400-Level Project Proposals 2014

A list of possible 400-level projects is given below, roughly organised by supervisor's name. You do not have to restrict yourself to this list, and can make up your own project topic. However, to do so you will have to arrange for a supervisor. You can find more information about the research interests of the staff members on the Computer Science Department Web pages.

You should email your project selections to the project co-ordinator, Steven Mills steven@cs.otago.ac.nz by **Friday 28th February**. Before making these selections you should talk to the supervisors of the projects you are interested in. In most cases the best way is to email them to make a time to do so – supervisors' email addresses are given with each project.

In order to allocate the projects as fairly as possible we ask that you give first, second, and third choice projects, and that you choose projects from three different supervisors. If you find multiple projects that look equally appealing, then you can give lists of projects as first, second, and third choices. You will still need to include at least three different supervisors, and at least two in your first two lists.

There are two different project papers - COSC480 and COSC490. The relevant Web pages (for COSC480 and COSC490 have more details, but the short version is that if you are enrolled in the Computer Science Honours degree you take COSC490, otherwise you take COSC480. Most projects are suitable for either COSC480 or COSC490, but a few might be only suitable for one or the other – this is indicated in each project description.

1. **Source Code Editor**

   Andrew Trotman andrew@cs.otago.ac.nz,
   David Eyers dme@cs.otago.ac.nz,
   Richard O'Keefe ok@cs.otago.ac.nz, 480/490

   Why oh why do we continue to write code in a single-sized single-font text editor when we have rich display capabilities on our desktops. Why cant I put a graphic image, or a URL, of a PDF into a comment in my C++ code? Why are my classes in separate files rather than separate workspaces? In this project we will challenge the current norm in source code editors by building a next generation text editor that allows us to alter font face, colour, size, to highlight lines of code, to attach images and PDFs to comments, to float comments in our code (rather than embedding them) and to navigate through our code in two dimensions (rather than just scrolling).

2. **Search Engine Pseudo Relevance Feedback**

   Andrew Trotman andrew@cs.otago.ac.nz,
   Richard O'Keefe ok@cs.otago.ac.nz,
   Matt Crane mcrane@cs.otago.ac.nz, 480/490

   One technique used to increase the quality of the results of a search engine is pseudo relevance feedback. Once the search engine has an answer set, analysis of that answer set returns a better set of search terms than the original query. Using this new set rather than the original query improved quality. Now, how many document should we analyze and how many terms should we use in the new query?

3. **Search Engine Query Performance Prediction**

   Andrew Trotman andrew@cs.otago.ac.nz,
   Richard O'Keefe ok@cs.otago.ac.nz, 480/490

   Search engines use a number of techniques to improve quality. One is pseudo relevance feedback, another is stemming. But the result of using these techniques is not always an improvement, sometime result quality is degraded – but when? The field of query performance prediction has, in the past, derived several scores from the results list and tried to correlate these with quality – unsuccessfully. In this project we will take a different approach: given two results lists for the same query, which is better? This alternative approach is a binary classifier and consequently we can expect better results than correlation scores.

4. **Quantifying Conceptual Density in Text**

   Anthony Robins anthony@cs.otago.ac.nz,
   Alistair Knott alik@cs.otago.ac.nz, 480/490

   Can we identify the density of connections between individual concepts in a body of text? In other words, can we identify discrete concepts, and quantify the relationships between them? This project will involve investigating existing natural language processing tools, and perhaps developing our own.

5. **Attractor Spaces in Hopfield Nets**

   Anthony Robins anthony@cs.otago.ac.nz480/490

   Hopfield networks (a kind of neural network) have dynamic behaviour that can be characterised in terms of gradient descent in a multidimensional attractor space. We know a fair bit about the structure of such spaces. This project will involve implementing and exploring a Hopfield type network to further develop our understanding of its dynamic behaviour.

6. **Serial Learning in Neural Networks**

   Anthony Robins anthony@cs.otago.ac.nz, 480/490

   Neural networks are very powerful learning systems. But most of them work best when all training data is available simultaneously (concurrent learning). In the real world humans and other animals learn different things on different occasions (serial learning). This project will review the current state of serial learning in neural networks, and further explore my "pseudorehearsal" solution.

7. **Simple Robot vision**

   Anthony Robins anthony@cs.otago.ac.nz, 480/490

   The department has many Lego Mindstorms robots. One of the sensors that we have is a simple camera capable of returning the coordinates of blocks of specified colours in the visual field. How well does it work? In particular, can we determine location on the basis of the arrangement of colours in the environment? This project may involve developing a code library/toolbox for using the camera.

8. **Learning Visually Guided Grasp Actions with a Robot Arm**

   Brendan McCane mccane@cs.otago.ac.nz,
   Steven Mills steven@cs.otago.ac.nz,
   Alistair Knott alik@cs.otago.ac.nz, 480/490

   The aim of this project is to design an algorithm allowing the robot to learn how to reach and grasp a target object within its field of view.

Learning will happen by reinforcement, through the 'joy of touch' principle. (The basic idea: any activation of the touch sensors functions as a reward; to maximise reward, the robot makes those movements which result in touch sensations.) The robot will learn a function mapping the visual representation of a target object onto a goal state of its hand/arm, and will then generate movements to bring its hand/arm into this goal state. The project involves a combination of machine vision, machine learning and motor control theory. There is a lot involved, and the robot is a new piece of kit - part of the project's content is to decide on sensible simplifications as we gain experience with the robot

9. **A Compiler for Lego Robots**

   Brendan McCane mccane@cs.otago.ac.nz, 480/490

   Current compilers for lego robots are either not free or not cross-platform (no native compiler for OSX). This project will develop an open source compiler for NQC (or a similar language) for the Lego robot platform.

10. **An n-gram Writer's Assistant**

    Brendan McCane mccane@cs.otago.ac.nz, 480/490

    The goal of this project is to develop a writing assistant that uses an n-gram corpus in addition to very simple grammar rules to help a writer construct a sentence. The target audience is for those with English as a second language. The main idea is that this assistant will be constructive – that is it will provide guidance on constructing a sentence as the sentence is being constructed, rather than after the fact as with many current grammar checkers.

11. **QR codes for security**

    David Eyers dme@cs.otago.ac.nz, 480/490

    QR codes are a class of 2D barcodes of various sizes and error correction capabilities that have a public, extensive and extensible schema for data that can be encoded in them. Open source software is available to generate and detect QR codes. This project focuses on the development of software that uses QR codes for computer security functions. No secure connection between two computing devices can be formed without a shared secret such as a SSH key. This project will investigate the use of QR codes as an independent channel to verify such secrets.

12. **Peer-to-peer Mega**

    David Eyers dme@cs.otago.ac.nz, 480/490

    Services like Dropbox and Mega have popularised cloud storage. Users can synchronise files between multiple devices conveniently. However, these services track content and its propagation centrally. Peer-to-peer systems such as BitTorrent do not rely on a central server to manage where content is stored, but cannot provide guarantees about the availability of content. This project aims to develop a tool that integrates the best features from both peer-to-peer and centralised network storage systems.

13. **CSS-based Security for Web Browsers**

    David Eyers dme@cs.otago.ac.nz, 480/490

    Many of today's web security breaches, such as cross-site scripting (XSS) attacks, rely on being able to inject JavaScript code into the webpage that a client views. This type of problem is usually approached from the server-side, but can also be fixed on the client-side, though, and that is the focus of this project. A transformation script, proxy and/or browser plug-in will be written that will uses information about the formatting (CSS) and structure (DOM) of the webpage being rendered, combined with a policy file, to permit or deny the execution of script tags at different points within page rendering.

14. **Web Pipes**

    David Eyers dme@cs.otago.ac.nz, 480/490

    Pipes are frequently used within shell scripting on Unix-like operating systems (pipes are also supported by many Windows kernels). They are a simple, yet highly effective inter-process communication paradigm, suitable for many common tasks. This project integrates named pipes into web browsers. This goes beyond using HTTP(S) for the data transport of a pipe: wget and friends can already do this. The browser integration is to allow pipe creation, configuration, and local and remote pipe linking to be done with user involvement, and with the involvement of the browser's JavaScript engine.

15. **Bringing Command Shells and Terminals Closer Together**

    David Eyers dme@cs.otago.ac.nz,
    Richard O'Keefe ok@cs.otago.ac.nz 480/490

    Apart from being able to edit the current command, most functions of the command shells that run within today's terminal emulators assume little more functionality than what is provided by a Teletype. That is, output from commands scrolls off the top of the terminal emulator (and possibly into scroll-back region, if active).

Enriching the connection between shells and terminal emulators can unlock so much more functionality. For example, why not track the difference between command entry, program input and program output? This would allow for folding and unfolding of the output of commands. Why not provide back-scroll regions for each command independently showing, say, an updating 4-line tail of multiple commands that are all running simultaneously.

By independently recording the output of each previous command, future text processing could be done directly from this recorded output, navigating to it using the enriched shell, and avoiding the need to actually re-run the command that produced the output.

16. **Proving the Correct Behaviour for an Online Ballot Box System**

    David Eyers dme@cs.otago.ac.nz, 480/490

    Many years ago, I quickly hacked together some electronic ballot box software for my college student association. The Basic Online Ballotbox (BOB) has now been adopted by a number of other organisations. The ballot system is explicitly designed so that software bugs cannot cause undetectable manipulation of election results, and it has had fairly thorough code review over the years. Even so, this design and review process has not systematically demonstrated BOB's correctness. This project will take advantage of the small size and simple workflow of the BOB codebase to apply existing static analysis tools to prove the correctness of its implementation, or to generate provably correct subsets of its code.

17. **Speeding up LaTeX**

    David Eyers dme@cs.otago.ac.nz, 480/490

    LaTeX is an open source typesetting package used by many (particularly academic researchers) due to the high quality of its output – particularly mathematical notation, its clean separation of document content from formatting, its vast extension library, and its plain-text source format. This project aims to apply analysis of how LaTeX runs, to speed up its operation for particular use cases, for example the generation of PowerPoint-like presentations. The pipe-dream is incremental LaTeX recompilation.

18. **A Software Gateway for Communication between WiFi and Sensor Devices**

    Haibo Zhang haibo@cs.otago.ac.nz, 480/490

    Even though WiFi devices and sensor devices use the same 2.4 GHz ISM band, they cannot communicate directly. This project aims to

design and implement a software gateway for protocol conversion between IEEE 802.11 (WiFi) and IEEE 802.15.4 (Sensor devices). A prototype will be developed for wireless fire alarming: sensor devices that detect fire can send an alarming message through WiFi networks to a smartphone.

19. **Learning Human Contact Patterns from Bluetooth Data Traces**

    Haibo Zhang haibo@cs.otago.ac.nz, 480/490

    Smart phones are becoming popular. Bluetooth technology allows two smart phones near each other to exchange messages independently. This project aims to learn the human contact patterns by analyzing the data communication traces collected from a set of Bluetooth devices. These patterns would be very useful to understand human activities and design efficient communication protocols for smart phone networks.

20. **Implementing a TDMA MAC Protocol for Wireless Sensor Networks**

    Haibo Zhang haibo@cs.otago.ac.nz, 480

    Since many industrial applications have stringent requirements on network delay and reliability, designing an efficient MAC protocol for reliable and real-time data transmission in wireless sensor networks (WSNs) still remains a challenging issue. As TDMA MAC protocols have the advantages of providing collision-free data transmission, the goal of this project is to develop a TDMA MAC protocol for WSNs. In this project, you will learn how to program with Contiki, the most popular operation system for WSNs, and have the chance to play with sensor hardware.

21. **Routing Module for Simulation of WirelessHART Networks**

    Haibo Zhang haibo@cs.otago.ac.nz, 480

    In a previous project, we have developed a simulator using Java and Matlab for simulation of WirelessHART networks. This project involves adding a routing module to the simulator. In this module, a set of routing protocols such as source routing, graph routing and delay-guaranteed routing will be implemented.

22. **How to Play Well**

    Michael Albert malbert@cs.otago.ac.nz, 480/490

    Trick taking card games, most notably bridge, are one area where the development of strong AI agents has been relatively limited. Playing these games with the cards face up (i.e. visible to all) is usually a completely trivial matter. Some agents have tried to use this as an

ingredient in search. Another approach is *Monte Carlo tree search* where an evaluation function is developed by using random play from the current and following positions. The aim of this project is to see whether combining these ideas might be productive. Euchre and Oh Hell are two candidate games.

23. **Semantic Relationships in Source Code**

    Richard O'Keefe ok@cs.otago.ac.nz, 480/490

    Some programming languages, notably C#, Java, and Smalltalk, have a standard way to attach "annotations" to code. This is usually used to provide the compiler and run-time system with more information like "this reference should not be null", "this is a test case", or "this is part of an Object-Relational Mapping". The nearest we tend to get to something for the programmer's edification is "don't use this any more".

    But chunks of source code are related to each other and to external documents in ways that the compiler and run time system do not need to know about, while maintenance programmers would love to know. The idea is to take an existing body of code that you are not familiar with, read it, and note what kinds of links are either informally specified or entirely missing but important to you, so that we can build up this catalogue. You should also review what has been done in Java and C# and Smalltalk. For more work, you might construct an actual set of annotation definitions that Java (or C# or Smalltalk) programmers might use and provide some tools (perhaps by extending an existing IDE) to exploit them.

24. **Processing patents**

    Richard O'Keefe ok@cs.otago.ac.nz, 480/490

    We have a copy of a collection of 100,000 US patents. Perhaps the most striking thing about patents is how weird the language is. There are some ideas about how we could measure it: few occurrences of 'I' and 'you', long complex sentences, more nominalisations. There are some ideas that haven't been tried yet, like sentiment analysis: do they use terms of approval or disapproval more or less than ordinary text?

    We have about 10,000 documents in English from the European parliament, several years' worth of IEEE articles in English, and a snapshot of the Wikipedia to compare the patents with.

25. **Improved Distributed Mosaicing**

Steven Mills steven@cs.otago.ac.nz,
David Eyers dme@cs.otago.ac.nz, 480/490

Some of our recent research has looked at finding optimal seamlines along which to 'cut' images when making mosaic images. Over the summer we have experimented with using distributed computing to accelerate this process. This project will continue this work and expand it to problems other than seamline detection, such as correcting colour and exposure differences between images. The aim is to create a fast image mosaicing system for processing aerial images on clusters of multi-core computers.

26. **Kinected Projection**

Steven Mills steven@cs.otago.ac.nz, 480/490

Projectors are often used in performance to provide lighting and visual effects. With depth sensors such as Microsoft's Kinect we can easily obtain a 3D model of the scene, and use that to identify surfaces on which to project information. The aim of this project is to investigate and develop methods for integrating multiple cameras, projectors, and Kinects to allow different patterns to be projected onto moving objects in a scene. For example, you might project a map of the world onto a sphere or digital make-up onto a person's face.

27. **Panoramas on the Sphere**

Steven Mills steven@cs.otago.ac.nz, 480/490

Panoramic images are commonly made by tiling images on a plane, but in many cases the natural space is a sphere. The use of spherical co-ordinates also leads to a number of interesting visualisations, such as Little Planets. This project will develop software to build panoramas directly in a spherical co-ordinate frame, and to visualise them in different ways.

28. **Buildings from Point Clouds**

Supervisor steven@cs.otago.ac.nz, 480/490

Many techniques for building 3D models from images yield a point cloud representation. Linking these disconnected points into a continuous surface is a hard problem in general, but in built environments there are many planar or near-planar surfaces. These planar regions are often separated by linear features in the original images, and are often at right angles to one another. This project will investigate techniques which make use of this information to make high quality 3D models of buildings from multiple images.

29. **Li-Fi: Data Communications through LED Light Bulbs**

    Yawen Chen yawen@cs.otago.ac.nz,
    Haibo Zhang haibo@cs.otago.ac.nz, 480/490

    LED bulbs can provide more than just illumination! Recent development of "Li-Fi" is an innovative way of wireless optical communications using the signals sent through the light bulb instead of radio waves as in Wi-Fi. This project aims to discover the advantages and disadvantages of LiFi, and explore the potential applications by investigating its special characteristics. One of the problems is how to design the layout of LED bulbs according to different application scenarios to achieve some objectives (e.g. efficiency/reliability/energy saving). See the BBC news (Oct. 2013) : 'Li-fi' via LED light bulb data speed breakthrough.

30. **Emerging Optical Network on Chip (ONoC): Efficient Topologies and Routing Algorithms**

    Yawen Chen yawen@cs.otago.ac.nz,
    Haibo Zhang haibo@cs.otago.ac.nz, 480/490

    With the increasing number of CPU cores on a processor chip, conventional electronic interconnects for core-to-core communication are becoming a bottleneck. Recent rapid advances have made the optical-based interconnection, Optical Network-on-Chip (ONoC), an attractive solution to breakthrough electronic interconnect limitations. To benefit from ONoC's advantages, this project is to design novel performance and energy-efficient topologies and routing algorithms for ONoC with the objective of achieving an optimal balance among performance, energy and complexity, by modelling, analyzing and evaluating the tradeoffs. See the HPC Hardware News (Jan. 2014): IBM Brings Nanophotonics to Real-world Manufacturing.

31. **Development of JTAG driver on Raspberry Pi**

    Zhiyi Huang hzy@cs.otago.ac.nz,
    Andrew Trotman andrew@cs.otago.ac.nz, 480/490

    The Raspberry Pi (RPI) is a small ARM-based computer promoted by Cambridge University for a couple of years now. Zhiyi has recently ported an OS called xv6 (Unix Version 6 adapted to PC by MIT) on RPI. xv6 on RPI will be used for studying OS in COSC440. To facilitate debugging while developing software in xv6, we need a debugger. JTAG (Joint Test Action Group) is a popular debug port used in embedded systems for debugging. Debuggers can communicate with chips via JTAG to perform operations like single stepping and breakpointing. It can also provide FUSE-like API for file systems. This project aims to design and implement an interface with JTAG on

RPI, based on open-source code like https://github.com/dwelch67/raspberrypi/blob/master/armjtag/README. The software will provide a base for further development of debugger and FUSE-like API for file systems.