Int. J. Human-Computer Studies (2000) **53**, 663–694
doi:10.1006/ijhc.2000.0420
Available online at http://www.idealibrary.com on IDEAL®

An evaluation of space-filling information visualizations for depicting hierarchical structures

JOHN STASKO
Graphics, Visualization and Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, USA.
email: stasko@cc.gatech.edu

RICHARD CATRAMBONE
School of Psychology, Georgia Institute of Technology, Atlanta, GA 30332-0170, USA

MARK GUZDIAL AND KEVIN MCDONALD
Graphics, Visualization and Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, USA

(Received 31 January 2000, and accepted in revised form 31 May 2000)

A variety of information visualization tools have been developed recently, but relatively little effort has been made to evaluate the effectiveness and utility of the tools. This article describes results from two empirical studies of two visualization tools for depicting hierarchies, in particular, computer file and directory structures. The two tools examined implement space-filling methodologies, one rectangular, the Treemap method, and one circular, the Sunburst method. Participants performed typical file/directory search and analysis tasks using the two tools. In general, performance trends favored the Sunburst tool with respect to correct task performance, particularly on initial use. Performance with Treemap tended to improve over time and use, suggesting a greater learning cost that was partially recouped over time. Each tool afforded somewhat different search strategies, which also appeared to influence performance. Finally, participants strongly preferred the Sunburst tool, citing better ability to convey structure and hierarchy.

1. Introduction

In most areas of computer science, early research efforts focus on developing new, innovative techniques and algorithms. As the area matures, one can and should expect more critical, analytical studies to emerge. The area of information visualization is no different. Early research has largely focused on the development of innovative visualization techniques. Relatively little empirical study of the effectiveness of the visualizations has been conducted, however. In the opening plenary at the 1998 Information Visualization Symposium, George Robertson stressed the importance of empirical evaluation in this area. Careful empirical study of a visualization tool can help us to better understand its relative strengths and weaknesses, the tasks for which it is most appropriate, and can suggest improvements.

1071-5819/00/110663 + 32 $35.00/0

This article describes an evaluation of two information visualization tools used to present hierarchies, specifically, computer directory and file structures. Hierarchies are certainly one of the most common and important information structures in computing. The first tool studied utilizes the Treemap display technique developed by Shneiderman (1992) and Johnson and Shneiderman (1991), a well-known rectangular, "slice and dice" methodology for representing hierarchical information structures. The second tool, a Sunburst display, utilizes a similar region-filling technique, but it employs a circular or radial layout methodology.

We chose to study the Treemap technique because it is very well known, has been used in many different application areas, and because we have used it successfully ourselves in the past for file manipulation tasks. Furthermore, the Treemap is generally regarded as being good in representing the attribute of the information structure portrayed through rectangle area (usually size), but not so good at conveying the structure of the hierarchy. This attribute, in fact, has led other researchers to suggest modifications such as a 3D, shaded "cushion" Treemap (van Wijk & van de Wetering, 1999) and a more square aspect ratio Treemap (Wattenberg, 1999).

The Sunburst tool is a new system that we developed and that we felt conveyed both area and structure well. We wanted to discover how it would compare to the Treemap, how well the tools would assist file browsing tasks, what strategies people would employ with each tool and we wanted to gain insight on how to improve such tools' capabilities.

Turo and Johnson conducted an earlier empirical evaluation of the Treemap algorithm when used on file hierarchies (Turo & Johnson, 1992). They compared people performing directory browsing tasks using a Treemap tool against people using the UNIX tcsh shell. Twelve people participated in the study, doing seven tasks. They were allowed a maximum of 5 min per task. Five questions concerned tasks that were local in scope, dealing with particular files or directories. All the participants answered these questions correctly, and a reliable time difference was found favoring the UNIX shell on two of the tasks. The authors attributed this difference to experience with the UNIX shell vs. inexperience with the Treemap. On two tasks that were more global in scope, the UNIX users failed to correctly answer questions six times, while the Treemap users answered all correctly. A reliable time advantage was found for the Treemap too.

The study described in this article differs from the one of Turo and Johnson in that it compares two visualization tools, as opposed to one visualization tool and command-line shell. Because the two visualization tools are relatively similar, the comparison is in some sense fairer, comparing "apples to apples".

The two visualization techniques we examine in this study are relatively similar and are examples of space-filling hierarchical visualizations. Many other types of visualizations for hierarchies do exist such as traditional 2D trees (Wetherell & Shannon, 1979; Kumar, Plaisant & Shneiderman, 1997), 3D cone trees (Robertson, Card & Mackinlay, 1993), hyperbolic trees (Lamping & Rao, 1996), pyramid-style displays (Beaudoin, Parent & Vroomen, 1996), and even outline-oriented views such as the Windows NT Explorer. Each of the different visualization styles better facilitates a different set of information exploration tasks. We focused on the two space-filling approaches because they seem well-suited to tasks involving file attributes such as type and size, and because we wanted to create a manageable study carefully comparing these two related visualizations.

2. Visualization tools

The first visualization tool employed in the experiment uses the rectangular, space-filling Treemap technique (Shneiderman, 1992). We implemented the Treemap algorithm described in Johnson and Shneiderman (1991) to build a file and directory browser for UNIX-based Sun workstations. The tool utilizes three windows: a control panel, a color legend window and the main file structure viewing window.

The main window is an 800×800 pixel display devoted to depicting the Treemap structure, an example of which is presented later in Figure 2. The presentation of directories and files proceeds by slicing out rectangular regions in alternating horizontal and vertical pieces. The area of a file/directory's rectangle corresponds precisely to its size. File types are colored according to a mapping we created that is presented in the legend window. For example, directories are white, postscript files yellow, executable files orange and so on. Whenever the user clicks once on a file or directory, its entire path is listed in the upper left of the window.† Double-clicking on a directory or file "refocuses" the display such that the selected directory or file is now the top or root of the display.

The control panel gives the user control of the basic interface options in the tool. One set of buttons controls which directory is the root or focus. Buttons exist for resetting back to the original hierarchy root directory or back up one directory to the parent of the current focus. Another set of buttons allow the user to vary the depth of files/directories shown from the root. The "Maximum" button is a convenient way to quickly jump to the maximum depth. The control panel also contains buttons for controlling alternative color renderings of files, namely one based on file age and a random mapping, but this functionality was unused in the experiment (i.e. participants were told not to use this feature since it was unrelated to their tasks). The legend and control panels are shown in Figure 1.

We used the Treemap (TM) algorithm variant in which no padding (borders) around directories is added. Most current Treemap implementations use borders, and while this can help convey structure, it can also use too much space and diminish the room for displaying the files. This may become problematic for larger directory structures as evident in our experiments, so we did not include it.

The Sunburst (SB) tool utilizes a similar space-filling visualization technique, but files and directories are laid out radially. The root or top directory being shown is at the center of the display. Successive levels of directories and files are drawn further away from the center. Each level is given equal width, but the circular angle swept out by a directory or file directly corresponds to its size. Thus, the size of any two files in the hierarchy can be compared via the angles they subtend. The absolute area of two "pie slices" can be used only to compare sizes on the same level, however. An example of this visualization is shown in Figure 3.

The Sunburst tool utilizes the same three windows as the Treemap tool. While developing the Sunburst, we learned of similar, independent, radial space-filling visualizations

† A mouse-over technique could be implemented instead, but that was not included for this experiment.

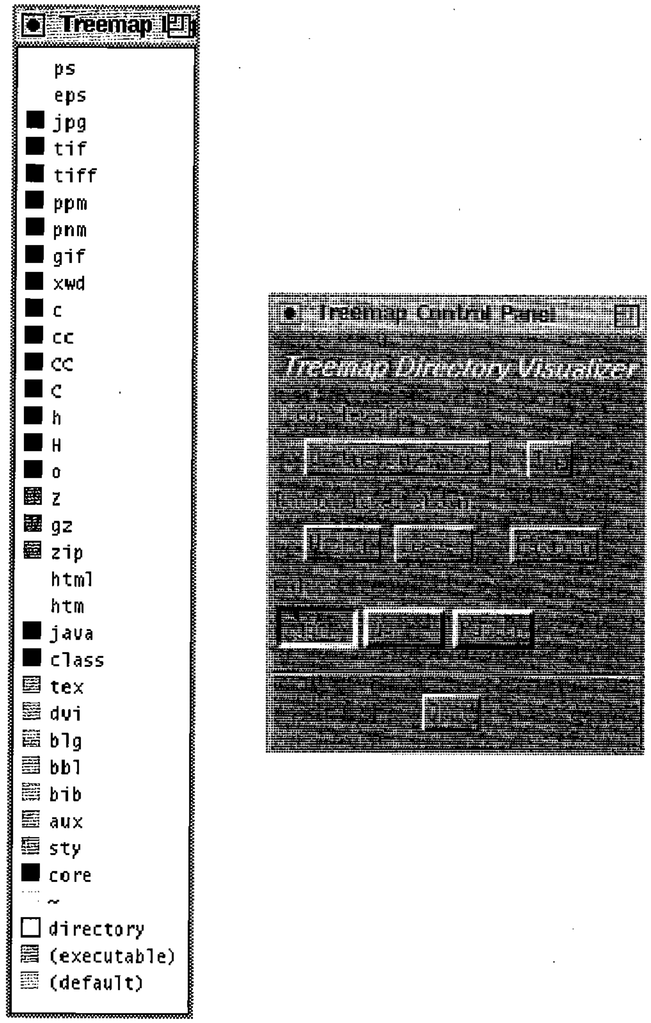

FIGURE 1. Legend and control panel windows for the Treemap tool. Sunburst's are identical.

being created by Chuah (1998) and Andrews and Heidegger (1998). In general, this radial layout methodology is a well-known idea just now being utilized more broadly in information visualization tools. We take the number of systems using it as some evidence of the potential utility of the idea.

3. Experiment overview

The goal of our study was to compare how the TM and SB tools would assist people in performing typical directory/file tasks. Each participant performed a series of tasks using

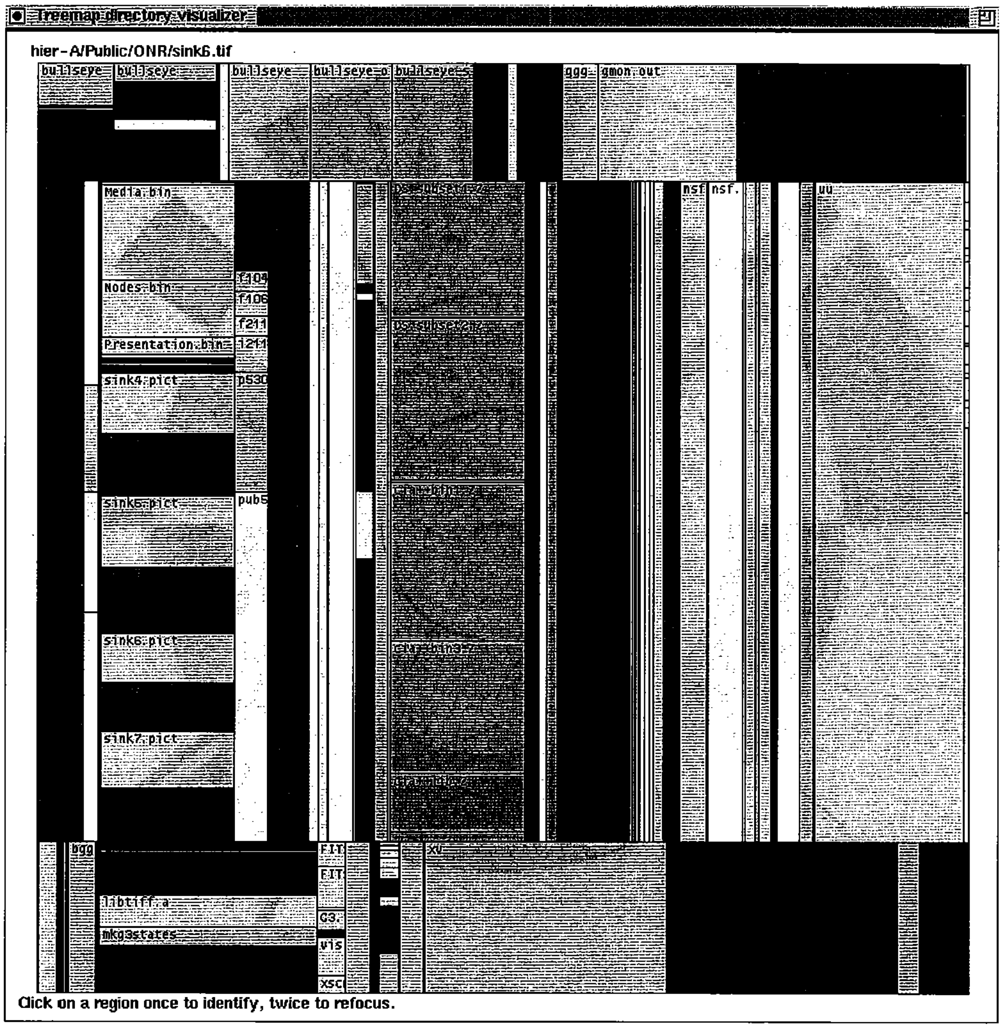

FIGURE 2. Treemap depiction of file Hierarchy A.

both tools, but on different hierarchies in order to avoid any learning effects due to working on the same hierarchy twice. All of the tasks required a person to find or identify a particular file or directory, or to make a comparison of two files or directories.

Presumably, each tool has certain strengths that would aid users in carrying out different types of tasks. We hypothesize that the TM tool will be better for tasks involving file sizes, such as a comparison of sizes or finding the largest file, with respect to both accuracy and time. This is because the Treemap does not explicitly represent directories, thus providing more area for showing the sizes of files. With TM, the viewer may have to compare rectangles of different aspect ratios, and with SB the viewer must compare angular slices.

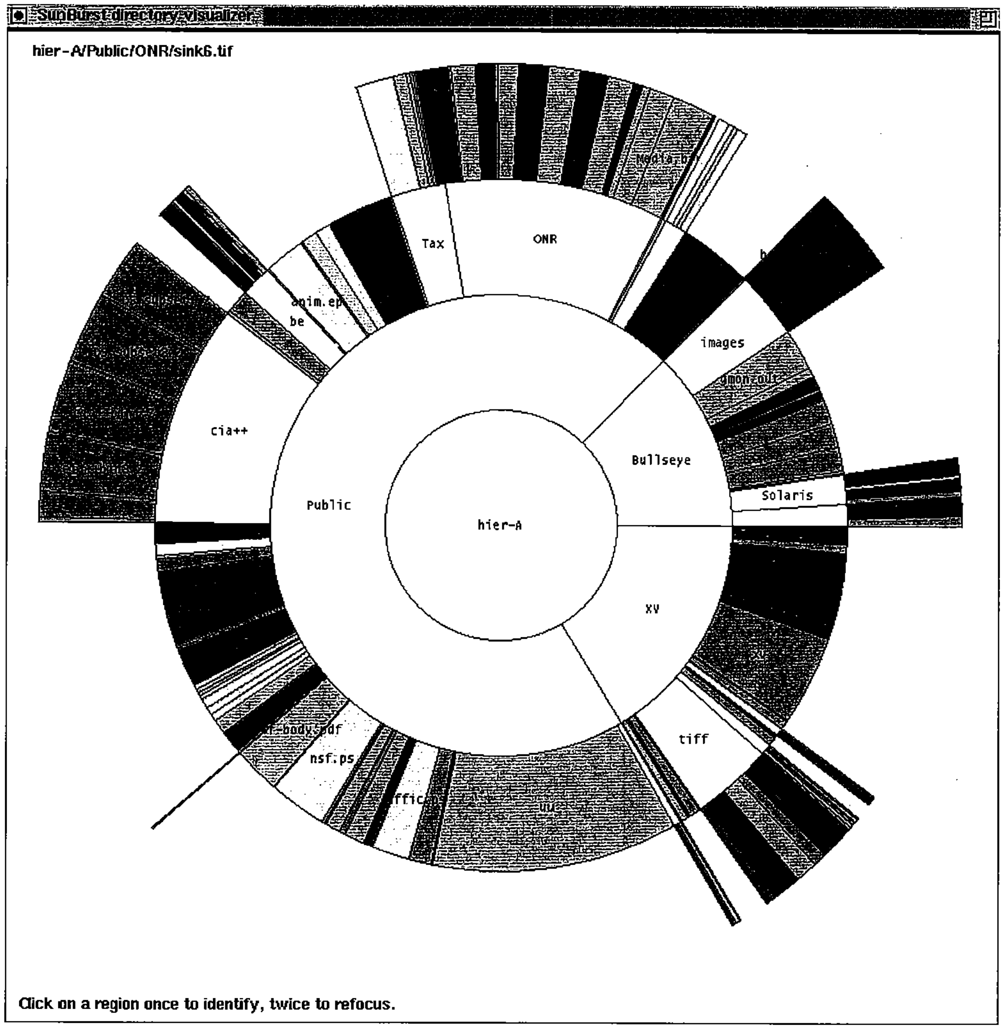

FIGURE 3. Sunburst depiction of file Hierarchy A.

For search tasks such as finding a particular file somewhere in the directory structure, we predict that the SB tool will be better. This is because the SB does explicitly represent directories and its visualization appears to convey the structure of the file system more clearly.

The intent of the present research was to examine whether these presumed differences in the tools would translate into actual performance differences on a variety of tasks dealing with files and directories. We hoped the study would provide us with a better understanding of how such tools could be incorporated into a desktop environment and how to improve the functionality of the tools.

4. Experiment 1

4.1. METHOD

4.1.1. Participants. Thirty-two students at the Georgia Institute of Technology participated in the experiment and were randomly assigned to the conditions described below. The students were primarily computer science majors, but students from other degree programs such as psychology or management also participated. Participants ranged from freshman and sophomore computer science majors taking their third computer science course to senior CS Ph.D. students completing their dissertation to Master's students from other disciplines taking a graduate human–computer interaction course. All participants had experience using computers and were familiar with the notions of files and directories. The notion of using such a visualization system to assist in file-related tasks was new to most of the participants, however.

4.1.2. Materials. Participants viewed the visualizations on a Sun SPARCstation workstation using the TM and SB visualization tools that we had created. Each task was read aloud to the participant as well as being written on a notecard for him or her to review.

4.1.3. Procedure. Each person began a session by learning how to use one of the two visualization tools. The person administering the session read through a prepared tutorial and showed the participant an example directory hierarchy on which the first visualization tool was running. The participant worked through a series of eight example tasks comparable to those used later in the actual study. The session proceeded only when the participant said he or she was comfortable with the system's visualization methodology and its user interface. After this training, the participant performed 16 tasks using the tool on a prepared hierarchy, and then completed a subjective questionnaire concerning that tool. This constituted phase 1 of a session.

Next, the participant trained on the second visualization tool and performed a comparable set of 16 tasks on a different hierarchy using the tool, again followed by a subjective questionnaire. This was phase 2 of a session. At the end of each session we administered a few general preference questions concerning both tools.

Participants' performance on each task was scored as correct or incorrect, with a maximum time limit of 60 s. If no answer was given in the allotted time, that task performance was labeled "timed out". We recorded if the task was accomplished correctly incorrectly or if the time expired with no answer given. We also noted the time taken to reply on both correct and incorrect responses.

In Experiment 1, we created and utilized two file hierarchies, called A and B, consisting of about 500 files and directories each. We used small sets of sample directories and files taken from our own personal file structures and combined them into one new, bigger hierarchy so that they would be typical examples of what a person would manipulate, as opposed to randomly generated files. Also, we made the two hierarchies similar in depth, number of directories and overall structure.

We needed to create the two different hierarchies so that a person would interact with a different one for each of the two visualization tools. We varied the ordering and

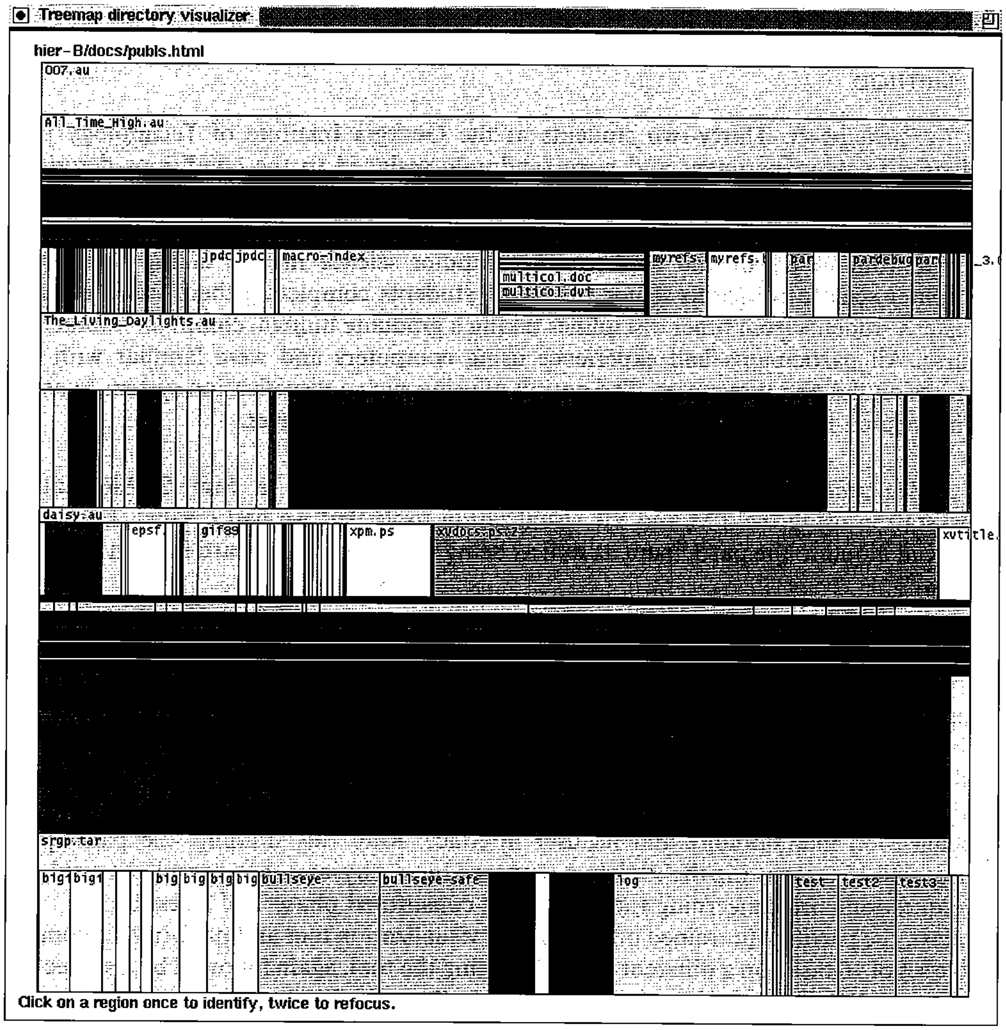

FIGURE 4. Treemap depiction of file Hierarchy B.

conditions across all participants. More specifically, eight people used TM on A in phase 1 then SB on B in phase 2; eight used SB on A then TM on B; eight used TM on B then SB on A; and eight used SB on B then TM on A. The A and B directory structures used in the experiment are shown as depicted by both tools in Figures 2–5. Here, maximum depth is illustrated, showing all files and directories.

The 16 tasks performed using each tool can be grouped into 11 broad categories.

- Identify (name or point out) the largest and second largest files (Questions 1, 2).
- Identify the largest (size) directory (Q3).
- Locate (point out) a file, given its entire path and name (Q4–7).
- Locate a file, given only the file name (Q8, 9).

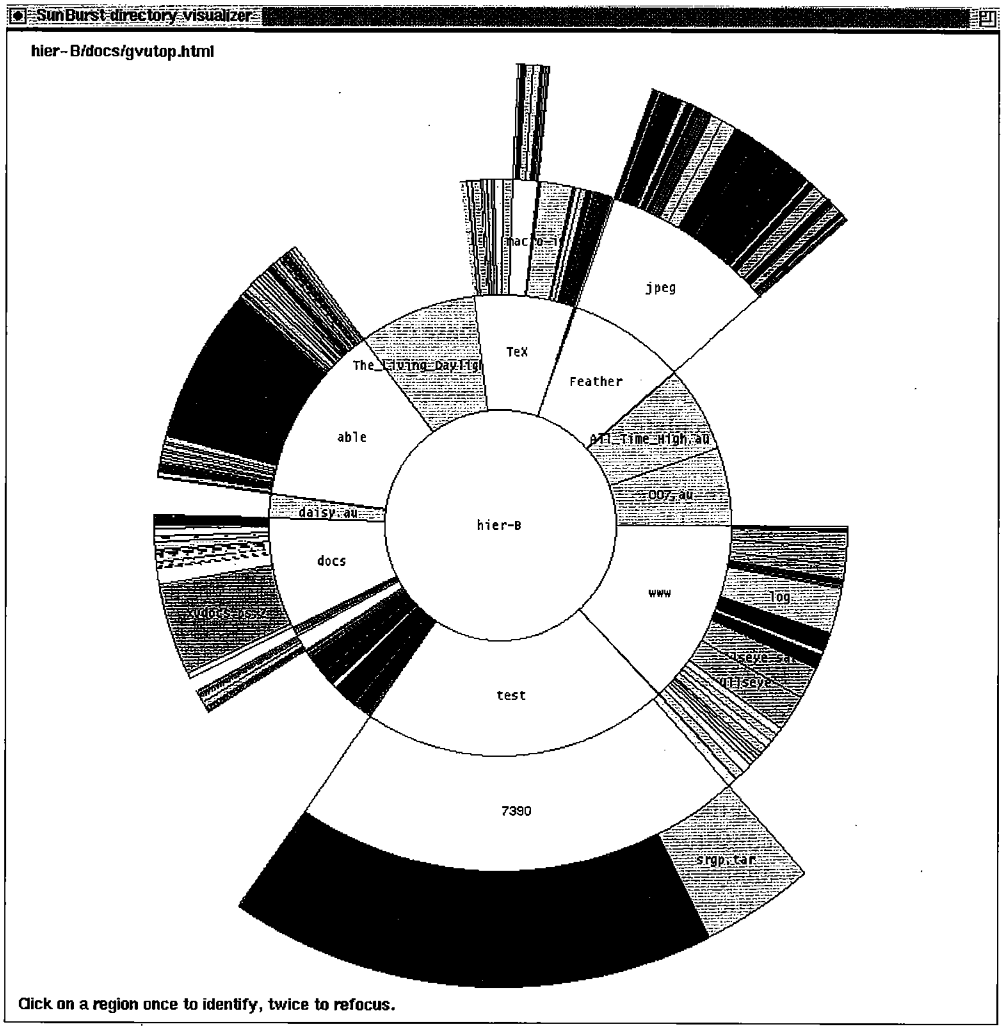

FIGURE 5. Sunburst depiction of file Hierarchy B.

- Identify the deepest subdirectory (Q10).
- Identify a directory containing files of a particular type (Q11).
- Identify a file based on type and size, specifically, the largest file of a particular type (Q12).
- Compare two files by size and identify the larger (Q13).
- Locate two duplicated directory structures containing the same files (Q14).
- Compare two directories by size and identify the larger (Q15).
- Compare two directories by number of files contained and identify the one with more (Q16).

We chose this set of tasks to be representative of typical operations that people perform with or on file systems. For instance, people look for the largest files when they are cleaning up and reclaiming space. Similar clean-up operations are often performed on duplicated directory structures, and clearly, the task of finding a particular file is used in many different activities.

Obviously, we had to identify particular files and/or directories to use in the actual tasks of the experiment. In doing so, we sought to create tasks that were challenging. For example, in designing a task comparing the sizes of two entities, we selected two files or directories that were relatively similar in size. On tasks 4–7, file searches, we intentionally selected files at varying depths within the hierarchy.

At the start of each new task, we reset the visualization to be focused on the root directory with a displayed depth of one, thus making sure that each person started each task from the same point. At the end of each phase of a session, we administered a series of 15 Likert-style questions and four open-format opinion questions concerning the tool just used. At the very end of a session, we asked each participant which tool they preferred, why and to speculate on the two tools' potential utility.

4.1.4. Design. The within-subjects variables were tool (SB vs. TM), phase of tool (first or second set of 16 tasks), and hierarchy (A vs. B). The dependent measures were whether or not a task was done correctly and the time taken to do a task (for tasks that were done correctly).

4.2. RESULTS AND DISCUSSION

The primary performance measure we wished to examine was the number of test questions participants correctly solved as a function of the tools participants used and whether the questions were part of the first or second set of 16 questions. Although the two hierarchies used in the experiment had similar numbers of files and directories, their structure and their constituent files' and directories' names and sizes did vary. Furthermore, preliminary analyses showed an effect on performance as a function of which hierarchy (A or B) participants worked on, so performance has been analysed separately for each hierarchy.

For total number correct on Hierarchy A, there was a main effect of tool (TM vs. SB), $F(1, 28) = 22.78$, $MSE = 5.33$, $p = 0.048$, but not of phase [performance on Hierarchy A when it was involved in the first set of 16 tasks vs. when it was involved in the second set of 16 tasks], $F(1, 28) = 9.03$, $p = 0.20$. An examination of Table 1 indicates that participants were more successful using the SB tool compared to the TM tool. The interaction of tool and phase was not significant, $F(1, 28) = 2.59, p = 0.12$, that is, neither tool showed an advantage in terms of improvement as a function of practice.

For total number correct on Hierarchy B, there was no main effect either of tool, $F(1, 28) = 0.06, MSE = 4.76, p = 0.81$, or phase, $F(1, 28) = 0.06, p = 0.81$. The interaction of tool and phase was also not significant, $F(1, 28) = 1.48$, $p = 0.23$.

To examine the data more closely, we considered participants' performance on a task by task basis. Table 2 presents the total number of correct completions as a function of tool, hierarchy, and phase, clustered by similar styles of tasks. The Appendix includes a table with performance for each task individually.

One of our hypotheses was that TM would facilitate better performance on size-related tasks because it provides more room for each file's or directory's representation.

TABLE 1
Total number of tasks successfully completed as a function of tool, hierarchy and phase in Experiment 1 (maximum = 16; standard deviations in parentheses)

Hierarchy A				Hierarchy B			
Tool	Phase	Correct		Tool	Phase	Correct	
TM ($n = 8$)	1	9.88	(3.23)	TM ($n = 8$)	1	11.50	(2.14)
SB ($n = 8$)	1	12.88	(1.96)	SB ($n = 8$)	1	10.38	(1.69)
TM ($n = 8$)	2	12.25	(1.75)	TM ($n = 8$)	2	10.75	(2.77)
SB ($n = 8$)	2	12.63	(2.00)	SB ($n = 8$)	2	11.50	(2.00)
TM (collapsed across phase)		11.06	(2.79)	TM (collapsed across phase)		11.13	(2.42)
SB (collapsed across phase)		12.75	(1.91)	SB (collapsed across phase)		10.94	(1.88)

Looking at the data, however, this hypothesis is not supported. Six tasks (1–3, 12, 13 and 15) involved some form of size assessment or comparison. Participants using SB performed just as well or better than those using TM on virtually all these tasks. Our intuition about TM supporting size assessment better than SB did not appear to hold.

We believed that SB would facilitate better performance on structure-related tasks, those that involved searching across the directory structure to assess different attributes. While all 16 tasks in some way involved finding files or directories, tasks 4–9 were explicit search-for-file tasks. Performance on these tasks generally was better with SB on Hierarchy A, but this was reversed on Hierarchy B where TM users performed better. Our intuition about SB being better in this respect was not strongly supported.

In addition to task performance, we also tracked and evaluated time to complete tasks. In particular, we analysed the average completion times per task on correctly answered tasks. We include only correctly completed tasks because we want to assess valid, successful performance with the tools. Furthermore, incorrect responses may have been quick guesses or based on some misunderstanding of the visualization.

To perform a statistical analysis of correct completion times, a standardized time to complete each task was calculated. That is, given that the average time for each task varied quite a bit from task to task (which is not surprising given that the tasks were of varying difficulty and complexity), a standardized time was calculated by subtracting the average time on that task (collapsed across phase and tool but not hierarchy since hierarchies are analysed separately) from the participant's time on that task and dividing the result by the standard deviation. Thus, for instance, a participant who took exactly the average time for the task would have a standardized time of 0 for that task. A participant who took two standard deviations shorter than the average for a task would have a standardized time of -2.0 for that task. This approach allowed us to then calculate an average standardized time across all (successfully completed) tasks.

TABLE 2

Average number of participants completing tasks or sets of tasks successfully as a function of tool, hierarchy and phase in Experiment 1. Maximum = 8 for all columns except the 1 + 2 columns where maximum = 16

	Hierarchy A						Hierarchy B					
Tool Phase	TM 1	SB 1	TM 2	SB 2	TM 1 + 2	SB 1 + 2	TM 1	SB 1	TM 2	SB 2	TM 1 + 2	SB 1 + 2
1–3. Find largest files or dir	6.0	7.7	6.7	7.7	12.7	15.3	4.7	4.0	4.0	6.0	8.7	10.0
4–7. Find file via path	6.3	7.0	7.0	7.5	13.3	14.5	7.5	6.8	7.0	6.5	14.5	13.3
8–9. Find file via name	2.5	6.0	5.5	5.5	8.0	11.5	7.5	6.0	8.0	7.5	15.5	13.5
10. Find deepest dir	5.0	8.0	7.0	8.0	12.0	16.0	8.0	8.0	6.0	7.0	14.0	15.0
11. Find dir contents	7.0	8.0	8.0	7.0	15.0	15.0	8.0	7.0	7.0	7.0	15.0	14.0
12. Find via size and type	6.0	7.0	5.0	5.0	11.0	12.0	8.0	7.0	5.0	7.0	13.0	14.0
13,15. Compare files/dirs by size	4.5	4.5	6.0	6.0	10.5	10.5	1.5	2.0	2.5	2.0	4.0	4.0
14. Find duplicate dirs	1.0	2.0	0.0	1.0	1.0	3.0	0.0	2.0	1.0	2.0	1.0	4.0
16. Compare dirs by contents	4.0	6.0	7.0	4.0	11.0	10.0	6.0	6.0	6.0	6.0	12.0	12.0

Note: For each set of tasks (e.g. 1–3 "Find largest files or dir"), the *total* number of successful completions for each task was summed and then divided by the number of tasks that make up the set. Thus, for each set, the average number of successful completions would have a maximum of 8.

Using this approach, the following standardized average times were observed:

Hierarchy A				Hierarchy B			
TM1	SB1	TM2	SB2	TM1	SB1	TM2	SB2
0.391	− 0.031	0.055	− 0.177	0.088	0.034	0.147	− 0.143

For Hierarchy A there was no reliable effect of tool [$F(1, 28) = 1.77$, $MSE = 0.262$, $p = 0.19$], phase [$F(1, 28) = 3.26$, $p = 0.082$], or their interaction [$F(1, 28) = 0.27$, $p = 0.61$] although the means suggest a trend towards faster performance with the SB tool.

For Hierarchy B there was no reliable effect of tool [$F(1, 28) = 1.25$, $MSE = 0.190$, $p = 0.27$], phase [$F(1, 28) = 0.15$, $p = 0.70$], or their interaction [$F(1, 28) = 0.59$, $p = 0.45$] although the means suggest a trend towards faster performance with the SB tool.

Now let us turn our attention to time data with respect to specific types of tasks. Below we examine the task timing results as they relate to our hypotheses about the tools. In the Appendix, we include a table listing average successful completion times for all individual tasks in the experiment.

We hypothesized that TM would afford better performance on file size-related tasks. While this was not borne out for correct task completions (SB users actually performed slightly better), it did generally follow with respect to completion times. The trend was for participants using TM to respond to size-related tasks more quickly as shown in Table 3.

On tasks that involved a participant finding a particular file, we felt that SB would be better because of its explicit structure representation. Table 4 shows the time results for tasks 4–9 that involved such search tasks. A small trend favored faster performance with SB, but the results were relatively consistent across the two tools.

Finally, a number of the later tasks in a session involved directory-related operations such as finding the deepest directory, finding a directory based on particular contents or comparing two directories. Table 5 shows the time results for these tasks. Note how

TABLE 3

Average completion times in seconds (for correct responses only) for file size-related tasks as a function of tool, hierarchy and phase in Experiment 1. Number of correct responses per condition indicated in parentheses

	Hierarchy A				Hierarchy B			
Tool Phase	TM 1	SB 1	TM 2	SB 2	TM 1	SB 1	TM 2	SB 2
1. Find largest file	11.6 (5)	20.7 (7)	11.2 (6)	12.7 (7)	15.2 (4)	20.7 (3)	18.0 (4)	18.2 (5)
2. Find second largest file	10.3 (6)	18.9 (8)	17.5 (6)	14.0 (8)	9.0 (4)	18.3 (3)	14.7 (3)	15.2 (5)
3. Find largest directory	13.3 (7)	15.1 (8)	12.9 (8)	11.9 (8)	16.7 (6)	25.3 (6)	18.8 (5)	19.8 (8)
13. Compare files by size	54.0 (3)	51.8 (4)	37.8 (5)	47.5 (4)	41.0 (1)	— (0)	59.0 (1)	47.0 (2)

TABLE 4

Average completion times in seconds (for correct responses only) for file-finding tasks in Experiment 1 as a function of tool, hierarchy and phase

	Hierarchy A				Hierarchy B			
Tool	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1	1	2	2
4. Find file via path	29.0 (7)	26.0 (6)	22.5 (8)	27.6 (8)	32.6 (8)	35.0 (6)	27.3 (6)	27.5 (6)
5. Find file via path	28.7 (7)	15.1 (8)	21.1 (8)	19.1 (7)	26.2 (8)	27.2 (8)	29.5 (8)	20.6 (8)
6. Find file via path	27.6 (8)	17.4 (8)	20.1 (8)	16.1 (8)	26.6 (8)	21.9 (8)	22.7 (8)	21.6 (7)
7. Find file via path	36.0 (3)	24.8 (6)	36.0 (4)	30.7 (7)	27.7 (6)	25.8 (5)	26.8 (6)	23.6 (5)
8. Find file via name	33.5 (2)	33.0 (6)	37.8 (4)	39.5 (4)	15.6 (8)	26.6 (5)	21.3 (8)	23.7 (7)
9. Find file via name	22.0 (3)	16.0 (6)	28.6 (7)	16.3 (7)	28.1 (7)	36.0 (7)	28.3 (8)	29.4 (8)

TABLE 5

Average completion times in seconds (for correct responses only) for directory attribute-related tasks in Experiment 1 as a function of tool, hierarchy and phase

	Hierarchy A				Hierarchy B			
Tool	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1	1	2	2
10. Find deepest dir	19.4 (5)	21.9 (8)	25.6 (7)	20.2 (8)	28.5 (8)	15.5 (8)	20.5 (6)	14.2 (7)
11. Find dir contents	28.3 (7)	15.6 (8)	19.9 (8)	14.3 (7)	23.1 (8)	22.4 (7)	27.3 (7)	20.6 (7)
15. Compare dirs by size	26.5 (6)	29.2 (5)	26.7 (7)	24.9 (8)	35.0 (2)	22.5 (4)	34.8 (4)	30.5 (2)
16. Compare dirs by contents	31.8 (4)	31.8 (6)	38.6 (7)	23.5 (4)	24.5 (6)	21.5 (6)	29.8 (6)	20.2 (6)

performance with SB was faster for the clear majority of these tasks. The explicit SB representation of directories may have been facilitating the difference noted here.

While the results from Experiment 1 suggest that the Sunburst tool might be more effective overall compared to the Treemap tool, the tasks were performed on relatively small hierarchies. In order to begin an examination of the generality of the findings, Experiment 2 used the same styles of tasks as in Experiment 1 but with larger hierarchies.

5. Experiment 2

5.1. METHOD

5.1.1. Participants. Twenty-eight students from the Georgia Institute of Technology participated in Experiment 2. These students had the same varied backgrounds as those in Experiment 1.

5.1.2. Materials. Experiment 2 utilized the same workstation and task styles as did Experiment 1. The same visualization systems were used as well to display the file hierarchies. The only difference was the file hierarchies themselves. More specifically, we created two new, larger and deeper A and B file hierarchies, each consisting of about 3000 files and directories, roughly six times larger than those of Experiment 1. As in Experiment 1, we built the two directory structures using sample files and directories from our own personal systems, and we made them roughly equivalent in structure and depth.

Figures 6–9 present the two directory structures as seen in each tool when the hierarchy is expanded to maximum depth.

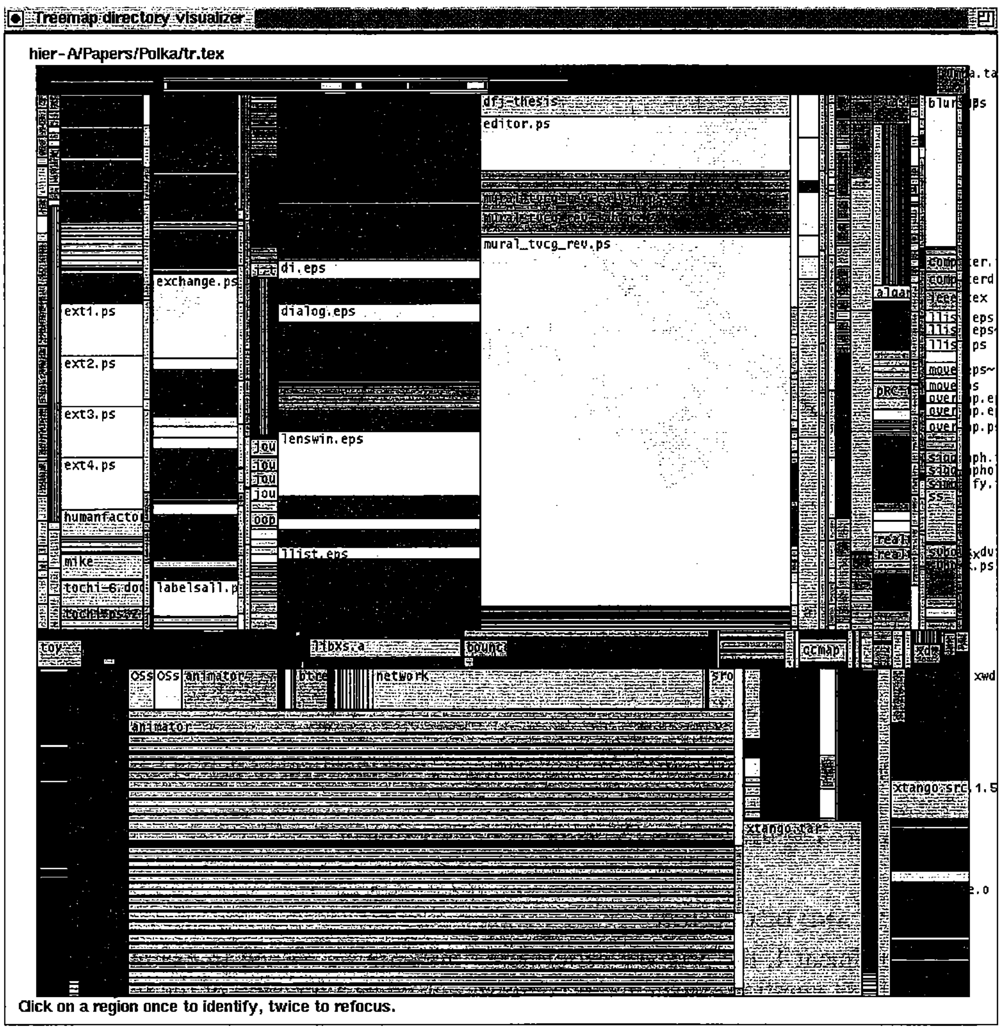

FIGURE 6. Treemap depiction of file Hierarchy A.

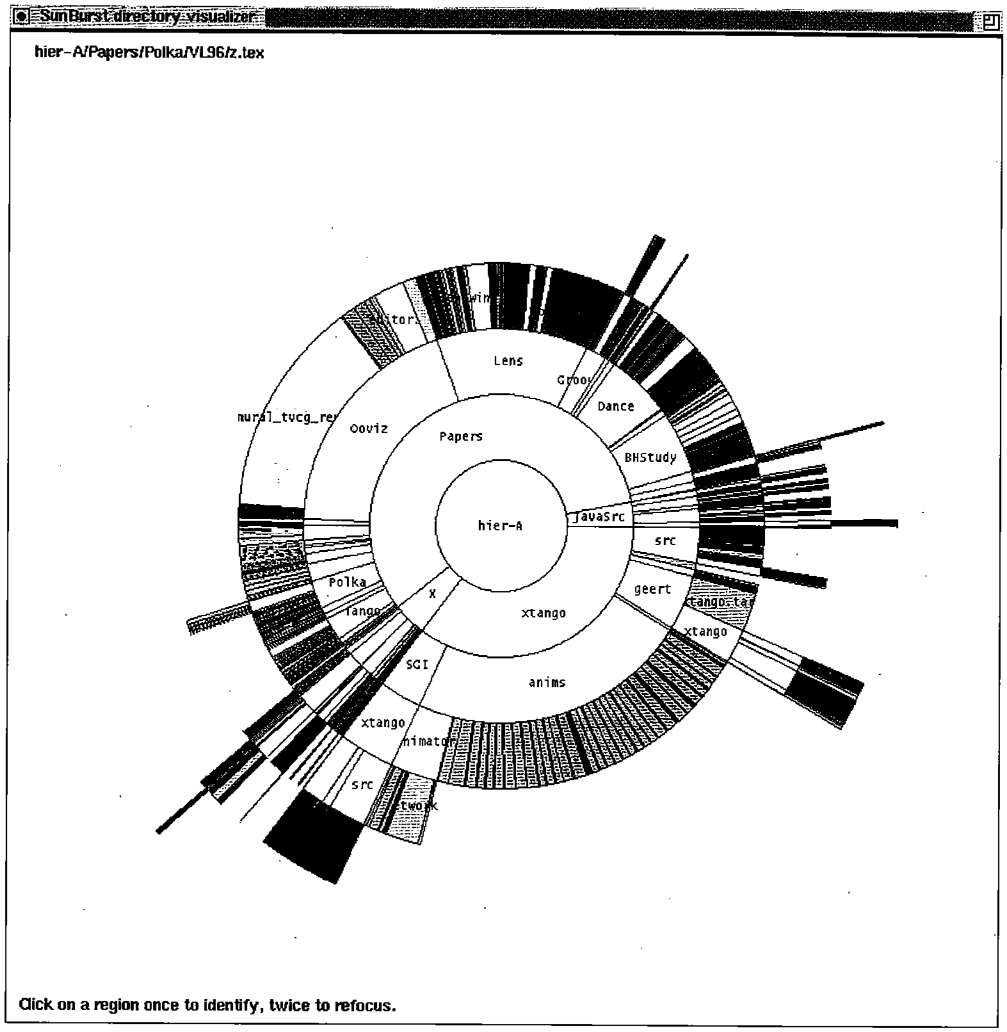

FIGURE 7. Sunburst depiction of file Hierarchy A.

5.1.3. Procedure. The procedure for Experiment 2 was identical to that of Experiment 1.

5.1.4. Design. The design was the same as in Experiment 1.

5.2. RESULTS AND DISCUSSION

As in Experiment 1, the primary performance measure we wished to examine was the number of test questions participants correctly solved as a function of the tools participants used and whether the questions were part of the first or second set of 16 questions. Once again, preliminary analyses showed an effect on performance as a function of which

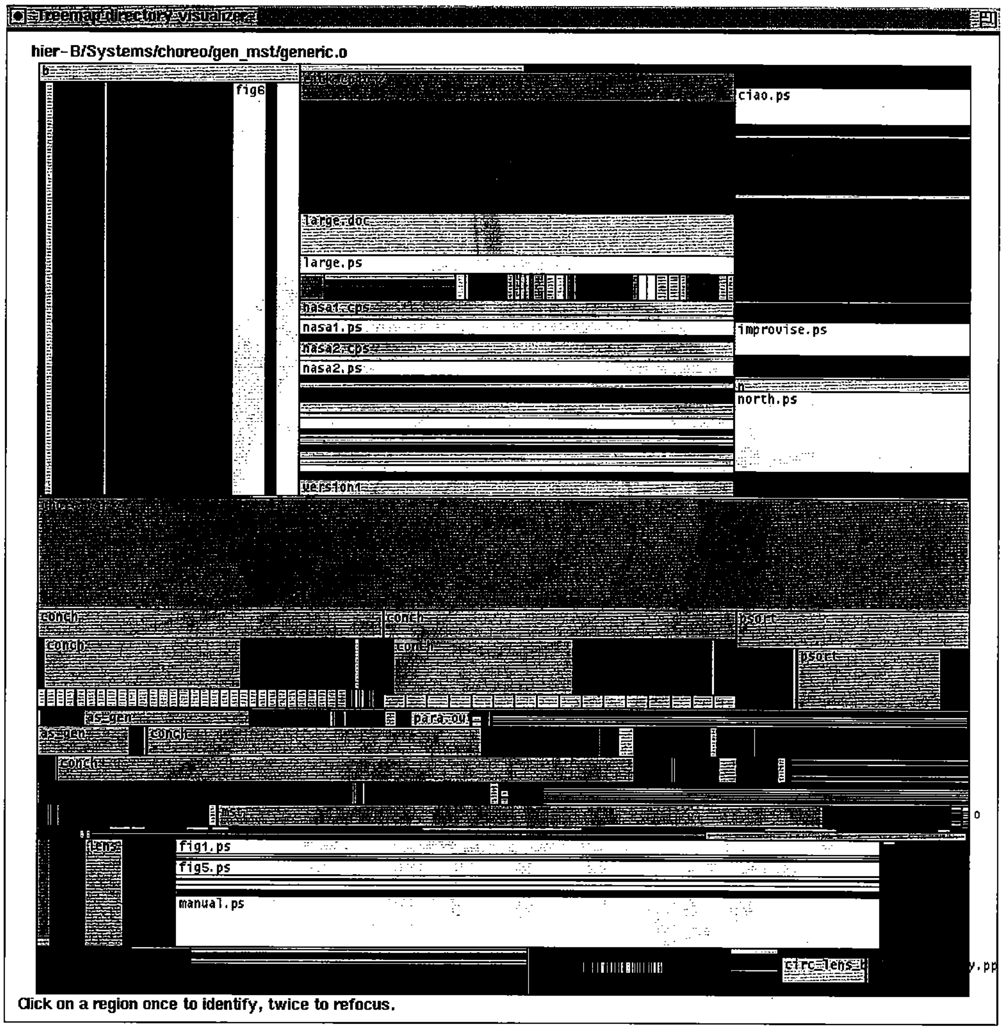

FIGURE 8. Treemap depiction of file Hierarchy B.

hierarchy participants worked on, so performance has been analysed separately for the A and B hierarchies.

For total number correct on Hierarchy A, there was no reliable main effect of tool, $F(1, 24) = 3.07$, $MSE = 2.62$, $p = 0.093$, or phase, $F(1, 24) = 3.94$, $p = 0.059$. However, the interaction of tool and phase was reliable, $F(1, 24) = 18.89$, $p = 0.013$. An examination of Table 6 indicates that while performance with SB stayed relatively stable from the first set of tasks to the second, performance with TM was relatively poor if it was the tool used in the initial phase but was comparable to SB if it was the tool used in the second phase (i.e. after the participant had gained experience first with the SB tool). This suggests that perhaps SB was easier to learn initially while the TM tool may have had some steeper learning costs.

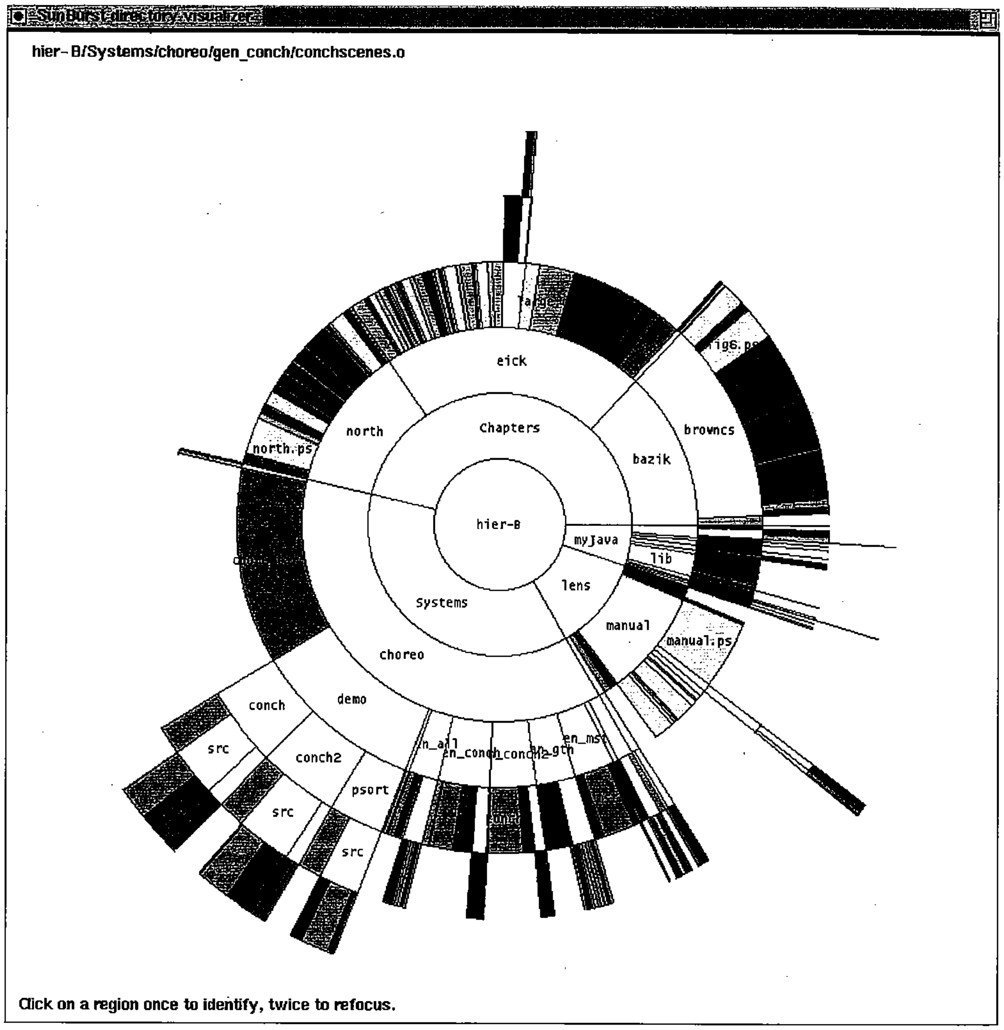

FIGURE 9. Sunburst depiction of file Hierarchy B.

For total number correct on Hierarchy B, there again was no reliable effect of tool, $F(1, 24) = 3.46$, $MSE = 4.55$, $p = 0.075$, or phase, $F(1, 24) = 2.27$, $p = 0.14$. The interaction of tool and phase was also not significant, $F(1, 24) = 2.84$, $p = 0.11$ although the direction of the means is the same as for Hierarchy A.

Taken together, performance on the two hierarchies suggest that SB is an easier tool to learn compared to TM. This is consistent with the findings in Experiment 1.

As was done for Experiment 1, consider performance (number of participants performing a task correctly) for the different types of tasks. Table 7 shows performance grouped by type of task. A complete listing of performance per individual task is included in the Appendix.

TABLE 6

Total number of tasks successfully completed as a function of tool, hierarchy and phase in Experiment 2 (maximum = 16; standard deviations in parentheses)

Hierarchy A				Hierarchy B			
Tool	Phase	Correct		Tool	Phase	Correct	
TM ($n = 7$)	1	8.71	(1.60)	TM ($n = 7$)	1	8.29	(2.14)
SB ($n = 7$)	1	11.43	(1.27)	SB ($n = 7$)	1	11.14	(2.67)
TM ($n = 7$)	2	11.57	(1.27)	TM ($n = 7$)	2	10.86	(1.57)
SB ($n = 7$)	2	11.00	(2.16)	SB ($n = 7$)	2	11.00	(2.00)
TM (collapsed across phase)		10.14	(2.03)	TM (collapsed across phase)		9.57	(2.24)
SB (collapsed across phase)		11.21	(1.72)	SB (collapsed across phase)		11.07	(2.27)

Recall that our hypothesis was that TM users would perform better on direct size-related tasks (1–3, 12, 13, 15) because of TM providing more space to represent sizes and its rectangular representation. This did not hold in Experiment 1 on the smaller hierarchies and again it did hold here as SB users generally performed as well as or better than TM users on these tasks. This difference was strongest on tasks 13 and 15 involving comparisons of size. SB users did quite well, in comparison, on those tasks.

On file search tasks 4–9, performance was relatively mixed as in Experiment 1 with neither tool showing a consistent performance benefit over the other.

Another aspect of the task data to examine is performance relative to ordering or phase. On Hierarchy A, TM participants improved slightly, but consistently, across almost all styles of tasks from phase 1 to phase 2. SB users performed slightly worse in the second phase across most of the different task styles. On Hierarchy B, ordering performance was relatively mixed, but one unique ordering effect did occur: on tasks 8 and 9 (find a file given only its name), participants performed better with SB on phase 1 (SB: 3.5, TM: 0.5), but strongly reversed that trend on phase 2 (SB : 2.0, TM : 6.0). What is interesting about this is that the participants who first used the TM tool did worse in both phases, and those that first used the SB tool did better in both phases, perhaps suggesting that initial use of SB somehow facilitates performance. We discuss this issue further in the next section with regard to the different tools promoting particular search strategies.

We analysed the time to successfully complete tasks using a standardized time calculation as was done in Experiment 1. The values below show the results:

Hierarchy A				**Hierarchy B**			
TM1	SB1	TM2	SB2	TM1	SB1	TM2	SB2
0.267	0.095	− 0.329	0.149	0.353	0.099	− 0.123	− 0.170

TABLE 7

Average number of participants completing tasks or sets of tasks successfully as a function of tool, hierarchy and phase in Experiment 2. Maximum = 7 for all columns except the 1 + 2 columns where maximum = 14

	Hierarchy A						Hierarchy B					
Tool Phase	TM 1	SB 1	TM 2	SB 2	TM 1 + 2	SB 1 + 2	TM 1	SB 1	TM 2	SB 2	TM 1 + 2	SB 1 + 2
1–3. Find largest files or dir	5.0	5.7	5.0	6.0	10.0	11.7	5.7	6.0	6.7	6.0	12.3	12.0
4–7. Find file via path	5.0	7.0	6.3	6.3	11.3	13.3	6.5	6.8	6.3	6.8	12.8	13.5
8–9. Find file via name	3.5	3.0	4.5	3.5	8.0	6.5	0.5	3.5	6.0	2.0	6.5	5.5
10. Find deepest dir	4.0	6.0	6.0	5.0	10.0	11.0	5.0	6.0	5.0	7.0	10.0	13.0
11. Find dir contents	2.0	2.0	5.0	3.0	7.0	5.0	1.0	0.0	0.0	0.0	1.0	0.0
12. Find via size and type	6.0	6.0	6.0	6.0	12.0	12.0	2.0	6.0	3.0	5.0	5.0	11.0
13,15. Compare files/dirs by size	2.5	5.5	5.0	5.0	7.5	10.5	1.5	4.5	4.0	5.5	5.5	10.0
14. Find duplicate dirs	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16. Compare dirs by contents	2.0	4.0	5.0	3.0	7.0	7.0	3.0	5.0	3.0	5.0	6.0	10.0

Note: For each set of tasks (e.g., 1–3 "Find largest files or dir"), the *total* number of successful completions for each task was summed and then divided by the number of tasks that make up the set. Thus, for each set, the average number of successful completions would have a maximum of 7.

For Hierarchy A, there was no reliable effect of tool [$F(1, 24) = 0.79$, $MSE = 0.209$, $p = 0.38$], phase [$F(1, 24) = 2.46$, $p = 0.13$], or their interaction [$F(1, 24) = 3.55$, $p = 0.07$] although the means suggest a trend towards a greater improvement in performance time from phase 1 to phase 2 for the TM tool.

For Hierarchy B, there was no reliable effect of tool [$F(1, 24) = 0.96$, $MSE = 0.165$, $p = 0.34$]. However, there was an effect of phase [$F(1, 24) = 5.86$, $p = 0.02$]. The interaction of tool and phase [$F(1, 24) = 0.45$, $p = 0.51$] was not significant. The means indicate that performance for both tools was faster in the second phase.

We informally analysed the average time for successful task completions for the different types of tasks as well. As in Experiment 1, participants generally performed more quickly using TM on tasks 1–3 that involved file size assessments (see Table 8). On task 13 that involved a comparison of file sizes, however, the performance trend favored SB. These results support our hypothesis about TM being better (faster) for size operations.

On tasks 4–9 involving file finding (see Table 9), the results were relatively mixed, not favoring SB as much as in Experiment 1. It is possible that once the hierarchy grows in size and the circular area of the slices in SB becomes even smaller, identification of individual files becomes more challenging.

As in Experiment 1, participants did generally respond more quickly with SB than with TM on tasks 10, 11, 15 and 16 (see Table 10). These tasks required participants to identify relevant directories and then assess the contents in some way. As noted earlier, the explicit depiction of directories in SB may have been a contributor to those results.

6. Subjective evaluation

In addition to usefulness and the ability to aid user tasks, the success of an information visualization tool also depends on users' subjective opinions of the tool's interface and utility. Recall that after performing the 16 tasks in each phase of a session, participants completed a short questionnaire concerning the particular tool just used. Participants responded to 15 statements using Likert-style replies ranging from 1 (strongly agree) to

TABLE 8

Average completion times in seconds (for correct responses only) for file size-related tasks as a function of tool, hierarchy and phase in Experiment 2. Number of correct responses per condition indicated in parentheses

	Hierarchy A				Hierarchy B			
Tool	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1	1	2	2
1. Find largest file	9.4 (7)	15.7 (7)	6.7 (7)	18.0 (7)	11.8 (6)	20.4 (7)	6.7 (7)	15.0 (6)
2. Find 2nd largest file	6.5 (2)	16.7 (3)	17.0 (1)	20.3 (4)	12.4 (5)	18.6 (7)	10.6 (7)	11.1 (7)
3. Find largest dir	20.7 (6)	21.3 (7)	12.7 (7)	10.0 (7)	11.8 (6)	28.2 (4)	13.3 (6)	17.6 (5)
13. Compare files by size	59.0 (1)	46.8 (4)	54.0 (3)	59.3 (3)	40.0 (1)	34.0 (3)	56.5 (2)	35.4 (5)

TABLE 9

Average completion times in seconds (for correct responses only) for file-finding tasks in Experiment 2 as a function of tool, hierarchy and phase

	Hierarchy A				Hierarchy B			
Tool	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1	1	2	2
4. Find file via path	25.0 (4)	17.1 (7)	18.7 (7)	18.3 (6)	31.2 (6)	14.7 (7)	25.1 (7)	19.7 (7)
5. Find file via path	22.2 (4)	27.1 (7)	15.5 (6)	24.6 (5)	39.1 (7)	19.2 (6)	26.7 (6)	23.7 (7)
6. Find file via path	28.0 (6)	29.6 (7)	18.9 (7)	39.4 (7)	21.2 (6)	21.4 (7)	21.0 (7)	22.7 (7)
7. Find file via path	37.7 (6)	31.6 (7)	36.0 (5)	35.3 (7)	33.9 (7)	24.0 (7)	25.6 (5)	29.5 (6)
8. Find file via name	36.5 (2)	44.5 (2)	16.5 (2)	31.4 (6)	27.0 (1)	50.7 (3)	21.0 (7)	46.5 (2)
9. Find file via name	32.0 (5)	28.0 (4)	26.9 (7)	20.0 (1)	— (0)	43.8 (4)	41.2 (5)	23.5 (2)

TABLE 10

Average completion times in seconds (for correct responses only) for directory attribute-related tasks in Experiment 2 as a function of tool, hierarchy and phase

	Hierarchy A				Hierarchy B			
Tool	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1	1	2	2
10. Find deepest dir	37.0 (4)	28.2 (6)	25.7 (6)	22.0 (5)	35.8 (5)	16.6 (6)	19.8 (5)	15.4 (7)
11. Find dir contents	17.5 (2)	25.0 (2)	33.2 (5)	46.0 (3)	44.0 (1)	— (0)	— (0)	— (0)
15. Compare dirs by size	28.0 (4)	26.4 (7)	23.9 (7)	27.6 (7)	43.0 (2)	36.8 (6)	50.7 (6)	37.0 (6)
16. Compare dirs by contents	60.0 (2)	46.3 (4)	44.6 (5)	33.3 (3)	35.7 (3)	36.2 (5)	47.0 (3)	37.2 (5)

5 (strongly disagree). The results of these surveys, broken out by the two experiments (different hierarchy sizes), are presented in Table 11.

Most of the statements assessed the utility of a tool for a particular type of task. While responses tended to indicate slight agreement of the tools' utility (scores less than 3), the strongest agreement came for the utility of the tools to identify file types (Statements 1, 2). The use of color appears to be effective for this purpose. Responses indicated slight agreement, though less strong, for using the tools to identify and compare size (S3, S4), to find files (S6, S7), and to navigate (S9). Participants again indicated slight agreement that they understood how to use the tools well (S10, S11). As for availability of the tools, subjects slightly agreed that there are times they would like to use the tools (S12), but slightly disagreed that they would like to have the tools available all the time (S13).

The strongest differences in opinion comparing one visualization tool to the other occurred on statements 3–5. Statements 3 and 4 concerned judgments of file size. Respondents felt that they were better able to judge file size with the TM tool, although this opinion was stronger for those people viewing the smaller hierarchy in Experiment 1.

TABLE 11

Subjective opinions averaged across participants in the two experiments. Each person completed the survey for both tools, immediately after using the tool, so there were 32 respondents per statement in Experiment 1 and 28 per statement in Experiment 2. The scale ranged from 1 — "strongly agree" to 5 — "strongly disagree"

Statement	Experiment 1		Experiment 2	
	TM	SB	TM	SB
1. I was able to figure out the types of the files using *toolname*.	1.91	1.72	2.00	2.11
2. I was able to figure out which files were Postscript files using *toolname*.	1.66	1.56	1.75	1.89
3. I was able to compare the sizes of files using the *toolname*.	2.41	3.13	2.50	2.86
4. I was able to figure out the largest file using *toolname*.	2.13	2.75	2.18	2.57
5. I was able to figure out which subdirectories were inside another directory using *toolname*.	2.00	1.50	2.64	1.71
6. I was able to find a particular file using *toolname*.	2.41	2.13	3.07	3.07
7. I was able to find a particular directory using *toolname*.	1.75	1.61	2.75	2.43
8. I was able to identify the files inside a subdirectory using *toolname*.	1.94	2.16	2.43	2.18
9. I was able to navigate around the different directories using *toolname*.	2.16	1.91	2.50	2.21
10. After the training session, I knew how to use *toolname* well.	2.63	2.31	2.64	2.54
11. After all the questions, I knew how to use *toolname* well.	2.50	2.28	2.64	2.32
12. There are definitely times that I would like to use *toolname*.	2.94	2.66	3.07	2.79
13. I would like to have *toolname* available for my use all the time.	3.34	3.22	3.32	3.18
14. I found *toolname* to be confusing to use.	2.94	3.25	3.00	3.36
15. I liked the *toolname* tool.	2.69	2.63	3.07	2.54

This echoed our observation of the experimental sessions, but recall that the objective performance data did not support this opinion. SB performed just as well or better than TM on accuracy, although it tended to take more time per task. Statement 5 concerned the ability to judge if a directory is inside another directory. Respondents felt that the SB tool did this better, particularly for the larger file structure of Experiment 2.

The final statement asserted that the participant "liked the tool". On the smaller hierarchy, the two average scores were virtually identical, 2.69-TM and 2.63-SB, indicating slight agreement with the statement. On the larger hierarchy, respondents more strongly favored the SB tool, 3.07-TM vs. 2.54-SB.

It is interesting to note that the participants who worked on the smaller file hierarchies in Experiment 1 felt more strongly (positive), in general, about the utility of the tools for the different tasks. This occurs for virtually every statement, and is particularly noteworthy for the TM users on statements 5–8. These all involve finding files or directories or looking inside directories. On question 7, in particular, TM users' assessment of the tool's utility to help find files differed by one full point, 1.75 on the smaller structure vs.

2.75 on the larger structure. It simply appears that performing any kind of task with the tools, such as identification, comparison and navigation, grows more difficult as the file hierarchy grows in size.

After each phase, we asked the participants to identify particular aspects of the tool that they liked and disliked. The most common "like" responses were the use of color for file types and the ability to see an overview of all the files. Many SB users also stated that they liked seeing the structural relationships of directories and files. Many TM users disliked its layout methodology. The word "cluttered" was often used and they disliked losing the directory context of files once the display was zoomed in. SB users most often disliked the fact that areas for files became very small in a larger file structure, making individual files difficult to find and see.

As a final survey at the end of the session, we asked each participant which tool they preferred overall and to list the tasks that each tool would be better for. Of the 60 participants across both experiments, 51 favored SB, 8 favored TM and one was unsure. Those favoring TM were split equally with 4 in Experiment 1 and 4 in Experiment 2. Among the 8 people preferring TM, 3 used it in the first phase of a session and 5 used it second.

When asked about the utility of the two tools for different types of tasks, about two-thirds of all the participants said that TM would be better for file size comparisons and about one-third stated that it would not be better for any task, with very few other types of replies. The responses for what SB would be better at varied a bit more. The general theme of the replies was that SB would be better for organizing and finding files, providing a global view of the directory structure and assisting navigation throughout. Particular responses also identified a preference for SB when performing the tasks of judging total space usage, learning about a large, unfamiliar hierarchy and moving files.

Observationally, we did note a clear preference for the SB tool among the participants. They preferred being able to see the entire structure and understanding directory–file relationships. File size/area comparisons with the SB tool did frustrate many participants, however. They often drew arcs with their fingers back to the center to help make size judgments. Comparing two different aspect ratio rectangles in order to evaluate size with TM frustrated some participants just as much though.

7. General discussion

Across the two studies there was a tendency for greater success in tasks in which the SB tool was used, particularly on the initial set of tasks. This suggests that the SB is easier to learn than the TM tool. One possible reason is that the SB tool explicitly depicts directory structures, thus promoting a clearer understanding of the directory structure, without significantly sacrificing the display of file types and sizes. The effects of these sorts of features can be systematically examined in future studies.

The time to correctly complete the various tasks was relatively mixed between the two tools. Nevertheless, the times per task evident in the experiments suggest that TM was faster for finding large files and directories, and SB was faster for identifying named files and directories and performing directory-related operations.

Our hypothesis that TM would be better for size-related tasks clearly was not upheld as SB users performed just as well or better on those tasks, with respect to correct task completion. TM users were generally faster to complete the tasks, however. Our hypothesis that SB would better support structure-oriented tasks generally appeared to hold, but not at a consistent, statistically significant level.

Even though we did not include other styles of file/directory manipulation tools such as the Windows Explorer or a UNIX shell in the study, it is possible to speculate how they would compare to the space-filling techniques examined here. Finding a particular file is facilitated in the other systems through the availability of a "find" operation, either as a specific tool or as a command in the shell. Clearly, a similar operation could be added to either of the space-filling visualizations, so we speculate that all the techniques would be similar in this regard. For attribute-based searches or comparisons such as finding the largest files, identifying directories with certain types of files or finding duplicated directory structures, we speculate that the space-filling visualizations will facilitate better performance. The outline and command-based tools simply do not afford these kinds of tasks. It is possible to do some of these types of operations in a UNIX shell with the *find* command, but this requires an understanding of its complex syntax involving regular expression-based searches, and is generally used only by true UNIX experts. The visualizations provided by Treemap and Sunburst appear to make these complex types of tasks more easily doable by relative novices.

7.1. STRATEGY DEVELOPMENT

In addition to the quantitative results reported above, we also observed the strategies employed by participants in carrying out the tasks. Task performance was clearly influenced by the strategy employed by each participant, and the development and choice of strategies was influenced by the tools used. Below we consider some of the strategies used on certain tasks.

Tasks 4–7 provided the participant with the name and path of a specific file. Participants were asked to locate the file and point it out to the observer. Participants using TM demonstrated three strategies in locating the files. Most began at the top level of the hierarchy, pressed the "deeper" button, double-clicked (focused) on a specific directory and repeated this process, moving further into the hierarchy until they located the file. For instance, if the path of the file in question was *hier-A/public/papers/infoviz.gz*, the participant would start at the top level (*hier-A*), press the "deeper" button to level two, locate the *public* subdirectory, double-click (focus) on it, press the "deeper" button to level three, find the *papers* subdirectory, double-click on it, press the "deeper" button to level four and locate *infoviz.gz*. The second strategy was similar to the first but eliminated the focus step from the process. Participants simply started at the top and pressed the "deeper" button, found the appropriate subdirectory, marked it with a finger, and then pressed "deeper" again. This process was repeated until the file was found. The final strategy utilized the legend and the "max" button to find a file. Participants would match the suffix abbreviation to a specific color on the legend, press the "max" button and begin looking at all files of the color until the correct file was found.

Using SB, participants typically employed one of two strategies in locating the file. The first strategy mirrors the first two methods used with TM. Participants would begin at the top of the hierarchy and press "deeper"; they would then locate the specific subdirectory and either focus on it or mark it with a finger. The participant would again press the "deeper" button, locate the next subdirectory in the path and repeat the process until the file was found. The second strategy made use of the explicit directory-showing nature of SB. Participants pressed the "maximum" button and executed a fan-like search from the center of the hierarchy. Many were able to find the file by this method alone; some, however, would focus (magnify) a specific subdirectory when the file slice was too small to discern the name.

Interestingly, on these tasks we noticed that the strategy used in the first phase of the experiment often influenced the method used to locate files in the second phase. For instance, if the TM tool was used first, the participant would often use the top-deeper-focus approach, rather than the max-fan, when working with SB in phase 2.

Task 13 asked participants to compare the sizes of two different files. Two distinct strategies emerged for the two tools. With TM, most users would locate one file (through a combination of magnifying/focusing on directories and using the "deeper" button) and ascertain its size. The participant would then press the "top" button and repeat the process for the second file. Upon seeing the second file, they would compare it with the memory of the first file's size and venture a guess. A few participants, however, located both files at the maximum level and compared them for size. This latter method was the most popular method using SB. Most users pressed "max", followed the paths, located each file and told the observer the name of the bigger one.

The sizing issues associated with SB caused the participants some confusion. When at the maximum depth, there were a number of times when participants found it very difficult to simply estimate the angles of the slices or to draw the boundaries of the element back to the root directory and estimate a comparison. This was especially difficult in the larger hierarchies of Experiment 2 in which elements on differing levels were too small to see at the maximum view.

Across the variety of tasks, one strategy frequently employed was to immediately move to the global (maximum) view of all files/directories and work from there. Our observation of participants as they carried out these tasks suggested that they used this strategy more with the SB tool, presumably because it afforded an overall depiction of the entire structure, including explicit presentation of directories. Another strategy was to move deeper and deeper into the hierarchy, one level at a time, to complete a task. This second strategy was more commonly used with the TM tool, seemingly to facilitate structural understanding. We did, however, note that this strategy was used more by participants using TM in phase 1 of a session. When TM was used second, participants more often used the "jump to max" strategy, presumably developed in phase 1 with SB. This may help explain the performance variation across the different orders of use for TM, especially in Experiment 2.

7.2. SYSTEM ENHANCEMENTS

In addition to comparing use of the two tools, we wanted to use this evaluation as a form of exploratory study that would provide ideas for future system enhancements and modifications. Participants in the experiments made a number of useful suggestions for improvements to the two tools and their interfaces. These included the following.

- Using mouse-over position to identify file names rather than requiring a single click and bringing the file forward or highlighting it in some way.

- Provide an explicit search-for-filename operation that highlights the file(s)' position in the structure.
- Allow file type colors to be filtered or brushed through the legend, thus allowing highlights of particular file types.
- Provide some form of focus + context or overview and detail (Card, Mackinlay & Shneiderman, 1991) capability to help viewers see more of particular, small files and directories while still viewing the entire hierarchy.
- Integration with a traditional file browser.
- The ability to select files (double-click) and invoke a type-specific command, such as previewing a postscript file.
- The ability to move files through the tool.

8. Conclusion

This article describes a study of two space-filling information visualization techniques for depicting file hierarchies. We compared rectangular (Treemap) and circular (Sunburst) layout methods. The rectangular method draws files inside their respective directories, while the circular method explicitly and separately shows a directory and its contents. The circular method more frequently aided task performance, both in correctness and in time, particularly so for larger file hierarchies. The explicit portrayal of structure appeared to be a primary contributor to this benefit. Overall, participants in the study preferred the circular technique

Certainly, this experiment is only a first step in a careful evaluation of these two techniques. Follow-up studies could examine issues such as alternative display algorithms, different file hierarchies, different types of hierarchies and different tasks. For example, future studies of TM could include versions that show hierarchical containment with borders as discussed in Section 2.

We can carefully analyse the strategies users employ when carrying out various file and directory tasks and attempt to optimize tools for these strategies. For instance, if an important piece of the strategy for locating a file is to find groups of related files, then an analysis of what features (e.g. color, shape) best aid this search can be systematically tested in this context. Future studies could also examine other hierarchical browsing tools such as the Windows Explorer and UNIX shells.

It might be fruitful to construct a GOMS model (Card, Moran & Newell, 1983) to account for the number of physical and mental operators needed to implement each strategy and to see if such a model successfully predicts performance (time to complete a task). If the model and empirical data support the usefulness of certain strategies over others, then it would be important to consider which aspects of the tools play the primary roles in affecting strategy development. Future research could examine the relationship of tool features and strategy development as well as guide efforts to make those strategies easier to implement with the tools and to make it easier for the user to identify the desired information on the screen.

The present study makes an important first step in moving past conjecture about the utility of information visualizations for examining hierarchies, and toward a more thorough and rigorous evaluation methodology. Clearly, further work evaluating information visualization is necessary. New visualization techniques, no matter how innovative, are not valuable unless they provide true utility and assist people with real tasks.

References

ANDREWS, K. & HEIDEGGER, H. (1998). Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. *Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, pp. 9–12. Raleigh Durham, NC, October.

BEAUDOIN, L., PARENT, M.-A. & VROOMEN, L. C. (1996). Cheops: a compact explorer for complex hierarchies. *Proceedings of the IEEE Visualization '96 Conference*, pp. 87–92. San Francisco, CA, October.

CHUAH, M. C. (1998). Dynamic aggregation with circular visual designs. *Proceedings of the IEEE Information Visualization Symposium*, pp. 35–43. Raleigh Durham, NC, October.

CARD, S. K., MORAN, T. P. & NEWELL, A. (1983). *The Psychology of Human–Computer Interaction*. London: Lawrence Earlbaum Associates.

CARD, S. K., MACKINLAY, J. & SHNEIDERMAN, B. Eds. (1998). *Readings in Information Visualization — Using Vision to Think*. Los Altos, CA: Morgan Kaufmann.

JOHNSON, B., & SHNEIDERMAN, B. (1991). Tree-maps: a space filling approach to the visualization of hierarchical information structures. *Proceedings of the IEEE Visualization '91*, pp. 284–291. San Diego, CA, October.

KUMAR, H. P., PLAISANT, C. & SHNEIDERMAN, C. (1997). Browsing hierarchical data with multi-level dynamic queries and pruning. *International Journal of Human-Computer Studies*, **46,** 103–124.

LAMPING, J. & RAO, R. (1996). The hyperbolic browser: a focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, **7,** 33–55.

ROBERTSON, G. G., CARD, S. K. & MACKINLAY, J. D. (1993). Information visualization using 3D interactive animation. *Communications of the ACM*, **36,** 57–71.

SHNEIDERMAN, B. (1992). Tree visualization with tree-maps: a 2-d space-filling approach. *ACM Transactions on Computer Graphics*, **11,** 92–99.

TURO, D. & JOHNSON, B. (1992). Improving the visualization of hierarchies with Treemaps: design issues and experimentation. *Proceedings of the IEEE Visualization '92*, pp. 124–131. Boston, MA.

VAN WIJK, J. & VAN DE WETERING, H. (1999). Cushion treemaps: Visualization of hierarchical information. *Proceedings of the IEEE Symposium on Information Visualization*, pp. 73–78. San Francisco, CA, November.

WATTENBERG, M. (1999). Visualizing the Stock Market. *ACM SIGCHI 1999 Extended Abtracts*, pp. 188–189, Pittsburgh, PA.

WETHERELL, C. & SHANNON, A. (1979). Tidy drawings of trees. *IEEE Transactions on Software Engineering*, **SE-5,** 514–520.

Appendix: Detailed performance results for Experiments 1 and 2

TABLE 12

Number of participants in Experiment 1 (small hierarchies) completing each task successfully as a function of tool, hierarchy and phase. Maximum = 8 for all columns except the 1 + 2 columns where maximum = 16

	Hierarchy A						Hierarchy B					
Tool Phase	TM 1	SB 1	TM 2	SB 2	TM 1 + 2	SB 1 + 2	TM 1	SB 1	TM 2	SB 2	TM 1 + 2	SB 1 + 2
1. Find largest file	5	7	6	7	11	14	4	3	4	5	8	8
2. Find 2nd largest file	6	8	6	8	12	16	4	3	3	5	7	8
3. Find largest dir	7	8	8	8	15	16	6	6	5	8	11	14
4. Find file via path	7	6	8	8	15	14	8	6	6	6	14	12
5. Find file via path	7	8	8	7	15	15	8	8	8	8	16	16
6. Find file via path	8	8	8	8	16	16	8	8	8	7	16	15
7. Find file via path	3	6	4	7	7	13	6	5	6	5	12	10
8. Find file via name	2	6	4	4	6	10	8	5	8	7	16	12
9. Find file via name	3	6	7	7	10	13	7	7	8	8	15	15
10. Find deepest dir	5	8	7	8	12	16	8	8	6	7	14	15
11. Find dir contents	7	8	8	7	15	15	8	7	7	7	15	14
12. Find via size and type	6	7	5	5	11	12	8	7	5	7	13	14
13. Compare files by size	3	4	5	4	8	8	1	0	1	2	2	2
14. Find duplicate dirs	1	2	0	1	1	3	0	2	1	2	1	4
15. Compare dirs by size	6	5	7	8	13	13	2	4	4	2	6	6
16. Compare dirs by contents	4	6	7	4	11	10	6	6	6	6	12	12

TABLE 13

Average completion times for participants in Experiment 1 (small hierarchies) in seconds (for correct responses only) as a function of tool, hierarchy and phase. Number of correct responses per condition indicated in parentheses

	Hierarchy A				Hierarchy B			
Tool Phase	TM 1	SB 1	TM 2	SB 2	TM 1	SB 1	TM 2	SB 2
1. Find largest file	11.6 (5)	20.7 (7)	11.2 (6)	12.7 (7)	15.2 (4)	20.7 (3)	18.0 (4)	18.2 (5)
2. Find 2nd largest file	10.3 (6)	18.9 (8)	17.5 (6)	14.0 (8)	9.0 (4)	18.3 (3)	14.7 (3)	15.2 (5)
3. Find largest dir	13.3 (7)	15.1 (8)	12.9 (8)	11.9 (8)	16.7 (6)	25.3 (6)	18.8 (5)	19.8 (8)
4. Find file via path	29.0 (7)	26.0 (6)	22.5 (8)	27.6 (8)	32.6 (8)	35.0 (6)	27.3 (6)	27.5 (6)
5. Find file via path	28.7 (7)	15.1 (8)	21.1 (8)	19.1 (7)	26.2 (8)	27.2 (8)	29.5 (8)	20.6 (8)
6. Find file via path	27.6 (8)	17.4 (8)	20.1 (8)	16.1 (8)	26.6 (8)	21.9 (8)	22.7 (8)	21.6 (7)
7. Find file via path	36.0 (3)	24.8 (6)	36.0 (4)	30.7 (7)	27.7 (6)	25.8 (5)	26.8 (6)	23.6 (5)
8. Find file via name	33.5 (2)	33.0 (6)	37.8 (4)	39.5 (4)	15.6 (8)	26.6 (5)	21.3 (8)	23.7 (7)
9. Find file via name	22.0 (3)	16.0 (6)	28.6 (7)	16.3 (7)	28.1 (7)	36.0 (7)	28.3 (8)	29.4 (8)
10. Find deepest dir	19.4 (5)	21.9 (8)	25.6 (7)	20.2 (8)	28.5 (8)	15.5 (8)	20.5 (6)	14.2 (7)
11. Find dir contents	28.3 (7)	15.6 (8)	19.9 (8)	14.3 (7)	23.1 (8)	22.4 (7)	27.3 (7)	20.6 (7)
12. Find via size and type	28.7 (6)	26.9 (7)	21.2 (5)	24.0 (5)	21.1 (8)	16.9 (7)	22.8 (5)	17.6 (7)
13. Compare files by size	54.0 (3)	51.8 (4)	37.8 (5)	47.5 (4)	41.0 (1)	— (0)	59.0 (1)	47.0 (2)
14. Find duplicate dirs	50.0 (1)	54.5 (2)	— (0)	31.0 (1)	— (0)	51.0 (2)	60.0 (1)	34.0 (2)
15. Compare dirs by size	26.5 (6)	29.2 (5)	26.7 (7)	24.9 (8)	35.0 (2)	22.5 (4)	34.8 (4)	30.5 (2)
16. Compare dirs by contents	31.8 (4)	31.8 (6)	38.6 (7)	23.5 (4)	24.5 (6)	21.5 (6)	29.8 (6)	20.2 (6)

TABLE 14

Number of participants in Experiment 2 (large hierarchies) completing each task successfully as a function of tool, hierarchy and phase. Maximum = 7 for all columns except the 1 + 2 columns where maximum = 14

	Hierarchy A						Hierarchy B					
Tool	TM	SB	TM	SB	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1 + 2	1 + 2	1	1	2	2	1 + 2	1 + 2
1. Find largest file	7	7	7	7	14	14	6	7	7	6	13	13
2. Find 2nd largest file	2	3	1	4	3	7	5	7	7	7	12	14
3. Find largest dir	6	7	7	7	13	14	6	4	6	5	12	9
4. Find file via path	4	7	7	6	11	13	6	7	7	7	13	14
5. Find file via path	4	7	6	5	10	12	7	6	6	7	13	13
6. Find file via path	6	7	7	7	13	14	6	7	7	7	13	14
7. Find file via path	6	7	5	7	11	14	7	7	5	6	12	13
8. Find file via name	2	2	2	6	4	8	1	3	7	2	8	5
9. Find file via name	5	4	7	1	12	5	0	4	5	2	5	6
10. Find deepest dir	4	6	6	5	10	11	5	6	5	7	10	13
11. Find dir contents	2	2	5	3	7	5	1	0	0	0	1	0
12. Find via size and type	6	6	6	6	12	12	2	6	3	5	5	11
13. Compare files by size	1	4	3	3	4	7	1	3	2	5	3	8
14. Find duplicate dirs	0	0	0	0	0	0	0	0	0	0	0	0
15. Compare dirs by size	4	7	7	7	11	14	2	6	6	6	8	12
16. Compare dirs by contents	2	4	5	3	7	7	3	5	3	5	6	10

TABLE 15

Average completion times for participants in Experiment 2 (large hierarchies) in seconds (for correct responses only) as a function of tool, hierarchy and phase. Number of correct responses per condition indicated in parentheses

	Hierarchy A				Hierarchy B			
Tool Phase	TM 1	SB 1	TM 2	SB 2	TM 1	SB 1	TM 2	SB 2
1. Find largest file	9.7 (7)	15.7 (7)	6.7 (7)	18.0 (7)	11.8 (6)	20.4 (7)	6.7 (7)	15.0 (6)
2. Find 2nd largest file	6.5 (2)	16.7 (3)	17.0 (1)	20.3 (4)	12.4 (5)	18.6 (7)	10.6 (7)	11.1 (7)
3. Find largest dir	20.7 (6)	21.3 (7)	12.7 (7)	10.0 (7)	11.8 (6)	28.2 (4)	13.3 (6)	17.6 (5)
4. Find file via path	25.0 (4)	17.1 (7)	18.7 (7)	18.3 (6)	31.2 (6)	14.7 (7)	25.1 (7)	19.7 (7)
5. Find file via path	22.2 (4)	27.1 (7)	15.5 (6)	24.6 (5)	39.1 (7)	19.2 (6)	26.7 (6)	23.7 (7)
6. Find file via path	28.0 (6)	29.6 (7)	18.9 (7)	39.4 (7)	21.2 (6)	21.4 (7)	21.0 (7)	22.7 (7)
7. Find file via path	37.7 (6)	31.6 (7)	36.0 (5)	35.3 (7)	33.9 (7)	24.0 (7)	25.6 (5)	29.5 (6)
8. Find file via name	36.5 (2)	44.5 (2)	16.5 (2)	31.4 (6)	27.0 (1)	50.7 (3)	21.0 (7)	46.5 (2)
9. Find file via name	32.0 (5)	28.0 (4)	26.9 (7)	20.0 (1)	— (0)	43.8 (4)	41.2 (5)	23.5 (2)
10. Find deepest dir	37.0 (4)	28.2 (6)	25.7 (6)	22.0 (5)	35.8 (5)	16.6 (6)	19.8 (5)	15.4 (7)
11. Find dir contents	17.5 (2)	25.0 (2)	33.2 (5)	46.0 (3)	44.0 (1)	— (0)	— (0)	— (0)
12. Find via size and type	22.3 (6)	21.7 (6)	13.2 (6)	18.7 (6)	25.5 (2)	27.8 (6)	13.7 (3)	15.4 (5)
13. Compare files by size	59.0 (1)	46.8 (4)	54.0 (3)	59.3 (3)	40.0 (1)	34.0 (3)	56.5 (2)	35.4 (5)
14. Find duplicate dirs	— (0)	— (0)	— (0)	— (0)	— (0)	— (0)	— (0)	— (0)
15. Compare dirs by size	28.0 (4)	26.4 (7)	23.9 (7)	27.6 (7)	43.0 (2)	36.8 (6)	50.7 (6)	37.0 (6)
16. Compare dirs by contents	60.0 (2)	46.3 (4)	44.6 (5)	33.3 (3)	35.7 (3)	36.2 (5)	47.0 (3)	37.2 (5)