THINKING INTERACTIVELY WITH VISUALIZATION

by

Remco K Chang

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Information Technology

Charlotte

2009

Approved by:

____
Dr. William Ribarsky

____
Dr. Robert Kosara

____
Dr. Zachary Wartell

____
Mr. Eric Sauda

____
Dr. Robert Anderson

ABSTRACT

REMCO K CHANG. Thinking interactively with visualization. (Under the direction of DR. WILLIAM RIBARSKY)

Interaction is becoming an integral part of using visualization for analysis. When interaction is tightly and appropriately coupled with visualization, it can transform the visualization from displaying static imagery to assisting comprehensive analysis of data at all scales. In this relationship, a deeper understanding of the role of interaction, its effects, and how visualization relates to interaction is necessary for designing systems in which the two components complement each other.

This thesis approaches interaction in visualization from three different perspectives. First, it considers the cost of maintaining interaction in manipulating visualization of large datasets. Namely, large datasets often require a simplification process for the visualization to maintain interactivity, and this thesis examines how simplification affects the resulting visualization. Secondly, example interactive visual analytical systems are presented to demonstrate how interactivity could be applied in visualization. Specifically, four fully developed systems for four distinct problem domains are discussed to determine the common role of interactivity in these visualizations that make the systems successful. Lastly, this thesis presents evidence that interactions are important for analytical tasks using visualizations. Interaction logs of financial analysts using a visualization were collected, coded, and examined to determine the amount of analysis strategies contained within the interaction logs. The finding supports the benefits of high interactivity in analytical tasks when using a visualization.

The example visualizations used to support these three perspectives are diverse in their goals and features. However, they all share similar design guidelines and visualization principles. Based on their characteristics, this thesis groups these visualizations into urban visualization, visual analytical systems, and interaction capturing and discusses them separately in terms of lessons learned and future directions.

ACKNOWLEDGMENTS

This thesis is not the work of a single person. It is a collective effort of a great group of people to whom I owe a great debt of gratitude. Without their support, this thesis would not have happened. I therefore dedicate this thesis to all the people who have helped and participated in my amazing journey at Charlotte. Aside from this acknowledgement, in rest of this thesis I will henceforth not use the pronoun "I", but will instead use the collective term "we" because this thesis truly is the work of many.

Of all the people that have had a hand in shaping this thesis, without a doubt the most important person is my thesis advisor, Bill Ribarsky. Not only has he been an advisor to my research, but he has also been a mentor on how to be an academic. More specifically, I am most grateful to Bill for giving me the opportunity to be part of the birth and growth of the Charlotte Visualization Center in the last five years. As a staff member (and the second "employee"), I have seen the Viscenter when it was just a big empty lab (back in Colvard), and grown to the point today where there aren't enough desks and spaces for all the students and faculties. As an observer and participant of this transition, I have learned a bit of everything from how to establish a research center, apply for grants and support, recruit personnel, to identify and pursue new research agendas. This kind of experience is truly transformative, and one that still leaves me amazed when looking back.

Since this thesis is a product and a testament to the research that took place in Charlotte in the last five years, I need to thank the crew that did all the hard work: from the Data Visualization Group, Dong Hyun Jeong, Tom Butkiewicz, Zach Wartell, Xiaoyu Wang, Wenwen Dou, and Tera Green; as well as folks that "jumped in" as we went along: Caroline Ziemkiewicz, Evan Suma, Alex Godwin, Robert Kosara, Mohammad Ghoniem, Heather Lipford Richter, Dan Keefe, and more recently Lane Harrison. If there is one thing that I am most proud of during my time here in Charlotte, it is that I helped establish an environment where this large group of people worked and played together. To my core team of collaborators/conspirators, thank you. And I mean it.

Outside of the visualization research, I have been fortunate enough to have had the opportunity to work with an extraordinary teacher and an architect. Eric Sauda, who is also a committee member on this thesis, has been the keeper of my sanity. Without a doubt, the one thing that I would miss the most should I leave Charlotte would be my weekly meetings with Eric and our ragtag team of researchers in urban visualization: Ginette Wessel, Elizabeth Unruh, and Jean-Claude Thill. Certainly our team has been successful academically – Ginette had been a publication machine prior to her leaving for graduate school at Berkeley, but it is not just the academic advancements that

I have been drawn to. Instead, it has been the opportunity to work in a stress-free environment where any idea, however crazy or irrelevant, could be discussed. My conversations with Eric have ranged from architecture, computer science, mathematics, cognitive science to politics, philosophy, and general gossip. As Eric put it, our weekly meetings in the past three years really have not been academic meetings, but more like "Remco's therapy hours."

In looking back at how I got to Charlotte, I must thank two people who were instrumental in welcoming me to the university. I had come to UNC Charlotte six years ago as someone merely looking to take a couple of classes, but instead, I found Larry Hodge's Future Computing Lab (FCL). With open arms, Larry invited me to sit in on the lab's weekly meetings and to participate in the social events with the students and faculty of the lab. When Bill came to Charlotte to start the Charlotte Visualization Center, Larry was the one who recommended me to Bill as a research scientist. The trust and confidence from Larry, as well as the friendship of the people in the FCL helped me settle into this new environment and establish myself. Min Shin in particular became a close friend and a confidant during my transitioning period, and for all that they have done for me, I will be forever thankful.

The path that I took to get to this degree has been somewhat unusual, and because of that, there have been a great deal of confusion and occasional frustration. Without the support of those close to me, I would not have stayed on the path for nearly as long as I have. To start with, from the bottom of my heart, I thank Nancy Pollard, my advisor for my Master's degree. Even after leaving Brown with my degree (and bailing out on getting the PhD back then), Nancy has given me advice and guidance when I needed them most. I've always felt somewhat apologetic to her because I'm sure that she had no idea how much work it would take when taking me on as a student over ten years ago. As someone who gave me the foundations to be the person who I am today (both academically and personally), I only wish that there is more I could do to give back. Without Nancy, I would not be writing this thesis, and I hope that with this thesis I've made her proud in some ways.

To my parents and my brother Phillip, I think that an acknowledgement is just not sufficient. The unwavering mental (and financial) support from Mom and Phillip during this exciting but unexpected period of my life cannot be described in words. It is just overwhelming to know that there are people who believe in everything I am and what I could be, even during my darkest times. Throughout this marathon to get the degree and the recognition, there were times when I doubted who I was or how I could ever complete the race. But all I needed to keep myself going was a phone call from them, and I knew that I had the greatest support regardless of how bad things might look. I am so grateful that I have them in my life (and in my back pocket), and I can only hope that I've become half the person they believe I am.

There are many more people who have contributed to my journey in the most unlikely ways: Terris Ko, Sonia Leach, Daniel Acevedo, Alice Rey, Joanne Houlahan, Aditya Paramatmuni, Behrooz Mostafavi, Amanda Cohen, Richard Souvenir, Jamie Payton, Steve Taylor, Cara Mitchell, Dee Ellington to name a few. But above all, I thank Caroline who has stuck with me through the past five years. She has been the keystone to everything I have done in this thesis both literally and figuratively. Literally, there has not been a paper with my name on it that she has not proof-read, or a crazy research thought that she has not carefully scrutinized and refined. But more importantly, she has provided the stability and security for me to set out on this venture. I always knew that I could count on her help for just about anything, from planning what's for dinner to running the analysis on the latest user study. There is no question that I could not have accomplished half of what I have done without having her in my corner and knowing that I have her help and support. I could not have asked for a better partner and friend through this journey. Thanks...

It has been a fantastic time, a time where I learned and grew. To everyone who believed in me and took this crazy ride with us, thank you and hope you enjoyed the trip!

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Visualization has often been thought of as a marriage of interaction and visual representation of data [190]. While many successful and informative visualizations contain only static imagery, adding interaction to the visualizations has the potential to assist the user in manipulating the visual representation of data and conducting comprehensive analysis of problems. As noted by Thomas and Cook in *Illuminating the Path: The Research and Development Agenda for Visual Analytics*, "a [visual analysis] session is more of a dialog between the analyst and the data... the manifestation of this dialog is the analyst's interactions with the data representation [167]." Pike et al. further elaborated on that notion and related interaction directly to knowledge construction: "a central precept of visual analytics is that it is through the interactive manipulation of a visual interface—the analytic discourse, that knowledge is constructed, tested, refined, and shared... These visual displays must be embedded in an interactive framework that scaffolds the human knowledge construction process with the right tools and methods to support the accumulation of evidence and observations into theories and beliefs [131]."

However, although most researchers in the visualization community believe in the sentiment that proper interactive techniques integrated into visual representation of data are important to using the visualization for analytical problem solving, there has been little research specifically in analyzing the effects of interaction in a visualization, or what "proper" interaction techniques should be for a visualization or an analytical task. In most instances when interaction techniques are introduced or examined, they are created as an addition to a novel visual representation design but not as the focus of the research. According to Yi et al. [190], "the representation component has received the vast majority of attention in InfoVis research. A cursory scan of a recent conference proceedings or journal issues in the area will uncover many articles about new representations of data sets, but interaction is often relegated to a secondary role in these articles. Interaction rarely is the main focus of research efforts in the field, essentially making it the 'little brother' of InfoVis."

Given the dearth of research on this topic, the goal of this thesis is to investigate the role of interaction within the context of visualization and visual analytics from the perspective of software development and design consideration. The topic of human-computer interaction in general is beyond the scope of this thesis, but it is through a focus on how analysts and investigators interact with visualization tools that this thesis contributes to the field of visualization and visual analytics. The thesis is therefore divided into three parts: maintaining interactivity in visualizations, example interactive visual analytical systems, and reasoning extraction through interaction logging. In maintaining interactivity, the thesis presents research in urban visualization in which large urban (3D) models are simplified to maintain interactivity. The simplification process is based on the concept of urban legibility such that the simplified models resemble the original model in a way that is difficult to quantify using traditional quantitative measurements. The finding of the research informs that simplification of large datasets is inevitable in maintaining interactivity, but the simplification must be done in a way that does not remove or reduce the core salient features within the original data. In the second section, multiple visual analytics systems are presented to demonstrate the degree and effects of interactivity through concrete examples that have been shown to be effective in allowing the users to perform analytical tasks. Four separate systems are presented and discussed which include a financial wire fraud analysis tool, a visual analytical tool to investigate global terrorism activities, a multi-windowed system for analyzing and comparing biomechanical motion, and an interactive tool that assists the user in understanding principle component analysis (PCA). Through these four examples, the role and effects of interaction are shown to enhance the analytical process during the use of visualization tools. Finally, in the last section on reasoning capturing and extraction, a user-study is presented which quantifies the amount of analytical activities that can be recaptured through examining user interaction logs. The result of the study demonstrates that non-domain experts can recover 60-80% of financial analysts' strategies, methods, and findings by analyzing the analysts' interaction logs when using a financial analytical software. This finding is significant in that it proves that interactivity, when appropriately coupled with visualization systems, does facilitate an analyst's analytical process and that the analysis process can be captured and extracted to a large extent.

This thesis, unfortunately, does not fully address how to design or evaluate interactions in visualizations in a generalizable manner (as in work by Lam on generalizing the cost of interaction [105]). Instead, this thesis has been positioned to be a starting point to a long line of future research for the visualization community with the hope that examples and the studies presented in this thesis can convince the community that interactivity is an integral component to designing visualization and visual analytical tools. In the discussion section (section 5) three specific areas are explored and discussed, including designing visualizations using coordinated multiple views, knowledge-related visualizations, and observations of how financial compliance analysts utilize visualizations. While these learned lessons are not formally evaluated, they are nonetheless potentially useful information to practitioners who wish to understand the motivation and intuition behind this thesis. Together with the conclusions and some future research directions as described in section 6, this thesis hopes to serve as a small step in fully developing a science of interaction in visualization.

CHAPTER 2: MAINTAINING INTERACTIVITY

When coupling interactivity with visualization, one critical consideration is to balance system resource consumption between interactions and the visual representations. Specifically, rendering an image based on a very large dataset often requires large amounts of storage, memory, rendering, and computational resources that could stress a computer's hardware capabilities. If interactions were to be added as a requirement to the visualization, the speed in which the computer needs to render a single image needs to be below 0.08 seconds, which is approximately 12 frames per second (fps). At 12 fps, users of a visualization system considers the rendering to be "smooth", whereas if the frame rate drops below 10 fps, users believe that the system is "choppy" or slow.

One common method for maintaining an interactive frame rate in a visualization system is to simplify or cluster the data into a manageable sizes in order reduce the load on the system during runtime. However, how to simplify the data remains an open question whose answer often depends on the data type and the targeted users. In the case of urban visualization, the difficulty is in simplifying large numbers of 2.5D building models. Traditional methods of simplifying a building could reduce a complex model into a simple box with 8 vertices. However, in a large urban environment where there are millions of buildings (such as a large city like Los Angeles), simplifying each building into a simple box still results in millions of polygons that are difficult to render and retain interactivity.

In this chapter, we present a method for simplifying large numbers of simple 2.5D urban models based on the architectural theory of "urban legibility." Aside from presenting a new algorithm for performing simplification of urban models, the main contribution of this work is the demonstration that the simplification algorithm must be carefully chosen such that the most salient features of the dataset are maintained during the simplification. By definition, simplification implies that certain data elements will be removed or altered from the original dataset. However, the "quality" of the simplification will depend on which data elements are removed or altered during the process. As this thesis shows in the following section (section 2.1), the metric for determining the quality of the simplification is a critical consideration that often goes beyond simple quantitative measurements but instead touches on cognition and humans' innate ability to see patterns.

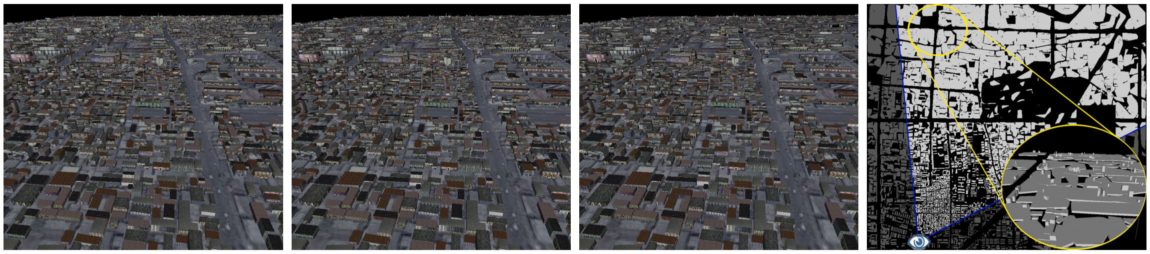
The second section of this chapter (section 2.2) describes an interactive visualization called UrbanVis that incorporates the above simplification mechanism for visual analysis of an urban environment. Since the system utilizes two windows (views), one for the 3D urban model, the other for information visualization, the simplification of the 3D urban model is set to be extremely aggressive in order to balance the system resources. The resulting 3D urban model, although appearing drastically differently from the original, still retains the "image of the city" and can assist the user in navigating the city successfully (section 2.2.4). This result validates the algorithm on utilizing urban legibility as a metric for simplifying a large dataset that preserves the salient features within the data.

To push the boundary further, the third section of this chapter describes the concept of "probes" and incorporates that technique into the UrbanVis system (section 2.3). With probes, the information visualization panel in the original UrbanVis system is no longer a stand-alone window, but is instead directly embedded into the 3D urban model view. Furthermore, with probes, multiple regions-of-interests can be displayed simultaneously, which requires additional computational and rendering resources. As demonstrated in this section, the simplification algorithm is robust and fast enough to accommodate multiple probes (3 to 4) at the same time while remaining interactivity.

2.1 Legible Simplification

Traditionally, research in the areas of mesh simplification and levels of detail has focused on complex models with natural shapes. However, with the advent of 3D global visualization tools for public use such as Google Earth, the ability to render and visualize a large collection of simple models such as buildings has become increasingly important. Beyond this, there is the need to make the rendering of urban spaces useful for tasks such as navigation or developing spatial mental maps.

Existing techniques for mesh simplification can have trouble with models of buildings, which are often nothing more than boxes with eight vertices in which polygon decimation results in a



(a) (b) (c) (d)

Figure 1: Levels of simplification of an urban environment. (a) original model with 285,039 polygons. (b) simplified model with 129,883 polygons, $\epsilon = 100$ $\alpha = 2$. (c) simplified model with 85,332 polygons, $\epsilon = 300$ $\alpha = 2$. (d) shows the same model in (c) from a top-down view. Notice that buildings far away are more aggressively aggregated than the ones near the view point. However, regardless of the amount of aggregation, all clusters obey the rules of legibility. The inset highlights the tall buildings (*landmark*) that are drawn separately to preserve the skyline.

mesh that no longer retains the appearance of a building. Furthermore, when polygon decimation is performed on a collection of simple meshes given a target polygon count, the smaller objects are often completely decimated because their removal causes less overall geometric error (Figure 2(b)). For a city-sized collection of simple buildings, this could mean the disappearance of an entire residential area in which the buildings tend to be smaller than that of commercial regions. This simplified version of the city model no longer resembles the un-simplified one.

In this section, we describe a technique that incorporates concepts from architecture and city planning as guidelines for performing mesh simplification. Specifically, we examine the concept of urban legibility on which Lynch [112] has identified through user studies that the “image” of a city can be categorized into *paths*, *edges*, *districts*, *nodes*, and *landmarks*. We believe that if these elements of legibility are preserved during the simplification process, the image of the city can also be maintained, thereby creating urban models that are better understood by users. Although these categories are qualitative measurements, each step of our algorithm is based on considerations of one or more of these concepts.

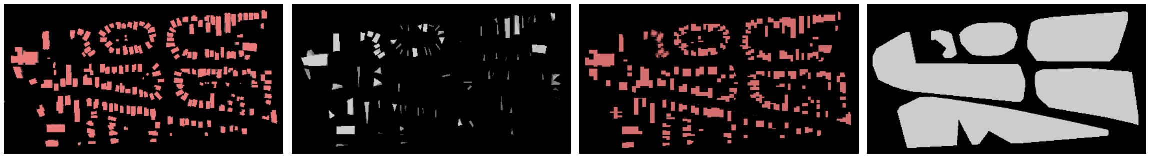
The key idea of our algorithm is based on merging of similar elements. Consider a row of identical houses separated by little space; when these houses are viewed from afar, we should be able to combine their geometries and render them together as one single model (Figure 1). To accomplish this goal, we break our algorithm down into five steps. Hierarchical clustering, cluster merging, model simplification, and hierarchical texturing are performed during pre-processing, and the runtime LOD selects the appropriate models to render. Hierarchical single-link clustering is adopted to cluster models of buildings following the principles of *paths* and *edges*. Polyline-based cluster merging and simplification creates logical *districts* and *nodes* while preserving *paths* and *edges*. Hierarchical texturing creates the appropriate amount of texture for each generated clusters, and finally, the LOD process enforces the preservation of significant *landmarks*.

2.1.1 Related Work

A tremendous amount of research has been put into mesh simplification. For a more comprehensive survey of mesh simplification techniques, see the work by Luebke [111]. Those techniques which are most relevant to our algorithm combine vertices based on their proximities and similarities to other vertices and thus are able to merge multiple meshes into one. Garland and Heckbert [60, 61] introduce QSlim, in which "virtual" edges are added between unconnected vertices that are within a user-specified Euclidean distance τ. These virtual edges are treated in the same manner as actual edges in the mesh. We also take inspiration from Jang et al. [90], who suggest that for man-made objects, removal of entire features is typically more visually understandable than vertex removal. Our goal of urban legibility is similar in that we emphasize understandability over geometric accuracy.

In the field of cartography, the concepts of "generalization" of both building ground plans and 3D building models have been active areas of research. Kada and Luo [94] use the concept of half-space to drastically reduce the complexity of 2D building ground plans. While this approach retains the overall appearance of the original building ground plan, it sometimes creates erroneous, self-intersecting lines. Anders [8], on the other hand, shares our idea of simplifying 3D urban models by aggregating nearby buildings. His algorithm projects the building models onto three orthogonal planes (length, width, and height) and creates a simplified model based on the projections. Unfortunately, this method only works for simple symmetric models that do not self occlude during the projections. In the case of complex models whose geometry need to change in a view-dependent manner, a new algorithm is necessary.

Although there have been numerous applications in 3D global visualization and GIS visualization,



(a) (b) (c) (d)

Figure 2: (a) Original (textured) models of buildings; (b) Models simplified using QSlim; (c, d) Textured and un-textured building models generated with our algorithm using the same number of polygons as the QSlim model

the majority of the research has focused on terrain simplification and visualization (see the survey by Losasso [110]). The use of simplification for displaying collections of building models has been mostly limited to discrete levels of detail in which buildings beyond a certain distance are not rendered. To the best of our knowledge, no existing simplification algorithm prioritizes higher levels of knowledge such as urban legibility.

Urban legibility was introduced by Lynch [112], and the idea has served as inspiration for building virtual worlds, wayfinding in virtual environments, and navigation through abstract data [38]. A number of researchers have also performed user studies to investigate the effectiveness of urban legibility in wayfinding in virtual environments (a comprehensive survey can be found in Dalton's work [38]). Most relevant to our work, Shalabi [149] has used concepts from urban legibility in conjunction with impostors to visualize urban environments of limited scale.

2.1.2 Urban Legibility

Existing techniques for mesh simplification are often not suited for simplifying large collections of objects such as buildings because vertex, edge, or face removal often destroys geometry that is essential to the comprehension and recognizability of such objects. Figure 2 shows a comparison between the original urban model, the model simplified using a traditional simplification method, and textured and un-textured simplified models using our algorithm. The simplified models in Figures (b, c, d) contain the same number of polygons.

It is obvious that these four models' appearances are very different, and arguably they convey different levels of recognizability to the viewer. However, when we use models generated from the three simplification methods in fly-through tests, we find that the amount of pixel error caused

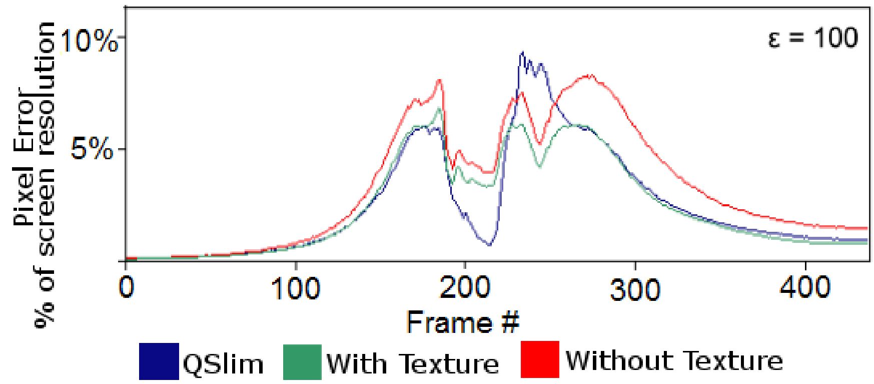


Figure 3: The amount of pixel error as percentages of the screen resolution during a fly-through using an urban model simplified by QSlim, and textured and un-textured simplified models generated using our algorithm. Pixel errors are calculated based on the number of pixel differences between images rendered using one of the simplification techniques and the original model.

by the different simplification methods are quantitatively nearly indistinguishable (Figure 3). In other words, although the different simplification methods create qualitatively and visually different models, the quantitative measurements of the goodness of these models are almost the same.

From this simple experiment, we realize that in order to create models that are legible and understandable to the users, we cannot rely solely on quantitative measurements such as pixel differences. Instead, a higher level knowledge that encompasses our understanding of urban models must be used to guide the simplification process and generate models that are legible to the users. For such high level knowledge, we turn to the concept of urban legibility.

2.1.2.1 Urban Legibility

Urban legibility is a concept that has been used for many years in the area of city planning. In his book *The Image of the City*, published in 1960, Lynch [112] surveyed residents of Boston and asked them to sketch out their neighborhoods in relation to the city (see Figure 4). Based on these sketches, Lynch defined these inhabitants' sense of legibility as "...the ease with which [a city's] parts may be recognized and can be organized into a coherent pattern."

"Coherent pattern" refers to cues that people use to "structure and identify the environment." Lynch further classified them into five types of elements:

Paths: Avenues of travel, such as streets, walkways, railroads, canals, etc.

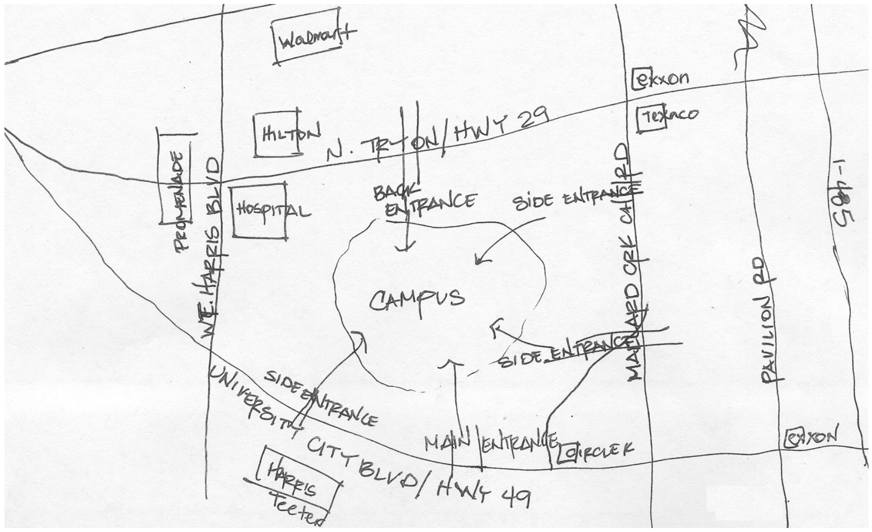


Figure 4: A hand-drawn sketch of the university area of Charlotte, NC. Based on a series of similar images of Boston, Kevin Lynch identified the five elements of urban legibility that he believed were the building blocks of people's mental map of an urban environment.

Edges: Linear elements not considered as paths, including structures or features providing boundaries. For example, shorelines, edges of development, walls.

Districts: Medium to large sections of the city which an observer mentally "enters." For example, a historical residential area.

Nodes: Strategic spots of intense activity and/or information flow, occurring most frequently at junctions of *paths*. For example, Times Square in New York City.

Landmarks: Recognizable objects that are distinctive to the observers. Examples include towers, sign posts, hills, etc.

Although Lynch defined these elements as cues used by the inhabitants of a city, we believe that these elements also help people better recognize a city and maintain spatial coherence from a bird's-eye view. We base our assumption on Haken and Portugali's information theory of urban environments, which states that the amount of information per building in a cluster of similar buildings decreases as the number of buildings in the cluster increases [75]. In order to identify these clusters so as to maximize information while simplifying geometry, we can turn to Lynch's theory of meaningful urban structures. Therefore, it is with Lynch's five legibility elements in mind that we devise our algorithm for simplifying and viewing a collection of buildings. It is our belief that for as

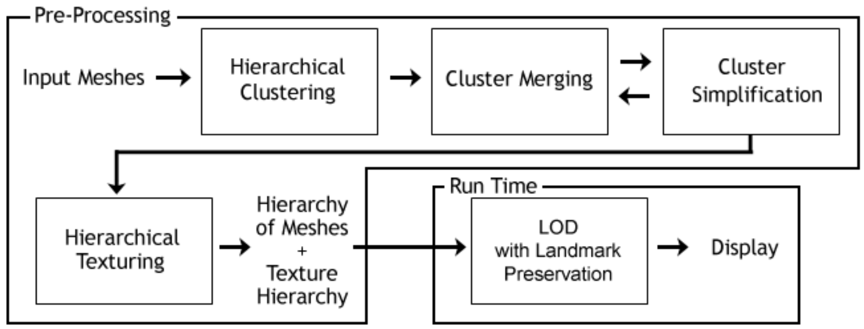


Figure 5: Flow chart of our algorithm showing the four steps in the pre-processing stage, and the LOD step during runtime.

long as the simplification algorithm retains these elements of legibility when reducing geometry or texture of an urban model, the resulting urban model will remain legible to the user.

Our algorithm consists of five steps. Hierarchical clustering maintains *paths* and *edges* when grouping similar buildings together; cluster merging combines the geometries of the buildings into a single model and creates *districts* and *nodes*; simplification reduces the geometric complexity of the model while preserving *paths*, *edges*, *districts* and *nodes*; texturing adds visual fidelity to the created model; and finally, the LOD process selects the appropriate models to render at runtime, preserving the *landmarks* in the scene. The following sections describe the five steps in our algorithm.

2.1.2.2 Hierarchical Clustering

Lynch considers *paths* as the predominant city elements, so it is critical that our clustering algorithm does not cluster buildings on opposite sides of a path. Our algorithm does not require knowledge of the street layout, only building footprints, with which we maintain both *paths* and *edges* by preserving the empty spaces between buildings. To achieve this, we use single-link clustering, which creates clusters that respect boundaries of paths. In contrast, k-means and complete-link clustering both produce oval-shaped clusters that do not consider *paths* or *edges* (Figure 6).

Single-link clustering is a greedy algorithm. At any given iteration, it finds the closest pair of elements or clusters (in Euclidean distance) and groups them together into a cluster. Therefore, for n number of elements, single-link clustering requires $n-1$ iterations, and produces a total of $n-1$ clusters. The greedy nature of the algorithm relies on distance minimization, and guarantees that

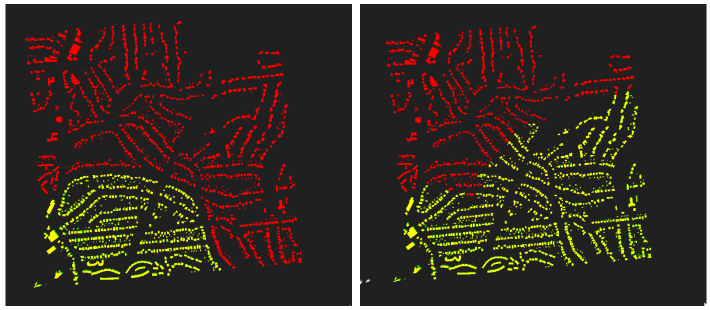


Figure 6: Left: single-link clustering. Right: complete-link clustering. There are two clusters in both images, the red buildings constitute one cluster, while the yellow ones fall into the other. Notice that single-link clustering creates clusters that follow along a path, whereas complete-link clustering, much like k-means clustering, creates oval-shaped clusters.

a building will be clustered with other buildings on the same side of the road before being grouped with buildings from the other side of the road, as long as the distances between the buildings are not wider than the width of the road. In most urban environments in which buildings are very close to each other, this property of single-link clustering ensures that clusters obey *paths* and *edges*.

Unfortunately, single-link clustering does not guarantee the resulting cluster hierarchy (dendrogram) is balanced, meaning that the depth of the tree is often far from the ideal depth of $lg(n)$. To create a moderately balanced tree, we use a distance metric that incorporates cluster size:

$$d(C_1, C_2) = min\{\frac{size(C_1) \cdot size(C_2)}{avgClusterSize^2} \cdot d(x, y) | x \in C_1, y \in C_2\} \tag{1}$$

where $size(C_x)$ denotes the number of buildings in cluster C_x, $d(C_1, C_2)$ is the cost of merging clusters C_1 and C_2, $d(x, y)$ is the Euclidian distance between the two closest vertices in buildings x and y, and $avgClusterSize$ is the average number of buildings in all the existing non-leaf nodes. With the introduction of $avgClusterSize$, we are able to reduce the depth of the tree from 726 to 170 in the Charlotte dataset, which contains 370,000 buildings.

Note that an optimal implementation of single-link clustering has the complexity of $O(n^2)$. However, our single-link clustering is a $O(n^3)$ algorithm. This is due to the fact that $avgClusterSize$ changes at every step, causing all distances to have to be recomputed. For efficiency, we pre-compute the nearest neighbors of each building and only consider the nearest neighbors during clustering. This reduces the complexity of the algorithm to $O(k^3)$ where k is the number of neighbors. We found empirically that a value of 50 for k is sufficient for all of our models in producing clusters that follow *paths* and *edges* while drastically reducing the clustering time.

2.1.2.3 Merging and Simplifying Clusters

Once the clustering process is complete, each node in the hierarchy contains a number of buildings that are geographically near each other and are roughly bounded by *paths* and *edges*. We then merge the buildings within each cluster into a single geometric model (called a merged hull), which contains and resembles the aggregate of the buildings. This merger often creates logical *districts* (Figure 7). The merging of clusters is recursively performed until all (non-leaf) nodes in the hierarchy contain a merged hull.

Because all the buildings are 2.5D, we can consider merging the footprints of the buildings separately from merging the heights of the buildings. This permits us to apply different rules to the footprints and the heights. Our merging algorithm begins by finding the 2D convex hull of all the buildings' ground vertices, and iteratively subdivides a line segment to form the largest possible triangular area between the original line segment and the two newly created ones (Figure 8). Based on the inverse of the one-mouth theorem [170], this process can continue until no line segment can be further subdivided without causing self-intersections.

(a) (b) (c)

Figure 7: Creating a *district* by merging two clusters (a) and (b) into (c)

The computation of this iterative line subdivision method has the worst case upper bound of $O(n^3)$ where n is the number of vertices. For the sake of efficiency when computing the merging of the clusters, we stop the subdivision process when the target number of edges is reached. Although the target number is a changeable parameter, it is imperative that the number of edges in a parent

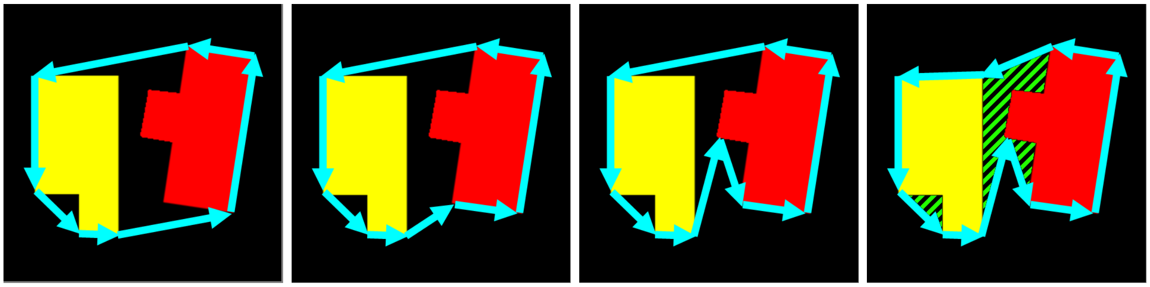


Figure 8: Steps of cluster merging and simplification. (d) The convex hull of the two merging clusters; (a, b, c) three iterative steps of line segment subdivision. Notice that the merger introduces geometric error, or negative space, as highlighted by green stripes in (d).

cluster is less than or equal to the sum of the two children's number of edges. By adhering to this rule, the cluster hierarchy remains monotonic in that traversing deeper down the hierarchy always results in more geometric detail. Empirically, we find that setting the parent cluster's number of edges to be 75% of the two children's combined edges results in a good balance between computation time and geometric detail.

Once the simplified merged hull has been calculated, we define the height of the cluster to be the weighted average height of all the buildings in the cluster, where the weight of each building is directly proportional to its area. Buildings with dramatically different heights are considered as *landmarks* and handled separately during the runtime LOD. The final polygonal model for each cluster (called a cluster mesh) is produced by protruding the merged hull towards the sky to the height of the cluster, and using OpenGL (GluTesselator) to create the triangulation of the roof.

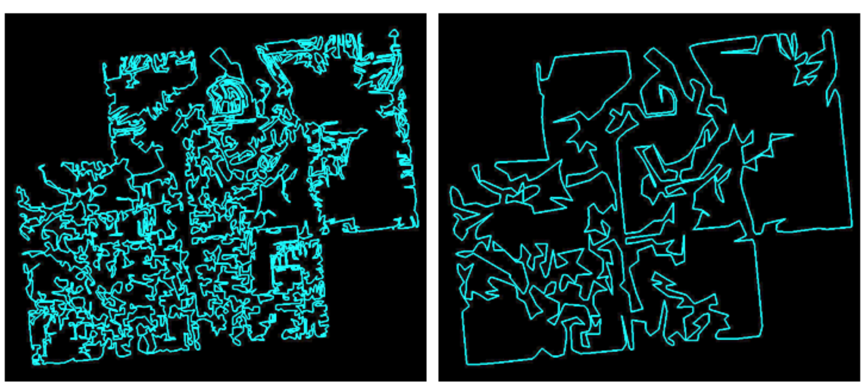


Figure 9: Polyline simplification. Left: 6000 edges; Right: 1000 edges.

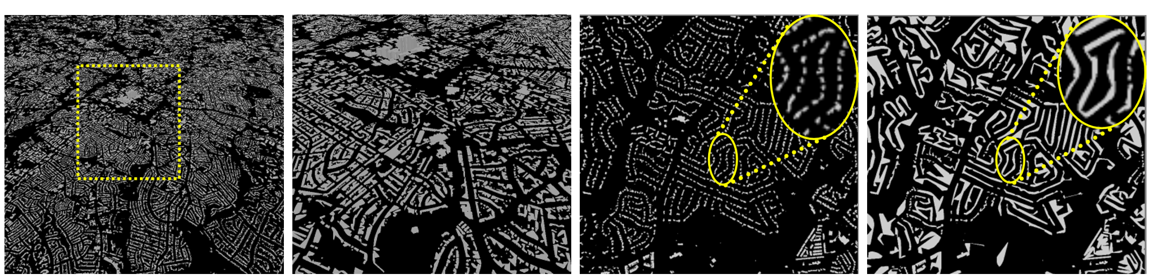
We find the result of our merging and simplification process to produce clusters that are understandable in the form of *districts* while preserving the elements of *path*, *edges*, and *nodes* (Figure 9). Furthermore, similar to the concept of half-edge collapse, no new vertices are created during this process, making this algorithm very useful when used in conjunction with OpenGL vertex arrays for both speed and memory conservation.

2.1.2.4 Negative Spaces and Level of Detail

During the hull merging and simplification process, geometric errors are introduced into the final mesh. We call these geometric errors "negative spaces" because geometry is added to previously empty spaces.

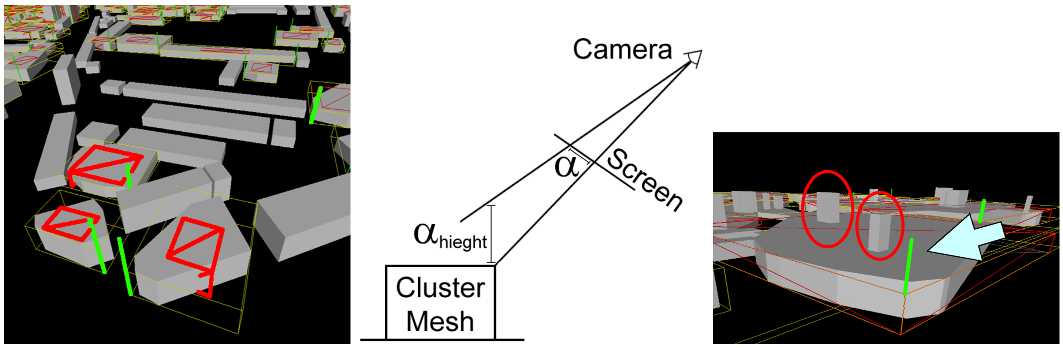
The area of the negative space of a cluster mesh is the difference in area between its footprint and the sum of the buildings' footprints. Our LOD algorithm will not render a cluster if the visual effect of this area is too large. In our implementation, we approximate this negative-space area as a rectangle with the same ratio in dimensions as the axis-aligned bounding box of the merged hull (Figure 11 (a)). During the LOD process, this negative-space area is converted into a 3D box with the same height as the cluster mesh. The camera-facing sides of the box are projected onto screen space, and the number of pixels is compared against a user-defined tolerance (ϵ). If the number of pixels is greater than ϵ, the cluster will not be rendered, and its descendants will be checked recursively (Figure 10).

The concept of *landmarks* is perhaps the most subjective of Lynch's categories. However, it is reasonable to assume that taller buildings have higher visual importance than shorter ones because



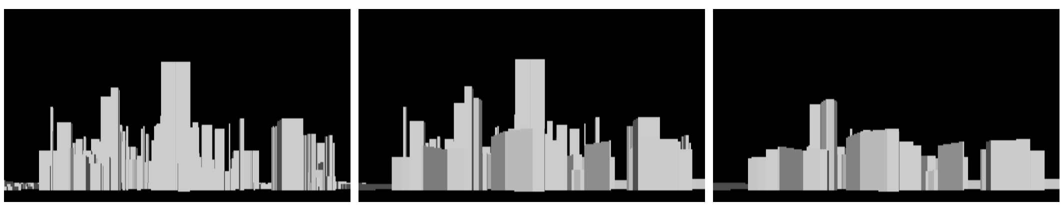
(a) (b) (c) (d)

Figure 10: (a) View of downtown Charlotte and its surrounding regions from afar; (b) what is actually being rendered when the selected yellow box region in (a) is enlarged. (c) using a pixel tolerance ϵ of 50; (d) setting ϵ to 500: notice that as ϵ increases, so does the amount of simplification. Nonetheless, both models obey the principles of legibility.



(a) (b) (c)

Figure 11: (a) The negative spaces are approximated as 3D boxes, shown in red. The green lines represent α_{height}, or acceptable height tolerance for each cluster; (b) Finding α_{height}: the user-defined height tolerance α is projected onto a cluster mesh and converted to height tolerance (α_{height}) in world coordinate. (c) Buildings in a cluster mesh (circled in red) that are taller than (α_{height}) (shown as a green line), are considered to be *landmarks* and are drawn separately over the cluster mesh.



(a) (b) (c)

Figure 12: Bottom row: (a) Original skyline (243,381 polygons) (b) Simplified skyline that resembles the original skyline ($\alpha = 2$, $\epsilon = 100$, 15,826 polygons) (c) Skyline that loses its resemblance to the original skyline by not maintaining *landmarks* ($\alpha = 10000$, $\epsilon = 100$, 13,712 polygons).

of their roles in defining the skyline. A user-defined threshold (α) in numbers of pixels is used to determine the acceptable error in height. During runtime, α is projected onto each cluster mesh and converted to a height value in world coordinate (called α_{height}), shown in Figure 11(b). If any building is taller than its cluster's α_{height}, the original building mesh is rendered along with the cluster mesh (Figure 11(c)). Figures 12(a, b, c) show the affects of changing α on a city's skyline.

2.1.2.5 Hierarchical Texture

As with the above urban geometry simplification, the purpose of our hierarchical texture approach is not visual quality in its strictest sense, but rather legibility of the urban environment at all scales. As such, the main goal for texturing is not necessarily to enforce small or even unnoticeable pixel errors. Instead, the goal is to create textures that maintain legibility and interactivity.

It is generally accepted that texture mapping is one of the most resource-intensive processes in graphics rendering. For our application, the texture problem is doubled because we generate $n-1$ new cluster meshes in which the geometries are often different from the original models, making it impossible to reuse textures.

To create side textures for each cluster mesh, we iteratively generate an image for each face by placing an orthographic camera such that its near clipping plane lies on the face. The combined images from all faces are set to fit into one single texture, with the resolution of each image proportional to the length of each face.

Texturing the roof is more difficult because the negative spaces between buildings are more visible from a top-down view. If the camera angle changes slightly from such a view, the viewer expects to see parts of the facades of the buildings. We take images of the roof from five different camera angles: a top-down view and 45-degree views from north, east, south, and west. Buildings are scaled to the height of the cluster mesh to avoid “shifting” between the camera angles. During runtime, the system chooses the texture that is closest to the look vector.

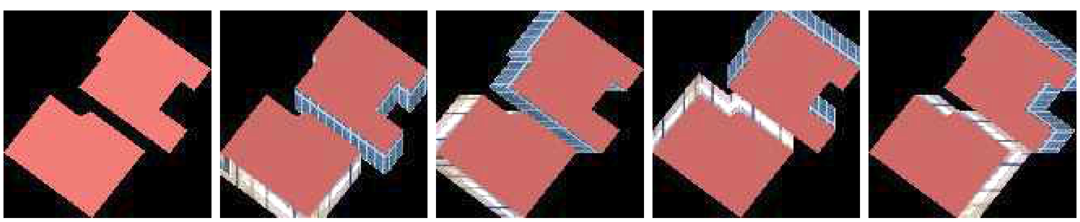


Figure 13: Five textures for the roof generated from different camera angles. Left to right: top down, south, west, north, and east.

Due to the limited texture resources, it is important that we constrain the amount of texture generated to a minimum. Analytically, we can find the upper bound of the maximum texture resolution required for any cluster mesh given the user-defined pixel tolerance ϵ. Since no cluster mesh should have a texture resolution that exceeds its maximum size on screen, the cluster mesh's maximum size defines the upper bound of the texture resolution. We know the footprint area of the cluster mesh, and the area of the negative space. So given ϵ, we can trivially compute the maximum size that a cluster mesh would ever appear on screen, and hence the maximum number of texture pixels for the roof t_{roof} and the sides of the cluster mesh t_{side}:

$$t_{roof} = (area_{clusterMesh}/area_{negativeSpace}) * \epsilon \tag{2}$$

$$t_{side} = (\sqrt{area_{clusterMesh}}/\sqrt{area_{negativeSpace}}) * \epsilon * 4 \tag{3}$$

In the case of t_{side}, we multiply the number of calculated pixels by 4 because there are four sides to each 3D negative space. Although the fact that t_{side} does not depend on the height of the cluster mesh is counter intuitive, it indicates that the geometry simplification and texturing are inter-dependent. During runtime, given ϵ, the LOD process chooses the appropriate geometry to render, and it guarantees that the associated texture for that geometry is optimal.

Due to the large size of the urban datasets, texture storage and retrieval is also a critical aspect in rendering. Conceptually speaking, we separate clusters into groups that are geographically close to each other and combine their textures into texture atlases. In practice, we consider three types of clusters - those in the cut, those above the cut, and those below the cut, where the cut is found by using the LOD process looking down at the city model. The clusters in the cut are clusters visible in the LOD and approximately equal in size. Together, they define groupings of buildings that are geographically close to each other.

Choosing an appropriate texture atlas size is also a challenge. Large atlases require fewer disk I/O operations and can make more efficient use of space, but we found the lower runtime texture memory needed for small atlases to be worth the tradeoff. In the Charlotte dataset, we use texture atlases of 256x256, and create a cut of approximately 12,220 groups.

During runtime, the geometries of the model are left in-core. However, to ensure that there is enough memory for the textures during runtime, we implement a simple priority queue in which the least recently used texture is swapped out when memory becomes a constraint.

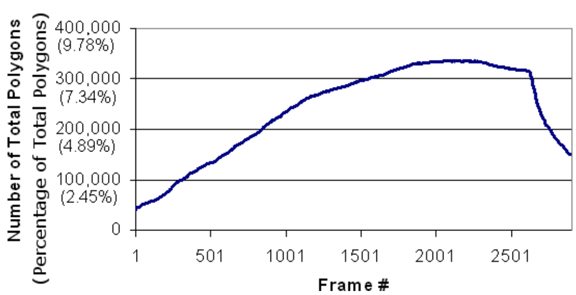
2.1.3 Results and Analysis

All graphs shown in this section are generated using a fly-through of the Charlotte model which contains 369,929 buildings and 4,088,254 polygons (with the parameters $\epsilon = 100$ and $\alpha = 2$). To showcase our system's speed and efficiency from all levels of detail, the camera's fly path starts zoomed-in locally on a number of buildings at the ground level, and then zooms out and up slowly until the entire city is in view from above. Note that the Charlotte model does not contain accurate height information or textures.

The computer used in generating the following statistics is a Pentium 4 3.0Ghz desktop with 2GB of RAM and an nVidia 7950 graphics card with 512MB of memory running Windows XP Professional.

Polygon Count: We demonstrate that our algorithm drastically reduces the number of polygons rendered during a fly-through in our Charlotte model. Figure 14 shows that the number of polygons rendered ranges between 1% to 8% of the total polygons.

Frame Rate: Figure 14 shows the frame rate of the fly-through of Charlotte using our algorithm (shown in blue) compared to the rendering of each building individually (shown in magenta). Both implementations use OpenGL Vertex Buffer Objects (VBO). However, our algorithm takes advantage of hierarchical frustum culling in removing clusters outside of the view frustum, whereas the implementation of rendering individual buildings does not cull buildings outside of view.



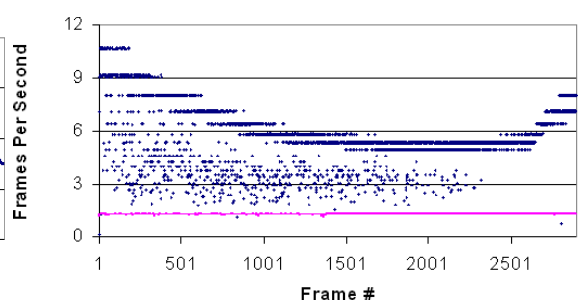


Figure 14: Left: The polygon count and the percentages of the total number of polygons of the original Charlotte model during a fly-through. Compared to the unsimplified Charlotte model (4 million polygons), our algorithm drastically reduces the polygon count to less than 9% from all view distances and angles during the fly-through. Right: The frame rate of the fly-through of Charlotte. The blue dots show the frame rate using our algorithm, and the magenta dots show the frame rate if every building is drawn individually in every frame. Note that the fluctuations in our frame rate correspond to the frames in which textures are loaded from file.

The fluctuations in the frame rate correspond to the frames in which the system fetches the needed textures from the file system. Our current implementation of texture-loading does not include pre-fetching of textures, so textures are loaded from file on an as-needed basis. While pre-fetching is a well-studied technique, we note that adding pre-fetching into our system is not a trivial task due to the already large memory requirement when viewing a textured, city-sized urban model. Because of the memory constraints, correctly identifying the needed textures for pre-fetching is a difficult problem that is outside the scope of this project and requires further investigation.

Preprocessing Time: Table 1 shows the amount of time required for each stage of preprocessing. Although the computation can be performed in parallel, for consistency's sake, all pre-processing is done on a single computer. Note that the large difference in clustering times between the two datasets is due to the fact that the distance metric of the clustering algorithm is based on distances between vertices. Therefore, the more detailed the buildings are, the longer it takes to calculate the minimum distance between them.

Table 1: Statistics and preprocessing times for the Xinxiang dataset and the Charlotte dataset.

	Xinxiang	Charlotte
Number of Buildings	95,042	369,929
Number of Polygons	814,592	4,088,254
Number of Ground Vertices	502,336	2,414,056
Clustering	8 mins	102 mins
Merging and Simplification	3 mins	15 mins

2.1.4 Discussion

Evaluation of the legibility of a simplified city model remains an open problem, as it is difficult to quantify a person's sense of spatial awareness in an urban environment. While spatial cognition is an active field, it has not often been related to the analysis of urban legibility, as defined by Lynch, in the context of urban visualization. As noted in Section 2.1.1, some researchers have tried to understand legibility via user studies [38], while others attempted to quantify it mathematically, but so far there has not been a generalized rule that can be applied to evaluate urban legibility. One promising direction is to develop benchmark localization and navigation tasks to evaluate user performance through user studies [38].

We have performed an informal expert evaluation on an exploratory system built on our simplification method called *UrbanVis* [25], which examines Census information within an urban environment.

The system was evaluated by surveying 14 experts with disparate backgrounds including geographic information system (GIS) experts, city planners in local government, school district planners, population experts in academia, and commercial real estate developers. The experts were given some time to familiarize themselves with the system, then asked for subjective feedback on the potential usefulness of the system in their daily tasks as well as any suggestions on future improvements. The results of the survey indicate that for the purposes of navigation and spatial awareness and understanding, urban models created based on our simplification algorithm remain legible at all levels of simplification, even in the most extreme cases.

This finding is significant in two ways. First of all, it suggests that the legibility elements introduced by Kevin Lynch do indeed form a basis for most people's mental understanding of an environment. This is interesting because an individual's concept of a "legible" environment is inherently subjective; how a person orients himself often depends on his familiarity with the surroundings. A long-time resident of a city could use a local restaurant as a landmark, whereas tourists might rely on skyscrapers and major roads for orientation. However, based on our survey results, Lynch's legibility elements may serve as a common ground between each person's mental image of the urban environment. All participants of our survey, with their disparate backgrounds, understood and oriented themselves successfully using the simplified urban model generated with our algorithm.

More importantly, the results of the survey strengthen the argument presented in Section 2.1.2 that pixel-level accuracy in simplification is often not the most important measurement in evaluating urban simplification algorithms. As Figure 3 shows, qualitatively different urban models can produce similar quantitative pixel errors. Our evaluation reinforces this point by demonstrating that even with an aggressive amount of simplification, the user retains spatial awareness of the environment. This indicates that if the simplified model resembles the user's mental image of the urban environment, the user can successfully orient himself and navigate the environment effectively.

2.1.5 Future Work

There are a number of opportunities for improving our algorithm. Currently, the clustering process merges clusters based on Euclidean distances between vertices of buildings, but we could also include other attributes such as buildings' colors, textures, sizes, and shapes in the distance function. Furthermore, it would greatly generalize our algorithm to integrate mesh decimation techniques in the pre-processing step so that we can accept true 3D building models of arbitrary geometric complexity. This extension along with improving the texture loading process to incorporate concepts such as pre-fetching will enable us to address a challenging problem in urban rendering, namely, flying freely over very large urban scenes at a birds-eye view and then diving in at any time for detailed close-ups of any building, with everything unfolding smoothly and naturally.

In the rendering aspect, we acknowledge that our system does not utilize OpenGL's Vertex Buffer Object (VBO) to its full potential, which limits the frame rates shown in the Results and Analysis section (Section 2.1.3). Two limitations prohibit the rendering engine from reaching ideal frame rates; first, OpenGL's current implementation of VBO limits each vertex to one surface normal. Unfortunately, in the case of 2.5D models, each vertex needs to represent two surfaces with different surface normals that cannot be blended. In order to render each face of a building with a different surface normal, we render each face independently after setting its surface normal (using GL_QUADS instead of GL_QUAD_STRIP). The second limitation of the rendering engine is due to updating the VBO on every frame. In our implementation, ground vertices and roof vertices share the same (x, y) coordinates and are stored in the same VBO. During runtime, however, the z values (height values) of the roof vertices need to be updated every frame based on their corresponding cluster meshes. Although no new vertices are introduced into the VBO, the z coordinate updates still slow down the rendering speed. Overall, our evaluation suggests that the bottleneck in our rendering system is not in the graphics card, but is instead in the data transfer between the CPU and the GPU. We believe that if these two issues can be resolved, the rendering speed should increase significantly.

Lastly, we are very interested in furthering the evaluation of how individuals understand their surroundings in an urban environment. We would like to examine through user studies how the first three images in Figure 1 differ perceptually and cognitively. If we can understand better the way mental images of urban surroundings are formed, we believe we can apply those concepts and create more simplified, yet more legible urban models.

2.2 UrbanVis: Coordinated Urban Visualization

In the previous section we introduce a method for simplifying large urban models while retaining their legibility. We show that simplification of urban models should not be done solely using traditional vertex, edge, or polygon removal strategies. Employing such strategies may produce quantitatively acceptable models, but not models that are understandable or legible. Instead, we contribute the concept of simplification through merger of 3D urban models in which geometries and textures of buildings are merged together to create new models. We argue that the resulting simplified models, although not quantitatively better than models created using traditional simplification methods, are more understandable and legible to the user. This is particularly true when the merger is guided by the concept of urban legibility as described by architects and urban planners. Each step of our simplification process seeks to retain or create elements of legibility. Single-link clustering groups buildings into clusters that adhere to the boundaries of *paths* and *edges*; cluster merging creates logical *districts*; polyline simplification maintains *paths*, *edges*, *nodes*, and *districts* while reducing the model's complexity; and the runtime LOD process renders only visually appropriate models while preserving the skyline by identifying *landmarks*. Finally, analytically determining the necessary amount of texture and applying it hierarchically to the models strengthens the visual fidelity of the urban scene.

Our quantitative and qualitative evaluations indicate that our method not only reduces the geometry of complex urban models, but successfully preserves the legibility of the environment on all scales. The resulting simplified models are faster to render and still effective in a user's navigation and orientation of the urban environment. As we demonstrate in this section, gaining the faster rendering speed without losing the overall understanding of the urban spatial awareness is critical in maintaining interactivity in visualization. Specifically, in this section, we integrate the 3D viewer described in the previous section with information visualization relating to a city and show that by utilizing a high degree of interactivity, a user can navigate through complex urban environments and understand their underlying information at the same time.

Most existing urban model visualization systems focus on layering a few dimensions of data over a 2D map or a 3D model with a limited number of buildings. Often the layering uses colors to depict the data, which quickly limits the number of layers that the user can see at the same time before the combinations of layers become too complex to understand. More importantly, existing systems limit the user's interactions when focusing on specific regions of interest. Specifically, many systems allow the user to drag a bounding box around the area of interest for zooming in. This interaction diminishes the user's understanding of the selected region of interest in relation to the rest of the city both in the sense of spatial relationships and the underlying depicted data.

From interviews with architects and urban planners, we recognize that visualization of an urban model must occur on all levels of abstraction. For example, when the architects and planners are asked to describe New York City, the descriptions always range from a global level such as "New York is large, compact, and crowded," to the local level such as "the area that I lived in had a strong mix of ethnic groups." Furthermore, there is often a strong sense of relationship in the localized descriptions, "the community that I lived in is more heterogeneous than the surrounding neighborhoods." These comments combined indicate that not only do urban visualization tools need to be multi-resolution, the tools also need to show relationships among neighborhoods in a focus-dependent manner.

Our approach is therefore quite different from existing ones. We build on the idea of urban legibility, as described in section 2.1.2.1. Rather than being just random collections of buildings, a city has certain parts that people intuitively understand and aggregate when describing it from different levels of abstraction. These understandings and aggregations are often based on people's tendencies towards neighborhoods of similar ethnicities and social backgrounds. Together, they form parts of the basis of the elements of legibility as defined by Lynch. Using these legibility elements, we build a tool that provides not only the spatial view but also an information display depicting abstract data such as demographic information, land use, etc. The spatial data is linked with the abstract data so that they maintain and provide the same understanding and aggregation through all levels of abstraction.

Using the tool we developed, *UrbanVis*, the user can find parts of a city that are defined in terms of their spatial layout or boundaries, and then explore their properties. How similar are the people living in a borough, district, or neighborhood? What is the distribution of ethnic groups throughout a city? Through these explorations, the user can begin to understand the properties of the city and envision how changes would impact the urban environment, not just in terms of the physical buildings, but also how such changes affect the social infrastructure. What happens to surrounding neighborhoods if we put a school here? How will changing an area from residential to commercial zoning affect the local economy?

Our approach is unique in that it builds an urban visualization on a clustering algorithm with the goal of providing physical and informational views to the user that are easy to understand from all levels of abstraction. By aggregating the data based on the elements of legibility, UrbanVis opens up many possibilities for exploration and re-examination of existing understandings of a city.

For the user evaluation, we surveyed fourteen experts with different occupational backgrounds ranging from real estate developers and urban planners to geographic information system (GIS) users. From this user evaluation we formally identify features of the system that were most useful to these professional urban experts as well as a range of possible future directions. We concluded that a majority of the participants believed our visualization tool enabled them to better perform their daily tasks as it provided new features that were not available in current commercial software systems.

2.2.1 Related Work

We build on work in urban planning and urban legibility and the connection between information visualization and geographical views.

2.2.1.1 Urban Legibility

Urban planning has focused largely on the use of social, economic, and political factors in evaluating urban growth and development [62]. The methodologies are adapted from the social sciences, and involve accumulation and analysis of complex data. There is relatively little emphasis given to the detailed form or geometry of the city. The work that does seek to connect social and political factors to urban form does so on a local rather than a city or regional scale [106].

Alternatively, urban design has focused on the form and geometry of the city. Traditionally, simple geometric models of the city have been the basis of discussion and design, either in planimetric view [142], in sequential perspective view [37] or using cognitive mapping [112]. There has been relatively little emphasis on policy factors in urban design, and little way to relate these issues to urban form.

There has been recent work seeking to explicitly link urban morphology and underlying economic, social and politics at all scales of urbanism. The work done at Harvard directed by Rem Koolhaas has sought to weave the economic, political and social factors explicitly into the development of urban form [31]. Mitchell, in his book *City of Bits* [117], wrote about the emergence of urban forms that will change from fixed ideas of space to shifting realms of intersecting digital and spatial networks. Neither of these efforts has been explicit about tools that will enable these new insights.

2.2.1.2 Information and Geographical Visualization

Commercial GIS software provides tools which manage and display georeferenced data. Commonly available GIS software products are ESRI's ArcGIS [49], Intergraph's GeoMedia [88], MapInfo's SpatialWare [116], and GRASS GIS [69]. Mapping components of the software allow the user to view, analyze, and present geographic data. These GIS software systems generate 2D or 3D representations using the constraints from data input into the program. In order to test alternative scenarios the input data must be altered and a new output generated. A frequent criticism of GIS software systems is that they are cumbersome, over-complicated, resource-hungry, and require specialist expertise to understand and use [39].

Several attempts have been made to solve these problems by combining information visualization views with geographical views. Geo-VISTA Studio [56] is noteworthy as an early attempt to integrate classical GIS visualization with information visualization views including parallel coordinates [87, 45]. Improvise [182], a system with similar principals but also with extended flexibility in browsing visualizations of relational data, has been used to visualize the US Census data [183]. A classic example of the combination of a geographical view with interactive querying is HomeFinder [186],

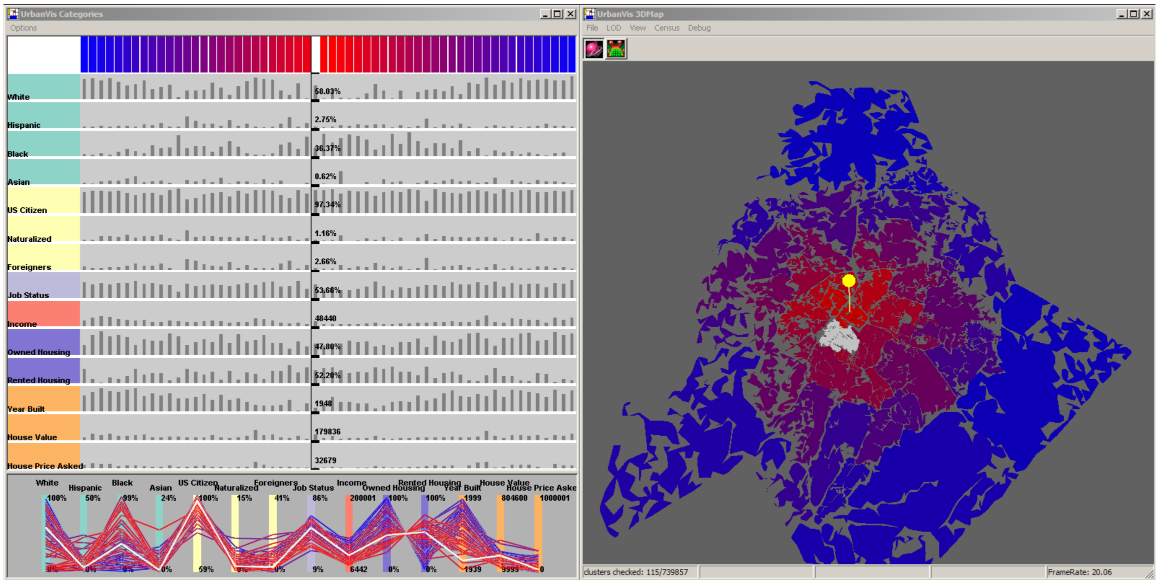
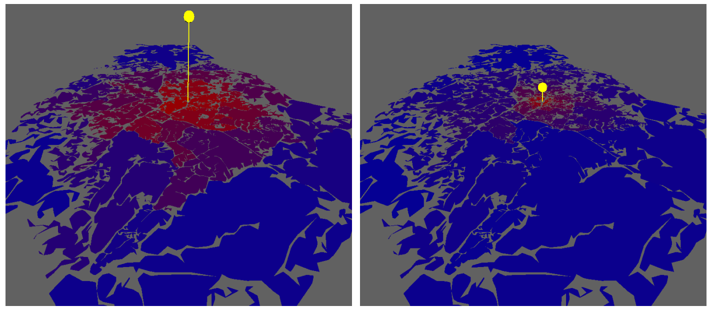


Figure 15: UrbanVis overview. The data view on the left shows demographic data of the areas around the focus point (focus in the middle). The model view on the right shows the clustered building models. The color gradient indicates the distance from the focus point, and provides a visual link between the two different data views (matrix view and parallel coordinates) and the model view. The data shown is census data for the city of Charlotte in Mecklenburg county, North Carolina. The straight lines in the lower half of the model view are where the city and county border South Carolina.

which lets users find houses that fulfill certain criteria. Dykes et al. [43] give a survey of the existing systems and future trends of geovisualization. Our system differs from existing approaches in that we focus on aggregation of the models in the geographical view, thus allowing the user to choose the desired level of focus without losing the overall context. Shanbhag et al. [150] use visualization of demographic data over time to validate partitionings. This is very close in spirit to our work, but lacks any data on the physical layout of buildings, separators, etc.

The task of evaluating the effectiveness of geovisualization systems is not trivial [53, 5]. Although researchers have tried to measure the effectiveness in a quantitative manner using task-based user studies [102, 168], the issue of measuring usefulness of geovisualization systems that are designed for exploratory purposes without specific tasks remains an open problem [53, 138]. For our system, which is designed to be exploratory in nature, we choose an evaluation based on the opinions of domain expert users.



(a) (b)

Figure 16: Changing the zoom level of the focal point (shown as a yellow sphere and a line connecting it to the ground). The color gradient from red to blue shows the proximity of the clusters to the focus. a) When the sphere is far away from the ground, the region of interest is larger, and the user can see an overview of the area at a glance; b) when the sphere is closer to the ground, the region of interest and clusters are smaller, thus allowing a more detailed inspection.

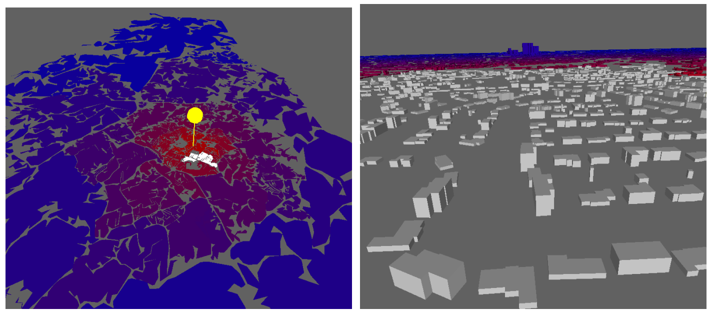
2.2.2 System Overview

The system uses two views (Figure 15): a 3D model view and a multi-dimensional data view. The views each have their own window, making window management easier on setups with two screens or projectors. The two views are fully linked and each accepts user interaction. The 3D model view shows clusters of buildings based on legibility elements and provides spatial awareness within the urban environment. The data view displays abstract information of the clusters shown in the 3D model view and adds an extra perspective for understanding the city. Together, the views allow the user to explore the urban model from both the geographical and the informational angles.

2.2.2.1 3D Model View

The 3D model view (Figure 15, right) shows the geometries of the buildings in the city, and thus acts as a navigation tool and the display for building clusters at the same time. The user can interactively navigate the city using either mouse or keyboard and view the city from any view distance or angle.

The focus which guides the aggregation of buildings is normally the eye point of the camera which the user controls. To decouple the clustering from the viewing, the eye point is represented by a yellow sphere which is connected to the ground with a thin line. The user can move the sphere around the map and also up and down to change the clustering: when the sphere is high above



(a) (b)

Figure 17: (a) User selects and highlights a cluster in the model (shown as white); (b) at any time, the user can change the view to looking at individual buildings instead of clusters.

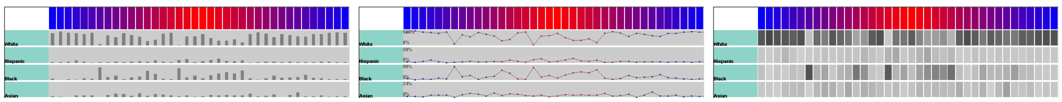


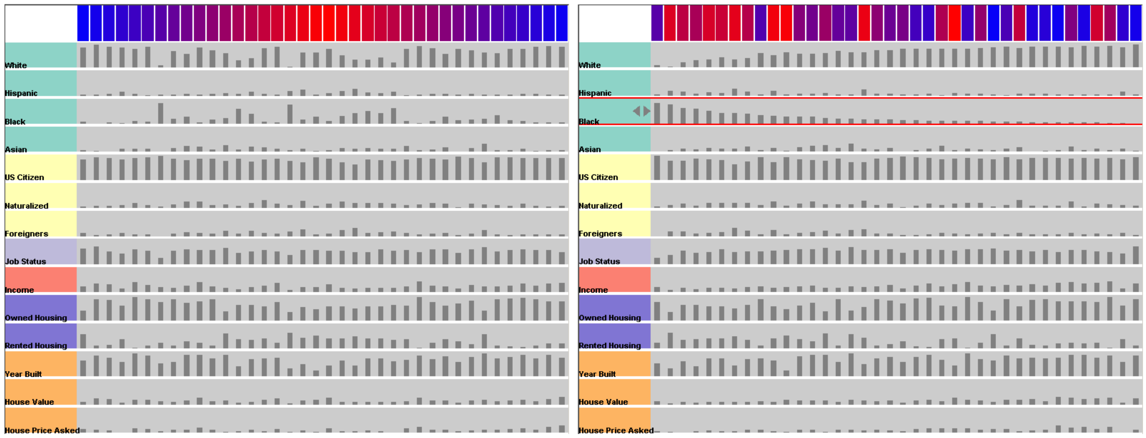
Figure 18: Different displays of the same data; (Left) Bar charts; (Middle) line charts; (Right) gradient grid charts.

the ground, the cluster sizes are larger (resulting in fewer overall clusters), allowing the user to see overviews of the entire area. When the sphere is lower to the ground, cluster sizes under the sphere are finer (resulting in more overall clusters), allowing the user to inspect a specific local region (Figure 16). The focus region is thus not a fixed area, but varies with distance from the focus point directly under the sphere. The degree of focus is shown on the buildings themselves as a color gradient from red to blue. These colors provide a link between the two views (see below), and give the user an indication of how narrow or wide the focus currently is.

The user can select and highlight any cluster by double-clicking on it (Figure 17a), and also view the urban model as individual buildings rather than clusters for a closer inspection of a neighborhood (Figure 17b).

2.2.2.2 Data View

The data view (Figure 15, left) consists of two parts that display the same information in different ways: a matrix panel and a parallel coordinates panel. Both show the data associated with buildings or building clusters relative to the position of the focus point. The examples in this project use demographic data from the 2000 Census, but any geographically linked data (e.g., traffic statistics, crime rates, etc.) could be shown.



(a) (b)

Figure 19: Sorting columns. a) normally, columns are sorted by the distances of their corresponding clusters to the focal point. The closer the clusters are to the focal point, the closer the column is to the middle of the screen (and more red in color); b) the user can also sort the columns based on a specific data dimension.

The top part of the data view can be switched between bar charts, line charts, or gradient charts (Figure 18). In any case, the view is organized in columns with each linked to a cluster. The columns are labeled with colors that correspond to cluster colors in the model view. The number of columns therefore changes dynamically with the number of clusters that are displayed as the user changes the level of detail or moves the focus around the city.

There are two ways to order the columns. Under normal use, the clusters closest to the focal point are drawn in the middle of the view (Figure 19a), which corresponds to the usual way the model view is used, i.e., the user will keep the focus close to the center of the view, and recenter if needed. The user can also sort the columns depending on the values of a selected category of data for quick identification of the clusters with the desired value ranges (Figure 19b).

Each row of the bar/line/gradient charts shows a specific dimension of the represented data. The graphs are color-coded to show groups of related categories, making quick identification and orientation easier. In Figure 19, there are 14 categories of data, separated into 6 different groups.

The bottom part of the data view shows the same data, but using parallel coordinates [87] to better show relationships between dimensions in the data. Like the matrix panel, the lines in the parallel coordinates view are color-coded to match the cluster colors, and the colors of the axes correspond to the colors of the rows in the matrix view.

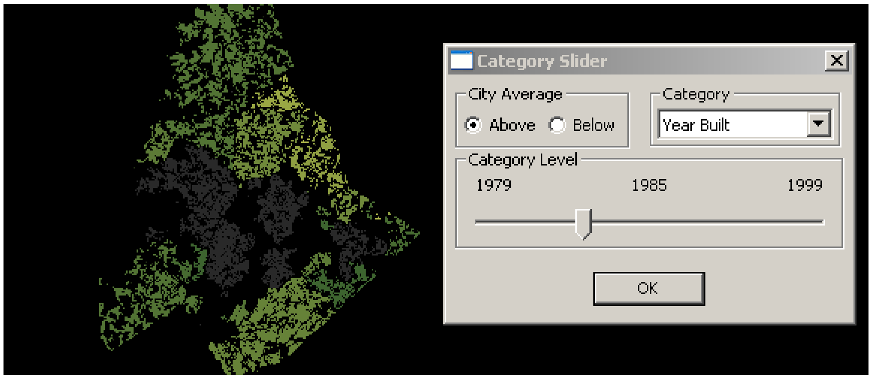


Figure 20: Using a slider to find buildings in the city that fit a specific criterion. In this example, only buildings that are built after 1985 are shown.

Although the two views depict the same data, we find that the different presentations of the data give the user different types of understanding. The matrix view shows the relationship between clusters of buildings that are close to each other. The user can thus quickly see the homogeneity of the neighborhoods around the focal point. Sorting the matrix by a data dimension can also reveal correlations between data properties.

The parallel coordinates view cannot show spatial relationships, but can easily reveal relationships between data dimensions, allowing the user to easily identify positive or negative correlations between categories.

2.2.2.3 Dimension Thresholding

The tools presented so far are mostly tailored towards the exploration of an urban model from the model view. For tasks that have specific search criteria, such as looking for areas with high percentages of certain ethnicities, we employ a simple slider to highlight the clusters that match the given criterion (Figure 20).

As the user moves the slider, the model and data views update interactively, highlighting the clusters that fulfill the criterion. To maintain legibility, the other clusters are shown as well, but in a darker color.

2.2.3 Application Scenarios

In order for the participants of our study to understand how the application might apply in real world settings, we provided them with scenarios in which a user might interact with the urban visualization tool. By using actual demographic data taken from the United States Census 2000 [174] for the county of Mecklenburg in Charlotte, North Carolina, we were able to apply the data to the 3D building layout of the area. The demographics utilized in this specific demonstration cover various categories such as ethnicity, citizenship, job status, income, and housing statistics. However, the system is not limited to these categories and is configurable to each user's needs.

A simple scenario provided to the user allowed for an immediate understanding of the possible everyday uses of the visualization tool. For instance, according to the Director of the UNC Charlotte Urban Institute, the city of Charlotte's annual "Charlotte Neighborhood Qualify of Life Study" looks for areas of high ethnic population with low levels of income to identify possible improvements to these regions through urban planning. Using our system, we can quickly identify the regions in Charlotte that fit the two criteria (Figure 21). Furthermore, upon further inspection, we identify that there are some characteristics of these neighborhoods that are of interest to the UNC Charlotte Urban Institute. Specifically, by examining the parallel-coordinates, we find that the level of Hispanic populations in these areas have a positive correlation with the percentage of foreigners and the percentage of people who rent housing (Figure 21). The relationships between these categories are not easily perceivable using current commercial software. As the Director of Land Use and Environmental Planning Division at UNC Charlotte said, "using current software requires going back and forth between ten different windows to find these relationships, whereas your system shows all those relationships in one simple view."

Another real life example was given by a real estate developer at Harris Associates. In his occupation, identifying areas with homogeneity in demographics is often very important when negotiating radical or new concepts in urban planning. In his experience, local governments of areas with high homogeneity in demographics are more likely to accept new concepts because of their shared demographic background. However, areas with high heterogeneity often result in disagreements between

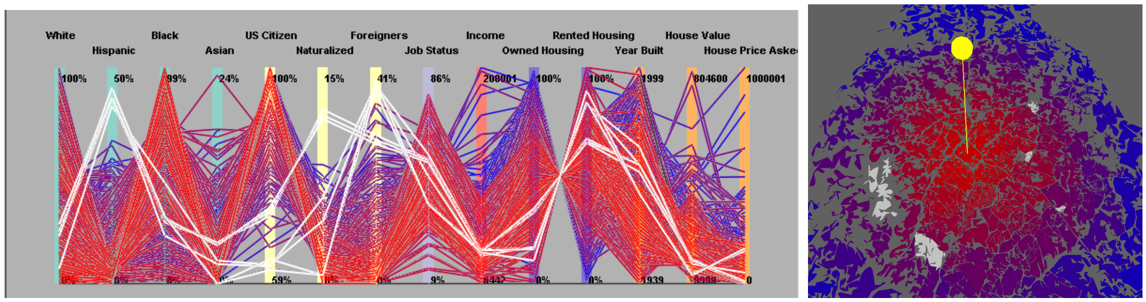


Figure 21: Case study 1: Finding neighborhoods with high Hispanic population near the downtown area: (right) the user starts by putting the focal point over the downtown region; (left) using brushing in the parallel coordinates window, the user highlights the regions that have high Hispanic population. Notice the positive correlation between the Hispanic population, the percentage of foreigners, and the percentage of residents who rent their housing in these selected areas.

the different demographic groups due to their differences in perspectives. Using our system, we allow the user to quickly identify the level of homogeneity of downtown Charlotte versus the town of Davidson (20 miles north of Charlotte) where the company is located (Figure 22). The figure indicates that downtown Charlotte is heterogeneous in demographics and citizenship status, whereas Davidson is much more homogeneous.

2.2.4 User Evaluation

We asked 14 expert users to evaluate our system from their own perspectives and identify the strengths and weaknesses of our system. These 14 experts have disparate backgrounds, ranging from independent real estate development, the Center for Real Estate at UNC Charlotte, the UNC Charlotte Urban Institute, Charlotte Mecklenburg County Geographic Information Systems Office, Planning Department, and School System. In the study, we first asked the experts to fill out a pre-test questionnaire that identified their backgrounds in urban studies and their proficiency levels with geographic information systems. Then we demonstrated features of our system, followed by a few simple scenarios in finding interesting characteristics of the census data in Charlotte. After the demonstrations, we asked the experts if our tool could be used in their areas of expertise. We concluded by asking them to give feedback on the usefulness of the system as well as any suggestions for future improvements. With their consent, the users were tape-recorded during these sessions. Furthermore, all experts agreed to have their comments, names, and affiliations appear in this

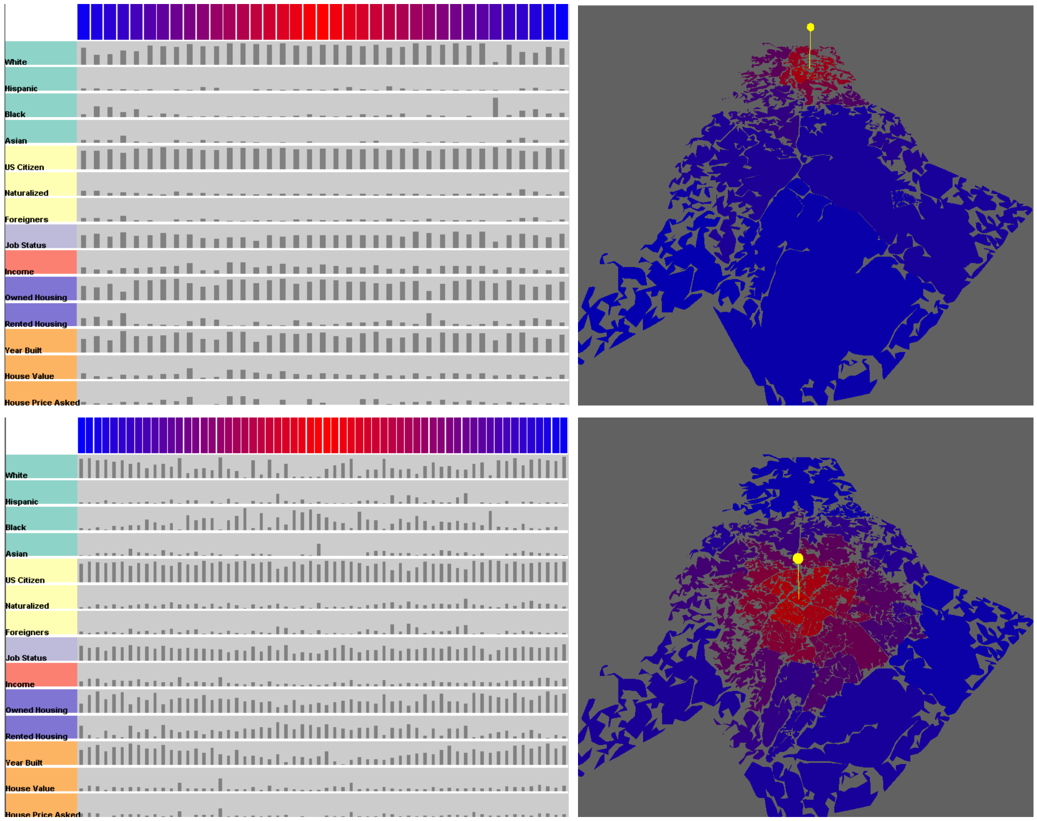


Figure 22: Case study 2: Showing the differences in the amount of homogeneity between downtown Charlotte (bottom) and the Davidson area. (top) Davidson area: notice that the bars in the Data View are all approximately the same height around the red clusters, indicating that the neighborhoods in the Davidson area tend to be more homogenized, whereas (b) in the downtown area of Charlotte, the differences between surrounding neighborhoods are more apparent.

publication.

2.2.4.1 Focus-Dependent and Dynamic Clusters

All but one of the experts agreed that the focus-dependent view with dynamic clusters helps in understanding not just the region of interest, but also its surrounding areas. A planning specialist of Charlotte Mecklenburg School System commented that using this technique would allow her to focus on the potential sites of a new school, and still show the "projections of future student populations based on surrounding new residential developments." Another planning expert continued to add, "new housing developments often impact the existing school systems in terms of student population and demographics" and implied that the dynamic clustering helped in visualizing the changes and seeing the potential new effects.

The one user who didn't find this technique useful commented that most projects he worked on had strict boundary requirements. With these restrictions, it had never been necessary for him to examine surrounding areas.

2.2.4.2 Integrated Displays

On the use of the two integrated views between the 3D Model View and the Data View, 13 out of 14 of the participants found the combination of the two to be useful. A senior systems analyst at Charlotte Mecklenburg County GIS Office attested that "[the integrated displays] are an asset in handling large amounts of data and faster user production rates because it provides a link between the 3D urban model and the data display." As the Director of the UNC Charlotte Urban Institute succinctly put it, the dual views provided the "here's what I'm looking for, and there's where it is" capability to understanding urban data relationships.

The only expert who did not find the integrated displays to be useful mentioned that GIS experts had been successful for years in using single displays and that having multiple screens sometimes caused confusion regarding on which screen to focus.

2.2.4.3 Multi-Dimension Visualization

Displaying multiple dimensions of data using the matrix view and the parallel coordinates allow the user to quickly see relationships within the urban data. A principal planner at the Charlotte Mecklenburg Planning Department asserted that "an asset to this visualization tool is that the selection of data makes demographic relationships instantly apparent in the area of specificity. In current systems, you have to design [the necessary queries], analyze them, and then modify the queries to find the correlations that you are looking for." The director of Mecklenburg County GIS Services further added, "Sometimes users have to go through a lot of different sources of data or running [statistical analysis] to find relationships. Your tool is providing an on-the-fly, interactive way of instantly noticing nearby statistical data and their relationships." All of our users shared the same sentiments and found the Data View to be useful, although one user commented that the Data View required some explanation before the relationships in the data became apparent.

The Director of Land Use and Environmental Planning Division at UNC Charlotte summarized the strengths of our system eloquently. “Essentially what you are providing with this tool is a spatially sensitive graphic display. The strength of this tool is the dynamic table that displays areas in a spatially understandable way. In other software systems, the user is required to scan the tabularly listed rows of a GIS database, which gives no indication of the rows’ geospatial locations or their relationships between one another. Another strong aspect is the fact that your focus area and peripheral areas are cohesively orientated. When that aspect is combined with the ability to change the level of detail through clustering, the user gains a new dimension [of understanding]. Changing the level of detail in other software programs becomes cumbersome from running [multiple repetitive] queries.”

Collectively, expert users saw the potentials of our system both academically and commercially. Academically, they recognized that our system offered an entirely new perspective in studying urban landscapes and felt the tool provided them with vivid mental maps of their own spatial awareness in an urban environment. Commercially, users believed that this system can help increase productivity and provide better execution of their daily tasks by substantially improving the way they interact with GIS databases.

2.2.5 Discussion

A few very interesting and important topics emerged from the discussions with our expert users. The most controversial is the results of dynamic clustering, on which we received informative but disparate feedbacks. There are a few experts who found the clustering to be confusing. Specifically, experts who work with the census data expected the clusters to follow the boundaries of the census tracts. Similarly, experts who work with zip codes or school districts wanted to see the clusters form boundaries and shapes that they are familiar with. In contrast, other experts praised the results of the dynamic clusters as the clusters provided “possibilities to new district boundaries” that one might not have been aware of.

While all experts agree that they retain spatial understanding of the city of Charlotte through all levels of simplification using our system, the question of what makes an urban model legible remains open. The one key point that everyone can can agree on is that the sense of legibility is very subjective and changes depending on each individual's level of understanding and perspective of a city. As mentioned before, urban experts with specific domain knowledge form their sense of legibility based on their domain of expertise; residents of a city orient themselves based on familiar establishments (such as a local restaurant) that might not be visually or mentally important to others. For tourists or visitors to a city, visually distinctive landmarks such as skyscrapers or major roads are important features for understanding the surroundings. Conversely, soldiers in an urban battlefield require a different set of training and understanding of a city to effectively communicate spatial relationships in a dynamic environment [109].

Although our approach uses geometry to create clusters that are understandable to all users, we note that creating *legible* cities to users of all backgrounds is not a trivial task and would require knowledge of the user's perspective of the city prior to creating the clusters. Finding interactive methods so that these clusters could be tailored towards the need of each individual user remains an important future direction of research for us.

Another topic of discussion that was commented on by almost all experts is the need to integrate data from different sources. For real estate developers, seeing how commercial buildings and public establishments such as drug stores and schools intertwine with residential neighborhoods is important in identifying the needs of the neighborhoods. Members of the Charlotte Mecklenburg GIS Office noted that seeing tax records of individual buildings on top of the Census data would give better understanding of the economic development of the city. Similarly, experts from the Charlotte Mecklenburg School system would like to see Census information blended with crime statistics to find better routes for students and school buses. Since our system performs clustering on a per-building level, assigning specific properties from different data types to each building is trivial. However, finding the necessary data and identifying the best way to represent sometimes conflicting sources of data requires more investigation. For example, commercial districts without any residents would not have any census information, and purely residential districts would not contain information on economic growth of the area. Integrating these two orthogonal sources of data into a cohesive view is important in enhancing the user's ability in seeing the patterns and relationships between the data.

Along the same lines, many users noted that seeing temporal changes in a city would be very interesting. While we agree that time is a very relevant factor in urban visualization, the challenge lies in the collection of data such that the 3D models of buildings and the additional sources of data match both spatially and temporally.

Lastly, some experts suggested a potential use of this system outside of our original design goals. Members the UNC Charlotte Urban Institute mentioned using our system as a tool to compare different cities. Specifically, urbanists have widely accepted that Charlotte as an emerging Southern city has mimicked the growth of Atlanta due to their similarities in locale and culture. It would be interesting to juxtapose the two cities in our system and see if such patterns of similarities are apparent. On a local scale, independent real estate developers from Harris Associates mentioned a similar use of our system. Namely, it is important for developers to foresee pockets of potential growth in a city. For any given developing region, if a developer can identify another similar but already established region in the city, the developer might be able to project the potential growths of the developing region based on the history of the established one. Although the idea of using our system as a predictive tool is still being investigated, we are very excited about the potential benefits that it could bring.

2.3 Probe-based Urban Visualization

A similarity across the majority of GIS applications and geospatial visualizations is the singularity of the viewing perspective. For example, in map-based visualizations, the user is generally restricted to viewing one region of the map at a particular zoom-level. When zoomed out to see the entire extent of the dataset, local trends and anomalies, which are often of interest, become suppressed and ultimately lost in the global picture, especially as the scale of the dataset increases. To discover and inspect these local details, the user must zoom in to a level at which they become visible. However, by doing so, one loses both the global overview and context of the local region. This both limits the user's spatial awareness and prohibits comparison between distant local regions.

In the model presented within this section, coordinated information visualizations are integrated directly within the main geospatial visualization. User defined regions-of-interest are linked to each coordinated visualization, delineating which data is presented in each visualization. Furthermore, these interfaces, which we call probes, allow the user to interact directly with the geospatial data within the regions-of-interest as well. By using multiple probes, the user can simultaneously observe and interact with many different local regions across the entire range of scales (ranging from global to the smallest individual units) without losing spatial awareness.

To illustrate the general usefulness of our probe concept for enhancing geospatial visualizations, we incorporate it within three unique existing applications. First we apply probes within a 3D geographic information system (GIS) environment used to visually explore the changes (new buildings, etc) detected (using aerial laser range-finding) to a urban area between years. The second application we augment is designed for visual analysis of census data across large urban areas. Finally, we create a new, entirely probe-based interface for an agent-based social simulation that models the various factions and behaviors of an entire country. In each case one can see benefits including uninterrupted spatial awareness, improved inspection and comparison capabilities, ability to view data at multiple scales simultaneously, and increased potential for collaboration among multiple users. These common benefits are then more elaborately discussed in a more general context within Section 2.3.4.

However, we note that the addition of probes to interactive urban visualizations could cause additional strain on the rendering process. In section 2.3.5, we discuss how probes affect the interactivity of UrbanVis as well as the potential issues with allowing for open-ended interactive techniques such as probes. Although these shortcomings do not outweigh the benefits that we have observed during our evaluation with experts in GIS and architecture, they are nonetheless important considerations in designing interactive urban visualizations.

2.3.1 Related Work

Donelson's [40] Spatial Data Management System presents a large projected display of a 2D graphical information space. The interface is two-handed, supporting panning and zooming. Two joysticks, a tablet, and two secondary monitors that are touch sensitive are provided. One monitor displays a "world view" of the entire information space along with a "you-are-here" rectangle which provides visual context for the user as he views a particular 2D region on the large display. The other monitor, the "key maps monitor," shows auxiliary information such as a chapter outline when the main screen displays text files, or a time-line when the main screen displays video.

Furnas [54] describes generalized fisheye views. In the spatial domain, the metaphor is a fisheye camera lens that shows higher detailed, less distorted imagery toward the center of the field-of-view and less detailed, compressed imagery toward the outer field-ofview. In addition to the geospatial example of Steinberg's famous *New Yorker* cover, "A View of the World from Ninth Avenue," Furnas presents experimental studies showing that people's concepts of complex non-spatial structures also exhibit fisheye character. Furnas presents Degree of Interest functions to describe fisheye display of information for both spatial and non-spatial data. He also acknowledges the significance in geospatial contexts of supporting multiple foci in fisheye views. He gives an example of a person who has lived in multiple states whose mental map of the geography is fisheye in character but with foci at each location in which he has lived. In the context of non-interactive cartography, Kadmon and Shlomi present a mathematical approach for such multi-focal map rendering [95].

More recently, Furnas [55] focuses not on the variations of geometric distortion, but on the different degree-of-interest (DOI) functions and how these determine what information is and is not included in the display. He discusses how this concept can be carried to non-visual domains as well.

Bier et al. [15, 14] present the Toolglass and Magic Lens, a see-through 2D, two-handed GUI interface. The Toolglass and Magic Lens are see-though windows whose positions are controlled by the user's second hand with a trackball+wheel device. The user's first hand controls a regular mouse and pointer. Graphical filters in a toolglass can be overlaid on other objects to reveal alternate visual representations while the mouse cursor continues to allow direct manipulation of the objects through the Toolglass. Bier et al. cite earlier works with similar concepts of filters for changing information in visualized systems but these earlier works lacked the metaphor of a movable lens. Viega et al. [177] extend the concept of Magic Lens to 3D including both flat, planar lenses and volumetric lenses.

Perlin and Fox [128] introduce the zoomable 2D Pad interface. This interface includes portal filters, which show "non-literal views of cooperating objects." For instance, when a portal filter is positioned over tabular data, within the portal a bar chart could be displayed.

Our concept of probes relates to this prior work as follows. We start with a View+Closeup [55] implementation of the Focus+Context metaphor. However, the user can define, place and scale multiple regions in the view for which the Closeup windows, or insets, are generated. The interactive manipulation of the regions-of-interest (ROI) boundaries and the fact that the view geometry within the ROI are drawn in an specialized manner borrows from Magic Lens and portal filters. However, while Magic Lens or portal filters just present an alternate rendering of the selected geometry in the main view, with probes an additional inset window displays secondary representations of the selected data. Unlike a standard View+Closeup inset in cartography, this inset is typically an alternate 2D InfoVis representation of the data in the ROI. Further, the inset window can contain interactive controls that affect the ROI, and the inset's InfoVis graphic supports linked brushing. Compared to a Toolglass, a probe inset pane with interactive controls decouples the ROI from the location of the controls. Significantly, probes are more than just labeled push-pins found in physical and digital 3D maps such as Google Earth [67]. Push-pins are not areal and labels are not dynamically varying InfoVis displays with optional 2D GUI controls. Commercial GIS tools such as ArcGIS [49] provide map views, tabular views and various basic graphing capabilities but it is not possible to interactively tie a multiplicity of these latter two view types to a multiplicity of ROIs on the map view.

2.3.2 Probes

The main building blocks of our design are probes. We define a probe as a pair consisting of a user-defined region-of-interest and a pane containing any variety of information visualizations coordinated to depict and interact with the data within that region-of-interest. The region-of-interest and the visualization pane are linked either directly (e.g. by a line) or indirectly (e.g. the region-of-interest and the pane's background are shaded the same color).

To create a probe, the user selects a region-of-interest (e.g. specifying a central focal point and extent radius, or through manual selection for irregularly shaped regions) and then chooses a location for the visualization pane to be overlaid directly within the main geospatial visualization. Once created, a probe visually presents a focused, local view into the dataset/model along with an intuitive visual linkage back to the overall dataset/model.

2.3.3 Applications

We begin by presenting the results of integrating our probe concept into three tested and published applications. In each case, we describe the limitations of the original application, and discuss the benefits gained by applying the probe concept.

It should be noted that inserting our "on-demand" probes within an application will never remove or limit existing capabilities and functionality, but always adds benefits such as extending beyond a single perspective, adding multi-focus and multi-scale inspection and interaction, and increased potential for collaborative use.

2.3.3.1 Urban Change in a 3D GIS Environment

The first application we integrate probes within is a 3D GIS visualization. This primary function of this application is to detect changes such as construction, deforestation, etc. in an urban environment between annual aerial LIDAR scans [21] (Figure 23). Aside from the primary 3D GIS view, a heatmap is presented on the side to depict the global distribution of the changes (in height and area) across the entire urban environment. Filtering is allowed on the heatmap, which controls the visibility of changes in the 3D view based on their area and height measurements.

Similar to most traditional GIS applications, this visualization allows for a single perspective that is directly tied to the viewable screen area. When the user zooms into a small region, it is difficult to maintain the global overview and context as the single perspective limits the users spatial awareness. Conversely, as the user zooms out, local details become suppressed and difficult to see. Furthermore, since the heatmap is tied directly to the user's perspective, there is no easy way to compare the trends and patterns of two or more regions without saving the images to file and comparing them offline.

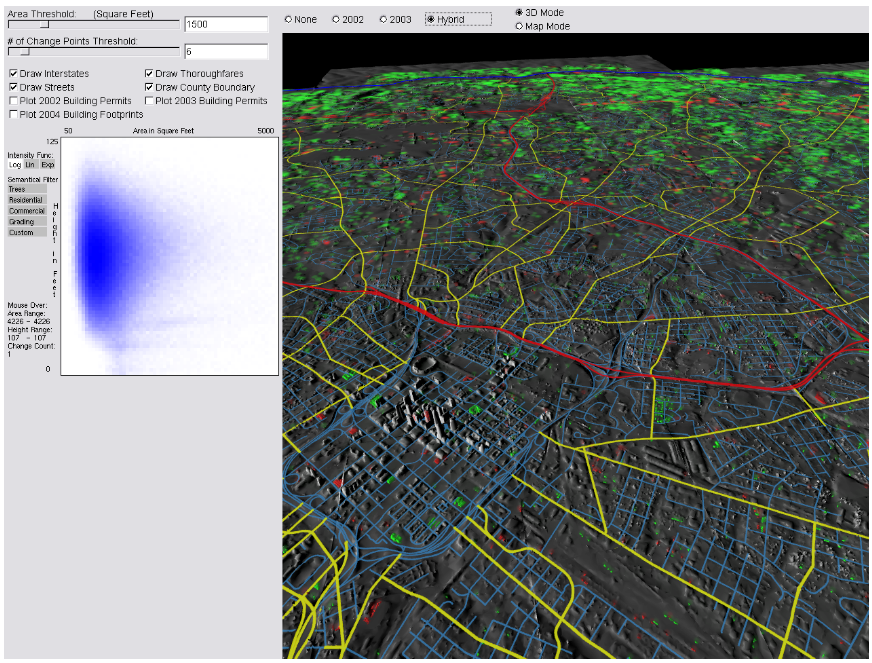


Figure 23: Shown here is the original interface for the urban LIDAR change application. The main window (right) presents a 3D fly-around view of the county. To the side, a heatmap (upper left) shows the global distribution of all changes across the entire county. It is a density-shaded scatter plot with the vertical axis tied to the heights of change models and the horizontal axis tied to the projected 2D areas of change models. Different types of changes (e.g. newly built houses) generally fall in predictable areas of the heatmap. By selecting regions within the heat map, the user can filter the changes presented in the main window.

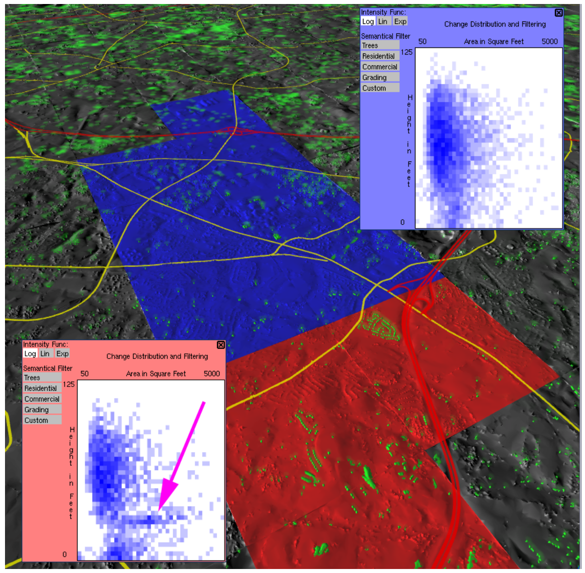


Figure 24: Shown here, the user has selected two regions-of-interest, the blue is a commercially zoned district, and the red mostly residential. The visualizations for each probe present a heat map showing the distribution (in height and area) of changes detected in the respective regions-of-interest. The magenta arrow points to a concentration of changes found only in the residential region. This region on the heat map can then be used as a global filter, revealing other similar new residential developments elsewhere.

Probes are introduced to this visualization to remedy these issues. A user defines a region-of-interest using a mouse, and a probe interface appears directly within the 3D view. Within the probe interface is the heat map visualization, now showing the distribution of only those changes within the region-of-interest. Also present are the filtering controls; here again their domain switches from global to local filtering. Multiple probes can be added on the same display, and they are differentiated based on the colors of the probes and the highlights of the regions-of-interest (Figure 24).

In the scenario presented in Figure 24, the user selects two regions-of-interest on the terrain. The first region, shown in blue, consists primarily of commercial buildings. The second region, shown in red, is a partially rural area that contains a number of new residential developments under construction at the time. It is clearly visible that the distributions in these two regions are different by examining their corresponding heatmaps. The magenta arrow in Figure 24 shows a concentration of changes found only in the second, residential region. This region can then be used to filter the entire county, revealing all the similar new residential developments.

Even in this simple example, the power of the probe interface is apparent. The user can now examine regions from afar so as to maintain spatial awareness in relation to the surrounding regions. With the heatmaps displayed directly in the 3D view, the user can easily relate the abstract information visualizations with their corresponding spatial locations. More importantly, comparison between locations is now trivial as the heatmaps can be juxtaposed for immediate comparisons. This can be done either visually, as shown in Figure 24, or directly, by requesting a comparison pane of the two probes. By selecting two probes, a third pane can be requested, which simply presents a difference image of the two probes' heatmaps. In a previous paper [21], a manually created difference image was created outside the application to illustrate the differences in changes between two regions. The ability to do this type of comparison directly within the application is a powerful improvement. Further discussion of direct comparison abilities follows in Section 2.3.3.3.

2.3.3.2 Census Data Exploration Tool

UrbanVis [25] is an application designed to explore an urban environment and its corresponding census information by combining a 3D urban model view with an abstract information visualization view (Figure 15). With the use of the yellow sphere as control, the user can interactively navigate an urban environment and explore relationships between spatial and abstract information in a multi-resolution manner.

Unlike the LIDAR system described in the previous section, UrbanVis already separates the region of interest (as denoted by the yellow sphere) from the visible screen area. This view independence allows the user of UrbanVis to explore the urban model while retaining spatial awareness. However, similar to the LIDAR system, UrbanVis allows for only a single perspective and therefore cannot support comparison of different localized regions.

By applying the probe concept to UrbanVis, the user can now interact with multiple regions of interest in the 3D model view. As shown in Figure 25, each region of interest is now accompanied by an information panel exactly like the one shown on the left of Figure 15. The information panels can be moved around directly on the 3D model view but are always connected to the yellow spheres by a (white) line to maintain a clear relationship between the two. When two information panels are placed next to each other, the differences and similarities between the two local regions become

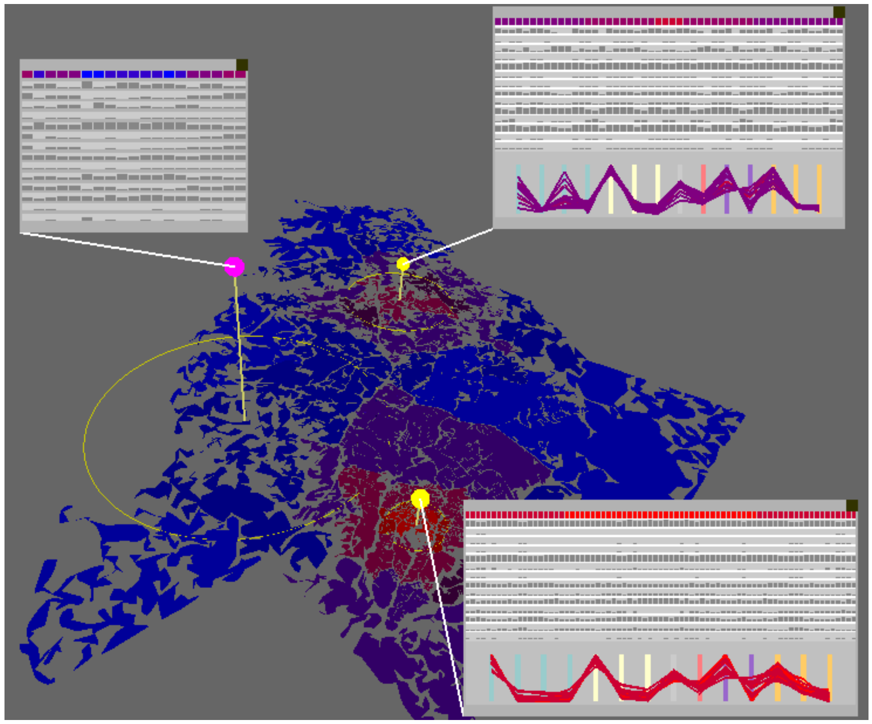


Figure 25: After modification to utilize probes, the user can now select multiple, variable sized regions-of-interest within the 3D view. Each region-of-interest is then linked to a resizable version of the original coordinated information visualization. By resizing the panels for each probe, the user can control the granularity/abstraction of the depiction of the data from the probe. Resizing is extended further in Figure 26.

apparent.

It is easy to see the comparison capability gained from using probes in UrbanVis; the user can now compare multiple local regions simultaneously within the application, without a navigational burden. However, another subtle but relevant advantage is that the resizable information panel allows the user to "annotate" regions using small information panels (Figure 26). These small information panels now act like glyphs in that they give an aggregated, high-level overview of the selected regions of interest without taking up much screen real estate.

2.3.3.3 Agent-based Social Simulation

Our agent-based simulation and visualization tool is created to visualize the results of a live agent-based simulation that allows a user to experiment with different social theories and scenarios in Afghanistan. Like the two visualizations described in Section 2.3.3.1 and Section 2.3.3.2, we apply the probe concept to an existing visualization of the agent-based system (Figure 27). However,

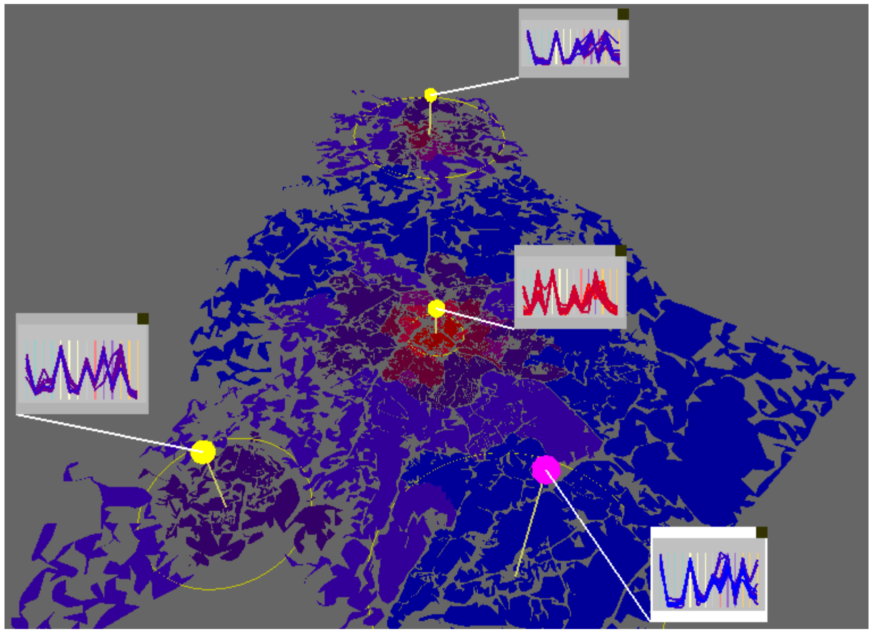


Figure 26: Shown here, the coordinated visualizations for each probe have been limited to solely the parallel coordinates view and resized to the point where each shows only the most general view of the associated data. Here the probes begin to resemble “flags” stuck in the map, giving a simple representation, allowing for quick visual comparison. (Assuming the user knows how to interpret them.)

unlike the previous two visualizations, the introduction of probes transformed the agent-based tool nearly completely. Like the original LIDAR system, the agent-based tool is also limited to a single perspective that is tied to the viewable map area. Similarly, the additional InfoVis views in the agent-based tool such as the bar chart and the time-series view are also tied to this single perspective. However, unlike the LIDAR system or UrbanVis, the main purpose of the agent-based tool is for the user to manipulate variables within the simulation and visualize the effects of the changes. Most of these variables are global in that they affect the simulation of the entire country, some are tied to fixed single locations or a specific regions. It is clear that without proper organization, an exponential number of controls are needed to capture all combinations of all the variables. In fact, Figure 6 shows some of the 150+ sliders that were needed to operate a few relatively simple social theories.

In addition to the issue of over-crowding from excessive sliders shown in Figure 27, the agent-based tool suffers another equally severe interaction issue, in that the sliders offer no spatial context in terms of their relationships to the corresponding geographical regions. Users and observers of this

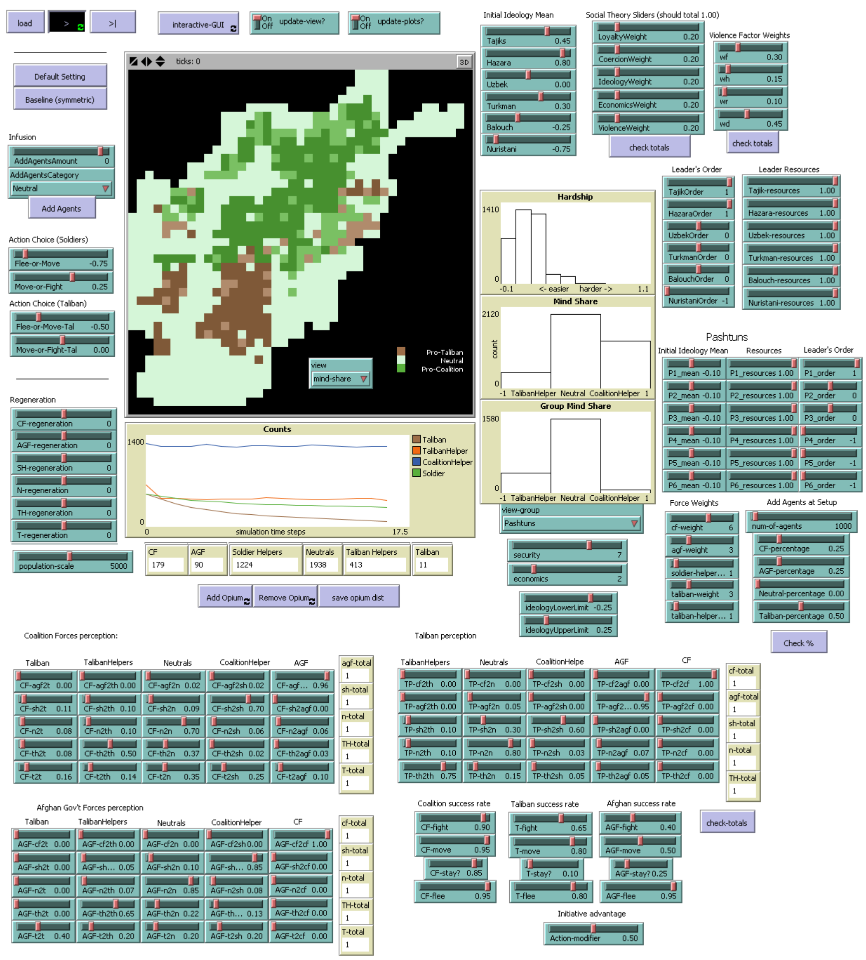


Figure 27: The agent-based social simulation's original interface. Notice the large portion of screen-space dedicated to sliders and other control elements, which are ambiguous in terms of their scope. The single map view allows for only one variable to be seen at a time. Likewise, the four graphs can only depict global statistics across the entire simulation.

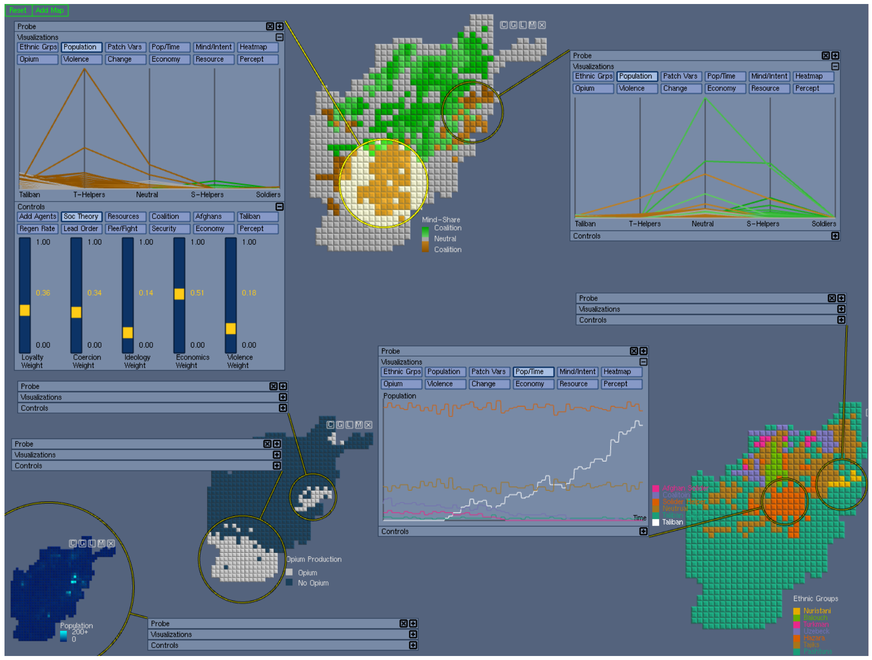


Figure 28: An example workspace in our new, probe-based interface. Notice that the user can add any number of different overview maps. Probes can then be inserted into these maps, spawning linked coordinated visualization/interaction panes. This extends observation and interaction across all levels, from global to individual cells.

agent-based simulation are often left wondering what slider to operate in order to affect a specific region of interest. This incongruity between the visualization and its controls greatly diminishes the effectiveness of the simulation as an experimental platform for testing social theories. By applying our probe concept, we can greatly increase the effectiveness of the interface. As can be seen in Figure 28, multiple instances of maps are now allowed, with each map colored based on a particular dimension in the data (e.g., ethnic group, loyalty, etc.). However, most importantly, the 150+ sliders can now be replaced by an "on-demand" tabbed control panel of sliders directly associated with each probe (Figure 29). This combination of sliders with geo-located probes makes the effect of each slider intuitive and obvious, in that interaction with a slider now only (locally) effects the region tied to its corresponding probe (i.e. whatever portion of the simulation the user has circled). It should be noted that the original, global controls can be replicated by simply creating a region-of-interest encompassing the entire simulation, as shown in the bottom-left of Figure 28. The implication of a visualization that has capabilities for both passive inspection and active manipulation is striking. As shown in Figure 29, the user has selected two nested regions to test the impact of an increased

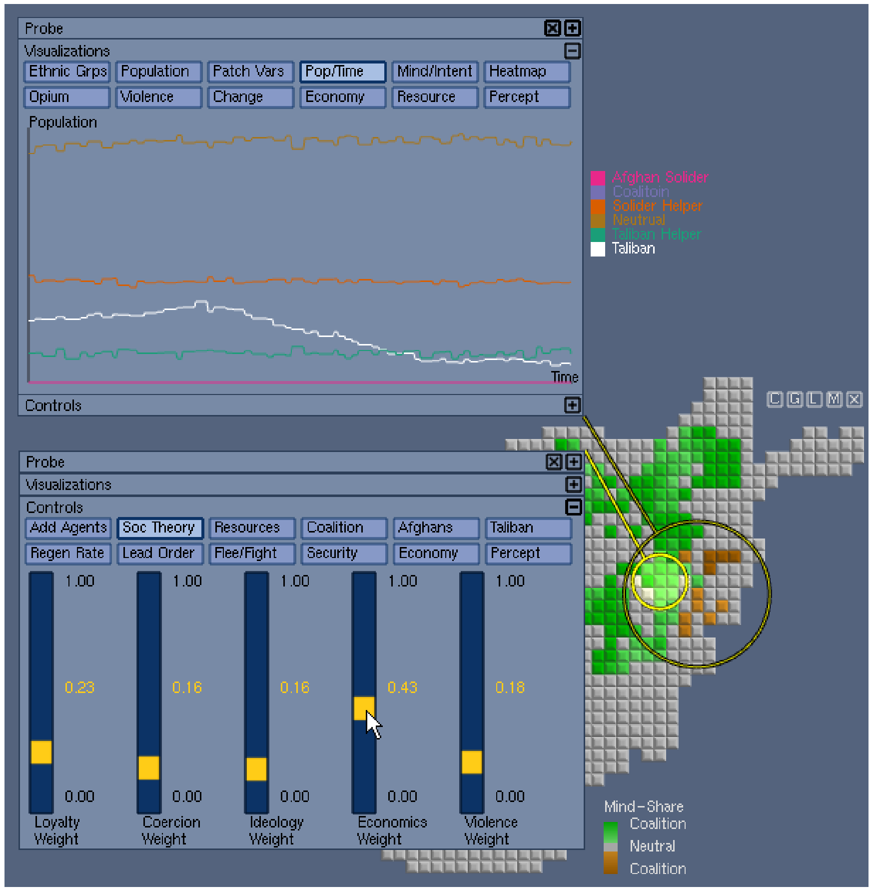


Figure 29: The use of localized control capabilities is shown in this scenario. Here, the user places a probe (the smaller circle) over the city of Kabul and expands the control section of that probe's interface and manipulates local variables to test out a new social theory within the city limits. A second probe (the larger circle) has been added to encompass the surrounding region, which has some pockets of Taliban loyalists (brown cells). This probe is setup to graph the relative populations of various factions over time. The user can easily see that after the new social theory is enacted within Kabul, the number of Taliban agents (white line) in the surrounding decreases.

economy in a small selected region and its effect in the surrounding areas. With the probe interfaces, the user can directly modify the economy of the small selected region and observe its effects in the probe associated with the surrounding areas. In this example, it appears that as the residents of the selected small region increase in wealth, the population of Taliban agents diminishes in the surrounding area.

A common task when testing social theories is to directly compare two regions-of-interest. With the probe interface this task becomes trivial. We are not limited to strictly visual comparison, but instead allow the user to directly compare multiple regions-of-interest through the use of a

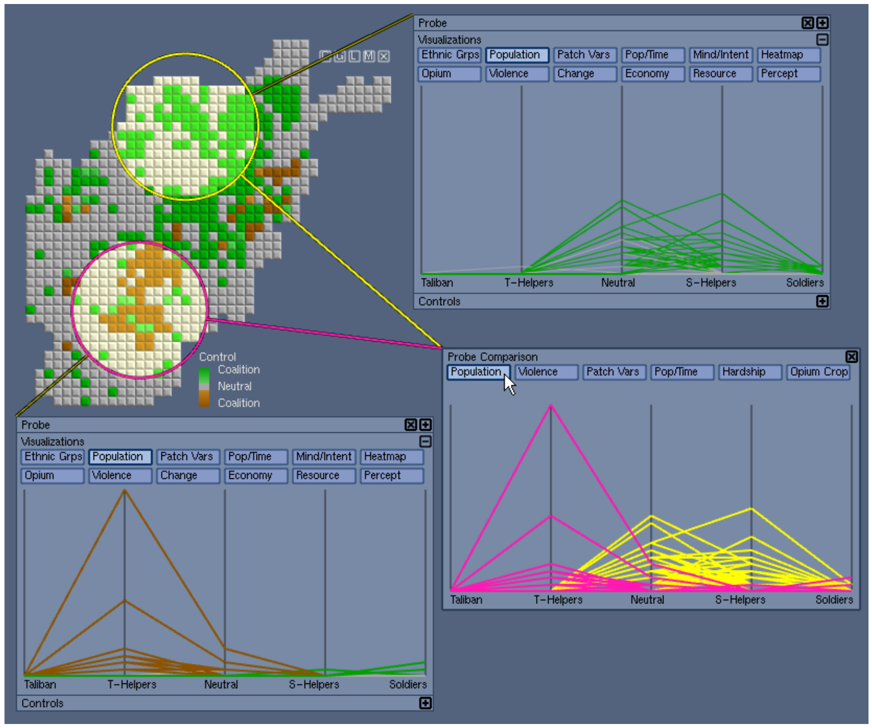


Figure 30: Shown here, the user has created two probes, one over a Taliban-controlled area (magenta circle/brown cells) and one over a Coalition-controlled area (yellow circle/green cells). Each is set to display the relative populations of each type of agent using parallel coordinates. Then by choosing to create a direct comparison (bottom right pane) of the two probes, the user can see the values from each region-of-interest together in a single visualization.

comparison pane. As shown in Figure 30, a comparison pane can be created between any two existing probes to visualize the relationships between the two regions-of-interest there are tied to. In this example, the user has selected a “union” operation, combining the two selected regions into a single view, immediately revealing the differences in population characteristics, while preserving (using color) the distinction between the two data sources. Numerous possible operations are possible within this framework, including the previously mentioned difference image of two heatmaps and the intersection between overlapping regions-of-interest.

The most obvious benefit gained by adding probes to this application is the ability to inspect multiple local regions at once. Where the original fixed coordinated visualizations once reflected the global model’s values, we now have any number of dynamically created visualizations each able to reflect the values in regions-of-interest of every size and shape imaginable. We can now provide superior comparison capabilities by directly presenting regions-of-interest together or against each other in their own visualizations.

The probe interface also allows for geospatial-based manipulation of the simulation and visualization; the user gains the freedom to choose regions-of-interest of any size and shape and interact with their properties directly, allowing for easy experimentation of complex social theories and immediate visualization of their effects. Finally, by replacing the static interface with a unrestrained workspace, enabling and encouraging the user to add and remove "on-demand" interface elements as needed, we not only remove the original clutter and wasted screen space, but also extend the single-user application into one that has potential to support multiple users.

2.3.4 Evaluation

We performed informal evaluations of our probe-enhanced applications to solicit feedback regarding the usefulness over the previous interfaces. Each of the three systems were presented first in their original form, and then with probes to two audiences, each with a mixture of both faculty members and graduate students. The first group consisted of thirteen participants from the Center for Applied GIScience at UNC Charlotte, while the second group consisted of eight participants from the College of Architecture at UNC Charlotte. A few participants had previous experience designing and working with the original UrbanVis.

2.3.4.1 View Independence

Using probes removes the burden of having to change zoom-levels to inspect local data. By preserving the global view, we ensure that the user always perceives the overall dataset. Visually depicting the selected regions-of-interest directly on the global view ensures that the user always knows the context of the local region. By using multiple probes, distant local regions can now be simultaneously inspected and directly compared onscreen, alongside the global view. This preserves maximum spatial awareness and decreases the navigation required to switch between zoom-levels.

Many participants identified the issue of loss of spatial awareness when constrained to a single perspective as one they have encountered in their work. Even though some of their existing applications have the ability to present multiple camera views (e.g. 3D modeling suites), one participant noted that "when trying to navigate in true 3D space, you often lose track of where something is in [the overall] space," while another elaborated that "where there's a lot of data... its important to be able to drill down" but that "sometimes you dive into detail on something and its not easy to navigate your way back out again." They saw our work as being a solution to this problem, in that "all [the probes] are organized by the overall metaphor of the map, so it really does help a lot [to know that] this window relates, in this way, to the other windows" and that this linkage "allows you to navigate more fluidly between different parts."

2.3.4.2 Multi-Focus Inspection

Probes allow the user to dynamically specify regions-of-interest and select from a wide variety of "drop-in" information visualizations. An assortment of methods are appropriate for selecting regions-of-interest: circular regions can be generated from a focal point and an extent, irregular regions can be selected manually unit-by-unit, etc. Extending the target of a coordinated visualization beyond what is merely onscreen at the time to these more flexible and precise regions improves both relevancy and accuracy. By allowing different visualizations to be tied to each probe, we can perform a wider range of inspections at one time than if we were limited to a traditional coordinated visualization interface.

By using multiple probes, the user can select multiple local regions and view their values side-by-side, or directly together in a comparison pane, always along with a global reference for overall context. This removes cognitive memory requirements, avoids change blindness, and speeds up comparison.

All participants appreciated the view independence and multi-focus aspects allowing them to access lower-level information about multiple local regions, while preserving the higher-level overview. The multiplicity of scales available for simultaneous visualization was also well received, with one participant specifically commenting that "having multiple scales is incredibly interesting, because at different scales you may be starting to visualize different processes."

The comparison abilities were also identified as attractive by the participants; one noted that having that capability in her application would make it "a lot easier to compare all my variables [while looking at it] quickly." Being able to investigate multiple regions without "having to go through the steps of selecting them and then opening up attribute tables" was described as "fast and intuitive."

2.3.4.3 Location-specific Manipulation

The creation of probes at a multitude of different shapes and sizes not only enhances inspection capabilities, but interaction capabilities as well. The user is no longer limited to applying adjustments and controls at specific predetermined scales. This extends once global controls into specific local regions, empowering the user to more precisely interact with the data.

Several participants expressed enthusiasm about our probe concept's potential to enhance their own existing projects with locally-tied interaction. Their projects included a landslide hazard analysis application, an interactive disease outbreak map, and a cellular urban growth simulation. The cellular urban growth simulation was the center of much discussion, as it had many parallels to the agent-based social simulation discussed in Section 2.3.3.3. In particular they saw the probes as an attractive method of being able to "change parts of the simulation and affect the simulation locally," and "a really exciting opportunity to take to the decision makers."

2.3.5 Caveats

An obvious but important issue, well know amongst spatial scientists and raised by several participants, that must be discussed in relation to this work is the modifiable areal unit problem (MAUP) [51]. Variations in how local areas are delineated can cause comparisons between the visualizations of their aggregated values to be misleading. A classic example is crime-mapping: while crimes are often recorded and reported per police beat, this region choice can be argued to be inferior and misleading compared to aggregating the reports by local neighborhoods with equal numbers of homes. Scale also plays a role in the MAUP, as local variation can be lost when aggregated into a larger region (a problem partially solved by our multi-scale probes) as well as misleading comparisons when comparing local regions of significantly different size (e.g. area or population.) In summary, often care must be given to how local regions are selected for aggregation, to ensure that the selections are meaningful, equal, and of similar scale; since we leave region selection to the user, a potential improvement to this work is providing assistance in selecting regions with similar characteristics for more accurate comparison.

There are possible scalability issues that may arise when probes are implemented within applications requiring significant processing to render their information visualizations. What may have been sufficiently fast to draw in a single inset view, may be too slow for deployment across multiple probes. This is especially true if the visualization requires extra calculation to aggregate information to condense itself to a smaller screen size. The UrbanVis application detailed in Section 2.3.3.2, for example, ran much slower under the strain of having to calculate multiple levels-of-detail and aggregations for each region-of-interest, something it was not originally designed to do efficiently.

Visual scalability can also quickly become an issue as the number of probes created increases, both in terms of screen real estate and overall cognitive load to the viewer. The two basic methods we used to help alleviate these issues were to make the probe interfaces collapsible (see Figure 28) and to make the probe interfaces resizable (see Figures 25 and 26). Collapsing a probe interface reduces the screen space needed to display it, maintains visibility of the region-of-interest, and reduces the overall visual complexity of the application. Resizing probe interfaces also achieves these benefits, and has the additional advantage of allowing the user to customize the complexity of the associated visualization. As shown in Figure 26, this can help with a shortage of screen real estate, as it permits the user to fit more, smaller visualizations onscreen at once. However, consideration must of course be made in regards to how the visualizations are resized downward into a more glyph-like form, in order to ensure that they are able to be correctly interpreted and meaningfully compared.

Another issue arises from overlapping regions of interest denoted by probes. This is particularly problematic and ambiguous if direct data manipulation is allowed on each probe, as is the case in the agent-based system. This overlap creates a one-to-many mapping issue, since there can be multiple controls affecting one area. There are some obvious solutions to alleviate this problem, such as prompting the user when a conflict arises. However, we find this problem to be more application and domain-specific, and effective solutions may be found on a case-by-case basis.

2.3.6 Discussion

Although in this paper we demonstrate the effectiveness of applying probe interfaces to geospatial visualizations, we believe that this concept can be applied to more abstract data spaces as well. The most obvious visualizations that can benefit from this are tools that present a spatial layout in which the locations of data items are of importance, such as in an organizational chart or graph layout. However, it is also conceivable that this type of interface can be extended to any information visualization that presents an overview that can be drilled into further. In theory, this probe-based interface should be very generalizable, and we look forward to exploring the possibility of applying this interface to other types of visualizations.

We remove fixed single perspective interfaces, and instead allow the user to dynamically insert interface elements anywhere they are needed. There is an immediate benefit of this style of interaction for collaboration, as there are no theoretical limits to the number of probes or map instances. Multiple users can interact with the same visualization at the same time without interfering with each other's views. An attractive interface device for deploying this kind of probe-based visualization is a multi-touch table, which has been demonstrated to be an effective medium for a multi-user environment. As the popularity of touch surfaces increases, we hope that our interface and its future extensions will be widely used and applied.

Brewer et al. [20] developed a prototype collaborative geovisualization environment and used it to perform interviews/informal evaluations with domain experts to ascertain what is expected/required of geospatial visualizations when they are to be used collaboratively. Some of their findings are particularly relevant to our work: the role of maps in a collaborative environment, drawing attention, and joint interface controls. Most of their participants mentioned that in a collaborative environment, the role of the maps was to provide the context in which discussion would take place. The importance of the map for conveying spatial characteristics and locations is vital when attempting to communicate a finding to a collaborator. Thus, our system succeeds in this aspect; results (in the form of visualizations) always have a direct visual link back to the map that provides spatial context. The need to draw attention to areas on the map (via circling or pointing) was also raised by most of their participants. Again, the way in which we link results back to their contextual locations explicitly draws attention from the results (the discussion topic) to the source on the map. Finally, while most participants saw the need for joint interface controls, they also raised the issue of potential conflicts. Solutions proposed included turn-based control and separate control panels for each user. A solution such as we show in Section 2.3.3.3, in which a single, global interface is replaced by on-demand controls, tied to local regions, has the potential to alleviate potential conflicts by allowing each user to manipulate variables for only their own specific regions-of-interest.

CHAPTER 3: VISUAL ANALYTICAL SYSTEMS

While the previous chapter demonstrates the necessity of performing feature-preserving simplification on large datasets for maintaining interactivity, a closer look at how interactivity is useful in a visualization is required for justifying the cost of altering (or removing from) the dataset.

Why is interaction useful to a visualization or visual analytical system? The intuition and consensus in the visualization and human computer interaction (HCI) communities are that interaction is the medium in which a user constructs, tests, refines, and shares knowledge, and that the interactive manipulation of visual elements is part of the analytical discourse in problem solving [131]. Unfortunately, most of what researchers in visualization and HCI understand about interaction is still based on unproven theories, observations, and conjectures [131]. Although there is a similar thread of research in the cognitive science community that investigates the relationship between interaction and problem solving through the study of distributed cognition [101], this research has yet to be directed at analyzing a user's interactions with a visualizations but has focused more broadly on people's interactions with the physical world.

The theoretical and cognitive understanding of interactions is beyond the scope of this thesis. The approach that we have taken instead is to design appropriate interactions, integrate them tightly into visual analytical systems, and study the effects of the combination. In cases when appropriate, we do directly compare highly interactive systems with those that are more static and cumbersome to use (such as the use of command-line interfaces). However, even in those cases, it is often insufficient to isolate the factors of interactions from the visual representations (e.g., section 3.4.4). What the proper methods are for evaluating the benefits of interactions still remains an open question that will be an important future direction for this line of research.

In this chapter, we focus on the design and implementation of interactive visualization systems. We present four successful systems that have demonstrated effectiveness in solving specific tasks.

The four presented systems have different foci both in terms of the types of task that the systems are designed to address and the techniques utilized in implementing the systems. What the four systems have in common is the fact that they are all highly interactive, and that the interactions are tightly coupled with the visual representations. The goal for presenting these four successful systems is to use them as case studies to further understand the role of interactions and visual representations from a design perspective, which will be presented later in the thesis (section 5.1).

The first system presented in this chapter is the WireVis system. WireVis is the result of a collaborative project with Bank of America where the task is to use visual analytical methods to identify suspicious financial wire transactions. Since Bank of America processes hundreds of thousands of wire transactions in a day, the visualization must be able to assist the user in sifting through large amounts of data quickly. The system is therefore designed with visual representations that display overall trends and patterns, and allow the user to interactively drill down into specific transactions or accounts for more detailed inspection. Much like the other systems presented in this chapter, it utilizes a multiple-coordinated view approach in which multiple windows show different aspects of the same data, and a user's interactions with one window are immediately reflected in the others.

The second system is a visualization that was specifically designed to examine the Global Terrorism Database (GTD). The GTD is a database of terrorism activities around the world that took place between 1970 and 1997. Although extensive, the events recorded in the GTD are not always complete. The dataset therefore contains a wide range of missing data or uncertain information. The challenge of the visualization is therefore to allow the user to search and explore the records for both conditions in which the user has detailed or little knowledge of the events. The system that we designed for the GTD is based on the 5 W's common to investigations (who, what, where, when, and why) so that five visualization panels are created to depict each of the W's. The user can interactively select, filter, zoom, and in general operate on any one of the panels and see the other panels react to the interaction. Using this interface, the user can input as much information as he is aware of (e.g., the range of time, the general geographical locations, or the weapons used in the attacks), and identify the other W's corresponding to the selection.

The third system presented in this chapter is a visualization management system that aims to allow a user to explore and compare biomechanical motion data. The motivation for this system is that biomechanical motion is inherently difficult to examine when presented as an animation sequence. On the other hand, if the motion is represented purely as an information visualization (e.g., an x-y plot where the x axis represents time and the y axis represents joint angle change), there is no way of knowing the motion of the 3D model that generates such a visualization. In the system presented in this section, the 3D model visualization window (animation sequence) is linked to multiple 2D information visualization windows. Multiple interactions are designed for both types of visual representations (both 3D and InfoVis) such that a user could zoom in a specific motion segment, or juxtapose two or more motion sequences and compare them in side-by-side and overlapping manners.

The last system differs from the first three in that there is no specific dataset to examine. Instead, the goal of the iPCA system is to investigate the use of interactive visualizations to gain intuitive understanding of a difficult mathematical concept called principle component analysis (PCA). PCA is typically performed on high dimensional data to identify the most prominent eigenvectors that can be used to reproject and "describe" the original data more efficiently. However, even for many of the expert users of PCA, the relationship between the reprojected eigenspace and the original data space is often difficult to grasp. Although the relationship is easily derivable and describable mathematically, the intuitive understanding of the relationship between the two spaces often eludes the end user. The iPCA is a research project that tries to identify whether the use of interactive visualization can aide such an understanding. In iPCA, the user interacts with the data in either the eigenspace or the original data space and observes how the interaction affects the data in the alternate space. Through interaction, the user can gain understanding of the mathematics of PCA and utilize it better to solve problems.

Through these four systems, we look to demonstrate that interactivity is indeed useful for a variety of tasks using visualizations. When interactivity is properly integrated with visual representations, the user can perform analysis and gain understanding of the problem or the analytical process which are the foundations and goals of visualization and visual analytics.

3.1 WireVis: Financial Fraud Analysis

Large American banks handle hundreds of thousands to millions of wire transfers per day. While most of these transactions are perfectly legal, a small amount is performed as part of criminal endeavors such as money laundering. The enormous amount of generated activity and the unconstrained nature of the data make it very difficult to find these few instances among all the legitimate ones. At the same time, strict regulations require banks to spend considerable effort to find and report these activities, or face significant fines or even being shut down.

The problems faced by risk managers and fraud analysts are exacerbated by the fact that an increasing number of transactions are purely digital and often involve a web of financial institutions around the world. Thus a bank's wire transfers may come from and go to individuals or businesses who are not the bank's customers. Often the bank is just a middleman for transactions that originate in different countries. In these circumstances, banks may know little about the individuals or businesses involved other than what is in the transaction record. Yet they must still exercise due diligence in discovering and reporting suspicious activity. For a large financial institution, this means monitoring hundreds of thousands of transactions per day, then investigating possibly suspicious ones in depth at considerable expense (and risk, if the monitoring is not effective). The problem is overwhelming and growing worse.

Hierarchical interactive visual analysis with multiple linked views can effectively attack this problem because it is geared toward the visualization and interactive exploration of massive datasets, integrating multiple methods from various disciplines such as information visualization, human computer interaction, and statistics.

In this section, we present *WireVis*, a multiview approach that assists analysts in exploring large numbers of categorical, time-varying data containing wire transactions. Our method is highly interactive, and combines a keyword network view, a heatmap, a search-by-example tool, and a new visualization called *Strings and Beads*. These four views together fully depict the relationships among accounts, time, and keywords within the transactions, and present the user with a global overview of the data, providing the ability to aggregate and organize groups of transactions for better investigation and analysis and the ability to drill-down into and compare individual records. Although the examples and results in this paper concentrate on wire transaction data, the approach is general and applicable to any type of financial transaction data. This method should be effective for any keyword-based data, semi-structured or not, with varying but substantial levels of activity over time.

This work presents substantial qualitative advances over current practice in investigating financial transactions, which involves blind queries followed by painstaking analysis of spreadsheets.

- It provides an overview that scales to hundreds of thousands to millions of transactions over any desired length of time.
- It provides tightly integrated views that look at patterns of activity over time and over keywords for clusters, sub-clusters, and so on.
- It replaces blind queries with contextual exploration, clustering, reclustering, and in-place analysis.
- It introduces powerful search-by-example techniques.

3.1.1 Related Work

Currently, analysts use spreadsheets to look at large data tables of transactions. Spreadsheets support various operations on rows and columns and give a detailed account of the data. However, they are not effective at providing a clear overview of trends and correlations. To fill this gap, several works in the information visualization field have been proposed, such as TableLens [135] and data sheets [46], and are now part of commercial information visualization suites.

Similarly, heatmap visualizations or correlation matrices have been used successfully in various application domains [2] and are now shipped in several commercial tools such as Spotfire Decision Site [161]. In genomics, this metaphor has been used for the visualization of massive gene arrays [147] and provides a compact overview of the data as well as a drill-down capability for detailed information. It has also been used to visualize social network data [81] as well as co-activity graphs in the arena of software visualization [63]. In the latter, structural and temporal patterns could be exhibited on correlation matrices. Although the orderability of matrices can enhance user performance with regards to certain tasks [64], finding the appropriate order on a correlation matrix is a task-dependent question that will be receiving increasing interest [157].

The need to follow temporal patterns in transactional data suggests the use of time-series data visualizations. Time-series data capture measurable quantities that change over time, such as stock values, climate data, etc. The users need to find trends and identify patterns in these datasets. Several works have focused on the display of this type of data, such as DiskTrees and TimeTubes [29], and provided periodic views [98, 176] supporting pattern identification. Recently more efforts [84] have been put into querying such data directly through user interaction. However, transactional datasets such as wire transfers could be more challenging to these kinds of tools due to the sparsity of the data points, as most accounts would fire a transaction only once in a while.

Monitoring keyword-tagged transactions over time also suggests the use of corpora exploration and visualization tools such as ThemeRiver [77]. However, this work can only handle a small set of topics or keywords simultaneously, whereas wire data bear on tens or hundreds of corporate-defined keywords. Moreover, ThemeRiver is geared towards the display of overall trends and temporal changes in topics, rather than outliers and isolated behaviors, or, in other words, rare anomalous transactions.

Systems offering coordinated multiple views have arisen as a suitable solution for complex multi-faceted datasets. Recent works such as Snap-Together visualization have studied the feasibility of such systems [124], provided a taxonomy [122] and model [17] of multiple-views systems, and studied the benefits and tradeoffs of such systems from a user perspective [125]. Other authors provided guidelines [11] for using multiple-views systems for information visualization. In the current work, we set out building a coordinated-multiple-views system in view of capturing as many aspects of transactional datasets as desired.

Due to the size of the data involved, we have to store it in a commercial relational database management system. While many visualizations are reading data from databases, we are not aware of work that is taking advantage of a specifically-designed database scheme for interactive visualization of large data sets directly from a database. Polaris [163] builds queries into databases based on visual specifications of the attributes to be shown. Tioga [6] allows the rapid development of visualizations that can be used to explore and drill down into databases. DBVis [99] uses pixel-oriented techniques to show as much data as possible from a database on a computer screen.

3.1.2 Monitoring Wire Transactions

In collaboration with Bank of America, we have tackled the problem of monitoring wire transactions. As we describe the nature of the data and the current practice in monitoring wire transactions, we will shed light on the requirements of this problem and how interactive visual analysis can bring about a drastic improvement in financial investigation.

Normalizing data in the various fields of a wire transaction is difficult as these fields are frequently open to interpretation at the point of origination. Roughly speaking, a wire transaction corresponds to a certain amount of money sent by a payer to a payee via a chain of intermediaries, with or without additional comments or instructions (Figure 1). The payee could be the real beneficiary (e.g., a person or business) or his account holder, with additional information as to what to do with the money. The transferred amount could come from the sender's account directly or via a third party. It could also be sent to the receiver's account directly or handled through a third party. Other information such as the address of the sender and receiver, additional comments, and instructions may be appended. As a matter of fact, a wire transaction could be best seen as a semi-structured data record with numerous optional free text fields.

Currently, financial analysts probe wire transactions based on various considerations:

- Official rules governing which transactions must be reported, e.g., when the amount of money exceeds a certain threshold.
- A potential risk inherent to the transaction, e.g., when the destination corresponds to a high

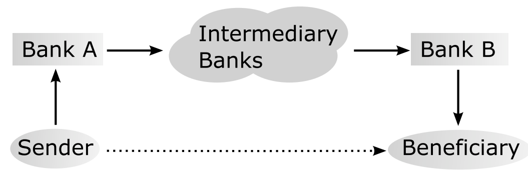


Figure 1: The flow of money and associated information in a wire transaction.

risk country or organization or when the transaction relates to a high risk activity.

While a set of filters can easily be set up to catch transactions matching a limited set of rules and automate a report generation process, risky or suspicious transactions are more evasive and uncertain. They are governed by an ever changing context where, for instance, geopolitical, economic and strategic motivations, and various actors are in play and where methods to hide illegal activity are constantly evolving as older methods are discovered and stymied. Hence, in the current practice, analysts query the transactional data over a time period, looking for certain keywords that may be indicative of high risk. Based on intelligence reports and previous analyses, investigators create a large list of keywords that best fit the international state of affairs. All transactions are filtered through this list of keywords, and the transactions that contain one or more keywords are displayed using a spreadsheet for investigation with transactions raising multiple red flags to be scrutinized more thoroughly. Additional information channels could then contribute to the investigation. For instance, home-grown expertise, the bank's own records, publicly available databases, and search engines could provide evidence for or against further action. An increase in false negatives would cast serious doubts on the financial institutions who would appear as purposely harboring fraudulent activities, not to mention having to pay large fines; whereas too many false positives could harm their relationships with their clients or irritate official agencies who would be wasting their time and resources on paranoid reports.

Adding to the complexity of the problem, the data includes one-time transactions in great numbers as well as repeated transactions. To hide from scrutiny, customers engaged in fraudulent activities could follow common temporal patterns of legal activity or try to break away from any fixed pattern. Fraudulent activity could also be distributed over a variable number of senders and receivers. At present, such distributed fraud is mostly beyond the reach of financial investigators. In fact, at

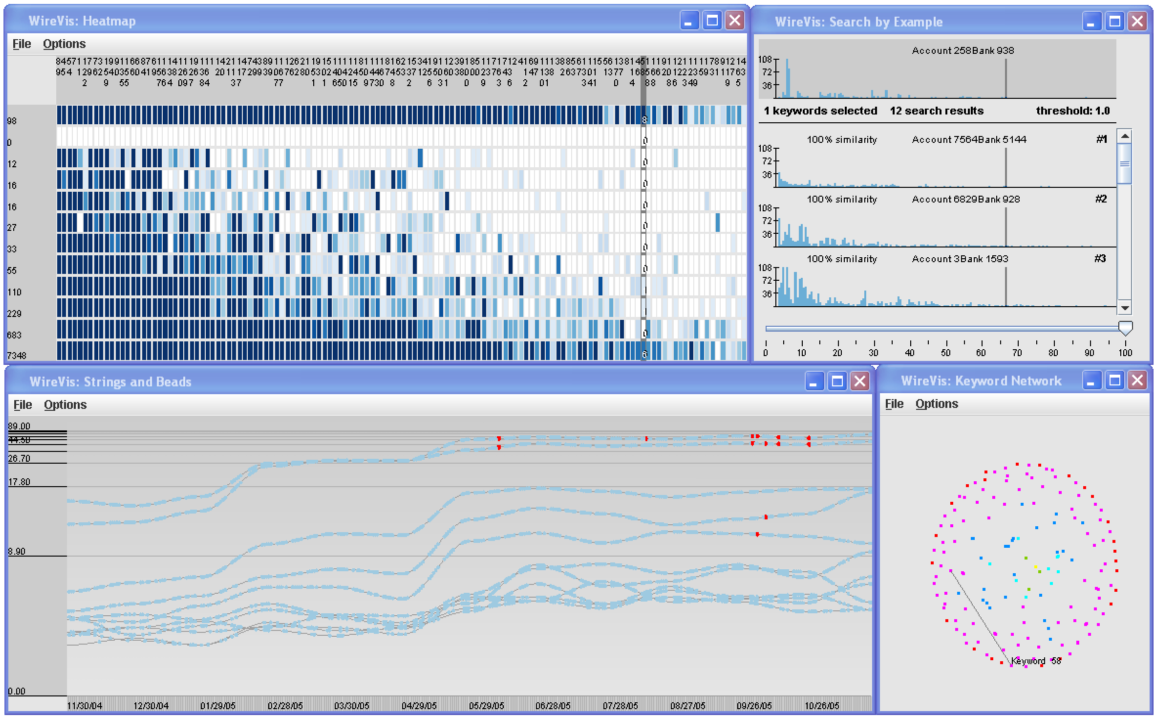


Figure 2: A view of the entire system showing the heatmap (top left), search by example (top right), keyword graph (lower right), and strings and beads (lower left).

present they do not have the capability to investigate all the patterns and activities they probably should. Effective analysis tools need to take into account this wide variety of scenarios and allow the user to see the transactions and patterns that match them.

In summary, financial analysts are faced with hundreds of thousands of transactions bearing predefined keywords over periods of time. This set of data is categorical due to the classification using keywords, and is by nature time-varying. However, the data are also semi-structured and certain records are unstructured. Furthermore, since most transactions contain a limited number of keywords, the data are therefore sparse when viewed as a relationship matrix between transactions and keywords.

3.1.3 Visual Analytics Tools for Monitoring Wire Transactions

WireVis uses four tightly coordinated views of transaction activity. The keyword network view is used to represent the relationships between keywords, the heatmap view shows relationships between accounts and keywords (see section 3.1.2), the search-by-example tool helps discover accounts of similar activities, and lastly, *Strings and Beads* depicts the transactions over time. All views rely on high interactivity along with the ability to see global trends and capabilities to drill-down into specific transaction records.

3.1.3.1 User-Centric Design

The design of WireVis was based on an analysis of the current work of fraud analysts with their existing tools. To manage the enormous amount of information, analysts first filter the data by geographic region using a set of specific keywords and other criteria (like amounts). This data is then inspected by hand, with additional tools like search engines to find out if businesses are legitimate, etc.

For WireVis, we wanted to keep as many of the working aspects of the existing system, while simultaneously enhancing the shortcomings. We therefore defined the following list of requirements for the system:

Interactivity. Despite the large amounts of data, WireVis must be highly interactive and respond to user input immediately.

Filtering. The current work method of filtering data using predefined keywords and other criteria must be kept to increase acceptance of the system. At the same time, we cannot rely on the data being filtered down, but must be able to show and work with all the data at once.

Overview and Detail. It must be possible to see aggregated views of all transactions of a day, week, or month in an overview, and then to drill down to the level of individual transactions when needed for the investigation.

Coordinated Multiple Views. No single view could fulfill all the requirements and show all the necessary data, so a system of coordinated views was designed that would allow the user to see different data, while being able to understand the connections between the views easily.

During the design phase of the project, we interviewed and communicated with members of the Risk Management, Compliance, and WireWatch (analysis) divisions of Bank of America on their current practices as well as their needs for monitoring fraudulent wire transfers. Furthermore, throughout the development phase, we maintained close communication with these groups and routinely showed them our progress and received feedback.

3.1.3.2 Data Aggregation

Since one cannot usually detect suspicious activity from single wire transactions, we need to visualize the activities of the corresponding accounts in order to detect suspicious behaviors. The transaction data are therefore first grouped according to the sending and receiving accounts, and the heatmap visualization tools show the accounts instead of each individual transaction. Even so, the number of accounts still range in the tens of thousands or more; thus, we hierarchically cluster the accounts. This provides the scalability needed, the high level overview that is imperative, and useful levels of abstraction. The complexity of the clustering algorithm is crucial because the analysts often need to perform reclustering as patterns in the transactions are discovered. Also, the clusters are often not optimal for the analyst's current purpose, so interactive reclustering has proved highly useful in exploratory visualization since it will provide much better clusters for further exploratory analysis [187]. Most existing clustering techniques such as k-means $O(kn)$ or single-link clustering $O(n^2)$ require minutes to hours to compute as n becomes large, which is unacceptable in our case.

Instead, we use a simple "binning" technique to find groupings of accounts based on frequency of keywords that occur in the transactions of the accounts. We treat each account as a point in k-dimensional space (where k is the number of keywords), and group the accounts based on their distances to the average point of all accounts. This method has the complexity of $O(3n)$ and can cluster tens of thousands of accounts in seconds. The wire analysts have approved this as a crude but effective way to explore the transaction space. The effectiveness is significantly enhanced because the method provides fast, dynamic overviews of the data while relying on the high interactivity and multiple views to assist in exploring the data space. Our procedure is to apply the hierarchical binning method as a preprocessing step, thus organizing all the data for a selected time period before launching interactive exploration. Then the user can use a strategy of selecting reasonably sized subsets for reclustering, thus maintaining interactivity. We have found that this strategy

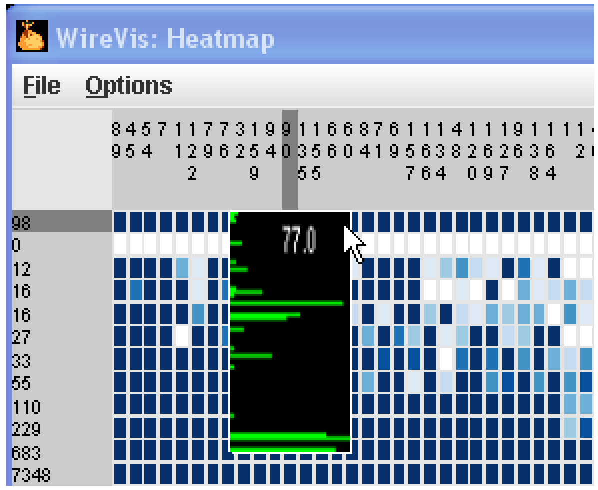


Figure 3: Heatmap with close-up view on a cell histogram displaying the number of keywords for each account. Each horizontal (green) line in the histogram represents an account in the cluster. The x-axis shows the number of “hits” of the keyword for that account. The purpose of showing the histogram is to combat the effect of using a cumulative sum in the Heatmap. By looking at the histogram, the user can quickly identify if the accounts in the same cluster contribute evenly to a specific keyword, or if there are abnormal distributions within the cluster.

works well in practice. An alternative could be to provide the user with a more exact but more time-consuming reclustering approach to be used at any point in the exploration. However, if this approach was not hierarchical, its value would be limited.

3.1.3.3 Keyword Network View

Depicting relationships between keywords is important for identifying questionable transactions. If a transaction contains two keywords that should not be related in the context of a wire transfer, it should be quickly identified and further inspected by an investigator. To show the relationships between keywords, we use a simple network graph as shown in Figure 4. A keyword is said to be related to another if both of them appear in the same transaction. The appearance of keywords in the same transaction forms the basis of the underlying relationship matrix in which the distances between the keywords are calculated based on the number of times that they appear together in transactions. In Figure 4, keywords closer to the center of the keyword network view are the most frequently appearing keywords, whereas keywords on the outskirts of the circle appear less frequently. When a keyword is highlighted, lines are drawn between the highlighted keyword to all relating keywords.

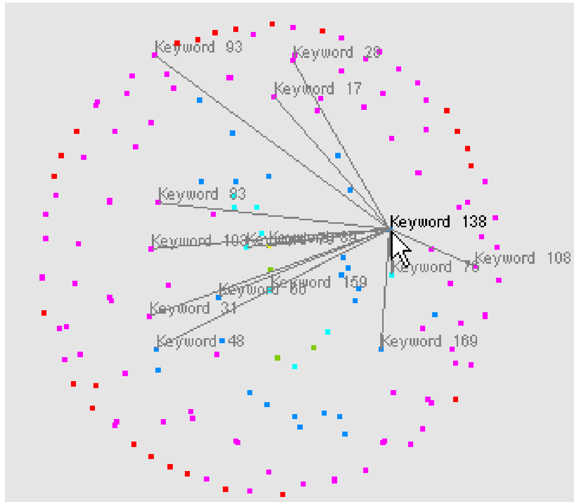


Figure 4: The keyword network view shows the relationship between keywords. The most frequent keywords appear in the middle of the view, while the less frequent ones appear on the outskirts of the circle. When a user highlights a specific keyword, lines are drawn from that keyword to all relating keywords.

3.1.3.4 Overview of Keyword-to-Account Relationships

We use a heatmap to display statistical measurements relating keywords (see section 3.1.2) and bank accounts, as the former occur in transactions involving the latter. Our heatmap uses a grid whose columns are the keywords of interest and whose rows are clusters of bank accounts being scrutinized. At the intersection of a given row and column, we color-code a value such as the number of hits for that keyword/column with regards to that account/row in the time-period encompassed in the data. Depending on the nature of the measurement displayed in the grid (e.g., sequential or diverging), various color schemes [19] can be applied to the visualization. We use a simple scheme where the saturation of the cell is proportional to the number of times a keyword appears for that set of transactions, since it is intuitively related to the heatmap concept. Then, the user can spot at a glance the accounts that are more frequently related to a given keyword or set of keywords (e.g., a high risk country heavily involved in money laundering). He can also see common keywords (full columns in the grid) which are likely to be filtered out in an investigative process. Likewise, accounts hitting all keywords (full rows in the grid) usually correspond to financial institutions rather than individual accounts and would also be deemed irrelevant to investigative work. Moreover, analysts can detect keywords displaying a similar activity, i.e. hitting the same accounts. In many cases, such coupling of keywords can be accounted for easily (e.g., Paris and France). In other situations, the coupling of two remotely related keywords will trigger further investigation. Therefore, the heatmap view makes it possible to visually compare patterns of behavior across different accounts. Our tools are enhanced with a user-configurable search-by-example capability (Section 3.1.3.7) that helps the analysts find accounts, respective to keywords, that are similar to a reference account. Search-by-example is a powerful tool for exploratory analysis, since it permits the user to quickly identify and search for behaviors of interest without having to specify those behaviors in detail.

Since analysts must try to grasp hundreds of thousands of transactions involving as many accounts, it is necessary that the heatmap be scalable in this dimension. Our tools perform clustering as described in Section 3.1.3.2, providing a high level abstraction of the data as a first overview. Then, the user can drill-down any set of clusters or keywords through direct interaction by selecting the desired subgrid or by expanding a cluster of his choice. The user can also hover on the heatmap and overlay the value associated to each cell. In case of aggregate data, the user can also overlay a glyph, e.g., a histogram, displaying the real distribution of low-level values that add up to the aggregate value as in Figure 3. By doing so, we strive to quickly provide the user with more detail at key points to enhance both exploration and enlightened decision-making.

3.1.3.5 The Strings and Beads Visualization

The ability to look for suspicious activities over a period of time is crucial to the analysts. To illustrate, when a terrorist attack took place somewhere in the Middle East, an intelligence agency had reasons to believe that the attack was supported and funded by individuals in the U.S. It requested bank wire analysts to search for wire transfers between the U.S. and the location of the attack. By looking for wire activities over a narrow range of time prior to the attack and on selected keywords, the analysts were eventually able to identify the culprit and report the incident. In addition to this event-driven scenario, it is also crucial to look at patterns of activity over periods of time (typically months), as these can bring up unusual behavior. Our coupled interface allows the user to quickly find these temporal patterns and activities.

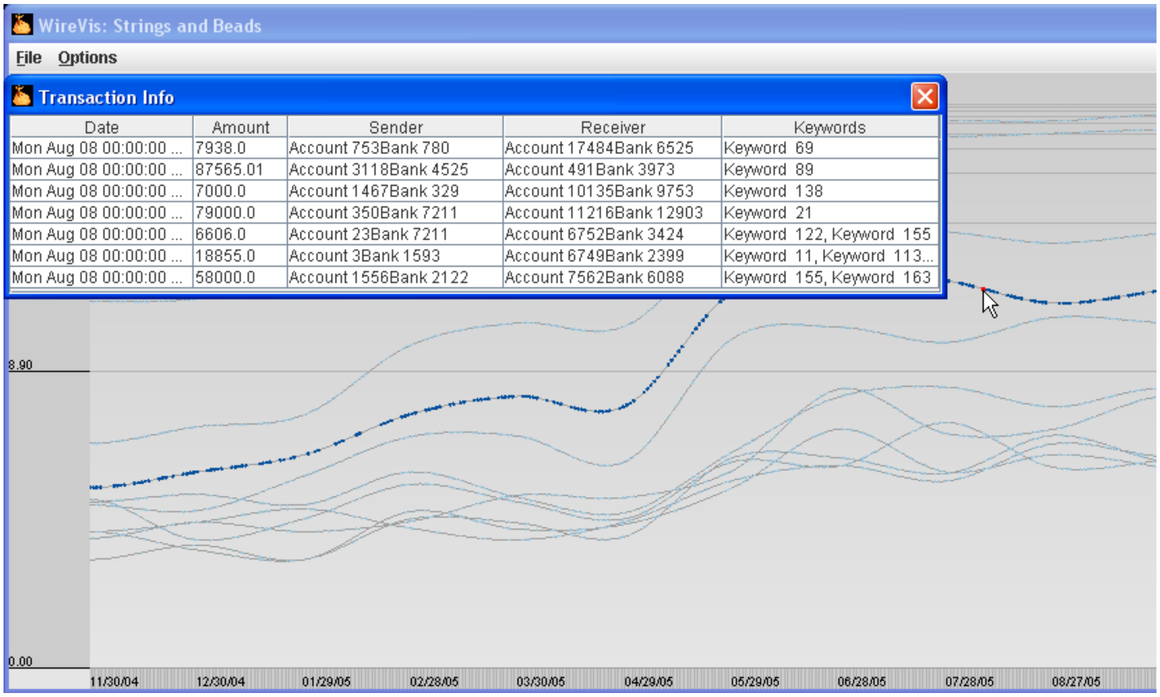


Figure 5: Double-clicking on a specific bead brings up the related wire transactions in a separate window. The user can interactively drill-down into individual transactions.

In order to support visualization of wire activity over time, we create the *Strings and Beads* view in which the *strings* refer to the accounts or cluster of accounts over time, and the *beads* refer to specific transactions on a given day. Together, the strings and the beads show the overall trends of the activities as well as the individual transactions. The x-axis of the view shows the progression of time, and the y-axis shows the "value" of the transaction, where value can be the amounts of the transactions, the frequency of activities, etc. Figure 5 shows that the Strings and Beads view is quite effective in giving an overview on top of showing specific detail. The strings shows the overall activities for selected accounts or clusters for an entire year, and the beads depict the details of a handful of transactions for that day.

Due to the fluctuation of the data, we choose to represent strings as splines instead of disjointed line segments. Since transactions do not take place over weekends or holidays and often vary drastically in amounts or frequency, representing the strings as line segments creates jagged lines, making it difficult to distinguish between different strings. Instead, we smooth out the strings as splines with the option to change the number of control points, which gives a good overview of trends over

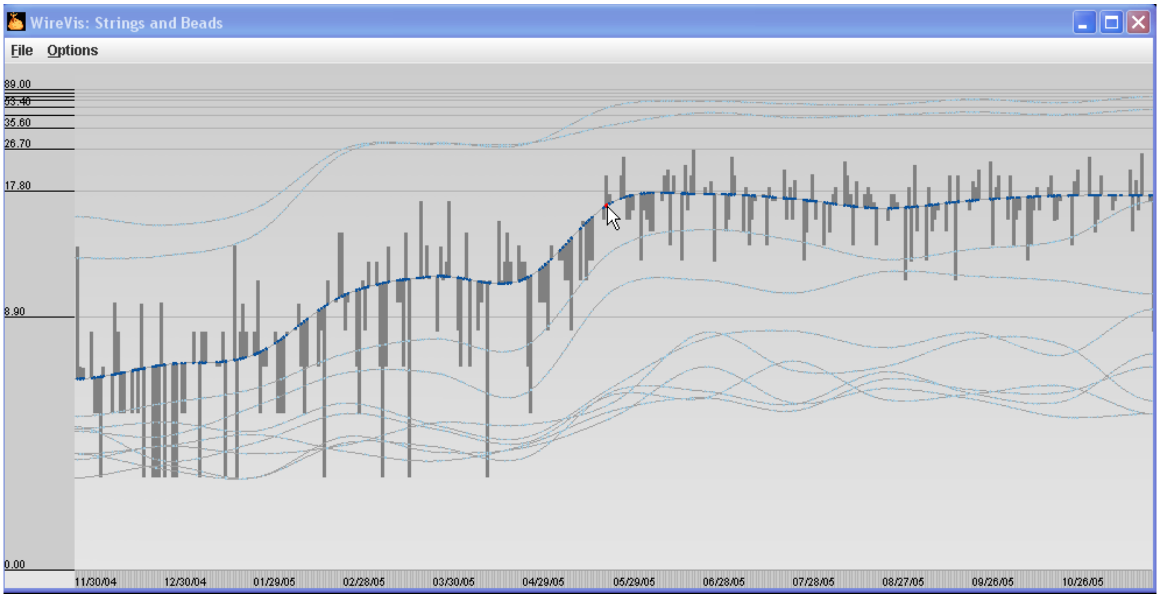


Figure 6: Turning on the option showing the original data, the user is able to see the real data points (shown as bars) on top of the smoothed String.

time. At any point, the user can display the original transaction values for detailed analysis, as shown in Figure 6.

To facilitate fast interactions with the Strings and Beads view, the analysts can quickly zoom in to the time period in question by brushing a range of time. To further examine the details of a specific wire transaction, the analysts can double-click on a bead to bring up the original wire information in a separate window as shown in Figure 5.

3.1.3.6 Coordination Between Views

The resulting clusters from binning provide a foundation for the coordination between the four views. Since the underlying data structure is the same, message passing between the views becomes trivial.

With the four views coordinated together so that an action performed in one view affects all other windows, the analysts can now interact with accounts, keywords, time, account values, etc. and see how the selections correlate in all dimensions. This is significantly more powerful than using the views separately.

For example, when an analyst hovers the mouse over the keyword names in the heatmap view, all

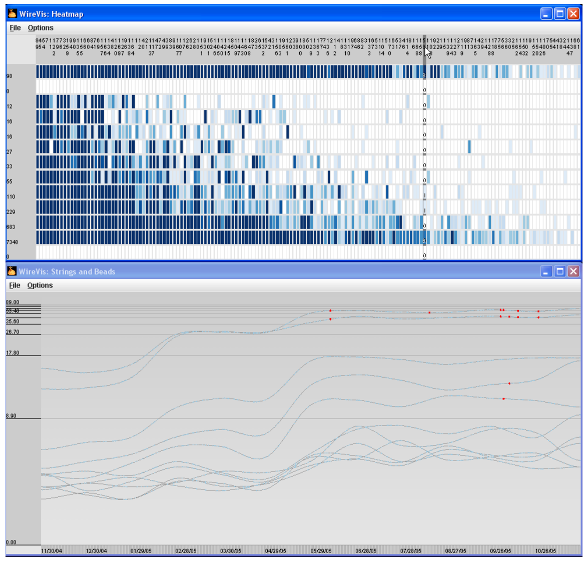


Figure 7: Highlighting keywords in the heatmap view shows the corresponding beads in the Strings and Beads view. The user can interactively see keyword occurrences over time.

cells in the heatmap are highlighted along with the number of occurrences of the keyword in each account cluster. At the same time, all the beads in Strings and Beads are highlighted to show when the transactions with such keyword occur (See Figure 7). This highlighting technique allows the analyst to search for suspicious keywords and see when these keywords occur over time.

Similarly, the analyst can hover over the dates in the Strings and Beads view, highlighting all the beads of a particular day. The heatmap then reacts by highlighting all the cells that contain the transactions of these beads and displaying the number of occurrences of the keyword for that day (See Figure 8). This allows the analyst to focus on specific dates and observe which accounts are transacting over what keywords over that period of time.

Such tight integrations occur throughout the keyword network view, the heatmap view, and the Strings and Beads views: selecting a string highlights a row in the heatmap, selecting a cell shows all the beads that contain such keyword, etc. All these quick actions permit rapid exploration over many accounts, keywords, and time ranges in terms of animated patterns at an almost subliminal level. The user can then pause, slow down, or go back to observe a pattern more closely.

Finally, both the heatmap and strings and beads view react to zooming (e.g., the cells in the heatmap view will only contain values over a certain time range when a user zooms into a time

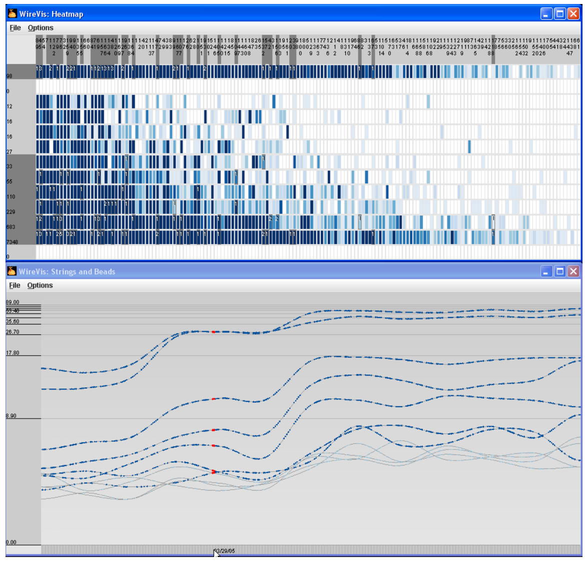


Figure 8: Hovering in the time axis in Strings and Beads highlights the beads of that particular day, as well as the related cells in the heatmap. The user can quickly browse through time and see keyword and account activity.

period in the Strings and Beads view), the analysts can see a global trend of the account activities over time by simple highlighting, but has the ability to further investigate specific incidences, events, and time ranges via zooming if necessary.

3.1.3.7 Search by Example

Because of the complex structure of the data and the observed patterns, it is difficult to define transaction patterns that one is looking for *a priori*. Once an interesting pattern is found, it is usually necessary to find not just transactions from and to the involved accounts, but also accounts that show similar activities.

An important feature of WireVis is therefore to search by example (see Figure 9). In a separate view, the user is shown the currently selected cluster or account to use as the prototype for a new search. Bars represent the number of hits for all the defined keywords, which the user can select as relevant criteria by clicking them. A slider is used to define the maximum difference for identifying a cluster as similar to the prototype. As the user moves the slider, the number of similar accounts grows or shrinks, giving the user a feeling for the space to explore. There is no separate search button; the search is performed whenever the user changes the criteria or the threshold, and results

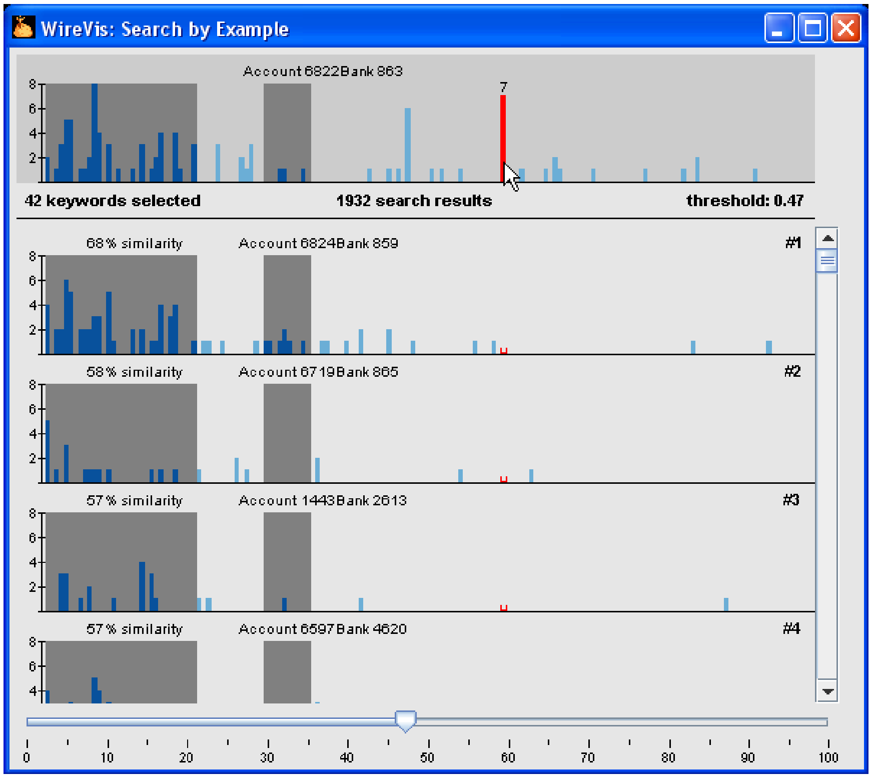


Figure 9: Search by Example. The user can select a prototype cluster/account as well as keywords of interest. The program shows all similar clusters according to a user-controlled similarity threshold.

are shown immediately.

The user can then select particular results, which are shown in other views for further investigation.

3.1.4 Connecting to a Database

The number of wire transactions processed by Bank of America ranges in the hundreds of thousands per day. Filtering the transactions using keywords reduces the number of transactions that require further examination by up to 90%, but the remaining 10% can still amount to tens of millions of transactions over the course of a year. Since this amount of data would not fit into the memory of a regular desktop computer, a strategy for storing and organizing the data to support interactive visual analysis becomes a critical aspect of the design process.

One method for organizing the data is to store pre-computed results of specific views in a hierarchical flat file structure. For example, the cells of the heatmap view for every possible cluster and keyword pair can be pre-computed and stored on file. However, it soon becomes evident that such a strategy is difficult to implement when both sub-selection of time and reclustering change the views and the cluster hierarchy. To create and maintain pre-computed results for all time ranges while retaining flexibility for reclustering in a flat file system would increase the storage requirement exponentially.

Instead, we store the data in a commercial relational database management system (RDBMS). The main advantage of using an RDBMS is the ability to scale the database as the amount of data increases. This can be done by fine tuning the RDBMS, adding more or better hardware, or distributing the database across multiple computers. On the flip side, connecting to an RDBMS over ODBC or JDBC using SQL queries has significant performance penalties when real-time interaction is the end goal. By storing and organizing wire transactions in an RDBMS, our design problem becomes minimizing the number of SQL queries sent to the RDBMS and the amount of data transferred for each query.

"Go to where the data is" is the motto that we follow when connecting the visualization component to the RDBMS. Conceptually, our goal is to perform all computations in the database, and only return enough results to the visualization component for it to render itself. Under this scheme, the visualization component is never aware of the wire transaction data (unless specifically requested by the user), but only receives enough information to create the visual elements.

3.1.4.1 Database Design

We create temporary tables for each visualization view (e.g. heatmap, Strings and Beads, etc). The tables contain information specifically for their corresponding views. For example, the temp table relating to the heatmap view has three main fields: cluster ID, keyword ID, and count, which is the information necessary for rendering the heatmap view.

During runtime, user interactions with the visualization components invoke stored procedures to perform data-related tasks such as binning/clustering, hierarchy traversal (drill downs). These procedures store the results of their computations in the temp tables, which are then fetched from the database into the visualization components to create the visual elements (Figure 10).

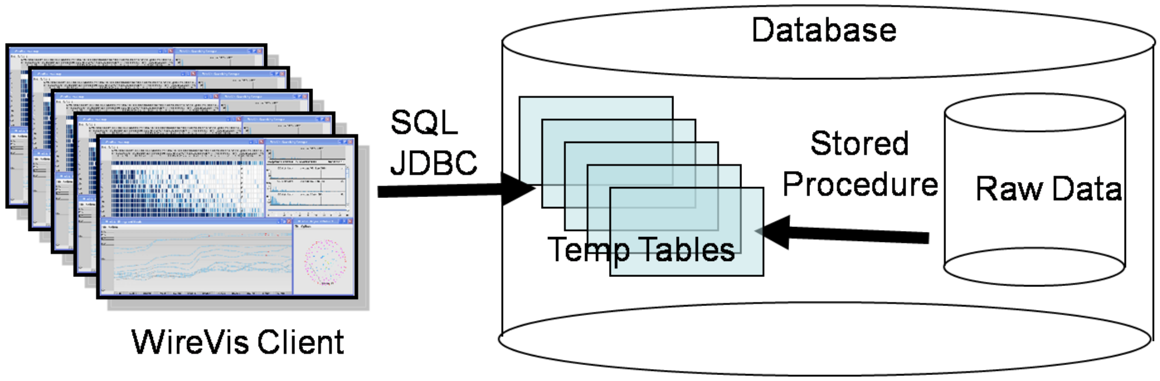


Figure 10: Connecting WireVis to a database. Instead of directly connecting to the raw transaction data via SQL queries, WireVis only receives enough information to render the visual elements. Stored procedures perform computations on the raw data and place the results in temp tables from which WireVis retrieves the information it needs.

3.1.4.2 Implementation

For our implementation, we use Microsoft SQL Server 2005 Express Edition as our relational database management system (RDBMS). Three tables are created to store the raw data, and three tables are used to organize it (Table 1).

Five temp tables are created to correspond to the three main visualization views (Table 2). Heatmap_Table contains information on rendering the Heatmap. Strings and Beads uses two temp tables, one for storing amount information, the other information regarding keywords and their frequencies. Transaction_Table is used to store results for querying specific transaction information (via double-clicking on a bead, see Figure 5). Lastly, the Keyword_Network_Table stores the positions of the keywords to create the network view.

A set of stored procedures is used to process the data in the data tables and insert them into the temp tables (Table 3). These stored procedures are invoked by the visualization front end when the user interacts with the system, and they store the results to the temp tables that are then retrieved by the visualization components.

Data in the temp tables is not easily updatable in our current design. When new data needs to be inserted into the database on a daily basis, the data would need to be processed overnight for it to be available the following day.

Table 1: Tables in the database. The first three tables are used to store the data, and the last three tables are used to organize the data for fast access. Note that the number of records shown in the last column are approximate for the reason of anonymity.

Table Name	Attributes	# Records
Accounts	Account_ID Account_Number Bank_Name	500k
Keywords	Keyword_ID Keyword_Text	250
Transaction	Transaction_ID) Sender Account_ID Receiver Account_ID Transaction_Date Transaction_Amount	7,000k
Account_Date_Keyword	Account_ID Keyword_ID Transaction_Date	7,500k
Trans_Account_Keyword	Transaction_ID Account_ID Keyword_ID Transaction_Date	21,000k
Trans_Keyword	Transaction_ID Keyword_ID Sender_Account_ID Receiver_Account_ID Transaction_Date Transaction_Amount	10,000k

3.1.4.3 Performance Results

We tested our database design on a desktop Windows computer with an Intel dual core 2.0 GHz processor, 1 Gigabyte of memory, and 80 Gb SATA hard drive. The database contains approximately 7 million records relating to one or more keywords over 13 months. In our experiments, both the database and the visualization components are run on the same computer.

Table 4 shows the amount of time required to perform each operation. The total number of records is approximately 7 million. However, as the user navigates lower into the hierarchy, the number of records used to perform each operation decreases. To demonstrate the scalability of the system, we show the performance of each operation using the full 7 million records, 3 million records (which could be the amount after one drill down), and 300 thousand records.

The user maintains real time interaction with the visualization front end during exploration. It

Table 2: Temp tables that correspond to the views in WireVis.

Table Name	Attributes
Heatmap_Table	Cluster_ID Keyword_ID Keyword_Count
SB_Table	Cluster_ID Date Total_Amount Num_Transactions Num_Keywords
SB_Data_Table	Cluster_ID Date Keyword_ID Keyword_Count
Transaction_Table	Cluster_ID Transaction_ID Date Amount Sender Account_ID Receiver Account_ID Keyword_String
Keyword_Network_Table	Keyword_ID X_Position Y_Position

is only when a cluster/recluster, a drill down, or a transaction request occurs that the user has to wait for the database to respond. The experiment was conducted on a mid-range computer with limited memory. With better hardware, we should be able to further decrease the response time of the database.

In addition, however, our performance testing revealed an unexpected result. The WireVis interface was found to be a quite efficient way to access the database and retrieve useful information in general. This is a potentially important result because our colleagues at Bank of America tell us that the database is notoriously hard to use with up to 80% of a user's time spent just making queries (and finding the appropriate information for use). One reason for this is that the user is "flying blind" and must plan and make many queries to probe the database for a given task. On the other hand, WireVis provides a highly interactive, exploratory capability for seeing information in context and getting further detail whenever needed. One uses this exploratory, probing capability *before* making specific queries, which can be launched from within WireVis. Since this process can

Table 3: Stored procedures that can be invoked by the visualization components via JDBC.

Stored Procedure	Description
Recluster	Clusters accounts at the top level
Cluster_Drilldown	Drills down into a cluster
Gen_Heatmap_View	Fills in the Heatmap_Table temp table
Gen_SB_View	Fills in the SB_Table and SB_Data_Table temp tables
Gen_Transaction_View	Fills in the Transaction_Table temp table

Table 4: Time required to execute each stored procedure (in seconds).

Stored Procedure	7 million	3 million	300 thousand
Recluster	57	30	8
Cluster_Drilldown	33	14	0
Gen_Heatmap_View	12	4	0
Gen_SB_View	12	4	0
Gen_Transaction_View	0	0	0

be performed so quickly, the user gets relevant information right away, not only information that she may specifically ask for but also related information that is discovered to be relevant for the task at hand. This is a general insight that will apply to other uses of databases such as this one, not just the uses described here. We plan to pursue this research path further in our ongoing work.

3.1.5 Case Studies

In order to assess the usefulness of our tools, we employed a sanitized dataset containing transactions sampled over twelve months. For privacy and proprietary reasons, account numbers, keywords and personal information have all been stripped.

While certain members of the Risk Management and Compliance groups have used the system, our expert evaluators at WireWatch were not able to do so. This is due to the fact that the WireWatch group is located in California and has specific security-related hardware and software restrictions that make the installation of our tool impossible in the scope of the first phase of the project. In order to receive feedback from these key collaborators, we asked James Price, Senior Vice President in Bank of America's Global Anti-Money Laundering organization, to observe video of interaction with our system during a teleconferencing session and provide his interpretation of the visualizations.

We categorize his observations into two groups: seeing normal, unsuspicious behavior and detecting activity that may indicate fraud.

3.1.5.1 Seeing Normal Behavior

The clustering provides an obvious separation between large corporations or financial institutions from small businesses and individuals. The first row of the heatmap contains accounts that transact in large amounts (as can be seen in the Strings and Beads view) in high frequency over a large range of keywords. These typically represent large institutions and can often be filtered out from consideration. The last row of the heatmap contain individuals that only have exactly one transaction over the course of the year (which can be verified by drilling-down into the sub-clusters), and can sometimes be filtered out because they might not contain sufficient indications of suspicious behaviors.

The keywords in the heatmap view are sorted based on their frequency in all transactions over all accounts. Thus, the most frequently occurring keyword appears furthest left in the view. From this view, one strategy could be, for example, to filter out the keywords that occur with expected frequencies, then recluster all accounts using the remaining keywords to provide a more focused overview of the activities.

3.1.5.2 Detecting Suspicious Activity

First, we identify a keyword that shows abnormal temporal patterns by hovering over the list of keywords in the heatmap view. During the hovering, we see in the Strings and Beads view that keyword 58 occurs only in the second half of the year as can be seen in Figure 11 (which is a detail of Figure 7). This peculiar time-based behavior prompts further investigation into the transactions involving that keyword.

By rubber-banding in the heatmap view and zooming into the column of keyword 58, we notice that not many transactions involve this word. Switching to the Strings and Beads view, we change the y-axis to show the amount of the transactions instead of the number (Figure 12).

There is a transaction near the end of the year of approximately three million dollars. This trans-

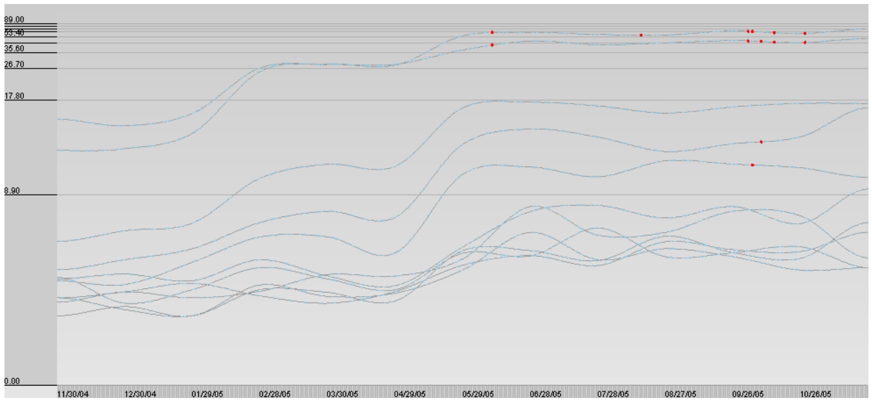


Figure 11: When brushing over keyword 58, we see that it only appears a few times, and all those are in the second half of the year.

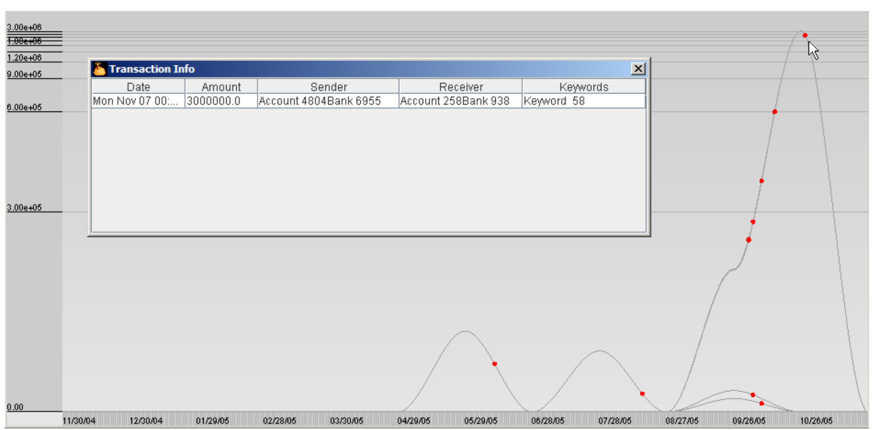


Figure 12: Changing the Y-axis to show the amount of the transactions and double-clicking on the highest bead shows the transaction details.

Figure 13: The receiving account's information in the search-by-example view shows that this is likely a bank or large company.

Figure 14: The sender, on the other hand, has only used this one keyword, and with a very high sum. Case study using sanitized real-world data. Fraud analysts found the combination of suspicious patterns sufficient to launch a full investigation.

action is peculiar because all other transactions involving this keyword have transaction amounts in the range of tens to hundreds of thousands of dollars. The fact that this particular transaction is of an amount much larger than others makes it stand out as an outlier.

Double-clicking on the bead representing that transaction reveals details about the transaction (Figure 13). Clicking on the receiver's name brings up the Search By Example window where we see that the receiving account most likely belongs to a bank or a large institution because of the number of transactions and the range of keywords it is involved in. More than likely, this large institutions is an intermediary handler of the transaction, and therefore is not of interest to the investigator. However, clicking on the sender shows that this account has had only one transaction over the past year (Figure 14). Not only is this transaction of a very large amount, but it also involves a keyword showing an abnormal temporal pattern. Although there is no single attribute of this transaction that would warrant an investigation into the origination account, the combination of all facets of this transaction leads to further investigation.

3.1.6 Discussion

By providing exploratory tools for the very specific data of wire transfers, we enable the experts to take a very analytical but still much less constrained approach than using other tools. While WireVis does not provide many of the other methods currently used in fraud detection, its tools are complementary and useful.

Considering that the only tools the analysts currently have are lists of text, they find the use of visual metaphors to identify questionable behaviors very interesting and promising. However, the WireWatch group has asked to be able to install and use the system on live data for further validation. We understand that making the integration of WireVis into their daily practice as seamless as possible is very important, and is one of our top priority items for the next phase of our project.

Since seamlessly supporting the work of the analysts is such a key aspect of this effort, WireVis is designed to be naturally extensible. Any statistical indicators and metrics can be plugged in depending on the need of the analysts or the latest intelligence information. For example, our clustering technique using binning is not intended to be the only solution to grouping accounts. It is only an example of how a fast clustering algorithm can enhance the analysts' ability to interact with the data. Depending on the need of the analyst, other clustering techniques can be used.

Although we provided four views that show relationships between accounts, keywords, and time, they are not the only four possible views that could be used. During our discussions with analysts, we have identified other views that can be helpful. Some examples of the views include a matrix or graph view showing the relationship between accounts and a geographical view showing the location of the senders and receivers. We will explore these possibilities in the future.

WireVis can be made significantly more powerful by adding an unstructured text analysis capability, such as that used in IN-SPIRE [187]. This would permit relating any words in the transaction fields to the keywords. Important relations could be uncovered and even new keywords found. We expect to develop WireVis along this line.

Finally, as indicated in the beginning, the WireVis approach described here can be applied to any transactional data or, indeed, to any keyword-based data over time. We will be expanding WireVis by coupling to analysis tools that search for money service businesses (i.e., businesses that deal in money transmission, check cashing, money orders, currency exchange, etc.).

3.2 GTD: Investigative Visualization

The war on terror has taken center stage since the 9/11 attacks. With the establishment of the Department of Homeland Security and corresponding international efforts, more analysts are participating in the investigation of world-wide terrorist activity. These analysts seek to reveal the patterns of activities of different terrorist groups and the relationships among them. To make their investigations complete, analysts need to uncover the facts of terrorist events and incorporate them into a broader context. However, until recently, most reports on terrorism have been scattered across different data sources, making it difficult to build a cohesive picture.

The Global Terrorism Database (GTD) project has consolidated both domestic and international terrorist activities between 1970 and 1997. With the wealth of data represented by the GTD, the challenge now becomes to understand and uncover important patterns and relationships. Unfortunately, analysts until now have had limited tools to help build hypotheses and identify trends, limiting the speed of understanding this important information.

We present an interactive visual analytics system to explore the GTD. Our system is designed around depicting one of the most fundamental concepts in investigative analysis, the five W's (*who*, *what*, *where*, *when*, and *why*). Four coordinated views are built to depict each of the first four W's (Figure 15), while the exploratory relation-probing nature of the system allows the analyst to build her own *why* out of the information at hand. We have informally evaluated this approach by presenting our system to three groups of expert investigative analysts with backgrounds in criminology and law enforcement. The results of this evaluation show that the system can assist the analyst in building an integrated understanding of terrorist activities.

The system presented in this paper represents significant improvements over existing practice in the following ways:

- It provides visual analysis of the five W's, which supports existing investigative thinking.

- It supports both strategic analysis of high-level patterns and tactical analysis of individual

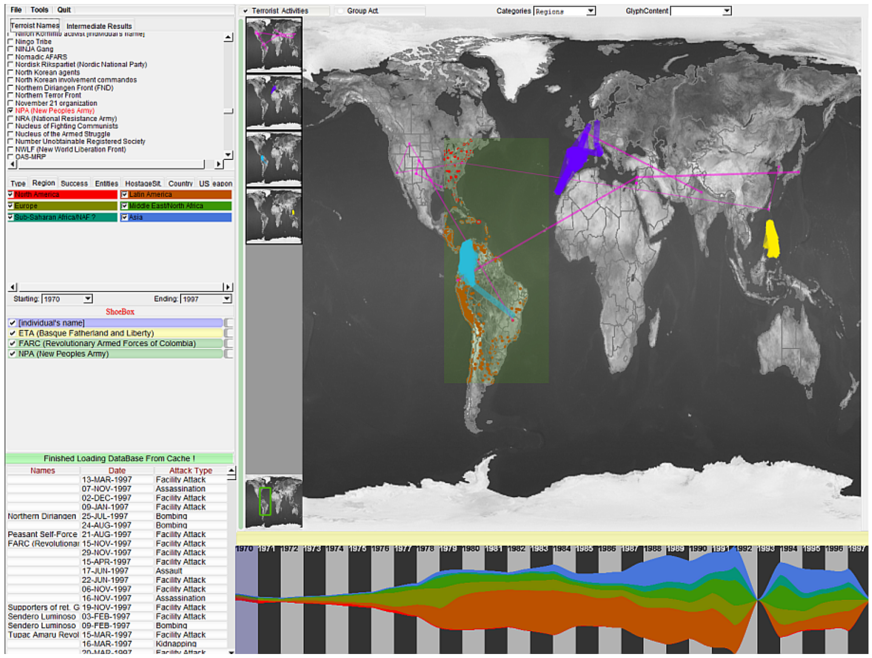


Figure 15: This is the overview of the entire system, including the map view (top right), temporal view (bottom right), entity view (top left) and activities panel (middle left).

events.

- It facilitates communication of investigative findings and hypotheses among analysts.

3.2.1 Related Work

There is an extensive literature devoted to the study and analysis of terrorism. These studies mostly focus on presenting the results of their analyses using qualitative descriptions or simple graphs (such as histograms or line graphs over time) to show trends of one or two variables [74]. In the case of the Global Terrorism Database, which contains more than one hundred dimensions, these descriptions and graphs simply do not sufficiently communicate the complex relationships among all the variables and, in particular, their detailed spatial relations over time. In this situation, it is difficult for the analyst to identify global patterns and trends or to formulate hypotheses and perform high level strategic reasoning.

The use of exploratory visual analytics concepts to understand complex relationships in terrorism activities is still in its infancy. The research up to this point can be divided into two groups: social network and geo-temporal visualizations. Social network visualization and analysis is a well established area in both the sociology [52] and visualization communities [152], but few systems have been applied to understanding large amounts of terrorist activities. Shen et al. [152] developed OntoVis, which utilized an ontology graph to visualize large heterogeneous networks and applied it to depict relationships between terrorist groups. Perer and Schneiderman [127] analyzed the GTD in their SocialAction system and could display relationships between terrorist groups and/or countries. On the other hand, although geo-temporal visualization is also a well established area of research, particularly in the field of geographical information systems (GIS), there has been little application of this approach to terrorism data. In fact, the only system that we're aware of that has been used directly in understanding terrorist activities by depicting temporal, geospatial, and multivariate aspects of terrorism was developed by Guo et al. [74], which does not focus on the relationships between individual entities. It is, however, relevant to mention that although most GIS systems do not directly address the issue of terrorism, systems such as Improvise by Weaver [182] and GeoVista by Takatsuka et al. [164] are example GIS tools that have shown great analytical capabilities in disparate types of data that could potentially include terrorism activities.

Surprisingly, we have found very few comprehensive systems that attempt to incorporate both the social and the geo-temporal aspects of terrorist activities. The most notable analytical tool that encompasses all areas is the Oculus suite of GeoTime [96] and Sandbox [188]. The combined system by Oculus has shown the ability to track and analyze the evolution of events and maintain the five W's throughout an investigation process. However, to the best of our knowledge, the system has not been applied to identifying similar trends and patterns between terrorist entities such as the GTD. A framework proposed by Zhu in a poster [195] demonstrated a structure for automatically identifying who, when, what, and where in a salient story regarding terrorist events. While their framework concentrates on extraction of information from unstructured text, our system uses existing terrorism data and focuses on the interactive exploration and analysis of the relationships within the data.

3.2.2 Global Terrorism Database

The U.S. Department of Homeland Security (DHS) has as a primary mission preventing terrorist attacks within the United States and reducing the vulnerability of the United States to terrorism. With support from the DHS, a team of researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START) has developed a new database designed to help analysts, practitioners, and policy-makers achieve DHS's mission while also providing a resource for the other investigative analysts, such as reporters.

The Global Terrorism Database, or GTD, provides detailed information on terrorist events that have occurred all around the world since 1970, including all domestic cases (e.g., an American attacking a target within the United States, or a Frenchmen attacking a target in France) as well as international events (where the perpetrator attacks a target in a foreign country, as with the 9/11 attacks). The current database (GTD 1.0) has over 60,000 incidents and 119 dimensions. Over 2000 terrorist groups (*who*) have been recorded in this database, connected with events on a 27-year time line from 1970 to 1997 (*when*). We use a set of well-defined categorical dimensions as the *what* in our system. For instance, we list attack type, target type and weapon type in the *what* view, so that the tactical incident can be analyzed from different aspects. Therefore, by mapping each individual incident to its location (*where*) and showing the rest of the W's, we offer a complete, highly interactive system to assist analysts in researching this critically important dataset (or any other similarly structured data).

3.2.2.1 Definition of Terrorism

It is relevant to mention that although individuals may have different interpretations of what constitutes a terrorism activity or who should be labeled as a terrorist, the definition of "terrorism" in the GTD is very clearly defined as "the threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious or social goal through fear, coercion or intimidation. [104] [120]" Based on such definition, any non-state actors engaged in terrorism are labeled as terrorists in the GTD and throughout this paper. However, since the purpose of this paper is to present a visual analytical system, any specific reference to entities have been removed to preserve anonymity.

3.2.2.2 Investigative Analysts

Since the first public release of the GTD in May 2007, there have been more than 110,000 hits to the web-based GTD interface. Generally speaking there are three types of users: the general public, investigative analysts, and terrorism experts and researchers, including counter-terrorism practitioners. These three groups of users correspond to three levels of use of the GTD in terms of sophistication and depth. The general public represents users who are largely unaware of historic global terrorist activities; investigative analysts are those with some knowledge of terrorist activities; and terrorism experts and researchers have in-depth knowledge of the groups and events, especially in the areas of their expertise. Surprisingly, most queries to START about the GTD comes from investigative analysts who are interested in identifying terrorist events and finding relationships or correlations between terrorist activities that are not previously known. The typical task for an investigative analyst could be to identify patterns of terrorist activities in a specific country over some time, to discover common targets of terrorist groups, or to find out if attacks occur near each other and around the same time, indicating the possibility of collaboration or coordination among the responsible groups. It is with the needs of investigative analysts in mind that we design our system. The aim is to create the system such that it is intuitive for analysts and yet powerful enough to support the investigative process by answering the five W's. In this way and because its analysis can be both general and highly detailed, it can also support the needs of expert users and researchers.

3.2.3 System Overview

In order to allow investigative analysts to freely explore terrorist activities and discover new trends and relationships, our system uses a different pane to depict each of the W's in a highly coordinated manner such that interaction with one of the panes will immediately affect the views and results of the others. This multi-view approach is flexible in nature and allows an analyst to inject knowledge

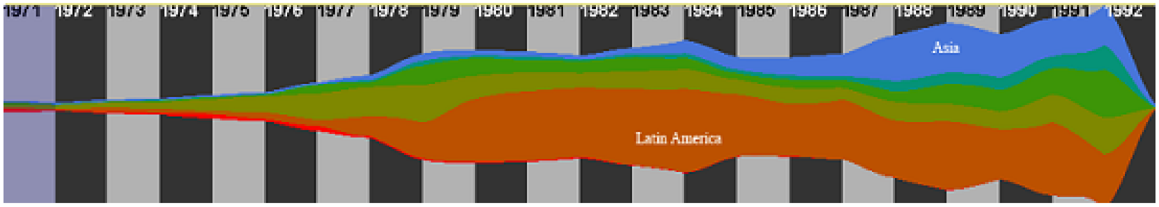


Figure 16: Regional ThemeRiver: this view contains both dimensional information and temporal information, indicating global terrorism trends from 1971 to 1992. Here it shows regional distributions, as labeled.

of any of the four W's in any order. As more information is given to the system, the more precise and detailed the results become, allowing the analyst to reduce irrelevant information and focus on the desired incidents or relationships.

To support the interaction of *where*, we create a zoomable map view that locates each terrorist incident globally (Figure 15 top right). An interactive ThemeRiver [77] shows the trend of a user-chosen variable over time, depicting *when* (Figure 15 bottom right). A list of checkboxes organized by dimensions allows the user to select the attributes of the incidents (*what*), and an updating panel details the terrorist groups that are of interest to the analyst (*who*). We further create a shoebox pane that serves as a container for evidence collection. Terrorist groups in the shoebox can be selected and compared to identify similar trends and patterns (Figure 15 middle left). Lastly, a simple tabular pane displays the original incident record in plain text, which allows the analyst to drill down to the details of each incident at any time during the investigation.

3.2.3.1 Where: Map View

Maps are arguably one of the easiest and most intuitive visualizations to understand [133], so the *where* view is the centerpiece of our system to visualize entities and their relationships. Since the majority of the incidents in the GTD contain geographical information of varying degrees of accuracy, our map is zoomable and pan-able to enable investigation at any granularity. In order to maintain the user's sense of location when zoomed into a small region, a mini-map is added to show the user's current view in the global scale. With each incident displayed on the global map, detailed geographic relationships and patterns become immediately apparent. Figure 17 shows the attacks in Egypt are all along the Nile River, while most attacks in northern Africa take took place along the coast of the Mediterranean Sea. The incidents can be colored based on categorical dimensions of the data such as attack method or weapon type (Figure 17) that can easily reveal patterns based on locations and trends over time. Display of numerical data such as number of casualties is shown using transparent circles (Figure 18), with the size of the circle corresponding to the value for each incident. In this way multiple relations with precise geographic patterns and distributions over time can be built up and understood.

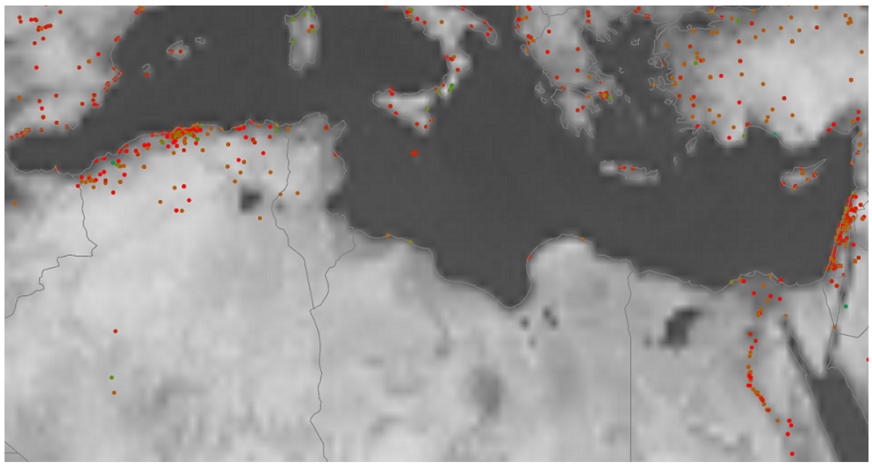


Figure 17: Global distribution of incidents are shown colored by attack type. The locations of the incidents show that attacks in North Africa occur largely along the coast and the Nile River.

One important feature of the map view is depicting the temporal sequence of activities of the terrorist groups by connecting the incidents with edges (Figure 18). Although the obvious use of this visualization is understanding the sequence in which incidents occur, the greatest benefit of connecting the incidents is to give a *shape* to each terrorist group. By relying on humans' ability to detect and compare different shapes, this visualization makes comparing between different groups intuitive. To better identify individual shape patterns, we provide a gallery view to collect shapes for each selected group (Figure 18 left). As can be seen, the groups that operate within a confined geographical region versus the ones that have global operations become immediately distinguishable. Furthermore, as shown in the evaluation section (section 3.2.5.2), using this kind of shape makes detecting certain errors and outliers straightforward and obvious (Figure 18 right).

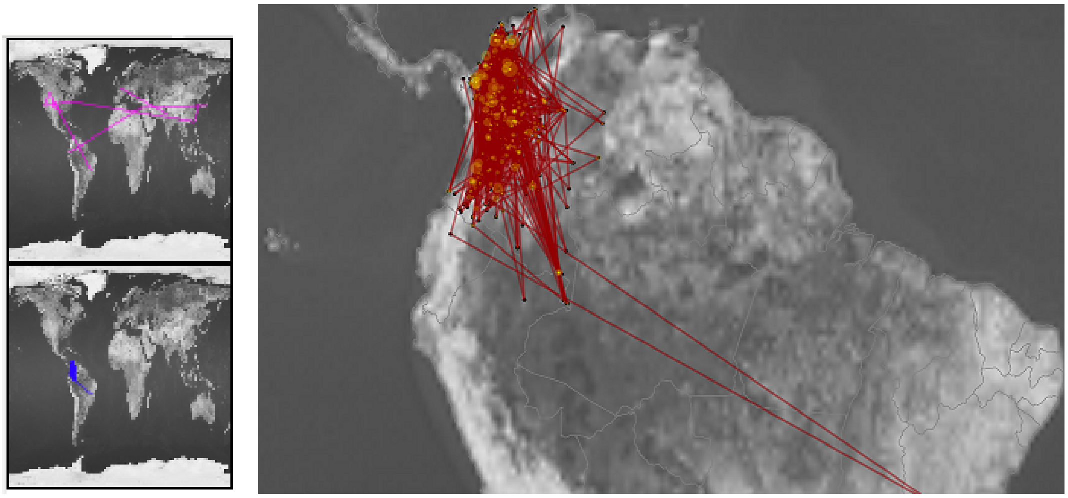


Figure 18: Shape Gallery (left) shows activity patterns for individual group. Activities of a terrorist group (right)are connected by lines to indicate the temporal sequence of events. The yellow circles represent the number of casualties per incident. The overall shape indicates that this group is mainly domestic but has one significant geographical outlier.

3.2.3.2 When: Temporal View

Understanding temporal patterns and trends in terrorist activities is the *when* aspect of an investigation. Our method for interacting with time, based on ThemeRiver [77], shows time in relation to categorical dimensions in the terrorist events. The streams in our ThemeRiver correspond to values in a categorical dimension that is interactively selected by the analyst. The ThemeRiver reveals global temporal trends and patterns, as well as the relative growth and decline among the streams over time, making it much more knowledge-rich than simple time slider. As an example, Figure 16 shows the overall increase of terrorism around the world as well as the growth of terrorism in each of the continents. Specifically, the rise of terrorism in Latin America (brown) in the early 80s is evident, as is the rapid growth of activities in Asia in the late 80s (light blue).

3.2.3.3 What: Activities Panel

To help organize over a hundred dimensions in the GTD dataset, we use a panel with tabs that correspond directly to the categorical dimensions in the data. Each tab represents a dimension, and the possible values for the dimensions are shown in different colors and unique labels in the form of checkboxes (Figure 19). The combination of these checkboxes denotes the *what* aspect of the terrorist incidents. Interacting with these categories by checking/un-checking corresponding checkboxes in the activity panel allows the analyst to filter and identify terrorist events that fulfill any investigative criteria. It is important to note that the colors shown in the activities panel are the same as the colors in the ThemeRiver as well as the map view and can therefore be used as a color legend for the entire view.

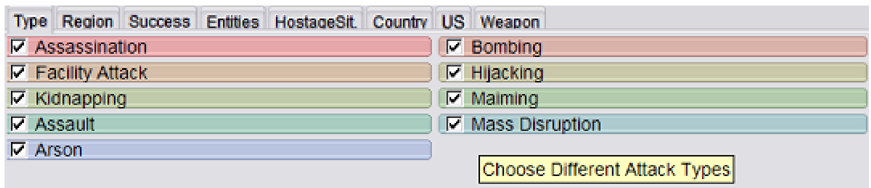


Figure 19: Activities Panel: This is an interactive filtering panel, which uses dimensions in the GTD to filter and color events in the other views.

3.2.3.4 Who: Entity View

We use a combination of three panels to enable the analyst to find any targeted terrorist groups among more than two thousand in the GTD. An alphabetical list of the groups allows the analyst to start the investigation by searching on specific names (Figure 20(left)). A correlated view connects to the map view, the ThemeRiver, and the activities panel in a way that filters the terrorist groups by *where*, *when*, and *what*. Lastly, we offer a shoebox as a container for collecting terrorist groups that are of interest to the analyst (Figure 20(right)). An individual terrorist group is inserted or removed from the shoebox manually, allowing the analyst to form and test hypotheses.

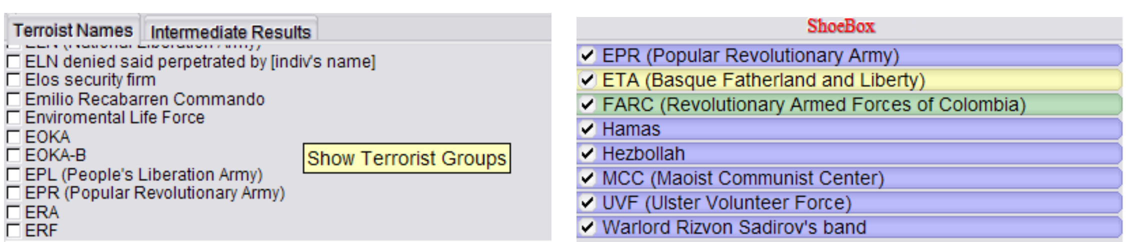


Figure 20: Entity view: The Terrorist Name tab (left) lists 2404 terrorist groups extracted from the GTD, while the Intermediate Results tab shows suggestions for possible terrorist groups. The Shoebox (right) is a container for collections of user-selected terrorist groups, providing the user an easy way to compare different terrorist groups.

3.2.4 Scenarios

To show the utility of our system, we identified a few scenarios in collaboration with the START center that demonstrate possible ways in which the system can be employed. Our focus in these scenarios is not only on the goals of the investigation, but also on the paths which an analyst might take to arrive at the conclusions. In both scenarios described below, we demonstrate that an analyst can begin to deduce the *why* of the attacks by identifying the other four other W's.

3.2.4.1 Linking Tactical Incidents and Global Strategy

In this scenario, an analyst begins by examining one of the most talked-about groups in recent years called Organization A[1] by selecting it from the entity view (*who*). It is widely known that this group is an Islamic political and paramilitary organization based in the Middle East that has recently been in conflict with another country. Based on their reputation, one might reason that the activities carried out by this group would concentrate in the Middle East. However, upon highlighting the group in the entity view, it becomes immediately evident that this group's activities between 1979 and 1997 are not limited to a specific region (Figure 21). A closer inspection in the map view (*where*) shows that out of the 200 attacks conducted by this group, the majority of the incidents take place in a specific country in the Middle East, but it is clear that they do indeed have a global strategy of operation in both Europe and South America.

One obvious outlier in the pattern of incidents is the major attack in Argentina in 1994 for which Organization A claimed responsibility. Double-clicking on that incident in the map view shows that the attack was on a Jewish community center in Buenos Aires that resulted in 96 deaths and more than 250 injuries. Although inconclusive, this incident suggests that Organization A's attacks are not based on their geographical locations, but influenced instead by the religious beliefs of its targets (*why*).

Examining the timeline (*when*) also reveals a change in Organization A's attack pattern over time (Figure 22). It appears that between 1984 and 1988, Organization A's primary attack method (*what*) was kidnapping. Starting in 1989, its strategy changed to a mix of bombing and facility

[1]The actual name of the group has been removed for the purpose of anonymity.

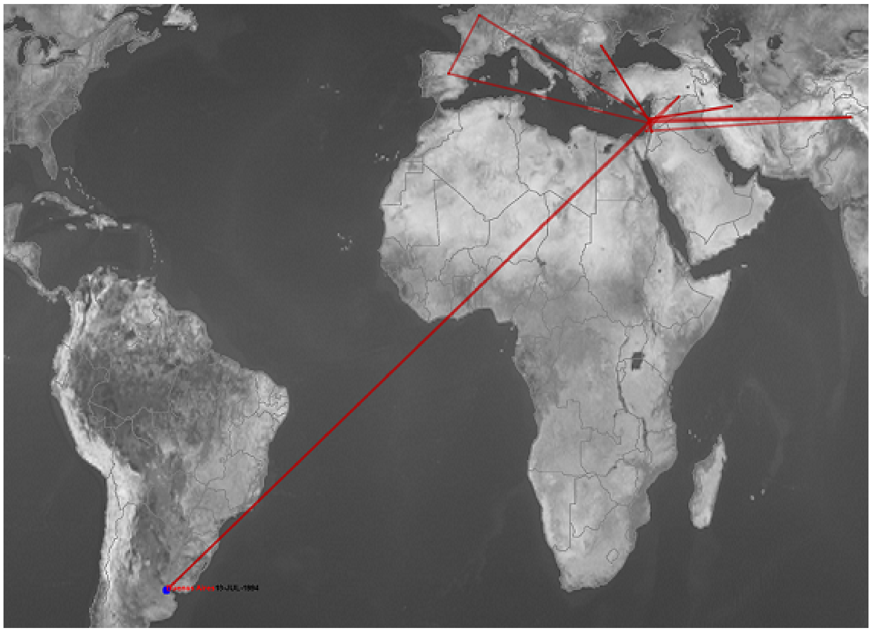


Figure 21: In this example, edges indicate that Organization A is an active international terrorist group, with a wide attack range. The highlighted city illustrates a significant outlier in Argentina in 1994.

attack. Finally, facility attack became the predominant attack method in the early to mid 90s, while the number of bombings decreased and eventually stopped.

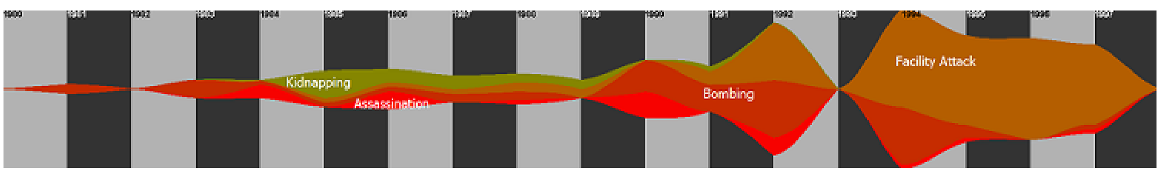


Figure 22: This image depicts changes in Organization A's attack methods over the last 17 years, which shifted from kidnappings to a mix of bombings and facility attacks, to almost entirely facility attacks. Note that the GTD does not contain data for the year 1993 [104].

The investigation in this scenario included all four of the W's and ended in a limited but plausible hypothesis of *why* Organization A conducted terrorism around the world. It uncovers both tactical decisions and methods of Organization A's attacks, but also makes clear the trend and pattern of their global operations.

3.2.4.2 Discovering Unexpected Patterns

By examining the overview shown in Figure 15, we can see that a great deal of terrorist attacks took place in the Philippines. Zooming into that specific region and selecting the entire country in the map view (*what*) lists all the terrorist groups active between 1970 and 1997. A quick search in the entity view (*who*) shows that Organization B† is one of the most active groups in the region (Figure 23). Highlighting this group in the visualization reveals that although active, Organization B is strictly domestic and has never performed activities outside of the Philippines.

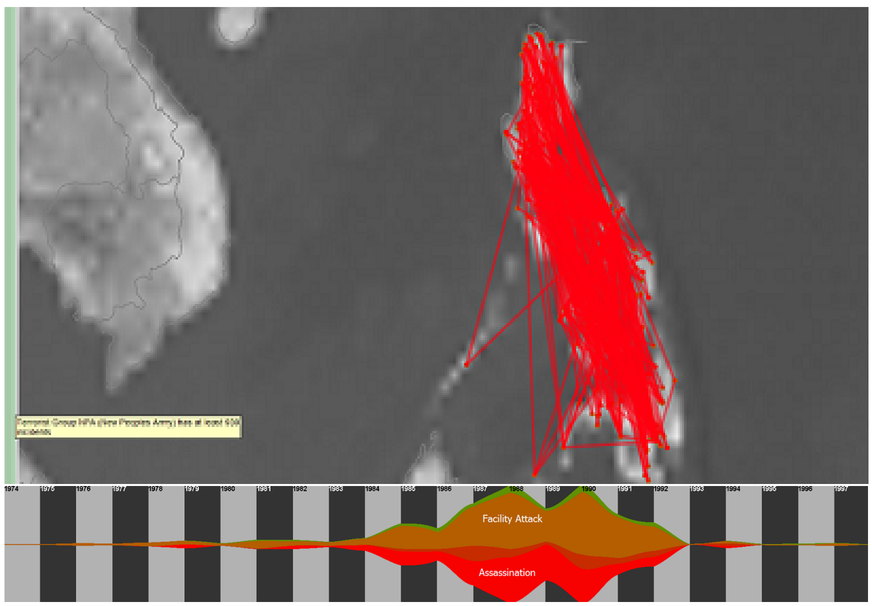


Figure 23: (Top) This image indicates that the Organization B was an active domestic terrorist group that was responsible for more than a thousand incidents within the Philippines. (Bottom) This result shows changes in Organization B's attack frequency during the time period of the dataset. Though it was very active in the late '80s, it suddenly disappeared in 1992.

An examination of the ThemeRiver (Figure 23), however, shows an interesting temporal pattern in Organization B's activities. It appears that while Organization B was indeed active during the '80s, it stopped operating entirely in 1992. This unexpected temporal behavior, unfortunately, cannot be explained using only the data within the GTD. In order to find out what happened in 1992 in the Philippines (*why*), we turn to newspaper archives and found out that in 1992 Fidel Ramos became the twelfth president of the Philippines, and immediately offered peace treaties to various terrorist groups including Organization B [153]. Although it is unclear whether this event is directly linked to the diminished activities of Organization B, it forms the basis for a plausible hypothesis for further investigation. This shows how evidence gained from the GTD visual analysis can provide a specific basis for continued investigations using open source or other data.

3.2.5 User Evaluation

We asked three groups of experts in law enforcement and criminology to evaluate our system (a total of 8 participants). Our experts are from the criminology department at the University of Maryland, the National Insurance Crime Bureau (NICB), and the Drug Enforcement Administration (DEA). The evaluation was conducted informally. First we presented our system by demonstrating the investigative scenarios described in the previous section. Then the experts were given some time to interactively explore the system and the GTD. Finally, we concluded the evaluation by asking them to give feedback and comments. All these experts gave consent to have their comments and affiliations published in this paper.

3.2.5.1 The Five W's (Who, Where, When, What, Why)

At the heart of our system is the integration of the five W's in a comprehensive and cohesive manner. This design is validated by one of our expert evaluators, who serves as a special agent in the NICB. He said, "The five W's you are using here are exactly similar to what we have in police reports. It is critically important for us to understand individual incidents from different aspects, like the *who*, *where*, *when*, *what*, *why*." Also, according to GTD data designers at University of Maryland, "In particular, your system presents users with an efficient means to access multiple dimensions of terrorist event data simultaneously. True to the goals of the project, it allows users to see where specific terrorist groups were operating during defined time periods and to discern the nature of the weapons and attacks that the groups engaged in. As such, it provides a streamlined mechanism for helping users to identify behavioral trends among terrorist groups over time. Traditionally, users working with the numeric version of the GTD would have to sort through tens of thousands of data rows to come up with information on cases that meet specific criteria. This tool greatly facilitates this effort."

3.2.5.2 Tactical vs. Strategic Analysis

Tactical analysis is traditionally considered to be the pursuit of short term analytical goals using resources at hand, while strategy is the longer-term pursuit to understand a group or a situation at a higher level. Our experts from law enforcement confess that the large numbers and density of reports they receive on a daily basis force them to think tactically, rather than strategically. They agree that an exploratory visual analytics system could help to reduce the amount of noise they have to sift through in order to see the broader picture and hone in on the suspicious outliers. Two experts explicitly pointed out that it has become more and more important for tools to help analysts take a step back from tactical analysis to strategic analysis. They agreed that our tool will assist this analytical pattern by visually providing a global pattern as well as details on demand. According to the law enforcement analyst from the DEA, both outliers and global patterns are critical to analysis. “It is very useful to directly mark those targets with their strategic shapes. Your system could benefit the Federal government’s interest in investigating both local terrorist groups and their attempts to develop ties with other international groups.”

3.2.5.3 Reducing Communication Gaps

Conveying knowledge and investigative results visually can drastically reduce the amount of effort spent on communication and reduce ambiguity. Since our system is designed to display the five W’s, it can present a complete and coherent picture of the current state of an analytical process. According to the analyst from NICB, the current practice of using text-based reports for sharing insights and hypotheses found during an investigation process tends to be error-prone and time-consuming. “[Your system] will greatly shorten the catch-up time between police shifts and guide them to focus more on things that they are interested in.”

Along the same lines, the expert in criminology praised our system for its intuitiveness and ease of use. Since the GTD is a large and complex dataset containing tens of thousands of records with over a hundred dimensions, it has been difficult for non-experts in terrorism investigation to utilize this wealth of data. With a fully interactive and exploratory interface, our system effectively shortens the distance between terrorism experts who prepare and collect the data and the users of varying levels of expertise and backgrounds who seek to identify patterns and trends in terrorist activities around the world.

3.2.6 Discussion

Encouragingly, the experts from law enforcement who participated in our evaluation were eager to use our system for their own purposes. During the evaluation, both of these analysts were already picturing their own data visualized in our system. They both consider the structure and nature of the GTD to be very similar to the financial and criminal reports that they investigate, and could see the tool having an immediate impact on multiple aspects of their day-to-day jobs, including tactical analysis of incidents, strategic thinking about global trends and patterns, and communication and reporting of their investigative findings with peers and superiors.

More importantly, they foresee an unexpected use of the system in predicting future trends and activities. Our system's ability to depict temporal trends clearly in relation to both geographical and other patterns suggests possible future directions of events. Although the predictive capability of the system is not one that we had considered, we are very excited about its promise and potential benefits in the field of law enforcement. Based on these positive responses, we plan to place our system in the hands of these and other experts for continued use and evaluation.

Our collaboration with social scientists has certainly been a fruitful and interesting one. Since most social scientists that we have worked with are not familiar with visualization, they have appreciated the new perspective that our visual tools bring to their data. However, this lack of familiarity with visualization has also led some of our collaborators to express wariness about the faithfulness of the visual representation to their data. It is our experience that our geospatial view has greatly helped to reduce this wariness, presumably due to the widespread use of maps in general. In future collaborations with social scientists, it may prove useful to build novel views on a foundation of more familiar ones.

The only major criticism of our system concerns the visual representation of overlapping incidents and overplotted lines in the map view. For instance, when multiple events occur in the same city or region, it is not easy to see the number of events that overlap each other (e.g., there are more than 700 incidents in Beirut alone). This may lead the user to underestimate the degree to which a specific location has been attacked. Similarly, with overplotted lines in a condensed region, the shape of the activities and their temporal relationships can easily be lost. We are investigating new visual representations that may alleviate these issues.

3.3 Interactive Analysis of Biomechanical Motion Data

Effective visualization of 3D motion is a complex problem, especially as it relates to experimentally collected data in biomechanics. Imaging modalities, such as biplane fluoroscopy combined with CT, are now able to capture high-speed motion of the bones of a joint at rates of 250 to 500 frames per second with sub-millimeter accuracy [173, 193]. These data allow for far more detailed study of a variety of complex motions in animals and humans than was previously possible. Several important visualization challenges arise from working with motion data sets collected with these technologies.

The first challenge in visualization and analysis of these data is understanding the complex spatial relationships that are present. This is a 3D problem, in the sense that the bones exist in a 3D space and, in many cases, the relationship between the 3D shape of the bones and their function (functional morphology) is one of the primary scientific research questions. Thus, effective 3D spatial understanding is an important feature of visualization systems appropriate for use with these data.

The second challenge in analysis of these motion data is that, in both clinical and experimental work, these data typically exist as part of a large database. For example, when a scientist designs an experiment to study chewing motions (the primary example used in this paper) she will typically collect data on tens to hundreds of chewing cycles. Questions posed during analysis may be of the form, what is a typical chewing motion as exhibited across the data? Or, they may be of the form, how does chewing change based upon the amount of food in the mouth, the type of food, or other variables? This style of analysis requires understanding the time-varying spatial data that describes a single chewing motion, and, beyond that, it requires understanding trends and anomalies in this time-varying spatial data across databases of numerous repeated motions.

Previous approaches to visualization of biomechanical motion data have benefited from animated and interactively controlled 3D graphics [97, 114, 89]. Our collaborators in evolutionary biology have also had positive experiences with 3D visualization of their data. In fact, several of them have found 3D views to be so useful that they have taught themselves how to use a combination of Maya and Matlab to produce their own 3D visualizations. In general, previous 3D visualizations presented in the literature have provided useful capabilities for investigating individual motions, but provided only limited capabilities for analysis and comparison of a set of motions. We present a framework that explicitly supports visualization of multiple related motion sequences, an important scientific task in this context.

Visualization of trends in time-varying and multi-variate data has a rich history of study within the information visualization community [140]. Our work is motivated by a desire to leverage the theories and techniques resulting from this work and bring these to bear within a system that targets time-varying 3D data. Since spatial relationships in these data are so important, they have tended to be visualized in the past with what have traditionally been termed scientific visualization techniques, or 3D spatial layouts where the spatial arrangement is pre-determined by the true 3D arrangement of bones in space. To observe motion over time, these 3D views have been animated, and often additional data attributes are visualized via color, texture, streamlines, and 3D data glyphs [97, 114, 89]. While these sophisticated 3D views are clearly valuable, evidence in the information visualization literature suggests that, in general, understanding trends over time through animation may not be the most effective strategy [140]. Based on this insight and other promising results in information visualization [33, 125, 139], we have been motivated to explore a new visualization framework that combines the strengths of both information and scientific visualization approaches and targets understanding of spatially complex, time-varying motion data.

The idea of bringing information and scientific visualization together is not new [76, 137], and several compelling examples tied to real scientific analysis now exist in the literature [126, 132, 169]. However, many challenges remain in this line of research, especially as it relates to specific forms of data. Analysis of detailed 3D motion, for example, raises the question of the most appropriate roles for animation, comparison views, and techniques such as parallel coordinates, which have been widely applied in general, but less so within the context of biomechanical analysis.

Specific to analysis of 3D biomechanical motion data, integrating 3D and 2D visualization techniques, as in the overview visualization of Figure 24, is particularly appealing because each brings a unique strength that compensates for the other's weakness. Analyzing motion trends using only 3D visualization tools imposes a high cognitive load on the user, since analysis often requires comparisons between multiple detailed motions that the user must keep in his working memory [105, 140]. On the other hand, when analyzing motion using only 2D visualizations, the abstract representations of the motion data do not provide the necessary context for the user to understand the 3D structure of the object or its movement in space. We propose that when the two are integrated together in a tightly-coupled manner, the user gains the benefit of both perspectives and can perform analysis of the motion in both space and time.

The high-level contribution of our work is presenting an integrated framework for 2D and 3D interactive visualization of experimentally collected biomechanical motion data sets. To this end, specific contributions of the work are:

1. The design of an overview technique for visualization of hundreds of repeated cyclic motions.
2. Methods for zooming, filtering, and exploring motion data via linked 2D and 3D views, including the ability to pass data generated through interaction with 3D views on to linked 2D visualizations, and vice versa.
3. The design of side-by-side and overlay-style coordinated views for comparison of the motion of bones in space and the resulting interaction between multiple bone surfaces.
4. A discussion of lessons learned, current limitations, and future directions as motivated by a driving real-world application.

In the following section, we provide some background on the data and scientific application (analysis of pig chewing behavior) discussed in this paper. This is followed by a discussion of related work in visualization. Then we present the motion visualization framework in detail, followed by more specifics of the driving application and initial feedback from domain scientists. Finally, we present a discussion of lessons learned in developing this system, including limitations and future directions.

3.3.1 Application Area and Data

The framework presented here is likely to apply to analysis of a number of experimentally-captured motions of interest to orthopedists, physical therapists, and evolutionary biologists. The example application driving the work in this paper comes from the field of evolutionary biology, where our collaborators are studying mastication in minipigs (Sinclair strain).

In general, the mechanics of the mandible, skull, and teeth working together in chewing motions are an interesting area of study, both in humans [57], and in animals [97]. Pigs, in particular, follow an unusual chewing pattern, called bilateral chewing, that is characterized by motion of the jaw up, then a food grinding motion to one side, then down, then up again, then sideways food grinding motion to the other side, then down again. This pattern repeats over several chewing cycles. (The alternating grinding from side-to-side pattern can be seen in the tracer paths in adjacent small multiples views in Figure 25.) This characteristic motion has been visualized previously [97], but only for investigating a single chewing motion at a time. Research questions require analysis of multiple instances of this motion (an important goal of this framework). For example: Can we catalog a "typical" chewing behavior? How does the movement of bones change over time based upon the amount of food in the mouth or the type of food?

Evolutionary biologists began their study by collecting data from multiple experimental trials in the lab. The trials captured motion from a number of different chewing-related behaviors, including food gathering, feeding on pig chow, and feeding on hard nuts (in the shell). Data were captured and processed using the X-Ray Reconstruction of Moving Morphology (XROMM) methodology citeBrown2009, in which highspeed biplane fluoroscopy is used to capture motion data during the experiment and a CT scan captured separately is used to reconstruct the 3D geometry of the bones and teeth. Computational tools are utilized to register the two sources together to reconstruct high-speed 3D motion data. These data are the source for the visualizations presented here. For this study, they include more than one hundred chewing motions (up-down motions of the jaw) collected

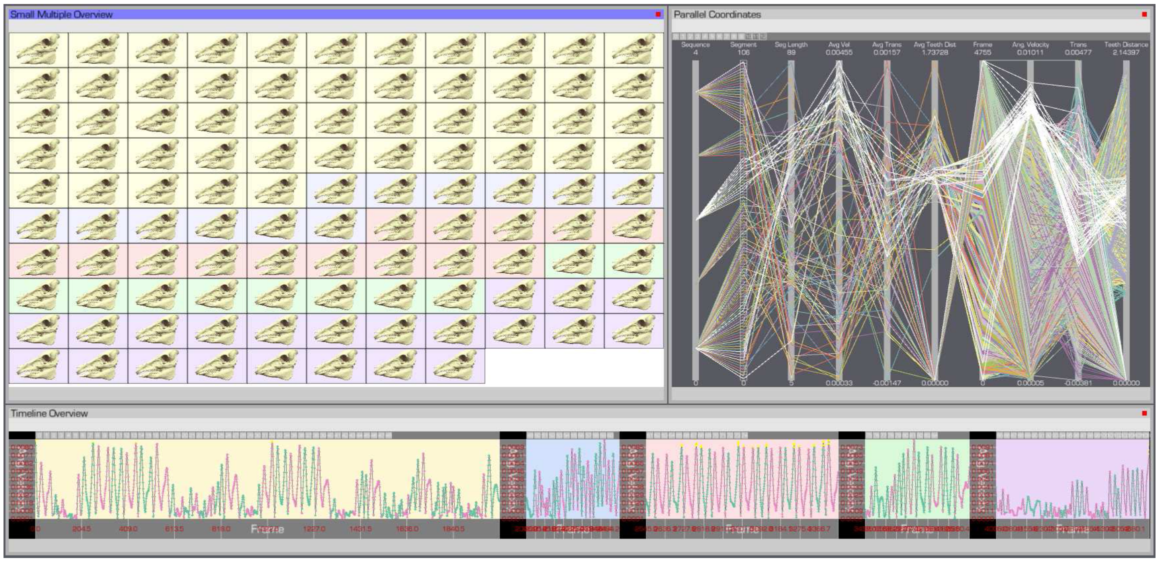


Figure 24: When data are first loaded into the visualization framework, an overview of the motion database is presented using three coordinated data views: 1. A small multiples view generated from snapshots of 3D renderings (top-left window in the figure). 2. A parallel coordinates view (topright), data dimensions plotted in this example are: trial number, chew cycle number, cycle duration, average angular velocity for the cycle, average translational velocity for the cycle, average distance of separation of the teeth for the cycle, then frame number and the same set of descriptive statistics but calculated at the single frame level rather than as averages over a cycle. 3. A 2D plot of data values over time (bottom), here angular velocity over time. All views are linked through both visual and interactive strategies. In this case, 108 chewing motions cycles from five different trials are displayed in this overview.

during five experimental trials.

3.3.2 Related Work

In this section, we relate our work to relevant research in evaluating animated visualization as a tool for trend analysis, visualizing 3D biomechanics, designing coordinated multi-view visualizations, and combining scientific and information visualization strategies.

3.3.2.1 Trend Analysis and Animated Visualization

In recent work, Robertson et al. examined the effectiveness of animation for visualizing trends in data [140]. While animation is often attractive for presentation purposes, the results of this work suggest that static small multiples views and static traces of trend lines over time may be more effective than animation in analysis of trends over time.

These findings have far reaching implications for visualizations of motion data, which are, nearly by default, viewed as animations. While viewing an animated visualization of the motion of an animal does seem natural and intuitive (after all the data are collected over time) the question is raised, are animated visualizations the right tool for analyzing these motion data?

Based upon our experience with collaborators and the results of previous animated 3D visualizations in biomechanics, we believe interactive/animated 3D visualizations do play some important role in analysis. However, the findings of Robertson et al. highlight the potential importance of alternative complimentary techniques and raise the issue of identifying the right mix of animated and static views for motion visualization. Our framework explores these issues and builds upon the static representations demonstrated by Robertson et al. A key component of our initial motion overview visualization is a small multiples visualization [172], which we have often found useful to construct as a set of tracer lines (See Figure 25), following in the style of the small multiples traces presented in [140].

3.3.2.2 3D Biomechanics Visualization

Several systems for 3D visualization of biomechanics data exist in the literature [27, 76, 114, 193]. A primary function of these tools is providing a view of anatomical features (bones, ligaments, etc.) positioned appropriately in 3D space. Almost all of these systems also support some form of motion playback, often with some interactive support for adjusting camera parameters and playback speed.

Beyond simply replaying experimentally captured data, 3D visualization systems also provide for visualization of derived data. For example, computing helical (or screw) axes to describe the motion of one bone relative to another is a technique that is gaining popularity within the biomechanics community [36, 57]. Viewing the position of this axis in space relative to anatomical landmarks in a 3D visualization can provide insight into the rotation and translational components of a complex motion [97, 89]. Other examples of 3D visualization of these data include applying color maps to bone surfaces to indicate the distance from one bone to another [10], and drawing 3D tracer curves to indicate the path some anatomical feature takes through space over time [27]. Our 3D visualizations employ each of these techniques. The focus of our investigation is not on the introduction of novel 3D visualization techniques, but rather on how a state-of-the-art 3D visualization of biomechanics may be leveraged within a system that interactively links it with complimentary 2D visualizations.

3.3.2.3 Multi-View and "Scientific-Information" Visualization

The benefits of using multiple coordinated 2D visualizations for data analysis have been well documented [33, 125, 139].

Our approach relates most closely to multi-view techniques that employ a combination of 2D and 3D views to investigate data that follow a pre-defined spatial distribution, thereby combining scientific and information visualization techniques. Several visualization and interaction techniques fitting this description have been documented previously, for example, linking 2D and 3D scatter-plots [132], brushing over multiple dimensions in 2D views to identify 3D features [126], and using parallel coordinates as an interface for exploratory volume visualization [169]. Our work follows closely in the spirit of these techniques, however, our overview visualization, coordinated views, and use of animation and interaction are designed specifically to target analysis of high-precision motion data sets. As such, we have a special emphasis on the role of animation within our framework, and we have utilized specific properties of the data, such as its cyclic nature, in designing several components of the framework, such as the small multiples overview.

3.3.3 Motion Visualization Framework

In this section, we describe a novel framework for visualization of scientific 3D motion data. Through a series of visual tools, the framework supports the typical visual information seeking mantra: "Overview first, zoom and filter, then details-on-demand" [155].

3.3.3.1 Small Multiples and Coordinated Views for a Motion Database Overview

When a data set is first loaded, an overview of the data is displayed using the three coordinated view windows seen in Figure 24: a small multiples view, a parallel coordinates view, and a 2D plot of descriptive statistics computed for each frame of the motion over time. These three views have been carefully chosen for their analytical capabilities in analyzing different aspects of a 3D motion sequence. The small multiples view displays a representative motion snapshot for each motion sequence. The 2D xy-plot is chosen for its intuitive nature in representing time, described by Ericson during his keynote address in the 2007 InfoVis conference [48]. Lastly, the parallel coordinates view is used to reveal relationships between data dimensions based motion statistics and derived quantities. Together, these views allow the user to explore the 3D motion sequence in space, time, and at a dimensional level.

3D Snapshot Small Multiples: The utility of small multiples displays for analysis of trends over time has been demonstrated in a 2D data context [140], but several open questions remain in developing a small multiples strategy for 3D motion data. Relevant full-scale 3D visualizations are typically interactive and detailed. How do these translate to small scales? If the individual multiples are to support the same style of interaction as in normal visualization, then how do the interaction strategies change to work within a much smaller window? Several questions that are specific to motion visualization also arise, including, how are motions assigned to a small multiple? One small multiple per frame of the motion data will result in far too many views to be useful. On the other hand, if a single multiple stands in for a sequence of frames in the data, then how does that single image best visually represent motion over time?

Our approach to assigning motion to a particular small multiples image is based upon a characteristic of our target data. The biological motions we examine (chewing, walking, flying, etc.) are almost always cyclic. It is quite common to segment motions such as these into cycles as a part of the analysis. In walking, the moment the foot first touches the floor can be used as an indicator of the beginning and end of a single stride. In the chewing examples presented here, the motions are divided into segments of chewing patterns (a single up and down motion of the jaw bone). One motion segment is assigned to each small multiple image.

The image displayed for each small multiple is a snapshot of a 3D visualization generated using our typical 3D rendering engine and then texture mapped onto a small rectangle to produce the array of multiples. The default view when data are first loaded into the system is shown in Figure 24. Here, each multiple is a snapshot of a 3D rendering of the bones posed during the initial frame of each motion segment. The user may change the frame that represents this view by mousingover a frame number along the x-axis of the 2D plot, or highlighting a specific marker in the same 2D plot, or he may change the 3D view by clicking on a particular small multiple, which enlarges the rendering to fill its parent window, hiding the other small multiples. At a larger size, the 3D view is now easier to manipulate. The display now switches into an interactive 3D rendering mode and activates typical mouse and keyboard-based interaction widgets for camera manipulation, showing and hiding particular bones within the view, and adding visualization glyphs such as axes of rotation and tracer lines to the view. After some manipulation of these viewing parameters, the user escapes from this interactive view and is returned to the small multiples display, which is then re-rendered so that all of the views match the camera and visualization settings set by the user in the interactive mode. The background color of each multiple is set to encode the trail from which the data come; the colors correspond to those in the 2D time plot (see bottom of Figure 24).

Figure 25 shows a small multiples display generated in this way. In the interactive mode, the user attached a tracer to the front tooth on the jaw, zoomed the camera in to focus on the tooth, turned off the rendering of the bones, and then returned to the small multiples display.

Integrated Time-Plots and Parallel Coordinates Views: Accompanying the small multiples display in the overview visualization are two 2D views: a parallel coordinates visualization and a 2D plot of motion data over time. The three views are linked together via interactive brushing and highlighting. For example, as the mouse moves over the time plot the corresponding small multiple view highlights. Conversely, moving the mouse over a small multiple image highlights the corresponding section of the time plot. Each line in the parallel coordinates view corresponds to a frame of motion data. Brushing over data in the parallel coordinates view highlights the corresponding frames and data values in the time plot. The value plotted on the Y axis of the time plot may be changed interactively to map to any data attribute that may be calculated for each frame of the data. In practice, values such as angular or translational velocity for a particular bone are useful. The angular velocity of the rotation of the mandible is plotted over time in the view shown in Figure 24.

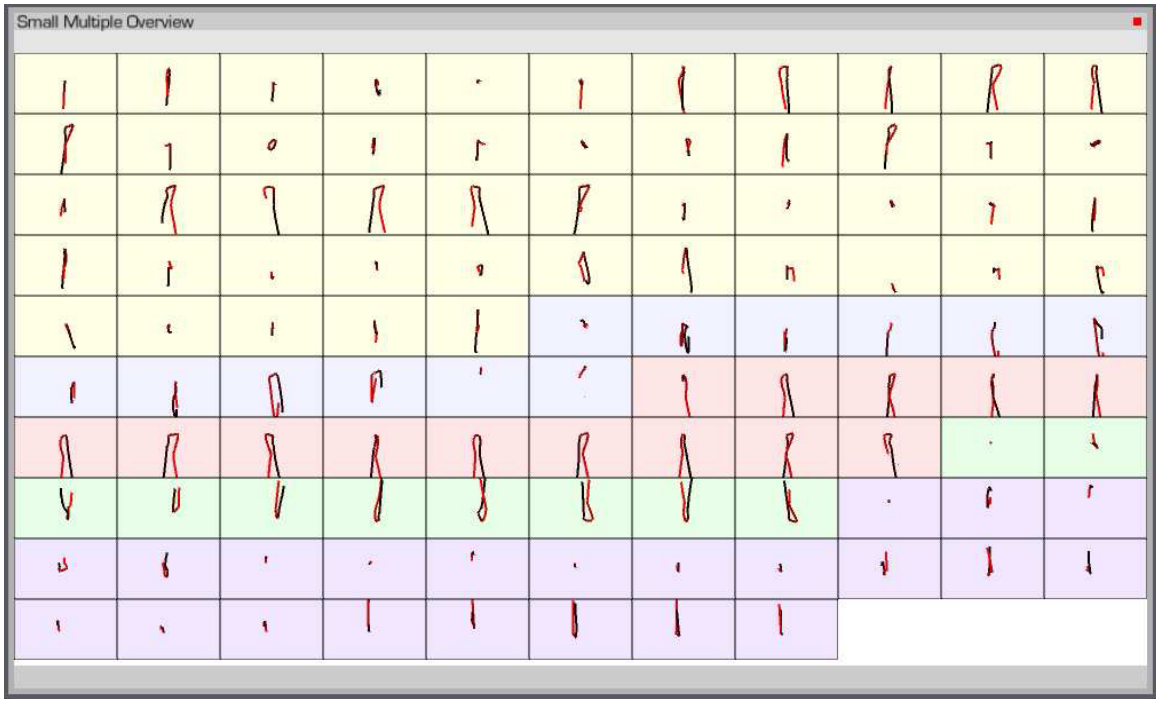


Figure 25: A small multiples display setup interactively by a user. To tune the display, the user zooms in to one of the small views, making an interactive 3D rendering of it fill the window. Then, he adjusts visualization and camera settings in the zoomed in view. When he returns to the small multiples view, each of the multiples is re-rendered using the new visualization settings. In this case, the user added a tracer curve to the visualization to trace out the path of the pigs front tooth. Then, he zoomed in on the location of the tooth and made the bones invisible. The characteristic bilateral chewing motion of the pig may be seen in many of the adjacent images. Look for a tracer that begins (black end of the curve) on one side, moves up, then comes down on the other side. In the image immediately to the left or right of this one, you are likely to find a tracer exhibits a similar pattern, but moving in the opposite direction. Cycles that capture food gathering behavior can also be identified in the display, characterized by tracers that are more compact than the elongated chewing motions. The background color for each multiple is set to encode the trial from which the data are drawn; the colors correspond to those used in the 2D time plot at the bottom of the screen (see Figure 24). Note: to better understand how the tracers were created, see the 3D view in Figure 26, which shows the bones together with a tracer placed in the same position as was used to generate these small multiples renderings.

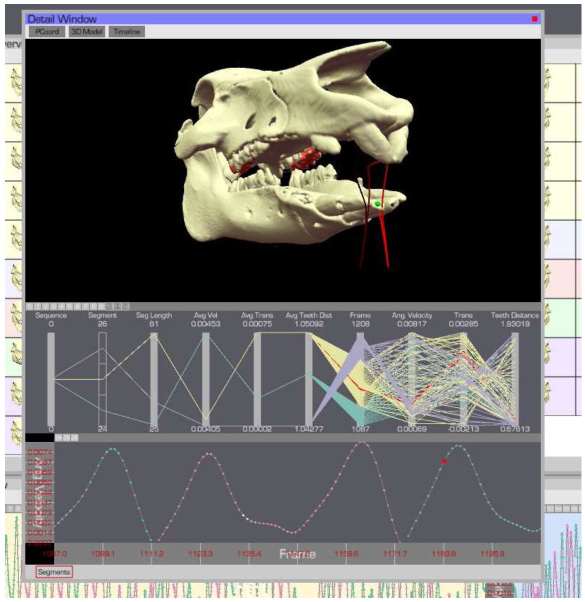


Figure 26: A coordinated multiple view window created by zooming in on a portion of the larger data set.

3.3.3.2 Filtering to Generate Zoomed-In Coordinated Views

Following Shneidermans mantra "overview first, zoom and filter, then details-on-demand," the system supports filtering down to specific segments or time ranges of the motion sequence. In the small multiples overview, a right mouse click on an image activates a menu, which is used to create a new coordinated view filtered to display just the segment of the motion that corresponds to the small multiple. Similarly, after brushing with the mouse to select a portion of the data in the time plot view, a right click and menu selection sends the selected data to a new multi-view zoomed-in window. Figure 26 shows this new zoomed-in window.

Interaction Between Views: The zoomed-in window contains three data views: the parallel coordinates plot, 2D time plot, and a new interactive 3D view. Both 2D views are similar to the overview versions with the exception that the data visualized is a subset of the original data. The 3D view is different from the overview. Rather than a small multiples representation, spatial trends are now depicted via a real-time 3D rendering of the data that is responsive to mouse and keyboard controls for adjusting viewing and visualization parameters. All views are linked visually and interactively. For example, the 3D view may be animated either through interaction in the 3D view or by mousing over the 2D time plot. In either case, the views advance together to display the active data frame as the animation plays. The 2D plot is not restricted to depicting time on the X axis. Other plots, for example, angular velocity vs. distance between bones, are also useful.

Generating and Visualizing Data through Exploration: A more interesting example of the tight linkage between these multiple views is the ability to seed new visualizations from data generated during analysis. Figure 27 illustrates an example of this. The user first filtered the data from the original overview to zoom in to a sequence of four main chewing patterns. Then, while interacting with the 3D view window, a tracer was created to mark the path of a point on the left condyle of the mandible. The path that this point travels through space is calculated and stored in a coordinate system relative to the pig's skull. The 3D points that make up the path then become available as a data source for the linked 2D views. In this example, the user brushed over high positive values for the vertical position (relative to the pig's skull) of the tracer using the parallel coordinates view. The white lines in this view show the highlighted data points. Since the views are linked, these values also highlight in yellow within the plot below of average distance between the teeth over time. The visualization shows that during jaw closing, the selected point on the condyle rotates backward and downward. At its low point, there is some side-to-side motion of the jaw as the teeth come together to grind food. The sideways motion is visible in the in the tracer shown in the 3D view.

Other 3D visualization systems have exported data generated during exploration to tools that may then be used to generate related 2D plots [97]. Important differences in the strategy described here are the tight coupling of the multiple views and the ability to build new views based upon data generated during exploration. A tracer placed in one view may generate data that are then used for exploration via interactive brushing and ultimately for a new filtering strategy. Then, based on this new filtering, a second zoomed-in coordinated view may be created.

3.3.3.3 Overlays and Side-by-Sides for Detailed Comparisons

Visual comparison of motion sequences occurs at all levels of the framework, and as the focus narrows, the method of visual comparison changes. As noted by Robertson et al. [140], small

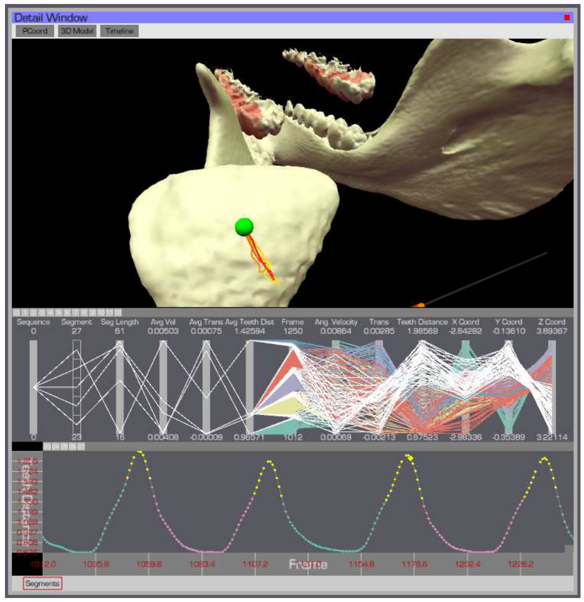


Figure 27: The tracer created in the 3D view window generates new data (x,y,z points over time) that become available for display in the other linked views.

multiples (or side-by-side windows) and overlays each have advantages in comparing motion. The advantage in using overlays is that "counter trends" are easily detectable, but overlays often suffer from visual clutter. On the other hand, side-by-side comparisons are less sensitive to visual clutter, but require more visual real estate to represent the same amount of information and require more time in visual scanning of all the windows. In this framework, the choice of the most appropriate mix of these styles of comparison is left to the user.

In motion overlay views, data for multiple sequences are plotted and rendered together. Figure 28 shows an example. Note that, even in the 3D view, the pig is rendered with two overlaid jaw bones, one corresponding to each of the motion sequences that is being visualized.

Side-by-side comparisons may be established informally by simply arranging zoomed-in coordinated views side-by-side on the screen. Alternatively, data selected in a time plot within the motion overview or any zoomed-in window may also be sent to an existing window, which then resizes to arrange the views appropriately for a side-by-side comparison, as in Figure 30.

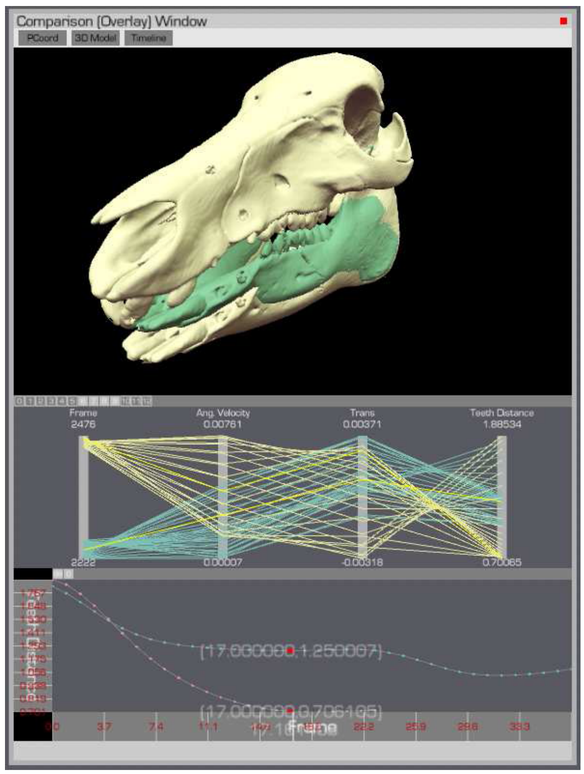


Figure 28: Detailed motion comparisons are supported via overlay-style visualization applied to each of the coordinated views. Data from multiple motion segments are plotted on the same axes and in the same registered 3D space to produce these visualizations.

3.3.4 Application To Experimentally Collected Biomechanics Data

This section describes application-specific implementation details, initial findings, and feedback from domain scientists for the study of pig chewing behavior introduced in section 3.3.1.

3.3.4.1 Processing Motion Data

Before loading the pig chewing motion data into the visualization framework, a simple Matlab script was prepared to segment the motion into cycles. While more advanced time series analysis could certainly be utilized in this step, the approach taken here is quite simple. The angular velocity of the jaw bone is already calculated for these data in a preprocessing step. Using this information, a sign change from negative to positive in the angular velocity is detected. This occurs regularly at the bottom of a chewing motion when the jaw stops opening and begins to close. The data frames where this occurs are saved to a file and loaded into the visualization system to seed the techniques, such as the small multiples display, that work based on segments of the motion data. In all, data from five different trials are loaded into the system resulting in 108 motion segments identified in this manner. All this data can be seen in the motion overview in Figure 24. Note that not all of the sequences correspond to a chewing cycle; some refer to food gathering. The small multiples traces view in Figure 25 provides a visual means for distinguishing chewing and food gathering patterns.

3.3.4.2 Identifying Spatial Relationships and Surface Interactions

Many biomechanical analyses require investigation of patterns in the interaction of surfaces, particularly bone-to-bone surface interaction within joints [114]. Chewing motions are interesting in this regard in that there are three areas of surface-to-surface interaction: the attachment of the jaw to the skull on the left and on the right, and the teeth. The occlusion and grinding patterns of the teeth are of particular interest. Figure 29 shows a series of zoomed-in data views setup as a side-by-side comparison. The vertical distance between the teeth has been calculated and plotted directly on the 3D view as a color map textured to the tooth geometry. These distance data are calculated for each frame of the motion in a preprocessing step and are rendered interactively in the 3D views using a texture-based color map implementation. Within this framework, these data may

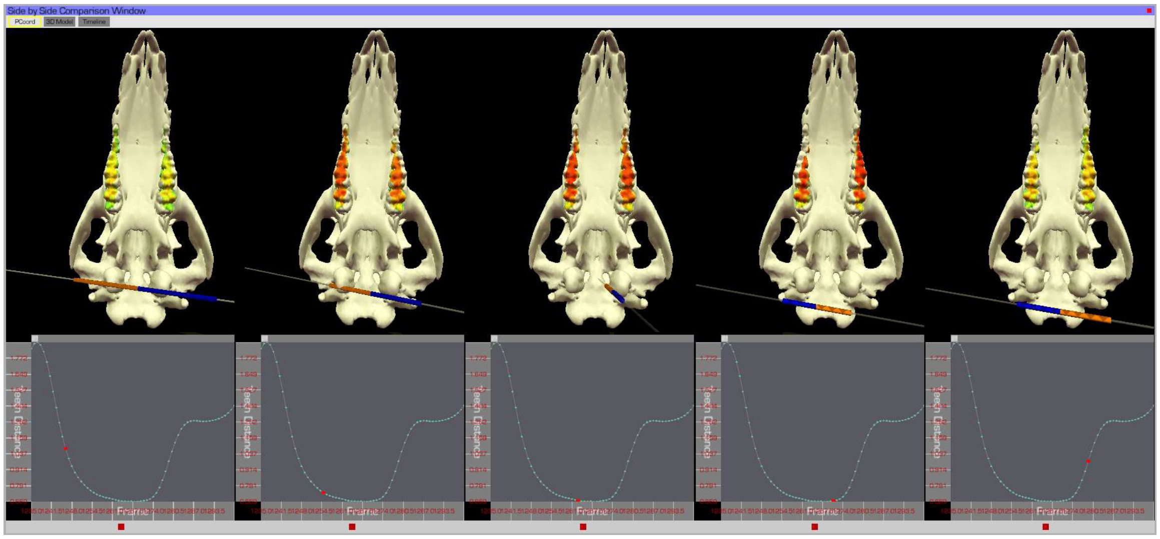


Figure 29: A sequence of side-by-side visualizations that demonstrate how the teeth slide against each other. The 3D view has been rotated so that we are looking up at the top rows of teeth, and the mandible has been hidden from view. A color map has been texture mapped onto the forms of the teeth to encode the vertical distance (defined by the principle axes of the skull) separating the teeth. The chewing sequence advances in time across the views from left to right. A 3D instantaneous helical axis describing the motion of the mandible relative to the skull is also displayed.

be viewed both in the 3D visualization and in 2D plots, where the average distance between the two sets of teeth is a useful variable to explore.

3.3.4.3 Identifying Clusters of Related Motion Sequences

Figure 30 demonstrates the use of multiple linked views for identifying and characterizing related motion sequences. The user has filtered down to a subset of the data (eleven chew sequences) that are visible in the parallel coordinates view and the 2D plot of average distance between teeth vs. frame number. The 2D plot has been arranged to overlay the chewing sequences, starting each at frame zero on the left side of the plot. In this arrangement, two similar clusters of motions are easily distinguishable, with an outlier that does not clearly fit into either pattern. Each cluster is likely to correspond to a different chewing behavior, for example, chewing and food gathering. Using the 2D views, we can clearly identify the two clusters and also explore the amount of variance within each cluster for different data variables.

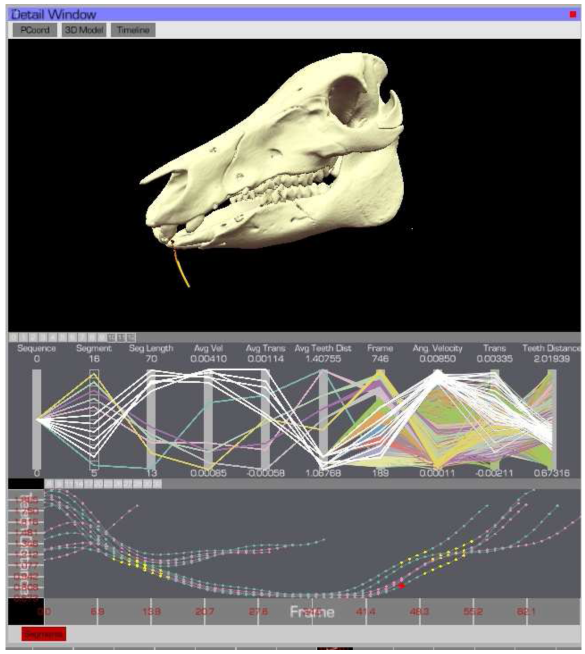


Figure 30: Clusters of motion cycles can be identified using a combination of the 2D plot and the parallel coordinates.

3.3.4.4 Feedback from Domain Scientists

While we have yet to perform an extended analysis of the use of this framework by domain scientists, we have collected some initial feedback from our collaborators based on our current implementation. Two high-level points of feedback are: first, there is widespread agreement that new analysis strategies are needed for working with these data, and second, the framework presented here is a drastic departure from current practice in fields such as evolutionary biology and orthopedics. A key point of departure is the notion that it may be possible to look at all of the data from an experiment at once via the overview visualization methods. During a feedback session with our collaborators, this point sparked considerable discussion, including discussion of the use of tracer lines within the small multiples views and the potential to extend this concept to more sophisticated tracers that also encode velocity, rotation angle, or other variables through color coding or other visual means. This feedback reinforces the importance of exploring the design space of potential small multiples views that are appropriate for motion visualization. During the same session, we also confirmed several characteristic features of pig chewing motions and investigated differences in the motion based upon the type of food (pig chow vs. nuts) through interactive exploration using this framework.

3.3.5 Discussion

In this section we discuss current limitations and future research directions suggested by this work.

3.3.5.1 Scalability of Small Multiples

One question raised by the current framework is, how will the techniques scale to databases of various sizes? The initial overview visualization, including the small multiples view, is perhaps the most interesting aspect of the framework to discuss with scalability in mind. One answer to the question is that the utility of the small multiples visualization scales with the skill and creativity of the user in constructing a useful small multiples display. The display in Figure 25 is useful for discerning some trends across 108 related motion sequences. With fewer motion sequences, alternative views, including those that feature the bones prominently, may be useful. With more motion sequences, this layout and others may still be useful, but certainly at some point the utility of a small multiples display crafted from snapshots of 3D renderings will reach a limit.

A complimentary technique to address the scalability of this small multiples motion overview may be the use of filtering within the small multiples view or within new instances of it in separate windows. Currently this display functions as a complete overview of the entire database; however, such an overview may also be appropriate for a subset of the original data. We have discussed this idea, but have not yet developed an implementation of it. Several interesting user interface issues remain in developing an interactive display of this form that supports fluid, interactive exploration.

3.3.5.2 Scalability and Interactivity

Adding 3D renderings to a multi-window information visualization system requires special design to maintain interactive framerates. While the 2D graphics utilized in typical information visualization techniques are relatively fast to render, 3D scientific visualizations often utilize the full extent of the rendering power provided by current graphics hardware just to render a single view of the data.

Working with the data set described here, our current implementation maintains interactive framerates, depending on the view options set, while rendering on the order of five instances the 3D scene in the filtered and comparison views. This seems to be a minimum level of performance for reasonable analysis using the framework. To extend the framework to applications that involve more intensive 3D rendering, new strategies for addressing multi-view 3D rendering will be required. One potential direction for this research is to use a prioritized rendering scheme, directing more rendering resources to the views that are actively being manipulated. Since the viewers attention is divided between several views in multi-view systems, artifacts in some views may be almost unnoticeable from a perceptual standpoint. An example would be the use of image warping [160] to support linked camera manipulation in several 3D windows. The view in the window that the user is actually manipulating might be rendered as a true 3D scene, while other (lower priority) views might update in an approximate fashion using a faster rendering technique.

3.3.5.3 Alternative Visual Representations for Motion

Previously demonstrated 3D animated small multiples displays have supported rendering just a handful of multiples [27], which give them a very different visual character than the display in Figure 25. With additional technical work to address rendering speed, it should be possible to develop animated 3D small multiples displays of tens to hundreds of animated 3D scenes. This raises the question of whether such a display would be useful in analysis of 3D motions. Our current work was motivated by the finding that static views may outperform animation in analysis of trends [140]. Based on this notion, adding even more animation seems to have more potential to distract than to clarify. Nevertheless, it would be interesting to investigate whether certain classes of 3D motion trends may be discernible through visualization in large-scale animated small multiples displays.

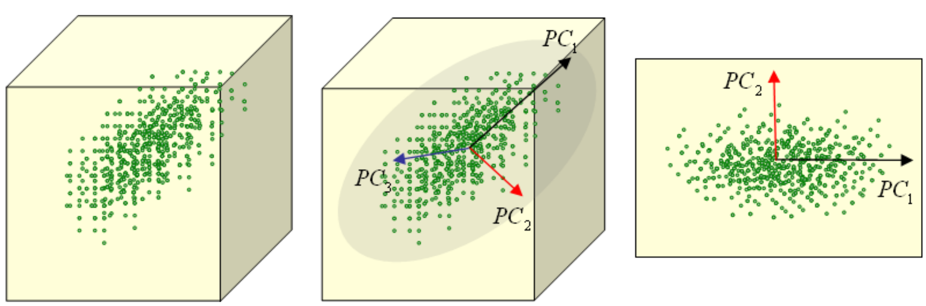
Currently, the most useful static small multiple representations for motion that we have discovered are of the form seen in Figure 25: simple geometric representations that describe motion from multiple frames of data in a single image. In addition to tracers, other visuals commonly found in 3D motion visualization applications may also be useful as small multiple images. Examples beyond tracer particles that fit this description include the average axis of rotation or the ruled surface swept out by an instantaneous axis of rotation over a sequence of frames of motion.

3.3.5.4 New Data Sources

In applying this framework to other data sets, one of the important next issues to address is handling biomechanical systems containing more than two bones, such as the spine or the wrist. The skull and jaw system is a special, rather complicated, case for a two-bone system, in that the jaw has multiple attachment points to the skull (the TMJ on both sides), and the two bones also interact as the teeth come together. Thus, many of the strategies employed here (interacting with 3D views to establish spatial points of interest, using multiple windows to focus on multiple points of interest, etc.) are likely to be relevant to analysis in complex systems with more than two rigid bodies in motion.

3.4 iPCA: Interactive Principle Component Analysis

Principle Component Analysis (PCA) is a widely used mathematical technique for high dimension data analysis. Just within the fields of computer graphics and visualization alone, PCA has been used in many different research areas [93]. At its core, PCA is a method that projects a dataset to a new coordinate system by determining the eigenvectors and eigenvalues of a matrix (Figure 31). This method finds the factors which explain the most variation among data points.



(a) (b) (c)

Figure 31: Illustration of principal component analysis. High-dimensional data (a) are plotted with respect to their first three Principal Components (PCs) (b) and first two PCs (c).

Although PCA is a powerful technique capable of reducing dimensions and revealing relationships among data items, it has traditionally been viewed as a "black box" approach that is difficult to grasp for many of its users [93, 154]. The process and result of the coordinate transform from original data space into eigenspace in PCA makes it challenging for the end user to identify the relationships between the input data and the data after the projection into eigenspace. This is especially problematic for novice users and students who need to use PCA but do not yet grasp how it works. Without a certain amount of background knowledge in the math behind PCA, it is often difficult for the user to perform effective analysis both in understanding how the original data items transform between coordinate systems and how the data dimensions relate to the principle components.

In order to assist the user in better understanding and utilizing PCA for analysis, we have developed a system called iPCA (interactive PCA) that visualizes the results of principle component analysis using multiple coordinated views and a rich set of user interactions. The four coordinated views in our system visualize the data items in original data space (Data View), the data items in eigenspace (Eigenvector View), the data items projected onto two principle components (Projection View), and the correlations between all data dimensions (Correlation View). User interactions in one view are immediately reflected in the others so that the user can easily identify a data item or a data dimension in the original data space and its counterpart in eigenspace.

To demonstrate the effectiveness of iPCA, we performed a comparative user study with a well-known commercial system called Interactive Data Exploration, which is part of SAS/INSIGHT. The two systems are similar in that both systems use the same mathematical functions for performing PCA calculations, but they differ in their approaches to interface and interaction design. While the visualizations and interactions in our system are fluid, dynamic, and coordinated, in SAS/INSIGHT, a more traditional menu-driven and command-line approach forms the basis of interaction. Using SAS/INSIGHT, the user iteratively inputs parameters into the system before clicking on a button (or typing in a command) to initiate the PCA process and generate the results as static images and charts.

Our user study involved 12 participants performing complex analysis tasks on high dimensional data using both iPCA and SAS/INSIGHT. We quantitatively measured the accuracy and speed of the users' analyses, and asked the participants for qualitative feedback on ease of use, preference, and effectiveness. Based on the quantitative results of the user study, we find that users were faster and more accurate in analysis tasks using our system. Participants' feedback indicates that our system better facilitates the understanding of PCA, is more intuitive to use, and is unanimously preferred over SAS/INSIGHT. Many participants attributed the success of our system to its high interactivity and transparency, which suggests that our system is successful in opening up the "black box" of principle component analysis.

The rest of the paper consists of six sections. First we discuss other research in visualizing PCA and the benefits of interaction. Then, we provide our system's interface design and the available sets of interactions. In section 3.4.4, we introduce the evaluation procedures and results, and conclude with discussions, conclusion, and future work.

3.4.1 Related Work

PCA has been applied in many disciplines for various purposes. In visualization, PCA is used mostly for dimension reduction. For example, Hibbs et al. [82] apply PCA to visually analyze microarray data. Wall et al. [178] demonstrate how to visualize gene expression data using PCA and how to interpret the results. However, while PCA is popular and effective as a tool, there have been few available products or research projects on assisting the understanding of PCA results. Mathematical applications such as MATLAB [166] and SAS/INSIGHT [146] can perform PCA and visualize its results accurately. An open-source visualization tool, GGobi [165], supports interactive analysis of data through PCA and can be linked to R (Statistical Computing Software) for additional statistical methods. Müller and Alexa [118] developed a system which allows the user to visually detect and create clusters of data elements in the PCA space. Müller et al. [119] further enhanced conventional information visualizations with PCA and demonstrated that this combination improved data analysis. All these PCA-based tools are powerful and employ various visualization techniques. However, they also share the same goal of utilizing PCA with the assumption that users are experts at mentally transforming data elements from their original space into the projected PCA space. Our work differs in that we intend to use interaction to make the transformation of coordinate spaces

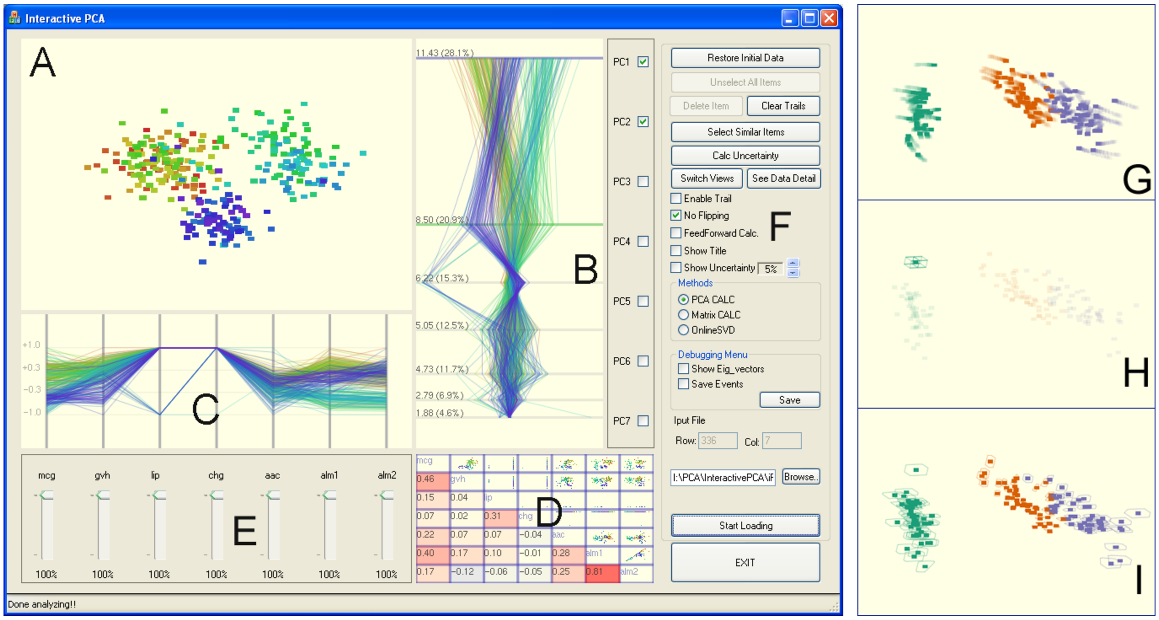


Figure 32: The system overview (left) showing the four views and the two control panels with the E. Coli dataset, and three examples with the Iris dataset (right). (A) Projection view. Data items are projected onto the two user-selected eigenvectors (in this case, the primary and secondary principle components). (B) Eigenvector view. Each eigenvector is treated as a dimension in this parallel coordinates view, and every data item is drawn as a line. (C) Data view. Another parallel coordinates view, but this time each dimension represents the dimensions in the original data, and each line represents each data item. (D) Correlation view. Pearson-correlation coefficient and the relationships (scatter plot) between each pair of variables are represented. (E) Dimension sliders. Each slider controls the amount of contribution of a dimension in the PCA calculation. (F) Control options. (G) shows the result of diminishing the first dimension (Sepal length) of the Iris dataset from 100% to 0%. The trails show how the data points move in PCA space in response to the change. The images (H) and (I) show 10% uncertainty in the data (in all dimensions). The possible locations for each data point are drawn in a hypercube (H) and in outlines (I) corresponding to the number of data item(s) selected.

intuitive to both novices and experts, and to show that by opening this “black box,” users can gain a deeper understanding of data analysis using PCA.

Interaction plays an important role in visualization for assisting users in understanding their data. Several user evaluations have found a benefit for interactive visual systems over traditional iterative input systems in understanding and using data. Ahlberg et al. [4] study the difference between using dynamic sliders and traditional text entry to visually explore periodic table data. They find that participants are faster with the dynamic slider interface on some but not all of their tasks. However, they do not find a clear difference in the participants’ subjective evaluation of the interfaces.

More recently, Callahan and Koenemann [23] compare an interactive visual tool, InfoZoom, against two traditional interfaces for online catalog browsing. With InfoZoom, users are more likely to complete tasks faster. The users also report higher ease of use and efficiency than traditional interfaces. In contrast, Combs and Bederson [32] compare a zoomable image browser to a static image browser and find no difference in performance, although users tend to (non-significantly) prefer the zoomable browser. Unfortunately, while these studies inform us of the value of interaction, the tasks are simpler than asking users to perform complex analysis using PCA. Although some research studies [145, 148] have been performed to understand the effects of interactions in a more complex analysis task, these studies are narrowed to finding the effectiveness and the limitations of their applications.

3.4.2 Interface Design

The overall interface design of our system, iPCA, is based on multiple coordinated views. Each of the four views in the system represents a specific aspect of the input data either in data space or eigenspace, and are coordinated in such a way that any interaction with one view is immediately reflected in all the other views (brushing & linking). The coordination between the views depicts the same data item or data dimension in both data space and eigenspace simultaneously, thus allowing the user to infer the relationships between the two coordinate spaces.

Along with two control panels, iPCA contains four distinct views: the Projection View (Figure 32A), the Eigenvector View (Figure 32B), the Data View (Figure 32C), and the Correlation View (Figure 32D).

Projection View: Two principal components (by default, the first and second most dominant eigenvectors) are used to project data points onto a two-dimensional coordinate system.

Data view: The Data View is located below the Projection View, and shows a parallel coordinates visualization of all data points in the original data dimensions. In this view, an auto-scaling function is applied to increase the readibility of data.

Eigenvector View: In the Eigenvector View, data points are shown in the eigenspace. The calculated eigenvectors and their eigenvalues are displayed in a vertically projected parallel coordinates visualization, with eigenvectors ranked from top to bottom by dominance. The distances between eigenvectors in the parallel coordinate view vary based on their eigenvalues, separating the eigenvectors based on their mathematical weights.

Correlation View: Pearson-correlation coefficients and relationships between variables are represented in the Correlation View as a matrix of scatter plots and values. Since correlations between dimensions are symmetric, repetition is avoided by separating the matrix into three components: the diagonal, the bottom triangle, and the top triangle. The diagonal displays the name of the dimension as a text string. The bottom triangle shows the coefficient value between two dimensions with a color indicating positive (red), neutral (white), and negative (blue) correlations. The top triangle contains cells of scatter plots in which all data items are projected onto the two intersecting dimensions. The colors of the data items are the same as the colors used in the other three views so that clusters are easily identified.

It is relevant to note that the selection operation in all views and the zooming-in mechanism in the Projection and Correlation views help users to focus their interest on a data item or items. Also, the Projection View and the Correlation View can be switched such that the Projection View takes up the lower right hand position and the Correlation View fills the main display. This simple switch operation allows the user to utilize the visual real estate for focusing either on a single projection of data or to examine in detail all (or one) scatter plot(s) in the Correlation View.

The two control panels include a set of dimension sliders (Figure 32E) that can be used to decrease or increase the contributions of each of the original data dimensions, whose purpose will be discussed further in the following section (Section 3.4.3). Several additional modes can also be specified in the other control panel to enhance understanding of the visual changes during data analysis (Figure 32F). The user can enable *trails* so that the path of each data point's recent motion is painted to the screen, making the movement of each point during interaction operations more apparent. The user can also choose to show *uncertainty* (Figure 32H and I) by setting a percentage of possible error in the dataset, which is reflected as bounding boxes around data items in the Projection View.

3.4.3 Interaction

Since iPCA is designed with high interactivity in mind, the types of available interactions are carefully considered. We categorize all the interactions in iPCA into two groups: interactions with the views, and interactions with PCA. Interactions with the views are operations that do not result in PCA calculations, and include brushing, filtering, zooming and panning, etc; whereas interactions with PCA will result in new PCA calculations, including operations that change the weights of dimensions, move data points in either data space and eigenspace, and removal of data points. Both types of interactions are embedded in the coordinated views such that all views react to all interactions.

3.4.3.1 Interacting with the Views

Interactions in this category are operations that do not cause the system to recompute PCA. As mentioned above, these operations include brushing, filtering of data items or dimensions, zooming and panning, etc. Although these interactions are standard in most InfoVis or visual analytics tools, they are nonetheless very important, and are essential in multiple coordinated views. The ability to allow the user to select a cluster of data items in one coordinate space and immediately see the corresponding items highlighted in the other coordinate space helps the user understand the relationship between the two.

The most notable interactions in this category are the different types of selections implemented in iPCA. iPCA allows the user to select data items in all four views. In Data View and Eigenvector View, where the visualizations are parallel coordinates, selection means clicking on a single line or brushing a range of items. In Projection View and Correlation View, the user can either click on a single dot or draw an enclosed space upon which all data items within the space will be selected.

3.4.3.2 Interacting with PCA

As mentioned previously, one of the biggest hurdles in effectively analyzing PCA results is in understanding the relationships between data space and eigenspace. While the interactions provided in the previous section allow the user to see a data item appear in different coordinate systems, the interactions do not immediately lead the user to see the relationship between the coordinate spaces. Specifically, eigenvectors are linear combinations of data dimensions, therefore, understanding which data dimension contributes to an eigenvector is a key point in comprehending how the coordinate spaces relate to each other.

In order to visually assist the user in recognizing how data space relates to eigenspace, we create a set of interactions that allow the user to alter the values of the data items. For example, if the user drags a data item in the Projection View towards the positive direction along the x-axis (increasing the data points value in the first principle component), the user should be able to immediately observe in the Data View how that change affects the values of that data item in the original data space, thus shedding light on the relationship between the first principle component and all dimensions in the original data space.

Similarly, if there is an obvious cluster in the Projection View, the user can interactively change the weights of a dimension to see its affect on the formation of the cluster. For example, if diminishing the contribution of a data dimension in PCA calculation down to 0% does not affect the clustering, then it should be clear that the cluster does not depend on that particular dimension.

While the concept of encouraging interactions that directly alter the values of data items seem counter-intuitive, the idea is not novel. *Spotfire* includes a “jitter” operation [3], and *Dust and Magnet* has a “dust shake” operation [192], both of which are designed to reveal occluded data items. In medical visualization, deformation or “cut-aways” modify the data to expose hidden structures underneath skin and flesh [115]. The interactions in iPCA share a similar goal, but instead of revealing hidden or occluded information, our interactions assist the user in revealing relationships between coordinate spaces.

Three specific interactions are implemented based on the concept of data alteration: modifying dimension contribution, adjusting data items, and removal of data items.

Modifying Dimension Contribution: Each slider in Figure 32E corresponds to a data dimension. By modifying the slider, the user can change the contribution of the data dimensions in the final PCA calculation. For instance, changing the dimension contribution to 50% indicates the weight change of the selected dimension to 0.5. This interaction allows the user to observe which data dimensions contribute to the projections of the data in eigenspace. By adjusting these sliders, a user can quickly test hypotheses about how the analysis would be affected if a dimension or set of dimensions were removed or considered less important. This makes it possible for a user to observe the formation and dispersion of clusters and to identify the cause of outliers.

Adjust Data Items: Values of data items can be modified in either the Projection View, Data View, or Eigenvector View. This interaction not only allows the user to see the relationship between a principle component and the contributing data dimensions as mentioned above, but also allows the user to test what-if scenarios. If the user suspects that a data item should appear in a certain cluster, the user can manually move the data item and see how the values of that data item would have to be modified.

Removing Data Items: In analysis using PCA, a common task is for the user to remove outliers. iPCA supports direct removal of data items from the system so that the user can observe how the projection from data space to eigenspace changes with the removal.

One caveat of these interactions is that they are computationally expensive. Modifying any data requires the re-computation of PCA, and in the cases of interactively adjusting sliders and moving data items on screen, PCA has to be re-calculated quickly to avoid lag or flickering. For very large datasets, this type of interactions has the potential of becoming a bottleneck in usability. In iPCA, the scalability issue is addressed by incorporating a faster version of singular value decomposition called online-SVD [18] which trades precision for speed. Brand demonstrates how online-SVD is faster than traditional SVD (see [18] for detail). The user has the option to use either traditional SVD or online-SVD depending on the speed and accuracy requirements as well as the scale of the data.

3.4.4 Evaluation

We conducted a comparative evaluation to assess the effectiveness of our system in relation to a well-known commercial tool, SAS/INSIGHT's Interactive Data Exploration. A total of 12 students (nine males) participated in the evaluation. Three of the 12 participants were undergraduate students

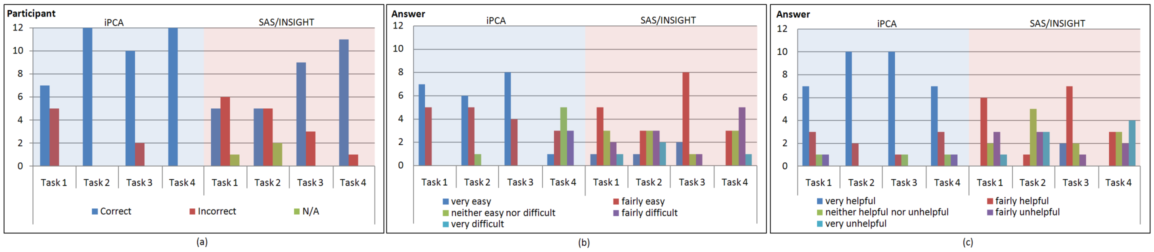


Figure 33: Results broken down by tasks for each of the two systems. (a) Number of participants who answered the task question correctly. (b) Task difficulty and (c) helpfulness of the system in solving the task, as reported by participants.

and nine were graduate students, and 11 of the participants majored in computer science and one in management of information science. Based on self reported familiarity, we found that nine participants were aware of PCA prior to the evaluation, and of the nine, three had used PCA in the past.

At the start of the evaluation, all participants receive a detailed explanation about PCA followed by a pre-evaluation background questionnaire. Each participant was provided a total of ten minutes to train with the two systems prior to the evaluation. The evaluation consisted of performing four analysis tasks using each system. The participants were given five minutes to perform each task and were requested to answer questions immediately after each task. The evaluation was conducted using an online website, where time spent and answers were saved into a database.

We performed the evaluation using three different datasets: the Iris dataset (150 data items $\times$ 4 dimensions), the E.coli dataset (336 data items $\times$ 7 dimensions) and the Wine dataset (179 data items $\times$ 13 dimensions). The Iris dataset was used in the training session whereas the E.coli and the Wine datasets were used in the actual evaluation. All three datasets are scientific results that are publicly available at the UCI Machine Learning Repository [10].

3.4.4.1 Procedure

Each participant was requested to use the two systems on different datasets. Therefore, six participants used iPCA first and the rest of the participants began with SAS/INSIGHT. The order in which datasets were given to each participant was counterbalanced with system order, so that six participants used the E.Coli dataset first and the rest used the Wine dataset first.

Four tasks were given to each participant during the evaluation of both systems:

- What is the most striking outlier you can find? An outlier is a point that does not fit the overall patterns of the dataset.
- Find a dimension that least affects the PCA outputs in the Projection View using first and second principle components.
- Find two dimensions with a highly positive correlation. Also find the class name and label of an outlier that does not follow that correlation.
- How does removing the first dimension affect the PCA results using the first and second principle components? List as many observations as possible.

The first three tasks are related to finding exact answers and the last one is a descriptive task asking the participant to describe the difference between including and excluding a specific dimension. Five minutes were given to solve each task. If time expired, partial answers were saved into the database. As soon as each task was completed, a post-task questionnaire was given to participants to track how they felt about the task. These questions included “How difficult was this task?” and “How helpful was the interface in solving the task?” A post-application questionnaire was given after a participant completed all four tasks. This questionnaire asked the participant to give feedback on their overall subjective opinion about each system. After a participant completed the evaluation using both systems, the participant completed an additional set of questions (post-study questionnaire) that described the preference, the ease of use, and the effectiveness of the system in analyzing data. Finally, the participant graded each system on a scale of ‘A’ to ‘F’.

3.4.4.2 Results

We present the results of our evaluation based on accuracy, speed, difficulty and usefulness, effectiveness, and preference. Both accuracy and speed are measured quantitatively; whereas the other three categories are analyzed based on the participants’ qualitative feedback.

Accuracy: Figure 33(a) shows the results of the participants’ accuracy in solving each task using both iPCA and SAS/INSIGHT. As shown, approximately 85% of the participants answered correctly using iPCA. On the other hand, when using SAS/INSIGHT, they were only able to answer correctly 62% of the time. Furthermore, when using SAS/INSIGHT, there were three instances in which a participant could not complete the task. One of the instances was due to the fact that the participant ran out of time. In the other two cases, the participants simply gave up and claimed that they were unable to find the solutions (see Figure 33(a)). Note that the accuracy difference is statistically significant across the two systems ($p < 0.01$).

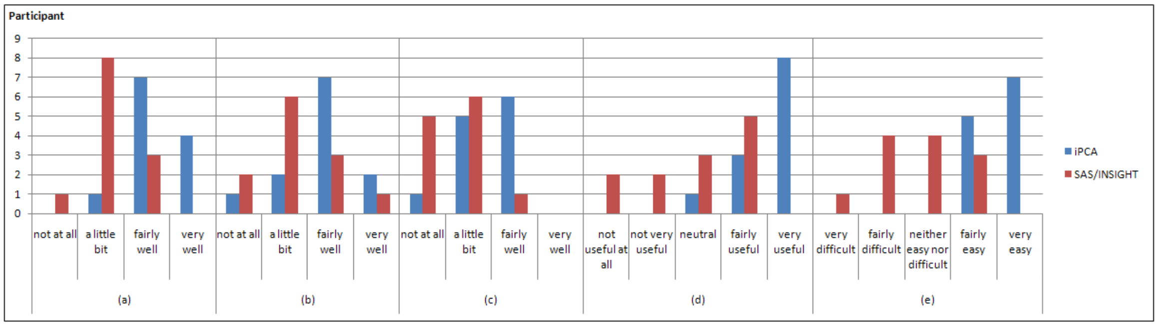


Figure 34: Participants' responses to a post-application questionnaire, filled out after solving all four tasks using one of the systems. (a) How well do you understand the application now? (b) How well do you understand PCA now? (c) How well do you understand the data you worked with now? (d) How useful was the system? (e) How difficult or easy was the system to learn?

Speed: Table 5 shows the overall average time spent solving each task. Participants spent less time in solving each task using iPCA except for task 4 (a descriptive question). This seems to be because participants tried to find as many differences as possible through interaction with dimensions.

On average, participants spent about 150 seconds using iPCA, and 170 seconds using SAS/INSIGHT. Although the difference is not statistically significant ($p = 0.17$), there is a trending effect towards a faster solution when using iPCA.

Table 5: Average time spent in solving each task.

Application	Task	Time Spent (seconds)
iPCA	Task 1	136.58
	Task 2	128.33
	Task 3	125.50
	Task 4	211.33
SAS/INSIGHT	Task 1	177.67
	Task 2	165.58
	Task 3	142.92
	Task 4	197.08

Difficulty & usefulness (post-task questionnaire): Figure 33(b) and (c) show how participants rated the difficulty of each task when using iPCA or SAS/INSIGHT, as well as how they rated the usefulness of each system in solving the task. Figure 33(b) indicates that about 81% of the participants found the tasks to be easy when using iPCA. On the other hand, only about 48% of the participants identified the tasks as being easy when using SAS/INSIGHT. Interestingly, although more than half of the participants mentioned that task 1 is easy to solve, Figure 33(a) indicates that the accuracy in solving task 1 is low (58% iPCA and 41% SAS/INSIGHT). This might be because most participants have little previous experience with finding outliers.

Figure 33(c) shows that about 90% of the participants identified iPCA to be helpful in solving the tasks; whereas only about 40% of the participants found SAS/INSIGHT to be helpful. Furthermore, only two participants (participant D and I) rated iPCA to be not helpful in solving a task (tasks 4 and 1, respectively); whereas ten participants indicated that SAS/INSIGHT was unhelpful in solving some tasks (one participant indicated SAS/INSIGHT was completely not helpful in solving all tasks).

Overall, we find that the more difficult a task was rated (very easy = 5, fairly easy = 4, etc), the more time the participants spent on solving it ($p < 0.0001$). However, solving a task with a "helpful" system (very helpful = 5, fairly helpful = 4, etc) did not decrease the time spent on the task ($p = 0.2586$). With a helpful system, the participants did solve the tasks more accurately ($p = 0.0027$), but participants did not rate the tasks to be less difficult ($p = 0.0966$).

Effectiveness (post-application questionnaire): Figure 34 shows the results of the five questions in the post-application questionnaire conducted right after the evaluation of each system. Of particular significance are the questions asking the participants how well they understood the application (Figure 34(a)), how well they understood the data (Figure 34(c)), how useful was the system (Figure 34(d)), and how difficult or easy was the system to learn (Figure 34(e)).

In answering how well the participants understood PCA, most participants did not indicate that they understood PCA "very well." However, the majority of the iPCA users indicated that they understood PCA "fairly well"; whereas the majority of the SAS/INSIGHT users only claimed "a little bit" of understanding.

In answering how well the participants understood the data, the majority of the iPCA users indicated that they understood the data either "fairly well" or "a little bit"; whereas the SAS/INSIGHT users either understood the data "a little bit" or "not at all."

Lastly, in answering about the usefulness of the system, the majority of the iPCA users found the system to be "very useful"; whereas SAS/INSIGHT users consistently ranked the system to be "fairly useful" and below, with four participants claiming the system to be "not very useful" or "not useful at all."

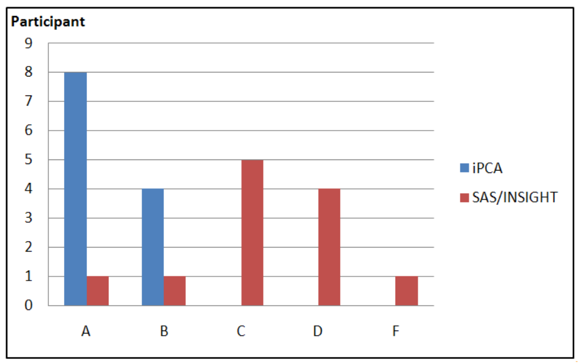


Figure 35: At the end of the evaluation, each participant grades the systems on a scale of 'A' to 'F'.

Preference (post-study questionnaire): After the evaluation, each participant ranked the two systems and described their pros and cons. Figure 35 clearly shows that most participants preferred iPCA over SAS/INSIGHT, giving iPCA eight A's and four B's. On the other hand, the majority of the participants gave SAS/INSIGHT a C or D grade, with one participant failing it by giving it an F.

When describing the pros and cons of iPCA, eight participants specifically pointed out the strength of iPCA as being "interactive," and eight participants described iPCA as "transparent." While a few participants gave constructive feedback on how to further enhance the iPCA tool (e.g., add the ability to rearrange the dimensions in the Correlation View), the only negative criticism for iPCA was that it did not generate printable reports similar to the static charts and numbers that SAS/INSIGHT generates.

For SAS/INSIGHT, two participants who were previously familiar with the SAS system pointed out that while they preferred iPCA over SAS/INSIGHT for analyzing PCA results, SAS is still a far more comprehensive and complete numerical and statistical analysis tool. One participant further noted that if he was allowed to use additional features in SAS outside of the tools specific to PCA, deeper analysis on the dataset could have been performed.

3.4.5 Discussion

Since our evaluation compared two systems (iPCA and SAS/INSIGHT) that use the same mathematical methods for computing PCA, we can safely assume that the increase in our participants' performance in using iPCA is attributed solely to the interface design and the set of interactions. Unfortunately, we are not able to further isolate the specific factor(s). Based on our evaluation alone, we cannot determine if the increase is due to the multiple coordinated views, the interactions, or the combination of the two. However, we do hypothesize that the "interactions with PCA" play a significant role in that the user's direct and continuous manipulation with PCA is rewarded with immediate visual feedback. This allows the user to "play" with the data and intuit the subtleties behind the coordinate transform between data space and eigenspace in a way that less interactive visualizations such as SAS/INSIGHT cannot achieve.

The "interactions with PCA" are also the most unique set of the interactions in iPCA. Unlike our other interactions that merely highlight or explore the data, the design decision behind the "interactions with PCA" is to focus on reasoning. In fact, we design the "interactions with PCA" to be less faithful to the data, but more revealing in discovering relationships between coordinate spaces and data dimensions. For example, most of our participants credited the rich interactions in iPCA to be the primary strength of the system, but two of our participants pointed out the fact that in modifying dimension contribution, moving a slider from 100% to 72% and taking a snapshot of the Projection View was not meaningful as the projection was not of the original data. Similarly, moving a data point across the screen seemed counter-intuitive as it directly modified the values of the data. While these concerns are valid, we contend that they miss the spirit of the interactions. It is true that the resulting images from these interactions cannot be considered by themselves, but it is during the direct manipulation of the data and coordinate spaces that the user gains insight about their relations and how changes in one affects the other, which is otherwise hidden. One very interesting future direction for our research will be to further understand why these types of interactions are successful, and examine the extent to which they can be applied.

CHAPTER 4: INTERACTION CAPTURING

In the previous two chapters, we first identified the need of maintaining interactivity through feature-preserving simplification of large datasets in visualizations. We then demonstrated that such simplification is necessary because interaction is an important aspect to visualization and visual analytics. However, an underlying assumption about interaction has yet to be addressed: since the beginning of this thesis, we have made the argument that interaction is an important component to analytics. Numerous sources have stated that it is through interactions that analysis and knowledge building occur [131, 190, 167], but just how much of a user's analysis and knowledge are actually in the user's interactions?

To answer this question, we set off a series of steps to better understand what is in a user's interactions. We begin by recruiting real financial analysts (from large financial institutes such as Bank of America, Wachovia, etc.) and monitor how they use our visual analytics system WireVis (section 3.1). In think-aloud sessions, analysts' interactions with the WireVis system were logged, screen-captured, and audio and video-recorded. After the sessions, questionnaires and interviews were conducted for the analysts to explain in more detail what their strategies, findings, and methods were.

In section 4.1, we present the two visualizations that were implemented to examine the analysts' interactions. The two system corresponds to the two types of analysis: operational and strategic. In visualizing operational analysis, an analyst's interaction logs are presented in a low-level manner such that each specific interaction is depicted visually in a temporal manner. In visualizing strategic analysis, the visualization shows the high-level goals of an analyst such that an analyst's focus on specific accounts, keywords, or dates would be apparent. The two visualizations are integrated such that one could initiate the other during an analysis process.

In the section after (section 4.2), the actual study with the financial analysts and its results are presented. The study consists of three parts. The first part is the recording of the analysts' interactions and their explanations of their activities. The second part consists of graduate students (who have no financial analysis background) using the two visualizations described above to analyze the recorded interaction logs of the analysts. Based solely on the visualization of the interaction logs, the graduate students were asked to take "guesses" of what they believed to be the analysts' strategies, methods, and findings. The third part of the study then compares the results of the analysts self-report with the graduate students' guesses. The resulting similarity, we propose, would indicate the amount of analysis and knowledge that could be recaptured from interaction logs.

The results of this study are significant. We found that up to 60-80% of the analysis could in fact be captured and extracted. Although the study is limited in that we studied a very specific visualization (WireVis) using two proprietary and non-generalizable interaction log visualizations, this study nonetheless serves as an example of the potential amount of analysis and knowledge that could reside in interaction logs. In order to answer the question of generalizability of the study and the results, the third section of this chapter (section 4.3) takes a step further and presents a model of the relationship between interaction capturing and visualization design. The model examines two aspects of the relationship: the modes of capturing (internal vs. external), and the criteria for assessing the effectiveness of the capturing. By understanding the different modes of capturing, the designer of the visualization (and the capturing mechanism) can have more complete sense of the limitations and capabilities of internal and external capturing methods based on the deployment environments. Once the mechanisms have been decided, the designer can utilize the three criteria presented in this model to determine and possibly predict how well the visualization could capture its users' intents and reasoning processes.

The path to perfectly understand how a user interacts with a visualization is still unclear, but the three sections presented in this chapter hopefully serves as an initial step towards a potential viable direction that could shed light into the relationship between a user's interactions in a visualization and the reasoning and analysis processes behind them.

4.1 Interaction Log Visualization

As the fields of information visualization and visual analytics mature, evaluation and verification become increasingly important. Recently, there have been tremendous efforts in such evaluation and verification, most notably the establishment and success of the BELIV workshops (Beyond time and errors: novel evaluation methods for Information Visualization) [1]. Some of the active research topics in this area include the evaluation of techniques, the measurement of insights, and metrics and methodologies for user studies. Together, these efforts are gradually validating the usefulness and power behind the science of information visualization and analysis.

In this project, we take a slightly different approach to the evaluation of visual analytics. Instead of identifying the usefulness of a visual analytical tool or measuring the amount of insight a user gains through the tool, we assume that the tool we are testing is useful and that the user can discover the unknown using it. The questions we ask are: what is the user thinking while approaching the task, what do they actually do, and how do the two relate to one another in the context of the analytic process?

If we can answer these questions, we can extract an analyst's interactions in a meaningful way, discover the reasoning behind the analysis process, and more importantly, relate the two to extract successful analysis strategies.

We performed a study that involved ten participants using the financial visual analytical tool WireVis (section 3.1) to investigate a synthetic wire transaction dataset that contains known suspicious activities. WireVis is a tool developed and tested with financial analysts at Bank of America, and is now in deployment at the bank's wire investigation center WireWatch. The participants' interactions were logged into files and captured on video and audio. Through interviews and analysis of the captured session, we annotated each user's session with his or her reasoning process and discoveries.

The resulting annotations for each user are stored in a document containing both time stamps and unstructured text descriptions of the user's analytical steps using the WireVis tool. Although this document contains a great deal of information on the user's reasoning process, it does not explain what the user actually does in executing the reasoning process. Conversely, the log file of the user's interactions describes the user's interactions, but not the user's thinking during the investigation.

While the two sets of data depict two different aspects of a user's analysis process, neither is complete on its own. We believe that it is the combination of the two that gives us a holistic view of how the user interacts with the analytical tool for executing specific strategies in the investigation process. With the two sets of data combined, we can begin to fully understand the user's analytical process.

We combine the two sets of data through the use of visual analytical tools. We have implemented two separate tools for visualizing and analyzing the operational and strategic aspects of the user's investigation process. On the operational side, we use a time-based view to examine a user's interactions with the annotations. On the strategic side, we use a treemap approach that combines multiple users' data to identify similarities and differences in the users' foci during their investigations.

There are two main contributions in our work. First we introduce a methodology for capturing user interaction and reasoning in financial fraud investigation. We then apply a visual analytical approach to analyze and relate the user's interactions with this reasoning process. We present two visual analytical tools for discovering both the operational and the strategic aspects of a user's investigation.

4.1.1 Related Work

We roughly categorize the current research in visualization and visual analytics for capturing the reasoning process of an analyst into three groups: evaluating visualization by measuring insight, capturing the user's interactions, and interactive construction of the reasoning process using a visual tool.

4.1.1.1 Evaluating Visualization by Measuring Insight

It has been proposed that the purpose of visualization is insight [123], and that the evaluation of any visualization is to determine the degree to which the visualization achieves this purpose. However, due to the elusive nature of insight, several varied definitions and measurement strategies have been proposed.

Using the definition that an insight is a unit of discovery, Saraiya et al. used a think-aloud method to evaluate bioinformatics visualizations [143, 144]. Rester and Pohl applied different methods to evaluate a visualization called Gravi++ that includes the use of insight reports [136]. More recently, Yi et al. performed extensive literature review and created a categorization of how people gain insight through information visualization [191].

Similar to these efforts, our work relies on the users telling us when they have made a discovery or gained insight. However, instead of using the result to determine the usefulness of a visualization, our interest is in correlating the reasoning and discovery process directly with the users' interactions.

4.1.1.2 Capturing User Interactions

Capturing user interactions for the purpose of understanding the user's behavior is very common both in academics and industry. Commercially, there are many off-the-shelf applications that range from capturing a user's desktop activities such as usability software to interactions on a website (which is a common feature in most web servers).

In the field of visualization, one of the most notable systems for capturing and analyzing user activities is the GlassBox system by Greitzer at the Pacific Northwest National Laboratory [71]. The primary goal of the GlassBox is to capture, archive, and retrieve user interactions [34]. However, it has also been shown to be an effective tool for capturing specific types of interactions for the purpose of intelligence analysis [35]. While GlassBox and most usability software are effective tools for capturing user activities, they focus primarily on low level events (such as copy, paste, a mouse click, window activation, etc), whereas the events captured in our system are at a higher level that corresponds directly to the data (such as what transaction the user clicked on). For more information on the differences in these two approaches, see the work by Heer et al. [78].

More recently, Jankun-Kelly et al. [91] proposed a comprehensive model for capturing user interactions within a visualization tool. Their work is unique in that they focus on capturing the effects of the interactions on the parameters of a visualization. Although it is unclear how this framework supports higher level event capturing, the direction is interesting and could lead to a more uniform way of capturing user interactions.

The systems and approaches above are all proven to be innovative and effective. However, their objectives differ from our goal in that none of these systems fully addressed our question of how much reasoning process can be recovered through the examination of interaction logs. It is with this question in mind that we expand on this area of research to capturing user interactions and look to extract reasoning processes embedded in them.

4.1.1.3 Interactive Construction of the Reasoning Process

An alternative approach to retrieving reasoning through interactions is for the analyst to create a representation of the reasoning process (usually in the form of a node-link diagram) while solving a complex task. There are a few recent systems in this domain, most notably the Aruvi framework by Shrinivasan and van Wijk [156], which contains three main views: data view, navigation view, and knowledge view. Data view is the visual analytical tool itself, navigation view is a panel for visually tracking the user's history, and lastly the knowledge view allows the user to interactively record his reasoning process through the creation of a node-link diagram.

Similarly to the Aruvi framework, the Scalable Reasoning System (SRS) by Pike et al. [130] allows its users to record their reasoning processes through the creation of node-link diagrams. However, unlike the Aruvi framework, the SRS focuses on the collaborative aspects of organizing the reasoning processes among multiple users and sharing their results across the web.

Most recently, Heer et al. [78] created a tool for visualizing users' histories within the commercial visualization tool Tableau [113]. Although the emphasis of this work is not on constructing or visualizing the reasoning process, the functionalities within the tool that allows for a user to edit and modify his interaction history could be used towards communicating his reasoning process effectively.

While there has not been a formal comparison between interactively constructing the reasoning process as mentioned above and our method of analyzing interaction logs, we hypothesize that the cognitive load of having to perform analytical tasks while maintaining and updating a representation of the reasoning process could be tiring [70]. We believe that the systems mentioned above will have better representations of the user's reasoning process. However, we argue that a transparent, post-analysis approach offers an alternative that can achieve comparable results without the effort from the analysts. Most likely the best solution is somewhere in between, and we look forward to analyzing the pros and cons of the two approaches.

4.1.2 User Experiment

In order to understand the user's reasoning process through his interactions, we conducted a qualitative, observational study of a financial visual analytical tool WireVis which was described in a previous section (section 3.1). WireVis was developed jointly with wire analysts at Bank of America for discovering suspicious wire transactions. It is currently installed at Bank of America's wire monitoring group WireWatch for beta testing. Although it has not been officially deployed, WireVis has already shown capabilities in revealing aspects of wire activities that analysts were not previously capable of analyzing.

4.1.2.1 Synthetic Data with Embedded Threat Scenarios

To preserve the privacy of Bank of America and their individual account holders, we created a synthetic dataset for the purpose of this study. Although none of the transactions in the dataset are real, we captured as many characteristics and statistics from real financial transactions as we could and modeled the synthetic data as closely to the real one as possible. The approach we took is loosely based on the methods described by Whiting et al. in developing the 2006 VAST contest [184].

For the purpose of the user experiment, it is important that the dataset is simple enough that users are able to look for suspicious transactions within the time frame of a study, but is complex enough that interesting and complicated patterns can be found. The synthetic dataset therefore contains 300 financial transactions involving approximately 180 accounts. Twenty-nine keywords are used to characterize these transactions, with some of them representing geographical locales (such as Mexico, Canada), and some representing goods and services (such as Minerals, Electronics, Insurance, Transportation, etc.) Each record of a transaction consists of the transferred amount, the sender and receiver's names, the date, and one or more keywords relating to the transaction.

Four different types of known threat scenarios were identified. Two cases of each of the four types were created and embedded into the synthetic dataset based on the approach proposed by Whiting et al. [185]:

Incompatible Keywords in a Transaction: Transactions with two or more keywords that do not belong together. For example, a transaction containing the keywords "car parts" and "baby food".

Accounts with Dual Roles: An account that has had transactions of different incompatible keywords is questionable. For example, an account that transacts on "gems" at one time and "pharmaceuticals" at another.

Keywords with Large Amounts: Transactions of certain keywords are expected to have corresponding dollar amounts. For example, a transactions from a local store on "arts and crafts" should not be in the millions.

Change in Amounts Over Time: An account with an established temporal and amounts pattern receiving a large sum outside of its norm should be examined further. For example, an account with a steady deposit of paychecks of fixed amounts on regular intervals receiving a transaction of a large amount.

4.1.2.2 Participants

Ten users volunteered for our user experiment, seven male and three female. All of the study participants were students at the University of North Carolina Charlotte. These participants were chosen based on their graduate-level research experience in either visualization, human-computer interaction, or virtual reality. Some of the participants have seen demos of WireVis or read the publication, but none had ever used the WireVis tool for analysis. All participants were familiar with InfoVis and visual analytics concepts.

The user experiment was conducted in the Human Computer Interaction (HCI) Usability Lab at UNC Charlotte. The goal of the experiment was to understand the users' behaviors and strategies as they interacted with the visualization while performing fraud detection analysis.

At the beginning of each study, each participant was trained on the concept of wire fraud and the use of WireVis for 10 minutes. We introduced each of the views and features, and then walked the participant through two scenarios of finding suspicious activities. The participant was provided a one-page overview of the functionality of WireVis and encouraged to ask questions. We also provided hints as to what kinds of patterns the participant might investigate in his analysis. In the real world, an analyst's training, experience, and actual world events would lead him to develop these strategies.

Following the training, the user was asked to spend 20 to 30 minutes using WireVis to look through the dataset to find other suspicious activities. We asked the participant to think-aloud to reveal his strategies. We specifically encouraged the participant to describe the steps he was taking, as well as the information used to locate the suspicious activities. Once the user identified a transaction as suspicious, he was encouraged to continue looking for others until the time limit was reached. At this point, a post-session interview was conducted for the participant to describe his experience and additional findings.

4.1.2.3 Capturing User Interaction and Reasoning

Several methods were used to capture each participant's session as thoroughly as possible. A commercial usability software Morae was used to capture the screen, mouse clicks, and keystrokes of the user's interactions. A separate webcam was used to record the user's audio and actions during the session. Lastly, functions built into the WireVis system captured the user's interaction with the tool itself.

Unlike Morae, the interactions captured directly in the WireVis system contain semantic information relevant only to the WireVis system. Instead of recording every mouse movement or keystroke, WireVis only captures events that generate a visual change in the system. For example, a mouse movement that results in highlighting a keyword in the Heatmap view will generate a time-stamped event noting that the user has highlighted a specific keyword.

Since the participants were encouraged to think-aloud during their sessions, we were able to review the video and audio recordings of the participants to recreate their reasoning and thinking processes. We ("the observers") used recordings from each participant's think-aloud session and post interview to construct a hierarchical annotation file that described in text different levels of their strategic and operational approaches in investigating fraudulent activities.

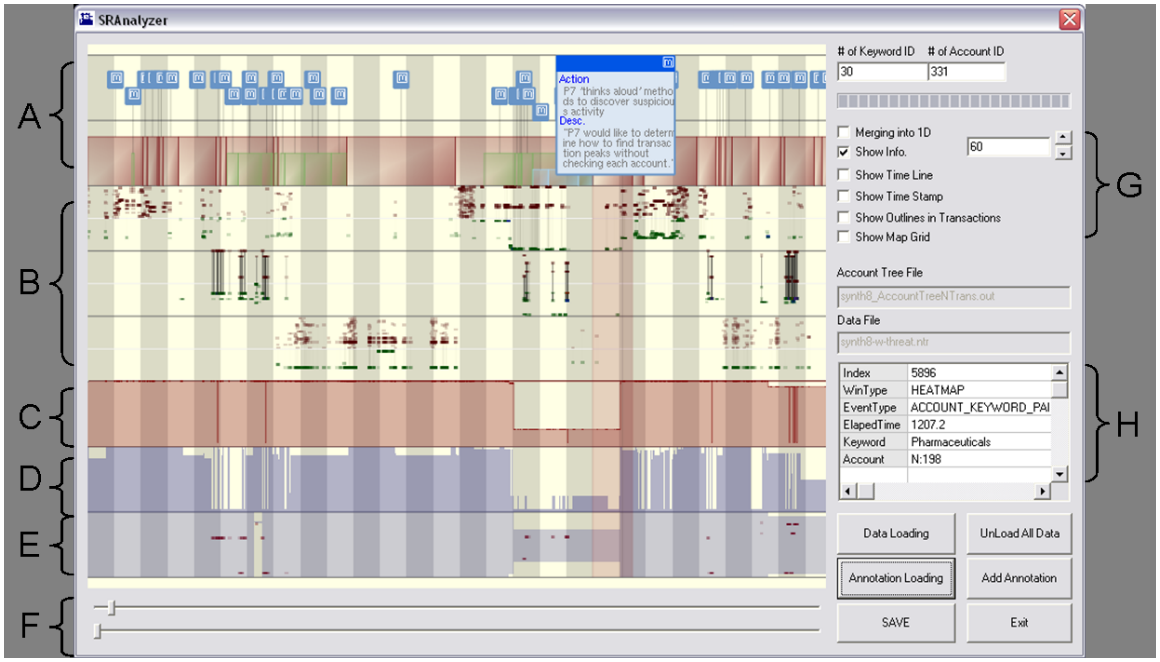


Figure 1: Overview of the operation analysis tool. (A) shows the participant's annotations. (B) shows the participant's interactions with the three views in WireVis. (C) represents the depths of a participant's investigation. (D) shows the areas of the participant's investigation, and (E) the time range. Sliders in (F) control the time scale, while checkboxes in (G) change various visualization parameters. (H) shows the detail information of a participant's selected interaction element.

4.1.3 Visual Analysis of User Interaction and Reasoning

The interactions and reasoning processes of our participants were very complex. Similar to intelligence analysts, our participants had varied preconceived knowledge about the keywords and their meanings. In interacting with WireVis, the participants also had disparate interpretations of the visualization in determining what appeared to be suspicious. Even when the participants identified the same embedded threat scenarios, their methods and approaches were decidedly dissimilar.

The complexity of a participant's interaction and reasoning process makes it very difficult to describe his behavior and intent quantitatively. We therefore propose approaching this complex problem using visual analytical methods. Two visual tools are created to investigate the operational and strategic aspects of a participant's investigation process. Using the operation analysis tool, we can analyze a participant's interactions in accordance to his reasoning process; while using the strategy analysis tool, we can compare the strategies of different participants or groups of participants. Together, these two tools provide the means to discover the relationship between user interactions and reasoning processes at all levels.

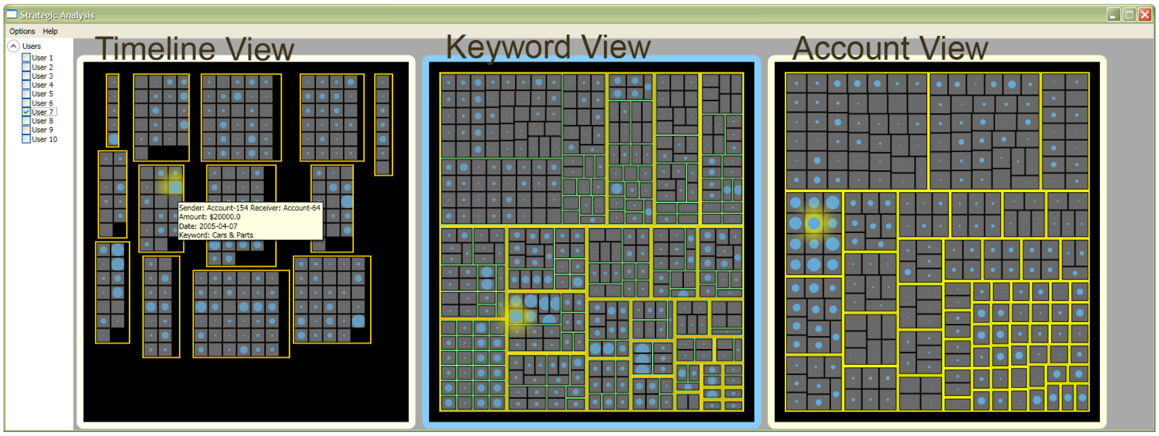


Figure 2: The same participant's interactions in Figure 1 are shown in using the strategy analysis tool. The left view shows transactions grouped by time, middle view shows grouping by keywords, and the right view shows grouping by accounts. The patterns in the account view indicate that the primary strategy employed by this participant was to look for accounts with similar activities.

4.1.3.1 Operation Analysis Tool

Our operation analysis tool is designed to support the analysis of a participant's operational interactions in relation to his annotations (Figure 1). The tool is implemented in openGL, and is fully zoomable and pannable and supports selections of interaction elements for detailed inspection. The x-axis of the main view represents time, with a striped background indicating the length of a fixed time duration (defaulted to 60 seconds per strip). The y-axis is divided into 5 sections, with each section supporting one aspect of the participant's investigation process. Figure 1 (A)-(E) show the 5 perspectives, which are the participant's annotations (A), the participant's interactions with the three views in WireVis (B), the depths of a participant's investigation (C), the areas of the participant's investigation (D), and the time range of the investigation (E).

The sliders in (F) allow the user to scale time, while checkboxes in (G) control various visualization parameters. The detail view in (H) depicts detailed information of a specific user-interaction element.

Annotation View: As mentioned in section 4.1.2.3, the results of our participants' think-aloud during the experiment are recorded into separate files. These annotations to the participant's investigation are shown in this view. As can be seen, our participants often exhibit a hierarchical structure in their reasoning process, with the highest level of reasoning depicting strategies they employ such as "seek all keywords related to the keyword *food*" The lower levels depict specific operations to execute those strategies, ranging from "search for keywords other than *food* relating to account 154" to "identify the receiver of a transaction (account 64) of account 154."

The hierarchical nature of the participants' reasoning are represented in the annotation view, with the higher level annotations shown above the lower ones as interactive floating text boxes [85]. The time range of each annotation is drawn as nested boxes using different colors. The user can select any particular annotation, and its corresponding time range is highlighted across all the other views (Figure 1).

WireVis Interaction View: WireVis uses multiple coordinated views to visualize different relationships within the data. In the WireVis Interaction view, we look to display the participant's usage pattern of the WireVis tool. The three rows in this view correspond to the three main interactive views in WireVis: Heatmap, Strings and Beads, and Search by Example. In each view, we can choose two different attributes of the participant's selection. In Figure 1 (B), the two attributes are keywords (shown as red dots) and accounts (shown as green dots).

On first glance, it is easy to see which views in WireVis the participant interacts with over time. On closer inspection, the distribution of the red (keywords) and green dots (account) also reveal high-level patterns in the investigation. Scattered red dots could indicate an exploration of keywords, whereas concentrated green dots (e.g., if the green dots are aligned horizontally) could reveal the participant's interest in a specific account. When both red and green dots appear together and are connected by a line, it denotes that the participant is investigating the relationship between the two (such as a cell in the Heatmap view in WireVis).

Depth View: On top of visualizing a participant's direct interactions with WireVis, it is also important to see some semantic information regarding the participant's investigation process. In this view, we visualize the "depth" of a participant's investigation by displaying the number of visible transactions in WireVis. For example, when the participant is looking at the overview in WireVis, our Depth view will be completely filled, indicating that all transactions are visible. As the participant zooms in to investigate specific keywords or accounts, the Depth view will show a drop in visible transactions (Figure 1 (C)).

The Depth view also indicates when a participant requests detailed information for a specific account or transaction (such as double-clicking on a bead in the Strings and Beads view). These interactions show up as a vertical line, which is easily distinguishable from a participant's operations for zooming in or focusing on a specific area in the data.

Areas View: While the Depth view shows the number of visible transactions in WireVis, it is also relevant to indicate interactions that highlight areas that the user has shown interests. These interactions are commonly used in WireVis through the "mousing-over" operation. As the participant mouses-over keywords, accounts, or transactions in WireVis, the system displays information about the highlighted data without requiring the user to change the zoom level or focus.

Using the mouse-over operation in WireVis is common, and often indicates an exploration process in which the participant is looking for suspicious areas for further investigation. In the Areas view, a high variation in a short amount of time could indicate such an exploration process, while a more leveled section suggests that the participant is investigating specific activities (Figure 1 (D)).

Time Range View: Time is an important aspect in discovering financial fraud, and WireVis provides views to explore the temporal dimension. In the Time Range view, we look to capture the participant's time-based investigation. The y-axis of the Time Range view denotes the dates represented in the data from more recent to least. A fully colored section indicates that the participant's investigation spans the entire time range, whereas a change would denote that the participant has zoomed in to a specific time period (Figure 1) (E).

The dots in the Time Range view indicate selections of transactions of a specific date. In WireVis, this is done by either mousing-over or double-clicking on a bead in the Strings and Beads view. A high concentration of the appearance of these dots often suggests that the participant has found some specific transactions and is looking to find out the details of these transactions.

4.1.3.2 Strategy Analysis Tool

As opposed to operation, strategy is a long term plan of action designed to achieve a particular goal. As shown in the Annotation view of the operation analysis tool (section 4.1.3.1), most of our participants exhibit both strategic and operational reasoning when investigating fraud. So besides addressing the question "what do the participants actually do" using our operation analysis approach, we also look to investigate the high level strategies that the participants employ while approaching the tasks. Through the use of our strategy analysis tool, we can identify each participant's areas of interest as well as comparing different participants' strategies.

We adopt treemap as the basis of our strategy analysis tool. The treemap visualization allows us to investigate similarities between our participants' strategies without considering the flow or speed in which our participants execute their strategies. As mentioned in section 4.1.3), our participants had varied preconceived knowledge about the keywords and their meanings, and therefore approached the investigation tasks differently. Many of them identified the same embedded fraud scenarios, but none of them shared the same path in discovering these activities. Using our modified treemap visualization, we can identify the participants' strategies without regard to the paths they have chosen.

The initial layout of the strategy analysis tool shows three different treemap views classifying the transaction data based on three attributes: time, keywords, and accounts (Figure 2). The three views are coordinated such that highlighting a transaction in one view also highlights the same transaction in the other two views. We choose transactions to represent the lowest level of the treemaps because they represent the lowest granularity of the data. A colored circle is displayed on each cell, and the size depicts the amount of time the participant's investigation has included that transaction. When comparing two participants or two groups of participants, the color of the circle indicates which of the two participants spent more time on the transaction (Figure 5).

Timeline View: Transactions in this view are classified based by their date. Each grouping contains transactions of the same month. As shown in Figure 2, the transactions in our synthetic data set span a 13 month period. Note that the participant depicted in this view did not perform his investigation based on the transaction dates as the circles appear fairly evenly through all 13 months.

Keyword View: This view applies two different classification criteria. On the top level, transactions are grouped based on keywords (shown as yellow cells in Figure 2). Each cell is then further subdivided by individual accounts (shown as green cells).

Since a transaction often contains multiple keywords, the same transaction could appear in more than one keyword cells. Similarly, every transaction contains two accounts, a sender and a receiver, so a transaction will always appear at least twice, once for each account. Due to these two reasons, the total number of transaction cells in this view are greater than those in the Time view. However, we find this layout more intuitive for understanding a user's strategy involving keywords. For example, in Figure 2, it is easy to see that the participant focused on a few specific keywords, but even more specifically on a few accounts relating to those keywords.

Account View: The Account View orders the transactions based on their corresponding sending accounts. As shown in Figure 2, this view makes clear that this participant's strategy in discovering financial fraud using WireVis is almost entirely based on the detailed investigation of one or two accounts. Time and keywords appear to be secondary considerations during his investigation.

4.1.4 Analyzing User Interactions and Reasoning

From the annotations and from processing the audio and videos of the interviews from the user experiment, we know that all participants identified several transactions they felt were suspicious. While the training and the hints we provided influenced participants' early fraud detection strategies, we noticed that the users quickly developed their own strategies to locate fraudulent transactions. In the following sections, we show examples of using our operation and strategy analysis tools to analyze the participants' interactions within WireVis and relate the interactions to the participants' annotations.

4.1.4.1 Operation Analysis Result

Figure 3 shows a participant's interactions. The participant started by examining keywords in the Heatmap (Figure 3 (A)), and identified a specific account for examination as evident by the subsequent appearance of that account in the Strings and Beads section of the WireVis Interaction view (shown as green dots in (Figure 3 (B)). The Areas view (Figure 3 (C)) indicate that while the

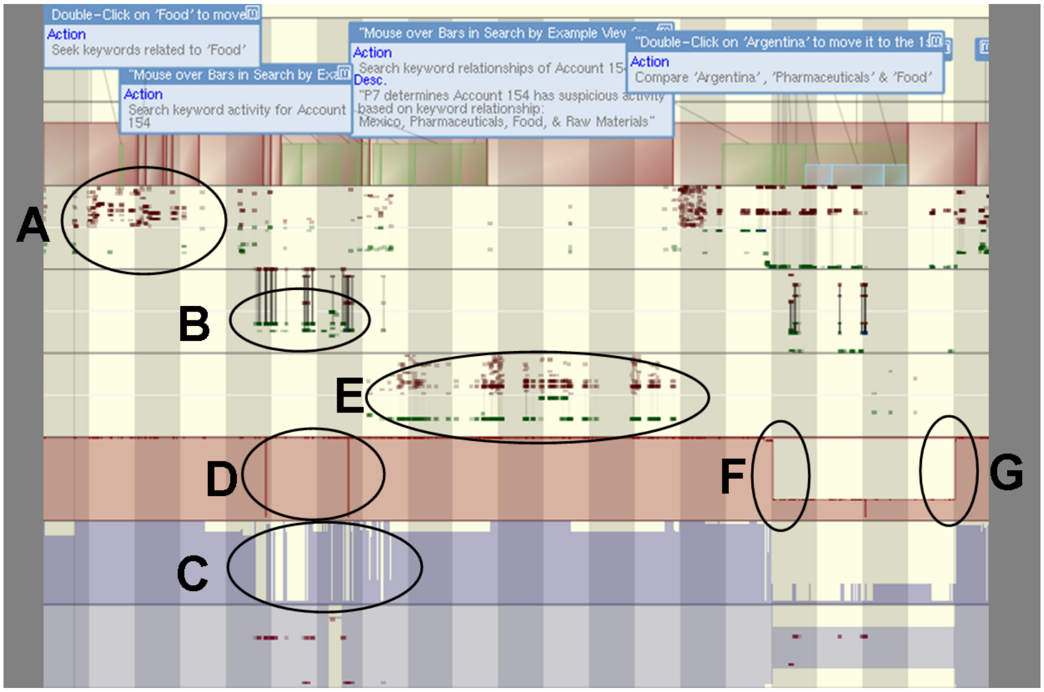


Figure 3: A participant's interactions and annotations are shown in the operation analysis view. (A) shows the participant's activities in searching for keywords in the Heatmap view. (B) shows that the participant utilized the Strings and Beads view for exploration (C), but did not perform extensive detailed inspections on any keyword or account (D). The participant then used the Search by Example view to discover accounts of similar behaviors (E), and drilled down into a specific account for further investigation (F). However, no suspicious activities were found, and the participant zoomed out again (G).

participant explored that account extensively in Strings and Beads, the lack of activities in the Depth view (Figure 3 (D)) show that the participant did not drill down for detailed investigation. This series of interactions suggest that the participant did not identify any suspicious behavior during this time.

The participant's annotations for this period corroborates with our analysis. He indicated that he was identifying a specific account (*account 154*) based on the keyword *food* that he thought was interesting. However, upon further investigation, he determined that the transaction involving *food* from *account 154* is not fraudulent because the amount is too low.

Nonetheless, the participant continued to believe that *account 154* was of interest. As shown in Figure 3 (E), he used Search by Example to identify other accounts that were similar to *account 154*. After he identified one (*account 128*), he used the Heatmap view (Figure 3 (F)) to drill down to examine the details of that account. Unfortunately, upon further investigation, no fraudulent activities were found, and the participant zoomed back out in the Heatmap view (Figure 3 (G)).

This observation is again corroborated by the participant's annotations. The only point of ambiguity in our analysis came at the end of the process when the participant zoomed out in the Heatmap view. We believe that although we had analyzed the participant's activities correctly, it was possible that the participant had indeed identified fraudulent activities while zoomed in on *account 128*. His subsequent zooming out of the Heatmap view could have been to start a different part of the investigation or to execute a new strategy. Without explicit annotations, we currently have no way to differentiate the two scenarios using only the participant's interactions.

This simple example shows that using the operation analysis tool to analyze a participant's behavior indeed corresponds to the participant's annotations describing his reasoning process. Although we found that different strategies could generate similar patterns in the tool, we believe that our tool can still provide a reasonable overview to the participant's intent.

4.1.4.2 Strategy Analysis Results

The strategy analysis tool can be applied either to analyze a single participant's strategy, or to compare the difference between the strategies of two participants. In this section, we demonstrate the use of the tool, and discuss how the participants' strategies can be discovered.

Analyzing a Single Participant: Visualizing a single participant's interactions in the strategy analysis tool reveals patterns of high-level reasoning processes, from which we can extract the participant's strategy without examining each interaction event. Since the patterns are independent of the order of task performed, or the time spent on each task, our tool can be applied to analyze investigation process of varying lengths.

Since we have examined the operational aspect of a particular participant using the operation analysis tool, we go on to identify the high level strategies employed by the same participant using the strategy analysis tool. As shown in Figure 2, the participant focused his investigation based on a specific account over a keyword or time-based approach. By highlighting that account, we can identify that the name of the account is *account 154*. This finding is consistent with the operation analysis result of the participant. While the two tools eventually reveal the same high-level strategy employed by the user, the strategy view reveals this account-centric method easily and intuitively, while using the operation analysis tool required a long process of reasoning and correlating with annotations.

Comparing Two Participants: Since our strategy analysis tool is not bound by the time or sequence of a participant's analysis, we can compare the two easily by displaying them simultaneously in the same view. Figures 4 and 5 show two comparisons of two different pairs of participants.

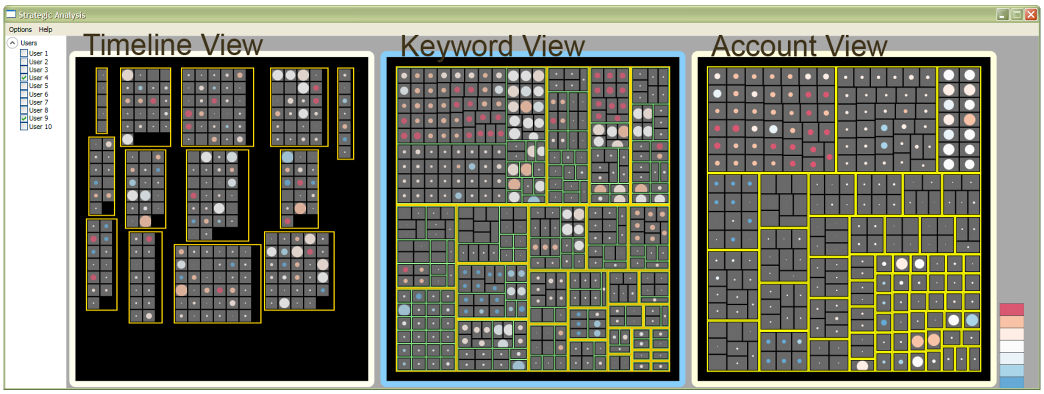


Figure 4: Two participants' interactions are compared using the strategy analysis tool. The red circles indicate a dominant inspection from one participant, and the blue circles indicate the dominance from the other. A white circle denotes that the two participants spent equal amount of time investigating the transaction. The size of each circle represents the total amount of time spent on investigating each transaction. In this case, the two participants exhibited similar strategies as most circles are white.

Since the color of a circle indicates which participant spent more time investigating a transaction, and the size of a circle shows the combined time for both participants, we can identify transactions that are of interest to both participants by identifying large white circles. In Figure 4, we can easily see two participants sharing a similar strategy as the circles are mostly white and large with very few bright red or blue circles. From this figure, we can conclude that both participants were interested in a few specific accounts, and they both spent most of their time investigating them.

On the other hand, Figure 5 tells a story of two participants with different strategies. The differences in colors and sizes of circles indicate that the two participants rarely share the same focus. The participant represented as blue utilized a more keyword-centric approach, spending most of his time looking for keyword-related possibilities; while the other participant focused more on searching for suspicious accounts.

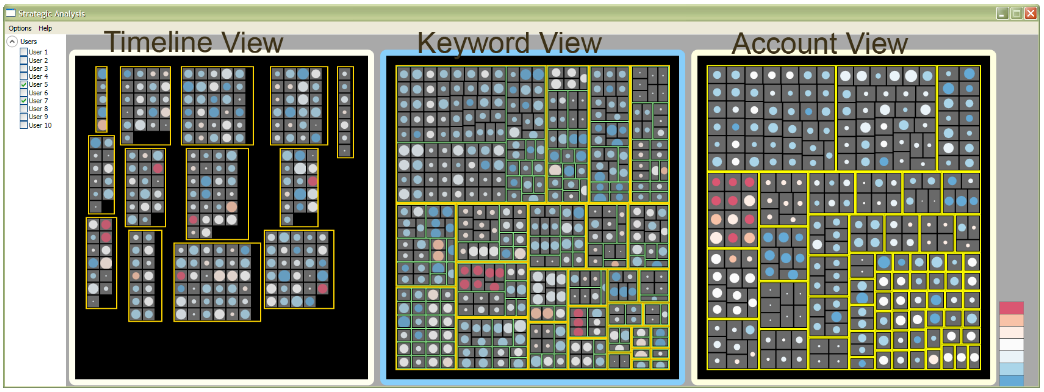


Figure 5: Contrary to Figure 4, this image depicts two participants with different strategies. It can be seen that one participant (red) employed an account-focused approach, while the other (blue) performed the investigation based on keywords.

4.1.4.3 Combining the Two Analysis Tools

We believe that the two tools presented in this section adequately capture both the operational and strategic aspects of a user's interaction and reasoning process. However, it is the combined use of the two tools that truly reveals a participant's intent and approach to investigating fraud. With the use of the operation analysis tool, we can identify localized patterns for specific investigation processes. These patterns can then be visualized in the strategy analysis tool to uncover the high-level strategy behind them, as well as compared against other participants to discover different approaches to similar tasks. Although our two tools are not tightly integrated at present time, we look forward to merging them into one single application in the near future.

4.1.5 Discussion

While the user experiment described in this paper achieved the goal of connecting the user's interaction with his reasoning process, there were some unexpected results that are worthy of further investigation. First of all, most of our participants expressed that they lacked domain expertise in financial fraud detection. However, many of them identified what they believed to be suspicious activities that were not part of the embedded threats scenarios and yet made very convincing arguments on why the activities could have been. These discoveries reaffirm the capabilities of WireVis in discovering suspicious trends and patterns, but they also bring to our attention that our synthetic data set is far from the "ground truth" we hoped it could be, and that the definition of suspicious activities is indeed subjective.

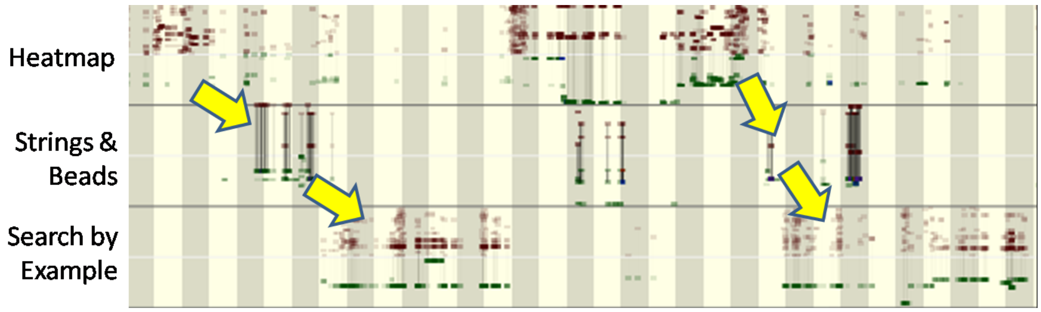


Figure 6: A participant's interactions in the operation analysis tool. Each yellow arrow represents how the sequence of operations while searching for suspicious activities are created.

Furthermore, the ways in which our participants interacted with WireVis were sometimes unexpected. Through the operation analysis, we find that almost all of our participants exhibit the flow of using Heatmap first, followed by Strings and Beads, and eventually Search by Example (Figure 6). This behavior could be a result of the sequence of the views presented during the training sessions (the three views were introduced in that order), or a reflection of the inherent functionality of each view (overview vs. detail presentation). We currently do not have any concrete arguments to support either possibility, but we are looking into better understanding this behavior.

We are still a long way away from being able to fully understand the relationship between a user's interactions and his reasoning process for all visualizations, but we believe that we have taken a small step in the right direction. Our confidence in our approach in monitoring and analyzing a user's behavior is supported by financial fraud analysts at Bank of America, who, at this point, have little or no way to review or understand other analysts' reasoning process in investigating fraud. By using our tool, high-level semantic user events in WireVis can be captured such that each analyst's actions can be replayed without the storage of massive amount of video (screen-capture) data. When these events are reviewed using our operation and strategy analysis tools, the analysts' interaction patterns and reasoning processes at different levels are revealed. Together, we believe our methodology and tools provide the foundation for transforming analysis processes into visualizations that are sharable, repeatable, and accountable.

4.2 WireVis Study

In this section, we extend the work described in previous section to understand the analysis strategies of real financial compliance analysts using WireVis. As mentioned previously, it is our hypothesis that when interacting with a well-designed visual analytical tool, a large amount of an analyst's reasoning process is embedded within his interactions with the tool itself. Therefore, through careful examination of the analyst's interaction logs, we propose that we should be able to retrieve a great deal of the analyst's reasoning process. To validate our hypothesis, we designed a study to quantitatively measure if an analyst's strategies, methods, and findings can be recovered through human examination of his interaction logs. Our study consists of four stages: user observation, transcribing, coding, and grading. In the user observation stage, we invited 10 financial analysts to use WireVis to identify potentially fraudulent wire transactions within a synthetic dataset in think-aloud sessions. The analysts' interactions were logged into file and at the same time their think-alouds captured on video and audio. These information were transcribed by the authors later into files that collectively were considered to be representative of the analysts' reasoning processes and used as the "ground truth" for the study.

Four coders who are students familiar with the WireVis tool examined each analyst's interaction log using the two log analysis tools (Operation and Strategic Analysis tools) that were described in the previous section. Through visual inspection and analysis of each analyst's interaction log, the four coders were asked to annotate what they believed the analysts' strategies, methods, and findings were. We then compared the coders' inferences with the ground truth, and the result became the basis of our claim on the types and amount of an analyst's reasoning process that were recoverable through the examination of interaction logs.

The result of our study has been most encouraging. Aside from a few specific, low-level types of findings, the four coders (who are not trained in financial fraud detection) were able to correctly retrieve 60% of the analysts' strategies, 60% of the methods, and 79% of the findings. This result indicates that some of an analyst's strategies, methods, and findings in using a visual analytical tool are indeed recoverable through human examination of interaction log. It is relevant to note that the extracted reasoning process is solely based on the analyst's activities within a visual analytical tool and does not include the overall intelligence analysis that often involves multiple tasks and tools such as searching through websites, phone discussions, the use of additional software, etc. However, our findings represent an important aspect of the intelligence analysis, and provide an example for visual analytics as a community to uncover a new path towards better understanding and capturing of an analyst's reasoning processes.

4.2.1 Evaluation

We conducted a user study to determine how much of an analyst's reasoning process can be recovered using just the captured user interactions. We evaluated this recovery in a quantitative fashion by comparing the process that was inferred by a set of coders against the ground truth determined from videos of the exploration process.

Four stages are designed as user observation, transcribing, coding, and grading. The comprehensive information of each stage is provided in the following subsections.

4.2.1.1 User Observation

In order to understand the user's reasoning process through his interactions, we first conducted a qualitative, observational study of users analyzing data with WireVis. We recruited 10 financial analysts with an average of 9.9 years (and a median of 8 years) of financial analysis experience who all worked in large financial firms in our area. All of the participants were either currently working as a financial analyst or had professional financial analyst experience. Eight of the users were professionally trained to analyze data for the purpose of fraud detection. Of the 10 analysts, six analysts were male and four were female.

To preserve the privacy of Bank of America and their individual account holders, we utilized the synthetic dataset described in the previous section (section 4.1.2) for the purpose of this study. Although none of the transactions in the dataset are real, we captured as many characteristics and statistics from real financial transactions as we could and modeled the synthetic data as closely to the real one as possible. The dataset was designed to be simple enough that users were able to look for suspicious transactions within the time frame of a study, but was complex enough that interesting and complicated patterns could be found. This dataset contained 300 financial transactions, with 29 keywords. Some keywords were the names of countries, such as Mexico, and others were goods or services, such as Software or Raw Minerals. We also developed four threat scenarios and injected a total of nine cases we deemed suspicious into the dataset. The threat scenarios included transactions in which keywords should not appear together, accounts with dual roles, keywords with unusually high transaction amounts, and accounts with suspicious transactional patterns appearing over time.

At the beginning of the study session, each participant was asked to fill out a demographic form and was then trained on the use of WireVis for approximately 10 minutes. The participant was also provided a one-page overview of the functionality of WireVis and encouraged to ask questions. Following the training, the user was asked to spend 20 minutes using WireVis to look through the dataset to find suspicious activities. We asked the participant to think-aloud to reveal his strategies. We specifically encouraged the participant to describe the steps he was taking, as well as the information used to locate the suspicious activities. Once the user drilled down to a specific transaction, he was asked to write it down on a Discovery Sheet for the purpose of recording and reporting his findings. Once the user documented a specific transaction, he was encouraged to continue looking for others until the time limit was reached. After the exploration process, a post-session interview was conducted for the participant to describe his strategies and additional findings.

Several methods were used to capture each participant's session as thoroughly as possible. Commercial usability software was used to capture the screen. A separate microphone was used to record the user's audio during the session. Lastly, functions built into the WireVis system captured the user's interaction with the tool itself as information relevant only to the WireVis system. Instead of recording every mouse movement or keystroke, WireVis captures events that generate a visual change in the system. For example, a mouse movement that results in highlighting a keyword in the Heatmap view will generate a time-stamped event noting that the user has highlighted a specific keyword.

4.2.1.2 Transcribing

The video and the think-aloud of each participant were used to create a detailed textual timeline of what each participant did during their session, along with the participant's self-reported reasoning and thinking process. While the created textual timeline is an interpretation and might not perfectly reflect the (internal) reasoning process of the participant, it was created based on the facts recovered from video and audio with conscious efforts in minimizing human bias. We therefore consider the resulting transcript to represent the "ground truth" of what each participant did during their analysis with WireVis.

During the transcribing stage, different strategies, methods, and findings in investigating fraudulent activities were identified to serve the grading process later. Specifically, we identified the following in the transcript:

- A *"Finding"* represents a decision that an analyst made after a discovery.
- *"Strategy"* is used to describe the means that the analyst employed in order to arrive at the finding.
- Also, the link between "finding" and "strategy" is captured by *"method"* which focuses on what steps the analyst adopted to implement the strategy for discovering the finding.

In a typical investigation, an analyst's *strategy* might be to search for a specific suspicious keyword combination based on his domain knowledge. For example, the analyst might determine accounts and transactions involving both the keywords Mexico and Pharmaceutical to be potentially suspicious. Using this strategy, the *methods* employed by this analyst could then be comprised of a series of actions such as highlighting or filtering those keywords, and drilling down to specific accounts and transactions. At the end of the investigation, the analyst would record his *findings* based on the encountered account numbers and transaction IDs along with their decision about whether the particular finding is suspicious or not.

4.2.1.3 Coding of Interaction Logs Through Visual Examination

We asked several people familiar with WireVis to view each participants' interactions and determine their reasoning. Specifically, we recruited four "coders" from our university, all of whom were familiar with WireVis (three male, one female). They then used the two interaction log analysis tools (Operation and Strategic Analysis tools) to view participant interactions, and created an outline of what occurred.

We first gave all coders comprehensive training on how to use the Operation Analysis Tool and Strategic Analysis Tool to examine the interaction logs of each analyst's investigations. We also provided a guideline of hierarchical coding procedures, asking coders to, in free-text format, provide hierarchical annotations within the visual analytical tools. The hierarchies are reflected as different levels of decision points and strategies extracted by the coders. We asked coders to identify and label findings, strategies, and methods for each analyst. In addition, coders were encouraged to annotate on the transitions if they could discover relationships between each decision point, such as when one strategy leads to multiple findings or one finding transforms to a new strategy.

All findings from the coders were recorded as annotations and linked to corresponding interaction events and time range. Each coder went through the 10 analysts' interaction logs one by one using the visual analytical tools, spending an average of 13.15 minutes reconstructing each analyst's reasoning process. Thus, at the end of the coding phase, we collected 10 sets of annotations from each coder, resulting in 40 sets of annotations overall.

4.2.1.4 Grading

We then compared the annotations the coders produced to the "ground truth" to determine how much of the reasoning process was able to be reconstructed by the coders. The comparisons are graded according to a set of pre-determined criteria by one of the authors, which we describe below.

The categories we used in the grading were in accordance with both transcribing and coding: finding, strategy and method. Generally speaking, "strategy" and "finding" do not necessarily have a one-to-one mapping relationship since some strategies may lead to multiple or null findings. But one "finding" always comes with a "method" in the sense that a method is always needed to make a decision.

For each finding, strategy, and method, we graded according to the following criteria: "Correctly Identified", "Incorrectly Identified", "False Detections" and "Never Identified". This combination was chosen because the four measurements covered all possible scenarios and yet were explicitly distinguishable. "Incorrectly Identified" indicated that a coder noticed some meaningful interactions but incorrectly interpreted them, while "False Detections" captured the scenarios in which a coder thought that certain action took place but in fact there was none. "Never Identified" involved actions that took place, but were not noticed or annotated by the coders.

		Ground Truth		C1	C2	C3	C4
P1	Finding	6	Correctly Identified	5	3	4	5
			Incorrectly Identified	0	1	0	1
			False Detections	0	0	2	0
			Never Identified	1	2	2	0
	Strategy	3	Correctly Identified	3	3	1	0
			Incorrectly Identified	0	0	1	3
			False Detections	0	0	1	1
			Never Identified	0	0	1	0
	Method	6	Correctly Identified	6	4	3	3
			Incorrectly Identified	0	0	1	2
			False Detections	0	0	0	0
			Never Identified	0	2	2	1
	Time Spent (min)			14.7	33.39	10.27	35.9

Figure 7: Grading results of participant 1. A participant's analysis process is separated into findings, strategies, and methods. This figure shows the results of four coders' annotations and how they match the participant's analysis according to the four grading criteria: correctly identified, incorrectly identified, false detections, and never identified.

Figure 7 illustrates the overall criteria used for grading. We determined that a "finding" was correct as long as the coders correctly identified there was a decision made during the analyst's investigation. But we did not ask them to determine what the outcome of that decision was (whether the certain transaction is suspicious, not suspicious or inconclusive). Additionally, if only a part of the coder's annotation was correct, for example if he determined a "strategy" was looking for five incompatible keywords but only identified four keywords correctly, we graded that annotation as

"Incorrectly Identified". This purpose for such a strict grading criteria is to minimize potential bias in the grading process.

4.2.2 Results

Both the quantitative and the observational results we obtained from grading are rich and informative. In this section, we first demonstrate quantitatively the amount of reasoning that can be extracted from analyzing interaction logs. We then describe some of the trends and limitations of the coding process using our interaction log analysis tools.

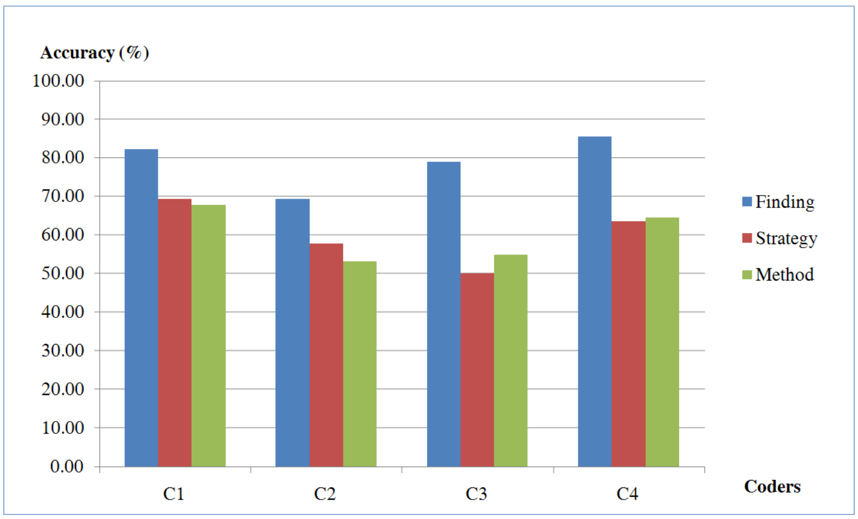


Figure 8: The average accuracy of the four coders correctly identifying "findings", "strategies" and "methods" of all ten participants.

4.2.2.1 How Much Reasoning Can We Infer?

Figure 8 shows the average accuracy of each coder's reconstructed reasoning processes of all participants. The results are separated into three categories as described in section 4.2.1.2: findings, strategies and methods. The results indicate that it is indeed possible to infer reasoning from user interaction logs. In fact, on average, 79% of the findings made during the original investigation process could be recovered by analyzing the captured user interactions. Similarly, 60% of the methods and 60% of the strategies could be extracted as well with reasonable deviation between the coders.

An interesting observation is that all coders performed better in extracting findings than strategies or methods. We will discuss a possible explanation for this phenomenon in section 4.2.3.

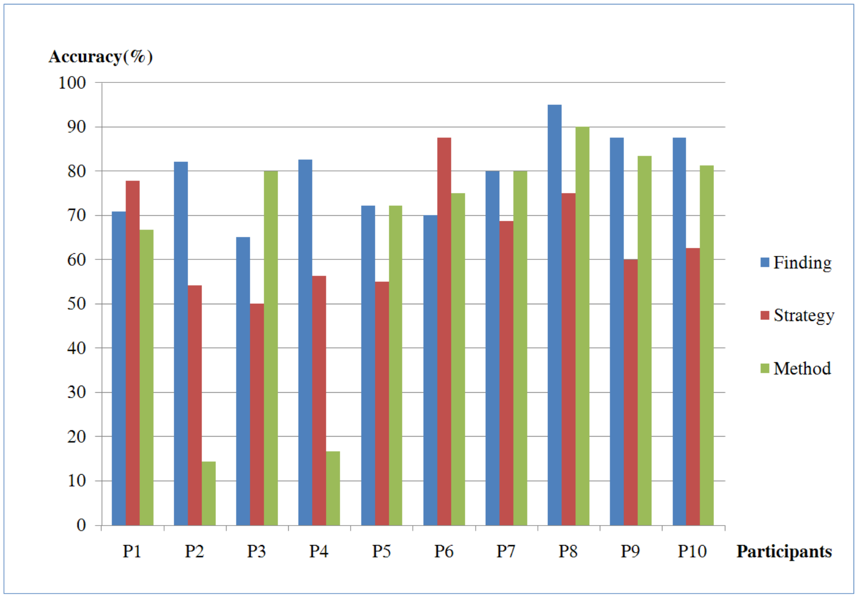


Figure 9: The average accuracy of correctly identifying “findings”, “strategies”, and “methods” based on the 10 participants.

Across participants:A different perspective from which to examine the results is to look for variations in accuracy across the 10 participants. Figure 9 shows the average accuracy of the coders in recovering the reasoning processes behind the 10 participants. This result indicates that there is a noticeable difference between accuracies in extracting reasoning processes for different participants. This finding leads to the conclusion that there are some analysis processes that are more difficult to follow than others. Although there is no definitive answer to why this is, our own investigation suggests that there are two plausible contributors. The first is the difference in experience in financial fraud detection between our participants and our coders. Since our coders have no training in fraud detection, it is natural that some of the strategies and methods in investigative processes are lost to them.

Another cause of this variation is manifested in the acute drop in the accuracy when extracting “methods” from P2 and P4’s analysis as shown in Figure 9. As the figure suggests, the coders were baffled by the methods of these two participants. Upon investigation in the video of the participant’s analysis process, we discovered that participants 2 and 4 focused their analysis on the irregularities in the time-series view in WireVis. Specifically, they closely examined “spikes” in the view (Figure 10)

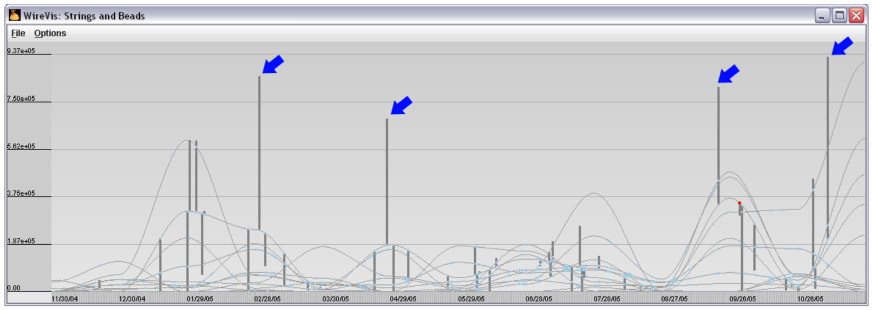


Figure 10: The time-series view in WireVis showing spikes that indicate sudden increases in the amounts or frequencies of wire transactions.

which indicate sudden increases in amounts or frequencies of wire transactions. Our coders had no way of seeing these visual patterns, so they were not able to identify the methods behind the participants' analyses.

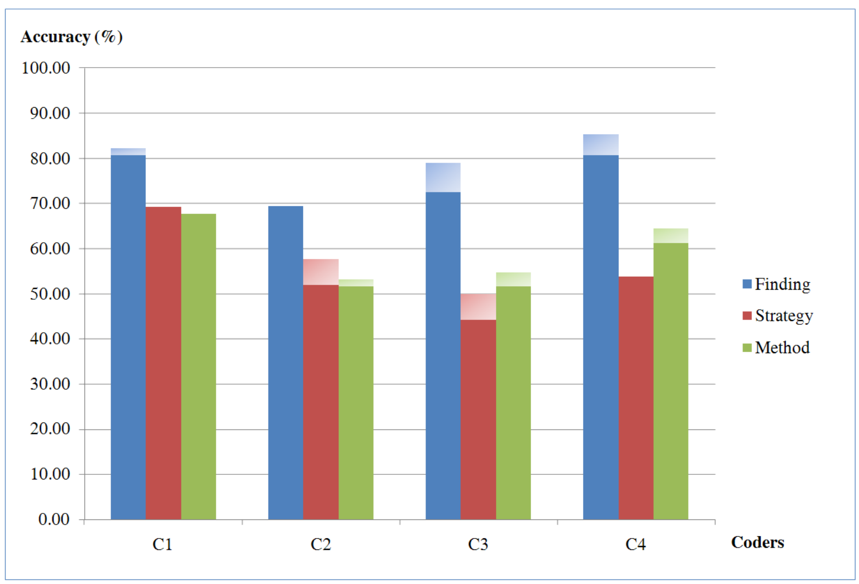


Figure 11: The accuracy of the coder's annotations in matching up to the 'findings", "strategies", and "methods" of the analyses. The semi-transparent areas indicate the decrease in accuracy compared to Figure 8. The difference between the two figures is that Figure 8 indicates the amount of reasoning that can be recovered, where as this figure shows how accurate the coders' annotations are.

Considering false detections: Since the purpose of this study is to figure out *how much* of the reasoning process can be extracted from interaction logs, we have reported the accuracy based purely on the number of "correctly identified" elements. However, it is relevant to make note of the number of times that our coders made detections that turn out to be inaccurate. Under our grading scheme, the number of annotations made by a coder often exceeds the number of elements in the transcription due to the false detections. For example, the grading result of participant 1 in Figure 7 shows that the number of "findings" in the ground truth is 6, however, coder 3 made a total of 8 annotations. He correctly identified 4 of the 6 elements, missed on identifying 2 of the 6 elements, and falsely detected 2 times when there were no corresponding elements in the ground truth.

With the "false detections" in mind, we re-examine the accuracy of the coders based not on how much of the reasoning process can be recovered, but on the accuracy of their annotations. Figure 11 shows the result of the coders' accuracies that include the coders' false detections. Not surprisingly, the accuracy of the coders all decrease slightly. The accuracy in extracting findings drop by 3% from 79% to 76%, strategies by 5% from 60% to 55%, and finally methods by 2% from 60% to 58%.

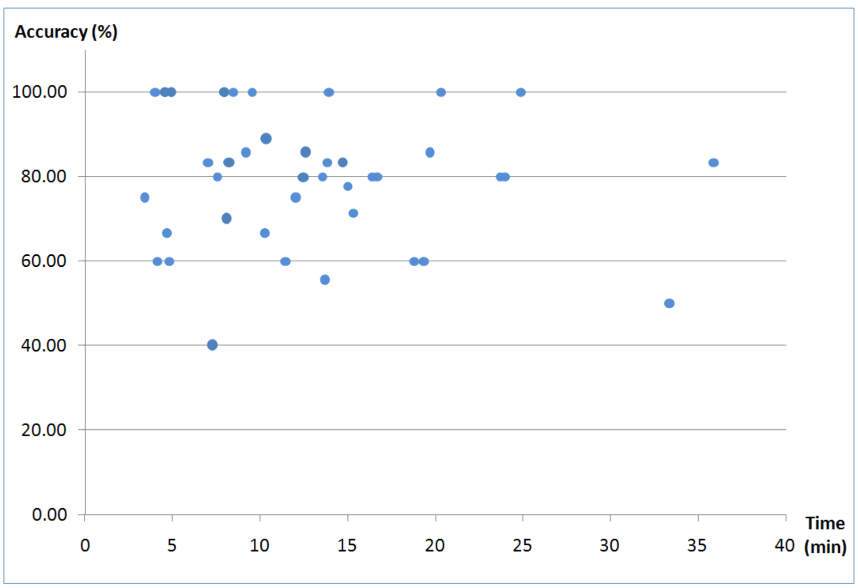


Figure 12: The accuracy of the coders in recovering "findings" of the participants and the amount of time spent.

4.2.2.2 Amount of Time Spent by Coders

One important aspect in extracting reasoning process is the amount of time necessary for analyzing the interaction logs. In this section, we discuss the effect of time spent by a coder in analyzing an individual interaction log, as well as the learning effect that the coders exhibit after gaining proficiency in extracting the participants' reasoning processes.

Capturing time spent by a coder: Built into our Operation and Strategy Analysis tools is the ability to track the amount of time that a coder spends using the tools. The coders were made aware of this feature and were told not to take breaks during an analysis. Since the coders directly annotated their discoveries into the Operation Analysis tool, the system was able to record the amount of time spent by each coder when analyzing an interaction log.

Furthermore, the system tracked when the coder started and stopped the annotations. The purpose of this feature was to separate the time spent in analyzing the interaction log from the time spent in annotating. On average, the coders spend 23.9 minutes analyzing one interaction log, of which 10.75 minutes were spent on annotation and the remaining 13.15 minutes on investigation.

Time spent vs accuracy: We examine the relationship between the time spent by a coder and accuracy. Overall, there is no correlation between the two. Figure 12 plots the relationship between the coders' time spent in analysis (not including time spent for annotation) and their accuracies in extracting "findings". With the exception of the two outliers in the far right, it appears that the coders are consistently successful when spending anywhere from 5 to 15 minutes. This suggests that spending more time in the analysis does not always yield better results. The two outliers represent the analysis of coders 2 and 4 in their first investigation (participant 1). As we will show in the following section, all coders become more proficient in their analysis as they gain experience.

Increase in accuracy: As shown in Figure 9, the accuracy of the coders increase as they gain experience in investigating interaction logs as all four coders began with examining participant 1's interactions and end with participant 10's. Based on analyses using Pearson's correlation coefficient, we find that the number of participants a coder has examined is positively correlated to the coder's accuracy. This correlation is statistically significant when extracting "findings" ($r(40) = .37, p < .05$) and "methods" ($r(40) = .52, p < .01$). Only in extracting "strategies" is the correlation weaker ($r(40) = .21, p = .182$). While the sample size is relatively small, these statistics nonetheless imply a subtle but potentially important discovery: with more experience in analyzing interaction logs, a coder could become more proficient in extracting an analyst's reasoning process.

4.2.3 Discussion

The study described in this paper is complex and intricate. On top of involving real financial analysts, the transcription process, the coding, and the grading were all performed with great care and consideration. Although many of the nuances encountered during the study do not affect the results and therefore have not been described in this paper, there are some findings that might be of interest to the community. First of all, during our informal debriefing of the coders, the coders discussed the strategies that they employed in analyzing the analysts' interaction logs. It turned out that our coders often began their investigation by looking for "gaps" in the timeline of the operational view (Figure 1), which are the byproducts of the analysts taking time to write down their findings in the Discovery Sheet (section 4.2.1). Based on the gaps, the coders looked for the analysts' findings, and then worked backwards to discover the strategies and methods used to derive the findings.

While this strategy may seem specific to this study and non-generalizable, we argue that in a real life scenario, analysts either directly make annotations in the visualization to make note of a finding, or they write down their finding on a piece of paper for future reference. Either way, there will exist a visible marker that suggests a relevant discovery by the analyst. Therefore, while we did not anticipate this strategy by the coders, we find their quick adoption of this method to identify the analysts' findings to be effective and relevant.

A second interesting trend pointed out by our coders concerns the usefulness of our visual tools for depicting the operational and strategic aspects of the analysis (section 4.1.3). According to the coders during the debriefing, all of them used the Operational Analysis tool first to gain an understanding of the overall impression of an analyst's interactions. However, the Strategic Analysis tool is often utilized to examine a specific sequence of interactions when the interactions appear random and jumbled. By presenting the results of the interactions from three perspectives (accounts, keywords, and time) in the Strategic tool, the coder could often identify the focus and intent behind the series of interactions. This finding not only validates our design of the tools, but also reconfirms the importance of visualizing both the strategic and operational aspects of an analysis process. In fact, most of the coders began their investigation by identifying the "findings" through looking for gaps in the interactions, followed by looking for "strategies" through examining the overall visual patterns in both the Strategic and Operational Analysis tools without focusing on individual user interactions. Finally, "methods" were extracted through the use of the Operational Analysis tool where specific interactions were examined in detail.

One last relevant aspect of our study is the measurement of "incorrectly identified" elements in the grading process. In all of our results shown in section 4.2.2, we do not take into account elements that have been graded as "incorrectly identified." As mentioned in section 4.2.1.4, any annotation by a coder that does not perfectly match the transcription is considered to be incorrectly identified. This includes scenarios in which a coder identifies the analyst's strategy to be examining 4 keywords when in fact the analyst was examining 5, or when a coder determines that the finding of the analyst is a transaction between accounts A and B instead of accounts A and C. If we were to give half a point to these incorrectly identified elements, the overall accuracy of extracting strategies increases drastically from 60% to 71%, methods from 60% to 73%, and findings from 79% to 82%.

4.3 Model of Visualization for Interaction Capturing

In the previous two sections, we describe how financial fraud analysts' reasoning processes in using a visualization could be captured and later extracted through the use of interaction log visualizations. Although the findings of the study is significant in that we demonstrate that 60-80% of the analysts' strategies, methods, and findings can be gleaned from analyzing interaction logs, the study does not shed light on why the study is successful or how to design future visualizations to perform the same capturing and recovery mechanisms. In searching for such explanations and design principles, we find that the visualization community has become increasingly aware of the concept that the "process" is often just as important as the "product" [73, 158]. Specifically, numerous systems and applications have been published in recent years that aim to capture a user's interaction history (or sometimes referred to provenance) [156, 171, 66, 68]. However, while these systems have all reported varying degrees of success, there still does not exist a set of fundamental or generalizable principles that could explain why these systems are successful, or how others could learn from these successes and apply the techniques to their own domain. To the best of our knowledge, while there are a number of success stories, there exist many other unpublished systems that have not been successful at capturing a user's analysis process using a visualization. The questions we seek to answer in this section are: how can one capture a user's analysis process? And why are some visualization systems more successful at capturing than others?

To answer the first question, we turn to van Wijk's operation of visualization model [175] and examine how a user interacts with a visualization. Based on the model, we propose that there are in fact two separate modes of capturing: internal and external capturing to the visualization. Internal capturing within the visualization includes methods such as screen capturing and interaction logging; whereas capturing external to the visualization includes the use of eye trackers, video camcorders, or advanced machinery such as EEG (Electroencephalography) and fMRI (functional Magnetic Resonance Imaging). These two modes together represent all possible methods of capturing that are available today, but choosing the appropriate methods will depend on the goal and context in which the visualization is used.

In order to answer the second question of why some visualizations are more successful at capturing than others, we further investigate the interaction logging component of the model. By examining systems in the visualization and HCI communities from an analysis capturing perspective, we derived a generalizable principle to evaluate the effectiveness of a visualization system in logging user's interactions. Specifically, we propose that how easy it is to capture a user's analysis process within a visualization can be expressed using three criteria: the visualization's interactivity, the semantics encoded in the captured user interactions, and information change caused by the interactions. Collectively, these three criteria represent the degree of ambiguity in relating a user's interactions to the analysis process. Using these three criteria, we posit that visualizations with high interactivity, semantically rich interactions and low information change during interaction would tend to be more successful at capturing a user's analysis process.

To validate our principle using the three criteria, we examine some existing visualization systems and postulate how much a user's analysis process could be captured through interaction logging.

For systems that cannot easily rely on simple interaction logging, we propose alternative capturing methods based on the van Wijk model and discuss the potential cost. While our validation does not involve actual modifications of these systems, we nonetheless hope that it can shed light into how others might be able to utilize our proposed model and principles and apply them to specific systems and domains.

Finally, we note that the principle proposed in this paper differs from general visualization design principles in that the goal of the former is to capture the user's analysis process while the goal of the latter is to present information visually and efficiently. Although the two do not directly contradict each other, additional care and consideration might be necessary in order to incorporate the two. In the discussion section, we examine the relationship between the two and consider the impact of their differences in designing visualization tools.

4.3.1 Related Work

As noted by Kindlmann [100] and Silva et al. [158], the lack of reproducibility of visualization research has the potential of hindering the advancement of visualization as a science. They argue that in order to recreate and extend specific visualization results, knowing the complete process of how the results are generated is just as important as the techniques used and the final outcome. This process of recording how a user interacts with a visualization is sometimes referred to as *provenance* tracking, which is defined by Anderson et al. [9] as "the logging of information about how data came into being and how it was processed."

Several visualization systems have taken provenance into consideration. However, there exist different interpretations of what constitutes a user's provenance. We roughly categorize these interpretations into two groups: data provenance and information provenance. Even though most visualization systems that capture provenance are not limited to tracking only one specific type, this categorization gives an overview of the on-going research.

4.3.1.1 Data Provenance

Data provenance refers to the logging of low-level interactions. It is the most prevalent type of provenance tracking, and is closely related to the undo/redo functions in nearly all applications today. The primary goal of these systems are often to capture and archive a user's interactions for the purpose of replaying the user's session at a later time. In interactive visualization, one of the most notable systems in data provenance is the GlassBox system by Greitzer [71, 34] which records low-level user's interactions in an analysis environment (such as copy, paste, mouse clicks, window activation, etc). In scientific visualization, VisTrails is an open-source provenance management system that provides infrastructure for data exploration and visualization through workflows [13]. The stored provenance allows the users to query, interact, and understand another session histories with the visualization tool. Jankun-Kelly et al. further generalized how to capture the visualization states as a set of parameters and actions into recordable form [91] that they referred to as the P-Set Model. This model is complete in that every user interaction with a visualization can be described within it.

4.3.1.2 Information Provenance

Groth and Streefkerk [72, 73] recently coined the term “information provenance” to distinguish systems that capture low-level user interactions from systems that record the information discovery process in using a visualization. In their model, they focus on recording the user's interaction independently from the data in a way that the same set of logged interactions can be applied to a different dataset. Additionally, a user can attach annotations to the user's interactions to further add semantic information. Under Groth and Streefkerk's definition, many recent visualization systems that record user interactions incorporate tracking of information provenance. In GeoTime, a user's annotations retain semantic connections to their corresponding events as well as the patterns displayed in a 3D representation [44]. Heer et al. presented methods for both capturing semantic interactions within an information visualization system as well as the mechanism for reviewing, editing, and annotating on those interactions [78]. Similarly, in the Aruvi system developed by Shrinivasan et al., the user's interactions are automatically stored into a visible history tree [156]. The user can also manually construct the state of the discovery using an interactive node-link diagram, which provides additional detail behind the user's interactions. Lastly, Gotz and Zhou incorporated automatic tracking of information provenance in their system HARVAST (which they referred to as insight provenance) and suggested a taxonomy of various types of action-tiered user interactions [68].

4.3.1.3 Utilizing the Captured Interactions

Aside from reviewing a user's interaction history, there has been little research in how either the captured interactions could be used. While all of the aforementioned systems have noted on the benefits of capturing user interactions, including communication, evaluation, training, etc., the details of how the results of the capturing could be utilized to achieve such benefits is sometimes unclear. A notable exception is the Active Reports project at Pacific Northwest National Laboratory [130, 129] where the focus is not on capturing a user's interactions, but on integrating the results into a live report such that sections of the reports are linked to the user's reasoning and analysis products generated by visualization systems.

The motivation of this research is based on the finding in the previous section (section 4.2) in that we assume that data and information provenance do not directly translate to a user's analysis process. Depending on the methods in capturing interactions and what is actually recorded, the amount of a user's analysis that can be reconstructed will vary. In this paper, we look to identify what constitutes a "good" method for capturing interactions and propose criteria for designing visualizations that would result in information provenance that will likely to be more closely related to the user's analysis process.

4.3.2 A Framework for Analysis Process Capturing

How much of an analysis process using a visualization can be captured? Clearly there is a theoretical upper bound of 100%, but intuitively that upper bound is not actually obtainable in practice. So the real questions are: how close can we get to that upper bound? And what do we have to do to get there?

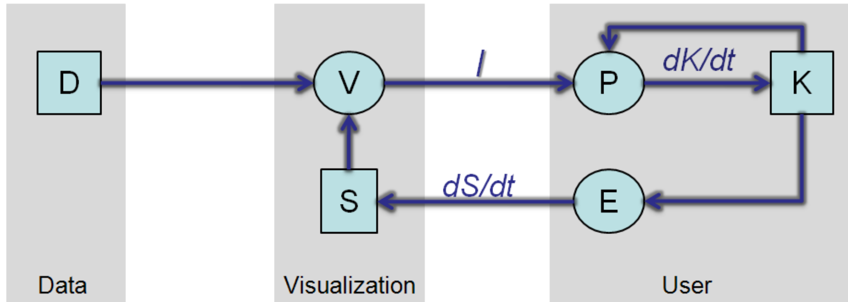


Figure 13: A model of visualization proposed by van Wijk.

To answer these questions, we turn to van Wijk's operational model of visualization to first understand how a user interacts with a visualization. The van Wijk operational model (Figure 13), although simple, distinctively depicts the flow and relationship between the user and the visualization. Specifically, there are two connections, I and dS/dt, between the user and the visualization. I stands for the images generated by the visualization that are perceived by the user, and the connection dS/dt represents the changes in the parameters of the visualization initiated by the user (through the use of a mouse, keyboard, or other input devices) that are applied to the visualization to generate the next sets of images I. Both of these connections can be captured directly within the visualization during user's analysis process by performing screen captures and interaction logging respectively. We refer to these two methods collectively as "internal capturing" (Figure 14(A)).

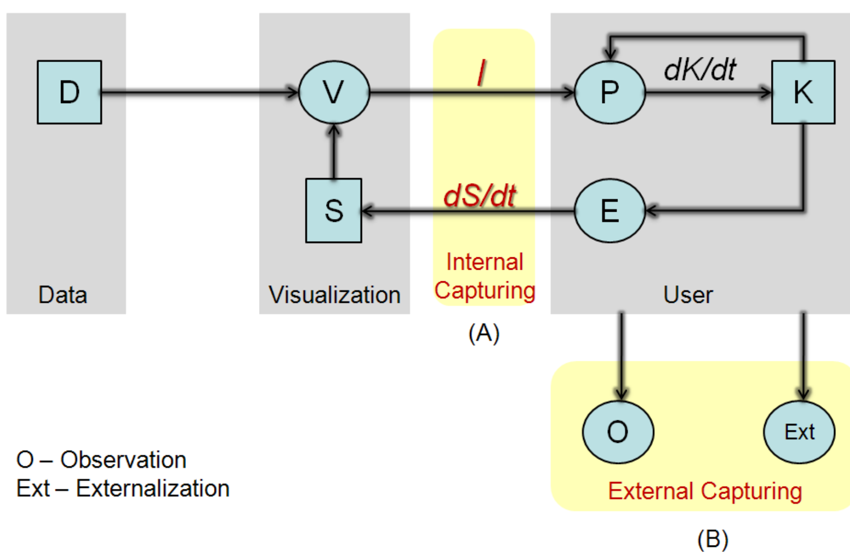


Figure 14: A model for capturing user's analysis process based on van Wijk's model of visualization. The yellow boxes (A) and (B) represent internal and external capturing methods respectively.

In real life, however, solving a complex task is not restricted to only using a visualization. The user could jot down discoveries on a piece of paper, or watch the news on the web to gather up-to-date information [108]. In order to fully capture a user's analysis process in solving a task, the user's activities outside of the visualization need to be captured and collected as well. We further categorize the capturing of these activities into two groups: externalization and observation. In externalization, the results that are explicitly externalized from the user of the reasoning process are collected and stored. These include the notes taken by the user during an investigation, or dictations taken using a voice recorder. In observation, information around the user is captured through the use of additional hardware and machinery. For example, eye trackers can track the user's focus, and a video camcorder can record the user's activities in an environment. In addition, advanced technologies such as EEG and fMRI can be used to monitor the user's neural activities. Together, externalization and observation are referred to as "external capturing" (Figure 14(B)).

We propose that these four capturing methods (capturing I, dS/dt, externalization, and observation) represent a complete theoretical categorization of all capturing mechanisms in visualization. In practice, however, the results and effectiveness will depend heavily on the implementation and accuracy of the methods as well as how the interactions gathered from the different methods are integrated into a cohesive story.

4.3.2.1 Internal Capturing

As shown in van Wijk's operational model of visualization, the relationship between a visualization and its user can be succinctly summarized with two variables, I and dS/dt. These two variables are the input and output of a user's process in using a visualization, and in many cases can be thought to be directly related. As the model demonstrates, I, or an image, is generated by a visualization given a visualization state S. In using a visualization, a user's interactions can be thought of as the means to modifying the visualization state (dS/dt) to create the images that would lead to the user solving a specific problem. Therefore, it is not difficult to see in this model that by capturing the three variables, I, S, and dS/dt, the system can later faithfully reconstruct a user's session in using a visualization.

In practice, however, I and S can be thought to contain the same information depending on whether or not replaying the user's session involves running the visualization. In fact, Jankun-Kelly et al. have proposed a formal model for capturing S accurately [91], and Bavoli et al. have shown that the captured visualization states S can be used to generated large numbers of I efficiently [13]. The main advantage of storing a series of visualization states S over storing a series of images I is that storing the visualization states often requires less disk space than images. However, in cases where the visualization itself could take considerable amount of time to generate I, storing I could still be more efficient.

Capturing and storing the user's interactions as dS/dt, on the other hand, can be very different depending on the purpose of the capturing. If the purpose of capturing is to faithfully reconstruct a user's session with a visualization, dS/dt can be captured as "events" generated by most operating systems (e.g., MouseClick event or a Keystroke event). Typically, visualizations that gather a user's interactions would record dS/dt in this fashion (see section 4.3.1.1). The advantage of storing a series of dS/dt is to further reduce the disk space requirement over storing individual visualization states S since consecutive visualization states often contain duplicate and redundant information.

However, if the purpose of capturing is to reconstruct a user's analysis processes, storing dS/dt as low-level events is inadequate. As Hilbert and Redmiles noted, such events do not carry enough information on their own to allow their significance to be properly interpreted [83]. Most visualization systems that seek to capture a user's analysis process (see section 4.3.1.2) therefore capture the user's interactions at a higher level that include additional contextual information. In some systems, the additional contextual information are semantically related to the specific data or application [41, 156, 141]; whereas some projects categorize the user interactions according to structures that are relevant to the domain or task [68, 103, 78]. In either case, it is clear that storing only low-level user interaction is not enough for gathering semantically-rich information, and subsequently not sufficient for reconstructing a user's analysis process.

4.3.2.2 External Capturing

As mentioned before, in most real-life analysis tasks conducted using visualization systems, not all of the analytical activities actually take place within a visualization. Since these activities are not directly part of the interaction between a user and a visualization system, van Wijk's model is no longer sufficient in describing these activities in relation to the use of a visualization. However, in order to fully understand an analyst's analysis process, it is still very important to consider these activities as they have immediate effects on how a user utilizes the visualization.

We propose that these activities that are external to the visualization can be categorized into two types: externalization and observation. Externalization denotes the methods to capture the artifacts users actively externalize during an analysis process. Examples of externalization include recording a user's think-aloud and saving the notes a user jots down, both of which intimately reflect the analysis process at a semantic level. Research in visualization has often relied on externalization mechanisms to understand the behaviors of the user. In particular, the think-aloud protocol is frequently used in evaluations and has been found to be effective in reflecting the user's analysis process [33, 41].

On the other hand, observation represents methods that monitor a user using a visualization during an analysis without requiring the user to actively externalize his thoughts. Besides human observers taking notes of what happens during the analysis, devices like eye tracker, video camcorder, EEG (Electroencephalography) and fMRI (functional Magnetic Resonance Imaging) could also be used to record information about user's eye movement, physical motions, neural activities, etc. As noted separately by Huang [33] and Convertino et al. [86], eye-tracking data offers additional insights into typical strategies used for accomplishing given tasks within a visualization environment. Although other devices like EEG and fMRI have seldom been used in experiments related to visualization systems, they are effective for studying brain functions during experimental tasks [26].

Relying solely on external capturing methods has been shown to be effective in discovering a user's (qualitative) mental model when interacting with visualizations for analysis tasks. In an experiment by Trafton et al. [171], video camcorders recorded both the physical environment in which the experiments took place, as well as how the participants interacted with the various computers in the environment (observation). At the same time, the participants were requested to provide think-alouds of their thoughts and take notes of their incremental discoveries (externalization). By combining and manually analyzing these recordings, Trafton et al. demonstrated that they were able to identify how the participants formed mental models of the analysis task and applied the models to solve problems.

4.3.2.3 Internal Vs. External Capturing

Figure 14 illustrates van Wijk's operational model of visualization after integrating both the internal and external capturing mechanisms. According to this model, the four capturing methods are distinguishable and independent from each other. However, in practice, the lines between the methods are sometimes blurry depending on the implementation of the visualization or the physical environment where the analysis takes place. As visual analytic systems become more mature, some analysis that has traditionally been performed outside of a visualization can now be done directly within the visualization. A prime example of this is the inclusion of annotation techniques [73, 80, 44] or "shoeboxes" [188, 156, 180, 130]) in visual analytical tools. Traditionally, such annotations are written on a piece of paper that would have to be collected externally, but in many recent visual analytics systems, annotations have become a part of the visualization that can be captured internally within the visualization.

Practically, one distinguishing factor that separates internal and external capturing is how intrusive they are to the analyst. Internal capturing methods can be implemented directly within the visualization, and are mostly transparent to the user. External capturing methods, on the other hand, often require physical devices or mechanisms that would alter the physical analysis environment and potentially change the analysis process. Certain eye trackers require the user to wear additional hardware [42] that could be cumbersome. Requiring analysts to perform think-alouds during their analysis could be an annoyance to other analysts [121], just as the use of EEG or fMRI are most likely unfeasible due to the monetary cost of the machinery and the cost of time in the setup process prior to use. Even more importantly, in most cases involving external capturing, the fact that the analysts are externalizing their thoughts (e.g., via think-alouds), or are reminded of potential observers (e.g., in the case of being recorded on video) could change their behavior significantly. As noted by Shapiro, performing the think-alouds protocol may slow down a participant's task performance and even alter the process of interest [151]. Similarly, the use of observational tools could solicit an effect known as social facilitation and inhibition in which the participant would either over perform or under perform depending on their confidence in performing the task [16, 194].

4.3.3 Criteria for Assessing Effectiveness in Capturing User Interactions

Given the generally high cost of applying external capturing methods in real-life settings, the visualization community has mostly focused on internal capturing methods for gathering semantic-level user interactions. However, as noted in section 4.3.1, interaction-capturing visualizations vary in their approaches and effectiveness, especially when considering how closely related the captured interactions are to the user's analysis process. So our questions are, why are some visualizations more effective than others in capturing? And is there general principle that can be applied for designing visualizations for better capturing?

We re-examine the internal capturing aspect of our modified model of visualization to better understand how the captured interactions relates to the user's analysis process (Figure 14(A)) . As shown in the model, the captured images (I) connect the visualization to the user's perception (P), while the captured user interactions (dS/dt) relate a user's exploration (E) to the visualization system. In this model, there is no direct link between the visualization and the user's analysis process (or knowledge, K). In order to reconstruct a user's analysis process using only dS/dt and I, one must first make sure that the captured dS/dt and I relate as closely as possible to the user's exploration and perceptual processes.

Based on this observation, we propose that visualizations that are more effective at capturing a user's analysis processes are in fact collecting dS/dt and I in such a way that the captured dS/dt and I can describe the user's intention behind the interaction and the focus and interest of the user's perception with minimal ambiguity.

By examining each access to the processes directly related to user's knowledge in the van Wijk's visualization model for the purpose of disambiguating the meaning of interactions, we identified 3 criteria for evaluating the effectiveness of interaction capturing in a visualization environment.

Semantics of user interactions: First, we examine dS/dt which carries a user's exploration process E and note that the context in which the interaction takes place determines how much semantics are attached to the user interaction. This means that if the user's interaction is in a context where the design and purpose of the visualization is not distinguishable, the meaning behind the interaction is difficult to interpret. We present this criteria as *semantics of user interactions* and propose that higher semantic encoding in the user's interactions would lead to a less ambiguous association between dS/dt and E.

Information change caused by user interactions: Second, we consider what a user perceives (P) between two images $I(S_0)$ and $I(S_1)$ where the change from visualization state S_0 to S_1 is caused by a user's interaction. Intuitively, if the semantic change between the two images is high, it is likely to be difficult to disambiguate the effects of the user interaction and determine which part of the image a user focuses on. This criteria is therefore referred to as *information change caused by user interactions*, and we propose that the more specific the information change is, the easier it is to understand P based on the images I. With these two criteria, we can begin to isolate and disambiguate the semantics of a given interaction.

Degree of interactivity: However, the underlying assumption is that there exists some user interactions to start with, which is not the case in many static and less interactive visualizations. We note that without an adequate amount of interactions that are recorded in an interpretable way, there is no interaction data to capture or analyze. Therefore, the last criteria we propose is referred to as *degree of interactivity* in which a high degree of interaction with a visualization is necessary for reconstructing the user's analysis process.

We believe that the three criteria are orthogonal to each other, and we therefore represent them as the three axes of a cube (Figure 15). In the following sections, we will examine each criteria more closely and attempt to quantify existing visualization systems by using these criteria. It is relevant to note that most complex visualization systems will not appear simply as a dot in the cube but will likely appear as a collection of dots if a visualization contains multiple interaction methods and different representations. However, in our examination of these systems, we will focus on specific techniques that best demonstrate the significance and relevance of each criteria.

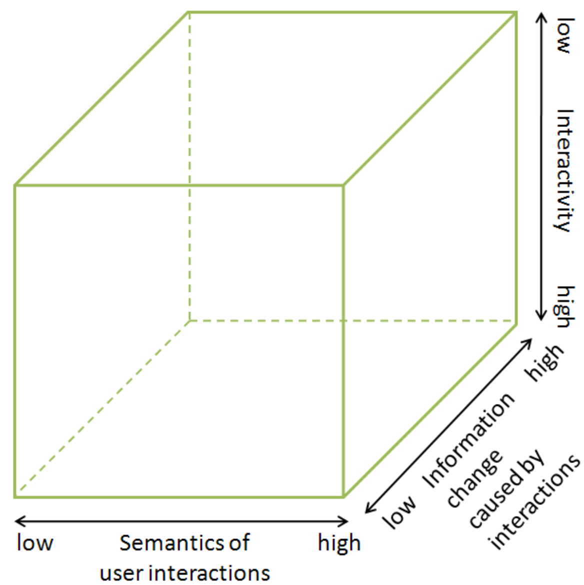


Figure 15: Three criteria for evaluating the effectiveness of capturing user interactions in a visualization environment.

4.3.3.1 Disambiguating dS/dt: Semantics of User Interactions

In the van Wijk's model of visualization, dS/dt is the only output from a user's interactive exploration process (P). In order to interpret dS/dt from the user to the visualization, we need to understand the *semantics* within the interactions. As noted in section 4.3.1, the capturing of interactions can be thought of as either capturing low-level or semantic-level user interaction events. It has been accepted in the visualization community that in reconstructing the user's analysis process, low-level user interactions are insufficient [83].

While the distinction between low and semantic-level capturing (or data vs. information provenance) has been defined, Gotz and Zhou have proposed that there exist additional categorizations of interaction types. Specifically, they characterized the user's activities into four tiers based on their semantic richness: Tasks, Sub-Tasks, Actions and Events. The Events tier corresponds to low-level user interactions. The Actions tier relates to semantic-level interactions and describes them as "atomic analytic steps" (such as explore, filter and zoom). The Sub-Tasks tier refers to concrete analytical goals that are tightly coupled with domain specific problems and the available features within the visualization (such as identifying trends in the financial markets). The Tasks tier categorizes the highest level of the user's analytical goals that are often open-ended or ambiguous (such as generating financial investment recommendations).

From the perspective of capturing a user's analysis process, more semantic information encoded within the user's interactions would lead to less ambiguity during interpretation. Unfortunately, as noted by Gotz and Zhou, user activities above the Actions tier are often domain specific and not easily generalizable. Most existing visualizations that provide frameworks for high-level interaction capturing therefore rely on capturing activities in the Actions tier and are subsequently limited in the encoding of semantics in the user's interactions [73, 68, 78, 156]. When the user's interactions aer more specifically coupled with clearly defined domain problem, researchers have demonstrated that high-level semantics can both be encoded in the interaction as well as extracted during interpretation [41, 92].

4.3.3.2 Disambiguating I: Information Change Caused By User Interactions

We consider the effect of a user's interaction that changes a visualization from generating an image $I(S_0)$ to $I(S_1)$ as the "information change caused by user interactions." Intuitively, for the purpose of disambiguating user interactions, a high amount of information change is not desirable. If a user interaction results in large amounts of information being communicated to the user all at once, it is difficult to interpret what part of the information change is perceived by the user as relevant.

We examine a few existing visualization and interaction designs based on the amount of information change. Highlighting is a common interaction technique that is used to reveal additional information about a visual object. In most cases, highlighting causes minor and specific information change that can be easily interpreted. Zooming, on the other hand, has the potential of changing the overall image I in a drastic way, but the amount of information is specific and localized. In interpreting an interaction that results in zooming, the intention behind the interaction is clear.

There are some interactions that cause high amount of information change. Animation, for example, displays a series of temporal frames given a single user interaction (such as a mouse click). Since the viewers need to keep the changing visual objects in memory for association [105, 140], visual objects with rather complex movement over a long period of time would result in high amounts of information change since the specific information relevant to the user would be lost. In this regard, complex animation such as that in Gapminder [58] would cause a higher amount of information change than simpler animations that are used to depict the transitioning between statistical states [79]. Another example of potentially high information change are the interactions within Coordinated Multiple Views (CMV). Many notable visualization systems apply the CMV interface, including Xmdv [181], Spotfire [3], and ILOG Discovery [12]. However, as Roberts noted [139], as the number of coordinated views increases, it becomes harder for the user to keep track of the contexts and relationships between the views. In terms of information change, this means that interacting with more coordinated views will result in higher information change as the simultaneous updates in all views make it difficult to isolate the meaning and intent behind the interaction.

4.3.3.3 Degree of Interactivity

The basic assumption made in considering the previous two criteria is that there are in fact some capturable user interactions within a visualization. According to Card, Mackinlay and Shneiderman who defined information visualization as "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" [24], the assumption might be true. However, not all visualizations incorporate the same degree of interactivity. In fact, literatures in ambient [159, 7] and casual InfoVis [134] suggest that analysis could be performed without any user interaction (Figure 16).

From the perspective of capturing a user's analysis process, a high degree of interactivity within the visualization is preferred. Ideally, the analysis process should be driven by the user's interactions so that there is a sufficient amount of captured information for every step of the user's analysis. If

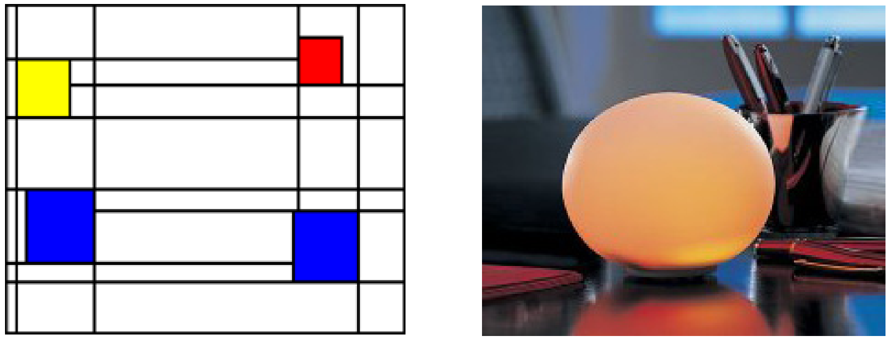


Figure 16: Examples of non-interactive ambient InfoVis: informative art (left), Ambient Orb (right)

the visualization is more static in nature, the user's analysis process would not manifest itself as recordable interactions, and will remain internal to the user.

Therefore we present the degree of interactivity of a visualization as the third and final criterion to capturing a user's analysis process. Examples of information visualization with low interactivity include systems in casual and ambient InfoVis where no user interaction is required; whereas the other end of the spectrum is exemplified by systems that rely on the user's interactions to *drive* the visualization. For example, in the data visualization software Tableau [113], the user's interactions are part of the process of constructing a query in VizQL [162]. Similarly, in ScatterDice [47], the interaction controls the transition between dimensions of a scatter plot. During the transition, the animation gives rise to the user's understanding between the data and the dimensions. Without the interaction, the visualization cannot express the relationships in the data effectively.

4.3.3.4 Principles for Efficient Interaction Capturing

Figure 15 shows the three proposed criteria as independent dimensions in evaluating the effectiveness of visualizations in capturing a user's interactions for the purpose of reconstructing a user's analysis process. We propose that for a visualization to be effective, it needs to rank highly in all three dimensions. In other words, visualizations with high interactivity, semantically rich interactions and low information change during interaction would tend to be more effective at capturing a users analysis process. It is important to note that the three criteria do not compensate for each other in that scoring highly on two dimensions and receiving a low score on one will still render the visualization ineffective in capturing. For example, a system with high interactivity, low information change but which captures user interaction with little semantic information would only result in gathering low-level user interactions that could not be used towards reconstructing a user's analysis process.

In the visualization community, many systems are designed with high interactivity as a core feature and would therefore rank highly under the criterion of *degree of interactivity*. However, many of them are also designed to be broadly applicable to multiple domains which limits their ability in capturing information about a user's behavior beyond the Actions tier and would therefore receive an average grade in *semantics of user interaction*. Finally, if these systems further employ interaction techniques that cause high *information change* such as multiple coordinated views or complex animation, it would further reduce their ability to capture a user's analysis process.

We specifically examine the results of the study described in section 4.2. While there are factors external to the design of the visualization that might affect the accuracy of the correlation (such as the effectiveness of the interpretation of captured user interactions, the effectiveness in integrating the internal and external captured artifacts, etc.), we focus on applying the proposed three criteria in evaluating our results. We find that WireVis, the visualization system used in this study, scores highly in *degree of interactivity* as well as *semantics of user interaction*. However, WireVis employs a multiple coordinated views interface which would make it difficult for the interpreters to disambiguate the intent behind user interactions that cause *information change* in the coordinated views. In practice, the negative effect of *high information change* is likely to be limited since WireVis only uses four coordinated views [139]. Our evaluation of this study is consistent with our earlier finding that many of the mis-correlations during the interpretation stage stem from the interpreters not knowing on which part of the visual change the experts were focusing.

4.3.4 Discussion

Given that most existing visualization systems that capture a user's interactions focus on capturing in the Action tier under Gotz and Zhou's classification [68], we feel that it is necessary to further discuss the practical differences between capturing interactions in the Action and Sub-Tasks tiers. Specifically, we demonstrate a case in which a user's interaction would be considered ambiguous if only Action-tier semantic information is encoded, but would be less ambiguous if additional Sub-Task or Task tier information is available.

Figure 17 shows an image that integrates multiple sources of data, including 2D imagery, 3D GIS information, and real-time cell phone usage of the inhabitants of Rome, Italy [22]. While this view conveys all three sources of information coherently, a user's interaction with this visualization might not be easily relatable to the view. If the user were to select a region-of-interest on the map and request for more detail, it might be difficult to interpret if the user is looking for information relating to the 2D imagery, the 3D location, or the cell phone if the user's interaction is encoded only with Action-tiered semantic information. Under Gotz and Zhou's definition [68] of Actions, the request for more detail would be defined as a "Query" command with additional parameters on the geo location of the interaction. This definition does not have the means to distinguish what the intent of the interaction is, or what the user's goal in performing the interaction might be. In contrast, if additional Sub-Task or Task tier information is available, the same interaction would not have the same degree of ambiguity as the context of the interaction would be clearer.

This example illustrates the limitation of capturing interactions only at the Action tier. In order to disambiguate such interactions, additional constraints and considerations in the visualization design might be necessary. Specifically, the contexts of a user's interactions need to be clear and uniquely distinguishable at all times. In the example above, this could mean only displaying one layer of information at one time.

While sacrificing visualization design principles for the sake of better interaction capturing would appear to imply that the two are contradictory, we suggest that in most instances they are not. In fact, two of the three criteria, i.e., rich *semantics in user interactions* and high *degree of interactivity* can be incorporated into visualization systems without contradicting general visualization design principles. The criterion for restricting *information change* is more controversial in that many well-established and effective visualization systems employ interactions that result in high information change. Coordinated multiple views is a prime example in which the goal of instrumenting effective capturing mechanisms interferes with general visualization designs, but it is also the only example

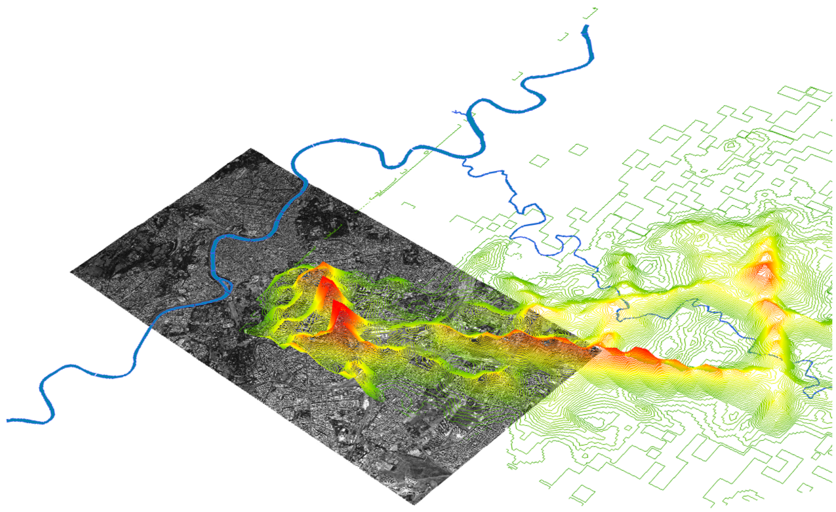


Figure 17: An image from MIT SENSEable City Lab depicting real time cell phone usage in Rome. This image integrates multiple sources of data into one single view.

that we have encountered. The case of complex animations causing high *information change* would appear to be another example due to the success and widespread acceptance of Gapminder [58]. However, as Robertson et al. [140] recently noted, such complex animations might not be the most effective technique in analyzing time-varying data. Instead, the use of complex animations is better suited for presentation of known trends and patterns.

CHAPTER 5: DISCUSSION

This thesis is, in spirit, a compilation of a wide array of research and publications on interactions and visualization. As in all publications, any findings that have not been validated cannot be printed or reported. However, from a perspective of a PhD thesis, there are a few interesting lessons that we have learned over the years that could be of interest to the visualization community. The purpose of this chapter is to report some of the intuitions that we have gained and general design principles that we have used in the development of these visualizations.

5.1 Coordinated Visual Analytical Systems

One of the most frequently utilized design principle in the visualizations presented in this thesis is the coordinated multiple views visualization (CMV), which is also sometimes referred as multiple-coordinated views. In essence, CMV is a design that simultaneously displays the same data in different views, but shows different attributes (perspectives) of the data in each view. Interactions in one view (such as brushing and selection) would also highlight the same data item in the other view(s). Using this mechanism, it is easy to identify patterns interactively in an exploratory fashion. For example, in WireVis, if a user highlights a specific keyword in the heatmap view, the user could see whether the selected keywords show a pattern in the temporal (strings and beads) view, or the keyword network view. Similarly, in UrbanVis, if the user highlights areas with high income in the Information View, those areas would appear in the 3D Model View such that the user could relate the same data under two perspectives.

In the iPCA system, this concept is pushed further to utilize a user's interactions to correlate two different views without specific highlighting or selection of data items. This differs from traditional CMV in that the emphasis is not to look for different perspectives of the same data items, but to look for understanding in the correlations between the perspectives themselves. In principle component analysis, the correlation is inherent in the mathematical transform of the spaces, and understanding the correlation imply having the high-level sense of the PCA technique. While the system and the design itself is successful for PCA, it is difficult to imagine using it to understand any two views in non-mathematical transform (such as the views in WireVis) because the correlations might not exist or easily identifiable through such interactions.

Nonetheless, we have found the use of CMV in designing explorative visualization and visual analytical systems to be ideal. In most large datasets that we have encountered, there is often no single view that is representative of the characteristics of the data. From the perspective of the end user (who has no prior knowledge of the data, but is looking to explore it), having multiple views that depict the different aspects of the data opens up the exploration process. Depending on the user's interest or prior knowledge, the user could begin the exploration process through any one particular perspective that is of interest or familiarity. In the case when the user has no background in the data or the task, the coordinated views give the user the sense of all the important perspectives of the data, and helps the user to begin building an overview of the dataset.

CMV designs are common in designing visualizations. There is in fact a small annual conference devoted to the methodologies of CMV-related research (Conference on Coordinated and Multiple Views in Exploratory Visualization). Our CMV designs are similar to the spirit of the conference. However, we differ in that we have a particularly strong emphasis on high degrees of interactivity. For instance, highlighting through mouse hovering is a common technique used in all of our systems that allows the user to "query" the visual elements without mouse clicks. While the highlighting technique is trivial to implement, we have found the implication to be significant. In WireVis, highlighting through mouse hovering allows the user to quickly navigate through all keywords and scan for patterns in the other views. In the biomechanical-motion analysis system, the highlighting allows the user to look at each frame in the 3D window by moving the mouse across the information visualizations to identify important key frames within the motion. With this technique coupled with a CMV design, the user could quickly explore a large dataset without having to "commit" to any specific action such as a mouse click. For a novice user who is new to a visualization, clicking on a mouse is often a much more “severe” action than moving a mouse around. The highlighting thus allows the user to get involved with using the system without the fear of “breaking something” by clicking on it. For an experienced user, this form of highlighting functions as a short-cut to analyzing a new dataset. For example, we have often observed WireVis users to begin the analysis by moving the mouse across either all keywords or accounts clusters to get a “sense” of the data.

Although it is not clear if the CMV design is applicable to all visualization designs, we have found this methodology to be particularly effective in designs for explorative visualizations and analytic applications. Especially when combined with high degrees of interactivity, the CMV interface has been shown to be useful to a wide range of users from novices to expert analysts alike. For future developments of visualizations that share similar goals and characteristics, it is a method worthy of consideration.

5.2 Knowledge and Visualization

Visualization has often been connected to data, information, knowledge, and insight in different ways. For many, data visualization refers to visualization of scientific data where the data are numerically sampled from physical objects and phenomenons (such as CT scans, weather models, etc.). This type of visualization tend to be 3D in nature, where the objective is to depict the physical object or phenomenon in a clearer way. Information visualization implies visualizing data that are a mixture of categorical and numeric. This type of visualization are typically more abstract (such as bar-charts and line graphs) and are central to this thesis.

The notion that scientific visualization is visualizing data but information visualization is not is the result of a naming convention that dates back to the founding of the visualization conference. However, it has been a source of confusion for the newer generations of researchers in the field. Chen et al. have made a brief attempt in clarifying the terminology, but a lot more could be done to disambiguate the terminologies between data, information, knowledge, and insight [28, 26].

The focus on this discussion is the utilization of knowledge to create or enhance visualizations. As noted by Chen et al. [28] and Wang et al. [179], knowledge is difficult to define. However, even without a clear definition of what knowledge is, it is still useful to consider the incorporation of domain knowledge in visualizations. For example, in our work on simplifying urban models using the concept of urban legibility, the knowledge used to drive the simplification is based on knowledge in the architectural community of how humans perceive an urban environment [112]. In section 2.1.2, we discussed the problems with using quantitative pixel-oriented measurements, and the benefits of using the qualitative (knowledge oriented) approach based on preserving the elements of legibility. As we demonstrated, when the goal of the final visualization is based on a relevant domain-specific knowledge, the visualization would tend to be "appealing" to the end users.

Knowledge, however, does not always have to be based on low-level perceptual or cognitive principles. In section 3.1, we discussed a user-centric method to understand how financial analysts at Bank of America approach the task of searching for suspicious financial wire transactions. The resulting design of the visualization is therefore a close reflection of the process and procedures in which the analysts perform their daily jobs. For example, we adopt the concept of searching for wire transactions based on keyword matches. The list of keywords used in WireVis is the same list used at Bank of America and reflects and encapsulates the collective knowledge of the analysts on wire transaction fraud. This list of keywords is frequently updated based on new intelligence report or current international state of affair that includes geographical locations, business types, goods and services, etc. Due to the comprehensiveness of the keywords, the analysts could construct hypotheses and notice unusual patterns of behaviors through examining the keywords alone (section 3.1.3.4).

In recent years, a new thread of research called "knowledge-assisted visualization" has been established that looks to promote the concept of utilizing external knowledge structures to aid the visualization in an intelligent manner. The external structures could be a simple database or ontology that organizes the data in a cohesive manner [65], a collection of logic-based statements [189, 59], or a more complex AI program that reacts to a user's interactions [107]. During a user's interactions with a visualization, the user could manually pull information from the external source (such as looking for an appropriate query to search for a specific pattern of network traffic [189, 59]), or have the visualization semi-automatically respond to the user's interests and identify key information for an analysis [107]. The challenges in knowledge-assisted visualizations are not trivial and include defining what knowledge is, identifying and constructing appropriate knowledge structures, and determining how to utilize and integrate a visualization to the knowledge structure. However, applications that utilize knowledge structures successfully have already demonstrated great promise and could become an integral component to visualizations in the future.

5.3 Understanding Financial Compliance Analysts

One of the greatest difficulties in designing and developing visual analytics applications is that the end-users are often difficult to find. As many of the visual analytical tools built to date are focused on domain experts (such as intelligence analysts), the number of potential users is inherently small. During our studies in understanding how financial analysts use the WireVis system (section 4.2), we were fortunate enough to have had the opportunity to work with real financial analysts from various financial institutes in Charlotte. Although we did not set out to do a case study on how financial analysts approach general analysis problems (similar to the case studies by Chin et al. [30] and Fink et al. [50] on intelligence and internet security analysts respectively), we nonetheless observed some interesting behaviors from the financial analysts that could be potentially relevant to the development of future finance-related visualizations.

The task that our financial analysts undertook in the study was to identify a few suspicious transactions in a sea of normal transactions, which has been described as "finding a needle in a stack of needles" by Bill Fox, the Senior Compliance Executive for financial crimes at Bank of America (who is also formerly the director of the Financial Crimes Enforcement Network, FinCEN, the institution responsible for administering the Bank Secrecy Act). In spirit, such a task is similar in nature to internet security where the goal is to locate suspicious and malicious internet (IP) traffic in thousands to millions of transactions. However, in practice, security analysts' jobs are slightly different from those of financial analysts in that IP traffic have distinct signatures and patterns that can be revealed by examining numbers alone (time, date, port number, etc.), whereas suspicious financial transactions are often text that need to be put into context. For example, a large transaction from China to the U.S. is not suspicious by itself. However, if the transaction is sent by a company manufacturing baby foods to a company selling car parts, the transaction would likely warrant a closer scrutiny.

In this sense, Bill Fox has described the jobs of financial compliance analysts at Bank of America's WireWatch group to be "half art, half science." Since no single financial transaction is suspicious by itself, financial analysts rely on their intuition to determine which transaction(s) to investigate. In a live session with Jim Price, the head of WireWatch, we observed how Jim was able to scroll through pages of Excel-like spreadsheets and locate transactions that had a high degree of interest. To those of us who had not been trained in financial analysis, all the rows in these spreadsheets appeared to be exactly the same. How Jim managed to weave together "stories" for the transactions to determine if they warranted further investigation was something that Jim himself could not explain, and it remains a great mystery to us.

In designing a visualization for these financial analysts, we were tasked to "bring the science to the art." Since the "art" (or the intuition) is difficult to train, but the use of a visual analytical tool is procedural and reproducible (and therefore more "scientific"), bridging the gap between the two would be the key to creating a successful tool. While we believe that WireVis achieved that goal to a certain extent, the tool itself does not represent all aspects of a financial analyst's investigation. As we have learned through various studies and observations, there are many aspects of financial compliance analysis that are beyond the scope of a single application.

Similar to the findings of Chin et al. and Fink et al. [30, 50], financial compliance analysis also requires a broad spectrum of tools, each for a different component of understanding the context of the targeted set of activities. Google, for example, is an integral part of analysis because not all the transactions' originators (sender) or recipients (receiver) hold accounts at Bank of America. Should the sender or the receiver be a customer of Bank of America, the analyst could correlate the person's financial activities across multiple transaction methods (credit cards, ATM withdraws, check, etc.) and pull together a "story" about the person. However, if Bank of America does not have direct access to the person's account activities, the analysts would rely on gathering as much information as they could online. In addition, TV, online news sources, intelligence reports, etc. often provide clues to the "hotspots" of suspicious activities both domestic and international. Given the multitudes of channels of information, WireVis only provides a small subset of the data that the analysts need to comb through on a regular basis.

However, just within this small subset of data (i.e., wire transactions), we have already observed that the amount of information often overwhelms the analysts. Specifically, we have found that even during a short investigation period (30 minutes) of using WireVis, the analysts had difficulties remembering what they had searched through or what the results of the analysis were after the investigation period. The analysts would therefore occasionally repeat their own analysis and identify the same suspicious transactions or accounts multiple times. It is relevant to note that these repeating investigations often did not follow the same paths, but would nonetheless lead to the same findings. For example, an analyst could begin an investigation on transactions relating to the keywords Mexico and Pharmaceutical and identify a specific transaction. Later on, the analyst could search on transactions above a certain amount and during a specific time period, and inadvertently discover the same transaction again. Empirically, we have found that the analysts could only remember the initial strategy of the investigation (e.g., to investigate transactions relating to the keywords Mexico and Pharmaceutical), but could not remember if or how the strategy changed during the course of the investigation, or results of the finding (e.g., information about the transaction or the account).

This observation of an analyst's memory during an investigation is interesting in multiple ways. For one, we realized that the current standard of reporting financial investigations (where the analysts would prepare the report after the investigation) inevitably leads to reports with inaccuracy. Specifically, during our study, we found that when the analysts were pressed to report what they did during the investigation, they had a tendency to fill in the memory gaps based on what they thought should have happened, which did not always reflect what actually took place during the analysis. The obvious implication of this finding is the need for visualizations and tools to keep track of what the analysts have done for the purpose of reporting. However, just as importantly, this observation raises the question of "how complete is an investigation?" Specifically, if analysts do not remember what analyses have been done, how would they know what investigations haven't been done?

Consider a hypothetical situation in which an analyst filed 100 Suspicious Activities Reports

(SARs) last year. If his fellow analysts had filed 50 SARs in the same time span, then 100 reports would seem to be a lot. However, this comparison is incomplete because there is no baseline of the number of SARs that should have been filed. Without this baseline, we have little or no way of judging the significance of filling 100 SARs in terms of sufficiency and efficiency in discovering financial fraudulent activities. While this theoretical baseline may never be obtainable, what we could know (through tracking an analyst's activities in a visualization) is where the analysts have searched within all the transaction records, but more importantly, where the analysts have *not* searched.

After our experience working with financial compliance analysts, we have discovered a great number of opportunities to better their investigation processes through the use of visual analytical tools. It is our opinion that understanding an analyst's search space is a critical area that has not received the necessary attention. The recent increase in research in the areas of interaction capturing and reasoning extraction have shown great potential, and in time we hope that we will be able to better understand financial compliance analysts through understanding their behaviors during investigations.

CHAPTER 6: CONCLUSION AND FUTURE WORK

We have presented work from a wide collection of papers and research efforts that include topics on maintaining interactivity in urban visualizations, applications of interactions in visual analytical tools, and understanding reasoning through examining a user's interactions. Through this thesis, we hope that we have demonstrated the importance of interactivity in visualizations and raised awareness that interaction plays just as important role (if not sometimes a more important role) as visual representation in visual analytical tasks. Although we are still a long way from truly understanding the nature and the science of interactions, we hope that this thesis serves as a small step forward in the right direction.

An important consideration that underlies the interaction techniques and their corresponding visual representations is how the interactions relate to the task that the visualization system seeks to analyze. Although this consideration is not explicitly described in each section of this thesis, it is nonetheless an important criteria in designing a successful visualization and visual analytical system. In the previous section, we note the general design principle behind coordinated multiple view (section 5.1) and the notion that knowledge could be embedded and utilized to enhance a visualization (section 5.2). When the two are combined together, general design guidelines for visualizing large scale data begin to emerge. As noted in section 5.2, incorporating known domain knowledge into the visualization could reflect processes and procedurals of an expert's reasoning process. The question of how to identify domain knowledge and discretize such relevant but abstract concepts into a visual system is at the heart of designing a successful visual analytical system.

The discussions about financial analysts (Section 5.3) is an example of the depth and understanding of the target user that forms the first step in being able to transform and embed domain knowledge into a usable visualization system. Although we briefly describe this process as a *user-centric design* in presenting the implementation of WireVis (section 3.1.3.1), the process does not adequately reflect what user-centric design means to the design and creation of the WireVis system. Of the routines, processes, methodologies, and characteristics of financial analysts that we studied and learned over the span of over a year (Section 5.3), only a small fraction is represented in the WireVis system. However, the small fraction that became WireVis is certainly important, and in many ways embodying of the analysts' fraud detection process. Although not all the systems described in this thesis underwent the same length and rigor in understanding the targeted users, similar processes of learning domain experts and analytical tasks took place. In designing the GTD (Global Terrorism Database) visualization, we collaborated with experts at the START Center to better understand the definition of terrorism and the process for investigating terrorism-related activities. Similar collaborations took place in designing the simplification algorithm for urban models to retain interactivity (with architects and urban designers), as well as designing the visualization for the analysis of bio-mechanical motion data (with evolutionary biologists).

When considering domain knowledge in the context of the coordinated multiple view (CMV) approach, embedding domain knowledge into interaction techniques and visual representations could be thought of as identifying perspectives of either the data or the targeted user's analytical processes. These perspectives are then correlated to appropriate visualizations to become a visualization system. Under the CMV design, every perspective is depicted using a specific visual representation, and the interaction techniques serve as the conductive tool that allows the user to synchronize and integrate the different visual representations into a cohesive analytical environment. In WireVis, the perspectives are based on the data and data types that are of interest to the financial analysts (where each visual representation is related to a perspective of the data, such as the relationship between keywords and accounts). In visualizing the GTD, the perspective is based on the process of investigation through the examination of the 5 W's (who, what, when, where, and why). In visualizing bio-mechanical motion, the perspective is to overcome a specific deficiency in how evolutionary biologists analyze motion data (in that watching movies of the motion data does not easily lead to analysis and comparison of the motions). Regardless of the domain, we have found that embedding deep understanding of domain knowledge in an interactive CMV environment leads to successful visualization that the end users find both useful and often enlightening.

With the visual representations embodying aspects of specific domain knowledge, the appropriate interaction techniques are sometimes obvious. Coordinated brushing and linking (such that interactions with one view is represented in another) is a clear compliment to CMV-based visualizations. However, as described in this thesis, there are often additional considerations in designing the interactions that affect the visual representations. Considerations such as maintaining an interactive frame rate, handling a large-scale and wide-scope dataset, capturing and extracting the user's interactions and reasoning processes, etc. often require modifications to the design and implementation of the visual representations. While the two aspects of visualization cannot be easily disentangled from each other, it is important to note that these considerations on interactions do not contradict the principles of visual representation designs that are based on domain knowledge and the analytical processes of the end users. In most instances, the "perspectives" do not need to modified, only the specific implementation of the visual representations (such as the underlying data structures) need to be altered. For example, in the case of WireVis, connecting to a large database required additional strategies that pushed the clustering onto the database as opposed to the visualization (see section 3.1.4). One of the most important contribution of this thesis is therefore in the overall consideration of the design of visual representations along with the appropriate interactions that make the analytical process possible.

Finally, in concluding this thesis, we isolate the three areas of work based on our contributions, and discuss some possible future directions for continuing the research.

6.1 Interactive Urban Visualization

We demonstrate that our work on legibility-based simplification of urban models that retain the salient features of a city is an important and necessary component to interactive visualizations of large urban environments. The criteria and metric of the simplification is based on the five elements of urban legibility, and we show that as long as these elements are preserved during simplification, the image of the city could be retained.

However, as we have found out, the image of a city is not just related to its geometric properties.

In fact, a city has many faces and can be viewed from different perspectives. Our work with UrbanVis therefore combines the simplified 3D model to an information panel that visualizes additional attributes of a city (such as the Census). We demonstrate that by integrating the two views through user interaction, it allows the user to get a more complete understanding of a city. In addition, we further extend the interaction to incorporate the use of probes, which opens up the possibility for a user to explore multiple regions of interests simultaneously. By juxtaposing multiple probes that represent different regions of interests, the user can compare the similarities and differences between them easily. While the misuse of probes could begin to push the limits of rendering speed at interactive rates in some urban visualization systems such as UrbanVis, the benefits of such interactivity nonetheless outweigh the potential cost in most scenarios that we studied.

In looking forward with urban visualization, we will be looking to understand the elements of urban legibility from a different perspective. We have begun investigating how inhabitants of a city form their mental picture of the city through the use of the legibility elements. The goal is to create a visualization that could semi-automatically generate sketch maps of a city by integrating multiple layers of GIS data (such as business types, zoning, traffic, crime rates, etc.). In this sense, we seek to expand the definition of urban legibility to incorporate non-geometric elements, which are becoming more and more important in understanding today's cities.

In addition, the generation of sketch maps could start to address how individuals understand cities differently. As noted in section 2.2.4, one challenge to creating a "legible city" is to accommodate each individual's sense of legibility. Our interactive sketch maps will allow the user to modify the importance of each GIS layer so as to assist the users in orienting themselves using the legibility elements that they are most familiar with. Understanding how a person forms a general sense of spatial awareness is an on-going research across multiple disciplines. However, we hope that with tightly integrated interactions within a visualization, we could begin to identify how interactive urban visualizations could assist people in better recognize and navigate their environments.

6.2 Applications of Visual Analytical Systems

We use four different visualization systems to showcase the benefits of tightly and appropriately integrated interactions. These four systems tackle four challenging problems: financial fraud analysis, global terrorism analysis, biomechanical motion analysis, and analysis with PCA. Although the four problems required different visualization approaches, the key component that connects them is the role of interactions that drives the analysis. Without interactions, a user could not explore large amounts of financial transactions to locate a few suspicious activities, input partial knowledge of terrorist incidents and identify global patterns and trends, select and compare motion segments and analyze similarities and differences, or correlate mathematical spaces to gain intuitive understanding between the two. Collectively, these systems empirically demonstrate the necessity of interactions and justify their cost.

From a design perspective, it is unfortunate that we still cannot easily identify the purpose and nature of the interactions used in these systems. Yi et al. [190] have categorized interaction techniques into seven categories (select, explore, reconfigure, encode, abstract/elaborate, filter, and connect), but these distinctions are specific to individual interaction techniques and do not reflect the combined effects of all the interactions in a visualization. In the four presented systems in this thesis, every system utilizes interaction techniques from nearly all the categories. For example, in WireVis, the user could select a transaction, explore by highlighting different keywords, reconfigure data through reclustering, encode and alter visual attributes of the strings and the beads, abstract/elaborate by zooming in and out of clusters, filter by selecting a region in the heatmap view, and connect keywords to reveal their relationships in transactions. However, while the interactions in WireVis could be deconstructed modularly into these seven categories, it is still unclear what the interactions in WireVis do collectively that allow the user to perform in-depth analysis.

The next steps in advancing the design and implementations of interactive visualizations should therefore be a deeper understanding of the nature of interactions. One observation that we've had as developers is that we always design visualization systems by creating visual elements first, and then deciding how we could add interactions to them. If visual representation and interactions are indeed equal [190], we should be able to reverse the process of designing visualizations occasionally and start the design with creating the interactions that are needed for the task before thinking of the visual representations.

While this goal seems reasonable, it is a relatively difficult task given the abstract nature of interactions. As designers, it is easy to "envision" what an interface would look like without considering how a user would interaction with it. On the other hand, if the design were to start with interactions without the aid of visuals, it is much more difficult to grasp what the system would "look like." It is therefore important for the researchers in this area to consider the next step in the taxonomies behind the categorization of Yi et al. [190] and attempt to identify the nature of interactions in a way that designers could utilize the taxonomy as a design guide.

6.3 Interaction Capturing and Reasoning Extraction

As we demonstrated in the four systems, using visualizations that are tightly coupled with appropriate visualizations, a user could have deep and complex analysis of the data. When a user becomes involved in using such a visualization, the reasoning process that the user utilizes to solve a task is reflected in the user's interactions. The goal of our research on interaction capturing is to find out if such reasoning process is capturable, and more importantly, extractable.

The result of this study is very significant. While there are many attempts at capturing user interactions for the purpose of extracting reasoning processes, their effectiveness have varied greatly, including several projects that concluded that such extractions could not be done. The result of our study show that with our visualization and our methodology for capturing and extracting a user's reasoning process, 60-80% of the user's strategies, methods, and findings are in fact recapturable. While we understand that there are many factors that could contribute to the results, we nonetheless demonstrate through our study that gathering a user's reasoning process through analyzing a user's interactions is promising, and there are many aspects to improve on the design and efficiencies of the capturing and extraction methods.

We discuss the lessons that we learned during the study in section 4.3. The observations and models presented in the section are both indicators that we would look to apply and refine in the future. One challenging aspect to furthering this research is in identifying and developing an appropriate visual analytical task (with a sufficient number of real-world users) before applying the capturing mechanism. Since we are interested in extracting reasoning processes, an expert user with strong domain knowledge would likely provide more relevant and appropriate interactions than a novice without a clear sense of how to approach a specific task.

Furthermore, we would like to identify a more generalizable method for both capturing and extraction. On the capturing side, we have captured high-level interactions that are only relevant to the particular task and visualization. In order to make the capturing method generalizable, we would need to adopt a structure similar to that of the P-Set Model [91], but utilize a structure suited for higher level (reasoning level) instead of the low-level structure (user-interface level) used in P-Set. On the extraction side, we have performed the extraction manually in our study (through the use of graduate students who are familiar with visualization research), but the ideal goal would be to train computers to do semi-automatic extraction. This is an area where we have begun to investigate, but will be seeking collaborators in machine learning or AI to fully understand the important features in the interactions. We do not foresee in the immediate future that computers or AI models would surpass people in understanding reasoning processes through analyzing interaction logs, but from the perspectives of practicality and generalizability, a semi-automatic extraction method would be important.

REFERENCES

[1] *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2006). Conference Chair-Enrico Bertini and Conference Chair-Catherine Plaisant and Conference Chair-Giuseppe Santucci.

[2] Abello, J., and van Ham, F. Matrix Zoom: A visual interface to semi-external graphs. In *INFOVIS: Proceedings of the IEEE Symposium on Information Visualization* (2004), IEEE Computer Society, pp. 183–190.

[3] Ahlberg, C. Spotfire: An information exploration environment. *SIGMOD Record 25*, 4 (1996), 25–29.

[4] Ahlberg, C., Williamson, C., and Shneiderman, B. Dynamic queries for information exploration: an implementation and evaluation. In *SIGCHI* (1992), ACM.

[5] Ahonen-Rainio, P. Concept testing of some visualization methods for geographic metadata. In *ScanGIS* (2003), pp. 103–117.

[6] Aiken, A., Chen, J., Stonebraker, M., and Woodruff, A. Tioga-2: A direct manipulation database visualization environment. 208–217.

[7] Ambient Devices. Ambient orb. http://www.ambientdevices.com/cat/orb/orborder.html.

[8] Anders, K.-H. Level of detail generation of 3D building groups by aggregation and typification. In *International Cartographic Conference* (2005).

[9] Anderson, E., Silva, C., Ahrens, J., Heitmann, K., and Habib, S. Provenance in comparative analysis: A study in cosmology. *Computing in Science & Engineering 10*, 3 (May-June 2008), 30–37.

[10] Asuncion, A., and Newman, D. UCI machine learning repository, 2007.

[11] Baldonado, M. Q. W., Woodruff, A., and Kuchinsky, A. Guidelines for using multiple views in information visualization. 110–119.

[12] Baudel, T., Haible, B., and Sander, G. *Visual Data Mining with ILOG Discovery.* Springer, 2004.

[13] Bavoil, L., Callahan, S., Crossno, P., Freire, J., Scheidegger, C., Silva, C., and Vo, H. VisTrails: enabling interactive multiple-view visualizations. *Visualization, 2005. VIS 05. IEEE* (Oct. 2005), 135–142.

[14] Bier, E. A., Stone, M. C., Fishkin, K., Buxton, W., and Baudel, T. A taxonomy of see-through tools. 358–364.

[15] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. 73–80.

[16] Blascovich, J., Mendes, W. B., Hunter, S. B., and Salomon, K. Social "facilitation" as challenge and threat. *Journal of Personality and Social Psychology 31* (1999), 422–429.

[17] Boukhelifa, N., Roberts, J. C., and Rodgers, P. A coordination model for exploratory multi-view visualization. 76–85.

[18] Brand, M. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications 415*, 1 (2006), 20–30.

[19] Brewer, C. Designing better maps: A guide for GIS users.

[20] Brewer, I., MacEachren, A. M., Abdo, H., Gundrum, J., and Otto, G. Collaborative geographic visualization: Enabling shared understanding of environmental processes. *Information Visualization, IEEE Symposium on 0* (2000), 137.

[21] Butkiewicz, T., Chang, R., Wartell, Z., and Ribarsky, W. Visual analysis of urban change. *Computer Graphics Forum (IEEE Eurovis Conference) 27*, 3 (2008), 903–910.

[22] Calabrese, F., and Ratti, C. Real time Rome. *Networks and Communications Studies 20*, 3-4 (2006), 247–258.

[23] Callahan, E., and Koenemann, J. A comparative usability evaluation of user interfaces for online product catalog. 197–206.

[24] Card, S. K., Mackinlay, J., and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[25] Chang, R., Wessel, G., Kosara, R., Sauda, E., and Ribarsky, W. Legible cities: Focus-dependent multi-resolution visualization of urban relationships. In *Transactions on Visualization and Computer Graphics* (2007), vol. 13, IEEE Computer Society, pp. 1169–1175.

[26] Chang, R., Ziemkiewicz, C., Green, T. M., and Ribarsky, W. Defining insight for visual analytics. *IEEE Computer Graphics and Applications 29*, 2 (2009), 14–17.

[27] Chen, J., Forsberg, A. S., Swartz, S. M., and Laidlaw, D. H. Interactive multiple scale small multiples. In *IEEE Visualization 2007 Poster Compendium* (2007).

[28] Chen, M., Ebert, D., Hagen, H., Laramee, R. S., van Liere, R., Ma, K.-L., Ribarsky, W., Scheuermann, G., and Silver, D. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl. 29*, 1 (2009), 12–19.

[29] Chi, E. H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R., and Card, S. K. Visualizing the evolution of web ecologies. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems* (1998), ACM Press/Addison-Wesley Publishing Co., pp. 400–407.

[30] Chin, Jr., G., Kuchar, O. A., and Wolf, K. E. Exploring the analytical processes of intelligence analysts. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 11–20.

[31] Chung, C. J., Ed. *Great Leap Forward*. Taschen, 2001.

[32] Combs, T. T. A., and Bederson, B. B. Does zooming improve image browsing? In *ACM Conference on Digital Libraries* (1999), pp. 130–137.

[33] Convertino, G., Chen, J., Yost, B., Ryu, Y.-S., and North, C. Exploring context switching and cognition in dual-view coordinated visualizations. In *CMV 03: Proceedings of the conference on Coordinated and Multiple Views In Exploratory Visualization* (Washington, DC, USA, 2003), IEEE Computer Society, p. 55.

[34] Cowley, P., Haack, J., Littlefield, R., and Hampson, E. Glass box: capturing, archiving, and retrieving workstation activities. In *CARPE '06: Proceedings of the 3rd ACM workshop on Continuous archival and retrieval of personal experiences* (2006), pp. 13–18.

[35] Cowley, P., Nowell, L., and Scholtz, J. Glass box: An instrumented infrastructure for supporting human interaction with information. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05* (January 2005), pp. 296c–296c.

[36] Crisco, J. J., Coburn, J. C., Moore, D. C., Akelman, E., Weiss, A.-P. C., and Wolfe, S. W. In vivo radiocarpal kinematics and the dart throwers motion. *The Journal of Bone and Joint Surgery 87* (2005), 27292740.

[37] Cullen, G. *The Concise Townscape*. Butterworth Architecture, 1961.

[38] Dalton, R. C. Is spatial intelligibility critical to the design of large-scale virtual environments? In *Journal of Design Computing 4. Special Issue on Designing Virtual Worlds* (2002).

[39] de Smith, M., Goodchild, M., and Longley, P. *Geospatial Analysis: A comprehensive guide*. Troubador Publishing Ltd, 2007.

[40] Donelson, W. C. Spatial management of information. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1978), ACM, pp. 203–209.

[41] Dou, W., Jeong, D. H., Stukes, F., Ribarsky, W., Lipford, H. R., and Chang, R. Recovering reasoning process from user interactions. *IEEE Computer Graphics and Applications Speicial Issue on Visual Analytics Evaluation* (2009). To Appear.

[42] Duchowski, A. T. *Eye Tracking Methodology: Theory and Practice*. Springer, January 2003.

[43] Dykes, J., MacEachren, A., and Kraak, M. *Exploring Geovisualization*. Elsevier, 2005.

[44] Eccles, R., Kapler, T., Harper, R., and Wright, W. Stories in GeoTime. pp. 19–26.

[45] Edsall, R. The parallel coordinate plot in action: Design and use for geographic visualization. In *Computational Statistics and Data Analysis* (2003), vol. 43, pp. 605–619.

[46] Eick, S. G. Visual discovery and analysis. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (2000), 44–58.

[47] Elmqvist, N., Dragicevic, P., and Fekete, J.-D. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (Nov.-Dec. 2008), 1539–1148.

[48] Ericson, M. Infovis keynote address: Visualizing data for the masses: Information graphics at the New York Times. *IEEE transactions on visualization and computer graphics 13* (2007).

[49] ESRI. ArcGIS. http://www.esri.com/products.html, 2007.

[50] Fink, G. A., North, C. L., Endert, A., and Rose, S. Visualizing cyber security: Usable workspaces. In *6th International Workshop on Visualization for Cyber Security (VizSec)* (2009). To Appear.

[51] Fotheringham, A. S., and Wong, D. W. S. The modifiable areal unit problem in multivariate statistical analysis. *Environment and Planning A 23*, 7 (1991), 1025–1044.

[52] Freeman, L. C. Visualizing social networks. *Journal of Social Structure 1* (2000).

[53] Fuhrmann, S., Ahonen-Rainio, P., Edsall, R., Fabrikant, S., Koua, E., Tobon, C., Ware, C., and Wilson, S. Making userful and usable geovisualization: Design and evaluation issues. In *Exploring Geovisualization* (2005).

[54] Furnas, G. W. Generalized fisheye views. vol. 17, ACM, pp. 16–23.

[55] Furnas, G. W. A fisheye follow-up: further reflections on focus + context. 999–1008.

[56] Gahegan, M., Takatsuka, M., Wheeler, M., and Hardisty, F. Introducing GeoVISTA Studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography. *Computers, Environment and Urban Systems 26*, 4 (2002), 267–292.

[57] Gallo, L. M., Airoldi, G. B., Airoldi, R. L., and Palla, S. Description of mandibular finite helical axis pathways in asymptomatic subjects. *Journal of Dental Research 72*, 2 (1997), 704713.

[58] Gapminder. http://www.gapminder.org.

[59] Garg, S., Nam, J., Ramakrishnan, I., and Mueller, K. Model-driven visual analytics. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (Oct. 2008), pp. 19–26.

[60] Garland, M., and Heckbert, P. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH 1997* (1997), ACM, pp. 209–216.

[61] Garland, M., and Zhou, Y. Quadric-based simplification in any dimension. In *ACM Transaction on Graphics, Vol. 24, No. 2* (2005), ACM, pp. 209–239.

[62] Garvin, A. *The American City: What Works and What Doesn't*. McGraw Hill, 2002.

[63] Ghoniem, M., Cambazard, H., Fekete, J.-D., and Jussien, N. Peeking in solver strategies using explanations visualization of dynamic graphs for constraint programming. In *SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization* (2005), ACM Press, pp. 27–36.

[64] Ghoniem, M., Fekete, J.-D., and Castagliola, P. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization 4*, 2 (2005), 114–135.

[65] Gilson, O., Silva, N., Grant, P. W., and Chen, M. From web data to visualization via ontology mapping. *Computer Graphics Forum 27*, 3 (May 2008), 959–966.

[66] Goodell, H., Chiang, C.-H., Kelleher, C., Baumann, A., and Grinstein, G. Collecting and harnessing rich session histories. pp. 117–123.

[67] Google. Google Earth. http://earth.google.com, 2005.

[68] Gotz, D., and Zhou, M. Characterizing users' visual analytic activity for insight provenance. *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (Oct. 2008), 123–130.

[69] GRASS. GRASS GIS. http://grass.itc.it, 2007.

[70] Green, T. M., Ribarsky, W., and Fisher, B. Visual analytics for complex concepts using a human cognition model. *Visual Analytics Science and Technology, 2008. VAST 2008. IEEE Symposium on* (November 2008), 91 – 98.

[71] Greitzer, F. Methodology, metrics and measures for testing and evaluation of intelligence analysis tools. PNWD-3550, Battelle-Pacific Northwest Division, Richland, WA, 2005.

[72] Groth, D. Information provenance and the knowledge rediscovery problem. pp. 345–351.

[73] Groth, D. P., and Streefkerk, K. Provenance and annotation for visual exploration systems. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1500–1510.

[74] Guo, D., Liao, K., and Morgan, M. Visualizing patterns in a global terrorism incident database. *Environment and Planning B: Planning and Design 34*, 1 (2007), 767–784.

[75] Haken, H., and Portugali, J. The face of the city is its information. In *Journal of Environmental Psychology* (2003), vol. 23, Elsevier, pp. 385–408.

[76] Hauser, H., Weiskopf, D., Ma, K.-L., van Wijk, J. J., and Kosara, R. Scivis, infovis bridging the community divide?! In *In IEEE Visualization (Panel Proceedings)* (2006), IEEE CS Press, p. 5255.

[77] Havre, S., Hetzler, E., Whitney, P., and Nowell, L. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (2002), 9–20.

[78] Heer, J., Mackinlay, J. D., Stolte, C., and Agrawala, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1189 – 1196.

[79] Heer, J., and Robertson, G. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1240–1247.

[80] Heer, J., Viégas, F. B., and Wattenberg, M. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 1029–1038.

[81] Henry, N., and Fekete, J.-D. MatrixExplorer: a dual-representation system to explore social networks. In *IEEE Transactions on Visualization and Computer Graphics* (2006), vol. 12, IEEE Computer Society, pp. 677–684.

[82] Hibbs, M. A., Dirksen, N. C., Li, K., and Troyanskaya, O. G. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics 6*, 115 (2005).

[83] Hilbert, D. M., and Redmiles, D. F. Extracting usability information from user interface events. *ACM Comput. Surv. 32*, 4 (2000), 384–421.

[84] Hochheiser, H., and Shneiderman, B. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization 3*, 1 (2004), 1–18.

[85] Hong, J., Jeong, D. H., Shaw, C. D., Ribarsky, W., Borodovsky, M., and Song, C. Gvis: A scalable visualization framework for genomic data. In *EuroVis 2005: Eurographics / IEEE VGTC Symposium on Visualization* (2005), pp. 191–198.

[86] Huang, W. Using eye tracking to investigate graph layout effects. pp. 97–100.

[87] Inselberg, A., and Dimsdale, B. Parallel coordinates: A tool for visualizing multidimensional geometry. In *IEEE Visualization* (1990), IEEE CS Press, pp. 361–378.

[88] Intergraph. GeoMedia. http://www.intergraph.com/geomedia/, 2007.

[89] Jan, S. L. V. S., Clapworthy, G. J., and Rooze, M. Visualization of combined motions in human joints. *IEEE Computer Graphics and Applications 18*, 6 (1998), 1014.

[90] Jang, J., Wonka, P., Ribarsky, W., and Shaw, C. Punctuated simplification of man-made objects. Tech. rep., Georgia Institute of Technology GVU Center GIT-GVU-05-04, February 2005.

[91] Jankun-Kelly, T., Ma, K.-L., and Gertz, M. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (March/April 2007), 357–369.

[92] Jeong, D. H., Dou, W., Stukes, F., Ribarsky, W., Lipford, H. R., and Chang, R. Evaluating the relationship between user interaction and financial visual analysis. *Visual Analytics Science and Technology, 2008. VAST 2008. IEEE Symposium on* (November 2008), 83 – 90.

[93] Jolliffe, I. T. *Principal Component Analysis*, second ed. Springer, 2002.

[94] Kada, M., and Luo, F. Generalisation of building ground plans using half-spaces. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2006), vol. 36.

[95] Kadmon, N., and Shlomi, E. A polyfocal projection for statistical surfaces. *Cartographic Journal, The 15* (1978), 36–41(6).

[96] Kapler, T., and Wright, W. Geo time information visualization. *Information Visualization 4*, 2 (2005), 136–146.

[97] Keefe, D. F., OBrien, T. M., Baier, D. B., Gatesy, S. M., Brainerd, E. L., and Laidlaw, D. H. Exploratory visualization of animal kinematics using instantaneous helical axes. *Computer Graphics Forum (EuroVis Special Issue) 27*, 3 (2008), 863870.

[98] Keim, D. A. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics March* (1996).

[99] Keim, D. A., Kriegel, H.-P., and Seidl, T. Visual feedback in querying large databases. 158–165.

[100] Kindlmann, G. Lack of reproducibility hinders visualization science. *IEEE Visualization Compendium, IEEE CS Press* (2006).

[101] Kirsch, D. Distributed cognition: A methodological note. *Pragmatics and cognition 14*, 2 (2006), 249262.

[102] Koua, E., MacEachren, A., and Kraak, M. Evaluating the usability of visualization methods in an exploratory geovisualization environment. In *International Journal of Geographical Information Science* (2006), vol. 20(4), pp. 425–448.

[103] Kreuseler, M., Nocke, T., and Schumann, H. A history mechanism for visual data mining. *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on* (0-0 2004), 49–56.

[104] LaFree, G., and Dugan, L. Introducing the global terrorism database. *Terrorism and Political Violence 19*, 2 (2007), 181–204.

[105] Lam, H. A framework of interaction costs in information visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 11491156.

[106] Lefebvre, H. *The Production of Space.* Blackwell, 1991.

[107] Liu, D., Yue, J., Wang, X., Raja, A., and Ribarsky, W. The role of blackboard-based reasoning and visual analytics in RESIN's predictive analysis. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on* (Dec. 2008), vol. 2, pp. 508–511.

[108] Liu, Z., Nersessian, N. J., and Stasko, J. T. Distributed cognition as a theoretical framework for information visualization. *IEEE TVCG 14*, 6 (2008), 1173–1180.

[109] Livingston, M., Rosenblum, L., Julier, S., Brown, D., Baillot, Y., Swan, J., Gabbard, J., and Hix, D. An augmented reality system for military operations in urban terrain. In *Proceedings of the Interservice / Industry Training, Simulation, and Education Conference* (2002), p. 89.

[110] Losasso, F., and Hoppe, H. Geometry clipmaps: Terrain rendering using nested regular grids. In *Proceedings of SIGGRAPH 2004* (2004), ACM, pp. 769–776.

[111] Luebke, D. A developer's survey of polygonal simplification algorithms. In *IEEE Computer Graphics and Applications (May/June)* (May 2001), pp. 24–35.

[112] Lynch, K. *The Image of the City.* The MIT Press, 1960.

[113] Mackinlay, J., Hanrahan, P., and Stolte, C. Show Me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1137–1144.

[114] Marai, G., Laidlaw, D., Demiralp, C., Andrews, S., Grimm, C., and Crisco, J. Estimating joint contact areas and ligament lengths from bone kinematics and surfaces. *Biomedical Engineering, IEEE Transactions on 51*, 5 (May 2004), 790–799.

[115] McGuffin, M. J., Tancau, L., and Balakrishnan, R. Using deformations for browsing volumetric data. In *IEEE Visualization* (2003), IEEE Computer Society, pp. 401–408.

[116] Mina, C. MapInfo SpatialWare: A spatial information server for RDBMS. In *24th International Conference on Very Large Data Bases* (1998), Morgan Kaufmann Publishers, p. 704.

[117] Mitchell, W. J. *City of Bits: Space, Place, and the Infobahn.* MIT Press, 1996.

[118] Müller, W., and Alexa, M. Visual component analysis. In *VisSym 2004* (2004), Eurographics Association, pp. 129–136.

[119] Müller, W., Nocke, T., and Schumann, H. Enhancing the visualization process with principal component analysis to support the exploration of trends. In *APVis '06* (2006), Australian Computer Society, Inc., pp. 121–130.

[120] National Consortium for the Study of Terrorism and Responses to Terrorism. Global terrorism database. http://www.start.umd.edu/data/gtd/, 2007.

[121] Nielsen, J. Evaluating the thinking-aloud technique for use by computer scientists. 69–82.

[122] North, C. A user interface for coordinating visualizations based on relational schemata: Snap-together visualization.

[123] North, C. Toward measuring visualization insight. *IEEE Computer Graphics and Applications 26*, 3 (2006), 6–9.

[124] North, C., and Shneiderman, B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. 128–135.

[125] North, C., and Shneiderman, B. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies 53*, 5 (2000), 715–739.

[126] Oeltze, S., Doleisch, H., Hauser, H., Muigg, P., and Preim., B. Interactive visual analysis of perfusion data. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 13921399.

[127] Perer, A.; Shneiderman, B. Balancing systematic and flexible exploration of social networks. *Transactions on Visualization and Computer Graphics 12*, 5 (Sept.-Oct. 2006), 693–700.

[128] Perlin, K., and Fox, D. Pad: an alternative approach to the computer interface. 57–64.

[129] Pike, W., Bruce, J., Baddeley, B., Best, D., Franklin, L., May, R., Rice, D., Riensche, R., and Younkin, K. The Scalable Reasoning System: Lightweight visualization for distributed analytics. pp. 131–138.

[130] Pike, W., May, R., and Turner, A. Supporting knowledge transfer through decomposable reasoning artifacts. In *40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007* (January 2007), pp. 204c–204c.

[131] Pike, W., Stasko, J., Chang, R., and O'Connell, T. Science of interaction. *Information Visualization* (2009). to appear.

[132] Piringer, H., Kosara, R., and Hauser, H. Interactive focus+context visualization with linked 2D/3D scatterplots. In *CMV 04: Proceedings of the Second International Conference on Coordinated & Multiple Views in Exploratory Visualization* (Washington, DC, USA, 2004), IEEE Computer Society, p. 4960.

[133] Plaisant, C. Information visualization and the challenge of universal usability. In *Exploring Geovisualization* (2005), Elsevier.

[134] Pousman, Z., Stasko, J., and Mateas, M. Casual information visualization: Depictions of data in everyday life. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1145–1152.

[135] Rao, R., and Card, S. K. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI* (1994), pp. 318–322.

[136] Rester, M., and Pohl, M. Methods for the evaluation of an interactive InfoVis tool supporting exploratory reasoning processes. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2006), pp. 1–6.

[137] Rhyne, T.-M., Tory, M., Munzner, T., Ward, M., Johnson, C., and Laidlaw., D. H. Information and scientific visualization: Separate but equal or happy together at last? In *IEEE Visualization (Panel Proceedings)* (2003), IEEE CS Press, p. 611614.

[138] Rinner, C. A geographic visualization approach to multi-criteria evaluation of urban quality of life. In *International Journal of Geographical Information Science* (2007), vol. 21, pp. 907–919.

[139] Roberts, J. C. State of the art: Coordinated & multiple views in exploratory visualization. In *CMV 07: Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization* (Washington, DC, USA, 2007), IEEE Computer Society, p. 6171.

[140] Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J. Effectiveness of animation in trend visualization. *IEEE TVCG 14*, 6 (Nov.-Dec. 2008), 1325–1332.

[141] Robinson, A. C., and Weaver, C. Re-visualization: Interactive visualization of the process of visual analysis. In *GIScience Workshop on Visual Analytics & Spatial Decision Support* (2006).

[142] Rowe, C., and Koetter, F. *Collage City*. MIT Press, 1978.

[143] Saraiya, P., North, C., and Duca, K. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics 11*, 4 (2005), 443–456.

[144] Saraiya, P., North, C., and Duca, K. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization 4*, 3 (2005), 191–205.

[145] Saraiya, P., North, C., Lam, V., and Duca, K. A. An insight-based longitudinal study of visual analytics. *IEEE TVCG 12*, 6 (2006), 1511–1522.

[146] SAS Institute, Inc. SAS/INSIGHT. http://sas.com/technologies/analytics/statistics/insight.

[147] Seo, J., and Shneiderman, B. Interactively exploring hierarchical clustering results. *Computer 35*, 7 (2002), 80–86.

[148] Seo, J., and Shneiderman, B. Knowledge discovery in high-dimensional data: Case studies and a user survey for the rank-by-feature framework. *IEEE TVCG 12*, 3 (2006), 311–322.

[149] Shalabi, S. M. Analysis of urban morphology for real time visualization of urban scenes. Master's thesis, Massachusetts Institute of Technology, May 1998.

[150] Shanbhag, P., Rheingans, P., and desJardins, M. Temporal visualization of planning polygons for efficient partitioning of geo-spatial data. In *Proceedings Information Visualization* (2005), IEEE CS Press, pp. 211–218.

[151] Shapiro, M. A. *Measuring Psychological Responses To Media Messages*. Lawrence Erlbaum Associates, 1994, ch. Think-Aloud and Thought-List Procedures in Investigating Mental Processes, pp. 1–14.

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

[153] Shenon, P. For Filipino new dealer, a hard road to recovery. New York Times, 1992.

[154] Shlens, J. A tutorial on principal component analysis. http://www.snl.salk.edu/ shlens/notes.html, 2005.

[155] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. in , page ,. In *VL 96: Proceedings of the 1996 IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), IEEE Computer Society, p. 336.

[156] Shrinivasan, Y. B., and van Wijk, J. J. Supporting the analytical reasoning process in information visualization. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 1237–1246.

[157] Siirtola, H., and Mäkinen, E. Constructing and reconstructing the reorderable matrix. *Information Visualization 4*, 1 (2005), 32–48.

[158] Silva, C., Freire, J., and Callahan, S. Provenance for visualizations: Reproducibility and beyond. *Computing in Science & Engineering 9*, 5 (Sept.-Oct. 2007), 82–89.

[159] Skog, T., Ljungblad, S., and Holmquist, L. E. Between aesthetics and utility: Designing ambient information visualizations. *Information Visualization, IEEE Symposium on 0* (2003), 30.

[160] Smit, F., van Liere, R., Beck, S., and Froehlich, B. An image warping architecture for VR: Low latency versus image quality. In *IEEE Virtual Reality 2009* (2009), p. 2734.

[161] Spotfire. Decision site for functional genomics. http://www.spotfire.com/, 2005.

[162] Stolte, C., and Hanrahan, P. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *Information Visualization, IEEE Symposium on 0* (2000), 5.

[163] Stolte, C., Tang, D., and Hanrahan, P. Multiscale visualization using data cubes. 7–14.

[164] Takatsuka, M., and Gahegan, M. GeoVISTA studio: a codeless visual programming environment for geoscientific data analysis and visualization. *Comput. Geosci. 28*, 10 (2002), 1131–1144.

[165] The GGobi Foundation, Inc. GGobi. http://www.ggobi.org.

[166] The MathWorks, Inc. MATLAB. http://www.mathworks.com/products/matlab.

[167] Thomas, J., and Cook, K., Eds. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE CS Press, 2005.

[168] Tobon, C. Evaluating geographic visualization tools and methods: An approach and experiment based upon user tasks. In *Exploring Geovisualization* (2005).

[169] Tory, M., Potts, S., and Moller, T. A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 7180.

[170] Toussaint, G. Anthropomorphic polygons. In *American Mathematical Monthly* (1991), pp. 31–35.

[171] Trafton, J. G., Kirschenbaum, S. S., Tsu, T. L., Miyamoto, R. T., Ballas, J. A., and Raymond, P. D. Turning pictures into numbers: extracting and generating information from complex visualizations. *Int. J. Hum.-Comput. Stud. 53*, 5 (2000), 827–850.

[172] Tufte, E. R. *Envisioning Information*. Graphics Press, 1990.

[173] University, B. X-ray reconstruction of moving morphology (xromm). http://xromm.org, 2009.

[174] U.S. Census Bureau. United States 2000 Census. http://www.census.gov, 2000.

[175] van Wijk, J. The value of visualization. *Visualization, 2005. VIS 05. IEEE* (Oct. 2005), 79–86.

[176] van Wijk, J. J., and van Selow, E. R. Cluster and calendar based visualization of time series data. 4–9.

[177] Viega, J., Conway, M. J., Williams, G., and Pausch, R. 3D magic lenses. 51–58.

[178] Wall, M. E., Rechtsteiner, A., and Rocha, L. M. Singular value decomposition and principal component analysis. In A Practical Approach to Microarray Data Analysis (D.P. Berrar, W. Dubitzky, M. Granzow, eds.) Kluwer: Norwell, MA, pp. 91-109., 2003.

[179] Wang, X., Jeong, D. H., Dou, W., Lee, S.-W., Ribarsky, W., and Chang, R. Defining and applying knowledge conversion processes to a visual analytics system. *Computers & Graphics* (2009). In Press.

[180] Wang, X., Miller, E., Smarick, K., Ribarsky, W., and Chang, R. Investigative visual analysis of global terrorism. In *Computer Graphics Forum* (2008), Computer Graphics Forum, pp. 919–926.

[181] Ward, M. O. XmdvTool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the conference on Visualization '94* (Los Alamitos, CA, USA, 1994), IEEE Computer Society Press, pp. 326–333.

[182] Weaver, C. Building highly-coordinated visualizations in Improvise. In *IEEE Symposium on Information Visualizaion* (2004), pp. 159–166.

[183] Weaver, C. Improvisational geovisualization of the 2000 United States Census. In *AutoCarto* (2006).

[184] Whiting, M. A., Cowley, W., Haack, J., Love, D., Tratz, S., Varley, C., and Wiessner, K. Threat stream data generator: creating the known unknowns for test and evaluation of visual analytics tools. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2006), pp. 1–3.

[185] Whiting, M. A., Haack, J., and Varley, C. Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software. In *BELIV '08: Proceedings of the 2008 CHI workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2008), pp. 74–81.

[186] Williamson, C., and Shneiderman, B. The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings Information Retrieval* (1992), ACM Press, pp. 338–346.

[187] Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V. Visualizing the non-visual: spatial analysis and interaction with information from text documents. 51.

[188] Wright, W., Schroh, D., Proulx, P., Skaburskis, A., and Cort, B. The sandbox for analysis: concepts and methods. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems* (New York, NY, USA, 2006), ACM, pp. 801–810.

[189] Xiao, L., Gerth, J., and Hanrahan, P. Enhancing visual analysis of network traffic using a knowledge representation. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On* (31 2006-Nov. 2 2006), pp. 107–114.

[190] Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1224–1231.

[191] Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. Understanding and characterizing insights: How do people gain insights using information visualization? In *BELIV '08: Proceedings of the 2008 CHI workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2008), pp. 39–44.

[192] Yi, J. S., Melton, R., Stasko, J., and Jacko, J. A. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization 4*, 4 (2005), 239–256.

[193] You, B., Siy, P., Anderst, W., and Tashman, S. In vivo measurement of 3D skeletal kinematics from sequences of biplane radiographs: Application to knee kinematics. *IEEE Transactions on Medical Imaging 20*, 6 (2001), 514525.

[194] Zanbaka, C. A., Ulinski, A. C., Goolkasian, P., and Hodges, L. F. Social responses to virtual humans: implications for future interface design. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 1561–1570.

[195] Zhu, Weizhong; Chen, C. Visual analysis of terrorism events extracted from the public knowledge bases. *Transactions on Visualization and Computer Graphics* (2007).