Logical Layout Recovery: approach for graphic-based features

Aysylu Gabdulkhakova, Tamir Hassan, Walter G. Kropatsch
Pattern Recognition and Image Processing Group
Technische Universität Wien
Favoritenstraße 9-11, 1040 Wien, Austria
{aysylu,tam,krw}@prip.tuwien.ac.at

Paper ID 36

Abstract. *In contrast to the existing approaches for document analysis and understanding this paper represents a system that considers a logical role for graphic content in predominantly textual, born digital PDF documents. This work was inspired by the idea of using structural graphic objects in order to clarify the logical layout even of complex mostly graphic documents. Based on visual cognition, geometric features and spatial relations, the proposed statistical method distinguishes illustrative graphic objects from structural graphic objects. We performed evaluation on two document domains - newspapers and technical manuals - and found the results to be reliable. We propose using logical information about the graphic content to be a new step towards domain-independent document understanding systems.*

1. Introduction

A human reader can easily rediscover the logical structure of any document from text properties (typesetting conventions) and layout. In ambiguous cases a human can additionally follow the meaning of the text paragraphs.

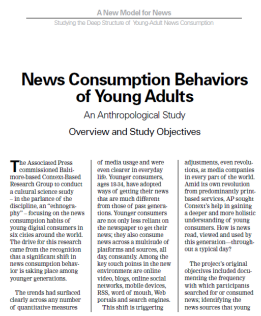
Document analysis and understanding systems presented in literature are focused on textual data in domain-specific documents (books, business letters, scientific papers, technical specifications). They determine a logical structure - heading, paragraphs, reading order - based on individual properties of a given document class. Graphic regions are detected as non-textual and provide no semantic information. The problem stems from a need to create a system capable for a broad class of documents and to reuse or repurpose the document content, represented by graphic objects. In natively digital PDF documents(*PDF Normal* or *Formatted Text and Graphics*) these objects are defined by the set of low-level primitives, such as lines, rectangles, curves and glyphs. Graphic objects either construct an illustrative region (non-structural elements) or distinguish logical blocks from each other (structural elements).

We classify documents into three types according to the layout complexity:

- simple, mostly textual documents: logical layout is obvious for both human and system, e.g. scientific papers(Figure 1);
- predominantly textual documents: logical layout is evident for human, but difficult for system, e.g. newspaper page(Figure 2);
- complex, mostly non-textual documents: sophisticated for human and system, e.g. creative design of magazines(Figure 3).

The proposed object-based approach addresses a problem of analysis and understanding of vector graphic objects that appear in predominantly textual natively digital PDF documents. Our heuristic rule-based method considers logical geometrical properties and mutual arrangement between the graphic and text objects. This approach enables grouping low-level primitives into higher-level logical blocks and finding structural graphic elements.

The remainder of this paper is organized as follows: in Section 2 we provide an overview to state-of-the-art research related to the problem.



A New Model for News

Studying the Deep Structure of Young-Adult News Consumption

News Consumption Behaviors of Young Adults

An Anthropological Study

Overview and Study Objectives

The Associated Press commissioned Bain-more-based Comero-Based Research Group to conduct a cultural science study – in the parlance of the discipline, an "ethnography" – focusing on the news consumption habits of young digital consumers in six cities around the world. The drive for this research came from the recognition that a significant shift in news consumption behavior is taking place among younger generations.

The trends had surfaced clearly across any number of quantitative measures of media usage and were even clearer in everyday life. Younger consumers, ages 18-34, have adopted ways of getting their news that are much different from those of past generations. Younger consumers are not only less reliant on the newspaper to get their news; they also consume news across a multitude of platforms and sources, all day consistently. Among the key touch points in the new environment are online video, blogs, online social networks, mobile devices, RSS, word of mouth, Web portals and search engines. This shift is triggering adjustments, even revolutions, as media companies in every part of the world. Amid its own resolution from predominantly print-based services, AP sought Comen's help in gaining a deeper and more holistic understanding of young consumers. How is news read, viewed and used by this generation – through-out a typical day?

The project's original objectives included documenting the frequency with which participants searched for or consumed news; identifying the news sources that young

Figure 1. Example of a mostly textual document with trivial layout

Bresciaoggi

Dinamica

Marte in campo, il calcio si ferma

Strage senza colpevoli

Alle parti civili restano le spese Veltroni: «Le paghino i partiti»

La verità è il filo sottile che non va spezzato

In auto contro un palo perde la vita a 29 anni

Figure 2. Example of a predominantly textual document with non-trivial layout

Gold MINE

Figure 3. Example of a complex ambiguous document

Section 3 and Section 4 describe two contributions of this paper: proposed method and evaluation tool respectively. Discussion of the obtained results, conclusion and direction for the future work is given in Section 5.

2. Related work

State-of-the-art methods provide various solutions to the problem of logical structure discovery. Most of them take into account image-based features and deal with document images rather than electronic documents. As soon as our system processes natively digital PDF documents in this Section we will focus on approaches that use object-based properties.

Anjewierden [1] created a system, AIDAS, that incrementally builds logical blocks and determines their role using shallow grammars. This approach is limited by the field of technical manuals. Chao and Fan [3] developed a method for information extraction in scientific papers. It uses both object-based and image-based approaches for establishing logical vector graphic entities. Déjean and Meunier [4] present a system that applies XY-cut-based algorithm for dividing the given document into logical blocks. Bloechle et al. [2] present an object-based system, Dolores(**Do**cument **Lo**gical **Res**tructuring), for restructuring textual and graphical content. The idea is based on using artificial neural networks which are trained to recognize the logical layout of the newspapers. Hassan [7] developed a system, PDF Extraction Toolkit[1], that particularly includes an object-based bottom-up method for extracting logical text blocks, vector-graphics objects(lines, curves, rectangles) and bitmaps. The GUI shows the rectangular bounding boxes of the detected component groups, although the primitives themselves may not be rectangular in shape.

The publications mentioned above do not describe the processing of graphical primitives in sufficient detail. However, they do not appear to distinguish between structural and non-structural objects. We believe that structural elements is a powerful feature of the documents, that, as well as human, document understanding systems can use for logical structure recovery.

This idea was roughly implemented by Gao et.al. [6]. Their image-based method aims extracting structural information from PDF book documents. In one of the processing steps they find separation lines, that visually distinguish different parts of the documents. In contrast, our approach is oriented for predominantly textual documents, such as newspapers, that provide rich variety of logical layout and different types of structural elements - lines, rectangles, bitmaps.

We decided to advance the PDF Extraction Toolkit from the point of analysis and understanding of vector graphics. The description of the invented methods can be found in [5], as well as in the next Section.

[1] pdfXtk: http://pdfxtk.sourceforge.net

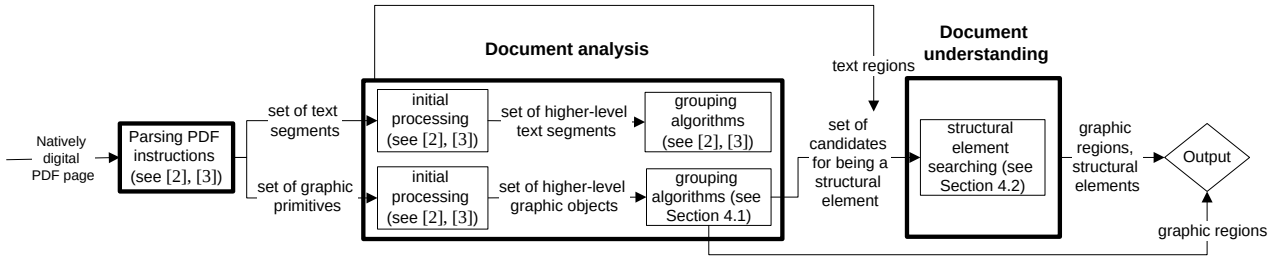


Figure 4. Stages of the whole algorithm for processing natively digital PDF page

3. Methodology

A system performs three tasks in order to represent a natively digital PDF document as a set of text regions, graphic regions and structural elements. First, the page content is extracted from PDF instructions and transformed into Java-object primitives. Page location of these primitives is defined by their bounding box coordinates in 2D Cartesian space. In *pdfXtk* we store the following types of primitives: line segments, rectangles, bitmaps, text segments (2-3 character-long text block).

Next, the grouping rules are applied in order to obtain higher-level graphic and text objects. In the final processing phase, we determine which of the graphical objects represent structural elements, as opposed to graphic regions. The diagram of the whole algorithm can be seen in Figure 4.

The remainder of this section describes our grouping rules and our methods for determining whether a vector object is structural. The task of processing text blocks is not addressed in this paper (see [[8, 7]] for a description).

3.1. Grouping algorithms

The grouping rules that we have devised take into account geometrical properties of the graphic objects as well as their mutual spatial arrangement. We classify these rules into two categories: intersection-based and distance-based. When applied in combination with each other, they enable higher-level objects to be constructed, which usually correspond to distinct logical objects in the document's structure.

3.1.1 Based on intersections

Lines. When the intersection between two lines is established, we group them together. For the next line, we check the intersection with each member of the group. The special case is when two lines construct a solid line, i.e. visually they are perceived as one. In this case we merge these line segments and no group is created.

Rectangles. This method is applicable not only to rectangles, but also to bitmap objects and complex figures (in the latter case, the bounding box is used). It is based on the assumption that two structural rectangle objects on the page are unlikely to intersect. Specifically, when the topmost coordinate of one rectangle is less than the bottommost coordinate of the other or when the leftmost coordinate of one rectangle is greater than the rightmost coordinate of the other.

Often advertisements or other separated content is enclosed in structural rectangles, which are very close to each other or even overlap. Hence, before merging such elements using the above rule, we check whether they enclose other objects. If yes, then the given pair of rectangles is not grouped.

Line and Rectangle. In predominantly textual documents, two types of intersection between line and rectangle objects can occur:

1. structural line intersects the rectangular object;
2. line segment intersects the rectangular object, both are part of a graphic region.

In order to avoid overmerging, three actions are sequentially performed:

a) the width ratio or height ratio, whichever is the larger, is compared to the given threshold;

b) we determine the type of intersection or, more precisely, the mutual arrangement of the intersecting objects (see Figure 5);

c) the ratio between both parts of the line, split at the intersection point, is compared to a given threshold.

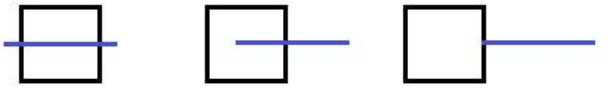


Figure 5. The grouping method for rectangles, based on the sizes and mutual arrangement of the given objects

Line and Text. There are a variety of ways in which line segments and text fragments can intersect each other. In our research we focused on four cases that commonly occur in newspapers:

a) line segment underscores text block;

b) line segment crosses the word;

c) line segments form the axes and text blocks represent labels (as in diagram);

d) text block is surrounded by lines, which distinguish it from the other part of a document.

Cases a) and b) can be distinguished from each other by the distance between their centres projected on the Y-axis: if the line is closer to the centre of the text bounding box than to its border, then case a) applies; otherwise case b). Cases c) and d) can also be distinguished by the distance between their centres, but projected on the X-axis: if the line is touching or intersecting the text bounding box, then case c) applies; otherwise case d).

Rectangle and Text. As in previous paragraph there are several possibilities of intersection between the given objects. More precise:

a) rectangle encloses text fragment;

b) rectangle intersects text fragment;

c) rectangle slightly touches the text fragment.

The last case occurs in tight layouts, where the rectangular bounding boxes of neighboring components often slightly overlap each other.

3.1.2 Based on distance

Lines. Here the distance-based rules consider the possibility of dashed lines. Line segments represent small objects with the distance between them less than or equal to the element size.

Rectangles. Two rectangles are considered as a single object if the distance between their centres is less than a given threshold. This threshold depends on two parameters: a granularity-level coefficient and the widths or heights of the rectangles. It is calculated by multiplying the first parameter with the sum of the second.

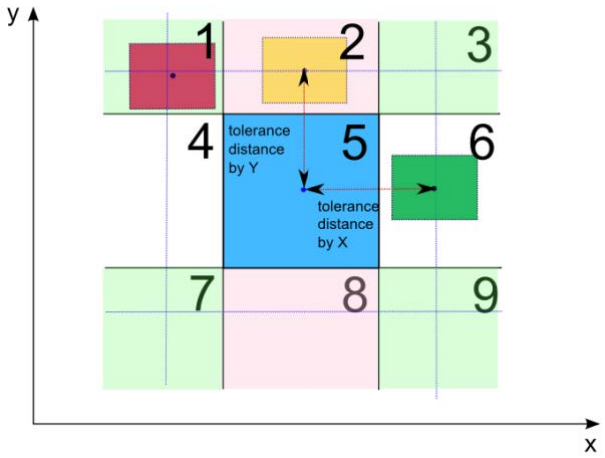


Figure 6. The grouping method for rectangles, based on the sizes and mutual arrangement of the given objects

The granularity-level coefficient depends on a size ratio of the given rectangles and is divided into 3 types: high (0.55), normal (0.6) and low (0.65). These numerical values were obtained experimentally. Next, the algorithm continues by detecting one of nine cases of mutual arrangement between two rectangles (as illustrated in Figure 6). If the current rectangle is located in area 4 or 6 towards the rectangle being considered (green and blue rectangles respectively), the threshold distance uses the sum of both widths. For cases 2 or 8 (yellow and blue rectangles), the threshold distance depends on their heights. For the remaining cases 1, 3, 7, 9 (red and blue rectangles), two threshold distances are counted using the widths and heights. Finally, the threshold distance is compared to the distance between the centres of the given objects projected on the appropriate axis. In cases 1, 3, 7 and 9 the comparison is conducted on both axes with the corresponding thresholds. It is worth noting that our system represents composite objects by their rectangular bounding box. In such a manner, graphic glyphs that form parts of logos, newspaper headings etc. are introduced as rectangular objects. A vivid example of the above glyphs is the heading of newspapers such as The Sydney Morning Herald, International Herald Tribune, etc. (Figure 7). Here, low-level primitives are positioned sequentially in one row/column. In order to detect this case, we refer

Figure 7. Newspaper heading representation in pdfXtk

to the golden ratio font rules [2].

Errors can occur with advertisements that have the same size and are close to each other (Section 3.1.1, Rectangle and Rectangle). These advertisements differ from glyphs as they also contain text. Moreover, the advertising boxes are usually filled with objects as large as at least one third of their size, whereas glyphs have a negligibly small filled area.

3.2. Finding structural elements

Line. Generally a structural line occurs as a horizontal/vertical line or rectangle that looks like a line, which does not intersect other non-structural objects (strokes, lines, rectangles, merged graphic regions, text fragments) and is not enclosed in any graphic region.

In Section 3.1.1, paragraph Line and Text, we considered four cases in which line and text objects can intersect each other. Case a) is a special case, where the line is neither structural nor illustrative, but rather an integral part of the formatting of the text. In case b) the line is likely to form part of an illustrative region. In cases c) and d) there is a pair of identical groups of straight lines with different semantics: in case c) these lines form part of a chart, whereas in case d) the lines are used as a ”barrier” and serve the purpose of separating the text paragraph from the remaining page content. Thereby we conclude that c) is an example of non-structural lines and d) is an example of structural lines. In order to detect the correct variant, the closeness of line segments and text fragments is taken into account.

Rectangle. Generally, a structural rectangle occurs does not intersect other graphic primitives and regions, but can enclose them. The special case is that of ”stand-alone” rectangle objects, which often look like and serve the same logical function as lines. Therefore, in Section 3.1.1, paragraph Rectangle and Text, case a), the rectangle can represent the bounding box of textual content and thus is more likely to be structural than in cases b) and c).

4. Evaluation

For this reason we created an interactive evaluation tool and performed estimation of the obtained results on a bitmap level using similarity measures from [9].

The developed system performs several tasks: ground truth image generation, resultant image generation and comparison of the above images. As an input it takes a binary image of a given PDF document at a fixed resolution 72dpi, which is sufficient for this purpose. Ground truth image generation is obtained by manual marking on a binary image the appropriate logical graphic regions. Resultant image is produced by automatical mapping XML file of the algorithm output to the binary image. In a given two segmentation images - ground truth and resultant - each logical type of a graphic object(structural line, structural rectangle, illustrative region) is represented by a specific color. Correspondence between two images is established via pixel-by-pixel color-value comparison. Interface of the system is shown in Figure 8.

In order to define the measures that determine the nature of overlapping between the regions, we borrow ideas from the approach proposed in [9]:

- *correct detection* regions are mainly overlapping;
- *partial detection* some overlapping detected, but not sufficient as in the first case;
- *over-segmentation* a single object in the ground truth is detected as two separate segments;
- *under-segmentation* - two segments in the ground truth are erroneously merged in the algorithms output;
- *incorrect detection* the types of region in ground truth and result of the algorithm are different (structural rectangle, structural line or graphic region);
- *false positives* the region is marked by the algorithm, but does not occur in the ground truth;
- *missed objects* the region is marked in the ground truth, but has not been detected by the algorithm.

4.1. Experimental results

Graphic-understanding approach was tested on predominantly textual electronic PDF documents of two domains: newspapers and technical manuals. Precisely, we took 100 pages from 10 different European newspapers taken from 15-17 April 2012, namely *Nuovo Quotidiano di Rimini, El Mundo del Siglo XXI, China Daily, Il Tirreno, Die Tageszeitung, Le Monde, L'Eco di Bergamo, Äripäev, International Herald Tribune, Bresciaoggi*; 30 pages from first 8 different technical manuals obtained by using a popular search engine. The results of our evaluation are given in Table 1 and Table 2 correspondingly.

4.2. Discussion

Choosing two particular document classes is caused by the purpose of testing approach on various data. Newspapers provide a rich layout variety and sparse complex vector-graphic objects, such as drawings. Technical manuals, vice versa, are represented by a simple logical layout and mostly include sophisticated figures. The algorithm demonstrates a high performance on both types of predominantly textual, natively digital PDF pages.

Important drawback for the proposed algorithm is that PDF is a descendant of a PostScript page description language. Limited number of rendering instruction causes an unexpected set of underlying operator structures even for a simple page layout. Thus such documents are easily perceived by human, but confuses algorithms that work directly on the operator level.

5. Conclusion and further work

A new approach for analysis and understanding of vector graphic content in natively digital PDF documents is described in this paper. It includes algorithms for grouping graphic primitives into higher-level logical blocks and for understanding their logical role in a document. The efficiency and reliability of the system was tested on newspapers and technical manuals and achieved good results. Choosing these document domains was caused by the need to prove that the provided heuristic rules perform well not only for a specific logical layout pages, but also for the technical figures and schemes.

The goal of this paper is to address a problem of studying the properties of graphical content in documents, as it is a powerful tool for dividing a page into logical blocks. The current implementation is oriented to predominantly textual documents. For the future work we propose to extend our set of heuristic rules by considering bitmap elements or text elements to be structural. This information can be further used to analyse complex mostly graphic documents such as magazines.

Acknowledgements

This work was funded in part by the Austrian Federal Ministry of Transport, Innovation and Technology (Grant No. 829602).

References

[1] A. Anjewierden. Aidas: Incremental logical structure discovery in pdf documents. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 374–378. IEEE, 2001. 2

[2] J. Bloechle, M. Rigamonti, K. Hadjar, D. Lalanne, and R. Ingold. Xcdf: a canonical and structured document format. *Document Analysis Systems VII*, pages 141–152, 2006. 2

[3] H. Chao and J. Fan. Layout and content extraction for pdf documents. In *DAS 2004: Proceedings of the International Workshop on Document Analysis Systems*, pages 213–224, 2004. 2

[4] H. Déjean and J. Meunier. A system for converting pdf documents into structured xml format. *Document Analysis Systems VII*, pages 129–140, 2006. 2

[5] A. Gabdulkhakova and T. Hassan. Document understanding of graphical content in natively digital pdf documents. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 137–140. ACM, 2012. 2

[6] L. Gao, Z. Tang, X. Lin, Y. Liu, R. Qiu, and Y. Wang. Structure extraction from pdf-based book documents. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 11–20. ACM, 2011. 2

[7] T. Hassan. Object-level document analysis of pdf files. In *Proceedings of the 9th ACM symposium on Document engineering*, DocEng '09, pages 47–55, New York, NY, USA, 2009. ACM. 2, 3

[8] T. Hassan. *User-Guided Information Extraction from Print-Oriented Documents*. PhD thesis, Citeseer, 2010. 3

[9] A. Shahab, F. Shafait, T. Kieninger, and A. Dengel. An open approach towards the benchmarking of table structure recognition systems. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 113–120, New York, NY, USA, 2010. ACM. 5

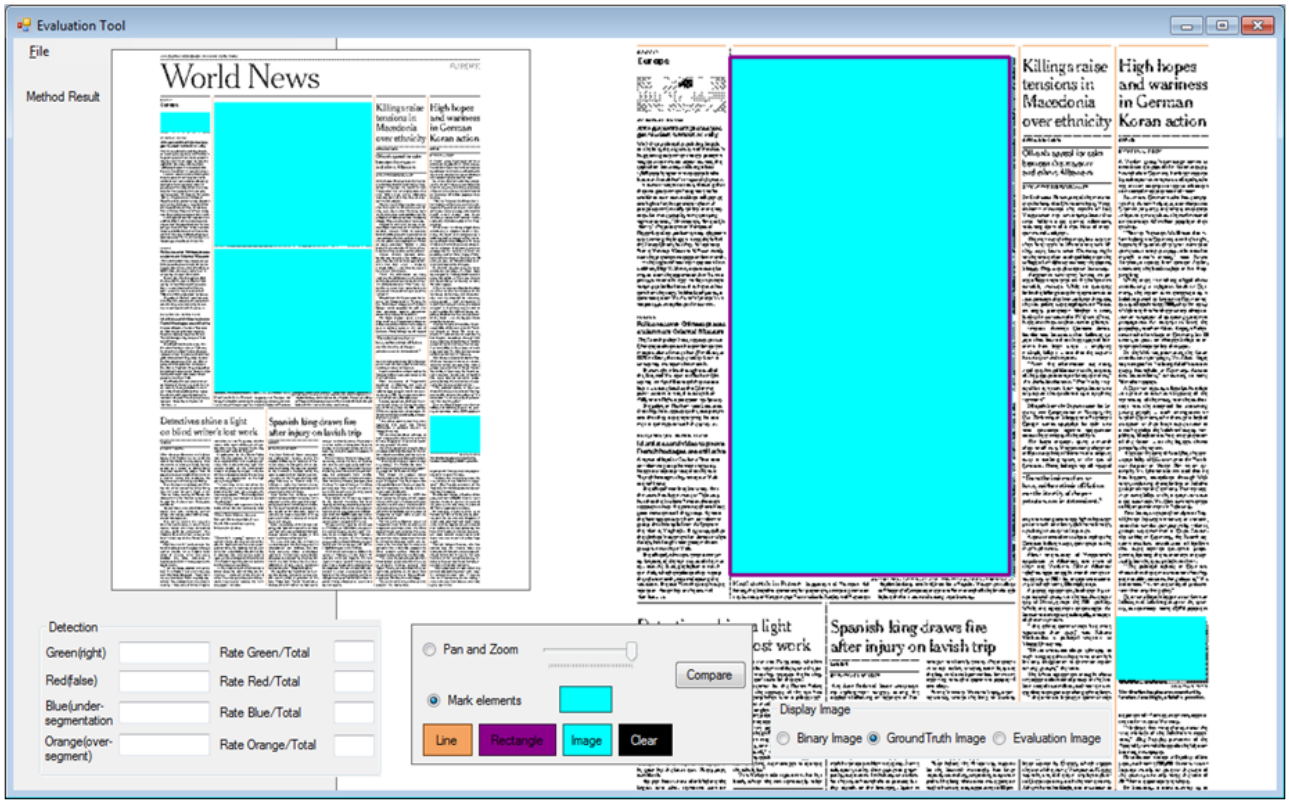


Figure 8. Interface of the evaluation tool

	Total	Retrieved	Correct detection	Incorrect detections	Partial detections	Over segment.	Under segment.	False positives	Missed objects
Structural lines	847	875	710(86%)	93(11%)	0	0	28(3%)	122	30
Structural rectangles	464	377	291(69%)	119(28%)	1(<1%)	11(2%)	4(<1%)	68	42
Graphic regions	364	670	291(82%)	55(15%)	2(<1%)	105(29%)	32(9%)	228	11

Table 1. Evaluation results on newspapers

	Total	Retrieved	Correct detection	Incorrect detections	Partial detections	Over segment.	Under segment.	False positives	Missed objects
Structural lines	92	101	85(92%)	17(18%)	0	0	0	1	0
Structural rectangles	13	21	12(92%)	3(23%)	1(7%)	1(7%)	1(7%)	8	1
Graphic regions	66	80	61(94%)	6(9%)	1(<2%)	13(20%)	3(4%)	1	1

Table 2. Evaluation results on technical manuals