Pattern Recognition Letters 35 (2014) 68–77

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A global learning approach for an online handwritten mathematical expression recognition system

Ahmad-Montaser Awal [a,*], Harold Mouchère [b], Christian Viard-Gaudin [b]

[a] Laboratoire d'étude des Mécanismes Cognitifs, Université Lumière Lyon2, Lyon, France
[b] LUNAM Université, Université de Nantes, IRCCyN/IVC, Nantes, France

ARTICLE INFO

Article history:
Available online 16 November 2012

Keywords:
Handwriting recognition
Bidimensional languages
Math recognition
Structural pattern recognition
Syntactic pattern recognition

ABSTRACT

Despite the recent advances in handwriting recognition, handwritten two-dimensional (2D) languages are still a challenge. Electrical schemas, chemical equations and mathematical expressions (MEs) are examples of such 2D languages. In this case, the recognition problem is particularly difficult due to the two dimensional layout of the language. This paper presents an online handwritten mathematical expression recognition system that handles mathematical expression recognition as a simultaneous optimization of expression segmentation, symbol recognition, and 2D structure recognition under the restriction of a mathematical expression grammar. The originality of the approach is a global strategy allowing learning mathematical symbols and spatial relations directly from complete expressions. A new contextual modeling is proposed for combining syntactic and structural information. Those models are used to find the most likely combination of segmentation/recognition hypotheses proposed by a 2D segmentation scheme. Thus, models are based on structural information concerning the symbol layout. The system is tested with a new public database of mathematical expressions which was used in the CHROME competition. We have also produced a large base of semi-synthetic expressions which are used to train and test the global learning approach. We obtain very promising results on both synthetic and real expressions databases, as well as in the recent CHROME competition.

1. Introduction

Since the emergence of new technologies such as digital pens, tablets, smartphones, etc., digital documents are used increasingly. Nevertheless, scientific documents are full of diagrams and equations. Those notations are indispensable for describing problems and theories using common universal languages. The main property of these languages is their two dimensional nature, where symbols are organized in a two-dimensional (2D) space. Among others, mathematical notation is a very important 2D language because it is used in almost all sciences.

In order to take advantage of pen-based technologies, it is necessary to build systems allowing the transfer of the handwritten traces (physical form) to a digital text (logical form). The stochastic nature of handwriting and the variability in writing styles make achieving this task quite difficult. Handwriting recognition has been an active domain of research since the 60s of the last century. Text recognition systems have known an important development in the last few years (Plamondon and Srihari, 2000).

However, these systems are limited to the recognition of characters organized in a sequence of words belonging to a given vocabulary. The spatial arrangement of these constituents is the horizontal line (or vertical in some Asian languages). This is not the case when the grammar controlling the language is itself of a 2D nature. This is particularly the case for MEs (Chan and Yeung, 2000a,b), schemas (Feng et al., 2009), diagrams (Yuan et al., 2008), tables (Coüasnon, 2001), chemical equations (Wang et al., 2009), musical scores (Szwoch, 2007), characters of some languages such as Chinese (Delaye et al., 2009), etc.

Many tools support the input of MEs into digital documents, but special skills are required to use them properly. For example, LaTeX and MathML require the knowledge of predefined keywords. Other tools, such as MathType or Microsoft equations, depend on a visual environment for adding mathematical symbols using the mouse. Those dependencies on the mouse or the keyboard increase significantly the time required to input an expression vs. drawing an expression.

In this paper, we will focus on the recognition of online handwritten MEs. In the next section we will present some related works. Section 3 describes how MEs are represented. The proposed ME recognition system (MEXREC) is then presented; we will particularly focus on our contribution with the global learning scheme, and the spatial relation modeling. Unlike existing systems,

* Corresponding author.
E-mail addresses: ahmad-montaser.awal@univ-lyon2.fr (A.-M. Awal), harold.mouchere@univ-nantes.fr (H. Mouchère), christian.viard-gaudin@univ-nantes.fr (C. Viard-Gaudin).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.10.024

the proposed system is fully configured and trained directly from complete mathematical expressions, with no constraints related to stroke time order. Finally, we present some results in Section 6.

2. Related works

Math recognition has been an active research area since the late 60s, first works were dedicated to the offline recognition of typed expressions (Anderson, 1968; Chang, 1970) or handwritten ones (Belaid and Haton, 1984). The process of 2D language recognition is mainly based on three sequential steps: segmentation, symbol classification and interpretation (structural and syntactic analysis). However, global approaches have been introduced quite recently to solve this problem by applying jointly these three steps.

Being able to segment the 2D ink-traces into its basic symbols is a very important stake. In an online signal we can impose the condition of lifting the stylus when moving from one symbol to the next one. This condition is readily acceptable only in some languages, like MEs (Rhee and Kim, 2009), chemical equations (Wang et al., 2009) and flowcharts (Yuan et al., 2008). However, this assumption does not solve the segmentation problem, which consists in grouping strokes belonging to the same symbol.

A ***stroke*** is the sequence of points between a pen-down and a pen-up of a stylus. The stroke is our basic unit and we assume that a stroke belongs only to one symbol. However, a symbol can be written with one or more strokes, which are not necessarily sequential. Existing works do not support this latter hypothesis, since interspersed symbols increase considerably the complexity of the segmentation process. Most of existing works consider that all strokes are present before starting the segmentation and recognition, taking into account the global context. Other approaches consider processing the strokes on the fly as they are input and though provide updated results for each new stroke, allowing a direct feedback to the user.

The number of possible segmentations for a set of strokes is defined by the Bell number (Eq. 1). For example, considering the presence of a set of seven strokes, we obtain $B_7 = 877$ distinct segmentations.

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k; \text{ with } B_0 = B_1 = 1 \tag{1}$$

Not all of these segmentations have to be considered, as time order and spatial information can be used to prune the segmentation space. A bottom-up analysis is applied in Dimitriadis et al., 1991 using horizontal/vertical projections. Similarly, bounding boxes can be used to group strokes (Chan and Yeung, 2001). Other spatial constraints have also been used; a new stroke is judged belonging (or not) to the same symbol of the previous one based on a distance measurement (Tapia and Rojas, 2003). Lehmberg et al. (1996) used the strokes' properties in addition to spatial information. The probability decides if a stroke must be grouped with one or more (up to three) of the following strokes. This method is very sensitive to stroke temporal order.

This last approach evolved to what is called "simultaneous segmentation and recognition", where the main criterion to group strokes is the probability of those strokes representing a given symbol. We will briefly explore possible classification approaches usually used in 2D language recognition.

The goal of the classification step is to identify the objects found during the segmentation process. Symbol classification is quite complicated in the case of mathematical symbols because of the large number of classes and their overlapping. In fact, when looking at different corpus of mathematical expressions (Raman, 1994; Garain and Chaudhuri, 2004), we can notice that more than 220 symbol classes are encountered. This number becomes much bigger if the fonts have to be detected. A large variety of classifiers have been used in 2D languages recognition systems: template matching (Rhee and Kim, 2009), structural approaches (Chan and Yeung, 2000a,b), K nearest neighbor (Prusa and Hlavac, 2007), support vector machines (SVM) (Keshari and Watt, 2007), fuzzy logic based classifiers (Macé and Anquetil, 2009) and neural networks (Dimitriadis and Coronado, 1995).

All those classifiers require a prior segmentation, i.e., having the symbols extracted first. Yet, some techniques such as hidden Markov models (HMM) (Kosmala et al., 1999) can perform segmentation and recognition, simultaneously. Another approach consists in proposing all possible segmentations and associating a cost with each candidate segmentation. Finally, the segmentation with the minimal cost is chosen (Fukuda et al., 1999). This last approach becomes more feasible thanks to dynamic programming techniques that allow an efficient exploration of the search space.

After identifying the symbols, the next stage of recognition consists in finding the physical and logical structures of the expression.

Structural analysis consists in finding the spatial relations between the symbols based on symbol structural information. Depending only on bounding boxes as a source of this information (Ha et al., 1995) might cause some ambiguities, but structural information could be based on symbol typographical centers (Zanibbi and Blostein, 2002), or even on bounding boxes with baselines adapted to the type of the symbol (Eto and Suzuki, 2001; Mitra et al., 2003).

When all the symbols are correctly recognized, it is still necessary to analyze the expression's 2D structure to produce the final output, for example to differentiate "2x" from "2^x". The intuitive way of defining a spatial relation is defining regions around each symbol (Lavirotte and Pottier, 1998; Zanibbi and Blostein, 2002). However, spatial relations are of a fuzzy nature. Thus, the use of fuzzy rules is very appropriate for such analysis (Zhang et al., 2005; Fitzgerald et al., 2006, 2007). Aly et al. proposed a method based on the distribution of certain relation features (Aly et al., 2009) using a training database. A normalized distribution map is constructed using relational spatial information (Eto and Suzuki, 2001). Each relation is then associated to a probability obtained from the estimated distribution. We will propose a similar statistical approach to learn spatial relation based on Gaussian distributions.

Syntactic analysis is usually the last step in expression recognition. The objective of this step can be summarized in the following three points:

- Assure the grammatical correctness of the recognized expression.
- Produce the derivation tree of the expression which is then easily transformable to a standard presentation format such as LaTeX or MathML.
- And, more importantly, use the global context in order to resolve the local ambiguities.

A context free grammar (CFG) can be perfectly used to produce a subset of mathematical expressions. CFGs have been efficiently applied to analyze 1D formal languages such as programming languages. However, analyzing 2D languages requires special algorithms and constraints in order to reduce the complexity (Miller and Viola, 1998). Two principal approaches have been investigated in the literature: grammar or graph based analysis.

Since 1D parsers are more efficient than bi-dimensional ones, (Tokuyasuy and Chou, 1999) proposed to apply iteratively syntactic analysis along horizontal and vertical axes respectively using a stochastic context free grammar. (Chan and Yeung, 2000a,b) proposed to transform the expression from its 2D form to a unidimensional one, and to analyze it using a "define clause grammar" (DCG). Based on structural information extracted from the expression, grammatical analysis could be applied using CFG (Garain and Chaudhuri, 2004). Probabilistic grammar is proposed in (Yamamoto et al., 2006), where each rule of the grammar is associated to a structural constraint and probability reflecting the certainty of its logical relation. Similarly, a fuzzy online structural analysis algorithm is proposed in (Fitzgerald et al., 2006) in order to cope with the nature of spatial relations. More recently, (Scott and George, 2010) proposed to associate each production rule of a context free grammar to a fuzzy function. We will propose a similar approach by associating each production rule to a Gaussian model specific to each spatial relation as we will see in Section 5.2.

Graph rewriting can also be used for syntactic analysis. The structure of the expression is represented by a graph. Rewriting consists in replacing a sub-graph by a single node containing the syntax of the sub-expression (Grbavec and Blostein, 1995; Kosmala et al., 1999). However, the calculation time of graph rewriting is long.

Classically, previous steps are applied sequentially. As a result, an error occurring in one step would be inherited by subsequent steps. Furthermore, some local ambiguities require the whole context to be resolved. This led to an approach based on a simultaneous application of all the steps, or what is called a "global approach" introduced by Lecun et al. (1998) which is more and more used recently. Indeed, the main steps are still the same but performed jointly in a global framework to recognize at a time the entire expression.

Yamamoto et al. (2006) proposed to model the whole recognition process by a stochastic context free grammar. The grammar takes into account the writing order and the 2D nature of symbols. The first type of production rules controls the production of symbols from the input strokes. Each of those rules is probabilistic, i.e., when a rule is applied a probability is directly calculated from the symbol classification module. The rest of production rules models the spatial relations and the syntax by calculating a structural probability using the bounding boxes. The CYK algorithm (Cock–Younger–Kasami) is used to find the most probable expression. The main disadvantage of this algorithm is its dependency with respect to the temporal order of strokes. As a result, the user must input strokes in a correct order or pre-processing methods must be applied.

In order to overcome the temporal order problem, Rhee et al. proposed a layer search framework (Rhee and Kim, 2009). The recognition problem is reformulated as a search of the most probable interpretation of a given stroke set. The expression structure is extended by adding symbol hypotheses, representing the different identity of symbols, and at each structural ambiguity a new branch is added, creating a search tree. The search of the most likely solution is carried out by the "best first search" algorithm.

Other examples of global approaches can be found in (Prusa and Hlavac, 2007; Shi et al., 2007). Regardless of the algorithm, a global based approach is generally formulated as an optimization of a global cost function.

Precisely, our system is based on a global approach methodology by simultaneously applying the main recognition steps (symbol segmentation/recognition and structure/syntactic analysis). The proposed architecture tries to break up the limits of existing methods by taking into consideration the following points. First of all, it will allow handling several segmentations, several recognitions, and several logical relationships to select the best possible interpretation of the input strokes. Thus, this will avoid any local ambiguity and take advantage of the global context. Secondly, differently from most proposed methods, strokes will be treated not only temporarily but also spatially. This will easily allow dealing with delayed strokes, such the "i" point, or even transforming a '=' symbol in a '≠' after the whole expression is written. Another distinctive feature of the system is its capacity to train the symbol classifier and the spatial relational model directly in the global context of the entire expressions, and that is what we will call "global learning". The classifier is referred to as a "global classifier" which has a reject capacity allowing it to discard wrong segmentations.

3. Representation of mathematical expressions

Before introducing the MEXREC system in Section 4, it is important to understand how a ME is usually represented. Two main families of trees are usually encountered to describe the same structural information. The symbol relational tree (SRT) presents the spatial (or logical) relations between symbols (or sub expressions) of the expression (Geneo et al., 2006; Rhee and Kim, 2009). On the other hand a baseline structure tree (BST) captures the structure of an expression by representing the relation between the symbols' baselines (Tapia and Rojas, 2005; Zanibbi and Blostein, 2002). Operator trees are another kind of description for MEs, but they are more concerned by the semantic of the ME than its layout.

We propose a variation of the SRT as follows. A relational tree is constructed by a 2D context free grammar allowing verifying the syntactic correctness of the expression.

A non-terminal node (NT) contains a sub-expression (SE) (the root contains the proposed solution), it is described by a set of strokes and the corresponding label string, produced by the combination of sub-expressions linked by a spatial relation R. The cost of a non-terminal node (including the root), called structural cost, $C_{struct}(R|SE)$ is the cost that two or more sub-expressions (SE_i) are in relationship by R to build a bigger sub-expression (SE).

Each terminal node (T) contains a set of strokes produced by the hypothesis generator. For each of these nodes, the symbol classifier produces a ranked list of labels with their recognition scores $C_{reco}(sh_i)$; where sh_i is a segmentation hypothesis (see Section 5.1). The rules that produce terminals are not associated with spatial relations. Conversely, a spatial relation is associated with each rule generating non-terminal nodes. As a simple example, the following context free grammar generates candidates similar to those of Fig. 1:

$sym \leftarrow x, y, 1, 2, \ldots$
$op \leftarrow +, -, \times, \ldots$
$formule \leftarrow subExp\ op\ sym \quad (operator)[HorizontalRule]$
$subExp \leftarrow subExp\ op\ sym \quad (operator)[HorizontalRule]$
$subExp \leftarrow symsym \quad (superscript)[VerticalRule]$
$subExp \leftarrow sym$

Finally, the global cost (C_E) of a candidate expression of n symbols connected by r relations has been defined as a weighted sum of the symbol recognition and structural scores associated to each node (ex: Eq. 3):

$$C_E = \sum_{i=1}^{n} C_{reco}(sh_i) + \alpha. \sum_{j=1}^{r} C_{struct}(R_j|SE_j) \tag{2}$$

The α factor is used to adapt the respective ranges of the structural and recognition costs. Alpha has been set experimentally to 0.18 using a validation dataset.

Fig. 1 shows an example of such a relational tree. In the tree of the left candidate expression, the non-terminal node '2' connects the terminals '5' and '6', that contain symbol hypotheses x and 2 respectively, by the relation R_1 ("superscript") producing the sub-expression x^2. The other non-terminal, node '1', is the root of the tree. It connects the nodes '2', '3' and '4' with the relation R_2 ("operator") producing the final result x^2-1. The global cost of this solution is:

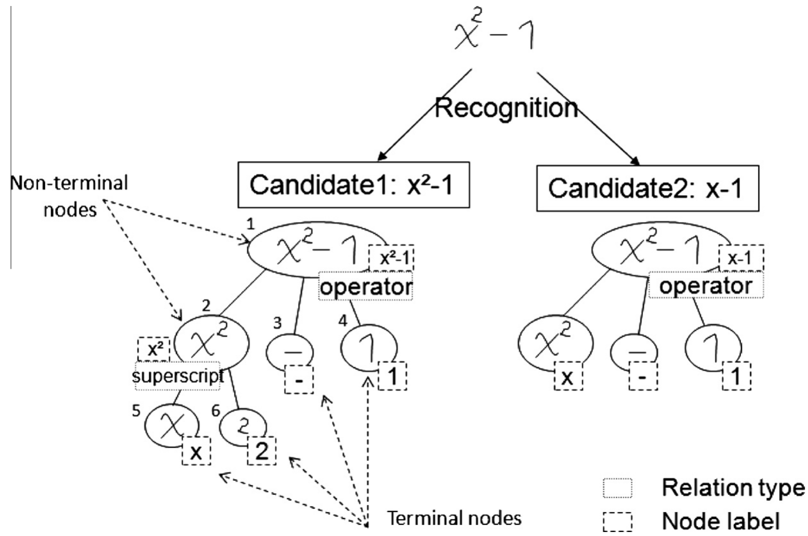

Fig. 1. Relational trees for two expression candidates.

$$C_E = C_{reco}(\text{‘}x\text{’}) + C_{reco}(\text{‘}2\text{’}) + C_{reco}(\text{‘}-\text{’}) + C_{reco}(\text{‘}1\text{’}) + \alpha.C_{struct}(R_1|x^2) + \alpha C_{struct}(R_2|x^2 - 1) \quad (3)$$

Fig. 1 illustrates a reduced parse tree with two solutions. Several relational trees will be built as explained in the next section by the recognition system; the one with the lowest cost is the proposed solution.

4. MEXREC: a mathematical expression recognition system

Considering a given expression as a set of strokes $E = \{s_1, s_2, \ldots, s_n\}$ representing symbols, recognizing an expression consists in finding the best possible grouping of strokes, identifying the symbol corresponding to each group, and finally interpreting the expression according to the language model. Those different steps participate in calculating the global cost function C_E. The general architecture of the MEXREC system is presented in Fig. 2.

4.1. Symbol hypothesis generator

The generator elaborates stroke combinations called a symbol hypothesis $sh \subseteq E$. However, many hypotheses are invalid due to under or over-segmentation. In the latter case, only a sub-part of a multi-stroked symbol is chosen; while in the first case strokes are grouped from two or more symbols.

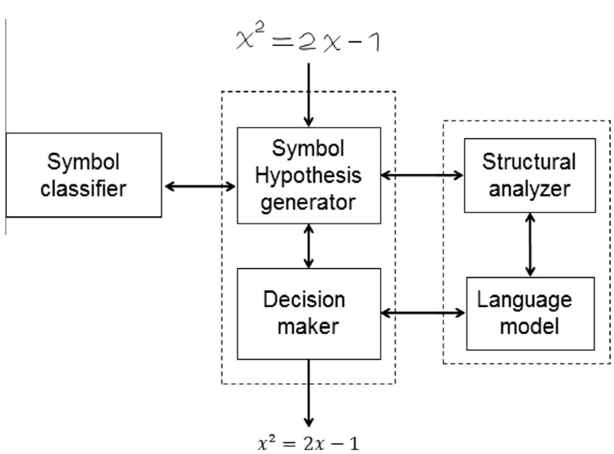

Fig. 2. System architecture.

The generator is based on an extension of a 2D dynamic programming algorithm (2D-DP) in order to allow group strokes which are not consecutive in time. This property is very important in math recognition because it is very frequent to input some delayed strokes to complete a symbol (ex: an extension of a fraction bar or a square root or transforming a symbol in another one by adding an additional stroke, such as ‘–’ transformed in ‘+’). This increases exponentially the number of hypotheses according to the number of strokes. In order to control the number of hypotheses some constraints are used:

a. Maximum number of hypotheses fixed experimentally to 500
b. Maximum number of strokes per hypothesis: small symbols could be written in one stroke, while some others could reach till seven strokes (ex: *arctan*). The limit was fixed to five strokes after studying the distribution of the number of strokes within the learning symbols database (only 0.4% of symbols have more than five strokes)
c. Maximum distance between strokes: avoid grouping strokes that are far away from one another (fixed experimentally to 70% of the average diagonal length of the stroke bounding boxes).
d. Maximum number of temporal jumps: a temporal jump takes place when a symbol is completed after starting another one (delayed strokes). We allow up to two temporal jumps per symbol.

4.2. Symbol classifier

The symbol classifier associates a recognition score and a label with each symbol hypothesis. We have chosen to use a Time Delay Neural Network (TDNN) as symbol classifier. For a given hypothesis sh_i, $p(C_j|sh_i)$ denotes the probability that this hypothesis being the class C_j; with $\sum_j p(C_j|sh_i) = 1$ as a “softmax” function is applied to the TDNN outputs. Many ambiguities of mathematical symbols could be resolved at the global context level. For this reason, some methods consider the best N candidates of the symbol classifier (Yamamoto et al., 2006). Similarly, other approaches delay the decision of ambiguous symbols identity and trying to resolve it globally (Rhee and Kim, 2009). We keep the *topN* candidates with a maximum of N ($topN \leqslant N$). The value of $topN$ is chosen to reach the condition: $\sum_{j=1}^{topN} p(C_j|sh_i) \geqslant k$. The goal of the threshold k aims at keeping only candidates with a strong confidence. In order to be easily combined with other costs (e.g. structural), the recognition score is then converted to a cost using a logarithmic function:

$$C_{reco}(sh_i) = -log(P(c = C_j|sh_i)) \tag{4}$$

Classically, a symbol classifier is used in an isolated symbol context. In this case, the input is always a unique symbol, and we expect the classifier to associate the correct class to this input. In our context, the role of the classifier is not only associating the input to a class but also providing a recognition score to each segmentation hypothesis. Thus, the classifier must have the capacity of identifying wrong segmentations and giving them high costs. In other words, we are in front of a rejection problem. In this paper, we will call the reject class the "junk class".

Among many existing methods we have explored the possibility of using a hybrid classifier containing in cascade a reject classifier and a symbol classifier (Awal et al., 2010a; Zhu et al., 2006). Another way of considering the wrong hypotheses is to add a specific class to the classifier to represent the rejection cases (Wilpon et al., 1990).

This class is added to the classification problem and must be considered during the training phase.

In this paper we use this last presented solution: include the junk class as an $N + 1$ output of the classifier.

Thus, the cost of associating the current hypothesis sh_i to the class C_j is calculated from the recognition score by:

$$C_{reco}(sh_i) = \begin{cases} Cost\_max\,; & \text{if } C_j = junk \\ -\log(P(c = C_j|sh_i)); & \text{otherwise} \end{cases} \tag{5}$$

The objective of giving a very high score to the junk hypotheses is to prevent the decision maker from choosing solutions containing theses hypotheses. A less crisp decision is possible by taking the complement of the junk probability $-log(1 - P(c = junk|sh_i))$. This allows more flexibility especially during the training phase.

4.3. Structural analyzer

Generally, structural analysis is based on the alignment and the size of symbols. So, for one symbol or sub-expression we consider its baseline position (y) and its x-height (h). These values are computed from the bounding box (BB) and depend also on the recognized symbol. For instance, for a letter with an ascender, the baseline is taken at the bottom of the BB, while the x-height is defined as one third of the height of the BB. If we consider a symbol like a 'x' without any ascender or descender, then the x-height is defined as the height of the BB.

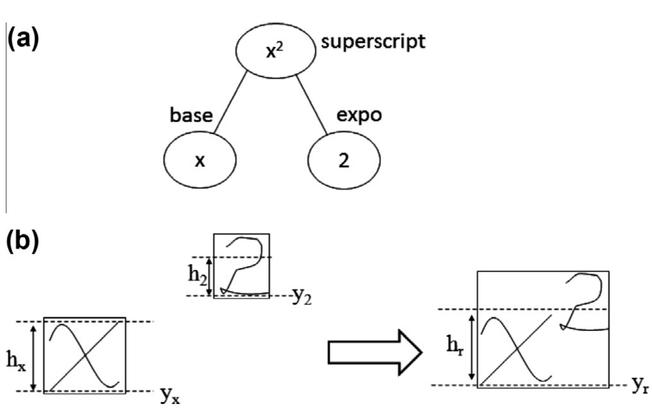

Fig. 3. Superscript relation: (a) tree representation, (b) spatial information.

A sub-expression is built from 2, 3 or 4 children depending on the type of the spatial relation. Its parameters h and y are updated according to the kind of the used relation. In the case of the superscript relation as displayed in Fig. 3, we obtain $y_r = y_x$, and $h_r = f(h_x, h_2)$. In this example, x and 2 are the children of the sub-expression x^2 by the relation "superscript" (and the sub-relations "base" and "exponent").

A structural cost is associated with each non-terminal node according to the type of the relation, using a function of the position and size of its children. An intuitive solution is to calculate a mean square error between the expected (ideal) positions and sizes for a given relation and the observed ones. However, ideal positioning of symbols within mathematical relations are difficult to define because of the fuzzy nature of those relations. In consequence, we define probabilistic costs which correspond to the matching of the observed positions and sizes with Gaussian models computed form a training dataset. The cost that a relation R produces a sub expression SE is given by the equation (see Section 5.2):

$$C_{struct}(R|SE) = -\log(p(R|SE)) \tag{6}$$

4.4. Language model

The language model is defined by a 2D grammar implemented as a combination of two 1D grammars. The first defines rules on the horizontal axis and the second on the vertical one. These rules are applied successively until reaching elementary symbols, and then a bottom-up parse (CYK) is applied to construct the relational tree of the expression. Each production rule of the grammar is associated to a spatial relation that describes the layout of elements of this rule. The application of a rule is penalized by the cost of the corresponding relation. So, each rule of this grammar is activated if its relation is more probable than other rules.

4.5. Decision maker

Finally, the decision maker selects the set of hypotheses E' that minimizes the global cost function and respects the language model using all the strokes of the input expression E.

The global cost of a candidate expression is that of the relational tree root returned by the syntax analyser $C(SE_{root})$ where the cost of a node in the tree is defined by the recursive formula:

$$C(SE_j) = \begin{cases} C_{reco}(sh_j); & \text{if } SE_j \text{ is terminal} \\ \alpha.C_{struct}(R|SE_j) + \sum_i C(SE_i); & \text{otherwise; with } \bigcup_i SE_i = SE_j \end{cases} \tag{7}$$

We will in the coming sections present the global learning schema that it uses to train the symbol classifier as well as the spatial relation learning method.

5. Global learning schema

As explained in (Lecun et al., 1998) the main idea behind the global learning is to model the recognition problem as a sequence of weighted graphs which allow taking a global decision by minimizing a derivable global cost function. This global cost function allows learning the different parts of the system thanks to gradient descent starting from the global context (the whole document). In our context of math recognition, we are close to reach this goal. As described in the previous section, we design a global cost which takes into account the recognition and the segmentation of symbols (thanks to our classifier with a junk class) and the spatial relations between these symbols. The next two sections present the in-context training phases for these two main cost functions: the symbol classifier and the spatial relation analyzer.

5.1. Symbol classifier training

It is not self-evident how to learn the reject class from isolated symbols. There is not an available junk example database, where potentially the junk class represents "everything except a valid symbol". This class should represent sub-parts of symbols, combination of sub-parts of many symbols or even a combination of complete symbols. However, for a simple expression composed of seven strokes, there are 877 possible segmentations and only one is the correct one. As a result the number of junk examples obtained from thousands of expressions could reach several millions. Storing a subset of those millions could be a solution, but it is difficult to choose which examples are more suitable to represent the junk class.

This leads us to the idea of the global learning of the symbol classifier, which is done by directly using the current results of the expression recognition system. So, the classifier is trained with the expression learning database. This training is based on ground truth for the current expression, and the result of the recognition given by the symbol recognizer in its current state. Initially during this training phase the hypothesis generator is used without the language model (the grammar) and a DP algorithm finds the best segmentation and recognition of the expression's symbols. This process is repeated on all the training expressions in order to update the symbol classifier. The classifier must have the capacity for iterative learning (such as a TDNN), where each new example updates the current state of the classifier using a stochastic gradient descent. Fig. 4 illustrates different situations that require updating the classifier. In the case of the symbols ('5', '=', '8'), since the '5' and '=' are correctly recognized, they do not participate in the updating process. In contrast, the symbol '8' is wrongly recognized as a '6', so it is used to train class '8'.

Concerning the segment combining the strokes '3' and '+', it does not have a correspondence in the ground truth and thus represents a wrong segmentation that must be classified as a junk. Furthermore this wrong segmentation was recognized as a '9'. As a result, this segment is used to update the classifier to favor its recognition as a junk. Finally, the segments corresponding to '3' and '+' do not appear in the solution, meaning that probably the local recognition scores of these hypotheses were low. Consequently, they have to participate to the current corrective learning action. As a result, the gradient error which is backpropagated into the network is computed from these selected cases. This process is repeated on the whole expressions database till the convergence of the symbol classifier. This global learning schema allows many learning strategies:

- *Pure global learning*: the classifier is initialized randomly (empty classifier). Then the training expression database is used in a global learning loop to train the classifier.
- *Isolated and then global learning*: before starting the global learning, the classifier is initialized by an isolated symbol database in order to better learn the less frequent classes in the expression database.
- *Global and then isolated learning*: isolated training is done after a global learning for the same reason.
- *Isolated during global learning*: isolated symbols could be used as expressions during the global learning. In this case each symbol is considered as an expression of one symbol.

In this paper, we use only pure global learning.

5.2. Spatial relation modeling

With the *geometric approach* used to evaluate the structural costs, ideal differences of position and size were hypothesized between every components and the corresponding sub-expression obtained when using a specific relation. For instance, in a "Left–Right" (*mrow* in MathML) relationship, the difference in the baseline positions is supposed to be zero. However, for other relationships, assuming ideal values is not trivial. This is why we propose in this section to learn the cost functions from a training set containing samples of the different relations.

The model is based on the differences of the baseline position (y) and x-height (h) of a sub-expression SE_i compared with its parent SE, defined as follows:

$$dh_i = (h_{SE} - h_{SE_i})/h_{SE}; \quad dy_i = (y_{SE} - y_{SE_i})/h_{SE}$$

The differences (dh, dy) are the normalized differences of the position and the size of a child node regarding the sub-expression (independent of the expression scale). The distributions of dh and dy values of each sub-relation of a relation are then modeled by Gaussian models. For example, as displayed in Fig. 5, the relation "superscript" implies two models of size differences, $g_{base}(dh)$ is related to the size difference between the base and the composed expression, $g_{exp}(dh)$ is the difference in size between the exponent and the expression.

We can conclude that the models of the relation "superscript" imply that a base having a size similar to that of the parent is very probable (relative difference being only 0.05). Conversely, the average size of the exponent should be smaller than the global sub-expression. It is worth to note that the size of the parent is not necessary larger than that of the base child; as a result dh can have negative values as shown in Fig. 5.

Structural costs are then computed from these Gaussian models. A sub-expression SE is produced by its sub-expressions

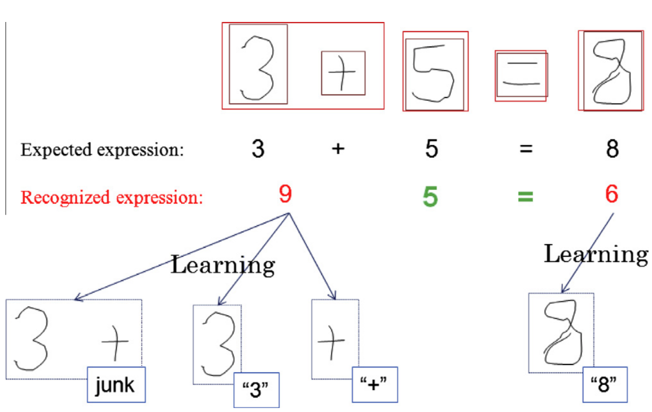

Fig. 4. Updates achieved during the global learning of the expression 3 + 5 = 8.

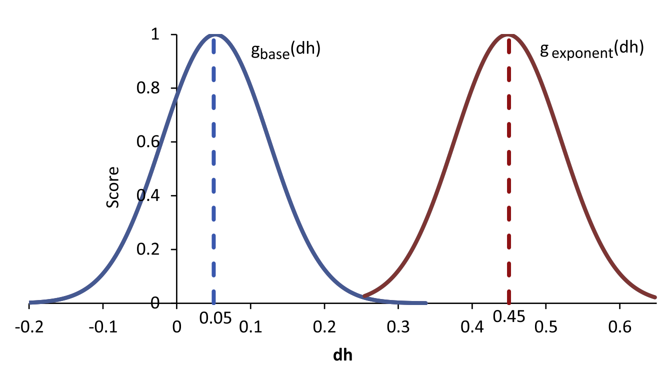

Fig. 5. Gaussian models of the size difference for the "superscript" relation.

$\{SE_1, \ldots SE_i, \ldots, SE_N; N = 2,3,4\}$ and linked by the relation R. We denote $p(SE|R)$ the probability that the sub-expressions SE_i forming SE are related by the relation R, where: $p(SE|R) = \prod_{i=1}^{N} p(SE_i|R_i^s)$; R_i^s is the sub-relation that connects the node of relation R and the ith child. For instance, the relation R = *superscript* has N = 2 sub-relations, $R_1^s = base$ and $R_2^s = exponent$. By applying the Bayes rule, the probability that a relation produces a sub-expression knowing its model is given by:

$$p(R|SE) = \frac{p(SE|R).p(R)}{p(SE)} \tag{8}$$

The term $p(SE)$ can be ignored because it is constant for all the relations, thus the probability of a relation R is:

$$p(R|SE) \propto \prod_{i=1}^{N} p(SE_i|R_i^s).p(R) \tag{9}$$

where $p(R)$ is the a priori probability of the relation R computed from the distribution of relations in the training database. The probability $p(SE_i|R_i^s)$ that a sub-expressions SE derives from the sub-expression SE_i is given by the Gaussian models g using the corresponding spatial information (dh_i, dy_i):

$$p(SE_i|R_i^s) = g_{dh}^{R_i^s}(dh_i).g_{dy}^{R_i^s}(dy_i) \tag{10}$$

where $g_{dx}^{R_i^s}(x) = e^{-\frac{(x-\mu_{R,i,dx})^2}{2\sigma_{R,i,dx}^2}}$ with $\mu_{R,i,dx}$ and $\sigma_{R,i,dx}^2$ are the Gaussian parameters for the ith child of the spatial relation R considering the feature x.

We have defined 11 relations based on two or three elements (Awal, 2010). For example, the "Left–Right" that is constructed from two elements is the most basic one. Other two element relations are the "Sub-script" and the "Super-script" that represent the index and exponent relations. Binary mathematical operations $(+, -, \rightarrow, *, \ldots)$ are represented by a three element relation called "Operator". Each of the other mathematical operations (such as $\sum, \int, (), \ldots$) is associated to a specific spatial relation. For example, the sum relation is modeled by two rules. A vertical one composed of three elements to produce the sum with its limits (such as $\sum_{i=0}^{N}$), the result of this rule is then associated within a horizontal rule to form the complete expression (such as $\sum_{i=0}^{N} x_i$).

During the relation learning phase, relations have been directly extracted from correctly recognized expressions by forcing a correct symbol classification from the ground truth. We adopt this strategy to avoid a handmade labeling of expressions at the relation level. Then the dh and dy of each relation are used to estimate the Gaussian parameters.

6. Experiments

6.1. Expression dataset

The proposed architecture requires a large expression database in order to train the symbol classifier and to model the spatial relations directly from expressions. Obtaining such a database from real handwritten expressions is possible, but not easy to achieve. We have proposed a tool (Awal et al., 2009) that allows producing any dataset of handwritten MEs from a given corpus and previously collected isolated symbols. It generates pseudo-synthetic handwritten MEs using a stochastic layout guided by the LaTeX string defining the expression.

Currently, the corpus is extracted from the "Aster" database proposed in (Raman, 1994). A set of 36 expressions is chosen covering a majority of mathematical domains (Awal et al., 2009). We will call it: "RamanReduced" corpus.

Each expression contains in average 11 symbols, representing 34 distinct classes. First, three databases (1st three lines in Table 1) are generated: each expression is artificially synthesized using isolated symbols from the CIEL database collected from 280 writers (Awal et al., 2009) and the IRONOFF database (Viard-Gaudin et al., 1999).

Expressions in the RamanReduced_CIEL DBs are generated using isolated symbols of only one writer at the time. A drawback of this technique is that the very same sample of a given symbol will be repeated in all the expressions of a given writer. For this reason, we have introduced the notion of virtual writers to produce the RamanReduced_IROCIEL DB. In this case, the number of isolated symbols is much bigger than the required samples. A variety of writers' samples is used to produce one expression in order to assure large symbol variability.

In addition we have collected real expressions where each expression is written entirely by one real writer. In the short term, real databases are only used for testing the system in order to evaluate its performances. Isolated symbols and also real expressions have been collected using a pen and paper technology.

We collected the RamanReduced_Real dataset within our research group. Ten writers wrote the expressions, and as a result 70 expressions have been collected. Secondly, we have collected a larger expression database in order to obtain a big database that can eventually be used as a real train database. A total number of 77000 MEs have been extracted from the French Wikipedia within 7,000 web pages. Those expressions have been filtered using two criteria: the symbol set and the length (between 3 and 49 symbols).

A subset of 6144 expressions has been chosen randomly to be collected using forms printed on *Anoto* papers. University students, professors, and researchers have participated in this collection (512 persons). Each writer filled out two forms for a total of 12 expressions. However, only 211 expressions are compatible with the RamanReduced corpus, thus we refer to this sub-base as RamanReduced_Wiki_CIEL (test).

Since there are few available handwritten expression databases, we have decided to share our real expression databases in order to enrich the comparison between different proposed systems and methods, which is actually not possible because each research group has their own expression databases. RamanReduced_Real and RamaReduced_Wiki_CIEL are publicly available[1]. A sub-part of our real expressions has been used to construct the CROHME database used in the online handwritten MEs recognition competition (Mouchère et al., 2011, 2012).

6.2. Evaluation

It is inappropriate to evaluate an expression recognition system only at the expression level, especially when dealing with long and complex expressions. In consequence, we have chosen three measures similar to those used recently in (Mouchère et al., 2011, 2012; Rhee and Kim, 2009; Zanibbi, 2011), we used the following measures:

$$SegRate = (correctly\ segmented\ symbols)/number\ of\ symbols$$
$$RecoRate = (correctly\ recognized\ symbols)/number\ of\ symbols$$
$$ExpRate = (correctly\ recognized\ expressions)/number\ of\ expressions$$

These same measures are also used in order to validate the recognizer during the training stage.

6.3. Results

The objective of these experiments is to measure the performance of the MEXREC system under different conditions. We focus

[1] AWAL_EM database : http://www.irccyn.ec-nantes.fr/spip.php?article638&lang = en.

Table 1
Constitution of the MEs databases.

Database	#Isolated DB	#Writers	Expression database		Symbol database
			#Expressions	#Symbols	#Symbols
RamanReduced_CIELTraining	CIEL	180	180 × 36 = 6480	180 × 412 = 74,160	180 × 34 = 6120
RamanReduced_IROCIELTraining	CIEL IRONOFF	200	200 × 36 = 7200	200 × 412 = 82,400	180 34 + 480 x 15 = 13,320
RamanReduced_CIELTest	CIEL	100	100 × 36 = 3600	100 × 412 = 41,200	100 × 34 = 3400
RamanReduced_RealTest	n.a.	10	70	784	n.a.
RamanReduced_Wiki_CIEL Test	n.a	20	211	1477	n.a

in this section on the results obtained to evaluate (i) the importance of the junk class, (ii) the impact of the training databases.

Note that the impact of the spatial relation modeling has been studied in (Awal et al., 2010b). Finally, we will compare the system with other systems that have participated in the CROHME 2011 competition.

6.3.1. Importance of the junk class

The symbol classifier is a very important stage of the global system. In addition to recognizing the symbols, it guides the segmentation process and participates in calculating the global cost function. As we have explained before, a classifier trained on isolated symbols has difficulties dealing with invalid segmentations (junk). Table 2 shows the performance of the expression recognizer using a global classifier (TDNN); compared to reference results obtained using an isolated classifier without rejection capacity (TDNN). For this experiment, RamanReduced_CIEL training set has been used.

We can observe that the system performance is significantly improved at the expression recognition level when using a classifier with rejection capacity (using an additional junk class trained in a global scheme). In fact, the *ExpRate* is improved from 25.6% to 61.8% on the RamanReduced_CIEL test DB, and from 11.4% to 27.1% on the RamanReduced_Real DB.

6.3.2. Impact of the training databases

The choice of the training database is essential to efficiently train the symbol classifier. It must be representative of the corpus domain, and also covers the variability of writing styles. When the global learning is based on synthetic expressions, the variability of the expressions can be augmented by enriching the isolated DBs used in the generation process.

Two different training databases are used to obtain the results presented in Table 3. The first, RamanReduced_CIEL is produced classically with the drawback of repeated samples. Where on the other hand the second DB, RamanReduced_IROCIEL, is produced using virtual writer technique. In both cases, the same Gaussian structural models, estimated with the RamanReduced_CIEL DB, are used.

We can conclude from the expression recognition rates shown in Table 3 that the introduction of virtual writers in the training database improves significantly the performance of the system. By doing this, it has been possible to use a bigger set of symbols samples (13,320 instead of 6120). The global performance of the system is improved by 6.1% on the synthetic test DB, and 8.2% on the real test DB (the 281 MEs).

6.3.3. CROHME competition

In fact, direct comparison with existing systems is inappropriate because systems are tested with different expression databases and different evaluation measures (Lapointe and Blostein, 2009) (Awal et al., 2010c). For this reason, an international competition has been held in the International Conference on Document Analysis and Recognition (ICDAR 2011) (Mouchère et al., 2011).

The training and test datasets were subdivided into two parts. Part-I contains 296 training expressions and other 181 for the test. The part-II consists of 921 training expressions and other 348 for the test. The part-II expression set includes the part-I expressions. The grammar controlling the part-II expressions is more complex than that of part-I, and thus the expressions are more difficult to recognize. Moreover, the number of distinct symbols is bigger in the part-II set.

Table 4 shows the results of our system, which was considered out of competition, compared to the winner one. At the time of the 2011 competition, our best system was trained on RamanReduced_IROCIEL using a global classifier and a Gaussian structural model. After this competition, we have updated the system by:

- increasing the usage of real expressions during the training phase with data from the HAMEX database (Quiniou et al., 2011)
- solving some problems in grammar definition and mathML generation
- improving the scaling normalization of symbol hypothesis before recognition

We can observe that at the expression level, our system outperforms the winning system. However, its performance is slightly lower than that of the winner system at the levels of symbol segmentation (87.56–88.07 in Part-I and 84.23–87.82 in Part-II) and symbol recognition (91.67–92.22 in Part-I and 87.16–92.56 in Part-II). However, the system was able to recover at the expression level (global level) and achieves a very good performance compared to the winner system (40.88–29.28 in Part-I and 22.41–19.83 in Part-II). This should mean that our system is more efficient in the interpretation stage. However, a deeper analysis of miss-recognized expressions in term of structure errors should be done. The slightly lower segmentation and recognition rates in the 2011 competition have been overcome in the updated version of our sys-

Table 2
System's performance with or without rejection capacity.

Test dataset	Symbol classifier	SegRate%	RecoRate%	ExpRate%
RamanReduced_CIEL	Isolated (wihout reject)	64.2	62.9	25.6
	Global (with reject)	86.9	84.6	61.8
RamanReduced_Real	Isolated (wihout reject)	50	46.6	11.4
	Global (with reject)	78.7	72.5	27.1

Table 3
System's performance using different training databases.

Test dataset	Training database	SegRate	RecoRate	ExpRate
Synthetic MEs (RamanReduced_CIEL)(3600 MEs)	RamanReduced_CIEL(6480 MEs)	91.4	88.7	64.9
	RamanReduced_IROCIEL(7200 MEs)	94.3	92.1	71.0
Real MEs(281 MEs)	RamanReduced_CIEL(6480 MEs)	83.0	77.4	40.9
	RamanReduced_IROCIEL(7200 MEs)	88.3	84.8	49.1

Table 4
Results on CROHME 2011 Test Set.

Dataset	Systems	SegRate	RecoRate	ExpRate
Part-I	Winner	88.07	92.22	29.28
	Our system in 2011	87.56	91.67	40.88
	Our updated system	89.79	95.21	63.54
Part-II	Winner	87.82	92.56	19.83
	Our system in 2011	84.23	87.16	22.41
	Our updated system	87.04	92.47	47.41

tem. We have succeeded not only to slightly outperform the winning system at the symbol segmentation and recognition levels, but also to increase the expRate by almost 50% for PART-I and 100% for PART-II (40.88 → 63.54 and 22.41 → 47.41 respectively).

7. Conclusion and perspective

A complete system of MEs recognition systems has been presented. The classic three steps of 2D language recognitions are applied simultaneously in order to reduce error propagation from one step to another. We approach the recognition problem as a search for the best possible interpretation of a sequence of input strokes. Unlike most existing works, we have considered a symbol classifier with a reject capacity in order to deal with the invalid hypotheses proposed by the hypothesis generator. This global approach is applicable for any 2D languages, like in our recent work on flowchart recognition (Awal et al., 2011).

The originality of our system stems from the global learning schema. This learning allows training the symbol classifier directly from mathematical expressions. The advantage of this global learning is to consider the junk examples and include them in the classifier knowledge. Furthermore, we have proposed a contextual modeling based on structural analysis of the expression. This analysis is based on models learnt directly from the expressions using the global learning scheme. Although in our current implementation the grammar has only a filtering role on the candidate expression outputs, it might also be possible to take advantage of the global learning strategy to update the spatial relationship models as it is done in the backpropagation step with the neural network classifier. With such a system, we have obtained the best results in the CROHME 2011 competition; we have no doubt that this will stimulate this research area and that for CROHME 2012, new comers will present very competitive systems.

References

Aly, W., Uchida, S., Fujiyoshi, A., Suzuki, M., 2009. Statistical classification of spatial relationships among mathematical symbols. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1350–1355.
Anderson, R.H., 1968. Syntax-directed recognition of handprinted two-dimensional mathematics in interactive systems for experimental. Appl. Math., 436–459.
Awal, A.-M., 2010. Reconnaissance de structures bidimensionnelles: Application aux expressions mathématiques manuscrites en-ligne, Ph.D. Thesis, University of Nantes.
Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2009. Towards handwritten mathematical expression recognition. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1046–1050.
Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2010a. A hybrid classifier for handwritten mathematical expression recognition. In: Document Recognition and Retrieval XVII, San Fransisco, pp. 1–10.
Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2010b. Improving online handwritten mathematical expressions recognition with contextual modelling. In: Internat. Conf. on Frontiers in Handwriting Recognition, Kolkata, pp. 427–432.
Awal, A.-M., Mouchère, H., Viard-Gaudin, C., 2010c. The problem of handwritten mathematical expression recognition evaluation. In: Internat. Conf. on Frontiers in Handwriting Recognition, Kolkata, pp. 646–651.
Awal, A.-M., Feng, G., Mouchère, H., Viard-Gaudin, C., 2011. First experiments on a new online handwritten flowchart database. In: Document Recognition and Retrieval XVIII, San Fransisco.
Belaid, A., Haton, J.P., 1984. A syntactic approach for handwritten mathematical formulae recognition. Pattern Anal. Machine Intelligence 6, 105–111.
Chan, K.-F., Yeung, D.-Y., 2000a. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. Pattern Recognition 33, 375–384.
Chan, K.-F., Yeung, D.-Y., 2000b. Mathematical expression recognition: A survey. Internat. J. Doc. Anal. Recognition 3, 3–15.
Chan, K.-F., Yeung, D.-Y., 2001. PenCalc: A novel application of on-line mathematical expression recognition technology. In: Sixth Internat. Conf. on Document Analysis and Recognition, Seattle, pp. 775–778.
Chang, S.K., 1970. A method for the structural analysis of 2-D mathematical expressions. Infor. Sci. 2 (3), 253–272.
Coüasnon, B., 2001. DMOS: A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems. In: Sixth Internat. Conf. on Document Analysis and Recognition, Seattle, pp. 215–220.
Delaye, A., Anquetil, E., Macé, S., 2009. Explicit fuzzy modeling of shapes and positioning for handwritten Chinese character recognition. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1121–1125.
Dimitriadis, Y.A., Coronado, J.L., 1995. Towards an ART based mathematical editor that uses online handwritten symbol recognition. Pattern Recognition 28, 807–822.
Dimitriadis, Y.A., Coronado, J.L., la, C.d., 1991. A new interactive mathematical editor, using on-line handwritten symbol recognition, and error detection-correction with an attribute grammar. In: First Internat. Conf. on Document Analysis and Recognition, St. Malo, pp. 885–893
Eto, Y., Suzuki, M., 2001. Mathematical formula recognition using virtual link network. In: Sixth Internat. Conf. on Document Analysis and Recognition, Seattle, pp. 762–767.
Feng, G., Viard-Gaudin, C., Sun, Z., 2009. On-line hand-drawn electric circuit diagram recognition using 2D dynamic programming. Pattern Recognition 42, 3215–3223.
Fitzgerald, J.A., Geiselbrechtinger, F., Kechadi, T., 2006. Structural analysis of handwritten mathematical expressions through fuzzy parsing. In: The Internat. Conf. on Advances in Computer Science and Technology, Puerto Vallarta, pp. 151–156.
Fitzgerald, J.A, Geiselbrechtinger, F. & Kechadi, T, 2007. Mathpad: A fuzzy logic-based recognition system for handwritten mathematics. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 694–698.
Fukuda, R., et al., 1999. A technique of mathematical expression structure analysis for the handwriting input system. In: Fifth Internat. Conf. on Document Analysis and Recognition, Bangalore, pp. 131–134.
Garain, U., Chaudhuri, B., 2004. Recognition of online handwritten mathematical expressions. Trans. Systems, Man Cybernet. 34, 2366–2376.
Geneo, R., Fitzgerald, J.A., Kechadi, T., 2006. A purely online approach to mathematical expression recognition. In: Internat. Workshop on Frontiers in Handwriting Recognition, La Baule, pp. 255–260.

Grbavec, A., Blostein, D., 1995. Mathematics recognition using graph rewriting. In: Third Internat. Conf. on Document Analysis and Recognition, MontReal, pp. 417–421.

Ha, J., Haralick, R.M., Philips, I.T., 1995. Understanding mathematical expressions from document images. In: Third Internat. Conf. on Document Analysis and Recognition, MontReal, pp. 956–959.

Keshari, B., Watt, S.M., 2007. Hybrid mathematical symbol recognition using support vector machines. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 859–863.

Kosmala, A., Rigoll, G., Lavirotte, S., Pottier, L., 1999. On-line handwritten formula recognition using hidden Markov models and context dependent graph grammars. In: Fifth Internat. Conf. on Document Analysis and Recognition, Bangalore, pp. 107–110.

Lapointe, A., Blostein, D., 2009. Issues in performance evaluation: A case study of math recognition. In: 10th Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1355–1360.

Lavirotte, S., Pottier, L., 1998. Mathematical formula recognition using graph grammar. In: Proceedings of the SPIE, pp. 44–52.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Lehmberg, S., Winkler, H.-J., Lang, M., 1996. A soft-decision approach for symbol segmentation within handwritten mathematical expressions. In: Internat. Conf. on Acoustics, Speech, and Signal Processing, Atlanta, pp. 3434–3437.

Macé, S., Anquetil, E., 2009. Eager Interpretation of on-line hand-drawn structured documents: The DALI methodology. Pattern Recognition 42, 3202–3214.

Miller, E.G., Viola, P.A., 1998. Ambiguity and constraint in mathematical expression recognition. In: The 15th National Conf. on Artificial Intelligence, Madison, pp. 784–791.

Mitra, J., et al., 2003. Automatic understanding of structures in printed mathematical expressions. In Seventh Internat. Conf. on Document Analysis and Recognition, Edinburgh, pp. 540–544.

Mouchère, H., et al., 2011. CROHME2011: Competition on recognition of online handwritten mathematical expressions. In: 11th Internat. Conf. on Document Analysis and Recognition, Beijing, pp. 1497–1500.

Mouchère, H., et al., 2012. CROHME 2012: Competition on recognition of online handwritten mathematical expressions. In: To appear in 13th Internat. Conf. on Frontiers in Handwriting Recognition, Bari.

Plamondon, R., Srihari, S.N., 2000. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Machine Intelligence 22, 63–84.

Prusa, D., Hlavac, V., 2007. Mathematical formulae recognition using 2D grammars. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 849–853.

Quiniou, et al., 2011. HAMEX – A handwritten and audio dataset of mathematical expressions. In: 11th Internat. Conf. on Document Analysis and Recognition. Beijing, pp. 452–456.

Raman, T.V., 1994. Audio system for technical readings. Ph.D. Thesis.

Rhee, T.-H., Kim, J.-H., 2009. Efficient search strategy in structural analysis for handwritten mathematical expression recognition. Pattern Recognition 42, 3192–3201.

Scott, M., George, L., 2010. Recognizing handwritten mathematics via fuzzy parsing. Technical report. University of Waterloo.

Shi, Y., Li, H.Y., Soong, F.K., 2007. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. In: Ninth Internat.l Conf. on Document Analysis and Recognition, Curitiba, pp. 85–858.

Szwoch, M., 2007. Guido: A musical score recognition system. In: Ninth Internat. Conf. on Document Analysis and Recognition, Curitiba, pp. 809–813.

Tapia, E., Rojas, R., 2003. Recognition of on-line handwritten mathematical formulas in the E-Chalk system. In: Seventh Internat. Conf. on Document Analysis and Recognition, Edinburgh, pp. 980–984.

Tapia, E., Rojas, R., 2005. Recognition of on-line handwritten mathematical expressions in the E-Chalk system – An extension. In: Eighth Internat. Conf. on Document Analysis and Recognition, Seoul, pp. 1206–1210.

Tokuyasuy, T.A., Chou, P.A., 1999. An iterative decoding approach to document image analysis. In: IAPR Workshop on Document Layout Interpretation and its Applications.

Viard-Gaudin, C., Lallican, P.-M., Knerr, S., Binter, P., 1999. The IRESTE On/Off (IRONOFF) dual handwriting database. In: Fifth Internat. Conf. on Document Analysis and Recognition, Bangalore, pp. 455–458.

Wang, X., Shi, G., Yang, J., 2009. The understanding and structure analyzing for online handwritten chemical formulas. In: Tenth Internat. Conf. on Document Analysis and Recognition, Barcelone, pp. 1056–1061.

Wilpon, J.G., Rabiner, L.R., Lee, C.-H., Goldman, E.R., 1990. Automatic recognition of keywords in unconstrained speech using hidden Markov models. IEEE Trans. Acoust. Speech Signal Process. 38, 1870–1878.

Yamamoto, R., Sako, S., Nishimoto, T., Sagayama, S., 2006. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. In: 10th Internat. Workshop on Frontiers in Handwriting Recognition, La Baule, pp. 249–254.

Yuan, Z., Pan, H., Zhang, L., 2008. A novel pen-based flowchart recognition system for programming teaching. Lect. Notes Comput. Sci. 5328, 55–64.

Zanibbi, R., et al., 2011. Stroke-based performance metrics for handwritten mathematical expressions. In: 11th Internat. Conf. on Document Analysis and Recognition, Beijing, pp. 334–338.

Zanibbi, R., Blostein, D., 2002. Recognizing mathematical expressions using tree transformation. Trans. Pattern Anal. Machine Intelligence 24, 1455–1467.

Zhang, L., Blostein, D., Zanibbi, R., 2005. Using fuzzy logic to analyze superscript and subscript relations in handwritten mathematical expressions. In: Eighth Internat. Conf. on Document Analysis and Recognition, Seoul, pp. 972–976.

Zhu, H., Tang, L., Liu, P., 2006. An mlp-orthogonal quassian mixture hybrid model for chinese bank check printed numeral recognition. Internat. J. Doc. Anal. Recognition 8, 27–34.