2009 10th International Conference on Document Analysis and Recognition

PDF-TREX: An Approach for Recognizing and Extracting Tables from PDF Documents

Ermelinda Oro
Department of Computer and Systems Science
University of Calabria
Via P. Bucci Cubo 41/C, Rende (CS), Italy
linda.oro@deis.unical.it

Massimo Ruffolo
High Performance Computing and Networking Institute
Italian National Research Council
Via P. Bucci Cubo 41/C, Rende (CS), Italy
ruffolo@icar.cnr.it

Abstract

This paper presents PDF-TREX, an heuristic approach for table recognition and extraction from PDF documents. The heuristics starts from an initial set of basic content elements and aligns and groups them, in bottom-up way by considering only their spatial features, in order to identify tabular arrangements of information. The scope of the approach is to recognize tables contained in PDF documents as a 2-dimensional grid on a Cartesian plane and extract them as a set of cells equipped by 2-dimensional coordinates. Experiments, carried out on a dataset composed of tables contained in documents coming from different domains, shows that the approach is well performing in recognizing table cells. The approach aims at improving PDF document annotation and information extraction by providing an output that can be further processed for understanding table and document contents.

1 Introduction

Tables are visual oriented arrangements of information widely used in many different domains as a ways to present and communicate complex information to human readers. Therefore, automatically extracting information contained in tables and storing them in structured machine-readable form is of paramount importance in many application fields. However, tables have many different layouts and are mostly contained in semi-structured and unstructured documents having various internal encodings (e.g. HTML, PDF, flat text). For these reasons table recognition and extraction is a very challenging problem that poses many issues to researchers and practitioners in defining effective approaches. As described in Section 2 a significant body of work concerning approaches and systems for table recognition and extraction (TREX) is currently available in literature. However, currently there is not a winning approach. An important limitation of TREX, as a "per se" research field, is the lack of available standard datasets that hinders objective and complete comparisons among existing approaches. Furthermore, TREX approaches for PDF documents suffer from the following limitations: (i) there is not a clear definition of exploited document features; (ii) handled objects are somehow high level document structures (e.g. text fragments) and graphical feature but is not clear how they are obtained and combined. This paper presents PDF-TREX an heuristic-based approach for table recognition and extraction from PDF documents. The PDF-TREX approach considers a PDF document as a Cartesian plane on which are placed *content elements* contained in 2-dimensional *visualization areas*. The heuristics, in order to recognize tables contained in a document, aligns and groups, in bottom-up way, content elements by exploiting only spatial relationships existing among them. So, the approach does not requires: (i) linguistic or domain knowledge; (ii) graphical metadata and ruling lines (iii) predefined table layouts. The scope of the approach is to extract a table as a set of cells equipped by 2-dimensional coordinates that form a grid on a Cartesian plane. The output obtained by PDF-TREX is an XML representation of extracted cells that can be further processed for understanding table contents and improving information extraction from PDF documents. For instance, in [6] is described a reasoning method that exploits a logic-based representation of extracted cells and domain knowledge for storing table contents as instances of a knowledge base. In the paper, to better show how the approach works, a running example is considered. PDF-TREX performances are evaluated by using a dataset composed by tables contained in PDF documents coming from different domains. This dataset aims at contributing to the definition of standard datasets in the TREX field. Experimental results show

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.12

that the PDF-TREX approach is well performing in recognizing table cells. The remainder of the paper is organized as follows. Section 2 briefly presents related work. Section 3 describes in detail the PDF-TREX approach. Section 4 shows experimental results and concisely discusses them. Finally, section 5 concludes the paper and sketches future work directions.

2 Related Work

Table recognition and extraction (TREX) has recently received as considerable attention as it can be considered a "per se" research field. Therefore, a large body of work concerning approaches and systems aimed at recognizing and extracting tables from documents having different internal encodings is currently available in literature. A survey due to Zanibbi et al. [10], provides a detailed description of already existing TREX approaches and systems. TREX approaches can be classified by using different criteria in: (i) *predefined layout-based*, *heuristics-based*, and *statistical-based*, by considering the adopted recognition and extraction method, as described in Wang [8]. These approaches use different knowledge engineering techniques founded on machine learning and statistics. (ii) *table recognition-oriented*, *labeling-oriented* and *cell classification-oriented*, by considering which kind of problem approaches aim at solving, as described in Pivk et al. [7]. (iii) *HTML-oriented*, *PDF-oriented* and *flat text-oriented* While many HTML-oriented TREX approaches and systems are available (e.g. [2]), just few approaches that deal with PDF and flat text document formats have been proposed in literature (e.g. [1, 3, 5, 9]). Furthermore approaches working on PDF and flat text documents are less performing than those working on HTML documents. This is due to the highly unstructured nature of PDF and flat text documents that pose complex issues in defining TREX methods. Yildiz et al. [9] describe a system called *pdf2table* based on an heuristic approach that performs two main tasks: table recognition, in which information organized in tabular structure are recognized, and table decomposition in which recognized elements are assigned to a table model. Authors provide an experimental evaluation of the approach based on a dataset composed by 150 documents, but the dataset is not available so, a direct comparison with such an approach is impossible. Hassan et al. [3] propose a method for detecting tables in PDF files. They group tables into three categories: tables with both horizontal and vertical ruling lines, tables which contain only horizontal ruling lines, and tables where ruling lines are absent. Authors present an experimental evaluation of the approach performed on a dataset obtained by searching the web by Google. But an actual comparison with presented results is not possible because documents retrieved by this method changes rapidly. Furthermore, this approach heavily exploit ruling lines so it is essentially not able to identify tables not embraced in a graphical grid. Often, in real situations, ruling lines do not border table cells but even groups of rows.

3 Tables Recognition and Extraction from PDF Documents

Tables recognition by human readers is a visual process based on spatial relationships existing among *content elements* (CE). In such a process alignment among groups of CE is observed in order to recognize final tables. The PDF-TREX approach starts from an initial set of *basic* CE, and by using a heuristic algorithm that reproduces the human behavior, aligns and groups them in bottom-up way in order to identify tabular arrangements of information. The scope of the approach is to recognize tables contained in PDF documents as 2-dimensional grids on a Cartesian plane and extract them as a set of cells equipped by 2-dimensional coordinates. In order to describe how the PDF-TREX approach works the following preliminary definitions are required.

Definition 1. Let consider a document page as 2-dimensional Cartesian plane, a *visualization area* (VA) is a rectangular area of a document page represented by the following 4-tuple $\alpha = \langle x^t, y^t, x^b, y^b \rangle$ where (x^t, y^t) and (x^b, y^b) are the Cartesian coordinates of two opposite vertices (top-left and bottom-right).

Definition 2. Let the set of strings Σ defined over the alphabet A a *content element* (CE) is a 2-tuple of the form $\varepsilon = \langle \sigma, \alpha \rangle$ where: (i) σ is a string in Σ; (ii) α is the VA related to σ.

Let be $\alpha_1 = \langle x_1^t, y_1^t, x_1^b, y_1^b \rangle$ and $\alpha_2 = \langle x_2^t, y_2^t, x_2^b, y_2^b \rangle$ two VAs, the following definitions hold.

Definition 3. α_1 and α_2 are said to be *horizontally overlapped* when by projecting α_1 from left to right along the Y-axis, α_2 is contained in (or intersect) such a projection. The following inequalities formally describe horizontal overlapping conditions: (i) $y_2^t \leq y_1^t \leq y_2^b \leq y_1^b$; (ii) $y_1^t \leq y_2^t \leq y_1^b \leq y_2^b$; (iii) $y_1^t \leq y_2^t \leq y_2^b \leq y_1^b$; (iv) $y_2^t \leq y_1^t \leq y_1^b \leq y_2^b$.

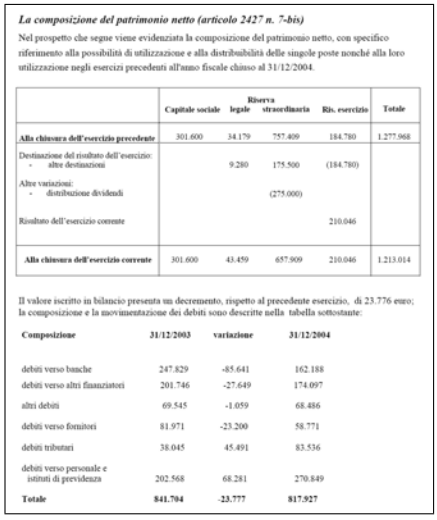
Definition 4. Let $\delta = \min((y_1^b - y_1^t), (y_2^b - y_2^t))$ The *horizontal overlapping ratio* between α_1 and α_2 is respectively one of the following expressions (depending from overlapping conditions presented in definition 3): (i) $\rho_h(\alpha_1, \alpha_2) = \frac{y_2^b - y_1^t}{\delta}$; (ii) $\rho_h(\alpha_1, \alpha_2) = \frac{y_1^b - y_2^t}{\delta}$; (iii) $\rho_h(\alpha_1, \alpha_2) = \frac{y_2^b - y_2^t}{\delta}$; (iv) $\rho_h(\alpha_1, \alpha_2) = \frac{y_1^b - y_1^t}{\delta}$.

Definition 5. When α_1 and α_2 are horizontally overlapped their *horizontal distance* is: (i) $D_h(\alpha_1, \alpha_2) = (x_2^t - x_1^b)$, if $x_1^b < x_2^t$; (ii) $D_h(\alpha_1, \alpha_2) = (x_1^t - x_2^b)$, if $x_2^b < x_1^t$. When α_1 and α_2 are not horizontally overlapped $D_h(\alpha_1, \alpha_2) = \infty$.

Vertically overlapped elements, vertical overlapping ratio and vertical distance are defined in analogous way.

3.1 The Heuristic Algorithm

The heuristic algorithm is organized in eight steps which behavior is described in the following by using as running example the PDF page shown in Figure 1 [1]. The input is assumed to be a single column document.



La composizione del patrimonio netto (articolo 2427 n. 7-bis)

Nel prospetto che segue viene evidenziata la composizione del patrimonio netto, con specifico riferimento alla possibilità di utilizzazione e alla distribuibilità delle singole poste nonché alla loro utilizzazione negli esercizi precedenti all'anno fiscale chiuso al 31/12/2004.

	Capitale sociale	Riserva legale	Riserva straordinaria	Ris. esercizio	Totale
Alla chiusura dell'esercizio precedente	301.600	34.179	757.409	184.780	1.277.968
Destinazione del risultato dell'esercizio: - altre destinazioni		9.280	175.500	(184.780)	
Altre variazioni: - distribuzione dividendi			(275.000)		
Risultato dell'esercizio corrente				210.046	
Alla chiusura dell'esercizio corrente	301.600	43.459	657.909	210.046	1.213.014

Il valore iscritto in bilancio presenta un decremento, rispetto al precedente esercizio, di 23.776 euro; la composizione e la movimentazione dei debiti sono descritte nella tabella sottostante:

Composizione	**31/12/2003**	**variazione**	**31/12/2004**
debiti verso banche	247.829	-85.641	162.188
debiti verso altri finanziatori	201.746	-27.649	174.097
altri debiti	69.545	-1.059	68.486
debiti verso fornitori	81.971	-23.200	58.771
debiti tributari	38.045	45.491	83.536
debiti verso personale e istituti di previdenza	202.568	68.281	270.849
Totale	**841.704**	**-23.777**	**817.927**

Figure 1. The Running Example Page

Elements Harvesting. In the harvesting step initial *basic* CEs that have the form described in Definition 2, are identified by accessing the PDF document, and then acquired and stored. In basic CEs: (i) the string σ is a characters sequence that do not contain blank chars; (ii) coordinates of VAs are assigned, starting from positional information contained in the PDF document, so that there are no couples of basic CEs that have both horizontal and vertical overlapping ratio greater than 0. By means of the CE idea spatial relationships existing among objects contained in a PDF document can be exploited for recognizing tabular arrangement of information. In Figure 2 are depicted basic CEs (graphically represented by rectangular box) acquired from a section of the input page. The figure also shows a visualization area (which box is highlighted by bold border) and the Cartesian plane in which its coordinates are defined. In the harvesting step *horizontal* and *vertical distance threshold values* (hT and vT) are computed. These values, used in next steps, are evaluated by analyzing white space sizes and horizontal and vertical distance distributions among visualization areas.

Lines Building. In this step *lines*, having the form described in the following definition, are built.

[1]The page has been obtained from a balance sheet of an Italian company

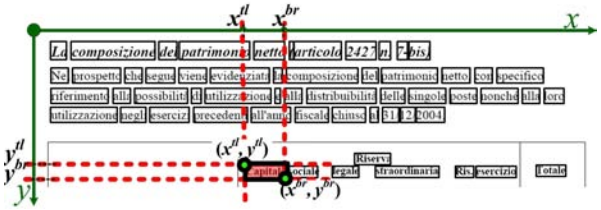


Figure 2. Basic Content Elements

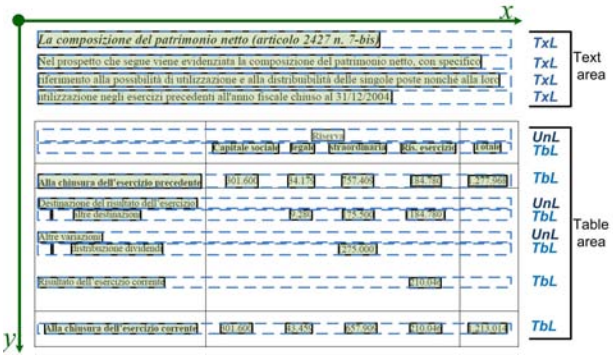


Figure 3. Elements, Segments, Lines, Text and Table Areas

Definition 6. A *line* is a 2-tuple $\lambda = \langle E, \alpha \rangle$ in which: (i) E is a set of CEs; (ii) horizontal coordinates of α correspond respectively with the minimum and the maximum horizontal coordinates of a page; (iii) vertical coordinates of α are assigned so that do not exist lines horizontally overlapped.

Basic CEs are assigned to a line when their horizontal overlapping ratio is over a given threshold[2]. It is noteworthy that CEs assigned to lines are obtained from basic CEs by adapting their vertical coordinates to those of the line that contain them. Figure 3 depicts lines (highlighted by dashed border) obtained in a section of the running example page.

Segments Building and Lines Tagging. Vertical sequences of lines can generate either text paragraphs or tables. The lines tagging step assigns to each line a tag: *text line (TxL)*, *table line (TbL)*, *unknown line (UnL)*. In order to tag lines, CEs contained in a line are grouped in *segments* having the form defined in the following.

Definition 7. A *segment* is a 2-tuple $\theta = \langle E, \alpha \rangle$ where: (i) $E = \{\varepsilon_i \mid \varepsilon_i = \langle \sigma_i, \alpha_i \rangle \wedge i = 1, ..., n \wedge \alpha_i = \langle x_i^t, y_i^t, x_i^b, y_i^b \rangle\}$ is a set of CEs; (ii) coordinates of α are: $x^t = \min_{i=1}^{n} x_i^t$, $y^t = \min_{i=1}^{n} y_i^t$, $x^b = \max_{i=1}^{n} x_i^b$, $y^b = \max_{i=1}^{n} y_i^b$.

To construct segments an agglomerative hierarchical clustering algorithm [4] is used. The algorithm works line by line. Initially each CE of a line is assigned to a clus-

[2]In experiments horizontal overlapping ratio has been set to 50%.

ter, then the algorithm agglomerates clusters until horizontal distance is lower than hT. Final clusters represent segments. A line that contains only one segment is tagged either as *text line* when the segment spans over half horizontal line length, or as *unknown line* otherwise. Whereas, a line is tagged as *table line* when it contains more than one segment. Figure 3 depicts segments (highlighted by rectangles having gray background) and line tags obtained in a section of the running example page.

Table Areas Building. This step aims at analyzing sequences of lines in order to identify table areas defined as shown in the following.

Definition 8. A *table area* is a 2-tuple $T_a = \langle L, \alpha \rangle$, where: (i) L is an ordered list of consecutive lines $L = \{l_i \mid l_i = \langle E_i, \alpha_i \rangle \wedge i = 1, ..., n \wedge \alpha_i = \langle x_i^t, y_i^t, x_i^b, y_i^b \rangle\}$ tagged only as table lines or unknown lines; (ii) vertical coordinates of α are: $y^t = \min_{i=1}^{n} y_i^t$, $y^b = \max_{i=1}^{n} y_i^b$ (iii) horizontal coordinates (x^t and x^b) of α are set by using horizontal coordinates of a VA of any line in L.

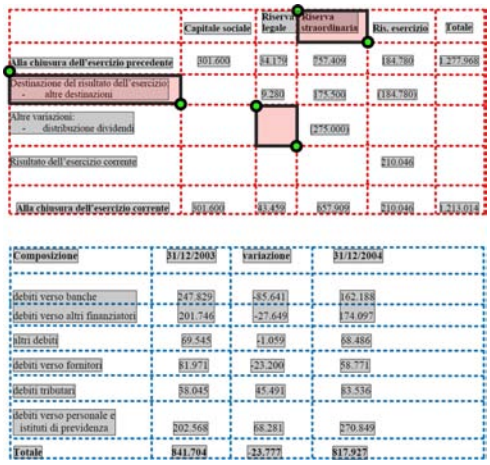


Figure 4. Blocks, Rows, Columns and the Final Cells Grid

Block and Row Building. Row building step aims at constructing table rows defined as shown in the following.

Definition 9. A *row* is a 2-tuple $r = \langle L, \alpha \rangle$ where: (i) L is a subset of contiguous lines contained in a table area; (ii) horizontal coordinates of α are set by using horizontal coordinates of related table area; (iii) vertical coordinates of α are assigned so that all rows of a table area are contiguous.

In order to recognize lines belonging to a same table row, *blocks*, defined as shown in the following, are used.

Definition 10. A *block* is a 2-tuple $\beta = \langle S, \alpha \rangle$ in which: (i) $S = \{s_i \mid s_i = \langle E, \alpha_i \rangle \wedge i = 1, ..., n \wedge \alpha_i = \langle x_i^t, y_i^t, x_i^b, y_i^b \rangle\}$ is a set of segments; (ii) coordinates of α are: $x^t = \min_{i=1}^{n} x_i^t$, $y^t = \min_{i=1}^{n} y_i^t$, $x^b = \max_{i=1}^{n} x_i^b$, $y^b = \max_{i=1}^{n} y_i^b$.

To construct blocks the same agglomerative hierarchical clustering algorithm [4] adopted for constructing segments is used. In this case, segments are the initial clusters, the algorithm agglomerates these clusters until vertical distance is lower than vT. Final clusters represent needed blocks. Normally each line constitutes a row, but when there is a block that embraces segments of a set of consecutive lines and only one line in the set is tagged as *TbL* whereas the others are tagged as *UnL*, the set of line is grouped in a single row. This behavior is very useful for recognizing rows which headers span on multiple lines as a unique logical structure. In Figure 4 rows obtained in the running example page (highlighted by dotted lines) are shown.

Column Building. Column building step aims at creating table columns defined as shown in the following.

Definition 11. A *column* is a 2-tuple $c = \langle S, \alpha \rangle$ in which: (i) S is a set of segments; (ii) horizontal coordinates of α are assigned so that all columns of a table area are contiguous; (iii) vertical coordinates of α are set by using vertical coordinates of related table area.

Columns are built by using vertical overlapping ratio among segments and distances among columns contained in table areas. Segments are assigned to the same column when they are vertically overlapped. When a segment overlaps more columns it is duplicated and assigned to each column, this trick allows to recognize column headers spanning on multiple columns.When there are columns composed by only one segment (for example column headers not overlapped with column values) horizontal distances are considered. When the distance between such a column and a consecutive column containing more than one segment is below the horizontal threshold hT the two columns are merged in a new single column. Figure 4 depicts (by using dotted lines) columns obtained in the running example table areas.

Table Building. This step generates final tables as sets of 2-dimensional cells defined as follows.

Definition 12. A *cell* is a 2-tuple $c = \langle \sigma_c, \alpha \rangle$ where: (i) α the is obtained by crossing a row and a column; (ii) let E be the set of elements embraced in α, σ_c is the string obtained by concatenating strings contained in E preserving visualization order from left to right and top to bottom.

Figure 4 shows final cells grid. It is worthwhile noting that the heuristic algorithm is able to recognize cells containing multi line row headers, null values and column headers spanning on multi column as highlighted in Figure 4 by using bold border.

Extraction. This step produces the output of the PDF-TREX approach by serializing in XML table cells recognized in previous step. The output can be further processed for fitting table layouts, understanding table contents, enabling information extraction and document annotation. For instance, the approach described in [6] allows to store table

	Precision	Recall	F-measure
Table Areas	0.8626	0.9849	0.9197
Table Cells	0.7532	0.9652	0,8461

Table 1. Experimental Results

contents as instances of ontology classes by selecting (in a semantic fashion) only information of interest.

4 Experiments and Discussion

The lack of standard datasets hinders the comparison of PDF-TREX approach performances to other existing similar approaches. This section describes results obtained by applying the PDF-TREX prototypical implementation to a dataset [3] composed of 100 PDF documents (written in different languages) coming from different domains and containing 164 tables. Documents have been analyzed by hand in order to identify table areas and table cells existing within them. Then results automatically obtained by the system prototype have been compared to those manually computed by using precision and recall measures. The evaluation of precision takes into account that table areas and cells can be merged or split. Let N_{rel}, N_{ret} and N_{rel_ret} respectively the numbers of correct table areas/cells, retrieved table areas/cells and correctly retrieved table areas/cells. When table areas/cells are merged ($N_{ret} < N_{rel\_ret}$) precision is computed by means of the following merging ratio $\frac{N_{ret}}{N_{rel\_ret}}$. When table areas/cells are split recall and precision are computed in traditional way. Recall and precision for tables areas are evaluated for the whole document, whereas table cells recall and precision are evaluated for each table. Results presented in table 1 show that the system is well performing in recognizing and extracting table areas. Regarding table cells it is worthwhile nothing that table cells recall is very high and close to table areas recall, whereas table cells precision is less than recall. This result is due to a cautious behavior of the PDF-TREX approach w.r.t. cell generation. This behavior allows to refine results by means of further reasoning on extracted table cells. In fact, to merge more spatial cells in a single logical cell is more simple than split a single spatial cell in more logical cells.

5 Conclusions and Future Work

This work presented the PDF-TREX approach for table recognition and extraction from PDF documents. The PDF-TREX approach considers a PDF document as a Cartesian plane on which are placed *content elements* contained in 2-dimensional *visualization areas*. The approach is designed for recognizing a wide variety of table layouts and does not use any graphical or linguistic document feature. It aligns and groups content elements in order to identify tabular arrangement of information. The approach is able to detect multi-line row headers and column headers spanning on multi row/column. The output of the approach is a set of 2-dimensional table cells that form a grid. Table cells can be seen as initial bricks well suited for understanding table layouts and contents by using spatial reasoning. PDF-TREX approach features contribute to improve document annotation and information extraction from PDF documents. The evaluation of system performances on a dataset composed of 100 documents and 164 tables shows that the system is well performing in recognizing and extracting table cells. Future work will focus on the extension of the approach: (i) for exploiting stylistic features and lexical information that enable to better reconstruct table cells; (ii) for constructing an object-oriented document model aimed at representing in structured way all PDF document content containers (e.g. pages, columns, paragraphs, images, tables); (iii) for enabling visual analysis of other document formats like HTML and flat text.

[3] The dataset is available at http://staff.icar.cnr.it/ruffolo/pdf-trex/dataset.zip

References

[1] Y. Aumann, R. Feldman, Y. Liberzon, B. Rosenfeld, and J. Schler. Visual information extraction. *Knowl. Inf. Syst.*, 10(1):1–15, 2006.

[2] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. In *WWW*, pages 71–80, 2007.

[3] T. Hassan and R. Baumgartner. Table recognition and understanding from pdf files. In *ICDAR*, pages 1143–1147, 2007.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[5] T. Kieninger and A. Dengel. An approach towards benchmarking of table structure recognition results. In *ICDAR*, pages 1232–1236, 2005.

[6] E. Oro and M. Ruffolo. Xonto: An ontology-based system for semantic information extraction from pdf documents. In *ICTAI*, 2008.

[7] A. Pivk, P. Cimiano, Y. Sure, M. Gams, V. Rajkovic, and R. Studer. Transforming arbitrary tables into logical form with tartar. *Data Knowl. Eng.*, 60(3):567–595, 2007.

[8] Y. Wang. *Document analysis: a table structure understanding and zone content classification*. PhD thesis, 2002. Adviser- Robert M. Haralick and Adviser-Ihsin T. Phillips Adviser- Linda Shapiro.

[9] B. Yildiz, K. Kaiser, and S. Miksch. pdf2table: A method to extract table information from pdf files. In *IICAI*, pages 1773–1785, 2005.

[10] R. Zanibbi, D. Blostein, and J. R. Cordy. A survey of table recognition. *IJDAR*, 7(1):1–16, 2004.