

*Zuerst ersch. in : Handbook of research on user interface design and evaluation for mobile Technology ; 1 / Joanna Lumsden (ed.). - Hershey, PA : Information Science Reference, 2008. - S. 576-593. - ISBN 978-1-599-04871-0  
<http://dx.doi.org/10.4018/978-1-59904-871-0.ch034>*

# Navigation Support for Exploring Starfield Displays on Personal Digital Assistants

Thorsten Buring  
University of Konstanz, Germany  
*[buring@inf.uni-konstanz.de](mailto:buring@inf.uni-konstanz.de)*

# Navigation Support for Exploring Starfield Displays on Personal Digital Assistants

## **Abstract**

Due to advances in hardware technologies, mobile devices are increasingly capable of handling large-scale data sets. While this development broadens the application scope of smartphones and PDAs, it also means that high information loads must be displayed on very limited screen real estate. A solution to this problem may be provided by starfield displays. Starfield displays maximize the data-pixel ratio by presenting data inside a zoomable 2D scatterplot. However, a drawback is that once users have zoomed into the information space they tend to become lost, due to the clipping of orientation cues. The article summarizes the research results of recent projects that were conducted to improve the navigation and orientation features of starfield displays on small screens. Several approaches such as smooth zooming, overview window, and fisheye view have been implemented and user-tested. The results may support interface designers when targeting mobile devices.

# INTRODUCTION

The continuous advances in mobile hardware technology and the ubiquitous availability of wireless networks lead to novel application domains for smartphones and Personal Digital Assistants (PDAs). While these devices were originally developed for simple Personal Information Management (PIM) tasks, they may soon serve as a truly mobile alternative to notebooks in many business scenarios. Major software companies such as SAP and Oracle are already responding to this development by providing initial extensions for mobile devices. Equipped with a PDA, field sales staff can access and browse the company database to retrieve product information while being on the road.

A bottleneck that impedes the development of mobile applications, in particular for data retrieval, is the small screen size of the devices. Current mobile interfaces tend to rely on conventional list- or table-based representations that make inefficient use of screen space. Having retrieved a large number of search results, users are forced to scroll and flip pages to identify records they are interested in. Even for comparably small data sets as in the left view in Figure 1, this approach is tedious, slow and error-prone.

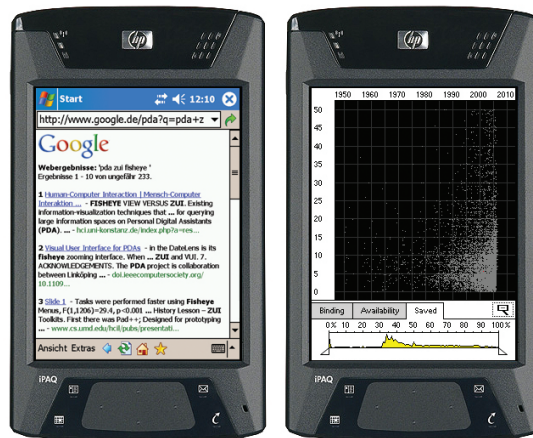


Figure 1: Comparing the data-pixel ratio of a list-based search engine output and a starfield display on a PDA.

A more promising interface strategy for small screens is that of starfield displays (Ahlberg & Shneiderman, 1994). Starfield displays are interactive 2D scatterplots that rely on space-preserving encodings rather than on textual representation of data. In the desktop world these interfaces have been found to improve the performance when searching in movie databases (Ahlberg & Shneiderman, 1994) and housing databases (Williamson & Shneiderman, 1992), or to aid drug discovery in commercial retrieval frameworks such as Spotfire (Ahlberg, 1996).

As shown in the right view in Figure 1, in a starfield display each data object is visualized by a small symbol that is spatially mapped against two scatterplot axes of data attribute dimensions. In this way, many thousands of items can be displayed on a single PDA screen, while users are provided with an effective overview that clearly reveals clusters, trends, and statistical outliers (Tufte, 1983). The problem with scatterplots, however, is that visual clutter from overlapping items is difficult to avoid. To prune visual clutter, starfield displays allow the users to explore the information space by moving between multiple representations and viewpoints (Dix & Ellis, 1998). This interaction is based on the metaphor of a zoomable user interface (ZUI). ZUIs follow the assumption that navigation in information spaces is best supported by tapping into our natural spatial and geographical ways of thinking (Perlin & Fox, 1993). In ZUIs, data objects are organized in space and scale. Users navigate this space by performing zooming (changing the scale) and panning (movement at constant scale) operations.

Developing effective starfield displays for pen-driven mobile devices is a challenging task. In this chapter we investigate two major aspects of the design. One is how to provide appropriate input mechanisms for controlling zoom and pan operations. On PDAs, primary input commands are limited to screen taps and a small set of hardware buttons. This places severe constraints on the interaction design. The other major design aspect deals with orientation inside the zoomable scatterplot. A general drawback with ZUIs is that, due to the clipping of orientation cues, users may easily lose their orientation once they have zoomed into the information space (Card, Mackinlay, & Shneiderman, 1999), (Hornbæk, Bederson, & Plaisant, 2002). This problem becomes increasingly troublesome as less display real estate is available, and is therefore particularly significant for mobile devices. To reduce orientation problems, the literature proposes three solutions (Spence, 2001): smooth zooming (Shneiderman, 1996)(B. Bederson & Meyer, 1998),(Shneiderman & Plaisant, 2005), overview+detail (Plaisant, Carr, & Shneiderman, 1995) and focus+context interfaces (Spence & Apperley, 1982). We implemented and adjusted these approaches to improve the navigation and orientation in starfield displays on PDAs. This chapter introduces each interface developed and summarizes the results of the user evaluations that were conducted to compare the solutions in terms of their usability. For an in-depth discussion of the evaluations see (Büring & Reiterer, 2005), (Büring, Gerken, & Reiterer, 2006a), (Büring, Gerken, & Reiterer, 2006b). The chapter concludes with an outlook on how to further improve starfield displays by incorporating visual cues for off-screen target locations and supporting advanced scaling interaction such as speed-dependent automatic zooming.

## 1 SMOOTH-ZOOMING

The first project is a proof-of-concept demonstration of a smooth-zooming starfield display for PDAs. Despite the potential benefits for visualizing data on small screens, there have been only a few attempts to port interactive scatterplots to mobile devices.

The first was the PalmMovieFinder (Dunlop & Davidson, 2000), which is based on the ideas of Ahlberg and Shneiderman's FilmFinder (Ahlberg & Shneiderman, 1994) and provides access to a collection of 71 movies. Exploration is limited to a simple two-level zoom in which a quarter of the display is magnified. More recent starfield applications are (Burigat & Chittaro, 2005) (Dunlop et al., 2004), which focus on map-based visualizations. Again, only discrete zoom steps are supported.

## 1.1 Interface

Our smooth-zooming prototype is implemented in Macromedia Flash and visualizes a movie database of 335 items. Having submitted a keyword query, retrieval results are represented by small rectangles inside a scatterplot (see Figure 2a). By default, the diagram dimensions are mapped to year of release (X) and lending frequency (Y). Other variables can be selected from the two color-coded combo boxes at the top of the interface. Alternative mappings are: popularity rating, section and language. Upon selecting a new mapping from a combo box, the scatterplot distribution and the axes labels are rapidly updated.

The prototype supports semantic and geometric zoom techniques to overcome the general drawbacks of scatterplot visualization, and to provide a work-around for those problems that apply specifically to scatterplot visualizations on small screens. Geometric zoom means that when zooming in on an object of interest (e.g. a street corner), the relevant map clipping simply becomes magnified. In contrast, an object that is zoomed semantically changes its visual representation depending on the amount of real estate available to it (Perlin & Fox, 1993).

As shown in the starfield display in Figure 1, large information spaces may cause data representations to become very small and thus hard to select accurately. Moreover, many data sets have an uneven distribution, causing the items to overlap and cluster together. Zooming reduces the information density and enables users to view a small clipping of the diagram surface in detail. To support orientation during the view shift, smooth scaling transitions have been proposed to help users to preserve their sense of position and context (Shneiderman, 1996).

Zooming in the prototype separates steering and scaling into two distinct actions. First, the users denote their current focus by tapping the screen with the stylus. A blue cross-hair appears. To zoom in on that position, users press and hold the upper button of the four-way PDA rocker switch. A smooth zoom animation is triggered, during which the rectangle items are continuously magnified and the view gradually shifts to center the focus point. The benefit of zooming in the scatterplot is that, at a sufficiently high magnification level, items can be accurately selected even if they are overlapped to a great extent by other items (see Figure 2 b). Users can stop the animation at any

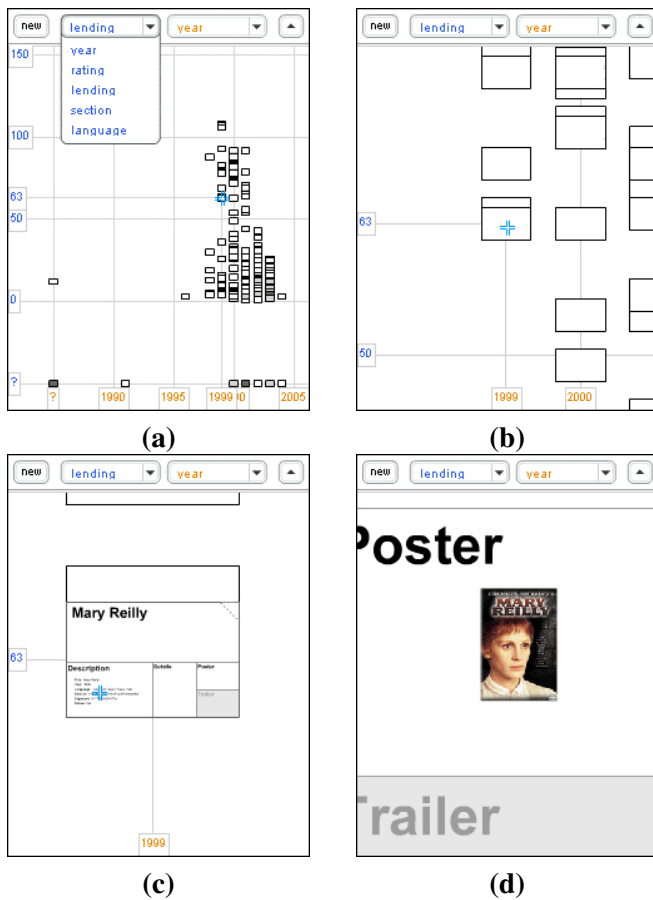


Figure 2: Smooth-zooming prototype: (a) default zoom-level, (b) geometric-zoom, (c) semantic-zoom, (d) subcontent-unit

time by releasing the button. Zooming out to the default scale is achieved by pressing and holding the lower rocker-switch button. To pan the view, users simply drag the diagram with the stylus.

While a scatterplot visualization is an effective tool for generating an overview of a large data set on limited screen real estate, it fails to provide detail information. In desktop applications, such information is often presented on-demand in a separate frame, but on mobile devices the screen real estate is too limited to reserve permanent space for a widget that is only temporarily of interest to the users. Another solution would be to use pop-ups, but this technique drags the user attention away from the diagram. On closing pop-ups, users need time to re-orientate inside the scatterplot. To provide a smooth and intuitive shift from overview to detail information, the prototype once more employs the zooming feature. At a certain scale level, the interface switches from geometric to semantic zoom. This causes the rectangles to turn into record cards, i.e. into frames that display the content of the DVD item the rectangle represents (Figure 2c). Apart from the title, a record card also holds multiple sub-sections with

additional information such as movie details, posters and trailers. To view the data, users tap the corresponding sub-section and perform a zoom-in operation to magnify the section's content to a readable size (see Figure 2d). Overall, the semantic zoom allows users to rapidly switch between overview and detail information without leaving the scatterplot visualization. Hence, orientation disruptions are avoided.

## 1.2 Evaluation Abstract

An informal user test based on observation and interviews was conducted with six participants (2 male and 4 female) in a laboratory setting to receive some general feedback on the intuitiveness of the starfield visualization and interaction. For the user test the starfield application was installed on a Hewlett-Packard iPAQ hx4700 Pocket PC with Windows Mobile 2003 and Flash Player 6. The device featured a 624 MHz processor, 64 MB SDRAM and a 480x640/64K color VGA touchscreen. During the experiment, participants were first asked to try out the interface on their own while thinking aloud. Subsequently, they had to solve a set of 12 retrieval tasks.

During the test the participants quickly understood the basic principles of the application and thus were able to solve the test tasks without requiring much support. In particular, the concept of using semantic zoom to switch from abstract representation to a content view was much appreciated and proved to be as intuitive as expected. On the other hand, users found it increasingly difficult to orientate with a growing magnification of the scatterplot. Accordingly, we observed frequent zoom-out operations to the default scale. Another problem revealed by the user test concerns the zoom interaction. The sequential procedure of first setting a focus point and then pressing a button to manipulate the zoom level was found slow and tedious. Users comments indicate a clear preference for a one-step and purely pen-based zoom interaction.

## 2 OVERVIEW+DETAIL

While smooth zooming supported users in the informal user test and was rated as a positive user experience, it could not prevent users from getting lost in the information space. This result also matches previous research (Card et al., 1999), (Hornbæk et al., 2002). A more powerful solution to preserve user orientation may be provided by overview+detail (o+d) interfaces. O+d interfaces are characterized by a multi-window layout, where one window is used to present details while the other one gives an overview of the information space (Plaisant et al., 1995). In addition to supporting orientation, overviews can also offer additional interaction possibilities such as panning and scaling. A difficult design question when dealing with o+d interfaces on small screens relates to the layout of the two windows. For both views, the usability increases with a larger size, since the presentation becomes less cluttered. However, a larger detail view means a smaller overview, and vice versa. This trade-off

problem assumes that the interface uses the most common side-by-side layout of the two windows. Alternative layouts include transparent or on-demand overviews that overlap with the detail view. These techniques require less screen real estate, but typically make the interface more difficult to understand and use (Shneiderman & Plaisant, 2005) (Plaisant et al., 1995).

Mobile interfaces make increasing use of overview windows to support orientation. One example is the recent Nokia web browser for S60 3rd Edition devices. When the software detects extensive scrolling it displays a miniature image of the web page. Despite the growing acceptance of o+d interfaces on small screens, there has not been much research on that topic as yet. A few studies have been conducted in the context of desktop computers. The results give the impression that users typically prefer o+d interfaces, while user performance varies with different task types and application domains. (Hornbæk & Frøkjær, 2001), for instance, investigated the usability of a linear, a fisheye, and an o+d interface for electronic documents. They found that all but one user preferred the o+d interface, and that essays written with it received higher grades from experienced tutors. On the other hand, users completed reading and question-answering tasks fastest when using the fisheye view. Similar results were found in a study on web browsers (Baudisch, Lee, & Hanna, 2004): the participants preferred the o+d interface, but the overall performance was best when using a fisheye distortion. Another study specifically researched the usability of zoomable user interfaces with and without overviews for map navigation (Hornbæk et al., 2002). Again, the participants preferred the o+d interface but when the participants used a map with a semantic structure (multiple levels with different content granularity), the authors found that the additional window significantly decreased the user performance. It was hypothesized that the rich navigation cues of the semantic zoom render an overview rather unnecessary.

## 2.1 Interfaces

To research whether o+d interfaces can be exploited to improve navigation in starfield displays, we developed two interfaces - one that featured an overview (overview interface) and one that relied solely on the detail view (detail-only interface). Both interfaces were implemented using the Microsoft .NET compact framework 1.1.

The **detail-only interface** (Figure 3a) shows a simplified layout of the original Flash prototype. Database items are encoded as small rectangles and positioned inside a scatterplot diagram. The axes are set to the variables of popularity rating (X) and year of release (Y). Tapping the scatterplot causes the rectangle closest to the pen position to be highlighted. This strategy also ensures that even very small representations can be accurately selected. Highlighting an item moves it to the foreground and the user is provided with the movie's exact attribute values by displaying additional labels on the scatterplot axes.

In line with the user suggestions from the informal usability test, we modified the zoom feature to be fully controllable via pen events. While touching the screen highlights an item, holding the pen for more 150 milliseconds will trigger a fluent zoom-in operation on that item. Hence users can highlight (set a focus point) and magnify items in a single action. It is also important to notice that, unlike in the Flash prototype, users are now prevented from zooming into the empty space between objects. A zoom operation is always bound to an item.

During the zoom, the highlighted item is magnified and centered. The information space contains three distinct semantic levels that are displayed depending on the current scale. In the default scale, the rectangles are too small to display any content information (Figure 3a). On magnifying them, the second semantic level is triggered. The items are enhanced with movie posters that scale together with the outlining rectangle (Figures 3b, c). Once the item is scaled to fill the entire window space, the system switches to the third semantic level. In a quick animation the poster zooms out to a smaller size and is positioned at the upper right corner of a scroll pane. The free screen space is used to display the textual content of the movie data (Figure 3d).

Users can interrupt a zoom operation at any time by lifting the pen. Moving from the default scale to the content view of the scroll pane magnifies the information space forty times and takes 1.8 seconds. When zooming out by pressing the button at the bottom of the interface, the item grows smaller and changes its representation back to a poster and then to a simple rectangle. If not previously interrupted, the operation automatically stops at the default scale. Panning is again implemented as the action of dragging the information space with the stylus.

The **o+d interface** features a multi-window layout, where the upper view is similar to the one in the detail-only application, but its height is decreased by about 35 percent. The free space is used to display an additional overview window (Figure 4a). The sizes of the two windows have been chosen such that the axes units used on the overview can be the same as those on the detail view, with the labels remaining legible.

The overview window presents a miniature view of the entire information space at a static scale. Each dot on the overview is a rectangle on the detail view. To support the users' orientation, the overview includes a field-of-view box represented by a yellow rectangle. The rectangle shows the portion of the information space that is currently displayed on the detail view. During zooming, users are thus provided with off-screen context that is missing in the detail-only application. Detail view and overview are tightly coupled, which means that manipulating one view immediately updates the other one. If users highlight an item on the detail view, the corresponding dot on the overview turns red. Panning on the detail view shifts the field-of-view box.

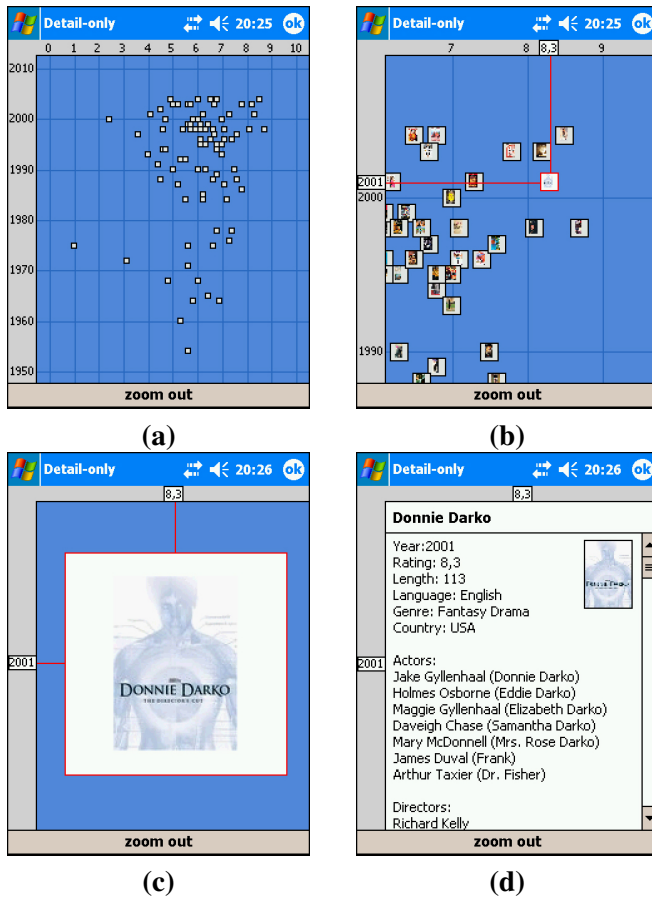


Figure 3: Detail-only interface: (a) default zoom-level, (b and c) zooming in, (d) record card view

Apart from visualizing off-screen content, the overview also provides additional navigation functionality. An alternative way to pan is to drag the field-of-view box on the overview. The detail view is continuously updated. Users may also jump to different locations by tapping outside the field-of-view box on the overview. The yellow rectangle automatically moves to center the pen position. To zoom on the overview, users first press the toggle button labeled *scale* and then draw a new field-of-view box by dragging the pen (bounding-box zoom). To remain consistent with the given ratio of the views, the box is limited to a fixed aspect ratio. Zooming on the overview is limited to the first two semantic levels of the information space. Hence to retrieve the textual content of an item, users still have to use the detail view.

## 2.2 Evaluation Abstract

To compare the usability of the two approaches, we conducted an experiment in which 24 participants had to complete 12 search tasks on an 85-item movie collection using both PDA starfield applications. The study was run on a Hewlett-Packard iPAQ

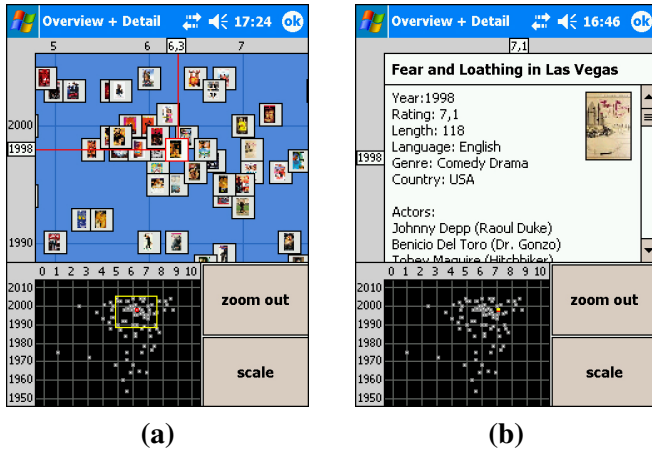


Figure 4: O+d interface to improve orientation.

hx4700 Pocket PC with Windows Mobile 2003. For a full report of the experiment see (Büring et al., 2006a).

Analyzing the results of the user test, we found that participants were significantly faster when using a detail-only ZUI compared to an overview-supported ZUI (379.34 seconds compared to 452.65 seconds,  $F(1, 23) = 16.5, p < 0.001$ ). This result is quite surprising since the logged interaction data shows that, as expected, the overview window had reduced the need for long-distance panning and zooming. We hypothesize that the loss in performance is due to the time required to frequently switch between the two views in the o+d interface. This corresponds to previous research (Baudisch, Good, Bellotti, & Schraedley, 2002). Moreover, the effect may have been amplified by the additional cognitive load introduced by the increased interface complexity. Another explanation would be that the rich navigation cues provided by scatterplot labels reduced the benefit of having an overview window (Hornbæk et al., 2002).

Unlike in (Hornbæk et al., 2002), our experiment did not show a significant difference in preference between the two ZUIs (10 participants preferred the overview interface, 13 the detail-only interface, one subject was unsure,  $X^2(1, N = 23) = 0.391, p < 0.532$ ). Participants who voted for the detail-only interface preferred the larger size of the detail window and the simplicity of the single-view layout. While they appreciated the orientation support, some users reported problems with zooming and panning on the overview. Due to the small size of the window, they found these functions rather imprecise and difficult to use. A larger overview would probably have improved the interaction but, as discussed earlier, could only be provided at the expense of a smaller detail view.

### 3 FISHEYE

While the o+d interface could not improve task-completion times or user satisfaction, the previous experiment nevertheless showed that overview information can reduce the need for unnecessary navigation.

A way to exploit this benefit without forcing users to switch between multiple views is to integrate both focus and context in a single view. A popular implementation of such an approach is the fisheye view (Furnas, 1986), a distortion concept that shows the focus in detail and the remote regions in progressively less detail. Other well-known focus+context solutions on desktop computers include the bifocal display ((Spence & Apperley, 1982)), the perspective wall ((Mackinlay, Robertson, & Card, 1991)) and hyperbolic views ((Lamping, Rao, & Pirolli, 1995)). Regarding mobile devices, an early example of a PDA application that utilizes a fisheye view is the WEST web browser (Björk et al., 1999). WEST uses a proxy server to split a web page into several chunks that are then presented as small cards in a left-to-right and top-to-bottom fashion on a PDA. There is always one card in focus that is displayed at a larger size than the other cards. The users can move the focus to enlarge different content chunks or they can zoom into a focused card to view it at full size. Later examples of mobile fisheye views are DateLens (B. B. Bederson, Clamage, Czerwinski, & Robertson, 2004) and AppLens (Karlson, Bederson, & SanGiovanni, 2005). These applications visualize calendar and application data in a tabular interface. The users can enlarge or minimize cells by tapping them with the stylus and thus distribute display size according to their focus.

#### 3.1 Interfaces

To examine the effect that a distortion strategy may have on the usability of a zoomable starfield display on PDAs, we implemented a fisheye view and a third design iteration of the detail-only prototype design. One requirement was that the applications should scale to larger information spaces of several thousands of items (in this case we used a book database). Moreover, dynamic query modules were added to the design to support data filtering. These tabbed controls at the bottom of the screen were used to provide the participants of the usability test with a more realistic search scenario and to emulate different levels of information density. In both applications the X-axis represents the year of publication and the Y-axis the sale price.

Compared to the previous design, several interaction and visualization details of the **detail-only interface** have been improved to support scalability. At the default scale, the scatterplot contains thousands of 1-pixel representations (Figure 5a). Moreover, since the screen holds fewer pixels than there are attribute units of the scatterplot dimensions, there may be thousands of items that are completely overlapped by other items. To make each item easily and accurately accessible by the users, the zoom animation - which is still triggered after a pen-down threshold of 150 milliseconds - has

been redesigned to follow a two-step algorithm: (1) only the scatterplot pane is magnified, i.e. the size of all the representations relative to each other remains unchanged, but the information space is stretched like a rubber sheet. At the same time, the focus object, which has been selected based on pen-proximity, moves to the center of the screen. Together, this causes a decomposition effect in which clusters of in-focus representations scatter, while other representations drift out of sight (Figure 5b; the circles added to the screenshot denote the position of the focus item). (2) once all clusters have been resolved, meaning that each gray pixel represents only items with the same attribute values, the zoom animation also takes effect on the representations. Gray pixels grow to small rectangles (Figure 5c) and then to record cards, which display information about the books they represent (Figures 5d-f). The transition between the two steps of the algorithm is fluent and automatic. No extra action, besides holding down the pen, is required from the users. One thing to note about the scatter effect is that it only aims to resolve clusters of items with different attribute values. To resolve overlappings of items with the same values for the scatterplot dimensions, a zooming approach for multiple data points as discussed in (Büring & Reiterer, 2005) would be a reasonable and consistent solution.

Zooming in from the default scale takes 1.8 seconds to reach the highest magnification level, and 1.3 seconds to zoom back out. The zoom is at its maximum when the record card in focus fills about 50 % of the diagram size, as in Figure 5f. Users can easily switch the focus to other record cards in the view by simply tapping them. The newly selected focus item is centered in a quick animation. Panning is implemented as *sliding* - a rate-based scrolling mechanism controlled by a pen-gesture. Users tap the display and drag the pen in the direction they want to move the viewport (this is the opposite of dragging the information space). The further they drag the pen, the more the animated view movement accelerates, and vice versa. The benefit of sliding is that long distances can be covered very quickly. A similar panning approach has also been discussed and user-tested in (MacKay, Dearman, Inkpen, & Watters, 2005). To distinguish between zoom and pan attempts, we use a distance threshold of 5 pixels. A zoom operation is initiated if users hold the pen down for more than 150 milliseconds without moving it more than 5 pixels, otherwise the input is interpreted as a panning gesture.

Another re-design aspect targets the record card presentation. To truly base the ratio of overview and detail information on the degree of zooming the users perform, the layout of the record cards changes as fluently and smoothly as the scale. First the application tries to display as much of the book title as possible, assuming that this is the most important portion of the content available (Figure 5d). In the case where vertical space is still left, it is filled with the scaled book cover. Any remaining horizontal space next to the cover is used to show additional textual details such as the author's name, shipping information, etc (Figure 5e). Much effort has also been spent on improving the scatterplot labeling. In the former versions, the labels remained static and, just like items that are not in focus, they moved out of the viewport during zooming. In

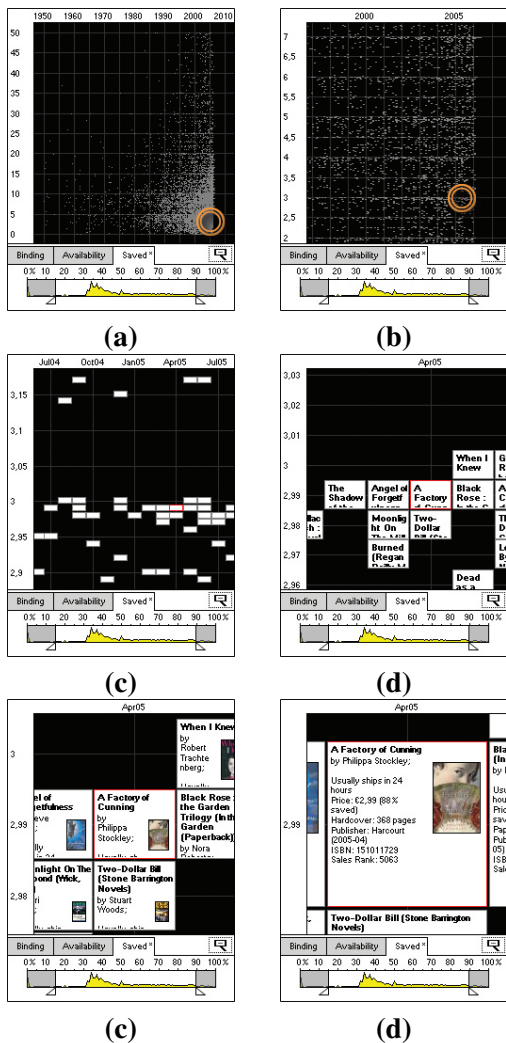


Figure 5: Exploring 7500 books on a PDA.

the re-design, labels are rapidly and continuously updated to mirror the current level of view granularity. For instance, in the default scale the X-axis is subdivided into decades and the Y-axis into units of EUR 5 (Figure 5a). While zooming in, the labels and grid lines drift apart. The gap is used to display new grid lines that, as soon as there is sufficient display space available, are equipped with labels of the next-smaller unit measure (e.g. Figure 5c). When the scale has reached its maximum, each scatterplot dimension is displaying its smallest attribute unit. The pre-defined granularities are 10 years, 5 years, year, and quarter for the X-axis and 5 euro, 1 euro, 10 cent, 5 cent, and 1 cent for the Y-axis.

The **fisheye interface** makes use of a discrete interface distortion. To prune visual clutter, users first tap the box-icon button below the diagram and then draw a bounding-box denoting the focus region (Figure 6a). Upon lifting the pen, the system responds by smoothly centering and magnifying the boxed region in 600 milliseconds

to about 75% of the diagram size (Figure 6b). This results in a layout in which more display space is available for the focus region (items drift apart, clutters can be resolved), while objects that are not in focus are still visible but are allocated less space. The preservation of context has the advantage that users can make informed decisions about where to navigate; they may, for instance, prevent themselves from panning into regions where there are no items. Users can also directly zoom into a context region by redrawing the bounding box to include a portion of the regions surrounding the current focus. To achieve this with an undistorted ZUI, users would need to previously zoom out before adjusting the focus.

Depending on the given information density, users may have to apply the fisheye zoom recursively to isolate items. An important property of the distortion applied is that it does not affect parallelism between lines. In contrast, a radial distortion may for instance have significantly increased the cognitive load to map items to the diagram labels. As with the detail-only application, the scatterplot labels are continuously updated during the magnification animation. However, to avoid confusing overlappings, the context regions are only labeled with the start or end unit of the axis. To return to a previous distortion setting, or to undistort the interface, users tap, or tap and hold the button with the magnifying glass and the minus sign. Due to the compressed information space in this interface, precise panning is more important than fast panning. Hence panning is implemented as drag&drop; users move items in and out of the focus by dragging the diagram pane with the stylus.

While the interface distortion allows users to resolve visual clutter, it does not reveal any detail information about book items. To shift from overview to content data, users tap the item they are interested in. Again, they are not required to accurately tap the 1-pixel representation; it is sufficient to tap in the item's vicinity. The selected object zooms to full screen in 250 milliseconds. During the animation, the gray pixel grows to the familiar record card representation showing all detail information about the book along with its cover (Figures 6c, d). On tapping anywhere on the record card, the item zooms back to its original size and position in the scatterplot. This approach is different to a conventional pop-up mechanism in that it leads the user's eye. The animation gives a visual cue about where the item comes from, and where it goes back to. Hence users can quickly access the content of different items, without the need to re-orientate after each zoom-out operation. The detail zoom can be applied to all visible items, regardless of whether they are located in the focus or in a context region, or whether the interface is currently distorted or not.

### **3.2 Evaluation Abstract**

We conducted an experiment in which 24 participants were asked to solve retrieval tasks using either of the two interfaces. Search tasks were based on an information space of 7,500 book items. While we tested our highly animated applications suc-

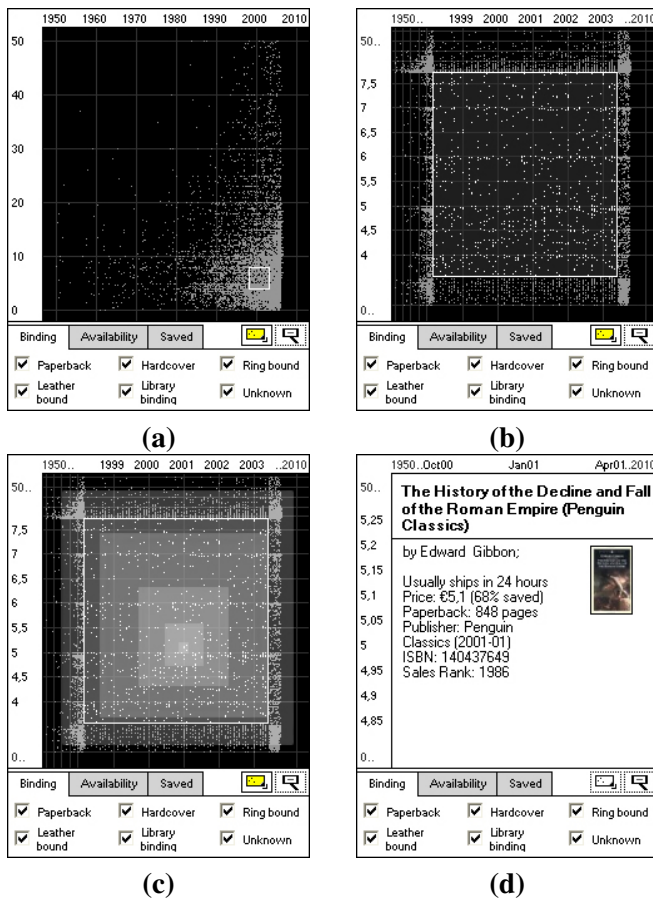


Figure 6: Fisheye interface: (a) selecting a focus region, (b) distortion result, (c) item zooms to full-size, (d) detail view

cessfully with a few hundred items on HP and Dell PDAs, current mobile devices are not yet powerful enough to cope with thousands of data records. Thus we simulated a standard PDA interface (240 x 320 pixel) on a stylus-operated Wacom Board connected to a 3 GHz Pentium 4 PC with 1 GB RAM. This apparatus allowed us to use pen-interaction while also being provided with sufficient processing power. To run on the test machine, both applications were ported from the compact framework to the full .Net framework. The experiment is fully reported in (Büring et al., 2006b).

The results of the usability tests were rather surprising. Due to the integrated overview, we had expected that the fisheye view would result in quicker navigation and less unnecessary exploration (see also (Schaffer et al., 1996)). However, it took users about the same time to complete 10 tasks with both interfaces (623.8 seconds for the detail-only interface compared to 612.4 seconds for the fisheye interface,  $F(1,22) = 0,002$ ,  $p=n.s$ ). Considering that the fisheye interface required far fewer actions, we must conclude that these actions took more time to execute. Hence it seems that drawing a bounding box and then accessing items one-by-one is cognitively more de-

manding than the more comprehensive zooming approach in the detail-only interface.

Regarding preference, 20 subjects preferred the fisheye interface and only three the detail-only interface ( $X^2(1, N = 23) = 12.565, p < 0.001$ ). This difference is highly significant. User statements suggest that the result is due to the better orientation features and the more precise navigation offered by the fisheye interface. This finding partly contradicts related research. For instance, two studies that compared overview+detail interfaces to a fisheye view found that although the fisheye improved user performance for many task types, subjects were still clearly in favor of the alternative interface (Hornbæk & Frøkjær, 2001) (Baudisch et al., 2004). While the rather artificial distortion of the fisheye view may have discouraged users in those experiments, we hypothesize that in our case the fisheye benefited from the abstract representation of the diagram. The distortion integrates very well into the scatterplot layout. This may be quite different for domains such as maps, in which a higher degree of fidelity to the standard layout is essential (Plaisant et al., 1995).

The preference for the fisheye view is also influenced by the users' preference for the more conventional drag&drop mechanism over the sliding technique (16:7,  $X^2(1, N = 23) = 3.522, p = 0.061$ , n.s.). We think the reasons for this result are twofold. First, all participants were used to drag&drop as it is supported by all major window systems. Sliding, on the other hand, was a highly unfamiliar technique. The other reason relates to the clipping approach in the detail-only prototype. Even though sliding enables users to pan at high speed in the information space, this feature does not compensate for the lack of overview information. If users do not know in which direction to move, speed is hardly an advantage. In addition, a drawback of the technique is that when users are sliding very quickly and pass an item, it appears only as a quick flicker of the screen. Thus, it takes users some time to react and then they have to back-track to find the item. Comments suggest that our participants found this procedure rather difficult.

## 4 OUTLOOK

As already mentioned, the usability of the fisheye may depend quite strongly on the type of information space displayed. When applied to map-based starfields, users are likely to become irritated by distorted landmarks such as streets or buildings. The detail-only interface featuring the geometric semantic zoom preserves the standard layout and thus may be better suited for geographic application domains. However, the interface suffers from a lack of overview information and is also in need of navigation features that allow the user to move long distances quickly and precisely. To overcome these problems, we suggest that the interface should be enhanced with two techniques proposed by previous research.

Map-based starfields usually feature points-of-interest (POI), i.e. locations that are emphasized by drawing a symbol on top of the map. POIs may, for instance, be gas stations or hotels in a navigation system. Given the detail-only interface, the overview could be significantly improved by visualizing off-screen objects. This can be accomplished by using a technique called Halos (Baudisch & Rosenholtz, 2003). Halos are rings that are drawn around clipped POIs and are just large enough to reach into the current view port (see Figure 7). Based on the halo curvature, users are able to estimate the distance and direction of off-screen items. When moving the view port, the rings are rapidly updated. The authors evaluated Halos in comparison to conventional arrows. They found that for navigation tasks, Halos led to a significant timesaving of up to 33%. The downside is that Halos do not scale well. Even a small number of POIs can cause Halos to clutter and thus lessen the technique's usability. To improve clarity, relevance thresholds have been suggested to limit the number of rings to be displayed (Baudisch & Rosenholtz, 2003). However, this technique assumes a prior definition of relevance by the user. Hence additional interaction widgets are required that must be designed such that the simplicity of the original visualization is not hampered.



Figure 7: Halos are rings that visualize off-screen points-of-interest.

Once Halos are displayed, the system must provide a way to quickly navigate to the corresponding off-screen locations. Due to visual blur, rate-based scrolling techniques such as sliding have been found hard to control. Separate zoom and pan operations are also not ideal, since they are slow and tedious to perform. A solution for this problem may be speed-dependent automatic zooming (SDAZ) (Igarashi & Hinckley, 2000). SDAZ couples scroll speed with scaling and thus combines zooming and panning into a single operation. When scrolling quickly the display zooms out, and when reducing scroll speed the display zooms back in. In this way, visual flow remains quite constant, which makes it easier for the users to navigate at high speed at the same time as they are being provided with an overview of the information space. While a preliminary informal evaluation for document and map navigation showed mixed results, more

recent user tests revealed a clear advantage of SDAZ when compared to conventional scroll, pan and zoom methods (Cockburn, Savage, & Wallace, 2005b) (Cockburn, Savage, & Wallace, 2005a) (Cockburn & Savage, 2003). Another study applied SDAZ to a small screen that mimicked the standard PDA resolution (Jones, Jones, Marsden, Patel, & Cockburn, 2005). The novel zoom interaction had the effect of worsening task completion times and from that, the authors concluded that the small screen space had reduced the impact of SDAZ. However, since the study lacked a control group, more research must be conducted to clarify this point. In addition, participants controlled the interface via a computer mouse, which is not a standard input mechanism for devices that typically feature a small screen. Thus, it would be interesting to evaluate whether pen and mouse interactions have different effects on the usability of SDAZ.

Another topic for further research may be to examine the scalability of zoomable starfield displays in terms of screen size. The interfaces developed not only meet the requirements of mobile devices, but could also provide a valuable solution for presenting very large sets of data on desktop computers and wall-sized displays.

## **5 CONCLUSION**

Several interaction strategies to improve the usability of starfield displays on PDAs have been implemented and evaluated. While smooth geometric and semantic zooming provided an intuitive metaphor for exploring information spaces, users still showed difficulties in preserving their orientation while navigating. An interface featuring a separate overview window was not able to improve user satisfaction and moreover, due to the small size of the control and the cognitive costs of visual switching, it worsened task completion times. The problem of visual switching could be avoided by introducing a fisheye distortion algorithm. Context regions are not clipped but are preserved by being contracted. More space can then be allocated to the items in focus. While task completion times remained similar to the default smooth-zooming starfield, users significantly preferred the fisheye view. This is an important result that may encourage designers to employ distortion strategies when displaying abstract information spaces on small screens. However, for less abstract data such as maps the detail-only interface featuring the geometric semantic zoom may still provide the better solution, especially when enhanced by techniques such as speed-dependent automatic zooming and Halos.

## **6 ACKNOWLEDGMENT**

This work was supported by the DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces".

## References

- Ahlberg, C. (1996). Spotfire: an information exploration environment. *SIGMOD Rec.*, 25(4), 25–29.
- Ahlberg, C., & Shneiderman, B. (1994). Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Chi '94: Proceedings of the sigchi conference on human factors in computing systems* (pp. 313–317). ACM Press.
- Baudisch, P., Good, N., Bellotti, V., & Schraedley, P. (2002). Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Chi '02: Proceedings of the sigchi conference on human factors in computing systems* (pp. 259–266). New York, NY, USA: ACM Press.
- Baudisch, P., Lee, B., & Hanna, L. (2004). Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. In *Avi '04: Proceedings of the working conference on advanced visual interfaces* (pp. 133–140). New York, NY, USA: ACM Press.
- Baudisch, P., & Rosenholtz, R. (2003). Halo: a technique for visualizing off-screen objects. In *Chi '03: Proceedings of the sigchi conference on human factors in computing systems* (pp. 481–488). New York, NY, USA: ACM Press.
- Bederson, B., & Meyer, J. (1998). Implementing a zooming user interface: experience building pad++. *Softw. Pract. Exper.*, 28(10), 1101–1135.
- Bederson, B. B., Clamage, A., Czerwinski, M. P., & Robertson, G. G. (2004). Date-lens: A fisheye calendar interface for pdas. *ACM Transactions on Computer-Human Interaction*, 11(1), 90–119.
- Björk, S., Holmquist, L. E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., et al. (1999). West: a web browser for small terminals. In *Uist '99: Proceedings of the 12th annual acm symposium on user interface software and technology* (pp. 187–196). New York, NY, USA: ACM Press.
- Burigat, S., & Chittaro, L. (2005). Visualizing the results of interactive queries for geographic data on mobile devices. In *Gis '05: Proceedings of the acm international workshop on geographic information systems* (pp. 277–284). New York, NY, USA: ACM Press.
- Büring, T., Gerken, J., & Reiterer, H. (2006a). Usability of overview-supported zooming on small screens with regard to individual differences in spatial ability. In *Avi '06: Proceedings of the working conference on advanced visual interfaces* (pp. 233–240). New York, NY, USA: ACM Press.
- Büring, T., Gerken, J., & Reiterer, H. (2006b, September-October). User interaction with scatterplots on small screens - a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2006)*, 12(5), 829–836.
- Büring, T., & Reiterer, H. (2005). Zuiscat: querying and visualizing information spaces on personal digital assistants. In *Mobilehci '05: Proceedings of the 7th*

- international conference on human computer interaction with mobile devices & services* (pp. 129–136). New York, NY, USA: ACM Press.
- Card, S. K., Mackinlay, J. D., & Shneiderman, B. (Eds.). (1999). *Readings in information visualization: Using vision to think*. San Francisco: Morgan Kaufmann Publishers.
- Cockburn, A., & Savage, J. (2003). Comparing speed-dependent automatic zooming with traditional scroll, pan, and zoom methods. In *People and computers xvii: British computer society conference on human computer interaction* (p. 87-102).
- Cockburn, A., Savage, J., & Wallace, A. (2005a). Comparing automatic and manual zooming methods for acquiring off-screen targets. In *People and computers xix: British computer society conference on human computer interaction* (pp. 439–454).
- Cockburn, A., Savage, J., & Wallace, A. (2005b). Tuning and testing scrolling interfaces that automatically zoom. In *Chi '05: Proceedings of the sigchi conference on human factors in computing systems* (pp. 71–80). New York, NY, USA: ACM Press.
- Dix, A., & Ellis, G. (1998). Starting simple: adding value to static visualisation through simple interaction. In *Avi '98: Proceedings of the working conference on advanced visual interfaces* (pp. 124–134). New York, NY, USA: ACM Press.
- Dunlop, M. D., & Davidson, N. (2000). Visual information seeking on palmtop devices. In *Proceedings of hci2000, volume 2* (p. 19-20).
- Dunlop, M. D., Morrison, A., McCallum, S., Ptaskinski, P., Risbey, C., & Stewart, F. (2004). Focussed palmtop information access combining starfield displays and profile-based recommendations. In F. Crestani, M. Jones, & S. Mizzaro (Eds.), *Proceedings of workshop on mobile and ubiquitous information access, lncs v2954* (p. 79-89). Springer.
- Furnas, G. W. (1986). Generalized fisheye views. In *Chi '86: Proceedings of the sigchi conference on human factors in computing systems* (pp. 16–23). New York, NY, USA: ACM Press.
- Hornbæk, K., Bederson, B. B., & 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.
- Hornbæk, K., & Frøkjær, E. (2001). Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. In *Chi '01: Proceedings of the sigchi conference on human factors in computing systems* (pp. 293–300). New York, NY, USA: ACM Press.
- Igarashi, T., & Hinckley, K. (2000). Speed-dependent automatic zooming for browsing large documents. In *Uist '00: Proceedings of the 13th annual acm symposium on user interface software and technology* (pp. 139–148). New York, NY, USA: ACM Press.
- Jones, S., Jones, M., Marsden, G., Patel, D., & Cockburn, A. (2005). An evaluation of integrated zooming and scrolling on small screens. *Int. J. Hum.-Comput. Stud.*,

63(3), 271–303.

- Karlson, A. K., Bederson, B. B., & SanGiovanni, J. (2005). AppLens and LaunchTile: two designs for one-handed thumb use on small devices. In *Chi '05: Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 201–210). New York, NY, USA: ACM Press.
- Lamping, J., Rao, R., & Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Chi '95: Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 401–408). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.
- MacKay, B., Dearman, D., Inkpen, K., & Watters, C. (2005). Walk 'n scroll: a comparison of software-based navigation techniques for different levels of mobility. In *MobileHCI '05: Proceedings of the 7th international conference on human computer interaction with mobile devices & services* (pp. 183–190). New York, NY, USA: ACM Press.
- Mackinlay, J. D., Robertson, G. G., & Card, S. K. (1991). The perspective wall: detail and context smoothly integrated. In *Chi '91: Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 173–176). New York, NY, USA: ACM Press.
- Perlin, K., & Fox, D. (1993). Pad: an alternative approach to the computer interface. In *Siggraph '93: Proceedings of the 20th annual conference on computer graphics and interactive techniques* (pp. 57–64). ACM Press.
- Plaisant, C., Carr, D., & Shneiderman, B. (1995). Image-browser taxonomy and guidelines for designers. *IEEE Softw.*, 12(2), 21–32.
- Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., et al. (1996). Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Trans. Comput.-Hum. Interact.*, 3(2), 162–188.
- Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *VI '96: Proceedings of the 1996 IEEE Symposium on Visual Languages* (pp. 336–343). IEEE Computer Society.
- Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface* (4. ed ed.). Pearson/Addison-Wesley.
- Spence, R. (2001). *Information visualization*. Addison-Wesley.
- Spence, R., & Apperley, M. (1982). Data base navigation: an office environment for the professional. *Behaviour and Information Technology*, 1(1), 43–54.
- Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, Connecticut, U.S.A.: Graphics Press.
- Williamson, C., & Shneiderman, B. (1992). The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *Sigir '92: Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 338–346). ACM Press.