Proposal for a Bachelor of Science Thesis: Interactive on-line handwriting recognition of mathematical formulae

Martin Thoma

January 14, 2014

1 The problem background

There are people who don't know how to write even simple mathematical formulae with LaTeX like

$$\pi/\alpha = \sum_{n=-\infty}^{\infty} \frac{\sin^2(c+n)\alpha}{(c+n)^2} = \int_{-\infty}^{\infty} \frac{\sin^2(c+n)\alpha}{(c+n)^2}\,\mathrm{d}n$$

or who need much time to do so. Currently, there are several online services, programs and apps that help to write mathematical formulae, but all programs I know have serious disadvantages:

- detexify.kirelabs.org recognizes **only symbols**,
- the formula editor of LibreOffice Writer 3.6 as shown in Figure 1 on the following page offers some guidance by grouping common operations while showing a WYSIWYG editor, but it has **no handwriting recognition**. Another drawback is the fact that it is **not available as an online service**, so you have to install LibreOffice which might not be possible on all devices.
- The "Daum Equation Editor" (see Figure 2 on the next page) is available online and offers guidance through the creation of equations, but does not offer handwriting recognition. Although it might be OpenSource, the **source code is difficult to find**. This means if you want to improve the recognition, it is not possible. It also makes use of Adobe Flash which is not available on many smartphones and tablet computers.
- Maple seems to offer handwritten symbol recognition (source), but on the one hand I was not able to test that, because it is **not available for free**. On the other hand you have to install additional software, it seems not to be available for tablet computers and it does only recognize single symbols.
- Wolfram Mathematica seems to be able to do complete formula recognition at least for simple formulae (source) by using Microsoft's Math Input Panel, but this is neither OpenSource nor available as an online service. Additionally it is not available for Linux systems, so I can't test it.

A more comprehensive list can be found at https://en.wikipedia.org/wiki/Formula_editor. A problem of some of the projects presented there is that they require the client to execute Java Applets which is a security risk.

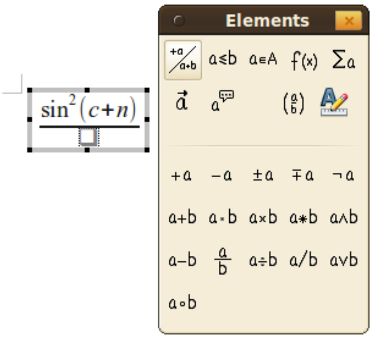
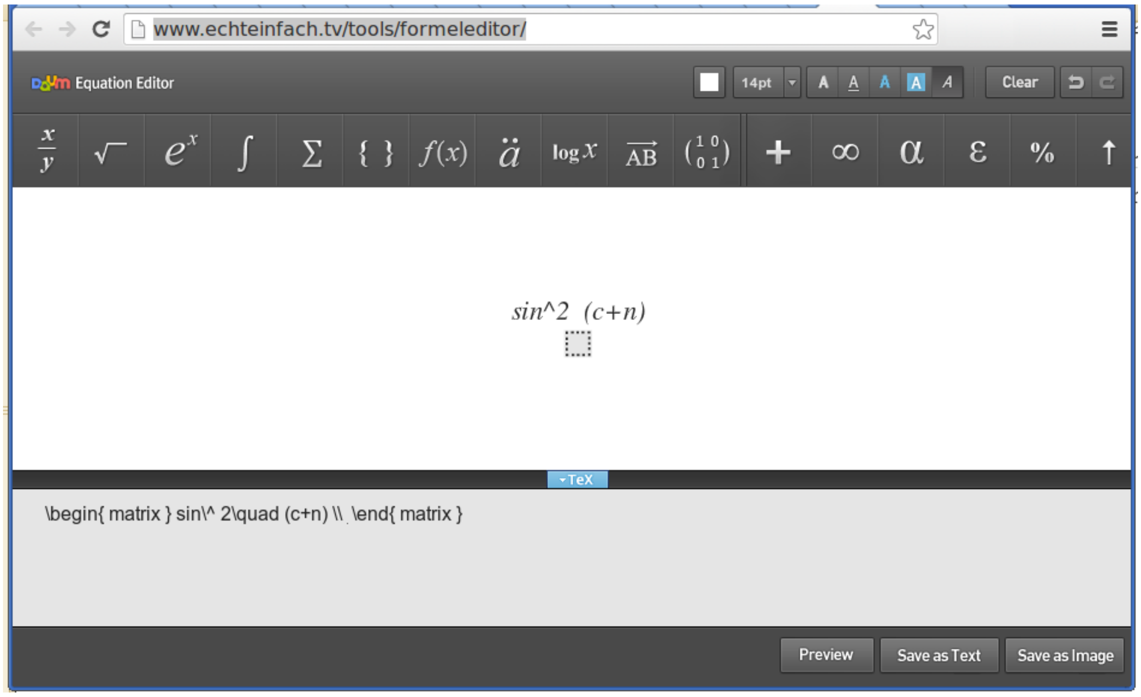


Figure 1: LibreOffice Writer 3.6 - Formula Editor



```
\begin{ matrix } sin\^ 2\quad (c+n) \\ \end{ matrix }
```

Figure 2: Daum Equation editor

2 The problem statement

What I would like to have is an interactive on-line handwriting recognition service, that is available as a web service which makes use of touchscreens. Additionally, it should be for free and OpenSource, the source code should be easy to find and documented. This means:

- **Service**: The program can be accessed over the web, so that the user does only have to have a modern browser. As a consequence, the software could be used with any device that has a touch screen.
- **On-line handwriting recognition**: The service starts recognizing while the user enters a formula.
- **Interactive**: The service offers symbols and constructs to the user before the user starts typing. These suggestions might change depending on what the user has typed before.
- **OpenSource**: Any license in this list: http://opensource.org/licenses

- **Easy to find**: Ideally, the project should have an own domain that contains the source code, the service and documentation. But it might be enough to provide an email address to a developer within the top of of the source code of the delivered HTML document.

This service should also encourage the users by techniques of "Gamification" to give as much meta information about their formulae as possible:

- Which problem domain does the formula belong to, e. g. "Euclidean geometry", "analysis" or "calculus"?
- Does the formula itself have a name, e. g. "Pythagorean theorem", "Fibonacci numbers" or "geometric series"?

This information should be used to create a formula database.

3 Significance

For me as a Linux user, there no software that I can test and which offers on-line, interactive math handwriting recognition. But the need of such a software is there.

But there are more reasons why this bachelor's thesis matters: Projects like LaTeX, Linux, Apache or Firefox have shown that OpenSource software can enrich the development in specific areas. The "Browser Wars" might be the most famous result of an active OpenSource community. Internet Explorer 6 had a market share of over 80% in 2003. Prequels of Firefox and the Mozilla foundation already existed, but Firefox 1.0 was released not until November 2004. After that, Firefox and other open browsers added many features that Internet Explorer had to compete with, like tabbed browsing, HTML4 standard conformance, support of the `<canvas>` tag and speed of HTML rendering and JavaScript execution.[1] Some of these questions are interesting for science such as many problems related to layouts and just-in-time compilation (JIT). With OpenSource software that makes it easy to find its source and offers good documentation, researchers can simply try their ideas without being blocked by having to try to access the source code.

Additionally, such a project might give researchers more time to concentrate on the tasks they really want to do rather than spending hours by learning LaTeX.

One last reason why this thesis matters is the formula database that gets created by users. This database might be used in follow-up work, e. g. a formula spotter for presentations or a math detector for speech.

4 Time schedule

70h Literature research about on-line handwriting recognition techniques and Gamification.

5h Defining browsers and devices that should get supported and required client side software like HTML5, CSS 3 and ECMAScript (better known as JavaScript). Also, required input methods like touchscreens and stylus should be mentioned.

20h Writing use cases. This is includes writing example formula that the user should type and the system should be able to recognize; finding people with different knowledge of LaTeX and from different fields who want to participate in user tests.

[1] www.evolutionoftheweb.com offers a graphical overview. Although supporting standards like HTML4 or CSS 2 is not done with one version, but rather an incremental process.

60h Implementing the core of the application: Handwriting recognition of digits and symbols by using only HTML, CSS and on the client side. This includes implementing a way for the user to enter new symbols and to correct the symbol that was suggested by the recognition system.

20h Introduce testers that already know LaTeX to the current system. At this point, the system does only do symbol recognition. The testers should train it, insert symbols like $a - z, A - Z, 0 - 9, \alpha - \omega, A - \Omega, \cdot, \circ, \dots$

10h Get feedback by the users. This feedback will not be included in the thesis, but the improvements will get documented.

60h Finding structures and ways how to enter them. Examples of structures that can be nested are sums:

```
\sum_{<some structure>}^{<another structure>} <a third structure>
```

Implement the recognition of those structures.

30h Observe “fresh” testers while they try to use the system.

70h Improving the software to fix problems that were found with user tests

50h Fix bugs, improve code quality and readability as well as documentation.

45h Usability testing: Try Hallway testing. The results of these tests get documented and will be part of the bachelor’s thesis. If possible, I would like to let the testers use their own devices.

10h Mentioning open questions and ideas how they could be analyzed with the service that was created.

5 Outline

I have described in which steps I would like to write the software, but almost all points include writing the bachelor’s thesis document. A first draft of the outline could be like this:

1. Introduction
2. Definitions
   a) Hardware: What is available and what is the distribution?
   b) Software: What is available and what is the distribution?
   c) Support of standards like HTML, CSS, ECMA-Script, Flash, Cookies, ...
   d) Choice of hardware, software and standards that should get supported as well as the choice of Libraries and the required server-side software
   e) Application to the domain of math recognition
3. On-line handwriting techniques
   a) Description of techniques in general
   b) Application to the domain of math recognition
4. Gamification techniques
   a) Description of techniques in general
   b) Application to the domain of math recognition in the web

5. Software Project
    a) Structure of the code
    b) Availability of documentation
    c) Availability of the service
6. Summary
    a) Future Work

Related Literature

[conf09] *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009.* IEEE Computer Society, 2009.

[Dete11] Sebastian Deterding. Meaningful Play - Getting "Gamificaiton" Right .`http://www.slideshare.net/dings/meaningful-play-getting-gamification` 2011. [Online; accessed 10-January-2014].

[Impe94] Sebastiano Impedovo. *Fundamentals in handwriting recognition.* Springer-Verlag. 1994.

[StTR09] Andreas Stoffel, Ernesto Tapia und Raúl Rojas. Recognition of On-Line Handwritten Commutative Diagrams. In *ICDAR* [conf09], S. 1211–1215.

[Zich11] Gabe Zichermann. *Gamification by Design.* O'Reilly Media. 2011.

This literature list is only a list that seems to make sense to me by now. As I proceed I might find more useful sources for the different topics. So I might add, but also remove elements from this list. Especially for Gamification I might read documents from gamification-research.org.