathcal{T} \times \mathcal{N}$. We precomputed two content sets $\mathcal{C}$, $\mathcal{P}$. They are cell word set and non-table plain text set. Elements of $\mathcal{C}$ are random, meaningless English character strings. Elements of $\mathcal{P}$ are the text ground truth file from UW CDROM III [9]. Sets $\mathcal{C}$, $\mathcal{P}$ are the contents of table entities and non-table entities in the generated LaTeX [10] file, respectively. We make sure every element in $\mathcal{C}$ is unique in both $\mathcal{C}$ and $\mathcal{P}$ and it can only be used once for a given file. This software writes out two files: a LaTeX file and a partial ground truth file. In the partial ground truth file, there are table, row header, column header and cell entities with their content and attributes such as cell starting/ending column number, etc.

3.3 DAFS File Generation Tools

Several software tools are used and some minimum manual work is required in this step. LaTeX turned the LaTeX files into DVI files. The DVI2TIFF software [11] converts DVI file to a TIFF file and a so-called character ground truth file which contains the bounding box coordinates, the type and size of the font, and the ASCII code for every individual character in the image. The CHARTRU2DAFS software [16] combines each TIFF file and its character ground truth file and converts it to a DAFS file [12]. The DAFS file has content ground truth for every glyph, which is the basis of content matching in the next step. Then line segmentation and word segmentation software [14] [15] segments word entities from DAFS file. Since we cannot guarantee a 100% word segmentation accuracy, a minimum of manual work using Illuminator [13] tool is required to fix any incorrect word segmentation results inside tables.

3.4 Table Ground truth Generator

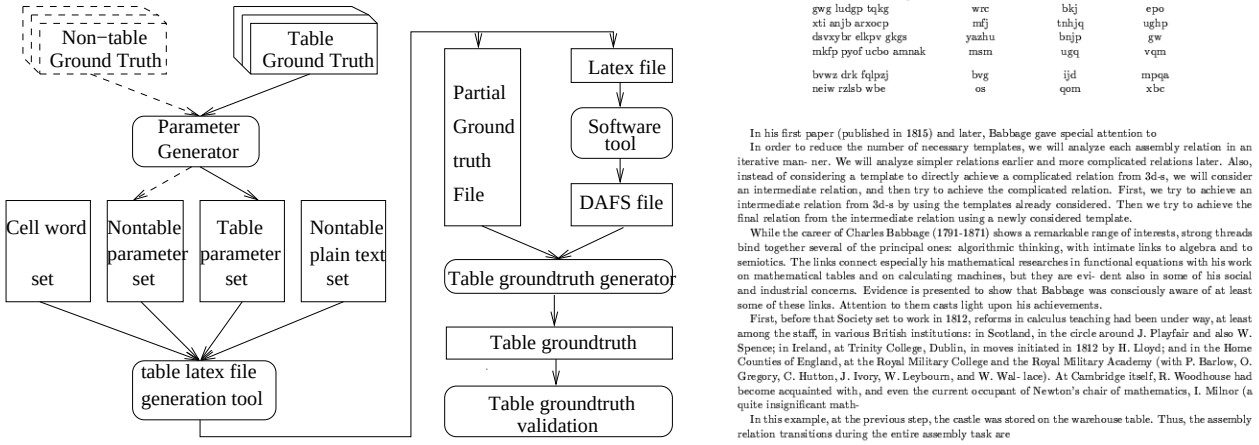
Since we know every word in the tables appears once, we can use content matching method to locate any table related entity of interest. Our software tries to locate any word contents from partial ground truth file in DAFS file. If not, an error is reported. Here is the way to make the previous step even simple. We only need run table ground truth generator twice. The only places we need look at are the files with some errors in the first run. After the correction, we run this software again to obtain the final table ground truth data.

3.5 Table Ground Truth Validation

For normal ground truthing work, validation is a required step to make sure that we get correct ground truth. Our table ground truth validation is also automatically done. It checks the geometric relations among table, row, column and cell entities. If there is any discrepancy, the page can be either removed or given to further manual checking.

4 Conclusion and Future Work

We developed a table and its ground truth automatic generation system and used it to develop our table detection algorithm [7]. Using this software tool, we generated a total of 560 document image ground truth with 482 table entities and $10,298$ cell entities. Since the table simulation work



(a) (b)

Figure 2. (a) Illustrates automatic table ground truth generation procedure; (b) Illustrates an example of generated table page.

was finished by several runs, we did not record the time that the manual checking part costed. However, the most time consuming part was taken by running the softwares. Using this synthetic data set, our table detection algorithm obtained around 90% cell correct identification rates on both real and whole image data sets [7].

To further extend our idea, we want to add more randomness in the generation results. In other words, we want to obtain new table entities which are reasonable but totally different from our input table styles. However, there is an irony. When we use more parameters, we can simulate more incoming table structures but we have less degrees of freedom to generate new tables. Our proposed system still has some limitations. First, we used LaTeX to generate table documents which comply to LaTeX layout rules. We cannot simulate some real world tables that do not fall into this category. Second, our content matching idea requires table cell contents are unique in a given page so we used random, meaningless English character strings as cell content. The results are good enough for those approaches using only geometric information but will have troubles when we consider syntactic/semantic analysis of the content of a table. They are also generally difficult problems for any table synthesis research. More research work is necessary to make improvement upon our current work.

APPENDIX

A Table Parameter Set

The elements of the table parameter set are as follows.

- Number of rows , $rownum$, and number of columns, $colnum$.
- One variable indicating if we have vertical lines, $vline$.
- Justification for each column, $J(1)$, ..., $J(colnum)$. The possible justification values for each column are center, left, right and decimal alignment.

  $J(i) \in \{\text{center, left, right, decimal alignment}\}, i = 1, ..., colnum$

- Lower bound of intercolumn distance, $iclb(1)$, ..., $iclb(colnum - 1)$.
- Upper bound of intercolumn distance, $icub(1)$, ..., $icub(colnum - 1)$.

  The real intercolumn distances,$ic(1)$, ..., $ic(colnum - 1)$, are uniformly distributed integer numbers between the appropriate lower bounds and upper bounds. The real length unit is millimeter.

$$\begin{aligned} &ic(i) \in [iclb(i), icub(i)] \\ &ic(i) \in Z, i = 1, ..., colnum - 1 \end{aligned} \quad (1)$$

- Interrow distance, $ir(1)$, ..., $ir(rownum - 1)$.

- Mean of cell length for one column, $clmean(1)$, ..., $clmean(colnum)$.
- Half range of cell length for one column, $clhr(1)$, ..., $clhr(colnum)$.

  We call the number of characters,including middle blank space, in one cell as cell length. Usually cells in the same column have similar lengths. For each cell, its real length is uniformly distributed integer numbers between the lower bound and upper bound of its column. i.e.

$$\begin{aligned} &cl(i) \in [clmean(i) - clhr(i), clmean(i) + clhr(i)] \\ &cl(i) \in Z, i = 1, ..., colnum \end{aligned} \quad (2)$$

  For the purpose of groundtruthing by content matching, we generated the cell contents offline. They are unique and were randomly generated. There are 9 groups of cell contents. The lengths of words of each group follow normal distribution whose standard deviation is 1 and mean values are 1, 2, ..., 9, respectively. If mean cell length is between 1 and 9, we just take one word from its appropriate group. If mean cell length is longer than 9, we select words from the group with mean length as 5 and put blank space between the selected words. The total character length follows (2).
- Horizontal line list. Although we are aimed at table detection without using horizontal lines, we can still generate the tables with some horizontal lines. In the horizontal line list, each horizontal line is specified by its row location and starting column index and ending column index.
- Spanning cell list. Sometimes one column header has different justification with other cells in the same column. Sometimes one cell can span over several columns or rows. In LaTeX we can create the cells spanning over several rows and columns. Our current system can only generate the cells spanning over several columns. In the spanning cell list, we specify them by row location and starting column index and ending column index.
- Empty cell list. Not all the tables are full. We locate the empty cells by row location and starting column index and ending column index.
- Row header list and column header list. Our table groundtruth data has three levels: table, row/column headers and cells. According to the specified the cell contents of row/column headers. we can get the bounding boxes for row/column headers as the union of those of located cells.

B Non-table Parameter Set

The elements of the non-table parameter set are as follows.

- List item, *li*. Its possible values are *yes* or *no*.
- Marginal note, *mn*. Its possible values are *yes* or *no*.
- Text column number, *tcn*.

References

[1] M. Rahgozar and R. Cooperman, "A graph-based table recognition system", *Document Recognition III, SPIE*, San Jose, California, pp. 192-203, 1996

[2] J. Shamilian, H.S. Baird and T.L. Wood, " A Retargetable Table Reader", *ICDAR'97*, pp.158-163, Ulm, Germany, August 1997

[3] K. Zuyev, "Table Image Segmentation", *ICDAR'97*, pp.705-708, Ulm, Germany, August 1997, pp.705

[4] T. G. Kieninger, "Table Structure Recognition Based on Robust Block Segmentation", *Document Recognition V*, pp. 22-32, January 1998

[5] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong, "Medium-independent table detection", *SPIE Document Recognition and Retrieval VII*, pp. 291-302, San Jose, California, January 2000

[6] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong, "Table Structure Recognition and Its Evaluation", *SPIE Document Recognition and Retrieval VIII*, San Jose, California, January 2001

[7] Y. Wang, I.T. Phillips and R. Haralick, "Automatic Table Ground Truth Generation and A Background-analysis-based Table Structure Extraction Method", Accepted to *ICDAR01*

[8] G. Liu, R.M. Haralick, "FLIR ATR Using Location Uncertainty," *Journal of Electronic Imaging*, vol. 9, no. 2, pp. 178–193, Apr. 2000.

[9] I. Phillips. "Users' reference manual", *CD-ROM, UW-III Document Image Database-III*, 1995.

[10] M. Goossens, F. Mittelbach and A. Samarin, "The LaTeX Companinon", *Addison-Wesley Publishing Company*

[11] T. Kanungo, "DVI2TIFF User Manual", *UW English Document Image Database - (I) Manual*, 1993

[12] RAF Technology, Inc., "DAFS:Document Attribute Format Specification", 1995

[13] RAF Technology, Inc., "Illuminator User's Manual", 1995

[14] J. Liang, "Document Structure Analysis and Performance Evaluation", *Ph.D thesis*, Univ. of Washington, Seattle, WA, 1999.

[15] Y. Wang, I. T. Phillips and R. Haralick, "Statistical-based Approach to Word Segmentation", *15th International Conference on Pattern Recognition, ICPR2000*, Vol. 4, pp.555-558, Barcelona, Spain, September 2000

[16] http://isl.wtc.washington.edu/ylwang/auttabgen.html