Constructing Hierarchical Continuity in Hilbert & Moore Treemaps

Willy Scheibel and Jürgen Döllner

University of Potsdam, Digital Engineering Faculty, Hasso Plattner Institute, Germany

Abstract

The Hilbert and Moore treemap layout algorithms are based on the space-filling Hilbert and Moore curves, respectively, to map tree-structured datasets to a 2D treemap layout. Considering multiple snapshots of a time-variant dataset, one of the design goals for Hilbert and Moore treemaps is layout stability, i.e., low changes in the layout for low changes in the underlying tree-structured data. For this, their underlying space-filling curve is expected to be continuous across all nodes and hierarchy levels, which has to be considered throughout the layouting process. We propose optimizations to subdivision templates, their orientation, and discuss the continuity of the underlying space-filling curve. We show real-world examples of Hilbert and Moore treemaps for small and large datasets with continuous space-filling curves, allowing for improved layout stability.

CCS Concepts

• ***Human-centered computing*** → ***Treemaps; Visualization design and evaluation methods;***

1. Introduction

Tree-structured data can be visualized using tree visualization techniques [SHS11]. One family of such techniques is the treemap [JS91], which allows for different layout algorithms with each their different layout characteristics [STLD20, SLD20]. To support a users' mental map [MELS95], one goal for optimization is the stability of the layout over time [TC13], e.g., by adhering to the *principle of visual-data correspondence* [KS14]. One approach to achieve stability for treemap layouts is the placement of nodes along a space-filling curve. Early examples for this approach are the Strip treemap that uses a sweep curve [BSW02] or Contour treemaps that uses Spiral and S-Shape curves [TS07]. As a more complex curve, the Hilbert curve was used to create Jigsaw Maps, which derive non-rectangular layout elements for nodes along the curve and thus create non-convex treemaps [Wat05]. Later, Tak and Cockburn proposed the Hilbert and Moore treemaps that are based on Hilbert and Moore curves, respectively, but result in rectangular treemaps [TC13]. Both are based on a node partitioning and subsequent recursive rectangular subdivision of the layout [SWBD21] and depend on a set of subdivision templates that are proposed in their original publication [TC13].

When using space-filling curves, the layout stability depends on their continuity across all nodes and hierarchy levels. Each discontinuity may result in *jumps* of individual layout elements, lowering the measured and perceived stability. In previous studies, this continuity was not ensured and an inferior stability was reported [SSV18, VSC*20]. We found the limiting factors were the choice of templates and their orientation during recursive subdivision. We contribute details to the choice of subdivision templates and their orientation to ensure a continuous space-filling curve.

2. Reconsidering Templates, Orientation, and Curve Affinity

The Hilbert and Moore treemap layout algorithms use a recursive rectangle subdivision along a space-filling curve, i.e., the Hilbert or the Moore curve, respectively. The high-level algorithm is defined as follows: For each group of sibling nodes, the ordered list of weights is recursively divided into up to four parts, forming quadrants (Figure 1 from supplemental material).

Templates. To lay out parts, a set of layout templates is instantiated and ranked by their aspect ratios. The template with the closest mean aspect ratio to a target aspect ratio is chosen to derive the actual layout element of a node. This target aspect ratio is a parameter to the layout process, where different values such as 1.0 or the golden ratio Φ can be considered [KHA10]. The original authors proposed eleven templates for two and three cuts [TC13]. We propose to extend this set to include single-cut templates and a no cut template for completeness, adding up to 14 templates (Figure 2 from supplemental material).

Orientation. We further propose to consider a local orientation of the current layout element as well as orientations of the sub-templates that are to be instantiated. We extend the definition of each template to define orientations of its sub-templates (Figure 2 from supplemental material). The current orientation and the orientation of the sub-template determine the orientation of the resulting layout element.

Curve Affinity. The first step to ensure a continuous space-filling curve is to consider the templates to be not universally applicable during subdivision, but to have an affinity to be used for Hilbert or Moore curves ($\mathcal{H}$ and $\mathcal{M}$). Further, for each node, continuity of the space-filling curve has to hold for all of its child nodes and for all of its sibling nodes. For effective layouting of siblings, this continuity requires a start and end point of each partial curve to be adjacent to their predecessor and successor partial curves. Similarly, child nodes have to continue the space-filling curve of a sibling of the parent node. As a result, the templates of the Moore treemap are only applicable to the root layout as the start and end of the Moore treemap are towards the center. The proposed orientations require further handling of special cases during template instantiation. The templates together with their orientations can be interpreted differently, regarding on the local orientation of the space-filling curve being clockwise or counterclockwise (CW and CCW). This virtually extends the templates with horizontally flipped versions.

3. Hilbert & Moore Treemaps with Continuous Curves

Following the proposed extensions allows to compute Hilbert & Moore treemaps with all nodes layed out on a continuous space-filling curve. This holds for each level of the tree-structured dataset as well as all leaf nodes. We evaluate the changes in the layouting process using both list datasets and actual tree-structured datasets. The software prototype is available as open source project on GitHub [SWB23]. Considering run-time performance, these additions do not introduce additional run-time complexity, as they only specialize the handling of templates during subdivision. An extension to the former implementation of Scheibel et al. upholds the linear run time of both layouting algorithms [SWBD21].

Debug Visualization. We show the resulting treemap layouts using a debug visualization: leaf nodes are rendered as colored rectangles where the color is mapped from their index in the datasets to grayscale. The color of one node does not change over multiple snapshots. To highlight the space-filling curve, we connect successor nodes (not necessarily siblings when handling actual tree-structured data) from center to center with a red line and orthogonal line routing, which would reveal a discontinued space-filling curve through crossings.

Popular Names Dataset. Using this visualization approach, we repeated the study of Sondag et al. using the popular names dataset from The Netherlands [SSV18]. The results are different but we show that the layout process does not introduce *jumps* (Figure 1). The highlighted space-filling curve shows no crossings, which indicates that no order of nodes changed in the layout with respect to the space-filling curve.

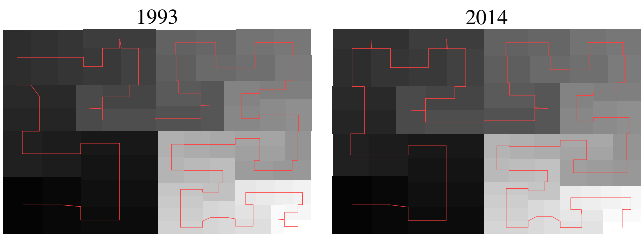

Figure 1: *Hilbert treemap layouts on the popular names dataset of 1993 and 2014, together with the corresponding space-filling curve [SSV18]. These layouts show a debug view to verify the continuity of the curve over the whole timespan. The target aspect ratio is chosen to be the golden ratio Φ and the subdivision optimization is the min-variance cut. The whole set of layouts as well as examples for Moore layouts can be seen in Figure 3 in the supplemental material.*

Software Repository Data. As examples of actual tree-structured datasets we use a number of different open source projects from GitHub with up to 100 000 leaf nodes (Table 1 from supplemental material). The datasets contain files with their directory structure of the source code repository and additional static source code metrics that can be used for weight mapping. The debug visualization shows that the space-filling curve is not discontinued (Kubernetes in Figure 2, additional examples in Figure 4–5 from supplemental material). Those examples show that the proposed extensions allow for a hierarchically continuous space-filling curve.

4. Conclusions

Optimizing treemap layouts for stability remains challenging. We discussed details on the subdivision process for Hilbert & Moore treemaps to further optimize their stability. By careful consideration of the templates and their orientation during subdivision, the continuity of the space-filling curves is ensured across all nodes and hierarchy levels. For this, we proposed to extend the number of templates, consider an affinity to the space-filling curves, and to further consider local orientations of the sub-templates. These variations have not introduced no increase in the run-time complexity and thus allows to be applicable for tree-structured datasets up to multiple hundreds of thousands nodes. For future work, we imagine to explore additional templates, subdivision approaches, and parameterization. We further see need for thorough evaluations of these and other treemap layout algorithms regarding datasets, parameterizations, and domains, as well as user perception [FSL*20].

Acknowledgments. We want to thank the anonymous reviewers for their suggestions to improve this paper. This work was partially funded by the Federal Ministry of Education and Research, Germany through grant 01IS20088B ("KnowhowAnalyzer").

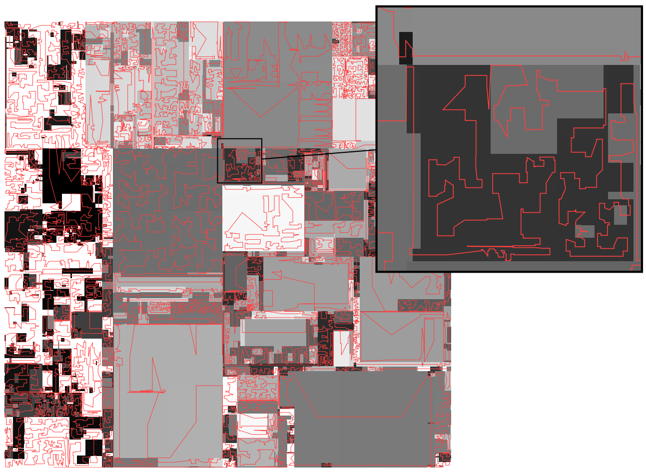

Figure 2: *A Hilbert treemap of a tree-structured software analytics dataset derived from the software project Kubernetes. The number of leaf nodes is 13 175 and the maximum depth is 16. The target aspect ratio is chosen to be 1.0 and the subdivision optimization is the min-variance cut.*

References

[BSW02] BEDERSON B. B., SHNEIDERMAN B., WATTENBERG M.: Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics 21*, 4 (2002), 833–854. doi:10.1145/571647.571649. 1

[FSL*20] FIEDLER C., SCHEIBEL W., LIMBERGER D., TRAPP M., DÖLLNER J.: Survey on user studies on the effectiveness of treemaps. In *Proc. 13th International Symposium on Visual Information Communication and Interaction* (2020), VINCI '20, ACM, pp. 2:1–10. doi:10.1145/3430036.3430054. 2

[JS91] JOHNSON B. S., SHNEIDERMAN B.: Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. In *Proc. 2nd Conference on Visualization* (1991), VIS '91, IEEE, pp. 284–291. doi:10.1109/VISUAL.1991.175815. 1

[KHA10] KONG N., HEER J., AGRAWALA M.: Perceptual guidelines for creating rectangular treemaps. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 990–998. doi:10.1109/TVCG.2010.186. 1

[KS14] KINDLMANN G., SCHEIDEGGER C.: An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2181–2190. doi:10.1109/TVCG.2014.2346325. 1

[MELS95] MISUE K., EADES P., LAI W., SUGIYAMA K.: Layout adjustment and the mental map. *Elsevier Journal of Visual Languages & Computing 6*, 2 (1995), 183–210. doi:10.1006/jvlc.1995.1010. 1

[SHS11] SCHULZ H.-J., HADLAK S., SCHUMANN H.: The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics 17*, 4 (2011), 393–411. doi:10.1109/TVCG.2010.79. 1

[SLD20] SCHEIBEL W., LIMBERGER D., DÖLLNER J.: Survey of treemap layout algorithms. In *Proc. 13th International Symposium on Visual Information Communication and Interaction* (2020), VINCI '20, ACM, pp. 1:1–9. doi:10.1145/3430036.3430041. 1

[SSV18] SONDAG M., SPECKMANN B., VERBEEK K.: Stable treemaps via local moves. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 729–738. doi:10.1109/TVCG.2017.2745140. 1, 2

[STLD20] SCHEIBEL W., TRAPP M., LIMBERGER D., DÖLLNER J.: A taxonomy of treemap visualization techniques. In *Proc. 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – Volume 3: IVAPP* (2020), IVAPP '20, SciTePress, pp. 273–280. doi:10.5220/0009153902730280. 1

[SWB23] SCHEIBEL W., WEYAND C., BETHGE J.: Hilbert and Moore treemap layouts prototype, 2023. varg-dev/hilbert-moore-treemap-layouts-prototype. doi:10.5281/zenodo.7934497. 2

[SWBD21] SCHEIBEL W., WEYAND C., BETHGE J., DÖLLNER J.: Algorithmic improvements on Hilbert and Moore treemaps for visualization of large tree-structured datasets. In *Proc. EuroVis 2021 – Short Papers* (2021), EG, pp. 115–119. doi:10.2312/evs.20211065. 1, 2

[TC13] TAK S., COCKBURN A.: Enhanced spatial stability with Hilbert and Moore treemaps. *IEEE Transactions on Visualization and Computer Graphics 19*, 1 (2013), 141–148. doi:10.1109/TVCG.2012.108. 1

[TS07] TU Y., SHEN H. W.: Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1286–1293. doi:10.1109/TVCG.2007.70529. 1

[VSC*20] VERNIER E., SONDAG M., COMBA J., SPECKMANN B., TELEA A., VERBEEK K.: Quantitative comparison of time-dependent treemaps. *Wiley Computer Graphics Forum 39*, 3 (2020), 393–404. doi:10.1111/cgf.13989. 1

[Wat05] WATTENBERG M.: A note on space-filling visualizations and space-filling curves. In *Proc. Symposium on Information Visualization* (2005), INFOVIS '05, IEEE, pp. 181–186. doi:10.1109/INFVIS.2005.1532145. 1