The University of Birmingham
School of Computer Science

Possible Projects to be Supervised by Alan Sexton

1 Introduction

This year I will be away for large parts of the Summer and will therefore not be taking on any MSc project students for supervision over the Summer. I will, however, be available to supervise final year undergraduate projects.

I am happy to supervise projects to develop applications that are command line, desktop GUI, web apps or mobile phone/tablet apps (I prefer Android to IOS or Windows Phone).

For programming languages I prefer OCaml or Java (in both of which I can give considerable support), although I will consider other languages as well.

The rest of this document lists various areas of interest to me and project titles in those areas that I am interested in supervising.

2 Document Analysis

Document analysis is the extraction of information from document representations (usually images). This can be offline (i.e. from scanned or photographed images) or online (i.e. from a smart phone or tablet), and can be from formatted structures (e.g. PDF), raw bitmap images or digital ink files (c.f. `http://www.w3.org/TR/InkML`).

Some of the possible projects in this area require specialised knowledge or interests or techniques, e.g. machine learning, pattern recognition, neural networks, image analysis, Android programming, Music reading, etc. I can give assistance in all these areas but please ask if you have any concerns.

The following is a non-exclusive list of possible project titles in this area:

- Optical character recognition
- Layout analysis (analysing pages to identify which parts are columns, diagrams, tables, headings, etc.)
- Handwriting analysis
- Mathematical formula recognition
- Diagram recognition (e.g. UML class diagrams, Entity Relationship diagrams, Plots etc.)
- Table recognition
- Music score recognition (this can optionally include midi output as well)
- Whiteboard recognition
- Form recognition (e.g. questionnaire forms)
- Ground truth extraction and labelling (automated assistance for extracting clips from images and labelling them by their names for use in training recognisers)
- Transcription support for historical documents (recognition of historical documents is generally beyond the current state of the art, but diagram analysis technology can greatly assist in supporting manual transcription of such documents)

For projects I am involved in in this area, please see `http://www.cs.bham.ac.uk/~aps/research/projects/as` and `http://www.cs.bham.ac.uk/~aps/research/projects/neumes`.

3 Mathematical Knowledge Management (MKM)

With better support for Mathematical Knowledge Management (MKM) and Mathematical Formula Recognition from the area of Document Analysis, we might eventually be able to clip an image of a formula from a scanned book and paste it into a mathematical tool such as Maple, Mathematica or Matlab, so that we can then manipulate it (e.g. integrate or differentiate it, find solutions or simply graph it). Further we could have a database of mathematical formulae and search through them in an intelligent manner, i.e. by formula rather then just by keywords (c.f. `http://dlmf.nist.gov/`). While the DLMF is a magnificent resource, it only provides a limited amount of mathematical knowledge. The true treasure trove of mathematical knowledge is in the published literature. High quality searching over that is a huge target (c.f. `https://eudml.org`, `http://search.mathweb.org`). The holy grail of Mathematical Knowledge Management is not just to be able to find mathematics, however, but to be able to ask mathematically meaningful questions and, using the repository of mathematical knowledge, to get intelligent and useful answers.

The standard solution for representing mathematical formulae in MKM, especially on the Web, is an XML language called MathML (c.f. `http://www.w3.org/Math`). MathML comes in two flavours: Presentation MathML, which tries to capture how a mathematical expression should look, and Content MathML, which tries to capture what it means. Tools to support MathML are a valuable component in MKM.

Blind and partially sighted people are almost excluded from the sciences because of the difficulty of reading mathematics. With suitable tools based on MathML (screen readers, MathML navigators etc.) we can open the door for the blind to the mathematical literature.

Translating from Content MathML to Presentation is a bit tricky but essentially possible, whereas fully automatic reverse translation is infeasible in the general case. However tools to support human aided versions of the latter are very possible and can help considerably.

Projects in this area include

- MathML translators (between Presentation MathML, Content MathML and Latex)
- MathML editors
- MathML search tools
- Mathematical Formula Recognition
- Mathematical Formula Entry on Tablets/Smart Phones

4 Compilers

I would like to supervise projects to develop a compiler for a simple language. The subject language choice is up to the student. Target processor could be ARM (for a Raspberry PI or an android smart phone) or MIPS (there are high quality MIPs simulators available such as SPIM). The implementation language should be OCaml (as it is much faster to develop this type of code in OCaml than in Java).

Related projects include

- an IDE for OCaml
- a Parser Generator for OCaml
- a Lexer Generator for OCaml
- Compiler modules (Abstract Syntax Tree tools, Optimisers, Code Generators, Garbage Collectors, Multicore support etc.)

5 Cycle Touring

I have an interest in long distance bicycle touring. With the free availability of OpenStreetMaps data `http://www.openstreetmap.org` and web service access to Google Maps, a number of systems have become available to assist with route finding/planning and with navigation (e.g. `http://bikehike.co.uk`, `http://www.mapmyride.com`, `http://www.cyclestreets.net`). Most are geared towards time trialling and racing rather than long distance touring. There is scope here for a number of projects to support the kind of flexibility and usability necessary for touring. For example:

- Desktop or Web applications for choosing routes, with TCX or GPX export options so that the route can be downloaded to Sat Nav devices.
- Adaptive Android apps to manage route changes during a tour — especially overriding OpenStreetMap data when it is wrong and recording routes to correct OpenStreetMap with after returning from a ride.
- Recording and (blog style) displaying of GPS ride data following a ride with support for adding photos taken, providing overviews of the tour and controlling detail levels viewable by different users.

6 Parallelism, Multicore Computing and Distribution

I am open to supervising projects in:

- General purpose parallel programming on NVidia graphics processing units
- Shared memory parallel programming on multicore CPUs
- Distributed memory programming on distributed networks of computers

In all these cases I am interested in exploring algorithms and systems over a range of application domains, for example image processing, fault tolerant computing, high performance computing, system utilities, programming language features and frameworks etc.

A particular sub-topic in this area I am interested in is projects using the *Erlang* programming language. *Erlang* is programming language designed for highly parallel, distributed and scalable systems (`http://www.erlang.org/`. Particular projects in Erlang include:

- Image processing and image processing libraries in Erlang
- Web application/service development in Erlang
- GUI frameworks and libraries for Erlang