

Chapter 4

RANDOM TABLE AND ITS GROUND TRUTH AUTOMATIC GENERATION

Although there is continuing interest in the table understanding problem ([12, 83, 25, 85, 107, 33, 34, 28]), there are no publicly available table ground truth data sets. In the UW document image database III(UW CDROM III) [76], there are 215 marked table zones but no structure data for them. Detailed table structure information is required for a table detection system evaluation [98]. Clearly, UW CDROM III cannot be directly used to evaluate table detection system.

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. According to Hu et.al's research [32], there may exist more than one acceptable "truth" and/or incomplete or partial "truth". Such problems were brought by the operators' bias and were not easily eliminated. However, studying synthetic data [63] at some research phase is a common practice in computer vision field. It has the advantage of extremely low cost, automatic creation of ground truth information, less bias aberrations 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 each 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 [98]. The software package is publicly available at [91].

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 (e.g. an XML DTD).

The remainder of this chapter is organized as follows. First, we give our table ground truthing specification in 4.1. The detailed automatic table ground truth generation algorithm is described in 4.2. Our future work direction is given in 4.3. We put our table parameter set and non-table parameter set definitions in APPENDIX A and B.

4.1 Table Ground Truth Specification

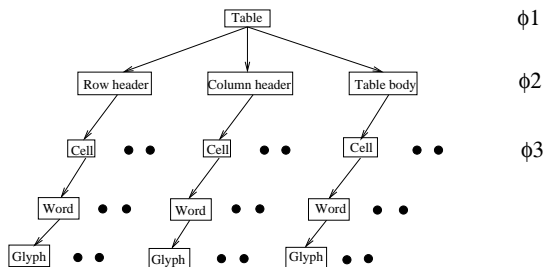


Figure 4.1: Illustrates a table hierarchy model

We defined the table structure in a hierarchical structure, as shown in Figure 4.1. 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.

4.2 Automatic Table Ground Truth Generation

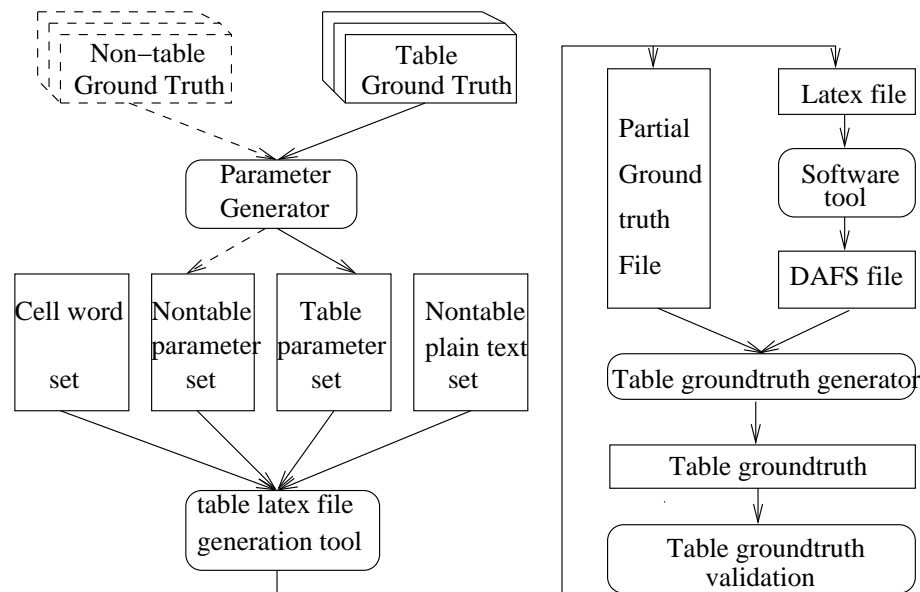


Figure 4.2: Illustrates automatic table ground truth generation procedure.

Figure 4.2 shows the diagram of the system and Figure 4.3 an example of the automatic table ground truth generation results. The following parts describe the automatic table ground truth generation procedure.

In order to speed up the programming process, automatic programming has been proposed. The method tries to develop geometric reasoning systems which can generate textual programs to control a robot from geometric information given by geometric models and task specifications. This direction is quite promising, however, there are many issues to be addressed before we have a complete automatic programming system; It is quite difficult to build a complete automatic programming system, though perhaps not impossible.

	bethzp	emi	erzub
	jtkypzfel	pllhrne	vuiokjyer
klb rxrv udos aoqs	oaw	vwi	wmzo
gwg ludgp tqkg	wrc	bkj	epo
xti anjb arxocp	mfj	tnhjq	ughp
dsvxybr elkpv gkgs	yazhu	bnjp	gw
mkfp pyof ucbo amnak	msm	ugq	vqm
bvwz drk fqlpzj	bvg	ijd	mpqa
neiw rzlsb wbe	os	qom	xbc

In his first paper (published in 1815) and later, Babbage gave special attention to

In order to reduce the number of necessary templates, we will analyze each assembly relation in an iterative manner. We will analyze simpler relations earlier and more complicated relations later. Also, instead of considering a template to directly achieve a complicated relation from 3d-s, we will consider an intermediate relation, and then try to achieve the complicated relation. First, we try to achieve an intermediate relation from 3d-s by using the templates already considered. Then we try to achieve the final relation from the intermediate relation using a newly considered template.

While the career of Charles Babbage (1791-1871) shows a remarkable range of interests, strong threads bind together several of the principal ones: algorithmic thinking, with intimate links to algebra and to semiotics. The links connect especially his mathematical researches in functional equations with his work on mathematical tables and on calculating machines, but they are evident also in some of his social and industrial concerns. Evidence is presented to show that Babbage was consciously aware of at least some of these links. Attention to them casts light upon his achievements.

First, before that Society set to work in 1812, reforms in calculus teaching had been under way, at least among the staff, in various British institutions: in Scotland, in the circle around J. Playfair and also W. Spence; in Ireland, at Trinity College, Dublin, in moves initiated in 1812 by H. Lloyd; and in the Home Counties of England, at the Royal Military College and the Royal Military Academy (with P. Barlow, O. Gregory, C. Hutton, J. Ivory, W. Leybourn, and W. Wallace). At Cambridge itself, R. Woodhouse had become acquainted with, and even the current occupant of Newton's chair of mathematics, I. Milnor (a quite insignificant math-

In this example, at the previous step, the castle was stored on the warehouse table. Thus, the assembly relation transitions during the entire assembly task are

Figure 4.3: Illustrates an example of generated table page.

4.2.1 *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 are marginal notes, 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 4.2. The table parameter set and non-table parameter set definitions can be found in APPENDIX A and B, respectively.

4.2.2 *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 [76]. Sets \mathcal{C} , \mathcal{P} are the contents of table entities and non-table entities in the generated L^AT_EX [24] 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 L^AT_EX 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.

4.2.3 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 [44] 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 [91] combines each TIFF file and its character ground truth file and converts it to a DAFS file [36]. 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 [55, 97] segments word entities from DAFS file. Since we cannot guarantee a 100% word segmentation accuracy, a minimum of manual work using Illuminator [37] tool is required to fix any incorrect word segmentation results inside tables.

4.2.4 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 locates any word contents from the partial ground truth file in the DAFS file. If this cannot be done, 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.

4.2.5 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.

4.3 Summary

We developed a table and its ground truth automatic generation system and used it to develop our table detection algorithm [98]. Using this software tool, we generated ground truth of a total of 560 document images with 482 table entities. Since the table simulation work was finished by several runs, we did not record the time that the manual checking part costs. However, the most time consuming part was taken by running the software. Using this synthetic data set, our table detection algorithm obtained around 90% cell correct identification rates on both real and whole image data sets [98].

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. Work in [5] can give us some inspiration, but more study is necessary to make improvement upon our current work.