Document Structure Analysis Algorithms: A Literature Survey

Song Mao[a], Azriel Rosenfeld[a], and Tapas Kanungo[b]

[a]Center for Automation Research
University of Maryland, College Park, MD 20742
Email: maosong,ar@cfar.umd.edu

[b]IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
Email: kanungo@almaden.ibm.com

ABSTRACT

Document structure analysis can be regarded as a syntactic analysis problem. The order and containment relations among the physical or logical components of a document page can be described by an ordered tree structure and can be modeled by a tree grammar which describes the page at the component level in terms of regions or blocks. This paper provides a detailed survey of past work on document structure analysis algorithms and summarize the limitations of past approaches. In particular, we survey past work on document physical layout representations and algorithms, document logical structure representations and algorithms, and performance evaluation of document structure analysis algorithms. In the last section, we summarize this work and point out its limitations.

1. INTRODUCTION

Electronic documents have many advantages over paper documents, including compact and lossless storage, easy maintainance, efficient retrieval and fast transmission. As a result, there has been extensive research on converting paper-based documents into electronic documents.

One of the major advantages of electronic documents is that an electronic document can have an explicit structure; it can be partitioned into a hierarchy of physical components, such as pages, columns, paragraphs, textlines, words, tables, figures, halftones, etc.; a hierarchy of logical components, such as (for example) titles, authors, affiliations, abstracts, sections, etc.; or both. This structural information can be very useful in indexing and retrieving the information contained in the document. Document understanding modules, such as Optical Character Recognition (OCR) and graphics recognition modules, can also be selectively applied to the structural components of document images. Physical layout and logical structure analysis of document images is a crucial stage in a document image analysis system. Numerous algorithms have been proposed to analyze the physical layout and logical structure of document images in many different domains. Previous surveys[1–4] of these algorithms have been given in relatively smaller scale, or are not current and do not categorize the surveyed algorithms in detail.

In this paper, we provide a detailed survey of these algorithms in the following three aspect: document physical layout representation and analysis algorithms, document logical structure representation and analysis algorithms, and performance evaluation.

Song Mao is now with the Communications Engineering Branch, U. S. National Library of Medicine, Bethesda, Maryland.

2. DOCUMENT PHYSICAL LAYOUT REPRESENTATIONS AND ANALYSIS ALGORITHMS

Document physical layout can be represented in various forms, independently of or jointly with document logical structure. Document style parameters have been used to represent document physical layout in.[5–9] These style parameters typically correspond to sizes of and gaps between document objects such as characters, words, lines or zones. While this representation method provides useful information, it does not fully reflect the spatial relations among document physical components. Document physical layout can be more fully represented by trees that are derived from a set of rules, as in.[10–12] Such a representation describes the spatial relations, in many cases hierarchical, among document physical components.

A disadvantage of rule-based representations is that the rules can become rather arbitrary. Representations based on formal grammars have the advantage that the type of grammar (in the Chomsky hierarchy) limits the types of productions that can be used, and hence constrains the rules that the language can satisfy. Systems that use grammars to describe hierarchical document physical layout are described at the end of this section. In grammar-based algorithms, a document is usually regarded as a sequence, i.e. a string, of features of physical components.

Document image physical layout analysis algorithms can be categorized into three classes: top-down approaches, bottom-up approaches and hybrid approaches. Top-down algorithms start from the whole document image and iteratively split it into smaller ranges. The splitting procedure stops when some criterion is met and the ranges obtained at that stage constitute the final segmentation results. Bottom-up algorithms start from document image pixels, and cluster the pixels into connected components such as characters which are then clustered into words, lines or zones. Hybrid algorithms can be regarded as a mix of the above two approaches. The Docstrum algorithm of O'Gorman,[13] the Voronoi-diagram-based algorithm of Kise et al.,[14] the run-length smearing algorithm of Wahl et al.,[15] the segmentation algorithm of Jain and Yu,[3] and the text string separation algorithm of Fletcher and Kasturi[16] are typical bottom-up algorithms. The $X-Y$-cut-based algorithm of Nagy et al.[17] and the shape-directed-covers-based algorithm of Baird et al.[18] are top-down algorithms. Pavlidis and Zhou[19] proposed a hybrid algorithm using a split-and-merge strategy. Surveys of page segmentation algorithms can be found in O'Gorman and Kasturi[20] and Jain and Yu.[3] A recent workshop[21] was devoted to addressing issues related to physical layout analysis.

Most of the algorithms mentioned above do not create hierarchical descriptions or allow users to specify document structure information. Furthermore, they do not provide methods of estimating algorithm parameters from groundtruth data. A rigorous empirical comparison of five document physical layout analysis using the PSET software package[22] can be found in Mao and Kanungo.[23] Liang et al.[24] propose a performance metric for evaluating document structure extraction algorithms. They describe a method for finding the optimal tuning parameters of their algorithm. They evaluated several document layout analysis algorithms on 1600 images from the UW-III dataset. An OCR zoning evaluation method based on string matching is proposed by[25] and a yearly conference[26] is devoted to the evaluation of OCR accuracy of various OCR algorithms. Yanikoglu and Vincent[27] describe an environment (called Pink Pather) for ground-truthing and benchmarking document page segmentation. They use a bitmap-level region-based metric.

A few researchers have developed document physical layout analysis algorithms that make use of grammatical methods. Kopec and Chou[28] describe an algorithm for segmenting a column of text that is modeled using a stochastic regular grammar. However, their algorithm assumes that it is given templates for the symbols in the language; this is not always the case, for example if we must analyze document pages in a previously unknown language. The algorithm also assumes that the page is segmented into columns by some other procedure, and it does not provide any estimation procedure for the model parameters.

Tokuyasu and Chou[29] recently proposed a communication theory approach to page segmentation. They used regular grammars to describe the structure of document page images in terms of axis-parallel rectangles obtained by subdividing the image vertically and horizontally, and they used a Turbo decoding approach to estimate the 2D image from the observations. However, they provided very limited experimental verification of their approach.

Krishnamoorthy et al.[30] describe a hierarchical document page segmentation algorithm that constructs a tree in which each node represents an axis-parallel rectangle. Users can specify grammars for individual blocks. However, in the presence of noise their parsing algorithm can fail, and no method of parameter estimation is provided. No objective function is minimized; thus the analysis is not optimal.

Spitz[31] described a system for style-directed recognition. While the user can specify the style interactively, the algorithm itself is a rule-based system.

3. DOCUMENT LOGICAL STRUCTURE REPRESENTATIONS AND ANALYSIS ALGORITHMS

Document logical structure can be represented by logical labels of document physical components. These logical labels usually are derived from a set of rules.[5, 8, 9, 32, 33] In this representation method, there is no description of semantic relations among logical components. To reflect these relations, document logical structures are represented by trees that are derived either from a set of rules[6, 10–12, 34] or from formal grammars.[30, 35–37] The document is regarded as a sentence which can be either a string of logical labels or a string of observed features of document physical components.

The grammatical rules used in most algorithms for either physical layout analysis or logical structure analysis[30, 35, 36] are deterministic. It is difficult for deterministic parsing methods to remove ambiguity in the parsing results; for the same input, multiple parse trees can be generated. In some applications, the input sentence is probabilistic (for example, derived from a preceding physical layout analysis), some inputs are not accurate, and they often have errors. Deterministic parsing cannot handle any of these situations. Tateisi and Itoh[37] augmented the grammars by a set of cost attributes and were able to select a parsing result that had least cost.

Tsujimoto and Asada[10] represented document physical layout and logical structure as trees. They posed document understanding as the transformation of a physical tree into a logical one using a set of generic transformation rules and a virtual field separator technique. The physical tree is constructed using block dominating rules. The blocks in the tree are classified into *head* and *body* using rules related to the physical properties of the block. Once the logical tree is obtained, logical labels are assigned to the blocks using another set of rules. The logical labels include *title*, *abstract*, *sub-title*, *paragraph*, *header*, *footer*, *page number*, and *caption*. To effectively use the information carried by field separators and frames, a virtual field separator technique, in which separators and frames are considered as virtual physical blocks in the physical layout tree, is used for tree transformation without increasing the number of transformation rules. They tested their algorithm on 106 pages from various sources and reported a 94/106 logical structure recognition accuracy. Errors were due to inaccurate physical segmentation, insufficient transformation rules, and the fact that some pages did not have hierarchical physical and logical structures.

Yamashita et al.[11] proposed a model-based method for logical structure analysis. The model is a tree-structured layout model which defines the minimum necessary information about the geometrical arrangement of document objects. Specifically, the model describes each document object's logical label, tree level, separator location, minimum and maximum numbers of constituent character strings, as well as its successor's orientation. The physical segments are character strings, lines, and picture elements, and they are segmented using extracted horizontal and vertical separators. The picture elements are removed. Logical labels are then assigned to character strings consistently with the layout model using a relaxation method. If contradictory labeling occurs, all related labels are deleted. If two or more labels are assigned to a character string, a confidence value is computed for all possible labeling paths and the path with the highest confidence value is retained. Seventy-seven Japanese patent application front pages were used for testing the algorithm, and fifty-nine of them were correctly labeled. Errors were due to incorrect recognition of the page number as part of the body, skew, blots, and connected character strings.

Kreich et al.[5] described an experimental environment called SODA (System for Office Document Analysis) for model-based document analysis. They first used a bottom-up approach to group connected components into text blocks, then found lines within each text block and words within each line. OCR and graphics recognition were performed on the document segments. The domain knowledge and metaknowledge, the physical layout and logical structure knowledge were stored in a knowledge base. Document objects were matched to the layout

and logical information in the knowledge base. A generalized Hamming metric was used in the matching process to calculate a confidence measure. A match was considered successful if its confidence measure was greater than a threshold. The experimental result on one letter was displayed. Otherwise, no quantitative performance data were reported.

Fisher[12] presented a rule-based system for recognizing the physical layout and logical structure of a document image without prior information about the document's format or content. The system automatically extracts the general physical layout of the document and transforms it into a logical structure. Three types of rules are used in the system: location cues, format cues, and textual cues. The system can reconstruct paragraphs broken during formatting, determine the read order of text blocks, and express its results in a document markup language. Text and nontext regions are assumed to be already identified. First, words are grouped into paragraphs or columns. Then text column boundaries and locations are identified. The physical layout and logical structure are determined using the appropriate rules. The algorithm's performance was highly dependent on the accuracy of the text/nontext segmentation process. The analysis results were expressed in Maker Interchange Format (MIF). No experimental results were given.

Derrien-Peden[34] proposed a frame-based system for analyzing document physical layout and logical structure. The document layout structure is obtained in three steps. First document columns and text blocks are obtained by recursively performing an $X-Y$ cut. Then lines are extracted using special rules. Finally, physical zones are obtained by analyzing their topographical features. The reading order is obtained by a depth-first search of the layout structure. Logical structure recognition is conducted in two steps: 1) paragraphs with the same features are grouped into classes, 2) logical labels are assigned to each class using a set of general layout rules. A knowledge base containing both the physical layout and logical structure models is used during the analysis procedure. No experimental results were reported for this algorithm.

Ingold and Armangil[35] proposed a document logical structure recognition method using a formal description of each document class that includes composition rules and presentation rules. The composition rules define the generic logical structure, and the presentation rules define the physical characteristics of the logical entities to be recognized. The composition rules are formally represented by Extended Backus-Naur Form grammars. The document description completely defines an analysis graph whose vertices are labeled with the classes of entities to be recognized. The successor of an entity is the logical label of the entity that will be evaluated after the current entity. Alternatives specify possible replacements for the current entity. Document analysis is achieved by finding a path through such a graph under the constraint that the typographic attributes of an entity on the path must match those of the corresponding document object. The authors assumed that the physical zones were already segmented out and OCR was performed on the logical objects. No experimental results were reported.

Brugger et al.[38] described a document logical structure model based on a statistical representation of patterns in a document class, i.e. on generalized N-grams. In an N-gram model, only the previous $N-1$ words can affect and can be used to estimate the probability of the current word in a sentence. The tree structure of document logical components is represented by the probabilities of local tree node patterns similar to N-grams. The logical tree is constructed from physical entities in conformity with the given model. There can be multiple valid trees, but only the tree with the best conformity with the model is selected. The model can be learned from samples. The physical segments were assumed to be available. Five memo pages were used in the experiments; one of them was used for training the model and the remaining four for testing the model.

Conway[36] used page grammars and page parsing techniques to recognize document logical structure from physical layout. The physical layout is described by a set of grammar rules, each of which is a string of components specified by a neighbor relationship. Possible neighbor relationships include above, left-of, over, left-side, and close-to, so that the layout is two dimensional. Context-free string grammars are used to describe logical structure. Both grammars are deterministic. The physical layout grammar has attached constraints to incorporate information such as font size, style, alignment and indentation. The physical segmentation is performed independently of logical structure recognition using a run length smoothing algorithm. No quantitative experimental results were reported.

Krishnamoorthy et al.[30] proposed a document logical structure recognition method that recursively applies grammars to horizontal and vertical projection profiles of the page. The parsing process is divided into four stages. In the first stage, the lengths of runs of zeros or ones in the thresholded projection profiles are thresholded into atoms. In the second stage, the atoms are grouped into molecules. In the third stage, logical labels are assigned to the molecules. In the fourth stage, contiguous entities of the same type are merged. The results of the segmentation and logical labeling processes are saved in a labeled $X-Y$ tree. This method transforms a two-dimensional segmentation and labeling problem into a one-dimensional segmentation and labeling problem in an $X-Y$ tree. The authors did not distinguish between physical layout and logical structure. Their algorithm was trained on twenty-one IBM journal pages, and was tested on twelve IBM/PAMI pages. The algorithm performance was reported in terms of percentage of labeled area and missed labels.

Saitoh et al.[7] presented a system for document segmentation, text area classification and ordering. This system is independent of the shapes of the physical blocks and is robust to document skew. Connected components are first extracted and classified. The connected components are then merged into lines which are merged into zones. The extracted zones are classified into body, caption, header and footer. A tree structure is generated from the classified zones using text area influence ranges. The order of the text is obtained by preorder traversal of the tree. The experimental dataset included 131 Japanese and English documents which were scanned with skew. The size of the final dataset was 393 images. The authors used three criteria to evaluate their segmentation and classification results, and three other criteria to evaluate their text ordering results.

Tateisi and Itoh[37] posed document logical structure analysis as a stochastic syntactic analysis problem. The document is modeled as a string of text lines and graphic objects. The text lines and graphic objects are segmented and classified in a preprocessing step, and the string is parsed using a stochastic regular grammar with attributes. Characters within text lines are recognized and their font sizes are determined. Each grammatical rule is associated with a cost. The parser retains possible parsing results in order of their total cost. The algorithm was tested on seventy pages of Japanese text taken from books and magazines. The authors reported an 86% average markup accuracy on manuals and an 82% average markup accuracy on technical papers for the parsing result with the least cost. When the parsing result with the second least cost was used, the average markup accuracy for the technical journals increased to 89%.

Niyogi and Srihari[6] presented a system called DeLoS for document logical structure derivation. In this system, a computational model is developed based on a rule-based control structure as well as a hierarchical multi-level knowledge representation scheme. In this scheme, knowledge about the physical layouts and logical structures of various types of documents is encoded into a knowledge base. The system included three levels of rules: Knowledge rules, control rules, and strategy rules. The control rules control the application of knowledge rules and the strategy rules determine the usage of control rules. A document image is first segmented using a bottom-up algorithm. The segmented blocks are then classified. Finally, the classified blocks are input into the DeLoS system and a logical tree structure is derived. The DeLoS system was tested on 44 newspaper pages. The performance results were reported in terms of block classification accuracy, block grouping accuracy, and read-order extraction accuracy.

Summers[33] described an algorithm for automatic derivation of logical document structure from generic physical layout. The algorithm is divided into segmentation of text into zones and classification of these zones into logical components. The document logical structure is obtained by computing a distance measure between a physical segment and predefined prototypes. For each logical label, a set of prototypes is specified. The prototypes include contours, context, successor, height, symbols, and children. The algorithm was tested on 196 pages from computer science technical reports. The input was the segmented text blocks. The labeling result of each text segment was characterized as correct, overgeneralized, or incorrect. Two metrics, precise accuracy and generalized accuracy, were used to evaluate the performance. Accuracies above 85% were reported.

Dengel and Dubiel[39] described a system (DAVOS) that is capable of both learning and extracting document logical structure. DAVOS is a concept formation system that learns document structure concepts by detecting distinct attribute values in document objects. The structural concepts are represented by relation patterns defined by a cut-and-label language. A GTree (Geometric Tree) is used to represent the concept language.

Unsupervised decision tree based learning techniques are used to build the GTree. Two learning techniques were compared, a bottom-up approach and a top-down approach (DAVOS). The authors used forty letters to train both systems. They then used the learned GTrees to classify another set of forty unknown letters. The evaluation results were reported in terms of precision, recall, and F value metrics. The DAVOS system outperformed the bottom-up system.

Lin et al.[8] proposed a method of analyzing the logical structure of book pages using contents page information. The contents page of a book contains a concise and accurate logical structure description of the whole book. Text lines are first extracted from the contents page, and OCR is then performed for each text line. The structures of the page number, head, foot, headline, chart and main text of the text page are analyzed and matched with information obtained from the contents page. The algorithm was tested on 235 pages. The experimental results were reported in terms of two labeling errors and the logical labeling identification rate.

Ishitani[9] proposed a document logical structure analysis system based on emergent computation. The system includes five interacting modules: typography analysis, object recognition, object segmentation, object grouping, and object modification. The interaction results in an adaptive system configuration which provides robust document analysis. The document image is first segmented into text lines, which are then classified into different types using special rules. The classified text lines are then grouped and classified into logical components using heuristic rules. The document objects that are incorrectly segmented can be modified by checking for logical consistency among objects. Modified objects are sent to other modules and new objects are created by module interactions. Since new logical structures are created, interactive computation among modules is induced by feedback between levels. This system was tested on 150 documents taken from various sources. The author reported a 96.3% average rate of correct logical object extraction.

Srihari et al.[40] proposed a information-theory-based method for automatic address interpretation in postal address fields of mail pieces. Shannon's entropy theory is used to characterize address components and their interaction. Interested logical components are city name, state abbreviation, ZIP code, ZIP+4 add-on, primary number, street name, building/firm name, et al. Experimental results are shown on a US postal address directory. Other postal address analysis methods include.[41]

Kim et al.[32] proposed a rule-based automated labeling module in MARS system (Medical Article Record System) to extract bibliographic records for the MEDLINE database. They derived rules from the results of a page layout analysis of medical journals and features extracted from OCR output. Therefore, both geometric and non-geometric features of journals are used in their labeling process. This system was tested on more than 11,000 articles in over 1,000 biomedical journals. The author reported a labeling accuracy that exceeds 96%.

4. ALGORITHM PERFORMANCE EVALUATION

The performance evaluation of an algorithm should address the following aspects: performance metric, experimental dataset, groundtruth specification, performance results, error analysis, and comparative evaluation. In this section, we survey document structure analysis algorithms with respect to these aspects. Document structure analysis of a particular type such as table recognition and their performance evaluation have been described in.[42, 43]

A meaningful and computable metric is necessary for quantitatively evaluating the performance of any algorithm. It is a function of the given dataset, the groundtruth and the algorithm parameters. A performance metric is typically not unique, and researchers can select particular performance metrics to study particular aspects of the evaluated algorithms.

Krishnamoorthy et al.[30] proposed a metric based on the percentage of area labeled and missed labels. Saitoh et al.[7] used three criteria to show the results of their algorithm, based on three proposed ways of using their experimental results. Niyogi and Srihari[6] reported their results using three metrics: block classification, block grouping, and read-order accuracy. Lin et al.[8] used two types of labeling errors and an identification rate to report the experimental results of their algorithm. A common aspect of these metrics is their lack of formal definitions; verbal descriptions are used instead.

Yamashita et al.[11] described a cost function based metric for selecting the result with the least cost. Kreich et al.[5] used a generalized Hamming metric to compute a confidence measure for matches between a document physical layout and logical structure knowledge base and a document object. Summers[33] defined precise and generalized accuracy metrics and reported the performance of his algorithm using these metrics. Dengel and Dubiel[39] used recall, precision and F value to evaluate the performance of their algorithm. These metrics are relatively formally defined and hence have less ambiguity in their interpretations. In,[9,10,37] experimental results were reported, but no clear definition of the performance metrics used was given. In,[12,34–36,38] no quantitative experimental results were reported, and hence it is hard to assess the performance of the algorithms.

Evaluation based on large-scale experimental datasets is crucial for objectively evaluating the performance of algorithms and assessing the state of the art. The groundtruth of a given dataset is necessary for scoring experimental results using that dataset. Some authors tested their algorithms on relatively large datasets. In,[7–10,33] more than 100 document images were used, and in,[6,11,30,37,39] tens of document images were used. Other authors,[5,12,38] however, tested their algorithms on very small datasets. In,[34–36] no dataset was specified. None of the authors clearly specified the groundtruth of the datasets used for testing their algorithms. Performance results and error analysis (if any) can be found in the descriptions of the individual algorithms.

Comparative performance evaluations are necessary for comparing the performance of algorithms on some common ground and identifying state-of-the-art techniques. However, for most algorithms, there is a lack of comparative evaluation. Dengel and Dubiel[39] performed a comparative evaluation of the bottom-up and top-down versions of his algorithm through learning and testing procedures.

In Table 1, we summarize the experiments and performance evaluations that have been performed for various logical structure analysis algorithms.

5. SUMMARY AND LIMITATIONS OF THE SURVEYED ALGORITHMS

Table 2 summarizes the surveyed algorithms in terms of key idea, physical layout representation, logical structure representation, logical labels, output representation, and application domain.

As pointed out in Section 2 and Section 3, most of the past work on document structure analysis has been limited in one or more respects:

1. Much of the work has not been based on formal models for document pages. The use of formal models has several important advantages:
   (a) In a formal model framework, one can use a model that has an appropriate level of complexity for a given class of documents.
   (b) Once a model has been chosen for a given document class, examples of the class can be used to estimate model parameters.
   (c) Formal models can be used for both analysis and synthesis of documents. A model can be validated by using it to synthesize document page images that can be compared to real page images of the given class. The model can also be used to generate synthetic page image data which can be used in controlled experiments.
2. Much of the work on logical structure analysis of documents assumes that physical layout analysis has already been performed.
3. Most of the work makes use of deterministic models. Such models fail in the presence of noise or ambiguity.
4. In some of the work, quantitative performance evaluation issues have been neglected.

While most document structure analysis algorithms are based explicitly or implicitly on document models, relatively few of them have provided formal definitions of these models. This has made it difficult to characterize the relation between the models and the performance of the algorithms. Furthermore, the parameter values in the algorithms have usually been manually selected.

Table 1. This table summarizes experiments involving various logical structure analysis algorithms in terms of experimental dataset, performance metric, groundtruth specification, performance results, error analysis and comparative evaluation. Note: N/S means not specified.

Authors	Year	Experimental Dataset	Performance Metric	Groundtruth Specification	Performance Results	Error Analysis	Comparative Evaluation
Tsujimoto and Asada[10]	1990	106 pages from various sources	N/S	N/S	94/106 accuracy	yes	none
Yamashita et al.[11]	1991	77 Japanese patent application front pages	cost function	N/S	59/77 accuracy	yes	none
Kreich et al.[5]	1991	one page	confidence measure	N/S	N/S	no	none
Fisher[12]	1991	one page	N/S	N/S	N/S	no	none
Derrien-Peden[34]	1991	none	N/S	N/S	N/S	no	none
Ingold and Armangil[35]	1991	none	N/S	N/S	N/S	no	none
Brugger et al.[38]	1993	five memo pages — one for training, four for testing.	N/S	N/S	N/S	no	none
Conway[36]	1993	none	N/S	N/S	N/S	no	none
Krishnamoorthy et al.[30]	1993	21 IBM journal pages for training, 12 IBM/PAMI pages for testing	% area labeled, missed labels	N/S	reported for each of 12 IBM journal and IEEE PAMI pages	no	none
Saitoh et al.[7]	1993	393 Japanese/ English pages for testing	six criteria based on result usage	N/S	results reported based on three criteria	yes	none
Tateisi and Itoh[37]	1994	70 Japanese pages from books/magazines	N/S	N/S	87% and 82% logical labeling accuracy for manuals and technical papers etc.	yes	none
Niyogi and Srihari[6]	1995	44 newspaper pages	block classification, block grouping, read order accuracy	N/S	reported for each, of 32 newspaper pages and read order accuracy	yes	none
Summers[33]	1995	196 pages from technical reports with corrected segmentation	Precise and generalized accuracy	N/S	85.5% logical labeling accuracy	no	none
Dengel and Dubiel[39]	1996	40 letters for learning, 40 letters for testing.	recall, precision, F value	N/S	reported for 40 letters	no	yes
Lin et al.[8]	1997	235 book pages	two types of errors, identification rate	N/S	reported for 235 pages	yes	none
Ishitani[9]	1999	150 pages from various sources	N/S	N/S	96.3% logical object extraction accuracy	no	none
Srihari et al.[40]	1999	US postal address directory	N/S	N/S	ZIP code, city name state, stree name	no	none
Kim et al.[32]	2001	over 11,000 pages from over 1,000 biomedical journals	labeling	N/S	96.7% labeling accuracy	yes	none

Document physical and logical structures vary greatly in complexity. If we could characterize the complexity of the document images in a given dataset, we could use appropriate analysis techniques. Existing document structure analysis algorithms have not addressed this issue; it too could be addressed if formal models were used.

The use of generative document models would enable us to simulate document images and perform controlled experiments to evaluate algorithms and study their breakdown points.

Deterministic models often cannot handle noise or ambiguity. Document pages are usually noisy due to printing, handling, photocopying, scanning, and faxing processes, and this can lead to ambiguous or false results. Document physical structure analysis procedures also have performance uncertainties and so may provide uncertain input to the logical structure analysis process. Stochastic models, represented by stochastic grammars and related parsing techniques,[44] could be used to address these problems. The input to the parser could be regarded as probabilistic to reflect uncertainty due to erroneous physical layout analysis results and document noise. Physical layout and logical structure analysis algorithms based on stochastic language models

Table 2. In this table, state-of-the-art document logical structure analysis algorithms are analyzed in terms of key idea, physical layout representation, logical structure representation, output representation, logical labels and application domain.

Authors	Year	Key Idea	Physical Layout Representation	Logical Structure Representation	Output Representation	Logical Labels	Application Domain
Tsujimoto and Asada[10]	1990	mapping a physical tree to a logical one	block dominating rules, tree	tree	not mentioned	title, abstract, sub-title, paragraph, header, footer page number, caption	various documents
Yamashita et al.[11]	1991	top-down layout model and relaxation labeling	tree	tree	ODA	title, author, affiliation, body column, block	patent applications
Kreich et al.[5]	1991	knowledge based analysis	document style parameters	logical labels	not given	sender, date, reference	not mentioned
Fisher[12]		rule-based	rules, tree	rules, labeling	MIF	section heading, figure, figure caption, page heading, page footings	not mentioned
Derrien-Peden[34]	1991	frame and macro-typographical based	tree	rules, labeling	MML	title, list, paragraph abstract	not mentioned
Ingold and Armangil[35]	1991	rule based, physical zones available	none	EBNF grammars, presentation rules	not mentioned	title, paragraph, section, chapter	not mentioned
Brugger et al.[38]	1993	N-gram model, physical zones available	none	tree	not mentioned	not mentioned	memo pages
Conway[36]	1993	page grammar	page grammars	context-free string grammar	SGML	title, heading, paragraph, figure	not mentioned
Krishnamoorthy et al.[30]	1993	page parsing, block grammar	block grammar, tree	block grammar, tree	not mentioned	title, author, abstract	journal pages
Saitoh et al.[7]	1993	text area influence rules	document style parameters	tree	not mentioned	body, caption, header footer	various documents
Tateisi and Itoh[37]	1994	stochastic grammars, physical zones available	none	grammar rules	not mentioned	headings, paragraph, list item	not mentioned
Niyogi and Srihari[6]	1995	rule-based, knowledge-based	rules	rules, tree	not mentioned	title, story, sub-story, photo, caption, graph	newspaper pages
Summers[33]	1995	logical prototype, matching, physical zones available	none	logical prototypes	not mentioned	paragraph, heading, list item	technical reports
Dengel and Dubiel[39]	1996	logical structure learning, physical zones available	none	GTree	not mentioned	sender, recipient, date logo, subject, footer body-text	letters
Lin et al.[8]	1997	OCR and rule based	document style parameters	logical labels	not mentioned	headline, content, figure, table, page number, head-foot	book pages
Ishitani[9]	1999	emergent computation, rule based	document style parameters	logical labels	not mentioned	headline, header, footer note, caption, program, formula, title, list	various documents
Kim et al.[32]	2001	OCR and rule based	zones	logical labels	database tables	title, author affiliation, abstract	biomedical journals

have been recently proposed in.[45, 46] Doermann et al.[47] proposed a method for lexicon acquisition from bilingual dictionaries based on learning.

A soundly designed experimental methodology should include: a meaningful and computable performance metric, large datasets with well-defined groundtruth, a training procedure and a testing procedure, a thorough error analysis and, finally, comparisons with other state-of-the-art algorithms. As described in Section 2.2.3, very few algorithms have used such complete experimental designs.

ACKNOWLEDGMENTS

This research was funded in part by the Department of Defense under Contract MDA 9049-6C-1250, Lockheed Martin under Contract 9802167270, the Defense Advanced Research Projects Agency under Contract N660010028910, and the National Science Foundation under Grant IIS9987944.

REFERENCES

1. R. M. Haralick, "Document image understanding: Geometric and logical layout," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 385–390, (Seattle, WA), June 1994.

2. G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 38–62, 2000.
3. A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, pp. 294–308, 1998.
4. H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation methods for character recognition: from segmentation to document structure analysis," in *Proceeding of the IEEE*, vol. 80, pp. 1079–1092, 1992.
5. J. Kreich, A. Luhn, and G. Maderlechner, "An experimental environment for model based document analysis," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 50–58, (Saint-Malo, France), September 1991.
6. D. Niyogi and S. N. Srihari, "Knowledge-based derivation of document logical structure," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 472–475, (Montreal, Canada), August 1995.
7. T. Saitoh, M. Tachikawa, and T. Yamaai, "Document image segmentation and text area ordering," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 323–329, (Tsukuba Science City, Japan), October 1993.
8. C. C. Lin, Y. Niwa, and S. Narita, "Logical structure analysis of book document images using contents information," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 1048–1054, (Ulm, Germany), August 1997.
9. Y. Ishitani, "Logical structure analysis of document images based on emergent computation," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 189–192, (Bangalore, India), September 1999.
10. S. Tsujimoto and H. Asada, "Understanding multi-articled documents," in *Proceedings of International Conference on Pattern Recognition*, pp. 551–556, (Atlantic City, NJ), June 1990.
11. A. Yamashita, T. Amano, I. Takahashi, and K. Toyokawa, "A model based layout understanding method for the document recognition system," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 130–138, (Saint-Malo, France), September 1991.
12. J. L. Fisher, "Logical structure descriptions of segmented document images," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 302–310, (Saint-Malo, France), September 1991.
13. L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**, pp. 1162–1173, 1993.
14. K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area Voronoi diagram," *Computer Vision and Image Understanding* **70**, pp. 370–382, 1998.
15. F. Wahl, K. Wong, and R. Casey, "Block segmentation and text extraction in mixed text/image documents," *Graphical Models and Image Processing* **20**, pp. 375–390, 1982.
16. L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**, pp. 910–918, 1988.
17. G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer* **25**, pp. 10–22, 1992.
18. H. S. Baird, S. E. Jones, and S. J. Fortune, "Image segmentation by shape-directed covers," in *Proceedings of International Conference on Pattern Recognition*, pp. 820–825, (Atlantic City, NJ), June 1990.
19. T. Pavlidis and J. Zhou, "Page segmentation and classification," *Graphical Models and Image Processing* **54**, pp. 484–496, 1992.
20. L. O'Gorman and R. Kasturi, *Document Image Analysis*, IEEE Computer Society Press, Los Alamitos, CA, 1995.
21. T. Breuel and M. Worring, eds., *Document Layout Interpretation and its Applications*, (Bangalore, India), September 1999.
22. S. Mao and T. Kanungo, "Software architecture of PSET: A page segmentation evaluation toolkit," *International Journal on Document Analysis and Recognition* **4**, pp. 205–217, 2002.
23. S. Mao and T. Kanungo, "Empirical performance evaluation methodology and its application to page segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**, pp. 242–256, 2001.
24. J. S. Liang, I. T. Phillips, and R. M. Haralick, "Performance evaluation of document structure extraction algorithms," *Computer Vision and Image Understanding* **84**, pp. 144–159, 2001.
25. J. Kanai, S. V. Rice, T. A. Nartker, and G. Nagy, "Automated evaluation of OCR zoning," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**, pp. 86–90, 1995.
26. S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The fifth annual test of OCR accuracy," Tech. Rep. TR-96-01, University of Nevada, Las Vegas, NV, 1996.
27. B. A. Yanikoglu and L. Vincent, "Pink pather: A complete environment for ground-truthing and benchmarking document page segmentation," *Pattern Recognition* **31**, pp. 1191–204, 1998.
28. G. E. Kopec and P. A. Chou, "Document image decoding using Markov source models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, pp. 602–617, 1994.
29. T. A. Tokuyasu and P. A. Chou, "Turbo recognition: a statistical approach to layout analysis," in *Proceedings of SPIE Conference on Document Recognition and Retrieval*, (San Jose, CA), January 2001.

30. M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**, pp. 737–747, 1993.
31. A. L. Spitz, "Style-directed document segmentation," in *Proceedings of 2001 Symposium on Document Image Understanding Technology*, (Baltimore, MD), April 2001.
32. J. Kim, D. X. Le, and G. R. Thoma, "Automated labeling in document images," in *Proceedings of SPIE Conference on Document Recognition and Retrieval VIII*, pp. 111–122, (San Jose, CA), January 2001.
33. K. Summers, "Near-wordless document structure classification," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 462–465, (Montreal, Canada), August 1995.
34. D. Derrien-Peden, "Frame-based system for macro-typographical structure analysis in scientific papers," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 311–319, (Saint-Malo, France), September 1991.
35. R. Ingold and D. Armangil, "A top-down document analysis method for logical structure recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 41–49, (Saint-Malo, France), September 1991.
36. A. Conway, "Page grammars and page parsing: A syntatic approach to document layout recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 761–764, (Tsukuba Science City, Japan), October 1993.
37. Y. Tateisi and N. Itoh, "Using stochastic syntactic analysis for extracting a logical structure from a document image," in *Proceedings of International Conference on Pattern Recognition*, pp. 391–394, (Jerusalem, Israel), October 1994.
38. R. Brugger, A. Zramdini, and R. Ingold, "Modeling documents for structure recognition using generalized n-gram," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 56–60, (Ulm, Germany), August 1997.
39. A. Dengel and F. Dubiel, "Computer understanding of document structure," *International Journal of Imaging Systems and Technology* **7**, pp. 271–278, 1996.
40. S. N. Srihari, W. Yang, and V. Govindaraju, "Information theoretic analysis of postal address fields for automatic address interpretation," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 309–312, (Bangalore, India), September 1999.
41. P. G. Mulgaonkar, "Automatic detection of address blocks on irregular mail pieces," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 672–674, (Miami, Fl), June 1986.
42. J. Hu, R. Kashi, D. P. Lopresti, and G. T. Wilfong, "Evaluating the performance of table processing algorithms," *International Journal on Document Analysis and Recognition* **4**, pp. 140–153, 2001.
43. M. Hurst, "Layout and language: An efficient algorithm for detecting text blocks based on spatial and linguistic evidence," in *Proceedings of SPIE Conference on Document Recognition*, pp. 56–67, (San Jose, CA), January 2001.
44. D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Prentice Hall, Upper Saddle River, NJ, 2000.
45. S. Mao and T. Kanungo, "Stochastic language models for automatic acquisition of lexicons from printed bilingual dictionaries," in *Document Layout Interpretation and Its Applications*, (Seattle, WA), September 2001.
46. T. Kanungo and S. Mao, "Stochastic language model for style-directed physical layout analysis of document images," *IEEE Transactions on Image Processing* . To appear.
47. D. S. Doermann, H. Ma, B. Karagol-Ayan, and D. W. Oard, "Translation lexicon acquisition from bilingual dictionaries," in *Proceedings of SPIE Conference on Document Recognition and Retrieval VIII*, pp. 37–48, (San Jose, CA), January 2002.