

User Interaction with Scatterplots on Small Screens - A Comparative Evaluation of Geometric-Semantic Zoom and Fisheye Distortion

Thorsten Büring*

Jens Gerken†

Harald Reiterer‡

ABSTRACT

Existing information-visualization techniques that target small screens are usually limited to exploring a few hundred items. In this article we present a scatterplot tool for Personal Digital Assistants that allows the handling of many thousands of items. The application's scalability is achieved by incorporating two alternative interaction techniques: a geometric-semantic zoom that provides smooth transition between overview and detail, and a fisheye distortion that displays the focus and context regions of the scatterplot in a single view.

A user study with 24 participants was conducted to compare the usability and efficiency of both techniques when searching a book database containing 7500 items. The study was run on a pen-driven Wacom board simulating a PDA interface. While the results showed no significant difference in task-completion times, a clear majority of 20 users preferred the fisheye view to the zoom interaction. In addition, other dependent variables such as user satisfaction and subjective rating of orientation and navigation support revealed a preference for the fisheye distortion. These findings partly contradict related research and indicate that, when using a small screen, users place higher value on the ability to preserve navigational context than they do on the ease of use of a simplistic, metaphor-based interaction style.

CR Categories: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.—Design, Human Factors, Interaction styles, Screen design, Evaluation;

Keywords: Small screen, PDA, scatterplot, zoom, fisheye, focus+context.

1 INTRODUCTION

Small devices such as smart phones and Personal Digital Assistants (PDAs) are becoming increasingly powerful and thus will soon serve as a truly mobile alternative to notebooks. Field sales staff, for instance, may use a PDA not only to manage business contacts and dates but also to access and browse the company database to retrieve product information while they are on the road. However, visualizing large data sets on a limited screen real estate requires the designer to rely on space-preserving encodings rather than on textual representation of data. A way to accomplish this task is by using starfield displays. The starfield approach consists of a scatterplot that is enhanced with additional features to support selection and zooming. The scatterplot in Figure 1 shows a visualization of a book database containing 7500 items on a PDA. Each book is represented by a gray pixel and mapped to a Cartesian position by the scatterplot dimensions of year of publication (X-axis) and sale price (Y-axis). The year of publication denotes a date consisting of year

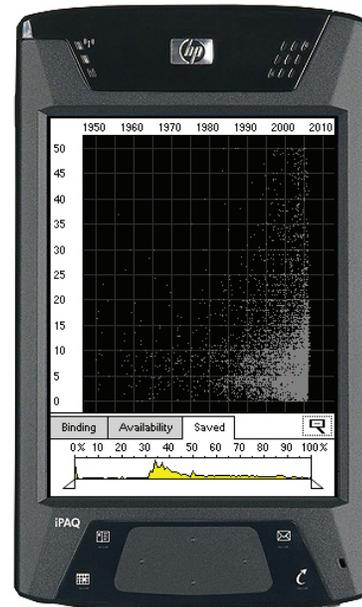


Figure 1: scatterplot visualization of a book database containing 7500 items. The X-axis represents the year of publication and the Y-axis represents the sale price.

and month ranging from January 1950 to December 2010. The sale price holds a value between 0 and EUR 50.

Starfield displays have been found useful in a variety of retrieval scenarios and serve as a standard component of commercial visual analysis tools such as Spotfire [1]. The problem with scatterplots however is that, given standard database sizes, the display usually becomes crowded with data items and thus overlappings cannot be avoided. When developing for small screens, this problem is even more acute: In Figure 1, the screen and attribute characteristics amount to a potential information density of 150 items per pixel. In other words, one gray pixel inside the scatterplot may represent up to 150 books.

There are two interesting research questions related to information density. One deals with the effective visualization of overlappings and has been discussed in [15] and [16], for instance. The topic this paper focuses on has not been addressed as yet and aims to provide interaction to prune visual clutter and to make each item easily accessible to the user. Existing solutions have the drawback that their usability deteriorates with increasing information density or that the interaction style cannot be leveraged for pen-driven devices. We therefore implemented and user-tested two novel interaction techniques to explore large scale scatterplot data on PDAs, one based on a zoomable user interface (ZUI) and the other one utilizing a fisheye distortion technique.

*e-mail: buering@inf.uni-konstanz.de

†e-mail: gerken@inf.uni-konstanz.de

‡e-mail: reiterer@inf.uni-konstanz.de

2 RELATED WORK

Scatterplots have been used to present abstract data since the early 1800s. Due to their aggregate representation they give an overview of the data that clearly reveals outliers, clusters and trends [32]. While paper-based diagrams are limited to a single view, interactive scatterplots allow the user to explore the information space by moving between multiple representations and viewpoints [12]. Starting out from the overview, and in accordance with Shneiderman's information-seeking mantra [31], the application must support the following kinds of interaction:

Details on demand

Since the overview of the information space does not reveal any details, the application must provide an interaction feature to display information about specific data items on demand.

Zoom

The details-on-demand approach assumes that all scatterplot items are selectable. Due to data congestion, this is often not the case. Moreover, overlappings hamper a precise examination of data subsets. Considering the scatterplot in Figure 1, users may for instance be interested in the distribution of book publication dates within the price range of EUR 3.50 to EUR 5 between 1990 and 1991 grouped by months. To reveal such information and to make each item accessible, users first need to zoom in on a region of interest. Zooming means changing the overall granularity of the view by decomposition [18], i.e. breaking a larger data group into smaller groups based on the given scatterplot dimensions.

Filter

Filtering for scatterplot visualization is usually realized via a time-multiplexing technique such as dynamic queries [4]. While users adjust control panels to define filters, the interface is rapidly updated, highlighting those items that lie within all the filter ranges. However, filtering does not guarantee that overlappings of data items are totally removed from the diagram. Thus, dynamic queries are a powerful option to support information seeking but do not allow scatterplot exploration by themselves. In this paper we therefore concentrate on how to provide interaction for view decomposition and details on demand.

Among other domains, interactive scatterplot visualizations have been applied to file systems [15], movie and real estate databases [3, 33] and pharmaceutical, demographic and economic data [1, 11]. Almost all systems provide a details-on-demand interaction by selecting items from the scatterplot interface. Details are then presented in a separate window, which allows users to view two data representations of different granularity at the same time. Though useful in desktop scenarios, this approach is hardly suitable for small screens. A separate window strongly affects the usability of a scatterplot interface as it takes up valuable screen space. A smaller scatterplot window in turn increases data clutter, which makes it more difficult for users to select specific items. Hence, on small screens, overview and detail information should be presented in the same view. The transition between the views must be designed such that the application supports rapid exploration of items without causing orientation disruptions. The latter may for instance occur in systems like [3] in which details are shown in a pop-up window. To close the pop-up, users have to click a small icon in the window corner, which frequently drags the attention and mouse cursor away from the original scatterplot focus.

While some systems do not manage overlappings at all, [3, 1] provide zoom bars to reduce visual clutter. Equipped with three thumbs, a zoom bar is a slider that controls the range boundaries of an axis dimension. Moving the two extreme thumbs, users can increase or decrease the upper and lower range boundary, causing a zoom in or a zoom out by changing the scale on the corresponding display axis [22]. Zoom bars have the advantage that rectangular

regions of any aspect ratio can be enlarged to full diagram size. Due to the position of the thumbs, users receive instantaneous feedback on the degree of zooming, and its similarity to a scrollbar may further support first-time users in operating the widget. On the other hand, the usability of a zoom bar deteriorates with a decreasing ratio of the physical slider size and the attribute range of the related scatterplot dimension. When visualizing large-scale data on small screens, the increased thumb sensitivity may render a precise zoom operation difficult if not impossible. Moreover, zooming in on a region of interest means having to subsequently modify up to four thumbs, which makes navigation rather tedious. Since sliders also take screen space away from the scatterplot, zoom bars can be considered inappropriate for small screens.

As an alternative to zoom bars, [1] offers point-directed zooming, which is triggered by mouse events. To avoid confusion with the item-selection action, this approach assumes a three-button mouse. Users have to press the second mouse button to zoom in and a third mouse button to zoom out. Unlike zoom bars, point-directed zooming allows zooming in both scatterplot dimensions simultaneously. However, while zooming provides a valid option for view decomposition on small screens, mouse commands do not. Devices such as PDAs usually feature a touch screen in conjunction with a pen or stylus. This kind of interaction has been found easy for novice users to learn, and was recently also incorporated into gaming devices such as the Nintendo DS (<http://www.nintendo.com/systemsds>). On the other hand, it limits primary input commands to no more than screen taps. This constraint must be taken into account when designing for small screens.

Despite the potential benefits, there have been hardly any attempts to implement interactive scatterplots on mobile devices. One of the very few is [13], which visualizes a movie database containing 71 items on a palmtop. The dimensions were chosen such that no overlappings of items with different attribute values would occur. We enhanced this approach with additional features and smooth zooming in [10]. Other applications such as [8, 14] focused on map-based visualizations of points of interest, but again the data sets used were very limited and thus information density was not an issue. With regard to this, we implemented two applications that would allow the exploration of many thousands of items on a scatterplot screen of 240 x 320 pixel.

3 SCATTERPLOT INTERACTION

The applications that have been developed feature different kinds of zoomable user interfaces (ZUIs) to allow for effective view decomposition and details on demand. While pen-interaction is assumed, the techniques presented may also be used in a split-screen scenario on desktop computers where, due to multiple views, there is little space available for the scatterplot. ZUIs are based on the assumption that navigation in information spaces is best supported by tapping into our natural spatial and geographical ways of thinking [27]. To implement this approach on a computer screen, data objects must be organized in space and scale. Users can navigate this space by performing zooming (changing the scale) and panning (movement at constant scale) operations. Since ZUIs make more efficient use of limited screen real estate, they thus provide a valuable solution for enhancing small displays.

A drawback with ZUIs is that users may easily become disoriented once they have zoomed into the information space. This is due to the clipping of orientation cues, an aspect that becomes increasingly significant as less display real estate is available. To overcome this problem, three solutions have been proposed: smooth zooming [6], overview+detail [28] and focus+context interfaces [17]. In a former experiment we have already investigated the usability of smooth zooming for a PDA scatterplot interface with and without overview

[9]. Although users acknowledged the orientation benefit of an additional overview window, they were significantly faster when using a detail-only ZUI. Next to the critical space trade-off, we assume that the decrease in user performance was also partly due to the visual switching between the overview and the detail window, which is cognitively very demanding [5]. In this paper we therefore want to extend the PDA scatterplot interface with the detail+context technique of a fisheye view. The latter provides overview information without the need for an additional window. It is contrasted with an advanced version of the detail-only ZUI.

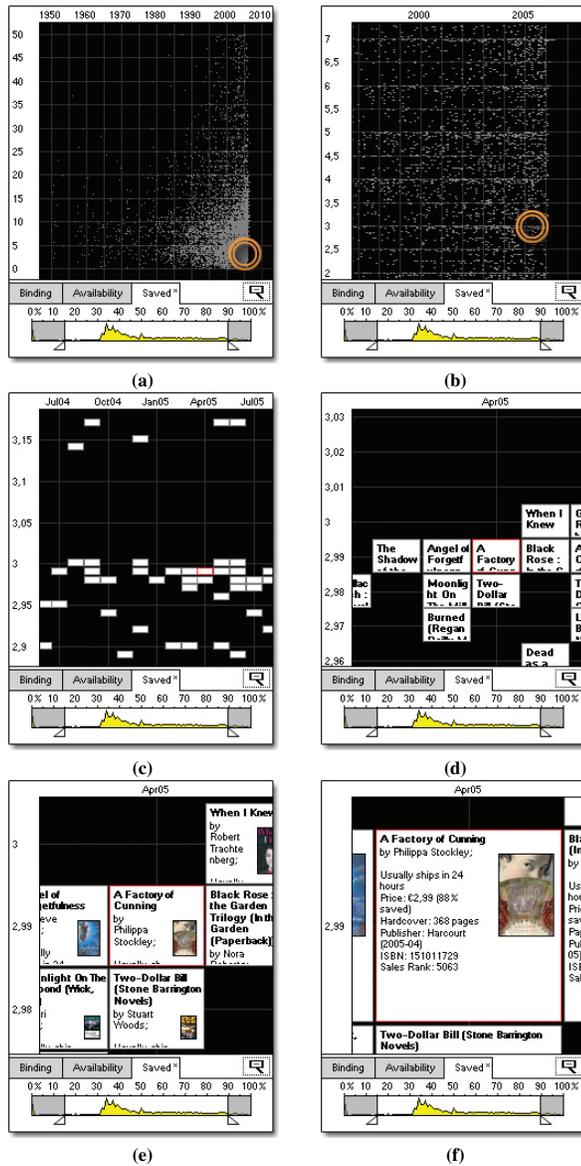


Figure 2: geometric-semantic ZUI: smooth transition from overview to detail by tapping and holding the pen on the display.

3.1 Geometric-semantic ZUI

We can distinguish two basic zoom concepts. Geometric zoom, which is often found in PDF readers such as Adobe Acrobat Reader and Apple's Preview, enlarges objects while zooming in and shrinks them when zooming back out. A more complex approach is semantic zooming. Here, information objects are given a different visual

representation, depending on the amount of real estate available to them [27]. This technique has for instance been used in a PDA web browser that displays websites as readable thumbnail views [26]. On zooming in, users are provided with the unabbreviated version of the corresponding page elements such as paragraphs or input boxes. A similar application facilitates image browsing on PDAs [25]. Images are presented as thumbnails and grouped by directories. Users can browse through the image hierarchies, while semantic zooming is used to display different thumbnail resolution at each level of detail. For the scatterplot visualization we implemented both geometric and semantic zooming to provide a fluent transition from scatterplot overview to detail information in a single view.

3.1.1 Interaction Design

The basic idea of the geometric-semantic ZUI is that users can control the ratio of overview and detail information by the degree of scaling they perform. The more they zoom in, the fewer items are shown, but more details are presented.

Users start the scatterplot exploration by tapping a region of interest and holding the pen down on the display (Figure 2a). The system responds by highlighting the item that is closest to the pen position. It becomes the "focus item". After an initial delay of 150 milliseconds a fluent zoom animation is triggered, which consists of two steps. First, only the scatterplot pane is magnified (Figure 2b, the circles have been added to the screenshot to denote the position of the focus item). Book representations that are far away from the focus item drift out of sight while the focus item itself moves smoothly into the center of the view. This reflects the assumption that books with similar attribute values are of more interest to the user than those whose year of publication and sale price differ widely from that of the focus item. While zooming in, the space gained through clipping is allocated to the remaining items and causes data clusters to scatter. As soon as the decomposition has reached a level at which all overlappings have been resolved, the second zoom step takes effect. Based on semantic zooming, the continuously increasing distance between items is now used to change their appearance. Gray pixels grow to small rectangles (Figure 2c) and then to record cards, which are used to display information about the books they represent (Figure 2d-f).

Zooming in from the default scale to the maximum zoom level takes 1.8 seconds. Users can, at any time, interrupt the operation by lifting the pen. This would for instance be necessary to examine the data distribution of a zoomed region or to choose another focus item. If not interrupted, the animation continues until the record cards have reached a size of 50% of the view size. In this state, the focus item is always centered in the view window. Users can access book details while surrounding items remain partly visible (Figure 2f). To zoom out, users press the button showing the magnifying glass with the minus sign in it. It is located below the diagram. Scaling down, the diagram pane shrinks and the items change their appearance back to a single pixel representation before being aggregated to the initial distribution and density. Again, users can stop the operation by lifting the pen, otherwise the animation continues until the default scale of the diagram has been reached. Zooming out takes 1.3 seconds at most and depends on the scale level.

An important feature is the highlighting mechanism. Due to the automatic selection based on proximity, zoom actions always target an item. This helps to avoid a scenario in which users zoom into the empty space between objects until all navigational cues have been clipped. This problem is known as *desert fog* [23] and can cause severe navigational problems for both novice and expert users.

As well as zooming, users can also pan the diagram via a pen gesture. Users tap the display and drag the pen in the direction they want to move the viewport. The speed of the animation that is triggered after the pen has moved at least 5 pixels away from its original

position is defined by the overall distance it moves. Thus, during panning, users can control the direction and speed through the pen position in real time. If the pen is dragged back to its original position, the movement slows down. Using this kind of sliding, it is possible to travel short and long distances very accurately. Users can bring the panning to a halt by lifting the pen. Another option for moving from one item in the view to another is to simply tap it. The tapped item now becomes the focus item and moves automatically to the center of the view.

3.1.2 Implementation

For the information space shown in Figure 1, the two scatterplot dimensions are characterized by a very different level of information density. The number of potential attribute values that each physical unit (pixel) on the Y-axis has to represent is several times larger than the corresponding number for each unit on the X-axis. To achieve an apparently uniform decomposition of both axes, a zoom operation magnifies the scatterplot dimensions with different scale factors. This results in the chessboard-like pattern of adjacent record cards (Figure 2c-f) and minimizes the empty space between objects. Hence users need to perform fewer navigational actions to move the viewport from one item to another.

Presenting book information on the magnified record cards (Figure 2f) has the advantage that no additional window for data output is needed. For each book there exist three information elements: book title, cover and textual details (i.e. author, shipping etc.). Screen space is allocated by a predefined priority rating. First, the application displays as many characters of the title as will fit into the enclosing rectangle (Figure 2d). If there is vertical space left, the image cover is scaled and placed on the right hand side beneath the title. If there is horizontal space left next to the cover, it is filled with textual details (Figure 2e). Due to this algorithm, users need to perform less zooming for high priority information such as book titles and more for low priority elements such as the ISBN code, for instance. Obviously, a possible improvement would be to replace the predefined layout by an intelligent solution that adjusts the priority rating of elements according to the user's specific information need. That way it would be easier to compare book attributes across items.

Orientation inside the diagram relies strongly on the scatterplot labels. Thus, during zoom operations, labels are rapidly and continuously updated. In the default scale the X-axis is subdivided into decades and the Y-axis into units of EUR 5. While zooming in, labels and grid lines drift apart. New grid lines are drawn between them and, as soon as enough space is available, they are labeled with the next-smaller unit measure. This algorithm continues until each axis is displaying its smallest unit. For the X-axis the unit measurements are: 10 years, 5 years, year, quarter (e.g. Jan01 means the first quarter in 2001) and for the Y-axis: 5 euro, 1 euro, 10 cent, 5 cent, 1 cent.

3.2 Fisheye Distortion

The concept of a fisheye lens for computer interfaces was first introduced in 1986 [17]. It is based on the distortion of a wide-angle lens that shows the focus in detail and the remote regions in progressively less detail. Compared to the ZUI, the fisheye has the advantage that both detail and context can be integrated in a single view at the same time.

Fisheye interfaces have been implemented in a variety of styles. One example for a PDA application is DateLens [7]. It shows a tabular calendar interface, in which cells that represent dates are enlarged or minimized according to the user's focus. The same approach has been used in [24] to provide multiple views of application data. The following section, however, describes the first attempt to apply a fisheye distortion to a scatterplot.

3.2.1 Interaction Design

To remove visual clutter in the fisheye scatterplot, users first hit the box-icon button below the diagram. They can now draw a bounding-box denoting the region of interest (Figure 3a). As soon as the pen is lifted from the display, the application centers the focus region and magnifies it in a smooth animation (600 milliseconds) to about 75% of the diagram size (Figure 3b). The surrounding regions contract. Objects that are not in focus are still visible but are allocated less space. In contrast, items in the focus region drift apart. Depending on the given information density, users may have to apply the fisheye distortion recursively to isolate items (Figure 3c). Panning is implemented as drag&drop, i.e. dragging the diagram with the pen in the direction opposite to that in which they want to move the viewport. This more conventional alternative to the sliding metaphor in the ZUI gives users more control but is also slower. By panning, users can move items in and out of focus.

Compared to the ZUI, the fisheye provides a better orientation by contracting and not clipping regions that are currently not of interest. That way, users can make informed decisions about where to navigate. By looking at the context, they may for instance prevent themselves from panning into regions where there are no items located. Moreover, it is possible to move the focus directly into a context region without having first to zoom out. Users just need to draw a new bounding-box at the desired position. Users can undistort the diagram by tapping and holding the button with the magnifying glass and the minus sign. If the button is only tapped but not held, the distortion settings of the previous step, if any, are restored.

To access the details of book items, users tap in its vicinity. The item closest to the pen position zooms to full screen in 250 milliseconds and presents the book information on the familiar record card representation (Figure 3d). Note that, unlike the ZUI, only the single item is magnified, not the entire diagram. On tapping the record card, the item zooms back to its original 1-pixel size and position. This mechanism is different to a pop-up in that the animation leads the user's eye back to the previously focused point in the diagram. Thus reorientation is supported. Moreover, users can tap anywhere on the record card to remove it and do not need to hit an X-button in a fixed position. The details of any non-overlapped item can be accessed, regardless of whether the interface is distorted or not, and regardless of whether the item of interest is located in a focus or context region.

3.2.2 Implementation

The algorithm used for distorting the interface is similar to orthogonal stretching as described in [29]. It has the advantage that many properties of the original layout are preserved for the benefit of the user's mental map of the interface. However, in our case, the stretching is applied to the diagram pane and not to the objects. While screen space is reallocated during distortion to magnify the focus, the "compass direction" of the items remains intact. The most important property, however, is that the distortion does not affect parallelism between lines. Unlike for instance when using radial distortion, items can still be mapped in straight lines to the diagram labels. Hence the algorithm proves especially useful for distorting a scatterplot interface.

During the distortion animation, the scatterplot labels are continuously updated. The same unit measurements are used as listed in Section 3.1.2. To avoid confusing overlapping of labels, the context regions are only labeled with the start or end unit of the axis.

4 EXPERIMENT

We have presented two very different interaction strategies to explore scatterplot interfaces with a high information density on small

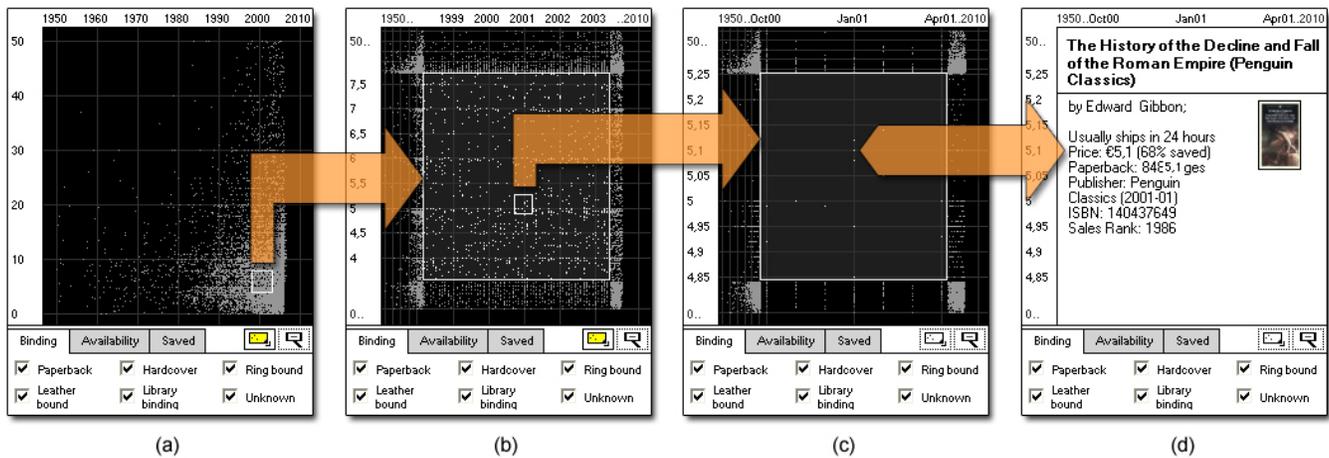


Figure 3: fisheye interface: a and b show a recursive distortion to remove visual clutter; in c the user taps an item to access its details; d shows a record card representation.

screens. To compare the interfaces in terms of usability, we conducted an experiment in which users had to complete various search tasks on a collection of 7,500 books. We chose this data set with respect to the scenario mentioned in the introduction: a field sales person is browsing a large product catalogue on a mobile device. The test focused on efficiency (task completion time) and subjective user ratings (system preference and user satisfaction). This section describes our hypotheses and experimental settings.

4.1 Interfaces

To provide the participants with more realistic search features and to allow us to emulate different levels of information density, we enhanced the scatterplot with a set of dynamic query modules [4] (see Figure 2, 3 below the diagram window). By ticking checkboxes, users could filter out books based on binding type or availability. Next to that, they could also control the relative amount of money they wished to save compared to the current list price of each book. For this continuous attribute, we added a range slider that was also used to visualize the distribution of books against savings. When ticking the checkboxes or moving the slider, the diagram is rapidly updated, hiding those items that did not satisfy the filter requirements.

4.2 Materials

The applications were implemented using the Microsoft .Net and the .Net compact framework 1.1. Since current mobile devices are not powerful enough to cope with memory-intensive applications, we ran the test on a Wacom Board connected to a 3 GHz Pentium 4 PC with 1 GB RAM. This allowed us to use pen-interaction while also being provided with sufficient processing power. As shown in Figure 4, the application interface took up a portion of 240 x 320 pixels of the Wacom Board display, mimicking the standard PDA resolution. Next to the screen-recording software, which also recorded the audio stream, we logged task completion times and zooming and panning attempts directly within the application. To measure user satisfaction with regard to functionality, we used the pragmatic quality (PQ) dimension of the Attrakdiff questionnaire [20], which consists of seven items on a 7-point semantic differential. We supplemented the Attrakdiff with two questions regarding the quality of navigation and orientation features of the two interfaces, both of them measured on a 7-point scale. System preference was also measured with additional questions, directly asking

the subjects which of the two interfaces they preferred and furthermore if they thought that the less preferred interface was nevertheless superior for certain tasks. We also asked which interface they thought was more efficient in terms of task-completion times and which of the two panning techniques (sliding versus the conventional drag&drop) they preferred. A pre-test questionnaire was used to collect demographic data.



Figure 4: simulated PDA interface on a Wacom Board.

4.3 Hypotheses

The hypotheses were:

1. *Task-completion time would be better for the fisheye interface.* To date, there have been only very few studies comparing the usability of fisheye views and ZUIs. One is [30], in which a fisheye view was contrasted with a traditional full-zoom view on hierarchically clustered networks. It was found that subjects using the fisheye interface were significantly faster at completing tasks and made fewer unnecessary navigational steps through the hierarchy. The authors concluded that the fisheye view helped the subjects to concentrate directly on the task itself, resulting in quicker navigation and less unnecessary exploration. Another study analyzed the usability of a fisheye technique under the additional constraint of a limited screen real estate [19]. Again the authors found that the fisheye was faster for navigation tasks, while a simple two-level zoom improved the completion time for monitoring tasks. For the scatterplot interface,

we expected to obtain similar results. Users would need to perform less unnecessary navigation, which would compensate for the more time-consuming distortion technique of the fisheye view. On the other hand, tasks that require direct access to the information of a single item would be completed significantly faster when using the geometric-semantic ZUI.

2. *Users would prefer the geometric-semantic ZUI to the fisheye interface.* At first glance this statement may seem to contradict the first hypothesis but it is based on the experience that interface efficiency does not necessarily correlate with user preference. Studies that compared ZUIs or 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 [19, 21, 5]. It appears that the rather artificial distortion may discourage users and thus decrease user satisfaction. Moreover, we assumed that the geometric-semantic ZUI would be preferred for its hedonic qualities. Due to the fly-through and slide metaphors, it reminds one in some ways of a computer game.

4.4 Tasks

During the experiment, and for each of the two interfaces, users had to solve a different set of 10 tasks after four training tasks. A task set comprised three different types of questions: A. *Visual Scan (2)* For example: how many books have been published since the year 2000 at a price of EUR 30? B. *Information Access (4)* Who is the author of the most expensive book published in the year 2005? C. *Comparison of Information objects (4)* Between August and November 2001 four books were published which are available at a price of EUR 8.53. Which is the one with the most pages?

4.5 Participants

For the study we selected 24 subjects, 11 male and 13 female. 23 of them were students at the University of Konstanz. Their ages ranged from 19 to 33 years. The other subject was an engineer and aged 50. The fields of study varied, with psychology students (7 subjects) being the largest group. Only one participant was a PhD student in computer science. The pre-test questionnaire also revealed that two of our subjects actually owned a PDA and a further ten had at least tried one and were therefore familiar with the general pen interaction concept. All of our users were regular PC and internet users. In order to test familiarity with zoomable user interfaces, we asked for google earth knowledge. Seventeen subjects were aware of google earth but only one had actually used it.

4.6 Design

We used a counter-balanced within-subjects design, balancing the two interface types and task-sets. This resulted in four different groups mirroring all possible variations of interface and task-set order. We randomly assigned six subjects to each group. For analysis, we mainly used repeated measures ANOVAs (RM-ANOVAs). Our independent variable was interface type (geometric-semantic ZUI and fisheye interface). The dependent variables were task-completion time (in seconds), system preference (geometric-semantic ZUI or fisheye interface), user-satisfaction (Attrakdiff PQ Scores), error-rate (number of incorrectly answered tasks) and navigation actions (panning and zooming attempts). Because of the different interaction techniques, we included navigation actions mainly for the sake of completeness but did not expect to be able to compare the two interfaces on the basis of this variable.

4.7 Procedure

The session started with a short written introduction and the pre-test questionnaire. Users were then introduced to the pen handling

on the Wacom Board, during which the pen was recalibrated by the subjects themselves. Next a tutorial video was shown, introducing the interaction of the first interface. After that, users had time to try the application on their own and ask questions. The test administrator asked subjects to try the interaction techniques as shown in the video. When participants showed that they had understood the interface, each of the fourteen task questions was presented to them successively as a printout. The first four questions were labeled as training tasks and subjects were still allowed to ask questions while working on them. Participants read all questions aloud and then pressed a "start task" button on the upper left of the screen. When they felt they could answer the question, users tapped a "say answer" button below the "start task" button. When the training tasks had been completed, no further support was given. After answering the questions in the first task-set, subjects completed the Attrakdiff questionnaire. Subsequently, the same procedure was repeated for the second interface and the second task-set. At the end, subjects also completed a preference questionnaire and were then given a movie theatre voucher worth EUR 10. Experiment sessions lasted about 60-75 minutes.

5 RESULTS

This section describes the results of the study, with the focus on our two hypotheses. During the analysis we discovered a significant interaction effect between system order, task-set order and interface type. Further analysis traced this effect back to a single task in one of the two task sets, which unintentionally resulted in a bias towards one of the interface types. We therefore excluded this task from further analysis. Moreover, we also excluded one subject from our task time analysis since their total task completion time was identified as an outlier ($> 3 * St.Deviation$).

5.1 H1: Task Completion Time

In our first hypothesis, we assumed that participants would be able to solve tasks significantly faster when using the fisheye interface. However, analysis revealed that it took users about the same time to complete the 10 tasks with both interfaces (623.8 seconds for the geometric-semantic ZUI compared to 612.4 seconds for the fisheye interface, see Figure 5a). The small difference is not significant, $F(1,22) = 0,002$, $p=n.s.$, and therefore contradicts our hypothesis. We then analyzed whether the interfaces differed for the three task types. We suggested that the ZUI might be faster for Information Access tasks but slower for the two other task types. Figures 5b to 5d show that this is indeed the case, however the rather small differences are not significant. Regarding navigation, the fisheye interface required far fewer actions but, since task times are similar, it seems that they required more time to execute. Hence we assume that drawing a bounding box is cognitively more demanding than the more direct zooming of the ZUI. We also analyzed whether other variables such as gender had an influence on task completion time. It took our male subjects significantly less time to complete the tasks (geometric-semantic ZUI: 529 seconds compared to 646 seconds, $F(1,22) = 7.5$, $p = 0.012$; fisheye interface: 521 seconds compared to 650 seconds, $F(1,22) = 6.9$, $p = 0.016$). PDA experience, on the other hand, showed no effect on task completion time. Our post-test questionnaire included one question where subjects were asked to choose the interface with which they thought it had taken them less time to complete the tasks. About 30% guessed incorrectly. This is not surprising, however, bearing in mind the small differences in the overall task times (in most cases between one and two minutes per task-set). The interfaces did not differ in terms of task accuracy - in both cases, more than 90% of correct answers were given on average.

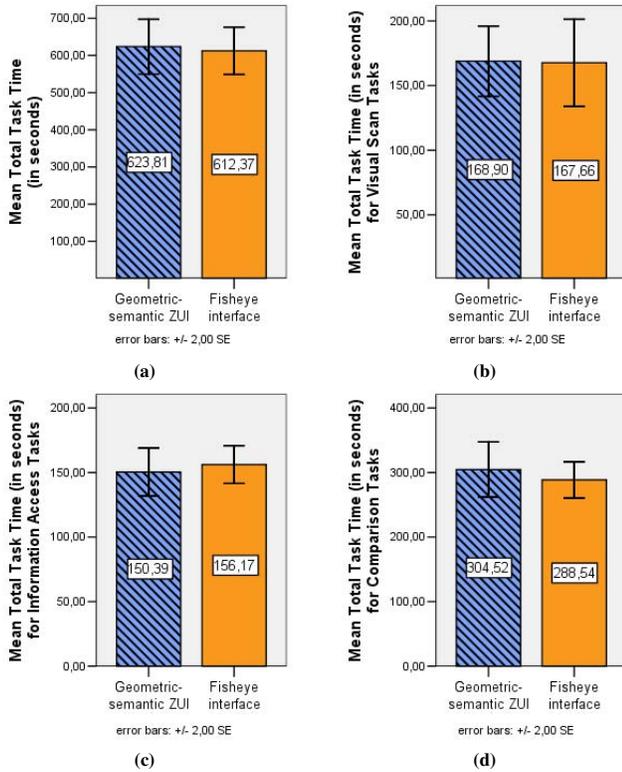


Figure 5: (a) Total Task Time, (b) Visual Scan, (c) Information Access, (d) Comparison

5.2 H2: Preference + Questionnaire Results

Our second hypothesis suggested that users would prefer the geometric-semantic ZUI to the fisheye interface. However, the results obtained strongly contradict this hypothesis: 20 subjects preferred the fisheye interface and only three the ZUI ($X^2(1, N = 23) = 12.565, p < 0.001$). In the search for reasons we first analyzed our Attrakdiff results. As can be seen in Figure 6, users rated the fisheye interface significantly better; 5.11 compared to 4.11, $F(1, 23) = 20.84, p < 0.001$. Furthermore our two additional questions regarding navigation and orientation features were clearly rated in favor of the fisheye interface (navigation: 5.79 compared to 4.83, $F(1, 23) = 9.6, p < 0.01$; orientation: 5.7 compared to 4.7 $F(1, 23) = 9.9, p < 0.01$). In a second step we took a closer look at the statements users gave as the reasons for their preference. Most users mentioned that the fisheye interface offered a better orientation (9 subjects), that drawing bounding boxes was the easier way to get to a certain area (8 subjects) and that it allowed a faster task completion (6 subjects). Furthermore, seven subjects mentioned that they had problems with the sliding technique of the ZUI. We assume that this was mainly caused by the users' unfamiliarity with this kind of panning. As a result some subjects accidentally triggered a zoom operation when actually trying to slide. Nevertheless, 18 subjects could think of a task where the non-preferred interface would be better. In the group that preferred the fisheye interface, seven subjects stated that in the ZUI it was easier to compare books that were spatially close. Other statements were that it was easier to access a single book (3 subjects), that the zooming actually enlarged the books (3 subjects) and one subject mentioned that the semantic zoom offered the possibility to discover other books of potential interest while navigating to a certain area. Regarding the group that in general preferred the ZUI, two subjects preferred the fisheye interface for tasks where they had to access a given area of

interest. Comparing the panning techniques, our subjects preferred the more conventional drag&drop mechanism over the sliding technique ($16:7, X^2(1, N = 23) = 3.522, p = 0.061, n.s.$).

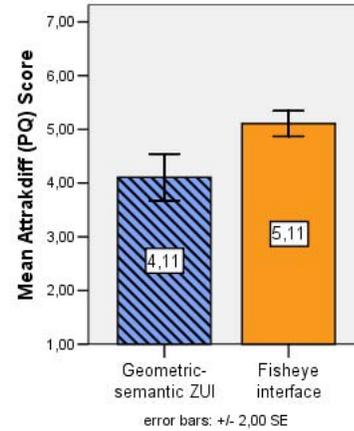


Figure 6: comparing Attrakdiff PQ Scores between interface types

6 CONCLUSIONS

In this paper we presented two interaction techniques for view decomposition and detail access of a scatterplot interface on small screens. One technique is based on the metaphor of a zoomable user interface. It allows users to control the ratio of overview and detail information presented on the screen in a continuously adjustable fashion. The fisheye interface on the other hand is based on a distortion algorithm that integrates both focus and context in a single view. We conducted an experiment that involved searching a book database on a simulated PDA screen using the two techniques and found that participants significantly preferred the fisheye view to the ZUI. User statements suggest that this result is due to the better orientation features and the more precise navigation offered by the interface. This finding is important because it contradicts previous research in which fisheye views have been found inferior to other interfaces in terms of user preference. We assume that fisheye techniques may integrate better with abstract information spaces such as diagrams, while user satisfaction decreases with domains such as maps, in which a higher degree of fidelity to the standard layout is essential [28]. Our results may also indicate that, when using a small screen, users place higher value on the ability to preserve navigational context.

In contrast to our expectations, we did not find a significant difference in task-completion time between the interfaces. Due to the context view, the fisheye interface required less navigation, but this advantage could not compensate for the slower interaction mechanism compared to the direct-zoom feature of the geometric-semantic ZUI.

7 ACKNOWLEDGMENT

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

REFERENCES

- [1] Christopher Ahlberg. Spotfire: an information exploration environment. *SIGMOD Rec.*, 25(4):25–29, 1996.
- [2] Christopher Ahlberg and Ben Shneiderman. 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*, pages 313–317, New York, NY, USA, 1994. ACM Press.
- [3] Christopher Ahlberg and Ben Shneiderman. Visual information seeking using the filmfinder. In *CHI '94: Conference companion on Human factors in computing systems*, pages 433–434, New York, NY, USA, 1994. ACM Press.
- [4] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, New York, NY, USA, 1992. ACM Press.
- [5] Patrick Baudisch, Bongshin Lee, and Libby Hanna. 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*, pages 133–140, New York, NY, USA, 2004. ACM Press.
- [6] Benjamin B. Bederson and Angela Boltman. Does animation help users build mental maps of spatial information? In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 28, Washington, DC, USA, 1999. IEEE Computer Society.
- [7] Benjamin B. Bederson, Aaron Clamage, Mary P. Czerwinski, and George G. Robertson. Datelens: A fisheye calendar interface for pdas. *ACM Trans. Comput.-Hum. Interact.*, 11(1):90–119, 2004.
- [8] Stefano Burigat and Luca Chittaro. 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*, pages 277–284, New York, NY, USA, 2005. ACM Press.
- [9] Thorsten Buring, Jens Gerken, and Harald Reiterer. 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*, pages 233–240, New York, NY, USA, 2006. ACM Press.
- [10] Thorsten Buring and Harald Reiterer. Zuiscat: querying and visualizing information spaces on personal digital assistants. In *MobilityHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 129–136, New York, NY, USA, 2005. ACM Press.
- [11] Gunjan Dang, Chris North, and Ben Shneiderman. Dynamic queries and brushing on choropleth maps. In *IV*, pages 757–764, 2001.
- [12] Alan Dix and Geoffrey Ellis. Starting simple: adding value to static visualisation through simple interaction. In *AVI '98: Proceedings of the working conference on Advanced visual interfaces*, pages 124–134, New York, NY, USA, 1998. ACM Press.
- [13] M. D. Dunlop and N. Davidson. Visual information seeking on palmtop devices. In *Proceedings of HCI2000, volume 2*, pages 19–20, 2000.
- [14] M. D. Dunlop, A. Morrison, S. McCallum, P. Ptaskinski, C. Risbey, and F. Stewart. Focussed palmtop information access through starfield displays and profile matching. In F. Crestani, M. Jones, and S. Mizzaro, editors, *Proceedings of workshop on Mobile and Ubiquitous Information Access, LNCS v2954*, pages 79–89. Springer, 2004.
- [15] Jean-Daniel Fekete and Catherine Plaisant. Interactive information visualization of a million items. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis '02)*, page 117, Washington, DC, USA, 2002. IEEE Computer Society.
- [16] Anna Fredrikson, Chris North, Catherine Plaisant, and Ben Shneiderman. Temporal, geographical and categorical aggregations viewed through coordinated displays: a case study with highway incident data. In *NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation*, pages 26–34, New York, NY, USA, 1999. ACM Press.
- [17] G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM Press.
- [18] Jade Goldstein and Steven F. Roth. Using aggregation and dynamic queries for exploring large data sets. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 23–29, New York, NY, USA, 1994. ACM Press.
- [19] Carl Gutwin and Chris Fedak. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 145–152, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
- [20] Marc Hassenzahl, Axel Platz, Michael Burmester, and Katrin Lehner. Hedonic and ergonomic quality aspects determine a software's appeal. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 201–208, New York, NY, USA, 2000. ACM Press.
- [21] Kasper Hornbæk and Erik Frøkjær. 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*, pages 293–300, New York, NY, USA, 2001. ACM Press.
- [22] N. K. Jog and B. Shneiderman. Starfield visualization with interactive smooth zooming. In *Proceedings of the third IFIP WG2.6 working conference on Visual database systems 3 (VDB-3)*, pages 3–14, London, UK, UK, 1995. Chapman & Hall, Ltd.
- [23] Susanne Jul and George W. Furnas. Critical zones in desert fog: aids to multiscale navigation. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 97–106, New York, NY, USA, 1998. ACM Press.
- [24] Amy K. Karlson, Benjamin B. Bederson, and John SanGiovanni. 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*, pages 201–210, New York, NY, USA, 2005. ACM Press.
- [25] Amir Khella and Benjamin B. Bederson. Pocket photomesa: a zoomable image browser for pdas. In *MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pages 19–24, New York, NY, USA, 2004. ACM Press.
- [26] Heidi Lam and Patrick Baudisch. Summary thumbnails: readable overviews for small screen web browsers. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 681–690, New York, NY, USA, 2005. ACM Press.
- [27] Ken Perlin and David Fox. Pad: an alternative approach to the computer interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64, New York, NY, USA, 1993. ACM Press.
- [28] Catherine Plaisant, David Carr, and Ben Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Softw.*, 12(2):21–32, 1995.
- [29] Manojit Sarker, Scott S. Snibbe, Oren J. Tversky, and Steven P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 81–91, New York, NY, USA, 1993. ACM Press.
- [30] Doug Schaffer, Zhengping Zuo, Saul Greenberg, Lyn Bartram, John Dill, Shelli Dubs, and Mark Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Trans. Comput.-Hum. Interact.*, 3(2):162–188, 1996.
- [31] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343. IEEE Computer Society, 1996.
- [32] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, U.S.A., 1983.
- [33] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *SIGIR '92: Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 338–346, New York, NY, USA, 1992. ACM Press.