# COSC480/490 Project Proposals for 2013

This is a list of suggested project proposals for COSC480 and COSC490 projects. If you have a project of your own that you would like to undertake instead, then you must discuss it with me or one of the staff. In the past we have often created special projects so that students can follow their own interests. Additionally, many of staff members have other interests that might be suitable for a project – you can find out about these by talking to them! In any case, suggesting a project that is not part of this list requires the agreement of a potential supervisor.

What you need to do:

- Select at least three projects that you think will interest you.
- Meet with the staff members who are offering those projects in order to discuss the possibility of taking them up. This is an absolute requirement – see below.
- Rank your top three choices in order of preference, and send them to me by email at steven@cs.otago.ac.nz by **3pm March 1st.**
- You will be advised early in week 2 of your project assignment.

Note that not all projects are suitable for both COSC480 and COSC490. An indication as to which project type(s) are suitable is given for each project. If you are really interested in a project that is not suitable for your project type then you may wish to talk to the staff member offering the project anyway – they may have related ideas that are more suitable.

Meeting with potential supervisors is important. We try to give students their first choices but we have to spread the work among supervisors and match projects to your particular abilities. If you make a choice without seeing the supervisor, we may ignore that choice when doing the allocation.

Steven Mills

400-level project co-ordinator

1. **Sorting algorithms (COSC480 or COSC490)**

   Michael Albert (malbert@cs.otago.ac.nz)

   The much derided bubble sort algorithm has an interesting property. In a single pass over *n* items it performs *n*-1 pairwise comparisons, and is capable of sorting $2^{n-1}$ different orderings – perfect performance when it succeeds in one pass. Subsequent passes, when required, lose this perfect efficiency. This project is to investigate the behaviour of *sorting operators* with an aim to generalising observations such as these. A combination of theoretical and experimental investigation using and extending PermLab is anticipated, and would especially suit a student with a good mathematical background

2. **Visualising permutation classes (COSC480 or COSC490)**

   Michael Albert (malbert@cs.otago.ac.nz)

   *Permutation classes* are simply collections of permutations. We can draw the graph of any permutation and this visual representation is very useful in identifying patterns that do or do not occur in it. How can we capture that same visual information when we are dealing with a large collection of permutations? For a few hundred, or even a few thousand, an animated view can help, but when there are millions? This project aims to explore different means of visualising such collections and integrating the best of them into PermLab.

3. **Computational modelling of synaptic plasticity (COSC480 or COSC490)**

   Lubica Benuskova (lubica@cs.otago.ac.nz)

   The project aims at extending the computational model of synaptic plasticity introduced by Benuskova and Abraham. The goal is to extend this model and computer code to several different experimental protocols taken from neurobiological literature in order to investigate its robustness and possible modifications. To keep compatibility with existing code, the preferred programming language is C++. The existing code is available at:

   http://www.cs.otago.ac.nz/homepages/lubica/gc.c.

   Objectives/timeline:

   - Reproduce the results from Benuskova, Abraham, Journal of Computational Neuroscience 22(2): 129-13, 2007, available at: http://link.springer.com/article/10.1007%2Fs10827-006-0002-x. This involves the study of the corresponding theory and understanding the above code and possibly writing and debugging an own computer code.
   - Extend, simulate and investigate the model for more inputs and variations in input activity.
   - Extend, simulate and investigate the model in order to reproduce at least one experimental protocol taken from the neurobiological literature on synaptic plasticity other than in Benuskova & Abraham 2007.

4. **Proving the correct behaviour for an online ballot box system (COSC480 or COSC490)**

   David Eyers (dme@cs.otago.ac.nz)

   Many years ago, I quickly hacked together some electronic ballot box software for my college student association. The Basic Online Ballot-box (BOB) has been adopted by a number of other organisations. The ballot system is 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.

5. **QR codes for security (COSC480 or COSC490)**

   David Eyers (dme@cs.otago.ac.nz)

   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. For Secure Shell (SSH) connections, for example, the key fingerprint of the remote computer you are first connecting to must be checked using an independent communication channel, in order to defeat so called "man in the middle" attacks. QR codes can be used to provide such an independent channel for verifying first-time SSH connections. The project will go further to encode security "capabilities" into QR codes, so that an authorised computer can temporarily delegate access rights to other computers (or smartphones) that have seen the dynamically generated QR code.

6. **QR codes for approximate location detection (COSC480 or COSC490)**

   David Eyers (dme@cs.otago.ac.nz) and Steven Mills (steven@cs.otago.ac.nz)

   (This is another QR code project – please see the introduction above for information about what QR codes are.)

   This project investigates the use of QR codes that encode information about the spatial location at which they are "planted" physically (e.g. on a poster board within a room). The project aims to develop smartphone software that can react to its physical context. For example, a smartphone could be configured to automatically silence its ringer if it detects that it is in a meeting room or a lecture theatre. At a basic level, decoding any such planted QR code will provide information about the location of the smartphone (e.g. room and facing direction). Detecting and decoding a set of planted QR codes in one room would provide further abilities to narrow down location.

7. **Bringing command shells and terminals closer together (COSC480 or COSC490)**

   David Eyers (dme@cs.otago.ac.nz) and Richard O'Keefe (ok@cs.otago.ac.nz)

   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.

8. **Peer-to-peer Mega (COSC480 or COSC490)**

   David Eyers (dme@cs.otago.ac.nz)

   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. Starting points for development are likely to include open source tools that provide similar functionality to Dropbox (e.g. ownCloud), and abstractions built over peer-to-peer software to better handle sets of files than BitTorrent (e.g. GitTorrent). A way to specify policy can be devised to balance between keeping distributed storage as up-to-date as possible, and minimising the network "cost" of keeping content synchronised.

9. **CSS-based security for web browsers (COSC480 or COSC490)**

   David Eyers (dme@cs.otago.ac.nz)

   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. I did forensics after an attack in which an intruder's account had been added into a WordPress installation. The intruder had cloaked themselves by using a "surname" field value that included JavaScript that deleted its own row from the HTML table displaying the user list. Thus a genuine administrator would not see this extra user when using the WordPress control panel pages.

   On the server-side, had WordPress "escaped" all of the data in the "surname" field, this problem would not have occurred. This type of problem 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 to permit or deny the execution of script tags at different points within page rendering. This system will then be tested to defend against attacks such as the one described above.

**10. Web pipes (COSC480 or COSC490)**

David Eyers (dme@cs.otago.ac.nz)

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. Shells create anonymous pipes between process invocations sequenced using the '|' character. Named pipes allow pipe stages to be connected together at a time independent from the time that they are created: the pipe endpoints sit in the filesystem (or a pseudo-filesystem on NT). Pipes can – in effect – span multiple computers, by including invocations of tools such as the Secure Shell (SSH), netcat, or wget within pipelines.

This project integrates named pipes into web browsers 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. An example would be a user navigating to a page that either uploads a file (like an HTML file upload control), or downloads a file, where the data source or destination can be local shell scripts targeting named pipes created by the browser.

**11. Speeding up LaTeX (COSC480 or COSC490)**

David Eyers (dme@cs.otago.ac.nz)

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.

Much of LaTeX's typesetting quality is due to it NOT being a WYSIWYG system: when generating output, it can try multiple different ways to space and hyphenate words in a paragraph to optimise the number of words per line for the best readability.

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.

**12. Indoor localisation with wireless sensor network (COSC480 or COSC490)**

Zhiyi Huang (hzy@cs.otago.ac.nz) and Haibo Zhang (haibo@cs.otago.ac.nz)

Wireless sensor networking (WSN) is a rapidly growing technology due to its low power consumption and cheap price. It has a wide range of applications, including smart energy, remote hospital care, industrial process control, home automation, and etc. In this project, we apply WSN to tracking the location of an indoor object such as a mobile phone user. Currently we have used dozens of wireless sensors (telosb) for localisation based on RSSI (received signal strength indication). The localisation accuracy can be up to 2-3 meters. In this project, we will investigate how to use the varying signal strength to further improve the accuracy of the localisation.

**13. A neural network model of cognitive modes and mode-switching (COSC480 or COSC490)**

Ali Knott (alik@cs.otago.ac.nz)

There's a lot of interest in cognitive science at the moment in the idea that brain processing happens in several different 'modes', which are implemented by distinct distributed networks of brain regions. For instance, there's one brain network which is active when you are engaged in a sensory or motor task, and a quite different network which is active when you are resting quietly, and yet another network which is active when your attention is drawn to a stimulus in the world. An interesting research question concerns how the brain decides which network to engage at any given time. There must be a control mechanism which selects the network which is most appropriate, but we don't know much about what this is. In this project, you will read up a bit about large-scale brain networks, and implement a (simple) model of the mechanism which performs this selection process.

**14. Discovering large-scale brain networks with functional connectivity analysis (COSC480 or COSC490)**

Ali Knott (alik@cs.otago.ac.nz)

This project is also on the topic of large-scale functional brain networks (see project 1 above). One way these networks have been identified is by analysing brain activity gathered by functional magnetic resonance imaging (fMRI): they can be identified as sets of brain regions whose activity is highly correlated. (Typically the activity of the regions in one network is anticorrelated with that of regions in other networks.) In this project, you will obtain some publicly available fMRI data, and some analysis tools, and explore some hypotheses about brain networks: specifically, you will be looking for brain regions involved in deciding which network to activate. We will probably use data and tools from the functional connectomes project.

This project would suit someone with some maths background, and an interest in correlational analyses.

**15. Graph Layout on a Sphere (COSC480 or COSC490)**

Shawn Martin (smartin@cs.otago.ac.nz)

A graph consists of nodes connected by edges. Graph layout algorithms are designed to draw graphs subject to various aesthetic criteria. Examples of such criteria include minimal number of edge crossings, low density of nodes, symmetry, et cetera. Most graph layout algorithms draw graphs in two dimensions. There are some layout algorithms that work in three dimensions, but the resulting drawing is typically very hard to visualize. What about drawing a graph on a sphere? The layout would be three-dimensional, but would be easier to visualize. In addition it would have some interesting geometrical advantages over two-dimensional layouts. In this project you would learn about graph drawing and implement a spherical layout algorithm. An interactive visualization component would also be considered.

**16. Multigrid Graph Layout for Image Distortion (COSC480 or COSC490)**

Shawn Martin (smartin@cs.otago.ac.nz) and Steven Mills (steven@cs.otago.ac.nz)

Algebraic multigrid methods are commonly used for numerically solving partial differential equations. They are fast iterative methods that are designed to solve sparse linear systems of equations have a certain specific form (diagonally dominant, symmetric, and positive definite). We have developed a certain class of graph layout algorithms that result in the same type of sparse linear system. Our graph layout algorithm is specialized to perform image distortion/magnification. In this project you will learn about algebraic multigrid methods and apply your knowledge to solving the linear systems specific to our graph layout algorithm. The end result will be a real-time image distortion/magnification application. A paper describing the algorithm and what it does can be found at http://dx.doi.org/10.1145/2425836.2425858.

**17. Molecular Design (COSC480 or COSC490)**

Shawn Martin (smartin@cs.otago.ac.nz)

Like Humpty-Dumpty, molecules can be broken into little pieces. Unlike Humpty-Dumpty, they can be put together again. The interesting thing is, however, that you might not have the same molecule you had when you started. In this project you will implement code to read data describing a set of molecules, produce the corresponding molecular graph (atoms are nodes, bonds are edges), break the molecule into pieces, and output a description of each piece. How the pieces can be put together will then be examined.

**18. Fast mobile pose estimation (COSC480 or COSC490)**

Brendan McCane (mccane@cs.otago.ac.nz)

Augmented reality (AR) applications require fast position and pose estimation so that the correct augmented image can be drawn to the screen. Most current solutions make use of markers to make matching between images easier, but markerless AR is the ultimate goal. The goal of this project is to investigate and implement a fast pose estimation algorithm suitable for use on mobile devices (smartphone, tablet etc). If that turns out to be too easy, then you can develop an AR application too!

**19. Skin and Bones (COSC480 or COSC490)**

Brendan McCane (mccane@cs.otago.ac.nz)

Many of you will have seen the recent “image” of King Richard III that was reconstructed from measurements of his recently discovered skull. Most such reconstructions are more artistic than scientific. I am working with a PhD student in Anatomy to make the process more scientific. The goal is to reconstruct skin surface data using bone measurements and a few auxiliary variables (e.g. body mass index). Most of the data has already been collected.

**20. Kinected Projection (COSC480)**

Steven Mills (steven@cs.otago.ac.nz)

Projectors are often used in performance to provide lighting and visual effects. If we add cameras to the mix, we can make these effects move with performers or other objects. 3D sensors, such as Microsoft's Kinect, add even more information. The aim of this project is to investigate and develop methods for integrating multiple cameras/projectors/Kinects to allow different patterns to be projected onto moving objects in a scene.

**21. Distributed Seam Finding (COSC480 or COSC490)**

Steven Mills (steven@cs.otago.ac.nz) and Zhiyi Huang (zhuang@cs.otago.ac.nz)

Some of my recent work has developed a new method for finding seamlines in mosaic images (http://dx.doi.org/10.1016/j.isprsjprs.2012.11.003). These are the lines along which you cut individual images in order to assemble them into a larger image. In theory, this method can be easily distributed across processing cores or a network of computers. The aim of the project is to investigate how this and other image mosaicing tasks can be efficiently distributed across a cluster of multi-processor machines.

**22. Texturing 3D Models (COSC480 or COSC490)**

Steven Mills (steven@cs.otago.ac.nz)

Various techniques exist for constructing 3D models from 2D images. The result of these is a 3D mesh model, which can then be textured by projecting the original images on to the model. However, each point on the model is usually visible in more than one image. If these images don't agree on what colour the model should be at that point then we need some way to choose or blend between them. The aim of this project is to investigate techniques for solving this problem.

**23. Vision-Based Quadrotor Control (COSC480 or COSC490)**

Steven Mills (steven@cs.otago.ac.nz) and Anthony Robins (anthony@cs.otago.ac.nz)

We have a quadrotor drone that can be controlled from a computer via WiFi. It is equipped with two cameras and an altimeter for sensors, and we'd like it to do something autonomous and interesting. Possibilities include seek-and-land (locate a target, fly over it, and land on top of it) or sense-and-avoid (determine time to impact to hazards and take action to avoid collisions).

**24. Generic Programming and Random Search (COSC480 or COSC490)**

Richard O'Keefe (ok@cs.otago.ac.nz) and David Eyers (dme@cs.otago.ac.nz)

If you need to find the best value of a function and you know nothing whatever about the function, random search is as good as it gets. There are many improvements that can be made if you think the function is reasonably nice, leading to things like "Improving Hit-and-Run". There are many options to try, so we'd like a meta-algorithm for constructing random search algorithms. The purpose of this project is to explore several ways of doing it, like a program that emits source code, the M4 macro processor, C++ template meta-programming, and partial execution.

**25. Designing for Error (COSC480 or COSC490)**

Richard O'Keefe (ok@cs.otago.ac.nz) and David Eyers (dme@cs.otago.ac.nz)

Computer systems like Novopay can do a great deal of harm when they fail to pay people. What would it be like to develop a system such as a payroll system in which

- inputs are recorded with provenance and authentication
- no inputs are ever deleted (although they might be moved to slower permanent storage)
- any action the system proposes to take, to refuse to take, or to take incorrectly can be vetoed, forced, or corrected by a user with sufficient authority
- such overrides are logged and linked to the relevant inputs
- there is always a sufficiently authorised user
- sufficiently authorised users can installed "common sense" rules to catch blatant stupidity before it happens?

What would it cost us in development time and performance to build systems this way?

**26. Patently obvious (COSC480 or COSC490)**

Richard O'Keefe (ok@cs.otago.ac.nz)

We have a number of document collections for doing information retrieval research on. I am particularly interested in a collection of 100,000 US patents. The Wikipedia collection has links between articles; and patents refer to each other. How can we visualise a collection of this many documents? How can we display them so that interesting patterns of properties or connections stand out so we notice them? This involves extracting information from XML and data visualisation (not infographics) and perhaps the use of a graph database and calculating properties of large graphs.

**27. Improved language identification (COSC480 or COSC490)**

Richard O'Keefe (ok@cs.otago.ac.nz)

In 2012 we had a project to identify the language a document was written in. (See the poster near RAOK's office.) This was quite successful, but explored a limited range of techniques. We'd like to try more techniques and perhaps a wider range of languages, and in particular we'd like to do better at cleaning the data.

**28. XML parsing within programming languages: safety, efficiency, and convenience (COSC480 or COSC490)**

Richard O'Keefe (ok@cs.otago.ac.nz) and David Eyers (dme@cs.otago.ac.nz)

Parsing and generating XML is a common task within today's software. Libraries for parsing and manipulating XML are available for most common programming languages. However, some languages, such as Scala, have chosen to handle certain XML operations right down at the level of their language compiler. The co-supervisors of this project have both been involved in efforts to retrofit Scala-like XML support into other programming languages, such as Python. Is it worth tightly integrating XML into a language? Is enough expressiveness provided in the low-level language features to capture the majority of use cases? What are the degrees of difference in speed and memory usage, and security against injection, from handling XML in different ways within a language? This project aims to study these questions, and potentially implement improved language support for XML.

**29. Power management for a distributed data base in a box (COSC480 or COSC490)**

Richard O'Keefe (ok@cs.otago.ac.nz) and David Eyers (dme@cs.otago.ac.nz)

Distributed data bases like Chord, Apache Cassandra, Tapestry, CouchDB, or Riak let you store information in multiple nodes of a network, and access it from multiple nodes, trying to stay available even when individual nodes fail. If you have a cluster computer made up of a large number of processors in the same cabinet, you might want to put a node in a reduced power or very low power state, not because it has failed, but because the system isn't currently being used heavily enough to justify using all that electricity. In a full power state, a node would participate fully in data base support, accepting updates and responding to queries. In a very lower power state, the only thing a node can respond to is a request to turn back on. What is a good policy for choosing which nodes to turn on and off? Are the distribution protocols designed to handle sporadic failures adequate to cope with frequent deliberate shut downs/restarts? What difference does having a reduced power state that can track updates but not answer queries make?

**30. Quantifying Conceptual Density in Text (COSC480 or COSC490)**

Anthony Robins (anthony@cs.otago.ac.nz) and Ali Knott (alik@cs.otago.ac.nz)

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.

**31. Attractor spaces in Hopfield nets (COSC480 or COSC490)**

Anthony Robins (anthony@cs.otago.ac.nz)

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.

**32. Serial learning in neural networks (COSC480 or COSC490)**

Anthony Robins (anthony@cs.otago.ac.nz)

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.

**33. Simple robot vision (COSC480 or COSC490)**

Anthony Robins (anthony@cs.otago.ac.nz)

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. Other projects based on Lego robots may be available – come and talk to me if you have ideas.