Visual Exploratory Analysis of Large Data Sets

Evaluation and Application

by

Heidi Lap Mun Lam

B.A.Sc., Simon Fraser University, 2001
M.A.Sc., Simon Fraser University, 2004

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate Studies

(Computer Science)

The University Of British Columbia

(Vancouver)

May, 2008

Abstract

Large data sets are difficult to analyze. Visualization has been proposed to assist exploratory data analysis (EDA) as our visual systems can process signals in parallel to quickly detect patterns. Nonetheless, designing an effective visual analytic tool remains a challenge.

This challenge is partly due to our incomplete understanding of how common visualization techniques are used by human operators during analyses, either in laboratory settings or in the workplace.

This thesis aims to further understand how visualizations can be used to support EDA. More specifically, we studied techniques that display multiple levels of visual information resolutions (VIRs) for analyses using a range of methods.

The first study is a summary synthesis conducted to obtain a snapshot of knowledge in multiple-VIR use and to identify research questions for the thesis: (1) low-VIR use and creation; (2) spatial arrangements of VIRs. The next two studies are laboratory studies to investigate the visual memory cost of image transformations frequently used to create low-VIR displays and overview use with single-level data displayed in multiple-VIR interfaces.

For a more well-rounded evaluation, we needed to study these techniques in ecologically-valid settings. We therefore selected the application domain of web session log analysis and applied our knowledge from our first three evaluations to build a tool called Session Viewer. Taking the multiple coordinated view and overview + detail approaches, Session Viewer displays multiple levels of web session log data and multiple views of session populations to facilitate data analysis from the high-level statistical to the low-level detailed session analysis approaches.

Our fourth and last study for this thesis is a field evaluation conducted at Google Inc. with seven session analysts using Session Viewer to analyze their own data with their own tasks. Study observations suggested that displaying web session logs at multiple levels using the overview + detail technique helped bridge between high-level statistical and low-level detailed session analyses, and the simultaneous display of multiple session populations at all data levels using multiple views allowed quick comparisons between session populations. We also identified design and deployment considerations to meet the needs of diverse data sources and analysis styles.

Table of Contents

Appendices

List of Tables

List of Figures

Acknowledgements

In writing this thesis, I struggled between the use of “We” and “I” to describe a body of work that constitutes the thesis. To me, frequent use of the pronoun “I” seems unnecessarily egotistic since I cannot claim the work to be mine alone. Theoretically, I might have substituted instances of “I” by “We” “to secure an impersonal style and tone, or to avoid the obtrusive repetition of “I” (Oxford English Dictionary 2007), or to hide behind those “We”s to absolve myself from taking responsibility for the work.

However, the true meaning behind those “We”s is to emphasize, acknowledge, and express my gratitude to my collaborators for their guidance and inspirations in these past four years: my thesis supervisor, Tamara Munzner; my thesis committee members, Diane Tang, and Joanna McGrenere; and my collaborators, most were also my internship mentors, Patrick Baudisch, Ronald Rensink, Robert Kincaid, and Daniel Russell.

Naturally, I would also like to thank my parents, who for many years, have tolerated my many idiosyncrasies and assumed them to be necessary evils of technology-related scholarship. I understandably have not been diligent in providing them with counter-examples.

Chapter 1

Introduction

A study in 2002 found that human beings collectively produced more than five exabytes, or 5×10^{18} bytes, worth of recorded information per year in the form of print, film, magnetic and optical storage media (Lyman and Varian 2003). The same study estimated that new stored information had grown by about 30% per year between 1999 and 2002 (Lyman and Varian 2003). Having such massive amounts of data is a double-edged sword. Ideally, availability of data makes it possible to make decisions based on data rather than intuition. For example, it has been argued that data analysts can obtain better insights into human behaviours from available databases to support decision making in both corporations and governments, and in a wide range of areas such as marketing, economics, and social policies (Levitt and Dubner 2005; Thomas and Cook 2005; Ayres 2007). In practice, however, human beings may be overwhelmed by the massiveness of data sets. Even as early as 1960, the flood of available data to individuals made information overload a subject of interest in clinical psychiatry (Miller 1960). By the mid 1990s, the term "information revolution" had became part of our vocabulary, and commentators on the social impacts of technology described challenges an individual had to face in our information age (e.g., Shenk 1997). If the burden on individual citizens to make informed decisions given the amount of data is heavy, the challenges on data analysts demand more than human diligence. The reason being that analysts have to routinely handle and analyze massive amounts of raw and potentially conflicting data gathered from various sources in multiple formats such as text, numbers, sounds, and images.

In 2008, a decade later, data overload is still considered to be a generic and difficult problem for the data analyst (Woods et al. 2002; Thomas and Cook 2005). Given our visual ability to process a large amount of information in parallel, visualization has often been considered as a vital component in the solution (Tukey 1977; Thomas and Cook 2005). According to Tukey (1977), "the greatest value of a picture is when it forces us to notice what we never expected to see" (p. vi).

Indeed, visualization has been applied to support a wide range of analyses in different domains. Examples include text documents with the ThemeView 3D visual landscape in In-SPIRE (Hetzler and Turner 2004) and ThemeRiver (Havre et al. 2002), computer source code with Seesoft (Eick et al. 1992), e-mails with Themail (Viégas et al. 2005), calendar data with DateLens (Bederson et al. 2004), tree analysis with Treemaps (Johnson and Shneiderman 1991) and TreeJuxtaposer (Munzner et al. 2003), and tabular database analysis with Table Lens (Rao and Card 1994) and Polaris (Stolte and Hanrahan 2000). Visualization systems have also been built to assist data exploration and analysis. One example is the GeoTime software by Oculus (Hetzler and Turner 2004).

In terms of visualization techniques, one main class of techniques display data in multiple visual information resolutions (VIRs). Examples of multiple-VIR techniques include zooming, focus + context, and overview + detail. In the list of visualization systems given above, panning and zooming techniques were applied to the ThemeView 3D visual landscape for users to access thousands of documents clustered by themes (Hetzler and Turner 2004). Overview + detail techniques were used in Themail (Viégas et al. 2005) and Seesoft (Eick et al. 1992). Seesoft displays up to 50,000 lines of source code with each line mapped to a thin row in the overview. Users can access the actual source code in a separate window. Themail displays gigabytes of e-mails by extracting keywords and displaying them in columns. Users can select words to view corresponding e-mail messages in detail. Focus + context applications in our example list include Table Lens to display up to 68,400 table cells on a 19-inch screen (Rao and Card 1994), DateLens to display up to six months of calendar data on small screens (Bederson et al. 2004), and TreeJuxtaposer to display up to 500,000 tree nodes (Munzner et al. 2003).

Despite continual efforts, displaying large data sets to support exploratory analysis remains difficult. Part of the challenge is scalability (Thomas and Cook 2005, p. 24–28). The sheer size of modern data sets requires novel visualization techniques that can display more data points and dimensions than available pixels on standard output devices. Displaying large data sets also requires a non-trivial amount of engineering effort to ensure system interactivity for data exploration. In addition to technical challenges, visualization system designers also need to consider perceptual and cognitive limitations of human operators.

Due to the complex interplay between the human operator and the visualization system, empirical evaluation of system use is paramount in providing effective visualization support for data analysis. However, our understanding of frequently-used visualization techniques such as animation (Tversky et al. 2002) and focus + context (Furnas 2006) remains incomplete. Another important aspect of evaluation that tends to be overlooked in visualization research concerns system deployment (Thomas and Cook 2005, p. 149). Deployment-related considerations involve handling real-world data and diverse analytical practices, and integrating the visualization system with established workflows and tools. Perhaps because of the lack of follow-through of research prototypes into the workplace, technology transfer and commercialization of prototypes remain rare despite the proliferation of visualization research. In short, building an effective visualization system to support visual exploratory analysis of large data sets requires understanding of both information-visualization specific issues and non-visualization specific deployment issues.

This thesis therefore investigated both information-visualization specific and general deployment considerations in system building, and crystalized findings as design guidelines and considerations. To scope the project, we focused on multiple-VIR visualization techniques. As no single research strategy can adequately answer a research question, we investigated different aspects of multiple-VIR techniques with four studies conducted using a wide variety of methods, including a qualitative summary synthesis, a laboratory experiment, an experimental-simulation study, and a field evaluation.

The thesis started by taking a high-level view to obtain a more comprehensive snapshot of knowledge about multiple-VIR interface use using a qualitative summary synthesis approach, reported in Chapter 4 as the first of four evaluations. We examined 19 existing experimental-simulation studies to extract high-level guidelines for multiple-VIR interface designs, looking at issues such as the amount of information displayed on overviews, the number of VIRs required for effective analysis, and methods to present the various VIRs in the interface. We also identified two research areas for further examination in this thesis: (1) overview creation in multiple-VIR interfaces, and (2) spatial arrangements of the different VIRs.

We investigated these two questions in our quantitative laboratory studies. The second study of the thesis is a laboratory experiment. Reported in Chapter 5, our laboratory experiment systematically measured visual memory costs incurred by two-dimensional geometric transformations that are frequently used in creating data overviews and integrating VIRs in multiple-VIR interfaces.

The third study of the thesis is an experimental-simulation study detailed in Chapter 6. We examined the use of overviews in multiple-VIR interfaces for single-level data.

While laboratory studies can be effective in studying visualization techniques, such approaches are necessarily limited by the need to study isolated factors such as task and visualization components using predetermined dependent variables (Plaisant 2004). To truly understand how a visualization technique is used and to understand deployment issues, we need to consider the technique as part of a visualization system in ecologically-valid settings.

The remaining efforts in this thesis were therefore devoted to addressing these questions using a specific application domain. We chose web session log analysis as our application domain since it is representative and relevant: challenges faced by web session analysts are also found in other analytical scenarios involving large data sets with complex compositions, and the need to analyze web log data to understand search behaviour increases with the growing importance of the Internet as an information source. Existing tools for web session log analysis are largely non-visual and do not adequately support data exploration. This application domain is therefore an opportunity for us to apply the knowledge and experience gained in the first three studies, such as guidelines for overview construction and VIR presentation, to better support analysis of large data sets by mitigating problems in existing analysis practices. We built Session Viewer and deployed it at Google Inc. The design and implementation of Session Viewer are detailed in Chapter 7. Chapter 8 reports the fourth, and last, study in this thesis, which is a field evaluation conducted to evaluate design choices made in the process and to further understand issues encountered by visualization systems in the workplace.

Chapter 9 discusses four questions addressed in our studies that remain research challenges. The chapter looks at the two research questions in the thesis: design choices in overview creation and design choices in spatial arrangements of VIRs in interfaces. We also discuss a question related to overview creation—potential roles of context in data analysis. The last open question addressed in Chapter 9 concerns approaches to evaluating visualization techniques and systems to sum up our experiences in the four evaluations conducted in this thesis. The chapter concludes the thesis by elaborating on contributions of the thesis and suggests future directions.

The remainder of this introductory chapter is structured as follows. After introducing two sets of terminology used throughout the thesis in Section 1.1, Sections 1.2 and 1.3 list the contributions of the thesis. Appendix A contains a list of published, submitted, and in-preparation papers related to the thesis.

1.1 Terminology

Two sets of terminologies are used throughout the thesis. The first set concerns the interface techniques under study (Section 1.1.1), and the second set concerns the strategies employed to study those techniques (Section 1.1.2).

1.1.1 Visual information resolution (VIR)

In this thesis, we devised the term **visual information resolution** (VIR) as a measure of visual information perceivability. Visual information is defined as datum values made accessible to the visualization system's users by showing them visually.

By our definition, displays with low VIR have comparatively lower visual information perceivability than displays with high VIR. Perceivability can be further characterized based on type, visual quantity, and visual quality of the displayed data. In terms of information type, VIR of interface views can differ if they display data from different levels of the hierarchy in the data organization: lower VIR shows data at higher levels of the hierarchy. For example, in Treemaps (Bederson et al. 2002), users can focus on different layers of the hierarchical tree at different VIRs in the display. In terms of quantity, interface views can differ in the amount of information displayed. One example is semantic zooming, where users are provided with different amounts of details in a view by zooming in and out. Both our metrics of information type and visual quantity are akin to Simon and Larkin's (1987) definition of informational equivalence of representations, where "two representations are informational equivalent if all of the information in the one is also inferable from the other, and vice versa" (p. 67).

In terms of quality, visual objects can display the same amount of data points with different visual encodings that result in different perceivability. One common example is the display of textual data. With the same font type, data displayed using small unreadable font sizes is considered to be of lower VIR than those displayed in larger readable font sizes. As for visual objects, the criteria of perceivability is less well defined. One example is the visual encodings used in our overview-use study in Chapter 6 where we encoded the same line graph data using two different types of encodings. The high-VIR encoding displays the y-dimension of line graphs using both space and colour, while the low-VIR encoding only uses colour, thus making the fine details of the displayed lines graph less perceivable. This characterization is akin to Simon and Larkin's (1987) definition of computational equivalence of representations, where "two representations are computationally equivalent if they are informationally equivalent and, in addition, any inference that can be drawn easily and quickly from the information given explicitly in the one can also be drawn easily and quickly from the information given explicitly in the other, and vice versa" (p. 67).

Taxonomies of multiple-VIR techniques exist. One example is Plaisant et al.'s (1995) taxonomy for image browsers. Detailed mapping between existing taxonomies to our terminologies is beyond the scope of our discussion here. In general, our terminologies differ by focusing on the visual encodings instead of on their expected functions: for example, focus (as in focus + context) or detail (as in overview + detail) can be thought of as **high VIR**, while context or overview is of comparatively **low VIR**.

Multiple-VIR interfaces can be further classified as **temporal** or **simultaneous** based on the way they display the multiple VIRs, as shown in Figure 1.1. **Temporal** interfaces, an example being the pan-and-zoom user interfaces, allow users to drill up and down the zoom hierarchy and display the different VIRs one at a time. In contrast, **simultaneous** interfaces show all the VIRs on the same display. We refer to interfaces that integrate and spatially embed the different VIRs as **embedded** displays, as in focus + context visualizations. When the different VIRs are displayed as separate views, we refer to these interfaces as **separate**, as in overview + detail displays. Since the different VIRs can occupy the entire display window, or be integrated as part of a single window, we explicitly differentiate the two by using the term **view** to denote separate windows or panes, and the term **region** to denote an area within a view.

Additional examples of multiple-VIR interfaces can be found at `http://www.cs.ubc.ca/~hllam/res_ss_interfaces.htm`.

1.1.2 Evaluation strategies in information visualization

A wide range of experimental designs have been applied to study information visualization techniques and systems. This thesis employed four different types of strategies to examine various aspect of multiple-VIR interface use. This section introduces terminologies used in reference to visualization evaluation strategies, explains experimental design, tasks, and measurements, and how the

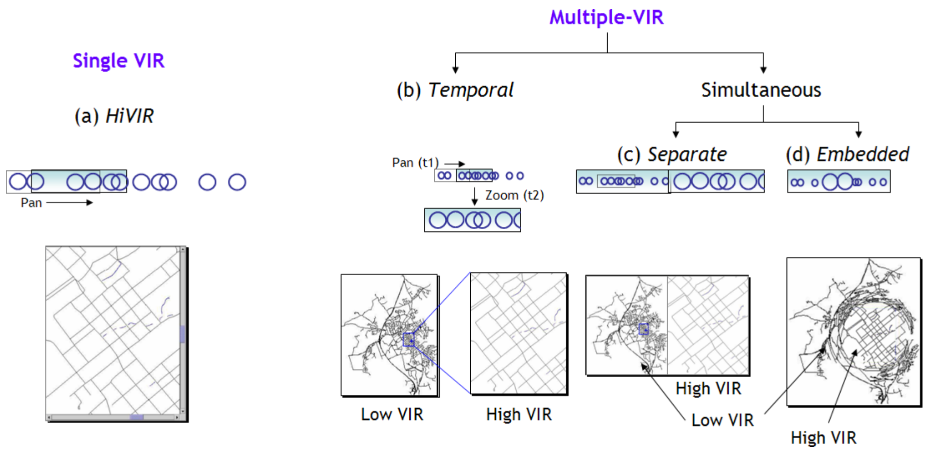


Figure 1.1: Our classification of interfaces. Interfaces can be classified based on the number of visual information resolutions (VIR). (a) Single-VIR interfaces display one VIR, as in panning interfaces. (b)–(d) Multiple-VIR interfaces show multiple VIRs. In this illustrate, each interface contains two VIRs: high (denoted as large circles) and low (denoted as small circles). (b) In the *Temporal* approach, users can pan around in the low-VIR view and zoom into an subarea as a high-VIR view, as in pan and zoom interfaces. (c) In the *Separate* approach, the low- and the high-VIRs can be placed in separate panels, as in overview + detail interfaces. (d) In the *Embedded* approach, the low- and the high-VIRs are embedded in a single unified display, as in focus + context interfaces.

four evaluations in this thesis relate to these terminologies and strategies.

There are a number of ways to classify experimental approaches. Some possibilities include the type of analysis performed on collected data (quantitative versus qualitative), the time span (short term versus longitudinal), the study environment (laboratory versus field), and the study designs (controlled experiments versus observational studies). Unfortunately, no one axis is adequate in classifying existing approaches. To facilitate discussion in this thesis, we adopt McGrath's (1994) taxonomy of research strategies, originally developed for social and behavioural sciences, as it covers a wide range of strategies and focuses on research goals.

McGrath (1994) classified research strategies with five axes into eight strategies, each represented as a slice in the strategy circumplex shown in Figure 1.2. The first three axes are based on three desirable features or criteria researchers wish to maximize in an experiment: A. Generalizability of evidence of different study populations; B. Precision of measurements of behaviours being studied;

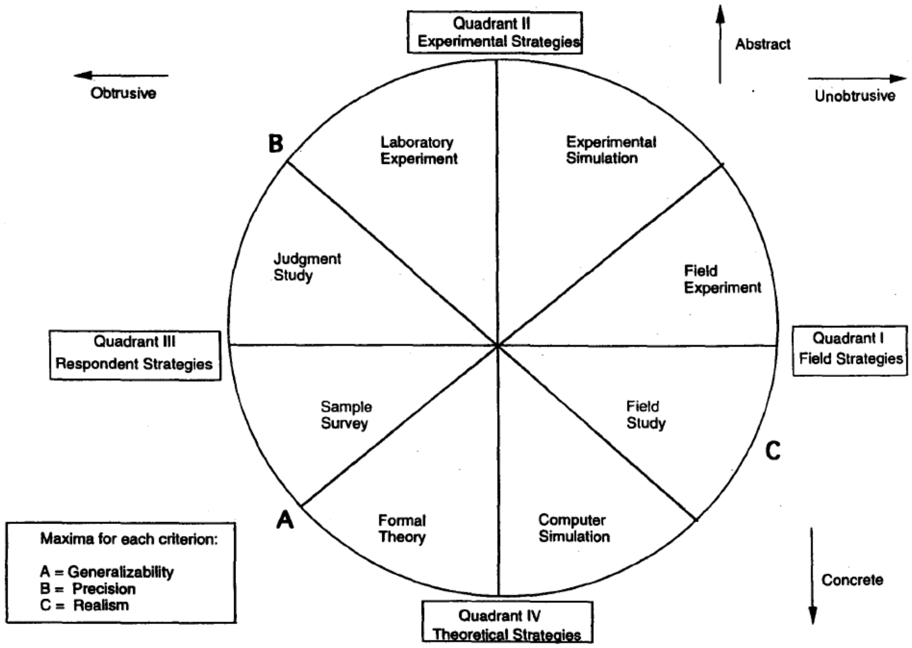


Figure 1.2: McGrath's research strategy circumplex. ©1994 by Morgan Kaufmann. Reprint by permission.

and C. Realism of the situation or context within which the measurements are being made in relation to where study results will be applied. The fourth axis is concrete-abstract, or "the degree to which the setting used in the strategy is universal or abstract vs. particular or concrete" (McGrath 1994, p. 156). The last axis is obtrusive-unobtrusive, or "the degree to which the strategy involves procedures that are obtrusive, vs. procedures that are unobtrusive, with respect to the ongoing human systems that are to be the object of the study" (McGrath 1994, p. 156).

With these five axes, McGrath (1994) derived eight research strategies grouped into four quadrants: field, experimental, respondent, and theoretical strategies. In this section, we discuss strategies employed in this thesis in the order of their presentation: the formal theory strategy of the theoretical strategies; the experimental strategies (laboratory experiment, experimental simulation); and the field strategies (field experiment, field study). We therefore will not discuss the respondent strategies in this discussion even though they have been used in surveys and questionnaires in visualization evaluations.

Quadrant IV: Theoretical Strategies

Of the two strategies in theoretical strategies included in McGrath's (1994) research strategy circumplex, the formal theory strategy include a class of evaluation method called systematic review. Though infrequently conducted, systematic reviews can provide snapshots of existing knowledge based on existing study results, where "the researcher focuses on formulating general relations among a number of variables of interest" that "hold over some relatively broad range of populations" (McGrath 1994, p. 158). The following description of the various types of systematic review is based on Chapter 13 of (Shadish et al. 2002).

Narrative review is a qualitative approach and describes existing literature using narrative descriptions without performing quantitative synthesis of study results. Most published papers have related work sections that can be considered as narrative reviews. Due to its descriptive and qualitative nature, narrative reviews can include study results gathered using dramatically different methods and therefore can potentially present a more realistic view of existing knowledge. Reflections made by the reviewers, especially when they are experienced in the area under review, can be very illuminating and thought provoking. Two excellent examples are Card et al.'s (1999) essays on various information visualization topics such as interaction and focus + context, by Tversky et al.'s (2002) review on effectiveness of animation, and Furnas's (2006) follow-up work on focus + context visualization techniques after his original paper on the topic ten years prior (Furnas 1986).

Despite its strength, narrative review can lead to incorrect conclusions as the readers must rely on the reviewer to weight the significance of each reviewed publication. In addition, the selection of reviewed publications can be biased as well. Quantitative approaches were therefore developed to mitigate these biases, which form the basis of **meta-analysis**.

Roughly, there are two main approaches to a meta-analysis: (1) vote counting and (2) study-effect analysis. Instead of describing study results, the **vote-counting approach** categorizes results as significantly positive, significant negative, or nonsignificant in reference to the research question. The category with the most entries is considered to best represent existing knowledge about the research question under analysis. Its simplicity confuses treatment effect and sample-size effects, as nonsignificant results may be due to lack of experimental power, rather than lack of effect. In other words, the vote-counting approach to meta-analysis can miscount studies with small effects, which can lead to incorrect conclusions such as missing side-effects of medications.

Meta-analysis that computes **effect size** is perhaps the most popular approach to meta-analysis in information visualization evaluation. Indeed, as discussed in Section 3.1.4, to the best of our knowledge, the only meta-analysis in information visualization computed effect size (Chen and Yu 2000). Effect size measure is used as a common metric for the diverse study outcomes so that the results can be meaningfully compared in a meta-analysis. However, using a standard metric usually leads to a drastic reduction of studies that can be brought under meta-analysis. In the case of Chen and Yu (2000), only 6 out of the 35 studies considered for the review met the researchers' criteria for inclusion in the meta-analysis.

Chapter 4 is a systematic review that took a mixture of the two approaches to obtain a snapshot of existing knowledge on multiple-VIR interface study results. Since our approach is not a conventional one, we termed it **summary synthesis** to avoid confusion. Due to the small number of experimental-simulation studies concerning multiple-VIR interfaces, we combined the inclusiveness and flexibility of narrative review with some of the rigor of meta-analysis by listing all applicable study results for each research question under consideration, instead of only reporting results that supported our conclusions, as in most narrative reviews. Section 4.1 further details the methodology.

Quadrant II: Experimental strategies

The two strategies included in the experimental strategy quadrant are laboratory experiment and experimental simulation.

Laboratory experiment is a strategy where the researcher "deliberately concocts a situation or behaviour setting or context, defines the rules for its operation, and then induces some individuals or groups to enter the concocted system and engage in the behaviours called for by its rules and circumstances" (McGrath 1994, p. 157). Typically, designs in laboratory experiments in information visualization are modeled after experiments in sciences, especially in experimental psychology, where "purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for changes that may be observed in the output response" (Montgomery 2001, p. 1). Examples include perceptual and cognitive studies to understand human limitations when interacting with visual displays. Section 3.1.1 discusses literature in this area.

In the context of information visualization, "input variables of a process or system" relate to properties of the visual display such as item density and variations in colour encoding or texture of display items. Stimuli used in laboratory experiments are mostly static images. Input variables are generally referred to as independent variables or experimental factors. The design is usually such that effects of these factors, such as reaction time and task accuracy, can be analyzed using established statistical methods such as t-tests and analysis of variance (ANOVA) F-tests.

To isolate study factors, tasks in laboratory experiments are generally simple abstracted tasks such as visual search tasks and visual memory tasks. The visual search paradigm is an experimental technique developed by experimental psychologists to study a number of visual processes, such as preattentive and attentive processes in vision (Wolfe 2000, p. 335). In this paradigm, participants are shown visual displays containing varying numbers of objects and are asked to determine whether a pre-specified target is included in the display. For example, a person might be asked to look for a red T in a display containing different numbers of blue T's and red O's, and, on trials where the target is present, a red T as well (e.g., in Treisman and Gelade 1980).

Another popular task used in laboratory experiments is the study of explicit memory using a three-phased task: a studying or encoding phase where the participant is exposed to the stimuli; a retention phase during which the stimuli is held in memory; and a testing phase where the participant is asked to recognize or recall information presented in the study phase (Wixted 1998, p. 265). Chapter 5 reports a study in this thesis that adopted the visual memory task to measure visual memory costs incurred by image transformations in interfaces.

Experimental simulation is a strategy where the researcher "attempts to achieve much of the precision and control of the laboratory experiment but to gain some of the realism (or apparent realism) of field studies" by "concocting a situation or behaviour setting or context" and "making it as much like some class of actual behaviour setting as possible" (McGrath 1994, p. 157). Over 50% of evaluations published in the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI) proceedings included formal evaluations employing this strategy (Barkhuus and Rode 2007). Like laboratory experiments, experimental-simulation studies mostly use factorial design, measure task completion time and task accuracy analyzed using established statistical methods. The major differences reside in bringing realism to the study setting, which include testing interactive interfaces displaying realistic data instead of static abstract stimuli, studying scenario-based tasks instead of simple abstracted tasks, and soliciting subjective participant feedback in addition to objective time and accuracy measurements. Tasks are chosen based on the intended application of the visualization technique or system under investigation, and usually comprise basic task operations such as locate, identify, compare, associate, distinguish, rank, cluster, correlate, and categorize (Amar et al. 2005; Roth and Mattis 1990; Tory and Möller 2004; Wehrend and Lewis 1990). To obtain feedback from participants regarding study interfaces, experimental-simulation studies typically gather subjective feedback in the form of questionnaires, which can be reported quantitatively such as in the NASA-TLX scale that measures mental workload (Zhang 2005), and qualitatively such as open-ended questions that solicit perceived positive and negative aspects of the interfaces.

The third study in this thesis took the experimental simulation strategy to study overview use in multiple-VIR interface to display single-level data. Section 3.1.2 surveys experimental-simulation studies that focused on multiple-VIR interface techniques and systems. Chapter 4 systematically reviews existing experimental-simulation studies in this area to derive design guidelines.

Quadrant I: Field strategies

Two strategies were included in the Quadrant I field strategies (Figure 1.2): field experiment and field study (McGrath 1994).

Both **field studies** and **field experiments** are strategies where the researcher studies a natural behavioural system. In field studies, the research "sets out to make direct observations of 'natural', ongoing systems, while intruding on and disturbing those systems as little as possible" (McGrath 1994, p. 157). This strategy is frequently employed during visualization designs to gather design requirements by understanding user characteristics such as workflow and expertise, and existing practices and problems of the target application. In this thesis, the task of interest is exploratory data analysis, discussed in Section 2.1, and the application domain is web session log analysis, discussed in Section 2.4.2. Section 7.1 reports interview findings of existing analysis practices and problems identified before system design. Section 8.1.1 reports similar findings from interviews conducted as part of our field work detailed in Chapter 8.

Field experiments, in contrast, give up some of the unobtrusiveness by modifying one aspect of the system, as the goals of these studies are frequently to assess the causal effects of the difference in that manipulated feature on other behaviours of the system. In the context of visualization evaluation, the modification often comes in the form of a visualization system, usually designed to address existing analysis concerns. The study therefore aims to evaluate the system by observing changes in task behaviours. To preserve the naturalness and unobtrusiveness of the method, researchers prefer having the participants use their own data performing participants' own tasks instead of prescribed tasks whenever possible. Data collected are qualitative and observational. Section 3.1.3 surveys field experiments performed to evaluate information systems.

Chapter 8 reports the field work conducted at Google Inc. to study the use of Session Viewer, a visual analytic tool built as part of this thesis to support web session log analysis. Our work is akin to field experiments as we introduced the tool at a workplace. However, we focused more on tool use than on behavioural changes brought about by the tool, so use of the term "experiment" could lead to misunderstanding. We labeled our work **field evaluation** instead.

1.2 Thesis Contributions: Evaluation

The evaluation aspect of this thesis involves four studies to investigate challenges in building visualization systems to support data analysis, each with a different research strategy based on McGrath's (1994) research circumplex, as shown in Figure 1.2. We started with a summary synthesis to systematically review existing multiple-VIR study results. From our findings, we identified two research areas for further investigations using a laboratory experiment, an experimental-simulation study, and a field evaluation.

Contributions from each study are described as follows:

1. Given the lack of understanding of multiple-VIR interface use and effectiveness in both the information visualization and human-computer interaction communities, we analyzed 19 existing multiple-VIR interface studies to extract high-level interface design guidelines. Chapter 4 details our findings.

2. We systematically examined the effects of two-dimensional geometric transformations and background grids on visual memory and defined a no-cost zone for each transformation type within which we did not detect performance degradations. We verified and refined two established design

guidelines in this context: we refined guidelines on preserving orthogonal ordering, and verified the effectiveness of background grids (Misue et al. 1995). Chapter 5 details the study.

3. In interfaces that provide multiple VIRs, low-VIR overviews typically sacrifice visual details for display capacity, with the assumption that users can select regions of interest to examine at higher VIRs. We examined and refuted this assumption for single-level data and proposed interaction costs as a factor. Chapter 6 details the study.

4. There have been very few long term and detailed evaluations of information visualization systems in the workplace using real-world data. We evaluated Session Viewer with seven web session log analysts at Google Inc. in a field evaluation and identified two design themes summarizing issues and implications for visualization system effectiveness in the workplace. Chapter 8 details the study.

The remainder of this section elaborates on each conducted evaluation in terms of study motivations and goals, study approach, and major findings.

1.2.1 Summary synthesis: multiple visual information resolution interface designs

Motivations and goals

Despite numerous evaluation efforts and the long history of applying multiple-VIR techniques to interface design, the use and effectiveness of these techniques remain unclear (Furnas 2006). The difficulty in studying these interfaces reflects their complexity; a large number of factors are at play that significantly affect their use. These factors include the match between task information requirements and the type and amount of information displayed, the supported interactions, the use of image transformations in the implementations, and user characteristics in terms of spatial ability, interface use, and task domain knowledge.

Chapter 4 details the summary synthesis in this thesis that aimed to provide a clearer snapshot of our existing knowledge based on empirical evaluations of multiple-VIR interface techniques and systems, with the goal of extracting guidelines for design and identifing research questions for the thesis.

Approach and major findings

Our summary synthesis analyzed 19 existing multiple-VIR interface studies and cast findings into a four-point decision tree to design multiple-VIR displays: (1) When are multiple VIRs useful? (2) How to create the low-VIR display? (3) Should the multiple VIRs be displayed simultaneously? (4) Should the multiple VIRs be embedded or separated?

We summarized our findings as design recommendations. We concluded that the number of VIRs should match the number organization levels in the data, and the information displayed in the low VIRs should be relevant, sufficient, and necessary for the supported task. Simultaneous display of the different VIRs was found to be suited to tasks where answers, or information scent leading to the answers (Pirolli et al. 2003), spanned multiple levels of the VIRs. Otherwise, temporal switching of the VIRs should be more appropriate due to simpler and more familiar interactions. The issue of spatial arrangements of the various VIRs remains an open question in research. The questions of overview creation and spatial arrangements of the various VIRs were further examined in the next three studies.

1.2.2 Laboratory experiment: visual memory costs of transformations

Motivations and goals

Geometric transformations are widely used in interface design, particularly in multiple-VIR visualization systems to create the low-VIR overview. In this study, we investigated the visual memory costs of four frequently used geometric transformations: scaling, rotation, rectangular fisheye, and polar fisheye.

Rotation has been used in *embedded* interfaces such as the Hyperbolic tree (Lamping et al. 1995) and to create interactive radial graph layout (Yee et al. 2002). Likewise, scaling is extremely popular; for example, scaling is frequently used to create the low-VIR display in multiple-VIR interfaces, for example, as the lowest zoom level in Summary Thumbnails that provide semantic zooming for webpages displayed on mobile devices (Lam and Baudisch 2005), and as the low-VIR overview in *separate* interfaces for documents (e.g., Hornbæk and Frokjær 2001, Hornbæk et al. 2003) and maps (e.g., Hornbæk et al. 2002).

Unfortunately, scaling only works to a certain extent: when the size of an image is reduced too far, its details become indiscernible. One possible remedy is to selectively scale visual objects such that readability is preserved for the part of the image relevant to the user, while the rest remains available in a reduced form to serve as context. The class of *embedded* techniques, a popular multiple-VIR technique, does so by providing both an unscaled focus (a high-VIR region) and a scaled-down context (a low-VIR region) in a single integrated image (Leung and Apperley 1994; Skopik and Gutwin 2005). Focus + context can be realized using a nonlinear transformation called a fisheye transformation, which has two main variants: rectangular and polar (Leung and Apperley 1994; Skopik and Brown 1992). There exists a large body of work using the fisheye transformation, such as the Fisheye menu to support item selection from a long list (Bederson 2000), Fishnet to display lengthy web documents (Baudisch et al. 2004), DateLens to display calendar data on small-screens (Bederson et al. 2004), and a two-dimensional graph display for large information spaces (Bartram et al. 1995).

While scaling, rotation, rectangular fisheye, and polar fisheye transformations can provide benefits in overview creation and VIR presentation, there is a danger that the transformed image may be too distorted to remain recognizable. This issue is a serious usability concern, since users need to be able to retain, or at least compensate for, their orientation in the visualization after the transformation. They also need to be able to associate displayed components before and after the transformation to equate the two views as the same, or at least holding the same information. Unfortunately, effects of these transformations on visual memory are largely unknown.

Our goal was therefore to systematically measure visual memory costs of these four two-dimensional geometric transformations to guide interface design. Also, we aimed to refine existing design guidelines, such as to mitigate incurred perceptual costs by preserving orthogonal ordering and by applying background grids.

Approach and major findings

Since we were mostly interested in visual memory cost, which is a perceptual cost, we modeled our study design after those in experimental psychology and conducted a laboratory experiment. Instead of using a fully interactive system with scenario-based tasks, we showed static abstract images and studied a three-phased visual memory task (encode, retain, test). We also only measured task completion time and accuracy without soliciting subjective feedback or recording observations.

For each transformation type, we defined a no-cost zone boundary after which we observed degradations in task time and accuracy. We refined the orthogonal-ordering guideline proposed by Misue et al. (1995) where we suggested that instead of preserving left-right and up-down ordering, providing an up-down indicator would suffice. We verified the use of background grids in mitigating visual memory costs in these transformations, and provided further insights as to how they compensated different transformations such as providing distance cues to compensate for distance distortions in rectangular fisheye transformations.

1.2.3 Experimental-simulation study: overview use

Motivations and goals

Creation of the low-VIR view is one of the first steps in the multiple-VIR design process. A low-VIR view corresponds to the overview in *separate* techniques, the context in *embedded* techniques, and the lowest zoom level display in zooming techniques. While it is obvious how to display data at the highest VIR in the detail, focus, or high-zoom displays, how and what to display in low VIRs can be difficult. Ideally, the low VIRs should map to all the data in the data population so that users can select an area of interest for detail explorations at higher VIRs.

When the data is structured at multiple levels that are relevant to the task, that structure can be used to create the lower-VIR views as low-level data can be aggregated and collectively represented by higher-level structures. For example, designers can represent individual species (e.g., Panthera tigris, Panthera leo and Panthera onca) by Genus (e.g., Panthera). Using multiple VIRs for data organized at multiple levels of detail was found to be effective in our summary synthesis, detailed in Chapter 4.

However, when the data has only a single level of inherent structure or has no known structure, designers have little guidance on low-VIR creation. The lack of known data structures may necessitate the display of every datum in the data population, and designers may need to sacrifice the amount of visual detail displayed for each datum to increase the low-VIR's display capacity. This approach is viable if the designer can assume users can recover lost visual details in higher-VIR displays. In other words, designers need to ensure sufficient and perceivable visual details to enable users to select areas of interest in the low-VIR display for further examination.

Displaying visual details does not always guarantee that users can access the displayed visual information. For example, information encoded by text with a font that is unreadable is not accessible to users. In other words, the usefulness of displayed text can be characterized by font readability. For graphical displays, the corresponding visual requirements are more difficult to define despite the rich history of perception research. One such requirement is visual salience.

In general, a visual object is salient when it attracts the user's attention more than its neighbours, and is therefore easily detected (Landragin et al. 2001). One way to achieve extreme visual salience is by visual pop-out, where visual objects with features that can be preattentively processed are spotted quickly and reliably on the display independent of the number of distractors and observer intent (Treisman 1985). However, this extreme approach can be inappropriate when it is unclear *a priori* which of several aspects of the data should be emphasized. Instead, a more appropriate strategy would be to encode visual objects with sufficient salience to enable overview use without having one aspect overpower the others. The low-VIR view would contain a variety of items of similar salience, where the visual target would not draw more attention than the non-targets but could be detected and accessed.

Based on pilot study results and Tullis's (1985) work on display characteristics and visual search time (discussed in Section 3.1.1 in the Related Work Chapter), we selected two perceptual parameters of visual salience out of a collection of six: target visual complexity and visual span. We established the boundaries of these requirements by showing that our participants universally chose to use the low-VIR displays only when the visual targets were structurally simple and spanned a small visual angle. We then focused on situations where these visual requirements were not completely met. The goal of this study was therefore to investigate whether distributing high-VIR details amongst multiple VIRs could relax perceptual requirements established for single low-VIR views, and if the spatial arrangements of high and low VIRs affect overview effectiveness.

Approach and major findings

While laboratory experiments are appropriate in measuring perceptual costs, they are unsuitable vehicles to better understand interface use since they tend to ignore interactivity and participant usage behaviours. The third study in this thesis took the experimental-simulation approach to understand overview use in multiple-VIR interfaces. We therefore studied fully-interactive interfaces with scenario-based tasks, and recorded detailed observations in addition to task completion time and accuracy to gain insights into interface use.

We found that, surprisingly, neither of our *separate* or *embedded* multiple-VIR interfaces provided performance benefits when compared to the optimal single-VIR interfaces. However, we did observe benefits in providing side-by-side comparisons for target matching in the *separate* interface. We conjectured that the high cognitive load of multiple-VIR interface interactions, whether real or perceived, is a more considerable barrier to their effective use than was previously considered.

Overview design in multiple-VIR interfaces is a complex issue and remains an open question in research, which is discussed further in the last chapter of this thesis, in Section 9.1.1.

1.2.4 Field evaluation: Session Viewer at Google Inc.

Motivations and goals

Interface use is a complex phenomenon that cannot be adequately studied in laboratories (Plaisant 2004; Shneiderman and Plaisant 2006). Finding effective methods to evaluate visualizations is an open research area, and is discussed in Section 9.1.4. Traditionally, the information visualization community has focused largely on experimental-simulation studies to compare between visualizations. Generally, these studies lack realism and have had limited success in discovering unexpected factors that affect interface use, in ensuring participant engagement during the study, and in studying domain expertise. Laboratory studies also cannot provide insights into prototype deployments in the workplace to discover issues that may be unrelated to information visualization techniques, but nonetheless may determine the outcome of technology transfers.

We therefore focused on the web session log analysis application domain and built a visual analytic tool called Session Viewer to examine some of these issues. Our goal in the fourth study in this thesis was therefore two-fold: to examine design choices made in building Session Viewer, and to study system requirements in the workplace.

Approach and major findings

We conducted a field evaluation at Google Inc. with seven log-analyst participants working on their own data and their own tasks, reported in Chapter 8. Taking a qualitative approach, we collected 20 hours of tool-use observations and grouped our findings into two design themes: (1) design implications in dealing with real-world noisy data, and (2) factors that lead to tool reception in the workplace.

In terms of visual-design findings, we found that noisy data requires substantial validation, and tools should convey the gist of the data. Tools should allow fluid data-view projection to support frequent analysis direction changes.

We examined design choices made during the creation of Session Viewer. We found that our analysts could effectively identify interesting sessions for further examination using Session Viewer's scrollable overview, which displays small multiples of sessions that are interactively reorderable, even though the overview did not simultaneously show all data. We also found that the *separate* visualization technique, when coupled with statistical data attributes, was surprisingly effective in supporting data cleaning and data selection. In terms of spatial layout, we found a tradeoff in optimizing screen space for single-population analysis and multiple-population comparisons.

We also identified three main considerations for system deployment. We found that the level of data configurability should be based more on target users' technical skills than on existing data schema. We believe the strongest determinant of our tool's reception was its unique contribution to the analysis process by bridging between existing analysis practices. However, the complexity of a powerful tool may deter its use. In our case, integration with current tool sets was found to be less crucial than we previously assumed.

1.3 Thesis Contributions: Application

The application aspect of this thesis involves our detailed design study of a information visualization system prototype called Session Viewer to support web session log analysis. Session Viewer is our proposed solution to address existing data exploration concerns of web session log analysts based on their current analysis practices, characteristics of web session log data, and the task of exploratory data analysis. The software is unique in its ability to handle multi-level data and support cross-level analysis. Chapter 7 covers the design of Session Viewer in detail.

Session Viewer plays a major role in this thesis as it provides a test bed to examine our design ideas based on knowledge gained in the first three evaluations in the thesis. We identified two research themes from the multiple-VIR design summary synthesis in Chapter 4: overview creation and the choice between the *embedded* and *separate* approach to spatially arrange VIRs. Specific aspects of these two themes were examined further in two subsequent evaluations: the visual-memory experiment in Chapter 5 studied perceptual costs of image transformation, and the overview-use study in Chapter 6 looked at visual details required for effective overview use.

In Session Viewer, we provided a concrete design example in the application domain of web session log analysis to further explore design choices associated with these two research themes. Section 7.6 describes our design evolution of Session Viewer and listed our design choices, while the last evaluation of this thesis, the field evaluation, examined the impact of our design choices.

In addition to our design study of Session Viewer, our application contributions are therefore:

1. We proposed a solution to create overviews of large data sets. Instead of data pre-selection, we allowed scrolling in our overview that displays session objects as small multiples with each session object comprised of events. We augmented our overview with visually and interactively linked session attribute for each session object. With session reordering based on attributes, we found in our field evaluation that analysts could effectively isolate interesting sessions in a population for further analysis.

2. We provided positive evidence for using the *separate* technique to display multiple levels of data. We found in our field evaluation that the *separate* technique provided close mapping between VIRs and analysts' concept of session logs, and the technique was found to be effective in supporting data validation and session selection.

Chapter 2

Background: Exploratory Data Analysis and Visualization

Given the focus of exploratory data analysis of the thesis, this chapter provides background information in general, and the thesis domain of web session log analysis in specific.

Data analysis has long resided in the realm of statistics, with established methods that summarize data populations and model patterns in the data. An essential step in data analysis is data exploration, where the analyst tries to understand the data to generate hypotheses. This step is arguably difficult, especially in situations where the data size under analysis is large, and when the data contains multiple subpopulations that are unknown prior to the analysis. Visualization systems, by taking advantage of the human visual system's ability to process large number of visual signals in parallel, has been suggested as a viable solution to aid data exploration.

This chapter surveys these areas in more depth. The survey begins with a survey of the exploratory data analysis (EDA) task, based mostly on the statistics and analytical methodology literature (Section 2.1), followed by a list of proposed roles played by visualization in data exploration (Section 2.2). These proposed roles are solidified into design challenges and requirements for visualization systems that support EDA (Section 2.3). The last section in this chapter focuses on the application domain of this thesis, web session log analysis. Section 2.4 explains the rationale of our domain choice and provides background information on the subject.

2.1 Exploratory Data Analysis

Data analysis is a complicated process, which is part of a larger context of inquiry. Tukey described the process of data analysis as a continuum from exploration to confirmation data analysis (Tukey 1986), that generally starts with data exploration. The term "exploratory data analysis" (EDA) was first coined by Tukey in 1977 in his seminal work of the same title (Tukey 1977). The goal of EDA is to discover patterns in data. The emphasis is to study data to obtain an understanding. This thesis focuses on EDA instead of the entire analysis process, since the goal of visualization is to support EDA (Section 7.3) and since there exist numerous statistical packages that address confirmatory analysis. Instead of discussing specific EDA statistical methods such as data transformation and residual analysis (Tukey 1986; Leinhardt and Wasserman 1979), we focus on EDA philosophies to understand how visualization can play a role in supporting the task of EDA.

According to Hartwig and Dearing (1979), the essence of EDA is skepticism and openness: skeptical of potentially inappropriate use and fallacies of data representations and analytical methods, and open to unanticipated patterns in the data. EDA therefore focuses on tentative model building and hypothesis generation in an iterative process of model specification, residual analysis, and model re-specification (Behrens 1997). In other words, analysts should form their hypotheses while studying the data, not before.

Ho (1994) further characterized the logic of EDA based on the work of Peirce in 1878, and suggested applying the process of abduction to EDA. Abduction is the process where analysts "look for a pattern in a phenomenon and suggest hypothesis" (Ho 1994, p. 15). Its purpose is to generate guesses of a kind that deduction can explicate and that induction can evaluate. Ho argued that in exploratory data analysis, "although there may be more than one convincing patterns, we 'abduct' only those which are more plausible" and that "exploratory data analysis is [therefore] not trying out everything" (Ho 1994, p. 16), since in general, it would be impossible to falsify every possibility. On the other hand, exploratory data analysis is not to make hasty decisions, as "researchers must be well-equipped with proper categories in order to sort out the invariant features and patterns of phenomena" (Ho 1994, p. 18).

Even though EDA is more a philosophy of approach than a prescribed method, several researchers have provided concrete steps to achieve these goals. The analysis starts with analysts studying the data. Hartwig and Dearing (1979) advocated a bottom-up approach that starts by understanding the data distribution of each data dimension value, building the understanding to correlate between dimension pairs, examining the network of relationships between the variables, and building models about the data. Sanderson and Fisher (1994) adapted the philosophy of EDA to sequential data exploration, or exploratory sequential data analysis (ESDA), for data with integral temporal components. Sanderson and Fisher (1994) described the process of ESDA as "Eight Cs", where the first three Cs (Chunks, Comments and Codes) are initial steps to understand the data, and the next four steps (Connections, Comparisons, Constraints and Conversions) are devoted to data exploration. The last step, Computations, is where analysts reach the conclusion of the analysis.

The outcome of EDA or abduction is therefore a set of plausible models that can be further assessed. Ho (1994) argued that the next stage in the analysis is to refine the hypothesis by drawing logical consequences using deduction, or "a process through which we start with general claims or general assertions and ask what follows from these premises" (Reisberg 2001, p. 411). However, since deduction relies on the truthfulness of the premises, empirical justification of the hypotheses with data is required. That is the next stage of analysis, or induction, "a process in which one begins with specific facts or observations and then draw some general conclusion from them" (Reisberg 2001, p. 378). Ho's induction process is akin to confirmatory data analysis (CDA) modes of Tukey (1986). In the first stage, or the rough CDA mode, analysts use probabilistic approaches such as confidence intervals or significant tests to initially assess the plausible hypotheses. In the next CDA mode, specific hypotheses are tested using a strict probabilistic framework following a decision theoretic approach. This process is cyclical, leading to a good description of the data by successive approximations.

A related area is the work on analytical reasoning and sense making. A comprehensive review is beyond the scope of this thesis but can be found in Chapter 2 of Thomas and Cook (2005). However, I will briefly mention Pirolli and Card's (2005) Sensemaking process, which is partly based on their earlier work on information foraging theory (Pirolli and Card 1999). Information foraging theory is widely known in the fields of human-computer interaction and information visualization, and has been applied to web-searching support tool designs (e.g., Olson and Chi 2003) and to model web usage behaviours (e.g., Card et al. 2001; Chi et al. 2001).

Briefly, Pirolli and Card's (2005) Sensemaking process has two main loops, the foraging loop based on Pirolli and Card (1999) and the sensemaking loop based on Russell et al. (1993). The bottom-up process in the foraging loop is akin to EDA as prescribed by EDA advocates such as Hartwig and Dearing (1979). According to Pirolli and Card (2005), analysts start with the step of "search and filter" to collect relevant external data sources into a temporary storage space or "shoebox", after which the documents in the shoebox are read to extract evidence to draw inferences and to trigger new hypotheses and searches.

In summary, data analysis operates in a cyclic fashion in terms of processes (abduction, deduction, induction) or modes (EDA, rough CDA, CDA). The analyst, when trained in different modes of analysis, moves fluidly between exploratory and confirmatory processes in a single analysis (Behrens 1997). Patterns and unexpected outcomes are regarded as starting points for hypothesis generation and future testings rather than as statistical conclusions. In addition, the analyst familiar with EDA will explore data patterns associated with the hypothesized main effect to make sure the CDA was not misled by unrecognized patterns that can lead to conclusions inconsistent with the data.

Given our visual capability to process large amounts of information in parallel, visualization is an attractive tool to support EDA.

2.2 Roles of Visualization in EDA

Advocates of EDA often recommend using graphical displays to represent data. According to Tukey, "the greatest value of a picture is when it forces us to notice what we never expected to see" (Tukey 1977, p. vi). Thomas and Cook summarized the six basic ways where visualization can amplify human cognitive capabilities (2005, p. 46):

1. Increasing cognitive resources, such as by using a visual resource to expand human working memory;

2. Reducing search, such as by representing a large amount of data in a small space;

3. Enhancing the recognition of patterns, such as when information is organized in space by its time relationship;

4. Supporting the easy perceptual inference of relationships that are otherwise more difficult to induce;

5. Perceptual monitoring of a large number of potential events;
6. Providing a manipulable medium that, unlike static diagrams, enables the exploration of a space of parameter values.

In addition to the visual display, modern visualization systems offer interactivities that further assist EDA, as interactivities allow analysts to quickly and fluidly "hold and assess" working hypotheses (Behrens 1997), either by viewing the data in different perspectives, or by emphasizing different parameters of a problem (MacEachren and Kraak 1997).

2.3 Challenges and Requirements for EDA Visualization Systems

Good (1983) posed two questions for EDA tool designers:

> How should we present a collection of k-tuples to match the cognitive powers of the analyst so that he can (i) see patterns in the data, and (ii) formulate sensible hypotheses about the data?

Existing EDA literature provides further and specific design requirements, such as to provide context for data interpretation, to provide re-representation and multiple representations of data, and to provide links between different data views. These requirements were part of the considerations in creating Session Viewer, the visual analytic tool in this thesis built to support web session log analysis. Design considerations of Session Viewer are listed in Section 7.3.

2.3.1 Provide context for data interpretation

Interpretation of data usually requires comparison, either to existing standards, or to related values. According to Woods et al. (2002), "Presenting data in the context shifts part of the burden to the external display rather than requiring the observer to carry out all of this cognitive work in the head." (p. 32). Outliers can only be detected when the analyst understands how the datum departs from, or conforms to, the typical expected case (Woods et al. 2002).

In addition to interpreting individual data, analysts often need to discover relationships in the context of the field of practice. Such a frame of reference is a fundamental prerequisite for depicting relations rather than simply making data available. Instead of organizing displays around pieces of data, displays should organize data in units meaningful to the application.

2.3.2 Provide re-representation and multiple representations of data, and allow fluid traversals

EDA advocates have stressed the fact that conventional views of data are more based on habits than for enhancing analysis, and analysis tools should support re-representation of data (Behrens 1997).

In addition, data analysis often requires checking multiple hypotheses, as "science is the holding of multiple working hypotheses" (Chamberlain 1965). Almost always there are multiple frames of reference that apply. Each frame of reference is like one perspective from which an analyst views or extracts meaning from data (Woods et al. 2002), and patterns and anomalies in data may only be obvious in certain views. Multiple representations of data are therefore needed to view data in different ways using multiple scales and perspectives, both spatially and conceptually (MacEachren and Kraak 1997; Sanderson and Fisher 1994).

Another important aspect of EDA tools is to allow analysts to shift perspectives fluently so as to support the fast data-view projection changes in the process (Behrens 1997; Woods et al. 2002), as the definition of relevance in data can be highly context sensitive (Woods et al. 2002).

2.3.3 Provide linking between data views

An individual datum is meaningless unless the data analyst can interpret its value in the context of other data. In some cases, the analyst may need to interpret data values in the context of the theory that unifies the data (Behrens 1997). Indeed, part of the analysis is to discover the multiple potentially relevant frames of reference and to find ways to integrate and couple these multiple frames (Woods et al. 2002).

EDA tools should therefore link multiple data views to allow propagation of changes in one plot to all relevant plots (Behrens 1997), and to identify relationships among variables (MacEachren and Kraak 1997).

2.4 Web Session Log Analysis

Given the general background on exploratory data analysis and possible roles played by visualization, this section focuses on the chosen application domain of this thesis: web session log analysis.

To better study exploratory data analysis, web session log analysis was chosen as the thesis's design example due to problem relevance and commonality with other analysis situations. In the year 2002, the size of the World Wide Web was about 533x10^3 terabytes (Lyman and Varian 2003). The same study reported that in the year 2002, each individual in the United States spent about 100 hours per year online (Lyman and Varian 2003). With the increasing importance, complexity and volume of the web, providing better web information-seeking support is essential and requires an understanding of web search usage behaviours.

Researchers have used methods ranging from field experiments and studies to web session log analyses to achieve this goal. Typically, field-study researchers observe participants perform their information-seeking activities on the web in participants' own environments, followed by interviews to further understand participants' tasks, motivations, thinking processes, and expectations. Published examples include Jones et al.'s (2001) study to identify methods people use to manage web information for re-use, Sellen et al.'s (2002) study to observe web activities conducted by knowledge workers, and Teevan et al.'s (2004) study on orienteering behaviours in directed search.

While such observational studies can reveal rich and detailed information and allow for deep understandings of naturalistic search behaviour in the context of users' goals, the approach is too labour intensive for large population analyses. Session log analysis is a more scalable alternative. Session logs are computer logs that capture user actions in units of sessions. Session logs can be obtained by server- or client-side logging. Server-side logs include transactions of web search engines, Intranets and web sites, while client-side logs are usually recorded by plug-in tools installed in users' web browsers. Even though session logs cannot capture user goals and intent, they do capture realistic search behaviours as users perform real information searches in their own environments uninterrupted by the data collection mechanism. However, session logs are difficult to analyze due to the large data size and complex compositions.

Before discussing existing difficulties in web session log analysis in Section 2.4.2, we first explain the structure and composition of web session logs.

Pane	Paneli	Que	Navigation_Event	Event_	URL	Title
1	15238	44	Begin_Section	1		
2	15238	45	Begin_Objective	1		
3	15238	45	Go to Start URL of Objecti	31	http://www.msn.com	
4	15238	45	Browser Navigate	59	http://search.msn.com/results.asp	MSN Search: london and tours
5	15238	45	Browser Navigate	118	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
6	15238	45	Browser Add Favorite	175	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
7	15238	45	Browser Navigate	204	http://www.london-tours.info/londo	Discovering London (Full Day Tour No 4) Ticke
8	15238	45	Browser Back	244	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
9	15238	45	Browser Navigate	265	http://www.london-tours.info/londo	Oxford, Stratford & Warwick Castle (Tour 23) Ti
10	15238	45	Browser Back	316	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
11	15238	45	Browser Back	320	http://search.msn.com/results.asp	MSN Search: london and tours
12	15238	45	End_Objective	360		

Figure 2.1: A sample web session log from user study described in Russell and Grimes (2007). Each row is an event, with the sequence number, participant ID, question number, navigation type, event time, URL, and webpage title.

2.4.1 Session logs

One method to obtain web session logs is by client-side logging. This method is commonly used in short-term user studies to understand web-usage behaviours. Figure 2.1 shows a sample log from one such study (Russell and Grimes 2007).

In session logs, the basic unit of analysis is a **session**, a time-stamped sequence of events. An **event** corresponds to a user action, such as submitting a query to the search engine, clicking on the next page link, or clicking on a web result. Each event has attributes, such as a time-stamp, URL, action type (e.g., web search, webpage click), search domain (e.g., Image, Product), and the submitted query.

Since a session is simply an ordered list of events, aggregates of event attributes become session attributes. Examples include the total number of events in a session and the total dwell time of a session. In addition, session logs may contain participant feedback at the session level that also constitute session attributes. Examples include task satisfaction and self-reported task outcomes.

A **session population** is a group of sessions with shared characteristics such as usage patterns. In short, a session log has structure at three levels: session population, session, and event.

In general, a session is a multi-dimensional data object. Most dimensions are single values (e.g., event count per session), but one dimension is a time-ordered sequence of event objects. Each event is itself a multidimensional datum. In other words, the session data object is a **multi-level data object**.

2.4.2 Log analysis: existing practices and problems

One analysis option is a detailed study of individual sessions. One example is Kellar et al.'s (2007) study to better understand four types of information-seeking tasks. These task categories were created by detailed studying of task descriptions in session logs. In their case, they looked at 40 task descriptions sampled from a larger set of logs generated by six participants over four days.

This detailed study approach can lead to interesting insights, but is very labor intensive. In Kellar et al.'s (2007) case, the task categories were created by ten focus-group participants over an hour, and further refined by the researchers themselves after the focus-group session. The large amount of time required for detailed analysis thus limits sample size, and sampling from a larger data set may be potentially biased, which may render general conclusions drawn inaccurate or even misleading.

A more scalable and commonly used alternative is to compute overall population statistics at multiple levels, such as unique term frequency at the query level and event type frequency at the session level (Jansen 2006), or more complex web usage mining methods to model and predict user behaviours (e.g., Pierrakos et al. 2003). While these statistical approaches are scalable and effective, they tend to be hypothesis-driven and confirmatory rather than data-driven and exploratory, and may not uncover unexpected trends or may obscure subpopulation differences in the data. In addition, without exploring the data in detail, hypothesis formation can be difficult.

The detailed and the statistical approaches to analyzing web session logs are therefore complementary, as one approach can potentially mitigate the shortcomings of the other. For example, the difficulty in selecting representative sessions for detailed analysis may be mitigated if the selection could be guided by statistical session attributes such as session duration distributions. For statistical analysts, being able to view representative sessions of various statistical populations in detail may facilitate hypothesis generation and uncover unexpected trends. The key challenge with session log analysis is therefore to bridge between detailed and aggregate analysis to better support data exploration.

The lack of cross-level analysis is not unique to web session log analysis. Tukey and others advocated exploratory data analysis using graphical plots to ensure adequate data exploration and understanding before applying statistical methods, and data analysis was considered as a continuum from exploratory to confirmatory analysis (Tukey 1986). Visual exploratory analysis (VEA) is an attractive approach given our visual capabilities to spot trends, patterns, and anomalies. In practice, effective VEA requires a sophisticated visualization tool. Building such a tool is the subject in Chapter 7 of the thesis.

Chapter 3

Related Work

Given that this thesis studies how interfaces that display multiple levels of data detail can support exploratory data analysis using four diverse research strategies, this chapter surveys related empirical studies employing each of the four strategies in Section 3.1: laboratory experiments (Section 3.1.1), experimental simulation (Section 3.1.2), field experiments (Section 3.1.3), and systematic reviews (Section 3.1.4).

In terms of applications, we survey the general area of visual analytics where interactive visual interfaces are used to support analytical reasoning in Section 3.2. More specifically, in Section 3.2.1 we cover visualization systems designed for visual exploratory analysis, as well as three categories of visualization techniques to display multiple data forms, multiple data levels, and multiple data dimensions.

Since our chosen application domain for our field evaluation in Chapter 8 is web session log analysis, this chapter also reviews visualization specifically designed for to visualize web session logs (Section 3.2.3, the more general computer logs (Section 3.2.4), and non-visualization approaches to web session log analyses (Section 3.2.5).

3.1 Empirical Studies

As the field of information visualization matures, researchers have begun to acknowledge the need to evaluate existing visualization techniques and systems (Chen and Czerwinski 2000). Indeed, a 2007 study found that over 90% of the papers that were accepted for the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI) in 2006 included formal evaluations, where only about half of the papers did in 1983 (Barkhuus and Rode 2007).

The main themes in empirical studies include laboratory experiments to study human perceptual and cognitive capabilities in interacting with visualizations, experimental-simulation studies to evaluate information visualization techniques, field experiments to study information visualization systems use in practical contexts, and systematic reviews of study results to obtain overviews of existing knowledge.

This section delineates previous efforts in these four areas of investigations, and how this thesis contributes to and extends existing work. Due to the large body of work that exists in perceptual and visualization-technique evaluations, this section focuses on work related to multiple visual resolution interface resolution (VIR) visualization techniques when discussing these two themes, as the study of multiple-VIR interfaces is the main focus of this thesis.

3.1.1 Laboratory experiments to study human perceptual and cognitive capabilities

Perception and cognition research in information visualization and related fields such as human-computer interaction and experimental psychology aim at establishing models and limitations of human perceptual and cognitive capabilities to better visualization designs. Ware (2004) focuses on subjects in vision research that are pertinent in visualization design and extracts design principles that can be applied to better information display.

The major areas of vision research study how humans perceive various kinds of visual stimuli such as colour, shape, motion, and depth, along with research on object recognition and visual attention. Knowledge gained from these kinds of studies has been applied to visualization design. Examples include using study results of colour perception to derive design guidelines to encode nominal or continuous data in interface design (e.g., Ware 2004, p. 123–138); using depth perception to create three-dimension visual representations on two-dimensional output devices (e.g., Hubona et al. 1999); using Gestalt laws of perceptual organization to facilitate visual search (e.g., Hornof 2001); and using pre-attentive processing for rapid visual numerical estimations (e.g., Healey et al. 1996).

In addition to establishing human limitations in interacting with visualization displays, researchers have also proposed models to describe human behaviours. Examples of models in interface design and evaluation include Card et al.'s (1983) GOMS (Goals, Operators, Methods, and Selection rules) model for observations and evaluations of interactions, Plumlee and Ware's (2006) general model for navigation-intensive information seeking, and Pirolli and Card's (1999) model on information foraging. Pirolli and Card's (1999) model on information foraging was discussed in Section 2.1 in the context of examining the task of exploratory data analysis to derive design requirements for Session Viewer, a visualization tool to support web session log analysis.

On visual memory and geometric transformations

Of particular interest to evaluating multiple visual information resolution (VIR) interfaces are studies on visual memory and on perceptual costs in geometric transformations, a technique frequently employed in multiple-VIR interface design, and studies on visual salience, a subject closely related to low-VIR overview designs.

Several studies examined the roles of visual memory in interface design. An example is Robertson et al.'s (1989) Data Mountain. Data Mountain makes use of our spatial memory and allows users to place thumbnails of documents at arbitrary positions on an inclined plane in a three-dimensional desktop virtual environment using a simple two-dimensional interaction technique. In a subsequent study, the researchers demonstrated that users could retrieve documents faster with better accuracy than the benchmark Microsoft Internet Explorer Favorites (Robertson et al. 1989), and the performance did not measurably degrade after six months (Czerwinski et al. 1999). While these studies demonstrated the use of spatial memory in interface design, they did not quantify perceptual limits applicable in designs.

Skopik and Gutwin (2005) looked at the effects of rectangular fisheye transformation on visual memory and found that distortions increased the time required to remember and find target nodes, but without affecting task accuracy. The researchers proposed and demonstrated the effectiveness of adding visual markers, called "visit wear", of the places previously visited by users to offset distortion effects and improve navigation.

Several studies have looked at effects of geometric transformations. Shepard, Cooper, and Metzler examined the effects of mental rotation in a series of experiments where participants were asked to determine whether two geometric objects were identical but viewed from different perspectives (Cooper and Shepard 1973; Shepard and Metzler 1971). Their experimental results show a clear linear relationship between the angle of rotation and response times, suggesting that participants mentally rotate one of the stimulus to match the other. While these results suggest a mechanism for mental rotation, the experiments did not aim to study visual memory costs.

Two other studies measured perceptual costs of geometric transformations in visual search tasks using abstract images based on time and accuracy measurements. Rensink (2004) found no measurable cost for translational shifts up to at least 2 degrees of visual angle, or 2 cm at a viewing distance of 55 cm. Performance was not measurably affected for rotations up to 17 degrees, but degraded sharply beyond that. Scaling was found to be invariant at a reduction factor of two, but created a measurable cost at a factor of four. In another series of experiments involving visual search on displays with nonlinear polar fisheye transformations, Lau et al. (2004) found that the studied transformations had significant time costs, with performance slowed by a factor of almost three under large distortions. Interestingly, the study did not find any benefits in adding background grids to their images. In fact, the researchers found that grids caused participants' performance to slow down, suggesting that the grids only added to the perceptual noise. However, these two experiments focused mostly on visual search. Even though visual search is a common component of many of the visual operations in information visualization (Ware 2004), other important factors are still at play. One of these factors is visual memory.

Our laboratory experiment in this thesis, reported in Chapter 5, contributes to these previous efforts by systematically quantifying visual memory costs in four types of geometric transformations: scaling, rotation, rectangular fisheye, and polar fisheye. The study also looked at if and how background grids can mitigate incurred visual memory costs. This study was conducted in a larger context of understanding low-VIR display creation and VIR integration in multiple-VIR interface design.

On visual salience

Visual salience is a broad topic and a vast amount of human vision research has been done to measure and to understand the phenomenon. In general, a visual object is salient when it attracts the user's attention more than its neighbours, and is therefore easily detected (Landragin et al. 2001). Since the study of visual salience has a long history in vision research, a comprehensive review of the literature is beyond the scope of this thesis, this section only includes work that is directly applied in interface design and evaluation.

To automatically evaluate interfaces, researchers have built predictive models to measure visual salience. Using plain character displays, Tullis (1985) identified six display characteristics that correlated with visual search times: (1) the overall density of characters on the display; (2) the local density of other characters near each character; (3) the number of distinct groups of characters; (4) the average visual angle subtended by those groups; (5) the number of distinct labels or data items; (6) the average uncertainty of the positions of the items on the display. Recent efforts in measuring visual salience were based on image processing and statistics models, for example, Rosenholtz et al.'s (2005) Feature Congestion model measured display element saliency using color and luminance contrast to quantify display clutter.

In terms of mechanisms, one well-studied area is preattentive vision, or visual pop-out, where researchers have discovered a limited set of visual properties such as colour and motion that are detected very rapidly and accurately by the low-level visual system (e.g., Palmer 1999, p. 554–560; Treisman 1985). The information visualization community has incorporated much of this knowledge into its design guidelines for visual encoding. One example is using visual features that can be preattentively and individually processed to encode multi-dimensional data (Ware 2004, p. 151–156), such as texture and colours (Healey and Enns 1999) or motion (Huber and Healey 2005). Another application is to highlight visual objects on displays by encoding them with pre-attentively distinct visual symbols.

Our efforts in applying measurements of visual salience in interface design reside in the area of low-VIR overview design. Typically, low-VIR overviews in multiple-VIR interfaces need to accommodate a large volume of data to allow users to select individual datum for further examinations at higher details. Such practice is in accordance with Shneiderman's (1996) information-seeking mantra of "overview first, zoom and filter, then details on demand" (p. 337). Displaying a large number of visual objects on the low-VIR overviews can lead to visual cluttering when the number of objects presented on the overview exceeds the perceptual capability of its users, thus rendering the overview ineffective. One proposed solution is attention filtering using colour and intensity coding to help users segregate their visual fields so that they can focus on regions that are pertinent for the task at hand. This approach has been verified in Yeh and Wickens's (2001) study on designs of electronic map displays.

The third study in this thesis, detailed in Chapter 6, examined a different aspect of the overview problem. Instead of taking the decluttering approach, we investigated how visual objects could remain visually available to users without being overtly salient, as in the phenomenon of visual pop-out. We adapted two of Tullis's (1985) display characteristics, the number of distinct groups of characters as visual complexity and the average visual angle subtended by those groups as visual span, to graphical displays and investigated the limits to which visual objects remained available for users to select for further detailed examinations. This study was conducted in a larger context of studying low-VIR overview creation and VIR arrangements in multiple-VIR interfaces.

3.1.2 Experimental-simulation studies to evaluate multiple-VIR techniques

While the vision science literature offers valuable advice to designers in their choice of visual encoding, the studies have generally focused on single static images. To better evaluate information visualization techniques, researchers have to consider interactivity. There exists a large amount of experimental-simulation studies to evaluate visualization techniques (see Chen and Yu 2000 for a meta-analysis). This section focuses on studies that aim at evaluating multiple visual information resolution (VIR) techniques such as zooming, focus + context, and overview + detail. Since Chapter 4 details a summary synthesis of 19 existing studies in this area to better guide multiple-VIR interface design, this section only briefly lists related work in the general area of multiple-VIR interface studies. Instead, the focus here is evaluations of overview use, which is the subject of the third study in this thesis detailed in Chapter 6.

Although study results of multiple-VIR interfaces are sometimes characterized as mixed (e.g., in Nekrasovski et al. 2006), the situation becomes clearer when we categorize the studies. In cases where the task required multiple levels of the displayed data, study results generally show that multiple-VIR interfaces outperformed their high-VIR counterparts. Examples include Schaffer et al.'s (1996) network repair task where the answers involved links at all levels of a network, and Hornbæk et al.'s (2003) essay-writing task where participants were required to summarize the main points of an electronic document. In both cases, interfaces that showed multiple levels of details simultaneously were found to better support the study tasks than interfaces that showed single data levels.

In cases where the data set structure had only a single intrinsic level, multiple-VIR interfaces were found to be beneficial only when the low-VIR display provided perceivable details required by the task. For text, perceivability is simply readability, and unreadable text on low-VIR overviews does not enhance task performance. The situation is well illustrated by Baudisch et al.'s (2004) study on information searches on web documents. Their multiple-VIR interfaces displayed web documents with guaranteed legible keywords, but surrounding text was too small to read. Their single-VIR interface was a high-VIR scrollable web browser. The study demonstrated performance benefits for their multiple-VIR interfaces over the high-VIR browser, but only for selective tasks. When the task only required reading the legible keywords, as in their *Outdated* task, their multiple-VIR interfaces outperformed their high-VIR browser. The superior performances supported by their multiple-VIR interfaces were probably due to participants' ability to answer the task questions based on information displayed on the low-VIRs alone. Since the low-VIR displays were considerably smaller than the high-VIR display, the low-VIR displays effectively concentrated task-relevant information. In contrast, when the task required reading text around these keywords, as in the *Analysis* task, having a low-VIR display did not result in performance benefits. One possible explanation is that participants did not find the low-VIR useful and focused instead on the high-VIR displays.

Similarly, in North and Shneiderman's (2000) study, their multiple-VIR interface had a low-VIR overview that displayed the names of geographic states in the United States. These names acted as hyperlinks for relevant passages in the high-VIR view that provided detailed census information for these states. Again, their single-VIR interface was a high-VIR scrollable browser. The study found that when the answers were available on the low-VIR overview, participants did not need the high-VIR view as their performance was not affected by the lack of interactive coordination between the low- and the high-VIR displays. In cases where the tasks required information that was only available on the high-VIR view, interactive coordination was crucial as participants used the low-VIR hyperlinks as shortcuts to reach relevant passages in the high-VIR view.

Another example is Hornbæk and Hertzum's (2007) study, which investigated visualizations for large numbers of menu items. Their Multifocus interface provided larger numbers of readable menu items in the low-VIR regions based on *a priori* significance, while the other interface implemented Bederson's (2000) Fisheye Menu and displayed unreadable items in the low-VIR regions at the extreme ends of the menus. Even though the study failed to find differences between these two interfaces in terms of participant performance, satisfaction ratings or subjective preference, eye-tracking results suggested that participants used the low-VIR regions more frequently in the Multifocus interface trials. The researchers thus questioned the use of screen space to provide unreadable text as being beneficial.

For non-textual graphic displays such as geographic maps, one study has demonstrated the costs of ineffective low-VIR overviews. Hornbæk et al.'s (2002) study found that having a low-VIR overview resulted in slower performance times and worse recall accuracy for their Washington map trials, and their Montana map trials had generally poor performance results. Their results suggest that the failure of the overviews was partly due to insufficient details provided to support their study tasks: the Montana map itself was single-level and did not offer enough meaningful map contents at low VIRs to guide region selections, and the Washington map display did not show enough details at the overview level to support their tasks.

Given the delicate balance between the need for concise yet perceivable visual objects on low-VIR views, the third study in this thesis, detailed in Chapter 6, was conducted to study perceptual requirements for low-VIR graphical visual targets to be reliably accessible to users of multiple-VIR interfaces.

3.1.3 Field experiments to understand visualization system use in the field

While experimental strategies can be effective in evaluating specific visualization techniques, such approaches fall short in evaluating visualization systems, as the number of factors that may influence system use and reception is large and, in many cases, unpredictable (Shneiderman and Plaisant 2006). Also, field experiments can study interface-use questions in ecologically-valid settings. This section therefore focuses on evaluations of whole visualization systems using field experiments.

To the best of our knowledge, only four sets of field experiments were conducted to look at visualization system use in exploratory data analysis. With expert meteorological forecasters analyzing data provided by the researchers, Trafton et al. (2000) found that users tended to be goal-directed when dealing with large amounts of data and mainly extracted qualitative information from visualizations. González and Kobsa (2003b) reported two studies on InfoZoom with five office workers and found that even though InfoZoom provided benefits in creative discovery, the stand-alone tool was not integrated into participants' daily analysis routine in the long run (González and Kobsa 2003a). Saraiya et al. (2004) compared the number of insights generated using five visualization tools for microarray data including Clusterview, TimeSearcher, Hierarchical Clustering Explorer, Spotfire, and GeneSpring. Their study found that these tools did not adequately link the data to biological meaning, different visualization tools resulted in different kinds of insights, and ineffective interaction mechanisms severely reduced tool usability. This initial study was followed by a more in-depth longitudinal study (Saraiya et al. 2006) with participants using their own data to address the motivation issue. The longitudinal study reported how data analysts used a combination of tools in their analysis processes and further examined the issue of interaction in these tools. Seo and Shneiderman (2006) evaluated the Hierarchical Clustering Explorer with three case studies and an e-mail user survey to evaluate the software.

Two studies looked at other uses of visualization systems. Bellamy et al. (2007) reported the design and pilot deployment of a visualization to monitor and manage compliance processes, and concluded that diagnostic visualizations should provide an integrated view of all required information at sufficient detail. Biehl et al. (2007) evaluated FASTDash, a visualization to improve team activity awareness. Their field experiment found that FASTDash improved team awareness, reduced reliance on shared artifacts, and increased project-related communication.

The fourth study in this thesis, detailed in Chapter 8, contributes to previous efforts in studying visualization systems in ecologically-valid settings. The chapter details a field evaluation of Session Viewer, a visualization tool to support web session log analysis. Our efforts were directed to examine particular design choices made during the creation of Session Viewer such as overview use and spatial arrangements of various VIRs, and to study design issues unique to visualization use and deployment in the workplace in general.

3.1.4 Systematic reviews to summarize existing visualization study results

No single user study can provide a complete picture of multiple-VIR interface use due to the large number of factors involved. Similar to the idea that groups composed of independent thinkers tend to be more accurate in their conclusions (Surowiecki 2004), systematic reviews can potentially produce more accurate views of existing research questions than any individual study.

To the best of our knowledge, two systematic reviews have been conducted to study visualization systems. The first is Chen and Yu's (2000) meta-analysis. Based on 35 experimental studies published between 1991 and 2000, the researchers isolated six studies that satisfied their selection criteria. They found two broad types of causal relationships: (1) effects of visual-spatial interfaces on information retrieval, and (2) effects of cognitive ability of users on information retrieval. The researchers concluded that for users with the same level of cognitive abilities, simpler visual-spatial interfaces tended to result in better task performance.

The second systematic review is Hudhausen et al.'s (2002) meta-study of the effectiveness of algorithm visualizations (AV). Due to the diversity of the 24 experimental studies under analysis, the researchers decided against a general statistical meta-analysis. Instead, the analysis employed a vote-counting approach within groups defined with dependent and independent variables in the studies. The meta-study found that how students used the AV technology was more important than what the visualization showed.

Two other reviews can also be included in this category, even though they did not directly examine visualization systems. Tversky et al. (2002) provides a narrative review to identify scenarios in which using animation in education is effective. The paper also explains the reasons behind ineffective use of animation based on cognitive principles of congruence and apprehension. Hornbæk (2006) reviewed 587 papers and included 180 to review usability measures employed in human-computer interaction research.

The summary synthesis in this thesis, detailed in Chapter 4, adds to their efforts and focuses on extracting design guidelines for multiple-VIR interface based on 19 existing experimental-simulation studies.

3.2 Visual Analytics

The term visual analytics was coined by Thomas and Cook (2005) to represent "the science of analytical reasoning facilitated by interaction visual interfaces" (p. 4). The field of visual analytics researches techniques in analytical reasoning, visual representations and interaction, and data representations and transformation to facilitate exploration and understanding of large data sets, as well as to produce, present and disseminate analysis results.

Given the vast scope of the topic, this section selectively reviews systems and techniques developed for visual data analysis and exploration, many of which influenced and inspired the design of Session Viewer, a visualization tool to support web session log analysis presented in Chapter 7. As web session logs analysis is the application domain of the thesis, this review focuses on visualizations that support exploratory analysis of web session logs in particular, and computer-based log analysis in general. To complete the discussion on session log analysis, Section 3.2.5 briefly explores non-visual approaches to web session log analysis.

3.2.1 Existing visualizations for data exploration

Many visualization systems have been developed to support general data exploration (Keim 2002), including commercial ventures such as Spotfire (spotfire.com), Tableau (tableausoftware.com), and Inxight (inxight.com). While these systems support exploration, their visualizations are typically standard graphical displays with single-level data, where each data attribute is a single value. However, these systems are not tailored for showing multi-level data such as web session logs, where at least one of the data attributes is also a multi-dimensional data object.

A wide range of visualization and interaction techniques have been developed to facilitate visual data exploration and analysis. This section focuses on visual techniques developed for data display, which can be classified as (1) multiple data-form displays showing the same data in multiple representations, (2) multiple data-level displays showing data at multiple levels of organization (or multiple visual information resolutions), and (3) multiple data-dimension display showing multiple attributes for each datum.

1. Multiple data-form display

Analysis often requires viewing the same data in different forms, for example, in linear and logarithmic scales. Roberts (2000) advocated a technique called Multiform to provide different representations, or forms, of the same data to allow analysts to view the data in a more multi-faceted manner. The rationale behind Multiform is based on the fact that different visualization techniques highlight different aspects of the data, and displaying multiple representations of the same data may increase the likelihood of knowledge discovery, as discussed in Section 2.3.2.

A related idea is to provide different views of the same data in the same form. One example is the reorderable matrix, an idea first introduced by Bertin (1981) and implemented in Table Lens (Rao and Card 1994). The rationale is that reordering visual data displays based on data attributes can reveal outliers, correlated features and trends in sample populations.

For data populations, the technique of small multiples provides a means to compare between populations. The idea was first proposed by Bertin as *collections* (Bertin 1981), and further advocated by Tufte (1983). Different forms of small multiples have been arranged in rows and columns to create univariate and bivariate matrices (e.g., MacEachren et al. 2003).

Techniques developed for multiple data-form display can be used to create low-VIR in multiple-VIR interfaces. For example, we adapted the techniques of small multiples and reorderable matrix to create a sessions-level low-VIR overview in Session Viewer (Chapter 7).

2. Multiple data-level display

In cases where the total number of data points far exceeds the display capability, techniques such as zooming, focus + context, and overview + detail have been used as solutions. The central idea behind these techniques is to display data at multiple visual information resolutions such that users can follow the sequence of "overview first, zoom and filter, then details on demand" as suggested by (Shneiderman 1996, p. 337). While the detailed data level may simply be the highest data resolution available, creating the overview requires data filtering, clustering, or visual compression of overview visual objects.

Taking the approach of data filtering, Furnas (1986) proposed the degree-of-interest function based on *a priori* significance of the data objects and their distance relative to the object under inspection, or the object in focus. The amount of display emphasis for data objects is proportional to their degree-of-interest distances. Data objects far away from the focus objects are de-emphasized on the display, for example by being displayed with fewer pixels, or not displayed at all. Jakobsen and Hornbæk (2006) further extended the distance parameter in the degree-of-interest function by separating it into syntactic and semantic distances. If the data is inherently hierarchical, the *a priori* function can reflect the data structure and the overview can display the highest level of the hierarchy allowed by available space. For example, Hornbæk and Frokjær (2001) developed an overview for electronic documents by displaying only the section and subsection headers in their low-VIR overview window.

Another approach of data filtering involves user interaction. Ahlberg and Shneiderman (1994) proposed tight coupling of dynamic query filters to selectively reduce the number of data points on a scatter plot such that the analyst can focus on a subset of the larger data.

In terms of data clustering, van Wijk and van Selow (1999) illustrated the use of pre-processing data to create a more visually manageable overview using a year-long timeseries data set with 52,560 data points. Based on the results of a cluster analysis, their interface only displayed averaged values of the clusters instead of all the data points.

In terms of visual compression of overview objects, Kincaid and Lam (2006) developed a more spatially compact line-graph encoding to display a large collection of line graphs. Instead of encoding both the x- and the y-dimension of line graphs by spatial positions, their visual encoding uses colour to encode the y-dimension, thus reducing the amount of space required to display each line graph. Their visual encoding also enables stacking of line graphs to avoid the inevitable visual cluttering in the overlay alternative.

Considerations in creating overviews constitute one of the design discussions in this thesis. In addition to creating the overview, displaying multiple levels of data involves considerations such as the number of display resolutions required and spatial arrangements of the different visual resolutions. These issues are explored in Chapter 4, our summary synthesis of existing study results, and in Chapters 7 and 8, where we detail Session Viewer's design and deployment.

3. Multiple data-dimension display

Most two-dimensional data plots and graphs can only display two variables at a time. Viewing all data dimensions using these displays thus requires multiple plots. Some visualizations work around this limitation by allowing users to dynamically select data attributes displayed, for example, Ward's (1994) XmdvTool and Stolte and Hanrahan's (2000) Polaris.

Other visualizations strive to simultaneously display as many data dimensions as possible. Parallel Coordinates (Inselberg 1985) takes such an approach and displays data points in n-dimensional space as polylines with vertices on the parallel axes. Dense-pixel displays (Keim and Kriegel 1994) map each data dimension value to a colored pixel and group the pixels belonging to each dimension into adjacent areas. Display arrangements of data provide detailed information on local correlations, dependencies and areas of interest of the data set. Since most dense-pixel displays use one pixel for each data dimension, this approach maximizes display capacity of output devices. The third common approach to display multi-dimensional data is glyphs (Littlefield 1983). A glyph is a graphical object that combines multiple visual features in a single object, with each feature encoding a data dimension. Many examples of glyphs have been developed, such as the whisker plot where each data value is represented by a line segment radiating out from a central point (Ware 2004, p. 184), and the Chernoff Faces, a more whimsical example where different data dimensions are mapped to the sizes and shapes of different facial features (Chernoff 1973).

Another approach to display multi-dimensional data is to reduce the high-dimensional data space to a displayable two, or three, dimensional space by multidimensional scaling. One example is Wise et al.'s (1995) ThemeScape to display text-based documents. Even though the distance relationships between the data points are preserved in multidimensional reduction, how each dimension in the high-dimensional data space relates to the displayed low-dimensional space is not clear, thus making it difficult to understand individual data dimensions. Multidimensional scaling is therefore used to show data clusters, rather than to read off individual data dimension values.

3.2.2 Existing interactions for data exploration

Our discussion of visual analytic related work continues with interaction techniques that were developed for (1) data querying and filtering, and (2) data-view coordination. These works influenced the design of Session Viewer.

1. Data filtering and querying

Analysts often need to isolate interesting subpopulations in large data sets to focus analysis. Dynamic query filtering allows users to progressively refine filter criteria aided by visual feedback of the results (Ahlberg and Shneiderman 1994). For data querying, pattern matching allows analysts to highlight sequences, as implemented in TimeSearcher for time-series data (Hochheiser and Shneiderman 2003).

Session Viewer employed both dynamic querying and pattern matching to filter and highlight sessions (Chapter 7).

2. Data-view coordination

View coordination is a well-known problem found in situations where information is displayed over multiple views and users need to relate visual objects between views. Smooth animation has been proposed as a solution to connect the different temporal views in zooming interfaces to preserve object constancy

(Robertson et al. 1989). Interactive brushing and linking have been proposed to visually relate simultaneously displayed regions or views in interfaces. Brushing is a rapid interaction technique that enables a user to "highlight, select, or delete a subset of elements being graphically displayed by pointing at the elements which a mouse or other suitable input device" (Ward 1994, p. 330). Linking is when "brushing elements in one view affects the same data in all other views" (Ward 1994, p. 330).

North and Shneiderman (1997) constructed a taxonomy of multiple window coordination based on navigation and selection. Coordination has been shown to facilitate the use of overview as a navigational aid when the different resolutions are needed in the task, since it provides "the ability to directly select a target in the overview to immediately locate its details", and the overview thus acts as "an improved scroll bar that facilitates exploration" (North and Shneiderman 2000, p. 736). Coordination between multiple views also allows selection of the same data object in multiple views, which helps users relate the different views and allows concurrent considerations of all the views in their data exploration.

Session Viewer took the multiple view approach to display various levels of session details coordinated by linking (Chapter 7).

3.2.3 Visualizations for web session logs

Visualizations have been used to display aggregate data derived from session logs for presentation or analysis. For example, Pass et al. (2006) surveyed traditional graphical plots to describe and evaluate search services. Despite having analysis goals similar to our tool, these static plots are more suited for presentation than exploration and discovery.

While interactive systems designed for session log analysis exist, they generally focus on website design evaluations based on traffic and user paths, rather than on search usage behaviours. Examples of website traffic visualizations include disk-tree and time-tube visualizations (Chi 2002) and a 3D structure (Wong and Marden 2001). User paths are often displayed as node-link graphs, as in VISVIP (Cugini and Scholtz 1999), WebViz (Pitkow and Bharat 1994), and WebQuilt (Waterson et al. 2002). Lee et al. (2001) took a different approach, displaying web traffic statistics with starfields and user paths with parallel coordinates. Hochheiser and Shneiderman (2001) used a multiple-coordinated visualization to show web visitation data. Other visualizations, such as 3D WebPath (Frecon and Smith 1998) and History tree (Kreuseler et al. 2004), displayed personal web-navigation histories and were designed to help users navigate rather than to analyze their usage behaviours. Generally in these analyses, analysts tend to look for different paths through a fixed set of web pages. For example, usability engineers use VISVIP to study how experimental participants use a single website to accomplish study tasks (Cugini and Scholtz 1999). In contrast, the goals of analyzing web session logs to study search behaviours are different. For example, Kellar et al. (2007) aimed to study four types of information-seeking activities. In that case, their participants could potentially go through an infinite number of pages across a large number of site domains. Visually depicting all user paths as trees or graphs is therefore challenging, and consequently, existing systems do not adequately address the needs of web search analysis.

The one exception is Card et al.'s (2001) Web Behavior Graphs, which show search structures of individual users as modified state diagrams to help researchers locate problem spaces within the web site under analysis, and identify usage behaviour patterns. Despite the richness of the information and insights obtained from the analysis, Card et al.'s (2001) approach is difficult to scale.

3.2.4 Visualizations for computer-based logs

An area highly related to web session log visualization is usability log visualization. Gray et al. (1996) used the coloured Bar Visualization to show usability sessions, with each bar consisting of a stack of colour-coded boxes encoding user activities types. However, the system does not allow comparison between populations or displays at multiple levels of detail. Many visualizations associated with usability log analysis use two- and three-dimensional graphs. Examples include population counts and summary statistics of events (Kay and Thomas 1995) and mouse activities (Guzdial et al. 1994), sequence patterns displayed as state transition diagrams to support Markov-based analysis (Guzdial et al. 1993), and a spreadsheet-like display for sequence alignment (Sanderson and Fisher 1994). These graphs are better suited for presenting analysis results than for data exploration, especially since many of them have to be generated manually.

There are also systems designed for computer log analysis, such as MieLog, which displays textual log entries as colour-coded bars based on their type (Takada and Koike 2002), AuthorLines, which shows e-mail participation and initiation counts based on author (Viégas and Smith 2004), and SnortView, which shows network-based intrusion detection system logs as two-dimensional time diagrams (Koike and Ohno 2004). However, since the analysis goals are typically detecting anomalies rather than identifying and characterizing populations, these systems are not well-suited for exploratory session log analysis.

3.2.5 Non-visual log analysis tools

Commercial statistics packages are frequently used for web log analysis, for example, Microsoft Excel (microsoft.com) and SPSS (spss.com). Analysts also build custom programs, from simple scripts to calculate summary statistics to elaborate algorithms to find usage patterns and population clusters (e.g., Pierrakos et al. 2003). For usability log analysis, systems such as MacSHAPA offers integrated support with a largely text-based interface (Sanderson and Fisher 1994).

3.2.6 Summary of Visual Analytics Related Work

Even though we surveyed a large body of visualization systems that may be used for web session log analysis, we decided to build our own visual analytic tool for three reasons. First, we found that generic visual data exploration systems were not tailored to showing our multi-level web session log data. Also, most visualizations built for web-log analysis were designed to display user paths of a limited number of webpages, and were thus inadequate in supporting analysis of web search behaviours in general that may involve a potentially unlimited number of webpages. Second, even though non-visual approaches to web session log analysis exist and are popular amongst analysts, they tend to focus on confirmatory analysis rather than exploratory data analysis, which is the focus of the thesis. Third, by building a system, we could apply our knowledge and experience gained in our first three evaluations in the design (Chapter 7) and examine our choices in an evaluation of the resultant system (Chapter 8).

Chapter 4

Summary Synthesis: A Study-based Guide to Multiple Visual Information Resolution Interface Designs

In this review, we analyzed 19 existing multiple-VIR interface studies to get a clearer snapshot of the current understanding of multiple-VIR interface use, and how to apply this knowledge in their design. To unify our discussion, we grouped the interfaces into single or multiple-VIR interfaces. For single-VIR interfaces, we looked at the *hiVIR* interface that shows data in detail and at the highest available VIR, for example, the "detail" in overview + detail interfaces. We considered three multiple-VIR interface types in this review: *temporal*, or temporal switching of the different VIRs as in zooming interfaces; *separate*, or displaying the different VIRs simultaneously but in separate windows as in overview + detail interfaces; and *embedded*, or showing the different VIRs in a unified view as in focus + context interfaces. Section 1.1.1 includes a more detailed explanation of our terminology. Since most of the existing multiple-VIR interface studies did not explicitly consider user characteristics such as visual-spatial ability, we did not address this important issue in our discussion.

To better guide design processes, this chapter is structured as a decision tree to create a multiple-VIR visualization, as shown in Figure 4.1. Our decision tree has four major steps:

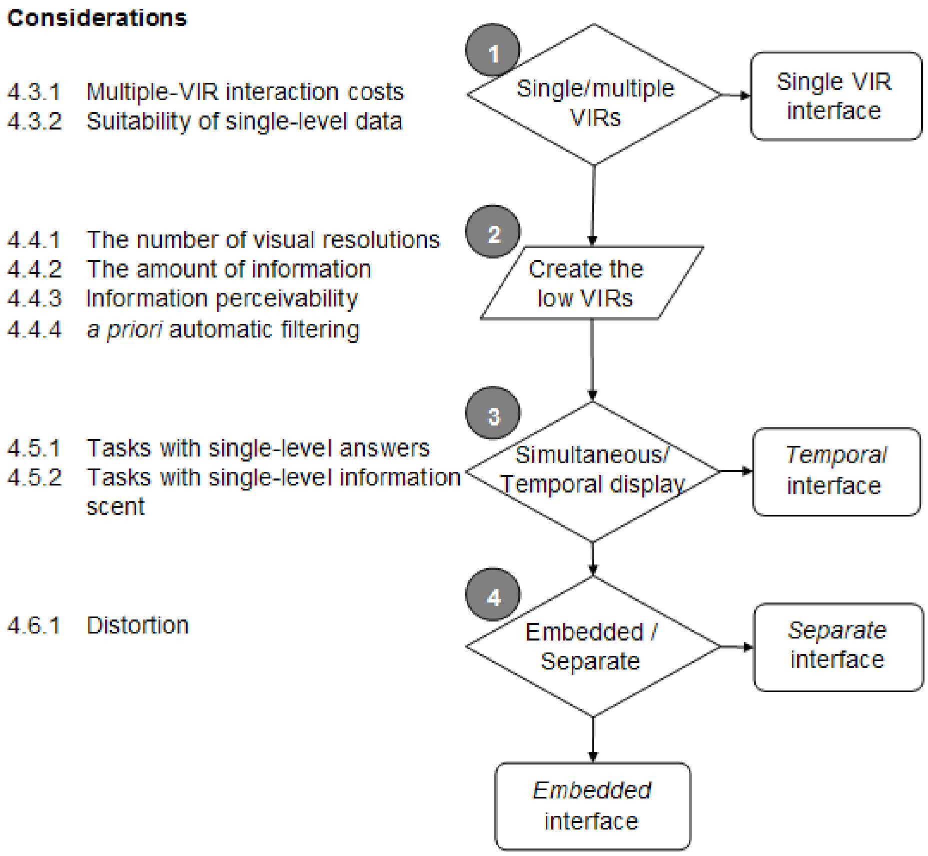


Figure 4.1: Decision tree to create a multiple visual information resolution display. There are four major steps in the decision process, each covered in a section in the chapter: (Decision 1/Section 4.3) Decide if multi-VIR is appropriate for the application; (Decision 2/Section 4.4) Decide on the number of resolutions, amount of data and visual information to be displayed on the low VIRs; (Decision 3/Section 4.5) Decide on the methods to display the multiple VIRs; (Decision 4/Section 4.6) Decide on the spatial layout of the multiple VIRs. Considerations at each decision point are listed with their respective section numbers.

DECISION 1 (Section 4.3): Single- or multiple-VIR interface

The first step in the process is to decide if a multiple-VIR interface is suitable for the task and data at hand. The choice is not obvious as multiple-VIR interfaces typically have more complex and involved interactions than their single-VIR counterparts. Section 4.3.1 discusses interaction costs reported in the reviewed studies. Section 4.3.2 discusses considerations in using multiple VIRs to display single-level data.

DECISION 2 (Section 4.4): Create the low VIRs

If the designer decides to use a multiple-VIR interface for his data, the next step in the design process is to create the low VIRs, which is a challenge with large amounts of data (Keim et al. 2006). In addition to the technical challenges in providing adequate interaction speed and in fitting the data onto the display device, the designer also needs to consider the appropriate levels of visual resolution provided by the interface. Study results indicate that providing too many levels of resolution may be distracting to users, as discussed in Section 4.4.1. Similarly, showing too much data in the low VIRs can also be distracting, as discussed in Section 4.4.2. In many cases, the data may have to be abstracted and visually abbreviated to increase the display capability of the low VIRs. Ellis and Dix (2007) provides a taxonomy of clutter reduction techniques that include sampling, filtering, and clustering. Section 4.4.3 discusses cases where designers had gone too far in their abstractions and study participants could no longer use the visual information in the low VIRs. Instead of abstraction, the designer could choose to selectively display or emphasize a subset of the data in the low VIRs, for example, based on the generalized fisheye degree-of-interest function (Furnas 1986). However, study results suggest that *a priori* automatic filtering may be a double-edged sword, as discussed in Section 4.4.4. Given all these considerations, we complete the discussion by re-examining the roles of low VIRs in Section 4.4.5 to help ground low-VIR design.

DECISION 3 (Section 4.5): Simultaneous or temporal display of the VIRs

Once the VIRs are created, the designer then needs to display them, either simultaneously as in the *embedded* or the *separate* interfaces, or one VIR at a time as in the *temporal* interfaces. Generally, *temporal* displays require view integration over time and can therefore burden short-term memory (Furnas 2006). On the other hand, simultaneous-VIR interfaces have more complex interactions such as view coordination in *separate* displays and the issue of image distortion frequently found in *embedded* displays. Our reviewed study found that for tasks that did not involve multi-level answers, or tasks that did not provide multi-level clues to single-level answers, displaying data with simultaneous multiple-VIR interfaces was not beneficial. Sections 4.5.1 and 4.5.2

consider the case when the study tasks did not require simultaneous display of VIRs, as in single-level answer with single-level clues.

DECISION 4 (Section 4.6): Embedded or separate display of the VIRs

If the choice is simultaneous display, the designer then has to consider the spatial layout of the different VIRs. The choices are to display the VIRs in the same view, as in the *embedded* interfaces, or by showing them in separate views, as in the *separate* interfaces. Both spatial layouts involve tradeoffs: the *embedded* displays frequently involve distortion, as discussed in Section 4.6.1, and the *separate* displays involve view coordination.

For each of these decision points, we summarized current beliefs and assumptions about multiple-VIR interface use, along with relevant study results. We also flagged situations where study results did not clearly support our previous beliefs based on existing literature.

4.1 Methodology

Ideally, we would like to perform a meta-analysis to translate results from different studies to a common metric and statistically explore relationships between study characteristics and study results, as a meta-analysis is more objective, thorough and systematic than qualitative approaches. However, recognizing that the reviewed studies are different in their implementations of the various multiple-VIR techniques, their study tasks and their data, and in some cases, their experimental design and measurements, meta-analysis may only be able to include a very small subset of existing studies. Indeed, only 6 of the 35 studies considered by Chen and Yu (2000) met their criteria for their meta-analysis, and the researchers had a long list of recommendations to visualization evaluators to standardize their study designs. Some of their recommendations, echoed by others (e.g., Plaisant 2004; Ellis and Dix 2006), are still active areas of research. One example is to create standardized task taxonomies for interface evaluations (e.g., Winckler et al. 2004; Valiati et al. 2006).

Perhaps a compromise worth making is to take a more qualitative, albeit less rigorous, approach to extract high-level themes from existing study results. That is the approach we took to extract design guidelines for multiple-VIR interfaces in this study. Instead of comparing between studies, we focused on pairwise-interface comparisons within each study to abstract generalizable usage patterns based on task, data characteristics, and interface differences. We collected an initial set of candidate papers by performing keyword searches on popular search engines (Google and Google Scholar) and large academic databases (ACM and IEEE digital libraries), along with our own collection of study publications accumulated over the years. From this initial set, we further located more study publications based on citations of the initial set. During the course of our synthesis, we continuously added new publications.

Since our goal was to understand multiple-VIR interface use, we differed from most systematic reviews as we did not have specific questions in mind when we began our review. Instead, we took a bottom-up and qualitative approach to find emergent themes from coded individual study findings. We started our process by first coding the papers based on the interfaces studied, as shown in Table 4.1, and the major study findings, as shown in Appendix B. We focused on objective measures of task time and accuracy since these measures were reported in all user studies we sampled. We then gathered study findings for each interface pair (e.g., *hiVIR* and *temporal*) to identify possible underlying reasons that may explain study results. We considered the interfaces (e.g., visual elements, interactions), the data displayed (e.g., level of organization details, levels of data), the tasks (e.g., task natures), and explanations provided by paper authors based on their observations and understanding of their studies. We therefore labeled these interface-pair findings, along with possible explanations, as considerations in design. We organized these considerations into a four-point decision tree, which became the framework of our review (Figure 4.1). For some comparisons, we could not abstract general results from the studies, and we explained our reasons for excluding these interface pairs when appropriate. Since many studies looked at more than two study interfaces, their study results were mentioned in more than one section of the chapter.

Our approach therefore may suffer from reviewer bias in our study inclusion and in our emphasis put on various study results. To ensure objectivity, or at least to convey to our readers the basis of our claims, we listed the studies we considered in each of the design considerations. Given that we collected only 19 papers, we believe explaining each set of study results qualitatively instead of attempting statistical meta-analysis would provide a more encompassing snapshot of our collective knowledge on multiple-VIR use. While we did count the number of studies that produced statistically significant results that support the claim for each design consideration, we did not take the vote-counting approach in systematic reviews (see the terminology section in 1.1.2), as we did not base our findings on the these numbers. Instead, we considered each study publication individually to identify evidence that might either support or refute our findings, taking into consideration possible explanations for study results regardless if they achieved statistical significance. In fact, we put more emphasis in the researchers' insights reported in their publications than on statistical results. Section 4.9 discusses other limitations of our review.

4.2 Summary of Studies

Table 4.1 lists the 19 studies reviewed, along with our encoding of the test interfaces as *hiVIR*, *temporal*, *separate* and *embedded*. Screen captures of study interfaces are available at `http://www.cs.ubc.ca/~hllam/res_ss_interfaces.htm`. In order to provide a reasonably concise review, we excluded studies where study results did not differentiate between study interfaces in terms of performance measures or usage patterns (e.g., Buring et al. 2006).

We considered all the interfaces in the reviewed studies except for the Saraiya et al. (2005) study, since their two "Multiple View" interfaces displayed the same data in separate views at the same VIR, but used a different graphical format. Since our review focused on multiple-VIR interfaces, we considered the issue of multiple presentation forms to be beyond the scope of our review.

Since this study aimed to provide an evidence-based guide to designers in using multiple-VIR interfaces, and not a review paper on existing multiple-VIR study results, we only provided enough study details to illustrate our points so as to maintain readability. For reference, Appendix B.1 provides brief summaries of each study, and Appendix B.2 lists the interfaces, tasks, data, and significant results for each of the reviewed papers.

For each design consideration, we listed studies included for the analysis. Each paper is designated with an identification letter which is used in subsequent tables. Please note that Hornbæk et al.'s online document study was reported in two papers: Hornbæk and Frokjær (2001) and Hornbæk et al. (2003). For completeness, we also included our overview-use study in this review. Chapter 6 details the study, and references to the study in this review is denoted as (Lam et al. 2007) for consistency.

ID	Authors	Paper Title	Sgl	Multiple		
			H	T	E	S
A	Baudisch et al. (2002)	Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming		x	x	x
B	Baudisch et al. (2004)	Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view	x		x	x
C	Bederson et al. (2004)	DateLens: a fisheye calendar interface for PDAs		x	x	
D	Gutwin and Skopik (2003)	Fisheye views are good for large steering tasks			x	x
E	Hornbæk and Frokjær (2001)	Reading of electronic documents: the usability of linear, fisheye and overview + detail interfaces	x		x	x
	Hornbæk et al. (2003)	Reading patterns and usability in visualization of electronic documents	x		x	x
F	Hornbæk et al. (2002)	Navigation patterns and usability of zoomable user interfaces with and without an overview		x		x
G	Hornbæk and Hertzum (2007)	Untangling the usability of Fisheye menus		x	x	x
H	Jakobsen and Hornbæk (2006)	Evaluating a fisheye view of source code	x		x	
I	Lam and Baudisch (2005)	Summary Thumbnails: readable overviews for small screen web browsers	x	x		
J	Lam et al. (2007)	Overview use in multiple visual information resolution interfaces	x		x	x
K	Nekrasovski et al. (2006)	An evaluation of pan and zoom and rubber sheet navigation		x	x	x
L	North and Shneiderman (2000)	Snap-together visualization: can users construct and operate coordinated visualizations?	x			x

Sgl = Single; H = HiVIR; T = Temporal; E = Embedded; S = Separate.

continued on next page...

...continued from previous page

ID	Authors	Paper Title	Sgl	Multiple		
			H	T	E	S
M	Pirolli et al. (2003)	The effects of information scent on visual search in the hyperbolic tree browser		x	x	
N	Plaisant et al. (2002)	SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation		x	x	
O	Plumlee and Ware (2006)	Zooming, multiple windows, and visual working memory		x		x
P	Saraiya et al. (2005)	Visualization of graphs with associated timeseries data	x	x		
Q	Schafer and Bowman (2003)	A comparison of traditional and fisheye radar view techniques for spatial collaboration			x	x
R	Schaffer et al. (1996)	Navigating hierarchically clustered networks through fisheye and full-zoom methods		x	x	
S	Shi et al. (2005)	An evaluation of content browsing techniques for hierarchical space-filling visualizations		x	x	

Table 4.1: Multiple-VIR studies reviewed. An 'X' in the cell denotes the study included an interface of the corresponding type: Sgl = Single; H = HiVIR; T = Temporal; E = Embedded; S = Separate. Note that Lam et al. (2007) is the third study in this thesis, reported in Chapter 6.

4.3 Decision 1: Single or Multiple-VIR Interface?

The first step in our design decision tree is to decide if a multiple-VIR interface is appropriate for the task and data at hand. To isolate situations where the additional low VIRs were found to be useful, we looked at studies that compared the single-VIR *hiVIR* interfaces to the three multiple-VIR interfaces: *temporal*, *embedded*, and *separate*.

It is generally believed that interfaces should provide more than one VIR (e.g., Card et al. 1999, p. 307). However, for users, having the extra VIRs means more complex and difficult VIR coordination and integration, which may be time consuming and require added mental and motor efforts. The topic of interaction costs in multiple-VIR interface is further discussed in Section 4.3.1.

Interaction costs may be justified if lower VIRs provided in addition to the basic *hiVIR* interfaces are useful to users. In general, usefulness of the additional lower VIRs hinges upon the levels of data structure required by the task. In other words, single-level data may not be suited for multiple-VIR display, as discussed in Section 4.3.2.

4.3.1 Consideration 1: multiple-VIR interface interaction costs should be considered

Interaction complexity can be difficult to measure and isolate. Commonly used objective measurements such as performance time and accuracy are aggregate measures and cannot be used to identify specific interaction costs incurred in interface use. In ten of our reviewed papers, researchers recorded usage patterns, participant strategies, and interface choice that revealed interaction costs (Table 4.2).

1. Interaction costs from usage patterns

Source	**Papers**
Usage patterns (Eye-tracking records)	G. Hornbæk and Hertzum (2007) M. Pirolli et al. (2003)
Usage patterns (Navigation-action logs)	E. Hornbæk et al. (2003) F. Hornbæk et al. (2002) H. Jakobsen and Hornbæk (2006)
Participant strategies	A. Baudisch et al. (2002) J. Lam et al. (2007)
Interface choice	E. Hornbæk and Frokjær (2001); Hornbæk et al. (2003) F. Hornbæk et al. (2002) J. Lam et al. (2007)

Table 4.2: Ten papers that reported interface interactions. Five reported usage patterns obtained either from eye-tracking records or navigation-action logs; two reported participant strategies; and two reported interface choice.

As shown in Table 4.2, 5 of the 19 studies reported usage patterns constructed based on eye-tracking records or navigation action logs. Two of the studies reported usability problems with their multiple-VIR interfaces (Hornbæk et al. 2002; Hornbæk and Hertzum 2007).

Hornbæk et al.'s (2002) study on map navigation reported that participants who actively used the low-VIR view switched between the low- and the high-VIRs more frequently, which resulted in longer task completion time. The researchers reported that using the additional low-VIR view may require mental effort and time moving the mouse, thus adding complexity in the interaction (p. 382). Indeed, navigation patterns showed that only 55% of the 320 tasks were solved with active use of the low-VIR view in their multiple-VIR interfaces (p. 380).

Hornbæk and Hertzum's (2007) study on fisheye menus reported large navigation costs in their *separate* and *embedded* interfaces, all implemented a focus-locking interaction mechanism (Bederson 2000). Even though these interfaces succeeded in facilitating quick, coarse navigation to the target, participants had difficulty getting to the final target since the menu items moved with the mouse. Based on eye-tracking data, the researchers reported that participants made longer fixations and longer scan paths with their *separate* and *embedded* interfaces than with their *temporal* interface, suggesting increased mental activity and visual search.

2. Interaction costs from participant strategies

As shown in Table 4.2, 2 of the 19 studies reported participant strategies in interface use.

In Baudisch et al.'s (2002) study on map path-finding and verification, some participants avoided continuously zooming in and out using the *temporal* interface by memorizing all the locations required in the task and answered the questions in a planned order. As a result, they could stay at a specific magnification without zooming back to the low-VIR view, thus effectively using the *temporal* interface as a *hiVIR* interface.

In Lam et al. (2007), participants developed a strategy to use the seemingly suboptimal *hiVIR* interface in a visual comparison task. The data consisted of a collection of line graphs that were identical except shifted by various amounts in the x-dimension. The task involved matching a line graph with the same amount of horizontal shift. Some participants took advantage of spatial arrangement of the *separate* interface by selecting candidate line graphs from the low-VIR view and displaying them in high VIR for side-by-side comparison. The majority of participants, however, developed a strategy to enable the use of the high-VIR view alone. Taking advantage of the mouse wheel and the tool-tips which displayed horizontal and vertical values of the line graph point under the cursor, participants scrolled vertically up and down with the cursor fixed horizontally at the point where the target peaked. As a result, they eliminated the need to visually compare line graphs. Instead, they tried to find another peak at the same x point numerically by reading off the tool-tips and avoided the need to interact with multiple VIRs.

3. Interaction costs from participants' interface choices

Another indicator of interaction costs is participants' active choice to use only one VIR in a multiple-VIR interface to avoid coordinating between the multiple VIRs. As shown in Table 4.2, participants could explicitly convert a multiple-VIR into a single-VIR interface in 2 of the 19 studies, and in Hornbæk et al.'s (2002) study on map navigation, the researcher recorded active pane use.

In Hornbæk's study on reading electronic documents, participants could expand all the document sections at once by selecting the pop-up menu item "expand all" in the *embedded* interface (Hornbæk and Frokjær 2001; Hornbæk et al. 2003). Six out of 20 participants chose to do this in one or more of the tasks. On average, they expanded 90% of the sections, thus effectively using the *embedded* interface as a *hiVIR* interface.

In Hornbæk et al.'s (2002) study on map navigation, 45% of participants did *not* actively use the low-VIR view in the *separate* interface, even though 80% of participants reported preference for having the extra low-VIR view.

In Lam et al.'s (2007) study on visual search and comparison of line graphs, participants could expand all initially compressed graphs in their *embedded* or their *separate* interface by a key press, thus effectively turning the multiple-VIR interface into a high-VIR interface. Their participants actively switched to the *hiVIR* interface in 58% of the trials.

We suspect this desire to use only a single VIR when given a multiple-VIR interface is more prevalent than reported. In many cases, participants were not provided with a simple mechanism to convert from the multiple-VIR interface to its single-VIR counterparts, while in other cases, sole use of one window in the *separate* interface could not be discerned without detailed interaction recordings such as eye-tracking records. Using multiple-VIR interfaces as single-VIR interfaces may explain some studies' inability to distinguish *hiVIR* interface and their multiple-VIR counterparts, for example in Lam et al. (2007), our overview-use study detailed in Chapter 6.

4.3.2 Consideration 2: single-level task-relevant data may not be suited for multiple-VIR displays

Multiple-VIR Effect	**Paper with single-VIR data**
No benefits	B. Baudisch et al. (2004) J. Lam et al. (2007)
Adverse effects	F. Hornbæk et al. (2002)
Mixed effects	E. Hornbæk et al. (2003)
Excluded	I. Lam and Baudisch (2005)

Table 4.3: Five papers that had single-level data and included a single-VIR interface for analysis comparison. In these cases, most multiple-VIR interfaces supported the same or worse performances than their single high-VIR counterparts.

The number of VIRs provided by the interface should reflect the levels of organization in the data as required by the task. Otherwise, users may need to pay the cost of coordinating between different VIRs without the benefits of rich information at every VIR. Among the seven studies reviewed that included a single-VIR interface, five of them used at least one set of single-level data (Table 4.3). Two studies failed to show performance benefits of multiple-VIR interface for single-level data in cases where the tasks required detailed information not provided by the low-VIR display alone. Hornbæk et al.'s (2002) study showed adverse effects in using multiple-VIR interfaces for single-level data. Hornbæk et al.'s (2003) study on online documents showed mixed results, as task nature affected the levels of data required, and consequently, interface use. We excluded Lam and Baudisch's (2005) study in this discussion as their *hiVIR* interface had almost nine times the number of pixels than their multiple-VIR interfaces, making direct comparisons difficult.

Baudisch et al.'s (2004) study on information searches showed a lack of benefit in using multiple-VIR interface for single-level data when the task could not be performed based on information showed on the low VIR alone. Their study interfaces displayed web documents with guaranteed legible keywords which constituted their low-VIR displays. When the task only required reading the keywords, as in their *Outdated* task, their multiple-VIR interfaces outperformed their high-VIR browser, probably because the low-VIR displays concentrated task-relevant information in smaller display spaces. In contrast, when the task required reading surrounding text which may be too small to be legible, as in the *Analysis* task, having the extra low-VIR display did not result in performance benefits for the single-level document data.

The situation is similar in Lam et al.'s (2007) study on visual-target search on a line-graph collection. Their multiple-VIR interfaces only showed performance benefits over their *hiVIR* interface when the visual targets could be directly identified on the low-VIR display, for example, in their *Max* task. Otherwise, having the extra low VIR did not seem to enhance participant performance since their data was essentially single-leveled.

Hornbæk et al.'s (2002) study on map navigation illustrates the adverse effects of displaying single-level data using a multiple-VIR interface. Despite having a similar number of objects, area occupied by the geographical state object, and information density on the maps, there were surprisingly large differences in usability and navigation patterns between the two study-map trials. The Washington-map trials had better performance time, accuracy and subjective satisfaction than the Montana-map trials. The researchers explained these differences by differences in map content and the number of organization levels: the Washington map had three levels of county, city, and landmark, while the Montana map was single-leveled with weak navigation cues at low zoom levels. As a result, unlike the multiple-level Washington map, the single-level Montana map data was not suitable for the multiple-VIR *temporal* interface had produced poorer performance results.

Hornbæk *et al.*'s online document study showed mixed results, which illustrated how task nature could affect the levels of data required, and how that difference could affect interface effectiveness (Hornbæk and Frokjær 2001; Hornbæk et al. 2003). In their question-answering task, participants were slower without being more accurate in their answers if they were given an additional low-VIR view. Based on reading patterns, Hornbæk and Frokjær suggested that the slower reading times were due to the attention-grabbing low-VIR view in the *separate* interface, which led participants to further explore the documents perhaps unnecessarily. In contrast, in the essay-writing task where participants were required to summarize the documents, having the extra low-VIR overview displaying data structure as section and subsection headers resulted in better quality essays without any time penalty when compared to the *hiVIR* interface. In other words, when the task required single-level answers, as in the question-answering task, having an extra low-VIR display had a time cost; when the task required multiple-level answers, as in the essay-writing task, the low-VIR display produced higher quality results.

4.3.3 Summary of considerations in choosing between a single or a multiple-VIR interface

In general, the amount of interaction efforts required to coordinate the multiple VIRs is non-trivial and should be considered. We found that when adding VIRs to the high-VIR display did not add task-relevant information, as in the case of using multiple VIRs to display single-level data, costs incurred in VIR coordination were typically not justified.

4.4 Decision 2: How to Create the Low VIRs?

Once the designer decided on taking the multiple-VIR approach, the next step in the process is to create the low-VIR display. Creating the low-VIR display in a multiple-VIR interface is a non-trivial task, especially when the amount of data involved is large. Study results suggest a delicate balance between displaying enough visual information for the low-VIR display to be useful and showing irrelevant resolution or information that becomes distracting. In Section 4.4.1, we discuss the adverse effect of displaying more levels of VIRs than supported by the data and required by the task. Section 4.4.2 discusses the related topic of displaying too much information on the low-VIR display.

Given the space constraints, designers usually need to find less space-intensive visual encodings for the data or reduce the number of data displayed on the low-VIR display. Section 4.4.3 discusses cases where the researchers had gone too far in their visual-encoding abstraction as their study participants could no longer use the visual information on the low VIRs. Section 4.4.4 looks at the trade-offs in using *a priori* automatic filtering to selectively show data on low-VIR displays.

Given all these considerations, we round up the discussion in Section 4.4.5 by re-examining the roles of low VIRs to help ground low-VIR designs. Study results suggest a more limited set of low-VIR roles than proposed in literature. While we found that study results supported the use of low VIR as navigational shortcuts to move within the data and to provide overall data structure, we failed to find supports to the common beliefs of using low VIR to aid orientation or to provide meaning for comparative interpretation of an individual data value.

4.4.1 Consideration 1: having too many visual resolutions may hinder performance

In general, the number of visual resolutions supported by the interface should reflect the levels of organization in the data. Otherwise, users may need to pay the cost of coordinating between the different VIRs without the benefit of rich information at each level. In cases where the extra VIRs were not useful for the task at hand, the irrelevant information could be distracting. These extra VIRs may at best be ignored, and at worst, may harm task performance.

Of the 19 studies reviewed, four looked at compound multiple-VIR interfaces when an additional low-VIR view was added to an already multiple-VIR interface (Table 4.4).

Effect	**Paper with compound multiple-VIR interface**	Low-VIR view added to
No benefits	A. Baudisch et al. (2002)	*temporal* zoom plus pan (z+p) display to create their overview plus detail (o+d) interface
	K. Nekrasovski et al. (2006)	*temporal* Pan&Zoom and their *embedded* Rubber Sheet Navigation interfaces
Adverse effects	F. Hornbæk et al. (2002)	*temporal* zoomable interface
Excluded	G. Hornbæk and Hertzum (2007)	*embedded* fisheye menu

Table 4.4: Four papers that had at least one compound multiple-VIR interface, created by adding an additional low-VIR view was to a multiple-VIR interface.

Since Hornbæk and Hertzum's (2007) study did not include an interface that was only *embedded* without the low-VIR overview, we could not discern the effects of having an additional low-VIR view and thus excluded it from this discussion. For the other three studies, perhaps because the multiple-VIR interfaces already displayed all the meaningful and task-relevant visual information levels supported by the data, having the additional low-VIR view did not enhance or even degrade participant performance.

Two studies showed a lack of benefit in providing additional low-VIR views (Table 4.4). Participants in Baudisch et al.'s (2002) study obtained similar performances using the overview plus detail (o+d) interface and their zoom plus pan (z+p) interface. The researchers reported that participants kept the *temporal* view zoomed to 100% magnification for tracing, thus effectively reduced the *temporal* component of the interface to a single-VIR display, and used the compound multiple-VIR interface as a *separate* interface (low-VIR + *temporal* used as high-VIR).

In Nekrasovski et al.'s (2006) study on large trees and visual comparison tasks, the overall tree view in the low-VIR overview provided task-relevant location cues. However, the information was not unique and necessary as the high-VIR view also provided a similar visual cue. As a result, the study failed to show performance benefits in having an extra low-VIR view in their interfaces even though participants reported reduced physical demand.

Hornbæk et al.'s (2002) study on map navigation suggested performance hinderance when an interface provided irrelevant levels of resolutions. One of their study interfaces was a *temporal* interface with an added low-VIR overview. They reported that participants who actively used the low-VIR overview had higher performance time, possibly because of the mental and motor efforts required in integrating the low- and high-VIR windows. Such costs were not compensated by richer information displays as the *temporal* interface already contained all the task-relevant visual resolutions and may have reduced, or even eliminated, the need for a separate overview (p. 381).

In some cases, study results indirectly suggested adverse effects on performance when the interfaces provided irrelevant VIRs. For example, in Plumlee and Ware's (2006) study that required matching three-dimensional object clusters, their *temporal* interface had many magnification levels that neither helped participants to locate candidate objects, nor were detailed enough for visual matching. Given that participants needed to memorize cluster objects between temporal view switching with the *temporal* interface, the extra zooming levels may have rendered the tasks harder. This extra cognitive load may explain the relatively small number of items participants could handle before the opponent *separate* interface supported better performance, when compared to results obtained in Saraiya et al. (2005).

Similarly, in Baudisch et al.'s (2002) study on static visual path-finding tasks and dynamic obstacle-avoidance tasks, their *temporal* interface and their *separate* interface seemed to have included more VIRs than their *embedded* interface. While the special setup in their *embedded* interface undoubtedly contributed to the superior participant performances, we did wonder if the extra VIRs may have distracted participants in the other two interface trials.

4.4.2 Consideration 2: having too much information on the low-VIR display may hinder performance

While it may be tempting to provide more rather than less information on the low-VIR display, study results suggest that the extra information may harm task performance. None of the 19 reviewed studies included low-VIR item density as a factor. However, we obtained indirect evidence by comparing between multiple-VIR interfaces that display different amounts of visual information in their low-VIR displays, and by comparing between low- and high-VIR displays for visual search tasks that only required the low-VIR displays.

As shown in Table 4.5, 15 of the studies included at least two multiple-VIR interfaces. Of the 15 studies, 11 showed similar amounts of information on the low-VIR displays and could not be used to understand the effects of task-irrelevant information. We also excluded Hornbæk's electronic document study since their low-VIR displays showed different kinds, rather than different amounts, of information (Hornbæk and Frokjær 2001; Hornbæk et al. 2003). We excluded Plaisant et al.'s (2002) study since it was unclear from the paper the number of items initially shown in their *embedded* SpaceTree interface.

Amount of Low-VIR Info	**Papers**
Similar (excluded)	A. Baudisch et al. (2002) B. Baudisch et al. (2004) C. Bederson et al. (2004) D. Gutwin and Skopik (2003) F. Hornbæk et al. (2002) J. Lam et al. (2007) K. Nekrasovski et al. (2006) O. Plumlee and Ware (2006) Q. Schafer and Bowman (2003) R. Schaffer et al. (1996) S. Shi et al. (2005)
Different	G. Hornbæk and Hertzum (2007) M. Pirolli et al. (2003)
Excluded	E. Hornbæk and Frokjær (2001); Hornbæk et al. (2003) N. Plaisant et al. (2002)

Table 4.5: Fifteen papers that included at least two multiple-VIR interfaces. We compared the amount of information displayed on the different low-VIRs to understand the effects of task-irrelevant information.

Our discussion here therefore focuses on the two studies that displayed similar kinds of information, but at different amounts, on their low-VIR displays:

1. Pirolli et al.'s (2003) study compared the *separate* file browser with the *embedded* hyperbolic tree browser. While the paper did not explicitly compare display capacities of the two low-VIRs, we estimated display volume based on paper figures. The low-VIR view of the *separate* file browser displayed about 30 items. In contrast, the capacity of the low-VIR region of the *embedded* hyperbolic tree browser was at least two orders of magnitude larger.

2. Hornbæk and Hertzum's (2007) study compared the *temporal* cascading menu to two *embedded* menu designs based on the Fisheye menu (Bederson 2000). While the lowest VIR of their *temporal* cascading menu only showed a list of alphabets, their *embedded* fisheye menus showed all menu items in font sizes based on relative distances from the focus.

In both of these cases, the researchers advised against putting too much visual information on the display. Pirolli et al. (2003) argued against the assumption of " 'squeezing' more information into the display 'squeezes' more information into the mind" (p. 51) since visual attention and visual search interact in complex ways. In fact, their study showed detrimental effects of display crowding. Pirolli et al. (2003) quantified information relevance as information scent. For their tree data set, they developed an Accuracy of Scent score, which was related to "(a) the ability of users to discriminate the information scent associated with different subtrees to explore and (b) the correctness of those choices with respect to the task." (p. 31). Their study found that their *embedded* hyperbolic tree browser interface led to slower performance times when compared to their *temporal* file browser under low information scent, possibly because their *embedded* interface displayed irrelevant information that was distracting.

Hornbæk and Hertzum (2007) came to a similar conclusion in their study on displaying menus with large numbers of items: "designers of fisheye and focus + context interfaces should consider giving up the widespread idea that the context region must show the entire information space" (p. 28). We excluded their *temporal* cascading menu results in this discussion since their *separate* and their *embedded* interfaces had severe usability problems, and were therefore not comparable to the *temporal* results. We therefore focused on the two *embedded* interfaces and compared between them instead. Their Multifocus menu displayed larger numbers of readable menu items than the Fisheye menu, but had lower coverage of the data set. Eye-tracking results indicated that participants made more use of context and transition regions in the Multifocus menu than with the Fisheye menu. The researchers thus suggested dispensing with the unreadable, and therefore inaccessible, transition regions in the Fisheye menu (p. 26).

Task Answer Location	**Papers**
low VIR	B. Baudisch et al. (2004) I. Lam and Baudisch (2005) J. Lam et al. (2007) L. North and Shneiderman (2000) P. Saraiya et al. (2005)
Both high VIR and low VIR	E. Hornbæk and Frokjær (2001); Hornbæk et al. (2003) H. Jakobsen and Hornbæk (2006)

Table 4.6: Seven papers that included a *hiVIR* and a multiple-VIR interface, classified by the locations from which participants could find answers to the tasks.

This situation is analogous to tasks where answers are apparent from the low-VIR display, and extra information in the *hiVIR* interface is therefore irrelevant. As shown in Table 4.6, seven of the reviewed studies included a *hiVIR* and a multiple-VIR interface. Five of them included tasks that could be answered using the low-VIR displays alone. We therefore attempted to understand effects of displaying unnecessary information by comparing participant performances between their multiple-VIR interfaces, where participants were likely to have consulted mainly the low-VIR displays, and their *hiVIR* interfaces, where participants needed to sieve through irrelevant information to locate task answers. However, except in the case of Lam et al. (2007) and North and Shneiderman (2000) where a *loVIR* interface was also studied, our findings were speculations as we could not be certain that participants focused on the low-VIR displays in the multiple-VIR interfaces.

In Baudisch et al.'s (2004) study on information searches on web documents, their *Outdated* task required participants to check if the web documents contained all four semantically highlighted keywords. In other words, the detailed readable content of the web documents displayed in their *hiVIR* interface was irrelevant to the *Outdated* task. Since their *separate* and their *embedded* interfaces concentrated these task-relevant semantic highlights in their low-VIR displays, the two multiple-VIR interfaces outperformed their *hiVIR* interface for this task.

In Lam and Baudisch's (2005) study on information search on webpages, their PDA-sized *temporal* interfaces, when rendered on desktop, supported equal performance as their desktop counterpart, even though the *hiVIR* interface had nine times more display space showing completely readable information. The researchers suggested that the extra information on the desktop display may have distracted participants and caused unnecessary searching and reading, which may have resulted in lack of performance benefits of having a larger display.

In Lam et al.'s (2007) study on visual target search in a large line-graph collection, one of the tasks involved finding the highest point in the data. The *loVIR* interface alone was adequate for the task, and not surprisingly, interfaces that included a low-VIR display were found to support better performance than their *hiVIR* interface. Observation data suggested that about half of the participants did not use the high-VIR display in the multiple-VIR interfaces for this task.

In North and Shneiderman's (2000) study on visual information search, interfaces that were equipped with a low-VIR view (i.e., their *loVIR* and *separate* interfaces) were found to be superior to the *hiVIR* interface for tasks that could be answered based on information on these low-VIR views alone.

Similarly Saraiya et al. (2005) found that their low-VIR, or single attribute, display was most helpful to analyze graphs at a particular time point, as "multiple attributes can get cluttered due to the amount of information being visualized simultaneously" (p. 231).

In short, instead of using physical item density as a measurement of space-use efficiency, a perhaps more useful consideration is the density of useful information on the display, which is arguably task or even subtask specific.

4.4.3 Consideration 3: displaying information is not sufficient; information has to be perceivable

The mere presence of information on the screen is not sufficient; the information needs to be perceivable to be usable. Text on the low-VIR display may need to be readable to be useful. As shown in Table 4.7, seven of the 19 studies reviewed looked at text data. Four studies included unreadable text in their interfaces, while two had only readable text. We excluded Bederson et al.'s (2004) study as both of their interfaces, the *embedded* DateLens and the *temporal* Pocket PC Calendar, used symbols to replace text in case of inadequate display area.

Text Readability	**Papers**
Some unreadable	B. Baudisch et al. (2004) G. Hornbæk and Hertzum (2007) H. Jakobsen and Hornbæk (2006) I. Lam and Baudisch (2005)
Only readable text	E. Hornbæk and Frokjær (2001); Hornbæk et al. (2003) L. North and Shneiderman (2000)
Excluded	C. Bederson et al. (2004)

Table 4.7: Seven papers that looked at text data, classified by the readability of the included text.

Study results showed that unreadable text displayed on low VIRs were ineffective shortcuts to high-VIR details, as single *hiVIR* displays resulted in similar participant performance despite displaying the information in a larger screen area and thus, having a larger search space.

In Baudisch et al.'s (2004) study on information searches on web documents, both of their multiple-VIR interfaces showed unreadable text except for a few keywords. When the task required reading neighborhood texts to these readable keywords, as in their *Analysis* task, the multiple-VIR interfaces failed to demonstrate performance benefits over the traditional *hiVIR* browser.

In Hornbæk and Hertzum's (2007) study on displaying large numbers of menu items, their *embedded* Fisheye menu displayed unreadable items at the extreme ends in the low-VIR regions. Eye-tracking results indicated that participants made very little use of low-VIR regions, thus suggesting their ineffectiveness (p. 26).

Jakobsen and Hornbæk's (2006) study looked at displaying program code using an *embedded* fisheye interface which displayed unreadable text in the low-VIR regions. The *embedded* interface showed time cost over the *hiVIR* interface in a task that involved counting conditional and loop statements, as participants spent more time in the *embedded* interface to find closing braces of a loop control structures that were unreadable in the low-VIR regions. The researchers thus suggested that interfaces should display readable text to allow direct use of the low-VIR view information (p. 385).

Lam and Baudisch's (2005) study reported similar findings. Their *temporal*

Thumbnail interface had unreadable low-VIR text, but their *temporal* Summary Thumbnail contained only readable low-VIR text. They found that participants using the Thumbnail interface had 2.5 times more zooming events, and when zoomed in, horizontally scrolled almost 4 times more, suggesting the ineffectiveness of the unreadable low-VIR text.

For graphical visual signals, two studies reported effects of showing insufficient details on the low-VIR display (Hornbæk et al. 2002; Lam et al. 2007). In Hornbæk et al.'s (2002) study on map navigation, the geographic map information provided by the low-VIR overviews may not have been sufficiently detailed for the study tasks, for example, to find a neighboring location given a starting point, to compare the location or size of two map objects, or to find two largest map object given a geographic boundary. For the Washington-map trials, having an extra low-VIR overview had time and recall accuracy costs, suggesting the burden of "switching between the detail and the overview window required mental effort and time moving the mouse" (p. 382). Indeed, "tasks solved with active use of the overview were solved 20% slower than tasks where the overview window was not actively used" (p. 380), possibly due to the insufficient information on the low-VIR overview that led to the large number of transitions between the overview and the detail window. Despite 80% indicated subjective preference for having the extra view, only 55% of participants actively used the low-VIR view.

Lam et al. (2007) qualified perceptual requirements for their low-VIR display as visual complexity and visual span. The study looked at displaying a large collection of line graphs for visual search and visual compare tasks, and found that in order for the low-VIR view to be usable, the signal had to be visually simple and limited to a small horizontal area. For example, in the task that required finding the highest peak in the data collection, the visual signals on the low-VIR displays were simple narrow peaks and could easily be found. In contrast, three-peak signals in their *Shape* task were complex and were less discernable in the low-VIR views. As a result, participants resorted to the the high-VIR views for these three-peak signals.

In short, designers need to provide enough details for visual objects on low-VIR displays to be usable. For text, the display objects should be readable if the tasks required understanding text content. For graphical objects, the criteria are less clearly defined.

4.4.4 Consideration 4: *a priori* automatic filtering may be a double-edged sword

Papers	**Filtering Effect(s)**
B. Baudisch et al. (2004)	pos
H. Jakobsen and Hornbæk (2006)	pos and neg
E. Hornbæk and Frokjær (2001); Hornbæk et al. (2003)	neg

Table 4.8: Four papers implemented *a priori* automatic filtering. pos = positive effects observed; neg = negative effects observed.

Designers often can only display a subset of the data on the low-VIR displays. One selection approach is based on Furnas's (1986) degree-of-interest function using *a priori* knowledge of data relevance with respect to the focus datum. Jakobsen and Hornbæk (2006) further differentiated the distance term in the function into semantic and syntactic distances to implement an *embedded* interface for source code. As seen in Table 4.8, of the three studies that implemented *a priori* automatic filtering, two suggested that automatic filtering could enhance task performance as the low-VIR displays concentrated useful information and reduced distractors. However, in two studies, some participants were confused by the selective filtering and became disoriented.

Instead of seeing filtering as a workaround to the display-size challenge and as a liability, there is evidence to suggest that filtering in itself can enhance task performance. When filtering selects task-relevant information for the low-VIR display, such intelligence avoids tedious manual searching and navigation in the high-VIR view, and possibly also avoids distractions by irrelevant information.

In Baudisch et al.'s (2004) study on information searches on webpages, their multiple-VIR interfaces semantically highlighted and preserved readability of keywords relevant to the tasks. These keywords were concentrated in smaller display spaces by reducing font sizes of surrounding texts. Such interfaces resulted in better participant performances as long as they still provided task-required layout information. For example, participants were faster when using either of their multiple-VIR interfaces for the *Outdated* task, and when using their web-column preserving *embedded* interface for the *Product-choice* task.

In Jakobsen and Hornbæk's (2006) study on displaying program source code, automatic and semantically selected readable context in their *embedded* interface avoided the need to manually search for function declarations in the entire source code. This advantage manifested in faster performance times in tasks where participants were required to search for information contained in the function declarations throughout the entire source code.

However, automatic filtering may be a double-edged sword, as filtering may result in disorientation and distrust of the automatic selection algorithm. In Hornbæk's study on reading electronic documents (Hornbæk and Frokjær 2001; Hornbæk et al. 2003), their *embedded* interface preserved readability only for the most important part of the document, with content importance determined by the interface *a priori*. Participants expressed distrust, both in their satisfaction feedback where they rated the *embedded* interface as confusing, and in their comments indicating that they "did not like to depend on an algorithm to determine which parts of the documents should be readable" (p. 142).

This problem may be worse with semantic filtering, where object visibility depends on the semantic relatedness of the object to the focus datum, rather than the geometric distance between screen displays. Selection of displayable context based on syntactic distance between the data point and the focus is arguably easier to predict than semantic selection. Consequently, it may be easier for users to understand and trust filtering algorithms based on syntactic distance only. Also, since context information is updated when the focal point changes, it may be more confusing to navigate with semantic-context updates, as pointer navigation is conceptually geometric rather than semantic. In Jakobsen and Hornbæk's (2006) study on program source code visualization, low-VIR regions replaced scrolling in the *hiVIR* interface and only displayed semantically-relevant source code based on focus. Participants were confused about the semantic algorithm that caused program lines to be shown and highlighted in the context area (p. 385).

Another problem of automatic filtering is that the selection may affect the amount of time users spent on different parts of the data. In Hornbæk's study on reading electronic documents (Hornbæk and Frokjær 2001; Hornbæk et al. 2003), the researchers found that participants spent approximately 30% less time on the initially collapsed sections displayed on their *embedded* interface than when displayed in full on the other interfaces.

In short, while *a priori* filtering may concentrate task-relevant information on low-VIR displays, selective filtering may incur user distrust and confusion, and may even affect how users explore the displayed data.

4.4.5 Consideration 5: the roles of the low-VIR displays may be more limited than proposed in literature

While the high-VIR display enables users to perform detail work, low-VIR roles are harder to verify. We therefore looked at four proposed uses of the low-VIR display based on published literature, and found that study results support only proposed claims for *separate* interfaces: low-VIR provides navigation shortcuts and overall data structure. We were unable to find strong support for low-VIR in *embedded* interfaces to aid orientation or to provide meaning for data comparison.

Supported: low-VIR view provides navigation shortcuts

Information showed in the low-VIR region or view can facilitate navigation by providing long-distance links, thus "decreasing the traversal diameter of the structure" in navigation (Furnas 2006). Coordinations between the low- and the high-VIR views enable users to directly select targets on low VIR displays for detail exploration. For example, North and Shneiderman (2000) found that low-VIR view of a list of geographic states acted as hyperlinks for the high-VIR detail census data.

Another way that low VIR assists navigation is by providing a map of available paths (Card et al. 1999). An example is the low-VIR overview in the *separate* interface in Hornbæk et al.'s (2003) online document study that showed section and subsection headers. For graphical displays, Baudisch et al.'s (2002) study found that participants used the low-VIR overview to navigate to targets and performed the detail work in the *hiVIR* display.

Low VIR can also be useful for refinding. In Hornbæk et al.'s study on electronic-document reading, reading pattern analysis showed that participants "used the overview pane to directly jump back to previously visited targets" and "the overview pane supports [sic] helps reader memorize important document positions" (p. 145) and resulted in participant preference and satisfaction, even though this apparent navigation advantage failed to materialize as time performance benefits (Hornbæk and Frokjær 2001; Hornbæk et al. 2003).

Supported: low-VIR view provides overall data structure

Low VIR can provide a data structure that may not be apparent in higher VIRs. For example, Hornbæk et al.'s study on reading electronic documents found that document section and subsection headers shown on the low-VIR view of their *separate* interface "may indirectly have helped subjects to organize and recall text" (p. 144), and led to higher quality essay without any time penalty.

Open: low-VIR region aids orientation

When the information space contains little or no information for which we can base our navigational decisions, the problem of "desert fog" occurs (Jul and Furnas 1998). Global context in *embedded* displays has been proposed to help users orient (Nigay and Vernier 1998), perhaps by providing visual support for working memory as the display gives evidence of where to go next (Card et al. 1999).

While we did not find evidence to study this role of the *embedded* low-VIR region, results from Hornbæk et al.'s (2002) study on map navigation may shed some lights on the topic.

Results from Hornbæk et al.'s (2002) study suggested that visual cues in data aided navigation. In their study, the Washington map contained rich visual cues for navigation. Participants were faster in navigation tasks performed using their *temporal* interface with the Washington map without the low-VIR view, suggesting that the map contained visual objects that aided navigation. In contrast, participants using the Montana map made a smaller number of scale changes when the low-VIR display was present, suggesting that the map itself did not contain enough visual objects for effective navigation, and participants needed the guidance of the low-VIR overview.

If visual objects displayed in low-VIR regions of *embedded* interfaces act similarly to navigational cues in the Washington map, it would be likely that low-VIR regions can aid orientation.

Open: low-VIR region provides data meaning

It is believed that data value is only meaningful when interpreted in relation to surrounding entities, and "the surrounding entities at different scales of aggregation exert a semantic influence on any given item of interest" (Furnas 2006). Again, we did not find *embedded* results to study this low-VIR region role. However, Saraiya et al.'s (2005) study on displaying time-series data as nodes in a graph may provide some understanding.

Saraiya et al.'s (2005) study included a *hiVIR* interface that showed all 10 time points simultaneously and a *temporal* interface that showed one data point at a time. Even though participants made more errors overall when using the *hiVIR* interface, thus suggesting having surrounding entities may be detrimental rather than helpful, a closer look at individual tasks showed mixed results.

We focused on tasks that involved all time points as they were more likely to involve comparative interpretations. The study reported the *temporal* interface supported faster task time in finding the topology trend of a larger graph and in searching for outlier time points. These two results suggested that despite having to identify trends or detect outliers, context provided in the *hiVIR* interface was detrimental rather than beneficial, possibly due to visual clutter. On the other hand, participants achieved better performance results with the *hiVIR* interface for the two tasks that involved finding outlier nodes and groups, and did not exhibit any performance differences for tasks that involved finding time trends.

Given the mixed results from Saraiya et al.'s (2005) study, we were unable to offer any insights into the role of low-VIR regions in providing data meaning for comparison.

4.4.6 Summary of considerations in low-VIR creations

Creating low-VIR displays is the second step in our decision tree (Figure 4.1). The first consideration is to determine the number of VIRs needed. Study results suggested that the number of VIRs in an interface should match the number of levels in the displayed data, as extra VIRs may hinder performance. Similarly, the low-VIR overview should only display task-relevant information, as extra information may be distracting. Information displayed should be perceivable in order to be useful. For text, readability is an important consideration; for graphical objects, the definition is less clear. Oftentimes, there are too many items in the data than what can be accommodated on the output device. Even though *a priori* selection of display data is an attractive solution, study results have found that doing so could lead to user confusion and distrust.

4.5 Decision 3: Simultaneous or Temporal Displays of the Multiple VIRs

The third decision in the process of creating a multiple-VIR interface is on VIR arrangements. For the designer, it is a choice between showing the VIRs simultaneously or one at a time, as in zooming techniques.

A well-known problem with zooming is that when the user zooms in on a focus, all contextual information is lost. Loss of context can be a considerable usability obstacle, as users need to integrate all information over time, an activity that requires memory to keep track of the temporal sequence and their orientations within that sequence (Herman et al. 2000; Furnas 2006). To alleviate these problems, a set of techniques collectively called focus + context were developed. Indeed, Card et al. (1999) stated the first premise of focus + context visualization as that "the user needs both overview (context) and detail information (focus) simultaneously" (p. 307). Another problem of zooming is that it "'uses up' the temporal dimension—making it poor for giving a focus + context rendering of a dynamic, animated world" (Furnas 2006).

Although this reasoning appears to be logical, empirical study results did not consistently support using simultaneous VIR displays: study results suggested that the *temporal* interface was surprisingly good for most tasks. We identified two situations where the simultaneous-VIR display provided performance benefits: when the answer to the problem involved information from ***all*** the available VIRs (Section 4.5.1), and when the different VIRs provided clues for the task (Section 4.5.2). Otherwise, temporal switching seemed adequate.

4.5.1 Consideration 1: tasks with single-level answers may not benefit from simultaneous VIR displays

In general, we found that simultaneous-VIR display was best suited for tasks that required multi-level answers. We focused on 10 of the 19 studies as they included a *temporal* and at least one simultaneous-display interface for comparison (Table 4.9). We excluded Hornbæk et al.'s (2002) study in this discussion since their *separate* interface, the zoomable interface with an overview, was effectively used as just a *temporal* interface most of the time.

Three of these 10 studies had at least one task that required multi-level answers, and all showed performance benefits in using their simultaneous-display interfaces for those tasks compared to their *temporal* interfaces.

In Bederson et al.'s (2004) study, the *embedded* DateLens interface was found to be more effective than the *temporal* Pocket PC interface in tasks that involved counting events within a 3-month time period in the calendar, for example, in counting scheduled events or appointment conflicts.

In Plaisant et al.'s (2002) tree browsing study, the SpaceTree *embedded* in-

Papers	**MA**	**SA**		
		MC	**SC**	**SB**
A. Baudisch et al. (2002)		x		
C. Bederson et al. (2004)	x		x	x
G. Hornbæk and Hertzum (2007)		x	x	x
K. Nekrasovski et al. (2006)			x	x
M. Pirolli et al. (2003)		x		
N. Plaisant et al. (2002)	x	x		
O. Plumlee and Ware (2006)		x		
R. Schaffer et al. (1996)	x		x	x
S. Shi et al. (2005)			x	
F. Hornbæk et al. (2002)**(ex)**				

Table 4.9: Ten papers that included a *temporal* and at least one simultaneous-display interface. MA = Multiple-level answers; SA = Single-level answers; MC = Multiple-level clues; SC = Single-level clues; SB = Single-VIR interface better supported tasks; ex = excluded from review.

terface trials were faster than the *temporal* Explorer interface on average and more accurate in a task that required listing all the ancestors of a node.

In Schaffer et al.'s (1996) re-routing task, participants were required to find an alternative route to connect two points in the network that were disconnected, and the route spanned all levels in the hierarchical network. The *embedded* interface supported faster task completion times and required only half the number of zooming actions when compared to the *temporal* interface. The advantage of the *embedded* interface could be its display of the ancestral nodes along with the children nodes at the lowest level of the hierarchy, since all of which were needed to find an alternative route.

On the other hand, the *temporal* interface seemed to offer better support for tasks with single-level answers, unless the clues required to reach the answers were also multi-level, as discussed in the next section.

4.5.2 Consideration 2: tasks with single-level information scent may not benefit from the simultaneous display of different visual resolutions

For tasks with single-level answers, simultaneous-VIR display was still helpful if the clues to the tasks spanned multiple data levels. As shown in Table 4.9, five of the nine included studies with multi-level clues to single-level answers, and all except the Hornbæk and Hertzum (2007) study demonstrated benefits in using simultaneous-VIR displays.

In Baudisch et al.'s (2002) study, their multiple-VIR interfaces supported equal or better performances than their *temporal* interface in the route-finding and connection-verification tasks. Even though the answer could be obtained in the high-VIR view alone, both tasks required global relative locations in the low VIR and detail information in the high VIR.

Pirolli et al. (2003) looked at a similar phenomenon called information scent. Their study suggested that the *embedded* hyperbolic tree interface may support faster task time than the *temporal* file explorer interface at high-scent tasks. In their *embedded* hyperbolic interface, participants could see more of the hierarchical structure in a single view and traversed tree levels faster. Under high-scent conditions where ancestor nodes provided clues to task answers, this feature could be advantageous. In contrast, under low information scent conditions, participants examined more tree nodes when using the *embedded* than the *temporal* interface, and resulted in slower task times.

Plaisant et al. (2002) reported that the *embedded* SpaceTree supported equal or better task times in the first-time tree node finding tasks than the *temporal* Explorer interface. Even though the researchers did not provide enough task instructions for us to judge if the the task provided multiple-level clues, the researchers did mention providing hints to participants that seemed to span multiple levels: "To avoid measuring users' knowledge about the nodes they were asked to find (e.g kangaroos) we provided hints to users (e.g. kangaroos are mammals and marsupials) without giving them the entire path to follow (e.g. we didn't give out the well known step such as animals)." (p. 62).

In Plumlee and Ware's (2006) study, the task required matching complex clusters of three-dimensional objects, and clues to the answers were present in both the low-VIR view, showing the location of the candidate targets, and in the high-VIR view, showing the details required in visual matching. Their *separate* interface was found to better support the task when the total number of objects per cluster was above five items, in which case participants could no longer hold all the clues in their short-term memory when using the *temporal* interface.

One possible exception to this hypothesis is Hornbæk and Hertzum's (2007) study. The researchers looked at the usability of fisheye menus showing 100 and 292 items. The study found that known-item search tasks were solved faster and more accurately with the *temporal* cascading-menu interface. However, due to the various implementation-dependent usability issues with the simultaneous-VIR interfaces, we could not discern relative interface effectiveness based on

VIR arrangement alone, and we therefore excluded it from our analysis.

Taking Considerations 1 and 2 together, we concluded that tasks with single-level answers and single-level clues would not benefit from simultaneous display of the different visual resolutions. Indeed, that seemed to be the case based on study results, even for the tasks that required object comparisons. As long as participants could keep task-required information in their short-term memory, the *temporal* interface seemed adequate, and at times, even resulted in better participant performances and feedback.

As shown in Table 4.9, five of the nine included studies had at least one task that required single-level answers and provided single-level clues. All except the Shi et al.'s (2005) study results supported this general conclusion. We excluded Hornbæk and Hertzum's (2007) visual searches in menus in this discussions due to the nontrivial usability issues with their simultaneous-VIR interfaces.

Bederson et al.'s (2004) study showed that the *temporal* Pocket PC was more appropriate for simple calendar tasks that involved checking start dates of pre-scheduled activities and tasks that spanned short-time periods.

In Nekrasovski et al.'s (2006) study, where the task was to compare topological distances between colored nodes in a large tree, their results showed that their *temporal* interface outperformed their *embedded* interface, even though the task required comparison between objects. Indeed, their *temporal* interface was rated by participants as being less mentally demanding and easier to navigate.

In Schaffer et al.'s (1996) study, even though the *embedded* interface supported faster task times than *temporal* in rerouting within a hierarchical network, participants did not seem to need simultaneous-VIR display to locate broken links at the lowest network level, as indicated by the lack of performance differences between the *temporal* and the *embedded* interface trials for this link-location task.

The exception is Shi et al.'s (2005) study, where researchers found that their *embedded* interface supported faster task times than the *temporal* interface. In Shi et al.'s (2005) case, there may be a speed-accuracy tradeoff: the researchers observed that in some cases, their participants ignored potential targets that occupied a small amount of space and missed the small targets in less than 3.75% of the trials. Even though the researchers did not report task error rates, they reported that this phenomenon may have a more severe and adverse impact on their *embedded* than on their *temporal* interface trials. Also, there were participants who gave up when using the *embedded* interface, but they only timed-out in the *temporal* trials.

In short, simultaneous-VIR display is appropriate for multi-level answers or single-level answers found by multi-level clues. Otherwise, the *temporal* interface seemed adequate.

4.5.3 Considerations in choosing between temporal switching or simultaneous display of the VIRs

In general, simultaneous VIR display, as in *embedded* or *separate* interfaces, requires more complex interactions, while *temporal* interfaces can be taxing on the user's memory. Study results suggested that temporal switching was more suitable for tasks that did not involve multi-level answers, or did not provide multi-level clues to single-level answers.

4.6 Decision 4: How to Spatially Arrange the Visual Information Resolutions, Embedded or Separate?

The last step in our decision tree is to decide between the two spatial arrangements of simultaneous-VIR display: the interface can embed the different VIRs within the same window or show them as separate views. Proponents of the embed approach argued that the different VIRs should be integrated into a single dynamic display, much as in human vision (Card et al. 1999; Furnas 2006). View integration is believed to facilitate visual search, as it provides an overview of the whole display which "gives cues (including overall structure) that improve the probability of searching the right part of the space" (Pirolli et al. 2003, p. 21), and integrated views of data is argued to "support and improve perception and evaluation of complex situations by not forcing the analyst to perceptually and cognitively integrate multiple separate elements" (Thomas and Cook 2005, p. 83). Also, it is believed that when information is broken into two displays (e.g., legends for a graph, or overview + detail), visual search and working memory consequences degrade performance as users need to look back and forth between the two displays (Card et al. 1999; Pirolli et al. 2003). On the other hand, spatial embedding frequently involves distortion, an issue discussed in Section 4.6.1.

The choice between these two spatial arrangements is unclear based on empirical study results. Oftentimes, perceived functions of the two interfaces bi-

Embedded* vs. *separate	**Papers**
Unable to compare	B. Baudisch et al. (2004) E. Hornbæk and Frokjær (2001); Hornbæk et al. (2003) K. Nekrasovski et al. (2006)
No difference	G. Hornbæk and Hertzum (2007) J. Lam et al. (2007) Q. Schafer and Bowman (2003)
embedded better	A. Baudisch et al. (2002) D. Gutwin and Fedak (2004)**(ex)**

Table 4.10: Eight papers that included both *embedded* and *separate* interfaces, classified by participant performances. ex = excluded from analysis.

ased study data and task selections. For example, studies tended to use trees or graphs for node finding to study *embedded* interfaces (e.g., Plaisant et al. 2002; Pirolli et al. 2003; and Shi et al. 2005) and spatial navigation for *separate* displays (e.g., North and Shneiderman 2000 and Plumlee and Ware 2006). As a result, the issue of spatial arrangement was frequently confounded in our reviewed studies.

As shown in Table 4.10, 8 of the 19 studies included both *embedded* and *separate* interfaces. We found it difficult to directly compare between the two simultaneous displays in three of the studies. For the remaining five studies, three did not find significant performance differences. Only Baudisch et al. (2002) and Gutwin and Fedak's (2004) studies demonstrated superior performance support of their *embedded* interfaces. In the case of Baudisch et al. (2002) the performance differences were possibly due to the unique implementation of their interface, while Gutwin and Fedak's (2004) results were possibly due to comparatively complex interactions required in their *separate* interfaces.

Mixed results in our reviewed studies, as shown in Table 4.10, may reflect the different tradeoffs in these interfaces. Also, in some cases, the benefit of providing multiple VIRs may be so large that the spatial arrangement may not matter (Tory et al. 2006, p. 12).

Of the three studies that we decided to be incomparable with the other five, two of them were excluded due to intentional implementation differences based on common perceived use of the two spatial arrangements: low-VIR view in the *separate* interface to display data overview, and low-VIR regions in the *embedded* interface to show background and supporting information. The first is Baudisch et al.'s (2004) study on web document search. Their *embedded* interface was designed to favour row discrimination and their *separate* interface favoured for column discrimination, thus adding another factor that influenced study results.

Hornbæk et al.'s study showed different kinds of information in their two multiple-VIR interfaces (Hornbæk and Frokjær 2001; Hornbæk et al. 2003). The low-VIR view of their *separate* interface provided document section and subsection headers and was optimal for showing overall structure in text documents and for encouraging detail explorations. In contrast, their *embedded* interface showed *a priori* determined text significant to the focal area, which promoted rapid document reading at the cost of accuracy.

The last study in the incomparable group did not intend to study spatial arrangement despite including both *separate* and *embedded* interfaces. In Nekrasovski et al.'s (2006) study on large tree displays, the goal of their *separate* interface was to investigate the use of an extra low-VIR view. Consequently, neither of their *separate* interfaces (*temporal* with overview and *embedded* with overview) could be directly compared with their *embedded* interface to discern effects of spatial arrangements.

In the five cases where direct comparison was possible, three studies did not find performance differences between the two simultaneous interfaces. The two exceptions were Baudisch et al. (2002) and Gutwin and Fedak's (2004) studies.

Even though Gutwin and Fedak's (2004) study on steering tasks showed significant results, we believe their results may be confounded by the relatively complex interactions required in their *separate* interfaces. The study included three *embedded* fisheye displays and two *separate* displays. In a series of two-dimensional steering tasks where participants were required to move a pointer along a defined path, the study found that the *embedded* interfaces supported better time and accuracy performances over the *separate* interface at all display magnifications. The researchers thus concluded that "the fact that fisheyes show[ed] the entire steering task in one window clearly benefited performance" (p. 207).

However, we believe a number of factors were involved in addition to the different VIR spatial arrangements. The first factor was differing effective steering path widths and lengths between interfaces. Of the five study interfaces, only one of the *separate* interfaces, the Panning-view, had an increased travel length at higher magnifications. All other interfaces had constant control/display ratios over all magnifications. As for the Radar-view *separate* interface, participants interacted with the low-VIR miniature view instead of the magnified high-VIR view, thus the actual steering path width was effectively constant over all magnifications.

We also found that interaction complexity differed greatly among the five interfaces. Their Panning-view *separate* interface had more complex panning interactions than the other interfaces, especially at higher levels of magnification of the steering path. The Panning-view *separate* interface required two mouse actions, mouse drag for panning and mouse move for steering, while the Radar-view *separate* interface required only mouse-drag on the miniature low-VIR view. In contrast, the *embedded* interfaces required only a single mouse action to shift the focal point and magnify the underlying path. This type of interaction, however, has the disadvantage of magnification-motion effect, where objects in the magnifier appear to move in the opposite direction to the motion of the lens, and is easier to overshoot the motion and slip off the side of the lens. We considered this motion effect as a third factor in the study.

Given the complex interplay of at least three factors that seemed to be implementation specific, we failed to extract general conclusions on VIR spatial arrangement based on Gutwin and Fedak (2004) study.

Baudisch et al.'s (2002) study looked at three tasks that required information from all VIRs: a static route-finding task, a static connection-verification taks, and a dynamic obstacle-avoidance task. Study results indicated that the *embedded* interface better supported all of the tasks and was preferred by participants. Their unique *embedded* interface implementation avoided many of the usability pitfalls in embedding high-VIR regions into low-VIR displays, which may explain its superior participant performance: first, the location for the high-VIR region was fixed, thus potentially avoiding disorientation with a mobile focus in respect to the context area and the associated complex interactions, and second, distortion was not used in the system. Instead, the researchers used different hardware display resolutions for the two different VIRs. In contrast, their *separate* interface seemed more interactively complicated than the usual implementation, requiring panning in both low- and high-VIR views and zooming in the high-VIR view. Nonetheless, we believe their study demonstrated an effective use of their *embedded* interface over their *separate* interface.

We conclude that there is not sufficient evidence to derive design guidelines in choosing between the two simultaneous displays, as it is difficult to draw conclusions based only on Baudisch et al.'s (2002) study.

4.6.1 The issue of distortion

One of the potential costs in embedding multiple VIRs within the same window is distortion. Based on Furnas's (1986) fisheye views and on studies of attention, Card et al. (1999) justified distortion since "the user's interest in detail seems to fall away from the object of attention in a systematic way and that display space might be proportioned to user attention". Also, Card et al. (1999) reasoned that "it may be possible to create better cost structures of induced detail in combination with the information in focus, dynamically varying the detail in parts of the display as the user's attention changes [...] Focus and context visualization techniques are 'attention-warped' displays, meaning that they attempt to use more of the display resource to correspond to interest of the user's attention" (p. 307).

Even though distortion is believed to be justified, it is still useful to examine the costs. The first problem is that distortion may not be noticed by users and be misinterpreted (Zanella et al. 2002), especially when the layout is not familiar to the user or is sparse (Carpendale et al. 1997). Even when users recognize the distortion, distance and angle estimations may be more difficult and inaccurate when the space is distorted (Carpendale et al. 1997), except perhaps in constrained cases such as bifocal or modified fisheye distortions (Mountjoy 2001). Also, users may have difficulties understanding the distorted image to associate the components before and after the transformation (Carpendale et al. 1997), or in identifying link orientation in the hyperbolic browser (Lamping et al. 1995).

To our knowledge, only three published studies measured effects of distortion directly and systematically. Lau et al. (2004) found that a nonlinear polar fisheye transformation had a significant time cost in visual search, with performance slowed by a factor of almost three under large distortions. In terms of visual memory costs, our laboratory experiment, reported in Chapter 5, found image recognition took longer and was less accurate at high fisheye transformation levels. Skopik and Gutwin (2005) reported a time penalty without compromising accuracy on refinding nodes in a highly-linked graph when the graph was transformed by a polar fisheye transformation.

It was difficult to tease out the effects of distortion based on the 19 papers we reviewed here, since none of the studies specifically looked at distortion as a factor. We could therefore only rely on observations reported in the papers to obtain insights. As shown in Table 4.11, 14 studies included an *embedded* interface, and 12 implemented distortion. The two exceptions were Baudisch

Papers	**Distortion**			**Effects**	
	None	**Text**	**Grid**	**pos**	**neg**
A. Baudisch et al. (2002)	x				
B. Baudisch et al. (2004)		x		x	
C. Bederson et al. (2004)			x	x	
D. Gutwin and Skopik (2003)**(ex)**				x	
E. Hornbæk and Frokjær (2001)		x		x	
G. Hornbæk and Hertzum (2007)**(ex)**		x			x
H. Jakobsen and Hornbæk (2006)		x		x	
J. Lam et al. (2007)	x				
K. Nekrasovski et al. (2006)					x
M. Pirolli et al. (2003)					x
N. Plaisant et al. (2002)					x
Q. Schafer and Bowman (2003)					x
R. Schaffer et al. (1996)			x	x	
S. Shi et al. (2005)			x	x	

Table 4.11: Fourteen papers that included at least one *embedded* interface. pos = Performance benefits demonstrated; neg = Problems reported; ex = excluded from review.

et al. (2002) and Lam et al. (2007). Baudisch et al. (2002) took a hardware approach and implemented their *embedded* interface with two different pixel resolutions and Lam et al. (2007) used two distinct visual encodings to represent the same data in two VIRs.

Interestingly, not all 12 studies reported usability or performance problems with visual distortion. In fact, seven studies reported performance benefits in using their distortable interfaces. We excluded Gutwin and Skopik (2003) in this analysis as we could not tease out the effects of distortion based on study results due to the large number of factors involved in the study, as discussed earlier in this section. The remaining six studies that demonstrated positive effects of distortion involved either text or grid-based distortions, suggesting that constrained and predictable distortions were well tolerated.

Five studies reported problems attributed to distortion, and all involved comparatively more drastic and elastic distortion techniques than text or grid-based distortions. We also excluded Hornbæk and Hertzum (2007) in our analysis since, even though the researchers reported usability problems with their various *embedded* and *separate* interfaces, it is unclear how distortion contributed to these problems. We therefore focused our discussion on the remaining four studies to further understand distortion costs.

Nekrasovski et al.'s (2006) *embedded* interface implemented Rubber-Sheet Navigation that allowed users to stretch or squish rectilinear focus areas as though the data set was laid out on a rubber sheet with its borders nailed down (Sarkar et al. 2003). The researchers attributed the relatively poor performance of their *embedded* interface to the disorienting effects of distortion (p. 18).

Plaisant et al.'s (2002) study found that their participants took longer to refind previously-visited nodes in a tree using the *embedded* hyperbolic and SpaceTree interfaces than with the traditional *temporal* Microsoft Explorer file browser. Among the two distortion interfaces, participants demonstrated better performance with SpaceTree than with the hyperbolic tree browser, which involved more drastic distortions. This result was predicted by the researchers as in SpaceTree, "the layout remains more consistent, [thus] allowing users to remember where the nodes they had already clicked on were going to appear, while in the hyperbolic browser, a node could appear anywhere, depending on the location of the focus point" (p. 62).

Pirolli et al.'s (2003) study also compared between a *temporal* file browser and the *embedded* hyperbolic tree browser. The researchers found that the hyperbolic tree browser supported better performance only for tasks with high-information scent. Even though the researchers did not explicitly report problems related to distortion, they suggested providing landmarks to aid navigation in the *embedded* hyperbolic tree browser, thus indicating potential interaction costs in hyperbolic distortions.

Schafer and Bowman's (2003) *embedded* interface implemented the radar fisheye view on maps. Their study reported both positive and negative effects of distortion. On the positive side, if noticed, the distortion enhanced awareness to the viewport in a collaborative traffic and sign positioning task using a map. However, users may not notice the distortion as it may not be caused by their direct action since the task was collaborative.

In short, while we believe interfaces that implement distortions were generally more difficult to use, constrained and predictable distortions were found to be better tolerated and may tip the tradeoff between showing more information simultaneously on the display and the risk of causing disorientation and confusion.

4.6.2 Considerations in spatially arranging the various VIRs

There are tradeoffs in using either of the two simultaneous displays, *embedded* and *separate*. *Embedded* interfaces tend to implement distortion, which may be difficult for participants to understand and may involve difficult interactions. For *separate* interfaces, view coordination has been found to be difficult. Study results regarding this question were mixed.

4.7 Summary: Design Recommendations

We summarize our findings as three recommendations to designers in creating multiple-VIR interfaces.

4.7.1 Provide the same number of VIRs as the levels of organization in the data

Furnas argued for the need to provide more than two VIRs in his 2006 paper:

> By presenting only two levels, focus and context, these differ from the richer range of trading off one against the other represented in the canonical FE-DOI. This difference must ultimately prove problematic for truly large worlds where there is important structure at many scales. There the user will need more than one layer of context.

In the same paper, he also argued that the levels of resolutions can be determined based on the scale bandwidth of the presentation technology and scale range of the information world (Furnas 2006, p. 1003).

Looking at the question from a different angle, study results suggested that the effectiveness in providing multiple VIRs, especially simultaneous display of different VIRs, was contingent upon the the number of organization levels in the data and the information needs of the task. In fact, we found that having extra VIRs may actually impede task performance, especially in *temporal* interfaces where users coordinate between the different VIRs using short-term memory. we believe that interface should therefore provide one VIR per data level.

4.7.2 Provide relevant, sufficient, and necessary information in the low-VIR displays to support context use

While the high VIRs should support detail work demanded by the tasks at hand, study results suggested that low-VIR views in *separate* interfaces were used in two ways: in navigation where they provided short-cuts to jump to different parts of the data; and in mental data organization if they displayed overall data structure. To be effective, designers need to include only sufficient, relevant, and necessary information in the low-VIR views. This finding is in accordance with Norman's (1993) Appropriateness Principle, where he stated that the visual representation should provide neither more or less information that is needed for the task at hand since extra information displayed may be distracting and render the task more difficult. In the case of multiple-VIR interfaces, displaying an inappropriate amount of information may tip the balance as the value of the display may not be sufficient to overcome the costs of having the extra visual resolutions. The amount of detail for each visual object displayed on low-VIR views is likely to be more than previously assumed in our community, judging from the number of ineffective low-VIR views created for the reviewed studies. For text documents, readability may be a requirement, as suggested in Jakobsen and Hornbæk (2006): the design should "saturate the context area with readable information" in building interfaces to display program source code (p. 386), and in Hornbæk and Hertzum (2007): "making the context region of the [fisheye menu] interfaces more informative by including more readable or otherwise useful information" (p. 28). For graphical displays, studies on visual search (e.g., Tullis 1985) and Lam et al.'s (2007) study provided guidelines, for example, visual signals should be simple and of narrow visual spans to be accessible, but the criteria still remain unclear.

4.7.3 Simultaneously display VIRs for multi-level answers or multi-level clues

Selecting the correct visualization technique to display data is important due to the inherent tradeoffs in the *temporal*, *separate*, and *embedded* techniques. While most *temporal* implementations offer familiar panning and zooming interactions, these interfaces require users to keep information in their short-term memories. Simultaneous-VIR displays, on the other hand, often require more complex and unfamiliar interactions such as view coordinations. Based on study results, we concluded that if the task or subtask needs information from multiple levels, either as part of the answer to the task or as clues leading to the answer, the interface should show multiple levels simultaneously. Otherwise, the *temporal* technique should be more suitable due to its simpler interface and more familiar interactions.

4.7.4 Open question: how should multiple VIRs be displayed simultaneously?

Unfortunately, we are unable to suggest guidelines in displaying multiple VIRs simultaneously, either as *embedded* or as *separate* displays, due to the difficulties in obtaining direct interface comparisons based on our set of reviewed studies.

4.8 Summary: Methodology Recommendations

Despite being able to use most of the study results in our analysis, we encountered difficulties in interpreting selective study results and had to exclude them from our analysis. Part of our difficulty may be due to differences between our goals and those of the reviewed studies: we aimed to tease out factors that affect visualization use instead of overall interface effectiveness. While evaluating visualization using experimental-simulation studies has been argued to be difficult due to the lack of standardized tasks, effective measurements to capture interface use, and ecologically validity (Plaisant 2004), we believe the method can be improved even without data and task repositories, novel measurements, or abandoning the experimental strategy for field strategy. To identify areas that could be improved upon, we looked at the four main scenarios that led to result exclusion in our summary synthesis:

1. Study interfaces were not comparable at the individual factor level, such as visual elements, information content and amount displayed, level of organization displayed, and interaction complexity;
2. Measurements were not sensitive enough to capture usage patterns, which were needed to understand factors at play in visualization use;
3. Studies investigated multiple interface-use factors, making it difficult to isolate effects of each;

4. Studies did not report sufficient details for our analysis, since we wished to extract effects of selective design factors in interface use instead of overall system or technique effectiveness.

We therefore argue that by using comparable study interfaces, capturing usage patterns in addition to overall performance measures, isolating interface-use factors, and by reporting more study details, we can increase consistency among experimental-simulation studies and increase their utility, since study results will then be more amenable to meta-analysis.

4.8.1 Use comparable interfaces

In order to understand factors influencing interface use, studies should identify possible factors at play, and if possible, vary one experimental factor at a time, as in factorial designs. For visual design, some factors include the interfaces' basic visual elements such as the number of views and the use of image distortion, the amount and type of information displayed, and the number of levels in the displayed data. For interaction, study designers should consider the required number of input devices, the types of action required, and the number of displays on which the action is applied.

Basic visual elements

While it is understandable that interfaces in empirical studies may be dramatically different in appearance, they should be comparable in their basic visual elements whenever possible to allow for direct comparison. For example, in Baudisch et al.'s (2004) study on visual searches on webpages, they included two interfaces that showed web documents at two levels of detail simultaneously. The *separate* interface had a scrollable detail page and an low-VIR view that showed the entire webpage by compressing all elements equally. The *embedded* interface was a non-scrollable browser that showed the entire webpage by differentially compressing pertinent versus peripheral content in order to keep the pertinent text readable. On the surface, the two interfaces were ideal candidates for studying the effects of spatial arrangement of the low- and the high-VIR components: in the *separate* interface, the two components were arranged as separate views; in the *embedded* interface, they were embedded into a single view.

However, there was another factor at play that affected performance results.

Since their interfaces displayed readable words pertinent to their study tasks as highlighted popouts, the spatial association between the original web documents and these popout words became important. Unfortunately, association by row was found to be more difficult in their *embedded* interface than by column, as their focus + context implementation selectively distorted the vertical dimension. On the other hand, their *separate* implementation proportionally reduced both vertical (row) and horizontal (column) dimensions. Their study results reflected the interfaces' ability to associate popouts with document rows and columns: their *embedded* interface better supported a task that did not require row-specific information (the *Product Choice* task), but not for row-dependent tasks (e.g., the *Co-occurrence* task). The *separate* interface results showed opposite trends.

Since Baudisch et al.'s (2004) study aimed to evaluate overall effectiveness of their novel *embedded* interface relative to two existing techniques, both the visual components' spatial arrangement and the row-column association with the highlighted popouts were part of their interface design and should be evaluated together. However, when we tried to tease out the effect of spatial arrangement to extract general design guidelines, we could not to isolate the effect and therefore could not include their study results in our analysis.

We encountered similar problems in analyzing Bederson et al.'s (2004) study on PDA-size calendar use. Their study looked at two interfaces: the Pocket PC calendar that provided a single level of detail per view (day, week, month, or year), and the DateLens interface that used a Table Lens-like distortion technique to show multiple levels of details simultaneously. Again, the study seemed to compare the effects of providing separate views one at a time, or embedding them in a single view.

Their study looked at a variety of calendar tasks that involved searching for appointments, navigation and counting scheduled events, and scheduling given constraints. While their study did not find an overall time effect, the researchers found a task effect and thus divided the tasks into *simple* and *complex* tasks based on task-completion time. The study concluded that the DateLens trials were faster in *complex* tasks, while the Pocket PC trials were faster in *simple* tasks.

On closer inspection, we realized that while their DateLens interface provided a day, week, month, and year view, it also provided a three-month and a six-month view, with the three-month view being the default in the study. On the other hand, the Pocket PC interface did not seem to have a corresponding three-month overview. If that were the case, since three of their six *complex* tasks (tasks 5, 10, 11) required scheduling and counting events within three-month periods, we could not determine if the benefits of the DateLens interface in these tasks came from providing a three-month overview, or from providing multiple levels of details in the same view. Again, our need to understand performance contribution from individual factors forced us to exclude these results from our analysis.

Information content

We encountered difficulties in direct comparison of interfaces with different information displayed, as the interfaces were often used for different purposes. One example is Hornbæk et al.'s study on online document reading (Hornbæk and Frokjær 2001; Hornbæk et al. 2003).

Their study looked at two interfaces that provided multiple levels of data simultaneously. The low-VIR view in their *separate* interface showed document header and subheaders and acted as a table of contents. Their *embedded* interface showed context based on a degree-of-interest algorithm, thus the content was dynamic based on the focal point of the document. Not surprisingly, participants used the two interfaces differently. Reading patterns indicated that when using the *embedded* interface, participants spent more time in the initial orientation mode, but less time in the linear read-through mode, suggesting that the *embedded* interface shortened navigation time by supporting an overview-oriented reading style. In contrast, reading patterns in the *separate* interface was found to be less predictable and "shaped by situation-dependent inspiration and associations", and "the overview pane grabs subjects' attention, and thereby leads them to explorations that strictly speaking are unnecessary" (p. 144), probably because display was similar to a table of contents. Study results reflected the different information content displayed in these low-VIR displays. Compared to the *embedded* interface, participants who used the *separate* interface produced better results in the essay tasks at the expense of time, and the study failed to find differences between the two interfaces for the question-answering tasks. While the different information content was arguably part of the interface design, we could not incorporate results from this study in our analysis as we could not separate out visual spatial effects from those of displaying different kinds of information in the low-VIR displays.

Levels of display

The total amount and levels of information provided by the interface is also important, as extra information or levels may be detrimental to performance. One example is Plumlee and Ware's (2006) study on visual memory in zooming and multiple windows. Their *temporal* interface had a continuous-zoom mechanism that showed intermediate levels of detail that did not seem to be present in the *separate* interface, which seemed to have only two levels based on the researchers' descriptions. Their study task required participants to match complex clusters of three-dimensional objects. To do so, participants needed to first locate clusters at the low-zoom level and match cluster components at the high-zoom level. Intermediate-zoom levels did not seem to carry task-relevant information as the clusters themselves were not visible given the textured backgrounds.

Plumlee and Ware (2006) stated that participants needed 1.5 seconds to go through a magnification change of at least 30 times between the lowest and highest zoom levels. During this time, participants needed to keep track of the components in various objects in their short-term memory. We wondered if having the extra levels of detail in their *temporal* interface unnecessarily degraded participants' visual memories and made the interface less usable. This extra cognitive load may explain the relatively small number of items participants could handle before the opponent *separate* interface became more appropriate for the task, in contrast to the results of a 2005 study on graph visualization by Saraiya et al. (2005). Saraiya et al.'s (2005) Single-Attribute *temporal* interface supported better performance than their Multiple-Attribute *hiVIR* interface even when the task involved a 50-node graph, each node with 10 time points. Due to the differing levels of data displayed in the two study interfaces, we excluded Plumlee and Ware's (2006) study from our analysis to understand the conditions in which simultaneous display of multiple data levels is beneficial.

Similarly, Baudisch et al. (2002) studied static visual path-finding tasks and dynamic obstacle-avoidance task using three interfaces each providing multiple levels of details. Their zoom and pan *temporal* interface and their overview plus detail *separate* interface seemed to support more levels of detail than their focus plus context *embedded* interface, which had two levels only. Their *embedded* trials were faster than the *temporal* and the *separate* trials for the static visual path finding tasks, and were more accurate in the dynamic obstacle-avoidance task. While the special hardware setup in their *embedded* interface undoubtedly contributed to the superior participant performance, we wondered if the extra resolutions may have distracted participants in the other two interface trials, even though we did include this study in our analysis of simultaneous displays of multiple levels of detail as we believe the difference in the number of display level was small.

Levels in data

Since researchers have argued that the interface should only display a different data resolution if it is meaningful to the task at hand (e.g., Furnas 2006), the number of displayed data levels is an important consideration. For example, in Hornbæk et al.'s (2002) study on map navigation, there were surprisingly large differences in usability and navigation patterns between the two study maps, despite being similar in terms of the number of objects, area occupied by the geographical state object and information density. The maps differed by the number of levels of organization: the Washington map had three levels of county, city, and landmark, while the Montana map was single-leveled. Perhaps for this reason, the study failed to find differences in participant performance when using the two study interfaces with the Montana map, but their participants were faster in a navigation task and more accurate in the memory tasks using just the *temporal* interface without an overview with the Washington map. We took advantage of this unintended data-level difference to examine how interfaces with multiple levels of display data support single-leveled data. These fortuitous opportunities for re-analysis were, however, rare.

Interaction complexity

In some cases, interaction style may be a factor in the study, and in others, unintended differences in interaction complexities among the interfaces studied may not be avoidable. Nonetheless, interaction complexity differences make comparison difficult, as seen in Hornbæk and Hertzum's (2007) study on fisheye menus.

In their 2007 study, Hornbæk and Hertzum's (2007) intention was to study the visual design and use of fisheye menus (Hornbæk and Hertzum 2007). They had four interfaces: a traditional cascading menu (*temporal*), the Fisheye menu as described by Bederson (2000) (*embedded*), the Overview menu (*separate*), and the Multifocus menu (*embedded*). The *separate* and the *embedded* Multifocus interfaces were both based on Bederson et al.'s (2004) Fisheye menu, and all three simultaneous-VIR interfaces implemented the focus-lock interaction to aid menu-item selection. The *separate* interface did not implement distortion, and showed a portion of the menu items based on mouse position along the menu, showing the field-of-view in the low-VIR view. The *embedded* Multifocus interface showed important menu items in readable fonts, and did not have an index of letters as in the *embedded* Fisheye or the *separate* interfaces.

Surprisingly, their *temporal* interface outperformed all other simultaneous-VIR interfaces. The researchers suggested that one possible reason was the relatively simple navigation in the *temporal* interface: their participants encountered obvious and severe difficulties in using the focus-lock mode in the other interfaces. While the researchers successfully identified a usability problem in the Bederson et al.'s (2004) Fisheye menu, we could not conclude if the visual designs of the other three simultaneous-VIR interfaces were truly inferior to the *temporal* interface that showed one VIR at a time.

Ensure comparable interface with follow-up studies

Our recommendation to use comparable interface can be difficult to implement. One challenge is to identify study elements prior to the study to ensure comparability. For example, in Hornbæk's *et al.*'s study on map navigation, the researchers did try to use comparable maps (Hornbæk and Frokjær 2001; Hornbæk, Frokjaer, and Plaisant 2003). Differences between the two study maps were only apparent after the study.

Another difficulty in adhering to these suggestions may be due to a conflict of evaluation goals: the goals of the original designs were to compare between systems at the overall-performance level, while our goal was to extract the effects of interface-use factors in systems. It is therefore difficult to modify original study designs without changing these goals, since the systems themselves are complex and are frequently incomparable at the interface-factor level.

In both cases, we believe follow-up studies are needed. Follow-up studies, either performed by the original researchers or by third parties, can take advantage of the knowledge gained in original studies or system-level studies, such as correcting mistakes made in original studies as in using different levels in data or different levels in interfaces. System-level evaluations can be used as a vehicle to identify factors, perhaps by detailed observations of how participants interact with the systems. These factors can then be studied in more detail and in isolation in subsequent studies. For example, Baudisch et al.'s (2004) Fishnet interface study identified at least two factors, the visual components' spatial arrangement and the row-column association with the highlighted popouts, which can be studied in isolation with appropriate study designs.

4.8.2 Capture usage patterns

In most reviewed studies, the main measurements were performance time, accuracy, and subjective preferences. While these measurements provided valuable information on overall interface effectiveness, efficiency and user acceptance, they may not be sensitive enough to illuminate factors involved in interface use and to tease out design tradeoffs, especially when the study failed to find overall performance differences between the interfaces.

While most reviewed studies reported experimenter observations on participant strategy and comments to interpret performance results, only 5 of the 19 studies reported usage patterns, constructed either based on eye-tracking records or navigation action logs (Table 4.2).

We found these five studies to be most useful in our analysis. For example, Hornbæk et al.'s (2003) study on online document reading used progression maps to investigate reading patterns. Progression maps showed visible parts of the document during the reading process. The researchers interpreted study results using reading patterns derived from these progression maps and provided a richer understanding of how the study interfaces were used. For example, their reading pattern explains the longer performance time in the question-answering task trials using the *separate* interface: "further explorations were often initiated by clicking on the overview pane", and "further exploration [of the displayed documents] happen[ed] because of the visual appearance of the overview and because of the navigation possibility afforded by the ability to click the overview pane". They therefore concluded that "the overview pane grabs subjects' attention, and thereby leads them to explorations that strictly speaking are unnecessary" (p. 144).

In another of their studies, Hornbæk and Hertzum (2007) looked at fisheye menu use. Despite not finding performance differences between their simultaneous-VIR interfaces, eye-tracking results showed interesting insights into how the interfaces were used: their participants used the low-VIR regions more frequently in the *embedded* Multifocus interface trials, possibly due to the readable information included. The researchers were therefore able to conclude based on usage pattern that designs should make "the context region of the [fisheye menu] interfaces more informative by including more readable or otherwise useful information" (p. 28).

Capture usage patterns with observations

We found that usage patterns provided rich insights in interface use regardless of statistical results. We therefore recommend recording and reporting detailed but non-intrusively collected usage patterns in study, from simple observations to detailed interactivity and eye-gaze logs.

4.8.3 Isolate interface factors

Information visualization systems are complex interfaces that typically involve visual encoding and interaction, and for some implementations, view coordination and image transformation. While simply identifying such factors is probably sufficient to evaluate system effectiveness, studying overall effects may obscure contributions from each factor, a difficulty we encountered during our analysis to draw design guidelines based on these factors.

That was the case when we looked at Gutwin and Skopik's (2003) study on two-dimensional steering, where at least three factors were at play. Their study looked at five *separate* and two *embedded* interfaces. In addition to the different spatial arrangements of the different levels of details in their interfaces, there were also different effective steering path widths and lengths and different interaction styles. Section 4.5 discusses potential factors in this study in detail.

Another type of difficulty we encountered in our analysis was to tease out usability factors involved in the *embedded* techniques. While showing all data as a single view in context may provide benefits, these techniques often require more complex interactions and image distortion, which has been shown to incur costs in orientation (Carpendale et al. 1997) and visual memory (Lam et al. 2006). Ideally, we would like to be able to study each of these factors in isolation. However, we were only marginally successful in teasing out the effects of distortion, as our study set had *embedded* interfaces that implemented different types and degrees of distortion. For example, Baudisch et al.'s (2002) study implemented their *embedded* interface with a hardware approach, using different pixel density in their displays to recreate the two regions, thus avoiding the need for distortion in their interface. Their study found performance benefits in all their tasks using their *embedded* display. In contrast, studies that implemented drastic and elastic distortion techniques reported null or mixed results, along with observed usability problems, for example the Rubber Sheet Navigation (Sarkar et al. 2003) in Nekrasovski et al.'s (2006) study, the Hyperbolic Tree browser in Plaisant et al. (2002) and Pirolli et al.'s (2003) studies, and the fisheye projections in Schafer and Bowman (2003). Despite this insight, our distortion classification is still rough, both in terms of classifying distortion types and performance effects. Section 4.6.1 further discusses distortion in *embedded* interfaces.

Isolate interface factors with follow-up studies

As in the case of using comparable interfaces, it may not always be possible to identify all relevant interface factors at experimental design. In addition, conducting fully-crossed experiments with a large number of factors may be too expensive. We therefore also recommend conducting follow-up studies to focus on a selected subset of the identified factors.

4.8.4 Report study details

One of the frustrations we had while analyzing our study set stemmed from the lack of details in study reporting. Indeed, Chen and Yu (2000) encountered similar problems in their meta-analysis. Since their meta-analysis synthesized significance levels and effect sizes, they had to exclude many more studies than in our qualitative analysis. Based on their experience, Chen and Yu recommended four standardizations in empirical studies: testing information, task taxonomy (for visual information retrieval, data exploration, and data analysis tasks), cognitive ability tests, and levels of details in reporting statistical results. They also asked for better clarity in visual-spatial properties descriptions and more focus on task-feature binding in studies. The researchers concluded that "it is crucial to conduct empirical studies concerning information visualization systematically within a comparable reference framework" (p. 864).

In addition to supporting Chen and Yu's (2000) recommendations, we have two further recommendations. We advocate reporting full task instructions. We also advocate documenting interface interactions with video, or even making the interface prototype software and trial experiments available for download. Allowing others to see or experience the exact instructions and interface behaviours seen by study participants would help reproducibility and clarify study procedures for later meta-analysis.

Although nine of the 19 reviewed papers provided detailed descriptions of the study tasks, only five provided actual task instructions. Since interface use can be severely affected by task nature such as the levels of detail required in task answers, it was difficult to analyze study results when the publications did not provide the written task instructions given to participants before the trials and any verbal hints given during the trials. For example, in our analysis, we needed to ascertain the factors that led to successful use of simultaneous-VIR displays. One possibility was when task instructions provided clues that span multiple data levels. Since we attempted to reinterpret the results based on different criteria, we encountered difficulties when the study did not provide enough task instructions for us to be certain if the task instructions provided multiple-level clues, for example in Plaisant et al.'s (2002) SpaceTree study where we had to guess based on study observations.

Even providing detailed task instructions may still be inadequate in some cases. For example, in Pirolli et al.'s (2003) preliminary task analysis study, their tasks were measured for information scent. Even though the researchers did provide a list of tasks, they did not cross-match the list with information scent scores, making it difficult for us to later associate task nature, information scent score, and study results. We therefore assumed the instructions of high information scent tasks provided useful clues at multiple levels of the tree.

For studies where interaction plays a pivotal role in study results, text descriptions of the interaction, no matter how detailed and carefully constructed, seem inadequate. One example is Hornbæk and Hertzum's (2007) study on the use of fisheye menu, where the focus-lock interaction was found to be one of the major usability problems. Despite the researchers' well-constructed descriptions, we did not fully understand the interaction until we tried the online fisheye menu prototype kindly provided by Bederson[1].

Report study details with online resources

We understand the strict page limits for research papers in many venues has required authors to make draconian choices in the amount of detail reported. Even without the page limits, such choices should be guided by the study goals and paper emphasis to ensure readability, as it is impossible to predict how study results may be used in future analysis. We therefore recommend that researchers provide study details as electronic supplementary materials in pub-

[1] `http://www.cs.umd.edu/hcil/fisheyemenu/`

lication venues that support archival availability of such materials, or as information posted on laboratory websites.

4.9 Limitations of Study

While we attempted to provide a comprehensive systematic review in the use and design of multiple-VIR interfaces, we were necessarily limited by our method, our own knowledge and time to include all relevant studies in our review. Section 4.1 discusses limitations in our methodology. Also, our synthesis was based entirely on the publications. In many cases, the goals of these reports were to directly compare interfaces as a whole, especially when one or more of the interfaces were novel. Given our goal to understand interface use, we often had to read the publications from a different perspective, and consequently, we may have misread or incorrectly inferred information from these publications.

4.10 Summary of Results and Implications for Design

We analyzed 19 existing multiple-VIR interface studies to extract design guidelines, and cast our findings into a four-point decision tree: (1) When are multiple VIRs useful? (2) How to create the low-VIR display? (3) Should the VIRs be displayed simultaneously? (4) Should the VIRs be embedded, or separated? We recommended that VIR and data levels should match, and low VIRs should only display task-relevant information. Simultaneous display of the different VIRs, rather than temporal switching between them, is suitable for tasks with multi-level answers, or task that provided multiple-level clues.

We identified two areas for further investigation in this thesis: the question of low-VIR display creation, and the issue of simultaneous-VIR display. We investigated these questions with the next three studies: two laboratory studies and a field evaluation. For completeness, we have included study results from the two laboratory studies in this review. The next two chapters report study details: Chapter 5 reports our laboratory experiment on visual memory costs in geometric transformation, discussed in this review in the context of distortion in Section 4.6.1; Chapter 6 reports our experimental-simulation study on overview use with single-level data, denoted in this review as Lam et al. (2007) and discussed in various sections as one of the studies analyzed. Our field evaluation continued our evaluation into overview use and simultaneous-VIR arrangement, and is reported in Chapter 8 after the discussion on the design of the study tool in Chapter 7.

Chapter 5

Laboratory Experiment: Visual Memory Costs of Image Transformations

The second study in this thesis further investigated issues of overview creation and spatial arrangements of visual resolutions discussed in the summary synthesis (Chapter 4).

Geometric transformations such as scaling, rotation, rectangular fisheye, and polar fisheye transformations are widely used in creating the low-VIR displays in multiple-VIR interfaces. Scaling, for example, has been used to create thumbnails for documents and rotation in graph navigation (e.g., Yee et al. 2002). Both fisheye transformations are often implemented in *embedded* interfaces, examples include rectangular fisheye transformations to realize text or grid-based distortions such as DateLens (Bederson et al. 2004) and polar fisheye transformations used in focus + context map applications such as those used in Schafer and Bowman's (2003) study. Section 4.6.1 discusses the various types of distortion implemented in *embedded* interfaces in more details.

Despite their wide-spread uses, there is a danger that the transformed images may be too distorted to remain recognizable. Unfortunately, the effects of these transformations on performance are largely unknown, as seen in our summary synthesis. Several design guidelines have been suggested to transform images with minimal disruption. These guidelines include:

- Maintain orthogonal ordering (left-right, up-down ordering), proximity (distance relationships between objects) and topology (inside-outside relationships) of the original image (Misue et al. 1995);
- Use visual cues to support the user's comprehension of geometric distortion (Carpendale et al. 1997). Background grids have been suggested as

the most effective of these (Zanella et al. 2002), as used in EPT (Carpendale et al. 1997).

- Use animation to retain the relationships among components displayed during transformation, and to avoid reassimilating the new display (Robertson et al. 1989). Many visualizations involving geometric transformation follow this principle, with earlier adopters being Pad++ (Bederson and Hollan 1994) and Table Lens (Rao and Card 1994).

While these guidelines may provide designers with some hints for handling geometric transformations, they are based mostly on casual experience, and are not detailed or quantitative enough for actual implementation. Clearly, different types of geometric transformations and different degrees of transformation incur different amounts of perceptual cost. Knowing these costs would help designers gauge cost-benefit tradeoffs in their applications. Quantifying the effectiveness of various techniques suggested by these guidelines to mitigate transformation costs would be also helpful. For example, since smooth animation may impose a heavy computational load, it would be useful to determine the largest transformation "jump" we can perceptually tolerate. Also, the presence of grids may create visual noise instead of being beneficial.

Extending earlier studies on geometric transformations and visual search (Rensink 2004; Lau et al. 2004), the goal of this work was to better understand and quantify the effects of two-dimensional geometric transformations on visual memory to guide interface and visualization design. In this study, we presented the first measurements of the effects of four types of geometric transformation on visual memory: scaling, rotation, rectangular fisheye, and polar fisheye transformations. These transformations were applied to automatically generated abstract images consisting of dots and connecting lines. Based on these results, we defined a no-cost zone boundary for each transformation type, after which task time and accuracy degraded. Based on our results, we refined two of the design guidelines mentioned above: Misue et al.'s (1995) orthogonal ordering requirement and the use of background grids to mitigate costs incurred by transformation (Zanella et al. 2002).

5.1 Experiments

We conducted ten experiments to investigate the effects of geometric transformations on visual memory. Two additional experiments were conducted as follow-up experiments. All experiments used a within-subject design. In each experiment, we considered only a single factor, the transformation type, looking at five levels of transformation degree. Each transformation level was blocked, with the order of level presentation partially counterbalanced across participants using the ordering listed in Appendix C.4.

Each level was tested using two phases, each with eight trials. In the learning phase, participants were presented with eight stimuli in sequence. In the recognition phase, they were shown another set of eight stimuli, 50% of which were shown in the learning phase. For each stimulus, participants were asked to determine whether it had been shown in the learning phase. Baseline performance was measured in terms of response time and accuracy obtained using untransformed test stimuli. This baseline was then compared with results of the transformed trials.

5.1.1 Transformations

We investigated four types of transformations to abstract images consisting of dots connected by lines: scaling, rotation, rectangular fisheye and polar fisheye. We also examined the effects of grid presence and grid type. Ten initial experiments were carried out:

- Scaling (1, 0.5, 0.33, 0.25, 0.2x reduction factor)

  Exp 1. no grid

  Exp 2. rectangular grid

- Rotation (0, 30, 45, 60, 90 degrees clockwise rotation)

  Exp 3. no grid

  Exp 4. rectangular grid

- Rectangular fisheye (0, 0.5, 1, 2, 3 transformation factor)

  Exp 5. no grid

  Exp 6. rectangular grid

  Exp 7. polar grid

- Polar fisheye (0, 0.5, 1, 2, 3 transformation factor)

  Exp 8. no grid

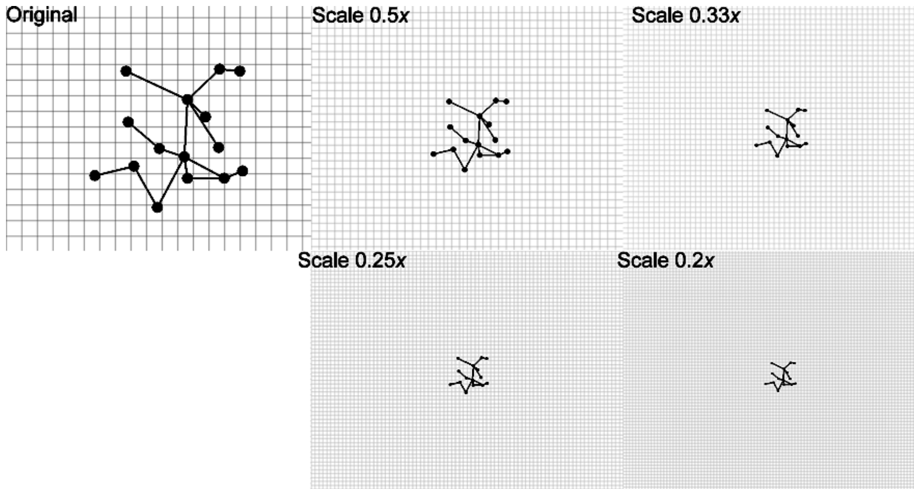


Figure 5.1: Sample stimuli for expt 2, the scaling transformation with rectangular background grids, darkened for printing purposes. Expt 1 used similar stimuli without background grids.

Exp 9. rectangular grid

Exp 10. polar grid

The choice of transformation ranges was based on two considerations. For scaling, there was a limit to which we could reduce stimuli size without severely compromising perceivable detail. Otherwise, we used pilot results to determine the start of performance degradation induced by the transformations. Based on our results, we extended two of the experiments: (1) experiment 4-ext: rotation with a rectangular-grid to study a wider range of rotations: 0, 90, 120, 150, 180, and (2) experiment 10-ext: polar fisheye with a polar grid to study the effects of transforming the sizes of the dot, and drawing the connecting lines in various coordinate systems. We did not include the translation transformation as it had previously been found to be robust in visual search tasks to at least 2 degrees of visual angle (Rensink 2004).

5.1.2 Stimuli

All experimental stimuli were randomly generated abstract images consisting of dots connected by lines. We chose to use abstract rather than photorealistic images in part to avoid semantic effects, such as the verbal effect found by Goldstein and Chance (1971), where recognition accuracy was considerably lower for

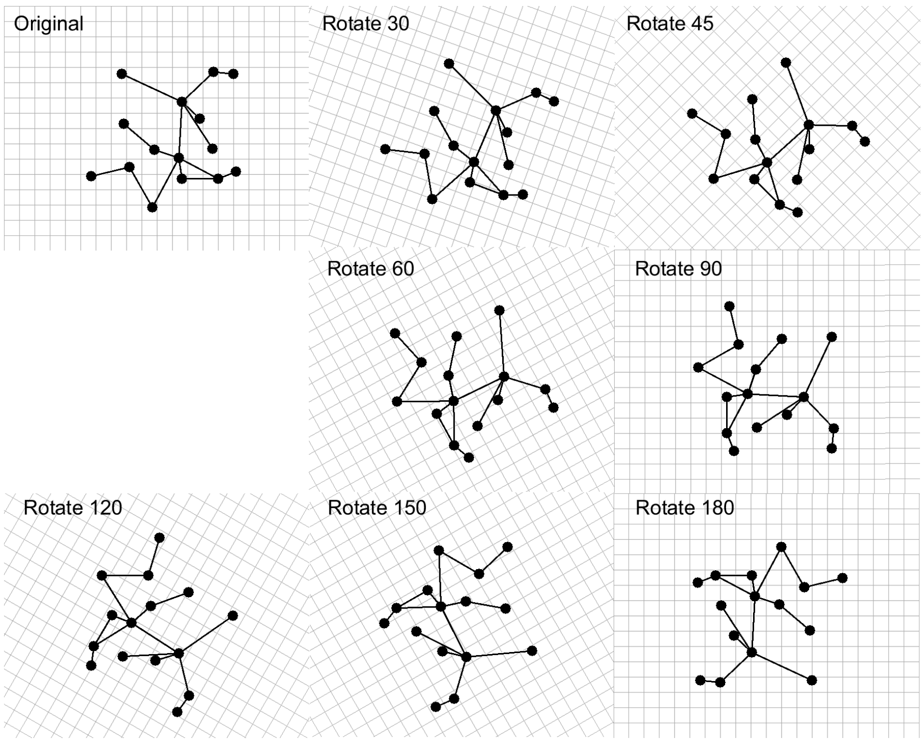


Figure 5.2: Sample stimuli for expt 4 and 4-ext, the rotation transformation with rectangular background grids, darkened for printing purposes. Expt 3 used similar stimuli for up to 90 degrees rotation without background grids.

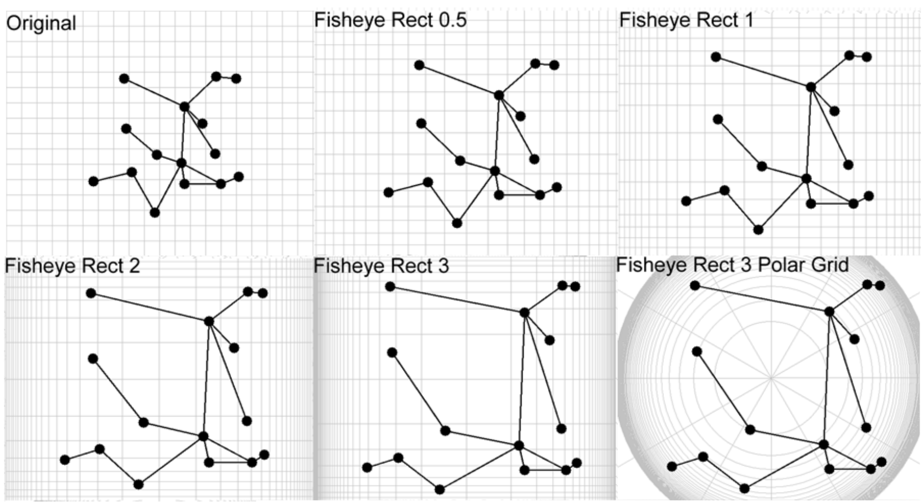


Figure 5.3: Sample stimuli for rectangular-grid fisheye rectangular experiment (expt 6), along with the maximally distorted image for the polar-grid variety (expt 7). Expt 5 used similar stimuli without background grids. Grids have been darkened for printing purposes.

objects difficult to name. Moreover, in the domain of information visualization, data is typically represented in abstract form. Our stimuli were similar to two-dimensional network graphs, but we believe these results generalize to many different encodings of information.

All original stimuli had a resolution of 300x300 pixels to ensure that all levels of transformations would fit onto the display screen. In the grid experiments, we filled the entire screen with the corresponding grid. We used a different set of images for each experiment, but the same experimental set for all participants. All images were generated in the same manner for consistency. Each consisted of 15 dots connected by lines. The number of dots was determined in pilot studies to optimize image memorability. The locations of the dots were randomly generated. The algorithm only guaranteed non-collision but not constant density of the dots.

Pilot studies showed that the task was too difficult if we only provided the dots. Lines were therefore added to link the dots to enhance stimuli memorability, similar to lines drawn between stars in astronomical constellations. The algorithm that added the lines did not guarantee that all the dots were joined as a single unit, but it did ensure all of the dots were connected to at least one other dot, namely, its nearest neighbour. The algorithm minimized line crossing, but did not control the number of topological features, for example loops.

When grids were added to the images, the thickness of the connecting lines was increased to two pixels to better distinguish the dot-line foreground from the grid background.

For the fisheye transformation experiments, we used a transformation function, taken from Leung and Apperley (1994):

$$T(x) = \frac{(d+1)x}{(dx+1)} \tag{5.1}$$

where $T(x)$ is the transformed value given input x, and d is the transformation factor. A larger d value leads to a higher degree of distortion.

Figures 5.1 to 5.4 show a series of stimuli showing all the transformation types and levels. Additional stimuli used in the experiment are included in Appendix C.3.

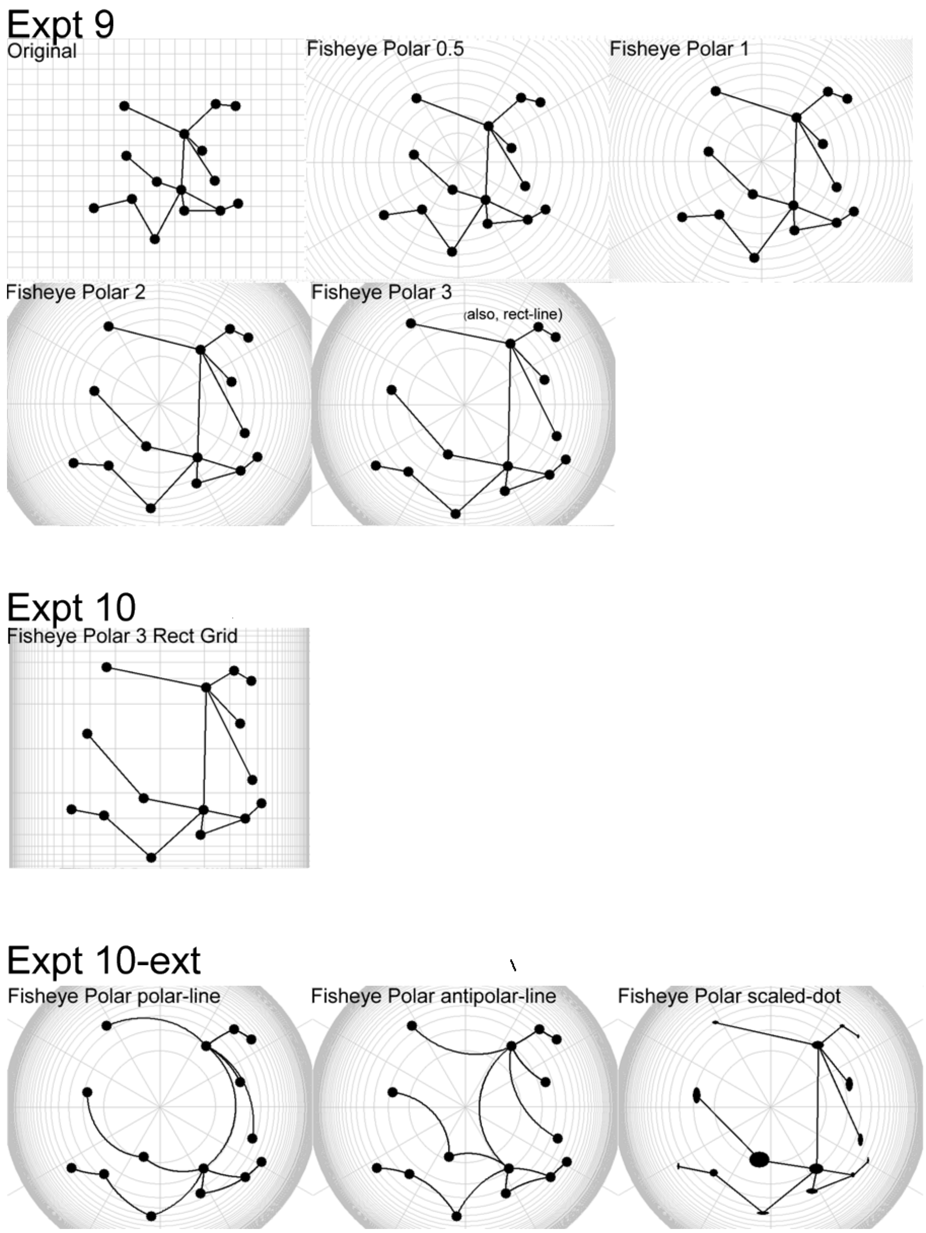


Figure 5.4: Polar fisheye transformations sample stimuli. First two rows show sample stimuli for the polar fisheye transformation with rectangular-grid in expt 9. Expt 8 used similar stimuli without background grids. The third row shows one example stimulus used in expt 10, transformed at the maximum transformation factor with a polar-grid. Stimuli in the last row were used for expt 10-ext. Grids have been darkened for printing purposes.

5.1.3 Participants

A different group of 20 participants was tested in each of the 12 experiments. All were university students with normal or corrected-to-normal vision. Their ages ranged from 18 to 34 years.

5.1.4 Protocol

For each of the 12 experiments, all 20 participants completed trials on all five levels of the test transformation, and the order of appearance of the levels were partially counterbalanced among the participants. The actual presentation orders used are listed in Appendix C.4. Each experiment had a separate pool of stimuli. The stimulus was randomly selected from a pool of 50 and each only appeared once in the experiment for each participant to avoid learning effects, but the same pool was used for all participants in each experiment. Prior to the actual experiment, participants were shown samples of original and transformed images to help them understand the transformation.

Each transformation-level session consisted of two phases: learning and recognition. In the learning phase, participants were asked to study eight untransformed images; each was displayed for 12 seconds and followed by a 2.5-second blank screen before the next image appeared. Participants were told they would need to recognize those images later on in the experiment, and that some of these images might be transformed in a manner similar to sample images shown during the training session. In the recognition phase, eight transformed images were shown to participants in sequence. Half of these had been shown in the learning phase in their original form. The participants' task was therefore to indicate whether they had seen the images in the learning phase. Task instructions presented to participants are included in Appendix C.2.

Prior to the experiment, participants were trained on the task using untransformed images in both the learning and the recognition phase. They were required to obtain at least 80% accuracy before starting the actual study.

Each experiment typically took 30 minutes. Participants were compensated for their time with five dollars. Based on our pilot experience, in order to do well on the tasks, participants needed to pay close attention to the test images during the learning phase. As an added incentive, we informed participants that high-accuracy scores would result in additional five-dollar bonuses.

5.2 Data Analysis and Result Summaries

Experiment		**No-cost zone**		
		Time	**Accuracy**	**Combined**
1.	Scaling: no-grid	$\geq 0.2x$	$\geq 0.2x$	$\geq 0.2x$
2.	Scaling: rect-grid	$\geq 0.2x$	$\geq 0.2x$	$\geq 0.2x$
3.	Rotation: no-grid	45°	45°?	45°
4-ext.	Rotation: rect-grid	60°	60°	60°
5.	Rect Fisheye: no-grid	$d = 1$	$d = 1$	$d = 1$
6.	Rect Fisheye: rect-grid	$d = 2$	$d = 2$	$d = 2$
7.	Rect Fisheye: polar-grid	$d = 2$?	$d = 2$	$d = 2$
8.	Polar Fisheye: no-grid	$d = 1$?	$d = 1$	$d = 1$
9.	Polar Fisheye: rect-grid	$d = 2$	$d = 2$	$d = 2$
10.	Polar Fisheye: polar-grid	$d = 2$?	$d = 2$?	$d = 2$

Table 5.1: Summary of experimental results: no-cost zones. A no-cost zone is the largest degree of transformation that can be compensated for without incurring a cost in performance. The combined result is the minimum of the time and accuracy results. Note that results from expt 10-ext are not included since they are inconclusive.

Experiment		**Tx Level**	**Performance Cost**	
			Time(s)	**Accuracy(%)**
1.	Scaling: no-grid	none	none	none
2.	Scaling: rect-grid	none	none	none
3.	Rotation: no-grid	60°	5.4 (3.4)	*69 (88)*
4-ext.	Rotation: rect-grid	90°	5.9 (4.1)	75 (88)
5.	Rect Fisheye: no-grid	$d = 2$	5.2 (4.6)	50 (88)
6.	Rect Fisheye: rect-grid	$d = 3$	3.9 (2.8)	75 (88)
7.	Rect Fisheye: polar-grid	$d = 3$	5.5 (3.5)	75 (94)
8.	Polar Fisheye: no-grid	$d = 2$	4.7 (3.7)	75 (94)
9.	Polar Fisheye: rect-grid	$d = 3$	5.6 (3.5)	75 (88)
10.	Polar Fisheye: polar-grid	$d = 3$	5.6 (3.8)	75 (88)

Table 5.2: Summary of experimental results: performance cost at the transformation levels just outside the no-cost zones, as shown in the Tx Level column. Baseline values are in parentheses for comparison. Italicized results are cases where the boundaries were estimated based on observed trends instead of statistical analyses. Note that results from expt 10-ext are not included since they are inconclusive.

We recorded two performance measures: response time and accuracy. Response time was defined as the period from which the image was shown during the recognition phase, to the time when a response was made. Accuracy was the percentage of answers that correctly identified whether the images had been shown in the learning phase. Blind guessing would lead to 50% accuracy, since half of the images shown in the recognition phase were present in the learning phase.

For the analysis of response times, we used a repeated measure single-factor Analysis of Variance (ANOVA) with transformation type as the factor for each experiment. We used the Greenhouse-Geisser adjustment and marked the results as adjusted if the sphericity assumptions were violated. Post-hoc analyses were performed for statistically significant results with Bonferroni correction and marked as corrected. For the accuracy results, we used the Friedman test for the initial analyses, and the Mann-Whitney test for post-hoc analyses. Only significant results are reported for the post-hoc analyses.

For each experiment, we mapped out a no-cost zone beyond which the performance began to degrade, as indicated by measurably higher response times and lower accuracy rates when compared to performance on untransformed images based on statistical analyses. Limitations of our no-cost zone definition are further discussed in Section 5.5.

Due to the large number of experiments, we summarized our results in Table 5.1. For cases where boundaries were not established by statistical analyses, we provided estimates based on result trends and marked them by a '?'. Table 5.2 lists the results immediately outside of the identified no-cost zones. Corresponding baseline values were provided in parentheses for comparison.

As the tables indicate, visual memory was robust against many forms of transformations to a large extent. Scaling did not impact performance down to a reduction factor of at least $0.2x$. Rotation did not seem to affect performance up to 45 degrees and both fisheye transformations had little effect on time or accuracy up to $d = 1$. The presence of grids generally extended these boundaries.

5.3 Detailed Results and Statistics

We now provide the detailed experimental results and data analyses for each of the four transformation types. For readability, details of the ANOVA and the post-hoc analysis results are listed in Appendices C.5.1 and C.5.2 respectively.

5.3.1 Scaling transformation

Figure 5.5 shows the results. Results showed no significant differences between the five levels, with or without adding grids to the images: time/no-grid: F(2.3, 43.2) = 0.67, p = .54, adjusted; accuracy/no-grid: χ^2(4, N=20) = 2.01; time/rect-grid: F(4, 76) = .60, p = .67; accuracy/rect-grid: χ^2(4, N=20) = 3.15, p = .53. Scaling over the ranges studied was not found to impact performance, and further reduction of stimuli would render them to small too discern details.

5.3.2 Rotation transformation

Figure 5.6 shows the results. For the no-grid experiment, we found a marginal main effect in response time (F(1.9, 35.8) = 2.92, p = .070). Post-hoc analysis indicated that performance degradation was measurable beginning at 60 degrees, at which participants took 5.4 s compared to the 3.4 s baseline. We also found a marginal main effect in accuracy (χ^2(4, N=20) = 8.75, p = .070) but could not identify a clear no-cost boundary.

For the rectangular-grid experiment, we failed to find a main effect in both time (F(2.6, 49.7) = 1.33; p = .27, adjusted) and accuracy (χ^2(4, N=20) = 7.16, p = .13), thus we were unable to locate no-cost zone boundaries based on these results.

Since we found relatively little performance degradation in the rectangular-grid results, we extended the range of rotation to cover 0, 90, 120, 150, and 180 degrees in expt 4-ext. In order to keep our five-level design, we did not revisit 30, 45, and 60 degree rotations in expt 4-ext, but we did include the 90-degree rotation condition as a reference point to compare with expt 4. The results are shown in Figure 5.6 as "Rectangular Grid Ext". In expt 4-ext, we obtained similar results for 0 and 90-degree conditions as in expt 4, albeit the 90-degree result was 8% higher numerically, but not significantly different. Unlike the case in expt 4, we found a main effect in response time (F(4, 76) = 5.05, p = .001) in expt 4-ext.

Post-hoc analysis indicated both the 90-degree and the 180-degree rotation trials were significantly slower at 5.9 s compared to the 4.1 s baseline. We also found a main effect in accuracy (χ^2(4, N=20) = 14.95, p = .005). Post-hoc analysis indicated the transformed trials were 14% less accurate than baseline. These results therefore suggested a no-cost boundary of 60 degrees. To determine the improvement provided by the rectangular grid, we compared the accuracy between the non-grid and grid trials from 30 to 90 degrees. Accu-

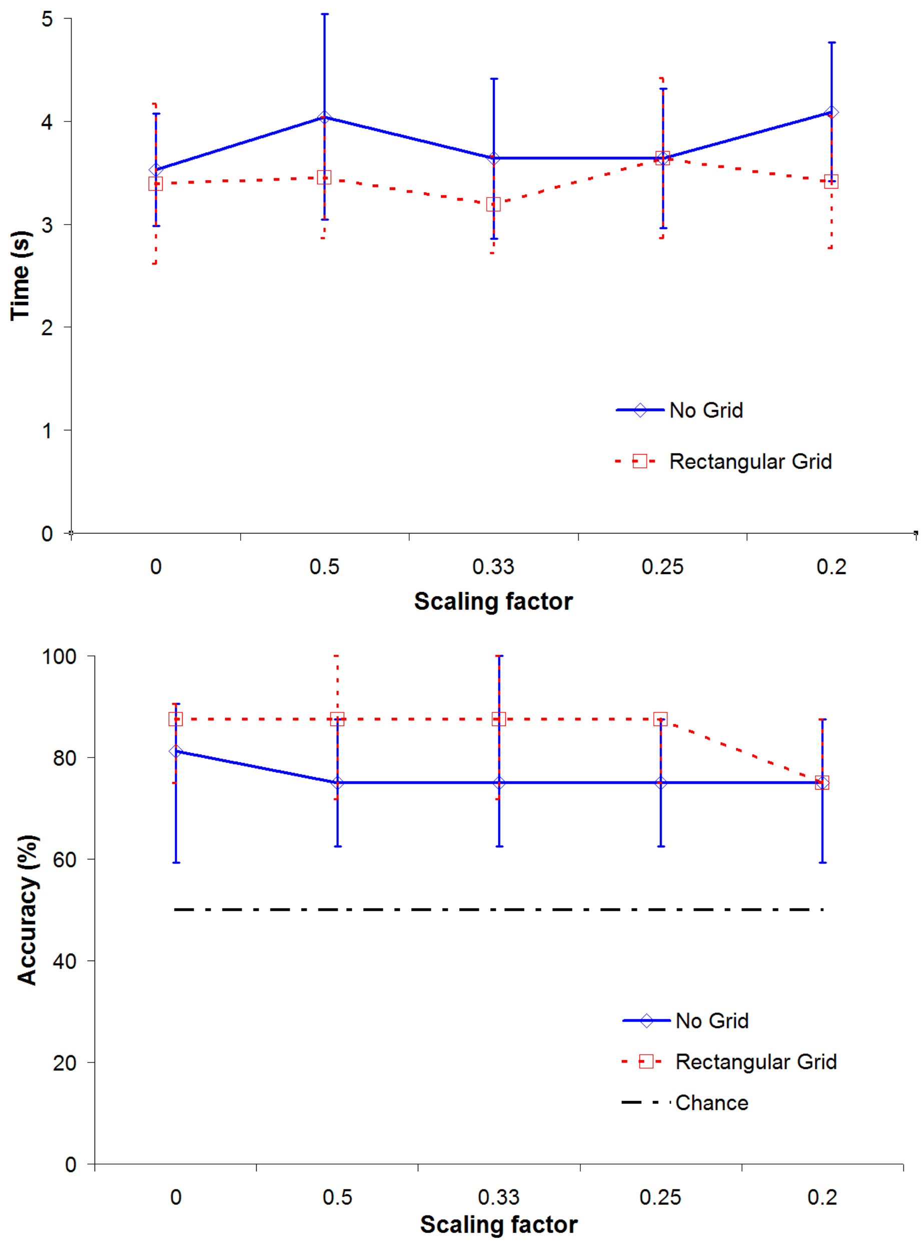


Figure 5.5: Results for the scaling experiments with N = 20. Response time data points are averages with 95% confidence interval bars. Accuracy results are medians with quartiles.

racy for the grid results were higher than their non-grid counterpart by 10% (two-tailed Mann Whitney test, p = .03). This increase in accuracy was not accompanied by an increase in time, thus ruling out any time-accuracy tradeoff.

5.3.3 Rectangular fisheye transformation

Figure 5.7 shows the results. For the no-grid experiment, we found a marginal main effect in response time (F(1.9, 36.2) = 2.83, p = .074, adjusted). It took 0.6 s longer for $d = 2$ and $d = 3$ trials than the 4.6 s baseline. We also found a main effect in accuracy (χ^2(4, N=20) = 43.80, $p < .001$) and the $d = 2$ and $d = 3$ trials were 33% less accurate than the rest of the trials. Using the one-sample z-test, we found that the accuracy for the $d = 2$ and $d = 3$ trials were at chance (Z(N=40) = 1.44; p = .15). These results indicated a clear no-cost zone boundary at $d = 1$.

For the rectangular-grid experiment, we found a marginal main effect in time (F(2.78, 52.9) = 2.63; p = .063, adjusted). Post-hoc analysis indicated that d = 3 trials were slower at 3.9 s when compared to the 2.8 s baseline, indicating a no-cost time boundary at $d = 2$. There was a strong effect in accuracy (χ^2(4, N=20) = 18.34, p = .001), with baseline and $d = 1$ trials being 15% more accurate than for $d = 3$, indicating a no-cost accuracy boundary at $d = 2$.

For the polar-grid experiment, the main effect in time was also marginal (F(4, 68) = 3.32; p = .051, adjusted), with a marginal time degradation at d = 3 (p = .077, corrected). While the task accuracy main effect remained, it was much smaller (χ^2(4, N=19) = 10.4, p = .034), with a no-cost accuracy boundary at $d = 2$.

5.3.4 Polar fisheye transformation

Figure 5.8 shows the results. We failed to find a main effect in time for the no-grid experiment (F(1.82, 34.5)=2.3; p = .12, adjusted). There was, however, a main effect in accuracy (χ^2(4, N=20) = 17.16, p = .002), with d = 2 and d = 3 trials being 20% less accurate than baseline, thus indicating a no-cost accuracy boundary at $d = 1$. A one-sample z-test analysis indicated that performance at d = 2 and d = 3 had not degraded to chance (Z(N=40) = 8.23; $p < .001$).

For the polar-grid experiment, we found a main effect in time (F(4, 76) = 6.08, p = < .001). Post-hoc analysis indicated $d = 3$ trials were 1.7 s slower than baseline and $d = 1$ trials, which took 4 s on average. This indicated a time

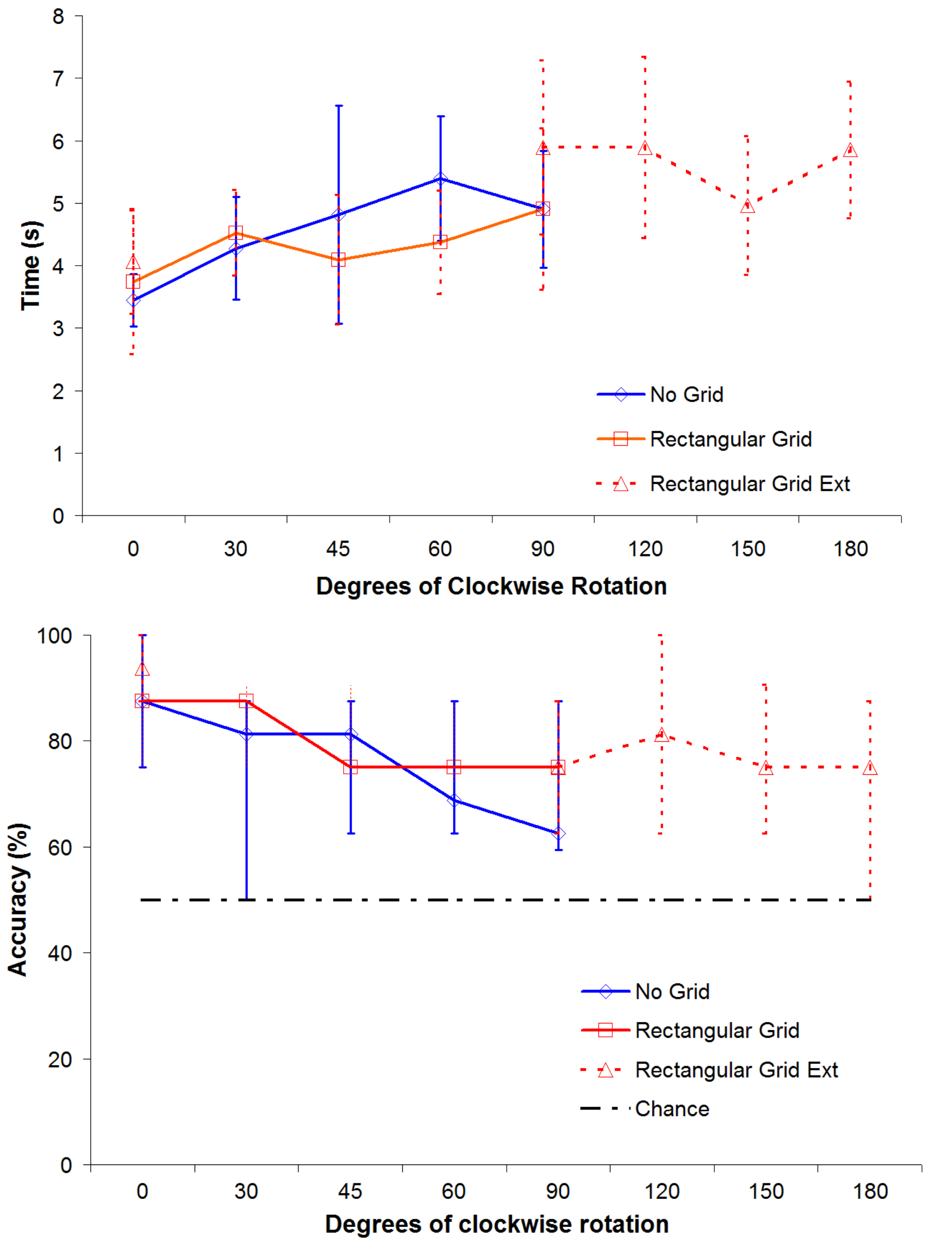


Figure 5.6: Results for the rotation experiments with N = 20. Response time data points are averages with 95% confidence interval bars. Accuracy results are medians with quartiles.

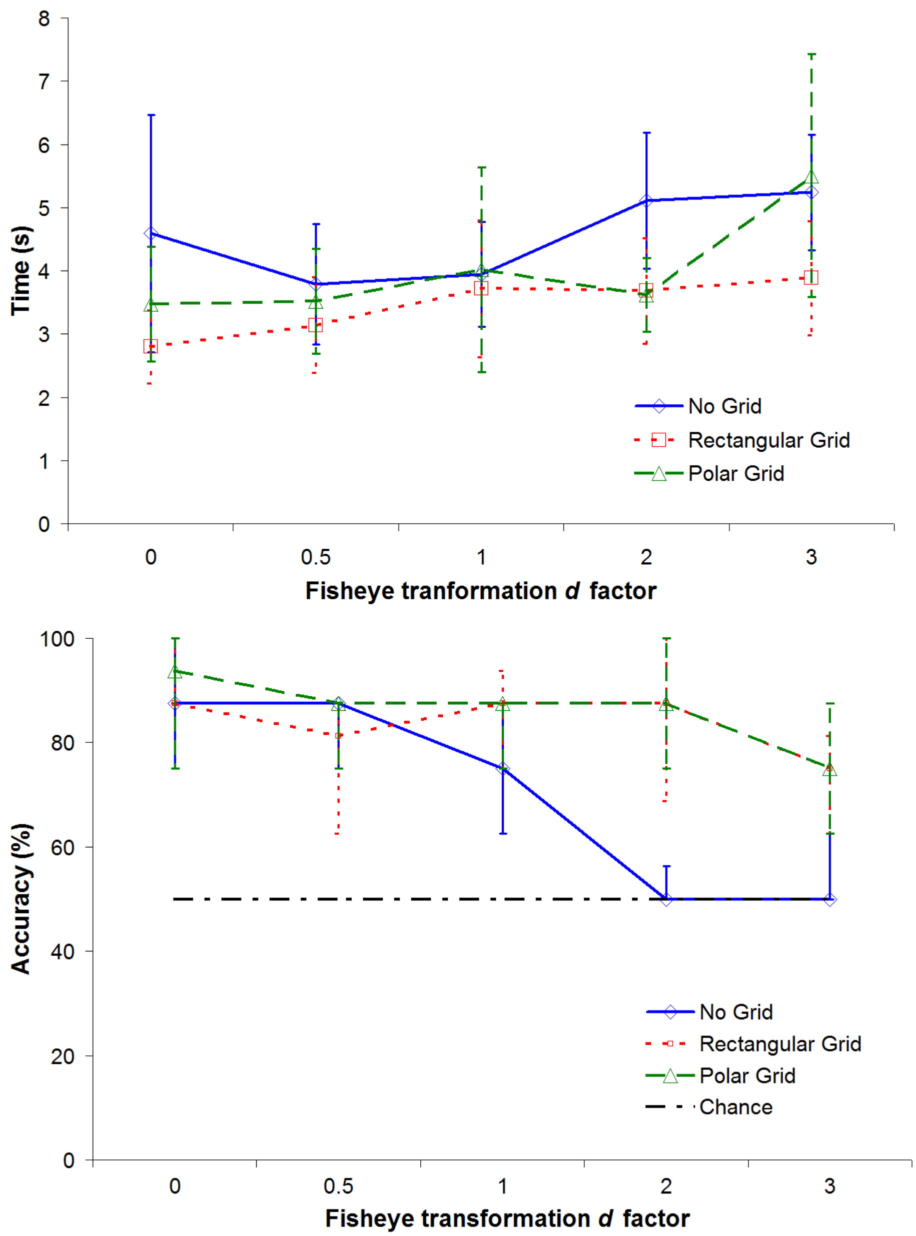


Figure 5.7: Results for the rectangular fisheye experiments with N = 20. Response time data points are averages with 95% confidence interval bars. Accuracy results are medians with quartiles.

no-cost zone boundary at $d = 2$. We failed to find a main effect in accuracy (χ^2(4, N=20) = 6.92, $p = .14$).

For the rectangular-grid experiment, we found a main effect in time (F(4, 76) = 4.32, $p = .003$). Post-hoc analysis indicated $d = 3$ trials were slower by 1.8 s than the 3.8 s baseline and $d = 1$ trials, thus indicating a no-cost time boundary at $d = 2$. We also found an accuracy main effect (χ^2(4, N=20) = 11.27, $p =$ .024). Post-hoc analysis indicated $d = 3$ trials were 12% less accurate than baseline, thus indicating a no-cost accuracy boundary at $d = 2$.

Despite extending the no-cost boundaries from $d = 1$ to 2, the presence of either polar or rectangular grids on polar fisheye transformed images did not substantially improve accuracy. This pattern was in stark contrast to that found in the rectangular fisheye experiments and suggests that there is something unusual about the polar fisheye transformation.

One possibility involves the shape of the lines connecting the dots. In experiment 10, the connecting lines were straight. If straight lines were less natural in the polar transformed images than in their rectangular counterparts, then this unnaturalness may have contributed to the lack of benefit of grids in the polar trials.

To test our hypothesis, we extended the polar fisheye experiment to look at line shape in experiment 10-ext, where the straight lines in the original images were drawn based on either a polar coordinate system (polar-line), a rectangular coordinate system (rect-line), or a mirror image of the ones drawn in the polar coordinate system (antipolar-line). The last case was included to tease out any potentially adverse effects induced by an unnatural transformation on the lines. Theoretically, transformation can be applied globally to the surrounding space, or locally to the objects in the space. In experiment 10, we assumed that space was transformed without affecting the sizes or shapes of the dots and the lines, as if they were pinned on the surface instead of completely adhered to the surface of transformation.

The only exception was in scaling, where we had to transform the dot size to avoid collision. To determine if this might account for the polar fisheye results, we also included a case where we transformed the size of the dots and keeping the lines in the rectangular coordinate system (scaled-dot) We failed to find a main effect in time (F(2.4, 45.5) = 2.09, $p = .13$), but did find a main effect in accuracy (χ^2(4, N=20) = 15.7, $p = .003$). Post-hoc analysis indicated that our participants made significantly more errors in the polar-line trials than base-line, and the accuracy was at chance (Z(N=20) = 1.45; $p = .15$).

Examples of these transformations are shown in the last row of Figure 5.4, and Figure 5.9 shows the results. In essence, the pattern found for the polar fisheye results does not appear to be due to the scaling of the dots, nor the shape of the lines connecting them. Instead, it appears to be that the polar fisheye transformation may simply be better suited to visual memory.

5.4 Discussion

Our results were used to map out no-cost zones in all the transformation types studied. We first compare our results to Lau et al.'s (2004) investigations, which were complementary to ours and studied visual search instead of visual memory. We then examine our results in the context of two design guidelines for using image transformations in interfaces: the use of background grids to mitigate perceptual costs incurred by image transformations (Zanella et al. 2002), and preserving horizontal/vertical ordering, proximity, and topology to minimize transformation-incurred disruptions (Misue et al. 1995).

5.4.1 Effects of image transformations

We compare our study result with those of Lau et al.'s (2004) investigations of perceptual costs in geometric transformations measured in visual search tasks to locate the figure "T" amongst a population of "L" figures. Both ours and Lau et al.'s (2004) study results suggested that invariance was possible for all geometric transformations for up to a point. Interestingly, this invariance appeared to be more extensive in recognition than search tasks. For example, search task performance degraded after a 50% reduction, while memory task performance remained unaffected even at 20% of the original size. Participants could also tolerate a larger degree in rotation (memory: 45°; search: 17°), and a larger amount of polar fisheye transformation (memory: $d = 1$; search: $d = 0.5$)[2].

While we applied the transformations to dot locations in most of our experiments, we found interesting results when we applied the polar fisheye transformation to dot sizes, and drew the connecting lines based on different coordinate systems. Contrary to our intuition, trials using images with lines drawn based on the polar coordinate system were the least accurate and equivalent to blind guessing, while corresponding trials with supposedly unnatural mirror images of

[2] The Lau et al. (2004) experiments used a different fisheye polar transformation function with a transformation factor c. A c value of 1.2 can be roughly translated to our $d = 0.5$.

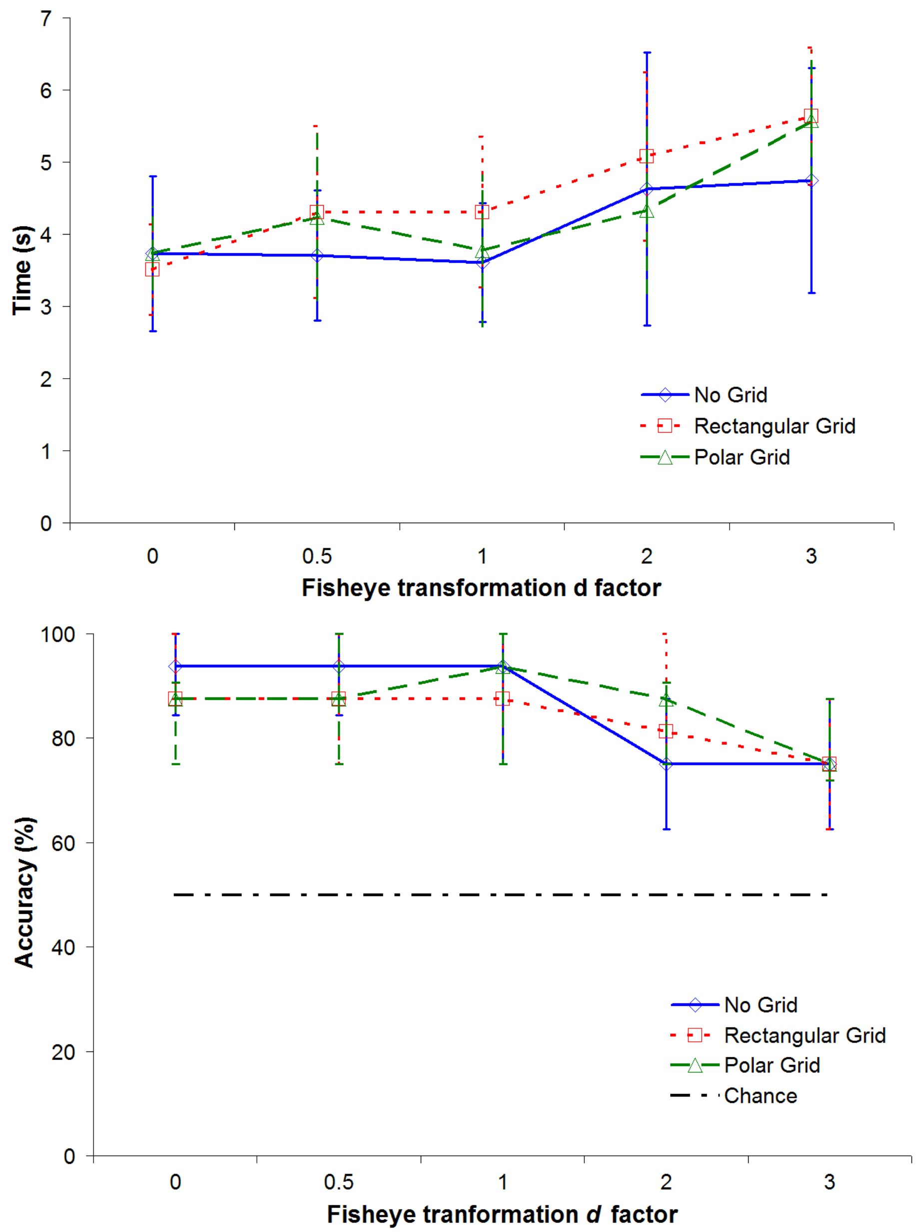


Figure 5.8: Results for the polar fisheye experiments with N = 20. Time data points are averages with 95% confidence interval bars. Accuracy results are medians with quartiles.

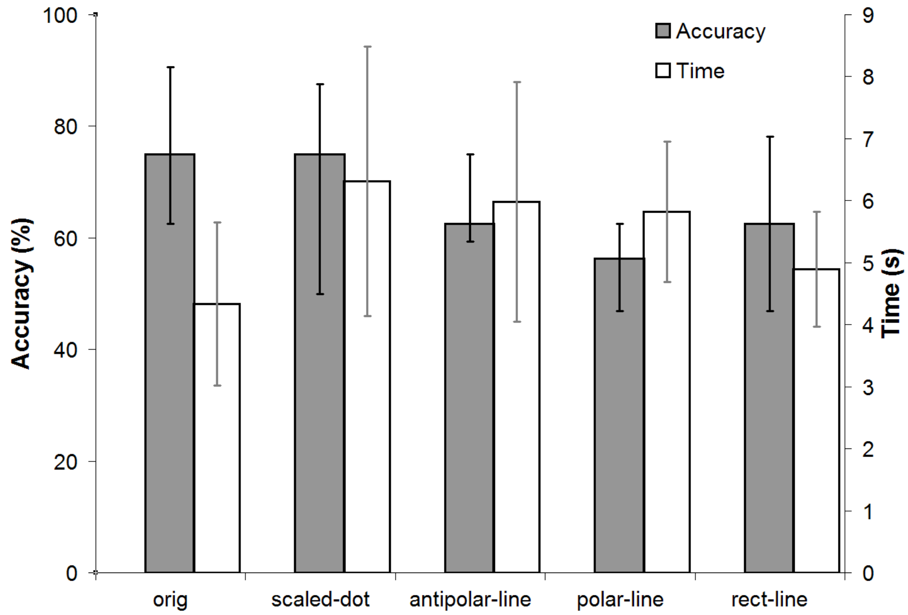


Figure 5.9: Results for the extended polar fisheye experiment with N = 20. Time data points are averages with 95% confidence interval bars. Accuracy results are medians with quartiles. orig = original image; scaled-dot = dot sizes transformed; antipolar-line = lines drawn as the mirror image in the polar coordinate system; polar-line = lines drawn in the polar coordinate system; rect-line = lines drawn in the rectangular coordinate system.

these lines exhibited better performance. These results suggest that distinctive local structure, rather than global consistency, was a more important factor in memorability. At large distortions, the lines in the polar-line images formed similarly rounded shapes, while corresponding antipolar-line images produced figures with enough acute angles to remain distinguishable, despite their blatant incongruity with the underlying transformation and with the coordinate system.

5.4.2 Effects of grids

In the design guidelines listed at the beginning of this chapter, Zanella et al. (2002) suggested using background grids to mitigate perceptual costs incurred by image transformations. We found that for visual memory, adding grids to the images appeared to help in two ways:

1. *No-cost zone extension.* The presence of either rectangular or polar grid

generally pushed the no-cost zone boundaries to higher levels. For example, the combined no-cost zone boundary for the fisheye transformations were increased from $d = 1$ to $d = 2$, and the rotation boundary was pushed from 45 to 60 degrees.

2. *Accuracy improvement.* Grids were found to improve accuracy. For rotation, participants were 10% more accurate in grid trials without spending extra time in the task, thus ruling out potential time-accuracy tradeoffs. In the case of rectangular fisheye transformation, we found that participants' accuracy improved from chance to baseline at $d = 2$, and to 75% at $d = 3$, again without time compensation. Interestingly, we failed to observe substantial improvement by adding grids to polar fisheye transformed images. Here, the grids appeared to simply elevate response times slightly, echoing the results for visual search (Lau et al. 2004).

To understand the apparent lack of performance improvement in polar trials, and to obtain further insights to the different transformation types and their interactions with grids, we revisited the design guidelines described at the beginning of the chapter.

5.4.3 Revisiting design guidelines

Design guidelines discussed at the beginning of the chapter were based mostly on design experiences and were mostly abstract. In this section, we explain our results based on Misue et al.'s (1995) guidelines to provide concrete examples, and suggest refinement on preserving orthogonal ordering based on our results.

Misue et al. (1995) suggested that horizontal/vertical ordering, proximity, and topology should be maintained to minimize disruptions incurred by image transformations. Scaling preserves all three; the limit of this transformation seems to be how far can one reduce the image before the details can no longer be perceived. This finding is consistent with the common interface design practice of using scaled-down versions of images to represent full-resolution file contents, especially when the file content is visually salient, as in the cases of most image files and graphically intense web pages. Indeed, various forms of thumbnails have been suggested for small-screen devices to avoid the laborious reauthoring of desktop-sized web pages for small screens (Woodruff et al. 2001; Wobbrock et al. 2002).

The rotation transformation violates horizontal/vertical ordering but maintains proximity and topology. Interestingly, rectangular grids fail to improve performance starting at a 90-degree rotation. Since our images did not have a clear up-down axis, this limit may be due to our inability to recognize the main vertical axis and the up direction in the image. Having a rectangular grid may help re-orientation, but only if the information provided by the grid is unambiguous. For example, the grid looked the same for 0, 90 or 180-degree rotations, and similarly for 30 or 120-degree and 60 or 150-degree rotations. Taken together, our results suggest a refinement to Misue et al.'s (1995) guideline on maintaining orthogonal ordering: transformation should preserve an orthogonal relationship between principal axes with a clear up and down.

For both fisheye transformations, proximity is violated while preserving horizontal/vertical ordering and topology. In that case, the perceptual challenge is to discern the relative distance between objects in the image. The polar fisheye transformation seemed to be much better tolerated than its rectangular counterpart, as accuracy was maintained at 75% even outside the no-cost zone in the polar case while corresponding rectangular trials showed chance performance. This result was not expected, as the polar transformation's rounded appearance does not look natural on a rectangular screen (Leung and Apperley 1994); among other things, it bends horizontal and vertical lines. Nonetheless, the polar fisheye transformation is generally preferred over its rectangular counterpart in map applications, since the distortion may be perceived as consistent with the effect of distorting a planar map onto a hemisphere, and the transformation preserves the angle of the original image (Skopik and Brown 1992; Churcher et al. 1997). The polar fisheye transformation may also be more familiar than rectangular, as the effect resembles that produced by the ultra-wide angle fisheye lens used in photography.

The number of transformation parameters and their degree of integration may further explain the smaller degree of degradation observed in our polar fisheye trials. In the rectangular case, the width and height are transformed separately. Rectangles that are the same distance from the focus point may not have the same size and shape. Objects may thus be distorted with different aspect ratios based on their horizontal and vertical distances, which may impose a higher mental load (Bartram et al. 1995). In contrast, the polar fisheye transformation only distorts radial distances, and may not incur the same problem as the rectangular case.

This issue may also explain the different effects we observed in our fisheye transformation trials. In the rectangular fisheye trials, adding a polar or rectangular grid improved accuracy from chance to 75% without time compensation. In contrast, neither a rectangular nor a polar grid improved performance in the corresponding polar fisheye trials. One possibility is that the grid, rectangular or polar, provided a powerful visual cue encoding standard distances in transformed images that helped to offset the difficulty in distance estimation when the image was distorted, as in the rectangular fisheye case. Since distance transformation is integrated in polar fisheye transformations, distance estimation may not be as difficult as in the rectangular case, thus nullifying potential benefits brought about by adding a grid.

Visual cues may also be used to aid recognition of objects. Researchers have investigated how the boundary of a scene affects target location after learning (Hartly et al. 2004), and how view-point changes affect scene recognition (Christou et al. 2003).

Smooth animation is another technique believed to alleviate the disruptive effects of image transformations (Robertson et al. 1989; Bederson and Boltman 1999). Similar to previous work on visual search (Rensink 2004; Lau et al. 2004), our current results suggest that the visual system could compensate for relatively large jumps in transformations. Both visual search and visual memory have thus been ruled out as valid reasons for requiring smooth animation. Nevertheless, the need for such animation may arise from some other considerations, and so further investigations are needed before advocating relaxing that design guideline.

5.5 Limitations of Study

Our study is limited in two aspects: the definition of no-cost zones and the choice of geometric transformation types.

5.5.1 Definition of no-cost zones

The main motivation behind defining no-cost zone for each transformation type is to connect to and enable comparison with Lau et al.'s (2004) study results. Lau et al.'s (2004) study also investigated perceptual costs of geometric transformations but in visual search tasks, and is therefore complementary to our study. However, the definition of no-cost zone is limited. For this study, we defined the boundary of no-cost zone as the first level of transformation at which we could measure performance degradation. Our identified boundaries were therefore limited by at least two factors: our ability to find statistically significant performance differences between different levels in our experiment, and the number of levels we tested in our experiments. It can be argued that our inability to detect performance differences between experimental levels could be due to lack of experimental power instead of true absence of performance degradation, even though our marginally significant cases had medium to large effect sizes (Appendix C.5.1).

5.5.2 Transformation type

In this work we adopted the view that geometric transformations simply affected object locations within a space. An equally valid view is to consider transformation on the space itself and the objects embedded within it. That view corresponds to transforming dot sizes and line shapes in addition to dot locations, so visual cues providing more information about how the space has transformed could improve performance. We briefly studied this issue in our extended study on polar fisheye transformation in experiment 10-ext, where we looked at effects of transforming dot sizes and their connecting lines drawn in various coordinate systems. Our results suggest that memorability may depend more upon local image structure than on global consistency with the underlying transformation and coordinate system. Further investigations are needed to establish this conclusion more firmly.

Our experiments looked at how a single and uniform transformation affects visual memory. In real-life situations, images may transform by parts and independently. It would be interesting to compare our results with those obtained using multiple transformations on a single image. We suspect the perceptual limits for multiple transformations will be much smaller than those established in our current set of experiments.

We decided on a small number of dots in the stimuli to create an acceptable level of task difficulty, but scalability is of interest. It would be interesting to see if the total number of dots in the stimuli would impact visual memory in similar ways if the stimuli contain local features that are individually salient and memorable. Also, in most information visualization interfaces, the whole purpose of fisheye transformation is to create space for new information to be displayed, instead of creating a large empty space as in our fisheye stimuli (Figures 5.3 and 5.4). We suspect having new information added to the stimuli would further reduce no-cost zone boundaries defined in this study.

5.6 Summary of Results and Implications for Design

We examined the effects of four different types of transformations on visual memory: scaling, rotation, rectangular fisheye, and polar fisheye. We found no-cost zones in all of the transformation types that exceed those found in Lau et al.'s (2004) work on visual search. We also found substantial benefits in applying grids to images for all of our transformation types except for polar fisheye. Our work therefore quantified the limits of our visual memory in coping with geometric transformations, and validated the use of grids as a visual cue to aid recognition of images.

The main contributions of our study are to provide empirical evidence to verify, exemplify, and to refine design guidelines that were based mostly on design experience and are abstract. Two main design implications drawn from study results are a refinement on Misue et al.'s (1995) guideline on orthogonal ordering where we suggested providing a clear up-down direction indicator may be sufficient, and the effectiveness of background grids to mitigate visual memory costs in geometric transformations.

Even though this study systematically quantifies visual memory costs of two-dimensional geometric transformations, it is difficult to apply the no-cost zone boundary values directly to interface design. First, in addition to our study limitation in the correct identification of no-cost zone boundaries as discussed in Section 5.5.1, these boundaries were determined based on collected task completion time and accuracy only. It is therefore unclear if other costs such as cognitive load may play important roles in task performance. Second, the abstract task of image recognition is difficult to extend to real-life tasks, as we are not sure how visual recognition affects visualization use in real-world systems such as multiple-VIR interfaces, even though common sense informs us that it has to be an important factor. For example, it is unclear if recognition of a transformed image guarantees usability of the transformed image. In other words, it may not be the case where image recognition implies that the user can identify individual nodes, or the relationships between nodes, in a transformed image of a network. For these reasons, direct application of no-cost zone boundary values in design is difficult.

To study interfaces under more realistic and applicable situations, we modeled our next study using the experimental-simulation strategy frequently used in human-computer interaction and information visualization studies. Our study first identified perceptual requirements for effective overview use in pilot investigations, and then examined these effects in detail in the actual study.

Chapter 6

Experimental-Simulation Study: Overview Use in Multiple Visual Information Resolution Interfaces

The third study in this thesis looked at overview use in multiple visual information resolution (VIR) interfaces with fully-interactive interfaces, scenario-based tasks, and recorded detailed observations. The goal of the study is to understand overview use in multiple-VIR display. More specifically, we studied perceptual requirements of overview graphical objects that permitted users to select areas of interest for further examinations in high-VIR displays, and examined how different spatial arrangements of the VIRs can support overview use.

We studied four interfaces: low VIR, high VIR, and two multiple-VIR interfaces where high and low VIRs were available in separate regions or embedded together. Our study data were unordered collections of line graphs synthetically created for specific visual characteristics at low and high VIRs. At low VIRs, we used colour encoding of the y-dimension to create a heatmap-like strip. At high VIRs, we used height coding in conjunction with colour for y-values to show a more traditional plot. To better study interface preferences, our participants could use any combination of VIRs in the multiple-VIR interfaces.

We found that in cases where our two perceptual requirements, visual simplicity and narrow visual span, were not met, participants using our multiple-VIR interfaces did not obtain better time and accuracy performance over the high-VIR interface, even when the multiple-VIR interfaces offered obvious benefits such as visually associating detailed plots with strips in a complex-target matching task, or side-by-side display in a visual comparison task. In fact, we were intrigued to find that at least 20% of participants chose to forego these benefits and devoted the entire interface to the high-VIR display. We conjecture that our results reflect the high interaction costs of multiple-VIR interfaces and the surprisingly stringent target visual requirements to enable effective overview use in multiple-VIR interfaces.

6.1 User Study Design

We took a different approach in this study since we had considerable difficulty in extending the visual-memory experiment results to design. While perceptual studies collect important information about our visual system, building results obtained from abstract tasks with static images into design guidelines for visualization is a challenging and long process, especially when visualization use is complex and dynamic, and our understanding of human vision, memory and cognition is still incomplete. Also, the complexity of both our visual system and visualization use make it difficult to isolate and identify factors to build models of interface use, and perceptual studies are not optimal in discovering new factors.

In designing our experimental-simulation study, we took what we believed to be the strengths of our visual-memory experiment and perceptual studies in general: rigorous experimental design with established protocols and tasks. We therefore took care to develop study tasks based on published task taxonomies (Section 6.1.1), used synthetic data to control visual features (Section 6.1.2), and used comparable visual elements to encode the data (Section 6.1.3). To better observe true interface use, we modified standard study design by allowing our participants to decide on interface use: our interfaces provided a simple mechanism to switch between interface modes, and our participants could use either single-VIR mode in all of the multiple-VIR trials. This study design choice resulted in interesting insights into interface use.

We studied four interfaces: two single-VIR (*LoVIR*, *HiVIR*) as comparison baselines, and two multiple-VIR (*Embedded*, *Separate*). We had four visual search and compare tasks, and collected three types of data: performance measurements as time and error rates; detailed observations of participant behaviours and strategies; and participant feedback from subjective questionnaires.

6.1.1 Study tasks

We developed four study tasks based on operations in published taxonomies for task diversity and generalizability, such as locate, identify, compare, associate, distinguish, rank, cluster, correlate, and categorize (Amar et al. 2005; Roth and Mattis 1990; Tory and Möller 2004; Wehrend and Lewis 1990). We used a scenario of monitoring and managing electric power in a control room to develop concrete examples of these abstract operations. We first piloted with 12 tasks listed in Table 6.1.

Based on pilot results and Tullis's (1985) work on display characteristics described in the Related Work chapter (Section 3.1.1), we identified two target characteristics that affected high- and low-VIR view use: visual complexity and visual span. Complexity referred to the number of peaks in the target, where simple targets had a single peak and complex ones had multiple peaks. Targets were considered local when they spanned less than 2 degrees of visual angle, or 2 cm of horizonal display width at a viewing distance of 55 cm. Otherwise, they were considered as dispersed.

Other display characteristics considered, but not further studied, include overall visual organization in the display, studied in question 8 and 10, and visual uniqueness of line graphs, studied in questions 1, 3, 5, 6, 7, and 11 in Table 6.1.

We selected four of the original twelve pilot tasks to address different aspects of these perceptual criteria, which were questions 2, 4, 9, and 12 in Table 6.1. In addition, visual instructions were also provided for participants to control for individual differences in visual analytical skills. Table 6.2 presents the task code names, and the domain instructions. Appendix D.2.2 contains all task instructions displayed on the study software.

Table 6.3 summarizes task characteristics based on the two study perceptual requirements.

6.1.2 Study data

We found in our pilot studies that data characteristics greatly influenced participant strategies. We therefore used synthetic data to ensure contrasting data characteristics, and developed tight criteria to create the study data collections.

Each data collection contains several data groups: original feature to match and search target, distractor, and background. To avoid target pop-out by

	Operation Class	**Task Instruction**
1	distinguish	Does location 107's power consumption profile differ from the rest of the locations monitored?
2	find extremum (vertical)	Which location has the highest power surge for the time period shown on the screen?
3	correlate	A fault occurred at 6:00, and resulted in a temporary power surge. Which location is affected the earliest?
4	find extremum (horizontal)	Which location has the most number of power surges?
5	compute derived value (standard deviation)	Which location has the most stable power consumption profile?
6	compute derived value (average)	Which location has the highest power consumption overall?
7	find anomalies	Identify the unique power consumption profile in this collection.
8	characterize distribution	The power stations are sorted by longitude of their location vertically, with top of the screen being the top of the country. Is the highest power consumers in the top, middle, or lower 3rd of the country?
9	correlate	A fault happened at location <x> at 6:00, causing a similar power surge in another location afterwards. Which one?
10	correlate + categorize	The following recording consists of recording of an entire year. Given power consumption increases with decreasing temperature, and Winter is the coldest season, which season do you think is at the beginning of the recording (Spring, Summer, Fall, Winter)?
11	filter + categorize	The recording is taken from a power control room in the UK during a football match on TV. It is known that UK citizens tend have a habit of making tea during breaks, thus causing power surges. Which location did not receive the broadcast?
12	compare	Find the power profile that is the same as that of location <x>.

Table 6.1: Instructions for the twelve pilot study tasks, along with their operations.

colour or position, and to control task difficulty and visual diversity, we created two distractor and five background populations, each containing 19 or 20 time series with characteristic patterns of peaks. Examples of these populations are given later on in the section so as to discuss them in the context of study tasks.

Domain Instruction	Visual Instruction
Max: Which location has the highest power surge for the time period shown on the screen?	Look for the brightest spot. You can mouse over and read the power off the tool-tip. Also notice the maximum power scale is shown above.
Most: Which location has the most number of power surges?	None needed.
Shape: A fault happened at location <x> at 6:00, causing a similar power surge in another location afterwards. Which one?	Look for a power surge of a similar shape as the one at location <x> at 6:00.
Compare: Find the power profile that is the same as that of location <x>.	All the profiles are exactly the same, except time-shifted by different amounts. The power surges of location <x> are in the middle of each column.

Table 6.2: Instructions for the four study tasks.

Task	Complexity	Span	Comparison
Max	simple	local	no
Most	complex	dispersed	no
Shape	complex	local	yes
Compare	simple	local	yes

Table 6.3: Summary of study task and data characteristics.

Each peak was created using a Gaussian function with a specified mean that translates to peak location, and variability that translates to peak width. The peak was scaled to the required height. In addition, we added a random noise of up to 2 pixels in absolute value to better mimic real-life data (Kincaid and Lam 2006). Figure 6.1 shows the targets and distractors for the *Max* task, Figure 6.2 for the *most* task; Figure 6.3 for the *Shape* task, and Figure 6.4 for the *Compare* task. The parameters used were determined based on pilot results.

For the *Max* task, the target peak was 10% higher, or 6% brighter on screen, than the distractor peaks and was at least 20% higher than the background peaks, as shown in Figure 6.1. For the *Most* task, the target line graph had six peaks of varying widths and heights at random x-positions, while the distractor line graphs had four peaks and the background graphs had three peaks or less, as shown in Figure 6.2. For the *Shape* task, the target and distractors were peak clusters of three narrow peaks with similar widths and different heights of low, medium and high. As shown in Figures 6.3 and 6.5, four configurations were created: (a) high-low-high, (b) low-high-low, (c) low-medium-high, and (d) high-medium-low. For the *Compare* task, all line graphs had the same three-peak configuration, but horizontally shifted by ±10, ±20, or ±30 pixels from the target, as shown in Figure 6.4. As a result, participants could use any of the peaks in the three-peak line graphs for comparison.

For each task, we generated a collection of 140 line graphs, each with 800 data points, for a total of 112,000 data points. These numbers were determined by the horizontal and vertical resolution of the display area, so that the entire collection could be visible without scrolling in *LoVIR*.

6.1.3 Interfaces

We used two visual elements to show xy-data, inspired by the Line Graph Explorer system (Kincaid and Lam 2006) that uses analogous but visibly different visual encodings for low- and high-VIR views. Both elements encoded the x-dimension in the same way, but their encodings of the y-data value differed:

1. Strip encoded the y-data with colour as a low-VIR strip of 6 pixels in height:

2. Plot doubly encoded the y-data with both colour and vertical spatial position as a high-VIR plot of 45 pixels in height:

Colour encoding was achieved by mapping y-value to saturation and brightness in the HSB space. To maximize line-graph detail perceivability, we mapped the normalized y-value y to saturation s and brightness level b using a sigmoidal function:

$$s = \frac{2}{1 + e^{-4(1-y)}} - 1; \; b = \frac{2}{1 + e^{-4y}} - 1 \tag{6.1}$$

Using these two visual elements, we built the four interfaces shown in Figure 6.1: (a) *LoVIR*, (b) *HiVIR*, (c) *Embedded* and (d) *Separate*. The display area for all the interfaces was 872 x 880 pixels. *LoVIR* showed the data collection using only the strips, while the *HiVIR* interface displayed only the plots.

Both *Embedded* and *Separate* provided strips and plots, showing only strips initially. In *Embedded*, left clicking on a strip added or removed a corresponding plot directly below, with the pair bounded by a one-pixel perimeter box to visually reinforce the association.

In *Separate*, left clicking on a strip added or removed the corresponding plot in the bottom panel, and marked or unmarked both the strip and the plot with separate perimeter boxes. The lower-plot window automatically resized with newly added plots and took up screen area from the upper-strip window for up to half the screen height, after which the sizes of both windows remained constant, and the lower-plot window accommodated newly added plots with vertical scrolling. Users could inactivate the automatic panel resizing by manually dragging the panel divider. Dragging the panel all the way to the top or the bottom of the screen allowed users to manually transform *Separate* to either *HiVIR* or *LoVIR*.

All interfaces had a panel on the far left to display the strip/plot numbers as text strings for plots or as graphical bars for strips, as shown in Figures 6.6, D.1, and D.2. Positions and states of the number displays were linked with those of the corresponding strips/plots.

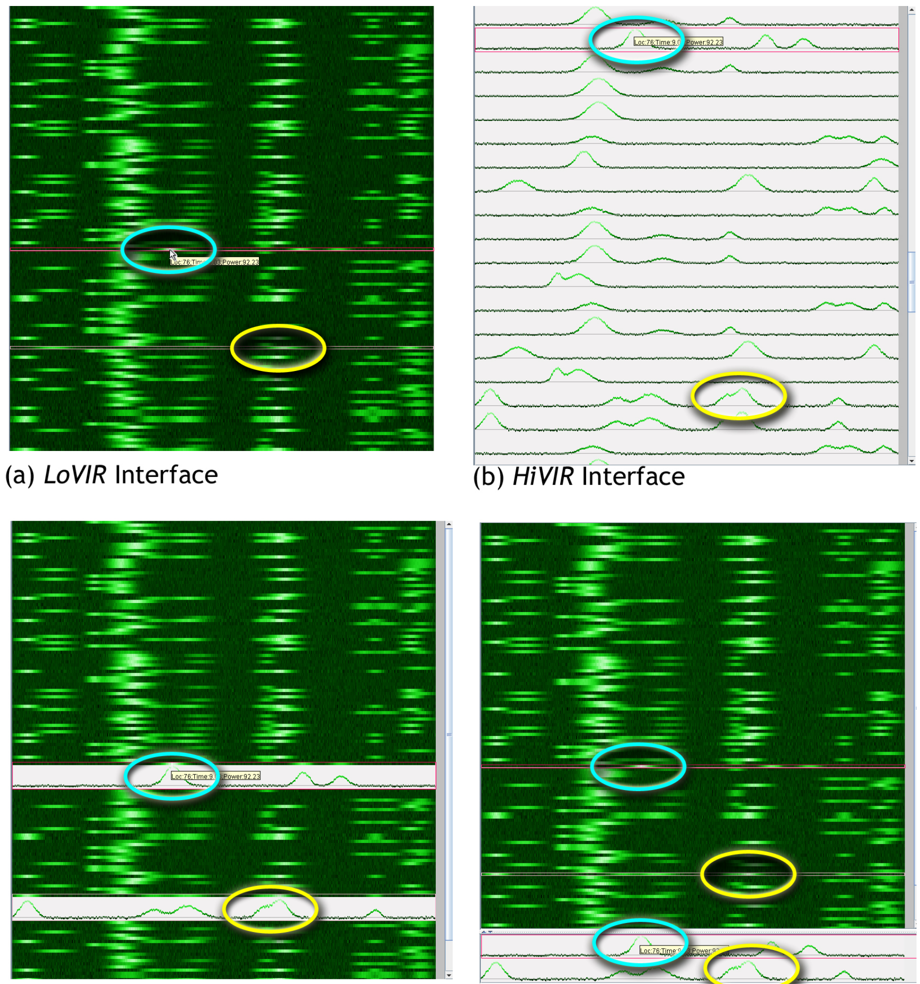
Common interactions

For consistency, we standardized a number of interactions, adding only slight interface-specific adaptations.

- *Scrolling.* A scrollbar supported vertical scrolling when display height exceeded panel height. *LoVIR* never required scrolling while *HiVIR* always did. *Embedded* and *Separate* became scrollable once a plot was added. Both the top and the bottom panels were separately scrollable in *Separate*. None of the interfaces required horizontal scrolling.

- *Mouse-click marking.* A left click toggle-marked a strip/plot. In *LoVIR*, *Embedded*, and *Separate*, the mark was a one-pixel box surrounding the entire strip in the low-VIR view. In *HiVIR*, we marked by coloring the plot background, because perimeter marking was not salient in the visually noisy plots. An example of highlighted plot background is shown in Figure 6.6 for plot 110.

- *Key-press global action.* For the single-VIR interfaces, participants could mark all strips/plots with the `O` key, and unmark them with the `ESC` key. For the multiple-VIR interfaces, pressing the `O` key added all plots to the

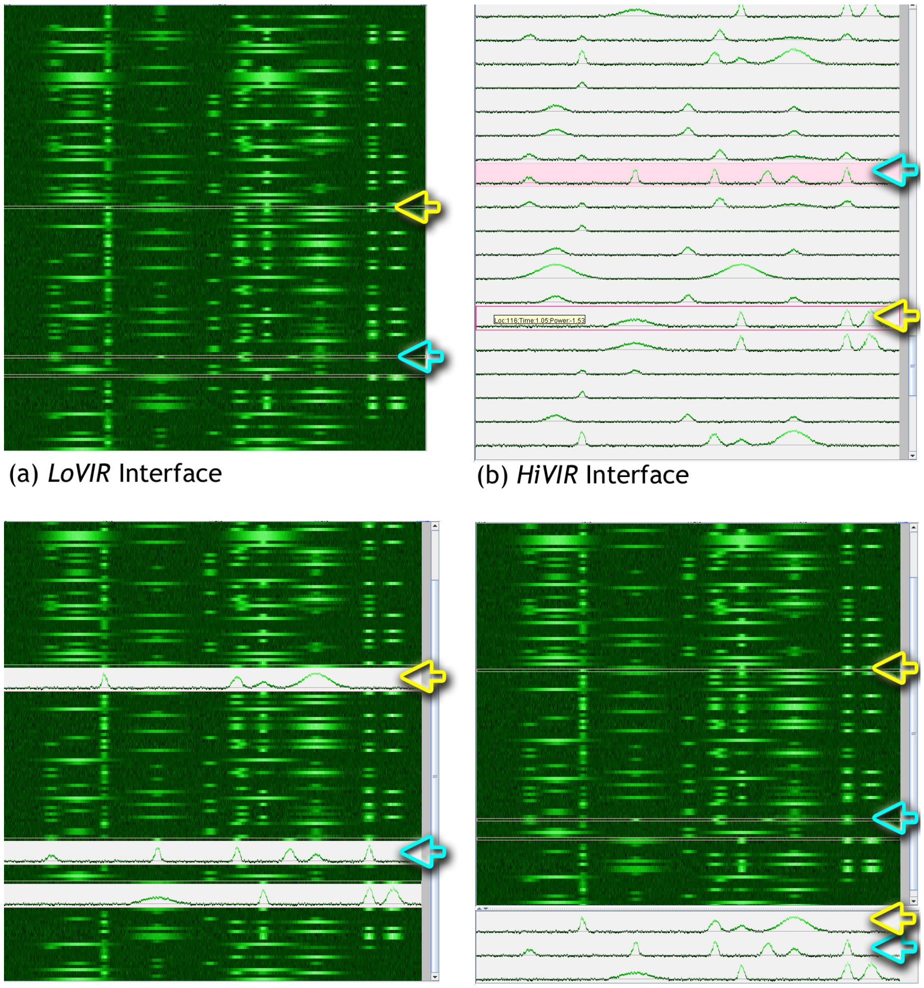


(a) *LoVIR* Interface (b) *HiVIR* Interface (c) *Embedded* Interface (d) *Separate* Interface

Figure 6.1: Main study panel showing *Max* task data: **(a)** *LoVIR*, **(b)** *HiVIR*, **(c)** *Embedded*, and **(d)** *Separate*. The targets are circled in cyan, and one of the distractors are circled in yellow.

high-VIR view in *Separate*, or opened all plots within *Embedded*. Pressing Esc restored the initial low-VIR view.

- *Mouseover highlighting.* For all the interfaces, a red one-pixel box appeared around the strip/plot perimeter on mouseover to provide visual feedback of the strip/plot in focus. In *Separate*, the strip-plot pair was highlighted for visual linking. Figures 6.1 and 6.6 show mouseover high-



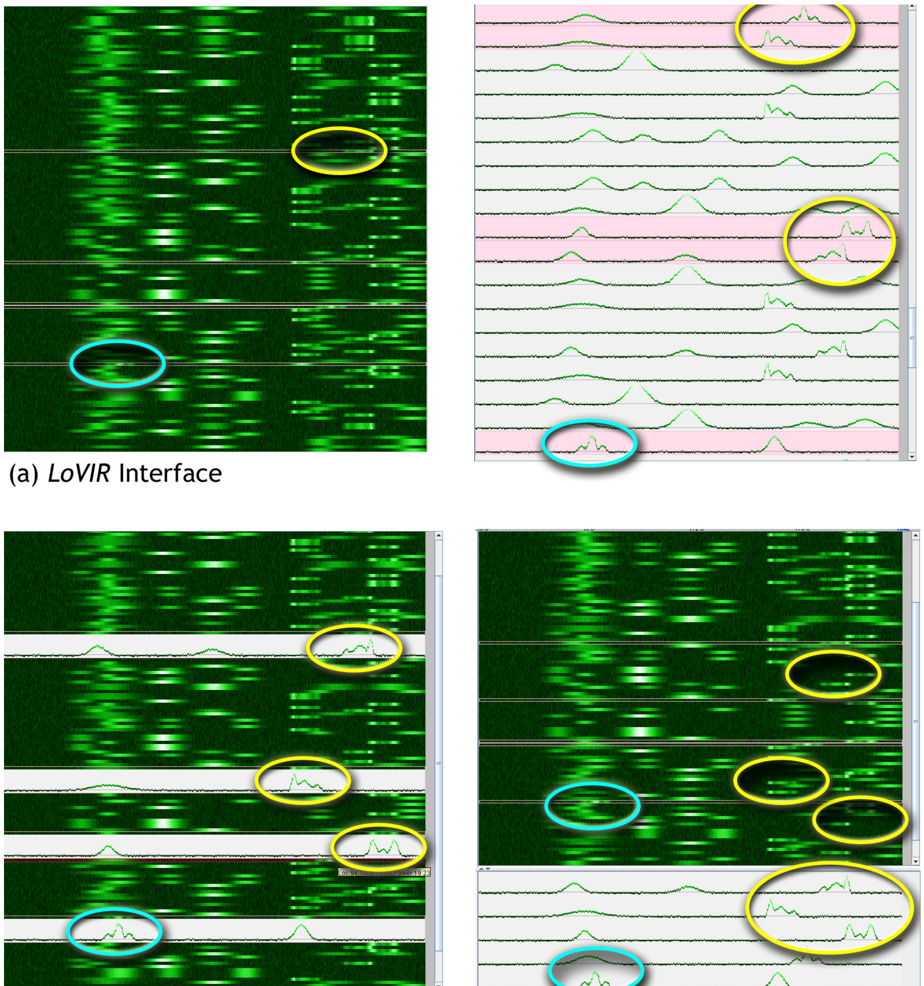
(a) *LoVIR* Interface (b) *HiVIR* Interface

(c) *Embedded* Interface (d) *Separate* Interface

Figure 6.2: Main study panel showing *Most* task data: **(a)** *LoVIR*, **(b)** *HiVIR*, **(c)** *Embedded*, and **(d)** *Separate*. The targets are annotated with arrows in cyan. The rest of the line graphs are distractors.

lighting.

- *Mouseover tool-tips.* For all the interfaces, mouseover triggered a tool-tip to immediately appear, displaying the x- and the y-value of the data point under the cursor and the strip/plot number of that row.



(a) *LoVIR* Interface

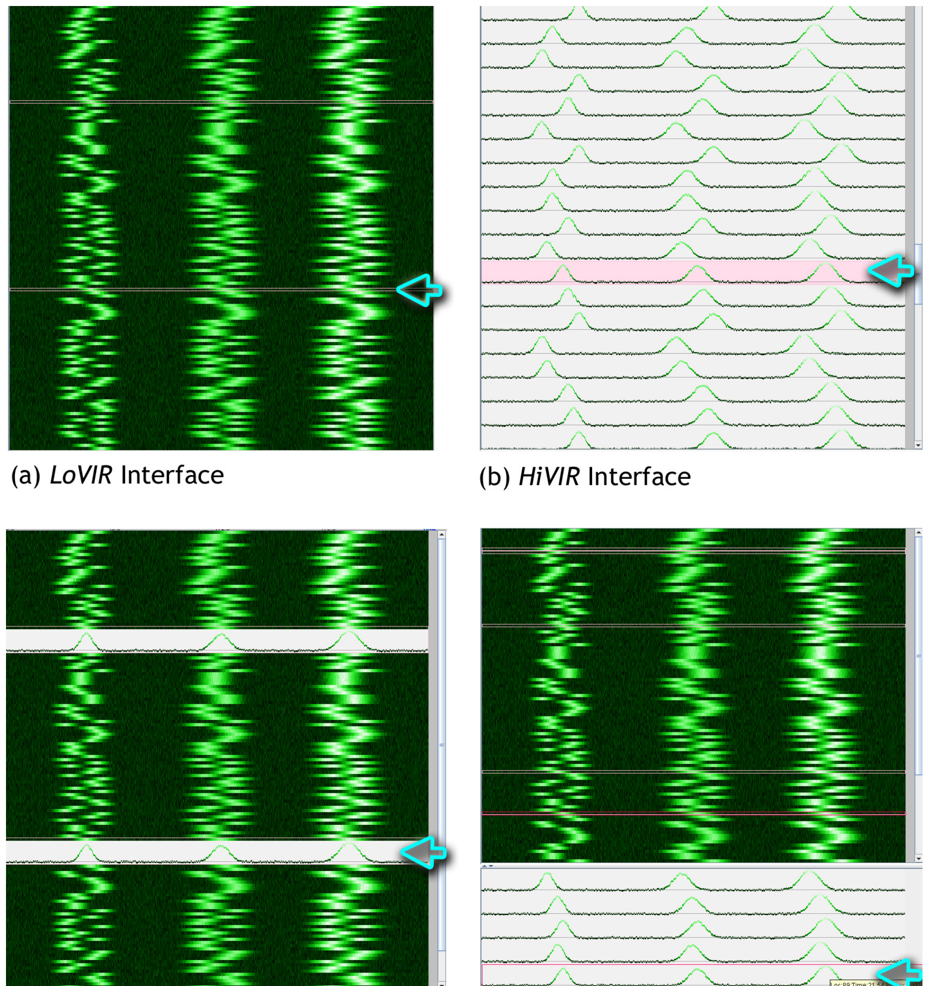
(c) *Embedded* Interface

(d) *Separate* Interface

Figure 6.3: Main study panel showing *Shape* task data: **(a)** *LoVIR*, **(b)** *HiVIR*, **(c)** *Embedded*, and **(d)** *Separate*. The targets are circled in cyan, and one of the distractors are circled in yellow.

6.1.4 Participants

24 participants, 15 of them female, were recruited using an online reservation system. The average age of participants was 26 years and ranged between 19 to 40 years. Most were university students, with less than half from the Department of Computer Science.



(a) *LoVIR* Interface

(b) *HiVIR* Interface

(c) *Embedded* Interface

(d) *Separate* Interface

Figure 6.4: Main study panel showing *Compare* task data: **(a)** *LoVIR*, **(b)** *HiVIR*, **(c)** *Embedded*, and **(d)** *Separate*. The targets are annotated with arrows in cyan. The rest of the line graphs are distractors.

6.1.5 Material

The study was conducted on a desktop machine with a 3.2GHz Intel P4 CPU, 1.5 GB of RAM, and Java 1.5.0_06, using a 19-inch LCD display with 1280 x 1024 pixels.

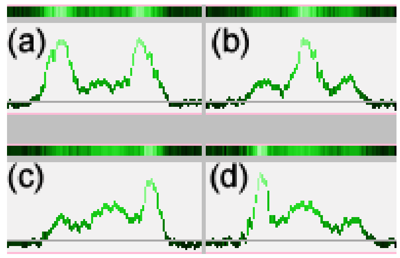


Figure 6.5: Sample targets for the *Shape* task. Four three-peak targets were created for the study: (a) high-low-high, (b) low-high-low, (c) low-medium-high, and (d) high-medium-low.

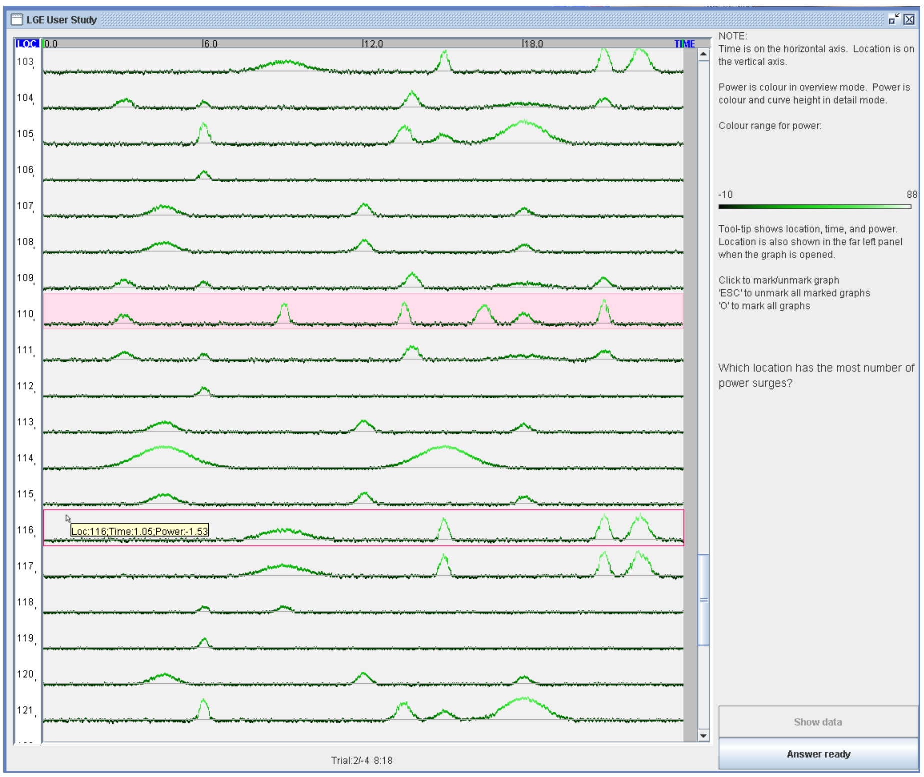


Figure 6.6: The *HiRes* study interface showing *Most* task data. The full display window had a narrow region on the far left with strip/plot numbers, and then a main panel in the middle whose contents depended on the interface. The far right panel contained study instructions: on top, information on visual encoding and available interface interactions; beneath that, task instructions, as provided in Table 6.2; on the bottom, the `Show Data` and `Answer Ready` buttons.

6.1.6 Study design and protocol

The study was a within-subject, two-factor design with interface and task being the two factors, each with four levels. All four interfaces were tested against the four tasks. Each task had four isomorphic data sets, one for each trial. The order of presentation of the interfaces was counter-balanced between participants. Task ordering was randomized, and data ordering was fixed to avoid repeats in interface/data pairing between participants.

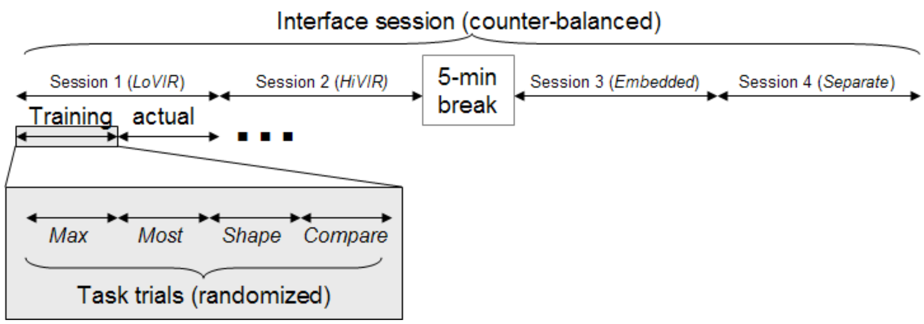


Figure 6.7: Experimental protocol for this study.

As depicted in Figure 6.7, the experiment consisted of four interface sessions, with one training and one actual task for each of the four interface/task combinations. Training for all four tasks preceded actual tasks for each interface session. The experimenter began by explaining the compact visual encoding used in the low-VIR views. Participants were then told about the structure of the study. They were encouraged to try out interface features and to explore new strategies for the different interfaces during training, as strategies developed for one interface might not be appropriate for another. Appendix D.2.1 contains the script for the verbal instructions. Since the correct answers were obvious when found, participants were not explicitly told to optimize for speed or accuracy.

The entire display window is shown in Figure 6.6 showing the *HiVIR* interface for the *Most* task. Figures D.1 and D.2 in Appendix D.2 shows the *Embedded* and the *Separate* interfaces respectively. For each task, participants first read the instructions in the right-hand panel of the study interface. When ready, they would press the `Show Data` button to display the data using the session interface. Once an answer was found, participants pressed the `Answer Ready` button to enter the answer in a dialogue box. Appendix D.2.2 contains all the instructions displayed in the experimental software interface.

For each interface/task combination, we allotted at least 10 minutes for participants to complete each training task. At the end of the 10 minutes, they had the option to end the training and be told the answer, or to continue the training. On average, participants took (3 ± 2) minutes to finish the training tasks, with similar averaged time over the four tasks. In terms of interfaces, the *Separate* training trials took four minutes on average, which was one to two minutes longer than the rest. Actual tasks had five-minute time limits, after which participants had to proceed to the next task without being informed of the correct answer. Breaks were allowed in between tasks, and there was a mandatory five-minute break after two interface sessions.

For each task, the experimenter observed participant mouse actions, verbal comments, and non-verbal signals including large-scale eye movement and signs of frustration. These observations were recorded as textual narrations in real time. For example: "Look for target in low-res. Press O to switch to high-res. Scan and scroll from top. Found answer, visual check without using tool-tip". We used these observations to help us interpret our performance time and accuracy results. We also developed a coding scheme for two kinds of usage behaviours:

- *Interface mode used to locate final answer.* The three categories were `LoVIR mode`, `HiVIR mode`, and `both`. The observation, only recorded for the two multiple-VIR interfaces, was later corroborated by the electronically recorded log of user actions.

- *Answer confirmation method.* The two categories were `visual comparison`, and `tool-tip/numeric confirmation`, differentiated based on back-and-forth tool-tip activations of the target and the candidate line graphs.

- *Visual search mode.* This observation was only collected for the *LoVIR* interface. The two categories were `serial search`, where the participant systematically inspected one strip at a time and in sequence, and `visual spotting`, where they surveyed the entire display simultaneously. Due to the narrow strips in *LoVIR*, serial search required the visual guide provided by the mouseover framing box, as shown in 6.1(a). For visual spotting, participants simply gazed at the display without any mouse interactions.

After the four interface sessions, participants filled out two questionnaires. The first questionnaire solicited subjective ratings of the four interfaces over the four tasks, while the second solicited the four interfaces' ease of use with a five-point rating scale (Table 6.4). The actual questionnaires used in the study are included in Appendix D.3. The entire study took about two hours, and participants were compensated with CDN $20.

Code	**Question**
find data	It is easy to find the data using the display.
compare data	It is easy to compare between data using the display.
navigate	It is easy to navigate within the display.
disorient	It is easy to get disoriented using the display.
remember	It is easy to remember individual power profiles using the display.
fun	It is easy to fun and enjoyable using the display.
effort	It requires a lot of effort to use the display.
frustrating	It is frustrating to use the display.
confidence	I have confidence in my answer when using the display.

Table 6.4: Five-point rating questions to solicit ease of use ratings for the four interfaces.

Study hypotheses

We developed three study hypotheses based on pilot observations and our beliefs about multiple-VIR interface use. H1 aimed to establish boundaries of our two selected perceptual requirements:

H1 The targets should be simple and span a limited region for a single low-VIR display to be usable. We believed that the *LoVIR* interface would be the most efficient for the *Max* task, where the visual target satisfied both criteria; insufficient but usable for *Shape* task, where the target was complex; and would be unusable for the *Most* task, where both criteria were violated.

In cases where the visual requirements were not completely satisfied, we hypothesized that selective display of high-VIR plots would mitigate the adverse effects of the lost perceivability, especially when the interface obviously supported the task. More specifically, our hypotheses were:

H2 When the targets were visually complex and could not be easily detected in the strips, embedded display of high VIR plots alongside the low-VIR strips would prime the search by promoting the learning of the unfamiliar and abstract strip.

In other words, the *Embedded* interface would better support the *Shape* task than the *HiVIR* or the *Separate* interfaces, as the *Embedded* interface put the corresponding plots right underneath the strips.

H3 When the targets were visually simple but similar to the distractors, precise identification of these targets using the low-VIR view would be difficult. However, users should still be able to select rough matches from the low-VIR view. The interface that displayed these potential matches in high VIR that allowed side-by-side comparisons would better support the task.

In other words, the *Separate* interface would better support the *Compare* task than the *HiVIR* or the *Embedded* interfaces.

6.1.7 Study design choices

As discussed at the beginning of the chapter, this study aimed to understand if users could still select regions of interests on low-VIR overviews that contained visual signals that did not fully comply with our two identified perceptual requirements: simple target with narrow visual span. Our goal of filling specific gaps in our understanding of multiple-VIR interface use led to eight main design choices.

1. *Synthetic data.* To create multiple isomorphic data sets with tight control over the visual characteristics of target, distractor and background graphs, we chose to generate synthetic data with real-world data characteristics.

2. *Unordered data.* While we used the visual encoding of the Line Graph Explorer system to build our interfaces (Kincaid and Lam 2006), we specifically avoided providing its sorting or clustering capabilities for two reasons. First, we wanted to focus on visual search and comparison based solely on visual qualities of individual targets, instead of the larger context. Pilot results showed that when the line graph collections as a whole showed larger trends, for instance clusters, the display was treated as a whole and participants did not selectively view individual line graphs in detail. Second, the power of reordering and clustering is already well understood (Kincaid and Lam 2006; Rao and Card 1994).

3. *Task domain and visual instructions.* To control for individual differences in visual analytical skills between participants, we provided specific

domain task instructions on control room monitoring and the visual operation on the encoded data. Our scenario provided a concrete unifying story, but did not require any specific expertise on the part of participants.

4. *On-the-fly interface switching.* To observe our participants' interface choices as another indicator of interface effectiveness, we allowed our participants to switch to either VIR of the multiple-VIR interfaces at any point, even though we provided an automatic mechanism to allocate screen space between the two VIRs.

5. *Only two discrete VIRs.* Some previous multiple-VIR interface studies have found that distortion-based interaction across a continuous range of VIRs can decrease performance and satisfaction (e.g., Nekrasovski et al. 2006). In this study, we choose to focus on the issue of spatial arrangement of separating low-VIR regions from, versus embedding them within, high-VIR regions. We thus used only two discrete VIRs, as in systems like TableLens (Rao and Card 1994), to avoid conflating the question of spatial arrangement with that of distortion. Distortion was studied in Chapter 5, where we measured visual memory costs of geometric transformations.

6. *Same platform and screen area across interfaces.* A common platform ensured consistent visual encoding, common interaction, and identical display areas.

7. *The full data set is simultaneously visible from the low-VIR interface to be used as an overview.* Our data set size was therefore limited to the display capability of the low-VIR view, which was 140 line graphs.

As a result of the last three design choices, vertical scrolling was needed when users chose to display plots.

6.2 Study Results

In this section, we present performance results for the actual tasks as time and error counts, coded observations, and subjective questionnaire results. We used the original interface grouping for all the results even when participants switched to single-mode use in the multiple-VIR interface trials. In a separate analysis, we did not find significant differences between the single-mode use and the multiple-mode use populations in the multiple-VIR interface trials.

Discussions of hypotheses are delayed to Section 6.3.

6.2.1 Performance time and error results

Performance time was defined as the period from which the participant pressed the `Show Data` button to the time when he pressed the `Answer Ready` button. We analyzed the time results using repeated measure two-factor Analysis of Variance (ANOVA) with interface and task as the two factors. When the sphericity assumption was violated, we used the Greenhouse-Geisser adjustment and marked the results as adjusted. Post-hoc analyses were performed with Bonferroni correction, and we report significant post-hoc results only.

Figure 6.8 shows the time results. Main effects of interface ($F(3, 69) = 5.97$, $p = .001$), task ($F(3, 69) = 34.45$, $p < .0001$), as well as interaction between the two ($F(9, 207) = 11.20$, $p < .0001$, adjusted) were found. For interface, post-hoc analysis indicated *LoVIR* trials were slower than *Embedded* or *Separate*. For task, all except the *Most* and the *Shape* results were different. For interface-task interaction, *HiVIR/Max* tasks were almost 3.5 times slower than the rest of the interfaces for the *Max* task. *LoVIR/Most* was almost 2 times slower than *HiVIR/Most*, *Embedded/Most*, and *Separate/Most*. *LoVIR/Shape* was 1.7 times slower than *HiVIR/Shape*, *Embedded/Shape*, and *Separate/Shape*.

Error measures were binary for each task: 1 when the participant provided an incorrect answer and 0 otherwise. We first analyzed the data using the Friedman test, and used the Mann-Whitney test with appropriate corrections for post-hoc analysis. We report significant results only. Figure 6.9 shows error results for each interface/task condition. Results showed that *LoVIR/Most* trials had 7 errors compared to the perfect scores of *HiVIR/Most* and *Embedded/Most*, and *LoVIR/Shape* trials had 6 errors compared to the perfect scores of *Embedded/Shape* and *Separate/Shape*. Along with the time results in Figure 6.8, we concluded that none of the interface/task results exhibited time-accuracy tradeoff: tasks that took longer also had more errors.

6.2.2 Observations

We quantified our observations by classifying each trial into one of the encoded categories. For multiple-VIR interfaces, we based our counts on the interface mode used at the time where participants found the answers, and the count results are shown in Table 6.5. For all the interfaces, the methods for answer confirmation are summarized in Table 6.6. For the *LoVIR* interface, the visual search modes used to locate the visual targets are shown in Table 6.7.

Task	**LoVIR**	**HiVIR**	**Both**	**LoVIR**	**HiVIR**	**Both**
		Separate			*Embedded*	
Max	14	0	10	9	0	15
Most	0	23	1	0	21	3
Shape	0	13	11	0	14	10
Compare	1	4	19	0	7	17

Table 6.5: Coded behaviour: Interface mode use for the two multiple-VIR interfaces.

Task	**Vis**	**Tip**	**Vis**	**Tip**	**Vis**	**Tip**	**Vis**	**Tip**
	LoVIR		*HiVIR*		*Separate*		*Embedded*	
Max	7	17	0	24	7	17	4	20
Most	24	0	24	0	24	0	24	0
Shape	18	6	19	5	22	2	19	5
Compare	3	21	2	22	13	11	2	22

Table 6.6: Coded behaviour: Answer confirmation mode: Vis = visual confirmation; Tip = Numeric read off tool-tips.

Task	**Search**	**Spot**	**Task**	**Search**	**Spot**
Max	2	22	*Most*	21	3
Shape	13	11	*Compare*	22	2

Table 6.7: Coded behaviour: Visual search mode use for the *LoVIR* interface: Search = Serial Search; Spot = Visual Spotting.

6.2.3 Subjective preference and questionnaire results

When asked to select the preferred interface overall, participants preferred both multiple-VIR interfaces over *LoVIR*, and *Separate* over *HiVIR* ($\chi^2(3, \text{N} = 14) = 15.00$, $p = .002$). None preferred *LoVIR*.

We also solicited two sets of subjective participant feedback with questionnaires. Results were first analyzed using the Friedman test, and the Mann-Whitney test was used for post-hoc analysis. The first questionnaire solicited subjective ratings of the four interfaces over the four tasks, as shown in Figure 6.10. To normalize the data, we divided the score for each interface by the sum of the scores for the task. Our results showed that *LoVIR* was preferred

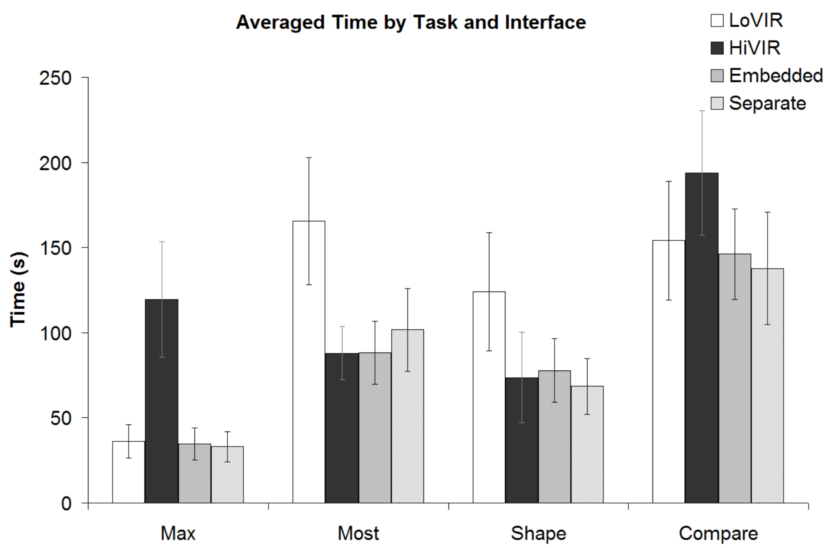


Figure 6.8: Averaged time results by task and interface. Error bars show 95% confidence level. (N = 24)

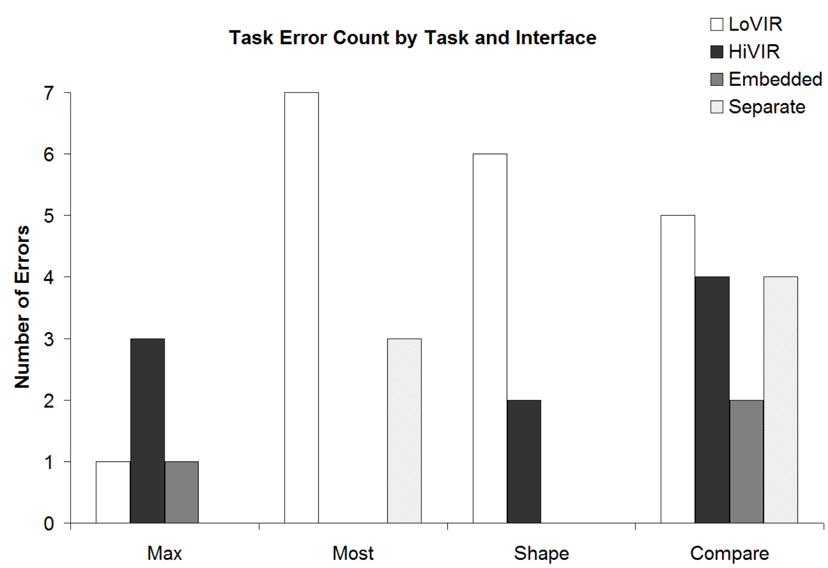


Figure 6.9: Total error results categorized by task and interface. (N = 24)

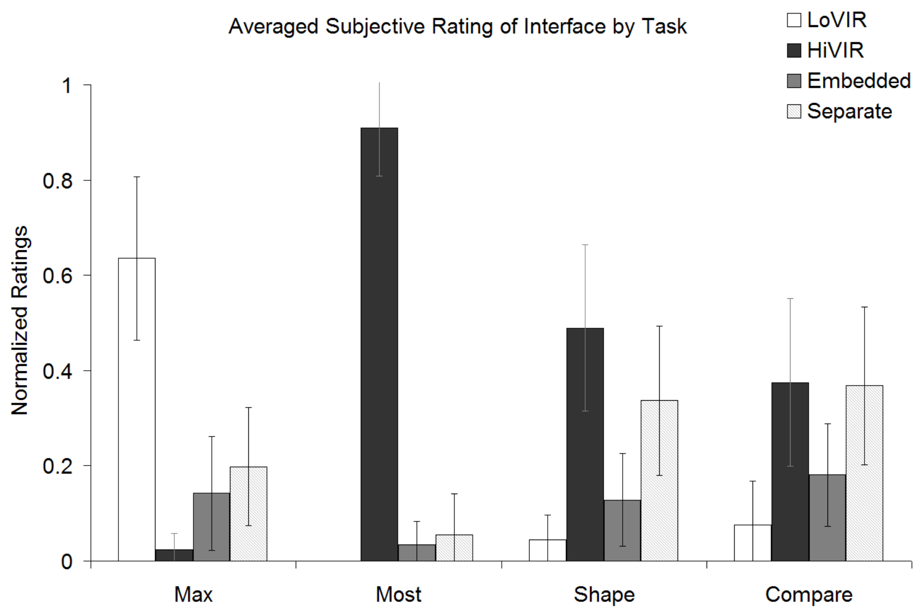


Figure 6.10: Subjective ratings for the four interfaces for each task. Error bars show 95% confidence intervals. (N = 24)

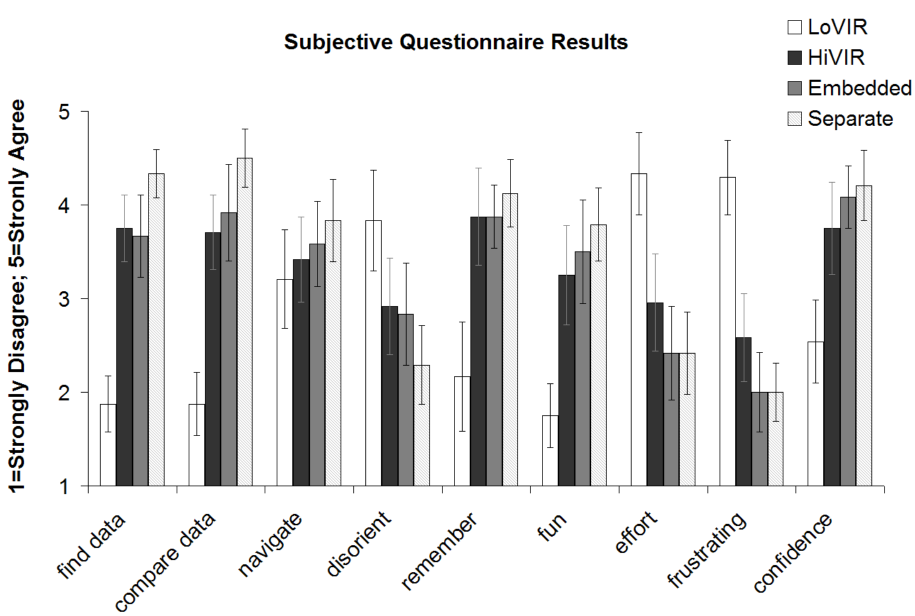


Figure 6.11: Subjective questionnaire results to solicit ease of use ratings for the four interfaces. Error bars are 95% confidence intervals. (N = 24)

for the *Max* task, while *HiVIR* was thought to be most useful in the *Most* task. For the *Shape* and the *Compare* tasks, both *HiVIR* and *Separate* were preferred over *LoVIR*.

We also obtained feedback on the interfaces' ease of use with a five-point rating scale. The questions are listed in Table 6.4, and the actual questionnaire used in the study is included in Appendix D.3. All except the `navigate` question produced significant results. As seen from Figure 6.11, *LoVIR* scored poorest in all the questions with significant findings, reflecting our participants' frustration with the interface. Only the `find data` question differentiated the other three interfaces: our participants found it easier to find data using *Separate* than *LoVIR* or *HiVIR*.

6.3 Discussion

We investigated whether established perceptual requirements for low VIR could be relaxed in multiple-VIR interfaces when selective data are shown at high VIR. With the *Max*, the *Shape*, and *Most* tasks, we studied the two perceptual requirements and showed that visual targets needed to be simple and span a limited visual angle to be reliably detected on the low-VIR overviews, thus confirming H1. Surprisingly, the merits of our multiple-VIR interfaces did not seem to relax these requirements based on participant interface choice and objective performances, thus we were unable to prove H2 or H3. We now discuss our three hypotheses in more detail, along with a more general discussion on multiple-VIR interface use.

6.3.1 H1: True. The low-VIR view alone is sufficient if the target is simple and spans a limited visual angle

For the visual complexity requirement, we compared the *Shape* to the *Max* task. The *Shape* task targets had three peaks, which were displayed as three bands with different colour intensities in the low-VIR view (Figures 6.3 and 6.5). Since these tri-band targets were more visually complex than the single bands in the *Max* task, our participants could not easily find the targets in the low-VIR view. When forced to rely on the low-VIR view, as in the *LoVIR/Shape* tasks, we observed that 13 out of 24 participants resorted to serial search to locate the target. Even when the targets were found, some participants could not confirm their answers visually and needed to crosscheck the y-values using the tool-tips.

Not surprisingly, our participants made more errors, took longer and assigned *LoVIR* the lowest subjective rating for the task.

In contrast, the *LoVIR* interface was effective for the *Max* task, where the majority of our participants (22 out of 24) could find the targets without resorting to serial search. Indeed, 63% of participants considered the overview mode to be sufficient and preferred *LoVIR* for this task. On occasions where the plots were also available, participants only used them to confirm their answers. In short, the low-VIR view is extremely effective for the *Max* task.

The difference in results were large and surprising given the small difference in the two sets of visual targets. We believe that even though the three-peak targets in the *Shape* task were distinctive, the complex structure may be too difficult for participants to process in the low-VIR view. Nonetheless, about half of our participants (21 out of 48 times) used both VIRs for target search in the multiple-VIR trials even though the rest bypassed the initial low-VIR view and switched to the high-VIR view.

The visual span boundary was established using the *Most* task and the *Max* task. Our *Most* task results show the extreme difficulty in using the low-VIR strip when the target spans a wide horizontal region. Unlike the case of the *Shape* trials, even serial searching in the low-VIR became difficult for participants in the *Most* trials, which required counting the number of peaks in each line graph. Participants almost always used the high-VIR plots for the *Most* task in the multiple-VIR tasks (45 out of 48 times), *LoVIR/Most* tasks were on average 89% slower and more erroneous than the *HiVIR/Most* tasks, and 91% preferred *HiVIR* for the task. This lopsided preference is worth noting since participants were performing the same serial-search operation using either the strips or the plots. One reason may be that counting peaks is arguably a detailed visual task that requires foveal vision. In order to count the peaks, participants would need to move their eyes along the strip to focus on individual peaks. Despite the visual aid provided by the framing highlight box, participants could not successfully fix their gaze on the desired strip, and frequently misread values from strips above or below the intended target. As a result, the low-VIR view was effectively unusable for the *Most* task due the dispersed nature of the targets. Interestingly, adding spaces between line graphs seems to add visual noise rather than to help guide visual scan. Line Graph Explorer (Kincaid and Lam 2006), the visualization system that initially presented the strip and plot visual encodings, has an option where users can add darkly coloured blank spaces between line graphs. Our early testers for the system found the blank spaces too visually jarring, and preferred to use the interface without them. Our result in this study is similar to Tullis' model indicating a positive correlation between target visual angle and search time (Tullis 1985), and we believe that the situation would be similar for long vertical targets.

6.3.2 H2: False. Embedding high-VIR plots in low-VIR strips did *not* enhance complex-target matching

We have established that the complex targets used in the *Shape* task could not be easily found by most of our participants, but they could still be serially searched. We now discuss if placing the more comprehensive and familiar plot alongside the unfamiliar and abstract tri-band strip would facilitate learning and prime visual search, enough to at least allow participants to narrow their search space by selecting a set of potential candidate line graphs for detailed examination.

We concluded that H2 was not supported since we did not detect any performance differences between the *Embedded*, *Separate* and *HiVIR* trials for the *Shape* task. The close proximity of the strip and the plot provided by the *Embedded* interface did not seem to have been sufficient for our participants to learn the less familiar colour strips to prime the visual search. In fact, having the extra overviews did not seem to provide any performance benefits, regardless of the spatial arrangement of the two VIRs.

Our observations helped to interpret our performance results: half of our participants in the multiple-VIR interface trials switched to the high-VIR mode to complete the tasks, suggesting that selectively providing high-VIR plots did not seem to provide enough detail for the task. To us, the switch was perplexing, as participants would have to memorize the visual target and scan six full screens to perform the task. This strategy turned out to be difficult for at least two of them, as they missed the targets in their first scan, and had to rescan the entire six screens to find the targets.

One possible explanation of the switching may be the visually different encodings of the VIRs. We attempted to minimize the effect of the difference by visually linking the two encodings with smooth animation in the *Embedded* interface, and also by instructing our participants with sample line graphs shown at the beginning of the training sessions. During the design of the visual encodings, we experimented with filling in the area beneath the line graphs, but that reduced the perceivability of the high-VIR encoding. Even though we could not discount the different encodings as a factor that hindered the use of the *Embedded* interface without further investigation, we believe our participants' choice of switching to the high-VIR display was based more on the interaction costs of the multiple-VIR interfaces. We will further discuss participants' choice at the end of the discussion, since it is not isolated to the *Shape* task.

6.3.3 H3: False. Providing side-by-side visual comparison with selective detailed plots did *not* enhance simple but similar target matching

Our last hypothesis studied whether providing obvious support for the task would relax the target perceptual requirements. We base our discussion on the results of the *Compare* task, where our participants were required to match the simple single-peak target that only differed from the distractors by a small horizontal shift.

Our results did not support H3, as we found that participants were equally slow and error prone for all four interfaces in this task. In other words, we did not detect performance benefits provided by the extra overview of both multiple-VIR interfaces, or even by the side-by-side comparison capability of the *Separate* interface.

Our observations provided insights to the performance results: our participants derived a successful strategy to work with the single-VIR interfaces. In the *HiVIR* interface, for example, participants took advantage of the mouse wheel and scrolled vertically up and down with the cursor fixed horizontally at the horizontal point where the target peaked. As a result, all they needed to do was to find another peak at the same horizontal point numerically by reading off the tool-tips, thus avoiding the need to directly and visually compare the plots themselves. A similar strategy was used in the *LoVIR/Compare* trials. Instead of using the mouse wheel, which was not available as the interface was not scrollable, participants tried to keep the horizontal position of the mouse constant while mousing vertically up and down.

Due to the success of the strategy, a few participants voluntarily switched to the high-VIR plots for the *Compare* task. We saw this happen in 4 out of 24 cases for the *Separate* trials, and 7 out of 24 cases for the *Embedded* trials. One participant in the *Separate/Compare* trial even chose to use the low-VIR view exclusively for the task, a surprising choice given our H1 findings.

We did observe evidence to suggest the merits of using the *Separate* interface in its intended form. For the 20 participants that used both the strips and the plots for the task, 13 found the side-by-side comparison sufficient and did not crosscheck between the originals and the target peaks by reading off the numeric values from the tool-tips. In contrast, with the *Embedded* interface, only two participants relied on visual crosschecking without tool-tips. Perhaps that is why our participants preferred the *Separate* interface for the *Compare* task along with the *HiVIR* interface, even though we could not detect any performance benefits.

6.3.4 Interaction complexity and spatial arrangements

Our results suggest a surprisingly conservative set of visual requirements for the overviews to be usable in a multiple-VIR setting: our participants reliably used the low-VIR displays only when the target was simple and spanned a narrow visual angle. Any deviation from that composition, as in a three-peak target, severely reduced the usefulness of the low-VIR displays.

Given the fragility of the low-VIR interface, it is therefore of great interest to see if selective displays of high-VIR details could compensate for some of the lost perceivability, or offer enough benefits to tolerate the loss. In short, will participants take advantage of the low-VIR interfaces to reduce the search space by first selecting a handful of potential candidates for further examination in detail? Even for tasks that seemed to be suitable for the multiple-VIR interfaces, at least 20% of our participants used only the high-VIR displays when given the choice of a multiple-VIR interface. Granted, using the high-VIR view alone may be impossible rather than simply difficult if a dramatically larger amount of data were used, for example, millions of points rather than the 112,000 used in our study. We did, however, observe considerable difficulties in using the high-VIR displays, and the need for seemingly elaborate strategies to enable their use.

We believe this interesting choice was due to the cost of interface interaction complexity, which may also explain the lack of performance benefits over the optimal single-VIR interface for each task. Although seemingly tedious and laborious, using the high-VIR plots has a low cognitive load: the only navigation available is scrolling, a relatively passive exercise, and the answer will usually be apparent sooner or later. In contrast, navigation in a multiple-VIR interface is complex, as it involves active selection of potential target candidates, an action that requires mental and visual concentration that may render decision making more difficult than when given less choices (Schwartz 2004). It also requires the physical effort in clicking on a relatively narrow strip and potentially the physical effort of scrolling. In our case, having only two VIRs may have made the target more difficult to select in low VIR. An alternative implementation would be to allow opening of a group of line graphs adjacent to the selected target, each opened at an intermediate VIR that gradually bridge between the two extreme cases. Nonetheless, we believe the rest of the costs would still remain in this alternative implementation.

Switching from a multiple-VIR mode to a single VIR can thus provide an easily perceived short-term benefit of lower cognitive load, despite potentially increasing the total time required to complete the task. Our study training for users required them to demonstrate proficiency in the use of all four interfaces, as is usual in single-session laboratory settings. We conjecture that users trained to demonstrate proficiency in a multiple-VIR interface may still not have internalized confidence in its use: that is, may not have adequately understood the longer-term cost of these short-term choices.

For the spatial arrangement of the VIRs, we did not detect differences in participant performances between the embedded or separate ones. The differing costs of the multiple-VIR interfaces may explain the lack of demonstrated differences. For example, embedding plots within the stacked strips can potentially disrupt the overview effectiveness of the low-VIR view, as indicated some of our participants' quick successive opening and closing of the same plot in the *Embedded* trials: open to see plot details, and close to better see the overview for visual search. As for the *Separate* interface, we observed the well-known problem of associating between separate views, where participants closed and reopened the plots in the high-VIR view, one at a time, to re-associate them to the strips.

Despite these costs, we did observe benefits in providing side-by-side comparisons in visual comparison tasks. When using the *Separate* interface, more participants relied on visual crosschecking without tool-tip activations to confirm their answers than in the *Embedded* trials. The merits of providing side-by-side visual comparison may explain the subjective preference results. The *Separate*, along with the *HiVIR* interface, were the preferred interfaces for the two visual comparison tasks, and overall, our participants found it easier to compare between line graphs when using the *Separate* interface than either single-mode interfaces.

6.4 Limitations of Study

This work was a first attempt to look at the interplay between high and low VIRs based on overview target perceivability. Obviously, a more systematic study of other established perceptual requirements, such as item density and grouping (Tullis 1985), is required to draw more precise conclusions. In addition, eye-tracking, instead of note-taking, should be used to allow more precise and objective measurements of interface use. It would also be interesting to separately test participants' visual abilities, which might shed more light on interface choice.

Despite our efforts to ensure diversity and generalizability, our work was naturally limited by our visual encoding and interactions, interface, and study design choices (Section 6.1.7). We also encountered challenges which may be inherent to this study approach (Plaisant 2004). For example, our study failed to find evidence to indicate multiple-VIR interface use for single-level data. Such null results may indicate inappropriate choice of tasks, or insufficient training for participants. My co-investigators and I took extra care in selecting study tasks: we started with published task taxonomies, created sample tasks suitable for our visual encoding, and reiteratively selected study tasks based on pilot results. Even during pilots, we were aware that benefits in our *Embedded* interface may not be fully realized in our set of study tasks. Even though we wondered if it could be due to the simplistic nature of our tasks, we were unable to create a more complex task that would truly test the use of the *Embedded* interface.

Another challenge encountered in this study was one of training. Insufficient training is frequently used to explain lack of use in complex, and often novel, interfaces (e.g., Plaisant 2004). In our case, many of our participants, from 20% to 96% depending on the task, switched to the high-VIR mode even when we believed using the multiple-VIR counterparts should be beneficial. As discussed in Section 6.3.4, we realized after the study that being able to demonstrate proficient interface use does not necessarily guarantee effective use, as participants may not adequately understand the benefits in using the interface and internalize confidence in its use. Encountering difficulties with participant training has also been the experience of Nekrasovski et al.'s (2006) study on tree visualizations.

6.5 Summary of Results and Implications for Design

Using a set of contrasting visual targets displayed on two visual information resolutions (VIRs), we started by establishing the criteria of two visual qualities for an effective low-VIR view, namely that the target should be simple and span a limited visual angle. When either of these boundaries were crossed, multiple-VIR interfaces did not enhance visual search over using a single high-VIR view, even though the multiple-VIR interfaces provided obvious benefits for the tasks, for example, integrated VIRs in *embedded* and side-by-side comparison in *separate*. We believe our results reflected the high cognitive costs in multiple-VIR interaction such as view coordination in our *Separate* interface. Such costs should be considered in interface design, especially in cases that lack obvious benefits in using a multiple-VIR design, as in our case of displaying single-level data. This issue was discussed in our summary synthesis in Section 4.4.

Perhaps because of these interaction costs, we did not find any performance differences between the two simultaneous-VIR arrangements in our study, and the issue remains an open research question. We discussed this issue in our summary synthesis in Section 4.6, and will further discuss it in the last chapter of this thesis in Section 9.1.3.

Chapter 7

Session Viewer Design

As mentioned in the Introduction chapter in Section 1.2.4, experimental strategies are necessarily limited by the need to study isolated interface factors using pre-defined, abstracted, and easily testable tasks performed by non-domain experts with questionable levels of motivation (Plaisant 2004). In our experimental-simulation study of overview use in multiple visual information resolution (VIR) interfaces, we attempted to alleviate some of these limitations by recording detailed observations of participants' interface choices and task strategies, which turned out to provide more valuable insights into interface use than the quantitative measurements. However, as discussed in the limitations of the study (Section 6.4), participant training and motivation, as well as study-task limitations, could have hindered our study of interface use.

We believe we needed to evaluate a fully functional visualization system under ecologically valid settings, such as with representative target users, who are domain experts working on their own tasks using their own data. We therefore continued our investigations in multiple-VIR interface use with a field evaluation. Specifically, we continued our efforts to study the two research questions identified in our summary synthesis: how to effectively create an overview and how to arrange VIRs spatially.

The overview creation and use question was first reviewed in our summary synthesis in Section 4.4 where we concluded that both providing too many data points on individual low-VIR views may hinder user performance, and data displayed also need to provide enough visual details to be useful. We verified two perceptual requirements, visual complexity and visual span, of graphical overview objects in our overview-use study in Chapter 6.

The question of VIR spatial arrangements was reviewed in our summary synthesis in Section 4.6. Although we could not offer design guidelines regarding the two approaches, our preliminary characterization of distortion usage for the *embedded* approach was a conjecture that more drastic distortions may be less tolerable than mild ones. The impact of distortion, or more generally geometric transformations, was the subject of our study in Chapter 5. We also further examined VIR spatial arrangements in our overview-use study in Chapter 6, where we noticed interaction costs for each of the approach.

In addition to continuing our investigations of these two research questions, we also wanted to explore workplace requirements of visual analytic systems.

Our application domain is web session log analysis. We explained the rationale behind our choice, detailed background information on current practices and problems in log analysis, along with the nature of session logs, in Section 2.4.2. To better focus on studying interface use in the workplace, we decided to implement our own software so that we could better tailor our test tool to our participants' data analysis needs. In addition, we could also incorporate our design guidelines derived from our first three studies in our tool design, and observe their impact on interface use in our field evaluation. In short, Session Viewer plays a major role in this thesis as our test bed to examine our design considerations in based on the first three evaluations in this thesis.

In this chapter, we detail our experience in building Session Viewer, a visual analytic tool to support web session log analysis. In the Background chapter (Chapter 2), we considered design implications and requirements based on the nature of web session logs (Section 2.4.1), the task of exploratory data analysis (Section 2.1), and potential roles played by visualization in data analysis (Section 2.2). We now continue our design-space exploration by considering our target users' existing analysis practices, workflow, and challenges encountered in their analysis (Section 7.1). To ensure our design would meet users' needs, we invited two end-users to participate in the design of Session Viewer, and obtained feedback from five target users (Section 7.2). Based on all these considerations, we derived our design goals and implemented them as tool features (Section 7.3).

The rest of this chapter details the visual components and interactivities in the final design of Session Viewer (Section 7.4), followed by a use-case scenario where we illustrate its use with an analysis performed on a set of web session log data gathered from a large-scale user study conducted by a third-party company (Section 7.5). Earlier versions of Session Viewer are presented in Section 7.6, along with four design evolution themes concerning overview creations and panel layouts.

7.1 Target Users: Web Session Log Analysts

We designed Session Viewer for experienced web session log analysts. To ground our designs, we first interviewed five analysts to understand their analysis goals, tasks and workflows. Each analyst had at least five years of analytical experience with at least one year in session log analysis. The interviews were semi-structured with an initial list of questions, but were largely driven by participants' descriptions of their analysis processes illustrated with their tools. The interviews were an hour long and were recorded.

The scripted interview questions were:

1. What are you trying to find when you analyze sessions? What are the main goals?
2. How do you come up with analysis hypotheses?
3. How do you go about analyzing / examining the session logs? For example, is there a protocol you follow?
4. Which aspects or properties of the logs are most important to your analyses? For example, do you look only at aggregates (and if so, what); do you look at individual sessions?
5. What kind of software tools do you generally use? Are they external or in-house, or self-built?
6. How would you like a software tool to help in your analyses? What would you like to be able to do?

Based on interview observations, we identified two analysis levels: detailed-session and statistical-aggregate. Of the five analysts we interviewed, two were detailed-session analysts, two were statistical-aggregate analysts, and one used both methods. While both types of analyses aimed to understand usage behaviour, they had different needs and illustrated the scope of session log analyses.

Detailed-session analysis aimed to understand usage behaviour by studying sessions in detail. In other words, analysts looked at one session at a time, following it event by event. The questions asked in detailed-session analyses were specific, but tended to be open-ended. For example, one investigation by our analyst was to understand the use of Boolean OR in queries. One of the goals in detailed-session analysis was to develop metrics to measure intended task goals and searcher satisfaction based on event attributes that could be recorded without searcher interventions, such as event action and duration. While searcher goals and satisfaction are paramount in improving search engine quality, these measures have to be provided directly by the searchers, and solicitation may disrupt workflow during studies.

To develop these metrics, our analysts explored sessions so as to relate event or session attributes to task nature or user satisfaction, usually gathered separately based on explicit searcher feedback. Typically, our analysts looked at less than 500 sessions per analysis. The moderate number was partly due to time and effort constraints, but more importantly, most analysts could form satisfactory hypotheses based on 100 to 200 pertinent sessions, for example, sessions with advanced search when the analysts studied advanced search behaviours. Statistical methods were sometimes used for further analyses.

Statistical-aggregate analysis also aimed to understand usage behaviour, but at the aggregate level based on established metrics. Typically, analysts compared different session populations resulted from experiments that investigated effects of algorithm or graphical interface element modifications. One example was to investigate effects of changes in search result presentations by the search engine on click-through rate, which is a standard metric to measure the number of searchers who clicked on individual links. Section 7.5 describes another example as a use-case scenario where we compared populations from a user study where participants performed different tasks.

7.2 Design Process

Session Viewer was developed using the user-centered design approach. We also involved two target users at various stages of our design process. Initially, we developed our first paper-based and interactive prototypes based on our understanding of the web session log data (Section 2.4.1), the general exploratory data analysis task based on literature (Section 2.1), and specific task details and analysis needs in web session log analysis obtained from interviews conducted to gather design requirements (Section 7.1). During this initial stage of the design, we sought feedback from an experienced detailed-session web session log analyst to iteratively improve our design. Our design was also guided by our own experience in using Session Viewer to analyze an existing session log generated by a large-scale user study, which we eventually used as a use-case scenario to illustrate the use of Session Viewer in log analysis (Section 7.5). The result of this design stage was our first fully functional prototype, SV1 (Section 7.6.1). This design stage lasted four months.

We tested SV1 with two target users. The first was the detailed-session analyst who had been providing us with feedback throughout our initial designs, and the second was a statistical-aggregate analyst who saw the prototype for the first time at this stage. The result of this design stage was SV2, modified from SV1 to support multiple session populations (Section 7.6.2). These two users were also invited to participate in subsequent design discussions of Session Viewer, and participated in the field evaluation for the final version. The field evaluation is detailed in Chapter 8.

In addition to our two design collaborators, we also obtained feedback from five target users for SV2. These sessions were informal observations where we began by briefing our testers with demonstrations and explanations of the various visual elements and interactivities of Session Viewer, followed by observations of users interacting with the software. Most of the time, our users used our sample study logs in these testing sessions, which generally lasted about 30 minutes. We also performed more in-depth and multiple-session testing with one selected user of the five, who had specific requests to improve usability and to extend functionality of the software. At the end of our second phrase of development, we consulted a visual designer to improve interface visual quality by inspecting the interface without using it in analysis. The second design phase lasted six months.

Based on feedback from users and the visual designer, we modified SV2 to create SV3 in two months. The total develop time for Session Viewer is therefore a year (SV1: four months; SV2: six months; SV3: two months). This chapter introduces SV3, and the next chapter details its field deployment (Chapter 8).

7.3 From Design Goals to Tool Features

This section explains design goals based on considerations of data object, exploratory data analysis task, and user needs. The section also highlights Session Viewer tool features that realized these goals. Detailed descriptions of visual components and interactivities in Session Viewer are delayed to Section 7.4.

7.3.1 User-defined data objects

Session Viewer manages two base data types in web session logs: session and event, each with its own list of attributes. Data objects in web session logs are discussed in the Background chapter in Section 2.4.1.

To reiterate, an event in web session logs represents a user action such as submitting a query to a search engine. Since each event object has a list of attributes such as action, duration, and property, Session Viewers allows users to define a list of **event states** to further categorize events in addition to the default grouping by sessions. For example, users can define event states based on event actions. A Search event, where the user submits a query to the search engine, can be defined as (`Event.action == SEARCH_CLICK`). Similarly, a ResultClick event, where the user selects a link from the result page returned by the search engine, can be defined as (`Event.action == RESULT_CLICK`). More examples of event states are discussed in the use-case scenario in Section 7.5.

Since each event is associated with a single session, users can define **session attributes** based on event states. For example, the session attribute #Search is defined as the total number of events belonging to the Search event state in each session.

Since events are time ordered, users can define an usage **pattern** which comprises of a sequence of event states and represents a particular usage behaviour. For example, denoting the Search event state as S and the ResultClick event state as R, the event sequence of S→R→S→R→S may represent search exploration. Section 7.4.5 further explains pattern matching in Session Viewer. More examples of usage patterns are discussed in the use-case scenario in Section 7.5.

7.3.2 Main visualization panels

This section introduces the main visual components in Session Viewer. Figure 7.1 shows a schematic diagram of the interface, and Figure 7.2 shows a screen capture. Session Viewer uses multiple coordinated views with standard linked interactions to display several session populations side-by-side for visual comparison.

Using our terminology for multiple visual information resolution (VIR) interfaces introduced in Section 1.1.1, Session Viewer is basically a *separate* interface with each data level displayed as a different VIR. Each VIR is housed in a panel, which are in turned grouped into panes. Interactions in Session Viewer link these panels, and are explained in Section 7.4.4.

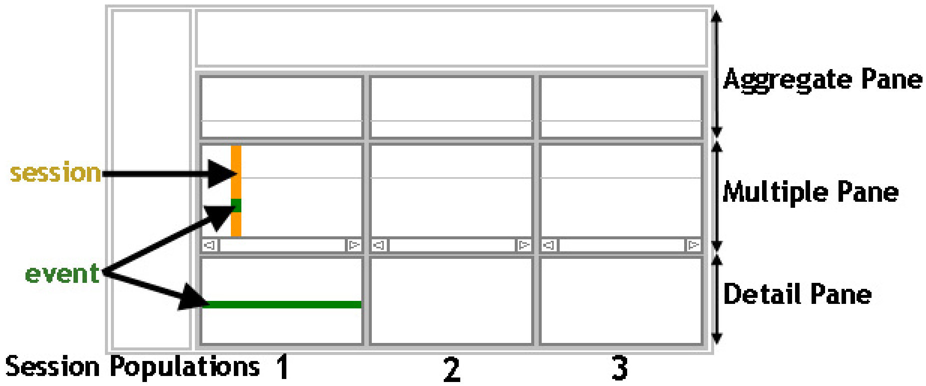


Figure 7.1: A schematic diagram showing the multiple coordinated views in Session Viewer. Each vertical view shows a population with three panes. Each pane corresponds to a data level and contains one or more panels.

Session Viewer displays each population in a vertical *view* and shows the data at three levels, each in a separate pane:

1. The *Aggregate Pane* corresponds to the session-population data level and contains panels that display population statistics such as counts, distributions and annotations. Panels in this pane are low-VIR views that provide aggregate-level overviews.

2. The *Multiple Pane* corresponds to the sessions data level and contains two linked panels of session attributes and sessions as collections of events. Panels in this pane are low-VIR views that provide sessions-level overviews.

3. The *Detail Pane* corresponds to the events data level and shows the logs for a selected session in a table, with one row per event. The single panel in this view is a high-VIR view that provides the highest level of detail available for each event object.

Multiple levels of data for each session population are displayed as separate views using the *separate* approach. For example, the Aggregate Pane acts as an aggregated overview at the aggregate-level to the Multiple Pane session attribute and sessions details. The Multiple Pane in turn acts as an sessions-level overview to the Detail Pane, which shows event details for an user-selected session.

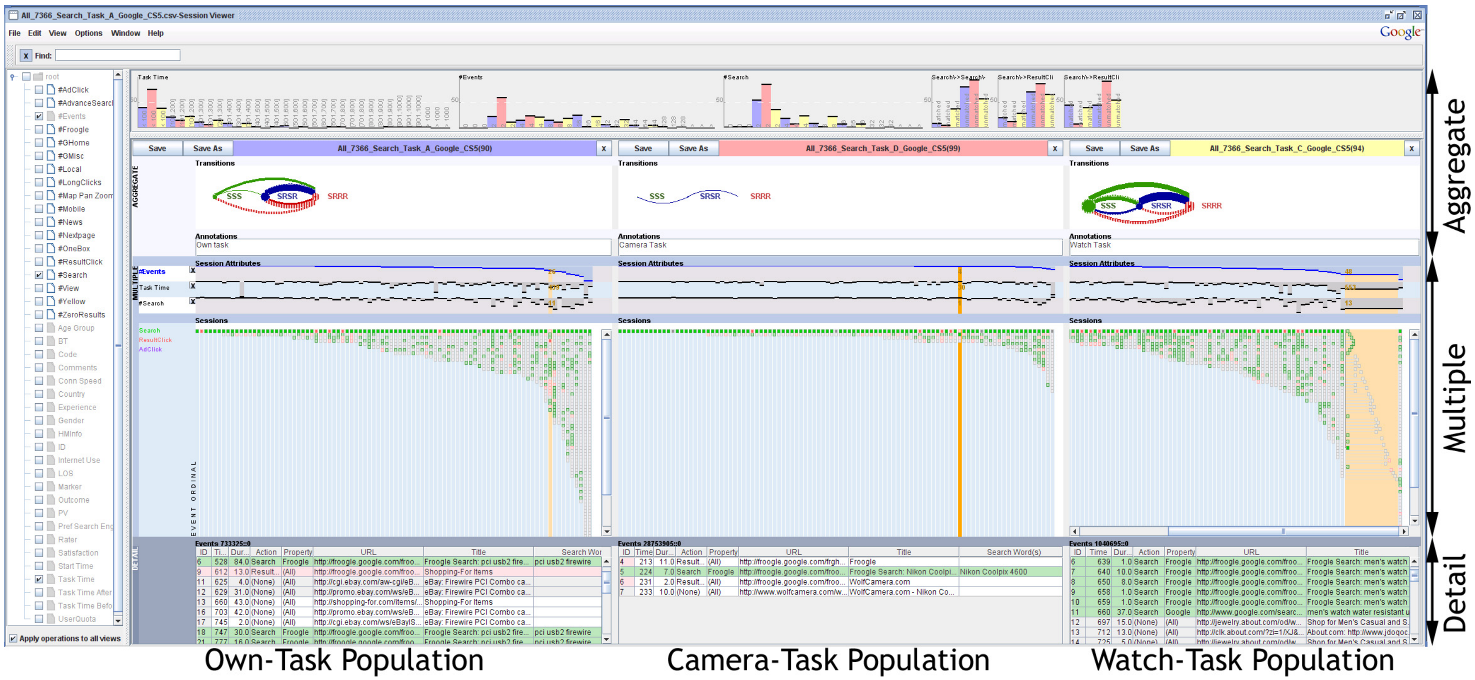


Figure 7.2: The main screen of Session Viewer showing user study data. Session Viewer displays multiple session populations in vertical views. From left to right: the Own-Task population where participants performed their own self-created tasks; the Camera-Task population with searches for a camera feature given the brand and model; and the Watch-Task population with searches to locate a watch based on given criteria. For each population, the session logs are shown in the Aggregate, the Multiple and the Detail Panes.

7.3.3 From design goals to tool features

We set the main design goal for Session Viewer to support session log analysis at both the statistical-aggregate and the detailed-session analysis levels and to bridge between the two. Here we highlight the features designed to achieve this goal.

1. Support analysis at the statistical-aggregate level

Even though statistical analysis leads to highly scalable, succinct, and comparable numeric descriptions of populations, proper statistical analysis requires matching data with methodological assumptions, which in turn requires understanding the data distributions. Also, rapid hypothesis testing is often difficult in practice, as most statistics packages require non-trivial data regrouping and formatting for different analyses. To address these concerns, Session Viewer:

- *Provides statistical summaries.* As shown in the Aggregate Pane in Figure 7.2, Session Viewer provides descriptive statistics as graphical plots commonly used in log analysis, such as histograms to show session distributions based on session attributes (Figures 7.3 and 7.5), and a event-pattern transition diagram to show the relative amount and transitions between event sequences (Figure 7.6). Providing aggregate-level overviews helps analysts identify subpopulations for exploration within Session Viewer or make informed choices of statistical methods for further analysis. The issue of developing aggregate-level overviews is further discussed in Section 7.6.4 in association with discussions on design choices and evolution.

- *Detects patterns.* Session Viewer provides a sequence-matching feature to locate usage patterns that is similar to regular-expression matching in strings, such as the Unix command `grep`. In our case, the "alphabets" are user-defined event states. Users can highlight sessions with specific event state sequences in the Multiple Pane to visually and rapidly estimate relative pattern prevalence for hypothesis testing. Section 7.4.5 further explains pattern matching in Session Viewer. An example is shown in Figure 7.15 and detailed in the use-case scenario in Section 7.5.

- *Enables session population comparisons.* The visual equivalent of comparative statistics is visual comparison between populations. Session Viewer provides side-by-side comparison of populations at all data levels, enabled

Figure 7.3: Two histograms from the Histogram Panel of the Aggregate Pane displaying statistical distributions of two session attributes: #Event, or total number of events per session, and #Search, or total number of search events per session. The histograms are constructed by merging corresponding histograms from displayed session populations. Histogram bars are colour-coded by session populations.

by shared scales in graphical plots across vertical views. For example, all histograms in the Histogram Panel share the y-scales, as shown in Figure 7.3.

2. Support analysis at the detailed-session level

Detailed session analysis can yield insights unavailable from aggregate metrics. However, analysts need to examine individual sessions, track events both within a single session as well as between sessions, and coordinate between the event webpages, the session logs, and their own annotations. Moreover, session selection is difficult, since the nature of sessions is difficult to discern from logs, especially when viewed as text similar to Figure 2.1. Oftentimes, detailed analysts need to use keyword search offered by text-based applications to locate interesting sessions. To address these issues, Session Viewer:

- *Displays events in the context of sessions.* As discussed in Section 2.3.1, an individual datum is only meaningful in context. In the case of web session log analysis, event contexts are adjacent events within a session and the larger session population. Context in Session Viewer is provided in a number of panes. Since events are time ordered and associated with a particular session, they are always presented in sequence within their sessions in both the Multiple and the Detail Panes. Each session is put in the context of the larger population in the Multiple Pane based on session attributes displayed as a series of bar charts similar to Table Lens (Rao and Card 1994).

  This requirement implies the need to display sessions at the level of events for the sessions-level overview to be useful. One challenge here was to provide enough task-relevant information on these overviews to allow effective use, as discussed in Section 4.4 of our summary synthesis regarding low-VIR creations. We also needed to provide sufficient visual details for users to isolate interesting sessions, as found in our overview-use study of perceptual requirements of graphical overview objects in Chapter 6. The issue of overview creation is further discussed in the context of design evolution in Section 7.6.3.

- *Integrates analysis resources.* The Aggregate Pane supports annotations for each population and the Detail Pane provides direct links to event webpages, as shown in Figure 7.4.

- *Guides session selection for detailed analysis.* Users can choose sessions by event sequences. Session Viewer highlights sessions based on user-defined event sequences defined in a pop-up dialogue box (Figure 7.12). Also, Session Viewer displays session-attribute distributions that are reorderable. The Multiple Pane in Figure 7.2 shows sessions reordered by total event counts, and the analyst selected sessions with high event counts for detailed study. The action is based on Bertin's reorderable matrix (Bertin 1981), extended to multiple views to show multi-level data objects. The Multiple Pane therefore provides multiple representations of the session data to guide session selection for detailed analysis, a requirement of visualizations to support exploratory data analysis discussed in Section 2.3.2.

  The challenge here was to find an effective spatial layout for the different panels. Our findings in the first three studies in this thesis suggested that there were tradeoffs in using either the *separate* or the *embedded* approach: the *separate* approach may incur problems in view coordinations, as found in the summary synthesis (Section 4.6) and the overview-use study (Section 6.3.3), and distortions frequently implemented in the *embedded* approach may incur perceptual costs, as found in the summary synthesis (Section 4.6.1) and the laboratory experiment (Section 5). The issue of panel layout is further discussed in the context of design evolution in Section 7.6.6.

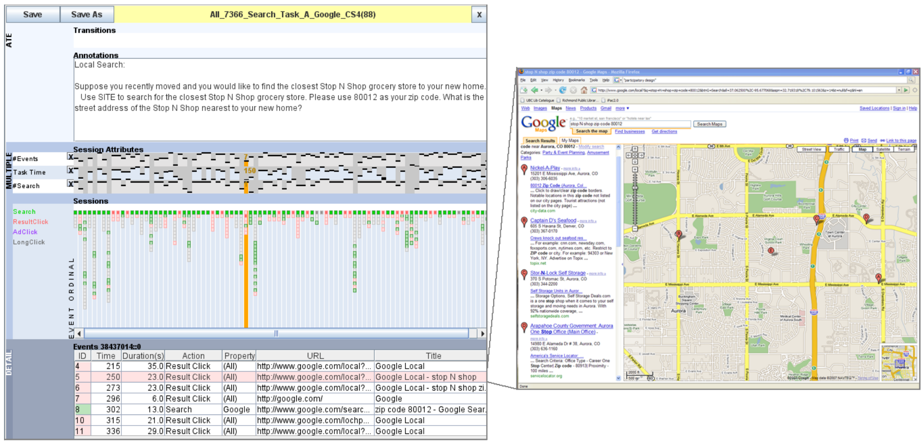


Figure 7.4: Session Viewer integrates analysis resources by providing an Annotation Panel for text entries and an Events Panel that links to an external web browser. Double-clicking on a row in the Events Panel table displays a recreated webpage based on event URL.

3. Bridge between the statistical-aggregate and the detailed-session analysis levels

While most analysts we interviewed acknowledged limitations of specific analysis methods, they also expressed difficulties in extending their practices to include multiple data levels as most tools did not adequately support cross-level analysis. For example, to better understand a particular aggregate metric in a statistical-aggregate analysis, analysts would need to associate and examine individual sessions with the selected metric values. Similarly, to guide session selection in a detailed analysis, analysts would need to calculate and plot distributions of relevant session attributes. Such data-processing steps are non-trivial and distracting to the main analysis.

Session Viewer encourages multi-level analysis by displaying session logs at three levels of detail. Using standard linked navigation and highlighting techniques, users can quickly move up and down the data hierarchy in their analysis, a requirement discussed in Section 2.3.3. For example in Session Viewer, clicking on the Transitions Panel in the Aggregate Pane (described below) highlights sessions with the specified event-state transitions in the Multiple Pane. Users can then select an individual session in the Multiple Pane to display event details in the Details Pane.

4. Connect to, rather than replicate, existing analysis tools

Instead of covering all aspects of data analysis, we envisioned Session Viewer as part of a toolkit and focused on supporting data exploration and hypothesis generation, or the exploratory data analysis step in data analysis as discussed in Section 2.1. Due to our specific process coverage, Session Viewer needs to connect with log sources and commercial statistics packages. To ease data transfer, Session Viewer imports data from various log sources and exports them in standardized formats.

7.4 Session Viewer: Visualization and Interactions

We now describe the individual panels in Session Viewer organized in three panes: Aggregate, Multiple, and Detail, as shown in Figure 7.2.

7.4.1 Aggregate Pane

This top pane has three panels showing population metrics and distributions: the Histogram Panel, the Transitions Panel, and the Annotations Panel.

The **Histogram Panel** serves two purposes: (1) displays session attribute counts and distributions, and (2) provides session filtering. Individual session population histograms for each session attribute are merged into a single histogram, such as the event count attribute histogram in Figure 7.3. Each population's data are colour coded. For example, in Figures 7.3 and 7.5, the yellow histogram bars belong to the Watch-Task population. Each histogram bar functions as a toggle button to filter in or out sessions with the corresponding session attribute values. Figure 7.5(b) shows the effect of filtering out sessions with less than three events in the red Camera-Task population.

The **Transitions Panel** displays pattern transitions flowing clockwise (Figure 7.6). Event patterns are defined in a separate dialog box (Figure 7.12). The following examples use the shorthand for Search (S) and ResultClick (R) event states: SSS may suggest query refinement; SRSR may suggest methodical result refinement; and SRRR may suggest result exploration. Arcs sharing the same originating and destination states are bundled to avoid overlapping. The Transitions Panel can be used to study usage behaviour transitions. For example, Figure 7.6 suggests that searchers in the depicted population tended not to

Figure 7.5: Histogram Panel. Individual histograms for each session attribute are colour-coded and merged into a single display. Each histogram bar functions as a toggle button to filter in or out corresponding sessions. (a) The original unfiltered Camera-Task population (red) mostly had sessions with two or less events. (b) The hollow red histogram bar indicates sessions with two or less events have been filtered out, as seen in the Sessions Panel.

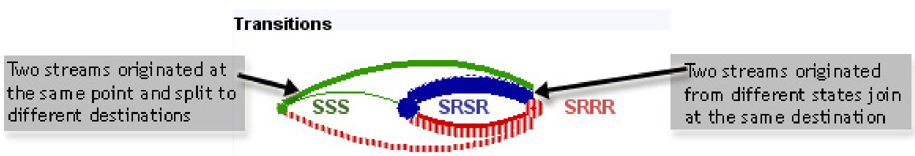


Figure 7.6: Transitions Panel. With S denoting a Search event state and R, a ResultClick event state, the display shows transitions between the pattern S→S→S, S→R→S→R, and S→R→R→R. The colours of the arcs are user-defined and represent the originating patterns.

refine their queries once they were in the methodical result refinement mode, as no transitions were found between SRSR, or the methodical result refinement pattern, to SSS, or the query refinement pattern.

The **Annotation Panel** is a text box to allow for note-taking, as shown in Figures 7.2 and 7.4.

7.4.2 Multiple Pane

The middle Multiple Pane has two panels that function as a unit, showing individual sessions and attributes as indicated by the shared horizontal scroll-bar in the schematic diagram in Figure 7.1. Each session occupies a unique vertical lane spanning both panels, as seen in the orange highlight in the schematic diagram in Figure 7.1 and in the screen capture in Figure 7.2.

The **Session Attributes Panel** shows user-selected attributes for each session displayed in a Table Lens-like chart (Figure 7.10). Since the panel is used for displaying trends instead of directly reading off individual attribute values, heights of as few as 10 pixels are acceptable for the bar charts.

Session attributes can be ordinal, such as the search counts, or categorical, such as task outcome (success, failure, given-up). Session Viewer allows users to mark sessions by clicking on the white space at the top of a session to place an orange-oval marker (Figure 7.7). Session marking is recorded as the marker session attribute and is therefore reorderable.

The **Sessions Panel** shows each session as a stack of colored rectangles (Figure 7.8). Each rectangle corresponds to an event colour-coded by an user-defined event state. Time flows from top to bottom. Events at the top of the display are therefore first events of the sessions, and events are displayed in time orders. The rectangle height is either uniform, thus encoding the position of each event (or event ordinal), to better display event sequences (Figure 7.8(a)), or encodes event duration to highlight long events (Figure 7.8(b)). Sessions are aligned at the start of the sessions by default, or aligned at a user-chosen common event. Figure 7.8(c) shows two examples of aligning by the first occurrence of a search event. The right-hand side sessions in Figure 7.8(c) has fewer events after the common query than those in the left, suggesting a more effective query string.

Users can click to expand individual sessions into two-dimensional displays to show usage behaviour, with the vertical dimension still encoding event ordinal or time while the horizontal dimension encodes unique event URLs. For example, Figure 7.9 shows three types of usage patterns. Vertical columns of event boxes indicate webpages with the same URLs. Based on detail logs displayed in the Events Pane, green columns of event boxes correspond to result pages returned from a general search engine, Google, annotated as SS (Search-engine Searches). Grey columns boxes correspond to result pages returned from site-specific search engines such as amazon.com, annotated as TS (Third-party Searches). Diagonal runs of event boxes represent a search behaviour where users continuously launch new webpages, suggesting exploration and annotated as TE (True Explorations).

Users can drag and drop attribute names in the Session Attributes Panel to reorder the sessions, and add or remove attributes displayed by checking or unchecking a list of user-defined session attributes displayed on the far left panel of Session Viewer (Figure 7.2).

The vertical display order of attribute names determines the horizontal sort order of the sessions of both the Session Attributes and the Sessions Panels. Figure 7.10 shows examples where session reordering can reveal correlations between session attributes. The attributes of interest are two subjective feed-

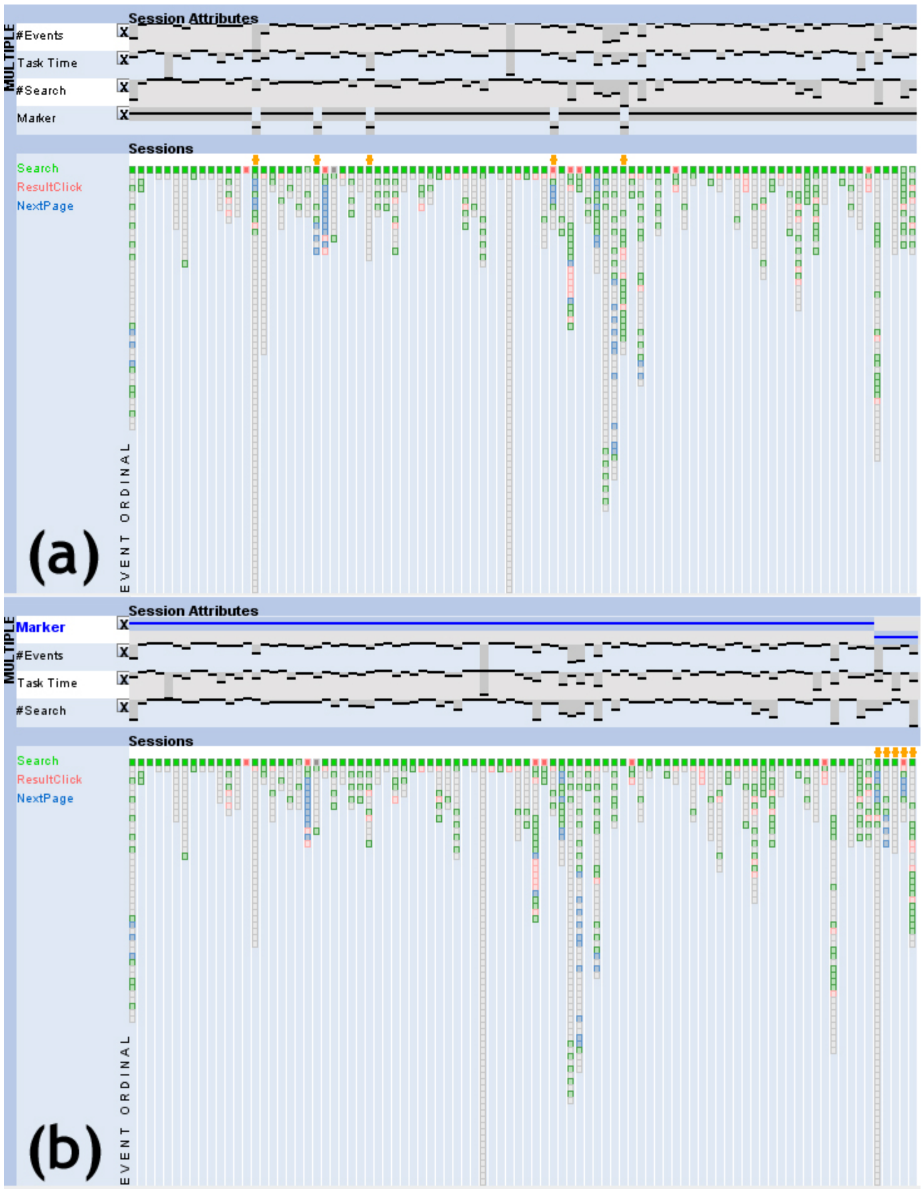


Figure 7.7: Session markers. (a) Users can flag a session by clicking on the white space at the top of the session to place an orange-oval marker. (b) This flag is recorded as the marker session attribute, which is reorderable.

back from the searchers who generated the logs: Task Outcome, which could be Success, Failure, or Given-up; and Satisfaction Score, which was numerical and

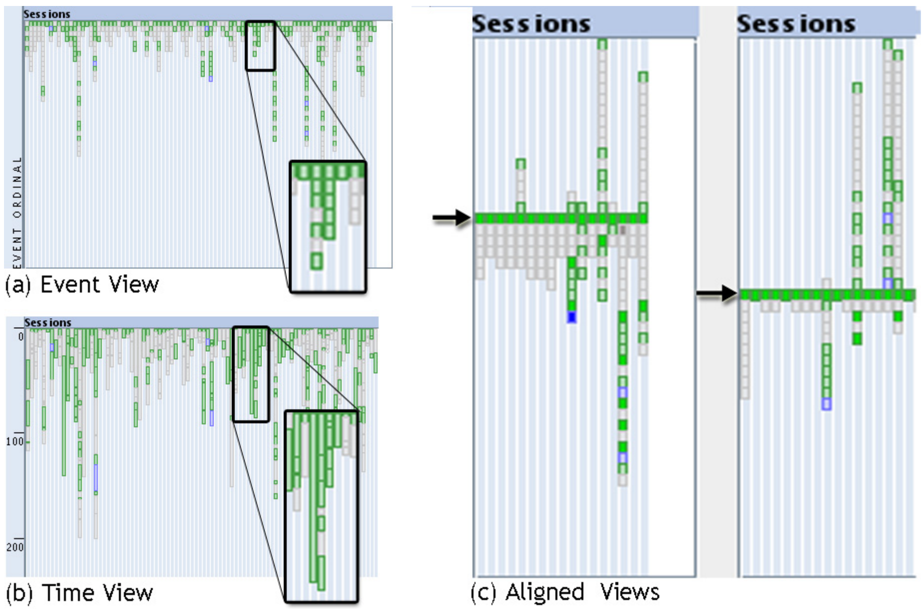


Figure 7.8: Sessions Panel. (a) Event view is the default, where each event rectangle has the same height to better show transitions. (b) In the Time view, the height of each rectangle encodes the event duration. (c) In the Aligned view, sessions are aligned vertically at an user-chosen common event (indicated with arrow annotations). Sessions without that event are filtered out.

ranged from one to seven.

Figure 7.10 shows two populations: top (a), and bottom (b1, b2). In Figure 7.10(a,b1), sessions with low Satisfaction Scores were highlighted in orange and reordered by the Task Outcome, with Failure and Given-up sessions in the far left. In Figure 7.10(a), the orange highlighting was concentrated on the left side of the display, indicating a strong correlation between Task Outcome and Satisfaction Score attributes in the top population. However, in Figure 7.10(b1), we did not see that same correlation in the bottom population. Instead, we noticed that #ResultClicks session attribute, or the total number events in a session belonging to the ResultClick event state, may be correlated with task outcome. Indeed, there were more red ResultClick events in the Sessions Panel on the right side of the display, where sessions with the Success Task Outcome clustered. To explore that correlation, we reordered the sessions by the #ResultClicks session attribute: sessions with low ResultClick counts were on the left in Figure 7.10(b2). Our hypothesis was confirmed since unsuccessful ses-

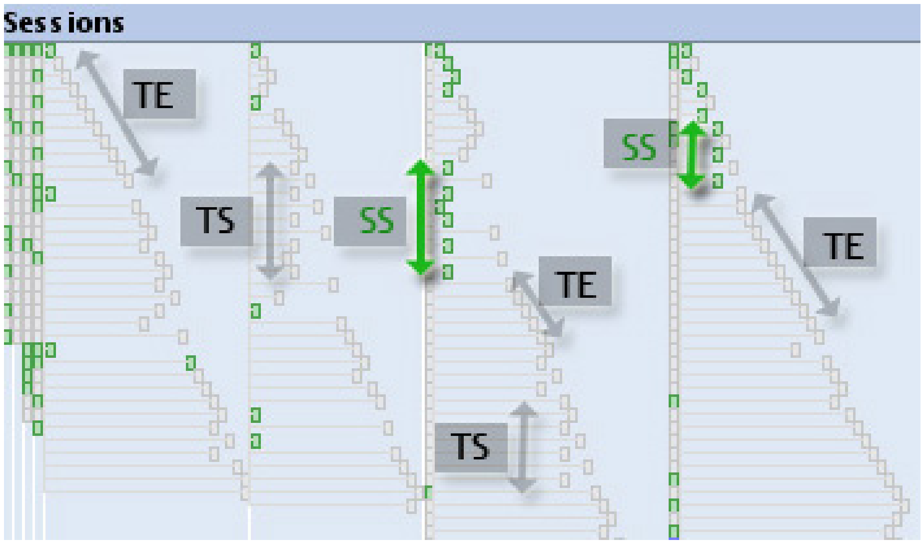


Figure 7.9: Expanded sessions in the Sessions Panel showing three search patterns: SS, Search-engine Searches; TS, Third-party Searches using third-party online sites as search engines; and TE, True Explorations of search results.

sions (highlighted in orange) were clustered on the left with low ResultClick counts.

7.4.3 Detail Pane

The low-level pane has a single panel, the **Events Panel**, that shows an individual session as a table (Figures 7.2 and 7.4). Each row shows an event, with columns displaying attributes such as event ID, start time, duration, URL, the specific query for Search events, and an embedded link to event webpage displayed in a separate browser using the recorded event URL (Figure 7.4).

7.4.4 Interactions and view coordinations

Session Viewer uses linking and navigation techniques for view coordinations (North and Shneiderman 1997), as shown in Figure 7.11. For example, as depicted in Figure 7.11(a), scrolling and reordering are limited to the Multiple Pane, but filtering in both the Aggregate and Multiple Panes can be initiated from Pattern-Matching or Session-Alignment actions, and from the Aggregate Histogram bars (Figure 7.11(b)). Highlighting a session in the Sessions Panel

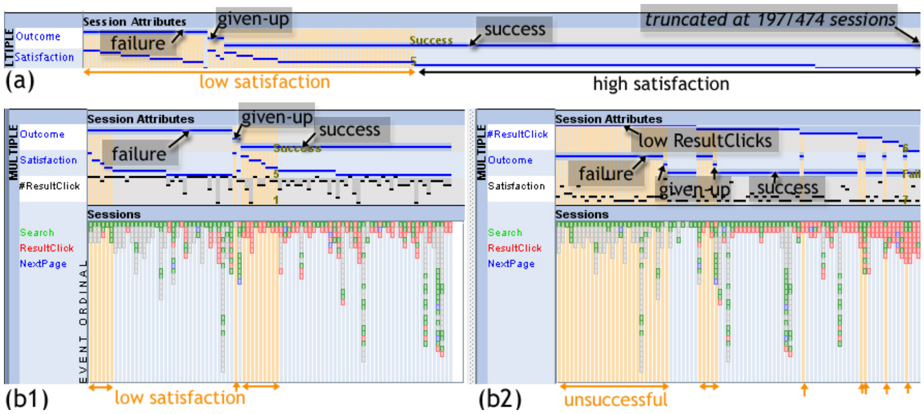


Figure 7.10: Examples to show how the Session Attributes Panel reveals trends and correlations. Top population: (a) Low satisfaction scores are highlighted in orange. Sessions are reordered by Outcome and by Satisfaction. Since the orange-highlighted sessions with low satisfaction scores cluster with the Failure and Given-up outcomes, the panel shows a high correlation between task outcome and satisfaction score in this population. Bottom population: (b1) Satisfaction score and task outcome are not correlated, as seen by the lack of clustering of low satisfaction scores in the Failure and Given-up sessions. (b2) Instead, we see the ResultClick event count correlates with task outcome, as shown by the cluster of highlighted unsuccessful sessions with low ResultClick counts on the left.

displays its events in the Events Panel and highlights the associated session in the Session Attributes, as shown in Figure 7.11(c).

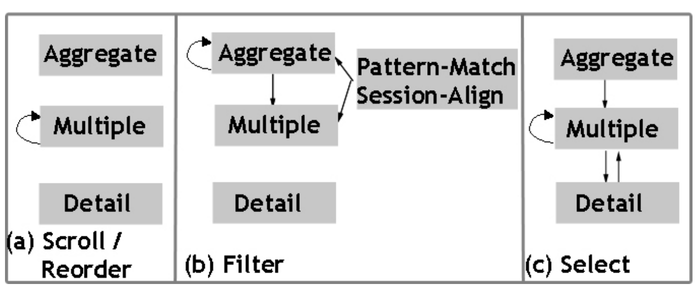


Figure 7.11: Interaction coordination scheme between the three data levels: (a) Session reordering and scrolling is limited to the Multiple Pane; (b) Filtering is initiated at the Aggregate level, or by the pattern-matching or session-alignment feature, affecting the Aggregate and the Multiple Panes; (c) Selection and highlighting can be initiated at all data levels.

7.4.5 Other tool features

Session Viewer provides three main tool features that produce session-analysis views and are dialog-box driven: (1) pattern matching, (2) session alignment, and (3) session population partitioning.

1. Pattern matching

Since events do not occur in isolation, Session Viewer offers pattern matching by highlighting sessions with certain patterns of event-state sequences in the Multiple Pane. Figure 7.15 shows two examples.

As discussed in Section 7.3.1, patterns are specific event-state sequences that users can define in Session Viewer using a dialog box shown in Figure 7.12, and pattern matching is similar to regular-expression matching in strings, such as the Unix command `grep`. In our case, the "alphabets" are user-defined event states.

Since a number of actions may be considered interchangeable, Session Viewer allows an alternative event state for each element in the pattern. For example, using event states Search (S), ResultClick (R), and NextPage (N), the topic exploration pattern can be specified as S→R→S/N→R if the NextPage event state is thought to be the same as the Search event state. Alternatively, users can use a wild-card state to ignore intervening events between specified sequences. Users can also specify time constraints at each step of the sequence. Session Viewer will look for the next event sequence in the pattern and return a match if the found event is within the specified time. Similar to the '+' operator in Unix-style regular-expression matching, each step in the defined event sequence can be matched to a single event in the actual session sequence, or a group of events with the same event state.

Further matching constraints are also available: Session Viewer can limit matching to patterns starting at, or ending with, a landmark, such as the beginning or the end of the session, a common search event, or events with atypical durations (e.g., very long or short events). For example, to limit searches for navigational pattern to sessions containing only the pattern, the first sequence in the pattern should be at the beginning of the session, and the last pattern sequence at the end: S[Start]→R[End]. The use of pattern matching is illustrated with the use-case scenario in Section 7.5.

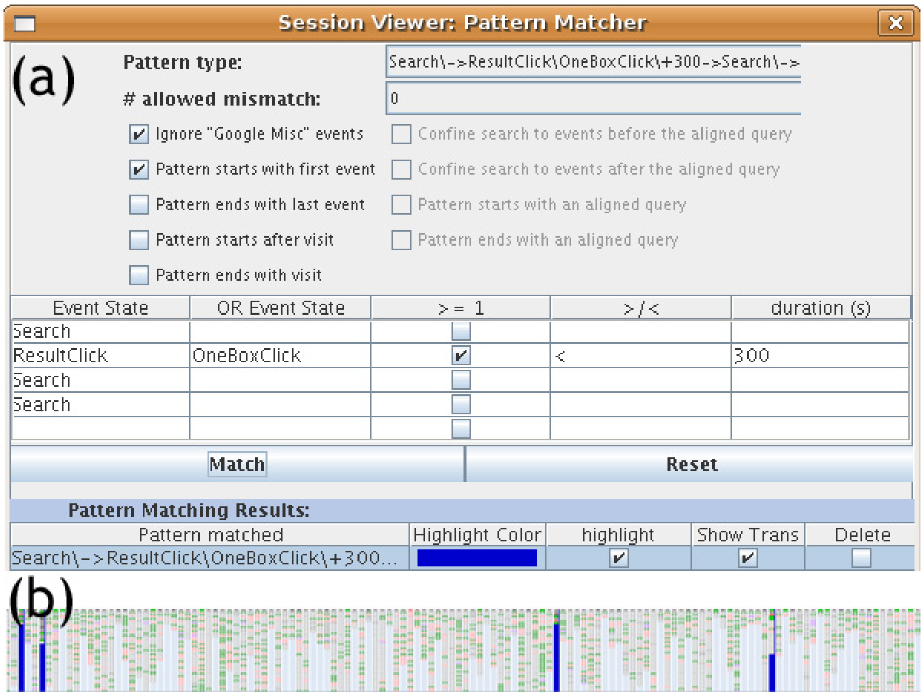


Figure 7.12: Pattern Matcher. (a) Session Viewer's Pattern Matcher dialog box. Users specify event sequences to be matched in the data using a tabular form-filling interface. For each event state, users can also limit the event duration and the number of matches (similar to the "+" operation in regular-expression matching). (b) A sample outcome of the matching, with matched sessions highlighted in blue.

2. Session alignment

Session Viewer by default aligns all sessions based on first events in each session. However, when the data contains a large number of sessions that have a search event with the same query, it is often interesting to compare these sessions based on that search event. Session Viewer provides a profile of queries used across all search events in the data, and the total number of sessions containing those searches. Once the user selects a query from the profile, Session Viewer displays sessions that have at least one search event with the selected query, and aligns these sessions at the first occurrence of this common search event. Figure 7.8(c) shows two examples, where the common search events are squares highlighted in green, annotated with arrows, and aligned in the y-dimension.

3. Session population partitioning

Users can create new session populations from a larger population. Session Viewer provides support for filtering to create new populations, and support for saving and reloading saved populations.

7.4.6 Implementation details

Session Viewer was written in Java using the JRE 1.5.0_06 library and the Java2D graphics library.

7.5 Use-Case Scenario: Exploring the Relationships between Task Type and Search Behaviour

To illustrate how Session Viewer can be used in web session log analysis, we now describe a series of analyses performed on web logs gathered from an external user study. Even though Session Viewer can be used on any session logs, we showcase study data for the rich labels provided by the original logs, such as task instructions, task outcome and user satisfaction.

The study was conducted separately from and before the development of Session Viewer by an independent third-party company called Keynote Systems Inc. (keynote.com). The study recruited close to 400 participants and generated about 6,000 sessions grouped by three experimental factors: search engine type, search domain (e.g., Image, News), and question variant. Question variant includes three defined search tasks, plus one where participants were asked to create their own tasks. A previous analysis grouped the sessions along the three experimental factors and identified two main populations: sessions where participants were given explicit instructions and those when they performed their own tasks (Russell and Grimes 2007). We revisited the data and further refined Russell and Grimes's (2007) results. We focused on tasks with explicit instructions. Inferring from task instructions, we found that different question variants were of different task nature and were not isomorphic, as assumed in Russell and Grimes's (2007) analysis. We further found that task nature (e.g., Directed/closed information search and Locate, a taxonomy from Rose and Levinson 2004) was reflected as usage patterns in logs and could be used to characterize session populations.

To customize Session Viewer for the study logs, we first defined a set of event states based on actions: Search, ResultClick, and NextPage. We did not code events unrelated to the search engine. We also defined a set of session attributes such as Task Time, or the total time duration per session; #Event, or the total number of events per session; and #Search, or the total Search event per session.

We then loaded two session populations with the same search engine and domain, but of different question variants. The center view in Figure 7.13 shows the populations for this question variant, which we called the Camera-Task population:

> Assume you are looking for a digital camera and a friend suggested the Nikon Coolpix 4600. Use <site> to search for information about the Nikon Coolpix 4600. How many megapixels is the image resolution of a Nikon Coolpix 4600 digital camera?

and the right-hand view is the population for this question variant, or the Watch-Task population:

> Assume you are looking for a man's watch as a gift for a friend or family member. Use <site> to search for a man's watch that is water resistant to 100 meters and under $100. What brand of watch did you choose?

We expected the populations to look similar as the tasks were supposedly isomorphic, but they looked different at all data levels. More specifically, the Camera-Task sessions were shorter and had fewer events than the Watch-Task sessions. We could immediately see from the Sessions Panel of the Multiple Pane that the Camera-Task sessions had fewer events (Figure 7.13). To better understand the event-count distribution, we reordered the sessions by dragging the #Events header in the Session Attributes Panel to the top (Figure 7.2).

However, since the total number of events per session may not correspond to session time duration, we focused on the Histogram Panel in the Aggregate Pane to check the distribution of the session attribute, Task Time. As seen from Figure 7.14, most of the sessions in the red Camera-Task population were under 100 seconds, while the yellow Watch-Task sessions were longer and more widely distributed.

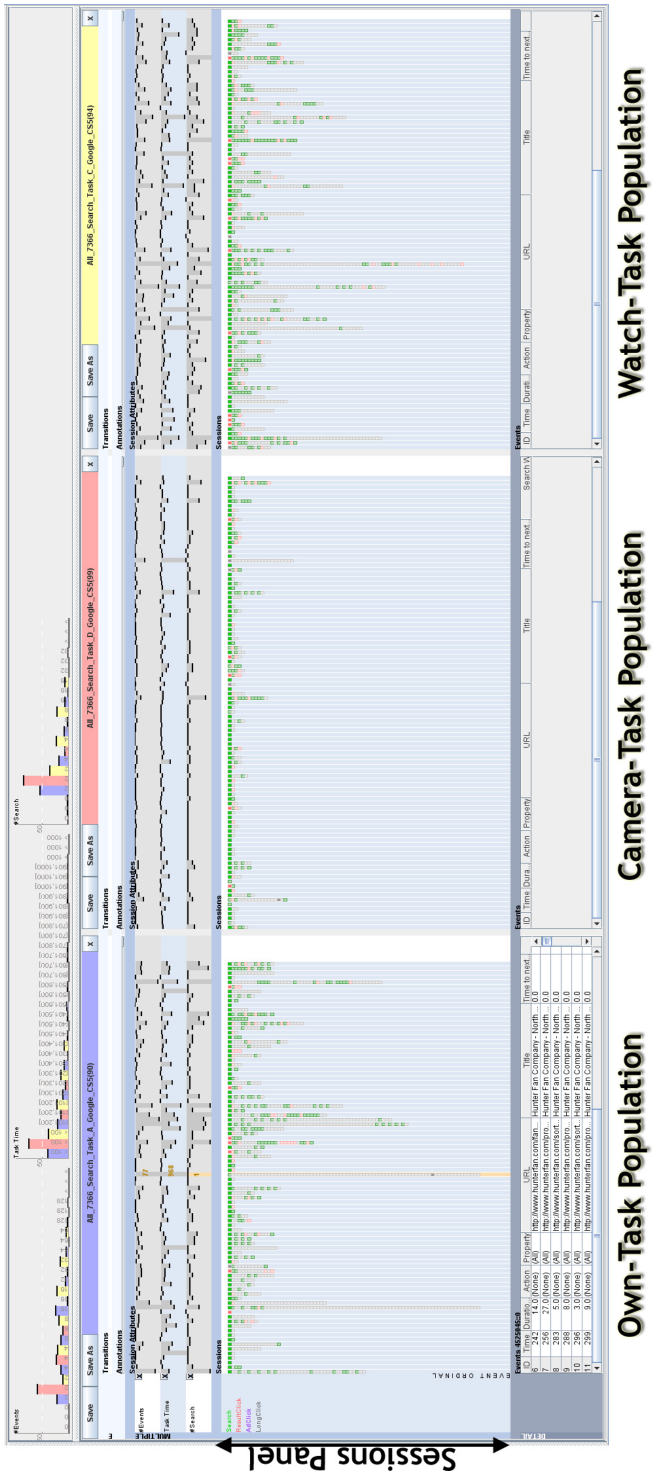


Figure 7.13: Session Viewer showing three session populations: Own Task, Camera Task, and Watch Task. Note that this figure differs from Figure 7.2 as sessions in the Sessions Panel are unsorted.

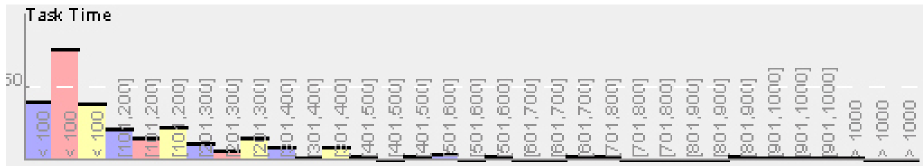


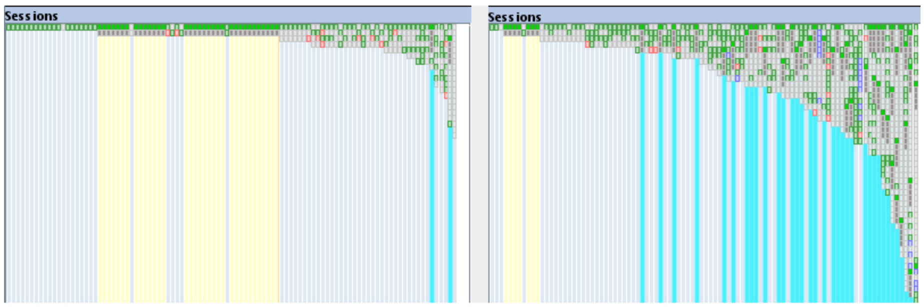
Figure 7.14: A histogram showing distributions of the session attribute, Task Time. Most of the sessions in the red Camera-Task population were under 100 seconds, while the yellow Watch-Task sessions were relatively more widely distributed.

These visual differences convinced us that the Camera and the Watch-Task populations were different. We then focused on identifying the reasons behind these differences, which led us to re-examine the task instructions to understand the task goals. Even though both tasks aimed to find a commercial product, they differed in nature: the Camera task directly looked for a property of a specific object, while the Watch task required exploration as only the properties of the object, rather than a specific identifier, were given. Using the framework for search goals proposed by Rose and Levinson (2004), we classified the Camera task as a Directed/closed informational search, whereas the Watch task is an Informational Locate task.

Once we hypothesized the difference in task nature based on task instructions, we searched for usage patterns within the session populations to better quantify task nature. Side-by-side visual comparison of the event state sequences in the Sessions Panels gave us a lead: different search patterns would be prevalent in session populations of different task types. We visually tested this hypothesis using the event-sequence matching feature in Session Viewer. Using S to denote a Search event and X to denote a non-search engine event, we defined four usage patterns identified in earlier detailed session analyses, which we further refined based on event-state sequences in the Sessions Panels:

1. *Short Navigation*: S[Start]→X[End], with the S event limited to first session events and the X event to last events.
2. *Topic Exploration*: S→X→X→X→X
3. *Methodical Results Exploration*: S→X→S→X→S
4. *Query Refinement*: S→S→S→S

Using the pattern-matching dialog box (Figure 7.12), we defined these patterns and highlighted the Short Navigation sessions in yellow and the Topic



(a) Camera Task Population (b) Watch Task Population

Figure 7.15: Sessions Panels for two task types. Short Navigation sessions are highlighted in yellow, and those with the Topic Exploration pattern are highlighted in aqua.

Exploration sessions in aqua. Figure 7.15 clearly shows that Short Navigation searches were more prevalent in the Directed/closed Camera population, while the Topic Exploration pattern was more common in the exploratory Locate Watch population. Encouraged by the visual differences, we highlighted the other search-behaviour patterns and observed similar results.

To test our visual finding on the entire study data, we manually labeled all other question variants and repeated the analysis with an external statistics package. As shown in Figure 7.16, our hypothesis was confirmed that task nature, as inferred by task descriptions, was reflected in usage patterns. In general, only 14% of the exploratory Locate-type tasks were Short Navigations compared to 37% in Directed/closed-type tasks. List-type tasks and undirected information searches were more similar in composition to Locate-type tasks than to Directed/closed-type tasks. As in the previous analysis, we also concluded that participants were more exploratory in their own tasks, as they were visually more similar to the exploratory Watch tasks than the Directed/closed Camera task at all data levels, as shown in Figure 7.2.

While the Methodical Results Exploration and Query Refinement patterns were understandably present in exploratory sessions, we wondered what participants were doing in those non-search-engine X events in the Topic Exploration sessions. To answer that question, we selected longer and more involved Topic Exploration sessions for detailed examination in the Events Panel.

To locate such sessions, we sorted the sessions again by the #Events attribute, and then focused on the high end of the distribution in the Sessions Panel. In Figure 7.2, we expanded the session with the largest event count

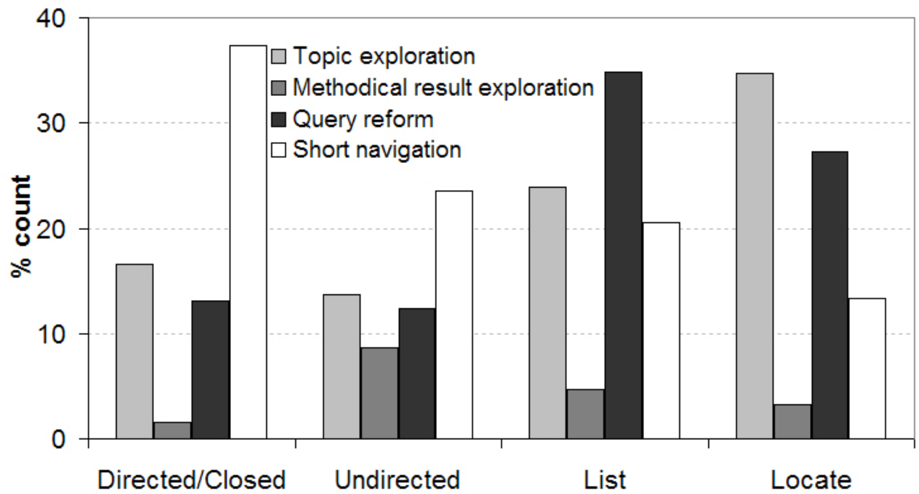


Figure 7.16: Confirming a hypothesis formed by exploration in Session Viewer: sessions of Directed/closed information tasks contain significantly more Short Navigational patterns than sessions of Undirected, List, or Locate task types.

in the Watch-Task population in two-dimensional form to better study usage behaviour. We found a strange pattern: the first half contained mostly non-search-engine events colored in gray while the second half contained mostly NextPage events colored in blue. In the first half, long sequences of events were punctuated by green Search events in the same horizontal lane, meaning the Search events had the same URL. To better understand this behaviour, we examined the individual events in the Events Panel and found two main search strategies. In the first half of the session, the participant used the search engine (Search events in green) to reach third-party websites such as amazon.com, walgreen.com and shopping.msn.com, and searched within those shopping sites (uncoded events in gray). In the second half, the participant used domain-specific searches (Froogle) that involved mostly NextPage events in blue. We were intrigued by the first half of the session, where participants used the search engine as a launching pad for exploration within third-party websites.

To determine if the behaviour was unique to this participant, we expanded several sessions and saw similar behaviours: search-engine searches (columns of green boxes, annotated with SS) punctuated by third-party sites searches (columns of gray boxes, annotated with TS) and true result explorations within these sites (diagonal gray boxes, annotated with TE), as shown in Figure 7.9.

This finding suggests a need to differentiate between this new Nested Search pattern and true Topic Exploration.

In summary, Session Viewer's side-by-side population comparison at multiple levels allowed us to quickly spot differences between the Watch-Task and the Camera-Task populations, which we had originally assumed to be the same. The pattern-matching feature allowed us to quickly test a hypothesis that the relative prevalence of different event sequences would be an important feature for characterizing different session populations. Session reordering guided our selection of interesting sessions for detail event-by-event examinations, where we discovered the Nested Search usage pattern.

Our experience in using Session Viewer for web session log analysis was positive. We further examined our design choices for Session Viewer and investigated visualization use in the workplace with a field evaluation conducted at Google Inc., which is the subject of the next chapter.

7.6 Design Evolution

The design presented in Section 7.4 in this chapter is the third version of Session Viewer, SV3. As described in Section 7.2, Session Viewer was developed based on the user-centered and participatory design approaches. Initially, one of Session Viewer's target users, a detailed-session analyst, was involved in the design of the first version (SV1). The addition of a statistical-aggregate analyst to our design team resulted in the second version (SV2). Based on target users feedback, we underwent a third round of design revisions before finalizing our design as SV3.

In this section, we describe our design evolution focusing on major design choices made during the process. We first explain SV1 (Section 7.6.1) and SV2 (Section 7.6.2) to ground our reflections on design evolution, and discuss four major themes in the changes: (1) using a scrollable Multiple Pane to provide sessions-level overview (Section 7.6.3); (2) providing visualizations for aggregate-level overviews (Section 7.6.4); (3) taking the multiple-coordinated approach to show multi-level data (Section 7.6.5); and (4) vertically stacking panels to facilitate comparisons between session populations (Section 7.6.6). Reflections on these choices based on our field evaluation observations are delayed until Chapter 8.

Figure 7.17: First version of Session Viewer (SV1), showing four panels (from top left and counter-clockwise): Session Attribute, Sessions, Detail, and a Dynamic Query Panel showing session-attribute ranges or categories.

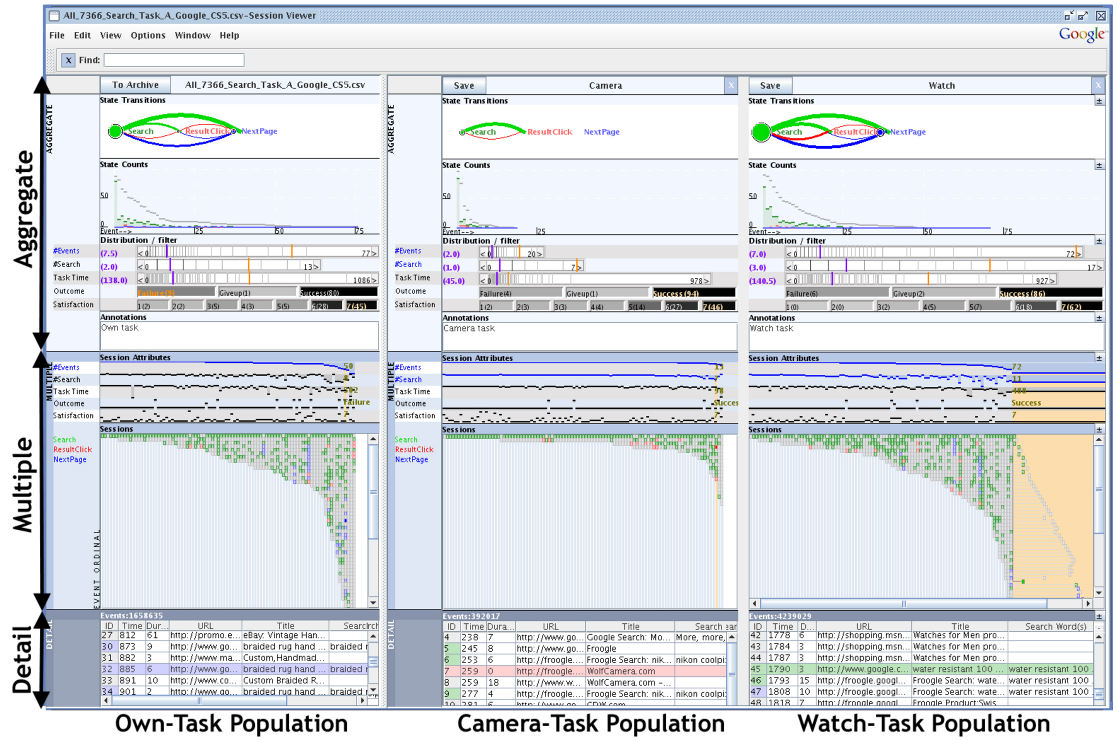


Figure 7.18: Second version of Session Viewer (SV2), showing the same three session populations as in Figure 7.2. The major changes between this and the last version are in the Aggregate Pane, where only the state-transition diagrams and the Annotation Panels are retained.

7.6.1 SV1: basic design

The first version, SV1, had a simple interface with only four panels: Session Attributes, Sessions, Events, and Dynamic Query (Figure 7.17). SV1 was designed mainly for detailed-session analysis of a single session population.

SV1 had the basic design of the Multiple Pane in SV3 on the left of the interface, with identical visualization and interactions as discussed in Section 7.4.2.

The right side of the interface housed two panes: Dynamic Query and Events. The Events Pane of SV1 was similar to the Details Pane in SV3, as discussed in Section 7.4.3, except it included a table of session attributes and corresponding values at the top of the pane, in addition to the SV3 Events Panel.

The Dynamic Query Panel in SV1 was not included in SV3. The panel displayed ranges or categories of session attributes for data filtering, modeled after Hochheiser and Shneiderman's (2003) dynamic query visualization. For continuous session attributes, users could adjust the range of a session attribute using a double-slider bar to control which sessions are displayed in the Sessions Panel. For categorical session attributes, each category had a set of grey toggle boxes labeled with category names and the number of displayed sessions corresponding to that category name. Users could include or exclude sessions in selected categories by clicking on the boxes.

7.6.2 SV2: supporting multiple populations

Since SV1 only showed one session population, we often had to launch multiple copies of SV1 to compare between session populations and to manually dissect populations. Even with multiple monitors, we found the comparisons difficult as it was difficult to arrange the all panels side-by-side. We therefore built the second version, SV2, to better support comparisons of multiple session populations at all data levels (Figure 7.18). As in SV3, SV2 had three panes: Aggregate, Multiple, and Detail. The visualization and interactions were similar to those in SV3 described in Chapter 7, except for the Aggregate Pane.

In SV2, the Aggregate Pane had four panels showing population metrics and distributions: the State Transitions Panel, the State Counts Panel, the Distribution/Filter Panel, and the Annotations Panel. While the Annotation Panel was identical to the one in SV3 (Figure 7.4), the rest of the panels were different.

The SV2 State Transitions Panel were visually identical to that in SV3 (Figure 7.6), except the transitions were based on event states instead of patterns.

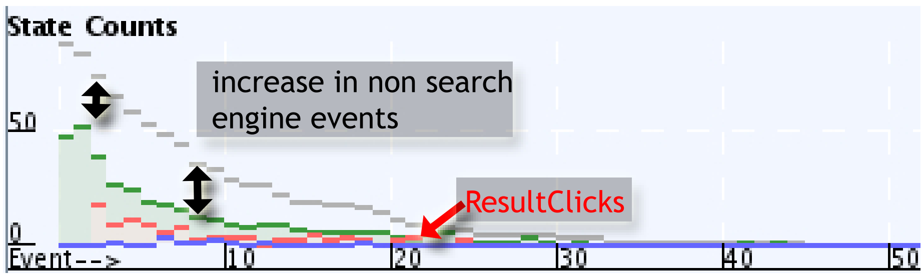


Figure 7.19: SV2 State Counts Panel was a bar chart with event ordinality on the x-axis and event counts on the y-axis for all events (gray bars) and individual event states (stacked and color-coded). Here, the later events were predominantly non-search engine events, as indicated by the increasingly large gap between the total event counts and coded event state counts.

SV2's State Transitions Panel was originally designed to detect unexpected event sequences and states, such as Search events after a long delay that may indicate user goal change.

SV2's State Counts Panel was a stacked bar chart with the x-axis being session event ordinal and the y-axis as event state counts (Figure 7.19). The panel was designed to monitor relative event state prevalence over the course of the sessions. For example, Figure 7.19 shows that while some of the initial decline in the green Search events was due to an increase in red ResultClick events, later events were mostly uncoded non-search-engine events, suggesting exploration.

The SV2 Distribution/Filter Panel evolved from SV1's Dynamic Query Panel and displayed ranges or categories of session attributes for data filtering (Figure 7.20). Users could filter continuous data using the double-slider bars. Filtering was guided by the stripe graphs, or vertical lines on the SV1 Dynamic Query slider bars which showed distributions. Median values were shown as purple vertical stripes and also as text on the right. As in SV1's Dynamic Query Panel, SV2's Distribution/Filter Panel displayed categorical session attributes as toggle buttons, which also provided categorical data filtering guided by the button labels that showed the categories session counts.

We now discuss the rationale behind these changes in four design discussions. The first two involve overview creation, and the last two involve spatial layout.

Figure 7.20: SV2 Distribution/Filter Panel. The continuous attribute Task Time was represented by a double-slider bar for filtering with the distribution displayed as a stripe graph on the slider bar. The categorical attribute Task Outcome was represented by a series of toggle buttons for filtering, with the category name and counts as labels, and the counts encoded with luminance in the button background.

7.6.3 Scrollable Multiple Pane and session partitioning to provide sessions-level overview

The Multiple Pane provides sessions-level overview in Session Viewer. The challenge in creating such an overview was to provide enough information for analysts to select interesting sessions for further studies in the Events Panel, without severely compromising display capacity. This is the classic challenge of information visualization.

Our summary synthesis informed us that in order for low-VIR overviews to be useful, we needed to provide sufficient task-relevant information (Section 4.4). In our case, we decided that session logs analysis required details at the event level. Also, we needed to display each event as simple visual objects, as guided by our overview-use study results in Chapter 6.

These requirements severely limited the number of displayable sessions and data dimensions. To understand the number of displayable sessions, we looked at a duo 1920x1200-pixel monitor setup, which was typical for our participants. If we were to encode each event by a single pixel, we could accommodate at most 3840 sessions each with at most 1200 events. While this top limit may be feasible visually, we needed a larger interaction target for our users to effectively select sessions and events without resorting to additional magnification tools. In fact, we found in our experience that we needed at least 5x5 pixels for each event rectangle, thus reducing the number of visible sessions five-fold. The number of data dimensions encoded on our event graphical object was also limited, as our overview-use study found that visually-complex objects were difficult to discern and use, even when the extra visual details were available at higher VIRs (Section 6.3). As a result, we forewent the use of glyph-like visual objects to increase the number of dimensions encoded, discussed in our Related Work

Section 3.2.1.

Given that the amount of sessions under analysis could be in the order of millions, we considered it impossible to display every event and their attributes available in the data. In other words, we needed to selectively display sessions, instead of displaying all available sessions. We also needed to selectively display event attributes.

We considered two methods to automatically select sessions: to provide scrolling in the Multiple Pane and and thus, only showed part of the Sessions-Panel overview at any one time; or to cluster the sessions prior to display and show sample sessions from each cluster as different populations. The latter approach was successfully applied to show time-series data by van Wijk and van Selow (1999) and advocated by Keim et al. (2006).

In SV1, we decided on scrolling since our target users were detailed-session analysts who tended to study hundreds of sessions in detail (Section 7.1). Also, the layout of SV1 enabled users to devote a large screen area to the Multiple Pane (Figure 7.17). Given that most of our target users had two 1920x1200-pixel monitor setup in their workplace, our detailed-session analyst could display most of his study sessions over the two screens.

In SV2, however, we needed to display multiple session populations. Even over two wide screens, we frequently observed the need for both vertical and horizontal scroll bars in the Multiple Pane. We thus revisited the design choice for scrolling over clustering while designing SV2.

While we found the idea of clustering attractive, it was difficult to preprocess our rich web session logs. Clustering criteria are likely to change with analysis focus. For example, in studying usage behaviours in queries using advanced search features such as double quotes, the clusters may depend on the type of advanced feature used. For analysis on modeling event transitions, the cluster feature may be event-sequence transitions. Also, our summary synthesis informed us that *a priori* data selection in overview creation may be a double-edged sword, as increased display capability may come with costs in user confusion and mistrust (Section 4.4.4).

Given this diversity, we decided against automatic session selection and allowed scrolling in the Multiple Pane, even though our users cannot view all sessions in the Multiple-Pane overview. Section 8.2.1 reflects on our choice of scrolling over automatic session preprocessing based on field evaluation results, which supported our decision.

In addition to filtering, we also implemented a means for users to manually manage sessions: by partitioning them into multiple session populations and view them selectively, and by filtering uninterested sessions to focus analysis. Session partitioning, discussed in Section 7.4.5, was introduced in SV2 and was retained unchanged in SV3. Our field evaluation did not find uses of session partitioning. Section 8.2.1 proposes a few explanations.

We took a similar approach in selecting event attributes for display: using only colours of the displayed event objects encoded user-defined event states that could be changed during analysis, and box length to encode event duration or ordinal (Section 7.4.2; Figure 7.8). Section 8.2.1 comments on our choice of encoding only one dimension, the event state, on the event graphical object in addition to time or event ordinal. Our field evaluation results supported our decision, as seen in the effectiveness of session selection by our analysts, discussed in the context of the design choice of *separate* over *embedded*.

Issues in overview creation in multiple-VIR interface design are further discussed in Section 9.1.1 of the last chapter of this thesis as one of the open research questions.

7.6.4 Visualizations to provide aggregate-level overviews

To support multiple-population analyses, we needed to develop aggregate visualization to better facilitate population comparisons.

SV1 provided aggregate information in the Dynamic Query Panel to support filtering, where the ranges of session attributes were shown numerically at the far right and left of each slider bar, and visually by the length of the slider bar. For categorical session attributes, SV1 showed the categories as boxes, labeled with the category name and the number of sessions belonging to each category.

In designing SV2, we enriched SV1's Dynamic Query Panel by adding stripe graphs on the slider bars. The goal was to better guide session filtering by showing session-attribute distributions, similar to the idea of scented widgets (Willett et al. 2007). The result was SV2's Filter/Distribution Panel. Unfortunately, initial testing of these panels in SV2 with selected target users was not encouraging. Even though they did use the Filter/Distribution Panel, most testers were confused by the distribution stripe graphs since they were more familiar with, and expected, histograms. In addition, interacting with the narrow dynamic-query slider bars was found to be difficult. In SV3, we therefore converted SV2's Filter/Distribution Panel to the Histogram Panel (Figure 7.3). Our motivation was three-fold. First, histograms were more familiar to our target users; second, using histograms would allow us to use a consistent visual representation for both categorical and continuous session attributes; and third, toggle-clicking on histogram bars was found to be physically simpler than dragging slider bars. However, histogram bar filtering on continuous session attributes is not precise, since each histogram bar usually represents a range of values. Section 8.2.1 discusses our design choice of using histogram bars to show aggregate session attribute distributions in the context of our field evaluation result, where we found two of our seven study participants actively used the histogram bars for session filtering.

We also added two aggregate visualizations in SV2: the State Transition Panel and the State Counts Panel. These panels were based on basic population statistical plots used in analyzing usability logs, such as state-transition diagrams and event-count plots, as discussed in the Related Work Section 3.2.4. However, initial user feedback was not positive: most of our testers ignored the event-state transition and the event-count diagrams. One tester explained that while in theory, event transitions could be very informative, there was too much noise in raw logs for the State Transition Panel to reveal transitions patterns. In SV3, we used the same visual representation to display pattern transitions instead of event state transitions to work around the data-noise problem. Since pattern definitions are more flexible, as Session Viewer allows wild-card and alternative event states in pattern matching (Section 7.4.5), patterns have a better chance to extract true usage behaviours. The choice to display pattern instead of event state transitions is discussion in Section 8.2.1, where we speculate on the lack of use of the Transitions Panel in our field evaluation.

In creating SV3, we discarded the State Counts Panel since our testers did not find it useful.

7.6.5 Multiple coordinated view to show multi-level data

Since session log data is a multi-level data object at the population, session, and event levels (Section 2.4.1), we displayed these levels as different visual information resolutions, in accordance to our findings in our summary synthesis (Section 4.3).

To bridge between the detailed-session and statistical-aggregate analysis levels, we decided to display all data levels simultaneously to facilitate cross-level activities such as reordering sessions in the Sessions Panel based on session attributes displayed in the Session Attributes Panel. As discussed in our decision tree in our summary synthesis (Figure 4.1; Section 4.6), we had two choices to display multiple levels of data simultaneously: *embedded* (putting all the levels into a single view) or *separate* (showing them as separate views).

Embedded techniques provide fluid interchange of the high-detail focus and background context in the same view, an advantage over the more commonly used *separate* techniques, which tend to incur the problem of view coordination. Integrating data views has also been argued to better support perception and evaluation of complex situations by helping analysts to perceptually and cognitively integrate multiple separate elements (Thomas and Cook 2005, p. 83). In our original paper-prototype design, we used a one-dimensional fisheye view with distortion in the vertical direction to display sessions in the Sessions Panel. Unfortunately, image distortion is frequently required to achieve the multiple visual information resolution in conventional displays. Using distortion in our case would be inappropriate as our analysts needed to estimate event durations based on displayed box length in the Sessions Panel, and distortion may make distance judgement difficult (Carpendale et al. 1997). In addition, integrating the various data levels in a single view in our case would be non-trivial, since we believe data levels in the web log data object as too inter-linked to be embedded in a single view using simple visual representations and interactions, especially given our considerations in visual memory costs incurred by image transformations (Chapter 5).

We therefore chose the multiple coordinated view approach to display the various data panels as separate views. The last design choice discussion in Section 8.2.1 reflects on our choice of taking the *separate* overview + detail approach over the *embedded* focus + context alternative.

7.6.6 Vertically-stacking multiple panel views for population comparisons

Session Viewer has a number of panels that are semantically related. Ideally, these semantic relationships should also be visually depicted. At the session-attributes level, the Session Attributes Panel is linked to the panel that provides filtering based on session attributes, such as the Dynamic Query Panel in SV1, the Distribution/Filter Panel in SV2, and the Histogram Panel in SV3. The two panels in the Multiple Pane are linked at the sessions level, where the session data are displayed in the Sessions Panel, the corresponding attributes are displayed in the Session Attributes Panel. At the event level, both the

Sessions Panel and the Events Panel display individual events.

It was easier to provide visual correlations between panels in SV1, since it only supported single-population analyses. At the session-attribute level, the Session Attributes Panel and the Dynamic Query Panel were aligned vertically such that each horizontal row housed the same session attribute (Figure 7.17). The Session Attribute Panel and the Sessions Panel were aligned horizontally such that each session occupied an individual vertical lane. While the event boxes in the Sessions Panel did not physically align with the event detail entries in the Events Panel table, the two event representations were at least visually congruent, where events were displayed in a linear order and the top event is at the top of the display, and the two panels were vertically aligned.

In designing SV2, we needed to display multiple session populations and considered three design alternatives: (1) display small multiples of SV1, (2) display a free-style workspace using the sketch-book metaphor, and (3) displays populations in vertical views for side-by-side population comparisons across all data levels.

Showing multiple SV1-like displays has the advantage of preserving the spatial layout of SV1, which intricately linked the four panels based on semantic relationships between visual objects on the panels as described above. However, comparing between populations would have been difficult and it would be difficult to compare between populations at all data levels. We thus abandoned the idea.

In an earlier unimplemented paper-prototype design, we used the sketch-book metaphor and envisioned a free-style workspace where users could drag and drop interesting panels and directly annotate on the workspace. We eventually abandoned that design, as we believed users could better mentally process the data if the same data panels were displayed for all populations and were arranged to reflect the data hierarchy. Our decision was based on two design principles. The first considered principle was the Naturalness Principle of Norman (1993), where he stated that experiential cognition is most effective when the represented information, in our case the three-level session log (population, session, and event), is closely matched by the visual representation, in our case the three-level panes (Aggregate, Multiple, and Details). The second design principle we considered was Baldonado et al.'s (2000) *rule of consistency* in their guidelines for multiple-view use in visualizations. We believed that since in data explorations where analysis freely moves between different levels of detail, having a consistent display would be cognitively less demanding and would allow the analyst to focus on the task at hand, rather than mentally organizing the data displays.

We therefore stacked all the panels vertically and arranged session populations horizontally to facilitate side-by-side comparison, and implemented a rigid layout to reduce cognitive load during analysis. These two choices have three design implications:

1. Providing a rigid layout reduces user control, which may hinder tool use. However, study observations did not provide evidence of this problem. The first design-choice discussion in Section 8.2.3 covers the choice between spatial consistency and user control.

2. While we kept the visual linking between the Session-Attributes and Sessions Panels by showing data from each session in a different vertical lane, we sacrificed SV1's spatial association between the Session Attributes and the Dynamic Query Panels. In SV2, we stacked the panels and put the Distribution/Filter Panel, SV2's version of the Dynamic Query Panel, on top of the Session Attributes Panel to allow for side-by-side comparisons between populations. This may render view coordination between the Sessions Panel and the Events Panel more difficult, a difficulty we indeed observed, as discussed in the design choice of taking the *separate* over the *embedded* approach in presenting the different panels in Section 8.2.1.

3. In terms of space efficiency, we believed SV1 to be better suited for displaying a single session population on conventional landscape monitor setup, as the Sessions Panel and the Events Panel, the two panels that probably required the most vertical space and should be displayed together, were horizontally arranged. However, we believe SV3 is better suited for multiple population analyses. Our beliefs were confirmed in the field evaluation. The design-choice discussion in Section 8.2.3 looks at effects of our choice to vertically stack panels into views to facilitate comparisons between session populations instead of displaying small multiples of SV1, where we found that perhaps our decision had compromised using Session Viewer for single-population analysis.

Chapter 8

Field Evaluation: Session Viewer at Work

As shown in our use-case scenario in Section 7.5, our experience in using Session Viewer for web session log analysis was positive and fruitful, as we discovered new insights based on task type in characterizing session populations. However, to better study design choices of Session Viewer and to discover deployment issues, we needed to investigate its use in ecologically-valid settings where the prototype could be used by a variety of analysts with diverse data and analysis styles.

Also, as discussed in Section 7.6, we made four main design choices in creating Session Viewer: (1) allowing scrolling sessions-level overview in the Multiple Pane; (2) using histograms and a pattern transitions diagram to provide aggregate-level overviews in the Aggregate Pane; (3) taking the multiple coordinated views approach with *separate* techniques to show the multiple levels of data; and (4) providing a fairly rigid panel layout with session populations displayed as views. These decisions were made based in part on our knowledge and experience gained in the first three studies, where we investigated various aspects of using multiple visual information resolution interfaces to display data. We needed to examine these choices.

We therefore conducted a field evaluation at Google Inc. The goal of the study was therefore two-fold: (1) to evaluate our design choices in making Session Viewer; and (2) to understand issues of visual analytic tool deployment in the workplace.

Over a period of two months, we observed 20 hours of tool use by seven web session log analysts. To ensure ecological validity and to keep participants engaged and motivated, our participants used their own data performing their own tasks in all of the training and actual study sessions.

We framed our findings as design themes:

1. Design implications of real-world data:

   (a) *Tool needs to convey the gist of the data quickly.* Data validation was found to be integral in analysis given the complex and noisy real-world data. We found that the scrollable overview and the *separate* (overview + detail) visualization technique worked well here.

   (b) *Designers need to base data-field configurability on user skills.* Since web session logs have established data fields, we originally assumed that Session Viewer could pre-configure data fields to ease tool setup costs. However, our engineering-centric users preferred a completely open interface to customize data-field definitions.

   (c) *Tool needs to support fluid data-view projections.* Since data patterns were seldom revealed in casual explorations with a few data views, our participants had to constantly refine their analysis goals and questions, each with a slightly different data view. We found that our rigid layout did not hinder tool use and may have reduced cognitive load during analysis.

2. Factors that lead to tool reception in the workplace:

   (a) *Tool needs to play a unique role in the analysis process.* As a visual analytic tool, Session Viewer supports a wide spectrum of data exploratory tasks, but plays a very specific and unique analysis role: it provides a missing link between the detailed-session and statistical-aggregate analysis approaches.

   (b) *Data transfer is not as crucial as assumed.* To our surprise, idea transfer, rather than direct data transfer, was the norm in our study.

   (c) *Tool power and complexity is a tradeoff.* Even though Session Viewer's flexibility attracted users, tool power came with a cost of complexity which may deter tool use for some users.

In this chapter, we illustrate each finding in the two design themes with observations collected from the field evaluation, followed by a design guideline derived from our observations. We also further explore each finding and design guideline pair in the context of design choices made during the creation of Session Viewer, and deployment issues encountered in our study.

8.1 Study

We collected data from interviews and think-aloud comments from participants as they analyzed their data using Session Viewer. Participants were recruited by e-mail invitations. Important requests from participants were implemented during the study, which included bug fixes and enhanced flexibility of existing tool features.

8.1.1 Participants

We recruited seven data analysts at Google Inc., three of them female, who routinely analyze large data sets. Based on the seven pre-study and five interviews conducted earlier to solicit design requirements discussed in Section 7.1, we identified two analysis approaches: detailed session and statistical aggregate. Among our seven participants, coded as P1 to P7, P1 is a detailed-session analyst, P3 uses both methods, and the rest are statistical-aggregate analysts. Two of our participants, P1 and P2, were involved in the design of Session Viewer, as discussed in our design process section (Section 7.2). Our two participant-designers and one of our participants were also previously interviewed, but were re-interviewed for the study since their analysis practices may have evolved since initial design a year prior.

Characteristics of the two analysis approaches found during Session Viewer design were described earlier in Section 7.1. Our second round of interviews yielded similar results. To reiterate:

Detailed-session analysis aims to answer specific but open-ended questions about usage behaviour, such as the use of Boolean OR in queries, or to develop standard metrics to measure task nature and user satisfaction. This description is based on four interviews.

Statistical-aggregate analysis also aims to understand usage behaviour, but at the aggregate level by comparing and characterizing different session populations based on established metrics. This description is based on eight interviews.

P1 is a Research Scientist in the area of search quality, with a focus on understanding user behaviours in web search. P1 is an experienced detailed-session analyst with over 15 years of experience, with four years in usability logs and two in web logs. Most of his analysis data came from field studies collected with client-side logging with installed plug-ins on his searchers' browsers. P1 analyzes by examining the sessions event-by-event in an Excel spreadsheet to look for usage patterns and to generate hypotheses. Once he feels confident with a hypothesis, he uses SPSS and Visual Basic to calculate aggregate statistics.

P2 is a Software Engineer with broad analysis interests, but with most analyses examining specific sessions as part of an experiment. One example of P2's experiments is to test for the effects of background colour of the sponsored link display region on the Google result page. P2 is an experienced statistical-aggregate analyst with over seven years of log-analysis experience, and four in web session logs. Generally, P2 begins the analysis by first processing relevant sessions into forms suitable for her custom-built scripts to calculate statistical metrics such as click-through rate, which measures the number of users who clicked on individual links on the general search result page. Metrics from different experimental populations are then compared to evaluate factor effects, such as the different colours tested. In some cases, P2 may regroup or further dissect populations to tease out distinctions.

P3 is a Software Engineer whose main analysis goal is to understand usage behaviours in location-specific queries that involve Google maps. P3 has 10 years of general analysis experience, with two years in logs and one year in session logs. P3 describes his session log analysis as "monitoring-like", aiming to find and understand inappropriate map search result inclusions in the main search-result page, and missing map results that should have been included. He uses an internal tool for statistical-aggregate analysis, where he selects interesting sessions with pre-defined filters, such as time stamp, and calculates metrics, such as click-through rate. On occasions, P3 augments his statistical tools with an in-house detailed-session analysis tool that displays session details to better understand search behaviours. However, P3 feels that his sets of tools lack coverage and are not well integrated to support his analytical needs.

P4 is a Software Engineer and an experienced statistical-aggregate analyst with over 13 years of analysis experience, with three in logs and two in session logs. His analysis goals are to observe behavioral differences between experimental populations and to detect subpopulations. One such example is to vary the type of advertisements shown on the search result page based on query keywords. Since existing in-house tools do not fully support exploratory definitions of subpopulations, P4's team has built tools to allow more flexible regrouping of sessions based on session attributes. P4 feels he needs to look at sessions at a more detailed level to better understand search behaviour.

Both P5 and P6 are Quantitative Analysts who focus at the statistical-aggregate analysis level. The main goal of their analyses is to characterize and understand search behaviour based on pre-defined metrics in experimental data, for example, looking at click-through rates of advertisements displayed on search result pages based on display positions and the total number of advertisements shown on the page. P5 is more experienced with over eight years in general analysis, and P6 is more junior, with about one year of experience. Both analysts use in-house scripting, statistics, and graphing tools. P5 also uses R, a statistics package, for data analysis.

P7 is a User Experience Researcher who has at least four year of analysis experience with web session logs. The main goal of her analyses is to characterize session populations by search behaviours. P7 is a statistical-aggregate analyst and uses SPSS for data analysis. In her previous field study using session logs collected with client-side logging, P7 characterized web usage behaviours of various web information seeking tasks, such as browsing and fact finding, based on dwell time, number of pages viewed, and the use of specific browser navigation mechanisms (e.g., bookmarks and history).

8.1.2 Procedure

Participants used Session Viewer on their own tasks and data. Each study session began with participants explaining their data and analysis goals. Participants were asked to verbally explain their actions during analysis. In addition, we also prompted participants to clarify verbal comments and actions and to provide reflective debrief summarizations at the end of the sessions. The study had three stages:

1. *Pre-evaluation interview*: a one-hour semi-structured interview with one experimenter to understand our participants' current practices. The interview covered topics such as routine analysis goals, processes and workflow, and tool use. The script used for the pre-study interviews is included in Appendix E.

2. *Training*: a one-hour one-on-one tutorial for each participant to learn to use the tool. To ensure realism and analysis engagement, all participants used their own data and tasks for the training sessions.

3. *Actual analysis*: two to eight one-hour data-analysis sessions with each participant, with a median of two sessions. These sessions were conducted

Participant(s)	Number of Actual Sessions
P1	8
P2 + P4	1
P3	3
P4 + P5	2
P6	3
P7	1

Table 8.1: Number of actual sessions for each participant. Three sessions involved two participants working together, while the rest had only one participant.

either bi-weekly or weekly depending on the participant's availability. Participants were asked to prepare analysis goals prior to the sessions. Since we were interested in expert behaviour, the primary researcher was present at all sessions to answer tool-related questions. Three sessions involved two participants working together, while the rest had only one participant. Table 8.1 shows the session count for each participant.

We conducted pre-study interviews before the training and actual analysis sessions. The script for the interviews is included in Appendix E. All participants had access to the software outside of the study but their uses were not monitored or recorded.

8.1.3 Setting and apparatus

Ideally, study sessions should take place in participants' office to better reflect actual working environments. However, we considered the think-aloud protocol too disruptive to fellow coworkers in Google's open working environment and conducted all except P1's sessions in a design office with a Pentium M 2.13GHz laptop with 1 GB of RAM running Windows XP in a dual-monitor setup (Figure 8.1(b)). The left laptop monitor, with 1280x1024 pixels, was used mainly to display recreated webpages in a separate web browser and Session Viewer dialog boxes, such as the Pattern-Matching dialog box (Figure 7.12). The 1920x1200-pixel monitor displayed the main Session Viewer window.

Since P1 had a mostly vacant office, we conducted his sessions there using a 2.5GHz desktop with 2.0 GB of RAM running Linux RedHat and two 1920x1200-pixel monitors, as shown in Figure 8.1(a).

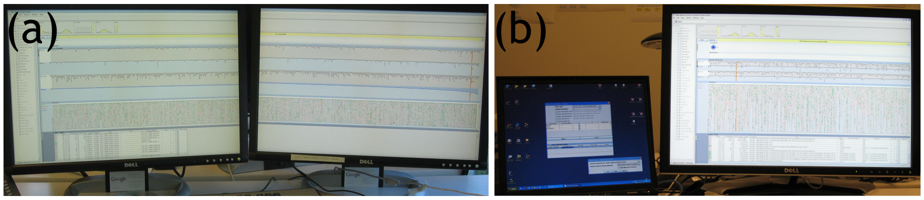


Figure 8.1: Study setups. (a) P1 used two 1920x1200-pixel monitors for the main Session Viewer window. (b) Other participants used a laptop/monitor setup, with the main Session Viewer screen on the 1920x1200-pixel monitor, and auxiliary windows, such as web browsers and Session Viewer dialog boxes, on the 1280x1024-pixel laptop.

8.1.4 Data collection and analysis

All 18 sessions with Session Viewer in use were video recorded. Table 8.1 shows session-count distribution for the seven participants. Our analysis focused primarily on these video recordings.

The first step in our analysis was to code the qualitative video data. Given our goal to discover tool-use patterns in data analysis, we decided to take a more data-driven and open method instead of pre-developing a set of expected behaviours as our coding scheme. Our approach was inspired by open and axial codings in grounded theory (Strauss and Corbin 1998; Corbin and Strauss 2007) and inductive coding in thematic analysis (Boyatzis 1998).

We first coded the video data into units of episodes. Episodes were meaningful data analysis sequences performed by participants and typically began with verbal statements from participants about analysis goals and ended with participants' reflections on the analyses. Since these analysis episodes were well bounded, only the primary experimenter was involved in their extractions from the video recordings. On average, we extracted two episodes from each study session. For each episode, we identified three types of information:

1. *Analysis*: initial goals or hypotheses, analysis strategies, generated or refined hypotheses during analysis sessions, observation or conclusions about the data under analysis, and any action items resulted from the analysis sessions. Examples include analysis goals (e.g., “I am interested in advanced search behaviours”), strategies (e.g., “Let's find sessions with advanced search in them”), and observations of the data set (e.g., “A lot of sessions with advanced search have ‘site:”’ and “Looks like people who use ‘site:’ are good searchers”). This type of information was coded based

on participants' verbal comments;

2. *Session Viewer use*: panel or feature used during analysis such as highlighting sessions with "site:" use. This type of information was coded based on observations of participants' interactions with Session Viewer and data displayed on Session Viewer;

3. *Session Viewer feedback*: participants' comments on Session Viewer. We further annotated this feedback as either impeding or facilitating our participants' analysis processes. This type of information was coded based on participants' verbal comments;

Once extracted, we assigned a code to each episode to describe the general nature of the analysis based on the associated analysis-type information. These codes were derived iteratively where we added, deleted, refined, or grouped codes as we assigned them to episodes. The final set of codes were reported as **findings**, where we only included major codes with at least three episodes with more than two participants.

To facilitate discussion, we further grouped our findings into two **design themes**: (1) implications in working with real-world data, and (2) factors that seemed to determine our tool's initial reception. For each finding, we also derived more general design statements as **guidelines**.

For each finding, if the use-type information involved tool components featured in our design choices identified during our design evolution (Section 7.6), we explored feedback-type information framed by our **design choices**. In cases where the feedback-type information was related to tool deployment, we discussed them as **deployment issues**.

8.2 Design Theme 1: Working with Real-World Data

In our previous analysis using Session Viewer described in the use-case scenario in Section 7.5, we looked at a large-scale user study conducted by a third-party company with over six thousand sessions. Even though the number of sessions was large for detailed analysis, the data was clean, structured, and rich: data were cleaned prior to the tool development, the sessions were grouped into populations based on experimental factors, and we were provided with task descriptions and user feedback for each session. Our analysis therefore took advantage of the structure and looked at populations grouped by experimental conditions, arranged side-by-side in Session Viewer.

By contrast, our participants' data were raw data from a variety of sources. Entire data sets were on the order of one hundred thousand sessions for our participants' ongoing experimental data. Also, our participants may only have had vague ideas about the data composition. From the visualization designer's point of view, to deal with noisy and large data sets, a visual analytic tool needs to:

1. Convey the gist of the data, as data validation is an integral part of the analytical process;

2. Gauge configurability of data fields on users' technical skills, as technical users would more likely augment existing data schema even in specialized systems with established data schemas;

3. Provide fluid data-view projections, as data signals such as patterns or outliers are difficult to find using a single data view.

8.2.1 Finding 1: data validation was integral in analysis

Due to data noise and size, our participants needed to select and process raw data into formats supported by Session Viewer prior to the study sessions. Data selection was arguably the first step of the analysis process for our participants.

All seven participants therefore started each study session with data validation based on their expectations, and usually detected unexpected data-parsing problems within a minute. For example, in the first analysis session, P1 almost immediately realized that there were unexpected duplicates in his data when he saw the Sessions Panel in the Multiple Pane. As seen in Figure 8.2, the colour-coded event sequences created distinctive visual patterns that made duplicates easy to detect.

In addition to the Sessions Panel, three participants used the Histogram Panel to obtain the gist of the data distribution. For example, in the second analysis session, P1 realized that the sessions in the data were long sessions since the histogram showed that most of them had at least 64 events and had long session lengths. He later confirmed that the selection did indeed exclude short sessions.

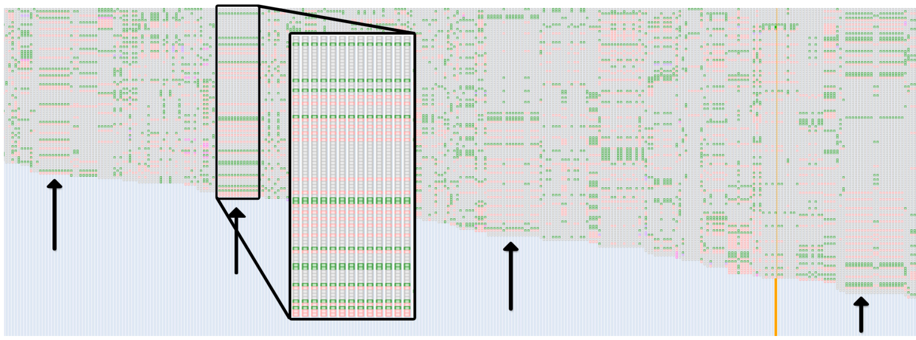


Figure 8.2: One of P1's screen showing duplicate sessions in his data. The arrows mark some areas with duplicates, and the call-out shows a detailed view of one of these areas, where the repeated sessions are highly salient.

In two other cases, participants needed more information to validate their data. For example, P6 labeled search events by experimental treatments. The first step of his analysis was to verify the relative distributions of the different experimental groups. He defined a set of event states for the groups and used the Pattern Matcher to highlight the corresponding sessions. To his surprise, one of the experimental groups, group 3, was over-represented. He looked at the Events Panel for group-3 search events, and wondered if his algorithm defaulted all search events to group 3, or if the library function he used to test for the experimental condition was at fault. Returning to the Pattern Matcher to highlight sessions with other group labels narrowed down his diagnosis, as he found search events that were labeled group 2 and invalidated the first hypothesis. After the study session, he checked the library function used in his parsing script and found that the function had been updated since his last use. Instead of checking if a session was actually included in the experimental group, the function checked if a session were eligible for the experimental condition, thus explaining the over-representation.

Guideline 1: convey the gist of the data

We found that the simple *separate* technique worked surprisingly well in data validation. Both the salient colour pattern and the highlighted view in the Sessions-Panel overview immediately revealed data-processing problems, such as P1's duplicates and P3's mislabeled events. The linked Events Panel provided event information that further helped with problem diagnosis.

Design choice: scrollable overview over automatic session preprocessing

One challenge we faced in conveying the gist of the data was in overview creation, both at the sessions level and at the aggregate level. This section discusses our design choice at the sessions level.

Providing effective overviews for session logs at the sessions level is a challenge due to the large number of sessions and the need to provide enough task-relevant information for session selection and comparison, as suggested by findings in our summary synthesis (Section 4.4). The overview also needs to provide enough visual details for its graphical objects to be useful, even though users can obtain more details in the high-VIR display, as seen in our overview-use study (Chapter 6). As discussed in Section 7.6.3, our Sessions-Panel overview requires scrolling, which in some sense has failed to provide an overview to the entire data.

Interestingly, we did not observe impediments brought about by our incomplete overviews in the field evaluation. We believe that was partly due to the way our participants used Session Viewer, and partly due to the provided session reordering based on session attributes.

Most of our study participants were statistical-aggregate analysts that typically handled data sizes much larger than the amounts supported by Session Viewer. Since Session Viewer could not load the entire data set, our participants therefore used Session Viewer as a tool to generate hypotheses or to verify experimental settings instead of to draw conclusions on usage behaviours. In short, they did not wish to perform exhaustive analysis on the data loaded on Session Viewer, but rather aimed to find interesting ideas to be tested in the entire data. As a result, obtaining a complete overview of the data in a single glance was not essential, as long as they could find interesting sessions to generate analysis leads.

Session Viewer provides session reordering to guide selection by session attributes. Once the sessions were reordered, our participants tended to sample sessions from the extreme ends of the distribution for sessions pertinent to the analysis at hand. One common operation we saw in our study was to reorder sessions by event-state count and to sample sessions at the high end of the distribution. For example, while studying advanced feature use in searches, P1 reordered sessions based on the number of advanced features used per session and looked at sessions with large numbers of advanced feature use.

Despite our initial success with our scrollable overview, the question of overview creation for large data sets remains a challenging research area, which will be discussed in more detail in Section 9.1.1.

Design choice: user data partitioning over automatic preprocessing

In our design, as discussed in Section 7.6.3, we decided against automatic preprocessing to either select or cluster sessions to increase overview display capacity, since we found *a priori* preprocessing to be a double-edged sword in our high-level synthesis (Section 4.4.4). Instead, we decided to provide data partitioning based on filtering for users to select pertinent sessions and to create subpopulations.

To our surprise, we did not observe explicit subpopulation creation using Session Viewer in our study, even though filtering was used to better focus analysis. In some cases, our participants had specific questions in mind while using Session Viewer, and had pre-grouped the sessions based on their questions. A common example was pre-grouping sessions by experimental conditions when analyzing evaluation data. In other cases, we suspected our participants were still inexperienced in using Session Viewer and preferred highlighting within single populations using the pattern-matching feature, presented in Section 7.4.5, instead of explicitly creating subpopulations. We could not tell if the lack of population partitioning using Session Viewer was due to the lack of user need, insufficient support of partitioning by the tool, or lack of tool experience in our participants.

Design choice: aggregate overviews—histogram over stripe graphs; pattern transitions over event-state transitions

Another major challenge in creating Session Viewer was to provide data overviews at the aggregate level. As discussed in Section 7.6.4, we struggled to find effective aggregate visualizations. In our design evolution, Session Viewer first displayed session attribute distributions using stripe graphs in SV2, which was replaced by histograms in SV3, since our target users were confused by the stripe graphs. To show usage patterns in terms of event transitions, SV2 showed event-state transitions, which was replaced by pattern transitions, as noise in data obscured potential signals in the event-state transition diagram.

We observed in our field evaluation that our participants did use the SV3 Histogram Panel to get a sense of the data based on session attributes. However, only two of our seven participants performed session filtering using the Histogram bars.

On the other hand, none of our participants used the Transitions Panel. We reasoned that since the Transitions Panel is probably the most abstract and advanced feature in the tool, and since our participants were still learning the more basic features in Session Viewer such as the Multiple and Detail Panes, the lack of use of the Transitions Panel is understandable, as the panel was mentioned during training but not emphasized. Nonetheless, such lack of use may reflect on the relatively low utility of the panel. It therefore remains to be seen if the pattern-transitions diagram, modified from the event-transitions diagram in SV2, is useful in analysis.

In retrospect, our difficulty in creating useful aggregate visualizations in our design of SV2 and SV3 is not surprising, as our target data is complex and noisy. Indeed, six of our seven participants are statistical-aggregate analysts who are trained to discover signals in their data. Signals that can be picked up by simple aggregate visualizations, even with the power of linked multi-level data views and fluid data projections, will most likely be found by established statistical and data-mining methods such as clustering and principal components analysis.

Design choice: *separate* over *embedded*

One of the main choices in creating Session Viewer was to determine the best way to show the different levels in web log data. As discussed in Section 7.6.5, we adopted a conventional *separate* overview + detail design instead of the *embedded* focus + context alternative, and created a multiple coordinated view interface.

While it is not possible to evaluate our choice of *separate* over *embedded* to display session populations and event details without studying an *embedded* Sessions Panel counterpart, we found in our study that the simple *separate* technique worked surprisingly well in data validation. The colour-coded events create salient patterns, especially when the sessions are long and the data contains duplicates, as in the case for P1 (Figure 8.2). Participants also found the the Sessions-Panel overview highlighted with one event state useful for session selections, as they could fluidly change the event state selected to look for interesting sessions (Figure 8.3). We suspected that such visual pattern searching would be harder if the overview were distorted. In fact, our participants preferred the uniform-sized event boxes in the unit-time view to the duration-coded

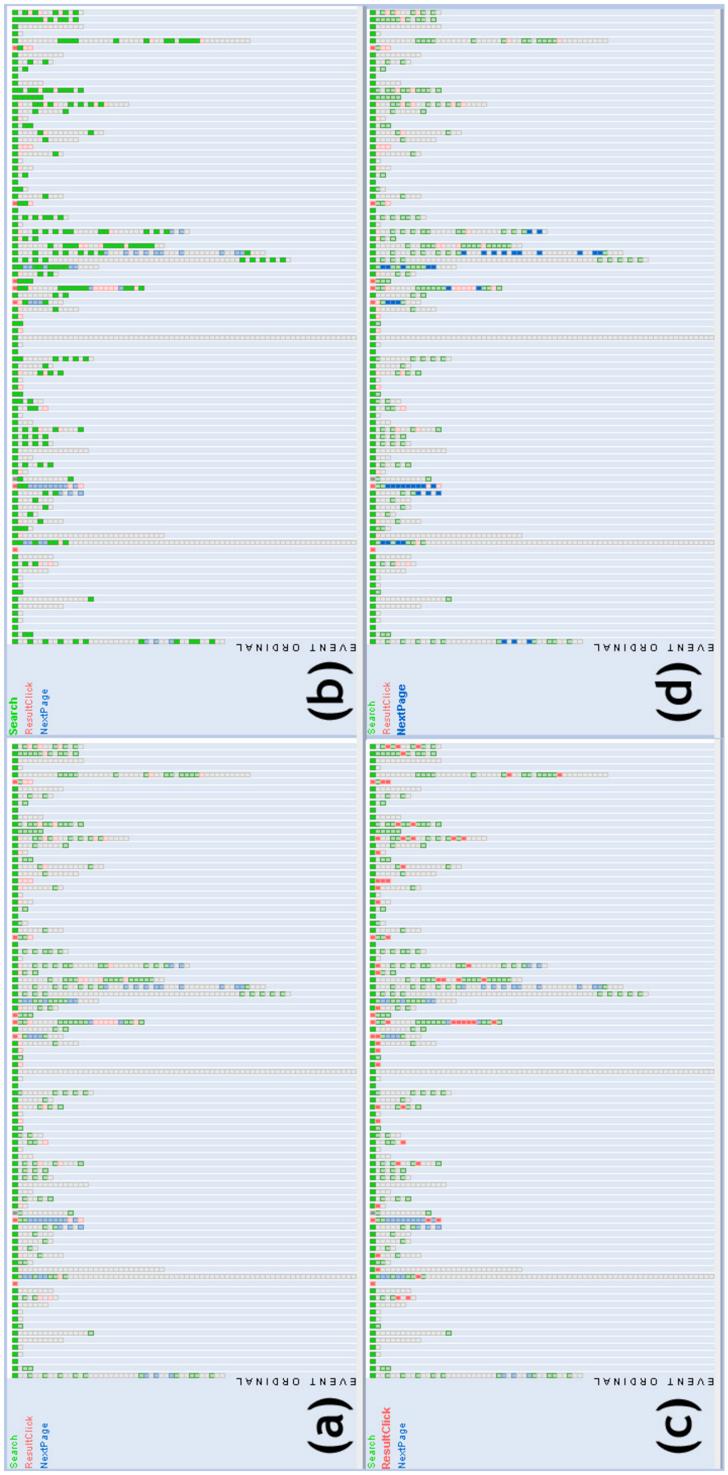


Figure 8.3: Session Viewer allows visual highlighting of one type of event state to facilitate session search. (a) No highlighting; (b) Search events highlighted; (c) ResultClick events highlighted; (d) NextPage events highlighted.

boxes in the length-time view as the uniform boxes better show event transitions (Figure 7.8).

On the other hand, we did observe difficulty in view coordination. For example, our detailed-session participant P1, who frequently viewed sessions on the Sessions Panel in the expanded form (Figure 7.15), often needed to use his finger to help him relate events in the Events Panel with those in the Sessions Panel, even though the events were visually linked by highlighting where the selected (and highlighted) event in one panel were also highlighted in the other. Gauging the tradeoffs between *embedded* and *separate* techniques is therefore difficult, and the topic is further discussed in Section 9.1.3.

8.2.2 Finding 2: data-field needs were diverse

Session Viewer was designed for web session logs with established data fields such as time stamp, duration, event action, and property. To simplify software configuration, we fixed the event attribute fields. However, our technical users added additional fields to the standard set to customize for their analysis tasks, which was unexpected. For the study, we coped with this demand by adding two unspecified string fields and three integer fields. Nonetheless, we were surprised how aggressively two of our participants used these fields. For example, P4 coded experimental conditions applicable for each event as integer values. Given the limited number of integer fields in our tool, he concatenated these values into a string and used the string field instead. During the study sessions, P4 needed to refer back to his coding scheme to understand the overloaded string field.

P2 also overloaded the unspecified fields. She encoded different parameters in the integer field based on the event-action type. For example, if the event was a web-result click, the integer field encoded the result's display position on the original search engine result page. For an advertisement click, the field encoded the number of advertisements shown on the search result page. While such overloading included more information for analysis, it also made event coloring based on the integer field impossible, since event-state definitions in Session Viewer did not allow conditional statements.

We recognized early in the tool design process that session attribute, event state, and pattern definitions have to be user-defined. For ease of use, Session Viewer solicits user input with simple form-filling dialog boxes such as the Pattern Matcher (Figure 7.12). While we did consider a more open-ended input approach in our first prototype designs, we decided against the idea since users would need to assume the burden to ensure syntactically and semantically correct data definitions, which would greatly increase efforts required in configuration. However, our participants and users outside of the study invariably found the definitions restrictive and requested a more open-ended script-like interface.

Guideline 2: gauge configurability of data fields on users' technical skills

Even though we understood that analytical tools should provide extensible data definitions to accommodate diverse data-configuration needs, we learned that the required flexibility would be more accurately gauged by considering target users' technical skills than by data schema, even in cases where the majority of the data fields were well established. In our case, our engineering-oriented participants were accustomed to defining their own data schema using script-like interfaces, and therefore tended to customize their data configurations based on the analysis at hand. Indeed, Bellamy et al. (2007) reported a similar finding in their pilot deployment of a visualization to monitor compliance processes.

Deployment issue 2: provide simple setup or flexible data fields

The tradeoff in flexibility is the time cost in the initial setup, where users need to compile configuration files for most of the data definitions in the tool. Our study found that the decision to sacrifice data-field and attribute configurability for ease of tool setup was incorrect. In retrospect, given the prevalence of script use in analysis at Google Inc., a more open-ended interface would fit better with their current practices. However, we did observe setup costs: the first few minutes in the study sessions were usually spent on defining event states and session attributes. Also, with an open-ended input interface, users would assume the burden to ensure data definition correctness. We further discuss this issue in Design Theme 2.

8.2.3 Finding 3: data signals were difficult to find

We found that our participants' data were too noisy for analysts to find usage patterns in casual explorations, especially when their analysis goals were too general. Indeed, we found in our study that our participants were most successful when they had specific analysis questions in mind. P2's goals were most specific: she wanted to see if the filtering criteria she used to calculate session statistics were correct. Since the raw data were very noisy, P2 needed to first filter out meaningless sessions before calculating session and event metrics, for example, session event count. In her analysis with us, she tested her filtering criteria using selected sessions from different filter settings and examined them in the Events Panel. Within a few minutes, she realized that one of her criteria would incorrectly filter out meaningful short navigational sessions.

All other participants' analysis goals were more open ended. For example, P1 wanted to understand behaviours of searchers who used advanced search features. For such open-ended analysis goals, analysts need to constantly refine the analysis question, ideally into a form suitable for statistical analysis. In the case of P1, he refined his question to look at sessions with the "site:" tag after seeing a large number of search queries in his sample session population that used the "site:" tag to limit the searches within specific domains. After quickly studying such sessions, he was surprised by the searchers' effectiveness and formed a hypothesis: searchers who used "site:" in their queries were experienced users.

In contrast, six of the eight analysis sessions that started with vague intentions were less successful, especially when analysts did not consciously and continuously try to refine the original questions. For example, P6 looked at a few sessions in the Events Panel without forming any hypotheses or questions. While debriefing at end of the study session, P6 realized that he should have prepared for the study session with more specific analysis questions. Similarly, P4 started his study session with the vague intention to "see how users behaved". After a few session-reordering operations, P4 ran out of ideas and turned to his analysis collaborator P5, who provided specific questions that propelled the analysis.

Even though successful analysis sessions tended to start with specific questions, we did observe unexpected discoveries. While pursuing a question from P5 by examining sessions in the Detail pane, P4 found an inconsistency in his event labeling. The discovery was unexpected. After the study session, P4 and his team further investigated the problem, fixed their experiment-labeling algorithms, and reran the experiments. In short, we found that driving questions were needed to propel the analysis and to maintain focus and interest, but having specific goals did not preclude incidental discoveries.

Guideline 3: support fluid data view and hierarchy traversals

We saw that our participants were very fluid in their hypothesis generations and testing as they constantly refine hypotheses based on data under study. Our tool needs to support this quality of data explorations. For participants with open-ended analysis goals, the need for fluid data-view projection was obvious as participants needed to constantly refine the analysis question, as illustrated by one of P1's analyses where he constantly switched between the Multiple and the Detail Panes. As seen in our study, analyses with specific goals could still be very open ended. For example, since P2 did not know the nature of the filtered items before the analysis, she carried out explorations that involved switching between all three data levels to better characterize incorrect filtering.

Design choice: spatial consistency over user control

One of our design goals was to bridge between the statistical-aggregate and the detailed-session analysis approaches (Section 7.3). To facilitate cross-level analysis, we made the choice to adopt a relatively rigid spatial layout: while users can selectively place session populations in the vertical views, the same data panels are displayed for all populations in a fixed order. Our motivation was to avoid the need for users to mentally organize the displayed data levels, as discussed in Section 7.6.6. However, doing so may hinder tool use.

In our study, we found that the rigid layout did not seem to hinder tool use, as our participants resized the panels to pick-and-choose the part of the tool with which they felt comfortable and were relevant for the analysis at hand as unused panels were resized to show only the panel titles. We observed that our participants were very comfortable in using only a few panels. While all of our participants used the more basic Multiple and Detail Panes, only selected participants used the other panels: only two of our seven participants used the Histogram Panel, and none used the Transitions Panel in the Aggregate pane.

Design choice: vertical views over small multiples

To facilitate comparisons between session populations, we decided to stack all panels vertically and to show each population as a view (Section 7.6.6). In doing so, we sacrificed some of the visual linking between semantically related panels in SV1, such as the link between the Sessions and the Events Panels.

In our study, we did observe usability problems related to these concerns.

One of these concerns was reported earlier in discussion of the view-coordination problem in taking the *separate* design (Section 8.2.1), where one of our detailed-session participants, P1, had trouble visually linking between expanded sessions in the Sessions Panel and event details in the Events Panel despite linked visual highlighting. Another problem observed was one of space use. Since most of our participants' monitor setups were in the landscape mode with wider horizontal widths than vertical heights, a vertically stacked view may not be the most efficient for single-population display since both the stacked colour boxes in the Sessions Panel and the table in the Events Panel required vertical space. Indeed, P1 jokingly called his monitor setup "vertically challenged".

These observations made us wonder if we sacrificed the SV1 layout for single-population analysis for multiple-population comparisons. While two of our seven participants compared between session populations, the rest used Session Viewer to display a single population.

8.2.4 Design Theme 1 summary: tool needs to be flexible for real-world data

Even though this seems to be an obvious conclusion, we only realized the degree of flexibility required of our tool during the study. For example, we were surprised by our users' requests to have a completely open-ended configuration interface for data-field definitions, even though Session Viewer has a relatively narrow target data and user set, and such flexibility will incur non-trivial initial parameter-configuration costs. In short, instead of gauging the need for data-field flexibility based on data, our observations suggested estimates should instead be based on the technical skill of users.

While we anticipated the need for fluid data-view projections to support the exploratory nature of hypothesis generation and data analysis, we were surprised by the difficulty in finding signals in our participants' data even with a visualization tool that, by and large, seemed to satisfy most of our participants' required operations. Our observations suggested the strong need for tool flexibility in data-view projection to support the fast mind-frame change in analysis, as we believe a visual analytical tool should not increase the already large cognitive load in analysis. For example, the reorderable sessions-level overview, coupled with the *separate* arrangement of panels, effectively supported the data-validation step, which turned out to be an integral part of our participants' analysis processes.

For the same reason, we arrived at an opposite conclusion for panel layout: we believe a rigid layout would be less cognitively demanding. We therefore avoided the need to mentally organize the large number of panels in Session Viewer by adopting a fixed and integrated layout: a three-level panel layout for each population, and a vertical-view layout for each population. In doing so, we may have compromised SV1's visual linking between semantically related panels.

8.3 Design Theme 2: Tool reception

Even though our tool was not polished and did not fully satisfy our participants' analysis needs, we have received numerous affirmations of its usefulness. For example, P4 volunteered to enhance Session Viewer to link to the actual webpage viewed by his searchers, instead of a recreated webpage based on the logged URL. P3 initiated a Windows laptop purchase to run Session Viewer, incorrectly assuming that the tool was Windows-specific since we used the Windows environment for our study. While we understand new visualization tools sometimes give rise to short-term excitements, a phenomenon reported by González and Kobsa (2003b), we are very positive about Session Viewer's future since we believe it uniquely fills a need in the analysis process.

From our observations, we identified three general factors that affected a tool's initial reception:

1. Tool should fill a unique role in the analysis process, even when it supports a broad set of ill-defined tasks;

2. The need to integrate with existing tools depends on the analysis roles played by the tool. In our case, it is not as crucial as we assumed at design;

3. Tool power and flexibility lead to complexity, which we find to be a difficult tradeoff.

8.3.1 Finding 1: gap in existing analysis-tool coverage

In our pre-study interviews, we noticed that there was a need for statistical-aggregate analysts to take a more detailed look at sessions, either to fine-tune experimental settings, or to obtain context to better interpret their statistical results. Unfortunately, most tools do not link the statistical-aggregate and the detailed-analysis levels. This discontent with insufficient tool coverage is best expressed by P4, who commented that it was worth the 10 hours of participation time to learn about Session Viewer even though he may not gain interesting data insights, as he feels that he needs to understand how searchers respond to experimental treatments at a deeper level, or to understand "what are we really doing to the searchers".

P3 also expressed similar concerns. In a previous role where he conducted animal-behavioural experiments, P3 worked with event sequences that provided information not available from aggregate statistics. He therefore felt that his web statistical analyses were incomplete as he was "unsure about what our searchers are really doing".

There is also a similar lack of tool support in detailed analysis, where the missing piece is an overview. Session selection from text-based logs is difficult, since the nature of sessions is difficult to discern. For example, P1 frequently used the "Find" function to locate interesting sessions in spreadsheet applications he used to view raw session logs.

Guideline 1: fill unique role in analysis

Session Viewer bridges between the statistical-aggregate and detailed-session analysis approaches by showing multiple data levels simultaneously. For statistical-aggregate analysts, being able to see session details is beneficial. For example, P4 needed to automatically divide the logs into individual tasks in his experiments. Originally, he believed a 30-minute period of inactivity would roughly capture task transitions. Viewing the sessions in detail, P4 realized that the 30-minute criterion missed too many task transitions and needed fine-tuning.

Sometimes, even though the analyst is aware of the finding, seeing an actual example deepens the understanding. For example, P3 found an extremely long session where the searcher struggled in a task for over an hour. Even though P3 was aware of the problem that led to the difficult search, seeing the searcher's detailed event log infused an empathy in P3 that "promoted the problem to bug level".

For our detailed-session analyst, the problem was effective session selection. P1 repeatedly expressed that it was much easier to isolate interesting sessions with the coordinated Multiple-Pane overview and the Detail-Pane session-event view using Session Viewer than with text-based software applications such as

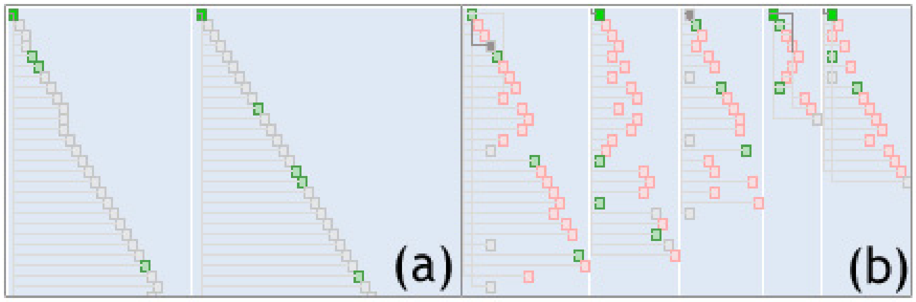


Figure 8.4: Two screen captures of P1's data, showing how the expanded 2-dimensional form of the sessions helped P1 quickly determine usage patterns in his data. (a) The diagonal line indicates that the searcher never returned to an earlier URL, which is in contrast to (b), where another searcher frequently returned to the web result page and examined search results more thoroughly.

spreadsheets. In his analysis of queries with advanced-search features, P1 defined a series of event states that corresponds to commonly used advanced search features, such as the double quotes and the "site:" tag. With the highlighted Sessions-Panel overviews (such as Figure 8.3), he could visually and quickly locate sessions that had multiple advanced search queries to examine in the Events Panel, and could switch between different search features fluidly, since he could display most of his 1000 study sessions over two wide-screen monitors (Figure 8.1(a)). He also found that he could quickly determine usage-behaviour patterns in his searchers with the expanded 2-dimensional session view in the Sessions Panel. Two examples are shown in Figure 8.4, where one of his searchers always returned to an earlier URL, while another almost never did.

Deployment issue 1: Session Viewer plays a unique and crucial role in log analysis

One of our design goals is to bridge between the statistical-aggregate and the detailed-session analysis approaches (Section 7.3). Our study verified Session Viewer's effectiveness in this regard, as Session Viewer had demonstrated its usefulness for both detailed-session and statistical-aggregate analysts in this study.

We believe Session Viewer's unique role in analyses made our participants more forgiving of its kinks and inflexibility. Indeed, our participants were willing to either work around tool deficiencies in their analyses, or in the case of P4, had plans to extend the tool. In contrast, González and Kobsa's (2003a) workplace study on InfoZoom demonstrated the opposite scenario. The researchers explained the lack of tool use was in part due to "the fact that our subjects already have robust software tools at hand to perform data analysis" (p. 96). In other words, despite its demonstrated usefulness in data cleaning, InfoZoom did not fulfill an unique and crucial role in its target users' data analysis task, where by and large, its target users' data analyses were well established with automated procedures. In our case, our target users were dissatisfied with their existing tools and were actively seeking better tools to support their analyses.

8.3.2 Finding 2: data transfer is less crucial than assumed

In a field evaluation to evaluate InfoZoom, González and Kobsa (2003a) reported that InfoZoom's failure in long-term adoption was in part due to integration problems with users' existing sets of analysis tools. Even though InfoZoom offered benefits in data cleaning, their participants did not adapt the tool into their daily routine as "the time saved on [data] cleaning was lost again when they had to transfer the data to another system for further analysis" (p. 96). Saraiya et al. (2006) also reported similar problems.

We found in our study that data transfer between Session Viewer and our analysts' existing tools was a less important issue than we originally anticipated. During the initial design stage, we realized that Session Viewer could not cover all aspects of web session log analysis and decided to provide standard formats to ease data transfer (Section 7.3). However, none of our participants used data exporting. Instead of data transfer, we observed numerous idea transfers. Typically, our participants took simple point-form notes for their analysis discoveries, similar to insights recorded in Saraiya et al.'s (2004) studies (Saraiya et al. 2004; Saraiya et al. 2006). After the study sessions, our participants further investigated these ideas with their usual tools and the entire data set. For example, when P1 continued his analysis on advanced search behaviour from a previous study session, we discussed the use of the "site:" tag in advanced searches and he commented:

> The thing I have noticed about "site:" is if somebody uses "site:" in there, the rest of the query is correct. So it is only the very advanced users. The prediction here is if you use "site:" and get it right, you get everything right too. I did *not* know that before we started this [Session Viewer study], so that's kinda cool.

Our experimenter then asked him where he gained this knowledge:

> A couple of days ago when we were looking through this [advanced search behaviour sessions], I just noticed that there were a lot of them [sessions with the advanced search feature "site:"] in there, and thought, jeez that's odd. So what I did was over the weekend, I actually ran analysis. Session Viewer inspired me to write a piece of code and I found that [confirmed the prevalence of "site:" use in advanced search], and then I started extracting that and thought, look at that, look at all those [sessions], and they [the searches] are all correct. And then I wrote another piece of code which validated that [hypothesis]. So it [Session Viewer] worked perfectly in letting me look through the data very quickly, extracting a feature, and then I wrote a subsequent piece of code that validated that hypothesis.

Even when the exploration did not result in hypotheses about the data, our participants may still have taken away insights in usage behaviours. For example, P3 hoped to extract behaviorial patterns from circuitous searches, which generally indicates difficulties. Even though the analysis sessions seemed to be fruitless explorations as P3 failed to locate good sample sessions with complicated search paths, P3 nonetheless found the sessions useful as he realized that his original mental model of usage behaviour was incorrect:

> It is interesting that [...] the extent to which these sessions in which people tumble for a long time before they go into local [map property] are actually not that much. If they want to go in, they just go in. [...] You get a few of these cases where people sort of realize that they are getting closer, but that is [...] not true. People don't [...] push hard necessarily.

The experience prompted him to consult with his colleagues who analyze search refinements, so that he would be able to better select problem sessions:

> For me, I think the next thing to do is try to learn more about what people [in the company] are thinking about revisions [and] know about statefulness. [...] So I am going to make a point to talk to somebody who has worked with that stuff [...] and probably will end up pulling a new one [set of sessions for the next analysis].

Guideline 2: tool's integration needs depend on role

We believe that the need for data transfer from Session Viewer to our users' typical analysis tools to be low, since our participants used the tool to understand data rather than to form conclusions. Our observation may be unique for tools that support large-data exploration instead of problem detection, where capturing the exact data for further analysis may be crucial.

Deployment issue 2: roles of Session Viewer in analysis

On the surface, our finding seems surprising as it contradicts InfoZoom González and Kobsa's (2003a) finding. Closer inspection reveals differences in our studies. We believe our participants' tasks differed from those reported in González and Kobsa's (2003a) study. In our case, the main goal of our participants' analysis with Session Viewer was data exploration rather than confirmation. In other words, our participants' goals were not to draw conclusions regarding the truth or falsehood of hypotheses, but rather to investigate the factors at play, to generate rough ideas about the data, and to provide preliminary evidence to support their hunches. This process is in accordance with that of exploratory data analysis (EDA) first advocated by Tukey (1977), and is discussed as one of our design considerations in Section 2.1.

A common practice in EDA is to avoid testing rough hypotheses with the same data that inspire them to avoid inflation of Type I error and overfitting (Behrens 1997). Indeed, when possible, exploratory data analysts conduct EDA on one data set to generate hypothesis and access the model on another in a process called cross-validation (Behrens 1997). Given that the volume of our participants' data were too large to be displayed in whole, it is not surprising that our participants only took the idea generated in the study sessions with them, instead of needing to transfer the data they had already seen.

In contrast, the main goals of González and Kobsa's participants seemed, on the surface, to be confirmatory rather exploratory. Most of their data analyses were performed using tools with routines and templates (González and Kobsa 2003b; González and Kobsa 2003a). Their participants only needed the visualization tool, InfoZoom, when their routine tools broke down or were insufficient, or for data cleaning and partitioning. It is the latter scenarios when data transfer became important, as the cleaned or isolated data had to be fed back into their participants' routine analysis tools.

In short, we believe the difference in analysis goals and the availability of data determine if data transfer would be an important consideration in tool reception.

8.3.3 Finding 3: tool power brought complexity

Even though Session Viewer is only one of the tools used in analysis, we intentionally built a powerful and integrated tool as we did not wish to impede analysis by having a limited set of operations. Our assumption was verified during the 12 months of iterative design and implementation with selected target users, as we received requests to provide more flexibility to our existing features. For example, we originally only allowed event states to be defined based on event-action type such as Search, ResultClick, and NextPage events. Based on user feedback, we now allow any event attribute to be used, including event action, property, duration, URL, and title.

The tradeoff is tool complexity. Even though our original state definitions were limiting, they ensured a mutually exclusive set of states that simplified event-colour coding. Otherwise, a single event may belong to multiple states. A common example from our study was defining states by action types (search click, web-result click) and event duration (long, short). Our participants were inevitably confused when an event that satisfied a certain defined state was not colored accordingly.

Having a powerful and flexible set of operations also makes it difficult for users to develop an accurate mental model of the tool. In fact, we often had to pause and think to translate our participants' analysis requests into Session Viewer operations during the study. Not surprisingly, it often took our study participants a while to figure out if Session Viewer could support their session-filtering needs. In fact, P3 called such translations "little puzzles to solve".

We illustrate the difficulty in translating analysis needs to operations with two examples. The first case shows a perfect fit between the analyst's need and Session Viewer's features. In analyzing session logs with advanced searches, P1 wanted to colour-code events with advanced features. Since the URLs of the search requests encoded the advanced feature used and Session Viewer provided string matching for the URL data field, he could define event states based on URL substrings.

In cases where the fit is not perfect, the translation can be tricky. For example, P4 wanted to reorder sessions based on the number of searches before an click event. Since session attribute counts looked at entire sessions, it was not possible to translate P4's request into Session Viewer operations. However, a workaround was to reorder the sessions first by search event counts followed by click counts, which at least clustered the sessions with large numbers of search and click events.

Tool richness may also be distracting. We found that our detailed-session analysts were better at extracting patterns from the large amount of information presented in the Detail Pane than our statistical-aggregate analysts, probably because they were more experienced in dealing with session details and variability.

Guideline 3: consider tool power and complexity as a tradeoff

We believe tool power and complexity is a difficult tradeoff, as on one hand, tool power attracts users to investigate tool functionality and in our case, become involved in tool design and development. On the other hand, we did observe user apprehension, especially during their first introductions to Session Viewer when they faced a large and complicated interface.

In retrospect, given the translation difficulty and tool complexity, we could have built a set of tools, each with simple and well-defined operations, instead of monolithic power tool. For example, P2 often uses simple UNIX commands in her analysis. By piping multiple commands together, for example `grep|uniq|wc` to get counts for unique words in a document, she can achieve most of her analysis goals without complex analysis software. Also, in the pre-study interview, P3 commented that instead of an integrated tool, having a set of individual tools, each simple and easy to use, would also be a good solution to his analysis problems, as he could display them all on his screens and select the appropriate tool to address different analysis needs.

Nonetheless, tool complexity may not be as deterring as it first seems, as we observed that our participants were very comfortable in using only a few tool features. While all of our participants used the more basic Multiple and Detail Panes, only selected participants used the other panels: only P1 and P2 used the Histogram Panel, only P3 used the session-alignment feature (Figure 7.8(c)), and none used the Transitions Panel in the Aggregate Pane. The Transitions Panel, which shows transitions between patterns, is probably the most abstract and advanced in the tool, which may explain the lack of use when our participants were still learning the more basic features. Since users can resize the panels, to a point where only a single title bar is visible, they can pick-and-choose the part of the tool with which they feel comfortable and is relevant for the analysis at hand. For example, P1 frequently resized the screen to show only the Detail Pane when he studied the session events, or to show only the Multiple

Pane when he looked for trends and correlations between session attributes.

8.3.4 Design Theme 2 summary: unique tool role determines reception

Even though it is premature to predict our tool's future, we believe that since Session Viewer fulfills a unique need in the analysis process, our target users will use the tool. Given that our target population includes engineering-oriented users, we also believe that some users may even tailor the tool to better suit their analysis needs. In fact, one group at Google Inc. is adapting Session Viewer to study usage behaviours in a three-dimensional modeling tool. While tool complexity will deter some users, we assumed providing a powerful and flexible tool would be more beneficial as users could pick and choose features based on their analysis needs, but the tradeoff remains contentious. Contrary to our previous design assumption, none of our participants requested data transfers. Instead, our tool was used mostly to generate analysis ideas.

8.4 Limitations of Study

It is still too early to predict the long-term future of Session Viewer as a visual analytic tool for web session log analysis. Due to time constraints and participant availability, we could only schedule two or three analysis sessions with most of our participants for this study over a period of two months (Table 8.1). Compounded with the fact that our participants' were still learning to use Session Viewer during the study, our observations and findings are thus biased towards the early stages in data exploration and analysis.

Our study included two participants who were also involved in the tool design discussions. We included their study sessions in our field evaluation as part of our participatory design process (Section 7.2). While we recognize a potential for biased results as our design collaborators may favour the tool, our observations were geared towards tool use and usability problem identification rather than subjective feedback, and in this discussion, our findings were used to identify broader design implications rather than to validate Session Viewer. In fact, we believe our design collaborators' deeper understanding of the tool allowed them to focus more on their analyses, which may arguably produce more realistic tool-use behaviours. Nonetheless, we recognize the need for a greater pool of participants with a larger number of study sessions to validate Session Viewer.

Also, except for one of our participants (P1), we conducted all analysis sessions at a design office instead of at participants' work environments, even though the analysis data and tasks were performed in the context of participants' own work. Our choice was a workaround for constraints imposed by our field environment. However, we recognize that our study setting, such as computer display configurations, have likely influenced tool use.

Our observations and findings are necessarily limited to our study environment, our participants, and our chosen qualitative analysis method, even though we attempted to tease out more general design guidelines by identifying contributing factors that were particular to our test setting in this chapter. For that reason, we believe while this study adds to existing knowledge in field evaluations of information visualization systems, our conclusions should be considered as deign or deployment considerations instead of claims.

8.5 Summary of Results and Implications for Design

We conducted a field evaluation to observe web session log analysts interact with Session Viewer. Over a period of two months, we observed 20 hours of tool use with seven analyst participants whose analytical approaches range from detailed session to statistical aggregate. In this study, we examined design choices made during the creation of Session Viewer, along with discoveries of deployment considerations of our prototype in the workplace.

In this chapter, we reported two design themes extracted from our observations: (1) design implications in working with real-world data, and (2) factors that lead to initial tool reception. Based on our study findings, we recommend that visualization tools designed to support large and complex data exploration should (1) convey the gist of data to accommodate data validation, (2) provide flexible data-field configurations based on users technical skills, and (3) provide fluid data-view switches to support frequent mind-frame changes in exploratory analyses.

As an example of visual analytic tools, we believe our tool's eventual adoption hinges on whether it provides a unique function in the analysis process, in our case, bridging between the detailed-session and statistical-aggregate analysis approaches. In terms of deployment considerations, we found that our tool needs to provide a more flexible interface for data schema configuration to accommodate the diverse needs of our analysts, especially when they are accustomed to open-ended interfaces. We found that the ease of data transfer between our tool and our analysts tool sets was less of a concern than in other visualization systems (e.g., González and Kobsa 2003a), much to our surprise. We believe that since the main use of the tool is to generate ideas for and validate findings of statistical analysis, idea transfer is probably the norm. On the other hand, we fear that tool complexity may deter some of our target users.

In terms of design choices, we validated the effectiveness of using a scrollable sessions-level overview for getting the gist of the data, especially for data validation. The question of overview creation is far from resolved and will be further discussed in the next chapter in Section 9.1.1. We also found our choice of using *separate* over *embedded* techniques was effective for session selections. However, we did observe problems in view coordination during the field evaluation. How to best arrange multiple visual information resolutions is thus far from understood, and the issue will be discussed in Section 9.1.3.

Our efforts in creating aggregate visualization were less successful: while two of our participants did use the Histogram Panel for filtering, the more abstract Transitions Panel was largely ignored. In contrast, our decision to provide a rigid layout did not seem to hinder tool use, even though we did observe screen space use problems for one of our detailed-session analysts when performing single-population analyses, suggesting a compromise between providing effective layouts for multiple- verses single-population analyses.

Chapter 9

Open Questions, Conclusions, and Future Work

This chapter concludes the thesis by first looking at four open research questions in Section 9.1, and discussions on thesis contributions and future work in Section 9.2.

9.1 Open Questions

This thesis touched on a vast number of topics. This section further discusses four interesting research questions examined in this thesis: creation of overviews for large data sets, roles of context in visualization systems, design choice of spatial arrangements in presenting multiple visual information resolutions, and challenges in evaluating visualization systems.

9.1.1 Creation of overviews

One of the few pieces of methodological guidance for designers of information visualization systems is Shneiderman's (1996) visual information-seeking mantra of "overview first, zoom and filter, then details on demand" (p. 337). While Shneiderman described the mantra as being descriptive and explanatory rather than prescriptive, it has been widely used to guide or justify design decisions (Craft and Carins 2005).

Assuming that Shneiderman's (1996) top-down approach is appropriate for exploratory data analysis, creating the overview remains a difficult problem, especially given the increasingly large data set sizes that far exceed display capacities. Even if all data points can fit onto the display screen, or screens as in the case of multiple-monitor settings, it may be beyond our ability to process the displayed data, especially when the overview does not provide an emergent structure.

In cases where the data itself has structures, the overview and other visual information resolutions (VIR) in the display can model inherent data structures. Our high-level summary synthesis on multiple-VIR design in Chapter 4 shows evidence for that belief, which was also advocated by Furnas (2006). For most hierarchical tree structures, there are fewer items in the higher levels, and the designer will therefore have a much better chance in fitting all of the higher-level data onto the overview. In short, visual designers can take advantage of available domain knowledge crystallized as taxonomies, data categories, and data structures.

For unstructured data, either because of lack of knowledge or lack of true data structures, the approach to create overview is less clear. The overview-use study in this thesis (Chapter 6) investigated this situation. Without the structure, the only other means to display all of the data points is to reduce the amount of details shown per datum. In the extreme, each dimension of each data point is reduced to a single pixel, an approach taken by the pixel-based visualizations such as VisDB (Keim and Kriegel 1994). While such approaches provide very high-density displays, it is unclear if they are effective overviews. In other words, designers may not be able to guarantee that users can find and select regions of interest on the overview for further examination in the high-VIR display.

Obviously, if the answer to the task can be obtained in the low-VIR display alone, and the visual feature carrying that information is salient, or at least visible, the low-VIR overview will be useful. The real question is, what if only part of the answer to the task is available on the overview, due to the inevitable detail reduction required to create the overview? The overview-use study detailed in Chapter 6 investigated whether a happy medium exists, where designers can maximize the amount of data shown on the overview, but also provide details on demand.

Unfortunately, our overview-use study failed to identify such a scenario that is universally true for all of our participants. While we did observe the "overview, zoom, details on demand" use in our multiple-VIR trials, at least 20% of our participants took the seemingly laborious route and used the high-VIR display alone in our visual search and visual comparison tasks. We therefore concluded that users need a lot of visual information on the overview to make the decision to demand details. For example, composite visual features, such as our tri-band target shown in Figure 6.5, were found to be too difficult to parse, as 56% of our participants switched over to the high-VIR display for the task (Table 6.5). Given that the increase in data size outraces the increase in display capacity, not to mention our own visual capacity, simple reduction of visual details to fit all data points onto the overview screen cannot be a viable long-term approach.

To address this challenge, Keim et al. (2006) modified Shneiderman's (1996) mantra to "analysis first, show the important, zoom, filter and analyze further, detail on demand" (p. 16). In other words, the data is pre-processed using automatic analysis methods to create an overview. The user can then work with a data subset of reasonable size, and gradually build up insights in the course of the analytical process. Keim et al.'s (2006) mantra is closer in spirit to Hartwig and Dearing's (1979) exploratory data analysis process, where both recognized the difficulty, if not impossibility, to analyze the entire data at the same time. Instead, analysts should study manageable subsets of the data and build their analysis gradually.

Indeed, pre-processing has been used to address the large-data challenge. For example, van Wijk and van Selow (1999) clustered a year's worth of time-series data of energy consumption by ECN employees and only displayed the seven most significant clusters in the visualization. Their visualization allows further analysis via user interactions, such as selecting an individual day from the display calendar to view the corresponding detailed energy consumption graph, looking for similar consumption patterns in other days, and refining the number of clusters to obtain meaningful consumption patterns over the year.

While van Wijk and van Selow (1999) demonstrated a successful use of clustering on one-dimensional time-series data in overview creation, such an approach can be difficult to apply in multi-dimensional data. For example, in the case of web session logs, clustering can be potentially performed on time duration, event-sequence transition, any event-type counts, or a combination of features.

A similar approach is to develop functions to extract the most salient patterns and relationships from a data set and display them as a coherent abstraction of the exploration and analysis process. Analytic tools should therefore filter out local details and random noise. Monmonier's (1992) concept of summary graphics and DiBiase's (1990) model maps emphasize giving up detail in exchange for a useful abstraction as a key to effective synthesis. This philosophy in visualization creation is also advocated by Tufte in his books (e.g., Tufte 1983).

While it makes sense for the analytic tool to filter out local details and noise or pre-grouped data into meaningful units before display, designers are faced with the issue of user trust. This is particularly true when the relevance metrics employed for filtering and clustering are opaque to the user (Woods et al. 2002). Hornbæk and his colleagues have used *a priori* filtering in their *embedded* interfaces evaluated in two studies (Hornbæk et al. 2003; Jakobsen and Hornbæk 2006). In both studies, participants have voiced their distrust and dismay at the hidden intelligence, as discussed our summary synthesis in Section 4.4.4.

Even if the tool's target users have learned through experience to trust its intelligence, the question remains, how does the tool distinguish between interesting patterns and local details? In his comments on current techniques to deal with data overload, Woods et al. (2002) concluded that "systems that reduce or filter available data are brittle in the face of context sensitivity" (p. 25). In other words, domain and task contexts give meaning and relative importance to individual data. In our field evaluation of Session Viewer (Chapter 8), we saw that while some of our participants wanted to see average usage behaviours, some looked for extreme cases to focus on problem searches not well supported by the search engine. In short, outliers that are noise to one analyst may be the focus of the analysis for another.

In addition to context sensitivity, data with non-obvious structures are difficult to summarize in any context. For example, session log analysis is still an active area of research where researchers are trying to construct web metrics (e.g., Dhyani et al. 2002), task taxonomies (e.g., Rose and Levinson 2004), user characteristics such as levels of search expertise, and an understanding of how these factors eventually influence usage patterns (e.g., user experience and information search and re-access (Aula et al. 2005)). In the development of Session Viewer, we struggled to provide high-level aggregate visualizations that meaningfully sum up session population, as discussed in Section 7.6.4. In retrospect, our difficulty probably reflects the lack of obvious ways to view the complex data.

Given these challenges in visual information reduction to fit data onto a overview, and challenges in pre-processing to extract main features in data, we took a different approach to create the Sessions-Panel overview in Session Viewer (Figure 7.2). While we did abstract individual session events to a box on the screen, we kept the size of the box at 5x5 pixels to make them physically selectable without the need of magnifying tools. As a result, the Sessions-Panel overview requires both vertical and horizontal scrolling with our participants' data, as we observed in our field evaluation to study Session Viewer use in the workplace (Chapter 8). In other words, our users can only see a subset of the data even with the "overview". To some information visualization researchers, this choice contradicts the point of having an overview, which is to provide a birdseye view of the *entire* data set.

Surprisingly, none of our participants mentioned this tool "defect" in our field evaluation (Section 8.2.1). I believe this is due to the nature of the task and the provision of session-attribute guided session reordering in Session Viewer.

As discussed in Section 2.1, unlike confirmatory analysis, the goals of exploratory data analysis is to understand data: to look for factors at play, form rough ideas about their relationships, and to collect preliminary evidence to justify the validity of these ideas (Behrens 1997). Perhaps for these goals, seeing the entire data set is less important than seeing the interesting parts of it. Indeed, Hartwig and Dearing's (1979) exploratory data analysis approach is essentially bottom up, not top down. By providing session reordering based on session attributes, users of Session Viewer can isolate the part of the data that is of interest to them, with selection either based on domain knowledge and experience or found by experimentation. Session Viewer's answer to the challenge of context sensitivity of data is to provide fluid shifting of data views based on criteria defined and controlled by users. In fact, we observed that despite loading thousands of sessions per population, our participants only examined less than 50 per population in detail during the study sessions. Once they were satisfied in their rough ideas, they moved on to another session population created to address a different analysis task.

To summarize, creation of overview (in *separate* displays) or context (in *embedded* displays) is central to multiple-VIR interface design and Shneiderman's (1996) visual information-seeking mantra. However, overview creation is increasingly difficult due to large data size and complex but non-obvious data structures. While several approaches exist, for example high-density display, visual information abstraction and *a priori* filtering or clustering, all approaches at best involves substantial tradeoffs, and at worst, are questionable in their effectiveness and long-term viability. We took the approach of coupling data distribution with the overview. Even though our approach helps guide users to the part of the overview that is of interest to them, it is still unclear if, and how, the incomplete overview impacts their analyses. The importance of an overview in data analysis and the lack of satisfactory approach in its creation make this an interesting research question.

9.1.2 The roles of context

While there is study evidence to support the proposed roles of overview in *separate* displays, such as providing a table-of-content like map for navigation and to convey overall structures, the roles of context in *embedded* displays, such as orientation and provision of meaning, is less clear. This issue was discussed in Section 4.4.5 of the chapter that details our summary synthesis of multiple-VIR interface study results to extract design guidelines.

In fact, even the definition of context is not well established (Furnas 2006). Since 1998, the JiangLab at Harvard University has been conducting experiments to study a related phenomenon called contextual cueing, or, the process in which our visual system uses global properties of an image to help the selection, recognition and control of action by prioritizing objects and regions in complex scenes (Chun and Jiang 1998, p. 30). In their experiments, context was defined as:

- *Spatial layout*, or the location of target with respect to those of the distractor objects (Chun and Jiang 1998);
- *Shape covariation*, or the association of target and distractor shapes in the display with varying locations (Chun and Jiang 1999), and
- *Regularities in motion trajectories* in dynamic visual environments (Chun and Jiang 1999).

Study results indicate that context can be useful in two ways:

- *Direct attention*. Olson and Chun (2002) reported that context is important in guiding attention to less familiar objects. If context is interpreted as "features in a scene that do not change", context can direct observers' attention to unfamiliar objects if they have repeated exposure to the same scene.
- *Provide structure and coherence to objects*. Biederman (1972) reported that the meaningfulness of a scene facilitated perceptual processing of the objects at an early stage. Chun and Jiang (1999) stated that the visual

> context of a scene facilitates recognition of objects relevant to that context. Knowledge about our highly-structured visual world may serve to reduce the large amount of uncertainty and complexity in the visual input.

Based on these study results, context seems to be important in our daily lives, as it can guide attentional deployment and facilitates visual behaviours such as visual search and object recognition, which are basic components of more complex tasks like object location and navigation. However, context has to be acquired before use. Studies found that context is implicitly learned, and what we learn from the display depends on previous tasks performed using the display (Jiang and Song 2004). Our visual system is surprisingly robust in transferring learned context (Jiang and Wagner 2004; Jong and Jiang 2005). Even though the learning is task dependent (Jiang and Song 2004), we have high capacities for context learning with long retention time (Jiang et al. 2004).

While study results from cognitive psychology are interesting, the challenge is to apply existing knowledge to visualization systems and user study designs. In short, we as a community need to connect literature from another discipline to ours, even though applying low-level findings to visualization system design and evaluation is a challenge, as discussed in Section 5.6 when we derived design guidelines from our visual-memory experiment. One example of cross connection is in image transformation. Studies from the JiangLab suggest that, even though our visual system is surprisingly robust in tolerating changes in these contextual patterns, caution is advised to visualization designers as not all transformations are equally well tolerated. For example, transformations like scaling, displacement, and regrouping do not affect the transfer of contextual learning, but topologies of the display should be preserved (Jiang and Wagner 2004). Jiang and Wagner's (2004) finding thus corroborate with our low-level visual-memory experiment results detailed in Chapter 5, and their conclusions corroborate with guidelines proposed by Misue et al. (1995) to mitigate the disorienting effect of image transformations.

Knowledge from another discipline can also benefit user study designs in understanding interface use. Even though lower-level understanding of how our visual system uses context in daily living should assist creation of tasks that can benefit from a *embedded* visualization, we continue to have difficulties in finding such tasks, as discussed in Section 6.4, where we discussed limitations of the overview-use study. JiangLab's study results suggest that context needs to be learned implicitly, and context learned in one task type may not be transferable to another. For example, context learned in visual search tasks cannot be transferred to change detection tasks, and learning from change detection tasks only moderately helps visual search tasks (Jiang and Song 2004). Since empirical user studies to evaluate visualization techniques tend to use a different set of data for each trial, participants often face new context at every trial. Without revisiting the same context, potential benefits of contextual cueing may not be realizable as performance benefits in interface use.

In summary, the roles of context in *embedded* displays remains an important and interesting research question. At a deeper level, how to apply and connect to knowledge from another discipline is perhaps a more difficult and broader issue.

9.1.3 Spatial arrangements of the VIRs

One open question identified from our summary synthesis of multiple-VIR interface design guidelines is one of spatial arrangements between the different VIRs when displayed simultaneously (Section 4.6). Since we found that distortion, frequently used in *embedded* systems, is disruptive to visual search and visual memory (Chapter 5, and also Section 4.6.1), and since web log data have too many data levels to be integrated into a single view, we have taken the *separate* approach to design Session Viewer (Chapter 7).

However, even though the *separate* technique seemed to be effective in supporting line-graph comparisons in our overview-use study (Chapter 6), and in supporting data verification and session selection in Session Viewer (Section 8.2.1), taking the *separate* approach is not without costs. We observed the classic view-coordination problem in *separate* interfaces despite visual linking and highlighting in both our overview-use study (Section 6.3.4) and in our field evaluation (Section 8.2.1). In both cases, the problem occurred when the two views were not spatially linked. In the overview-use study, the *Separate* interface showed the low-VIR strip panel on top of the high-VIR plots. In Session Viewer, the sessions in the Sessions Panel were also placed on top of the event-detail table in the Events Panel. Since in both cases the visual objects (line graphs and events) were arranged vertically, a possible improvement in the layout may require putting the low- and the high-VIR views side-by-side such that corresponding elements in both views can be aligned vertically, for example, as in the first Session Viewer design (SV1, Section 7.6.1).

In general, perhaps interactive linking and brushing are not enough to resolve the view-coordination problem; we may also need to spatially associate graphical elements in the separate views. However, this is mere speculation and the issue of spatial arrangements in multiple-VIR design is still an open research question.

9.1.4 Evaluating information visualization

Evaluating information visualization systems and techniques is difficult (Plaisant 2004). To understand interface use, this thesis conducted a summary synthesis (Chapter 4), a laboratory experiment (Chapter 5), a experimental-simulation study (Chapter 6), and a field evaluation (Chapter 8).

In general, experimental strategies (laboratory experiment and experimental simulation) are better suited to study visualization techniques than systems, as these strategies tend to focus on a limited number of pre-defined factors each with a small number of levels. To avoid confounds, experimenters need to isolate interface visual and interactive elements, control data characteristics, and use simple and generic tasks. Abstraction of tasks and data may have the advantage to produce more generalizable results by looking at specific aspects of the tested system with larger numbers of participants, and study results are potentially amenable to high-level summary analyses or even meta-analyses, as discussed in Section 4.8, where we discussed improvements on current study methods to increase the utility of individual experimental-simulation studies.

At one extreme, experimentalists can abstract tasks to simple operations such as visual search and measure human perceptions. In our laboratory experiment in Chapter 5, we looked at two-dimensional geometric transformations and measured its effect on visual memory. However, we had difficulties in applying our results to design, as the interplay between human visual perception and visualization is complex, and our understanding of it is still incomplete for us to isolate and identify factors to build models of interface use (Section 5.6).

In our experimental-simulation study in Chapter 6, we used fully interactive interfaces and scenario-based tasks to gain insights that can be used to improve on interface design before predictive models of interface use can be built in terms of basic psychological measures (Section 6.5). However, to ensure generalizability, we abstracted visual factors (Section 6.1.3), controlled data characteristics (Section 6.1.2), and developed tasks based on taxonomies (Section 6.1.1). For example, we only retained the strip and plot visual elements and corresponding strip-plot switching interactions in our test interface, and removed all reordering and clustering interactions from the original Line Graph Explorer interface (Kincaid and Lam 2006) that implemented these visual encodings. However, such abstractions may render the tested factors unrealistic and overly simplistic. For example, our choice of task was limited by the need for objectively measurable task outcomes, reasonable task performance time, and simplistic tasks that can be performed by non-expert and quickly trained participants (Plaisant 2004). The combination of simplistic tasks and insufficient participant motivation may account for the lack of use of the multiple-VIR interfaces, as we believe our participants may have failed to see the long-term benefits in learning to use the more interactively complex interfaces (Section 6.3.4).

Also, the measurements, or dependent variables, in studies of experimental strategies are typically objective time and accuracy measurements. These measurements, adopted from the field of experimental psychology, are now being questioned as being suitable to measure usability. For example, Hornbæk (2006), after reviewing 180 studies selected from 587, suggests our community should look at both objective and subjective measures, learnability, satisfaction, and how these parameters develop over time. A bi-annual workshop called BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV) aims to find new measurements to quantify interface usability.

Measurements in studies of experimental strategies are generally fixed prior to the experiment, making uncovering unexpected interface factors difficult, if not impossible. The experimental-simulation study detailed in Chapter 6 illustrates this point. Our most interesting conclusions were drawn from observations of interface mode used by participants to locate their answers, instead of from our dependent variables. We could only conjecture on the lack of use in our multiple-VIR interface based on our recorded observations, not from our objective time and accuracy measurements. Even though our statistical findings may change if our challenges in task creation and training were resolved, I believe the types of interface use problems observed during our study, such as view coordination and decision making, will remain valid.

Limitations in study realism and factor discovery may render experimental strategies inappropriate to study visualization systems, since system use is heavily influenced by context of use, such as working environment, user characteristics (domain expertise, engagement/incentive, and individual differences), task, and data. In interface use, there are many potentially important factors involved, and some of them surprising and only identified after extensive piloting, much like Taleb's (2007) Black Swans [3]. For example, Reilly and Inkpen (2007) unexpectedly found that study environments significantly affected visualization effectiveness.

For these reasons, field experiments are more ecologically valid for system evaluations. Longer-term studies provide more complete pictures of system use (Shneiderman and Plaisant 2006), since true usage patterns can only emerge over time, as found by González and Kobsa in their evaluations of InfoZoom (González and Kobsa 2003b; González and Kobsa 2003a). Also, using the user's own data and task, rather than synthetic ones, provides a more realistic study setting and ensures participant motivation and engagement, as found in Saraiya et al.'s (2004) studies on analyzing micro-array data with various visualizations (Saraiya et al. 2004; Saraiya et al. 2006).

The field evaluation conducted to understand how potential users use Session Viewer revealed unexpected results that cannot be found using laboratory experiments or experimental-simulation studies (Figure 1.2). As discussed in Chapter 8, the study uncovered a stronger need of data-field configurability required in the software than expected, and a much weaker need for data transfer.

Perhaps because of the advantages of realistic settings and the ability to find unexpected results, there is a steady increase in field experiments in human-computer interaction research, from 7% in 1983 to 14% in 2006 (Barkhuus and Rode 2007). Indeed, researchers have expressed discontent in experimental strategies, and have suggested looking into more qualitative, exploratory, and long-term evaluations (e.g., Plaisant 2004; Shneiderman and Plaisant 2006). Nonetheless, field experiments are not without problems. For example, it is often difficult to generalize results obtained from field experiments, since they tend to study specific systems in-depth under specific settings with only a handful of participants.

In short, both experimental and field strategies have merits and problems, and it remains an open research question to search for better evaluation methods to study interface use. My own unsubstantiated belief is that since both the human-computer interaction and information visualization communities are still developing effective metrics to quantify usability, our previous focus on experimental strategies (Barkhuus and Rode 2007), especially on laboratory experiments, is perhaps premature, as experimenters need more open ended, observation based, and exploratory methods to discover and identify important

[3] According to Taleb, black swan is a large-impact, hard-to-predict, and rare event beyond the realm of normal expectations

factors in interface use before they can be quantified with quantitative methods.

9.2 Thesis Conclusions and Future Work

The larger goal of this thesis is to see how visualizations can be used to support exploratory data analysis (EDA) of large data sets with complex structure. The thesis begins by evaluating different aspects of multiple visual information resolution (VIR) interface design in laboratory settings, then focuses on the web session log analysis application domain to further investigate issues that arise during the actual design, implementation, and deployment of visualization systems in the workplace. The thesis ends by reflecting on four open research questions in the use of multiple-VIR interfaces to support EDA of large data sets.

For the evaluation aspects of the thesis, we employed a range of methods to study multiple-VIR interface to examine our two research areas for the thesis: (1) overview creation and use; and (2) VIR spatial arrangements.

The **summary synthesis** (Chapter 4) reviewed 19 multiple-VIR studies to provide evidence for design decisions in creating a multiple-VIR interfaces, a knowledge gap in both the information visualization and human-computer interaction communities. In addition to extracting guidelines for multiple-VIR interface design (Section 4.7), experience gained in conducting the summary synthesis was crystallized as methodology guidelines to increase the utility of experimental-simulation studies by making them more amenable to high-level analysis (Section 4.8).

In terms of design recommendations, study results suggested that the number of VIRs in a system and the levels of organization in the supported data should match, and low VIRs should only display task-relevant information as extra information had been found to impede task performance. Simultaneous display of the multiple VIRs was found to be suitable for tasks with multi-level answers or clues to reach the answers. Otherwise, temporal switching of the different VIRs was found to support better performance, possibly due to the simpler and more familiar interface and interaction.

In terms of methodology recommendations, we recommended using comparable study interfaces, capturing usage patterns in addition to overall performance measures, isolating interface-use factors, and reporting more study details to increase consistency among experimental-simulation studies and increase their utility in contributing to meta-analyses.

The **visual-memory laboratory experiment** (Chapter 5) systematically measured the visual memory costs of two-dimensional geometric transformations such as scaling, rotation, rectangular fisheye, and polar fisheye transformations. Based on response time and accuracy results, we defined a no-cost zone for each transformation type within which we did not detect performance degradations. We found that the scaling transformation is well tolerated, as we could not detect performance degrading within our experimental range. Among the two fisheye transformations, the polar fisheye transformation was found to be better tolerated than its rectangular counterparts. We verified two guidelines to mitigate visual memory costs of the two-dimensional geometric transformations. Misue et al.'s (1995) guideline to preserve orthogonal ordering in displays was verified and refined to only providing an up-down direction, and the effectiveness of background grids was verified.

The **overview-use experimental-simulation study** (Chapter 6) examined the assumption that users can select regions of interest to examine at higher VIRs. We examined and refuted this assumption for single-level data and proposed interaction costs as a factor. We found that that our participants would reliably use the low-VIR overviews only when the visual targets were simple and had small visual spans. Otherwise, at least 20% chose to use the high-VIR view exclusively. We therefore concluded that use of multiple VIR for single-level data is likely to be inappropriate, as scant benefits in having multiple VIRs cannot compensate interaction costs, such as the need for making selection decisions using the low-VIR display and view coordination in *separate* interfaces.

Our application contributions in this thesis is **our design study of Session Viewer**. Session Viewer is our test bed to examine guidelines from the first three laboratory evaluations in the thesis, where we tested the guidelines' applicability in a concrete, mature, and fully functional visualization system, tested in an ecologically-valid setting. We therefore chose the application domain of web session log analysis. After surveying the problem space of log data, the exploratory data analysis task, and the specific needs of session analysts at Google Inc., the target environment for Session Viewer, we proposed a visualization solution to bridge between the high-level statistical-aggregate and the low-level detailed-session analysis styles, which was implemented as Session Viewer, presented in Chapter 7.

We proposed and implemented a scrollable sessions-level overview augmented by session attributes to address the overview creation question examined in our first three studies. We took the *separate* approach and display session data at the aggregate, multiple, and detail levels. While the aggregate and detail levels correspond to levels in existing analysis approaches, the middle multiple level helps bridge between the two and facilitates cross-level analyses. To aid comparisons between session populations, we used the multiple coordinated view approach to support side-by-side comparisons at all data levels.

Our **Session Viewer field evaluation**, detailed in Chapter 8, was the fourth and last study in this thesis. Taking a qualitative approach, we obtained findings based on 20 hours of observations with seven session analysts working on their own data and tasks. We examined the impact our design choices (Section 7.6) to test the validity of our design guidelines. We also identified two main deployment issues of visual analytic tools in the workplace.

We summarize our findings as visual design choice and deployment considerations:

1. Visual design choices:

    - Scrollable overviews may be a viable solution to the challenge in overview creation for large data sets with complex compositions when users can discern data distribution to selectively view interesting portion of the data, and when the goal of the analysis is to derive, rather than confirm, hypotheses (Section 8.2.1);
    - Our difficulties in creating effective aggregate visualizations to characterize session populations reflect on the complex composition and lack of obvious structure in our data, which support the need for exploratory, rather than confirmatory, analysis (Section 7.6.4).
    - The *separate* technique is indeed effective in supporting data verification and session selection, even though our choice was based more on the potential problems in using *embedded* techniques than on the merits of using the *separate* approach (Section 8.2.1). Despite the observed effectiveness of the *separate* approach in our study, it is still unclear if *separate* techniques would be superior in displaying multiple data levels than *embedded* techniques, as we did not directly study and compare the two approaches;
    - While showing data panels in a fixed layout for all session populations can facilitate comparisons between populations and avoid user confusion in visualization systems with multiple panels, rigid layout

may result in suboptimal display for single-population analyses (Section 8.2.3).

2. Deployment considerations:

    - The required degree of flexibility in the configuration of data fields and definitions can be more accurately gauged by the degree of technical skills of users, not by the degree of acceptance of existing data schema, as more technical users tend to augment existing schema for specific analysis tasks (Section 8.2.2);
    - The need for data-transfer support is dependent on the analysis task. In our case, our tool supports exploratory data analysis where our study analysts used Session Viewer to generate ideas about the data instead of to draw conclusions. As a result, idea transfer, instead of data transfer, was the norm found in the study (Section 8.3.2).

In terms of future directions, it would be interesting to look at how interface interactions impact multiple-VIR use. We identified interaction costs as an important consideration in both our summary analysis (Section 4.3.1) and in our overview-use study (Section 6.3.4). However, interaction has not been a focus in the information visualization community. While Yi et al.'s (2007) user-intent based classification of interactions in visualizations is a useful initial step, we also need a common language to discuss interaction costs, and metrics to quantify such costs and their impact on interface use.

Even though the thesis is coming to an end, Session Viewer is an ongoing project at Google Inc. There are both short and long term developments for Session Viewer, and plans to follow up with our study participants.

The short-term items are engineering work to make Session Viewer more accessible to our users. We plan to address the usability and design issues revealed in our field evaluation. We plan to provide more flexible definitions of session and event attributes, for example, to replace form-like dialog boxes (e.g., for pattern sequences in Figure 7.12) with a script-like interface given our technical target users. To address the issue with our limited data-field configurability, we plan to adapt an open-ended configuration file instead of improving upon our current hybrid of fixed data and open fields. These modifications will pave the way to expanding Session Viewer to other computer-generated logs, for example usability logs.

There are also three major areas for longer term developments. The first two may improve the scalability of Session Viewer. Even though it would be very difficult to produce a visual analytics tool that can handle the amount of data typically involved in statistical web session log analysis, Session Viewer can take advantage of such analysis by displaying how the current displayed populations compare to the larger data populations, much like how Session Viewer places individual sessions in the context of its session population at the Multiple data level. A more challenging, but potentially fruitful, approach is to aggregate the sessions prior to display and informed users of clustering operations performed so that they can modify the operations if needed. Tesone and Goodall (2007) aggregated data prior to display to maximize user's situation awareness in massive data.

The second area for future development is to investigate how Session Viewer may benefit from machine intelligence. Predefined aggregate statistical visualizations, examples being the ones we implemented in the Aggregate Pane in the second and third versions of Session Viewer, may be too simplistic to address analysts' needs. Automatic clustering in statistical analysis is still an open area of research and user trust may be an issue, as discussed in Section 4.4.4. However, I believe that visualizations may be able to help users customize clustering algorithms (e.g., in Nam et al. 2007), which may improve both clustering algorithms and visualizations.

The last area of future development is to increase the flexibility of display panels. While I believe the fixed three-level data of Aggregate, Multiple and Detail helps orient users and reduces cognitive load, panels within each data level should be configurable. Ideally, Session Viewer should provide a list of visualizations for each data level, and users can pick appropriate visualizations for each task.

In terms of evaluation, even though our study spanned a period of two months, it is still too early to conclude if our tool is successful. Since Session Viewer is available inside Google Inc., we plan to follow up with our participants in six months' time to monitor long-term tool use.

Bibliography

Ahlberg, C. and B. Shneiderman (1994). Visual Information Seeking: Tight Coupling for Dynamic Query Filters with Starfield Displays. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'94)*, pp. 313–317.

Amar, R., J. Eagan, and J. Stasko (2005). Low-Level Components of Analytic Activity in Information Visualization. In *Proc. IEEE Symposium on Information Visualization (InfoVis'05)*, pp. 111–117.

Aula, A., N. Jhaveri, and M. Kaki (2005). Information Search and Re-access Strategies of Experienced Web Users. In *Proc. 14th Intl. World Wide Web Conf. (WWW 2005)*, pp. 583–592.

Ayres, I. (2007). *Super Crunchers: Why Thinking-by-Numbers is the New Way to be Smart*. Bantam.

Baldonado, M. W., A. Woodruff, and A. Kuchinsky (2000). Guidelines for Using Multiple Views in Information Visualization. In *Proc. ACM Advanced Visual Interfaces (AVI'00)*, pp. 110–119.

Barkhuus, L. and J. Rode (2007). From Mice to Men: 24 years of Evaluation in CHI. In *Alt.Chi*.

Bartram, L., A. Ho, J. Dill, and F. Henigman (1995). The Continuous Zoom: a Constrained Fisheye Technique for Viewing and Navigating Large Information Space. In *Proc. ACM SIGCHI Symposium on User Interface Software and Technology (UIST'95)*, pp. 207–215.

Baudisch, P., N. Good, V. Bellotti, and P. Schraedley (2002). Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'02)*, pp. 259–266.

Baudisch, P., B. Lee, and L. Hanna (2004). Fishnet, a Fisheye Web Browser with Search Term Popouts: A Comparative Evaluation with Overview and Linear View. In *Proc. ACM Advanced Visual Interfaces (AVI'04)*, pp. 133–140.

Baudisch, P. and R. Rosenholtz (2003). Halo: a Technique for Visualizing Off-Screen Locations. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*, pp. 481–488.

Bederson, B. (2000). Fisheye Menus. In *Proc. ACM SIGCHI Symposium on User Interface Software and Technology (UIST'00)*, pp. 217–226.

Bederson, B. and A. Boltman (1999). Does Animation Help Users Build Mental Maps of Spatial Information? In *Proc. IEEE Symposium on Information Visualization (InfoVis'99)*, pp. 28–35.

Bederson, B., A. Clamage, M. Czerwinski, and G. Robertson (2004, March). DateLens: A Fisheye Calendar Interface for PDAs. *ACM Trans. on Computer-Human Interaction (TOCHI) 11*(1), 90–119.

Bederson, B. and J. Hollan (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proc. of ACM SIGCHI Symposium on User Interface Software and Technology (UIST'94)*, pp. 17–26.

Bederson, B., B. Shneiderman, and M. Wattenberg (2002). Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. *ACM Transactions on Graphics (TOG) 21*(4), 833–854.

Behrens, J. (1997). Principles and Procedures of Exploratory Data Analysis. *Psychological Methods 2*(2), 131–160.

Bellamy, R., T. Erickson, B. Fuller, W. Kellogg, R. Rosenbaum, J. Thomas, and T. V. Wolf (2007). Seeing is Believing: Designing Visualization for Managing Risk and Compliance. *IBM Systems J. 46*(2), 205–218.

Bertin, J. (1981). *Graphics and Graphic Information-Processing*. Walter de Gruyter.

Biederman, I. (1972). Perceiving Real-World Scenes. *Science 177*(4043), 77–80.

Biehl, J., M. Czerwinski, G. Smith, and G. Robertson (2007). FASTDash: A Visual Dashboard for Fostering Awareness. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'07)*, pp. 1313–1322.

Boyatzis, R. (1998). *Transforming Qualitative Information: Thematic Analysis and Code Development*. Sage Publications.

Buring, T., J. Gerken, and H. Reiterer (2006). Usability of Overview-Supported Zooming on Small Screens with Regard to Individual Differences in Spatial Ability. In *Proc. ACM AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, pp. 223–240.

Card, S., J. Mackinlay, and B. Shneiderman (1999). *Readings in Information Visualization: Using Vision to Think*. San Francisco, California: Morgan Kaufmann.

Card, S., T. Moran, and A. Newell (1983). *The Psychology of Human Computer Interaction*. CRC.

Card, S., P. Pirolli, M. van der Wege, J. Morrison, R. Reeder, P. Schraedley, and J. Boshart (2001). Information Scent as a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'91)*, pp. 498–505.

Carpendale, M., D. Cowperthwaite, and F. Fracchia (1997). Making Distortions Comprehensible. In *Proc. IEEE Symposium on Visual Languages*, pp. 36–45.

Chamberlain, T. (1965). The Method of Multiple Working Hypothesis. *Science 148*, 754–759.

Chen, C. and M. Czerwinski (2000). Empirical Evaluation of Information Visualizations: An Introduction. *Intl. J. of Human-Computer Studies (IJHCS) 53*, 631–635.

Chen, C. and Y. Yu (2000). Empirical Studies of Information Visualization: A Meta-Analysis. *Intl. J. of Human-Computer Studies (IJHCS) 53*, 851–866.

Chernoff, H. (1973). Using Faces to Represent Points in k-dimensional Space. *J. of the American Statistical Association 68*, 361–368.

Chi, E. (2002). Improving Web Usability Through Visualization. In *Internet Computing*, pp. 259–263.

Chi, E., P. Pirolli, K. Chen, and J. Pitkow (2001). Using Information Scent to Model User Information Needs and Actions and the Web. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'01)*, pp. 490–497.

Christou, C., B. Tjan, and H. Bulthoff (2003). Extrinsic Cues Aid Shape Recognition from Novel Viewpoints. *J. of Vision 3*(1), 183–198.

Chun, M. and Y. Jiang (1998). Contextual Cueing: Implicit Learning and Memory of Visual Context Guides Spatial Attention. *Cognitive Psychology 36*, 28–71.

Chun, M. and Y. Jiang (1999). Top-Down Attentional Guidance Based on Implicit Learning of Visual Covariation. *Psychological Science 10*(4), 28–71.

Churcher, N., P. Prachuabmoh, and C. Churcher (1997). Visualization Techniques for Collaborative GIS Browsers. In *Proc. International Conf. on GeoComputation*.

Cooper, L. and R. Shepard (1973). Chronometric Studies of the Rotation of Mental Images. In W. Chase (Ed.), *Visual Information Processing*. New York, NY: Academic Press.

Corbin, J. M. and A. C. Strauss (2007). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3rd Edition ed.). Sage Publications.

Craft, B. and P. Carins (2005). Beyond Guidelines: What Can We Learn from the Visual Information Seeking Mantra? In *Proc. IEEE Information Visualization (IV'05)*, pp. 110–118.

Cugini, J. and J. Scholtz (1999). VISVIP: 3D Visualization of Paths through Web Sites. In *Proc. 10th Intl. Workshop on Database & Expert Systems Applications: Web-Based Information Visualization (WebVis)*, pp. 259–263.

Czerwinski, M., M. van Dantzich, G. Robertson, and H. Hoffman (1999). The Contribution of Thumbnail Image, Mouse-over Text and Spatial Location Memory to Web Page Retrieval in 3D. In *Human-Computer Interaction: INTERACT '99*, pp. 163–170.

Dhyani, D., W. Ng, and S. Bhowmick (2002). A Survey of Web Metrics. *ACM Computing Surveys (CSUR) 34*(4), 469–503.

DiBiase, D. (1990). Visualization in the Earth Sciences. *Earth and Mineral Sciences, Bulletin of the College of Earth and Mineral Sciences, Penn State University 59*(2), 13–18.

Eick, S., J. L. Steffen, and E. S. Jr. (1992). Seesoft—a Tool for Visualizing Line Oriented Software Statistics. *IEEE Transactions in Software Engineering 18*(11), 943–956.

Ellis, G. and A. Dix (2006). An Exploratory Analysis of User Evaluation Studies in Information Visualization. In *Proc. ACM AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, pp. 1–7.

Ellis, G. and A. Dix (2007). A Taxonomy of Clutter Reduction for Information Visualization. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 13*(6), 1216–1223.

Frecon, E. and G. Smith (1998). WebPATH: A Three Dimensional Web History. In *Proc. IEEE Symposium on Information Visualization (InfoVis'98)*, pp. 3–10.

Furnas, G. (1986). Generalized Fisheye Views. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'86)*, pp. 16–23.

Furnas, G. W. (2006). A Fisheye Follow-Up: Further Reflection on Focus + Context. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*, pp. 999–1008.

Goldstein, A. and J. Chance (1971). Visual Recognition Memory for Complex Configurations. *Perception and Psychophysics 9*(2), 237–241.

González, V. and A. Kobsa (2003a). A Workplace Study of the Adoption of Information Visualization Systems. In *Proc. International Conf. on Knowledge Management (I-KNOW'03))*, pp. 92–102.

González, V. and A. Kobsa (2003b). Benefits of Information Visualization Systems for Administrative Data Analysts. In *Proc. IEEE International Conf. on Information Visualization (IV'03)*, pp. 331–336.

Good, I. (1983). The Philosophy of Exploratory Data Analysis. *Philosophy of Science 50*, 283–295.

Gray, M., A. Badre, and M. Guzdial (1996). Visualizing Usability Log Data. In *Proc. IEEE Symposium on Information Visualization (InfoVis'96)*, pp. 93 – 98.

Gutwin, C. and C. Fedak (2004). A Comparison of Fisheye Lenses for Interactive Layout Tasks. In *Proc. Conf. on Graphical Interface (GI'04)*, pp. 213–220.

Gutwin, C. and A. Skopik (2003). Fisheye Views are Good for Large Steering Tasks. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*, pp. 201–208.

Guzdial, M., P. Santos, A. Badre, S. Hudson, and M. Gray (1994). Analyzing and Visualizing Log Files: A Computational Science of Usability. Technical Report GIT-GVU-94-08, Georgia Institute of Technology.

Guzdial, M., C. Walton, M. Konemann, and E. Soloway (1993). Characterizing Process Change Using Log File Data. Technical Report GIT-GVU-93-44, Georgia Institute of Technology.

Hartly, T., I. Trinkler, and N. Burgess (2004). Geometric Determinants of Human Spatial Memory. *Cognition 94*(1), 39–75.

Hartwig, F. and B. Dearing (1979). *Exploratory Data Analysis.* Sage Publications.

Havre, S., E. Hetzler, P. Whitney, and L. Nowell (2002). ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 8*(1), 9–20.

Healey, C., K. Booth, and J. Enns (1996). High-Speed Visual Estimation using Preattentive Processing. *ACM Trans. on Computer-Human Interaction (TOCHI) 3*(2), 107–135.

Healey, C. and J. Enns (1999). Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 5*(2), 145–167.

Herman, I., G. Melancon, and M. Marshall (2000). Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 6*(1), 24–43.

Hetzler, E. and A. Turner (2004). Analysis Experiences Using Information Visualization. *IEEE Computer Graphics and Application 24*(5), 22–26.

Ho, Y. (1994, April). Abduction? Deduction? Induction? Is There A Logic of Exploratory Data Analysis. In *Paper presented at the American Educational Research Association*.

Hochheiser, H. and B. Shneiderman (2001). Using Interactive Visualizations of WWW Log Data to Characterize Access Patterns and Inform Site Design. *J. of American Society of Information Sciences 52*(4), 331–343.

Hochheiser, H. and B. Shneiderman (2003). A Dynamic Query Interface for Finding Patterns in Time Series Data. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*, pp. 522–523.

Hornbæk, K. (2006). Current Practice in Measuring Usability: Challenges to Usability Studies and Research. *Intl. J. of Human-Computer Studies (IJHCS) 64*, 79–102.

Hornbæk, K., B. Bederson, and C. Plaisant (2002). Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. *ACM Trans. on Computer-Human Interaction (TOCHI) 9*(4), 362–389.

Hornbæk, K. and E. Frokjær (2001). Reading of Electronic Documents: The Usability of Linear, Fisheye and Overview+Detail Interfaces. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'01)*, pp. 293–300.

Hornbæk, K., E. Frokjaer, and C. Plaisant (2003). Reading Patterns and Usability in Visualization of Electronic Documents. *ACM Trans. on Computer-Human Interaction (TOCHI) 10*(2), 119–149.

Hornbæk, K. and M. Hertzum (2007). Untangling the Usability of Fisheye Menus. *ACM Trans. on Computer-Human Interaction (TOCHI) 14*(2).

Hornof, A. (2001). Visual Search and Mouse-Pointing in Labeled Versus Unlabeled Two-Dimensional Visual Hierarchies. *ACM Trans. on Computer-Human Interaction (TOCHI) 8*(3), 171–197.

Huber, D. and C. G. Healey (2005). Visualizing Data with Motion. In *Proc. IEEE Visualization 2005*, pp. 527–534.

Hubona, G., P. Wheeler, G. Shirah, and M. Brandt (1999). The Relative Contributions of Stereo, Lighting, and Background Scenes in Promoting 3D Depth Visualization. *ACM Trans. on Computer-Human Interaction (TOCHI) 6*(3), 214–242.

Hudhausen, C., S. Douglas, and J. Stasko (2002). A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing 13*, 259–290.

Inselberg, A. (1985). The Plane with Parallel Coordinates. *The Visual Computer (Special Issue on Computer Geometry) 1*(2), 69–92.

Jakobsen, M. R. and K. Hornbæk (2006). Evaluating A Fisheye View of Source Code. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*, pp. 377–386.

Jansen, B. (2006). Search Log Analysis: What Is It; What's Been Done; How to Do It. *Library and Information Science Research 28*(3), 407–432.

Jiang, Y. and J. Song (2004). Spatial Context Learning in Visual Search and Change Detection. *Perception & Psychophysics 67*(7), 1128–1139.

Jiang, Y., J. Song, and A. Rigas (2004). High-Capacity Spatial Contextual Memory. *Psychonomic Bulletin & Review 12*(3), 524–529.

Jiang, Y. and L. Wagner (2004). What Is Learned in Spatial Contextual Cueing: Configuration Or Individual Locations? *Perception & Psychophysics 66*(3), 454–463.

Johnson, B. and B. Shneiderman (1991). Tree-maps: A Space-filling Approach to the Visualization of Hierarchical Information Structures. In *Proc. IEEE Visualization 1991*, pp. 275–282.

Jones, W., H. Bruce, and S. Dumais (2001). Keeping Found Things Found On the Web. In *Proc. ACM Conf. on Information and Knowledge Management*, pp. 119–126.

Jong, J. and Y. Jiang (2005). Connecting the Past with the Present: How Do Humans Match An Incoming Visual Display with Visual Memory? *J. of Vision 5*, 322–330.

Jul, S. and G. Furnas (1998). Critical Zones in Desert Fog: Aids to Multiscale Navigation. In *Proc. ACM SIGCHI Symposium on User Interface Software and Technology (UIST'98)*, pp. 97–106.

Kay, J. and R. Thomas (1995). Studying Long-Term System Use. *Communications of the ACM 38*(7), 61–69.

Keim, D. (2002). Information Visualization and Visual Data Mining. *IEEE Trans. Visualization and Computer Graphics (TVCG) 7*(1), 100–107.

Keim, D. and H.-P. Kriegel (1994). VisDB: Database Exploration Using Multidimensional Visualization. *IEEE Computer Graphics and Applications 14*(5), 40–49.

Keim, D., F. Mansmann, J. Schneidewind, and H. Ziegler (2006). Challenges in Visual Data Analysis. In *Proc. IEEE Information Visualization (IV'06)*, pp. 9–16.

Kellar, M., C. Watters, and M. Shepherd (2007). A Field Study Characterizing Web-Based Information-Seeking Tasks. *J. of the American Society for Information Science and Technology 58*(7), 999–1018.

Kincaid, R. and H. Lam (2006). Line Graph Explorer: Scalable Display of Line Graphs Using Focus+Context. In *Proc. ACM Advanced Visual Interfaces (AVI'06)*, pp. 404–411.

Koike, H. and K. Ohno (2004). SnortView: Visualization System of Snort Logs. In *Proc. ACM Workshop on Visualization and Data Mining for Computer Security (VizSec '04)*, pp. 143–147.

Kreuseler, J., T. Nocke, and H. Schumann (2004). A History Mechanism for Visual Data Mining. In *Proc. IEEE Symposium on Information Visualization (InfoVis'04)*, pp. 49–56.

Lam, H. and P. Baudisch (2005). Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'05)*, pp. 681–690.

Lam, H., T. Munzner, and R. Kincaid (2007). Overview Use in Multiple Visual Information Resolution Interfaces. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 13*(6), 1278–1283.

Lam, H., R. Rensink, and T. Munzner (2006). Effects of 2D Geometric Transformations on Visual Memory. In *Proc. Symposium on Applied Perception in Graphics and Visualization (APGV'06)*, pp. 119–126.

Lamping, J., R. Rao, and P. Pirolli (1995). A Focus+Context Technique Based On Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'95)*, pp. 401–408.

Landragin, F., N. Bellalem, and L. Romary (2001). Visual Salience and Perceptual Grouping in Multimodal Interactivity. In *Proc. International Workshop on Information Presentation and Natural Mulimodal Dialogue (IPNMD)*, pp. 151–155.

Lau, K., R. Rensink, and T. Munzner (2004). Perceptual Invariance of Nonlinear Focus+Context Transformations. In *Proc. ACM Symposium on Applied Perception in Graphics and Visualization (APGV'06)*, pp. 65–72.

Lee, J., M. Podlaseck, E. Schonberg, and R. Hoch (2001). Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising. *Data Mining and Knowledge Discovery 5*(1/2), 59–84.

Leinhardt, S. and S. Wasserman (1979). Exploratory Data Analysis: An Introduction to Selected Methods. *Sociological Methods 10*, 311–365.

Leung, Y. and M. Apperley (1994, June). A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Trans. on Computer-Human Interaction (TOCHI) 1*(2), 126–160.

Levitt, S. and S. Dubner (2005). *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything*. HarperCollins Publishers Inc.

Littlefield, R. (1983). Using the GLYPH Concept to Create User-Definable Display Formats. In *Proc. NCGA'83*, pp. 697–706.

Lyman, P. and H. Varian (2003). How Much Information? Available at http://www.sims.berkeley.edu/research/projects/how-much-info-2003/.

MacEachren, A. and M. Kraak (1997). Exploratory Cartographic Visualization: Advancing the Agenda. *Computers & Geosciences 23*(4), 335–343.

MacEachren, A., D. Xiping, F. Hardisty, G. Diansheng, and G. Lengerich (2003). Exploring High-D Spaces with Multiform Matrices and Small Multiples. In *Proc. IEEE Symposium on Information Visualization (InfoVis'03)*, pp. 31–38.

McGrath, J. (1994). Methodology Matters: Doing Research in the Behavioral and Social Sciences. In R. Baecker, J.Grudin, and S. G. W. Buxton and (Eds.), *Readings in Human-Computer Interaction: Toward the Year 2000*. Morgan Kaufmann.

Miller, J. (1960). Information Input Overload and Psychopathology. *J. of Psychiatry 116*, 695–704.

Misue, K., P. Eades, W. Lai, and E. Sugiyama (1995). Layout Adjustment and the Mental Map. *J. of Visual Languages and Computing 6*(2), 183–210.

Monmonier, M. (1992). Summary Graphics for Integrated Visualization in Dynamic Cartography. *Cartography and Geographic Information Systems 19*(1), 23–36.

Montgomery, D. (2001). *Design and Analysis of Experiments* (5th Edition ed.). John Wiley & Sons, Inc.

Mountjoy, D. (2001). *Perception-Based Development and Performance Testing of a Non-linear Map Display*. Ph. D. thesis, North Carolina State University, North Carolina, USA.

Munzner, T., F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou (2003). TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. *ACM Transactions on Graphics 22*(3), 453–462.

Nam, E., Y. Han, K. Mueller, A. Zelenyuk, and D. Imre (2007). ClusterSculptor: A Visual Analytics Tool for High-Dimensional Data. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST'07)*, pp. 75–82.

Nekrasovski, D., A. Bodnar, J. McGrenere, T. Munzner, and F. Guimbretière (2006). An Evaluation of Pan and Zoom and Rubber Sheet Navigation. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*, pp. 11–20.

Nigay, L. and F. Vernier (1998). Design Method of Interaction Techniques for Large Information Spaces. In *Proc. of Advanced Visual Interfaces (AVI'98)*, pp. 37–46.

Norman, D. (1993). *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Perseus Books.

North, C. and B. Shneiderman (1997). A Taxonomy of Multiple Window Coordinations. Technical Report CS-TR-3854, Univ. Maryland Computer Science Dept.

North, C. and B. Shneiderman (2000). Snap-Together Visualization: Can Users Construct and Operate Coordinated Visualizations. *Intl. J. of Human-Computer Studies (IJHCS) 53*(5), 715–739.

Olson, C. and E. Chi (2003). ScentTrails: Integrating Browsing and Search on the Web. *ACM Trans. on Computer-Human Interaction (TOCHI) 10*(3), 177–197.

Olson, I. and M. Chun (2002). Perceptual Constraints on Implicit Learning of Spatial Context. *Visual Cognition 9*, 273–302.

Palmer, S. (1999). *Vision Science*. Cambridge MA: Bradford.

Pass, G., A. Chowdhury, and C. Torgeson (2006). A Picture of Search. In *Proc. First Intl. Conf. on Scalable Information (INFOSCALE '06)*, pp. Article No. 1.

Pierrakos, D., G. Paliouras, C. Papatheodorou, and C. Spyropoulos (2003). Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction 13*, 311–372.

Pirolli, P. and S. Card (1999). Information Foraging. *Psychology Review 106*, 643–675.

Pirolli, P. and S. Card (2005). Sensemaking Processes of Intelligence Analysis and Possible Leverage Points as Identified Through Cognitive Task Analysis. In *Proc. Intl. Conf. on Intelligence Analysis*.

Pirolli, P., S. Card, and M. van der Wege (2003). The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browswer. *ACM Trans. on Computer-Human Interaction (TOCHI) 10*(1), 20–53.

Pitkow, J. and K. Bharat (1994). WebViz: A Tool for Worldwide Web Access Log Visualization. In *Proc. First Intl. World Wide Web Conf. (WWW1)*, pp. 271–277. Elsevier.

Plaisant, C. (2004). The Challenge of Information Visualization Evaluation. In *Proc. ACM Advanced Visual Interfaces (AVI'04)*, pp. 109–116.

Plaisant, C., D. Carr, and B. Shneiderman (1995). Image-Browser Taxonomy and Guidelines for Designers. *IEEE Software 12*(2), 21–32.

Plaisant, C., J. Grosjean, and B. Bederson (2002). SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In *Proc. IEEE Symposium on Information Visualization (InfoVis'02)*, pp. 57–64.

Plumlee, M. and C. Ware (2006). Zooming versus Multiple Window Interfaces: Cognitive Costs of Visual Comparisons. *Proc. ACM Transactions on Computer-Human Interaction (TOCHI) 13*(2), 179–209.

Rao, R. and S. Card (1994). Table Lens: Merging Graphical and Symbolic Representations in An Interactive Focus Plus Context Visualization for Tabular Information. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'94)*, pp. 318–322.

Reilly, D. and K. Inkpen (2007). White Rooms and Morphing Don't Mix: Setting and the Evaluation of Visualization Techniques. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'07)*, pp. 111–120.

Reisberg, D. (2001). *Cognition: Exploring the Science of the Mind* (Second ed.). W. W. Norton & Company.

Rensink, R. (2004). The Invariance of Visual Search to Geometric Transformation. *J. of Vision 4*.

Roberts, J. (2000). Multiple-View and Multiform Visualization. In *Visual Data Exploration and Analysis VII. Proc SPIE 3960*, pp. 176–185.

Robertson, G., S. Card, and J. Mackinlay (1989). The Cognitive Coprocessor Architecture for Interactive User Interfaces. In *Proc. ACM SIGCHI Symposium on User Interface Software and Technology (UIST'89)*, pp. 10–18.

Rose, D. and D. Levinson (2004). Understanding User Goals in Web Search. In *Proc. 13th Intl. World Wide Web Conf. (WWW13)*, pp. 13–19. Elsevier.

Rosenholtz, R., Y. Li, J. Mansfield, and Z. Jin (2005). Feature Congestion: A Measure of Display Clutter. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'05)*, pp. 761–770.

Roth, S. and J. Mattis (1990). Data characterization for Intelligent Graphics Presentation. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'90)*, pp. 193–200.

Russell, D. and C. Grimes (2007). Assigned Tasks Are Not the Same as Self-Chosen Web Search Tasks. In *Proc. IEEE 40th Annual Hawaii Intl. Conf. on System Sciences (HICSS'07)*, pp. 83.

Russell, D., M. Stefik, P. Pirolli, and S. Card (1993). The Cost Structure of Sensemaking. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'93)*, pp. 269–276.

Sanderson, P. and C. Fisher (1994). Exploratory Sequential Data Analysis: Foundations. *Human-Computer Interaction 9*, 251–317.

Saraiya, P., P. Lee, and C. North (2005). Visualization of Graphs with Associated Timeseries Data. In *Proc. IEEE Symposium on Information Visualization (InfoVis'05)*, pp. 225–232.

Saraiya, P., C. North, and K. Duca (2004). An Evaluation of Microarray Visualization Tools for Biological Insight. In *Proc. IEEE Symposium on Information Visualization (InfoVis'04)*, pp. 1–8.

Saraiya, P., C. North, V. Lam, and K. Duca (2006). An Insight-Based Longitudinal Study of Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 12*(6), 1511–1522.

Sarkar, M., S. Snibbee, O. Tversky, and S. Reiss (2003). Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In *Proc. ACM SIGCHI Symposium on User Interface Software and Technology (UIST'89)*, pp. 81–91.

Schafer, W. and D. A. Bowman (2003). A Comparison of Traditional and Fisheye Radar View Techniques for Spatial Collaboration. In *Proc. ACM Conf. on Graphical Interface (GI'03)*, pp. 39–46.

Schaffer, D., Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman (1996, March). Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Trans. on Computer-Human Interaction (TOCHI) 3*(2), 162–188.

Schwartz, B. (2004). *The Paradox of Choice: Why More Is Less.* New York, NY: HarperCollins.

Sellen, A., R. Murphy, and K. Shaw (2002). How Knowledge Workers Use the Web. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'02)*, pp. 227–234.

Seo, J. and B. Shneiderman (2006). Knowledge Discovery in High-Dimensional Data: Case Studies and a User Survey for the Rank-by-Feature Framework. *IEEE Trans. on Visualization and Computer Graphics 12*(3), 311–322.

Shadish, W., T. Cook, and D. Campbell (2002). *Experimental and Quasi-Experimental Designs.* Houghton Mifflin Company.

Shenk, D. (1997). *Data Smog: Surviving the Information Glut.* New York, NY: Harper Collins Publishers.

Shepard, R. and J. Metzler (1971). Mental of Three-Dimensional Objects. *Science 171*, 701–703.

Shi, K., P. Irani, and B. Li (2005). An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations. In *Proc. IEEE Symposium on Information Visualization (InfoVis'05)*, pp. 81–88.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy of Information Visualization. In *Proc. IEEE Visual Languages*, pp. 336–343.

Shneiderman, B. and C. Plaisant (2006). Strategies for Evaluating Information Visualization Tools: Multi-Dimensional In-Depth Long-Term Case Studies. In *Proc. ACM AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, pp. 1–7.

Simon, H. and J. Larkin (1987). Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science 11*(1), 65–99.

Skopik, A. and M. Brown (1992). Graphical Fisheye Views of Graphs. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'92)*, pp. 83–91.

Skopik, A. and C. Gutwin (2005). Improving Revisitation in Fisheye Views with Visit Wear. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'05)*, pp. 771–780.

Stolte, C. and P. Hanrahan (2000). Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases. In *Proc. IEEE Symposium on Information Visualization (InfoVis'00)*, pp. 5–14.

Strauss, A. C. and J. M. Corbin (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (2rd Edition ed.). Sage Publications.

Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations.* New York, NY: Doubleday.

Takada, T. and H. Koike (2002). MieLog: A Highly Interaction Visual Log Browser Using Information Visualization and Statistical Analysis. In *Proc. USENIX Conf. on System Administration*, pp. 133–144.

Taleb, N. (2007). *The Black Swan: The Impact of the Highly Improbable.* New York, NY: Random House.

Teevan, J., C. Alvarado, M. Ackerman, and D. Karger (2004). The Perfect Search Engine Is Not Enough: A Study of Orienteering Behavior in Directed Search. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'04)*, pp. 415–422.

Tesone, D. and J. Goodall (2007). Balancing Interaction Data Management of Massive Data with Situational Awareness through Smart Aggregation. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST'07)*, pp. 67–74.

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

Tory, M., A. Kirkpatrick, M. Atkins, and T. Möller (2006). Visualization Task Performance with 2D, 3D, and Combination Displays. *IEEE Trans. Visualization and Computer Graphics (TVCG) 12*(1), 2–13.

Tory, M. and T. Möller (2004). Rethinking Visualization: A High-Level Taxonomy. In *Proc. IEEE Symposium on Information Visualization (InfoVis'04)*, pp. 151–158.

Trafton, J., S. Kirschenbaum, T. Tsui, R. Miyamoto, J. Ballas, and P. Raymond (2000). Turning Pictures Into Numbers: Extracting and Generating Information From Complex Visualizations. *International J. of Human Computer Studies 53*, 827–850.

Treisman, A. (1985). Preattentive Processing in Vision. *Computer Vision, Graphics, and Image Processing 31*(2), 156–177.

Treisman, A. and G. Gelade (1980). A Feature-Integration Theory of Attention. *Cognitive Psychology 12*, 97–136.

Tufte, E. (1983). *The Visual Display of Quantitative Information.* Cheshire, CT: Graphics Press.

Tukey, J. (1977). *Exploratory Data Analysis.* Addison-Wesley.

Tukey, J. (1986). Data Analysis, Computation and Mathematics. In L. V. Jones (Ed.), *The Collected Works of John W. Tukey. Volume IV: Philosophy and Principles of Data Analysis: 1965-1986*, pp. 753–775. (Original work published 1972).

Tullis, T. (1985). A Computer-Based Tool for Evaluating Alphanumeric Displays. In B. Shackel (Ed.), *Human-Computer Interaction: INTERACT '84*, pp. 719–723. Elsevier Science.

Tversky, B., J. Morrison, and M. Betrancourt (2002). Animation: Can It Facilitate? *Intl. J. of Human-Computer Studies (IJHCS) 57*(4), 247–262.

Valiati, E., M. Pimenta, and C. Freitas (2006). A Taxonomy of Tasks for Guiding the Evaluation of Multidimensional Visualizations. In *Proc. ACM AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, pp. 1–6.

van Wijk, J. and E. van Selow (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proc. IEEE Symposium on Information Visualization (InfoVis'99)*, pp. 4–9.

Viégas, F., S. Golder, and J. Donath (2005). Visualizing Email Content: Portraying Relationships From Conversational Histories. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*, pp. 979–988.

Viégas, F. and M. Smith (2004). Newsgroup Crowds and Authorlines: Visualizing the Activity of Individuals in Conversational Cyberspaces. In *Proc. 37th Annual Hawaii Intl. Conf. on System Sciences (HICSS'04)*, pp. 40109.2.

Ward, M. (1994). XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proc. IEEE Visualization (Vis'94)*, pp. 326–333.

Ware, C. (2004). *Information Visualization: Perception for Design* (2nd ed.). Morgan Kaufmann.

Waterson, S., J. Hong, T. Sohn, J. Landay, J. Heer, and T. Matthews (2002). What Did They Do? Understanding Clickstreams with the WebQuilt Visualization Systems. In *Proc. ACM Advanced Visual Interfaces (AVI'02)*.

Wehrend, W. and C. Lewis (1990). A Problem-oriented Classification of Visualization Techniques. In *Proc. IEEE Visualization (Vis'90)*, pp. 139–143.

Willett, W., J. Heer, and M. Agrawala (2007). Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 13*(6), 1129–1136.

Winckler, M., P. Palanque, and C. Freitas (2004). Tasks and Scenario-based Evaluation of Information Visualization Techniques. In *Proc. ACM Conf. on Task Models and Diagrams (TAMODIA'04)*, pp. 165–172.

Wise, J., J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow (1995). Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents. In *Proc. IEEE Symposium on Information Visualization (InfoVis'95)*, pp. 51.

Wixted, J. (1998). Remembering and Forgetting. In K. A. Lattal and M. Perone (Eds.), *Handbook of Research Methods in Human Operant Behavior (Applied Clinical Psychology)*. Springer.

Wobbrock, J., J. Forlizzi, S. Hudson, and B. Myers (2002). WebThumb: Interaction Techniques for Small-Screen Browsers. In *Proc. ACM SIGCHI Symposium on User Interface Software and Technology (UIST'02)*, pp. 205–208.

Wolfe, J. (2000). Visual Attention. In K. D. Valois (Ed.), *Seeing* (2nd Edition ed.). San Diego, CA: Academic Press.

Wong, B. and G. Marden (2001). Effectively Exploiting Server Log Information for Large Scale Web Sites. In *Proc. South African Institute for Computer Scientists and Information Technologists (SAICSIT)*, pp. 223–227.

Woodruff, A., A. Faulring, R. Rosenholtz, J. Morrison, and P. Pirolli (2001). Using Thumbnails to Search the Web. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'01)*, pp. 198–205.

Woods, D., E. Patterson, and E. Roth (2002). Can We Ever Escape from Data Overload? A Cognitive Systems Diagnosis. *Cognition, Technology & Work 4*, 22–36.

Yee, K., D. Fisher, R. Dhamija, and M. Hearst (2002). Animated Exploration of Graphs with Radial Layout. In *Proc. IEEE Symposium on Information Visualization (InfoVis'01)*, pp. 45–50.

Yeh, M. and C. Wickens (2001). Attentional Filtering in the Design of Electronic Map Displays: A Comparison of Color-Coding, Intensity Coding, and Decluttering Techniques. *The J. of the Human Factors and Ergonomics Society 43*(4), 543–562.

Yi, J., Y. Kang, J. Stasko, and J. A. Jacko (2007). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. In *Proc. IEEE Symposium on Information Visualization (InfoVis'07)*, pp. 1224–1231.

Zanella, A., M. Carpendale, and M. Rounding (2002). On the Effects of Viewing Cues in Comprehending Distortions. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'02)*, pp. 119–128.

Zhang, Y. (2005). Subjective Mental Workload Measures. *Ergonomia IJE&HF 27*(3), 199–206.

Appendix A

Previously Published or In-Preparation Papers

The majority of work described in this thesis has either been published in conference proceedings or journals, submitted, or is in preparation for publication. Materials from published work are adapted and expanded in this thesis. Here is a list of publications related to this thesis.

Published journal and conference papers:

1. Lam, H., T. Munzner, and R.A. Rensink (2006). The Invariance of Visual Long-term Memory to Geometric Transformation. *Journal of Vision 6* (6), 983a.

2. Lam, H., T. Munzner, and R.A. Rensink (2006). Effects of 2D Geometric Transformations on Visual Memory. In the *Proceedings of the ACM 3rd Symposium on Applied Perception in Graphics and Visualization (APGV'06)*, pp. 119–126.

3. Lam, H., T. Munzner, and R. Kincaid (2007). Overview Use in Multiple Visual Information Resolution Interfaces. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 13*(6), 1278–1283.

4. Lam, H., D. Russell, D. Tang, and T. Munzner (2007). Session Viewer: a Visual Exploratory Analysis of Web Session Logs. In the *proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST'07)*, pp. 147–154.

5. Lam, H. and T. Munzner (2008). Increasing the Utility of Quantitative Empirical Studies for Meta-analysis. In the *Proceedings of the ACM CHI Workshop on BEyond time and errors: novel evaluation methods for information visualization (BELIV'08)*, pp. 21–27.

Submitted journal paper:

1. Lam, H. and T. Munzner. A Study-Based Guide to Multiple Visual Information Resolution Interface Designs.

In-preparation journal paper:

1. Lam, H., D. Russell, D. Tang, and T. Munzner. Session Viewer: Visual Exploratory Analysis of Web Session Logs.

The following includes a list of copyrighted work that has been reused or further developed in this thesis:

1. Part of Chapter 4 is based on an earlier work "Increasing the Utility of Quantitative Empirical Studies for Meta-analysis" to be published in the Proceedings of the ACM CHI 2008 Workshop of BEyond time and errors: novel evaLuation methods for Information Visualization ©ACM, 2008.

2. Chapter 5 is based on an earlier work "Effects of 2D Geometric Transformations on Visual Memory" published in the Proceedings of the ACM 3rd Symposium on Applied Perception in Graphics and Visualization ©ACM, 2006. `http://doi.acm.org/10.1145/1140491.1140515`.

3. Chapter 6 is based on an earlier work "Overview Use in Multiple Visual Information Resolution Interfaces" published in IEEE Transactions on Visualization and Computer Graphics 13(6) ©IEEE, 2007. `http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.70583`.

4. Chapter 7 is based on an earlier work "Session Viewer: a Visual Exploratory Analysis of Web Session Logs" published in the Proceedings of the IEEE Symposium on Visual Analytics Science and Technology ©IEEE, 2007. `http://ieeexplore.ieee.org/10.1109/VAST.2007.4389008`.

Appendix B

Summary Synthesis Reviewed Studies

This appendix summarizes the key aspects of the multiple-VIR interface studies reviewed in Chapter 4. We first summarize each study in Section B.1. In Section B.2, we list study interfaces, tasks, data, and statistically significant results for each study.

B.1 Study Summaries

A. Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming (Baudisch et al. 2002)
This study compared three visualization techniques: (1) focus plus context screens: wall-size low-resolution displays with an embedded high-resolution display region; (2) overview plus detail; and zooming/panning to extract information from large static documents and avoid collisions in a driving simulation.

B. Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view (Baudisch et al. 2004)
A user study that helps practitioners determine which visualization technique–fisheye view, overview, or regular linear view-to pick for which type of visual search scenario in viewing webpages on browsers.

C. DateLens: A Fisheye Calendar Interface for PDAs (Bederson et al. 2004)
This study compared between two types of calendar visualizations: DateLens and Pocket PC 2002 calendar for both simple and complex tasks.

D. Fisheye Views are Good for Large Steering Tasks (Gutwin and Skopik 2003)
This study tested the effects of magnification and representation on user performance in a basic pointing activity called steering–where a user moves a pointer along a predefined path in the workspace. Researchers tested three types of fisheye at several levels of distortion, and also compared the fisheyes with two non-distorting overview + detail techniques.

E. Reading of electronic documents: the usability of linear, fisheye and overview + detail interfaces (Hornbæk and Frokjær 2001) and
Reading Patterns and Usability in Visualization of Electronic Documents (Hornbæk et al. 2003)

This study explored reading patterns and usability in visualizations of electronic documents using a fisheye, an overview + detail, and a linear interface with question answering and essay tasks.

F. Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview (Hornbæk et al. 2002)
This study compared zoomable user interfaces with and without an overview to understand the navigation patterns and usability of these interfaces using map data.

G. Untangling the Usability of Fisheye Menus (Hornbæk and Hertzum 2007)
This study investigated whether fisheye menus are useful as compared to the hierarchical menu and two variant of the fisheye menu based on known-item search and browsing tasks.

H. Evaluating a Fisheye View of Source Code (Jakobsen and Hornbæk 2006)
This study compared the usability of the fisheye view with a common, linear presentation of program source code.

I. Summary Thumbnails: Readable Overviews for Small Screen Web Browsers (Lam and Baudisch 2005)
The study compared Summary Thumbnails—thumbnail views enhanced with readable text fragments—with thumbnails, single-column interface, and a desktop interface in a number of web-informational search tasks.

J. Overview Use in Multiple Visual Information Resolution Interfaces (Lam et al. 2007)
The study looked at overview use in two multiple-VIR interfaces with high-VIR displays either embedded within, or separate from, the overviews using finding and matching tasks.

K. An Evaluation of Pan and Zoom and Rubber Sheet Navigation (Nekrasovski et al. 2006)
This study evaluated two navigation techniques with and without an overview. The techniques examined are conventional Pan and Zoom Navigation and Rubber Sheet Navigation, a rectilinear Focus+Context technique.

L. Snap-together Visualization: Can Users Construct and Operate Coordinated Visualizations (North and Shneiderman 2000)
This study explored coordination construction and operation in Snap-together visualization operating an overview-and-detail coordination, a detail-only and an uncoordinated interface to display census data.

M. The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser (Pirolli et al. 2003)
The paper presents two experiments that investigated the effect of information scent (tasks with different Accuracy of Scent scores) on performance with the Hyperbolic Tree Browser and the Microsoft Windows File Browser.

N. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation (Plaisant et al. 2002)
The study compared SpaceTree–a novel tree browser with dynamic rescaling of branches of the tree–with the hyperbolic tree browser and the Windows explorer in a series of locate, refind, and topology-related tasks.

O. Zooming, Multiple Windows, and Visual Working Memory (Plumlee and Ware 2006)
The paper presents a theoretical model of performance that models the relative benefits of these techniques when used by humans for completing a task involving comparisons between widely separated groups of objects based on a user study of zooming and multiple windows interfaces.

P. Visualization of Graphs with Associated Timeseries Data (Saraiya et al. 2005)
This study evaluated and ranked graph+timeseries visualization options based on users' performance time and accuracy of responses on predefined tasks.

Q. A Comparison of Traditional and Fisheye Radar View Techniques for Spatial Collaboration (Schafer and Bowman 2003)
This study compared an enhanced design that uses fisheye techniques with a traditional approach to radar views in spatial collaboration activities.

R. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods (Schaffer et al. 1996)
This experiment compared two methods for viewing hierarchically clustered networks: the traditional full-zoom techniques provide details of only the current level of the hierarchy; and the fisheye views, generated by the variable-zoom algorithm, that provides information about higher levels as well.

S. An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations (Shi et al. 2005)
The paper presents two experiments that compared a distortion algorithm based on fisheye and continuous zooming techniques for browsing data in the TreeMap representation with the drill-down method in browsing task with or without the need for context.

B.2 Study Interfaces, Tasks, Data, and Results

Interfaces: We classified study interfaces based on the taxonomy used in the article, as *hiVIR (H)*, *temporal (T)*, *separate (S)*, or *embedded (E)*. We listed all the categories to which the interface were categorized. For example, a zoomable interface with an overview would be classified as "*separate + temporal*", or "*S+T*". We also included the names of the interfaces if they were provided in the original study papers.

Significant Results: We listed the statistically significant time and accuracy results, using the interface taxonomy of *H*, *T*, *S*, and *E*. Even though many studies reported questionnaires and observations, we do not include them due to space constraints.

A. Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming (Baudisch et al. 2002)

Interfaces:

- T: z+p (Traditional pan-and-zoom)
- *[S+T]*: o+d (low-VIR window + a smaller *temporal*)
- E: f+c (Fixed high-res region with surrounded by low-res without distortion. Panning interaction only.)

Task(s):

1. Static task: Find route in a map
2. Static task: Verify connection in a network
3. Dynamic task: Avoid collision in a computer-game like environment

Data:

- For static tasks: spatial map data
- For dynamic tasks: a computer-game like environment with a driving scene with falling objects. Some of which were visible at low VIRs (i.e., the rocks), and some only at high VIRs (i.e., the nails)

Significant Result(s):
Time:

- $E < T$ (*Find route*; *verify connection*)
- $E < [S + T]$ (*Find route*; *verify connection*)

Accuracy:

- $E > [S + T]$ (*Avoid collision*)

B. Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view (Baudisch et al. 2004)

Interfaces:
(Note:all interfaces were augmented with semantic highlights of keywords in the documents, each keyword highlighted with a different colour)

- H: Linear (Traditional browser interface with vertical scrolling)
- S: Overview (hiVIR plus a low-VIR view showing the entire webpage fitted vertically to a fixed horizontal width)
- E: Fisheye (A non-scrollable browser with readable and non-readable texts, depending on the user selection.)

Task(s):

1. *Outdated*: check if page contained all four search terms
2. *Product choice*: find cheapest notebook with four features
3. *Co-occurrence*: check if page contained any paragraphs that contained both search terms
4. *Analysis*: check how many times Mrs. Clinton was mentioned, with "Clinton" being the search term

Data: Web documents

Significant Result(s):
Time:

- $S < H$ (*Outdated*)
- $E < H$ (*Outdated*, *Product choice*)
- $E < S$ (*Product choice*)
- $H < E$ (*Co-occurrence*)

Accuracy:

- $E > H > S$ (*Co-occurrence*)

C. DateLens: A Fisheye Calendar Interface for PDAs (Bederson et al. 2004)

Interfaces:

- T: Pocket PC (Default Pocket PC calendar, providing separate day, week, month and year views)
- E: DateLens (A Table Lens-like distortion that can show multiple levels of details simultaneously, with the default configured as a 3-month view)

Task(s):

1. *Searching*: find the start and/or end dates of appointments
2. *Navigation and Counting*: navigate to particular appointments or monthly views, and count pre-defined activities
3. *Scheduling*: schedule an event of various time spans

Data: Calendar data

Significant Result(s):
Time:

- $T < E$(*Check schedule*, *Count Mondays/Sundays in a month*, *Find the closest free Saturday night/Sunday*)
- $E < T$(*Count conflicts/free days in a 3-month period*, *Find freest/busiest two-week period in the next three months*, *Find a start date for a specific activity*, *Find freest half-day in a month*)

Percent completed task:

- $E > T$, **except** for two tasks to find schedule details about specific activities

D. Fisheye Views are Good for Large Steering Tasks (Gutwin and Skopik 2003)

Interfaces:

- (E1)Sarker-and-Brown fisheye
- (E2)Round-lens fisheye
- (E3)Flat-lens fisheye
- (S1)Panning view
- (S2)Radar view

Task(s): 2D-steering task that required participants to move a pointer along a path that is defined by objects in a visual workspace. In order to perform the task, participants needed to use the high-VIR view for accurate steering, and the low-VIR view to pan around.

Data: Abstract 2D paths: horizontal, diagonal, step, curve

Significant Result(s):
Time:

- $E \leq S$ (at all magnification levels)

Accuracy:

- $E \geq S$ (at all magnification levels)

E. Reading of electronic documents: the usability of linear, fisheye and overview + detail interfaces (Hornbæk and Frokjær 2001) and Reading Patterns and Usability in Visualization of Electronic Documents (Hornbæk et al. 2003)

Interfaces:

- H: Linear(Traditional vertically scrollable interface)
- S: Overview+Detail (*hiVIR* plus a low-VIR overview of the entire document, reduced by 1:17 in size on average, and coordinated with the high-VIR view. In the low-VIR view, only the section and subsection headers of the document were readable, with the rest of the document shrunk to fit within the available space.)

- *E*: Fisheye (Non-scrollable browser with only the most important part of the document was readable. The relative importance determined by the interface *a priori*. Participants could expand or collapse different parts of the documents by a mouse click.)

Task(s):

1. *Essay*: read a document, from memory: (a) write 1-page essay, stating the main theses and ideas of the documents; (b) answer 6 incidental-learning questions
2. *Question-answering*: answer 6 questions

Data:Electronic text documents

Significant Result(s):
Time:

- $E < H$ (*Essay*)
- $E < S$ (*Essay*)
- $H < E$ (*Question-answering*)

Effectiveness:

- $S > H$ (*Essay: Author's grading*)
- $S > E$ (*Essay: Author's grading*, *Essay: # correct incidental-learning questions*)
- $H > E$ (*Essay: # correct incidental-learning questions*)

F. Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview (Hornbæk et al. 2002)

Interfaces:

- *T*: Zoomable User Interface (ZUI) (Displayed a map, zoomable at 20 scale levels)
- *[S+T]*: ZUI with Overview (*Temporal* plus a low-VIR view that was one-sixteenth the size of the zoomable window)

Task(s):

1. *Navigation*: find a well-defined map object
2. *Browsing*: scan a large area, possibly the entire map for objects of a certain type
3. *Label cities and counties*: write down as many objects within the a map area from memory
4. *Recognize cities*: circle all cities within a county and cross out cities that were believed to be outside of the county

Data: Geographical map:

- Washington map: 3 levels (county, city and landmark)
- Montana map: single level

Significant Result(s):
Time:

- $T < [S + T]$ (*Navigation*)

Accuracy:

- $T > [S + T]$ (Washington map: *Label cities and counties*, *Recognize cities*)

G. Untangling the Usability of Fisheye Menus (Hornbæk and Hertzum 2007)

Interfaces:

- *T*: Hierarchial menu (Traditional cascading menu. For the smaller data set, the menu had two VIRs. For the larger data set, the menu had three VIRs, or two submenus.)
- *S*: (A low-VIR pane showing an index of letters of the items included in the menu, and a high-VIR pane showing menu items. The portion of the items showed was determined by the mouse position relative to length of the menu)
- *[E+S]*: Fisheye (The low-VIR pane showed an index of letters of the menu items. The high-VIR pane showed all the menu items, with a regular font-sized region surrounded by decreasing font sizes. At the two extreme ends, the items were unreadable.)
- *E*: Multi-focus (Showed two types of high-VIR regions: the mouse-selected menu items, and those that were determined to be significant based on *a priori* importance).

Task(s):

1. *Known-item search*
2. *Browsing*

Data:

- alphabetical data with 100 items
- categorical data with 292 items (4x8x8)

Significant Result(s):
Time:

- $T <$ all interfaces (*Known-item search*)

Accuracy:

- $T >$ all interfaces (*Known-item search*)

H. Evaluating a Fisheye View of Source Code (Jakobsen and Hornbæk 2006)

Interfaces:

- H: Linear (Vertically scrollable and displayed all the program lines)
- E: Fisheye (No vertical scrolling, but selectively displaying semantically relevant parts of the source code based on the lines displayed in the focal region. The selection was determined by a modified version of Furnas' degree-of-interest function (Furnas 1986), where semantic distance was also considered along with syntactic distance and *a priori* significance.)

Task(s):

1. *One-step navigation*
2. *Two-step navigation*
3. *Determine field encapsulation*
4. *Determine delocalization*
5. *Determine control structure*

Data: Program source code

Significant Result(s):
Time:

- $E < H$ (*Two-step navigation*: 15%, *Determine delocalization*: 30%)

I. Summary Thumbnails: Readable Overviews for Small Screen Web Browsers (Lam and Baudisch 2005)

Interfaces:

- T: Summary Thumbnail / Thumbnail (Scaled-down image of the original webpage fitted to the width of the PDA screen, with or without preserving the readability of the text)
- H: Desktop (Original, unscaled desktop-sized webpage)

Task(s): Information searches

Data: Web documents

Significant Result(s): No significant differences in performance time or task accuracy

J. Overview Use in Multiple Visual Information Resolution Interfaces (Lam et al. 2007)

Interfaces:

- H: hiVIR (Stacked line graph plots, encoding the x and the y line graph values with space, and the y-values doubly encoded with colour.)
- S: separate (Low-VIR interface with strips that encode the y-values of the line graph data with colour alone. Mouse-click on strip displays high-VIR plots in a separate panel.)
- E: embedded (Low-VIR regions of strips. Mouse-click on strip displays high-VIR plots in place.)

Task(s):

1. *Find highest point*
2. *Find most number of peaks in line graph*
3. *Match a small region of line graph*
4. *Match entire line graph*

Data: 140 line graphs, each with 800 data points

Significant Result(s):
Time:

- $S < H$ (*Find highest point*)
- $E < H$ (*Find highest point*)

K. An Evaluation of Pan and Zoom and Rubber Sheet Navigation (Nekrasovski et al. 2006)

Interfaces:

- T: PNZ (The traditional pan and zoom interface augmented with a visual cue to indicate the location of the target branch as coloring of the node regardless of the allotted screen presence)
- E: RSN (Implemented the Rubber Sheet Navigation (Sarkar et al. 2003), augmented with a Halo-like arc served as the visual cue (Baudisch and Rosenholtz 2003), as the actual target may be off screen)
- *[T+S]*, *[E+S]*: PNZ+OV, RNS+OV (Add low-VIR overview in addition to their *temporal* or to their *embedded* views)

Task(s): Compare the topological distances between colored nodes in a large tree and determine which of the distances was smaller

Data: Large trees

Significant Result(s):
Time:

- $T < E$
- $[T + S] < [E + S]$

L. Snap-together visualization: can users construct and operate coordinated visualizations (North and Shneiderman 2000)

Interfaces:

- H: detail-only (Displayed census information grouped by geographic states)
- S: coordination / no-coordination (*hiVIR* plus a low-VIR pane that displayed an alphabetical list of states included in the census)

Task(s):

1. *Coverage*: answer present or absent of objects
2. *Overview patterns*
3. *Visual / nominal lookup*
4. *Compare two or five items*
5. *Search for target value*
6. *Scan all*

Data: United States census data

Significant Result(s):
Time:

- $S(\pm coord) < H$ (*Coverage*, *Overview patterns*)
- $S(+coord) < H|S(-coord)$ (*Nominal lookup*, *Compare*, *Search*, *Scan*)

M. The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser (Pirolli et al. 2003)

Interfaces:

- T: Microsoft File Browser
- E: Hyperbolic tree browser (Lamping et al. 1995)

Task(s):

1. *Information Retrieval*: simple, complex
2. *Comparison*: local, global

Data: CHI'97 BrowseOff tree, except trimming it to four levels with 1436 nodes, and 66 nodes at the lowest level

Significant Result(s):
Time:

- $E < T$ (High-scent tasks)
- $T < E$ (Low-scent tasks)

N. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation (Plaisant et al. 2002)

Interfaces:

- T: Microsoft Explorer file browser
- $E_{hyperbolic}$: Hyperbolic tree browser (Lamping et al. 1995) (Lays out a tree based on a non-Euclidian hyperbolic plane)
- $E_{spaceTree}$: SpaceTree (Dynamically rescales the tree branches for the available screen space, preserves ancestral nodes but elides the rest into a triangular icon)

Task(s):

1. *Node searches*
2. *Search of previously visited nodes*
3. *Topology questions*

Data: CHI'97 BrowseOff tree with over 7000 nodes

Significant Result(s):
Time:

- $T < E_{hyperbolic}$ (*Node searches*: 1 out of 3 tasks)
- $E_{spaceTree} < T$ (*Node searches*: 1 out of 3 tasks)
- $T < E_{hyperbolic}$ (*Refind previously visited nodes*)
- $E_{spaceTree} < E_{hyperbolic}$ (*Refind of previously visited nodes*)
- $T < E_{spaceTree}$ (*Refind of previously visited nodes*)
- $E_{spaceTree} < T$ (*Topology*: list all ancestor nodes)
- $E_{hyperbolic} < E_{spaceTree}$ (*Topology*: local topology)

Accuracy:

- $E_{spaceTree} > E_{hyperbolic} > T$ (*Refind of previously visited nodes*, *Topology*: overview)

O. Zooming, Multiple Windows, and Visual Working Memory (Plumlee and Ware 2006)

Interfaces:

- T: Zooming (Continuous zoom mechanism)
- S: Multiple Windows (Two VIRs: up to two high-VIR windows selected from a low-VIR view. The targets were clusters of 3-D geometric objects. Their low-VIR view showed only the location of the candidate targets, but not the details. At the intermediate levels, the target locations and details were camouflaged by the textured background. The highest VIR presented enough target details for the visual comparison)

Task(s): Multiscale comparison task to find a cluster that matched the sample set of 3D objects.

Data: Six targets, each a cluster of 3D geometric objects with 1 to 7 items, each item taken from five possible shapes

Significant Result(s):
Time:

- $T < S$ (sets with one or two items)
- $S < T$ (sets with five or seven items)

Accuracy:

- $S < T$

P. Visualization of Graphs with Associated Timeseries Data (Saraiya et al. 2005)

Interfaces:

- H: Multiple-Attribute Single-View (MS) (Displayed all 10 time points simultaneously as simple glyphs, representing the nodes of the graph)
- T: Single-Attribute Single-View (SS) (Displayed the value of the time points as colour of the nodes, linked with a user-controlled slider bar to view the other nine time points)

Task(s):
1 time point:

- Read value, search node

2 time points:

- Determine change in values

10 time points:

- Determine time trend, topology trend
- Search time point, search trend
- Identify a outlier group

Data: 50-node graph, each node showing a timeseries with 10 time points

Significant Result(s):
Time:

- $T < H$ (*Topology trend*,)
- $H < T$ (*Outlier*, *Search time point*)

Accuracy:

- $T \geq H$ (all tasks **except** *Outlier*)
- $H > T$ (*Outlier*)

Q. A Comparison of Traditional and Fisheye Radar View Techniques for Spatial Collaboration (Schafer and Bowman 2003)

Interfaces:

- S: Traditional (Contained a low-VIR view linked to a high-VIR view)
- E: Fisheye (Fisheye low-VIR view coupled with a high-VIR view)

Task(s): Collaborative traffic and road-sign positioning. 2 participants, each with partial information to position signs

Data: Map

Significant Result(s): Participants required less verbal communications with E than S

R. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods (Schaffer et al. 1996)

Interfaces:

- T: Full-Zoom (Displayed children nodes of a single parent at the same level)
- E: Fisheye (Displayed the same children nodes along with all the ancestral nodes acting as context)

Task(s): Find and repair a broken telephone line in the network by rerouting a connection between two endpoints of the network that contained the break

Data: Hierarchical network of 154 nodes with 39 clusters

Significant Result(s):
Time:

- $E < T$ (*Repair*)

S. An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations (Shi et al. 2005)

Interfaces:

- T: Drill-Down (Traditional TreeMap display, where the display showed only nodes from the same level of the same branch of the tree)
- E: Distortion (Retained all the ancestral levels of the displayed nodes, using distortion to fit all the nodes in the display)

Task(s):

1. *Browsing*: find an image
2. *Browsing with Context*: find target based on its neighboring images and their interrelations, or context, defined as "a set of images spatially and hierarchically related in a certain configuration" (p. 86) This context was held constant for all trials, and involved multiple levels of the tree

Data:
2 hierarchies, both had 30 different images and >300 files of other formats

- Deep: 6 levels, <=3 subdirectories/level
- Wide: 3 levels, <=6 subdirectories/level

Significant Result(s):
Time:

- $E < T$ (*Browsing*: 65% faster with wide, 156% faster with deep; *Browsing with Context*: 61% faster with wide, 84% faster with deep)

Effectiveness:

- $E > T$ (gave up)
- $T > E$ (timed out)

Appendix C

Visual-Memory Experiment Materials

This appendix includes materials for the laboratory experiment of the thesis detailed in Chapter 5, including the informed consent, participant instructions given during the study, sample experimental stimuli used in the study, experimental design of presentation ordering, and detailed analysis results.

C.1 Informed Consent

Consent Form and Sign-in Sheet

UBC Visual Cognition Laboratory Kenny Building Room 3204 Phone: 604-822-9653
Ethical Clearance# B05-0391
Principal Investigators: Dr. Ronald A. Rensink, Heidi Lam

In this experiment your task will be to indicate whether or not you remember an image as having been shown to you.

You will receive $10.00 cash for your one hour of voluntary participation. If you would like to stop the experiment at any time, you may do so without penalty. The data from your experimental session will be coded by number so that your anonymity will be preserved.

This experiment is being conducted in the Visual Cognition Laboratory of the Psychology Department of the University of British Columbia, under the guidance of Dr. Ronald Rensink. If you have any questions, please feel free to ask.

If you agree to participate in this experiment, please sign-in below.

*indicate if you are wearing
G: glasses
C: contacts
N: no corrective eyewear

S	**Name**	Gender **M/F**	Handed-ness **R/L**	Eyewear* **G/C/N**	**Age**	**Date**	**Time**	**Signature**
0								
1								
2								
3								

C.2 Participant Instructions

Instructions were incorporated into the experiment software and were displayed at various point during the experiment. Before the experiment, participants were briefed about the overall procedure and shown samples of original and transformed images for the type of transformations tested.

Instructions for the learning phase for both the training and the actual block

You will be shown 8 images for 12 seconds each, one after another. The images consist of a number of dots on the screen joined by lines. Your task is to study the images carefully so that you can recall them later on in the experiment. This memory task is difficult, so please take the time to study the images carefully.

Instructions for the recognition phase for both the training block using untransformed images

You will be shown 8 images, half of which you have already seen in the previous part. For each image, please press the "A" key if you have seen it in the last block, or the "L" key if you haven't.

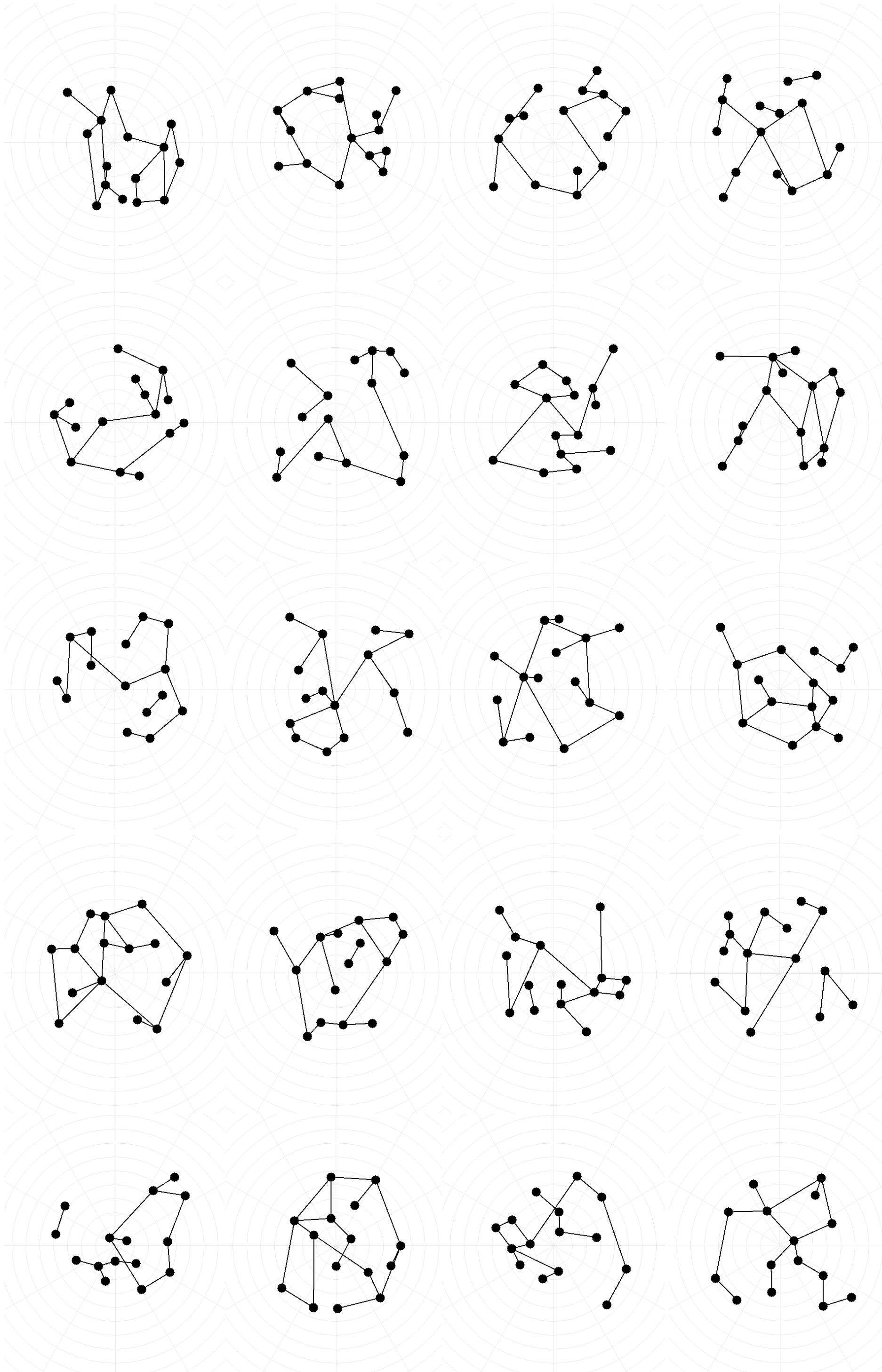
Instructions for the recognition phase for both the training block using both untransformed and transformed images

You will be shown 8 images, half of which you have already seen in the previous part. Some of the images may be transformed as shown to you by the experimenter, but will still be considered as the "have been seen".

For each image, please press the "A" key if you have seen it in the last part, or the "L" key if you haven't.

C.3 Sample Experimental Stimuli

20 images actually used in the study are included to provide a better sense of the experimental stimuli:



C.4 Experimental Design

For each transformation type, we tested five levels of transformation degree (denoted as 1, 2, 3, and 4), with the first level being the zero transformation baseline (denoted as 0). To fully counterbalance the presentation order of the five levels for each experiments, we need to recruit 5!, or 120, participants for each experiment. Since we only recruited 20 participants for each experiment, we could only partially counterbalance our presentation order as follows:

(0, 1, 2, 3, 4), (1, 3, 4, 2, 0), (2, 0, 3, 4, 1), (3, 4, 0, 1, 2), (4, 2, 1, 0, 3),
(0, 2, 1, 4, 3), (2, 4, 3, 1, 0), (1, 0, 4, 3, 2), (4, 3, 0, 2, 1), (3, 1, 2, 0, 4),
(0, 3, 4, 1, 2), (3, 1, 2, 4, 0), (4, 0, 1, 2, 3), (1, 2, 0, 3, 4), (2, 4, 3, 0, 1),
(0, 4, 3, 2, 1), (4, 2, 1, 3, 0), (3, 0, 2, 1, 4), (2, 1, 0, 4, 3), (1, 3, 4, 0, 2)

C.5 Experimental Result Analysis

C.5.1 Single-factor ANOVA results

This section shows results of the single-factor repeated measure ANOVA for the 10 experiments plus the two follow up studies reported in Chapter 5. For all tables in this section, Sum Sq denotes Type III Sum of Squares; df denotes Degree of freedom; Mean Sq denotes Mean Square; and a * denotes a situation where the Greenhouse-Geisser adjustment was applied. The partial eta squared (η_p^2) was also included as an estimate of effect size. Time values are expressed in milliseconds.

Source	**Grid**	**Sum Sq**	**df**	**Mean Sq**	**F**	**p-value**	η_p^2
Scale*		5.36E6	2.28	2.35E6	0.67	0.54	0.03
Error (Scale)*		1.53E8	43.24	3.54E6			
Scale	R	2.50E6	4	6.24E5	0.60	0.67	0.03
Error (Scale)	R	7.95E7	76	1.05E6			

Table C.1: ANOVA time results for the scaling experiments. Scale = Scaling Transformation; R = rectangular grid.

Source	Grid	Sum Sq	df	Mean Sq	F	p-value	η_p^2
Rotate*		4.39E7	1.88	2.33E7	2.92	0.07	0.13
Error (Rotate)*		2.86E8	35.78	7.80E6			
Rotate*	R	1.57E7	2.61	5.99E6	1.33	0.28	0.07
Error (Rotate)*	R	2.23E8	49.67	4.49E6			
Rotate Ext	R	5.27E7	4	1.31E7	5.05	0.001	0.21
Error (Rotate Ext)	R	1.98E8	76	2.60E6			

Table C.2: ANOVA time results for the rotation experiments. Rotate = Rotation Transformation; Rotate Ext = Rotation Transformation extended experiment.

Source	Grid	Sum Sq	df	Mean Sq	F	p-value	η_p^2
RFish*		3.46E7	1.90	1.82E7	2.84	0.07	0.13
Error (RFish)*		2.32E8	36.17	6.41E6			
RFish*	R	1.67E7	2.78	6.01E6	2.63	0.06	0.12
Error (RFish)*	R	1.21E8	52.89	2.28E6			
RFish*	P	5.83E7	1.87	3.11E7	3.32	0.05	0.15
Error (RFish)*	P	3.34E8	35.53	9.39E6			

Table C.3: ANOVA time results for the rectangular fisheye experiments. RFish = Rectangular Fisheye Transformation; R = rectangular grid; P = polar grid.

Source	Grid	Sum Sq	df	Mean Sq	F	p-value	η_p^2
PFish*		2.44E7	1.82	1.34E7	2.34	0.12	0.11
Error (PFish)*		1.97E8	34.51	5.73E6			
PFish	R	5.32E7	4	1.33E7	4.32	0.003	0.19
Error (PFish)	R	2.34E8	76	3.08E6			
PFish	P	4.45E7	4	1.11E7	6.08	<0.0001	0.24
Error (PFish)	P	1.39E8	76	1.83E6			

Table C.4: ANOVA time results for the polar fisheye experiments. PFish = Polar Fisheye Transformation; R = rectangular grid; P = polar grid.

C.5.2 Post-hoc analysis results

The following table lists all statistically significant pairwise comparison post-hoc analysis of the repeated-measure ANOVA time results in Appendix C.5.1. Both the significance levels and the 95% confidence intervals were adjusted for multiple comparisons using the Bonferroni correction. Time values are expressed in milliseconds.

Level 1	Level 2	Mean Diff	Std Error	Sig	95% CI
Rotate 0	Rotate 60	-1946	482	.007	-3476 — -415
Rotate 0	Rotate 90	-1455	440	.037	-2850 — -60
Rotate 0 R	Rotate 90 R	-1823	559	.041	-3596 — -50
Rotate 0 R	Rotate 180 R	-1788	433	.006	-3164 — -410
RFish 0.5	RFish 2	-1315	411	.048	-2621 — -9
RFish 0.5	RFish 3	-1449	418	.026	-2776 — -123
RFish 0 R	RFish 3 R	-1087	334	.041	-2146 — -29
PFish 0 R	PFish 3 R	-2128	617	.027	-4086 — -172
PFish 0 P	PFish 3 P	-1838	380	.001	-3043 — 633
PFish 1 P	PFish 3 P	-1087	418	.004	-3134 — 480

Table C.5: Statistically significant pairwise comparisons of time results. Mean Diff = difference between Level 1 and Level 2; Std Error = Standard Error; Sig = The mean difference is significant at the .05 level; 95% CI = 95% confidence interval for the difference with lower and upper bounds. RFish = Rectangular Fisheye Transformation; PFish = Polar Fisheye Transformation; R = rectangular grid; P = polar grid.

Appendix D

Overview-Use Study Materials

This appendix includes materials for the experimental-simulation study of the thesis detailed in Chapter 6, including the informed consent, participant instructions given prior and during the study, and the questionnaire to solicit participants' subjective feedback on the interfaces at the end of the study.

D.1 Informed Consent

The University of British Columbia
Department of Computer Science
201-2366 Main Mall
Vancouver, B.C. V6T 1Z4
Phone: (604) 822-3061, Fax: 1-604-822-5485

Consent Form

Comparative study on a novel visualization technique for high-volume line-graph data

Principal Investigator:
Name: Dr. Tamara Munzner (Professor)
Department: Computer Science, UBC
Phone:
Email:

Co-investigator:
Name: Heidi Lam (PhD student)
Department: Computer Science, UBC
Phone:
Email:

Purpose of the Experiment
This experiment is conducted to evaluate the effectiveness of two visualization techniques to display line-graph (XY plots) data on a single computer screen.

Study Procedures
In this experiment your task will be asked to answer a number of questions based on the data displayed to you by two different interfaces on the screen. Prior to the actual experiment, you will be given a training session where you will be familiarized with the two interfaces and the format of the questions. There will be two questions about the data in this session. The actual sessions will be in the same format as the training session, except that there will be 20 questions per session.

Risks
We intend for your experience in this study to be pleasant and stress-free. There are no known medical or psychological risks associated with this research, an experimental session should feel equivalent to viewing a TV program or playing a computer game. You will have the opportunity to take breaks between sessions during the experiment. During the experiment, if you feel fatigue or any discomfort, please notify the investigator. .

Confidentiality
Your identity will be kept strictly confidential. We will ensure that all recorded data are accessible only by project investigators and are kept secure in a locked faculty office for paper documents, or on a Computer Science department file server protected by password for electronic documents. All data from individual participants will be coded so that your anonymity will be protected in any publicly available reports, papers, and presentations that result from this work.

Compensation
You will receive an honorarium of 10 Canadian dollars per hour for your participation. Experiments typically last between one and two hours.

Contacts and usage for information about the study
This research study is funded by the operating and strategic grant programs of the Natural Sciences and Engineering Research Council of Canada (NSERC). Portions of this research will be used for graduate theses. If you have any questions or desire further information about this study, you may contact Dr. Tamara Munzner by email or phone as listed above. We would be pleased to explain to you the purpose and methods used in this study in academic detail after your participation has concluded, and to furnish you with our results when they become available.

If you have any concerns about your rights or treatment in this or any other UBC experiment, you may contact the Research Subject Information Line in the UBC Office of Research Services at (604) 822-8598. You have been given a copy of this consent form for your records. You do not waive any legal rights by signing this consent form.

Consent
Your participation in this study is entirely voluntary. You are free to withdraw at any time, without any repercussions to you whatsoever. If you chose to withdraw, your reimbursement will be pro-rated for the amount of time that you spent participating in the experiment.

Your signature below indicates that you have received a copy of this consent form for your own records, and that you consent to participate in this study.

____
Subject Signature Date

____
Printed Name of the Subject

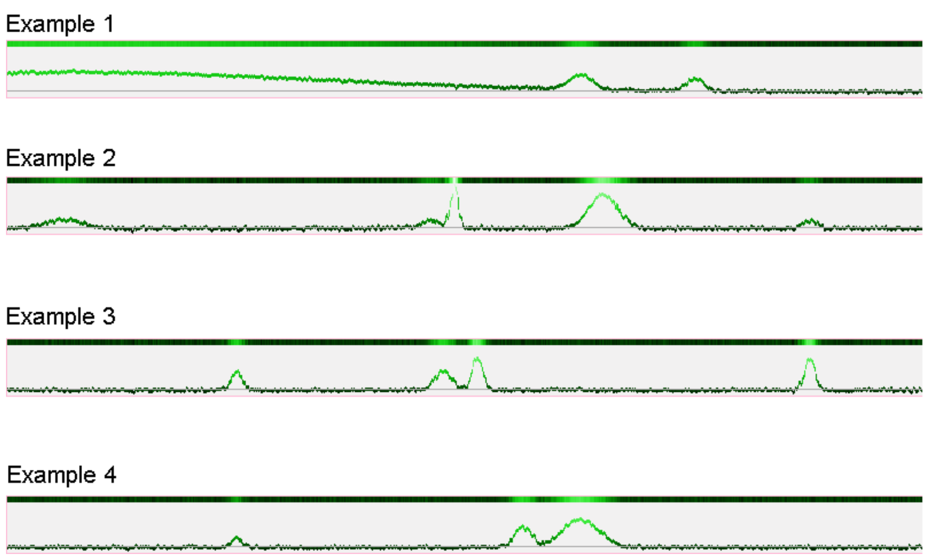
D.2 Participant Instructions

Two types of instructions were provided to participants for the study: verbal instructions to brief participants about the study, and written instructions incorporated into the experimental software interface.

D.2.1 Verbal briefing instructions

The first is verbal instructions given prior to the study to brief participants about the overall study procedure and show them samples of two visual encodings used in the study. The script used in this briefing is as follows:

You will be working with a collection of 140 xy-plots, or what we call power profiles using four different interfaces. Each plot has a number of power surges, which are essentially peaks in the xy-plot. There are two ways to display these graphs. I will show you some examples (show the following picture):



(Pointing to Example 1, the detail graph) Here is one example of a xy-plot. You can see the baseline (point to the flat portion), and some peaks above the baseline (point to the peaks). We call these peaks power surges, and they can be of different heights and widths. The thin strip above the detail graph is the compressed version (point to the overview graph). We call that the overview mode. Instead of showing the heights of the peaks by vertical space and colour, we use only colour. You will see a scale for the colour coding for each task.

(Pointing to Example 2, 3, and 4, double peak) Sometimes the peaks run into each other. If you were asked to count them, they count as 2 peaks.

Any questions so far? (Let participant study the examples)

So you will be answering a bunch of questions based on the data presented as these graphs, sometimes in the detail mode, sometimes in the overview mode, and sometimes, in both. For each interface, you will first be trained with four tasks, and repeat those four tasks again for the actual study. You will repeat the same four tasks in all the other interfaces. So in all, you do the same four tasks four times. There will be a 5-minute break at mid-point.

Each interface is slightly different. You will have the chance to play and study with each interface and find the best way to do the tasks during the training sessions. I will be helping you during the training session, just ask me if you have any questions.

I will be taking notes on how you solve the tasks during the actual study to collect some observations. Just don't mind me.

D.2.2 Instructions on the study software interface

Instructions were incorporated into the study software. As shown in Figures D.1 and D.2, the far right panels showed task instructions for each trial. The top of the panel showed information on visual encoding and available interface interactions. Beneath that showed task instructions, as provided in Table 6.2. On the bottom showed study control buttons: `Show Data` and `Answer Ready`.

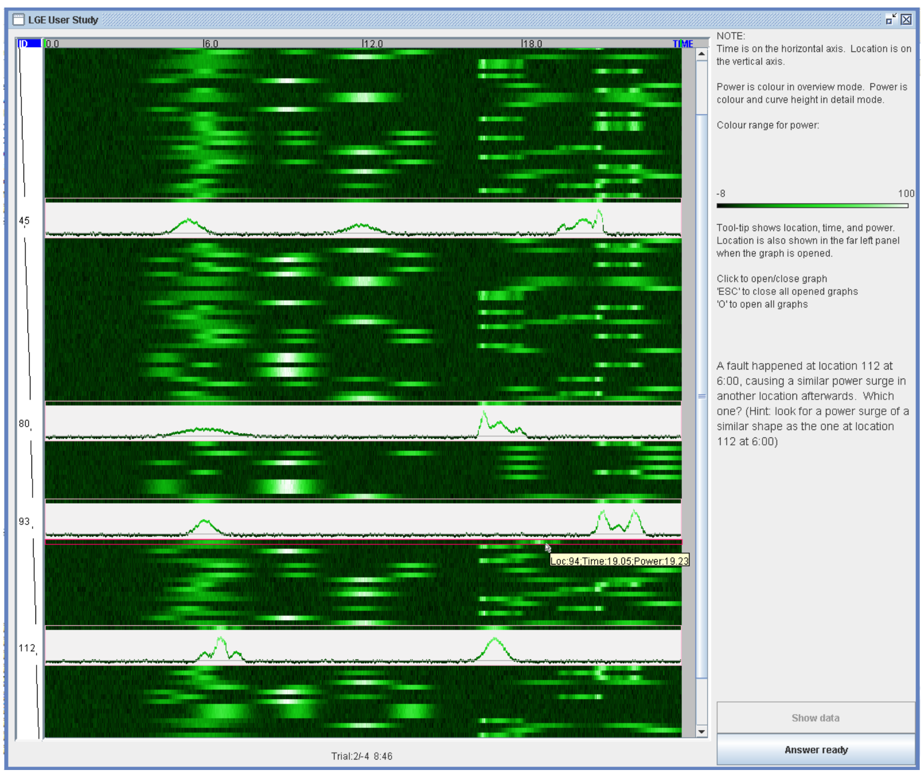


Figure D.1: Experimental software interface showing the *Embedded* interface displaying data for the *Shape* task.

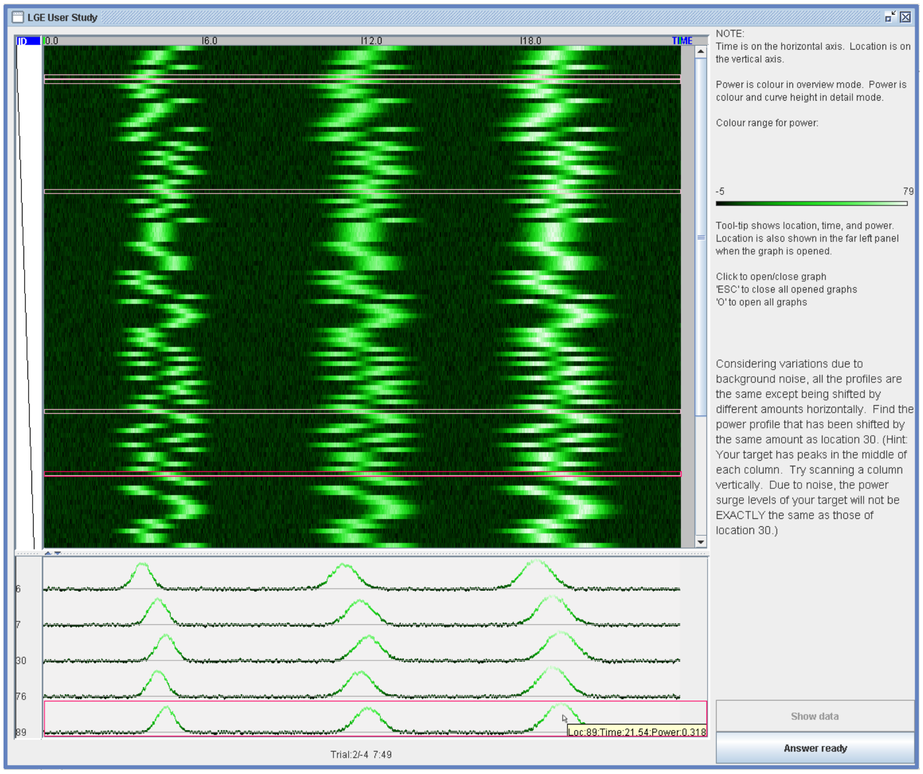


Figure D.2: Experimental software interface showing the *Separate* interface displaying data for the *Compare* task.

Written instructions were displayed at various points during the study as pop-up dialog boxes.

Instructions at the start of the study

You are an electric power management engineer in a control room. Your task is to monitor power consumptions of 140 locations in the area. For each location, the power consumption is shown as a time-power graph. Each graph consists of a number of power surges as peaks of various heights and widths. Sometimes, these peaks may overlap.

There are 4 training and 4 actual task(s) for each of the 4 interfaces in this study. The description of each task and the question you need to answer is displayed on the right hand panel of the screen. Once you understand the task, please click the ‘Show data’ button to display the data you need to answer the question. This will start the clock for the trial. You have a time limit of 10 minutes for each training trial, and 5 minutes for each actual trial.

Instructions for the start of each training block

Please take the time to try out all the above interactions and experiment different ways to find the answer during the 4 training trial(s). This experimentation will save you time in the long run.

[interface-specific instructions in Section D.2.2]

When you have the answer, please click the 'Answer ready' button to display the answer box. This will stop the clock for the trial. Enter the answer, and press 'Next' for the next trial.

There is one and only one unique and obvious answer for each task. Please press 'OK' to start the training.

Instructions for the four interface blocks

1. For the *HiVIR* interface, the line graphs will ONLY be displayed in the overview mode. Individual time-power graphs are displayed as thin strips, with the power levels colour coded.

   The available interactions are as follows:

   - Left mouse click: mark/unmark graph
   - 'ESC' button press: ummark all previously marked graphs
   - 'O' button press: mark all graphs
   - Mouse hover: highlight the entire graph and show a tool-tip with $< location >; < time >; < power >$

2. For the *LoVIR* interface, the line graphs will ONLY be displayed in the detail mode. Power levels are encoded both by space and colour.

   The available interactions are as follows:

   - Left mouse click: mark/unmark graph
   - 'ESC' button press: ummark all previously marked graphs
   - 'O' button press: mark all graphs
   - Mouse hover: highlight the entire graph and show a tool-tip with "$< location >; < time >; < power >$"

3. For the *Embedded* interface, the line graphs will first be displayed in the overview mode. Individual time-power graphs are displayed as thin strips, with the power levels colour coded. In the detail mode, power levels are encoded both by space and colour. You have the option of displaying just in the overview mode, or in both the overview and the detail modes, with the overview graph stacked on top of the detail graph.

   The available interactions are as follows:

   - Left mouse click on an overview graph: open/close detail graph
   - Left mouse click on a detail graph: close detail graph

   - ‘ESC’ button press: close all previously opened detail graphs
   - ‘O’ button press: open all detail graphs
   - Mouse hover (overview/detail graph): highlight the entire graph and show a tool-tip with “$< location >; < time >; < power >$”

4. For the *Separate* interface, the line graphs will first be displayed in the overview mode. Individual time-power graphs are displayed as thin strips, with the power levels colour coded. In the detail mode, power levels are encoded both by space and colour. You have the option of displaying just in the overview mode, or in both the overview and the detail modes, each in their own resizable panel.

   The available interactions are as follows:

   - Left mouse click on an overview graph: open/close detail graph in the bottom panel
   - Left mouse click on a detail graph: close detail graph in the bottom panel
   - ‘ESC’ button press: close all previously opened detail graphs
   - ‘O’ button press: open all detail graphs
   - Mouse hover (overview/detail graph): highlight the entire graph and show a tool-tip with “$< location >; < time >; < power >$”
   - Mouse drag over overview/detail panel division line: resize overview/detail panels.

Instructions at the end of each training block / start of actual block

You have completed the training session. If you are comfortable with the interface and the task, please press the OK button to start the actual experiment. Otherwise, please contact the experimenter for additional training.

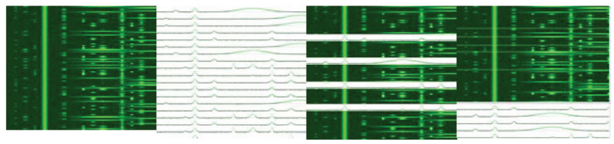
Instructions at the end of the study

Congratulations! You have completed the study. Please fill out the questionnaire. Thank you for your time!

D.3 Participant Questionnaires

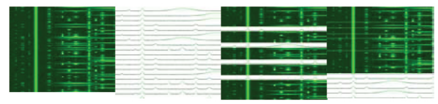
LGE Study: post-test questionnaire 1 of 5

How effective are the interfaces?
(*Select one of the circles for each interface)*

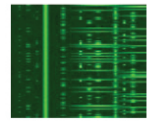


Statements	Overview ONLY	Detail ONLY	Detail in overview	Detail separated from overview
	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree
It is easy to find the data using the display.				
It is easy to compare between data using the display.				
It is easy to navigate within the display.				
It is easy to get disoriented using the display.				
It is easy to remember individual power profiles using the display.				
	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree
It is easy to fun and enjoyable using the display.				
It requires a lot of effort to use the display.				
It is frustrating to use the display.				
I have confidence in my answer when using the display.				
	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree	Strongly disagree … Strongly agree

Which interface do you prefer for the following tasks? (*check all that applies)*



Tasks	**Overview ONLY**	**Detail ONLY**	**Detail in overview**	**Detail separated from overview**
If a fault happened at location <x>, causing a power surge in location <x> and a similar one in another location. Which one?	☐	☐	☐	☐
Find the power profile that is the same as that of location 41. (wide separation)	☐	☐	☐	☐
Find the power profile that is the same as that of location 41. (narrow separation)	☐	☐	☐	☐



For the "overview only" interface:

List the three most **negative** aspect(s) of the display when performing the study task.

.

.

.

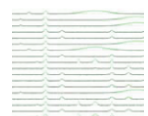
List the three most **positive** aspect(s) of the display when performing the study task.

1.

2.

3.

Please indicate any further comments on the "overview only" interface:



For the "detail only" interface:

List the three most **negative** aspect(s) of the display when performing the study task.

1.

2.

3.

List the three most **positive** aspect(s) of the display when performing the study task.

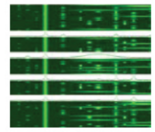
1.

2.

3.

Please indicate any further comments on the "detail only" interface:

LGE Study: post-test questionnaire 4 of 5



For the "detail in overview" interface:

List the three most **negative** aspect(s) of the display when performing the study task.

1. ______
2. ______
3. ______

List the three most **positive** aspect(s) of the display when performing the study task.

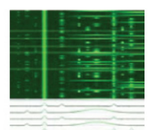
1. ______
2. ______
3. ______

Please indicate any further comments on the "detail in overview" interface:

______

______

______



For the "detail separated from overview" interface:

List the three most **negative** aspect(s) of the display when performing the study task.

1. ______
2. ______
3. ______

List the three most **positive** aspect(s) of the display when performing the study task.

1. ______
2. ______
3. ______

Please indicate any further comments on the "detail separated from overview" interface:

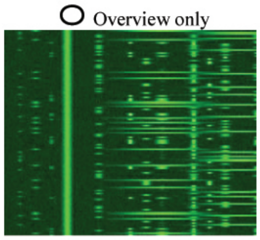
______

______

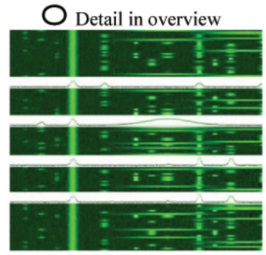
______

My **overall** preferred interface for all the study tasks is (*check one*):

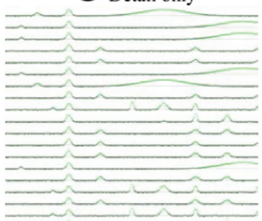
O Overview only



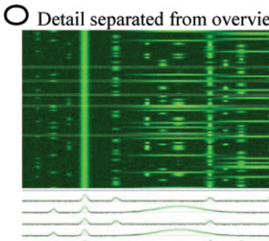
O Detail in overview



O Detail only



O Detail separated from overview



Please indicate any overall comments here:

End of the questionniare. Thank you for your time!

Appendix E

Session Viewer Field Evaluation Materials

This appendix contains two interview scripts for the Session Viewer field evaluation detailed in Chapter 8.

E.1 Pre-Session Interview Script

I'm going to ask you to think back to the last session logs you analyzed. (give a minute)

1. To start, let's talk about [Why do you look at logs / What are you trying to find when you analyze sessions / What are the main goals of your analysis?]Are those typical goals for you? (how so? how not?)
   (a) What is it about the logs that make you use them?Which aspects / properties of the logs are most important to your analyses?
   (b) How do you go about analyzing / examining the session logs?
   (c) Is there a process you follow? Can you describe it? Show me? [may mix in with #4]
2. How do you come up with analysis questions? Hypotheses? How do you confirm your hypotheses?
3. Do you tend to look at aggregates or individual sessions?
4. Which software tools do you generally use for log analysis?
   (a) Are they commercial software? Google software? or self-built?Can you show that to me?
5. Do you often use those tools? Which other tools do you use and why? Are they effective?

E.2 Post-Session Interview Script

Reflecting on the analysis you just did...

1. What did you find / learn about your data?How do these findings compare to those you normally find?
2. How does using Session Viewer compare to the tools that you normally use?
   (a) How does the scope/levels of the data you examined using Session Viewer differ than with your usual sets of tools?
3. Do you think about your data differently when using Session Viewer?
4. Do you ask different questions / form different hypotheses when using Session Viewer?
5. In the analyses you just did, in what ways did Session Viewer aid your analytic precesses? And ways in which it impeded them?
   (a) Which panels / features of Session Viewer best/least supported your analysis? Why?
6. In terms of those panels/feature not found to be useful, are there circumstances under which you could see yourself using them? Which?

Appendix F

UBC Research Ethics Board Certificates

This appendix includes all Certificates of Approval for the research conducted in this thesis administrated by the UBC Research Ethics Board.

The University of British Columbia
Office of Research Services and Administration
Behavioural Research Ethics Board

Certificate of Approval

PRINCIPAL INVESTIGATOR	DEPARTMENT	NUMBER
Munzner, T	Computer Science	**B06-0119**

INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT

UBC Campus ,

CO-INVESTIGATORS:

Booth, Kellogg, Computer Science; McGrenere, Joanna, Computer Science; Rensink, Ronald, Psychology

SPONSORING AGENCIES

Natural Science Engineering Research Council

TITLE:

Information Visualization System and Technique Evaluation

APPROVAL DATE	TERM (YEARS)	DOCUMENTS INCLUDED IN THIS APPROVAL:
FEB 21 2006	1	Jan. 31, 2006, Advertisement / Consent form

CERTIFICATION:

The application for ethical review of the above-named project has been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approved on behalf of the Behavioural Research Ethics Board
by one of the following:
Dr. Peter Suedfeld, Chair,
Dr. Susan Rowley, Associate Chair
Dr. Jim Rupert, Associate Chair
Dr. Arminee Kazanjian, Associate Chair

This Certificate of Approval is valid for the above term provided there is no change in the experimental procedures

The University of British Columbia
Office of Research Services
Behavioural Research Ethics Board
Suite 102, 6190 Agronomy Road, Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR:	DEPARTMENT:	UBC BREB NUMBER:
Ronald Rensink	UBC/Arts/Psychology, Department of	H05-80391

INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:

Institution	Site
UBC	Vancouver (excludes UBC Hospital)

Other locations where the research will be conducted:
N/A

CO-INVESTIGATOR(S):

N/A

SPONSORING AGENCIES:

Natural Sciences and Engineering Research Council of Canada (NSERC) - "Space and Structure in Attentional Processing"
The Boeing Co.

PROJECT TITLE:
Space and Structure in Attentional Processing

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: July 6, 2008

AMENDMENT(S):

AMENDMENT APPROVAL DATE:
February 6, 2008

Document Name	Version	Date
Consent Forms:		
Consent Form	2008-01-29	January 29, 2008
Other Documents:		
Complete List of Stimuli	2008-01-29	January 29, 2008

The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approval is issued on behalf of the Behavioural Research Ethics Board
and signed electronically by one of the following:

Dr. M. Judith Lynam, Chair
Dr. Ken Craig, Chair
Dr. Jim Rupert, Associate Chair
Dr. Laurie Ford, Associate Chair
Dr. Daniel Salhani, Associate Chair
Dr. Anita Ho, Associate Chair