Searching Documents with Text and Mathematical Content Using a Pen-Based Interface

George Labahn
University of Waterloo

WHJIL Workshop 2020

Outline

Brush-Search Team

University of Waterloo

- **PIs**: George Labahn, Frank Tompa
- **Research Associate**: Mirette Marzouk
- **PhD Students**: Avery Hiebert, Besat Kassaie
- **Masters Students**: Vincenzo Heska, Kiki Ng
- **Undergraduate Research**: Kevin Wang

Huawei

- **Huawei Collaborator**: Michael Feng

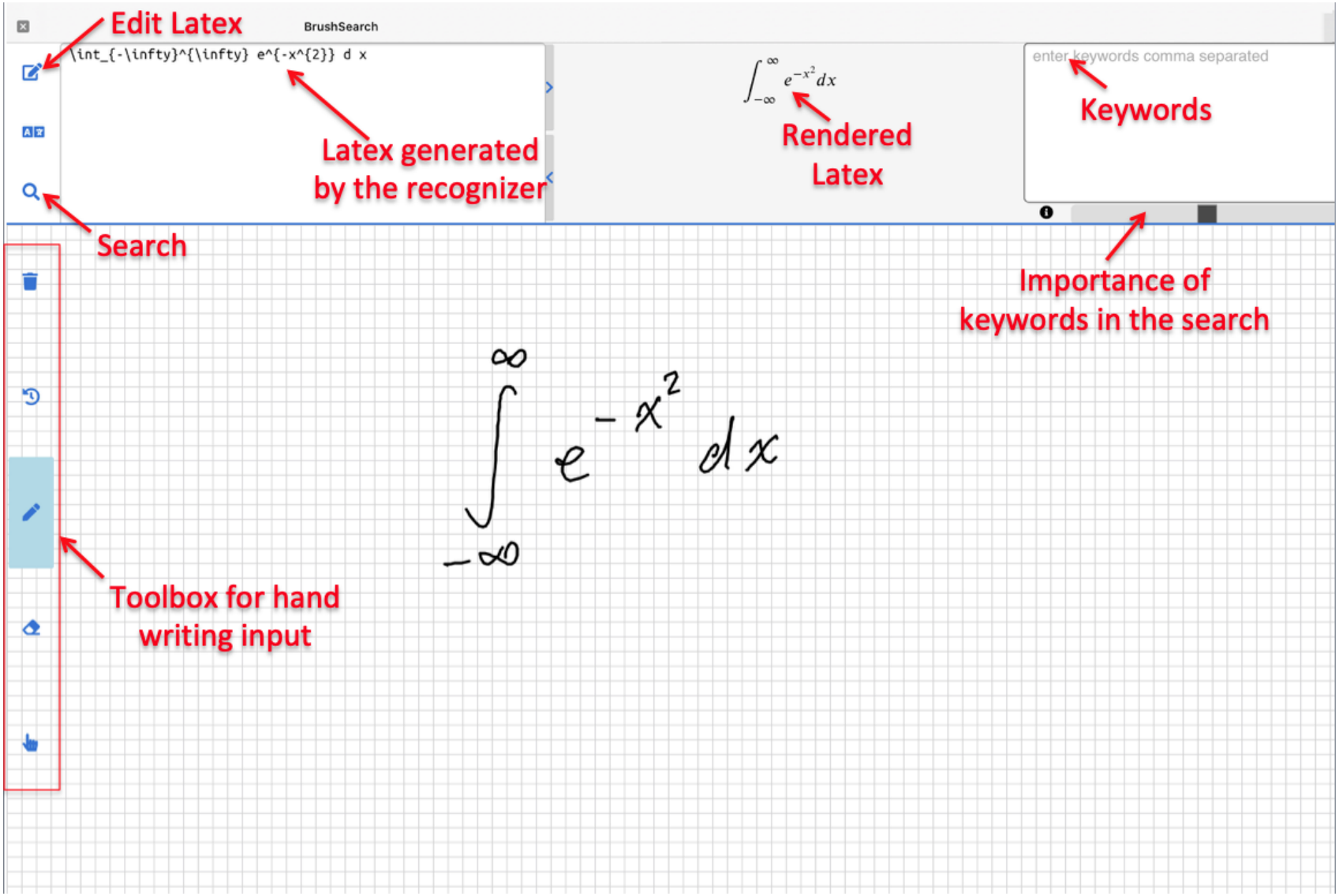
Objective

Build **MathBrush-Search**, a math-aware search system

- An intuitive front end (recognizer and user interface) that accepts handwritten mathematical formulas and supports use of natural gestures to specify constraints and wildcards

- A math-aware search engine that uses text and mathematical content, combining their semantics, and considering users' provided constraints to get the most relevant search results

MathBrush-Search System Demo



MathBrush-Search Road Map

Current Status :

- **Front-end**: simple web interface that interacts with our MathBrush recognizer to generate Latex of recognized expression.

- **Search engine**: our Tangent-L search engine that receives mathematical formula and keywords to generate relevant search results.

Approach :

Use, customize and significantly enhance components from **MathBrush recognizer** and **Tangent-L** and build a user interface for best user experience.

Math-aware Front-end: What is There?

Handwriting interfaces

- Generate LaTeX and MathML.

  Example: Equation editor from MyScript

- Integrate with CAS to do mathematics.

  Example: **MathBrush** from University of Waterloo

Text-based search interfaces with LaTeX expressions

- Search the web or allow different domains.

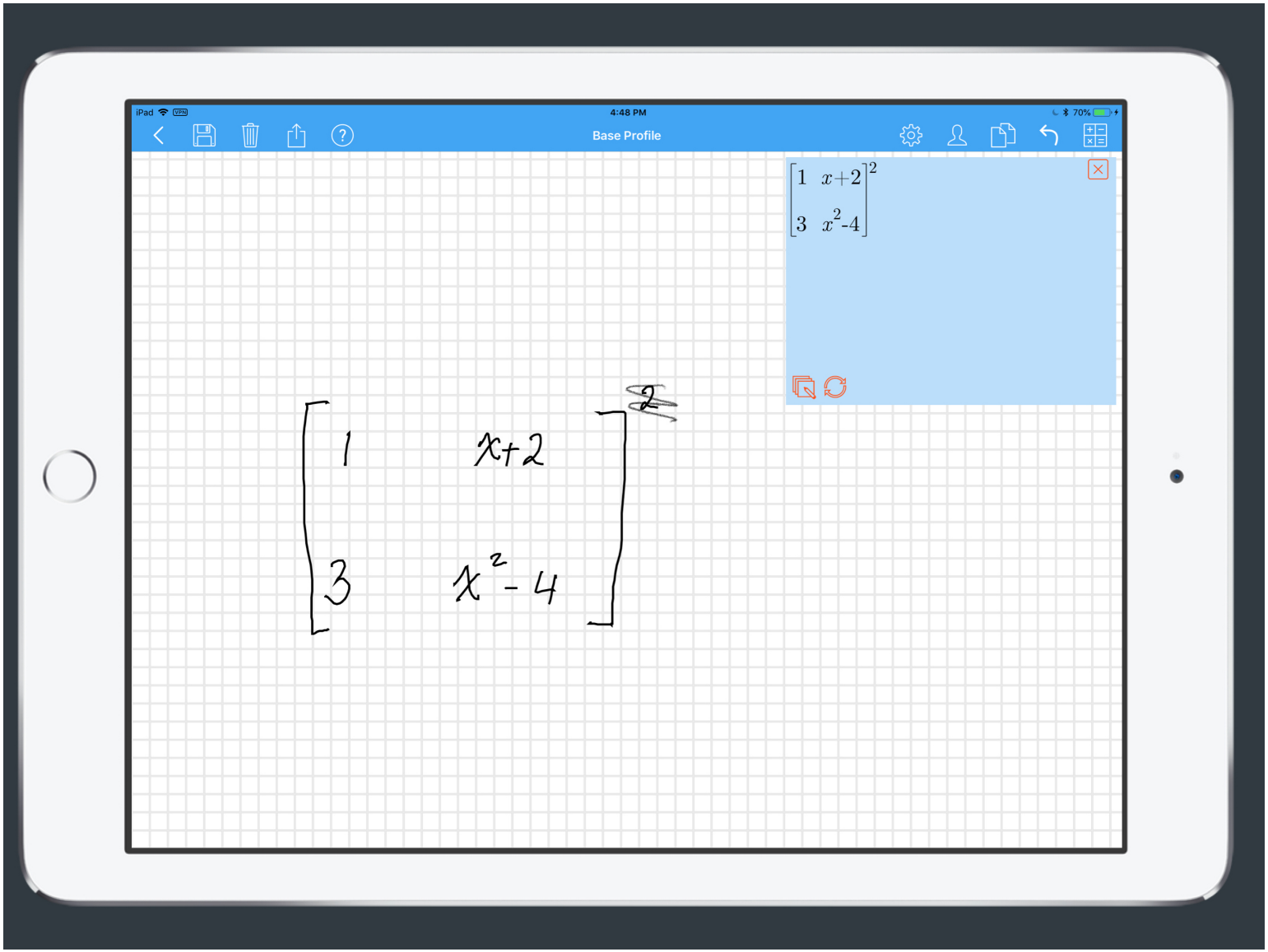
  Examples: search engines, SearchOnMath, and Approach0

- Search specific contents.

  Examples: Wolfram Alpha, Math StackExchange, arXiv

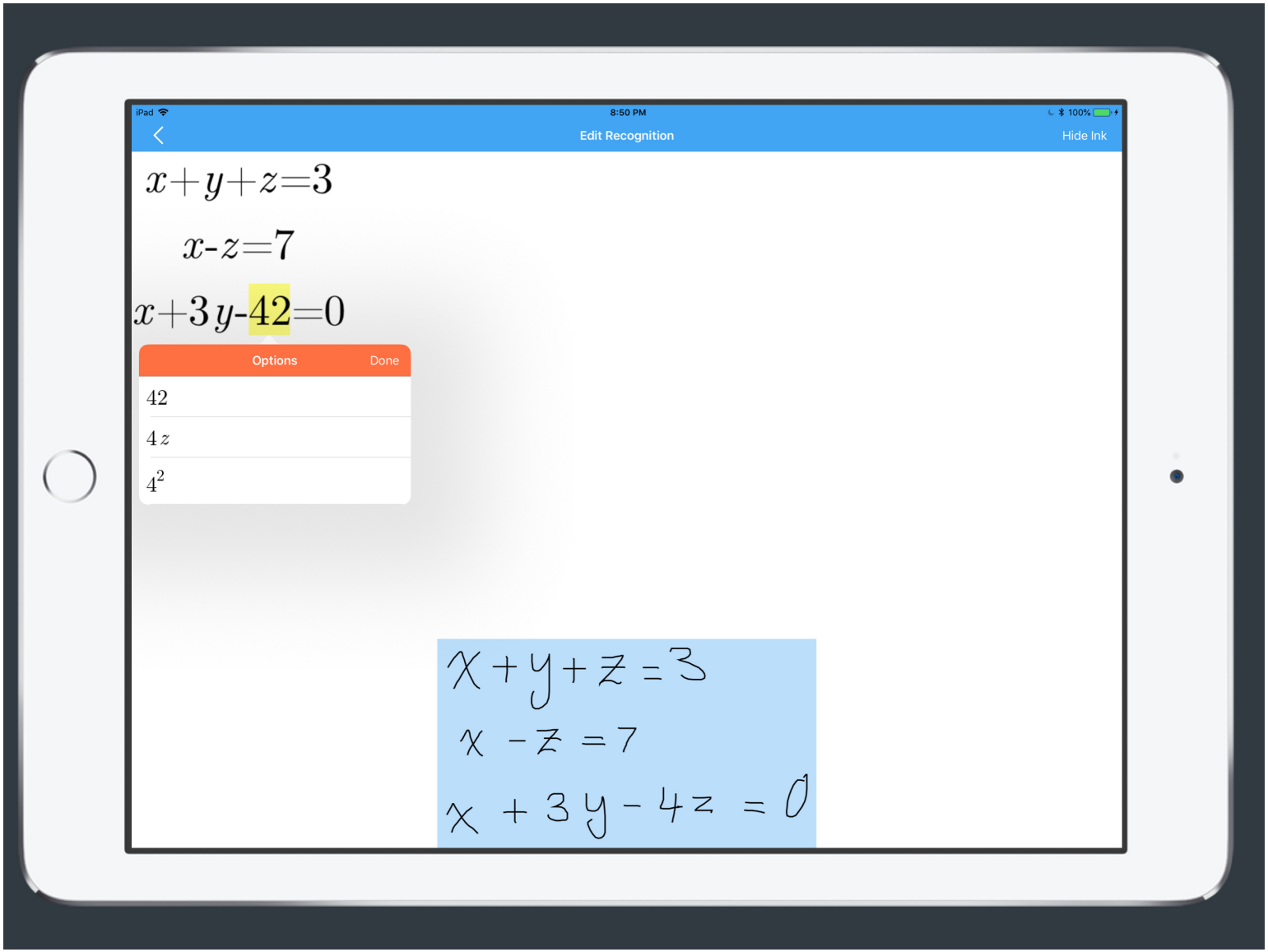
MathBrush App

Handwritten Input



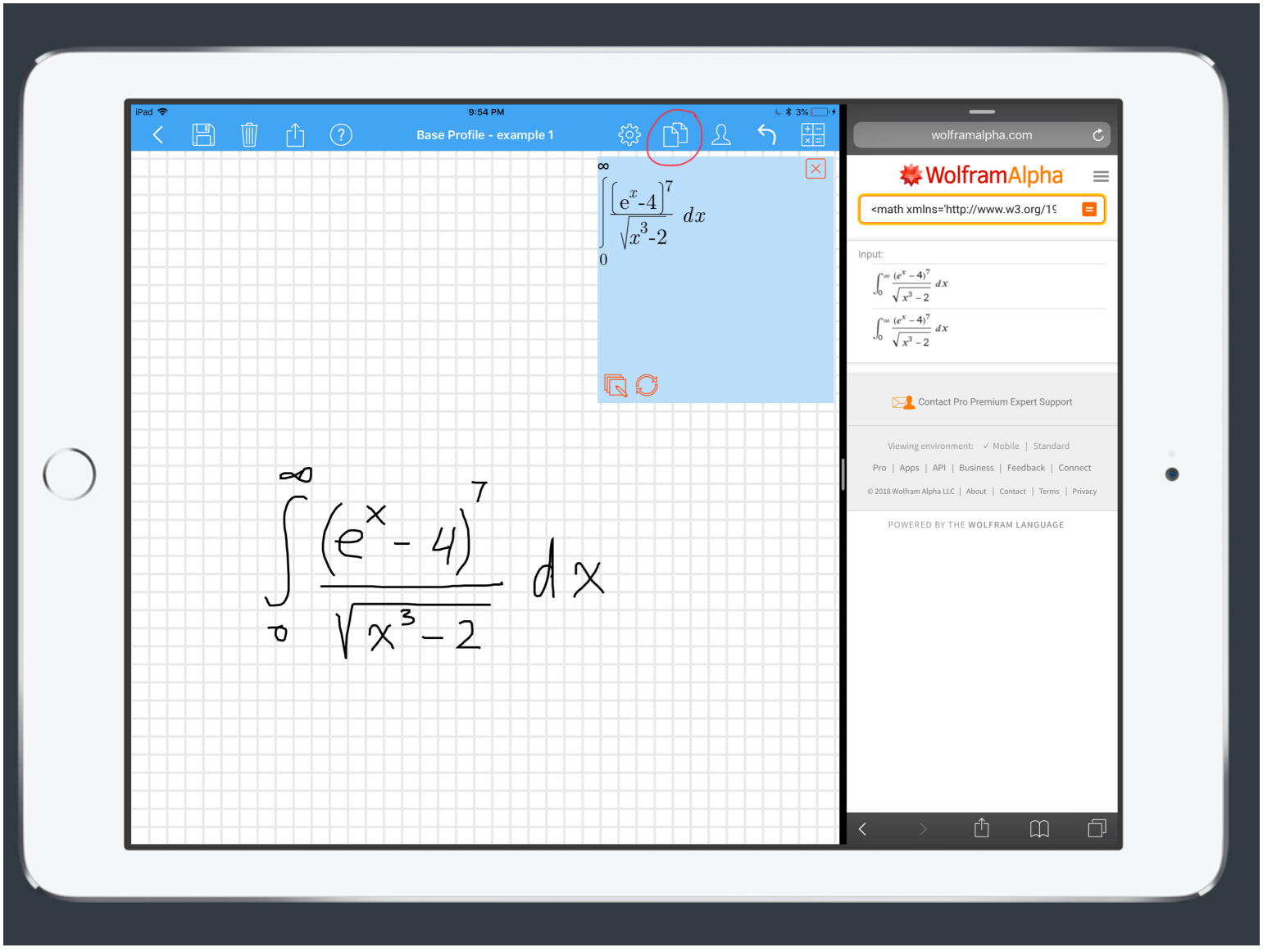
MathBrush App

Correcting Recognition



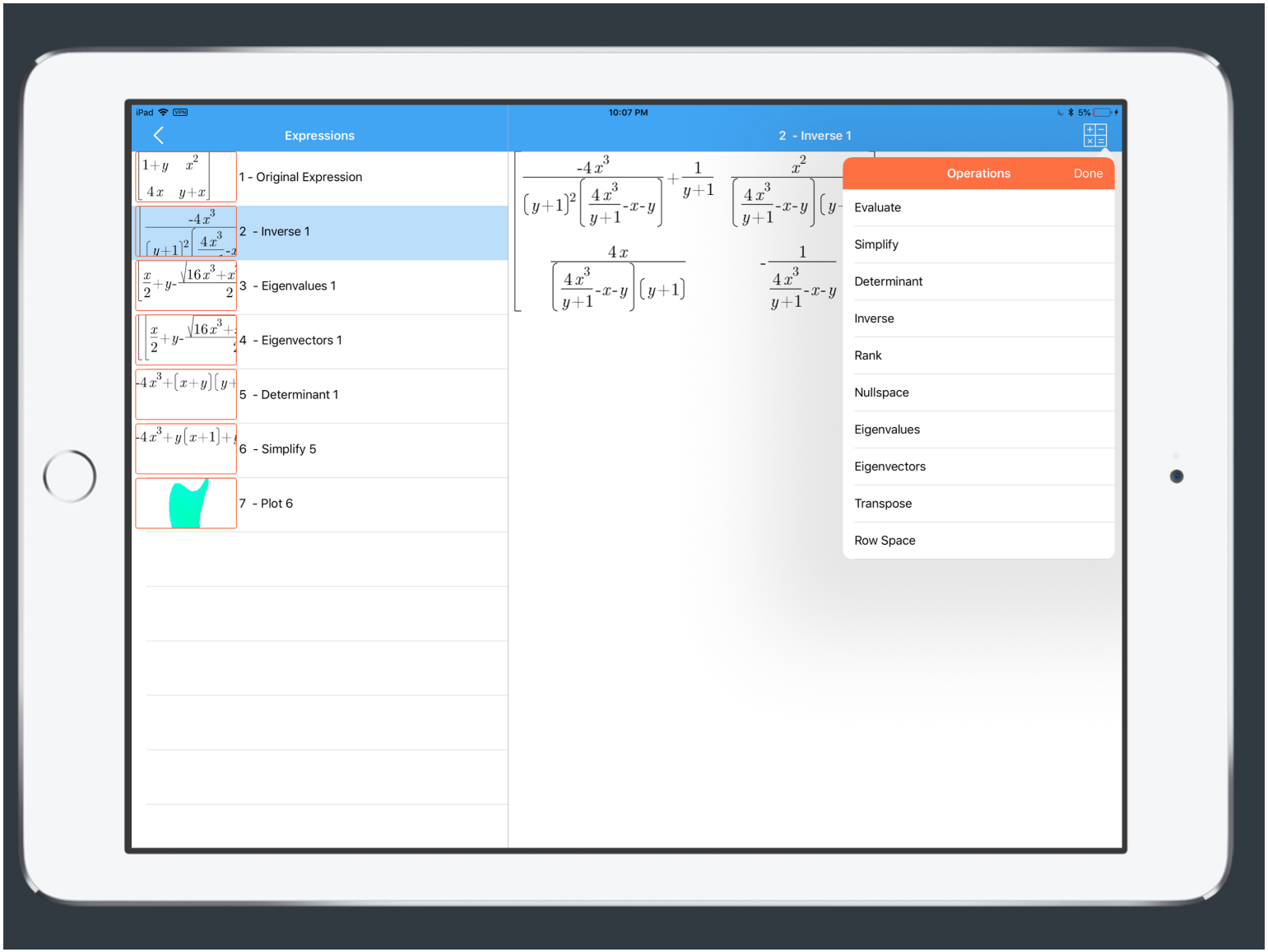
MathBrush App

Exporting LaTeX/MathML



MathBrush App

Manipulation with Computer Algebra System



Math Recognition: Why Hard

- Two-Dimensional
- No dictionary
- Ambiguity (structure and semantic)
- More involved gestures
- Recognizing partial expressions

Math Recognition: Process

Recognition Phases:

- Segmentation
- Symbol Recognition
- Structural Analysis

Recognition Approaches:

- Sequential
- Integrated
- End-to-end

MathBrush Recognizer

- Follows the integrated approach and depends on grammar for recognizing mathematical handwritten expressions
- Recognizes a wide range of mathematics
- Provides ranked alternatives for recognition
- Generates Latex and MathML for input math expressions
- Allows for symbols training
- "Won" 2012 CROHME competition for math recognition

Why Another Front-end?

- Recent recognition techniques that follow end-to-end approach promise better results
- Dependency on grammar limits the ability for extending supported mathematics recognition
- Need for supporting gestures in the interface and the recognizer
- Inability to recognize wildcard symbols
- Lacking a way to communicate wildcards and constraints to the search engine using an intuitive user interface

Integrating online mathematical recognition with a more expressive pen-based query language is one of the novel features of MathBrush-Search

Early approaches

- Text Search

  e.g. treat formulas as bags of words

- Exact Match

  e.g. match formulas only if exact

Approximate formula match approaches

- Normalized Match

$$e.g. \quad \int \frac{2+x}{x^2}dx \quad \text{matches with} \quad \int \frac{2+y}{y^2}dy$$

- Subexpression Match

$$e.g. \quad \cos(x) \quad \text{matches with} \quad \sin(x)$$

$$\int \frac{p(x)}{q(x)}dx \quad \text{matches with} \quad \int \frac{1}{\sqrt{x^4+3}}dx$$

Math-aware Search Engines

Combining Mathematics and Keywords

- Associate keywords with expressions near by and propagate association with formulas dependency
- Capture the semantics of mathematical formulas based on the sequences of words around them

Tangent-L Search Engine

- Based on Lucene framework
- Indexes both text and formulas syntactic features
- Performs comparably to state-of-the-art math retrieval systems

Symbol Layout Tree

- Nodes: represent symbols and visually explicit aggregates
- Edges: capture the spatial relationships between objects

Tuples

- Tuples represent paths in SLT
- Indices built over various paths
- To match, tuples of query and in the database are compared

Challenging ourselves: ARQ Math Competition

- ARQ Goal: Advance math-aware search and the semantic analysis of mathematical notation and text
- Data size: Archived posts from Math StackExchange (~1 million questions; ~28 million LaTeX formulas)
- Task: Given a posted question (in 2019) as a query, search answer posts (2010-2018) and return relevant answers
- Objective: Compare Tangent-L with other systems and compare different configurations when running Tangent-L

ARQ Math Competition (cont.)

ARQMath Task: Given a math question look among all answers and create an ordered list of possible answers to that question. For example,

Question Could anyone tell me how I can approach this problem?

$$\lim_{n\to\infty} n^{\frac{1}{n}} = 1$$

Relevant You can use AM $\geq$ GM:

$$\begin{aligned} \frac{1+1+\cdots+1+\sqrt{n}+\sqrt{n}}{n} &\geq n^{\frac{1}{n}} \geq 1 \\ 1-\frac{2}{n}+\frac{2}{\sqrt{n}} &\geq n^{\frac{1}{n}} \geq 1 \end{aligned}$$

Non-Relevant If you just want to show it converges, then the partial sums are increasing but the whole series is bounded above by

$$1+\int_1^{\infty} \frac{1}{x^2} = 2$$

Why Another Search Engine?

- Search engines that allow users to query documents using keywords and formulas together have not yet shown themselves to be effective
- Support for specifying syntactic and semantic preferences for approximate matches of mathematical expressions, including the use of expressive wildcards and constraints as well as rich semantics, is unique to this project
- Combine semantics from mathematical formulas and their accompanying text, including each document's context, to identify the most relevant documents has not been investigated

System Extensions

Wildcards Specifications

- Wildcards are not adapted in the mathematical context
- Formal definition of wildcards and constraints: data type, ranges, significance, and connecting multiple constraints

Constraints	Acceptable Matches
$x + y + c$ $x[numeric, min = 1, max = 100]$ $y[constant]$ $c[expression]$	$100 + y + 2$ $10 + y + (x - 1)$ $50 + y + x^2$
a $a[function, par = [y]]]$ $y[sequence, maxItems = 10, P = [[1, mc]]]$ $m[numeric, min = 1, max = 3]$ $c[variable]$	$f(2x, 4, 3)$ $f(3c, x, y, 10)$ $g(f)$

Future System Extensions

Math Recognition

Enhance the recognition by using end-to-end ML approaches instead of the current grammar-based

Rethinking Math Recognizer

- Integrate recent ML advancements from the domain of NLP
- Make better use of context to resolve ambiguities
- The Transformer architecture may help with both goals

Challenges

- How best to represent 2D structure?
- Converting handwritten math to token-based representation
- Small datasets compared to typical NLP problem
- MathBrush system can help with last problem

Future System Extensions

Search Engine

- Incorporating math transformation rules and query expansion to improve formula search
- Upgrading Tangent-L to interpret rich wildcard symbols and constraints and to relate formulas and sub-formulas to the semantic content obtained from the accompanying text and document context
- Weighing users' defined preferences to combine keyword and formula importance and rank the search results accordingly

Future System Extensions

Front-end and Search Engine

- User feedback can be incorporated in two ways:
  - Improve the recognition of user input based on user corrections
  - Adjust the model for computing a document's relevance based on user interactions with search results
- User studies: conduct user studies for different options of the query interface and system interactions
- Wildcards: adapt and support the proposed wildcards specifications in the user interface, the recognition, and the search engine

Conclusion

- Finish v1.0 integrating MathBrush recognizer and Tangent-L search engine
- Working on v1.5:
  - Evaluate the Tangent-L performance and compare different configurations
  - Start investigation of the support for simple wildcards
  - Experimenting with different ML recognition approaches
- Future versions for supporting other future extensions (gestures, implementation, better use of semantics).

Conclusion

- Finish v1.0 integrating MathBrush recognizer and Tangent-L search engine
- Working on v1.5:
  - Evaluate the Tangent-L performance and compare different configurations
  - Start investigation of the support for simple wildcards
  - Experimenting with different ML recognition approaches
- Future versions for supporting other future extensions (gestures, implementation, better use of semantics).

THANK YOU!