Student Research Contest

Extended Abstracts

1. Development of a Lunar Regolith Database
   Arielle Baine, Spelman College.
2. Identifying Worms in a Binary Picture
   Nicole Dawson, Spelman College.
3. A Case Study for Developing Applications for a Mobile Device: the iPhone/iTouch Platform
   Derek Hardison, Roanoke College.
4. The Evolution of Swarm Intelligence
   Mark Lucas, Roanoke College.
5. Exploring Biological Networks by Integrating Annotation: A Case Study of Congenital Heart Defects
   Tahirali Motiwala, Wofford College.

Name: Arielle Baine

Title: Development of a Lunar Regolith Database

Advisors: James Thomas Hale, Sr and Dr. Alfred Watkins, Spelman College

Abstract

Lunar regolith simulants (and lunar dust simulants) are synthetic analogs that approximate, to a known extent, one or more regolith properties or characteristics at a particular lunar location or region. The primary characteristics of the simulants are inherited properties of the material. In particular, primary characteristics include particle composition, particle size distribution, particle shape distribution, and bulk density. These simulants are used by engineers and scientists in order to produce effective products, such as reliable hardware that will function long term, on future lunar missions. For engineers and scientists to utilize the lunar regolith simulants they must have understanding of the simulants, as well as the lunar regolith reference material. Some lack this understanding and rely on low-fidelity or inappropriate substitutes, such as baby powder, toner or printer cartridges, and dirt or sand because they match one (1) primary characteristic of a lunar sample. It has come to the attention of MSFC and geologists that this problem arises simply because these engineers and scientists are unaware of the risk that inappropriate simulants cause. A realistic solution would be to create a repository combining Figure of Merit (FoM) ratings (a quantitative comparison, between 0 and 1, produced by a complex algorithm of two particular materials composed of geologic components), “fit-for-purpose” matrices, attainment surveys, data from completed tests using simulants, and other beneficial documents and references. This repository would be fashioned into a database, entitled “Lunar Simulant Database for Users”, that would be used as a comprehensive and effective tool for researching simulants, including but not limited to, their properties and behaviors. Then, the information gained can be applied to the engineers and scientists’ respective applications. Moreover, from this database it can be understood that certain materials cannot be substituted for lunar regolith simulants and lunar samples without added risks, even if these materials are cost effective and easy to obtain. Engineers and scientists should focus on understanding lunar regolith simulants and how the appropriate simulant selection can lead to successful lunar missions.

Identifying Worms in a Binary Picture

Nicole A. Dawson,Spelman College
Advisors: James Thomas Hale,Sr,Spelman College,Dr. Alfred Watkins,Spelman College,
Dr. Badri Roysam, Rensselaer Polytechnic Institute
Mentor: Tenicka Turnquest, Rensselaer Polytechnic Institute
Department of Computer and Information Sciences
Spelman College

The Farsight Project is a project dedicated to the development and dissemination of a next-generation toolkit of image analysis methods to enable quantitative studies of complex & dynamic tissue microenvironments that are imaged by modern optical microscopes. Some examples of these microenvironments are brain tissue stem cell niches, and the development of embryonic tissue and tumors (http://www.farsight-toolkit.org/wiki/The_Worm_Project). Within the Farsight project lays the Worm Project. It is also a toolkit designed to track and observe C. Elegans worms impacted by outside influences such as pheromones. The worms were placed in a petri dish wherethey were given pheromones and photographed every few minutes to observe the movements and reaction of the worms. There is currently a Farsight Toolkit that generated the results of the C. Elegans worms' reaction and behavior. This worm project provides a "new range of quantification metrics for nematode social behaviors" (http://www.farsight-toolkit.org/wiki/The_Worm_Project) that will hopefully one day help others study humans, namely the human brain. The algorithm that the worm project is based on was created by Nicolas Roussel.Roussel created an algorithm that simultaneously tracks imaging fields that contain multiple worms (http://www.farsight-toolkit.org/wiki/The_Worm_Project).

In order to understand the Farsight Project a series of tutorials on the Visual Toolkit (VTK), Insight Toolkit (ITK), MatLab, Python, and Microsoft Visual studio were done aswell as a thorough study of the Farsight Wiki Page and the Worm Project wiki page.We studied under Dr. Badri Roysam at Rensselaer Polytechnic Institute and one of his graduate students Tenika Turnquest. Turnquest worked on the worm project as a whole and gave me a portion of her research problems.

One problem was Changing the colors of the worms so that we would be able to identify them in the grayscale pictures I was given .Turnquest also converted the grayscale pictures to binary images therefore making it easier to identify the worms and manipulate C++ and python code in order to display the worms location. The pictures were already magnified a great deal so that we could see them but trying to distinguish between a portion of the Petri dish, a small fragment of dirt or anything that got in the way of the lens and the actual worm itself was difficult.

The second problem was to find the exact location of the worm. I had to identify the pixel value and x and y coordinates of the worms in order to identify them.

Futre Research

The third problem was making sure that we identified and tracked the correct worm. However due to time restraints I was not able to use Nicolas Roussel's algorithm for tracking and identifying the worms.

A Case Study for Developing Applications for a Mobile Device: the iPhone/iTouch Platform

Derek Hardison
Roanoke College
Tenable Security
dhardison@tenablesecurity.com

Developing an application on the Apple iPhone is a process that is tightly integrated with Apple's Xcode and UI Builder IDE. The specifications for the application I developed for my institution included tools to view school events, search for faculty/alumni/students, and view profiles of students enrolled in a professor's class. Apple's adjudication to implement Objective C as the API programming language is a tenable choice; it offers features that allow Apple to ensure quality of iPhone core services regardless of applications installed by a user.

A majority of work done to develop the user interface is performed through Apple's UI Builder. There was an option to design only by code; however, for this project it was unnecessary. Writing the user interface strictly by hand would have made the program much more abstruse to developers looking to extend it, and would have added avoidable overhead. The UI Builder allows a developer to drag and drop commonly used objects onto a canvas which consequently reduces the amount of code one has to sift through. The user interface is attached to event handlers written by a developer; these event handlers are called to perform whatever routine is desired. In general, different user interface objects allow for different actions (e.g. button tap).

All three of the tools developed implement the default iPhone table view. Granted that there are alternatives, for our purposes it offered all the features required in addition to simplicity (the ability to list news events/names and scrolling). In order for a routine to be performed, each user interface object must be connected to a variable. These variables, which are allocated in Xcode, effectively become proxies between the frontend (UI) and backend (main code).

Apple's iPhone API offers methods which may be overloaded to customize various features of user interface objects. For example, to set the height of a certain row in your table there is a method which is called by the system that passes the row index and expects some number to be returned. This method was frequently overloaded in all three of the projects created for my institution. For instance, the page that displays student details in the class roster implements a table view with one large row. To enumerate the process, the row height is calculated based on the height of the text and images in the window. Interestingly enough, the API does not perform this automatically for the developer.

The fulcrum in the learning process was memory management. There was a pretension about Apple's memory management implementation which made it appear that variables were randomly released. After further investigation it was discovered that Objective C ran a counter on objects which kept track of the number of places referencing it. In other words, memory was released when the garbage collector checked a memory pool and a counter was zero. More specifically, my largest issue stemmed from returning variables from methods; after a method call finished the retain counter was decremented to zero (generally). The variable would hang around until the garbage collector sneaked in and tacitly released the memory (which led the application to seemingly crash at random intervals).

In my opinion, the neatest feature about Apple's iPhone is the ability for applications to exchange information. Each application that extends the iPhone API is encapsulated in its own address space; that is, applications do not have direct access to other applications' memory addresses. This allows Apple to prevent developers from altering the core functions of the device and other applications. To enumerate, the operating system manages the communication of data by passing messages between programs (copies of the data). Later on, the data is reviewed by the receiving application which may decide the action to be performed. As a result, the application that received the message interprets the data how it pleases, and does not perform any action as a consequence of another program executing one of its routines directly.

References

1. "Memory Management Programming Guide for Cocoa." *Developer Connection*. 2009-08-18. Apple, Web. 14 Sep 2009. <http://developer.apple.com/mac/library/documentation/Cocoa/Conceptual/MemoryMgmt/MemoryMgmt.html>.
2. "iPhone Reference Library." *Developer Connection*. Apple, Web. 14 Sep 2009. <http://developer.apple.com/iphone/library/navigation/index.html>.

The Evolution of Swarm Intelligence

Mark C. Lucas

September 2009

Roanoke College

NP Complete problems were of interest even before Eniac was first activated in the 1940’s. As the computation power of machinery has gone from hundreds of operations per second to thousands to millions to billions, the tractability of problems such as K-Sat has not increased. Problems whose solutions run in exponential or factorial time are just not practically solvable for any but the simplest of cases (Yates). The fact that many of these problems are easily stated and intuitively understood makes them exceptionally useful in studying alternative problems solving techniques.

Problems which are NP Complete or NP Hard are not interesting solely from an academic standpoint, but also from an applied or practical point of view as well. In fact, examples of Graph Coloring Problems and Traveling Salesman Problems can be found in several different kinds of industry. For instance, the route which a robotic arm navigates around a steel plate while drilling holes is a TSP. Trying to make maps of the ever changing globe is a new graph coloring problem seemingly every year. The robot arm wastes time if it travels a lot of extra distance, and time is money. A map maker has to pay more to print for every color that he uses, so using the fewest number of colors while still making sure no two adjacent countries are the same saves him money.

Generally speaking, there are three approaches to take when attacking a problem which is NP Complete. The first is to do an exhaustive search of the solution space. This provides a provably best answer, but is also, as previously mentioned, prohibitively expensive from a time standpoint. The second is to try to find some equivalent problem which would run in polynomial time, and which still gives a provably correct answer to the NP Complete problem; that is, to show that P = NP for some case of an NP Complete problem and a problem which runs in Polynomial time (Cook). Unfortunately, many people have been looking for this “Holy Grail” of computing for decades with no luck. In fact, some computer scientists believe that the answer has not been found because it does not exist and are working to prove as much. The third approach is to develop some heuristic which gives an answer which one can be reasonably sure is reasonably close without being able to prove either (Eiben, et al).

The goal of this research was to become acquainted with the third method, looking at how heuristics work and what methods might be useful when approaching an instance of an NP Complete problem. The problem chosen was the aforementioned Traveling Salesman Problem, and the heuristics used were Genetic Algorithms and Particle Swarm Optimization.

The research consisted of four parts. The first consisted of problem and method research, including a basis in graph theory as well as the history of GA's and PSO. The second phase was practical, focusing on an original solution using a modified GA. The third stage was practical as well, in which an original solution using PSO was also developed. The fourth phase was comparison, in which the two heuristics were used to develop circuits in a head to head competition. The results of stages one and two were presented at MathFest 2008 in Madison, Wisconsin. This talk will present a brief overview of these first two stages, but will focus on stages three and four as well as conclusions made and lessons learned. This research was completed in April, 2009.

Works Cited

Cook, Stephen. "The P Versus NP Problem". Problem Statement, Millennium Problems, Clay Insitute. http://www.claymath.org/millennium/**. May, 1998.**

Eiben, A. E., Smith, J.E. Intrduction to Evolutionary Computing. Springer, New York, 2003.

Yates, Mike. What Computers Can't Do. Plus Magazine (online Edition). http://plus.maths.org/issue5/turing/. May 1998.

Georgia Tech. "The Traveling Salesman Problem". March 2008. http://www.tsp.gatech.edu/problem/index.html.

Tahirali H. Motiwala & Dr. Anil Jegga
Biomedical Informatics Department
Cincinnati Children's Hospital
Wofford College of Spartanburg, SC
motiwalath@email.wofford.edu
Dr. Angela & George Shiflet

Exploring Biological Networks by Integrating Annotation: A Case Study of Congenital Heart Defects

Understanding the molecular basis of biological systems in development and disease is critical for development of innovative intervention methods and to improve health care. Research in genomic sciences and technologies is undergoing a significant transformation driven, in part, by recent advances in: (i) rapid and inexpensive sequencing of various genomes, (ii) gene-chip technology, and (iii) easy availability of cheap and enormous computational power. While the first two advances resulted in the generation of vast amounts of published information and experimental data, the enormous computational power has provided hope that nuggets of biological knowledge can be extracted from this data deluge. Using congenital heart defects (CHD) as a test case, in the current study, we integrate several types of "omics" data and then apply functional annotations and network-based bioinformatic approaches to (a) identify and rank potential novel CHD candidate genes and (b) identify transcriptional and post-transcriptional regulatory mechanisms underlying CHD. Specifically, using an intersection of known genes associated with heart development, CHDs, and heart abnormality in mouse models as a training set, we rank a subset of the human genome (test sets) potentially associated with CHD. Integration of the various data revealed likely downstream targets of miRs-1, 133, 143, 145, and 196a-2; specifically of interest were HAND1 and GATA6 as targets of miR-196a-2. Also, potential regulatory interactions between miRs-1, 133, 143, 145, DUSP6, and MKL2 were hypothesized. These results demonstrate the strength of integrating "omics" and network analysis data in the prioritization and study of regulation of potential disease genes.