2013 12th International Conference on Document Analysis and Recognition

Table of Contents Recognition and Extraction for Heterogeneous Book Documents

Zhaohui Wu†, Prasenjit Mitra†‡, C. Lee Giles†‡
†Computer Science and Engineering, ‡Information Sciences and Technology
Pennsylvania State University, University Park, PA 16802, USA
zzw109@psu.edu, pmitra@ist.psu.edu, giles@ist.psu.edu

***Abstract*—Existing work on book table of contents (TOC) recognition has been almost all on small size, application-dependent, and domain-specific datasets. However, TOC of books from different domains differ significantly in their visual layout and style, making TOC recognition a challenging problem for a large scale collection of heterogeneous books. We observed that TOCs can be placed into three basic styles, namely "flat", "ordered", and "divided", giving insights into how to achieve effective TOC parsing. As such, we propose a new TOC recognition approach which adaptively decides the most appropriate TOC parsing rules based on the classification of these three TOC styles. Evaluation on large number, over 25,000, of book documents from various domains demonstrates its effectiveness and efficiency.**

I. Introduction

Digitization and OCR technologies are constantly increasing the number of books available online. However, most of them are full text with limited structural information such as pages and paragraphs. More sophisticated book structure, such as chapters, sections, subsections, etc., is often missing, making it difficult for users to efficienty search and navigate inside a digital book. For example, a TOC provides at a glance the entire structure and layout of a book, making its recognition an important feature for book structure extraction and understanding. Due to this, TOC recognition has been extensively studied. However, most methods used ad hoc rules derived from small datasets for specific domains. Whether such rules are effective on large scale heterogeneous book documents is unknown. TOC detection based on predefined connectors such as dot lines [2] will not work on TOCs without these connectors, as shown in Fig. 1a. Indentation and font size based rules [4] will work for TOCs whose entries vary in font size and whose TOC levels are indicated by indentation, such as Fig. 1a and Fig. 1b, but would fail in parsing TOCs that have no indentation and have no font size variety, such as the one shown in Fig. 1c. Clustering based TOC hierarchy extraction methods such as [9], [14] would be ineffective if all TOC entries are in the same visual format (such as the one in Fig. 1c), since they assume that TOC entries at the same level share visual formats while those in different levels do not. Based on these examples, we contend that new methods are needed to address the TOC recognition problem for large scale book documents with various TOC styles. However, the challenge is that it is difficult to find universal rules or templates that govern all possible TOC styles. Though there are many types of TOC, the basic style of a TOC can be detected from its layout, numbering, and other features. For example, it is easy to tell whether a TOC is hierarchical or flat, or with section numbers or without. We observed that

(a) TOC without section numbers and connectors (b) TOC with section numbers and indentation (c) TOC without section numbers and hierarchy

Fig. 1: Examples of TOCs

any TOC should belong to at least one of the three basic styles, i.e., "flat", "ordered", and "divided". In addition, these basic TOC styles can help decide the appropriate TOC parsing rules. If a TOC is flat, i.e, all entries are in the same (or sufficiently similar) visual format, then we can simply parse all the entries one by one; otherwise it will be hierarchical. Then we need to further consider the indentation and font size for parsing. If a TOC is well numbered, i.e., every entry starts with a section number such as "1.2" or "1.2.1", then we can simply parse it based on the ordered numbers. If it is divided, i.e., it can be divided into multiple visual blocks, then we can detect the hierarchy by detecting blocks and sub-blocks. This motivated our proposed approach for TOC recognition, which adaptively chooses the appropriate rules according to the basic TOC style features. We evaluated it on two datasets containing both PDF and OCRed books from various domains. The results demonstrate its effectiveness and efficiency: our method significantly outperforms the baselines in multiple measures in both of the two datasets while keeping its runtime close to that of the baselines and linear in the size of dataset.

II. Related Work

There is a rich literature on TOC recognition, where most work can be summarized into three categories based on the different features they used, including structure, content, and visual features. Structure feature based methods have been extensively studied in early TOC recognition, mainly deriving parsing rules for TOC entries based on observations from specific tasks. For example, CyberMagazine managed academic journals by automatically obtaining bibliographic database schema and hypertext schema from TOC images [1]. Luo et al. proposed detecting TOC pages by finding predefined connectors such as dot lines for Japanese documents [2]. Lin et al. parsed TOC entries based on simple patterns including: Headline, Chapter number + Headline, Headline + Page

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.244

number, and Chapter number + Headline + Page number [3]. Tsuruoka et al. used the indentation and font size to extract structural elements such as chapters and sections in a book. This technique is limited to the TOC of documents with identifiable and consistent font and spacing properties [4]. Bourgeois et al. presented a document understanding system using probabilistic relaxation based on visual features, which are a combination of typography and the spatial relations between blocks, deduced from the physical layout [5]. Belad proposed a method based on part-of-speech tagging which can be applied in TOC recognition for article field identification by reassembling in the same syntagm title or authors, words having similar tags. However, it is language dependent and was only tested for small journal and proceeding datasets [6]. Mandal et al. used page number-related heuristics for TOC detection on document images [7]. Bourgeois et al. extracted the overall hierarchical logical structure using reading order and reference information of books, by combining spatial property and semantic knowledge from the TOC [8].

The above methods, designed for their corresponding tasks, might not apply on general book documents. Recent works focused on exploring content and visual features or properties of TOC to develop more generic recognition methods. A widely used property is that TOC entries have textual similarity or content association with the related sections [9], [10], [11], [15]. The TOC entries will usually appear as chapter or sub-chapter titles in the body text of a book. Text matching or text similarity measuring techniques are then used to detect TOC entires, while they differ in their specific similarity measurements. Visual feature based methods assume document elements belonging to the same component usually share visual formats and use this general property to detect the entries in the same level [9], [14], [16]. Besides, Sarkar and Saund enumerated various graphical and perceptual cues that provide cues to TOC parsing [12]. Dejean and Meunier used a logistic regression algorithm to train a model based on linked pairs of text blocks for each document [13]. This was a general approach to address the TOC recognition problem for a relatively large scale number and various types of documents. However, it requires a model be trained for every document, which might be difficult to scale. Their experimental results showed that supervised learning provides few improvements over the general properties based methods. Most recently, Jayabal et al. studied the challenges in generating bookmarks from TOC entries in PDF e-books [16]. Our method does not reply on a single property but adaptively chooses the most confident TOC parsing rules according to the classification of TOC styles.

III. TOC Recognition

In general, to effectively automate the recognition and extraction of the TOC of documents, three sub-tasks are to be addressed: TOC detection, TOC parsing and TOC linking. TOC detection locates the boundary of the TOC, or detects all the TOC pages within a document, usually based on explicit heuristics. TOC parsing extracts the semantics (section number, title, page number) and the hierarchy of the TOC, after which the TOC is interpreted as a tree where each node represents an entry in the TOC. TOC linking determines the corresponding content in the body text w.r.t each entry in the TOC. The overall process of our TOC recognition procedure is illustrated in Fig. 2.

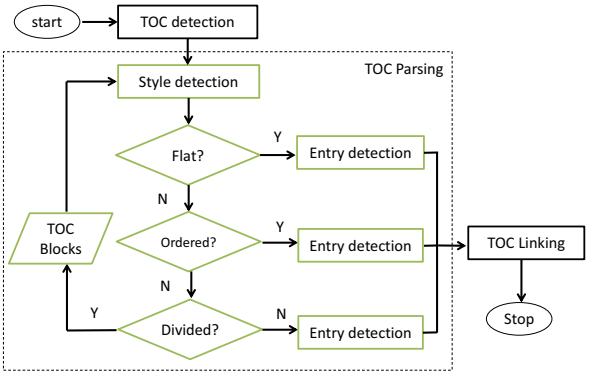

Fig. 2: The flow chart of the overall TOC recognition process

A. TOC detection

TOC detection aims to find the start and the end part of a TOC. For a well-formated book, its TOC usually appears after title page and copyright notices, and in journals, the abstract; and before the list of tables or figures, the foreword, the preface. The start of a TOC will be clearly indicated by "Contents" or "Table of Contents" in a whole line, though there are some rare indicators such as "'Contents at a Glance', "Contents of Volume I", "Check List", "Text Subjects", etc. To detect the start, we simply search for those strong indicators from the beginning of the book to the end of the Kth page, $K = min(20, N/5)$, where N is the total number of pages in a book. Empirically, we found that the TOC appears within the first 20 pages of a document. N/5 is used to reduce the K of a document with a total number of pages ¡ 100. Detecting the end is more difficult than detecting the start. We define the end as the line whose next $m \geq 5$ lines do not exist with any legal page numbers. Legal page numbers are nondecreasing monotonically Arabic numbers. Suppose the previous legal page number is p, the page number of current line q is legal if $p <= q <= p'$, where p' is the page number of the next line. Usually, we take the last word (if it is a number) as the page number. Sometimes, due to OCR or PDF extraction errors, the number is divided into multiple words. For example, "15" is divided into "1" and "5". We thus connect all the continuous numbers at the end of the line. After locating the start line, we go through each line from the start line to the end of the Kth page until finding the end line. m is set to be 10.

B. TOC Parsing

After locating the start and end line of a TOC, we parse it using different rules based on the forementioned styles. The style classification can also be done during the TOC detection when the parser goes through line by line to detect the end. For each line, we calculate and store all the potential features for future use. It detects three properties about the style of a TOC, i.e., whether it is flat, ordered or divided.

A TOC (or a TOC block) is *flat* if all the entries share the visual format, including the start x coordinate, the line height (or average font size), and the line spaces. The start x is the start x coordinate of the leftmost word in a line. The line height

is average height of all words in the line. A line space is the height of the space between two adjacent lines. Flat TOCs usually exist in proceedings, journals, or short books without sub-chapters or sub-sections. The three blocks of TOC in Fig. 1c are all flat. And if the two larger line spaces among the three blocks are equal to the other line spaces, then this TOC will be flat.

A TOC (or a TOC block) is *ordered* if all the entries start with a section number such as "1.2.1", sometimes ahead by "Chapter" or "Section". There are some rare cases that TOC entries start with ordered numbers but have nothing to do with section numbers. For example, they are page numbers or just indicate the order of an entry. To rule out these, we limit the format of section numbers to "a.b.c", "a-b-c-d", without restriction to number of levels.

A TOC (or a TOC block) is *divided* if either one of the three conditions is satisfied: 1) there are centered lines whose start x coordinates are larger than those of other lines; 2) there are lines with larger line height; 3) there are larger line spaces. The centered lines and the lines with the largest height are considered as the first level entries, which usually are titles of Parts, Chapters, or Sections. The largest line space indicates the boundary between two adjacent blocks. A first level block is usually a Part, Chapter, or Section. An typical example of divided TOC is shown in Fig. 3. The first level blocks are labeled by green frames, which are divided based on the largest line spaces. They can further be divided into two sub-blocks based on the largest line height: one sub-block is the Part title entry; and the other containing the left entries, is a flat block.

If a TOC is flat, we then simply parse all the entries one by one. The content between two adjacent appropriate page numbers is considered as the title of an TOC entry. The TOC will have only one level and all entries will be in the same level. Notice that some entries have multiple lines, which needs to be connected together to form the whole title. If a TOC is not flat, then the start x coordinates differ, which means there exists indentation; or the line heights or line spaces vary, which indicates "divided".

If it is ordered, as shown by Fig. 1b, the parsing is then based solely on the ordered section numbers. The hierarchy of the TOC is then determined by the order and level of section numbers. Section "x.y.z" will be a child entry of section "x.y" and a sibling entry of section "x.y.v". Noted that "ordered" is not mutually exclusive to the other styles, i.e., an ordered TOC can be either "flat" or "divided". So we can also put it ahead of the "flat" procedure. Now the TOC is either divided or with indentation.

If it is divided, we first divide it into TOC blocks according to the conditions needed. The key problem is to decide which lines should be the first entry of a block. We use a simple voting strategy on 1) "whether it is centered", 2) "whether it has the largest line height", and 3) "whether the line space before it is the largest". The entries that get at least one vote will be selected as the leading entry of the corresponding block. For all the selected entries, we then proof check based on the following four features: the start x coordinate, the line height, whether section numbers (if there exist) are in the same level, and "whether the last word is a page number" to ensure that they are visually similar (an exceptional entry

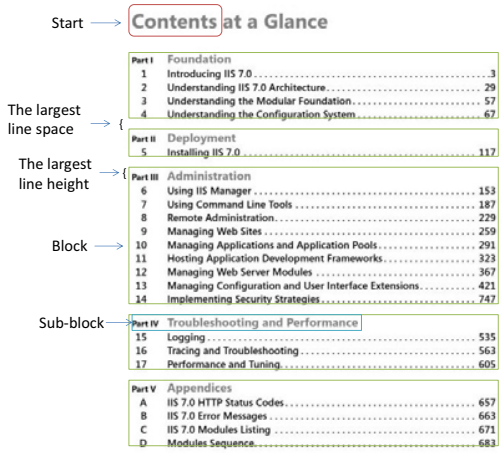

Fig. 3: An example of divided TOC

will be kicked out). For each divided block, we do the same procedure again until all the entries are parsed. Otherwise, the TOC must have indentation and equal line spaces. The TOC level is determined by the indentation. Entries in the same indentation will be in the same level of the TOC hierarchy.

C. TOC Linking

TOC linking is finding the exact page to which a TOC entry refers. If the extracted TOC has a perfect page number sequence, then we only need to find the difference d between a TOC page number and its actual page number in the book. We can sample a few entries to find their exact pages based on title matching. This could save a lot of time if it is a very long book with a long TOC. If all the samples agree on d, we add d to every TOC page number. However, the TOC detection procedure, with emphasis on finding the start and end of TOC, does not guarantee the correctness and completeness of the resulting page number sequence. For example, OCR errors could happen between "1" and "l", or between "3" and "8". The first case will cause the absence of the page number of an entry while the second will result in a wrong number. The TOC detection does not handle those errors, so we need to either correct the irregular page numbers, or find the exact page numbers for those whose lack of page numbers based on title matching. We adopt the same vague title matching technique used in [17].

IV. Evaluation

A. Dataset

The first dataset we evaluated is gathered from Structure Extraction competition at ICDAR [17], [18], which are free books from Internet Archive[1]. By combining all the ground truth set from 2009, 2011 and 2013, we get 1040 unique OCRed books with manually labeled TOC, containing TOC pages, hierarchical entries, and page numbers. Note that the page numbers are the actual page number of the book, which is usually a little bigger than the page numbers appearing in book TOC. Each book has a scanned image PDF and DjVu XML document. The DjVu XML provides coordinates of each

[1]http://archive.org/

word, as well as the line, paragraph, page information. Our experiments are based on the XML documents. All those books are old and out of copy books in art, literature, science, law, history, etc.

The other dataset we use consists of academic PDF book documents collected from CiteSeer[2] repository. CiteSeer crawls homepages of mostly academic authors for freely available academic documents. Most of CiteSeer is scholarly papers. We collected documents satisfying the following two rules: 1) there exists table of contents in its first 20 pages; and 2) the number of its total pages is larger than 200. Using only rule 2), we get 196,425 documents from all the crawled PDFs in the CiteSeer repository. By adding rule 1), the number decreases to 25,680. These documents are then considered as candidate books for our evaluation. By manually check 200 documents randomly sampled from the collection, we found they include Phd thesis (61.5%), technical report (12.5%), conference proceeding (7%), textbook (19%), from computer science, information science, engineering, environmental, biology, etc. The PDF files are extracted using an open source tool pdfbox [3]. This subcollection of CiteSeerX can be made available for others.

B. Measures and Baselines

Since a TOC entry contains three elements: title, page number, and level; to measure the effectiveness of a TOC recognition algorithm, we will have multiple metrics based on each single element or different combinations among them. Vague matching is used for title. Two titles A and B are "matched" if $D = LevenshteinDist(A, B) * 10/Min(Len(A), Len(B))$ is less then 0.2 and if the distance D between their first and last five characters (or less if the string is shorter) is less than 0.6 [17]. If an entry has a matching title linking to the same physical page in the ground truth, we call it is a matching link. If an entry has a matching title at the same TOC level in the ground truth, we call it a matching level. If an entry is both a matching link and a matching level, we call it a complete matching entry. For each of these metrics, we use precision, recall, and F1 value to measure the performance.

Though there has been great deal of work on this topic, none seems to be recognized as state-of-the-art. As such we set up two baselines based on two most commonly used strategies in the literature. The first baseline is based on heuristics using section numbers and indentation [7]. The section number parsing procedure is as the same as that for the "ordered" TOCs while the indentation based rule assumes entries in the same indentation are at the same level of the TOC hierarchy. The second baseline is based on the assumption that TOC entries belonging to the same level share a visual format, as reported in [9], [14], [16]. For each line, we cluster entries based on: the start x coordinate, the line height, whether section numbers (if there exist) are on the same level, and "whether the last word is a page number". We set the maximum number of clusters as 5. We compute the cosine similarity between the current entry and center of each cluster and then add it into a close enough cluster or make it a new cluster. All the entries in the ith generated cluster are considered as the entries in the ith TOC level. In the experimental results, we call the first "Baseline 1", the second "Baseline 2" and ours "Adaptive". To make the comparisons fair, we use the same TOC detection and TOC linking methods proposed in Section II for all the three methods.

[2]http://citeseerx.ist.psu.edu/

[3]http://pdfbox.apache.org/

Metrics	Methods	Precision	Recall	F-measure
Matching titles	Baseline 1	55.3%	51.4%	53.2%
	Baseline 2	69.7%	65.1%	67.3%
	Adaptive	**78.7%**	**76.4%**	**77.5%**
Matching links	Baseline 1	45.3%	42.1%	43.6%
	baseline 2	58.5%	54.7%	56.5%
	Adaptive	**63.4%**	**62.9%**	**63.1%**
Matching levels	Baseline 1	45.8%	48.9%	47.3%
	Baseline 2	56.2%	52.3%	54.2%
	Adaptive	**61.3%**	**58.6%**	**59.9%**
Complete matching entries	Baseline 1	30.4%	25.5%	27.7%
	Baseline 2	35.7%	33.8%	34.7%
	Adaptive	**43.1%**	**40.5%**	**41.8%**

TABLE I: Results on Structure Extraction Dataset

Metrics	Methods	Precision	Recall	F-measure
Matching titles	Baseline 1	69.4%	67.2%	68.3%
	Baseline 2	77.2%	72.5%	74.8%
	Adaptive	**83.4**%	**82.8**%	**83.1**%
Matching links	Baseline 1	61.7%	60.0%	60.8%
	baseline 2	69.7%	64.1%	66.8%
	Adaptive	**73.8**%	**72.5**%	**73.1**%
Matching levels	Baseline 1	62.0%	59.1%	60.5%
	Baseline 2	65.7%	63.9%	64.8%
	Adaptive	**72.7**%	**69.4**%	**71.0**%
Complete matching entries	Baseline 1	43.1%	40.5%	41.8%
	Baseline 2	44.8%	43.0%	43.9%
	Adaptive	**59.8**%	**54.2**%	**56.9**%

TABLE II: Results on academic PDF book dataset

C. Results

We compare our methods with the baselines on the two datasets and show the results based on different metrics, including matching titles, matching links, matching levels and complete matching entries. The results on the Structure Extraction dataset is shown in Tab. I and the results of academic PDF book dataset are shown in Tab. II. Since it is hard to manually label the entire ground truth on a large number of books, the result in Tab. II is based on 200 samples randomly selected from our large scale dataset. Since they contain multiple book document types from various domains, we argue they are representable of the whole dataset. In particular, we present three observations based on the results.

First, the results demonstrate that our method outperforms the two baselines in all metrics on both datasets. Baseline 1 performs worst most likely because it considers only section numbers and indentation thus lacks the capability to parse the TOCs without ordered section numbers and indentation. For example, it will fail on TOCs as shown in Fig. 3. Baseline 2 is a stronger than baseline 1 since it uses more features to identify the TOC level based on clustering. It is able to parse a "divided" TOC without section numbers and indentation, as long as the line height can indicate the TOC levels. However, it cannot handle a flat TOC, or a divided TOC whose blocks are divided by the largest line spaces. In addition, it is sensitive to the similarity threshold set in the clustering and to OCR or PDF extraction errors in the section and page numbers.

Second, all methods' performance decreases when the metrics require stricter matching. The F-measure of matching

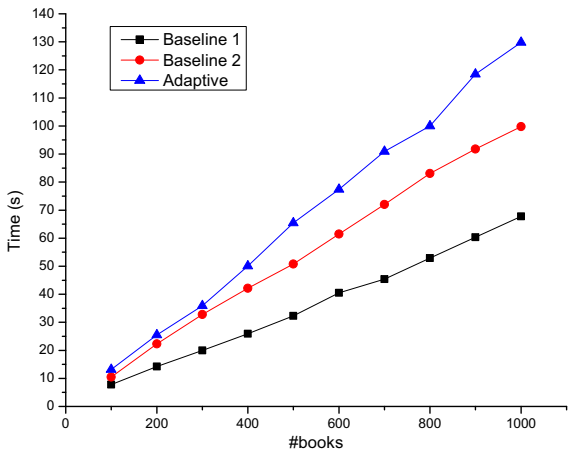

Fig. 4: Runtime on the Structure Extraction Dataset

titles is significantly higher than that of completely matched entries, indicating that identifying the correct page numbers and TOC level at the same time is much harder than extracting the title alone. However, matching links is only slightly higher than matching levels, suggesting that TOC linking is a relative easier task than TOC parsing. Notice that the difference between the two is slightly smaller on the academic book dataset than that on Structure Extraction dataset, suggesting there might exists more "ordered" books in academic books.

Third, it is worth noting that all methods perform better on the academic book dataset than on Structure Extraction dataset, which we believe is mainly due to their difference in data quality. Most of the academic books are in a modern PDF format while most of the DjVu XML files are OCRed from scanned images. Those modern academic PDF books have better formats and more regular TOCs than the older books from Internet Archive. Besides, we found that OCR errors occur more often than PDF extraction errors when manually checking samples from both datasets. We plot the runtime of all three methods on the Structure Extraction dataset. Fig. 4 shows that our method's runtime is the same magnitude as the other two baselines. On average, for every 100 books, our methods cost around 3 more seconds than Baseline 2. The results on the PDF academic book dataset is similar while the runtime is much larger, since it costs more time for PDF extraction (it costs 5 20 seconds for a book). All our experiments were conducted on 2.35GHz Intel(R) Xeon(R) 4 processor with 23GB of RAM, and Red Hat Enterprise Linux Server(5.7) installed. Runtime of all the three methods is linear to the input size, or the number of books. All are essentially rule based methods which need no inter-document information. This makes the process inherently parallelizable at the document level and could easily be done on a distributed environment for a large scale dataset.

V. Conclusions and Future Work

We proposed a new TOC recognition approach which adaptively selects TOC parsing methods according to the determined TOC styles. We presented three basic styles for TOCs, namely "flat", "ordered", and "divided", which, we contend, cover all possible TOCs. Extensive experimental study shows this approach is effective and efficient on two different book datasets containing over 25,000 books from various domains. However issues to be addressed include improving performance and handling decorative content detection and multiple TOCs recognition, which are relatively rare in nonfiction books. Decorative content usually includes the page header, foot, and border content, whose layout will usually be inconsistent with TOC entries. Some books may also have a TOC for each of its chapters. For those books, we only detect the first TOC. It would also be interesting to characterize TOCs as a function of domain and time of publication.

VI. Acknowledgments

We gratefully acknowledge partial support from the National Science Foundation (NSF).

References

[1] S. Satoh, A. Takasu, and E. Katsura An Automated Generation of Electronic Library Based on Document Image Understanding. In *ICDAR95*, pages 163-166, 1995.

[2] Q. Luo, T. Watanabel and T. Nakayama. Identifying Contents Page of Documents. In *ICPR96*, page 696-700, 1996.

[3] C. Lin, Y. Niwa and S. Narita. Logical Structure Analysis of Book Document Images Using Contents Information. In *ICDAR97*, pages 1048-1054, 1997.

[4] S. Tsuruoka, C. Hirano, T. Yoshikawa and T. Shinogi. Image-based Structure Analysis for a Table of Contents and Conversion to XML Documents. In *DLIA01*, Seattle, 2001.

[5] F. L. Bourgeois, H. Emptoz, and S. S. Bensafi Document Understanding Using Probabilistic Relaxation: Application on Tables of Contents of Periodicals In *ICDAR01*, pages 508-512, 2001.

[6] A. Belad. Recognition of Table of Contents for Electronic Library Consulting. *IJDAR*, 4(1):35-45, 2001.

[7] S. Mandal and S.P. Chowdhury and A.K. Das and B. Chanda. Automated Detection and Segmentation of Table of Contents Page from Document Images. In *ICDAR03*, pages 398-402, 2003.

[8] F. He, X. Q. Ding and L. R. Peng. Hierarchical Logical Structure Extraction of Book Documents by Analyzing Tables of Contents. In *SPIE Conference on Document Recognition and Retrieval*, pages 6-13, 2004.

[9] H. Djean and J. L. Meunier. Structuring Documents According to Their Table of Contents. In *DocEng05*, pages 2-9, 2005.

[10] S. Yacoub and J. A. Peiro. Identification of Document Structure and Table of Content in Magazine Archives. In *ICDAR05*, pages 1253-1257, 2005.

[11] Xiaofan Lin and Yan Xiong. Detection and Analysis of Table of Contents Based on Content Association. *IJDAR*, 8(2):132-143, 2006.

[12] Prateek Sarkar and Eric Saund. On the Reading of Tables of Contents. *IAPR International Workshop on Document Analysis Systems*, pages 386-393, 2008.

[13] Herve Dejean and Jean-Luc Meunier. On tables of contents and how to recognize them. *IJDAR*, 12(1):1-20, 2009.

[14] Liangcai Gao, Zhi Tang, Xiaofan Lin, Xin Tao and Yimin Chu. Analysis of Book Documents' Table of Content Based on Clustering. In *ICDAR'09*, pages 911-915, 2009.

[15] Simone Marinai, Emanuele Marino and Giovanni Soda. Table of contents recognition for converting PDF documents in e-book formats. In *DocEng'10*, pages 73-76, 2010.

[16] Yogalakshmi Jayabal, Chandrashekar Ramanathan and Mehul Jayprakash Sheth. Challenges in generating bookmarks from TOC entries in e-books. In *DocEng'12*, pages 37-40, 2012.

[17] Antoine Doucet, Gabriella Kazai, Bodin Dresevic, Aleksandar Uzelac, Bogdan Radakovic and Nikola Todic Setting up a Competition Framework for the Evaluation of Structure Extraction from OCR-ed Books *IJDAR*, 14(1):45-52, 2010.

[18] Antoine Doucet, Gabriella Kazai, Jean-Luc Meunier ICDAR 2011 Book Structure Extraction Competition In *ICDAR'11*, pages 1501-1505, 2011.