User Experiments with Tree Visualization Systems

Alfred Kobsa

University of California, Irvine

ABSTRACT

This paper describes a comparative experiment with five well-known tree visualization systems, and Windows Explorer as a baseline system. Subjects performed tasks relating to the structure of a directory hierarchy, and to attributes of files and directories. Task completion times, correctness and user satisfaction were measured, and video recordings of subjects' interaction with the systems were made. Significant system and task type effects and an interaction between system and task type were found. Qualitative analyses of the video recordings were thereupon conducted to determine reasons for the observed differences, resulting in several findings and design recommendations as well as implications for future experiments with tree visualization systems.

CR Categories 1998: H.5.2 [User Interfaces]: Graphical User Interfaces (GUI) H.1.2 [User/Machine Systems]: Human factors I.3.6 [Methodology and Techniques]: Interaction Techniques

Keywords: information visualization, experimental comparison, task performance, accuracy, user satisfaction, user interaction, design recommendations

1 INTRODUCTION

Comparisons between information visualization systems can provide valuable information about the effectiveness and the ease of use of such systems. An analysis of the reasons behind observed differences in effectiveness can help identify both points of weakness and success factors of visualization systems.

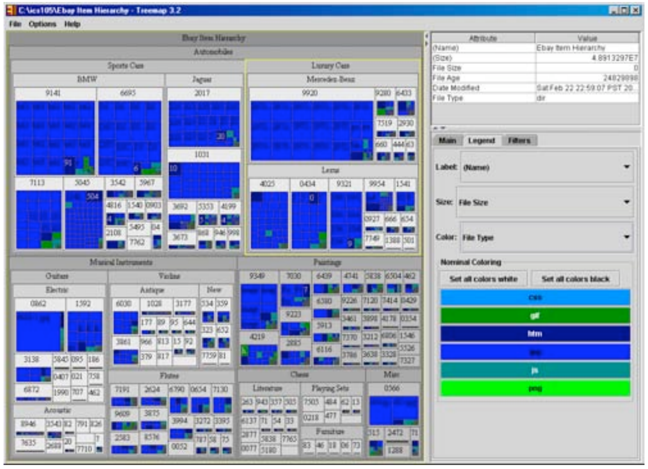
Relatively few such studies have been conducted so far (see Section 6 for an overview). In this paper, we describe a between-subjects experiment in which we compared five well-known visualization systems for (strict) trees, and Windows Explorer as a baseline system. While the five systems are in principle able to visualize any tree structure, their functionality suggests that they were mostly designed for the visualization of file directories and websites.

Section 2 provides a brief overview of these systems. Section 3 describes the observational and analytic methods of our study. Section 4 presents the overall quantitative results with respect to correctness, speed of task performance and user satisfaction. Section 5 discusses the quantitative and qualitative results and their implications individually for each system. Section 6 reviews related work, and Section 7 summarizes the conclusions from this study and proposes implications for future experiments with tree visualization systems.

2 VISUALIZATION SYSTEMS USED IN THIS STUDY

2.1 Treemap 3.2[1]

Treemap [6, 21] is a space-filling visualization that depicts subordination of tree nodes through recursive subdivision of rectangles that represent these nodes. The size of a rectangle is proportional to the size of the corresponding node, and users can choose among several partitioning algorithms. Labels and color inside the visualization convey user-selectable information about files. Filtering of files by size and age grays them out, and filtering by depth of nesting removes undesired substructures. Figure 1 shows our test tree when viewed with Treemap.



Attribute	Value
(Name)	Ebay Item Hierarchy
(Size)	4.8913297E7
File Size	0
File Age	24829898
Date Modified	Sat Feb 22 22:59:07 PST 20
File Type	dir

Label: (Name)

Size: File Size

Color: File Type

Figure 1. Test tree seen with Treemap (different colors denote different file types)

2.2 SequoiaView 1.3

SequoiaView (formerly Cushion Treemaps [25]) visualizes trees in a similar manner as Treemap. It goes beyond Treemap though by supporting a 2 1/2D appearance through shading and spotlighting. SequoiaView also provides additional functionality for filtering, sorting, highlighting and explicit inclusion or exclusion of files. Filtering does not grey out but omits the filtered files. Figure 2 shows our test hierarchy when viewed with SequoiaView in the default "squarified cushion treemap" visualization.

E-mail: kobsa@uci.edu, WWW: http://www.ics.uci.edu/~kobsa. The work described here has been partly supported by a grant of the UCI Center for Research on Information Technology and Organizations (CRITO). The author wishes to thank Frank van Ham, Catherine Plaisant, Ramana Rao, Ben Shneiderman and Jarke van Wijk for making their systems available and providing valuable input on the experimental tasks, and to F. Agagu, R. Krishna, S. Ratanotayanon, I. Seredkin, N. Su, S. Sun, T. Shubha, T. Zhengming and M. Yeh for their assistance in the experimental analysis. Thanks are also due to the reviewers of this paper for their helpful recommendations.

[1] A more recent release, Treemap 4.0, was not yet available at the time of this experiment.

IEEE Symposium on Information Visualization 2004
October 10-12, Austin, Texas, USA
0-7803-8779-1/04/$20.00 ©2004 IEEE

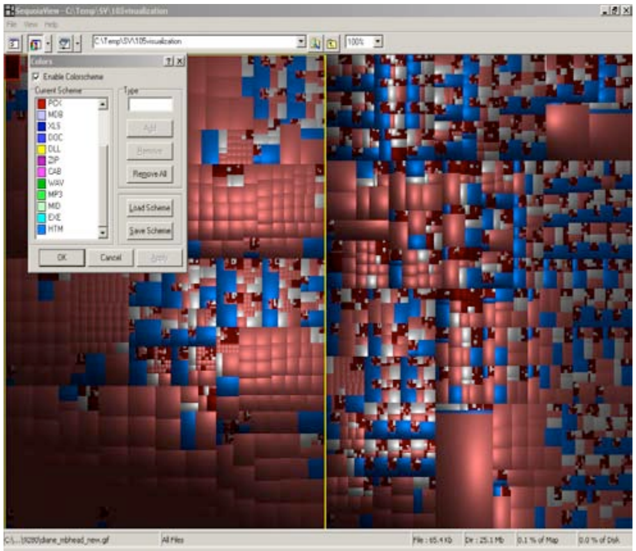


Figure 2. Test tree seen with SequoiaView (different colors denote different file types)

2.3 BeamTrees

BeamTrees [24] features both a space-filling Treemap-like visualization, and a 3D node-link visualization which is shown in Figure 3. Each individual beam represents a directory, and the slices of the beams represent files. Different file types have different colors. Users can rotate and magnify the display, brush files and folders to obtain information about them, change the proportions of the visualized objects, and change the color scheme.

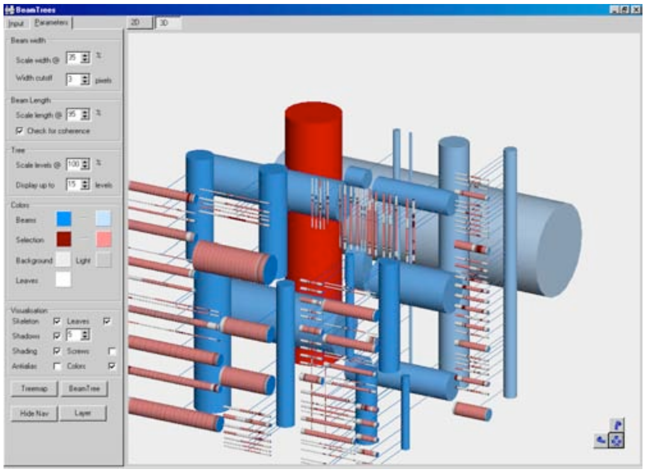


Figure 3. Test tree seen with BeamTrees (different colors denote different file types and the current selection)

2.4 Star Tree Studio 3.0

Star Tree (formerly Hyperbolic Browser [9-11]) presents trees in a node-and-link fashion and thereby follows a distortion-based visualization paradigm. Specifically, it uses fish eye distortion as the basis for a focus + context presentation and interaction. Users can shift the focus by dragging operations, and obtain information about folders and files by clicking on them. There is also a "Find and Replace" function where user can choose which field of the node properties to search. The results will be highlighted in the tree. Figure 5 shows our test tree with Star Tree.

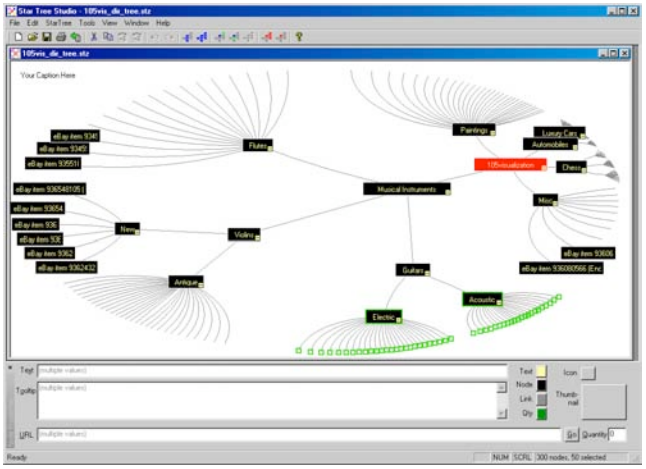


Figure 4. Test tree seen with Star Tree (different colors denote highlighting and selection)

2.5 Tree Viewer

The TreeViewer [7] visualizes trees in a form that closely resembles botanical trees (see Figure 5). The root directory is the tree stem and subdirectories are branches (multiple subdirectories of a directory branch off one by one). Terminal directories are "bulbs" at the end of branches, and files are discs-shaped "fruits" on top of the bulbs. Files and directories at the same level are displayed in the same color. The size of a directory is represented by the thickness of the corresponding branch or bulb, and the size of a file by the size of its fruit.

The user interface is divided into two parts: the left panel displays the 3-dimensional visualization, and the right panel contains an Explorer-like presentation of the same tree. Clicking on a directory in the left-hand panel will highlight the corresponding visualization (in Figure 5, the visualization of directory '9920' is highlighted in yellow), but not vice versa. Users can move and rotate the tree, and zoom in and out. They can also change the colors of the tree, leaves, branches and the background, and change the general appearance of the tree.

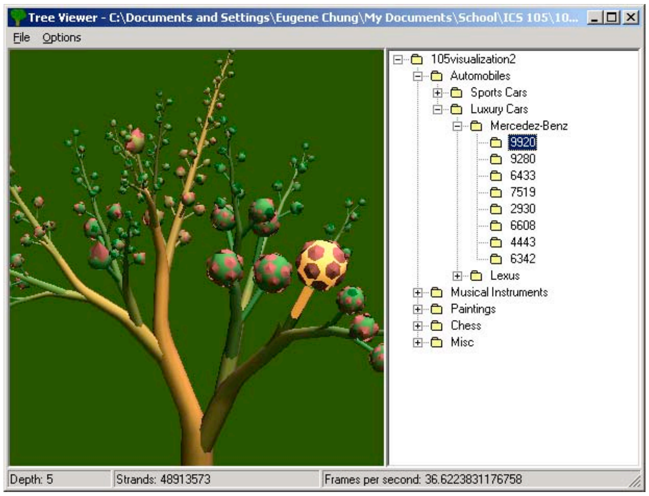


Figure 5. The test tree seen with Tree Viewer (same colors denote same hierarchy levels)

2.6 Windows Explorer

Windows Explorer is the integrated file browsing and management tool for the Windows operating system. It employs an expandable tree structure to represent the directory hierarchy, folder icons to represent directories, and different icons to represent files. For file management, Windows Explorer provides functions such as sorting, searching, and file detail viewing.

The window is divided into two panes. The left pane displays a view of the folder structure. The right pane shows details of the files and folders within a particular selected folder. Above these two panes is a toolbar with shortcuts for controlling Explorer features. The area along the bottom of the window gives information about objects currently selected within the folder and file detail panes. A menu bar at the top of the window provides access to the full set of commands and options of Windows Explorer.

3 Experiment

3.1 Goals

The aim of the experiment was to determine whether solving tasks in the six systems differs with respect to task completion times, accuracy and user satisfaction, and why this may be the case. The null hypothesis was that there are no such differences. The purpose of the experiment was not to pit the systems against each other, but rather to use observed quantitative differences as foci of interest in a subsequent qualitative analysis. A video analysis that was conducted as part of the experiment therefore aimed at identifying problems that subjects seemed to encounter and, if possible, at relating them to the quantitative results. Such findings can then provide an empirical basis for design recommendations.

3.2 Test Hierarchy

The test hierarchy was a subset of a taxonomy of items on eBay. It contained 5 levels and a total of 5799 nodes. Figures 1-5 show the hierarchy in the five different graphical visualization systems. The second-level subtrees were automobiles, musical instruments, paintings, chess, and miscellaneous (in order of the total number of nodes). The specific subset was chosen since the node names and their hierarchical relationships seemed easily understandable without special domain knowledge.

3.3 Tasks Employed

Users had to solve 15 tasks in the test hierarchy. Tasks were generated and selected by the experimenters in an iterative brainstorming process based on whether or not they were interesting and would naturally occur in the analysis of the respective data sets by a hypothetical eBay vendor, and would not require specific domain knowledge. The task selection was also informed by a very early version of the InfoVis 2003 contest tasks. In some cases, questions had to be rephrased using a more technical terminology in order to make them unambiguous. The following 15 tasks were eventually selected:

Q1. What is the maximum depth of the eBay hierarchy?
Q2. Is the tree balanced or unbalanced? (A tree is unbalanced if its depth in one branch is at least two more than in another branch).
Q3. Find the name of the parent directory of the directory "BMW".
Q4. Which directory has the greatest number of immediate subdirectories?
Q5. Which file was modified most recently?
Q6. Find the total number of bottom-level directories under "Automobiles".
Q7. Which bottom-level directory of the eBay item hierarchy has the largest number of files?
Q8. Find a file of type .css and write down its name.
Q9. Find the directory that contains the most .png type files.
Q10. Find all files added on Feb. 19, 2003. Write down their file names.
Q11. Locate the file labeled 1990.htm.
Q12. What is the name of the largest file in the eBay items hierarchy?
Q13. Find the number of directories WITHOUT a file of type .js.
Q14. Which directory includes a deeper hierarchy: "Flutes" or "Guitars"?
Q15. Find the two duplicate occurrences of the directory "1869". Indicate their pathnames.

Questions 1-4, 6, 7, 9, 14 and 15 refer to the structure of the tree hierarchy, while Questions 5, 8 and 10-13 refer to attributes of subdirectories or files. In the following, we will refer to these two different task types as “structure-related tasks” and “attribute-related tasks”.

3.4 Subjects

48 subjects participated in the experiment. They were students with a major or minor in Information and Computer Science or Engineering who had at least one year of experience working with computers. Prior studies with the adoption of information visualization systems by administrative data analysts [4, 5] had indicated that this subject population was reasonably representative for the target user group, due to the analytical capabilities they have in common. In another preliminary experiment, 20 sample subjects from the same population all ranked in the 90th percentile in a half-hour online spatial abilities test by H. J. Eysenck (most of them even in the 95th percentile).[2] Spatial ability was consequently not controlled for any more in the remainder of the experiment.

The data of one subject had to be discarded since this person did not follow the instructions thoroughly enough.

3.5 Experimental Design

The experiment took place in a small laboratory on the campus of the University of California, Irvine. A between-subjects design was used, with the visualization system as the independent variable. The 48 subjects were randomly assigned to one of the six conditions. Groups of 2-4 students received half an hour of instruction on the visualization system they were assigned to, and familiarized themselves with the eBay hierarchy. Thereafter they solved controlled practice tasks for another twenty minutes. During this practical training they received additional instruction from 2-3 experimenters.

Subjects then began the experiment. They had to answer each of the 15 questions, write down the answer(s), and self-time the duration of task completion. Subjects were instructed to abort unsolved tasks after 5 minutes and to note down this fact. Their interaction was recorded by screen capture software. At the end of the experiment, they completed a final satisfaction questionnaire for about 10 minutes.

In the quantitative analysis, the correctness of users' task performance and their task completion times were measured based on their answers in the answer sheet. The user satisfaction data were taken from the final questionnaire. A pair-wise Chi square test was performed to measure the effect of the system used on task correctness, and an ANOVA (with Fisher's PLSD) to analyze the system effect on task completion times and user satisfaction. All significant differences found will be discussed below.

In the qualitative video analysis, two observers independently watched the recording of subjects’ screen interaction during the experiment (they also viewed the preceding controlled practice

[2] The result is not very surprising since “proficiency in spatial ability has long been associated with success in cognitively demanding educational tracks and occupations such as engineering, architecture, physics, chemistry, and medical surgery” [20].

tasks). The observers were asked to record anything noteworthy in the videos, specifically with regard to interaction problems that applied to more than one subject. They were familiar with the results of the global quantitative analysis and were also instructed to look for evidence that might explain the observed differences between the systems with respect to task completion times, correctness and user satisfaction. After completing their independent observations, the two observers would come together, compare their notes, get back to the videos if needed to resolve disputes, and come up with a common report including design recommendations where appropriate.

4 Quantitative Results

To ascertain readability and compactness when indicating the statistical significance of observed differences, we will use the symbols < and << to denote differences at the 5% and 1% levels (the "better" system will always be on the right-hand side). For the same reason, we omit the degrees of freedom.

4.1 Correctness

Figure 6 shows (from bottom to top) the number of correct answers for each system, the number of incorrect answers, the number of cases in which subject indicated having exceeded the 5-minute timeout, and the number of cases when subjects gave no answer.

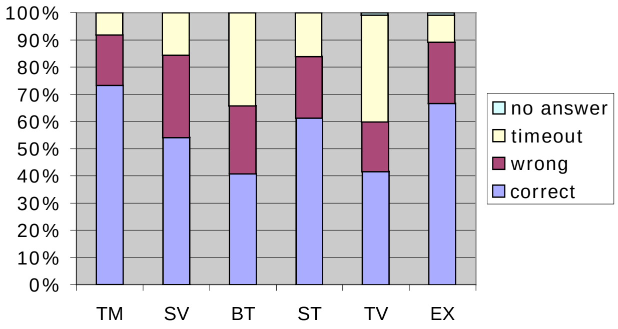


Figure 6. Correctness of answers[3]

TreeMap yielded the highest number of correct answers (73.3%), while the correctness of BeamTrees (40.8%) and Tree Viewer (41.7%) was the lowest. The timeout rate was highest for Tree Viewer (39.2%), and lowest for Treemap (8.1%) and Windows Explorer (10%). The following system differences in the counts of correct answers vs. all other outcomes were statistically significant:

BT << TM, SV, ST, EX
TV << TM, ST, EX
TV < SV

Figure 7 shows the data of Figure 6 broken down by structure-related tasks (s) and attribute-related tasks (s). The differences between answer correctness per task type are significant for BeamTrees (p=0.001) and Tree Viewer (p<0.0001), and also for all systems combined (p < 0.001).

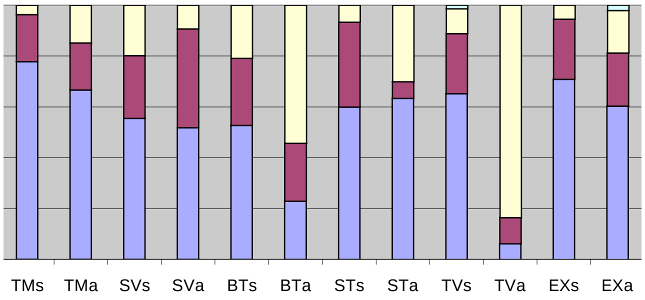


Figure 7. Correctness broken down by structure (s) and attribute-related tasks (a). See Figure 6 for the legend.

Finally, a comparison of the answer correctness within each task type reveals the following significant differences between systems:

Structure-related tasks	**Attribute-related tasks**
BT << TM	BT << TM, SV, ST, EX
< EX	
SV << TM	TV << TM, SV, ST, EX
< EX	< BT
ST < TM	

4.2 Speed of Task Performance

Figure 8 shows the average total performance times for the 15 tasks, per system. Whenever subjects indicated having exceeded the 5-minute limit, the task performance time was set to 300 seconds. The very few cases in which subjects gave no answer were disregarded. Windows Explorer and Treemap had the shortest task performance times (101.2 and 106.5 seconds per task), while BeamTrees had the highest (188.4 seconds per task).

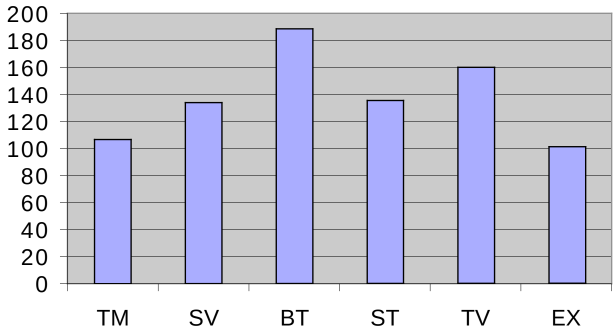


Figure 8. Average task completion times (in seconds)

An ANOVA revealed a significant effect of the system used (p < 0.0001). Fisher's PLSD showed that the following differences were significant at the 5% level (>) and at the 1% level (>>):

BT >> TM, SV, ST, EX
BT > TV
TV >> TM, EX
TV > SV
SV > TM, EX
ST > TM, EX

The effect of task type and the interaction between system used and task type is also highly significant (p<0.0001). We therefore break down the average task completion times by task type (see Figure 9).

[3] There were only two cases of "no answer", one for Tree Viewer and one for Windows Explorer.

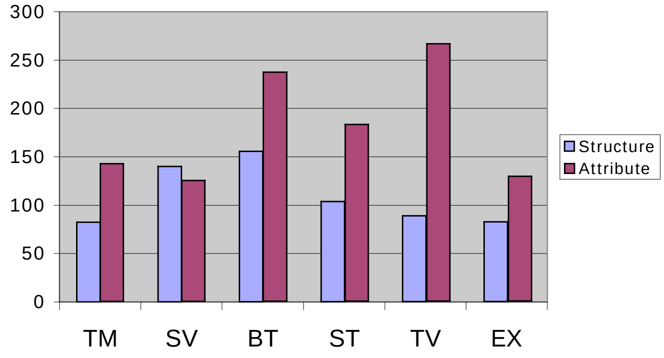


Figure 9. Average task completion times by task type and by system used (in seconds)

The differences of the task completion times between task types are significant at the 1% level for all systems except SequoiaView. An ANOVA with Fischer's PLSD reveals the following statistically significant differences between the task completion times for the six systems, within each task type.

Structure-related tasks	**Attribute-related tasks**
BT >> TM, TV, EX	BT >> TM, SV, EX
BT > ST	BT > ST
SV >> TM, TV, EX	TV >> TM, SV, ST, EX
SV > ST	ST >> SV
	ST > EX

As can be seen, Tree Viewer and Star Tree change their relative positions considerably depending on whether structure-related or attribute-related tasks are performed.

4.3 User Satisfaction

Figure 10 shows subjects' responses to questions about the ease of use and effectiveness of the system they had worked with, and whether they would use the system again for analyzing data. Scales ranged from –3 (worst rating) to +3 (best rating).

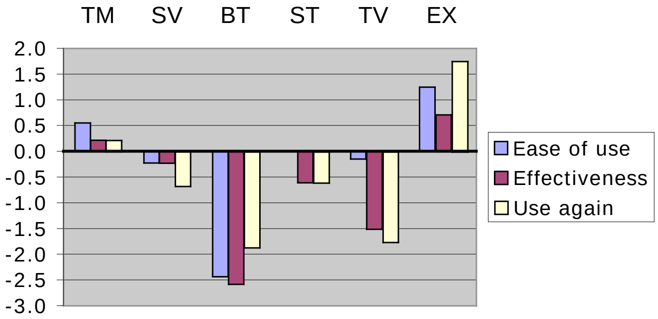


Figure 10. Perceived easy of use and effectiveness, and replies to "use system again?"[4]

The following differences in the responses to these usability questions are significant at the 5% or 1% level:

Ease of use	**Effectiveness**
BT << TM, SV, ST, TV, EX	BT << TM, SV, EX
	BT < ST
EX > SV, TV	TV << TM, EX

Use system again?
BT << EX
BT < TM
EX >> SV, TV
EX > ST
TM > TV

5 Individual System Discussion Augmented by Results from the Qualitative Video Analysis

5.1 Treemap

Treemap turned out to arguably be the best visualization system overall in this study. It does not surpass the baseline system Windows Explorer though. Treemap users excelled both in the correctness of their answers and in their task solution times, which came close to those of Windows Explorer. There were merely two questions (namely Q5 and Q10) that most subjects were unable to answer correctly. Treemap also received comparatively high user satisfaction ratings, which however fell far short of Windows Explorer.

The video analysis revealed that Treemap's most useful feature was its rich color coding and filtering functionality (filtering has meanwhile been considerably enhanced in a more recent system release). Subjects used it frequently, specifically for the attribute questions (however, many tried to directly find the result visually first before resorting to color coding or filtering). To make small objects stand out better, they would increase the contrast by coding them in a bright color and all others in a dull color (black in the extreme case).

In contrast, the videos confirmed concerns from the literature that global structure tasks (such as Q1, Q2, Q4 and Q7) are somewhat difficult to solve in Treemap. However, the quantitative analysis demonstrates that in comparison with other systems, Treemap is nevertheless still in the upper midfield with regard to these questions.

Based on the video analysis it seems that the usefulness of Treemap can be enhanced by integrating more (string) search functionality in its filtering pane, by a function that highlights small search results upon demand (e.g., with a bright halo), and by a detail-on-demand functionality that also includes statistics on the subtrees (such as path lengths and numbers of files and directories), to reinforce users' visual impression. A more detailed discussion can be found in [1].

The quantitative analysis showed no significant differences between Treemap and Windows Explorer. While it is true that the Explorer data may be somewhat skewed due to the fact that all subjects had been using Windows for a long time, it is doubtful though whether increased practice would enable Treemap users to clearly outperform Explorer users.

5.2 SequoiaView

SequoiaView proved to be an "average" system with respect to answer correctness, overall task solution times, and user satisfaction. Users had troubles answering the attribute-related questions Q5, Q10 and Q13 correctly. They also took comparatively long answering structure-related questions, which is surprising at first since SequoiaView was originally designed to improve users' perception of tree structures by replacing the 2D rectangles of Treemap with shaded 2 1/2D "cushions". However, the resulting spatial appearance did seemingly not help very much in our domain, which contained relatively few directories with many files of mostly similar sizes (see Figure 2). None of the subjects answered Q7 correctly.

[4] Star Tree's mean ease-of-use rating was 0.

The video analysis confirmed that subjects had indeed problems with structure-related questions, specifically regarding level and sibling detection. To figure out the tree structure, users tended to move the mouse around (to prompt the system to delineate the current file and embedding directory), or would use the depth-limiting function that the system provides.

Subjects frequently employed SequoiaView's powerful filtering mechanisms (even though its interface challenged them quite a bit); system functions that deliver meta-information about the current file or directory (such as its path, number of files and subdirectory, creation date, etc.); and options for changing the maximum visible depth of the tree. They used color coding far less than Treemap users, which may be explained by the fact that it is far less powerful than in Treemap.

Setting filter options, color options or a maximum depth in SequoiaView introduces a hidden state, and there is no indication of this state at the interface level nor a reset button to return to a "factory configuration". We saw that a majority of users occasionally forgot the current state. Sometimes they would recall it after running into problems and losing time, but sometimes they would also inadvertently make mistakes. More observed problems and design recommendations are discussed in [1].

5.3 BeamTrees

BeamTrees arguably achieved the worst quantitative results in our experiment, both with respect to correctness, task performance times and user satisfaction. Figure 7 shows however that as far as the correctness of structure-related tasks is concerned, BeamTrees compares quite well with other systems.

The video analysis confirms these quantitative results and provides more insights into the underlying reasons. The visual representation of *local* structural relationships certainly belongs to the strengths of the system. Users had few problems answering Q1-Q4 and also did this comparatively quickly. They had far more troubles though analyzing global structures in the 3D visualization, which seems to be due to the fact that beams at the same level did not appear to be at the same level in the visualization, and vice versa. This problem affected particularly questions like Q2. A second problem for global structure-related tasks may have been the fact that the lengths and sizes of beams bear little relationship with properties of the visualized directories. The reason for users' problems with attribute-related questions, finally, is mostly the lack of functionality beyond the display of path and size information.

Current strengths of the system that are usually frequently exploited by users include magnification, rotation and highlighting of beams, and "tooltips" that give limited information about beams and disks within beams. Users seemingly missed an undo function and/or a system reset option. [1] makes a number of additional design recommendations.

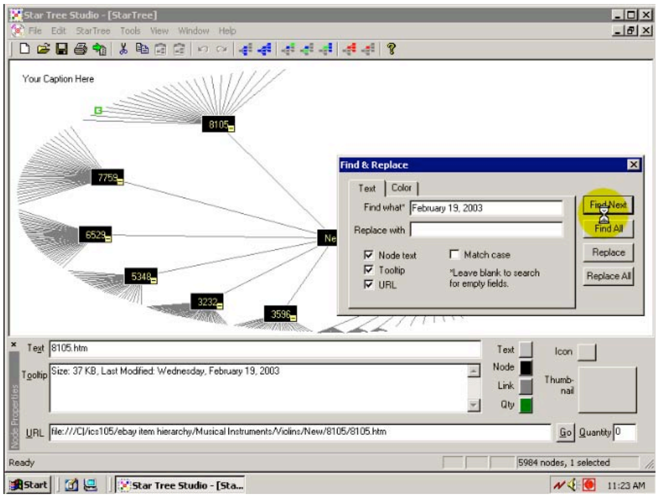
5.4 Star Tree

The Star Tree performance was "average", not only on a global level but also for nearly every individual question. The only major exceptions are Q5, Q6, Q7, Q9 and Q12, for which the correctness was very low. These tasks require subjects to rotate the tree structure in order to scan bottom-level files and directories, and for some tasks to additionally obtain file details that are not visually represented. The videos showed that subjects overlooked subtrees and files when performing these operations. Some users would occasionally try to better visualize the tree structure by using the "Bottom Orientation" instead (which puts the root node on top rather than the center). We noticed though that deep subtrees were nevertheless too much splayed over to the sides, requiring users to still rotate along the tree structure to browse it, and thereby making errors.

Another problem that we noticed is that nodes with the same distance from the center are not necessarily on the same tree level. This makes comparisons not entirely trivial and thus counting of levels is sometimes needed. Like in BeamTrees, users overall had few problems though solving *local* search problems once they identified the right neighborhood, and parent-child relationships were very easy to see.

In the video analysis, we also identified a number of minor usability problems and documented recommended revisions in [1]. While these problems are certainly not severe, we saw strong indications that they slowed users down and caused errors. Here we restrict ourselves to a few suggestions that can be motivated by Figure 11:[5]

- The find dialog box occluded the tree graph and the detail-on-demand displays, and many users constantly dragged it around. It might better be implemented as a small (pop-up) panel, in a toolbar or as a dockable item.
- Search results were often marked in a very inconspicuous manner (see the tiny rectangle in the left upper corner of Figure 11) which users occasionally overlooked. Better visible codes should be introduced, like coloring the entire path to a result, and/or marking the result by a halo.
- The search function should be made more powerful, give feedback on failed searches, and summarize the results of successful searches (in addition to marking them in the graph).



Find what* February 19, 2003

Replace with

☑ Node text ☐ Match case

☑ Tooltip *Leave blank to search for empty fields.

☑ URL

Text 8105.htm

Tooltip Size: 37 KB, Last Modified: Wednesday, February 19, 2003

URL file:///C|/ics105/ebay item hierarchy/Musical Instruments/Violins/New/8105/8105.htm

Quantity 0

Ready

5984 nodes, 1 selected

Figure 11. Some interface problems with Star Tree

In the final questionnaire, several users suggested a function that would count the number of nodes in a directory, seemingly because the fisheye perspective made a manual count difficult if the number of nodes became large. The addition of zoom and sorting functionality was also proposed. Star Tree was rated the aesthetically most pleasing system in the final questionnaire, but this result is not statistically significant.

5.5 Tree Viewer

Tree Viewer also trailed most other systems in performance, particularly for attribute-related tasks. One reason is that Tree Viewer lacks basic functionality (e.g., search operations) and data

[5] The picture shows a snapshot of a screen recording. The yellow circle around the hourglass-shaped cursor symbol was inserted by the recording software to mark the curser position.

displays (e.g., of file attributes) that are required or at least very helpful in some tasks. Subjects therefore found Q5 and Q7-Q13 impossible or extremely difficult to answer.

The video recordings revealed that Tree Viewer's Explorer-like listing of directories on the right-hand side was used very frequently (subjects also indicated this in the final questionnaire). On the one hand, this Explorer-like panel was employed in all tasks that cannot be answered in the tree visualization at all, or only in a very cumbersome manner (such as when the solution was visible in the tree, but the answer still required finding its name, which was difficult since there was no direct link from the visualization into the Explorer hierarchy but only in the opposite direction). On the other hand, the directory panel was also frequently used for questions like Q14, which one would expect to be solved in the visualization proper. The reason for relying on the directory nesting seems that continuing branches representing a directory cannot be easily followed at the branching points. To figure out what part of a branch constitutes a directory, subjects would click on the directory in the Explorer panel, and then frequently remain in this panel to complete the task.

An unexpected result of the video analysis was that users of Tree Viewer had serious troubles with Q2: those who tried the visualization all got the answer wrong while those who used the Explorer-like panel all got it right. The quantitative data had originally not identified this question as a problem spot. Q2 is indeed very hard to answer in the visualization. In general, the tree does appear to be fairly balanced. The different sizes of branches, the turns that branches take, the fact that same-level branches split off at different heights, and finally the occlusion of branches all make it difficult to tell which two branches are on equal levels. While these characteristics make the visualization more natural in appearance, the end result is that users have a hard time assessing the global structural characteristics of a tree.

Design recommendations based on the video analysis include detail-on-demand for tree constituents, two-way tight coupling between the tree visualization and the Explorer panel, more color-coding options, search facilities, and the possibility for left/right shifts of the visualization. For more details and additional recommendations see [1].

The rating for aesthetical appeal of Tree Viewer was slightly above average (0.25 on a scale from –3 to 3). However, the standard deviation of these ratings was by far the highest for all systems (1.98). User comments ranged from "I like the tree design" to "I hate this tree".

5.6 Windows Explorer

Windows Explorer showed a very good overall performance, both with regard to correctness, speed of task completion, and user satisfaction. Any comparison with other systems must be viewed with a grain of salt though since the subjects can be assumed to be highly skilled at least in its basic functionality. On the other hand, one should not overestimate the potential practice effects for the other systems since all tested visualization systems are relatively simple and since the training was quite thorough.

Explorer users had considerable difficulties with Q13 since it required expanding and scanning all directories until a solution is found. None of our subjects was able to solve the problem. For similar reasons, they also had troubles with Q7. They also did not well for Q2, which required them to compare the depth of different subtrees.

6 Related Work

[8] compared three commercial information visualization systems for multidimensional data with respect to task completion times and error rate. [12-14] extended this comparison to two different forms of synchronous collaborative data visualization. [26] analyzes the extent to which three 3-D browsers for hierarchies support the seven central tasks of information visualization systems postulated by [22].

[23] compared two space-filling visualizations for directory hierarchies with regard to task correctness, task completion times and user satisfaction: a Treemap-style visualization, and Sunburst which depicts hierarchies in concentric rings. The innermost ring thereby represents the root directory, and is divided into sectors whose angles correspond to the sizes of its files and subdirectories. The angles of subdirectory sectors are then projected onto the next outer ring and become again subdivided in the described way. While participants in the study preferred the circular visualization, the quantitative results of the study were inconclusive.

[2] compared four different visualization paradigms (among them a Treemap-like approach) with respect to their ability to communicate the topology of the tree and support comparisons of node size. The Treemap-style visualization turned out to be the slowest for most tasks.

It is difficult though to compare these two results about Treemap-like visualizations with our findings for Treemap 3.2 since the visualizations used are too different. The tested prototypes in these studies were not full-fledged visualization systems as in our case, but merely demonstrated the "bare bones" of the different visualization paradigms. For instance, Treemap 3.2 uses superpositioned hierarchical bars with directory labels that make the hierarchical structure more noticeable. This is completely lacking in the simplified reconstruction of [23] and [2].

[19] compared an older version of Star Tree, Windows Explorer, and SpaceTree which is a graph visualizer that allows subtrees to be dynamically expanded and contracted. The authors found statistically significant differences in the error rates and task completion times of different task types and were often able to explain them by properties of the visualizations that facilitated or impaired these tasks. No general pattern across task types could be found though.

[3, 11, 16] evaluated various versions of Star View and standard 2-D browsers similar to Windows Explorer, but were unable to find performance differences. [16-18] found performance gains of Star View though when the information scent (i.e., clues to the direction and proximity of sought information) was high, and performance losses otherwise. Finally, *The Great CHI '97 Browse-Off* contest [15] yielded Star View as the winner, ahead of Windows Explorer. [18] shows however that individual differences between subjects have a considerable higher effect on performance than differences between the two browsers, which may explain this outcome.

7 Summary and Conclusion

This study compared several information visualization systems for tree hierarchies in a between-subjects experiment. It unveiled statistically significant differences with respect to accuracy of task performance, task completion times and user satisfaction. It explained differences by referring to characteristics of the visualization paradigm, interface problems and missing functionality that make certain types of tasks difficult and/or slow down users. The study also showed a significant interaction between system used, and structure-related versus attribute-related task types.

While at least one system achieved the same performance as Windows Explorer, none of them showed benefits for users that went significantly beyond this baseline. The systems are also stand-alone rather than tightly integrated (e.g. into a file or website management system). [4, 5] however found that the integration into users' current work environment is an extremely important factor for the adoption of a visualization system by data analysts.

Two more conclusions can be drawn from the results, which may be relevant for future experiments. First, this study showed the merits of distinguishing structure and attribute-related tasks, for which some systems behave differently. So far, only local/global retrieval/comparison tasks have been distinguished in pure concept hierarchies [16-18]. It is unclear though whether the detected performance differences of all systems in these two task types can be attributed to a higher innate difficulty of attribute-related tasks, or due to the fact that the design of current visualization systems lends itself more to the appropriate visualization of structure than the appropriate visualization of attributes (which is somewhat at odds with the fact that Explorer subjects exhibited the same differences).

A second observation that may have consequences for future experiments is that the extreme outliers in the Tree Viewer and BeamTrees data have mostly been caused by a lack of functionality beyond the pure visualization. Once this problem is corrected, the effect size of differences in the visualization paradigms is likely to become smaller. This fact will have to be taken into account when planning the design, and specifically the sample size, of future experiments with tree visualization systems.

REFERENCES

[1] F. Agagu, R. Krishna, S. Ratanotayanon, I. Seredkin, N. Su, S. Sun, S. Tandon, Z. Tang, and M. Yeh, "Video Analysis of Tree Visualisation Systems," School of Information and Computer Science, University of California, Irvine, CA, Memo 2004.

[2] T. Barlow and P. Neville, "A Comparison of 2-D Visualizations of Hierarchies", IEEE InfoVis, San Diego, CA, 2001, pp. 131-138.

[3] M. P. Czerwinski and K. Larson, "The New Web Browsers: They're Cool But are They Useful?" People and Computers XII: Proceedings of the HCI '97 Conference, 1997

[4] V. González and A. Kobsa, "Benefits of Information Visualization Systems for Administrative Data Analysts", Sixth International Conference on Information Visualisation (IV03), London, England, 2003, pp. 331-336. http://www.ics.uci.edu/~kobsa/papers/2003-IV-kobsa.pdf

[5] V. González and A. Kobsa, "A Workplace Study of the Adoption of Information Visualization Systems", I-KNOW'03: 3rd International Conference on Knowledge Management, Graz, Austria, 2003. http://www.ics.uci.edu/~kobsa/papers/2003-KIV-kobsa.pdf

[6] B. Johnson and B. Shneiderman, "Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information", IEEE Visualization'91, San Diego, CA, 1991, pp. 284-291.

[7] E. Kleiberg, H. van de Wetering, and J. J. van Wijk, "Botanical Visualization of Huge Hierarchies", InfoVis 2001: IEEE Symposium on Information Visualization, San Diego, CA, 2001, pp. 87-94.

[8] A. Kobsa, "An Empirical Comparison of Three Commercial Information Visualization Systems", IEEE Symposium on Information Visualization, San Diego, CA, 2001, pp. 123-130. http://www.ics.uci.edu/~kobsa/papers/2001-INFOVIS-kobsa.pdf

[9] J. Lamping and R. Rao, "The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies," *Journal of Visual Languages and Computing*, vol. 7, pp. 33-55, 1996.

[10] J. Lamping, R. Rao, and P. Pirolli, "A Focus+Context Technique Based in Hyperbolic Geometry for Visualizing Large Hierarchies", CHI'95 ACM Conference on Human Factors in Computing Systems, 1991, pp. 189-194.

[11] J. Lamping, R. Rao, and P. Pirolli, "A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies", CHI'95, ACM Conference on Human Factors in Computing Systems, Denver, CO, 1995, pp. 401-408.

[12] G. Mark, K. Carpenter, and A. Kobsa, "A Model of Synchronous Collaborative Information Visualization", Sixth International Conference on Information Visualisation (IV03), London, England, 2003, pp. 373-381. http://www.ics.uci.edu/~kobsa/papers/2003-IV-mark-kobsa.pdf

[13] G. Mark and A. Kobsa, "The Effects of Collaboration and System Transparency on CIVE Usage: An Empirical Study and Model," *Presence*, vol. 14, 2005, http://www.ics.uci.edu/~kobsa/papers/2005-presence-kobsa.pdf

[14] G. Mark, A. Kobsa, and V. González, "Do Four Eyes See Better than Two? Collaborative versus Individual Discovery in Data Visualization Systems", Proceedings of the Sixth International Conference on Information Visualisation, London, England, 2002, 249-255. http://www.ics.uci.edu/~kobsa/papers/2002-IV-kobsa.pdf

[15] K. Mullet, C. Fry, and D. Schiano, "On Your Marks, Get Set, Browse! (The Great CHI'97 Browse Off)", Human Factors in Computing Systems, CHI '97 (Extended Abstracts), Atlanta, GA, 1997. http://www.acm.org/sigchi/chi97/proceedings/panel/kem.htm

[16] P. Pirolli, S. K. Card, and M. M. Van Der Wege, "The Effect of Information Scent on Searching Information Visualizations of Large Tree Structures", Advanced Visual Interfaces, AVI 2000, Palermo, Italy, 2000, pp. 261-272.

[17] P. Pirolli, S. K. Card, and M. M. Van Der Wege, "Visual Information Foraging in a Focus + Context Visualization", CHI-2001, Seattle, WA, 2001

[18] P. Pirolli, S. K. Card, and M. M. Van Der Wege, "The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 10, pp. 20-53, 2003.

[19] C. Plaisant, J. Grosjean, and B. B. Bederson, "SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation", IEEE Symposium on Information Visualization (InfoVis'02), Boston, MA, 2002, pp. 57-64.

[20] D. L. Shea, D. Lubinski, and C. P. Benbow, "Importance of Assessing Spatial Ability in Intellectually Talented Young Adolescents: A 20-Year Longitudinal Study," *Journal of Educational Psychology*, vol. 93, pp. 604-614, 2001.

[21] B. Shneiderman, "Tree Visualization with Tree-Maps: 2-D Space-Filling Approach," *ACM Transactions on Graphics*, vol. 11, pp. 92-99, 1992.

[22] B. Shneiderman, "The Eyes Have It: a Task by Data Type Taxonomy for Information Visualizations", IEEE Symposium on Visual Languages, Boulder, CO, 1996, pp. 336-333.

[23] John T. Stasko, R. Catrambone, M. Guzdial, and K. McDonald, "An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures," *International Journal of Human-Computer Studies*, vol. 53, pp. 663-694, 2000.

[24] F. van Ham and J. J. van Wijk, "Beamtrees: Compact Visualization of Large Hierarchies", InfoVis 2002, Boston, MA, 2002, pp. 93-100.

[25] J. J. van Wijk and H. van de Wetering, "Cushion Treemaps: Visualization of Hierarchical Information", IEEE Symposium on Information Visualization (INFOVIS'99), San Francisco, CA, 1999

[26] U. Wiss, D. A. Carr, and H. Jonsson, "Evaluating Three-Dimensional Information Visualization Designs: a Case Study of Three Designs", IEEE Conference on Information Visualization, London, England, 1998, pp. 137 -144.