Hierarchical Flip Zooming: Enabling Parallel Exploration of Hierarchical Visualizations

Staffan Björk
PLAY: Applied research on art and technology
Interactive Institute, Box 620, SE-405 30 Göteborg, Sweden
staffan.bjork@interactiveinstitute.se

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).

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.

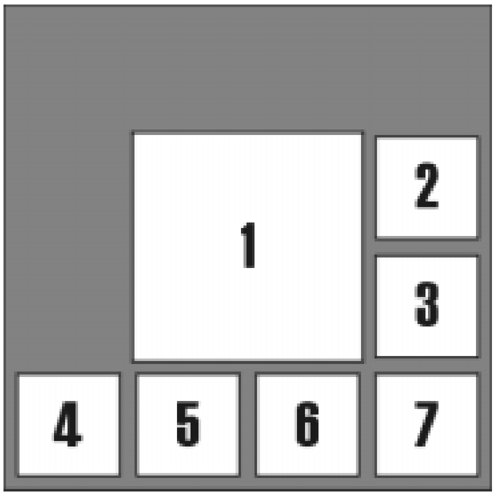

Figure 1: Flip Zooming layout with focus on the first object.

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.

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 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, next page). As these tiles are already shown as context tiles, their appearances are unchanged.

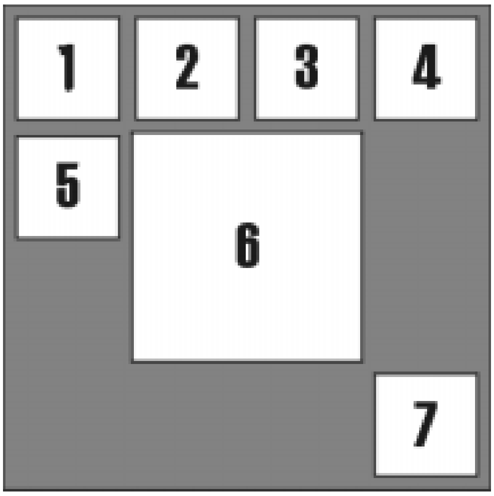

Figure 2: Flip Zooming layout with focus on the sixth object.

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 out-going link for each operator. Using only the three navigational operators, this meets Furnas' requirement of having a small number of out-going 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 cursor-less 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

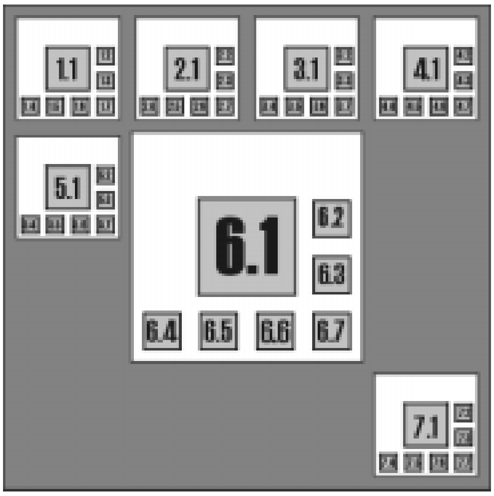

Figure 3: Hierarchical Flip Zooming of hierarchical information.

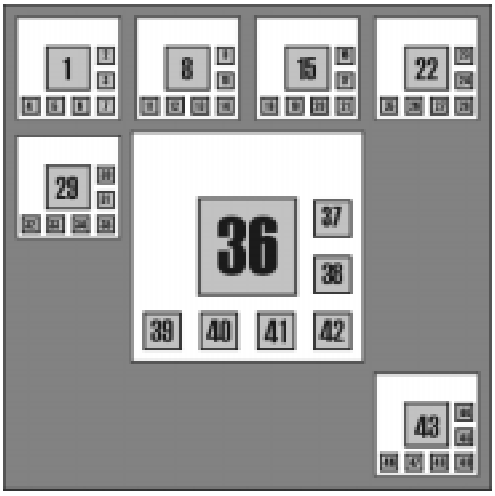

Figure 4: Hierarchical Flip Zooming of sequential information.

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), 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.

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).

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 cur-

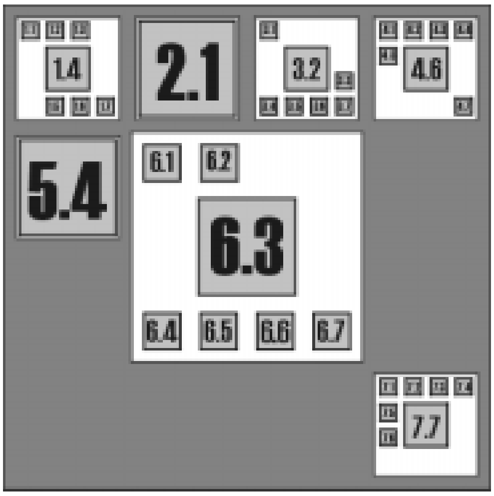

Figure 5: The Hierarchical Flip Zooming visualization after some user interaction.

sor 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 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.

Figure 6: The Hierarchical Image Browser.

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.

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.

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, next page). 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.

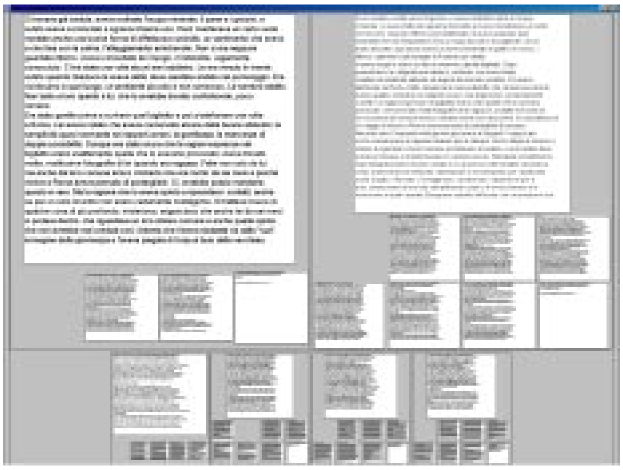

Figure 7: The Digital Variants prototype showing six variants of "The Orient-Express" by Italian writer Francesca Sanvitale.

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

Lars Erik Holmquist developed the original Flip Zooming technique, and has offer valuable insights in much of the work presented here. The work presented here was part of the project Effective Display Strategies for Small Screens, funded by SITI, the Swedish Information Technology Institute.

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.