InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures

Jing Yang, Matthew O. Ward and Elke A. Rundensteiner
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609
{yangjing,matt,rundenst}@cs.wpi.edu [*]

Abstract

Radial, space-filling (RSF) techniques for hierarchy visualization have several advantages over traditional node-link diagrams, including the ability to efficiently use the display space while effectively conveying the hierarchy structure. Several RSF systems and tools have been developed to date, each with varying degrees of support for interactive operations such as selection and navigation. In this paper, we describe what we believe to be a complete set of desirable operations on hierarchical structures. We then present InterRing, an RSF hierarchy visualization system that supports a significantly more extensive set of these operations than prior systems. In particular, InterRing supports multi-focus distortions, interactive hierarchy reconfiguration, and both semi-automated and manual selection. We show the power and utility of these and other operations, and describe our on-going efforts to evaluate their effectiveness and usability.

Keywords: Radial space-filling hierarchy visualization, multi-focus distortion, structure-based brushing.

1 Introduction

Among the 22 papers presented in InfoVis 2001, 5 papers were devoted to hierarchy visualization techniques and their evaluation. This phenomenon reveals the importance of hierarchy visualization due to the ubiquitousness of hierarchical data structures. Recent evaluations [27, 2] of existing hierarchy visualization techniques reveal that the radial, space-filling (RSF) hierarchy visualization is a promising one in that it has the advantage of efficiently using the display space while conveying the hierarchical structure better than other space-filling techniques, such as treemaps [13, 25].

However, compared to many mature hierarchy visualizations, such as those found in Microsoft Windows Explorer and other file system browsers, the RSF methods still suffer from limited navigation, selection, and modification tools. Moreover, one of the inherent drawbacks of the RSF hierarchy visualization methods is that the small slices in an RSF display are sometimes difficult to distinguish. Distortion techniques to address this issue [1, 28] have limitations, such as supporting only single focal points and reducing display space usage efficiency.

In this paper, we present InterRing, a radial, space-filling hierarchy visualization tool that addresses many of these limitations. In InterRing, we use a multi-focus + context distortion approach that is more flexible and more efficient in display space usage than existing approaches, and provide support for a wide variety of other navigation operations, such as zooming/panning, rotation, and drilling-down/rolling-up. We expand the functionality of the tool by allowing users to interactively reconfigure the hierarchy via drag and drop operations. We also provide tools for both manual and semi-automatic selection, and an assortment of strategies for providing visual feedback for these operations. In addition, we use a structure-based node coloring strategy to redundantly convey the hierarchical structure in the RSF display as well as for linking to other displays related to the hierarchical data.

The remainder of this paper is organized as follows: Section 2 describes the background of this work, Sections 3, 4 and 5 describe details of the navigation, selection, and modification tools, and section 6 presents the structure-based coloring approach in InterRing. Section 7 summarizes our work and presents open issues for future work.

2 Background

Most of recent hierarchy visualization research has concentrated on the challenge of displaying large hierarchies in a comprehensible form [7]. Among the efforts, there are improved node and link diagrams [11, 4], cone trees [5, 24] and their variations [16, 17, 29], collapsible cylindrical trees [7], botanical trees [14], radial layouts [3, 8, 32], hyperbolic layouts [19, 18, 22, 20, 21], skeletal images [12], and castles [15]. Space-filling hierarchy visualization techniques are a significant category among all the techniques in that they use display space very efficiently.

Among the space-filling approaches, there are rectangular space-filling techniques, such as treemaps and its variations [13, 25, 30, 26], and radial (also called circular) space-filling techniques [1, 6, 28]. Compared to the rectangular space-filling techniques, the radial methods have been shown to work better in revealing hierarchical structures [27, 2]. An RSF hierarchy visualization is generated using the following rules:

- Deeper nodes of the hierarchy are drawn further from the center;
- Child nodes are drawn within the arc subtended by their parents;
- The sweep angle of a leaf node is proportional to one of the node's properties or otherwise set to a uniform size;
- The sweep angle of a non-leaf node is the aggregation of all its children.

[*] This work is supported under NSF grants IIS-9732897, IRIS-9729878, and IIS-0119276.

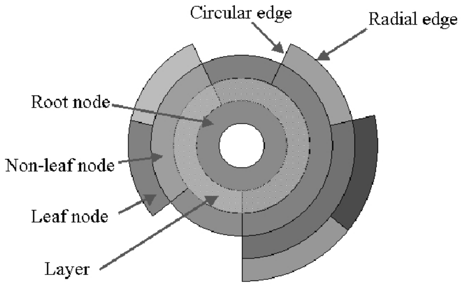


Figure 1: An RSF display with labeled components

Figure 1 shows an example of an RSF display with labels on the distinct components.

RSF techniques have an inherent drawback in that small slices can be difficult to distinguish. This drawback can be overcome by using context + focus techniques. Andrews and Heidegger's radial space-filling system [1] uses two semi-circular areas as a form of two-level *overview and detail*. Stasko and Zhang felt that this was not smooth and flexible enough in alternating between global and detailed views [28]. They proposed three distinct distortion methods to solve this problem: the angular detail method, the detail outside method, and the detail inside method. The angular detail method shrinks the entire hierarchy and pushes it aside. The selected area is then enlarged and put in the center of the display. While this looks natural to the users, it does require significantly more space. The detail outside method shrinks the entire hierarchy into the center. The selected nodes are expanded to be a new complete circular ring-shaped region around the overview. The detail inside method pushes the entire hierarchy outward and shows the selected items inside the entire hierarchy. Using this method the selected items in the overview can be viewed more clearly. However, the above distortion techniques have several limitations such as supporting only a single focal area and reducing the display space usage efficiency.

In mature hierarchical visualizations, such as Microsoft Windows Explorer, an extensive set of navigation, selection, and modification tools have been provided to users, such as drilling-down/rolling-up, copy/cut/paste/new/delete, and single-selection/multi-selection/query. Those operations are popular operations on hierarchical data structures, though few can be seen in existing RSF systems.

Our recent research efforts on very high dimensional visualization have led to the development of the Visual Hierarchical Dimension Reduction (VHDR) framework [9, 10, 31]. In VHDR, we construct a dimension hierarchy from a high dimensional dataset and generate lower dimensional (LD) spaces using clusters from this structure. A hierarchy visualization tool was needed to enable users to visualize, modify, and perform selections on the dimension hierarchy. Since the screen space available in our application for the hierarchy visualization is small (the data visualization occupies most of the screen) and the dimension hierarchy is usually large (it can contain tens or hundreds of nodes), we selected the radial space-filling technique for its strengths in space efficiency and ease of interpretation.

However, we found existing implementations of RSF displays lacked many essential capabilities. The functionality required within the VHDR framework includes:

- Multi-focus + context displays, since LD spaces can be constructed using clusters throughout the hierarchy, and users often want to examine details of several regions of the hierarchy at the same time.
- Tools to modify the hierarchy structure, since users want to adjust the automatically generated dimension hierarchy according to their knowledge of the data set and domain.
- Multiple ways of performing selection on clusters on the hierarchy, since users need to select clusters at different levels of the hierarchy to construct the LD spaces.

These requirements could not be satisfied by existing RSF techniques, although they are common tasks found in other forms of hierarchy visualization. Hence we designed and implemented InterRing, an RSF hierarchy visualization that satisfies the requirements of our VHDR framework. We believe the resulting functionality, however, is of general utility in visual explorations and interactive modification of hierarchically structured data.

The essential operations we have identified for interaction with hierarchically structured data consist of:

selection: the process of selecting some clusters from the hierarchy for further operations, such as highlighting, masking, or distortion;

reconfiguration: the process of interactively modifying the hierarchy structure;

distortion: the process of enlarging some objects in the display while maintaining the context of surrounding objects;

drill-down/roll-up: the process of exposing/hiding sub-branches of the hierarchy;

pan, zoom, and rotation: the process of modifying the focus, scale, and orientation of the display.

We have implemented instances of all of these operations in InterRing, many of which were inspired by and can be found in existing visualization systems. The multiple selection (Section 4) is an extension of the structure-based brushing found in [10]. The modification operation (Section 5) is inspired by Microsoft Windows Explorer. The multi-focus distortion (Section 3.1) is similar to the operations found in the Table Lens system [23]. The operations of drill-down/roll-up (Section 3) are inspired by the Microsoft Windows Explorer and the structure-based brushing tool found in XmdvTool [10]. The rotation operation is inspired by the cone trees [5] and other 3-D node-link tree visualization systems. In addition, the idea of the structure-based coloring (Section 6) is similar to the coloring used in hierarchical parallel coordinates [9].

We now describe the implementation of these operations in InterRing in more detail.

3 Navigation Tools

3.1 Distortion

Distortion is a procedure that results in the enlargement of selected parts of a display while reducing the screen allocation of other parts. It is useful in helping users examine details of the display, and makes other interactive operations, such as selection, easier to

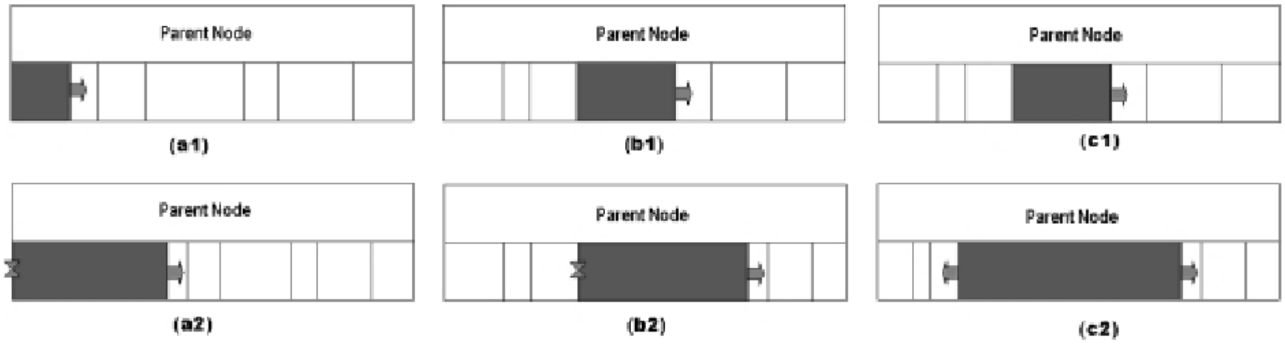


Figure 2: Circular Distortion Illustration. In (a1), the pinned node (node in red)'s non-dragged edge is located at the edge of its parent node. Thus the non-dragged edge is fixed in both one-direction and two-direction distortion (a2). The siblings between the dragged edge and the right edge of the parent node are contracted to provide space for the enlarged pinned node. (b1) and (b2) illustrate one-direction distortion. The non-dragged edge is fixed, and the siblings between the dragged edge and the right edge of the parent node are contracted to provide space for the pinned node while the siblings on the other side remain unchanged. (c1) and (c2) illustrate two-direction distortion. Both the non-dragged edge and the dragged edge are moved. Siblings on both sides are contracted to provide space for the pinned node.

perform. In Section 2, we mentioned several different distortion approaches to provide focus + context in a radial, space-filling hierarchy visualization, such as Andrews and Heidegger's [1] two semi-circular approach and Stasko and Zhang's angular detail, detail outside, and detail inside approaches [28]. Those approaches have the following disadvantages:

- There is a big visual "jump" before and after the distortion operation, that is, the display before the distortion is quite different from the display after the distortion. This disadvantage is often remedied using complex animations so that users can follow the changes.
- Space usage efficiency of the focus + context display is generally lower than the original RSF display; more screen space is needed.
- It is difficult to have multiple foci in one display.

In InterRing, we provide a new distortion approach, modeled after the Table Lens [23], addressing these drawbacks in that:

- It is easy for users to follow;
- No extra space is needed for the focus + context display;
- Multiple foci can coexist.

Our distortion approach includes circular distortion and radial distortion. When we discuss circular distortion, we assume that the thickness of all the layers of the hierarchy is fixed, and their circular edges are movable. Alternatively, when we discuss radial distortion, we assume that the circular edges of all the nodes are fixed and only the circular boundaries between hierarchy layers are movable. In practical operation, these two kinds of distortions can be performed independently of each other.

3.1.1 Circular Distortion

The basic ideas of the circular distortion are that:

- A distortion is limited to the angle range of a parent node;
- A node is increased or decreased in size by decreasing or increasing the size of its siblings;
- When the sweep angle of a non-leaf node is increased or decreased, all its children's sweep angles are also increased or decreased proportionally so that they are always in the angle range of the parent node;
- A minimum angle is set for the hierarchy, which is inversely proportional to the number of the leaf nodes contained in the hierarchy. The sweep angle of a node cannot be smaller than the minimum angle.

To help users to *snap* a node edge (select an edge of a node without needing to place the cursor accurately on it) to perform a distortion, we provide a *pin* function. After users click on a node, this node and all its ascendant nodes are highlighted and pinned in the sense that when performing a distortion on a pinned node, either one edge of it is fixed (one-direction distortion) or the center of the node is fixed (two-direction distortion). When the cursor is in a layer that contains a pinned node, it will automatically snap to one of the two circular edges of the pinned node. Thus selecting a circular edge of a pinned node is rather easy. The ascendant nodes of the clicked node are pinned together with the clicked node to provide multiple distortable nodes to users.

There are currently two circular distortion strategies in InterRing: one is *one-direction* distortion, and the other is *two-direction* distortion. In one-direction distortion, when users drag one circular edge of a pinned node (the *dragged edge*), the other circular edge of that node (the *non-dragged edge*) will be fixed, while in two-direction distortion, not only the dragged circular edge, but also the other circular edge can move.

When the non-dragged edge is located at the edge of the parent node of the pinned node, it is fixed in both one-direction distortion and two-direction distortion. The pinned node is expanded in size when the dragged edge moves away from the non-dragged edge, and contracted when the dragged edge moves towards the non-dragged edge. The siblings of the pinned node are contracted or expanded proportionally to their original sweep angles to give up or occupy the amount of space acquired or abandoned by the pinned node (see Figure 2 (a1) and (a2)).

When neither circular edges of the pinned node are located at the edge of the parent node, one-direction distortion and two-direction distortion perform differently. In one-direction distortion, the non-dragged edge is fixed and the siblings on the non-dragged edge side (*siblings of group 1*) will not be affected by the distortion. The pinned node is expanded or contracted by contracting or expanding the siblings on the dragged edge side (*siblings of group 2*, see Figure 2 (b1) and (b2)). In two-direction distortion, the dragged-edge moves the same way as in one-direction distortion. However, siblings of group 1 also contract or expand in the same ratio as the siblings of group 2, thus the non-dragged edge is also moving. In this approach, siblings of groups 1 and 2 are distorted to the same extent (see Figure 2 (c1) and (c2)).

If the dragged edge is located at the edge of the parent node of the pinned node, dragging and releasing this edge leads to the same effect as dragging and releasing its parent's corresponding edge. In either distortion approach, all the descendents of the distorted nodes are also moved and changed proportionally to fit the new angle range of their ascendants. These changes all happen gradually with the user's mouse movement and are easy to follow. To indicate the region that has been distorted to remind users that their perception and judgments may be influenced by the distortion, we provide an option of drawing borders surrounding the distorted region. The distortion borders are yellow lines and arcs in Figures 3 and 4.

Figures 3 to 6 show a hierarchy containing 677 nodes derived by clustering the Cars data set. Figure 3 compares one-direction distortion and two-direction distortion. Figure 3(a) shows the hierarchy before distortion; the cursor is on the pinned node to be enlarged (lower right part of the ring). It has two siblings. Figure 3(b) shows the consequence of one-direction distortion. The cursor is on the dragged edge. The sibling and all its descendents adjacent to the dragged edge are contracted to provide room for the pinned node. The other sibling is not affected by this distortion. In Figure 3(c), both siblings are contracted to provide room for the pinned node.

The above distortion is localized to the angle range of the parent node of the pinned node that is distorted. We can apply distortions to pinned nodes located in different areas of the InterRing display without affecting each other, or even apply distortion to a node that is a child of another node that has already been distorted. In this way we can achieve multi-focus distortion. Figure 4 shows a multi-focus distortion using two-direction distortion. Figure 4(a) shows the hierarchy before distortion; the cursor is on the first focus. In Figure 4(b) the first focus region has been enlarged and the cursor is on the second focus region. Figure 4(c) shows that the second focus region is also enlarged.

3.1.2 Radial Distortion

Similar to the circular distortion, a layer is pinned (pinned layer) before a radial distortion to help users to snap one of the radial edges of this layer. When users drag a radial edge (dragged edge) of the pinned layer, the other radial edge (non-dragged edge) of the pinned layer is fixed. The thickness of the pinned layer is expanded or contracted by contracting or expanding the thickness of the layers on the dragged edge side.

A minimum layer thickness is set. The thickness of all the layers is larger than or equal to this minimum thickness to help users distinguish different layers. The radius of the outer radial edge of the outermost layer remains unchanged during the radial distortion process. Thus the maximum radius of the RSF will not be changed by the distortion.

Figure 5 shows the use of radial distortion to examine details of the higher levels of a hierarchy. Figure 5(a) shows the hierarchy before distortion; the pinned layer is outlined in red. In Figure 5(b) the outer radial edge of the pinned layer has been pushed outwards by decreasing the thickness of its outer layers. In Figure 5(c) the inner radial edge of the pinned layer has been pushed outwards to increase the thickness of its inner layers.

Figure 6 shows the use of radial distortion to examine details of the lower levels of a hierarchy. Figure 6(a) shows the hierarchy before distortion; the pinned layer is outlined in red. In Figure 6(b) the inner radial edge of the pinned layer has been pushed inwards by decreasing the thickness of its inner layers. In Figure 6(c) the outer radial edge of the pinned layer has been pushed inwards to increase the thickness of its outer layers.

3.2 Other Navigation Tools

We have implemented many other navigation tools besides distortion in InterRing, including the following operations:

Drill-Down/Roll-Up: Drilling-down/rolling-up are used to hide/show all the descendants of a cluster by simple mouse clicking. It helps users prevent the display of branches that are not of interest for the current analysis task.

Zooming and Panning: Zooming in/zooming out and panning operations allow users to enlarge the canvas and move around to examine details of the display.

Rotation: The rotating operation is a unique and necessary operation for the circular InterRing display. In the rotation mode, users can rotate the InterRing display around its center in both directions by clicking the left mouse (anti-clockwise rotation) or right button (clockwise rotation) in the blank region of the InterRing display window. This operation helps users rotate clusters of interest to particular angles and avoids cluttering the labels of the selected clusters.

4 Selection Tools

4.1 Selection

The purpose of selection is to isolate a set of nodes in the hierarchy. These nodes can then be highlighted, masked, moved, or deleted. We provide brushing operations in InterRing to perform the selection task.

We have implemented two different brushing mechanisms. One is *simple brushing* and the other is *structure-based brushing*. Simple brushing is just a clicking operation. When users click the left button of the mouse on a node of the InterRing display, this cluster will be selected. When a user clicks it one more time, it will be unselected.

Structure-based brushing was first proposed in [10] and we adapted it for use in InterRing. It allows users to select multiple nodes at the same time. In InterRing, the structure-based brushing is performed on a node and all of its descendents, according to a certain property of the nodes in that cluster. For example, this property can be the number of leaf nodes contained in a cluster. When users click the right button of the mouse on a cluster of the InterRing display, a small dialog pops up. Users can select a threshold between 1 and the numbers of leaf nodes contained in the cluster through this dialog. We can consider the cluster and all its descendants as a tree rooted at the cluster. A structure-based selection will be done on this tree using the threshold. The selection starts from the root cluster. If its number of leaf nodes is less than or equal to the threshold, it is selected and the process stops. Otherwise this process is repeated for all its children. Using this brush, we can select approximately uniformly sized clusters and cover all the leaf nodes of the root cluster at one time. We define a leaf node as covered if either itself or one of its ancestors is selected. When the threshold equals 1, it means all the leaf nodes of the cluster will be selected. When the threshold equals the number of leaf nodes of the cluster, then this cluster itself will be selected.

Since the structure-based brushing works on a cluster, we can apply it multiple times to different clusters using different thresholds. Hence we can select nodes at different levels of detail in different regions.

The combination of the single and structure-based brushing operation makes the selection both flexible and easy to use. Users can easily switch between the two brushing operations by clicking the left or right mouse button.

Figure 7 gives an example of a series of structure-based brushing operations. First, structure-based brushing was applied to the root node and the threshold is set to 6 (Figure 7 (a)). Figure 7 (b) shows the result of this selection. Then, we applied structure-based brushing again on a region that we want to have a lower level of detail and set a large threshold. Figure 7 (d) shows the selection result of the two brushing operations. Part of the hierarchy is selected at a lower level of detail than the other parts of the hierarchy.

The structure-based brushing in InterRing is more flexible than its original form [10] in that:

- In InterRing, there can be any number of selected regions by applying the structure-based brushing on multiple clusters. While in the original form, there is only one selected region.

- In InterRing, structure-based brushing can be used together with simple (single node) brushing, while it is hard to combine it with other selection tools in the original form.

4.2 Highlighting

In InterRing, selected nodes are highlighted using a highlight color. We have explored two different highlighting strategies for selected clusters: the *highlight-all* strategy and the *highlight-part* strategy. The first strategy is to cover the whole node with the highlight color in the InterRing display. The problem with this approach is that the original color of a node conveys information about the hierarchical structure. It no longer conveys such information when it is changed to the highlight color. Thus our second strategy, *highlight-part*, is to highlight only part of the node, namely an arc through the center of the node. Hence the node is highlighted and the structure information is retained.

We have provided users an option of showing labels for selected clusters. Once this option is selected, the names of the selected clusters are shown on the display, with each name's left bottom corner located at the center of the corresponding cluster.

5 Modification Tools

Modification tools are provided to users so they can reorganize the structure of the hierarchy. This is important because automated clustering tools often generate ill-formed hierarchies with semantically unrelated nodes sharing the same cluster. Users familiar with the domain can usually find ways to improve the quality of the hierarchy. In InterRing, we have implemented a simple drag and drop modification operation. For example, in the modification mode of InterRing, users can remove a sub-cluster from a cluster by dragging and dropping it into any other cluster, which then becomes its new parent. The color of the removed cluster is retained to help users keep track of it. Figure 8 shows a highlighted cluster being relocated with all of its descendants.

There are many other possible operations that could be useful for hierarchy reorganization, such as cut/copy/paste and split/merge. In our future work, we will incorporate these operations into InterRing, and also explore using highlighting or animation to help users perform the restructuring process more intuitively.

6 Structure-Based Coloring

In InterRing, we use a structure-based coloring strategy to emphasize the hierarchical structure and link InterRing to other displays related to the hierarchical data. The principles of structure-based coloring is similar to that in [10], but is changed to fit the RSF display:

- nodes belonging to the same cluster should have similar colors;

- a parent node's color should be derived from the colors of its children;

- a child with a larger sweep angle contributes more to its parent's color than siblings with smaller sweep angles.

We have tried two different approaches to the structure-based coloring: *middle color* assignment and *average color* assignment. In both of these approaches, a linear colormap is mapped to the 360 degrees of the circle of the InterRing display. We use hue as the variable for the colormap, although other color scales could be used. In both approaches all the leaf nodes are assigned colors based on the degree of the center of their subtended angles. The difference lies in that for the middle color assignment, non-leaf nodes are assigned colors based on their position in the circle, while in the average color assignment, the colors of non-leaf nodes are the weighted average of their children's colors. The sweep angle of each child is used as the weight in the average calculation.

Figure 9 (a) shows an InterRing display using the middle color assignment, while Figure 9 (b) shows the same hierarchy using the average color assignment. We have presented these two figures to several users and it was reported that (b) is preferred since there are often some color jumps in (a). That is, an inner node has the same color with one of its grandchildren, while none of its children contain this color. Thus we determined that the average color assignment is superior to the middle color assignment.

When visualizing large hierarchies that contain thousands of leaf nodes or more, mapping only one colormap to the whole circle may not be enough, since the color differences between the adjacent leaf nodes can be too small to be distinguished. In this case, we propose an approach to divide the whole circle into several angle segments, separated by the edges of some large clusters, and assign a colormap to each of the segments. Thus the color distance of adjacent leaf nodes would be enlarged. Alternately, user-controlled distortion of the colormap could be used to allocate a larger percentage of colors to a selected subset of the hierarchy.

7 Conclusion and Future Work

In this paper, we have proposed improvements to radial, space-filling hierarchy visualizations from the aspects of navigation, modification, selection, and coloring. Although there is still much formal evaluation work to be done, its performance in our VHDR framework has shown InterRing provides intuitive and effective solutions to the interaction tasks required. This leads us to conclude that InterRing may provide a good solution to other common hierarchy interaction tasks. The main contributions of this paper are:

- a multi-focus, intuitive, space efficient distortion process;
- navigation tools such as drilling-down/rolling-up, zooming in/zooming out, panning, and rotation operations;
- hierarchy reorganization tools;
- brushing mechanisms and highlighting strategies;
- structure-based coloring to convey hierarchical structure information.

We are currently in the process of formally evaluating many aspects of our work to find out how well these tools perform and how we can improve them. In particular, we are developing and conducting evaluations for comparing one-direction distortion and two-direction distortion, comparing brushing in InterRing with the structure-based brush that exists in XmdvTool [10], and comparing structure-based coloring with other coloring approaches.

In addition, we believe that other hierarchy visualizations, such as treemaps and node-link graphs, also would benefit from such a set of tools for supporting user interactions. Many of the ideas in this paper, such as the distortion approach, can be readily applied to other hierarchy visualization techniques, which is an avenue we hope to explore.

8 Acknowledgments

We gratefully acknowledge our colleagues in the XmdvTool group at WPI for their contributions to this research. A special thanks goes to Prof. David Brown, who gave many valuable suggestions for this work.

References

[1] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semicircular discs. *IEEE Information Visualization Symposium 1998, Late Breaking Hot Topics Paper, p. 9-12*, 1998.

[2] T. Barlow and P. Neville. A comparison of 2-d visualization of hierarchies. *Proc. of Information Visualization 2001, p. 131-138*, 2001.

[3] G. D. Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Upper Saddle River, N.J: Prentice Hall, 1999.

[4] A. Bruggemann-Klein and D. Wood. Drawing trees nicely with tex. *Electronic Publishing, 2(2), p. 101-115*, 1989.

[5] S. Card, G. Robertson, and J. Mackinlay. The information visualizer, an information workspace. *Proc. of ACM CHI'91, Conference on Human Factors in Computing Sys-tems, p. 181-188*, 1991.

[6] M. Chuah. Dynamic aggregation with circular visual designs. *Proc. of Information Visualization '98, p. 35-43*, 1998.

[7] R. Dachselt and J. Ebert. Collapsible cylindrical trees: A fast hierarchical navigation technique. *Proc. of Information Visualization 2001, p. 79-86*, 2001.

[8] P. Eades. Drawing the trees. *Bulletin of the Institute of Combinatorics and its Applications, p. 10-36*, 1992.

[9] Y. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *Proc. of Visualization '99, p. 43-50*, Oct. 1999.

[10] Y. Fua, M. Ward, and E. Rundensteiner. Navigating hierarchies with structure-based brushes. *Proc. of Information Visualization '99, p. 58-64*, Oct. 1999.

[11] G. Furnas. Generalized fisheye views. *Proc. of Computer-Human Interaction '86, p. 16-23*, 1986.

[12] I. Herman, M. S. Marshall, G. Melancon, D. Duke, M. Delest, and J.-P. Domenger. Skeletal images as visual cues in graph visualization. *Data Visualization'99, p. 13-22*, 1999.

[13] B. Johnson and B. Shneiderman. Tree maps: A space-filling approach to the visualization of hierarchical information structures. *Proc. of Visualization '91, p.284-91*, 1991.

[14] E. Kleiberg, H. van de Wetering, and J. van Wijk. Botanical visualization of huge hierarchies. *Proc. of Information Visualization 2001, p. 87-94*, 2001.

[15] B. Kleiner and J. Hartigan. Representing points in many dimensions by trees and castles. *Journal of the American Statistical Associatioin, June 1981, p. 260-272*, 1981.

[16] H. Koide and H. Yoshihara. Fractal approaches for visualizing huge hierarchies. *Proc. of the 1993 IEEE Symposium on Visual Languages, p. 55-60*, 1993.

[17] H. Koide and H. Yoshihara. Interacting with huge hierarchies: Beyond cone trees. *Proc. of Information Visualization '95, p. 74-81*, 1995.

[18] J. Lamping and R. Rao. The hyperbolic browser: A focus+context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing, Vol. 7 No. 1, p. 33-55*, 1996.

[19] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *Human Factors in Computing Systems, CHI 95 Conference Proceedings*, 1995.

[20] T. Munzner. H3: Laying out large directed graphs in 3d hyperbolic space. *Proceedings of the 1997 IEEE Symposium on Information Visualization, p. 2-10*, 1997.

[21] T. Munzner. Drawing large graphs with h3viewer and site manager. *Proceedings of the Symposium on Graph Drawing GD 98, p. 384-393*, 1998.

[22] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3d hyperbolic space. *Proceedings of the VRML95 Symposium, ACM SIGGRAPH*, 1995.

[23] R. Rao and S. Card. Exploring large tables with the table lens. *Proc. of ACM CHI'95 Conference on Human Factors in Computing Systems, Vol. 2, p. 403-4*, 1995.

[24] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3d visualization of hierarchical information. *Proc. of Computer-Human Interaction '91, p. 189-194*, 1991.

[25] B. Shneiderman. Tree visualization with tree-maps: A 2d space-filling approach. *ACM Transactions on Graphics, Vol. 11(1), p. 92-99*, Jan. 1992.

[26] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. *Proc. of Information Visualization 2001, p. 73-78*, 2001.

[27] J. Stasko, R. Catrambone, M. Guzdial, and K. Mcdonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *Int. J. Human-Computer Studies, Vol. 53, p. 663-694*, 2000.

[28] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualization. *Proc. of Information Visualization 2000, p. 57-65*, 2000.

[29] F. van Ham, H. van de Wetering, and J. van Wijk. Visualization of state transition graphs. *Proc. of Information Visualization 2001, p. 59-66*, 2001.

[30] J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. *Proc. of Information Visualization 1999, p. 73-78*, 1999.

[31] J. Yang, M. O. Ward, and E. A. Rundensteiner. Visual hierarchical dimension reduction for exploration of high dimensional datasets. *Technical Report WPI-CS-TR-02-22*, 2002.

[32] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of graphs with radial layout. *Proc. of Information Visualization 2001, p. 43-50*, 2001.

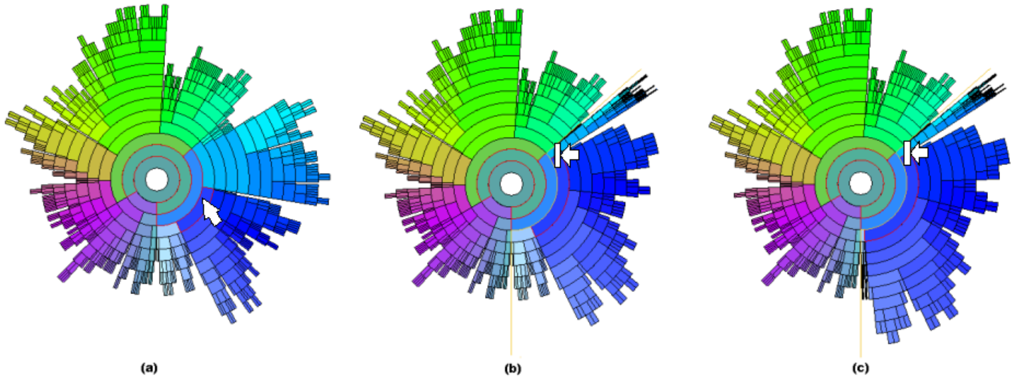


Figure 3: One-Direction Distortion and Two-Direction Distortion. (a) shows the hierarchy before distortion; the cursor is on the pinned node to be enlarged (lower right part of the ring). It has two siblings. (b) shows the consequence of one-direction distortion. The cursor is on the dragged edge. The sibling adjacent to the dragged edge is contracted to provide room for the pinned node. The other sibling is not affected by this distortion. In (c), both siblings are contracted to provide room for the pinned node.

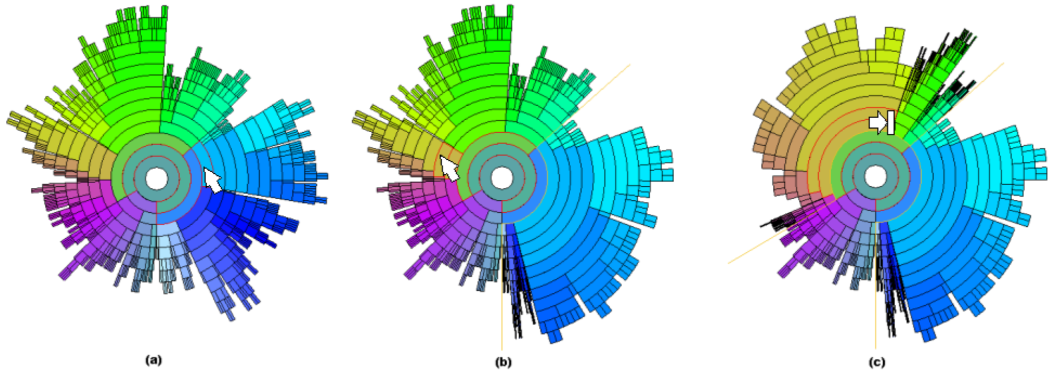


Figure 4: Multi-Focus Distortion. (a) shows the hierarchy before distortion; the cursor is on the first focus. In (b) the first focus region has been enlarged and the cursor is on the second focus region. (c) shows that the second focus region is also enlarged.

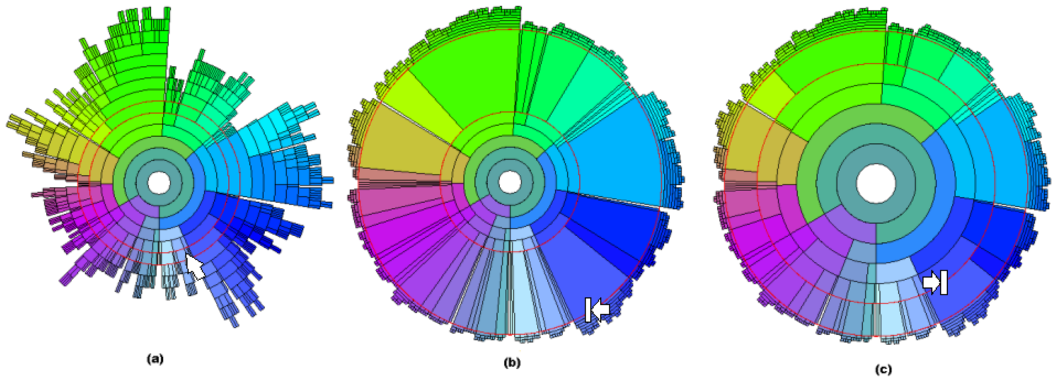


Figure 5: Radial Distortion Enlarging Inner Layers. (a) shows the hierarchy before distortion; the pinned layer is outlined in red. In (b) the outer radial edge of the pinned layer has been pushed outwards by decreasing the thickness of its outer layers. In (c) the inner radial edge of the pinned layer has been pushed outwards to increase the thickness of its inner layers.

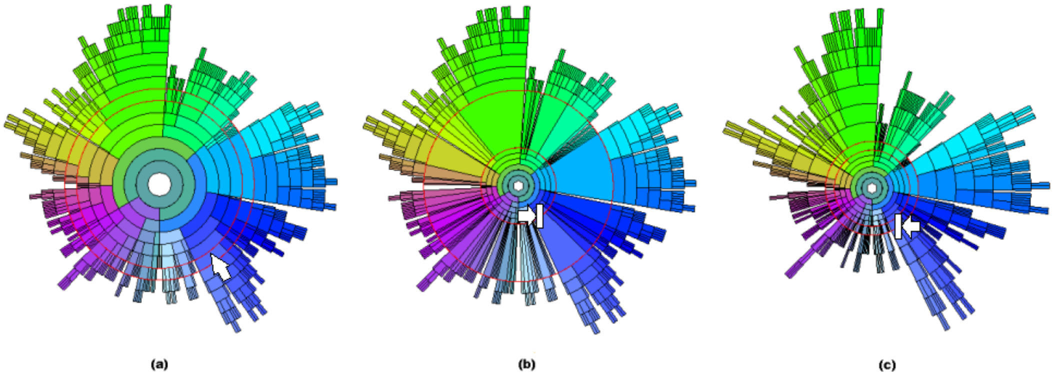
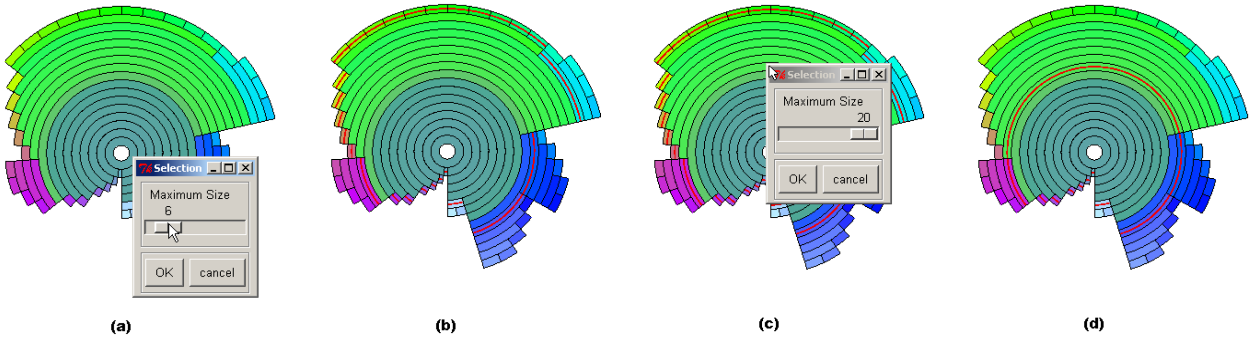


Figure 6: Radial Distortion Enlarging Outer Layers. (a) shows the hierarchy before distortion; the pinned layer is outlined in red. In (b) the inner radial edge of the pinned layer has been pushed inwards by decreasing the thickness of its inner layers. In (c) the outer radial edge of the pinned layer has been pushed inwards to increase the thickness of its outer layers.



Maximum Size
6

Maximum Size
20

Figure 7: Structure-Based Brushing. (a) shows applying structure-based brushing on the root node and setting the threshold to 6. (b) shows the selection result. (c) shows applying structure-based brushing again on a node where a lower level of detail is needed. (d) shows the selection result of these two brushing operations.

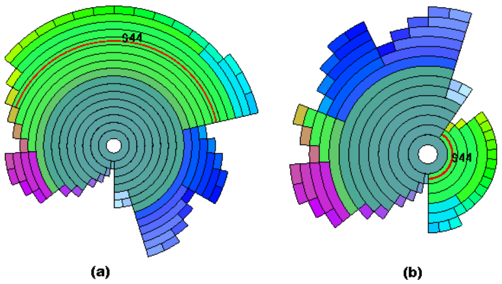


Figure 8: Modification. In (a) a cluster is highlighted with a red arc through it. In (b) the highlighted cluster has been moved. Its new parent is the root node.

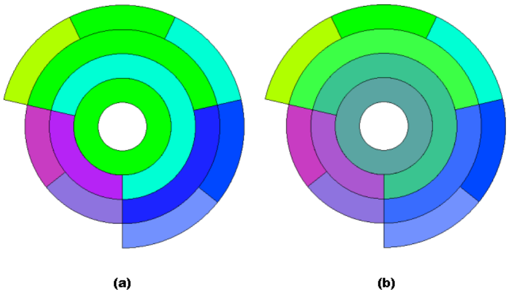


Figure 9: Structure-Based Coloring. In (a) the hierarchy is colored using the middle color assignment. In (b) the same hierarchy is colored using the average color assignment.