Identifying and Understanding Tabular Material in Compound Documents

A.Laurentini and P.Viada
Dipartimento di Automatica ed Informatica, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino–Italy

Abstract

Tables are important components of technical documents. This paper addresses the following problems: i) identifying a tabular component in a scanned image of a compound document containing text, drawings, diagrams, etc.; ii) understanding the content of the table in order to convert the table into electronic format. As far as the authors are aware, the problems addressed are new. An algorithm for performing both the above tasks has been studied and implemented. Preliminary experimental results indicate satisfactory performance for many table lay–out styles.

1. Introduction

Large amounts of compound technical and scientific documents are produced today by computerized systems. The handling of documents in electronic format offers many appealing features for editing, storing, transmitting and searching. Often the users of electronic document handling systems are faced with the problem of acquiring older paper documents in electronic format . Of course, it is possible to create the document manually with the new electronic tools, but this is an expensive and tedious task. In addition, the accuracy of the results can be dubious, especially for components like numerical data and mathematical formulae not suited to easy error checks. A complete solution to this problem would be provided by a system capable of automatic acquisition in electronic format of the paper documents. In principle [1–2], such a system should perform the following tasks, starting from the scanned pages of the document:

1) dividing the document into regions containing components such as text, pictures, engineering drawings, mathematical formulae, tables, etc.
2) understanding the content of each component
3) understanding the logical relations between components
4) transforming the document into some electronic standard format.

No system capable of performing all the above unrestricted tasks is available today. Nevertheless, especially in the last few years, much work has been done on the points listed above, and many interesting results have been obtained. A number of researchers have tackled the problem of segmenting a document image, although in many cases the algorithms proposed were able only to distinguish text blocks from other types of component[3–5]. Much work has been done on identifying and understanding text, and many systems able to read text in mixed–mode documents have been implemented[6–8]. Useful results have been obtained in understanding some other components, like engineering drawings[9–11], circuit diagrams[12] and mathematical formulae[13]. Some methods have also been proposed for understanding the logical structure of a document[14–17].

Little attention has been paid till now to certain other components like tables, which can be found in compound documents.

Tables, often with numerical entries, are frequently found in scientific and technical documents. Although tables could be constructed as a collection of separate textual and graphic components, many popular electronic publishing systems have modules capable of dealing with tables as logical and lay–out objects. This provides the user with useful capabilities for editing a table, when inserting or deleting rows and columns for instance. The importance of tables has been recognized by computer manufacturers, and industrial standards for exchanging tabular material between different applications have been defined, e.g. the Digital Table Interchange Format(DTIF). Today, although the ISO standard ODIF (Office Document Interchange Format) [18] for interchanging compound documents in final and processable form does not provide a format for tables, there exists a commitment to enhance ODIF with formats for items such as tables and mathematical formulae.

The purpose of this paper is to present an algorithm for:
–identifying a table in a scanned compound document
–understanding the table, that is identifying the logical table structure in order to be able to convert the table into an electronic format.

2. Logical and Lay–out Table Structure

From a logical point of view, we consider a table as a set of elements T_{ij}, arranged in i rows and j columns. The elements T_{ij} are usually text blocks, but in some cases they also consist of other objects, like drawings, pictures or mathematical formulae. A closer look very often allows us to per-

0-8186-2915-0/92 $3.00 © 1992 IEEE

ceive a table as one relation of a relational data base. Each column contains a set of values of a data–item; each row collects the values of the attributes of an entity which is uniquely identified by the values of one or more attributes, usually in the first columns.

Usually, the first row of the table does not contains values of attributes, but is devoted to specifying the attributes themselves. This can require more than one row, with elements often in a hierarchical arrangement. We will call these rows *header rows*. Tables with one or more header row will be indicated as *horizontal tables*: this is by far the more common case. When the meaning of rows and columns is interchanged, we have the relatively rare case of *header columns* and *vertical tables*. In some particular cases, a table could be considered both horizontal and vertical, like for instance the familiar 7–bit ASCII code table.

When dealing with tables in electronic format, we are usually interested in header rows since , when a table continues across page or column boundaries, these rows are usually repeated at the top of the new page or new column for improving readability. In this first study however we have not attempted to identify header rows, except for the simple cases of tables with a particular perimeter lay–out, like the table in Fig.1(b). In this case the upper leftmost indented edge allows us to identify the first row r_1 as a header row.

The lay–out of the tables produced by modern electronic publishing systems is regular and relatively simple. In addition to various styles available for single (or double) vertical or horizontal ruling lines, it is possible to join together cells with the same content and to highlight a cell with particular background patterns.

On the contrary, in printed documents there is a very large variety of lay–out styles and not uniform lay–out rules across the same table can be found. Many significant lines may be missing; there may be hanging lines, or lines not properly connected. Therefore, particular care has been devoted to producing a table recognition algorithm which is sufficiently robust to be able to deal with a variety of table lay–out styles.

3. Basic Features of the Algorithm

We describe first the general features and limits of the algorithm implemented. The algorithm is restricted to tables containing text blocks. Therefore to identify a table and understand its structure means to locate a rectangular area in the scanned page and divide the text contained into blocks corresponding to the table entries. For each block the algorithm supplies:

–The geometric information necessary to locate the block on the page. This information is used by a conventional OCR program which reads the text blocks successively.

–One or more pairs of row and column indices. The algorithm is able to identify the cells which have been obtained by joining together two or more elementary cells. For instance, the text block shown in Fig. 1(a) is labeled by the algorithm with the four pairs of indices (4,2), (4,3), (5,2), (5,3). A non–elementary cell must be rectangular.

The algorithm detects tables where text and ruling lines are vertical or horizontal. No diagonal tables are allowed. The text blocks can be both horizontal and vertical in the same table.

The ruling lines of the table should not be connected to lines belonging to other components, as page frames. For working out these cases it is possible to interactively indicate the rectangular area containing the table.

The algorithm is able to cope with a large number of cases relative to the ruling lines. They can be single or double, have different widths, be totally or partially missing and have slightly incorrect alignments. The general idea of the algorithm is to find two kind of elements in the same area:

–block of text arranged in regular horizontal and vertical patterns

–horizontal and vertical lines

and to compare their relative positions. The two processes of identifying and understanding a table are mixed together and the algorithm attempts, in various steps, to construct an *ideal* table in accordance with the elements extracted from the scanned image. Ideal tables are tables where no ruling line between table entries is missing, or where in each cell only one element T_{ij} is contained. In an ideal table two or more cells with equal content could have been joined together(Fig.1).

The construction of the ideal table is carried on by attempting: i) to extend certain hanging ruling lines; ii)to add new *virtual* lines, in order to create a rectangular cell for each entry of the table. If this construction is possible, after the execution of all the steps of the algorithm a table is identified in a rectangular region of the page and its structure un-

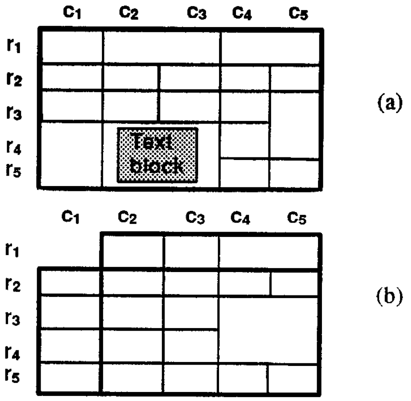


Fig.1. Examples of ideal tables.

derstood. Otherwise, the construction fails at some point and the algorithm stops.

4. The Steps of the Algorithm

A functional block diagram of the algorithm is shown in Fig. 2. For identifying the text, a bottom–up approach has been used, following the general lines of the algorithm presented by Fletcher and Kasturi in [5]. After determining the *four–connected components*, a filtering process is applied to find the rectangular areas containing characters. This process takes into account geometric attributes such as surface and vertical and horizontal dimensions of the smallest enclosing rectangle of each connected component.

For finding horizontal and vertical lines, we first determine the horizontal and vertical *runs*[1], that is sequences containing a sufficient percentage of black pixels. The interval between black pixels of the run must not exceed a certain threshold. This threshold, like others of the algorithm, is adaptive and depends on the average dimension of the characters. As we are interested only in horizontal and vertical lines, only runs with a length exceeding a suitable threshold are selected. This accounts for slightly skew tables.

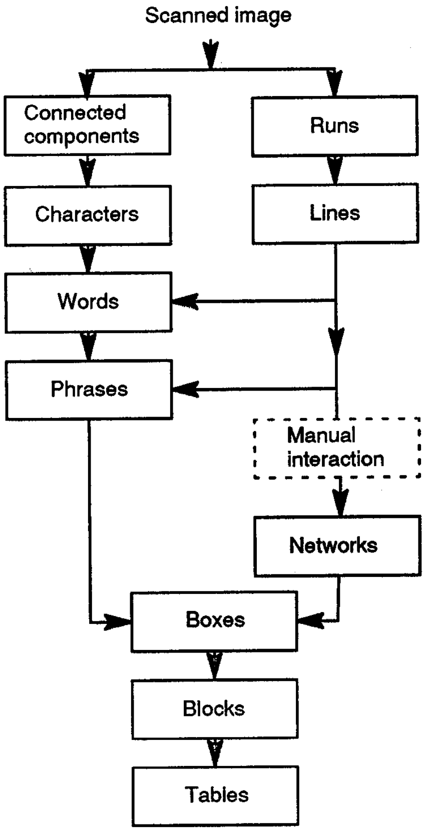


Fig.2–Functional block diagram of the algorithm

In the following step, runs sufficiently near each other are merged together to obtain horizontal and vertical *lines*. The thresholds involved in this process also depends on the character dimension.

Collinear characters satisfying suitable geometric conditions are merged together into *words*, and words are merged into *phrases*. Only the horizontal and vertical merging directions are considered. No merging of elements separated by one of the previously determined vertical or horizontal lines can take place. Some further adjustment of the previously determined connected components is also performed. Errors due to noise, such as the decomposition of a single character into two or more components may be corrected.

The *network* generator identifies a rectangular area containing lines which are candidate ruling lines of a table. At least one horizontal and one vertical ruling line are required. These lines must be connected within a certain threshold. Some adjustment of the hanging lines whose endpoints are near other lines is performed, and some new lines are created. Examples of adjustment and addition of lines are shown in Fig. 3 (a) and (b). Line L in Fig. 3(a) is extended up to L_1, since the gap Δ is less than the character height. Owing to a wider gap, a new line L_3 is added in Fig. 3 (b). The purpose of this step is to complete, as far as possible without considering text elements, the ruling lines of an ideal table. The tests performed in this step on the lines and on their rela-

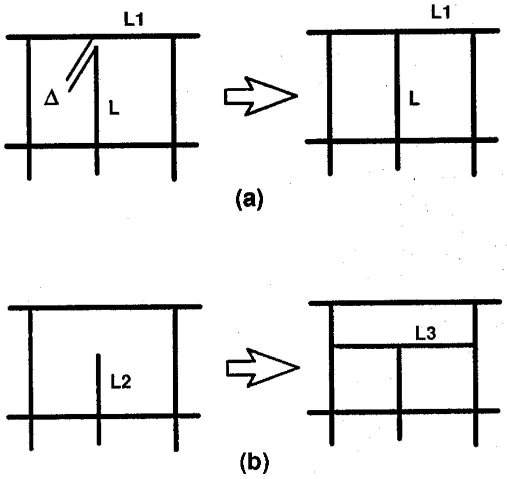


Fig.3–Examples of adjustment (a) or addition (b) of lines.

tive position allow us to discard many non–tabular areas, like those containing line drawings with horizontal and vertical lines. One test consists in comparing the distance between two adjacent horizontal lines with a forbidden distance interval, related to the character height. A very small distance is allowed, since this is the case of a double ruling line.

In the step *boxes*,the arrangement of the lines previously determined is compared with the arrangement of the text blocks identifyed in the same area. Their compatibility is verified . The lay–out of the perimeter of the table and of each cell is checked. At this point, there can be cells containing many text blocks. The arrangement of the text blocks in these areas is checked in the following step *blocks*. This is done by considering the horizontal and vertical projection profiles of the text blocks. If the text blocks are not arranged in a regular pattern,the construction of the table fails. Otherwise, the algorithm attempts to add horizontal and vertical ruling lines for constructing the ideal table. For doing this, the algorithm considers the projection profiles and possible short ruling lines in the profile gaps.

After inserting the missing vertical and horizontal ruling lines, the table structure is fully understood. In the last step, information relative to dimension and position of each cell is prepared for an OCR program. Each cell is flagged with one or more pairs of row and column indices. Finding composed cells is easily done by extending all the ruling lines of the ideal table to the table boundaries.

A sample page containing two tables, text and line drawings scanned at the resolution of 300 d.p.i.'s is shown in Fig.4(a). The algorithm succeeded in identifying the two tables. The corresponding ideal tables are shown in Fig.4(d). In the bottom table a number of lines have been added or adjusted by the algorithm, showing that the table has been correctly understood.

5. Preliminary Experimental Results

The algorithm has been experimentally tested so far in about 20 cases . In all these cases it was able to identify the tables in the compound pages scanned; however in some cases not all the table components where correctly identified.

According to the first experimental results, the algorithm does not appear to be very sensitive either to the resolution of the page scanned, or to the noise. We believe that it will work well also at relatively low resolution, with a substantial reduction of the computational complexity.

We plan to make more extensive tests of the algorithm, and to improve its behavior by relaxing some of restrictions mentioned and adding further capabilities. In particular we plan to address the problems of dealing with tables also containing non–textual elements and identifying header rows. We believe that in a number of cases the identification of header rows does not require a high level semantic approach, but can be performed by examining and comparing simple features of the textual elements contained in each cell, like the number and type of characters.

References

[1] K.Y.Wong. R.G.Casey and F.M.Wahl,"Document analysis systems", *IBM J.Res. Dev*, Vol. 6, pp. 647–656, 1982

[2] S.N.Srihari,"Document Image Understanding", *Proc. IEEE Computer Soc.Fall J. Comp. Conf.*, Dallas, pp.87–96, 1986

[3] T.Taxt, P.J. Flynn and A.K.Jain,"Segmentation of Document Images",*IEEE Trans. PAMI*, Vol. 11, pp.1322–1329, 1989

[4] D. Wang and S.N.Srihari,"Classification of Newspaper Image Blocks Using Texture Analysis", *Comput. Vision,Graphics,Image Processing*, Vol.47, pp.327–352, 1989

[5] L.A.Fletcher and R.Kasturi,"A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images",*IEEE Trans. PAMI*, Vol.10, pp910–918, 1988

[6] D.G.Elliman and I.T.Lancaster,"A Review Of Segmentation and Contextual Analysis Techniques for Text Recognition", *Pattern Recognition*, Vol.23, pp. 337–346, 1990

[7] G. Ciardiello et al,"An Experimental System for Office Document Handling and Text Recognition", *Proc. ICPR'88*,pp. 739–743, 1988

[8] T. Akiyama and N.Hagita,"Automated Entry System for Printed Documents", *Pattern Recognition*, Vol.23, pp. 1141–1154, 1990

[9] R.Kasturi, et al.,"A System for Interpretation of Line Drawings", *IEEE Trans. PAMI*, Vol.12, pp.978–991, 1990

[10] D.Dori,"A Syntactic/Geometric Approach to Recognition of Dimensions in Engineering Machine Drawings", *Comput. Vision,Graphics, Image Processing*, Vol.47, pp. 1–21, 1989

[11]M.Ejiri, et al.,"Automatic Recognition of Engineering Drawings and Maps", in *Image Analysis Applications*, New York: Marcel Dekker, pp. 73–126, 1990

[12]A.Okazaki, et al.,"An Automatic Circuit Diagram Reader with Loop–Structure–Based Symbol Recognition", *IEEE Trans. PAMI*, Vol.10, pp.331–341, 1988

[13] Okamoto and Miyazawa,"An Experimental Implementation of Document Recognition System for Papers Containing Mathematical Expressions", *Proc. SSPR 90* , pp.335–350, June 1990

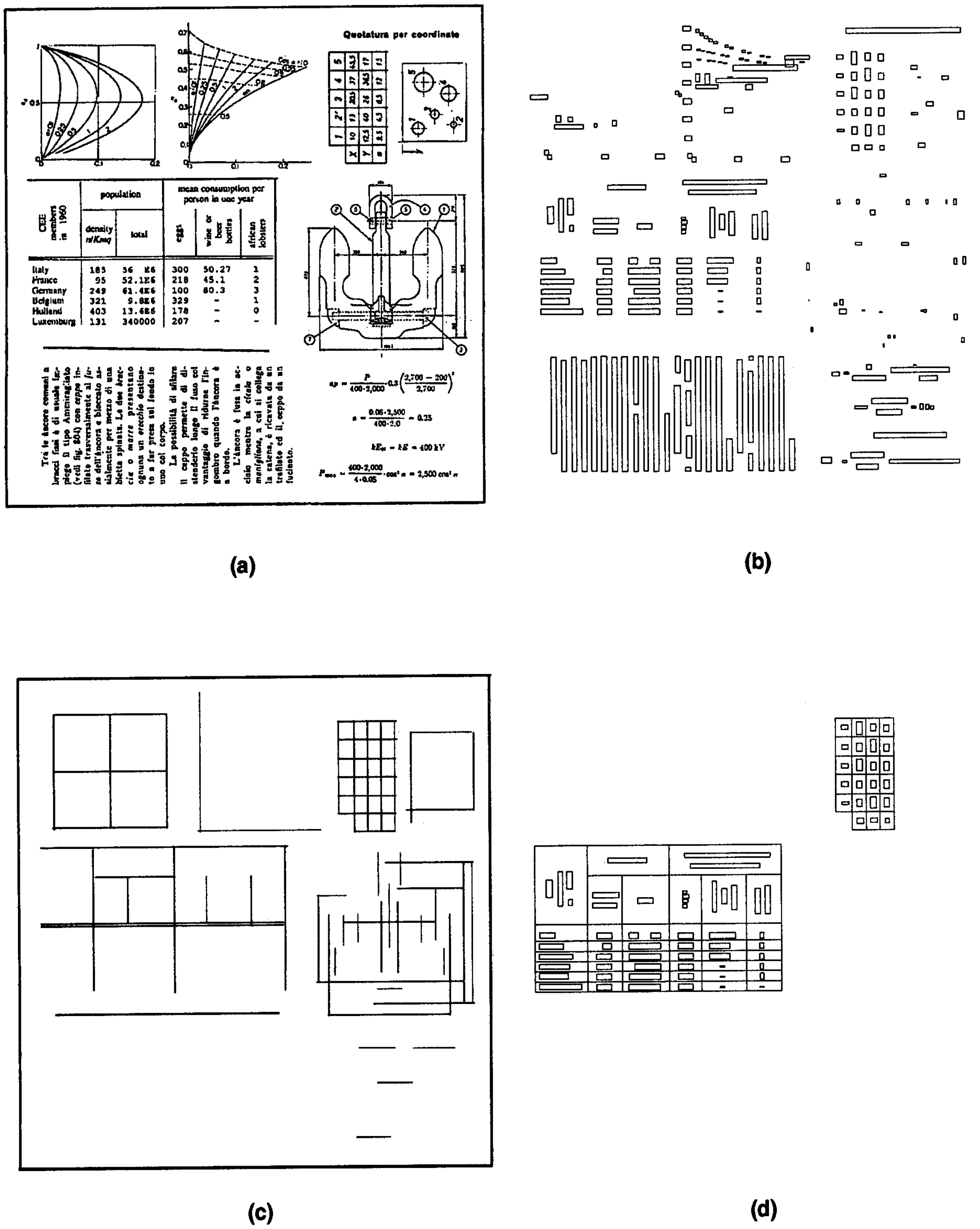
[14] D.Niyogi and S.N.Srihari,"A Rule–Based System for Document Understanding", *Proc. AAAI–86 : Fifth National Artificial Intelligence Conf.*,Philadelphia, 1986

[15] J.Higashino, et al.,"A Knowledge–based Segmentation Method for Document Understanding", *Proc. 8th ICPR*,Paris, pp.745–748, 1986

[16] S. Tsujimoto and H. Asada,"Understanding Multi–articled Documents", *Proc. 10th ICPR*, pp.551–556, 1990

[17] G.Nagy and S. Seth,"Hierarchical Representation of Optically Scanned Documents", *Proc. 7th ICPR*, Montreal, pp347–349,1984

[18] ISO IS 8613, Office Document Architecture(ODA) and Interchange Format, Geneva, 1988



Quotatura per coordinate

	X	Y	ø
1	10	12.5	8.5
2'	13	40	4.3
3	20.5	26	6.3
4	37	36.5	12
5	45.5	17	15

CEE members in 1960	population		mean consumption per person in one year		
	density n/Kmq	total	eggs	wine or beer bottles	african lobsters
Italy	185	56 E6	300	50.27	1
France	95	52.1E6	218	45.1	2
Germany	249	61.4E6	100	80.3	3
Belgium	321	9.8E6	329	-	1
Holland	403	13.6E6	178	-	0
Luxemburg	131	340000	207	-	-

Tra le ancore comuni a bracci fissi è di usuale impiego il tipo Ammiragliato (vedi fig. 804) con ceppo infilato trasversalmente al fuso dell'ancora e bloccato assialmente per mezzo di una biètta spinata. Le due braccia o marre presentano ognuna un orecchio destinato a far presa sul fondo in uno col corpo.

La possibilità di sfilare il ceppo permette di distenderlo lungo il fuso col vantaggio di ridurne l'ingombro quando l'ancora è a bordo.

L'ancora è fusa in acciaio mentre la cicala o maniglione, a cui si collega la catena, è ricavata da un trafilato ed il, ceppo da un fucinato.

$$\alpha_p = \frac{P}{400\cdot 2{,}000}\cdot 0.3\left(\frac{2{,}700-200}{2{,}700}\right)^2$$

$$a = \frac{0.06\cdot 2{,}500}{400\cdot 2.0} = 0.25$$

$$kE_{st} = kE = 400\,\mathrm{kV}$$

$$P_{max} = \frac{400\cdot 2{,}000}{4\cdot 0.05}\cdot\cos^2\alpha = 2{,}500\cos^2\alpha$$

(a)

(b)

(c)

(d)

Fig.4 - A compound page containing two tables (a); the phrases (b) and the lines (c) extracted; the ideal tables constructed by the algoritm (d).