A Very Efficient Table Detection System from Document Images

S. Mandal, S. P. Chowdhury, A. K. Das
CST Department
Bengal Engineering College (DU)
Sibpur, Howrah
{sekhar,shyama,amit}@cs.becs.ac.in

Bhabatosh Chanda
ECS Unit
Indian Statistical Unit
Calcutta 700 035, India
chanda@isical.ac.in

Abstract

The requirement of detection and identification of tables from document images is crucial to any document image analysis and digital library system. Here in this paper we report a very simple but extremely powerful approach to detect any table in any form that may be present in a document page. The algorithm rely on the observation that the tables has distinct columns whose physical implication is in the presence of substantially larger gaps between the fields than the gaps between the words in text lines. This deceptively simple observation has led to the design of a simple but powerful table detection system with a low computation cost and achieving an efficiency close to 100%. Moreover, mathematical foundation of the approach is also established including formation of a regular expression for ease of implementation.

1. Introduction

Millions of paper documents are being produced everyday adding an ever ending wealth of information to the human society. Practical use of these documents demand indexing, viewing, printing and extracting the intended portions in a fast and flexible way through electronic media. With the maturity of the document image analysis such systems are coming in the market. These include digital document libraries, vectorization of engineering drawings and form processing systems [14, 11, 12, 17, 1] to name a few.

Here in this paper we present a fully automated technique for detection and segmentation of tables from the document images.

Common task for a typical document image analysis (DIA)system starts with skew correction and identification of the constituent parts of the document image to text, graphics, half-tones etc. The graphics portion may be vectorised and text portion may be put to OCR. However any table that may be present in the document needs to be identified and requires special treatment because the fields are inter-related and individually carry a little sense. It may be noted that the table detection/segmentation step may be followed by table recognition step where the goal is to find out the logical or layout structure of the table. For such recognition problems it is usually assumed that the tables are already segmented out from the document or the whole document is a table [9]. In this paper our goal is limited to segmenting out tables of any kind from the scanned document for subsequent processing in a simple and efficient manner.

2. Past Work

Table detection and segmentation is done by many researchers [3, 21, 6, 20]. Watanabe et al. [21] have proposed a tree representation to capture the structures of various kinds of tables. Table structure detection is also reported in [10, 3]. Zuyev [22] described a table grid and defined the compound cell and simple cell of a table based on table grid. Node property matrix is used by Tanaka [18] in processing of irregular rule lines and generation of HTML files. Unknown table structure analysis is proposed by Belaid [2]. Tersteegen et al. proposed a system for extraction of tabular structure with the help of predefined reference table [19] and Tsuruoka [20] proposed a segmentation method for complex tables including rule lines and omitted rule lines. In [6] a technique is described to separate out tables and headings present in document images. Ramel et al. [16] used a flexible representation scheme based on clear distinction between the physical table and its logical structure. Detection and extraction of tables is done by the analysis of graphics line present in the table in the context of the representation scheme. For tables without the rule lines a multilevel analysis of the of the layout of text component is carried out to capture the regularities of the text present in a typical table.

Only a few approaches for table detection are based on the textual contents of the document; we describe two rep-

resentative [13, 9] from them. In [13] individual words are clustered and a block segmentation graph is constructed based on the overlaps of the individual items (words) in consecutive lines. Note that in tables the overlaps are limited to individual columns and the block segmentation graph will be distinctly different from a block of normal text. The authors claim that this approach works fine for ASCII file and may be extended for a scanned document. However the algorithm contains too many heuristics and no guideline is given regarding the choice of the parameters used for segmentation. In [9] a structured approach based on dynamic programming is taken to find out which input line(s) can be taken as a part of a table. This is done by computing some characteristics like *score*, *merit* and *line correlations* to ascertain the gain (or loss) if the candidate line is taken (or rejected) as a part of the table. This approach has a strong theoretical foundation but the couple of empirical constants used in the characteristic measures need to be fixed without *a priory* knowledge. As a result the detection rate is limited to 81% for the scanned image and only 83% for ASCII text. This reflects the difficulty in mapping the theoretical proposition to a practical implementation.

3. Proposed Work

The objective of the present work is to find out any table that is present in a scanned document using simple checks on the structural properties of the document thereby avoiding costly solutions.

This work is the continuation of our earlier work on segmentation where the document image containing text, graphics, half-tones are segmented. The work starts with the gray scale image of the page. Half-tones are removed first [5]. The image is then binarised [15] and skew corrected [7]. Further processing is done column wise; so multicolumn document is stripped into separate columns. Graphics are extracted next leaving text zones only [4] which is used as the input for table detection in the present work.

3.1. Observation

To formulate the rules for detection of the tables from document images we have scanned 52 pages containing different types of tables. The observations are listed below.

- Tables may be bounded by boxes. Rows and columns may have horizontal and vertical rule line.
- Tables without any box and rule lines are also common.
- The gap between the fields (columns) is significantly larger than the normal word gap in a text line. This feature is applicable to all tables and distinguishes table rows from normal text lines.

3.2. Steps for table detection

The table detection algorithm depends primarily on **A)** formation of word clusters in text lines and **B)** finding out the set of consecutive candidate text lines which would form a table. As a preprocessing step component labelling is done first to find out the median of the height and width of the components. Using these medians all vertical and horizontal rule lines are eliminated. Unlike some works on table detection which rely on the presence of the rule lines our approach is to remove them to get unified tabular forms and to extend the gap between the fields; a characteristics which is primarily exploited in our work.

A. Formation of Text line clusters

This is done by coalescing the words in a line. Normally a text line would be converted to a single rectangular block while a row of a table would consists of multiple smaller blocks. Such a word coalescing depends on the accuracy in finding the normal word gap and finding out consecutive connected component in a single text line.

We next mathematically formulate the cluster formation. Consider a binary image $I_{M\times N}$, which consists of connected components $C_k(k = 1, 2, \ldots, L)$, as defined in standard text [8] with their usual meanings. Let $L(C_k)$, $R(C_k)$, $T(C_k)$ and $B(C_k)$ be the 4 extreme points of the k_{th} connected component in 4 directions (i.e.; left, right, top and bottom) respectively.

Suppose function F guarantees that the two connected components are in the same text line. Then function F may be represented as

$$F(C_a, C_b) = \begin{cases} 1 & \text{if} \quad (T(C_a) \leq B(C_b) \\ & \qquad AND \\ & \qquad B(C_a) \geq T(C_b)) \\ 0 & \text{otherwise} \end{cases}$$

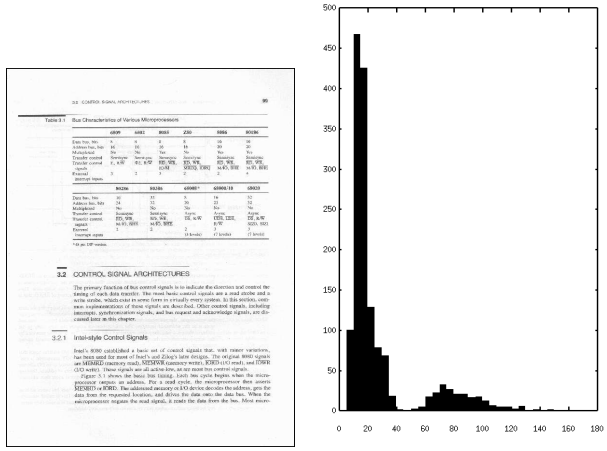
Note that for any three connected components C_a, C_b and C_c; if $F(C_a, C_b) = 1$ and $F(C_b, C_c) = 1$ then $F(C_a, C_c) = 1$; i.e., transitive property holds on relation F for connected components.

Cluster formation requires information on inter-word gap. This may be obtained from the histogram H of distance D between two consecutive connected components C_a and C_b. The distance function D is defined for computing the horizontal distance between any two consecutive connected components, as below

$$D(C_a, C_b) = L(C_b) - R(C_a)$$

where $b = \min_x\{L(C_x) - R(C_a)\}$ such that $(F(C_a, C_x) = 1$ AND $L(C_x) > R(C_a))$. The histogram H registers the intermediate character gap of two consecutive characters. It may be noted that if there is only one font in the document then we will get two distinct humps in H; first one for character gap in a word and the second one for the word gap. An

example page and corresponding histogram is shown in figure 1(a) and (b). If there is more than one font we may find



(a) (b)

Figure 1. Example of a page and histogram; (a) document page; (b) histogram of the preprocessed page with a horizontal scaling factor of 5.

few other humps however first hump will be most prominent followed by the second hump. Our intention is to find out the word gap in the normal text in a document page so that we could combine the consecutive words into a single cluster. For doing so we have taken the upper boundary (υ) of the second hump. Morphological closing operation with a structuring element of area $(\upsilon \times 1)$ will form the clusters denoted as V_w (where $w = 1, 2, \ldots, Q$). The cluster formation will be dictated by the following two conditions:

1. If there are two connected components C_m and C_n $(1 \leq m, n \leq L)$ having the relations
   - $F(C_m, C_n) = 1$
   - $D(C_m, C_n) \leq \upsilon$

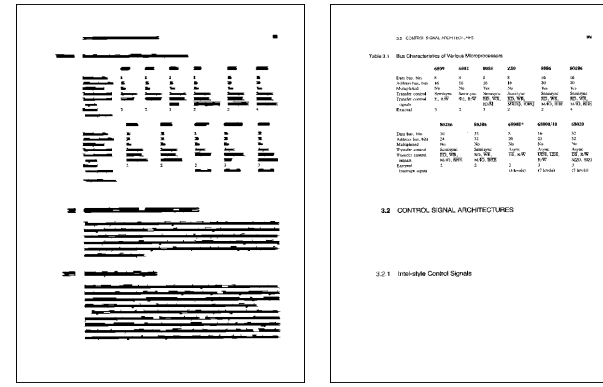
   then C_m and C_n should belong to the same cluster.
2. $V_a \cap V_b = \emptyset \;\; \forall\, a, b \mid (1 \leq a, b \leq Q \;\; AND \;\; a \neq b)$

A typical cluster formation using the computed structuring element from the histogram is shown in fig. 2(a).

B. Selection of candidate text line for table(s)

Clusters are formed by coalescing the connected components so it has all the physical properties of a connected component. Thus we can directly apply the previously defined five functions $L,\ R,\ T,\ B,$ and F on these clusters. Let there are total T number of distinct text lines which are represented as TEL_a where $a = 1, 2, \ldots, T$.

Each text line is nothing but a set of clusters such that a cluster should not be shared by more than one text lines. Two clusters will be in two different text lines if they have



(a) (b)

Figure 2. Example of cluster formation; (a) Cluster formed from image shown in fig 1(a); (b) candidate lines.

a positive intermediate vertical gap and two cluster will be in same text line if they overlapped in horizontal projection. Number index of the text lines are assigned in raster scan order which implies that for two text lines TEL_a and TEL_b we will reach TEL_a first for all $(a < b)$, such that $(1 \leq a, b \leq T)$.

Candidate text line selection is done primarily by taking all lines that has more than one cluster (see fig. 2(b)). It may be noted that all words in a text line are coalesced to a single cluster whereas we get multiple clusters for table rows. Mathematically

$$CND(TEL_a) = \begin{cases} 1 & \text{if} \quad (CNT(TEL_a) > 1 \mid (1 \leq a \leq T) \\ 0 & \text{otherwise} \end{cases}$$

Where CNT counts the number of clusters in a line.

It may be noted that the primary selection is not enough to select all potential candidate lines; we may miss some of the rows of the tables. Those lines should be included for better performance and to prevent splitting errors. Imposing candidature to some of the lines which have not been considered in primary selection is the next task. Imposing candidature to non selected text lines is based mathematically on the following: A function namely VD, calculates the vertical distance between two consecutive text lines, is defined by

$$VD(a) \;=\; min(T(V_i) \;-\; B(V_j))$$

for all $V_i \;\in\; TEL_{a+1}$ AND for all $V_j \;\in\; TEL_a$ such that $(1 \leq i, j \leq Q)$ AND $(1 \leq a < T)$.

MWG computes the median of the intermediate candidate line gaps and is given by

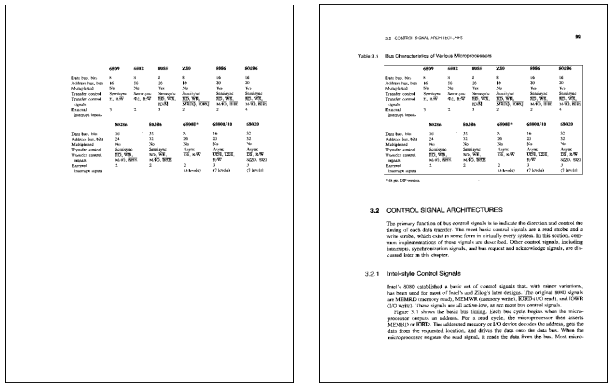
$$MWG \;=\; median(VD(a))$$

for all a $CND(TEL_a) = 1$ AND $CND(TEL_{a+1}) = 1$ such that $(1 \leq a < T)$

Based on the function VD and the measure MWG we impose the candidature on some text lines whose candidature is currently false if the following conditions are satisfied.

1. $CND(TEL_{a-1}) = 1$
2. $VD(a-1) \leq (\eta \times MWG)$
3. $CND(TEL_{a+1}) = 1$
4. $VD(a) \leq (\eta \times MWG)$
   where $(1 < a < T)$ and η is a constant [1].

This gives candidature to those lines which has initially failed to fulfill the characteristics of a candidate text line but its neighbour lines are candidate text lines and the intermediate distance with the neighbours are less than the tolerance value ($\eta \times MWG$). Now, a table in the document page is the set of more than one consecutive candidate text lines which have intermediate line gap less than ($\eta \times MWG$). Heading lines (from tables and others) which have been primarily selected as candidate lines are eliminated as isolated lines because no table has a single row. An example of the extraction of table from the candidate lines is given in figure 3. Table detection from the candidates text lines can also be



(a) (b)

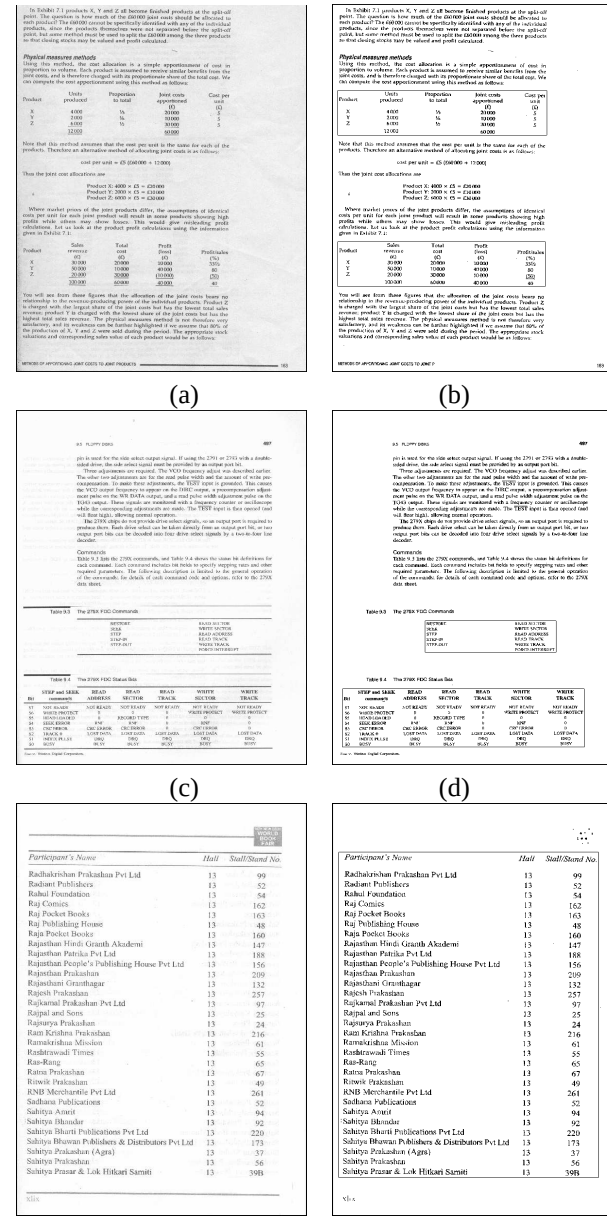
Figure 3. Example of table extraction from candidate lines; (a) extracted lines for fig 1(a); (b) Table within the preprocessed image.

modeled by a regular expression as elaborated below.
Let a initial candidate text lines be represented as C, non candidate text line be represented by N, and intermediate gap between two consecutive text lines be represented by T. If the gap is less than or equals to ($\eta \times MWG$) then the string represented by the regular expression CT((CT) $\cup$ (NTCT))$^+$ will correspond to a table in the page.

[1] set to 1.5 in our experiment

4. Experimental results

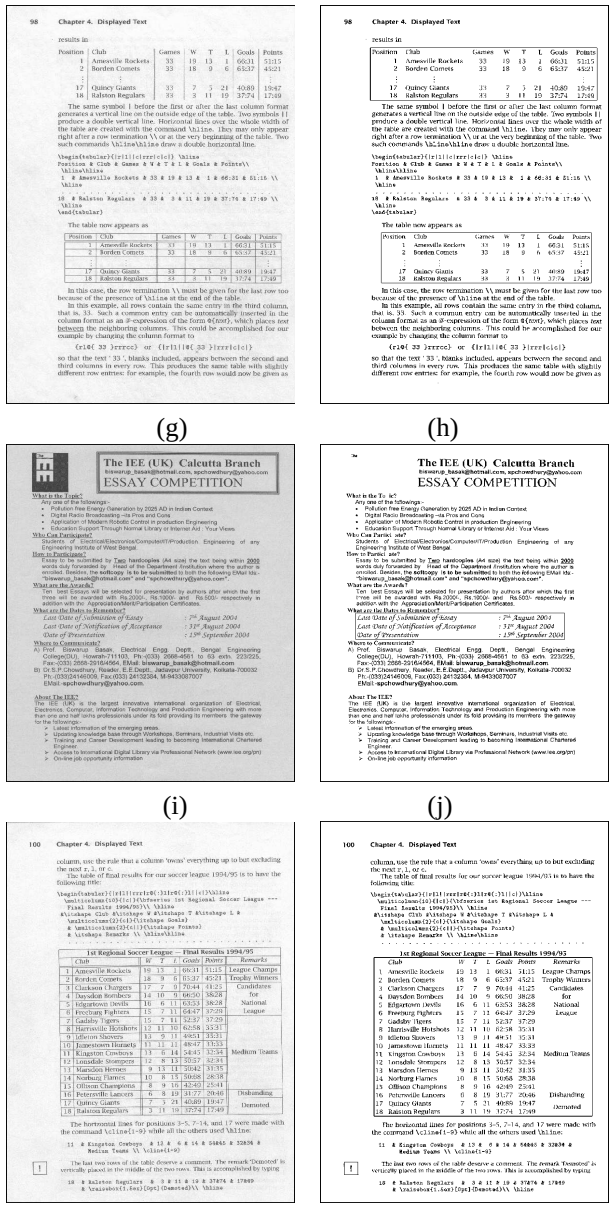
The experiments are done on the dataset using document pages from University of Washington's document image database (UW1 & UW2) and our own collection. About



(a) (b) (c) (d) (e) (f)

Fig. 4 – continued to next page

300 document pages are tested of which ≈48% pages contain table(s). All the programs are written in C and the tests are carried out in a COMPAQ DS 20E server running digital UNIX. Table 1 and table 2 shows the performance figures while fig. 4 shows a few results with typical tables. The average time for detection and identification of the table(s)



(g) (h)

(i) (j)

(k) (l)

(m) (n)

Figure 4. Figures (continued from previous page) in the left are the original page image and in the right are segmented portions within a box on preprocessed image.

for a page is about 1 second including the preprocessing operations. The results we have got are highly encouraging considering the simplicity and low computation cost of our approach. As stated earlier our approach does not rely on rule lines and performance is equally well for both types of the tables; in fact we are removing the rule lines as a preprocessing step.

Table 1. Overall table detection performance.

	Page with table		Page without table
	Table row	Text line	Text line
Detected as table row	97.21%	0.00%	0.09%
Detected as text line	2.79%	100.00%	99.91%

Table 2. Errors in table detection.

	Page with table	Page without table
Input page	143	157
Table present	218	0
Detected table	226	4
Insertion error	0.00%	0.09%
Merging error	0.00%	N.A.
Deletion error	2.79%	N.A.
Splitting error	3.67%	N.A.

The strength of the approach can also be verified from the results given in fig. 4. We see that the algorithm successfully singled out different types of tables from the document pages. There may be multiple tables in a page where text lines are present between the tables (see fig. 4(a) & (b)) and there may not be any text lines between two successive tables (see fig. 4(c) & (d). Full page table is also segmented as shown in fig. 4(e) & (f). In fig. 4(g) & (h) two tables are segmented from a page properly rejecting a near tabular structure in between the detected tables. Next in fig. 4(i) & (j) a hidden table detection from a page containing a notice is shown. In fig. 4(k) & (l) it is notable that the table is properly detected though there were a couple of rows with blank fields; a job which is difficult for any table detection algorithm. Finally in the last example (fig. 4(m) & (n)) we see that top and bottom row lines with a lot of blank fields have not been included showing deletion error in extreme cases.

Regarding other characteristics of our algorithm it may be noted that the chance of insertion error is practically nil implying no or extremely low merging error as reflected in the performance figures. Though the performance of the al-

gorithm is very good but it has its blemishes too. It may not be able to separate two tables appearing side by side; though this is rare but in that case merging would be 100% for those tables. Multiline headings may also cause problem as these have bigger fonts with more gaps than the normal one. However as a discriminating criterion vertical projection profile may help as multiline headings will not have their word gaps in column-like fashion. Another source of concern is the tabular display math-zones like matrices and determinants; they are likely to be identified as tables.

5. Conclusion

We conclude this paper by reiterating the plus point of our approach. First and foremost it is fully automatic with only one threshold (η). It can extract tables and table like structures like lists. Performance is quite satisfactory and does not affected by the change of font and page style as the clustering is done on parameters computed for each page. Moreover it is based on very simple observation leading to a low cost high performance implementation. Finally a mathematical treatment is presented and also a regular expression is proposed using which the implementation can be done quickly through compiler tools. We hope that in future we would able to improve our algorithm so as to avoid merging error for multiple tables appearing side by side and increase our dataset for a higher level of confidence on the performance measures.

References

[1] H. S. Baird. Digital libraries and document image analysis. In *Proc. 7th Internatiolal Conference on Document Image Anlysis; Vol I*, pages 2–14, Los Alamitos, California, 2003. IEEE Computer Society.

[2] Y. Belaid, J. L. Panchevre, and A. Belaid. Form analysisi by neural classifi cation of cells. In *Proc. of 3rd IAPR WOrkshop on Document Analysis Systems (DAS'98)*, pages 69–78, Nagano, Japan, 1998.

[3] S. Chandran, S. Balasubramanian, T. Gandhi, A. Prasad, R. Kasturi, and A. Chhabra. Structure recognition and information extraction from tabular documents, 7. *IJIST*, (4):289–303, 1996.

[4] A. K. Das. *Document Image Segmentation: A morphological approach*. PhD thesis, Bengal Engineering College (Deemed University), Sibpur, India, 1998.

[5] A. K. Das and B. Chanda. Text segmentation from document images: A morphological approach. *Journal of Institute of Engineers (I), 77, NOvember*, pages 50–56, 1996.

[6] A. K. Das and B. Chanda. Detection of tables and headings from document image: A morphological approach. In *International Conf. on Computational linguistics, Speech and Document Processing (ICCLSDP'98); Feb. 18–20,Calcutta, India*, pages A57–A64, 1998.

[7] A. K. Das and B. Chanda. A fast algorithm for skew detection of document images using morphology. *Intl. J. of Document Analysis and Recognition, 4*, pages 109–114, 2001.

[8] R. C. Gonzalez and R. Wood. *Digital Image Processing*. Addision-Wesley, Reading, Mass., 1992.

[9] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *SPIE Document Recognition and Retrieval VII*, pages 1–12, 2000.

[10] K. Itonori. Table Structure Recognition based on Textblock arrangement and Ruled Line Position. In *ICDAR93*, pages 765–768, 1993.

[11] S. H. Joseph. Processing of engineering line drawings for automatic input to cad. *Pattern Recognition, Vol. 22*, pages 1–11, 1989.

[12] E. Katsura, A. Takasu, S. Hara, and A. Aizawa. Design considerations for capturing an electronic library. *Informmation Services and Use*, pages 99–112, 1992.

[13] T. G. Kieninger. Table structure recognition based on robust block segmentation. In *Proceedings Document Recognition V, SPIE, vol. 3305*, pages 22–32, San Jose, California, Jan 1998, 1998.

[14] J. Liu and X. Wu. Description and recognition of form and automated form data entry. In *Proc. Third Int. Conf. on Document Analysis and Recognition, ICDAR'95*, pages 579–582, 1995.

[15] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. SMC, 9, No. 1*, pages 62–66, 1979.

[16] J.-Y. Ramel, M. Crucianu, N. Vincent, and C. Faure. Detection, extraction and representation of tables. In *7th International Conference on Document Analysis and Recognition, Vol. I*, pages 374–378, 3-6 August '03, Edinburgh, U. K., 2003.

[17] S. Satoh, A. Takasu, and E. Katsura. An automated generation of electronic library based on document image understanding. In *Proc. ICDAR 1995*, pages 163–166, 1995.

[18] T. Tanaka and S. Tsuruoka. Table form document understanding using node classifi cation method and html document generation. In *Proc. of 3rd IAPR Workshop on Document Analysis Systems (DAS '98)*, pages 157–158, Nagano, Japan, 1998.

[19] W. T. Tersteegen and C. Wenzel. Scantab: Table recognition by reference tables. In *Proc. of Third IAPR workshop on Document Analysis Systems (DAS'98)*, pages 356–365, Nagano, Japan, 1998.

[20] S. Tsuruoka, K. Takao, T. Tanaka, T. Yoshikawa, and T. Shinogi. Region segmentation for table image with unknown complex structure. In *Proc. of ICDAR'01*, pages 709– 713, 2001.

[21] T. Watanabe, Q. L. Luo, and N. Sugie. Layout recognition of multi-kinds of table-form documents, 17. *IEEE transactions on Pattern Analysis and Machine Intelligence*, (4):432–446, 1995.

[22] K. Zuyev. Table image image segmentation. In *Proc. ICDAR97, Ulm, Germany*, pages 705–707, August. 1997.