BioN: a novel interface for biological network visualization

by

Lisa McGarthwaite

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Human Computer Interaction

Program of Study Committee:
Julie Dickerson, Co-Major Professor
Steven Herrnstadt, Co-Major Professor
Heike Hofmann

Iowa State University
Ames, Iowa
2008

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Information Visualization impacts every day life. As life continues to become more technologically enhanced, increasing amounts of data are being collected, stored, and analyzed. Technology assists researchers and scientists not only to make new discoveries, but also to create new ways to explore the information they collect. This paper contains a small preview of the vast field of Information Visualization. From the various fields of visualization, visualization history, and current findings, we investigate the field's impact. After studying the current technologies and tools for visualizing networks, we believe there is a more optimal solution than ones currently in use. We propose BioN, a new, novel touch-based interface for exploration and discovery of large, multivariate biological networks. The new program incorporates the ability to see the networked data in multi-windowed and multi-graphed representation. This ability will allow users to exploit the inherent strengths in the different graphs formats.

CHAPTER 1. OVERVIEW

"A picture is worth a thousand words"
Anonymous

1.1 Introduction

THE GOAL OF this paper is to present, in brief, the history and current state of Information Visualization, IV principles, techniques, and recommend a new interface for exploring large multivariate biological networks. For a field that has spanned centuries, IV is only now becoming utilized for not only research, but also everyday life. Being able to easily perceive information and disseminate it is a critical factor in our lives. From pill-bottle labels to DNA analysis, the design of information touches our lives in every way.

Among the many reasons why visualizations are needed are visualizations enable external cognition, creating tools outside the mind that can boost mental activities (Ware 2000), and it helps to show complex data in a way that is accessible for viewers. Seeing data encoded with multi-attributes helps with our short-term and/or working memory. Comparisons are also easier when a lot of data can be shown in the same space. The process to create and/or use information visualizations is relatively simple (see Figure 1). The four stages include collection and storage of the data, preprocessing to transform the data into an understandable state, the display hardware and graphics to produce the visualization, and finally, the human perceptual and cognitive system to make sense of what is seen. However, the implementation and factors that go into choosing what is seen is complex and difficult. The cycle also does not show the importance of context and intent of the visualization. Today the IV field is vast, with new tools continuously being created.

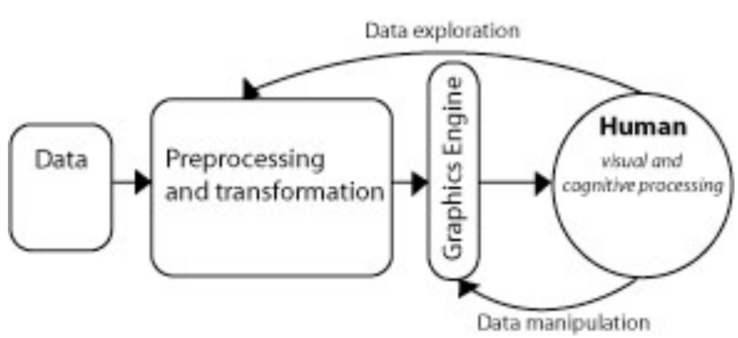

Figure 1. Diagram of Visualization Process. Adapted from Ware 2000.

After examining numerous topics in IV, we turn our attention to proposing a new User Interface (UI) for investigating and exploring large, complex biological networks. To understand what is available to-date, we conducted a competitive analysis of biological visualization tools that are designed for this task. However, the field of information visualization is still developing and much room for improvement exists. Typical network visualizations rely on tree or node-link based visualization of data. More recent work in social networks has yielded matrices for network visualization. We propose a hybrid visualization that uses a variety of graphing methods. To ascertain requirements and desired content/interactions for this new tool, we interviewed biologists in the field. Based on these conversations we devised a new hybrid visualization utilizing multi-touch tables, and named this new tool BioN (**Bio**logical **N**etwork). BioN will have the ability to recognize multi-person and multi-gestures to enable scientists to directly manipulate data, thus relieving the need for external hardware and reducing interaction time. Taking the current visualization tools functionality, we devised new gestures, visuals, and dynamic interactions.

1.1.1 Hypothesis

Touch-based visualizations utilizing a variety of visualization representations for biological networks will enable scientists to more easily explore and investigate biological network data.

CHAPTER 2. REVIEW OF LITERATURE

Nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under thy observation in life.
Marcus Aurelius Antoninus

2.1 Introduction

THE FOLLOWING LITERATURE review is organized into sections dealing with unique areas of Information Visualization (IV). Some of these areas are incredibly detailed and deserve an entire book in their own right. Rather than attempt the almost impossible feat of covering all angles of IV, this author hopes to highlight key topics and provide new insights into how topics might be viewed and/or arranged. While a part of a specific stage for visualization creation, the following paragraphs give a high-level overview of what is covered in each section. Before one can begin to create new graphics, he/she needs to understand what visualization fields exist and previous work that has been created. Next, while planning the visualization, one should be aware of the principles of human cognition and graphing. The domains of data and visualization techniques to show and interact with the data also need to be considered. To create the visualization the designer needs to be familiar with design conventions, available tools, and software. Finally, the visualization needs to be proven effective for the goals it set out to achieve. Below is a diagram showing the sequence of the literature to be covered, and the part of the design process it belongs to.

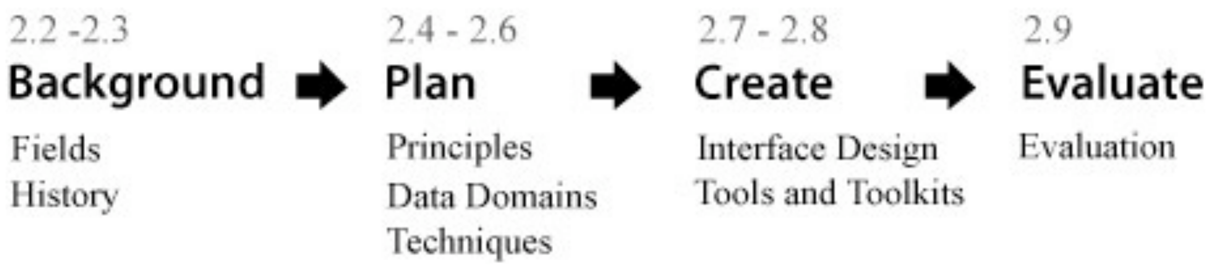

Figure 2. Flow diagram of literature review

FIELDS

While no standard number of visualization fields has been identified, this author feels there are five distinct types of visualization: Artistic, Knowledge, Data, Scientific, and

Information Visualization. While there are many ways that these fields overlap, there are distinct differences in their goals and designs.

HISTORY

Before one can think about the future, he/she needs to understand the importance of the past. The field of information visualization is older than most people would believe. From the beginning of human history, man has tried to show his thoughts and ideas in a visual manner. Information Visualization started as a small concept for mathematicians and scientists. It experienced a "golden era" as well as dark times of little innovation. Beginning in the 1800's, statisticians such as William Playfair have worked to show data in a standard scientific way. Modern-day gurus such as Edward Tufte and Ben Shneiderman continue to advance and explore the possibilities of visualization.

PRINCIPLES

When creating Information Visualizations (IV's), the designer must be aware of many aspects. Since humans are the end users, the creator of IV's must design for the human cognitive ideal. The human pre-attentive span is vast, yet our attentive state is very limited. We work with limitations, and visualizations need to be designed to compensate and extend our capabilities. Designers of visualizations also have to be aware of conventions used in the graphing community. Violating these principles can lead to confusion for the end viewer and unattractive charts.

DATA DOMAINS

Interest in a dataset is often based on a specific quality of the data. While this author has found limited work on the topic, we feel that Information Visualization tools usually have a dominant data characteristic, i.e. there is some aspect of the data that is the most crucial to show (such as a change over time or hierarchy). These characteristics include, but are not limited to, Categorical, Hierarchical, Network, Temporal, Spatial, and Textual. We aim to show general visualization methods based on these characteristics and give specific examples.

TECHNIQUES

Researchers are finding new ways to make interfacing and manipulating data easier for the user. Perhaps two of the most familiar types, thanks to Ben Shneiderman, are

Overview + Detail and Focus + Context. His mantra, "Overview first, zoom, details on demand", has influenced most major visualizations. Newer techniques include Dynamic Previews, Tours, Dimensions, and many more. Combining these techniques, developers are able to create unique visualizations and interactive graphics.

It has been proven that visualizations that are too cluttered are almost of no use. Static diagrams used to be the only way to read and/or explore data. With the advances of technology, users are now able to directly interact and change the data that they are presented with. Through the use of interaction, the creator is able to customize the display shown without sacrificing the complexity of the information. While the user used to be limited to point/click input, new advances allow the user to use touch, gesture, and even use explore using virtual reality to investigate data.

INTERFACE DESIGN

To create a useful visualization, the interface to the data must fit with the mental model of the user. Those interfaces that are successful in meeting the user's expectations in terms of usability, aesthetics, and function are the tools that excel in the real world and are the considered the most useful. The designer must also take into account presentation technology, target audience, typography, imagery, layout, and color.

TOOLS AND TOOLKITS

To create the myriad of visualizations available today, software developers have been coding toolkits to allow IV developers to easily produce and experiment with visualizations. While some tools are meant for the beginner IV creator, others allow for rich data design. In this section we will look at examples of some current popular software.

EVALUATION

Evaluations of Information Visualizations are still in the early stages. Few tools have been put under the microscope, so to speak, for any length of time. To be truly useful, visualizations will have to start being proven effective for the task they are designed for. Only recently have experts begun to create criteria for domain-specific tasks that visualizations must work for. Tasks taxonomies are being created and are the basis for judging visualizations.

2.2 Visualization Fields

Classification lies at the heart of every scientific field.
Lohse et. al 1994

THE USES FOR visualizations are diverse. While experts and designers alike agree on few names for the different domains and the content they include, this author believes that there are differences that have not been previously discussed. In addition, this author has found limited to no previous mention of one type of visual representation field: the Artistic. The sectioning off of these categories is a non-trivial task. Frequently, the content that each field uses has root in more than one domain and below are the main categories that this author believes exist today. This list is non-exhaustive; as such, this author does not believe these are the only categories or that each category is totally distinct from its relatives. Rather, there are discrete characteristics that these fields embody that separate themselves from the others.

2.2.1 Artistic Visualization

Since pre-historic times, man has had a need to represent what he has seen. While there never has been a consensus of what is art, most would agree that art tries to represent information in a personally meaningful way and point of view. From the early days of medieval art, to the subsequent move to realistic portrayal, art has continued to be influenced by the culture of the time. While there are too many historic movements to mention, the ideals of some, like visualizations today, continue to explore the notion of expressing some abstract quality of the world. Art movements such as Cubism and Surrealism explored radically different ways to view the world around us. Cubism broke down objects into an object's most distinct features. Other cubists played with the idea of expressing time in a 2D medium. For example, Duchamp's painting, "Nude Descending a Staircase #2", tried to visually show the passage of time of a person walking down a staircase. Surrealist artists, such as Salvador Dalí, played with the notion of perspective, representation, and the idea of how we see a "normal" world. One of example of Dalí's paintings is "Christ on the Cross," which shows two distinct perspectives: an aerial view from above the cross, and the other from the perspective of one looking into the distance.

Current art influences and is influenced by technology. Huge datasets and database contents are becoming widely available to the world. No longer are computer scientists or engineers alone creating visualizations. With cheaper hardware and user-friendly development kits, artists are able to create artistic works based on actual data. Called Visualization Art, Data Art, or creative information visualization, this movement uses underlying interaction and data visualization techniques to allow the artist to make a statement using current data sets (Viégas and Wattenburg 2007) and to allow the user to make a personal impression or interpretation of information (See Figure 1) (Lau 2007). While visualization art may use techniques from other visualization fields, it is not critical that the user is able to identify or make accurate inferences about the data. As such, visualization artists use a variety of novel techniques to represent their data and are not overly concerned with the best cognitive/perceptual approaches. The overall goal is to deal with aesthetics and emotional qualities (Vande Moere 2007). A current example is an installation piece called Sensity (Stanza 2004). This work collects data across an urban environment infrastructure through the use of a sensor network that collects and publishes data online. The output of the sensors is the emotional state of the city and is used to create installations and sculptural artifacts. Types of data collected include information on movement of people, air pollution, and vibrations and sounds of buildings. Visualization designers have much they can learn from the Artistic field, for the Arts have for hundreds of years experimented and developed techniques for ways people represent and perceive the world.

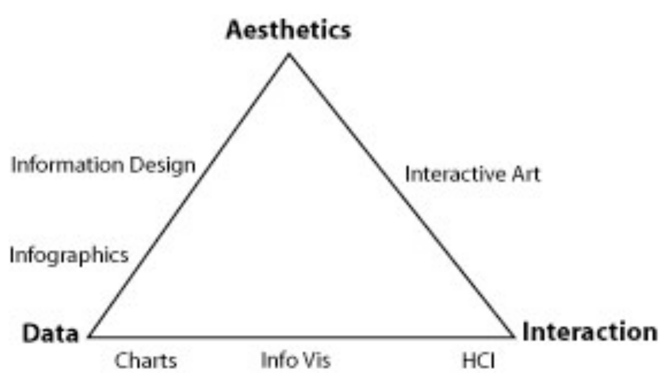

Figure 3. Data, Aesthetics, and Interaction. Adapted from Lau (2007)

2.2.2 Knowledge Visualization

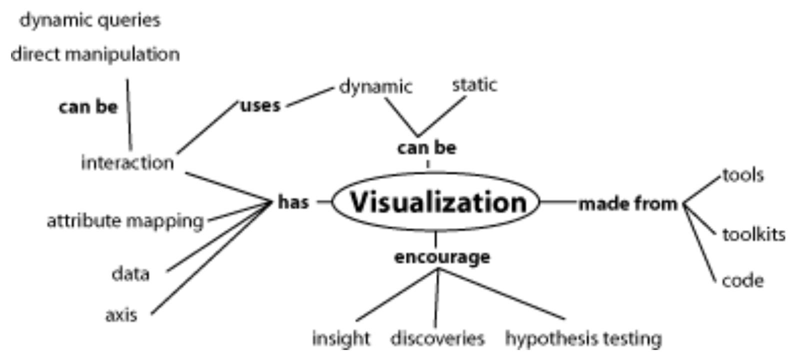

Figure 4. Mind Map of Visualization

Knowledge occurs when data is made meaningful to an individual. The only problem that that creates is that another person may not know what one individual considers "knowledge". Knowledge Visualization (KV) aims to improve the communication and remembrance of information that is learned. Spatial strategies help people store, retrieve, acquire, communicate and use resources and knowledge (Sigmar-Olaf and Keller 2005). If one is able to organize data in his/her own mental view, it correlates that he/she begins to understand the data. "Helping students to organize their knowledge is as important as the knowledge itself, since knowledge organization is likely to affect student's intellectual performance" (Sigmar-Olaf and Keller 2005). Visual representations are often processed more effectively than propositional ones. KV's are effectively used by experts to help guide and increase comprehension through new content. Some common visualization tools include mental mapping, freestyle maps, guide maps, and information maps.

While KV aims to foster new insights into experiences, perceptions, and attitudes, KV does impose some limitations. The very nature of how the knowledge is shown restricts the bounds of representation. For example, if a common node-link structure is followed, it forces the user to conform their mental view to this type of map. Most maps only allow static content (Sigmar-Olaf and Keller 2005), thus the user is not able to show dynamic transitions or effects. These kinds of maps present know-what or know-how knowledge and often leave out the know-where aspect. Knowledge is continuing to be distributive, and knowing where to find resources can be critical. Finally, KV faces some tough usability challenges. KV's are usually quick sketches using pencil and pen. Allowing for collaboration, mistake fixing, or backward tracking can be difficult.

While KV has its differences from other visualization domains, this author would argue that any work that is done in the realm of visualization should first begin with KV. Further, any effective visualization should lead to a KV, or try to incorporate KV within it. Once we see data represented, we automatically begin to construct our own mental model of how the new data correlates to what we already know. KV has uses in Education, Cognitive Psychology, and Human Computer Interaction (HCI).

Examples of visual formats include sketches, diagrams, images, objects, interactive visualizations, information visualization applications, imaginary visualizations, and stories. Beyond the mere transfer of facts, knowledge visualization aims to further transfer insights, experiences, attitudes, values, expectations, perspectives, opinions, and predictions. Knowledge Visualization integrates methods from a variety of fields, such as Visual Communication, Communication Sciences, Visual Perception and Knowledge Management.

2.2.3 Data Visualization

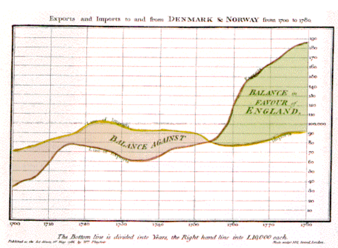

Figure 5. William Playfair's export and import chart (1785)

Developed for use in statistics (see 2.3 for the history), these types of visualizations aim to accurately present collected data. Often used to make comparisons, show gaps, or patterns, DV's have a long history of use in both the academic and commercial world. Like all visualizations, DV aims to show complex data in a digestible manner for viewers. While DV has strong ties to Information Visualization (IV), DV tries to show the raw data with all its inherent variability and uncertainty (Unwin et al. 2006). These graphics present findings from data, and are usually used to confirm or present findings to others. For the most part, DV's have remained a static presentation, along the lines of infographics and charts. An interactive example from this domain is Jonathon Harris's Word Count (Harris 2004). This interactive graphic shows the most commonly used English words ranked by frequency.

2.2.4 Scientific Visualization

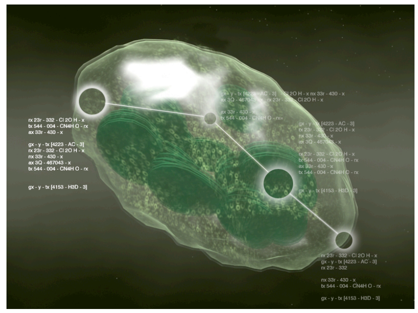

Figure 6. Organelle Visualization from MetNet at Iowa State University

Evolved in the late 1980's, Scientific Visualizations (SV) are a based on factual observations and phenomena from the real world in complete accuracy (Rhyne et al. 2003). The field aims to help users understand and explore data. SV is closely linked with Information Visualization; however, it has a more natural modeling structure (e.g. wind flows or anatomy). As such, the creators of SV's usually do not have a problem mapping their data to a spatial representation. This author would also argue that SV has more intention to educate users than to encourage new discoveries, although it certainly can be used for such tasks. Since SV's have a natural mapping structure and known data, SV creators are able to create simulations. A user can then see exactly what happens in, for example, a beating heart. The user is also able to test hypotheses by changing conditions around the visualization, but the underlying structure and functions remain the same. A great educational tool, simulations allow users to learn complicated or dangerous tasks in a risk-free environment.

2.2.5 Information Visualization

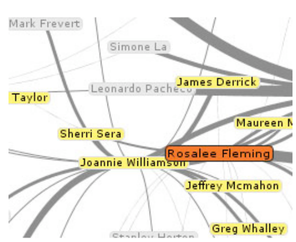

Figure 7. Network (Trampoline Systems 2006)

Unlike Scientific Visualization, Information Visualization (IV) tends to try to visualize abstract, multidimensional data (Shneiderman and Plaisant 2005). These data sets often do not have apparent, clear structures that can be modeled. Matured in the mid-1990's, this field continues to grow (Rhyne et al. 2003). While the term IV is used as an umbrella for all visualizations, it does have a specific purpose. To psychologists, IV is a representational mode used to show data in a visual-spatial manner. For those in the computer science domain, IV means the use of computer-supported, interactive, visual representation of abstract non-physical based data to increase cognition (Sigmar-Olaf and Keller 2005). Most of all, IV is used to discover information in data. Frequent tasks for IV are to discover patterns, trends, clusters, outliers, and gaps (Shneiderman and Plaisant 2005). IV's have structure and meaning embedded by the symbols, words, icons, shapes, and glyphs that are used to encode multivariate data (Sigmar-Olaf and Keller 2005).

IV designers and/or creators also face many challenges. IV's require well-prepared and well-structured data, which explains why networks are still hard to create for they usually are not well structured. Visualizations for large-scale datasets are still a struggle to represent due to limited computer screen size, resolution, and the limited working memory of users. While metaphors can help in the construction of visuals, it is very challenging to find a metaphor that fits the abstract data that IV's use. Since complex tools are needed to visualize these datasets, users are also faced with the technical challenge of learning new visualization systems.

2.2.6 Correlations Among Visualization Domains

While these visualization fields have distinct differences, in many cases the methods for the visualizations overlap or the fields grew out of each other. Figure 8 shows this author's mental model of how the fields correlate. The width of the line represents the strength of the connection between fields. For example, the artistic domain has always had strong roots to the Knowledge field, as art is a personal representation of some idea or feeling. Knowledge Visualization helps us map out our ideas of data, which is what Information, Data, and Scientific Visualizations try to create. Information Visualization often uses plots and charts from DV as part of its representations. While all these fields have differences, in one respect they are related. All try to accomplish the same basic function by visualizing information, data, or ideas.

Figure 8. Author's Mental Model of Domain Ties

2.3 Visualization History

If you want to understand today, you have to search yesterday.
Pearl Buck

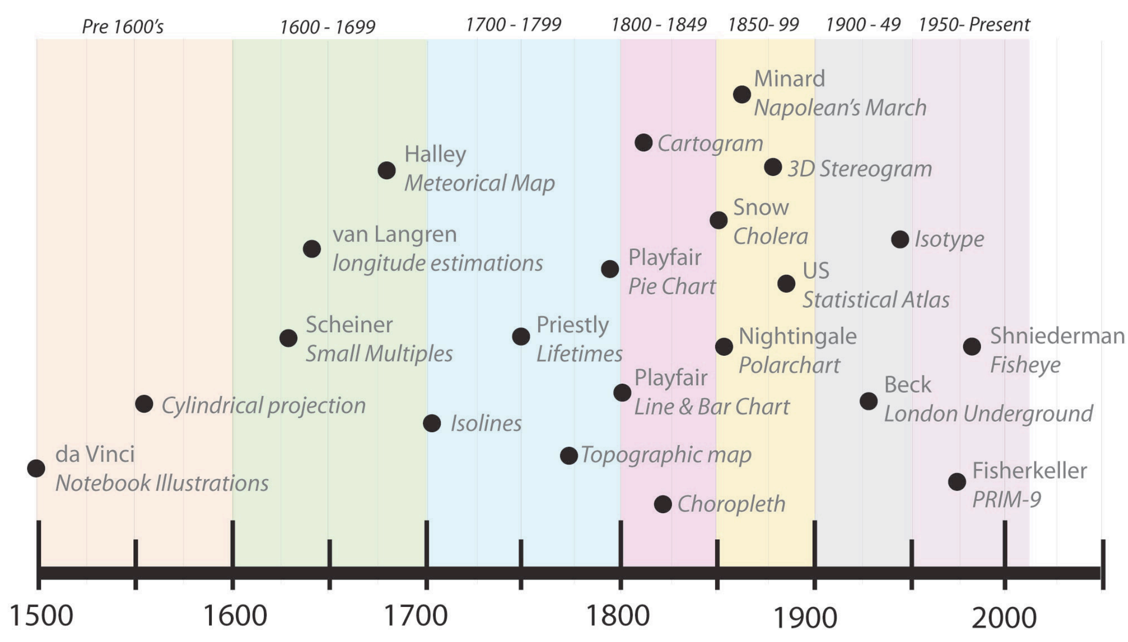

Figure 9. Timeline for Data Visualization History

VISUALIZATION HAS A long and varied history. While man has long tried to depict information, it was not until the 1600's that modern methods of graphing data began. Many of the most common forms that we are familiar with were developed during the 1800's. Today, many novel techniques for visualizing and interacting with information continue to be created. For a snapshot of the history see Figure 9. Each of the following time periods described correlate to a block of time, indicated by color.

2.3.1 Pre-1600 – 1600's

The earliest known visualizations dealt with simple geometric diagrams, from positions of the stars to simple maps. One of the earliest known examples of visualization is from the 10th century. The diagram shows the changing position of seven of the heavenly bodies over space and time (Friendly 2006). Other works include the town layout found in Babylon in 6200 b.c. Visualization continued in the 14th century with the plotting of theoretical functions, and relationships between tabulating values and plotting them. By the 16th century, scientists were using the newly developed triangulation method to make mapping more accurate, which resulted later in the first modern cartographic atlas by Abraham Ortelius in 1570. Technology, during this time, created the camera obscura (an instrument that allowed the user to capture an image, most notably used in paintings).

The 16th century continued to advance the technology available to visualizations. Pressing issues during this time were concerned with physical measurement used for astronomy, maps, navigation, and territorial marking. For instance, Descartes and Fermat developed the analytic geometry and coordinate systems. The theories of error of measurement, estimation, and probability were developed. Statistics for demographics began to arise.

Notable visualization designers began to be recognized. Christopher Scheiner (1630) introduced a new idea that later data visualization expert Edward Tufte would name "small multiples." These multiple images were used to show the locations of sunspots for a 3-month period. Michael Florent van Langren created what might be the first statistical graphic in 1644. He used a horizontal line to place the 12 known estimates of the difference in longitude between Toledo and Rome (see Figure 10). He chose to represent this data graphically, rather than in tables. If he had not done so, the large gaps between the estimates would not have been so easily identifiable.

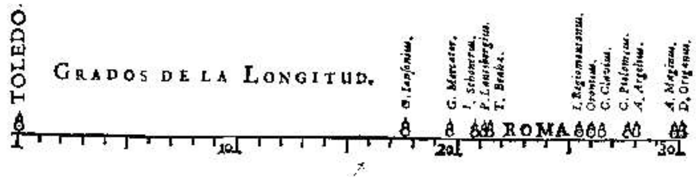

Figure 10. van Langren Longitude Estimations (1644)

2.3.2 1700's

With the beginnings of statistics and interest in data, graphic representation began to expand. Maps began to show more than just locations. Isolines and contours were invented, and thematic mapping of actual physical properties began. Edmund Halley (1701) created isolines to show contours on coordinate maps. Introduced by Phillippe Buache and Marcellin du Carla-Boniface, contour and topographic maps were used.

Another notable development during this time was the creation of timelines, begun by Jacque Barbeu-Dubourg. A famous example of this type of representation is from Joseph Priestly in which he showed a timeline of biographies of 2,000 famous people (See Figure 11).

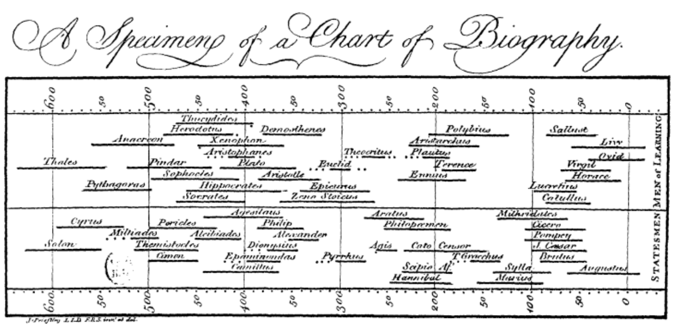

Figure 11. John Priestly Biography Timeline (1765)

One of the period's most famous names is William Playfair. He is attributed the creation of most of the graphical forms we still use today: the line graph, bar chart, circle chart, and the pie chart. He used these techniques to show the British taxes, price of wheat, wages, and reigns of monarchs. These techniques were so new at the time that Playfair had to devote many pages to the explanation of how to use these graphics.

2.3.3 Early – Mid 1800's

The beginning of the 1800's saw the explosion of graphics and mapping. Most of the modern statistical forms were finalized: bar, pie, histograms, line graphs, time-series, scatter plots, etc. Cartography advanced from single maps to complex atlases on a variety of topics.

William Smith ushered in a pattern of using cartography to show quantitative data. Baron Charles Dupin in the 1820's developed the use of continuous shading to show the literacy distribution and degree in France, which is probably the first unclassified choropleth map. Just a few years later, in 1825, the Ministry of Justice in France created the first national system of crime reporting. André Guerry, a lawyer, used these mapping techniques to compare ranking of departments on pairs of variables, such as crime versus literacy.

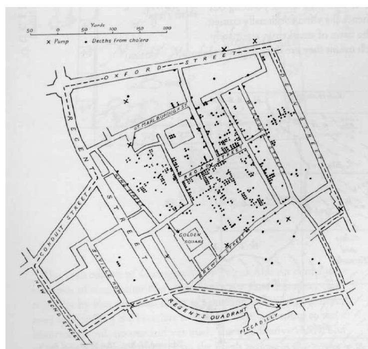

Figure 12. Dr. John Snow Cholera Map (1855)

It was during this time that cholera first appeared in Great Britain, killing over 52,000 people, in an epidemic that lasted for over 18 months. Cholera epidemics continued over the next few years with similar death rates. Dr. John Snow in 1855 created his famous dot map (See Figure 12) that marked the locations of deaths due to cholera. This map showed that the deaths were clustered around a single water pump. What was so remarkable about this map was that Snow showed the number of deaths at precise locations. Dr. Robert Baker, a physician at the time, also tried to show the cholera deaths. However, his map showed the districts affected by the disease, but it did not pinpoint locations.

Another major person in the field was Charles Minard. Like Playfair, he was renowned for his graphical displays. In 1844, his "tableau-graphique" showing the transportation of goods was the precursor of the modern mosaic plot.

2.3.4 Late 1800's: The Golden Age

The rapid growth of visualizations had been established by the 1850's. Statistical charts were used in official state offices throughout Europe. They were used for social planning, industrialization, commerce, and transportation. So diverse are the developments in this time that covering them all is not feasible. However, a few themes stand out. Maps began to leave the 2D world behind and explore 3 and higher dimensional spaces. Gustav Zeuner, from Germany, and Luigi Perozzo, from Italy, constructed 3D surface plots of population data. Contour diagrams, while developed earlier, expanded in the applications to which they were applied. Edwin Abbott's *Flatland* even suggested that possible views in four and more dimensions might be possible.

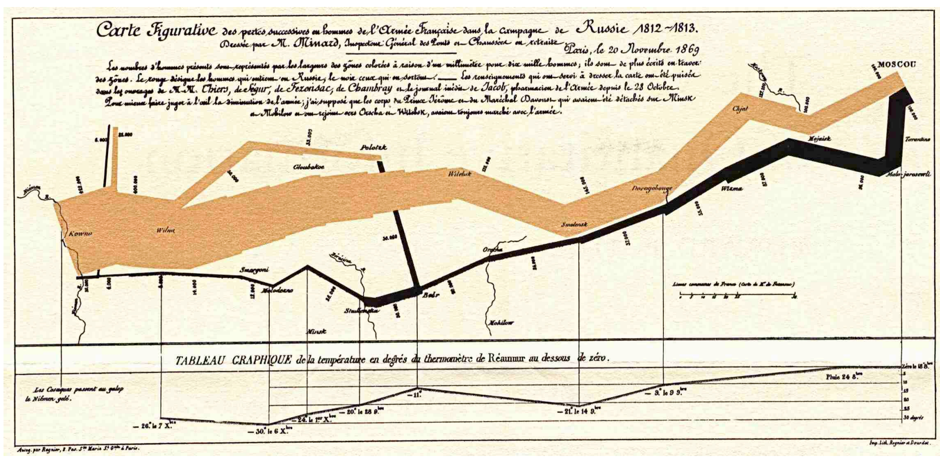

Figure 13. Charles Minard Napoleon's March (1869)

Secondly, graphical innovations continued being produced, notably the flow diagram, divided circle diagrams on maps, polar charts, scales, and shapes on maps. Charles Minard created a graphic during this time that is still regarded as one of the best in the history of visualizations. His flow map of the March of Napoleon (see Figure 13) showed the failed attempt of Napoleon's March to Moscow. The time, temperature, number of men, and other variables are recorded for the entire campaign. Also during this time, Florence Nightingale created a polar area chart to show the causes of death during the Crimea War. Her work lead to sanitation changes for treatment of wounded soldiers in the battlefield.

The contributions by Francis Galton (1822-1911) cannot be left out. He is well remembered for his work in correlation and regression. However, less known is part that visualizations and graphing played in his discoveries. His insight lead to the discovery that isolines of equal frequency would appear as concentric ellipses, and that the locus of the lines of means $y \mid x$ and of $x \mid y$ were the conjugate diameters of these ellipses. These discoveries were the result of visual analysis from applying smoothing to his data. Perhaps his most notable discovery was that counter-clockwise patterns of winds around low-pressure zones, combined with clockwise rotations around high-pressure zones.

The collection of political and governmental data was widespread during this time, and reports using graphics were published regularly. With all the new forms of graphing, a need arose for the standardization for graphical presentation. The International Statistical Congress recommended that maps and diagrams accompany official publications. State-sponsored statistical atlases ensured that a Golden Age of Graphics ensued. These detailed atlases became time capsules of popular methods, often representing the best work of the period.

2.3.5 1900-1950's

The innovations of the previous time could not be kept up forever. The next 50-year period was to see few innovations in the graphical community. With declining enthusiasm for "pictures," the call for quantification and formal statistical methods became the norm in social sciences. However, graphical representations did not lie dormant. Graphics made the transition to mainstream culture, entering English textbooks, school curriculum, and standard use in government. The use of graphics in other fields lead to significant insights in biology, physics, and other sciences. Created by H. Beck, the world-famous graphic of London Underground subway system during this time period (see Figure 14). The world of graphical representation was awaiting new technologies and ideas. Upcoming computational power and modern statistical methodology would spur the field on to new innovations.

Figure14. H. Beck's Map of the London Underground (1933)

2.3.6 1950's – Today

The dormancy statistical representation faced by the graphing community began to lift during the mid 1960's due to three significant developments: John W. Tukey called for the recognition of data analysis as a branch distinct from mathematical statistics. He also began a wide variety of new, simple, effective displays (stem-leat plots, boxplots, two-way table displays). Next, Jacque Bertin published a paper that would help organize the visual and perceptual elements of graphics according to features and relations in data. Finally in 1957 the programming language FORTRAN allowed statisticians the computation power necessary to move beyond the hand-drawn maps and graphics. In addition, new themes emerged such as multivariate data, Fourier function plots, Chernoff faces, star plots, clustering, and trees.

Perhaps one of the most revolutionary developments was the PRIM-9 developed in 1974 by J. Tukey, J.H Friedman, and M. Fisherkeller (see Figure 15). PRIM-9 stands for Picturing, Rotation, Isolation, and Masking in 9 dimensions. Created by statisticians and computer scientists, this tool was the first multidimensional, dynamic, and interactive visualization system in the world. Most of the techniques that were used in this system were revolutionary and are still the basis for high dimensional data display today (Friedman and Stuetzle 2002).

Figure 15. PRIM-9 (1974)

The last quarter of the 20th century visualization blossomed into a mature and multi-disciplinary research area. New software tools were developed for a wide range of visualization methods and data types. Describing all the new developments is beyond the scope of this paper, but a few that stand out are:

- High-dimensional, interactive and dynamic computing systems
- New types of direct manipulation
- Increased attention to the cognitive and perceptual aspects of data and visualization

The 1980's and '90's gave rise to the desktop computer, which allowed software for dynamic graphics to become more available. New general systems for dynamic, interactive graphics with data manipulation and analysis were created. Today, scientists and designers have a wide array of tools to create visualizations.

2.4 Visualization Principles

Failure comes only when we forget our ideals and objectives and principles.
Nehru

IN ORDER TO create effective visualizations, one must know the limitations and abilities of one's users and the basic building blocks of how the human mind and memory functions. Research has pointed out that the human perceptual system is limited, not only in how quickly we can recognize something, but also how long we can remember variables. In addition, standards for graphics have been in use for a long time. These principles can help determine scales, colors, and methods to plot data. This section is divided into the two main areas of cognitive principles and graphing principles.

2.4.1 Cognitive

The first consideration, when beginning to create graphics, is to understand the human cognitive abilities, the study of which is called *cognetics*. Cognitive load is intrinsic, extrinsic, and germane. Intrinsic is caused by the task itself. The extrinsic load is caused by type of material, representation, and interaction necessary for learning the material. Germane load is the amount of conscious cognition. It touches on processes that are directly relevant for learning (Sigmar-Olaf and Keller 2005). These areas touch on our perception, attention, and memory abilities. We have a conscious and unconscious mind. The unconscious mind controls processes of which you are not aware of at the time, i.e. one is not paying attention to, not thinking of (Raskin 2000). An unconscious event can trigger an event to move to the conscious mind, e.g. an empty stomach lets you know when to eat. The distinction between these two states is not always clear-cut. Have you ever wanted to bring something to your mind and not been able to totally recall it? This "tip-of-the-tongue" phenomenon is an illustrative case where the unconscious mind and conscious mind blend (Raskin 2000). However, there are abilities we know belong to the unconscious (perception) and to the conscious (attention). Raskin provides a table of properties that display the differences between the unconscious and conscious mind (see Table 1).

Table 1. Properties of the Unconscious and Conscious Mind (Raskin 2000)

Property	Conscious	Unconscious
Engaged by	Novelty, emergencies, danger	Repetition, expected events, safety
Used in	New circumstances	Routine situations
Can handle	Decisions	Non-branching tasks
Accepts	Logical Propositions	Logic or inconsistencies
Operates	Sequentially	Simultaneously
Controls	Volition	Habits
Capacity	Tiny	Huge
Persists for	Tenths of seconds	Decades (lifelong)

2.4.1.1 Perception

What we actually see is not necessarily the world that is. Buddhists and Ancient Greek philosophers have known this fact for a long time. Hence, ancient philosophers favored theories that can be proven solely by reason and gave examples of why this method is true, such as Plato's Allegory of the Cave. J. Raskin (2000) argued that if cameras were limited to what our eyes can actually see, photography would never have been invented. To understand the human perceptual abilities, one must first examine the one organ available for our vision: the eye.

While the structure of the eye is well documented, it is still less certain how the eye actually function. The eye, while complex, is restricted in many ways. For vision, humans are dependent on focusing light on the retina, a light sensitive, smooth, curved thin layer of nerve cells at the back of the eye. The retina has two main areas, the fovea and optic disc. The fovea is a dip in the retina and opposite the lens. While it is the size of a thumbnail held at arm's length, it is also a region of high acuity. This acuity is what enables us to focus and read. When light enters the pupil, it is focused on the retina by the lens. The light then hits two types of nerve cells in the retina, rods and cones. Rods are highly sensitive and detect sudden flashes and movements. Cones enable the vision of color. The rods and cones react to light, and these reactions are signaled to the brain via the optic nerve and indicate brightness, color, and contour.

During vision, the eye continually makes sweeps across a scene. These moves, called *saccades*, take approximately 200 milliseconds to initiate. When the eye stops to investigate an object or part of a scene, it is called a *fixation*, usually lasting around 250-500 milliseconds long (Chen 2004, Healey 2007). Our perception is continually restructuring the sensory input we receive. There are two stages of attention, pre-attentive and attentive. The pre-attentive state has unlimited capacity and uses low-level vision system. In this stage, four main types of pre-attentive tasks have been found: target detection, boundary detection, region tracking, and counting/estimation. Target detection discovers the absence or presence of a target in a scene. An example would be to find a circle amongst a group of squares. Two features in this type of task are color and shape. Boundary detection determines where one region ends and another begins. Region tracking groups elements with unique features that are moving in time or space. Finally counting/estimation helps us to determine how many of selected features exist.

Unique features help us distinguish an object at a glance. Features agreed upon by the scientific community as being pre-attentive include: line, length, width, size, curvature, number, terminators, intersection, closure, color, intensity, flicker, direction of motion, binocular luster, stereoscopic depth, 3D depth cues, and lighting direction (Chen 2004, Deller et al. 2007, Healey 2007, Raskin 2000). Recent studies suggest that several features of an object are not processed separately but affect each other (Raskin 2000). A hierarchy for these features exits, e.g. color is more pre-attentive than shape, and some types are favored over others in certain tasks (Mckinlay 1986) (see Figure 16).

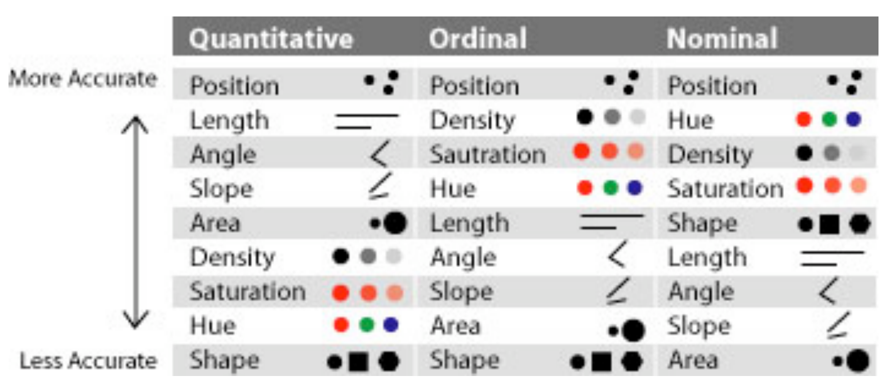

Figure 16. Visual Encoding Accuracy by Task type.

For example, color is easier to detect than shape in boundary detection. Luminance-on-hue preference has been observed (Healey 2007). A combination of pre-attentive features is usually not pre-attentively detectable (see Figure 17). The interference of features with one another is asymmetric, e.g. random color interferes with shape detection and hue-on texture.

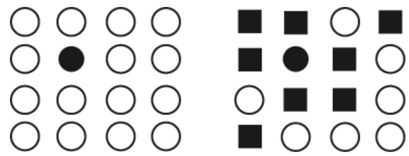

Figure 17. Color is pre-attentive, but color and shape is not.

Several working theories to date that try to explain how we our perceptual system works are feature integration, texton, similarity, guided search, and feature hierarchy. Feature integration proposes that if the target has a unique feature, then there is a given access feature map to detect if any activity is occurring. Texton theory states that an early visual system detects a group of features, called textons, which are usually elongated blobs, terminators, or crossing of line segments. Similarity theory supposes our search ability varies continuously, depending on type of task and the display condition and the types can be seen below.

- The visual field is segmented into structural units, which share some common property.
- There exists a limited resource that is allocated among structural units. Each unit is compared to the structural model.
- Units are grouped hierarchically, and a poor match between template and unit = rejection of other units grouped strongly with this.

Guided search believes that there is an activation map made based on bottom-up (feature categorization) and top-down (user-driven attempt) during visual search. There is one map created for each feature. Attention is drawn to peaks in the activation. Weight is task dependant between top-down and bottom-up (Healey 2007). Another common way psychologists believe we group features are by Gestalt principles of contrast, closure, repetition, alignment, unity, and proximity (see Figure 18).

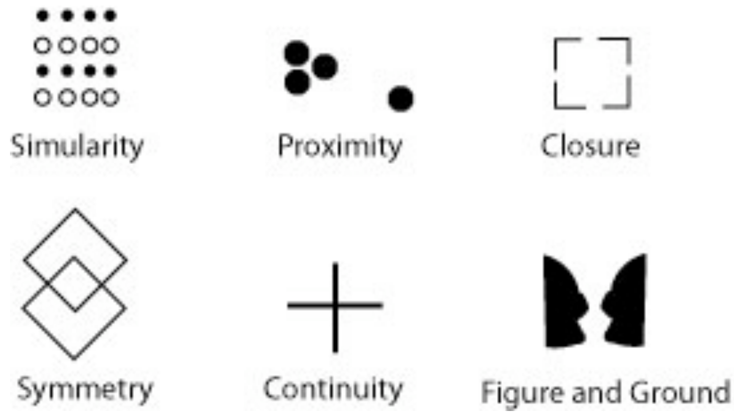

Figure 18. Examples of Gestalt

By knowing these theories and features, certain tasks have been found easier and harder for human perception**.** Cleveland and McGill (1984) ranked 10 elementary principles for tasks in graphs. These are, from most to least accurate:

- Position along a common scale
- Positions along nonaligned scales
- Length, direction, angle
- Area
- Volume, curvature
- Shading, color saturation

Whenever possible, the designer of graphics should enable investigation of the graph in the most simple means possible. Using not only pre-attentive features, but also incorporating them into higher up elementary tasks will yield visualizations that are easy to understand and navigate (Chen 2004). This encoding can be dependant on the type of task the user is doing. As such, preattentive features should be carefully incorporated.

Since our perceptual system is limited, it is often easy to confuse the viewer and make one perceive things that are not really there. Such has been the success of magic tricks and slight of hand. This area of perception is optical illusions. Our visual system is adapted to standard situations, and artificial manipulations can cause wrong interpretations of the visual scene. One illusion is caused by change blindness. Our visual system is not a camera; what we see is an ongoing dynamic construction project. Change blindness occurs when very large objects in a visual scene are not noticed, usually caused by brief disruption between images (Chen 2004, Healey 2007). They are most likely if changes are arranged to occur simultaneously with some kind of irrelevant, brief disruption in visual continuity like eye saccade, shifts of the picture, flicker, eye-blink, or a film cut in a motion sequence. The blindness we see is not a failure of our visual acuity, but rather due to a lack of or inappropriate attentional guidance (Healey 2007). Other illusions occur due to adaptation. The nerves in our eye fatigue after responding to the same stimulus for several seconds. Try staring at a bright red rectangle for a few seconds. If you then look at the complementary color (green) a white rectangle should appear briefly. This fatigue also sets us up for a motion-after effect. The motion perceived depends on contrast within the image (see Figure 19).

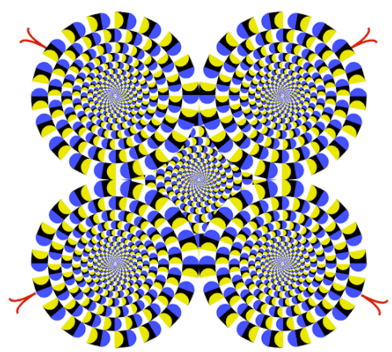

Figure 19. Rotating Snakes Illusion (Healey 2007)

Finally, the power of context in an environment plays a part. Angles-in configuration appears to be closer than an angles-out. Angles-in appears when we are near the front of an object, such as a ticket counter, and angle-out occur at the far end of a room (See Figure 20). Other features that can cause optical illusions include texture and non-photorealism. The implication for visualization is that the designer must be aware of object features and guide the user's eye and mind (Healey 2007).

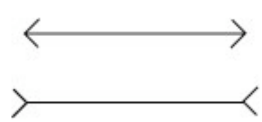

Figure 20. Angles. Which line is longer? They are both the same

2.4.1.2 Attention

Once the eye has focused and made a selection, the second stage of perception occurs: attention. Whenever the user encounters something novel, non-routing, or threatening, the situation is brought to the conscious mind (Raskin 2000). That which becomes the focus is called the Locust of Attention (LoA) or Locust of Control. The LoA identifies the source of what changes our attention. We are able to see and hear more than what the LoA is, but we cannot completely controls what becomes the LoA. The LoA operates sequentially, and one can only consider one question or control one action at a time (Raskin 2000). The LoA can be external, internal, or mixed (Cheng et al. 2007). An external source will be a cue in the environment that interrupts ongoing thought, such as a loud noise. An internal cue will be created within the current cognitive system goals. External and internal are best placed at opposite ends of the spectrum, with complex tasks mixing the two. Three distinct networks within the brain have been identified with processes called alerting, orienting, and executive control (Cheng et al. 2007). In addition, orientation occurs in the stages: disengage, moving, and engaging with the new focus. Executive control is the processes that are required to solve conflicts, correct errors, and plan ahead. These states describe the process of how our LoA changes.

There exists various dimension of attention, and these are done through priming and manipulation of cues. These cues can be spatial, semantic, and/or timing. Spatial information may or may not be provided in a cue. “Flagging” a stimulus does not necessarily signal where it appears. A flag could be a color change in a background or an object form. These cues can be direct, indirect, or mixed. Direct cues explicitly show the location. Indirect cues point to areas but not an exact local. Some cues will inform us when a target will change, others where it will change. Secondly, semantic cue information involves linguistic, iconic (graphic), deictic (physical pointing), and no information. These cues can help in the processing of the target. Finally, timing is shown in relation to the occurrence of the stimulus. This cue may be anticipated, concurrent, or retrospective (which means it could occur before, during, or after the beginning of the target). For tasks that take a large cognitive load, reactive cues may be more useful. Priori cuing is useful since it allows the target to be processed more effectively and decreases response times.

2.4.1.3 Memory

Once we have focused our attention from our perceptions, what we perceive can be stored in memory. However, perceptions do not automatically become memories, and we cannot assume a user will remember what they read just 5 seconds ago (Raskin 2000). There is not a single working memory supporting cognition; rather there are several limited-capacity systems (Sigmar-Olaf and Keller 2005). The most commonly known are short-term and long-term memory. However we also have visual, spatial, associative, auditory, and sensory memory. In this section, we will take a closer look into short-term, long-term, and visual memory systems.

The first type of memory is short-term. While short-term memory is often linked to working memory, the two are distinct. Short-term memory is described as the ability to hold a small amount of information in a highly available, active state. This type of information can be recently processed sensory input, items recently retrieved from long-term memory, or the result of recent mental processing. Working memory, on the other hand, is the theoretical framework that refers to structures and processes used to temporarily store and manipulate data. A better term might be working attention. Many place the average duration of short-term memory as short as 30 seconds, or as long as one minute. In order for information to be used past short-term, it must be periodically repeated or rehearsed. This repetition can be done out loud or by thinking about the information. Consolidating information to long-term is enhanced if any relationship exists between the item in short-term memory and long-term memory. George Miller in 1956 argued that short-term memory had the approximate span of seven items plus or minus two. Recent studies show this number is roughly correct, but memory spans vary widely with population and test material. Span can also depend on the number of characteristics of tested words. Other known effects are:

- Word-length effect: fewer words of longer duration words can be recalled
- Phonological similarity effect: more words can be recalled when they are very similar or occur frequently in the language
- Single semantic category: more words can be recalled when they are taken from a single semantic category versus different categories

Chunking can also increase the ability to recall. While we may only remember +-7 items, putting items in a unit can increase memory. For example, a telephone number is grouped by area code (3), region (3), and number (4) for a total of 10 items. Putting each unit in a meaningful word or phrase can also improve recall.

Long-term memory is the ability to bring back information from storage. This type of information is stored as meaning and can last from 30 seconds to decades. Long-term differs in structure and function from short-term memory. Long-term actually leads to a physical change in the structure of neurons. Information once in long-term memory does not necessarily stay there. Any information is subject to decay or being forgotten. It can take several recalls/retrievals from memory for the information to be maintained for years. The information is also dependant on the depth of processing it receives. Certain sensory information is stored with memory. For example, color has been proven to be stored and is another clue for memory retrieval (Sigmar-Olaf and Keller 2005). Two different types of long-term memory that exist are declarative and procedural. Declarative refers to all memories that are consciously available. Two types of these memories are episodic (specific events in time), and semantic (knowledge about the external world). Procedural memory deals with the process of moving the body or using objects, i.e. riding a bike.

Finally, the third type of memory to be discussed is visual. Visual working memory stores information that we see from one fixation to the next. It allows us to remember configuration, location, and orientation of visual material since we do not keep a visual model of the world in our mind. We are sensitive to detail in the center of our visual field, missing 99% of what is in our visual field (Sigmar-Olaf and Keller 2005). Visual memory uses different cognitive processes than those used in other memory systems (Chen 2004), but this system is limited to a small number of simple visual objects and patterns, usually holding 3-5 of them from one second to the next (Sigmar-Olaf and Keller 2005, Plumlee and Ware 2006). As new objects are seen and added to visual memory, others are dropped. One object stored in visual memory can have several attributes: not just 3 colors, but color/shape/pattern can be stored as a single entity as long as they are bound to the same object. Attributes are simple, and it is not possible to increase information capacity if one object has less characteristics. In addition, one complex object may take up the entire visual memory (Plumlee and Ware 2006). Visual features are temporarily grouped with links to verbal-propositional information. Deeper semantic coding is needed for items to be processed into long-term memory. Semantic meaning of a scene can be activated in memory in 100 milliseconds (Sigmar-Olaf and Keller 2005).

2.4.2 Graphics

While cognitive principles are crucial to facilitate understanding, the principles for creating and/or reading graphics are just as important. Even if a graphic is perceptually easy to view, the end user still may not be able to make sense of the data. Although some graphing methods have been a matter of convention, others principles are still being discovered.

Perhaps one of the best known in the field for making usable graphics is Edward Tufte. With his revolutionary book in 1982 and the subsequent 2nd edition, *The Visual Display of Quantitative Information*, he ushered in a new wave of thinking about displaying data (Tufte 2001). One of his most popular beliefs is the lie-factor in graphics. Data should be displayed truthfully, without manipulating aspect ratios or other features to make the data say something it really does not, i.e. the size of effect shown in the graphic divided by the size of effect in the data should be close to 1. Another key point is using only enough ink to draw the graphic known as the data-ink ratio. Too much ink is redundant and wasteful and can lead to chart junk (useless graphics or ornamentation). While these decorations can be inviting to viewers, they do not add anything to the data. Many designers add these, but the additions make it appear that the data itself is not important or exciting. Finally, data displays should aim to show as much information as possible. Only with information available at various levels of discreteness can discoveries occur.

Besides Tufte, another notable name is Cleveland. His principles aim to create easier-to-use and clearer graphics. Like Tufte, he wants the data to be the main focus on a display. This feat is accomplished by being aware of contrast between the data points and the background. Correct aspect ratios are critical, as well as providing data at a good resolution. Guidelines should be used and emphasized. Without these marks, we can become lost within the data. Grid lines offer landmarks as well as easy access to the values the marks are plotted on. The terminology on charts should be easily understood. In addition, extensive captions should be used on data that is complex or is novel to the viewer. The viewer should be able to understand and gain a clear vision of the main idea of the chart. Following Tufte's and Cleveland's advise, one can see the before and after effect in Figure 21.

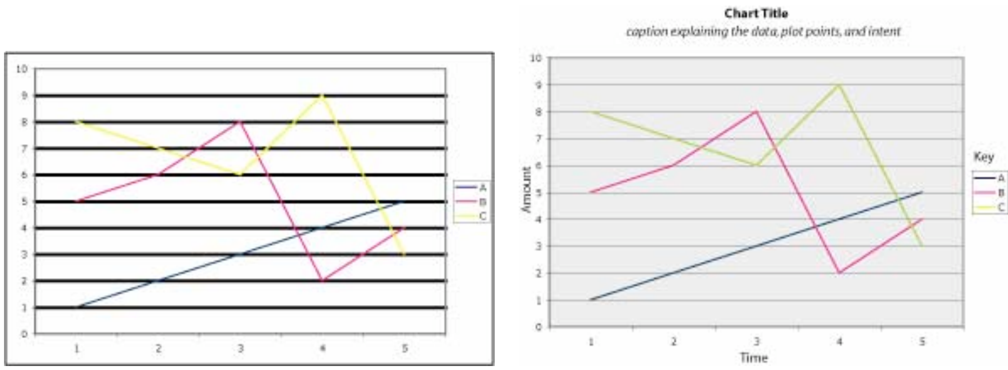

Figure 21. Before and after applying Tufte's and Cleveland's Principles

To summarize the work concepts, below is a list of things to keep in mind when creating a graphic **(**Tufte 2001**,** Emerson 2008):

- What is most important?
- Is the graph clearly labeled (title, key, axis)?
- Are the grid lines de-emphasized?
- How much ink is used for the data versus the rest of the graphic? Show as little non-data elements as you can.
- Is the data accurately plotted? What is the aspect ratio?
- What is the medium, printed or digital?
- Is the language easy to understand?
- Minimize the cognitive load for the user; e.g. if you are trying to show differences, plot the difference rather than the original data points.
- People look at the graphic before the text, if they read the text at all. Put the conclusion in your caption, clearly stated.
- Put as much information as you can into the graphic.

2.5 Data Domains

The goal is to transform data into information, and information into insight.
Carly Fiorina

THERE ARE MANY different forms of Information Visualizations. To-date, this author has come across limited articles that have tried to define general data categories that IV's try to represent**.** Shneiderman in 1996 presented his Type by Task Taxonomy (Shneiderman and Plaisant 2005). His type taxonomy contains the following:

1D: Textual docs, lists of names, all which can be organized in sequential order.

2D: Map data: each item covers some part of a total area.

3D: World data, structure modeling, or results represented by volumes and surfaces

Multidimensional: n attributes become points in an n dimensional space.

Temporal: Items/Events with a start and finish time

Tree: Inheritance and relationships

Network: Relationships

His list of the types of data, this author feels, needs to be updated and rearranged. His use of the term "dimension" is not the most appropriate. 1D types can be more than just textual. Any object that is encoded with a characteristic, such as color or shape, becomes one-dimensional. Rather, dimension should be thought not as the space an object occupies, but the number of characteristics encoded within the object. Secondly, 2D type is constrained to map data, or data that covers area, and 3D is defined as world data or volume. Both 2D and 3D then are concerned with not just geographic information, but Space. Space focuses on more than just physical attributes. For example, ambient visualizations try to show characteristics of a space within the actual physical location. We have reworked his original taxonomy (see Figure 22). In the following, we define basic categories or domains we believe most Information Visualizations (IV) fill and give examples of typical visualizations. However, we do not make the claim that each IV only fits one category. Many of today's IV's use multi-dimensional data to represent data sets. Rather, in this separation, we try to pull out main characteristics that IV's are based on.

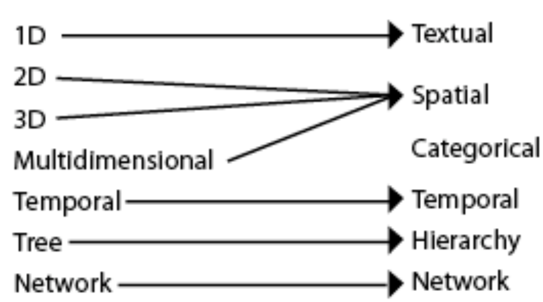

Figure 22. Data domains classification. Shneiderman (Left) and this author (Right)

2.5.1 Hierarchical

While Shneiderman names this category Tree, a better term is Hierarchical. Again, his term is limiting. Hierarchies aim to show inheritance and relationships among entities, but more than just tree structures achieve this goal. Entities can be directly or indirectly linked. Direct links are to an entities' parent or child. Indirect links extend up or down a set of links. For example, two coworkers who are not the other's boss, but the chain of command for both would meet somewhere further up the hierarchy. Perhaps one of the most pervasive visualizations of hierarchies is the folder structure on most operating systems. Folders and files can be placed in other folders. Lists, column, and icons then display the contents that are located on a specific level of the hierarchy. Another common visualization is a Table Lens (see Figure 23). Here the tree structure is broken down into recursive boxes. Other tools include Cone trees (Robertson et al. 1991), TreeMap, and MoireTrees.

Figure 23. Examples of Hierarchies: Tree (Nakamura 2004) and Tree Map (Shneiderman 2006)

2.5.2 Categorical

Perhaps one of the hardest data to visualize concerns categorical information. Based on groupings of data, this type of visualization tries to show how data is segmented by qualitative means rather than quantitative measures. Visualizations for the most part have been limited to Venn diagrams, which use overlapping circles to show where categories intersect. Newer tools include mosaics and category maps. Mosaics are able to show multiple categories in one plot (Unwin et al. 2006). Categories within categories are also possible. The middle graphic of Figure 24 shows a mosaic of casualties from the Titanic. Vertical categories include the class of passenger (first, second, third, crew), and horizontal categories display whether the person was male, female, or a child. The length and width of the bars give indication of how many people belonged to that group. Thus, it is easy to distinguish that women and children had a better survival rate than men due to being boarded in rowboats first. In addition, the class of the passenger mattered, the better survival rate was for those in first class (again since higher classes were able to board lifeboats sooner). Finally, another use of categories is demonstrated by the Category map developed by Yang et al. (2002). Their category map allows the user to browse well-organized structures of the Internet. This self-organizing map is able to compress and transform complex information space into a two dimensional representation. Neighboring nodes with the same label form a region with the same concept. Users are able to change view and system parameters in the visualization. The Category Map then acts as a more traditional navigational tool.

Figure 24. Categorical Visualizations: alternative Venn diagram (Lu and Dietrich 2004), Mosaic (Yul Huh 2004), and Category Map (Yang et al. 2002)

2.5.3 Network

Another difficult data type to display is a network. Unlike hierarchies, networks are not well structured. Entities can have apparent random connections with other entities. Typical visualizations today can handle small-world networks using the traditional node-link model (see Figure 25). However, these visualizations do not scale well due to line, node, and labels crossings (called *occlusions*). Attempts to bring networks into a 3D space have experienced the same, if not more, difficulties. Occlusion and line crossings become even more troublesome. Not only can nodes be overlaid side by side, but depth is also a factor. A node in the foreground can easily obscure a receded one. The sheer size of networks is another concern. Node-links diagrams become so large that the whole network is not visible in one view. A new version of node-link diagrams for networks is a Pivot Graph developed by Martin Wattenberg (2006). Using a grid-based approach, the tool focuses on the relationships between node attributes and connections. The user is able to specify an attribute and "roll-up" or use a selection technique. Roll-up allows the nodes to be aggregated and the edges contracted. A selection results in a sub-graph. Using these methods, the user is able to shrink the graph and reduce complexity, but the true topology of the network is not preserved.

Developed in the mid 90's, hyperbolic visualizations have experienced a rise in popularity for networks. These visualizations place entities and/or attributes around the rim of an ellipse. Links are then mapped between them (see Figure 25). Different types of hyperbolics exist. Examples of include radial convergence (fixed number of attributes are laid along the perimeter of the circle and connections mapped between them), radial implosion (multiple layers of attributes circles within a main circle or nodes within the main circle and linked to each other and edge nodes), oval implosion (same as radial implosion except outside shape is an oval), centralized radial network (where nodes are aligned along the outside of the circle but they all map to a central node or group of nodes), and radial grouping (where attributes are grouped according to some criteria in concentric circles within the main circle). Like node-link, hyperbolics have limitations. Links cross and merge as well in this type of graphic. While attributes are always on the edges of the hyperbole, they have to be very small to be displayed and usually require some form of interaction technique to be read. Interaction is more critical since hyperbolics try to condense a large amount of data in a predefined space.

Finally, a third type of network visualization has emerged. Used mainly in social network visualization, matrix representations (see Figure 25) have many advantages over node-link and hyperbolics. Since there are no links, node occlusion and link crossing does not occur. Clusters become immediately visible. A recent comparison of node-link and matrix representations, performed by Ghoniem et al. (2004), has identified many advantages of matrix over node-link. In this comparison, matrix-based representations outperformed node-link on most of the tested tasks. These tasks were to: estimate the number of nodes, estimate the number of links, find most connected nodes, find a specific node, find a common neighbor, and finally, find a path between nodes. One limitation found of matrices is that path finding is still non trivial. While in node-link diagrams this is trivial due to the available links between nodes, in matrices the same task requires aligning and matching nodes back and forth between corresponding rows and columns. This task is both tedious and error prone (Henry et al. 2007). Without links to guide the eye, it is hard to discern neighbors and routes within the data.

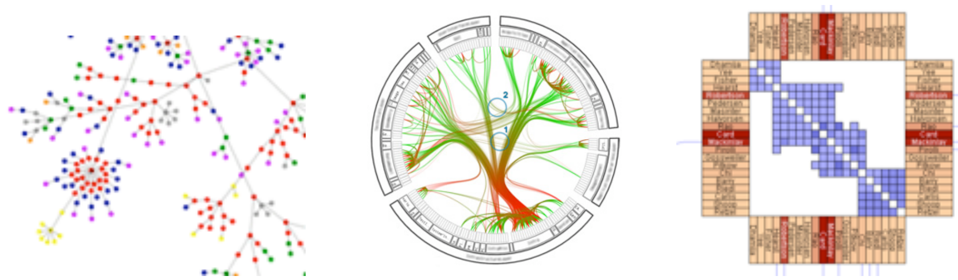

Figure 25. Network Visualizations. Node-link (Salathé 2006), Hyperbolic (Holten 2006), and Matrix (Henry et al. 2007)

2.5.4 Spatial

Geographic information is one of the most recognizable and easily understood domains for viewers. We navigate the world around us from the time we can crawl, so the concepts of space and dimension are extremely familiar. The ability to read maps allows us to move around and find new locations. However, space is more than just representing an actual location on paper or screen. Space is concerned with mapping information to a specific area. Mapping is done by correlating details to a 2D/3D representation (such as maps or globes), placing information in an actual location (Ambient, Augmented Reality), or by providing a sense of place in a totally virtual space (such as Virtual Reality). Geographic information is often crucial. Knowing where something occurred can help scientists and readers understand why the event happened, such as finding an earthquake along the Pacific Rim. Cartography and the use of globes take a literal approach to translating information to a 2D/3D representation. Data is overlaid on top of the known geography (see Figure 26). Data can be used to show information density or specific location of elements. Ambient and augmented reality tries to place information into an actual space for the user to see. Overlaying this information in the actual location lets the user see information concerning that environment at a glance and can inform one on the varying conditions of a location (see Figure 26). Finally, when one is in a totally virtual environment, having spatial clues helps in navigation and way finding. Virtual Reality (VR) does this by recreating the actual environment. A more novel approach is done by the application Chat Circles (Donath et al. 1999)(see Figure 24). In this visualization, members in a chat room are represented as circles. The user can then place their circle near other circles to begin a discussion. The user can only see and participate in the discussion they are near to. Groups are easily seen and act like social cues in real life. Even those who do not participate can be seen rather than remain unknown to the other members.

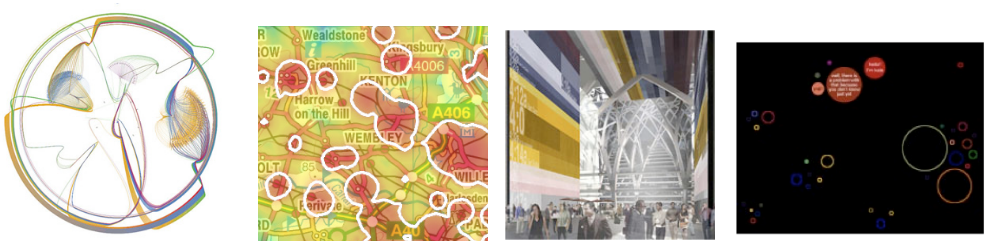

Figure 26. Space Visualizations: Globe (Spahr 2003), Cartography (Lightfoot and Steinberg 2008), Ambient (Rodenbeck 2007) and Virtual Space (Donath et al. 1999)

2.5.5 Temporal

Time is often a critical element of data. Often *when* an event happened can be just as important as *where* it occurred. Temporal data lets us learn from the past, plan for the present, and predict the future. However, time is not always a trivial component to visualize. How we think of time varies. Timed data can be thought of as cyclical or sequential (Parry 2007, Aigner et al. 2007). Seasons and weather patterns are usually repeated over time. As such, this type of visualization is commonly uses a radiating polar chart, circles, and even spirals. On the other hand, specific events unfold step-by-step. These visualizations include the timeline, sankey and flow diagrams (see Figure 27). Another structure of time is branching, when an event occurs and causes two or more separate time events to happen. Most time-based visualizations are customized since it is hard to consider all aspects of the data in a generic way (Aigner et al. 2007). When using temporal visualizations, most users are interested in evolutions in data over time. These evolutions include finding patterns, trends, and anomalies. Common tasks for temporal data is to locate when something started, an event during a sequence, or when an event concluded. It is important to provide tools for the user to be able to move throughout the events in the timeline or give visual clues to how the data changes. Common interactive graphics today employ multiple scales, sliders, overviews, and/or sparklines. The use of interaction can be key with temporal visualizations, as users often need to move between multiple representations and granularity of the time (Aigner et al. 2007). Static diagrams can benefit from motion lines such as speedlines, flow ribbons, or strobe silhouettes (Joshi and Rheingans 2005). The use of banding (vertical shaded bars) can also help separate different time periods.

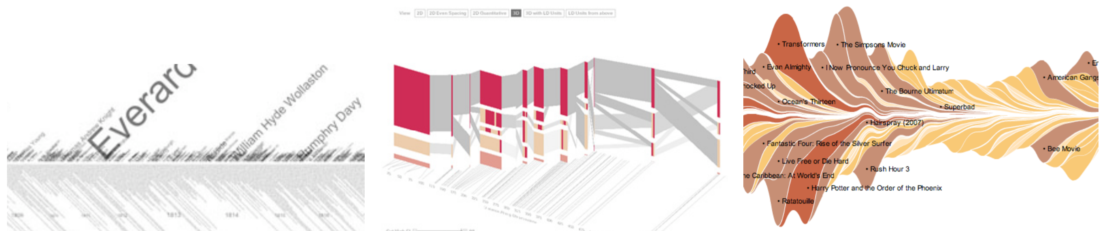

Figure 27. Temporal: time line (Harrison 2005), sankey diagram (Fry 2008), and time flow (Bloch et al. 2008)

2.5.6 Textual

Text is another important type of data. This form of information is how we communicate with one another. Often we try to find relationships from whom we talk to and what is said. An important goal in almost any textual visualization is the ability to find patterns. This task includes not only finding who said something, but also when. For example, the visualization Loom (Donath et al. 1999) (see Figure 28) uses threads to show communications between members in a Usernet group. The dialog characteristics can then be easily seen. The group discussion pictured below has lively debates, with threads being posted often and close together. Another way to visualize text comes from the same creators of Loom. Conversation Landscapes uses line plots to show when text was typed and how long a post was (see Figure 28). Tag clouds, where more common text is displayed larger and brighter than less common text, is beginning to spread to mainstream applications (see Figure 28). Finally, arc diagrams (see Figure 28) allow patterns to be easily identified in a sequence. Sequences that are repeated and/or connected are shown with a corresponding arc. The more often the connection is found, the darker and thicker the line becomes (Wattenberg 2002). This visualization has found use in music, DNA sequencing, and textual comparisons.

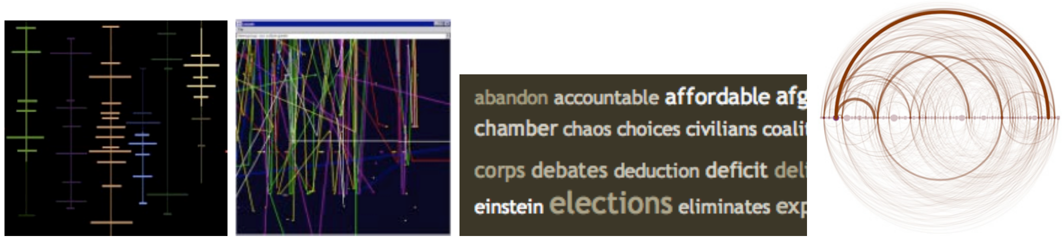

Figure 28. Textual Visualizations: Conversation Landscape (Donath et al. 1999), Loom (Donath et al. 1999), tag cloud (Mehta 2006) and arc diagrams (Dittus 2006)

2.6 Information Visualization Techniques

Overview, filter and zoom, details on demand.
Ben Shneiderman

WHEN GETTING READY to create visualizations, there are many techniques to consider. A designer should be aware not only of how the user will receive output and input data, but also ways to build the visualization and allow the user to interact with the system. This section includes the discussion of interaction devices available today, as well as visualization and interaction techniques. Finally, tasks specific to network visualizations will be presented.

2.6.1 Devices

Today, there are wide assortments of devices that the user can interact with to enter data into an application, system, or tool. Although not as varied as input devices, output devices can further help the user explore a tool. New and novel techniques are continually being explored with the intention of making the hardware integrate seamlessly with the user.

2.6.1.1 Input

When one thinks of input, the more traditional methods are brought to mind such as the keyboard and mouse. These pointing devices are how most users have learned to operate computers. Specialized devices have also spawned the creation of their own specialized input devices. A common example is for video game systems. The controller has unique keys that are mapped to functions within the game. Individual game companies even develop their own version of these input devices, e.g. Nintendo's Wii. Other devices that are widely used are trackballs, styluses, tablets, and joysticks.

Newer devices are always being created. While traditional input devices relied on simple button pushing or keyboard entry, today's input devices can react to user movements, eye position, voice, and even brain wave activity. Phone companies use voice recognition to help their customers through complicated phone message systems. Touch-based displays offer a very intuitive approach to data entry. Kiosks employ this technique to allow the user to navigate menu systems. Other touch devices, such as the recently released iPhone, recognize hand gestures that trigger specific actions on the interface. This technique is employed on personal digital assistants (PDAs) through the use of a stylus. The touch-based display's precision is so refined it can enable a user to point to and select just a single pixel (Shneiderman and Plaisant 2005). Eye tracking, while not as common, is another way to give input into a system. While the user is wears a head-mounted device, the system is able to determine where one is gazing and center focus on that point. This technique can be especially valuable to someone with limited to no functional motor capability. Unfortunately, the cost of such devices is prohibitive for most users.

2.6.1.2 Output

Output devices can encompass more than just the standard visual output. Many systems incorporate multi-modal approaches to giving feedback and presenting data. The more means of communication with the user, the more effective the system can be. In case the user has any type of physical impairment (visual, hearing, motor), it is critical to allow them to receive information in more than one mode. Devices have been developed to target one or two of the five senses, but no one system has successfully integrated all five together in one product. Audible feedback is a common method for output. Many systems use a short beep to alert the user to a condition, e.g. when a computer is turned on or off. A sound when a button is clicked also lets the user know that the system recognized the user's action. However, sound cues can quickly become irritating and ineffective. This type of feedback cannot always be counted on since the user may choose to mute his/her device. Another limitation is for hearing-impaired users, who are unable to detect such a noise. Besides vision and hearing, the sense of touch has been thoroughly investigated in the field of haptics. Haptic devices can give users a sense of touch in a virtual space. The Phantom is a pen-based system that allows the user to interact with and "feel" objects. Unfortunately, these devices are usually expensive and not widely manufactured.

2.6.2 Visual Techniques

2.6.2.1 Overview + Detail

Overview + Detail (O+D) is a technique used to show two levels of information from a single dataset. Its goal is to present the user with an overview of where the user is while showing specific details about the data (usually through the use of zooming or panning). Normally, this type of technique is accomplished with two screens or two separate views of the data. A common use of O+D is in mapping applications and video games (see Figure 29**).**

Figure 29. Example of O+D: Google Maps and the video game Wheels of Steel Convoy

2.6.2.2 Focus + Context

Focus + Context (F+C) is a technique very similar to Overview + Detail (O+D). F+C is concerned with the ability to see particular details and still retain the general context of where the data is situated. The most important data then becomes the focal point at full size and detail. The display area far from the focal point is displayed smaller or omitted. One aspect that differentiates F+C from O+D is that F+C is usually accomplished within one window or view. A popular method of obtaining this technique is through distortion techniques such as a fisheye lens (see Figure 30). Thus, data can be presented at various scales, and the user can select an area of interest to enlarge while preserving the data context.

A downfall of this technique is that many F+C use distortion to such an extent that the data surrounding the area of interest are pushed so far back that they are no longer readable or the context is lost. Examples of this type of technique are used in calendar applications for mobile devices. Other F+C applications include Topical Fisheye (Gansner et al. 1999), TreeJuxtaposer (Munzner et al. 2003), and The Whale Hunt (Harris 2007).

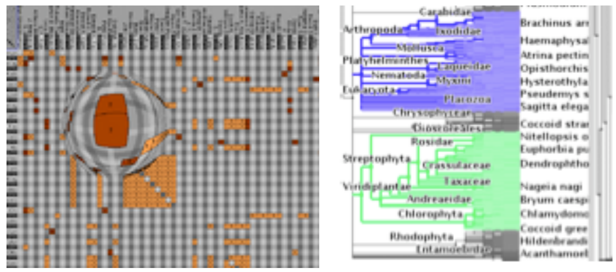

Figure 30. Fisheye Distortion (Fekete 2004) and TreeJuxtaposer (Munzner et al. 2003)

2.6.2.3 Multi-window

Another technique beside Over + Detail (O+D) and Focus + Context (F+C) is the use of multiple windows. Past approaches for visualization have centered on zooming, clustering, filtering, and layout techniques to allow the user to explore his/her data. With large datasets, the data often does not fit in one view, but most visualizations allow the user only one window for the visualization. If one desires more views, he/she would have to run an identical copy of the same program, or another tool along side their current one. This method requires a large overhead in computational power, and the tool functionalities might not be the same between applications (Namata et al. 2007). The ability to see the same data represented in different scales or visualization methods can bring out features of the data not seen in just one view (Unwin et al. 2006, Namata et al. 2007, Shneiderman and Plaisant 2005). Multiple windows also allow the user to see the global structure and sub graphs of complex datasets (Henry et al. 2007, Namata et al. 2007). Multiple windows allow flexibility for views, structure, and feature search (North and Shneiderman 2000). For example, in one view a user may have a global view; while in another view they are able to drill down to get further details. This is very reminiscent of overview + detail views. Other times the combination of different visualizations can be extremely helpful. Nodetrix, developed by Henry et al. (2007), combines matrices and node-link methods. Both have their own strengths, and the combination tries to blend their capabilities. Coordination of actions between the views affords more power to the user. One should be able to brush/link a data point in one view and have the other view(s) auto-update. The tools afforded in one view should also be present in the other views. Current tools with multi-window and multi-representation include gGobi, Dualnet, SocialAction, and PairTrees.

One downfall of having multiple windows is the cognitive load it places on the viewer. The user must switch their attention between views resulting in a strain on working and visual memory. Plumlee and Ware (2006) did a comparison with dual window approach for multi-scale representations. In the comparison they tested a Zoomable User Interface (ZUI) with a dual-window interface. While the multiple window representation did take more time, the ZUI error rate was more significant, especially when pattern comparisons were made that involved more complex patterns than could be held in visual working memory.

2.6.2.4 Labeling

An important part of large datasets is the ability to know what one is looking at. Even when data points are differentiated with any number of pre-attentive features, a textual identifier is usually easier to work with. With large visualizations labels for all data points can become very cumbersome. When data is tightly clustered, labels overlap, making reading impossible. Even when the user is able to zoom in on a data point, labels are still a concern. Some systems do not label the data points at all. This absence forces the users to go one-by-one through the nodes to investigate properties. Most visualizations use static or dynamic labeling. Static techniques focus on finding the best possible layout for labeling. Dynamic techniques include the text being displayed in a rollover or tooltip (cursor sensitive). Another possibility includes displaying labels for objects that are similar to the ones the user is currently looking at (sampling). Finally, labels could appear at specified scale/zoom settings (Fekete and Plaisant 1998).

To help overcome these difficulties, Fekete and Plaisant (1998) introduced a new technique called excentric labeling. This dynamic technique allows the user to select a node or group of nodes and the labels associated with those will be displayed. When the user places his/her cursor over a region of nodes, the labels arrange themselves around the cursor region with lines leading to the node they are associated with (see Figure 31). The line matches the attribute style of its data point. Each label is left justified and does not overlap. However, the authors acknowledge that showing only 20-30 labels at a time is optimal with this strategy.

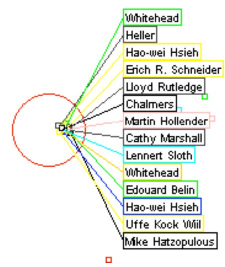

Figure 31. Excentric Labeling from Fekete and Plaisant (1998)

2.6.2.5 Previews

Especially in the digital realm, one does not have the cues that real-life provides. For example, just by looking at a magazine we can estimate how much content it contains. For web-based media and digital applications, there are no such clues. Previews, also referred to as a "visual scent" (Willett et al. 2007), are a way to expose and make transparent the content for the viewer. This display helps viewers make more informed choices. A web-based preview was implemented in WebTOC (Nation et al. 1997), a browser tool that allows users to visualize websites with a hierarchical table of contents (see Figure 32). The structure of the site is exposed as well as valuable cues to where and what type content is contained on the site.

Willett et al. (2007) explored another type of visual scenting by adding social navigation clues to common UI controls. Their work embedded navigational clues in common UI widgets such as slider bars and radio buttons (see Figure 32). After conducting an experiment with their "scented" widgets versus traditional widgets, the authors found that not only did users prefer the scented widgets, but they also lead to more unique discoveries with new data. However, this trend tempered off with time when using the same sites, possibly due to increasing familiarity with the data terrain.

Figure 32. WebTOC and Visual Scent radio buttons

2.6.2.6 Dimension

Dimension is the ability to represent objects in a familiar space. While traditional approaches use 2D, 3D is gaining in popularity. Modern tools can even go beyond. Although time can be thought of as an additional dimension (as found in the new 4D ultrasounds), it usually is regarded as another attribute of the data. The PRIM-9 is able to display data up to the 9th dimension. 2D, with attributes of length, width, and height, is most commonly seen in information visualization and data mining. Since most of the data for this field are abstract, there is no natural physical model to simulate. 3D is more common in scientific data. The added attribute of depth affords life-like models of objects. 3D is naturalistic, and very familiar in applications using real-world materials. However, 3D tools can be hard to operate, take more operations to complete tasks, or be harder to understand. Occlusion can occur from overlapping data. Users can also get lost in the data without appropriate spatial cues and landmarks. Creators of 3D tools should provide an overview, history keeping, landmarks, teleportation (the ability to quickly go somewhere else from a given location), and x-ray vision (the ability to see through objects to see data that has become occluded).

There is debate among researchers about the use of dimension and its impact on visualizations. The difference seems to be task and user dependant. Those with poor spatial ability will not be effective using 3D.

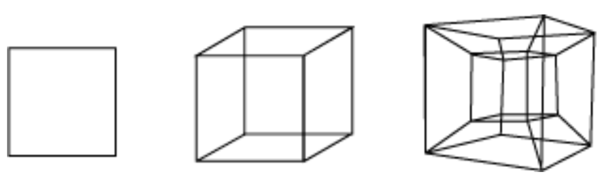

Figure 33. Dimension: 2D, 3D and 4D

2.6.2.7 Animation

Animation is a useful tool for displaying attributes that change over time. Animation can include moving a static image within a scene, an object changing as it moves, or attribute changes (Bederson and Boltman 2005). Another common use for animation is changes between scales, as used in zoomable user interfaces. Interpolation is another time when animation is key. This type of animation help preserve the user's mental model of the data from one state to the next (Henry et al. 2007). Animation within the interface can also cue users to situations when their input is needed or when choices need to be made. However, this type of animation can, and indeed many times is, distracting and irritating for the user. Bederson and Boltman (2005) carried out an experiment to determine if animation helped the user learn spatial information. While users explored a family tree, researchers were looking for impact on navigation techniques, recall, reconstruction of the tree, and user opinions between interfaces that used no animation or one that did. While no effect was seen for navigation (but viewers', who used the animation version, time was no worse) or recall, animation did seem to improve the user's ability to learn spatial position. Users also reported feeling that the animation did help with the tasks. Finally, they found that the transition effects and animations of objects independently helped the user to solve problems.

2.6.3 User Tasks and Interaction Techniques

Interactions that graphics support are based on user tasks/needs. When datasets are complex and large, the user needs to be able to mark the data, explore it at various levels, and change attributes of the visualizations. All these actions enable the user to analyze the information, make discoveries, and test hypothesis. Numerous authors have tried to give an overview of basic user tasks (See Table 2). While many of their tasks are common between authors, there seems to be a divide between high-level and low-level tasks. Shneiderman (1996) and Lee et al. (2006) seem to focus more on the interaction techniques rather than user tasks. Pillat et al. (2005), Winchler et al. (2004), and Yi et al. (2007) come closer to identifying actual user tasks.

Table 2. Authors and their identification of User Tasks

Shneiderman (1996)	Winchler et al. (2004)	Pillat et al. (2005)	Lee et al. (2006)	Yi et al. (2007)
Overview, zoom, filter, details-on-demand, history, relate, extract	Locate, identify, distinguish, categorize, cluster, distribution, rank, compare, associate, correlate, emphasize, rank, reveal	Identify, determine, visualize, compare, infer, configure, locate	Retrieve, filter, derive, extreme sort, range, distribution, anomalies, cluster, correlate, adjacent nodes, scan, set operation	Select, explore, reconfigure, encode, abstract/elaborate, filter, connect, undo/redo, System reconfiguration

In order to unify these methods, this author presents four main overall user goals. All the tasks and techniques from the previous authors should fit within the new classification. These main goals are Explore, Identify, Analyze, and Manipulate. These goals reflect an investigative cycle for visualizations (see Figure 34). Pfitzner et al. (2001) use a similar cycle (formulation, initiation, review, refine), except they label their cycle for User Interaction Phases, which is a understandable since interaction is based on user goals. During the investigative cycle, a user must first explore the data. Then he/she locates some point of interest and performs further analysis. Next, one might manipulate the display to bring a trend or pattern out more or to test a hypothesis. While these steps might not occur in this order, and may be repeated, they all are important steps in using interactive visualizations.

Figure 34. Cycle of Investigation

Exploration is the ability to look at the data. The user should not be limited to just one view, but rather be able to move around and see the data from a variety of viewpoints, scales, and even visualization types. Identify is concerned with locating a specific item that the user has in mind. To locate the user can filter and mark the data. Thirdly, analyze is the ability to make decisions about the data. Sub-tasks for analytical work includes making correlations, comparisons, connections, abstractions, and elaborations. Finally, manipulation allows the user to change aspects about the data and the visualization to make points of interest more clear or to make discoveries of new points of interest. Sub-tasks within manipulation include reconfiguring, encoding, and filtering. Table 3 identifies common sub-goals for each area.

Table 3. User main goals and tasks

Explore	**Identify**	**Analyze**	**Manipulate**
Search, scan, view, move, zoom	Retrieve, locate, select, mark	Determine, distinguish, derive, compare, contrast, correlate, connect, abstract, elaborate, categorize, infer	Reconfigure, encode, filter, emphasize, sort, undo/redo

Each of these goals has a variety of interaction techniques that can be used to support them. Indeed, many can fit more than one category. Below is a list of common techniques and the goals they match.

Table 4. User main goals and interaction techniques

		Goals			
Technique		***Explore***	***Identify***	***Analyze***	***Manipulate***
Zoom		X	X		
Pan		X			
Overview		X			
Filter/Drill-Down		X	X	X	X
Fisheye		X	X		
Brush/Link			X	X	
Place mark			X	X	
Preview		X		X	
Highlight			X		
Directed-walk		X			
Tours		X	X	X	X
Small Multiples		X	X	X	
Tooltip/hover		X	X		

Table 4. (continued)

Touch/Gestures		X	X		X
Magnetic Effect		X	X		X
Rotate		X	X	X	X
Jitter				X	X
Aggregation				X	X
Transformation				X	X
Weight				X	X
Move					X
Subset/reclassify				X	X
Visual technique				X	X
Colorize			X	X	X
Change Size				X	X
Orientation			X	X	X
Change Font					X
Change Shape			X	X	X
Agitate		X			X

2.6.3.1 Direct Manipulation

Many of the interaction techniques presented (such as brush/link, drag-and-drop, touch, and gestures) have to do with some form of direct manipulation. Direct manipulation allows the user to interact with the data on a straightforward level. While this interaction technique is inherently powerful for exploring the data, there are pitfalls that can occur. First of all, these techniques can be a problem for visual-impaired users and those with motor difficulties. Incorporating direct manipulation techniques may be misleading as the user may overestimate or underestimate what they can do. Next, many direct manipulation techniques use a metaphor for the interaction, such as a pen or pencil. Finding the right metaphor can be very difficult. Finally, the reaction time of the system to the user has to be quick for it to be useful to a viewer (Shneiderman and Plaisant 2005).

2.6.3.2 Graph Specific Tasks

In addition to the main user goals, each type of graph will have its own specific tasks that are used for analysis and exploration, such as a network. These tasks revolve around how the visualization shows data, in this case with nodes, links, and paths. Viewers are interested not only in the topology, but adjacency, accessibility, connectivity, attributes (nodes and links), and paths. Common tasks for nodes are to find how many there are, a specific node, a common neighbor, and the most connected node. Users might also be interested in the number of links in a graph, specific link between two nodes, or types of links between nodes. Paths are important to find, such as the longest/shortest path, or the most common path. Finally, viewers might want to discover where clusters are formed, which nodes act as central nodes, the role of the node, or which nodes act as pivots to other clusters (Ghoniem et al. 2004, Lee et al. 2006).

2.7 Interface Design

Poorly designed information and communication technologies cause frustration, confusion, and anger, as well as contribute to social exchanges marked by hostile comments.
Jef Raskin

THE USER INTERFACE (UI) is extremely important for any application that does not rely solely on a command-line interface. "As far as the customer is concerned, the interface is the product" (Raskin 2000 pg 5). The UI is the first thing the user sees and what they have to use to navigate and control any tool. "Complex tasks may require complex interfaces, but that is no excuse for complicating simple tasks" (Raskin 2000 pg 2). If a UI is difficult to operate, a user will often abandon it in favor of something that is easier to use, even if it the other tool is an inferior product. New research even suggests and provides initial evidence that a more visually appealing UI also enhances usability (Cawthon and Vande Moere 2007). A visually appealing interface that builds off of perception and cognitive principle is often more usable. A balance must be found between creating a beautiful interface and just a usable one. Having either a beautifully useless or a functionally ugly UI is not optimal, but perhaps an ugly interface is the more preferable of the two. Interface design should strive for consistency, universal usability, informative feedback, appropriate error messages, error prevention, easy reversal of actions, reduce short-term memory load, and support internal locus of control (Shneiderman 2003, Shneiderman and Plaisant 2005). The question then arises is how to satisfy these criterion. By examining the components of UI design, the designer can strive to meet all these needs. Components of the UI include user goals/tasks, screen layout, navigation, modes, color, typography, feedback, and icon/symbol usage.

2.7.1 Design Considerations

Before the design for a User Interface (UI) can begin, the designer needs to be aware of the goals and intentions of the user. The designer must always be aware that he/she is not the target audience. Many different methods have been proposed to facilitate the collection of user tasks, such as User-Centered Design, Participatory Design, and GOMS. User-Centered design tries to target a design to a target audience through the use of user and task analysis. User analysis includes defining a target audience's wants, needs, and accommodations. Personas, a cliché of a user, affinity diagrams, and surveys are often utilized. Task analysis can be much more complex. While many designers rely on observation, some times this is not possible, or the user does not act the same way when they are being watched. Other techniques include task flow diagramming, task hierarchy, and storyboarding. Participatory design encourages designers to bring in actual users to help create the design. This method can be extremely helpful for user adoption after the UI has been created, and it can help the designer gain a better idea of user wants and needs. However, this technique can be costly in terms of money and time (Shneiderman and Plaisant 2005). Finally, GOMS (Goals, operations, methods, and selection) tries to break down the user's actions into concrete steps. Goals are what the user wants to accomplish with the interface. Operations are the perceptual, motor, and cognitive tasks that need to be executed. Methods are the ways the users will achieve the goal. Finally, selection chooses between several methods available to achieve the goal. This method is helpful for describing the steps in the decision making process while the user is carrying out interactive tasks. GOMS can be extended to include if-then rules to describe various conditions the user could encounter (Shneiderman and Plaisant 2005).

2.7.2 Modes and UI Controls

A gesture is a sequence of user actions that are completed automatically once begun, such as typing a word. Modes are how the interface responds to a gesture. Modes can cause confusion, errors, and restrict the scope of an activity. For example, holding the shift key plus typing will convert the letters to uppercase, but as soon as the shift key is not pressed the letters will return to lower case. Many systems have different responses to a particular gesture, and the response depends on the context. When the response is different to a gesture, the system is in a different mode. Modes can be created by toggle conditions, and these modes are difficult to label (Raskin 2000). For instance, would one label a control Lock or Unlock? If it is Lock, upon seeing the label for the first time the user might think the control is Locked, or perhaps they have to press it to Lock the control. Giving the user choices can lead them to wonder about alternatives or lead to confusion about what they can select. Radio buttons are more appropriate in this case as they are not modal. Once a toggle has been set, the user can forget that it is on. The caps lock key is a prime example. A way around modes is to use *quasimodes*, modes that are maintained kinesthetically. It has been found that the act of holding down a key or other forms of physically holding an interface in a particular state does not induce mode errors (such as in the example of using the shift key to type capital letters or using the command + tab to cycle through sets of choices). This phenomenon occurs because signals reporting back from our muscles do not fade. At best, modes should be avoided. If they must be used, modes should be clearly marked and the commands for one mode are not the same in another mode.

Customization and preference controls can be frustrating modes. While many designers use these tools to allow for a varying array of skill sets, they induce added complexity and usually result in UI arrangements that are not optimal. If the user forgets that they turned off a setting in preferences, they could encounter errors when trying to perform an action or get frustrated when what they want to occur does not happen. Raskin (2000) even argues that using preferences can be a detriment to productivity. However, if the interface is truly deplorable, preference settings could help improve it.

Controls should also give an indication of their effects, which is an attribute called affordance. The more clues the designer can give on how to operate the tool, the fewer errors will result. Picture a typical door in your mind. Where is the handle, and what kind of handle does it have? The handle can automatically tell you if you need to push, pull, or turn it to open the door.

Most commands apply an action on an object. The result is the same whether you select the object first or the action. Using a noun-verb interaction does not set up a mode and is less error prone. Speed is also improved since you do not have to move your attention away from the content. Commands can be provided in a variety of ways. However, there are tradeoffs in terms of complexity of information returned and speed of interaction (see Figure 35). For example, a single click is very easy and quick to perform, but it cannot result in complex information returned to the user.

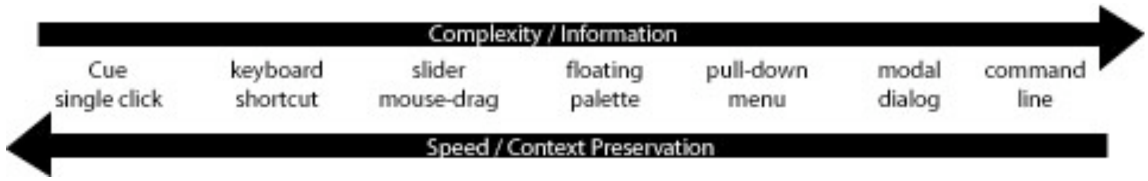

Figure 35. Hierarchy of UI Controls from Unwin et al. 2006

2.7.3 Layout

Before even the final look of a display is chosen, a layout needs to be defined. Layouts are the skeleton on which the visualization rests. If the underlying structure is not consistent or clear, anything that is designed on top of it will not be either. Structuring data and tools allows the user to navigate easily and know what to expect. When placing items on the display, one should keep in mind Fitt's and Hick's Laws. Fitt's Law describes how quickly a user can get to a target on a monitor. Items nearer to the pointer and items that are large are easy to get to. Corners are also easy points to reach. However, some are more accessible than others. The order of access is as follows: Lower right corner, Upper left corner, Upper right corner, and Lower left corner (Thissen 2004). Before you can move your cursor to your target, you first have to decide what to click. Hick's Law states when "you have to choose to take one among *n* alternative actions and when the probabilities of taking each alternative are equal, the time to choose one of them is proportional to the logarithm to the base 2 of the number of choices, plus 1" (Raskin 2000 pg 96). Simple stated, it means making decisions takes time. Giving more options at one time is usually faster than multiple menus. Scrolling is also an issue in designs. Many times the design is too large for a single screen. However, users usually do not like to scroll. Items that are the most crucial should be placed in locations where the user will not have to scroll to (Thissen 2004). Apple Inc. (www.apple.com) does a great job of this technique by designing their web sites so that the most crucial information is always visible with out have to resize the window or scroll. For example, on web pages, the main navigation is almost always on the upper left-hand side of the screen. This position is immediately visible when a page loads and stays visible when the window is resized. Horizontal scrolling should be avoided as much as possible.

Placement of items on a display should be well planned. The principles of Gestalt can help determine how to make groupings and related items easy to detect. Groupings should be meaningful, with consistent sequences in orderly formats. Surrounding blank spaces or boxes can set off groups. Highlighting, background shading, color, or font choices can show related items. Effective designs usually contain a middle number of groups, from 6-15 (Shneiderman and Plaisant 2005).

2.7.4 Navigation

Visualizations should be easy to navigate. A common saying heard is that interfaces need to be intuitive. However, what does intuitive really mean? It means that the interface should seem familiar to a user. Items are placed where one would naturally associate or place them (they are looking for markers and familiar features). This mental mapping is called a cognitive map. The closer a UI can come to a user's cognitive map, the more successful it will be. When a UI does not match a user's mental map, they can be come disorientated, frustrated, and insecure (Thissen 2004). There are three main types of users to consider when designing navigation: beginners, intermediate, and advanced users. Each group needs different support for navigating. For example, beginners need a lot of cues and guidance such as overviews. Intermediate users know the system, but like to learn new short cuts and are still helped by cues. Experts know the system and can zero in on points of interest. While each user group has slightly different needs, it does not follow that there has to be three different interfaces for navigation.

When designing navigation there are important aspects to keep in mind. When exploring digital space, the user needs plenty of guidance and landmarks. From any point in the visualization, the user needs answers to the following questions: Where am I now?, What is the structure?, Where have I been?, What is available here?, Have I seen everything?, Have I overlooked anything important?, Where is the info that is relevant to me?, and finally, Will I succeed quickly? (Thissen 2004). The design should allow the user to move around quickly and easily. Not all navigation will be linear; e.g. web navigation is full of hyperlinks allowing the user to move anywhere in or out of a website.

Many designers use metaphors to help the user have a familiar background to start from. Metaphors can be highly effective and are used often to make orientation and navigation easier. The use of a known concept helps the new interface seem familiar and cuts down on extra learning. However, metaphors can become too pervasive and can actually distract users from the content. A use of a common metaphor is the folder. When computers were first created, their main use was in business. Businesses routinely used folders to hold important papers. This idea was transferred to electronic files. This metaphor is currently reaching the end of its usefulness today. Folders become lost within folders, and users forget where files are placed. Other users do not like to use folders and pile files on the desktop. More often users use the search capability of the computers' operating system to locate desired content. Certain characteristics should be kept in mind when using or creating new metaphors. First of all, the metaphor should fit the topic and content. It should also be simple but not so simple that it is boring. The whole point of a metaphor is to create a familiar situation, and the more realistic the representation the more a user will trust the design. Perhaps most importantly, metaphors should be used consistently and uniformly. Mixing them causes confusion and destroys their effects.

2.7.5 Color

Color is a key component of any design. It can be used to add personality, but also to separate content and provide visual cues. There are many considerations that one needs to keep in mind when using color. There are types of color based on whether the visual will be on printed media or digital. The cultural considerations can affect how users will react to color choices. Finally, considerations on the number of colors and the ability of users to perceive them are important.

2.7.5.1 Color Theories

There are three main theories of color: Additive (RGB), Subtractive (CMYK), and HSL (Hue, Saturation, Lightness). Additive color is based on the combination of light waves. When all light rays are added together they will create white. This type of color model is used for digital display. Subtractive color is obtained by adding pigments together. When colors are added together, they create a dark brown-black color. CMYK, based on subtractive theory, is used for print media. Standing for Cyan, Magenta, Yellow, and Black, these are the four main colors that are used to create all other colors for print. Both CMYK and RGB are subsets of the possible colors that the human eye can detect. Of all the colors, only a few are perceived as "true." For example, most people will pick out one shade of yellow that is a *true* yellow. Green almost has this same characteristic. Finally, HSL stands for Hue, Saturation, and Lightness. Hue is the actual shade, Saturation is how much of the hue there is, and Lightness is brightness level. HSL describes points in a cylinder whose central axis graduates from black to white. Angles around the axis correspond to saturation and distance from the center is lightness (see Figure 36).

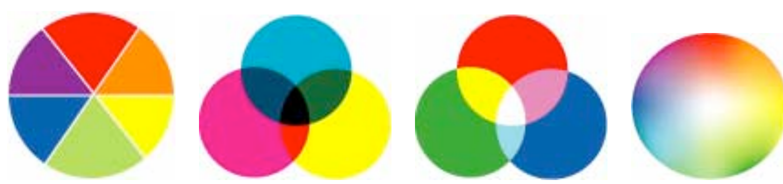

Figure 36. Color wheel, CMYK, RGB, and HSL

2.7.5.2 Color Schemes and Patterns

Choosing an appropriate color scheme for visualization is made easier with the help of pre-defined color schemes. Common schemes include monochromatic (tints and shades of one color), analogous (colors next to each other on the color wheel), complementary (colors opposite each other on the color wheel), split-complementary (same as complementary except choosing two colors from either side of the two complimentary colors), triadic (3 colors equal distant from each other on the color wheel), and tetradic (two pairs of complimentary colors). Monochromatic colors are unobtrusive and restful. No one color stands out. In analogous use, one color becomes dominant and the other hues are accents. Again, there is no stark contrast. Complimentary colors, on the other hand, provide great contrast.

Another factor to keep in mind, when mapping attributes to colors, is the type of information the colors are representing. A limited color palette is advisable as color becomes non-preattentive when a large number of colors are used. Variations of color also intuitively mean different patterns (see Figure 37). Tracking one variable through different levels is best achieved by sequential colors (monochromatic or analogous). With many different categories, multiple non-related colors are best. Finally, when using two attributes and their degrees can be represented with two colors. To choose colors, many tools are available including ColorBrewer (Brewer and Harrower 2008) and Color Consultant Pro (2008).

Figure 37. Color patterns. Sequential, Categorical, and Diverging

2.7.5.3 Color and Culture

When using color, one must take care with not only the combinations of colors chosen, but also in choosing the hues. Some colors have very similar meanings across cultures, but others do not. For example, while the Western world perceives red as representing love or passion, it has embodied evil in Egypt. Table 5 gives a listing of colors and the various associations with them.

Table 5. Color and associated meanings (Thissen 2004)

Color	Western meaning	Other Meanings
Red	Love, Passion, Power	Hebraic: splendor, Egypt: evil
Blue	Depth, tranquility	Buddhists: wisdom, Middle East: fidelity
Yellow	Optimism, warmth	American Indians: death, Asia: prosperity
Green	Hope, nature, youth	Hindu: death, France: unlucky
White	Purity, innocence	Asia: death
Purple	Royal, power,	Homosexuality
Orange	Energy, fun	Japan: love and happiness
Black	Death, evil, elegant	Buddhism: oppression, Egypt: rebirth

2.7.5.4 Color Considerations and Color Blindness

Careful consideration must be also given to the number of colors in a display. Users are unable to differentiate between multiple colors easily that are not adjacent. Colors can also be perceived differently on contrasting colors around them. For example, a hue may appear lighter on a darker background, or two different colors may appear similar on certain hues (see Figure 38). Too many colors can also overwhelm the user. Common practice has been to limit the number of colors to around 6. The use of transparency (or the Alpha channel) of color can be a powerful tool to show differences in data. This technique allows the most important trends and patterns to stand out while the other data recedes into the background.

Figure 38. Color contrasts. The inner blocks on the left are the same, while the ones on the right are different

Finally, not all users can see alike. A small percentage of the population has different types of color blindness. Typically, this condition is seen in males. This blindness is the inability to perceive differences in various hues of color and can affect normal life in a number of ways. For example, one of the most common types of color blindness is red/green blindness. All traffic lights in the U.S. use red and green. A person with this blindness must rely on the position of the lights to discern when to stop or go. Weather maps are also another source of frustrations as the majority use red and green. Other types of color deficiencies include: protanopia (red/green deficit), deuteranopia (another form of red/green deficit), and tritanopia (yellow/blue deficit). See Figure 39 for a look at what people with these blindnesses perceive. To help those with color blindness, designers need to remember to vary the intensity and not just the hue. If one wants to check how those with color blindness see a design, tools such as VisCheck (Dougherty and Wade 2008) are available. VisCheck is able to take a given website and render it according to a specific colorblindness.

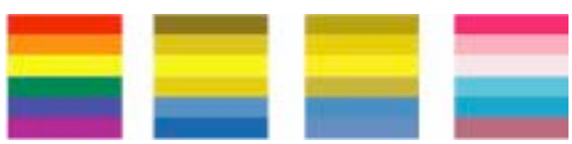

Figure 39. Colors as seen by a person with normal vision, protanopia, deuteranopia, and tritanopia

2.7.6 Typography

Typography is the building block of design. Nothing else can communicate so much information. However, it is not just the words that letters form that are important.

Typography has a form and shape and conveys the personality, mood, and meaning of the writer's ideas. The two main types of type are Serif and Sans Serif. What differentiates these two types is the use of serifs, recognized as decorative hooks and curlicues on letters. Serif fonts have traditionally been used in printed media (See Figure 40). The serifs on the letters ground the letters, enabling easy reading. Sans serifs have found their use in digital media. Since the resolution on most computer monitors is poor (average graphics being 72 dpi versus 300 for print), sans serif has been found to be more readable as the fine strokes and edges of serif fonts cannot be displayed well. However, above a certain size, such as 16 point, serifs are legible. The opposite is true for print work; when writing for print, serif fonts are best for body copy and sans serif are useful for headings that are around and over 20 points. For monitor fonts Myriad, Minion, Verdana, and Arial are best. Popular serif fonts include Times and Garamond. Another method for making type more readable is anti-aliasing. This technology eliminates the stepladder like effect of vector graphics. Instead, letters are shown by pixels creating a smooth effect (Thissen 2004).

Figure 40. Letterform showing serifs

The structure of type is key. Structure is provided by spacing, justification, and layout. Not only is the type of font important, but spacing between lines and letters is as well. Leading is the space between lines of text. This term is derived from the days of letterpresses. Bars of lead were used to physically separate the lines of text. Digital displayed type needs more spacing than print. Thissen recommends at least 1.5 to 2 lines for running text. Kerning, on the other hand, is the spacing between individual letters. Many fonts do not leave enough room between certain characters, and the two letters can appear to run into each other. Typographers also have to worry about the alignment of text. Left-justified, ragged paragraphs are easier to read. Right-justified text is very difficult to read. Designers should also keep empty spaces around text. These can act as resting spots for the eyes, as well as help to emphasize content. Finally, the more structure you can use with typography the better. It makes navigation easier. An example of adding structure is to create lists.

When using typography, background can play a key in readability. A strong contrast is needed so type does not blend in or become lost (see Figure 41). Background pictures can seem like a good idea, but it can make reading more difficult. The best solution is black text on white background. Any highly saturated background can be too intense.

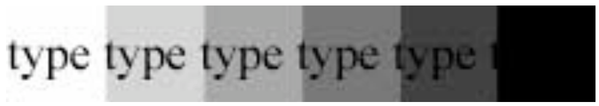

Figure 41. Type with varying contrasting background

2.7.7 Icons, Symbols, and Imagery

Many designs rely on the use of icons, symbols, and imagery to summarize data or to describe functions available by a system. Good symbols and icons are easy to recognize (see Figure 42 for universally known symbols and logos). Well-crafted icons, symbols, and images are easily perceived, motivate users, and allow users to react more quickly. However, many times the design is perceived as obscure or the designer chose a bad metaphor.

Figure 42. Universal Symbol for Man and Apple Logo

Imagery is common tool for interfaces. We react quickly and intensely to photographs. A user can recognize a clear picture in 1/1000 of a second. We are also able to remember a very large number of pictures. Our memory capacity is larger for images than it is for text, and we can process images with less mental exertion (Thissen 2004). Photographs also have the potential to have great, often unconscious, influences on behavior.

Icons are a 64x64 pixel representation of an action, a noun, or functions. Icons require little space and are language independent. Even children and the non-literate can understand them. However, their meanings can be obscure and limited (Thissen 2004, Raskin 2000, Shneiderman and Plaisant 2005). The context they are used in can affect their perceived meaning (See Figure 43). Icons must be visually distinct and represent the underlying idea well. It is best to present them at a reasonable size, usually larger than the text. Combining a label with them, or even just a tooltip, improves their effect. Icons should be used sparingly, and only for necessary occasions. Icons should be grouped by similar or task-based functionality, e.g. many word processors group the copy and paste icons together. Finally, icons should be functional, not just pretty. If they do not help the user with accomplishing a goal more quickly, they should not be used. When creating them, define the advantage for using them and the purpose they serve. Designers should keep in mind the target audience and any previous knowledge their audience has. Sketches of possible looks should be done with user testing to determine strengths and weaknesses. Similar to icons, symbols are pictorial representations that express more than can be detected at first glance. Take for example the Apple Inc. symbol (see Figure 42). While it is a literal interpretation of the company name, the apple stands for much more. In the Bible, the apple represents knowledge. When Adam and Eve ate the apple, they gained knowledge. The bite out of the Apple's symbol implies that the user who has eaten of it has gained knowledge. This underlying meaning is not apparent for those who are not familiar with the mythology. Many times the symbol needs explanation.

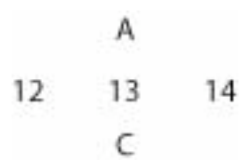

Figure 43. Context matters: from left-to-right it read 12 13 14, but top-down it is A B C

2.8.8 Feedback

Feedback is crucial for the user. Since computer systems are basically black-box designs, the user is unaware of actions that are taking place behind the screen. Without knowing what is happening, the user can become frustrated, lost, or try to re-perform actions. Feedback needs to happen within 1/10 of a second for the user to feel like the system has responded immediately to their action (Thissen 2004). At about one second the user will still think the system is acting ok, but any longer and some type of alert needs to be given, whether through the user of a dialog message or a loading bar. Reactions to the user can take the form of visible highlighting, sounds, messages, or cursor changes. If dialog messages are used, the address must be appropriate for the user group. Almost no one can understand a message that is written by and for the developer of the actual computer system. Messages must be friendly, polite, and give possibilities for decisions (Thissen 2004).

Feedback is also necessary for the various UI controls. Many common menu controls use background highlighting and shading to denote user choice. Font style and size are also common attributes to change. Designers must also be aware of conventions. For example, in web design it is common for hyperlinks to be underlined or blue in color. The method of feedback should be carefully chosen.

2.8 Visualization Tools & Toolkits

We shall neither fail nor falter; we shall not weaken or tire...
give us the tools and we will finish the job.
Winston Churchill

THE NEED FOR visualizations are growing, now more than ever. While for the technically inclined it can be relatively easy to pick up a new language, those not familiar with computer science concepts can struggle with the technology that is needed to create the application they envision. Often domain-specific knowledge is required to not only determine what needs to be visualized, but also how to implement the strategy (Heer et al. 2005). Due to this specialization, there is a growing demand for tools to help create these systems. Tools and toolkits aim to help bridge the gap from the desire to create something and the unfamiliarity with the technical know-how. Tools are products provided to visualize a given type of data set. While users can choose different visualizations, one is unable to combine visualization characteristics or change the underling tool. Examples of tools include exploRase, Manet, and Mondrian. Toolkits, on the other hand, allow for greater customization of the visualization. These toolkits target different user groups. Some toolkits are programming based, enabling the user to specify almost all parameters and types of visualization. Others are widget-based, and the user can choose from among pre-implemented visualizations or combine them to form a new tool. Current favorites in the field include processing, prefuse, and Infovis. Other tools and toolkits include: gGobi, Flare, Piccolo, Ploticus, Pajek, Tulip, and Guess.

Fekete (2004) and Shneiderman worked to identify criteria that software tools must meet in order to be labeled for and used in Human Computer Interaction. These criteria include:

1. Part of the application was built using the tool (data structures, presentation part, interaction)
2. Learning time: long
3. Building time: short
4. Methodology imposed or advised
5. Communication with other subsystems
6. Extensibility and Modularity

2.8.1 Processing

Ben Fry and Casey Reas developed Processing (http://www.processing.org/ or http://www.proce55ing.net) in 2004 from ideas they explored while at the Aesthetics and Computation Group at MIT Media Lab (Fry and Reas 2008). One of its main goals was to introduce programming in the context of digital art and make the computer an accessible medium for artists (Reas and Fry 2003, Reas and Fry 2004). Both wished to create a language that would make images responsive instead of creating a visual programming language. Processing translates code, written in its own language, into Java and then compiles it to an executable Applet. A custom 2D/3D engine is included. Today, Processing consists of a web site, programming language and environment for learning computational design, a sketchbook for rapid prototyping, a 2D and 3D graphics API, and rendering engine for Java. Processing is an open project with an active community of a few thousand people who are using the software. The toolkit allows users, who are students, artists, and researchers, to program images, animation, and interactions. Available as a free download, Processing is cross-platform supported (on Mac and Windows). A wide variety of visualizations and interactions are possible with this toolkit.

A wonderful example of what is possible with this language is the Fidg't Visualizer (see Figure 44) (Protohaus 2008). Fidg't visualizes data from a variety of social networking sites such as Flickr and LastFM. Users specify desired networks and then can create magnets, which are key words or tags. Corresponding users or user content that fit that magnet will then gravitate toward it. Fidg't then lets one discover what topics are popular in a network, common characteristics between users, and common content. Users can also locate other users or compare the user network to a random network.

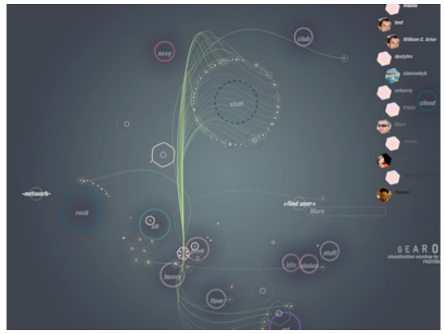

Figure 44. Fidg't Visualizer

2.8.2 Prefuse

Created by graduate students at UC Berkeley in a computer theory course, Prefuse was initially only a set of support classes for experimenting with different visualization algorithms. The name came from a song, by the group prefuse 73, that the creators were listening to while they committed their code to a repository. While the name is different, it is not unusual. Many Java toolkits are named after music elements (e.g. Jazz, Piccolo) (Prefuse 2008). Prefuse was written in Java, using the Java 2D graphics library. Today, Prefuse is an extensible software framework.

Prefuse is free open source API written in Java. To date, there are two main versions of Prefuse, the standard Java implementation and prefuse flare, which includes animations and visualizations for ActionScript and the Adobe Flash player. The toolkit was modeled after the information visualization reference model, a software architecture pattern that decomposes the visualization process into a series of discrete steps (Heer et al. 2005). These steps include data acquisition, modeling, visual encoding, and presentation. Prefuse can be used to build not only standalone applications, but also to embed visual components in large applications and web applets. Processing and representing information can be a difficult and long process, and prefuse was built to simplify these goals.

Prefuse includes support for

1) Table, graph, and tree data

2) Layout and data encoding techniques
3) A library of interaction controls (common interactive direct manipulations)
4) Animation support
5) View transformation (pan/zoom)
6) Dynamic queries
7) Integrated text search
8) Physical force simulation engine
9) Coordinated multiple views
10) Built-in SQL like expression language
11) Support for queries to databases, and
12) An API.

Users of prefuse are expected to have a general working knowledge of Java, including set up and project creation. Experience with Swing and databases are also beneficial. Many new visualizations have been created using prefuse by people in both academics and industry, such as the NameVoyager (see Figure 45), DocuBurst, Zone Manager, and Social Action. Prefuse is free for commercial and non-commercial use under the terms of a BSD license.

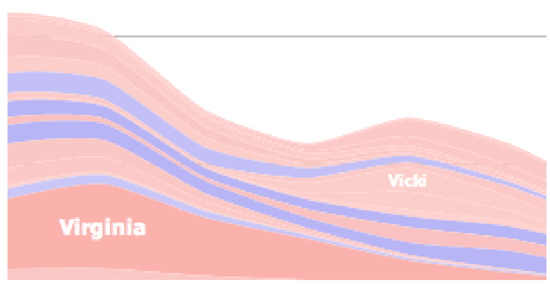

Figure 45. NameVoyager created by Martin Wattenberg (www.babynamewizard.com)

2.8.3 InfoVis Toolkit

The InfoVis toolkit (http://ivtk.sourceforge.net/) was created by mainly by Jean-Daniel Fekete as a way to create, extend, and integrate advanced 2D information visualization (IV) components into interactive Java Swing applications (Fekete 2004). The toolkit, built with approximately 30,000 lines of code, is intended for the application-level programmer. The InfoVis toolkit provides support for tables, trees, and graphs. Nine visualizations are included with the toolkit are Scatter Plots, Time Series, Parallel Coordinates and Matrices for tables; Node-Link diagrams, Icicle trees, and TreeMaps for trees; Adjacency Matrices and Node-Link diagrams for graphs. Components for use with the visualization include dynamic labeling, fish eye lenses, sliders, and control panes to configure and control the visualization. Mechanisms and components also include the ability to select, filter, and perform generic IV tasks (i.e. dynamic queries, filters, selection, sorting, attribute manipulation).

The InfoVis toolkit is organized into five main parts, which are tables, columns, visualization, components, and input/output. Visualizations can be stacked. The underlying structure of visualizations is a table of columns. Columns contain objects of homogeneous types, such as integers or strings. Each visualization has a list of attributes that are associated with specific columns. Attributes include color, size, label, transparency, and sorting order. Trees and Graphs are derived from tables. Key characteristics of InfoVis are its unified data structure, small memory footprint, accelerated graphics support, unified set of interactive components, and extendable framework.

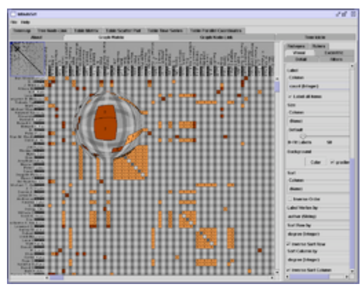

Figure 46. Matrix with Fisheye distortion

2.9 Evaluation

Supposing is good, but finding out is better.
Samuel Clemens

ONCE A VISUALIZATION system has been created it is not enough to use the tool and to naively assume that it has reached its full potential. While most designers do not take the final step in the design process, a tool needs to be evaluated and proven effective for the task it was created for. Evaluation can happen at different times in the development cycle. Ideally, the visualization should be tested at various periods while still in development. These tests can catch problems before the product is deployed. Just as there are many kinds of visualizations, there are different types of evaluation. Evaluations can be done to make sure a tool meets its design objectives or to test content. Other evaluations are done for holding accountability, decision-making (e.g. whether to continue funding), or accreditation. Most evaluations for visualizations aim to prove effectiveness of a given tool. However, effectiveness depends on both the data used and the users.

Evaluations are not done for a variety of reasons. They require a lot of thought, effort, and time. Many visual designs are the product of some new novel concept or idea, and the design might not need to be proven effective, easy to use, or useful. However, when the visualization is going to be used as a tool for research or business, making sure it is effective is highly important. Evaluating can be a difficult process, especially for visualizations (Foley 2006). After all, visualizations are meant to inspire insight, and how can that be evaluated? Most tasks focus on testing how users find numbers or patterns, but this does not necessarily lead to discoveries. Testing might not even prove what a researcher hopes. While a tool might perform well for the given tasks, that does not make it useful to other users. Evaluations for visualizations should test not only the user interface (UI) controls, but also how easy it is to accomplish tasks. In a 2004 survey of literature of 50 information visualization user studies, Plaisant identified four trends: 1) controlled experiments, 2) usability evaluations, 3) controlled experiments comparing 2 or more tools, and 4) case studies. Comparisons of 2 or more visualizations must be done very carefully. Each tool must be the best possible, for a great implementation of low-effective design may perform better than poor implementation of a high-effective design (Foley 2006).

The evaluation itself must have strict criteria. The results from it must be accurate, reliable, and most importantly, valid. The evaluator must follow a strict process for the evaluation. First of all, he/she must define a list of objectives for the foundation of the evaluation, i.e. why is this evaluation being done. Points of interest might be to improve materials, determine time to learn, or ease of use. Next, the audience for the report has to be defined. The terminology, depth of detail, and types of information collected are going to differ based on who the end reader will be. Once the goal and audience are identified, the specific objectives of the evaluation need to be determined. For example, interest in user attitudes, learning outcomes, or quality of the tool. These objectives should be written questions or statements. Then the evaluator(s) must determine what resources they will need to conduct the evaluation, e.g. number of users, data collection instruments, and equipment. At the same time resources are being decided upon, the types of evidence that will be collected must be chosen. The types might depend on the sample size, control of outside factors, testing environment, or need for statistical reporting. Once the type of evidence is known, data-gathering techniques can be decided. The test is then run and evidence collected. Analysis is the next step. How far the analysis goes depends on the type of evaluation. Evaluations meant to deliver informative feedback to the designer do not necessarily need to have in-depth statistical analysis. On the other hand, if the evaluation is to test the outcome of a product, it might. Finally, a written report is created detailing the evaluation goals, process, and results.

2.9.1 Types of Evaluations

. When planning an evaluation, one must take into account what kind of evaluation to perform. Three different types of tests exist: usability, formative, and summative. Usability tests are used to test a user's interaction with a tool/product. These types of tests ask the user to perform common functions with the product. Usability tests test User Interface controls and functionality. Evaluators are looking for time required to complete tasks, productivity, errors, and user satisfaction. However, usable does not always equate to useful. For a product to be successful, it has to fulfill a need. Formative evaluations test the product according to its stated goals/objectives and usability. Formative evaluations are carried before the product is finalized. The findings from these reports are sent back to the product developers for further refinement. These evaluations can be done throughout the life of the product. Carrying out these types of evaluations can help identify if a product is useful and useable. Summative evaluations are done at the end of the product development. Like formative, summative is concerned with how a product has met its goals, purpose, and effectiveness. Many times summative evaluations are done to decide whether to continue a program or product.

2.9.2 Visualization Evaluation

Evaluating visualizations can be more difficult than ordinary products and programs. Visualizations are concerned with representing abstract information for the purpose of insight. While the exact numbers (and the accuracy, speed, and error avoidance) are important, a tool is not effective if it cannot help a user come to some new discovery or insight (Foley 2006, Shneiderman and Plaisant 2006). These tools can help answer questions that were not originally thought. Unlike other tools, users of visualizations often need to look more than once at a dataset, at different views of the data, and over a longer time period (Plaisant 2004). Visualizations also facilitate collaboration, proving that having a multiple set of eyes on the same data can rouse new questions and hypotheses. Shneiderman and Plaisant (2006) propose that documenting usage and expert users' success in attaining their professional goals can assess a tool's effectiveness.

Different types of evaluations include task-based, cognitive-based, experimental, comparison, and in-depth long-term case studies. Task-based evaluations use task lists for the end user to perform. By doing these tasks, a tool's functionality, uncertainty, and cause and effect can be exposed. Cognitive-based tasks use the GOMS model of documenting low-level tasks that the user must perform to complete a task. Users are usually given three different types of tasks; read, compare, and find trends. This model assumes the evaluator knows the scanning sequence the user performs, and it only tests for facts. Experimental evaluations are usually used to test new and novel visualizations. It relies on mostly quantitative analysis with only a partially qualitative. This type of test also relates to comparison evaluations. In these evaluations multiple visualizations are tested with the same dataset and task list. The effectiveness then is in relation to other tools. However, one must be cautious when conduction comparisons as multiple tools usually do not have the same functionality or features. Many researchers use "seeded" datasets as well as benchmark tests in their evaluation. Seeded datasets are when the researcher is aware of insights that the user should find. Finally, case studies document the processes of only a few users of a system. Shneiderman and Plaisant (2006) argue for the use of in-depth long-term case studies to really get at the heart of a tool's effectiveness. They believe that controlled experiments of features are too narrow a focus, and controlling all the difference between tools is too hard. Taking an ethnographic stance, this method observes a few expert users of a tool over the course of a few weeks or months. The users are observed, interviewed, surveyed, and their performance automatically logged. Long-term studies are notorious for being expensive and difficult to conduct. The researcher must observe the user from training all the way to proficiency with the tool. This method also requires that the researcher be in close touch with all test subjects and in their real work environment.

CHAPTER 3. METHODS AND PROCEDURES

The true method of knowledge is experiment.
William Blake

IN THIS CHAPTER we will address the importance of why creating visualization tools for biological networks is crucial to the field of biology and bioinformatics. A brief introduction will be given for current knowledge in biology, biological visualizations and tools. We will then cover the process undergone to create a new interface design for biological network visualization.

3.1 Background

Many biologists today are interested in specific genes of plants and animals. While many different types of genes have been identified, gene roles and functions are still not clearly understood or defined. Discovering the roles of genes is extremely important for work in identifying important characteristics such disease or age resistance. Universities and corporations around the world are creating tools and visualizations to help scientists explore genes and their roles.

3.1.1 Biology

DNA (Deoxyribonucleic acid) contains specific genetic instructions that are used in the development and function of all living matter; basically it contains the blueprint needed to construct all cellular components. DNA sequences are also involved in regulating genetic information. In order for a organism to survive, it must regulate cellular processes. The area of DNA that contains this information is a gene. A *gene expression* is the process in which inheritable information is made into a functional gene product, such as protein or RNA. A gene contains genetic information as well as the sequence for Ribonucleic acid (RNA), a nucleic acid that transmits genetic information from DNA to proteins. RNA is crucial because it assists cells in the creation of proteins. The process, stimulated by enzymes, to convert an entire gene to RNA is called transcription. Certain types of RNA even regulate which genes are active. RNA, usually single-stranded, has different types, and one kind is messenger RNA (mRNA). mRNA defines one or more protein sequences, and is ephemeral, i.e. more is made as needed. The mRNA is used to make a matching protein sequence, a process called translation. mRNA delivers this sequence to a cell's ribosome to create needed proteins. These proteins are then responsible for building the structure of the organism's body and helping different chemical reactions take place. These reactions are called pathways. Pathways can be very complex, requiring a lot of different resources to function properly. Many different pathways can exist within a single cell. Pathways are important to maintaining homeostasis within an organism. Metabolic networks are a collection of pathways.

Scientists are able to knock-out (get rid of) or silence (suppress) specific genes in order to test different conditions. To identify the function of unknown genes, scientist use *microarray analysis*. Microarrays, also known as gene chips, contain 100-20,000 DNA gene samples and possibly 2 – 80 experiment conditions. These chips are made of glass or nylon substrates. Each chip contains specific DNA samples plotted in the array by a robotic printer. A fluorescent labeled mRNA from an experimental condition is spread over the chip. The mRNA will bond strongly/weakly with certain DNA. Using a laser scan, sensors detect the various levels with which the sample expressed each gene. The level that each gene is expressed can then lead to recognizable patterns and help identify function (Seo and Shneiderman 2005).

The ability to recognize gene functions faces several obstacles. For example, identification can be difficult with genes that exhibit similar profiles. Another obstacle includes the sheer volume of data that is generated with these types of experiments. Scientists can also compare gene interactions using identified pathways. However, there are various pathway databases, and these databases are based on primary publications. Unfortunately, consistent terminology and identifiers have not been agreed upon. In addition only certain organisms have even moderately documented pathways. Finally, agreed upon pathways are subject to change based on the introduction of new research.

3.1.1.1 Investigative process

To understand genes behavior and regulation scientist run complex tests using one or more genes. The results of these tests are then run through various statistical evaluations.

Often, the use of visualizations at this point can show clusters of genes as well as any gene that does not behave as expected. Since some genes have very similar profiles, it can be hard to determine what exactly causes reactions to take place. Scientists rely on databases of stored experiments and papers to illuminate what genes they are looking at and to help determine roles. After an interesting gene(s) is found, the next step is determining its overall function within a cell. Pathways, the chain of chemical reactions, are needed to show cause and effect relationships. Known pathways are also stored in databases for scientific use. See Figure 47 for the investigative process.

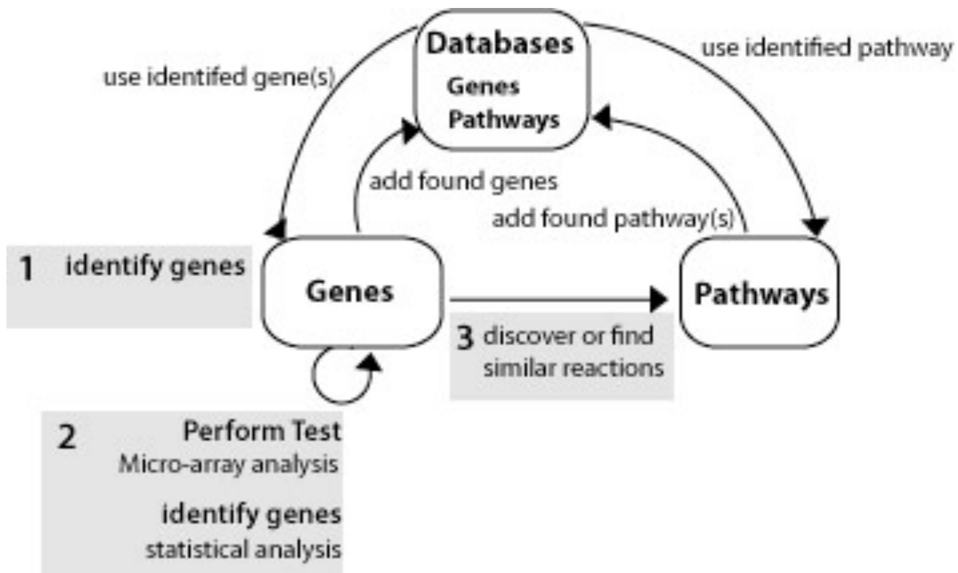

Figure 47. Discovery process

3.1.2 Biological Visualization Tools

While researchers can determine interesting genes and expressions using statistical measures, it is hoped that the use of visualizations of genetic data will decrease research time and lead to new discoveries. Tools have already been developed for each stage of the discovery process (see Figure 47).

Several databases exist containing biological data available to scientists. One of the most difficult aspects for using these resources is the fact that there is no standard method of organization, and the scientific community has not agreed on universal identifiers for different genes. TAIR (2008) provides the AraCyc tool. AraCyc is a visualization tool for biochemical pathways of *Arabidopsis thaliana* (mustard plant) and is supported by the

Pathway Tools software. AraCyc includes a mix of information extracted from peer-reviewed literature and computationally predictions. The MetNet group at Iowa State University provides MetNetDB (MetNet 2008). This database contains information on networks from the metabolic and regulatory interactions in *Arabidopsis thaliana*. Database information is based on information from biologists. Interactions that are stored include transcription, translation, protein modification, assembly, allosteric regulation, and translocation from one subcellular compartment to another. The database also provides AraCyc-curated pathways and AGRIS-curated regulatory networks. Data is derived from a collection of other web databases. MetNetDB also provides a Curator tool that can be used to both query and modify the database.

Once the known biological data is collected, scientists then have the task of finding interesting genes within an organism. The use of tools to visualize statistical significance greatly speeds up this process. GeneVis (Baker et al. 2002) is a particle-based system that provides an environment for visually exploring genetic regulatory networks. It simulates genetic network behavior based on probabilistic occurrences of gene-protein interactions. Two different visualizations are provided: visualization of the movement of regulatory proteins and visualization of the relative concentration of those proteins. Different representational models are offered, including a protein interaction representation, a protein concentration representation, and a network structure representation. Protein interaction focuses on the activities of individual proteins. Protein concentration displays the relative spread and concentration of proteins. Finally, network structure depicts the genetic network dependencies present in the simulation. GeneVis offers some interactive components. These include animations between representations and three types of viewing lenses (fuzzy lens, base-pair lens, and ring lens). The simulation starts at the cell level. A large circle in the middle of the visualization represents a chromosome. Various genes (represented by small spheres) are then plotted on the chromosome. Small fuzzy dots around the genes and chromosome represent proteins. Color is used to distinguish the type of protein each dot represents. With the dynamic nature of the tool, one can see not only when a protein binds with one gene, but also its resulting effect in the environment. Users can then switch between three lenses to see protein interaction, concentration, or representation for one protein type.

exploRase (MetNet 2008) is statistical-based gene visualization tool written in R and available from Iowa State University. The purpose of this tool is to allow the user to explore and analyze multivariate Systems biology data. It handles transcriptomic, metabolomic, and proteomic data. A graphical user interface is provided on top of the script-based R language. This visualization tool allows users to load biological data and analyze 'omics data from the context of metabolic and regulatory networks. Three files are necessary to use exploRase, but once loaded the files are save as a set. The user is also able to save calculated statistics. exploRase uses various chart and plot representations (dotplot, scattergrams, parallel coordinate plots, etc), which are configurable by the user. The system also supports coordinated multi-window display. A selection (brush) in one window links the action to the other windows. The user is also able to sort tables of metadata and link this information via color-coding to the display.

After a scientist has identified interesting genes, he/she must then move on to the step of discovering how the gene interacts in the organism. Pathways can be thought of the chain of chemical reactions that take place in a cell. By knowing how and what is affected, scientists can narrow the role of certain genes. Cytoscape (Cytoscape 2008) is an open source software platform for visualizing molecular interaction networks. It integrates interactions with gene expression profiles as well as other state data. Additional features can be added as plugins. The plugin functionalities can range from profiling analyses, layout direction, and file format support. The user of Cytoscape is able to customize visualization of his/her data in various formats (hyperbolic, node-link, etc). The user can also map attributes to different colors, line thickness, and border color. The visualization is laid out in a 2D plane and allows the user to choose a layout algorithm. Interactive tools include zoom/pan, overview, and marking. Filtering is made possible by selecting nodes and/or interactions based on data (threshold, p-value, gene expression level). Cytoscape can also find active sub networks and clusters. The user session can be saved for future work. This files contains network, attributes, desktop states, properties and visual styles of the tool. Finally, Cytoscape network visualization data can be exported to a static image in a variety of formats. The Cytoscape team highly encourages other teams to create plugins for this tool. FCModeler (MetNet 2008) is a Java plugin to Cytoscape. All the plugins are designed to visualize network data from the MetNet database. The goal of these plugins is to provide a modeling framework for biologists to explore hypotheses, analyze network structure, and visualize results of experiments for different types of omics data. The plugins feature a sub-graph creator, dot layout, and algorithms for finding cycle and paths.

3.2 Process

In order to understand what it takes to create a biological visualization, a thorough background in Information Visualization is needed, along with analysis of current visualizations. Finally user input from persons in the biological area is needed to ensure usefulness of the visualization to intended users and that the tool's interface fits with their mental model. Modern visualization systems need to take advantage of the abilities of human perception and cognition. Gearing characteristics of the UI to be easily identified facilitates the identification of data. For a more complete background of IV please see Chapter 2. By looking at various visualizations one can be inspired to create something new or to incorporate techniques. Understanding the UI is also critical in determining the structure and information design. Completing competitive analyses put focus on the capabilities that are expected and desired from a new tool. This analysis can also help to identify weaknesses or areas for further development.

After completing readings and looking at different tools, this author believes that a more optimal visualization can be achieved for biological pathway exploration. Current visualizations are focused primarily on one task/visualization method, and all are standard mouse/keyboard and monitor display. Only a few tools allowed for multiple coordinated windows. In addition, all tools are based on monitor display, and therefore work with a limited visual field. Biological visualizations should be data rich, but the focus should be on the data rather than how to control the tool. Unfortunately, this is not the case with current implementations.

To create this new tool, we needed to meet with and talk with the actual end users of such a system. To this end, we conducted interviews with four biologists. The biologists consisted of three graduate students and one post-doc from Iowa State University. These were informal interviews with the intention of learning what research and tasks each user performed and their level of comfort with technology and visualization tools. Almost all the interviewees used visualization tools, mainly exploRase, to help identify interesting genes. However, only a couple biologists were beginning to learn and use pathway visualizations. While the interviewees used some visualization tools, they did have some complaints and wants that the current tools did not have. After talking with them a number of interesting themes emerged.

3.2.1 Interview Findings

One of the first topics discussed was how the user utilized the tool and what they were looking for. Somewhat surprising for this author, the processes and actions were relatively similar across the users. Once the data was loaded, the scientist was concerned with the same basic operations. He/she was interested in identifying difference between genes, similarities, patterns, anomalies, outliers, and groups. More granular identification was also discussed, such as discovering the levels of certain attributes, and how much change had occurred. Only one user did time-based experiments, but the ability to detect exactly what change, and by how much was important for all.

A few users expressed the desire for more ease-of-use with visualization tools. The first situation identified was the actual download and installation of a program. One user said that at one point she had to install three different packages just to use one tool. All these packages were located on different sites, and each had its own versions and requirements. The actual installation process could also prove to be long: left over a few hours to finish. Some installations also require user input; meaning the user must be present at the time of install. Documentation, which is often left to the last minute, is very important, especially for complex tools. For example, in exploRase, the user must run a script to start the program and to alter it in other ways. One user interviewed said that her lab mates were always amazed at even the simple scripts that she was able to run. This amazement points out the need for tools to provide simple documentation, or better, to have the functions built into the product. Since most users of biological visualizations do not have extensive computer knowledge, it is not reasonable to expect them to be able to know and use complicated terminal-like commands. If they user must rely on documentation to learn operation commands, the documentation should be clearly laid out, use simple terms, and provide visuals of what the user must enter.

Another part of ease-of-use deals with the user interface (UI). One user reported her frustration with not being able to maximize the display area of the tool. Since the visualization should become the most important part of the tool, it should hold focal point and allow the user to see the data at the largest desirable size. The use of screen real estate then becomes a critical factor. Some programs try to overcome this limitation by allowing the user to use multiple windows that are not connected together, e.g. exploRase or even Adobe Illustrator. By doing so the user defines how large each window can be. There are downfalls to this customization however. Juggling multiple windows can become a cognitive strain. Only a few attributes can then be compared or remembered between windows. The user also has to navigate around the multiple control palettes. One biologist identified that she usually only had 2-3 different visualization windows open at a time. However, the ability to see different visualizations in the windows can lead to new discoveries and insight into the data. Another topic that was discusses was that the actions or functions that can be applied to the visualization should be easily identified. While icons can be ambiguous, they can help the user quickly access commands. While labels added to icons can help in identification, one should avoid long labels that cannot fit in the icon size. This factor does not mean, however, that the label should be large in size. Modern interactive systems should always allow access to the user of the various constraints. The ability to change the levels of attributes should be a default option. Also known as dynamic filtering or querying, the ability to manipulate the visualization helps the user test hypothesis and identify areas of interest.

During the interviews, there seemed to be two quite different types of users. The novice visualization user had little to no technical background. They had only just begun using visualization systems and relied on the expertise of other users to learn tools and scripts needed to execute functions. Expert users on the other hand, had significant technical skills, able to write their own programs and execute scripts on current tools.

The contrast of these two types of users brought up an interesting dilemma. One expert user stated that he did not use visualizations in part because of trust issues. Trust figures in a few different areas of visualization. Current visualization tools do not expose the operations they perform on the data, or the formulas on which they base graph drawing or data manipulations. It also takes time to establish trust with a tool. One user stated that she performed the statistical operations by hand and would compare the results to what the tool provided. Only when the more accurate the comparisons were, and the more often they were accurate, would trust increase. Finally, visualizations need to expose how trustworthy the data is that it is relating. In much of biology, certain functions are thought to exist and are based on current research. However, even though a function is presented, it might not be entirely accurate.

Besides trust, a common request of future visualizations dealt with collaboration. Most of the biologists expressed a desire to take the data out of the visualization for presentation purposes. This presentation could be for publication or to simply show colleagues during meetings. Often times they would discover something interesting in the data and be unable to store it for sharing. Some would resort to screen captures, but doing so does not preserve the data and conditions that went into making that particular visual. Another user expressed the wish to take the visual and then adapt it and simplify it for teaching or publications purposes. She was seeking to take the visualization and construct a knowledge map based on what her discoveries. Another biologist stated that along with being able to take the data out, she also wanted to store information in the tool. Often times the user would perform complicated actions and have to record the steps that she did to obtain the visual. These were kept externally to the program. Any sights that were gained also had to be noted externally and kept track of. Interestingly, after researching biological visualizations, other papers pointed out that being able to collaborate using one visualization could be very useful. Having two sets of eyes and minds can divide up tasks and find points of interest. While none of the interviewees mentioned such ability, this author believes that it could prove extremely useful.

3.2.2 Prototype

After conducting interviews, the next phase of development was to choose the medium and visual look-and-feel of the visualization. The traditional ways to implement visualizations are to use the keyboard/mouse and Windows, Icons, Menus, Pointer (WIMP) paradigm. While these methods have yielded some unique and useful tools, this author believes that a different device could prove to be more intuitive. We propose to implement our new visualization using a multi-touch table interface. Not only is the screen real estate increased (touch tables range in size from 30 to 42 inches in diagonal), but touch tables also offer a very intuitive interaction style. The WIMP paradigm can be used with discretion, but it is not expected as compared to monitor-based systems. In addition, multi-touch tables enable multiple users to collaborate at the same time on the same screen. Data on the table can be easily manipulated, updated and shared. While touch tables are more expensive than an ordinary computer, they are less expensive than 3D systems such as CAVE's or projection systems. Building a touch table can also be achieved. Open source toolkits are available for programmers to develop applications. Current touch tables systems include Microsoft's Surface (Surface 2008), Mitsubishi's Diamond (Diamond 2008), and Pompeu Fabra University's reacTable (Jordá et al. 2007). Another development is Wirmachenbunt's Prototouch, a touch table that is also sensitive to touch pressure (Wirmachenbunt 2008). While all these tables use multi-touch technology, each one has unique features. The Diamond can recognize an individual's touch based on a unique receiver (in their case, the chair the user sits in). The reacTable was built as a synthetic musical instrument and allows the user to manipulate conditions based on interaction with physical objects on the table. Touch tables have even found their way into artistic uses. For example, Jonathon Harris (2008) in his latest visualization, *I want you to want me*, commissioned for the New York Museum of Modern Art, utilizes a touch table to allow viewers to interact with information gathered from online dating companies. At the touch of the screen the viewer is able to create dynamic queries and move objects around.

Choosing such a system involves creating not only the visual layout, but also a set of gestures to control the visualization. Fortunately with the introduction of the Apple iPhone, more people are becoming aware of gesture recognition technology and commonly used gestures. Gesture and touch-based displays can achieve all the interactions that a traditional keyboard/mouse input can support and more. Gestures can encompass one-hand, two-hand, and object manipulation. The visualization would need a variety of gestures, from menu opening/closing to changing the viewing angle, and window configuration. Only a few studies have been conducted on gestures. While there is no concrete set of universal gestures, many tools are recognizing a few basic staples. Often the same gesture can be used with either one or two hands (see Table 6). Gestures have also been proven easy to learn and use, requiring little training (Wu et al. 2006).

Table 6. Touch-table Gestures

One hand one finger	One hand multi-finger	Two handed	Object
Point, tap, double tap, select, drag, lasso, scroll, rotate	"piano-chord" select, rotate, minimize, maximize, swipe, scroll, reveal	Rotate, Minimize, maximize, gather /pile	Change attributes, rotate, turn on/off

To adapt common gestures to our proposed tool, we first had to define the basic tasks the user would perform. After defining these common tasks, we then were able to create a set of gestures (see Figure 48 for an example). Gesture reuse in the form of gesture combinations would enable our users to minimize the amount of training needed. The ability to combine gestures would also make the interface even more powerful (such as shrinking an element while rotating). Not only were we concerned with the physical movement of the hand, but also the appropriate feed back to give to the user. Some other touch tables have shown small spots where fingers rest, other show rippling waves starting at the point of contact. Other feedback cues could be given with an animation, sound, or color change.

Figure 48. Sample gesture

Besides the gestures, we also needed to create a dynamic interface. After comparing what other visualization provided, our tool would have to implement the common functionalities, as well as new methods we had gotten from the user interviews. The new interface would utilize dynamic menus that open and close based on user touch. The main idea was to create an interface where the controls were visually out of the way: letting the data become the focal point. For an example of an initial prototype see Figure 49.

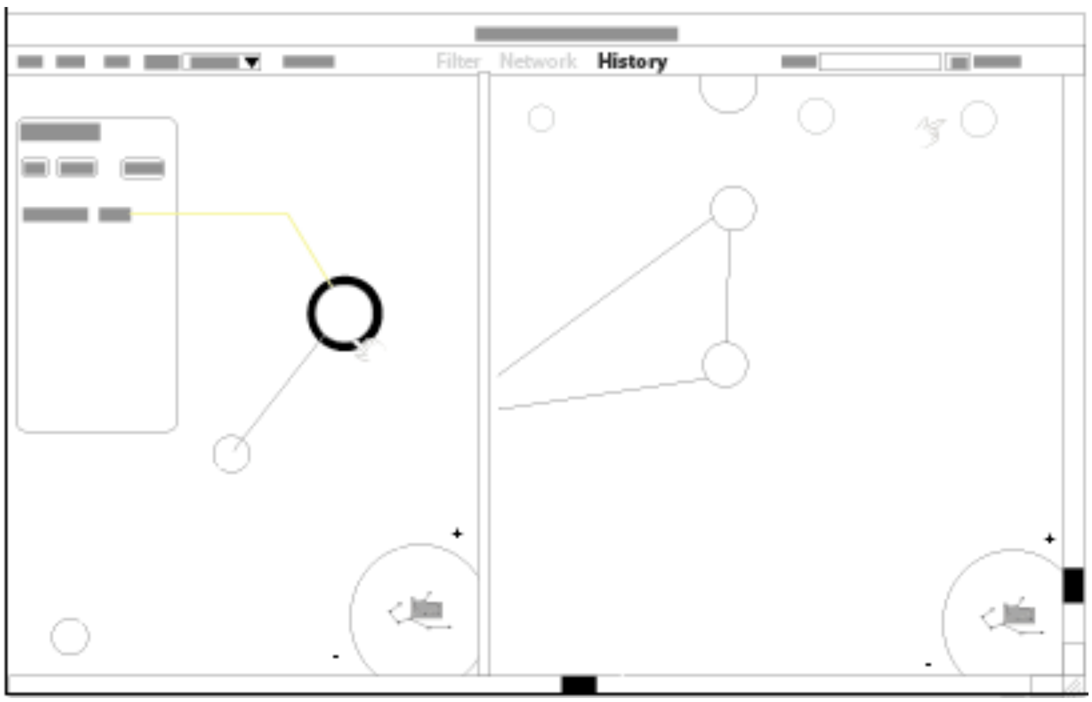

Figure 49. Early Wireframe

CHAPTER 4. RESULTS

However beautiful the strategy, you should occasionally look at the results.
Winston Churchill

BASED ON THE information gathered and studied, we propose a new visualization system for biological network exploration. Current implementations for biology are focused and limited at the display of networks in a single graph type; node-link. These tools do not take advantage of developing technology for touch-based displays or new interaction techniques. Often the user interface (UI) for current tools does not allow the user to maximize the actual visualization area, and the controls dominate the visual plane. Finally, current tools do not seem to take into account the human perceptual system; often overloading visualizations with characteristics that complicate the visualization and do not allow the user to make pre-attentive judgments.

The BioN (**Bio**logical **N**etwork) interface is designed for multi-touch display and offers a complementary monitor-based implementation. This visualization tool is designed to help scientists explore biological networks in the hope of inspiring new insight, discovery and comparison of gene function. BioN will address and attempt to solve some of the shortcomings of current visualization tools.

4.1 Device

BioN is designed for two different types of display devices. First and foremost, BioN will be developed for a multi-touch table. We believe that a touch-based visualization will afford scientists a novel and more easy way of interacting with their data. Since most scientists do not come from a strong technical background, the more the UI can be simplified and be made visual the more accessible it will become to users. However, the tool must also provide ways for the more technically inclined and advanced users to manipulate the data. Touch-based displays also tap into a user's kinetic memory, i.e. muscles can remember positions and actions without conscious thought after practice (much the same way a ballet dancer can automatically assume different positions without correction). Being able to move and rearrange data also allows users to exploit our perceptual system for locating objects by place. The larger screen area of the touch table also takes advantage of increased area for displaying data. Finally, touch devices allow simultaneous collaboration between two or more users. The table can read multiple touch inputs and even determine, by some external cue, who is touching the table. Unfortunately, touch devices do face some hurdles. Although the devices are more expensive than ordinary desktop displays, they are not as expensive as more virtual displays, such as a CAVE. Secondly, while some gestures for touch are intuitive for the user, more complex combinations would have to be taught and memorized. To date, there is no standard library of gestures, making learning to operate such a device more time-consuming. Finally, implementing an application to be touch-sensitive is not trivial. Depending on how the table can be used, a programmer might have to rely on device-specific drivers and socket level communication.

Since most institutions do not have access to a touch table, a corresponding monitor-based tool will be created. While lacking some of the more dynamic and innovative interactive features, the monitor-based system will contain the core functionalities of BioN. Monitor-based solutions have the advantage of widespread use, and most users are familiar with the setup and use of the device. Some measure of interaction is available with this technology, but the interaction can be disjointed, i.e. a user must make use of external devices to control the actions that take place on the monitor. Monitor-based applications also have very limited screen real estate. Most monitors can range between 13 – 17 inches in diagonal. Depending on a computer's resolution, the amount of material that can be displayed on such a surface is limited. Finally, only limited collaboration is possible with monitors. Users have to sit closely side-by-side to see the data. Projection devices are available, but usually only one person can be in charge of what data is being shown. Many applications today are moving to online collaboration; which lets more than one user contribute on a single dataset. However, the different users cannot control a single device, have to be using separate machines, and do not know who is editing what at a specific time.

4.1.1 Touch Table

While touch tables can be implemented in a few different formats, they share many of the same features. A touch table is usually composed of 4 main components. First, a large acrylic screen is the main area of interaction. This screen is usually able to process multiple inputs from users or objects. The images that are shown on the table are broadcasted there by a projection system (see Figure 50). Its ability to read touch-based input is from infrared sensors that are placed underneath the screen and from cameras placed above the table (see Figure 51). When an object touches the panel, the light reflects and is picked up by infrared cameras. Finally, a touch table uses some basic desktop computer hardware to function: memory and a CPU. Touch tables can make use of wireless communication with objects on the screen with WiFi and Bluetooth. Iowa State University is developing an API named Sparsh (a Sanskrit word for touch) for universal programmability with touch devices. A default set of touch recognition and gestures are provided, allowing the programmer to combine and define more complicated tasks from the provided basic functionality. The API should be able to be accessed in almost any programming language that is desired.

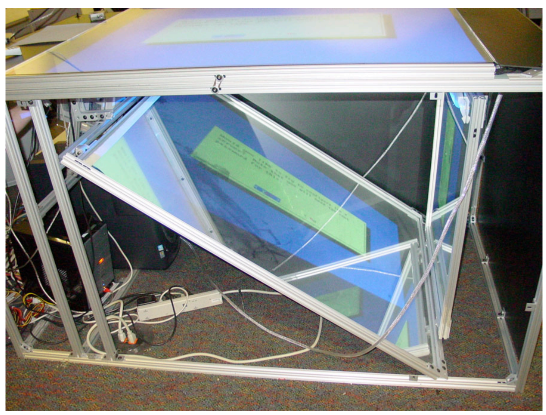

Figure 50. Touch Table

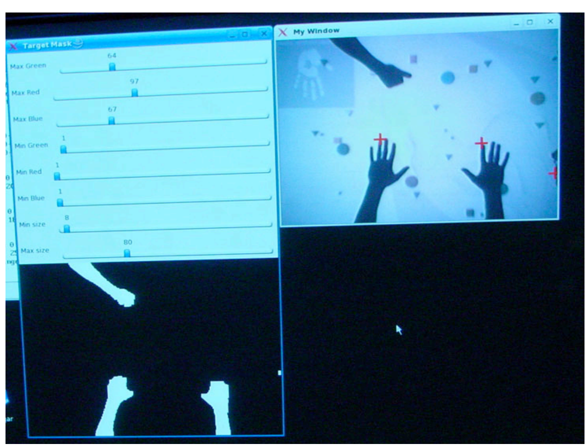

Figure 51. Overhead Camera view of multi-person using a touch table

4.1.2 Monitor

Current monitor-based systems use a variety of technology. The more common approach to date is the use of LCD (Liquid Crystal Display) technology. LCD's are flat panel devices that are made up of color or monochrome pixels. This panel is placed in front of a light source or reflector. More advanced monitors are being built with LED (Light-Emitting Diodes). LED's are able to convert electric energy directly into light of a single color. Because most of the energy is converted to energy in the visible spectrum not much energy is left to create residual heat. Both these technologies use the standard light-based color system and can display all colors within the RGB spectrum. LED are preferable to LCD due to LED's high-energy efficiency, which translates to power-saving and environmentally friendly aspects. Monitor-based programs are well understood, and the WIMP paradigm has been around since the 1980's. A variety of interaction devices are available for monitors, including keyboard, mouse, and joystick input.

4.2 BioN

4.2.1 User Interface

The user interface for BioN will be laid out conceptually on three distinct layers (see Figure 52) each with a specific range of interaction. The lowest layer will be a static background. Holding just the background color and network name, all other layers will be laid upon this layer. For the monitor-based implementation, the layer will also contain the main menu of the application. Next, a semi-static layer will contain the data. The data will be able to be manipulated by the user using gestures or mouse interaction. Finally, the tools and controls will be on a dynamic layer. The menus will be able to be moved, resized, closed, or collapsed by the user. They are located on the upper layer in order that they will float atop the data and thus not be obscured by the data shown.

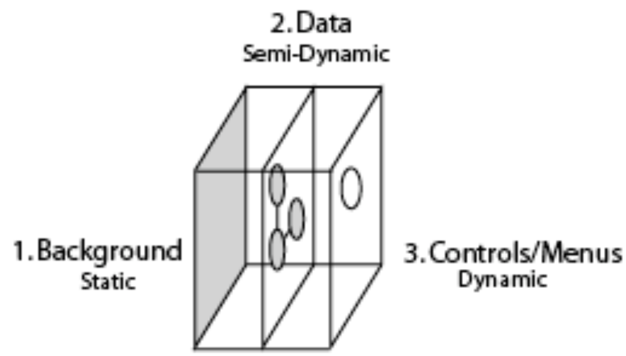

Figure 52. Touch Table conceptual UI layers

While the basic functionalities for both the touch-based and monitor-based systems would be almost similar, the user interface (UI) would have some fundamental differences. The UI for monitor-based implementation will have more limited interaction. For example, the user will be limited to a dual-window representation. This limitation is imposed because BioN will be contained in one main window. All controls and tools will have to fit within a more limited area (see Figure 53). As such, the placement of the menus will be in a static location, but the user will be able to open or collapse them. In addition, the monitor-based application will only be able to use a direct-type of select; that is clicking an element or holding and dragging the mouse to define the selection area.

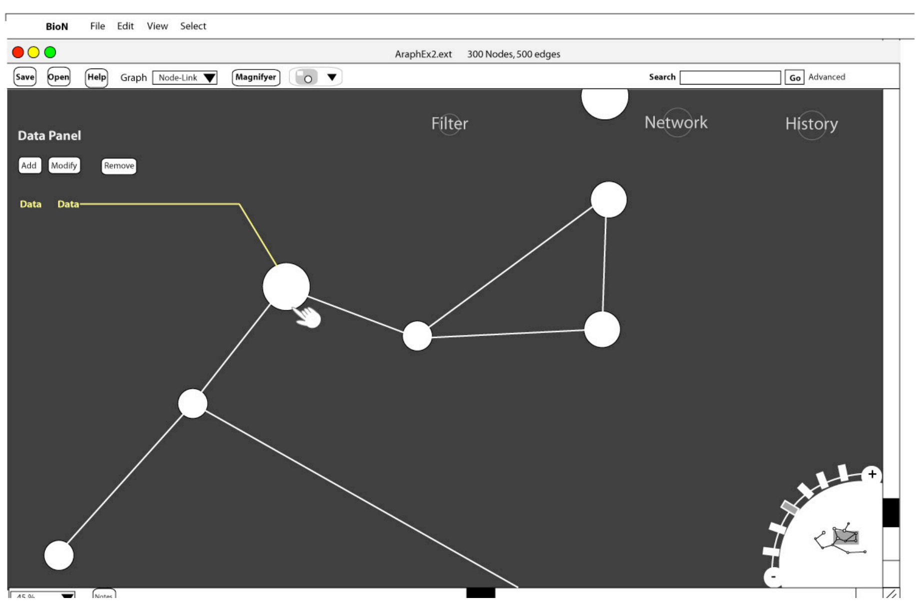

Figure 53. BioN Monitor application

The touch table based implementation will allow the user greater flexibility for menu and toolbar placement, as well as more direct interaction with the data (see Figure 54). All menus will be able to be moved by the user, allowing the user to push unwanted menus away from data they are interested in. The larger screen real estate of the table enables the user to create as many sub-graphs as he/she wishes.

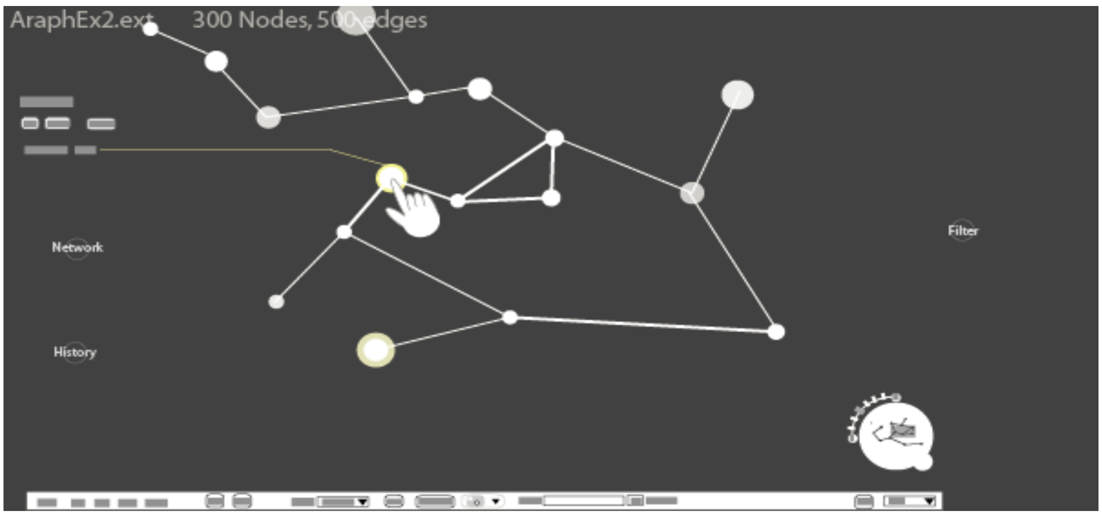

Figure 54. BioN touch table application

4.2.2 Capabilities

Supporting current biological visualization features, BioN also will incorporate new functionalities that existing tools do not have. These additions are concerned with network representation, use of screen real estate, and interactions. BioN has the ability to allow the user to utilize multiple window and graphical representations of networks. The new windows can be different networks or sub-graphs of the current network. Node-link, matrix, and arc diagrams will be available for network representations. Animations will be used to transition between one representation and the next. Further, the user will have control over all window placement and display size. The user interface (UI) of this tool tries as much as possible to fade into the background letting the data become the main focus. As such, all menu and tools have the ability to be placed dynamically at user specified regions, to be closed, or to be collapsed. If the user is collaborating with other individuals, the other users can switch the orientation of the menus by simply touching a corner of the menu and pulling it to the desired orientation. Besides the main menu, BioN will have controls for the network, preferences, history, notebook, camera, export, data panel, filter, search, HUD with zoom, and magnifier.

4.2.2.1 Network

The network panel (see Figure 55) is available to allow the user to change the encodings for the elements in the visualization. Unlike other programs, there will be a limited offering for one to configure (such as node color, node opacity, node size, edge color, and edge width). The reasoning behind this limitation is that only a small number of attributes can be selected pre-attentively or remembered in short-term or visual memory. In addition, allowing the user to completely configure the UI is usually not the best practice, as they do not have the experience or knowledge necessary to create very effective visuals. Some of the options for font choice and size will be able to be changed instead from the Preferences. The attributes that are available to change will also depend on the type of representation that the user chooses from. The Network panel, as well as all other control panels, is semi-transparent, allowing the user to see the effects of their choices in the background and to keep the viewer in the same context of their data.

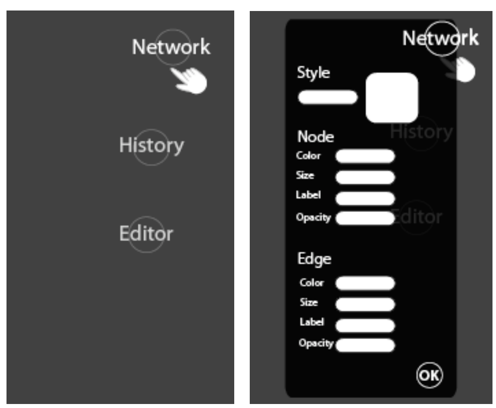

Figure 55. Network Encodings

4.2.2.2 Preferences

From the preferences panel, the user will be able to change underlying network configurations (e.g. layout algorithm, font, font size, etc). From this menu the user will be also able to access the underlying statistics used in the program for the graphs and representations. This ability will lay bare how the visuals are laid out and any dependencies or theoretical foundations.

4.2.2.3 History

The history tool allows the user to see what actions have been performed. One can undo selected actions or move to a specific point in the action sequence. The user will have the ability to preview the network at each point in the history. The default history will keep track of the past ten actions.

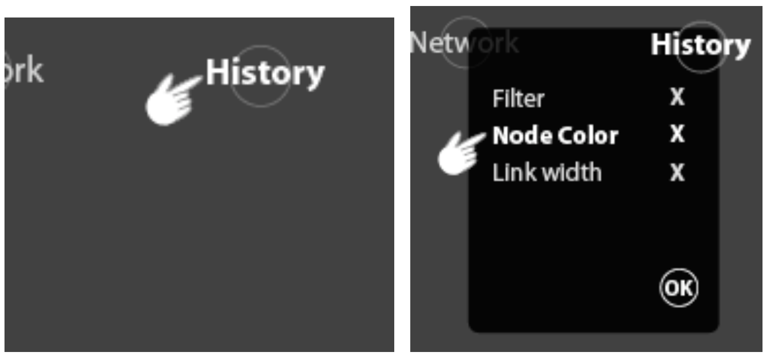

Figure 56. History

4.2.2.4 Notebook

From the interviews, we recognized that a simple text editor was missing within most biological visualizations. These tools are meant for complex tasks and data. As such, a researcher may need to take copious notes to log what actions they took to discover specific findings, or to teach others how to arrive at a visualization. The notebook feature in BioN tries to meet this need (see Figure 57). Using this tool, a user can enter notes about the dataset or actions they perform. A simple text editor will be available for simple spelling and grammar. The text log will be stored with a session and automatically available the next time the users opens the specific dataset. The user will also be able to print out their log.

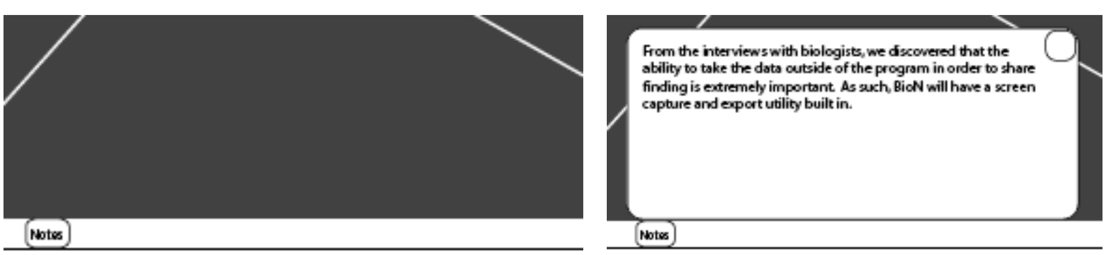

Figure 57. Notebook

4.2.2.5 Camera

Another tool that is lacking from current visualizations is simple screen capture technology. If the user wishes to take snap shot of the screen, they must rely on a separate tool. BioN will incorporate the function of a camera to allow the user to take a) a screen capture of the entire BioN screen, a selection of the BioN screen, or a time-elapsed screen shot (see Figure 58).

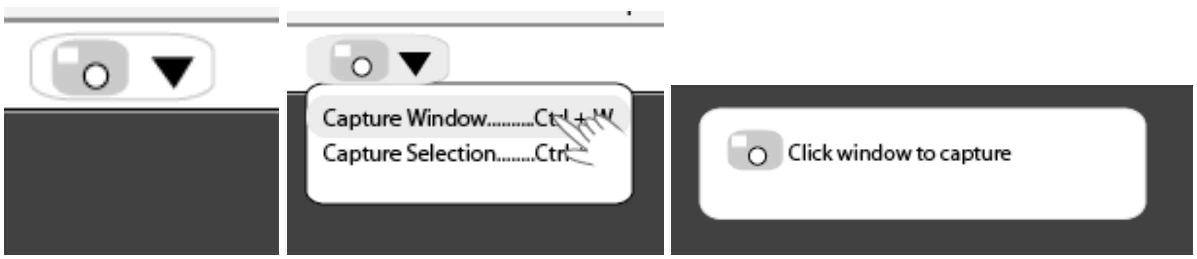

Figure 58. Camera

4.2.2.6 Export

After the discussions with the biologists, we learned that the ability to take the data outside of the program in order to share findings is extremely important. Since screen capture will only produce an image of the visualization without the underlying data, BioN will also have the functionality to export selected sub-graphs to HTML and Encapsulated PostScript (EPS). The HTML output will contain an image of the sub-graph and information on the data that is contained in the image. All this information will be contained in a single web page that the user then can upload to other web servers or share online (see Figure 59). All elements will display their label. The EPS graphic will allow the user to bring the image into other formatting applications, like Adobe Illustrator, to further manipulate the visual. One can then add additional markup and/or change the current setup of the image (such as changing colors or positions of elements).   This functionality will allow researchers to make their own graphics for presentation, publication, or knowledge maps.

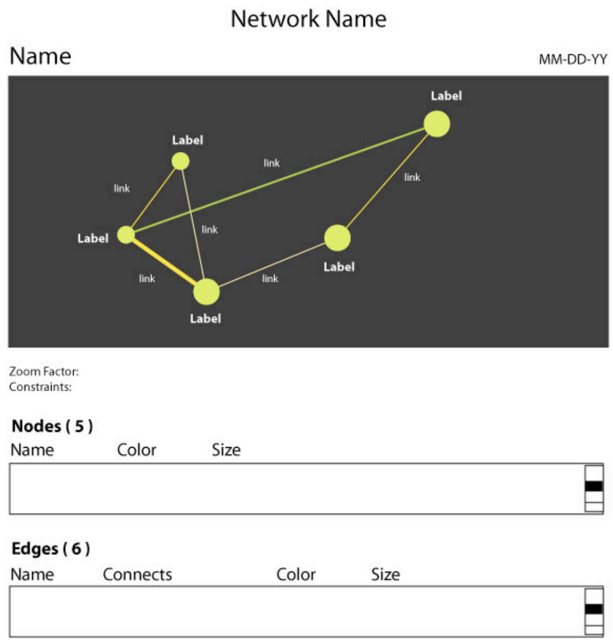

Figure 59. HTML Export

4.2.2.7 Data Panel

The data panel will contain the value of the element(s) selected: name and description (see Figure 60). The data in this panel will not depend on the type of element picked, i.e. the data can be a node, link, or other element. Showing the data when it is chosen rather than at all times, will de-clutter the screen and allow the user to only focus on elements of their interest. Like other software, the user will be allowed to add, modify, or delete attributes. For the monitor-based solution, the user will be able to rollover a selected element to have it appear in the data panel.

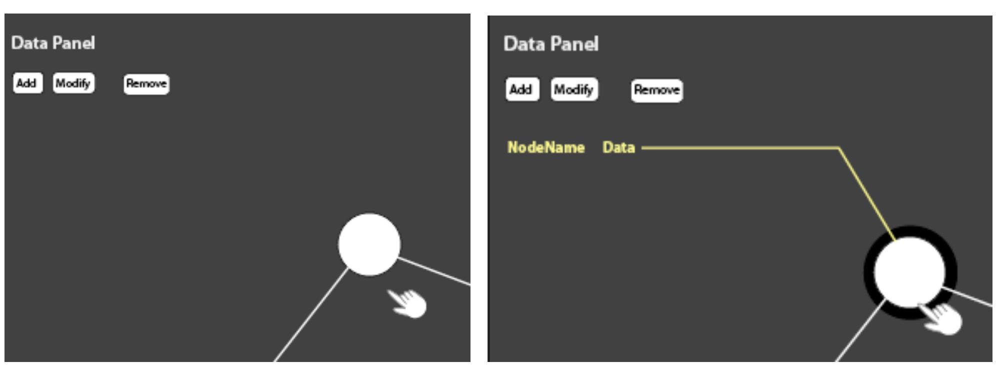

Figure 60. Data Panel

4.2.2.8 Filter

A filter based on user input will be available to narrow down elements shown on the screen. This filter will appear as a semi-transparent overlay atop the data (see Figure 61). The user will be able to choose from a variety of constraints to filter on, e.g. node type or edge strength. Once the user begins to specify constraints, the data underneath the filter screen will begin to change. Data that does not fit the constraints will fade away. By having this screen semi-transparent, the user will be able to visually see what certain filters are doing to the data as they input constraints. In addition, the filter will display the count of elements that remain after the selected criteria. This feature allows the user to have a sense of how many elements would be left after placing constraints on the network. A filter can be saved for later and will be available from the filter options. Once the filter has been created, the filter menu will change appearance in order to alert the user that a filter has been created and is in effect.

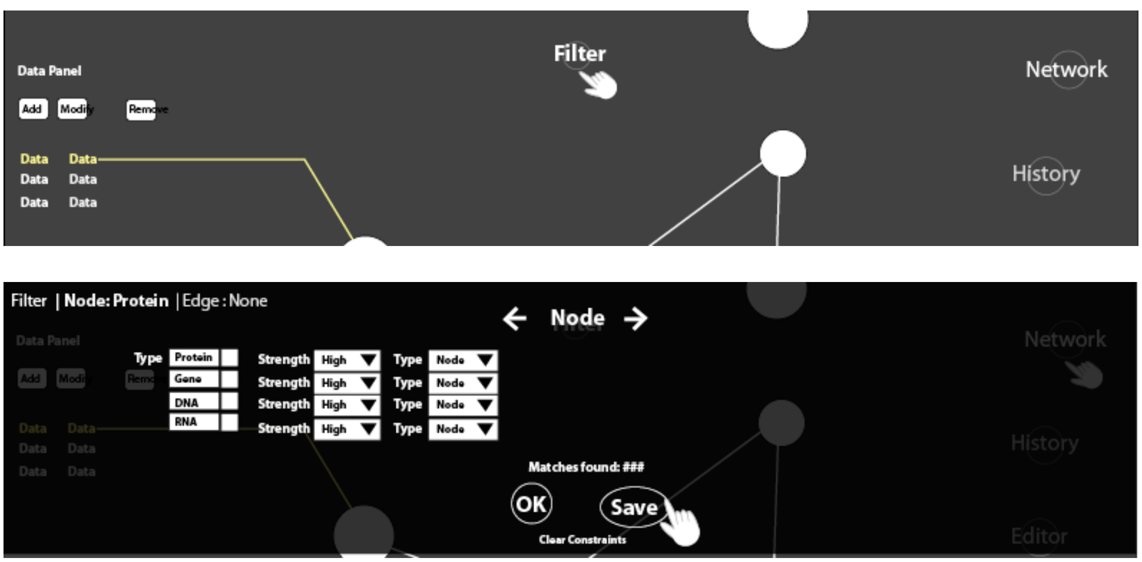

Figure 61. Filter

4.2.2.9 Search

The search function will allow the user to find a specific element within the network. When possible, hints and auto-completion will be available to cut down on spelling errors and typing time. If an element is found, it will be highlighted and the entire viewing area will move so that element will be in the center of the visual pane.

4.2.2.10 HUD and zoom controls

In very large networks it is crucial for the user to have a guide enabling them to know where they are. Zoom/pan controls all facilitate the investigation of data. The overview and pan/zoom controls for BioN will be contained in Heads-up Display (HUD) (see Figure 62). The HUD will show the current position of the user in relation to the network, by displaying the viewing area within a shaded rectangle, as well as providing a visual cue to the depth of zoom. The user will be able to move the viewing rectangle within the HUD and consequently move the viewing area.

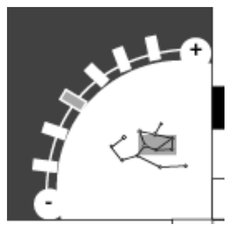

Figure 62. HUD and zoom controls

4.2.2.11 Magnifier

Because many networks can grow to thousands of nodes, the user will need a way to focus in on an area of interest without having to zoom in further. To provide this focus + context, BioN will provide a magnifier tool (see Figure 63). Acting as a type of fisheye distortion, all elements underneath it will be enlarged and labeled. Since labels will only be shown at very zoomed-in levels to avoid label occlusion and clutter, this tool will provide an excentric labeling technique.

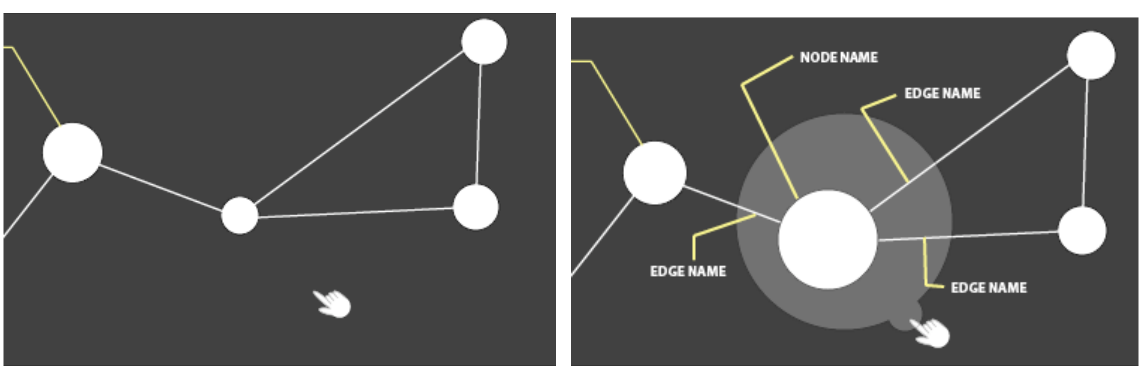

Figure 63. Magnifier tool

4.2.2.12 Multi-window, Multi-representation, and Multi-network

A very useful feature of BioN is the ability to see representations of the network in different windows and even different representations (see Figure 64). Matrix-based representations have been proven to be more effective than node-link diagrams for multiple tasks. However, node-link is more effective for path finding. Arc diagrams can also help identify areas of high concentration. Based on sequences, arc diagrams could prove very useful identifying common paths. Being able to utilize a variety of representations will allow the user to explore a network more effectively. Unfortunately due to a monitor's small screen real estate, BioN will be limited to a dual window representation for the monitor-based application. However, the touch table will be able to support as many windows as the user desires. Windows can be the entire network or sub-graphs of a network. To create a new window the user can either a) select all and right click and choose "Create new window", or use the option "Create new Window" under the Edit menu. To create a new window for a sub-graph, the user will select the desired elements and right-click or use the Edit menu and select "Create new window". To change the representation of the network, the user must make the desired window the active window and then use the "Graph Type" option on the main tool bar. BioN makes use of the brush/link technique so that any selection in one window will select the corresponding element in the other window(s).

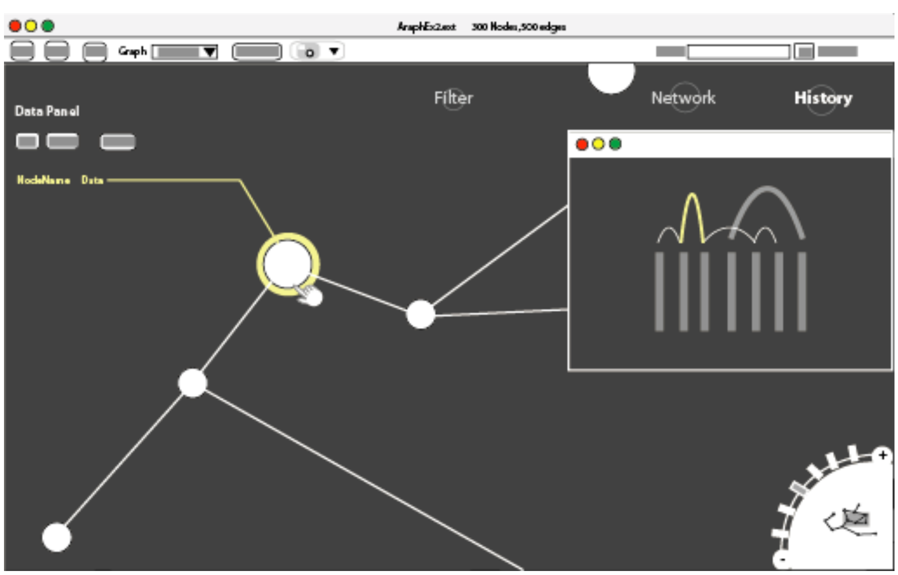

Figure 64. Multi-window, multi-representation

Users of BioN will also have the capability of having multiple networks represented on the same screen (see Figure 65). While a new window will be needed to visualize a different graph type, displaying multiple networks of the same representation does not have this limitation. The ability to have simultaneous networks has many advantages. For example, common nodes that are shared between networks do not have to have to be drawn twice.

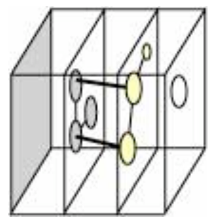

Figure 65. Multi-network conceptual model

The elements can incorporate visual encodings to show that they are shared, e.g. the opacity of a layer or node colors can identify a layer. Any element that is shared between networks could become more opaque, or have a different coloring (see Figure 66). A viewer can then easily determine what is shared between networks and what is not. A default setting will be used with allowances for limited user configuration. A filter could then be run to perform set functions such as intersections, unions, or complements.

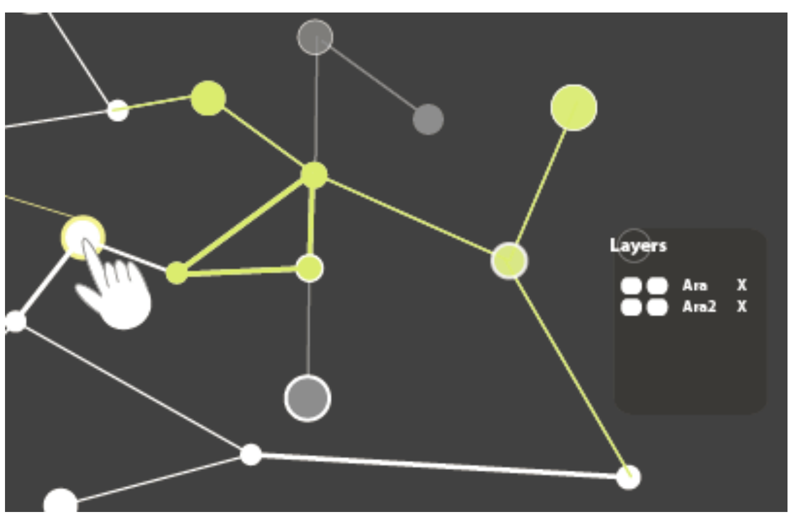

Figure 66. Multi-network

In order to keep track of the different networks, BioN will incorporate a Layer palette, much like Adobe Photoshop or Illustrator. This palette will only appear after the user has selected to use the multi-network option. Using the palette, a user can then turn off, select, or delete layers at will.

4.2.3 Interactions

With either the monitor-based or touch table based implementations a variety of interaction techniques are used. As stated prior, basic functionalities will be the same for both devices, but the capabilities and interactions available will be different. First, both representations will allow user-defined multi-window, multi-representation. The ability to dynamically search and filter the data will also be present. Finally, manipulating the network visual encodings will also be available. However, the touch-table implementation will afford the user more freedom of placement of windows, tools, and menus. One will also be able to create as many sub windows as desired. The interactions used to implement these features are based on the capabilities of the chosen system.

4.2.3.1 Monitor

Table 7. Monitor-based Interactions

Level	Action	Method	Feedback
Basic	Select	Click	Highlight/dim, update data panel
	Reveal	Rollover	Labels appear and element is highlighted by color change/drop shadow
	Open menu for an element	Click + hold	Highlight element and open menu
	Toggle Menu	Double Click	Show/Hide menu
Intermediate	Lasso	Click + drag	Highlight selected elements and area
	Move	Click + hold + drag	Highlight and move element
	Zoom	Click icon or scroll wheel	Animation
	Pan	Click + hold + drag	Cursor change and screen movement

Table 7. (continued)

Advanced	Aggregate	Lasso + right click + option	Highlight selection area, open menu, animate to meta-node
	Create subgraph	Lasso + right click + option	Highlight selection, open menu, create subgraph in new window

Monitor-based implementations rely on the use of external pointing devices to control interaction. Most commonly this is carried out with the use of a mouse and keyboard. BioN makes use of the five basic actions of the mouse: click, double click, rollover, on press, and on release. Using these basic inputs, the user is able to control the interface. A keyboard can also supply input. Keyboard input will allow the user to input data into the notebook as well as use short-cut keystrokes. For every action that the user takes, there must be feedback in some form for the user. Table 7 outlines possible actions, how the user accomplishes them, and the feedback that is given.

4.2.3.2 Touch

Touch-based implementations afford the user truly direct interaction with the visuals. One is able to touch the screen and use gestures to control interactions. A benefit of this type of interface is that a gesture can be done anywhere, and the user does not have to move to a toolbar or external pointing device to initiate the interaction. An internal keyboard can be used with the table to allow a user to input data. The interactions that take place on the table affect either global or local elements. Local interactions interact with specific elements of the visualization, e.g. a menu or node. Global interactions affect the entire display such as zoom or pan actions. In addition the interactions are either dynamic or static. Dynamic interactions change the appearance of the visual, while static gestures are used to highlight or select elements. See Table 7 for further gesture classification.

Table 8. Gesture Classification

	Static	Dynamic
Local	Select (direct, fuzzy, hard-select), open/close	Rotate, resize, magnify
Global	Select (direct, fuzzy, hard-select), open/close	Zoom, pan

The table implementation can produce the same effects of a pointing device. Instead of an external device, the user's fingers and hand acts as the input device. Basic touch recognition includes: tap, double tap, and hold. Using these basics the user is able to perform select, open/close, drag, zoom, pan, and aggregate gestures. Table 9 provides a small listing of possible actions for the touch based applications. For a list of possible gestures see Appendix A.

Table 9. Touch-based Interactions

Level	Action	Method	Feedback
Basic	Select	Touch	Highlight, update data panel
	Open	Touch + hold	Highlight and open menu
	Toggle Menu	Double tap	Show/Hide menu
	Reveal	Move Magnifier	Enlarge elements and show labels
Intermediate	Lasso	Gesture	Highlight selected nodes and area
	Pile	Gesture combination	Group elements
	Zoom	Gesture or tap icon	Animate zoom
	Pan	Gesture	Screen movement
	Move	Touch + hold and drag	Highlight + movement
Advanced	Aggregate	Gesture combination	Highlight selection area, animate to meta-node
	Create Sub-graph	Gesture combination	Highlight selection, open menu, create sub-graph in new window

CHAPTER 5 SUMMARY AND DISCUSSION

In my end is my beginning
T.S. Eliot

IN THIS PAPER we have covered many areas of Information visualization. Visualization is continuing to be recognized as vital to information dissemination and scientific discovery. Due to our limited short-term and working memory, visualizations act as external cognitive tools, allowing us to see and revisit large amounts of data. Understanding the various fields that visualizations cover can help the creator understand the basic concerns and representational methods for different visualization domains. Furthermore, looking at what other fields have done can inspire developments and advances in related fields.

History has shown that visualization has and continues to be advancing new ways of enabling users to present and explore data. From the early town maps to the latest in 3D Virtual Reality, scientists continue to explore ways to display data and impart information. Before modern computer systems, data visualization was limited to paper-based designs. This process was not only time-consuming, but also expensive to reproduce, and most graphics were printed in black and white. Today, visualization designers have a wide variety of resources to pull from. Sophisticated graphics rendering, new interaction techniques, and new devices allow designers the ability to create novel graphical tools. In addition, toolkits and frameworks are being continually developed and refined to shorten the time from idea to realization.

Designers of visualization tools need to keep in mind not only the cycle of viewing information visualization, but also the process cycle to create visualizations. Much like Shneiderman's visualization mantra, perhaps a new one should be developed for visualization creation: plan, create, evaluate. Each phase of the cycle is concerned with important issues. Visualizations will not be useful unless they take into account basic visual and cognitive principles of users. Overloading visualizations with unneeded dimensions and attribute encodings can render the data unintelligible and useless. Research is being done that proves there is not one universal solution to attribute display and encoding. Designers must keep in mind the type of data and task that a user will need to perform. Different designs have become status quo for certain types of data, e.g. trees for networks or timelines for temporal data. In addition, the UI needs to fit user expectations and facilitate user interactions with data. Somewhere a balance must be made between beautifully useless and functionally ugly. While functionality would ultimately win, a user might not enjoy the application enough to stick with a tool long enough to learn all its capabilities. A design that incorporates the theories of Gestalt and preattentive attributes is more likely to not only be more visually appealing, but also more usable. All aspects of the design must be put under the microscope. A tool is the culmination of all its parts, from the color combinations, the font size and choice, graphic representation, to element placement and terminology.

Finally, visualizations need to be continually tested and improved. While some visualizations are the test beds for innovative representations and interaction techniques, to be widely used and useful, a tool needs to be evaluated. Visualizations should be tested from the conception, during implementation, and after dissemination. Evaluations should reflect real tasks and goals of the end users of the system. Often collaborating with users can give designers insight into the mental model and preferences of the users they are designing for.

Biological research is an ever-growing field of science. The interest in the roles and functions of genes is a hot topic in biology and bioinformatics. Discovering new insights could lead to advances in areas such as disease prevention and curing genetic defects. However, the results of lab tests can result in huge datasets. The process to insight is time consuming. Specific genes of interest must be identified. Scientists then perform microarray analysis to try to determine gene function. When genes has a similar profile, pathway comparisons could help to uncover if more than one gene has a role in the function of a chemical reaction. Separate tools are available to hold information on genes and pathways, gene analysis, and pathway analysis. However, many current visualization tools do not allow users to view pathways in more than one representation. Many of these tools could also benefit from usability testing and UI improvements.

In order to understand the needs of our users, we conducted informal interviews with biologists. After meeting with them, we were able to find common themes throughout the discussions. All expressed an interest in a more usable and user-friendly interface. The application should be a simple download and install, i.e. not forcing the user to download multiple files and forcing them to sit through a long installation process. The ability to maximize the data viewing area was also important. In addition, almost all expressed the desire for added functionality to bring the data they found in the tool outside of the application for sharing with coworkers and for publications. While the users did not specifically mention collaboration, this ability will more than likely be important in the future. Having more than one pair of eyes for large datasets can enable the users to encounter more insights and discoveries. Finally, trust is a hurdle that all visualization tools must face. Users have to trust that what they are seeing in the visualization is what is really reflected in the data. To this end, the methods to display the data and any operations that are performed on it should be made transparent to the user.

At present, BioN is a theoretical UI. Much work needs to be done to implement the tool in its entirety. To ensure that the UI is on the correct path, BioN needs to be brought back to biologists and discussed in its wireframe state. Once discussions have brought out further concerns and insights, BioN will be ready for an iterative implementation cycle. The UI would then be continually brought back to the end users and experts in the UI field for evaluation. Finally, once complete, BioN would undergo usability tests with seeded datasets. The value of using known datasets will show if BioN users can reach insights that have already been discovered. Usability testing will also highlight any flaws or inconsistencies with performing necessary tasks.

Construction of such a dynamic interface faces a variety of challenges. To start, no solid API is available for touch-based implementations. While an Iowa State University team is attempting to create such architecture, many of the complex gestures will need to be defined at the application level. This UI will be a complex undertaking, requiring multiple programmers. A fully fleshed-out requirement and specification document will be needed to ensure that BioN is implemented correctly. New and intuitive icons will be needed, as well as visual ways to distinguish between the data and UI controls. The incorporation of multiple representations will also need further specification. While animation has been shown effective for transformation between graph types, animation methods will need to be created to transition between the three options available in BioN. The correlation between attribute encodings between representations will have to undergo testing. All these requirements translate to the need for extensive user testing throughout the lifecycle of development. Even once BioN reaches completion, users of the tool will need training for this novel interface. While we feel the gestures for controlling the interface will be simple to understand and learn, this will need to be proven. Even with all these challenges, BioN would offer scientists a wonderful new visualization tool for exploring, comparing, and visualizing biological networks.

APPENDIX

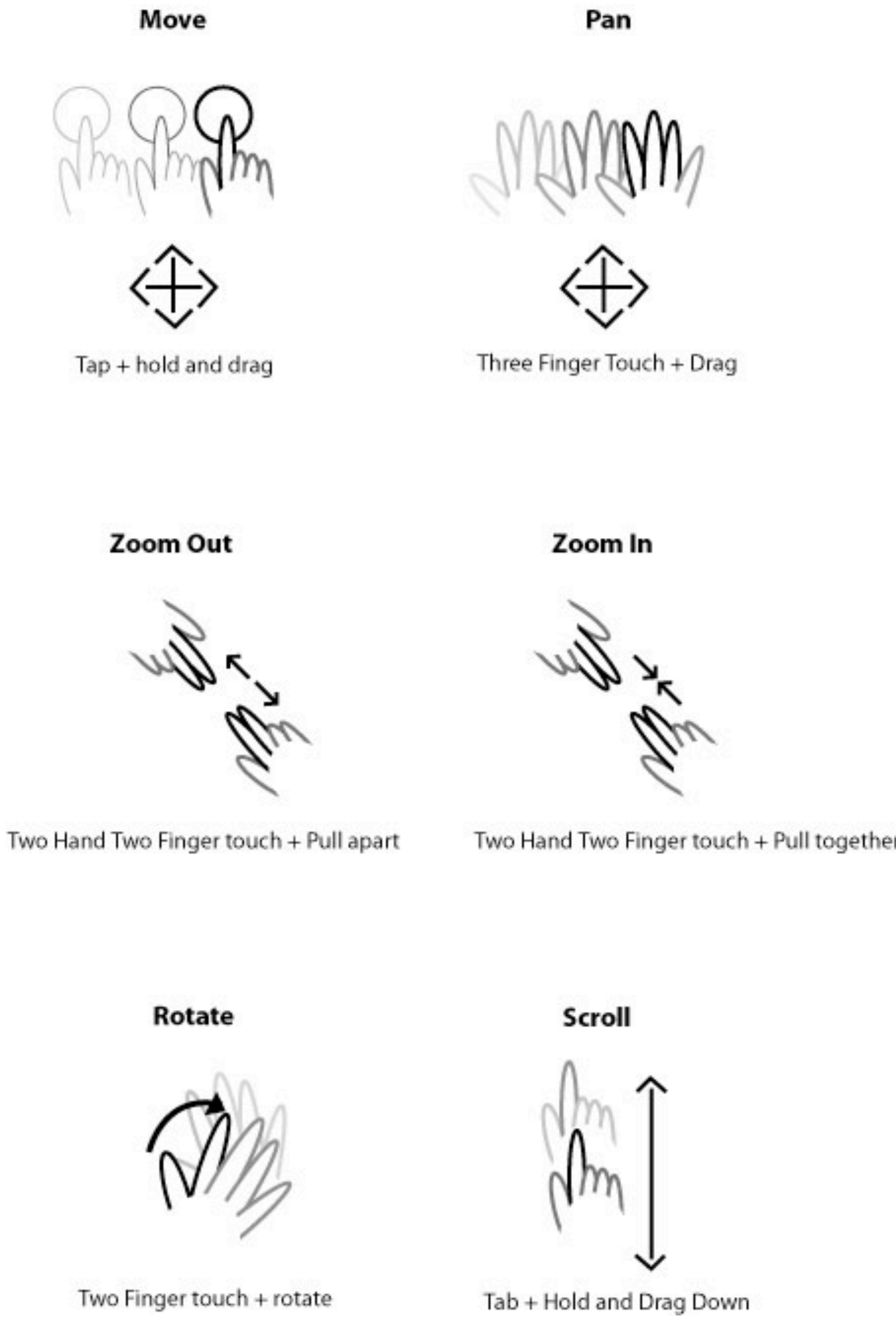

Figure 67. Basic Gestures

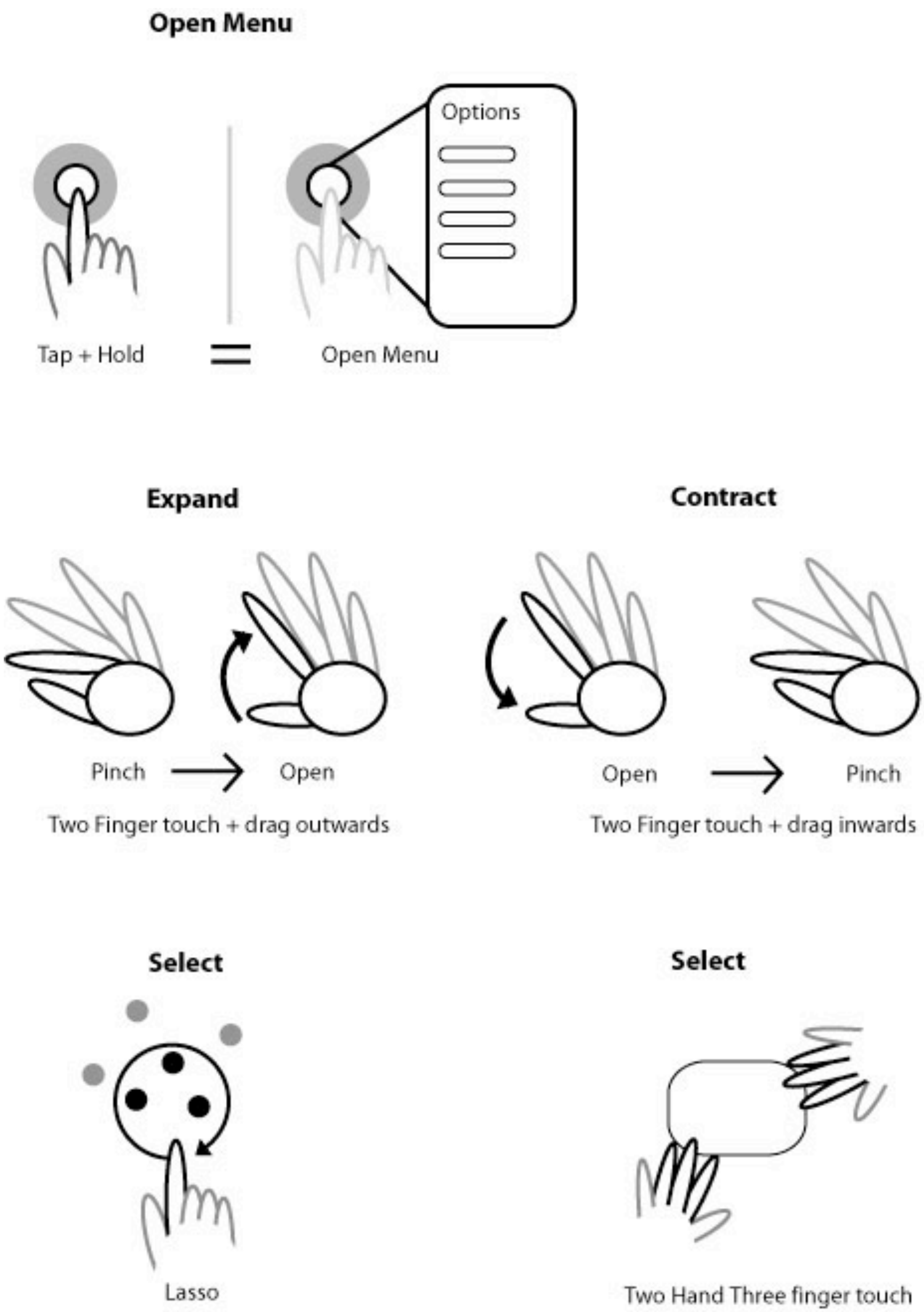

Figure 68. Intermediate Gestures

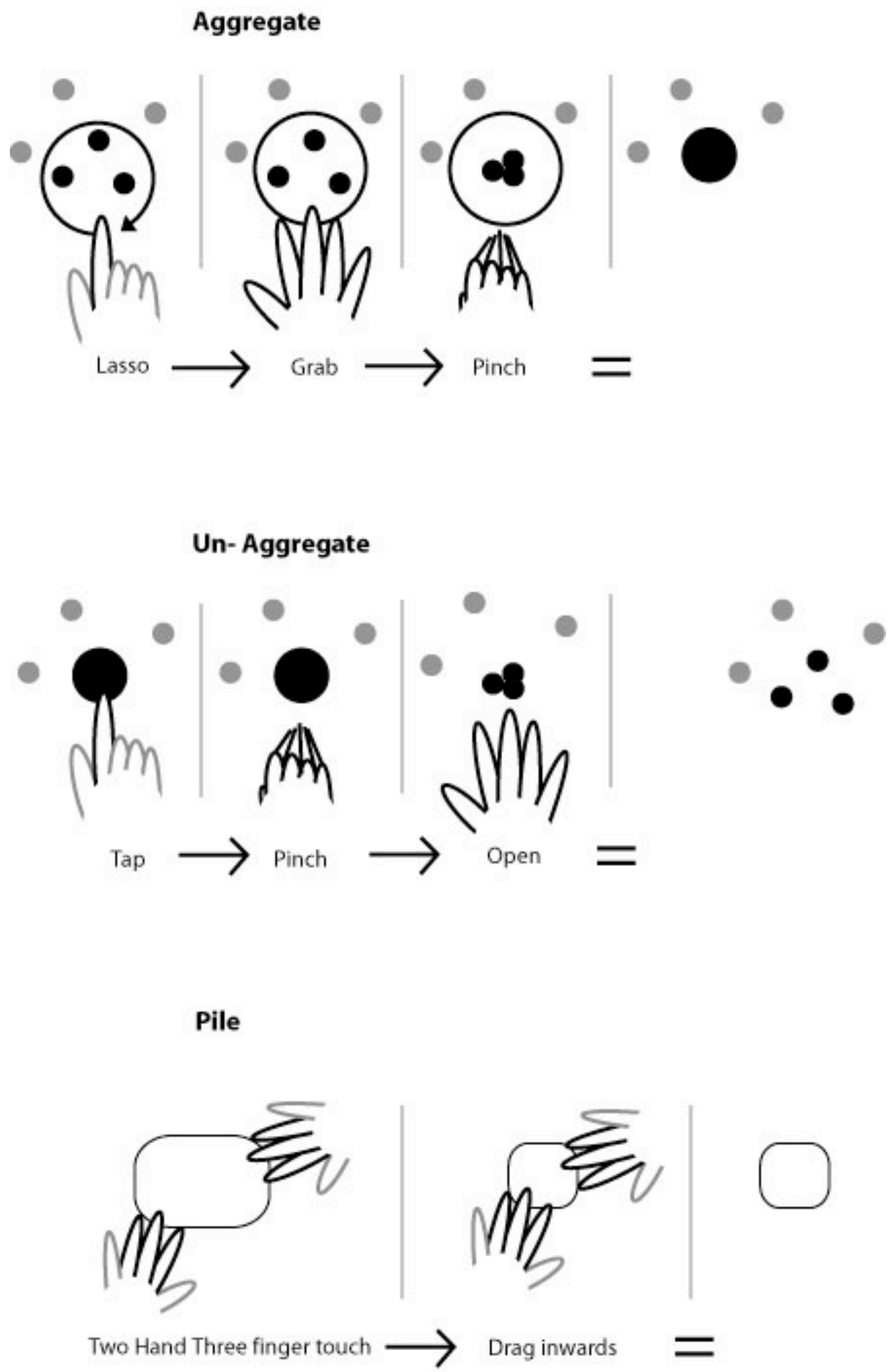

Figure 69. Application Specific Gestures

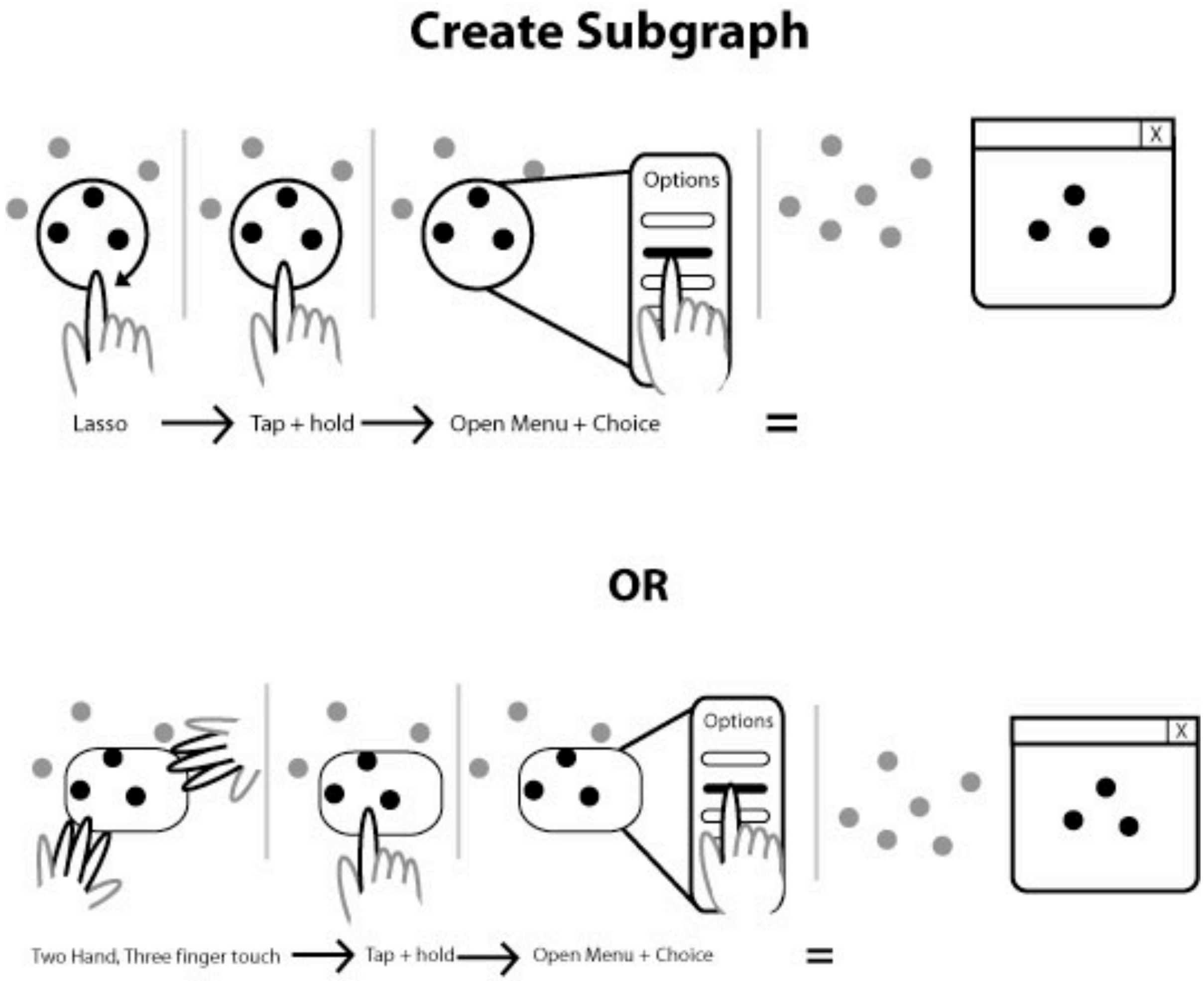

Figure 70. Subgraph Gestures

BIBLIOGRAPHY

Aigner, W., S. Miksch, W. Muller, H. Schumann, and C. Tominski, (2007). Visualizing Time-Oriented Data – A Systematic View. *ELSEVIER*. March 2007.

Baker, C.A., M.S.T Carpendale, P. Prusinkiewicz, and M.G. Surette (2002). GeneVis: Visualization Tools for Genetic Regulatory Network Dynamics. *IEEE Visualization* 2002 Oct. 27 – Nov. 1, 2002, Boston, MA, USA

Bederson, B. B. and A. Boltman (2005). Does Animation Help Users Build Mental Maps of Spatial Information? *The Craft of Information Visualization Readings and Reflections.* Morgan Kaufmann

Bloch, M., L. Byron, S. Carter, and A. Cox (2008). The Ebb and Flow of Movies: Box Office Receipts 1986 – 2007. New York Times. http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html Accessed March 31, 2008

Brewer, C. and M. Harrower (2008). ColorBrewer. http://www.personal.psu.edu/cab38/ColorBrewer/ColorBrewer_intro.html Accessed March 4, 2008.

Cawthon N. and A. Vande Moere (2007). The Effect of Aesthetic on the Usability of Data Visualization. *IEEE International Conference on Information Visualization (IV'07*), IEEE, Zurich, Switzerland, pp. 637-648

Chen, C. (2004). *Information Visualization: Beyond the Horizon. 2nd Edition.* Springer.

Cheng, P. C-H., S. Wood and R. Cox (2007). Dimensions of attentional processing. *Attention Management in Ubiquitous Computing Environments (AMUCE 2007), workshop of UBICOMP 2007,* Innsbruck, Austria

Cleveland, W. S. and R. McGill (1984). Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association,* Vol. 79, No. 387. pp. 531-554

Color Consultant Pro (2008). Code Line Communications. http://www.code-line.com/software/colorconsultantpro.html Accessed March 4, 2008.

Cytoscape (2008). http://cytoscape.org/index.php Accessed March 10, 2008

Diamond (2008). Mitsubishi Electric Research Laboratories. ©2008. http://www.merl.com/projects/DiamondTouch/ Accessed March 9, 2008.

Dittus, M. (2006). IRC Arcs. http://mardoen.textdriven.com/irc_arcs/ Accessed March 31, 2008

Deller, M., A. Ebert, M. Bender, S. Agne, and H. Barthel (2007). Preattentive Visualization of Information Relevance. from *HCM 2007*. Sept. 28, 2007, Augsburg, Bavaria, Germany. Copyright 2007 ACM.

Donath, J., K. Karahalios and F. Viegas, (1999). Visualizing Conversation. in *Proceedings from the 32nd Hawaii International Conference on System Sciences*.

Dougherty, B., and A. Wade (2008). VisCheck. http://www.vischeck.com/ Accessed March 4, 2008.

Emerson, J. (2008). Visualizing Information for Advocacy. http://backspace.com Accessed February 18th, 2007.

Fekete, J-D. (2004). The InfoVis Toolkit. in *Proceedings of the 10th IEEE Symposium on Information Visualization (InfoVis'04*), IEEE Press, 2004, pp. 167-174.

Fekete, J-D. and C. Plaisant (1999). Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. in *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*. May 15-20, 1999, ACM, New York, 512-519.

Fekete, J-D. and C. Plaisant (2005). Interactive Information Visualization of a Million Items. *The Craft of Information Visualization Readings and Reflections.* Morgan Kaufmann.

Foley, J. (2006). Just Another Pretty Visualizaton? The What Where, When, Why, and How of Evaluating Visualizations. *Presentation at The Symposium on the Future of Visualization 2006.* www.viscenter.uncc.edu/symposium06/OnlineContent/Slides%5CFoleySlides.pdf Accessed April 4, 2007.

Friedman, J. H., W. Stuetzle (2002). John W Tukey's Work on Interactive Graphics. *The Annals of Statistics*. Vol. 30. No. 6, 1629-1639

Friendly, M. (2006). A Brief History of Data Visualization. *Handbook of Computational Statistics: Data Visualization.* Springer-Verlag.

Fry, B. (2008). Isometric Blocks. http://benfry.com/isometricblocks/ Accessed March 31, 2008

Fry, B. and C. Reas (2008). Processing. http://www.processing.org Accessed February 15, 2008.

Gansner, E. R., Y. Koren, and S. North (1999). Topological Fisheye Views for Visualizing Large Graphs. AT&T Labs. *IEEE Transactions on Visualization and Computer Graphics*.

Ghoniem, M., J-D. Fekete and P. Castagliola (2004). A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. From *the IEEE Symposium on Information Visualization*. Oct. 10-12, 2004.

Harris, J. (2004). Word Count. Copyright 2004. http://www.wordcount.org Accessed January 15, 2008.

Harris, J. (2007). The Whale Hunt. Copyright 2007. http://www.thewhalehunt.org Accessed March 4, 2008.

Harris, J, and S. Kamvar (2008). I want you to want me. Copyright 2008. http://www.iwantyoutowantme.org Accessed March 4, 2008.

Harrison, C. (2005). Visualizing the Royal Society Archive. http://www.chrisharrison.net/projects/royalsociety/ Accessed March 31, 2008

Healey, C. G (2007). Perception in Visualization. http://www.csc.ncsu.edu/faculty/healey/PP/index.html Accessed September 14, 2007.

Heer, J., S.K. Card and J.A. Landay (2005). prefuse: A Toolkit for Interactive Information Visualization. from *CHI 2005,* April 2-7, 2005. Copyright ACM 2005

Henry, N., J-D. Fekete and M.J. McGuffin (2007). NodeTrix: Hybrid Representation for Analyzing Social Networks. *Inria Futurs*. June 21, 2007.

Jordá, S, G. Geiger, M. Alonso, and M Kaltenbrunner (2007). The reacTable: Exploring the

Synergy between Live Music Performance and Tabletop Tangible Interfaces. *Proceedings of the first international conference on "Tangible and Embedded Interaction" (TEI07).* Baton Rouge, Louisiana.

Joshi, A., and P, Rheingan, (2005). Illustration-inspired techniques for visualizing time-varying data. in *Proceedings of IEEE Visualization 2005*, pp. 679-686.

Lau A. and A. vande Moere. (2007). Towards a Model of Information Aesthetic Visualization. *IEEE International Conference on Information Visualization* (IV'07), IEEE, Zurich , Switzerland, pp. 87-92

Lee, B. and C. Plaisant, C. Parr, C. Sims, J-D. Fekete, and N. Henry (2006). Task Taxonomy for Graph Visualization. Copyright 2006 ACM

Lhose, G.L., K. Biolsi, N. Walker, and H.H. Rueter (1994). A classification of Visual Representations. *Communications of the ACM, 37*(12), 36-49

Lightfoot, C. and T. Steinberg (2008). Travel-time Maps and their uses. http://www.mysociety.org/2006/travel-time-maps/ Accessed March 31, 2008

Lu, D. and L. Dietrich (2004). Interactive Poster: Resource System Reference Database. *IEEE Information Visualization Conference*. http://www.tenableinfo.net/ Accessed April 1, 2008

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Trans. Graph*. 5, 2 (Apr. 1986), 110-141.

Mehta, C. (2006). US Presidential Speeches Tag Cloud. http://chir.ag/phernalia/preztags/ Accessed March 31, 2008

MetNet (2008). MetNetDB. http://www.metnetdb.org/ Accessed March 10, 2008.

Munzer, T, F. Guimbretiére, S. Tasiran, L. Zhang, and Y. Zhou (2003). TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. in *ACM SIGGRAPH 2003 Papers (San Diego, CA).* SIGGRPAH '03. ACM, New York, NY, 453 – 462.

Nakamura, Y. (2004). Ecotonoha. https://www.ecotonoha.com/ecotonoha.html Accessed March 31, 2008

Namata, G. M., L. Getoor, B. Staats and B. Shneiderman (2007). A Dual-View Approach to Interactive Network Visualization. in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management.* November 2007. CIKM '07. ACM, New York, NY, 939-942.

Nation, D.A, C. Plaisant, G. Marchionini, and A. Komlodi (2005). Visualizing websites using hierarchical table of contents browser: WebTOC. *The Craft of Information Visualization Readings and Reflections.* Morgan Kaufmann.

North, C. and B. Shneiderman. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. from *AVI 2000, Palermo, Italy*. Copyright 2000 ACM.

Parry, J. (2007). Visualization Techniques for Temporal Information. http://www.joeparry.com/blog/2007/06/visualization-techniques-for-temporal.html

Pfitzner, D., V. Hobbs, and D. Powers (2001). A Unified Taxonomic Framework for Information Visualization. from *2nd Australian Institute of Computer Ethics Conference (AICE2000)*. 2001: Australian Computer Society, Inc.

Phrotohaus (2007). Fidg't. Protomobl Inc. Copyright 2007.

http://www.fidgt.com/visualize. Accessed February 15, 2008.
Pillat, R., E. R.A. Valiati, and C. M.D.S. Freitas (2005). Experimental Study on Evaluation of Multidimensional Information Visualization Techniques. from *CLIHC 2005*, October 23-26 2005. Cuernavaca, Mexico. ACM

Plaisant, C. (2004). The Challenge of Information Visualization Evaluation. May 25 - 28, 2004. Copyright ACM 2004.

Plumlee, M. D. and C. Ware (2006). Zooming Versus Multiple Window Interfaces: Cognitive Costs of Visual Comparison. in *ACM Transactions o Computer-Human Interaction*, Vol. 13, No. 2, June 2006, pages 179-209. ACM 2006

Prefuse (2008). http://www.prefuse.org. Accessed March 4, 2008.

Raskin, J. (2000). *The Humane Interface: New Directions for Designing Interactive Systems.* Addison-Wesley.

Reas, C. and Fry, B. (2003). Processing: a learning environment for creating interactive Web graphics. in *ACM SIGGRAPH 2003 Sketches & Applications* (San Diego, California, July 27 - 31, 2003). SIGGRAPH '03. ACM, New York, NY, 1-1.

Reas, C. and Fry, B. (2004). Processing.org: programming for artists and designers. in *ACM SIGGRAPH 2004 Web Graphics* (Los Angeles, California, August 08 - 12, 2004). SIGGRAPH '04. ACM, New York, NY, 3.

Rhyne, T., M. Tory, T. Munzner, M. Ward, C. Johnson, and D.H. Laidlaw (2003). Information and Scientific Visualization: Separate but Equal or Happy Together at Last. in *Proceedings of the 14th IEEE Visualization 2003 (Vis'03).* (October 22 - 24, 2003). IEEE Visualization. IEEE Computer Society, Washington, DC, 115.

Robertson, G., S.K. Card, and J.D. Mackinlay. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. in *Proceedings of ACM CHI 1991: 189-194,* New Orleans, LA.

Rodenbeck, E. (2007). Data visualization, SOM, and the Transbay Tower in San Francisco. http://content.stamen.com/som_transbay_tower Accessed March 31, 2008

Salathé, M. (2006). Websites as graphs. http://www.aharef.info/2006/05/websites_as_graphs.htm Accessed March 31, 2008

Seo, M. and B. Shneiderman (2005). Interactively Exploring Hierarchical Clustering Results. *The Craft of Information Visualization Readings and Reflections.* Morgan Kaufmann.

Shneiderman, B. (2003). *Leonardo's Laptop: human needs and the new computing technologies.* MIT

Shneiderman, B. (2006). Treemaps for space-constrained visualization of hierarchies. http://www.cs.umd.edu/hcil/treemap-history/ Accessed March 31, 2008

Shneiderman, B. and C. Plaisant (2005). *Designing the User Interface*. Addison-Wesley Publisher.

Shneiderman, B. and C. Plaisant (2006). Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. from *BELIV 2006 Venice, Italy.* Copyright 2006 ACM

Sigmar-Olaf, T. and T. Keller (eds.) (2005). *Knowledge and Information Visualization: Searching for Synergies.* Springer.

Spahr, J. (2003). Website Traffic Map. http://designweenie.com/portfolio/index.php/page/140 Accessed March 31, 2008

Stanza (2004). Sensity. http://www.stanza.co.uk/sensity/index.html Accessed February 15, 2008.

Surface (2008). Microsoft Corp. http://www.microsoft.com/surface Accessed March 9, 2008

TAIR (2008). AraCyc. http://www.arabidopsis.org/biocyc/introduction.jsp Accessed March 10, 2008.

Thissen, F. (2004). *Screen Design Manual: Communicating Effectively Through Multimedia.* Springer.

Trampoline Systems (2006). Enron Email Explorer. http://www.trampolinesystems.com Accessed April 1, 2008

Tufte, E. (2001). *The Visual Display of Quantitative Information, 2nd Edition.* Graphics Press.

Unwin, A., T. Martin, and H. Hofmann (2006). *Graphics of Large Datasets: Visualizing a Million*. Springer Berlin / Heidelberg.

Viégas, F.B. and M. Wattenberg (2007). Artistic Data Visualization: Beyond Visual Analytics. HCII, 2007.

Ware, C. (2000). *Information Visualization: Perception for Design*. Moran Kaufmann.

Wattenberg, M. (2002). Arc Diagrams: Visualizing Structure in Strings. in *Proceedings of the IEEE Symposium on information Visualization (infovis'02)* (October 28 - 29, 2002). INFOVIS. IEEE Computer Society, Washington, DC, 110.

Wattenberg, M. (2006). Visual exploration of multivariate graphs. in *Proceedings of the SIGCHI Conference on Human Factors in Computing System*s (Montréal, Québec, Canada, April 22 - 27, 2006). CHI '06. ACM, New York, NY, 811-819.

Willett, W., J. Heer, and M. Agrawala. (2007). Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Information Visualization.*

Winckler, M. A., P. Palanque, and C. M.D.S. Freitas (2004). Tasks and Scenario-based Evaluation of Information Visualization Techniques. from *TAMOIA 2004, Prague, Czeck Republic*. Copyright ACM 2004.

Wirmachenbunt. (2008). Prototouch. http://www.wirmachenbunt.de/archives/110 Accessed March 31, 2008

Wu, M.; C. Shen, K. Ryall, C. Forlines, and R Balakrishnan (2006). Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. *IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TableTop)*, pp. 185-192, January 2006

Yang, C, H. Chen, and K. Hong, (2002). Internet Browsing: Visualizaing Category Map by Fisheye and Fractal Views. in *Proceedings of the International Conference on Information Technology*.

Yi, J.S., Y. ah Kang, J.T. Stasko and J.A. Jacko (2007). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. Copyright IEEE.

Yul Huh, M. (2004). Line Mosaic Plot: Algorithm and Implementation. *COMPSTAT 2004 Symposium*. Physica-Verlag/Springer.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, thank you to Dr. Julie Dickerson for her guidance, patience and support throughout the research and the writing of this thesis. Her insights and words of encouragement have often renewed my hopes and determination for completing this work on time. I would also like to thank my committee members, Dr. Heike Hofmann and Steven Herrnstadt, for their time and contribution. A big thanks to the biologists who gave me some of their very valuable time to meet with me and look over my ideas: Heather Babka, Matthew Mouscou, Ling Li, and Suh-Yeon Choi. In finishing this work, I cannot forget Starbucks, whose workers never minded the very long hours I put in at the store and without whose coffee I do not think I would have been able to finish.

VITA

NAME OF AUTHOR: Lisa Catherine McGarthwaite

DATE AND PLACE OF BIRTH: June 15, 1982, Granite Falls, MN

DEGREES AWARDED:

B.A. in Computer Science, Graphic Design, Saint Mary's University of Minnesota, 2005

HONORS AND AWARDS:

Presidential Scholarship, 2001-2005
St. Thomas Moore Scholarship, 2001-2005
HCI Graduate Student Fellowship, 2006

PROFESSONAL EXPERIENCE:

Research Assistant, Human Computer Interaction program, Iowa State University, 2006-2008
Virtual Intern, Dairy Programs, USDA, 2003-2008
VoIP Speed Team Intern, IBM, Summer 2007
Co-op Intern, IBM, January 2006 – August 2006
Research Intern, Graphics and Visualization Laboratory, University of California at Santa Cruz, Summer 2005
Federal IT Intern, AAPD/Microsoft, Summer 2004

PROFESSONAL PUBLICATIONS

McGarthwaite, L. (2005). Client-Server versus peer-to-peer Architecture: Comparisons for Streaming Video. *Proceedings of the 5th Winona Computer Science Undergraduate Research Seminar*, April 20-21, 2005.