Recognition of Handwritten Mathematical Expressions

Nicholas E. Matsakis & Paul A. Viola

Artificial Intelligence Laboratory
Massachusetts Institue Of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Our goal is to automatically recognize handwritten mathematical expressions, converting them from pen strokes to formats commonly used by computer mathematics packages. This problem differs significantly from that of recognizing standard prose. In prose, significant constraints can be put on the interpretation of a character by the context it is in. For example, in "t?ree" one can guess the missing letter. However, in mathematics few such simple constraints are present, so that in an expression such as "$3+?$" the context does little to disambiguate the unknown character. The structure of mathematical expressions is also significantly more complex than that of prose. So whereas prose is written linearly from left to right, mathematical symbols can be written above, below, and inside one another, and these spatial relationships are crucial to the interpretation of the expression. This means that even if all the characters are correctly recognized, there still remains the nontrivial problem of interpreting the structure of the expression. (e.g. 245, 2^{45}, and 24^5 are all distinct expressions with the same characters). Hence, it is difficult to apply the Markovian methods which have been applied to parsing one-dimensional texts such as standard prose or programming languages.

Motivation: A great deal of technical work and education still takes place on a blackboard or in a notebook despite the powerful capabilities computers provide for computation and visualization. Typically, a scientist or engineer will do their "thinking" on paper, and then use a computer to visualize specific quantities and relations that are important to their work. This is unlike the task of composing and editing a document, which many people can complete entirely with a computer. One reason that it is difficult to use a computer for the creative stages of technical work is that, unlike regular text, the data used in this process is very difficult to enter with a keyboard and mouse, requiring the user to learn a special command language or use an often cumbersome "point and click" graphical editor. A pen-based interface would be much more natural means of entering diagrams, graphs, and equations into a computer and could add the power of a computer to the creative process from the beginning.

The recognition of mathematical expressions is a natural first step towards such a system, as mathematics is critical to virtually all technical writing and the body of previous work on the recognition of handwritten letters and words could be applied to the problem, since these are major subcomponents of such expressions.

Previous Work: There is a growing body of work on recognizing mathematical expressions. One common approach is to model the grammatical nature of mathematics in one way or another. [1] used a context-free grammar to define legal expressions and [2] used a probabilistic context-free grammar to provide a maximum likelihood framework the problem, searching for the most likely interpretation in a best first search. A previous effort from our group [4] also parsed an expression as a probabilistic CFG using a novel contraint based on the convex hulls of subexpressions. [3] uses a somewhat different approach, breaking the problem into the subproblems of grouping the strokes into symbols, parsing the symbols based on a geometric grammar, and then verifying the final expression for syntactic correctness.

Approach: We begin by considering an expression to be recognized as a time-ordered series of strokes, collected with some device such as a pen tablet. Recognition is performed in two distinct stages, with hard decisions being made at the end of each one. In the first stage, called *partitioning*, the set of strokes are partitioned into symbols, where each symbol consists of one or more strokes. In the second stage, called *parsing* the structure of the expression is determined, based on the layout of the symbols on the page.

Handwritten expressions typically contain more than a single symbol, so the partitioning stage is necessary to determine the quantity, location, and identity of the symbols in an expression. Partitioning is done using a maximum likelihood approach, where the optimal partition is intended to be the one whose individual symbols are most likely under a set of symbol models. Because the number of possible partitions grows exponentially with the number of strokes, it is impossible to search every partition for the most likely. Other systems [3] have constrained the user to complete a symbol before writing the next, thereby using the timing of the strokes to constrain the search. Our system uses a *minimum spanning tree* to constraint the search space. Since an MST is based on the spatial arrangement of the strokes, the user can write the strokes in any order and the system can still recognize the expression.

Once the expression has been correctly partitioned into symbols, there still remains the problem of determining the structure of the resulting typeset expression. We do this using a geometric grammar whose elements are inspired by the atomic elements of TEX's typesetting engine. In this grammar, each character belongs to a single grammar type and combines with other characters in well defined ways based on simple relationships between their bounding boxes. In addition, simple characters also have a baseline which aids in determining when a character is superscripted.

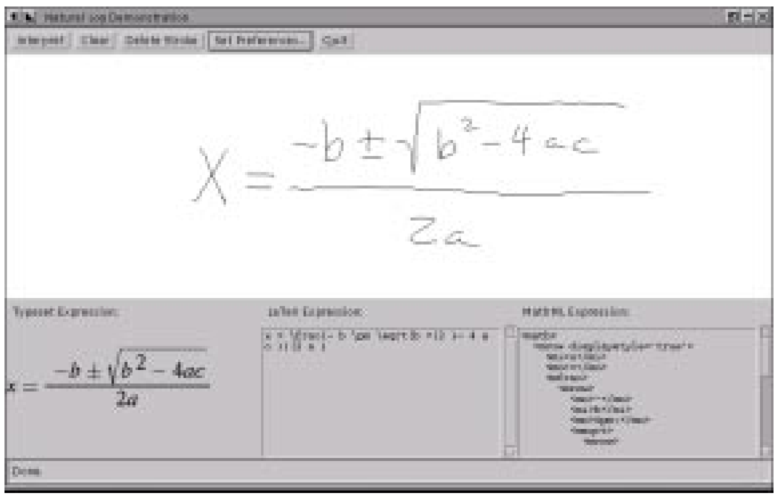

Figure 1: Screen image of the recognition system

Impact: Solving this problem could have a significant impact on the use of computers in the fields of education, engineering, and scientific research. The most immediate applications would simply be improvements in interfaces for existing systems, but often new interfaces open up the possibility for new applications. For example, this technology could be used to help elementary school students with their math or allow scientists to do most of their work on a notebook sized computer with a touchscreen.

Future Work: Though our current system shows much progress, recognizing nested fractions, summations, and radicals, it provides restricted support for some forms, such as superscripts, and no support for others, such as integrals. We hope to augment its parsing framework by adding a stage where related characters are first clustered together followed by a stage where clusters are combined into larger expressions. This should allow for significantly more complex expressions to be parsed. In addition, we are also working on the problem of multiuser recognition, since the current system has only been trained on handwriting from a single user.

Research Support: This work has been supported by a grant from the Nippon Telegraph and Telephone Corporation.

References:

[1] Yiannis Dimitriadis and Juan López Coronado Towards an ART based Mathematical Editor, that uses On-line Handwritten Symbol Recognition *Pattern Recognition, Vol 28, No 6*, pp. 807-822, 1995.

[2] J. Hull. Recognition of Mathematics Using a Two-Dimensional Trainable Context-Free Grammar. *Masters Thesis*, Dept. of EECS, MIT, Jun 1996.

[3] H. Winkler and H. Fahrner and M. Lang. A Soft-Decision Approach for Structural Analysis of Handwritten Mathematical Expressions. *International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[4] E. Miller and P. Viola. Ambiguity and Constraint in Mathematical Expression Recognition. *Proceedings of AAAI -98* pp. 784-791.