

## Chapter 3

# TABLE STRUCTURE UNDERSTANDING PROBLEM

### **3.1 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, document image analysis researchers have successfully developed many outstanding methods for character recognition, page segmentation and understanding 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 understanding is an important problem in the document layout analysis field. Its application can be found in image-XML conversion, information retrieval, and document classification, etc.

We developed an automatic table ground truthing system. It can analyze any given table ground truth and generate documents having similar table elements while adding more variety to both table and non-table parts. Ground truthing is tedious and time-consuming. Using our novel content matching ground truthing idea, the table ground truth data for the generated table elements become available with little manual work. We make this software package publicly available at [91].

We formulate table detection, table decomposition and table structure understanding problems in the whole document hierarchy. We propose a statistical based table detection algorithm. To systematically evaluate and optimize the algorithms, a

performance evaluation protocol using an area overlapping measure is addressed for table structure understanding evaluation.

In this chapter, we give a literature review in Section 3.2. We present a document structure model in Section 3.3. We give our background structure definitions in Section 3.4. We give a formal table structure understanding problem statement in Section 3.5.

### **3.2 Literature Review**

Tables are an important means for communicating information in documents, and understanding such tables is a challenging problem in document layout analysis. Although there are different document formats, e.g. document images, ASCII text documents, web documents, etc, here we focus on table structure understanding in document images.

Some tables are similar to ruled forms. Many papers were published for form processing [103, 106, 10]. In form data extraction, templates are constructed from empty forms and correlated with fill-in forms. Form processing emphasizes extracting data from some given forms. Table structure understanding mainly deals with those tables embedded in the documents. The tables usually have unknown format. Table structure understanding aims to segment tables from the document pages and extract the information from these tables.

Table structure understanding includes two subproblems: table detection problem and table decomposition problem. Since table detection is a crucial step for table structure understanding, it is also a topic which is of interest to many researchers. By the strategies or methods in which algorithms are used in the table detection, the algorithms can be classified into three categories: 1) predefined table layout based, 2) heuristics based and 3) statistical or optimization based. The predefined table layout based algorithms use a set of strict, predefined table layout information to

detect tables. For a given type of images, it usually can have a satisfying detection performance. However, its extension ability is very limited. The heuristics based algorithms use a set of rules or pre-defined syntax rules of the grammars to derive decisions. The complex heuristics are usually based on local analysis. It sometimes has a even more complicated post-processing part. As for statistical or optimization based algorithms, they either do not need parameters or the needed free parameters which are used in the process are obtained via off-line training processes. The estimated parameters are used in the decisions which govern the decision making.

Since table itself is a complicated two dimensional structure, its groundtruthing is also an interesting research topic. Up to now there is no publicly available large table image data sets. However, some researchers reported their unsuccessful table groundtruthing work in some large data set.

Chandran et.al [12] gave a clear algorithm to extract the structure of the table, regardless of the presence or the absence of lines. They had some assumptions about the table. The assumptions made their work fairly easy.

Green et. al [25] developed a strategy for extracting the underlying relational information from the images of printed tables. Given a table model, the visual clues that exist in the images were used for extracting first the physical, and then the logical structure of the tables. A table model was addressed to extract logical table information from table image.

Shamilian et.al [85] designed and implemented a system architecture for reading machine-printed documents in known predefined tabular-data layout styles. The row-and-column structure of structure of horizontal tables suggests an analogy with relational data bases.

Zuyev [107] proposed an approach which introduces a concept of a table grid which can serve for advanced methods of table structure. Table hypothesis generation was guided by visual clues. Classification of generated hypothesis generally requires some threshold values. They were obtained by an analysis of the connected components

projection profile.

Kieninger et.al [46, 45] presented a bottom-up approach to the table detection algorithm. First an arbitrary word was selected as the seed. Then the local adjacency connection relation was used to expand to a whole text block. Later, some postprocessing to the initial block segments were done to get the refined results.

A dynamic programming table detection algorithm was given in [33]. By defining table quality measures, it detected tables based on computing an optimal partitioning of a document into some number of tables. Its high-level framework is independent of any particular table quality measure and independent of the document medium. The algorithm works for both ASCII and image documents. 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.

Klein [50] introduced three approaches for an industrial document analysis system. The three approaches include: searching for a set of known table headers, searching for layout structures which resemble parts of columns, and searching for groupings of similar lines. Approach 1 was not tolerant enough toward some kinds of even minor aberrations and it able to spot about 80% of all table headers. Approach 2 yielded 90% correctness in a test on 1200 real documents. No experimental results were reported for approach 3.

Two papers [34, 28] reported their research on the table decomposition problem. Handley [28] presented a table analysis system which reconstructed table formatting information from table images whether or not the cells are explicitly delimited. Inputs to the system are word bounding boxes and any horizontal and vertical lines that delimit cells. Using a sequence of carefully-crafted rules, multi-line cells and their inter-relationships are found even though no explicit delimiters are visible. Hu et.al [34] presented algorithms that recognize tables in ASCII text. First hierarchical clustering was used to identify columns and then spatial and lexical criteria were used to classify headers. The algorithm was tested on 26 Wall Street Journal articles in

text format (WSJ database) and 16 email messages. The overall agreement was 82% for the WSJ documents and 73% for the email documents.

The principle that for every document analysis task there exists a mechanism for creating well-defined ground-truth is a widely held tenet. Past experience in the research community with standard data sets providing ground-truth for character recognition and page segmentation tasks supports this belief. However, Hu et.al [32] reported a number of serious hurdles connected with the groundtruthing of tables from UW Document Image Database (UWI) [77]. From their experience, there may exist more than one acceptable “truth” and/or incomplete or partial “truth” for table entities.

### 3.3 Document Structure Model

We formally define a *Rectangle Layout Structure*(*RLS*) as a triple  $(C, R, Q)$ , where

- $R$  is a rectangular area;
- $Q$  is a physical label type (e.g. page, textblock, table, etc.);
- $C$  is either empty or a set of rectangle layout structures  $\{C_n, R_n, \theta_n\}_{n=1}^N$ , satisfying that  $\{R_1, \dots, R_n\}$  is a cover of  $R$ .

We denote by  $\mathcal{D}$  the set of RLS in a given document image. Some functions are associated with  $\mathcal{D}$ .

#### 1. Attributes

- We denote by  $\mathcal{F}$  the set of format attributes (column number, row number, column justification, etc.).
- $S : \mathcal{D} \rightarrow \mathcal{F}$  specifies the format attributes for each rectangle layout structure.

## 2. Measurements

- We denote by  $\Lambda$  the measurement space.
- $V : \wp(\mathcal{D}) \rightarrow \Lambda$  specifies measurement made on subset of  $\mathcal{D}$ .

Using this model, we have the below definitions.

A *page* is a  $\text{RLS}(C, R, Q)$ , where

- $Q$  is the label page;
- The RLS in  $C$  must have labels from a set {textblock, table, horizontal/vertical blank block}.

A *textblock* is a  $\text{RLS}(C, R, Q)$ , where

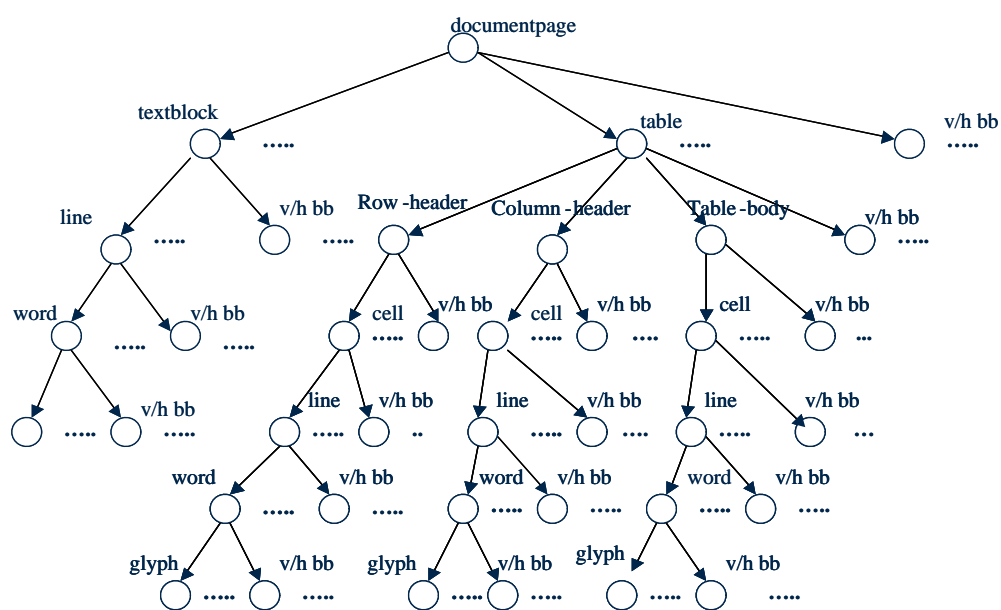
- $Q$  is the label textblock;
- The RLS in  $C$  must have labels from a set {line, horizontal/vertical blank block}.

A *table* is a  $\text{RLS}(C, R, Q)$ , where

- $Q$  is the label table;
- The RLS in  $C$  must have labels from a set {table body, row header, column header, horizontal/vertical blank block};
- One RLS of  $C$  must have the label table body;
- At least one RLS of  $C$  must have the label horizontal blank block;
- At least one RLS of  $C$  must have the label vertical blank block.

Similarly, we can define *row header*, *column header* and *table body*, etc.

We give an example of a document hierarchy model using this idea shown in Figure 3.1.



v/h bb: vertical/horizontal blank block

Figure 3.1: Illustrates a document hierarchy model

### 3.4 Background Analysis Structure Definition

Although some background analysis techniques can be found in the literature([1, 2]), none of them, to our knowledge, have extensively studied the statistical characteristics of their background structure. Instead, they mainly use heuristic rules on their background structure. Our background analysis was based on some basic units: horizontal and vertical blank blocks. These signature-like features are designed to give us more information on the distributions of the big foreground chunks in a given document entity. We use several definitions to describe this feature. We will use these definitions in table structure understanding and zone content classification algorithms.

Assume black pixels are foreground and white pixels are background.

Definition 1: Let  $\mathcal{Z}$  represent a *document entity* with  $R$  rows and  $C$  columns. Let  $(x_1, y_1)$  be the coordinate of its lefttop vertex,  $\mathcal{Z} = \{(r, c) \in Z \times Z | x_1 \leq r < x_1 + R, y_1 \leq c < y_1 + C\}$ .

Definition 2: Let  $p$  be a *horizontal white run*  $p$ ,  $p = ((r_1, c_1), \dots, (r_n, c_n))$ , where  $(r_i, c_i) \in \mathcal{Z}$ ,  $r_i = r_{i-1}$ ,  $c_i = c_{i-1} + 1$ , for  $i = 2, \dots, n$ , and pixel  $(r_1, c_1)$ ,  $(r_n, c_n)$  must have a black pixel or the document entity border on its left and right side, respectively. For each run, we call the location of the starting pixel of the run and its horizontal length as  $\text{Row}(p)$ ,  $\text{Column}(p)$ , and  $\text{Length}(p)$ , respectively.

Definition 3: Let  $\mathcal{HR}$  be a *horizontal blank block*,  $\mathcal{HR} = (p_1, \dots, p_n)$ , where  $\text{Row}(p_i) = \text{Row}(p_{i-1}) + 1$ ,  $\text{Column}(p_i) = \text{Column}(p_{i-1})$ ,  $\text{Length}(p_i) = \text{Length}(p_{i-1})$ , for  $i = 2, \dots, n$ . Clearly, the same idea can be applied to define *vertical white run* and *vertical white blank block*,  $\mathcal{VR}$ .

Definition 4: A horizontal blank block  $\mathcal{HR} = b_r \times b_c$ , with lefttop vertex coordinate  $(x_{b1}, y_{b1})$ , is a *large horizontal blank block* if and only if it satisfies the following conditions:

1.  $\frac{b_c}{C} > \theta_1$ , where  $\theta_1$  is 0.1;
2.  $x_{b1} \neq x_1$  and  $x_{b1} + b_c \neq x_1 + C$ , where  $C$  is the column number in the document entity.

Definition 5: A vertical blank block  $\mathcal{VR} = b_r \times b_c$ , with lefttop vertex coordinate  $(x_{b1}, y_{b1})$ , is a *large vertical blank block* if and only if it satisfies the following conditions:

- Its row number and column number are large enough compared with the current document entity. Specifically,  $b_r \gg mh$  and  $\frac{b_c}{mw} > \theta_2$ , where  $mh$  and  $mw$  are the median height and median width of text glyphs in the document zone.  $\theta_2$  is empirically determined as 1.4;
- It does not touch left or right side of the document zone bounding box, i.e. **(2)**.  $x_{b1} \neq x_1$  and  $x_{b1} + b_c \neq x_1 + C$ ,

where  $C$  is the column number in the document entity.

### 3.5 Table Structure Understanding Problem Statement

Table structure understanding problem has two subproblems: table detection problem and table decomposition problem.

With the defined document structure model, we have definitions as below:

A *tablezone* is a  $RLS(C, R, Q)$ , where

- $Q$  is the label tablezone;
- The RLS in  $C$  must have labels from a set  $\{\text{word, horizontal/vertical blank block}\}$ ;
- At least one RLS of  $C$  must have the label word.

A *textzone* is a  $RLS(C, R, Q)$ , where

- $Q$  is the label *textzone*;
- The RLS in  $C$  must have labels from a set {word, horizontal/vertical blank block};
- At least one RLS of  $C$  must have the label word.

A *documentpage* is a  $RLS(C, R, Q)$ , where

- $Q$  is the label *documentpage*;
- The RLS in  $C$  must have labels from a set {tablezone, textzone, horizontal/vertical blank block}.

Figure 3.2 shows a hierarchy including *documentpage*, *tablezone* and *textzone* entities. The problem of *table detection* can be formulated as: *Given a page  $\phi$  having a rectangle area  $R$  and a set of words,  $W$ , table detection constructs a  $RLS(C, R, Q)$ , where*

- $Q$  is the label *documentpage*;
- $P((C, R, Q)|\phi)$  is maximized;
- Each word of  $W$  participates exactly one RLS in  $C$  or its descendent.

Figure 3.3 shows a table structure. Under *tablezone* level, we have row header, column header and cell entities. Given a detected table entity, *table decomposition* problem is to determine the its structure and identify its elements such as rowcolumn headers, cells, etc.

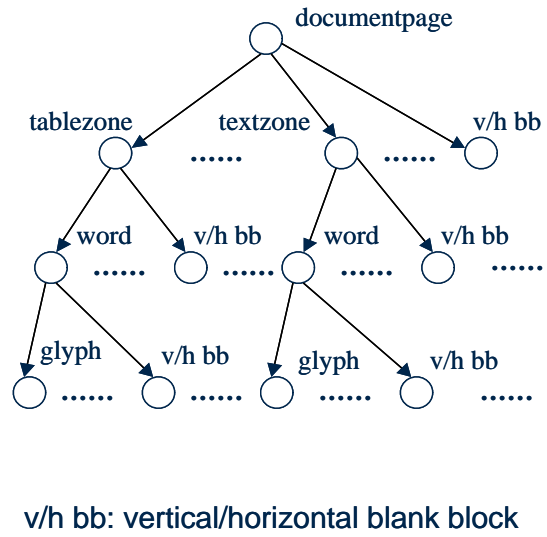


Figure 3.2: Illustrates a table hierarchy model for table detection problem

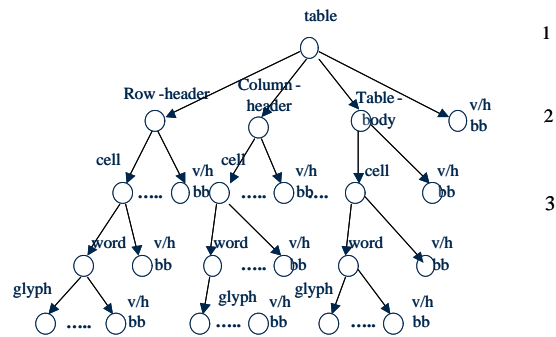


Figure 3.3: Illustrates a table hierarchy model for table decomposition problem