# Detection of Tables in Scanned Document Images

T Kasar, S Adams and T Paquet
Laboratoire LITIS - EA 4108
Universite de Rouen, FRANCE 76800
Email:thotreingam.kasar@inv.univ-rouen.fr
{Sebastian.Adams,Thierry.Paquet}@litislab.eu

P Barlas and C Chatelain
INSA Rouen
FRANCE 76800
Email: {philippine.barlas, clement.chatelain}@insa-rouen.fr

***Abstract*—The paper presents a method to detect table regions in document images by identifying the column and row line-separators and their properties. In order to enhance dark thin line-like structures, the input image is first smoothed with a Gaussian filter followed by gray-scale morphological black-hat operation. This pre-processed image is adaptively-thresholded and a run-length approach is employed to extract horizontally and vertically-aligned lines. From each group of intersecting horizontal and vertical lines, we extract a set of 26 low-level features and use an SVM classifier to test if it belongs to a table region or not. The performance of the method is evaluated on a dataset containing various types of table structures and compared with that of Tesseract OCR system.**

## I. Introduction

Tables are compact and efficient for summarizing relational information present in diverse document classes such as newspaper articles, scientific papers, forms, invoice, product descriptions or financial statements. While it is relatively easy for humans to spot a table, a precise definition of a table still remains elusive. Due to the inumerable possible types of table layouts, it is difficult to model a table and automatic understanding of tables from generic documents still reamins a daunting task.

The process of automatic understanding of tables involves the following two modules i) table region detector that identifies regions in the document that correspond to tables and ii) table structure recognizer that extracts relational information from the identified table region to derive the logical structure of the table; direct OCR will simply fail since the fields of a table are inter-related and individually carry a little sense. The focus of our work in this paper is on the problem of table detection. Just like the need for preprocessing steps like skew correction or text-graphics separation in any optical character recognition (OCR) system, localizing table regions is also an indispensable step in table processing systems. Robust segmentation of table regions is essential to ensure a more reliable table structure recognition.

## II. Review of related work

In this section, we give a brief review of prior research on the problem of table detection in scanned images that contain at least some ruling lines. One of the earliest works on identifying tabular regions in document images is the method proposed by Watanabe et al. [1]. The method identifies individual item blocks enclosed by vertical and horizontal line segments on the basis of the interpretation of detected lines. Firstly, line segments are detected and the corner points are subsequently determined. The connective relationships among the extracted corner points and hence the individual item blocks are interpreted using global and local tree structures.

Laurentini and Viada [2] proposed a method to detect tables where text and lines are horizontal or vertical. Text regions are identified using a bottom-up approach and the detected characters are grouped into words and subsequently phrases. Based on threshold on the horizontal and vertical run length, lines are obtained. The arrangement of these detected lines is compared with that of the text blocks in the same area. Further, using the horizontal and vertical projection profiles, the algorithm attempts to add missing horizonal and vertical lines in order to fully understand the table structure.

Green and Krishnamoorthy [3] proposed a model-based top-down approach for table analysis by a hierarchical characterization of the physical cells. Horizontal lines, vertical lines, horizontal space and vertical space are used as features to extract the table region. Elementary cell characterization is performed to label individual cells in such a way that the cells belonging to an underlying nesting or overlapping of logical units can be properly extracted. These raw labels are matched to a table model such that the relational information in the table can be extracted.

Cesarini et al. [4] present a system for locating table regions by detecting parallel lines. They use a recursive analysis of the modified X-Y tree of a page to identify regions surrounded by horizontal (vertical) lines. The search is refined by looking for further parallel lines that can be found in deeper levels of the tree. The hypothesis that a region corresponds to a table is verified by the presence of vertical (horizontal) lines or spaces in the regions included between the two parallel lines. After the complete tree analysis, sub-tables belonging to one table are merged while tables smaller than a given threshold were discarded. The method requires that at least two parallel lines are present.

Gatos et al. [5] detect horizontal and vertical rulings and progressively identify all possible types of line intersection. Table reconstruction is then achieved by drawing the corresponding horizontal and vertical lines that connect all line intersection pairs.

In [6], the authors use the layout analysis module of Tesseract OCR to detect the position of tab-stops. These candidates are grouped into vertical lines to find tab-stop positions that are vertically aligned. Pairs of connected tab lines are adjusted such that they end at the same y-coordinate. Table regions are determined based on analysis of the column layout of the page and the column partitions. The method however require the presence of large text regions (paragraphs) so that the column layouts can be reliably estimated.

Methods such as the ones proposed in [7], [8] do not rely on the presence of lines but use only text information. In [7], tables are assumed to have distinct columns so that the gaps between the fields are substantially larger than inter-word gaps in normal text lines. As the authors pointed out, the method works only for Manhattan layout and may fail for complex documents. All lines are removed as a pre-processing step. This can result in inaccurate detections for partially-filled tables. The detected table boundaries even for correct detections can still have large discrepencies when compared with the ground-truth. Moreover, it is difficult to interpret individual data of a table meaningfully, even if the characters are recognized correctly. Line delimiters, if present, can be utilized for better localization accuracy and to simplify the process of retrieving the spatial and geometrical relationships amongst different blocks of a table.

In our approach, we seek to identify horizontal and vertical lines present in the image and learn a classifer to detect tables based on the properties of the detected lines. The method can detect complex table structures, insensitive to the layout or the number of columns in the page.

## III. System description

### A. Extraction of horizontal and vertical lines

We employ a run-length approach to extract lines along the horizontal and vertical directions. In this work, it is assumed that the table is printed on a white background. So, in order to enhance dark and thin line-like structures, the input image $I$ is first smoothed with a Gaussian filter and the grayscale morphological black-hat operation is then performed.

$$I_\sigma = I * G_\sigma \tag{1}$$

$$I_p = I_\sigma - (I_\sigma \bullet S_N) \tag{2}$$

Where $\sigma$ represents the variance of the Gaussian filter, $S_N$ is a square structuring element of size $N$, $*$ and $\bullet$ denote the 2-D convolution and the morphological closing operation respectively. The Gaussian smoothing operation helps to maintain the continuity of narrow gaps between line segments whereas the effect of black-hat operation is to highlight 'small' dark structures while suppressing 'wide' dark structures at the same time. The variance $\sigma$ of the Gaussian function controls the amount of smoothing. The size $N$ of the structuring element decides the maximum width of the line that can be detected by the system and is empirically set to 7 in this work.

The pre-processed image $I_p$ is then thresholded adaptively with $k_1$ times its maximum intensity as the threshold value. Since our method relies only on the line information and not on the textual information, $k_1$ is fixed to a low value of 0.05 so that even 'weak' lines show up after thresholding. The resulting binary image $I_{bw}$ is subjected to a run-length count along the rows and columns to obtain horizontal and vertical lines. If the count of 'ON' pixels in a particular direction starting at a pixel location exceeds a threshold value $l$, the segment is accepted as a line. All pixels with run-lengths less than the specified threshold value are ignored. The threshold $l$ decides the shortest line that can be detected by the system and is adaptively set to 1/15 times the width of the input image.

It may be mentioned here that the document is assumed to have no skew, and hence a skew detection and correction step may be invoked, if necessary. Combining the output of the line segments obtained from the two directions, we get a composite image $I_L$ that contains all the detected horizontal and vertical lines.

### B. Extraction of table features

We perform a connected component (CC) analysis on the line image $I_L$ and proceed to validate those CCs that are comprised of at least 3 intersecting horizontal (or vertical) lines as tables or not. Let $H_i, i = 1, 2, n$ and $V_j, j = 1, 2, m$ denote the set of horizontal and vertical lines that constitute the CC respectively. The indices $i$ of the horizontal lines are sorted from top-to-bottom order while the indices $j$ sorted from left-to-right order. Let $L_H^i, H_{start}^i$ and $H_{end}^i$ represent the length, starting and ending positions of the line $H_i$ respectively. The inter-line spacing bewteen $H_{i+1}$ and $H_i$ is given by the L$_\infty$-norm between the starting (or ending) points of the two lines; $H_{spacing}^{i,i+1} = |H_{start}^{i+1} - H_{start}^i|_\infty$. Similarly, $L_V^j, V_{start}^j, V_{end}^j$ and $V_{spacing}^{j,j+1}$ denote the same for the vertical line counterparts.

Since we search for tables that are enclosed by lines, the following features are computed from each CC (group of intersecting horizontal and vertical lines):

$$f_1 = \frac{L_H^1}{Max(\{L_H\})} \tag{3}$$

$$f_2 = \frac{L_H^n}{Max(\{L_H\})} \tag{4}$$

$$f_3 = \frac{|H_{start}^n - H_{start}^1|_\infty}{Max(\{L_V\})} \tag{5}$$

$$f_4 = \frac{L_V^1}{Max(\{L_V\})} \tag{6}$$

$$f_5 = \frac{L_V^m}{Max(\{L_V\})} \tag{7}$$

$$f_6 = \frac{|V_{start}^m - V_{start}^1|_\infty}{Max(\{L_H\})} \tag{8}$$

The lines that constitute a table normally have some degree of regularity in their arrangement, lengths and spacings between adjacent ones. Tables also occupies a significant portion of the image area in general. These characteristics are captured by the following features:

$$f_7 = \frac{Median(\{L_H\})}{Max(\{L_H\})} \tag{9}$$

$$f_8 = \frac{Std(\{L_H\})}{Mean(\{L_H\})} \tag{10}$$

$$f_9 = \frac{Std(\{H_{spacing}\})}{Mean(\{H_{spacing}\})} \tag{11}$$

$$f_{10} = \frac{Std(\{H_{start}\})}{Mean(\{H_{start}\})} \tag{12}$$

$$f_{11} = \frac{Std(\{H_{end}\})}{Mean(\{H_{end}\})} \tag{13}$$

$$f_{12} = \frac{Median(\{L_V\})}{Max(\{L_V\})} \tag{14}$$

$$f_{13} = \frac{Std(\{L_V\})}{Mean(\{L_V\})} \tag{15}$$

$$f_{14} = \frac{Std(\{V_{spacing}\})}{Mean(\{V_{spacing}\})} \tag{16}$$

$$f_{15} = \frac{Std(\{V_{start}\})}{Mean(\{V_{start}\})} \tag{17}$$

$$f_{16} = \frac{Std(\{V_{end}\})}{Mean(\{V_{end}\})} \tag{18}$$

$$f_{17} = \frac{Height(CC)}{Height(InputImage)} \tag{19}$$

$$f_{18} = \frac{Width(CC)}{Width(InputImage)} \tag{20}$$

In addition, since the constituent lines of a table intersect each other, a CC belonging to a table will exhibit a high negative value of Euler number due to a large number of line-intersecting points. Tables are normally enclosed by lines on both horizontal and vertical sides. Unlike tables, graphic objects and line-drawings tend to have 'open' lines that do not intersect other lines at one of its end. Due to the presence of 'open' lines, such non-table objects tend to have a much higher number of convex deficiency regions. These characteristics are captured by the following features:

$$f_{19} = \#OpenHor.Lines \tag{21}$$

$$f_{20} = \#OpenVert.Lines \tag{22}$$

$$f_{21} = EulerNumber(CC) \tag{23}$$

$$f_{22} = \#IntersectionPoints \tag{24}$$

$$f_{23} = \frac{\#(CCBordersPixels == ON)}{Perimeter(BoundingBox(CC))} \tag{25}$$

$$f_{24} = \#ConvexDeficiencyRegions \tag{26}$$

$$f_{25} = \frac{Area(ConvexDeficiencyRegions)}{Area(Imfill(CC, holes))} \tag{27}$$

$$f_{26} = \frac{Perimeter(ConvexHull(CC))}{Perimeter(CC)} \tag{28}$$

### C. Table classification with an SVM classifier

We use the LibSVM toolbox [9] to implement an SVM classifier using the above features. The classifier is trained using an RBF kernel on a training data that comprises 339 tables and 345 nontable examples. The optimal parameters of the SVM classifier are obtained using a grid-search with 5-fold validation process.

For a given test image, lines are detected as described in Section III-A and each group of intersecting horizontal and vertical lines are validated using the trained classifier. Isolated lines and CCs that contain less than 3 intersecting horizontal or vertical lines are not considered for table classification.

## IV. Experimental results

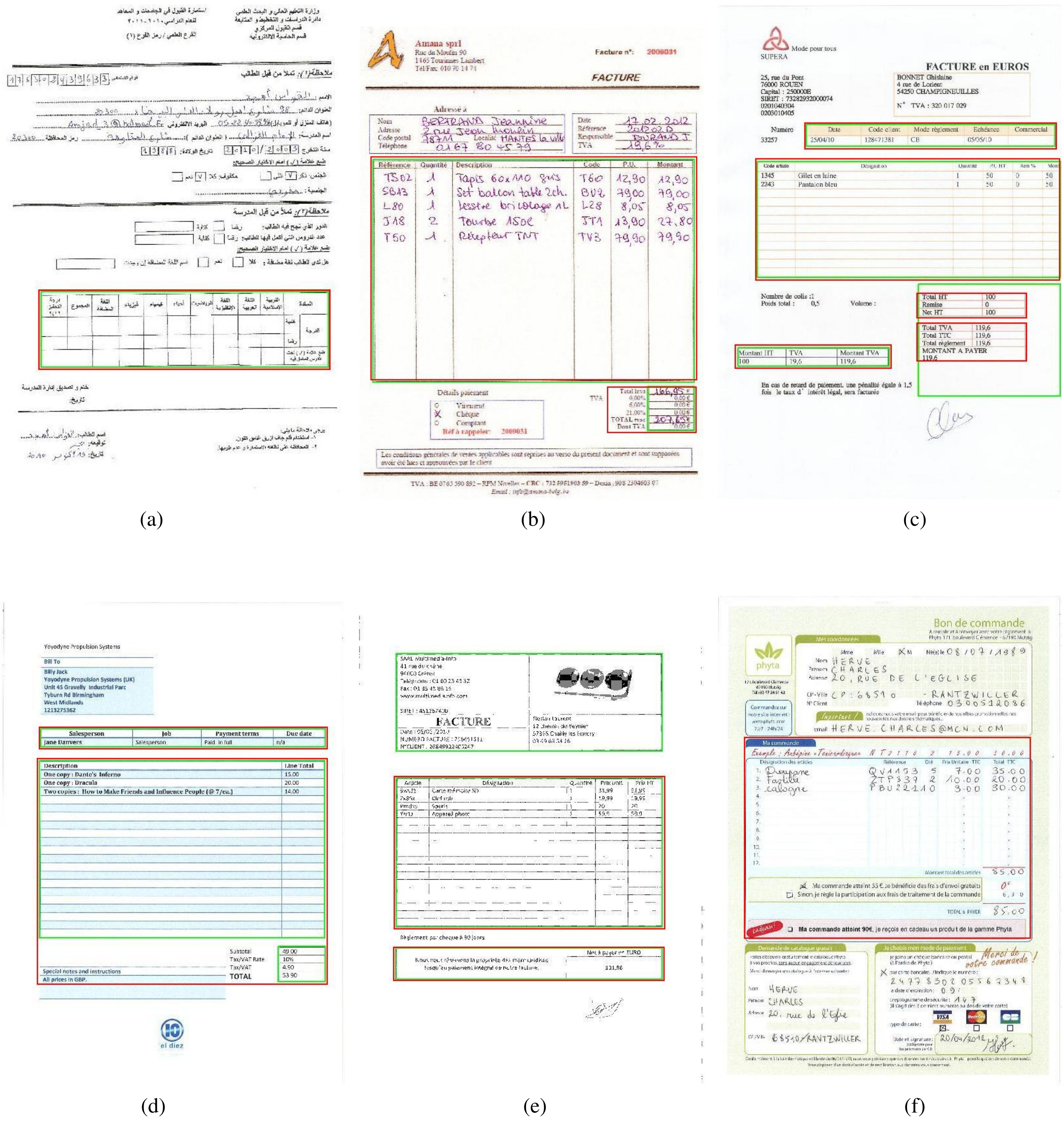
The method has been tested on 190 scanned documents and fax images that contain various types of table structures. Our dataset contains textual information in 3 differnet scripts namely, French, Arabic and English. In addition, these documents contain both printed and handwritten text. The image resolution varies from 100 to 300 dpi.

There are a total of 294 tables in all that have been manually tagged. The ground-truth is represented by rectangles that circumscribe the table. To evaluate the performance of our method, we use the evaluation metrics that are employed in [6]. Our method yields 273 detections, out of which 235 are Correct; 10 Partial; 5 Under; 2 Over; 40 missed and 21 False detections. We also test the performance of the table detector of Tesseract OCR system on the same data, which yields 154 detections and the comparative results are shown in Table I. It is observed that the tesseract table detector rely on the text information, which results in large localization errors; about 45% detections are only partially correct. Also, a large number of tables (approx. 53%) are missed by Tesseract since it is difficult to localize the type of tables considered in our experiment using the text information alone. On the other hand, our method yields accurate table detections with significantly fewer instances of missed detections. An example instance of each type of detections are shown in Figure 1.

TABLE I
Comparative results of the proposed method and Tesseract table detector.

| Evaluation metric | Tesseract | Proposed method |
|---|---|---|
| Correct | 17 | 235 |
| Partial | 69 | 10 |
| Under | 17 | 5 |
| Over | 20 | 2 |
| Missed | 157 | 40 |
| False | 31 | 21 |

The performance of our method is also evaluated based on the area precision and area recall measures. The area precision is the ratio of the area of the detected table regions that actually belong to table regions in the ground-truth image. The area recall measure calculate the percentage of the ground-truth table regions that are detected as tables by the algorithm. The corresponding values of precision and area recall for the Tesseract system are also given in Table II.


(a)

(b)

(c)

(d)

(e)

(f)

Fig. 1. Example instances of (a) Correct (b) Partial (c) Under (d) Over (e) False and (f) Missed detections. The red boxes represent the ground truth while the table regions detected by our method are represented by green rectangles.

TABLE II
PRECISION AND RECALL MEASURES OF THE PROPOSED METHOD AND TESSERACT.

| Evaluation metric | Tesseract | Proposed method |
|---|---|---|
| Area Precision | 46.4% | 83.1% |
| Area Recall | 30.3% | 83.2% |

## V. DISCUSSION AND FURURE WORK

We have presented a new method for table detection in scanned document images using the properties of line separators. Unlike other image-based techniques, our method relies only on the presence of lines and do not require any text analysis and hence do not require specialized binarization process. All the lines can obtained reliably thanks to the action of morphological black-hat operation. Since we employ run-length to obtain the lines, the method is also not affected by characters that may touch the lines. Identification of the row and column line separators during the table detection stage implicitly gives the information about each cell of the table which is useful in subsequent table processing steps to understanding its logical structure.

While the method performs well for tables completely enclosed by lines, there are, in practice, other types of table layouts that may contain only parallel lines that separate the table header or with no lines at all. Our future work is to extend the method to handle tables without border lines too using additional cues from the image.

## REFERENCES

[1] T. Watanabe, H. Naruse, Q. Luo and N. Sugie, “Structure analysis of table-form documents on the basis of the recognition of vertical and horizontal line segments”, Proc. Intl. Conf. Document Analysis and Recognition, pp. 638-646, 1991.

[2] A. Laurentini and P. Viada, “Identifying and understanding tabular material in compound documents”, 2, Proc. Intl. Conf. Pattern Recognition, 1992.

[3] E. A. Green and M. S. Krishnamoorthy,“ Model-Based Analysis of Printed Tables”, Proc. Intl. Conf. Document Analysis and Recognition, pp. 214-217, 1995.

[4] S. Cesarini, S. Marinai, L. Sardi and G. Sorda, “Trainable table location in document images”, Proc. Intl. Conf. Pattern Recognition, pp. 236-240, 2002.

[5] B. Gatos, D. Danatsas, I. Pratikakis and S. J. Perantonis,“Automatic table detection in document images”, Proc.Intl. Conf. Advances in Pattern Recognition, pp. 609-618, 2005.

[6] F. Shafait and R. Smith, “Table Detection in Heterogeneous Documents”, Proc. Intl. Workshop. Document Analysis and Systems, pp. 65-72, 2010.

[7] S. Mandal, S. P. Chowdhury, A. K. Das, B. Chanda, “A simple and effective table detection system from document images”, Intl. Jl. Document Analysis and Recognition, 8(2), pp. 172-182, 2006.

[8] T. G. Kieninger, “Table structure recognition based on robust block segmentation”, Proc. Document Recognition V, SPIE, 3305, pp. 22-32, 1998.

[9] C. C. Chang and C. Lin, “LIBSVM: a library for support vector machines”, http://www.csie.ntu.edu.tw/cjlin/libsvm.