2011 International Conference on Document Analysis and Recognition

Mathematical Formula Identification in PDF Documents

Xiaoyan Lin, Liangcai Gao*, Zhi Tang
Institute of Computer Science and Technology
Peking University
Beijing, China
{linxiaoyan, gaoliangcai,
tangzhi}@icst.pku.edu.cn

Xiaofan Lin
Vobile Inc
Santa Clara, CA, USA
xiaofan@vobileinc.com

Xuan Hu
College of Software
Beihang University
Beijing, China
huxuan@sse.buaa.edu.cn

***Abstract*—Recognizing mathematical expressions in PDF documents is a new and important field in document analysis. It is quite different from extracting mathematical expressions in image-based documents. In this paper, we propose a novel method by combining rule-based and learning-based methods to detect both isolated and embedded mathematical expressions in PDF documents. Moreover, various features of formulas, including geometric layout, character and context content, are used to adapt to a wide range of formula types. Experimental results show satisfactory performance of the proposed method. Furthermore, the method has been successfully incorporated into a commercial software package for large-scale Chinese e-Book production.**

Keywords-mathematical expression recognition; formula extraction; PDF document

I. INTRODUCTION

Nowadays an increasing number of documents are available in the PDF format, which can greatly facilitate document exchange and printing. Consequently research on PDF document analysis is receiving more and more attention [1] and significant progress has been made in recognizing basic components of the PDF documents (e.g., headings, paragraphs, table, etc) [2, 3]. However, as a crucial component of the documents, mathematical formulas are still recognized at accuracy too low to be useful in practical applications. In order to make use of this valuable resource in PDF documents, it is imperative to introduce better mathematical expression recognition methods. Identifying the regions of mathematical expressions is the first step in this task.

An advantage of PDF document analysis is that the character and layout information obtained from the PDF parser is much richer and more accurate than that acquired from OCR. In this sense, we can expect better results from PDF document recognition.

However, there still remain some challenges when extracting formulas from PDF documents. First, in the PDF content stream, a mathematical expression element may be composed of several different types of objects (e.g., text, image, graph). Therefore, the content stream extracted from the PDF document cannot be directly used as the logical mathematical expression elements. For example, in a PDF document generated by LaTeX, a root symbol is made up of a graph object representing the horizontal line, and a text object which is a radical character. The mismatch between PDF objects and mathematical elements is one of the biggest obstacles in formula extraction from PDF documents. Second, PDF documents are usually generated by different tools, and the objects used to render the mathematical expressions vary significantly in the different PDF generation programs. In addition, several PDF versions are widely used at present and they have different internal structures. To properly handle each type of PDF documents, the task of matching PDF objects to mathematical expression elements becomes even more difficult.

To overcome the above problems, Rahman et al. [2] presented that a PDF document can be rendered to an image, and then be analyzed by traditional recognition methods designed for image documents. However, this method would lose a lot of useful information which can be extracted directly from PDF documents. In this paper, a preprocessing step is proposed to focus on the above problems.

Baker et al. [4] proposed a formula recognition method for PDF documents for the first time. However, they assumed that the formula regions are already manually clipped out before recognition. To our best knowledge, there is no published work addressing how to identify formula regions directly from PDF documents. In this paper, a method is proposed to identify regions of both the isolated and embedded mathematical expressions in PDF documents. Preprocessing is first applied so that precise information of PDF documents can be fully utilized. Then by using various types of features (e.g., layout, characters, and context), we combine both rule-based and learning-based methods to adapt to a wide range formula types.

The rest of paper is organized as follows: Section II reviews relevant work. Section III introduces our formula identification method for PDF documents. Experimental results are presented in Section IV. We conclude this paper with a future research plan in Section V.

II. RELATED WORK

Traditional formula identification methods focus on image-based documents. According to the types of features used, the existing methods can be classified into three categories: character-based, layout-based, and image-based.

The first category of methods [5-7] identifies the formula

* Liangcai Gao is the corresponding author.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.285

regions mainly through the character features (e.g., specific math symbols or function names). These methods recognize characters by OCR engine and the outliers from OCR are considered as the candidates of mathematical expression elements. In [5], characters not recognized as Japanese are regarded as math symbols. Along this direction, Suzuki et al. [6] added verification rules according to character positions and sizes in order to develop a dedicated OCR system for mathematics documents. Kacem et al. [7] constructed a fuzzy logic model to identify math symbols, and then utilized math symbols' features (bounding box, relationship between symbols, etc.) to merge or expand the character regions to form the formula area. The shortcoming of character-based methods is that they overemphasize individual characters' features without considering other global features such as geometric layout. Besides, they heavily depend on character recognition results of the OCR system, in which recognition errors are inevitable.

The second category of methods [8-12] detects formula areas through layout features (line heights, line spacing, alignments, etc). For an isolated formula line, the line height and spacing is larger than those of ordinary text lines, and it is usually centrally aligned with a formula serial number at the end of line. There are many variations of constructing the quantitative models of layout features. In [8, 9], layout features are used to build decision trees based on predefined rules. These rule-based methods can only handle several specific types of documents. In [10, 11], Garain built quantitative models based on the statistics collected on a set of documents. Several crucial thresholds are set based on the statistics, which are very sensitive to the ratio of text lines and formula lines. In [12], Jin et al. exploited a machine learning technique (Parzen windows) to identify mathematical formulas. However, the features utilized in their method are limited, and thus it is not adaptive enough to deal with the varieties of formula layout. In [13], another learning-based method using computational geometry is presented. Its classifiers are trained to distinguish mathematics notations from English and may not be applicable in documents in other language, such as Chinese.

The third category of methods extracts mathematical expressions through image segmentation technique, without using the character or layout features [14]. Although this technique does not rely on the character recognition results, the segmentation thresholds required by this technique are hard to set, especially for documents of unknown types.

In summary, the existing formula extraction methods are mostly designed for image-based documents, and various types of features are not fully utilized and combined. Consequently, there exists no robust method to identify diverse types of formulas. Thus, we propose a method to address this need.

III. PROPOSED METHOD

Fig. 1 shows the workflow of the proposed method. The major steps include:

1. **Preprocessing**: Match the different types of objects (text, image, and graph) to the mathematical expression elements.

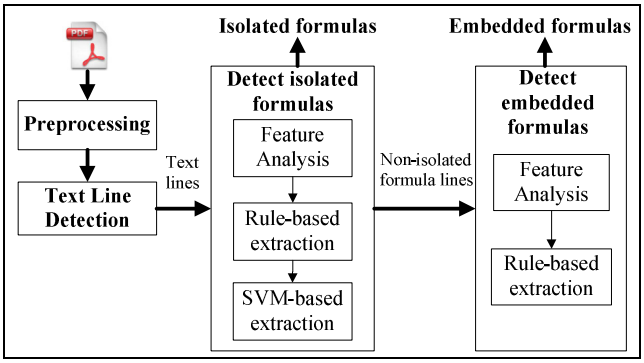

Figure 1. Workflow of the proposed formula extraction method

2. **Text line detection**: Text lines in the page are extracted to be used as the basic units in the following steps.

3. **Feature analysis**: Character and layout features representative of formulas are extracted.

4. **Formula area detection**: Rule-based method and Support Vector Machines (SVM) classifier are used and combined to identify the isolated formula areas. Rule-based method is used to detect the embedded formula areas.

A. *Preprocessing*

The goal of preprocessing is to match the original symbols parsed from a PDF document to the corresponding mathematical expression elements. There are three types of mathematical expression elements to be processed, including mixed math symbols, mathematical functions, and numbers. This step will output the descriptive details of the mathematical expression elements, such as locations, bounding boxes, baselines, fonts, which can be used as the character or layout features in the following steps. Preprocessing is critical because the better result it gets, the richer and more precise information of PDF document can be fully utilized later. Solutions are designed for different types of mathematical expression elements:

Mixed math symbols: Some math symbols are composed of different objects (graph, text, etc.) in the PDF document content stream. For example, one mathematical expression element may be made up of several graph objects and/or text objects. Through observing the parsing results of a large set of PDF documents, we find that mixed math symbols can be classified into three categories according to the composition of the mathematical expression elements: 1) Some elements are composed of one text object and one graph object. For example, a root symbol is composed of a graph object representing the horizontal line, and a text object which is a radical character. 2) Some elements are composed of several graph objects. For example, a vertical delimiter is made up of several vertical short line objects. 3) One single graph object represents a mathematical expression element. For example, fraction is represented as a horizontal line graph object. Accordingly, we have established a number of identification methods for each type of the mixture symbols.

For instance, to extract a root symbol from the PDF objects, we first locate all horizontal lines from the graph objects. Then, we search the radical character by searching

text objects whose Unicode is equal to " √ ". If so, the graphic object and the text object, which are adjacent to each other, are combined into a root symbol element.

PDF documents generated by different tools mainly differ in the composition of the mixed math symbols. In this paper, we take PDF documents of Version 1.3 generated by LaTeX as an example to describe our approach. The proposed method can work well on different versions of PDF documents through replacing the matching strategies.

Named mathematical functions: We detect the named mathematical functions by using a mathematical function dictionary created from the official LaTeX documentations. The sequence of characters representing named functions is extracted and grouped into a mathematical expression element tagged as a named mathematical function.

Numbers: Numbers can be detected among a string of characters by regular expression, and the characters representing a number are combined into a mathematical element tagged as a number.

B. Text Line Detection

After the preprocessing step, detailed information of both the candidate mathematical expression elements and the text are available (bounding box, baseline, font and font size, etc). Then text line detection is carried out. Reliable identification of text lines benefits the detection of other layout components, such as paragraphs and columns. For the purpose of formula extraction, text lines can serve as a basic unit of mathematical expressions. The isolated formula extraction can then be simplified into distinguishing formula lines from non-formula lines. In this paper we employ a branch-and-bound text line finding algorithm proposed in [15] to detect text lines .

C. Identification of Isolated Formulas

1) Feature Analysis

The features of isolated formulas can be classified into three categories: geometric layout features, character features and context features, whose definitions are listed in Table I.

The geometric layout features describe the text lines' layout in a whole page and they are the most important features for the isolated formulas. Generally speaking, the geometric layout features are extracted from the result of text line detection or character recognition, whose performance varies with different typesetting styles or document quality. Thus, it is a main bottleneck in processing image-based documents. Fortunately, this is not a significant problem for PDF documents, since the precise layout information of the text lines and the characters are already obtained in the preprocessing and text line detection steps.

Character features specify if certain mathematical symbols or named functions exist in text lines. Character features are simple but still effective. However, we have noticed that these features have not been fully utilized in previous work because a lot of math symbols cannot be correctly recognized by OCR systems. Fortunately, here we do not have to worry about this because almost all the math symbols can be correctly parsed from the PDF document in the preprocessing step discussed in Section III.A.

TABLE I. FEATURES OF THE FORMULAS

("I" denotes isolated formula, and "E" denotes embedded formula.)

Name	Definition	I	E
Geometric layout features			
AlignCenter	The relative distance of the line's horizontal center and the page body' horizontal center.	√	
V-Width.	The variation between two lines' widths.	√	
V-Height	A line's height.	√	
V-Space	The space between two successive lines.	√	
Sparse-Ratio	The ratio of the characters' area of the line's area.	√	
V-FontSize	The variance of the font size.	√	
SerialNumber	Whether there is a formula serial number in the end of the line.	√	
Italic	Whether the character is in italic.		√
Character features			
MathFunction	The named math functions (sin, cos, etc.), defined in the math function dictionary.	√	√
MathSymbol	Math symbols are categorized into: binary relations, binary operations, Greek letters, delimiters, functions, integral, fraction, square.	√	√
Context features			
Relationship	Whether the preceding/following character is a formula element.		√
Domain	Describe operand domains of particular math symbols such as the integral symbol.	√	√

Context features describe the relationship between characters, based on the math symbols' domain. Context features are used to merge or to expand the characters' areas into a formula's area.

2) Isolated Formula Detection

Since the isolated formulas have obvious layout features, the layout features are used as the dominant features to detect isolate formulas, and the character features are used as auxiliary features. By utilizing these features, we adopt both rule-based and learning-based methods to detect isolated mathematical expressions:

Rule-based method: First, we use the character features to filter out the lines which are very unlikely to be formula lines. It is necessary to implement this step, for there are some text lines (for example, title lines as noted in [11] as a different corner case) sharing layout features with the formula lines. Too strict rules may filter out true formula lines and cause low recall rate. So currently the filtering rules are very relaxed. A line is filtered out only when it does not satisfy any of the following two rules: 1) A named function appears in the line; 2) At least one math symbol appears in the line. After the filtering step, title lines will be filtered out because they usually contain neither math symbols nor functions.

Second, the geometric layout features of isolated expressions are exploited to calculate the confidence level of classifying a line as a formula line. Geometric layout features in Table I are utilized as binary features through comparing the features with some thresholds, which are set through statistical analysis. For example, if a line's *V-Height* is larger than the text lines' average height of the page, the line has this feature. We divide the importance of features into three levels, based on the statistics collected on a large number of PDF documents. Confidence scores are set according to the levels. For example, the three levels correspond to 5, 2, and 1

respectively. If a line has a feature, the corresponding score is added to that line's confidence score.

When the accumulated confidence level of a line is higher than a threshold (*vIF*), the line is recognized as an isolated formula. The value of the threshold is obtained by statistical analysis on the data set.

Machine learning-based method: To decide if a line is an isolated formula line is a classical binary classification problem. Like many pattern recognition problems, the key problem is to extract discriminating features. The geometric layout and character features listed in Table I are employed as a nine-element vector in our experiments. In our implementation, LIBSVM, an optimized implementation of Support Vector Machine (SVM) [16] is used to build the SVM classifiers. Radial Basis Function (RBF) is employed as the kernel function of SVM. The classifier is trained on the labeled data to predict whether a line is an isolated formula line.

Hybrid method: For each text line, the rule-based method is first executed to calculate its confidence level as formula line. If the confidence level is higher than *vIF*, it is recognized as an isolated formula, otherwise, a SVM classifier is employed to decide whether the line is an isolated formula.

D. Identification of Embedded Formula

The goal of this step is to detect the areas of the mathematical expressions in the text lines. The mathematical expressions in line include equations, variables and functions. There are very few layout differences between the embedded expressions and the ordinary text. Therefore, detecting embedded expressions relies mainly on character features combined with supplementary layout features. A rule-based method is adopted to detect embedded formulas.

1) Feature Analysis

Geometric layout features: In standard typesetting, mathematical symbols are italic or bold to distinguish from the ordinary text. This can be used as an important feature to identify embedded formulas. However, under the influence of the informal typesetting the font style information sometimes is difficult to extract without heuristics, especially in the image-based documents. Fortunately, for PDF documents, font styles can be obtained in the PDF document content stream. We exploit this distinctive feature to detect the embedded formulas.

Character features: As the layout features of the embedded mathematical expressions are limited, the most significant features of the embedded formulas are the character features. We divide the known math symbols into eight categories (defined in the "MathSymbol" row of Table I). Then these classes of symbols are used to look up the dictionary during the detection process.

Context features: The context features in Table I, reflect the relationships between characters and math symbol domains, and they are used to merge or to expand the areas of the math symbols in order to form the formula area.

2) Embedded Formula Detection

First, for each character or a sequence of characters in the non-isolated formula lines, layout and character features are used to calculate the likelihood of the character being a math symbol. We divide importance of each class of math symbols into different levels according to the uniqueness of each type of symbols. Similar to the isolated formulas, the confidence level is calculated according to the importance level. A character is recognized as a mathematical expression element when its accumulated confidence score is larger than a threshold (*vEF*).

Second, for those characters tagged as math symbols, the area of embedded expressions is obtained by merging areas of math characters using context features defined in Table I.

IV. EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed method on different types of formulas, we collect the data set from mathematics textbooks written in English[1] and Chinese. In total 421 pages are collected. Experiments are carried out on 200 randomly selected pages, which contain 5743 ordinary text lines, 1541 isolated formulas and 3237 embedded formulas.

For the hybrid method and learning-based method, the data set is divided into five equal parts and five-fold cross-validation is employed for training and testing. In our experiments, the thresholds used to determine the area as formula areas, *vIF* , *vEF*, are assigned values of 6 and 2, respectively. An example of our formula extraction result is shown in Fig. 2.

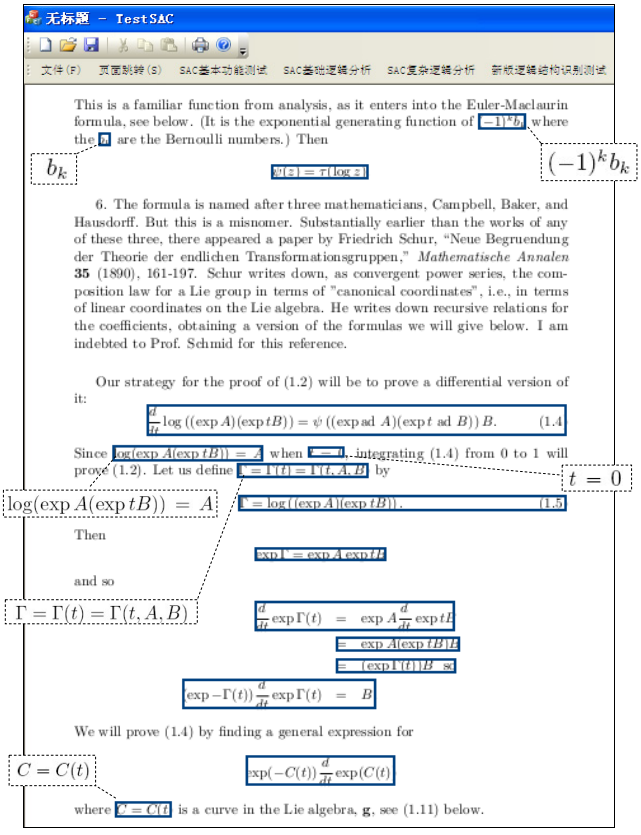

Figure 2. Example of the formula extraction

[1] http://www.math.harvard.edu/~shlomo/

A. Performance Evaluation

TABLE II. RESULTS OF THE ISOLATED FORMULA IDENTIFICATION

Method	Precision	Recall	F1
Rule-based	90.54%	90.66%	90.60%
Learning-based	94.33%	97.01%	95.64%
Rule-based + Learning-based	94.45%	97.91%	96.14%

TABLE III. RESULTS OF THE EMBEDDED FORMULA IDENTIFICATION

Method	Precision	Recall	F1
Rule-based	83.05%	84.18%	83.61%

We compare the performance of the hybrid method of isolated formula identification with the rule-based method and the learning-based method in Table II. Results of the embedded formula identification is presented in Table III. The evaluation metrics include three numbers: 1) **Precision** is the probability that the extracted bounding boxes match the formulas' areas; 2) **Recall** is the probability that the formulas are detected; 3) **F1** is the harmonic mean of precision and recall.

From the evaluation metrics, it is seen that the F1 of the hybrid method is higher than that of rule-based and learning-based methods by 5.54% and 0.50%, respectively.

The program is implemented in C++ and the tests are run on a 2.50GHz PC with 2GB RAM. On average, it takes 10 seconds to detect formulas from 200 pages. For the hybrid method and learning-based method, it takes less than 1 second to train the SVM classifier on a training set of 160 pages.

B. Analysis of the Experimental Results

The main cause of isolated formula identification errors is that some short text lines containing math symbols are recognized as isolated formulas. The precision and recall rates of the embedded formula extraction are lower than those of isolated formulas. There are a number of causes: 1) Some embedded formulas are only partially recognized for the deficiency of the merging and expanding rules. 2) Some math symbols (e.g., ⊙) in the text line cannot be recognized by the PDF parser, therefore there are no character features to distinguish them from ordinary text.

V. CONCLUSIONS

In this paper, a formula identification method targeting PDF documents is introduced. It involves several steps: preprocessing, isolated and embedded formulas extraction. And the experimental results show satisfactory performance of the proposed method. The contributions of this paper are as follows: 1) Problems and difficulties of detecting mathematical expressions in PDF documents are fully analyzed, and an automated solution is provided. 2) Various types of features, including character features, geometric layout features, and context features, are deeply explored. In addition, all of these features are combined with a carefully defined weight scheme according to different types of formulas. 3) The rule-based approach and learning-based approach are combined to complement each other to improve the performance.

In the future, we would apply our approach to process different PDF documents produced by various tools and improve the preprocessing procedure to adapt to different versions of PDF documents. Another interesting research direction is to automatically adapt the method's parameters through machine learning.

VI. REFERENCES

[1] W.S. Lovegrove and D. F. Brailsford, "Document analysis of PDF files: method, results and implications," Electronic publishing, Sep. 1995, 8: pp. 207-220.

[2] F. Rahman and H. Alam, "Conversion of PDF documents into HTML: a case study of document image analysis," Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers (ACSSC 03), Nov. 2003, pp. 87-91.

[3] H. Déjean and J.-L. Meunier, "A system for converting PDF documents into structured XML format," Proc. of Document Analysis Systems (DAS 06), Jan. 2006, pp. 129-140.

[4] J. Baker, A. P. Sexton and V. Sorge, "A linear grammar approach to mathematical formula recognition from PDF," Proc. Springer Symp. Intelligent Computer Mathematics (ICM 09), Jul. 2009, pp. 201-216.

[5] K. Inoue, R. Miyazaki and M. Suzuki, "Optical recognition of printed mathematical documents," Proc. of the third Asian Technology Asian Technology Conference in Mathematics, 1998, pp. 280-289.

[6] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida and T. Kanahori, "Infty: an integrated OCR system for mathematical documents," Proc. ACM Symp. Document Engineering 2003, Nov. 2003, pp. 95-104.

[7] A. Kacem, A. Belaid and M. Ben Ahmed, "Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context," IJDAR, vol. 4, no. 2, Dec. 2002, pp. 97-108.

[8] J.-Y. Toumit, S. Garcia-Salicetti and H. Emptoz, "A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical Documents," Proc. of International Conference on Document Analysis and Recognition (ICDAR 99), Sep. 1999, pp. 119-122.

[9] S. P. Chowdhury, S. Mandal, A. K. Das and B. Chanda, "Automated segmentation of math-zones from document images," Proc. of International Conference on Document Analysis and Recognition (ICDAR 03), Aug. 2003, pp.755-759.

[10] U. Garain and B. B. Chaudhuri, "A syntactic approach for processing mathematical expressions in printed documents," Proc. of the 15th International Conference on Pattern Recognition (ICPR 00), Sep.2000, pp. 523-526.

[11] U. Garain, "Identification of mathematical expressions in document images," Proc. of the tenth International Conference on Document Analysis and Recognition (ICDAR 09), Jul. 2009, pp.1340-1344.

[12] J. Jin, X. Han and Q. Wang, "Mathematical formulas extraction," Proc. of International Conference on Document Analysis and Recognition (ICDAR 03), Aug. 2003, pp. 1138-1141.

[13] D. M. Drake and H. S. Baird, "Distinguishing mathematics notation from English text using computational geometry," Proc. of International Conference on Document Analysis and Recognition (ICDAR 05), Aug. 2005, pp. 1270–1274.

[14] T.-Y. Chang, Y. Takiguchi and M. Okada, "Physical structure segmentation with projection profile for mathematic formulae and graphics in academic paper images," Proc. of International Conference on Document Analysis and Recognition (ICDAR 07), Sep. 2007, pp. 392-396.

[15] M. B. Thomas, "High performance document layout analysis," Proc. Symp. on Document Image Understanding Technology (SDIUT 03), Apr. 2003.

[16] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm.