Recognition of Printed Mathematical Expressions Using Two-dimensional Stochastic Context-Free Grammars

Francisco Álvaro, Joan-Andreu Sánchez, José-Miguel Benedí
Instituto Tecnológico de Informática
Universitat Politècnica de València
Valencia, Spain
{*falvaro,jandreu,jbenedi*}*@dsic.upv.es*

***Abstract*—In this work, a system for recognition of printed mathematical expressions has been developed. Hence, a statistical framework based on two-dimensional stochastic context-free grammars has been defined. This formal framework allows to jointly tackle the segmentation, symbol recognition and structural analysis of a mathematical expression by computing its most probable parsing. In order to test this approach a reproducible and comparable experiment has been carried out over a large publicly available (InftyCDB-1) database. Results are reported using a well-defined global dissimilitude measure. Experimental results show that this technique is able to properly recognize mathematical expressions, and that the structural information improves the symbol recognition step.**

***Keywords*-mathematical expression recognition; symbol recognition; structural analysis; context-free grammars; stochastic parsing;**

I. Introduction

Mathematical Expression (ME) recognition has been an active research field in the last years, due to the rapid growth of human interface devices and the great interest in transcribing scientific papers into electronic documents [1]. Most of the works that have studied the handwritten ME recognition problem have focused in the on-line approach [2], [3], [4] which uses temporal information about the stroke input. Off-line recognition deals with the image representation of ME, which can be printed or handwritten [5].

Printed ME recognition is an important problem for scientific document image analysis [1]. It has multiple applications like scientific document digitization, information retrieval or accessibility for blind people. Off-line recognition of printed ME can be divided into two major steps [1]: symbol recognition and structural analysis. Symbol recognition is responsible for image segmentation and properly detection of mathematical characters. Structural analysis aims to determine the relations among mathematical symbols in order to build a complete ME. Both problems are closely related and misrecognitions in the symbol recognition step usually cause errors in the analysis phase. Also, structural information can help to solve the symbol recognition step.

Several approaches have been studied to solve the ME recognition problem. Research of ME recognition based on trees [6] or graphs [7], [8] uses algorithms for these data structures. Otherwise, grammar-based approaches [9], [5] employ formal grammars and their corresponding parsing algorithms. Stochastic Context-Free Grammars (SCFG) are a powerful formalism that can tackle both problems in a natural way. SCFG for ME recognition has been studied in previous works [9], [5] but some issues are not sufficiently described. In this work we will focus on the off-line recognition of printed ME using SCFG.

Most of previous works on ME recognition are not easily comparable between them. There is a shortage of large, representative, publicly available, groundtruthed data sets [10]. Nowadays, the UW-III database [11] that is a small database with degraded images, and the InftyCDB-1 database [8] that is a large database with good-quality images, are good resources for printed ME recognition.

Automatic performance evaluation of ME recognition systems is an issue still in development [10]. Published works have reported several partial error metrics like symbol error rate or operator recognition rate [4]. Some global error measures were presented in [10], [12].

Our first contribution in this work is to provide a detailed description of an off-line printed ME recognition system based on SCFG. The second contribution of this work is to describe an experimentation on a publicly available dataset that makes easier the comparison with other systems.

The remainder of the paper is organized as follows. First, a background of the problem is given in Section II. Then, the formal statistical framework based on a two-dimensional extension of SCFG is explained in Section III. Section IV describes the ME recognition system developed in this work, and Section V presents the experiments performed using a publicly available database and a well-defined global performance evaluation metric.

II. Background

Off-line recognition of printed ME has been tackled using different techniques for each recognition step.

ME image segmentation is often performed by computing the connected components [5] or applying the projection-profile cutting method [13]. Segmentation problem is not an easy task, given that some mathematical symbols are

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.247

IEEE computer society

composed of multiple components $(i, j, =, :)$. In addition, image degradation causes regular symbols to be split into several components. Image degradation also adds noise and causes the appearance of touching characters which is a hard problem.

Recognition of mathematical symbols is carried out using pattern recognition techniques. Several classifiers has been used to deal with this task, and a comparison can be found in [14].

Structural analysis is usually the most different part of ME recognition systems. Two main approaches to solve this problem can be distinguished. First, several works have used trees or graphs in combination with heuristic functions. Zanibbi *et al.* [6] presented a system that built a baseline structure tree which was transformed through lexical and syntactical steps. Eto and Suzuki [7] defined a method that computed the minimum spanning tree over the network that linked the symbols of the ME.

Second, there are some papers that tackle the structural analysis using formal grammars. Grammar-based ME recognition started with the early work of Chou [9] that proposed to use SCFG in order to solve this task. Other proposals have been presented using definite clause grammars [4] or graph grammars [15]. Yamamoto *et al.* [2] presented a statistical formulation for on-line parsing of handwritten ME. They defined a two-dimensional extension of SCFG and how to compute spatial relations probabilities by using probability functions and the Cocke-Younger-Kasami (CYK) algorithm. Another interesting work was proposed by Průša and Hlaváč in [5]. It was penalty based and it used SCFG for off-line recognition solving the segmentation and symbol recognition jointly with the structural analysis phase.

In the present work, we adapted the statistical framework presented in [2] for on-line ME recognition, to the off-line printed ME recognition problem. Thereby, we developed an approach similar to [5] but defining a probability based framework that provides a proper scenario to develop a system where all the steps involved in ME recognition can be automatically learnt. The segmentation, symbol recognition and structural analysis can be tackled altogether. This framework allowed us to directly integrate statistical distributions into the model. Although the system is not fully developed, it has incorporated the main parts.

III. Two-dimensional context-free parsing

SCFG are a powerful formalism of syntactic pattern recognition that has been extensively used for string patterns. However, it is possible to slightly modify this formalism in order to model two-dimensional problems. In this work, we are interested in modeling ME using SCFG. Hence, a 2D extension of SCFG and its corresponding version of the CYK parsing algorithm are defined bellow based on [2], [5].

A. 2D SCFG

A SCFG can be defined as a tuple $G = (N, T, P, S, \Pr)$ where N is the set of nonterminal symbols, T is the set of terminal symbols, P is the set of derivation rules $A \rightarrow \alpha$ and S is the starting symbol of the grammar. Each production rule has attached a probability $\Pr(A \rightarrow \alpha) \in]0, 1]$, and $\sum_{\forall \alpha} \Pr(A \rightarrow \alpha) = 1$. This model can be represented in Chomsky Normal Form (CNF) and it results in two type of rules: terminal rules $(A \rightarrow t)$ and binary rules $(A \rightarrow BC)$.

The 2D extension of SCFG introduces mainly two differences. First, in the 2D case, terminal and nonterminal symbols describe two-dimensional regions. This means that terminal and nonterminal symbols of the grammar contain some features like 2D coordinates. Second, the production rules have an additional parameter that describes the spatial relation among regions. This relation is defined as $A \xrightarrow{spr} \alpha$, where spr denotes the spatial relation that models the rule. Common spatial relations for ME recognition are: *horizontal*, *vertical* (above or below relations), *inside*, *subscript* and *superscript*. Terminal productions do not contain the spatial relation because there is no relation with only one symbol.

B. CYK parsing for 2D SCFG

Once 2D SCFG are defined, we are able to model the ME structure and to parse an input sample to obtain the most probable derivation. We perform this task by using the CYK algorithm, but it must be slightly modified to work with 2D SCFG.

First, we define the way to compute the probabilities of the derivations in a similar way as in [2]. In this paper we suppose equal the prior probability of all expression hypotheses. The probability of a terminal rule $\Pr(A \rightarrow t)$ is obtained from a mathematical symbol classifier as the probability that region t belongs to class c such that $(A \rightarrow c)$. The probability of a binary rule $\Pr(A \xrightarrow{spr} BC)$ models the spatial relation spr between B and C regions. For that reason, it must be defined a function or distribution that represents the probability that regions B and C were arranged according to spr.

Figure 1 shows the CYK parsing algorithm for 2D SCFG. Given a 2D SCFG and a sample input x composed of n symbols, the most probable derivation is computed. The $\oplus$ operator computes the smallest rectangle containing both regions. The $\uplus$ operator adds an element to the set if it doesn't appear in the set or if the element is already present, the probability is maximized.

Looking at the 2D CYK algorithm, a remarkable difference is that the parsing table is indexed by only one value. On the standard CYK parsing two indexes explain the positions that define some substring. In the 2D case, there is a table level for each subproblem size, and these levels store a set of elements which contain their two-dimensional space information. For that reason, at the initialization loop the built subproblems are added at $t[1]$ level, that is

Input: 2D SCFG $G_s = (N, T, P, S, \Pr, spr)$ in CNF and $x = \{x_1, x_2, \ldots, x_n\} \in T^*$
Output: $\widehat{\Pr}_{G_s}(x)$: probability of most probable derivation

for all $i = 1 \ldots n$ **do**
  for all $(A \to x_i) \in P$ **do**
    if $\Pr(A \to x_i) > 0.0$ **then**
      $t[1] := t[1] \cup (A, x_i, \Pr(A \to x_i))$

for all $j = 2 \ldots n$ **do**
  for all $a = 1 \ldots j-1$ **do**
    for all $c_1 = (B, r_1, p_B) \in t[a]$ **do**
      for all $c_2 = (C, r_2, p_C) \in t[j-a]$ **do**
        for all $(A \xrightarrow{spr} BC) \in P$ **do**
          $prob := p_B \cdot p_C \cdot \Pr(A \xrightarrow{spr} BC)$
          if $prob > 0.0$ **then**
            $t[j] := t[j] \uplus (A, r_1 \oplus r_2, prob)$

return $(S, x, p) \in t[n]$

Figure 1. CYK parsing algorithm for 2D SCFG. (A, r, p) represents that nonterminal symbol A accounts for region r with probability p.

to say that they cover one input symbol. After that, the parsing process continues by building new subproblems of increasing size, where the spatial relation model contributes to the probability of each possibility.

Finally, the time complexity of the algorithm is $O(n^4|P|)$ whereas the time complexity of the classical CYK is $O(n^3|P|)$. However, in the following section this complexity will be discussed and reduced.

IV. Mathematical Expression Recognition

We have developed a system based on 2D SCFG for recognition of printed ME[1]. In this work, we manually defined the grammar in order to account for all ME that appeared in the InftyCDB-1 database. As a result, a wide range of expressions were modeled, except left subscripts and superscripts $({}^{2}_{1}a)$ or matrices. After parsing an expression, the system output the LaTeX representation of the recognized ME. In the following, the steps involved in a ME recognition system are detailed.

A. Symbol Recognition

Given an image of a ME, the first step is to segment this image into symbols. In this work, the method chosen to solve the segmentation problem was to compute the connected components of the input image. For all of these regions, a mathematical symbol classifier was used to determine the class of each one. In our case, the Nearest Neighbor (NN) classifier was chosen with the Euclidean distance and each bounding box was normalized to a fixed size [14]. A mathematical symbol can belong to multiple classes due to misclassification or different interpretations. For that reason, the symbol recognition process classified each terminal in several nonterminals that represented its possible interpretations. Finally, the parsing process decided the most probable interpretation taking into account the ME structure.

The NN classifier computed the Euclidean distance between vectors, but in the CYK algorithm probabilities were needed. Formally, given an image x, let $\hat{p}_c$ be the nearest prototype of class c from a labeled set, and let $d(x, \hat{p}_c)$ be the distance between them. The probability of x to belong to the class c was obtained as

$$p(x \mid c) \propto e^{-d^2(x, \hat{p}_c)}$$

The CYK table was initialized with the classification results obtained for each symbol. There are two main segmentation problems. First, the system currently is not able to deal with touching characters. Second, an important problem was the multiple connected components symbol detection. In this work, we merged close components and then the mathematical symbol classifier was used to obtain the probability of being a certain symbol. These hypotheses were also added to the CYK parsing table, and the structural analysis decided the most probable derivation.

B. Structural analysis

We implemented the algorithm of Figure 1 to perform the ME recognition, but some details of the system need additional comments. The input of the developed system was an image of a ME of n connected components, as a result of the segmentation step. Thus, segmentation and symbol recognition hypotheses were used to initialize the CYK parsing table.

Once the CYK parsing table was initialized, the algorithm went on building new hypotheses of increasing size. The probability of a new problem derived from other two subproblems c_1, c_2 of minor size was computed as

$$p_B \cdot p_C \cdot \Pr(A \xrightarrow{spr} BC)$$

where p_B and p_C probabilities were obtained from the CYK table, but the spatial relation probability $\Pr(A \xrightarrow{spr} BC)$ had to be defined. In this work we manually defined probability functions for each type of spatial relation based on geometric features [2]. Thus, the function ideally would provide a high probability value given two regions and a certain spatial relation, if they were arranged according to that relation. Likewise, a lower probability value was expected for unlikely regions relative positions. Examples of regions features are the vertical centroid, point scale, or the horizontal center.

There was a scaling issue with the probabilities of mathematical symbols of multiple connected components given

[1]The software is available at http://users.dsic.upv.es/~falvaro

that the basic unit of the CYK was a connected component, not a symbol. Classifier probability of single component symbol were inserted into the $t[1]$ level of the CYK table. However, symbols composed of two connected components had to be inserted into the $t[2]$ level, which probability should be the product of three different values. Therefore, these probabilities were scaled when they were added to higher levels of the CYK table, in order to not favor these type of constructions. We also limited the number of multiple connected components detection to 2, so currently the system could not detect properly symbols split in more than two components.

The CYK algorithm presented in Section III had time complexity $O(n^4|P|)$, but it was reduced as follows. It should be noted that each region at a certain level of the CYK table is checked to be merged with all the hypotheses of another level. However, given the nature of ME, we know which regions of the space contained likely elements. Hence, we limited the hypotheses search space in a similar way as in [5]. After completing one level of the CYK table, the hypotheses were sorted according to their horizontal coordinate, given that the ME grew in that direction. Then, when elements contained in a specific region $\mathcal{R}$ of the space were required, they could be easily obtained in $O(\log n)$ over the sorted set. Consequently, the loop $c_2 \in t[j-a]$ was changed to $c_2 \in \mathcal{R}$. Performing the partial sort using a $O(n \log n)$ algorithm, the improved time complexity was $O(|P|n^3 \log n)$.

Finally, when the parsing process finished, the most probable hypothesis of size n covered by the initial symbol of the grammar was retrieved from $t[n]$. If the expression was not fully recognized, the system looks for the most probable hypothesis of minor size $(t[n-1], t[n-2], \ldots)$ until it finds a valid ME.

V. Experiments

We developed a ME recognition system based on 2D SCFG, and we carried out some experiments to validate this approach. The publicly available InftyCDB-1 database [8] was used to perform the experiments. It has 21K ME, which in turn contain 157K mathematical symbols belonging to 212 classes, and each symbol and expression is annotated with many useful information. We discarded those ME that only contained one symbol because they didn't have any structural information (25% of 21K ME). From the remaining ME, we discarded those ME that contained touching characters, or symbols composed by more than 2 connected components ($\equiv$,$\leqq$) because they were not modeled by the defined SCFG (10.73% of 21K ME).

Finally, 13K ME were left for experiments, which represented around 85% of the expressions with more than one symbol. From this data, 3K expressions were selected as a test set, and the train set was composed by the 10K remaining expressions. This partition was done randomly.

Table I
Experiment results for the InftyCDB-1 database.

#Symbols	$2-7$	$8-14$	$15-21$	≥ 22	Total
%	60.19	21.37	7.6	10.84	100
EMERS	0.8±1.5	2.6±3.1	4.1±4.2	8.4±9.3	**2.25±3.8**
SER(NN)	5.5	5.7	5.6	7.7	5.76
SER(ME)	4.5	4.9	4.2	5.6	**4.68**

The total number of mathematical symbol classes for the experiment was 183.

Before recognizing the ME, the samples were preprocessed using image filters in order to remove noise. As explained in Section IV, the NN classifier was used to perform the symbol recognition. For that reason, we extracted and normalized to a fixed size the mathematical symbols of each ME from the training set, and these were used as prototypes of the NN classifier [14].

Automatic ME performance evaluation is not an easy task due to representation ambiguity of the groundtruthed data [10] (usually as LaTeX or MathML). In this work we used EMERS [12] as a global performance metric. MathML format directly represents a ME including its tree structure, thus, this measure computes the edit distance between trees of the recognized expression and the groundtruth data. It should be noted that this measure doesn't provide an error value, but a comparable dissimilitude value instead, so a zero-distance result means that the ME is perfectly recognized.

The test set results were divided into several intervals according to the ME number of symbols. Experiment results are shown in Table I. First row specifies the number of symbols of the ME samples whose results are reported in each column. Second row (%) shows the percentage of samples that comprised each interval over the whole test set. Third row (EMERS) shows the mean and standard deviation of the EMERS edit distance. Fourth row shows the Symbol Error Rate (SER) of the NN mathematical symbol classifier. Last row shows the SER obtained after parsing the ME, taking into account structural information of the ME.

The experiment results showed that the average EMERS metric over the test set was 2.25. It can be appreciated that the EMERS distance increased as the size of the ME was greater. The percentage of ME that had EMERS distance equal to zero were 56.72%, however this metric is pessimistic. We realized that the automatic ME evaluation remains an open problem [10] because EMERS measure suffers the ME representation ambiguity problem. Figure 2(d) shows an example of this situation, where the same ME can be built from different expression trees.

Regarding symbol error rate, it can be seen that structural information improved symbol recognition. Furthermore, results showed that this improvement was greater with larger ME, due to they contained more structural information than

shorter ME. Final symbol error rate over the test set was 4.68%.

Figure 2 shows examples of the most common errors. Similar symbol misrecognition is usual, specially with symbols as $\{(1,l),(o,0)\}$ (a) or small characters like $\{,.\}$ that are recognized as superscripts or subscripts (b). Spatial relations misdetection (c) is another source of errors.

VI. Conclusion and future work

In this work we presented a printed ME recognition system based on a two-dimensional extension of SCFG. We defined a statistical framework to tackle this problem by using stochastic parsing methods. We performed an experiment over a publicly available and large database, and the results were presented using a well-defined ME global automatic performance measure. Results showed that structural information of ME improved mathematical symbol recognition.

Future work will be focused in automatic learning of spatial relation distributions, for instance, Gaussian mixtures could be directly incorporated to this model by providing the $\Pr(A \xrightarrow{spr} BC)$ probability. In this way, all the issues involved in ME recognition process will be learnt from the training set. It also could be improved the ME used grammar and the segmentation of touching symbols and noise components.

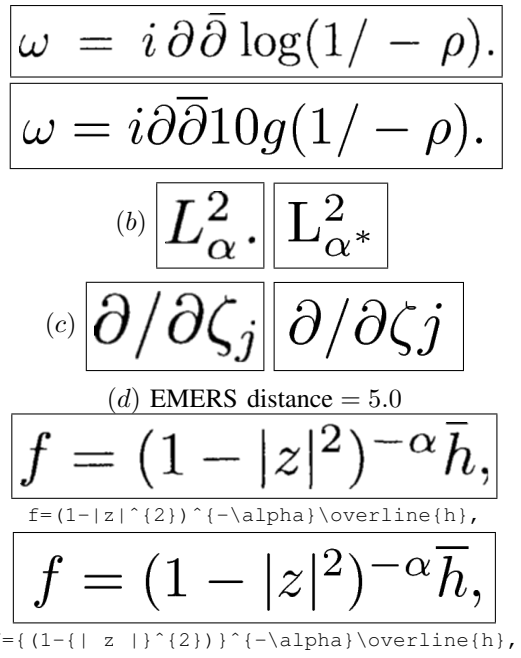

Figure 2. Examples of common errors in ME recognition. For each case, the input image and the recognized expression are displayed.

Acknowledgment

Work supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), the MITTRAL (TIN2009-14633-C03-01) project, the FPU (AP2009-4363) grant, and by the Generalitat Valenciana under grant PROMETEO/2009/014.

References

[1] K. Chan and D. Yeung, "Mathematical expression recognition: a survey," *International Journal on Document Analysis and Recognition*, vol. 3, pp. 3–15, 2000.

[2] R. Yamamoto, S. Sako, T. Nishimoto, and S. Sagayama, "On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar," *IEIC Technical Report*, 2006.

[3] U. Garain and B. Chaudhuri, "Recognition of online handwritten mathematical expressions," *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 34, no. 6, pp. 2366–2376, 2004.

[4] K.-F. Chan and D.-Y. Yeung, "Error detection, error correction and performance evaluation in on-line mathematical expression recognition," *Pattern Recognition*, vol. 34, no. 8, pp. 1671 – 1684, 2001.

[5] D. Průša and V. Hlaváč, "Mathematical formulae recognition using 2d grammars," *International Conference on Document Analysis and Recognition*, vol. 2, pp. 849–853, 2007.

[6] R. Zanibbi, D. Blostein, and J. Cordy, "Recognizing mathematical expressions using tree transformation," *Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1–13, 2002.

[7] Y. Eto and M. Suzuki, "Mathematical formula recognition using virtual link network," in *International Conference on Document Analysis and Recognition*, Washington, DC, USA, 2001, pp. 762–767.

[8] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "Infty- an integrated OCR system for mathematical documents," in *Proc. of ACM Symposium on Document Engineering*, Grenoble, 2003, pp. 95–104.

[9] P. A. Chou, "Recognition of equations using a two-dimensional stochastic context-free grammar," in *Visual Communications and Image Processing IV*, W. A. Pearlman, Ed., vol. 1199, 1989, pp. 852–863.

[10] A. Lapointe and D. Blostein, "Issues in performance evaluation: A case study of math recognition," in *International Conference on Document Analysis and Recognition*, Washington DC, USA, 2009, pp. 1355–1359.

[11] I. Phillips, "Methodologies for using UW databases for OCR and image understanding systems," in *Proc. SPIE, Document Recognition V*, vol. 3305, 1998, pp. 112–127.

[12] K. Sain, A. Dasgupta, and U. Garain, "EMERS: a tree matching–based performance evaluation of mathematical expression recognition systems," *International Journal on Document Analysis and Recognition*, pp. 1–11, 2010.

[13] X.-D. Tian, H.-Y. Li, X.-F. Li, and L.-P. Zhang, "Research on symbol recognition for mathematical expressions," in *International Conference on Innovative Computing, Information and Control*, Washington, DC, USA, 2006, pp. 357–360.

[14] F. Álvaro and J. A. Sánchez, "Comparing several techniques for offline recognition of printed mathematical symbols," *International Conference on Pattern Recognition*, vol. 0, pp. 1953–1956, 2010.

[15] S. Lavirotte and L. Pottier, "Mathematical formula recognition using graph grammar," in *In Proc. of the SPIE*, vol. 3305, no. 1, 1998, pp. 44–52.