A Study-Based Guide to Multiple Visual Information Resolution Interface Designs

HEIDI LAM
and
TAMARA MUNZNER
University of British Columbia

Displaying multiple visual information resolutions (VIRs) of data has been proposed for the challenge of limited screen space. We review 19 existing multiple-VIR interface studies and cast our findings into a four-point decision tree: (1) When is multiple VIR 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 recommend that VIR and data levels should match, and low VIRs should only display task-relevant information. Simultaneous display, rather than temporal switching, is suitable for tasks with multi-level answers.

Categories and Subject Descriptors: H.5 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces: Evaluation / methodology

General Terms: Information Visualization
Additional Key Words and Phrases: focus and context, overview and detail, zoomable user interfaces, fisheye view

1. INTRODUCTION

Visualization designers often need to display large amounts of data that exceed the display capacity of the output devices, and arguably the perceptual capacity of the users. Displaying the data at multiple visual information resolutions (VIRs) has been suggested as a workaround for this design challenge. Examples of multiple-VIR interfaces include zooming, focus + context, and overview + detail interfaces.

Even though it is generally believed that visualization interfaces should provide more than one visual resolution of the data (e.g., p. 307, [Card et al. 1999]), we are still uncertain as to when and how multiple-VIR interfaces are effective despite numerous evaluation efforts [Furnas 2006]. The difficulty in studying these interfaces reflects their complexity; a large number of factors are at play that significantly affect their use. Some of these factors include the match between task information requirement 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.

In this review, we analyze 19 existing multiple-VIR interface studies to get a clearer snapshot of our current understanding of multiple-VIR interface use, and

---

...

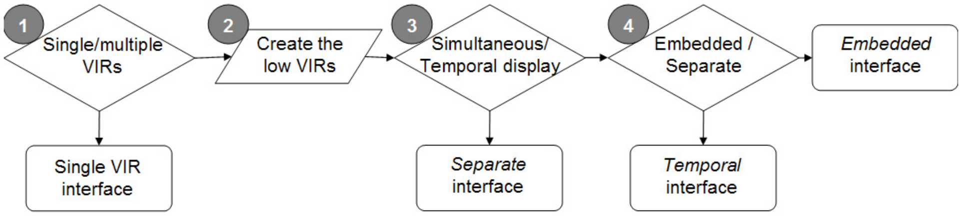

Fig. 1. Decision tree to create a multiple visual information resolution display. There are four major steps in the decision process, each covered in a section in the paper: (1/Section 4) Decide if multi-VIR is appropriate for the application; (2/Section 5) Decide on the number of resolutions, amount of data and visual information to be displayed on the low VIRs; (3/Section 6) Decide on the methods to display the multiple VIRs; (4/Section7) Decide on the spatial layout of the multiple VIRs.

how to apply this knowledge in their design. To unify our discussion, we group the interfaces into single or multiple VIR interfaces. For single-VIR interfaces, we look at the *hiVIR* interface that shows data in detail and at the highest available VIR, for example, the "detail" in overview + detail interfaces. We consider 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. Since most of the existing multiple-VIR interface studies did not explicitly consider user characteristics, we will not address this important issue in our discussion.

To better guide design processes, our paper structure is based on a decision tree to create a multiple-VIR visualization, as shown in Figure 1. Our decision tree has four major steps:

DECISION 1 (Section 4): 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. Subsection 4.1 discusses the interaction costs reported in the reviewed studies. Subsection 4.2 discusses using multiple VIRs to display single-level data.

DECISION 2 (Section 5): Create the low VIRs.
If the designer decided to use an multiple-VIR interface, the next step in the design process is to create the low VIRs, which is a challenge with large amount of data [Keim et al. 2006]. In addition to the technical challenges to provide adequate interaction speed and to fit the data onto the display device, the designer also needs to consider the appropriate levels of resolution. Study results indicate that providing too many levels of resolution may be distracting to the users, as discussed in Section 5.1. Similarly, showing too much data in the low VIRs can also be distracting, as discussed in Section 5.2. In many cases, the data has to be

abstracted and visually abbreviated to increase the display capability of the low VIRs. Section 5.3 discusses cases where the designers have gone too far in their abstraction, and the study participants could no longer use the visual information on 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 result suggest that *a priori* automatic filtering is a double-edged sword, as discussed in Section 5.4. Given all these consideration, we round up the discussion by re-examining the roles of low VIRs in Section 5.5 to help ground the low VIR design.

DECISION 3 (Section 6): Simultaneous or temporal display of the VIRs.
Once the low VIRs are created, the designer would then need 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, examples include the need for view coordination in *separate* displays, and the issue of image distortion frequently found in *embedded* displays. Study results indicate that tasks that do not absolutely require simultaneous display of data do not benefit simultaneous multiple-VIR interfaces. Section 6.1 and 6.2 consider the case when the study tasks did not require simultaneous display of VIRs, as in single-level answer or single-level information scent.

DECISION 4 (Section 7): Embedded or separate display of the VIRs.
If the choice is simultaneous display, the designer then has the 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 of the spatial layouts involves tradeoffs. For example, the *embedded* displays frequently involves distortion, as discussed in Section 7.1, and the *separate* displays involves coordination between views.

For each of these decision point, we summarize current beliefs and assumptions about multiple-VIR interface use, and list situations where study results did not clearly support our beliefs as design considerations. Recognizing that the reviewed studies are different in their implementations of the various multiple VIR techniques, their study tasks and data, and in some cases, experimental design and measurements, we do not attempt to compare between studies. Instead, we focus on pairwise-interface comparisons within each study to abstract generalizable usage patterns based on task, data characteristics, and interface differences. Since many studies looked at more than two study interfaces, some of the study results are mentioned in more than one section of the paper. We define interface effectiveness as objective measures of task time and accuracy, since these measures are reported in all user studied we sampled. For some comparisons, we were unsuccessful in abstracting general results from the studies. Section 9 discuss in detail the limitations of our review.

Since this paper aims 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 provide enough study details to illustrate our points so as to maintain readability. For reference, Appendix A briefly summarizes the study papers, listing the interfaces, tasks, data, and significant results.

2. TERMINOLOGY

In this paper, we use the term *visual information resolution* (VIR) as a measure of displayed visual information quality: displays with low VIR have comparatively less visual information than displays with high VIR. Much of the existing literature denotes these VIRs by their expected functions: for example, focus + context or overview + detail. In this paper, we name these VIRs based on their visual encodings: focus or detail can be thought of as a region of high VIR, while context or overview is of comparatively low VIR.

We further classify multiple-VIR interfaces as *temporal* or *simultaneous* based on the way they display the multiple VIRs. *Temporal* interfaces, an example being the zooming user interfaces, allow the 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 window, or 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.

3. SUMMARY OF STUDIES

Due to the large amount of user studies included in this review and the amount of important study details under consideration, we only provide a list of the study in the main body of the paper, and delay study details to an appendix. Table I lists the 19 studies reviewed in this paper, along with the test interfaces based on our categorization of *hiVIR*, *temporal*, *separate*, and *embedded*.

We include all interfaces in the user studies in this review, except for the Saraiya et al. [2005] study, since their two “Multiple View” interfaces displayed the same data in a separate view at the same VIR, but used a different graphical format. Since our review focuses on multiple VIR interfaces, we consider multiple forms of presentation to be beyond the scope of our review.

Note that Hornbæk et al.’s online document study was reported as two papers: Hornbæk and Frokjaer [2001] and Hornbæk et al. [2003].

4. 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 addition low VIRs are useful, we looked at studies that compared the single-VIR *hiVIR* interfaces to the three spatial multiple VIR interfaces: *temporal*, *embedded*, and *separate*.

It is generally believed that interfaces should provide more than one VIR (p.

Authors	**Paper Title**	**Single hiVIR**	**Multiple**		
			Temporal	**Embedded**	**Separate**
Baudisch et al. [2002]	Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming		x	x	x
Baudisch et al. [2004]	Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view	x		x	x
Bederson et al. [2004]	DateLens: A Fisheye Calendar Interface for PDAs		x	x	
Gutwin and Skopik [2003]	Fisheye Views are Good for Large Steering Tasks			x	x
Hornbæk and Frokjaer [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
Hornbæk et al. [2002]	Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview		x		x
Hornbæk and Hertzum [2007]	Untangling the Usability of Fisheye Menus		x	x	x
Jakobsen and Hornbæk [2006]	Evaluating a Fisheye View of Source Code	x		x	
Lam and Baudisch [2005]	Summary Thumbnails: Readable Overviews for Small Screen Web Browsers	x	x		
Lam et al. [2007]	Overview Use in Multiple Visual Information Resolution Interfaces	x		x	x
Nekrasovski et al. [2006]	An Evaluation of Pan and Zoom and Rubber Sheet Navigation		x	x	x
North and Shneiderman [2000]	Snap-together visualization: can users construct and operate coordinated visualizations	x			x
Pirolli et al. [2003]	The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser		x	x	
Plaisant et al. [2002]	SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation		x	x	
Plumlee and Ware [2006]	Zooming, Multiple Windows, and Visual Working Memory		x		x
Saraiya et al. [2005]	Visualization of Graphs with Associated Timeseries Data	x	x		
Schafer and Bowman [2003]	A Comparison of Traditional and Fisheye Radar View Techniques for Spatial Collaboration			x	x
Schaffer et al. [1996]	Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods		x	x	
Shi et al. [2005]	An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations		x	x	

Table I. A summary of multiple-VIR studies reviewed

307)[Card et al. 1999]. However, for the users, having the extra VIRs means more complex and difficult coordination and integration, which may be time consuming and required added mental and motor efforts. This topic is further discussed in Section 4.1.

These interaction costs may be justified if the extra VIRs are useful to the users. In general, the usefulness of the additional low VIRs hinges upon the levels of data structure that is important to the task. In other words, single-level data may not be suited for multiple-VIR display, as discussed in Section 4.2.

4.1 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 when using an interface. Better measurements of interaction costs include usage patterns, participant strategies, and interface choice. In this section, we look at the issue of interaction costs in multiple-VIR interfaces.

4.1.1 *Interaction Costs as Seen in Usage Patterns.* Five of the 19 studies reported usage patterns, constructed based on eye-tracking records [Pirolli et al. 2003; Hornbæk and Hertzum 2007] or navigation action logs [Hornbæk et al. 2002; Hornbæk et al. 2003; Jakobsen and Hornbæk 2006]. Two of them reported usability problems with their multiple-VIR interfaces [Hornbæk et al. 2002; Hornbæk and Hertzum 2007].

Hornbæk et al. [2002]'s 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 authors 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). In fact, navigation patterns show that only 55% of the 320 tasks were solved with active use of the low-VIR view (p. 380).

Hornbæk and Hertzum [2007]'s study on fisheye menus reported large navigation costs in their *separate* and *embedded* interfaces, all interfaces involved some variant of focus locking [Bederson 2000]. Even though these interfaces succeeded in facilitating quick, coarse navigation to the target, participants had difficulty in 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.

4.1.2 *Interaction Costs as Seen in Participant Strategies.* Two of the 19 studies reported participant strategies in interface use [Baudisch et al. 2002; Lam et al. 2007].

In Baudisch et al. [2002]'s 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 the required magnification for the remainder of the task without zooming back to the low-VIR view, thus effectively using the *temporal* interface as a *hiVIR* interface.

In Lam et al. [2007]'s study, their 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 y-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 the participants, however, developed a strategy to enable the use 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, the 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.

4.1.3 *Interaction Costs as Seen in Participants' Interface Choice.* Another indicator of interaction costs is participants' active choice to use only one VIR of a multiple-VIR interface to avoid coordinating between the multiple VIRs. In two of the 19 studies, participants could convert the multiple-VIR interface into a single-VIR interface [Hornbæk and Frokjaer 2001; Hornbæk et al. 2003; Lam et al. 2007], and in Hornbæk et al. [2002]'s study on map navigation, the researcher recorded active pane use.

In a study on reading electronic documents, the participants could expand all the document sections at once by selecting the pop-up menu item "expand all" in the *embedded* interface [Hornbæk and Frokjaer 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. [2002]'s study on map navigation, 45% of the participants did *not* actively use the low-VIR view in the *separate* interface, even though 80% of the participants reported preference for having the extra low-VIR view. In Lam et al. [2007]'s study on visual searching and comparing of line graphs, the participants could expand all the initially compressed graphs in their *embedded* or the *separate* interface by a key press, and they 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, the 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, for example with eye-tracking. 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].

4.2 Consideration 2: Single-Level Task-Relevant Data May Not Be Suited for Multiple-VIR Displays

The number of VIRs provided by the interface should reflect the levels of organization in the data as required by the task. Otherwise, the users may need to pay the cost of coordinating between different VIRs without the benefit 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 [Baudisch et al. 2004; Hornbæk et al. 2002; Hornbæk et al. 2003; Lam and Baudisch 2005; Lam et al. 2007]. Two show the lack of benefit of using multiple-VIR interface for single-level data when the low VIR did not contain enough information for the task [Baudisch et al. 2004; Lam et al. 2007], while Hornbæk et al. [2002]'s study shows adverse effects in using multiple-VIR interfaces for single-level data. Hornbæk et al. [2003]'s study on online documents illustrates how task nature affects the levels of data required, and consequently, interface use. We excluded Lam and Baudisch [2005]'s study in this discussion as their *hiVIR* interface was almost nine times the size of their multiple-VIR interfaces, making direct comparisons difficult.

Baudisch et al. [2004]'s study on information searches shows the lack of benefit of 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, but surrounding text could be too small to read. 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 view concentrated the relevant information for the task in a smaller space. In contrast, when the task required reading text around these keywords, as in the *Analysis* task, having an extra low-VIR view did not result in performance benefits.

The situation is similar in Lam et al. [2007]'s study on visual-target search on 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 an extra low VIR did not seem to enhance participant performance.

Hornbæk et al. [2002]'s study on map navigation illustrates the adverse effects of displaying single-level data using a multiple-VIR interface. Despite having 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 researcher explained these differences by differences in content and the number of levels of organization of the two maps: the Washington map had three levels of county, city and landmark, while the Montana map was single-leveled. As a result, the Montana map had weak navigation cues at low zoom levels, and is arguably less suited for the multi-level zoomable *temporal* interface than the Washington map.

Hornbaek et al.'s online document study illustrates how task nature can affect the levels of data required on an interface [Hornbæk and Frokjaer 2001; Hornbæk et al. 2003]. In their question-answering task, the participants were slower if they were given the additional low-VIR overview without being more accurate in their answers. Based on reading patterns, Hornbaek and Frokjaer suggested that the slower reading times was due to the attention-grabbing low-VIR view in the *separate* interface, which led the participants to further explore the documents perhaps unnecessarily. In contrast, in the essay-writing task, the section and subsection headers on the low-VIR overview resulted in better quality essays without any time penalty when compared to the *hiVIR* interface.

5. DECISION 2: HOW TO CREATE THE LOW VIRS?

Creating the low VIR in a multiple-VIR display 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 distractors. In Section 5.1, we discuss the adverse effect of displaying more levels of VIR than those supported by the data and required by the task. Section 5.2 discuss 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 encoding or reduce the amount of data on the low-VIR display. Section 5.3 discusses cases where the researchers have gone too far in their abstraction as their study participants could no longer use the visual information on the low VIRs. Section 5.4 looks at the tradeoffs in using *a priori* automatic filtering to select low-VIR items.

Given all these consideration, we round up the discussion in Section 5.5 by re-examining the roles of low VIRs to help ground low- VIR designs. Study results suggests a more limited set of roles low-VIR, or context, plays in multiple-VIR interface than is believed. While we find study results support the use of context as navigational shortcuts to move within the data (5.5.1) and to provide overall data structure (5.5.2), we fail to find support to the common beliefs of using context to aid orientation (5.5.3) and to provide meaning (5.5.4) to interpret data in *hiVIR*.

5.1 Consideration 1: Having Too Many Visual Resolutions Can Hinder Performance

In general, the number of visual resolutions supported by the interface should reflect the levels of organization in the data. Otherwise, the 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 can be distracting, which may at best be ignored, and at worst, 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:

(1) Baudisch et al. [2002] added a low-VIR view to the *temporal* zoom plus pan (z+p) display in their overview plus detail (o+d) interface
(2) Hornbæk et al. [2002] added a low-VIR view to the *temporal* zoomable interface
(3) Hornbæk and Hertzum [2007] added a low-VIR view to the *embedded* fisheye menu
(4) Nekrasovski et al. [2006] added low-VIR views to their *temporal* Pan&Zoom and their *embedded* Rubber Sheet Navigation interfaces.

Since Hornbæk and Hertzum [2007]'s study did not include an interface that is only *embedded*, we cannot comment on the effect of having an additional low-VIR view. For the other three studies, perhaps because the multiple-VIR interfaces already displayed all the meaningful visual information supported by the data and required by the tasks, having the additional low-VIR view did not enhance or even degrade performance.

In terms of performance, participants in Baudisch et al. [2002]'s study obtained similar results using the overview plus detail (o+d) interface when compared to their zoom plus pan (z+p) interface. The researchers reported that their participants kept the detailed view zoomed to 100% magnification for tracing, thus effectively reduced the *temporal* component of the interface to a single-VIR display.

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

Hornbæk et al. [2002]'s study on map navigation suggests the performance hinderance of having irrelevant levels of resolution in the interface. One of their study interfaces is 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 not being compensated as the *temporal* interface may have reduced, or even eliminated, the need for a separate overview (p. 381).

In some cases, study results indirectly suggest the adverse effects on performance when the interfaces provided irrelevant VIRs. For example, in Plumlee and Ware [2006]'s study that required matching clusters of 3D objects, their *temporal* interface had many magnification levels that neither helped the participants to locate the candidate objects due to the textured background, nor were detailed enough to view the individual components of the objects for visual matching. Given the participants needed to keep track of the components in various objects in the task in their short-term memory when using the *temporal* interface, the extra zooming levels may render the tasks harder. This extra cognitive load may explain the relatively small number of items the participants could handle before the opponent *separate* interface became more appropriate for the task, when compared to the results found by Saraiya et al. [2005]. Similarly, in Baudisch et al. [2002]'s study on static visual path-finding tasks and dynamic obstacle-avoidance tasks, their *temporal* interface and their *separate* interface seemed to support more VIRs than their *embedded* interface, which had two different VIRs only. While the special setup in their *embedded* interface undoubtedly contributed to the superior performance of their participants when using the interface, one wonders if the extra VIRs may have distracted the participants in the other two interface trials.

5.2 Consideration 2: Having Too Much Information on the Low-VIR Display Can Hinder Performance

While it may be tempting to provide more, rather than less, information in the low-VIR display, study results suggest that the extra information may harm task performance. None of the 19 studies we reviewed looked at low-VIR item density as a factor. However, we can obtain indirect evidence by comparing between multiple-VIR interfaces that display different amount of visual information in their low-VIR displays, and by comparing between low- and high-VIR displays for visual search tasks with answers that were apparent from the low-VIR display.

Fifteen of the studies included at least two multiple-VIR interfaces [Baudisch et al. 2002; Baudisch et al. 2004; Bederson et al. 2004; Gutwin and Skopik 2003; Hornbæk and Frokjaer 2001; Hornbæk et al. 2003; Hornbæk et al. 2002; Hornbæk and Hertzum 2007; Lam et al. 2007; Nekrasovski et al. 2006; Plaisant et al. 2002; Plumlee and Ware 2006; Pirolli et al. 2003; Schafer and Bowman 2003; Schaffer et al. 1996; Shi et al. 2005]. Of the 15 studies, 10 showed similar amount of information for the low-VIR displays for their multiple-VIR interfaces [Baudisch et al. 2002; Baudisch et al. 2004; Bederson et al. 2004; Hornbæk et al. 2002; Gutwin and Skopik 2003; Lam et al. 2007; Nekrasovski et al. 2006; Schafer and Bowman 2003; Schaffer et al. 1996; Shi et al. 2005]. We excluded Hornbaek's electronic document study since the low-VIR displays showed different kinds of information [Hornbæk and Frokjaer 2001; Hornbæk et al. 2003]. We also excluded Plaisant et al. [2002]'s study since it was unclear from the paper the number of items initially showed in their *embedded* SpaceTree interface. The remaining two studies displayed similar kind of information, but at different amounts, in their low-VIR displays:

(1) Pirolli et al. [2003]'s study compared between the *separate* file browser and the *embedded* hyperbolic tree browser. While the paper did not explicitly compare the display capacities of the low-VIRs of the two interfaces, we can estimate based on the figures shown in the paper. The low-VIR view of the *separate* file browser could only display 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 [2007]'s study looked at the *temporal* cascading menu and compared it to two *embedded* menu designs based on the fisheye menu. While their *temporal* cascading menu only showed a list of alphabets at the lowest VIR, their *embedded* fisheye menus showed all menu items in various font sizes depending on the visual resolution.

In both of these cases, the researcher 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) based on the complex ways with which visual attention and visual search interact. In fact, their study shows that crowding the display with information that are not required for the task may even be detrimental, as in the case with their *embedded* hyperbolic tree browser interface, which led to slower performance time when compared to their *temporal* file browser under low information scent, possibly because their *embedded* interface displayed irrelevant information that were distracting.

Hornbæk and Hertzum [2007] came to a similar conclusion in their study on displaying large menus: "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). Their *temporal* cascading menu elided information and only displayed parts of the space, but yet, the interface outperformed all of their *separate* and *embedded* interfaces included in their study.

This situation is analogous to tasks where the answers are apparent from the low-VIR display, and the *hiVIR* interface therefore showed irrelevant information. Seven of the reviewed studies included both *hiVIR* and a multiple-VIR interface [Baudisch et al. 2004; Hornbæk and Frokjaer 2001; Hornbæk et al. 2003; Jakobsen and Hornbæk 2006; Lam and Baudisch 2005; Lam et al. 2007; North and Shneiderman 2000; Saraiya et al. 2005]. Five of them studied tasks that could be answered based solely on the *lowVIR* interface alone [Baudisch et al. 2004; Lam and Baudisch 2005; Lam et al. 2007; North and Shneiderman 2000; Saraiya et al. 2005].

In Baudisch et al. [2004]'s study on information searches on web documents, their *Outdated* task required their 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 were irrelevant to the *Outdated* task. In contrast, their *separate* and their *embedded* interfaces concentrated the relevant semantic highlights in their low-VIR sections, and allowed participants to spot and determine if the web document contained all the keywords. As a result, both of the multiple-VIR interfaces tested in the study outperformed their *hiVIR* interface for this task.

In Lam and Baudisch [2005]'s study looking at information search on webpages, their PDA-sized *temporal* interfaces supported equal performance as their desktop counterpart, even though the *hiVIR* interface had nine times more display space and displayed completely readable information. The authors suggested that the extra information on the desktop display may have distracted the participants and caused unnecessary searching and reading.

In Lam et al. [2007]'s study on visual target search among a large line-graph collection, one of their tasks involved finding the highest point in the data, a task where their *loVIR* interface alone was found to be adequate, and not surprisingly, most suited for the task. In North and Shneiderman [2000]'s study on visual information search in census data, interfaces that were equipped with a low-VIR view 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.

5.3 Consideration 3: Information Presence Is Not Adequate; It Has To Be Perceivable

The mere presence of the information on the screen is not sufficient; the information needs to be perceivable to be usable. For text, visual information on the low-VIR display may need to be readable. Of the 19 studies reviewed, seven looked at text data. Five studies included unreadable text in their interfaces [Baudisch et al. 2004; Bederson et al. 2004; Hornbæk and Hertzum 2007; Jakobsen and Hornbæk 2006; Lam and Baudisch 2005], while two had only readable text [Hornbæk et al. 2003; North and Shneiderman 2000]. We excluded Bederson et al. [2004]'s 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.

Studies show that when the unreadable texts on the low-VIR display were the targets of the visual search, the unreadable text provided inadequate information and rendered the low-VIR display superfluous, as the single *hiVIR* display resulted in similar performance measures despite displaying the information in a larger screen area and thus, having a larger search space.

In Baudisch et al. [2004]'s 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 any performance benefits when compared to the traditional *hiVIR* browser for the task.

Another example is Hornbæk and Hertzum [2007]'s study on displaying of large numbers of menu items. One of their *embedded* interfaces, the *Multi-focus*, provided readable menu items in the low-VIR regions based on *a priori* significance, while the other *embedded* interface implemented [Bederson 2000]'s Fisheye Menu, and displayed unreadable items at the extreme ends in the low-VIR regions. Even though the study failed to find differences between the two *embedded* interfaces in terms of performance, satisfaction ratings, or subjective preference, eye-tracking results suggested that participants used the low-VIR regions more frequently in the *Multi-focus embedded* interface trials, leading the researcher to question the use of screen space in providing unreadable low-VIR regions as being beneficial (p. 26).

Jakobsen and Hornbæk [2006]'s study looked at displaying program code using an *embedded* fisheye interface, which displayed unreadable text in the low-VIR regions to show code structural features. Nonetheless, the *embedded* interface did not offer performance benefits over the *hiVIR* interface in a task that involved counting conditional and loop statements. Rather, participants spent more time using *embedded* interface to find the closing brace of a loop control structure, which was not visible in the low-VIR regions. The authors thus suggested that interfaces should display readable text to allow direct use of the information in the low-VIR view (p. 385).

Lam and Baudisch [2005]'s study reported similar findings. Their *temporal* Thumbnail interface had unreadable text as the low-VIR view, but their *temporal* Summary Thumbnail contained only readable text. They found that participants using the Thumbnail interface had 2.5 times more zooming events, and when zoomed in, horizontally scrolled almost four times more, suggesting the ineffectiveness of the unreadable text.

For graphical visual signals, two studies reported the effects of showing insufficient details on the low-VIR display [Hornbæk et al. 2002; Lam et al. 2007]. In Hornbæk et al. [2002]'s 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. Washington map trials with an extra low-VIR overview had slower performance times and worse recall accuracy, suggesting the burden of having an extra low-VIR view as "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, leading 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 the participants actively used the low-VIR view.

Lam et al. [2007] qualified the perceptual requirements for their low-VIR display as visual complexity and visual span. Their study looked at displaying a large collection of line graphs for visual search and visual compare tasks. They 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 that could be easily found. In contrast, the visual signals in their *Shape-matching* task were complex, as they were composed of three peaks, and were less discernable in the low-VIR views. As a result, many participants resorted to viewing the signals in the high-VIR view instead.

5.4 Consideration 4: *A priori* Automatic Filtering Is a Double-Edged Sword

To create the low-VIR display, designers often need to select a subset of the data to display. One approach is based on Furnas [1986]'s Degree of Interest function, using *a priori* knowledge of information relevance and relative distance to the focal point. Jakobsen and Hornbæk [2006] further differentiated the distance term in the function into semantic and syntactic distances to implement an *embedded* interface for source codes. Doing so can result in enhanced task performance as the low-VIR displays concentrate useful information and reduce distractors. However, users may be confused by the selective filtering and becomes disoriented.

Instead of seeing filtering as a workaround to the display-size challenge and as a liability, there are evidences to suggest filtering in itself can enhance task performance. When the information selected to be displayed in the low-VIR view is directly relevant for the task, such intelligence avoids tedious manual searching and navigation in the high-VIR view, and possibly also avoided distractions by irrelevant information. For example in Jakobsen and Hornbæk [2006]'s 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 the participants were required to search for information contained in the function declarations throughout the entire source code.

Instead of removing the data, the interface can visually highlight relevant objects and limits the search space to the highlighted objects. In Baudisch et al. [2004]'s study on information searches on webpages, their interfaces semantically highlighted and preserved the readability of keywords relevant to the tasks. Even though the semantic highlights were present in all of their interfaces, the ones that concentrated the keywords on a single screen exhibited performance benefits, as long as the interface still provided adequate layout information that was required for the task. 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. Skopik and Gutwin [2005]'s study used landmarks called visit wear to facilitate refinding in a highly-linked graph. The visit wear interface highlighted previously visited nodes based on the amount of visits and the time elapsed since the last visit. The study found that the presence of visit wear improved their participants' ability to find items and locations previously visited, as the highlighting reduced the need for visual memory.

However, automatic filtering is a double-edged sword, as filtering may result in disorientation and distrust of the automatic selection algorithm. In Hornbæk and Frokjaer [2001] and Hornbæk et al. [2003]'s study on reading electronic documents, their *embedded* interface preserved readability only for the most important part of the document, with content importance determined by the interface *a priori*. The participants expressed distrust, both in their satisfaction feedback where they rated the *embedded* interface as confusing, and in their comments indicating 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. 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 the users to understand and trust the filtering algorithm that uses only syntactic distance. Also, since the context information is updated when the focal point changes, it may be more confusing to navigate when the context information changes semantically, as the pointer navigation is conceptually geometric rather than semantic. In Jakobsen and Hornbæk [2006]'s study in displaying program source codes, the embedded low-VIR regions replaced scrolling in the *hiVIR* interface, and only displayed semantically relevant parts of the source code based on the lines displayed in the focal region. Participants were confused about the semantic relationship 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 how the users view the entire dataset. In Hornbæk and Frokjaer [2001] and 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.

5.5 Consideration 5: The Roles of the Low-VIR Display May Be More Limited Than Are Believed

While the high-VIR display enables the users to perform detail work, it is considerably more difficult to pinpoint the use of the low-VIR display in multiple-VIR displays. We therefore look at proposed uses of the low-VIR display based published literature, and find that study results support the claim that low-VIR provides shortcuts for navigation (5.5.1) and overall structures (5.5.2). We have not been able to find strong support for other proposed roles of the low-VIR display, including aiding orientation (5.5.3) and providing meaning (5.5.4).

5.5.1 *Supported: Context provides shortcuts for navigation.* 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]. The role to coordinate between the low- and the high-VIR views enables the use of context to directly select targets for detail exploration.

For example, the low-VIR view of a list of geographic states included the census data acted as hyperlinks for the detail, high-VIR view [North and Shneiderman 2000]. Another way context 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. 2003]'s online document study, showing section and subsection headers. For graphical displays, Baudisch et al. [2002]'s study found that participants used the low-VIR overview to navigate to targets and performed the detail work in the *hiVIR* display.

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

5.5.2 *Supported: Context provides overall structure.* Context can provide a structure of the entire data that may not be apparent in higher VIRs. For example, in Hornbæk and Frokjaer [2001] and Hornbæk et al. [2003]'s study on reading electronic documents, context provided by the low-VIR view in their *separate* interface showed section and subsection headers in the documents. "The overview pane may indirectly have helped subjects to organize and recall text" (p.144), and led to higher quality essay than produced by the participants using the *separate* interface without any time penalty.

5.5.3 *Partial support: Context 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 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].

Study results suggest that the low-VIR display is only useful for orientation when the data itself contains poor visual cues. In [Hornbæk et al. 2002]'s study on map navigation, the multi-leveled Washington map contained rich visual cues for navigation, and participants did not seem to require an extra overview when navigating within the Washington map. In fact, they were faster in the navigation tasks when using the no-overview interface with the Washington map. In contrast, the participants using single-leveled Montana map made a smaller number of scale changes when the low-VIR display was present, suggesting that the low-VIR overview was used an navigation aid that helped reduced the need for zooming.

5.5.4 *Open: Context provides 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]. While we have not found studies that explicitly measure or validate using the low-VIR display for comparative interpretation, one of the reviewed papers offer some insights.

In Saraiya et al. [2005]'s study on displaying time-series data as nodes in a graph, their *hiVIR* interface showed all ten time points simultaneously, while the *temporal* interface showed one data point at a time. Even though participants had more errors overall when using the *hiVIR* interface, suggesting having surrounding entities may be detrimental rather than helpful, a closer look at individual tasks showed mixed results. We focus on tasks that involved all time points as they are more likely to involve comparative interpretations. The study reported that participants using the *temporal* interface were faster in finding the topology trend of a larger graph and in searching for outlier time points. These two results suggest that despite having to identify trends or detect outliers, context provided in the *hiVIR* interface was detrimental rather than beneficial, possibly due to visual cluttering. On the other hand, the 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. It is therefore difficult to draw broad conclusions about the effectiveness of context from this study.

6. DECISION 3: SIMULTANEOUS, OR TEMPORAL, DISPLAYS OF THE MULTIPLE VIRS

The third decision in the process of creating a multiple-VIR interface is to decide how to arrange the different VIRs. For the designer, it is a choice between showing them simultaneously or showing them one at a time as in zooming techniques.

A well-known problem with zooming is that when the user zooms on a focus, all contextual information is lost. Such a loss of context can become a considerable usability obstacle, as the users would need to integrate all these information over time, requiring memory to keep track of the temporal sequence, and their orientation within that sequence [Herman et al. 2000][Furnas 2006]. To alleviate these problems, a set of techniques were developed, collectively called focus+context, that allow the user to focus on some detail without losing the context. 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 provide support for the superiority of simultaneous display of the different VIRs: study results suggest that the *temporal* interface is surprisingly good in supporting most tasks. Based on study results, we identified two situations where the simultaneous display of multiple-VIR interfaces demonstrated performance benefits: when the answer to the problem involved information from ***all*** the VIRs available in the interface (6.1), and when the different VIRs provided clues for the answer (6.2). Otherwise, temporal switching seemed adequate.

6.1 Consideration 1: Tasks with Single-Level Answers May Not Benefit from the Simultaneous Display of Different Visual Resolutions

In general, we found that simultaneous display of multiple VIRs is best suited for tasks that required answers that spanned multiple levels in the data. We focus on ten of the 19 studies as they included a *temporal* and at least one simultaneous-display interface as a basis for comparison [Baudisch et al. 2002; Bederson et al. 2004; Hornbæk et al. 2002; Hornbæk and Hertzum 2007; Nekrasovski et al. 2006; Plaisant et al. 2002; Pirolli et al. 2003; Plumlee and Ware 2006; Schaffer et al. 1996; Shi et al. 2005]. We excluded Hornbæk et al. [2002]'s 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 ten studies had task questions that required multi-level answers: [Bederson et al. 2004; Plaisant et al. 2002; Schaffer et al. 1996], and all showed performance benefits in using their simultaneous-display interfaces for those tasks than with their *temporal* interfaces.

In Schaffer et al. [1996]'s re-routing task, the participants were required to find an alternative route to connect the two endpoints in the break, and the route spanned multiple 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. In Plaisant et al. [2002]'s tree browsing study, the SpaceTree *embedded* interface 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 Bederson et al. [2004]'s 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.

On the other hand, the *temporal* interface seemed adequate for tasks with single-level answers, unless the clues required to reach the answers were also multi-leveled.

6.2 Consideration 2: Tasks with Single-Level Information Scent May Not Benefit from the Simultaneous Display of Different Visual Resolutions

In cases where the answer to the task did not span multiple levels of VIRs, simultaneous display of multiple VIRs can still be helpful if the clues to the final answer spanned multiple levels. Of the nine studies that included a *temporal* display and and at least one simultaneous-display interface [Baudisch et al. 2002; Bederson et al. 2004; Hornbæk and Hertzum 2007; Nekrasovski et al. 2006; Pirolli et al. 2003; Plaisant et al. 2002; Plumlee and Ware 2006; Schaffer et al. 1996; Shi et al. 2005], five provided multi-level clues to single-level answers [Baudisch et al. 2002; Hornbæk and Hertzum 2007; Pirolli et al. 2003; Plaisant et al. 2002; Plumlee and Ware 2006], and all five except the Hornbæk and Hertzum [2007] study demonstrate benefits in using simultaneous-VIR display.

In Baudisch et al. [2002]'s 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, the participants could see more of the hierarchical structure in a single view, and traverse tree levels faster. Under high-scent conditions, this feature could be advantageous. In contrast, under low information scent conditions, the 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 the 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 the 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 [2006]'s study, the task required matching complex clusters of 3D objects, and the 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 the 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 [2007]'s 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 various interface, it is difficult to isolate the actual time used to find the items and determine the relative effectiveness of the visual displays.

Taking Considerations 1 and 2 together, we believe that tasks with single-level answers and single-level clues would not benefit from simultaneous display of the different visual resolutions. Indeed, that seems to be the case based on study results, even when the tasks required comparison between objects. As long as the participants can keep track of the information in their short-term memory, the *temporal* interface seemed adequate, and at times, even resulted in better performance and participant feedback.

Of the nine studies that included a *temporal* display and at least a multiple-VIR interface [Baudisch et al. 2002; Bederson et al. 2004; Hornbæk and Hertzum 2007; Nekrasovski et al. 2006; Pirolli et al. 2003; Plaisant et al. 2002; Plumlee and Ware 2006; Schaffer et al. 1996; Shi et al. 2005], six had tasks that required single-level answers with single-level clues [Bederson et al. 2004; Hornbæk and Hertzum

2007; Nekrasovski et al. 2006; Plumlee and Ware 2006; Schaffer et al. 1996; Shi et al. 2005]. All except the Shi et al. [2005]'s study results supported this general conclusion.

In Schaffer et al. [1996]'s study, even though the *embedded* interface supported faster task completion times than *temporal* when rerouting in a hierarchical network, the participants did not seem to need the simultaneous display of the VIRs to locate the broken link at the lowest level in the network, as indicated by the lack of performance differences between the trials using the *temporal* and the *embedded* interface for this task. Bederson et al. [2004]'s study showed that the *temporal* Pocket PC was more appropriate for simple calendar tasks that involved checking the start dates of pre-scheduled activities, and tasks that spanned short-time periods. In Nekrasovski et al. [2006]'s study, where the task was to compare the 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 Hornbæk and Hertzum [2007]'s visual searches in menus, their *temporal* interface was the traditional cascading menu. The study found that known-item search tasks were solved faster and more accurately with the *temporal* interface than with the various *embedded* and *separate* interfaces.

The exception is Shi et al. [2005]'s study, where the researchers found that their interface which embedded multiple levels resulted in faster task times than the *temporal* interface. In Shi et al. [2005]'s case, there may be a speed-accuracy tradeoff: the researcher 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.

7. 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 display of the VIRs. At the highest level, the interface can integrate the different VIRs within the same window to create an *embedded* display, or show the different VIRs in *separate* views. Proponents of *embedded* 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" (p. 21) [Pirolli et al. 2003]. 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 the 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 involve distortion, an issue we will discuss in Section 7.1.

The choice between these two spatial arrangements is unclear based on empirical study results. Eight of the 19 studies included both *embedded* and *separate* interfaces. We find it difficult to directly compare between the two simultaneous display directly in three of the studies [Baudisch et al. 2004; Hornbæk et al. 2003; Nekrasovski et al. 2006]. For the remaining five studies, two did not find significant performance differences [Lam et al. 2007; Schafer and Bowman 2003]. Only Baudisch et al. [2002] and Gutwin and Fedak [2004]'s study demonstrated superior performance support of the *embedded* interface. In the case of Baudisch et al. [2002] the performance differences is possibly due to the unique implementation of their interface, while Gutwin and Fedak [2004]'s results are possibly due to comparatively complex interactions required in their *separate* interfaces.

Our mixed result may reflect the different tradeoffs in these interface. Also in some cases, the benefit of providing multiple VIRs may be so large that the spatial arrangement may not matter (p.12) [Tory et al. 2006].

Of the three studies that we consider incomparable, two of them are incomparable due to intentional implementation differences based on common perceived use of the two spatial arrangements: the *separate* interface provides the low-VIR as an overview of the data, while the *embedded* interface shows background and supporting information in the low-VIR regions. Also, the perceived functions of the two interfaces generally colored the 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; Shi et al. 2005]), and spatial navigation for *separate* displays (e.g., [North and Shneiderman 2000; Plumlee and Ware 2006]). As a result, the issue of spatial arrangement is confounded.

For example, Hornbæk et al.'s study showed different kinds of information in their two multiple-VIR interfaces [Hornbæk and Frokjaer 2001; Hornbæk et al. 2003]. The low-VIR view of their *separate* interface provided document section and subsection headers which was optimal for showing overall structures in text documents and encouraged more detail explorations. In contrast, their *embedded* interface showed *a priori* determined significant text in relation to the focal area, which promoted rapid document reading at the cost of accuracy. The second study is Baudisch et al. [2004]'s study on web document search on browsers, where their *embedded* interface was designed to favour row discrimination, and their *separate* interface was designed for column discrimination.

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. [2006]'s 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, their *temporal+separate*, nor their *embedded+separate* could be directly compared with their *embedded* interface.

In the five cases where direct comparison was possible, three studies did not find performance differences between the two interfaces [Hornbæk and Hertzum 2007; Lam et al. 2007; Schafer and Bowman 2003]. The two exceptions are Baudisch et al. [2002] and Gutwin and Fedak [2004]'s studies.

Even though Gutwin and Fedak [2004]'s study on steering tasks showed significant results, we believe their results may be confounded by the relatively complex interaction required in their *separate* interfaces. The study included three *embedded* fisheye displays and two *separate* displays. In a series of 2-dimensional steering tasks where participants were required to move a pointer along a path that is defined by the objects in a visual workspace, the study found that the *embedded* interfaces supported better time and accuracy performance over the *separate* interface at all magnification. The authors thus concluded that "the fact that fisheyes show the entire steering task in one window clearly benefited performance"(p. 207).

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

We also find the the interaction complexity differ greatly among the 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. It 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-resolution view. In contrast, the *embedded* interfaces required only a single mouse action, where mouse move would shift the focal point and magnify the underlying path. This type of interaction, however, has the disadvantage of magnification-motion effect, where the objects in the magnifier appear to move in the opposite direction to the motion of the lens, and was easier to overshoot the motion and slip off the side of the lens. Given the complex interplay of at least three factors that seem to be implementation specific, it is difficult for us to draw broad conclusions based on this study.

In Baudisch et al. [2002]'s study looked at three tasks that required information from all the VIRs: a static route-finding, a static connection-verification, and a dynamic obstacle-avoidance task. The results indicated that the *embedded* interface better supported all of the tasks, and the participants preferred their *embedded* interface. Their unique *embedded* interface may explain why their study found performance differences between the two interfaces. The *embedded* interface avoided many of the usability pitfalls in embedding high-VIR region into low-VIR displays: 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. Second, distortion was not used in the system. 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.

We therefore 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 on a single study.

7.1 The Issue of Distortion

One of the potential costs in embedding multiple VIRs within the same interface is distortion. Based on Furnas [1986]'s 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 when the layout is sparse [Carpendale et al. 1997]. Even when the users recognize the distortion, distance and angle estimations may be more difficult and inaccurate when the space is distorted [Carpendale et al. 1997]. When the distortion is less severe, as in bifocal or modified fisheye distortions, one study reported that distance and angle estimations were not adversely affected by the distortion [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 measure the effects of distortion directly. Lau et al. [2004] found that 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, Lam et al. [2006] found image recognition took longer and were less accurate at high fisheye transformation levels. [Skopik and Gutwin 2005] reported that a time, but not accuracy, penalty on refinding nodes in a highly-linked graph when the graph was transformed by a polar fisheye transformation.

It is difficult to tease out the effects of distortion based on the 19 papers we reviewed here, since none of the studies specifically look at distortion as a factor. We can therefore only rely on observations reported in the paper to obtain insights. Fourteen studies included an *embedded* interface [Baudisch et al. 2002; Baudisch et al. 2004; Bederson et al. 2004; Gutwin and Skopik 2003; Hornbæk and Frokjaer 2001; Jakobsen and Hornbæk 2006; Hornbæk and Hertzum 2007; Lam et al. 2007; Nekrasovski et al. 2006; Plaisant et al. 2002; Pirolli et al. 2003; Schafer and Bowman 2003; Schaffer et al. 1996; Shi et al. 2005], and 12 implemented distortion. The two exceptions are Baudisch 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 distinct visual forms to represent the same data in different 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 [Baudisch et al. 2004; Bederson et al. 2004; Gutwin and

Skopik 2003; Hornbæk and Frokjaer 2001; Jakobsen and Hornbæk 2006; Schaffer et al. 1996; Shi et al. 2005]. We excluded Gutwin and Skopik [2003] in our 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.

For the other six studies, study results suggest that constrained and predictable distortion is well tolerated, as all their *embedded* interfaces involve either text ([Baudisch et al. 2004; Hornbæk and Frokjaer 2001; Jakobsen and Hornbæk 2006]) or grid-based ([Bederson et al. 2004; Schaffer et al. 1996; Shi et al. 2005]) distortions.

Five studies reported problems attributed to distortion, and all involved more drastic and elastic distortion techniques [Hornbæk and Hertzum 2007; Nekrasovski et al. 2006; Plaisant et al. 2002; Pirolli et al. 2003; Schafer and Bowman 2003]. We 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 focus our discussion on the remaining four studies to further understand the costs of distortion.

Schafer and Bowman [2003]'s *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, the users may not notice the distortion as it may not be caused by their direct action since the task was collaborative.

Nekrasovski et al. [2006]'s *embedded* interface implements Rubber-Sheet Navigation that allows users to stretch or squish rectilinear focus areas as though the dataset 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. [2002]'s 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. [2003]'s study also compared between a *temporal* file browser and the *embedded* hyperbolic tree browser. The researcher 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.

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

8. SUMMARY: DESIGN RECOMMENDATIONS

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

8.1 Recommendation: 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 (p. 1003) [Furnas 2006].

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

8.2 Recommendation: Provide Relevant, Sufficient and Necessary Information in the Low-VIR Displays to Support Context Use

While the content of high VIRs should support detail work demanded by the tasks at hand, study results suggested that context in the low VIRs can be used in two ways: in navigation where it provided a short-cut to jump to different parts of the data and in mental data organization if it provides overall structures of the data. To be effective, there should be sufficient, relevant and necessary information in the displays to support the task at hand. Otherwise, the value of the display may not be sufficient to overcome the costs of having extra visual resolutions. The amount of detail required in context is likely more than what most of us assumed, judging from the number of ineffective interface contexts created for these studies. In the case of 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, [Hornbæk and Hertzum 2007]). For graphical displays, studies on visual search (e.g., [Tullis 1985]) and Lam et al. [2007]’s study provide guidelines, for example, the visual signals should be simple and of narrow visual spans.

8.3 Recommendation: Display the VIRs Simultaneously for Multi-Leveled Answers, or Clues to These Answers, that Cannot be Kept in Short-Term Memory

Selecting the correct visualization techniques to display data is important due to the inherent tradeoffs in the *temporal*, *separate* and *embedded* displays. While most *temporal* implementations offered familiar panning and zooming interactions, these interfaces required the users to keep information in their short-term memories. Simultaneous displays of multiple VIRs, on the other hand, often resulted in the need to coordinate different VIRs, and more complex and unfamiliar interactions. Based on study results, we conclude 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 and with enough detail for the task. Otherwise, the *temporal* interface seemed to be more suitable.

8.4 Open: When Should Multiple VIRs be Displayed Simultaneously?

Unfortunately, we are not able to suggest guidelines in displaying multiple VIRs simultaneously due to the difficulties in obtaining direct interface comparisons based on our set of study papers.

9. LIMITATIONS OF OUR REVIEW AND ANALYSIS

While we attempt to provide a comprehensive, study-based, review in the use and design of multiple-VIR interfaces, we are necessarily limited by our own knowledge and time to include all the relevant studies in our review. In order to provide a reasonably concise review, we excluded studies where their results did not differentiate between the study interfaces in terms of performance measures or usage patterns.

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.

Acknowledgements

We appreciate many discussions with Ted Kirkpatrick on paper structure, and we thank François Guimbretière, Diane Tang, and John Dill for their feedback on paper drafts.

REFERENCES

Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. 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)*. 259–266.

Baudisch, P., Lee, B., and Hanna, L. 2004. Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. In *Proc. ACM Advanced Visual Interface (AVI'04)*. 133–140.

Baudisch, P. and Rosenholtz, R. 2003. Halo: A technique for visualizing off-screen locations. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*. 481–488.

Bederson, B. 2000. Fisheye menus. In *Proc. ACM SIGCHI Symposium on User interface software and technology (UIST'00)*. 217–226.

Bederson, B., Clamage, A., Czerwinski, M. P., and Robertson, G. G. 2004. Datelens: A fisheye calendar interface for pdas. *ACM Trans. on Computer-Human Interaction (ToCHI) 11,* 1 (Mar.), 90–119.

Card, S. K., Mackinlay, J. D., and Shneiderman, B. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, California.

Carpendale, M. S. T., Cowperthwaite, D. J., and Fracchia, F. D. 1997. Making distortions comprehensible. In *Proc. IEEE Symposium on Visual Languages*. 36–45.

Furnas, G. W. 1986. Generalized fisheye views. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'86)*. 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)*. 999–1008.

Gutwin, C. and Fedak, C. 2004. A comparison of fisheye lenses for interactive layout tasks. In *Proc. Conf. on Graphical Interface (GI'04)*. 213–220.

Gutwin, C. and Skopik, A. 2003. Fisheye views are good for large steering tasks. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*. 201–208.

Herman, I., Melancon, G., and Marshall, M. S. 2000. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 6,* 1, 24–43.

Hornbæk, K., Bederson, B., and Plaisant, C. 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 Frokjaer, E. 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)*. 293–300.

Hornbæk, K., Frokjaer, E., and Plaisant, C. 2003. Reading patterns and usability in visualization of electronic documents. *ACM Trans. on Computer-Human Interaction (ToCHI) 10,* 2, 119–149.

Hornbæk, K. and Hertzum, M. 2007. Untangling the usability of fisheye menus. *ACM Trans. on Computer-Human Interaction (ToCHI) 14,* 2.

Jakobsen, M. R. and Hornbæk, K. 2006. Evaluating a fisheye view of source code. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*. 377–386.

Jul, S. and Furnas, G. W. 1998. Critical zones in desert fog: aids to multiscale navigation. In *Proc. ACM SIGCHI Symposium on User interface software and technology (UIST'98)*. 97–106.

Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. 2006. Challenges in visual data analysis. In *Proc. IEEE Information Visualization (IV'06)*. 9–16.

Lam, H. and Baudisch, P. 2005. Summary thumbnails: Readable overviews for small screen web browsers. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'05)*. 681–690.

Lam, H., Munzner, T., and Kincaid, R. 2007. Overview use in multiple visual information resolution interfaces. In *To apper in Proc. IEEE Symposium on Information Visualization (InfoVis'07)*.

Lam, H., Rensink, R. A., and Munzner, T. 2006. Effects of 2D geometric transformations on visual memory. In *Proc. Symposium on Applied Perception in Graphics and Visualization (APGV'06)*. 119–126.

Lamping, J., Rao, R., and Pirolli, P. 1995. A focus+context tehcnique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'95)*. 401–408.

Lau, K., Rensink, R. A., and Munzner, T. 2004. Perceptual invariance of nonlinear focus+context transformations. In *Proc. ACM Symposium on Applied Perception in Graphics and Visualization (APGV'06)*. 65–72.

Mountjoy, D. N. 2001. Perception-based development and performance testint of a non-linear map display. Ph.D. thesis, North Carolina State University, North Carolina, USA.

Nekrasovski, D., Bodnar, D., McGrenere, J., Munzner, T., and Guimbretire, F. 2006. An evaluation of pan and zoom and rubber sheet navigation. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'06)*. 11–20.

Nigay, L. and Vernier, F. 1998. Design method of interaction techniques for large information spaces. In *Proc. of Advanced Visual Interface (AVI'98)*. 37–46.

North, C. and Shneiderman, B. 2000. Snap-together visualization: can users construct and operate coordinated visualizations. *Journal of Human-Computer Studies 53,* 5, 715–739.

Pirolli, P., Card, S. K., and van der Wege, M. M. 2003. The effects of information scent on visual search in the hyperbolic tree browswer. *ACM Trans. on Computer-Human Interaction (ToCHI) 10,* 1 (Mar.), 20–53.

Plaisant, C., Grosjean, J., and Bederson, B. 2002. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proc. IEEE Symposium on Information Visualization (InfoVis'02)*. 57–64.

Plumlee, M. and Ware, C. 2006. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *13,* 2, 179–209.

Saraiya, P., Lee, P., and North, C. 2005. Visualization of graphs with associated timeseries data. In *Proc. IEEE Symposium on Information Visualization (InfoVis'05)*. 225–232.

Sarkar, M., Snibbee, S., Tversky, O., and Reiss, S. 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)*. 81–91.

Schafer, W. and Bowman, D. A. 2003. A comparison of traditional and fisheye radar view techniques for spatial collaboration. In *Proc. ACM Conf. on Graphical Interface (GI'03)*. 39–46.

Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., and Roseman, M. 1996. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Trans. on Computer-Human Interaction (ToCHI) 3,* 2 (Mar.), 162–188.

Shi, K., Irani, P., and Li, B. 2005. An evaluation of content browsing techniques for hierarchical space-filling visualizations. In *Proc. IEEE Symposium on Information Visualization (InfoVis'05)*. 81–88.

Skopik, A. and Gutwin, C. 2005. Improving revisition in fisheye views with visit wea. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'05)*. 771–780.

Tory, M., Kirkpatrick, A. E., Atkins, M. S., and Mller, T. 2006. Visualization task performance with 2d, 3d, and combination displays. *IEEE Trans. Visualization and Computer Graphics (TVCG) 12,* 1, 2–13.

Tullis, T. S. 1985. A computer-based tool for evaluating alphanumeric displays. In *Human-Computer Interaction: INTERACT '84*, B. Shackel, Ed. Elsevier Science, 719–723.

Zanella, A., Carpendale, M. S. T., and Rounding, M. 2002. On the effects of viewing cues in comprehending distortions. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'02)*. 119–128.

Received ...; ...; accepted ...

A. REVIEWED STUDIES: INTERFACES, TASKS, DATA AND RESULTS

This appendix summarized the key aspects of the multiple-VIR interface studies reviewed in the article. For each study, we listed the study interfaces, tasks, data and statistically significant 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 belong. 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.

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*)

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 color)

—*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*)

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 predefined 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

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 the participants to move a pointer along a path that is defined by objects in a visual workspace. In order to perform the task, the 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)

Reading of electronic documents: the usability of linear, fisheye and overview+detail interfaces [Hornbæk and Frokjaer 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*. The 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*)

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*)

Untangling the Usability of Fisheye Menus [Hornbæk and Hertzum 2007]

Interfaces:

—T: Hierarchial menu (Traditional cascading menu. For the smaller dataset, the menu had two VIRs. For the larger dataset, 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*)

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%)

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

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 color.)

—S: separate (Low-VIR interface with strips that encode the y-values of the line graph data with color 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*)

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:

—$T < E$
—$[T + S] < [E + S]$

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*)

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)

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:

—$S < T$

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)

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 color 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*)

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

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*)

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)