A Review of Overview+Detail, Zooming, and Focus+Context Interfaces

ANDY COCKBURN

University of Canterbury

and

AMY KARLSON and BENJAMIN B. BEDERSON

University of Maryland

There are many interface schemes that allow users to work at, and move between, focused and contextual views of a dataset. We review and categorize these schemes according to the interface mechanisms used to separate and blend views. The four approaches are overview+detail, which uses a spatial separation between focused and contextual views; zooming, which uses a temporal separation; focus+context, which minimizes the seam between views by displaying the focus within the context; and cue-based techniques which selectively highlight or suppress items within the information space. Critical features of these categories, and empirical evidence of their success, are discussed. The aim is to provide a succinct summary of the state-of-the-art, to illuminate both successful and unsuccessful interface strategies, and to identify potentially fruitful areas for further work.

Categories and Subject Descriptors: D.2.2 [**Software Engineering**]: Design Tools and Techniques—*User interfaces*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*

General Terms: Human Factors

Additional Key Words and Phrases: Information display, information visualization, focus+context, overview+detail, zoomable user interfaces, fisheye views, review paper

ACM Reference Format:

Cockburn, A., Karlson, A., and Bederson, B. B. 2008. A review of overview+detail, zooming, and focus+context interfaces. ACM Comput. Surv. 41, 1, Article 2 (December 2008), 31 pages DOI = 10.1145/1456650.1456652 http://doi.acm.org/10.1145/1456650.1456652

1. INTRODUCTION

In most computer applications, users need to interact with more information and with more interface components than can be conveniently displayed at one time on a single

This work was partially funded by New Zealand Royal Society Marsden Grant 07-UOC-013.

Author's addresses: A. Cockburn, Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand; email: andy@cosc.canterbury.ac.nz; A. Karlson, B. B. Bederson, Human-Computer Interaction Lab, Computer Science Department, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742; email: {akk,bederson}@cs.umd.edu.

screen. This need is dictated by pragmatic, technological, and human factors. The pragmatic issues concern form, specifically factors such as the size, weight, and fashion of displays that are used for varied tasks in diverse locations, as well as the cost of construction. Technological limitations constrain the ability of displays to match the breadth and acuity of human vision. Humans, for instance, can see through a visual angle of approximately $200^{\circ} \times 120^{\circ}$ (without turning the eyes or head), while typical computer displays extend approximately 50° horizontally and vertically when viewed from normal distances [Woodson and Conover 1964]. Similarly, the resolution of the human eye, with approximately 180 cone-receptors per degree in the fovea [Ware 2004], can perceive around 200 dots per centimeter (510 dots per inch) at a viewing distance of 50 centimeters (20 inches)—far more than the typical display resolution of 40 dots per centimeter (100 dots per inch). Finally, even if a perfect display could be constructed, displaying all information simultaneously is likely to hinder the user's ability to distinguish between important and irrelevant information.

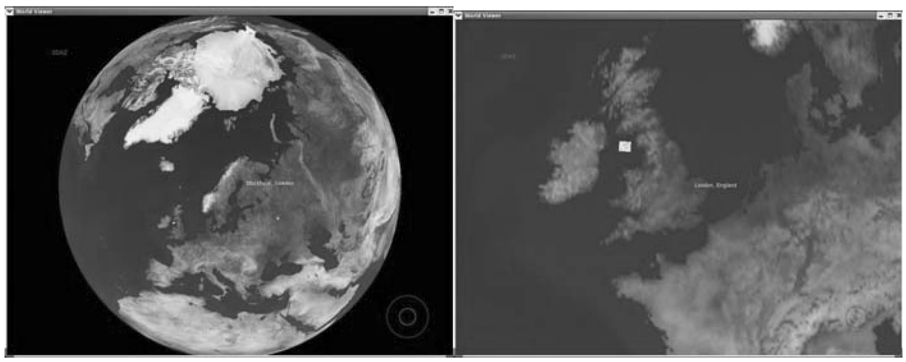
The traditional interface mechanisms for dealing with these display trade-offs involve allowing information to be moved (typically through paging, scrolling, and panning) or spatially partitioned (through windowing, menus, and so on). Although scrolling and windowing are standard in almost all user interfaces, they introduce a discontinuity between the information displayed at different times and places. This discontinuity can cause cognitive and mechanical burdens for users, who must mentally assimilate the overall structure of the information space and their location within it, as well as manipulate controls in order to navigate through it. For example, Grudin [2001] reports that reducing the discontinuity between views "reduces the cognitive load by allowing rapid glances to check on information; for example between a bird's-eye view of an entire graphical object and a close-up of a detail."

An alternative to moving or spatially partitioning the data space is to vary the scale at which the data is displayed, enabling multiple views at varying levels of detail. Other approaches involve broadening the user's field of view to allow the natural processes of human vision to determine focal and peripheral views, or systems can supplement the information space with cues that highlight salient items within their context. All of these schemes offer potential advantages derived from their ability to allow users to rapidly and fluidly move between detailed views and broad overviews.

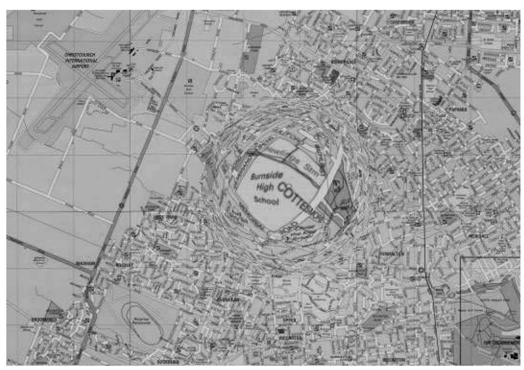
1.1. Road Map

The objective of this article is to summarize the state of research on interfaces that allow users to work at multiple levels of detail, and to identify effective and ineffective uses of them. This review is motivated by two factors. First, advanced interaction techniques are increasingly being deployed in desktop interfaces, yet research sometimes exists that can advise developers about the techniques' potential efficiency impacts. For example, Apple released their Mac OS X Dockwhich allows items to dynamically enlarge as the cursor approaches them, but research has now shown that the visual distortion of this fisheye effect (described later) harms targeting performance. Second, although there have been previous reviews of related work (Section 2), they are now dated and cover only a subset of research. In particular, they were published when the research focus was on developing new interaction techniques rather than on empirical analysis of their effectiveness. Recently, however, many researchers have turned their attention to empirically assessing the efficiency and effectiveness of their techniques. Through this review we aim to provide a succinct summary of the state-of-the-art in interfaces for working with focused and contextual views, and in the understanding of their performance issues as derived from empirical studies.

(a) Google Maps: an example overview+detail display. The overview inset in the bottom righthand corner of the display allows users to see the context of the detailed region.



(b) Two views in a zooming interface. Detailed and contextual views are temporally segregated within a single display.



(c) A fisheye view is one example of a focus+context interface. The focal region is magnified and displayed within its surrounding context.

Fig. 1. Example overview+detail, zooming, and focus+context interfaces.

Our review is organized into four categories which distinguish systems according to their varying uses of space, time, or visual effect. First, we describe work on overview+detail interfaces, which use a spatial separation to partition contextual and detailed information (see Figure 1(a) for an example). We then describe zooming interfaces, which use a temporal separation between the views (Figure 1(b)). The third category, called focus+context, eliminates the spatial and temporal separation by displaying the focus within the context in a single continuous view (fisheye views are one example, shown in Figure 1(c)). The final, cue, category modifies the way in which items are depicted in order to highlight, suppress, or contextualize them. For example, search results can be highlighted or nonresults visually deemphasized to draw the user's attention to elements of interest. The cue category depends on the availability of semantic information about elements in the dataset, and can be used to enhance any of the other three techniques.

Within each category we review features, foundations, and objectives, and we identify commercial exemplars where they exist. We also review research systems that demonstrate novel variations on the theme. After describing the categories, we summarize empirical work that identifies the strengths and weaknesses of the techniques, presented in two broad categories of user task: low-level mechanical tasks such as target acquisition; and higher-level cognitive tasks such as the user's ability to search or comprehend the information space. We finish the article by presenting summary guidelines and agendas for further research.

2. PREVIOUS REVIEWS

There have been several previous reviews of interfaces supporting focused and contextual views. Leung and Apperley [1994] provided the first comprehensive review of the set of visualizations that visually distort the information space (called *distortion-oriented* techniques). They used a metaphor of a rubber sheet to unify the theoretical properties of different methods that display the focal area within the surrounding context. The rubber-sheet analogy had previously been used by Tobler [1973], Mackinlay et al. [1991] and Sarkar et al. [1993] to describe visualizations. Keahey [1998] further described the tools and techniques that can be used to achieve focus and context through nonlinear magnification. Subsequently, Carpendale and Montagnese [2001] presented a mathematical framework, called the *elastic presentation framework*, that formalizes the relationship between diverse visualization schemes that present regions at different scales within, or on top of, one another.

Reviews have also addressed different application domains. Plaisant et al. [1995] describe user tasks for image browsing, considering alternative interfaces for overview+detail, zooming, and focus+context approaches. Herman et al. [2000] review graph visualization and navigation schemes, including focus+context views, with an emphasis on graph layout algorithms. In a sidebar to their paper, Kosara et al. [2002] provide a three-part taxonomy of interface methods for focused and contextual views: spatial methods, dimensional methods, and cue methods. Unfortunately, the sidebar presents only a few sentences on each technique.

Following the research emphasis of the time, the focus of these reviews had been on describing the properties of the visualizations, rather than on empirically comparing their effectiveness at supporting the users' tasks.

In his enlightening recent paper, Furnas [2006] also unifies disparate interface strategies for achieving focused and contextual views by revisiting the fisheye degree-of-interest function [Furnas 2006, FE-DOI, Section 5.1] that he originally proposed twenty years earlier [Furnas 1986]. A major theme of his new theoretical analysis is that many research visualizations primarily address the representational issues of *how* information is displayed, whereas the user's primary concern is with *what* data is selected for display: He contends that representation should be a secondary consideration to content. In earlier work than the 2006 paper, Furnas [1997] also presented a theoretical analysis of the problem of effectively navigating within information visualizations.

Several books on information visualization also cover overview+detail, zooming, and focus+context interfaces, notably the following: Ware [2004], which excels in the psychological foundations of visualization; Spence [2007], which provides a well-structured introduction to information visualization in a manner accessible to undergraduates; Chen [2004], which provides a rich coverage and distillation of research across the entire field; and Card et al. [1999], which provides a collation of the seminal research papers in information visualization. The goal of information visualization is to help users gain insight into the content of their data spaces, and consequently all of these books cover a much broader range of topics than those discussed in this article. Due to their scope, the books can only give limited direct coverage of interfaces for (and empirical research on) strategies for helping users work with focused and contextual views.

3. OVERVIEW+DETAIL: SPATIAL SEPARATION

An overview+detail interface design is characterized by the simultaneous display of both an overview and detailed view of an information space, each in a distinct presentation space. Google Maps, shown in Figure 1(a), is a canonical example of an overview+detail interface. Due to the physical separation of the two views, users interact with the views separately, although actions in one are often immediately reflected in the other.

Many forms of overview+detail interfaces exist, both in the standard desktop environment and in research systems, with important features including the ratio of scales in the detail and overview regions, the relative size and positioning of the views, mechanisms for navigation control, and the coupling between overview and detail displays.

In the remainder of this section we describe exemplar overview+detail interfaces and discuss the design issues involved in mapping their user-support to their discriminating features.

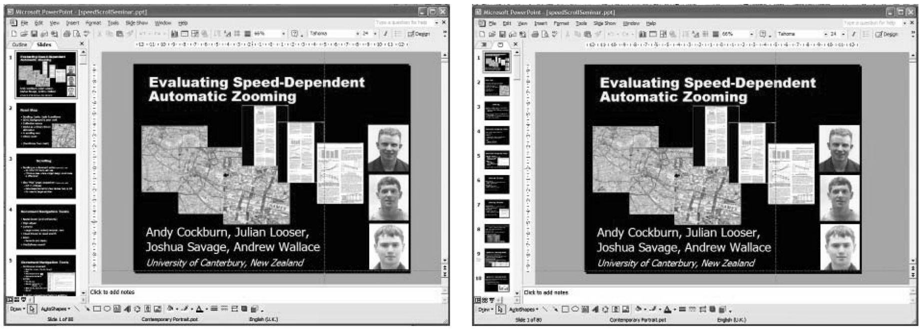
3.1. Scrollbars, Embellished Scrollbars and Thumbnail Overviews

Scrollbars constitute a familiar component of graphical user interfaces, and each can be considered to provide a one-dimensional overview of the document. The position of the scroll thumb (also called the *knob*) within the scroll trough (also called the *gutter*) portrays location in the document, and the length of the scroll thumb shows the portion of the document that is visible.

Although scrollbars typically encode only spatial information, several researchers have experimented with variants that additionally portray semantic information. *Value bars* [Chimera 1992] embellished the scroll trough with dashed lines, the thickness of which depended on numerical values extracted from columns in formatted data files, to depict the location of "interesting" or outlier data. Hill et al. [1992] used similar techniques to convey the read and edit history of document regions. Scroll troughs have also been used to indicate document passages that best match search queries [Byrd 1999] and to provide awareness of other people's locations and actions in real-time collaborative tasks [Gutwin et al. 1996].

As the level of detail presented in the scroll trough increases, the scrollbar becomes a first-class overview window. Many everyday applications, such as Microsoft PowerPoint and Adobe Reader, support *thumbnail* document overviews that blur the boundary between scrollbar and overview for navigation (see Figure 2).

Designing thumbnail overview interfaces involves addressing a trade-off in scale ratios: low scale ratios allow thumbnails that clearly represent the underlying dataset, but at the cost of increased overview scrolling to access distant thumbnails, while high scale ratios allow rapid access to more distant pages, but at the cost of visual clarity. Many applications therefore allow users to directly control the size of the thumbnail



(a) five thumbnails in the overview (b) ten thumbnails in the overview

Fig. 2. PowerPoint's overview+detail interface. The scrollable thumbnail overview is on the lefthand side of each window, and the thumbnails can be scaled by configuring the width of the overview region.

region and its scale factor (Figure 2 shows Microsoft PowerPoint with two overview-region sizes).

Although it seems natural that thumbnails should allow shortcut navigation (clicking on a thumbnail should cause that region of the information space to appear in the detailed view), it is less clear whether the overview and detailed regions should be synchronized so that they continually display corresponding locations. Synchronization is not likely to cause disorientation, but it is less powerful, as it forbids independent exploration of document regions in the two views. Many variant implementations are possible: For example, Microsoft PowerPoint implements a one-way synchronization in which the overview is synchronized with the detail view (scrolling the detail causes corresponding movement in the overview), while the detail is unsynchronized with the overview (scrolling the overview has no side-effect on the detail).

3.2. Standard Overview+Detail Views

The first examples of overview+detail views (that we are aware of) are in computer games of the early 1980's, such as Defender, which provided a broad overview of the game's information space at the bottom of the screen. They are now widely used in many application areas, particularly mapping and image-editing tools. Google Maps (Figure 1(a)), for example, has a small, inset overview region which includes an interactive rectangular subregion that corresponds to the area shown in the detailed view. Panning actions in the detail view are immediately reflected within the overview. However, panning actions in the overview are less tightly coupled with the detail area; the detail view is only updated to match actions in the overview when the user completes the panning activity by releasing the mouse. This slight decoupling allows users to explore the overview without altering the information presented in the detail view, while at the same time reducing computational and network load.

Given the wide variety of application areas for overview+detail interfaces, it is unlikely that generic guidelines for the "right type" of view coupling can be provided, but we note that it is becoming standard practice to allow interactive exploration of the overview without influencing the detail view, while manipulation of the detail view is immediately reflected in the overview (e.g., Microsoft Word and PowerPoint, Adobe Reader, and Google Maps all operate in this manner). We further note that it is common to enable users to control whether the overview is visible, implying that designers believe that overviews may not be useful to all users or for all tasks. More generally, however, the visualization problem of coordinating multiple views is gaining increasing

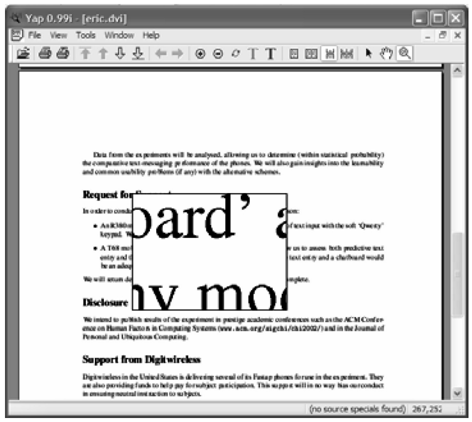


Fig. 3. Z-based overview+detail separation. In the "Yap" dvi file previewer, a magnified detail region is shown in a lens that follows the user's cursor.

attention, but reviewing this is beyond the scope of this article, see Roberts [2007] for a recent review of the state-of-the-art.

3.3. Lenses and Z-Separation

The aforementioned systems and techniques all separate the overview and detail regions on the x- and y-coordinates. Several systems, however, separate the views on the z-coordinate, with overviews overlaying or blending with the background detail.

Lenses are moveable display regions that overlay the default window. Although lenses can be similar to *fisheye distortions* (described later), we categorize them as overview+detail because they separate (on the z-plane) the detail and overview.

Lenses have been used as a magnification metaphor in standard desktop environments since the early 1990's. Figure 3 shows a magnification lens in the document previewer "Yap" where the magnified region follows the user's cursor. Bier et al. [1993] introduced toolglass widgets and magic lenses together under the same design philosophy. Toolglass widgets are resizable see-through windows, normally controlled by the user's nondominant hand, that allow users to perform specialized operations on the data-space objects over which they are positioned. For example, to change the color of an object in a drawing application, the user would place the color-selector toolglass over the target using his/her nondominant hand, and then click-through the toolglass using the dominant hand. Magic lenses have the same form as toolglass widgets, but they transform the visualization of underlying objects to allow focused viewing of specific attributes.

Although lenses are normally much smaller than the underlying detailed area, the lens can be the same size as the detailed region, with transparency visually separating the overview and detail "layers." Display hardware is also available to implement the layered separation. PureDepth[1] manufactures a dual-layer LCD screen, with a small (1–5cm) separation between front and rear LCD layers. Images displayed on the front layer can partially or entirely occlude those displayed on the rear layer. Claims that this technique aids cognitive separation of data on different layers have not yet been supported by experimental results [Aboelsaadat and Balakrishnan 2004].

[1]www.puredepth.com.

4. ZOOMING: TEMPORAL SEPARATION

The second basic category of interface supporting both focused and contextual views is based on zooming, which involves a temporal separation between views; users magnify (zoom in to focus) or demagnify (zoom out for context) a dataset *in place* rather than seeing both views simultaneously. Naturally, zooming and overview+detail features can be combined, but in this section we address the isolated issues of zooming.

Some of the design issues for zoomable interfaces that are discussed in this section include the transition between zoom states (continuous or discrete zoom levels, and the use of animation), the interaction controls for zooming in and out, the coupling between controls for movement (panning) and zoom, and the use of semantic zooming to alter the presentation of data items at different scale levels.

4.1. Standard Desktop Applications of Zooming

Many standard desktop interfaces allow the user to zoom the dataset. The controls for zooming are commonly presented as toolbar widgets for specifying magnification as a percentage of normal or zoom selection dialog boxes. For applications that support a zoom mode (e.g., Adobe Reader) all mouse clicks and drags are interpreted as zoom actions while in that mode. Some interfaces, however, support zoom control through specific interaction sequences using the mouse or keyboard (e.g., double clicking, CTRL+"+", CTRL+scrollwheel), and thus have no visual representations. This approach raises two potential problems: First, the user may not discover how to activate the zoom functions; and second, s(he) may be unable to reverse the action. For example, in an informal study for this article, only one of six computer scientists who were also experienced Microsoft Word users had learned that holding the control key while using the mouse scrollwheel was a shortcut for zoom; similarly, none of them realized that double clicking with the left button is a zoom-in shortcut in Google Maps, and when shown this feature it took most of them many attempts to determine how to reverse the action without the widgets (double clicking the right button).

The problem of reversing zooming actions is exacerbated by the fact that undo, that is, the standard mechanism for action reversal, is normally reserved for actions that modify the data-state and not for those that modify the view. Region-select for zoom-in should be offered with caution (although powerful for appropriate uses) because there is no equivalent action for zooming out.

4.2. Zooming Toolkits and Models

Many zoom-based research systems have been developed to demonstrate interaction and visualization techniques [Card et al. 1999]. The Pad system [Perlin and Fox 1993] was the first fully zoomable desktop environment, and it introduced two important concepts: *semantic zooming*, which allows objects to be represented differently at different scales; and *portals*, which allow links between data objects and filters on their representations. Pad prompted extensive further research on these and related topics, and toolkits were developed to ease the implementation of zoomable interfaces, including Pad++ [Bederson et al. 1996], Jazz [Bederson et al. 2000], Piccolo [Bederson et al. 2004a], and ZVTM [Pietriga 2005]. Several application domains have been explored using these tookits, including drawing tools for children [Druin et al. 1997], authoring tools [Furnas and Zhang 1998], image browsers [Bederson 2001], and software visualization [Summers et al. 2003].

The experience of building many zoomable applications across multiple domains revealed domain-independent design challenges, which Furnas and Bederson [1995] addressed with space-scale diagrams, a framework for conceptualizing navigation using

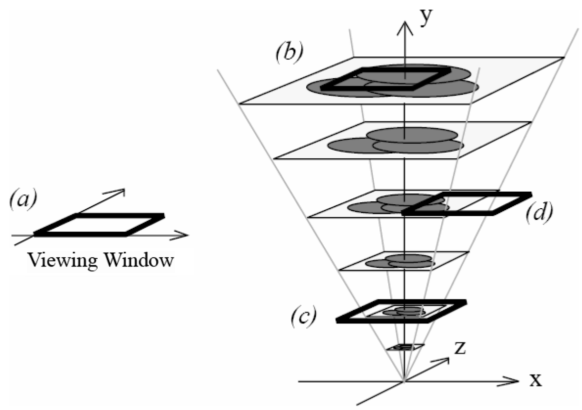


Fig. 4. A space-scale diagram [Furnas and Bederson 1995]. The viewing window: (a) is shifted rigidly around the 3D diagram to obtain all possible pan/zoom views of the original 2D surface, for example; (b) a zoomed-in view of the circle overlap; (c) a zoomed-out view including the entire original picture; and (d) a shifted view of a part of the picture.

zoom and pan. Figure 4 shows a space-scale diagram, with scale on the y-axis and space on the x/z-axes. The user's viewport (Figure 4(a)), shown by the superimposed region, remains a constant size during manipulation of scale and space. Clearly, as scale increases (zooming in goes up the y-axis) the amount of panning required to traverse the region also increases. Van Wijk and Nuij [2004] formalized the mathematics involved in generating smooth and efficient animation trajectories in pan-zoom space (further discussed in the folowing). Another general usability issue associated with zooming is "desert fog" [Jul and Furnas 1998], a term which captures navigation problems caused by an absence of orienting features due to their increased separation at high zoom levels. To combat desert fog, Jul and Furnas introduce *critical zones* (a cue-based technique) to provide a visual demarcation of regions that are guaranteed to yield further information when zoomed. Alternatively, Bederson [2001] suggests designing the interaction to be closely linked to the content so it is impossible to navigate to empty regions.

4.3. Animation

Some methods for changing zoom scale, such as updating a magnification widget in a toolbar, jump the user immediately to the new view. However, animation is frequently used to help users assimilate the relationship between pre- and post-zoom states [Bederson and Boltman 1999; Tversky et al. 2002]. Animation causes a brief period of automatic zooming: Rather than controlling the animation, the user simply observes it.

Designing zoom animations requires finding a suitable transition speed that reveals the relationship between zoom states without slowing the user's overall interaction: Research suggests values between 0.3 and 1.0 second are appropriate [Card et al. 1991; Bederson and Boltman 1999; Klein and Bederson 2005], depending on the application domain and distance traveled. Sometimes, however, users need rich cues to maintain comprehension and orientation throughout the animation, demanding longer animated periods and more detailed representations throughout the transition. To this end, Van Wijk and Nuij [2004] developed a set of equations for reducing optical flow during pan/zoom trajectories between two known points in a space-scale diagram (from one point in space at one level of zoom to another point at a different level of zoom).

4.4. Automatic Zooming and Parallel Zoom/Pan Control

Van Wijk and Nuij's formulae calculate smooth and efficient trajectories for motion between *known* points. Most information browsing, however, involves dynamic modification of paths through space and scale in response to user feedback. Traditional interfaces impose a temporal separation between users' panning and zooming actions, but two research threads are investigating concurrent control for motion and zoom.

First, several systems offer users explicit parallel control of movement and zoom. Examples include bimanual systems [Guiard 1987] that assign panning and zooming to devices controlled in different hands [Bourgeois and Guiard 2002]. OrthoZoom Scroller [Appert and Fekete 2006] is a one-handed variant of this approach for one-dimensional (vertical) scrolling. It maps vertical mouse actions to vertical panning, and horizontal mouse actions to zooming.

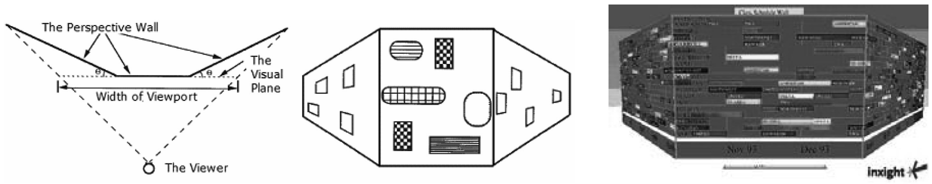
Second, some systems use an automatic connection between zooming and panning, calculating one from the user's manipulation of the other. Mackinlay et al. [1990] introduced the first related technique with their three-dimensional point-of-interest movement functions, which moved the view towards a specified point by a constant percentage of the remaining distance in each animation cycle, thus producing slower velocities and easier control in the latter stages of object approach (similar to zooming in). Addressing a similar problem in three-dimensional worlds, Ware and Fleet [1997] introduced *depth-modulated flying*, which automatically adapted the viewer's velocity in "fly-by" visualizations in response to the user's control of altitude (similar to zoom level). Tan et al. [2001] then inverted the speed/zoom binding with "speed-coupled flying with orbiting," in which the viewpoint altitude is automatically adjusted in response to velocity. In more standard desktop applications, Igarashi and Hinckley [2000] implemented several *speed-dependent automatic zooming (SDAZ)* systems to reduce disorientation when users scroll rapidly. By automatically zooming away from the document as the scroll-rate increases, the pixel-rate of movement is kept within human perceptual limits.

One potential limitation of combined zooming and panning is that it may impede users in forming and exploiting a spatial understanding of the information space: The data moves to different locations in the screen, dependent on space and scale settings. To investigate whether stable spatial representations aided users' ability to navigate in documents, Cockburn et al. [2006] designed a simple *space-filling thumbnails* interface, which constrained the user to two temporally separated views: a detailed view for reading the document, and a fully zoomed-out view in which all pages are represented as tiled thumbnails within a single screen. Although highly constraining, the interface ensured spatial stability within the window as scrolling and panning were disabled.

Evaluations of these systems are described in Section 7.

5. FOCUS+CONTEXT: SEAMLESS FOCUS IN CONTEXT

The previous methods discussed for managing focused and contextual information had separated the two views in either space (overview+detail) or time (zooming), leaving the user to assimilate their global relationship. The third approach, called focus+context, integrates focus and context into a single display where all parts are concurrently visible: The focus is displayed seamlessly within its surrounding context. Research into *fisheye view* interfaces, which diminish or suppress information that lies away from the focal area, is particularly well represented in this category. Many focus+context interfaces, including some fisheye views, use differential scale functions across the information surface, leading to intentional distortion such as that shown in Figure 1(c). Consequently, the term "distortion-oriented view" is also commonly applied in this



(a) bird's-eye view (b) head-on view (c) the running system

Fig. 5. The perspective wall (with permission of Mackinlay et al. [1991]).

category. By presenting all regions in a single coherent display, focus+context systems aim to decrease the short term memory load associated with assimilating distinct views of a system, and thus potentially improve user ability to comprehend and manipulate the information.

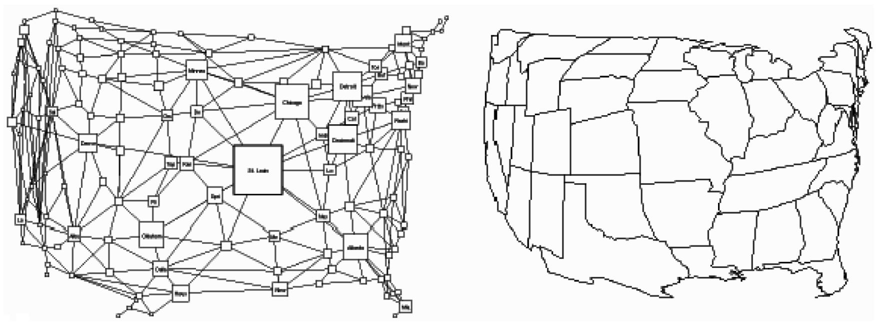
This section provides a brief review and introduction to the wide range of systems, techniques, and theories encapsulated within focus+context research. Recurring themes include the following: selective presentation versus distortion of data items (i.e., several systems calculate the user's degree of interest in data items to determine which are displayed, while others use the degree-of-interest value to change the position and scale of their presentation); system- versus human-controlled focus+context (i.e., some systems attempt to infer the user's degree of interest through interaction behavior such as the location of his/her cursor, while others defer to user manipulations of the focal region, or simply exploit users' natural capabilities for focused attention and peripheral awareness, as with wide-screen displays); and single versus multiple foci (i.e., while most systems only support a single focal point, some support an arbitrary number of concurrent focal points).

5.1. Visions and Theoretical Foundations

Spence and Apperley [1982] described the first focus+context visualisation, which they named the *bifocal display*. It used a metaphor of paper stretched across four rollers, with two foreground rollers giving a rectangular focal region, and two distant rollers on either side giving receding side-plate displays. The user changed his/her focus by sliding the paper in either direction across the frame. Nearly a decade later, Mackinlay et al. [1991] implemented the concept in their *perspective wall*, shown in Figure 5.

The theoretical foundations for focus+context interfaces were established by Furnas [1986], who described a "generalized fisheye view" formula for calculating the user's *degree of interest (DOI)* in objects in the data-space: $\mathrm{DOI}(x|y) = \mathrm{API}(x) - \mathrm{D}(x, y)$, where x is a data element, y is the current focus, $\mathrm{API}(x)$ is the a priori interest in object x, and $\mathrm{D}(x, y)$ is the spatial or semantic distance between x and y. Furnas described how his formula could be applied to a variety of information domains, with data objects being removed from the display when they fell below a threshold DOI value. While Furnas's DOI formula determines *what* information should be included in the display, much of the subsequent research has primarily addressed the issue of *how* information is presented [Furnas 2006].

Extending fisheye views, Sarkar and Brown [1992] presented geometric transformations that produce visual distortions of graphs and maps, based on the Euclidean distance between vertices or coordinates and the user's focal point. The transformations determine each point's size, position, and level of detail in the display (see Figure 6). The Sarkar and Brown algorithm has been heavily used in fisheye visualizations. Lamping et al. [1995] present an alternative focus+context transformation for hierarchical



(a) a Cartesian transformation of a graph of major U.S. cities

(b) a polar transformation of a map of the U.S.

Fig. 6. Sarkar and Brown [1992] fisheye transformations.

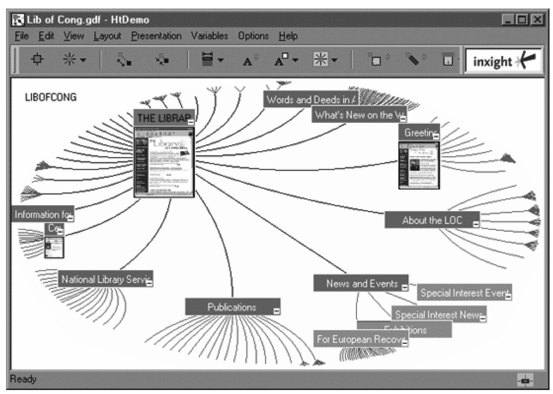


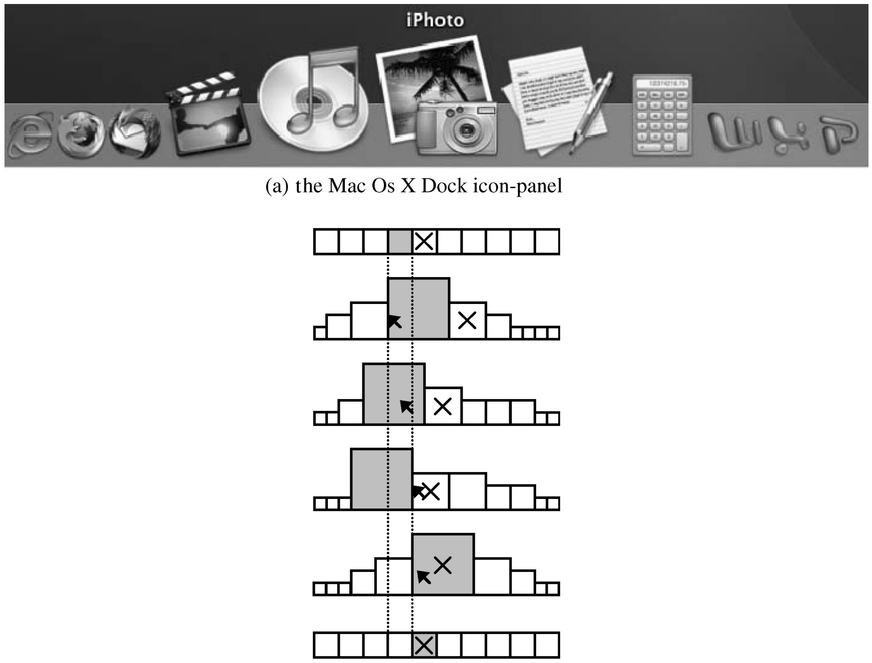
Fig. 7. The Hyperbolic tree browser (with permission of Lamping et al. [1995]).

data structures based on hyperbolic geometry (see Figure 7). Several other systems also demonstrate focus+context techniques for tree visualization, including SpaceTree [Plaisant et al. 2002], which uses extensive zooming animation to help users stay oriented within its focus+context tree presentation, and the TreeJuxtaposer [Munzner et al. 2003], which uses focus+context methods to support comparisons across hierarchical datasets.

Two potential problems with fisheye views are, first, misinterpretation of the underlying data, and second, challenges in target acquisition, both of which are caused by distortion of the information space. As Figure 1(c) shows, north/south and east/west gridlines distort around the center of a fisheye, leading to ambiguity regarding location and direction within the lens. Zanella et al. [2002] propose using grids and shading to alleviate misinterpretation. The fisheye target acquisition problem is caused by the lens displacing items away from the actual screen location used to activate them, as shown in Figure 8.

5.2. Sample Applications

Many systems demonstrating focus+context have been developed. Empirical evidence of focus+context success is presented in Section 7.



(a) the Mac Os X Dock icon-panel

(b) target movement caused by distortion. Items can be displayed at different locations (middle rows) from the actual locations needed to select them (top and bottom rows)

Fig. 8. Fisheye distortion effects with icon-panels.

5.2.1. Fisheye Views for Targeting in Desktop Applications. The Mac OS X Dock icon-panel (Figure 8(a)) incorporates the first large-scale deployment of fisheye-style effects. Items in the icon-panel expand as the user's cursor moves towards them, providing a dynamic and rich visual effect that users may appreciate.

Theoretically similar to the icon-panel, fisheye menus were developed by Bederson [2000] to help in selecting items in long menus, for example, selecting a country name from a combo-box. Each line of text in the fisheye menu is displayed in an extremely small font, with the item and font size increasing as the cursor moves closer. To partially resolve the problem of items moving as the cursor approaches, users can "lock" the lens by dragging into a region to the right of the normal menu.

5.2.2. Fisheye Documents. Furnas's original description of the fisheye technique used computer programs as an example: When the fisheye DOI formula calculated a low value for a program region, it was removed from the display. Although manual controls for removing program block statements and methods are now standard features of integrated development environments such as Microsoft Visual Studio 2005 and Netbeans (see Figure 9), automatic expansion and contraction using the DOI algorithm remains largely confined to research labs (the Mylar/Mylyn program visualization extension to the Eclipse IDE is one example of widespread use of DOI [Kersten and Murphy 2005]). Everyday word processors also include manual structure-based data-hiding capabilities such as the "Outline View" of Microsoft Word, which allows successive folding and unfolding of document sections. While systems such as these provide discrete visualization control where regions are either included or excluded, other systems such as the

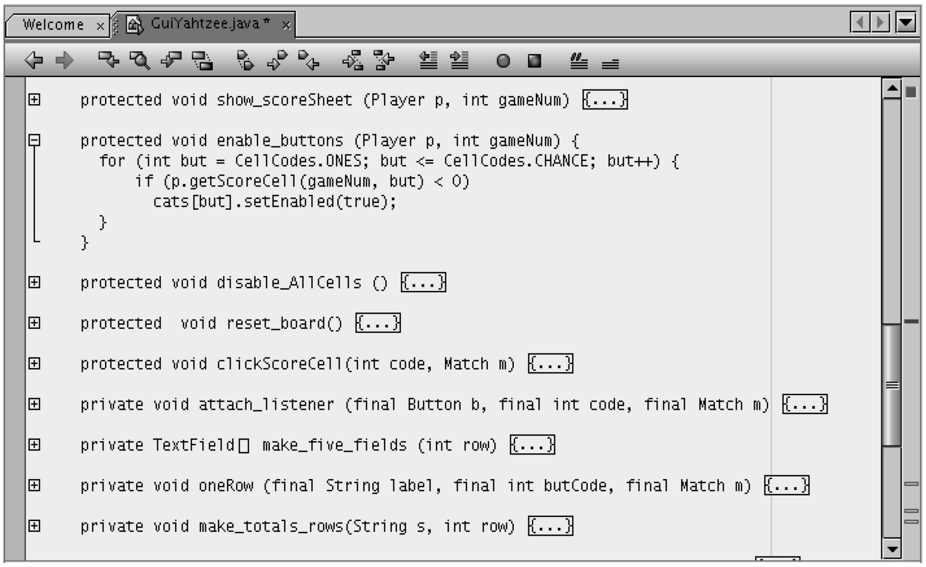


Fig. 9. Netbeans IDE allows code detail to be removed from the display, allowing a broader contextual view of the program code.

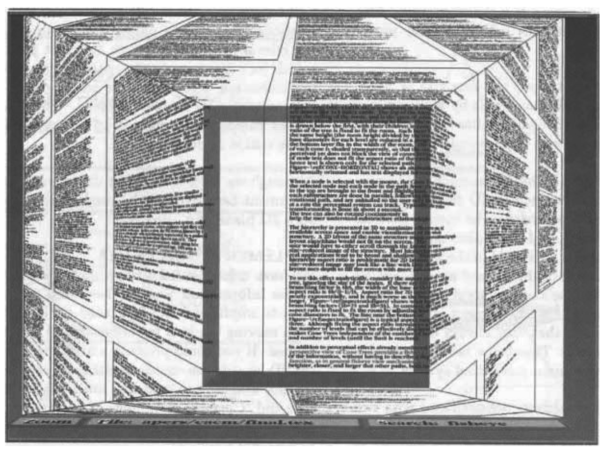


Fig. 10. The document lens (with permission of Robertson and Mackinlay [1993]).

document lens [Robertson and Mackinlay 1993] use continuous functions to diminish document regions with distance from the focus (see Figure 10).

5.2.3. Fisheye Tables. Furnas's original paper also described how fisheye views could be applied to tabular data such as calendar entries, with TableLens [Rao and Card 1994] providing the first interactive demonstration (see Figure 11). TableLens presents a compact overview of large datasets, displaying all rows and columns simultaneously by encoding values as small bars. Fisheye effects are available to selectively expand rows and columns to reveal the attribute values of the associated data. Expansion can be applied independently to rows and columns, allowing multiple focal points, but preserving the familiar rectangular format of cells.

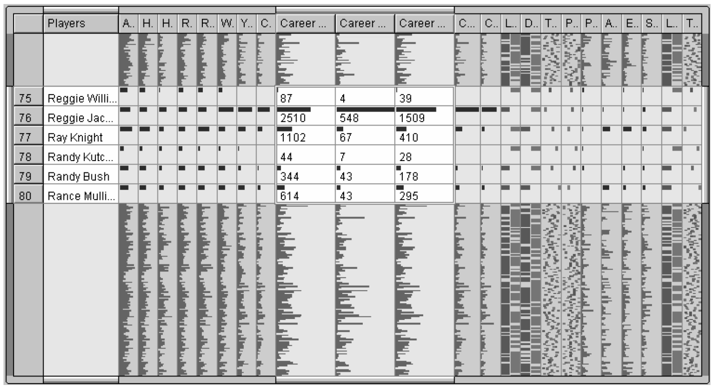
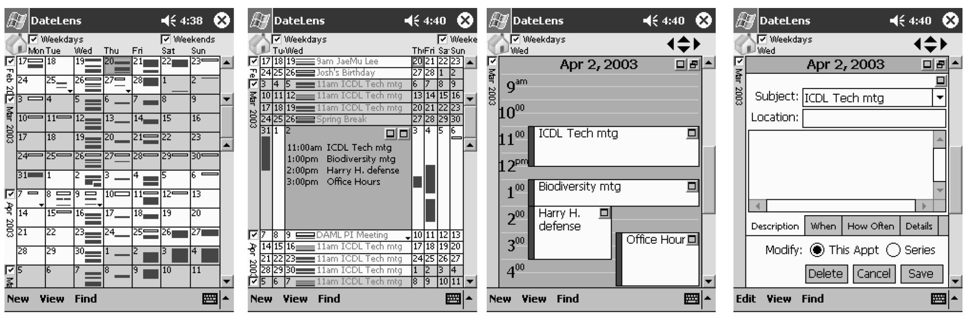


Fig. 11. TableLens (with permission of Rao and Card [1994]).



(a) overview (b) one-day zoom (c) one-day focus (d) appointment focus

Fig. 12. The DateLens interface with the view configured to show 12 weeks at consecutive levels of detail. All transitions between views are animated.

Bederson et al. [2004b] applied concepts from TableLens within the DateLens fisheye calendar tool (see Figure 12). Designed with the constrained display space of PDAs in mind, DateLens supports visualization of different time spans (days, weeks, months), as well as a variety of search and presentation tools to illuminate patterns and outliers.

5.2.4. Wide Field-of-View Systems. Another approach to providing concurrent focus+context is to increase the size and resolution of the display, allowing natural human methods such as gaze direction, peripheral vision, and control of proximity to the work surface.

Several display technologies can be used to produce a wide field of view: large low-resolution displays, multimonitor environments, large high-resolution displays, large mixed-resolution displays, and heads-up displays. Large low-resolution displays simply enlarge each pixel, which is useful for distant viewing but offers little advantage for close work due to low detail. Large high-resolution displays are created either by placing multiple monitors alongside one another [Grudin 2001], or by tiled projection to minimize seams between the outputs from multiple graphics card heads [Robertson et al. 2005].

Mixed-resolution displays offer an inexpensive method for producing both a large display surface area and a detailed focal region. Baudisch et al. [2002] demonstrated the technique using a small high-resolution display region (1024 × 768 pixels, ~15 inch flat-panel display) within a large 4 × 3ft, 1024 × 768 pixel projected display

Fig. 13. A mixed-resolution large display using a 1024 × 768 LCD panel focal area inset within a 1024 × 768 projected contextual display (with permission of Baudisch et al. [2002]).

(see Figure 13). The images presented in the two displays were stitched together in software to ensure that panning actions in one caused corresponding updates to the other. This technology, however, is a stop-gap measure until high-resolution large displays such as the tiled wall displays by Guimbretiere et al. [2001], are available at low cost.

Finally, heads-up displays can also emulate a wide field of view [Patrick et al. 2000], but current devices are limited to relatively low resolution, reducing their ability to support detailed views.

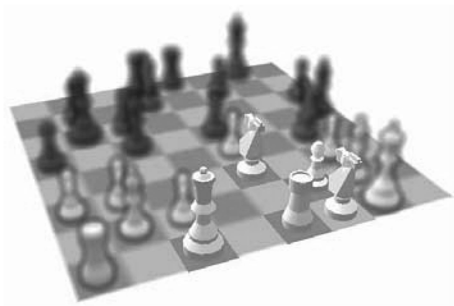
Lessons from empirical studies of large displays are presented in Section 7.

6. CUE-BASED TECHNIQUES

Almost all of the overview+detail, zooming, and focus+context approaches described earlier modify the size of objects in order to support both focused and contextual views. These scale modifications can be applied purely to the graphical portrayal of objects, or semantically so that only objects with certain properties are scaled.

Cue-based techniques, on the other hand, modify how objects are rendered and can introduce proxies for objects that might not be expected to appear in the display at all. They can be used in conjunction with any of the aforesaid schemes, and are typically applied in response to some search criteria. Data items satisfying the criteria are then displayed in modified form to alert the user to their presence. Strong consideration and support of information scent, that is, “the strength of local cues, such as text labels, in providing an indication of the utility or relevance of a navigational path leading to some distal information” [Pirolli et al. 2001, p. 507], is particularly important in the design of cue-based techniques.

Given this broad definition of cue-based techniques, much of the work on information visualization could be included within this category. We confine ourselves to a few examples that are particularly pertinent to the problems of focus and context.



(a) Illustration depicts depth-of-focus *blurring* (with permission of Kosara et al. [2001])

(b) Halo depicts items beyond the window edge (with permission of Baudisch and Rosenholtz [2003]).

Fig. 14. Two cue-based techniques.

6.1. Cue Techniques for Highlighting Focal Objects

Kosara et al. [2001] described a *semantic depth-of-field* technique that provides a natural interface for drawing the user's attention to focal items. With this technique, items that satisfy search criteria are displayed in focus, while all others are slightly blurred (see Figure 14(a)).

Several researchers have examined techniques for adding cues to the presence of search terms in Web pages such as normal displays and overviews with the Popout Prism [Suh et al. 2002] and fisheye views with Fishnet [Baudisch et al. 2004]. Bederson et al. [2004b] used visual demarcations in the scrollbar trough to convey the presence of search matches across several months of appointment data in the PDA calendar DateLens, and similar markings are used in the Netbeans IDE (right of Figure 9) to convey where recent edits have taken place in program code. Lam and Baudisch [2005] made text boxes more visable in scaled down Web pages by displaying summary text at a readable size.

6.2. Cue Techniques for Extending Context Beyond the Window Edge

Contextual cues about information lying outside the main display region can be added with *decorations*. City Lights, developed by Zellwegger et al. [2002], used window-edge decorations to indicate the existence, size, and/or location of objects that lay beyond the window frame. In Halo (Figure 14(b)), a variation of City Lights, Baudisch and Rosenholtz [2003] explored the use of arcs as decorators, as though each off-screen object were a street lamp just tall enough to cast its circle of light into the screen viewport. Nearby objects cast short, highly curved arcs, while far objects cast long, subtly curved arcs. In this way, object direction and distance are encoded as arc placement and curvature.

7. EMPIRICAL EVALUATIONS

Many overview+detail, zooming, and focus+context interfaces have been developed since the 1980's, but until recently there was relatively little empirical evidence of their success. We briefly review this empirical research, grouping the evaluations based on whether they primarily address low-level aspects of interaction such as target acquisition, or high-level user aspects such as the ability to comprehend the information space.

Table I. Low-Level Evaluations of Mechanical Manipulation and Target Acquisition

Author and Year	O+D	Zoom	F+C	Description
Ramos et al. 2007	⟷			Target acquisition. Shows that pressure-activated lenses can aid acquisition of small targets.
Cockburn et al. 2006	⟷	⟷		Document navigation. Thumbnail-enhanced scrollbars (O+D) and several zooming variants are outperformed by a zooming interface that presents all pages as a stamp-sheet of page-thumbnails.
Gutwin and Skopik 2003	⟷	⟷	⟷	Steering tasks (dragging through a constrained path). Results showed that O+D and pan and zoom were slow compared to several fisheye variants.
Pietriga 2007	⟷	⟷	⟷	Mechanical aspects of multiscale searching. Compared O+D, pan and zoom, fisheye F+C, and DragMag interfaces for identifying abstract tasks. O+D performed best, followed by DragMag, F+C, then pan and zoom.
Shoemaker and Gutwin 2007		⟷	⟷	Multipoint target acquisition. Compared pan, zoom, and two fisheye techniques for selecting pairs of points. A fisheye allowed fastest task completion. Combined zooming and panning was not allowed.
Gutwin 2002			⟷	Target acquisition. Describes the fisheye moving target problem, and proposes and evaluates "speed-coupled flattening," which eases it.
McGuffin and Balakrishnan 2002; Zhai et al. 2003			⟷	Target acquisition. Targets that maintain stationary centers, but which expand in motor space as the cursor approaches are faster to acquire than static ones.
Hornbaek and Hertzum 2007			⟷	Menu item acquisition. Analyzes menu item selection with fisheye menus. Demonstrates they are slower than traditional hierarchical menus.
Guiard et al. 2001; Guiard et al. 2004		⟷		Target acquisition. Careful analysis of zooming as a tool for high index of difficulty pointing tasks.
Cockburn et al. 2005		⟷		Document navigation. Calibration of perceptual issues of the relationship between scroll-speed and zoom, and comparative evaluation of automatic zooming versus traditional scrolling.
Ramos and Balakrishnan 2005		⟷		Target acquisition. Compares parallel input mechanisms for zooming and panning (a.k.a zliding). Unimanual stylus "zliding" (pressure for zooming and dragging for sliding) outperforms bimanual methods.
Appert and Fekete 2006; Mackay et al. 2007		⟷		Document navigation. Compares several zooming variants, concluding that OrthoZoom's concurrent unimanual control for pan and zoom outperforms bimanual controls and automatic zooming.

This summary of evaluative work largely excludes cue-based techniques: Providing a cue to highlight objects that match some semantic criteria will aid users in finding them.

7.1. Low-Level Evaluations

Most of the low-level evaluations, summarized in Table I, each investigate only one category of technique (overview+detail, zooming, or focus+context), either comparing performance with and without that method or analyzing its interaction characteristics.

The low-level problem of target acquisition efficiency (the time taken to select items of different sizes at different distances) is particularly well represented in this set of studies. It is a well-understood topic in HCI research, with an accurate and robust model of performance called Fitts' law [Fitts 1954]. Although Fitts' law is traditionally applied to targets that are continuously visible, it also accurately models target acquisition across large distances with many different styles of zooming interface [Guiard et al. 2007, 2004, 2001].

Target acquisition tasks are important in human-computer interaction research because they largely isolate the mechanics of interface control from the context of use. However, even in the constrained domain of target acquisition there are many experimental variants that can influence results. Two important considerations are how the targets are depicted at various scale levels, and whether each target is selected once or multiple times. The target revisitation issue is important because some interfaces will provide better support for the formation and exploitation of human spatial memory.

Recent research is investigating how to best operationalize, control, and automate the methodology for evaluating multiscale interfaces [Guiard et al. 2007; Mackay et al. 2007; Pietriga et al. 2007].

7.1.1. Linking of Controls for Pan and Zoom. In investigating how best to manipulate the concurrent control of panning and zooming, Ramos and Balakrishnan [2005] found that unimanual "zliding" with a stylus, which used pressure for zoom and dragging for pan, outperformed a variety of techniques that used bimanual separation. Evaluations of OrthoZoom Scroller [Appert and Fekete 2006; Mackay et al. 2007], described in Section 4.4, support this result that concurrent and unimanual control of pan and zoom outperforms serial or bimanual separation. Speed-dependent automatic zooming provides another mechanism for coupling movement and zoom level, but while results show that it can offer improvements over traditional scrolling techniques and over serial separation of pan and zoom [Tan et al. 2001; Cockburn et al. 2005], Appert and Fekete's [2006] study show it is inferior to OrthoZoom's parallel and manual control.

In certain contexts it is possible to remove panning altogether. The space-filling thumbnails system for document navigation (described Section 4.4) demonstrates this approach by constraining users to two views: a fully zoomed-in view for reading, and a fully zoomed-out view that shows all document pages as tiled thumbnails at stable locations within one window [Cockburn et al. 2006]. Their evaluation demonstrated that it allowed users to quickly form and exploit their spatial memory in aid of rapid navigation, significantly outperforming other panning, zooming, and overview+detail interfaces, particularly when tasks involved revisiting previously seen pages.

7.1.2. Target Acquisition with Distortion-Based Views. When a visualization dynamically responds to the location of the cursor, as is common with fisheye lens systems such as the Mac OS X dock (Figure 8(a)), then there can be a disparity between the visual location of the item and the cursor location needed to acquire it (see Figure 8(b)). Fisheye menus [Bederson 2000] exhibit this problem, and therefore support a fisheye "lock" to allow users to stabilize the focal region. Hornbaek and Hertzum [2007] closely analyzed interaction with fisheye menus, concluding that users make little use of the nonfocal display region and that performance with fisheye menus is inferior to traditional hierarchical menus. Gutwin [2002] also studied the impact of item movement with fisheyes, demonstrating the problem can be eased through "speed-coupled flattening," which reduces the fisheye effect as the velocity of the cursor increases. To avoid the moving-target problem, an alternative solution is to delay applying distortion until a focus object has been selected, as demonstrated by the hyperbolic tree browser [Lamping et al. 1995].

In related work on acquiring dynamic targets, McGuffin and Balakrishnan [2002] and Zhai et al. [2003] showed that acquisition times are reduced when discrete targets expand around their center to fill an enlarged motor space, even when the expansion starts after 90% of the movement toward the target is complete. Both papers suggest modifications to the fisheye Mac OS X dock that would allow it to maintain the appealing visual effect, but without the adverse effects of target movement.

Two studies of low-level interaction have suggested tasks where distortion-oriented fisheye views may provide an advantage over other techniques. First, in steering tasks where the user has to move the cursor through a constrained path, Gutwin and Skopik [2003] demonstrated that three different fisheye distortion algorithms all outperformed panning, zooming, and an overview+detail view. Second, when the task involves selecting two points that can only be selected at a high scale-ratio, then multiple fisheye views outperform interfaces that only permit panning at a constant high scale, or which only permit zooming [Shoemaker and Gutwin 2007]. The study did not include evaluation of the ability to use multiple overview+detail lenses.

7.1.3. Interface Mechanics of Visual Search. Searching for a target is more demanding than mechanically acquiring one that is clearly demarcated, and consequently most evaluations involving searching are presented in the following section on high-level evaluations. However, in aiming to create objective criteria for assessing visual search support, Pietriga et al. [2007] developed a simple, mechanical methodology for assessing the efficiency of interfaces for multiscale search using a grid of nine candidate rectangular targets to find the unique target with rounded, rather than square, corners. The objects with round corners are only discriminable at a high scale. They tested their methodology by comparing performance across four interface types: pan and zoom, fisheye distortion lens, DragMag [Ware and Lewis 1995] overview+detail lens, and an overview+detail system that supported panning and zooming in both views in a manner similar to Google Maps. Results showed that the overview+detail with pan and zoom interface was fastest, followed by the DragMag overview+detail interface, then the fisheye lens, and the pan and zoom interface last.

This work is particularly interesting because it explicitly aims to remove the cognitive skills that are normally involved in resolving search tasks, addressing instead motor and perceptual skills. The results are also interesting because they clearly show that in this context, performance is best when overview+detail facilities are combined with zooming.

7.2. High-Level Evaluations

While the low-level evaluations focus on the mechanics of interface manipulation, the high-level evaluations are diverse, examining a wide range of task domains, as summarized in Table II.

7.2.1. Single Versus Multiple Foci. Schaffer et al. [1996] conducted the first high-level empirical study of focus+context techniques, comparing fisheye and full-zoom interfaces for navigation through two-dimensional graphical networks. The participants' tasks involved finding and fixing broken network connections in hierarchically organized network diagrams. The fisheye view allowed multiple focal and contextual regions, while the zooming condition did not. Participants completed tasks more quickly and with fewer navigation actions when using the fisheye interface, but unfortunately, it remains unclear what caused the improved performance, whether the fisheye or the multiple foci. Similar concerns exist for Shoemaker and Gutwin's [2007] positive results for multiple fisheye lenses: Their control interfaces did not allow multiple lenses. An experiment by Plumlee and Ware [2002] highlights these concerns by showing that interfaces supporting multiple overview+detail views outperform zooming interfaces when demands on visual memory are high.

Table II. High-Level Evaluations Including Comprehension of Information Spaces

Author and Year	O+D	Zoom	F+C	Description
North and Shneiderman 2000	↔			Navigating textual census data. O+D interfaces work best when actions in the overview and detail views are coordinated. Coordinated views outperform detail alone.
Beard and Walker 1990	←	→		2D text-tree navigation. Overview allowed faster navigation than scrollbar with resizing. Primarily, a test of O+D versus traditional scrolling.
Ghosh and Shneiderman 1999	←	→		Medical histories. The O+D interface to Lifelines [Plaisant et al. 1996] allowed tasks to be completed more quickly than a zooming interface.
Plumlee and Ware 2002	←	→		Abstract graphical task. Evaluation confirms a theory that zooming outperforms multiple O+D views when demands on visual memory are low, but the inverse when demands are high.
Hornbaek et al. 2002	←	→		Map navigation. Tests user performance using a zooming interface with and without an overview. Finds some tasks are faster without the overview due to the cost of assimilating data.
Hornbaek and Frokjaer 2003	↔		↔	Reading comprehension. Compared linear text with O+D and fisheye text interfaces. Comprehension highest with O+D, lowest with fisheye. Fisheye had fastest reading, O+D slowest.
Baudisch et al. 2004	↔		↔	Web browsers. Evaluates a cue-enhanced fisheye Web browser with and without an overview. Overview and fisheye performed similarly. Overview was popular; fisheye less so.
Schaffer et al. 1996		←	→	Graph editing. Compared hierarchical zooming with a continuous fisheye. Fisheye supported faster task completion.
Donskoy and Kaptelinin 1997		←	→	Iconic file identification and drag-and-drop manipulation. Zooming was faster and preferred to scrolling. Fisheye was slowest and least preferred. Failed to demonstrate the value of animation.
Gutwin and Fedak 2004		←	→	Various tasks on mobile devices. Panning vs. two-level zoom vs. fisheye. Fisheye fastest for one task, zooming fastest for another task. Panning slowest for all.
Büring et al. 2006		←	→	Searching scatterplots on a mobile device. Zooming vs. fisheye distortion. No speed benefit was found for either technique, but fisheye was significantly preferred.
Baudisch et al. 2002.	←	—	→	Static graph navigation and dynamic driving simulation tasks. A mixed-resolution focus+context display outperforms overview+detail screens and zooming.
Nekrasovski et al. 2006.	←	—	→	Hierarchical navigation. A pan and zoom was faster than a "rubber-sheet" fisheye technique. Overview provided no benefit, but had some strong subjective ratings.
Pirolli et al. 2001; Plaisant et al. 2002; Pirolli et al. 2003			↔	Hierarchical navigation. Search strategies and performance when using the hyperbolic tree browser and SpaceTree compared to Windows Explorer. Results depend on task characteristics such as information scent and revisitation.
Zanella et al. 2002			↔	Map interpretation. Grids help users interpret the effect of fisheye distortion on spatial layout. Shading is less effective.
Cockburn and Smith 2003; Jakobsen and Hornbaek 2006			↔	Source-code navigation. Investigates the value of manual elision and DOI fisheye techniques over a traditional linear representation of code. Fisheye techniques were generally no worse and often better than flat code listings.

Table II. *Continued.*

Author and Year	O+D	Zoom	F+C	Description
Bederson et al. 2004			⟷	Mobile calendars. Fisheye calendar allows complex tasks to be completed more quickly. Fisheye was found easier to use than a traditional alternative for most tasks, but somewhat less preferred overall.
Bederson and Boltman 1999		⟷		Spatial memory. Using animation in zooming helps users form a spatial model of the information space.

7.2.2. Understanding Overview+Detail. In directly comparing overview+detail and zooming interfaces, Ghosh and Shneiderman [1999] analyzed task completion times for tasks that involved extracting medical information from two versions of the Lifelines system [Plaisant et al. 1996]. The overview+detail interface allowed tasks to be completed more quickly. However, it is now common for desktop applications to include both overview+detail and zooming capabilities. To better understand the contribution of each of these components to interaction, Hornbaek et al. [2002] evaluated user performance in map navigation tasks when using a zooming interface that either had or did not have an additional overview+detail region. They also controlled whether the zooming functions did or did not support semantic zooming; when semantic zooming was on, the labels in the map were tailored to provide appropriate detail for that zoom level (similar to Google Maps), but when semantic zooming was off the labels were only legible when the map was zoomed-in. Surprisingly, and contradicting Pietriga et al.'s [2007] analysis of low-level interaction, their results showed that overview+detail regions increased task completion times when semantic zooming was enabled, and they suggest that this cost is due to the overview being made redundant by the semantic detail. When semantic zooming was disabled there was no difference between performance with overview+detail and zooming interfaces. The participants preferred the interface with the overview despite their slower performance, stating that it helped them orient themselves within the information space.

7.2.3. Impact on Spatial Memory. The study by Hornbaek et al. [2002] also produced results on the impact of overview+detail interfaces on spatial recollection, finding that recall was better after using the nonoverview interface. With zooming interfaces, Bederson and Boltman [1999] showed that animated zoom effects helped users form a spatial model of their information space. This result supports those in nonzooming domains, which also show that animation helps users comprehend display transitions [Gonzalez 1996; Klein and Bederson 2005]. While animation aids comprehension, there is no evidence that it also aids task performance time. Chui and Dillon [1997] found no significant main effect of animation in a hierarchical file-browsing task, although they suggest that users gain benefits once accustomed to it. Similarly, Donskoy and Kaptelinin [1997] failed to show task-time benefits for animation when coupled with overview+detail, zooming, or fisheye views.

Spatial comprehension remains a concern for fisheye lenses that distort the information space [Carpendale and Cowperthwaite 1997]. Zanella et al. [2002] examined a variety of display enhancement schemes aimed at enhancing users' spatial comprehension of distorted space, concluding that simple parallel grid lines best support comprehension.

7.2.4. Reading, Editing, and Text Documents. Hornbaek and Frokjaer [2003] compared reading patterns and usability issues associated with three forms of electronic document presentation: traditional "flat" text, an overview+detail interface that enhanced the flat-text view with thumbnails of each page on the lefthand edge of the window, and a fisheye text view that displayed text regions in different font sizes. In the fisheye view, headings and the first and last paragraphs of each section were continually displayed in a normal-sized font, while other paragraphs were diminished unless clicked on by the user. Their evaluation tasks involved reading scientific documents using the different forms of presentation and then either writing short summaries or answering questions about them. Their results showed that the fisheye view encouraged faster reading, but at the cost of diminished comprehension. The overview interface allowed participants to rapidly navigate through the document, and although they spent more time reading, they scored better in comprehension tests. As in other studies, the participants' preference rankings favored the overview+detail interface. These results are supported by the findings of Baudisch et al. [2004], who compared three forms of a cue-enhanced Web browser (a traditional linear view, a fisheye view, and an overview). Their participants generally preferred the overview+detail interface, while the fisheye view polarized opinions. Their performance data suggested that the benefits of fisheye views are strongly task dependent.

7.2.5. Computer Program Navigation. Source code is a specialized class of structured text that has long been considered an appropriate domain for the application of focus+context techniques [Furnas 1986]. Many commercial and open-source development environments now allow users to collapse/restore the display of methods (e.g., Netbeans, as shown in Figure 9), yet until recently there was no empirical evidence of the effectiveness of this feature. Cockburn and Smith [2003] compared user performance in source code navigation tasks using traditional linear scrollable listings with two fisheye-based implementations that miniaturized method contents. For most tasks, the reduced scrolling required with the fisheye views resulted in faster task completion, but the study also raised questions about the generalizabilty of the results to more complex tasks in which users would have to continually manage the open/closed states of methods. Jakobsen and Hornbaek [2006] avoided this last issue by developing a fisheye source code viewer that used a DOI function to automatically display/suppress code blocks based on the user's focal region. The authors compared their fisheye viewer to a standard linear list view in performing basic and advanced code navigation tasks. They not only found the fisheye viewer to support significantly faster task times overall, but that it was also preferred to the linear listing. While many studies have found fisheye implementations to be well liked by participants, this work stands out as one of the few for which a fisheye technique offers users a performance benefit over the traditional alternative.

7.2.6. Searching and Tree Navigation Studies. Section 7.1.3 described Pietriga et al.'s [2007] recent work on operationalizing the evaluation of multiscale interfaces for search. Their work explicitly eliminates cognitive task components, yet it is reasonable to suspect that in real use, the cognitive aspects of tasks may cause interaction effects between task type and interface type (examples follow).

Beard and Walker [1990] conducted the first empirical comparison of overview+detail and zooming techniques for hierarchical navigation; they also included a scrolling interface as an experimental control. The overview+detail and zooming interfaces both outperformed scrolling alone.

Interest in the efficiency of focus+context interfaces for search was stimulated by the success of the hyperbolic tree browser [Lamping et al. 1995] during a "browse-off" panel session at ACM CHI'97 [Mullet et al. 1997] when a team using it outperformed five other teams using competing browsers. Previous studies, however, had failed to show significant advantages [Lamping et al. 1995; Czerwinski and Larson 1997]. Pirolli et al. [2003, 2001] ran a series of studies to scrutinize interaction with the hyperbolic tree browser (see Figure 7) in comparison to the standard file browser of Windows Explorer. Their studies demonstrated interaction effects between task and interface factors, with the hyperbolic tree performing well on tasks with strong scent (i.e., when nodes provided strong cues to the information content of further outlying nodes), but poorly when scent was weak. Through analysis of eye-tracking data, they demonstrate that the hyperbolic browser's slower performance in low scent conditions is due to impaired support for visual search in the high-density peripheral views.

Plaisant et al. [2002] used the CHI'97 browse-off dataset to compare performance in tree searching tasks, using Windows Explorer, the hyperbolic tree browser, and their own focus+context SpaceTree system. Again, their results varied across tasks, with Windows Explorer outperforming the others for tasks that benefited from multiple open branches (e.g., revisiting previously opened branches), and SpaceTree performing well for certain types of topological overview tasks (e.g., viewing ancestors).

Nekrasovski et al. [2006] recently compared performance in hierarchical tree browsing tasks using zooming and distortion-oriented focus+context techniques, both with and without an overview. Contrary to their predictions, the pan-and-zoom technique was significantly faster than focus+context. Although the presence of an overview did not impact task times, subjective measures of physical demand and enjoyment favored an overview; these findings are in agreement with the findings of Hornbaek et al. [2002]. Donskoy and Kaptelinin [1997] also showed that zooming outperformed a fisheye view for tasks involving identification and drag-and-drop manipulation of file icons.

In contrast, the results of Büring et al. [2006] were more positive for fisheyes. They compared pan-and-zoom to a fisheye view for searching scatterplot graphs on a mobile device, finding no significant difference in task times, but subjective results strongly favored the fisheye view. The authors note that the preference for fisheyes contradicts previous studies [Donskoy and Kaptelinin 1997; Hornbaek and Frokjaer 2003; Baudisch et al. 2004; Gutwin and Fedak 2004], but they suggest this may be due to the nature of their tasks, which did not require relative spatial judgments.

There have also been several evaluations of cue-based interfaces. Performance benefits have been demonstrated for cue-based techniques that visually emphasize items [Suh et al. 2002; Baudisch et al. 2004; Lam and Baudisch 2005] or group them [Dumais et al. 2001] in response to search criteria.

7.2.7. Other Evidence of Task Effects in Cross-Interface Evaluations. The task-specific performance merits of fisheye views are also echoed in two studies on the use of fisheye techniques on mobile devices. First, Gutwin and Fedak [2004] compared a single-focus fisheye interface with two-level zoom and panning interfaces for three tasks: image editing, Web browsing, and network monitoring. The fisheye best supported navigation, the zooming interface best supported monitoring, and traditional panning was slowest for all tasks. The fisheye problems with moving targets may explain zooming's success in the monitoring task. However, the authors' limited support for zooming (two levels only) may explain why it was outperformed by the fisheye in the navigation task. Second, an evaluation of the fisheye calendar DateLens (described Section 5.2.3) in comparison to a standard commercial calendar for PDAs (Microsoft Pocket PC 2002) showed that the fisheye allowed improved performance of complex tasks, but little performance difference for simple ones [Bederson et al. 2004b]. And like other studies, user support of the fisheye distortion was mixed; on a task-by-task basis, the fisheye approach was more often found easier to use than the traditional interface, yet the latter was ultimately preferred by a slight majority of users. It remains unclear whether the modest subjective preference ratings for fisheye views are due to their novelty or due to an enduring distaste for their behavior.

7.2.8. Wide Field-of-View Systems. There has been extensive design and evaluation research based on solving and easing low-level interaction problems, such as selection, that arise when using multimonitor and wall-sized displays. Most of this work does not specifically address issues of working with both focused and contextual views. Importantly, however, Czerwinski et al. [2002] have noted the potential benefits for women of working with wide field-of-view systems, and Robertson et al. [2005] provided a review of the user experience of working with large displays.

Baudisch et al. [2002] provided the only quantitative study (to our knowledge) of focus and context issues with wide field-of-view systems. In the first of two experiments they compared user performance using three interfaces (zooming, overview+detail, and their mixed-resolution focus+context display) in static graph navigation tasks using a map of London and a circuit diagram. Their overview+detail condition was produced by using two side-by-side monitors. The focus+context screen was faster (by up to 56%) in all tasks, and they attributed the difference to the costs of switching between views when using overview+detail or zooming. The second experiment involved a dynamic visualization (a driving simulator) using the focus+context screen or the overview+detail display. Again, the split attention costs of the overview+detail screens resulted in significantly worse performance.

8. SUMMARY

We have presented four interface approaches that allow users to attain both focused and contextual views of their information spaces. Overview+detail systems allow concurrent focus and context views that are spatially segregated in the x, y, or z dimensions. Zooming systems allow the entire display space to be dedicated to either focused or contextual views by temporally segregating their display. Focus+context systems present both focus and context within a single display. This is achieved through a variety of techniques, including selectively removing, diminishing, or magnifying items, and various displays supporting a wide field of view. Finally, cue-based systems modify the display of items to highlight or suppress them, dependent on context or search criteria. Many systems support combinations of these techniques.

None of these approaches is ideal. Spatial separation demands that users assimilate the relationship between the concurrent displays of focus and context. Evidence that this assimilation process hinders interaction is apparent in the experiments of Hornbaek et al. [2002], who note that "switching between the detail and the overview window required mental effort and time." Baudisch et al. [2002] made similar observations of the costs associated with split attention with overview+detail interfaces. The temporal separation of zooming also demands assimilation between pre- and post-zoom states. This difficulty was observed by Cockburn and Savage [2003] in their evaluations, who note that "the abrupt transitions between discrete zooming levels . . . meant that the participants had to reorient themselves with each zoom action." Animation can ease this problem [Bederson and Boltman 1999], but it cannot be removed completely. Baudisch et al. [2002] describe how some users rapidly completed tasks using a zooming interface by first zooming out and memorizing all target locations, but the burden of memorization is likely to be ineffective for many tasks. Finally, many focus+context systems distort the information space, which is likely to damage the user's ability to correctly assess spatial properties such as directions, distances, and scales. Even in nonspatial domains, evaluations of distortion-based techniques have largely failed to provide evidence of performance enhancements, and several studies have shown that fisheye views can damage fundamental components of interaction such as target acquisition.

Regardless of the usability challenges raised by each of the techniques, overview+detail and zooming interfaces are now standard components of many desktop graphical user interfaces. The usability benefits they can provide often outweigh their costs, as demonstrated by several evaluations. For example, Hornbaek and Frokjaer [2003] showed that when reading electronic documents, both overview+detail and fisheye interfaces offer performance advantages over traditional flat text interfaces. Although rare, fisheye views are starting to appear in desktop components such as the Mac OS X Dock. In this case the usability benefits are purely cosmetic, yet many users are happy to trade-off efficiency for fashion and design luster, which is not surprising, since aesthetics and enjoyment are essential aspects of the computer experience [Norman 2004].

It therefore seems clear that supporting both focused and contextual views can improve interaction over constrained single-view software. The question then becomes which style of interface, or which combination of styles, offers the greatest performance advantages, for what tasks, and under what conditions? The current state of research fails to provide clear guidelines, despite a recent surge in empirical analysis of the techniques' effectiveness. Results to date have revealed that the efficiency of different techniques is dependent on many factors, particularly the nature of the users' task. For example, Hornbaek et al. [2002] showed that fisheye text views allowed documents to be read faster than overview+detail, but that participants understood documents better with overview+detail; Schaffer et al. [1996] showed that hierarchical fisheye views outperformed full-zoom techniques, but it remains unclear whether an alternative implementation of zooming (continuous rather than full) would have defeated fisheye views; Bederson et al. [2004b] showed that a table-based fisheye interface for calendars on PDAs outperformed a standard interface for complex tasks, but for simpler tasks, the normal interface was more efficient and preferred.

We offer the following concluding comments and recommendations, with the cautionary note that it is important for designers to use their own judgment as to which approach to pursue. We also encourage researchers to continue to empirically analyze these techniques to solidify our understanding of the trade-offs inherent in the different approaches.

Overview+Detail. Several studies have found that overview+detail interfaces are preferred over other techniques. For particular tasks such as document comprehension, no alternative has been found more effective. Notable disadvantages of overview+detail are the additional use of screen real estate (which may be more effectively used for details) and the mental effort and time required to integrate the distinct views. The real estate issue can often be addressed by offering users control over whether to display the overview. Even so, the time costs of this technique may make it suboptimal for dynamic activities, as shown by Baudisch et al.'s [2002] driving simulation. For consistency with state-of-the-art interfaces, users should be able to browse the overview without influencing the detail view, but changes in the detail view should be immediately reflected in the overview.

Zooming. Temporal separation of views can easily create substantial cognitive load for users in assimilating the relationship between pre- and post-zoom states; zooming is easy to do badly, as indicated by many studies in which it has performed poorly. Animating the transition between zoom levels can dramatically reduce the cognitive load (but fine-tuning the duration of the animation is important). There is also evidence that users can optimize their performance with zooming interfaces when concurrent, unimanual controls for pan and zoom are supported.

Focus+Context. There are several distinct strategies for displaying the focal region within its surrounding context: removal of nonfocal items from the display, distortion to alter the scale of items according to a calculated interest value, and specialized wide field-of-view displays. Most of the empirical research on focus+context interfaces has investigated distortion-oriented visualizations. Results in favor of these techniques have been produced for tasks that involve gaining a rapid overview of the data space or quickly following graph representations that have a clear structure. However, distortion-oriented displays are likely to impair the user's ability to make relative spatial judgments, and they can cause target acquisition problems. Other empirical work on wide-screen displays suggests that large high-resolution displays will help users employ their natural capabilities for focused visual attention and peripheral vision.

ACKNOWLEDGMENTS

Many thanks to the anonymous reviewers for their extremely detailed and constructive comments. This work was partially funded by New Zealand Royal Society Marsden Grant 07-UOC-013.

REFERENCES

ABOELSAADAT, W. AND BALAKRISHNAN, R. 2004. An empirical comparison of transparency on one and two layer displays. In *Proceedings of the British HCI Conference, People and Computers XVIII*, 53–67.

APPERT, C. AND FEKETE, J. 2006. OrthoZoom scroller: 1D mutli-scale navigation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 21–30.

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 *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 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 *Proceedings of the Conference on Advanced Visual Interfaces (AVI)*. ACM Press, 133–140.

BAUDISCH, P. AND ROSENHOLTZ, R. 2003. Halo: A technique for visualizing off-screen locations. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 481–488.

BEARD, D. AND WALKER, J. 1990. Navigational techniques to improve the display of large two-dimensional spaces. *Behav. Inf. Technol. 9*, 6, 451–466.

BEDERSON, B. 2000. Fisheye menus. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 217–225.

BEDERSON, B. 2001. PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, ACM Press, 71–80.

BEDERSON, B. AND BOLTMAN, A. 1999. Does animation help users build mental maps of spatial information? In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*. IEEE Computer Society, 28–35.

BEDERSON, B., GROSJEAN, J., AND MEYER, J. 2004a. Toolkit design for interactive structured graphics. *IEEE Trans. Softw. Eng. 30*, 8, 535–546.

BEDERSON, B., HOLLAN, J., PERLIN, K., MEYER, J., BACON, D., AND FURNAS, G. 1996. Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *J. Visual Lang. Comput. 7*, 1, 3–31.

BEDERSON, B. B., CLAMAGE, A., CZERWINSKI, M. P., AND ROBERTSON, G. G. 2004b. DateLens: A fisheye calendar interface for PDAs. *ACM Trans. Comput.-Hum. Interac. 11*, 1, 90–119.

BEDERSON, B. B., MEYER, J., AND GOOD, L. 2000. Jazz: An extensible zoomable user interface graphics toolkit in Java. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 171–180.

BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. 1993. Toolglass and magic lenses: The see-through interface. In *Proceedings of the International Conference on Computer Graphics an Interactive Techniques. ACM SIGGRAPH*, ACM Press, 73–80.

BOURGEOIS, F. AND GUIARD, Y. 2002. Multiscale pointing: Facilitating pan-zoom coordination. In *Extended Abstracts of the ACM Conference on Human Factors in Computer Systems (CHI)*, 758–759.

BÜRING, T., GERKEN, J., AND REITERER, H. 2006. User interaction with scatterplots on small screens—A comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Trans. Visual. Comput. Graph. 12*, 5, 829–836.

BYRD, D. 1999. A scrollbar-based visualization for document navigation. In *Proceedings of the Conference on Digital Libraries*, 122–129.

CARD, S., MACKINLAY, J., AND SHNEIDERMAN, B. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan-Kaufmann.

CARD, S. K., ROBERTSON, G. G., AND MACKINLAY, J. D. 1991. The information visualizer, an information workspace. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 181–186.

CARPENDALE, M. AND COWPERTHWAITE, D. 1997. Making distortions comprehensible. In *Proceedings of the IEEE Symposium on Visual Languages (VL)*, IEEE, 36–45.

CARPENDALE, M. AND MONTAGNESE, C. 2001. A framework for unifying presentation space. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 61–70.

CHEN, C. 2004. *Information Visualization: Beyond the Horizon*. Springer, London.

CHIMERA, R. 1992. ValueBars: An information visualization and navigation tool. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 293–294.

CHUI, M. AND DILLON, A. 1997. Who's zooming whom? Attunement to animation in the interface. *J. Amer. Soc. Inf. Sci. 48*, 11, 1067–1072.

COCKBURN, A., GUTWIN, C., AND ALEXANDER, J. 2006. Faster document navigation with space-filling thumbnails. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 1–10.

COCKBURN, A. AND SAVAGE, J. 2003. Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods. In *Proceedings of the British Computer Society Conference on Human-Computer Interaction, People and Computers XVII*, 87–102.

COCKBURN, A., SAVAGE, J., AND WALLACE, A. 2005. Tuning and testing scrolling interfaces that automatically zoom. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 71–80.

CZERWINSKI, M. AND LARSON, K. 1997. The new Web browsers: They're cool but are they useful? In *Proceedings of the British HCI Conference, People and Computers XII*. Springer.

CZERWINSKI, M., TAN, D., AND ROBERTSON, G. 2002. Women take a wider view. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 195–202.

DONSKOY, M. AND KAPTELININ, V. 1997. Window navigation with and without animation: A comparison of scroll bars, zoom, and fisheye view. In *the Companion Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 279–280.

DRUIN, A., STEWART, J., PROFT, D., BEDERSON, B., AND HOLLAN, J. 1997. KidPad: A design collaboration between children, technologists, and educators. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 463–470.

DUMAIS, S., CUTRELL, E., AND CHEN, H. 2001. Optimizing search by showing results in context. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 277–284.

FITTS, P. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exper. Psychol. 47*, 381–391.

FURNAS, G. 1986. Generalized fisheye views. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 16–23.

FURNAS, G. 1997. Effective view navigation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 367–374.

FURNAS, G. 2006. A fisheye follow-up: Further reflections on focus+context. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 999–1008.

FURNAS, G. AND ZHANG, X. 1998. MuSE: A multiscale editor. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 107–116.

FURNAS, G. W. AND BEDERSON, B. B. 1995. Space-Scale diagrams: Understanding multiscale interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 234–241.

GHOSH, P. AND SHNEIDERMAN, B. 1999. Zoom-Only vs. overview-detail pair: A study in browsing techniques as applied to patient histories. Tech. Rep., 99–12, HCIL, University of Maryland, College Park.

GONZALEZ, C. 1996. Does animation in user interfaces improve decision making? In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 27–34.

GRUDIN, J. 2001. Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. In *Proceedings of the ACM Conference on human factors in computing systems (CHI)*, 458–465.

GUIARD, Y. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *J. Motor Beha. 19*, 4, 486–517.

GUIARD, Y., BEAUDOUIN-LAFON, M., BASTIN, J., PASVEER, D., AND ZHAI, S. 2004. View size and pointing difficulty in multi-scale navigation. In *Proceedings of the ACM Working Conference on Advanced Visual Interfaces (AVI)*. ACM Press, 117–124.

GUIARD, Y., BOURGEOIS, F., MOTTET, D., AND BEAUDOUIN-LAFON, M. 2001. Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. In *Proceedings of the IHM-HCI Conference, People and Computers XV*, 573–587.

GUIARD, Y., DU, Y., AND CHAPUIS, O. 2007. Quantifying degree of goal directedness in document navigation: Application to the evaluation of the perspective-drag technique. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 327–336.

GUIMBRETIERE, F., STONE, M., AND WINOGRAD, T. 2001. Fluid interaction with high-resolution wall-size displays. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 21–30.

GUTWIN, C. 2002. Improving focus targeting in interactive fisheye views. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 267–274.

GUTWIN, C. AND FEDAK, C. 2004. Interacting with big interfaces on small screens: A comparison of fisheye, zoom, and panning techniques. In *Proceedings of the Graphics Interface Conference.* Canadian Human-Computer Communications Society, 213–220.

GUTWIN, C., ROSEMAN, M., AND GREENBERG, S. 1996. A usability study of awareness widgets in a shared workspace groupware system. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, 258–267.

GUTWIN, C. AND SKOPIK, A. 2003. Fisheye views are good for large steering tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 201–208.

HERMAN, I., MELANCON, G., AND MARSHALL, M. 2000. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. Visual. Comput. Graph. 6*, 1, 24–43.

HILL, W., HOLLAN, J., WROBLEWSKI, D., AND MCCANDLESS, T. 1992. Edit wear and read wear. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 3–9.

HORNBAEK, K., BEDERSON, B., AND PLAISANT, C. 2002. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact. 9*, 4, 362–389.

HORNBAEK, K. AND FROKJAER, E. 2003. Reading patterns and usability in visualizations of electronic documents. *ACM Trans. Comput. Hum. Interact. 10*, 2, 119–149.

HORNBAEK, K. AND HERTZUM, M. 2007. Untangling the usability of fisheye menus. *ACM Trans. Comput. Hum. Interact. 14*, 2, 6.

IGARASHI, T. AND HINCKLEY, K. 2000. Speed-Dependent automatic zooming for browsing large documents. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 139–148.

JAKOBSEN, M. AND HORNBAEK, K. 2006. Evaluating a fisheye view of source code. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 377–386.

JUL, S. AND FURNAS, G. 1998. Critical zones in desert fog: Aids to multiscale navigation. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 97–106.

KEAHEY, T. 1998. The generalized detail-in-context problem. In *Proceedings of the IEEE Symposium on Information Visualization*. IEEE Press, 44–51.

KERSTEN, M. AND MURPHY, G. 2005. Mylar: A degree-of-interest model for IDEs. In *Proceedings of the 4th International Conference on Apect-Oriented Software Development*. ACM Press, 159–168.

KLEIN, C. AND BEDERSON, B. 2005. Benefits of animated scrolling. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 1965–1968.

KOSARA, R., MIKSCH, S., AND HAUSER, H. 2001. Semantic depth of field. In *Proceedings of the IEEE Computer Graphics and Applications, Symposium on Information Visualization (InfoVis)*, IEEE Computer Society Press, 97–104.

KOSARA, R., MIKSCH, S., AND HAUSER, H. 2002. Focus+Context taken literally. *IEEE Comput. Graph. Appl. 22*, 1, 22–29.

LAM, H. AND BAUDISCH, P. 2005. Summary thumbnails: Readable overviews for small screen Web browsers. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 681–690.

LAMPING, J., RAO, R., AND PIROLLI, P. 1995. A focus+context technique based on hyperbolic geometry for visualising large hierarchies. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 401–408.

LEUNG, Y. AND APPERLEY, M. 1994. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput. Hum.-Interact. 1*, 2, 126–160.

MACKAY, W., APPERT, C., BEAUDOUIN-LAFON, M., CHAPUIS, O., DU, Y., AND FEKETE, J. 2007. Touchstone: Exploratory design of experiments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 1425–1434.

MACKINLAY, J., CARD, S., AND ROBERTSON, G. 1990. Rapid controlled movement through a virtual 3D workspace. In *Proceedings of the ACM SIGGRAPH International Conference on Computer Graphics and Interactive Techniques.* ACM Press, 171–176.

MACKINLAY, J., ROBERTSON, G., AND CARD, S. 1991. Perspective wall: Detail and Context Smoothly Integrated. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 173–179.

MCGUFFIN, M. AND BALAKRISHNAN, R. 2002. Acquisition of expanding targets. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 57–64.

MULLET, K., FRY, C., AND SCHIANO, D. 1997. On your marks, get set browse! In *Extended Abstracts of the ACM Conference on Human Factors in Computer Systems (CHI)*, 113–114.

MUNZNER, T., GUIMBRETIERE, F., TASIRAN, S., ZHANG, L., AND ZHOU, Y. 2003. TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph. 22*, 3, 453–462.

NEKRASOVSKI, D., BODNAR, A., MCGRENERE, J., GUIMBRETIERE, F., AND MUNZNER, T. 2006. An evaluation of pan & zoom and rubber sheet navigation with and without an overview. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 11–20.

NORMAN, D. 2004. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Books, New York.

PATRICK, E., COSGROVE, D., SLAVKOVIC, A., RODE, J., VERATTI, T., AND CHISELKO, G. 2000. Using a large projection screen as an alternative to head-mounted displays for virtual environments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 478–485.

PERLIN, K. AND FOX, D. 1993. Pad: An alternative approach to the computer interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 57–64.

PIETRIGA, E. 2005. A toolkit for addressing HCI issues in visual language environments. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL-HCC)*, 145–152.

PIETRIGA, E., APPERT, C., AND BEAUDOUIN-LAFON, M. 2007. Pointing and beyond: An operationalization and preliminary evaluation of multi-scale searching. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 1215–1224.

PIROLLI, P., CARD, S., AND VAN DER WEGE, M. 2001. Visual information foraging in a focus+context visualization. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 506–513.

PIROLLI, P., CARD, S., AND VAN DER WEGE, M. 2003. The effects of information scent on visual search in the hyperbolic tree browser. *ACM Trans. Comput.-Hum. Interact. 10*, 1, 20–53.

PLAISANT, C., CARR, D., AND SHNEIDERMAN, B. 1995. Image-Browser taxonomy and guidelines for designers. *IEEE Softw. 12*, 2, 21–32.

PLAISANT, C., GROSJEAN, J., AND BEDERSON, B. 2002. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, 57–64.

PLAISANT, C., MILASH, B., ROSE, A., WIDOFF, S., AND SHNEIDERMAN, B. 1996. LifeLines: Visualizing Personal Histories. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 221–227.

PLUMLEE, M. AND WARE, C. 2002. Modeling performance for zooming vs. multi-window interfaces based on visual working memory. In *Proceedings of the ACM Working Conference on Advanced Visual Interface (AVI)*, 59–68.

RAMOS, G. AND BALAKRISHNAN, R. 2005. Zliding: Fluid zooming and sliding for high precision parameter manipulation. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 143–152.

RAO, R. AND CARD, S. K. 1994. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 318–322.

ROBERTS, J. 2007. State of the art: Coordinated and multiple views in exploratory visualization. In *Coordinated and Multiple Views in Exploratory Visualization*, ETH, Switzerland, IEEE Press, 61–71.

ROBERTSON, G., CZERWINSKI, M., BAUDISCH, P., MEYERS, B., ROBBINS, D., SMITH, G., AND TAN, D. 2005. The large-display user experience. *IEEE Comput. Graph. Appl. 25*, 4, 44–51.

ROBERTSON, G. G. AND MACKINLAY, J. D. 1993. The document lens. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 101–108.

SARKAR, M. AND BROWN, M. 1992. Graphical fisheye views of graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 83–91.

SARKAR, M., SNIBBE, S., AND REISS, S. 1993. Stretching the rubber sheet: A metaphor for visualising large structure on small screen. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM Press, 81–91.

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. Comput. Hum-Interact. 3*, 2, 162–188.

SHOEMAKER, G. AND GUTWIN, C. 2007. Supporting multi-point interaction in visual workspaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 999–1008.

SPENCE, R. 2007. *Information Visualization*. Pearson Education Limited, Harlow.

SPENCE, R. AND APPERLEY, M. 1982. Database navigation: An office enironment for the professional. *Behav. Inf. Technol. 1*, 1, 43–54.

SUH, B., WOODRUFF, A., ROSENHOLTZ, R., AND GLASS, A. 2002. Popout prism: Adding perceptual principles to overview+detail document interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 251–258.

SUMMERS, K., GOLDSMITH, T., KUBICA, S., AND CAUDELL, T. 2003. An empirical evaluation of continuous semantic zooming in program visualization. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, 155–162.

TAN, D., ROBERTSON, G., AND CZERWINSKI, M. 2001. Exploring 3D navigation: Combining speed-coupled flying with orbiting. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 418–425.

TOBLER, W. 1973. A continuous transformation useful for districting. *Annals New York Acad. Sci. 219*, 215–220.

TVERSKY, B., MORRISON, J., AND BETRANCOURT, M. 2002. Animation: Can it facilitate? *Int. J. Hum-Comput. Studies 57*, 247–262.

VAN WIJK, J. AND NUIJ, W. 2004. A model for smooth viewing and navigation of large 2D information spaces. *IEEE Trans. Visual. Comput. Graph. 10*, 4, 447–458.

WARE, C. 2004. *Information Visualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA.

WARE, C. AND FLEET, D. 1997. Context sensitive flying interface. In *Proceedings of the Symposium on Interactive 3D Graphics*, 127–130.

WARE, C. AND LEWIS, M. 1995. The DragMag image magnifier. In *Companion Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 407–408.

WOODSON, W. AND CONOVER, D. 1964. *Human Engineering Guide for Equipment Designers*. University of California Press, Berkeley, CA.

ZANELLA, A., CARPENDALE, S., AND ROUNDING, M. 2002. On the effects of viewing cues in comprehending distortions. In *Proceedings of the Nordic Conference on Human-Computer Interaction (NordiCHI'02)*, 119–128.

ZELLWEGER, P., MACKINLAY, J., GOOD, L., STEFIK, M., AND BAUDISCH, P. 2002. City lights: Contextual views in minimal space. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI)*, 838–839.

ZHAI, S., CONVERSY, S., BEAUDOUIN-LAFON, M., AND GUIARD, Y. 2003. Human on-line response to target expansion. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 177–184.

Received April 2006; revised July 2007; accepted December 2007