J.Natn.Sci.Foundation Sri Lanka 2016 44 (4): 367-377
DOI: http://dx.doi.org/10.4038/jnsfsr.v44i4.8019

RESEARCH ARTICLE

Reproducing tables in scanned documents

M.A.C. Akmal Jahan[1*] and Roshan G. Ragel[2]
[1] *Postgraduate Institute of Science, University of Peradeniya, Peradeniya.*
[2] *Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Peradeniya.*

Revised: 20 April 2015; Accepted: 17 March 2016

Abstract: Digitisation is a process of representing real world objects in digital format. The rapid conversion of material available in printed form to editable digital form requires a significant amount of work if we are to maintain the format and the style of the electronic documents similar to their printed counterparts. Most of the existing digitisation procedures cover only text, and they have to go beyond OCR (text only) for making the text inside objects such as tables searchable, while preserving the format and converting the objects to an editable form to modify and reprint the content, for which the processes of detection and recognition are important. In past research the table detection process mostly followed certain assumptions such as i) either having rule lines or no lines and ii) Manhattan or multi-column layout. However, the recognition process assumed that the considered tables have already been detected and fails to preserve their formatting features for future modifications and re-printing. To address these issues, we proposed a simple and fast algorithm using local thresholds for word space and line height, which locates all types of tables and extract their formatting features. From the experiments performed on 353 records, we have achieved a much higher detection ability than the earlier algorithm and is superior as it performs extended layout analysis, elimination of header-footer and bulleted-numbered sections before the reconstruction of tables. While reconstructing the extracted table, the most prominent features of the tables are preserved with the earlier formats. The algorithm has an advantage of linear complexity as it performs the conversion locally.

Keywords: Digitisation, format preservation, table detection.

INTRODUCTION

Complete digitisation of printed materials play a significant role in building digital libraries in which modifying, re-printing, and searching of context are some important tasks. An enormous amount of manual effort is required to maintain the format and appearance of electronic documents as identical to their printed documents. The focus here is converting printed documents into editable format while preserving both the content and the format of the records.

In the past, most of the optical character recognition (OCR) approaches have focused only on character recognition of the text and extraction of format features of the characters. Nonetheless, most of the printed documents contain not only characters but also associated non-text objects such as tables, charts and graphics. This leads to a challenge in detection and preservation of their features for future editing during the digitisation process. Therefore, the current digitisation has to go beyond OCR (text only) for making the text inside objects such as tables searchable, while preserving the format and making the tables editable to modify and reprint the content. Detection and recognition are the two primary processes involved when handling tables in printed documents. The table detection process mostly follows certain assumptions such as having rule lines or no lines and Manhattan or multi-column layout. Further, the recognition process assumes that the considered tables have already been detected and fails to address the preserving of their formatting features for future modifications and re-printing. Therefore, the objective of this research study was to determine a way to detect, recognise and reconstruct tables from document images to help the process of preserving and reproducing documents with features. As earlier studies have completely addressed the OCR related issues for text (Ajward *et al.*, 2010), we focused on locating different types of tables in documents and treating them with their existing structural features to reconstruct them. Therefore,

* Corresponding author (akmaljahan@fas.seu.ac.lk)

this study handles tables in three steps: i) locating tables; ii) recognition of tables; and iii) reproducing them.

Tables vary in structure from regular text and they are composed of ruled lines, decorated lines or no lines. The basic fabric of a table depends not on the line but in its building blocks such as rows, columns and fields. Tables have physical and logical structures (Zanibbi *et al*., 2003). The physical structure determines the regions of a table and is used in region detection in a document. The logical structure determines the relational information of a table such as integral parts, how they form a table, and helps in table structure recognition. Recently, many researchers have focused on table recognition rather than locating the table, or table detection (Kieninger & Dengel, 2001; Wang *et al*., 2001; Yildiz *et al.,* 2005; Watanabe *et al*., 2006; Oro & Ruffolo, 2009; Fang *et al.*, 2011). Most of them assume that the table area is already known, and their work has focused only on the extraction of its logical structure. The existing table recognition systems lack the facility to locate tables in a document and the other non-text objects for reproducing them with the existing formatting. Rarely a few works cover both table detection and recognition (Laurentini & Vaida, 1992; Namboodiri, 2004). The current trend of document analysis recently turned into detection of tables in camera-captured images with different illumination, contrast and distortion (Wonkyo *et al.,* 2015). There is no study in the recent past, which addresses the reproduction of tables with formatting.

Table detection processes can have different approaches based on the input document types such as scanned images and electronic text documents (Yildiz *et al*., 2005; Oro & Ruffolo, 2009; Fang *et al*., 2011). The features used for table detection are geometric features such as ruled lines (Hu *et al*., 2000; Cesarini *et al*., 2002; Gatos *et al*., 2005; Kasar *et al*., 2013), pixel distributions, white gaps (Mandal *et al.*, 2006), header and the trailer pattern of the table (Harit & Bansal, 2012). Most of the past studies have only focused on single column page layout, while a few recent studies have focused on multi-column page layout (Wang *et al*., 2001; Shafait & Smith, 2010). Namboodiri (2004) addressed the problem of table detection and recognition for online handwritten documents whereas Chen and Lopresti (2011) addressed this issue for off-line handwritten documents. Laurentini and Vaida (1992) used horizontal and vertical ruling lines to identify tables and excluded non-tabular areas from the printed documents. Hu *et al*. (2000) proposed a method, which does not depend on ruling lines and is document independent. Gatos *et al*. (2005) located and reconstructed tables with intersection points from horizontal and vertical lines. Mandal *et al*. (2006) proposed an algorithm, which assumes the presence of substantially larger gaps between columns and table fields. This system can locate the tables only if they do not contain any ruling lines or are already removed in pre-processing. Also, tables with multi-line headings or heterogeneous placement of cells can result in erroneous detection in this approach. Harit and Bansal (2012) proposed a method for table detection using header and trailer patterns, which depend on rational indications rather than functional meaning. Shafait and Smith (2010), and Smith (2009) have used tab-stop detection for the layout analysis of document images and then used the alignment information of columns for finding tables. Their algorithm does not work when full page tables are present.

From the overall analysis, it was noted that all the previous works handle only a certain page layout, table category with assumptions and consider some definite type of tables rather than all range of tables in different categories. Therefore, we need to identify a novel way to locate all types of tables from the text content of scanned documents with different layouts to reproduce them with their formatting features. The work recently presented by Jahan and Ragel (2014) focused on locating all the different types of tables in printed text. However, their algorithm has some limitations on handling i) header and footer section with rule lines; ii) bullets and numbering points; and iii) multiple column document layout, which finally lowers the performance of the algorithm. Moreover, their algorithm does not represent the way to reproduce a table after locating. For this reason, in this paper we extend the work presented by Jahan and Ragel (2014), and propose an extended algorithm to eliminate the issues. However, the issues identified in previous studies such as heterogeneous placement of cells and full page tables on the page, still need to be resolved. The algorithm suggested in this paper can address the detection of all categories of tables with Manhattan or multi-column layout, with, without or partial ruled lines and a way to reconstruct them. The existing algorithm (Jahan & Ragel, 2014) is improved by applying several modules to eliminate the header-footer section and bullets-numbering parts to purify the algorithm and enhance the throughput.

METHODOLOGY

The sample document images consist of several parts of the content such as text and non-text objects. The text content consists of regular text lines, headings, a full range of tables with or without rule lines, equations,

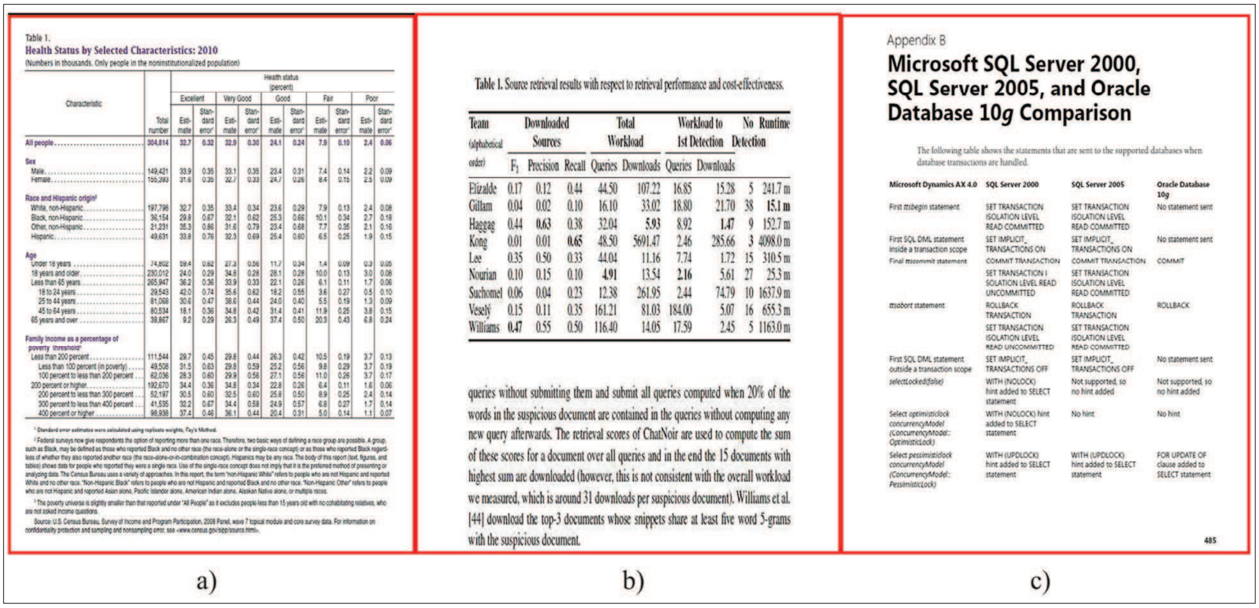

Figure 1: Different types of tables in document image: a) table with partially and fully bounding lines; b) tables with parallel lines; c) tables without rule lines

header, footer and page numbers with different font sizes, types, styles and page layouts. The background of the document is white or gray coloured, and there are no graphical objects or watermarks on the page. The document pages are with Manhattan or multi-column layouts and saved as TIFF, JPEG or PNG image format. Around 350 scanned image documents with tables were categorised into three types such as tables with: a) fully and partially bounding lines; b) parallel lines; and c) without rule lines as shown in Figure 1.

Pre-processing

Pre-processing plays an important role in the detection process. Initially, it involves the process of binarisation in which colour images or grey-scale images are converted to a binary image using adaptive threshold mechanism (Gatos *et al.*, 2004). The marginal non-textual noisy border is removed, and image enhancement is accomplished by dilation using a structuring element as shown in Figure 2.

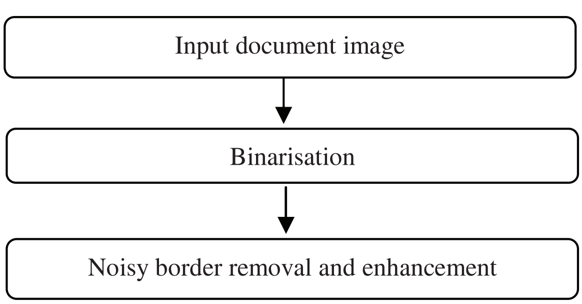

Figure 2: Pre-processing of document images

Algorithm

The new algorithm for reproducing tables is implemented by extending the previous algorithm by Jahan and Ragel (2014) as presented in Figure 3, which follows a number of stages: i) threshold computation for document layout analysis; ii) threshold computation for word space and line height; iii) eliminate header and footer sections; iv) removing bullets and numbering; v) table detection within text content; and vi) recognition and reconstruction of tables with the formatting features.

Threshold computation for layout analysis

In the initial step, all multi column layout documents have been separated and each single column in the page has been treated with the table detection algorithm. Here, we have used a threshold value for inter-column space. The algorithm for layout analysis is given in Figure 4.

Removing bullets and numbering from the content

The algorithm is improved by eliminating bullets and numbering from the content as they interrupt during threshold calculation of the standard word space. Here,

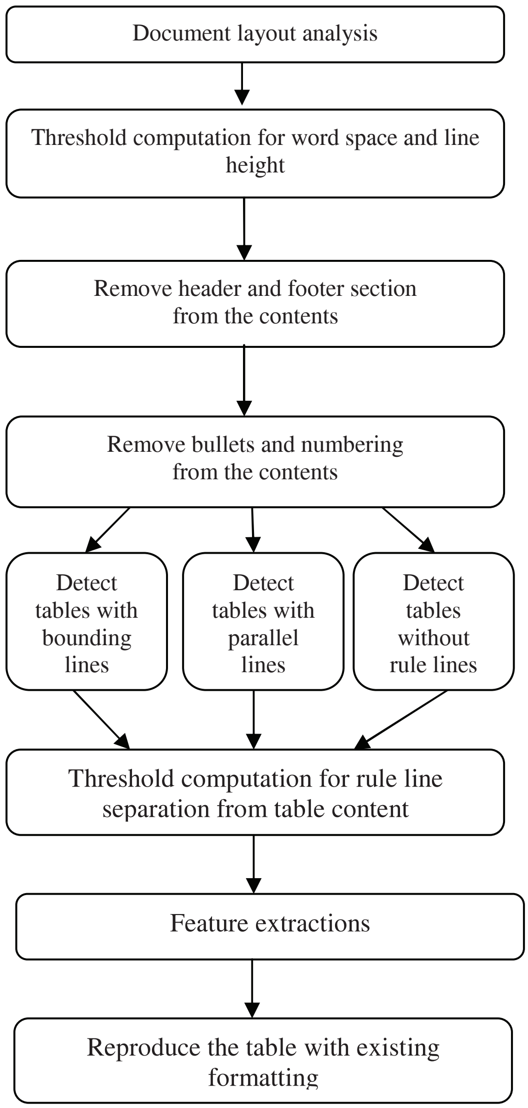

Figure 3: A complete flow of the algorithm for locating and reconstructing tables

```
for x=1 to C do
   Y(x)←Vertical projection profile
   T(x)← Total of Y(x)
   R(x)←T(x))/100
   if R(x) is greater than 5 then
      Partition(x)←Y(x)
   end if
end for
```

Where

C - Number of column

T - Sum of vertical projection profile of a column

R - Threshold value for a text column

Figure 4: Algorithmic representation of layout analysis

the first word space in every line is compared with the maximum size of the word space of that particular line. When the first word space and the maximum size of the word space in a text line are equal, it is considered as a bulleted/numbered text line and eliminated from the content of the document.

Threshold computation for word space and line height

Threshold computation for word space is computed for a standard text line by eliminating other spaces greater than or less than the word space, using the assumption of having comparatively more number of gaps in a line (have a higher number of characters) that can produce a lengthy text line, which can be a standard text line as shown in Figure 5.

Initially, the maximum size of an inter-word gap in a standard text line is considered as a word space, and the height of the standard text line is considered as line height. Subsequently from the experiments, the range of thresholds for word space and line height are determined using the standard text line by

```
WS < ws < 2*WS

LH < lh < 1.5*LH
```

`WS` - computed word space from the standard text line

`ws` - threshold for word space

`LH` - computed line height from the standard text line

`lh` - threshold for line height

Removing headers and footers from the documents

As shown in the algorithm in Figure 6, we have compared the threshold for word space and line height with the consequent three lines on top and bottom of the page. If the first line is a rule line and the other two lines are not related to tabulated data or paragraph content, then that part has been eliminated from the content page and considered as a header. Similarly, if the last line is a rule line and the other consecutive lines are not related to tabulated data or paragraph content, then that part is considered as the footer section and eliminated from the content.

Detecting tables from the documents

Part of the algorithm works on three consecutive passes such as extraction of the table with: i) fully and partially bounding lines; ii) parallel lines and iii) no ruling lines. Here we have checked the word space and the height of each line against the calculated thresholds and extracted

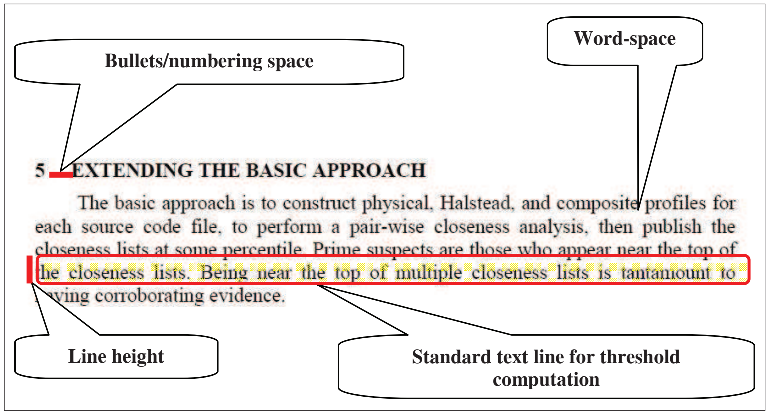

Figure 5: A sample document portion with text measurement

```
if S(1) is greater than ws and S(2) is less than or equal to ws or
  S(1) equal to 0 and S(2) is less than or equal to ws and H(1) is less than lh or
  S(1) is greater than ws and S(2)is less than or equal to ws and S(3) is less than or equal to ws then
  start=2 else
if S(1) is greater than ws and S(2) is equal to 0 and S(3) is less than or equal to ws then
  start=3 else
if S(1) equal to 0 and S(2) is greater than ws and S(3) is less than or equal to ws and H(1) is less than lh then
  start=3 else
  start=1
end if
end if
end if
if S(n) is greater than ws then
  terminate=n-1 else
if S(n) is less than or equal to ws and S(n-1) is equal to 0 and S(n-2) is less than or equal to ws then
  terminate=n-2 else
if S(n) is greater than ws and S(n-1)is equal 0 and S(n-2) is less than or equal to ws then
  terminate=n-2 else
  terminate=n
end if
end if
end if
```

Where
S - maximum length of word space in a line
ws - threshold of a word space
start - start of the text content
terminate - end of the text content
H - height of a line
lh - threshold of a line height

Figure 6: Algorithmic representation for eliminating header-footer content

the text rows that belong to tabulated data. This part of the algorithm is similar to the one presented in the work by Jahan and Ragel (2014).

Recognition of tables and reconstructing them with the formatting features

From the three types of detected tables and their contents, the tabular information is extracted using the ground-truth that the rule lines of tables can have continuous and more number of pixels than the tabulated contents in both vertical and horizontal directions.

The algorithm in Figure 7 shows the extraction of the thickness of the rule line. Similarly, all the other line features have been extracted.

```
X ← Horizontal projection profile
Size ← length of X
maxTH ← maximum range of X
minTH ← minimum of X
midTH ← mid of X
for n=1 to number_of_lines do
  for i=1 to size do
    if X (i)is not equal to minTH or midTH or maxTH and X(i+1) is equal to minTH or midTH or maxTH then
      LS(n)=i
    end if
    if X(i)is equal to minTH or midTH or maxTH and X(i+1) is not equal to minTH or midTH or maxTH then
      LE(n)=i
    end if
  end for
  LT(n)=LE(n)-LS(n)
end for
Where
 LT - Thickness of the rule line
 LS - Line start
 LE - Line end
```

Figure 7: Algorithmic representation of part of feature extraction of a table

This recognition process executes some computational steps as given below:

(a) Threshold computation for separating lines from the tabular content and line identification

Threshold values were computed from a range of maximum number of pixels to separate the rule lines with the content of the table using both horizontal and vertical projection profiles of the detected table. Then the identified lines were separated from the content.

(b) Line feature extraction

Features of the tables' rule lines were extracted according to the following to reconstruct the table in the documents: i) the number of rows and columns; ii) the number of vertical and horizontal lines; iii) the height of each row; iv) the width of each column; v) the thickness of horizontal and vertical rule lines; and vi) the height and width of each table.

(c) Reproduce the tables with the formatting features

From the extracted line features, the tables were reconstructed by passing the extracted geometric features to construct an HTML file. Therefore, the users can edit and update the tables quickly.

RESULTS AND DISCUSSION

Comparison of the new extended algorithm with the earlier algorithm given by Jahan and Ragel (2014) as shown in Figure 8 shows that all three types of tables such as with bounding lines, parallel lines and without lines show a comparatively higher number of table detection with the earlier algorithm. The tables with bounding lines show a comparatively higher detection. This type of tables can be extracted easily as it is bounded by at least a single vertical line. We compare the line heights with the threshold to eliminate the interruption of main headings and equations within the text content. Therefore, they yield the highest document detection with the vertical ruled line information. For the tables with parallel lines, the current extended algorithm shows more improvements, which significantly increases the document detection rate. Although the new extended algorithm gives many improvements, there are several factors that still influence the accuracy of detection of tables in all three categories.

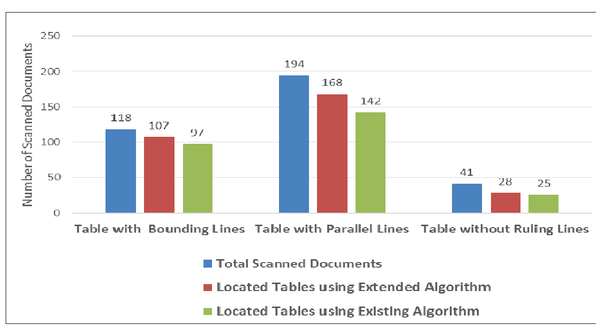

Figure 8: Comparison of table detection in scanned documents using extended and existing algorithms

The following section shows some examples of located and reconstructed tables from the printed documents. Figure 9 depicts how a reconstructable table is extracted from a scanned document. Initially, the table portion is located on the document page, and the extracted information from the table is reused to reconstruct the table without tabulated content. We consider that recognition and producing the table content (text) is out of the scope of this paper. However, such recognition can be performed with well-established techniques like OCR.

As indicated in the results, we have identified several factors that influence table detection in scanned documents. Among the 41 collected documents that contain tables without ruling lines, 13 tables have not been located in the documents.

Problem 1: The table has multiple text lines within one column and a single line in the other columns in a particular row as shown in Figure 10 (a). As a result, when comparing the threshold value with inter-column space, the tabulated text lines have not been extracted except the first text line. Here, five of the documents have not been detected.

Problem 2: The document contains tables and their contents only on the entire page that lead to the deficiency of text lines for threshold computation. Therefore, threshold computation is proceeded using the tabulated data rather than the standard text line that leads to erroneous detection. Here, two of the documents have not been identified.

Problem 3: The differences in font type and style within a single page as shown in Figure 10 (b) leads to an erroneous situation when computing a standard word space. Two of the documents could not be identified.

Problem 4: The optimisation process in threshold calculation is also one of the reasons for the erroneous situation in four of the documents. In addition, as this type of a table solely depends on the word space threshold value rather than the ruling lines of tables, it shows a challenge to produce a higher accuracy.

In the tables with parallel lines, among 194 documents, 26 documents have not been identified. Many reasons obstruct the detection of these tables using the given algorithm.

Problem 1: The table, as shown in Figure 10(e), comprises a black and grey cell structure that completely interrupt the threshold computation as our algorithm only works on grey and white background, and four documents produced error results.

Problem 2: In some cases, the table is not correctly structured as shown in Figure 10(g), which leads to the algorithm being more confused when differentiating the table component from the text elements in the document. Therefore, one of the documents could not produce the positive output for table detection.

Problem 3: In some cases, the document contains only a large table and the contents of a minimum number of extra text lines on the particular page. Such documents can lead to the deficiency of enough text lines for threshold computation. This type of tables are not detected in two of the documents.

Problem 4: The given algorithm has the facility to eliminate bullets and numbering from the text contents as shown in Figure 10(h). However, the space existing in the different numbering structure as illustrated in Figure 10(i) interrupts the threshold calculation and comparison. Therefore, these types of tables are not detected in three of the documents.

Problem 5: In some documents, the table has multiple text lines within one column and a single line in the other columns in a particular row as shown in Figure 10(a). As a result, when comparing the threshold value with inter-column space, the tabulated text lines have not been extracted except the first text line. Here, the tables have not been detected in eleven of the documents.

Problem 6: Sometimes the document can have different font types and styles. For example, a paper that has a regular text and programming codes can be differentiated by various font types, styles and inter-word gaps as shown in Figures 10(c) and 10(d). This will lead to a problem in threshold calculation as it has different word-spaces compared to the regular text line. This has caused an adverse effect on the detection of five of the documents.

From the analysis of the reasons for errors, most of the individual components of the algorithm successfully yielded the output except a few steps of the algorithm. As we have analysed the results for 194 tables with parallel lines, the performance of individual components of the algorithm is as follows.

Component 1 - almost all the documents completely give the precise output.
Component 2 - three of the documents yield errors.
Component 3 - eleven documents show error.
Component 4 - almost all the documents work completely with this component
Component 5 - eleven documents show error.
Component 6 - almost all the detected documents produce a positive output.

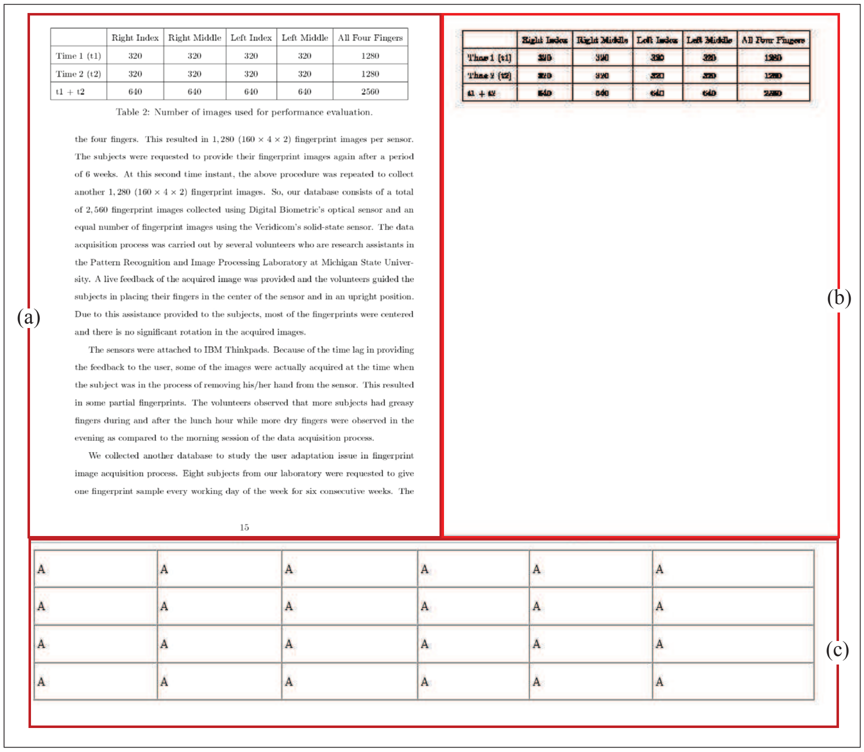

Figure 9A: Reconstruction of the table: (a) table with bounded lines; (b) located table; (c) reconstructed table

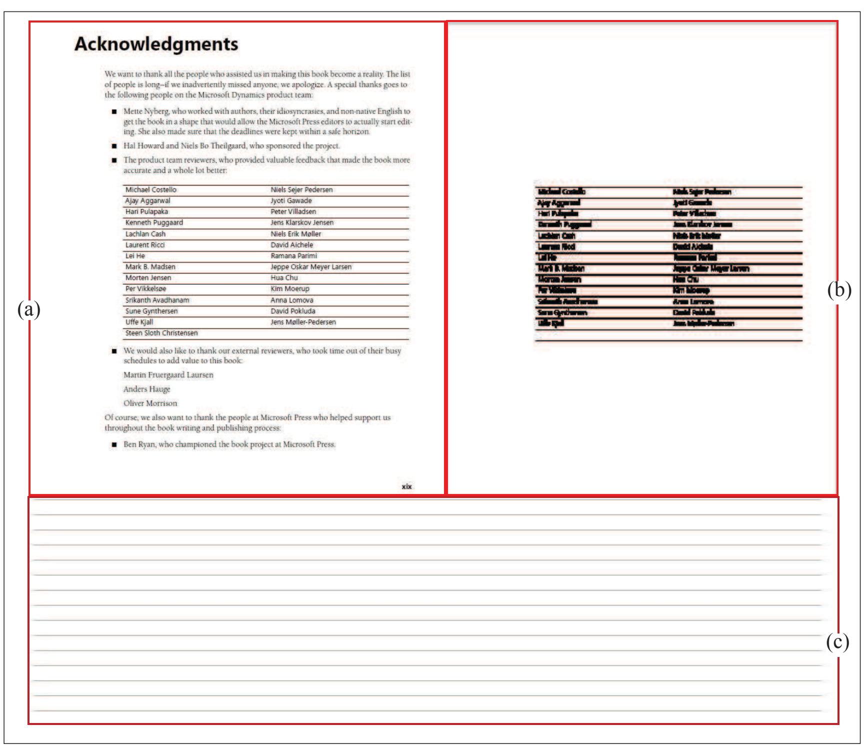

Figure 9B: Reconstruction of the table: (a) table with parallel lines; (b) located table; (c) reconstructed table

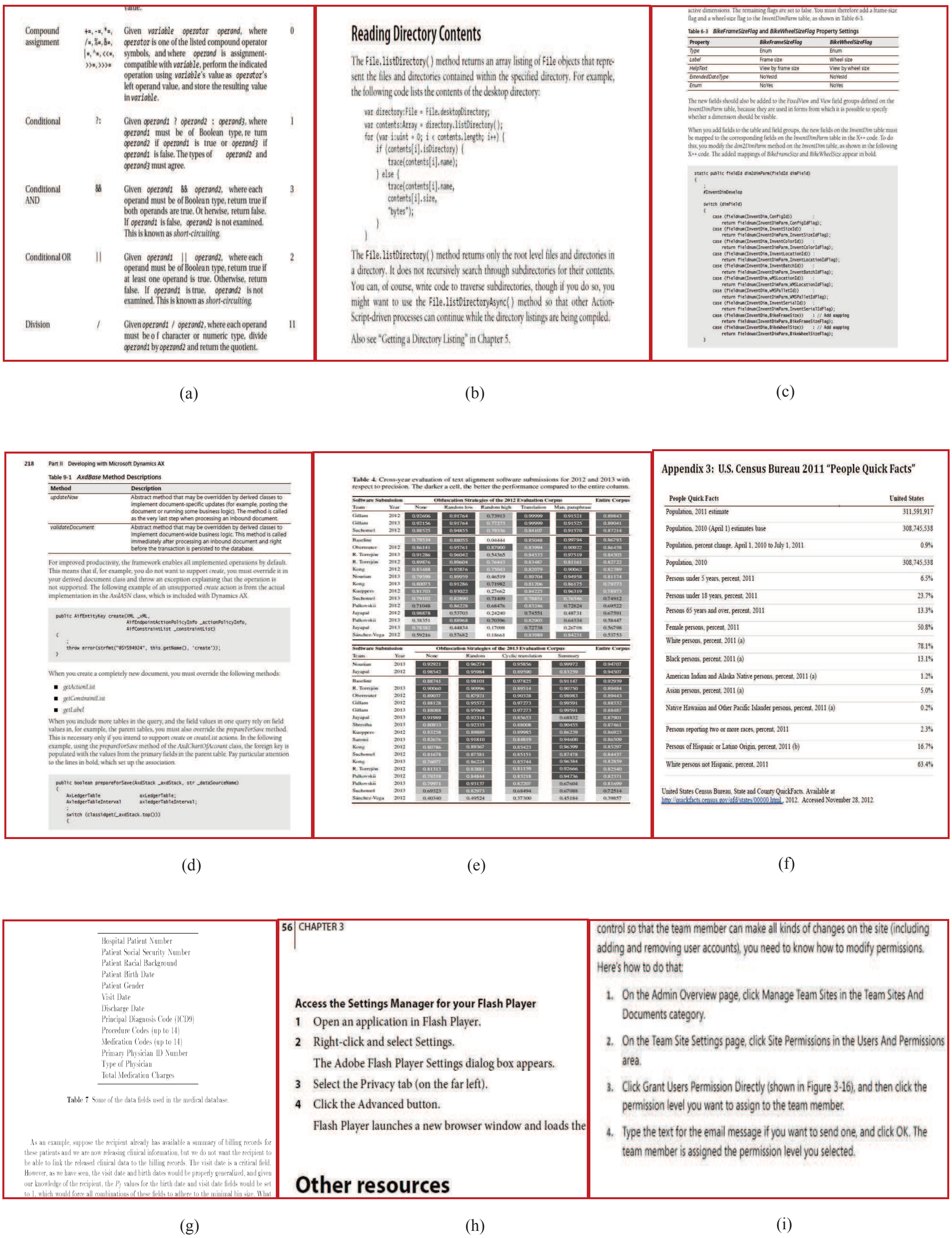

Figure 10: Factors contributing to false detection of table: (a) multiple text lines within a column in a particular row; (b) heterogeneous font types and styles on a single page; (c) different font types and styles; (d) different inter-word space; (e) grey and black table cell; (f) full lengthy table; (g) unorganised table structure; (h) numbering style 1; (i) numbering style 2

Similarly, as we have analysed the results for 41 tables without ruling lines, the performance of the individual component of the algorithm is as follows.

Component 1 - almost all the documents completely give the precise output.
Component 2 - allmost all the documents completely give the precise output.
Component 3 - thirteen documents do not show precise output.
Components 4, 5 and 6 - all the remaining documents produce positive results.

Time complexity is one of the measurements that plays a significant role when we select an efficient algorithm for a problem domain. Here, the time complexity of the proposed algorithm is linear and produces O(N) complexity, where N is the number of text or rule lines on a document page. Initially, in the first component of the algorithm-threshold computation for layout analysis, the complexity is O(1) as it only handles the vertical projection profile of the document image. The second and third step of the algorithm such as computing the number of spaces in each line, handling bullets and numbering lines and calculating threshold values for line height and word space, completely depend on the number of visible rule or text lines on the document page, and they still behave as O(N)-linear complexity. The next subsequent step of removing the header and footer is in O(1) complexity as it follows a single line of the document image. The other major stage of the algorithm locating of table behaves linearly as it still depends on the number of the entire rule or text lines of the document page and the complexity is O(N). The final stage of the algorithm for recognising and reproducing the tables only depends on the pixels of a few tabular lines. This table portion is comparatively smaller in size than the text portion of a document page and can be negligible. Therefore, the algorithm completely depends on the number of pixels and traverse in the vertical direction of the document page. The computational steps are still determined by the number of lines on a document page. Hence, from the overall analysis of the complete algorithm, it behaves as linear, and the time complexity of the complete algorithm is O(N). The algorithm works on a single document page at a time, and the size of a document image may very much depend on the document.

CONCLUSION

In this paper, we have proposed and evaluated an extended new algorithm for locating tables from scanned documents. From the experiments, the system has shown comparatively much higher detection ability than the existing algorithm addressed in Jahan and Ragel (2014). We have determined an automated local threshold for line height and word space, which can be determined from a particular document and completely depend on it. Although the intermediate processes such as i) layout analysis using local threshold; ii) eliminating bullets and numbering lines and iii) eliminating header and footer section from the text content results in more improvement within the extending algorithm, the process of threshold computation for word space and line height can be enhanced in the future to achieve robustness of the system. The motivation towards the extraction of tabular features can be used to reproduce the detected tables with their real formatting features for future editing and updating purposes. The algorithm has a linear complexity of O(N), which will be advantageous while processing a large number of scanned documents during the rapid conversion of the traditional library system into the digital system. Since the scope of this research is to reproduce printed documents in their real formatting, existing OCR techniques can deal with the text related issues, and our attempt can completely support table related issues to achieve the goal. We believe that our work can tremendously help and support to reproduce old and rare printed books or any other printed documents to preserve their format and republish them with updates.

REFERENCES

1. Ajward S., Jayasundara N., Madushika S. & Ragel R. (2010). Converting printed Sinhala documents to formatted editable text. *Proceedings of the 5th International Conference on Information and Automation for Sustainability (ICIAFs)*, volume 5, Colombo, Sri Lanka, 17 – 19 December, pp. 138 – 143.
DOI: http://dx.doi.org/10.1109/iciafs.2010.5715649
2. Chen J. & Lopresti D. (2011). Table detection in noisy off-line handwritten documents. *Proceedings of the 11th International Conference on Document Analysis and Recognition*, Beijing, China, 18 – 21 September, pp. 399 – 403.
3. Cesarini F., Marinai S., Sarti L. & Soda G. (2002). Trainable table location in document images. *Proceedings of the 16th International Conference on Pattern Recognition*, Quebec City, Canada, 11 – 15 August, pp. 236 – 240.
DOI: http://dx.doi.org/10.1109/icpr.2002.1047838
4. Gatos B., Danatsas D., Pratikakis I. & Perantonis S.J. (2005). Automatic table detection in document images. *Proceedings of the 3rd International Conference on Advances in Pattern Recognition,* Bath, UK, 22 – 25 August, pp. 609 – 618.
DOI: http://dx.doi.org/10.1007/11551188_67
5. Gatos B., Pratikakis I. & Perantonis S.J. (2004). An adaptive binarization technique for low-quality historical documents. *Proceedings of the 6th International Workshop*

on Document Analysis Systems, Florence, Italy, 8 – 10 September, pp. 102 – 113.
DOI: http://dx.doi.org/10.1007/978-3-540-28640-0_10

6. Harit G. & Bansal A. (2012). Table detection in document images using header and trailer patterns. *8th Indian Conference on Computer Vision, Graphics and Image Processing*, Mumbai, India, 16 – 19 December, pp.1 – 8.
7. Hu J., Kashi R., Lopresti D. & Wilfong G. (2000). Medium-independent table detection. *SPIE Document Recognition and Retrieval* **VII**: 291 – 302.
8. Jahan M.A.C.A. & Ragel R.G. (2014). Locating tables in scanned documents for reconstructing and republishing. *Proceedings of 7th International Conference on Information and Automation for Sustainability,* Colombo, Sri Lanka, 20 – 22 December, pp.1 – 6.
9. Fang J., Gao L., Bai K., Qiu R., Tao X. & Tang Z. (2011). A table detection method for multipage PDF documents via visual separators and tabular structures. *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR'11),* Beijing, China, 18 – 21 September, pp. 779 – 783.
10. Kasar T., Barlas P., Adam S., Chetalain C. & Pacquet T. (2013). Learning to detect tables in scanned document images using line information. *Proceedings of the 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, 25 – 28 August, pp.1185 – 1189.
11. Kieninger T. & Dengel A. (2001). Applying the T-RECS table recognition system to the business letter domain. *Proceedings of the 11th International Conference on Document Analysis and Recognition,* Beijing, China, 18 – 21 September, pp. 518 – 522.
DOI: http://dx.doi.org/10.1109/ICDAR.2001.953843
12. Laurentini T. & Vaida P. (1992). Identifying and understanding tabular material in compound documents. *Proceedings of the 11th International Conference on Pattern Recognition*, The Hague, The Netherlands, 30 August - 03 September, pp. 405 – 409.
DOI: http://dx.doi.org/10.1109/icpr.1992.201803
13. Mandal S., Chowdhury S.P., Das A.K. & Chanda B. (2006). A simple and effective table detection system from document images. *International Journal on Document Analysis and Recognition* **8**(2): 172 – 182.
14. Namboodiri A. (2004). On-line handwritten document understanding. *PhD thesis,* Michigan State University, USA.
15. Oro E. & Ruffolo M. (2009). PDF-TREX: an approach for recognizing and extracting tables from PDF documents. *Proceedings of the 10th International Conference on Document Analysis and Recognition,* Barcelona, Spain, 26 – 29 July, pp. 906 – 910.
16. Shafait F. & Smith R. (2010). Table detection in heterogeneous documents. *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, Boston, MA, USA , 9 – 11 June, pp. 65 – 72.
DOI: http://dx.doi.org/10.1145/1815330.1815339
17. Smith R. (2009). Hybrid page layout analysis *via* tab-stop detection. *Proceedings of the 10th International Conference on Document Analysis and Recognition,* Barcelona, Spain, 26 – 29 July, pp. 241 – 245.
DOI: http://dx.doi.org/10.1109/icdar.2009.257
18. Wang Y., Phillips I.T. & Haralick R. (2001). Automatic table ground truth generation and a background-analysis-based table structure extraction method. *Proceedings of the 6th International Conference on Document Analysis and Recognition,* Washington, USA, 10 – 13 September, pp. 528 – 532.
19. Watanabe T., Naruse H., Luo Q. & Sugie S. (1991). Structure analysis of table-form documents on the basis of the recognition of the vertical and horizontal line segments. *Proceedings of the 1st International Conference on Document Analysis and Recognition*, Saint – Malo, France, 30 September – 02 October, pp. 638 – 646.
20. Wonkyo S., Hyung I.K. & Nam I.C. (2015). Junction-based Table Detection in camera captured document images. *International Journal of Document Analysis and Recognition* **18**(1): 47 – 57.
21. Yildiz B., Kaiser K. & Miksch S. (2005). Pdf2table: a method to extract table information from pdf files. *Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI'05)*, Pune, India, 20 – 22 December, pp. 1773 – 1775.
22. Zanibbi R., Blostein D. & Cordy J.R. (2003). A survey of table recognition: models, observations, transformations, and inferences. *International Journal of Document Analysis and Recognition* **7**: 1 – 16.