Frutiger Ltd Std 65 Bold, 56pt

Interaction Techniques for a Strata Treemap

Oh-hyun Kwon*, Junghong Choi† and Kyungwon Lee‡ — Frutiger Ltd Std 65 Bold, 25pt

*‡Division of Digital Media, Ajou University, †HCI Lab, Yonsei University
*kwonoh6@ajou.ac.kr, †junghong@yonsei.ac.kr, ‡kwlee@ajou.ac.kr — Frutiger Ltd Std 65 Bold, 16pt

AJOU UNIVERSITY

digitalMEDIA

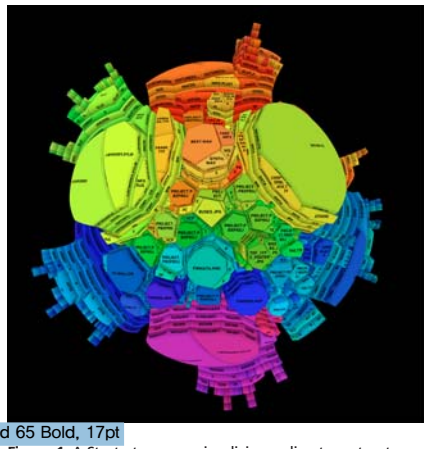
Frutiger Ltd Std 65 Bold, 32pt

Abstract

Frutiger Ltd Std 55 Roman, 18pt

The Strata treemap[1] is a new approach to hierarchy visualization that improves hierarchical structure visibility without distorting node sizes or missing nodes. In this poster, we present useful interaction techniques and visual cues that fit the Strata treemap.

More information and a demo video can be found at http://idlab.ajou.ac.kr/stratatreemap



Frutiger Ltd Std 65 Bold, 17pt

Figure 1. A Strata treemap visualizing a directory structure (6138 nodes, 12 levels). The weight of each leaf node corresponds to the file size. Structure-based coloring is applied.

Frutiger Ltd Std 55 Roman, 17pt

Objectives

Internal nodes have to be shown in order to improve hierarchical structure visibility. It is necessary to find a new space to naturally illustrate the internal nodes and avoid node size distortion. Node size distortion in radial space-filling techniques is caused by an increase in the space available for the extended circle radius. In Strata treemap, that space is used to provide a natural padding between sibling nodes. Fig. 2 shows a simple overview of idea of the Strata treemap;

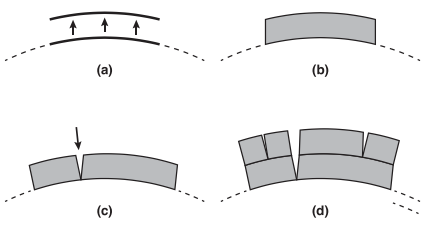


Figure 2. Basic idea of the Strata treemap

1. Formulate a treemap on the convex surface.
2. Extrude the surface of each node along its surface normal at the centroid (Fig. 2a, 2b), not by extending the radius of the sphere as done in RSF techniques.
3. Stack nodes according to the hierarchy structure (child nodes are stacked on their parent) (Fig. 2d).

Methodology

Navigating

Frutiger Ltd Std 65 Bold, 21pt

When a user selects a node(s), the point-of-interest can then be moved to the center of the selected node(s) for the camera to rotate (orbit) around the selected node(s). When the camera position changes, the labels rotate to correct the reading direction.

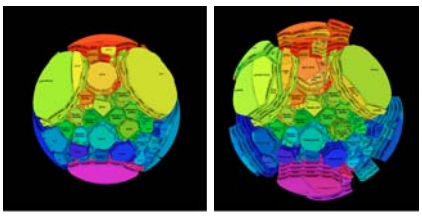


Figure 4. Peeling-off/Stacking-up. From root level, the left figure shows 3 levels and the right figure shows 7 levels.

Peeling-off/Stacking-up

To hide/expose sub-branches of the hierarchy, users can specify the local/global maximum level of visible nodes (Fig. 4).

Highlighting

Highlight the node(s) that have structural relations (e.g., ancestors, descendants, equal-level) with the currently highlighted node(s) in the hierarchy (Fig. 3a-c).

Highlight the node(s) that have similar attributes (e.g., file-type, weight, modified time) with the currently highlighted node(s) in the hierarchy (Fig. 3d).

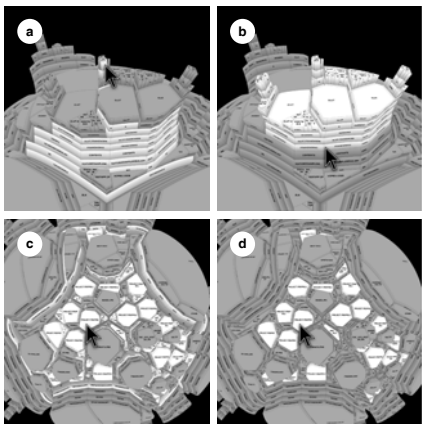


Figure 3. Highlight the node(s) that have structural relations (e.g., ancestors (a), descendants (b), equal-level (c)) or similar attributes (e.g., equal-file-type (d)) with the currently highlighted node (under the mouse cursor).

Transforming Node Height

To display lower level nodes clearly, the node height can be adjusted in such a way that the node height decreases as the level increases (Fig. 5a, 5b). To emphasize highlighted or selected nodes, users can increase the heights of these nodes or decrease the heights of the other nodes. (Fig. 5c, 5d).

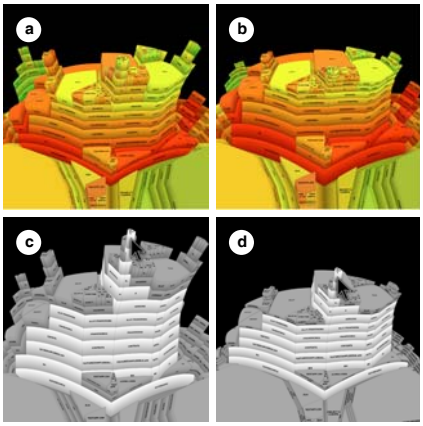


Figure 5. Transforming node height. Default height (a). The node height decreases as the level increases (b). Emphasizing the highlighted nodes by increase its height (c) or decrease the height of the other nodes (d).

Transforming Node Position

This technique is useful when users want to modify the layout (Fig. 6). If users compare node(s) across different portions of a layout, for example, users can manually/automatically reposition the node(s) until they are close enough to compare. The layout can be stably updated because it is based on a Voronoi diagram.

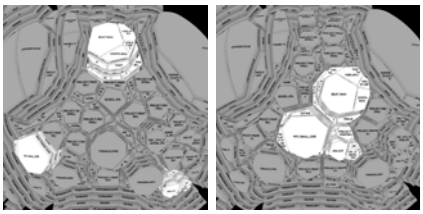


Figure 6. Transforming node position. To modify the layout, users can manually/automatically reposition the nodes.

Conclusion

Since these interaction techniques effectively utilize the geomertrical structure of the Stratat treemap, users can easily investigate a complex hierarchical data with the Strata treemap.

Our future work will includes techniques involving the use of transparency, layout methods using a curved Voronoi diagram and the partial surface of a sphere.

References

Frutiger Ltd Std 65 Bold, 25pt

1. Junghong Choi, Oh-hyun Kwon, and Kyungwon Lee. Strata treemaps. In ACM SIGGRAPH 2011 Posters, SIGGRAPH '11, pages 87:1. 2011.
2. Hao Lü and James Fogarty. Cascaded treemaps: examining the visibility and stability of structure in treemaps. In Proceedings of Graphics Interface, pages 259–266. 2008.
3. Hans-Jorg Schulz, Steffen Hadlak, and Heidrun Schumann. The design space of implicit hierarchy visualization: A survey. IEEE Transactions on Visualization and Computer Graphics, 17(4):393–411, 2011.

Frutiger Ltd Std 55 Roman, 16pt