An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures

John Stasko[1], Richard Catrambone[2], Mark Guzdial[1] & Kevin McDonald[1]

Graphics, Visualization and Usability Center
[1]College of Computing
[2]School of Psychology
Georgia Institute of Technology
Atlanta, GA 30332-0280
stasko@cc.gatech.edu

Technical Report GIT-GVU-00-03
February 2000

Abstract

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 systems 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.

Keywords: Information visualization, Treemap, hierarchy, evaluation, empirical study

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.

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 and Johnson[JS91, Shn92], 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 3-D, shaded "cushion" Treemap to help illustrate structure[vWvdW99].

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[TJ92]. 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 five minutes 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 versus 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."

2 Visualization Tools

The first visualization tool employed in the experiment uses the rectangular, space-filling Treemap technique[Shn92]. We implemented the Treemap algorithm described in [JS91] 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.[1] 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 around directories is added. While that addition can help convey structure, it becomes problematic for larger structures in that the padding uses up too much space, and the room for files becomes too small.

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 only be used 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 being created by Chuah[Chu98] and Andrews[AH98]. In general, this 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 both tools, but on different hierarchies in order to avoid any learning effects due to working on the same hierarchy

[1] A brush-over technique could be implemented instead, but that was not included for this experiment.

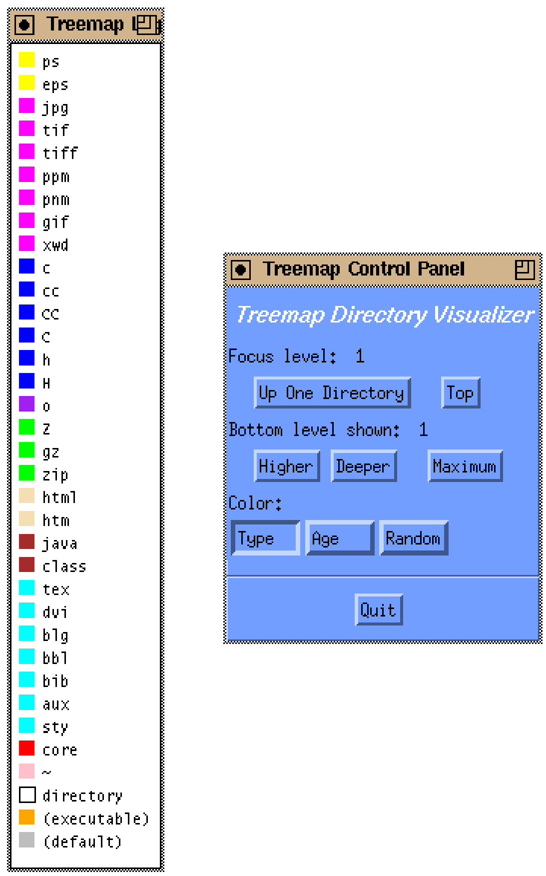

Figure 1: Legend and control panel windows for the Treemap tool. Sunburst's are identical.

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.

Each tool is presumed to have certain strengths that would aid users in carrying out different types of tasks. For instance, the TM tool may 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. Nevertheless, each tool harbors potential problems when comparing sizes: With TM, the viewer may have to compare rectangles of different aspect ratios and with SB, the viewer must compare angular slices. Alternatively, the SB tool may be better for search tasks such as finding a particular file somewhere in the directory structure. This is because the SB does explicitly represent directories and its visualization appears to convey the structure of the file system better. 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.

Furthermore, 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 CS 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. However, the notion of using such a visualization system to assist in file-related tasks was new to most of the participants.

4.1.2 Materials

Participants viewed the visualizations on a Sun SPARCstation workstation using the TM and SB visualization tools that we had created, and that were described earlier in the article. Each task was read aloud to the participant as well as being written on a notecard for them 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 was 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 considered phase 2 of a session. At the end of each session we administered a few general preference questions concerning the tools.

Participants' performance on each task was scored as correct or incorrect, with a maximum time limit of 60 seconds. If no answer was given in the allotted time, that task performance was labelled "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 sample directories and files taken from our own personal file structures so that they would be typical examples of what a person would manipulate, as opposed to randomly generated files. We needed to create the two different structures so that a person would interact with a different one for each of the two visualization tools. We varied the ordering and 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 through 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)
- Identify of 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)

Obviously, we had to identify particular files and/or directories to use in the actual tasks of the experiment. In general, we sought to create tasks that were relatively 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 of the same type, a file search, 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 value.

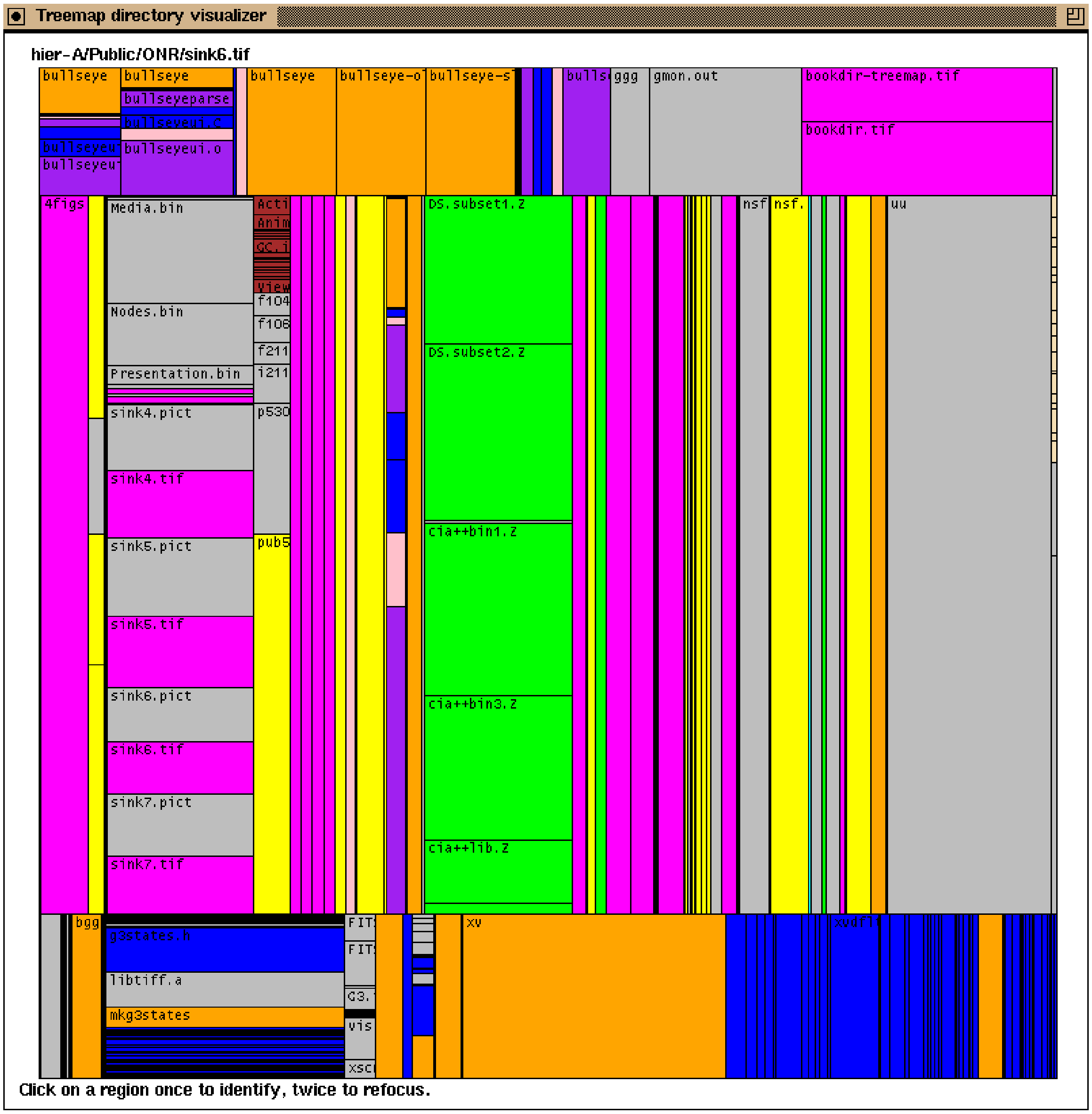

Figure 2: Treemap depiction of file Hierarchy A.

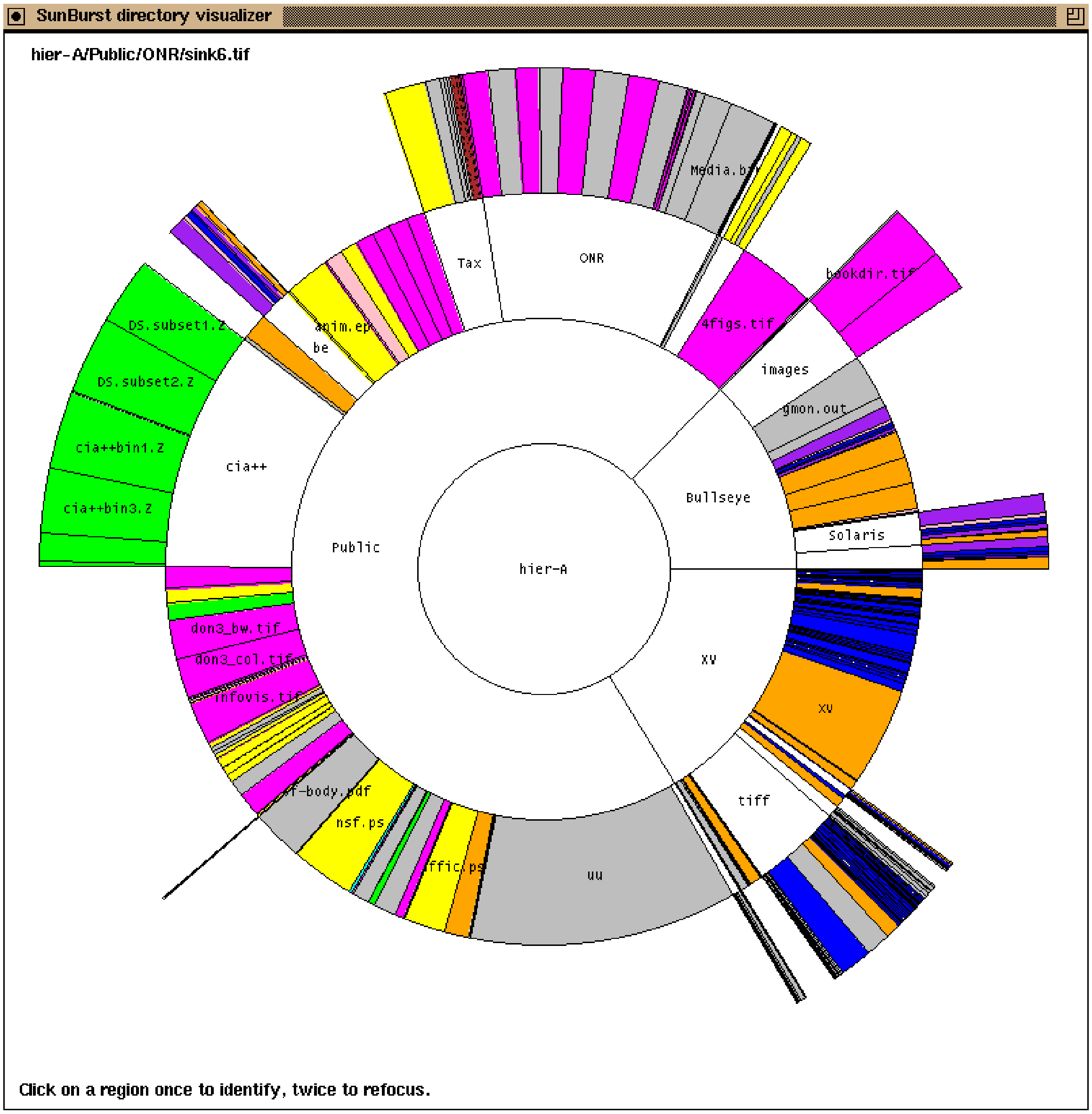

Figure 3: Sunburst depiction of file Hierarchy A.

Figure 4: Treemap depiction of file Hierarchy B.

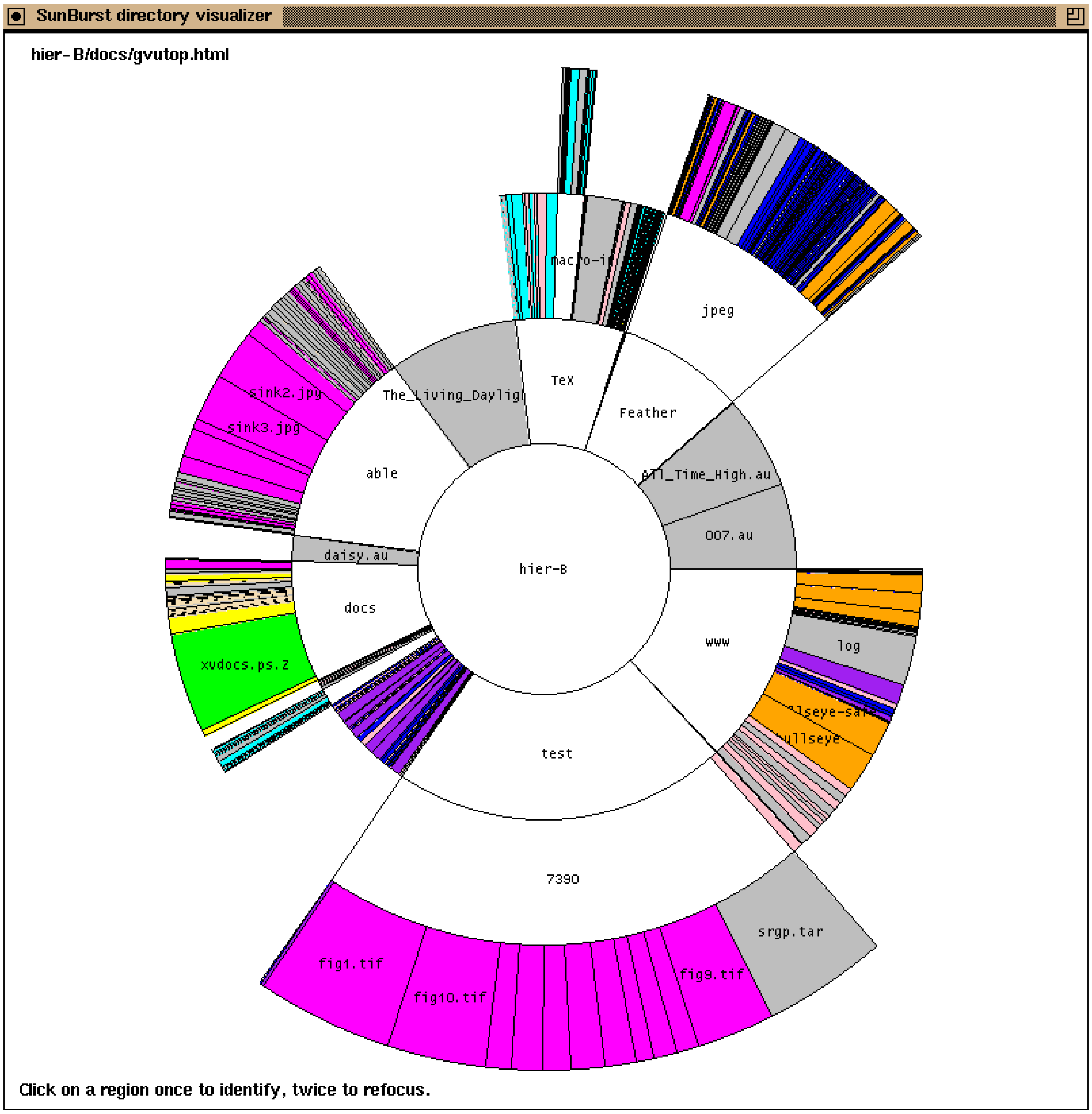

Figure 5: Sunburst depiction of file Hierarchy B.

Hierarchy A			
Tool	**Phase**	**Correct**	
TM ($n = 8$)	1	9.88	(3.23)
SB ($n = 8$)	1	12.88	(1.96)
TM ($n = 8$)	2	12.25	(1.75)
SB ($n = 8$)	2	12.63	(2.00)
TM (collapsed across phase)		11.06	(2.79)
SB (collapsed across phase)		12.75	(1.91)

Hierarchy B			
Tool	**Phase**	**Correct**	
TM ($n = 8$)	1	11.50	(2.14)
SB ($n = 8$)	1	10.38	(1.69)
TM ($n = 8$)	2	10.75	(2.77)
SB ($n = 8$)	2	11.50	(2.00)
TM (collapsed across phase)		11.13	(2.42)
SB (collapsed across phase)		10.94	(1.88)

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

4.1.4 Design

The within-subjects variables were tool (SB vs TM), position of tool (first or second), 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 varied substantially. Furthermore, preliminary analyses showed an effect on performance as a function of which hierarchy (A or B) participants worked on, so for simplicity here, performance has been analyzed separately for each hierarchy.

For total number correct on Hierarchy A, there was a main effect of tool [TM versus SB], $F(1, 28) = 22.78$, $MSE = 5.33$, $p = .048$, but not of position [first set of 16 tasks versus second set], $F(1, 28) = 9.03$, $p = .20$. 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 position was not significant, $F(1, 28) = 2.59$, $p = .12$, 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, $F(1, 28) = 0.06$, $MSE = 4.76$, $p = .81$, or position, $F(1, 28) = 0.06$, $p = .81$. The interaction of tool and position was also not significant, $F(1, 28) = 1.48$, $p = .23$.

To examine the data more closely, we can consider participants' performance on a task by task basis. Table 2 lists the total number of correct completions as a function of tool, hierarchy, and phase.

Focusing on Hierarchy A first, tasks 7-10 were important contributors toward SB's performance benefit, particularly in a session's first phase. These questions all involved locating a file or directory. Performance with SB also was better, though not as strongly, on tasks 1-2 for finding the two largest files. With respect to order across phases, performance with TM improved notably on tasks 8-10 and 16. Performance with SB across the two phases was quite comparable on all tasks.

On Hierarchy B, performance was quite similar with the two tools on virtually all tasks, as well as across the two phases with a particular tool.

Six tasks (1-3, 12-13 and 15) involved some form of size comparison or assessment. Across the two hierarchies, on these 12 tasks (six per hierarchy), performance with SB was at least better than

	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	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 2: Number of participants 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.

or equal to that with TM on every task. Our intuition about TM supporting size assessment better than SB did not appear to hold.

All 16 tasks in some way involved finding files or directories. Tasks 4-9, however, were explicit search-for-file tasks. Performance on these tasks generally was better with SB on Hierarchy A, and slightly better with TM on Hierarchy B. Our intuition about SB being better in this respect received limited support at best.

Table 3 lists the average completion times, per task, on correctly answered tasks. Because we include only successful task completion times, each value in the table may be based on times from anywhere from 1-8 participants, with a few tasks having no correct responses, as indicated by a dash. (For actual correct responses per task, consult Table 2.) Consequently, a statistical analysis of these results would be inappropriate. Here, we simply note values of interest, and will only identify larger differences with relatively many correct replies per task.

On tasks 1-3 involving size determination, participants generally replied more quickly with TM than with SB. On tasks 4-9 involving file finding, this trend was reversed with SB supporting faster replies. Both these trends matched our earlier intuitions about the affordances of each visualization. On tasks 5-16 the broad trend is faster correct replies with SB, especially when comparing the tools in the first phase of a session. Relatively large time differences favoring SB occurred for tasks 5, 6, 7 and 11 in phase 1 on Hierarchy A, tasks 9 and 16 on phase 2 of Hierarchy A, and task 10 of Hierarchy B. The only large difference favoring TM was task 8 on phase 1 of Hierarchy B.

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 tasks as in Experiment 1 but with larger hierarchies.

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

Table 3: Completion times in seconds (for correct responses only) as a function of tool, hierarchy, and phase.

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 question materials as did Experiment 1. The same visualization systems were used as well to display the file hierarchies. The only difference was in the size of the hierarchies. The A and B hierarchies each consisted 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.

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

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

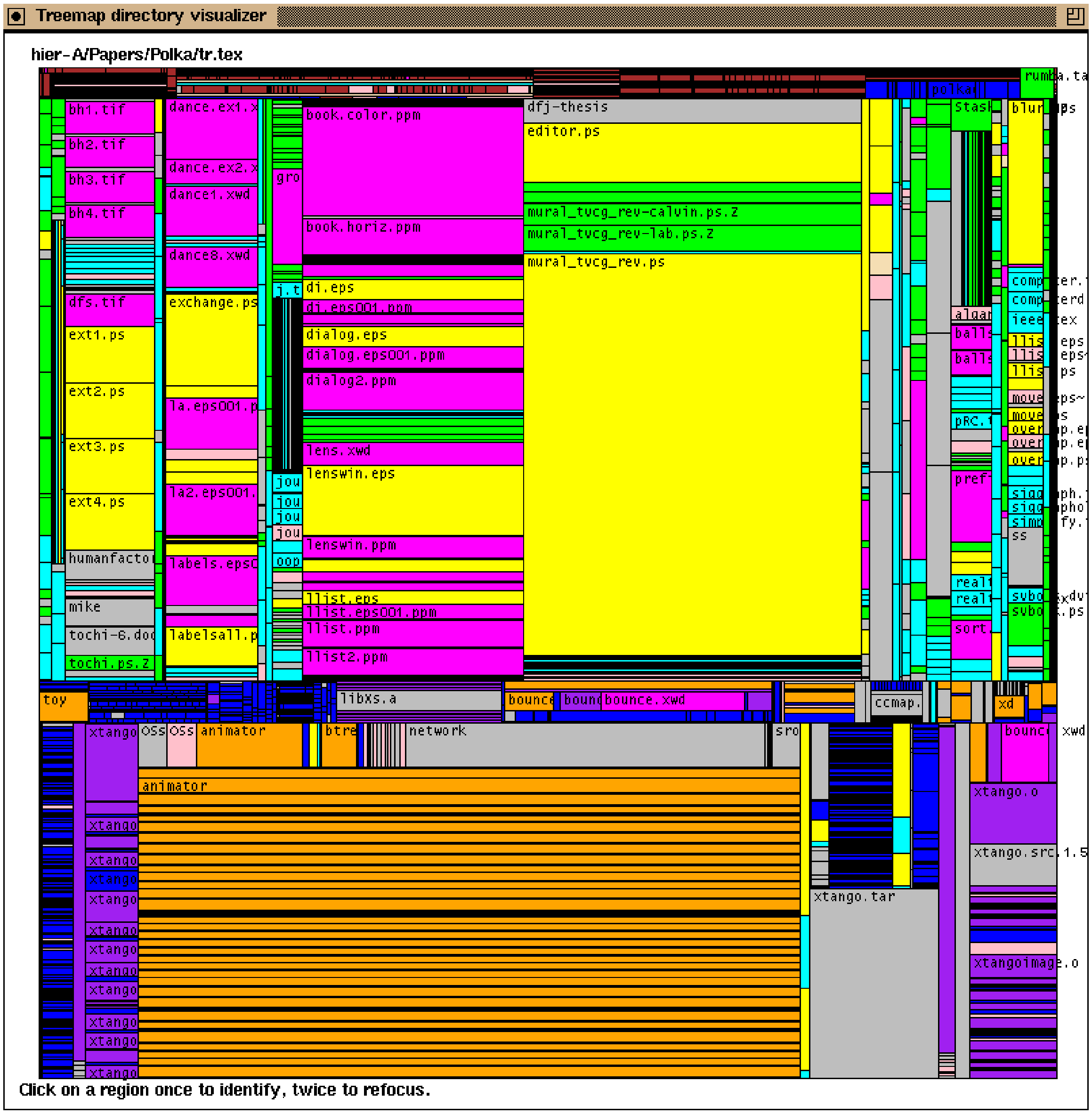

Figure 6: Treemap depiction of file Hierarchy A.

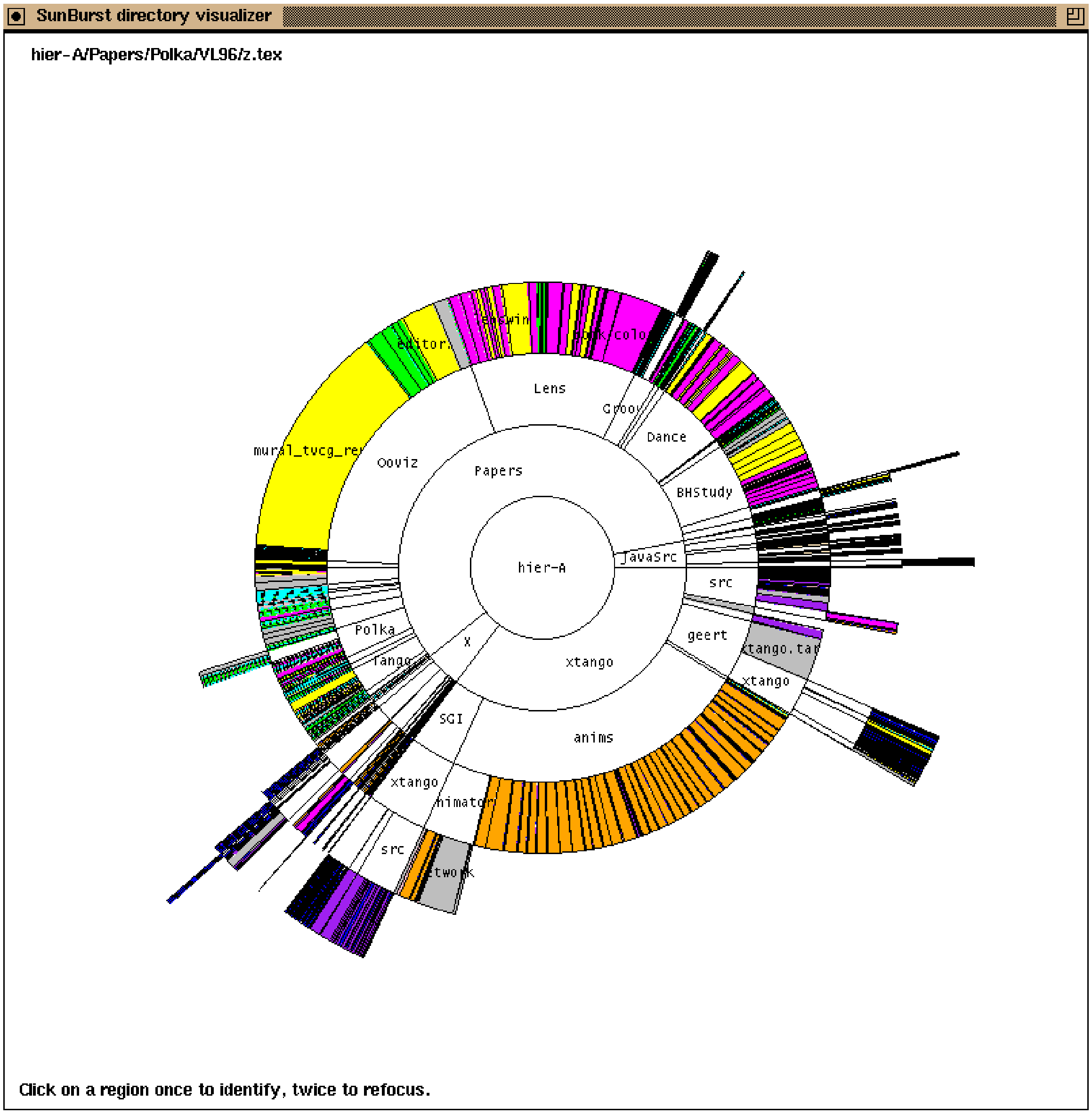

Figure 7: Sunburst depiction of file Hierarchy A.

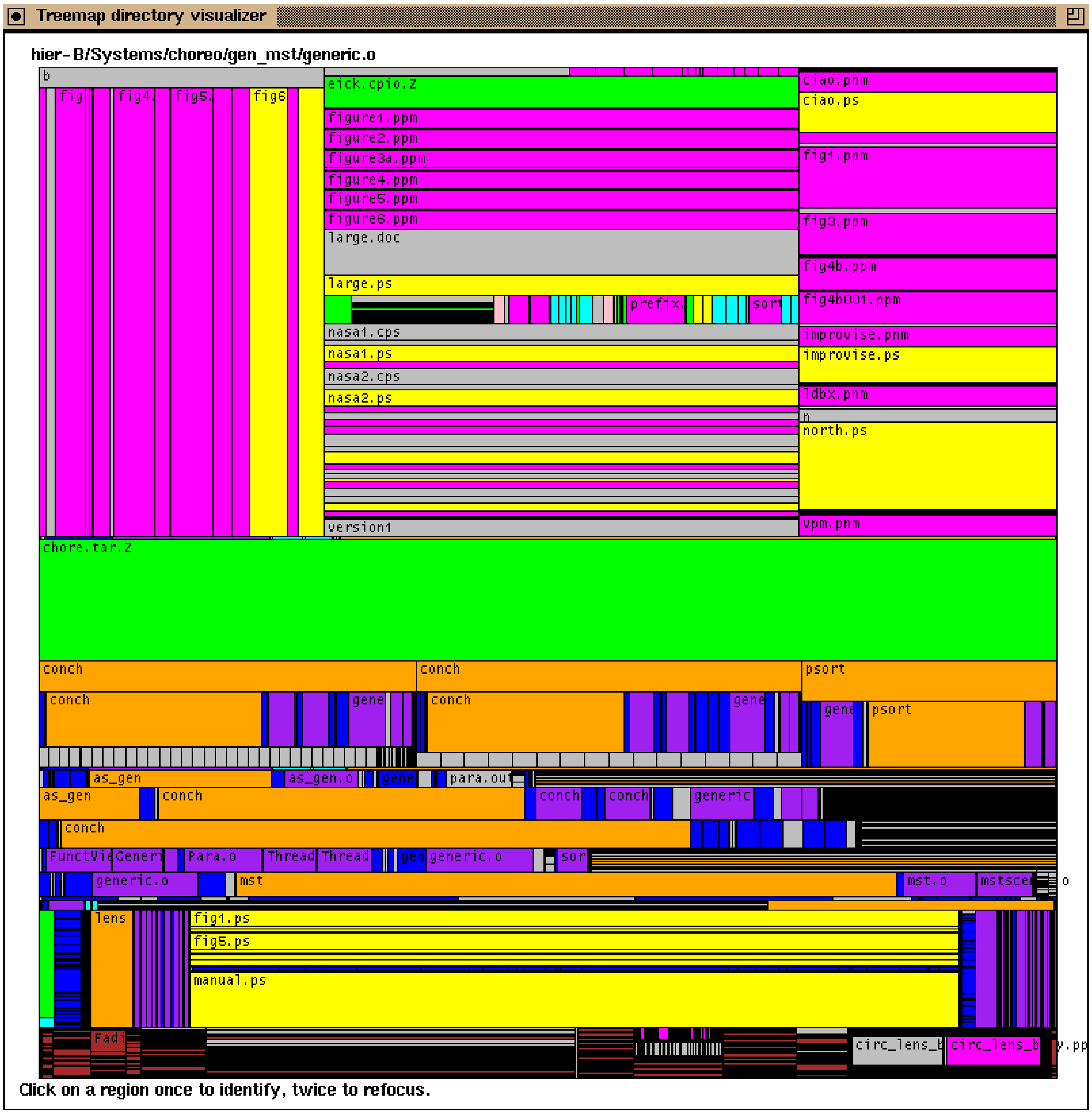

Figure 8: Treemap depiction of file Hierarchy B.

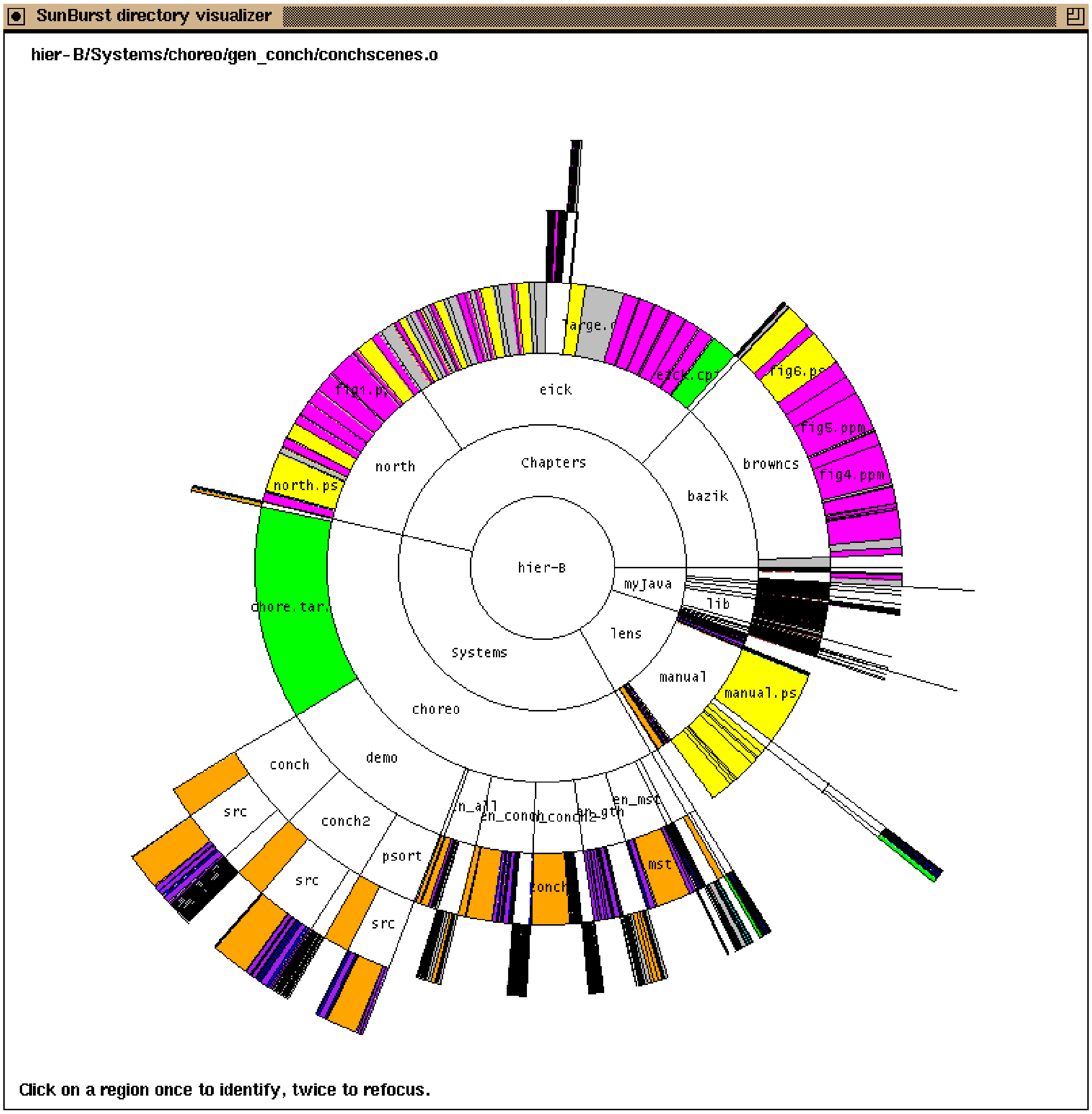

Figure 9: Sunburst depiction of file Hierarchy B.

Hierarchy A				Hierarchy B			
Tool	**Phase**	**Correct**		**Tool**	**Phase**	**Correct**	
TM ($n = 8$)	1	8.71	(1.60)	TM ($n = 8$)	1	8.29	(2.14)
SB ($n = 8$)	1	11.43	(1.27)	SB ($n = 8$)	1	11.14	(2.67)
TM ($n = 8$)	2	11.57	(1.27)	TM ($n = 8$)	2	10.86	(1.57)
SB ($n = 8$)	2	11.00	(2.16)	SB ($n = 8$)	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)

Table 4: Total number of tasks successfully completed as a function of tool, hierarchy, and phase (Maximum = 16; standard deviations in parentheses).

showed an effect on performance as a function of which hierarchy participants worked on, so for simplicity, performance has been analyzed separately for the A and B hierarchies.

For total number correct on Hierarchy A, there were marginal main effects of both tool, F (1, 24) = 3.07, MSE = 2.62, p = .093, and position, F (1, 24) = 3.94, p = .059. The interaction of tool and position was reliable, F (1, 24) = 18.89, p = .013. An examination of Table 4 indicates that there was a trend favoring performance when participants used the SB tool as well as a tendency for superior performance on the second set of tasks. However, the most striking feature of Table 4 is that while performance with SB stayed relatively stable from the first set of tasks to the second, performance with TM was relatively poor in the initial session but improved if used in the second session (i.e., after the participant had gained experience first with the SB tool). This suggests that perhaps SB is easier to learn initially while the TM tool may have some steeper learning costs.

For total number correct on Hierarchy B, there was a marginal main effect of tool, F (1, 24) = 3.46, MSE = 4.55, p = .075, but not of position, F (1, 24) = 2.27, p = .14. The interaction of tool and position was also not significant, F (1, 24) = 2.84, p = .11.

Taken together, performance on the two hierarchies suggest that SB is somewhat more effective than TM overall and that some of this benefit may be due to easier learning of the SB tool.

Performance (number of participants performing a task correctly) in the experiment on a task by task basis is shown in Table 5. Recall that our intuition was that TM users would perform better on direct size-related tasks (1-3, 12-13, 15) and SB users would perform better on the search tasks (4-9).

Focusing on Hierarchy A, performance favored SB relatively consistently on almost all tasks across the different phases. The difference was most notable in phase 1 on tasks 4-5 and 15. In phase 2, task 9 went strongly against this trend, with a 7-1 performance benefit for TM. With respect to ordering, TM participants improved slightly, but consistently, across almost all tasks from phase 1 to phase 2. SB users performed roughly equally, per question, across the two phases.

On Hierarchy B, the latter tasks 10, 12-13, and 15-16 were the more notable contributors in favor of SB. A unique ordering effect occurred here as well: On tasks 8 and 9 (find a file given only name), participants performed better with SB on phase 1 (1-3 and 0-4), but strongly reversed that trend on phase 2 (7-2 and 5-2). 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. We discuss this further in the next section with regard to the different tools promoting particular search strategies.

As in Experiment 1, on tasks 1-3, 12-13, and 15 involving direct assessment or comparison of size, SB users performed as well or better than TM, countering our intuition. Tasks 4-9, finding a named file, favored SB in phase 1 but often favored TM in phase 2, similar to the peculiar ordering

	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 5: Number of participants 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	TM	SB	TM	SB	TM	SB	TM	SB
Phase	1	1	2	2	1	1	2	2
1. Find largest file	9.4	15.7	6.7	18.0	11.8	20.4	6.7	15.0
2. Find 2nd largest file	6.5	16.7	17.0	20.3	12.4	18.6	10.6	11.1
3. Find largest dir	20.7	21.3	12.7	10.0	11.8	28.2	13.3	17.6
4. Find file via path	25.0	17.1	18.7	18.3	31.2	14.7	25.1	19.7
5. Find file via path	22.2	27.1	15.5	24.6	39.1	19.2	26.7	23.7
6. Find file via path	28.0	29.6	18.9	39.4	21.2	21.4	21.0	22.7
7. Find file via path	37.7	31.6	36.0	35.3	33.9	24.0	25.6	29.5
8. Find file via name	36.5	44.5	16.5	31.4	27.0	50.7	21.0	46.5
9. Find file via name	32.0	28.0	26.9	20.0	–	43.8	41.2	23.5
10. Find deepest dir	37.0	28.2	25.7	22.0	35.8	16.6	19.8	15.4
11. Find dir contents	17.5	25.0	33.2	46.0	–	–	–	–
12. Find via size and type	22.3	21.7	13.2	18.7	25.5	27.8	13.7	15.4
13. Compare files by size	59.0	46.8	54.0	59.3	40.0	34.0	56.5	35.4
14. Find duplicate dirs	–	–	–	–	–	–	–	–
15. Compare dirs by size	28.0	26.4	23.9	27.6	43.0	36.8	50.7	37.0
16. Compare dirs by contents	60.0	46.3	44.6	33.3	35.7	36.2	47.0	37.2

Table 6: Completion times in seconds (for correct responses only) as a function of tool, hierarchy, and phase.

effect noted above for tasks 8 and 9. Note that in this experiment with the larger hierarchies, no participant was able to find the duplicated directory structures (task 14).

Average time for the successful task completions are shown in Table 6. As in Experiment 1, participants replied more quickly using TM on tasks 1-3 involving size assessments. On tasks 4-9 involving file finding, the results were relatively mixed, not favoring SB as much as in Experiment 1. It appears that once the hierarchy grows in size, the circular area of the slices in SB become even smaller, making identification of individual files more challenging. Nevertheless, tasks 10, 15 and 16 did show speed benefits for the SB tool. All these tasks involved some form of directory identification, then assessment of contents. As noted earlier, the explicit depiction of directories in SB may be a contributor to those results.

Statement	Experiment 1 TM	Experiment 1 SB	Experiment 2 TM	Experiment 2 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

Table 7: 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."

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 5 (strongly disagree). The results of these surveys, broken out by the two experiments (different hierarchy sizes), are presented in Table 7.

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 (Statement 1, Statement 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. 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 versus 2.54-SB.

It is interesting to note that the subjects who worked on the smaller file hierarchy 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 versus 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 the file 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. Also, 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 compare 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 as well as a suggestion that this tool was easier to learn than the TM tool. There are several possible reasons for this. For instance, 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. It would have been inappropriate to test each time difference for significance as a function of tool. It would also have been inappropriate to combine the times for the tasks in order to get an overall time, or perhaps a small group of times for subsets of tasks, since even related tasks were not designed to be approached in exactly the same way. Nevertheless, the times in Tables 3 and 6 suggest that TM was faster for finding large files and directories, and SB was faster for identifying named files and directories.

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 through 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 to 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.

Tasks 15 and 16 had participants perform tasks involving directory identification. Task 15 was similar to task 13 in that it asked participants to compare two directories based on size while task 16 was a variation on this, instructing participants to compare two directories based on the number of files contained within them.

The strategies used in task 15 were identical to task 13. Using TM most participants would locate one directory (through a combination of magnifying/focusing on directories and using the "deeper" button) and ascertain its size. The participants would then press the "top" button and repeat the process for the second directory. Upon seeing the second directory, they would compare it to their memory of the first directory's size and venture a guess. Using SB, most participants pressed "max," followed the paths, located each directory, and told the observer the name of the bigger one.

A slight variation of this strategy was used in task 16. With TM, most participants would locate one directory (through a combination of magnifying/focusing on directories and using the "deeper" button) and rather than judging its size, they would count the number of files in the directory. The participants would then press the "top" button and repeat the process for the second directory, and finally they simply identified whichever had more files. In most cases, it was not necessary to count all the files; most participants could tell just by looking at the directories which had more files. With SB, most participants pressed "max," followed the paths, located each directory, counted the number of files in each and told the observer the name of the directory with more. In some instances, the participant would double-click (magnify) the specific directory, estimate a count, return to the top and repeat the process for the second directory.

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.

It might be fruitful to construct a GOMS model[CMN83] simply 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.

7.2 System Enhancements

In addition to comparing use of the two systems we examined, 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:

- 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[CMS98] 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 two empirical studies 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 directory, 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 a larger file hierarchy. 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. We can carefully analyze 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 tool such as the Windows Explorer and UNIX shells.

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

References

[AH98] Keith Andrews and Helmut 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, October 1998.

[Chu98] Mei C. Chuah. Dynamic aggregation with circular visual designs. In *Proceedings of the 1998 IEEE Information Visualization Symposium*, pages 35–43, Raleigh Durham, NC, October 1998.

[CMN83] Stuart K. Card, Thomas P. Moran, and Alan Newell. *The Psychology of Human-Computer Interaction*. Lawrence Earlbaum Associates, 1983.

[CMS98] Stuart K. Card, Jock Mackinlay, and Ben Shneiderman, editors. *Readings in Information Visualization – Using Vision to Think*. Morgan Kaufmann, 1998.

[JS91] Brian Johnson and Ben 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, October 1991.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: A 2-d space-filling approach. *ACM Transactions on Computer Graphics*, 11(1):92–99, January 1992.

[TJ92] David Turo and Brian Johnson. Improving the visualization of hierarchies with Treemaps: Design issues and experimentation. In *Proceedings of the IEEE Visualization '92*, pages 124–131, Boston, MA, October 1992.

[vWvdW99] Jarke van Wijk and Huub 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, November 1999.