Interactive Revision Exploration using Small Multiples of Software Maps

Willy Scheibel, Matthias Trapp and Jürgen Döllner

Hasso Plattner Institute, University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, Germany

1 Introduction: Software Map

The *software map* [Bohnet and Döllner 2011] is a visualization technique based on the 2.5D treemap [Bladh et al. 2004] used to depict the structure and characteristics of one revision of a software system. It is configured by *map themes*, consisting of a mapping of metrics to visual variables (e.g., ground area, height, color, texture). A software map inhibits the following properties:

- Nested nodes and their padding depict the hierarchy.
- One revision is visualized using one map theme.
- A leaf node depicts a measurable software entity.
- The ground area of the leaf node depicts its size.
- Its height represents a mapped metric, defined by the map theme.
- Its color represents a mapped metric, defined by the map theme.

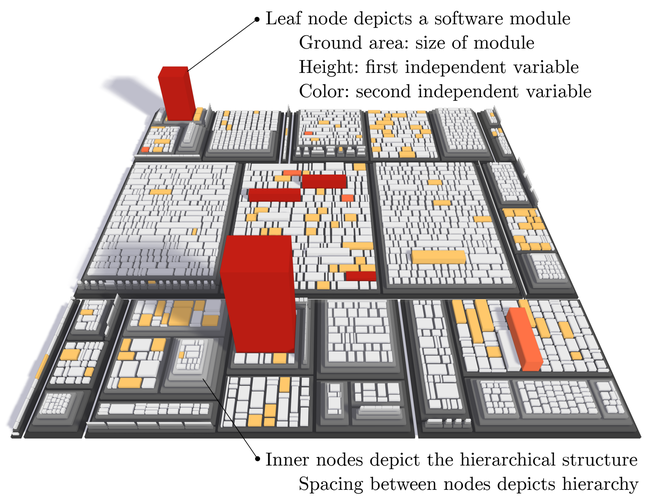

Figure 1: *A software map for the depiction of a software system.*

While software maps are useful to visualize one revision using one map theme, the interactive revision exploration is challenging. A software system under development is a tree comparison problem with both changing metric values and topology (category 3 or 4, depending on the metrics used [Guerra-Gomez et al. 2013]). To support the use case, the following concepts and characteristics of a softare map are missing:

- Depicting multiple revisions using the same map theme.
- Depicting multiple map themes using the same revision.
- Comparison between different depictions of the same software system.

We propose an extension to software maps where multiple instances using different revisions and map themes are arranged into a matrix, constituting a small multiples visualization [van den Elzen and van Wijk 2013].

2 Concept

We build a small multiples visualization of software maps by arranging the software maps in a grid. The x-axis is used for the time-component (i.e., differing revisions) and the y-axis for different map themes. To configure each software map, we differentiate between the *base configuration* (including, e.g., the dataset, the layouting algorithm and the node padding) that is used for all software maps and a *per small multiple configuration* that specifies the used revision and map theme. A graphical user interface allows to configure the revision per column and the map theme per row. Navigation techniques for the virtual 3D scenes are zoom, pan, and rotate, while the virtual cameras are all synchronized. A focus+context technique allows for highlighting one or multiple software maps for direct comparison.

Base Configuration Dataset: POCO Layout: Squarified Margin: 0.05		Columns: Revisions	
		Revision 100	Revision 150
Rows: Software Map Themes	Risk of Knowledge Drain	POCO Squarified (Margin 0.05) Revision 100 Risk of Knowledge Drain	POCO Squarified (Margin 0.05) Revision 150 Risk of Knowledge Drain
	Technical Debts	POCO Squarified (Margin 0.05) Revision 100 Technical Debts	POCO Squarified (Margin 0.05) Revision 150 Technical Debts

Figure 2: *Matrix-based configuration.*

3 Implementation

The prototype is implemented using C++, Qt5 and OpenGL while relying on *attributed point clouds* [Trapp et al. 2013] as geometry representation on the GPU and multi-frame sampling [Limberger et al. 2016] for the shading. Two rendering pipelines were implemented and evaluated: a multi-pass approach where each small multiple is rendered in its own draw call and a single-pass approach where all small multiples are rendered using one draw call.

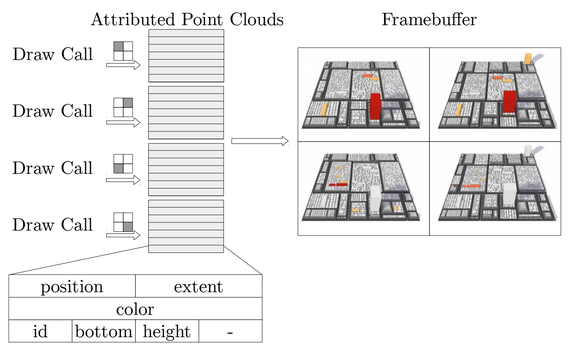

Figure 3: *The multi-pass rendering pipeline using per-pass viewport manipulation in multiple draw calls.*

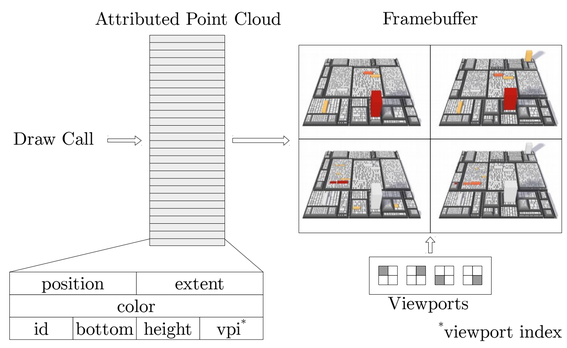

Figure 4: *The single-pass, single draw call rendering pipeline using virtual viewports and screen-space vertex displacement.*

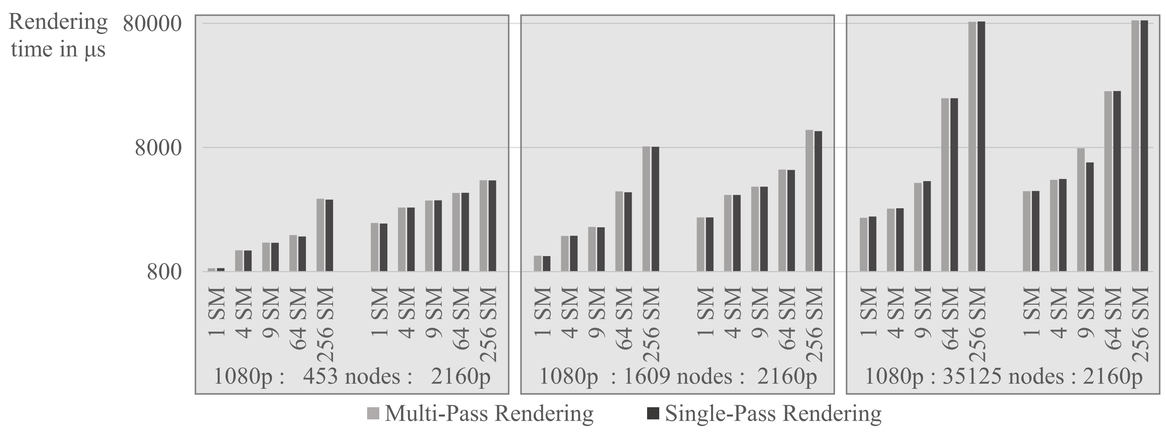

Figure 5: *Rendering performance comparison of single-pass rendering and multi-pass rendering (logarithmic scale), depending on the size of the software project (number of nodes), the resolution and the number of small multiples. The test system was an Ubuntu 14.10 x64 machine with an Intel Xeon at 8 × 2.8Ghz with 6GB RAM and an Nvidia GTX 680 graphics card.*

References

Bladh, T., Carr, D., and Scholl, J. 2004. Extending tree-maps to three dimensions: A comparative study. In *Computer Human Interaction*, vol. 3101 of *LNCS*.

Bohnet, J., and Döllner, J. 2011. Monitoring code quality and development activity by software maps. In *Proc. of the 2nd Workshop on Managing Technical Debt 2011*, ACM.

Guerra-Gomez, J., Pack, M. L., Plaisant, C., and Shneiderman, B. 2013. Visualizing change over time using dynamic hierarchies: Treeversity2 and the stemview. *IEEE TVCG 2013 19*, 12.

Limberger, D., Tausche, K., Linke, J., and Döllner, J. 2016. Progressive rendering using multi-frame sampling. *GPU Pro^7: Advanced Rendering Techniques*.

Trapp, M., Schmechel, S., and Döllner, J. 2013. Interactive rendering of complex 3d-treemaps. In *Proc. of GRAPP 2013*.

van den Elzen, S., and van Wijk, J. J. 2013. Small multiples, large singles: A new approach for visual data exploration. *Computer Graphics Forum 32*, 3.

4 Interactive Revision Exploration

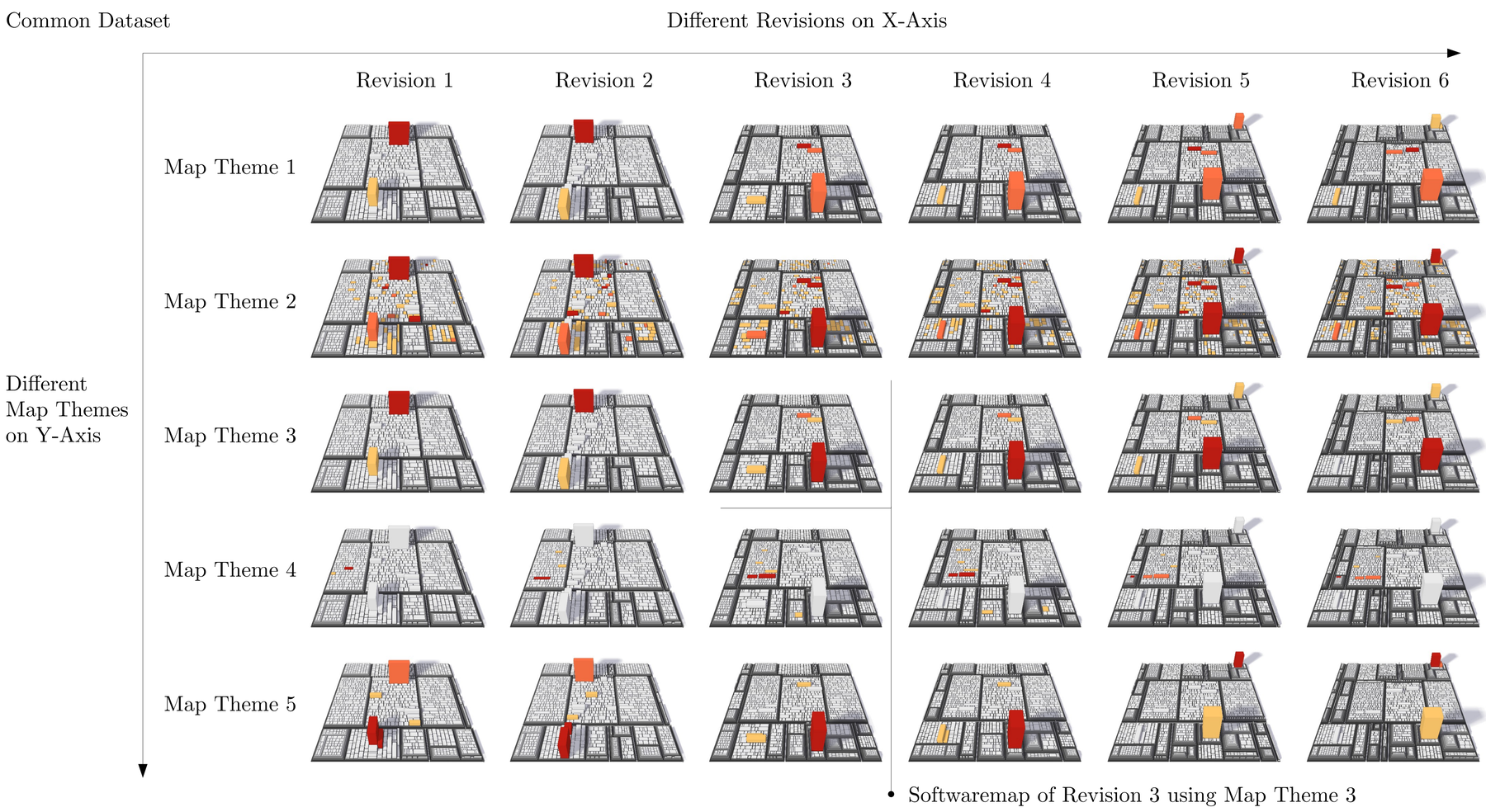

Figure 6: *A small multiples visualization of software maps showing 6 revisions (columns) and 5 different software maps themes (rows).*

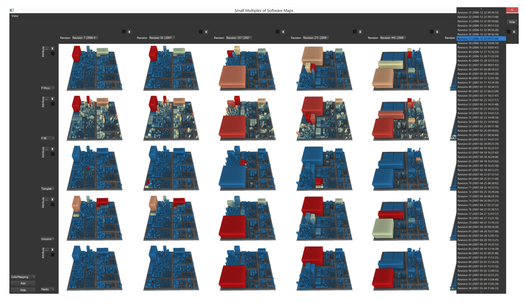

Figure 7: *Configuration user interface.*

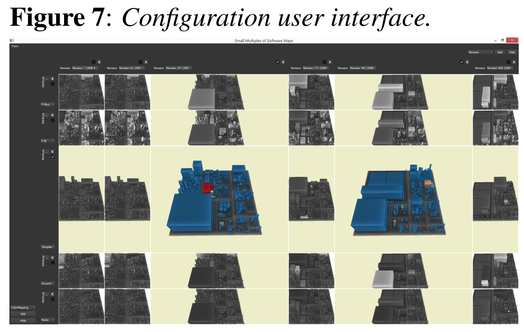

Figure 8: *Focus+context.*

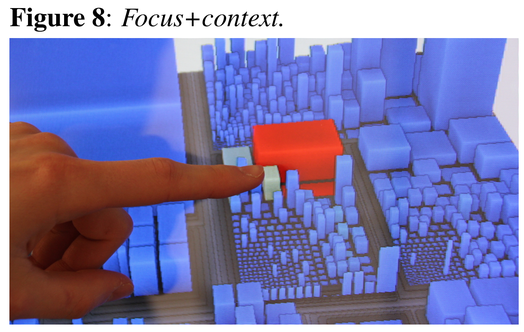

Figure 9: *Touch navigation.*

Willy Scheibel, M.Sc.
willy.scheibel@hpi.de
+49(0)331 5509 3914

Computer Graphics Systems Group
Hasso Plattner Institute

Prof. Dr. Jürgen Döllner
office-doellner@hpi.de
www.hpi3d.de

www.hpi3d.de

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

seerene™
https://www.seerene.com

4DnD·Vis
SPONSORED BY THE
Federal Ministry of Education and Research