A Method of Table Detection in Metafiles

A. O. Shigarov, I. V. Bychkov, G. M. Ruzhnikov, and A. E. Khmel'nov

Institute of System Dynamics and Control Theory, Siberian Branch, Russian Academy of Sciences, ul. Lermontova 134, Irkutsk, a/ya 292, 664033 Russia

e-mail: shigarov@icc.ru

Abstract—A method is proposed for the detection of statistical tables that use metafiles as input data; the latter fact allows one to apply this method to documents of different formats. In this method, the table detection process is viewed as a bottom-up segmentation of a document page, i.e., segmentation from simple elements of a page to more complicated ones. The experimental evaluation of the method shows that it is efficient as applied to a wide class of statistical tables.

Key words: Document analysis and recognition, table extraction from documents, table detection.

INTRODUCTION

When solving many scientific and practical problems, one has to extract data from tables that are contained in various documents. Currently, many methods and systems for extracting tables from documents are developed that allow one to automate this process. The surveys of papers on table extraction and processing that have appeared in recent years [1–4] show a growing interest in this problem. The first step in table extraction is the detection of tables in documents, i.e., a search, on document pages, for areas that are the images of tables.

The available table detection methods use, as a rule, bitmap images or ASCII texts as input data. At the same time, it is interesting to apply a certain exchange format such as PostScript [5], PDF [6], or EMF [7] for presenting input data in table detection methods. These formats are more informative compared with bitmap images and ASCII texts because, in addition to texts and graphics, they contain font metrics of the output text as well as information on the print order of this text. This helpful information can be used for more efficient and accurate table detection. In [8], the authors suggest extracting tables from PDF files and argue that they are not aware of other methods for extracting tables from PDF files.

In the present paper, we propose a heuristic table detection method that is oriented to EMF metafiles. Unlike other exchange formats, EMF can be interpreted by GDI API [7] (a part of Windows API). This fact makes the processing of EMF a sufficiently simple and feasible procedure. In this case, documents of different formats, for example, DOC, XLS, PDF, ASCII text, and HTML can be printed into EMF metafiles. It is assumed that the tables in the documents printed into metafiles are not bitmap inclusions. Note that we are not aware of the existence of table extraction systems and methods oriented to metafiles.

1. SPECIFIC FEATURES OF STATISTICAL TABLES

The complexity of table detection is largely attributed to the great diversity of possible ways of table representation. Many of the existing table detection methods are oriented to various features of tables that are usually determined by the standards and conventions adopted in a certain domain. The method proposed in the present paper is oriented to the features of the so-called statistical tables. These tables are used in the state statistical reports of Russia, the United States, the European Union, China, and Japan, as well as in financial reports of various companies. Figure 1 shows an example of a statistical table with the description of its basic elements.

An ordinary statistical table has a head, a stub, a body, and may have overcuttings (cut-ins) inside the body. The headers of the columns of such a table may form a hierarchy, the spanning headers always being over the corresponding nested headers. Moreover, such a table may have a full, partial, or no ruling. The ruling of a table may be formed either by graphic primitives (lines, boxes) or by the characters of pseudo-

Precision and recall of the method

Format	Number of tables	Precision	Recall
PDF	132	84.1%	96.2%
DOC	248	80.9%	91.9%
XLS	45	93.0%	88.8%
HTML	15	87.5%	93.3%

Fig. 1. Example of a statistical table.

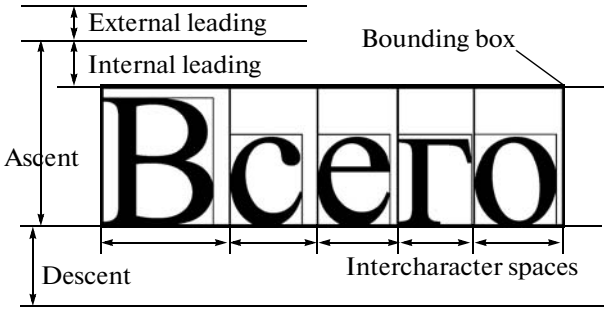

Fig. 2. Example of a text element.

graphics and some other characters; in the latter case, we will say that a table has a text ruling.

2. DATA OBTAINING FROM METAFILES

To print documents into metafiles, one can use a virtual EMF printer. Each metafile obtained by printing a document corresponds to one page of the document. In metafiles, text output instructions correspond to records of the types EMR_EXTTEXTOUTW and EMR_SMALLTEXTOUT [7]. Rule output instructions often correspond to records of the type EMR_BITBLT [7]. Using the context of a given metafile, the data of these records are interpreted according to the coordinate spaces and the mapping modes of the metafile: one determines the positions of the text output, intercharacter spaces, and some font metrics. In this case, the records corresponding to the instructions that print a text outside a page area or print a text in the same color as the background color of the area that bounds this text are ignored.

Using the metafile processing technique described, for each record one can obtain one or several structures, called text elements, which determine individual sequences of nonblank characters of this text. Figure 2 shows an example of a text element.

Each text element contains three sets: (1) characters, (2) intercharacter spaces, and (3) font metrics (external and internal leadings, ascent and descent, font pitch, and width of space character), as well as a bounding box. In addition, as a result of metafile processing, one obtains rules (ruling lines) from the records corresponding to graphic output instructions.

The data obtained are subject to preprocessing; as a result, a text ruling is eliminated from the text and is combined with the remaining ruling. Moreover, words are recovered (by a word is meant a sequence of consecutive nonblank characters) because different parts of the same word may correspond to different text elements, and, conversely, several words may occur in the same text element. After preprocessing, most text elements correspond to individual complete words.

3. TABLE DETECTION ON A DOCUMENT PAGE

In [9], the authors suggest a table detection method oriented to bitmap images that employs structures similar to the text elements considered in the present paper. These structures, called connected components, correspond to individual words and, just like text elements, have bounding boxes. The authors of [9] suggest that connected components should be combined into structures, called word blobs, provided that these components are situated in the same line of a text and the spacing between these components does not exceed a certain threshold. As a result, a text line will most likely contain a single word blob structure, while a row of a table will most likely contain a few such structures. This assumption is used in order to identify rows of a table and to distinguish them from the text lines.

This idea underlies the method proposed here. Note that, in the method of [9], the detection process uses too simplified assumptions on the arrangement of several tables on a page; for example, it is assumed that there should be empty lines or text lines between the tables; otherwise the detection will be inaccurate.

In the method proposed here, we construct the table detection process as a bottom-up segmentation of a page: from simpler elements of the page to more complicated ones. First of all, closely spaced text elements that are not separated by rules are grouped into larger structures, called text blocks, which are a certain analog of the word blob structures in the method of [9]. Figure 3 illustrates the formation of text blocks from text elements on a page.

Further, text blocks are grouped into lines so that the text blocks belonging to the same line are in the transitive closure of the relation “the intersection of the projections of two text blocks is nonempty.” Thus, the bounding boxes of the lines do not intersect.

Then, a white space (i.e., an area free of text blocks) inside a bounding box of each line is segmented; in this process, vertical gaps between text blocks are distinguished among the segments, for example, as is shown in Fig. 4. Moreover, the algo-

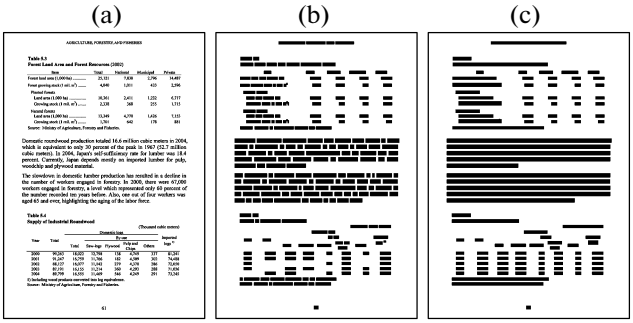

Fig. 3. Example illustrating the formation of text blocks; (a) original page, (b) bounding boxes of text elements, and (c) bounding boxes of text blocks.

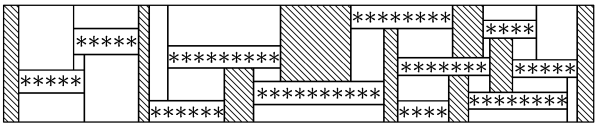

Fig. 4. Example illustrating the segmentation of the row space; dashed boxes are vertical gaps.

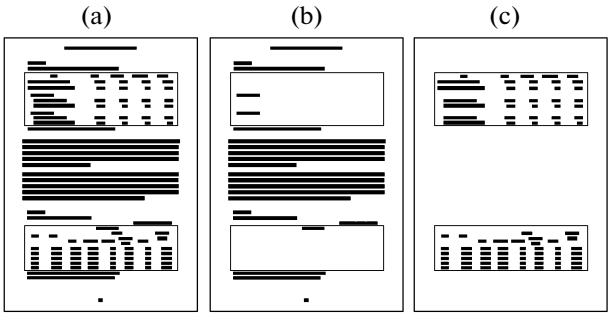

Fig. 5. Example illustrating the marking of tabular regions on a page; (a) original page, (b) lines that are not rows of tabular regions, and (c) rows of tabular regions.

rithm tries to reconstruct empty lines using the font metrics of the text elements.

Then, structures called tabular regions are formed. Each tabular region includes a sequence of consecutive text lines. A line belonging to a tabular region must satisfy the following conditions: (a) it must contain at least two text blocks, (b) the width of the white space of the line with respect to its total width must not exceed a predetermined threshold, and (c) the lower boundary of any vertical gap between the lines must coincide with the lower boundary of its bounding box. Figure 5 illustrates how the lines satisfying the conditions listed above are marked out on a page.

In addition, any two lines from the same tabular region must jointly satisfy the condition under which each vertical gap from the upper line should correspond to at least one vertical gap from the lower line, such that the upper boundary of this gap coincides with the upper boundary of the lower line, while the intersection of the projections of these two vertical gaps onto the axis X exceeds a predetermined threshold. Figure 6 shows a fragment of the page with lines belonging to the same tabular region.

When searching for sequences of lines that constitute tabular regions, the lines of a page are passed top-down. As soon as such a sequence is detected, its lines are removed from further searching; thus, the tabular regions obtained do not have common lines. Then, for each tabular region, a set of its own vertical gaps is formed from the vertical gaps of its lines.

The tabular regions obtained may be either tables or parts of tables; they also may be a text with tabular layout. Tabular regions that constitute the same table correlate with each other with respect to the positions of the projections of their vertical gaps onto the axis X, as, for example, the tabular regions shown in Fig. 7.

This feature is used in the method proposed for combining tabular regions into tables and determining the table boundaries. Here the width of the white space of any line situated between the text regions of the same table should not exceed a predetermined threshold. Moreover, the number of consecutive empty lines that may be situated between the tabular regions of the same table is also bounded by a predetermined value. Figure 8 illustrates the results of detecting tables by our method.

In some rare cases, the boundaries of several tables detected by the method described may intersect (i.e., the tables may have common rows). Note that a sufficiently efficient automatic separation of common rows between intersecting tables requires the analysis and interpretation of the text contents of these tables. In the method proposed here, we assume that the boundaries of intersecting tables must be determined by a user.

Year	Total	By use				Imported logs[1)]
		Total	Saw-logs	Plywood	Puip and Chips	Others	
2000	99.263	18.022	12.798	138	4.749	337	81.241
2001	91.247	16.759	11.766	182	4.509	302	74.488

Fig. 6. Example of rows that belong to the same tabular region; the rows are framed by rectangles; vertical gaps of the rows are crosshatched.

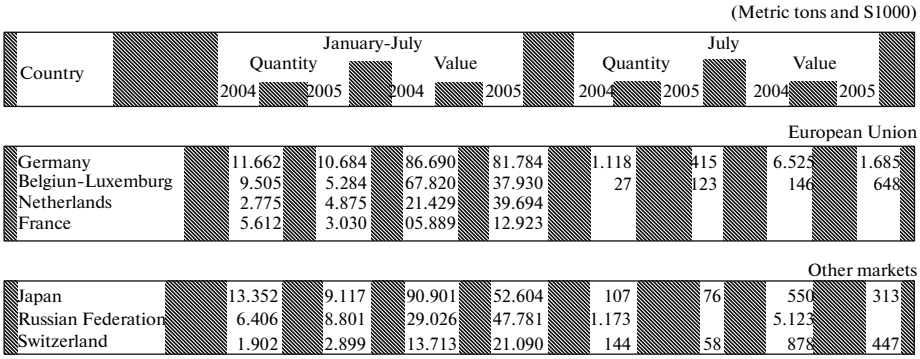

Fig. 7. Example illustrating the arrangement of tabular regions on a page; tabular regions are framed by rectangles; vertical gaps in tabular regions are crosshatched.

4. EXPERIMENTAL EVALUATION

The experimental evaluation of the method was carried out according to the criteria proposed in [10]. A table is assumed to be correctly detected if at least its body is correctly detected; i.e., each row in the body of the table is identified as a part of this table, while each row that does not belong to this table is not erroneously identified as a row of its body. We used two criteria to evaluate the efficiency of the detection method: precision, which is the percentage ratio of the number of correctly detected tables to the total number of detected tables, and recall, which is the percentage ratio of the number of correctly detected tables to the total number of available tables.

As experimental data, we used the state statistical reports of Russia ("Regions of Russia: Socioeconomic Indices 2002"; "Agriculture of the Irkutsk Region in 1993–1998," etc.), the United States ("Tobacco: World Markets and Trade 2005," etc.), the European Union ("Eurostat Yearbook 2006–2007"), Japan ("Statistical Handbook of Japan 2007"), as well as the financial reports of various companies ("Boeing Co., Annual Report 2006"; "OJSC Aeroflot–Russian Airlines, Consolidated Financial Statements for the Year Ended December 31, 2006"; "OAO AK Transneft', Consolidated Financial Statements for the Year Ended December 31, 2006"; etc.). These documents were represented in PDF, DOC, XLS, and HTML formats.

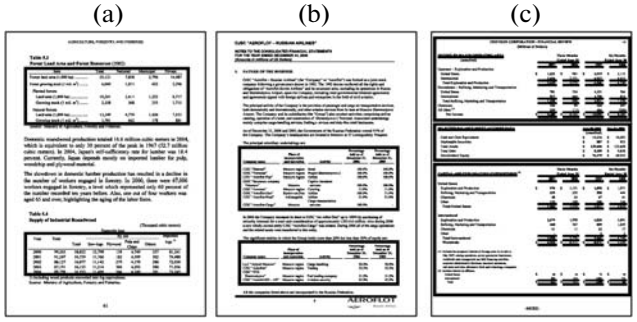

Fig. 8. Example illustrating the result of detection; rectangular regions identified as tables are framed by rectangles on pages.

Altogether, we processed 345 pages, which contained 440 tables. These pages also contained texts in tabular form and figures with text captions. The table shows the measured precision and recall for each format.

The experimental evaluation shows that this method can be applied to detecting statistical tables in documents in different formats. Note that the precision of the method can be improved by carrying out table segmentation (partition of a table into individual cells) and a functional analysis of the tables (i.e., the determination of the role of cells in a table).

CONCLUSIONS

Statistical tables have essential similarity in the arrangement of their components. This similarity has allowed us to make some assumptions on these tables and formulate the heuristics used by the table detection method proposed. The use of metafiles as data sources allows one to apply this method to documents represented in different formats (for example, PDF, DOC, XLS, HTML, etc.).

Based on the method proposed, we developed a system for table extraction from documents of different formats that detects and segments tables in documents represented as metafiles. The method can be used for constructing a table extraction system aimed at automatic transformation of tables from documents into a relational view.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research (project no. 08-07-00163-a) and by the program "Leading Scientific Schools" of the President of the Russian Federation (project no. NSh-1676.2008.1).

REFERENCES

1. A. C. Costa Silva, A. M. Jorge, and L. Torgo, "Design of an End-to-End Method to Extract Information from Tables," Int. J. Doc. Anal. Recognit. **8** (2), 144–171 (2006).
2. D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-Processing Paradigms: A Research Survey," Int. J. Doc. Anal. Recognit. **8** (2), 66–86 (2006).
3. D. Lopresti and G. Nagy, "A Tabular Survey of Automated Table Processing," *Lect. Notes Comput. Sci. 1941* (Springer, 2000), pp. 93–120.
4. R. Zanibbi, D. Blostein, and J. R. Cordy, "A Survey of Table Recognition: Models, Observations, Transformations, and Inferences," Int. J. Doc. Anal. Recognit. **7** (1), 1–16 (2004).
5. *PostScript Language Reference*, 3rd ed. (Addison–Wesley, 1999).
6. *PDF reference. Adobe,* 5th ed.
7. Microsoft Developer Network, Available from http://msdn.microsoft.com.
8. T. Hassan and R. Baumgartner, "Table Recognition and Understanding from PDF Files," in *Proc. of the 9th Int. Conf. on Document Analysis and Recognition (ICDAR 2007), Morretes, Sept. 23–26* (IEEE Computer Society, 2007), pp. 1143–1147.
9. S. Mandal, S. P. Chowghury, A. K. Das, and B. A. Chanda, "A Simple and Effective Table Detection System from Document Images," Int. J. Doc. Anal. Recognit. **8** (2), 172–182 (2006).
10. J. Hu, R. Kashi, D. Lopresti, and G. Wilfong, "Medium-Independent Table Detection," in *Document Recognition Retrieval VII* (IS\&T/SPIE Electronic Imaging, San Jose, 2000), pp. 291–302.