Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations

John Stasko and Eugene Zhang
GVU Center and College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
stasko & zhange@cc.gatech.edu

Abstract

Radial, space-filling visualizations can be useful for depicting information hierarchies, but they suffer from one major problem. As the hierarchy grows in size, many items become small, peripheral slices that are difficult to distinguish. We have developed three visualization/interaction techniques that provide flexible browsing of the display. The techniques allow viewers to examine the small items in detail while providing context within the entire information hierarchy. Additionally, smooth transitions between views help users maintain orientation within the complete information space.

1. Introduction

Information hierarchies are evident throughout computer science as well as society in general. Organization charts, family ancestry diagrams, object-oriented program classes and file/directory systems are all examples of hierarchical structures. The most common way to visualize a hierarchy is by using a tree. Many different node-edge tree layout algorithms have been developed ranging from simple 2D planar algorithms[15, 4] to 3D algorithms[8] to hyperbolic methods[6].

Most tree drawing algorithms utilize edges between nodes to indicate parent-child structure. Space-filling techniques such as the Treemap[5, 12] use more implicit containment and geometry features to present a hierarchy. These methods are very good at communicating different attributes, such as size and type, of nodes in the hierarchy, but they are not able to convey structure as well as traditional node-edge diagrams. Refinements of Treemaps such as the use of more balanced aspect ratios[14] and 3D shading[13] even have been suggested to help convey structure better.

The use of circular or radial space-filling[1] techniques has been suggested as an alternative to the (rectangular) Treemap technique[3, 1]. Developers believed that the radial layout methodology better conveys a hierarchy's structure, without sacrificing the use of area to present attributes of the hierarchy's nodes.

We have built a radial visualization system called Sunburst that we have been using as a file system examination tool. An example Sunburst visualization is shown in Figure 1. Additionally, we conducted an experiment comparing the use of Sunburst to a Treemap-style tool in performing typical file and directory-related tasks[11, 9, 10]. It is difficult to briefly summarize objective task performance (successful completion and time taken) in the experiment, but performance was relatively comparable using the two tools, with a trend favoring Sunburst, particularly when comparing initial use of each. Subjectively, participants in the experiment strongly preferred the Sunburst tool.

As part of the study, we asked participants to critique the two different visualization techniques. While Sunburst received a number of different positive comments, one particular weakness was consistently cited: Participants pointed out the small “slice” size of files and directories that were near the periphery of the visualization and said that it was difficult to distinguish the different attributes (name, type, size, quantity, etc.) of such slices. Figure 1 shows examples of this phenomenon in the narrow-banded slices of peripheral files in the top and right regions of the visualization, in particular. Obviously, as the file system being examined grows in size, this problem becomes more prevalent.

This issue led us to investigate ways of improving the Sunburst tool to remedy the problem. We did not want to change the fundamental attributes of the visualization since it seemed to be generally useful. Rather, we sought to aug-

[1] We use the term “space-filling” here, but this is somewhat a misnomer in that the radial techniques do not completely occupy the display space as does the Treemap, for instance.

0-7695-0804-9/00 $10.00 @ 2000 IEEE

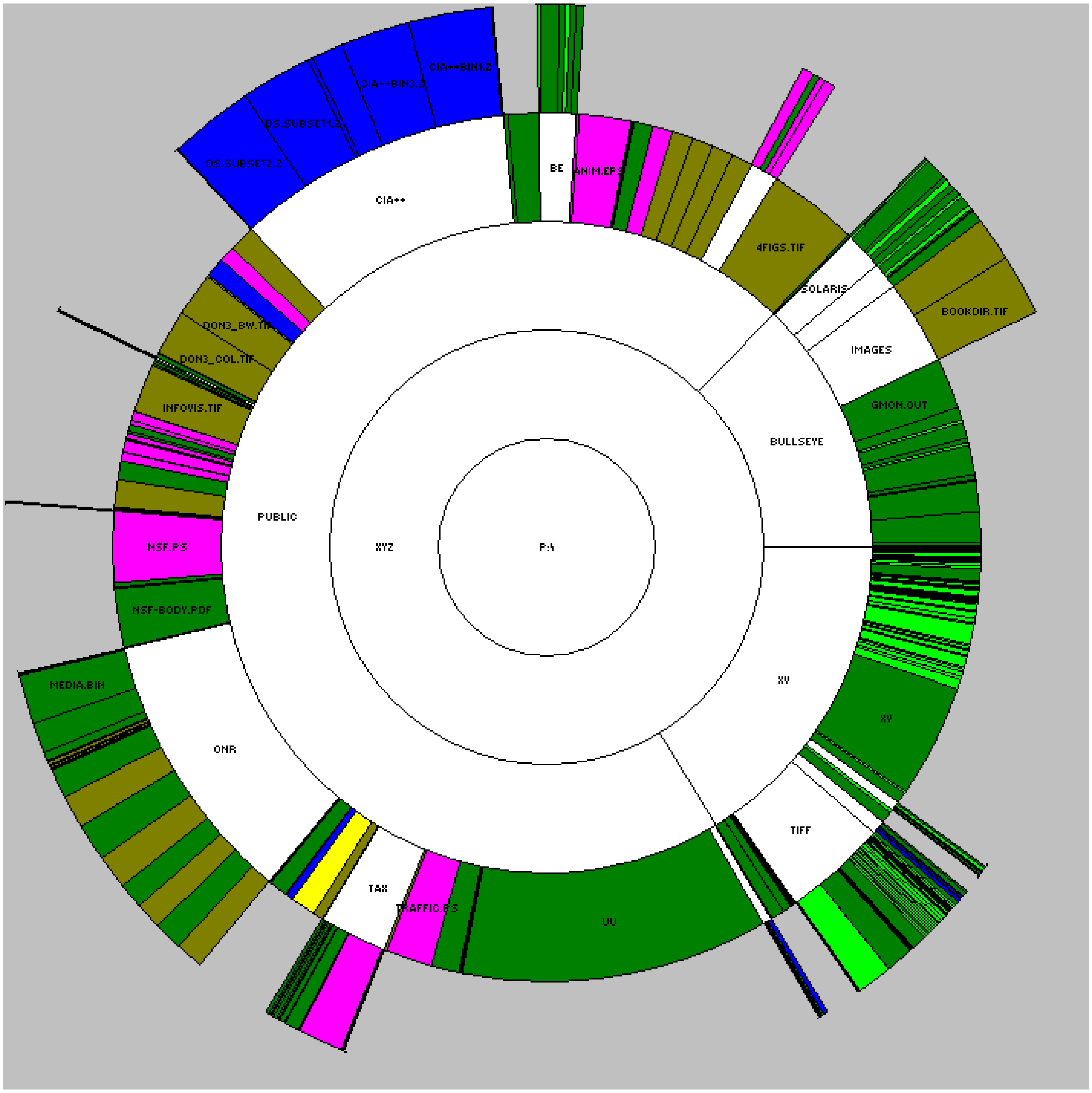


Figure 1. Example Sunburst visualization of a file structure.

ment the system to help people distinguish and examine the small, peripheral items (files or directories) in the hierarchy being presented.

2. Background

Existing tree visualization systems have addressed this general item-size problem and provided techniques to help viewers examine peripheral nodes. Plaisant, Carr and Shneiderman provide a good summary of general image browsing techniques in [7], and below we focus on two of the most pertinent projects that have influenced our work.

Andrews and Heidegger's radial space-filling system[1] uses two semi-circular areas to represent a file system, as shown in Figure 2. Selecting a focus (typically small) directory in the left overview window makes that directory and its descendant file/directories appear in the other view. This provides a form of two-level "overview and detail" information visualization.

The hyperbolic tree browser uses a node-edge tree visualization technique, but it is implemented using hyperbolic geometry[6]. In doing so, the tree is drawn in a more space efficient manner—All the nodes and edges fit within a circle, and the viewer is able to focus on elements to bring them to the center of the display. The system animates transitions in focus to help viewers track changes.

We felt that aspects of these two approaches could help to make the Sunburst tool more useful, but each also has drawbacks. Like Andrews and Heidegger's system, we anticipated providing some notion of a "focus" area, but we still wanted to be able to smoothly and flexibly alternate between both global and detailed views, and we wanted to maintain a full circular image thus giving each element more display real estate. We found the animated transitions of the hyperbolic tree browser to be appealing and useful, but we still wanted to utilize a space-filling visualization methodology and we wanted to avoid the relatively significant node position changes that the hyperbolic browser can exhibit.

Based on our experiences with the Sunburst system, observations from the experimental study of it and Treemap, as well as analysis of related systems such as those discussed above, we developed a set of design guidelines for an improved version of Sunburst:

- Maintain a full circular space-filling methodology and global view of the entire hierarchy
- Allow a more detailed examination of small peripheral files and directories, but keep it in context of the entire information structure, that is, both should be visible simultaneously
- Ensure that the overview of the entire hierarchy remains relatively stable in layout to promote comprehension and familiarity
- Maintain a balance between the visibility of both the hierarchy overview and the detailed focus display, when present
- Be more space-efficient (leave less blank boundary area), particularly when presenting the peripheral files
- Avoid the use of multiple windows or scrollbars, if possible, that would force viewers to mentally connect separated areas or lose context of the entire space
- Allow the viewer to easily track display changes between a global view and a detailed view of the peripheral nodes

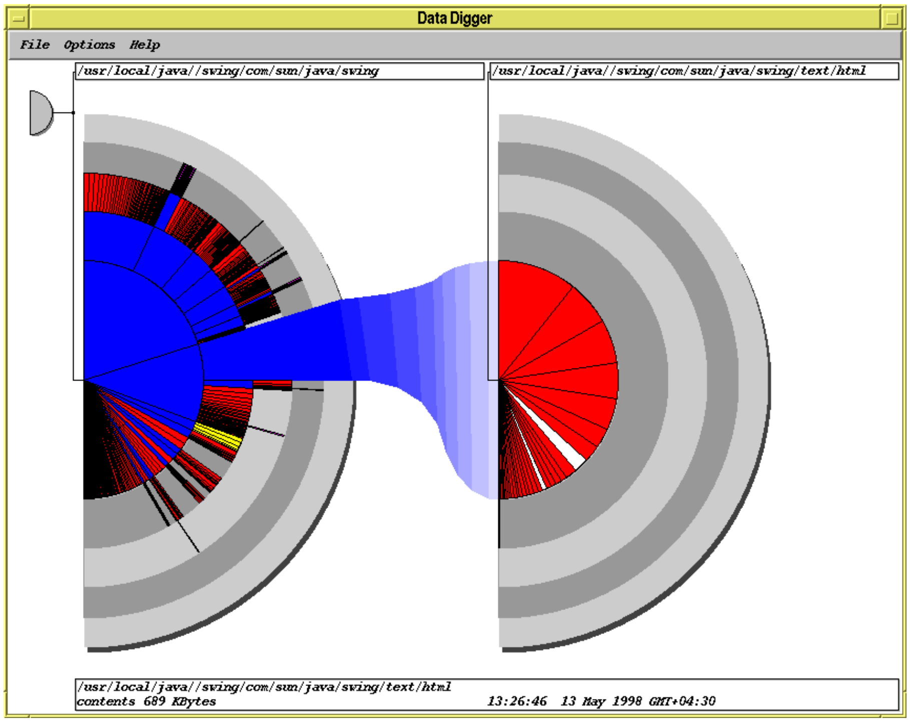
In the next section, we describe three techniques developed to meet these requirements. The techniques are not simply new visualizations. Rather, they are combinations of new visualizations together with navigation and interaction techniques to provide a more complete system useful for illustrating hierarchies. This kind of more integrated methodology, visualization coupled with powerful, flexible navigation, is becoming more prevalent in information visualizations developed to address challenging problems today.

3. Designs

All three techniques that we have created follow the general format of the Sunburst system. Files and directories deeper in the hierarchy are drawn further from the center. Child nodes are drawn within the arc subtended by their parents. The angle of an item's arc corresponds precisely to the item's size in the file structure. The color of an item can represent the file's type or its most recent modification date. (Directories are always white.) User interface controls allow the viewer to move deeper or higher in the file hierarchy and to show all files at once. When the mouse settles momentarily over an item, its name and path are shown in a pop-up balloon.

Due to space concerns, we only present the three final display methods here, but all are the result of an extensive design and prototyping process. We iterated through many different design ideas. Some proved lacking once they were prototyped; Others turned out to be impractical to implement. The three final methods described here each have relative strengths and weaknesses, but none appears clearly best via an initial assessment.

The three techniques are similar (and different from the original Sunburst) in that each shows a selected focus section of the hierarchy within the context of an overview of the entire hierarchy. By simply double-clicking on an item,



/usr/local/java//swing/com/sun/java/swing

/usr/local/java//swing/com/sun/java/swing/text/html

/usr/local/java//swing/com/sun/java/swing/text/html
contents 689 KBytes 13:26:46 13 May 1998 GMT+04:30

Figure 2. Andrews and Heidegger's connected semicircle visualization technique. (Image courtesy of Keith Andrews.)

the viewer selects it for more focused display. The three methods differ in how they display the focus region. Below we describe each method in more detail and compare the three techniques.[2]

3.1. Angular Detail method

When the viewer selects an item for detailed examination in the *Angular Detail* method (Figure 3 shows six frames captured from an example sequence of this process with a focus selection made in the lower left of the structure), the drawing of the entire hierarchy first shrinks and moves to the boundary of the enclosing window on the side diametrically opposite where the selection was made. Next, the selected item appears to extend out of the overview, and then it and its children expand radially outward to occupy a larger display area. Within the overview, the selected item is colored in an orange cross-hatch pattern and a circular bullseye icon is drawn over it to indicate its position within the global structure. If a viewer returns to the default global view of the hierarchy, the small overview moves to the center of the display and grows back to its default size.

In this method, the user interface allows viewers to control the radii size and the sweep angle of all the items in the focus region. Both controls can help increase the arc lengths (circumference) of elements in the local structure, thus providing better visibility.

3.2. Detail Outside method

The *Detail Outside* method, shown in Figure 4, uses a relatively different focus methodology than the Angular Detail method. When a viewer selects an item, the entire hierarchy shrinks in the center of the display and stays there. The selected item emerges from the overview on the edge of the overview closest to its position, and the item expands to be a new complete circular ring-shaped region around the overview. Further, all its children expand to their appropriate relative positions and sizes outside the selected item. Just as is done in the Angular Detail method, the selected focus item is highlighted via the cross-hatch pattern and icon.

3.3. Detail Inside method

The *Detail Inside* method, shown in Figure 5, works similarly to Detail Outside, except that the selection of an item

[2] Describing the dynamic operations of these three techniques is difficult in the (static) text of this article. We have produced a video presentation that summarizes the methods. Viewing it is the best way to understand the operations.

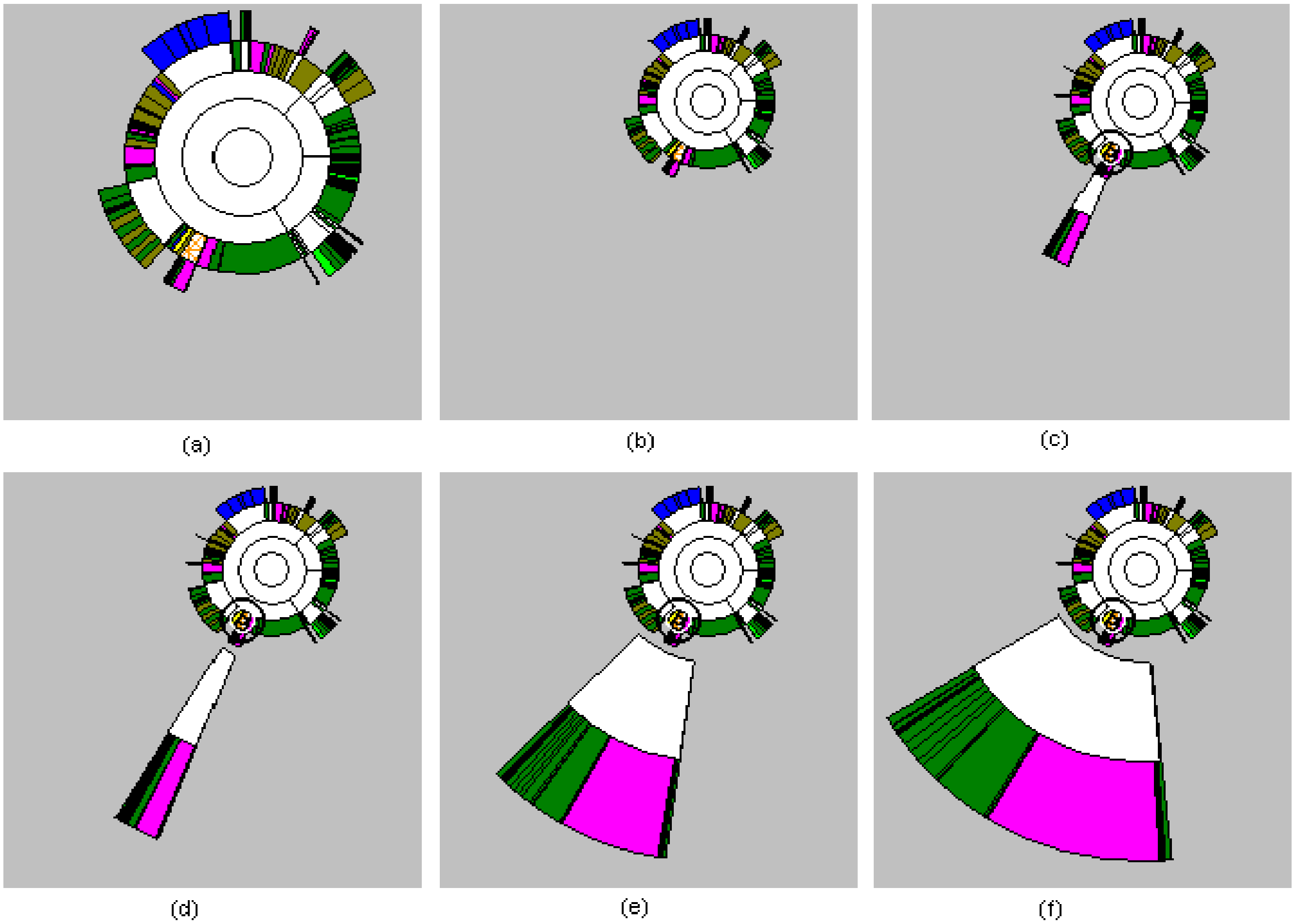


Figure 3. Sequence of snapshots from the Angular Detail method. The overview is shrunk and pushed to the opposite side from where the focus region grows outward.

for detailed examination results in the drawing of the entire hierarchy being pushed outward and taking on a large ring shape. The selected item then extends inward to the center of the window, is drawn as a circle, and expands radially to occupy the center of the image. All its children fill in subsequent circular layers extending out from the center just as is done in the basic Sunburst methodology.

In all three methods, items can be selected for detailed examination from the general view of the hierarchy, the smaller overview, or the expanded focus area. When the display already is showing a focus item, a new focus selection causes the current detailed view to disappear, then the overview smoothly moves to its new position (under the Angular Detail method) or it remains in its current position (under the other two methods). Next, the newly selected item grows from its current position in the overview display region, slides to its new position in the detail view region, and smoothly expands its sweep angle to the appropriate size. Being able to select items for detailed study from an already focused view is important in supporting the user to select extremely small items that might be difficult to pick otherwise.

Each state change on the display is illustrated via smooth animated transitions between viewing configurations. This includes changes from focus to overview, overview to focus, focus to different focus, and changes in viewing method. We feel that smoothly animated state changes allow the viewer to track the context of focus changes, assimilate updates to the display more easily, and maintain a more clear sense of the entire information hierarchy. Smooth, animated transitions also offload the viewer's understanding of dis-

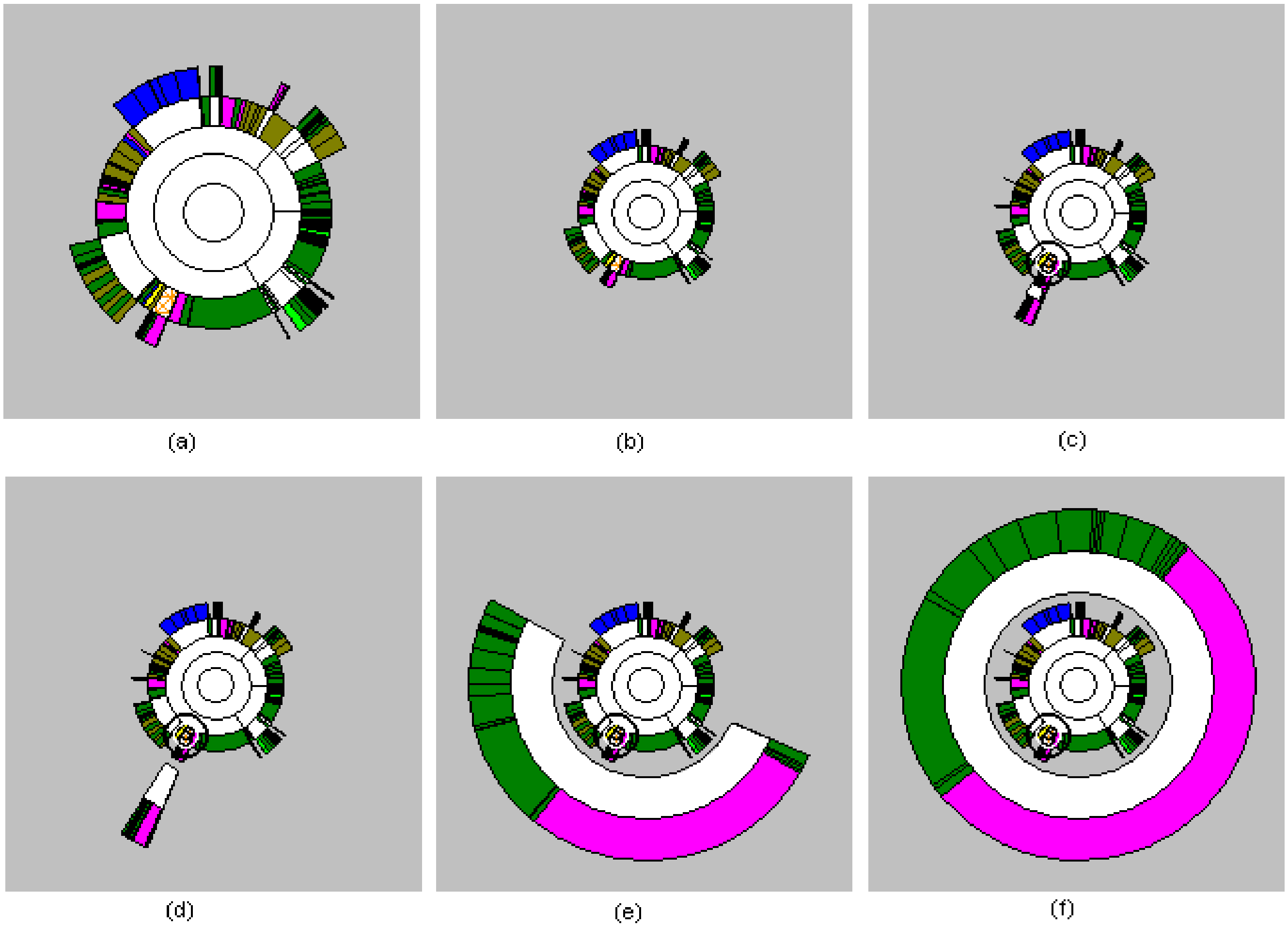


Figure 4. Sequence of snapshots from the Detail Outside method. The overview shrinks into the center and the focus area is drawn in a ring around the overview.

play updates from their cognitive system to their perceptive system[2, 8]. The speed and pacing of the animations is user-controllable, with default values determined via informal testing and feedback.

3.4. Comparison and Analysis

As mentioned earlier, each of the three techniques has relative strengths and weaknesses. Here, we more carefully assess the characteristics of each display.

Angular Detail – Because this method orients the overview to the side opposite the focus area and the focus area itself appears to grow out from its customary position, this method is, in some sense, the most "natural." By that, we mean that it reconfigures the display the least, and it is easy for the viewer to resolve positioning of the focus area within the entire hierarchy. On the negative side, this method typically utilizes the least area of the display window, so it can be relatively inefficient as a space filling method. Also, this method is the most user-controllable (focus radii and sweep angle), which can be interpreted either as a positive or negative.

Detail outside – This method is relatively intuitive and visually consistent in that the overview area is always shrunk in the center of the display window, thus exhibiting a miniature view of the entire hierarchy. Taking the focus item and expanding it to be a complete ring can be somewhat visually disconcerting, however. A very small slice is changed to be a complete ring, and all its children are drawn relative to that. This will probably require the viewer to mentally assimilate the change more than the Angular Display method. An important advantage of this technique is

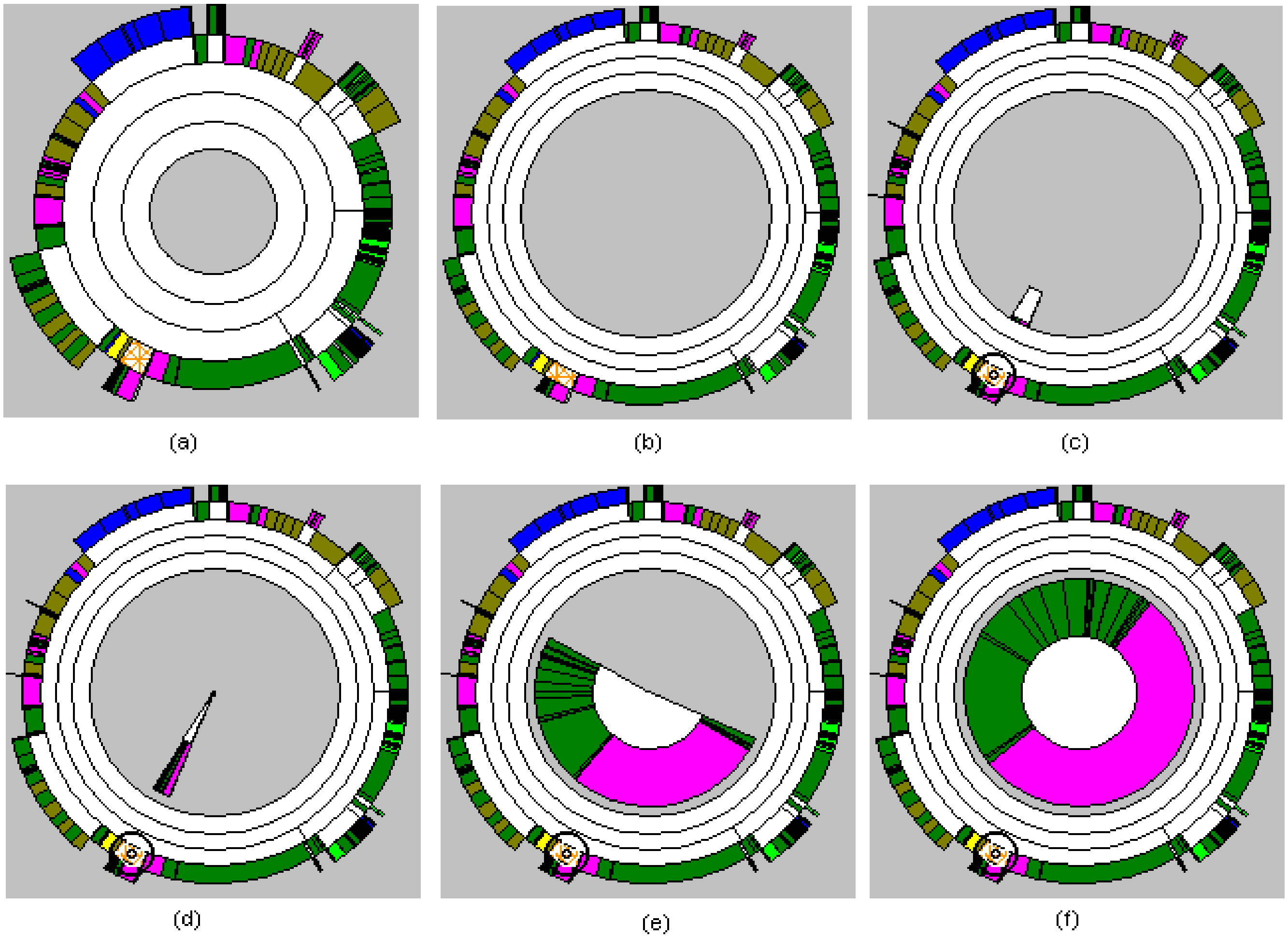


Figure 5. Sequence of snapshots from the Detail Inside method. The overview is shrunk and pushed outward and the focus region is drawn in the center circular area.

that the focus region is given a large area and a full 360 degrees to be drawn within, thus supporting easier inspection and analysis. This method is also relatively space efficient.

Detail inside – This method may be the least intuitive and may exhibit the most significant display change because it distorts the overview by pushing it outward toward the window boundary. However, by pushing the overview outward, it actually increases the arc length (circumferences) and therefore the visibility of the individual items in the overview. Even though the focus region is relatively small in the center of the display, it utilizes a full 360 degrees of sweep angle which is often sufficient to show a local structure that has a relatively small sweep angle and little depth to its descendants.

In general, to focus on a directory and its children, one can either increase the sweep angle of that segment, increase the radius to it, or both. Since a sweep angle can expand to at most 360 degrees, increasing the angle will not be too beneficial for a directory whose size is already large. For much smaller directories, the gain from expanding the sweep angle becomes substantial. As for radius increases, in both the Detail Outside and Detail Inside methods, the maximum radius for expansion is bounded by the maximum radius of the display window. In the Angular Detail method, a larger potential increase in the segment's radius is possible since the slice can extend across the entire display window, thus making it more helpful for focusing on relatively large directories with many children and levels.

4. Implementation Issues

The Sunburst system augmented by the techniques discussed here was implemented using Visual Basic and the Windows' API. The demonstration shown in an accompanying video was filmed while running on a personal computer with an Intel Pentium III processor (550 mhz).

The hierarchical file structure shown in the video and in the example figures in this article consists of approximately 500 files and directories. The file system is initially read and stored in a Microsoft Access database. Attributes for each item include file-related attributes (path, name, date last modified, size, and so on), hierarchical linking attributes (level from root, parent, etc.) and display attributes (relative start angle, sweep angle, etc.).

At the core of all three methods is an animation update routine that utilizes the current display positional parameters and the destination positional display parameters, and linearly interpolates between them. The display parameters include the center position, the innermost and the outermost radii of the hierarchy, and the start angle and end angle for the animated region. This routine is called at most three times for each transitional scenario. For instance, in the case of a transition from the overview to a focused display in the Angular Detail method, the animation routine in invoked as such:

- Animate the the shrinking and sliding of the overview to its final size and the the side towards its final destination, respectively. The center and the radii change over the course of animation, while the start and end angles remain 0 and 360 degrees.
- Animate the growing and sliding of the focus elements out of their positions inside the shrunken overview to their destination position. During this process, center and radii change while angles remain constant.
- Animate the expansion of the focus elements from their original angles to the exaggerated angles. This process involves the changing of angles only.

Because the graphics primitive we used to draw individual files or directories only draws complete pie-shaped segments emanating from the center of a circle, we must always redraw the hierarchy from the outside in, in order to have inner elements overlay the central areas of the more peripheral elements.

In terms of performance, the main bottleneck for drawing and animating the display is simply the number of file/directory segments to be drawn. Each operation requires trigonometric, floating point calculations to determine the coordinates of the "pie slice" being drawn. We utilize three primary "short cuts" to reduce the total number of items drawn in a display, and thus speed up redraw times.

First, in the normal Sunburst view, we only draw segments with a sweep angle greater than 0.0001. Actually, this occurs because the conversion in attempting to draw segments of smaller angles results in an error, making the angle appear to be greater than 360 degrees. We could probably increase this general cut-off value without a significant degradation in the view, but we want to communicate to the viewer extremely small areas with many densely packed files and directories.

Second, we pre-draw and cache particular key frame images of the file hierarchy expanded out completely, so that these can simply be loaded as needed, rather than redrawn repeatedly. In particular, we store the normal view of the entire hierarchy, the ending shrunken view of the hierarchy used as the overview in both the Angular Detail and Detail Outside methods, and the "pushed out" overview of the complete hierarchy shown in the Detail Inside method. Of course, this needs to be done for both the color-by-type and color-by-age modes. To highlight the individual focus item in the overview, we simply draw the cross-hatch pattern and circle icon over the image at the correct location.

Third, during transitions, we only draw elements with a relatively large sweep angle, i.e., the largest 100 elements. Using this approach, we are able to maintain relatively consistent redraw times no matter how large the hierarchy is. Further, if we only redraw elements larger than a particular angle, some focus areas may have no elements of that size and nothing would be redrawn! At the end of transitions, we simply use the cached images for the overview region. With ongoing increases in processor speeds and graphics capabilities, we would expect soon to be able to redraw all items during view transitions and still maintain sufficient animation speeds.

Because of this last optimization, all our techniques scale up well to handle larger file hierarchies. We have found performance to be reasonable on a hierarchy of almost 70,000 files and directories.

5. Conclusion and Future Work

We have introduced three visualization/navigation techniques to help viewers explore small, indistinguishable portions of radial, space-filling displays of hierarchies. The techniques always keep some overview of the entire hierarchy present to preserve context, and they utilize smooth, animated transitions in state to help people track changes in focus. They differ in where and how the focus and overview regions are shown. Each of three techniques has relative advantages and disadvantages, and it is likely that subjective preference would dictate which method an individual uses.

While we have described these focus+context update techniques within the context of our radial Sunburst tool, we expect that they could be generalized to other information visualizations and browsers of large information spaces. Each of our techniques is an example of a more general algorithm for handling the relative placement of and transition between overview and detail portions in a multi-focus information visualization.

A number of future improvements to the techniques, as well as the development of alternative methods, are possible:

- Currently, each level in an overview or focus area has the same radius. It would be interesting to see if some scaling of the radii on different levels could improve the displays. Alternatively, levels could be collapsed to utilize space better.
- Updates for new focus selections picked from an already focused view could be made smarter by performing fewer display changes in certain circumstances.
- For the Angular Detail method, it appears possible to automatically calculate the value and proportion between the expansion angle and the radius per level in order to maximize use of the display window.
- It would be challenging to allow multiple foci within the structure, but this might be useful to facilitate comparisons.
- A fourth general method is possible—one that expands a focus item radially in place (increasing its sweep angle) while compressing all other items. This technique would have the advantage of having less overall update and motion than the three techniques presented here, but it would also distort relative sizes between the focus and other items more.

Our objective in this article was to introduce and describe three display techniques, but clearly, an important area of future work would be an evaluation of their utility. It would be interesting to learn how these techniques would perform in an experimental evaluation of typical file/directory tasks such as in [11, 9, 10]. In addition to utility, personal subjective preference of the methods would matter as well. Informally, we presented all three techniques to our local research group and asked for preferences. As expected, all three methods had backers with no clear-cut favorite. A more careful subjective evaluation of the techniques would be helpful in understanding the relative advantages and disadvantages of each more clearly, and might elicit useful suggestions on further improvements as well.

References

[1] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *IEEE Information Visualization Symposium 1998, Late Breaking Hot Topics Procs.*, pages 9–12, Raleigh Durham, NC, Oct. 1998.

[2] R. Baecker and I. Small. Animation at the interface. In B. Laurel, editor, *The Art of Human-Computer Interface Design*, pages 251–267. Addison-Wesley, Reading, MA, 1990.

[3] M. C. Chuah. Dynamic aggregation with circular visual designs. In *Proceedings of the 1998 IEEE Information Visualization Symposium*, pages 35–43, Raleigh Durham, NC, Oct. 1998.

[4] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Upper Saddle ARiver, NJ, 1998.

[5] B. Johnson and B. Shneiderman. Tree-maps: A space filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Visualization '91*, pages 284–291, San Diego, CA, Oct. 1991.

[6] J. Lamping and R. Rao. The hyperbolic browser: A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, 7(1):33–55, Mar. 1996.

[7] C. Plaisant, D. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Software*, 12(2):21–32, March 1995.

[8] G. G. Robertson, S. K. Card, and J. D. Mackinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):57–71, Apr. 1993.

[9] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. Technical Report GIT-GVU-00-03, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, Feb. 2000.

[10] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, 2000. (To appear).

[11] J. Stasko, M. Guzdial, and K. McDonald. Evaluating space-filling visualizations for presenting hierarchies. In *IEEE Information Visualization Symposium 1999, Late Breaking Hot Topics*, pages 35–38, San Francisco, CA, Oct. 1999.

[12] D. Turo and B. Johnson. Improving the visualization of hierarchies with Treemaps: Design issues and experimentation. In *Proceedings of the IEEE Visualization '92*, pages 124–131, Boston, MA, Oct. 1992.

[13] J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Proceedings of the '99 IEEE Symposium on Information Visualization*, pages 73–78, San Francisco, CA, Nov. 1999.

[14] M. Wattenberg. Visualizing the stock market. In *ACM SIGCHI 1999 Extended Abstracts*, pages 188–189, Pittsburgh, PA, May 1999.

[15] C. Wetherell and A. Shannon. Tidy drawings of trees. *IEEE Transactions on Software Engineering*, SE-5(5):514–520, Sept. 1979.