

Detection, Extraction and Representation of Tables

J.-Y. Ramel, M. Crucianu, N. Vincent, C. Faure

*Laboratoire d'Informatique
Université de Tours - France
{ramel,vincent}@univ-tours.fr*

*LTCI – CNRS
GET – ENST Paris - France
cfaure@tsi.enst.fr*

Abstract

We are concerned with the extraction of tables from exchange format representations of very diverse composite documents. We put forward a flexible representation scheme for complex tables, based on a clear distinction between the physical layout of a table and its logical structure. Relying on this scheme, we develop a new method for the detection and the extraction of tables by an analysis of the graphic lines.

To deal with tables that lack all or most of the graphic marks, one must focus on the regularities of the text elements alone. We propose such a method, based on a multi-level analysis of the layout of text components on a page. A general graph representation of the relative positions of blocks of text is exploited.

1. Introduction

The electronic exchange of complex documents often relies on such formats as PDF and Postscript or on images of the documents. While these portable formats do convey a rich visual information to the user, they do not encourage the efficient reuse, organization and distribution of content because the visual structure of a document is not explicitly represented in the corresponding file. The reconstruction of this structure (in terms of components and relationships that are meaningful to the reader) from the visual aspect of a document is a prerequisite to a more flexible exploitation of its content. Our work concerns the extraction, from such a representation, of both the physical and logical structure for a wide-ranging class of documents [1]. Following a general trend [2], we retained XML for the target representation.

We focus in the following on the detection of tables and the reconstruction of their structure. This is an important aspect of document understanding because tables are pervasive and the condensed information they convey is highly relevant to the reader.

The detection of tables in unrestricted documents is a challenging problem for two reasons. First, tables are so diverse that they hardly have any easy to identify characteristic in common. While in some cases we should favor the analysis of groups of graphic lines, in others we must rather pay attention to relatively regular configurations of text elements.

Second, most tables share the property of containing both text elements and graphic lines with other components of a document, such as underlined text, headers, footers or figures. It follows that for a reliable detection of tables we should not ignore the presence of these other components.

To reconstruct the structure of a table we need a model able to encompass the rich variety of existing tables. In the next section we put forward such a model, relying on a multi-level representation. We then provide, in section 3, a brief description of the data we are working on. Section 4 presents the methods we developed for the detection and extraction of tables having borders and rules marked by graphic lines. In section 5 we turn to the analysis of the layout of text elements.

2. How to represent tables

In order to characterize the structure of tables for a wide-ranging class of documents, a rich and flexible representation framework is needed. Both the physical layout and the logical structure of a table must be described. While the information concerning the original layout may help visualization, the logical structure enables the presentation of tables through different media and an automatic processing of their content.

2.1. Diversity of the tables

The simplest tables are regular matrices of cells: all the cells of a line have the same height and all the cells of a column have the same width. The borders of all the cells are marked by graphic lines.

However, very few tables follow such a strict prescription. Regarding the physical layout, we can often note the presence of cells extending over several lines or columns, as in Figure 1, and misalignments between the borders of neighboring cells. Also, most of the time, only a few of the borders and rules of a table are marked by graphic lines.

Model	Best results			Best mean results				
	Par	Test1	Test2	Par	Test1		Test2	
		NMSE	NMSE		NMSE	Sd	NMSE	Sd
TAR	18	0.097	0.280	–	–	–	–	–
RNN-CBPTT	15	0.092	0.251	15	0.094	0.0064	0.281	0.034

Figure 1. Example of a table

Tables frequently show a complex logical structure, comprising several levels of line and/or column groups (as in Figure 1), each group having its own header. Extended cells and suitably selected table rules serve to bring this hierarchy to light.

2.2. Existing representations

An appropriate representation framework should be able to effectively describe both the physical layout and the logical structure of any table, including those displaying irregular layout and/or complex structure.

The best-known table representation schemes were put forward by the World Wide Web Consortium (W3C) in the specification of XHTML [3] and by the Organization for the Advancement of Structured Information Standards (OASIS) [4]. We must evaluate the generality and the effectiveness of these representation schemes.

2.2.1. Tables in HTML 4.01 and XHTML 1.1.

According to the description introduced by the W3C in the specifications of HTML 4.01 and XHTML 1.1 [3], a table can contain a header, a footer and a body. Table lines are explicitly defined as being composed of header cells and data cells. Each header cell can specify its span as a list of lines or columns and each data cell can specify a list of header cells affecting it.

Columns are implicitly defined as cell alignments. Together with the column groups, they only serve to describe visual formatting attributes shared by the corresponding cells. Misalignments between the borders of neighboring cells are not allowed. Extended cells are allocated to one line (sometimes arbitrarily) and are described as spanning several lines and/or columns. The specification of the logical structure of a table is limited to the partition in a header, a footer and a body, together with the distinction between header and data cells.

2.2.2. OASIS representation in SGML and XML.

According to the representation scheme put forward by the OASIS [4], using SGML and XML as target languages, a table can be composed of several independent segments.

Each segment is a group of lines and contains a segment header and a segment body. Each line is composed of cells; there is no distinction between header and data cells.

Columns are defined for every segment, but they only serve to describe visual formatting attributes shared by cells. Misaligned cell borders are not allowed. Extended cells, allocated to one line (sometimes arbitrarily), are described as spanning a number of lines and/or an explicitly identified set of columns.

To specify the logical structure of a table we can only use table segments (i.e. groups of lines) and distinguish between lines belonging to a header or to a body.

2.3. A multi-level representation

An analysis of the two representation schemes shows that they share the same weaknesses. First, irregular physical layouts are difficult to represent. Misaligned cell borders are simply not allowed and ad-hoc solutions are provided for the definition of extended cells. Then, rows and columns do not play a symmetrical role since cells explicitly belong to lines but only implicitly to columns. To end with, rather limited means are supplied for the description of the logical structure of a table.

Underlying these weaknesses is the fact that in the existing representation schemes physical layout and logical structure are not clearly set apart. We placed this separation at the heart of a new representation framework, based on the distinction between virtual and real cells. According to our framework, every table is built on a perfectly regular matrix of virtual cells. The virtual cells are the smallest rectangles obtained by extending all the borders between real cells up to the bounding box of the entire table, as in Figure 2. Then, there are no irregularities in the layout of the virtual cells. A real cell (defined by thick borders in Figure 2) is composed of one or several virtual cells and is not necessarily a rectangle.

Model	Best results			Best mean results				
	Par	Test1	Test2	Par	Test1		Test2	
		NMSE	NMSE		NMSE	Sd	NMSE	Sd
TAR	18	0.097	0.280	–	–	–	–	–
RNN-CBPTT	15	0.092	0.251	15	0.094	0.0064	0.281	0.034

Figure 2. Virtual cells and real cells

Real cells have a content and may receive logical labels (e.g. “header”). Logical lines group together real cells, lower-level logical lines or lower-level logical columns. Logical columns play a perfectly symmetrical role. Hierarchies of logical lines and/or columns can be defined. At the top level, the logical structure of a table is described as an ordered set of either logical lines or logical columns. However, two complementary top level logical descriptions—one in terms of lines, the other in terms of columns—can also be devised.

In our representation framework, the description of the physical layout of a table is thus based on the virtual cells, while the logical structure is built upon the real cells. While remaining rather simple, this framework provides enough flexibility to fulfill our requirements concerning both physical layout and logical structure. Following our proposal, we developed [5] a DTD (Document Type Definition) for the representation of tables in XML.

Before presenting the methods we devised for detecting tables and extracting their structure, we must provide a few details concerning the documents we process.

3. Description of the corpus

The documents we are working on are intended for internal communications and for exchanging information between organizations. Among them we can find internal reports, financial statements or analyses, technical or commercial notes, slide shows, etc. written in several languages and produced with various applications, usually by amateur communicators. Most of the pages (in a total of 700) are composite—they contain text paragraphs, graphics, tables and images—and have very diverse layouts. A page of such a document is shown in Figure 3.



Figure 3. Document (left) and associated data (right)

3.1. Initial representation of the data

The exchange format employed for representing the documents consists of a set of printing instructions given to a virtual printer. Each instruction concerns an element of one of the following types: text, graphic line, rectangle, polygonal line, ellipse, image or Bézier curve. Several attributes are specified for every element, including its bounding box. An example is shown on the right side of Figure 3, with handles displayed for every element.

We thus have a direct access to every component of a document, which is not the case when the only data available is the image of a document. On the other hand, the relation between these components and the visual aspect of a document is usually complex. We often encounter invisible items (e.g. text covered by opaque

graphics), visually continuous components (words, graphic lines) can be composed of several independent elements and visually identical components can be obtained by various means. Some preprocessing must then be done in order to come close to the visual aspect of a document.

3.2. Characteristics of the tables

The tables we found in our corpus of documents are very diverse. First, they can have all, some or none of the borders and rules marked by graphic lines. When vertical rules are not marked, the spacing between neighboring columns is usually large, but can sometimes be small, comparable to the spacing between words in a text paragraph. Several types of vertical alignment are frequently employed in a same table.

Then, cells extending over several lines and/or columns occur very often. Also, individual cells may contain very different entities such as text paragraphs, graphics, images or a mix of the three. This may lead to significant variation in the dimensions and the “occupancy rate” of the cells.

To end with, we found a few “false tables”, which are sets of partially aligned text elements surrounded by rectangles. At first sight these compounds look like tables, but a semantic analysis shows that this is not the case.

A study of the tables in our corpus allowed us to verify that different perceived regularities complete each other to let the reader detect the presence of a table and identify its structure. We also found that the more complex a document and the more irregular the layout of a table, the more graphic lines are employed as borders and rules. While the extraction of tables should rely on a combined analysis of the graphic lines and of the content (spacing, alignments), methods able to independently analyze the two should then also be developed. For highly irregular table layouts we should favor the study of the graphic lines. When such graphic marks are absent, we are only able to analyze the content of a table, but we can expect highly regular configurations.

3.3. Preprocessing for the extraction of tables

In order to restore the visual identity of the components we exploit, some preprocessing is required. We start by eliminating elements that are invisible, either because they are hidden by others or because they have the same color as the local background. Then, we are trying to find the longest continuous horizontal or vertical graphic lines. To obtain them we are splitting rectangles, assimilating thin rectangles to lines, extracting horizontal and vertical segments from polygonal lines, and eventually merging the line segments that are close to each other.

Regarding the text components, we are looking for two levels of description: lines and blocks of text. To obtain the lines, we are merging text elements that are aligned and close to each other. The bounding box of each line is cropped by eliminating at both ends the unprintable characters that may be present. When text lines are equally spaced and their vertical projections are connected, they are merged into a block. Text lines and blocks can be split if they cross vertical or horizontal graphic lines.

4. Table extraction from graphic lines

The first problem we must solve is to detect the presence of tables and to separate each of them from the rest of the document. In the following we provide a solution for tables having all the borders and some of the rules marked by graphic lines [6]. The second problem is the reconstruction of the structure of each table. An analysis of the graphic marks only allows us to extract that part of the structure they emphasize.

The first step in the detection process is the construction of the set of minimal rectangles, from the lists of horizontal and vertical graphic lines. A rectangle is considered to be minimal if it does not share any border with some other rectangle it may contain. We then find the connected components of the set of minimal rectangles.

For any table having the borders and some rules marked by graphic lines, the minimal rectangles corresponding to individual cells (or sets of cells) form a partition of the bounding box. To identify the tables among the connected sets of minimal rectangles, we compare the area of the bounding box of each ensemble to the sum of the areas of the minimal rectangles it contains; a small difference designates a table (left part of Figure 4). We also consider that a table must be composed of at least two rectangles.

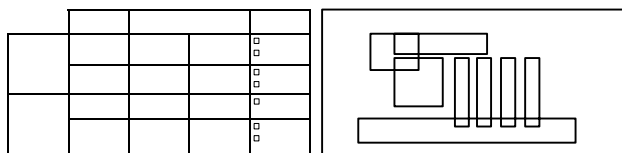


Figure 4. A table (left) and a non table (right)

If all the graphic lines contributing to the minimal rectangles of a table reach its bounding box, the table is regular (every real cell contains only one virtual cell) and we can directly identify the content of each cell. An example of such a table is shown in Figure 5. For this table the finer structure cannot be extracted from the graphic lines alone, but rather by an analysis of the text elements.

When some of the graphic lines contributing to the minimal rectangles of a table do not reach its bounding box, the table is irregular. In order to describe its structure, the graphic lines are first extended up to the bounding box

of the table and the virtual cells are identified. Then, each real cell having borders marked by graphic lines is expressed as a cluster (not necessarily rectangular) of virtual cells and its content is extracted. Just as for regular tables, the (virtual and real) cells may in fact correspond to sets of cells and an analysis of the text elements is needed.

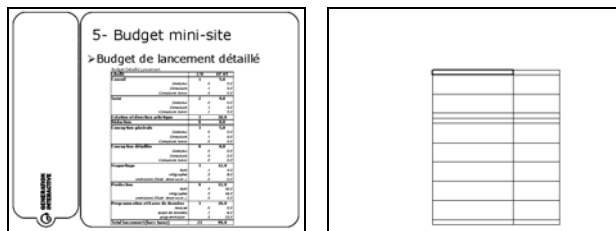


Figure 5. Detection and extraction from graphic lines

In our corpus of documents we detect 80 out of the 81 tables having the borders and some of the rules marked by graphic lines. We also label as tables 23 of the 129 composite objects that are not tables but figures, headers or footers. Most of these candidates can be rejected by noting that all their cells are empty, but for the others an analysis of the text elements should be performed.

5. Analysis of the layout of text elements

Few approaches for detecting tables are based only on the textual content of a document. We mention here two of these approaches. The first one relies on the bottom-up clustering of the words in a document and the associated detection of columns [7]. In the second approach [8], a score is computed for every partition of a document into one or more table and non table elements. The features taken into account are the white space correlations and the vertical connected components. In order to increase the robustness of the detection method—needed for such a wide-ranging class of documents—we decided to put forward a new approach, based on a multi-level analysis.

In a simplified view, our method is a top-down study of layout regularities. More specifically, we begin with the examination of the global configuration of the blocks of text (see §3.3), we then take into account their more local characteristics (alignments, spacing) and we end by a study of the lines of text.

We represent the configuration of the blocks with the help of a directed graph, where the nodes are the blocks of text and the edges correspond to the relations between blocks. The space surrounding a block is cut into eight areas, as shown in Figure 6. The label of the edge from node A to node B contains a list of the areas around block A that are crossed by block B, together with the corresponding proportions of the width or height of block B (see Figure 6). This type of graph can also be used for other analyses of the layout of a document.

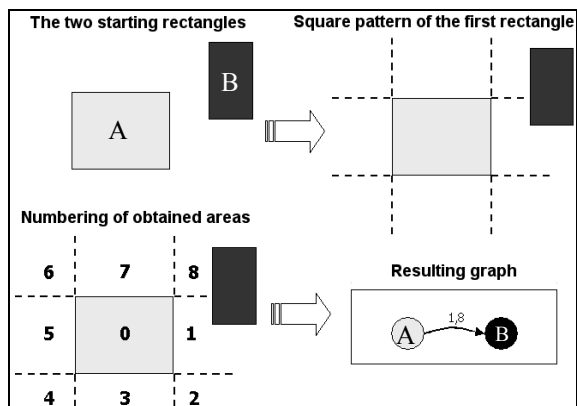


Figure 6. Areas of influence and associated graph

From the directed graph we build an undirected graph, where an edge is present between two nodes if a significant share of the first is in area x of the second and a significant share of the second is in area y (with $x-y=0 \pmod{4}$) of the first. Line cliques (fully connected components) are extracted from this graph based on areas 1 and 5. Likewise, we get a new undirected graph and the associated column cliques based on areas 3 and 7.

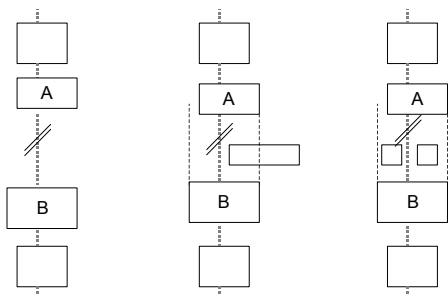


Figure 7. Cases where column candidates are split

Blocks of text that are aligned with members of a clique are then added to the clique. The vertical alignment (left, center, right) counts for column cliques and the horizontal alignment (top, middle, bottom) for line cliques.

The resulting line and column candidates do not take into account information concerning the spacing between blocks and direct neighborhood. To correct the results we split a candidate when the distance between two consecutive blocks is beyond a threshold (depending on characteristics of the candidate) and when “intruders” not belonging to the candidate are found (see Figure 7). Finally, a new undirected graph is created, where an edge is present between two blocks of text if they belong together to a same line or column. The connected components of this graph are our table candidates.

Figure 8 shows the table candidates obtained in a document. A different gray level is associated to each table candidate. Only one table will not be detected because of the lack of alignment of the text blocks.

By the analysis of lines and columns can be identified the detailed internal structure of the tables and rejected candidates containing a single line of text or badly misaligned lines in one column.

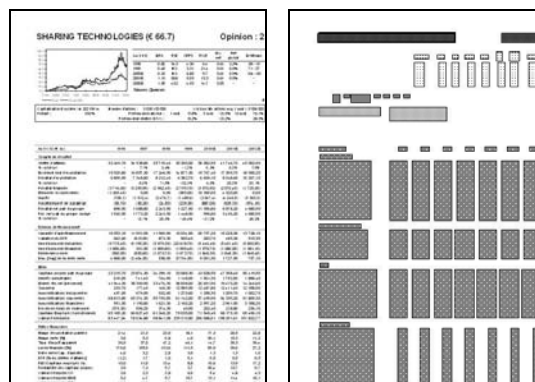


Figure 8. Detection of table candidates from text

6. Conclusion

We proposed a flexible representation scheme for tables, together with new methods for the detection of tables in a wide-ranging class of documents and the extraction of their structure. These methods can be further refined in order to complete the reconstruction of the entire logical structure of complex tables. Knowledge of the graphic marks must be combined to the information concerning regularities in the layout of text components.

References

- [1] N. Vincent, M. Crucianu, R. El Ayadi, C. Faure, M. Giba, M. Venet, “From Exchange Format to High Level Document Description”, *DLIA 2001*, Seattle, Washington, USA.
- [2] Y. Wang, I.T. Phillips and R. Haralick, “From Image to SGML/XML Representation: One Method”, *DLIA'99*, Bangalore, India.
- [3] W3C, “XHTML 1.1 – Module-based XHTML”, *W3C Recommendation*, 2001, <http://www.w3.org/TR/xhtml11/>
- [4] N. Walsh, “XML Exchange Table Model Document Type Definition”, *Technical Memorandum TR 9901:1999*, OASIS, 1999, <http://www.oasis-open.org/specs/tm9901.html>
- [5] M. Crucianu, R. El Ayadi, N. Vincent, “On the representation of tables in XML”, *Internal Report 244*, Laboratoire d’Informatique, University of Tours, 2001.
- [6] F. Cesarini, S. Marinai, L. Sarti, G. Soda. “Trainable table location in document images”. *ICPR'02*, Québec, pp. 236-240.
- [7] T.G. Kieninger, A. Dengel, “A paper-to-HTML table converting system”, *DAS'98*, Nagano, Japan, 1998.
- [8] J. Hu, R. Kashi, D. Lopresti, G. Wilfong, “A system for understanding and reformulating tables”, *DAS 2000*, Rio de Janeiro, Brazil, 2000, pp. 361-372.