Recognizing Mathematics:
Techniques, Challenges and Opportunities

Richard Zanibbi

Department of Computer Science
Rochester Institute of Technology
Rochester, NY, USA
Overview

Motivation
- Applications
- Motivation from the pattern recognition perspective

Challenges and Techniques
- Key problems
- Algorithms that attempt to solve these problems
- Performance evaluation

Opportunities
Motivation:
Applications, and the Nature of our Problem
Existing Applications

Math Input
2D drawings vs. strings (e.g. LaTeX), template editors (e.g. MS EE)
OCR, symbol retrieval by drawing (e.g. Σ, ∃, ∫, Π, ∇)

CAS (Computer Algebra System) integration
Mathematical sketching (e.g. plotting and animating computations)

Education: Electronic Whiteboards, Computer-Assisted Tutoring

Math Document Accessibility for the Visually Impaired

Digital Libraries: Formally Encoding Math Content in Documents
(Early) Pen-Based Entry:  
**Natural Log and FFES/DRACULAE**


* http://www.cs.rit.edu/~rlaz/ffes/

* Pen Math 2004

Mouse and Keyboard Entry: XPress

\[ \int_{0}^{1} \frac{\tan^2(x) \, dx}{\sqrt{x^2 + 2x + 1 + \ln(x^2 + 1)}} \]


* [http://eqn.xero.ca](http://eqn.xero.ca)
The common support of $\sigma$ and $v$ is just the zero set of $f(z)$. We introduce the functions:

$$
\sigma_a(r) = \int_{||z-a|| \leq r} \sigma, \quad v_a(r) = \int_{||z-a|| \leq r} v.
$$

Both are positive increasing functions of $r$. Then,

$$
v_a(r) = \frac{(n-1)! \sigma_a(r)}{r^{n-1}(2n-2)}. \quad (2.00)
$$

If we write $z = x + iy$, then the Laplacian is:

$$
\Delta = \sum_{j=1}^{n} \left( \frac{\partial^2}{\partial x_j^2} + \frac{\partial^2}{\partial y_j^2} \right) = 4 \sum_{j=1}^{n} \frac{\partial^2}{\partial z_j \partial \bar{z}_j}.
$$

An easy calculation shows that:

$$
\sigma^2 = \sum_{j=1}^{n} \left( \frac{\partial^2}{\partial x_j^2} + \frac{\partial^2}{\partial y_j^2} \right).
$$

If we write $z = x + iy$, then the Laplacian is:

$$
\Delta = \sum_{j=1}^{n} \left( \frac{\partial^2}{\partial x_j^2} + \frac{\partial^2}{\partial y_j^2} \right) = 4 \sum_{j=1}^{n} \frac{\partial^2}{\partial z_j \partial \bar{z}_j}.
$$

An easy calculation shows that:


Symbol Lookup: Maple
Online Evaluation: Pen-Calc


CAS Integration: 

**MathBrush and MathInk**

\[
\sin(x)^2 + \sqrt{\sin(x)^2 + (x - 1)^20 + \cos(x)^2}
\]

\[
\sin(\alpha) = \frac{2S_{\triangle ABC}}{b \cdot c}
\]


* http://www.cs.uwaterloo.ca/scg/mathbrush/

Classroom: \textit{JMathNotes/E-Chalk}


* \url{http://mathfor.mi.fu-berlin.de/pmwiki/pmwiki.php?n=Main.HomePage}
Pen-Input for Computer-Assisted Tutoring: CMU Prototype


Properties of Math Recognition (from a Pattern Recognition Perspective)

A non-trivial computer vision problem with smaller input sizes and more constraints than problems involving natural scenes.

Semi-formal visual language recognition problem: simpler than NLP, but involves ambiguities, dialects, and context-dependency in the analysis of symbol identity, layout, and semantics.

Complete end-to-end system requires all key PR techniques: segmentation, classification, parsing, and machine learning.

Opportunity: A good domain for studying recognition algorithm combinations, and theories predicting their effect.
Challenges and Techniques: What We’re After, How We Try to Get It
Four Key Challenges in Math Recognition

1. Detect Expressions
2. Detect and label symbols (OCR)
3. Symbol layout analysis
4. Syntactic & semantic analysis

Detect = Segment, Label = Classify, Analyze = Parse/Translate

Often addressed in sequence, but these challenges interact (e.g. symbol identity influences layout and vice versa)
**Challenge 1:**

**Detect Expressions**

**INPUT:** Images, Strokes (Digital Ink), Markup (e.g. HTML): characters

**OUTPUT:** Math/non-math regions, expressions within math regions

---

denoted $\theta$, takes value in $\Omega = \{\omega_1, \ldots, \omega_c\}$ with probabilities $\{p(\omega_i), \ldots, p(\omega_c)\}$, respectively and that $x$ is a realization of a random vector $X$ characterized by a conditional distribution $p(x|\theta)$, $\theta \in \Omega$. Thus, the task is to find a measurable mapping $\psi: \mathbb{R}^d \rightarrow \Omega$ such that the expected loss function $R(\psi) = E\{L(\psi(x), \theta)\}$, called risk, is minimal. Here $L(\omega_1, \omega_i)$ is the loss incurred by taking action $\omega_i$ when the class is $\omega_j$. In this paper we assume, without loss of generality, that $L(\omega_i, \omega_i) = 0$ for $\omega_i = \omega_j$ and $L(\omega_i, \omega_i) = 1$ for $\omega_i \neq \omega_j$ and then $R(\psi) = P(\psi(X) \neq \theta)$ is called the probability of error. It is well known that an optimal rule $\psi^*$ (the Bayes rule) which minimizes $R(\psi)$ is of the following form $\psi^*(x) = \arg\max_{\theta} p(x|\theta)$ where $p_i(x) = P(\theta = \omega_i | X = x)$, $i = 1, \ldots, c$ are the posteriori probabilities. Let $R^*\eta$ denote the Bayes risk, i.e., the error of the Bayes rule. In practice we rarely have any information about the distribution of the pair $(\theta, X)$, instead there is in our disposal a training set $\eta_s = \{(\theta_1, X_1), \ldots, (\theta_s, X_s)\}$, i.e., a sequence of pairs $(\theta, X)$ distributed like $(\theta, X)$, where $X_s$ is the feature vector and $\theta_i$ is its class assignment. An empirical classification rule $\psi_s$ is a measurable function of $X$ and $\eta_s$. It is natural to construct a rule which resembles the Bayes rule, i.e., by replacing $p_i(x)$ by its estimate $p_s(x)$. A popular nonparametric classification technique is the kernel classifier being defined as follows

$$\psi_s(x) = \arg\max_{\theta} \sum_{s=1}^S I(\theta_s = \omega_i) W\left(\frac{x - X_i}{b}\right).$$  \hfill (1.1)

---


Techniques for Detecting Expressions

Images

Course classification for CC’s; projection analysis (displayed) and region-growing based on CC typography and operator dominance (embedded) - (Kacem et. al, 2001)

Detect embedded expressions based on textline n-grams for math vs. text (Garain & Chaudhuri, 2004) (report 97% rec. rate)

CC Neighbour Graph Attributes (pruned Delaunay triangulation) (for (isolated) displayed expressions vs. text: ~99.9% rec. rate)

Correspondence between text OCR and math OCR output (Kanahori & Suzuki, 2006)

Digital Ink

Gestures (Tapia and Rojas, 2003; LaViola & Zeleznik, 2004)

Region-growing based on symbol identity and size (Li, Zeleznik & Miller, 2008)
Challenge 2: Detect & Label Symbols (OCR)

**INPUT:** Connected components (images), strokes (ink)

**OUTPUT:** Location and identity of symbols (may include alternatives)

---


OCR Techniques: Images

Refining commonly confused classes via SVMs (Malon, Uchida & Suzuki, 2008)

Recursive center-of-mass splitting feature (Sexton & Sorge, 2005)

Clustering CC’s to detect and correct confusions (via majority vote) and expedite manual correction (Suzuki et al., 2003)
OCR Techniques: Digital Ink

Max # strokes per symbol + estimating optimal classification of stroke partitions (Smithies, Novins & Arvo, 1999)

MST partitions used to segment strokes; classification via PCA on evenly sampled strokes (15 components) + Bayesian classification (Matsakis, 1999)

Legendre-Soblev basis functions (Char & Watt, 2007) + SVM (Keshari & Watt, 2007/2008), improved confidence measures using decision boundary distances (Golubitsky & Watt, 2009)

Ensemble of Binary Classifiers + AdaBoost + MS Classifier (Laviola & Zeleznik, 2007)

NN-based template matching + HMM (Garain & Chaudhuri 2004)

Elastic template matching (dynamic programming) (Chan & Yeung, 2001)
Challenge 3: Symbol Layout Analysis

**INPUT:** Symbol locations and labels

**OUTPUT:** Tree representing symbol layout; tokens for operators (=), relations, arguments ('50.2') and functions (e.g. ‘sin’); e.g. LaTeX, Presentation MathML

---


Layout Analysis Techniques

Attribute Grammar w. top-down parse (Anderson, 1968), Graph Grammars (Grbavec & Blostein, 1995), Context-Sensitive: (Lavirotte and Pottier, 1997); DCG-based operator-driven hierarchical decomposition (Chan & Yeung, 1999); Top-down parse of CFG (LR along baseline) (Fateman et al., 1996)

Stochastic context-free grammar tolerating noise (bit flipping) (Chou, 1989) - more recent SCFG (Miller & Viola, 1998); Soft-decisions (Winkler, Fahrner & Lang, 1995)

Spatial relationships from relative BB positions (Wang & Faure, 1988); Modified to include normalized BB sizes and centers (Eto & Suzuki, 2001; Ali, Uchida and Suzuki, 2008)

Adjacency graph (nodes: symbol alternatives, edges: inline, super, subsc with penalties) + MST to determine minimum cost layout tree (Eto & Suzuki, 2001)
Layout Analysis Techniques, Cont’d

Group operator arguments (dominance analysis) + merge adjacent terms (Chang, 1970; Lee & Wang, 1997)

Projection Profile Cutting (Okamoto & Miao, 1991; Garain & Chaudhuri, 2004; Raja et al., 2006)

Baseline Extraction (Zanibbi, Blostein & Cordy, 2002), with MST-based partitioning, revised layout model with “attractor” points (Tapia & Rojas, 2004), with fuzzy relationships (Zhang, Blostein & Zanibbi, 2005)

Matrices

Clustering elements by proximity and/or region growing (Lee & Wang, 1997; Tausky, 2007)

Estimation of rows and columns via projection analysis, handling ellipses (Kanahori & Suzuki, 2002; Tausky, 2007)
Challenge 4: Syntactic & Semantic Analysis

**INPUT:** Layout tree for tokens

**OUTPUT:** Operator tree representing concepts/operations, e.g. as Lisp, Content MathML, or OpenMath

\[
\int \frac{x^2 - z^2}{z-1} \frac{dz}{z+r} = \frac{\pi}{1+r} \left( \frac{r^2 - \cos \pi x}{\sin \pi x} - \frac{r^4 - \cos \pi x}{\sin \pi x} \right)
\]

---


---


---

Further Challenges for Syntactic & Semantic Analysis

Key Issues

Precedence, associativity of operators (expression grammars)

Correct mapping of operator and argument tokens to valid for appropriate mathematical context (operator semantics)

Identifying relationships between equations (e.g. above)

Inferring implicit operations (e.g. implicit multiplication)

Interpolation (for ellipsis in matrices (Sexton et al., 2006; Tausky 2007))

Additional Processing: Math. Domains and Verification

Document Classification

By MSC using symbol frequencies estimated from the arXiv (Watt, 2008)

Error Detection and/or Repair

Syntax error detection (Toyota, Uchida & Suzuki, 2006; Fujiyoshi, Suzuki & Uchida, 2008)

Sheet 5 Area 0 Line 4 Position 0752
SYNTAX ERROR: "less" is in unexpected position
[w_2] → < w' > ∈ B_2(F, ω, L_F)

Sheet 17 Area 3 Line 26 Position 26434
STRUCTURAL ERROR: "LeftPar" may not have a subscript

\[ \beta = \min\left(\beta, \frac{\varepsilon}{1 + \varepsilon}\right) \]

Error-correcting parsing (via heuristics) for token, syntactic, semantic errors (Chan & Yeung, 2001)

Modifying OCR results using symbol layout n-grams (Smirnova & Watt, 2008; Watt 2008)
Performance Evaluation

Metrics

Recognition Rates (“% correct”)
  • for symbols, expressions, baselines (ZBC, 2001), placement of symbols on baselines (ZBC, 2001), operators in op. tree (C&Y, 2001)

Recall and Precision (detection tasks, including per-class rates for OCR)
  • equal error rate (false negative = false positive )

Performance Metrics for Complete Systems

Problem: “Cascading Errors”

(Chan & Yeung, 2001) Recognition rate for symbols + operators in op. tree

(Garain & Chaudhuri, 2004) Adaptation of C&Y: layout errors weighted down as symbol depth in layout tree increases

\[
\gamma = 1 - \frac{S_c + \sum_i E_i \times \frac{1}{i+1}}{S_t + \sum_i R_i \times \frac{1}{i+1}}.
\]
<table>
<thead>
<tr>
<th>Reference</th>
<th>Origin</th>
<th>Availability</th>
<th>Dataset Composition</th>
<th>Ground truth Acquisition</th>
<th>Ground truth composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashida et al. [7]</td>
<td><em>Archiv der Mathematik Commentarii Mathematici Helvetica</em></td>
<td>Not available due to copyright on the digital libraries</td>
<td><strong>Symbol Recognition:</strong> 1,400 pages (43,495 typeset expressions) <strong>Structure Analysis:</strong> 700 pages (21,472 typeset expressions)</td>
<td>Automatic recognition and manual correction</td>
<td>Expression/symbol bounding boxes and labels. Expression structure (in extended MathML format)</td>
</tr>
<tr>
<td>Takiguchi et al. [39]</td>
<td><em>Fundamental Formulas of Physics</em> [27]</td>
<td>Not specified</td>
<td>15 typeset expressions (289 symbols)</td>
<td>Not specified</td>
<td>Not specified</td>
</tr>
<tr>
<td>Phillips [34] (U. of Washington Databases)</td>
<td>Various Sources</td>
<td>Available on CD-Rom</td>
<td>Mathematical Content: 25 pages (approx. 100 typeset expressions)</td>
<td>Double entry and triple verification.</td>
<td>For the math expressions: LaTeX and expression/symbol bounding boxes and labels (in Xfig format).</td>
</tr>
<tr>
<td>Suzuki et al. [37] (Inftry)</td>
<td>30 English articles on pure mathematics</td>
<td>Available at <a href="http://www.inftyproject.org">http://www.inftyproject.org</a></td>
<td>467 pages (21,056 typeset expressions)</td>
<td>Manual ground truthing</td>
<td>\LaTeX, MathML and IML</td>
</tr>
</tbody>
</table>

*Adrien Lapointe, MSc. Thesis pp. 45-46 (Queen’s University, 2008)*
Opportunities
Opportunities

Pattern Recognition

• Combine algorithms (e.g. OCR, parsers) and recognition strategies (i.e. complete systems, subsystems) develop theory to predict effect of combinations.
  • AdaBoost (Freund and Schapire, 1996), Graph Transformer Networks (LeCun et al., 1998), Recognition Strategy Language (Zanibbi et al. 2005)
  • Language models: n-grams, grammars, inference/abstraction of math domains/dialects
  • Adapt/develop features, algorithms, training (machine learning) data sets (esp. handwritten data) & performance metrics

Information Retrieval

• Annotate mathematical information, integrate search for math in document images, web pages, and mathematical knowledge databases (e.g. Mizar, Coq)
Thank you.

Acknowledgements

David Tausky and PenMath 2009 Org. Committee
Xerox Corporation (UAC Grant w. Bill Stumbo)
CAT-EIS / NYSTAR
Additional Slides
Opportunities: The Human Element

Human-Computer Interaction

• Accessibility: evolve math recognition systems for the visually impaired

• Experiments comparing visualizations for program state (e.g. input editor and CAS state), and recognition feedback (task-dependent?)

• Multi-modal input: drawing (pen, mouse & keyboard), string languages (e.g. LaTeX), OCR, and voice

Education

• Experiments testing whether/how math recognition may help students and instructors (classroom, notes/materials, online courses, tutoring)
Four Key Challenges in Math Recognition

1. Segmenting Mathematical Expressions
   • **INPUT:** Images, Strokes (Digital Ink), Markup (e.g. HTML): characters
   • **OUTPUT:** Math/non-math regions, expressions within math regions (e.g. lines of a derivation, elements in a matrix)

2. OCR (Segment and Classify Symbols)
   • **INPUT:** Connected components (images), strokes (ink)
   • **OUTPUT:** Location and identity of symbols (may include alternatives)

3. Syntax Analysis: Parsing Symbol Layout
   • **INPUT:** Symbol locations and labels
   • **OUTPUT:** Baseline structure tree, tokens for operators/relations (=), arguments (‘50.2’) and functions (e.g. ‘sin’); e.g. LaTeX, Presentation MathML
Key Challenges, Cont’d

4. Semantic Analysis

- **INPUT**: Baseline structure for tokens

- **OUTPUT**: Operator tree, representing the concept/operations represented, e.g. Lisp, Content MathML, OpenMath