To the Graduate Council:

I am submitting herewith a dissertation written by Joshua R. New entitled "Visual Analytics Techniques for Interactive Exploration of Correlation Data". I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Dr. Jian Huang, PhD Advisor

We have read this dissertation
and recommend its acceptance:

Dr. Elissa Chesler

Dr. Michael Langston

Dr. Lynne Parker

Accepted for the Council:

Carolyn R. Hodges, Vice Provost
and Dean of the Graduate School

(Original signatures are on file with official student records.)

Visualization Techniques for Interactive Exploration of Correlation Data

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Joshua R. New

May 2009

Dedication

First and foremost, I would like to thank my heavenly Father, through whom all things are possible. I would like to thank the several people in my life who have taught me important life-altering lessons of which I can only name a few here: My father, Kenneth New, for cultivating in me at an early age a curiosity and desire for understanding; Ray Brannon for taking the time to thoroughly teach me about computers into the early morning hours as well as serving as a model of an admirable man; Bubba Ogle for the years of patient instruction in the art of volleyball which has acted as a source of stress relief, accomplishment, and most importantly friendship with so many good people; and my grandfather, Vernon Hayes, who has served as an ever-consistent model of a good helper, husband, and Christian. I also thank my family for their support and good will.

Acknowledgments

I would like to begin by expressing my gratitude to the faculty of The University of Tennessee's Department of Computer Science for their excellent quality of their classes during my program. It has truly been an awe-inspiring and humbling experience to witness the quality of the students and education at this institution.

In general, I would also like to thank my PhD Committee members for their intellectual contributions, guidance, and feedback. In specific, I would like to thank the members of my PhD committee for accepting me as a PhD candidate and each for specific reasons: Dr. Jian Huang, my PhD advisor, for providing an engaging atmosphere in which to do research, his financial support through the years, exciting job opportunities, as well as exposure to many of his contacts at the forefront of visualization; Dr. Elissa Chesler for introducing me to the exciting new field of systems genetics, the intriguing problems to which computational capabilities can be leveraged for understanding biology, and the many great interdisciplinary conversations involving new domain-specific visualizations to attempt; Dr. Michael Langston for allowing me to work cooperatively with his very capable team of students, paraclique results, and his unyielding barrage of knowledge in all things computational and graph theoretic; and Dr. Lynne Parker for the best AI and machine learning courses I've ever taken, her truly polished and engaging teaching style, and her feedback on the learning systems of this study.

Along with Dr. Jian Huang and Dr. Langston, I would also like to acknowledge the University of Tennessee's Department of Electrical Engineering and Computer Science for five years of financial support. Counting the $16,500 yearly stipend, UT/ORNL summer internships, and the $9,000/semester out-of-state tuition costs, I am very grateful to acknowledge this contribution of nearly $150,000 to my education.

There are several acknowledgments I should make in reference primarily to the software developed in this thesis. Thank you Dr. Chesler et. al. for the microarray data from the many ongoing studies; Dr. Langston, Jon Scharff, John Eblen, Yun Zhang, Andy Perkins, Jeremy Jay et. al. for the pearson correlation, paraclique results, other biological data processing, linkage disequilibrium work, and visualization system feedback; Vivek Philip for friendly discussions on systems genetics, possible software improvements, and taking over some of the visualization; Shawn Ericson for several sample test datasets from his Java graph interaction package; Arne Frick for the original GEM3D layout code; Pauulius Micikevicius for his GPU-based all-pairs shortest path code; and many others for their feedback, direction, ideas, and contacts.

Abstract

Domain scientists hope to address grand scientific challenges by exploring the abundance of data generated and made available through modern high-throughput techniques. However, the impact of this large volume of data is limited unless researchers can effectively assimilate the entirety of this complex information and integrate it into their daily research; interactive visualization tools are called for to support the effort. Specifically, typical scientific investigations can make use of novel visualization tools that enable the dynamic formulation and fine-tuning of hypotheses to aid the process of evaluating sensitivity of key parameters and achieving data reduction. These general tools should be applicable to many disciplines: allowing biologists to develop an intuitive understanding of the structure of coexpression networks and discover genes that reside in critical positions of biological pathways, intelligence analysts to decompose social networks, and climate scientists to model and extrapolate future climate conditions. By using a graph as a universal data representation of correlation, our novel visualization tool employs several techniques that when used in an integrated manner provide innovative analytical capabilities. Our tool integrates techniques such as graph layout, qualitative subgraph extraction through a novel 2D user interface, quantitative subgraph extraction using graph-theoretic algorithms or by querying an optimized B-tree, dynamic level-of-detail graph abstraction, and template-based fuzzy classification using neural networks. We demonstrate our system using real-world workflows from a couple large-scale systems.

Contents

List of Figures

Chapter 1

Introduction

The amount of digital information created, captured, and replicated in the year 2006 alone was 161 exabytes (billion gigabytes) which is equivalent to 12 stacks of books reaching from the earth to the sun (30 million times more than have ever been written) and is expected to increase six fold to 988 exabytes per year by 2010 [Gantz et al., 2007]. As the amount of scientific data increases, computational tools are necessarily leveraged to traverse from data to meaningful scientific results. The ability to identify interesting features in the data while incorporating inherent uncertainty is a central research problem. This problem incorporates several research avenues which must be addressed in an integrated manner for maximum impact.

First, the role of interactive visualization is increasingly necessary for allowing experts to make sense of large data. For example, the Department of Homeland Security has established the National Visualization and Analytics Center (NVAC) with the purpose of countering future terrorist attacks through the use of visual analytics, which it defines as the science of analytical reasoning facilitated by interactive visual interfaces [Thomas and Cook, 2005] and issued a call for computational tools that enable human-information discourse. One common counter-argument to visualization is that particular analytical processes can be wholly automated, so why the need for visualization? Visualization rests upon the assumption that no matter how good pattern recognition and automation is, the best it can be is semi-automatic within the context of the entire scientific process; there is no magic to jump from fuzzy concepts to fully substantiated and verifiable specifications. The quality and accuracy of an analytic process is a complex tradeoff due to uncertainty innate to the data, its use, as well as its method of representation. For example, which method of normalization is appropriate for a specific type of statistical analysis, which threshold is appropriate for a certain complexity of graph analysis, and how do you map text and missing features to representations amenable for specific machine learning methods? The expert user's domain knowledge plays a vital role in balancing tradeoffs adequately for the scientific question at hand. It is common for such investigations to leverage human-in-the-loop control over an iterative process of data visualization, user input, and computer operations on the data. The unique capabilities of real-time interactivity using modern, hardware-accelerated graphics to take advantage of the human eyes broadband pathway to the brain and widely applicable algorithmic tools are thus called upon to facilite the process of knowledge discovery.

Second, general data structures must be utilized which can incorporate multiple types of data items and uncertainty in the relationship between those items while concurrently being optimized for common methods of interaction. The most common data framework for scientific data is the idea of entities with properties, referred to as multivariate data, which lead to the rise of the spreadsheet and database storage schemas with distinct entities in rows and a list of properties (aka. features or attributes) in columns. In addition to entities with properties, there are often connections or relationships such as correlation that is of interest to scientists. In this work, we use a weighted-edge graph and database as universal data representations of large complex data. The graph is represented as a limited-precision, lower- triangular adjacency matrix for efficient storage, cache performance, and algorithmic simplicity. The database uses a proprietary B-tree format optimized for traditional boolean range queries, unlike traditional database formats. The database can also be used to store and interactively query results from algorithms that are too computationally intense to run in real-time. While domain experts are often familiar with a simulation or experiment, they frequently do not have the proper tools for telling the visualization tools what to show. These general, flexible, and performance-driven data structures thus function as a common basis for interdisciplinary discussion as well as for analytic techniques during real-time visualization.

Third, novel data-driven algorithms should be developed and integrated with existing analytic techniques to exponentially increase data processing capabilities. To make scientific advances, specialists seek out ways to analyze their data until it highlights some new property, gives rise to process insight, or points toward a paradigm shift. Since the ways scientists can alter their data is limited by the vocabulary of applicable algorithms, it is important to leverage existing techniques while also developing more powerful or applicable ones. Based upon our very general data structures, we have integrated several known methods such as common graph-theoretic algorithms [Gross et al., 2004] and image-processing techniques to the graph structure in combination with common techniques such as database querying, statistics calculations, modeling/mining, and artificial intelligence techniques on the database structure. Novel data-driven algorithms include dynamic level-of-detail graph abstraction for operation over multiple scales such as relationships between paracliques [Chesler and Langston, 2005], algorithms for addressing the unsolved problem of optimized axis ranking in parallel coordinate plots [Inselberg, 1985], and a processing method for opening the black box to understand properties learned by an autonomous agent based upon Adaptive Resonance Theory [Carpenter and Grossberg, 1987]. These data-driven algorithmic capabilities can be used synergistically with one another as well as with visualization analytics algorithms.

Fourth, novel visualization and high-level input mechanisms are necessary to allow new and diverse methods of human-computer interaction. In the software package which integrates most of the techniques in this dissertation, we typically represent the data items and relationships as a common node-link graph. In order to do so, a meaningful position in which similar/related items are close together is calculated for each vertex using existing algorithms integrated with improvements in runtime, parameter settings, and generalization to 2D as well as 3D layouts including GEM3D [Bru and Frick, 1996], Fruchterman-Reingold [Fruchterman and Reingold, 1991], and energy barrier [Davidson et al., 2001] graph layout algorithms. While this method of representation is intuitive and allows for very precise viewing and detail control, scientists often need a high-level, birds-eye view of their data. For this, a novel qualitative subgraph extraction technique using a novel 2D user interface based on block tridiagonalization [Bai et al., 2004] to maximize the data displayed while minimizing the screen space required. In visualization, users often don't know what's interesting until they see their data. We incorporate this to quickly process intuitive, semantically rich inputs for identifying objects of interest in combination with LoD vertices for template-based fuzzy classification using neural networks. Domain-specific visualization has been added using image processing techniques for the automatic generation of spectral karyotypes [Schrock et al., 1996].

The specific contributions in data structures, algorithms, and visual interaction contained in this thesis demonstrate powerful ways of integrating algorithmic computation and adaptive machine intelligence in innovative, uncertainty-tolerant visualizations that powerful computers, meaningfully directed by qualitative concepts of human users, can utilize to unravel intrinsic patterns in complex datasets. We demonstrate our integrated system using real-world workflows from a large-scale systems genetics study of mammalian gene coexpression as well as supercomputer-driven climate modeling of large, multivariate data.

In the remainder of this dissertation, I describe relevant domain-specific and technical background work in chapter 2, the application of novel analytic capabilities for systems genetics data in chapter 3, advancements in the context of parallel coordinate plot axis ranking in 4, a new method for understanding the learned capabilities of SFAM [Carpenter et al., 1991] systems in chapter 5, and close with conclusions and future work in chapter 6.

Chapter 2

Background

We will apply the graph-based visual analytics framework to several application-specific domains. While the proposed framework should be general enough for application to nearly any domain, I provide background information on several that are relevant to on-going collaborative research between The University of Tennessee and Oak Ridge National Laboratory.

2.1 Systems Genetics Data

"Making sense of genomics is risky,
But with database builders so frisky
Gene expression in brains
May one day explain
A mouse's obsession with whiskey."
-Poet Laureate of the Neuroscience Program, University of Illinois at Urbana-Champaign, November 27, 2006

Let us consider an analogy familiar to the field of computer science: a variable stored at a location in the main memory of a computer. In genomics, one can consider the entire memory space roughly corresponding to the genome, a location-specific variable as a gene, and the value stored in each variable as the genotype at that location. The value of a genotype is transmitted by each parent. The fact that each location can take on different genotypes is termed polymorphism, since the same genome location for different individuals may hold (parts of) different genes or non-gene DNA sequences.

The entire set of genotypes across the genome defines the genetic makeup of an organism, while a phenotype defines the actual physical properties, or traits, of the organism. Although genetic makeup is not the sole factor influencing an organism's phenotype, it is often a strong causative predictor of the trait. Consider common traits relating to physical appearances as an example. Having exactly the same genotypes, identical twins have strikingly similar appearances (phenotypes), yet due to environmental influences they may not look exactly the same.

It is of great interest to unravel the inner workings of how genotypes influence molecular networks to affect a phenotype such as agility, seizures, and even drug addiction, to name a few. Geneticists have already achieved great success in associating a genotype and phenotype for a trait determined by one gene (i.e. monogenic traits), but much present attention is now focused on traits that are determined by many genes (i.e. complex traits). These traits are continuously distributed random variables and thus referred to as quantitative traits. Linear modeling is used to identify genotypes that predict phenotype values. The location of these genotypes are quantitative trait loci (QTLs) [Abiola et al., 2003]. Detected via statistical methods [Doerge, 2002], QTLs are stretches of DNA highly associated with a specific phenotype, analogous to genetic landmarks which roughly indicate the position of the active gene. QTLs are not defined at very fine granularity; they usually correspond to areas large enough to hold several genes. The genetic polymorphism (genotypes) in neighboring areas of a set of loci, as a group, influence structure and function on both molecular and organismic scales.

For decades, scientists have systematically randomized and then stabilized genetic variation in groups of mice to effectively create a population of clones. These mice, called "recombinant inbred" (RI) strains, function as a reference population which is used by groups worldwide in order to allow a basis of comparison and integration across different experiments [Chesler et al., 2003]. This is very important from a statistical standpoint as it implies that the potential size of the combined datasets is theoretically unbounded, resulting in extremely high dimensional data. Sufficient confidence is currently allowing integration of diverse biological data across levels of scale in an approach related to systems biology, “systems genetics.” This integrative approach for multiscale and multiorgan phenotypic datasets has only become feasible in recent years and relies heavily on statistical techniques, complex algorithms, high-performance computing and visualization.

Let us consider an analogy familiar to the field of computer science: a variable stored at a location in the main memory of a computer. In genomics, one can consider the entire memory space roughly corresponding to the genome, a location-specific variable as a gene, and the value stored in each variable as the genotype at that location. The value of a genotype is transmitted by each parent. The fact that each location can take on different genotypes is termed polymorphism, since the same genome location for different individuals may hold (parts of) different genes or non-gene DNA sequences.

The entire set of genotypes across the genome defines the genetic makeup of an organism, while a phenotype defines the actual physical properties, or traits, of the organism. Although genetic makeup is not the sole factor influencing an organism’s phenotype, it is often a strong causative predictor of the trait. Consider common traits relating to physical appearances as an example. Having exactly the same genotypes, identical twins have strikingly similar appearances (phenotypes), yet due to environmental influences they may not look exactly the same.

It is of great interest to unravel the inner workings of how genotypes influence molecular networks to affect a phenotype such as agility, seizures, and even drug addiction, to name a few. Geneticists have already achieved great success in associating a genotype and phenotype for a trait determined by one gene (i.e. monogenic traits), but much present attention is now focused on traits that are determined by many genes (i.e. complex traits). These traits are continuously distributed random variables and thus referred to as quantitative traits. Linear modeling is used to identify genotypes that predict phenotype values. The location of these genotypes are quantitative trait loci (QTLs) [Abiola et al., 2003]. Detected via statistical methods [Doerge, 2002], QTLs are stretches of DNA highly associated with a specific phenotype, analogous to genetic landmarks which roughly indicate the position of the active gene. QTLs are not defined at very fine granularity; they usually correspond to areas large enough to hold several genes. The genetic polymorphism (genotypes) in neighboring areas of a set of loci, as a group, influence structure and function on both molecular and organismic scales.

For decades, scientists have systematically randomized and then stabilized genetic variation in groups of mice to effectively create a population of clones. These mice, called "recombinant inbred" (RI) strains, function as a reference population which is used by groups worldwide in order to allow a basis of comparison and integration across different experiments [Chesler et al., 2003]. This is very important from a statistical standpoint as it implies that the potential size of the combined datasets is theoretically unbounded, resulting in extremely high dimensional data. Sufficient confidence is currently allowing integration of diverse biological data across levels of scale in an approach related to systems biology, "systems genetics." This integrative approach for multiscale and multiorgan phenotypic datasets has only become feasible in recent years and relies heavily on statistical techniques, complex algorithms, high-performance computing and visualization.

2.2 Climate Data

"Prediction is very difficult, especially if it's about the future."
-Niels Bohr, Nobel laureate in Physics

Climate scientists have very large and ever-growing datasets using ground-truth data from centuries of measurements. However, these datasets have many problems that hinder their effective analysis: varying measurement times, irregular grid locations, quality of measurements (occasionally dependent upon untrained individuals or faulty equipment), inaccurate bookkeeping, differing number of climatological variables over time, a plethora of missing values, and changing standards of measurement over time. These are but a few of the sources of uncertainty inherent in climate data.

To be sure, scientists are always working on ways to remove this uncertainty, collect more accurate measurements with greater certainty, and continue to improve as more advanced technology becomes deployable. Nevertheless, climatologists have been tasked with extrapolating to predict the future of our planet's climate. This is often approached through creating statistical models from the collected data for individual variables to determine the relationships among the variables and incorporating these into complex, number-crunching simulations which can carry the current state of the climate into the future.

Recent work at Oak Ridge National Lab has focused on the quantitative segmentation of the global climate into ecoregions, also known as climate regimes [Hargrove and Hoffman, 2004]. These areas which have been spatio-temporally clustered based on similar environmental factors can have many uses in management, legislation, ecological triage, and comparison of standard simulation models [Hoffman et al., 2005].

Oak Ridge National Lab has several high performance computing resources that are leveraged to keep them on the cutting edge of research and innovation. In the relevant climate work, supercomputers are used to run parallel k-means clustering for ecoregion identification as well as for model-fitting routines to predict one variable in terms of others. This process creates a multitude of data which can be difficult for even an expert in the field to look through efficiently or effectively during the process of knowledge discovery. Visual analytics tools are called for to aid this process.

Building on much related work from statistical plots, we propose the use of parallel coordinates to allow intuitive visualization and interaction with data across any number of dimensions. We also plan to provide automated trend detection algorithms to highlight potential inter- and intra-variable relationships of interest. These trends and data selections are then projected onto a geo- registered map which is the typical method of presentation for climate scientists.

2.3 Graph Representation and Interaction

A graph is a universal concept used to represent many different problems with vertice representing objects of interest and edges representing the relationship between those objects. Visualization is a powerful tool to leverage for decomposing and understanding important graph properties in a dataset. There are many ways to visualize graphs and the most common way involves a layout algorithm which preserves the strenght of the topological relationships in the positioning of the vertices. While more restrictive layouts, such as trees, should be used when possible, this work will address the general case of graph interaction. In relation to this work, we categorize methods to comprehend graph properties as: (i) those solely depending on algorithms, i.e. the algorithmic approach, and (ii) those incorporating human input as an integral component, i.e. the interactive approach. Let us review both approaches in turn.

2.3.1 The Algorithmic Approach

Algorithmic research to automatically compute graph properties of various kinds has been extensively studied. Well known examples include clique, strongly connected components, induced subgraph, shortest paths, and k-connected subgraph. Let us use clique analysis as a representative example. By filtering out edges with weights below a certain threshold, a gene network with high co-regulation should appear as a complete subgraph, or a clique. Hence, it is natural to consider clique analysis in gene expression data analysis.

However, clique analysis is an NP-complete problem. Even though more efficient fixed-parameter methods [Langston et al., 2006] are currently being used, it is still a very time consuming procedure to compute. It is also hard to treat edges with negative weights in the context of clique analysis, so common approaches typically preprocess the graph to convert all edge weights to absolute values. The impact of information loss due to thresholding is hard to evaluate and is further complicated by the presence of noise. While partially resolved by paraclique [Langston et al., 2006] methods in which a few missing edges are acceptable, additional problems are introduced such as the meaning of paraclique overlap which may be handled differently depending on the working hypothesis.

Such shortcomings apply to different graph algorithms in varying degrees, but are generally inherent with graph theoretic analysis. However, this should in no way prevent graph algorithms from being used for suitable problems. From this perspective, it would be greatly advantageous to develop a visual, effective and efficient feedback framework. In this framework, a human expert is enabled to quickly identify imperfect portions and details of the data, and not only remove irregularities but also to significantly reduce the dataset's complexity by interactively constructing various levels of abstraction. The resulting problem space would be more appropriate for graph theoretic analysis to be applied. In fact, some undertakings in visualization research have already adopted similar approaches [Raymond et al., 2002].

Here we note that our goal is neither to accelerate all computation in a scientist's workflow nor replace computation solely with visualization. We hope to develop a visualization framework which allows navigation through gene expression data and segmentation of the appropriate data for further study. In this way, s/he can flexibly choose and apply the right computational tool on the right kind of problem.

2.3.2 The Interactive Approach

Much related work in visualization follows the Information Seeking Mantra proposed by Shneiderman [Shneiderman, 1996]. That is: overview first, zoom and filter, and then details on demand. At each of the three stages, there are a number of alternative approaches, many of which are highly optimized for a specific application. A key driving application in this area has been visualization of social networks [Perer and Shneiderman, 2006].

To provide an overview, the graph can be rendered in the traditional node-link setting or adjacency matrix [Abello and Korn, 2002], and more recently as a self-organizing map [Kreuseler and Schumann, 2002]. When using the common node-link model, it is pivotal to develop a sufficient hierarchy of abstraction to deal with even moderately sized graphs. Solely relying on force directed methods (i.e. spring embedding [Mutton and Rodgers, 2002]) for graph layout cannot resolve visual clutter and may still significantly hamper visual comprehension.

Structural abstraction can be computed either bottom-up or top-down. In bottom-up approaches, one can cluster strongly connected components [Kumar et al., 1999], or by distance among nodes in the layout produced by a spring embedder [van Ham and van Wijk, 2004]. Top-down approaches are often used for small scale or largely sparse graphs in which hierarchical clusters are created by recursively dropping the weakest link [Auber et al., 2003]. More comprehensive systems employ clustering algorithms that consider a number of different node-edge properties [Abello et al., 2006].

Semantic-based abstraction is a more powerful mechanism for providing an overview, zooming, or giving details. This approach is tied to its intended application since it requires specific domain knowledge of the semantic information [Shneiderman, 2006]. When combined, structural and semantic abstraction can prove to be very effective [Shen et al., 2006]. Also in [Shen et al., 2006], it is shown that overview and level-of-detail (LoD) enabled browsing can be based on induced subgraphs of different sizes.

There are many well-known packages that have evolved over time to specifically address visualization of gene correlation data using node-link diagrams such as Cytoscape [Shannon et al., 2003] and VisANT [Hu et al., 2004]. These tools are built to be web accessible and thus render node-link diagrams using 2D layouts. While 2D layouts are accepted by the community, such packages neglect modern 3D acceleration hardware, rarely scale well beyond hundreds of nodes, and do not leverage 3D operations that have proven to be the preferred representation and navigation techniques for our users. Due to the common 2d framework, and in contrast to Shneiderman's principle, biologists are typically forced into a workflow in which filtering must be first applied and a global overview of the entire dataset simply isn't possible. Our software leverages both OpenGL and efficient C compilation to facilitate interaction with tens of thousands of nodes while maintaining interactive performance with complex visual analytics tools not currently available in these packages. Current work involves integration with a lightweight API [Shannon et al., 2006] to allow web-based interaction and data-sharing so our software may be used synergistically with such well-developed packages.

In contrast to the node-link model, an adjacency matrix is a clutter free interface. While an adjacency matrix interface for large data is limited by the resolution of the display, it is still ideal for a bird's eye view [Abello and Korn, 2002]. Some patterns such as clique and bipartite subgraphs could be very distinctive when examined in an adjacency matrix. However, a proper order of vertices is critical. The community has studied this problem at length. In [Henry and Fekete, 2006], a comprehensive survey on automatic vertex order is included. In general, binary, undirected graphs are the most straightforward. While weighted graphs needed more complicated algorithms, graphs with negative weights are less studied. Based on adjacency matrices, LoD type of browsing is often supported as well [Abello and Korn, 2002].

Due to the complexity involved in computing a high quality overview of a graph, researchers have also attempted to use self-organizing maps [Kreuseler and Schumann, 2002]. Self-organizing maps are a dimension-reduction technique which adjusts weights in a manner similar to neural networks to discretize the input space in a way that preserves its topology. The end result is (usually) a 2D field that can be conveniently rendered as a terrain.

By creating a spatial layout for a graph, it can be interactively visualized while preserving the data's underlying topological relationships. Typical interaction methods include focus+context methods (i.e. zoom and filter), graph queries using language-based methods [Sheng et al., 1999], and filtering databases of graphs using graph similarity metrics, typically based on non-trivial graph theoretic algorithms [Raymond et al., 2002].

Social networks are currently a primary driving application of interactive methods for graph visualization. This has resulted in non-binary, non-positive definite weights not being as thoroughly studied. Also, tools for extracting highly connected subgraphs from this data in a way that addresses the inherent uncertainty appear to be lacking. Whereas neural networks have already been used for volume segmentation [Tzeng et al., 2003], similar approaches have rarely been attempted in graph visualization. In this work, we propose several tools that allow traditional quantitative drill-down as well as qualitative selection and filtering techniques to aid domain experts with their analysis.

2.4 Parallel Coordinate Plots

Parallel coordinates, popularized in large part by [Inselberg, 1985], have become increasingly popular as a scalable technique for visualization and interaction with large multivariate data. A parallel coordinate plot (PCP) is a generalization of a Cartesian scatterplot in which axes are drawn parallel to one another. This type of diagram highlights the more common case of parallelism, rather than orthogonality, present in higher-dimensional geometry. PCPs also allow an arbitrarily large number of dimensions to scale intuitively within the plane, whereas human perception degrades quickly as dimensions higher than three are projected to a 2D display.

PCPs developed as a way to accurately visualize and thereby gain insights from multidimensional geometry. From their onset, several mathematical properties were proven which enhanced their interpretation by defining analogues between parallel coordinate space and two-dimensional Cartesian scatterplots [Moustafa and Wegman, 2002]. These included the point-line, translation-rotation, and cusp-inflection point dualities [Inselberg, 1985, Inselberg and Dimsdale, 1994]. This technique quickly found its way into Vis [Inselberg and Dimsdale, 1990].

There has been much research to alleviate some of the inherent weaknesses of PCPs such as visual clutter when dealing with large data. Techniques for clutter reduction include clustering, subsampling, and axis redirection. In [Fua et al., 1999], the authors use a clustering algorithm to create a hierarchical representation for PCPs and render a graduated band to visually encode variance within a cluster. In [Johansson et al., 2005b], the authors use K-means clustering, a high precision texture to reveal specific types of clusters, multiple transfer functions, and an animation of cluster variance to accurately convey the clustered data while minimizing clutter. In [Ellis and Dix, 2006], the authors provide a sampling lens and demonstrate that random sampling of lines within the lens is the optimum choice in the tradeoff between their accuracy metric and performance. In [Wegman and Luo, 1996], the authors use the grand tour algorithm to generate a d-space general rigid rotation of coordinate axes which can be used to confirm separability of clusters.

Perceptual properties such as the importance of axis orderings were considered as early as [Wegman, 1990]. While [Wegman, 1990] gives equations for selecting an order from a set of axes, he in no way addresses optimality criteria. The work most similar to this paper is the work of [Peng et al., 2004] in which a dataspace clutter metric was introduced in combination with heuristic algorithms for determining axis ordering. However, we more systematically address the axis ordering problem with our introduction of customizable metrics, globally optimal ranking, and near-optimal ranking algorithms with lower theoretical complexity.

PCP implementations often operate on binned data and even uncluttered PCPs often demonstrate data ambiguities. The traditional straight-line rendering can be augmented by using energy-minimization to render curved lines as in [Zhou et al., 2007]. Likewise, the authors in [Graham and Kennedy, 2003] used Gestalt principles of good continuation in order to address ambiguities from line crossovers and shared values. Binning can cause outliers to dominate data expression or be filtered out altogether which has lead [Novotny, 2006] to more thoroughly analyze the preservation of outliers while capturing the major patterns in PCPs.

In additional to use solely as a visualization technique, PCPs can be used as an intuitive navigation and query methodology. Recent research has demonstrated the ability to interactively alter axis ordering, interactively brush range queries, and use axis scaling to refine those queries [Steed et al., 2007]. The introduction of a navigation element to the visualization leads naturally to more complex data mining techniques. In [Ferreira and Levkowitz, 2003], the authors use parallel coordinates, and its circular variant, as multidimensional visualization elements in a taxonomy of other techniques. They also highlight user interface and analysis concerns, highlighting the strengths and weaknesses of PCPs in the context of other visualization methods.

There are also several variants of the traditional parallel coordinates rendering. These include the circular variant in which axes are drawn as spokes of a wheel [Ferreira and Levkowitz, 2003]. This approach was extended by [Johansson et al., 2005a] who created three-dimensional parallel coordinate renderings by placing axes like tent pegs in a circle around a central axis. PCPs were also extruded into three-dimensional renderings by treating each line as a plane that could arbitrarily be transformed [Wegenkittl et al., 1997].

Chapter 3

Dynamic Visualization of Coexpression in Systems Genetics Data

3.1 Introduction

Recent systems genetics research offers near term hopes in addressing scientific questions long-deemed unapproachable due to their complexity. Current research is uncovering how the genetic makeup of an organism is associated with the organism's traits on both molecular levels, such as gene expression or protein abundance, as well as physical levels including body height or tendency toward alcohol addiction. The systems genetics approach is a method to integrate data across all levels of biological scale to uncover molecular and physiological networks from DNA to function. Novel computational tools are called for to support the effort.

The central dogma [Doerge, 2002] for genetic studies is that strings of information known as genes are stored in the DNA sequence (genome) of an organism, each gene can be transcribed into messenger RNA (i.e. a transcript), and ultimately into proteins which affect the behavior or morphology of the organism. This multi-step process by which a gene's sequence of nucleic acids (ATCG) is converted into mRNA transcripts is known as gene expression. Gene expression directs the process of cellular differentiation, in which specialized cells are generated for the different tissue types. The regulation of gene expression (i.e. gene regulation) controls the amount and timing of changes to the gene product. This is the basic mechanism for modifying cell function and thereby the versatility and adaptability of an organism. Therefore, gene expression and regulation function as a bridge between genetic makeup and expression of observable traits.

Despite its vital importance, determining the precise roles of given transcripts remains a fundamental challenge. This is due in large part to the complex machinery employed for gene expression in which some gene(s) may regulate the simultaneous transcription levels of other genes. This regulation leads to statistically correlated, or co-expressed, genes in which one gene is expressed at high levels only when the other is as well. While collecting gene expression data already requires great technical sophistication and resources, the limited functionality of current computational tools to discover structural patterns of co-expressed genes from the collected data presents another grand challenge. Without an ability to efficiently and comprehensively explore the problem space, full genome-scale gene expression data are still of limited value for today's hypothesis-driven research.

Genes act alone or in groups during the process of gene expression and regulation. Biological pathways are defined by the connectivity of upstream and downstream effects of genes and gene products, including their action in the regulation of the expression of other genes. The search for genes co-expressed in a common group, which likely affect observable traits as a functional unit, often starts with using microarray technology to profile transcript abundance. A microarray is a device containing microscopic DNA probes and is capable of measuring the expression levels from thousands of genes for a given sample [Geschwind, 2000]. From microarray data, biologists can statistically construct massive correlation matrices that describe pair-wise gene co-expression. The key challenge then is the representation, decomposition, and interpretation of this genetic correlation matrix.

By treating the correlation matrices as adjacency matrices, it is natural to consider correlations of gene expression in the setting of a graph, where vertices represent genes and edges represent the strength of correlation between pairs of genes. In this analogy, a group of genes that co-express would necessarily form a network or subgraph consisting of "highly correlated" genes.

Although it seems straightforward to directly apply classic graph algorithms to discover those highly correlated subgraphs, this approach by itself is not sufficient. Many common graph problems such as clique finding are NP-complete. Even for moderately sized problems, the computation time is still often overwhelming. Hence, to ensure that problems are computationally tractable, the current practice is to apply data filtering steps to dramatically reduce the density of the graph and dimensionality of the data. The value of key parameters, such as correlation threshold, is often decided by an educated guess based on the data size, algorithmic complexity, and the hardware available. Unfortunately, picking a slightly different threshold often eliminates many of the subtly important correlations and thereby drastically changes the solutions of graph algorithms.

In addition, scientists in many cases cannot exactly define the term “highly” with rigor as it is a qualitative criterion with uncertainty. The uncertainty aspect is further exacerbated by the noise present in current microarray data, inaccuracies introduced during data collection, and residual errors in subsequent statistical analyses. To date, it has been hard for scientists to evaluate the true value of gene co-expression as well as the sensitivity of an “optimal” result computed using expensive graph algorithms.

In this work, we designed a visualization system to provide domain scientists with tools to evaluate the validity and sensitivity of key parameters in their research hypotheses. By allowing realtime feedback of connectivity, determination of biological relevance is facilitated by allowing more thorough analyses of their empirical data. Our main effort focuses on providing interactivity from a number of features beyond fast rendering rates. A user can interactively explore and filter the data to create meaningful subgraphs by leveraging four complementary methods: (i) semi-automatic segmentation of highly correlated subgraphs with a 2D focus+context graphical user interface segmented using block tridiagonalization, (ii) quantitative database queries on data of interest using traditional compound boolean range queries, (iii) qualitative queries on points of interest using neural networks, and (iv) dynamic extraction of subgraphs using several fast graph theoretic algorithms. In addition, these meaningful subgraphs may be used as templates to perform template-based searches through the entire dataset. The whole process of template creation, template-based search, extraction of graph metrics, and displaying statistical and visualization results is interactive.

Modern microarray data is noisy and complex; visualization alone is not the answer. By providing interactive visual analytics tools such as graph algorithms, neural network analysis and level-of-detail control, we bring a human expert into the loop to negotiate the tradeoff between data size and algorithmic complexity by intuitively tuning key parameters with realtime feedback for addressing the scientific question at hand. We demonstrate our system using datasets from a real-world, large-scale, genetical genomics study of mammalian gene co-expression. In this study, the influence of genetic differences among individuals is considered as a source of expression covariation.

3.2 Approach

Our goal is to develop a visualization system in which a human expert discerns uncertainties in the data and guides the system to segment a large graph using a set of automated tools through an interactive interface. The key components of the system include the 2D interactive interface, modules to select subgraphs both qualitatively and quantitatively, and the neural network based classifier that uses selections as a template. We start the discussion by describing the exact set of input data to our system.

3.2.1 The Data

The only required data is a matrix containing gene-gene correlation values. While all of our testing data use Pearson's correlation, different metrics of correlation are treated no differently in our system. In addition, we handle a database of information corresponding to each gene as can be seen using three relational tables (Figure **??**). Specific information about the object of interest is stored in the Gene table while information relating to computed gene networks is stored in the Paraclique table. Since our driving application is to identify the genes that cause variation in complex traits, it is necessary to show the relationship or distance between genes and QTLs. For that, we need an additional relational table describing the exact location of QTLs in the unit of megabases.

Graph theoretic algorithms provide valuable information that is otherwise hard to discern about the data. However, many such algorithms incur long compute times and are far from being interactive. For those algorithms, it is then necessary to pre-compute and store their results for visualization at run-time. In this work, for example, we pre-compute and store each gene's membership in any of the paracliques. The resulting data can easily be stored in a relational table.

We treat all data in the relational tables as attributes of individual vertices, and the correlation values as an attribute specific to each edge. This is a very generic model that is applicable to a variety of application domains and is a boon to scientists typically involved in spreadsheet science. Based on these data, it is then the job of the visualization system to facilitate interactive, hypothesis-driven study by the user.

3.2.2 A Clutter-Free Interface for Graph Abstraction

A major difficulty with graph visualization is the visual clutter caused by the sheer complexity of the data. As discussed in Section **??**, an adjacency matrix provides a concise interface for overviewing the data in a way that is free from visual clutter. However, the 3D space is still a natural domain for user cognition. The added dimension can be used to convey additional data, allowing navigation through node-link rendering and full appreciation of structural cues in the data. For our application, we have designed a framework (illustrated in Figure 3.1) where an overview is provided through a 2D adjacency matrix. Users can arbitrarily select subsets of interesting vertices and create abstractions for further interactions in 3D.

Unlike popular datasets studied in most previous works, the range of edge weights in our data is `[-1.0,1.0]`. In addition, to let users decide about uncertainty issues, we would like to avoid taking a threshold at this stage of processing due to possible information loss. This makes it hard to directly leverage existing vertex reordering algorithms like those surveyed by Mueller [Mueller et al., 2007] or Henry et al. [Henry and Fekete, 2006].

Block tridiagonalization (BTD) is a mature numerical algorithm that permutes row and column elements of a matrix in such a way as to cluster nonzero elements in blocks along the diagonal [Bai et al., 2004]. This algorithm always preserves the eigenvalues of the original matrix within a specified error tolerance. It iterates until the following criteria are met: (1) the final matrix has small bandwidth relative to the size of the matrix, and (2) any off-diagonal blocks in the final matrix have either low dimension or are close to a low-rank matrix.

The BTD algorithm was developed to improve both performance and storage efficiency for solving eigen problems. The smaller a block is in a matrix, the lower the corresponding rank in most cases. Thus, the optimization goal of BTD is to minimize block sizes on the diagonal, and correspondingly reduce block sizes off-diagonal as well.

The result of BTD is often characterized as minimization of bandwidth, because non-zero entries are clustered around the diagonal. It is very significant to our research. In our application, the minimization of diagonal block sizes through global optimization provides a reliable means to abstract a large graph into a set of minimal basic "building blocks," each of which represents a densely correlated subgraph. The vertices in these subgraphs appear in contiguous segments along the diagonal. The off-diagonal blocks determine how these"building block" subgraphs are interconnected. In this way, we can conveniently reassemble the original graph using the minimized diagonal blocks, and show more appreciable structures with significantly less clutter.

Let us consider the illustration in Figure 3.1 from a biological perspective. In this example, we show four graph patterns that are often of interest to geneticists. Since every data entry in an adjacency matrix represents an edge, selections made in adjacency matrices are on edges and only indirectly on vertices. The green subgraph is a clique of highly correlated genes that potentially operate as a unit. The orange subgraph is a bipartite graph, used in gene-phenotype mapping, in which the trait is correlated with a number of related genes which would be of experimental interest. The blue subgraph is a perfect matching graph which functions as bridges between subgraphs. The red subgraph is a star containing a "hub" gene which could be targeted for knock-out and affect expression in many structures.

For our real world datasets, BTD has been able to consistently generate permutations that compress the majority of non-zero to the diagonal. This enabled us to crop a stripe along the diagonal and rotate that stripe to a horizontal position, as shown in Figure 3.1, bottom. We refer to the horizontal strip as the BTD belt.

BTD belt is a more efficient use of precious screen resources. Our datasets typically contain several thousand genes so the adjacency matrix is typically very large. Although it is possible to downsample the matrix for on-screen viewing, the essential high frequency details in the matrix could be hard to distinguish.

We show an example of the BTD belt computed from a gene co-expression study of brain development in Figure 3.2. There are 7,443 genes in this dataset. In an adjacency matrix, one can make selections with square shaped bounding boxes. In BTD belt, these square bounding boxes becomes diamond shaped. Ten sample diamond shaped selections have been specified in Figure 3.2 and magnified to show details.

From the BTD belt interface, a user can select diagonal blocks that are perceived to be "highly correlated". Letting a human expert decide what can be considered as "highly correlated" is our way of handling data uncertainty. We note that the functionality of detecting high correlation in a general setting is a hard problem, particularly when the acceptable tolerances of error can only be qualitatively determined in a subjective manner.

In this regard, BTD can be considered as a computational tool for creating data abstraction. Figure 3.3 shows a simple example in which ten subgraphs have been selected using diamond shaped bounding boxes. Each subgraph is abstracted as a super node in the LoD graph. From the much simplified graph, the interconnections among the graph nodes are clearly discernable. As expected from the BTD representation, a-c and g-i form cliques with high edge weights. Interestingly, node a has a strong negative correlation (as indicated by the dark line color) with subgraphs d, g, and j. LoD visualization of BTD selections complements BTD in the sense that while these multiple separated clusters may not share many edges (and thus may not be readily visible in the BTD belt), the edges that do exist have strong negative correlations. The LoD representation implies that subgraph a is a potential down-regulator for these three major networks.

Since the BTD-belt is a permuted and rotated adjacency matrix, much information is visible along the diagonals which correspond to an individual vertex. Therefore, the BTD belt immediately shows the major graph structures in which each vertex participates. These properties can be used to quickly determine the role of specific genes in varying numbers and types of networks.

At a higher level of abstraction, several large structures share many edges where their diagonals cross as seen in Figure 3.3 where subgraphs g-i form a bipartite-like structure with j. To be specific, the bipartite structure visually represents the edges induced by the intersection of each graph's vertex set. The striping pattern of the bigraph structure implies it is typically the case that a single vertex in g-i is connected to many vertices in j and not vice versa. This allows biologists to get a view of how multiple structures may interact, or be regulated, through genes that they have in common.

To allow further data abstraction, users are able to dynamically generate level-of-detail (LoD) representations that facilitate simultaneous operation across multiple levels of scale. At any point in our system, the current working graph can be saved and is rendered, along with all other saved subgraphs, in the background of the current working graph. The LoD graph can be generated by taking all user-defined subgraphs and treating them as supernodes. By default, the edge connecting two supernodes has a weight defined as the average of all edge weights between vertices within the two corresponding subgraphs. Optionally, feature vectors of topological metrics can be calculated on each subgraph for querying and graph pattern matching, using methods described in Sections 3.2.3 through 3.2.5, to find similar graph structures rather than similar data items. This LoD graph can subsequently be treated as a normal graph and all provided analytics tools in our system can be used to perform increasingly complex analysis at higher levels of abstraction.

3.2.3 Quantitative Queries

Information contained in relational tables (Figure **??**) needs to be studied in an integral fashion with gene networks. A common comprehensive tool for accessing information in those formats is compound boolean range queries. Unfortunately, it is a difficult process to efficiently integrate a commercial database management system with realtime visualization systems. For this application, we have extended the functionality of a recent visualization data server to provide essential data management functionalities. The core of that data server is a simplified B-tree that is optimized assuming no run-time insertion or deletion in the B-tree [Glatter et al., 2006]. Using this B-tree, the database in our system, with compound boolean range queries over 8 features, can be queried in O($log\ n$) time at a rate of 10 million vertices per second on a 2.2Ghz Athlon64. This results in interactive, sub-millisecond response to sets of dynamic queries even for large graphs.

The effect of querying is data filtering and thereby complexity reduction. One of the most common data filtering operations for biologists is thresholding. With visualization, this threshold choice can be applied to a graph interactively and result in both visual and statistical testing for proper threshold selection. Besides thresholding, we also provide more power by also allowing multiple, dynamic, quantitative queries over any computable attributes of a vertex or edge. Sample queries include all genes within a 100-megabases distance to a target QTL, or all genes in the current subgraph that are significantly related to a paraclique of interest.

The queried genes also form a subgraph, no different from those qualitatively selected via the BTD belt interface. Our querying system attains realtime performance, making it feasible to visually evaluate the effects or sensitivity of key parameters, such as what threshold value to use.

3.2.4 Dynamic Fuzzy Classification

When interacting with complex data, it is often useful to provide an automated arbitrator between the user and the data so that repetitive tasks are off-loaded from the human user. One such important task is to exhaustively search for genes that match a certain pattern and classify the data accordingly while handling the innate uncertainty. A key motivation is to alleviate users from the need to manually browse through the entire dataset and thus specifying the feature they think they are seeing in an overly rigorous manner. It is desirable to have a proper level of fuzziness into this iterative feedback loop. While elaborate agents can take a long time to develop and are too complex to train in realtime, we have implemented a simpler AI system which users can train at run-time.

In our system, we use a feedforward, multilayer, back-propagation neural network capable of quickly learning items of interest and displaying similar items to the user. From this perspective, qualitative selection allows users to visually perceive uncertainties and decide how to best guide the computational process, while quantitative queries provide an exact means to request a subset of data. With all datasets, the neural network classifier can be trained with subsecond efficiency.

3.2.5 Graph Properties

While our neural network treats input attributes indifferently, the choice of which properties to feed into the classifier is quite important. Besides domain-specific database variables, one may also employ several integrated graph properties for graph similarity classification. Those include: degree of vertex, transitive closure, connected components, edge expansion, and shortest path.

Each of these properties has a unique biological interpretation. Degree is one of the simplest useful metrics that can be calculated and allows biologists to determine statistical correlation between genes and gene networks as there are often many loosely connected genes and few highly-related genes. Transitive closure minimizes the longest distance to all other nodes and can be used to find the core of a genetic structure. Connected components allow biologists to visualize only related data within a given subgraph. Edge expansion shows relationships within a given subgraph without regard to threshold. Shortest path allows biologists to see how specific genes most directly regulate one another. It could also be used to further investigate a favorite location in the graph and gather only the local correlates view of the specific gene(s) under study.

These properties may be used as extra variables in the feature vector used for training a neural network. By adding these to the relational data that are queried, we provide more information to be leveraged for fuzzy classification.

3.3 System Implementation

In this section, we describe details of our system that may be of interest to readers who would like to implement a similar system. Several performance numbers for our system may be found in Table 3.1 The number of vertices, edges, and range of absolute values of edge weights are shown, respectively, along with performance metrics referenced in the sections below. The columns of "2D" and "3D" show the running times (in seconds) of Fruchterman and Reingold's [Fruchterman and Reingold, 1991] method operating in 2D and 3D, respectively. The column of "BTD" shows the timing results of the BTD permutation process in seconds. The time to complete neural network training from a few examples and counterexamples along with classification of the entire dataset is recorded in number

Table 3.1: Datasets and Timing Results (in Seconds)

$\lvert V\rvert$	$\lvert E\rvert$	$\lvert Weight\rvert$	2D	3D	BTD	NN(ms)
254	401	[0.57,0.95]	0.203	0.282	0.1	16
2,150	6,171	[0.97,1.00]	16.20	17.19	6.1	31
7,443	695,122	[0.85,1.00]	336.0	318.3	50.9	141
12,343	28,338	[1,74]	883.3	912.7	234.7	172

of milli-seconds (ms) in the column “NN”.

Since this discussion is not restricted to one of only gene expression applications, we use four test datasets detailed in Table 3.1 to indicate the scalability of the system. The datasets include genotype correlation datasets used to study human lung cancer, medical bibliographic references, mouse behavior, and web architecture made available by the developers of GeNetViz [Zhang et al., 2005].

The primary dataset, which we use throughout the rest of this paper, for our driving application involves regulation of 7,443 genes for research of mammalian brain and behavior [Chesler et al., 2005]. Visualization results concerning complex traits in this dataset are shown in Section **??**.

3.3.1 Graph Layout

Our system attempts to load any pre-processed data available or subsequently generates the files if they are not available. Upon startup with a valid weighted-edge graph, the system inspects the graph to determine specific properties which would necessitate a special layout, as in the case of a bipartite graph. Otherwise, it generates 2D and 3D layouts using either Kamada-Kawai [Kamada and Kawai, 1989] energy minimization or Fruchterman and Reingold’s [Fruchterman and Reingold, 1991] force-directed placement. Both layout algorithms are the standard implementations which use simulated annealing to slowly freeze the layout in place. The layout is said to converge when it reaches a maximum number of iterations, reaches a small percentage of the initial system temperature, or does not change significantly between iterations.

Like most regular layout algorithms, in our implementation the vertices are initially placed randomly and then iterated through four main phases until convergence: random impulse, impulse away from all other vertices to keep them from overlapping, impulse toward the center to keep the system from continuously expanding, and per-vertex impulse toward or away from its connected neighbors in proportion to the edge weight to preserve graph topology. The entire process is $O(M|V|^2)$ where $|V|$ is the number of vertices and M is the number of iterations ($M \sim |V|$ but varies significantly). Through experiments we found that the performance differences between 2D and 3D graph layout algorithms are quite minor as shown in Table 3.1.

The software has been designed such that other graph layout algorithms, such as the fast multipole multilevel method (FM^3) [Hachul and Junger, 2004] or LinLog [Noack, 2004]. Such layout algorithms have different strengths and weaknesses in conveying specific properties in the resulting topology. Custom algorithms can be easily incorporated in our application either procedurally or by simply loading a graph layout file. By default, a single layout is computed for a graph and its appearance is modified at run-time but the user may also switch interactively between multiple layouts. Additionally, the user may dynamically swap between the customary 2D view and the 3D view which tends to convey more relationship information.

It is also noteworthy that clustering is another term often used by biologists in their research. Although in our work "cluster" and "dense subgraph" have similar semantic meanings, the goal of our approach is quite different from the basic goal of popular clustering algorithms. Our goal is to adequately handle uncertainty in the pursuit of coregulated (putatively cooperating) genes forming a network, and the nature of the interconnections among those networks. We use BTD belt, queries and neural network to computationally assist the discovery and realtime fine-tuning of those dense subgraphs of interest. We do not solely rely on graph layout algorithms to reveal dense clusters.

3.3.2 Rendering

Vertices are rendered after the edges without the depth buffer to prevent edges from occluding data points. We support the option of rendering vertices as splats (also known as billboards or impostors) or quadrics which can take arbitrary shape (usually spherical). The splatting implementation is the typical geometry-based primitive scheme using preclassification and thus results in slightly blurry vertices but has the advantage that it is typically fast. While both options render in realtime on most single computers, framerate tests were conducted for the 7,443-node graph at several resolutions on a powerwall using Chromium. This resulted in 16 fps quads vs. 246 fps splats on an 800x600 viewport, and 5 fps quads vs 17 fps splats at 3840x3072 (9 monitors).

There are many interactive rendering options such as color table generation and weight-mapping mechanisms for coloring edges. A default set of these has been provided to express differentiation between solely positive and negative edges (up and down regulation) or to enhance contrast between edges with similar weights. The rendering mechanism is adaptive and can optionally adjust properties such as the number of subdivisions in the quadrics based upon the current frame rate. Semi-transparent vertex halos [Tarini et al., 2006] are rendered using splatting for enhanced depth perception. Intuitive click-and-drag interaction and continuous rotation during manipulation circumvents problems with 3D occlusion and aids perceptual reconstruction of the topology through motion parallax. The system also has dozens of minor utilities including the ability to change the color table used by all elements of the program, take a screenshot, create video, print statistical information for the current graph, and output gene lists.

3.3.3 Neural Network

We use a multilayer, feedforward, back-propagation, online neural network for realtime classification which is able to learn while in use by employing stochastic gradient descent. Our implementation closely follows the description in [Russell and Norvig, 2002]. Results are given for a neural network with the number of input nodes corresponding to the number of provided attributes, 30 hidden nodes, 2 output nodes, a learning rate of 0.4, a sigmoid threshold function, and a hard max used to simply select the most likely output. The unusually large number of hidden nodes provides sufficient degrees of freedom for any problem domain and could be reduced if training speed or overfitting become issues. Each of the two output nodes corresponds to likelihood that the user does or does not want to see the object.

Neural network interaction involves only a few easy steps. The user left-clicks on an arbitrary number of vertices to select them as examples. Similarly, right clicking on a vertex adds the vertex as a counter-example. The user may use any filtering or processing techniques previously mentioned to aid the process of defining the training set. Once all examples and counter-examples have been selected, the entire dataset is processed to only show items like the examples or to segment the data using color. Training the neural network and classifying all other data items is in the order of dozens of milliseconds, shown in Table 3.1, and is transparent to the user.

Deciding the proper training set is critical when attempting to achieve accurate classification for multivariate data due to the high-dimensional decision space. In our system, we provide two alternative approaches for the user definition of large training sets. First, we allow selection of the training set using supernodes in the LoD graph. Each supernode represents a network of genes, typically segmented through BTD selections or database queries, such that a few example supernodes can correspond to hundreds of individual data points. In this way, users can visually interact with the data while quickly selecting entire groups of vertices as examples or counterexamples. Second, the application allows the import/export of vertex data for the current working graph using ASCII files. This can be used to analyze the corresponding data with much more sophisticated statistical packages such as Statistical Analysis Software (SAS) or statistics programming languages such as R. These analytics tools or their batch programs can be called during run-time to either fully segment or partially classify the data. The result can be stored in a vertex list file and then utilized as a training set.

The accuracy of our neural net implementation was independently debugged using a different application. Our debug case was a benchmark of pixel-based classification of 10 letters selected from the "Artificial Character Database" at UCIs Machine Learning Repository which were converted into 8x12 binary images. Several tests were conducted using cross validation with mean-squared error to measure overfitting as well as determine the optimum parameter settings. The best neural network results in correct classification 96.72% of the time on new input characters (i.e. not part of the training set).

3.4 Results

3.4.1 Overview: Data and Workflow

While early microarray studies emphasized differential expression and comparative analyses, modern applications [Jansen and Nap, 2001] emphasize correlation of gene expression across large sets of conditions, including environments, time points and tissues. Increasingly, this data is being collected in a context of natural genetic variation [Chesler et al., 2005], where it can be integrated with multiple data sources such as genome sequencing, QTL analysis, and disease relevant phenotypic data. For this application we focus on gene expression analysis conducted with a particular emphasis on those traits related to brain and behavior in the laboratory mouse.

A primary source of covariation in gene expression are single nucleotide polymorphisms (SNPs). Studies in genetical genomics [Jansen and Nap, 2001] attribute variation and covariation in gene expression to the influence of these differences in DNA sequence. The use of recombinant inbred strains allows biologists to study replicate populations of mice with identical genomes. These populations allow indefinite aggregation of data across studies as new technologies for characterization of mice become available. When traits are assessed across genetically identical individuals, the correlations among traits are assumed to be due to common genetic regulation. By finding and analyzing statistical correlations between genotypes and phenotypes, geneticists hope to discover and interpret the network of causal genotype-phenotype relationships that determine a trait of interest.

Systems genetics research often follows a workflow of finding a gene network, finding regulators of that network, and then performing a focused gene perturbation experiment to determine the role of the associated network on gene expression or function. To begin, a "large" gene correlation graph must be sifted through, to find a highly connected subgraph which corresponds biologically to a gene network in which genes are expressed together, presumably to regulate or subserve a common function. They must then find a small set of causative genes, highly correlated with the subgraph and likely to regulate co-expression, to be used as targets of focused investigation. By manipulating the expression of these genes, the function of the gene network can be determined through observation of expressed phenotypes. Proof of causality occurs when the gene manipulations recapitulate network relations. It should be noted that while standards of “large” are highly application dependent, even graphs with less than 10k vertices exhibit a combinatorial space that is overwhelming and, indeed, presents a rather large and unique problem unlike dealing with volume datasets.

In this section, we showcase results for publicly available biological data which has been the subject of several previous studies. Whole brain mRNA gene expression data was obtained using the Affymetrix U74Av2 microarray for each of the strains in the BXD mouse population and subsequently processed using Robust Multi-Array (RMA) normalization [Chesler et al., 2005]. Throughout the paper, we use Pearson’s correlation over 7,443 genes of this dataset as our driving application. The associated database in our system is used for querying, interactive neural network training, and constructing dynamic level-of-detail (LoD) graph features; it contains information relating to typical systems genetic analyses for each gene such as: the chromosome, position (in megabases), paraclique membership and connectivity, broad-sense heritability indices, and QTL mapping [Wang et al., 2003] locations with p-values from QTL Reaper (sourceforge.net/projects/qtlreaper).

3.4.2 Discovery of Novel Networks

Typical biologists bring a large amount of domain-specific knowledge to their investigative process, for which many tools exist but are usually challenged by purely data driven investigation of networks. One approach to discovery of novel systems genetics networks is the use of computational tools which allow extraction of highly connected subgraphs in a qualitative fashion. By providing block tridiagonalization in which clusters around the diagonal constitute highly related genes, biologists can easily select potentially novel gene networks. Indeed, this $O(|V|^2)$ algorithm quickly extracts dense subgraphs and can be treated as a rough approximation to the NP-complete problem of paraclique enumeration in this context.

In Figure 3.4, the user has selected four BTD regions and dynamically generated a level-of-detail graph. As is expected, the selection 1 is most unrelated to the rightmost selections and therefore placed far away from the other selections with a negative correlation to selection 3. By selecting LoD vertices 2-4 as examples and 1 as a counterexample, neural network training on entire subgraphs is used to perform template-based search for similar genes in the original data. The resulting classification from the database information is the graph in the right center which has been extracted through the application of domain-specific knowledge in combination with several computational tools (BTD selection, LoD graphs, and NN template matching). This highly-connected subgraph contains genes which are similar to cliques 2 and 3 and bipartite structure 4 selected from the BTD and gives biologists a potentially novel network of two highly-related dense subgraphs to inspect for related function(s). This is currently being applied to create a comprehensive bipartite graph of gene networks (represented by LoD vertices) on one side and all network regulators (network interface genes) on the other. The ability to interactively and qualitatively search across multiple levels of detail has given biologists several tools for which they can not only solve current problems but also find new ways to address more difficult problems.

The central motivation of our system is to enable more in-depth and flexible expert-driven analysis by providing a diverse set of computational tools. However, there are other more established algorithmic tools, such as graph analysis, that are of value to scientific research. Those tools can be leveraged from within our system through our internal B-tree based data structure which allows queries from algorithmic solutions at a rate that facilitates realtime rendering and interaction. In the section below, we present a significant use case that demonstrates parts of our system to discover network interface genes.

3.4.3 Use Case: Discovery of Network Interface Genes

We now demonstrate our application with a biologically significant use case. Once gene networks have been extracted, it is of primary interest to determine the identity of the gene products that regulate these networks. Using either qualitative BTD selection or algorithmic network extraction, the total decomposition of a genetic correlation matrix into disjoint subgraphs can be achieved. With each disjoint subgraph treated as a structure, finding mRNA transcripts with strong correlations to multiple structures would lead to the discovery of “interface genes”. These mRNA transcripts regulate expression of genes in those structures, and thereby couple multiple networks and biological processes. The detection of these transcripts and the analysis of their gene’s regulatory polymorphisms could lead to the discovery of major genetic modifiers of large biological networks.

Our domain experts have found paraclique extraction to be the most useful and general algorithmic technique. Although choosing the proper threshold is a hard problem in general, by way of repetitive experimentation, statistical and combinatorial analysis, 0.85 is a perferred threshold for extracting paraclique in the dataset at hand [Chesler and Langston, 2005]. Paracliques with more than ten vertices were extracted, resulting in thirty-seven dense subgraphs, and stored in the systems database. We show the largest and third largest paracliques corresponding to gene networks ready for further study in Figure 3.5.

In this use case, the challenge is to identify candidate genes that may be the common regulators of expression for a large number of other genes, and to determine which functional biological characteristics or disease related traits the network may be involved with. Some facts are known about this situation: 1) each gene's physical location within the genome is near the location of one of the genotype markers associated with the gene expression level; 2) the interface gene may be a member of one of the dense subgraphs, but it must be highly connected to members of both dense subgraphs; and 3) the biological regulator is likely to be in the same pathway as the genes it regulates.

By creating a level-of-detail graph in which the edge weight for supernodes is defined as the percent of connectivity to the adjacent gene or supernode, we can use the hot-cold coloring scheme to visually elucidate the correlation between multiple structures. This allows for an intuitive representation of network distance which can allow biologists to identify functional modules and their relationship to each other in forming the pathways underlying specific biological processes. The data is then filtered via threshold to retain only the strongest correlations between individual transcripts and entire networks. Once such a network is constructed, additional queries may be posed that relate to additional candidate genes and their association to the genetic polymorphisms that regulate the network.

In this use case, the technique described above was applied to visually identify a specific gene of interest whose transcript is connected to over 99% of both the largest and third largest paracliques as shown in Figure 3.5. The gene of interest, Pr1 or Tmem37 (Affymetric probe set ID 95464_at), has thus been implicated as part of the regulatory pathway that coregulates the two large networks. This gene encodes a voltage-dependent calcium channel gamma subunit-like protein which modulates electrical properties of neuronal cells. It can be hypothesized that the paracliques are related to neuronal activity. Further testing was then accomplished through data export and communication capabilities to specialized bioinformatic pathway analysis tools. By further analysis of expression for the gene of interest, using genenetwork.org, it was revealed that its transcript abundance is correlated with stereotyped locomotor behavior induced in BXD mice by drugs such as cocaine [Jones et al., 1999] or methamphetamine [Grisel et al., 1997].

The next stage in the workflow is to enumerate candidate genes which reside at chromosomal locations nearby the gene of interest's regulatory QTLs. To do this, we must first run an analysis of genotype associations to the expression of interface gene Pr1 in order to identify QTLs that regulate its expression. This analysis, which we have performed using GeneNetwork.org and linear modeling in SAS v9.1, reveals putative regulatory loci at specific locations on chromosomes 1, 2, 6 and 14. By using our tool's integrated data query capability, we can then highlight co-expressed genes that reside in regions provided by the other tools. The candidate genes may help regulate the networks since they are located in close physical proximity to one another, their transcripts are highly correlated to many paraclique members, and the QTL that regulates them also regulates many members of both paracliques. This leads to the identification of a limited number of candidate regulators including Pax3, Lama5, Mkrn2, and Dhrs4.

We have now derived testable hypotheses regarding the mechanisms by which the networks are co-regulated and can validate this new knowledge with in vivo experimentation. It follows from the analysis of coexpression that these two paracliques are controlled by a common genetic regulator. A high genetic correlation implies that the biomolecules represented by the connected vertices have values that are determined by the same genotype. Expression of the interface gene Pr1 can also be correlated with biological function and traced back to a regulatory QTL. Pr1 and the two dense subgraphs have been associated with specific traits measured in BXD mice. A small number of candidate regulators from positions near the regulatory QTL has also been identified. The resulting visualization reveals networks which go from locomotor responses to specific drugs down to the connected molecular pathways which underly them.

While Figure 3.5 represents only a few steps from a typical workflow, the result of this finding captures overwhelming complexity. As early forays into systems genetic analysis have demonstrated, biological processes such as mammalian behavior involve a complex interplay of expression from many hundreds of genes across multiple chromosomes and biological systems. For this reason, we expect similar linked views of multiple tools to be the norm for systems genetic visualization. The tool presented herein allows users to retain networks and their relationships as well as rapidly isolate genes and subgraphs based on their connectivity. The tools demonstrated represent a flexible and dynamic approach which allows users to scale up from single genes or traits of interest found in web based tools to results of global analyses such as clustering and high-performance combinatorial analyses.

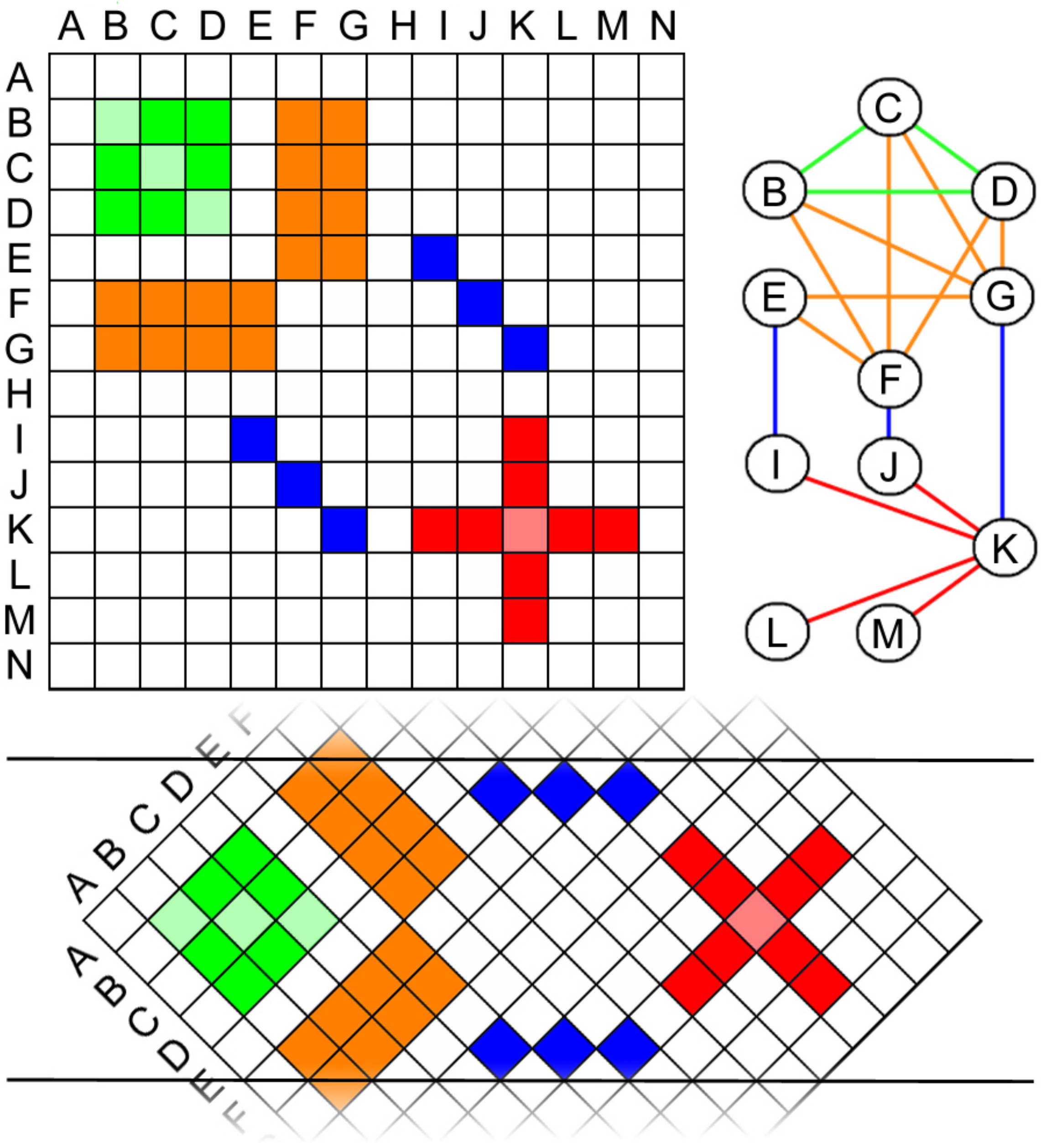


Figure 3.1: Illustration of a permuted adjacency matrix with common graph patterns (top), and extraction of the BTD belt for qualitative selection (bottom).

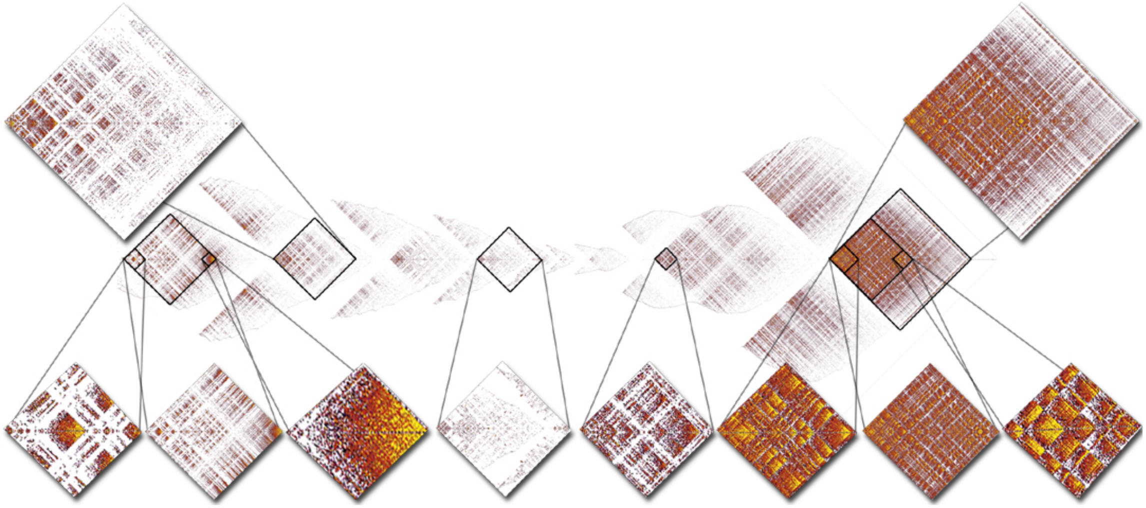


Figure 3.2: A BTD belt, with magnified views, from a real-world mammalian gene co-expression study of brain development involving 7,443 genes.

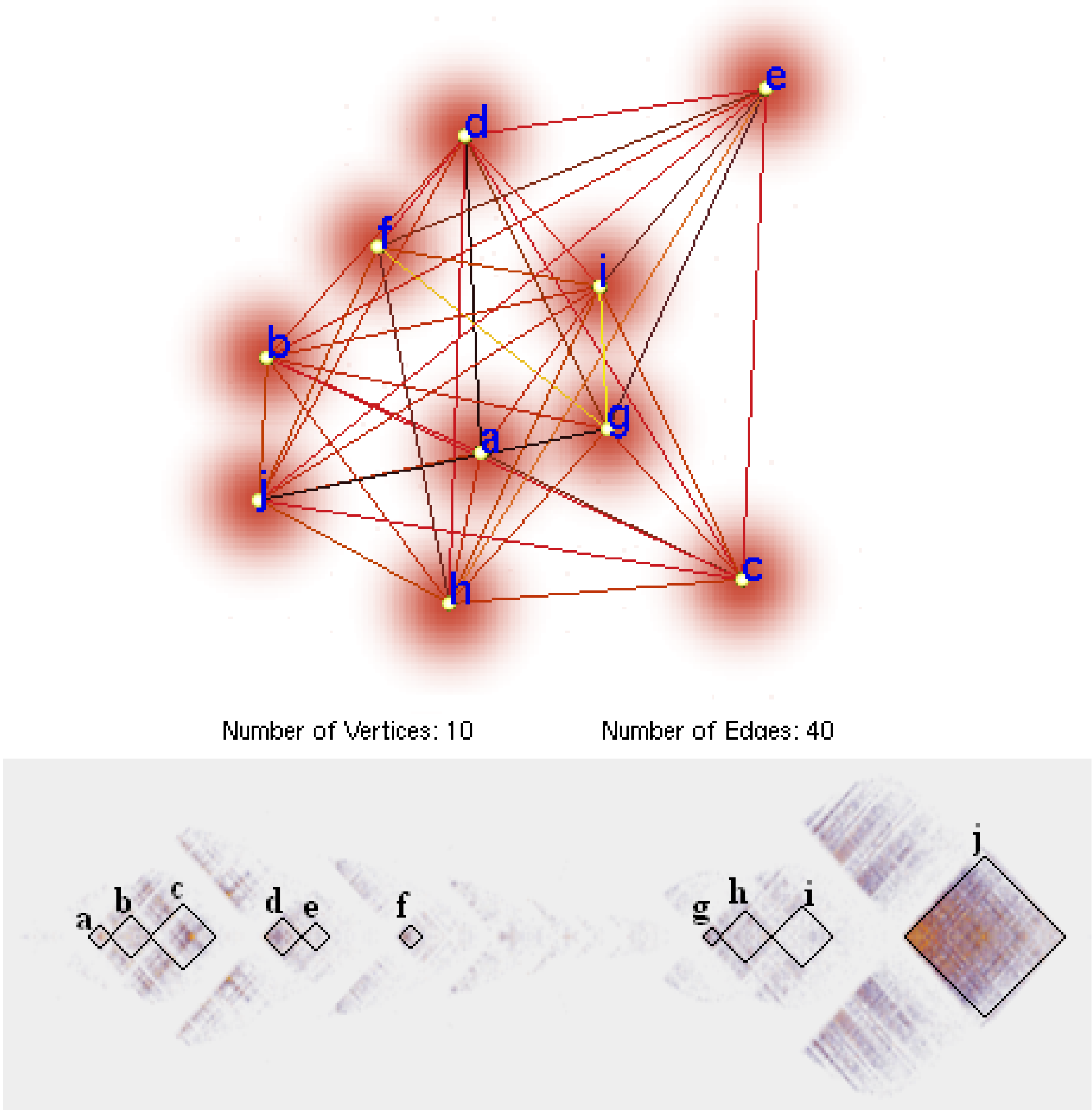
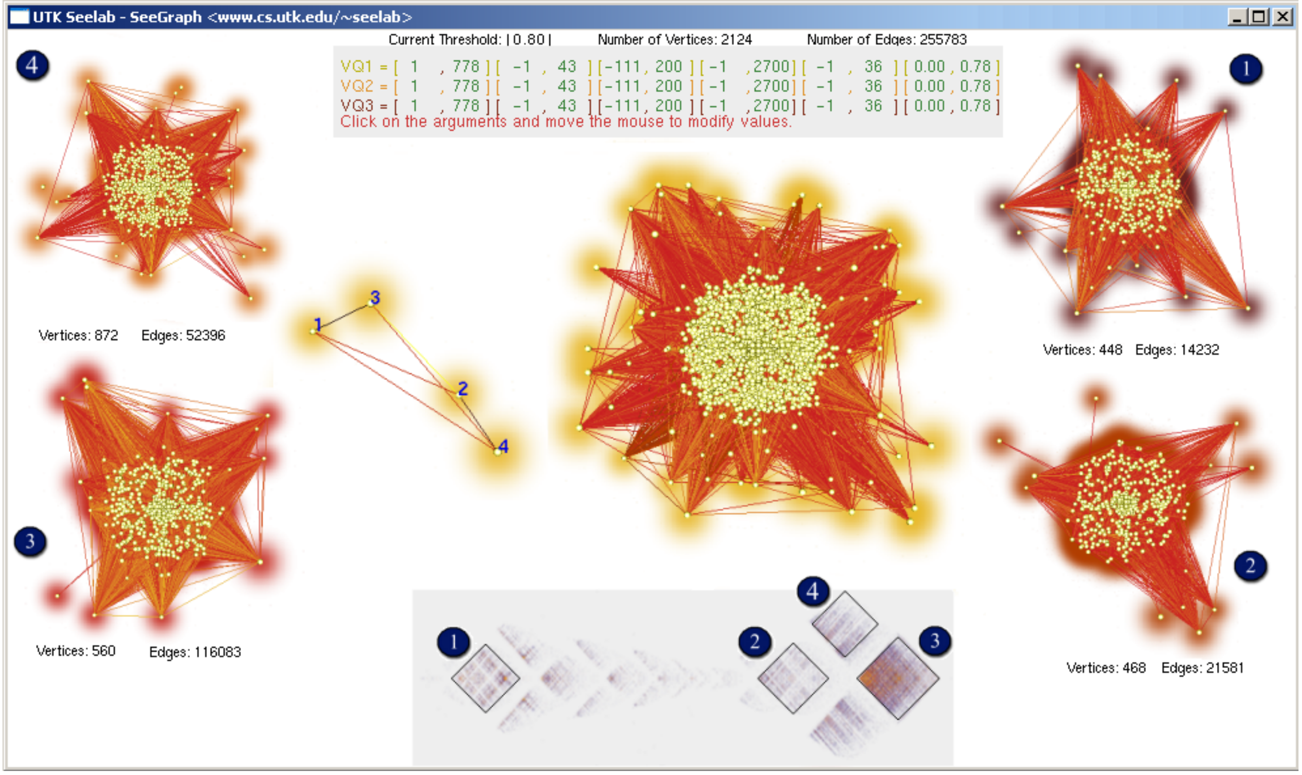


Figure 3.3: A 2D level-of-detail graph created from brushed BTD belt selections to show correlations among BTD structures.



UTK Seelab - SeeGraph <www.cs.utk.edu/~seelab>

```
Current Threshold: | 0.80 |        Number of Vertices: 2124        Number of Edges: 255783
VQ1 = [ 1   , 778 ] [  -1  ,  43  ] [-111, 200 ] [-1   ,2700] [  -1   ,  36  ] [ 0.00 , 0.78 ]
VQ2 = [ 1   , 778 ] [  -1  ,  43  ] [-111, 200 ] [-1   ,2700] [  -1   ,  36  ] [ 0.00 , 0.78 ]
VQ3 = [ 1   , 778 ] [  -1  ,  43  ] [-111, 200 ] [-1   ,2700] [  -1   ,  36  ] [ 0.00 , 0.78 ]
Click on the arguments and move the mouse to modify values.
```

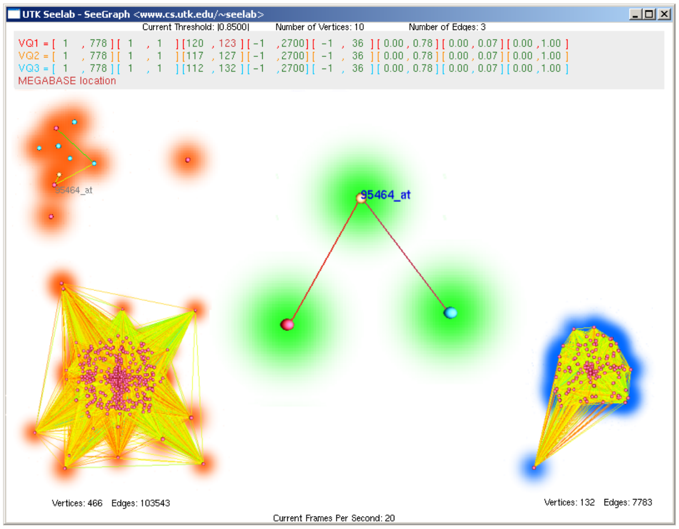
Vertices: 872 Edges: 52396

Vertices: 560 Edges: 116083

Vertices: 448 Edges: 14232

Vertices: 468 Edges: 21581

Figure 3.4: BTD selections (bottom) qualitatively extract gene networks (sides), are rendered using dynamic level-of-detail (center left), and used for template-based classification of entire subgraphs in the original data (center right) for other regulatory mechanisms



```
UTK Seelab - SeeGraph <www.cs.utk.edu/~seelab>
            Current Threshold: |0.8500|        Number of Vertices: 10         Number of Edges: 3
VQ1 = [  1   , 778 ] [  1   ,  1   ] [120  , 123 ] [ -1   ,2700] [  -1  ,  36  ] [ 0.00 , 0.78 ] [ 0.00 , 0.07 ] [ 0.00 ,1.00 ]
VQ2 = [  1   , 778 ] [  1   ,  1   ] [117  , 127 ] [ -1   ,2700] [  -1  ,  36  ] [ 0.00 , 0.78 ] [ 0.00 , 0.07 ] [ 0.00 ,1.00 ]
VQ3 = [  1   , 778 ] [  1   ,  1   ] [112  , 132 ] [ -1   ,2700] [  -1  ,  36  ] [ 0.00 , 0.78 ] [ 0.00 , 0.07 ] [ 0.00 ,1.00 ]
MEGABASE location

   Vertices: 466   Edges: 103543                                                  Vertices: 132   Edges: 7783
                                         Current Frames Per Second: 20
```

Figure 3.5: In this screenshot, two gene networks (bottom left and right) have been discovered with a single putatively corregulating gene as a potential target of knock-out study (center) with proximity information for other potential regulatory genes (top left) undergoing further study. This illustrates the discovery of candidate genes which can affect expression of several genes throughout the genome that play a role in the locomotor response of mice exposed to methamphetamine and cocaine.

Chapter 4

Pairwise Axis Ranking for Parallel Coordinates of Large Multivariate Data

4.1 Introduction

Visualization and computational tools are necessary for the analysis of large multivariate data. Parallel coordinates rendering has proven very useful in this area as it allows intuitive visualization of a multidimensional attribute space. It is very natural to visually "chain together" a sequence of variables in a layout that scales linearly with data dimension. Ideally, a parallel coordinate rendering should provide a view containing sufficient information to guide users to the most interesting parts of the underlying data. There are numerous challenges addressed to varying degrees by current literature relating to proper axis ordering, axis scaling, axis shape, number of axes, rendering technique, clutter reduction, interactivity, etc. In this paper, we aim to address the problem of automatically selecting the proper order of axes by ranking them based upon an underlying system of metrics which specifies relationships between each of the variables.

The traditional approach for axis ordering is to rely on the user to drag axes into positions to discover and elucidate a desired pattern. This goal is quite achievable when the data at hand is manageably small in terms of the total number of data points and variables that must be considered at any particular time. However, without sufficient computational tools, this task is quite daunting as we increasingly need to handle datasets with hundreds or thousands of variables. Our goal is to systematically address the axis ordering problem with scalable algorithmic methods.

A parallel coordinates plot (PCP) can only show a handful of axes on most screens without cognitively overloading the user or obscuring patterns due to visual clutter. Therefore, the task of selecting a relatively small subset of increasingly large multivariate datasets becomes ever more complex. Indeed, the ability to choose the "right" handful of multivariate relationships can be highly context-specific. To approach this, we note that users innately leverage spatial locality and relate a given attribute to at most two (the axes before and after) other attributes in a PCP. By doing so, we reduce the problem space of multivariate relationships down to the sequence of bivariate relationships which PCPs innately represent. We provide a pair of algorithms to optimally or near-optimally rank the variables of an N-variable dataset based upon its $O(N^2)$ bivariate relationship while abstracting the concept of "right" to a user-specifiable metric (correlation, positive skew, etc.).

The novelty of our work stems from our taking an optimization driven perspective to explore the full potential of using parallel coordinates to visualize datasets with a large number of concurrent variables. We provide a pair of algorithms to autonomously find the most interesting patterns. We also devise a PCP rendering method to better reveal patterns based upon underlying bivariate relationships visually in parallel coordinates. We demonstrate our system on IPCC climate simulation data. In total, we show parallel coordinate renderings with axes selected from thousands of variables.

4.2 Metrics

When attempting to determine an optimized axis layout, the criteria that makes such a layout desirable may vary depending upon the task at hand. For example, it is often the case that users want layouts which minimize the amount of visual clutter so that patterns can be easily detected; on the other hand, researchers testing their own clutter reduction methods may be interested in testing on only the most cluttered layouts.

We propose that the proper level of abstraction necessary for this type of problem is to optimize based on the level of some user-defined metric which should be designed to capture the property of interest. In this way, we propose a general set of algorithms which can be applied for truly optimal axis layout in whichever manner is deemed relevant by the user, codified by a matrix of normalized metric values. To relate parallel coordinates using the common spreadsheet metaphor, each data item corresponds to a row and each column to an axis in the PCP. The final metric value thus encodes the information of interest in how one such dimensional axis relates to all other dimensions for the given data.

The problem then becomes, which metric captures the properties of potential interest for my type of data and the question I want to investigate? While listing all possibilities is innately intractable, we find that a surprising few satisfy most needs along with mechanisms for creating variants suited for a particular purpose. In this section, we focus on a single pair of variables to develop quantitatively defined metrics which capture distinctive patterns in data space. In this paper, we provide the general mechanism to use any given metric to autonomously reveal meaningful patterns in large data. This capability facilitates the user's first need of quickly previewing the most unique pattern in a real-world dataset containing thousands of variables.

Data space is the traditional environment for metric calculation since a metric is mathematically defined as a measure of distance. Metrics used by our system are square matrices with the number of rows and columns equal to the number of data dimensions. As such, there is a plethora of applicable mathematical metric definitions, frequently involving calculation of trans-dimensional variance over all data points. The art of the system is to sufficiently constrain the possible metric space to a measure appropriate for the current investigation. However, beginning users should not be expected to know or to care about individually defining their own golden standard metric. In that regard we provide several useful ones with widespread applicability while leading into possibilities for the more sophisticated user.

One of the most commonly-used data space metrics is correlation. While this is typically measured using Pearson's correlation, we have found that other measures of correlation such as entropy-based mutual information or even literature keyword correlation are better than Pearson's in certain domain-specific contexts. While correlation does not necessarily translate into causation, it is often used as an indicator that can direct an analyst's attention to potentially novel knowledge discovery or warrant further investigation to determine the active mechanism of causation.

More complex non-linear relationships are useful in contexts that warrant increased specificity. For example, general modeling techniques such as Bayes' rule for estimating risk or physics-based models for estimating a property of interest. Once the metric has been calculated and used to extract a dimensional ranking, other dimensions which are highly related to the computed metric are immediately visible and can in turn be used to increase or decrease the complexity of the current model.

4.3 Ranking Algorithms

Our goal is to develop a general system which autonomously generates a near-optimal axis ordering by ranking variables to obviate key bivariate relationships for a given dataset. The only required data is a matrix containing values for each axis pair which roughly denote the importance or strength of a relationship between two attributes based upon a user-selected or computationally defined metric. Here we present a pair of algorithms which provide an optimality/time tradeoff based upon a user's given data size, computational resources, and time constraints.

4.3.1 Optimal Ranking

Once given an $N \times N$ matrix corresponding to all pairwise attribute metrics, the problem can be solved in the domain of graph algorithms by treating it as an adjacency matrix. In this scheme, there is a graph of N vertices and N^2 edges from which we want to extract what we shall refer to as an "optimized k-walk" where k is the number of axes desired in the parallel coordinate plot. This can be seen as a generalization of the Traveling Salesman Problem in which the salesperson must make an optimized visit to $k \leq N$ cities and reduces to the classical problem for $k = N$.

The brute force method for solving an "optimized k-walk" is to simply take every possible N *choose* k subset and permute every subset's k variables to find the maximum sum of edge weights between consecutive pairs. This method is $O((N \ choose \ k) * k!)$ or $O(\frac{n!}{(n-k)!})$ and can be used to find a subset of k values which maximizes the fitness of the resulting ranking derived from this simple optimization equation:

$$\sum_{i=1,k-1} Weight(i, i-1) \tag{4.1}$$

While this method is guaranteed to find a globally optimum axis ranking, it is NP-complete and therefore computationally intractable for all but the smallest datasets. Even for a dataset with only $N = 63$ variables and $k = 7$ axes, if calculating the optimum layout from 7! axis arrangements took only 1 millisecond, the entire calculation would still take approximately 6.5 days (our result using a 2.1Ghz Intel Core 2). We stopped the global search algorithm for $N = 126$ variables using $k = 7$ axes after running 3 months. This algorithm is embarrassingly parallel in that each N *choose* k set of axes could be passed to a core, permutation tested, and respond back with a fitness which hashes into a sorted data struct. However, we felt that this was unnecessary as approximate algorithms would suffice. For this reason, we implemented some simple alternatives to this brute force approach that typically calculate a layout so near to optimal that the difference is negligible.

4.3.2 Greedy Pairs Algorithm

Due to the NP-complete nature of the true optimization problem, we developed several approximation algorithms which make various degrees of fitness/time tradeoffs. We include only one such algorithm here as a simple example upon which other algorithms could be based.

When we calculate an axis layout, it would make sense to keep the pairs with the strongest relationship next to one another rather than adding single axes greedily. In this greedy-based algorithm, shown in figure 4.1, we begin by finding the k largest weights in the graph using the naive selection algorithm in $O(k|V|^2)$ time, where $|V|$ is the number of vertices in the graph (dimensions in the dataset) and k is the number of axes to be shown in the final PCP. While this results in the highest pairwise values, order matters since we would like to string these pairs together in a way that maximizes the total fitness. We choose k pairs, resulting in $2k$ axes, despite only needing k axes because we may have perfect pair overlap. Since each weight has two associated axes, each pair is permuted to find the pairwise sequence which maximizes the sum of weights from the first k consecutive axes (thus selecting those axes discarding any additional axes). This algorithm runs on the order of $k|V|^2 + k!$ and typically performs negligibly close to the pure optimal algorithm. This algorithm stably sorted 734 axes in 3.6 milliseconds on a 2.1Ghz Intel Core 2.

4.4 Rendering

Traditional rendering of parallel coordinates involves rendering the multidimensional data as a series of polylines. The intersections between the polylines and the parallel axes spatially indicate the values for each observation. Ideally, the viewer's visual cognition system will identify patterns in the lines indicating relationships between variables. However, such a simple display easily becomes cluttered for even small datasets and trends are difficult to discern.

The goal of our research is to automatically make trends highly visible to the user. Accordingly, we have developed a novel method of parallel coordinate rendering that emphasizes variable relationships with easily perceived 3-dimensional cues. Instead of treating each line separately, we render the series of lines as a planar surface and shade each point on the surface according to the number of lines that intersect at the point. Our 3-D approach enables the human perceptual system to quickly parse the display to find interesting trends, which may emerge as ridges or valleys.

Our renderer first rasterizes all polylines and calculates the depth complexity, or the number of times each pixel is drawn into. Pixels of high complexity represent locations where many lines intersect. We then use this depth complexity image to calculate a normal map. The lines are cleared and a single bump-mapped quadrilateral is drawn in their place. Each fragment's normal is retrieved from the normal map texture, and its depth complexity is used to index into an RGBA transfer function. The normal and color are used to perform traditional Phong lighting.

Example renderings of our method for two artificial datasets are shown in figure 4.2. Occlusion in traditional line renderings often masks or subdues trends. By enhancing the parallel coordinate display with a normal map derived from the depth complexity image, these trends are strongly emphasized through color and specular and diffuse lighting. For the uncorrelated dataset in (a), the depth complexity image has nearly constant slope, yielding slow color changes and flat surfaces in the enhanced rendering. The correlated dataset in (b), however, contains a strong ridge where many observations overlap.

The resulting heightfield can be scanned quickly for prominent features or manipulated by the user. The light source can be translated interactively in three dimensions to investigate the surface cues through shading changes. The opacity of regions of low or high complexity can be modulated with a transfer function widget. Since our approach uses color to denote depth complexity, individual lines are not colored separately. To support this, we allow the user to further modulate opacity with a second transfer function indexed by each line's value on the selected axis.

4.5 Results

We have tried many metrics but will use Pearon's correlation in results for this section as it is the easiest to visually verify. Since our optimization framework maximizes the metric under various constraints, a target of 1 or -1 will yield highly correlated or inversely correlated results. If you set the target to $[-0.2, 0.2]$, then the system will instead show things that are mostly uncorrelated, very dissimilar from [-1,1]. We will showcase some of our results in sections 4.5.2- 4.5.3 after first describing a dataset currently undergoing active exploration in section 4.5.1.

4.5.1 Climate Simulation Data

We use the greedy pair algorithm to determine an optimal ordering of axes based upon correlation in the following examples. The proposed system is compatible with any multivariate data and the examples presented in this section will utilize climate data. The climate data used here contains 63 physical variables recorded on a monthly basis for 10 years (2000-2009) of IPCC climate simulation. In total, we consider $63 \times 12 \times 10 = 7560$ attribute dimensions by treating time steps independently. Moreover, the simulation grid corresponding to land points constitutes 7,311 polylines for each axis layout. Most systems will not contain so many dimensions, but we wanted to demonstrate the speed and flexibility of our system on such a large, real-world dataset. Due to this large datasize, we use a greedy pairs algorithm for the most timely performance unless otherwise noted.

Throughout this paper, we typically use 7 axes based upon the limits of human cognition to 7 units of information for short-term memory [Miller, 1956].

4.5.2 Ostentatious Patterns

When we use the Pearson's correlation metric on the climate data, the system returns the most highly correlated variables. In this example we begin with only 63 climate variables for January of 2000 and compute the truly globally optimum layout using the correlation metric. As shown in figure 4.3, the system has detected several variables which are highly correlated. This type of plot verifies the system is working correctly since all these variables are differing measures of temperature that should be roughly correlated; left to right these are: TREFMXAV-maximum average temperature at two meters, ZBOT-temperature and humidity at two meters, TBOT-atmospheric air temperature, THBOT-atmospheric air potential temperature, TV-vegetation temperature, TSA-air temperature at two meters, and TG-ground temperature.

In this example, we add the temporal dimension for an order of magnitude more variables. Since we treat variables from separate timesteps independently, the system has selected a layout which shows the common-sense relation of an attribute's self-correlation across disparate timesteps. As shown in figure 4.4, snowfall during the summer months of 2000-2009 is somewhat consistent. This example was chosen for a few reasons. First, it highlights the fact that the system can detect patterns which may be surprising and unexpected, prompting the user to create metric variants through constraints. Second, while repeating axes are not allowed in this example, additional removal of correlation with other variables throughout time can be a desirable property and is supported by our system. Third, the years for the layout axes are unordered and a constraint which has time increasing to the right may be more intuitive.

4.5.3 Constraints for Innate Patterns

By taking the above constraints into account, a user may now be interested in seeing inversely correlated variables (by subtracting the constrained matrix from a matrix of ones). The result can be seen in figure 4.5. RSSUN/RSSHA are measures of leaf stomatal resistance which is dependent upon the incident photosynthetically active radiation. In this PCP, the system has selected axes which are inversely proportional in equidistant months (alternating April/October) in which the direct rays of the sun are at their maximal relative difference. This example shows that the system can display patterns which are novel and interesting to naive users but which make complete sense to domain-specific experts.

Global warming is a common concern that scientists attempt to verify and understand when looking at climate data. One of many ways to gauge global warming is in the variance of snow depth throughout many years. By using correlation, our system produced the PCP shown in figure 4.6 which shows a strong correlation in snow depth throughout the years. As may be expected, there are many locations which have no snow (red line at bottom), a few that have a little snow (blue area), and more that are typically covered in snow (green area). However, there are some highlighted ridges in our rendering corresponding to grid points whose snow depth have varied significantly and should be checked for location of important polar ice caps.

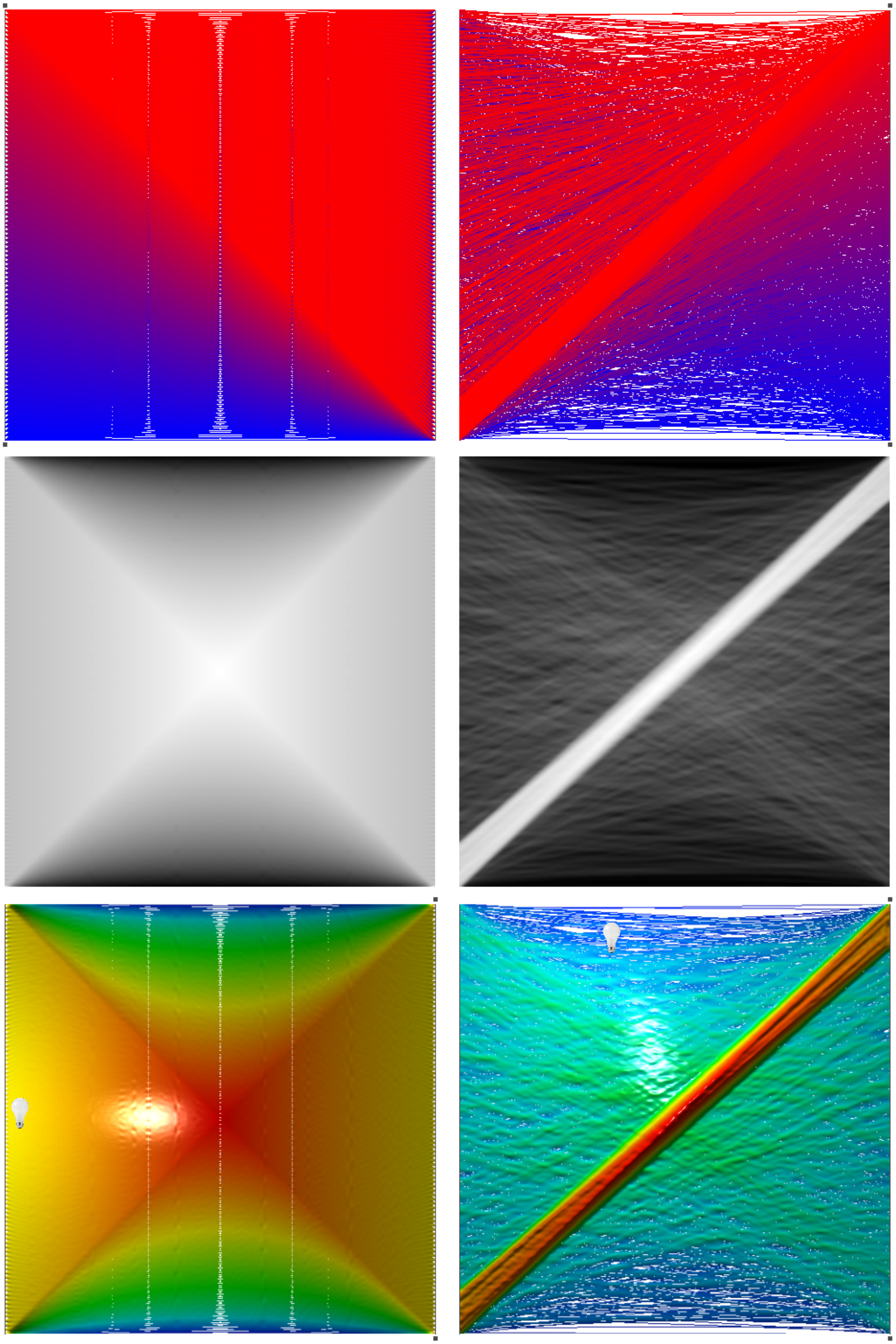
```
int getFitness(int numAxes,
               int* layout, graph* g) {
  int i, v1,v2, fitness=0;
  for(i=1; i<numAxes; i++) {
     v1=layout[i-1]; v2=layout[i];
     fitness += g->adjmat[v1][v2];
  }
  return fitness;
}

int getHighestEdges(int numAxes, graph* g,
     int** tX, int** tY, int allowRepeats) {
  int i,j,k, val,max, axis,got;
  int tarX[numAxes], tarY[numAxes];
  for(axis=0; axis<numAxes; axis++) {
     max=SHRT_MIN; // find k best pairs
     for(i=0;i<g->numVerts;i++)
       for(j=0;j<i;j++) { //entire matrix
         got=check(tarX,tarY,i,j,allowRepeats);
         val = g->adjmat[i][j]; //new max?
         if (val!=INAN && got==0 && max<val) {
             max=val; tarX[axis]=j; tarY[axis]=i;
         }//end max check
     }//end O(V^2)
  }//end naive k best pairs O(kV^2)
  *tX = tarX; *tY = tarY; return 0;
}

int* maxPermPairs(int num, int* tarX, int* tarY,
                  graph* g) {
   int i,k, max, fitness = -1;
   int a[num], layout[num+1],bestLayout[num+1];
   for(i=0; i<num; i++) a[i]=i;
   for_all_permutations_of_a[i] {
      for(k=0;k<(num+1)/2;k++) { //best perm?
        layout[2*k  ] = tarX[a[k]];
        layout[2*k+1] = tarY[a[k]];
      }
      if((num+1)%2==1)
        layout[num]=tarX[a[(num+1)/2]];
      fitness=getFitness(num+1,layout,g);
      if(fitness>max) {
         max = fitness;
         memcpy(bestLayout,layout);
      }
   } // End O(num!)
   return bestLayout;
}

int* greedyPairs(int numAxes=7, graph* g) {
   int *tarX, *tarY;
   getHighestEdges(numAxes, g, &tarX, &tarY, 0);
   return maxPermPairs(numAxes, tarX, tarY, g);
}
```

Figure 4.1: Pseudocode for the quick, near-optimal greedy pairs algorithm.



(a) Uncorrelated

(b) Highly Correlated

Figure 4.2: Detecting trends in parallel coordinate displays made easier with 3-D surface cues. **(top)** Traditional line rendering of two generated datasets. Column (a) represents an extremely uncorrelated dataset where every data item on the first axis is connected to every data item on the second. Column (b) is a dataset where half of the observations are randomly generated and half are randomly offset from an inverse relationship. **(middle)**

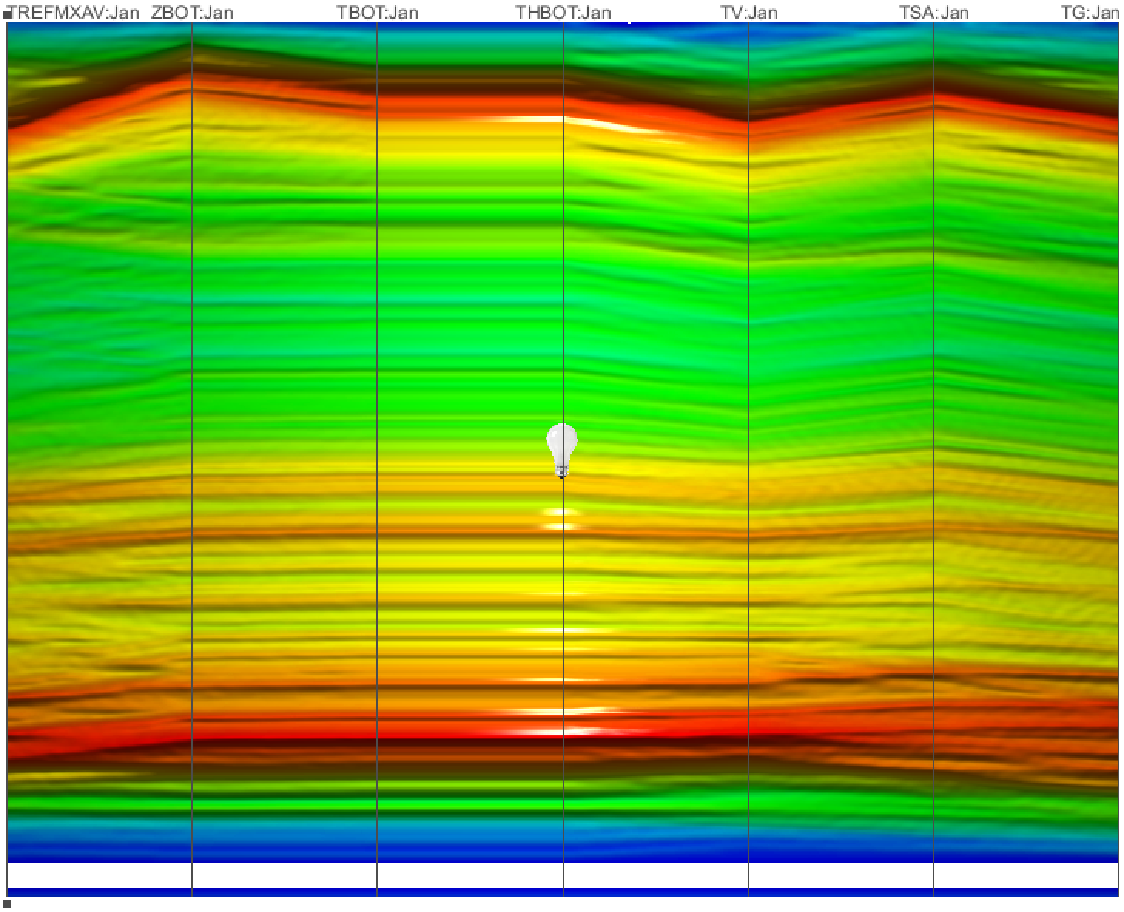


Figure 4.3: The system finds a strong correlation between various measures of temperature in Jan'00.

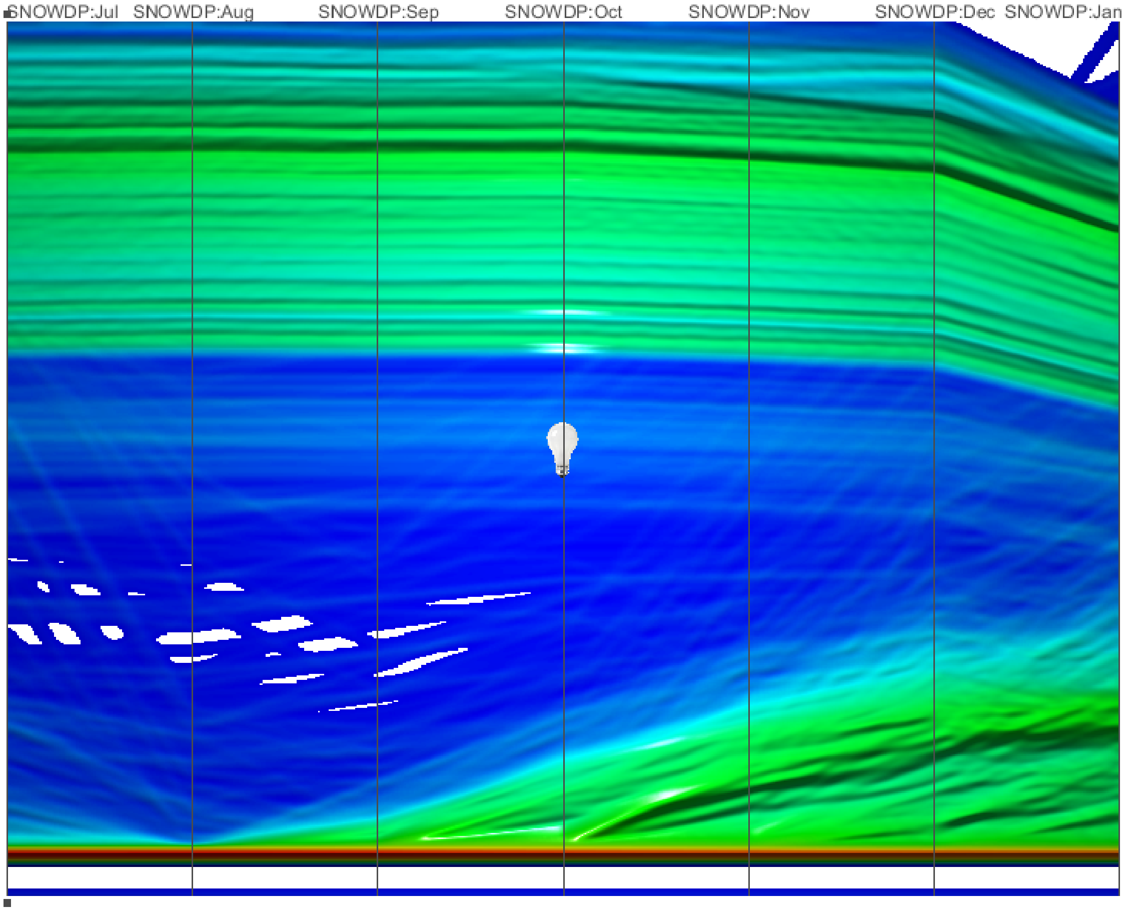


Figure 4.4: Constraints are included to keep the system from finding repeated results of self-correlation through time.

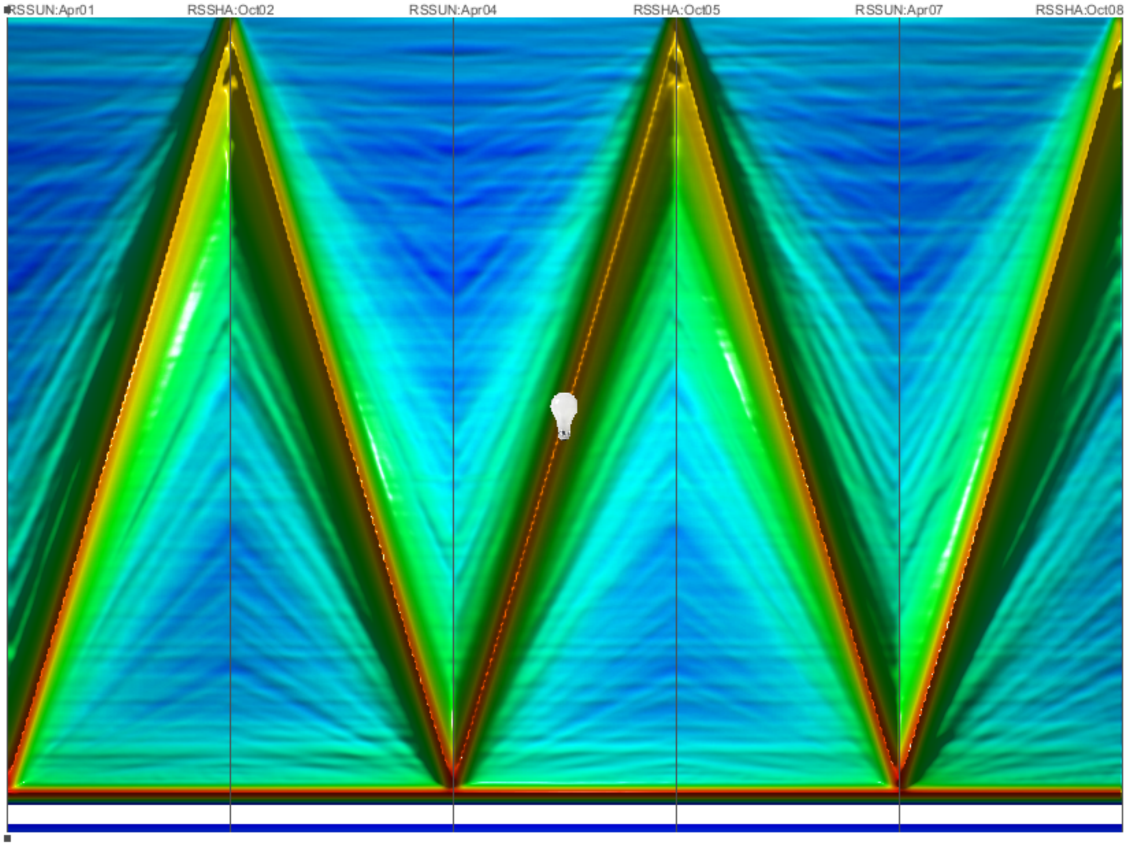


Figure 4.5: Inverse correlation with consistent time constraints which relates the variance of radiation intensity on leaves as a function of the earth's tilt throughout the seasons.

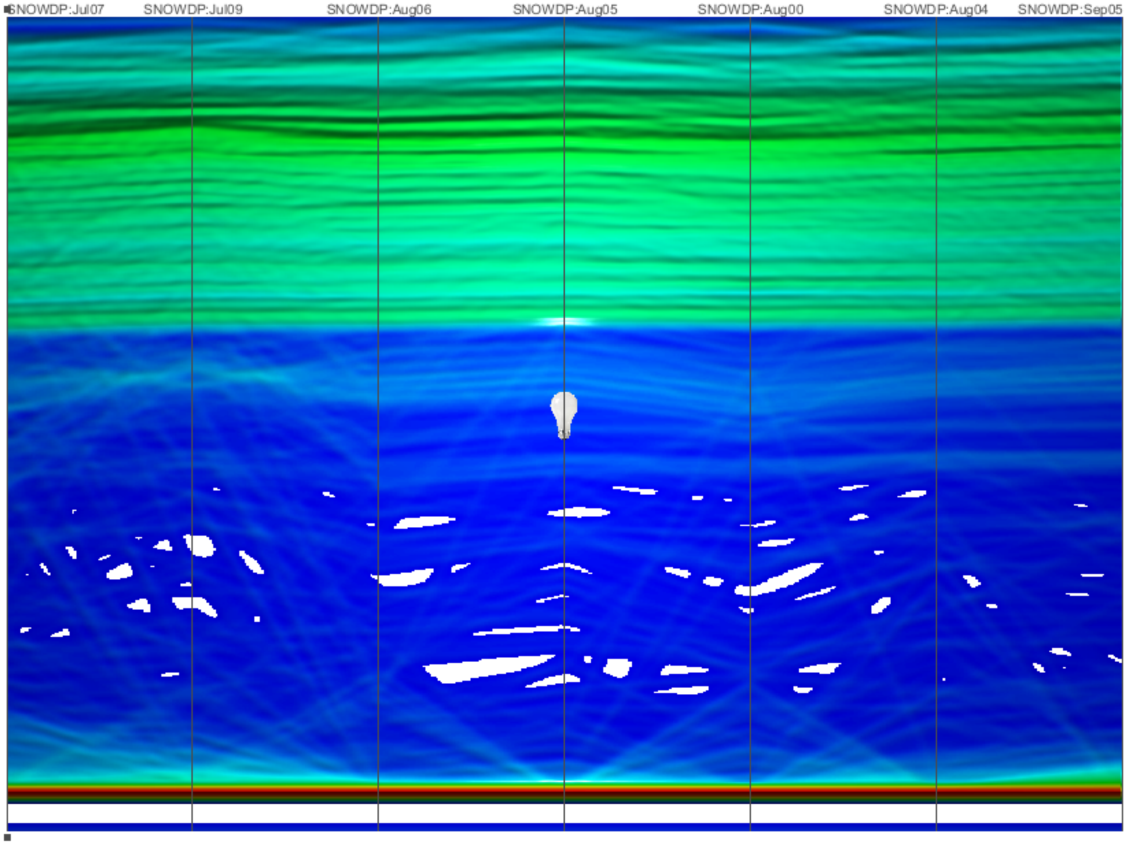


Figure 4.6: One way of measuring global warming showing strong correlation of snow depth between years. Our rendering technique also shows V-shaped highlights corresponding to grid locations that my warrant further investigation for snow/ice melting.

Chapter 5

Opening the SFAM Black Box (demonstrated using CSI/climate data)

Chapter 6

Conclusions

As with most research, the number of possible directions, ideas, and contributions increase exponentially as time progresses. During my stay, I have compiled a long list of features that people have recommended, I have considered building upon, or other ideas which I haven't a clue how to address yet. I will outline some of these here along with possibilities of using the general software package developed to address multiple other problems in various fields.

6.1 Dynamic Visualization of Coexpression

In conclusion, it has been shown that the integrated computational tools not only provide research scientists and analysts a way to visualize their data, but it also allows complex querying and filtering for drill-down, graphical analysis, and statistical output. Each of these are facilitated by combinations of a 3D spring-embedded layout, efficient B-tree processing, neural networks, matrix operations, and graph algorithms.

While our visualization tool enables significant biological discoveries, its full potential can only be leveraged when used in combination with mature applications and data management systems for genetical genomics datasets. For this purpose, our future work includes integration with the Gaggle API [Shannon et al., 2006], a web-enabled, platform-independent, multi-application, data-sharing framework for widespread use among systems biologists available at gaggle.systemsbiology.org. In addition to the linked viewports framework, we are also considering domain-specific hybrid visualizations such as [Henry et al., 2007]. This tool is also currently being integrated as a correlation visualization tool to complement the genome-phenome data integration tools in the Ontological Discovery Environment (ODE) suite at ontologicaldiscovery.org. Ongoing work is currently being conducted with biologists to implement new functionality relating to domain-specific requests for handling linkage disequilibrium, QTL analysis, integration of genetic information across multiple scales, multiple time points, and different graph types.

6.2 Axis Ranking for Parallel Coordinates

In conclusion, we have provided a general mechanism for the optimized ranking for axis ordering in parallel coordinates visualizations along with algorithms to manage sub-optimal ranking tradeoff for time depending upon data size. We have developed a depth-enhanced parallel coordinate renderer that uses surface cues to more effectively display trends over traditional line drawings. The results demonstrated clearly show the automatic ranking and selection of meaningful patterns in PCP axes for large, time-dependent, multivariate, climate data.

There are several partially completed items that we consider as potential future work. First, formalize work on defining constraints on the bivariate matrix to provide intuitive control for users while pruning the search space. Second, we have already coded up a learning system to help users determine which metrics are important for a specific pattern of interest and now analyzing the results. Third, there are several other graph-based algorithms which could be used for determining an axis ordering such as pairwise shortest path between two specific variables of interest. Fourth, the optimization framework presented here optimizes on the basis of bivariate relationships but a more general multivariate trend detection mechanism would allow the detection of nonlinear (sinusoidal) patterns.

Bibliography

Bibliography

[Abello and Korn, 2002] Abello, J. and Korn, J. (2002). Mgv: A system for visualizing massive multidigraphs. *IEEE Trans. Visualization and Computer Graphics*, 8(1):21–38.

[Abello et al., 2006] Abello, J., van Ham, F., and Krishnan, N. (2006). Ask-graphview: A large scale graph visualization system. *IEEE Trans. Visualization and Computer Graphics*, 12(5):669–677.

[Abiola et al., 2003] Abiola, O., Angel, J. M., Avner, P., Bachmanov, A. A., and et al., J. K. B. (2003). The nature and identification of quantitative trait loci: a community's view. *Nature Reviews Genetics*, 4(11):911–916.

[Auber et al., 2003] Auber, D., Chiricota, Y., Jourdan, F., and Melancon, G. (2003). Multiscale visualization of small world networks. In *IEEE Syposium on Information Visualization*, pages 75–81.

[Bai et al., 2004] Bai, Y., Gansterer, W. N., and Ward, R. C. (2004). Block tridiagonalization of effectively sparse symmetric matrices. *ACM Trans. Math. Softw.*, 30(3):326–352.

[Bru and Frick, 1996] Bru, I. and Frick, A. (1996). Fast interactive 3-d graph visualization. In *Proceedings of Graph Drawing '95*, pages 99–110. Springer-Verlag.

[Carpenter and Grossberg, 1987] Carpenter, G. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37(1):54–115.

[Carpenter et al., 1991] Carpenter, G., Grossberg, S., and Reynolds, J. (1991). Artmap: Supervised real-time learning and classification of stationary data by a self-organizing neural network. *Neural Networks*, 4:565–588.

[Chesler and Langston, 2005] Chesler, E. J. and Langston, M. A. (2005). Combinatorial genetic regulatory network analysis tools for high throughput transcriptomic data. In *RECOMB Satellite Workshop on Systems Biology and Regulatory Genomics*, pages 150–165.

[Chesler et al., 2005] Chesler, E. J., Lu, L., Shou, S., Qu, Y., Gu, J., Wang, J., Hsu, H. C., Mountz, J. D., Baldwin, N. E., Langston, M. A., Hogenesch, J. B., Threadgill, D. W., Manly, K. F., and Williams, R. W. (2005). Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, 37(3):233–242.

[Chesler et al., 2003] Chesler, E. J., Wang, J., Lu, L., Qu, Y., Manly, K. F., and Williams, R. W. (2003). Genetic correlates of gene expression in recombinant inbred strains: a relational model system to explore neurobehavioral phenotypes. *Neuroinformatics*, 1(4):343–357.

[Davidson et al., 2001] Davidson, G. S., Wylie, B. N., and Boyack, K. W. (2001). Cluster stability and the use of noise in interpretation of clustering. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 23–30. IEEE Computer Society.

[Doerge, 2002] Doerge, R. W. (2002). Mapping and analysis of quantitative trait loci in experimental populations. *Nature Reviews Genetics*, 3(1):43–52.

[Ellis and Dix, 2006] Ellis, G. and Dix, A. (2006). Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724.

[Ferreira and Levkowitz, 2003] Ferreira, M. C. and Levkowitz, H. (2003). From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 09(3):378–394.

[Fruchterman and Reingold, 1991] Fruchterman, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164.

[Fua et al., 1999] Fua, Y. H., Ward, M. O., and Rundensteiner, E. A. (1999). Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Visualization '99*, pages 43–50.

[Gantz et al., 2007] Gantz, J., Reinsel, D., Chute, C., Schlichting, W., McArthur, J., Minton, S., Xheneti, I., Toncheva, A., and Manfrediz, A. (2007). Idc - the expanding digital universe: A forecast of worldwide information growth through 2010. Technical report.

[Geschwind, 2000] Geschwind, D. H. (2000). Mice, microarrays, and the genetic diversity of the brain. *Proc. National Academy of Sciences*, 97(20):10676–10678.

[Glatter et al., 2006] Glatter, M., Mollenhour, C., Huang, J., and Gao, J. (2006). Scalable data servers for large multivariate volume visualization. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1291–1298.

[Graham and Kennedy, 2003] Graham, M. and Kennedy, J. (2003). Using curves to enhance parallel coordinate visualisations. In *IV '03: Proceedings of the Seventh International Conference on Information Visualization*, pages 10–16.

[Grisel et al., 1997] Grisel, J. E., Belknap, J. K., O'Toole, L. A., and et al., M. L. H. (1997). Quantitative trait loci affecting methamphetamine responses in bxd recombinant inbred mouse strains. *Journal of Neuroscience*, 17(2):745–754.

[Gross et al., 2004] Gross, J. L., Yellen, J., Burke, Edmund, de Werra, Dominique, Kingston, and Jeffrey (2004). CRC Press.

[Hachul and Junger, 2004] Hachul, S. and Junger, M. (2004). The fast multipole multilevel method. *GD '04: Proceedings of the Symposium on Graph Drawing*, pages 286–293.

[Hargrove and Hoffman, 2004] Hargrove, W. and Hoffman, F. (2004). Potential of multivariate quantitative methods for delineation and visualization of ecoregions. *Environmental Management*, 34(5):39–60.

[Henry et al., 2007] Henry, N., Fekete, J., and McGuffin, M. J. (2007). Nodetrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309.

[Henry and Fekete, 2006] Henry, N. and Fekete, J. D. (2006). Matrixexplorer: a dual-representation system to explore social networks. *IEEE Trans. Visualization and Computer Graphics*, 12(5):677–685.

[Hoffman et al., 2005] Hoffman, F., Hargrove, W., Erickson, D., and Oglesby, R. (2005). Using clustered climate regimes to analyze and compare predictions from fully coupled general circulation models. *Earth Interactions*, 9(10):1–27.

[Hu et al., 2004] Hu, Z., Mellor, J., Wu, J., and DeLisi, C. (2004). Visant: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, pages 5–17.

[Inselberg, 1985] Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91.

[Inselberg and Dimsdale, 1990] Inselberg, A. and Dimsdale, B. (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of IEEE Visualization'90*, pages 361–378.

[Inselberg and Dimsdale, 1994] Inselberg, A. and Dimsdale, B. (1994). Multidimensional lines i: Representation. *SIAM J. Appl. Math.*, 54(2):559–577.

[Jansen and Nap, 2001] Jansen, R. C. and Nap, J. P. (2001). Genetical genomics: the added value from segregation. *Trends in Genetics*, 17(7):388–391.

[Johansson et al., 2005a] Johansson, J., Cooper, M., and Jern, M. (2005a). 3-dimensional display for clustered multi-relational parallel coordinates. In *IV '05: Proceedings of the Ninth International Conference on Information Visualisation*, pages 188–193.

[Johansson et al., 2005b] Johansson, J., Ljung, P., Jern, M., and Cooper, M. (2005b). Revealing structure within clustered parallel coordinates displays. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 17.

[Jones et al., 1999] Jones, B. C., Tarantino, L. M., Rodriguez, L. A., and et al., C. L. R. (1999). Quantitative-trait loci analysis of cocaine-related behaviours and neurochemistry. *Pharmacogenetics*, 9(5):607–617.

[Kamada and Kawai, 1989] Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15.

[Kreuseler and Schumann, 2002] Kreuseler, M. and Schumann, H. (2002). A flexible approach for visual data mining. *IEEE Trans. Visualization and Computer Graphics*, 8(1):39–51.

[Kumar et al., 1999] Kumar, S. R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling emerging cyber-communities automatically. In *Proc. 8th Intl World Wide Web Conf.*

[Langston et al., 2006] Langston, M. A., Perkins, A. D., Saxton, A. M., Scharff, J. A., and Voy, B. H. (2006). Innovative computational methods for transcriptomic data analysis. In *SAC'06: Proceedings of the 2006 ACM Symposium on Applied Computing*, pages 190–194, New York, NY, USA. ACM Press.

[Miller, 1956] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.

[Moustafa and Wegman, 2002] Moustafa, R. E. A. and Wegman, E. J. (2002). On some generalizations of parallel coordinate plots. In *Seeing a Million: A Data Visualization Workshop.*

[Mueller et al., 2007] Mueller, C., Martin, B., and Lumsdaine, A. (2007). A comparison of vertex ordering algorithms for large graph visualization. *International Asia-Pacific Symposium on Visualization*, pages 141–148.

[Mutton and Rodgers, 2002] Mutton, P. and Rodgers, P. (2002). Spring embedder preprocessing for www visualization. *IEEE Syposium on Information Visualization*, 00:744–749.

[Noack, 2004] Noack, A. (2004). An energy model for visual graph clustering. *GD '04: Proceedings of the Symposium on Graph Drawing*, pages 425–436.

[Novotny, 2006] Novotny, M. (2006). Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900.

[Peng et al., 2004] Peng, W., Ward, M. O., and Rundensteiner, E. A. (2004). Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 89–96. IEEE Computer Society.

[Perer and Shneiderman, 2006] Perer, A. and Shneiderman, B. (2006). Balancing systematic and flexible exploration of social networks. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):693–700.

[Raymond et al., 2002] Raymond, J., Gardiner, E., and Willett, P. (2002). Rascal: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644.

[Russell and Norvig, 2002] Russell, S. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach (2nd Ed)*. Prentice Hall.

[Schrock et al., 1996] Schrock, E., Manoir, S. D., Veldman, T., Schoell, B., Wienberg, J., Ferguson-Smith, M. A., Ning, Y., Ledbetter, D. H., Bar-Am, I., Soenksen, D., Garini, Y., and Ried, T. (1996). Multicolor spectral karyotyping of human chromosomes. *Science*, 273:494–497.

[Shannon et al., 2003] Shannon, P. T., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 11:2498–504.

[Shannon et al., 2006] Shannon, P. T., Reiss, D. J., Bonneau, R., and Baliga, N. S. (2006). The gaggle: an open-source software system for integrating bioinformatics software and data sources. *BMC Bioinformatics*, 7:176.

[Shen et al., 2006] Shen, Z., Ma, K. L., and Eliassi-Rad, T. (2006). Visual analysis of large heterogeneous social networks by semantic and structure. *IEEE Trans. on Visualization and Computer Graphics*, 12(6):1427–1439.

[Sheng et al., 1999] Sheng, L., Ozsoyoglu, Z. M., and Ozsoyoglu, G. (1999). A graph query language and its query processing. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Austrialia*, pages 572–581. IEEE Computer Society.

[Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages*, number UMCP-CSD CS-TR-3665, pages 336–343, College Park, Maryland 20742, U.S.A.

[Shneiderman, 2006] Shneiderman, B. (2006). Network visualization by semantic substrates. *IEEE Trans. Visualization and Computer Graphics*, 12(5):733–741.

[Steed et al., 2007] Steed, C. A., Fitzpatrick, P. J., Jankun-Kelly, T. J., and Yancey, A. N. (2007). Practical application of parallel coordinates to hurricane trend analysisy. In *Proceedings of IEEE Visualization'07*.

[Tarini et al., 2006] Tarini, M., Cignoni, P., and Montani, C. (2006). Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244.

[Thomas and Cook, 2005] Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr.

[Tzeng et al., 2003] Tzeng, F. Y., Lum, E. B., and Ma, K. L. (2003). A novel interface for higher-dimensional classification of volume data. In *Proceedings of IEEE Visualization '03*, pages 505–512.

[van Ham and van Wijk, 2004] van Ham, F. and van Wijk, J. (2004). Interactive visualization of small world graphs. In *IEEE Syposium on Information Visualization*, pages 199–206.

[Wang et al., 2003] Wang, J., Williams, R. W., and Manly, K. F. (2003). Webqtl: web-based complex trait analysis. *Neuroinformatics*, 1(4):299–308.

[Wegenkittl et al., 1997] Wegenkittl, R., Loffelmann, H., and Groller, E. (1997). Visualizing the behavior of higher dimensional dynamical systems. In *Proceedings of IEEE Visualization '97*, pages 119–126.

[Wegman and Luo, 1996] Wegman, E. J. and Luo, Q. (1996). High dimensional clustering using parallel coordinates and the grand tour. Technical Report 124.

[Wegman, 1990] Wegman, J. E. (1990). Hyperdimensional data analysis using parallel coordinates. *J. Am. Stat. Assn*, 85(411):664–675.

[Zhang et al., 2005] Zhang, B., Kirov, S., Snoddy, J., and Ericson, S. (2005). Genetviz: A gene network visualization system. In *UT-ORNL-KBRIN Bioinformatics Summit*.

[Zhou et al., 2007] Zhou, H., Yuan, X., Chen, B., and Qu, H. (2007). Visual clustering in parallel coordinates. In *Proceedings of IEEE Visualization'07*.

Curriculum Vitae
Joshua New

Contact Information

Department of Electrical Engineering & Computer Science (EECS)
University of Tennessee
212 Claxton Complex, 1122 Volunteer Blvd.
Knoxville, TN 37996

Cell: (423) 653-6384
Office: (865) 974-5400
Email: new@cs.utk.edu
URL: www.cs.utk.edu/∼new

Education

The University of Tennessee, Knoxville (UTK) Knoxville, TN
May 2009 Ph.D. candidate in Computer Science, Visualization
Current GPA: 3.9/4.0; Ph.D. Adviser: Professor Jian Huang

Jacksonville State University (JSU) Jacksonville, AL
May 2004 M.S. in Computer Systems and Software Design
Graduate GPA: 4.0/4.0; Outstanding MCIS Graduate Award

May 2001 B.S. in Computer Science and Mathematics with Physics minor
Undergraduate GPA: 3.5/4.0

Technical Skills

Platforms: Windows 3.1/95/98/2000/CE/ME/NT/XP/Vista, MS-DOS; SGI Irix
Linux Ubuntu/Debian/Red Hat/Fedora/SUSE; Mac OS X; Sun Solaris

Languages: C, C++, Java, Matlab, Fortran/90, Visual Basic, Ada, Lisp, R, Mathematica,
Tcl/Tk, Perl, Python, PHP, HTML; dBase, SQL, VBA;
WinBatch, DOS batch, shell scripting (sh,csh,tcsh,bash,ksh,zsh)

Libraries: Open-source Graphics Library (*OpenGL*), Visualization ToolKit (*VTK*),
system-independent GUI toolkit (*Qt*), Message Passing Interface (*MPI*),
UTK SeeLab's Viz CookBook (*VCB*), Intel's Image Processing Libraries (*IPL*),
computer vision library (*OpenCV*)

Professional Experience

Title: Graduate Research Assistant
Employer: SeeLab, The University of Tennessee at Knoxville
Dates: February 2005-present

- Developed SeeShader, a glut-based framework for easily implementing General Purpose computation on Graphical Processing Units (*GPGPU*) applications
- Developed SeeBrain, a GPU-accelerated DT-MRI fiber bundle extraction, visualization, and query system for real-time, multi-dataset analysis in collaboration with Vanderbilt radiologists.
- Other primary projects mentioned in ORNL internships below.

Title: Research Intern
Employer: Oak Ridge National Laboratory
Dates: Summers 2005, 2007, 2008

- Developed SeeGraph, a general 3D layout and graph visualization package applied to phenotype correlation of mice by ORNL biologists; extended with graph and matrix algorithms, image processing, statistics, data mining, and novel interaction techniques for microarray data analysis.
- ORNL/UTK collaboration to leverage supercomputers for climate simulation and prediction using parallel clustering for ecoregion classification, visualization with GIS tools, multivariate data analysis using parallel coordinate plots, and maintenance of parallel model fitting software.
- Helped develop a suite of new visualization tools for large volumetric data that can be used by practitioners and application scientists on arbitrarily-sized clusters driving any tiled display.

Title: General Tools Intern
Employer: Vital Images, Inc.
Dates: Summer 2006

- Enhanced usability of Vitrea, a workstation product used worldwide to inspect medical imagery, with changes to tools including 3D angiography, vessel probe and measurements, brain perfusion, cardiac, colonography, and lung analysis as well as 5-button mouse support for Qt.
- Disambiguated click-and-drag behavior for tool mode versus rotation mode using several event-emulation bypass methods customizable using XML on a per-tool basis. Usability testing of bypass methods was conducted and later deployed in the public release.

Title: Graduate Teaching Assistant
Employer: The University of Tennessee at Knoxville
Dates: September 2004-January 2005

- Developed, provided, organized, and graded course material for UTK's graduate level CS552 - Software Engineering under the guidance of Dr. Stacy Prowell.
- Developed grading rubrics for exam material, mentored groups on deliverables for the semester-long project, and lectured regularly.

Title: Graduate Research Assistant
Employer: Knowledge Systems Laboratory, Jacksonville State University
Dates: September 2001-May 2004

- Developed Med-LIFE, a medical system for visualization and interaction with MRI data allowing neurophysiologically-based image processing and fusion, interactive learning agents for database mining and disease segmentation, contextual zoom, and 3D isosurface extraction.
- Implemented a gesture recognition for applied computer vision to live camera feed for direct manipulation of a 3D object via spatial and temporal hand gestures.

Title: Computer Specialist
Employer: Ft. McClellan
Dates: June 1997-August 2001

- Primary contributor to the inventory tracking software used to close the military post which was responsible for the successful transfer of 11,200,000 tons of material in 215 shipments amounting to a total value of approximately $104,314,000. Installed and customized for use by the Department of Homeland Security, Wastren Inc., and the Joint Powers Authority.
- Expert knowledge and application in Microsoft Access with Visual Basic and much experience using SQL, Word, Excel, PowerPoint, Paint Shop Pro, Visio, and similar software packages.
- Software/hardware/network administrator and troubleshooter, responsible for proper operation and communication of 40 computers in the personal property section of the Transition Force.

Awards

2nd Place - Best ACM Mid-Southeastern Chapter graduate paper presentation	2003
Outstanding Graduate in Mathematics Award, Jacksonville State University	2001

Professional Services

Conference Committee	IEEE Visualization Conference	2008–2009
Student Volunteer	IEEE Visualization Conference	2006–2009
Referee	IEEE Trans. on Vis. and Computer Graphics	2005–2009

Association Memberships

IEEE (Institute of Electrical and Electronics Engineers)	2004–Present
ACM (Association for Computing Machinery) and local chapter	2004–Present
Student Chapter of Mathematical Association of America	1997–2002
President of Student Chapter of Mathematical Association of America	1998–2001
Student Chapter of Association of Information Technology Professionals	1997–2002
American Institute of Physicists	1998–2002

Selected Publications

1. **Joshua New**, Wesley Kendall, Jian Huang, Elissa Chesler, "Dynamic Visualization of Gene Coexpression in Systems Genetics Data", IEEE Transactions on Visualization and Computer Graphics, **14**(5), pp. 1081-1094, 2008.

2. Jian Huang, Markus Glatter, Wesley Kendall, Brandon Langley, **Joshua New**, Roberto Sisneros, Forrest Hoffman, David Erickson, "Time-Varying Multivariate Visualization for Understanding the Climate Science of the Terrestrial Biosphere", Proceedings of the 13th Annual Community Climate System Model Workshop, Breckenridge, Colorado, June 2008.

3. **Joshua New**, Master's Thesis: "An Advanced User Interface for Pattern Recognition in Medical Imagery: Interactive Learning, Contextual Zooming, and Gesture Recognition", Archives of the JSU and MCIS Libraries, Jacksonville, AL, 2004.

4. **Joshua New**, Erion Hasanbelliu, Mario Aguilar, “Med-LIFE: A Diagnostic Aid for Medical Imagery”, Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences, Las Vegas, Nevada, 2004.

5. **Joshua New**, “Heterogeneous ARTMAPs for Image Segmentation”, Proceedings of the JSU Graduate Research Colloquium, Jacksonville, AL, 2004.

6. **Joshua New**, “A Method for Temporal Hand Gesture Recognition”, MCIS Technical Report, Jacksonville, AL, 2004.

7. Mario Aguilar, **Joshua New**, Erion Hasanbelliu, “Advances in the use of neurophysiologically-based fusion for visualization and pattern recognition of medical imagery”, Proceedings of the 6th International Conference on Information Fusion, Cairns, Australia, 2003.

8. **Joshua New**, Erion Hasanbelliu, Mario Aguilar, “Facilitating User Interaction with Complex Systems via Hand Gesture Recognition”, Proceedings of the 2003 Southeastern ACM Conference, Savannah, GA, 2003.

9. **Joshua New**, “A Method for Hand Gesture Recognition”, Proceedings of the ACM Mid-Southeast Chapter Fall Conference, Gatlinburg, TN, 2002.

10. Mario Aguilar, **Joshua New**, “Fusion of Multi-Modality Volumetric Medical Imagery”, Proceedings of the 5th International Conference on Information Fusion, Annapolis, MD, 2002.

11. **Joshua New**, Erion Hasanbelliu, “Med-LIFE: A System for Medical Imagery Exploration”, Proceedings of the JSU Graduate Research Colloquium, Jacksonville, AL, 2002.

Invited Presentations

1. “SeeGraph: A Visual Analytics System for Correlation Data” as invited presenter at the 2nd annual Gaggle Workshop at the Institute for Systems Biology in Seattle, WA, 2008.

2. “SeeShader: A System for General Purpose Computation on the GPU” as special edition of the UTK computer graphics course, 2007.

3. “SeeBrain: A System for Comparative Visualization of Brain Nerve Fiber Tracts”, invited presenter at the local chapter of the ACM, 2006.

4. “Fiber Renderer: A System for Visualizing Queries of DT-MRI tracts”, presented at Vanderbilt University’s Institute of Imaging Science as a source of potential research funding and collaboration, 2006.

5. “SeeGraph: A System for Visualizing Weighted-Edge Graphs”, research meeting at Oak Ridge National Laboratory, 2005.

6. “An Advanced User Interface for Pattern Recognition in Medical Imagery: Interactive Learning, Contextual Zooming, and Gesture Recognition”, JSU Studio Component Business Presentation, Jacksonville, AL, 2004.

7. “A Method for Temporal Hand Gesture Recognition”, JSU Graduate Research Symposium, Jacksonville, AL, 2004.

8. “Heterogeneous Collection of Learning Systems for Confident Pattern Recognition”, JSU Graduate Research Symposium, Jacksonville, AL, 2004.

9. “Advances in the use of neurophysiologically-based fusion for visualization and pattern recognition of medical imagery”, 6th International Conference on Information Fusion, Cairns, Australia, 2003.

10. “The Sword: A Role Playing Game for Demonstrating Computer Graphics Techniques”, JSU Graduate Research Symposium, Jacksonville, AL, 2003.

11. “Pliable Display Technology: Contextual Zoom as a Learning System Interface”, JSU Graduate Research Symposium, Jacksonville, AL, 2003.

12. “Facilitating User Interaction with Complex Systems via Hand Gesture Recognition”, 41st annual ACM Southeast Conference, 2003.

13. “Computer Vision: Gesture Recognition from Images”, ACM Mid-Southeast Chapter Fall Conference, 2002.

14. “Fusion of Multi-Modality Volumetric Medical Imagery”, IEEE 5th International Conference on Information Fusion, Annapolis, MD, 2002.

15. “Med-LIFE: A System for Medical Imagery Exploration”, JSU Graduate Research Symposium, 2002.

16. “Sonoluminescence”, Jacksonville State University, College of Arts and Sciences 3rd Annual Undergraduate Research Symposium, 1999.