Expresso: Typesetting Handwritten Mathematical Expressions on the Post-PC Tablet Computer

[Project Proposal]

Josef Lange
University of Puget Sound
1500 N. Warner St.
Tacoma, WA 98416
jlange@pugetsound.edu

ABSTRACT

This paper sets forth to describe my project fulfilling the capstone requirement of the Bachelor's Degree in Computer Science at the University of Puget Sound. My project is to research, design, and implement a tablet-based software solution for the capture of handwritten mathematical expressions and compilation of its mathematical representation in an appropriate representative language (such as LaTeX or MathML).

The objective of my project is considerably challenging for an undergraduate Computer Science student to undertake. This proposal explains how the process will be segmented incrementally in several phases of research, design, and implementation so as to produce tangible results regardless of how rigorous the subject matter becomes.

Subjects tackled in my project include image processing, artificial intelligence, application development for the Apple iOS platform (particularly in the tablet form factor), as well as the adherence to a highly disciplined development cycle. I will attempt to most accurately and most briefly explain any uncommon concepts described hereafter.

1. INTRODUCTION

The utility of tablet has only been recently fully realized, with the nascence of the "Post-PC" tablet, migrating away from the standard desktop usage paradigm commonly associated with computing. The modern tablet focuses on media consumption and so-called "basic" computing. In today's reality, most "Post-PC" tablets are incredibly powerful both in hardware and in software, supporting complex and challenging computations including the decoding of video, image processing, interpretation of multiple inputs, managing several network connections, and displaying high-resolution two- and three-dimensional graphics.

My project aims to utilize this now-popular device class for the capture and conversion of handwritten mathematics to relevant language to typesetting systems. Predominantly, this functionality has been researched and implemented on desktop systems, due in part to the prior lack of power in tablet devices. With modern tablets, the pieces line up: the devices already have built-in digitizers and are now powered with processors comparable in processing power to modern desktop and notebook systems.

2. EXISTING SOLUTIONS

2.1 General Handwriting Recognition

Handwriting recognition is a complex problem, only a segment of which my project wishes to solve. Existing solutions to handwriting recognition exist, and some perform at a useable level. Frequently, these systems require a period of "training" before any recognition can be achieved. Others use artificial intelligence to conclude what a sequence of handwriting is meant to be. For full-on text recognition, this is useful and almost required for any success. Commercial products exist, including but not limited to software built into Microsoft's Windows and Apple's Mac OS X Operating Systems. Additionally, many third-party companies have produced software for a similar purpose.

2.2 Mathematical Expression Capture

General handwriting recognition solutions are frequently overpowered for the niche of mathematical expressions, in which there are fewer possible logical constructs, most of which follow a fairly standard form. Because of this, the subset of things needing recognition, and possibilities for what they could be recognized as, is significantly smaller. Academic projects, including those by Matsakis[2], Smithies et. al.[3], and Levin[1] exhibit valid solutions for the desktop model of usage, though are not implemented in a way that is ultimately accessible for the common user.

3. PROPOSED SOLUTION

My project will, if successfully carried out, produce a feature-complete, useable piece of software for the Apple iOS platform. This software will have a singular function: to capture handwritten mathematical expressions via the digitizing screen, and compile the interpreted characters into the appropriate mathematical representational language (namely LaTeX or MathML).

There are numerous components to such a system: capturing the digitized drawing done by the user, separating individual glyphs, identifying said glyphs, comparing the size and location of said glyphs, deducing an intended mathematical representation, and compiling that representation into useable code. In addition to working software, a strong focus on useful and good-looking user interface will be applied.

The proposed implementation has several optional system features (not apparent to the end user), including but not yet limited to distributed learning-based artificial intelligence, which if successfully implemented, could stand to improve accuracy of the software over time. These optional features will be researched and implemented if time and skill allows.

4. MOTIVATION & IMPORTANCE

The "Post-PC" era is no longer a myth. Commercial sales and institutional interest in keyboard-mouse-screen computing systems is decreasing while interest in tablet devices continues to grow. Implementing useful and elegant solutions– which take advantage of this radically different device's usage paradigm–to significant computer science problems is of utmost importance both for the sake of the field and for computing itself.

Tablet computing is quickly becoming the primary form of computing for many users, at home, at work, and at school. Shouldn't the devices be more useful than the systems they've replaced?

In addition to this project's aforementioned idealistic importance, a practical importance exists: previous solutions to this problem have surely been successful but not widely available or useful for the common user or educational institution. The shortcoming of prior solutions, in one way, is the availability and ubiquity of handwritten input.

On a desktop or even a notebook system, a graphics tablet is nearly a required resource for any such work. On a modern tablet device, similar algorithms can be implemented utilizing the built-in digitizer to create a solution with little to no additional equipment necessary.

With the growing ubiquity of the Post-PC tablet (particularly the Apple iPad) in education, a solution such as this can really make an impact in bringing the classroom fully into the digital age.

5. PROJECT ROADMAP

5.1 Research

First and foremost, additional heavy research is required. Each small piece of information leads to additional sources of information and additional projects that have attempted to achieve (with varying levels of success) the same goal. There is much to be learned from each general field from which this project draws. It will be helpful to carefully analyze previous projects of this nature to find commonly-used methodologies when approaching this context. Research may also help to steer this project from any abysses of hard work. The best implementation is frequently the simplest one, and as such this project aims to avoid overly-complex solutions.

5.2 Design

A great deal of design must be done on such a big and complex project. The first task is to intelligently segment the project into smaller, possible-to-implement pieces that can be connected to create a whole system. This will require a strong knowledge of design patterns and general software engineering concepts. It is this project's intention to be comprised of several software components that may be useful on their own, and when combined properly, efficiently and elegantly solve the intended problem.

5.3 Implementation

This project, as mentioned before, will need to be segmented, with each segment well-designed, in order for implementation to be manageable. The will be implemented on the Apple iOS platform using the Cocoa Touch framework. Cocoa Touch is developed primarily in Objective-C, which is compatible at compilation with C and C++. The intention is to compile and distribute for the iPad device, though the iPhone and iPod touch devices are equally capable, albeit slightly more difficult to use with such a small screen and digitizer. Apple provides a Development Program for institutions of higher education, which will negate any costs of development that Apple normally charges (namely the annual $99 membership) for developers to be able to compile their code to devices for testing and usage.

6. REFERENCES

[1] M. Levin. *CellWriter: Grid-Entry Handwriting Recognition*. PhD thesis, University of Minnesota, Minneapolis, MN, USA, 2007.

[2] N. E. Matsakis. *Recognition of Handwritten Mathematical Expressions*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

[3] S. Smithies, K. Novins, and J. Arvo. A handwriting-based equation editor. Technical report, Department of Computer Science, University of Otago, Dunedin, New Zealand, 1999.