Supporting Asynchronous Collaboration for Interactive Visualization

Jeffrey Michael Heer

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-166
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-166.html

December 17, 2008

Supporting Asynchronous Collaboration for Interactive Visualization

by

Jeffrey Michael Heer

B.S. (University of California, Berkeley) 2001

M.S. (University of California, Berkeley) 2004

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Maneesh Agrawala, Chair
Doctor Stuart K. Card
Professor Marti A. Hearst
Professor Joseph M. Hellerstein

Fall 2008

The dissertation of Jeffrey Michael Heer is approved:

Chair	Date
	Date
	Date
	Date

University of California, Berkeley

Fall 2008

Supporting Asynchronous Collaboration for Interactive Visualization

by

Jeffrey Michael Heer

ABSTRACT

Supporting Asynchronous Collaboration for Interactive Visualization

by

Jeffrey Michael Heer

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Maneesh Agrawala, Chair

Interactive visualizations leverage human visual processing to increase the scale of information with which we can effectively work. However, most visualization research to date relies on a single-user model, overlooking the social nature of visual media. Visualizations are used not only to explore and analyze data, but to communicate findings. People may disagree on how to interpret data and contribute contextual knowledge. Furthermore, some data sets are so large that thorough exploration by a single person is unlikely. Such scenarios arise regularly in scientific collaboration, business intelligence, and public data consumption. This thesis recasts interactive visualizations as not just analytic tools, but social spaces supporting collective data analysis. To this aim, I introduce theoretical design considerations guiding the invention of social visual analysis tools and present the design, implementation, and evaluation of interactive systems based on these principles.

The first such system is *sense.us*, a web site supporting asynchronous collaboration across a variety of visualization types. The site supports view sharing, discussion, graphical annotation, and social navigation and includes novel interaction elements. User studies of the system reveal emergent patterns of social data analysis, cycles of

observation and hypothesis, and the complementary roles of social navigation and data-driven exploration.

Based on design considerations and lessons learned from sense.us, this dissertation also introduces new techniques to support collaborative interaction around visualizations. The *scented widgets* system embeds visualizations of social activity in common user interface controls to enhance collective information foraging. A *generalized selection* framework represents collaborative annotations as declarative queries over visualized data, enabling annotation of dynamic data across multiple visualization views. Interactive query relaxation enables users to further generalize selections along data dimensions of interest. New *graphical histories* for visualization support analysis and accelerate collaborative sharing of findings, and a framework for *animated transitions* better communicates the relationship between views in an analysis session. As evidenced in a series of evaluative studies, these components enable teams to collaborate more effectively as they conduct visual data analysis.

Professor Maneesh Agrawala

Dissertation Chair

For my parents,

Michael and Star Heer

Table of Contents

List of Figures

List of Tables

Acknowledgements

Though a thesis is ostensibly one's own, in truth this dissertation would never have been possible without the collaboration and support of others.

As a "feral" graduate student and serial intern, I was particularly lucky to benefit from a wide array of mentors and colleagues. Stu Card, Ed Chi, and Peter Pirolli at PARC first sparked and directed my love of research. At Berkeley, James Landay, Marti Hearst, Anind Dey, Jennifer Mankoff, Joe Hellerstein, Marc Davis, and Peter Lyman (may he rest in peace) shaped my thinking and helped me look at the world in new and fascinating ways. Jock Mackinlay, Chris Stolte, and colleagues at Tableau software showed me what real software engineering is all about, and that *"working smarter, not harder"* is an achievable proposition. George Robertson, Danyel Fisher, Desney Tan, and colleagues at Microsoft helped me further hone my research skills and widen my perspective. Martin Wattenberg and Fernanda Viégas at IBM demonstrated the value of creativity and joy in the research process and contributed much to my thinking in this thesis. Through it all, my advisor Maneesh Agrawala has been instrumental; as a teacher, supporter, sounding-board, challenger, colleague, and friend. Anything I have been able to achieve is in large part due to the influence of these wonderful mentors.

My dissertation research was generously supported first by the American Society for Engineering Education through a National Defense Science & Engineering Graduate Fellowship, and later by graduate fellowships from Microsoft and IBM. I am also indebted to the participants of the studies I have conducted and am grateful to the user communities of the *prefuse* and *flare* toolkits for their help and generosity.

I was fortunate to be surrounded by amazing graduate colleagues in my years at Berkeley. A particular acknowledgement is due to the cohort I began graduate school with, including Chris Beckmann, Guillermo Diez-Canas, Tara Matthews, and Alan Newberger. I doubt a greater set of office mates has ever been assembled. The subsequent denizens of the Berkeley Institute of Design (BiD) have also been a constant source of inspiration, insight, and needed diversion. I am thankful to them all, including Ryan Aipperspach, Andy Carle, Scott Carter, Ana Ramírez Chang, Brien Colwell, Omar Khan, Nicholas Kong, Scott Lederer, Dave Nguyen, Lora Oehlberg, Divya Ramachandran, Tye Rattenbury, Jingtao Wang, and Wesley Willett. Thanks also to the VisLab members—Rob Carroll, Floraine Grabler, David Jacobs, Kenrick Kin, and Jamie O'Shea—and the GUIR-heads of a bygone era—Richard Davis, Katherine Everitt, Jason Hong, Scott Klemmer, Yang Li, and Jimmy Lin. I am also grateful for the wonderful colleagues I've met at the School of Information, notably danah boyd, Andrew Fiore, Nathan Good, Elizabeth Goodman, Joe Hall, Dan Perkel, Christo Sims, and Yuri Takhteyev.

Finally, I dedicate this thesis to the family and friends who make life worth living. To the indefatigable Berkeley Crew and Seattle Hipsters for never letting me forget that hard work is best complemented by hard play. To Kara, to whom my academic acknowledgement is overdue. To Daniela, for her support, energy, and endless inquisitiveness. To my brother David, for his incomparable humor and the gift of vicarious rock stardom. To my grandfather Rupert, the living embodiment of perseverance, and to my grandmother Margie, whose departure still stings but whose embrace is ever-present. And to my parents Michael and Star, who taught me that the good life is founded upon love and that nothing is out of reach.

Thank you.

1 Introduction

"With a collaborative spirit, with a collaborative platform where people can upload data, explore data, compare solutions, discuss the results, build consensus, we can engage passionate people, local communities, media and this will raise—incredibly—the amount of people who can understand what is going on.

And this would have fantastic outcomes: the engagement of people, especially new generations; it would increase knowledge, unlock statistics, improve transparency and accountability of public policies, change culture, increase numeracy, and in the end, improve democracy and welfare."

—Enrico Giovannini, Chief Statistician, OECD. June 2007.

Visual representations of information often lead to new insights by enabling viewers to see data in context, observe patterns, and draw comparisons. In this way, visualizations leverage the human visual system to improve our ability to process large amounts of data. In their anthology *Using Vision To Think*, Card, Mackinlay, and Shneiderman [35] describe how visualization supports the process of sensemaking, in which information is collected, organized, and analyzed to form new knowledge and inform further action. They emphasize the ways visualization exploits an individual's visual perception to facilitate cognition.

In practice, however, sensemaking is also a social process. People may disagree on how to interpret the data and may contribute contextual knowledge that deepens understanding. As participants build consensus or make decisions they learn from their peers. Furthermore, some data sets are so large that thorough exploration by a single person is unlikely. This suggests that to support sensemaking, visualizations should also support social interaction. In this spirit, a recent report [171] names the design of collaborative analysis tools as a grand challenge for visualization research.

These considerations are not just hypothetical. In one instance, the manager of a business group described to us how quarterly reports are disseminated within his organization via e-mail. Heated discussion takes place around charts and graphs as the group debates the causes of sales trends and considers future actions. However, writing about particular trends or views is difficult, involving awkward references to attached spreadsheets from the e-mail text. Furthermore, the discussion is scattered and disconnected from the visualizations, making it difficult for newcomers to catch up or others to review and summarize the discussion thus far. According to the manager of the group, the analysis process could benefit from a system for sharing, annotating, and discussing the visualized data. Such scenarios regularly arise in business [89], intelligence analysis [143, 171], and public data consumption [60].

Experiences with deployments of visualizations hint at ways that social phenomena already occur around visualizations. For example, Wattenberg [186] describes the response to NameVoyager, an online visualization of historical baby name trends. Playful yet often surprisingly deep analysis appeared on numerous blogs as participants discussed their insights and hypotheses. Observing use of a physical installation of the Vizster social network visualization [86, 87], we noted that groups of users, spurred by storytelling of shared memories, spent more time exploring and asked deeper analysis questions than individuals. Similarly, Viégas et al. [179] found that users of the PostHistory e-mail archive visualization immediately wanted to share views with friends and family and engage in storytelling.

While suggestive, these observations provide only a circumstantial understanding of the social aspects of analysis with visualizations. In the case of the NameVoyager and PostHistory, the findings were essentially accidental. Vizster was designed for playful interaction, but in a synchronous and less analytic context. It would therefore be valuable to replicate these findings to deepen our understanding of this type of interaction. Furthermore, if social interaction is an important accompaniment to visualization, it is natural to look for ways to support and encourage it.

The social aspects of visualization have taken on new importance with the rise of the Web, enabling collaboration between participants acting in different geographic locations and at different times. One reason for this interest is that partitioning work across both time and space holds the potential of greater scalability in group-oriented analysis. For example, one decision making study found that asynchronous collaboration resulted in higher-quality outcomes—broader discussions, more complete reports, and longer solutions—than face-to-face collaboration [10]. However, this distributed, asynchronous style of collaboration introduces new challenges for visualization research.

1.1 Thesis Problem and Approach

This thesis focuses on the central problem of how to design visualization systems that support and catalyze social sensemaking by analysts and decision-makers collaborating asynchronously. In particular, we examine what forms of mediated social interaction might leverage the myriad skills and inclinations of a group to result in more effective analyses—i.e., those that generate more insights, cover a broader range of data, and bring multiple perspectives to bear. As we elaborate in subsequent chapters, social activities in analysis include coordinating work among participants, monitoring the activity and progress of others, interacting with and discussing shared artifacts, and disseminating findings. Distributed, asynchronous collaboration further complicates these tasks due to the lack of immediate monitoring and feedback that people naturally perform in physical settings.

To address these problems, we review prior work in visualization, computer-supported cooperative work, and social psychology to develop guidelines for the design of social data analysis systems. The considerations constitute hypotheses as to what mechanisms may facilitate social sensemaking with shared data visualizations. The considerations also highlight important sub-problems which we address in this thesis:

- *Fostering awareness* of other analysts' activity,
- *Referring to and annotating data* in a robust and reusable fashion, and
- *Disseminating findings* through view sharing, presentation, and story-telling.

We apply these considerations to the design and implementation of a range of visualization systems to support social data analysis. Each of these systems is evaluated through empirical user studies of people exploring, analyzing, or communicating data.

1.2 Thesis Contributions

This thesis contributes new principles and systems enabling collaborative data analysis with interactive visualizations. The contributions can be categorized into three areas:

I. Design considerations for collaborative visual analytics. We analyze various theoretical principles and design criteria faced by developers of collaborative visual analysis tools. The resulting considerations address the social, organizational, and interface design concerns requisite for supporting asynchronous collaboration around data visualizations.

II. The design and evaluation of novel collaborative visual analysis environments.

   a. We introduce *sense.us*, a web-based visualization system built in accordance with our design considerations that enables social data analysis of 150 years of United States census data. The site provides novel interaction elements for view sharing, discussion, graphical annotation, and social navigation.

   b. Based on user studies of *sense.us*, we develop an initial characterization of social data analysis patterns around visualizations. The studies reveal cycles of observation and hypothesis, social information foraging, and the complementary roles of social navigation and data-driven exploration.

III. Interaction techniques and interface components for visual data analysis. We develop and evaluate new extensions to a variety of analysis systems, each intended to address a sub-problem identified by our design considerations.

Though inspired by collaborative concerns, these techniques can also benefit analyses performed by individuals.

a. *Scented widgets* integrate embedded visualizations into user interface controls to improve navigation of information spaces. They improve collaborative analysis by enhancing awareness of social activity, helping analysts allocate attention to both popular and neglected data regions.

b. *Generalized selection* techniques enable “data-aware” forms of annotation that support selection of time-varying data and re-application of annotations across various visual encodings. Interactive query relaxation enables users to formulate more advanced selections by generalizing from simple, initial selections. These selection techniques improve social data analysis by providing data-centric annotation mechanisms that persist across many different visual representations of a data set.

c. *Graphical histories* of visual analysis trails support the analysis process and facilitate communication of analytic findings. These visual representations of analysis sessions assist collaboration and dissemination through the sharing of selected views and analysis stories.

d. *Animated transitions* between data graphics better communicate the relationship between sequentially presented views. We contribute the design and evaluation of animated transitions that enable viewers to more effectively stay oriented when sequentially shown related visualizations. These transitions can enhance engagement, presentation, and story-telling.

1.3 Thesis Outline

CHAPTER 2 begins by covering related work. The remainder of the thesis is organized into two areas: an investigation of principles and system designs for collaborative visual analysis, followed by a collection of targeted techniques to further enhance social data analysis and communication.

1.3.1 Principles and Systems for Collaborative Visual Analysis

CHAPTERS 3 and 4 introduce design guidelines and system designs for collaborative visual analysis tools, and present findings from system deployments. CHAPTER 3 synthesizes results from prior work in social psychology, organizational studies, peer-production, and computer-supported cooperative work to provide design considerations for the development of collaborative visual analysis tools. The considerations cover the division and allocation of work (§3.1); grounding (§3.2) and reference (§3.3); incentive structures (§3.4); markers of identity and reputation (§3.5); the effects of group size and diversity (§3.6); and the process of communicating findings and building consensus to inform decision-making (§3.7).

Guided by the design considerations, CHAPTER 4 introduces *sense.us*, a web site for collaborative visual analysis of 150 years of United States census data. It describes the design of novel mechanisms for supporting group foraging, view sharing, and discussion by a general audience (§4.1) and related implementation issues (§4.2). It then presents findings from both laboratory studies and live deployments of the sense.us system (§4.3) and discusses the implications (§4.4).

1.3.2 Interface Techniques supporting Social Data Analysis

CHAPTERS 5 through 8 describe novel interface techniques for supporting analysis suggested by the previous chapters. Though inspired by insights into social processes, these techniques provide benefits for individual users as well.

CHAPTER 5 introduces *scented widgets*: small visualizations embedded into user interface controls to assist users' navigation through information spaces. After discussing related work (§5.1), the chapter posits design concerns for embedded visualizations (§5.2) and discusses the system implementation (§5.3) and example applications (§5.4). It then presents the results of a controlled study using scented widgets to provide social navigation cues in the form of visualized activity metrics in sense.us (§5.5).

CHAPTER 6 describes *generalized selection* techniques to point to items or regions of interest in an interface. These techniques enable "data-aware" annotation, and support advanced queries by generalizing from an initial, simpler selection. The chapter starts by discussing issues in reference, selection, and interactive querying (§6.1) and then develops a system for generalized selection of time-varying information in a view-independent manner and applies it to both data visualization (§6.2) and graphics (§6.3) applications. After discussing the system implementation (§6.4), the chapter presents empirical results (§6.5) demonstrating that users make more accurate selections with our techniques.

CHAPTER 7 presents a design space analysis (§7.1) and system implementation (§7.2) of a *graphical history* system that captures and visualizes analysts' interaction histories in the Tableau visual analysis tool. The chapter introduces techniques for improving the scalability of history displays and enabling collaboration through the sharing and export of salient visualization states. It then validates our system design decisions through an analysis of collected user history logs (§7.3).

CHAPTER 8 focuses on *animated transitions*, which many users cited as a compelling aspect of social visualization systems such as sense.us and can be used to assist presentation by better communicating the relationship between related visualization views. After reviewing past uses of animation in user interfaces (§8.1), the chapter develops animation design guidelines (§8.2) and a system supporting animated transitions between data graphics (§8.3). It then describes two experiments evaluating the benefits of various transition types (§8.4) and discusses the implications for improving animation design (§8.5).

Finally, CHAPTER 9 summarizes the contributions of this thesis (§9.1), describes recent developments (§9.2), and outlines remaining challenges for improving user interfaces for social sensemaking (§9.3).

2 Related Work

A rich and varied body of prior work in both visualization and computer-supported cooperative work (CSCW) is pertinent to the design of collaborative visual analysis tools. This chapter reviews the background work framing this thesis. Related work specific to a particular contribution is introduced in later chapters.

2.1 Information Visualization

Since the introduction of data graphics in the late 1700's [172], visual representations of abstract information have been used to demystify data and reveal hidden patterns. The advent of graphical interfaces has enabled direct interaction with visualized data, giving rise to over a decade of information visualization research. Visualization research seeks to supplement human cognition by leveraging human visual capabilities to make sense of abstract information [35], providing means by which humans with constant perceptual abilities can grapple with increasing quantities of data.

The potential for using visual imagery to enhance cognition is strong. A large part of the human nervous system has evolved to process visual information; in the human brain, over 70% of the receptors and 40% of the cortex are implicated in vision processing [187]. Furthermore, there is empirical evidence that imagery can be an efficient means of communication; one psychological experiment found that visual imagery can be learned and remembered at twice the bandwidth of text [116]. As the information sources around us continue to increase in both number and output, more effective means of leveraging the relatively static cognitive capabilities of humans can help reduce information overload.

2.1.1 Graphical Perception

One of the most influential theorists of information visualization is Jacques Bertin, who prior to the computer revolution wrote a series of books providing a detailed examination of the use of graphic marks to aid human information processing [16, 17]. Though his prescriptions were based on his own judgment, he carefully and methodically provided a conceptual framework for analyzing visualizations, including the identification of different types of data (e.g., categories, numbers, maps, networks) and so-called retinal variables for visual encoding such as position, size, shape, color, and orientation. Though written prior to the invention of graphical user interfaces, Bertin understood the value of interaction: his discussion of visual permutation matrices [17] includes the description of a mechanical system for reorganizing rows and columns of a tabular visualization. Such foundational work has been furthered by incorporating results from perceptual psychology [140, 187], including gestalt grouping principles, pre-attentive visual phenomena, and models of color vision.

Numerous theorist-practitioners have continued to advance information visualization theory. John Tukey advocated the inclusion of exploratory data analysis in statistics [175], using graphical representations to explore data before conducting confirmatory analysis. William S. Cleveland and colleagues conducted a series of experiments comparing the effectiveness of different visual encodings for conveying disparate data types [50, 51, 52]. His findings provide a rank-ordering of visual mappings for different data types. For example, position and length are quite effective at conveying quantitative values, but area less so. Color, shape, and texture excel at conveying categorical attributes, but are not effective for communicating numerical data. Mackinlay [124] extended and formalized this knowledge into a computational model for the automatic generation of static data graphics. Edward Tufte is also an important contributor, prescribing a number of design guidelines [172, 173, 174] including the use of small multiples displays for multi-dimensional data, maximizing the ratio of data-ink to non-data-ink, and eliminating uninformative elements (or *chart junk*).

2.1.2 Interaction and Sensemaking

More recently, visualization has grown to encompass not only visual design principles, but software and interaction design. Early work focused on the use of interactivity, such as *brushing* techniques to select and highlight visualized data points [8]. Based on years of experience designing visualizations, Shneiderman [162] proposed a task by data type taxonomy for information visualization, providing a list of tasks that visualizations should support and a classification of the different data types subject to visualization. He identified the interactive tasks of getting an overview of a collection, zooming in on items of interest, filtering out uninteresting items, getting more details for items on demand, highlighting relationships between items, providing an interactive history, and extracting and exporting collections of data. He identified the data types of 1-, 2-, 3-, and n-dimensional data as well as time, trees, and networks. This characterization is incomplete, missing the explicit recognition of nominal, ordinal, and quantitative data types, all of which can constitute individual dimensions within a visualization. However, Shneiderman does take care to point out the explicitly 2- and 3- dimensional nature of some data sets, such as geographic data or molecular models, and the particularly important semantics of dates and time. In contrast, Ware [187] starts with a basic entity-relationship dichotomy of data, from which more complex data types can be constructed.

Card et al. [35] further frame Shneiderman's list of characteristic tasks by grounding information visualization in the larger process of sensemaking. Sensemaking is the cyclical process in which humans collect information, examine, organize, and categorize that information, isolate dimensions of interest, and use the results to solve problems, make decisions, take action, or communicate findings [35, 143, 146, 154].

Visualizations enhance the sensemaking cycle by aiding the search for information, facilitating the discovery of patterns, and providing means for evaluating various hypotheses. Russell et al. [154] describe the structure of observed sensemaking tasks, formulating a model of the phases of sensemaking and describing the cost structures of these different phases. They introduce the notion of a "learning loop complex", in which information is iteratively schematized, and then the schemas refined in response to problematic outliers ("residue") that do not fit in the current scheme. Other studies have found that the sensemaking model fits a number of other analytic tasks, including business intelligence [142] and intelligence analysis [143]. Qu and Furnas [146] studied an online sensemaking task and note various means by which sensemakers schematize collected information, including the social process of borrowing and adapting existing schemas used by others.

In much of the literature on sensemaking, however, the social nature of sensemaking is often downplayed. Even in cases where the sensemaking task was conducted collaboratively (e.g., by Russell et al. [154]), the analysis of the sensemaking process focuses on individual cognitive processes and information seeking behaviors. Given the importance of sensemaking for framing visualization, it is necessary to extend our understanding to encompass the social factors affecting the sensemaking cycle and apply that knowledge in the design of collaborative visualization systems.

2.2 Social Software and Collaborative Visualization

The study of how computer systems can enable collaboration is referred to as *computer-supported cooperative work*, or CSCW. Because collaboration occurs in a variety of situations, CSCW scholars often use a "time-space" matrix [106] to outline the conceptual landscape. The time dimension represents whether users interact at the same time or not (synchronously or asynchronously). For example, instant messaging is a largely synchronous communication medium, while email is asynchronous. The space dimension describes whether users are collocated or geographically distributed. Figure 2.1 illustrates this 2 X 2 matrix, annotated with a number of existing technologies and practices.

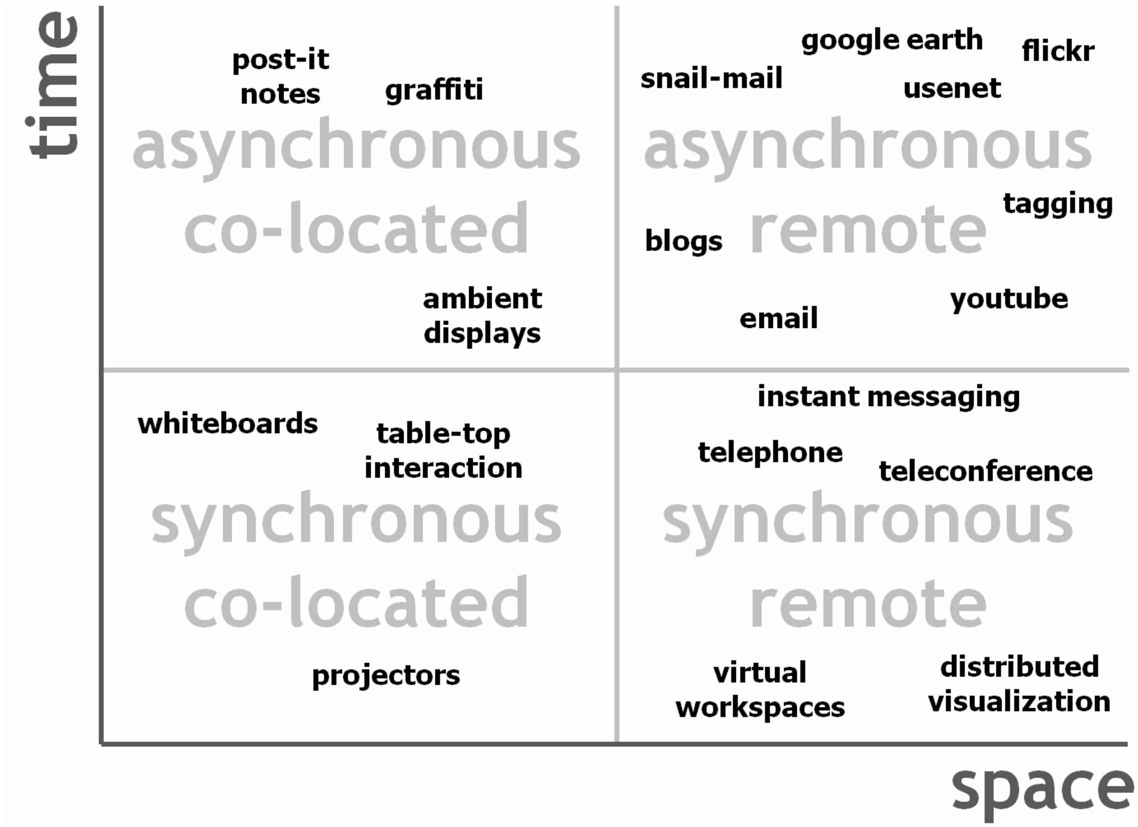


Figure 2.1. Space-time matrix for classifying collaborative applications. The two axes differentiate between same place (collocated) and different place (remote), and same time (synchronous) and different time (remote). The asynchronous-remote quadrant (top right) describes many current online services, but has received little research attention with respect to collaborative visual analysis.

2.2.1 *Collaborative and Distributed Visualization*

Most work on collaborative visualization has focused on synchronous scenarios: users interacting at the same time to analyze scientific results or discuss the state of a battlefield. Collocated collaboration usually involves shared displays, including wall-sized, table-top, or virtual reality displays (e.g., [58, 72]). Systems supporting remote collaboration have primarily focused on synchronous interaction [4, 24], such as shared virtual workspaces [1, 43] and augmented reality systems that enable multiple users to interact concurrently with visualized data [13, 44]. In addition, the availability of public displays has prompted researchers to experiment with asynchronous, collocated visualization (same place, different time); for example in the form of ambient displays that share activity information about collocated users [38].

Some researchers, particularly in scientific visualization, have used the web as a platform for visualization. Lefer and Pierson [119] developed a web architecture that leverages a network of workstations to provide visualization images on demand. Jankun-Kelly et al. [105] additionally enable interaction over the web, using the web browser as an exploratory interface to server-side visualization technology. Rhyne [148] describes a design concept for providing web-based visualizations of data from the United States Environmental Protection Agency, presaging a number of recent developments. Although these systems provide access to visualizations through the web, they remain grounded in a single-user model of usage. To support social sensemaking around visualizations, designers must forge a deeper connection between visualization and collaborative technologies.

Crucially, most prior work in collaborative visualization has focused on technical mechanisms for synchronizing user actions, such as enabling distributed users to adjust the parameters of a shared visualization in real-time. Shared control provides a valuable foundation for collaborative work, but does address the higher-level process of analytic sensemaking taking place with and around the visualizations. In contrast, this thesis develops new systems, interface mechanisms, and design guidelines that not only allow people to collaboratively view visualizations, but attempt to support their engagement in social sensemaking and data analysis.

2.2.2 Asynchronous Collaboration with Visualizations

In this thesis, we focus on remote asynchronous collaboration—the kind of collaboration that is most common over the Web. Though web-based social software is currently of great interest to businesses and researchers, as noted by Viégas and Wattenberg [180], little research attention has been dedicated to asynchronous collaboration with interactive visualizations. Instead, users often rely on static imagery when communicating about these interactive systems. Images of the visualization are transferred as printouts or screenshots, or included in word-processing or presentation documents.

At the time we began this thesis research, only a few research projects had looked at supporting asynchronous collaboration with visualizations. One such project is Collaborative Annotations on Visualizations [66], which enables users to attach graphical, audio, and text annotations to frames of a visualization movie. However, the film metaphor used by the system is aimed more at presentation than interactive exploration and the system provides scant support for extended discussions or social navigation. Brennan et al. [22] present their first steps towards a collaborative framework for visual analysts, describing the need for pointing behaviors to direct attention and enabling users to develop private visualizations and later fuse and compare these constructed views. Keel [109] examines how an analyst's spatial and temporal organization of data might be automatically mined to suggest organizations to other analysts, a process akin to collaborative filtering that he terms "indirect collaboration." None of these projects, however, provide an end-to-end system for supporting socially situated visual analysis.

A few commercial visualization systems provide asynchronous collaboration features. Online mapping systems such as Google Maps, Yahoo! Maps, and Google Earth provide bookmarks (i.e., URLs) that can be shared among users. The visualization company Spotfire provides DecisionSite Posters [164], a web-based system that allows a user to post an interactive visualization view that other users can explore and comment on. The Posters apply only to a subset of Spotfire's full functionality and do not allow graphical annotations, limiting their adoption [180].

One common feature of these systems is *application bookmarks*, that is, URLs or URL-like objects that point back into a particular state of the application: e.g., a location and zoom level in the case of Google Maps. This pattern is not surprising; for users to collaborate, they must be able to share what they are seeing to establish a common ground for conversation [49].

One of the primary uses of bookmarks is in discussion forums surrounding a visualization. Some systems use an *independent discussion* model, where conversations are decoupled from the visualization. For example, Google Earth provides threaded discussion forums with messages that include bookmarks into the visualized globe. In such systems there are unidirectional links from the discussion to the visualization, but no way to discover related comments while navigating the visualization itself.

A stream of related work comes from wholly or partly visual annotation systems, such as the image annotations in flickr or the anchored conversations of Churchill et al. [45]. Such systems enable *embedded discussion* that places conversational markers directly within a visualization or document. Discussion of a specific item may be accessed through a linked annotation shown within the visualization itself. These systems may be seen as the converse of independent discussions, allowing unidirectional links from the space of work into a conversation.

Other general mechanisms for supporting asynchronous collaboration around a shared artifact are relevant to visual analysis. Both textual and graphical annotation of a view or collection of data elements could aid discussion and analysis [28, 32, 180]. Annotation in the form of labeling or "tagging" [76, 130] could be applied to provide categorization and additional retrieval cues. Also, notification mechanisms [29] have proven valuable to support workgroup awareness [61] and manage the time-distributed nature of asynchronous work.

3 Design Considerations for Collaborative Visual Analytics

One challenge to achieving the benefits of asynchronous collaborative analysis is to determine the appropriate design decisions and technical mechanisms to enable effective collaboration around visual media. To create effective collaborative visual analysis environments, one must address a number of design questions. How should collaboration be structured, and what shared artifacts can be used to coordinate contributions? What are the most effective communication mechanisms?

In this chapter we develop design considerations to guide the development of collaborative visual analysis systems. We wish to better support collaborative analysis by grounding design decisions in both practical and theoretical knowledge of social interaction. A theoretically-grounded design framework can be applied to contrast existing systems and guide the future research and development of social visual analysis systems. Towards this aim, we review research in analytics, social psychology, peer-production, organizational studies, and computer-supported cooperative work to identify a set of design considerations to inform the development of asynchronous collaborative visual analysis systems.

The goal of this chapter is to identify key issues to guide work in collaborative visual analytics. We group our design considerations into seven inter-related areas:

- Division and allocation of work (§3.1)
- Common ground and awareness (§3.2)
- Reference and deixis (§3.3)

Previously published by the author and Maneesh Agrawala in [91] and [93].

- Incentives and engagement (§3.4)
- Identity, trust, and reputation (§3.5)
- Group dynamics (§3.6)
- Consensus and decision making (§3.7)

For each of these areas we discuss the aspects underlying effective collaboration and suggest specific mechanisms by which they can be achieved. In the subsequent sections we discuss the results and recommendations from prior work and synthesize design considerations for social data analysis. We use these considerations to guide the design of the visual analysis systems presented in later chapters.

3.1 Division and Allocation of Work

A fundamental aspect of successful collaboration is an effective division of labor among participants. Collaboration involves both the segmentation of effort into proper units of work and the allocation of individuals to tasks in a manner that best matches their skills and disposition. Primary concerns are how to split work among multiple participants and meaningfully aggregate the results.

Benkler [12] describes the role of modularity, granularity, and cost of integration in the peer production of information goods, drawing on examples such as online discussions, open source software, and Wikipedia. Modularity refers to how work is segmented into atomic units, parallelizing work into independent tasks. The granularity of a module is a measure of the cost or effort involved in performing the task. The optimal granularity of modules is closely tied to the incentives for performing the work. For example, in online scenarios where the incentives tend to be small and non-monetary, a small granularity helps facilitate participation, encouraging people to participate in part due to the ease of contributing. A variety of granularities enables different classes of contribution to emerge.

The third aspect of Benkler's model is the cost of integration—the effort required to usefully synthesize the contributions of each individual module. Collaborative work will only be effective if the cost of integration is low enough to warrant the overhead of modularization while enforcing adequate quality control. There are a number of approaches to handling integration: automation (automatically integrating work through technological means), peer production (casting integration as an additional collaborative task given to trusted participants), social norms (using social pressures to reduce vandalistic behavior), and hierarchical control (exercising explicit moderation).

Collaborative visual analytics can similarly be viewed as a process of peer production of information goods. Such goods may include the observations, questions, and hypotheses generated in the analysis process as well as tours or presentations of analysis results. Questions for collaborative visualization include how to facilitate the modularization of work. The first step is determining the units (modules) of contribution and their granularity. Existing frameworks for aiding this task include structural models of visualization design and sensemaking processes. Once modules have been identified, designers can attempt to reduce the cost structure for these tasks. Another important concern is the prescription of particular task types or roles—which aspects should be formally inscribed in the system and which should be left open to negotiation and definition by work groups themselves?

These observations imply the following design considerations:

- ***Modularity and Granularity***: *Identify appropriately-scoped units of work that form basic analytic contributions.*
- ***Cost of Integration***: *Synthesize work in a manner that lowers integration costs and improves scalability while maintaining quality.*

3.1.1 The Information Visualization Reference Model

One model for identifying modules of contribution is the information visualization reference model [35, 88], a general pattern for describing visualization applications (Figure 3.1). The model decomposes the visualization process into data acquisition and representation, visual encoding of data, and display and interaction. Each phase of this model provides an entry point for collaborative activity. Contributions involving data include uploading data, cleaning or reformatting data, moderating contributed data (e.g., to safeguard copyright or privacy concerns), and affixing metadata (e.g., providing keyword tags). Additional contributions of varying granularity lie in the application of visual encodings. Examples include matching data sets with existing visualization components, editing visual mappings to form more effective visualizations, and authoring visualization software components. Important issues for future work include the accuracy and provenance of contributed data sets. The primary focus of this thesis, however, is at the level of interaction, where we consider how collaborative visual analysis and exploration can be conducted most effectively.

3.1.2 The Sensemaking Model

To better understand analytic contributions, we consult the sensemaking model [35, 143,154], which grounds the use of information visualization in a theory of how people search for, organize, and create new knowledge from source information. Social issues accrue at each phase of the model: how do people communicate, how do they judge the contributions of others, how are groups formed, and what motivates contributions? We touch on each of these issues in subsequent sections. As indicated by the numerous interconnections in Figure 3.2, the sensemaking process has a much higher degree of coupling than the information visualization reference model, carrying implications for the granularity and integration of contributions.

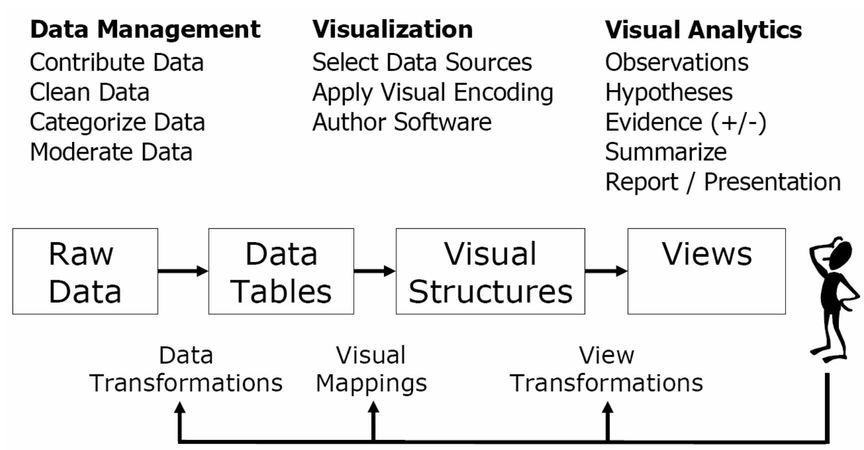


Figure 3.1: The information visualization reference model. Source data is mapped into tables that are visually encoded and presented in interactive views. Collaboration may occur at the level of data management, visualization, or analysis. Image adapted from Card et al. [35].

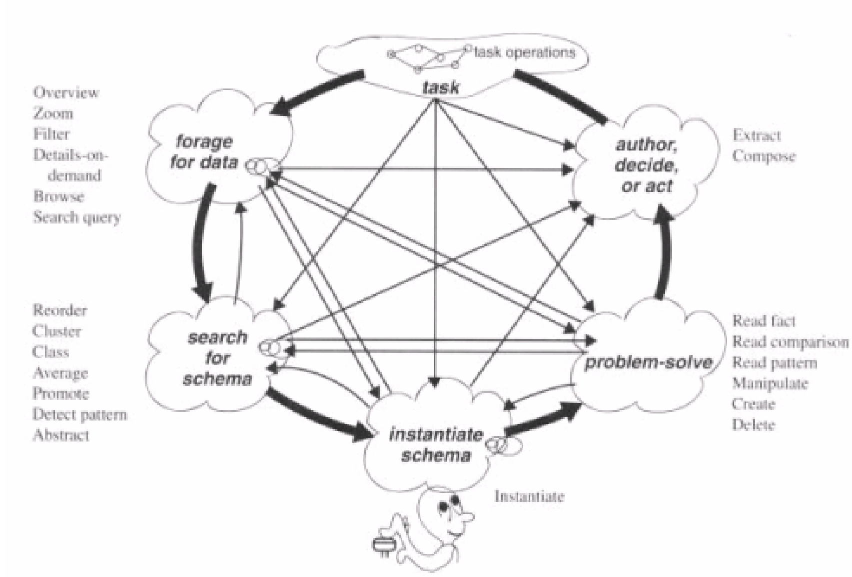


Figure 3.2. The sensemaking cycle. The diagram depicts the phases and loops of the sensemaking process, annotated with tasks. Image taken from Card et al. [35].

Intelligence analysis provides examples of both cooperative and competitive models of work [171]. In cooperative scenarios, modules may be of fine granularity and pooled such that collaborators can immediately make use of others' work. Examples include finding relevant information sources, identifying connections between sources, and positing hypotheses. Such work may involve tightly coupled collaboration, requiring awareness and communication among participants. In competitive scenarios, work is not integrated until a later stage of sensemaking, such as detailed, evidence-backed hypotheses or recommended actions. While lacking the benefits of resource pooling, this approach encourages individual assessment and can reduce groupthink bias. Accordingly, it may benefit collaborative visualization systems to support both fine-grained and coarse-grained work parallelization. This observation dove-tails with Grudin's [80] guideline that collaborative software features should not impede usage by a single-user usage, suggesting selective visibility as a design consideration.

- ***Selective Visibility***: *Analysts should be able to attend to or ignore social activity as desired.*

If adopting a competitive model, a main concern is to integrate the end results of the sensemaking process. How can analytic conclusions or suggested actions be presented, compared, and evaluated? If cooperative models are used, either across collaborators or within teams, we should consider social issues effecting each phase of sensemaking.

Information Foraging

The first phase of sensemaking is information foraging [142]. Given the underlying metaphor of foraging for food, an activity often performed by social packs of animals, social information foraging [141] seems a natural extension. Technologies for collaborative foraging could help pool findings, such as discovery of relevant information, and support notification updates. Design challenges include how to structure and categorize shared findings, such as identified trends or outliers, and provide task-sensitive retrieval mechanisms by which others can access them. Furthermore, systems could make the foraging behaviors of others visible by analyzing and displaying activity traces, facilitating social navigation [62] of data sets. Visualizing aggregate foraging behaviors is metaphorically similar to the scent trails left by ants foraging for food. In this form, general usage can be treated as an implicit collaborative contribution, a possibility discussed further in section 3.2.

Information Schematization

The next phases of sensemaking concern the construction and population of information schemata that organize findings from the foraging process. Schematization could be conducted in a collaborative fashion by enabling information organization and discussion amongst collaborators. One challenge is to synthesize the contributions of various collaborators in a manner that reduces the cost of integration, resulting in accessible forms such as summaries of arguments and evidence. To this aim, shared external representations [201] for manipulating the information can help structure asynchronous collaboration. For example, discussion forums aggregate contributions through the accretion of comments and replies in a sequential fashion, where as wikis (e.g., Wikipedia) and open source software rely on human editing backed by a revision management system to integrate and moderate contributions. Alternatively, systems with highly structured input such as NASA ClickWorkers [12, 135] or von Ahn's "games with a purpose" [183] rely on statistical aggregation. Clearly, the form of the collaborative artifact strongly affects the cost of integration; it may be more costly to find information in a sprawling discussion forum than in a group-edited document or statistical summary. As the number of collaborators or the complexity of contributions increases, the need for mechanisms facilitating aggregation becomes more acute. Future research is needed to devise and evaluate new external representations for structuring collaborative visual analytics.

Some existing research suggests mechanisms for representing and integrating analytic contributions, primarily focused on tasks of argumentation. The analyst's sandbox [195] provides a visual environment for spatially organizing hypotheses and positive and negative evidence. Coupled with revision management, the analyst's sandbox might also serve as a shared editing environment for collaborative analysis. Brennan et al. [22] introduce a tool for comparing and integrating the work of independent analysts. Their system uses a logic programming approach to merge network diagrams of collected evidence. Billman et al.'s CACHE [18] system supports the analysis of competing hypotheses; each analyst maintains a matrix of hypotheses and evidence and provides numerical measures of the reliability of evidence and assessments of the degree to which evidence confirms or disconfirms the hypotheses. The CACHE system statistically aggregates these ratings to form a group assessment. Argumentation systems such as Zeno [79] allow users to structure an argument into claims, constraints, and evidence. Similar to CACHE, Zeno can then automatically evaluate the current level of support for the provided claims. While each of these systems suggest possible approaches to structuring the creation of information schemata, further investigation is needed to compare (and potentially hybridize) these approaches. Usability and expertise are also important concerns; techniques that work well for professional analysts may not be appropriate for supporting collaborative visualization for a general audience on the Web.

- ***Shared Artifacts****: Structure social sensemaking through shared, editable representations.*

Problem-Solving, Decision-Making, and Action

The final phases of sensemaking involve problem-solving and action. These phases may or may not take place within the collaborative analysis environment. Furthermore, the analysts themselves may not be decision makers, thus mechanisms for presenting and disseminating analytic findings to others are often crucial components of collaborative work. Findings gained from analysis may serve as input to collaboration in other media, suggesting the need to both facilitate external access to the contents of the visual analysis environment and extracting content for use in other systems. If collaborators conduct problem-solving and decision making within the system, aforementioned issues regarding communication, discussion, and consensus must be addressed. We discuss these issues further in section 3.7.

3.2 Common Ground and Awareness

Inspired by linguistics, social psychologists have investigated fundamental prerequisites for successful communication. Clark and Brennan describe the concept of common ground [49], the shared understanding between conversational participants enabling communication. Through shared experience and discussion, people constantly monitor their mutual understanding. For example, facial expressions, body language, and backchannel utterances such as "uh-huh" and "hmm?" provide grounding cues of a participant's current level of understanding. Participants use both positive evidence of convergence of understanding and negative evidence of misunderstanding to establish common ground.

Surprisingly, an imperfect shared understanding is often sufficient. The principle of least collaborative effort states that conversational participants will exert just enough effort to achieve successful communication [47]. Collaborative effort may be applied during both a planning stage, in which a participant formulates their next utterance, and an acceptance stage, in which a participant ascertains if partners have understood the utterance. This principle serves as an evaluation guide for collaboration mechanisms, as different mechanisms may effect the amount of effort needed for collaborators to effectively communicate. For example, multiple studies have shown that the media of communication affects the cost structure of collaborative effort [22, 73]: views of a shared visual environment minimize the need to verbally confirm actions that can be assessed visually. However, media effects such as latency can hamper the efficiency benefits of such cues [73].

At both general and detailed levels, grounding theory provides a useful guide for design decisions. When collaborating with visualizations, participants can more easily ground each others' actions and comments when able to see a shared visual environment [22, 73]. Thus, visualization tools should provide mechanisms for bookmarking or sharing specific states of a visualization. Collaborators must be able to share views to specific visualization states both within the visualization environment itself and across other media. For example, the results of visual analysis might be shared more effectively as part of a web page or blog, where a dedicated readership and familiarity with collaborators better establishes the necessary common ground with respect to the subject matter. At minimum, the ability to easily pass around pointers (e.g., URLs) to specific views is indispensable, and therefore collaborative visualizations should be able to explicitly represent and export their internal state space [180, 186].

- ***View Sharing***: *Enable sharing of views across media with bookmarks (e.g., URLs).*
- ***Content Export***: *Support embedding of views in external media (e.g., email, blogs, reports).*

3.2.1 Discussion Models

Given the ability to access a shared viewpoint, one must still determine the forms of discussion and annotation around that view. For example, one could use visualization bookmarks within a standard discussion forum, interspersing links to desired views within the text. This form of independent discussion is unidirectional, linking from text to the visualization. Most existing systems, including Decision Site Posters [164], Many Eyes [181], and Swivel [169], provide support for independent, unthreaded comments. Another approach is embedded discussion, placing conversational markers directly within the visualization, such as comments over annotated geographic regions in Wikimapia [190]. This approach provides unidirectional links that point from the visualization to text.

Grounding might be further facilitated by tying discussion to the visualization state space. A "doubly-linked" comment might link to specific views as in independent discussion, while also enabling all such discussions to be retrieved in situ as visualization views are visited. Our hypothesis is that directly associating commentary with specific states of the visualization will facilitate grounding by disambiguating the context of discussion, while also enabling serendipitous discovery of relevant discussion during exploration. Evidence for this hypothesis could take the form of simplified referential utterances or facilitation of reader comprehension. We explore this form of commentary in CHAPTER 4.

- ***Discussion**: Support commentary; consider implications of discussion model on grounding.*

3.2.2 Activity Awareness

Another important source of grounding comes from awareness of others' activities, allowing collaborators to gauge what work has been done and where to allocate effort next [37, 61]. Within asynchronous contexts, participants require awareness of the timing and content of past actions. The need for coordination suggests that designs should include both history and notification mechanisms (e.g., [29]) for following actions performed on a given artifact or by specific individuals or groups. Browseable histories of past action are one viable mechanism, as are subscription and notification technologies such as RSS (Really Simple Syndication) and Atom.

- ***Artifact Histories**: Provide histories of actions performed on artifacts.*
- ***Notification**: Support notification subscriptions for views, artifacts, people, and groups.*

3.2.3 Social Navigation

User activity can also be aggregated and abstracted to provide additional forms of awareness. Social navigation [62] involves the use of activity traces to provide navigation cues based on the behavior of others, allowing users to purposefully navigate to past states of high interest or explore less-visited regions (termed "anti-social navigation" by Wattenberg and Kriss [186]). Navigation cues may be added to links to views with low visitation rates or to action items such as unanswered questions and unassessed hypotheses. Our Scented Widgets system in CHAPTER 5 explores the use of augmenting interface navigation controls to provide such activity cues.

- ***Action Flags***: *Mark needed future actions: unanswered questions, need for evidence, etc.*
- ***Social Navigation***: *Make social activity visible, indicate popular and neglected data regions.*

3.3 Reference and Deixis

A vital aspect of grounding is successfully referring to artifacts, people, places, or other items. As both Clark [46] and Brennan [22] explain, reference can take on many different forms. We focus on reference in spatial contexts. When collaborating around visual media, it is common to refer to specific objects, groups, or regions visible to participants. Such references may be general ("north by northwest"), definite (named entities), detailed (described by attributes, such as the "blue ball"), or deictic (pointing to an object and saying "that one", also referred to as indexical reference). Once the referent has been successfully established and grounding has been achieved between participants, collaboration can move forward.

Clark [46] surveys various forms of spatial indexical reference, grouping them into the categories of pointing and placing. Pointing behaviors use some form of vectorial reference to direct attention to an object, group, or region of interest, such as pointing a finger or directing one's gaze. Placing behaviors involve moving an object to a region of space that has a shared, conventional meaning. Examples include placing groceries on a counter to indicate items for purchase and standing across from the teller to indicate that you will be the purchaser. In addition to directing attention, indexical reference allows patterns of speech and text to change. Participants can apply the principle of least collaborative effort and use deictic terms like "that" and "there" to invoke indexical referents, simplifying the production of utterances.

Hill and Hollan [99] further discuss the various roles that deictic pointing gestures can play, often communicating intents more complicated than simply "look here". They describe how different hand gestures can communicate angle (oriented flat hand), height (horizontal flat hand), intervals (thumb and index finger in "C" shape), groupings (lasso'ing a region), and forces (accelerating fist). While other forms of reference are often achieved through speech or written text, deictic reference in particular offers important interface design challenges for collaborative visualization. Our hypothesis is that methods for performing nuanced pointing behaviors can improve collaboration by favorably altering its cost structure. Hill and Hollan make this claim explicitly, arguing for "generally applicable techniques that realize complex pointing intentions" by engaging "pre-attentive vision in the service of cognitive tasks".

- ***Pointing****: Support nuanced pointing through selection techniques and visual effects.*

3.3.1 Graphical Annotation

Freeform graphical annotations can provide an expressive form of pointing. Drawing a circle around a cluster of items or pointing an arrow at a peak in a graph can direct the attention of remote viewers. The angle of the arrow or shape of the hand-drawn circle may communicate emotional cues or add emphasis. Although such drawing and vector graphic annotations allow a high degree of expression, without any explicit tie to the underlying data they may only apply to a particular view in the visualization. Freeform annotations can persist over purely visual transformations such as panning and zooming, but if they are not data-aware they may become meaningless in the face of data-oriented operations such as filtering or drill-down.

3.3.2 Brushing and Dynamic Queries

A standard way to point in a visualization is through brushing [8, 127] and dynamic queries [2]: selecting and highlighting a subset of the data through direct manipulation of the display or auxiliary query controls. Naturally, these selections should be sharable as part of the state of the visualization. In addition, a palette of visual effects richer than simple filtering and highlighting can let users communicate different intents. For example, a user selecting a range of values in a chart might have one of any number of intents. If the user is interested in the specific points selected, those points should be prominently highlighted. However, if the user is primarily interested in the range of the contained values, the range interval should be given visual prominence.

Brushing-based forms of pointing have the advantage that the pointing action is tied directly to the data, whether modeled as a vector of selected tuples, a declarative query, or both. "Data-aware" representations allow a pointing intention to be reapplied in different views of the same data, enabling reuse of references across different choices of visual encodings. Data-aware annotations could also enable users to search for all commentary or visualizations that reference a particular data item. As data-aware annotations are machine-readable, they might also be used to export subsets of data and help steer automated data mining (e.g., [197]). Finally, machine-readable selections might be used as input for achieving more generalized forms of reference. For example, one might point to a particular object, but formulate a broader selection by abstracting from the properties of that object (e.g., "select all items blue like this one"). In this way, other forms of reference might be achieved in both human and machine readable form. We develop such techniques through a generalized selection framework for collaborative annotation in CHAPTER 6.

3.3.3 Ambiguity of Reference

An additional concern is ambiguity of reference. Clark et al. [48] demonstrate how people's common ground can affect ambiguity resolution: two people with greater familiarity might successfully communicate using ambiguous references, while a third participant remains confused. Asynchronous collaboration may be more susceptible to ambiguities than synchronous collaboration because participants don't receive immediate feedback or grounding cues from other collaborators. As a result, designers of pointing interactions must also consider the ease with which pointing actions can be interpreted unambiguously. The implicit interplay between gesture and text is often fluidly performed and subconsciously interpreted in synchronous interactions. As a result, systems may need to link text and references more concretely in asynchronous settings. For example, a text comment involving multiple deictic terms may need to link those terms explicitly to visual annotations, as the gestural cues used in face-to-face communication are not available for disambiguation.

3.4 Incentives and Engagement

If collaborators are professionals working within a particular context (e.g., financial analysts or research scientists) there may be existing incentives, both financial and professional, for conducting collaborative work. In a public goods scenario, incentives such as social visibility or sense of contribution may be motivating factors. Incorporating incentives into the design process may increase the level of contribution, and could provide additional motivation in those situations that already have established incentive systems.

Benkler [12] posits an incentive structure for collaboration consisting of monetary incentives, hedonic incentives, and social-psychological incentives. Monetary incentives refer to material compensation such as a salary or cash reward. Hedonic incentives refer to well-being or fun experienced intrinsically in the work. Social-psychological incentives involve perceived benefits such as increased status or social capital.

3.4.1 Personal Relevance

A number of observations of social use of visualization have noted that visualization users are attracted to data which they find personally relevant [87, 180, 186]. For example, in collaborative visual analysis of the occupations of American workers (CHAPTER 4), people often start by searching for their own profession and those of their friends and family, similar to the way people search for names in the popular NameVoyager visualization [186]. The hypothesis is that by selecting data sets or designing the presentation such that the data is personally relevant, usage rates will rise due to increased hedonic incentive. For example, geographic visualizations may facilitate navigation to personally relevant locations through typing in zip codes or city names, while a visualization of the United States' budget might communicate how a specific user's taxes were allocated rather than only listing total dollar amounts.

- ***Personal Relevance:*** *Increase engagement by increasing the personal relevance of the data.*

3.4.2 Social-Psychological Incentives

In the case of social-psychological incentives, designers can manipulate the visibility of contributions for social effects. Ling et al. [120] found that users contributed more if reminded of the uniqueness of their contribution or if given specific challenges. Ling et al. also found that participants contributed more when given group goals rather than individual goals, a finding at odds with existing social-psychological theory. Cheshire [40] ran a controlled experiment finding that, even in small doses, positive social feedback greatly increases contributions. He also found that visibility of high levels of cooperative behavior across the community increases contributions in the short term, but has only moderate impact in the long term. These studies suggest that social-psychological incentives can increase contribution rates, but such increases depend on the forms of social visibility. One incentive for visual analysis may be to display new discoveries or responses to open questions. Feedback mechanisms such as voting for interesting comments might also foster more contributions.

- ***Social-Psychological Incentives***: *Facilitate positive feedback and visibility of contributions.*

3.4.3 Game Play

Game dynamics can also be used to increase incentives. For example, von Ahn's "games with a purpose" [183] reframe otherwise tedious data entry tasks as actions within online games, successfully leveraging game dynamics to engage users in the construction of information goods. Elsewhere, I discuss various examples in which playful activity contributes to visual analysis [87], applying insights from an existing theory of playful behavior [31] that analyzes the competitive, visceral, and teamwork building aspects of play. For example, scoring mechanisms create competitive social-psychological incentives. Game design might also be used to allocate attention. For example creating a team-oriented "scavenger hunt" analysis game could focus participants on a particular subject matter. Salen and Zimmerman [155] provide a resource for the further study of game design concepts.

- ***Game Play***: *Game dynamics can increase engagement and be applied to direct effort.*

3.5 Identity, Trust, and Reputation

Aspects of identity, reputation, and trust all influence the way people interact with each other. Within a sensemaking context, interpersonal assessments affect how people value, consider, and respond to the individual contributions of others. Other things being equal, a hypothesis suggested by a more trusted or reputable person will have a higher probability of being accepted as part of the group consensus [131]. For social sensemaking in a computer-mediated environment, design challenges accrue around the various markers of identity and past action that might be transmitted through the system. For example, Donath [59] describes how even a cue as simple as one's e-mail address can lead to a number of inferences about identity and status.

3.5.1 Identity Presentation

Many theorists try to understand interpersonal perception via the signals available for interpretation by others. Goffman [75] distinguishes between expressions given and expressions given off to indicate those parts of our presentation of self that are consciously planned (e.g., the content of our speech) or unconsciously generated (e.g., a wavering of voice indicating nervousness), each of which is interpreted to form opinions of a person. Donath [59] classifies such signals into conventional signals—low cost signals that are easy to fake (e.g., talking about going to the gym)—and assessment signals—more reliable signals that are difficult to fabricate (e.g., having large muscles).

Other researchers have focused on the way media with different capacities for transmitting such signals affect interpersonal assessment. For example, most computer mediated communication filters out non-verbal cues, stripping many of the signals "given off" by participants. Despite these missing cues, Walther [184] argues that online relationships can be as deep and meaningful as face-to-face interactions through explicit sharing of personal information. However, due to diminished cues and asynchronous interaction, such online relationships can require longer time spans to develop. These diminished cues allow for a greater role of imagination and speculation when assessing another person. Furthermore, many researchers find that such diminished cues give rise to "deindividuation" effects that have both desirable and undesirable consequences. For example, people that are shy in face-to-face interactions often show greater rates of participation, but anti-social "flaming" is also more prevalent online [165, 184].

When considering the implications of identity assessment for collaborative visualization systems, designers should also take the context of deployment into account. If collaborators are already familiar to each other, they may require little additional support to make assessments of identity and reputation, instead relying on existing channels through which assessments can be made. It may be enough to simply identify collaborators' individual contributions with recognizable names. Still, it may prove valuable for visual analysis environments to interface with external communication channels, both for sharing and interpersonal assessment. Many organizations maintain online personnel directories to aid awareness and collaboration; visual analysis systems should be able to leverage such existing artifacts.

On the other hand, if collaborators begin as strangers, mechanisms for self-presentation and reputation formation need to be included in the system design. Possible mechanisms include identity markers, such as screen names, demographic profiles, social networks, and group memberships. Considerations include the type of personal information germane to the context of visual analysis; for example, is a playful or professional environment desired? Attributes such as age, geographic location, interests, and skills might help assess a collaborator's background knowledge, affecting the confidence one places in hypotheses. Of course, this picture is complicated if such measures are self-reported, because such self-reports may be subject to fabrication.

- ***Identity Markers***: *Indicate collaborators' identities in a contextually-appropriate manner.*
- ***User Profiles***: *Support awareness of other's backgrounds and skills.*

3.5.2 Reputation Formation

The development of interpersonal assessments over time leads to reputation and trust formation. In the case where participants only interact through the system itself, means of gauging a user's past actions or contributions are needed to not only aid awareness (c.f., §3.2) but to facilitate reputation formation. Observations of past actions provide implicit means of reputation formation, allowing collaborators to make inter-personal judgments grounded in past activity. One challenge for design is to consider what pieces of information are most informative for reputation formation.

Some systems instead provide explicit reputation mechanisms, such as seller ratings in online markets such as eBay [147]. In a visual analysis environment, collaborators might rate each other's contributions according to their interestingness or accuracy. Such ratings may help identify contributions with higher relevance, provide a reputation metric for contributors, and additionally constitute a social-psychological incentive for high quality contributions.

- ***Activity Histories***: *Personal action histories allow past contributions to be assessed.*
- ***Activity Summaries***: *Badges or activity summaries aid reputation and visibility of contributions.*

3.6 Group Dynamics

The makeup of collaborative groups is another aspect important to social sensemaking. Many scenarios, such as business and research, may involve work groups that are already well established. In such cases, standard group management and communication features common to many collaborative applications may be sufficient. However, when organizing effort in public goods scenarios, explicit mechanisms for assisting group formation may aid collaborative visualization efforts.

3.6.1 Group Management

At a basic level, formal group management mechanisms must present means for addressing issues of scalability and privacy. Group management mechanisms can support the coordination of a work group on a specific task within a larger collaborative environment, providing notification and awareness features at the group level. Groups also provide a means of filtering contributions, improving tractability and reducing information overload for participants who may not be interested in the contributions of strangers. Finally, groups provide a means of limiting contribution visibility, providing one mechanism for individual privacy within large-scale online scenarios.

An alternative approach to explicit group management is to support groups already formed in other mediated environments. Such support requires a decentralization of the analysis process, enabling collaborative visual analysis technologies to be embedded in external media. Examples include embedding an interactive visualization into a blog entry or introducing visual analysis applications into existing social environments such as Facebook. This strategy is common with existing social data analysis sites like Swivel [169] and Many-Eyes [181]: the longest and deepest discussions tend to occur around visualization screenshots posted to an external blog.

- ***Group Management***: *Group creation and management mechanisms address issues of scale and privacy.*

3.6.2 Group Size

One challenge for group management is the choice of group size. Larger groups may be able to achieve more through a larger labor pool, but can incur social and organizational costs [26]. For example, larger groups are more likely to suffer from the free rider problem [81] or social loafing [120] due to diluted accountability. Pirolli [141] describes a mathematical model of social information foraging that measures the benefit of including additional collaborators in information gathering tasks. His analysis finds that beyond certain sizes, additional foragers provide decreasing benefits, suggesting that an optimal group size exists, dependent on the parameters of the foraging task. An important future experiment would be to evaluate Pirolli's model through application to real visual analysis teams.

- ***Group Size:*** *Optimal group size determination can improve efficiency of analysis.*

3.6.3 Group Diversity

Another issue facing group formation is the diversity of group members. In this case diversity can include the distribution of domain-specific knowledge among potential participants and differences in attributes such as geographical location, culture, and gender. Organizational studies [53, 159] find that increased group diversity can lead to greater coverage of information and improved decision making. However, diversity can also lead to increased discord and longer decision times.

Various measurements of diversity may be applied to suggest a set of group members that will provide adequate coverage for an analysis task. Such measurements might come from analyzing differences between user profiles and structural features of the social networks of the participants [30]. Such networks may be explicitly articulated or inferred from communication patterns, such as the co-occurrence of commenters across discussion threads. Wu et al.'s [196] study of organizational information flow found that information spreads efficiently among homophilous (similar) group members but not across community boundaries, further suggesting the value of identifying structural holes and directing bridging individuals in the social network towards particular findings. By constructing user profiles based on demographic data, social connectivity, and prior usage, automated systems may be able to help suggest relevant tasks to appropriate community members.

- ***Group Diversity**: Appropriate within-group diversity can result in more complete results.*

3.7 Consensus, Dissemination, and Decision Making

The need to establish group consensus arises in many phases of the sensemaking cycle. Examples include agreement about the data to collect, how to organize and interpret data, and making decisions based upon the data. Collaborators may reach consensus through discussion or through the aggregation of individual decisions.

3.7.1 Consensus and Discussion

Mohammed [131] combines various contributions in social psychology and organizational studies to posit a model for cognitive consensus in group-decision making. Mohammed's model takes into account the assumptions, category labels, content domains, and causal models possessed by each participant, and how they evolve through discussion. One tangible recommendation that comes from this work is to clearly identify the points of dissent, creating focal points for further discussion and negotiation. From a design perspective, collaborators need communication mechanisms that allow points of dissent to be labeled and addressed. Collaborative tagging [76, 130] is one potential candidate. Formalizing contributions in structured argumentation systems [18, 79] may be another avenue. For example, a content analysis of contributions to the sense.us system (CHAPTER 4) found that users used free-text comments to post observations, questions, and hypotheses. These categories could be formally represented to help structure discussion and voting.

Scheff [158] notes that consensus requires more than participants simply sharing a belief; participants must think that their beliefs are the same, and achieve realization that others understand one's position. Users need feedback loops to gauge mutual understanding. Along these lines, it may be useful to consider the effects of multiple communication channels on decision processes. Collaborative visualization environments that provide messaging, in either synchronous or asynchronous forms, might provide backchannels for negotiation and non-public discussion. The integration of instant messaging into the e-mail applications such as Lotus Notes and GMail are examples of weaving different communication channels into a single system.

The value of different forms of consensus can vary based on the task at hand. Hastie [83] found that group discussion improved accuracy when decision tasks had demonstrably correct solutions because groups could evaluate their output. When task outcomes are open-ended, consensus through discussion is harder to evaluate. In a simulated graduate admissions task, Gigone and Hastie [74] found little value in discussion, as group decisions were well-matched by simply averaging prior individual decisions.

One design implication that again arises is to use voting or ranking systems. Mechanisms for expressing support or disdain for hypotheses could aid data interpretation and further identify controversial points. For example, Wikimapia [190] users can vote on the accuracy of labeled geographic regions and Swivel [169] supports ratings of interestingness. A game-like variation on this approach is the creation of prediction markets [168]: individuals can be given a limited amount of "points" or "currency" that they can use to vote for hypotheses they find promising. Hypotheses that are later validated could reap payback rewards for their supporters.

- ***Voting and Ranking:*** *Quantitative measures can be used to measure consensus and lower integration costs.*

3.7.2 Information Distribution

An important dimension of group consensus is the distribution of information across group members. Both Stasser [166] and Gigone and Hastie [74] find that it is difficult for groups to pool information effectively, and therefore, decision-making is biased in the direction of the initial information distribution. They hypothesize that the lack of effective pooling may be due to the persistence of individual decisions made prior to discussion or to information shared prior to the group meeting. The prior decisions and information set a common ground for discussion and thus the shared information is likely to be a focus of discussion, biasing conversation against unshared information. Thus, improving collective information foraging may help inform group decision-making by changing the information distribution. Collaborative analysis environments may facilitate better information pooling by providing a record of findings and opinions that participants can survey prior to decision-making and discussion.

3.7.3 Presentation and Story-Telling

Common forms of information exchange in group sensemaking are reports and presentations. Narrative presentation of analysis "stories" is a natural and often effective way to communicate analytic findings, and is a primary use of Spotfire Decision Site Posters [164]. Furthermore, users of online sites such as Many-Eyes [181] and Swivel [169] use external media such as blogs and social bookmarking services as external communication channels in which to share and discuss findings from visualizations. Viewers of analysis stories may also find value in conducting follow-up analysis and verification on parts of the story, enabling presentations to serve as a catalyst for additional analysis.

The challenge to collaborative visualization is to provide mechanisms to aid the creation and distribution of presentations. GeoTime Stories [63] supports textual story-telling with hyperlinks to visualization states and annotations. In CHAPTER 4, we introduce the sense.us system, which supports the creation of tours by creating a trail of bookmarked visualization views. However, neither system allows these stories to be easily exported outside the respective applications. In CHAPTER 7, we introduce a graphical analysis history tool that improves upon these systems with support for building and exporting presentations semi-automatically from interaction histories.

- ***Presentation****: Support creation and export of presentations/tours for telling analysis stories.*

3.8 Conclusion and Future Directions

The overarching goal of this thesis is to design socio-technical systems that improve our collective analytic capabilities by promoting an effective division of labor among participants, facilitating mutual understanding, and reducing the costs associated with collaborative tasks. The design considerations for collaborative visual analytics presented in this chapter are the results of our attempt to identify the social interactions underlying successful collaborations and in many cases suggest mechanisms for facilitating them. We summarize these considerations in Table 3.1.

Table 3.1. Selected design considerations for collaborative visual analytics.

Design Consideration	Description	Section
Modularity and Granularity	Identify appropriately-scoped units of work that form basic analytic contributions.	§3.1
Cost of Integration	Synthesize work in a manner that lowers integration costs and improves scalability while maintaining quality.	§3.1
Selective Visibility	Analysts should be able to attend to or ignore social activity as desired.	§3.1
Shared Artifacts	Structure social sensemaking through shared, editable representations.	§3.1
Artifact Histories	Provide histories of actions performed on artifacts.	§3.2
View Sharing	Enable sharing of views across media with bookmarks (*e.g.*, URLs).	§3.2
Content Export	Support embedding of views in external media (*e.g.*, email, blogs, reports).	§3.2
Discussion	Support commentary; consider implications of discussion model on grounding.	§3.2
Notification	Support notification subscriptions for views, artifacts, people, and groups.	§3.2
Action Flags	Mark needed future actions: unanswered questions, need for evidence, etc.	§3.2
Social Navigation	Make social activity visible, indicate popular and neglected data regions.	§3.2
Pointing	Support nuanced pointing through selection techniques and visual effects.	§3.3
Personal Relevance	Increase engagement by increasing personal relevance of data sets.	§3.4
Social-Psychological Incentives	Facilitate positive feedback and visibility of contributions.	§3.4
Game Play	Game design elements can provide incentives and be used to direct effort.	§3.4
Identity Markers	Indicate collaborator's identities in a contextually-appropriate manner.	§3.5
User Profiles	Support awareness of others' backgrounds and skills.	§3.5
Activity Histories	Personal action histories allow past contributions to be assessed.	§3.5
Activity Summaries	Activity indicators or summaries aid reputation and visibility of contributions.	§3.5
Group Management	Group creation and management mechanisms address issues of scale and privacy.	§3.6
Group Size	Optimal group size determination can improve efficiency of analysis.	§3.6
Group Diversity	Appropriate within-group diversity can result in more complete results.	§3.6
Voting and Ranking	Quantitative measures can be used for consensus and to lower integration costs.	§3.7
Presentation	Support creation and export of presentations for telling analysis stories.	§3.7

Considering these considerations in turn provides a research agenda for collaborative visual analytics, surfacing hypotheses in need of study and suggesting new technical mechanisms. We envision future research projects of varying scopes. Researchers may focus on new visualization and interaction techniques for supporting collaboration (e.g., CHAPTERS 5-8 of this thesis). Such research should propose novel mechanisms, ideally accompanied by an empirical evaluation. As listed above, new discussion models, pointing techniques, and story-telling interfaces are all candidates.

Research into targeted techniques needs to be balanced with the design, deployment, and evaluation of collaborative visual analysis environments. Such systems should enable real-world groups to engage in social data analysis. Studies of system usage should measure the benefits of collaborative analysis in ecologically valid settings and inform best practices for combining collaboration mechanisms. A number of important experiments, such as those involving group management and incentives, may be best conducted in real-world settings (e.g., [120, 147, 196]) and interfacing with the Internet is critical to understanding how findings are disseminated and how collaborative visual analytics can be more deeply weaved into the Web.

Accordingly, we use these design considerations to select the subset of open problems considered in the remainder of this thesis.

- CHAPTER 4 applies these considerations to the design, implementation, and evaluation of sense.us, an end-to-end web application for social data analysis.
- CHAPTER 5 presents the design and evaluation of new social navigation cues for improving awareness and facilitating navigation between visualization views.
- CHAPTER 6 introduces a visual query interface that supports pointing via data-aware annotations that persist across time-varying data and changes of visual encodings.
- CHAPTER 7 explores the use of graphical histories to review and revisit analysis trails and to create tours and presentations for telling analysis stories.
- CHAPTER 8 studies animated transitions designed to improve users' comprehension of the relationship between consecutive views while increasing hedonic incentive.

4 sense.us: A Web Application for Collaborative Visual Analysis

To better catalyze and support collaborative visual analysis, we designed and implemented a website, *sense.us*, aimed at group exploration of demographic data. Guided by the design considerations of the previous chapter, the site provides a suite of interactive visualizations and facilitates collaboration through view bookmarking, doubly-linked discussions, graphical annotation, saved bookmark trails, and social navigation through comment listings and user profiles. We then ran user studies to observe closely how people engage in social data analysis. The studies also allowed us to evaluate the new design elements in the site and suggest directions for future work.

4.1 sense.us: A Site for Collaborative Visual Analysis

The *sense.us* system is a prototype web application for social visual data analysis. The site provides a suite of visualizations of United States census data over the last 150 years (see Figures 4.1 and 4.2) and was designed for use by a general audience. We built sense.us to apply our design considerations in a real system that we could deploy and study. Our primary goal was to create an environment for catalyzing and observing collaborative data exploration.

The primary interface for sense.us is shown in Figure 4.1. In the left panel is a Java applet containing a visualization. The right panel provides a discussion area, displaying commentary associated with the current visualization view, and a graphical bookmark trail, providing access to views bookmarked by the user.

Previously published by the author, Fernanda Viégas, and Martin Wattenberg in [89].

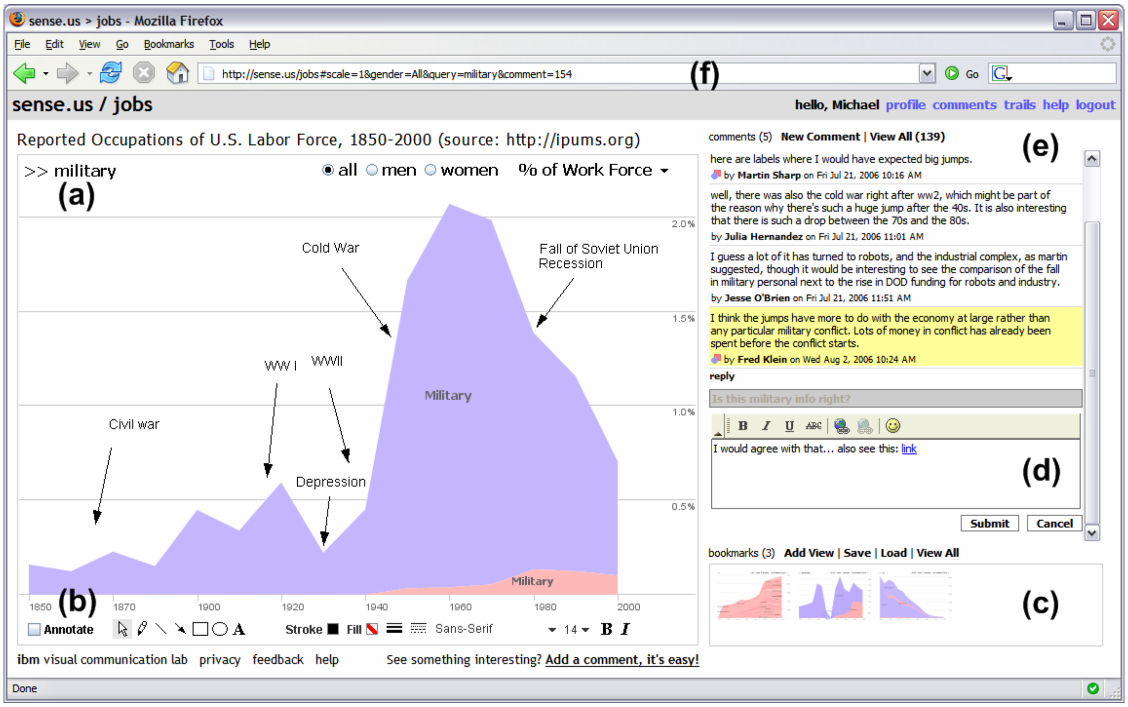


Figure 4.1. The sense.us collaborative visualization system. (a) An interactive visualization applet, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage of the work force in military jobs is shown. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. Bookmarks can be dragged onto the text field to add a link to that view in the comment. (e) Threaded comments attached to the current view. (f) URL for the current state of the application. The URL is updated automatically as the visualization state changes.

The visualization in Figure 4.1—named the *Job Voyager*—depicts U.S. occupation data from 1850 to 2000. The colored bands represent occupations sub-divided by gender. The height of a band represents the number of people employed in the given job, either as a percentage of the labor force within the given census year or as raw counts. Bands are stacked on top of each other such that the total height of the stacks represents the summed total of the visible data. Users navigate to different views of the data by typing in keyword queries to filter for matching prefixes, filtering based on gender, and by selecting between normalized data and raw counts. Other available data visualizations are shown in Figure 4.2, and include statistical maps, stacked time-series of immigration data, and an interactive population pyramid.

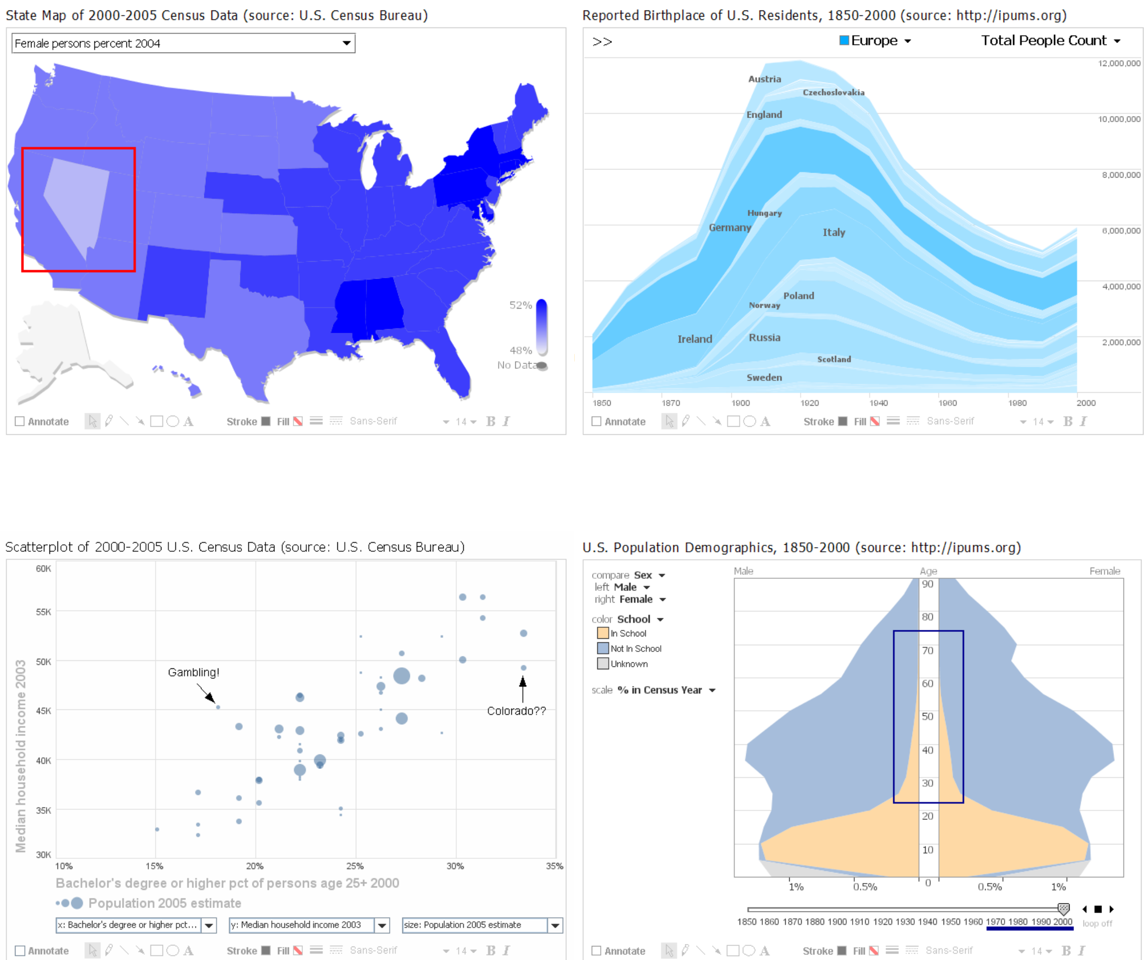


Figure 4.2. Sample visualizations from sense.us. Clockwise from top-left: (a) Interactive state map. The image shows the male/female ratio of the states in 2005. (b) Stacked time-series of immigration data, showing the birthplace of U.S. residents over the last 150 years. The image shows the number of U.S. residents born in European countries. (c) Population pyramid, showing population variation across gender and age groups. Additional variables are encoded using stacked, colored bands. The image visualizes school attendance in 2000; an annotation highlights the prevalence of adult education. (d) Scatter plot comparing education and income levels for each state.

We chose to visualize census data for multiple reasons. First, it is a large and socially interesting data set with public policy and education implications. Second, many of the data series should be familiar to a general audience, providing access to potentially larger user base. Third, as the data is about people and their history, we hoped that many users would find the data personally relevant. When designing the visualizations for the site, we attempted to provide enough analytic functionality (e.g., data selection, filtering, and normalization) to enable to rich exploration while otherwise trying to minimize the overall complexity of the interface.

For each of the provided visualizations, sense.us facilitates social data analysis through view sharing, commentary, and annotation. With a straightforward bookmarking mechanism, sense.us supports collaboration with features described in detail below: doubly-linked discussions, graphical annotations, saved bookmark trails, and social navigation via comment listings and user activity profiles.

4.1.1 View Sharing

When collaborating with visualizations, participants must be able to see the same visual environment in order to ground each others' actions and comments (CHAPTER 3). To this aim, the sense.us site provides a mechanism for bookmarking views. The system makes application bookmarking transparent by tying it to conventional web bookmarking. The browser's location bar always displays a URL that links to the current state of the visualization, defined by the settings of filtering, navigation, and visual encoding parameters. The visualization state is listed in the URL as a set of human-readable (and editable) name-value pairs. As the visualization view changes, the URL updates to reflect the current state (Figure 4.1f), simplifying the process of sharing a view through e-mail, blogs, or instant messaging by enabling users to cut-and-past a link to the current view at any time. To conform to user expectations, the browser's back and forward buttons are tied to the visualization state, allowing users to navigate to previous views.

4.1.2 Doubly-Linked Discussion

To situate conversation around the visualization, we created a technique we call *doubly-linked discussion*. The method begins with an “independent” discussion interface in which users can attach comments to particular states (or views) of a visualization. Comments are shown on the right side of the web page and grouped into linear discussion threads (Figure 4.1e). Each comment shows the thread topic, comment text, the author's full name, and the time at which the comment was authored. Clicking on a comment takes the visualization to a bookmarked state representing the view seen by the comment's author.

Users can add comments either by starting a new thread or posting a reply to an existing thread. When a "New Comment" or "Reply" link is clicked, a text editor appears at the site where the comment will be inserted and the graphical annotation tools (discussed next) become active. Upon submission, the comment text and any annotations are sent to the server and the comment listing is updated.

The interface is based on links from the commentary into the visualization. Our system also provides links in the other direction: from the visualization into the discussion. As a user changes parameters and views in the visualization, they may serendipitously happen upon a view that another person has already commented on. When this occurs, the relevant comments will automatically appear in the right-hand pane. Our intuition is that this "doubly-linked" discussion interface, which combines aspects of independent and embedded discussion, facilitates grounding and enables the visualization itself to become a "place" [D] in which social interaction can occur.

4.1.3 Pointing via Graphical Annotation

In real-time collocated collaboration, participants commonly use both speech and gesture, particularly pointing [46, 99], to refer to objects and direct conversation. For asynchronous collaboration, graphical annotations can play a similar communicative role. We hypothesized that graphical annotations would be important both for pointing behavior and playful commentary. To add a pictorial element to a comment or point to a feature of interest, authors can use drawing tools (Figure 4.1b) to annotate the commented view. These tools allow free-form ink, lines, arrows, shapes, and text to be drawn over the visualization view. The tools are similar to presentation tools such as Microsoft PowerPoint and are intended to leverage users' familiarity with such systems.

Comments with annotations are indicated by the presence of a small shape logo to the left of the author's name in the comment listing (see Figure 4.1e). When the mouse hovers over an annotated comment, the comment region highlights in yellow and a hand cursor appears. Subsequently clicking the region causes the annotation to be shown and the highlighting to darken and become permanent. Clicking the comment again (or clicking a different comment) will remove the current annotation and highlighting.

We refer to this approach as *view annotation*, which operates like an "acetate layer" over the visualization, in contrast to *data-aware* annotations directly associated with the underlying data. Aside from the freedom of expression it affords, view annotation also has a technical advantage: it allows reuse of the identical annotation system across visualizations, easing implementation and preserving a consistent user experience. We implemented a free-form view annotation mechanism so that we could first study pointing behaviors in an unconstrained medium, we will revisit data-aware annotations in CHAPTER 6.

4.1.4 Collecting and Linking Views

In data analysis it is common to make comparisons between different ways of looking at data. Furthermore, storytelling may play an important role in social usage of visualizations, as discussed in [179]. We hypothesized that the ability to embed multiple view bookmarks into a single comment would both facilitate comparison and enable the telling of stories that range over multiple views.

To support such multi-view comments and narratives, we created a "bookmark trail" widget. As a user navigates through the site, he or she can click a special "Add View" link to add the current view to a graphical list of bookmarks (Figure 4.1c). Bookmarks from any number of visualizations can be added to a trail. A trail may be named and saved, making it accessible to others.

The bookmark trail widget also functions as a short-term storage mechanism when making a comment that includes links to multiple views. Dragging a thumbnail from the bookmark trail and dropping it onto the text area creates a hyperlink to the bookmarked view; users can then directly edit or delete the link text within the text editor. When the mouse hovers over the link text, a tooltip thumbnail of the linked view is shown.

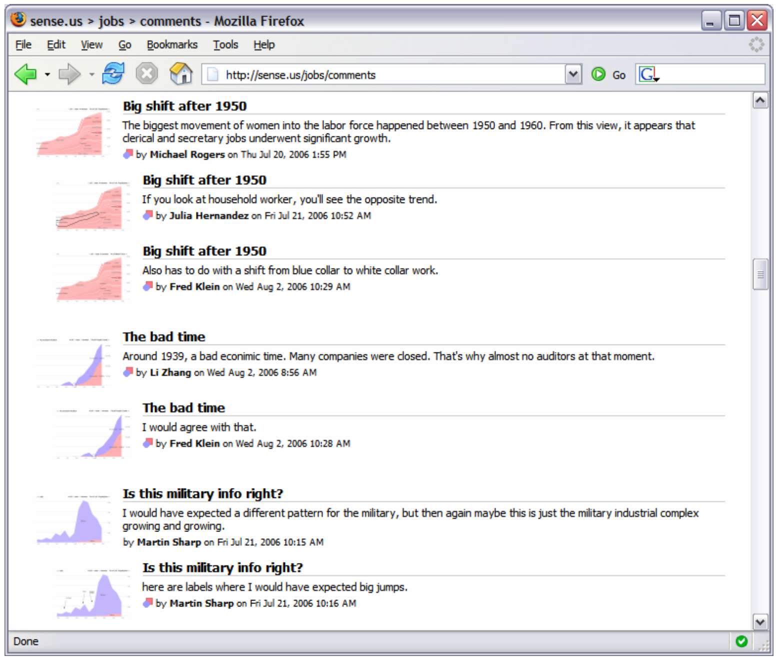


Figure 4.3. The sense.us comment listing page. Comment listings display all commentary on visualizations and provide links to the commented visualization views. Indented comments indicate replies within a conversation thread.

4.1.5 Awareness and Social Navigation

Social navigation (§3.2.3) leverages usage history to provide additional navigation options within an information space. Our initial system supports social navigation through comment listings and user profile pages that display recent activity. Comment listings provide a searchable and sortable collection of all comments made within the system, and can be filtered to focus on a single visualization (see Figure 4.3). Comment listing pages include the text and a thumbnail image of the visualization state for each comment. Hovering over the thumbnail yields a tooltip with a larger image. Clicking a comment link takes the user to the state of the visualization where the comment was made, displaying any annotations included with the comment. The author's name links to the author's profile page, which includes their five most recent comment threads and five most recently saved bookmark trails. The view also notes the number of comments made on a thread since the user's last comment, allowing users to monitor the activity of discussions to which they contribute.

Although more elaborate social navigation mechanisms are possible, we wanted to observe system usage with just these basic options. We were particularly interested in observing the potential interplay between data-driven exploration and social navigation. By allowing discussions to be retrieved unobtrusively while a user explores the data, potentially relevant conversation can be introduced into the exploration process. Meanwhile, comment listings and indications of recent posts may help users find views of interest, making social activity a catalyst for data exploration.

4.1.6 Unobtrusive Collaboration

Finally, while designing sense.us we also wished to follow a common CSCW design guideline: collaborative features should not impede individual usage [80]. Hence we did not litter views with prior annotations or commentary. Rather, commentary on a visualization is retrieved and displayed unobtrusively on the right side of the screen and graphical annotations are displayed "on demand" by the user.

4.2 Implementation Notes

While many aspects of sense.us rely on well-known techniques, this section provides implementation details for the more complex features: application bookmarking, doubly-linked discussions, and graphical annotations.

4.2.1 Application Bookmarking

Bookmarks of visualization state are implemented as a set of name-value pairs of visualization parameters, listed using standard URL query syntax. Normally, changing the browser's URL will force a reload of the page to prevent security attacks. Because a reload would cause a disruptive restart of the visualization applet, the bookmark URL encodes the query string as a page anchor—using the URL # delimiter instead of the standard ? delimiter—so that the URL updates in place. Furthermore, updated URLs are put into the browser's history stack, so that the browser's back and forward buttons have their usual behavior [129]. When a visualization URL is updated due to use of the back or forward buttons or manual typing, scripts send the updated URL to the applet, which is parsed and used to update the current visualization state.

4.2.2 Doubly-Linked Discussions

The bookmarking mechanisms alone are not sufficient to support doubly-linked discussions. To see the challenge in linking from a view state back to all comments on that view, consider the visualization in Figure 4.1. When a user types "military" into the top search box, they see all jobs whose titles begin with the string "military." On the other hand, if they type only "mili," they see all titles beginning with "mili"—but this turns out to be the identical set of jobs. These different parameter settings result in different URLs, and yet provide exactly the same visualization view. More generally, parameter settings may not have a one-to-one mapping to visualization states. To attach discussions to views we therefore need an indexing mechanism which identifies visualization states that are equivalent despite having different parametric representations.

We solve this indexing problem by distinguishing between two types of parameters: *filter parameters* and *view parameters*. Filter parameters determine which data elements are visible in the display. Rather than index filter parameters directly, we instead index the filtered state of the application by noting which items are currently visible, thereby capturing the case when different filter parameters give rise to the same filtered state. View parameters, on the other hand, adjust visual mappings, such as selecting a normalized or absolute axis scale. Our current system indexes the view parameters directly. The bookmarking mechanism implements this two-part index by computing a probabilistically unique hash value based on both the filtered state and view parameters. These hash values are used as keys for retrieving the comments for the current visualization state. One limitation is that the hashing scheme assumes that the data set is static; data updates can require re-hashing the space of views.

4.2.3 Graphical Annotation

The graphical annotations take the form of vector graphics drawn above the visualization. When a new comment is submitted, the browser requests the current annotation (if any) from the visualization applet and saves the annotation with the comments. The applet saves annotations in an XML format, compresses it using gzip, and encodes the result as a base 64 string before passing it to the browser. When comments are later retrieved from the server, the encoded annotations are stored in the browser as JavaScript variables. When the user requests that an annotation be displayed, the encoded annotations are passed to the applet, decoded, and drawn.

4.3 Usage Observation of sense.us

To gain a preliminary understanding of asynchronous collaboration practices with visualizations, we ran exploratory user studies of the sense.us system. The studies had two specific goals: first, to better understand emergent usage patterns in social data analysis; second, to learn how well the various features of the sense.us system supported such social analysis. We ran the studies in two parts: a pair of controlled lab studies and a 3-week live deployment on the IBM corporate intranet. To analyze the data, we employed a mixed-methods analysis approach combining qualitative and quantitative observations.

Laboratory Study

We first ran a pilot study with 6 subjects (2 female, 4 male), all of whom were members of our immediate research team. Comments and annotations made in the pilot were visible in a subsequent 12 subject (3 female, 9 male) study, with subjects drawn from our greater research lab. Subjects were at least peripherally familiar with each other and many were co-workers. Ages ranged from the early-twenties to mid-fifties and education varied from the undergraduate to the doctoral level, spanning backgrounds in computer science, design, social science, and psychology. Concerned that our lab's focus in collaborative software might bias results, we replicated the lab study in a university environment with an additional 12 subjects (5 female, 7 male). Subject variation in age, education, and social familiarity remained similar.

Subjects conducted a 25 minute usage session of the sense.us system. A single visualization was available in the study: a stacked time-series of the U.S. labor force over time, divided by gender (Figure 4.1). Users could navigate the visualization by typing in text queries (matched to job title prefixes), filtering by gender, and setting the axis scale, either to total people count or percentage values.

This data set was chosen for several reasons. First, most users should have no difficulty relating to job data. Second, like many other real world data sets, there are data collection issues, including missing data and unclear or antiquated labels. Third, we suspected the data would be an interesting test case for annotations, as in many visualization views text seemed sufficient for referencing spikes or valleys in the data.

Participants were first given a brief tutorial of system features, including how to navigate the visualization and create comments and annotations. They were then instructed to use the system however they liked--no specific tasks were given. However, users were told that if they felt at a loss for action, they could browse the data for trends they found interesting and share their findings. An observer took notes and we used a think-aloud protocol. The software also logged user actions. We ran subjects in sequential order, such that later participants could view the contributions of previous subjects but not vice versa. The system was seeded with 5 comments, each an observation of a particular data trend.

After the study, subjects took a short exit questionnaire about their experiences. Participants were asked to rate on a 5-point Likert scale to what degree (1) they enjoyed using the system, (2) they learned something interesting, (3) others' comments were helpful in exploring the data, and if they found annotations useful for (4) making their own comments or (5) understanding others' comments. Subjects were also asked free response questions about what they liked, disliked, and would change about the system.

Live Deployment

We also conducted a live deployment of the system on the IBM corporate intranet for 3 weeks. Any employee could log in to the system using their existing intranet account. Eight visualizations were available in the system, among them the visualizations of Figures 4.1 and 4.2 and a scatterplot of demographic metrics (see Figure 4.5). We also introduced two visualizations specific to the company: stacked time-series of keyword tagging activity and individual user activity on *dogear* [130], an internal social bookmarking service. The site was publicized through an email newsletter, an intranet article, and individual emails.

Study Findings

In the rest of this section, we report observations from these studies, organized by commentary, graphical annotations, navigation patterns, and use of doubly-linked discussion. As variation in content and tone differed little across studies, the discussion incorporates data aggregated from each. The data analyzed was drawn from 12.5 hours of qualitative observation and from usage logs including 258 comments: 41 from the pilot, 85 from the first study, 60 from the second, and 72 from the deployment.

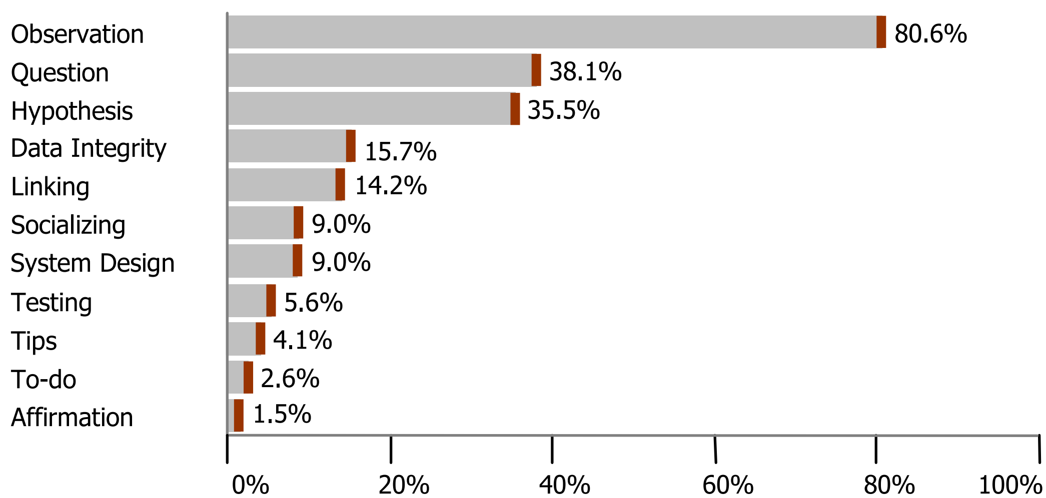


Figure 4.4. Content analysis categorization of sense.us comments. The chart shows the prevalence of different aspects of discussion. Categories are *not* mutually exclusive.

4.3.1 Commentary: Observation, Question, and Hypothesis

We first examined how comments were used to conduct social data analysis—was there a recognizable structure to the discussions? To find out, we performed a formal content analysis [115] on the collected comments. The members of our research team independently devised a coding rubric based upon a reading of the comments. We then compared our separate rubrics to synthesize a final rubric that each author used to independently code the comments. The final coding rubric categorized comments as including zero or more of the following: observations, questions, hypotheses, links or references to other views, usage tips, socializing or joking, affirmations of other comments, to-dos for future actions, and tests of system functionality. We also coded whether or not comments made reference to data naming or collection issues, or to concerns about the web site or visualization design. To facilitate objectivity, we tested the inter-rater reliability of the coded results using Cohen's kappa statistic. The lowest pair-wise kappa value was 0.74, indicating a satisfactory inter-rater reliability. Figure 4.4 shows the prevalence of the content categories across the collected commentary.

Most commentary on sense.us involved data analysis. A typical comment made note of an observed trend or outlier, often coupled with questions, explanatory hypotheses, or both. A typical reply involved discussing hypotheses or answering questions. In total, 80.6% of comments involved an observation of visualized data, 35.5% provided an explanatory hypothesis, and 38.1% included a question about the data or a hypothesis. Most questions and hypotheses accompanied an observation (91.6% and 92.2%, respectively) and half the hypotheses were either phrased as or accompanied by a question (49.0%).

For example, participants in both lab studies discovered a large drop in bartenders around the 1930's and posted comments attributing the drop to alcohol prohibition. In the live deployment, one user commented on a scatterplot view, asking why New Hampshire has such a high level of retail sales per capita (Figure 4.5). Another user noted that New Hampshire does not have sales tax, and neither does Delaware, the second highest in retail sales. In this fashion, discussion involved the introduction of contextual information not present in the visualization. For instance, Figure 4.1 includes a timeline of events that was iteratively constructed by multiple users, while the graph of teachers in Figure 4.6 notes the introduction of compulsory education.

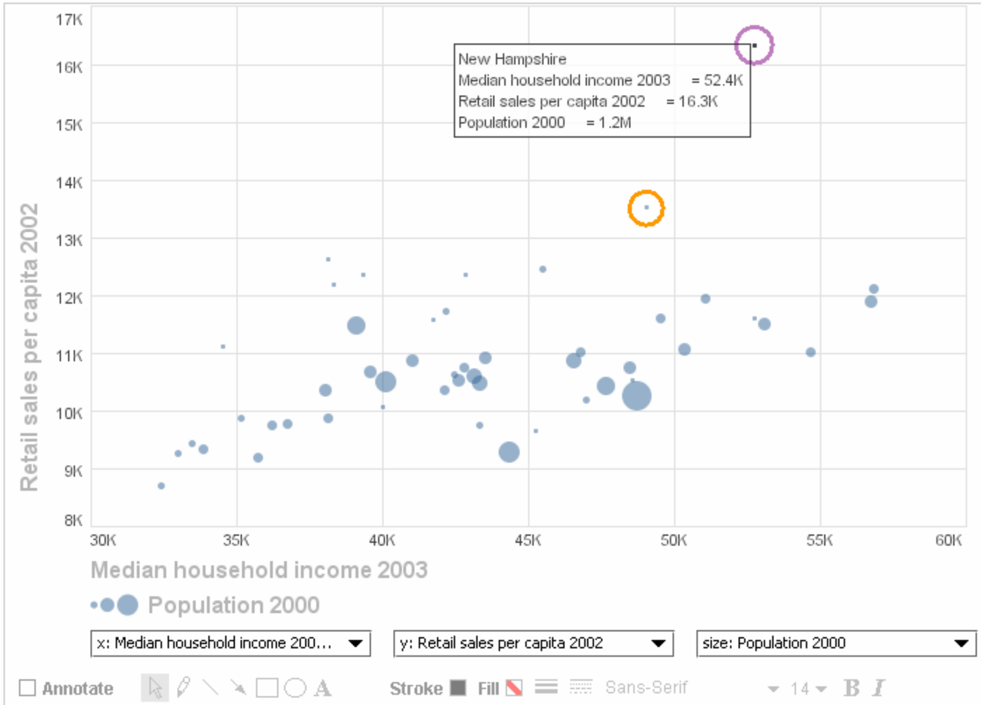


Figure 4.5. Scatterplot of U.S. states showing median household income (x-axis) vs. retail sales (y-axis). New Hampshire and Delaware have the highest retail sales.

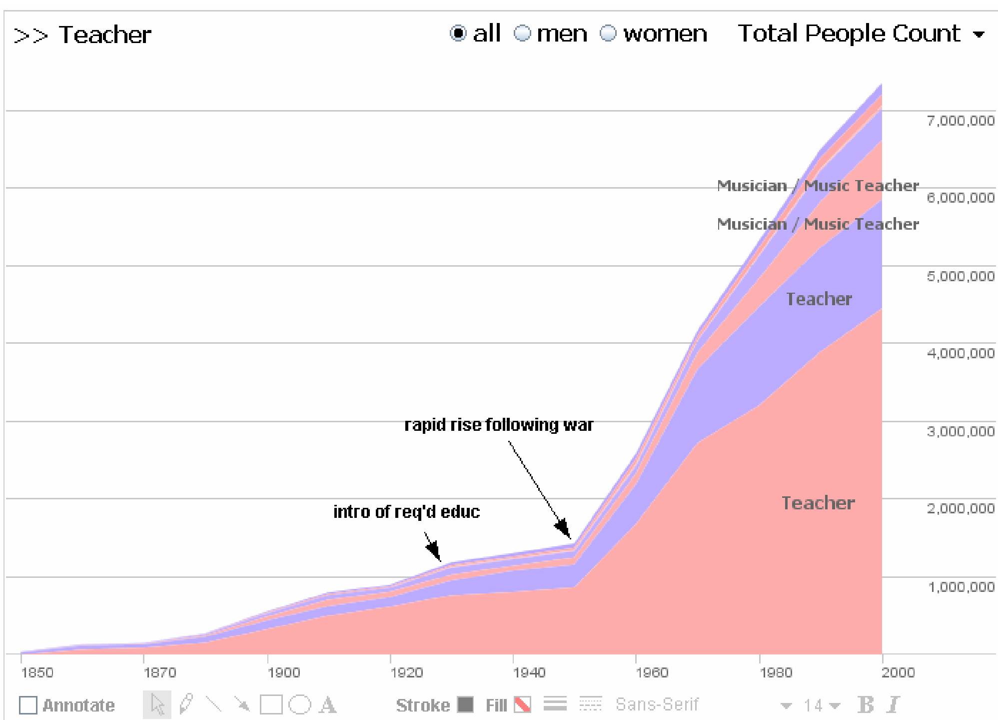


Figure 4.6. Visualization of the number of teachers. Annotations indicate the start of compulsory education and the rise of teachers in the post World War II era.

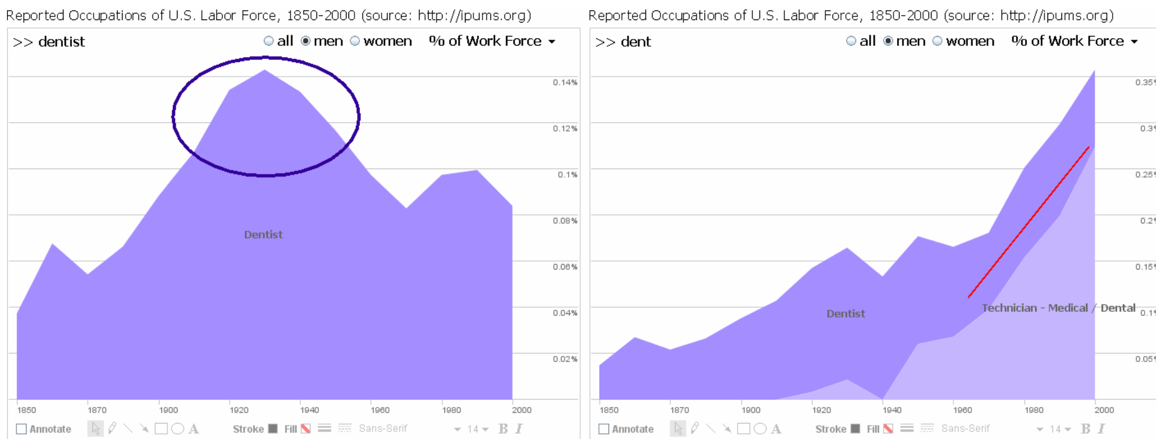


Figure 4.7. Social data analysis of dentistry. (a) Left: A number of subjects contributed hypotheses to explain an observed peak and subsequent decline in the percentage of dentists, including improved prenventative measures such as fluoridation of the water supply. **(b) Right:** Another subject linked to a view of dentists and hygienists, suggesting growth of the dental profession and stratification of labor among doctors and assistants.

One instance of social data analysis occurred around a rise, fall, and slight resurgence in the percentage of dentists in the labor force (Figure 4.7). The first comment (one of the five seed comments) noted the trends and asked what was happening. One subject responded in a separate thread, *"Maybe this has to do with fluoridation? But there's a bump... but kids got spoiled and had a lot of candy??"* To this another subject responded *"As preventative dentistry has become more effective, dentists have continued to look for ways to continue working (e.g., most people see the dentist twice a year now v. once a year just a few decades ago)."* Perhaps the most telling comment, however, included a link to a different view, showing both dentists and dental technicians. As dentists had declined in percentage, technicians had grown substantially, indicating specialization within the field. To this, another user *asked "I wonder if school has become too expensive for people to think about dentistry, or at least their own practice when they can go to technical school for less?"* Visual analysis, historical knowledge, and personal anecdote all played a role in the sensemaking process, explicating various factors shaping the data.

Another role of comments was to aid data interpretation, especially in cases of unclear meaning or anomalies in data collection. Overall, 15.7% of comments referenced data naming, categorization, or collection issues. One prominent occupation was labeled

"Operative," a general category consisting largely of skilled labor. This term had little meaning to subjects, one of whom asked *"what the hell is an operative?"* Others responded to reinforce the question or to suggest an explanation, e.g., *"I bet they mean factory worker."* Another subject agreed, noting that the years of the rise and fall of operatives seemed consistent with factory workers. Other examples include views missing data for a single year (1940 was a common culprit), leading users to comment on the probable case of missing data.

Some users were less interested in specific views than in recurring patterns. One user was interested in exploring careers that were historically male-dominated, but have seen increasing numbers of females in the last half-century. The user systematically explored the data, saving views in a bookmark trail later shared in a comment named "Women's Rise." Similarly, a mathematically-minded participant was interested in patterns of job fluctuations and created a trail of recurring distributions. Another searched for jobs that had been usurped by technology, such as bank tellers and telephone operators. In each case, the result was a tour through multiple views.

Overall, 14.2% of comments referenced an additional view, either implicitly in the text or explicitly through drag-and-drop bookmark links. Although 22 of the 24 lab study subjects (87.5%) saved at least one view to the bookmark trail, only 14 (58.3%) created one or more drag-and-drop bookmark links. The amount of view linking varied by user, ranging from 0 to 19 links with an average of 2.17.

Comments served other purposes as well. A number were simple tests of system functionality (5.6%), often deleted by the user. Some included tips for using the system (4.1%), noting how to take advantage of specific features. Overall, 9.0% of comments referenced the site design, either in the form of usage tips or feature requests. A few comments included to-dos for future work (2.6%), such as later adding a link to a relevant wikipedia article. Others served solely as affirmations to another comment (1.5%). For example, people stating "I agree with that" to support a hypothesis. In many cases, study participants would note out loud "that is interesting!" without posting a comment to the system.

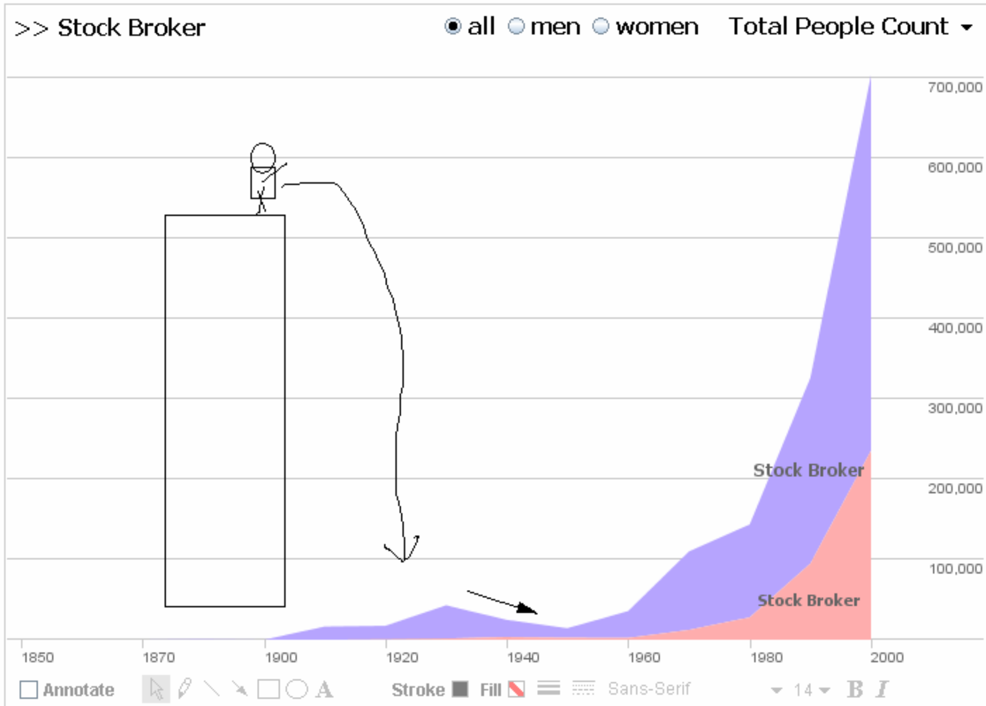


Figure 4.8. Annotated view of stock brokers. The attached comment reads *"Great depression 'killed' a lot of brokers"*.

Finally, some comments were social in nature (9.0%). Most pointed out trends in the data, but did so in a joking manner. One user built a view comparing female lawyers and bartenders, writing "Women at the bar… and behind the bar." In the pilot study, one of our lab members annotated a drop in stock brokers after 1930 with a picture of a person's trajectory off a skyscraper (Figure 4.8). This elicited smiles and laughter from subjects in the subsequent study, one of whom replied with an affirmation simply saying "Whoa!"

We also analyzed the structural aspect of comments. Excluding comments from the pilot study, deleted test comments, and those written by the paper authors, 195 comments were collected. Of those, 140 (71.8%) started new discussion threads while 55 (28.2%) were replies to existing threads. The average thread length was 1.35 comments (SD = 0.82), with a maximum of 5 comments. In some cases, discussion spanned multiple threads.

4.3.2 Graphical Annotation: Pointing and Play

Next we examined how graphical annotations were used and to what degree they contributed to social data analysis. Of the 195 non-pilot, non-deleted comments, 68 (35.9%) included annotations. The vast majority of annotations (88.6%) involved pointing to items or trends of interest. Such annotations were often accompanied with deictic references [21] (e.g., "this spike") in the text commentary, an indication of the use of annotations as a form of pointing [46] that provides grounding [49] for discussion (c.f., CHAPTER 3). The remainder (11.4%) involved more playful expression, such as drawn smiley faces and the visual commentary of Figure 4.8.

Across these annotations, a total of 179 "shapes" were drawn, with the options being free-form ink, lines, arrows, rectangles, ovals, and text. The prevalence of different shapes is shown in Figure 4.9. Arrows were the most popular shape (25.1% of shapes), and were used to point to items as well as to situate information provided by text captions (24.6%). Subjects primarily used ovals (17.9%) to enclose regions of interest, and used free-form ink drawn with the pencil tool (16.2%) for pointing, enclosing irregularly shaped regions, and freeform drawing. Of the rest, lines made up 14.5% of all shapes and rectangles only 1.7%.

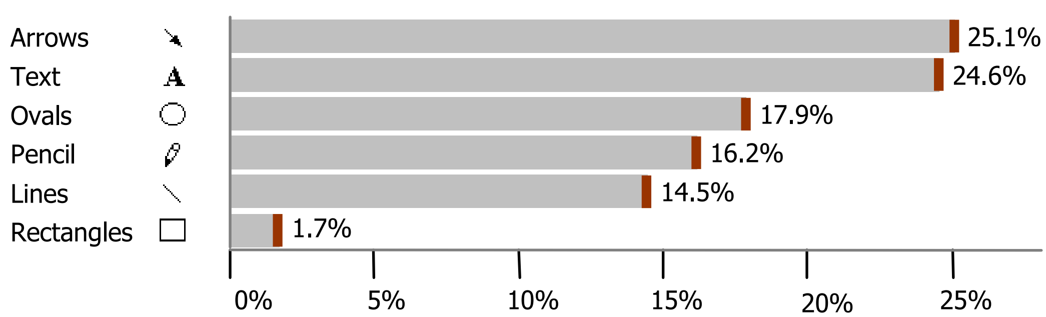


Figure 4.9. Usage of sense.us graphical annotation tools.

A few users, particularly those with experience in graphic design, noted that graphical annotations were their favorite feature. Other users noted that the annotations were often unnecessary for comments where text could describe the trend(s) of interest. A few of these users added annotations to such views anyway, saying the annotations were "surprisingly satisfying," enabling "personal expression." Exit survey results somewhat reflected these views, as users ranked annotations more useful for writing their own comments (M = 3.5/5.0, SD = 0.85) than understanding others' comments (M = 3.2/5.0, SD = 0.90). This difference, however, did not reach statistical significance ($t(23) = -1.67$, $p < 0.108$, two-tailed).

4.3.3 Visitation and Navigation: Voyagers and Voyeurs

We then investigated how users navigated the visualizations. Most users began exploring the data directly, starting from the default overview and drilling down. A few immediately went to the comments listing to see what others had done. Many participants searched for their own occupations and those of friends and family. Other strategies included browsing for items of interest found in the overview (*"Wow, look how the poor farmers died out"*) and formulating queries based on an over-arching interest, such as gender balance.

Looking to the usage logs, navigation by interaction with the visualization or attached commentary was by far the most common navigation technique. As shown in Figure 4.10, this accounted for 70.5% of navigation actions. The second most popular was the back and forward buttons at 17.5%, leveraging our integration of the visualization with browser history mechanisms. Following a link from the comment listings accounted for 8.7% of all views, while the final 3.3% were due to clicking a bookmark in the bookmark trail.

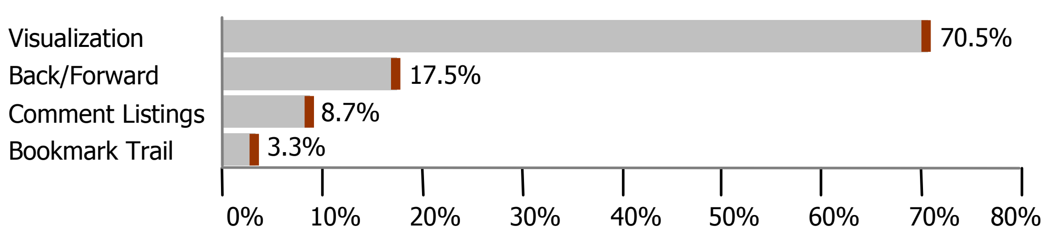


Figure 4.10. Usage of sense.us navigation mechanisms.

At some point, every subject explored the comment listings. Some felt they would find interesting views more quickly. Remarks to this effect included *"I bet others have found even more interesting things"* and *"You get to stand on the shoulders of others."* Other subjects were interested in specific people they knew or discovering what other people had investigated. Said one participant, *"I feel like a data voyeur. I really like seeing what other people were searching for."* Switching between data-driven exploration and social navigation was common. Views discovered via comment listings often sparked new interests and catalyzed more data-driven exploration. After some exploration, participants routinely returned to the listings for more inspiration. Thus, the system engaged users in a feedback loop between data-driven investigation and social activity traces. In the survey, the question "Did you find other people's comments useful for exploring the data?" received the highest marks (M = 4.46/5.0, SD = 0.63).

4.3.4 Doubly-Linked Discussions

We also wanted to investigate participant reaction to the doubly-linked model of comments. All users understood the model readily and no problems were reported when users wanted to comment on a specific view. The model became more problematic when users wanted to comment on multiple views. In this case, the user had to choose one view as primary, comment on that, and then reference the other views, either indirectly in the text or by creating a link from the bookmark trail. Some users expressed the opinion that creating links was a workable solution, while others wanted to be able to simultaneously compare multiple views for purposes of both analysis and commentary.

One important aspect of doubly-linked discussions is the problem of determining identical views, despite potentially differing visualization parameters. In this respect, we found our indexing scheme improved the odds of discovering existing commentary while navigating the visualization. Across both lab studies, 28.2% of all unique visits to a visualization state were to a view that had been reached through two or more differing parameter settings. Without the view indexing, there would be a much higher potential for "cross-talk," where users post comments concerning similar observations on related views, unaware of each other. Nonetheless, cross-talk was observed in a total of six cases, typically when both normalized and absolute axis scales led to similar views. In two cases, participants added linking comments that bridged the related discussions.

4.3.5 User Experience

Figure 4.11 shows the responses to our post-study survey. Overall, users found using sense.us both enjoyable and informative. In the exit survey, the question "Did you enjoy using the system?" received an average rating of 4.0/5.0, with stdev. 0.52. The question "Did you learn something interesting using the system?" received an average rating of 4.2/5.0, $SD = 0.65$. Users also provided usability remarks and suggested additional features. The next section addresses a number of these suggestions.

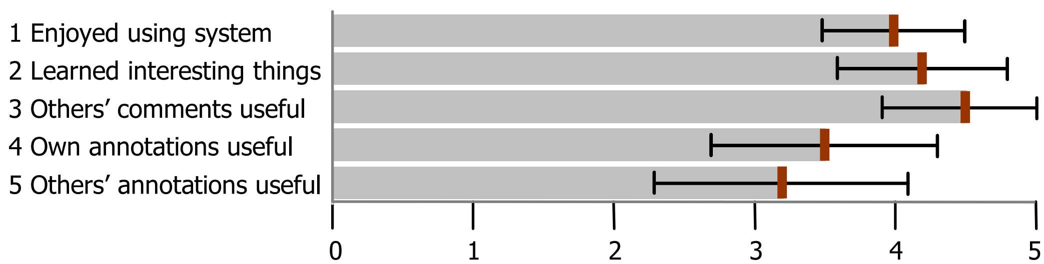


Figure 4.11. Results of post-study survey. Error bars indicate the standard deviation.

4.4 Discussion

The usage we observed echoed some of the earlier findings about social data analysis [186]. In particular, we saw cascading conversation threads in which users asked questions, stated hypotheses, and proposed explanations, all in a social context. A significant number of comments were playful or joking, as were a few graphical annotations. Wattenberg and Kriss have hypothesized that one of the spurs to social data analysis is a situation in which each user brought a unique perspective to bear [186]. In the case of job data, this unique perspective was the set of professions of friends and family of the user. We did indeed see people exploring in this fashion, covering a broad set of the data.

On the other hand, we observed a somewhat more businesslike tone to analysis than was seen previously [186]. This tone was likely in part due to the corporate and laboratory settings of use. The presence of an observer in the lab likely also influenced results, though many users reported they had fun conducting social data analysis.

In the next sections, as we describe research directions suggested by reactions to sense.us, a number of which are subsequently addressed in later chapters.

4.4.1 Embedded Social Navigation Cues

The doubly-linked discussion model was probably the most effective and well-liked novel feature of sense.us. If there was any frustration with this feature, it was that users had to navigate to a precise location to see related comments. This shortcoming, coupled with the high rate of within-applet navigation (Figure 4.9), raises an intriguing question for future research: would it be helpful to embed social navigation cues in the visualization or interface widgets themselves, and if so, how best to do this?

For example, a dynamic query widget used to filter the visualization might include visual cues of how many people have visited or commented on the views reachable using the widget, providing information scent by which the user can purposefully navigate towards either popular or unpopular views. Such widgets could aid the discovery of interesting trends that simply had not yet been seen. In our context, one might imagine a slider—controlling a view parameter—with marks indicating the presence of comments at specific parameter values. One can devise similar techniques for other widgets. The next chapter on *Scented Widgets* explores this possibility.

4.4.2 Enhanced Commentary and Navigation Features

A second approach, suggested by many users, is to show commentary related, though not directly attached to, the current view. Requested features include showing comments from other views that contain links to the current view (*"trackbacks"*), and related commentary on *"near-by"* or *"similar"* views. The latter could help alleviate cross-talk. Along these lines, there are appealing possibilities for generalizing the notion of view indexing, for example, suggesting conversations on views deemed semantically similar to the current view. This would require an index of visualization state providing not just equality comparisons, but distance measures. Such a retrieval model might be used to provide additional benefits, such as general searchability and data-aware auto-complete mechanisms.

Users have also suggested using visitation data or explicit ratings of "interestingness" to suggest views of potential interest. Others suggested supporting keyword tagging of comments [76, 130] and mining usage data. For example, both manual and automated tagging of questions or action items could be used to help direct collaborative effort.

The scope of comment visibility is a larger issue that affects all discussion models. What happens when the amount of discussion becomes untenably large, or users don't want their activity exposed to everyone? The ability to form groups and limit comment visibility to group members was one feature requested by users to support privacy and make discussion-following both more relevant and tractable.

4.4.3 Annotations

The graphical annotations saw significant usage, despite mixed reactions from users. Though they were used primarily for pointing, many users did not always find them necessary for disambiguation. We expect that the value of annotations varies significantly depending on the type of visualization being referenced. Regardless, annotations were used regularly for pointing and sometimes for socializing.

If annotations prove helpful, a second challenge would be to extend them to cover dynamic or evolving data sets. The decoupled nature of view annotations can prove problematic when the underlying data changes. Similar problems have been investigated in the context of document annotation [28]. CHAPTER 6 explores "data-aware" annotations that translate user selections into declarative queries over the underlying data, allowing annotations to be applied to time-varying data and different visual encodings.

4.4.4 Bookmark Trails and Story-Telling

Although individual usage varied substantially, most lab study users (87.5%) did use the bookmark trails, which proved essential for comments that included multiple views. Multiple users remarked on the usefulness of the bookmark trails and wanted to more easily share trails as first class objects. At times, users were frustrated when following multiple links in a comment, as the original comment would disappear when a new view was loaded, requiring use of the back button to perform "hub-and-spoke" browsing. In response, users suggested adding a dedicated "presentation" mode to facilitate tours and storytelling. Along these lines, CHAPTER 7 of this thesis introduces an automated graphical history viewer for visual analysis that supports the generation of presentations and CHAPTER 8 introduces animation techniques that better depict the relationship between subsequent states in a presentation of an analysis session.

4.5 Summary

In this chapter, we investigated mechanisms supporting asynchronous collaboration around interactive information visualization, seeking to more tightly tie the perceptual and cognitive benefits of visualization to social processes of sensemaking. To do so, we implemented a collaborative data visualization site, sense.us. We then observed usage of the site, in order to better understand the social dynamics surrounding collective use of visualizations as well as the efficacy of the particular features of the site.

The features of the site—doubly-linked discussions, bookmark trails, graphical annotations, and comment listings—were all exploited by users. The doubly-linked discussions successfully enabled users to fluidly transfer attention between visualization and commentary. Bookmark trails and graphical annotations were also well used, enabling tours through multiple views and pointing to items of interest, respectively. Finally, users routinely alternated between data-driven exploration directly within the visualization and social navigation through comment listings and user profiles to discover new views of interest.

5 Scented Widgets: Improving Navigation Cues with Embedded Visualizations

The success of an interactive visualization depends not only on the visual encodings, but also on the mechanisms for navigating the visualized information space. These navigational mechanisms can take many forms, including panning and zooming, text queries, and dynamic query widgets. However, effective navigation relies on more than input techniques alone; appropriate visual cues can aid navigation by guiding and refining an exploration. In this chapter, we develop enhanced navigation aids suggested by our design considerations (CHAPTER 3) and studies of sense.us (CHAPTER 4) and use them to provide social navigation cues designed to facilitate social data analysis.

Both psychological and sociological considerations suggest approaches for improving navigation cues. Pirolli and Card's [142] information foraging theory models the cost structure of human information gathering analogously to that of animals foraging for food. One result of this theory is the concept of information scent—a user's "(imperfect) perception of the value, cost, or access path of information sources obtained from proximal cues." Improving information scent through better proximal cues lowers the costs of information foraging and improves information access.

While effective information scent cues may be based upon the underlying information content (e.g., when the text in a web hyperlink describes the content of the linked document, it serves as a scent), others may involve various forms of metadata, including usage patterns. In the physical world, we often navigate in response to the activity of others. When a crowd forms we may join in to see what the source of interest is. Alternatively, we may intentionally avoid crowds or well-worn

Previously published by Wesley Willett, the author, and Maneesh Agrawala in [191].

thoroughfares, taking "the road less travelled" to uncover lesser-known places of interest. In the context of information spaces, such social navigation [59] can direct our attention to hot spots of interest or to under-explored regions.

The previous chapter described the sense.us system, with which groups of users perform visual data analysis by authoring comments and annotations around visualizations. In usage studies we found that users fluidly switch between data-centric analysis and social navigation. After exhausting a line of inquiry, participants mine listings of comments left by other users to find new views of potential interest and to understand which areas have been explored. However, without explicit social navigation cues, users must continuously switch between the visualization and a separate list of comments.

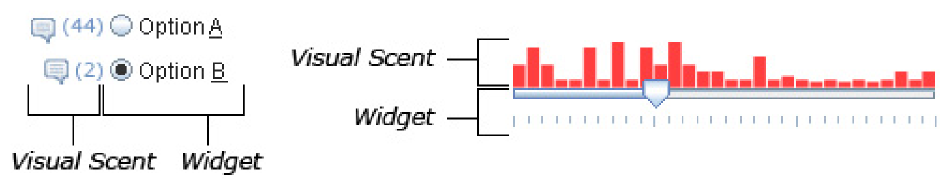


Figure 5.1. Widgets with visual information scent cues. Left: Radio buttons with comment counts. **Right**: Histogram slider with data totals.

In this chapter, we show that social activity cues can improve such social data analysis by enabling social navigation within the analytic environment of the visualization. We introduce *scented widgets*: enhanced user interface widgets with embedded visualizations that provide information scent cues for navigating information spaces (see Figure 5.1 for examples). We propose design guidelines for adding embedded visualizations to common user interface controls such as radio buttons, sliders, and combo boxes. We then present a Java-based toolkit-level software framework that enables developers to add scented widgets to their user interfaces and bind the widgets to backing data sources. This framework allows visual navigation aids to be added to existing applications with minimal modifications to application source code. We also provide results from an initial evaluation of scented widgets in a social data analysis application. The results show that using scented widgets to provide social navigation cues help users make up to twice as many unique discoveries in unfamiliar datasets, but that these benefits equalize as users become more familiar with the data.

5.1 Related Work on Navigation Cues

Researchers have proposed numerous navigation mechanisms to improve human-information interaction. In such interfaces, users may navigate along both spatial and semantic data dimensions. Examples of spatial navigation include maps and virtual worlds; examples of semantic navigation include web hyperlinks and dynamic query filters [2]. Navigation cues may be derived from the information content being explored (e.g., data distribution or landmarks) or from metadata, such as accumulated usage patterns. This last scenario is an example of social navigation [62], in which aggregated activity patterns are presented to promote awareness of other users' actions within the information space. All such navigation cues provide proximal information that helps users stay oriented and gauge the relevance of distal information content.

One class of navigation cues seeks to facilitate spatial browsing, such as zoomable 2D canvases. Overview displays are one common approach, while other approaches embed navigation cues directly in focal display regions. For example, Halo [6] and City Lights [200] use marks near the periphery of a display to provide information about the relative position of off-screen elements.

Semantic navigation examples provide cues based on the information content itself. In visualization, histogram sliders [56] and other data-driven variants [65] facilitate navigation to data regions of interest by summarizing the data distribution queried by the slider. On web pages, hyperlink text usually offers navigation cues about the content of the link target. This is the reason that human web surfers and modern web search indices rely on link text [23, 139]. Olston and Chi's ScentTrails system [139] facilitates search and browsing of web sites by scoring documents in response to a text query and then enlarging hyperlink text to indicate paths to highly ranked documents. ScentTrails outperforms both searching and browsing alone in information-seeking tasks.

Another strategy is to provide information scent cues based on metadata. For example, social navigation is often based on displaying aggregated activity patterns. Blogs and discussion forums regularly include the number of posted comments in the link text of hyperlinks to discussions, while the del.icio.us social bookmarking service encodes the number of users who share a web bookmark in gradated red backgrounds for link text. Hill et al. [98] explore the use of social navigation cues in a document editor, placing usage histograms within the scroll bar to indicate the prevalence of reading and editing activity throughout the document. Similarly, Björk and Redström [19] use color marks to indicate edits and search results along all edges of document frames. In the domain of collaborative visualization, Wattenberg and Kriss [186] gray-out visited regions of a visualization to provide "anti-social navigation" cues to promote analysis of unexplored regions.

Our work generalizes techniques such as histogram sliders and Hill et al.'s read and edit wear, providing design considerations and a toolkit-level framework for embedding navigation cues in a variety of interface widgets. We contribute a general framework providing both data- and metadata-driven visual cues for navigating semantic dimensions in an information space.

Though not focused on navigation cues, a few additional projects share commonalities with scented widgets. Baudisch et al.'s Phosphor [7] design provides real-time collaboration cues by using afterglow effects to highlight widget usage. Hill and Gutwin's Multi-User Awareness UI [97] provides toolkit-level widget support for synchronous collaboration, such that users can see in real-time which interface widgets collaborators are using. Our scented widgets framework also provides a toolkit-level augmented widget suite, but one targeted at visual navigation cues rather than synchronous activity awareness

5.2 Design Considerations for Scented Widgets

In designing a framework for encoding scent within widgets we consider; (1) the types of information metrics that can serve as navigation cues in scented widgets, (2) the matching of these encodings with the navigation models of the set of standard widgets, (3) the kinds of visual encodings used to convey this data, and (4) the modification of the standard widgets to accommodate scenting.

5.2.1 Information Scent Metrics

The first step in providing navigation cues is selecting the data source from which the cues will be derived. While the appropriate data source usually depends on the specifics of the application, several kinds of data and metadata can be useful aids for navigation. One approach is to derive metrics directly from the information content. For example, a simple metric for interactive visualization is the number of visible data elements in each application state. This metric provides a sense of the density of data across the information space. More complicated metrics can be computed from the data itself, and may involve input from the user. Users might type in queries, as in ScentTrails [139], and subsequently use scenting cues that indicate relevance scores. Alternatively, advanced users might use an expression language to enter in their own calculations over a visualized data set.

Social activity metrics are another potential data source, providing cues for social navigation. Interactive visualization applications such as sense.us capture a number of social activity metrics that are typically invisible to users, but which could serve as valuable navigation cues. For example, displaying the number of visits to a view, comments on a view, or edits of a view, could guide users towards the relevant or most

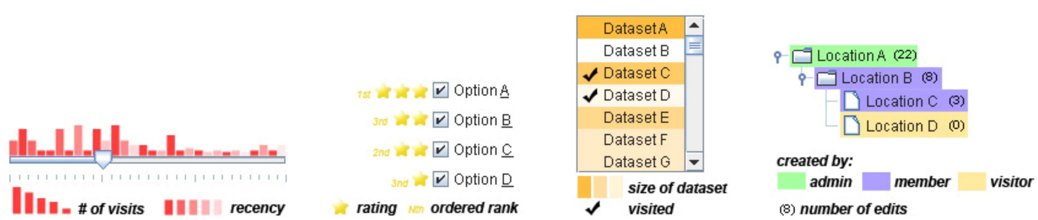


Figure 5.2. Examples of several scent encodings. From left to right: 1. A slider with visit totals encoded as a bar chart with recency encoded as opacity. 2. Checkboxes with star rankings encoded using icons and rank values displayed as text. 3. A list box with dataset sizes encoded using opacity and a visited/not visited value encoded using an icon. 4. A tree with author categories encoded using hue and edit totals encoded as text.

interesting views. Similarly, indicating the author of a comment or an edit could help users navigate to useful views. Temporal data regarding changes in any of these measures (e.g. recency or frequency information) are also candidates for display, as is location-based metadata. Our approach is premised on the notion that surfacing these sorts of activity metrics facilitates navigation.

5.2.2 Navigation and the Display of Visual Scent

Scent cues are specifically designed to aid navigation. Therefore scent cues should only be applied to interface elements that provide a way to navigate (i.e. change views) within the application. Moreover, widgets that represent a single navigation choice, such as buttons, should display only one scent value, while widgets such as combo boxes and sliders that offer multiple navigation choices should include scent cues corresponding to each navigational choice.

5.2.3 Visual Encodings

Scented widgets embed a visualization of information scent metrics within a standard interface widget such as a slider, button, or combo box. Standard widgets are usually designed to fit within a small screen-space and a goal of our scented widgets designs is to add information to these widgets without adversely impacting user interface design.

We begin by considering a basic language of visual encodings for data. These include visual variables such as position, size, angle, color, and shape [16, 35, 124]. As noted by Cleveland [50] and Mackinlay [124], some encodings are more suitable than others for displaying different types of information. For example, position encodings are more accurate than length encodings for quantitative data, which in turn are more accurate than area encodings. For nominal data, color encodings are better than position.

We can leverage these encodings in two distinct ways to convey information on or within a widget. One approach is to directly alter the attributes of the widgets that correspond to a given encoding. For example, a button's color could be based on the number of times the application state it leads to has been manipulated by users.

Because widget sizes, shapes, and layouts are typically fixed, we can apply only a few of the visual variables (hue, saturation, lightness, and texture) directly to the widgets without disrupting the layout and impeding usability. However, visual variables such as position and length are typically more effective for displaying quantitative data. Therefore, as a second option, we can embed small visualizations that support these encodings into the widgets. Examples include bar charts over a slider (e.g., Figure 5.1, [8]) and small, word-sized line charts (similar to Tufte's sparklines [174]) integrated with widget text.

5.2.4 Modifying Widgets

Based on these observations, we have selected seven different scent encodings to support within our framework. Direct encodings include the hue, saturation, and lightness properties of the widget. We also include four types of embedded visualizations: inset text, shape/icon, bar chart, and line chart. The examples in Figure 5.2 show several of these encodings applied to standard Swing widgets, while Table 1 describes each supported encoding type. We avoid encoding scent onto a widget's existing text labels, as label formatting is often modified by the application to convey highlighting, selection, keyboard shortcut combinations, and other information.

5.2.5 Design Guidelines and Feature Requirements

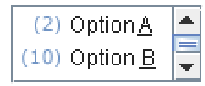
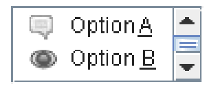
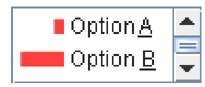
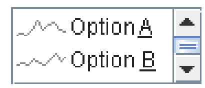
Through inspection of the design space of widgets and study of related work [124, 160], we have developed a set of guidelines for the design of scented widgets.

Scent Encoding Guidelines

Modes of scenting should maximize comparability and consistency across the interface.

More specifically: All widgets visualizing the same scent data should use matching visual encodings. Rationale: Encoding the same data differently across widgets complicates visual comparison.

Table 5.1. Scent encodings supported by scented widgets

Name	Description	Example
Hue	Varies the hue of the widget (or of a visualization embedded in it)	
Saturation	Varies the saturation of the widget (or of a visualization embedded in it)	
Opacity	Varies the saturation of the widget (or of a visualization embedded in it)	
Text	Inserts one or more small text figures into the widget	
Icon	Inserts one or more small icons into the widget.	
Bar Chart	Inserts one or more small bar chart visualizations into the widget	
Line Chart	Inserts one or more small line chart visualizations into the widget	

Modes of encoding should reinforce semantic relationships between the widget scent and encodings in the application. Rationale: Conflict between the scent and the other parts of the application will lessen the effectiveness of both. For example, avoid encoding scent using color if the application already uses color to display unrelated information.

Visualizations showing the same scent data should be scaled identically (e.g. linearly, logarithmically, etc.) across all widgets. Rationale: Scaling the same type of data differently across widgets undermines accurate visual comparison.

Modes of encoding should respect existing interface conventions. Rationale: User interface conventions tend to be well established and accepted by users, so scenting cues should not conflict with them. For example, a scent encoding should not repurpose text or icons commonly used elsewhere in the interface to encode unrelated data.

Encodings which make some elements markedly more salient than others, such as opacity, should be used with discretion. Rationale: If a widget is more salient than those around it, it is more likely to be used for navigation than its neighbors. Depending on the application, such enhancement may or may not be a desirable result.

Layout Guidelines

Interfaces should be laid out so that scented widgets are sufficiently proximal to allow comparisons between them. Rationale: Proximity aids judgments of position-based encodings and visual scent is most easily compared when graphic marks are adjacent.

Scented widgets should be grouped, sized, aligned, and oriented similarly in order to provide common axes on which to compare scent. Rationale: Without common axes it is difficult to compare marks across scented widgets, even if they show the same type of data.

Composition Guidelines

The overall number and type of scented widgets in a given interface should be small enough to allow easy comparison and visual tracking of changes. Rationale: The inclusion of too many scented widgets (and thus too many scent indicators) is likely to pollute the view, increasing cognitive load and making use more difficult.

Widgets should include identifiers (icons, tooltips, text, or a legend) that indicate what the scent cues correspond to. Rationale: It may be difficult for new users to discern what the cues indicate.

Many of these guidelines are addressed by our implementation. We deal with concerns about cross-widget consistency by grouping similarly-scented widgets and encoding them according to a shared configuration. While the distribution and layout of widgets in a user interface is clearly within the purview of developers, sizing, alignment and scaling can be fixed consistently across these groups.

5.3 Implementation of Scented Widgets

Using the preceding design analysis as a guide, our scented widgets framework provides toolkit-level support with which developers can quickly add visual scent cues to existing applications without writing a substantial amount of new code. The framework is implemented using Java Swing and takes advantage of the platform's Pluggable Look and Feel functionality, which allows the appearance of a wide range of standard interface widgets to be changed at runtime. In this section we discuss the design decisions made in our implementation, with the goal of providing guidance for developers building their own scented widget systems.

5.3.1 Rendering and Interaction

When implementing scented widgets, rendering and interacting with individual widgets is a primary concern. Ideally, the components for rendering visual scent cues should be implemented in a modular fashion, such that application developers can reuse them across disparate widget types.

A number of implementation paths are possible. One might implement custom widgets from scratch, but this approach involves re-implementing basic rendering and interaction mechanisms. Another strategy is to subclass existing widgets, overriding rendering and input handling techniques as needed. This approach is more efficient, requiring only targeted changes to widget behavior, but can still prove problematic. For example, restrictive access permission to members of the widget parent class may make it difficult to access parts of the widget state. Furthermore, both approaches require that developers explicitly use custom widget types in applications. Retro-fitting an existing application to use scented widgets then requires updating every widget definition in the application.

To avoid these limitations, we use Java's Pluggable Look and Feel layer to create a custom collection of scented widgets that can be installed without changing existing UI code. We extend Swing's default "Metal" Look and Feel and adjust the internal layouts of the Swing widgets to accommodate the embedded scent visualizations. Scented widgets also intercept user interface events as needed (e.g., allowing a mouse hover over an embedded visualization to trigger a custom tooltip for that graphic). Finally, we provide configurable renderers that are responsible for drawing the embedded visualizations. We use these scent renderer objects across the full widget set, promoting code reuse and ensuring consistent scent appearance in each widget type. Table 5.1 illustrates the encodings currently supported by our renderer.

5.3.2 Scent Configuration and Widget Groups

To map a backing data set onto visual scent cues, developers must provide a visual specification. For a group of related widgets, the visual specification indicates which data values to visualize and how to visualize them. Visual specifications define the names and data types of the variables to display in each scented widget and provide specific details about how the scent should be displayed. Specifications also maintain default values for encodings that are not determined by a variable. For example, a developer encoding a variable as a bar chart might specify default hue, saturation, or lightness values for the bars or add custom legend text or graphics.

In many cases, multiple widgets will show data from the same source, and the visualizations should be consistent across this group. Moreover, manipulation of a widget can alter the application state and require updates to the scenting of all related widgets. Our framework models these dependencies in a widget group abstraction that monitors all widgets that should update in response to one another. Upon creation, developers associate a widget group with a visual specification and a backing data source. When a widget is added to the group, our framework automatically configures the widget to use the group's specification, ensuring consistent scent cues. The widget group then analyzes the widget to determine the set of potential values it can take. For example, a button can only be pressed or unpressed, whereas a slider can potentially take one of a multitude of values. The framework uses this set of potential values to determine the possible application states reachable at any given time. Next, the widget group adds listeners to the widget, allowing updates to both the widget's selection state and underlying data model to be processed by the framework.

5.3.3 Data Management

To track the current state of the application, every widget group models state as a set of name-value pairs for each widget in the group. When a widget value is changed (e.g., moving a slider, selecting a radio button, etc.), the widget updates its state pair. In some cases changing the value of a widget can affect the way other widgets in the application work. Thus, every time a widget changes state, the widget group requests new scent data for all the other widgets in the group to update their scent values.

To populate scented widgets with data, developers must implement the data source interface, which provides scent data in response to queries. Scent queries consist of the current state, the visual specification, and a reference to the widget. The data source returns scent data—such as numbers, strings, or arbitrary Java objects—as sets of arrays for each variable defined in the visual specification. These arrays contain scent values for each state reachable using the widget under consideration. For quantitative and ordinal data, scent data objects can also provide a range over which the data will be scaled before rendering. Scaling may be linear or logarithmic, as configured in the visual specification.

Given the vast number of potential scent metrics, we expect that developers will build their own data source implementations that handle scent query requests in a domain-specific manner. However, our framework provides some tools that can help developers create custom data sources. For example, a caching layer caches query results and supports customizable replacement policies. Additionally, an SQL database helper aids developers in writing the code necessary to retrieve scent data from relational databases. The helper provides support for translating state objects and visual specification variables into SQL statements. A series of callbacks allow developers to customize the mapping between specified variable names and database column names and to generate custom database keys from widget values. The helper then handles all data transfer, packaging the results of queries into scent data instances.

5.3.4 Usage Example

The scented widgets API design is intended to allow developers to incorporate information scent cues into the widgets in their existing applications without substantial code revision. In the example given in Figures 5.3 and 5.4, we demonstrate how developers can use our framework to provide scenting on interface widgets.

First we create a VisualSpecification and assign a scenting variable to it (lines 2-3). The system uses the assigned variable name to query the DataSource. The QUANTITATIVE and BARCHART arguments specify the data type of the variable and the visual encoding. Since we do not provide any other configuration details, the system relies on default settings for the other parameters. In this case, the system scales the quantitative scent values it receives from the DataSource and encodes them using a default color scheme.

Next we access the global ScentRegistry (line 6) to create a WidgetGroup (line 10). The widgets in this group will be scented using the encodings given in our VisualSpecification, with data values drawn from a VisitDataSource object. The VisitDataSource is a custom database wrapper that implements the DataSource interface to provide visit data about each of the widget states. Finally, we create a standard Java Swing slider and list box (lines 14-15) and, using a single line of code for each one, we register them with the WidgetGroup (lines 16-17). Thus, the system will query scent data from the DataSource and supply it to the widgets, which in turn will render themselves using the scent-enabled custom Look and Feel. The system refreshes the scent cues on each member of a widget group whenever a change is made to another member.

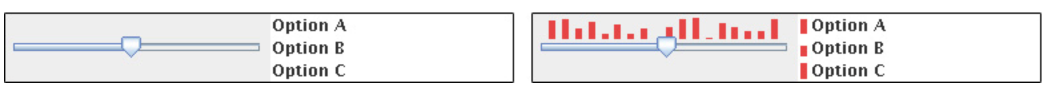


Figure 5.3. Widgets from the usage example, before and after scenting.

```
01  //Create the VisualSpecification and define the scent encoding
02  VisualSpecification myVspec = new VisualSpecification();
03  myVspec.addVariable("numVisits", ScentConstants.QUANTITATIVE,
04      ScentConstants.BARCHART, SwingConstants.VERTICAL);
05  //Get a ScentRegistry reference
06  ScentRegistry sr = ScentRegistry.getInstance();
07
08  //Create a WidgetGroup using the VisualSpecification and a data source
09  // defined by the developer which implements DataSource
10  sr.initWidgetGroup("myWidgetGroup", myVspec,
11      new CachedDataSource(new VisitDataSource()));
12  //Create and register widgets, providing a name for the widget and
13  // the name of the WidgetGroup to which it should belong
14  JSlider myJSlider = new JSlider(1,20);
15  JList myJList = new JList(new Object[] {"Option A","Option B","Option C"});
16  sr.register("myWidgetGroup", "sliderValue", myJSlider);
17  sr.register("myWidgetGroup", "listValue", myJList);
```

Figure 5.4. Sample code for the usage example of the Scented Widgets framework.

5.4 Applications

As a preliminary evaluation of our framework, we have built three prototype applications that demonstrate diverse use cases for adding visual scent cues to traditional widgets.

5.4.1 HomeFinder with Histogram Sliders

The first application is a re-implementation of the HomeFinder [192], a geographic scatter plot visualization of available housing that uses dynamic query widgets to filter the view. Figure 5.5 shows our version of the application visualizing San Francisco apartment listings automatically harvested from craigslist.org RSS feeds. Scented widgets show the number of available apartments across rental prices, neighborhoods, and number of bedrooms, providing an example of a data-driven scent metric. We used the prefuse toolkit [85] to provide the scatter plot and generate the query widgets, which we then registered as scented widgets. A custom data sourse provides scent data that summarizes data in the underlying prefuse data table. We created the widget's visual specification with just one variable, the number of available houses, and configured it to use linearly-scaled bar charts.

5.4.2 Collaborative Authoring with Activity Indicators

The next application is a collaborative text editor, in which multiple authors access a document to simultaneously edit it. An example of our prototype is shown in Figure 5.6. Each author is assigned a unique color to identify the text segments they have edited. A scented list widget shows all authors who have viewed the document and a line chart of authors' daily edits. The combined interface allows authors to assess both textual editing patterns and the temporal activity of editors. To implement the prototype, we built a custom data source which models editing activity over time. A listener registered with the text editor aggregates editing events and posts them to a server. The visual specification includes two visual variables, one for hue and one for the line chart.

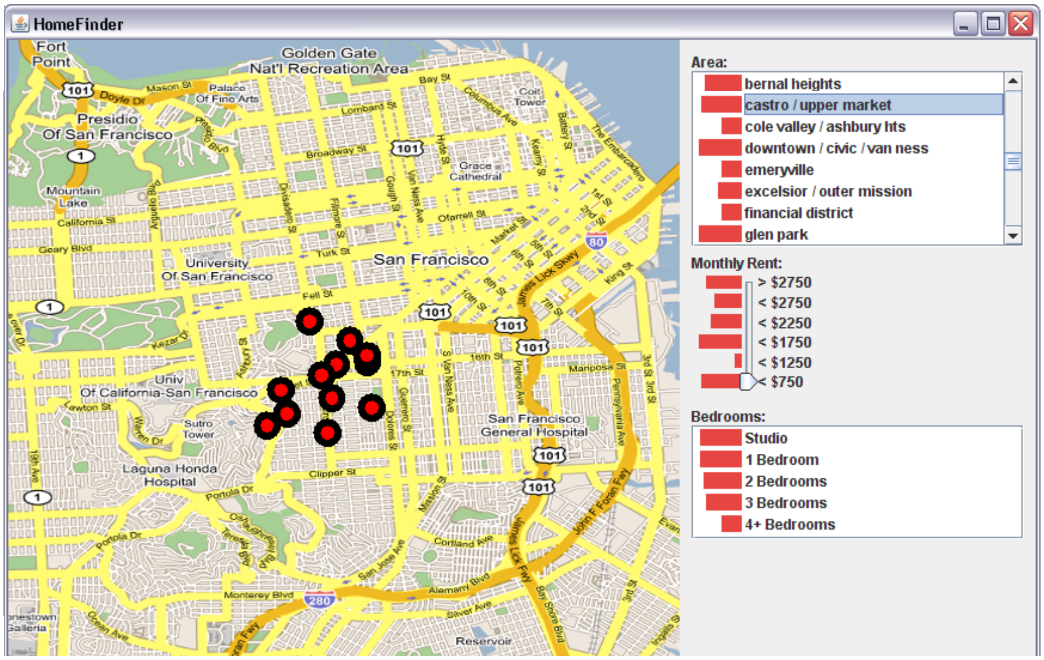


Figure 5.5. HomeFinder with histogram widgets. A scatter plot and scented query widgets show available apartments from craigslist.org.

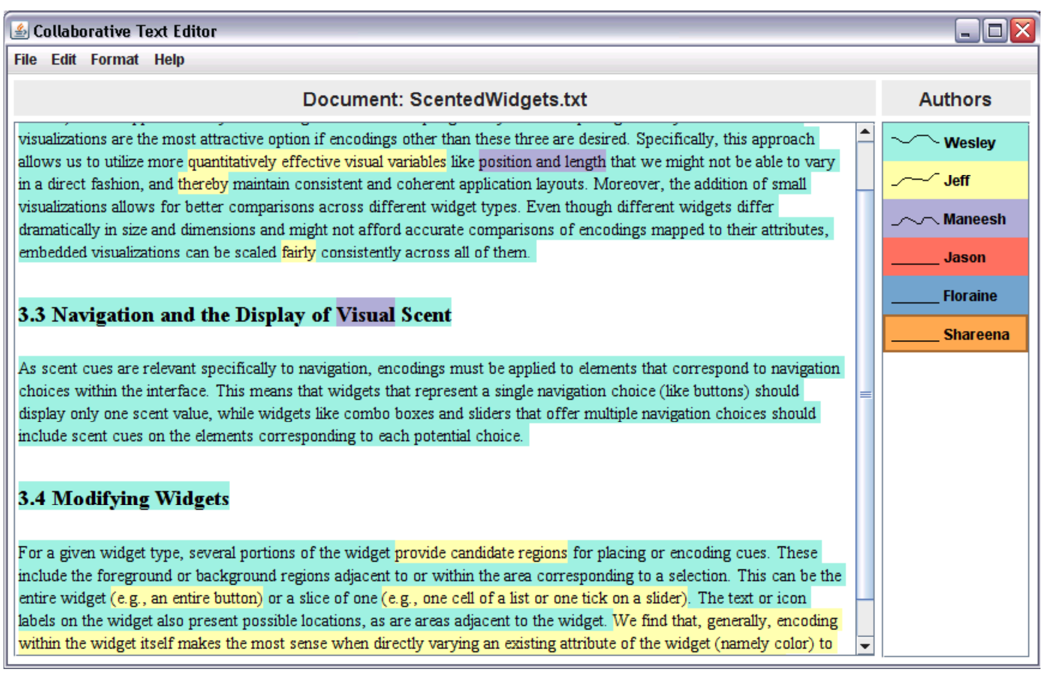


Figure 5.6. Collaborative text editor. A scented list widget identifies authors by color and displays a chart of editing activity over time.

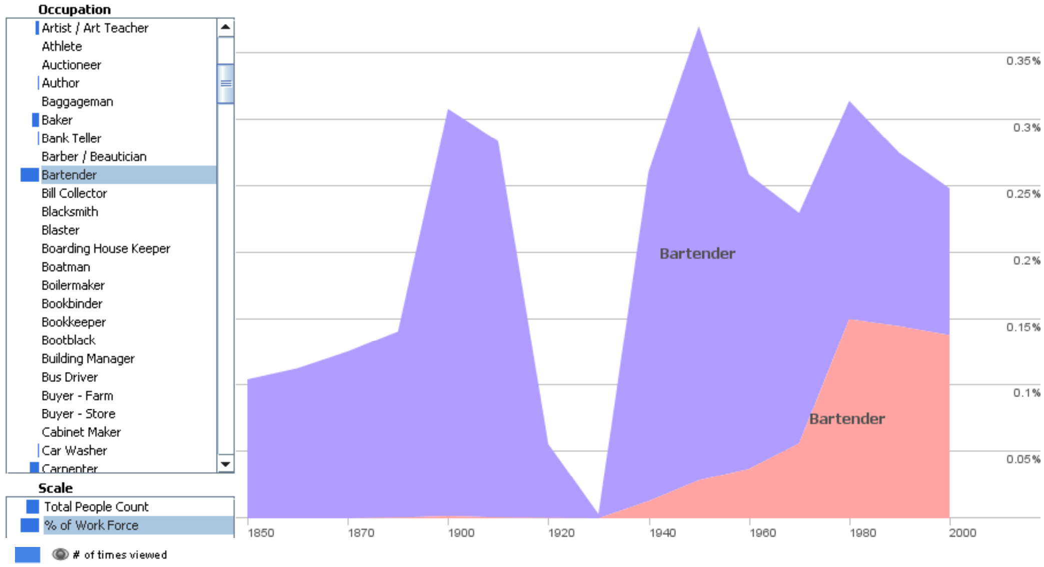


Figure 5.7. Social data analysis application with social navigation scent cues. Stacked time-series show the U.S. labor force, broken down by gender, from 1850-2000. The current view shows the percentage of the labor force that worked as Bartenders, with a drop during Prohibition. Scented Widgets are used in the dynamic query widgets to show visitation rates in all views reachable from the current view.

5.4.3 Social Data Analysis with Social Navigation

The third application uses scented widgets to add social navigation cues for collaborative data analysis to our sense.us system (CHAPTER 4). Figure 5.7 shows an interactive stacked area chart of the United States labor force from 1850-2000, broken down by occupation and gender. Dynamic query widgets on the left allow users to navigate to specific occupations and toggle normalization of the data (i.e., view relative percentages or total worker count). As users explore the data, the system records their visitation patterns to an external database. Scented widgets then visualize these visitation patterns, indicating both highly visited and neglected views. For example, by scanning the list widget and noting which elements do not have bar charts, one can see which data items have not yet been visited by an analyst.

The visual specification involves a single variable—the number of visits to each view—and specifies a bar chart encoding for the data. We used log scaling because the visitation data exhibited a power law distribution. We also built a variant of this application that shows the number of comments made on each view.

5.5 Evaluation of Social Navigation Cues

While prior work has explored various forms of data-driven scent cues [6, 19, 56, 65, 139, 200], less research attention has focused on visualizing social navigation cues [98, 186]. Therefore, we conducted a controlled experiment in which we asked subjects to perform information foraging tasks using the social data analysis application in Figure 5.7. We hypothesized that subjects would be more likely to revisit highly visited views using scented widgets, would make more unique discoveries using scented widgets, and would express a preference for scented widgets over traditional widgets. The study included twenty-eight participants (12 female, 16 male), all of whom were either graduate or undergraduate students, and were recruited through campus mailing lists. Participant ages ranged from 19 to 32 (M = 25.3, SD = 3.8).

5.5.1 Experiment Design

We asked subjects to find evidence either for or against specific hypotheses in a collaborative visualization of the United States labor force. We gave them an introductory tutorial to the system, and then asked them to complete three tasks. For each task, we presented subjects with one of the three following task hypotheses:

T1: Technology is costing jobs by making occupations obsolete.

T2: In the last half-century, women have joined the work force, but stereotypically male jobs remain almost entirely male.

T3: The number and variety of jobs directly related to the nation's food supply has diminished greatly since the 1800s.

For each task, we gave subjects 15 minutes to explore the data set and collect evidence relevant to the task hypothesis. The task hypotheses were intended to be of similar depth and diversity. We instructed subjects to make at least seven observations that provided evidence either for or against the current task hypothesis. At least two of the observations had to be unique findings on views not yet commented upon. Subjects were asked to note their observations by leaving new comments on the corresponding views.

For each task, we presented subjects with one of three scenting conditions. The conditions consisted of *no scent*, in which we used standard dynamic query widgets, *comment scent*, in which bar charts indicated the number of comments made on a view, and *visit scent*, in which bar charts indicated the number of prior visits to a view. To populate the interface with scent, we collected anonymized activity metrics from our study of the sense.us system (CHAPTER 4) and supplemented them with a small amount of manual seeding to balance the metrics across conditions. Subjects in the previous sense.us study used a similar visualization to freely explore the data. We removed comments posted on views that were not reachable in the current version of the visualization. Our seed data consisted of a total of 1096 visits and 172 comments distributed across 154 views. Both visits ($R^2 = 0.96$) and comments ($R^2 = 0.90$) exhibited a power law distribution, and so we scaled them logarithmically for display in the scented widgets.

The study employed a 3 (Task) x 3 (Scent) between-subjects design. We counter-balanced task and scent pairings and presentation order using a Latin Square. Subjects performed all tests in a laboratory environment using standard desktop PCs connected to a web server hosting the visualization and usage data. After completing the tasks, subjects filled out a survey that asked them to rate the scenting conditions on perceived utility and user experience.

5.5.2 Results: Revisitation

Our first hypothesis was that social navigation cues would increase the likelihood that users would visit views that others had visited previously. To test this hypothesis, we created three vectors, each representing the number of visits to each view in each scenting condition. We removed the starting overview from consideration, because users saw this view regardless of scenting condition. We then compared these visitation vectors to the visitation vector for the underlying activity measure used to

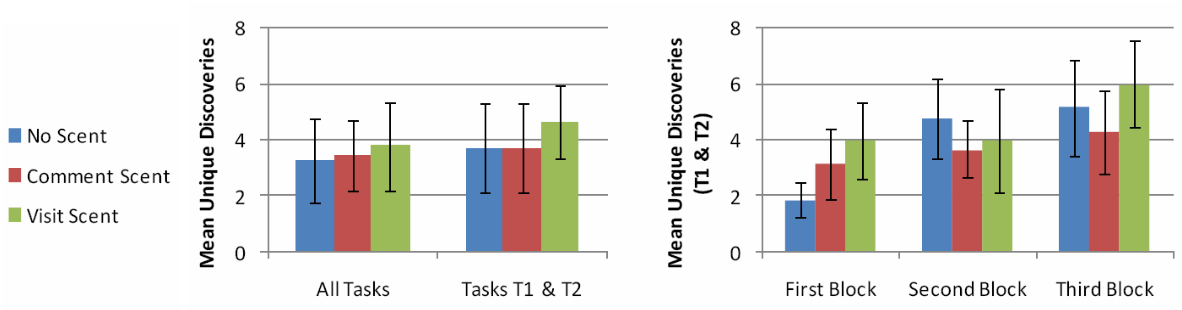


Figure 5.8. Experiment results. Left: Mean unique discoveries for all tasks and just tasks T1 and T2. **Right:** Mean unique discoveries for tasks T1 and T2, divided into blocks by order of presentation. The differences in the first block are statistically significant.

seed the scented widgets. Using Pearson's product-moment statistic, we found correlations of $r(493) = 0.200$ for *visit scent*, $r(493) = 0.217$ for *comment scent*, and $r(493) = 0.181$ for *no scent* ($p < 0.01$ in all cases). These results suggest that users in the *visit scent* and *comment scent* conditions were more likely to visit the same views that were visited in the seed data than users in the no scent condition. However, we note that the correlations are not very strong. We believe that the semantics of the tasks also affect visitation patterns and likely had an effect on these correlations.

5.5.3 Results: Unique Discoveries

Next, we analyzed the data to check if scented widgets help users make unique discoveries—relevant observations that have not yet been commented upon. Our hypotheses were that (a) scented conditions would have a higher occurrence of unique discoveries and (b) performance would improve over subsequent trials, regardless of the scenting condition, due to learning effects. To compute a metric of unique findings we collected all comments on visualization states that had no comments at the beginning of the trial. We manually walked through each of these comments, decrementing the tally for any comments that clearly had no bearing on the task hypothesis (e.g., jokes, unrelated questions, etc.). The result was a count of unique discoveries made in each task trial, across a total of 83 samples (due to a software glitch, one subject skipped a trial).

As shown in Figure 5.8, scenting provided limited benefits over all tasks. The data are not normally distributed and so we used non-parametric tests (the Kruskal-Wallis H

and Mann-Whitney *U* statistics) for statistical analysis. Based on these tests, the differences in unique discoveries between scenting conditions did not reach significance ($H(2) = 1.245$, $p = 0.537$).

However, there was a significant main effect for task hypothesis ($H(2) = 11.154$, $p = 0.004$). Pairwise comparisons using Mann-Whitney tests found that unique discovery counts for task hypotheses T1 (technology, $M = 4.2$, $SD = 2.4$) and T2 (gender, $M = 4.3$, $SD = 2.4$) were not significantly different ($p = 0.456$), but that both were significantly different ($p = 0.008$ and $p = 0.002$, respectively) from T3 (food, $M = 2.6$, $SD = 1.3$) ($p = 0.008$ and $p = 0.002$, respectively). Looking at the distribution of unique discoveries revealed that T3 netted substantially fewer comments. Examining the data, we found that a lower number of views were relevant to this task hypothesis and thus there was a limit on the number of possible unique findings. Subjects commented on only 25 unique views in T3, compared to 101 in T1 and 111 in T2.

We then analyzed the data according to the order in which the tasks were performed and found a significant main effect for task ordering ($H(2) = 6.341$, $p = 0.042$), indicating learning effects. The number of unique discoveries increases monotonically with practice, with significant differences between the first ($M = 3.0$, $SD = 1.7$) and subsequent blocks ($M = 3.6$, $SD = 2.1$ and $M = 4.4$, $SD = 2.6$). We then looked at the effects of scent within each block. Based on our earlier task analysis, we omitted the trials in T3. In the first block of trials, *visit scent* ($M = 4.1$, $SD = 1.6$) averaged 2.2 times more unique findings than *no scent* ($M = 1.9$, $SD = 0.4$) and *comment scent* ($M = 3.6$, $SD = 2.2$) averaged 1.7 times more. These differences were significant ($H(2) = 6.613$, $p = 0.037$). Pairwise comparisons found that *visit scent* resulted in significantly more unique findings than *no scent* ($p = 0.029$). The difference between *comment scent* and *no scent* failed to reach significance ($p = 0.053$), as did the difference between the two scenting conditions ($p = 0.281$). Analyses for the second and third blocks of tasks found no significant effects for scent ($H(2) = 0.45$, $p = 0.799$ and $H(2) = 1.338$, $p = 0.512$).

5.5.4 Results: User Preferences

We analyzed survey responses and found that users significantly preferred both scented conditions to the non-scented condition across the board (Table 5.2): for finding undiscovered views, for finding discovered views, for finding interesting views more quickly, and in terms of enjoyment. We conducted a one-way ANOVA for each of these questions; each found a significant effect ($F(2,78) \geq 7.402$, $p < 0.002$ in all cases). In each case, we performed post-hoc comparisons using Fisher's LSD test and found significant differences at the 0.05 level between the scented and non-scented conditions, but found no significant difference between the two scented conditions. Furthermore, users did not find either scenting condition to be cluttered or disruptive ($M = 1.6/5$, $SD = 1.0$ for both), and rated both about equally helpful overall ($M = 3.7/5$, $SD = 0.9$ for both). Users were evenly split between the scented conditions as to which condition was their favorite (14 comment, 12 visit, 1 no scent, 1 abstention), and the *no scent* condition was consistently ranked as the least favorite (24 no scent, 2 comment, 1 visit, 1 abstention).

The few complaints about scented widgets were largely related to users wanting the widgets to display different kinds of information. Five subjects expressed interest in toggling between multiple types of scenting information. One subject who interacted with our social data analysis application also voiced discomfort with the inability to turn off scent indicators, stating that she preferred to explore without being influenced by the browsing paths of previous users.

Table 5.2: User survey results. All ratings are on a 5 point scale.

	Visits		*Comments*		*No Scent*	
Survey Ratings	M	*SD*	M	*SD*	M	*SD*
Finding undiscovered views	4.1	*0.9*	4.2	*0.9*	1.7	*1.0*
Finding discovered views	4.1	*1.1*	4.2	*1.0*	1.9	*1.3*
Finding interesting views	3.5	*1.0*	3.6	*1.0*	2.6	*1.1*
How enjoyable	4.1	*0.7*	4.1	*0.7*	3.3	*1.2*

5.5.5 Discussion

The results suggest that subjects found scent useful for navigating the data when it was new to them, but as they learned the data, they relied on scent less. As their familiarity with the data increased, subjects may have transferred from social to semantic navigation of the data. Some caution is warranted in this claim, however, as we found advantages for scent after removing T3 from consideration. On the other hand, we only asked subjects to find a minimum of two unique discoveries, and so our results may be conservative. If users were asked to maximize unique discoveries, the differences between scented conditions might become stronger. As it stands, the results suggest that scenting increases unique discoveries in unfamiliar data even when unique discoveries are not the primary concern.

The reduced impact of social navigation cues over time seems plausible given the limited complexity of the data set—it is not complicated, nor particularly large. The finding also has a nice intuitive analogue; in many tasks social navigation is unnecessary after one becomes familiar with one's environment. A resulting hypothesis is that social navigation cues assist unfamiliar users in becoming oriented. Another hypothesis is that social navigation cues become increasingly useful for larger data sets as more time is needed to become familiar with the data. We leave further investigation of these hypotheses to future work.

It is also possible that earlier exposure to scent cues was partly responsible for the decreasing reliance on social cues we observed. All subjects that encountered the no scent condition in later blocks had already been exposed to at least one of the sets of scenting data. More careful study is needed to assess if exposure to scent affects subsequent behavior in other conditions.

At first glance, the results seem to suggest that visit scent may be preferable to comment scent. Though visit and comment scent fare equally well in user preference ratings, visit scent results in more unique discoveries than comment scent. However, the differences between the two are not statistically significant. Still, there are reasons to suspect benefits for visit scent. One hypothesis is that uninteresting views may be visited but are unlikely to accrue comments, so visitation metrics provide cues absent in comment scent. Another hypothesis is that, because commented views are visited more than uncommented ones, high visitation rates may be a good indicator of commentary. Indeed, analyzing the recorded activity metrics finds the expected correlation between visitation and commenting ($r(154) = .603$, $p < 0.01$). Further study is needed to determine which social navigation cues are to be preferred. In response to both this uncertainty and user requests, we recommend supporting user controls over the display of visual scent cues.

Finally, it is worth reiterating that we primarily drew the activity metrics used in the study from general, unstructured exploration sessions. We were interested in determining if making such activity traces visible impacts analysis, as one can collect this data easily and unobtrusively. However, one could also collect activity metrics in a more structured fashion. If visitation and commenting data are associated with users' tasks or hypotheses, scented widgets could display scent data specific to the current task. However, task-specific scenting requires design mechanisms that allow task metadata to be associated with usage data in a lightweight fashion.

5.6 Future Work

Several limitations in the current system stand to be addressed in future work. One issue is widgets supporting multiple selections. In a multiple selection list box, a user can select one item from a list and then use a modifier key (typically shift or ctrl) to select additional items. As selecting a new item in the list in addition to the currently selected one leads to a different state than selecting only one, the number of potential states grows combinatorially. In such cases we can use lazy querying of scent data to alleviate resource concerns, but unresolved design issues remain. To handle multiple selections, scent can be updated not only when a widget value is changed, but also when a modifier key is depressed. Making scent displays modal solves some design issues, but requires further study.

Furthermore, while the framework supports a number of embedded visualizations, this set is by no means exhaustive. We could also use common visualizations such as pie charts, stacked bar charts, and density plots at a widget scale. We will support these examples in the future by extending the framework's scent renderer.

Finally, we have found that the most time-intensive part of applying scented widgets is implementing a data source. Further support for data management would reduce implementation time. Our SQL data source helper (Section 5.4.3) is one example, as it greatly speeds development when using a backing database. In future work we may provide toolkit support for other data sources of interest. For example, support for accessing data in visualization toolkits such as prefuse [85] could accelerate the creation of data-driven scented widgets.

5.7 Conclusion

In this chapter we introduced scented widgets, user interface components enhanced with embedded visualizations to aid information foraging. We proposed guidelines for incorporating small embedded visualizations and other visual cues into standard user interface designs. We then presented a toolkit-level framework for adding visual scenting cues to widgets in the Java Swing user interface toolkit. With a backing data source in place, our framework allows developers to quickly add visual navigation cues to existing applications with minimal changes to source code, typically with only a few additional lines of code.

6 Generalized Selection via Interactive Query Relaxation

This chapter explores the development of "data-aware" selection and annotation techniques supporting both individual and collaborative data analysis. As noted in CHAPTER 4, sense.us users drew graphical annotations to point to specific data points and regions—88.6% of annotations were specific to data items or regions. This observation suggests that the expressiveness of free-form view annotations might be sacrificed in favor of annotations that can be applied to dynamic data and across varied visual representations. Furthermore, we can exploit the structure of data-aware annotations to provide more advanced selection mechanisms. We also show how these techniques are applicable to contexts outside of interactive visualization, such as graphics editing.

Pointing to an item or region of interest is common in everyday communication because it grounds the subject of the conversation or action. In the physical world, people coordinate their gestures, gaze, and speech to indicate the objects under discussion [22, 46]. In graphical user interfaces, reference (or selection) remains of critical importance, but is realized through a more limited set of actions, such as clicking or lassoing items of interest. Most interfaces model selections as a simple collection of selected items. While this approach is simple to implement, it makes it difficult for users to specify higher level selection criteria.

Consider the visualizations of reported homicides in Los Angeles shown in Figures 6.1 and 6.5. Analysts collaborating around these visualizations might refer to regions or attributes of interest [46], such as "East L.A.", "homicides in the month of May", or

Previously published by the author, Maneesh Agrawala, and Wesley Willett in [92].

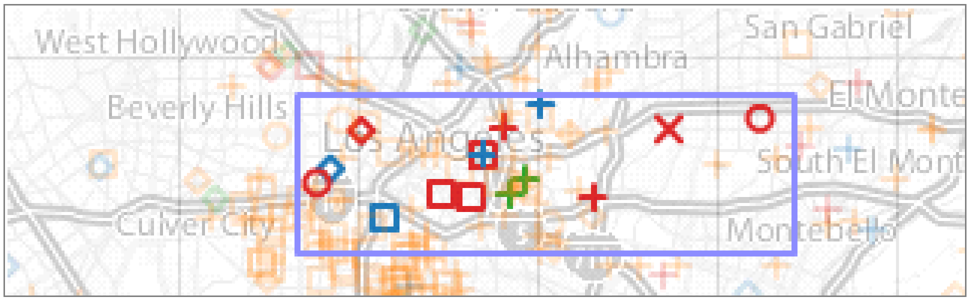


Figure 6.1. Map of reported homicides in Los Angeles, 2007. Color indicates cause of death, shape indicates the victim's race (the complete view is shown in Figure 6.5).

"all gunshot victims". Similarly, an analyst may point to an item and refer to "all items blue like this one," generalizing a reference based on properties of the item [21, 49].

One way to express such selections is to use a higher level query language such as SQL. For example, the SQL WHERE clause

```
('2007-05-01' ≤ date AND date ≤ '2007-05-31')
```

selects all homicides in the month of May 2007. The query encodes the structure of the selection declaratively, and applying the query results in a set of selected items. Systems such as DEVise [122], VQE [55], and Improvise [188] have recognized that maintaining query structure increases the expressiveness of visualization applications. Each of these systems provides graphical user interfaces for visually instantiating such general queries.

In this chapter, we also focus on building a selection interface that represents the selection as a declarative query over the attributes of interface objects or underlying data. Selection queries are modeled in a SQL -like query language and as in earlier systems (e.g., [55, 122, 138]) users create selection queries through direct manipulation. Our system visualizes the structure of the query and highlights the data or interface objects selected by the query. This formulation supports both selection of specific items and selections based on attributes of the data, which may vary over time.

The unique contribution of our work is to couple this query-based approach with generalization mechanisms that allow users to expand their selections based on an initial selection. This approach enables generalized selections such as "select all victims with the same age as this one" over both static and dynamic data. A query relaxation engine analyzes the attributes and network topology of the underlying data to automatically generate such selections. Users interactively select generalization criteria through a pop-up dialog (providing choice mediation [126]) or by repeatedly clicking to cycle through a set of alternate selections (providing repetition mediation in a manner similar to [156]).

We begin by reviewing related research on selection and reference. Next, we demonstrate our approach in both a data visualization system and a vector graphics drawing program and describe our system architecture. We then describe a user study of our selection techniques in a data visualization application, finding that subjects used query relaxation to more effectively author selections. Finally, we discuss future work and conclude.

6.1 Related Work on Reference and Interactive Querying

Our work on interactive query relaxation draws on research on direct manipulation selection techniques, including brushing, linking, and dynamic queries, as well as query relaxation techniques from the database community. We consider each in turn.

6.1.1 Selection Techniques and Reference

As described in CHAPTER 3, social psychologists have examined the basic prerequisites for communication, including reference: indicating items, people, and places to be discussed. Clark and Brennan [22, 46, 49] explain that spatial reference to visible objects and regions takes many forms. Such references may be general (e.g., "north by northwest"), definite (e.g., named entities), detailed (e.g., described by attributes, such as the "blue ball"), or deictic (e.g., pointing to an object and saying "that one"). People often apply multiple forms of reference in tandem, across modalities such as speech and gesture. These observations led us to include pointing and reference as an important design consideration for collaborative visual analysis. However, graphical interfaces rarely support such fluid and general forms of reference.

Clark [46] further divides deictic reference into pointing and placing. Pointing involves vectorial reference, such as pointing a finger or directing one's gaze to a specific item. Placing involves referencing a region of space imbued with a shared meaning, such as placing groceries on a counter to indicate items for purchase. To varying degrees, graphical interfaces use both forms of reference. Pointing actions using the mouse cursor are the most common. Placing also occurs, most notably in drag-and-drop, where drop targets have defined semantics. However, systems rarely support interactive specification of new "places." In interfaces, such places may include both spatial regions and abstract spaces defined by data attributes.

Our selection query and relaxation model enables interactive generalization of deictic references and specification of placing regions whose contents may change over time.

6.1.2 Dynamic Queries, Brushing, and Linking

Our work is closely related to selection techniques used in information visualization. Dynamic queries [2] typically take the form of widgets, such as range sliders, with which users incrementally filter visualizations. Brushing [8, 39, 127] enables selection through direct manipulation, typically via clicking, lassoing, or "painting" over items of interest.

One class of systems focuses on interactive selection within visualizations [2, 8, 68]. Martin and Ward [127] introduce multi-dimensional brushing, in which users can brush over projected data using 2D selection regions. Their system then considers the min/max values of the brushed points to compute a hypercube enclosing the brushed points in all dimensions. Hypercube construction is a specialized form of query relaxation: the items initially selected are extended to a full hypercube. Hochheiser and Shneiderman's time boxes [100] are dynamic queries that select all time-series that pass through brush regions; our approach generates a similar tool through relaxation of range queries.

Another class of systems uses visual query mechanisms to create visualizations and specify linking relationships for coordinated brushing across visualizations. Snap-

Together Visualization [137] implements linking using “primary key actions” that communicate the individual tuples that have been selected. Chen’s compound brushing [39] provides a graphical data-flow language enabling user-created brushing operations across visualizations.

Some of these systems explicitly represent the structure of selection queries. Linking in DEVise [122] is specified through chains of linked plots, specified in part with brushing gestures. The system maintains a declarative query structure to perform linking across views. Improvise [188] supports coordinated queries authored in an auxiliary tree editor for defining and linking visualizations. Derthick et al.’s Visual Query Environment (VQE) [55], provides a form-based interface for specifying intentional (declarative) queries coupled with brushable visualizations for specifying extensional queries (selection of specific items). Olston et al.’s VIQING [138] provides a direct manipulation interface for specifying queries; users rubber-band a set of visualized tuples to select them and they drag visual canvases on top of one another to join the underlying data. Polaris [167] allows specification of both queries and visualizations by dragging database column names from a list onto “shelves” for visual variables such as position, color, and shape.

Our work follows in the tradition of these systems, enabling users to interactively select visualized data or other interface objects. Similar to DEVise, VQE, and Improvise, our system uses a declarative query model that supports coordination and reuse across visualizations. Like VIQING, our system supports the creation of declarative selection queries through direct manipulation of the visualization itself. Most importantly, our system is unique in using query relaxation to interactively generalize selection queries.

6.1.3 Query Relaxation

The database community has developed query relaxation with the goal of creating “cooperative” databases that return information beyond that specified by a standard query. Query relaxation expands the query selection criteria to include additional relevant information, often by consulting a semantic model of the data domain. For example, a user seeking to travel from New York to Boston might query for morning flights. If no matches are found, relaxed queries might instead return train routes in the same time frame.

Gaasterland [69] introduces query relaxation techniques in deductive databases, using logic rules to specify legal relaxation constraints. Chu et al. provide query relaxation for relational databases [41] and XML documents [42], using type-abstraction hierarchies (hierarchical ontologies) to find semantically similar query results. Hierarchies can be hand-authored or generated by unsupervised clustering [41, 102].

Our work adapts query relaxation techniques to support generalized selection in graphical interfaces. As described in the following sections, our system supports configurable relaxation operations based on the attributes of interface items and relations between them. In most cases our system can produce a variety of relaxations from an initial query.

We provide interaction techniques that enable users to relax selection queries, cycle through the generated selections, and combine relaxed selections as desired. These techniques are modeled after mediation interfaces that disambiguate input among multiple alternatives (e.g., [103, 126, 156]). For example, text editors such as Microsoft Word set the cursor position on a single click, select a word on a double click, and select a paragraph on a third click. By cycling through the alternatives users can find the appropriate selection.

6.2 Example: Information Visualization

We have integrated our generalized selection and query relaxation techniques with flare (http://flare.prefuse.org), an open-source visualization toolkit for the Adobe Flash Player.

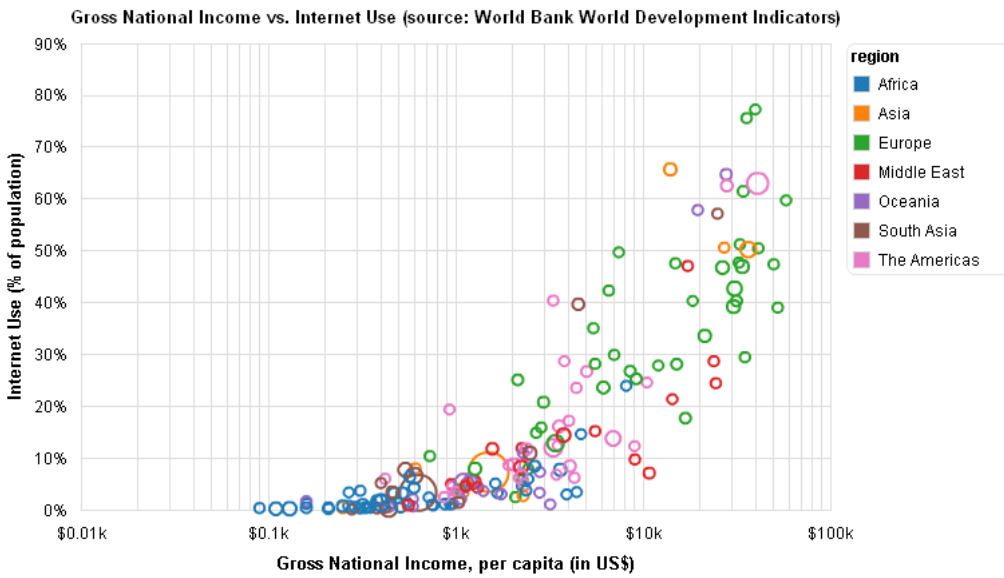


Figure 6.2. World development statistics. The visualization plots income against internet usage for the world's countries.

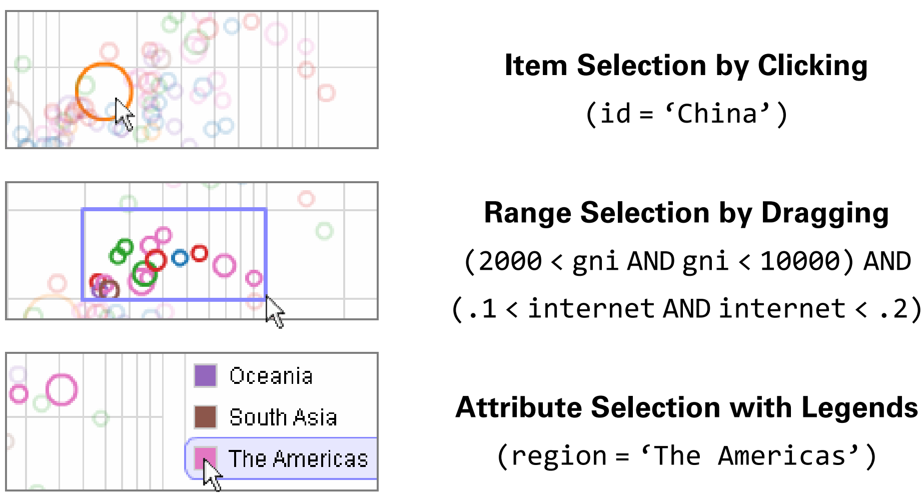


Figure 6.3. Basic selection operations and resulting query WHERE clauses. Images are close-ups from the plot in Figure 6.2.

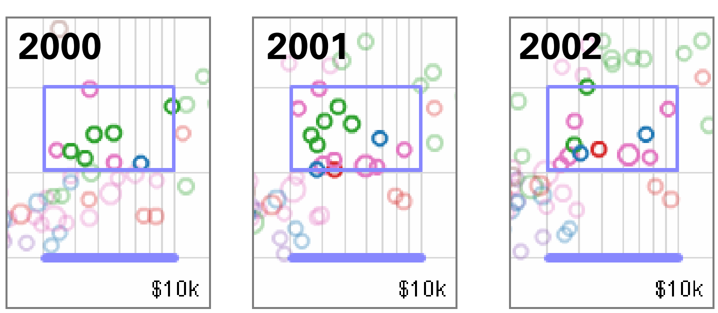


Figure 6.4. Selection over time-varying data. The selection updates dynamically as data points pass through the selection range. The sequence spans the years 2000-2002.

6.2.1 Basic Brushing and Selection

Our selection framework supports common brushing and dynamic query operations. Figure 6.2 is a scatter plot of development statistics from the World Bank [194], including per capita income, internet usage, and population data for the countries of the world (c.f., Gapminder [71]). As shown in Figure 6.3, our framework translates selection operations in the visualization into declarative queries over the visualized data. The selection query is in turn used to generate interactive range brushes and highlighting effects.

Users can click an item to select it (Figure 6.3, top), and optionally hold the shift key to select multiple items. Users can click and drag over the visualization to create a range query (Figure 6.3, middle). The range is persistent and users can reposition and resize the range as they desire. Users can also drag along the axis labels of the chart to create one-dimensional ranges. Additionally, all legends also function as dynamic query selectors (Figure 6.3, bottom). Users can select collections of items in discrete legends or select ranges in continuous legends, just as they can in the chart.

6.2.2 Selection Reuse

Because our system maintains the structure of the selection query, it can reapply the selection dynamically over streaming or time-varying data sets. Figure 6.4 illustrates countries passing in and out of a range selection as the data pages through each year.

Our selection system can also reapply queries across different visualizations of a data set and thereby supports linking across views. Figure 6.5a shows a visualization of homicides in Los Angeles in 2007, collected by the L.A. Times [121]. Color indicates the cause of death and shape indicates the victim's race. The selection highlights Hispanic victims in central L.A. Figure 6.5b shows the same data plotted as a scatter plot of incident date and victim's age. The selection made in the geographic view is preserved across views: range criteria for latitude and longitude from the geographic view appear as interactive ranges within query histograms next to the scatter plot. Our system inspects the clauses of the selection query to generate the additional range visualizations and thereby ensure that the structure of the selection query is visible.

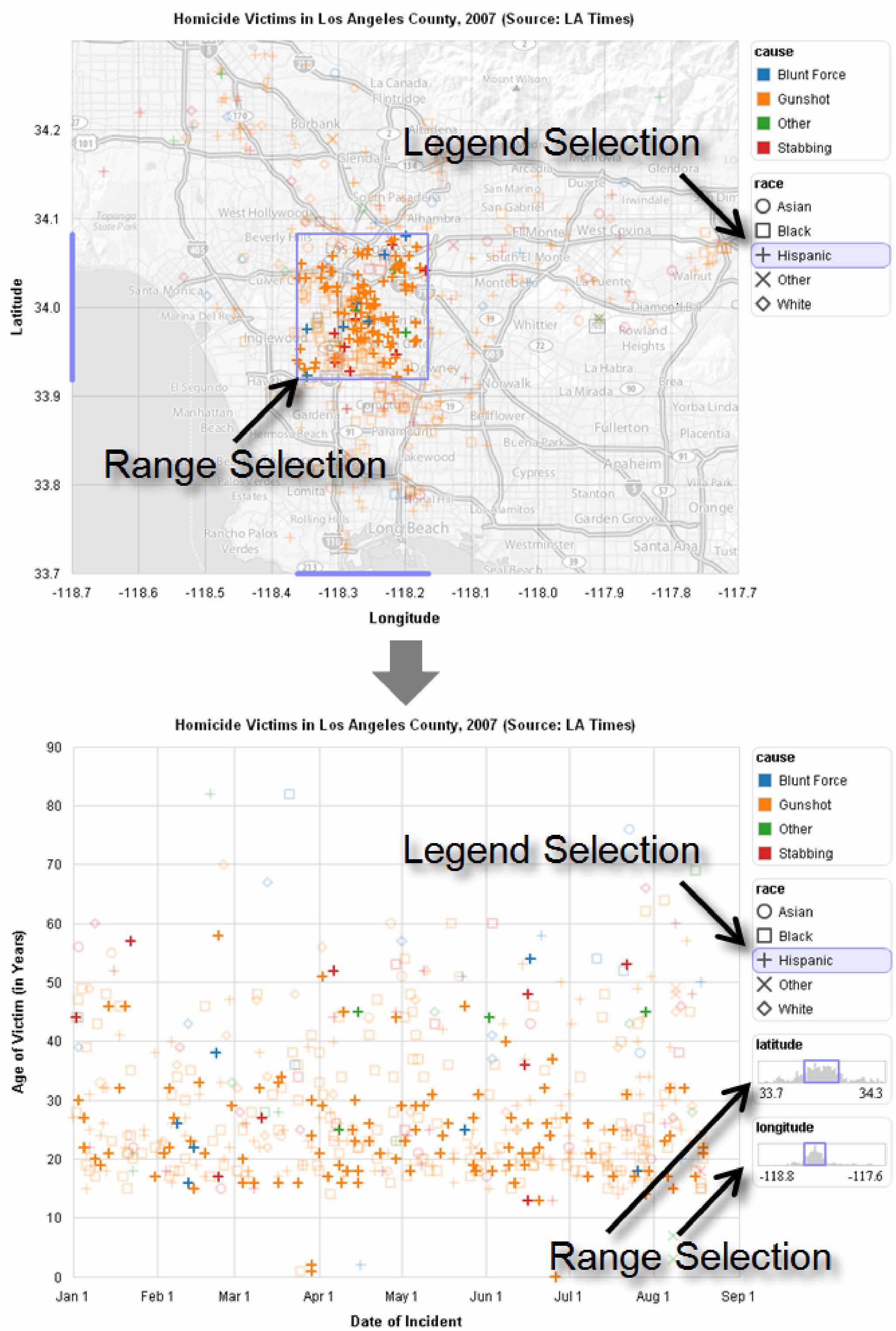


Figure 6.5. Reported homicides in Los Angeles County, 2007. (a) Left: Geographic distribution of homicides, including the cause of death (color) and victim's race (shape). A selection highlights Hispanic victims (using a legend selection) in central L.A. (using a range selection). **(b) Right:** The same data plotted using incident date vs. victims' ages. The selection made in the geographic display has been mapped to the scatter plot. Our system extracts the latitude/longitude ranges from the selection query and generates appropriate dynamic query widgets.

6.2.3 Data-Aware Annotation

In addition to exploration, selections are important for indicating items for collaboration and presentation (CHAPTER 3). Users can add text annotations as they explore a data set. Our system links the annotation to the data using the current selection query. When collaborators view each others' annotations, the system applies the saved query. Because our system enables reuse of queries across different views, collaborators can view each others' annotations under different visual encodings, potentially providing additional perspectives in subsequent collaborative analysis.

Furthermore, we use the query structure to rank and filter annotations. For example, when a query results in an empty result set due to external filtering criteria, it might be helpful to omit the result from the list of relevant annotations. In addition, we can compare the data columns referenced by the query with the visualized data columns to form a similarity measure between the selection query and the current view. We apply this measure to sort annotations according to their relevance to the current view.

6.2.4 Query Relaxation: Generalizing to Related Selections

Our system also supports the construction of generalized selections from simpler selections using query relaxation techniques. Users can pick an item or region of interest and generalize the selection to include additional items related to the initial selection (e.g., "select all items like this one").

Consider the date-by-age scatter plot in Figure 6.5b. Clicking an individual item queries the backing data tuple. Figure 6.6 depicts the use of repeated clicks to cycle through relaxed queries for the date attribute, expanding the selection to include items in the same day, week, and month. In this case, our query relaxer generates sequential relaxations by traversing a hierarchical calendar model of time.

A click-and-hold invokes a dialog box, with which the user can choose attributes of interest, such as cause of death, race, and age (Figure 6.7, left). The relaxed query selects all items that match the attribute values of the initially selected items (Figure 6.7, right). In this fashion, users expand selections based on attributes of interest.

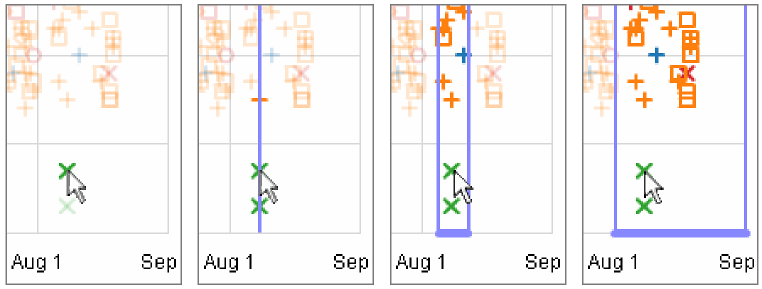


Figure 6.6. Relaxation of date ranges. One click selects an incident, two selects the day the incident occurred, three selects the week, and four selects the month.

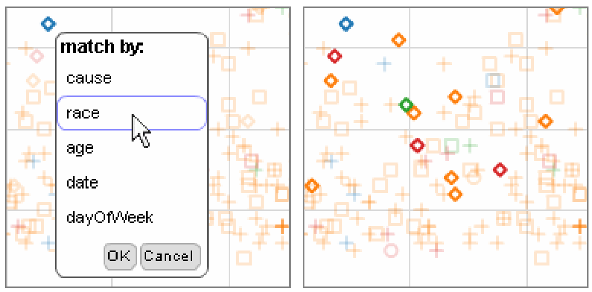


Figure 6.7. Relaxation by attributes. A click-and-hold action invokes a dialog for relaxing selections using one or more attributes. Above, a user selects all victims whose race matches the initial selection.

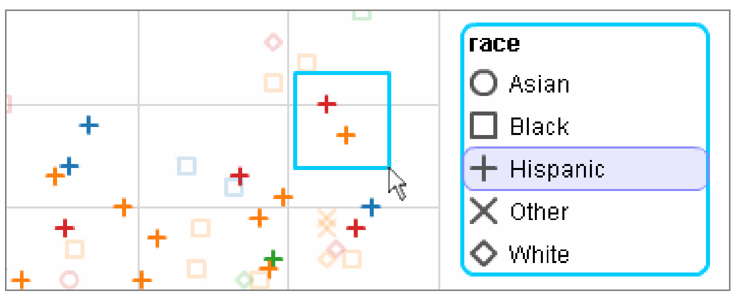


Figure 6.8. Range selection relaxed along the 'race' attribute. The generalized query selects all victims whose race matches that of any victim within the range bounds. Matching colors for the range selection and legend border indicate the relaxation relation.

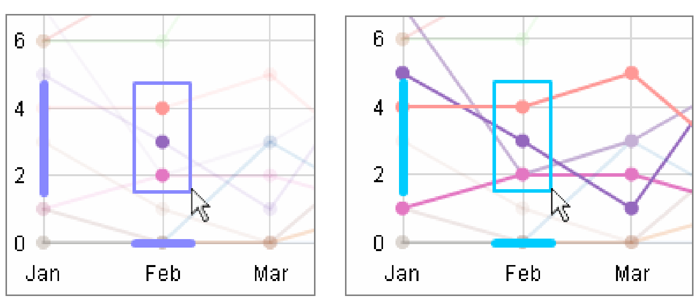


Figure 6.9. Time-searcher created by query relaxation. A user selects a range and relaxes the selection to create a tool that selects the time-series that pass through the range. Moving or resizing the range updates the relaxed query results. The images are cropped close-ups of a time-series of homicide counts by age group.

We also apply query relaxation to multiple items and to range queries. Figure 6.8 shows a 2D range selection relaxed along the 'race' dimension. The resulting query selects all victims whose race matches that of any victim contained within the range bounds. Figure 6.9 shows a similar relaxation in a time-series visualization. The plot shows aggregated homicide counts over time, grouped by age into 5-year bins. Creating a range query over this visualization selects all individual data points within the range. Relaxing the query along the age dimension selects all time-series that pass through the selection range. Because we retain the query structure, subsequent resizing or repositioning of the range results in dynamic updates to the selection, enabling interactive querying similar to Hochheiser and Shneiderman's TimeSearcher [100].

6.2.5 Alternate Output Modalities

Developers can further extend or customize how selections are presented. Our system can output selection queries in a SQL -like syntax to be exported (as in Figure 6.3) to databases or hand-modified by proficient users. It can also map selection queries into a natural language representation, providing automated captioning for selections and potentially aiding visually impaired users. Using simple templates, our system generates text descriptions of selections. For example, our captioner outputs "All items from August 1 to August 31" for Figure 6.6 (right panel) and "All items with race equal to 'White'" for Figure 6.7.

6.3 Example: Vector Graphics Editor

Although our primary motivation for building generalized selection techniques comes from data visualization, our approach is applicable in other visual interfaces. To demonstrate the generalizability of our approach, we have applied our selection techniques in a vector graphics editor, similar to programs such as PowerPoint and Visio. As in the earlier visualization examples, users can select both individual items and ranges, create data-aware annotations (e.g., for design reviews), and generalize selections through query relaxation. The principal difference is that for the vector graphics editor, no translation between visual and data variables is needed, as the data set being queried consists of the graphic objects themselves.

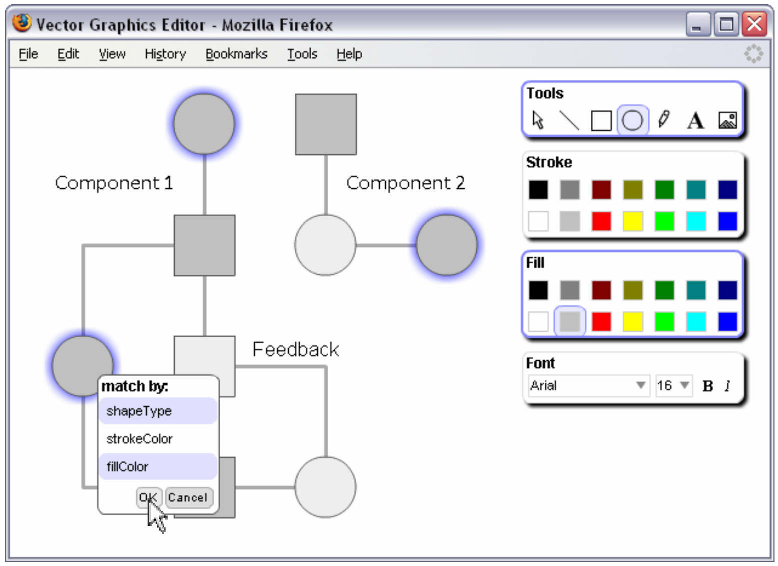


Figure 6.10. Vector graphics editor. Palettes on the right provide drawing operations. Our selection framework has been applied to enable generalized selection: here a user uses attribute relaxation to select items with a matching shape and fill color.

As before, clicking and holding over an item provides a dialog allowing users to generalize their selection to items with matching shapes, colors, and fonts (Figure 6.10). For example, one can click a text object and generalize by the font type to select all matching text boxes, enabling subsequent batch editing. Thus, our system automatically generates operations similar to the *Select > Same* and *Select > Object* menu commands in Adobe Illustrator.

Moreover, the query relaxer supports additional forms of query relaxation. The drawing editor includes connectors, which link items in an underlying network. This network provides a substrate on which to perform query relaxation. As shown in Figure 6.11, one click selects an item, two clicks also selects all items one hop away, and three clicks selects the entire connected component. We describe other forms of relaxation over networks in the implementation section.

6.4 Implementation of Generalized Selection

We implemented our generalized selection techniques in the ActionScript 3 programming language. A selection controller enables selection over visual items in the Flash Player scenegraph, using queries over the properties and sub-properties of these objects. In addition to processing input events, the controller coordinates query generation, query visualization, and query relaxation components.

6.4.1 Initialization

The controller takes as input both a container object holding the selectable objects and a schema mapping describing the accessible properties of interface objects. When visual variables (e.g., position, color, shape) are determined from backing data, the schema object maintains this mapping, including scale transforms (e.g., ordinal, linear, log scales).

6.4.2 Query Generation

Our query builder converts selection interactions into queries. For example, shift-clicking two items in the geographic plot of Figure 6.5a generates a query of the form

```
SELECT * FROM data WHERE (id = 10556 OR id = 10548)
```

The query directly selects items via unique IDs (e.g., primary keys). For simplicity, we show only the WHERE clause for the rest of the examples in this section. Dragging a range creates a query of the form

```
(-118.371 ≤ longitude AND longitude ≤ -118.164) AND
    (33.915 ≤ latitude AND latitude ≤ 34.089)
```

As specified by the schema mapping, our system replaces visual variables such as x and y with backing data variables such as latitude and longitude. Similarly, clicking on a legend generates a clause for the corresponding attribute value, e.g., `(cause = ‘Gunshot’)`.

Selection queries are represented internally as a tree of query operators, including nodes for literal values, variables, comparison operators, and Boolean logic. By default, query clauses generated in the same region of the interface are combined in an OR clause and the results are then combined by AND clauses. For example, creating two y-axis range selections and clicking the "Stabbing" legend entry in Figure 6.5b could result in the query clause:

```
((0 ≤ age AND age ≤ 10) OR (30 ≤ age AND age ≤ 40))
AND (cause = 'Stabbing')
```

6.4.3 Query Visualization

The query visualizer is responsible for visually conveying the structure of the query and indicating the items selected by the query. The query visualizer first traverses the query operator tree to construct an index of the various clause types (e.g., item selections, ranges, attribute selections, and nested queries). The visualizer can highlight individual query results using any visual highlighting effect, such as fade, blur, glow, and spotlight [110] effects. We use fade and blue transparent overlays as the default highlighting mechanism. For range clauses, the visualizer generates range brush controls, which users can interactively drag or resize (e.g., Figure 6.1). The query builder updates range clauses in response to the drag and resize actions. For attribute selections, the visualizer highlights each selected attribute in the legend or palette displays (Figure 6.3). For the results of query relaxation, the visualizer highlights both the initial selection and the relaxed attributes using matching colors (Figure 6.8).

6.4.4 Query Relaxation

Query relaxation generalizes the query structure to create expanded selections based on the properties of interface items. We define a relaxation operation based on the semantic structure of the attributes of the underlying data and a policy for generating relaxed queries by traversing this structure. Here we consider three forms of relaxation and their corresponding semantic models.

Relaxation using Semantic Hierarchies

Hierarchies are a common structure for modeling a data domain. For example, we can hierarchically organize time into days, weeks, months, of years, as in the example shown in Figure 6.6. Similarly, we might hierarchically organize geographic regions into neighborhoods, cities, counties, and states. We can also generate semantic hierarchies in a data-driven fashion. For instance, an analyst might apply hierarchical clustering (c.f., [41]) to analyze her data, and use the resulting cluster trees to describe the data at different levels of abstraction.

Our system includes a general software interface for specifying hierarchical ontologies. We also include basic ontologies for common data types such as time (e.g., days, weeks, months, years) and numbers (e.g., relaxing by increasing powers of ten). Application designers can provide their own ontologies for custom data types, whether hand-crafted or data-driven.

To perform relaxation, we traverse these semantic hierarchies. With each relaxation step, the relaxer moves one level higher in the hierarchy and generates a query that selects all values in the current sub-hierarchy. For instance, relaxing date as in Figure 6.6 results in the query

```
SELECT * FROM data WHERE
     RELAX('date', 1, SELECT * FROM data WHERE id = x )
```

The relaxation is specified as a nested query in which the initial selection is a subquery. We use the result set of the initial selection's subquery as input to a relaxation operator. The relaxation operator takes two additional parameters: the name of the semantic structure to use and a parameter specifying a traversal policy. In the example above, the parameter 'date' indicates that the semantic hierarchy for dates should be used, and the parameter '1' indicates that the query should be relaxed by one level of abstraction, to include all items that occurred on the same day as the initial selection. A level of '2' would relax the selection to all items in the same week. The relaxation operator outputs a new query clause that can be analyzed by the query visualizer. For example, in the example above the relaxation operator returns a comparison clause for the selected day, (date = '2007-08-05').

Relaxation using Attributes

For some types of data and attributes, semantic structures are not available. When no explicit semantic structure is provided, our system assumes a "flat" hierarchy and relaxes the query to select all items with attributes exactly matching those contained in the initial selection.

The resulting relaxed queries select all items with some subset of attributes matching items contained in the initial selection, as in Figures 6.7-6.10. Consider Figure 6.5b. If the initial selection is a single object `(id = 10556)`, relaxation of the 'race' attribute results in the query:

```
SELECT * FROM data WHERE
      (race IN SELECT race FROM data WHERE (id = 10556))
```

Because the hierarchy here is "flat," we can forego the relaxation operator. As before, the relaxed query is specified in terms of a nested subquery. In the example above, the inner query returns the set of 'race' attributes present in the result set of the initial selection `(id = 10556)`.

If we modify our initial selection, the result set of the relaxed query also updates. For instance, if we relax a range query (Figures 6.8 and 6.9), we can interactively update the range bounds, which refines the inner "selection" query, dynamically changing the input to the relaxation. If dynamic updates are not desired, we can collapse the query structure by evaluating the inner query to generate a query without nesting (i.e., without any inner queries). To generate a "collapsed" query, we evaluate the relaxation clause, replacing it with a static clause such as `(race = 'Asian')`.

Relaxation using Networks

General network (graph) structures can also serve as semantic structures for query relaxation. Our internal query language includes traversal policies for such network structures. We have implemented traversals for selecting neighbors, connected components, ancestors or descendants (for DAG structures), or all items along the shortest-paths between items in the initial selection. Figure 6.11 depicts relaxations over a network in our vector graphics editor. As before, the relaxed query takes the form of a nested query:

```
SELECT * FROM data WHERE
     NEIGHBOR-OF( SELECT * FROM data WHERE id = x )
```

The formulation of this query is similar to the semantic hierarchy example, except that the semantic structure and traversal policy are implicit for the NEIGHBOR-OF operator.

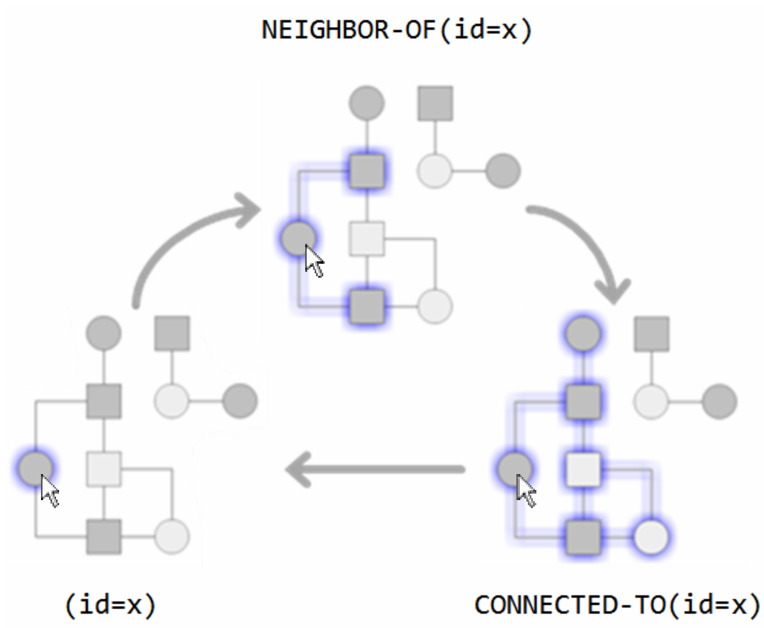


Figure 6.11. Query relaxation of networks. Connectors link visual items in a network. Query relaxation can be performed on the network structure. Here, one click selects an item, two clicks selects connected items, three clicks selects the connected component.

Configuration

Application designers can parameterize the query relaxation process by providing semantic structures and traversal policies for data attributes and specifying ordering constraints among attributes. Furthermore, a rule engine allows the query relaxer to consider different attributes based on the context of the selection. For example, the vector graphics editor contains a rule that enables relaxation of typeface attributes when the initial selection query only contains textbox items.

6.4.5 Query Reuse

One advantage of our framework is that it can reapply selections across changes of visual encodings. As shown in Figure 6.5, our system generates new query widgets as needed to convey the complete structure of the selection. However, some expressions do not map from one view to another in a straightforward fashion. Consider a pair of 2D range selections, such as two latitude/longitude ranges. These selections result in a selection clause of the form

`(R1x AND R1y) OR (R2x AND R2y),`

where `R1x` denotes the X component of the first selection, `R1y` denotes the Y component, and similarly for `R2` and the second selection. If we change visual encodings, we might naïvely generate independent query histograms for these ranges: one for X and one for Y, as in Figure 6.12a. However, Figure 6.12a incorrectly depicts the selection, instead conveying the following query structure:

`(R1x OR R2x) AND (R1y OR R2y).`

The confusion is due to the use of Boolean operators in our visual query language: the convention is to OR all selections made within a query component and then AND the selections from separate components. Our solution is to use multivariate query widgets when confronted with multiple 2D ranges. Figure 6.12b shows a scatter plot histogram that depicts multiple 2D selection ranges and allows interactive refinement.

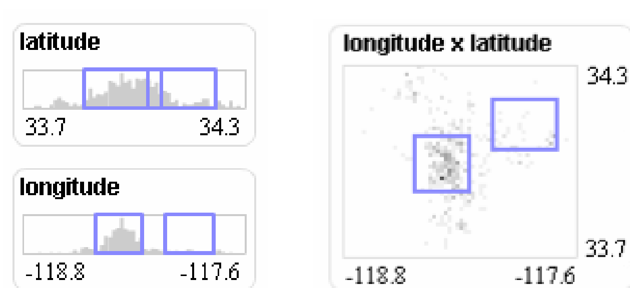


Figure 6.12. (a) Left: 1D components may incorrectly communicate multiple ranges. (b) Right: A scatter plot histogram for 2D ranges.

6.5 Evaluation

To better understand the use of our selection techniques, we conducted a user study of a data visualization application supporting generalized selection. We asked subjects to perform both interpretation and authoring tasks using our selection framework. We were interested in how subjects interpreted visual representations of selection queries and if these interpretations changed in response to interactive use. We were also interested in whether subjects would use only direct clicking and dragging for selection or whether they would also use query relaxation. Finally, we wanted to investigate if the choice of selection mechanism had a subsequent impact on selection accuracy.

Sixteen subjects (11 female, 5 male) aged 18-27 (M = 21.2, SD = 2.43) participated in the study. All subjects were students at our university, studying subjects such as biology, business, engineering, political science, and statistics. Subjects were recruited through the X-Lab (http://xlab.berkeley.edu), a research participation service.

6.5.1 Methods

Subjects completed a set of tasks interpreting and authoring selections in a scatter plot visualization of homicides in Los Angeles (Figure 6.5b). The data set contained 627 data points noting the incident date and victim's age, race, and cause of death. The visualization showed a plot of date vs. age, with race and cause encoded by shape and color, respectively. The study consisted of three phases with 12 tasks each. Subjects required 30-45 minutes to complete the study.

In phase 1, we presented visual selection queries to subjects and asked them to describe, as completely as possible, the subset of the data highlighted in the view.

In phase 2, subjects first completed an interactive tutorial that provided descriptions of the selection operations and asked subjects to practice each selection before proceeding. For the study, relaxation was performed through repetition mediation only. Subjects could repeatedly click to cycle through relaxations of the cause, race, date, and age attributes. Subjects were then given text descriptions of subsets of the data and asked to make matching selections in the visualization. The provided text descriptions include:

- Victims who were exactly 60 years old.
- Victims killed by Blunt Force between March 1 and April 1.
- Victims killed by the same causes that killed any victims over 80 years old.

While users could complete all tasks by directly specifying the selection via clicking and dragging, they could also complete a subset of these tasks using query relaxation. Thus, subjects could apply relaxation as they saw fit. For example, in the third description above, subjects might either select the matching causes directly or select all victims over 80 and perform query relaxation.

The task in phase 3 was the same as in phase 1; subjects were again asked to interpret pre-defined selections, though the selections were different from those shown in phase 1. Afterwards, subjects completed a short survey.

In each phase, the selection cues were systematically varied to thoroughly cover the query structures expressible with our techniques, including 1D and 2D ranges, category selections, disjunctions within variables, and conjunctions across variables. In addition, multiple selections involved generalizing from a subset of the data. We used the same distribution of query structures in each phase of the study.

6.5.2 Results

We were interested in three primary questions. First, how did subjects interpret selections, and did interpretations change with interactive use? Second, which selection operations did subjects use to create selections, and what effect did they have on subjects' accuracy? Third, what did subjects think of the selection techniques?

Selection Interpretation

To analyze subjects' descriptions of observed selections in phases 1 and 3, we coded each response into one of four categories:

- *Structure-correct* responses accurately reported the structure of the selection query, including basic clauses and appropriate disjunctions and conjunctions.
- *Result-set-correct* responses did not report the query structure correctly, but specified criteria which resulted in the same query result set.
- *False-conjunctions* correctly identified each query sub-clause but combined them inaccurately.
- We coded as *incorrect* all other responses that failed to describe the query structure and results.

Note that all structure-correct responses also produce correct results. We do not count the structure-correct responses in the result-set-correct category. False conjunctions were prevalent when the stimulus involved a range generalization (e.g., selecting all categories contained within a range, as in Figure 6.8). Table 6.1 shows the percentage of responses in each category.

Table 6.1. Responses in selection interpretation tasks.

Response Type	Phase 1	Phase 3	Average
structure-correct	73%	78%	75%
result-set-correct	3%	4%	4%
false-conjunctions	11%	7%	9%
incorrect	13%	11%	12%

Ideally, users would understand selections even if they have not previously used the software. To test if authoring selections changed how subjects interpreted selections, we compared the distributions of coded results from phase 1 (before interaction) and phase 3 (after interaction) across all tasks. We found no significant difference in the distribution of response types across study phases ($\chi^2(3, 362) = 2.26$, $p = 0.521$).

However, both phases included two tasks in which the selections were created by relaxing a range query along a categorical attribute. These selections result in a potentially confusing display: a range brush is visible but selected items exist outside the range (see Figure 6.8). Unsurprisingly, 96% of all false-conjunctions occurred in these cases, in which subjects identified both the range and category criteria but did not understand the relation between the two.

To see if interactive user affected interpretation of such range relaxations, we analyzed just the tasks involving relaxation of a range selection and found a significant difference across phases ($\chi^2(3, 60) = 9.22$, $p = 0.027$). The number of structure-correct and result-set-correct responses increased in phase 3, with a corresponding drop in false-conjunctions. However, “correct” cases only accounted for 39% of responses, suggesting that even with exposure our subjects found relaxation of range selections hard to interpret. We also note that this analysis involves a relatively small amount of data, as each subject saw only two such range relaxations per phase.

Selection Authoring

To analyze the selection queries authored in phase 2, we similarly coded the responses into categories. In this case, we used only three categories: structure-correct, result-set-correct, and incorrect queries. Overall, subjects created selections that matched the text descriptions: 62% structure-correct, 20% result-set-correct, and 18% incorrect.

We were also interested in whether or not subjects would use query relaxation. Eleven of 16 users (68%) used multi-click relaxation to respond to a task, over a total of 28 tasks (15%). We hypothesized that subjects would be more accurate using query relaxation, and divided the responses into those that used relaxation and those that did not. We found a significant difference in the distribution of response types between the groups ($\chi^2(2, 192) = 11.45$, $p < 0.003$), with structure-correct responses comprising 89% of relaxation-generated responses, compared to 57% of responses made through other means.

We hypothesized that the difference might be due to individual differences — users who apply relaxation may be more advanced and perform better overall. To test this possibility, we divided the responses according to whether or not the subject used relaxation at any point. We found no significant difference in selection accuracy between the two groups ($\chi^2(2, 192) = 4.15$, $p > 0.10$), suggesting that the subjects who used query relaxation were not significantly more accurate overall.

These results suggest that subjects may make more accurate selections when using query relaxation in tasks amenable to relaxation. However, we note that the nature of the task likely plays a crucial role in shaping subject performance.

Subject Preferences

At the end of the experiment, subjects were asked to rate the techniques presented within the experiment on 5-point Likert scale. Overall, subjects found the selection techniques helpful ($M = 3.75/5$, $SD = 0.45$) and did not find them confusing ($M = 1.75/5$, $SD = 0.77$). We also asked subjects to rate query relaxation. Overall, subjects rated query relaxation favorably ($M = 3.86/5$, $SD = 1.10$). However, the rating distribution was bi-modal, split between those who used query relaxation ($M = 4.36/5$, $SD = 0.50$, $N = 11$) and those who did not ($M = 2.40/5$, $SD = 0.55$, $N = 5$). The difference between groups was significant ($t(14) = 7.04$, $p < 0.001$). Overall, subjects' comments were positive (*"it's very useful to find matching characteristics"*), but also suggested usability improvements for the visualization application. For example, the fading effect applied to unselected items made it difficult to sequentially select ("shift-click") individual items.

6.6 Discussion and Summary

In this chapter, we presented a framework that models selections of interface objects as declarative queries in a SQL -like language, capturing both the structure and content of a selection. Users create selection queries through direct manipulation; our system visualizes the structure of the query and highlights the results. Selection queries enable evolving selections over dynamic objects and streaming data. Our system can reapply selections across applications, without loss of structure. Users generalize these selections via interactive query relaxation, expanding their selections according to one or more attributes of interest. Results from our user study suggest that users successfully interpret and author selections using our system and that query relaxation may improve selection accuracy in amenable tasks.

Though the majority of subjects relaxed a range query to successfully complete a task, many had difficulties interpreting relaxed range queries they had not created themselves (e.g., Figure 6.8). While most subjects correctly interpreted individual query components, they often did not recognize the generalization relation. This result suggests that simplified selections may be more appropriate for collaboration, as in the use of selection queries to specify annotations. Accordingly, we recommend "collapsing" nested structures that contain subqueries when using selections to communicate with a general audience, by evaluating the nested relaxation query. This limitation suggests future work in designing visual representations. Are there intuitive ways to indicate nested query structures without requiring an additional auxiliary interface?

Another avenue for future work is to further extend the query relaxation mechanisms. Other application domains might suggest new semantic models or traversal policies for relaxation. New relaxation types may be best expressed using additional input gestures, and expert users may want to configure the relaxation engine at runtime.

A primary motivation for developing our selection techniques is to support web-based collaboration around visualizations. The generalized selection framework provides a base for the creation of data-aware annotation facilities that enable analysts to select both items, properties, and regions of interest in a manner that is robust to time-varying data. Furthermore, our system can re-apply annotations across different visual encodings of an underlying data set, enabling analysts to compare and contrast selections in a range of visualizations. As we show in section 6.5, analysts can use our selection techniques to author annotations for sharing interesting data items and regions encountered during collective data analysis.

7 Graphical Histories for Visual Analysis

When investigating data with interactive visualizations, users regularly traverse the space of possible views in an iterative fashion. Exploratory analysis often results in a number of hypotheses, leading to multiple rounds of question-answering. Analysts may generate unexpected questions that they investigate immediately or revisit later. After conducting analysis, users may need to review, summarize, and communicate their findings, often in the form of reports or presentations.

By surfacing users' interaction history, we can facilitate analysis and communication. History mechanisms such as undo enable revisitation of previous states in a variety of applications (e.g., [4, 14, 24, 33, 57, 64, 96, 105, 107, 111, 113, 114, 118, 123, 128, 144, 153, 161]). As Shneiderman [162] notes, history tools can play an important part in the visualization process, supporting analysis by enabling users to review and revisit visualization states. As noted in CHAPTER 3, history is also a valuable adjunct to social data analysis. Graphical history tools can aid coordination by summarizing others' activity and can facilitate the construction of presentations and tours.

Interaction histories can also benefit research and development. History log analysis of both individual and aggregate usage can identify common usage patterns and thereby assist usability evaluation. Researchers can also study interaction patterns to better understand and model analysts' sense-making processes [105].

Designers of visualization tools must consider a large design space of potential features and system architectures when designing history tools. These design decisions entail trade-offs in the types of history representations and operations that can be provided.

Previously published by the author, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala in [94].

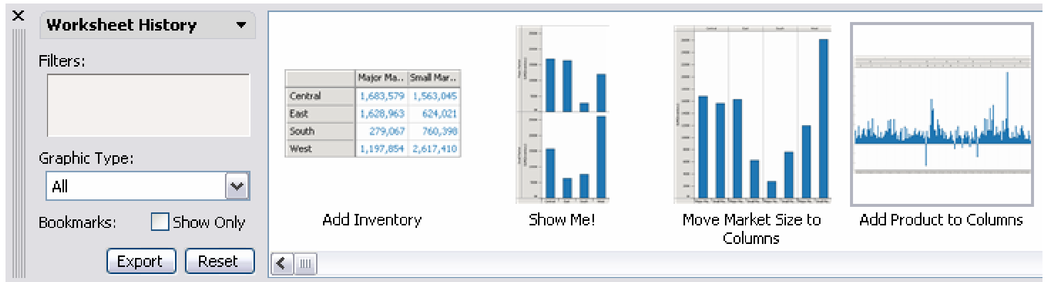


Figure 7.1. A graphical history interface. Thumbnails show previous visualization states and labels describe the actions performed.

For example, while it is easy to log low-level input events such as key presses and mouse clicks [153], users can more readily take advantage of semantically meaningful models. Users might perform operations on an interaction history, including editing, aggregation, bookmarking, annotation, and search. Architecture and interface design need to account for such operations. Furthermore, interaction histories can grow large quickly, and thus history mechanisms must scale accordingly. Scale concerns arise at the data level, where histories can benefit from compact description, and at the visual level, where history interfaces should be perceptually effective and space efficient.

In this chapter, we explore the design of graphical history tools to support visual analysis. We first present the results of a design space analysis, enumerating design decisions for the software architecture and graphical interface of history systems. Our analysis is intended to provide an overview of important design considerations and thereby help practitioners incorporate graphical history tools into their own visualization applications.

Next, we present the design and implementation of graphical history tools to support analysis, communication, and evaluation in Tableau, a commercial database visualization system [125, 167]. Based on our design space analysis, we describe the design of graphical history tools supporting analysis and communication. Although our primary contribution is a design study of history tools for visual analysis, our graphical history prototype also contributes novel techniques for improving scalability, searching histories for relevant views, and generating presentations from history subsets. Furthermore, we have used our history model to support evaluation by analyzing individual analysis sessions and aggregate usage patterns. We describe our visual history analysis tools and how we have applied them to improve the design of Tableau's user interface.

7.1 Design Space Analysis of Interaction Histories

Architects of interactive history systems face a number of design decisions impacting the representations and operations available to users. To design our history tools, we first conducted a design space analysis to enumerate these decisions. We surveyed prior work spanning general history mechanisms [14, 64, 70, 134, 182] and interface designs in the areas of graphical design tools [64, 111, 113, 128, 161], web browsing [4, 36, 96, 107, 108, 185], and visualization [24, 33, 57, 78, 105, 114, 118, 123, 144, 153]. In this section, we outline the design space of history tools using examples pulled from this body of work.

7.1.1 History Models

Actions vs. States

We model interaction histories as movement through a graph of application states. Nodes in the graph represent discrete states of the application and edges represent the actions that transform one state into another. A state is defined by the settings of interface widgets and the application content (e.g., document, data, etc). At the architectural level, developers must decide if their history system will maintain sequences of states, actions, or both, and how such history items—discrete representations of an action or historical state—will be organized.

Software engineers often refer to action logging as the command object model [70]. Command objects encapsulate an interface action, typically providing both do and undo methods that apply the operation or its inverse. To traverse the history, a sequence of commands can either be done or undone in order. This approach requires that suitable inverse (undo) operations are defined for all actions.

An alternative is to log the individual states of the application. Traversing the history then involves restoring the application state to a stored configuration, removing the need to sequentially apply undo actions. However, the drawback of this approach is that the state representation can become memory inefficient.

The action and state approaches are not mutually exclusive and hybrid approaches are possible. For example, an action-based history mechanism might periodically cache the state to reduce the number of operations required by history traversal. A state model might also log metadata about the operations that were applied between states. For example, the WebQuilt web logging system [185] stores URLs (states) but also notes the index of the link clicked in the previous page, modeling web browsing at the level of individual links.

In surveying the literature, we have found that action logging is prevalent within graphic design tools, where large content models can make state models memory-inefficient. In contrast, state logging (as URLs) is common for web browsing histories. Visualization systems have utilized both approaches. As we will discuss later, this choice affects the range of history operations that users may perform, particularly with respect to editing and selective undo.

A common approach in visualization is to describe the visualization in terms of a chain of visual encoding operators that are applied to the data to generate the visualization state. Jankun-Kelly et al. [105] introduce a general model for visualization state as a set of parameters, and actions as transformations of these parameters. In CHAPTER 4, we demonstrated that identical visualization views can be reached through different parameter sets. In particular, different filtering criteria may yield the same result set. Thus, accurate analysis of revisitation may require that state models include an index of the underlying content in addition to parameter settings.

One modeling issue specific to visualization is its data-driven nature: application states are dependent on the backing data set. If a visualized data set includes streaming or editable data, a faithful history system must also take the changes to the data into account. It may be that users want historical states to update with changes to the data, thereby keeping their analysis current (a form of selective redo, discussed later). If not, users could use data management systems that support versioning or provenance [11]; however, such systems may entail an unacceptable storage cost. As many visualization views depend only on a subset or aggregate of the backing data, creating an extract of the data for a "snapshot" of the visualization state may be a feasible solution in many cases.

History Organization

History items may be organized in various ways. The stack model places items on both undo and redo stacks. This approach does not support branching histories, as the redo stack is cleared when new actions occur. A timeline model stores items in the linear order in which they occur. Branching models [182] store items in a tree structure, and actions performed after undo operations form a new branch of the tree. Additionally, history models may perform content indexing and organize history items by other metadata properties.

Hierarchical Command Objects

Systems may represent history items at multiple granularities. For example, one can group a sequence of low-level actions into a higher-level action through hierarchical command objects [134]. Grouped actions may provide a better semantic description of a user's intention. To construct groupings, developers can craft inference rules [113] based on the type and timing of actions. However, groupings also raise challenges for representing and navigating hierarchical history items in a user-friendly manner.

Local and Global History

One can organize history items by the objects to which actions are applied. For example, a spreadsheet may maintain separate histories per worksheet, while a graphics editor could maintain local histories for objects in the scene. Edwards et al. [64] propose a transactional model to support local histories in which actions may have global side-effects. In all cases, applications must support the ability to merge local histories into a global timeline.

7.1.2 Visual Representation of History

Visual Presentation

One simple presentation of a history item is a text description of the state or action, commonly found as menu text for undo and redo actions. Text descriptions should be easy to understand, and may require subtle design decisions. For example, web history systems have carefully considered different abbreviation approaches for web page titles and URLs [4, 107]. While text may be helpful for describing actions performed in a visualization, they are less well suited for the graphical nature of a visualization state.

Graphical representations of histories are also common. Some depictions involve abstract properties: for example, the color of a history item glyph might represent the type of action performed. Most common, however, are thumbnail images used to aid users' recognition of the previous interface state—an approach particularly relevant for visualizations [123]. Multiple studies have found benefits for thumbnails in web browsing [107, 193], with one study suggesting that a thumbnail size of about 120 pixels square is enough to enable 80% accurate recognition of a visited web site [108]. Other projects [111, 113, 193] enhance thumbnails to improve comprehension by highlighting changes and applying strategic cropping and callouts.

Spatial Organization

Depending on the underlying history model, a number of visual organizations of history items are possible. A common approach is a linear sequence of items, like a comic strip [113, 128]. Such an organization facilitates visual scanning of the history, and typically enables navigation by clicking an entry. A similar approach is to provide a quantitative timeline [57, 144, 161] that shows the time duration between actions and which users can navigate using a slider control.

Branching histories typically use a node-link tree diagram to show history branches. Prior work has adapted both the sequence [4, 24, 33, 96, 114] and timeline [57] metaphors into branching tree displays. Klemmer et al. [111] present an inline branching design that places collapsible history branches within a linear comic strip.

Other representations are also possible. Behavior graphs [33] are an alternative representation of branching histories that we will discuss in section 7.3.1. Another approach is to use a content-centric representation using variables other than time. WebQuilt [185] visualizes aggregate surfing behavior in a network diagram showing traversed links between web page thumbnails. Ma's Image Graphs [123] display history states as thumbnails connected in a graph layout and depict actions between states using iconic edge representations.

7.1.3 Operations on History

Designers must also consider the set of operations that graphical history tools should support.

Navigation

For end-users, the fundamental operation of history systems is navigation to states in the history. Undo and redo (or back and forward) actions are common navigation operations found in many applications. Another approach is for users to click the thumbnail of a history state to directly return to that state. Some systems use a "time travel" metaphor with a timeline slider. For branching histories, a graphical view of the history can help users differentiate branches. Content-based navigation is also possible, such as navigating to the point in time that an object was last edited [161].

Editable Histories

Other operations may involve editing the history, as users may wish to revise the history or replay past actions. In state-based history models, deleting a past state from the model does not affect any of the other states. In action-based models, editing has side-effects. Deleting a past action involves rolling back the history prior to the selected action and re-applying the subsequent actions. However, some subsequent actions may depend on a side-effect of the deleted action, in which case we need rules to ensure integrity.

While history editing is more complicated for action-based models, it enables unique operations. Selective undo [14] allows the replay of past actions after revising the history. Similarly, selective redo [57, 113, 114] enables users to copy and reuse chains of actions. Kurlander and Feiner [113] use this mechanism to support macro creation. For example, a user might re-apply a sequence of visualization transforms to a new subset of data [57, 114].

Metadata and Annotation

Users may also wish to add metadata and annotations to history items. Bookmarking [87, 107], keyword tagging, text comments [78, 89, 111, 144], and audio annotations [78] are all potentially useful. Usage scenarios for visualization include analysts creating bookmarks for important findings [107], leaving text notes to describe a view to a collaborator [111], and recording audio annotations to aid “think-aloud” evaluation protocols [78].

Search and Filter

As histories grow large, users may need means beyond visual search and scrolling to find past states of interest. Search tools are one solution. Metadata such as time, action type, bookmarks, and annotations are all potential search domains. Although some web design history systems (e.g., [111]) provide filtering tools, most visualization histories lack search capabilities. A notable exception is VisTrails [33], which enables querying-by-example to find related visual exploration sessions across multiple users.

Export

To enable communication, it is often important to export and share parts of a history. For web-based systems, one can distribute a URL [33, 89], but desktop applications are typically more cumbersome. Klemmer et al. [111] print out thumbnails and text annotations as paper reports. However, nearly all history tools are lacking more nuanced support for exporting histories into external media.

7.1.4 *Design Space Summary*

In this section, we have categorized a range of design decisions that arise when crafting an interactive history system. These decisions include how to represent and organize historical data (e.g., states, actions, or both), how to visually present histories (e.g., linear or branching layout), and what interactive operations the history should support (e.g., navigation, editing, search, and export). Still, the task remains of deciding which route to take when designing a system. The features and context of use of the underlying visual analysis tool can further inform the design process for history tools. To illustrate this process, we now apply our design space analysis to develop an interactive history system in the context of Tableau, a database visualization system.

7.2 Graphical History in Tableau: A Case Study

Based on the considerations raised by our design space analysis, we designed a history interface supporting analysis and communication in Tableau, a commercial visual analysis system. We now describe Tableau and present the design of our graphical history tools.

7.2.1 The Tableau Visual Analysis System

Tableau is a commercial system, based on Polaris [167], for visualizing the contents of databases. As shown in Figure 7.2, the Tableau interface includes a list of available database fields and a workspace in which users can select fields and drag them onto shelves corresponding to visual encodings such as position, color, shape, and size. Tableau is based on a specification language called VizQL. VizQL statements are generated from the contents of the interface shelves and they specify both the data that should be visualized (as database query statements) and how the visualization should appear (as visual specification statements). This formalism supports a range of visualizations, including bar charts, time series, scatter plots, and heat maps, as well as analytic operations such as filtering, sorting, and drill-down [167].

Akin to Microsoft Excel, Tableau supports multiple worksheets. Each state of a Tableau worksheet is described by a VizQL statement. Tableau's original history model used a state-based logging approach, with each worksheet organizing VizQL statements on undo and redo stacks. This model does not support branching histories, except through duplication of worksheets. Undo and redo buttons provide some support for history navigation, but the model does not provide text descriptions for

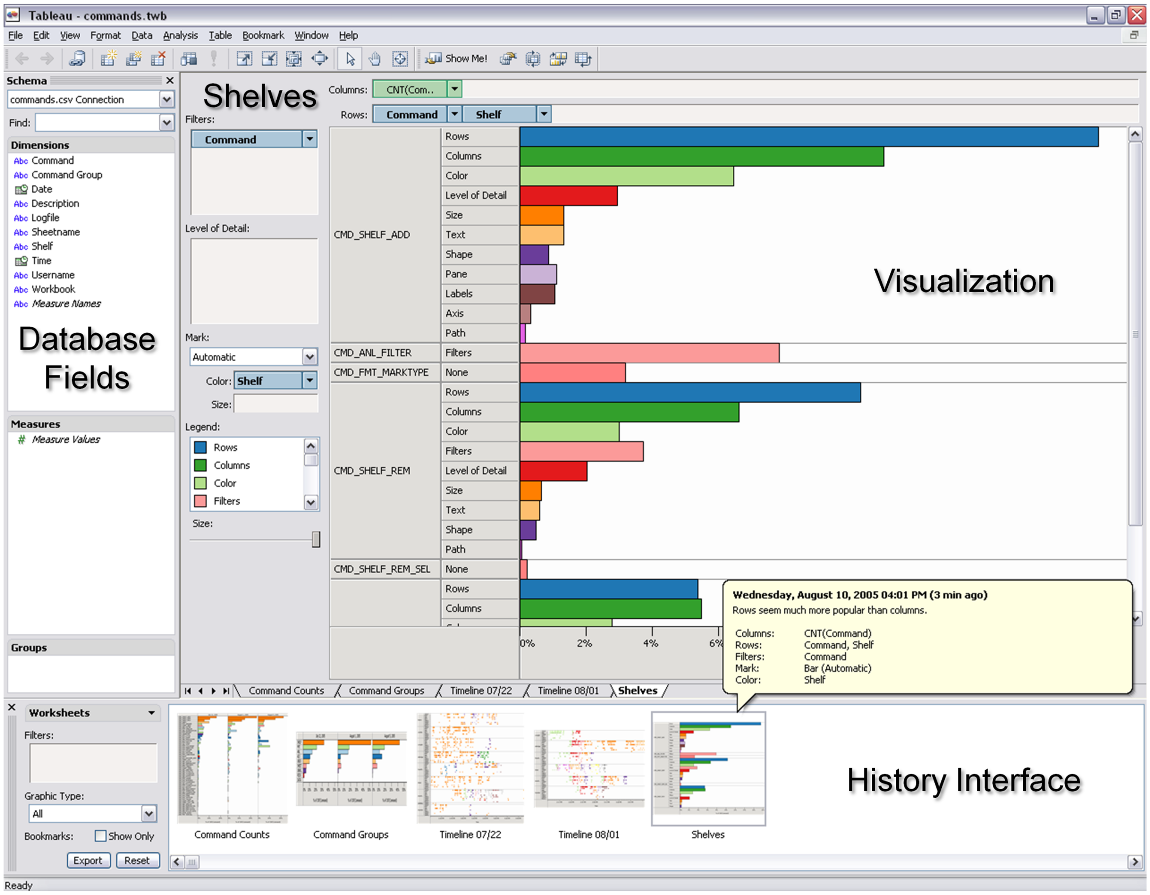


Figure 7.2. The Tableau visual analysis tool, visualizing data collected from aggregated history usage logs. The panel on the left provides a list of database fields. Fields can be dragged onto visual encoding shelves on the right to create visualizations. Multiple worksheets are accessible by the named tabs underneath the visualization. The panel along the bottom shows a history viewer, currently providing an overview of the state of each worksheet. A tooltip provides details-on-demand for the selected history item.

undo/redo actions. In the following sub-sections, we describe a redesigned model to better support analysis and communication.

7.2.2 A Re-designed History Model

In crafting a history model for Tableau, we wanted to maintain the existing, clean approach of declaratively modeling state as VizQL statements. However, VizQL statements alone are not enough. To provide a more informative user interface and to support usage evaluation, we also wanted to record historical data that enables us to provide high-level descriptions of user actions. As a result, our improved history model uses a hybrid state/action approach as identified in our design space analysis.

History items in Tableau still record states as VizQL statements, but we also introduced aspects of action-based logging. We created a classification scheme for each action supported by the interface. When an action occurs, the history system notes its unique identifier and any arguments, and then stores the command description and current VizQL statement as a history item. As VizQL statements are concise, declarative representations of the interface state, we do not incur an unreasonable memory overhead. Having a record of actions allows us to create text descriptions, improving the cues for undo and redo within the interface. Our classification scheme groups actions into five top-level categories: *shelf* (add, remove, replace), *data* (bin, derive field), *analysis* (filter, sort), *worksheet* (add, delete, duplicate), and *formatting* (resize, style) commands.

In addition to basic history items, our model supports composites of grouped sub-items, similar to hierarchical command objects [134]. All history items support data fields such as a timestamp, bookmark status, and text annotations. We organize items in a branching structure for each worksheet, replacing the prior stack model. When a user visits a past state and performs an action, a new analysis branch is added to the model (see Figure 7.5). Our history abstraction also supports merged histories, implemented as a composite history view of worksheet histories. By default, the state model does not include the database contents and changes to the database will cause historical states to update to reflect the current data. However, users can manually create data extracts if they wish, ensuring a static data set.

7.2.3 Visual Design of the History Interface

We designed our history representation with the understanding that graphical history should aid analysis in an unobtrusive fashion. The visualization should serve as the primary focus of attention and the history as an auxiliary display. We wanted to ensure that the graphical history "pays for" its screen real estate, using only the space needed for effective presentation and navigation of history items.

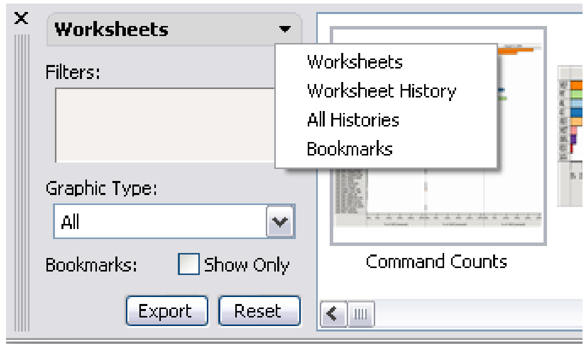


Figure 7.3. History interface. History can be filtered by data fields (via drag-and-drop), chart type, and bookmarks.

As a result, the history model is depicted using a sequential, comic-strip display (Figures 7.1 and 7.2 bottom), including a thumbnail image and text description for each history item. As tree diagrams can require a lot of screen space, we present branching histories inline: branch contents are listed sequentially, with sibling branches sorted by the timestamp of the first item (see Figure 7.5). We position the history viewer along the bottom of the interface. Users can optionally hide the viewer to make more space for the visualization. Hovering the mouse pointer over a history item reveals a tooltip with details-on-demand (Figure 7.2). The tooltip lists the time the state was first visited in absolute and relative ("3 min ago") time. The tooltip includes annotations added to the item and a summary of visual encodings: which data fields are placed on which shelves.

The interface provides four modes, accessible via a drop-down menu (Figure 7.3):

- Worksheets mode presents an overview of the current state of all worksheets, with a thumbnail and name for each (Figure 7.2).
- Worksheet History mode presents the history of a worksheet. Thumbnails are captioned with action descriptions (Figures 7.1, 7.6b).
- All Histories mode is similar to Worksheet History mode but depicts the merged global history across all worksheets.
- Bookmarks mode shows all views that have been bookmarked. Captions include the source worksheet name and a timestamp.

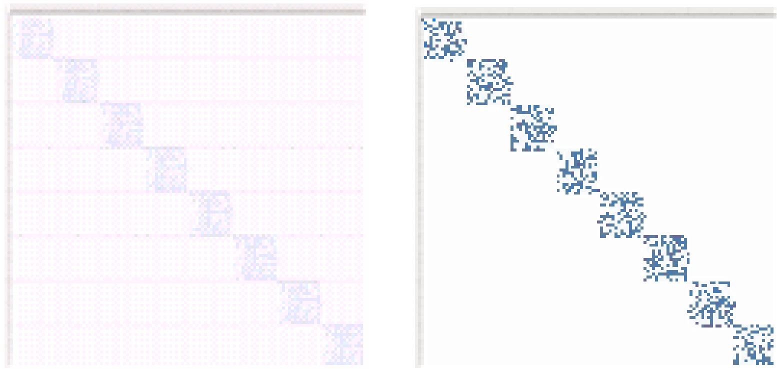


Figure 7.4. Adjusting thumbnail contrast. The image on the left is an overview thumbnail generated by down-sampling that suffers from "wash out". On the right, high-frequency elements such as gridlines have been removed and pixel values are adjusted such that the data color in the image matches the color encoding palette.

Thumbnail Image Generation

The history viewer provides thumbnails of visualization states to aid recognition. Thumbnail size introduces a trade-off between screen usage and recognizability. Based on Kaasten et al.'s [108] experimental results, we chose 120 pixels square. As noted by our design space analysis, some graphics editors [111, 113] provide enhanced thumbnails that highlight differences between history states and perform selective cropping. As changes in Tableau regularly involve complete updates of the visualization, this approach did not seem appropriate. Views in Tableau often require scrolling, so we reasoned that a thumbnail that provides an overview of the display as well as historical data would be the most useful for analysis.

To generate the overview images we render the visualization at its native resolution and then scale the resulting image. We place limits on the image buffer size, cropping the image if the width or height exceeds a threshold size (currently 10,000 pixels) to constrain memory usage. We also avoid extreme aspect ratios by non-uniformly scaling the image if the image dimensions differ by more than a factor of 2. In some cases, down- sampling a large overview can result in a "washed out" image. For large images, our thumbnail generation routine first modifies the visualization, removing high-frequency visual elements such as gridlines and element borders. In the resulting thumbnail, pixels with brightness over a threshold value are shifted towards the nearest color (in RGB color space) found in the color encoding palette (Figure 7.4).

7.2.4 *Navigating and Managing History*

By visualizing past analysis states, our graphical history display facilitates revisitation. Users can click a thumbnail to skip back to a prior state. If a user performs analysis operations while visiting a prior state, the history system creates a new analysis branch and shows it in the graphical history. However, as these histories can quickly become unwieldy, we have implemented additional techniques to reduce the complexity of the display and filter unneeded views. Figure 7.5 depicts our model and how it is mapped into a visual display.

Manual Editing

We support manual editing so that users can delete unwanted states from the history. As we use state-based logging, deleted states are simply removed from their history branch and do not impose side effects on other history items. However, one caveat is that deletion can result in an incomplete timeline in Worksheet History mode.

Chunking

When a group of related actions are performed in sequence, they may be better represented as a single higher-level event. For example, in a word processor the keystrokes [c][h][u][n][k] might be represented as the word [chunk]. To support such chunking our system provides hierarchical history items. In the spirit of Kurlander and Feiner [113], we have hand-crafted a set of "chunking" rules to coalesce actions into a grouped history item. As new states are added to the history, the rules evaluate if the new state should be chunked with the previous state. The history system applies a set of predicates expressing the chunking conditions and if any evaluate to true (and no exception rules do) the new state is chunked with the previous state.

In an analysis of user activity (sec. 7.3.3), we found that rapid sequences of formatting actions are common and could benefit from aggregation. Accordingly, we include a rule that chunks history items if the most recent state was the result of a formatting operation. Similarly, a quick succession of sort or filter actions (less than 30 sec. apart) likely indicate a multi-step configuration of the view and so we chunk them together. A rapid series shelf actions to build up (or take down) a view are also common, and so we chunk them when separated by less than 5 sec. We also support exception cases: large time durations—possibly indicating a break between sessions—prevents any chunking, as does bookmarking or annotating a state.

When our rules determine that two actions should be chunked, the thumbnail in the history view updates in-place and no new thumbnails are added to the view. Users can click in the history view to skip to a state prior to the chunked sequence. However, undo events will step back through each of the chunked actions individually. We believe that this interaction is less complicated than an interface providing explicit level-of-detail controls (e.g., [113]).

Undo-as-Delete

Undo actions may be the result of varied intentions. For example, an undo may be viewed as a navigation action, moving to a previous state with the intention of later rolling forward again. Alternatively, as Shipman and Hsieh [161] have noted, an undo may serve as a "delete" operation to recover from a mistake or an undesirable action. In our effort to improve the scalability of graphical histories, we hypothesized that most uses of undo fall within the latter category, such that "undone" states are rarely revisited. As described in section 7.3.3, we empirically tested this idea, finding that undo actions were over 12 times more common than redo actions.

As a result, we developed a new history management technique we call undo-as-delete. As a user performs actions, new items are added to the graphical history. When a user clicks the back button to perform an undo, the last state is removed from the graphical history. The underlying model still maintains the state, and thus subsequent redo actions work as expected, with the history item returned to the graphical history. However, if the user does not execute a redo before creating a new analysis branch, the undone branch is discarded as in a stack-based organization.

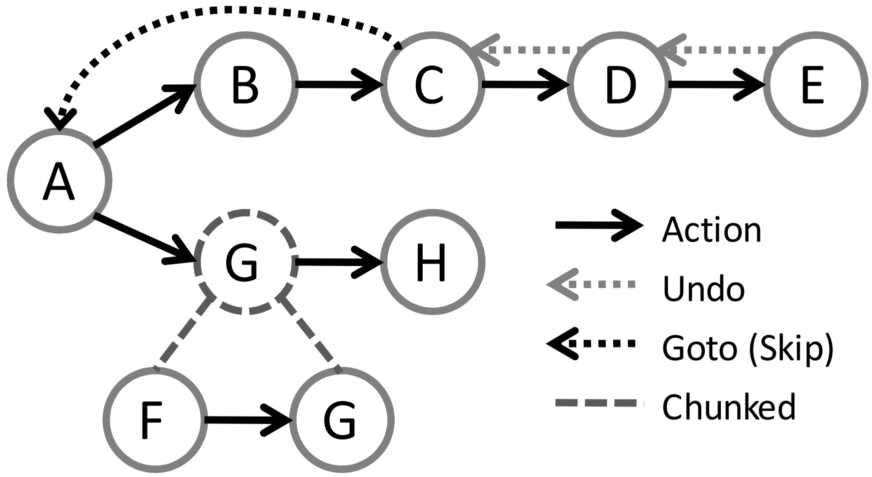


Figure 7.5a. History management. A user performs actions to go from state A to state E, performs two undo actions, and then skips back to state A. The user performs new actions to go to states F, G, and H. Chunking rules determine that states F and G should be coalesced.

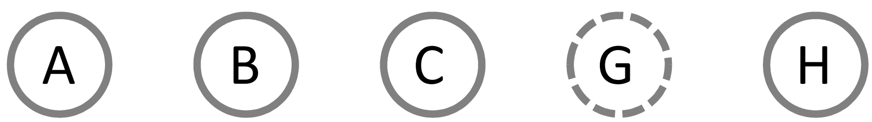


Figure 7.5b. Visual presentation of history model. The states in Figure 7.5a are presented in a linear sequence. States D and E are culled by undo-as-delete (§3.4.3), and states F and G are coalesced due to chunking rules (§7.2.4). Branches (starting at states B and G) are listed inline.

This approach enables our system to cull a large amount of unneeded history, reducing the complexity of the history model. Users can still create branching histories by navigating to past states using the graphical history rather than the undo button. New actions will then result in a new branch without deleting the previous branch, thus preserving the analysis trail. Similarly, we disable undo-as-delete if user interaction suggests a view is important: bookmarking or annotating the current state exempts it and previous states on the same branch from deletion. In future research we plan to see if other indicators such as selections might also indicate importance.

History Navigation and Management Example

Assume a user performs four actions in a row, as shown in Figure 7.5, to move through states A→B→C→D→E. The result is a linear list of states. The user then undoes the previous two actions, moving back to state C. Our undo-as-delete rules automatically hide states D and E to reduce the complexity of the history. The user next explores an alternative analysis, first skipping back to state A by clicking its thumbnail in the graphical history view, and then performing new operations to move through states A→F→G→H. When the user performs a new operation to go to state F, the undo-as-delete rules delete states D and E from the underlying model. Furthermore, when chunking rules determine that states F and G are similar, the system chunks the states into the single entry G in the graphical display. As shown in Figure 7.5b, the interface places the abbreviated branch G→H in sequence after branch B→C, as sibling branches are sorted by the timestamp of the first entry in the branch. Similarly, if the user were to skip back to state B and perform new actions, the interface would place the new branch in the list sequence directly after state C.

7.2.5 Operating on History

As discussed in our design space analysis, operations on history models can support sensemaking, search, and communication. Guided by these concerns, we have incorporated operations for affixing metadata to history states, dynamic querying of the history interface, and exporting histories to support sharing and presentation.

Metadata: Bookmarks and Annotations

By right-clicking an item, users can use a context menu to bookmark the state or add a text annotation (Figure 7.6b). Bookmarked views are then available in the Bookmarks mode of the history viewer. We have also considered adding a keyword tagging feature as a generalization of the bookmarking feature. Text annotations are available in tooltips when the mouse hovers over an item (Figure 7.3).

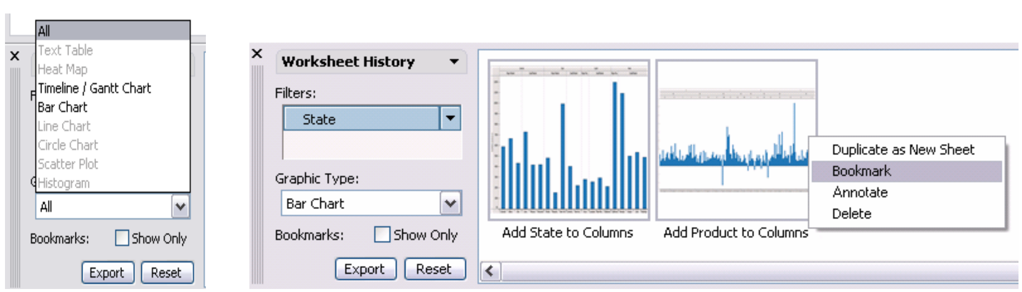


Figure 7.6. (a) Filter by chart type. A selection menu highlights the chart types available in the interaction history. **(b) Filtered history** showing bar charts that include the data field "State". A context menu provides operations on history items.

Search and Filter

Even with mechanisms for combating scale (sec. 7.2.4), histories can grow large. To help retrieve states of interest, we have introduced multiple search features (Figure 7.6). We hypothesized that the type of visualization and the visualized data fields are salient aspects with which users might recall past states. Our history viewer supports filtering by data field by reusing the shelf metaphor for visual encodings. Users can simply drop a data field into the history filter shelf to limit the view to only those states that include the data field. A combo box allows users to further limit the history view to specific chart types (Figure 7.6a). These filtering operations are implemented by indexing the VizQL expressions stored with each history item. Users can also use a checkbox to limit the view to bookmarked history items.

History Export and Sharing

Finally, the history viewer provides export features to share and communicate findings. By clicking the "Export" button, users can view a menu of export options. Our system can export selected history states as a saved Tableau file, allowing reloading of the states as a set of worksheets. Visualization views for selected history states can also be exported as either bitmap or vector images, and can be embedded in reports and presentations. By exporting Tableau visualizations in the Windows Metafile format, we can export a set of history states directly into PowerPoint slides as editable graphics. Analysts can automatically generate a slide deck from a set of selected history states and then annotate and edit exported visualizations in PowerPoint directly, as in Figure 7.7.

7.3 Using History to Evaluate Visualization Design

While the previous section focuses on graphical histories to support end-users, we also used histories to evaluate Tableau. By exporting and aggregating history logs, we can analyze user behavior. Here we present two analysis approaches and discuss some of our findings.

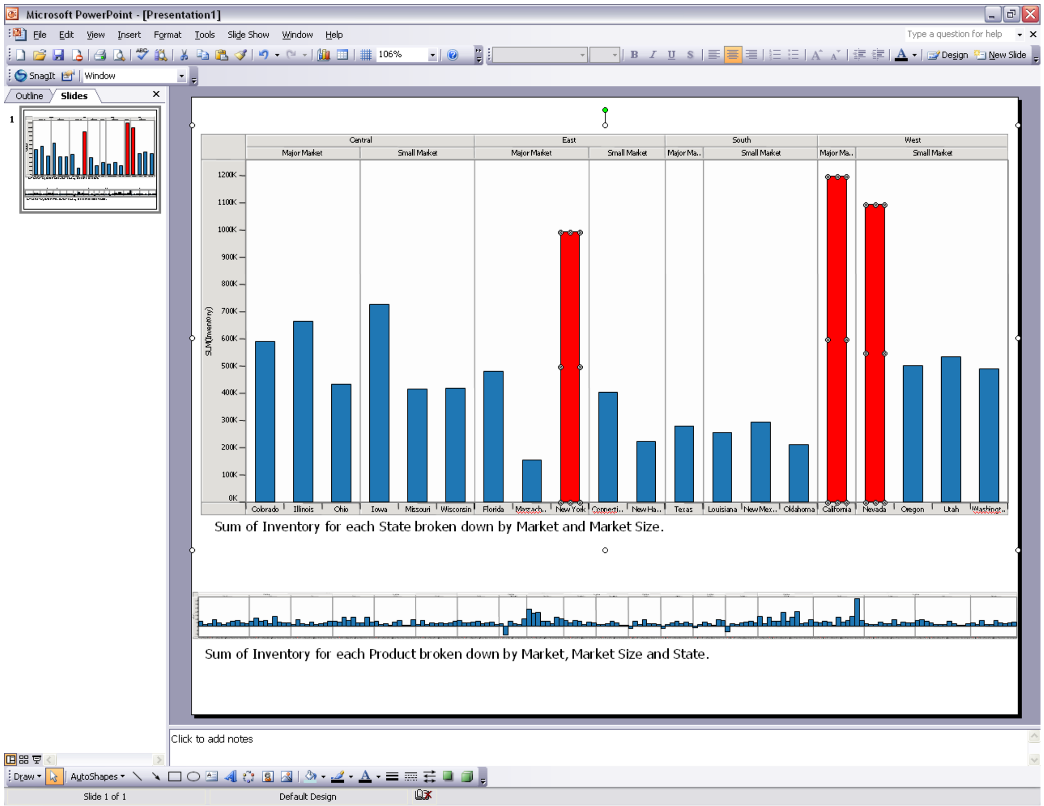


Figure 7.7. Tableau visualization exported into PowerPoint. The "Export" feature can seed presentations with captioned, editable versions of Tableau visualizations.

7.3.1 Analyzing Individual Usage with Behavior Graphs

We have explored tools for analyzing individual usage sessions. One technique we have found useful is behavior graphs, which we model after Card et al.'s web behavior graphs [36]. Figure 7.8 shows a Tableau session visualized in a behavior graph. The graph reads in a snake-like fashion. Actions are listed left-to-right except for Undo events, which are placed right-to-left on a new row. Subsequent actions resume left-to-right ordering on a new row. Vertical columns often contain the same state, making revisitation patterns clear. Color indicates the types of actions performed by users. We have found these visualizations particularly useful for understanding patterns of branching and revisitation.

Figure 7.8. Tableau behavior graphs depict behavior in an analysis session. Actions except undo and goto are placed left-to-right. Undo actions (red) move right-to-left on a new row. Goto actions (green) indicate navigation performed in the history viewer.

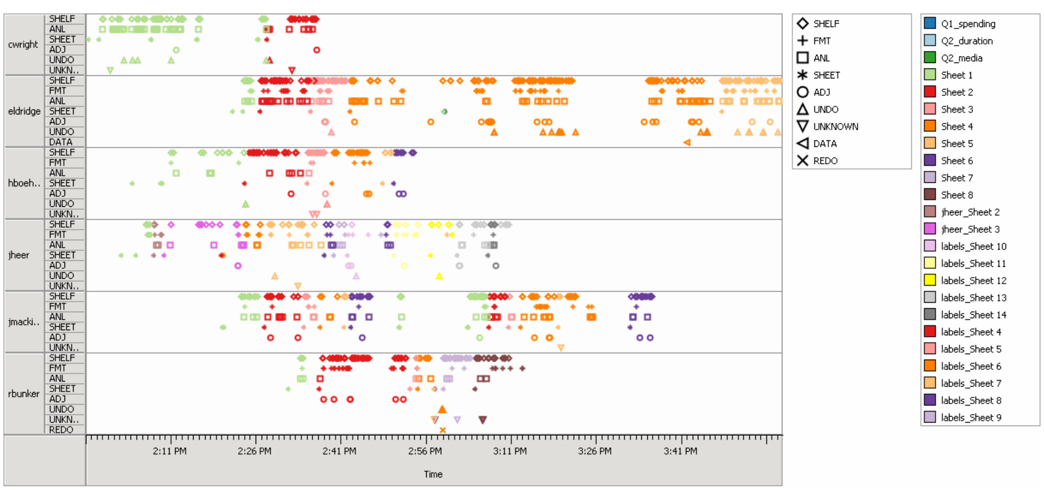


Figure 7.9. Aggregate analysis of Tableau usage. Each row shows the timeline for a different user. Shapes indicate command types; color indicates worksheet usage. The color patterns indicate different worksheet usage and revisitation patterns across users.

7.3.2 Analyzing Aggregate Usage

Analysis of aggregate usage is also important for determining usage patterns. For these and other history analysis tasks, we have used Tableau itself. First, we map each history log into a tabular format. Columns in these tables include timestamps, session ids, user ids, worksheet names, and actions performed. We store the resulting logs in a database which we then visualize in Tableau. Our taxonomy of commands (sec. 7.2.2) enables us to analyze command usage at multiple levels of granularity. Figure 7.2 shows Tableau visualizing the results of collected history logs: the primary display shows a histogram of command usage, while the graphical history display contains thumbnails for other analyses. An analysis of aggregated usage timelines is shown in Figure 7.9.

7.3.3 Findings

By analyzing user histories, we have made a number of findings to improve the design of Tableau's interface and estimate the impact of our history management techniques. Here we describe four such examples. We collected all usage data using a version of Tableau that includes our augmented history model, but without a graphical history interface. Usage data has been collected from 9 Tableau employees and 27 customers willing to share their data. The data consists of 20,192 actions from 36 users, with a median of 350 actions per user. Of these, 17,401 actions result in visual history items, as non-visual actions—such as opening a workbook or adding a derived field—are not included in the history interface.

The Undo / Redo Ratio

As we designed our history interface, we wanted information about how users used the existing undo and redo features. Looking at the usage logs, we found a total of 1,023 undo events and 82 redo events: undo was ~12.5 times more common than redo. Thus, most undone actions were never revisited, a finding that supports our undo-as-delete model for managing histories (sec. 7.2.4).

The Prevalence of Formatting

When analyzing command usage, we found that formatting actions, in which users adjust size and styling, accounted for 23.8% of all actions. Furthermore, format actions regularly occurred in succession: 73.6% of all formatting actions were followed by an additional formatting action. In response, we crafted chunking rules (sec. 7.2.4) that coalesce all formatting events. As sequences of consecutive resize events were common, our subsequent development effort has also focused on improving Tableau's automatic view sizing routines.

Use of Automated Presentation Tools

Tableau's automated presentation features (named "ShowMe") [125] help users create more perceptually effective visualizations. We used history data to evaluate usage of these features by end-users. For example, we found a relatively low rate of mark type adjustment (560 mark changes among 8,248 shelf changes, for a 6.8% error rate), suggesting that the automatic selection of mark types was helpful. We also discovered that analysts used ShowMe features throughout usage sessions, suggesting that ShowMe commands had become a regular part of their visual analysis.

Estimated Impact of History Management Techniques

Finally, we have used collected history data to estimate the savings provided by our chunking rules and undo-as-delete. Table 1 shows the number of states culled when applying our techniques to the collected history data; 61.7% of states are either removed or chunked. Thus, we might expect presented histories to be as little as 40% the size they would be without our techniques. We note, however, that this is an estimate from recorded data and as such does not include manual deletion of history items or the effects of bookmarking and annotation.

Table 7.1. Estimated reductions from history management.

Management Technique	Items culled	% culled
Undo-as-Delete	941	5.4%
Chunking Formatting Actions	4,139	23.8%
Chunking Filter & Sort Actions ($\Delta t \leq 30s$)	1,432	8.2%
Chunking Shelf Actions ($\Delta t \leq 5s$)	4,228	24.3%
Total Items Culled (out of 17,401)	*10,740*	*61.7%*

7.4 Summary and Future Work

In this chapter, we have introduced a design space analysis of history systems in the context of interactive visualization and used it to develop a prototype history interface for the Tableau visualization system. Our analysis served as a useful guide for navigating the design decisions we faced while architecting history interfaces to support visual analysis and communication. Our resulting history model integrates history management and undo/redo functionality and provides an editable, graphical history that supports branching analysis histories within worksheets and merged global histories across worksheets. Our graphical history interface allows revisitation of previous views and is designed to complement Tableau's visual analysis features by providing overview displays of visualization states both within and across worksheets.

Our history tools introduce a suite of novel features. Our undo-as-delete feature provides an empirically-motivated mechanism for helping improve the scalability of history displays, while preserving the capability for branching histories. Our search, filter, and annotation features enable users to retrieve previous visualization views based on the data fields involved, the type of chart, and bookmarked status. Users can then export selected history items to multiple formats, including presentations in which users can edit Tableau visualizations as native vector graphics.

We have also applied our history model to support evaluation of the Tableau system. Our visual analysis of history logs has inspired multiple improvements to Tableau's user interface, including better view sizing and automated presentation methods, and has informed the design of our graphical history tools.

In future work, additional mechanisms for managing history may be of help. For example, our chunking rules are hand-crafted and highly specific to Tableau. Could we develop a more general characterization of analytic tasks to reuse design knowledge across visual analysis tools? Another potentially useful feature would be automated estimates of the saliency (or "importance") of visited views. Such estimates could inform semantic zooming of history displays, chunking, and more automated forms of presentation generation. How should features such as timing, revisitation, and interaction influence such a model?

Another area for future research is to more richly explore the social application of graphical histories. How might we create summary graphical histories to provide a useful overview of collaborators' work? Could representations of aggregate use improve awareness and coordination, or assist social navigation? Future work on analysis histories might also further assist the creation of presentations. Our current approach enables manual selection of history views in conjunction with search and filtering, and export of those views into external media. Other visual analysis tools [31, 60, 85] have explored explicit sharing and story-telling features. Novel tools that use recorded user histories to structure presentations or story-telling may benefit social data analysis.

8 Animated Transitions in Statistical Data Graphics

In both analysis and presentation, it is common to view a number of related data graphics backed by a shared data set. For example, a business analyst viewing a bar chart of product sales may want to view relative percentages by switching to a pie chart or compare sales with profits in a scatter plot. Similarly, she may wish to see product sales by region, drilling down from a bar chart to a grouped bar chart. Analysts regularly perform such incremental construction of visualizations in tools such as Excel, Tableau, and Spotfire.

The visualization challenge posed by each of these examples is to keep the readers of data graphics oriented during transitions, such that they understand the correspondence between graphics. Ideally, viewers would accurately identify elements across disparate graphics and understand the relationship between the current and previous views. Staying oriented between views is particularly important in collaborative settings such as presentations and story-telling, where viewers not interacting with the data are at a disadvantage to predict the results of transitions. In the last chapter we showed how history tools support the creation of presentations and tours through a data set. In this chapter, we focus on the use of animation to better communicate the relationship between visualization views presented in sequence.

Animation is one promising approach to facilitating perception of changes when a view transitions between related data graphics. Previous research has found that animated transitions may help keep viewers oriented [150, 176], facilitate learning [9] and decision-making [77], and increase levels of engagement [176]. Users of sense.us

Previously published by the author and George Robertson in [90].

(CHAPTER 4) often cited the animated transitions between views as a favorite feature, suggesting that engaging animations may also provide a hedonic incentive for system use (CHAPTER 3). However, others have noted that animation can be problematic [7, 20, 176]. Animation is no guarantee of improved performance, involves timing issues that static depictions avoid, and may mislead if the animations violate the semantics of the data. Consequently, efforts to add animation to data graphics require careful study.

In this chapter, we investigate the design of animated transitions between statistical data graphics backed by a shared data table. We extend theoretical treatments of data graphics to include transitions and introduce a taxonomy of transition types. We then posit design guidelines for animated transitions and apply these principles in DynaVis, a visualization system featuring animated data graphics. Our primary contribution, however, is two controlled experiments conducted to assess the effects of animated transitions on object correspondence and value estimation tasks. We find that animated transitions significantly improve graphical perception at both syntactic and semantic levels of analysis.

8.1 Animation: A Double-Edged Sword

Animation has proven popular in user interfaces due in part to its intuitive and engaging nature. Moreover, the perceptual literature suggests that animation may be used to improve interaction and understanding. First, motion is highly effective at attracting attention, and unlike many other visual features is easily perceived in peripheral vision [140]. These results suggest that we may fruitfully apply animation to direct attention to points of interest. Second, animation facilitates object constancy for changing objects [140, 150], including changes of position, size, and color, and thus provides a natural way of conveying transformations of an object. Third, animated behaviors can give rise to perceptions of causality and intentionality [129], communicating cause-and-effect relationships and establishing narrative. Fourth, animation can be emotionally engaging [176, 186], engendering increased interest or enjoyment.

However, each of the above features can prove more harmful than helpful. Animation's ability to grab attention can be a powerful force for distraction. Designers abuse object constancy if an object is transformed into a completely unrelated object, establishing a false relation. Similarly, incorrect interpretations of causality may mislead more than inform. Engagement may facilitate interest, but can also make misleading information more attractive or may be frivolous—a form of temporal "chart junk" [172]. Additionally, animation is ephemeral, significantly complicating comparison of items in flux.

Furthermore, there remain a number of issues when applying animation, such as time/error tradeoffs. Animations that are too slow may prove boring or degrade task times, while those that are too fast may result in increased errors. Optimal times may be hard to predict and subject to both the complexity of the scene and the familiarity of the viewer. These and other issues have led some researchers to instead advocate the use of static depictions of changes [7, 176]. The upshot is that animation is a double-edged sword—designers must take both the benefits and pitfalls under consideration.

8.1.1 Principles for Animation Design

Given the vast design space available to animators and the pitfalls of animation misuse, researchers have proposed guidelines for crafting effective animations. Lasseter [117] shares principles of hand-drawn character animation, such as squash-and-stretch, exaggeration, anticipation, staging, and slow-in slow-out timing. Zongker and Salesin [202] selectively apply these principles to create animated presentations in their Slithy framework. They advocate making all movement meaningful, eschewing principles that promote the agency of animated items over the semantics of the animation, such as squash-and-stretch and exaggeration. On the other hand, they endorse the use of anticipation and staging to direct attention and partition animations such that only one action happens at a time.

The psychologists Tversky et al. [176] cast a skeptical eye on animation, finding no benefit for communicating the workings of complex systems when the animated and static depictions are informationally equivalent. However, they make an exception for animated transitions in visualizations and suggest two high-level principles for effective animation. Their Congruence Principle states "*the structure and content of the external representation should correspond to the desired structure and content of the internal representation*" and their Apprehension Principle states that "*the structure and content of the external representation should be readily and accurately perceived and comprehended.*" Interestingly, the congruence principle echoes Mackinlay's expressiveness criteria for automatic generation of static data graphics [124], suggesting that accepted guidelines for visualization might also be applied to animation. We revisit these principles in greater detail later in the chapter.

8.1.2 Animation in Information Visualization

Animation in interactive visualization has been a topic of research for over the last decade and a half. Some research has focused on systems issues, developing frameworks for applying animation in user interfaces. Hudson and Stasko [101] introduced user interface toolkit support for animation and the Information Visualizer [149] enabled animation and level-of-detail control with a cognitive coprocessor that was leveraged by a number of pioneering visualizations (e.g., [150]). Other research has focused on designing animations to facilitate perception. One approach is to use motion as an additional visual variable within which to encode data [5]. Another is to use animation to facilitate understanding of transitions between different states of an interface. We focus on this second approach.

Animated transitions have received much attention within tree visualization. Cone Trees [150] use animated rotations at multiple levels of a tree to bring selected items into view. Yee et al. [199] introduce valuable heuristics for animating transitions in radial tree layouts. SpaceTrees [145] and DOITrees [84] animate tree branches as they expand and collapse. Both apply staging, breaking up animations into distinct phases. For example, a transition within SpaceTree might involve first collapsing a subtree, translating the viewing region, and then expanding newly visible subtrees.

In many cases, the evaluation of animated transitions has relied on anecdotal evidence, leaving questions as to their effectiveness. Some systems, however, have been the subject of formal studies of animated transitions. StepTree [20], a 3D treemap visualization, uses animated fading and resizing to "zoom" into subtrees. A controlled experiment found mixed results in revisitation tasks: one set of users successfully used navigation shortcuts in animated conditions, while others made more errors relative to static transitions. Bederson and Boltman [9] found that animated transitions within a family tree explorer improved subjects' abilities to reconstruct the tree from memory, evidence of facilitated learning. Robertson et al.'s studies of polyarchy visualizations [151] found that use of animated transitions improved both task time and user satisfaction. Simple transitions (e.g., translation rather than rotation), lasting about 1 second, gave the best performance, though user preferences varied.

More recently, animated transitions have been applied within statistical data graphics. The Name Voyager [186] stacked area chart visualization uses animation when data is filtered, often including scale changes that involve animating gridlines and axis labels. These and other related uses of animation are applied in the visualizations within the Many Eyes [181] web service. Gapminder [71] uses animated data graphics in both presentation and analysis scenarios. Examples include movement of marks to convey change over time, subdivision of marks to indicate a drill-down operation, and shape morphing and translation to animate from a stacked area chart to a scatter plot.

While these visualizations have proven popular and engaging, little research has been conducted to characterize the design space of transitions between statistical data graphics and assess how animated transitions affect graphical perception. This paper seeks to take the first steps in filling the gap. We start by considering the various transitions a statistical data graphic might undergo.

8.2 Transitions between Statistical Data Graphics

As Kosslyn [112] describes, data graphics can be considered at three levels of analysis: syntax, semantics, and pragmatics. Syntax concerns the actual visual marks and their composition. Semantics focuses on the meaning of the graphic—the underlying data values and relations that the marks represent. Pragmatics focuses on connotations that go beyond the semantic interpretation. We limit our discussion to the first two: syntax and semantics.

Data graphics contain different classes of syntactic elements. These include framing marks such as axes and gridlines, identifying marks such as labels, and data-representative marks such as points, bars, and lines. Perceptual analysis at the syntactic level involves recognition of which class a mark belongs to and perception of visual properties such as position, shape, and color, both in absolute terms and relative to other marks. Analysis at the semantic level, on the other hand, requires that one associate these syntactic properties of the graph with the data they represent. This analysis involves identifying marks as representatives of specific data points and interpreting the absolute and relative values of visually encoded elements.

Both levels of analysis are needed to formally model the state of a data graphic. At the semantic level, one must represent the data dimensions (or schema) being visualized (often a subset of the full schema of the backing data table), filtering and ordering conditions, and the actual values of data elements. The resulting syntactic elements are determined by encoding operators that map the semantic description to visual objects with properties such as position, size, shape, transparency, color hue, and value [124].

We model transitions between graphics as state changes within this characterization. Analytic operators make changes to the semantic model of the data graphic, editing the data schema, data values, or visual mappings. This in turn results in changes to the graphical syntax. In static transitions, the original syntactic form is simply replaced with the new one. The challenge of designing animations is to visually interpolate the syntactic features such that the animation effectively communicates semantic changes.

8.2.1 A Taxonomy of Transition Types

To better inform the design of animated transitions, we crafted a taxonomy of the various types of transitions between data graphics. We identified the following transition types by considering the syntactic or semantic operators one might apply to a data graphic.

View Transformation

View transformations consist of a change in viewpoint, often modeled as movement of a camera through a virtual space. Examples include panning and zooming. View transformation is a purely syntactic operator as the data schemas and visual encodings remain unchanged.

Substrate Transformation

Substrate transformations consist of changes to the spatial substrate in which marks are embedded. Examples include axis rescaling and log transforms as well as bifocal and graphical fisheye distortions.

Filtering

Filter transitions apply a predicate specifying which elements should be visible. As a result, items are added or removed from the display. Filtering does not change visual encodings or data schemas, but a substrate transformation such as axis rescaling may be desired to improve space usage if the filtered view has a different range of values.

Ordering

Ordering transitions spatially rearrange ordinal data dimensions. Examples include sorting on attribute values and manual re-ordering.

Timestep

Timestep transitions apply temporal changes to data values. Apart from the time slice from which data is drawn, the data schema does not change. For example, a business analyst might transition between sales figures for the current and previous year. Axis rescaling may further refine the view for significant changes of value.

Visual Encoding Change

Visual encoding transitions consist of changes to the visual mappings operating on the data. For example, a user might visualize data first in a pie chart and then in a bar chart, or might edit the palettes used for color, size, or shape encodings.

Data Schema Change

Data schema transitions change the selection of visualized data dimensions. For example, starting from a univariate bar chart, one might wish to visualize an additional data column, resulting in a number of possible bivariate graphs. Such transitions may be accompanied by changes to the visual mappings, as one could present the bivariate data as a stacked or grouped bar chart, a scatterplot, or a small multiples display. Changes of schema may be orthogonal, in which an independent dimension is added or removed, or nested, in which the schema change traverses a hierarchical relation between dimensions of the data table, such as roll-up and drill-down operations.

8.2.2 Design Considerations for Animated Transitions

Before crafting transitions for the types identified above, we sought principles to guide our design process. After reviewing literature in perception, visualization, and user interface design, we arrived at the following considerations. Our guidelines take the form of specific recommendations for adhering to Tversky et al.'s [176] Congruence and Apprehension principles of effective animation.

Congruence

Maintain valid data graphics during transitions. To ensure viewers' mental models are congruent with the semantics of the data, we suggest that, as much as possible, intermediate interpolation states remain valid data graphics. While some violations are unavoidable, such as during shape deformations, this rule seeks to minimize unwarranted attributions to the data. Entailments of this principle include avoiding uninformative animation, and considering the relation between axes and the data marks during transitions.

Use consistent semantic-syntactic mappings. To aid understanding, similar semantic operators should have suitably similar transitions across different types of data graphics. For example, a designer might standardize the filtering of items in and out of the display across graphic types to improve consistency and learnability.

Respect semantic correspondence. If syntax violates semantics, viewers might make poor interpretations. For example, we should not use marks representing specific data points to depict different data points across a transition. Thus some data schema changes should involve the removal and addition of marks even if the data graphic type remains unchanged. In multivariate conditions, where marks may correspond to multiple values, designers must apply nuanced judgment.

Avoid ambiguity. Avoid ambiguous semantics across transitions. For example, timesteps in bar charts could involve animated changes of bar heights. The same animation might be used in a data schema change in which an unrelated variable is swapped into the bar chart. However, not only do such animations abuse object constancy (see previous guideline), the ambiguity increases the risk of misinterpreting the transition. Ideally, semantic operators should have noticeably different transitions.

Apprehension

Group similar transitions. The Gestalt principle of Common Fate [140, 198] states that objects that undergo similar visual changes are more likely to be perceptually grouped, helping viewers to understand that elements are simultaneously undergoing the same operation.

Minimize occlusion. If objects occlude each other during a transition, they will be more difficult to track, potentially harming perception [198].

Maximize predictability. An animation can reduce cognitive load and improve tracking if the target state of a transitioning item is predictable after viewing a fraction of its trajectory. This observation suggests slow-in slow-out timing—not only are starting and ending states emphasized, the acceleration profile may improve spatial and temporal predictability.

Use simple transitions. Complicated transforms with unpredictable motion paths or multiple simultaneous changes result in increased cognitive load. Simple, direct transitions alleviate confusion, impose less memory burden, and improve predictability. Perceptual research provides evidence that people find translation and divergence (expand/contract) motions easier to understand than rotation [15].

Use staging for complex transitions. Some transitions are inherently complex and do not lend themselves to simple transitions. In such cases, one can break up the transition into a set of simple sub-transitions, allowing multiple changes to be easily observed. For example, separating axis rescaling from value changes may aid change tracking.

Make transitions as long as needed, but no longer. Transition stages and dwells between them must be long enough for accurate change tracking, but when too slow can result in longer task times and diminished engagement [7, 151]. Prior research [34, 151] suggests transition times around 1 second, though transitions with minimal movement can likely be performed faster. Empirical testing may be needed to determine optimal parameters.

8.3 Animated Data Graphics in DynaVis

Guided by the transition taxonomy and design principles, we built DynaVis, a visualization framework supporting animation and direct manipulation of data graphics. In this section we describe a subset of the animated transitions DynaVis supports, including those found in Figures 8.1-8.5. We also note here that all our animations use slow-in slow-out timing with a quadratic velocity profile.

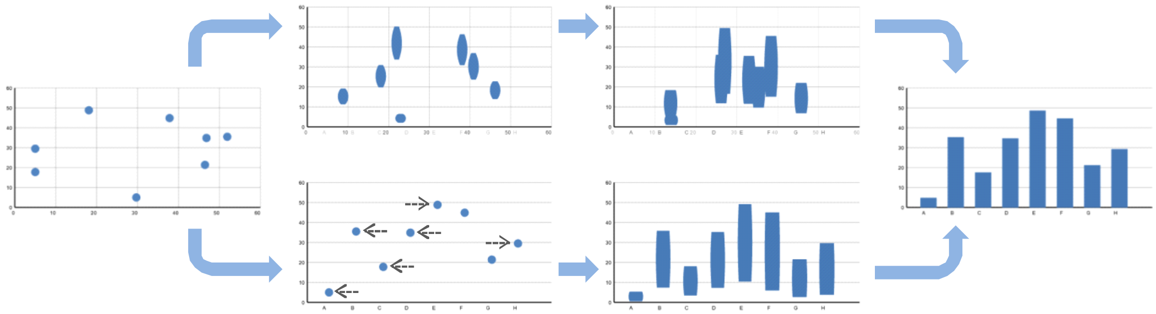


Figure 8.1. Animating from a scatter plot to a bar chart. The top path interpolates between the starting and ending states. The bottom path is staged: the first stage moves points to their x-coordinates and updates the x-axis, the second morphs points into bars.

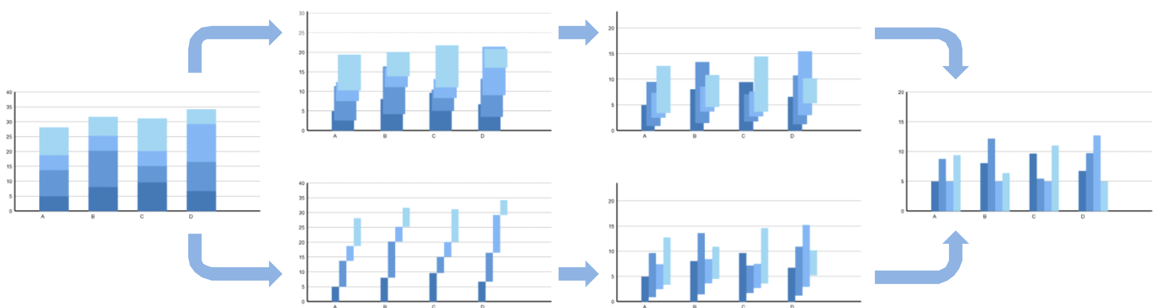


Figure 8.2. Animating from stacked bars to grouped bars. The top path interpolates between the starting and ending states. The bottom path is staged: the first stage changes the widths and x-coordinates of bars, the second drops the bars to the baseline.

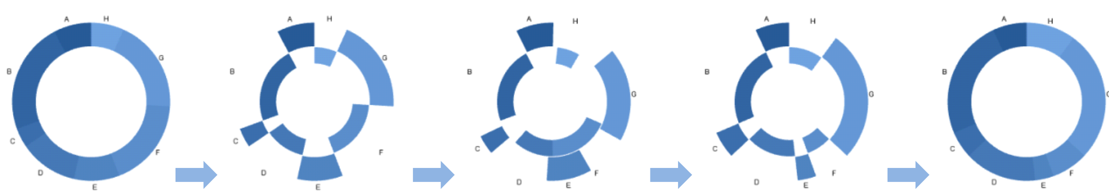


Figure 8.3. A multi-stage animation of changing values in a donut chart. Stage 1: Wedges split into two rings. Stage 2: Wedges rotate until centered on their final position. Stage 3: Wedge values update, changing size. Stage 4: Wedges reunite into a single ring.

Filtering

Different data graphics afford different techniques for the entry and exit of filtered items. For example, bars in a bar chart may grow from a baseline or layers in stacked area chart might fall from the "sky" (as in [71]). While such behaviors may be engaging, we instead opted for a consistent presentation across data graphics by fading items in and out using alpha blending. This design also avoids the non-meaningful changes inherent in these other movements.

Sorting

A straightforward sorting animation directly translates the positions of elements. While this improves on static transitions, we noticed that occlusion sometimes complicated object tracking, particularly when three or more items overlapped. In response, we implemented staggering, issuing small delays in movement onset to subsequent elements. This separates items' starting and ending times, making small but noticeable decreases in the amount of overlap.

Substrate Transformation

Large changes of value may require axis rescaling. To make such changes clear, axis labels and gridlines move to depict scale changes, smoothly fading in and out when added and removed. For example, when changing from a quantitative to an ordinal scale, old labels and gridlines first fade out and then new ones fade in. Axis animation also communicates other changes, such ass transitions from linear to log scale. We suspect that axis animations may also assist learning of different scales.

Timesteps

For most changes of value over time, we animate the change directly, such as changing the heights of bars in a bar chart. Timesteps may require an axis rescaling performed in a separate stage either before or after the value change, as appropriate. However, in cases such as stacked bars, pie, and donut charts, items may translate while also changing size. To separate these changes, we experimented with more extreme stagings that separate translation and size changes. To construct multi-stage animations that avoid occlusion sometimes required unintuitive animations, such as the multi-ring configuration for donut charts in Figure 8.3.

Visualization Changes

For changes in visualization type, we applied the design guidelines above to move and reshape elements. For example, to go from a bar chart to a pie or donut chart, we morph bars into wedges and interpolate positions in polar coordinates (c.f., [199]). However, the conventional clockwise order of radial graphs causes massive occlusion, as interpolating marks travel overlapping paths. DynaVis resolves the issue by using counter-clockwise ordering for radial graphs. Similarly, direct interpolation of stacked bars to grouped bars creates occlusion (Figure 8.2). Instead, we interpolate x-coordinates and widths first, and y-coordinates and heights in a second stage.

Data Schema Changes

Data schema changes can prove complicated, potentially changing both the visible data and its visualization. Figure 8.1 depicts animation from a scatter plot to a zero-aligned bar chart, in which bivariate points become univariate bars. The backing data table remains constant but the visualized dimensions change: the transition removes the quantitative variable on the x-axis and replaces it with nominal labels. Direct interpolation of this change translates and morphs items simultaneously. DynaVis instead transitions to a dot plot first, updating the x-axis and interpolating horizontal positions. A second stage grows the points into bars. We treat other orthogonal schema changes similarly.

Nested schema changes such as drill-down may involve both filtering and visualization changes. For example, drill down in a bar chart segments bars to form a stacked bar chart, a transition to grouped bars (Figure 8.2) might then follow. Similarly, scatter plot points can split or merge upon drill-down and roll-up.

In data schema changes, animation is only appropriate when there is a data dimension shared between the starting and ending states. Without a shared structure between graphics, animation may be ill-defined or misleadingly convey false relations. In such cases, we advocate using either static or dissolve transitions (as in cinema) to indicate the independence between graphics.

8.3.1 Implementation Notes

We implemented DynaVis in the C# programming language using the Direct3D graphics framework. We define data graphics such as bar charts and scatter plots using a bundle of separate visual encoding functions that assign position, shape, color, transparency, and other visual properties to data marks, axes, gridlines, and labels. Each of these encoding functions are decoupled from the transition machinery.

However, we do not assign visual variables directly to visual items. Instead, we assign values to a special *Transitioner* object used to help construct transitions.

A centralized *TransitionManager* is responsible for constructing animated transitions and invoking the necessary visual mappings. The *TransitionManager* is similar in some respects to the Information Visualizer's cognitive coprocessor [149], supporting interpolation as well as composite parallel and sequential transitions. In fact, a *Transitioner* object is a specialized parallel transition of a set of visual items.

All analytic operations (sorting, drill down, etc) are routed through the *TransitionManager*, which then builds the resulting transition. Transition construction involves executing visual encoding functions on *Transitioner* objects and then applying timing and staging operators on the results. For example, duration and delay operators determine timing, while composition operators aggregate sub-transitions into parallel or sequential transitions. A splitting operator decomposes a single *Transitioner* into multiple transitions. For example, horizontal and vertical movements might be split into separate stages of movement. The split operator takes as input a *Transitioner* object, a predicate for matching visual items to process, and a set of visual variables to extract, outputting a new parallel transition involving the extracted variables. Finally, the staggering operator assigns delays to sub-transitions, spacing out the starting times within an otherwise parallel transition. All transitions have been hand-coded into a rule system using a simple transition description language consisting of the above operators. Future work is needed to investigate both automatic determination and direct manipulation of transition descriptions.

Within a single stage of animation, interpolation of visual variables typically involves linear interpolation of values (or polar interpolation in radial graphs). DynaVis supports smooth morphing of shapes by interpolating between polyhedral meshes defining shape surfaces. To ensure performance, our mesh generation routines were crafted to provide predetermined vertex correspondences, enabling interpolation of mesh vertices without the need for costly vertex correspondence calculations.

8.4 Evaluation of Animated Transitions

Though guided by design principles, crafting animated transitions still involves a number of trade-offs. We need empirical data to gauge the actual effectiveness of transitions. In this section, we present two experiments that assess the effect of animated transitions on graphical perception. We describe our experimental designs and present the results, deferring detailed discussion to the next section.

Twenty-four subjects (10 female, 14 male), all from the greater Puget Sound area, participated in both experiments. Subjects ranged from 26 to 62 years of age (M = 49.6, SD = 10.7). Subjects were screened for familiarity with common data graphics and came from professions requiring the use of data graphics, including small business owners, college professors, analysts, and administrators.

Both experiments were conducted using standard desktop PCs. Subjects were seated in front of 21" LCD monitors running at 1600 x 1200 pixel resolution; each visualization occupied 1000 x 600 pixels.

8.4.1 Experiment 1: Object Tracking

Our first experiment was designed to test the effects of animated transitions at the syntactic level of analysis. Subjects were asked to follow two objects across a transition and identify the locations of the objects in the final graphic. As accurate object correspondence is a prerequisite to further comparison, we believe this provides a useful measure of a transition's effectiveness.

Six transition conditions were chosen to provide coverage of the taxonomy of section 3.1. The transitions tested were bar chart to donut chart (visualization change), stacked to grouped bars (drill-down), sorting a bar chart (ordering), scatter plot to bar chart (data schema and visualization change), zoom and filter in a scatter plot (both rescaling and filtering), and timestep in a scatter plot (timestep and occasional rescaling). In pilot testing, we noticed a reliance on labels in the bar to donut and sorting transitions, so to better study the effects of animation on both data marks and labels, we also added versions of these transitions without labels.

Figure 8.4. Experiment 1 trial stimulus. Subjects were shown a data graphic and two target objects were highlighted; the initial display was visible for 3 seconds. This was followed by a static or 1.25-second animated transition. The display was masked 3 sec. after transition onset. Subjects then clicked where they believed the target objects to be. The sequence above depicts an animated bar chart to donut chart transition.

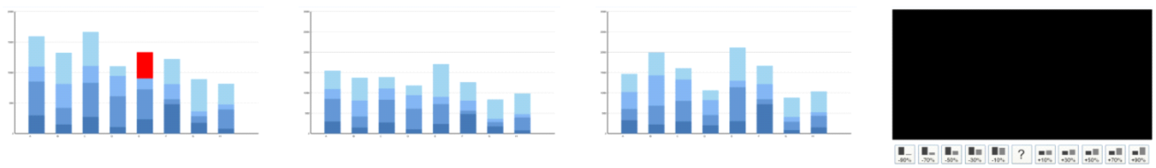


Figure 8.5. Experiment 2 trial stimulus. Subjects were shown a data graphic and a single target object. This was followed by a static or 2-second animated transition. The display was masked 3 seconds after transition onset. Subjects provided estimates of the percentage change of the target object, using buttons ranging from -90% to +90% in 20% increments. A '?' button was provided for situations of uncertainty. The sequence above depicts a staged animation involving scale and value changes of stacked bars.

As shown in Figure 8.4, in each trial subjects were first shown an initial data graphic. Two targets were sequentially highlighted in the graph, the first in red and the second in orange. After the initial graph was visible for 3 seconds, a transition would begin. Static transitions were immediate; animated transitions were 1.25 seconds in duration. We masked the display 3 seconds after the transition onset, at which point subjects were to click the final locations of the targets. To prevent "cheating," we required that subjects keep the mouse pointer in a bounded region away from the graphic until the display was masked. We instructed subjects to make their best guess if unsure and to click the center of the display if they had no guess.

We used informal pilot studies to test other variants of this task. Using only a single target allowed subjects to ignore much of the transition, limiting generalizability. We also tried a reversed version, in which subjects viewed a transition and identified where selected items had come from. This task, however, proved too error prone to be useful.

The experiment used a 3 (Animation) x 2 (Size) within-subjects design for each transition type. The size condition varied between 8 elements (4x4=16 in the case of stacked bars) and 16 elements (8x4=32 in the case of stacked bars). The animation condition varied between static transitions, animated transitions where all changes were directly interpolated, and various forms of staged animation. Each subject performed 6 replications of the 3*2*8=48 cells for a total of 288 trials. We counter-balanced all trials to ensure equal data distributions and target sizes across conditions.

Staging in the bar to donut and sorting cases involved staggering animation onsets for each element with short delays to reduce occlusion. All others involved non-overlapping stages. We staged the stacked to grouped bars transition by first changing the widths of bars and then having them fall into place. In the scatter plot to bar chart condition, we first move scatter plot points horizontally, then morph them into bars. In the remaining scatter plot conditions, we performed rescaling separately from either the filtering or timestep operation.

The dependent measure was average error, which we measured as the average pixel distance from the location of subjects' mouse clicks to the respective target objects. We computed error optimistically, such that if participants accidentally clicked the targets in reverse order their error rate would not be adversely affected.

Results

The results for animation conditions are shown in Figure 8.6, finding a strong advantage for animation. Repeated Measures ANOVA found significant differences at the .05 level for each transition type ($F(2,286) >= 22.03$, $p < 0.001$). Post-hoc comparisons between animation and staged animations using Fisher's LSD test were significant at the .05 level for the Zoom & Filter ($p = 0.026$) and Timestep Scatter Plot ($p = 0.002$) conditions. Sort Bars ($p = 0.051$) and Bar to Donut ($p = 0.071$) differences were significant at the .10 level. Timestep Scatter Plot is the only transition in which staged animation has more error than direct animation. In this case, there were two transitions (a rescale and then movement) in a short time period, potentially compounding opportunity for error.

Analysis across the size condition revealed that tracking error increased with size in all conditions except the Stacked to Grouped Bars transition. Repeated Measures ANOVA results for all transition types except Stacked to Grouped Bars, Zoom & Filter, and Timestep Scatter Plot were significant at the .05 level ($F(2,143) >= 19.13$, $p < 0.001$). Increasing the number of elements noticeably increased error rates in the Bar to Donut transitions when labels were removed, but a similar interaction did not take place in the Sort Bars transition.

8.4.2 Experiment 2: Estimating Changing Values

Our second experiment focused on the semantic level of analysis. We asked subjects to follow a single target across a transition and estimate the percentage change in value in the underlying data. Our goal was to test the hypothesis that animation facilitates graphical perception of changing values over time. Experiment 2 used the same 3 x 2 within-subjects design as before. However, Experiment 2 involved only four transitions: timesteps in Scatter Plot, Grouped Bars, Stacked Bars, and Donut Chart displays. Subjects performed 6 replications of the 3*2*4=24 cells for a total of 144 trials.

Staged animation for Scatter Plot and Grouped Bars conditions consisted of axis rescalings (if needed) followed by timestep animations. In the Stacked Bars and Donut Chart conditions we tested highly staged animations, such that objects never change position and value simultaneously. For Stacked Bars, this meant that each stack level would update separately, starting from the top stack sequentially down to the bottom stack. For Donut Charts, this involved the multi-stage animations of Figure 8.3.

Figure 8.5 depicts a sample trial for Experiment 2. We presented subjects an initial graphic for 3 seconds before transition onset, with only a single target highlighted. We lengthened animations to 2 seconds in this experiment to comfortably accommodate the multi-stage animations. We masked the display after 3 seconds, at which point a panel of buttons appeared with which the user could enter their estimate of the target's percentage change in value. The buttons ranged from -90% to +90% by increments of

20% and indicated percentage change both textually and graphically. We instructed subjects to make their best guess estimate, or click a '?' button if they were at a loss.

The dependent measure was estimation error, measured as the percentage the smaller value was of the larger, regardless of order. This measure more equitably handles proportional differences in value (i.e., in percentage change, -50% halves the value and +90% almost doubles it, while in the adjusted measure the differences are -50% and +52.6%). In pilot tests, we tried using this measure as the response variable, but it proved less intuitive than percentage change. Before the experiment, participants were informed of the difference between negative and positive changes, and practice trials revealed correct answers so subjects could calibrate their estimates.

Results

The results for animation conditions are shown in Figure 8.7. Repeated Measures ANOVA results were significant at the .05 level for the Scatter Plot ($F(2,286) = 257.82$, $p < 0.001$), Grouped Bars ($F(2,286) = 20.25$, $p < 0.001$), and Donut Chart ($F(2,286) = 3.183$, $p = 0.043$) transitions, but not for Stacked Bars ($F(2,286) = 1.50$, $p = 0.224$). Although staged animation had lowest average error for both the Scatter Plot and Grouped Bars, post-hoc analysis found no significant differences between animated conditions. For the Donut Chart, animation was significantly more accurate than both static ($p = 0.043$) and staged animation ($p = 0.024$) transitions.

Figure 8.7 also depicts the distribution of unknown ('?') responses, where subjects were unwilling to make an estimate. Static transitions were much more likely to result in unknown responses, as were transitions involving scale changes. Axis rescaling appears to have increased estimation difficulty for all animation conditions.

For the size condition, Repeated Measures ANOVA results are significant at the .05 level only for the Donut Chart ($F(2,183) = 15.54$, $p < 0.001$) condition, for which the error rate was significantly lower when more elements were present. For all other conditions, size did not have a significant effect.

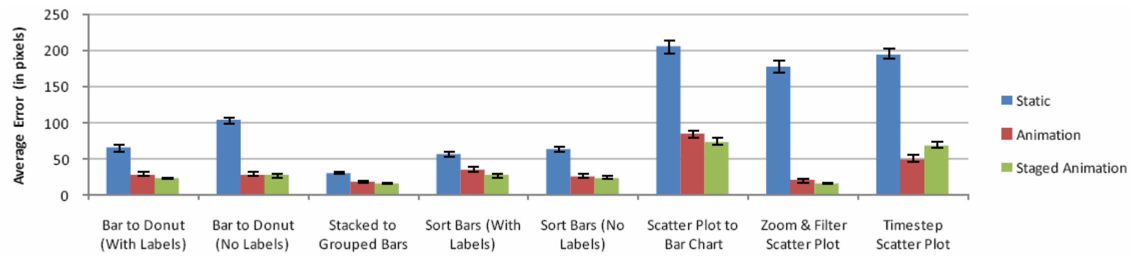


Figure 8.6. Experiment 1 results for animation conditions. Animation is sig. better than static across all conditions. Except for Timestep Scatter Plot, staged animation outperforms animation. Post-hoc analysis finds sig. differences between animation and staged animation at the .05 level for Zoom & Filter and Timestep Scatter transitions and at the .10 level for Bar to Donut and Sort Bars transitions.

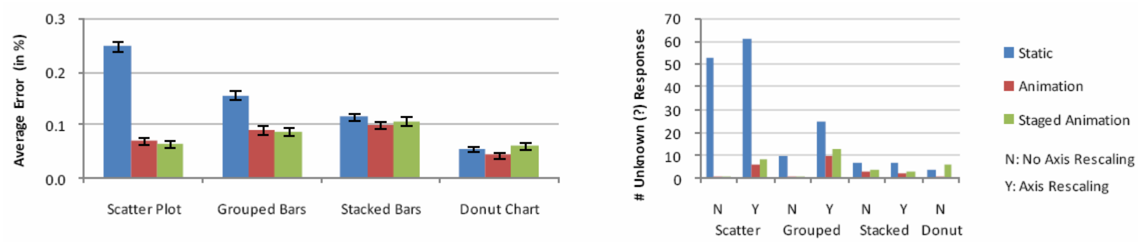


Figure 8.7. Experiment 2 results for animation conditions. **Left:** For Scatter Plot and Grouped Bars conditions, animation sig. outperforms static transitions. Staged animation outperforms animation, but not significantly. Stacked Bars show no sig. difference, while animation is sig. better than static transitions and staged animation in the Donut Chart. **Right:** The number of unknown (?) responses was higher for static transitions, but occurred for animation conditions when axis rescaling was performed.

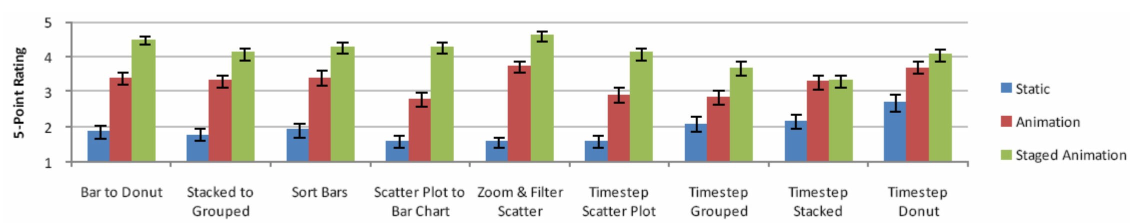


Figure 8.8. Preference survey results. Overall, staged animation is preferred to animation, which is preferred to static transitions. Statistically significant differences are found for all transition types. Post-hoc analysis finds that preference for staged animation is sig. at the .05 level for all transitions except the Timestep Stacked Bars and Timestep Donut conditions, in which an extreme form of staging was applied.

8.4.3 Subjective Preferences

After the experiments, subjects completed a survey measuring their preferences. For each transition in the experiments, subjects rated static transitions, animation, and staged animation on a five-point Likert scale according to how effectively they conveyed the changes between graphics, with 5 indicating most effective. The resulting ratings are shown in Figure 8.8. An ANOVA was conducted on ratings for each transition type; all were significant at the .05 level. For all transition types except Timestep Stacked Bars and Timestep Donut, post-hoc analysis found that staged animation was significantly preferred to animation ($p < 0.003$ in all cases). For the remaining two transitions, no significant difference between animation conditions was found ($p = 1$ and $p = 0.322$, respectively), mirroring the increased error for staged animation in these conditions in Experiment 2. In all cases, both animations were preferred to static transitions ($p < 0.001$).

Subjects also responded to a set of overall preference questions, again measured using a five-point Likert scale. Subjects reported that animated data graphics made it easier to understand transitions ($M = 4.20$, $SD = 0.66$) and were fun and engaging ($M = 4.54$, $SD = 0.59$). Subjects also responded that they would use animated transitions in their own data analysis ($M = 4.17$, $SD = 0.64$) and presentations ($M = 4.36$, $SD = 0.77$). Subjects expressed a desire to use animated data graphics immediately, including a college instructor who felt they would help her more effectively teach data graphics to her students.

8.5 Discussion

We now discuss the experimental results, identifying trends of interest, suggesting best practices, and noting areas in need of further inquiry.

8.5.1 Animation Improves Graphical Perception

The main result of the study was that animation improved graphical perception over static transitions at both syntactic (object tracking) and semantic (change estimation) levels of analysis. Even in highly predictable transitions, such as the stacked bars to grouped bars conditions, animation had a significantly lower error rate. As we masked each trial stimulus, the better performance in highly predictable cases may in part be due to improved transfer to memory. Survey results also revealed strong preferences for animation, as subjects rated it more helpful and engaging. Furthermore, staged animation was significantly preferred to direct animation in most cases. This argues strongly for the efficacy of animation for depicting transitions between data graphics.

8.5.2 Trade-Offs Between Design Principles

The experimental results also shed some light on the trade-offs involved between competing design principles, as principles that aid object tracking might not always aid semantic analysis. For changes of value within a scatter plot, object tracking error was significantly higher with staged animation, in which axis rescaling and value changes occurred in separate stages. We hypothesize that these multiple stages with shorter durations provide more opportunities for losing targets. However, staged animation resulted in more accurate change estimation (though not significantly so) and was significantly preferred. Multiple subjects further commented that staging was less demanding and that they preferred slower animations (stages were faster in Experiment 1). As a result, we endorse the use of staged animation for scatter plots, but recommend timing each stage at a full second, rather than a half-second each.

Other trade-offs involved the use of heavy staging in stacked bars and donut charts in Experiment 2. On one hand, multi-stage transitions separate value changes from translations, potentially improving change estimation. On the other hand, they are more complicated. Our performance results give more weight to the latter concern, as heavily staged animation resulted in increased error. These were also the only cases in which preference ratings for staged animation were not significantly higher—evidence for user preference reliability. The multi-stage examples proved overly complex, arguing that it is preferable to minimize unnecessary motion than perform “do one thing at a time” [192] staging. Finally, most subjects laughed upon first viewing the multi-staged stacked bars transition. This might prove less than desirable during a presentation of one’s analysis.

8.5.3 The Case for Staging

Overall, simple staging proved helpful, though the advantages are not overwhelming. Except for value changes in scatter plots, staging had lower error rates for object tracking, in some cases significantly so. We suspect this was largely due to minimizing occlusion. This suggests that other techniques that reduce the effects of occlusion, such as alpha blending and outlining marks, might further improve object tracking. Simple staging (e.g., separating axis rescaling from value changes) also had significantly higher preference ratings and lower (though not significantly so) error rates for change estimation. As a result, we recommend the use of simple staging, but believe further study is needed to reliably assess the effects of multi-stage transitions. Future experimentation is particularly needed in regards to timing and dwells, as we included no pauses between stages except for that provided by slow-in slow-out timing.

8.5.4 The Effects of Axis Rescaling: Avoid If Possible

Axis rescaling made change estimation difficult, increasing overall error and the number of unknown ('?') responses. However, the use of animation tempered these effects, suggesting that movement helped subjects make sense of scale changes. The results suggest that, if possible, common scales should be used across timesteps to remove the need for axis rescaling. For cases where axis rescaling is needed, subjects significantly preferred staged animation. Furthermore, we believe our animations could be improved—our animations faded axis gridlines in and out during the scale change, sometimes removing landmarks in mid-transition. Retaining grid lines through the scale change, and then fading them out gently after all other transitions have been completed, may improve perception of changes.

8.5.5 The Intricacies of the Donut: Smaller Wedges Are Better?

Though not directly related to the design of animated transitions, our experiments revealed some interesting properties of donut charts. First, change estimation errors were noticeably lower for the donut chart than other graphs, an interesting observation given the ongoing debate over the efficacy of radial graphs (c.f., [67, 163]). Additionally, donut charts are the only graphic for which performance significantly improved as the number of elements increased. As the number of donut wedges increases, their average size decreases. Smaller wedges are more rectilinear, exchanging angular judgment for more accurate length judgment [50]. Furthermore, smaller items may be generally more amenable to change estimation, at least up to a lower bound; a hypothesis supported by Weber's Law of psychophysics [50]. This suggests that similar benefits might be achieved in bar charts through appropriate sizing. Further study is needed to evaluate this possibility.

8.6 Summary

In this chapter, we have explored the effects of animated transitions on graphical perception of changes between related data graphics. Two controlled experiments found significant advantages for animation across both syntactic and semantic tasks, providing strong evidence that, with careful design, animated transitions can improve graphical perception of changes between statistical data graphics.

We began by situating transitions in a theoretical model of data graphics, developing a taxonomy of transition types. We introduced perceptually-motivated design principles for crafting animated transitions and used them to develop transitions within our DynaVis visualization framework. We then presented a pair of experiments conducted with 24 participants balanced across age, gender, and professions, investigating the effectiveness of static transitions, animation, and staged animations for both syntactic (object tracking) and semantic (value change estimation) tasks.

In addition to finding significant advantages for animation, our experiments provided further insights. There was evidence that staged animation, such as staggered movements to reduce occlusion and separate stages for axis rescaling and value changes, provide additional benefits. This claim is strongly backed by subject preferences and consistently (though at times marginally) supported by error measures. The results further discourage the use of complex multi-stage transitions, favoring simple staging over aggressive "do one thing at a time" [202] staging. Still, further study into the use of timing and dwells is needed. Study results suggest additional improvements, such as including techniques to mitigate occlusion, avoiding axis rescaling when possible, and persisting axis gridlines as landmarks when rescaling is unavoidable. Furthermore, a potentially interesting interaction was observed between smaller mark sizes and increased accuracy of change estimation.

Overall, subjects were highly enthusiastic about animated data graphics, and felt that it facilitated both improved understanding and increased engagement. The vast majority of participants wanted to use animated data graphics in their own analysis and presentation. Some participants even went to lengths after the study to thank us for "allowing" them to participate, and expressed impatience for the release of animated data graphics in commercial products.

9 Conclusion

This dissertation identifies a short-coming in the current paradigm of visualization research: despite the social nature of visual media, most research to date relies upon a single-user model of visual analysis. This limitation inhibits teams from engaging in social processes of sensemaking that can improve the coverage and depth of analysis and foster the dissemination of findings. In response, this thesis contributes new principles and systems for enabling collaborative data analysis with visualizations.

9.1 Review of Thesis Contributions

The central problem addressed by this thesis is how to design visualization systems that support and catalyze social sensemaking by analysts collaborating asynchronously. To that aim, we synthesized results from research on social psychology, computer-supported cooperative work, and peer-production to develop design considerations (CHAPTER 3) to guide the development of social visual analysis tools.

We applied these considerations to the design of *sense.us* (CHAPTER 4), a web-based visual analysis environment featuring novel mechanisms for sharing and discussing visualized data. The site contributes novel collaboration mechanisms, including a doubly-linked discussion model and view linking techniques that more tightly couple textual commentary and visualization states. We also conducted usage studies of sense.us in the lab and in a live deployment, resulting in the first empirical characterization of asynchronous social sensemaking with online visualizations. We found that social interaction catalyzed cycles of observations, questions, and hypotheses, enriched subjects' interpretation of the data, and spurred additional analysis sessions. To our knowledge sense.us is the first end-to-end system effectively coupling direct visual data analysis with social interpretation and deliberation.

Our design considerations and experiences with sense.us also identified an important set of sub-problems that suggested new components for improving social data analysis:

- Unobtrusive *awareness and social navigation cues* that analysts can use to allocate their attention in accordance with others' actions.
- Robust *pointing and annotation techniques* for referring to and selecting dynamic data subject to any number of visual representations.
- *History and presentation interfaces* for constructing, sharing, and viewing tours and presentations for telling analysis stories.

We addressed these needs through the design, implementation, and evaluation of interface techniques for a variety of analysis systems.

We developed *scented widgets* (CHAPTER 5)—standard user interface controls imbued with visual navigation cues—to provide enhanced social navigation cues for social data analysis environments. We contributed guidelines for the design of visualizations embedded within UI controls and a toolkit architecture with which developers can easily add a variety of visual information scent cues to user interface widgets. We also conducted a controlled study using scented widgets to visualize collective visitation and commenting activity within sense.us and found that subjects used scented widgets to identify and visit both popular views and under-explored regions of the data.

Our *generalized selection* techniques (CHAPTER 6) serve as the basis for collaborative annotations that apply to time-varying data across a wide range of visual encodings. We contributed direct manipulation techniques for authoring selection queries of visualized data and an interactive query relaxation engine that enables users to construct more complicated queries by generalizing from a simpler, initial selection. In a controlled study, users created significantly more accurate selections of visualized data using our techniques, potentially improving clarity of communication.

We also contributed *graphical histories* (CHAPTER 7) that record the analysis process and facilitate subsequent sharing by enabling review and revisitation of previously visited states and by generating presentations from a selection of views. Our history interface was informed by a design space analysis of the options and trade-offs involved in architecting history systems and introduces new techniques for improving history management, search, and visualization. Such history tools can be a valuable accompaniment to social data analysis, supporting shared activity histories and dissemination of successful analysis patterns in addition to aiding story-telling.

Lastly, we examined the design of *animated transitions* (CHAPTER 8) that better communicate the relationship between subsequent views in an analysis story. We contributed design principles for creating effective animations and conducted a pair of formal experiments finding that appropriate animated transitions can both increase viewer engagement and improve viewer's ability to understand how consecutive visualization views are related.

Taken together, the design principles, systems, and interaction techniques described in this thesis demonstrate effective ways of facilitating social forms of sensemaking with interactive visualizations.

9.2 Recent Developments

Since this thesis work began, a number of new web applications have been introduced that support asynchronous collaborative visualization and which have been influenced by this dissertation research. Websites such as Swivel.com [169] provide social-network-style platforms for conversation around data, along with basic charting capabilities, and has proven popular with bloggers. Tableau Software launched its Tableau Server product [170], which much like Spotfire's DecisionSite Posters [164] allows users to collaborate asynchronously with intranet-based visualizations. Little has been published about usage of these systems, however.

One new system where results have been reported is the Many Eyes website [181]. Many Eyes is freely available on the public internet and allows users to upload their own data. Unlike data-oriented sites like Swivel, Many Eyes lets users apply more than a dozen interactive visualization techniques. Users may then have discussions about the visualizations, though the collaboration capabilities are more basic than in sense.us. The experiences on and around the site [181] lend support to the idea that visualization can catalyze discussion. While these discussions can be analytical, they also can be purely social, partisan, or game-like. In addition, the move from a closed setting to the public internet has made clear that these discussions can be highly distributed [54], with a significant proportion of collaboration occurring off the site using an embedded visualization. Designing for this type of multi-site conversation suggests a whole new set of challenges for facilitating discussion and awareness.

Another recent development is the collaborative generation of data monitored and presented largely through visualizations. One example is the aggregation of sensor data from sensor networks or mobile phones, such as in the Personal Environmental Impact Report project [178]. Another new phenomenon is eye-witness reporting of everything from train and bus conditions to election monitoring (e.g., [177]), shared through services such as Twitter. More work is needed to both characterize and further the reach of such community-driven information ecologies.

Still, these systems have yet to provide rich collaboration mechanisms such as those explored in this thesis. As data visualization becomes a first-class citizen on the Web, we hope mechanisms for collaborative analysis will also become commonplace.

9.3 Limitations and Future Work

By helping initiate research into social data analysis, this thesis opens the door to new lines of inquiry; we hope it serves as a prelude to a continuing stream of research. As suggested by the design considerations in CHAPTER 3, there is a great deal of future work that can be done to further collaborative data analysis. Here we elaborate some of the limitations of this thesis and corresponding opportunities for future research.

9.3.1 Synthesizing Collaborative Contributions

As described in §3.1.2, we might further improve collaborative analysis systems by designing shared artifacts that coordinate collaboration and provide a means for integrating contributions. Beyond textual discussion, what external representations will support collaborative analysis? How do such artifacts affect grounding and the cost of integration? How can individual contributions be better synthesized? We might (semi-)automatically merge separate data views (e.g., [21]) to form aggregated contributions. Prior work in evidence matrices [18], argumentation systems [79], and analytic "sandboxes" [195] also suggest possible representations. Future research might consider more complicated linking structures, such as tying discussion to multiple views, as well as conducting formal evaluations of the effects of varied discussion models on grounding and integration. Other beneficial methods might include summarization techniques or visual representations such as meta-visualization of social activity and contributions. In general, treating contributions such as comments, annotations, tags, votes as data that can in turn be visually analyzed could provide a powerful substrate to support synthesis and reflection in social data analysis.

9.3.2 Pointing, Naming, and Reference

A central issue in supporting asynchronous collaboration with visualizations is referring to trends, outliers, and data regions in a display (§3.3). This thesis has explored both free-form graphical annotations (CHAPTER 4) and data-aware visual query techniques (CHAPTER 6) for pointing and referring to visualizations. Still, given the importance of reference to successful communication, future work might provide further benefits. For example, hybrid selection techniques that couple the strengths of data-aware and free-form annotations might prove more usable than visual queries (c.f., §6.5, 6.6) while providing facilities for selections to persist across views or time-varying data. Interesting challenges arise when dealing with aggregated data. For example, pivot tables and charts (e.g., [167]) are commonly used for analysis in business intelligence, and are constructed by aggregating values (e.g., summing, averaging). Thus an individual mark in a visualization may correspond not to a single tuple in the original source data, but a collection of tuples. Should annotations of this data apply only to the aggregated data, or can they be meaningfully tied to the source tuples as well? Going forward, we also suspect that analysts will want to refer not just to data elements or visualized trends, but to users' comments, annotations, and other contributions. To support reference to all such artifacts, future systems will require a more general mechanism for naming and referring to the various elements within the analysis environment.

9.3.3 Computation as Collaborator

The thesis work presented here, and indeed much of the current crop of social software on the web, focuses on interactions between people and information, with computational technology used primarily as a communication medium. An area of future work is to explore how computation might become a first-class collaborator in social data analysis. How might statistics and data mining algorithms be incorporated in the analysis process? A straightforward example would be the use of visualization to present and explore the output of data mining routines (c.f., [86, 157]), which analysts might then discuss and debate. Another interesting application would be the use of pattern matching to suggest trends either positively or negatively correlated with a current trend of interest, or to run a confirmatory analysis of a hypothesis. However, such applications may increase the complexity of the interface and require research characterizing their design space. Computation might also be used to help allocate effort and suggest tasks to collaborators. For example, mining past contributions, user profiles, and inferred social networks may enable systems to productively direct collaborators to tasks in need of attention. Future research exploring the potentially interleaved roles of people and software in analysis systems might provide better ways of leveraging the unique capabilities of each.

9.3.4 Supporting the Information Life-Cycle

This thesis has focused on collaborative analysis using data sets and visualizations that we pre-selected. This simplification enabled us to develop and study systems explicitly for visual analysis. However, as indicated in Figure 3.1, visual analysis is a relatively late stage in the visualization pipeline.

One limitation of this work is that it neglects the laborious process of data preparation underlying most visualizations. Issues of data collection, cleaning, integration, and formatting are an "elephant in the room" of visualization research. The Web makes the problem more acute, as the world's important data is stored not only in the massive databases of government and industry, but in thousands of small tables scattered across the web in various formats. New tools are needed to enable search, discovery, and extraction of relevant data, lower the threshold for data cleaning and integration, and leverage collaboration to amortize the costs involved. Such tools will in turn require new methods for tracking and visualizing data provenance [11] to support transparency. These challenges lie in the intersection of HCI, information retrieval, and database research and go hand-in-hand with effective social data analysis.

Other parts of the information life-cycle are also ripe for study. Improved data acquisition tools should also support collaborative information foraging. How can groups coordinate their search for source materials? Recent work has begun to investigate these questions for internet search [132, 133]; more work is needed in the context of data analysis. Once the data has been collected, there may be many ways to visualize it. Current web-based tools [169, 181] provide a library of pre-built visualization "widgets." How might more powerful visualization authoring and customization tools enhance analysis? For example, visualization authoring tools applied to a data model for collaborative contributions (§9.1) would create opportunities for communities to construct their own shared representations of social activity. Lastly, it appears that many data-driven discussions will be distributed across the web [54]. How can the fruits of collaborative analysis be more effectively exported and embedded in external media such as web pages, e-mail, and presentations?

9.3.5 Evaluating Social Data Analysis

Given the wide variety of use cases, evaluation poses another challenge for social analysis systems. In this thesis we have used a variety of methods, ranging from a mixed-methods study combining qualitative observations, quantitative measures, and content analysis (CHAPTER 4) to formal, quantitative experiments of specific interface techniques (CHAPTER 5, 6, 8). Rigorously characterizing social interactions "in the wild" across myriad web services and social contexts is an important, yet difficult, proposition. Furthermore, the potential influence of social context, including familiarity, hierarchical relations, and social capital among collaborators, further complicates the design of replicable, controlled experiments. An important piece of the puzzle is determining relevant measures and objects of study. Our experiences with this thesis work suggest that important candidates include discussion-reply structures, use of deictic references, coverage of the data and visualization state space, rate and scale of contributions, and, if expressible, quality of outcome. Still, there is much yet to learn in this space and some methodological flexibility is likely a virtue. As we are still in need of more exploratory study, we suspect grounded theory approaches such as content analysis [115], which emphasize the bootstrapping of theories from collected data, will prove a useful and pragmatic means of making sense of social sensemaking.

9.3.6 Applications of Social Data Analysis

Another important avenue for future work is in applications research, applying and extending the work presented in this thesis to more targeted domains. One domain is education, where collaborative visualization could be used to teach data analysis and statistics to students using real-world data to address real-world problems. A second domain is scientific research, where the techniques described here could be integrated with more highly-specialized workflows. For example, many data analysts use programming tools such as MATLAB and R. How might such command-line driven interfaces be supported with collaborative interaction around data, analysis procedures, and visualizations? Third, the large amount of public data available on the Web—supporting causes such as public health and political transparency—is a rich resource that has only begun to be tapped. How can we improve the availability of this data in a manner that enables easy visualization and sharing across the Internet? As noted previously, better data management, visualization creation, and dissemination interfaces are needed to enable public uptake and hopefully, as Giovannini (CHAPTER 1) envisions, ultimately improve democracy and welfare.

9.4 Closing Remarks

The amount of information available to us continues to increase at a dizzying rate. Visualization, in concert with data management technologies, is a means to keep abreast of this rising tide and convert information into insight. But visualization alone will not be enough. The magnitude of the data at hand and the diversity of expertise needed to fully analyze it demand more: that our information interfaces enable us to work together to more effectively forage, analyze, point, argue, and disseminate. This dissertation hopes to push us closer towards systems that marshal our collective wisdom to make sense of the information that surrounds us.

Bibliography

1 Agrawala M., Beers A., Froehlich B., Hanrahan P., MacDowall P., Bolas M. The Two-User Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space. In Proc. *ACM SIGGRAPH 1997*: 27–332. 1997.

2 Ahlberg C., Shneiderman B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. *ACM Conference on Human Factors in Computing Systems (CHI'94)* 1994; 313–317.

3 Anupam V., Bajaj C.L., Schikore D., Schikore M. Distributed and Collaborative Visualization. *IEEE Computer 1994*, **27**(7): 37–43.

4 Ayers E.Z., Stasko J.T. Using Graphic History in Browsing the World Wide Web. *Proc. World Wide Web*, 1995.

5 Bartram L. Enhancing Visualizations with Motion. In *Proc. IEEE InfoVis 1998*, May 1998.

6 Baudisch P., Rosenholtz R. Halo: a technique for visualizing off-screen objects. In *Proc. ACM CHI 2003*: 481–488. Fort Lauderdale, FL. Apr. 2003.

7 Baudisch P., Tan D., Collomb M., Robbins D., Hinckley K., Agrawala M., Zhao S., Ramos G. Phosphor: explaining transitions in the user interface using afterglow effects. In *Proc. ACM UIST 2006*: 169–178. Oct. 2006.

8 Becker R.A, Cleveland W.S. Brushing Scatterplots. *Technometrics* 1987; **29**(2): 127–142.

9 Bederson B.B., Boltman A. Does Animation Help Users Build Mental Maps of Spatial Information? In *Proc. IEEE InfoVis 1999*: 28, San Francisco, CA, Oct 1999.

10 Benbunan-Fich R., Hiltz S.R., Turoff M. A comparative content analysis of face-to-face vs. asynchronous group decision making. *Decision Support Systems* archive 2003; **34**(4): 457–469.

11 Benjelloun O., Das Sarma A., Halevy A., Theobald M., Widom J. Databases with Uncertainty and Lineage. *VLDB Journal,* **17**(2):243–264. March 2008.

12 Benkler Y. Coase's Penguin, or, Linux and the Nature of the Firm. *Yale Law Journal,* **112**(369). 2002.

13 Benko H., Ishak E.W., Feiner S. Collaborative Mixed Reality Visualization of an Archaeological Excavation, *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2004)* 2004; 132–140.

14 Berlage T.A. Selective Undo Mechanism for Graphical User Interfaces based on Command Objects. *ACM Transactions on Computer-Human Interaction*, **1**(3): 269–294, 1994.

15 Bertamini M., Proffitt D. Hierarchical Motion Organization in Random Dot Configurations. In *Journal of Experimental Psychology: Human Perception and Performance*, **26**(4):1371–1386, 2000.

16 Bertin J. *Semiology of Graphics: Diagrams, Networks, Maps.* Madison, Wisconsin: The University of Wisconsin Press, 1967/1983.

17 Bertin J. *Graphics and Graphic Information Processing*. Walter de Gruyter, 1982.

18 Billman D., Convertino G., Shrager J., Pirolli P., Massar J. Collaborative Intelligence Analysis with CACHE and its Effects on Information Gathering and Cognitive Bias. *Human Computer Interaction Consortium Workshop* 2006.

19 Björk S., Redström, J. Window Frames as Areas for Information Visualization. In *Proc. of the Second Nordic Conference on Human-Computer Interaction*: 247–250. Aarhus, Denmark. 2002.

20 Bladh T., Carr D.A., Kljun M. The Effect of Animated Transitions on User Navigation in 3D Treemaps. In *Proc. Information Visualisation 2005*: 297–305, Jul 2005.

21 Brennan S.E. How conversation is shaped by visual and spoken evidence. In: Trueswell, Tanenhaus (Eds). *Approaches to studying world-situated language use: Bridging the language-as-product and language-as-action traditions*. MIT Press: Cambridge; 2005. 95–129.

22 Brennan S.E., Mueller K., Zelinsky G., Ramakrishnan I.V., Warren D.S., Kaufman A.. Toward a Multi-Analyst, Collaborative Framework for Visual Analytics, *IEEE Symposium of Visual Analytics Science and Technology* ,2006.

23 Brin S., Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proc. Seventh International Conference on World Wide Web*: 107–117, 1998

24 Brodlie K., Brankin L., Poon A., Banecki G., Wright H., Gay A. GRASPARC: a problem solving environment integrating computation and visualization. *Proc. IEEE Visualization*: 102–109, 1993.

25 Brodlie K.W., Duce D.A., Gallop J.R., Walton J.P.R.B., Wood J.D. Distributed and Collaborative Visualization. *Computer Graphics Forum* 2004; **23**(2): 223–251.

26 Brooks, F.P. The *Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley; 1975.

27 Brown J.S., Duguid P. *The Social Life of Information*, Harvard Business School Press, Boston. 2000.

28 Brush, A. J., Bargeron, D., Gupta, A., and Cadiz, J. J. Robust annotation positioning in digital documents. In *Proc. of CHI 2001*. pp. 285–292. (2001)

29 Brush A.J., Bargeron D., Grudin J., Gupta A. Notification for shared annotation of digital documents, In *Proc. ACM CHI 2002*. 2002.

30 Burt R.S. Structural holes and good ideas. *American Journal of Sociology* 2004; **110**(2): 349–399.

31 Caillois R. *Man, Play, and Games*. Free Press of Glencoe; 1961.

32 Cadiz J., Gupta A., Grudin J. Using Web Annotations for Asynchronous Collaboration Around Documents. In *Proc. CSCW 2000:* 309–318, 2000.

33 Callahan S.P., Freire J., Santos E., Scheidegger C.E., Silva C.T., Vo H.T. Managing the Evolution of Dataflows with VisTrails. *Proc. IEEE Workshop on Workflow and Data Flow for Scientific Applications (SciFlow)*, 2006.

34 Card S.K., Moran T., Newell A. *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates. 1983.

35 Card S.K., Mackinlay J.D., Shneiderman B. *Readings in Information Visualization: Using Vision To Think*. Morgan-Kaufmann; 1999.

36 Card S.K., Pirolli P., Van Der Wege M., Morrison J.B., Reeder R.W., Schraedley P.K., Boshart J. Information Scent as a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability. *Proc. ACM CHI*: 498–505, 2001.

37 Carroll J., Rosson M.B., Convertino G., Ganoe C.H. Awareness and teamwork in computer-supported collaborations. *Interacting with Computers*, **18**(1):21–46, 2005.

38 Carter S., Mankoff J., Goddi P. Building connections among loosely coupled groups: Hebb's rule at work. *Journal of Computer-Supported Cooperative Work*, **13**(3):305–327 2004.

39 Chen H. Compound Brushing. *Proc. IEEE InfoVis '03*, pp. 181–189. Oct 2003.

40 Cheshire C. Selective Incentives and Generalized Information Exchange. *Social Psychology Quarterly* 2007; **70**(1).

41 Chu W.W., Yang H., Chiang K., Minock M., Cho, G., Larson C. CoBase: A Scalable and Extensible Cooperative Information System. *Journal of Intelligence Information Systems*, 6(2):223–259. 1996.

42 Chu, W.W., Liu, S. CoXML: Cooperative XML Query Answering. In B. Wah (ed.), *The Encyclopedia of Computer Science and Engineering*. John Wiley & Sons Inc, 2007.

43 Chuah M.C., Roth S. Visualizing Common Ground, *Information Visualization (IV)*: 365–372. 2003.

44 Chui Y.-P., Heng P.-A. Enhancing View Consistency in Collaborative Medical Visualization Systems Using Predictive-Based Attitude Estimation, *First IEEE International Workshop on Medical Imaging and Augmented Reality (MIAR '01)*: 292. 2001.

45 Churchill E. F., Trevor J., Bly S., Nelson L., Cubranic D. Anchored conversations: chatting in the context of a document. In *Proc. ACM CHI 2000*: 454–461. 2000.

46 Clark H.H. Pointing and placing. In: Kita S (Ed). *Pointing. Where language, culture, and cognition meet*. Erlbaum; 2003. 243–268.

47 Clark H.H., Wilkes-Gibbs D. Referring as a collaborative process. *Cognition*, **22:** 1–39. 1986.

48 Clark H.H., Schreuder R., Buttrick S. Common ground and the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior,* **22**: 245– 258. 1983.

49 Clark H.H., Brennan S.E. Grounding in Communication. In: Resnick LB, Levine RM, Teasley SD (Eds). *Perspectives on socially shared cognition*. American Psychological Association; 1991. 127–149.

50 Cleveland W.S., McGill R. Graphical Perception and Graphical Methods for Analyzing Scientific Data. *Science*, 229: 828–833. 1985.

51 Cleveland W.S. *The Elements of Graphing Data*. 2nd ed. AT&T Bell Laboratories. 1994.

52 Cleveland W.S. *Visualizing Data*. AT&T Bell Laboratories. 1993.

53 Cummings J. Work groups, structural diversity, and knowledge sharing in a global organization. *Management Science*, **50**(3): 352–364. 2004.

54 Danis C.M., Viégas F.B., Wattenberg M., Kriss J. Your Place or Mine? Visualization as a Community Component. In *Proc. ACM CHI 2008*: 275–284. Florence, Italy. 2008.

55 Derthick M., Kolojejchick J. A., Roth S. An Interactive Visual Query Environment for Exploring Data. In *Proc. ACM UIST '97*: 189–198. Oct 1997.

56 Derthick M., Harrison J., Moore A., Roth S.F. Efficient multi-object dynamic query histograms. In *Proc. IEEE InfoVis 1999*: 84. Oct. 1999.

57 Derthick M., Roth S.F. Enhancing Data Exploration with a Branching History of User Operations. *Knowledge Based Systems*, **14**(1-2): 65–74, March 2001.

58 Dietz P.H., Leigh D.L. DiamondTouch: A Multi-User Touch Technology. In *Proc. ACM UIST*: 219–226. 2001.

59 Donath J.S. Identity and Deception in the Virtual Community. In: Smith M, Kollock P (Eds). *Communities in Cyberspace*. Routledge; 1998.

60 Dorling D., Barford A., Newman M. Worldmapper: The World as You've Never Seen It Before. *IEEE Transactions on Visualization and Computer Graphics*, **12**(5):757–764. Sep/Oct 2006.

61 Dourish P., Belotti V. Awareness and coordination in shared workspaces. In *Proc. ACM CSCW*: 107–114. 1992.

62 Dourish P., Chalmers M. Running Out of Space: Models of Information Navigation, *Human Computer Interaction (HCI'94)* 1994.

63 Eccles R., Kapler T., Harper R., Wright W. Stories in GeoTime. In *Proc. IEEE VAST.* 2007.

64 Edwards W.K., Igarashi T., LaMarca A., Mynatt E.D. A Temporal Model for Multi-Level Undo and Redo. In *Proc. ACM UIST*: 31–40, 2000.

65 Eick S. Data Visualization Sliders. In *Proc. ACM UIST 1994*: 119–120. Marina del Rey, California. Nov. 1994.

66 Ellis S.E., Groth D.P. A Collaborative Annotation System for Data Visualization. In *Proc. of Advanced Visual Interfaces 2004*. Gallipoli, Italy. 2004.

67 Few S. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Oakland, CA: Analytics Press. 2004.

68 Fishkin K., Stone M. Enhanced Dynamic Queries via Movable Filters. In *Proc. ACM CHI 1995*: 415–420. May 1995.

69 Gaasterland, T. Cooperative Answering through Controlled Query Relaxation. *IEEE Intelligent Systems*, 12(5):48–59. Sep-Oct 1997.

70 Gamma E., Helm R., Johnson R., Vlissides J. Command, in *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley: 233–242, 1995.

71 Gapminder. http://www.gapminder.org

72 General Dynamics. Command Post of the Future. [WWW document] http://www.gdc4s.com/content/detail.cfm?item=2a58f8e2-ef2b-4bb1-9251-42ee4961dd7f (accessed 28 November 2007).

73 Gergle D., Kraut R.E., Fussell S.R. Language efficiency and visual technology: Minimizing collaborative effort with visual information. *Journal of Language & Social Psychology*, **23**(4): 491–517. 2004.

74 Gigone D., Hastie R. The Common Knowledge Effect: Information Sharing and Group Judgment. *Journal of Personality and Social Psychology*, **65**: 959–974. 1993.

75 Goffman E. *The Presentation of Self in Everyday Life*. Anchor Books: New York; 1959.

76 Golder S.A., Huberman B.A.. The Structure of Collaborative Tagging Systems. *Journal of Information Science*, **32**(2). April 2006.

77 Gonzales C. Does Animation in User Interfaces Improve Decision Making? In *Proc. ACM CHI 1996*: 27–34, Vancouver, BC, Apr 1996.

78 Goodell H., Chiang C., Kelleher C., Baumann A., Grinstein G. Collecting and Harnessing Rich Session Histories. *Proc. Int. Conf. on Information Visualization*: 117–123, 2006.

79 Gordon T.F., Karacapilidis N. The Zeno argumentation framework, *Sixth International Conference on Artificial Intelligence and Law*, ACM Press: New York; 10–18. 1997.

80 Grudin J. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM*, **37**(1):92-105, 1994.

81 Hardin R. The Free Rider Problem. In: Zalta EN (Ed). *The Stanford Encyclopedia of Philosophy*, 2003.

82 Harrison S., Dourish P. Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems. In *Proc. ACM CSCW 1996*:67–76. Boston, MA. 1996.

83 Hastie R. Experimental Evidence on Group Accuracy. In: Grofman B., Owen G. (Eds), *Decision Research*; **2**:129–157. JAI Press; 1986.

84 Heer J., Card S.K. DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. In *Proc. Advanced Visual Interfaces 2004*: 421-424, Gallipoli, Italy, June 2004.

85 Heer J., Card S.K., Landay J.A. prefuse: A Toolkit for Interactive Information Visualization. In *Proc. ACM CHI 2005*: 421-430. April 2005.

86 Heer J., boyd d. Vizster: Visualizing Online Social Networks. In *Proc. IEEE InfoVis 2005*: 32–39, October 2005.

87 Heer J. Socializing Visualization, *CHI 2006 Workshop on Social Visualization* 2006.

88 Heer J, Agrawala M. Software Design Patterns for Information Visualization, *IEEE Transactions on Visualization and Computer Graphics*, **12**(5): 853–860. Sep/Oct 2006.

89 Heer J, Viégas F, Wattenberg M. Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization. In *Proc. ACM CHI 2007*: 1029–1038. 2007.

90 Heer J., Robertson G.G. Animated Transitions in Statistical Data Graphics. *IEEE Trans. on Visualizations and Comp. Graphics*. **13**(6):1240–1247, Nov/Dec 2007.

91 Heer J., Agrawala M. Design Considerations for Collaborative Visualization. In *Proc. IEEE VAST 2008*: 171–178. Sacramento, CA, 2007.

92 Heer J., Agrawala M., Willett W. Generalized Selection via Interactive Query Relaxation. In *Proc. ACM CHI 2008*:959–968, 2008.

93 Heer J., Agrawala M. Design Considerations for Collaborative Visualization. *Information Visualization*, **7**(1):49–62, 2008.

94 Heer J., Mackinlay J.D., Stolte C., Agrawala M. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Trans. on Visualization and Comp. Graphics*, **14**(6):1189–1196, Nov/Dec 2008.

95 Herlocker J., Konstan J., Terveen L., Riedl J. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, **22**(1):5–53, 2004.

96 Hightower R., Ring L., Helfman J., Bederson B., Hollan J.D. Graphical Multiscale Web Histories: A Study of PadPrints, *Proc. ACM Hypertext and Hypermedia*: 58-65, 1998.

97 Hill J., Gutwin C. Awareness support in a groupware widget toolkit. In *Proc. ACM SIGGROUP 2003*: 258-267. Sanibel Island, FL. Nov. 2003.

98 Hill W.C., Hollan J.D., Wroblewski D., McCandless, T. Edit wear and read wear, In *Proc. ACM CHI 1992:* 3–9. 1992.

99 Hill W.C., Hollan J.D. Deixis and the Future of Visualization Excellence, *IEEE Visualization*: 314–319. 1991.

100 Hochheiser H., Shneiderman B. Dynamic query tools for time-series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3:1–18. 2004.

101 Hudson S.E., Stasko J.T. Animation support in a User Interface Toolkit: Flexible, Robust, and Reusable Abstractions. In *Proc. ACM UIST 1993*: 57–67. Atlanta, Georgia, Nov 1993.

102 Huh S.-Y., Moon K.-H., Lee H. A data abstraction approach for query relaxation. *Information and Software Technology*, 42:407–418. 2000.

103 Igarashi T., Matsuoka S., Kawachiya S., Tanaka H. Interactive Beautification: A Technique for Rapid Geometric Design. *Proc. ACM UIST 1997*: 105–114. 1997.

104 Jankun-Kelly T., Kreylos O., Ma K.-L., Hamann B., Joy K.I., Shalf J., Bethel E.W. Deploying web-based visual exploration tools on the grid. *IEEE Computer Graphics and Applications*, **23**(2), 40–50, 2003.

105 Jankun-Kelly T.J., Ma K.-L., Gertz M. A Model and Framework for Visualization Exploration. *IEEE Trans. on Visualization and Comp. Graphics*, **13**(2): 357–369, 2007.

106 Johansen, R. *Groupware: Computer Support for Business Teams*. The Free Press, New York, 1988.

107 Kaasten S., Greenberg S. Integrating Back, History and Bookmarks in Web Browsers. *Extended Abstracts ACM CHI*: 379–380, 2001.

108 Kaasten S., Greenberg S., Edwards C. How People Recognize Previously Seen Web Pages from Titles, URLs, and Thumbnails. *Tech Report 2001-692-15*, Department of Computer Science, University of Calgary, Alberta, Canada, 2001.

109 Keel, PE. Collaborative Visual Analytics: Inferring from the Spatial Organization and Collaborative Use of Information. *IEEE Symposium on Visual Analytics Science and Technology:* 137–144. 2007.

110 Khan, A., Matejka, J., Fitzmaurice, G., Kurtenbach, G. Spotlight: directing users' attention on large displays. In *Proc. ACM CHI 2005*: 791–798. Apr 2005.

111 Klemmer S.R., Thomsen M., Phelps-Goodman E., Lee R., Landay J.A. Where Do Web Sites Come From? Capturing and Interacting with Design History. In *Proc. ACM CHI*: 1–8, 2002.

112 Kosslyn S.M. Understanding Charts and Graphs. *Applied Cognitive Psychology*, **3**:185–226. 1989.

113 Kurlander D., Feiner S. Editable Graphical Histories. In *Proc. IEEE Workshop on Visual Language*: 127–134, 1988.

114 Kreuseler M., Nocke T., Schumann H. A. History Mechanism for Visual Data Mining. In *Proc. IEEE InfoVis*: 49–56, 2004.

115 Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*. 2nd ed., Sage. 2004.

116 Landauer T.K. How Much do People Remember? Some Estimates of the Quantity of Learned Information in Long-Term Memory. *Cognitive Science*, **10**(4): 477–493, 1986.

117 Lasseter J. Principles of Traditional Animation applied to 3D Computer Animation. In *Proc. ACM SIGGRAPH 1987*: 35–44, July 1987.

118 Lee J.P., Grinstein G.G. An Architecture for Retaining and Analyzing Visual Explorations of Databases. In *Proc. IEEE Vis*: 101–108, 1995.

119 Lefer W., Pierson J.M. Using network of workstations to support a web-based visualization service. In *Proc. Euro-Par'99:* 624–633, 1999.

120 Ling K., Beenen G., Ludford P., Wang X., Chang K., Cosley D., Frankowski D., Terveen L., Rashid A.M., Resnick P., Kraut R. Using social psychology to motivate contributions to online communities. *Journal of Computer-Mediated Communication*, **10**(4), 2005.

121 Los Angeles Times Homicide Map, 2007. http://www.latimes.com/homicidemap/

122 Livny M., Ramakrishnan R., Beyer K., Chen G., Donjerkovic D., Lawande S., Myllymaki, J., Wenger K. DEVise: Integrated Querying and Visual Exploration of Large Datasets. In *Proc. ACM SIGMOD 1997*: 310–312. May 1997.

123 Ma K.-L. Image Graphs: A Novel Interface for Visual Data Exploration. In *Proc. IEEE Visualization*: 81–88, 1999.

124 Mackinlay J. Automating the design of graphical presentation of relational information. *ACM Transactions on Graphics* **5**(2): 110–141. 1986.

125 Mackinlay J.D., Hanrahan P., Stolte C. Show Me: Automatic Presentation for Visual Analysis. *IEEE Trans. on Visualization and Comp. Graphics*, **13**(6): 1137–1144, 2007.

126 Mankoff J., Hudson S.E., Abowd G.D. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *Proc. ACM UIST 2000*: 11–20. 2000.

127 Martin A.R., Ward M.O. High Dimensional Brushing for Interactive Exploration of Multivariate Data. *IEEE Visualization*: 271–278. 1995.

128 Meng C., Yasue M., Imamiya A., Mao X. Visualizing Histories for Selective Undo and Redo. In *Proc. 3rd Asian Pacific Computer and Human Interaction*: 459, 1998.

129 Michotte A. *The Perception of Causality* (T. Miles & E. Miles, Trans.) London: Methuen. (Original work published 1946), 1963.

130 Millen D.R., Feinberg J., Kerr, B. Dogear: Social Bookmarking in the Enterprise. In *Proc. of CHI 2006*: 111–120. 2006.

131 Mohammed S. Toward an Understanding of Cognitive Consensus in a Group Decision-Making Context. *The Journal of Applied Behavioral Science.* **37**(4): 408–425. Dec 2001.

132 Morris M.R., Horvitz E. SearchTogether: An Interface for Collaborative Web Search. In *Proc. ACM UIST 2007*: 3-12. 2007.

133 Morris M.R. A Survey of Collaborative Web Search Practices. In *Proc. ACM CHI 2008*: 1657–1660.Florence, Italy. 2008.

134 Myers B.A., Kosbie D.S. Reusable Hierarchical Command Objects. In *Proc. ACM CHI*: 260–267, 1996.

135 NASA Clickworkers. [WWW document] http://clickworkers.arc.nasa.gov/ (accessed 6 November 2008).

136 Neuberg B. AJAX: How to Handle Bookmarks and Back Buttons. http://www.onjava.com/pub/a/onjava/2005/10/26/ajax-handling-bookmarks-and-back-button.html, 2005.

137 North C., Shneiderman B. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. In *Proc. Advanced Visual Interfaces 2000*:128–135. May 2000.

138 Olston C., Stonebraker M., Aiken A., Hellerstein J.M. VIQING: Visual Interactive Querying. In *Proc. IEEE Visual Languages 1998*:162–169. Sep 1998.

139 Olston C., Chi E.H. ScentTrails: Integrating browsing and searching on the Web. *ACM TOCHI*, **10**(3): 177–197. Sep. 2003.

140 Palmer S. *Vision Science: Photons to Phenomenology*. MIT Press. 1999.

141 Pirolli P. Social Information Foraging. Chapter 8, *Information Foraging: Adaptive Interaction with Information*. Oxford University Press. 2007.

142 Pirolli P., Card S.K. Information Foraging. *Psychological Review* 1999; **106**(4): 643–675.

143 Pirolli P., Card S.K. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of International Conference on Intelligence Analysis* 2005.

144 Plaisant C., Rose A., Rubloff G., Salter R., Shneiderman B. The Design of History Mechanisms and their Use in Collaborative Educational Simulations. In *Proc. CSCL*: 348–359, 1999.

145 Plaisant C., Grosjean J., Bederson B.B. SpaceTree: Supporting Exploration in a Large Node-Link Tree, Design Evolution and Empirical Evaluation. In *Proc. IEEE InfoVis 2002*: 57–64, Oct. 2002.

146 Qu Y., Furnas G.W. Sources of Structure in Sensemaking. In *Extended Abstracts of ACM CHI 2005*. 2005.

147 Resnick P., Zeckhauser R., Swanson J., Lockwood K. The Value of Reputation on eBay: A Controlled Experiment. *Experimental Economics* 2006; **9**(2): 79–101.

148 Rhyne, T.-M. Scientific Visualization and Technology Transfer. *IEEE Computer*, **28**(7):94–95 , 1995.

149 Robertson G.G., Card S.K., Mackinlay J.D. The Cognitive Coprocessor Architecture for Interactive User Interfaces. In *Proc. ACM UIST 1989*: 10–18, Williamsburg, VA, Nov 1989.

150 Robertson G.G., Card S.K., Mackinlay, J.D. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proc. ACM CHI 1991*: 189–194, New Orleans, LA, Apr 1991.

151 Robertson G.G., Cameron K., Czerwinski M., Robbins D. Animated Visualization of Multiple Intersecting Hierarchies. *Journal of Information Visualization*, **1**(1):50–65. Palgrave, 2002.

152 Robertson G.G., Fernandez R., Fisher D., Lee B., Stasko J. Effectiveness of Animation in Trend Visualization. *IEEE Trans. on Visualization and Comp. Graphics*, **14**(6):1325–1332. Nov/Dec 2008.

153 Robinson A.C., Weaver C. Re-Visualization: Interactive Visualization of the Process of Visual Analysis. *Proc. GIScience Workshop on Visual Analytics & Spatial Decision Support*, 2006.

154 Russell D.M., Stefik M.J., Pirolli P., Card S.K. The Cost Structure of Sensemaking, In *Proc. ACM CHI 1993*. 1993.

155 Salen K., Zimmerman E. *Rules of Play: Fundamentals of Game Design*. MIT Press; 2003.

156 Saund E., Fleet D., Larner D., Mahoney J. Perceptually-supported image editing of text and graphics. In *Proc. ACM UIST 2003*: 183–192. 2003.

157 Seo J., Shneiderman B. A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization*, **4**(2):99–113. 2005.

158 Scheff T.J. Toward a Sociological Model of Consensus. *American Sociological Review*, **32**(1): 32–46. 1967.

159 Schultz-Hart S., Frey D., Lüthgens C., Moscovici S. Biased Information Search in Group Decision Making. *Journal of Personality and Social Psychology*, **78**(4): 655–669. 2000.

160 Senay H., Ignatius E. Rules and principles of scientific data visualization. Tech. Report GWU-IIST-90-13, The George Washington University. 1990.

161 Shipman F.M., Hsieh H. Navigable History: A Reader's View of Writer's Time. *The New Review of Hypermedia and Multimedia*. **6**(1): 147–167, 2000.

162 Shneiderman, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proc. IEEE Visual Languages*: 336–343, 1996

163 Spence I., Lewandowsky S. Displaying proportions and percentages. *Applied Cognitive Psychology*, **5**:61–77, 1991.

164 Spotfire Decision Site Posters. [WWW document] http://spotfire.com/products/decisionsite_posters.cfm (accessed 28 November 2007).

165 Sproull L., Kiesler S. Computers, Networks, and Work. *Scientific American,* **265**(3): 84–91. Sep 1991.

166 Stasser G., Titus W. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology,* **57**: 67–78. 1985.

167 Stolte C., Tang D., Hanrahan P. Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):52–65. Jan 2002.

168 Surowiecki J. *The Wisdom of Crowds*. Random House; 2004.

169 Swivel. [WWW document] http://www.swivel.com (accessed 28 November 2007).

170 Tableau Server. [WWW document] http://www.tableausoftware.com/products/server (accessed 6 November 2008).

171 Thomas J.J., Cook K.A. (Eds). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press; 2005.

172 Tufte E. *The Visual Display of Quantitative Information*. Graphics Press. 1983.

173 Tufte, E. *Envisioning Information*. Graphics Press. 1990.

174 Tufte E. *Beautiful Evidence*. Graphics Press. 2006.

175 Tukey J. *Exploratory Data Analysis*. Addison-Wesley. 1977.

176 Tversky B., Morrison J., Betrancourt M. Animation: Can It Facilitate? *Int. J. Human-Computer Studies*, **57**:247–262, 2002.

177 Twitter #Vote Report. [Web document] http://blog.twittervotereport.com/ (accessed 6 Nov 2008).

178 Urban Sensing Project, CENS/UCLA. [Web document] http://urban.cens.ucla.edu/ (accessed 6 Nov 2008).

179 Viégas F.B., boyd d., Nguyen D.H., Potter J., Donath J.S. Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. In *Proc. of the 37th Hawaii International Conference on System Sciences*. 2004.

180 Viégas F.B., Wattenberg M. Communication-Minded Visualization: A Call to Action. *IBM Systems Journal*, **45**(4). 2006.

181 Viégas F.B., Wattenberg M., van Ham F., Kriss J., McKeon M. Many Eyes: A Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics (InfoVis'07),* **13**(6): 1121–1128. 2007.

182 Vitter J.S. US&R: A New Framework for Redoing. *IEEE Software*, **1**(4): 39–52, 1984.

183 von Ahn L. Games with a Purpose. *IEEE Computer* 2006; **39**(6): 92–94.

184 Walther J.B. Computer-mediated communication: Impersonal, interpersonal and hyperpersonal interaction. *Communication Research* 1996; **23**(1): 3–43.

185 Waterson S., Hong J.I., Sohn T., Heer J., Matthews T., Landay J.A. What Did They Do? Understanding Clickstreams with the WebQuilt Visualization System. In *Proc. AVI 2002*: 94–102, 2002.

186 Wattenberg M., Kriss J. Designing for Social Data Analysis. *IEEE Transactions on Visualization and Computer Graphics* 2006; **12**(4):549–557.

187 Ware C. *Information Visualization: Perception for Design*. 2nd ed., Morgan Kaufmann. 2004.

188 Weaver C. Building Highly-Coordinated Visualizations In Improvise. In *Proc. IEEE InfoVis 2004*:159–156. 2004.

189 Wexelblat A., Maes P. Footprints: History-Rich Tools for Information Foraging. In *Proc. ACM CHI 1999*: 270–277. 1999.

190 Wikimapia. [WWW document] http://wikimapia.org (accessed 28 November 2007).

191 Willett W., Heer J., Agrawala M. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics,* **13**(6): 1129–1136. Nov/Dec 2007.

192 Williamson C., Shneiderman B. The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system In *Proc. ACM SIGIR 1992*: 338-346. Copenhagen, Denmark. Jun. 1992.

193 Woodruff A., Faulring A., Rosenholtz R., Morrsion J., Pirolli P. Using Thumbnails to Search the Web. In *Proc. ACM CHI*: 198–205, 2001

194 World Bank World Development Indicators, 2007. http://devdata.worldbank.org/data-query/

195 Wright W., Schroh D., Proulx P., Skaburskis A., Cort B. The sandbox for analysis: concepts and evaluation. In *Proc. ACM CHI 2006*. 2006.

196 Wu F., Huberman B.A., Adamic L.A., Tyler J.R. Information flow in social groups. *Physica A: Statistical and Theoretical Physics*, **337**(1–2): 327. 2000.

197 Yang D., Rundensteiner E.A., Ward M.O. Analysis Guided Visual Exploration to Multivariate Data. In *Proc. IEEE Symposium on Visual Analytics Science and Technology* 2007.

198 Yantis S. Multielement Visual Tracking: Attention and Perceptual Organization. *Cognitive Psychology*, **24**(3):295–340. July 1992.

199 Yee K.-P., Fisher D., Dhamija R., Hearst M. Animated Exploration of Graphs with Radial Layout. In *Proc. IEEE InfoVis 2001*: 43–50, 2001.

200 Zellweger P.T., Mackinlay J.D., Good L., Stefik M., Baudisch P.. City lights: contextual views in minimal space. In *Extended Abstracts of ACM CHI 2003*: 838–839. Fort Lauderdale, FL. Apr. 2003.

201 Zhang J., Norman D.A. Representations in Distributed Cognitive Tasks. *Cognitive Science:* **18**(1): 87–122. 1994.

202 Zongker D., Salesin D. On Creating Animated Presentations. In *Proc. Eurographics/SIGGRAPH Symp. on Comp. Animation*: 298–308, 2003