Flip Zooming
The Development of an Information Visualization Technique

Staffan Björk

1 Introduction

This thesis describes the development of an information visualization technique, *Flip Zooming*. The goal has been to explore how an interactive presentation technique for computers can be applied to display different types of information in different areas of use. The work consists of the development of the technique itself, and a number of distinct applications (the terms *visualizations* will be used in the following) using the technique. In addition, the thesis offers reflection, which has been stimulated by the development and applications of the Flip Zooming technique, on the information visualization research field as a whole. The thesis will show that information visualization is highly dependent on the data to be presented and on the purpose for which the data is displayed. This means that the technique is presented, and should be seen as, a framework of guidelines and methods rather than as a predetermined formula that one can apply to a given set of criteria to produce an optimal visualization.

1.1 Representations Using Computers

The invention and realization of computers in the 20th century have greatly influenced our work and lives. They are among the most complex devices created by humans, and are the inanimate devices that are easiest to conceive of as being able to interact with humans.

In order to be interactive and adaptive, a computer must be able to receive input, process it, and present the result. The importance of these abilities can be seen in the development of computer systems. Contemporary computers are described in terms of how these abilities have

improved: they have hard drives and cd-roms from which information can quickly be obtained, extremely fast processors that can manipulate large pieces of information at a time, graphical displays with high resolutions to present information, fast network connections to world-wide networks, all controlled through *Graphical User Interfaces* using *Windows-Icons-Menus-Pointers (WIMP)* styled interaction to provide efficient information processing.

But how should computers receive information? In *Computers in Context* [17], Dahlbom and Mathiassen at length discusses the problem of how computers should receive information, or how to transform knowledge into information and information into data. They state that codifying knowledge "is a difficult task as soon as we aim beyond anything but the most formalized and routinized type of knowledge." [17, p. 33]. An information system, no matter how effective at manipulating data, is of no help – or may even be counter-productive – to the user if the relevant knowledge and correct information has not been encoded into the system.

But, even if the computer has received the correct information and has processed it correctly, the information is not yet useful to the user. Unless the data can be translated back into information, and that information can be the basis for new knowledge, any interactive computer application fails its task. This problem has become especially pronounced with the large storage space available on present day computers, and the speed with which they can manipulate the data stored within. Both these factors have increased dramatically in recent years (and still do), without a similar rate of growth in the effectiveness of presenting the result of those computations. As more and more computers become connected together in networks such as the Internet, this problem becomes even more apparent.

Unless one takes care in how the information is displayed visually, one risks an incomprehensible presentation, making the information presented by the computer worthless to the user. Facilitating presentations that are intelligible and comprehensible is the goal of *Information Visualization*.

2   Information Visualization

The first use of the term information visualization can be found in the work of Stuart Card *et al*., where it is described as an application area for the *Cognitive Coprocessor Architecture* [72]. This architecture was designed to support the problems of multiple interacting agents and supporting smooth animation in highly interactive interfaces. Given this frame of reference, information visualization was defined as a domain where "2D and 3D animated objects (or visualizations) are used to represent both information and the structural relationships of information. Direct manipulation of these objects causes changes in the actual structure of the information or changes in the actual information" [72, pp. 11-12].

Although this description was initially used for the aim of staking out an application domain for visualizations in the *Information Visualizer* [11] system, the term has come to signify a larger scientific research field. Especially, work done earlier by Furnas [22] and Spence & Apperley [83] has been seen as some of the earliest examples of information visualization (even work done as early as in the 70's can be said to be using information visualization, c.f. [20, pp. 26-34]). As the use of the term has expanded, a newer definition of information visualization has been given:

> *The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.* [10, p. 7]

Using the information processing powers of computers, and especially the ability to quickly recompute the presentation, information visualization is seen as a way of offering users the ability not only to receive a clear and comprehensible presentation, but to modify it at will to highlight points of interest.

In the definition above, information can be seen as simply meaning abstract data. However, both the definitions of the terms, as well as the distinction between them, are problematic and can be stated in various forms. Dahlbom and Mathiassen explains information, in a self-admittedly vague form, as: "Information is something that gives knowledge, something that is related to the communication, transmission, or dissemination of knowledge" [17, p. 25]. They give a slightly more formal definition of data: "*Data* are a formalized representation of information,

making it possible to process or communicate that information" [17, p. 26]. This thesis uses the terms in a related fashion: *data* is used to describe what computers manipulate whereas *information* is taken to be what a human perceives from a comprehensible presentation. The use may, however, differ somewhat from paper to paper.

The stress on abstract data in the definition of information visualization comes from the wish to distinguish the research field from the related field of scientific visualization, which typically presents views of data collected from sensors or mathematical models. The two fields in many cases overlap, and can be seen as parts of (computer) *visualization*, or the "use of *computer-based*, *interactive* visual representations of data to amplify cognition" [10, p. 7].

Even though the definition given above describes the goal of information visualization, it does not state how to achieve this goal nor does it answer the question: what factors can information visualization techniques influence to ease understanding? Looking at the research field of Informatics, one can find one possible answer to the question in *The Infological Equation*. Börje Langefors, who first proposed the equation [REF], describes it as a conceptual equation with the following definition:

$$\mathbf{I} = \mathbf{i}\,(\,\mathbf{D}\,,\mathbf{S}\,,\mathbf{t}\,)$$

Here **I** stands for the information (or knowledge) produced using the interpretation process **i** on the data **D** during the time **t**, given the pre-knowledge **S**. Even though its various components have been debated and more detailed variants have been proposed (c.f. [15]), the original definition is sufficient for the purpose of describing information visualization.

Using the equation, information visualization can be described as increasing the expected value of **I** by creating interactive visual presentations that (1) decrease the time **t** needed, so that the process can be repeated using other data using the saved time, (2) lower the requirements of users' pre-knowledge **S**, thus increasing the number of potential users, or (3) make the interpretation process **i** more effective by specializing the visualization for a particular data type **D** or use situation. In all cases, the goal is to make the information available, and comprehensible, with as little effort as possible for the user.

Most research on information visualization has been conducted with the aims of (1) or (2). Reasons for this may be that the potential gain

seems greater, i.e., the techniques may be used for large groups of data types, and can quickly be learned by novice users. Time is measured quantitatively, meaning that experiments aimed at reducing **t** can collect "hard" figures from experiments to give statistical evidence for the technique. The wish to lower the requirements of **S** can at least partly be ascribed to the influence from the research on human-computer interaction, which has long focused on making computer systems easy to use in the sense that they should be intuitive and their use self-evident.

As no information visualization techniques claims to be able to show all types of information, one can say that, to a certain degree, all have the aim of making **i** more effective by specialization (3). However, most techniques are often adapted for more general types of information (cf. graphs, trees or text) after being successful used in a specific area of use, in order to expand the usefulness of the technique. This is in contrast to specializing the technique to provide dedicated support for more specific type of information such as subway maps, computer programs, or personal diaries.

2.1   A Note on Artifacts and Cognition

One can ask why computers should be used to create presentations of information where the underlying data has been processed. Can we not just use our own minds to mentally perform the same changes in a presentation? To transform the presentation mentally can be described as operating on a *mental model*. Norman [64] states that these "models are essential in helping us understand our experiences, predict the outcomes of our actions, and handle unexpected occurrences. We base our models on whatever knowledge we have, real or imaginary, naive or sophisticated." [64, p. 38]

However, although the human mind is very good at manipulating mental models in certain ways (e.g. recognizing faces and making use of associations), manipulation of a logical or mathematical nature is difficult for most people. This kind of thinking is easier to perform if one has appropriate tools such as pen and paper. As Norman states, "The powers of cognition come from abstraction and representation: the ability to represent perceptions, experiences, and thoughts in some medium other than that in which they have occurred, abstracted away from irrelevant details." [65, p. 47].

5

Cognitive artifacts are artifacts that help the processes of human thinking by representing the above mentioned perceptions, experiences, and representations in an abstract form. Seen simply as storage devices, computers can be seen as one of the most powerful instances of this type of artifact. However, computers can also manipulate information for us, and on the basis of that information, initiate actions and processes. Having such abilities, computers are not only judged by how well they store and present information, but how well they process information. They can, therefore, be said to not only be cognitive artifacts but *information-processing systems* and "adaptivity to an environment is their whole *raison d'être*." [82, p. 22].

Information visualizations can be seen as cognitive artifacts residing within the computer that processes information. As such, they can be manipulated to change the representation of data to highlight interesting objects and characteristics, provide overviews, and show information structures. In doing so, they can adapt to suit the user's current activity. Norman makes a similar point when talking about cognitive artifacts in general: "The critical property of the representations supported by cognitive artifacts is that they are themselves artificial objects that can be perceived and studied. Because they are artificial, created by people, they can take on whatever form and structure best serves the task of the moment." [65, p. 51]

Card *et al.* identify information visualizations as techniques to amplify cognition [10]. Based on a study of visualizations of static diagrams [53], they identified six major categories in which information visualization can amplify cognition: by *increasing resources* available; by allowing a *reduced search* cost of those resources; by enabling *enhanced recognition of patterns* within the resources; by enabling *perceptual inference* operations; by allowing *perceptual monitoring* of the resources as they change over time; by giving the user a *manipulable medium* where exploration of the resources can be done using computational tools to change the appearance of the visualization [10, p. 16].

Thus, the use of information visualization offers a possibility to enhance human cognition and understanding. By externalizing not only the representation of the data displayed, but also some of the manipulation of the data, we can gain powerful tools to help the process of reasoning. The transformed manipulation can, by using the same kind of techniques that humans are good at when manipulating mental models (e.g. perceiving objects seen at an angle), be made easy or even trivial to

6

comprehend. By thus allowing the manipulation of not only the data but also the presentation of the data to be made by computational devices, the solution to problems may become self-apparent when presented correctly. Talking about problem-solving in general, Simon elegantly states the basic underlying assumption of information visualization: "Solving a problem simply means representing it as to make the solution transparent." [82, p. 132]

3   Background

Information visualization is an engineering-oriented discipline, and as such it builds its theoretical frameworks from evidence present in implemented systems (as well as from theories developed in other sciences). Therefore, it is motivated to give an overview of some of the most noteworthy information visualization techniques before describing the theories and methods developed in the discipline. When describing these techniques, other techniques that have later been influenced by them, or otherwise have similarities, will be mentioned.

Some areas of information visualization will not figure in the following overview. Not being relevant to the work of the thesis, these include various forms of refined or enhanced GUI components and ways of adjusting visualizations due to interaction with them (e.g. Dynamic Queries [81]), and techniques that rely on the movement of the user's viewpoint within a 3-dimensional space (i.e. the various forms of Virtual Reality systems). Further, methods where the presentation of "objects" is spread out over several separate areas of the space, or parametrically, will not be included (e.g. Parallel Coordinates [38] and the Influence Explorer and Prosection Matrix [91]).

## 3.1   Early Techniques: Fisheye Views and the BiFocal Lens

As mentioned earlier, the work by Furnas on *Fisheye Views* [22], is usually seen as one of the earliest examples of information visualization. Based on the notion that an *a priori* rate of importance has been imposed on the various parts of an information structure, and that the user controls a focus of interaction, a numerical value can be calculated for each piece of information. Using a threshold value, the information can then be filtered to only show those pieces that are either of global importance or important due to their proximity to the focus. By having an unaltered presentation near the user's interaction focus and an altered presentation outside the focus, a view similar to that produced by a fisheye lens is achieved. Describing the technique, Furnas introduced the important terms of *focus*, *Level of Detail (LOD)*, and *Degree of Interest (DOI)*. It should be noted that although the fisheye technique is described using the metaphor of a lens, the technique was initially used on text-based presentations (see for example the work on *SuperBook*

[18], a system for browsing hypertext documents). Greenberg *et al.* constructed the *Fisheye text viewer* [28], and used multiple foci to inform users of workspace awareness in shared electronic workspaces [29].

A more recent and concise description of the notions proposed by Furnas can be found in *Generalized Fisheye views* [23]. A similar approach, *Fractal Views* [51], makes it possible to set (approximately) the number of information pieces shown purely from a Fractal Value function, at the risk of not displaying the whole path from the focus to the root of the information structure.

An application using a fisheye view is highly dependent on which a priori importance rating is used on its information. When working on structured information, e.g. hierarchical structures, the problem of creating a rating system is trivial, but in the case of less structured information, e.g. directed graphs, a less straight-forward degree-of-interest function must be chosen. In the work of *SemNet* [19], a graphical visualization of large knowledge bases using a 3-dimensional space, Fairchild *et al.* identify three possible techniques: Clustering to form hierarchical structures; using 3-dimensional perspective to create an automatic balance between focus and context; or by sampling the available information to create a view according to a density function with a maximum at the focal point.
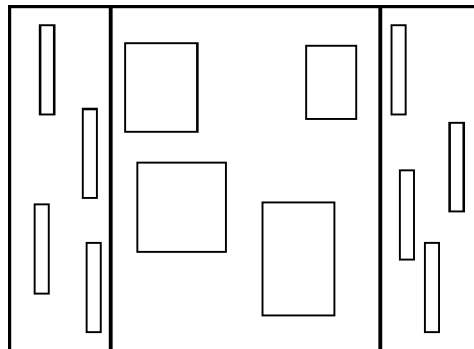

Fig. 1. Schematic figure of the Bifocal Lens.

Arriving independently at similar conclusions to those of Furnas, the work on the *BiFocal Lens* [83] by Spence and Apperley starts from an altogether different viewpoint. Designing an electronically enhanced office, the BiFocal Lens was one part of an integrated environment

9

where individual components should be connected to provide a smooth working environment with seamless transferal and manipulation of information. To show more information on a display than is possible using a "windowing system" where one only sees a small area of the display through a "window" (not to be confused with the use of windows in graphical user interfaces), the Bifocal Display used a composite structure consisting of one central un-compressed view and two horizontally compressed views to the left and right of the central view (see **Figure 1**). Being implemented on a text-only display, each piece of information had to be presented in two different versions and there was no difference in compression near the central view and far away from it. Thus, it can be said to have two levels of focus, and provide a bifocal view. Describing the system, Spence and Apperley introduced the terms of *focus*, *data context*, and *information levels*.

The work with Fisheye Views and BiFocal Lenses identified the possibility to use several different modes of representation for individual pieces of information (in the case of FishEye Views, one mode is to not show the piece at all). By varying the modes of the different pieces, a multitude of presentations of the same information becomes possible, putting emphasis on different areas or regions of the information. Taking information from the user's interaction with the system (i.e. navigation in the information space), an algorithm can change the modes of the various information pieces in a coherent fashion, creating a visual presentation that adjusts its presentation to the user's preferences.
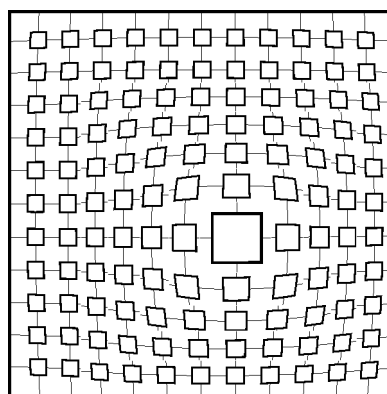
Fig. 2. Schematic figure of the Graphical Fisheye Views.

10

Graphical Fisheye Techniques. Based on the ideas of Fisheye views, Sarkar and Brown developed *Graphical Fisheye views* [76,77]. These showed graphical presentations of graphs (including vector-based maps) where a normal layout was distorted by a user-controlled focus to increase detail within the focused area (see **Figure 2**). An important concept introduced in this work was the explicit notion of a normal layout strategy that was used as a basis for the distorted layout. Based on user experience, the distortion function was specialized depending on domain-specific information about the data to be shown. When showing information that was visually familiar to the users (e.g. maps), the initial transformation proved to give unnatural results due to the independent horizontal and vertical (Cartesian) transformation of positions. This was solved by introducing polar transformation where the dimensions are interdependent. A similar system, *CATGraph* [44], maintains the focus at the center of the display using a distortion transformation based on the asymptotic behavior of the ArcTangent function. It should be noted that the first description of using fisheye lens for computer presentations appears in a (unpublished) Ph.D. thesis [20]. Here, the DECR (Detail Enhancing Continuity Retaining) lens, with properties similar to a fisheye lens, is used to provide detailed views while retaining an overview of all information.

The work on graphical fisheye views has further been developed using the concept of a *Rubber Sheet* [78]. Here, the presentation area is seen as a stretchable 2-dimensional plane. The user magnifies areas of interest on this plane by selecting regions and "stretching" them to a desired size. A noteworthy advantage of the method is that not only individually identifiable objects (or clusters of them) can be chosen for magnification, but also arbitrary user-defined areas. A number of techniques that are visually similar to the rubber sheet have been developed. Kaltenbach [43] adopted a rubber sheet metaphor when designing techniques to solve screen management problems in hypertext systems. Schaffer *et al.* developed the *variable-zoom* method [79] to visualize hierarchical clusters in progressive detail. *VIKI* [80] used a multiple focus presentation developed from graphical fisheye views to create a spatial hypertext system.

The various graphical fisheye techniques showed the possibility of placing information in a space and to distort the pieces' positions. This is in contrast to the earlier fisheye views, where the presentation of

information depended on its "structural distance" from a user-defined focal point.

3.2   The Information Visualizer

As mentioned in the beginning of this chapter, the term Information Visualization was first coined in the work on the Cognitive Coprocessor Architecture [72]. The three components of the cognitive coprocessor architecture, a 3-Dimensional user environment called 3D/Rooms built on the earlier Rooms system [32], and various information visualizations were used to create a user interface called the *Information Visualizer* [11].

In the description of the user interface, the concept of *Information Workspace* was introduced to design the locality of different forms of information. The main idea behind the concept is that information can be differentiated into different storage areas based on the importance and frequency of use of individual pieces of information. Information that is often needed or of immediate use should be placed in an *Immediate Storage* area, where the cost to access the information (in measures of time and mental effort) is low. Less often used information can be stored in *Secondary Storage* areas, where it does not clutter the user's activities. Large piece of information or rarely used pieces are placed in *Tertiary Storage* areas. A description of one example of such a workspace can be found in the paper by Rao *et al.* [69], where many of the visualization techniques described below are briefly described, as well as a few techniques developed outside the Information Visualizer project.
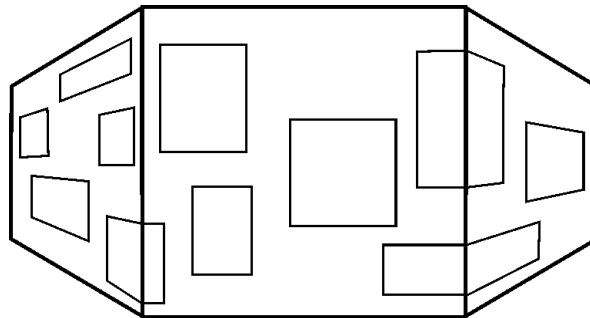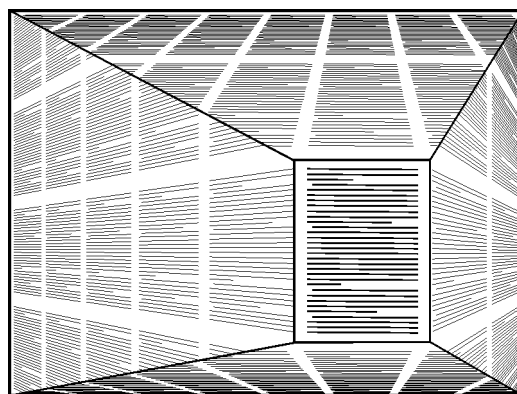
Fig. 3. Schematic figure of the Perspective Wall.

A great number of information visualizations which has influenced the research field was developed for the Information Visualizer. Similar in appearance to the BiFocal Lens, the *Perspective Wall* [57] uses a three-parted display using horizontal compression, and the inventors of the Perspective Wall acknowledge the BiFocal Lens as a conceptual ancestor to their technique [57, p. 176]. In the technique, the information to be displayed is shown on a wall with three visible segments: one centrally placed segment seen from straight ahead, and two adjacent segments seen from an angle (see **Figure 3**). Being implemented using a 3D graphical system, the compressed horizontal views in the Perspective Wall could be based on the same presentation as the central view by utilizing 3D transformations. A further advantage of using 3D transformations was that the information near the central view would be less compressed than information further away due to the effect of perspective.

Cone Trees [73] was designed as a companion visualization technique to the Perspective Wall. While the latter was developed to visualize information that had a serial structure (especially temporal sequences), Cone Trees was designed to visualize tree structures. Using a 3D space, the visualization laid out the nodes of the tree in a branching cone-like structure with the top node placed near the "ceiling" of the space and the leaves of tree place near the "floor". All nodes were arranged so that all children of a parent node were placed on the directrix (bottom circle) of a cone where the parent node constitutes the vertex (top point). One drawback of the technique is that, due to the 3D projection, some nodes can be hidden behind other nodes. To mitigate

13

this occlusion, every level of the Cone Tree could be rotated, making it possible to view all individual nodes. Interestingly, Cone Trees are stated to be more effective for visualizing unbalanced trees than balanced ones due to the difficulty of tracking rotation on a balanced tree. *Reconfigurable Disc Trees* [39] can be seen as a generalized form of Cone Trees, using discs as the basic component of the visualization. The use of discs allow the dynamic transformation of the visualization to create three different type of trees: disc trees that reduce the number of occluded nodes compared to Cone trees, compact disc trees which increase the number of nodes that can be displayed, and plane disc trees that can be mapped onto a 2D plane without visual overlap of the different nodes.

A later development of the Perspective Wall was the Document Lens [74]. Wishing to provide a visualization for information that had been placed in a rectangular 2D presentation, a moveable lens was introduced that magnified what it was placed over. As the shape of information to be visualized was assumed to be paramount (as is the case with e.g. text), the lens provided a linear magnification instead of a magnification similar to a optical fisheye lens (which would distort the magnified text). To avoid obscuring the information right next to the lens, the surrounding area was split into four trapezoid areas and distorted (using 3D transformations) to fit together as a whole. The final visualization can most easily be described as a truncated pyramid where the truncated part is the moveable lens (see **Figure 4**).

Fig. 4. Schematic figure of the Document Lens.

14

All the above techniques were developed to present information as 2-dimensional projections, even though they may use 3D perspective to achieve distortions of the projections. The "flatness" of the visualization was natural as the techniques were primarily intended for information that are of an inherently 2-dimensional nature. However, these techniques can be extended to work for 3-dimensional information [13]. One of the problems identified when applying the techniques on 3-dimensional spaces is the possibility of occlusion by objects between the viewer and the focus (or foci). A solution to this problem is a *Visual access distortion* [12] function that clears the line-of-sight by using a radially constrained repelling distortion. This technique allows for multiple foci by applying the visual access distortion functions in order of distance to the viewer, but it does not provide the user with a method for directly selecting occluded object as new foci.

The *Butterfly* visualization [59] was developed to visualize citation databases searches. Motivated by earlier work [70], the system was developed to present an interactive presentation that was manipulable while searches were being performed. Each article presented by the application is shown as a "butterfly", where the head contained information about author, publication data etc., the left wing the references in the article, and the right wing citers of the article. To further ease the use of the application, asynchronous query processes were automatically created to minimize response time.

Using a distortion function similar to that of the Bifocal Display, but with independent distortion in horizontal and vertical dimensions, the *Table Lens* [71] visualizes tabular information. Three independent methods of manipulating the distortion function enable the user to interact with the focus of the Table Lens: *Zoom*, which changes the proportion in which space is divided between focused area and context; *Adjust*, which modifies the amount of information being part of the focus without changing the size of the focus; and *slide*, which moves the location of the focus within the context.

The *Spiral Calendar* and the *Time Lattice* [58] were developed to visualize calendar information in the Information Visualizer. The Spiral Calendar progressively shows several calendars, ranging from calendars showing years to day and individual meetings, using an expanding counter-clockwise spiral structure. The Time Lattice uses a 3-dimensional space to place multiple peoples' calendars in a virtual space where they were presented collectively. Three shadows from the presen-

tation of the calendars are projected onto a floor and two walls, informing about time slots that are un-booked by all persons as shown, as well as showing the standard distributions of how individual people book their days and hours.

Apparent from the numerous examples, many different visualization techniques were developed by the researchers working with the Information Visualizer system. Besides showing a multitude of specific techniques, the work on the Information Visualizer gave the important the insight that information with structurally different attributes requires different information visualizations.

3.3 Treemaps & the Continuous Zoom: Visualizing Hierarchies

Treemaps [41] is an information visualization technique for trees, initially motivated to find ways of showing large directory structures on hard disks. Using a flat presentation area, Treemaps recursively partitions the available space among the nodes in rectangular slices (see **Figure 5**).

Differing from many other techniques for visualizing trees, which typically leave more than half the available space empty, Treemaps is a *space-filling* approach that uses all display area to visualize nodes. However, the children of a node receive all the space given to that node
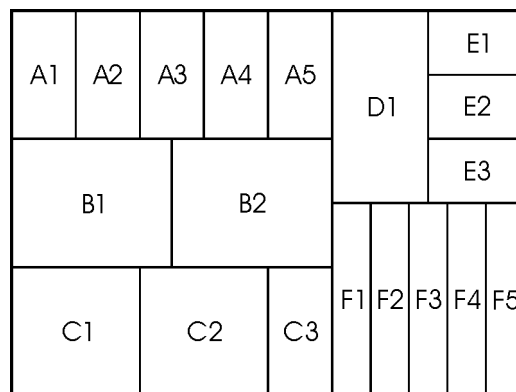
Fig. 5. Schematic figure of a Treemap. Elements starting with the same character belong to the same branch. A, B, and C belong to one main branch of the tree, while D and E constitute a sub-branch that together with F makes up the other main branch of the tree.

in the presentation, meaning that internal nodes are implicitly displayed. This has two implications: First, the structure of the tree must be made explicit by using a technique that does not require the display of internal nodes. This is solved by Treemaps by alternating between slicing horizontally and vertically as the available space is recursively divided, showing internal nodes as local groups "cut" in the same fashion. Second, as internal nodes of the tree are implicitly shown, Treemaps is limited to displaying tree structures where there is no information in the internal nodes.

By saving a piece of space at each node (called nesting by the authors), the information in the internal nodes can be shown at the cost of reducing the number of leaf nodes that can be displayed. However, it is possible that the visual presentation of the structure can become unclear by this addition, forcing further use of space to show the structure explicitly.

Examples of specializations of Treemaps for particular areas of use include the Tennis Viewer [40] and stock market visualizations [93]. Later refinements of Treemaps have included 3D shading to ease readability [92].

Also visualizing tree structures, the *Continuous Zoom* [2] is not space-filling but allows the user to control the size of individual nodes and clusters of nodes. Further, clusters of nodes can be "closed", so that they are effectively pruned from the presented hierarchy.

Closely related to tree structures are directed acyclic graphs. By using a fisheye view, Furnas and Zacks were able to adapt visualization techniques designed originally for trees to work on *Multitrees* [25], special cases of directed acyclic graphs that can be transformed into hierarchical structures which are not trees but where every child of a node forms the root of a tree.

Treemaps and the Continuous Zoom present hierarchies using *enclosure*, i.e. showing the structural properties of the visualized tree by putting children of nodes within the parent node. In contrast, Multitrees (as well as Cone Trees [73] described on page 13) use *connection*, where children nodes are linked to their parents by using explicit lines between them. The two techniques both have advantages and disadvantages: Enclosure makes efficient use of the space available but has problems showing information stored in internal nodes, while connection uses much space as background but can present information stored in internal nodes.

3.4 Pad and Pad++: Zooming Interfaces

The *Pad* and *Pad++* user interface model [67,3] makes use of a (seemingly) infinite 2D space to allow a larger information space than that on traditional graphical user interfaces. The system allows a user to have all information placed on an work surface and to navigate between them using scrolling instead of having a fixed space on which to manipulate several windows (opening, closing, and switching between them). To allow overview of the working area, and to allow detailed viewing on information, the Pad system provides the user with the functionality to zoom in or out from the working area. An important concept introduced with the models was the notion of changing the appearance of displayed objects as users magnified it, or *Semantic Zooming*. For example, a calendar object can change from just showing the current year to showing all the months or a book object can change from just showing the title to showing the chapter index. Further, *Portals* were introduced in order to provide an overview when the user had zoomed in on a detail. These moveable frames allow a view of other areas (at other levels of magnification) within a small area of the zoomed-in view, and can be "entered" to change viewpoint. A variant of these, *Portal Filters*, modifies the view of information to provide alternative views, e.g. to show tabular data as a bar chart. The *Magic Lens* and the *Toolglass* are similar, more general versions, of portals that allow manipulation of the information viewed [7,85,8].

A later version of the user interface has been implemented in the *Jazz* toolkit [4], which was used to implement KidPad [6]. Other systems using similar zooming techniques include *Tabula Rasa* [21], the *Event Horizon* user interface model [86], especially designed for use on small displays, and the DataSplash system. The last of these have been extended with the principle of *constant information density*, ensuring that the same amount of information is shown at all levels of detail (either only in the z dimension [95], or in x, y, and z-dimensions [96]). Pook *et al.* [68] have introduced the concepts of context layer and history layer to aid navigation of zooming user interfaces.

3.5 The Hyperbolic Tree: Alternative Geometries

The Hyperbolic tree [52] introduced the use of the non-euclidean hyperbolic space to layout large hierarchies. The main benefit of placing the objects of an hierarchy in a hyperbolic space is that this space increases exponentially with increased distance, compared to linearly in normal euclidean space. Thus, the exponential growth of the hierarchies is countered by a growth of the space itself. By then projecting the space onto a 2-dimensional plane to display it, a visualization is achieved where a focus has a substantial part of the available area, yet all other information is shown (limited only by the resolution of the display). The visualization can be changed by moving the focus, changing the projecting function without changing the underlying layout. Even though the translation of viewpoints in hyperbolic space does not match intuitive movement of objects in normal space (objects which are moved rotate), this can largely be mitigated by counter-rotating the central node of the hierarchy to preserve the initial facing. As an interesting similarity, it can be noted that the *CATGraph* fisheye technique [44] mentioned earlier visually resembles the Hyperbolic tree, although it is based on an Euclidean space model.

The use of hyperbolic space has been applied to show information in three dimensions [61, 62 pp. 19-66], similar to the expansions done on many other information visualization techniques. In these cases, the layout of information within a 3-dimensional hyperbolic space is projected onto a sphere in an Euclidean 3-dimensional space before being projected onto a 2-dimensional plane.

3.6 Theories, Taxonomies and Data Models

Most work on information visualization combines theories with examples of applications or techniques. However, there exists a significant body of work that tries to find common denominators and metrics for information visualizations. This section gives a brief overview of the theoretically-oriented work than pertains to this thesis. For a more complete overview, the reader is referred to the collection *Readings in Information Visualization: Using Vision to Think* [10].

The work by Tufte on graphical presentation of data is probably the most often cited work of non-interactive presentation theory. Many of

the terms and concepts articulated in his book can be identified within information visualizations as well as within statical graphical presentations. In *The Visual Display of Quantitative Information* [87], Tufte defines a number of graphical concepts: *Data-Ink ratio*, the ratio between ink used to print the actual data and ink used for the whole presentation; *Chartjunk*, the use of decorations to make the presentation more interesting or to appear (wrongly) more scientific; *Multifunctioning Graphical Elements*, the use of a graphical element for several graphical purposes; *Data Density*, the ratio between number of pieces of information shown and the area used to show them; and *Small multiples*, the connection of several presentations which use the same combinations of variables by an additional variable.

Tufte elaborates on several of these themes in *Envisioning Information* [88], as well as introduces several new concepts. He describes several techniques to show information with more than two variables on 2-dimensional surfaces, or as he denoted it, *Escaping Flatland. Micro/ Macro readings* allow properly arranged complex data structures to have easily accessible aggregated data, and is probably the technique most obviously applicable to visualizations (c.f. Card *et al.*, where it is noted as one technique for reduction of information to create contextual presentation areas [10, p. 307]). *Layering and Separation* is used to minimize possible erroneous interpretations and enrich understanding by creating distinct layers and separating different types of information within one presentation. Further, several way of using color to add information depth to presentations are introduced, as well as how presentations with narrative are created through the use of maps and time-series.

Clearly, many of the above concepts can be applied to interactive computer programs displaying information: change ink for pixel usage to create Data-Pixel ratio, etc. However, there is relatively little explicitly stated use of this theoretical material in the information visualization field. There are at least two reasons for this. First, many of the concepts used are not easily quantified, and thus possible to reproduce in algorithms. Second, the theories are descriptive rather than constructive, making them useful to analyze information visualization that have been constructed, but do not help in constructing novel visualizations. Third, it is not clear that all concepts and rules of thumbs of static presentations are suitable for interactive presentations. As one exception, the work of Apperley *et al.* [1] bears strong visual likeness to Tufte's example of a travel itinerary [87, p. 31]. In later work [89, p. 146], Tufte

20

has started to apply his theories to computer interfaces, but has yet not addressed the specific problems of interactive visualizations.

Given these possible reasons, it is not surprising that many of the theories within information visualization are only loosely, if at all, based on the accumulated theoretical material from the older fields related to graphical presentations. The theoretical work within information visualization closest related to the older theories has primarily been concerned with generating graphical presentations from stored data, and has not emphasized interaction (thereby falling somewhat outside the definition of information visualization on page 3 above). Examples include work to design compositional algebra to encode graphical design criteria [56] and automated design of presentations based on task-analysis [14].

Leung and Apperley developed an early taxonomy of distortion-oriented techniques [54], dividing techniques into two distinct classes: those using continuous magnifying functions and those using piecewise continuous magnifying functions. They further divide piecewise continuous functions into functions that have constant or varying magnification. The work also presented a unified theory of distorting presentations based on the *Rubber Sheet* metaphor introduced by Sarkar [78], and distinguished between a transformation function of an image, and its derivative, the magnification function.

Using four orthogonal visual transformation [84], Spence proposed a taxonomy to graphical presentations. Using these transformations, he describes the different between the seemingly similar motivation of the BiFocal display [83] and Fisheye views [23].

Furnas and Bederson introduced *Space-Scale Diagrams* [26] to provide an analytic framework for multiscale interfaces. Although the main focus lies on describing zooming interfaces (including issues such as fractal grids), and using the diagrams to solve navigational issues regarding such techniques, they were shown to adequately illustrate fisheye views and image warpings. In his later work, *Effective View Navigation* [24], Furnas introduces ways of defining requirements for *efficient view transversal* and *view navigability* in various data structures, as well as giving examples of how to improve these for often used types of information such as lists.

Building on a description of information visualizations [49] where distortions were identified as non-linear magnification transformations, Keahey and Robertson introduced *Nonlinear Magnification Fields* [50]. These abstract representations allow the creation of distorted presenta-

tions with magnified areas without having to define explicit focal areas. Further, the fields allow *data-driven* magnifications, where properties of the information to be shown can automatically adjust the magnification field. The concept of Nonlinear Magnification Fields have been used to address the *Generalized Detail-In-Context* problem [46], or how to effectively use the space gained by distorting presentation to enhance the visualization of the area of interest. The concept has further been expanded for use in 3-dimensional representations [47] and used to create *Area-Normalized Thematic Views* [48], maps that maintain their overall structure while showing regions in proportion to their encoded information. A detailed description of the earlier work on nonlinear magnification can be found in [45], including the identification of the levels of application: *image-level*, where the magnification operates on an pre-determined image of some information; *render-level*, where the position of objects are affected by the magnification but not the presentation of the objects; and *data-level*, where the objects representations are affected by the magnification.

Describing visualizations as *Interactive Externalizations* [90], Tweedie defines three aspects for categorizing information visualizations: the data and data structure of the information used (which may be used "raw" or be refined into constructed values and structures), methods of interaction (direct or indirect manipulation), and the visual feedback explicitly given by input and output.

Categorizing visualizations due to how they structure a *layout strategy*, how they provide *navigation and interaction*, and how they make use of *clustering*, Herman *et al.* [33] offer a extensive survey of visualization techniques for graphs. Although extending beyond the scope of this thesis, it does give numerous examples of information visualization techniques.

From the above examples, it is apparent that there does not exist a common agreement within the information visualization research field about how to categorize the various information visualizations techniques into a taxonomy. As the research field is a widely diverse field, and still developing, this can be seen as a still potent research issue, rather than the effect of disagreement.
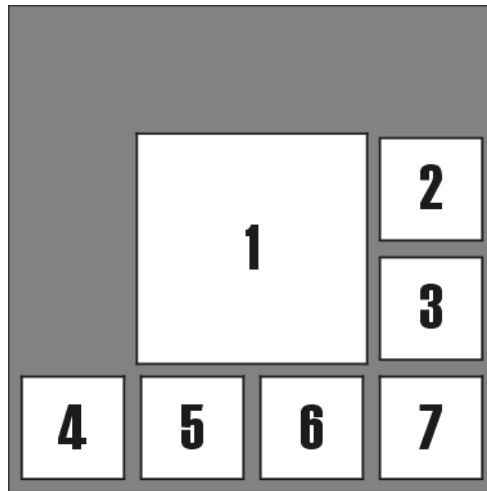
4 Flip Zooming

The information visualization technique Flip Zooming has been implemented in two prototypes by Holmquist before the work presented in the thesis (c.f. [9, 35]). The following section describes the refined technique that was developed based on the experiences of these prototypes. The main difference between the original technique and the refined one is that the latter makes a trade-off between being space-filling and minimizing the movement of information between different views.

Flip Zooming, the subject of this thesis, is an information visualization technique that visualizes discrete sets of ordered information. Each piece of information is presented as an independent *tile* on a common background. The tiles are arranged so that the order of the information is presented in a left-to-right, top-to-bottom fashion, i.e. using the same layout strategy as in writing in scripts of european languages[1]. One of the tiles is attributed the *focus tile*, and is centered in the available display area with the other tiles distributed so that the above- mentioned ordering is maintaining. The other tiles are designated *context tiles*. The focus tile is given more space so that a more detailed view of the information presented on the tile is possible (see **Figure 6**).

---

1. The implications of using other spacial arrangements, suited to the socio-cultural background of the user, have not been addressed in the work presented in this thesis.
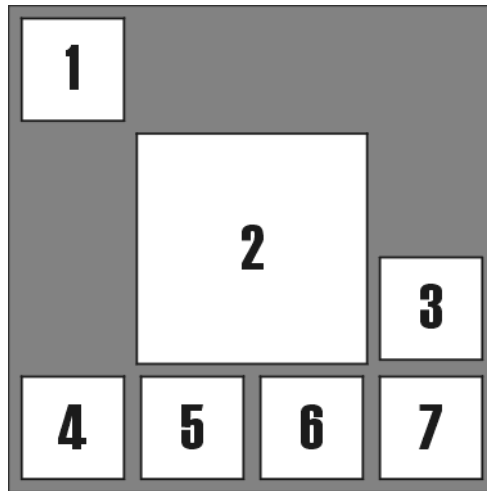
Fig. 6. Schematic figure of Flip Zooming. The presentation consist of seven tiles where the first tile is the focus tile. (Figures 6 to 8 are taken from paper 2).

The Flip Zooming technique allows users to change which tile is the focus tile by using two methods: navigating sequentially or using random access. Sequential movement provides the movement of the focus to the tile prior or posterior to the current focus tile. As this navigation only requires two operators, it is usually facilitated by binding the operations to physical buttons on the user's interface. Random access allows the user to select any tile as the new focus tile, and is usually facilitated by the use of a pointing device such as a mouse, which allows free movement over the display area.

When the focus tile is changed, all tiles between the old focus tile and the new one are moved to preserve the ordering of the sequence. Thus, all tiles have three positions: one position if before the focus tile, another position when it is the focus tile (where is it given more space), and a third position when it is after the focus tile. **Figure 7** shows the same information set as in **Figure 6** after the focus tile has been changed to the second tile, while **Figure 8** shows the information when the sixth tile is the focus tile.

Fig. 7. Schematic figure of Flip Zooming. The focus tile is the second tile and, thus, placed in the centre of the available space. The first tile has been moved, compared to its place in Figure 6, to a place above and to the left of the focus.

Even though Flip Zooming presents discrete sets of information, many forms of information can easily be transformed to be presented with the technique. The initial use of the technique [34,35] transformed the contents of a web page into several "pages", each one presented as a separate tile in the visualizations. This form of transformation allows information that is sequential in one dimension (unlike e.g. tabular information or images which depend on ordering of information in two dimensions) to be presented by the technique.
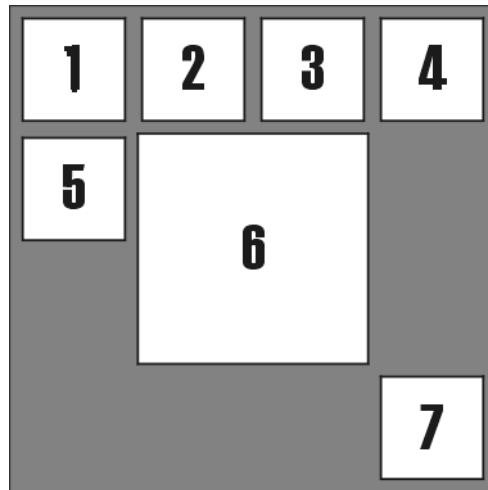
Fig. 8. Schematic figure of Flip Zooming. The sixth tile is the focus tile with all tiles that are before it in the ordering placed above and/or to the left of it.

A more extensive description of the Flip Zooming technique can be found in paper 2 of the thesis. This description also describes how Flip Zooming visualizations can be transformed into hierarchical visualizations.

4.1 Relating Flip Zooming to Previous Work

Flip Zooming belongs to the group of information visualizations techniques called *Focus+Context* techniques. The aim of these techniques is to allow the user a detailed view of information of interest while at the same time provide a complete overview of the information presented. The techniques provide the user with one or more areas on the display where the information presented is detailed and given extra space, the *Focus/Foci*, while the rest of the display is designated the *Context*, in which the remaining information is presented in a compressed, distorted or in some other way manipulated form so that it fits that area. Even though much of the earliest work within information visualization belongs to Focus+Context techniques, the first use of the term

Focus+Context in the literature of the field seems to be within the description of the Document Lens [74].

Many different ways of defining the focus/foci area(s) have been used in information visualization techniques. If the information is presented as being placed on a surface that is deformed, the focus area is defined as a specific part of that surface, which may or may not coincide with boundaries of individual pieces of information (examples of such techniques include the Perspective Wall [57] and the Document Lens [74]). Other techniques allow the user to select individual pieces of information as focal points, changing the appearance of the pieces individually (c.f. BiFocal Display [83], Table Lens [71], Rubber Sheets [78]). Some techniques adapt to the type of information presented, using objects as focal points when presenting graphs, while using arbitrary points on the display area as focal points when displaying continuous information such as maps (c.f. Graphical Fisheye views [76]). Techniques using Nonlinear magnification fields do not define explicit focal regions but generate them from having local plateaus in the fields (c.f. [46]). Flip Zooming visualizes collections of distinct pieces of information, and as such, it uses these pieces to define the focus of the presentation. As the information is shown in a sequential order, the focus can be *expressed as the index of an object in the sequence*.

Many Focus+Context techniques do not define explicit levels for the different sizes and appearances objects in the visualization can have, but instead let a layout strategy decide these factors based on the overall presentation. One notable exception is the BiFocal Lens [83], where objects are presented in one of two forms, one used when the objects are within the focus area, and one used when they are in the context area. Flip Zooming uses the same technique, but only one object at a time uses the focused view, as the focus is defined as being one object. However, the ratio between the focus object and the objects constituting the context can be changed interactively.

Focus+Context techniques have used several different approaches to reduce the information density in non-focal areas to make it feasible to show all information in the display area. Card *et al.* [10, pp. 307-308] identifies five techniques: *Filtering*, *Selective Aggregation*, *Micro-Macro Readings*, *Highlighting*, and *Distortion*. Filtering removes elements that are deemed least important to allow the remaining information to be shown (c.f. FishEye Views [22]). Selective Aggregation, called Semantic Zooming when applied to Zooming Interfaces (c.f.

[3,67,95]), presents aggregated versions of information when space is limited, but display detailed components when there is sufficient space (c.f. [79] and the structured layouts in [78]). Micro-Macro Readings allow the structure presentation of individual components to reveal information that is on a structural level, and is one of the reasons why Focus+Context techniques try to present all information within the display area. Highlighting is similar to that of Micro-Macro Readings but simultaneously identifies individual pieces of information while relating them to the structural information in the presentation. Distortion is the technique where the amount of space required for various objects in the presentation is reduced by changes in size, viewing perspective, or the space that the object is located in. Most of the visualizations within Focus+Context techniques use distortion for creating contexts (c.f. [50,73,78,94]). Flip Zooming uses a linear transformation to reduce the size required by elements, which maintains the proportions of elements as well as leaves the internal presentation unmodified (except for size). Thus, Flip Zooming belongs to the category of distorting techniques, even if the label distorting is somewhat misleading.

The various Focus+Context techniques have used a wide variety of approaches to layout the information presented. Many make use of positions that are inherent in the data to be presented, e.g. vectorized maps, to create a Focus+Context presentation by calculating new positions based on the original positions using a translocation function (c.f. Rubber Sheets [78]), or by creating a presentation on a surface which is then manipulated (c.f. Nonlinear Magnification Fields [46]). Information that does not have inherent positions but have an ordinal structure usually has a layout algorithm where one or two dimensions are used to preserve the ordering (c.f. the Document Lens [74] and the Perspective Wall [57]). In cases where the elements to be visualized have strong but relative relationships, such as in hierarchies, visualization techniques use layout algorithms that maintain these relationships and make efficient use of the *space* in which the elements are placed in (c.f. the Hyperbolic Browser [52, 94] for layout in a hyperbolic space, and, although not Focus+Context techniques, Tree maps [41] and Cone Trees [73] for layouts in two and three dimensions respectively). Flip Zooming visualizes sequential and distinct sets of information on a two-dimensional surface maintaining a left-to-right, top-to-bottom ordering of the elements. Unlike many techniques, the elements are moved as the user interacts with the visualization, but in a way that maintains the

ordering. In addition, the elements only move between three different positions. Sequential but continuous information, e.g. longer texts, can be visualized with Flip Zooming by simply dividing the information in appropriately-sized groups (as when a text is divided into pages).

Flip Zooming visualizations can be used recursively to create an hierarchical structure of Flip Zooming visualizations, something not seen in other information visualizations. Besides allowing the Flip Zooming technique to be used for visualizing hierarchical data structures (where the information is stored in the leaves), this allows for a visualization where the nodes are independently manipulable visualizations with separate foci and contexts.

Although Flip Zooming contains the word zooming is it not a *Zooming Visualization*. The techniques within this category, c.f. Pad and Pad++ [67,3], provide smooth zooming and panning operations and do not provide a global context to the information zoomed upon. As Flip Zooming maintains the same proportions between the focused presentation and the context presentation of an object, the focused object could be described as a zoomed-in view. However, the technique does not provide intermediate views and, unlike a zooming visualization, provides a global context.

4.2 The Thesis: Flip Zooming - the Development of a Visualization Technique

As mentioned in the previous section, Flip zooming had been implemented in two prototypes before the work presented in the thesis. The papers of this thesis constitute all subsequent work to date. Taking a basis from the two earlier prototypes that presented single types of information, web pages and images respectively, the technique was generalized to be usable for many types of information. While doing so, a number of other prototypes were developed to study the aims of (1) decreasing the time t needed to interpret the data D and (2), of lowering the requirements of users pre-knowledge S, mentioned in the beginning of the section. However, as more complex prototypes were developed, to be used in more complex environments, the focus changed towards (3) specializing the visualization for a particular data type D. In doing so, the potential of using Flip Zooming in various situations became clearer even though the technique itself became more difficult to

describe. Instead of being a precise and structured technique, it evolved into a system of guidelines and rules of thumb.

The thesis consists of six papers: all of which have been published in 1999 and 2000. The papers are presented without any changes except for the required reformatting to fit the format of the thesis. All implementation of the visualization parts of the applications have been done by the author, while some other parts of the implementations, such as parsing (in paper 3) and database management (in paper 4) have been done by others. The papers are as follows:

1. Björk, S., and Holmquist, L.E.: *Exploring the Literary Web: The Digital Variants Browser.* In s*eminar book on Literature, philology and computers '98*, Edinburgh, UK.

2. Björk, S. *Hierarchical Flip Zooming: Enabling Parallel Explorations of Hierarchical Visualizations.* In *Proceedings of Conference on Advanced Visual Interfaces (AVI 2000)*, pp. 232-237, ACM Press, 2000.

3. Björk, S., Holmquist, L.E., Redström, J., Bretan. I., Danielsson, R., Karlgren, J., and Franzén, K.: *WEST: A Web Browser for Small Terminals.* In *Proceedings of ACM CHI Conference on User Interface Software and Technology (UIST '99), CHI Letters Vol. 1, Issue 1*, pp. 187-196, ACM Press, 1999.

4. Björk, S., Redström, J., Ljungstrand, P., and Holmquist, L.E.: *POWERVIEW: Using information links and information views to navigate and visualize information on small displays.* In *Proceedings of Handheld and Ubiquitous Computing 2000 (HUC2k)*, Springer-Verlag, 2000.

5. Björk, S. Holmquist, L.E and Redström J: *A Framework for Focus+Context Visualization.* Abridged version in *Proceedings of IEEE Symposium on Information Visualization (InfoVis '99)*, pp. 53-56, IEEE Press, 1999. Full version in *CD-ROM Proceedings of IEEE Visualization 1999*, IEEE Press, 1999.

6. Björk, S., and Redström, J.: Redefining the Focus and Context of Focus+Context Visualizations. Abridged version in *Proceedings of IEEE Symposium on Information Visualization (InfoVis 2000)*, pp. 85-90, IEEE Press, 2000. Full version in *CD-ROM Proceedings of IEEE Visualization 2000*, IEEE Press, 2000.

5   Research method

Information visualization is a relatively young multi-disciplinary research field with a focus on developing and refining visualization techniques. The dominating way of introducing a new technique is by example, putting a strong emphasis on the implementation and practical construction of prototype applications that use the technique. Both the age and the applied nature of the field have had the consequence that there does not exist an extensive body of work regarding research methods within information visualization. One of the few places where a research method for the field is described is in the paper on the Information Visualizer [11], where Card *et al.* briefly describe the development of information visualizations in the light of what the call the *systems research paradigm*. They summarize the paradigm as consisting of four components:

Exploratory Design. Working from inspiration or synthetic approaches of existing applications, new designs are constructed to demonstrate the feasibility of an idea for a visualization technique. Working iteratively, the design can be refined using techniques such as initial participatory design [60] or heuristic evaluation [63]. Papers 1, 3, and 4 of the thesis are examples of exploratory designs where the feasibility of variations of the Flip Zooming technique has been studied.

Abstraction. After constructing prototype systems and applications, a design space can be spanned by using the designs as points of reference. This identifies the essence of different techniques and allows them to be applied in other instantiations and for other areas of use. Paper 2 in the thesis is an example of how the Digital Variants Browser and the Hierarchical Image Browser (first described in [37]) applications can be abstracted to describe the underlying hierarchical visualization technique.

Theories and Empirical Observations. These are used to further refine the description of the design space. This serves several purposes: it identifies research areas that have not yet been addressed; it aids in contrasting different visualization techniques using some metrics; it

helps define what research is to be regarded as a part of the research field. Paper 5 of the thesis shows how information visualizations can be described, as well as how new visualizations can be created by combining several different visualizations. Paper 6 identifies some presumptions about concepts used within Focus+Context techniques, and shows how alternative techniques, still being within the category of Focus+Context techniques, can be constructed if these presumptions are abandoned.

Codification. The body of knowledge built from the three preceding components need to be codified in order to be transmitted to other people who need to build similar systems. This is primarily done by publishing papers at scientific conferences, but may also include technical reports and the creation of web sites. All the papers in the thesis have been published at international conferences, thereby applying to the rule of codification. Paper 1 is noteworthy in this context since it not only was presented at scientific conferences, but also since a significant proportion of the attendees were part of the target user group for the presented system.

Although the above paradigm describes the work in this thesis, the description does not relate the paradigm to other methods of research within the scientific community. To do this, we must take a step back from information visualization and look at its "parent" research field, Human-Computer Interaction.

5.1   Methods Within Human-Computer Interaction

Information visualization is regarded as one component in the large and diverse research field of *Human-Computer Interaction (HCI)*. Being a relatively new multi-disciplinary field that studies the interaction between humans and man-made artifacts, many of the theories and methods used come from "parent" disciplines (e.g. computer science, psychology, design, engineering etc.). As HCI involves elements from both science and design, it cannot be said to solely be a natural science or a design discipline. As Mackay and Fayard state in [55], "HCI cannot be considered a pure natural science because it studies the interaction between people and artificially-created artifacts, rather than naturally-occurring phenomena, which violates several basic assumptions of natu-

32

ral science. Similarly, HCI cannot be considered a pure design discipline because it strives to independently verify design decisions and processes, and borrows many values from scientists." Greenberg and Thimbleby make the same point, identifying HCI as being both an engineering discipline (which Mackay and Fayard have grouped together with the design disciplines) and a scientific discipline [27].

Looking at the problem of practising science and design together, Mackay and Fayard [55] have proposed a framework for describing the scientific method used within HCI as well as describing the aims of specific research within the field using a triangulation method. The proposed framework is based on a synthesis of two of the most frequently used models of scientific research: the *deductive* and the *inductive* (see **Figures 9** and **10**).

Described briefly, the deductive approach starts with a theory. A hypothesis is formed from this theory by looking at a specific case predicted by the theory when applied to a specific set of criteria (in other words, deducing degrees of freedom in the theory by instantiation leads to an hypothesis). An experiment is thereafter designed and conducted to verify (or disprove) the hypothesis. The experiment takes place in an environment where as many variables as possible that affect the experiment can be controlled. The results from the experiment are then used to accept, revise or falsify the hypothesis, which leads to new experiments. As Mackay and Fayard state, a scientist using this model "values *reliability*, which means that the same results will be obtained if the experiment is repeated under the same conditions, and *validity*, which means that the results can be generalized beyond the specific experimental setting in the laboratory" [55, p. 226].
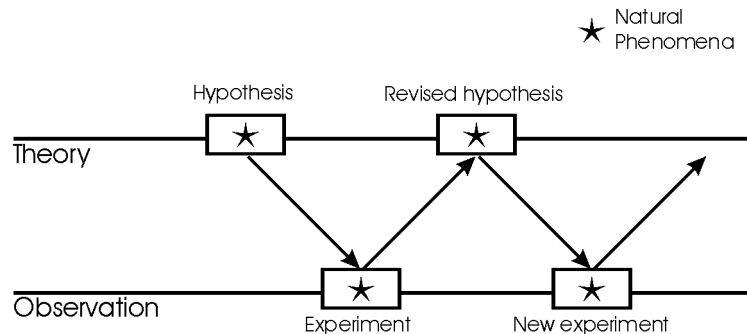
**Fig. 9.** The deductive model of scientific research (adapted from [55]).

In contrast, the inductive model starts by observing a phenomenon in the world. By trying to describe the phenomenon as accurately as possible, a framework is developed. The researcher then resumes observing the phenomenon (or related phenomena) and modifies or replaces the framework as needed.
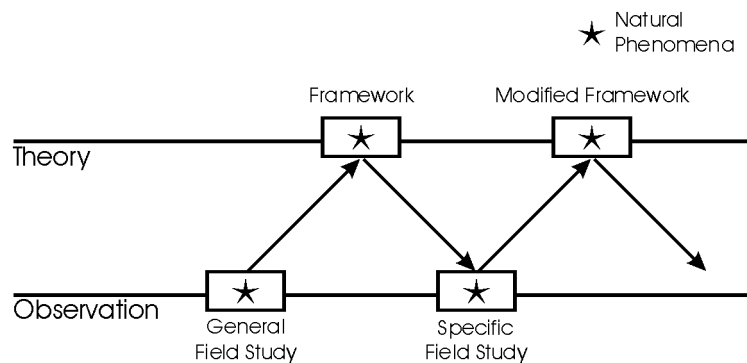
Fig. 10. The inductive model of scientific research (adapted from [55]).

Mackay and Fayard identify two obstacles to seeing HCI as a (natural) science. First, it involves the design of new artifacts, thus the researcher is not just observing the world but also changing it. Second, HCI does

34

not study human behavior in an artificial environment or artifacts as isolated objects, but the *interaction* between humans and artifacts. Based on these objections, Mackay and Fayard create a synthesized model where the oscillation between theory and observation (which occurs in both models) is interjected by the design of artifacts (see **Figure 11**). Note that the model does not prescribe that the design of artifacts always takes place between theory and observation phases (c.f. Mackay and Fayard's use of the model to describe their own work [55]).
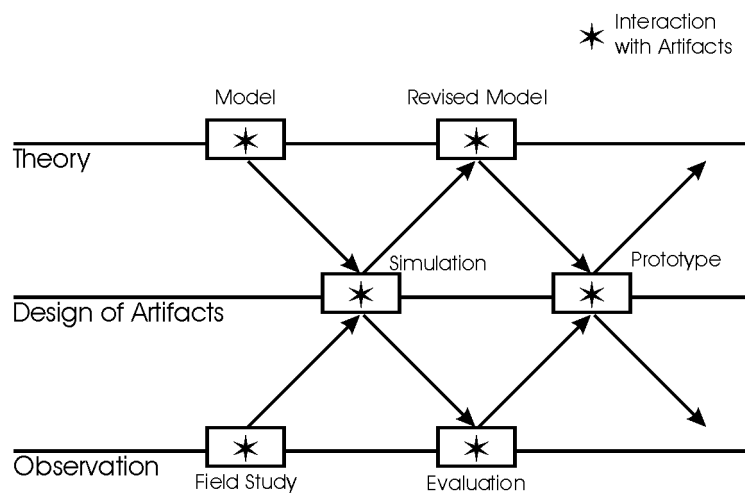
Fig. 11. Mackay and Fayard's model of HCI research. Note that as HCI is multidisciplinary, the box labels should not be seen as the only techniques possible, rather they should be seen as illustrative examples (adapted from [55]).

It should be noted that the experiments and field studies in both the deductive and inductive models make use of one or more techniques developed within the specific research field studied. As a multidisciplinary field, HCI uses techniques from various research fields as appropriate due to external conditions. Thus, observations performed within HCI can use a vast range of techniques, from the controlled experiment of psychology to "quick and dirty" ethnographical studies, from the use of cognitive models to heuristic evaluations.

In the book "The Invisible Computer" [66], Norman describes *Human-Centered Development*, which describes a process for the development of products. The process requires *field studies*, *rapid prototyping*, *user testing*, *graphical design*, and *technical writing*. Although the process is described for commercial product development, and for projects involving large groups of people, parts of the process can be used to illustrate some of the different techniques used in HCI research.

Field studies within HCI (c.f. *mobile informatics* [16]) are based on observing how practitioners perform their daily work, especially noting how they use various artifacts to organize and coordinate tasks. Based on these observations, proposals of computer systems to enhance and ease the work are created. Field studies are motivated by the possibility of gaining important insights into how actual work is conducted, something which can be overlooked when doing formal analysis only. Even though these insights most likely refer to what information is needed (and where), they may also indicate how the information should be displayed.

Rapid prototyping linked with user testing allows for the design of artifacts to test and evaluate ideas and hunches. By iterating design rapidly, systems can be incrementally improved and go through a series of initial testing. When the test users are people working within the project, the evaluation process becomes one of *heuristic evaluation* [63].

Graphical design is important to the design of both computer devices and computer interfaces since unaesthetic appearances may not only lessen the use of the system but may also make the system less comprehensible. The work of Tufte [87,88,89] on how to envision information can be seen as guidelines in graphical design for information visualization.

Technical writing is according to Norman the key to the entire operation. By writing a simple and elegant manual *before* the actual design, the product will be designed from a user perspective rather than a technological perspective. When such manuals are completely successful, the design becomes so intuitive that the user needs no instruction manual. The procedure of writing technical manuals can even be seen as the creation of frameworks or theoretical models of systems.

Because it is developed for the commercial development of products, Norman's model assumes that all tasks will, out of necessity, be restricted to a tight time schedule. When applying the model to HCI research, these limitations are not so severe, providing, for instance, the

possibility of performing the traditional laboratory experiments of experimental psychology rather than rapid user studies, or iteratively developing a technique by testing different variants and putting it to use in different environments.

A Note on Validation. As is the case in any scientific research, information visualization requires some form of validation to substantiate findings and make claims about specific techniques. One way of determining how successful a visualization technique is in visualizing information is to compare it to other techniques on a specific set of tasks and data. By performing quantitative experiments, the technique can thus gain validation by statistical analysis of collected data. This manner of testing is often used within the field of human-computer interaction. However, work of this kind is seldom reported within the information visualization field.

There may be several reasons for this. First, many different visualizations are constructed to show different types of information (e.g. hierarchies, graphs, maps), making the comparison between them impossible or fabricated, and making the outcome of the visualization evident before the systems are compared. Second, if two visualizations show the same type of information, they may be specialized towards different tasks, making the choice of a fair experimental task difficult. Third, and unfortunately, many of the techniques are used by commercial companies, which makes it difficult or expensive to get hold of the original source code for the visualization (to ensure that the actual technique and not a replica is used). Even when the techniques are publicly available, it may not be possible to customize them to specific experiments, which might mean that the outcome could be more dependent on the design of the experiments than on the inherent differences of the techniques[1]. Given these difficulties in objectively comparing different information visualization techniques to one another, it is not surprising that many evaluations compare two different variations of the same information visualization (c.f. [5,75]), where the variation typically lies in disabling one specific feature of interest[2].

---

1. This problem can, at least partially, be mitigated by having the experiment performed by researchers that have not participated in the creation of any of the techniques.

In this situation, therefore, it is necessary to use additional methods to validate the findings in information visualization techniques. By letting people use computer applications with specific use domain in which the technique has been implemented, empirical data can be collected to validate the usability of the technique. By comparing the visualization techniques against the most used graphical user interface (which is seen as a baseline), one can show that proposed techniques are advantageous to the systems currently used. By performing formative and qualitative evaluations, one can refine the technique stepwise to design new, improved, visualizations. By testing the technique in a variety of situations, one can find the strengths and weaknesses of the visualization and identify suitable areas of use. By making commercial products using the technique, one can set the technique to a form of evolutionary testing against other commercialized techniques. What is important is that the technique, through applications implementing it, is put to use. In the process, one maintains a design-oriented study from a user perspective that is judged on the success (value) of the applications.

The Digital Variants browser has been presented, demonstrated, and positively received, at a scientific seminar for the intended users of the system (paper 1). A qualitative evaluation has been performed on the WEST browser using ten test subjects (paper 3). The PowerView prototype (paper 4) has been quantitatively and qualitatively evaluated in experiments conducted by people not directly involved with the implementation of the application (detailed description can be found in [30 and 31]). The developed framework of Flip Zooming has been used to present slides at conference presentations (c.f. [37]), and has been used to construct information kiosk presentations. Looking at the empirical material presented in conferences on information visualization techniques, Flip Zooming has been comprehensively evaluated in various use situations employing various evaluation techniques e.g. formative, qualitative, and heuristic, making its validation on par with other Focus+Context techniques developed.

Even given the large amount of empirical data collected about Flip Zooming, it is hard to say that it is objectively better or worse than other information visualizations. Few other visualization techniques address

---

2. For a similar enumeration of problems in reconstructing interfaces, in the context of reproducing experiments without explicitly stated underlying theories, see Greenberg and Thimbleby [27].

the same particular information and use, which makes objective comparisons difficult. Even though Flip Zooming has proven to be feasible by several experiments, and is appealing in its simplicity, arguments in favour of it should be seen as normative rather than objective. Examples of such arguments can be seen in the theoretically-oriented papers of the thesis (2, 5, and 6), which propose new techniques, frameworks, and interpretations of concepts based on other values than those provided by instrumental observation.

5.2   Using Mackay and Fayard's Model

In order to provide an overview of the development of Flip Zooming, Mackay and Fayard's model for research in HCI will in this section be applied to the work presented in the thesis. There are several reasons for using this model. It

- provides a brief description of the work performed.
- offers a clear division of the development into a number of distinct groups. Each group is based around one of the papers in the thesis.
- gives an analysis for the causes behind the course of action taken within each division.
- provides a description of the research techniques used in each group during the development of Flip Zooming.

The reader is referred to the individual papers for more detailed descriptions of the applications, including walk-throughs of the interfaces.

Origins of Flip Zooming. As stated earlier, the Flip Zooming technique had already been invented and refined before the work described in the thesis was initiated. Based on a model of how to visualize web pages, Holmquist developed the Zoom Browser [34]. Generalizing the techniques and applying it to a collection of images resulted in the Flip Zooming Image Browser, which was used in a formative evaluation of the Flip Zooming technique [9]. The evaluation showed that users perceived the technique as providing a good overview but that it had an unclear structure. This lead to a modification of the technique where the efficiency of screen usage was reduced to allow the use of a central area for the tile in focus, as well as leading to the implementation of a general framework. This framework was then used to create a new, hierarchical, image browser [37]. **Figure 12** shows the early development of Flip Zooming.
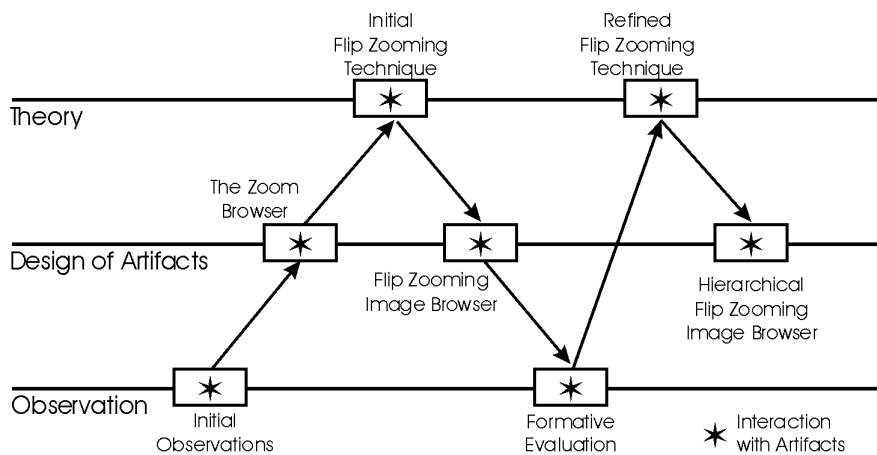

**Fig. 12.** Early development of the Flip Zooming technique.

The Digital Variants Browser. Applying the refined Flip Zooming technique for presenting different variants of the same text resulted in the construction of the Digital Variants Browser (paper 1). Earlier applications using Flip Zooming had the possibility of showing several documents at once (by appending the tiles showing one document to the end of a sequence showing another document). This was sufficient when different documents were to be shown but not adequate when variants of one text were to be shown. However, the implicit possibility of using hierarchies in the refined Flip Zooming technique allowed the separation of the tile representing each document into a number of independent Flip Zooming visualizations. Each of these visualizations could then be placed in an outer Flip Zooming visualization, so that a user could not only chose which "page" to view in a document, but also which document was of primary interest (see **Figure 13**).

As one of the most common activities when studying text variants is to compare two variants with each other, Flip Zooming was modified to have two foci, which were placed side-by-side at the top of the display area. This allowed a user to switch between viewing two different documents without having to change the focus of the visualization or having to cope with other documents between the two relevant ones. To further ease the task of comparing two documents, the *two inner foci* of the focused documents were placed at the top of the inner visualizations so that no tiles would lie between the two inner foci.
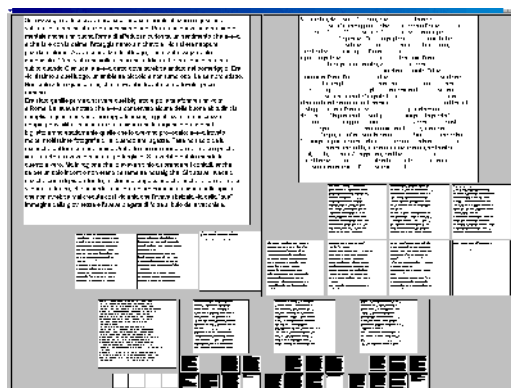

Fig. 13. The Digital Variants Browser.

41

The Digital Variants Browser received an informal evaluation by people within the user target group of the application. **Figure 14** shows the development of the Digital Variants Browser.
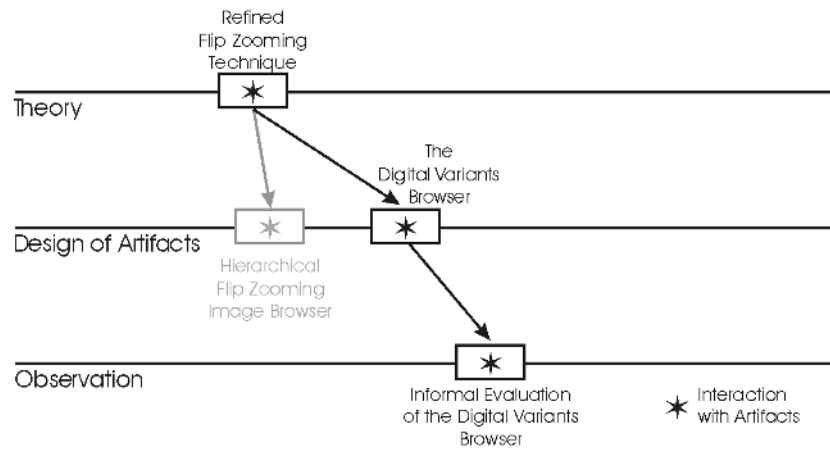
Fig. 14. Development of the Digital Variants Browser.

Hierarchical Flip Zooming. Both the Hierarchical Flip Zooming Image Browser and the Digital Variants Browser used the ability of the revised model to present generic objects to create recursive Flip Zooming visualizations. Based on the experience of using these systems, and an informal evaluation, a model for Hierarchical Flip Zooming (paper 2) was developed. The model identified concepts such as using visualizations recursively, allowing different nodes in visualization to each have a focus, identifying what operators were required to navigate the hierarchy, and showing by example that different layout strategies are appropriate due to different use situations. **Figure 15** shows the development of the Hierarchical Flip Zooming technique.
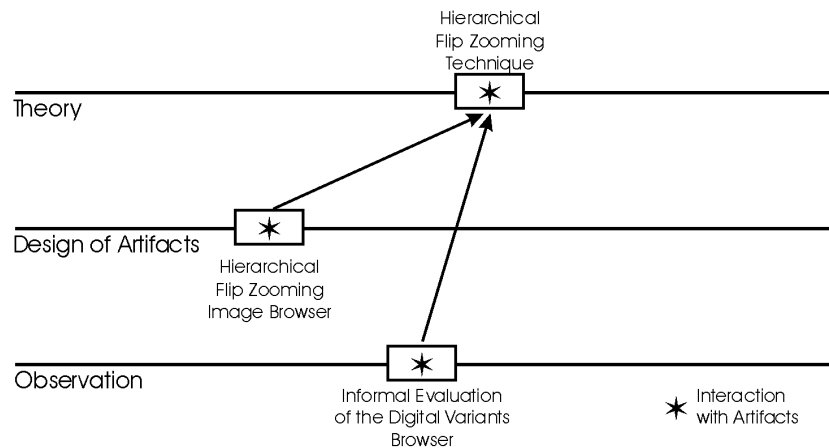
**Fig. 15.** Development of Hierarchical Flip Zooming.

Fig. 16. The WEST Browser.

The WEST Browser. The work with the WEb browser for Small Terminals, the WEST browser (paper 3), was motivated by the wish to explore how the Flip Zooming technique could be applied to small screens. Similiar to the design goal of the Zoom Browser [34], the WEST browser display web pages but is designed to work on displays with a very low resolution (160*160 pixels). In order to create a discrete set of information from a web page, the information is first divided into a number of *cards* using chunking techniques, each small enough be to viewed clearly if assigned the whole display area.

Given that the number of cards could become large if the web page contained much information, and the use of extensive scrolling-like navigation was unacceptable, the cards were stacked into decks, and these decks into decks of decks, creating an hierarchical structure. The structure was displayed using the Hierarchical Flip Zooming technique, but as the screen resolution did not allow enough space, only one deck, and its cards (or sub-decks) were shown at a time (see **Figure 16**). To be able to read the contents of a card, the user could make the visualization use the whole display area to display a single card.

An analysis of user tasks identified the use of three different presentation styles of a card; *Thumbnail view*, a scaled-down version of the full-screen presentation card, used for navigating information that was familiar from "normal" web browsing; *Keyword view*, a view where text reduction techniques summarized a card in three words, used to find points of interest; and *Link view*, where only the links contained in a card were displayed, used to provide quick navigation to other web pages. Thus, depending on what task was to be performed, the user could switch between the different views to the view that was most efficient at that given moment.

44

The WEST browser showed how Hierarchical Flip Zooming could be used together with text reduction and chunking techniques to support the browsing of web pages of PDAs. Further, it used a hierarchical structure, but due to the limitations of the display area only showed one node and its children at a time. The process of designing the browser, together with feedback from the qualitative evaluation, lead to the development of a model for Flip Zooming on small displays (see **Figure 17**).
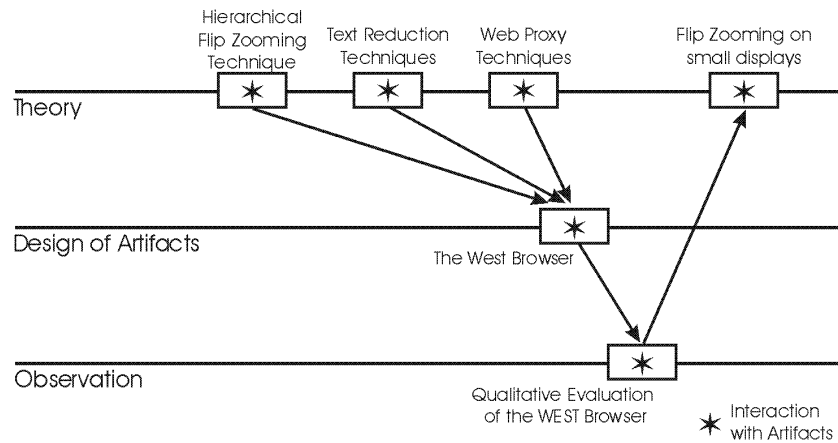
Fig. 17. Development of the WEST browser.

**Fig. 18.** PowerView. The focus is on the To Do List.

PowerView. Further exploring how Flip Zooming could be used on small displays caused the identification of a problem of interaction overhead with PDAs. The applications in these devices usually require the full use of the display area, both to show as much information as possible and to ease scrolling navigation by allowing longer "scrolling steps" to be taken. This, however, causes an interaction overhead when the user has to switch between two or more applications to be able to perform a task. Based on the model for using Flip Zooming on small displays, which already offered an alternative to scrolling, a model for integrating both the visualization and the interaction of several different applications was formed.

The model made use of *information links* to connect pieces from all different types of information into a number of contexts that were defined by one focal piece of information. For example, an entry in the contact list could have a context consisting of all meetings with that person, all e-mail communication with the person, and a list of mutually agreed upon tasks. The model also stressed the need for consistency between how interaction was enabled for different information types. Using this model, the PowerView application (paper 4) was developed, which integrated the functionality of a calendar, a to-do list, an e-mail reader and a contact list into one system (see **Figure 18**).

Using the navigation operators identified in the Hierarchical Flip Zooming model, all information in the system could be accessed using a pointing device as well as using only two buttons on a PDA, a normal button and a jog-dial (which provided three different input actions). This

allowed the application to be used single-handedly for information-retrieval tasks, unlike most PDA applications that require one hand for holding the device and one for manipulating the pointing device.

The PowerView application was evaluated in a laboratory setting, measuring the ease-of-use of Flip Zooming as well as the functionality of one-handed navigation provided by the system ([31,30]). Although the evaluation showed that the users' found that the technique provided them with an effective overview, the users did not use the possibility of single-handed navigation even though some of the tests in the experiment strongly favored this style of navigation. For a more detailed description between this difference between use and usability that the experiment identified, see [31]. **Figure 19** shows the development of the PowerView.
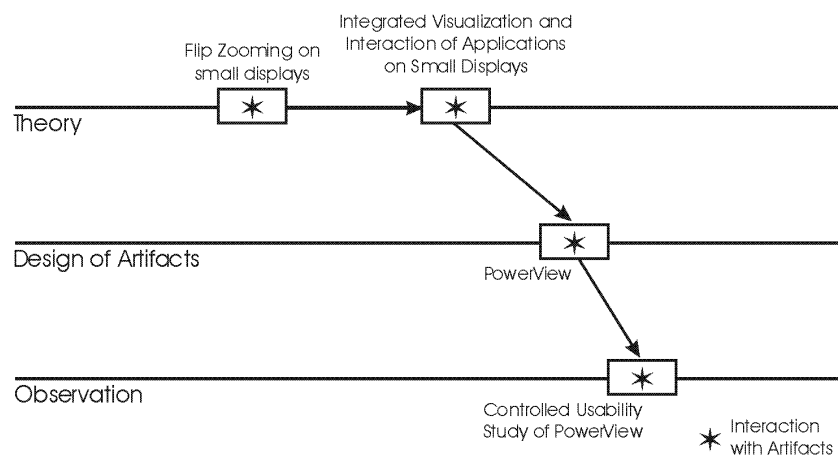

Fig. 19. Development of PowerView.

A Framework for Focus+Context Visualizations. Based on models for various forms of Flip Zooming techniques, and on observations of information visualization techniques developed in the research community, a general framework (paper 5) for Focus+Context visualizations was developed (see **Figure 20**).

The framework made use of the notion of *higher-order visualizations* that visualizes the presentation generated by other visualizations. The basic information visualization **IV**, according to the framework, is a first-order visualization, which is divided into three components: **[D]**, the set of data to be presented; **V**, the visualization function that is applied on the data; and **I**, the interaction provided by the visualization, which may include both modifying parameters in the visualization function and modifying the data itself. Higher-order visualizations, denoted **IV'**, **IV''**, etc., use the same division of three components but in this case the set to be presented is *a set of information visualizations* **[IV]** rather that a set of data.

Besides allowing a descriptive model of Focus+Context techniques, the framework illustrated how different techniques can be synthesized into novel visualizations techniques.
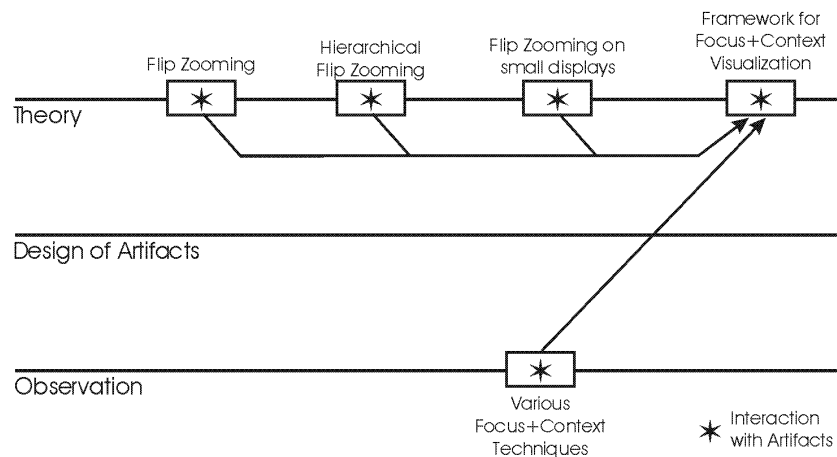
Fig. 20. Development of the framework for Focus+Context visualizations.
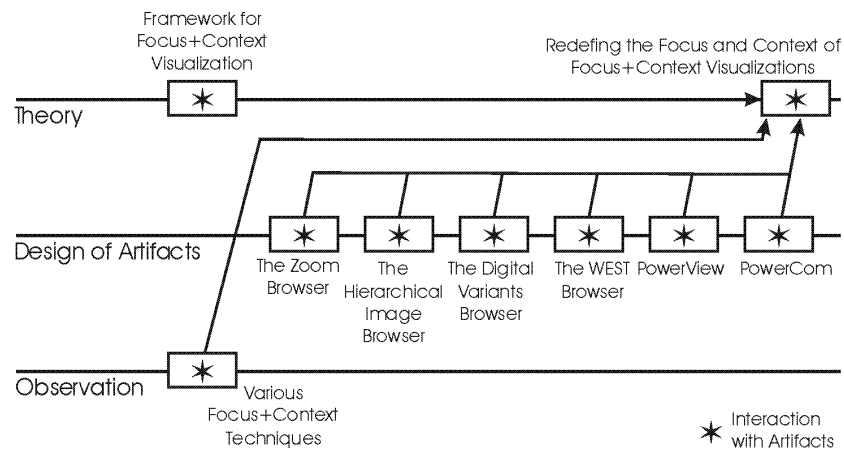
Fig. 21. Contributing observations and theories for the identification of preconceptions within Focus+Context techniques.

Redefining the Focus+Context of Focus+Context Visualizations. The work on Flip Zooming applications had required changes from the original technique each time it was applied for new areas of usage. These changes forced the reconsideration of common presumptions of Focus+Context techniques (e.g. always providing a focus, always providing a continuous context). Although the work of many researchers showed that these notions were possible to transgress, this was either done implicitly or done in the context of a specific application or use area. Paper 6 contains the collected observations about the presumptions and explicitly shows, using examples of Flip Zooming, how these presumptions can be overruled on a general level to create alternative Focus+Context visualizations. **Figure 21** shows the material that formed the theoretical and empirical basis for the last paper in the thesis.

6 Conclusion

The development of Flip Zooming has been carried out by taking a design stance. To study the feasibility of the technique, a number of different visualizations have been implemented and put to use to gain experience and empirical data. Each of these visualizations has been constructed as a separate, and independent, artifact that has a value and use in itself. One attest to this is that the layout algorithm for each visualization has been implemented from scratch.

Beside the ambition to apply Flip Zooming in new areas of use to test its feasibility, the actual choice of platform and application area of the visualizations has in many cases been the result of serendipitous encounters. From such encounters, explicit research questions have been raised which in turn have determined the range of appropriate hardware and the constitution of user groups. The design and construction of the visualizations have then served to provide possible answers to the questions, and the specific applications have been validated or rejected according to their value in use.

Like other information visualization techniques, Flip Zooming does not offer a recipe that can be followed step-by-step in order to construct an interactive visualization on the basis of some initial requirements. For each visualization, the technique has been modified (a detailed summary of these modifications can be found in paper 6), either as to how it presents information or as to how one interacts with it, to a point where it can be difficult to see that the different applications use the same visualization technique. The reasons for the modifications have been numerous: the screen area available (both in available pixels and available inches), the type and amount of information to be visualized, the interaction that should be possible, and the location where that interaction should be possible.

As an example, a general algorithm for placing the tiles was initially developed for Flip Zooming based solely on the number of tiles and the size ratio between a context tile and the focus tile. This algorithm was then modified to allow hierarchical structures (described in paper 2), enabling a trade-off between the number of tiles in a visualization and number of levels in a hierarchy. However, each implementation of the technique called for a change in this algorithm, until no algorithm that could be explicitly (and mathematically) described was distinguishable. When the Digital Variants browser (paper 1) was developed, the place-

50

ment of focus was changed to better support the user's activities. With the WEST prototype (paper 3), the number of tiles was fixed, using hidden hierarchies whenever the number of tiles exceeded the predetermined number. In PowerView (paper 4), the algorithm did not place any tiles to the left and right of the focus, due to the shape of the display, and it used different algorithms for different information types. For example, in the overview of a month the tiles representing days were ordered in seven columns to correspond to the days of the week.

Even though several prototypes using Flip Zooming have been developed, the rather vague layout scheme has not become more detailed or structured. If anything, it has become less structured, as some of the examples explicitly break away from the scheme (paper 1). This does not mean that one cannot make generalizations about the visualizations. The general Flip Zooming technique has been described as a visualization for linearly ordered structures of discrete data [36]. Looking at the work contained within this thesis, paper 2 describes the hierarchical nature and general interaction provided by the technique, stripped of references to specific visualizations. Paper 5 goes further, and shows how different Focus+Context visualizations, including Flip Zooming, can be combined by regarding information visualizations as visualizations of visualizations. Paper 6 shows how the general layout of Flip Zooming has to be modified as the technique is applied to different areas of use. To summarize, the Flip Zooming technique should be seen as a collection of heuristic rules of thumb rather than a strict and predetermined method.

The guidelines developed while using Flip Zooming can be categorized into three groups; the guidelines dealing with the actual presentation of information by the visualization; the guidelines that are applicable to other Focus+Context visualization techniques besides Flip Zooming; and the guidelines that are of a more general nature, being applicable to most information visualizations.

6.1 Guidelines for Flip Zooming

Looking at the development of the heuristics for presenting information, Flip Zooming is based on the division of information into a number of sequential pieces of information, each placed and presented individually. As Holmquist states "there is quite a lot of freedom in deciding exactly how to construct a zoomed layout, both when it comes to the placement of the focus and the layout of the context elements" [36, p. 10]. The guidelines developed for presenting information with Flip Zooming are as follows:

Present Discrete sets of Information. The information should be displayed in rectangular tiles, each tile used to present its information as an individual piece of information. Besides being a natural way of presenting discrete sets of information, the division of information into separate units provides a common unit for selecting areas of interest.

Provide Clear Ordering. The tiles should be placed so that a left-to-right, top-to-bottom ordering is maintained, akin to how letters are placed in European languages to form words, sentences and whole texts. This ordering enables the same method of searching in Flip Zooming as in texts.

Provide a Non-Distorted Focus. One of the tiles is designated *the focus tile* and is given more space than the other tiles, while maintaining the original ratio between that tile's width and height. The purpose of this increased space is to allow a richer presentation of the information within the focus tile. Maintaining the width-height ratio makes the focus a zoomed-in version of the normal tile, so that the identification of the new presentation as a more detailed view of the normal tile is cognitively simple.

Provide a Locus for the Focus. The focus should be placed centrally in the presentation and should have the same location independently of which tile is the focus tile. Having the focus tile fixed to one place provides a static locus for the user's attention.

52

Provide Sequential and Random Access. Two different methods of changing focus should be provided: sequential movement to move the focus one step forward or backward, and random access to jump to any tile. The sequential method allows the user to go through the information set in an ordered fashion moving forward or backward, while random access allows the user to jump to any tile that catches the user's attention. These interaction methods should preferably be facilitated by different input devices to avoid confusion between the methods.

Maintain the Information Ordering. The user can change which tiles is the focus tile, causing a change in the layout but without changing the sequential ordering. This maintains the internal ordering of the information set at all times and enables sequential searching of the information.

Minimize Presentation Changes. When focus is changed, the number of context tiles moved should be minimized. This procedure lessens the cognitive load when focal changes occur, and limits the locations of any given tile to a few possible places.

Provide Location-Independent Interaction. The information shown in each tile can be manipulated and interacted with independently of the tile's placement. As Flip Zooming visualization can be shown within a tile, this allows for hierarchical visualizations where each visualization can be interacted with independently, as well as use of other information visualization techniques within Flip Zooming visualizations.

Of these guidelines, the third, fourth, fifth and eighth have been transgressed in the various applications due to requirements of the hardware used or the use area supported. It can be argued that those guidelines that have not been changed between applications are characteristics rather than guidelines. However, it is preferable not to treat them as such in order not to hinder the application of Flip Zooming in situations where these characteristics would be disadvantageous.

6.2 Guidelines for Focus+Context Techniques

The guidelines for Focus+Context visualization techniques in general are described in paper 6 when discussing preconceptions about Focus+Context techniques but are restated here to provide a complete enumeration.

Optimize the Number of Foci. The optimal number of foci is dependent on the intended use of the visualization. Focus+Context techniques usually restrict the user to having one focus within the presentation with the remaining information constituting a context. However, many techniques have shown that it is possible to allow the user to change the number of foci during the interaction with the visualization or to avoid having explicit foci at all (for an example of the latter, see Nonlinear Magnification Field [50]). The Digital Variant Viewer described in paper 1 illustrates this guideline by providing two outer foci to enable comparison between texts.

Locate the User's Center of Attention. A focus does not have to coincide with the user's center of attention. Most Focus+Context techniques have been developed for use on desktop computers where it is reasonable to assume that the user's center of attention is on the computer. When Focus+Context techniques are used on mobile computing devices, this assumption can be erroneous as the user's attention may be on an activity outside the device. The PowerCom application is an example of how a visualization may be designed to support activities rather than facilitate them.

Provide one Focus for each Activity. Multiple foci are feasible if the application is to support several activities. Most information visualizations are designed to support one activity that the user is performing, and to provide one focus for viewing the currently interesting information. However, if the application intends to support several activities that the user switches between, but does not perform simultaneously, it is feasible to provide one focus with its own context for each activity. Even though the ability to switch between different foci may introduce several views of the same information set that may differ greatly, these different views can match the user's shifts between activities without

54

requiring extensive interaction to match the view of the information with the current activity. Described briefly in paper 6, the functionally expanded version of PowerView, the PowerCom application, provides an example of how several foci may be provided to a user to support several activities.

Facilitate Differentiable Contexts allow User-Controlled Contexts. Hierarchical Flip Zooming provides the user with one context for each visualization in the hierarchy. These contexts are hidden when their focus is toggled to use all available display area within its visualization. This allows the user to customize the overall context so that only the relevant parts of the global context are visible. The Digital Variants Browser and the Hierarchical Image Browser [37] allows the user to hide inner contexts to both provide larger inner foci and allow the foci of different inner visualizations to be compared more easily.

Offer one View for each Activity. Enabling multiple views of information without focus change can provide specialized support for different types of activities. Depending on what activity the user is performing, different presentations of both focus and context may be advantageous. The WEST browser exemplifies the concept of multiple views by enabling three different views in order to facilitate three different search methods. PowerView uses different views to help the user gain an overview, to navigate the different information domains, and to have an integrated view of related information of various types.

Allow Heterogeneous Contexts. The type of information shown in various contexts can vary, allowing for heterogeneous information sets to be visualized. This allows for the construction of applications that support activities that require several information types simultaneously. The PowerView application illustrates how such a heterogeneous context can be applied to create a task-oriented approach to information visualization.

6.3   Guidelines for Information Visualization

Looking at a more general level, the development of Flip Zooming has given rise to a number of guidelines about information visualization as a whole and on its application in various situations.

Characteristic-Dependent Visualization. Understanding the characteristics of the information to be presented is important for visualizations to be successful. This reasoning can be seen in all work on information visualization, but for information visualization techniques to become used outside research, all techniques need to be further specialized towards the specific information visualized. As a trivial example, the calendar view in PowerView illustrates this by modifying the layout algorithm for tiles to resemble the layout in normal calendars.

Use-Dependent Visualization. The intended use of the information is as important as the characteristics of the information. This applies not only to what interaction should be possible in the visualization, but also to how the actual visualization should present information. An example of the use of this guideline can be found in how the Flip Zooming layout was modified in the Digital Variants browser. Due to the need of enabling comparisons of different pages, the layout algorithm was changed in two different ways for the two levels of visualization in the application.

Device-Dependent Visualization. The device on which the visualization is going to be used is of vital importance. The various computer devices (e.g. stationary computers, laptop computers, handheld PDAs, palmheld PDAs, and mobile phones) currently available have very different screen characteristics such as size, resolution, width-height ratio, and color range, forcing trade-offs as visualization techniques are applied on new hardware platforms. Further, the various forms of input facilitated by the devices (e.g. mice and keyboards) have implications for how the visualization (and the information shown by it) can be interacted with. WEST and PowerView show how the characteristics of the used device, and its input facilities, greatly affect the visualization technique.

56

Location-Dependent Visualization. As computer devices are beginning to be used in areas outside the predictable environment of an office desk, the assumptions of the user's capabilities to interact with a visualization change. Besides leading to use situations where the information needed can differ from that of an office, the environment may pose restrictions on the use. For example, a visualization used when the user is mobile has to consider the possibility that light sources may change during use, that the user and the device may move (and shake), and that the user may not be able to use both hands to interact with the device. The PowerView application shows how Flip Zooming was adapted to suit mobile situations.

6.4 Final Remarks

These lists of guidelines and rules of thumbs are not complete. By further studying the applications, and by developing new applications for other types of information and areas of use, the lists can be expanded. Further, it should be noted that as the lists are lists of guidelines, not all are applicable in any given situation, and two guidelines may very well contradict each other, requiring a trade-off. Looking at the individual applications and theories that are the result of the development of Flip Zooming, the contribution of the thesis includes:

- The design, construction and evaluation of the Digital Variants Browser.
- The design, construction and evaluation of the WEST Browser.
- The design, construction and evaluation of PowerView.
- The generalization and description of the Flip Zooming technique as an hierarchical visualization technique.
- The implicit description of the Flip Zooming technique by the use of a collection of individually different applications that are related by the way presentation and interaction is facilitated.
- The framework that allows visualizations to be described as nested visualizations, allowing a clear description of how various visualization techniques can be intertwined within one computer application.
- The identification of various preconceptions about Focus+Context techniques. By identifying these, the thesis indicates how they can be transcended, making it possible to expand the design space of Focus+Context techniques.
- The comparison of Flip Zooming and other visualization techniques, highlighting features and areas of applicability to the techniques.
- The enumeration of guidelines and rules of thumb identified while working on the various Flip Zooming applications.

In conclusion, the ambition of all the papers in this thesis has been to develop Flip Zooming in order to facilitate access for more people to digital information while still using traditional computer displays. The different papers have focused on different types of information, different groups of users, and different areas of use – all in order to provide a multifaceted exposition of the technique's feasibility. In doing so, the papers contribute to the more general goal of enabling people generally to gain more knowledge from information presented with the help of information technology.

58

7   References

1. Apperley, M., Fletcher, D., Rogers, B., Thomson, K. Interactive Visualization of a Travel Itinerary. In *Proceedings of Advanced Visual Interfaces 2000 (AVI 2000)*, pp. 221-226, ACM Press, 2000.

2. Bartram, L., Ho. A., Dill, J., and Henigman, F. The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces. In *Conference Proceedings of User Interface Software and Technology '95 (UIST '95)*, pp. 207-215, ACM Press, 1995.

3. Bederson, B., Hollan, J., Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Conference Proceedings of User Interface Software and Technology '94 (UIST'94)*, pp. 17-26, ACM Press, 1994.

4. Bederson, B.B., and McAlister, B. Jazz: An Extensible 2D+Zooming Graphics Toolkit in Java. *Technical Report HCIL-99-07, CS-TR-4015, UMIACS-TR-99-24*, Computer Science Department, University of Maryland, College Park, MD. Available at http://www.cs.umd.edu/~bederson/papers/. Last visited 2000-07-21.

5. Bederson, B.B., and Boltman, A. Does Animation Help Users Build Mental Maps of Spatial Information? In *Proceedings of IEEE Symposium on Information Visualization '99 (InfoVis '99)*, pp. 28 - 35, IEEE Press, 1999.

6. Benford, S., Bederson, B.B., Åkesson, K., Bayon, V., Druin, A., Hansson, P., Hourcade, J.P., Ingram, R., Neale, H., O'Malley, C., Simsarian, K.T., Stanton, D., Sundblad, Y., and Taxén, G. Designing Storytelling Technologies to Encourage Collaboration Between Young Children. In *Proceedings of ACM SIGCHI Conference of Human Factors in Computing Systems (CHI 2000)*, pp. 556- 563, ACM Press, 2000.

7. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., and DeRose, T.D. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics (SIGGRAPH '93)*, pp. 73-80, ACM Press, 1993.

8. Bier, E.A., Stone, M.C., Fishkin, K., Buxton, W., and Baudel, T. A taxonomy of see-through tools. In *Conference proceedings on Human factors in computing systems (CHI '94)*, pp. 358-364, ACM Press, 1994.

9. Björk, S., and Holmquist, L.E. Formative Evaluation of a Focus+Context Technique. Poster presentation at *Annual Conference of the British HCI Group (HCI '98)*, Sheffield, UK, 1998. Full version (technical report) available at http://www.viktoria.informatik.gu.se/groups/play/publications/1998/formative_full.pdf.

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

11. Card, S.K., Robertson, G.G., and Mackinlay, J.D. The information visualizer, an information workspace. In *Proceedings of Human factors in computing systems (CHI 91)*, pp. 181-186, ACM Press, 1991.

12. Carpendale, M.S.T., Cowperthwaite, D.J., and Fracchica, F.D. 3-Dimensional Pliable Surfaces: For the Effective Presentation of Visual Information. In *Conference Proceedings of User Interface Software and Technology '95 (UIST '95)*, p. 217-226, ACM Press, 1995.

13. Carpendale, M.S.T., Cowperthwaite, D.J., and Fracchica, F.D. Extending Distortion Viewing from 2D to 3D. In *IEEE Computer Graphics and Applications*, July-August, 1997. Reprint in Card, S.K., *et al.* (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

14. Casner, S.M. Task-analytic approach to the automated design of graphic presentations. In *ACM Transactions on Graphics*. Vol. 10, No. 2, pp. 111-151, ACM Press, 1991.

15. Dahlbom, B. (ed.). *The Infological Equation*, Gothenburg Studies in Informatics, Report #6, Göteborg University, 1995.

16. Dahlbom, B., and Ljungberg, F. Mobile Informatics. In *Scandinavian Journal of Information Systems*, Vol. 10, No. 1&2, 1999.

17. Dahlbom, B., and Mathiassen, L. *Computers in Context*, Blackwell Publishers Inc., 1993 (reprint in 1996).

18. Egan, D.E., Remde, J.R., Gomez, L.M., Landauer, T.K., Eberhardt, J., and Lochbaum, C.C. Formative design evaluation of superbook. In *ACM Transactions on Information Systems*, pp. 30-57, ACM Press, 1989.

19. Fairchild, K.M., Poltrock, S.E., and Furnas, G.W. SemNet: Three-Dimensional Representations of Large Knowledge Bases. In R. Guidon (ed.), *Cognitive Science and Its Applications for Human-Computer Interaction*, pp. 201-233, Lawrence Erlbaum Associates. Reprint in Card, S.K., *et al.* (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., 1999.

20. Farrand, W.A. *Information display in interactive design.* Doctoral dissertation, Department of Engineering, University of California, Los Angeles, California, 1973.

21. Fox, D. *Tabula Rasa: A Multi-scale User Interface System.* Doctoral dissertation, New York University, NY. Available at: http://www.fox-thompson.com/dsf/diss/ (Last visited 2000/07/24).

22. Furnas, G.W. *The FISHEYE View: A New Look at Structured Files*. Bellcore Technical Report, 1981. Reprinted in: Card, S.K., *et al.* (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

23. Furnas, G.W. Generalized Fisheye Views. In *Conference proceedings on Human factors in computing systems '86 (CHI '86)*, pp. 16-23, ACM Press, 1986.

24. Furnas, G.W. Effective View Navigation. In *Proceedings of CHI'97*, pp. 367-374, ACM Press, 1997. Reprinted in: Card, S.K., *et al.* (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

25. Furnas, G.W., and Zacks, J.Multitrees: enriching and reusing hierarchical structure. In *Conference proceedings on Human factors in computing systems (CHI '94)*, pp. 330-336, ACM Press, 1994.

26. Furnas, G.W., and Bederson, B.B. Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Conference proceedings on Human factors in computing systems '95 (CHI '95)*, pp. 234-241, ACM Press, 1995

27. Greenberg, S., and Thimbleby, H. The weak science of human-computer interaction. In *CHI '92 Research Symposium on Human Computer Interaction*, Monterey, California, 1992.

28. Greenberg S., Gutwin, C. and Cockburn, A. Awareness Through Fisheye Views in Relaxed-WYSIWIS Groupware. In *Proceedings of Graphics Interface '96 (GI '96)*, pp. 28-38, Morgan-Kaufmann Publishers, 1996.

29. Greenberg, S., Gutwin, C. and Cockburn, A. Using Distortion-Oriented Displays to Support Workspace Awareness. In *People and Computers XI (Proceedings of the HCI '96)*, pp. 299-314, Springer-Verlag, 1996.

30. Hellstrand, M. *Comparing the PowerView interface with the traditional Windows CE interface: Novice users favour two-handed navigation over one-handed on PDAs*. Master thesis in psychology, Department of Psychology, Lund University, 1999. Available at http://www.viktoria.informatik.gu.se/groups/play/publications/1999/hellstrand.pdf (last visited 2000/07/24)

31. Hellstrand, M., Goldstein, M., Anneroth, M., Werdenhoff, J., and Chincholle, D. *An Action Control but no Action: Users Dismiss Single-Handed Navigation on PDAs*. In *Proceedings of First Nordic Conference on Computer-Human Interaction 2000 (NordiCHI 2000)*.

32. Henderson, D.A., and Card, S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical

user interface. In *ACM Transactions on Graphics*, Vol. 5, No. 3, pp. 211-243, ACM Press, 1986.

33. Herman, I., Melançon, G., and Marshall, M.S. Graph Visualization and Navigation in Information Visualization: A Survey. In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, pp. 24-43, IEEE Computer Society, 2000.

34. Holmquist, L.E. Focus+Context Visualization with Flip Zooming and the Zoom Browser. In *Extended Abstracts of ACM SIGCHI Conference of Human Factors in Computing Systems (CHI' 97)*, pp. 263-264, ACM Press, 1997.

35. Holmquist, L.E. The Zoom Browser: Showing Simultaneous Detail and Overview in Large Documents. In *Human IT*, Vol. 2, No. 3, pp. 131-150, ITH, Borås, Sweden, 1998.

36. Holmquist, L.E. *Flip Zooming: Focus+Context Visualization of Linearly Ordered Discrete Visual Structures*. In Holmquist, L.E. *Breaking the Screen Barrier*, Ph.D. thesis, Department of Informatics, Göteborg University, Sweden. ISSN: 1400-741X.

37. Holmquist, L.E., and Björk. S. A Hierarchical Focus + Context Method for Image Browsing. In *SIGGRAPH 98 Sketches and Applications*, p. 282, ACM Press, 1998.

38. Inselberg, A. Multidimensional Detective. In *Proceedings of IEEE Symposium on Information Visualization '97 (InfoVis '97)*, pp. 100-107, IEEE Press, 1997.

39. Jeong, C.S., and Pang, A. Reconfigurable Disc Trees. In *Proceedings* of *IEEE Symposium on Information Visualization '98 (InfoVis '98)*, pp. 19-25, IEEE Press, 1998.

40. Jin, L., and Banks, D.C. TennisViewer: A Browser for Competition Trees. In Card, S.K., *et al*. (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

41. Johnson, B., and Shneiderman, B. Tree-maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In *Proceedings of IEEE Visualization '91*, pp. 284-291, IEEE Computer Society, 1991.

42. Kadmon, N., and Shlomi, E. A polyfocal projection for statistical surfaces. *Cartograph*, Vol. 15, No. 1, pp. 36-40, 1978.

43. Kaltenbach, M., Robillard, F., and Frasson, C. Screen Management in Hypertext Systems with Rubber Sheet Layouts. In *Proceedings of Hypertext '91*, pp. 91-105, ACM Press, 1991.

44. Kaugars, K., and Brazma, A. CATGraph: Visualizing Large Labeled Graphs. *New Mexico State University Technical Report NMSU-TR-92-CS-08*, April 1992.

45. Keahey, T.A. *Nonlinear magnification*. Ph.D. Dissertation, Indiana University Computer Science, 1997. Available at http://www.cs.indiana.edu/hyplan/tkeahey/research/papers/thesis.html (last visited 2000/07/28)

46. Keahey, T.A. The Generalized Detail-In-Context Problem. In *Proceedings of IEEE Symposium on Information Visualization '98 (InfoVis '98)*, pp. 44-51, Information Visualization Symposium, IEEE Press, 1998.

47. Keahey, T.A. Visualization of high-dimensional clusters using non-linear magnification. In *Visual Data Exploration and Analysis VI*, volume 3643 of SPIE, January 1999.Available at http://www.cs.indiana.edu/hyplan/tkeahey/research/papers/spie.99.html (last visited 2000/07/28).

48. Keahey, T.A. Area-normalized thematic views. In *Proceedings of International Cartography Assembly*, August 1999. Available at http://www.cs.indiana.edu/hyplan/tkeahey/research/papers/ica.99.html (last visited 2000/07/28).

49. Keahey, T.A., and Robertson, E.L. Techniques for non-linear magnification transformations. In *Proceedings of the IEEE Symposium on Information Visualization '96 (InfoVis '96)*, pp. 38-45, IEEE Press, 1996.

50. Keahey, T.A., and Robertson, E.L. Non-Linear Magnification Fields. In *Proceedings of IEEE Symposium on Information Visualization '97 (InfoVis '97)*, pp. 51-58, IEEE Press, 1997.

51. Koike, H. Fractal Views: A Fractal-Based Method for Controlling Information Display. In *ACM Transactions on Information Systems*, Vol. 13, No. 3, pp. 305-323, ACM Press, 1995.

52. Lamping, J., Rao, R., and Pirolli, P. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Conference proceedings on Human factors in computing systems '95 (CHI '95)*, pp. 401-408, ACM Press, 1995.

53. Larkin, J., and Simon, H.A. Why a Diagram is (Sometimes) Worth Ten Thousand Words. In *Cognitive Science*, Vol. 11, No. 1, pp. 65-99, 1987. SKAFFA

54. Leung, Y.K., and Apperley, M.D. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. In *ACM Transactions on Computer-Human Interaction*, Vol. 1, No. 2, pp. 126-160, ACM Press, 1994. Reprint in Card, S.K., *et al.* (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., 1999.

55. Mackay, W.E., and Fayard, A-L. HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines. In *Proceedings*

of ACM Designing Interactive Systems '97 (DIS '97), pp. 223-234, ACM Press, 1997.

56. Mackinlay, J. Automating the design of graphical presentations of relational information. In *ACM Transactions of Graphics*. Vol. 5, No. 2, pp. 110-141, ACM Press, 1986.

57. Mackinlay, J.D., Robertson, G.G., and Card, S.K. The Perspective Wall: Detail and Context Smoothly Integrated. In *Conference proceedings on Human factors in computing systems '91 (CHI '91)*, pp. 173-179, ACM Press, 1991.

58. Mackinlay, J.D., Robertson, G.G., and DeLine, R. Developing calendar visualizers for the information visualizer. In *Proceedings of the ACM symposium on User interface software and technology '94 (UIST '94)*, pp. 109-118, ACM Press, 1994.

59. Mackinlay, J.D., Rao, R., and Card, S.K. An organic user interface for searching citation links. In *Conference proceedings on Human factors in computing systems '95 (CHI '95)*, pp. 67-73, ACM Press, 1995.

60. Muller, M.J., and Kuhn, S. Participatory design. In *Communications of the ACM*, Vol. 36, No. 6, pp. 24-28, ACM Press, 1993.

61. Munzner, T. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, pp. 2-10, Information Visualization Symposium, IEEE Press, 1997.

62. Munzner, T. *Interactive Visualization of Large Graphs and Networks*. Ph.D. Dissertation, Stanford University, June 2000. Available at: http://graphics.stanford.EDU/papers/munzner_thesis/ (last visited 2000/07/24).

63. Nielsen, J., and Molich, R. Heuristic evaluation of user interfaces. In *proceedings of Empowering people: Human factors in computing system (CHI '90)*, pp. 249-256, ACM Press, 1990.

64. Norman. D.A. *The Design of Everyday Things*. Bantam Doubleday Dell Publishing Group, 1990.

65. Norman, D.A. *Things that make us smart - Defending human attributes in the age of the machine*, Perseus Books, 1993.

66. Norman, D.A. *The Invisible Computer*, MIT Press, 1998.

67. Perlin, K., and Fox, D. Pad: an alternative approach to the computer interface. In *Proceedings of the 20th annual conference on Computer graphics (SIGGRAPH '93)*, pp. 57-64, ACM Press, 1993.

68. Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. Context and Interaction in Zoomable User Interfaces. In *Proceedings of Advanced Visual Interfaces 2000 (AVI 2000)*, pp. 227- 231, ACM Press, 2000.

64

69. Rao, R., Pedersen, J.O., Hearst, M.A., Mackinlay, J.D., Card, S.K., Masinter, L., Halvorsen, P-K., and Robertson, G.G. Rich interaction in the digital library. In *Communications of the ACM*, pp. 29-39, Vol. 38, No. 4, ACM Press, 1995.

70. Rao, R., Russell, D.M., and Mackinlay, J.D. System components for embedded information retrieval from multiple disparate information sources. In *Proceedings of the sixth annual ACM symposium on User interface software and technology '93 (UIST '93)*, pp. 23-33, ACM Press, 1993.

71. Rao, R., and Card, S.K. The Table Lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceeding of Human factors in computing systems (CHI' 94)*, pp. 318-322, ACM Press, 1994.

72. Robertson, G.G., Card, S.K., and Mackinlay, J.D. The cognitive coprocessor architecture for interactive user interfaces. In *Proceedings of the ACM SIGGRAPH symposium on User interface software and technology '89 (UIST '89)*, pp. 10-18, ACM Press, 1989.

73. Robertson, G.G., Mackinlay, J.D., and Card, S.K. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of Human factors in computing system (CHI '91)*, pp. 189-194, ACM Press, 1991.

74. Robertson, G.G., and Mackinlay, J.D. The Document Lens. In *Proceedings of User interface software and technology '93 (UIST '93)*, pp. 101-108, ACM Press, 1993.

75. Rodden, K., Basalaj, W., Sinclair, D., and Wood, K. Evaluating a Visualization of Image Similarity as a Tool for Image Browsing. In *Proceedings of IEEE Symposium on Information Visualization '99 (InfoVis '99)*, IEEE Press, 1999.

76. Sarkar, M. and Brown, M.H. Graphical Fisheye Views of Graphs. In *Proceedings of Human Computer Interaction '92 (CHI '92)*, pp. 83.91, ACM Press, 1992.

77. Sarkar, M. and Brown, M.H. Graphical Fisheye Views. In *Communications of the ACM*, Vol. 37, No. 12, pp. 73-84, 1994.

78. Sarkar M., Snibbe, S.S., Tversky, O.J., Reiss, S.P. Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In *Proceedings of ACM User interface software and technology '93 (UIST '93)*, pp. 81-91, ACM Press, 1993.

79. Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., and Roseman, M. Navigating hierarchically clustered networks through fisheye and full-zoom methods. In *ACM Transactions on Computer-Human Interaction*, Vol. 3, No. 2, pp. 162-188, ACM Press, 1996.

80. Shipman, F.M., Marshall, C.C., and LeMere, M. Beyond location: hypertext workspaces and non-linear views. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia (HYPERTEXT '99)*, pp. 121-130, ACM Press, 1999.

81. Shneiderman, B. Dynamic Queries for Visual Information Seeking. In *IEEE Software*, Vol. 11, No. 6, pp. 70-77, IEEE Press, 1994.

82. Simon, H.A. *Science of the Artificial*. 3rd ed. Cambridge, MA: MIT Press, 1996.

83. Spence, R., Apperley, M., Data base navigation: an office environment for the professional. In *Behavior and Information Technology*, Vol. 1, No. 1, pp. 43-54, 1982.

84. Spence, R. A taxonomy of graphical presentation. In *INTERACT '93 and CHI '93 conference companion*, pp. 113-114, ACM Press, 1993.

85. Stone, M.C., Fishkin, K., and Bier, E.A. The movable filter as a user interface tool. In *Conference proceedings on Human factors in computing systems (CHI '94)*, pp. 306-312, ACM Press, 1994.

86. Taivalsaari, A. *The Event Horizon User Interface Model for Small Devices*, Technical Report TR-99-74, Sun Microsystems Laboratories, 1999. Available at http://www.sunlabs.com/technical-reports/1999/ (last visited 2000/07/24).

87. Tufte, E.R. *The Visual Display of Quantitative Information*, Cambridge University Press, 1983.

88. Tufte, E.R. *Envisioning Information*, CT: Graphics Press, 1990.

89. Tufte, E.R. *Visual Explanations: Images and Quantities, Evidence and Narrative*, CT: Graphics Press, 1997.

90. Tweedie, L. Characterizing Interactive Externalizations. In *Conference proceedings on Human factors in computing systems (CHI '97)*, ACM Press, 1997.

91. Tweedie, L., Spence, R., Dawkes, H., and Su, H. Externalizing Abstract Mathematical Models. In *Conference proceedings on Human factors in computing systems (CHI '94)*, pp. 406-412, ACM Press, 1994.

92. van Wijk, J.J., and van de Wetering, H. Cushion Treemaps: Visualization of Hierarchical Information. In *Proceedings of IEEE Symposium on Information Visualization '99 (InfoVis'99)*, pp. 73-78, IEEE Computer Society, 1999.

93. Wattenberg, M. Visualizing the Stock Market. In *Extended Abstracts of Computer-Human Interaction 2000 (CHI 2000)*, pp. 188-189, ACM Press, 1999.

94. Wilson, R.M., and Bergeron, R.D. Dynamic Hierarchy Specification and Visualization. In *Proceedings of IEEE Symposium on Informa-*

tion Visualization '99 (InfoVis '99), pp. 65-72, IEEE Computer Society, 1999.

95. Woodruff, A., Landay, J., and Stonebraker, M. Constant Density in Zoomable Interfaces. In *Proceedings of Advanced Visual Interfaces (AVI '98)*, pp. 57-65, ACM Press, 1998.

96. Woodruff, A., Landay, J., and Stonebraker, M. Constant density visualizations of non-uniform distributions of data. In *Proceedings of User interface software and technology (UIST '98)*, pp. 101-108, ACM Press, 1998.

Exploring the Literary Web:
The Digital Variants Browser

Staffan Björk, Lars Erik Holmquist

Abstract. This paper describes the Digital Variants Browser, a novel prototype for visualizing large amounts of text information. It is specifically designed to help the study of several different versions or writing stages of the same text by providing an overview of the entire text material. The overview visually shows the changes between the different versions of the text and makes comparisons easier. The browser has been developed at the Viktoria Institute in association with the Digital Variants Project and utilizes Flip Zooming, a focus + context visualization technique. The paper discusses the development of the techniques used in the prototype.

1   Introduction

The increased availability of digital resources, e.g. literary databases, and the possibility of getting information over networks, have vastly enhanced the possibilities of humanities scholars. Researchers have no longer to travel to specialised libraries to find rare books and documents, but can instead download texts and corpora on their computers. The scholar can then work with the text by making advanced searches, annotating the text and comparing it with other texts using manual or automatic procedures. This has allowed an increasing number of textual experts (i.e. philologists) to do comparative studies of several different variants of the same text. But the problems that exist while working with a large text, such as giving a clear overview and providing an effective presentation, increase dramatically if several texts are involved, especially if the task is to compare one text with another.

The aim of the Digital Variants Project, promoted by the Department of Italian at the University of Edinburgh, was to provide researchers with a comparative tool for studying contemporary authors' variant texts. The DV project (http://www.ed.ac.uk/~esit04/digitalv.htm) makes text available on the Internet in different stages of writing, showing the

complex writing phenomena underlying the final version of a literary work. This is useful in both studying the development of a particular text, and of writing techniques in general. However, in order to let users interact with the material in a fruitful way, a new, specialized, application had to be developed, as current applications for working with texts do not give the required functionality. The following functions were identified as essential for the application to help researchers work with digital texts:

- Explorative presentation. The application has to support the user in exploring texts in order to discover new data and find new connections.
- Enabling Comparison. To study several variants (or, in philological terms, 'witnesses') of the same document, it is imperative that one can compare one text to another.
- Showing a text as one piece of work. This is crucial since printed texts are mostly regarded as 'indivisible' objects, i.e. one can not get a complete understanding of a text without being able to look at it as a whole.
- Showing all variants of a text. Not only a single text needs to be visualized, but also many variants have to be visualized at the same time while keeping the presentation clear.
- Enabling Search methods. All the material should be searchable and all the results (i.e. all the hits) of a search should be displayed simultaneously in their respective context.

An additional, technical requirement of the application was:

- To be a light-weight application. If users should use the application with average computing facilities, the application cannot rely on heavy computational procedures that can be only performed by last-generation machines.

At the Viktoria Institute of Gothenburg, we have been developing the Flip Zooming technique, a general focus+context visualization technique that addresses the above mentioned issues. As a constructive contribution the Digital Variants project, we developed an application based on our visualization technique that enables users to compare several variant of a text. This paper describes the application, the Digital Variants Browser (DV Browser), and gives some examples of its usability.

Before presenting details about the design and implementation choices made, a background to the most relevant theoretical issues related to the project is given.

1.1 Information retrieval methods

Computer systems that have the primary aim of providing users with information, such as image sets or documents, are called Information Retrieval Systems (IRS). Various forms of these systems have been developed over the years and they have been the focus of extensive study within the computer science, as many of the systems are vital to industry and the society. One definition of IRS is: "An Information Retrieval System is a system that is capable of storage, retrieval, and maintenance of information" (Kowalski 1996, p. 2).

Even though the problem of visualizing the information is not mentioned, the definition shows that IRS have many of the same requirements as those of the DV Browser. Thus, a study of the techniques used in information retrieval systems was essential for developing the browser.

Over the years many different types of information retrieval systems have emerged (for an extensive description, see Salton 1989). The oldest, and most common, information retrieval systems are called conventional text retrieval systems, and include techniques such as database searches, formal query statements and elementary string matching. These systems, however, do not give users supplementary information about the material (except for what is explicitly requested), so the user must know in advance what he/she is looking for. This is could be a problem when new material needs to be explored as the only way for a user to get an overview of the material is to literally read the whole material. When the items stored in an information retrieval system become too large and too many -- this is the case of whole books or libraries -- simple search methods break down or, put in other words: "In reality ... the number of words or sentences included in many items may be so large that a complete text comparison between different information items become impossible" (Salton 1989, p. 275).

One way to mitigate this problem is to assign identifiers or tags to documents using markup languages such as SGML and XML. This technique is called indexing, and can be done either by human experts or

by automatic indexing using computers. Other, more advanced models, which help facilitate search techniques are vector space modelling, clustering and fuzzy set extensions. All these techniques analyse the text material in order to build abstract representations of the text, so that comparisons can be made. However, they do not provide a clear way of showing how the visualization of the text corresponds to the abstract visualization. Similarly, Linguistic approaches such as text analysis and language analysis also construct abstract representations, but these representations are based on syntactic and semantic interpretations.

Looking at the techniques just mentioned, it soon became clear that these were not appropriate for the study of digital variants. The common text retrieval systems do not promote an explorative study of the material, and the creation of a system that can perform automatic indexing suited for a particular comparative study requires so much knowledge of the text, or knowledge about what the users want, that its use would be self-defeating. Vector space modelling, clustering, fuzzy set extension and linguistic approaches improve searching, and help find similar but not identical texts fragments, but they do not provide a clear overview of the original text.

In fact, all these methods were developed before Graphical User Interfaces (GUI) were introduced, or were developed without taking into consideration the possibilities offered by graphical representations. When GUIs started to be developed for mainstream use in the beginning of the 1980's, the applications using information retrieval techniques were converted to the new systems, but they did not take full advantage of the new technology of data representation.

Visual information can be extremely rich in content, and humans are good at exploring and comparing information by quickly scanning information using their eyesight. Therefore, using graphical representations of the different variants of a text seemed to be a promising approach. Methods that use graphical representations to display information are called visualization techniques. Our aim with using such techniques was to create an application that could both show a large amount of information and promote explorative searches.

1.2   Visualization techniques

During recent years, many techniques for visualizing data set on computers have been developed, but most of them showed only the structure of the data set or a subgroup of it. In other words, "The [visualizations techniques] focused on Information Retrieval Systems are investigating how best to display the results of searches, structured data from DBMSs and the results of link analysis correlating data" (Kowalski 1996, 190).

Our browser needed a technique that would focus on how to show the whole material and use search strategies that people have developed in every-day life for working with large amounts of data. This would lead to an intuitive, easy to learn tool. But what search strategies do people have? They spread out manuscripts on their desks to get an overview; put up complicated diagrams on the wall and take a step back to get the full picture; arrange notes and pictures on large surfaces to find meaningful patterns. An equivalent to these strategies applied to computer graphics can be found in the various techniques collectively termed focus + context visualization techniques. These techniques attempt to give users both the overview and the details they need at the same time. Such techniques take their inspiration from the way that human vision works: they show the centre of attention, the focus, with a high amount of detail, while showing the surrounding information, the context, with much less detail. Thus, users can move their centre of attention to different areas, and at the same time maintain a general overview.

1.2.1 A summary of focus + context techniques and applications. The generalized fisheye view provided the basis for much of the subsequent work on focus + context presentation (Furnas 1986). An early graphical focus + context technique was the bi-focal display, which introduced horizontal distortion of the material outside the focus (Spence and Apperley 1982). Techniques that introduce both horizontal and vertical distortion include the rubber sheet view (Sarkar et al. 1993) and the graphical fisheye view (Sarkar and Brown 1994). The continuous zoom (Bartram et al. 1995) provided a fisheye view of graphs while maintaining the relative locations of the nodes in the graphs. Several techniques have used distortion inspired by geometry and optics to provide a focus and context, including the perspective wall (Mackinlay et al. 1991) and the document lens (Robertson and Mackinlay 1993). Several applications using these techniques provide views of the World Wide Web, including the

hyperbolic tree browser (Lamping et al. 1995), zippers (Brown and Weihl 1996), the web book and web forager (Card et al. 1996), image-trees (Wistrand 1996), and the zooming web-browser based on the PAD++ toolkit (Bederson et al. 1994, 1998a and 1998b). An overview of distortion-oriented techniques is given in Leung and Apperley 1994, and a relevant discussion of effective view navigation is given in Furnas 1997. For further details about focus + context techniques, and other information visualization techniques, 'Readings in Information Visualization: Using Vision to Think' (Card et al. 1999) provide a start-of-the-art presentation of research conducted in this field of research.

1.2.2 Evaluations of focus + context applications. Evaluation of a zooming web-browser based on the PAD++-interface showed that experienced users found answers to questions significantly faster than when using a traditional browser (Bederson et al. 1998a) and fisheye views showed a slight advantage over scrolling on traditional displays (Hollands et al. 1989). The performance of fisheye and full zoom views of a telephone network showed subjects completing a task more efficiently using the fisheye view (Schaffer et al. 1996). An evaluation of elastic windows showed that this system allows faster performance than traditional window systems in some tasks (Kandogan and Shneiderman et al. 1997). Finally, the CZWeb browser proved to be useful for navigating and understanding the organization of the World Wide Web (Fisher et al. 1997).

2   The Flip Zooming technique

2.1   Assumptions

As is evident from the discussion above, a large number of focus + context techniques have been developed. However, few seem to have found their way into real-world products. One reason may be that many of the techniques involve advanced calculations and hence require expensive hardware to be used effectively. Another reason may be that they are best suited for specialized data sets, such as graphs, and that there is no easy way to adapt existing data for use with these techniques. Finally, it is possible that some of these techniques are simply too complicated to

use for the non-specialist, and that the advantages a user can have from these methods are countered by the efforts needed to understand and learn how to use them.

When the Flip Zooming technique was developed, it was seen as very important that the solutions were practical and put low demand on the hardware that users would need. The technique should also be easy to understand, and flexible enough to adapt to many different data sets. What was needed was a flexible, light-weight focus + context technique.

2.2 Development

After studying the earlier focus + context techniques, experiments started with the goal of finding various ways to apply a focus view to a set of images laid out sequentially on a 2-dimensional surface by using thumbnails. The first attempt was inspired by the document lens (Robertson and Mackinlay 1993). A problem with this view was that the thumbnails close to the focus were smaller than those further away, which was undesirable, since the information closest to the focus is often regarded as more important for the context. The second attempt was inspired by the rubber sheet view (Sarkar et al. 1993) but without deformation of the individual thumbnails. This view had the advantage of a clear correlation between the image distribution in the non-focus and focus views, but made poor use of the display.

The next method, in which the technique started to be applicable, laid out the data set (e.g. the pages of a document or a set of images) sequentially as a collection of thumbnail representations. When a thumbnail was brought into focus, it zoomed to full size and was placed approximately in the middle of the display. The remaining thumbnails were reduced in size and arranged around the focus by putting the thumbnails that come before the focus image in the sequence above and to the left of the focus page, and the following thumbnails were placed below and to the right (for a more detailed description of the development up to this part, see Holmquist 1998).

After evaluating the third method, it was found that it did not communicate well the exact place of the focus image in the sequence, and it was understood that a clear structure of the presentation was more important than maximizing the screen use. Therefore, in the development of the next method we decided to place emphasis on having a con-

sistence presentation: this meant that the focus would always appear in the same area with thumbnails placed before and after it as required (see Björk and Holmquist 1998).

The system was also expanded so that the data set visualized was generic and each component could be a separate application (i.e. could react to input from the user). This meant that the system could be tested in a number of settings, as a variety of different applications could easily be added to the system. As the data items were components, they could themselves be application using the Flip Zooming technique that meant that hierarchical structures could easily be facilitated. Hierarchical structures are important in visualization techniques when we have to visualize extremely large sets of information, or relations between distinct items (see Holmquist and Björk 1998).

2.3   Use description

After giving a brief description of the actual use of the technique, users navigate through a data set by flipping through it, like the pages in a book. When they want to examine an entry, such as a document page, they selected it by clicking on a page image. The image is then zoomed to a readable size. The surrounding pages are reduced in size and re-arranged to accommodate the expanded image of the page. When users want to view a new entry, they can either flip to the next or previous item in the sequence clicking on a button or using a keyboard short-cut, or they can select randomly a new entry from another part of the document by clicking on the relevant page. If the user specifically wishes to only see one entry, simply clicking on the focus, which hides all other entries and uses all available screen space to show the focus, facilitates this. This is useful for studying one entry in higher detail when the context is unnecessary, or to provide an icon or index for a presentation that has extremely small space in a hierarchical structure.

Looking back at the features required by the DV Browser, Flip Zooming seemed to be a good candidate for visualizing the text material. It provides a presentation that encourages explorative searching, as the whole information set is presented and the user can access any part of the information from different points of view. It shows a text as one piece of information with a clearly ordered structure by using the traditional metaphor of pages, and containing all pages within a frame. Each

variant of a text is treated as a single piece of information in an "outer" Flip Zooming presentation that allows all variants to be shown and compared to each other at once. The technique does not require heavy computations, making it an alternative for users without powerful (and thus expensive) computer systems.

However, the general Flip Zooming technique does not provide text search functions nor specifically addresses the issue of comparing two items of the data set presented. These features would have to be added to the general (generic) technique for becoming applicable in the DV Browser.

3   The Digital Variants Browser

The Flip Zooming technique was developed using rapid prototyping, so the natural choice for developing a platform was to use the Java programming language. This has proven to be a good choice, as it has been easy to expand and enhance the system for the specific requirements of the DV Browser. Another advantage is that the portability of Java applets makes easy to use the tool in conjunction with web browsers -- and access of the material trough the Internet is one of the main objectives of the DV Project.

The application takes a number of text documents, each one a different version of the same text, and creates a flip zooming presentation of each of them. All these presentations are then used as components in an

Fig. 1. A typical view of the DV Browser showing six variants in which each variant text is composed of between one and eight pages.

outer flip zooming presentation to allow the change of focus between the whole variants (see Figure 1).

As the Flip Zooming technique was developed as a generic system for information visualization, and not as a dedicated technique for researchers in literature, it does not provide all the functionality required for the work these users perform. Therefore, in the DV Browser the Flip Zooming technique was redesigned to facilitate two functions: searching and comparing.

3.1 Comparing

Since the DV Browser should allow comparison between different versions or writing stages of a text, and each variant text was to be shown with the Flip Zooming technique, this meant that the focus in each presentation would be compared with a focus in another variant text. The original technique placed the focus in the centre, as this is where the main attention is naturally placed. But if the attention moves between two foci in different variant texts, one will pass both foci's context. As this makes the actual comparison more difficult, the foci were placed

Fig. 2. A view of the DV Browser being used to compare six writing stages of the short story "Orient-Express", written by Francesca Sanvitale (in Separazioni, Turin: Einaudi, 1997, 179-189). Each variant is presented with the Split Zooming technique while the collection of variants is presented with the BiSplit Zooming technique.

above the context. This version of Flip Zooming is called Split Zooming.

Most focus + context technique usually only provide one focus to the user, which is natural as humans only can focus on one thing at the time. However, when we are comparing two things we are rapidly changing focus between two different areas of information. If we have to change the presentation each time we change focus, an unnecessary sensory information load is created, as not only the focus changes, but also the context. This led us to the development of a second version of Split Zooming called BiSplit Zooming. This version was realized with two independent foci, so that the change of the layout would be minimized when comparing variant texts (see figure 2).

3.2 Searching

As mentioned above, many alternative search methods have been developed in the IRS community. These are usually based on purely textual searches, while Flip Zooming provides the means for "graphical" searches. But why not combine the two? There is no inherent incompatibility with the different search techniques so both were used in the DV Browser. The basic presentation of the material is graphical, but textual search can be invoked whenever required. Instead of showing the results of the search in a separate list or window, as is usual in IRS, the searched expression in all variant texts are highlighted in the actual pages by using boxes or colours. This makes it extremely easy to find the different hits within a single variant text or to find corresponding occurrences in two different texts. Areas which are found to be especially interesting can then be focused on by using the focus + context
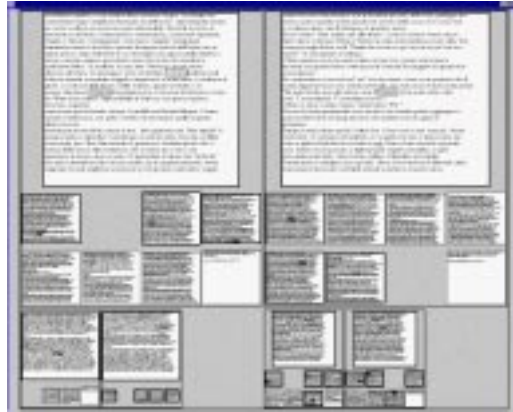
Fig. 3. The DV browser after a search for Umberto in the variants of "Orient-Express" has been performed. Each page that contains the word is shown with an extra wide frame, and the actual position of the word is marked. When used on a display that provides colours, the frame and the position of the words are marked with a distinct colour.

technique. A view of using the Digital Variant to search for the word Umberto in the short story "Orient-Express" is given in Figure 3.

3.3 Future Work

The DV Browser shows the value of providing overview when studying digital texts, especially studying different versions of the same text -- or, more in general, pieces of information that need to be compared. Nonetheless, the application is a prototype and to be fully operational in a professional setting it should be enhanced with further functionality. The system is orthogonal to most information retrieval system in the sense that it can be combined with other search techniques (e.g. vector space modelling or clustering) in order to provide more advanced support for searches or alternative views. This would be especially useful for finding places where one word has been replaced by a synonym or finding phrases or sentences that have changed between revisions.

The DV Browser has not yet been tested by a sufficient number of users, but it is our aim and hope, in the near future, to let a number of researchers evaluate the application. Though no formal evaluation has been performed, the prototype has been shown in several conferences receiving positive response. Another possibility would be to make it available as a Java applet on the DV web site, so that any Internet user can use the application.

Concluding, we believe that the DV Browser and the Flip Zooming technique can provide a better overview of complex textual traditions, and that focus + context methods can be useful for situations where large amounts of texts need to be presented and compared.

4 Acknowledgements

5   Bibliography

1. Bartram, L., Ho, A., Dill, J., Henigman, F., 'The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces', in Proceedings of ACM UIST '95 (New York: ACM Press, 1995), 207-215.

2. Bederson, B., and Hollan, J., 'Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics', in Proceedings of ACM UIST '94 (New York: ACM Press, 1994), 17-26.

3. Bederson, B. B., Hollan, J. D., Stewart, J., Rogers, D., Vick, D., Ring, L. T., Grose, E., and Forsythe, C., 'A Zooming Web Browser', in Forsythe, C., Ratner, J., and Grose, E. (eds.), Human Factors in Web Development, (Mahwah, NJ: Lawrence Erlbaum, 1998a), 255-266.

4. Bederson, B., Meyer, J., 'Implementing a Zooming User Interface: Experience Building Pad++,' in Software: Practice and Experience, vol. 28, n. 10 (1998b), 1101-1135.

5. Björk, S. and Holmquist, L. E., 'Formative Evaluation of a Focus + Context Visualisation Technique'. Poster at Human Computer Interaction '98. Available at http://www.viktoria.informatik.gu.se/play/publications/1998/formative.pdf.

6. Brown, M. H., and Weihl, W. E., 'Zippers: A Focus + Context Display of Web Pages', Proceedings of World Conference of the Web Society, Association for the Advancement of Computing in Education (AACE), (1996). Also available as DEC-SRC Tech Report 140.

7. Card, S. K., Robertson, G., and York, W., 'The WebBook and the Web Forager: An information workspace for the World Wide Web', in Proceedings of ACM SIGCHI '96 (New York: ACM Press, 1996), 111-117.

8. Card, S. K., Mackinlay, J. D., and Shneiderman, B (eds.) Readings in Information Visualization: Using Vision to Think, (San Franciso, Ca: Morgan Kaufmann Publishers 1999).

9. Fisher, B., Agelidis, M., Dill, J., Tan, P., Collaud, G., and Jones, C., 'CZWeb: Fish-eye Views for Visualizing the World Wide Web', in Smith, M., Salvendy, G. and Koubek, R. J. (eds.), Design of Computing Systems: Social and Ergonomics Considerations (Amsterdam: Elsevier, 1997), 719-722.

10. Furnas, G. W., 'Generalized Fisheye Views', in Proceedings of ACM SIGCHI '86 (New York: ACM Press, 1986), 16-23.

11. Furnas, G. W., 'Effective View Navigation', in Proceedings of ACM SIGCHI '97 (New York: ACM Press, 1997), 367-374.

12. Hollands, J. G., Carey, T. T., Matthews, M. L., and McCann, C. A., 'Presenting a graphical network: a comparison of performance using fisheye and scrolling views', in Salvendy, G. and Smith, M. (eds.), Designing and Using Human-Computer Interfaces and Knowledge Based Systems (Amsterdam: Elsevier, 1989), 313-320.

13. Holmquist, L. E., 'The Zoom Browser: Showing Simultaneous Detail and Overviwe in Large Documents'. in Human IT, (3), vol. 2, n. 1, (Borås, Sweden: ITH, 1998), 131-150.

14. Holmquist, L. E., and Björk, S., 'A Hierarchical Focus + Context Method for Image Browsing', in Proceedings of SIGGRAPH '98 (New York: ACM Press, 1998), 282.

15. Holmquist, L. E., and Björk, S., 'Formative Evaluation of Flip Zooming: Towards Effective Integration of Detail and Context on Small Displays', in Viktoria Research Report VRR-98-1, Gotenburgh: Viktoria Institute. Internet: http://www.viktoria.informatics.gu.se/publications/.

16. Kandogan, E., and Shneiderman, B., 'Elastic Windows: Evaluation of Multi-Window Operations', in Proceedings of ACM SIGCHI '97 (New York: ACM Press, 1997), 250-257.

17. Kowalski, G., Information Retrieval Systems - Theory and Implementation (London: Kluwer, 1996).

18. Lamping, J., Rao, R., and Pirolli, P., 'A focus+context technique based on hyperbolic geometry for viewing large hierarchies', in Proceedings of ACM SIGCHI '95 (New York: ACM Press, 1995), 401-408.

19. Leung, Y. K. and Apperley, M. D, 'A Review and Taxonomy of Distortion-Oriented Presentation Techniques', in ACM Transactions on Computer-Human Interaction, vol. 1, n. 2 (New York: ACM Press, 1994), 126-160.

20. Mackinlay, J. D., Robertson, G. G., and Card, S. K, 'The Perspective Wall: Detail and Context Smoothly Integrated', in Proceedings of ACM SIGCHI '91 (New York: ACM Press, 1991), 173-179.

21. Robertson, G. G., and Mackinlay, J. D., 'The Document Lens', in Proceedings of ACM UIST '93 (New York: ACM Press, 1993), 101-108.

22. Salton, G., Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer (Reading, Ma: Addison-Wesley, 1989).

23. Sarkar, M., Brown, M. H, 'Graphical Fisheye Views', Communications of the ACM, vol. 37, no. 12 (New York: ACM Press, 1994), 73-84.

24. Sarkar M., Snibbe, S. S., Tversky, O. J., and Reiss, S. P., 'Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens', in Proceedings of ACM UIST '93 (New York: ACM Press, 1993), 81-91.

25. Schaffer, D., Zuo, Z., Greenberg, S., Bartram, S., Dill, J., Dubs, S., and Roseman, M., 'Navigating hierarchically clustered networks through fisheye and full-zoom methods', in ACM Transactions on Computer-Human Interaction, vol. 3, n. 2 (New York: ACM Press, 1996), 162-188.

26. Spence, R., Apperley, M., Data base navigation: an office environment for the professional, in Behavior and Information Technology, vol. 1, n. 1 (1982), 43-54.

27. Wistrand, E., 'Creating Image Context using ImageTrees', in ACM SIGCHI '96 Conference Companion (New York: ACM Press, 1996), 281-282.

5.1 Web pages

28. Digital Variants Project: http://www.ed.ac.uk/~esit04/digitalv.htm. Last visited 2000-05-17.

29. The PLAY studio: http://www.viktoria.informatik.gu.se/groups/play. Last visited 2000-05-17.

30. The Viktoria Institute: http://www.viktoria.informatik.gu.se. Last visited 2000-05-17.

31. "Computers, Literature and Philology: An International seminar": http://www.ed.ac.uk/~esit04/seminar.htm. Last visited 2000-05-17.

Hierarchical Flip Zooming:
Enabling Parallel Exploration of Hierarchical Visualizations

Staffan Björk

Abstract. This paper describes hierarchical Flip Zooming, a focus+context visualization technique for hierarchical information sets. It allows for independent focus+context views at each node of the hierarchy and enables parallel exploration of different branches of the hierarchy. Visualization, navigation and interaction in the Flip Zooming technique is described as well as how the technique fits into existing models of information visualization. Examples of applications using the technique are given.

Keywords. Information Visualization, Focus+Context Visualization, Hierarchies, Flip Zooming

1   Introduction

Information visualization has been acknowledged as a powerful way of presenting information on computer displays. To enable larger amounts of information to be presented, many techniques use hierarchies, either inherent hierarchical structures or by superimposing new categories on them. The hierarchies are not only use to structure the visualization, but also to filter out objects, i.e. not show objects outside the branch being explored, or to enable abstractions or summarizations based on the structure.

When exploring such hierarchies, the user is usually limited to choosing one branch to view. The other branches are distorted, removed, occulted, or compressed to a more abstract representation in order to give the chosen branch more room. These techniques work well when the user knows exactly what to look for, and can easily navigate within the visualization without getting lost. However, in situations where a

user is looking for several objects to be viewed simultaneously, a visualization should enable interaction with, and viewing of, many different levels of the hierarchy at once. These kinds of views are also important if a user needs to see the information as structured by some other form of classification than the one provided, i.e. transforming the appearance of the information structure on the fly. Such a view would allow users to locate several items in parallel which are located deep in different branches of the information structure, and allow users to compare such items to each other as they could be shown clearly together. To do this, the visualization needs to give the user control over how to visualize the hierarchy at each level and at each node of the information structure.

This paper presents such a visualization, the *Hierarchical Flip Zooming* focus+context visualization technique. Following a background of information visualization techniques, the original non-hierarchical Flip Zooming technique is described, followed by how the hierarchical Flip Zooming visualization is created by using nestled instantiations of the non-hierarchical visualization. To illustrate the method, examples of applications developed using the hierarchical Flip Zooming visualization are presented. Finally, some conclusions and areas of future work are given.

2 Background

Hierarchical structures have been used since the very beginning of information visualization on computers. In the *FISHEYE view* [6,7], Furnas used the hierarchical structures of computer programs and scientific articles to create a structured view of the material. In these views, nodes higher up in the structure, representing more abstract notions, such as definitions of functions or chapter headings, were given higher priority than nodes lower in the hierarchy, e.g. variable declarations or subheadings. By using the overall priorities of these nodes and their proximity to a maneuverable cursor, the system could generate a view in which the information near the focus was presented clearly while at the same time show the overall structure of the information. This type of visualization is called a *focus+context visualization*.

Focus+context has been the basis for many visualization techniques. Examples include the *Bifocal Display* [26,27], the *Document Lens* [21],

the *Perspective Wall* [18], the *Rubbersheet View* [23], and the *Table Lens* [19]. Sarkar and Brown [22] used Furnas' idea of fisheye views to create presentations of graphs using geometrical transformations. Schaffer et al. [24] transformed connected graphs into hierarchical clusters in order to visualize them using fisheye and full-zoom visualizations. [17] provided a unified theory for focus+context techniques using a rubbersheet analogy. Other ways of classifying the techniques have used *Space-scale diagrams* [9], *Non-Linear Magnification Fields* [14,15], or multiple dimensions of transformation [25]. [3] gives a framework for describing focus+context visualizations that allow for different techniques to be nestled hierarchically.

2.1 Connection and Enclosure

Card et al. [4] describe two ways of how visualizations present hierarchical data: by *Connection* or by *Enclosure*. Connection uses visual links between parents and children in the structure, while enclosure visualizes children within the parents. Connection is often used when there exists information both in the nodes and the leaves of the structure, e.g. binary search trees and genealogical trees, while enclosure is used when most or all information is found in the leaves, e.g. file structures on computer systems. Connection is not usually as space efficient as enclosure, but can present information in nodes and leaves in the same way.

Examples of focus+context visualizations using connection include *Cone Trees* [20], which present wide hierarchical structures, and the *Hyperbolic Tree* [16], which presents hierarchies in a hyperbolic space. Visualizations that use enclosure include the *Continuous Zoom* [1] and *Treemaps* [12].

2.2 Interaction and Navigation

In most visualization techniques, interaction with the visualization is done either on a global level, changing the viewer's point of view, or moving a focus that changes the overall presentation. However, few techniques allow for two or more foci, i.e. they do not let the user have several independent points of interaction where the user can change the

visualization. The notion of such polyfocal views actually predates interactive information visualizations [13], but have yet to be incorporated into visualization techniques. The rubbersheet view [23] is an exception to this, allowing several different parts of a rubbersheet to be deformed.

Furnas [8] proposed requirements to compare view traversibility and navigability efficiency between different information structures. Although he assumed that only a small subset of any structure can be visualized at any given time, his model of view transversal and view navigation is applicable to information visualization in general. One of the points Furnas makes is to differentiate between the logical structure graph of an information structure and the viewing graph of it. When doing this, Furnas makes implicit use of a cursor that links the logical structure and the viewing graph, so that there always exists one node in the viewing graph that corresponds to a node in the logical structure graph, i.e. every state of the viewing graph has one element of the logical structure in focus.

3   Flip Zooming

The first Flip Zooming application was the Zoom Browser [10]. This browser presented several web pages, each divided into a number of tiles, in the same display area. After trying the visualization technique used in the application on other information types, and performing formative evaluations, a generalized technique was developed. In the following, the generalized non-hierarchical Flip Zooming technique is described. This provides a basis for the description of the hierarchical Flip Zooming visualization, as it is created by recursively using the basic Flip Zooming technique to visualize each level.

3.1   The Flip Zooming technique

Flip Zooming belongs to the focus+context group of information visualizations. It visualizes information that consists of a number of distinct objects with a sequential order. Each object is presented in a rectangular area called a *tile* with one tile designated as the focus. This focus is placed in the center of the display area with the other tiles placed around

it, giving the focus significantly more of the screen space available (see figure 1 on page 6).

Unless the information to be visualized already consists of suitably distinct objects, it must be transformed into discrete objects. This can be compared to printing a text. Unless the text is short enough to be printed on a single page, the text must be structured into a number of distinct sections, each suitable to print on a single page. Similarly, Flip Zooming requires the information to be split into pieces suitable to be presented in the area given to a tile. In the case where the information consists of a number of distinct objects but no clear sequential order exists, one must be superimposed on the information. One example of this is a collection of unrelated image files, which can be given a sequential order by using the alphanumerical ordering of the file names.

Every focus+context technique must provide a way of showing information in the presentation in a more compact form than that of the focus in order to make it fit the context area. Card et al. [4] defines five such ways of reducing information: *filtering* away the least significant information, creating fewer case by *selective aggregation*, using so-called *micro-macro readings* where larger coherent structures arise from the details, *highlighting* of the most important information, and *distortion* of the graphical presentation of the information. Flip Zooming uses a form of distortion, a simple perspective scaling function that presents context tiles as focus tiles viewed from a distance.

Unlike most distortion techniques (c.f. [18,21]), Flip Zooming presents both focus and context objects as seen from straight ahead. This makes the recognition of any given object easier as only the location and size of the object can change. Further, the linear scaling solution differs from most other distortion techniques in that it does not hide or remove objects.
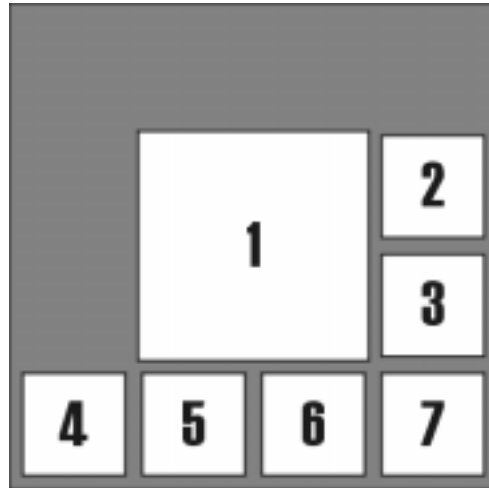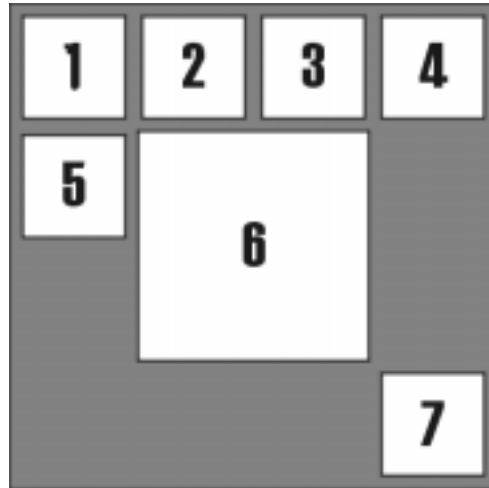
Fig. 1. Flip Zooming layout with focus on the first object.

3.2 Placement of Tiles

The focus tile in a Flip Zooming visualization is placed in the center of the display area since that area is a natural focal point. The context tiles are placed around it so that the sequential order of the data set is preserved. This is done by having tiles sequentially before the focus tile above and to the left of the focus, starting in the left top corner. Tiles after the focus are placed below and to the right of the focus, starting in the right lower corner (see figure 1). Using the simile of text, the tiles before the focus tile are left justified and the tiles after the focus are right justified. The placement of focus and context tiles thus allows for a left-to-right, top-to-bottom, reading.

The central placement of the focus makes the placement of context tiles unbalanced (except when the median tile of the set is the focus tile). This leaves certain areas of the display area unused but gives a strong spatial indication of where in the sequence the focus tile is, e.g. if there are many tiles below and to the right of the focus, the focus tile is early in the sequence. The placement strategy also limits the number of places where a tile is placed. Due to left and right justification of the context

Fig. 2. Flip Zooming layout with focus on the sixth object.

tiles, any given tile has only three places in the display area: one when the focus is on a tile *after* the tile is question, one when the focus is on a tile *before* the tile, and one when the tile *is* the focus tile. This makes retrieval of tiles earlier viewed easier, as spatial memory can be used to find tiles, regardless of the current state of the visualization.

Whenever the focus is changed, the previous focus is changed into a context tile, i.e. scaled to a smaller size, and moved to its appropriate place. The new focus tile is then moved to the focus position and shown in increased detail. All (context) tiles between the new focus and the old focus are moved from one side of the focus position to the other in order to maintain the sequential order of the data (see figures 1 and 2). As these tiles are already shown as context tiles, their appearances are unchanged.

The tiles in a visualization are not packed together side by side. Although this would maximize the screen space used, this is avoided to keep a clear separation between the different information objects. Further, the relation between the locations of any two tiles is not fully rigid. A tile that is before another tile in the sequence will be above or to the left, or both, of the latter tile in the presentation. Which of these possibilities are true changes as the visualization is transformed. Therefore, even in the case when the information presented has been transformed

from one whole piece to several small, it would not be advantageous to pack the tiles together without space between them.

3.3 Navigation

The Flip Zooming technique allows for both moving the focus sequentially through the tiles and choosing the focus by random access. Any tile can be chosen as focus by positioning and selecting a tile with an input device such as a mouse or a touch sensitive screen. Moving a cursor (shown as a box around a tile) forward and backward in the presentation allows the users to sequentially explore the data set. A tile does not automatically gain the focus when a cursor is placed on it. To do this, the user selects the tile marked by the cursor using a select operator.

The cursor was introduced into the system to avoid unnecessary changes of focus, allowing the change of focus between tiles widely separated without requiring every tile between them to briefly become a focus. Not only does this avoid putting unwanted tiles in focus, it lowers the cognitive overhead as unnecessary changes in the visualization are avoided. If the number of tiles in the presentation is small, the cursor can be removed, as it may create more cognitive overhead than stepping through a small number of foci tiles would. The sequential navigation allows the whole visualization to be explored by using only three operators; two if no cursor is used. This makes the visualization feasible to use on devices that have no pointing devices, or in situations where one does not want to use up many of the input operators available.

The Flip Zooming technique allows for a further level of detail of the focus tile. In the *full-focus view*, the user is shown only the focus, thus providing a detailed presentation of any single tile when the user deems detail more import than context, or when the context is unnecessary. Using Card et al. classification of techniques to reduce information [4], the full-focus view can be seen either as filtering (removing the context tiles) or distorting (occluding the other tiles). The full-focus view is invoked using direct access devices in the same fashion as a context tile is made into the focus tile, i.e. clicking on the tile, but in this case the focus tile is clicked upon. Using the navigational operators, the full-focus view is accessed by placing the cursor on the focus and then using the select operator. In the full-focus view, the cursor (if used at all) is not shown, and sequential movement simply changes the focus without

showing any context tiles. All the actions that invoke the full-focus view are also used to deselect the full-focus view. If the cursor is being used, it is placed on the focus tile when the normal focus+context view is used again.

By mapping the select operator to activate the full-focus view when the cursor is on the focus tile, the select operator is defined for all possible locations of the cursor. Thus, all operators used for sequential movement can be used at any given time and a user does not have to deal with different functional states of the visualization. Using the terms introduced by Furnas [8] and mentioned in the background section, each node in the viewing graph of a Flip Zooming visualization has one outgoing link for each operator. Using only the three navigational operators, this meets Furnas' requirement of having a small number of outgoing links at each node. It does not fit his requirement of having a short distance between any two nodes in the graph. However, in the case of random access input, the situation is opposite: the distance is always one but the user can choose any node at any time. By combining the two input modes depending on information and task, the technique meets Furnas' requirements of efficient view traversibility.

Unlike the examples given by Furnas, Flip Zooming allows the user to manipulate the state of the focus, i.e. change between normal focus view and full-focus view. This means that the viewing graph of a Flip Zooming visualization has two nodes for each node in the information structure. Further separating the viewing graph of a Flip Zooming visualization from other viewing graphs is the fact that the cursor of the visualization can be moved without changing the state of the presentation. Only when the select operator is used is the focus changed to where to cursor is. By adding a token to the viewing graph that shows the focus of the presentation, Furnas system can be expanded to incorporate Flip Zooming visualizations.

3.4   Interaction with Tile Objects

Flip Zooming was designed to allow every tile to be an interactive object, allowing the user to both give input and receive output from each individual tile. Thus, each tile must provide methods for presenting its information in a correct way, and how to changes its appearance due to input. In this way, each tile can be seen as an application in a window-

based graphical user interface. In order to let the tiles receive input, all direct-access input (mouse input etc.) is sent to the tile upon which the action occurred and all keyboard input, except navigational commands, is sent to the tile that the cursor is on (the focus in the case of a cursorless system). This means that Flip Zooming visualizations actually support two different kinds of foci, one visual focus providing a detail view of information and one input focus allowing keyboard input. In this paper, the term focus refers to the visual focus.

For random access navigation to work, Flip Zooming imposes two restrictions on the functionality and appearance of a tile. First, the tile must send any unused mouse commands, e.g. mouse clicks on un-clickable areas, back to the visualization. Second, the tile must have areas that clearly are unused for interaction with *that* tile, or have areas with the affordances of focusing the tile. The two restrictions together ensure that there always are areas on any given tile to click on to focus the tile.

The restrictions on tiles are similar to the restrictions on windows in window systems. These must have title bars so that the windows can be moved by dragging, and borders for manipulating the size of the windows. Tiles in Flip Zooming have borders, but these are passive as their only function is to show the boundary. Also, instead of always having a title bar at the top of its area, a tile can have the area which allows the tile to be focused where it is most suited depending on what information is shown on the tile.
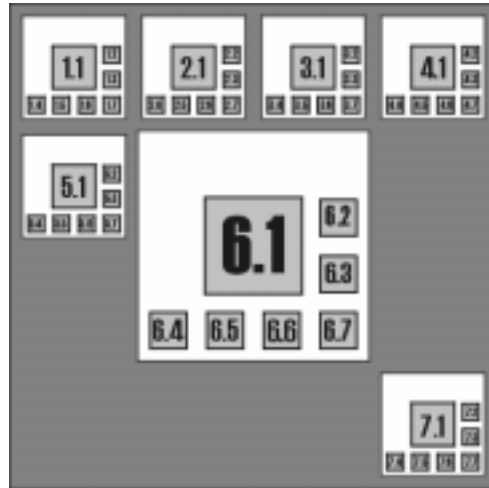
4   Hierarchical Flip Zooming

As mentioned above, tiles in a Flip Zooming visualization can be interactive objects. One example of such interactive objects are Flip Zooming visualizations. By using Flip Zooming visualizations as tiles within another Flip Zooming visualization (see figure 3 and 4 on pages 12 and 13, respectively), the visualization as a whole becomes a two-leveled visualization. The individual visualizations can be interacted with independently and their presentation of information is self-governed with the exception that the outer visualization decides the size and location of the inner visualizations. Even when an inner visualization is not the focus tile of the outer visualization, a user can change its focus. Further, by using the two different levels of full-focus views available, the whole

visualization can be made to show only one inner visualization or to only show one tile in one of the inner visualizations, making the whole visualization look as if it only had one level of tiles. This hierarchical visualization, called Hierarchical Flip Zooming, can be of any desired depth and can have different depths in different parts of the visualization.

Hierarchical Flip Zooming belongs to the group of hierarchical visualizations using enclosure as every child of a node in the information structure is represented by a tile and is shown in a visualization representing the parent node. Unlike most other techniques using enclosure (c.f. [1,12]), Hierarchical Flip Zooming is not space-filling, i.e. it does not use all display space available to present the leaves of the information structure. Besides being used to clarify the distinctness of each tile, this "unused" space is needed for navigation of visualizations that are nestled within other visualizations. As in the case of interaction with tiles in the basic Flip Zooming technique, every tile must have areas unused for interaction with that tile so that the tile is focusable by random-access methods. The empty space in an inner visualization provides such unused areas and allows the inner visualization to be set as focus in the outer visualization. As with other information visualizations that use enclosures to visualize hierarchies (e.g. [12]), Flip Zooming is best used with information structures that have little or no information in the nodes.

When enough Flip Zooming visualizations are used recursively, the innermost visualization will have too little space to display tiles in a comprehendible way. This is similar to the breaking point problem existing in all information visualizations techniques. Due to the modularity of Flip Zooming, this problem can be address in two fashions. First, the visualization may show the individual tiles so that they are unintelligible but the structure of the tiles is visible, to maintain a coherent presentation throughout the system. A second alternative is to use an aggregated view, i.e. show some kind of summary or example of the tiles. What solution to use can be dictated either by user preference or by the type of information to be presented.
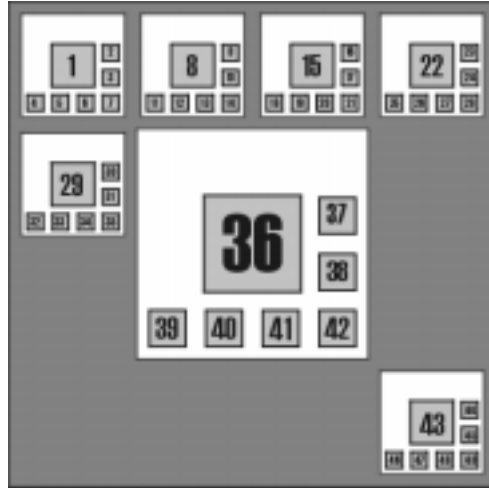
Fig. 3. Hierarchical Flip Zooming of hierarchical information.

4.1   Structuring objects in the Hierarchy

In order to use a hierarchical information visualization, the information to be presented must be in an hierarchical structure. If the information already is an hierarchy, the translation to the visualization becomes trivial. In the case of Flip Zooming, each node in the hierarchy becomes an Flip Zooming visualization tile placed in its appropriate place while each leaf in the hierarchy become a "basic" tile (see figure 3).

When the information to be presented is not a hierarchical structure, some transformation must be performed similar to the transformation required for non-sequential information to be presented by the non-hierarchical Flip Zooming visualization. Instead of just separating the information into a number of categories, the information can also be placed in subcategories. In the example of printing a text, a non-hierarchical division was to fit a suitable amount of text onto each page. When using hierarchies in the division, these pages are still created, but they can be collected into larger groups representing chapters. The transformation of sequential information into a hierarchical structure can be seen as a form of folding (see figure 4).

Fig. 4. Hierarchical Flip Zooming of sequential information.

4.2 Navigation

The independence of each individual visualization within a hierarchical Flip Zooming visualization means that navigation can be done at several different levels at once. Every visualization in the presentation has a focus which can be changed independently of where in the hierarchical structure it is, and visualizations within other visualization can be manipulated regardless if it is a focus or not. One benefit of this is that several inner visualizations can have full-focus views, allowing their focus tiles to be compared to each other even though they are in different branches of the hierarchy. To allow navigation of the whole visualization, two new operators were introduced. These operators, up and down, allow for the cursor to be moved from a tile containing a visualization to the focus of that visualization and vice verse.

   The freedom of manipulating individual visualizations enables the user to decide what objects are interesting as context objects. However, these chosen context objects are always presented in a predetermined way. This allows the user to influence what kind of context the visualization should present while at the same time making it impossible for
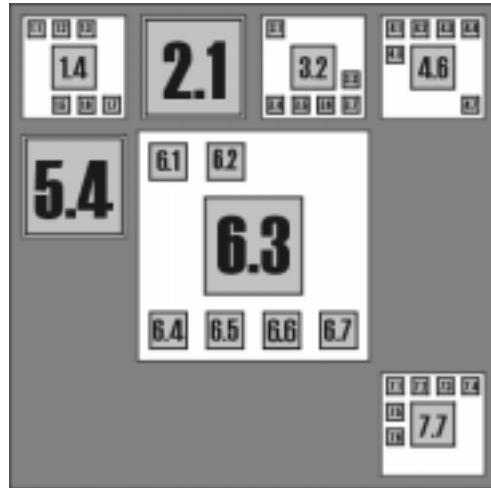
Fig. 5. The Hierarchical Flip Zooming visualization after some user interaction.

the user to break the ordering of the information (see figure 5). This allows other users, who have previously seen the information, to quickly recognize it and identify what parts constitute the context to the focus.

The viewing graph of a hierarchical Flip Zooming visualization becomes complex. Each level of the visualization is a complete visualization, with its own viewing graph and focus token. However, these views are connected as the cursor can transverse between different visualizations. Naïvely connecting these graphs with links between nodes does not work. Looking at the visual aspects of the graph, the focus token of an outer visualization will have to mark several nodes at once, representing an inner visualization, when that inner visualization is the focus of the outer visualization. In addition, the shared cursor changes focus in the visualization in which it is currently located but does not change the state of any of the other visualizations. Functionally, the viewing graph changes appearance as the visualization changes appearance. For instance, full-focus views used in outer visualizations may hide inner visualizations so that they cannot be manipulated. Representing this in the viewing graph would equal to removing links dynamically from the viewing graph as the user interacted with the

visualization. Similarly, when inner visualizations are made visible, new links should be created.

One possible way of creating a viewing graph would be to make one node for each possible state of the system as a whole. However, this would make the viewing graph grow exponentially as the hierarchies in the information structure increases. As the states of the inner visualizations are practically independent of the outer visualizations, a clearer representation may be achieved by representing an inner visualization as a node in the viewing graph of the outer. Thus, the viewing graph of a hierarchical Flip Zooming visualization may be better visualized using an enclosure technique, or in other words, by using the hierarchical presentation technique used by the visualization itself.

Regardless of exactly how the viewing graph is constructed, Hierarchical Flip Zooming meets the requirements of efficient view navigation better than the basic Flip Zooming technique. The improvement is due to the shorter path possible when using only the navigational operators. Instead of being linear to the size of the information structure as the basic technique, the distance is logarithmic as the user navigates a tree structure.

5   Example Applications

The possibility of using Flip Zooming to provide hierarchical visualizations has been explored in a number of applications. Below, two examples of these are given.
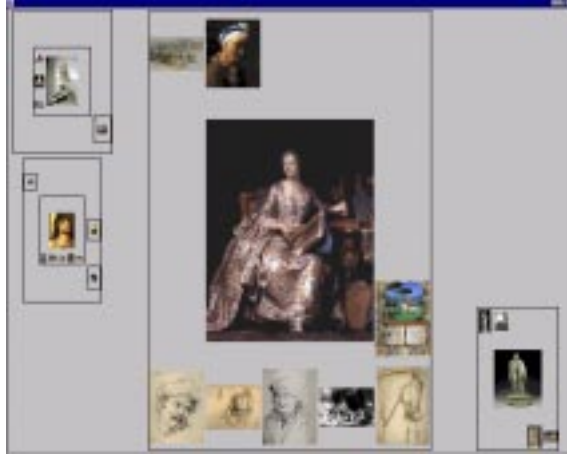
Fig. 6. The Hierarchical Image Browser.

5.1 Hierarchical Image Browser

In the Hierarchical Image Browser [11], the Flip Zooming technique was used to present images of paintings and sculptures. The images were placed in a Flip Zooming visualization according to their style, and the visualizations themselves were grouped in outer Flip Zooming visualizations representing the different sections of a museum (see figure 6). The ability to select foci in each of the visualizations allowed very small images to be enlarged to any desired size. This allows comparisons of different styles or schools of art to be easily performed by enlarging images from different parts of the visualization. The application also provided the additional functionality for users to create their own grouping of images by using cut-and-paste operations.

Fig. 7. The Digital Variants prototype showing six variants of "The Orient-Express" by Italian writer Francesca Sanvitale.

5.2 Digital Variants

This prototype [2] was developed to support literature research on variants of the same texts, e.g. to compare translations, study a text's development or classify the lineage of texts where this is unknown. Each of the texts was transformed into a Flip Zooming visualization by dividing the text into a number of tiles comparable to a page in a book (for longer texts further divisions into chapters could be performed). These presentations were then placed in an outer Flip Zooming visualization.

As the most important functionality of the prototype was to allow comparison between texts, two foci were used on the top level of the visualization. These two foci were placed at the top of the presentation area (see figure 7). Although this moved the foci so that they were not fully surrounded by the context, this enabled the sizes of the two foci to be equal at all times and meant that a user's line-of-vision would not have to pass through other objects when switching between the two foci. The prototype supported user annotations and text searches where results were color-coded.

6 Conclusions

In this paper, the hierarchical Flip Zooming technique, as well as the basic non-hierarchical Flip Zooming technique it uses, has been presented. The basic technique allows for a Focus+Context visualization of a sequentially ordered set of data. By using the visualization recursively, a hierarchical Flip Zooming technique is created which presents hierarchical information structures where each branch of the structure can be explored simultaneously. Such visualizations enable comparison of leaves in the hierarchical structure even when these are widely separated. This is achieved while still maintaining a sequential ordering, left-to-right, top-to-bottom, on every level of the hierarchy.

By giving examples, the technique has been shown to be applicable in diverse application fields. Future work with the hierarchical Flip Zooming technique include looking at the problem of using the technique on small screens, using the technique as the basis for a windowing system, and performing detailed user studies.

7 Acknowledgements

8 REFERENCES

1. Bartram, L., et al. The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces, in *Proc. of UIST '95*, pp. 207-215, ACM Press, 1995.

2. Björk, S. and Holmquist, L.E. The Digital Variants Browser: An explorative tool for literature studies. In *Proc. of Computers, Literature and Philology*, Edinburgh, UK, 1998.

3. Björk, S., et al. A Framework for Focus+Context Visualizations. In *Proc. of IEEE Information Visualization '99*.

4. Card, S.K., et al. (Eds.) *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, California, pp. 1-34, 1999.

5. Carpendale, M.S.T., et al. Extending Distortion Viewing from 2D to 3D. In *IEEE Computer Graphics and Applications*, July-August, 1997.

6. Furnas, G.W. The FISHEYE View: A New Look at Structured Files. Bellcore Technical Report, 1981. Reprinted in: Card, S.K., et al. *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

7. Furnas, G.W. Generalized Fisheye Views, In *Proc. of CHI '86*, pp. 16-23, ACM Press, 1986.

8. Furnas, G.W. Effective View Navigation. In *Proc. of CHI '97*, pp. 367-374, ACM Press, 1997.

9. Furnas, G.W. and Bederson, B.B. Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proc. of CHI '95*, pp. 234-241, ACM Press, 1995.

10. Holmquist, L.E. The Zoom Browser: Showing Simultaneous Detail and Overview in Large Documents. In *Human IT*, Vol. 2 No. 3, ITH, Borås, Sweden, 1998.

11. Holmquist, L.E. and Björk, S. A Hierarchical Focus + Context Method for Image Browsing. In *SIGGRAPH '98 Sketches and Applications*, ACM Press, 1998.

12. Johnson, B., and Shneiderman, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proc. of IEEE Visualization '91*, pp. 284-291, IEEE, New York, 1991. Reprinted in: Card, S.K., et al. *Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

13. Kadmon, N., and Shlomi, E. A polyfocal projection for statistical surfaces. In *Cartograph*, J. 15, no. 1, pp. 36-40, 1978.

14. Keahey, T. and Robertson, E.L. Non-Linear Magnification Fields. In *Proc. of IEEE Visualization '97*, Information Visualization Symposium, IEEE Press, 1997.

15. Keahey, T. The Generalized Detail-In-Context Problem. In *Proc. of IEEE Visualization '98*, Information Visualization Symposium, IEEE Press, 1998.

16. Lamping, J., et al. A focus+context technique based on hyperbolic geometry for viewing large hierarchies. In *Proc. of CHI '95*, ACM Press, 1995.

17. Leung, Y.K, Apperley, M.D. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. In *ACM Trans. on Computer-Human Interaction*, Vol. 1, No. 2, pp. 126-160, 1994.

18. Mackinlay, J. D., et al. The Perspective Wall: Detail and Context Smoothly Integrated. In *Proc. of CHI '91*, pp. 173-179, ACM Press, 1991.

19. Rao, R. and Card, S.K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In *Proc. of CHI '94*, ACM Press, 1994.

20. Robertson, G.G., et al. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proc. of CHI '91*, ACM Press, 1991.

21. Robertson, G.G., Mackinlay, J.D. The Document Lens. In *Proc. of UIST '93*, pp. 101-108, ACM Press, 1993.

22. Sarkar, M. and Brown, M.H. Graphical Fisheye Views. In *Comm. of the ACM*, Vol. 37, No. 12, pp. 73-84, 1994.

23. Sarkar M., et al. Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In *Proc. of ACM UIST '93*, pp. 81-91, ACM Press, 1993.

24. Schaffer, D., et al. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. In *ACM Trans. on Computer-Human Interaction*, Vol. 3, No. 2, June 1996, pp. 162-188.

25. Spence, R. A taxonomy of graphical presentation. In *INTERACT '93 and CHI '93 conference companion*, pp. 113-114, ACM Press, 1993.

26. Spence, R., New concepts in Visual Displays. In *Proc. of Vision and Visual Display Units*, Institute of Opthalmology, London, pp. 6-1 to 6-9, 1981.

27. Spence, R., Apperley, M., Data base navigation: an office environment for the professional. In *Behavior and Information Technology*, Vol. 1 No. 1, pp. 43-54, 1982.

28. Tweedie, L. Characterizing Interactive Externalizations. In *Proc. of CHI '97*, ACM Press, 1997.

WEST: A Web Browser for Small Terminals

Staffan Björk, Lars Erik Holmquist and Johan Redström, *PLAY*

Ivan Bretan, *Telia Mobile*

Rolf Danielsson, *Telia Research*

Jussi Karlgren and Kristofer Franzén,
Swedish Institute for Computer Science

Abstract. We describe **WEST,** a **WE**b browser for **S**mall **T**erminals, that aims to solve some of the problems associated with accessing web pages on hand-held devices. Through a novel combination of text reduction and focus+context visualization, users can access web pages from a very limited display environment, since the system will provide an overview of the contents of a web page even when it is too large to be displayed in its entirety. To make maximum use of the limited resources available on a typical hand-held terminal, much of the most demanding work is done by a proxy server, allowing the terminal to concentrate on the task of providing responsive user interaction. The system makes use of some interaction concepts reminiscent of those defined in the Wireless Application Protocol (WAP), making it possible to utilize the techniques described here for WAP-compliant devices and services that may become available in the near future.

Keywords. Hand-held devices, web browser, proxy systems, focus+context visualization, text reduction, flip zooming, WAP (wireless application protocol)

1   Introduction

The World Wide Web (WWW) currently consists of about half a billion pages, offering users a vast range of informational resources. However, these pages are almost exclusively designed for use with desktop computers, i.e. computers with large high resolution screens, powerful processors, and an abundance of primary and secondary storage.

Parallel to exponential growth of the web during the last few years, digital mobile telephony has evolved to become a basic commodity in the United States and many parts of Europe. Particularly in the Nordic countries, penetration can be as high as 60% (Finland). The world's largest manufacturers of mobile phones predict that there will be 1 billion mobile telephones in use in five years time. Currently, mobile communication is still mostly synonymous with voice telephony, but this is almost certain to change pending new mobile data communication technologies being deployed, increasing data speeds and improving usability. In particular, this development should be viewed in the context of network technologies such as GPRS (General Packet Radio Service), allowing for data speeds in the range of 115 kbps and service technologies such as WAP (Wireless Application Protocol) which sets an industry standard for web-like, interactive applications for use with mobile telephones.

The work presented here focuses on this encounter between the WWW and mobile telephony, and more specifically on the need to provide gateways between mobile technologies and existing web resources. Although mobile terminals require specially designed formats for optimal usability due to the constraints of the user environment, it is not likely that all information available on the web will be translated into these format in advance. Thus, there is need for some kind of automatic on-the-fly transformation of existing web content to mobile formats, in order not to shut mobile users out from the bulk of web resources.

In dealing with this issue, the crucial problem is not as much a lack of bandwidth (which the new network technologies are dealing with) or the conversion from one mark-up language to another, but rather developing techniques for the adaptation of information to the usability requirements of mobile terminals. Innovative use of techniques for information filtering and information visualization seem to be a fruitful approach in dealing with this problem, as such techniques deal with issues that are part of the problem of providing information on small mobile devices.

New ways are needed to present web resources and to navigate among and within web pages, which is the target domain of the work described here. The constraints on information presentation posed by small terminals made it necessary to combine several different strategies in order to achieve a sufficiently compact presentation. In different fields of research, several techniques for creating compact representations have been developed. In WEST, techniques from computational
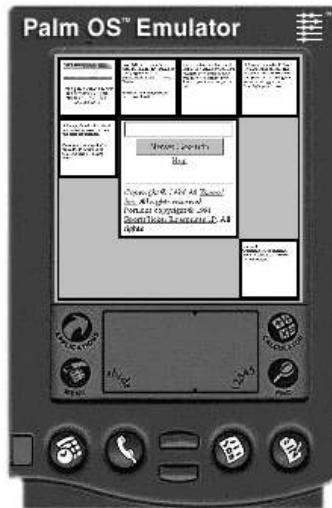
Fig. 1. The WEST browser on a simulated
Palm OS[a] display

linguistics and information visualization were combined. The original
web-pages were compressed both in terms of their linguistic content by
means of text reduction, and in terms of their visual presentation, and
were then presented to the user by means of focus+context visualiza-
tion.

The rest of the paper is organized as follows: First we give a brief
overview of the WEST system and its components. We then give a
background in related work, required for implementation of the system.
A detailed example, where we see how a user may interact with the sys-
tem follows. We then describe each of the components of the system in
further detail, and give an account of an early user test of the system.
Finally, conclusions and future work are discussed.

2 The WEST Browser

WEST (**WE**b browser for **S**mall **T**erminals) is a web browser specifi-
cally designed for use on hand-held devices with limited resources (see
Figure 1). Although most of the actual implementation was done in

Java running on a standard PC, in order to achieve a realistic simulation of the conditions of mobile computing we based the design of WEST on the capabilities and limitations of popular PDAs, such as the 3Com Palm™ line of hand-held computers. Such a device would typically have a small touch-sensitive black-and-white or grey-scale screen with a resolution of about 160x160 pixels, a memory of about 0.5-4 MB, a processor running at about 10-20 MHz, and no provision for traditional keyboard input. These capabilities have proven to be quite adequate for the tasks which such devices are currently required to perform, but are of course far below the specifications of any current desktop computer. The challenge was to work within these limitations but still provide the user with a workable browsing experience, and in the process attempt to overcome the navigation problems that would typically occur on a small terminal.

Our solution consisted of two parts:

- A *proxy server* (running on the user's ISP server), that would take a standard HTML page and transform it in real-time into a format suitable for browsing on small screens
- A *client application* (running on the hand-held terminal), that would allow the user to view and interact with the web pages as provided by the proxy server

The reason for letting the bulk of the processing of the HTML pages be done on the proxy server rather than by the terminal was to relieve the comparatively under-equipped terminal of resource-intensive tasks, thus allowing it to concentrate its resources towards providing responsive user interaction. Furthermore, by stripping away unwanted information on the server rather than on the client, a saving on bandwidth might also be made.

The proxy processing was comprised of several stages:

- A *chunking stage,* where an HTML page was divided into a number of smaller pages, or *cards*, which were then collected into groupings, or *decks*
- *A text reduction stage,* where a set of keywords summarizing each card were extracted from the text
- A *link extraction stage,* where all the hyper-links on each card were extracted

The resulting cards, with supporting keywords and links, were then passed to the client. The client application would then provide the following display modes:

- Thumbnail view: Here, a focus+context visualization comprising miniature views of the cards (or top-most card of each deck) was provided
- *Keyword view:* Here, rather than presenting thumbnails, the keywords extracted from each card were presented
- *Link view:* Similar to the keyword view, but rather than displaying keywords, this view showed the links available on each card

(A *pure text view,* showing only the text with no images or formatting, was not included in this prototype but could be useful in some situations and might be added later.)

Each view allowed the user to zoom in completely on a card, providing a fully readable view of the content. The user interacted with the views using the *flip zooming* focus+context visualization technique [16], through which the system provided an overview of the material with simultaneous access to the individual cards.

3 Related Work

In designing the system used in the WEST prototype, previous work from several different research areas were applied: Proxy systems to provide intermediate formats of the web pages; text reduction algorithms to find suitable keywords in the pages; and information visualization techniques to display information on the limited screen space available. Some properties of WAP, the Wireless Application Protocol, were also considered.

3.1 Wireless Application Protocol (WAP)

WAP [32] is a de facto standard for providing Internet-based content and services to wireless devices such as cellular telephones, and requires resources to be coded in a dedicated mark-up language called WML (Wireless Mark-up Language) adapted to the limitations of such

devices. Although the WEST architecture is not specifically designed to work in conjunction with WAP, there is potential for interesting synergies when it comes to user interaction.

Firstly, the concept of deck (approximately corresponding to a page in WWW, i.e. a single resource transfer) and card (sub-unit of deck, i.e. a single display object) in WAP lends itself very well to visualization using flip zooming. The overview mode captures the collection of cards, i.e., the deck, whereas the zoomed view corresponds to viewing an individual card. Because of this nice correspondence, we have adopted the WAP terminology of cards and decks in the WEST system, although WAP protocols are not currently used in WEST.

Secondly, given a PDA type of device with WAP capabilities, a WEST browser could readily be converted into a WAP browser, i.e. processing WML instead of HTML. However, when it comes to mobile phones with small text-only displays, this is of course a completely different issue. In this setting, the simplest way of using a WEST browser would be to navigate in zoomed-only mode (i.e. without the context overview). A more advanced solution would involve creating overviews through keyword or link-extraction.

3.2 Proxy Systems

The notion of using a proxy server to mediate between the Internet and thin clients is well established [9, 23]. A proxy of this kind can have many functions: coding and conversion of protocols; filtering, compression and conversion of information, etc. The WEST proxy could be tailored to include protocol functions, but the work presented here focuses on the information handling aspects of a web proxy for mobile devices. By removing unwanted or unnecessary information, and by compressing and restructuring (chunking) the information, it can be made to better suit the usability demands of mobile terminals. When it comes to information compression, we can distinguish between *lossy* and *non-lossy* compression. This distinction is normally applied to images [9, 23], but in the context of WEST we will be dealing with lossy text reduction in the shape of text summarization techniques.

WEST has in common with the *Top Gun Wingman* browser for the 3Com Palm™ PDA [10] the basic principle of hiding complexity (such as HTML parsing and analysis) in the proxy server to off-load the hand-

held device. The idea of saving screen real estate by using text compression has been put to use in another proxy-based system known as the *Digestor* [2]. Proxy systems can also be used as support systems in "surgical" extraction of information from WWW and other sources, providing semi-automatic conversion of such pre-determined content into format suitable for thin clients, such as WML or Web Clippings in the Palm VII™ PDA. *Panama* from Oracle [26] is an example of such a system, which converts HTML and other formats into XML, from which selected WML fragments can be generated by means of stylesheets.

It should be noted that the work presented here does not take a stand as to whether pre-authored content (e.g., a WML source) or automatically converted and filtered content (e.g., HTML–>WML or HTML to simplified HTML) is the strategically correct way to produce services for the user of wireless handheld devices. We content ourselves with the observation that there will be a demand from mobile users for accessing arbitrary web-based resources, particularly in the initial deployment period where dedicated mobile services, WAP-based or not, will emerge slowly. Initially, the range of services and content for mobile use will be limited, since information providers will be reluctant to invest in parallel coding of content. Gradually, this will change (particularly if systems such as Panama are used, which allow for re-use of existing web resources), but there will always arise situations where users want to access material not pre-adapted to dedicated mobile formats. The WEST approach tries to address the needs of such users.

3.3   Text Reduction

For the keyword view, a text had to be summarized into a few words. We call this technique *text reduction,* to distinguish it from traditional text summarization. The major challenge for traditional text summarization techniques is two-fold: understanding which regions of a text bear the most pertinent information, and cobbling those bits of information together into a coherent summary. In the case of small screens, the space requirements are more demanding, which actually makes the task somewhat easier. Coherence will not be an issue, since the aim will be to extract a small number of information-bearing terms from the text, mak-

ing the task closer to the field of index term selection than that of text summarization.

Index terms are typically selected based on term frequency as originally proposed by Luhn [24], selecting suitably frequent terms to represent a document. However, the most frequent words in a text are usually form words, which bear little or no topical information ("is", "and" and the like). These words must be filtered out either through the application of a judiciously composed stop list or through the application of estimates of term specificity [30]. These are terms that occur in all documents in a document base and have no indexing power; terms that occur in few documents are more useful to that end. Typically, the two measures are combined, in a standard "tf.idf" formula (e.g. [29]). This was the basis for the keyword extraction algorithm used in our application.

3.4   Viewing Web Pages on Small Screens

Although personal digital assistants and other hand-held devices have been available for a number of years, the problems associated with user interface design for small terminals have only recently started to attract attention from the human-computer interaction research community [18, 25]. While many general principles for human-computer interaction also apply to small terminals, they can not always be taken for granted, and to simply transfer interaction components from desktop computers will often lead to unexpected problems [15].

Earlier research in information visualization techniques have focused mainly on maximizing the use of screen space on ordinary computer screens. A number of *focus+context techniques* have been developed to give users access to simultaneous overview and detail. General focus+context visualizations techniques such as the *Generalized Fisheye View* [11] or techniques developed for text documents, such as *SeeSoft* [7] or the *Document Lens* [27], might be adapted to the WWW. General zooming or multi-scale interaction techniques which have also been used for visualizing web pages include *PAD++* [1], *Cone Trees* [28], *Hyperbolic Trees* [22] and *Elastic Windows* [19], and techniques developed specifically with the web in mind include the *WebBook* and the *Web Forager* [6], *Zippers* [5] and *CZ Web* [8].

Although most of the techniques above have been developed for use on traditionally-sized screens, some of them might feasibly be adapted

for use on small screens. However, many of these techniques have advanced requirements in the form of computational resources for performing smooth graphical transformation and providing responsive interaction, and while they may often have proved useful on desktop machines, hand-held devices such as those on which the WEST system are intended to be used, are currently for the most part not capable of any advanced visual calculations. The focus+context technique *flip zooming* that was used in this project was also originally developed for ordinary screens, but because it is not very resource-intensive it has proven possible to transform it to smaller devices. For ordinary screens, it has previously been used for visualizing web sites [14], and has been generalized to handle hierarchical material such as hierarchically ordered image collections [17]. As part of the WEST project, we have evaluated flip zooming as an alternative to scroll bars on small screens [3].

4   Interaction in WEST

To give a better idea of how the WEST browser works, we will now give a detailed account for how a user may interact with the system. This will take the form of a complete interaction scenario, with an illustration for each screen the user will see.

The example page viewed in a traditional browser on a 160x160 pixel display

As our example, we have used a page reporting baseball news at the Yahoo Sports site. The page was comprised mostly of text – 319 words, or about 1500 characters. There were 15 links to other pages, plus a banner advertisement and a search function. As the figure above will attest, viewing this page on a traditional browser on a 160x160 pixel screen presents serious problems. Only a very small part of the page would

then be available at any time, giving almost no clues to the size or context of the material.

4.1 Flip Zooming in WEST

The interaction in WEST is based on *flip zooming,* a tile-based focus+context visualization technique. Flip zooming allows users to navigate a data set consisting of sequentially ordered discrete objects, e.g. images or pages of text. One object is in focus, the other objects provide the context. Users can move the focus backwards or forwards in the data set, or select any visible object as focus object by pointing at it. Users can also zoom in further on an object, allowing it to occupy the whole screen. Objects are ordered in a left-to-right, top-to-bottom fashion, so that any object that is after another object in sequence will be placed to the right and/or below the preceding object.

Earlier user studies of flip zooming applications [4] have indicated that users may become confused if thumbnails and focus objects are allowed to change their positions on the screen, or if the display is too packed with information. For this reason, we limited the maximum number of objects on the display at any one time to seven, which allowed us to keep the focus object at the center of the display with sufficient room to display the context objects at a reasonable size. It may seem that in some cases we are not using the available screen estate to the maximum, but this is a conscious trade-off to provide a clearer and more easily-understood display.

In WEST, some objects on the display are in fact representations of several objects, since they represent the top-most card of a deck. In this case, when zooming in on such a card, a user would be presented with a view of all the cards in the deck, which could then be navigated as usual. In this way, the user is in fact navigating a hierarchy comprised of decks and cards. In the example this hierarchy is only one deck deep, but there is no reason why it could not be more complex.

4.2 Interaction Example

We will now follow a user interacting with the sample page using the WEST browser. The user wants primarily to read about her favorite Chi-

cago Cubs player, Sammy Sosa, and possibly chat about his exploits with other supporters.[1]

1. Thumbnail view, whole page, with first card in focus

1. Initially in WEST, the viewer is presented with the *thumbnail view,* which gives an overview of the whole web page in flip zoom format. Here, each card is presented as a thumbnail image, not large enough to be readable, but still giving a sense of the overall nature of the page – e.g. image-heavy, text-heavy, many or few links, etc. The first card or deck is in focus, with the others presented as context. (Unfortunately, there is currently no clear visual indication of if a thumbnail represents a single card or a deck, something which might be addressed in future versions of the browser.)

2. Keyword view, whole page, with first card in focus

2. The user now chooses to switch to the *keyword view,* to see if she can locate some information about Sammy Sosa.

---

1. The authors know very little about the game of baseball, and apologize in advance for any errors in this account that may be spotted by more knowledgeable readers!

3. Keyword view, whole page, fourth card in focus

3. The keywords on the fourth card in the sequence indicate that Sosa is mentioned. The user focuses on that card. This can be done either by explicitly pointing at the card, or by moving the focus sequentially until the desired card is reached. (Since what here looks like a single card may in fact be the top card in a deck, user will actually often be navigating among decks in this manner.)

4. Keyword view, a deck open, first card in focus

4. The card in question is in fact a deck, consisting of two sub-cards in total. By zooming in on the visible card, the first card in the deck, the deck is opened and displayed. The keywords indicate that some kind of ceremony has taken place, involving Sosa and home-runs.

5. Thumbnail view, a deck open, first card in focus

5. The user now switches back to a thumbnail view of the deck, showing the original HTML formatting of the cards.

6. Thumbnail view, zoomed in completely on a card

6. The user zooms in completely on the first card in the deck and reads the text on the card. It is indeed interesting news about Sammy Sosa. Staying in this view, the user can now advance to the next or previous card in the deck (e.g. by pressing a specified button on the PDA or tapping on a portion of the card with the pen), to read the full story. (If the card on view happens to be the last in a deck, when advancing, the first card in the following deck will be shown.)

7. Thumbnail view, a deck open, first card in focus

7. The user now wants to chat with other supporters about this development. She zooms out again, returning to the overview of the deck.
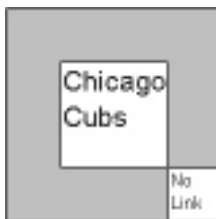
8. Link view, a deck open, first card in focus

8. The user now switches to the *link view,* since she is looking for a link to the chat page.

9. Link view, whole page, with fourth card in focus

9. Not finding the link she is looking for in this deck, she zooms out to reveal link view for the whole page.

10. Link view, whole page, with seventh (last) card in focus

10. She sees a link to the chat room on the very last card in the page, and focuses on that card. By clicking on the link while the page is in focus she will be transported to the chat-room page, meaning that the current web page/deck will be removed from the screen and the chat-room page/deck will be displayed.

5   Description of the Components

The components of the WEST system were designed as a number of modules that could be individually improved and expanded as the system was developed. In the following, we will describe each of these pieces separately.

5.1   Pre-processing, Including Card Chunking

Proxy servers for real-time pre-processing of web information to be accessed using a mobile terminal is a proven technique used for instance

in current web services for palm-sized PDAs. In WEST, we made use of a proxy server to:

- Filter and reduce the contents of web pages in order to adapt them to the capabilities of the mobile browser (this would mean among other things to get rid of JavaScript, image maps, frames etc.)
- Convert the reduced web page into n sub-pages (cards), each of which can be readily presented on a mobile-sized display (e.g. 160x160 pixels). Cards are inter-linked to form a deck by arranging them into a suitable reading-order
- Produce alternative renderings of these cards corresponding to different levels of detail. Typically a card can be displayed in its full size, in reduced size and minimized. These alternative renderings are not necessarily derived from graphical reduction – in WEST, one alternative when reducing card size is to use automatic text summarization

Key element such as headings, paragraphs, pictures, tables etc. provide hints on how the original page is structured. These hints are used in the "chunking" of the page into cards, i.e. determining break-points for card creation. The maximum allowed size of a card is of course a limiting factor, which sometimes means that the information contained within a card's minimal natural page-chunk cannot be presented without some modifications (for instance image or font size adjustments), or by splitting up the information into two cards. (For more information on the chunking algorithm see the appendix.)

The cards produced by the proxy are arranged into several decks linked together in the original reading order. Because of the limitations of the display, each deck was limited to seven cards, the maximum that could comfortably be displayed using the flip zooming variant we had chosen. If a page consisted of more than 49 cards (seven decks with seven cards each) some of the decks would in turn have to contain sub-decks of cards, creating a deeper hierarchy.

5.2   Extraction of Keywords

To extract the keywords that were to represent each card, the method chosen had to be suitably general to handle any kind of material. Since there is no way in advance to tell what type of web page a user will be browsing, the system should be equally at home at whatever topic it was

subjected to, including general news, sports, entertainment, and so on. It would be feasible to allow the creator of a page to specify which keywords are most relevant, but this would require that pages were specially constructed for the purpose of this system, and as mentioned, the intention was to give users access to all pages of the web without any prerequisites.

Our text reduction algorithm relies on the fact that typically several texts or text chunks will be compressed and displayed simultaneously. The text chunks are all short, approximately thirty words. The word tokens in the text chunks are tabulated for frequency, after the application of a stop list filter of form words. Each chunk is represented by a list of words sorted by frequency. These raw frequency counts are then modified by inverse document frequency [30] – each word will have its raw frequency count divided by a factor depending on the number of texts it has been found in. Thus, if a text has two words with equal frequency, where one occurs in three texts and the other only in the text at hand, the latter term will be weighted higher.

The text reduction procedure thus disfavors words that are evenly spread out over the chunks at hand, and aims at representing each chunk by as unique words as possible, in that given context of chunks. Words that occur disproportionately often in a given chunk, compared to other visible chunks, are favored above the more generally frequent words. But words with high frequencies that occur in many texts are not discarded. They are set aside and used to generate a header for the group of text chunks under consideration, and can be used for hierarchical reduction of the entire group, although this is not done in the current version of WEST. A group of chunks can be reduced to words such as "baseball", "scores", "season"; individual chunks can be more finely reduced to "sosa", "homered", etc. As mentioned previously, taking advantage of these headers could be particularly fruitful when using small text-only displays where the contextual overview does not fit.

As it stands today, the algorithm does not make use of morphological analysis, thesauri or lexical categories, all of which would increase the reliability of the results. Adding surface level linguistic processing is a modular issue and can be done without a system redesign: there are several efficient general-purpose linguistic analysis components suitable for this purpose.

5.3 Link Extraction

To facilitate a view of which links were available on a given web page a simple link extraction procedure was created. This went through each of the cards constructed in the chunking processing and created a similar deck structure, but where the content of each card would only be the hyper-links.

5.4 Web Page Rendering

For the graphical presentation of the different cards, each individual card had to be rendered as if it were a web page. However, we were unable to write a full-scale web rendering engine within the constraints of this project. Instead, we used the rendering engine provided by the HotJava Web Browser [31] to produce an image of each card as displayed on a screen of the required size (160x160 pixels). The same images where also graphically compressed to intermediate and thumbnail size. These pre-rendered images were then used by the system for the graphical presentation.

5.5 Presentation and Interaction

Based on the flip zooming technique, the WEST browser presents each web page as a number of discrete objects, representing individual cards or decks of cards. The user navigates between different objects by using directional buttons or by directly choosing the object to focus upon with a pen or other pointing device. For sequential reading of a whole page, a user would generally switch to a full-screen view and then advance through the cards by pressing a designated "forward" button.

Each view in WEST only presents one level of the hierarchical structure of decks and cards that represent the web page. To move between levels, the user zooms in on the object in focus (usually by clicking or tapping with the pen on it) and will thus go one level deeper into the structure. To go up one level, the user clicks or taps on the "white space" between the objects. This navigation might also be facilitated by the use of "up" and "down" buttons, for moving up and down in the hierarchy, analogous to zooming out and zooming in.

When the user goes down one level in the hierarchy, the focus object takes over the whole screen space to show its content. If the current focus represents a single card, this card will be allowed to fill the screen completely to facilitate reading. In the case of the focus representing a deck, however, a view of all the objects in the deck is presented.

The system provides three different modes in which the material can be viewed: thumbnail, summary and link view. When switching from one view mode to another, the position of the focus in the hierarchical structure is maintained, enabling the user to navigate in a suitable view mode to locate a card, and then change to another mode (typically the thumbnail view) to actually view the card. In the prototype, the user switched between the different views by accessing a pop-up menu.

6   User Experience

To gain some insight in how the WEST prototype performed with inexperienced users, we performed a qualitative evaluation in which the prototype was compared with the HotJava browser [31]. It is important to note that the test was in no way intended as a "fair" comparison between two browsers, since the HotJava browser was not developed with the intention of being used on very small screens. Rather, the intention was to gauge novice users initial reaction to the WEST browser, and the other browser was provided as a reference point only.

A test group consisting of ten subjects, all expert computer users but with no experience of browsing the web on a PDA, were set a number of tasks to perform both in the WEST system and the traditional browser. The tasks consisted of finding specific items in the material, and in some cases required returning to a part of a page which had previously been visited. The tests were performed on a traditional computer screen, but both browsers were given the same screen size as a typical PDA to operate in, i.e. a window of 160x160 pixels.

A questionnaire given to users after the test indicated that they thought that the prototype provided a better overview than the HotJava browser, ranking it on average 3.40 points higher in this respect (5.30 vs. 1.90, standard deviation being 0.68 and 0.99 respectively) on a scale of 1 to 7 with 7 being the best. It also showed that users thought searching was easier with WEST than with HotJava, ranking it on average

2.25 points higher on the same scale (5.55 vs. 3.30 with standard deviation 0.90 and 0.95 respectively). However, it was also noted that the flip zooming interaction technique took some time to get familiarized with, providing some initial difficulties.

Although we did not collect any quantitative measures during this preliminary experiment, the positive reactions of the users did provide us with an indication that the ideas behind the system should be worth pursuing further.

7   Future Work

At the moment, the system can be improved primarily in the following areas: improving the chunking of pages; improving the techniques used for text reduction; and improving the means of interaction with the system to make it useful in various realistic situations. We might also consider the division of tasks between the proxy and the client. At the moment, most of the work is performed on the proxy to off-load the client machine as much as possible. With faster hand-held machines, there is no reason to believe that not more or maybe most of the information processing such as keyword- and link extraction could take place on the client rather than on the server.

The chunking process still leaves much room for improvement, since often the provided cards are not of the optimum size for the available screen space. Improving the chunking is difficult, however, since there will have to be a balance between producing chunks that are logically coherent to the user, and chunks that are of maximum size. To achieve maximum chunk size it is sometimes necessary to break the pages at inconvenient places, even breaking text in mid-sentence, but this should be avoided for the sake of the user. A more thorough analysis both of page structure and user behavior will be needed to improve this process. Also, integrating the chunker more closely with the actual rendering of the HTML pages would make the judging of available space much easier.

The text reduction algorithm as it now stands is very simple. It is based on well established and understood techniques from text indexing, which guarantees a predictable, stable, and somewhat mediocre result. There are two well known bottlenecks in this type of information access

techniques: 1) we have too little knowledge of texts as texts to be able to answer the question of what a certain text is about, and 2) we have too little knowledge of what the text will be used for and why the user wants it. The second problem is somewhat less pressing for this specific application: we know that the text needs to be compressed, and we know what the context is, namely what else is being displayed at the same time. This knowledge we already utilize to some extent, since we are able to generate a header for the texts in view at any given time. The first problem is harder. Knowledge of texts is limited if we view texts as simple bags of words. In future work, we plan to utilize stylistic information [20] to reduce different types of text differently: a legal text might be reduced to a paragraph header, while a long-winded error message might be reduced to a generic icon. We intend to experiment with using text structure to tailor the chunking algorithm so that it will feed homogenous bits of text to the reduction algorithm (e.g. [13]). We might use language technology such as surface syntactic analysis [21] and text extraction techniques [12] to extract topical terms and other topical items such as names, links, or dates from the text segments. We are currently running a pilot project for multi-document summarization, to be able to impose a middle level of analysis: the idea is to collapse several texts into one short summary, whereupon that summary in turn can be reduced.

Finally, it might be possible to improve the interaction with the WEST system in certain usage situations. Using a pen to interact with a hand-held device is sometimes undesirable, since it requires the user to hold the device with one hand (or place it on a flat surface) and use the pen with the other. Essentially, flip zooming only requires four navigational actions to navigate a hierarchical data set (move the focus back, move the focus forward, zoom in and zoom out), and while the WEST browser requires additional input for switching among the different views, it is in many cases possible to use perform the majority of the navigation using only four buttons without relying on a pen. This might allow users to navigate with more precision and efficiency in some situations, and ideally it might even be possible to construct the system so that all navigational buttons were accessible using just one hand, thus freeing up the other hand for other tasks. This would make the human-computer interaction far more flexible, as there might be many situations when having one hand free would be beneficial: while talking in a phone, taking notes, etc.

8   Conclusions

Truly mobile web access will evolve along several paths. One path is the development of the "stripped-down" web, reminiscent of browsing with text-only browsers such as Lynx. The other extreme will result from miniaturizing standard computers into hand-held devices capable of handling the same resources as stationary machines. These paths will of course cross, and we will see combinations of dedicated mobile resources and advanced hand-held computers. No matter what, the restrictions of mobile terminals will always hold with respect to the usage environment. We believe work like WEST is important because it focuses on ways to enable advanced interaction on small devices, ways that are largely independent of the capabilities of both the network and the terminal.

By constructing the WEST system, we have shown how material on the World Wide Web can be made available for mobile users and others who are restricted to accessing the web from small terminals. By placing the major work-load on the proxy server, and by providing a novel combination of visualization and text summarization, existing web pages can already be made much more suitable for such devices. In the future, with the continued acceptance of hand-held devices and high-speed wireless network, browsing the web from a PDA or a mobile phone will be a common occurrence. In these cases, systems such as WEST may aid in making this a much more pleasurable and productive experience.

9   Acknowledgments

10 References

1. Bederson, B., Hollan, J., Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proc. UIST '94,* ACM Press, 1994.
2. Bickmore, T.W. and Schilit, B.N. Digestor: Device-Independent Access to the World Wide Web. In *Proc. Sixth International World Wide Web Conference,* pp. 655-663, 1997.
3. Björk, S. and Redström, J. An Alternative to Scrollbars on Small Screens. In *Extended Abstracts of CHI '99,* ACM Press, 1999.
4. Björk, S. and Holmquist, L.E. Formative Evaluation of a Focus + Context Visualization Technique. In *Proc. HCI '98* (poster presentation)*,* The British HCI Society, 1998.
5. Brown. M.H., Weihl, W.E., Zippers: A Focus+Context Display of Web Pages, in *CD-Rom Proc. WebNet '96,* Association for Advancement of Computing in Education (AACE), 1996.
6. Card, S.K., Robertson, G.G. and York, W. The WebBook and the Web Forager: An Information Workspace for the World Wide Web. In *Proc. CHI '96,* pp. 111-117, ACM Press, 1996.
7. Eick, S.G., Steffen, J.L. and Sumner, E.E. SeeSoft - A Tool for Visualizing Line Oriented Statistics Software. *IEEE Transactions on Software Engineering,* 18(11), 1992.
8. Fisher, B., Agelidis, M., Dill, J., Tan, P., Collaud, G., Jones, C., CZWeb: Fish-eye Views for Visualizing the World Wide Web. In *Proc. HCI International '97,* pp. 719-722, Elsevier, Amsterdam, 1997.
9. Fox, A., Gribble, S.D., Chawathe, Y. and Brewer, E.A. 29. Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. *IEEE Personal Communications* (invited submission), Sept. 1998.
10. Fox, A., Goldberg, I., Gribble, S.D., Lee, D.C., Polito, A. and Brewer, E.A. Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the USR PalmPilot. In *Proc. IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware '98),* Lake District, UK, 1998.
11. Furnas, G.W. Generalized Fisheye Views. In *Proc. CHI '86,* pp. 16-23, ACM Press, 1986.
12. Grishman, R. Information Extraction: Techniques and Challenges. *Materials for Information Extraction (International Summer School SCIE-97),* ed. Maria Teresa, Pazienza, Springer-Verlag, 1997
13. Hearst, M. and Plaunt, P. Subtopic Structuring for Full-length Document Access. In *Proc. ACM SIGIR '93,* ACM Press, 1993.

14. Holmquist, L.E. Focus+Context Visualization with Flip Zooming and the Zoom Browser. In *Extended Abstracts of CHI '97,* ACM Press, 1997.

15. Holmquist, L.E. When Will Baby Faces Grow Up? In *Proc. HCI International '99,* 1999. (to appear)

16. Holmquist, L.E. and Ahlberg, C. Flip Zooming: A Practical Focus+Context Approach to Visualizing Large Information Sets. In *Proc. HCI International '97,* pp. 763-766, Elsevier, Amsterdam, 1997.

17. Holmquist, L.E. and Björk, S. A Hierarchical Focus + Context Method for Image Browsing. In *SIGGRAPH '98 Sketches and Applications,* ACM Press, 1998.

18. Johnson, C. (ed.). *Proc. First Workshop on Human Computer Interaction with Mobile Devices.* URL: http://www.dcs.gla.ac.uk/ ~johnson/papers/mobile/HCIMD1.html, 1998.

19. Kandogan, E., and Shneiderman, B. Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser. In *Proc. UIST '97,* pp. 169-177, ACM Press, 1997.

20. Karlgren, J. and Cutting, D. Recognizing Text Genres with Simple Metrics Using Discriminant Analysis. In *Proc. COLING 94,* Kyoto, 1994. (In the Computation and Language E-Print Archive: cmp-lg/ 9410008).

21. Karlsson, F., Voutilainen, A., Heikkila, J. and Anttila A. (eds.) *Constraint Grammar,* Berlin: Mouton de Gruyter, 1995.

22. Lamping, J., Rao, R. and Pirolli, P. A Focus+Context Technique Based On Hyperbolic Geometry for Viewing Large Hierarchies. In *Proc. CHI '95,* ACM Press, 1995.

23. Liljeberg, M., Helin, H., Kojo, M., and Raatikainen, K. MOWGLI WWW Software: Improved Usability of WWW in Mobile WAN Environments, in *Proc. IEEE Global Internet 1996 Conference,* London, England, November 20-21, 1996.

24. Luhn, H. P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development,* 1 (4) 309-317, 1957. Reprinted in *Luhn, H.P.: Pioneer of Information Science, selected works.* Claire K. Schultz (ed.). New York: Sparta, 1968.

25. Marcus, A., Ferrante, J.V., Kinnunen, T., Kuutti, K. and Sparre, E. Baby Faces: User-Interface Design for Small Displays. In *CHI '98 Summary,* pp. 96-97, ACM Press, 1998.

26. *Oracle Project Panama, Connecting Oceans of Information.* Oracle White Paper, March 1999. URL: http://www.oracle.com/mobile/ panama/panamawp.htm

27. Robertson, G.G. and Mackinlay, J.D. The Document Lens. In *Proc. UIST '93,* pp. 101-108, ACM Press, 1993.

28. Robertson, G.G., Mackinlay, J.D. and Card, S.K. Cone Trees: Animated 3D Visualization of Hierarchical Information. In *Proc. CHI '91,* ACM Press, 1991.

29. Robertson, S.E. and Sparck Jones, K. *Simple, proven approaches to text-retrieval.* Technical report 356, Computer Laboratory, University of Cambridge, 1996.

30. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation,* 28:1, pp. 11-20, 1972.

31. Sun Microsystems. *HotJava HTML Component.* URL: http://java.sun.com/products/OV_hotjavaProduct.html

32. WAP Forum, *Wireless Application Environment Overview,* February 3, 1999. URL: http://www.wapforum.org

11 Appendix: The Page Chunker

To divide a page of HTML code into a number of pieces or *chunks*, each suitable for displaying as a single full-screen card on a small display, a page chunking program was developed. It was based on an existing HTML parser (or more accurately, an SGML parser with a description of HTML's elements) written in Java by Richard M. Tobin and available at the following address:

```
http://www.cogsci.ed.ac.uk/~richard/ftp-area/
html-parser
```

First, the page chunker establishes a number of constants, such as the size of a card (e.g. 160x160 pixels), the typical width of a character, the height of a line, maximum number of lines that can fit on a card, and so on. It then reads a piece of HTML code from an URL and performs a number of operations depending on the HTML elements encountered. Operations include:

- Setting flags for elements that can not be split and/or that are suitable as break-points (e.g. H1-H6, HR, A, IMG)
- Reducing the value for the total remaining space on the card (e.g. IMG)
- Adapting the width of the HTML element to the maximum available (e.g. PRE, HR, TABLE)
- Adapting the total size of the HTML element (width and height) to the maximum available (e.g. IMG, APPLET, OBJECT)

Additionally, some tags are replaced with tag combinations that will be handled in a more predictable way during page rendering; for instance, the paragraph tag <P> was replaced with
&NBSP;&NBSP;&NBSP;

The chunker also makes sure that no tags are left "open" on a card, e.g. an opening <H1> with no corresponding closing </H1>. HTML elements are then added to the card until it is full, or as close to full as the algorithm can manage, at which time a new card is started.

After creating the cards, a number of decks are created. The current design of the flip zooming display in WEST limits the number of cards simultaneously visible at any time to 7. For simplicity's sake the deck

creation algorithm simply tries to create a maximum of 7 decks with as equal a number of cards as possible. The resulting HTML files are saved in a file structure corresponding to the decks (i.e. one directory for every deck) which can then be read by the WEST browsing component.

Pseudocode for the page chunker is as follows:

```
chunkPage (parameter: URL for HTML page)

  parse HTML page
  save away header (<head> ... </head>)
  chunkBody; (divides page into cards)
  collect cards into decks and create corre-
   sponding file structure


chunkBody (parameter: HTML element)
   if not (tag=skiptag)
   then
    modify tag if needed, and add the (starting)
      tag (e.g. <IMG>), including attributes
      (e.g. an image) to body
   else
    if (tag=<p>) then reduce available space on
      this card with no. of characters on a line
   for (all sub-elements)
    if (element=text)
     then
      addText; (add this text to the new card)
     else (i.e. element=tag)
      addTag; (add this tag to the new card)
    if not (tag skipped) then add finishing tag
      (e.g. </IMG>)


addText
   (divide until it fits)
  while (number of characters added so far +
    length of new string >= maxlength)
    if (tag can not be split)
```

```
      then
        add string to current body
        add finishing tags to all open start tags
         and finish this body
        add corresponding start tags to new body
      else
        check where it is suitable to break (at end
         of paragraph/sentence etc.)
        add what we can fit in to the current body
        add finishing tags to all open start tags
         and finish this body
        add corresponding start tags to new body
    if (text left)
      then add remaining text to current body

addTag
  if (break condition) (i.e. is this a tag that
   can cause the creation of a new card?)
   then
   if (available space on current card is less
    than 10% of maximum size)
     then
       add finishing tags to all open starting
        tags and finish this body
     add corresponding start tags to new body
  chunkBody; (continue chunking body until done)
```

POWERVIEW

Using information links and information views to navigate and visualize information on small displays

Staffan Björk, Johan Redström, Peter Ljungstrand, Lars Erik Holmquist

PLAY: Applied research on art and technology
The Interactive Institute, Box 620, SE-405 30 Göteborg, Sweden
http://www.playresearch.com
{staffan.bjork, johan.redstrom, peter.ljungstrand,
lars.erik.holmquist}@interactiveinstitute.se

Abstract. PowerView is a PDA application designed to support people with situational information, primarily during conversations and meetings with other people. PowerView was designed to address a number of issues in interface design concerning both information visualization and interaction on small, mobile devices. In terms of information visualization, the system was required to provide the user with a single integrated information system that enabled quick access to related information once an object of interest had been selected. In terms of interaction, the system was required to enable easy and efficient information retrieval, including single-handed use of the device. These problems were addressed by introducing *Information Links* and *Information Views*. An evaluation of the application against the standard application suite bundle of the PDA, a Casio Cassiopeia E-11, proved the interfaces equivalent in usability even though the PowerView application uses a novel interface paradigm and the test subjects were given no training time with the system.

1   INTRODUCTION

The popularity of Personal Digital Assistants (PDAs) has increased rapidly in the last few years. One area where PDAs have become especially widespread is among mobile workers, as such devices give users access to digital information while on the move. However, the environment in which PDAs are used is often quite different from a typical office environment, in which many environmental variables can be predicted. In contrast, when designing the user interface for PDAs not only must one assume that the environment may lack comfortable working positions, have bad lighting and be distracting, but also that it may change during a single use session. For instance, using a PDA on a subway means that the lighting changes as the subway car moves between stations. Further, a slight shaking can be expected during the whole ride, and if one hand is occupied with holding a handle in order for the user to keep his or her balance, the use situation becomes even further distanced to the traditional office environment.

In addition to the constraints posed by the physical environment, the small size and form factor of PDAs introduce several new constraints for human computer interaction design. PDAs typically have much smaller screens, less computational power and
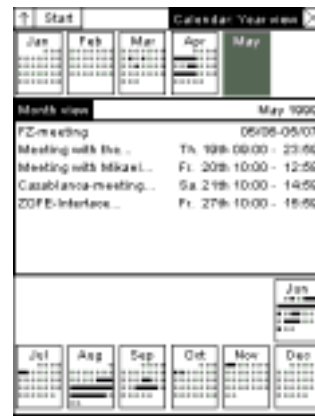
memory, and perhaps most important, limited input facilities. Most PDAs rely primarily on stylus-based input using a touch-sensitive screen, something that demands the use of both hands.

These new situations of computer use do not only create new interface problems regarding user interaction with applications, new requirements are also posed on how information should be visualized on the limited display area. With PowerView, we wanted to explore design issues such as how information visualization techniques can be applied on PDAs, how information navigation can be designed in order to enable fast retrieval of information relevant to the situation at hand and how to allow for single-handed use. In this paper we describe the PowerView application, the design issues and their respective solutions, as well as how the application works in practice.

2   POWERVIEW

PowerView was designed to support the user with relevant information during activities such as conversations and meetings with other people. To do this, we designed an integrated interface to the most common kinds of information stored on PDAs, i.e., contact lists, emails, tasks and meetings. Even though Power-View technically is an application running under Windows CE, it was designed not to use any of the GUI components of Windows CE in its user interface in order to fully explore alternative interaction techniques. In doing so, we tried to avoid or minimize the use of widgets commonly used in large screen GUIs, e.g. buttons, menus, checkboxes and window manag-

Fig. 1. Typical screenshot from the PowerView application.

ing operators (see **Fig. 1** for an typical screenshot of the application). A short description of the system has been published [2].

As different variants of PDAs provide different interaction possibilities in terms of display area, computational power, and input methods, the choice of device greatly influenced the design of the application. The PowerView application was implemented on a Casio Cassiopeia E-11 (see **Fig. 2**). The Cassiopeia is one of the more common PDAs on the market. It is relatively powerful, and it was used as it offered the possibilities of quick prototyping in the high-level object-oriented language Waba (a subset of Java). The device weighs slightly less than 200 grams and can be held by one hand. It is equipped with a touch-sensitive screen capable of displaying 240*320 pixels and 4 shades of gray, a 49 MHz NEC Vr4111 CPU, and 8 Mb RAM. Besides the use of a stylus for input, it has six buttons (excluding the on/off button) of which two can easily be accessed when programming the device in Waba. One of the two buttons is an exit button, similar to the escape key on a keyboard. The other one is an "Action Control" (see **Fig. 2**) that can be used in three ways: to rotate upward, to rotate downward, or to select by pressing it (inwards). These operations correspond to using the up and down arrow keys and the enter key on traditional computer keyboards.

Fig. 2. The Casio Cassiopeia E-11 with a close-up on the Action Control.

2.1 Design Issues

PowerView was designed to address a number of issues in interface design concerning both information visualization and user interaction on PDAs.

Information Visualization. Providing a user with as much information as he or she needs to perform a task is an almost ever-present problem when designing computer applications. Many information visualization techniques have been developed and claim to give users efficient views of information (e.g., [5,6,10,12,14]). This problem often becomes more difficult with PDAs, as users often need access to almost the same information as they do on their desktop computers despite having only a fraction of the space of an ordinary display available.

Several information visualization techniques have recently been applied to devices with small screens [2,4,16]. However, when techniques developed for desktop displays are to be used on PDAs, they often need to be modified to fit the new constraints posed by the small devices. This includes not only the limited display area (which may require designers to abandon ideals such as showing all available information), but also taking into account limited computational powers, memory space and changing networking capabilities.

Information Navigation. Regardless of how the information is visualized, PDAs need to use several separate views to present information that could be presented simultaneously on a device with a larger display. Increasing the resolution of PDA displays to show more information does not resolve this problem, since the limited size of the screen would make the presentation too small to be readable. Thus, users need to navigate between several different views on PDAs in order to access the same amount of information as displayed on a single view on a desktop or laptop. Each switch between two views requires the user to re-focus on the new information presented, and also requires the user to relate it to the previous information in order to make sense of it. These transitions adds an overhead cost to the interaction as the user must explicitly choose what view to switch to, make the switch, observe the effects of the switch and make sense of the new information displayed. This added overhead takes time and concentration from the user's intended activities.

This overhead can partly be mitigated by closely mapping how the information is visualized with how the underlying information is structured. For example, if an address book can not show all entries at once, it is feasible to divide the presentation of contacts into groups where all names in a group starts with the same letter. In this way,

if the user must navigate to find a contact, each change of view corresponds to moving from one letter to another.

When the number of items in a group becomes too large, or the number of groups becomes too large to be displayed simultaneously, further divisions are required, creating hierarchical structures of views. These structures require the user not only to switch between different views when moving between items, but also to move between different levels of views, as some views will be used to categorize other views. This creates further overhead, as the user is presented with the same information on several different levels of detail. This becomes most apparent when a user moves from viewing an individual piece of information to viewing another individual piece of information of another information type, as the user has to move from the bottom of a hierarchical structure to the top, switch to viewing another hierarchical structure and navigate to the bottom of that structure. A typical example of this problem is when switching between reading an email from a person to looking for that person's phone number in the contact list application.

While applications like Microsoft Outlook or the Active Desktop on Windows CE devices do integrate a few common types of information, they are still organized in a way that requires the user to explicitly switch between different views or modules in order to obtain the desired information. The PowerView application described in this paper aimed at taking this integration one step further to better support quick retrieval of related information once a certain interest, or focus, had been selected.

Interaction Constraints. One of the major differences between the use of desktop computers and PDAs is that many PDAs rely on stylus-based input. This has several implications for interaction design. First, input is slower than using a keyboard, which makes text-based operations, such as searching, less attractive. Second, when the user manipulates an object on the display with the stylus, the stylus and the hand holding the stylus covers parts of the screen. Third, if the user is walking, riding a bus, etc., the PDA is likely to be slightly in motion, which makes high-precision stylus-based manipulation on the screen more difficult to accomplish.

Ethnographical studies have shown that in many work situations, it is not feasible to require that users have both hands available for interacting with a device [9]. For instance, the user might be using a mobile phone with one hand while she is trying to retrieve information from the PDA. Various new input forms have been introduced that allow single-handed control, e.g. by using key cards [15], enhanced track point devices [8], or making a device tilt-sensitive [13]. However, all of these require new, or at least modified, hardware that have yet to become integrated in publicly available PDAs.

Interestingly, many commercially available PDAs already have buttons that can be used by the hand holding the device. Currently, such buttons are only employed to a limited extent, for instance to scroll in menus, as commercially available PDA user interfaces require point-and-click interaction with the stylus for nearly all operations (an interaction style seemingly inherited from mouse operations on desktop computers). Instead of designing new input devices, we wished to take advantage of these already present buttons to explore single-handed interaction. The use of such buttons

for navigation avoids some of the problems associated with stylus-based interaction, but requires that the navigation can be achieved using only a few degrees of freedom.

In order to improve usability in situations where the user only has one hand available for interaction, PowerView was designed to be possible to control solely using the buttons available on the PDA (see **Fig. 2**). This constrained the interface to be possible to control using four degrees of freedom only, corresponding to forwards and backwards, select/enter and exit/up.

Context of Use. One difference between stationary and mobile IT support is that while users have to bring their task or problem to the stationary computer, they can bring their mobile devices with them for use when and where they need them (cf. [11]). Since mobile devices such as PDAs rarely have the capabilities of stationary computers, they are not likely to be the complete solution to the users' problems. Instead, they are more of a support in activities where, ideally, the users' main focus is on the activity taking place rather than the technology supporting it. One implication of this is that applications on PDAs should be able to support activities while requiring as little attention (in the form of interaction) as possible, since the user may have focus on an activity outside the device.

As an illustration of this difference between stationary and mobile computing, we might think of the how users typically work with text on a PDA in comparison to a stationary computer: on stationary computers users often work with word processors in order to write full texts like this paper; on a PDA short notes during a meeting or a phone call are more likely. Even though both tasks could be accomplished on both platforms, this typical usage illustrates a basic difference in the design objectives of PDAs compared to desktop computers.

3   Using PowerView

In order to illustrate how the PowerView interface works, a sequence showing a typical interaction with the system is given in figures **Fig. 3-8**. In this example, the user wishes to find out what meetings are booked with Mikael Goldstein. To do this, the user locates and selects the name in the Contact list. This creates a *Context View* that provides information linked to the person, including past and future meetings booked with him, email sent to and received from him, and tasks that are related to him. Within this Context View, the user then choses to obtain more information about meetings, getting a detailed list of all meetings with Mikael Goldstein.

Initially, the user is presented with the *Overview*, in which information from all domains are visible in four different tiles (see **Fig. 3**). By clicking on one of the tiles with the stylus, the system sets that tile as focus tile and changes the visualization accordingly. Clicking on the focus tile with the stylus activates one of the *navigational views*, from which the user can navigate to individual data entries. To find Mikael Goldstein, the user sets the meeting tile as the focus tile (see **Fig. 4**) and activates the navigational view for meetings by selecting it again.

As soon as the meeting tile of the Overview is selected, the system switches to the Address Book View. This navigational view divides all contacts into tiles based on the first letter of the surname, representing the number of contacts in the context tiles as black lines, and the full names in the focus tile (see **Fig. 5**). Similar to the Overview, the user can move the focus between tiles by clicking on them or by using the action control. By moving the focus to the tile containing contacts with surnames starting with G, the contact Mikael Goldstein is identified (see **Fig. 6**). As the individual piece of information is now shown, it can be selected in order to switch to the Context View information view.

The Context View (see **Fig. 7**) visually resembles the Overview as it depicts information from all information domains in four separate tiles. However, the information shown in the context tiles is selected because it is *linked* with the object in focus. This gives a limited context containing only the information that the user has previously deemed relevant. In this view, the user can decide to look at one of the information objects in detail by moving the focus between the tiles. To examine the meetings associated with Mikael Goldstein, the user simply moves the focus to the Calendar (**Fig. 8**).

It should be noted that as all interaction in the example, and in the application as a whole, can be performed using the Action Control (see **Fig. 2**) only. This ensures that every step can be performed using the hand that is holding the device.

4   Interface Design

In order to create an integrated user interface for the PowerView application, all the design issues discussed previously had to be solved together. We addressed this by introducing *Information Links* and *Information Views*. By using the two concepts together, a unified presentation was created that at every point focused on supporting the user with information and allowed for a coherent way of navigation for all information types.

Fig. 3. The Overview, the initial view in the PowerView application.

Fig. 4. The Overview with focus on the AddressBook.

Fig. 5. Initial view in the Addressbook.

4.1 Information Links

In order to solve the problem of extensive information navigation to move between individual pieces of related information of different types, it was necessary to find a way of showing several different information types simultaneously. To do this, we needed a system to describe what information was related to a chosen piece of information. *Information links* was introduced to solve this problem.

Information links simply indicate a (semantic) connection between two pieces of information, where the two pieces of information can belong to different information domains. Information links enable the collection of various pieces of information that together form a heterogeneous context to an object (focus) the user selects, while still preserving a homogenous structure for every information domain. Thus, the information links form a semantic layer of relations between objects in the different data types on top of the storage of each data type. Information links differ from hyperlinks in that they are not used to traverse different presentations (e.g. web pages), but rather to define a context for any given piece of information. The strategy of using links or connections between objects to represent semantic properties is frequently used in linguistics (e.g., WordNet and other semantic networks) and cognitive science (e.g., connectionism) among others.

The type of connection provided by an information link is consciously designed to be explicit, i.e. that the user determines if two pieces of information should be linked together and can choose any criteria for doing so. This makes the system, and the visualization, flexible and enables the user to adapt the visualization in some respects to the environments in which the application is used. While this makes it necessary for the user to perform additional actions in order to establish these links, this effort can be made when the users wants to and has the time.

4.2 Information Views

An information view is a collection of correlated objects displayed together to help the user with some

Fig. 6. The Address book with focus on the Letter G.

Fig. 7. The Context view with the entry Mikael Goldstein selected.

Fig. 8. The Context View with focus on the Calendar.

activity. The objects are active in that they can change their appearance depending on the user's actions and are used to activate changes in the application, including switching to other information views. However, an information view should always be identifiable independently of the current state of the objects in it. The information views can present several different types of information, distinguishing them from (most) 'standard' applications, and are designed to function together with each other to support the user in more complex tasks that traditionally would have required the use of several different applications.

The information presented in an encyclopedia about individual countries can be used as an example of an information view, with the exception that it is a static presentation. Such presentations often give a collection of several different types of information, e.g., a map showing geographical data, a box containing demo graphical data, and a text body describing history, religion etc. These provide an informative overview of the country. However, depending on what one needs to know about the country in question, different such combinations of information presented in various ways, e.g., as maps, tables, etc., are very useful.

As the PowerView application was designed to support the user with information during meetings and conversations, three categories of tasks could be identified: showing what information was available on the device, selecting the information of interest, and presenting the selected information. Further, these tasks have to be completed in this order, which made the design of how the information views should be used together easier.

Focus+Context Visualization. Based on our experiences of working with the Flip Zooming visualization technique [7], including applying it to small screens [4], we decided to base each information view on a Flip Zooming visualization. Flip Zooming belongs to a class of information visualizations techniques called focus+context visualization. These are characterized by having one central object, the focus, presenting more detailed information, while simultaneously presenting contextual information in the surrounding area. Flip Zooming does this by dividing the information into a number of rectangular tiles of which one is the denoted the focus tile and the remaining context tiles. When creating Information Views using the Flip Zooming technique, each object in the Information View is simply mapped to a tile in the Flip Zooming visualization. The tile selected as focus is given most of the display area and the user can change which tile is in focus using random access methods such as a stylus or mouse. As the tiles are ordered sequentially, it is also possible to "flip" an object into focus by navigating "forward" and "backward". Having a focus tile allows the user to have a more detailed presentation of information before changing to another information view by selecting it. This enables some exploration at every point without hiding tiles or immediately changing information view.

The Flip Zooming technique allows for hierarchical visualizations [1], i.e., the ability to use one information visualization within another. This allowed for a natural mapping of how the information in the application is structured to how it is visualized, with an information view at each node in the hierarchy. The tasks handled by the different information views had to be presented in a certain order. That order determined the

order in the hierarchy, with the information view responsible for presenting the available information at the top and the information view presenting a chosen piece of information at the bottom. The hierarchical Flip Zooming technique is possible to navigate using only four operators, i.e., forwards, backwards, select and exit/up. Thus, it meets the requirement for single-handed use with the Cassiopeia device.

On ordinary desktop displays, it is possible to show all information visualizations used in a hierarchical Flip Zooming visualization simultaneously. For several reasons, this approach was not suitable for PDAs. Firstly, the limited display area of PDAs made it impossible to intelligibly show all information views simultaneously. Secondly, as each information view was designed to help the user with one particular activity, the information given in the surrounding information views would not add information vital for that activity, and thus distract the user and lower the usability of the application as a whole. By limiting the application to show only one information view at any one time, we gave up the idea of having a global context consisting of all information views, in order to have a local context for every view that helps solve the task associated with that view. It should be noted that this limitation differs from most applications, where the presentation of more information of one type is prioritized over the possibility of showing information of several different types throughout the application.

Fig. 9. Flow model of interaction with the PowerView application.

4.3 Description of the Information Views used in PowerView

For each category of sub-tasks, one or more information views were created. First, The OverView information view shows a summary of the information available from all four information types. Second, to allow the user to select a piece of information, one or more Navigational views (depending on the number of objects and detail in the information structure) were created to navigate within each of the four information domains. Third, the Context View is used to present the information selected by the user. (See **Fig. 9** for a model of how the information views are related.)

OverView. The presentation of the top level of the hierarchy is named the Overview view, as it presents information of all four information types. Here, four objects representing four separate types of information are presented: contacts, email, tasks and meetings. When the user selects one of the objects, PowerView switches to the navigational view corresponding to the information type of the object.

Navigational Views. After selecting an information domain in the OverView, the user is presented with the navigational view for that domain. This view shows all available information in that domain in an abstract form, ordered into objects according to the nature of the information domain. By choosing one of these objects, the user can move to another navigational view that only presents the selected part of the information. Thus, each navigation view helps the user with the task of choosing a region or an item of an information domain, and the navigational views as a whole provide the user with a structured navigation for selecting an individual piece of information.

As PowerView handles four information domains, four groups of navigational views were created. The information domain in the previous usage example was navigated by using only one navigational view. In cases where the information domain is structured into hierarchical structures, the user would have to move to other navigational views before being able to select a piece of information, and then switch to the Context View. For a flow scheme of such an interaction example, without the changing of focus in either the OverView or the Context View, see **Fig. 10**.

Depending on the amount of information and the structure of the information, different numbers of navigational views were required for each domain. In the case of meetings, different navigational views was created for handling years, months, days and hours, while for email only one navigational view separating received, sent and draft email was required. For the purpose of exploring information visualization on PDAs, we did not deem it necessary to create numerous navigation views for each information domain. In use situations where the amount of information requires more navigational views, these can easily be incorporated into the application when needed.

Context View. When the user selects an individual piece of information at the bottom level in a navigational view, the system switches to the Context View. In this view, the selected piece is displayed together with all information linked to it with information links. For instance, if a meeting is selected, the Context View shows information about people associated with that meeting, as well as email received from or sent to them and tasks that have to be done before the meeting. By mixing the information domains in this fashion, the problem of having to navigate through all the information in the system to move between two related pieces of information of different types was reduced.

Fig. 10. Interaction flow scheme.

As the types of information displayed vary depending on which piece of information the user selects, the Context View was designed to be able to show individual pieces of information from all domains simultaneously. This had the added benefit that the same Context View (with different information) could be used from all navigational views.

5   USER EVALUATION

In order to evaluate the application, the PowerView interface was benchmarked against the standard application bundle in Windows CE. Sixteen paid university students (10 women and 6 men, aged 17-43) were given 7 tasks to be performed on both systems in two different user situations. None had any prior experience of a PDA but all were familiar with using the Windows operating systems on stationary computers. The experiment was conducted at the Usability Lab at Ericsson Research in Kista, Sweden.

The evaluation showed that the users perceived that the arrangement of information was significantly better on the PowerView application ($F[1,15]=8.497$, $p=0.011$). Although users received no description of the PowerView interface, and were only allowed six minutes to freely familiarize with the interface before the experiment, no significant difference between task completion time could be found. However, none of the users utilized the single-handed navigation offered by the action control, despite situations where two-handed navigation led to physical discomfort (e.g., holding a mobile phone by the neck).

6   CONCLUDING REMARKS

PowerView allows mobile users to access information using an interface designed to work in a supportive rather than attention-demanding fashion. The use of information links and information views offers a new solution to the presentation and navigation of information on devices with small displays, breaking away from the traditional concept of using one application for each information type.

PowerView also supports single-handed navigation and retrieval of information in the entire application due to the restricted degrees of freedom in the interface. Although this was a feature that subjects in the evaluation did not employ, we have argued that single-handed navigation may be necessary in some use contexts since users may have the other hand occupied. Despite the restricted degrees of freedom in the interaction process, the usability evaluation showed that the PowerView application was equivalent in usability for completely novice users in comparison to the Windows CE application bundle.

The use of information views and information links in applications has only briefly been explored in the work described here. Future work is needed to fully validate the generality and usability of these concepts. Considering PowerView, the evaluation identified several possible improvements regarding both interaction and visualization, which will be addressed in future research. Further, PowerView has been identified as

a possible basis for an application to organize communication and information together, which will also be explored in the future.

7   ACKNOWLEDGEMENTS

8   REFERENCES

1. Björk, S. Hierarchical Flip Zooming: Enabling Parallel Exploration of Hierarchical Visualizations. In Proc. of AVI 2000, ACM Press, 2000. (To appear)

2. Björk, S., Holmquist, L.E., Ljungstrand, P., and Redström, J. PowerView: Structured Access to Integrated Information on Small Screens. In Ext. Abstracts of CHI'2000, ACM Press, 2000. (To appear)

3. Björk, S., Holmquist, L.E. and Redström, J. A Framework for Focus+Context Visualization. In Proc. of IEEE Information Visualization '99, IEEE Press, 1999.

4. Björk, S., Holmquist, L.E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., and Franzén, K. WEST: A Web Browser for Small Terminals. In Proc. of ACM UIST '99, ACM Press, 1999.

5. Card, S.K., Mackinlay, J.D., and Shneiderman, B. Information Visualization. In Card, S.K., Mackinlay, J.D., and Shneiderman, B. (Eds.) Readings in Information Visualization: Using Vision to Think, pp. 1-34, Morgan Kaufmann Publishers, San Francisco, California, 1999.

6. Furnas, G.W. Generalized Fisheye Views. In Proc. of CHI '86, pp. 16-23, ACM Press, 1986.

7. Holmquist, L.E. Focus+Context Visualization with Flip Zooming and the Zoom Browser. In Ext. Abstracts of CHI '97, ACM Press, 1997.

8. Kawachiya, K., and Ishikawa, H. NaviPoint an input device for mobile information browsing. In Proc. of CHI 98, pp. 1-8, ACM Press, 1998.

9. Kristoffersen, S., and Ljungberg, F. "Making Place" to Make IT Work: Empirical Explorations of HCI for Mobile CSCW, In Proc. of Group'99, ACM Press, 1999.

10. Leung, Y.K., and Apperley, M.D. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. In ACM Transactions on Computer-Human Interaction, Vol. 1, No. 2, pp. 126-160, ACM Press, 1994.

11. Norman, D.A., The Invisible Computer, The MIT Press, Cambridge, Mass., USA, 1998.

12. Rao, R., Pedersen, J.O., Hearst, M.A., Mackinlay, J.D, Card, S.K., Masinter, L., Halvorsen, P-K., and Robertson, G.G. Rich Interaction in the Digital Library. In Communications of the ACM, Vol. 38, No. 4, pp. 29-39, ACM Press, 1995.

13. Rekimoto, J. Tilting operations for small screen interfaces. In Proceedings of the ACM symposium on User interface software and technology (UIST) '96, page 167, ACM Press, 1996.

14. Sarkar, M., and Brown, M.H. Graphical Fisheye Views. In Communications of the ACM, Vol. 37, No. 12, pp. 73-84, ACM Press, 1994.

15. Sugimoto, M., and Takahashi, K., SHK single hand key card for mobile devices. In Proc. of CHI '96, pp. 7-8, ACM Press, 1996.

16. Taivalsaari, A. The Event Horizon User Interface Model for Small Devices, Technical Report TR-99-74, Sun Microsystems Laboratories, 1999. Available at http:// www.sunlabs.com/technical-reports/1999/

A Framework for Focus+Context Visualization

Staffan Björk, Lars Erik Holmquist and Johan Redström

Abstract. Focus+context visualization techniques aim to give users integrated visual access to both details and overview of a data set. This paper gives a systematic account of such visualization techniques. We introduce the notion that there are different levels of information visualization, with focus+context being a second-level visualization, and illustrate this with examples. We then provide a formal framework for describing and constructing focus+context visualization and relate this to the examples. A description of a software framework based on the principles of the theoretical framework follows, and we give some examples of how different focus+context visualization applications have been constructed using this framework. Finally, we discuss the implications of the formal framework and outline some future work in this area.

1   Introduction

Information visualization is widely acknowledged as a powerful way of helping users make sense of complicated data, and a great number of methods for visualizing and working with various types of information have been presented. However, all information visualization techniques will have to comply to one inherent limitation: they will need to limit themselves to the available area of a computer screen. A common solution to this problem is to provide some kind of movable view-port to the data, which can be controlled through the manipulation of scrollbars or other means. Zooming interfaces have also been introduced to let users control the amount of data shown, e.g. [3]. Sometimes, however, it might be important to give users access to both overview and detailed

information at the same time; such techniques include [21], with separate areas for overview and detail-on-demand information.

Here, we will concentrate on a certain family of techniques, that attempt to integrate both detail and overview on the same display area in an effort to not divide the user's attention. Some terms which have been used for such techniques include *fisheye views*, *distortion-based presentations* and *detail-in-context visualizations*. In the following we will use the term *focus+context visualizations*, which is wide enough to encompass all the properties we will be discussing.

2   Related Work

Although the origin of focus+context visualization can be traced back to non-interactive distortion-based techniques for visualization of map data [14], the first computer-based interactive method was introduced with the *FISHEYE View* [8], more known as the *Generalized Fisheye View* [9]. This original fisheye notion was in fact a general interaction framework for information filtering according to the user's current point of interest in the material, rather than a specific visualization technique, and was shown to be applicable to various types of data, notably structured programs and tree structures. (Some confusion has been the result of several other techniques using the term "fisheye", and currently fisheye visualization is often more closely associated with distortion-based techniques that give the graphical impression of the fisheye-lens of a camera.) In connection with the Generalized Fisheye View, important concepts such as the *Degree of Interest* (DoI) function and the *Level of Detail* (LoD) were introduced.

Another early interactive example of focus+context visualization was the *Bi-Focal Display* [29], where a graphical focus+context display was applied to a calendar display, introducing distortion in the horizontal dimension. A somewhat similar technique, the *Perspective Wall* [20], used a 3D perspective to achieve the same effect. The *Document Lens* [24] developed the concept further by combining a perspective view with a magnifying-glass effect to give combined detail and overview presentation of a document. Other techniques that use various forms of distortion to display two-dimensional images or maps include the *Graphical Fisheye View* [25] and *Rubbersheet View* [26], and forays

have been made into extending such techniques to three dimensions [7].
Flip zooming [11] was developed to visualize sequentially ordered
material, and it has been used for visualizing documents [12] and hierar-
chically ordered image collections [13]. Techniques developed specifi-
cally for visualizing graphs and hierarchies include *Hyperbolic Trees*
[18], the *Continuous Zoom* [2], and *Cone Trees* [23].

Among papers seeking to classify or formalize focus+context tech-
niques, [19] is probably the most widely cited. It gives an overview of
the various techniques and provides a unifying theory in the form of a
rubber-sheet analogy. [10] introduced *Space-scale diagrams* as a frame-
work for analysis of multi-scale (or zooming) interfaces, and showed
that such diagrams could also be used for describing focus+context
techniques. So-called *Non-Linear Magnification Fields* [16] have been
introduced as an abstract representation of distortion-based magnifica-
tion techniques, and these have since been more generally applied to the
problem of detail-in-context visualization [17]. [28] introduced several
dimensions of transformation, *X, Y, Z,* and *W,* where the W-transforma-
tion corresponded directly to the Generalized Fisheye View.

3 The Focus+Context Visualization Process

3.1 Levels of Representation

When describing information visualization, it is often sufficient to
describe the underlying data, how the data is represented and what
manipulation or interaction this representation will allow [6, 30].
Manipulation can be either manipulating the data itself, or, if the visual-
ization is interactive, manipulating the way in which the data is pre-
sented. Focus+context visualizations can also be described in this way.
However, we argue that it is useful to describe a focus+context visual-
ization as a *second-level visualization*, i.e. a visualization of a visualiza-
tion.

To clarify this, consider the rubbersheet metaphor as described in
[19]. Here, a focus+context visualization is compared to a sheet of rub-
ber that has an image of some sort printed on it, e.g. a map or document.
The rubbersheet is tied up in a rigid frame, representing the fixed size of
the screen. Magnification of a certain area can then be achieved by

stretching part of the sheet, and due the limited space available within the frame, other areas will shrink correspondingly. According to our distinction, we would say that manipulating a second-level visualization corresponds to manipulating the rubbersheet itself. Manipulating the first-level visualization, however, would correspond to some manipulation of what information is actually printed on the sheet.

This distinction is important, since in many cases it might be interesting to be able to perform manipulations at *both* levels of visualization. Separating the levels in this way will make the different types of interactivity clearer, and will also make it easier to account for how we can combine different focus+context visualizations with different types of information visualization techniques. In the following, some examples will be given to illustrate this.

3.2   Example 1: Structured high-level computer program

Here, the data consists of a sequence of code that represents a computer program. One way to visualize and interact with a program would be to show it as a succession of lines, indented according to their place in the program structure, in which the user can scroll up and down. The program might also be represented as uniformly sized pages of text, which the user can switch between (this would reflect the way the program would look when printed on a laser printer and might be useful when making changes according to comments written on a print-out). We might also isolate the various components of the program, such as functions and data structures, and show these as nodes in a hierarchically ordered tree; this would represent the inherent hierarchical structure of the program.

On any of these visual representations, we can then apply a focus+context visualization technique. In the case of lines of indented text we might choose to use the Generalized Fisheye View [9]. If we have text separated into uniformly-sized pages, we might use the Document Lens [24] or the Zoom Browser [12]. If we choose to have the program represented as a set of hierarchically ordered objects and functions, we might want to use the Hyperbolic Tree Browser [18] or Cone Trees [23].

Considering the interaction that might be possible in the system, users should of course be able to manipulate the data itself by making

changes in the code; these changes will directly affect the data, and will be reflected in the first-level visualization as changes in the text, indentation, hierarchical structure, etc. But users can also manipulate the focus+context visualization by means of changing the focus, increasing or decreasing the degree of magnification, etc. These changes are occurring in the second-level visualization, and will not change the actual data, only the way it is shown to the user.

3.3 Example 2: Geographical elevation data

When creating a geographical model of a certain area, the data can be described as a number of data triplets, with the two first values representing coordinates in the plane, and the third component representing the altitude. A common way to represent this type of data is to create a graphical map in two dimensions, where gray-scales or colors indicate the altitude. In some cases, however, it might be useful to use a table of the underlying numerical values, perhaps for working with the data in a spreadsheet application. Alternatively, we might create a fully 3-dimensional representation of the data, which could be rotated and viewed from different angles.

A 2-dimensional map is the most common representation used for this kind of data in focus+context visualization, as it is suited to for many distortion-based techniques, such as the Rubbersheet View [26] and the Graphical Fisheye View [25]. A very different, but still valid, type of focus+context view can be given of the tabular data with a technique such as the Table Lens [22]. In the case of a fully three-dimensional representation there may be a natural focus+context effect in the use of perspective: the parts that are close to the point of view will be more into focus than parts further away. However, for a more generalized focus+context view of 3-dimensional data, methods such as those presented in [7] might be used.

Considering the interactivity, if the map data is only to be viewed as-is, users might only interact with the information at the focus+context, i.e. second, level of visualization, by changing the focus and magnification, etc. However, if the user is going to change the data in some way, say do some manual corrections to the survey values, this interaction will take place at the first level, and be directly reflected in the table, map or other underlying visual presentation.

4 A Formal Description

We will now describe the focus+context visualization process in a more formal manner.

4.1 Visualizations

Any information visualization starts with a set of data, i.e. the information to visualize. A visual representation of this data set – or some set of data derived or constructed from this set – can be constructed based on the values or inherent structures of this data. Let us define this information visualization as:

IV ([D], V, I)

Here, **IV** is some form of information visualization in which **[D]** is the set of underlying data, **V** is how the data is presented visually, and **I** the interactivity or manipulation possible in the information visualization.

We must here distinguish between two different ways of manipulating **IV**. If **I** affects **[D]**, we can use **IV** according to **I** to manipulate the underlying data set **[D]**. This would for instance correspond to making changes to the data in a spreadsheet or a word processor. A different mode of manipulation is when **V** is affected by **I**, i.e. when a user can manipulate **IV** in order to change the way **[D]** is presented. An example of this is the case with visual information searching through dynamic queries [27], where the user can customize the visualization to show certain aspects of the data, without making any changes to the underlying data set.

4.2 Second-level Visualizations

If we instead of using **[D]** in the formula above insert some information visualization **IV**, or rather, a structure of visualizations, **[IV]**, we will have a second-level visualization, **IV'**:

IV' ([IV], V', I')

Here **IV'** is the new second-level visualization, **[IV]** is the underlying set of information visualizations, **V'** is the second-level visual compo-

nent, and **I'** is the interaction or manipulation possible in this visualization. This formula will now enable us to import any information visualization set **[IV]**, with its constraints **V** and **I** for how the structure can be visualized and changed, and apply any suitable new visualization and interaction method to this representation. Of course, in the same way as certain representations are only suited to certain types of data, **[IV]** may have to meet some constraints in order to fit into a certain second-level visualization **IV'**.

4.3   Focus+Context Visualization

We will now describe focus+context visualization as an instance of a second-level visualization **IV'**. It will take any set of information visualizations **[IV]** as its input, given that **[IV]** is compatible with the focus+context visualization technique in question. We apply a visual presentation component **V'** and some interaction **I'** that reflects the focus+context method chosen. As we incorporate some underlying information visualization **[IV]** rather than some data set **[D]**, we can focus on the aspects of **V** and **I** that are unique to focus+context techniques.

Interaction. The most notable aspect of interaction in focus+context visualization is the ability to select a focus and have the presentation changed accordingly. A convention introduced in [9] is to call the point (or rather, object) in focus **'.'** (dot). Now, we can ask how other objects in the underlying visualization **[IV]** are related to **'.'**: given a **'.'** ∈ **[IV],** how important is another object **x** ∈ **[IV]**? According to the same convention, this can be termed the *Degree if Interest*, **DoI.** In order to answer this, we have to describe the relation between **'.'** and **x,** or rather, the "distance" between **'.'** and **x**. The distance will depend on how closely the two objects are related to each other, but also of the individual properties of **x**. In [9] the function *Level of Detail* was used to establish a measure of this distance. The level of detail of an object **x** reflect where in a hierarchical structure it belongs; objects belonging to higher levels (i.e. more abstract) are said to have a lower level of detail, and hence they are more important when providing a general context. Let us use:

W (. , x)

Where **W** is the weighted distance between '**.**' and **x**, or in other words the importance of **x** given '**.**' (where '**.**' and **x** ∈ **[IV]**).

However, there are other ways of controlling how closely related two objects are as well. We might for instance let the user link objects to each other, ensuring that whenever one of them is in focus, the other one will be brought forward as well. We might also allow for other ways of weighting the objects besides using their position in a hierarchy, making it possible for individual objects to have an independent "importance factor" associated with them. Furthermore, we might want to use a tool similar to the focal length on a camera, controlling how big the difference between the focus and context should be. At one extreme the use of such a tool would imply that nothing but '**.**' is seen, and at the other that there is no difference between '**.**' and the rest, i.e. a maximal and a minimal difference between '**.**' and the rest of **[IV].**

Visualization. Given that we know which object is in focus, and how important the other objects in **[IV]** are in relation to it, we can create a visual presentation. As the available resources are limited, some constraints have to be met. This makes it useful to introduce a threshold function, **T**. **T** depends on the size of the screen, **s,** its resolution, **r,** and the computational resources, **c,** available (at least if real-time interactivity should be possible). Hence we have:

T (s, r, c)

The threshold function **T** gives a value of how close an object will have to be to '**.**' in order to be visualized. In order to determine whether a certain object **x** should be visualized or not, the weighted distance **W (. , x)** is compared with **T**:

W (. , x) > T

However, in some focus+context techniques objects are never excluded, meaning that **T** is not used to determine whether **x** should be visualized or not (or, alternatively, that **W (. , x) > T** for every '**.**' and **x** ∈ **[IV]**).

W (. , x) can also be used in order to determine which, if any, transformations of **x**'s underlying visual presentation **IV** (which is presented according to **V** in the underlying representation) should be made, e.g. distortion or scaling. For example, **x** can be given an amount of space on

the screen proportional to its distance to focus as defined by $\mathbf{W}(\,.\,,\mathbf{x})$ in which case $\mathbf{V'}$ can be a simple scaling of the image produced by $\mathbf{V}$. $\mathbf{W}$ can also be used to determine where to display $\mathbf{x}$ in relation to '.', representing $\mathbf{W}$ with actual distance between objects on the screen.

Besides functions depending on '.' and $\mathbf{W}(\,.\,,\mathbf{x})$, transformations of the underlying representations and rules for screen layout can also be applied. For instance, structural aspects of **[IV]** can be used to determine where on the screen a certain object should be placed. If the objects in **[IV]** are ordered sequentially, say, as the pages in a book, we might want them to be ordered in the same way on the screen, whereas if **[IV]** is presented hierarchically, we would want the focus+context presentation to reflect this accordingly.

5 Applying the Framework

Having defined the formal framework, we can now use it to describe some of the examples presented earlier.

Considering the first example, the structured computer program, we have one set of data that is the code being edited, which we can term **[C]**. We can then choose to have some interactive representations of it: a line-based representation, or one based on discrete uniformly-sized pages of text, or one based on a hierarchically ordered set of components. Let us call them $\mathbf{CV_L}$ (line-based code visualization), $\mathbf{CV_P}$ (page-based), and $\mathbf{CV_H}$ (hierarchical), respectively. Examining the components $\mathbf{I}$ and $\mathbf{V}$ of each representation, we see that the visual component $\mathbf{V}$ in the first case is a long sequence of lines of code, in the second it is a number of sequentially ordered pages of equal size, and in the third $\mathbf{V}$ is a number of differently sized chunks of code each representing a logical unit of some sort, presented in a tree structure. Similarly, in the first case $\mathbf{I}$ allows us to move up and down in the sequence of lines; in the second, it will allow us to switch back and forth between discrete pages of code; and in the third, it allows us to navigate the hierarchical structure of the program. If we term these components $\mathbf{V_L}$ (line), $\mathbf{V_P}$ (page) and $\mathbf{V_H}$ (hierarchy), and $\mathbf{I_L}$, $\mathbf{I_P}$, and $\mathbf{I_H}$, respectively, we have the following formulas:

$$\mathbf{CV_L = IV\ ([C], V_L, I_L)}\quad \textbf{(line-based visualization)}$$

$$\mathbf{CV_P = IV\ ([C],\ V_P,\ I_P)}\quad \textbf{(page-based)}$$

$$\mathbf{CV_H = IV\ ([C],\ V_H,\ I_H)}\ \textbf{(hierarchical)}$$

We can now insert these representations into a focus+context visualization. Common for all of these will be that the $\mathbf{I}$ component will allow the user to move the focal point, '.', in some way. In the Generalized Fisheye View, this will be through focusing on a single line; in the Document Lens and The Zoom Browser we can focus on a single page; and the in the Hyperbolic Tree and Cone Tree, we can move a certain point in the hierarchy into focus. These interactions, which we can term $\mathbf{I_L}$' (line-based interaction), $\mathbf{I_P}$' (page-based) $\mathbf{I_H}$' (hierarchical), respectively, correspond directly to the interactive components of the first-level representation.

The visual component $\mathbf{V}$' in the various cases has these properties: In the Generalized Fisheye View, only certain lines of code will be shown according to their degree-of-interest, with most detail being shown nearest to the focus; this we will term $\mathbf{V_L}$'$_{\mathbf{DoI}}$ (line-based degree-of-interest view). In the Document Lens the pages surrounding the focus will be distorted according to the combined perspective and optical metaphor used, but will keep their relative position. This we can call $\mathbf{V_P}$'$_{\mathbf{F}}$ (page-based focus+context view with fixed position). With the Zoom Browser, all surrounding pages will be shrunk to the same size, and re-arranged sequentially according to the browser's left-to-right, top-to-bottom convention; this we call $\mathbf{V_P}$'$_{\mathbf{S}}$ (page-based view with sequential position). Finally, in the Hyperbolic Tree Browser and Cone Trees, the act of focusing on one component will affect how the other components are shown according to their place in the hierarchy, so that components farther away in the hierarchy will be less visible, with close objects more visible. This we will call $\mathbf{V_H}$'$_{\mathbf{H}}$ (hierarchical view based on hyperbolic geometry) and $\mathbf{V_H}$'$_{\mathbf{3D}}$ (hierarchical view based on 3D-perspective), respectively.

We can now describe any of the focus+context applications in this example in a formal way. For instance, the Generalized Fisheye view (let us call it $\mathbf{GF}$) becomes:

$$\mathbf{GF = IV'\ ([CV_L],\ V_L'}_{\mathbf{DoI}},\ \mathbf{I_L')}$$

In the same way, the Hyperbolic Tree (**HT**) used on our hierarchically ordered program becomes:

$$\mathbf{HT = IV' \,([CV_H],\ V_H{'}_H,\ I_H{'})}$$

Using Cone Trees (**CT**) on the hierarchical ordering gives us a similar formula:

$$\mathbf{CT = IV' \,([CV_H],\ V_H{'}_{3D},\ I_H{'})}$$

The other focus+context examples can be constructed according to the same principles.

We can also do some novel combinations. Say that we want to apply the Hyperbolic Tree view to a set of uniformly-sized sequential pages. Since the only structure we have access to is the discrete pages in sequential order, $\mathbf{I_P}$, we will have to base the interaction on this, but the visualization can still be done using hyperbolic geometry. Let us call this new Hyperbolic Tree variant **HT**:

$$\mathbf{HT_P = IV' \,([CV_P],\ V_H{'}_H,\ I_P{'})}$$

Since the visualization is designed to reflect a hierarchical structure, **HT'** might not be of much practical use, but the important point is that such novel applications can be constructed in this framework.

Similarly, returning to the map example, we may term the underlying geographical data **[G]**. If we choose to represent it as a static 2-dimensional map, **M,** we may have a visual component $\mathbf{V_{M2D}}$ (2-dimensional map) but no interaction component (resulting in **I** being empty). We can then apply, say, a Rubbersheet View to this map, with the visual component being that of rubbersheet deformation, $\mathbf{V_R}$, and the interactive component being that of rubbersheet interaction, $\mathbf{I_R}$. The Rubbersheet View (**RV**) visualization of a static map would then be:

$$\mathbf{RV = ([M],\ V_R,\ I_R)}$$

Where $\mathbf{M = ([G],\ V_M,\ I),}$ and **I** is empty. However, we might want to have an interactive rather than a static map as first-level representation of **[G]**. For instance, if we want to have a zoomable map, being able to zoom in on certain parts for further visualization in the Rubbersheet view, we may have $\mathbf{M_Z = ([G],\ V_M,\ I_Z)}$, if $\mathbf{I_Z}$ is the zooming interaction

and M_Z is the resulting zooming representation of the map. This can then be inserted in the Rubbersheet view, resulting in a new variant:

$$RV_Z = ([M_Z], V_R, I_R)$$

An interesting scenario would be to add some more complex interaction to the first-level representation, say a set of dynamic query sliders [27] to facilitate advanced visual data retrieval. We would then insert the interaction I_{DQ} for the dynamic query searching, getting the resulting dynamic query-based map visualization M_{DQ}. By applying a Rubbersheet view we would then get a focus+context application which included dynamic query searching of the map data:

$$RV_{DQ} = ([M_{DQ}], V_R, I_D)$$

This might in fact be quite a useful application, since it will combine an advanced visual query method with the detail and overview supported by the Rubbersheet. Thus, the formal system has been shown to handle both existing focus+context applications, and novel combinations of first- and second-level visualizations.

6   A Software Package Supporting the Model

As we have seen, it is possible to generate different focus+context visualizations given the same underlying representation, or to apply the same focus+context visualization to a number of different representations, by varying the parameters described in the theoretical framework. This property of the formal description makes it suitable for implementation as a general software platform. We have constructed such a software package, to support the creation of focus+context visualizations of information visualizations consisting of sequentially ordered discrete visual objects. The reason for this choice of underlying visualization is that the package grew out of our work with flip zooming [11, 12], which was developed specifically for this type of visualizations. However, the implementation of a general software package has allowed us to implement some quite novel variations of the original flip zoom concepts.

6.1 A Discrete Focus + Context Software Package

The package was constructed using the Java Abstract Window Toolkit [1]. It is based on two types of Java classes: *f+c (focus+context) components* and *f+c containers*, corresponding to **IV** and **IV'** respectively. An f+c component is based on a standard Java window component, with the added functionality needed to interface with a focus+context visualization. In terms of the formal description presented above, components must provide ways to facilitate event handling related to the interaction **I'** given by a higher-level visualization **IV'**. The **V** and **I** portions of the components provide the painting of the component on the screen, and the handling of input from keyboard and mouse, for instance to facilitate manipulation of the underlying data set **[D]**.

The f+c components are stored within f+c containers, in the same way as **[IV]** is used in **IV'**. An f+c container is a Java subclass of the f+c component class, meaning that it inherits the properties of the component and must facilitate the same functionality. An advantage of this is that it is possible to insert an f+c container into another f+c container, making higher-order visualizations possible. Further functionality is needed in order to support the focus+context visualization; most notably, the containers interaction portion **I'** has to allow for sequential transversal and the random access of focus objects.

The visualization **V'** consists of two parts: The *f+c layout manager* and the *f+c visualizer*. The layout manager, which handles how the components are placed on the screen area, can be implemented according to a number of different strategies, giving rise to a number of different presentation styles. It determines the size and position of the components and provides methods for how to change the layout when setting, changing and losing focus, or when objects in **[IV]** are inserted or removed during execution. The actual drawing of the components is done by the f+c visualizer, which has access to the different visualization functions **V** in the underlying visualizations in **[IV]**.

6.2 Examples of different implementations

We have used the software framework to implement a number of sample applications. In the following, we will briefly describe some of these,
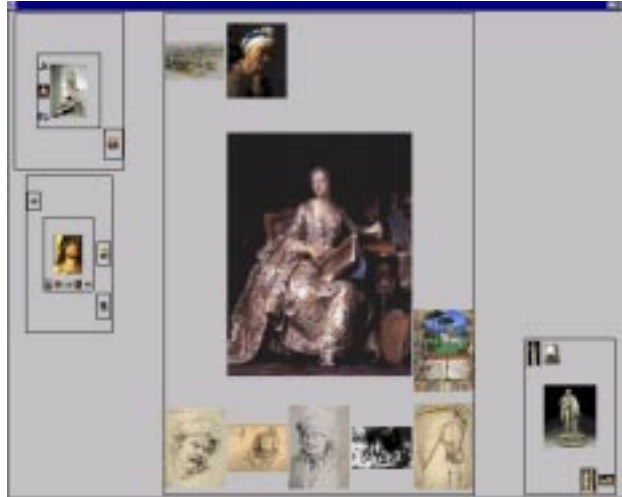
Fig. 1. The Hierarchical Image Browser

focusing on how **IV** and **IV'** are related to each other. (More details on the applications can be found in the references.)

The Hierarchical Image Browser. The *Hierarchical Image Browser* [13] was designed to explore the possibilities of using hierarchies to present large image sets in a structured way (see **Figure 1**). The hierarchies might for instance reflect the way art is exhibited in a museum, i.e. being placed in different rooms, sections and floors according to the types of paintings. The images in the set **[IV]** were ordered into containers **IV'** according to their placement in the hierarchy. Further, these containers were ordered in higher level containers **IV''**, **IV'''**, etc., according to the hierarchical structure. This application shows how the general software framework allowed us to insert focus+context visualizations into higher-level focus+context visualizations, thus reflecting the general nature of the theoretical framework.

The Digital Variants Browser. Developed as an aid to literature researchers, the *Digital Variants* application [4] presented several versions of one text to facilitate comparative studies (see **Figure 2**). The application accommodated a number of document variants **IV,** each of
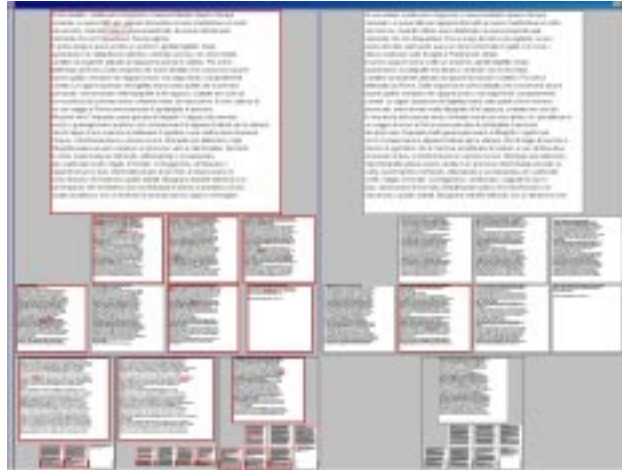
Fig. 2. The Digital Variants Browser; a total of
six documents are shown, two are in focus

which was presented in a focus+context display **IV'**. This set of
focus+context visualizations **[IV']** was then visualized in a third-level
focus+context visualization **IV"** of slightly different sort, namely one
that allowed for two simultaneous foci, facilitating the comparison of
two texts. This application shows how we could use the software frame-
work to create second- and third-level focus+context visualizations with
slightly different interactive and visual properties.

The WEST Browser. The WEST browser, a *WE*b browser for *S*mall
*T*erminals [5], was developed for use on small mobile devices, such as
Personal Digital Assistants (see **Figure 3**). Due to the limitations in dis-
play area (160 x 160 pixels) and computational power, both the space
factor **s** and the computational factor **c**, put constraints on the visualiza-
tion. To solve these problems, webpages were pre-processed in a num-
ber of steps to create a suitable structure **[IV]**. First, a web page was
stripped of banners and divided into a number of small chunks, *cards*,
each which would fit into the allowed screen space. The cards were then
ordered in a hierarchical structure with no more than seven children to
any node. All images in the original web page were scaled to the appro-
priate size and saved in the representation **[IV]**. Further, each of the
cards was analyzed in order to find links and keywords. These were

Fig. 3. The WEST Browser allows for several different views of the same web-page source

used as complementary structures of the webpage in **[IV]**. Thus, the preprocessing delivered three sets of **[IV]**: one based on the graphical look of the cards, one based on the extracted keywords and one based on the links.

The interface **I'** of the WEST browser facilitated navigation between the different levels of cards representing one webpage, but also the traditional functionality **I** associated with a web browser, such as the ability to follow links and use a history list. The user could also switch between three views: normal webpage, keyword view and link view, thus visualizing different components of **[IV]** in the same higher-order visualization **IV'**. This application shows how the framework allowed us to construct a complex interactive visualization of several different underlying visualizations.

7 Discussion and Future Work

In this paper, we first presented arguments for separating focus+context visualizations into first- and second-level visualizations, supported by some intuitive examples. We then presented a formal framework for describing properties of such aggregated visualizations and the relations between them. This enabled us to describe our initial examples in a formal way, thus validating the formal framework. We showed that the

Fig. 4. The Focus + Context Desktop, incorporating several different applications.

framework allowed us to construct some novel combinations of first- and second-level information visualizations. We also described some work with a general software package based on the formal framework, including example applications that uses hierarchies of focus+context visualizations and multiple underlying visualizations.

We can now see that according to our formal description, any **IV** that fulfils the constraints posed by **IV'** can be incorporated into **[IV]**. This means that we can incorporate any information visualization **IV** into any higher-level visualization **IV'**. This opens a lot of interesting possibilities: there is for instance nothing to stop us from applying several focus+context visualizations **IV, IV', IV''**, etc. to each other. As we saw with the hierarchical image browser and the Digital Variants browser, this can in fact be a very useful technique for combining different types of views or building a hierarchical visualization**.**

In the software package, we also have the possibility of using different types of applications within a f+c container as long as they fulfil the specified criteria for being a f+c component. One example of such an application is the *Focus+Context Desktop* (see **Figure 4**), which incorporates any application displayed in a Java window, including web browsers, web-cameras, file directory browsers and telnet clients, into a common workspace based on focus+context visualization (similar sys-

tems include [3, 15]). Future work should include evaluating such systems, as well as further experiments with nested focus+context visualizations, and applications that have heterogeneous types of underlying visualizations

The framework given in this article is not limited to focus+context visualizations, and it should be possible to use it to describe and construct many other types of interesting higher-level visualizations. Similarly, it should be possible to construct a software framework that supports other types of visualizations apart from focus+context techniques. (As we have seen, the Java language is quite suitable for the construction of such software.) However, we need to better understand the properties of the visualization components, (**V, V'**, etc.) and the interaction components (**I, I'**, etc). In particular, if we could isolate the necessary properties required for a certain higher-level visual component **V'** and interactive component **I'** to be compatible with the lower-order **V** and **I**, we will be able to state more clearly whether a certain combination of visualizations is likely to be practically useful or not. For instance, in the example section, we gave only an intuitive motivation for why Hyperbolic Trees might not be well suited to visualizing sequential data; if such relations could be expressed more formally, the usefulness of the framework should be increased quite significantly.

If extended in such a way, the framework might allow us to better explore the properties of novel visualizations even before they are implemented. It might provide answers to questions such as: What focus+context visualizations are best suited to a specific underlying visualization? How can different visualizations be combined in a focus+context visualization? How does the interactivity of a underlying visualization affect a focus+context visualization and vice versa? Our hope is that by making the distinction between different levels of visualization explicit, and by introducing a formal system that supports this notion, new possibilities within the design space of both focus+context techniques and information visualization in general will become available.

8   Acknowledgements

9   References

1. Arnold, K. and Gosling, J. *The Java™ Programming Language,* Second Edition, Addison-Wesley, 1998.
2. Bartram, L., Ho, A., Dill, J., Henigman, F., The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces, in Proceedings of ACM UIST '95, pp. 207-215, ACM Press, 1995.
3. Bederson, B., Hollan, J., Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *Proceedings of ACM UIST '94,* ACM Press, 1994.
4. Björk, S. and Holmquist, L.E.: The Digital Variants Browser: An explorative tool for literature studies. *Proceedings of Computers, Literature and Philology,* Edinburgh, UK, 1998. (To appear)
5. Björk, S. and Redström, J. An Alternative to Scrollbars on Small Screens. *Extended Abstracts of CHI '99,* ACM Press, 1999.
6. Card, S.K., Mackinlay, J.D and Shneiderman, B. Information Visualization. In Card, S.K., Mackinlay, J.D and Shneiderman, B. (eds.) *Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers, San Francisco, California, pp. 1-34, 1999.
7. Carpendale, M.S.T., Coperthwaite, D.J. and Fracchia, F.D. Extending Distortion Viewing from 2D to 3D. *IEEE Computer Graphics and Applications,* July August, 1997.
8. Furnas, G.W. The FISHEYE View: A New Look at Structured Files. Bellcore Technical Report, 1981. Reprinted in: Card, S.K., Mackinlay, J.D and Shneiderman, B. *Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.
9. Furnas, G.W. Generalized Fisheye Views, *Proceedings of CHI '86,* pp. 16-23, ACM Press, 1986.
10. Furnas, G.W and Bederson, B.B. Space-Scale Diagrams: Understanding Multiscale Interfaces. *Proceedings of CHI '95,* ACM Press, 1995.
11. Holmquist, L.E. Focus+Context Visualization with Flip Zooming and the Zoom Browser. *Extended Abstracts of CHI '97,* ACM Press, 1997.
12. Holmquist, L.E. The Zoom Browser: Showing Simultaneous Detail and Overview in Large Documents. *Human IT,* vol. 2 no. 3, ITH, Borås, Sweden, 1998.
13. Holmquist, L.E. and Björk, S. A Hierarchical Focus + Context Method for Image Browsing. *SIGGRAPH '98 Sketches and Applications,* ACM Press, 1998.

14. Kadmon, N., and Shlomi, E. A polyfocal projection for statistical surfaces. *Cartograph,* J. 15, 1, 36-40, 1978.

15. Kandogan, E., Shneiderman, B., Elastic Windows: Evaluation of Multi-Window Operations. *Proceedings of CHI '97,* pp. 250-257, ACM Press, 1997.

16. Keahey, T. and Robertson, E.L. Non-Linear Magnification Fields. *Proceedings of IEEE Visualization '97, Information Visualization Symposium,* IEEE Press, 1997.

17. Keahey, T. The Generalized Detail-In-Context Problem. *Proceedings of IEEE Visualization '98, Information Visualization Symposium,* IEEE Press, 1998.

18. Lamping, J., Rao, R., Pirolli, P., A focus+context technique based on hyperbolic geometry for viewing large hierarchies. *Proceedings of CHI '95,* ACM Press, 1995.

19. Leung, Y.K, Apperley, M.D, A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction,* vol. 1 no 2, pp. 126-160, 1994.

20. Mackinlay, J. D., Robertson, G. G., Card, S. K, The Perspective Wall: Detail and Context Smoothly Integrated. *Proceedings of CHI '91,* pp. 173-179, ACM Press, 1991.

21. Plaisant, C., Milash, B., Rose, A. Widoff, S., and Shneiderman, B. Lifelines: Visualizing Personal Histories. *Proceedings of CHI '96,* ACM Press, 1996.

22. Rao, R. and Card, S.K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. *Proceedings of CHI '94,* ACM Press, 1994.

23. Robertson, G.G., Mackinlay, J.D. and Card, S.K. Cone Trees: Animated 3D Visualizations of Hierarchical Information. *Proceedings of CHI '91,* ACM Press, 1991.

24. Robertson, G.G., Mackinlay, J.D., The Document Lens. *Proceedings of UIST '93,* pp. 101-108, ACM Press, 1993.

25. Sarkar, M. and Brown, M.H. Graphical Fisheye Views. *Communications of the ACM,* vol. 37 no. 12, pp. 73-84, 1994.

26. Sarkar M., Snibbe, S.S., Tversky, O.J. and Reiss, S.P., Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. *Proceedings of ACM UIST '93,* pp. 81-91, ACM Press, 1993.

27. Shneiderman, B. Dynamic Queries for Visual Information Seeking. *IEEE Software,* 11(6), 70-77, 1994.

28. Spence, R. A taxonomy of graphical presentation *INTERACT '93 and CHI '93 conference companion,* pp. 113-114, ACM Press, 1993.

29. Spence, R., Apperley, M., Data base navigation: an office environment for the professional. *Behavior and Information Technology,* vol. 1 no. 1, pp. 43-54, 1982.

30. Tweedie, L. Characterizing Interactive Externalizations. *Proceedings of CHI '97,* ACM Press, 1997.

Redefining the Focus and Context of Focus+Context Visualizations

Staffan Björk & Johan Redström, *PLAY, The Interactive Institute*

Abstract. The increasing diversity of computers, especially among small mobile devices such as mobile phones and PDAs, raise new questions about information visualization techniques developed for the desktop computer. Using a series of examples ranging from applications for ordinary desktop displays to web-browsers and other applications for PDAs, we describe how a focus+context technique, Flip Zooming, is changed due to the situation it is used in. Based on these examples, we discuss how the use of "focus" and "context" in focus+context techniques change in order to fit new areas of use for information visualization. By doing so, we hope to enrich the understanding of how focus+context techniques as a whole may be further developed.

1   Introduction

Computers are now beginning to be used in almost every kind of situation. Especially, mobile information technologies such as PDAs and cellular phones have recently introduced a major break-through of computer use in everyday situations. These new devices are rapidly becoming more powerful, matching the computational power and memory storage of desktop computers only ten years ago. However, they are used in situations were computers have not been used before, and are limited by small displays. While aspects of information technology design such as having an eye-catching or trendy exterior are rapidly becoming more important, there is still much work to be done on how to develop interaction design such as information visualizations that fit the many different situations people want to use them in. This is likely to be an important issue since the recent success of new and most notably mobile forms of computing indicate that there is a great interest for "off the desktop"-technology among consumers. If this interest corresponds to a real need among users, it is not unlikely that this use may constitute the majority of all computer use in the near future.

The approach to modify computers and their software to the environment they are used in, has been labeled "situated computing". As the term situated computing may be given many interpretations, and the research community has yet to give an explicit definition of the term, we will here stay with a rudimentary notion of situated technology as technology developed for a specific situation or setting with the aim to make it more transparent in relation to a task or situation. This implies not only that interaction

designers have to, for instance, customize the information visualization as such, but also that they have to choose the proper platform for presenting the information, may it be an ordinary desktop display, a wall-sized information board or a PDA.

As an illustration of the challenges and constraints posed by situated computing, we will present experiences on how a focus+context visualization technique had to be adapted for various situations, including use on PDAs. The aim of this paper is to highlight some of the implicit presumptions of focus+context techniques we have identified while working with Flip Zooming. Even though these presumptions also can be found in other researchers' work, they have become obvious to us when developing the technique in new application areas. To illustrate the presumptions, we first present a number of Flip Zooming applications developed for different areas of use, in which the necessary adaptations are identified and described. We then discuss ways of re-interpreting the terms focus and context in focus+context techniques.

2   Focus+Context Visualization

The basic idea with focus+context visualizations is to enable users to have the object of primary interest presented in detail but having an overview or a context available at the same time. Following the early work of Furnas [11], Spence & Apperley [30] and others, a number of visualization strategies have been developed. The *Graphical Fisheye Views* [24] and the *Rubbersheet View* [25] display two-dimensional maps and images using graphical distortions. The *Perspective Wall* [19] and the *Document Lens* [23] make use of perspectives to give a combined overview and detail presentation. The *Table Lens* [20] used a spreadsheet-like presentation to visualize information. Techniques developed to visualize graphs and hierarchies include the *Continuous Zoom* [1], the *Hyperbolic Tree Browser* [17], and *Cone Trees* [22].

In some ways, the early work on focus+context visualization was "situated". For instance, the Fisheye View [10] by Furnas used the structural properties of programming languages to support the many different levels a programmer has to work on simultaneously. Further, the BiFocal Display [28] by Spence and Apperley was designed to meet the needs of the "office professional" who has to deal with a number of different kinds of information more or less simultaneously. This work opened up for a perspective on "context" as something that well may be outside the document or application in focus. Since then, however, most focus+context techniques have been based on a rather static notion of what the context should be, often leaving it at representing the pages surrounding the page the user is reading in a long document.

2.1   Definitions

In order to be able to use different situations of usage to discuss and redefine focus+context visualization, we first need to clarify what focus+context visualizations are. As an introduction, it might also be useful to see how the terms "focus" and "context" are defined in general. "Focus" has been defined as *a center of activity, attrac-*

tion, or attention [9], and *a position, or condition, of sharp definition of an image* [8]. "Context" has been defined as *the interrelated conditions in which something exists or occurs* [9] and *the parts of a discourse or treatise that precede and follow a special passage and may fix its true meaning* [8].

Work to classify and categorize the various focus+context techniques developed have used a number of analogies including *Space-scale Diagrams* [12], *Rubbersheets* [18], *non-linear magnification Fields* [16], *multiple dimensions of transformation* [29], and *higher-order visualizations* [5]. Tweedie gives a description of information visualization techniques as *Interactive Externalizations* [31]. For a more complete categorization of both focus+context and other information visualization techniques, see [7].

In case there exists a definition of focus+context visualization that researchers agree upon, it is implicit in the literature. However, the following description of focus+context techniques comes rather close to being such a definition:

> "[focus+context] start from three premises: First, the user needs both overview (context) and detail information (focus) simultaneously. Second, information needed in the overview may be different from that needed in detail. Third, these two types of information can be combined within a single (dynamic) display, much as in human vision." [7, p. 307]

Initial Remarks. Assuming that the description from [7], is an accurate description of how researchers define focus+context, one can make some initial remarks on what the premises are. First, the users requirements are described as needs of information, and that there is a need to have access to information at hand on at least two levels of detail simultaneously. There is also a more implicit premise that the information visualization will provide both of them. This can be contrasted with a situation where sources of information outside the computer provides contextual information to something displayed on the screen or when the computer supports the user with contextual information about something outside the computer. By designing information visualizations that provide both levels of detail, one makes an implicit assumption that the computer will have the users, more or less, complete attention.

Second, the fact that the information needed in the overview and in detail may differ, does not only imply that the actual presentation may be different (which is the case in most focus+context visualizations), but that the very kind of information may be different in the two cases. Finally, we can see that the notion of that both focus and contextual information can be combined within a single display, does not imply that it is the best solution at all times. However, neither do many focus+context techniques allow the user to choose between having a focus-only view or having a focus+context view, nor do many of them make use of more than one display.

Looking at the focus+context visualization techniques that have been developed, there seems to be some implicit requirements and limitations. Even though we do not think that any of these are due to the premises in the "definition" used above, they can be explained by using the definition as a basis. For the areas of use that these techniques have been developed, these limitations are often motivated, but in order for

focus+context techniques to be applicable in other types of situations, it may be necessary to modify or abandon these limitations.

3   Applications

In the follow examples, we will show how the notion of focus and context changes as a visualization technique is used in a number of different applications developed to support a variety of tasks. The technique in question is Flip Zooming. Briefly, the Flip Zooming technique is an information visualization technique that presents discrete and sequential information in a number of tiles. These tiles are presented in a left-to-right, top-to-bottom fashion that maintains the sequential structure of the information and allows the user to select one tile as the focus. This focus is placed in the center of the display area and is given proportionally more screen space. The other tiles are moved so that the sequential ordering is maintained.

The examples are given in the chronological order they were developed. This should not, however, be seen as a description of a kind of evolution where the later examples are superior to the previous examples. Even though Flip Zooming has been refined during the whole development, the later examples mainly differ in terms of their area of use and not in the perfection of a technique.
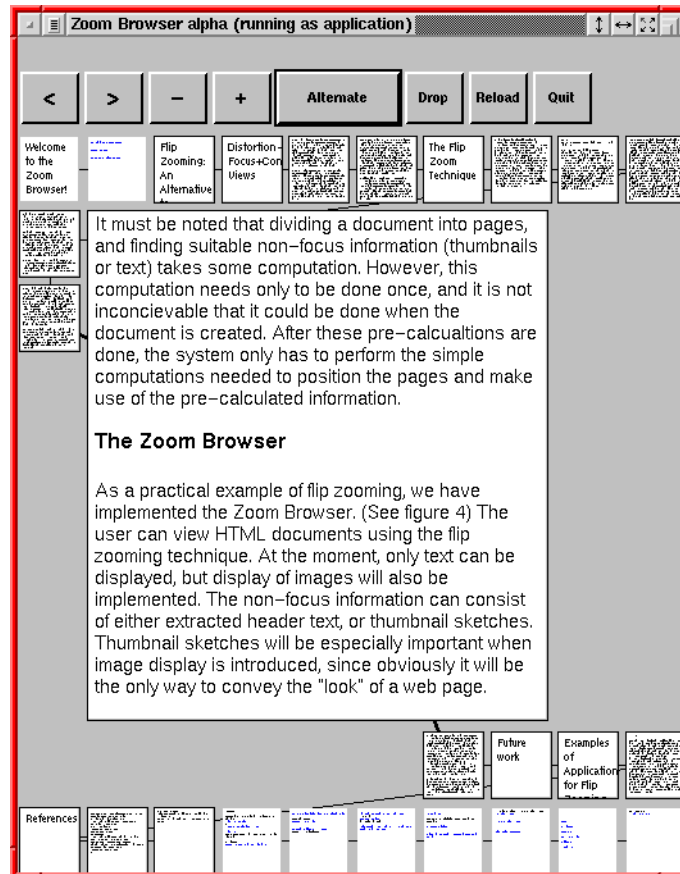
Fig. 1. The Zoom Browser.

3.1 The Zoom Browser

The first application to use the Flip Zooming technique was the Zoom Browser [13]. In this web browser, each web page is split into a suitable number of tiles and the tiles constituting a web page are linked together by a line (see figure 1). The user can gain detailed information about a tile by selecting it. New tiles are be added to the visualization by following the hyperlinks in the tiles.

The Zoom Browser conforms to the basic notions of focus+context techniques. It provides the user with a focus and a context, the presentation of information in the focus tile and in the context tiles differs, and the two types of presentation are combined within a single display. However, it offers the user the possibility to view the context information in three different ways: as thumbnails, as summaries, or as a mixture of both. Further, the user can choose to not have any focus, creating a view where all tiles are given an equal amount of screen space.
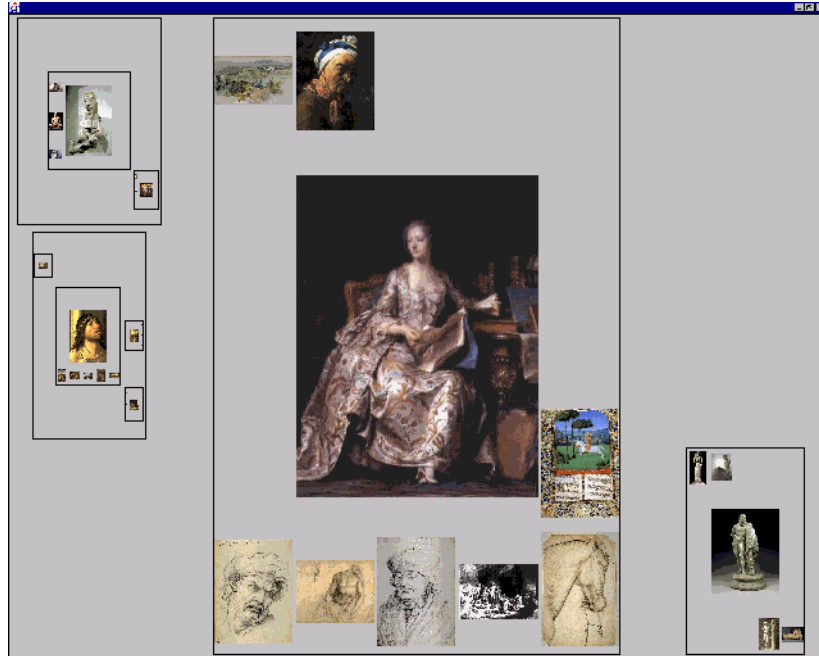
Fig. 2. The Hierarchical Image Browser.

3.2   The Hierarchical Image Browser

One noteworthy feature of Flip Zooming is that it allows for hierarchical information visualizations [2], i.e., that Flip Zooming visualizations can be visualized within one another. In the Hierarchical Image Browser [14], the Flip Zooming technique was used to present images of paintings and sculptures. The images were placed in Flip Zooming visualizations according to their style, and the visualizations themselves were grouped in outer Flip Zooming visualizations representing the different sections of a museum (see figure 2). Since users might want to take a closer look at the details of a painting, it is possible to make the object in focus fill the whole screen.

The Hierarchical Image Browser has the basic functionality of focus+context techniques, such as providing a focus and a context, but allows the user to manipulate the appearance of both the focus and the context by manipulating the inner visualizations. Further, since the context is divided into several levels by the use of a nestled visualization, there is a clear distinction between different levels of context, ranging from local to global. As the user can hide the context in each of the visualizations, the appearance of the context as a whole can be manipulated without changing the focus.
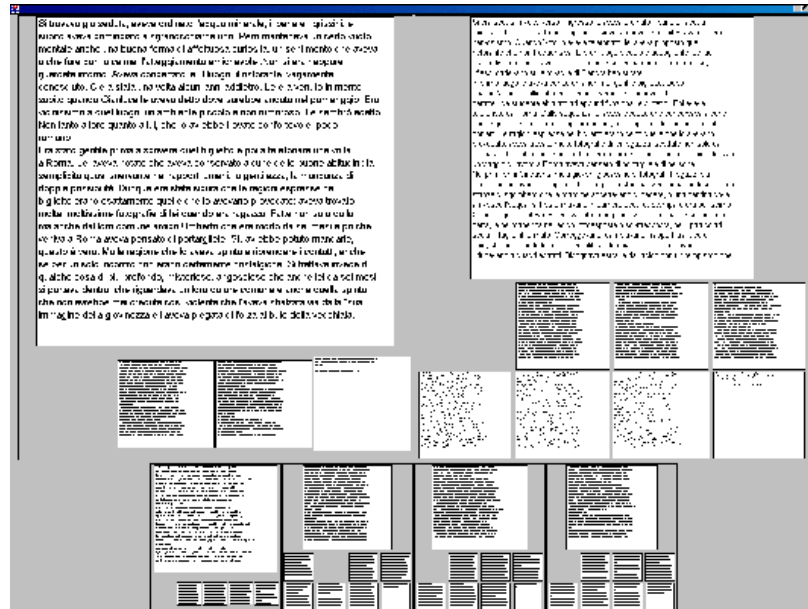
Fig. 3. The Digital Variants Browser.

3.3 The Digital Variants Viewer

The Digital Variants Viewer [3] was developed to support literature research on variants of the same texts, e.g. to compare translations, study a text's development or classify the lineage of texts. Each text is shown in a Flip Zooming visualization by dividing the text into a number of tiles comparable to a page in a book. These presentations were then placed in an outer Flip Zooming visualization, creating a two-layered visualization. To provide comparison of two of the inner visualizations, the outer visualization provided two foci placed together at the top of the display area. The user selects one of the two outer foci as locked, indicating that when an outer context is selected, the unlocked focus is changed. To ease the comparison of the two focus tiles of the inner visualization that had been chosen as foci in the outer visualization, the layout strategy of these inner visualizations had to be modified (see figure 3).

In addition to deviating from "normal" focus+context techniques in the same way as the Hierarchical Image Browser did, the Digital Variants Viewer had two foci in the outer visualization. All parallel visualizations of the documents placed the foci slightly separated from the context, but still as combined into a single visualization.

Fig. 4. The WEb browser of Small Terminals (WEST). All three views are shown in their actual resolution of 160*160 pixels. The views are, from left to right, the thumbnail view, the keyword view and the link view.

3.4 WEST

The WEST (WEb browser for Small Terminals) application [6] was designed for browsing web pages using PDAs. A basic assumption was that, due to the very small screen, users would be more focused on retrieving specific information from familiar pages than on general browsing. Similar to the Flip Zoom Browser, web pages were divided into small parts that were shown in different tiles. However, the very limited screen display, 160*160 pixels, meant that only a few of the tiles could be shown simultaneously. The tiles were ordered hierarchically, in order to provide an efficient and structured navigation. To further aid navigation, each tile could present its information in three ways: thumbnail, summary or link view (see figure 4). These different views were designed to support different ways of navigating the information. The thumbnail view supports navigation if the visual appearance of the web page is known and enables users to search for visual landmarks. The keyword view enables users to gain an overview of the content of the page. Finally, the link view was designed to support the search for specific links to other information, for instance to the individual articles on a news service.

The WEST browser was a further step way from the traditional focus+context techniques. It is a hierarchical visualization, but only shows local context as it only presents one level at a time. Further, the user can adjust the view on, i.e., change the appearance of, both focus and context depending on how the application is used.
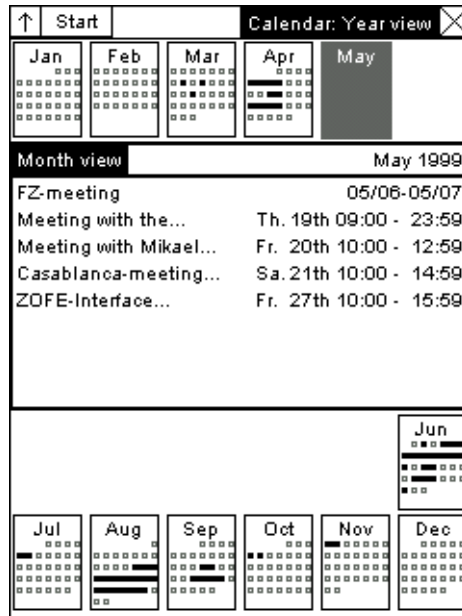
Fig. 5. PowerView application.

3.5 PowerView

The PowerView application [4] was developed to provide access to the most common information on PDAs, i.e. address entries, meetings, e-mail, and to-do lists. As PDAs often are used in public areas when the user is interacting with other people, for instance by having a conversation or a meeting, we developed the application to be used in a supportive role to provide information for other activities. In other words, it was assumed that the information presented was used to accomplish something that would take place "outside" the device. The application was built around a number of information views, in which several different types of information could be presented simultaneously. Each view was designed to support a task, e.g., to retrieve information about a meeting or a person.

   To provide the user with a relevant context once an individual piece of information had been retrieved, we introduced information links between items that enabled the user to indicate whenever two pieces of information were related even though they belonged to different information domains (see figure 5). Thus, the information links form a semantic layer on top of the information structure that enables the application to customize what is shown in the context objects depending on what is in focus. Similar to the WEST browser, it only provides local context in each view, but because of the informations links, PowerView can have heterogeneous contexts where several different types of information are mixed in one view.
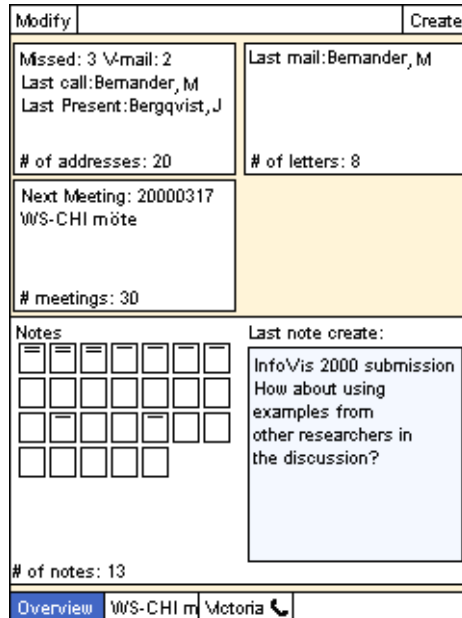
Fig. 6. PowerCom application.

3.6   PowerCom

Based on PowerView, a new application called PowerCom is being developed. The aim is to explore how functionality such as telephony can be integrated into a unified information visualization technique. One of the main ideas is to make the application provide contextual information about activities and events taking place in the users environment. Such support will include to present relevant information when the user receives a phone call, or meets someone and has a spontaneous meeting. This is achieved by making events such as phone calls trigger the generation of a new context view in the same way as when the user selects an item in order to retrieve information. The main difference in the visualization compared to PowerView, in this respect, is that PowerCom will have to handle several context views simultaneously since there might be several activities and events taking place at a time. By having these context views available, the user can easily switch between them without having to select the relevant entry and retrieve the information manually (see figure 6).

The PowerCom application goes one step further in trying to match the focus and the context of the information visualization with the user's focus of attention and the user's context. As this focus can quickly change between different activities and any activity can require several types of information to perform, the application is designed to support rapid changes of focus, and to provide a unified presentation of several different kinds of information.

4 Discussion

When developing these applications, we have each time been forced to expand, reduce or modify the Flip Zooming technique. During these modifications, we have exposed a number of preconceptions about Focus+Context techniques in general. These preconceptions should not be seen as negative in themselves; in fact, in most cases they are probably the most sensible approach. However, in some case alternative methods or variations may be advantageous or even required for a technique to be feasible, and in these cases, the preconceptions limits the usability of the technique.

4.1 Focus

Focus or Foci? A seemingly trivial observation of the name focus+context visualization is that focus is not in its plural form, foci. In most applications, this might be a natural limitation since users often only work with one object at a time. However, some activities require users to directly compare objects or switch between several different objects very rapidly. As long as a focus+context visualization technique requires explicit user input to change focus between different parts of the visualization, there always exists a small interaction overhead cost, making it more difficult to switch between activities. One way to remove or mitigate this overhead is to allow several foci at the same time or to make it very easy to switch between different foci.

In its original form, Flip Zooming did not allow users to work with several foci in parallel. However, multiple foci were enabled in the Digital Variants Viewer in order to support the comparison of documents. In the PowerCom application, multiple foci were also introduced but were not visualized simultaneously as they represented the different activities the user was switching between, but not performing at once. The RubberSheet view [25] exemplifies another variant of having multiple foci, useful when the user needs to gain more detail from several places within a visualization of continuous information, e.g. a map.

Of historical interest, it should be noted that the idea of polyfocal visualization has actually been introduced before any information visualization techniques were developed for computers [15].

Focus vs. center of Attention. Traditionally, focus+context techniques change focus when a user has selected a new point of interest, in the form a particular position on the display or a distinguishable object. However, when the user is performing an activity or task involving other applications, or even people and objects outside the computer, the focus of the visualization does not match the center of the user's attention: while the focus selected by the user in the visualization still is a focus in that it might be close to describing the user's interest, it is not the only focus that interaction designers have to acknowledge in order to create usable technology. In the case of traditional focus+context techniques that have been developed for use on a desktop computer, the difference between the focus of the visualization and the user's center of attention might not have been a great problem, since the PC is usually occupies it's users more

or less complete attention. However, when designing for mobile users, or for any setting where the environment has to be acknowledged, understanding this distinction might be important.

When working with the PowerView and PowerCom applications, we tried to design with this distinction in mind. Some consequences of this line of reasoning was that several different information types should be presented together, and that each view in the applications should focus on helping the user with a simple task. Looking at related work, Spence and Apperley designed the BiFocal Display [30] as an information visualization technique dealing with a number of different kinds of information more or less simultaneously.

4.2 Context

Differentiable context. When using a hierarchical visualization, the distinction between local and global context and the possibility of many levels in-between, is almost trivial: some pieces of information are more closely related than others, depending on where in the hierarchy they are located. This can be used to create different presentations of the context objects depending on how far from the focus they are located. This is, for instance, used in the Fisheye View [10] and in several graphical focus+context techniques based on distortion, where the distortion increases with the distance from the object in focus (c.f. [17]). However, the activity at hand might also require the user to be able to manipulate how the context is presented.

Focus+Context techniques are based on the assumption that a user needs both overview and detail information. For most activities this is true, but not necessarily all the time. Sometimes only the focus is required to complete a task, sometimes only the context, and sometimes only part of the context. Taking the example of working with pages in a book, just reading a page requires only the focus. To see how many pages the text consists of, or how the pages are distributed in chapters, the whole context is needed but to see where a page is in a chapter only requires the context of the chapter. Being able to distinguish and manipulate different parts of the context to change its appearance, including what is visible, we define as to have a differentiable context.

Due the modularity of hierarchical Flip Zooming visualizations, both the Hierarchical Image Browser and the Digital Variants Browser automatically supports a differentiable context. By selecting a full-focus view on the outermost, middle, or inner visualization, the user can effectively zoom in by removing the global context, remove the "middle" context while still maintaining a local and a global context, or remove the local context. The granularity with which the user can manipulate the context using this type of interaction is equal to the depth of the hierarchy. Looking at related work, Sarkar and Brown [24] use a function G(x) to determine where context objects should be placed, giving two examples (based on Cartesian and polar geometric transformations). By introducing variants of these functions with parameters that the user can control, these techniques could be extended to allow the user to modify the presented of the context in a fashion similar to that found in dynamic queries [26]. However, it is difficult to see how the user would have proper feedback in the visualizations without

the introduction of new visual elements, e.g. boundaries, to differentiate between the different parts of the context. Giving the user some form of control over the degree of interest (DOI) function in [11,20] would also enable user defined the context presentation, but would be most beneficial if the context is easily differentiable.

Multiple Views of context. Similar to the point made about focus, the word context is in singular form in focus+context visualizations. This is often intuitive, as the context of something is everything that is perceived to be near it. However, what is near it depends highly on the viewer's point-of-view. Changing the point-of-view may radically change the perception of the context (and the focus) without changing what is the focus of the visualization. Translated to focus+context visualizations, changing the point-of-view becomes the possibility to easily change the whole context, or the presentation of the focus and context, while maintaining the same focus.

The Zoom Browser and the WEST browser both used multiple ways of presenting the same focus and context in order to provide the user with several ways of accessing the information visualized. This was partly because the user could be assumed to need several different ways to solve a task and partly because all information in each individual tile could not be presented clearly at once. The PowerCom applications also used multiple context but for another reason. As it enables the user to rapidly switch between different activities, each represented by a selected object in the application, it is only natural that each of these has it own context. Viewed in this fashion, PowerCom can be described as an application of the personal role manager approach [27].

Although not a focus+context technique, dynamic queries [26] can be used in combination with such techniques to allow multiple views of a context. In this case, the dynamic queries would not be used to filter information but rather to select different view modes of the presentation.

Homogenous and Heterogeneous contexts. Most information visualizations only visualize one type of information, or present different types of information using the presentation visualizing the highest common denominator. The use of such a homogenous context makes it possible to use the inherent structure of the information, e.g. to sort files and contact lists alphabetically, emails and meetings chronologically, in order to create a structured presentation. By doing so, understanding the context becomes easier and navigating through the information simpler. However, many times one can find different types of information that have a common connection, e.g. an email from a person and the entry in the contact list denoting that person. Presenting such information together as a context creates a heterogeneous context.

As mentioned in the section about focus limitations, both the PowerView and PowerCom application used heterogeneous contexts. By doing this, they differ from most visualization techniques, which may show different types of information, but either merge all information into one presentation or treat all information as belonging to a more general type that incorporates all types present. In the first case, the interaction is performed by manipulating the presentation as a whole, while in the second case, all objects are interacted with in the same fashion and respond identically. By presenting a heterogeneous context, where manipulating different objects give differ-

ent results, PowerView and PowerCom allow for a more task-oriented approach to information visualization.

With the PowerCom application, the use of audible information (in the form of phone calls) was introduced as a part of the system. A phone call is quite different from other feedback from the system such as the sounds that accompany error messages, since it is not only a complement to some information already available in visual form but a channel of communication that may or may not be supported by visually presented information. By creating a system that makes use of additional perceptual modes such as hearing, more possibilities are added to what might constitute the focus and the context. This expansion can be seen as a tentative step towards what [7, p. 7] calls information perceptualization.

5   Concluding remarks

Using a number of examples from our own experiences, we have tried to illustrate how methods and principles from information visualization can be applied in novel settings and be made to fit constraints posed by situated computing. It has been our ambition to illustrate two issues that interaction designers will have to face when even more forms of computer use enters the agenda. First, to what extent existing techniques might be applied, and second, to explore what we can learn about existing techniques doing this. As for the first ambition, we have described what modifications had to be made of the Flip Zooming technique and how they turned out. As for the second, we presented a number of preconceptions or premises of focus+context visualizations that seem to have been restricting development of focus+context visualization in novel use situations. Such premises include the use of one focus only, the use of homogeneous contexts, to consider the interaction between man and computer only and not involve the users environment as a whole, to name a few.

While the applications presented here differ from more traditional focus+context visualizations in some ways, we still think that they qualify as such. While the notion of "focus" and "context" has been given partly new interpretations, this has been in order to pursue the basic ideas of focus+context visualization. In other words, in order to realize the essential ideas of focus+context techniques in new situations of computer use, one must move beyond the traditional notion of focus+context.

6   Acknowledgements

7 References

1. Bartram, L., Ho. A., Dill, J., and Henigman, F. The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces, in *Proceedings of ACM UIST '95*, pp. 207-215, ACM Press, 1995.

2. Björk, S. Hierarchical Flip Zooming: Enabling Parallel Exploration of Hierarchical Visualizations. In *Proceedings of AVI 2000*, ACM Press, 2000. (To appear)

3. Björk, S., and Holmquist, L.E. The Digital Variants Browser: An explorative tool for literature studies. In *Proceedings of Computers, Literature and Philology*, Edinburgh, UK, 1998.

4. Björk, S., Holmquist, L.E., Ljungstrand, P., and Redström, J. PowerView: Structured Access to Integrated Information on Small Screens. In *Extended Abstracts of CHI '2000*, ACM Press, 2000. (To appear)

5. Björk, S., Holmquist, L.E. and Redström, J. A Framework for Focus+Context Visualization. In *Proceedings of IEEE Information Visualization '99*, IEEE Press, 1999.

6. Björk, S., Holmquist, L.E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., and Franzén, K. WEST: A Web Browser for Small Terminals. In *Proceedings of ACM UIST '99*, ACM Press, 1999.

7. Card, S.K., Mackinlay, J.D., and Shneiderman, B. (Eds.) *Readings in Information Visualization: Using Vision to Think*, pp. 1-34, Morgan Kaufmann Publishers, San Francisco, California, 1999.

8. Davidson, G.W., Seaton, M.A., Simpson, J. (Eds.) *Concise English Dictionary*. Wordsworth Editions Ltd., Hertfordshire, UK, 1994.

9. *Encyclopedia Britannica Online*. http://www.eb.com/

10. Furnas, G.W. *The FISHEYE View: A New Look at Structured Files*. Bellcore Technical Report, 1981.

11. Furnas, G.W. Generalized Fisheye Views. In *Proceedings of CHI '86*, pp. 16-23, ACM Press, 1986.

12. Furnas, G.W., and Bederson, B.B. Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proceedings of CHI '95*, pp. 234-241, ACM Press, 1995.

13. Holmquist, L.E. The Zoom Browser: Showing Simultaneous Detail and Overview in Large Documents. In *Human IT*, Vol. 2, No. 3, pp. 131-150, ITH, Borås, Sweden, 1998.

14. Holmquist, L.E., and Björk. S. A Hierarchical Focus + Context Method for Image Browsing. In *SIGGRAPH 98 Sketches and Applications*, p. 282, ACM Press, 1998.

15. Kadmon, N., and Shlomi, E. A polyfocal projection for statistical surfaces. *Cartograph*, Vol. 15, No. 1, pp. 36-40, 1978.

16. Keahey, T., and Robertson, E.L. Non-Linear Magnification Fields. In *Proceedings of IEEE Symposium on Information Visualization '97*, IEEE Press, 1997.

17. Lamping, J., Rao, R., and Pirolli, P. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of CHI '95*, pp. 401-408, ACM Press, 1995.

18. Leung, Y.K., and Apperley, M.D. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. In *ACM Transactions on Computer-Human Interaction*, Vol. 1, No. 2, pp. 126-160, ACM Press, 1994.

19. Mackinlay, J.D., Robertson, G.G., and Card, S.K. The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings of CHI '91*, pp. 173-179, ACM Press, 1991.

20. Rao, R., and Card, S.K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In *Proceedings of CHI '94*, page 222, ACM Press, 1994.

21. Rao, R., Pedersen, J.O., Hearst, M.A., Mackinlay, J.D, Card, S.K., Masinter, L., Halvorsen, P-K., and Robertson, G.G. Rich Interaction in the Digital Library. *In Communications of the ACM*, Vol. 38, No. 4, pp. 29-39, ACM Press, 1995.

22. Robertson, G.G., Mackinlay, J.D., and Card, S.K. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of CHI '91*, pp. 189-194, ACM Press, 1991.

23. Robertson, G.G., and Mackinlay, J.D. The Document Lens. In *Proceedings of UIST '93*, pp. 101-108, ACM Press, 1993.

24. Sarkar, M., and Brown, M.H. Graphical Fisheye Views. In *Communications of the ACM*, Vol. 37, No. 12, pp. 73-84, ACM Press, 1994.

25. Sarkar, M., Snibbe, S.S., Tversky, O.J., and Reiss, S.P. Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In *Proceedings of ACM UIST '93*, pp. 81-91, ACM Press, 1993.

26. Shneiderman, B. Dynamic Queries for Visual Information Seeking. *IEEE Software*, Vol. 11, No. 6, pages 70-77, 1994.

27. Sheiderman, B., and Plaisant, C. The Future of Graphic User Interfaces: Personal Role Managers. Keynote address, in *People and Computers IX*, pp. 3-8, British Computer Society HCI'94, Cambridge University Press, 1994.

28. Spence, R. New concepts in Visual Displays. In *Proceedings of Vision and Visual Display Units*, Institute of Opthalmology, London, pp. 6-1 to 6-9, 1981.

29. Spence, R. A taxonomy of graphical presentation. In *INTERACT '93 and CHI '93 conference companion*, pp. 113-114, ACM Press, 1993.

30. Spence, R., and Apperley, M. Data base navigation: an office environment for the professional. *Behavior and Information Technology*, Vol. 1, No. 1, pp. 43-54, 1982.

31. Tweedie, L. Characterizing Interactive Externalizations. In *Proceedings of CHI '97*, pp. 375-382, ACM Press, 1997.

Note: [10,18,20,23,26,30,31] are reprinted in Card, S.K., Mackinlay, J.D., and Shneiderman, B. (Eds.) Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann Publishers, San Francisco, California, 1999.