Locating Tables in Scanned Documents for Reconstructing and Republishing

MAC Akmal Jahan  
Post Graduate Institute of Science  
University of Peradeniya  
Peradeniya, Sri Lanka

Roshan G Ragel  
Department of Computer Engineering  
University of Peradeniya  
Peradeniya, Sri Lanka

Abstract — Pool of knowledge available to the mankind depends on the source of learning resources, which can vary from ancient printed documents to present electronic material. The rapid conversion of material available in traditional libraries to digital form needs a significant amount of work if we are to maintain the format and the look of the electronic documents as same as their printed counterparts. Most of the printed documents contain not only characters and its formatting but also some associated non text objects such as tables, charts and graphical objects. It is challenging to detect them and to concentrate on the format preservation of the contents while reproducing them. To address this issue, we propose an algorithm using local thresholds for word space and line height to locate and extract all categories of tables from scanned document images. From the experiments performed on 298 documents, we conclude that our algorithm has an overall accuracy of about 75% in detecting tables from the scanned document images. Since the algorithm does not completely depend on rule lines, it can detect all categories of tables in a range of scanned documents with different font types, styles and sizes to extract their formatting features. Moreover, the algorithm can be applied to locate tables in multi column layouts with small modification in layout analysis. Treating tables with their existing formatting features will tremendously help the reproducing of printed documents for reprinting and updating purposes.

Keywords: OCR-optical character recognition, table detection, format preservation

I. INTRODUCTION

A pool of knowledge base depends on the source of learning resources, which can vary from ancient printed documents to present electronic material. Traditional libraries play a significant role in the dissemination and safekeeping of such material. The rapid conversion of material available in traditional libraries to digital form needs a huge amount of manual effort, if we are to maintain the format and the look of the electronic documents as same as their printed counterparts. Therefore, proper digitization of printed documents plays a significant role in building digital libraries. Modifying, re-printing and searching are some important functions beyond mere digitization to maintain the digital library continuously. Therefore, we can conclude that the requirement of digitization here is converting printed documents into editable digital format, while preserving both the content and the format of the documents.

In recent past, the OCR technology worked well in the digitization process of scripts in different languages.

Typically, most of the OCR approaches focus only on character recognition of the script and it advances to focus on extraction of format features such as font size, plain text, italics, bold with different font sizes. Typical process of converting printed documents into an editable format is performed by scanning documents to images and converting them to editable text by using OCR techniques. Layout analysis, character recognition and language modelling are the important processes in OCR for a particular script. However, most of the printed documents contain not only characters but also some associated non text objects such as tables, charts and graphics in an image format. Most of the existing OCR techniques face challenges in detecting these kinds of objects during the digitization process of printed documents. Particularly presence of such objects affects the layout analysis, which is the initial step of the OCR process. For example, if the presence of a table in a document is not identified by OCR, there will be problems in identifying lines and figures inside the table and they will be treated as regular sentences. Practically, this would produce more erroneous results in text analysis in any type of script. In addition to the recognition process of the contents in the documents, editing process needs to identify the formats of the contents to maintain the uniformity within pre-processing and post processing such as updating and reproducing the contents. Most of the OCR techniques do not concentrate on the format preservation of the content. Therefore, this will be a challenging task when reproducing printed documents for reprinting or republishing purpose.

Therefore, we need to identify a way to solve the problem of preserving and reproducing documents with features discussed above. In this paper, we focus on locating and extracting different types of tables with text portion in documents. Treating tables with their existing formatting features will do a great job when reproducing the documents for future need.

The rest of the paper is organized as follows. In Section II, we present a literature survey on table detection and recognition and in Section III, we present our methodology. In Section IV, we present our experimental results and a discussion on the results in Section V and we conclude the paper in Section VI.

II. LITERATURE SURVEY

Generally, tables vary in structure from regular text. They can be designed by using ruled line, off-line, decorated line or without using lines. However, the basic structure of a table does not depend on the line but in its building blocks...
such as rows, columns and fields. Tables have physical and logical structures [1]. The physical structure determines the regions of a table in a document whereas the logical structure determines the relational information of the table such as constituent parts and how they form a table. Therefore, all parts in a table have both physical and logical structures, which are used in table region detection and table structure recognition respectively. Most of the recent research on table recognition assumes that the table region is already known and the work focused only on extraction of its logical structure. Since the existing table recognition systems lacks the facility to locate tables and the other non-text objects, it affects the layout analysis of the document image. To overcome the problem, tables can be handled in two steps: a) table detection (or locating tables), and b) table recognition.

Table detection and recognition in documents can be handled by two different approaches based on the input document type such as scanned image and electronic text documents such as pdf, HTML, word and ASCII [5-7]. Recently, there are many researches focus on table recognition rather than locating the table (or table detection) [2-3], [5-11]. However, there are some work focus on table detection and not table recognition. Rarely any work cover both table detection and recognition [10, 17]. In the table detection process, there are several ways that use geometric features such as ruled lines [12,13,15,19], pixel distributions, white gaps [4], header and trailer pattern of table [21]. Most of the past works only focus on single column page layout, whereas some recent work focuses on multi column page layout [8], [16].

Namboodiri addressed the problem of table detection and recognition for online hand written documents [17] whereas Jin Chen and Daniel addressed this issue for off-line handwritten documents [18]. Laurintine and Vaida use horizontal and vertical ruling lines to identify tables and exclude non-tabular areas from the printed documents [10]. Hu et al. proposed a method, which does not depend on ruling lines and medium independent [15]. Gatos et al. [13] locate and reconstruct tables with intersection pairs, which are detected by horizontal and vertical lines’ intersection points. Tables are reconstructed by drawing the lines. Mandal et al. [4] proposed an algorithm, which assumes the presence of substantially larger gaps between columns and table fields. This system can locate the tables if they do not contain any ruling lines or which should be removed in pre-processing. However, this algorithm does not show any way to handle tables when two or more of them appear side by side. Also tables with multi line headings or heterogeneous placement of cells can result in erroneous detection. In contrast to the above methods, our approach can locate tables with their existing formats and can resolve the problem in multiline heading as well.

Harit and Bansal [21] proposed a method for table detection using header and trailer patterns. However, the pattern depends on cognitive cues rather than functional meaning. Shafait et al. [16] and Smith et al. [22] have used tab-stop detection for doing the layout analysis of document images and then used the alignment information of columns for locating the tables. This algorithm does not work when full page tables are present. Since we are focussing on all type of tables in a page and not concern about column layout analysis of the page, our algorithm can handle this type of tables easily.

Most of the existing methods focus only on table detection through several assumptions such as i) there is a specific layout such as single column or multi column and white background in document image; ii) table is constructed with horizontal and vertical ruled-line; and iii) there are no decorated lines or complex table structure.

From the overall analysis, all past work handle only specific page layout and specific table category rather than all range of tables in different categories. Therefore, we need to identify a novel way to locate all types of tables from text in scanned documents. The algorithms proposed in this paper can address detection of all categories of tables with or without ruled lines.

III. METHODOLOGY

The sample document images consist of normal text lines, headings, a wide range of tables, equations, header, footer and page numbers with different font sizes and styles. The background is white and grey coloured. The assumptions we made here are as follows:

- There are no graphics and watermarks in the document images.
- The minimum height of a table is equal to or greater than 3 regular text lines. With this assumption, most of the heading and equation parts can be eliminated from being detected as tables.
- In a standard full lengthy text line, there would be more number of characters compared to other text lines.
- The document image is single column layout and for a professional document, there is no text or image portion on left or right side of a table (that is, tables are in line with text).

The methodology as presented in Figure 1 follows a number of stages and they are: 1) data collection and pre-processing of the scanned documents; 2) threshold computation for word space and line height; and 3) extraction of the table with and without rule lines. Let us look at details of each stage in the rest of this section.

![Fig. 1: System modules](image-url)
A. Our Algorithm

The algorithm is driven from several steps of computations. Initially, all horizontal lines in the image were extracted using X projection profile and their heights were calculated using Y projection profile. Spaces between characters and words were computed in all lines and the line which had the most number of gaps was considered as a standard text line. The assumption we made here is if there are more number of gaps, there would be more number of characters and that would be a standard text line.

Word space is assigned in all text line by considering the maximum size of the space in each line. The threshold value for a word space in a standard text line is computed as in equation (1).

\[
WS < ws < 2\times WS
\]

(1)

Where

- \(WS\): word space
- \(ws\): threshold for word space

Similarly, line height (\(LH\)) is assigned for all lines. The threshold value for the standard line height is computed as in equation (2).

\[
LH < lh < 1.5\times LH
\]

(2)

Where

- \(LH\): line height
- \(lh\): threshold for line height

The algorithm for the table detection is given as follows:

\[
\text{for } x=1 \text{ to } N \text{ do}
\]

1. if \(LH=3\times lh\) AND number of gaps= 0
   locate table type A
   end if
2. if \(WS>ws\) AND \(LH\leq lh\)
   locate table type B and C
   end if
3. if number of gaps=0 AND \(LH<lh\) AND \(WS(x-1)> ws\) OR \(WS(x+1)>ws\)
   locate parallel rule line
   end if

\text{end for}

B. Experimental Procedure

The document pages were scanned from several documents such as books, research papers and journals into an image format such as tiff, jpg or png. Initial part of this work is to extract different categories of tables from different document images. Around 300 scanned images were considered for the testing process. The scanned documents with tables were categorized into three types: A) tables with fully and partially bounding lines; B) tables with parallel lines; and C) tables without rule line. Pre-processing of a scanned document plays a significant role in the detection process. It involves binarization, noisy border removal, and enhancement. Binarization in which colour images or grey-scale images are converted to binary image using adaptive threshold technique [23]. Dilation method is applied for the enhancement with the structuring element. The pre-processed images are applied with the algorithm to locate tables.

![Type A-Table with partially bounding lines](image1)

![Type A-Table with fully bounding lines](image2)

![Type B-Table with parallel lines](image3)

![Type C-Table without rule line](image4)
In this section, we are showing some examples for each type of tables we defined. Figures 2 and 3 depict example tables of type A with fully and partial bounding lines and Figures 4 and 5 depict example tables of type B with parallel lines. Figures 6 and 7 depict example tables of type C without rule lines.

IV. RESULTS AND DISCUSSION

The categorized tables presented in the last section are tested with the proposed algorithm.

Table 1 illustrates that the different type of table detection from tables with fully and partially bounding lines to tables without ruling lines. Category A shows the highest percentage of detection whereas table with parallel line shows lower performance in detection.

Finally, we can pass a name to Get-Command to find out if this name will be interpreted as a command and if it is, what type of command it is, alias, application, cmdlet, external script, filter, function or script. In this usage, Get-Command is like the Volt which command as standards. Let me show you what mean.

TABLE I. EXPERIMENTAL RESULT OF SCANNED DOCUMENTS

| Category of Tables                                                      | Total Number of Scanned Documents | Correctly Detected Tables |
|------------------------------------------------------------------------|-----------------------------------|---------------------------|
| Table with fully and partially bounding lines (Type A)                 | 110                               | 91                        |
| Table with parallel lines (Type B)                                     | 135                               | 91                        |
| Table without ruling lines (Type C)                                    | 53                                | 40                        |

In Cape Town, there were 1,784 child deaths under 5 years of age in 2004 (Table 2.2). The majority of the deaths occurred in the postnatal period (49.3%) and a sizable proportion in the early neonatal period (22.7%). The age distribution of the deaths in the Boland Region is shown in Table 2.3 and it is very similar to that observed in Cape Town.
Among the collected documents, 53 have tables without any ruling lines (type C), in which 40 tables have been located. Seven documents cannot be located due to the differences with font type, size and style within the page and two-column table structure in which within a row one column have multiple text lines and the other column with single data. Another 6 documents cannot be identified due to the header and footer section in which section or document title and page number with large word space between them interfere the detection process.

Similarly, out of 135 tables with parallel lines, 91 documents are detected. The main reason for not to identify these tables is the assumption of the threshold for the word space made in the algorithm such that threshold is equal to 1.5 * word space. The second reason is header and footer section in which, section or document title and page number with space between them interferes the detection process in 5 documents.

Figures 9-14 depict the output produced by our algorithm for the three different types of input documents presented in Figures 2-7.
particular document with its own format and style, this documents with their existing formats. (iii) extraction of fe
atures from headers and footers; and (ii) handling issues arise from headers and footers in a line with large space be
tween them. The algorithm can be applied to handle tables in multi-column layout with small modification in layout
analysis. Since, we have determined a local threshold for line height and word space, which depends only on the
particular document with its own format and style, this would be more advantageous in applying different layouts,
formats in document images. Moreover, any layout of document with white and grey background also can be
treated with this algorithm.

The future work of this research is to focus on (i) improving the accuracy of the system by eliminating the
issues arise from headers and footers; and (ii) handling tables with decorated lines; (iii) extraction of features from
the located tables to reconstruct and reproduce the documents with their existing formats.

V. CONCLUSIONS
In this paper, we have proposed an algorithm for locating tables from scanned documents. From the experiments,
the system has shown 75% overall detection rate for 298 documents in all different type of tables in ranges of scanned
documents. The system produces erroneous results, when the document contains header or footer section or document
title and page number in a line with large space between them. The algorithm can be applied to handle tables in
multi-column layout with small modification in layout analysis. Since, we have determined a local threshold for line
height and word space, which depends only on the particular document with its own format and style, this
would be more advantageous in applying different layouts, formats in document images. Moreover, any layout of
document with white and grey background also can be treated with this algorithm.

The future work of this research is to focus on (i) improving the accuracy of the system by eliminating the
issues arise from headers and footers; and (ii) handling tables with decorated lines; (iii) extraction of features from
the located tables to reconstruct and reproduce the documents with their existing formats.

REFERENCES
[1] R. Zanibbi, D. Blostein and J.R. Cordy, “A survey of table
recognition: models, observations, transformations, and inferences,” International Journal of Document Analysis and Recognition, vol. 7, pp.1-16, 2003.

[2] T. Kieninger and A. Dengel, “Applying the T-RECS table recognition
system to the business letter domain,” In Proc. Int. Conf. on Document Analysis and Recognition, pp. 518-522, Seattle, WA, USA, 2001.

[3] Thomas G Kieninger, “Table structure recognition based on robust
block segmentation”.

[4] S. Mandal, S. Chowdhury, A. Das, and B. Chanda, “A simple and
effective table detection system from document images,” Int. Journal on Document Analysis and Recognition”, vol.823-3), pp. 172-182,
2006.

[5] B. Yildiz K Kaiser and S. Miksch, “PDF to Table: A method to
extract table information from pdf files,” Proceeding of International conference on Artificial Intelligence (IJCAI’05), 2005, pp 1773-1775.

[6] E. Oro, M. Ruffolo, “PDF-TREX: An approach for recognizing and
extracting tables from PDF documents,” Proceeding of International conference on Document Analysis and Recognition, pp. 906-910,
2009.

[7] Jing Fang, Liangcai Gao, Kun Bai, Ruiheng Qiu, Xin Tao, Zhi Yang,
“A table detection method for multipage PDF documents via visual separators and tabular structures,” Proceeding of international conference on Document Analysis and Recognition(ICDAR’11), 2011

[8] Y. Wang, R. Haralick, and I. T. Phillips, “Automatic table ground
truth generation and a background-analysis-based table structure extraction method,” In Proc. Int. Conf. on Document Analysis and Recognition, pp. 528-532, Seattle, WA, USA, 2001.

[9] T. Watanabe, H. Naruse, Q. Luo and S. Sugie, “Structure analysis of
table-form documents on the basis of the recognition of the vertical and
horizontal line segments,” International conference on Document Analysis and Recognition, pp. 906-910, 2006.

[10] T. Laurentini, P. Vaida, “Identifying and understanding tabular
material in compound documents,” Proceeding of the international conference on pattern recognition, 1992.

[11] Edward A. Green, Mukki S. Krishnamoorthy, “Model-based analysis
of printed tables”.

[12] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, “Trainable table
location in document images,” In Proc. Int. Conf. on Pattern Recognition, pp. 236-240, Canada, Aug. 2002.

[13] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis. “Automatic
table detection in document images,” In Proc. Int. Conf. on Advances in Pattern Recognition, pp. 612-621, Path, UK, Aug. 2005.

[14] A.C. e Silva, “Learning rich hidden markov models in document
analysis: Table location.,” In Proc. Int. Conf. on Document Analysis and Recognition, pp. 518-522, Seattle, WA, USA, 2001.

[15] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong, “Medium- independent
table detection,” In Document Recognition and Retrieval”, VIII
(IS&T/SPIE Electronic Imaging, pp. 44–55, 2001.

[16] F. Shafait and R. Smith, “Table detection in heterogeneous
documents,” In Proceedings of the 9th IAPR International Workshop
on Document Analysis Systems, pp. 65–72, 2010.

[17] A. Namboodiri, “On-line handwritten document understanding,” Ph.D
thesis, Michigan State University, 2004.

[18] A. Chen and D. Lopresti, “Table detection in noisy off-line
handwritten documents,” In Proceedings of the 11th International
Conference on Document Analysis and Recognition, pp. 399–403,
2011.

[19] T. Kasar, B. Barlas, S. Adam, C. Chetlain and T. Paquet, “Leaning to
detect tables in scanned document images,” In Proc. Int. Conf. on Document Analysis and Recognition, pp. 399–403, 2011.

[20] T. Kasar, B. Barlas, S. Adam, C. Chetlain and T. Paquet, “Learning to
detect tables in scanned document images,” 12th International Conference on Document Analysis and Recognition, 2013.

[21] S. Mandal, S.P. Chowdhury, A.K. Das, and Bhabatosh Chanda, “A complete system for detection and identification of tabular structures from
document images,” ICIAR 2004.

[22] Gaurav Hari, Anukriti Bansal, “Table detection in document images
using header and trailer patterns,” ICVGIP, December, 2012.

[23] Ray Smith, “Hybrid page layout analysis via tab-stop detection,” 10th
International conference on Document Analysis and Design (ICDAR),
2009.

[24] B.Gatos I.Pratikakis, S.J Perantonis, “An adaptive binarization
technique for low quality historical documents,” IARP Workshop on
Document Analysis Systems (DAS 2004), pp. 102-113.