Interactive Poster:
Exploration of the 3D Treemap Design Space

Hans-Jörg Schulz* Martin Luboschik† Heidrun Schumann‡

University of Rostock, Germany

Abstract

Inspired by Venn diagram layouts, the Treemap [6] is one of the most prevalent implicit tree visualization techniques. Ever since its publication, it has been modified and extended in many ways. This work presents a way to generate 3-dimensional Treemap visualizations by a 4-step procedure. It can be used for rapid prototyping and comparing different 3D Treemap layout approaches, to devise user studies on 3D Treemap layouts or for educational purposes.

Keywords: tree visualization, 3D Treemap, implicit graph layout

1 Introduction

The original Treemap [6] is a **2D**, **implicit** layout technique that uses **containment** of **rectangles** to indicate parent-child-relationships. These rectangles are aligned **parallel to the axes**, alternating between horizontal and vertical layout (**Slice and Dice**). Over the years, researchers have modified and extended the original Treemap with regard to all of these characteristics:

- the dimensionality has been extended to 3D, i.e. as in Step-trees [4] or Treecubes [10]
- the implicit edge representation has been partially modified to explicitly drawn edges in Elastic Hierarchies [13]
- the containment relationship has been substituted by overlap in the Beamtree technique [11]
- the used graphics primitives have been changed from rectangles to circles [12] and convex polyhedra [2]
- the alignment to axes has been turned into radial arrangements (as in Pietrees [8])
- the layout mode has been enhanced from the original Slice-and-Dice method [6] to other techniques like Squarified Treemaps [5] and Quantum Treemaps [3]

All of the above characteristics can be changed in combination (e.g., using overlap of circles in a radial arrangement), yielding a large number of possible Treemap configurations. Or framework is the first to provide the means to systematically explore this vast set of Treemap techniques. In this work, we focus on 3-dimensional Treemaps that completely rely on implicit edge representations. This pretty much fixes the first two items in the above list, but leaves the others to be freely combined within our framework.

2 A new 3D Treemap in 4 Steps

When devising a 3D Treemap configuration from within our framework, its parametrization is done in four steps:

*e-mail: hjschulz@informatik.uni-rostock.de
†e-mail: luboschik@informatik.uni-rostock.de
‡e-mail: schumann@informatik.uni-rostock.de

1. **Specify the containment relationship:** In this step, the user specifies how the parent-child-relationship should be encoded in the implicit representation. There are three possible choices: containment, adjacency and overlap. All of them are exemplified in Figure 1 using cuboids as graphics primitives.

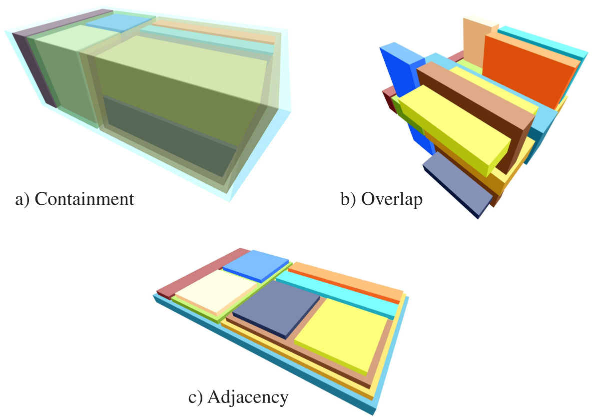

Figure 1: Implicit edge representations exemplified.

2. **Choose the graphics primitive:** After deciding about the edge representation in the first step, this step is about specifying the 3-dimensional representation of nodes. Primitives that have been used for 3D Treemaps up to now are cuboids [4, 10], cylinders [11, 12] and frustums of pyramids [1]. Yet, as shown in Figure 2, other graphical primitives like spheres can be imagined. Hence, it makes no sense to consider only a fixed set of graphics primitives, instead primitives should be provided to the framework in a plug-in manner.

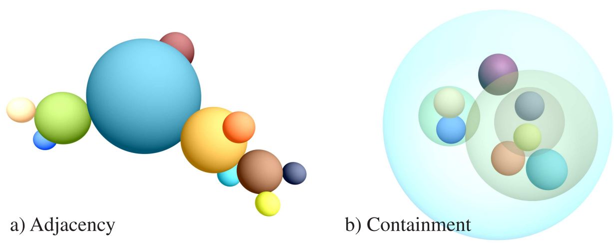

Figure 2: Two possible 3D Treemap configurations that use spheres as graphical primitves.

3. **Select a layout method:** Layout methods basically describe how the available space is distributed among the leaves of the hierarchy. While basic strategies like "Slice and Dice" or "Sphere Packing" are available by default, more sophisticated techniques can be realized using the plugin concept as the need arises. This flexibility allows users to create and experiment with new techniques that use alternative or uncommon layout methods. Examples for such are depicted in Figure 3.

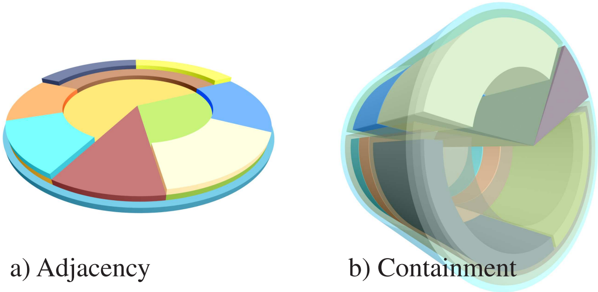

Figure 3: Two possible 3D extensions of the Pietree technique [8], in which radial alignment is combined with an axes-parallel stacking of the used primitives resulting in a hybrid alignment approach.

4. **Decide upon its alignment:** This step specifies the alignment of the layout. It is dependent on the layout method and provides an alignment parameter to it. This dependency is inevitable, as not all layout techniques can easily be used for both, axes-parallel and radial arrangement. Furthermore, besides configurations that are entirely axes-parallel (like those in Figure 1) and configurations that are entirely radial (like the ones depicted in Figure 2), there exist hybrid alignment approaches (like the ones depicted in Figure 3).

3 Our Framework

Our framework provides an interface that allows users to specify 3D Treemap configurations according to the 4-step specification process described above. Then, one or more specified configurations can be used to visualize a given hierarchy. The framework can be used to view different 3D Treemap configurations side by side to allow for their simultaneous exploration. Additionally, a standard treeview is linked to these views to ensure that all nodes of the hierarchy can be selected easily, even if they are occluded or otherwise hard to pick from the 3D representation. Thus, highlighting a node in the treeview is propagated to all of the other views and highlights the very same node there. This mechanism is very useful for comparing new 3D Treemap prototypes with existing techniques. Moreover, comparing and contrasting with familiar techniques makes it easier for users to get accustomed to novel alternative designs.

Secondly, the described concept of a modular, stepwise parameterization of 3D Treemap configurations provides a new systematical basis for conducting user studies. It is possible to analyze the influence of the individual characteristics by keeping everything else fixed and altering only the parametrization of interest. This allows to investigate questions like "Which implicit edge representation technique is best suited for identification and comparison of data items in case of axes-parallel cuboids that are arranged in a Slice-and-Dice fashion?"

A prototype of the described framework will be available to the InfoVis-attendees for a hands-on demonstration at the poster desk. So far, it includes all edge representations, primitives, alignments and layouts that are necessary to specify most of the existing 3D Treemap configurations like Treecube [10] or StepTrees [4]. Since they are provided in the said modular fashion, also other possible combinations of these edge representations, primitives, alignment- and layout-strategies can be generated with the prototype.

4 Conclusion

With our framework we have developed a platform for rapid prototyping of 3D Treemap visualizations and their interactive evaluation. Different layout aspects have been singled out into a modular concept that allows to easily put together new Treemap configurations from known building blocks. This enables the user to go beyond the number of known 3D Treemap techniques by systematically exploring the range of possible layout combinations. That way, the user can really find the very technique that perfectly fits task and data, even if it has not been described before. To further increase the number of choices for the user, future work will include the adaptation of our framework to 3-dimensional extensions of other well-known implicit techniques like Sunburst [9] and Icicle Plot [7].

References

[1] K. Andrews, J. Wolte, and M. Pichler. Information pyramids^{TM}: A new approach to visualising large hierarchies. 1997.
[2] M. Balzer and O. Deussen. Voronoi treemaps. In *InfoVis 2005*, pages 49–56, 2005.
[3] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.
[4] T. Bladh, D. A. Carr, and J. Scholl. Extending tree-maps to three dimensions: A comparative study. In *APCHI 2004*, pages 50–59, 2004.
[5] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *Data Visualization 2000*, pages 33–42, 2000.
[6] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Visualization 1991*, pages 284–291, 1991.
[7] J. B. Kruskal and J. M. Landwehr. Icicle plot: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983.
[8] R. O'Donnell, A. Dix, and L. J. Ball. Exploring the pietree for representing numerical hierarchical data. In *HCI 2006*, 2006.
[9] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *InfoVis 2000*, pages 57–65, 2000.
[10] Y. Tanaka, Y. Okada, and K. Niijima. Treecube: visualization tool for browsing 3d multimedia data. In *IV 2003*, pages 427–432, 2003.
[11] F. van Ham and J. J. van Wijk. Beamtrees: Compact visualization of large hierarchies. In *InfoVis 2002*, pages 93–100, 2002.
[12] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of large hierarchical data by circle packing. In *SIGCHI 2006*, pages 517–520, 2006.
[13] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *InfoVis 2005*, pages 57–64, 2005.