A Comparison of 2D Nested Visualizations of Hierarchies

Petra Neumann
Department of Computer Science
University of Calgary
Calgary, Canada T2N 1N4
pneumann@cpsc.ucalgary.ca

ABSTRACT
Trees are a very common way of structuring data hierarchically and are applied in a wide variety of application domains. Due to a lack of evaluation techniques for tree visualizations there is still no general agreement in the information visualization community about which tree visualizations should be used for which tasks in which domain. Therefore, researchers developing new visualizations for tree structures have to find their own ways of assessing their tree layouts.

This paper describes a comparative study between two containment tree layouts. The ArcTree visualization, a one-dimensional variation of the Treemap layout, is compared to the traditional slice-and-dice Treemap. The study aims at finding differences in performance of both layouts in depicting global and local structural information of tree data. Measures of answer time and error rate performance suggest that participants solved tasks faster using the ArcTree layout visualization while maintaining about the same accuracy.

Author Keywords
Evaluation, information visualization, hierarchies, ArcTree, Treemap

INTRODUCTION
Evaluation in the field of Information Visualization is a growing research area. Only recently have studies to a greater extent begun to assess the potentials and limitations of visualization tools. An awareness of the need for evaluation techniques in the field has arisen since an increasing number of information visualization tools are reaching customers. The TREEMAP visualization [3] is one such system that has made it into commercial products. Considering this fact it is surprising that the TREEMAP algorithm performed badly in controlled experiments when compared to other visualizations for hierarchical data [1, 7].

The original Treemap visualization presents tree structures in a two-dimensional space-filling approach with a nested representation similar to Venn diagrams (c.f. Figure 1(a)). Each node in the Treemap is drawn as a rectangle with a given size. For drawing a Treemap, the available screenspace is partitioned into rectangular sections. The first rectangle forms the root of the tree. The tree layout algorithm partitions the root rectangle into n parts with n being the number of the root's children. The algorithm is recursive; however, nodes are partitioned vertically at even levels and horizontally at odd levels.

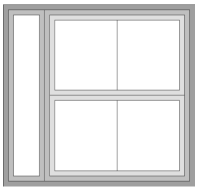
Typically, space-filling tree layouts like the Treemap perform worse in depicting a tree's structure when compared to traditional node-link tree layouts which use an additional two-dimensional axis to layout tree levels. Space-filling layouts shine when users are interested in reading leaf nodes since, in general, more display space can be reserved for their display. In this study we want to investigate how a one-dimensional Treemap, called the ArcTree (c.f. Figure 1(b)), performs compared to the original two-dimensional slice-and-dice Treemap layout. Using a controlled experiment we examine how well participants are able to read the tree's topology in both visualizations.

The paper is structured as follows: After introducing the motivation for the study, we will describe the experiment in detail. Results will then be statistically assessed followed by a discussion of the results and concluding remarks about the study.

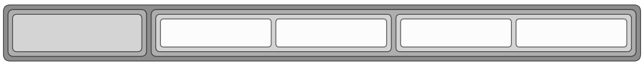
MOTIVATION
The motivation for this study lies in previous research on ARCTREES which were developed to depict relational information in tree structures [5]. A one-dimensional Treemap-layout, similar to the one described in [8], was developed for the visualization since it uses limited screen space (c.f. Figure 1(b)). The basic layout algorithm is similar to the Treemap. Nodes are nested inside their parent nodes. The algorithm is recursive but does not change partitioning orientation, partitioning is always performed with horizontal splits. This leads to a layout that can be read from left to right and top to bottom (through the nesting) similar to the reading direction for traditional node-link diagrams.

Unlike other non-space-filling approaches tree visualizations using containment to depict level information can be easily constrained in space. However, space-filling tree layouts typically perform worse in depicting tree topology since no additional horizontal or vertical axis on the screen is used to depict level information. However, a good understanding of the tree layout is often crucial for tree visualizations. Therefore, the developed tree layout[1] is empirically evaluated by comparing it to a regular slice-and-dice Treemap [3] to deter-

[1]We will call the layout the ARCTREE LAYOUT in the remainder of the paper.



(a) Treemap layout.



(b) ArcTree.

Figure 1. Tree layouts used in the study showing the same data structure.

mine if an improvement in the depiction of tree topology can be achieved. The Treemap is chosen for a comparison since it is visually similar to the ArcTree but has a different layout metaphor and design. The Treemap layout has also been part of several tree studies [1, 7] that assessed its potential in different situations. It typically performed worse compared to other approaches. An improvement in the depiction of tree topology compared to the the traditional Treemap layout will encourage further use and development on the ArcTree layout, as well as further studies comparing the technique to other tree layouts.

DESCRIPTION OF THE EXPERIMENT

The purpose of the presented study was to evaluate two different tree visualization techniques for visualizing hierarchical data. We used several different tasks for evaluating the depiction of local and global structural information of the tree. A pilot study was conducted with ten participants using a fully-crossed within-subjects 3x2 design. We hypothesized that the ArcTree layout would perform better according to answer time and accuracy in the study since it uses, in our opinion, a more intuitive tree layout. The full pilot study and its design will be described in the following.

Subjects

Study participants were 10 computer science researchers and colleagues (9 graduate students, 1 professor) all of which had previous experience in the human-computer interaction and information visualization field. There were 6 male and 4 female participants. From being exposed to hierarchical data in particular file systems on a regular basis all participants had previous experience working with hierarchical data and some of its typical visualizations. All participants had never or rarely used any of the two visualization techniques. The Treemap had been used previously by three participants while the ArcTree visualization had been used previously by only one participant.

Materials

For visualizing hierarchical data using the Treemap layout the Treemap 4.1.1 software published by the University of Maryland's Human-Computer Interactions Lab was used [6]. This software was written by or under the supervision of the original publishers of the Treemap algorithm. The software displays the needed slice-and-dice Treemaps using a configurable offset around each node. The ArcTree layouts were created using a software written by the authors. The layout was adapted to be similar in color and proportional in node size to the Treemap layouts created with the above mentioned program. To create specific tree data the yEd Java Graph Editor software by yWorks was used [9]. Trees were created based on node count, maximal depth, and maximal child count constraints. Trees were saved in the .graphml format. A conversion program was written to convert these `.graphml` trees into the `.xml` and `.tree` data structures used by the Treemap and ArcTree visualizations ,respectively. For each task different screenshots were taken from the above mentioned visualization programs. The screenshots for the ArcTree visualization all had the same size while the screenshots for the Treemap visualizations had to be adjusted in order to make all nodes visible. If the screen space for the Treemap visualization was too small to display all nodes the program simply clustered nodes. To avoid such a clustering the size of the visualization was increased until all nodes became visible and then the screenshot was taken. Offset size for the Treemap was set to correspond to the vertical offset size in the ArcTree visualization. The area of the screenshot for the Treemap was never smaller than for the ArcTree visualization but often larger. The ArcTree screenshots were taken at a resolution of 132,871 pixels. Treemap screenshots varied between an area of 133,856 and 1,060,656 pixels. All screenshots were shown to the participants on a 21" UltraSharp Dell Monitor at a resolution of 1600x1200 (1,920,000) pixels using analog input.

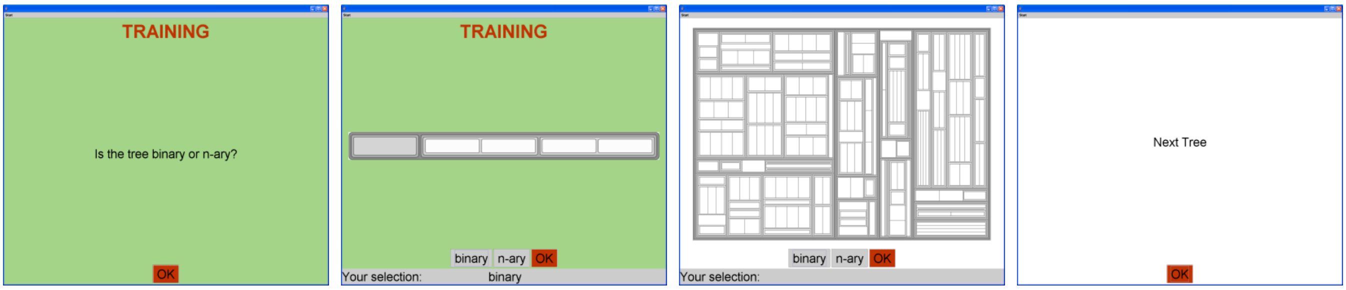
Methods

The experimental design was a 2 (tree layout) x 3 (tree size) repeated-measures design. Tree layout and size were the independent variables, while error rate and time were measured as dependent variables in the study. Tree size was chosen as an additional independent variable since it was felt that for smaller trees there might be less of a difference between tree visualizations than for large trees. Small (20 nodes), medium (160 nodes), and large (300 nodes) trees were tested in the study.

Each study session (about 40 minutes long) involved a single participant working through four stages.

Stage 1: *Pre-test questionnaire*. Participants completed a questionnaire asking about their computer experience, familiarity with tree structures, and the two tree visualization techniques.

Stage 2: *Training*. Participants were introduced to both visualization techniques. Each technique was explained using paper examples and also compared to traditional node-link tree diagrams. Each participant had to complete at least two paper based tasks which involved labeling nodes in Treemap and ArcTree displays based on a tree that was presented with labels as a node-link diagram. Participants could choose to perform more tasks if they felt they needed more practice.



(a) Question Screen. (b) Sample Training Screen. (c) Sample Study Screen. (d) Intermediate Study Screen.

Figure 2. Sample screens taken from program used in the study.

Stage 3: *Study*. For the study participants had to go through seven tasks using a program that displayed the screenshots and recorded answers, time, and error for each tree of each tasks. It first displayed the task on a welcome screen indicating that a training session was to follow next (c.f. Figure 2(a)). At this screen the examiner explained the tasks and how to solve them using paper based examples with node-link tree diagrams. The participants would then go through a training session of four trees (two for each layout) using the study program during which answers were discussed with the participants (c.f. Figure 2(b)). After each tree an intermediate screen was presented to give participants time to recover. By pressing an “OK” button participants could then proceed to the next tree. After the training session, participants were told now to be as correct in their answers as possible but if they felt they could not answer correctly after a certain time they were told to make a good guess. For each task the study contained 6 trees (3 for each layout, 2 for each size). Each tree was presented on a screen similar to the one in Figure 2(c). Eight random sequences of tree layouts were created to which participants were assigned. The order of the size of trees was also randomized across tasks. For each task the same tree structure was presented for one size for both tree layouts in order not to bias the results. After each screen, participants were presented with an intermediate screen (c.f. Figure 2(d)). By clicking an “OK” button they could then proceed to the next tree until they reached the next task for which the described process would be repeated.

Stage 4: *Semi-structured interview*. Participants were questioned about their performance in the study, with which visualization they felt more confident about their results, which one they preferred, and where they would see possibilities to improve either one of the visualizations.

Tasks

The tasks used to evaluate the depiction of tree structure using tree visualizations were based on general assumptions about typical tasks that need to be performed when analyzing hierarchically structured data. Typical real world tasks involve finding certain nodes in a tree, relating them to their context (ancestors, siblings, and descendants), or finding groups of nodes by identifying patterns in tree structures. In this study we concentrated on the latter two tasks which involve the analysis of the tree’s topology rather than an analysis of labels or other textual data which would be necessary for searching inside a tree. The depiction of global and local structural information was tested in seven tasks in this study. The first four tasks concentrated on global structural information while the last three tasks concentrated on the depiction of structure in context of one or two specifically marked nodes.

1. *Binary or n-ary*. Participants had to decide if the tree was binary or n-ary. For binary trees all non-leaf nodes had one or two children. Most nodes had two children. Nodes in an n-ary tree could have more than two children. Finding n-ary trees in this task was considerably easier than finding binary trees since participants had to find only one outlier to identify a tree as n-ary while they had to search a whole tree for outliers in order to verify that it was binary.

2. *Balanced or unbalanced*. Participants had to decide if the tree was balanced or unbalanced. In a balanced tree path lengths from the root to any leaf would not differ by more than 1 whereas in unbalanced trees they could differ by 2 or more. In this task, again, it was easier to find unbalanced trees since only one exception had to be found whereas for the balanced trees the whole structure had to be checked to verify the right answer.

3. *Number of levels*. Participants had to find the deepest node(s) in the tree and count at which level the node resided in the tree.

4. *Largest branch*. Participants had to select (by clicking on the screenshot) the root of the largest branch in the tree. The largest branch was the one with the most nodes below it (excluding the root node). The position clicked on was indicated with a red cross.

5. *Number of siblings*. For this task a node was highlighted in the screenshot. Participants had to count the number of siblings of the node.

6. *Level*. Participants had to determine at which level a highlighted node resided in the tree.

7. *Deepest common ancestor*. Participants had to find the deepest common ancestor of two highlighted nodes.

Problems

Several problems were encountered when preparing data for this study:

- The Treemap program displayed node level information using a linear gray scale while leaf nodes were all colored in white. The ArcTree layout was adjusted to this layout by also using a linear gray scale. However, the leaf nodes were also colored according to their level in the tree since this coloring could not be adjusted in the program. By using a gray scale to depict level information the ArcTree layout was somewhat compromised since it typically used a smaller display space and was created to be used with color scales having a higher number of JND's (just noticeable differences) [4]. However, the gray scale was chosen to make both visualizations as similar as possible in order not to bias in favor of one visualization. The goal of the study was to show how the use of a different layout technique could improve the depiction of structural information. Using a different color scale for the ArcTree visualization could have biased the results.

- Although the ArcTree visualization could have displayed more than 300 nodes the Treemap program used often started to cluster nodes when trees became larger even at the largest possible window size. Therefore, the maximal possible node size was set to 300 nodes for this study.

- For some of the trees with 300 nodes that had a large depth the Treemap algorithm could not display the tree in a window that would fit on the monitor without clustering nodes. Therefore, several data sets had to be recreated using flatter trees that could be displayed inside the given screen side.

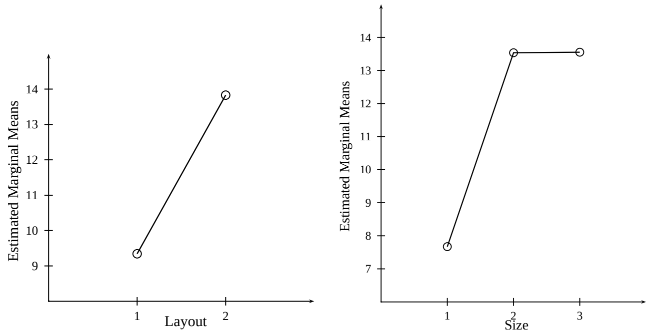
RESULTS

For each participant we calculated mean response times (RT) as well as response accuracy (RA) for each layout and each tree size. RT and RA were analyzed using a repeated measures analysis of variance with post-hoc (Tukey) comparisons. Layout(2) and size(3) were used as the factors in the analysis. Using $\alpha < 0.05$ an ANOVA of RT found a significant difference in performance between layout, size, and layout*size. Participants were significantly faster with the ArcTree as can be seen in Figure 3(a). The post-hoc analysis found that there was a significant difference in performance for both trees between size 1 (the smallest tree) vs. size 2 and 3 but no significant difference in performance between size 2 vs. 3. This finding is illustrated in Figure 3(b). A plot showing an analysis of the layout*size interaction is shown in Figure 4.

Although people did make more mistakes with the Treemap layout an ANOVA ($\alpha < 0.05$) of RA found no significant difference in response accuracy between both layouts.

DISCUSSION

The results obtained from the study showed that participants answered faster with the ArcTree visualization than with the Treemap at no significant difference in accuracy. The results will be discussed in more detail in the following.



(a) Layout: 1 = ArcTree, 2 = Treemap.

(b) Size: 1 = small, 3 = large.

Figure 3. Estimated marginal means (RT) for layout and size factors.

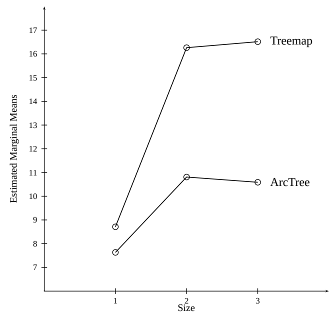


Figure 4. Estimated marginal means (RT) for layout*size.

Interpretation of results

From the ANOVA and post-hoc analysis of the RT times we learned that people answered faster using the ArcTree visualization. This was true for every tree size although the difference in RT was a lot bigger for the medium and large trees. For the small trees participants were in average only 2 seconds slower with the Treemap while for the medium and large trees the difference in mean answer time was about 6 seconds. Findings collected during stage 4, the semi-structured interview showed that participants were in general more confident about their results with the ArcTree visualization (8 out of 10 participants). This could be due to the fact that participants found it easier to find a strategy for answering questions using the ArcTree layout because it involved an easier node layout strategy than the Treemap. One participant who had used the Treemap before felt more comfortable with the Treemap because of previous experience but also stated to be more confident about the correctness of the answers with the ArcTree layout. Two participants stated to be overwhelmed by the size of the Treemap display. In particular, node level tasks for nodes in the middle of the Treemap display were found to be hard by several participants. Since the ArcTree layout displays nodes in a sequential order from left to right counting levels only involved following the containing nodes in vertical direction which helped orientation in the tree. There was no significant difference in response accuracy between both layouts. Even though people made 13 errors with the Treemap and only 5 with the ArcTree, the overall error rate was very low for both trees and, therefore, no significant difference was found. In general, response times were faster at equal accuracy even though the ArcTree display used less screen space for each of the trees in the study.

Relation to other works

BARLOW and NEVILLE performed a comparative study of four tree visualizations including an organization chart, icicle plot, Treemap, and tree ring [1]. Similar to our approach, the study looked at the ability of the trees to facilitate an interpretation of parent-child and sibling relations, the ability to support the comparison of node size, and preferences of the participants. The study compared trees at sizes 16 and 100 nodes. Some of the tasks from this study were the same as in our study. The authors analyzed the following tasks: binary vs. n-ary, balanced vs. unbalanced, deepest common ancestor, number of levels, and three largest leaves. The study found the Treemap to always be less effective than at least one other tree visualization and also rated lowest in user preference. Similarly to concerns raised by users of our study users performed badly for tasks involving counting of node levels. Due to its nested layout, the Treemap should not be used for comparing node sizes (node area is not proportional to node size due to the used offset) the Treemap performed worst for the last task. Due to the poor performance of the Treemap in the first experiment the authors unfortunately did not further test the Treemap in a second experiment involving a more detailed analysis of a decision tree. In general this study had similar aims as our study but compared trees a little differently. For example, in the binary vs. n-ary tasks, the authors claimed to use binary trees where each node had two or no children. This is mathematically impossible for the node sizes the authors claim to use[2] but even if they changed the node size of the trees for this tasks users would have to look for a regular pattern in the tree to determine if it was binary rather than looking for outliers like in our setup. The authors also did not make an analysis of RT and RA over all tasks but analyzed the responses for each tasks individually. While this procedure might answer some questions, more general statistics would also have been interesting since in a typical real world setting tasks would not be performed separately from each other but interchangeably or even at the same time. However, in our study a more thorough per-task analysis should have been performed to better asses the potentials and limitations of the tested visualizations for certain tasks. From the raw data collected during the study such an analysis would be possible, however, a higher number of trees per size and layout per task should be tested to make such an analysis more conclusive. For a follow-up study this will be considered.

Another study by STASKO et al. compared the Treemap to the Sunburst method, a radial space filling tree visualization [7]. Participants performed analysis—mostly node size comparison—and search tasks using the two tools. The study found that for most tasks the Treemap visualization performed worse for correctness and time. The Sunburst technique was also preferred by most users. The authors hypothesized this to be due to a more explicit portrayal of structure in the Sunburst visualization. The authors also noted that for the Treemap there was a significant improvement in performance over time which might be due to a greater learning cost for the visualization. The study design differs from our study in one general layout point: Treemaps were displayed without using an offset around nodes which makes it very hard for the Treemap to convey structural information, however making it possible to compare leaf node sizes (in contrast to the version used in the study presented above).

[2] For example, in a tree with 15 nodes each node can have two children. At 16 nodes at least one node has to have a single child or more than two children.

Impact for practitioners

The discussed findings are very encouraging for further development and evaluation of the ArcTree layout. It could be shown that the layout, even in its compromised state, performed better according to time while using less display space. Several improvements were suggested for both layouts by the participants during the semi-structured interview. Color coding of level information, rather than using a gray scale, was the improvement most often called for. Most participants found it confusing that in the Treemap layout all leaf nodes were colored in white. Several participants expressed the wish to color the leaf nodes according to level as well. Improvements for reading level information for both visualizations could involve using a bigger offset space between nodes or using additional tools to convey level information (like zoom lenses). One participant suggested introducing a tool, resembling a horizontal line that could be slid over the ArcTree visualization to make it easier to compare level information. Path highlighting, splitting of the visualizations, and node highlighting techniques were also suggested improvements. While some participants felt that the size of the nodes and offsets was fine, some felt that they were too small for reading the trees comfortably on the screen. This concern has to be taken seriously in future development of the visualizations since not all users of a visualization tool will have an equally good eye sight. Additional tools or layout strategies should be developed to create a more equal, or just larger, display for nodes. While such research has already been performed for the Treemap [2] some new techniques still have to be found for the ArcTree visualization.

Critical reflection

The presented tree study is only valid in context of the used tasks, trees, and visualization techniques. Although it was attempted to make the study as general as possible the tasks and trees used are limited. In real world tasks it might often, if not usually, be the case that trees are larger than 300 nodes. The study cannot predict how performance with the tree visualizations will change with increasing node number for the given tasks. Depending on the structure of the tree (flat vs. deep) a limit for the number of displayable nodes might be reached after a certain threshold for each visualization technique. After this point, nodes would have to be clustered in the displays and a general overview of the tree with all nodes visible at the same time would become impossible. If one wanted to test the two visualizations with larger trees the study would have to be made interactive. Users would have to be able to expand clustered nodes or collapse nodes to hide information that is not currently needed. Making a further study involving interaction techniques which include the above mentioned improvements of the tree layouts (highlighting techniques, search or counting aids, etc.) should also be included in follow-up studies. However, in that case it would be critical to include interaction techniques that either work for both visualization techniques or similar in order not to bias the results.

This study is also limited in the types of tree visualizations evaluated. The ArcTree visualization was tested against a fairly similar but known to typically perform badly visualization technique. An evaluation comparing it to a visualization technique that does not have the inherent limitations of containment tree layouts would have made the results of the study richer and probably more valuable.

The tasks used in this study were kept at a very abstract level. RT and RA might have been quite different for the presented trees if the tasks were actually connected to real world data and if tasks would have been rephrased to match such data. Instead of letting users simply count levels in an empty tree structure participants could, for example, have been asked to determine the lowest rank in a command hierarchy.

Research agenda

The initial hypothesis for this study was that there would be a difference in performance for the tasks used to determine the depiction of tree topology for the two visualization techniques. Now that it has been established in a pilot study that the ArcTree visualization is capable of depicting tree topology and users can answer most tasks correctly, it is important to prove these findings in a wider study using more participants in different age groups and of different background. Using different age groups is important in order to test how well participants with generally worse eye sight (under the assumption that older people typically require reading glasses etc.) can perform tasks with the currently small offset and node displays in the ArcTree visualization. For older age groups, than the one tested in the pilot study, the findings might be quite different.

In the initial research on ArcTrees [5] the tree layout was developed to allow for an additional display of relational information on top of the layout. In a full ArcTree with additional display of relational information, arcs connect two nodes that are related in some way. If users use this visualization technique to primarily analyze such relational information it might be more important for the tree visualization to depict local structural information well for the connected nodes, rather than global structural information (as tested in tasks 1 through 4). A future study should probably concentrate on tasks similar to tasks 5 through 7 and compare the ArcTree with several other visualization techniques besides the Treemap. In case a better tree visualization is found for the depiction of local structural information further research could involve finding ways of incorporating additional relational information on this tree rather than the original ArcTree layout.

CONCLUSIONS

This paper aims at assessing the depiction of tree topology with the ArcTree layout in a comparative study. The ArcTree layout, a one-dimensional variation of the Treemap layout was compared to the traditional slice-and-dice Treemap layout. Significant differences in answering time were found between the two layouts while answers were given with an approximately equal error rate. These findings suggest that even though the ArcTree display uses less display space, it performes better or equal in comparison to the Treemap layout. Certainly, this study is only a first step in a more careful evaluation of the ArcTree visualization. Future studies could include other tree visualizations, tasks, or tree data to better asses the layout's potentials and limitations.

ACKNOWLEDGMENTS

I would like to thank Sheelagh Carpendale and Saul Greenberg for helpful insight on constructing the study. Special thanks go to Stephanie Smale who always helped me during the study organization and the statistical analysis. I would also like to thank Mark Hancock for introducing me to SPSS and helping me to set up and perform the analysis for this study.

REFERENCES

1. Barlow, T. and Neville, P., A Comparison of 2-D Visualizations of Hierarchies, in *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*, IEEE Press, Los Alamitos, CA, USA, 2001, 131–138.

2. Bruls, M., Huizing, K., and van Wijk, J., Squarified Treemaps, in *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000)*, IEEE Press, Los Alamitos, CA, USA, 2000, 33–42.

3. Johnson, B. and Shneiderman, B., Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information Structures, in *Proceedings of IEEE Visualization '91*, IEEE Press, Los Alamitos, CA, USA, 1991, 284–291.

4. Levkowitz, H. and Herman, G. T., Color Scales for Image Data, *IEEE Computer Graphics and Applications 12* (1992), 1, 72–80.

5. Neumann, P., *Visualization of Relations in Hierarchical Data*, Diplom thesis, University of Magdeburg, Magdeburg, Germany, 2004.

6. Plaisant, C., Shneiderman, B., Chintalapani, G., and Aris, A., Treemap 4.1.1, Website, 2004, `http://www.cs.umd.edu/hcil/treemap/`.

7. Stasko, J., Catrombone, R., Guzdial, M., and McDonald, K., An evaluation of space-filling information visualizations for depicting hierarchical structures, *International Journal on Human-Computer Studies 53* (2000), 5, 663–694.

8. Turo, D. and Johnson, B., Improving the Visualization of Hierarchies with Treemaps: Design Issues and

Experimentation, in *Proceedings of the IEEE Conference in Visualization*, Boston, MA, USA, 1992, 124–131.

9. yEd, yEd - JavaTM Graph Editor, yWorks GmbH, Program available at: `http://www.yworks.com/en/products_yed_about.htm`, 2004, version used: 2.2.1.

APPENDIX: ARCHIVAL REPORT

This section contains additional material used or collected during the study.

Experimental materials

This section contains materials used during the study:

Pre-session questionnaire

The following questions were asked on the pre-session questionnaire. The full design of the questionnaire can be found in the attached ethics application.

1. Circle the number that best indicates how long you have used computers for
2. Circle the number that best indicates how often you work with or analyze tree structures (i.e. file systems, organizational charts, family trees, etc.):
3. Circle the number that best indicates your experience with the Treemap Visualization
4. Circle the number that best indicates your experience with the ArcTree Visualization
5. Are you currently a student at the University of Calgary?
6. What is your gender (for statistical purposes)?

Semi-structured interview questions

The following questions were generally used during the interview. Based on responses from the participants or suggestions or concerns they raised different questions were sometimes introduced.

1. The visualizations were created to help viewers understand and analyze data. Were you able to understand the organization of the data with these designs?
2. Which of both visualizations were you most comfortable with using?
3. With which visualization did you feel more confident about your results?
4. Do you see any serious problems with either of the presented visualizations?
5. Did you have any serious problems answering the questions? If yes, what part of the visualization do you think caused the problems?
6. Do you have any other suggestions?

Pilot study results

Statistics were created using SPSS for Windows 13.0 (evaluation version). Some of the data from the analysis is presented in the following. The following abbreviations are used: ATS = ArcTree small, ATM = ArcTree medium, ATL = ArcTree large, TMS = Treemap small, TMM = Treemap medium, TML = Treemap large.

Descriptive Statistics

See Table 1 and Table 2.

Mean	Std.	Deviation	N
ATS	6.6319	1.06445	10
ATM	10.8093	3.01088	10
ATL	10.5884	2.40799	10
TMS	8.7112	3.07253	10
TMM	16.2607	4.35235	10
TML	16.5145	4.93788	10

Table 1. Analysis of response time

Mean	Std.	Deviation	N
ATS	.10	.316	10
ATM	.10	.316	10
ATL	.30	.483	10
TMS	.10	.316	10
TMM	.60	.699	10
TML	.60	.699	10

Table 2. Analysis of response accuracy

Estimated Marginal Means

See Table 3 and Table 4 for information on response time statistics. See Table 5 and Table 6 for information on response accuracy statistics. Table 7 introduces a pairwise comparison between layout*size.

Raw Data

Table 8 and Table 9 introduce raw data collected during the pilot study.

(I) layout	(J) layout	Mean Difference (I-J)	Std. Error	Sig.(a)	95% Confidence Interval for Difference(a)	
					Lower Bound	Upper Bound
1	2	-4.486(*)	1.050	.002	-6.861	-2.110
2	1	4.486(*)	1.050	.002	2.110	6.861
Based on estimated marginal means						
* The mean difference is significant at the .05 level.						
a Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).						

Table 3. Pairwise comparison Layout 1 and Layout 2 according to time.

(I) layout	(J) layout	Mean Difference (I-J)	Std. Error	Sig.(a)	95% Confidence Interval for Difference(a)	
					Lower Bound	Upper Bound
1	2	-5.863(*)	.527	.000	-7.055	-4.671
1	3	-5.880(*)	.702	.000	-7.468	-4.292
2	1	5.863(*)	.527	.000	4.671	7.055
2	3	-.016	.339	.962	-.783	.750
3	1	5.880(*)	.702	.000	4.292	7.468
3	2	.016	.339	.962	-.750	.783
Based on estimated marginal means						
* The mean difference is significant at the .05 level.						
a Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).						

Table 4. Pairwise comparison size 1, 2, and 3 according to time.

(I) layout	(J) layout	Mean Difference (I-J)	Std. Error	Sig.(a)	95% Confidence Interval for Difference(a)	
					Lower Bound	Upper Bound
1	2	-.267	.130	.070	-.560	.026
2	1	.267	.130	.070	-.026	.560
Based on estimated marginal means						
* The mean difference is significant at the .05 level.						
a Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).						

Table 5. Pairwise comparison Layout 1 and Layout 2 according to error rate.

(I) layout	(J) layout	Mean Difference (I-J)	Std. Error	Sig.(a)	95% Confidence Interval for Difference(a)	
					Lower Bound	Upper Bound
1	2	-.250	.134	.096	-.554	.054
1	3	-.350(*)	.107	.010	-.591	-.109
2	1	.250	.134	.096	-.054	.554
2	3	-.100	.208	.642	-.571	.371
3	1	.350(*)	.107	.010	.109	.591
3	2	.100	.208	.642	-.371	.571
Based on estimated marginal means						
* The mean difference is significant at the .05 level.						
a Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).						

Table 6. Pairwise comparison size 1, 2, and 3 according to error .

size	(I) layout	(J) layout	Mean Difference (I-J)	Std. Error	Sig.(a)	95% Confidence Interval for Difference(a)	
						Lower Bound	Upper Bound
1	1	2	-2.079(*)	.855	.038	-4.013	-.146
1	2	1	2.079(*)	.855	.038	.146	4.013
2	1	2	-5.451(*)	1.517	.006	-8.883	-2.020
2	2	1	5.451(*)	1.517	.006	2.020	8.883
3	1	2	-5.926(*)	1.433	.003	-9.169	-2.684
3	2	1	5.926(*)	1.433	.003	2.684	9.169
Based on estimated marginal means							
* The mean difference is significant at the .05 level.							
a Adjustment for multiple comparisons: Least Significant Difference (equivalent to no adjustments).							

Table 7. Pairwise comparison layout*size according to time.

Participant	ATS	ATM	ATL	TMS	TMM	TML
1	6.66	9.27	14.20	6.79	19.75	16.04
2	6.86	9.49	11.98	7.35	16.94	15.86
3	6.84	8.63	7.84	6.34	11.68	11.55
4	5.10	8.39	9.32	6.37	12.50	14.20
5	8.09	13.29	12.42	7.98	16.61	15.94
6	8.51	12.06	9.92	16.44	25.16	27.35
7	5.60	6.80	8.83	8.44	16.72	15.96
8	5.81	13.56	7.29	8.85	10.46	11.79
9	6.05	16.70	13.79	11.18	18.58	22.84
10	6.80	9.91	10.29	7.36	14.22	13.61
mean	6.631928571	10.80927143	10.58835714	8.7112	16.26074286	16.51445714

Table 8. Raw data of mean answer times for tree layout and size over each task.

Participant	ATS	ATM	ATL	TMS	TMM	TML
1	0	0	0	0	1	0
2	0	0	1	0	0	0
3	0	0	1	0	0	1
4	0	0	1	0	0	0
5	0	0	0	0	1	0
6	0	1	0	1	1	1
7	0	0	0	0	2	1
8	0	0	0	0	0	1
9	1	0	0	0	0	2
10	0	0	0	0	1	0
total	1	1	3	1	6	6

Table 9. Raw data of errors per participant for tree layout and size over each task.