

Facilitating Visual Queries in the TreeMap Using Distortion Techniques

Kang Shi, Pourang Irani, and Pak Ching Li

Department of Computer Science, University of Manitoba
Winnipeg, Manitoba, R3T 2N2, Canada
shikang@gmail.com, {Irani, pkli}@cs.umanitoba.ca

Abstract. TreeMap is one a common space-filling visualization technique to display large hierarchies in a limited display space. In TreeMaps, highlighting techniques are widely used to depict search results from visual queries. To improve visualizing the queries results in the TreeMap, we designed a continuous animated multi-distortion algorithm based on fisheye and continuous zooming techniques. To evaluate the effectiveness of the new algorithm, we conducted an experiment to compare the distortion technique to the traditional highlighting methods used in TreeMaps. The results suggest that the multi-distortion technique is only effective with small result sets but not as effective as simple highlighting for large search result sets.

Keywords: TreeMap, distortion, searching, visual query, fisheye, continuous zooming, visualization.

1 Introduction

Search techniques are widely used in many areas of interaction such as searching in file systems (hierarchical data). Generally, a search action includes three steps: users submit search criteria to the system, the system performs the search in the database, and finally the results of the search are represented to the users. In most cases, such as search in Windows Explorer, the search results are presented in plain text to the users. However, this type of representation is not effective and has some disadvantages. One disadvantage is that the large quantities of search results can not be represented in a limited display space. The users need to scan and scroll and only can view a part of the result set. Another disadvantage is that users have difficulty in knowing the priority and the relationship between two or more search results. In addition, users also find it difficult to know the relationship between a search result and a non-result.

In this paper we introduce a visualization technique to represent search results in hierarchical data. Hierarchical data, typically represented in the form of a tree, are widely used in a variety of areas such as a file system. However, large hierarchies require complex interaction models as they typically exceed the available display space. Space-filling visualization, such as TreeMaps [4] or Sunburst [5], is one approach to resolving this problem. TreeMaps make efficient use of the display area

and provide limited structural information. In the TreeMap, the display area is divided into nested rectangular regions to map an entire hierarchy of nodes and their children. Each node uses an amount of space relative to the weight of the item being represented in the hierarchy.

To facilitate a search results visualization technique in the TreeMap, we applied animated fisheye and semantic zooming techniques to build a multi-distortion algorithm. With this distortion algorithm, users get the information about the priority of each result and can visually identify the relations between the results. Furthermore, some result nodes that are not visible in the TreeMap could be expanded to a visible size. We conducted an experiment to evaluate the effectiveness of the distortion technique. Results show that the distortion technique is effective only under a few conditions, but highlighting is still quite effective for presenting search nodes in TreeMaps.

2 Related Work

This paper is primarily inspired by the concept of continuous semantic zooming (CSZ) developed by Schaffer et al [2]. This technique is characterized by two distinct but interrelated components: continuous zooming [1] and presentation of semantic content at various stages of the zoom operation. CSZ manages a 2D display by recursively breaking it up into smaller areas. A region of interest becomes the focus and as the continuous zoom is applied successive layers of a display “open up”. At each level of the operation the technique enhances continuity through smooth transitions between views and maintains location constraints to reduce the user’s sense of spatial disorientation. The amount of detail shown in parts of the display is controlled by pruning the display and presenting items of non interest in summary form. The approach we have implemented is similar in concept to the continuous semantic zoom: smooth transition between zoom levels is applied and content visibility is increased as the nodes enlarge.

To implement the animated distortion, Shi et al [3] developed a uni-distortion algorithm to continuously change the size of a single node in the TreeMap. The algorithm is similar in concept to the continuous semantic zoom: smooth transition between zoom levels is applied and content visibility is increased as the nodes enlarge. Compared to the traditional method of browsing file contents in the TreeMap, the uni-distortion algorithm allows users to open nodes and view the contents without opening multiple layers of the hierarchy. While a target node in the TreeMap is selected, the uni-distortion algorithm executes in three steps: computing the neighbour nodes of the target node, changing the size of the neighbour nodes, and distorting neighbour nodes.

The distortion algorithm in the TreeMap can also be applied to the representation of search results. For instance, after viewing the TreeMap, users may typically be interested in searching for items with specific content (for example, files with certain keywords). Typically this type of interaction will result in obtaining a set of result items with each item having a degree of priority. In this case, the distortion algorithm

can be applied to multiple nodes simultaneously. The multi-distortion algorithm is an extension of the uni-distortion algorithm with two new rules.

3 Algorithm

The multi-distortion algorithm is expanded from the uni-distortion algorithm. Some major differences between the multi-distortion technique and the uni-distortion technique are as follows. In the multi-distortion algorithm, a target node cannot become a neighbour of other target nodes, and one node cannot become a neighbour of two target nodes. Furthermore, additional mappings are used in the multi-distortion algorithm. The distortion method has one primary attribute: the amount of distortion (amplitude). This attribute can be used to distinguish the levels of significance for each result. Therefore, significance is assigned to the amplitude of the distortion. For instance, a distortion with large amplitude can imply higher priority content than a distortion with small amplitude. The behavior of the multi-distortion algorithm is depicted in figure 1. Figure 1 (a) is the initial state of the TreeMap. The highlighted nodes are the search result nodes which will get distorted. Figure 1 (b) to (d) shows that the size of the nodes increased due to the distortion. The amplitude of the distortion indicates the priority of the node. Larger amplitude represents higher priority. Figure 1 (e) to (f) show that the size of nodes decrease because of the distortion. After state (f), the TreeMap returns back to the original state of the TreeMap (a).

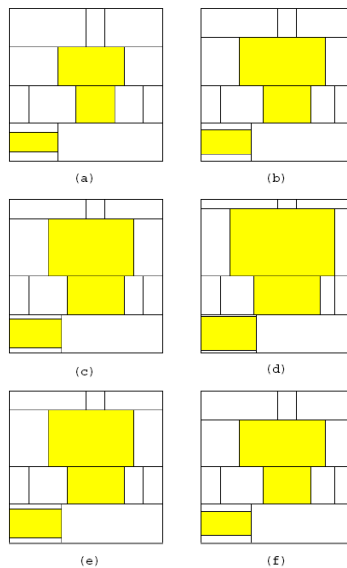


Fig. 1. The behavior of the multi-distortion algorithm. Figure (a) is the original state, figure (b)-(d) indicate the size of each node increased, figure (e)-(f) indicate the size of each node decreased. After state (f), the state of TreeMap goes back to (a).

Algorithm 1 describes the distortion of one node in the multi-distortion technique. It starts by determining the neighbours of the node of result (node A). If the neighbours are not other result nodes or neighbours of other result nodes, it decreases the size of these nodes. The size of A increase at the same time to provide the distortion effect. Then the entire TreeMap is redrawn. This process is repeated until an external event stops the distortion process or the node has reached a fixed maximum width or height.

Algorithm 2 describes the final multi-distortion algorithm. A1, A2, ..., An are search result nodes which should be distorted. Algorithm 11 is applied one by one on these result nodes to generate multiple distortions.

Algorithm 1.

```

OneNodeinMultiDistortAlgorithm(A)
global gLeftNeighbour, gRightNeighbour
global gTopNeighbour, gBottomNeighbour
bool isResult \ \Whether a node is a search result
bool isNeighbour \ \Whether a node is a neighbour of another search result
gLeftNeighbour = ComputeLeftNeighbour(A)
gRightNeighbour = ComputeRightNeighbour(A)
gTopNeighbour = ComputeTopNeighbour(A)
gBottomNeighbour = ComputeBottomNeighbour(A)
while DISTORTING = true and A.width < MAX_WIDTH and A.height < MAX_HEIGHT
do
    if not gLeftNeighbour.isResult and not gLeftNeighbour.isNeighbour
    then DistortLeft(amount)
    if not gRightNeighbour.isResult and not gRightNeighbour.isNeighbour
    then DistortRight(amount)
    if not gTopNeighbour.isResult and not gTopNeighbour.isNeighbour
    then DistortTop(amount)
    if not gBottomNeighbour.isResult and not gBottomNeighbour.isNeighbour
    then DistortBottom(amount)
    RedrawTreeMap(ROOT)
    Sleep(sleep int erval)

```

Algorithm 2.

```
MultiDistortAlgorithm( $A_1, A_2, \dots, A_n$ )  
OneNodeinMultiDistortAlgorithm( $A_1$ )  
OneNodeinMultiDistortAlgorithm( $A_2$ )  
...  
OneNodeinMultiDistortAlgorithm( $A_n$ )
```

4 Experiments

To evaluate the effectiveness of the multi-distortion algorithm, we designed this experiment to compare the distortion technique to the conventional method of showing search results (highlight) in the TreeMap. We defined priority as how often search keywords occur in a node, or whether all search keywords occur in a node. Priority in the distortion technique is mapped onto the amplitude of the animated distortions. In the highlight approach priority is indicated by the level of saturation. We anticipated the following effects from our experimental data:

Hypothesis 1: With small results set (number of results ≤ 5), the distortion technique is more effective than the highlight technique (speed and accuracy).

Hypothesis 2: With large results set ($5 < \text{number of results} \leq 10$) the highlight technique is more effective than the distortion technique (speed and accuracy).

4.1 Method

Twelve students participated in this experiment. Half of the subjects were assigned to one condition: Distortion first, and half of them were assigned to the other condition: Highlight first. Subjects were familiar with the concept of searching in windows file systems. All subjects were familiar with the highlight technique and the TreeMap, but none had any experience using distortion to represent search results in the TreeMap.

One hierarchy containing one thousand files was used for this experiment. Two different types of search keywords were used for the experiment: long and short. Using the short search keywords, the search would generate 3, 4, or 5 search results (small set). Using the long search keywords, the search would generate 6, 8, or 10 search results (large set). Six short search keywords and six long search keywords were used in the experiment. To reduce learning effects, I used two sets of search keywords (Set A and Set B) which would generate the same numbers of search results but in entirely different positions in the TreeMap. Half the subjects started the experiment with the Distortion method and the other half started with the Highlight method. After completing the tasks using one set of keywords, the subjects switched to use the other set of keywords with the other method.

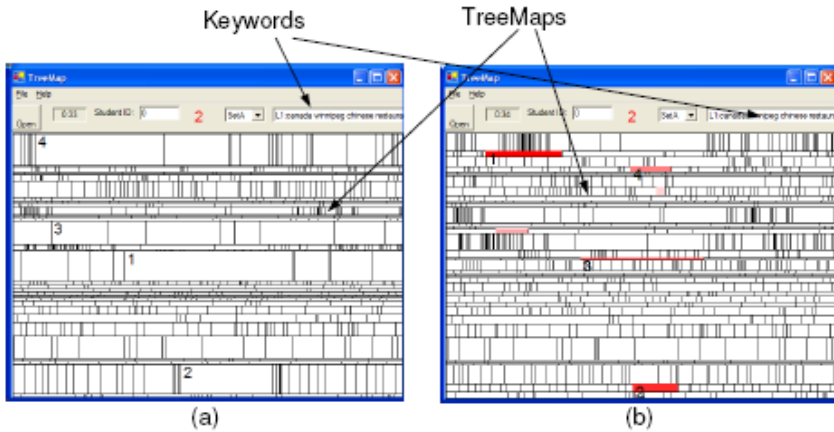


Fig. 2. Interface used in the experiment. Users are required to identify all search results and their priorities with (a) the distortion method, and (b) the highlight method.

As shown in Figure 2, this experiment consisted of two types of representations: distortion and highlight. In the distortion method, all search results were animated in and out using distortion algorithms until the subjects identified all results or the subjects withdrew from the task. The amplitude of the distortion represented the priority of each result. A result with larger amplitude has a higher priority. In the highlight method, all search results were filled by a color. Subjects were required to identify all the results or could withdraw from the task. The saturation of the color represented the priority of the result. A result with higher saturation has a higher priority.

4.2 Procedure

In this experiment, the task of the subjects is to identify the priority of the search results from highest to lowest. Before starting the experiment, each subject got familiarized with both representations. Each participant performed the tasks with six different short keywords and six different long keywords using both methods. The 12 trials were executed in the following sequence S1, S2, S3, ..., S6, L1, L2, L3, ..., and L6 (S represents the small sets of search results and L represents the large sets of search results). No time limit was set for the tasks, and the subjects were free to finish the trial if they could not identify all search results.

We recorded the time to execute the task, the number of results identified by the subjects, and whether the subject identified the correct priority for each search result in all conditions. In summary, the whole experiment involved: 12 subjects \times 2 main conditions \times 2 type of search result sets \times 6 trials = 288 trials in total.

4.3 Results

To test hypotheses 1 and 2 mentioned at the beginning of this section, we recorded the time it took the participants to select the items that are in the result set. Accuracy in

Table 1. The average completion times and average error rates for small and large result sets with both methods (standard deviations are in parentheses)

		Time	Error Rate
Distortion	Small Set	5.01 (1.47) sec	4.48%
	Large Set	7.06 (2.83) sec	7.43%
Highlight	Small Set	2.43 (0.66) sec	3.49%
	Large Set	3.30 (1.11) sec	2.65%

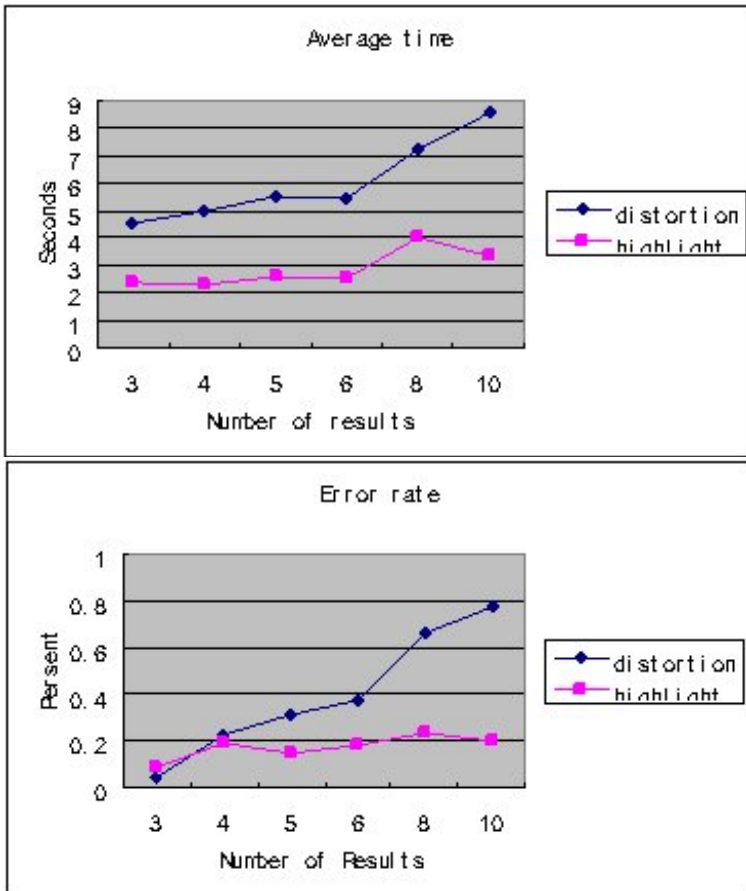


Fig. 3. Average time and average error rate with distortion and highlight methods

selecting the order of importance was also recorded, i.e. to get a perfect score the participant would need to select the items from the highest priority to the lowest priority in the correct order.

Time to locate search items and accuracy in determining the correct priority were analyzed using a repeated measure uni-variate ANOVA and a paired sample T Test. The results are summarized in Tables 1, and the average time and average error rate are described in Figure 3. An alpha level of 0.05 was used for all statistical tests. The main effect of representation type on time to complete the task and the main effect of representation type on accuracy were significant, $F(1, 11) = 44.61, p < 0.001$ and $F(1, 11) = 19.25, p = 0.001$ respectively. Participants performed significantly faster and more accurately when the search results were highlighted than when they are presented using distortion.

The main effect of the size of the search result on time to complete the task was significant, $F(1, 11) = 11.68, p < 0.001$. Participants performed significantly faster when they were presented with a small set of search results (less than five items in the result set) than when they were presented a large set of results. However, the main effect of the size of the search result on accuracy was not significant $F(1, 11) = 3.63, p = 0.083$.

The participants completed the tasks faster when the search results were highlighted vs. distorted in a small result set ($T(1, 11)=6.88, p < 0.001$). However, The analysis also suggests that participants were not less accurate in identifying the priority of items in the result set with the distortion technique than with the highlight representation in the small result set ($T(1, 11)=1.16, p = 0.27$). This does not support Hypothesis 1. In the large result set participants performed the task significantly faster and were more accurate when the results were highlighted than when they were distorted ($T(1, 11)=5.27, p < 0.001$ and $T(1, 11)=7.06, p < 0.001$, respectively). This is a support of hypotheses 2.

5 Conclusion

In this paper we introduced a new multi-distortion visualization technique to facilitate the visualization of search results in the TreeMap, and conducted an experiment to evaluate the effectiveness of the new technique. The results of the experiment suggest that the multi-distortion technique is not as effective as simple highlighting for showing search result sets. This is primarily due to the high level of distractibility created by the animations from the multi-distortion technique. Another possible reason is having nodes distort at different levels of amplitude may not enhance focus and user concentration. The distortion might only be beneficial when one node or a few nodes (less than 3 nodes) need to be in focus.

However, the motivation in using animated distortions was to allow small nodes that are not clearly visible, to become visible. An alternative method to display the distortions simultaneously would have been to show them in series starting from the highest priority to the lowest priority. This would have allowed users to locate nodes that would have not been visible otherwise but still provide a fair indication of items in the tree. We are pursuing this research track in the near future.

References

1. Bederson, B.B., Hollan, J.D., Perlin, K., Meyer, J., Bacon, D., Furnas, G.W.: Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages and Computing* 7(1), 3–32 (1996)
2. Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., Roseman, M.: Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM TOCHI* 3(2), 162–188 (1996)
3. Shi, K., Irani, P.P., Li, P.C.: An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations. *IEEE InfoVis*, pp. 81–88 (2005)
4. Shneiderman, B.: Tree visualization with TreeMaps: a 2D Space-Filling Approach. *ACM TOG* 11(1), 92–99 (1990)
5. Stasko, J.T., Zhang, E.: Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. *IEEE Infovis*, pp. 57–65 (2000)