CCSC Consortium for Computing Sciences in Colleges

Southeastern Region

20th Annual Southeastern Conference

Student Research Contest

Extended Abstracts

November 10 and 11, 2006
Lipscomb University
Nashville, Tennessee

Student Research Contest

Extended Abstracts

1. An extensible framework for the mathematical manipulation of music
   Elom K. Abalo, Wofford College, Spartanburg, SC.

2. Peak-finding for coherent two-dimensional spectroscopy
   Keshia Clarke, Spelman College, Atlanta, GA.

3. Utilizing clip planes to improve the clarity of rendered images
   Danielle Edwards, Spelman College, Atlanta, GA.

4. Accessible accessibility
   Drew Fleming, Roanoke College, Salem, VA.

5. Study of mobile agent algorithmic implementation and development
   Christopher Smith, Roanoke College, Salem, VA.

6. Natural language database interface portability
   Hampton Smith, Roanoke College, Salem, VA.

7. Machine learning on an AIBO robot
   Jackie Soenneker, Centre College, Danville, KY.

8. Multi-dimensional chromatography informatics
   Katrina Stewart, Spelman College, Atlanta, GA.

9. Image tracing and tracing server development
   Marisa Toodle, Spelman College, Atlanta, GA.

10. Behavioral based visual queries
    Jessica Young, Roanoke College, Salem, VA.

"An Extensible Framework for the Mathematical Manipulation of Music"
Elom K. Abalo
Wofford College
Spartanburg, SC
abaloek@wofford.edu
Dr. Sloan, Department of Computer Science

Music is most often described as an art, yet when one decides to study it thoroughly it easily takes the form of a science dealing with waves, superposition, etc. It is this very malleability of music which makes it an interesting subject to analyze scientifically. The sounds heard in a symphony are essentially a collection of notes arranged in a certain order and pattern to yield the final result. With the advent of computers and music stored in digital format, the possibilities for experimentation are virtually limitless. What if one were to meddle with the individual notes, change the waveforms, or attempt to play a musical work in a different key or different meters, brief apply different rules?

By choosing to view music as a highly multi-leveled organized structure of "sound objects" in time, the idea of changing the various parameters of sound envelopes, waveforms, etc., on different levels becomes then an interesting and challenging task to accomplish with the medium of computer science. This prompts the need for an easily modifiable environment or structure to accomplish such tasks.

The goal of this project was the creation of an extensible sound manipulation architecture using *Mathematica* with the primary incentive being a mixed scientific and artistic curiosity. A curiosity about what is observed or rather heard for example when one plays a piece of music backwards, or one plays "Are You Sleeping, Brother John?" with a set of notes whose sound envelopes, waveforms, textures vary as specified functions of time. The two essential phases were the implementation of a structure to play sounds, and a system for the exploration of granular synthesis.

The results of the first phase include: a function to define a sound whose frequency transitions smoothly from one frequency to another, a set of sound envelope functions to shape the waveforms, a function to combine a list of sounds according to their positions in time thus producing harmony, a function to set the tempo and determine the durations of standard notes, and finally an overall function which applies all the previous operations on a list of strategically placed parameters. The output of the last function is then fed into one of *Mathematica's* built-in function for conversion to sound.

The indispensable function developed in this phase is *glissValues[f1_,f2_,dur_]* which as described earlier produces a sinusoidal wave whose frequency varies smoothly from *f1* to *f2* over the time range *dur*. The sounds obtained with glissValues are similar to the continuous tone of a slide whistle. The output of *glissValues[ ]*, a list of amplitude values, can then be sent as input to any of the seven envelope functions currently developed. Choosing *ExpDecEnv[list_]* for example would apply an exponentially decaying envelope to the sound. As mentioned in the title one of the key aspects of the architecture is its extensibility which means for example that the user is allowed to design any fantastic sound envelope curve or even re-arrange the code of the functions to obtain a wanted effect. With a collection of sounds to which envelopes have been applied one can organize them in time with *Chord4[list1_, list2_, offset]* which superposes two offset amplitude lists, then set the tempo and standard note durations with *Tempo[MM_, Num, Denom_]* by passing the metronome setting number *MM*, the numerator and denominator of the time signature *Num* and *Denom*. All of the previous tasks are combined in *ChordFuncEnv4[list_]* which takes a list consisting of all the parameters given for the above functions and creates the amplitude list for the specified song which can then be heard with *ListPlay[ ]*. Such an input list was devised for "Are you Sleeping Brother John?" and successfully played.

The granular synthesis phase consists of a function to define a very short sound or sound grain to which a sound envelope is applied, a function to create many grains with different characteristics while positioning them in time, a function to superpose these grains taking into account their overlapping thus creating a new sound object called a cloud, and finally a function to take in the specifications of many clouds to output a compiled list which can be played using *Mathematica's* built-in sound functions. Most of the functions used in granular synthesis are borrowed or derived from those created earlier in the first phase. Other secondary functions helping in the visualization of the sound objects, although not necessary in the process of sound production, are also provided.

Using the previous phase as a basis, the new functions for granular synthesis consist notably of *Grain4[freq1_, freq2_, amp_, duration_, waveform_, envfunc_, offset_]* which using *glissValues[ ]*'s properties defines a short sound whose frequency transitions from *freq1* to *freq2* with an amplitude *amp*, a duration *duration*, a waveform *waveform* which can be sinusoidal, square, or any other shape, an envelope *envfunc* which the user is at liberty of designing, and assigns an offset *offset* to each grain. The ultimate function however is *Cloud6[grainDensity_, grainAmplitude_, grainWaveform_, grainEnvelope_, grainDuration_, cloudDur_, upFreqBound_, lowFreqBound_, μ_, σ_, timeIntervals_]* which creates a sound cloud consisting of sound grains with grain density *grainDensity*, grain amplitude *grainAmplitude*, grain waveform *grainWaveform*, grain envelope *grainEnvelope*, grain duration *grainDuration*, cloud duration *cloudDur*, upper frequency boundary *upFreqBound*, and lower frequency boundary *lowFreqBound*. The grains are normally distributed throughout the cloud with mean μ and standard deviation σ. Each of the parameters is applied accordingly over subdivisions of the cloud duration: *timeIntervals*. *Cloud6*[ ] for example was used to play "Are you Sleeping John?" with sound clouds of 500 grains per second, grains with of sinusoidal waveforms, triangular envelopes, grain durations of 0.05 seconds, and hyperbolically increasing frequency boundaries. The mean for each cloud was the frequency of the conventional note.

Overall the inspiration for this project comes primarily from the concept of microsounds explored by Curtis Roads [3] and the *Musica* architecture designed for *Mathematica* [2]. Combining aspects from both yielded a *Musica*-like architecture with the sound cloud generator properties described by Curtis Roads. It allows one to work with standard concepts of music (durations, time signatures, etc.) just as *Musica*, but unlike *Musica* it allows the user to work with a much larger variety of sounds instead of the conventional sinusoidal waves. Conversely, this framework allows one to create sound clouds but also assimilate them to sound objects which can be manipulated, and entered in a list to be played like conventional notes.

One of the possible developments from this project is a framework which allows users to create variations on a melody. For example, it could be developed so that it would read pre-recorded music on a CD in MIDI format, convert it into its internal representation, thus allowing the user to change the characteristics of the sound, add extra "sound objects", apply exotic mathematical mappings to restructure the original melody, brief experiment virtually without limit.

References
[1] Chandra, A. *A Composer's Guide to Sound Production with Mathematica.* http://library.wolfram.com/infocenter/MathSource/3538/, 1992.
[2] Merziotis, C. *Musica.* http://library.wolfram.com/infocenter/MathSource/4945/, 2003.
[3] Roads, C. *Microsound*. The MIT Press, Cambridge-Massachusetts, 2001.
[4] Roads, C. *The Computer Music Tutorial*. The MIT Press, Cambridge-Massachusetts, 1996.
[5] Suits, B.H., *Physics of Music-Notes*. http://www.phy.mtu.edu/~suits/Physicsofmusic.html, 1998-2006
[6] *Time Signature*. http://en.wikipedia.org/wiki/Time_signature, 2006.
[7] Wellin, P., Gaylord, R., and Kamin, S. *An Introduction to Programming with Mathematica*. Cambridge University Press, 2005.

**Peak-Finding Algorithms for Coherent Two-Dimensional Spectroscopy**

Keshia Clarke
Spelman College
Atlanta, Georgia
keshiaclarke@yahoo.com
Faculty Sponsor: Dr. Charles Hardnett, Computer Science Department
Dr. Peter Chen, Chemistry Department

Spectroscopy is the use of light and electromagnetic radiation, to study the structure and behavior of matter. Understanding the shape, structure, composition, and behavior of atoms and molecules in spectroscopy helps predict whether certain molecules will react with one another, study how certain atoms and molecules are arranged, and identify these atoms and molecules in different samples.

As noted by Chen [1], in the early 1900s, scientists were only able to use film to show the wavelength emission spectra of compounds and therefore led to only a qualitative analysis. Currently, a one-dimension spectroscopy is being used that graphs intensity and wavelength to give a qualitative and quantitative analysis. The problem with the conventional one-dimensional spectroscopy is that the plot is complicated with the overlapping of three processes occurring at the same time. These three processes are the basics of optical spectroscopy: rotational, vibrational, and electronic. For the future, the ability to separate and sort peaks is explored using a new coherent two-dimensional form of resonance Raman spectroscopy [Figure 4]. This experimental technique distributes normally congested and disordered rotational-vibrational peaks that have the same vibrational sequence in order by rotational quantum number.
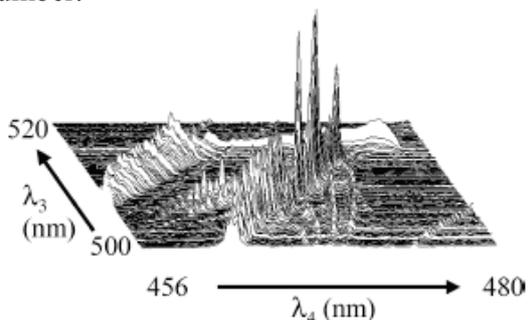

Figure 4. Experimental 2D data shown in the form of a stacked plot. The intensity of the detected light (arbitrary units) is shown as a function of input wavelength λ_3 and output wavelength λ_4.

Our solution framework involves two-dimensional peak-finding algorithms based on one-dimensional peak-finding approaches. This poster focuses on the aspects of one-dimensional approaches from which we are developing an appropriate two-dimensional approach. The computer must define a real peak which extends the need for a specialized peak finder. Exploring peak-finding algorithms for spectroscopy is conducted in the context of the IGOR software. IGOR Pro 5 is an extraordinarily powerful and extensible scientific graphing, data analysis, image processing and programming software tool for scientists and engineers that was purchased for this research project. This interactive software has already been programmed to handle large sets of data relatively fast. The peak-finding algorithms taken into consideration are the Gaussian, Lorentzian, Voigt, ExpGaussian, ExpConvExp functions that are already built-in IGOR functions [Figure 2]. The program also provides the ability to create your personal fitting functions using IGOR Pro's built-in programming language. By simply typing in arithmetic expressions, IGOR can now analyze specific data to the exact parameters of the user. IGOR's basic curve fitting capability offers automatic calculation of the model curve, curve fit residuals, and confidence and prediction bands. These curves can be automatically added to a graph of your data. Each function has an idea of curve fitting based on finding a mathematical model that fits the inputted data.

---

[1] P. Chen and C. Joyner, "Peak Separation and Sorting by Coherent 2D Resonance Raman Spectroscopy," *Analytical Chemistry*, vol. 77, pp. 5467-5473, June 2005.

The following graph illustrates the different types of mathematical approaches that can be taken:
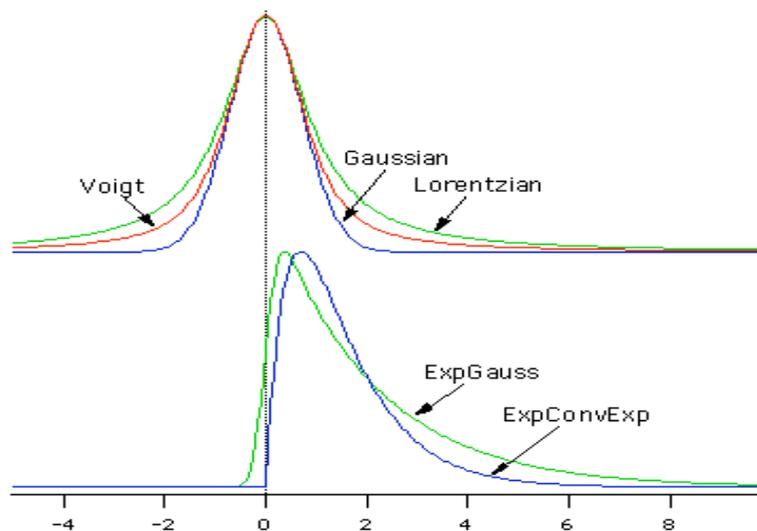
[2] **Figure 2**

In one dimension, the Gaussian function is the probability function of the normal distribution,

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \, e^{-(x-\mu)^2/2\,\sigma^2},$$

[3] . **Figure 3**

One problem is determining which peak-fitting function best matches the spectroscopy graph. For the Chemistry data that we were analyzing, the vital information at the highest peaks was of most concern. Each peak-fitting function has to be tested against each graphed peak to find the best matching fit for the entire graph. Once a match is found, then the actual data can be read and analyzed in its proper form. When using the numerous peak-fitting functions of IGOR, the shape of each peak graphed was taken into concern. For example, if the Gaussian function [Figure 3] was used to fit a tall, slender peak, then a portion of the top part of the peak will eventually be cut off when just looking at the fitted graph. In this case, Lorentzian, Voigt, ExpGauss, and ExpConvGauss would definitely have to be tested because Gaussian just filtered out information that could have been of essential value to complete the analysis of the data that was inputted.

Peaks are found in numerous types of data. The best use for these peak-finding algorithms would be to apply them to two-dimensional graphs. The graphs are already used to decongest one-dimensional data and the algorithms will further provide the analysis for this data. For better performance and efficient use of time, an automatic peak-finder would be created with the selection of more than one function. With the click of a button, a graph could be analyzed to tell the number of important peaks that were found in any shape possible. For future use, new two-dimensional peak fitting functions could be implemented in IGOR since the Gaussian function could also be implemented in two dimensions. Rather than using one-dimensional functions on two-dimensional graphs, just use a two-dimensional peak-fitting function.

[2]"Multi-peak fitting." From *IGOR Pro Manual* –technical graphing and data analysis software
http://www.wavemetics.com/products/igorpro/dataanalysis/peakanalysis/multipeakfitting.htm

[3] Eric W. Weisstein. "Gaussian Function." From *MathWorld*—A Wolfram Web Resource.
http://mathworld.wolfram.com/GaussianFunction.html

Utilizing Clip Planes to improve the clarity of rendered images

Danielle Edwards
Computer Science/Computer Engineering
dedward4@spelman.edu
Spelman College

Presented at the 2006 CenSSIS REU summer internship presentations
Advisors: Dr. David Kaeli (Northeastern University)
Mr. James Hale (Spelman College)

ABSTRACT

Clarity, in this context, is defined as the ability to clearly visualize cancer tumors in moving parts of the body. Most medical images are rendered using a gray scale; meaning, different portions of the image are drawn in different shades of gray according to their densities. This can become problematic because oft times the tumor and the surrounding tissues have the same density therefore having the same gray scale value causing both the tumor and the surrounding tissues to be rendered in the same shade of gray. This causes tumors to become harder to locate on medical images and even more difficult to visualize accurately when found. This, in addition to the fact these tumors are in moving parts of the body, such as the lungs, greatly increases the risk of oncologist irradiating healthy cells.

A clip plane allows the oncologist to remove all obstructions that block the thier line of sight to the tumor, such as tissue with the same gray scale value, while leaving portions of the image intact to reference the position of the tumor

Accessible Accessibility:
Javascript Application for Mobile Accessibility Solutions
Drew Fleming
Roanoke College
Salem, Virginia
ajfleming@roanoke.edu
Advisor: Dr. Anil Shende, Computer Science

In the summer of 2006, I set out to create software that can help visually impaired individuals by writing software that will allow web content to be parsed in such a way that a basic screen reader can audit the content more efficiently and accurately. This idea came about though a literature survey which illustrated a few things very clearly. Firstly, as Jennifer Mankoff [2] pointed out in a recent research paper, developers without accessibility experience have a difficult time developing and testing the pages for accessibility. Also, automated tools that attempt to make content accessible have had little success[2]. Given this, there has been a need for desktop applications such as Homepage reader by IBM[1]. This is software that is installed on one's computer that can read the text aloud via the computers speakers. This software has advantages over the traditional screen reader because it provides more options to users as well as having screen reading capabilities to other Microsoft Windows applications. In addition to Homepage Reader, there has also been the development of several other standalone applications such as JAWS. However, there is not a way for a visually impaired user who may not have Homepage Reader or JAWS to use a computer in an efficient way. For this reason is why I decided to develop software that can allow users to use the Internet efficiently on any computer.

The idea for a piece of software written in Javascript seemed like an easy solution to the problem. This would allow users to be able to get accessible content from just about any web page via web based software. A basic screen reader (most modern operating systems have at least one built in) could do the job of auditing the parsed content to the user as long as the computer he or she is using has a sound card and speakers. My approach to this problem was fairly straight forward. A basic web page with one text area located at the top of the page was created. This web page contains an invisible division at the bottom of the page. The user then enters a URL and a php function goes and grabs the HTML code from that page (with a GET function call). This text is returned to a local variable in a Javascript file and several functions parse out content by removing the actual code from the page. Removed content is based on whether or not it slows down a visually impaired users mobility such as text decorations. Also, the colors of the page are changed to a dichromatic contrast and a larger text size that can be read more easily. This feature is especially useful to users with low vision. Images are also replaced with their alternate text (if available).

To add further to the increased mobility, there is an option to section up the page based on certain tags such as lists and tables. This feature was implemented to help decrease the amount of time it takes to read a page. According to a recent study by Chieko Asakwawa [3] petitioning developers to increase audit speeds for their screen readers, it was noted that the most comfortable listening speed for a blind user is between 800 and 1000 words a minute. It was also noted that most screen readers do not support speed above 900 words per minute and thus, if a blind user is trying to find a section of text that is 2000 words away on the page, he or she would have to sit through 2 minutes (at the least) of useless information. Because most websites will use lists or tables to divide their pages in to sections such as links of interest, and main articles, the content of the page can be skipped though allowing the user to find what he or she is looking for more quickly. A feature that is slated to be implemented will be a way for users to use the

program through their entire web journey. This does not work now because hyperlinks are not set up to call the Javascript function when they are clicked because they reside inside of a division which cannot 'see' the Javascript file containing the functions. This should be a fairly easy fix and should be implemented soon. Also, a table of contents feature, a table of links, and moving the main content to the top of the page are currently in the works to be implemented.

There are however, several additional inherent drawbacks to this solution, even if all were to work as planned. Firstly and probably most obviously, the speed of the program is dependent on the status of the sever its hosted on and its current traffic as the file has to be downloaded to be executed, which could exhaust the server if many users are connected. This can be a serious drawback as the intention of the program is designed to allow visually impaired users an easier, faster journey on the web. Secondly, there are limitations of Javascript that will not allow this program to ever be as useful as some of the commercial desktop application such as JAWS or homepage reader, mainly the fact that a commercial application can be several hundred megabytes in size, which is not practical for downloading upon every visit to the page. Finally, inclusion of screen readers and speakers on all publicly available computers may not be as widely available as was originally planned.

Even given these issues, the software still has promise and once the current issues are resolved, the software will be tested for usefulness. The best way to test this software is to let visually impaired users try it out and get feedback from them. The Virginia School for the Deaf and Blind and myself have been in contact and we will be testing once development is complete. Regardless of the usefulness of the software, insight was gained in the issues in and surrounding the field of web accessibility for visually impaired individuals.

References

[1] Hanson, Vicki L. The User Experience: Designs and Adaptations
In SIGCAPH Comput. Phys. Handicap, New York, NY, USA., 2003.

[2] Mankoff, Jennifer. If Your Webpage Accessible? A Compative
Study of methods for assessing web page accessibility for the blind.
In Proceedings of SIGCHI conference on Human Factors in Computer Systems.
Portland, OR, USA., 2005.

[3] Takagi H. et al. Asakawa, C. Maximum listening
speeds for the blind. In Proc. Conf. of International
Community for Auditory Display, Ivory, M.Y., 2003.

Study of Mobile Agent Algorithmic Implementation and Development
Christopher Smith
Roanoke College
Salem, Virginia
cssmith@roanoke.edu
Faculty Sponsor: Dr. Anil Shende, Computer Science

1 Introduction

A mobile agent is a piece of software that can move itself from computer to computer over a mobile agent enabled network. These agents are completely autonomous, performing their set tasks and then reporting the results to the end user. For this system to work, all of the computers that the agents are going to be executed on must be running a mobile agent protocol.

Aglets is a Java based mobile agent protocol. It allows for the creation of mobile agents, which can then perform various tasks. The development of Aglets is an on-going process, since Aglets is an open source program. It is complete enough to start developing applications that can use these mobile agent protocols.

Along with the development of the mobile agent protocols, various algorithmic issues have been identified in the literature ([1], [2], [3]). These issues have been studied, and in some cases provably correct algorithms have been developed to alleviate the issues. Nowhere in the literature, however, has anyone stated that the newly developed algorithms have been implemented into a real mobile agent protocol. A complete literature survey appears in the full paper.

The purpose of this project was to implement one of the more prominent algorithms that has been developed, the black hole search algorithm ([2]), into a mobile agent protocol, Aglets. We implemented the black hole search algorithms for a ring network ([1]) and a general 2-connected network ([2]). The purpose of this was to see how easily the new algorithms could be implemented into real-world scenarios, as well as testing the algorithmic claims of the algorithm creators.

The full text of this paper tackles a couple of topics. First, it is written as a tutorial for someone to follow in order to start developing mobile agents in the Aglets protocol. Second, it describes the process taken in implementing the algorithms. Finally, it discusses our on-going work on algorithmic issues with mobile agents.

2 Results

The install of the Aglets software was very straightforward. Other than the few intermittent problems we encountered, we were able to get Aglets installed on all of the servers in one day.

The Black Hole algorithm is fairly simple, but it relies on some features that the Aglets protocol does not have. The most notable of which is the whiteboard. The mobile agent model used by [2] requires a place on each server that agents can read from and write to. This is called the whiteboard. While in the case of Aglets, it was very simple to create and implement a whiteboard into the protocol, this may not be the case with other protocols. It would also be easy to create an agent on each server to act as a whiteboard, but in a real world situation, either the whiteboard agents need to be executing all the time, even when not performing the black hole search, or they have to created once an agent successfully enters a server.

Another feature that the Aglets API did not provide that would have been useful for our algorithms would be a way to detect other servers that are in communication with the current server. While this would not have helped the ring network algorithm, this would have helped out a lot in the more general algorithms. In the algorithms that we chose to implement, we were able to work around not having this feature through the DNS functionality we programmed, along with the knowledge the agent has about the network. The other algorithms have considerably less knowledge about the network, hence cannot create their paths, nor do they have a way to figure out what servers they can travel to. They required some feature such as this. We would have to program in this additional functionality ourselves.

The rest of the algorithm was fairly easy to implement. The Aglets API provides a lot of useful functions and classes. After discovering the power that Aglets provides to the programmer, it was fairly easy to program the algorithms. A full comparative analysis of the published algorithm and the implementation appears in the full paper.

3 On Going Research

In [2], they present an algorithm to find one black hole using two agents assuming the network graph is 2-connected. In that algorithm, each agent incrementally explores a disjoint part of the network, marking nodes to be safe. That algorithm depends on the existence of a decomposition (shown in [2]) of the unsafe region of the network into 2 partitions with certain properties.

We are currently working on a general solution for locating n black holes. Our algorithm finds n black holes using $n + 1$ agents assuming the network graph is $n + 1$ connected. Similar to the above algorithm, our algorithm depends on the existence of a decomposition of the unsafe region into $n + 1$ partitions with certain properties. We are currently working on constructively proving the existence of such a decomposition. We can show that, if such a decomposition exists, then one of the $n + 1$ agents of our algorithm will correctly locate n black holes.

References

[1] Stefan Dobrev, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Mobile search for a black hole in an anonymous ring. *Lecture Notes in Computer Science*, 2180:166–??, 2001.

[2] Stefan Dobrev, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Searching for a black hole in arbitrary networks: optimal mobile agent protocols. In *PODC '02: Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pages 153–162, New York, NY, USA, 2002. ACM Press.

[3] Nicola Santoro. Mobile agents computing: Security issues and algorithmic solutions. In Mario Coppo, Elena Lodi, and G. Michele Pinna, editors, *ICTCS*, volume 3701 of *Lecture Notes in Computer Science*, page 22. Springer, 2005.

Natural Language Database Interface Portability

Hampton Smith

Roanoke College

Salem, Virginia

hwsmith@roanoke.edu

Faculty Sponsor: Dr. Adrienne Bloss, Department of Computer Science

Natural Language Processing is the study of computerized systems capable of understanding or generating utterances in a natural language such as English. Natural language was first applied to database querying in the early 1970's as a way of overcoming the unintuitive interface and high learning-curve of formalized querying-languages such as SEQUEL (now SQL) [4]. Such early systems, like the 1973 LUNAR system and the 1978 LADDER system were created for research purposes, and were mostly hard-coded for their specific purpose [1]. However, wide-spread acceptance of these Natural Language Database Interface (NLDBI) systems has never been achieved, in part because the task of adapting an existing NLDBI to a new database ("porting" the system) is complicated and labor intensive [3]. Many systems exist, and still more have been proposed, but most fall into one of two categories: production-grade systems, which provide maximum flexibility, but are often heavy-weight and require detailed knowledge of natural language processing; and ad-hoc systems, such as ELF [2], which are intended for non-technical users and attempt to analyze a given database and generate an interface automatically.

The goal of this study is to develop a flexible, light-weight, and portable NLDBI that translates English queries into formal SQL queries and that is targeted for use by individuals— such as database designers—who are familiar with the vocabulary and functioning of databases, but have no background in natural language processing.

The backbone of the system is an attribute grammar parser implemented in Common Lisp that can accept a given grammar and match input utterances against it, synthesizing attributes as it parses. This is used as the basis for translation, with attributes being used to map English expressions used in certain contexts to equivalent SQL expressions. This method keeps the system light-weight since it bypasses the common step of reducing the natural language utterance down to some logical representation, relying rather on the structure of the parse tree to indicate semantics.

Since the primary difficulty in NLDBIs is that no one-to-one mapping exists between English utterances in general and SQL queries, the system dynamically generates the translation grammars used for mapping English to SQL based on a customization file created by the database designer for the database in question. The structure of this customization file is designed around concepts which should be familiar to an individual possessing a facility with databases (such as columns and rows of tables), while avoiding concepts which are specific to natural language processing. By creating this customization file, the database designer describes the structure of the database and maps English words and expressions to that structure.

By way of example, here is an excerpt from such a file:

```
attribute "employees"
        source table = "S"
        primary key = "NAME"
        field name = "EMPLOYEES"
end attribute

entry "parts"
        source table = "S"
end entry
```

The translation from the customization file to the dynamically generated translation grammar is itself accomplished using a hard-coded meta-translation grammar, and once a specific customization file is compiled, it need not be compiled again.

The final product of this study was a complete, though basic, NLDBI which allows a database-savvy individual to describe a database, and then have a translation grammar automatically generated. It includes a front-end into which a user can type an English query and have an equivalent SQL query returned. Creating a customization file requires only an extremely limited understanding of natural language processing. Since the parsing and the translation occur in a single pass, the system is extremely light weight and well suited for creating a simple NLDBI quickly with little technical knowledge of parsing, understanding, or translating natural language, while still allowing flexibility in how the database is described and the translation implemented. Furthermore, since the compilation from customization file to translation grammar is itself accomplished with a translation grammar, the basic capabilities of the system are easily broadened to encompass more complicated database structures.

Since the system skips rendering of the English utterance into some logical representation, it lacks much of the robustness available in other systems. Similarly, since the semantic understanding of the English query is based on the structure of the parse tree, rather than a logical representation, anaphora resolution is nearly impossible. More work is required to allow the system to be useful for creating more sophisticated interfaces.

REFERENCES

1. FA Barros, A DeRoeck. Resolving Anaphora in a Portable Natural Language Front End to Databases. Proceedings of the fourth conference on Applied natural language processing, Stuttgart, Germany, 1994. Published online at the Association for Computing Machinery, www.acm.org. Accessed February, 2006.

2. Elf Software Corporation. www.elfsoft.com. Accessed November, 2006.

3. JS Kaplan. Designing a Portable Natural Language Database Query System. ACM Transactions on Database Systems (TODS), 1984, vol. 9, issue 1, pages 1-19. Published online at the Association for Computing Machinery, www.acm.org. Accessed February, 2006.

4. Y Li, H Yang, HV Jagadish. NaLIX: an Interactive Natural Language Interface for Querying XML. Proceedings of the 2005 ACM SIGMOD international conference on Management of data, Baltimore, Maryland, 2005, pages 900-902. Published online at the Association for Computing Machinery, www.acm.org. Accessed February, 2006.

Machine learning on an AIBO robot

Jackie Soenneker
Centre College
Danville, KY
jackie.soenneker@centre.edu
Faculty Sponsor: Dr. Christine Shannon, Computer Science Department

Reinforcement learning is a type of machine learning that involves an agent (in our case, a robot) who exists in an environment and has a goal. The agent chooses its next action based upon its state, a description of the relevant aspects of its environment, and is given a reward when it takes that action. This reward can be either positive or negative, depending on the desirability of the new state into which the action moves the agent. Over time, the agent uses these positive and negative rewards that it receives to build up a policy. This policy is a function of the state and an action, and gives the probability that the agent chooses that action when in that state. By using this policy to decide which actions to take, the robot tries to maximize its total reward and reach its goal. Another feature of reinforcement learning is the balance that is sought between exploration and exploitation. When the robot is choosing actions exploitatively, he is choosing the action that experience tells him is best. Since this can easily lead to overlooking a better choice because the robot has no experience with that action, reinforcement learning programs also do some exploration by randomly choosing actions some percentage of the time. Our research, though focused on reinforcement learning, also used a method called Gradient Descent to have the robot optimize his performance by improving the values of certain parameters.

In our research, we used a Sony AIBO ERS-7 robotic dog. The robot had many sensors, but we almost exclusively used the camera mounted in the robot's nose. To communicate with the AIBO, we used www.pyrobotics.org's Pyrobot program. Vision-processing software permitted us to add filters to the camera image to detect the bright pink of the robot's ball. Our program could then respond to the coordinates of the center of the largest blob of this color. In response to input from the camera, the robot could move his head up, down, left, or right, move forward or rotate. Our Gradient Descent experiment was designed to determine the optimal settings for the size of these movements.

We conducted two types of experiments in our research. In both experiments AIBO and his ball were positioned on the floor, and AIBO would attempt to visually locate and then walk toward the ball, stopping and wagging his tail when he neared it. In the Gradient Descent experiment, the program started off with a base set of values for the magnitudes of the various movements (the "base policy") and randomly generated "n" new polices/sets of parameters by randomly incrementing, decrementing, or leaving untouched each of the parameters in the base policy. Each of these new policies were tested by AIBO three times. During testing, each episode would begin with AIBO and his ball always positioned in exactly the same places and AIBO would look for and move towards his ball. A record was kept of how many actions ("steps") were required to find the ball under each policy. Using these results, the program calculated an adjustment vector that was added to the base policy to form a base policy that was presumably an improvement on the values of the parameters. Another set of "n" random policies was generated from the new base policy, tested, used to calculate a newer base policy, etc. This continued and the values of the parameters gradually changed so as to produce improved performance.

This program was quite successful. Under the original base policy, over ten trials it took an average of 196 steps to find the ball. In the best base policy gained after twenty-one iterations through the "generate, test, create new base policy" process, AIBO required an average of only 93 steps to reach the ball, less than half of the original number of actions required to complete the task.

In the second experiment, we used Temporal Difference learning, a type of reinforcement learning. The central part of our TD learning programs was the Q function, which is defined by

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\}$$

Our program implemented this as a large table. When supplied with the state AIBO was in and the action AIBO was going to take, Q returned the current estimate of the total future reward for taking that action in that state. Use of this function allowed AIBO to estimate the best action to take in a given situation and to predict the reward that taking that action would produce. When the action was taken and the actual reward obtained, the Q function was updated to more accurately predict for that state and action. Our TD learning programs had two "modes" in which AIBO could be run. The first was a teaching mode in which AIBO was told exactly what to do by the program and made no decisions. The purpose of this mode was to populate the Q function's table with data. In the second mode, the "learning mode", AIBO decided what to do based upon the data in the Q function that was gathered in the first mode. As the name implies, it was in this second mode that AIBO did his actual learning by making choices with the Q function and updating the Q function as rewards were given, slowly improving his ability to correctly predict the best action in a given state. Also, in order to have AIBO explore new actions, a certain percentage of the time in learning mode, AIBO chose his next action at random.

We had two different versions of the Temporal Difference learning program. In the original version, the state consisted of four pieces of information: AIBO's horizontal and vertical head positions and the horizontal and vertical positions of the ball in AIBO's view. This version rewarded AIBO for finding or catching sight of the ball, and punished him for every action required to find the ball, for losing sight of the ball, or for trying to center the ball in his view and his body with his head when he was already centered. This version had a twenty-percent randomness factor in its learning mode. In the second version of the program, we decided to try to improve AIBO's learning by giving him some more information. In this version the state included all the elements of the first state, plus information about the last action taken and the size of the ball in AIBO's view. We also extended the reward/punishment mechanism, and added punishments for trying to center upon a ball when no ball was in view and for doing the opposite of an action immediately after doing that action. In this version of the program, the randomness factor was five percent.

The results from the Temporal Difference programs were also fairly encouraging. The first version of the TD learning program was something of a prototype that underwent continual changes and was never thoroughly tested, but the second version of the program was a bit more stable and went through more testing and learning. When the second program first switched from teaching to learning mode, AIBO was requiring an average of about 360 steps to find the ball, while after about 225 learning-mode episodes the average path-length of all the episodes had fallen to around 220 steps, two-thirds of the original path-size.

This research was a continuation of previous research, and we intend for this to become an ongoing project in which we and others will continue to explore applications of these and perhaps other means of machine learning.

Sources that we used in our research include the following:

Kohl, Nate and Peter Stone. "Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion." *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA 2004). Pp. 2619-2624, New Orleans, LA, May 2004.

Smart, William D., and Leslie Pack Kaelbling. "A Framework for Reinforcement Learning on Real Robots." < http://www.cse.wustl.edu/~wds/content/papers/aaai1998/aaai1998.pdf>.

Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction*. London: The MIT Press, 1998.

Multi-Dimensional Chromatography Informatics

Katrina Stewart (kstewart@spelman.edu, Spelman College)

Ashley Johnson (anjohnson@spelman.edu, Spelman College, research partner),
Dr. Alfred Watkins (alwatkins@spelman.edu, Spelman College, advisor),
Dr. J. Dimandja (jdimandja@spelman.edu, Spelman College, advisor)

Student Session Abstract

KEYWORDS: Gas Chromatography, Database, Visualization, Kovat's Index, Retention Time, Analyte

Chromatography is an analytical method in which the constituents of a sample are separated and identified. In Gas Chromatography the sample is in a gaseous form. Gas chromatography is able to analyze any sample that can be volatized, or formed into vapor. Gas Chromatography is a popular instrument of analysis in many industries, but can be cumbersome and time-consuming in research endeavors.

Currently, the initial data is stored in text documents (or .txt. files). This data includes retention time for a control mixture and time data for each compound in the sample. Next data is analyzed manually, manipulated, and placed into 2-dimensional charts using Microsoft Excel. To do this, a lengthy equation is used to calculate the retention index (KI) for the compounds in the sample. This equation must be applied to every compound in the sample. Because there are often dozens of compounds in a compound, this is a very lengthy and cumbersome process. After the KI values for each compound in a sample are found, they are compared with known KI values to identify the compounds. This is done through graphical analysis using the knowledge of thousands of known KI values. Because of the large number of known KI values this can also be a lengthy process. The goal of this research is to automate the manual process, store experimental data in a database and generate visualizations of the data viewable via a website. A database application would automate this process and could make the data more accessible by providing functionality over the Internet within a web interface.

Using MySQL and PHP we will develop software that enhanced the data processing capabilities in GCxGC method development. This project has been broken down into three parts. The initial stage involved the development of data processing software that provided the retention index numbers for the analytes in a one-dimensional GC run. In the second stage visualization software that will display two-dimensional retention index plots will be developed. Finally, there will be the development of data processing visualization software that will provide fingerprinting information between GCxGC chromatograms.

Image Tracing and Tracing Server Development

Marisa Toodle
Spelman College
Atlanta, GA
mtoodle@spelman.edu
James Hale, Computer Science

The first project involved validating a tracing program that was developed by Alex Tyrell, a graduate student at Rensellear Polytechnic Institute. He created a program that used a model to conform to the complex vasculature of vessels. This program takes medical images (in this case, of tumors) acquired using fluorescence microscopy, and segments 3D tubular structures such as vessels. A model, which is a superellipsoid, expands and contracts to accommodate the complex vasculature of vessels. The validation of the program was checked by running it on several images, and once the tracings were finished, find all errors.

Errors of the program usually fit into one of three categories: noise, boundary errors, and random tracings. Noise errors are tiny dots scattered throughout the image caused by light and artifacts. Random tracings are tracings that appear in blank areas. This is the least common, and boundary errors are the most common. Boundary errors are when the boundaries of the tracings are incorrect. They may be greater than or less than the actual vessel boundary.

After removing all errors, a test was done on the tracings to get the accuracy of the program, and the average was 93%, with 96% being the highest and 92% being the lowest. With this accuracy, there should be a way to allow users to upload images of their own and show the results without us being the middle man, thus creating my second project: the development of an image server. The website should allow users to upload images from their computer, open Matlab, and run the tracing program on the images, and once the tracing is done, display the results on screen and save them to a folder. Since there were various updates and installations that needed to be made to just get a basic webpage to work, I am currently working on the completion of this project.

Behavioral Based Visual Queries
Jessica Young
Roanoke College
Salem, VA
jdyoung@roanoke.edu
Faculty Sponsor: Dr. Stephen Hughes, Mathematics, Computer Science, and Physics

People regularly engage in decision making using multi-dimensional data. For example, when buying a new car there are several factors (MPG, size power, price, etc) that people will want to examine. Although multi-dimensional data like this is ubiquitous, analyzing it can be extremely difficult. The basic way to organize this information is to use a table format; however, working with large amounts of data quickly makes the table organization unwieldy. People find it difficult to extract meaningful information from the tables because they experience information overload. It is possible to extract this information from the data through the use of a specialized query language, but problems still exist. Query languages often require an aptitude for logical relations and explicit training on the syntax of the language. There have been some attempts to simplify the input for these queries using sliders [1, 2]. However, no amount of training can overcome gaps in the expressiveness of the query language. For example, there is often no easy way to query a database to get an overall understanding of similarity within the dataset.

Another approach to dealing with multi-dimensional data is the use of visualization. While visualization allows the viewer to quickly gain an overview of the data, the display is limited to two- or three- dimensions. This means that the viewer must deal with imprecision brought on by aggregating or culling data to be displayed. Furthermore, the interpretation of a visualization is largely subjective as it requires the viewer to make personal judgments. This is especially true when the display is statically created using pre-established parameters instead of user-defined input, for example dendrograms [3].

An interesting alternative is to generate dynamic visualizations using Dörner's interactive behavioral animation framework [4]. In this model, each row of a database table can be represented by a visual data avatar that is programmed with certain behaviors. This research explores how users can specify queries by using familiar interface components (e.g., sliders) to introduce stimuli to a collection of avatars. As each avatar reacts to the stimuli and other avatars, emergent behaviors can be interpreted as a query result.

To demonstrate this concept, an interface was constructed that explores the effects of two basic behaviors. Specifically each avatar is directed to move toward "similar" avatars and to move away from dissimilar avatars. The interface enables two types of queries on multi-dimensional data.

The first type of query allows users to look at the avatars that are the most similar to one another. To do so the user defines similarity by expressing an acceptable degree of noise; those elements within this threshold move together forming clusters. By dynamically adjusting the threshold the viewer can gain an appreciation of the different levels of similarity that exist in a dataset according to the user's preferences as to the type of analysis. Furthermore, the viewer is able to further refine and weight which attributes should be considered in the similarity function. This flexibility is a strength of the interface because it allows the user to analyze the data as he or she wishes.

The second query type isolates how groups of data compare to specified criteria. The user sets criteria points that define a specific range of values for a given dimension. Those avatars with the ideal values as defined by the range will gravitate towards the criteria point. By using multiple criteria points, groups form based on which criteria are met. These groups can be interpreted in a manner similar to that of Venn diagrams (Figure 1. note: the circles are added for the purpose of discussion and are not present in the actual interface). For example, with two criteria points, those avatars closest to point a satisfy its criteria; those closest to point b satisfy its criteria; those avatars between the two points satisfy both points' criteria, and the last group satisfies neither.
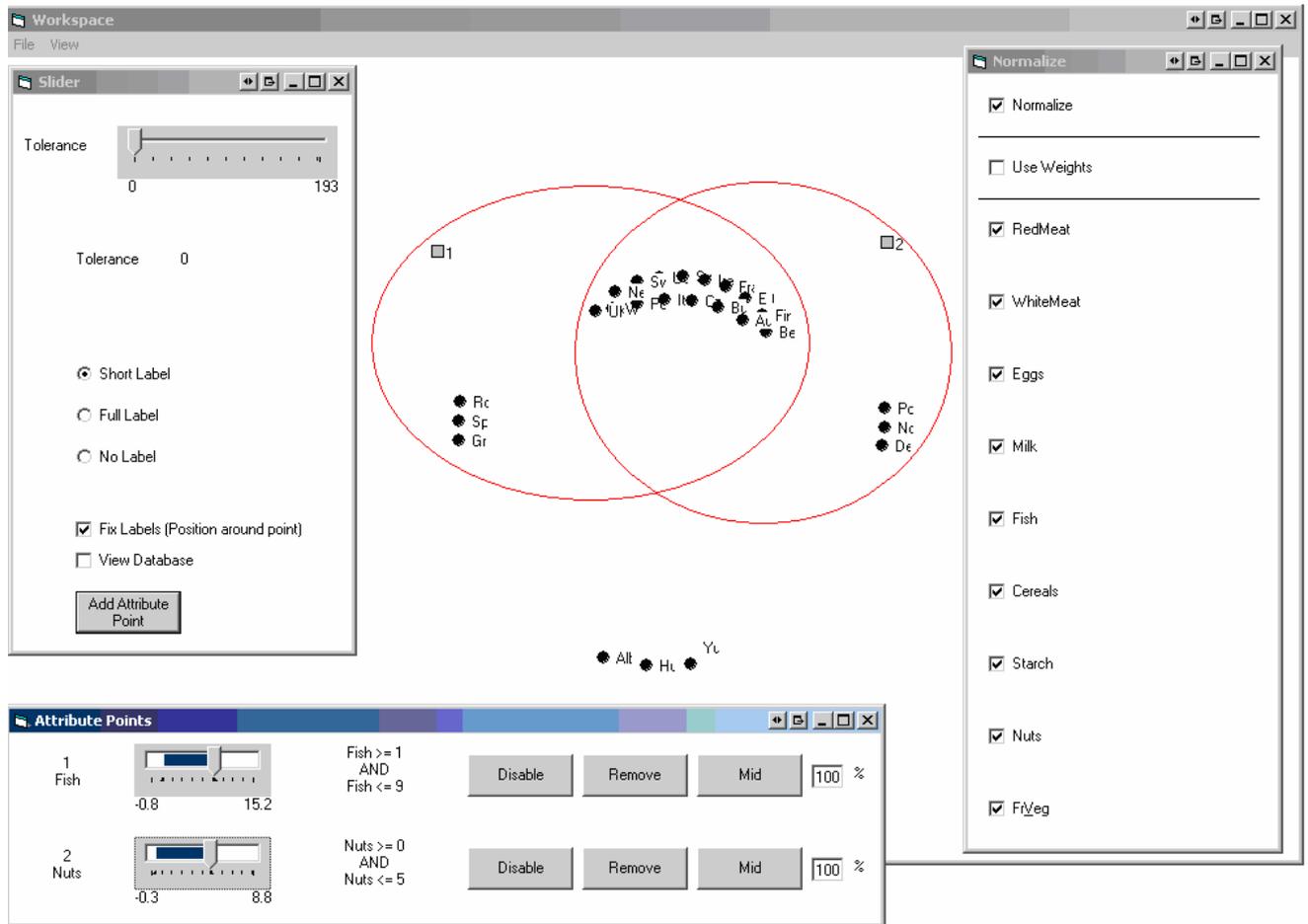
Figure 1: User query and Venn diagram

Behavioral based visual querying provides a better approach to examining multi-dimensional data; it supplies a balance between the explicit details of query languages and the big picture of visualizations. It uses familiar interface components to express queries in a flexible manner that does not require the knowledge of a query language. The avatars are driven by simple behaviors that give the viewers an objective way to express both similarity and criteria queries.

There is currently still work that can be done on this project. The interface will need to be tested to see the user's understanding of the results. There are plans to evaluate the effectiveness of the tool for physics students working with an astronomical dataset. The results of the testing will likely suggest further refinements based on user needs. Another plan is to look into different methods for calculating the similarity between the points.

References

1. Ahlberg, C. and B. Shneiderman. *Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays*. in *ACM Conference on Human Factors in Computing Systems*. 1994. New York.
2. Eick, S.G., *Data visualization sliders*, in *7th annual ACM symposium on User interface software and technology*. 1994: Marina del Rey, California, United States p. 119-120.
3. Aldenderfer, M.S. and R.K. Blashfield, *Cluster Analysis*. Quantitative Applications in Social Sciences, ed. M.S. Lewis-Beck. 1984, Newbury Park: Sage. 88.
4. Dörner, R. and C. Ware, *Visual interactive stimuli techniques: interactive tools for exploring data using behavioral animation*. Journal of Visual Languages and Computing, 2004. **15**: p. 161-181.