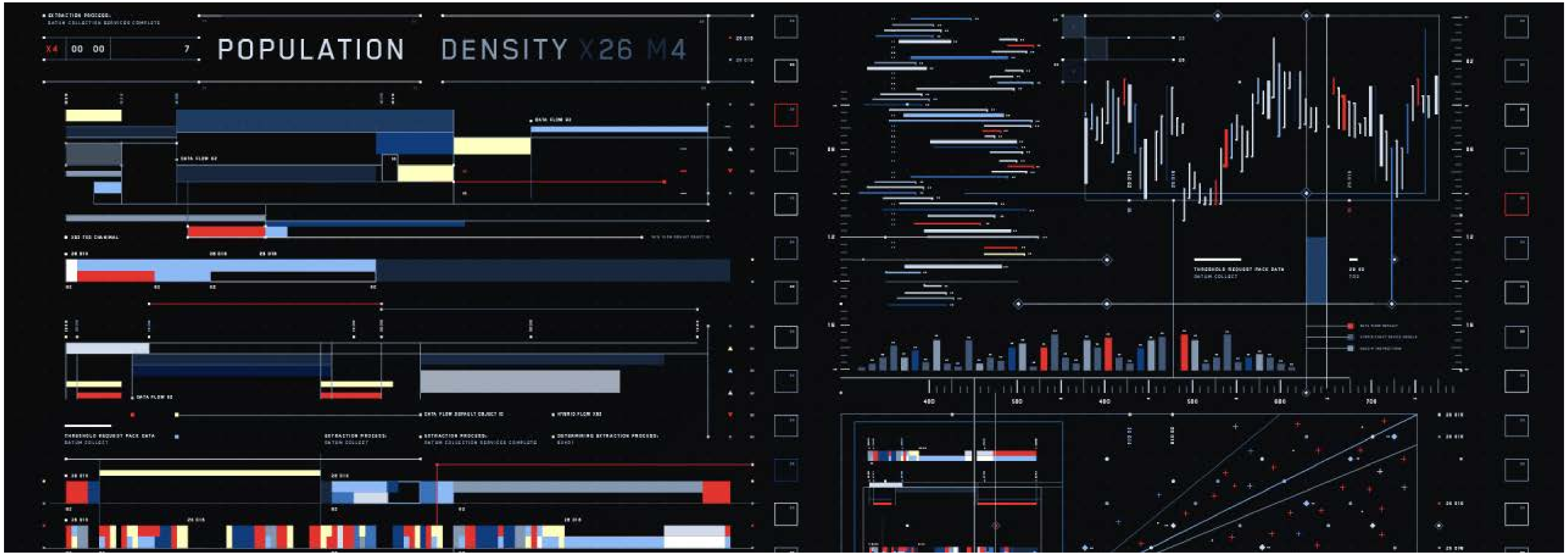
Methods and Techniques of Information Visualization

MA Seminar, WiSe 2016/2017, Computer Graphics Systems Group



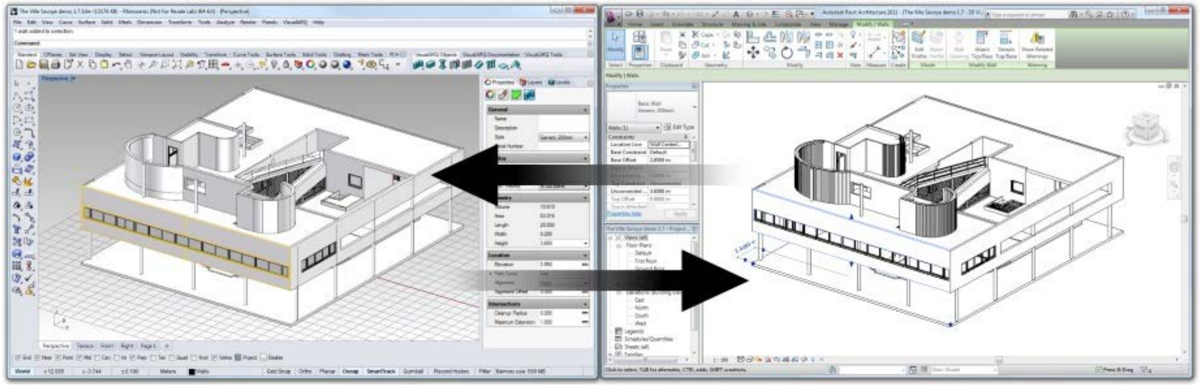
Tutors: Dr. Benjamin **Hagedorn**, Jan **Klimke**, Daniel **Limberger**, Willy **Scheibel**

Instructor: Prof. Dr. Jürgen **Döllner**

Extraction of Building Mesh Envelopes from IFC BIM Data Files

VS**bimlod**

The Industry Foundation Classes (**IFC**) file format is currently the standard file format for storing multi-layered, multi-dimensional Building Information Modelling (**BIM**) data. In addition to containing a wealth of information about a given building design, an IFC file can also contain the 3D geometric representation of the building model at different Levels of Detail (**LOD**). Extracting this 3D building data can useful for visualizing the interior and exterior elements of a building in a real-time 3D scenario.



Tasks

- Investigate the IFC file format and how it stores 3D geometric data for building models (e.g., BIMServer)
- Investigate and implement a procedure to extract 3D geometric data for buildings from this file format into other 3D file formats (e.g. OBJ, FBX); IFC sample data is available [5]
- Investigate how different building model LOD instances can be extracted
- Investigate how this extracted 3D model can be presented in a simple OpenGL 3D application (using Qt for GUI interface)
- Technologies: C++, OpenGL, Qt

Visualization Technique for "Automated Generation of Explosion Views" of IFC Geometry Data

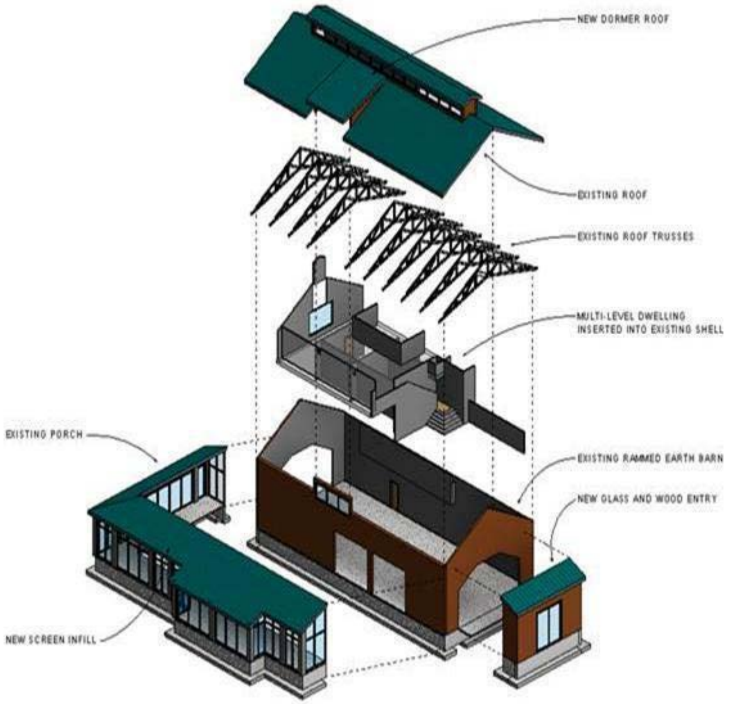
HPI Hasso Plattner Institut

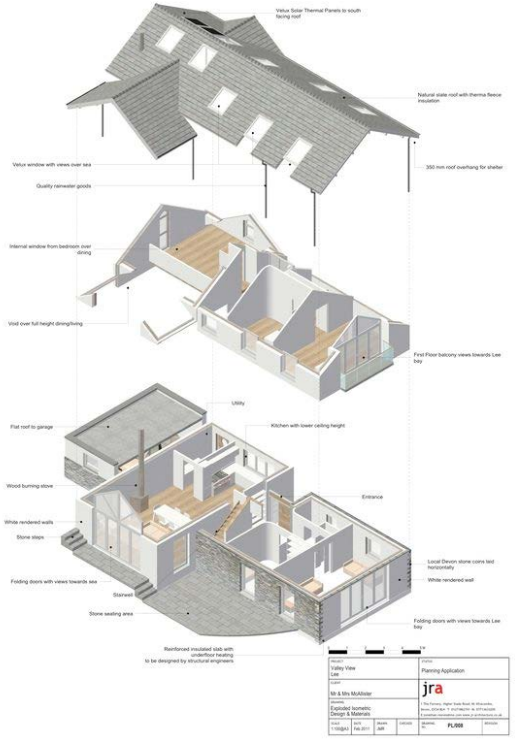
VSexplode

A useful visualization method for examining 3D model data is being able to "explode" the geometry data of a model based on a given set of group attributes. This creates a 3D visualization of all the model component relations and how they fit together. This can be further expanded to include comparison between exploded 3D model faces corresponding point cloud data groups.

Tasks

- Investigate and implement a method for exploding extracted IFC geometry data and projecting within a selected projection the relation of all the model components. [6]
- Investigate and implement an optimal non-standard projection method for displaying the exploded view – e.g. Orthographic / 2D planar
- Investigate and implement a method for labeling and/or visualizing connections of the exploded model parts in relation to the complete geometry model (e.g., line connections between exploded parts and complete model with text annotations extracted from IFC data labels).
- Technologies: C++, OpenGL, Qt





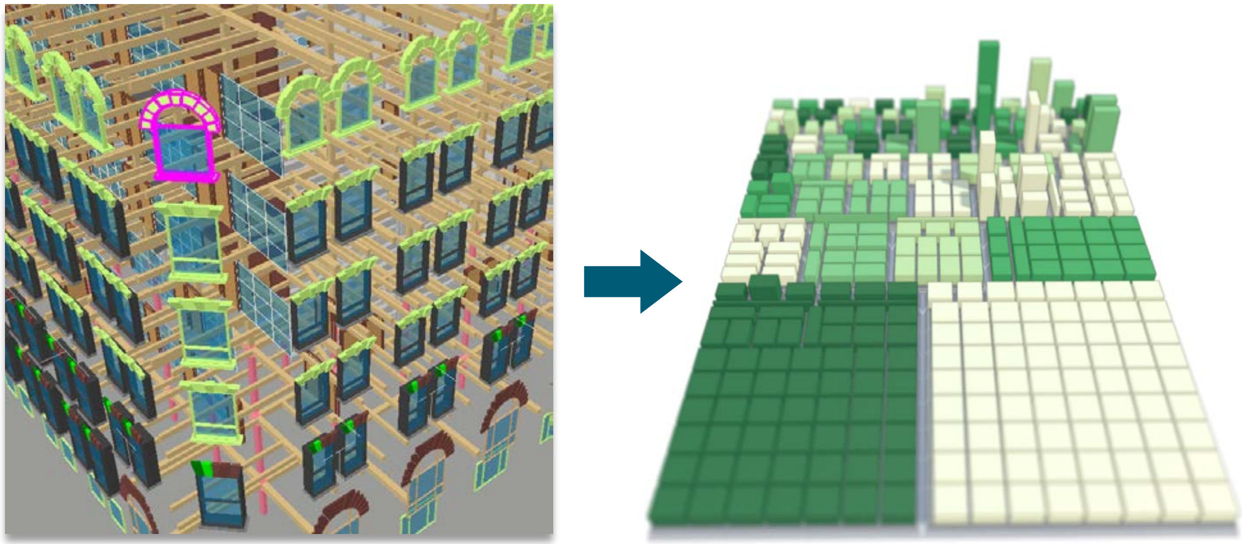
Visualization of BIM Data Through Treemaps

BHifctree

The Industry Foundation Classes (IFC) file format, is currently the standard file format for storing multi-layered, multi-dimensional Building Information Modelling (BIM) data. It contains a wealth of information about a building's components, their properties and relationships. A Treemap visualization approach shall be used to decompose an IFC-based building model and visualize its structure in a systematic way.

Tasks

- Investigate the IFC data model and file format and how to extract building components and their attributes from IFC and store this data in an existing database schema (possibly extent the schema)
- Develop a visualization techniques that transforms IFC data into a hierarchical treemap
  - Implement hierarchy builders that create domain specific hierarchies for BIM Models
  - Implement hierarchical layouting based on the domain-specific hierarchy (space-based, object-type-based)
  - Provide basic interaction tools to explore this data
- Apply this approach to a complex example building (Digital 210 King Street)



3D Symbols in 3D Information Cartography

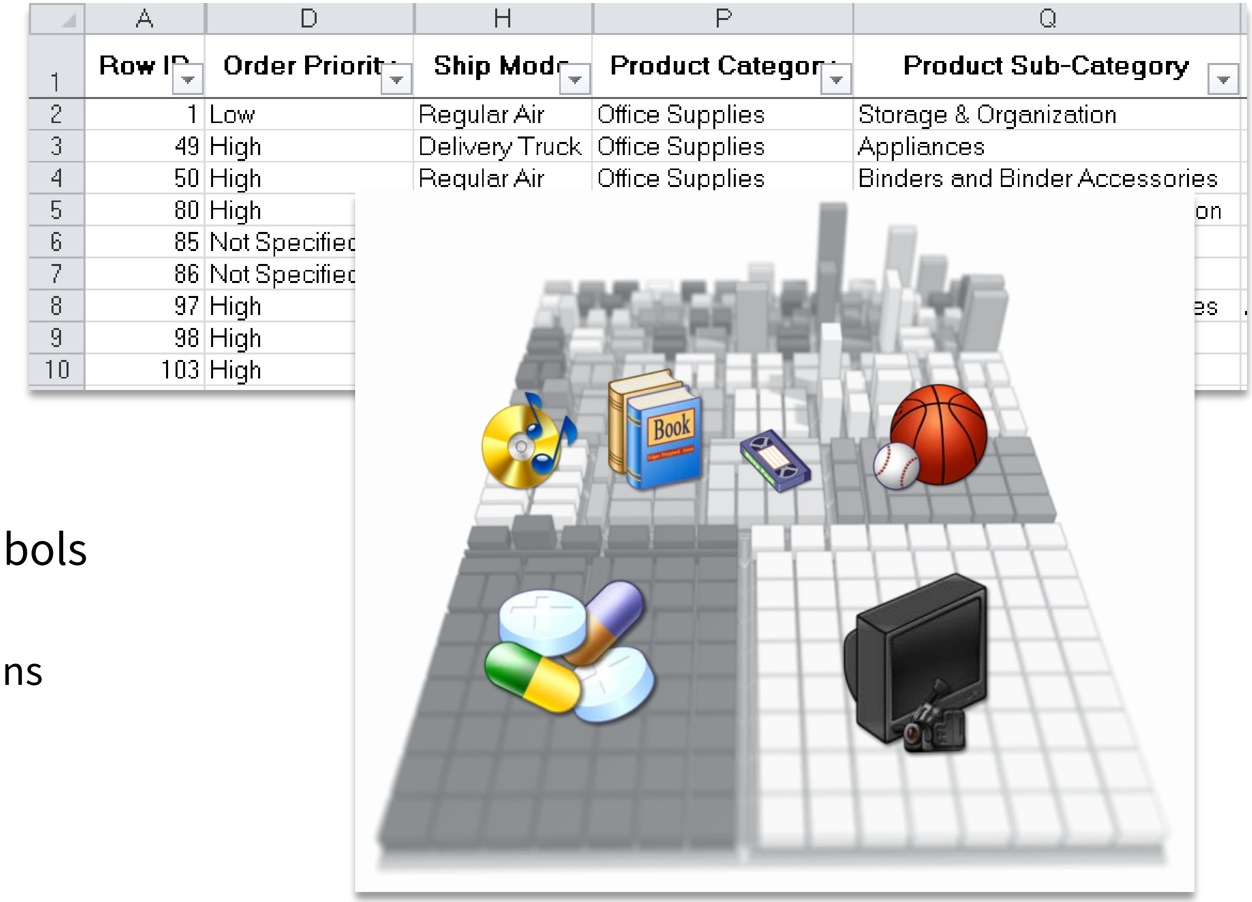
Hasso Plattner Institut

BHsymbols

Information graphics often makes use of symbols can to effectively communicate qualitative and quantitative information. This work aims on applying 3D symbols for 3D information cartography efficiently.

Tasks

- Develop a concept on how to apply 3D symbols to visualize: data items, data quality, data quantity, data aggregation
- Create a 3D symbol catalogue for "Superstore Sales" sample data (8000 orders; 15 data dimensions: ship mode, product category & subcategory, …)
- Extend an existing web-based framework with 3D-symbols
  - Server-side layout computation
  - Efficient transmission of symbol data to client applications
  - Client-side rendering of symbols
  - Mapping attribute data to symbol size, color, texture, …
  - (3D symbol management, animation)

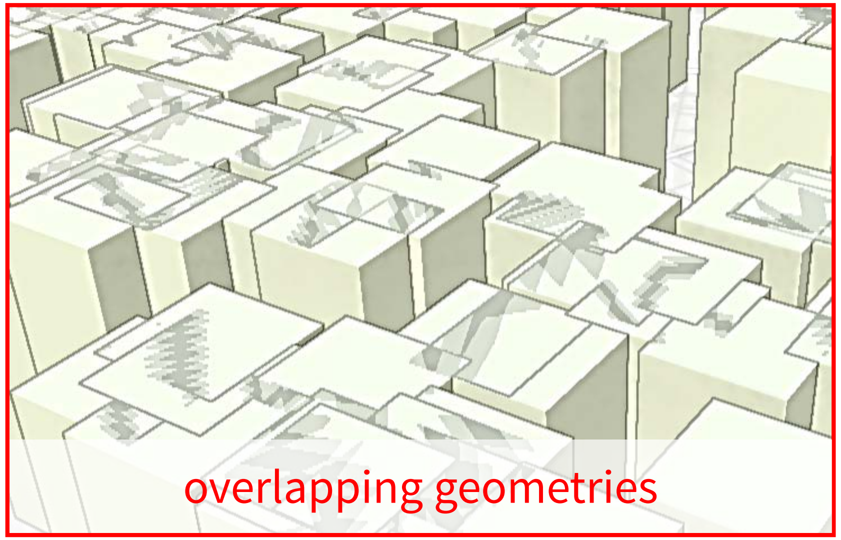


Partially Space-filling Layout for Geo-located Data

HPI Hasso Plattner Institut

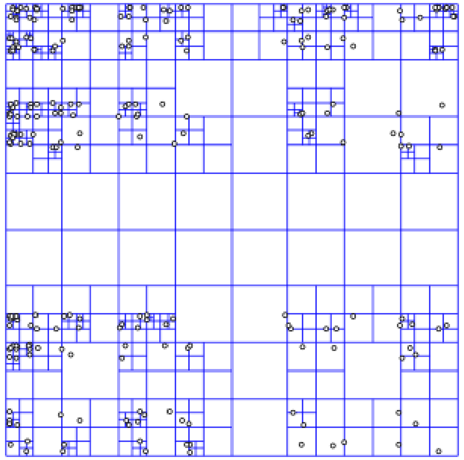
JK**geolayout**

Geolocated data is often spatially-dense, which leads to overlapping geometries when rendering this data. The idea is to develop a layout algorithm that rearranges data items that are close together and preserve location and topology.



Tasks

- Identify relevant layout algorithms (Quadtree, R* tree, Grid Map, Spatially Ordered Treemap, Weighted Map) based on requirements
- Develop a hybrid, (hierarchical) location-aware layout algorithm and integrate it into an existing framework for 3D information cartography [4]
- Demonstrate the layout algorithm with a dense data sample of ImmoScout24 Berlin data samples
- Evaluate performance and limitations of the algorithm



+

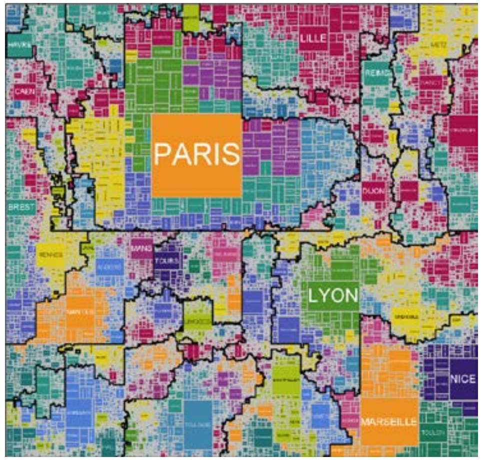
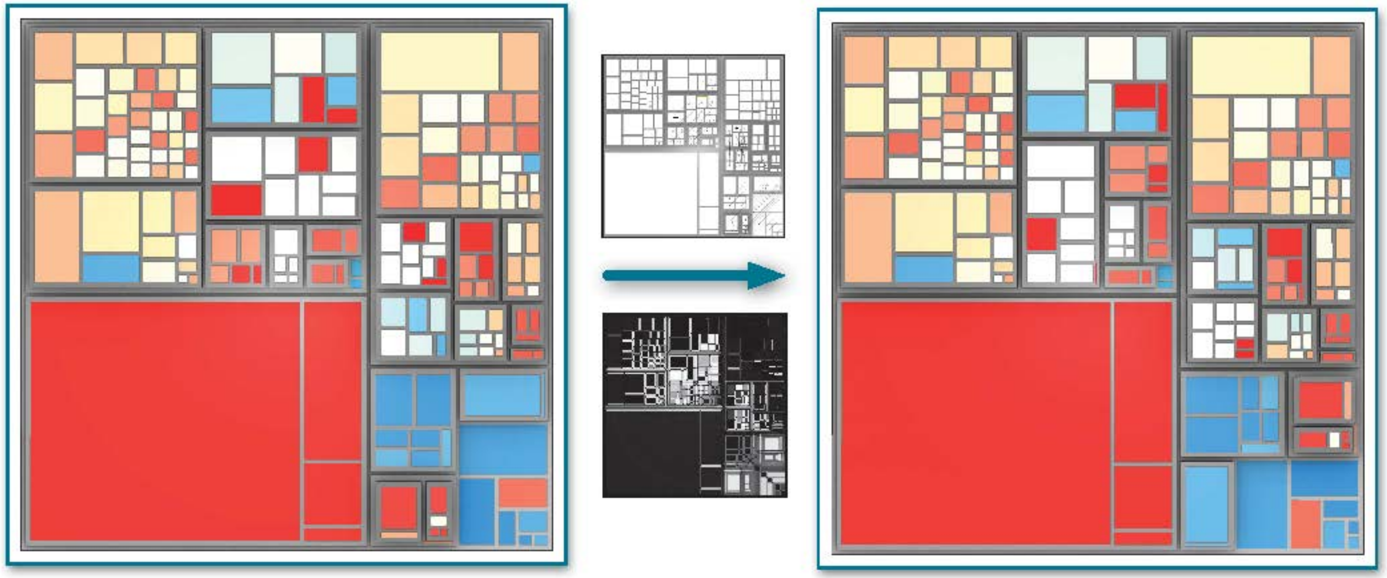


Image-based Comparison of Treemaps

SHcompare

Layout Stability in treemaps can be measured on a visual-item basis using different metrics (e.g., Average Distance Change, Relative Direction Change). Another approach to quantify layout stability is an image-based comparison of treemap depictions.

pHash* is an open-source C++ software library that allows for the computation of such similarity metrics for images with respect to human perception.



Tasks

- Read the related work and understand the basics of *pHash* and layout stability
- Use the *pHash* library to build an application that computes image similarity metrics
- Evaluate the provided features of pHash for the specific use case of treemap depictions
- Compare users perception to *pHash* metrics

* [www.phash.org]

Visualization of Node Links in 2.5D Treemaps

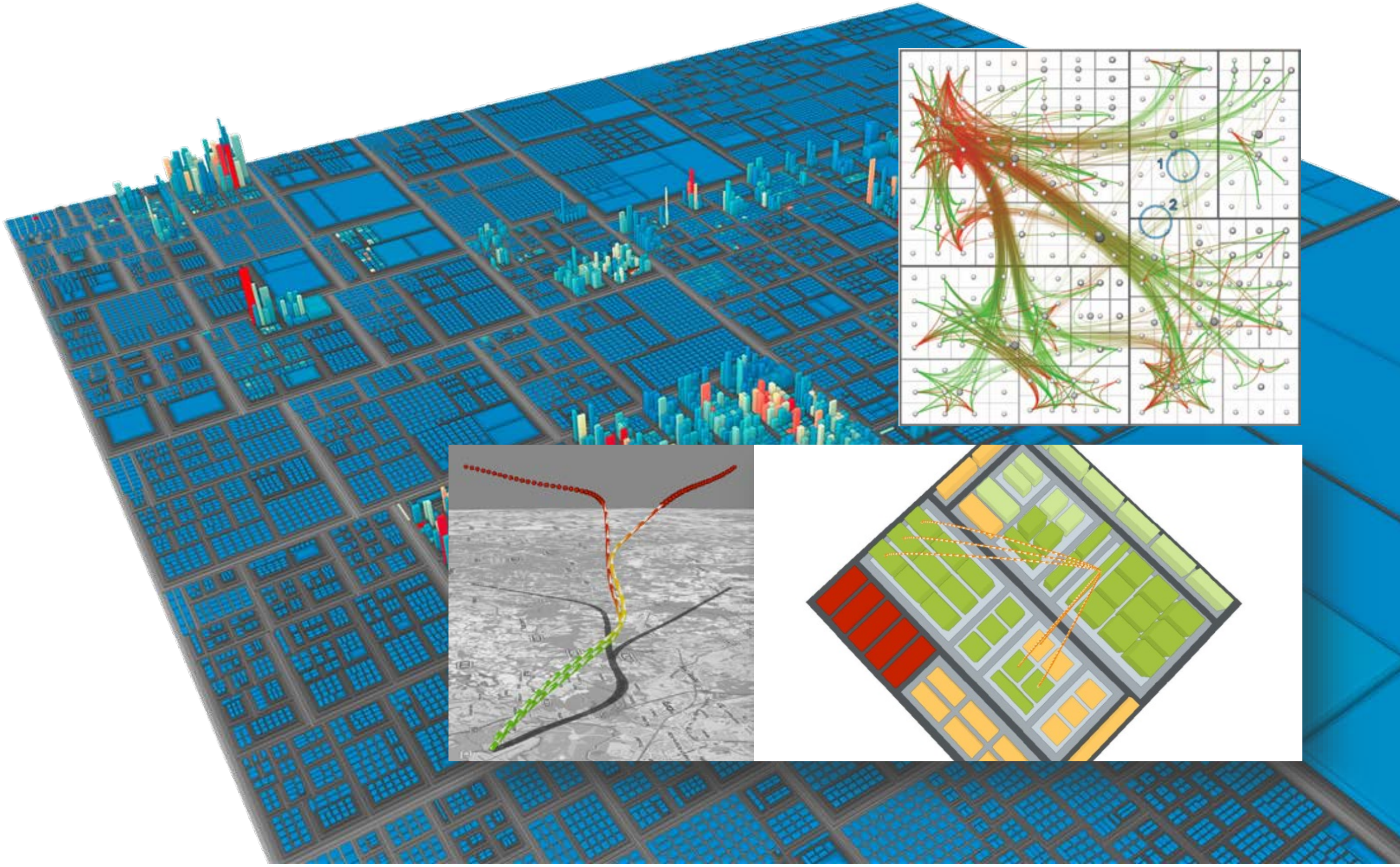
Hasso Plattner Institut

SBlink

Leaf nodes in a treemap can have additional relations orthogonal to the base hierarchy ("secondary links"), e.g., for example includes or references between files. To express these relations, we want to incorporate an interactive visualization technique for these links in the visualization of a 2.5D treemap.

Tasks

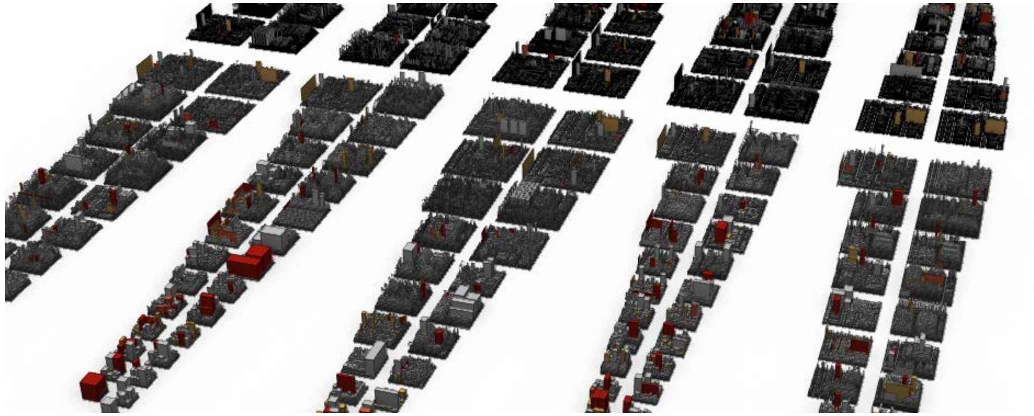
- Implement a parameterizable rendering technique to display 3D tubes that are textured, animated, and shaded
- Integrate this rendering technique into the Arboretum framework
- Implement hierarchical edge bundling using the treemap hierarchy to guide the direction of the tubes
- Propose a suitable parameterization to express direct, successive, and external includes
- Visualize the include relations in a given software system dataset
- Evaluate the scalability of the implementation w.r.t. number of nodes + links

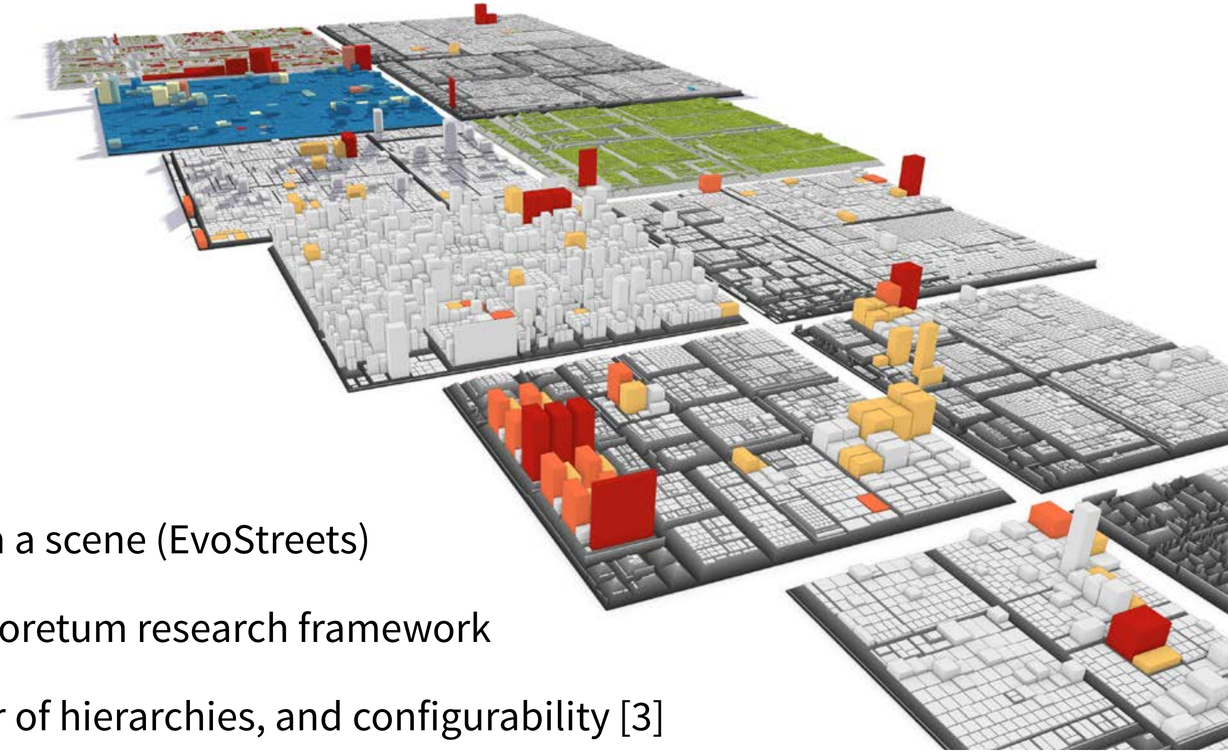


Treemap Landscapes

WSlandscape

2.5D Treemaps visualize one hierarchy at a time or merge many hierarchies into one prior to spatialization.
We want to explore a visualization where multiple hierarchies are represented in a single scene of multiple treemaps.





Tasks

- Implementation of a layouting algorithm for multiple treemaps in a scene (EvoStreets)
- Integration into a treemap landscape prototype based on the Arboretum research framework
- Test the scalability of the visualization w.r.t. visible items, number of hierarchies, and configurability [3]
- Test datasets are a reduced set of the GitHub projects (13) and a full dataset of all GitHub projects parsed by Seerene

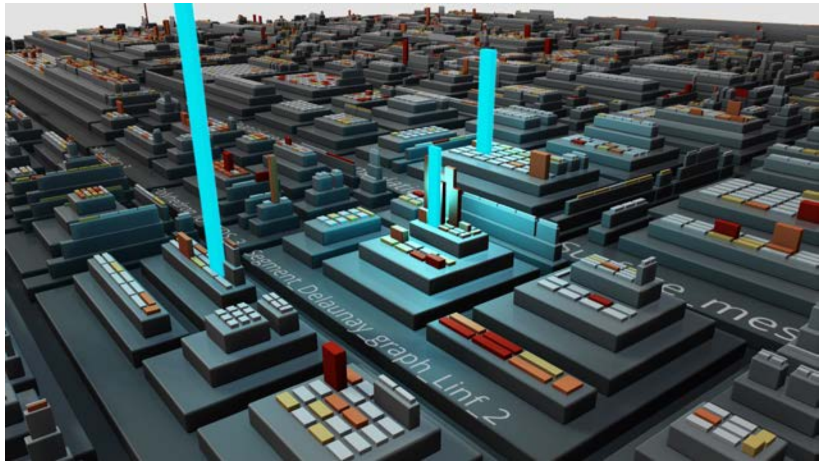
Vulkan-based 2.5D Treemap Visualization

HPI Hasso Plattner Institut

WSvulkan

Current implementations of 2.5D Treemap visualizations often use OpenGL for rendering. The new Vulkan API was built to enable for direct control of GPU memory, computation, and rendering.

A test should show advantages or reveil disadvantages of Vulkan-based implementations.



Vulkan INDUSTRY FORGED

Tasks

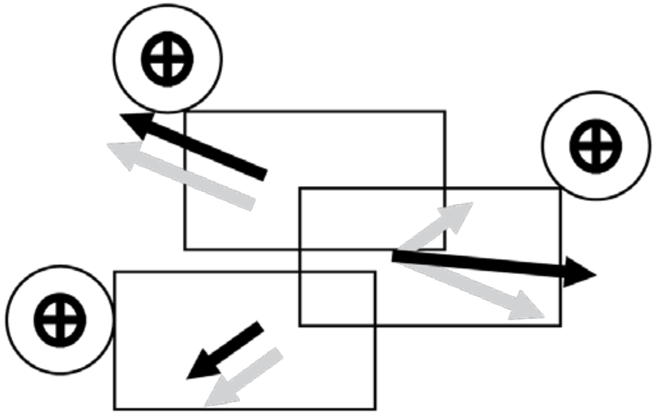
- Implement a Vulkan-based rendering of 2.5D treemaps based on the Arboretum research framework.
- Compare Vulkan and OpenGL implementation w.r.t. approaches, complexity, and performance.

Requirements: Vulkan 1.x and OpenGL 3.2 enabled GPU and driver

Automated 2D Label Placement

DLcartolabel

In this project, various well-known, automated label placement algorithms must be implemented and evaluated.

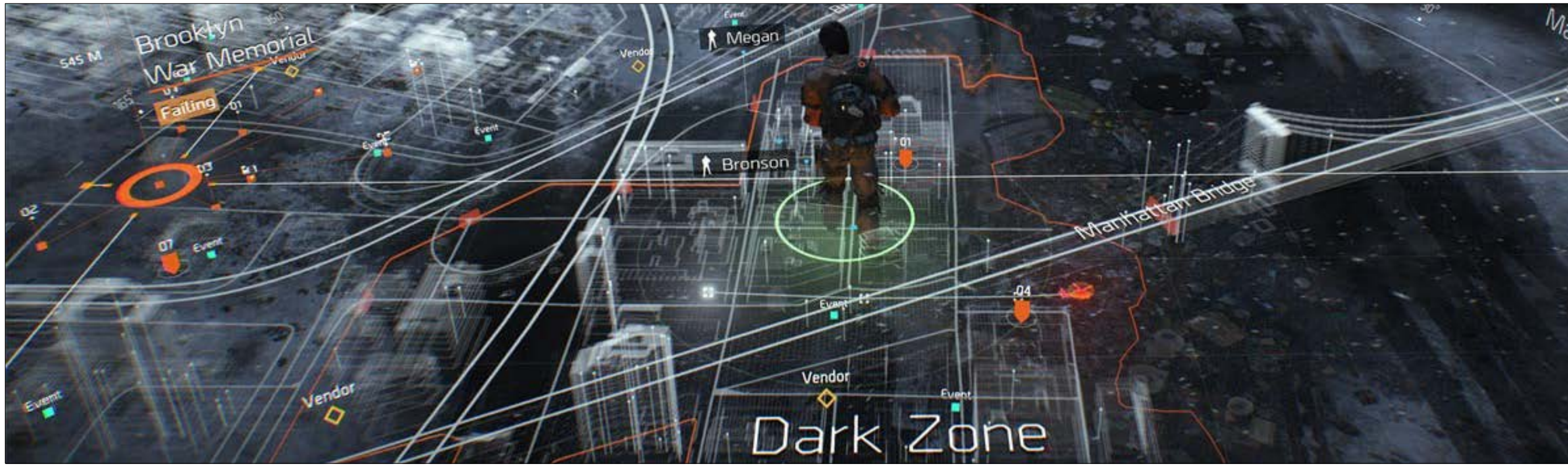


Tasks

- A test that renders at least two different cartographic 2D maps and integrates OpenLL must be setup.
- About four standard algorithms must be implemented, e.g., Random Placement, Greedy Depth-First, Discrete Gradient Descent, Hirsch's Algorithm, Zoraster's Algorithm, Simulated Annealing, …
- For all implemented placement algorithms, the performance must be evaluated. [2]
- An OpenLL API extension should be suggested, that enables configuration of these label placement algorithms

Use Case driven Evaluation of a Real-Time Labeling API

In this project, various scenarios reflecting common labeling use cases will be setup.
The OpenLL C++ OpenGL implementation is then evaluated within these scenarios.



Tasks

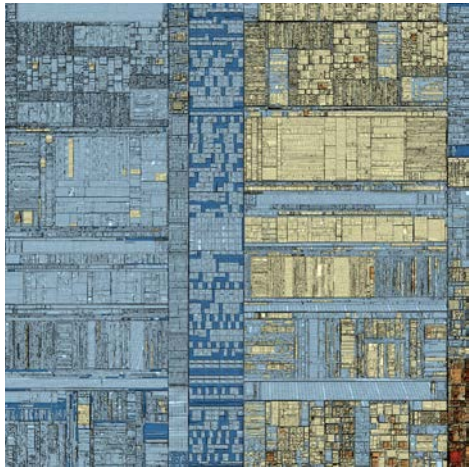
- A test that renders common CG scenes (e.g., Crytek Sponza, Teapot, …) and integrates OpenLL must be setup.
- The following test scenarios must be created:
  - explicit placement for 2d and 3d placement & descriptive placement for automated, adaptive placement
  - caching of glyphs & super sampling of glyph rendering
- For all implemented scenarios, the performance must be evaluated.

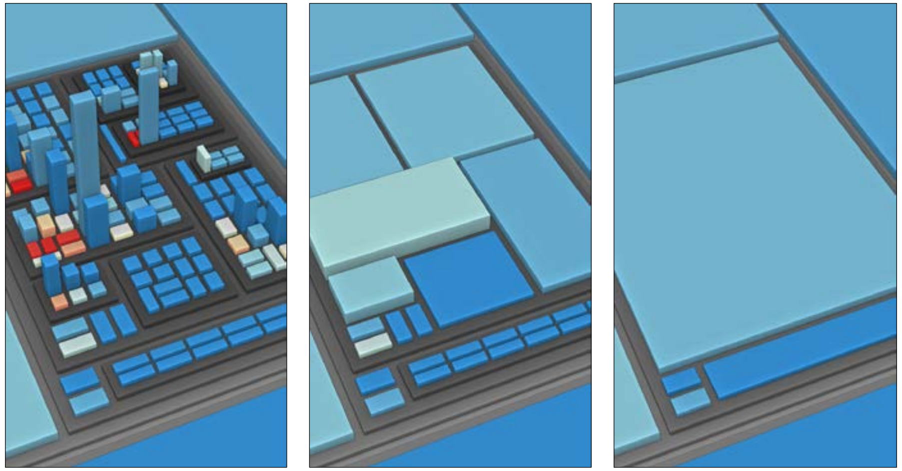
Evaluation of Aggregates in 2.5D Treemaps

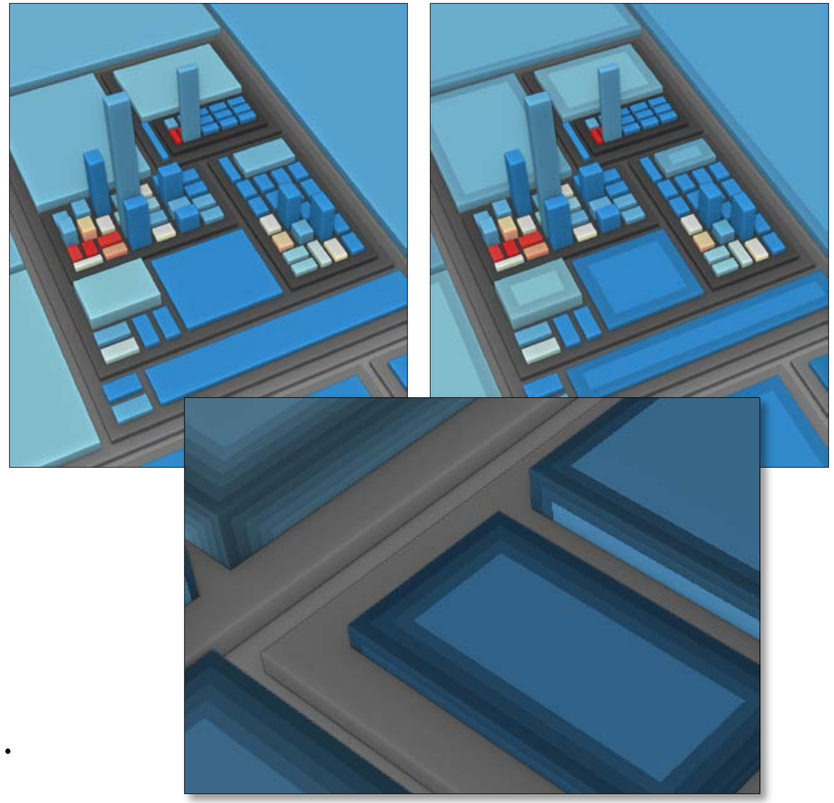
Hasso Plattner Institut

DLevalaggr

In this project accumulated attribute mapping and nested margins must be evaluated in two separate user studies.







Tasks

- The aggregation implementation of arboretum must be setup (merge and adapt).
- A user study that evaluates the performance of nested margins must be conducted.
- A user study that evaluates the performance of accumulated color attribute w.r.t. custom attribute weighting functions must be conducted. [1]

Grading

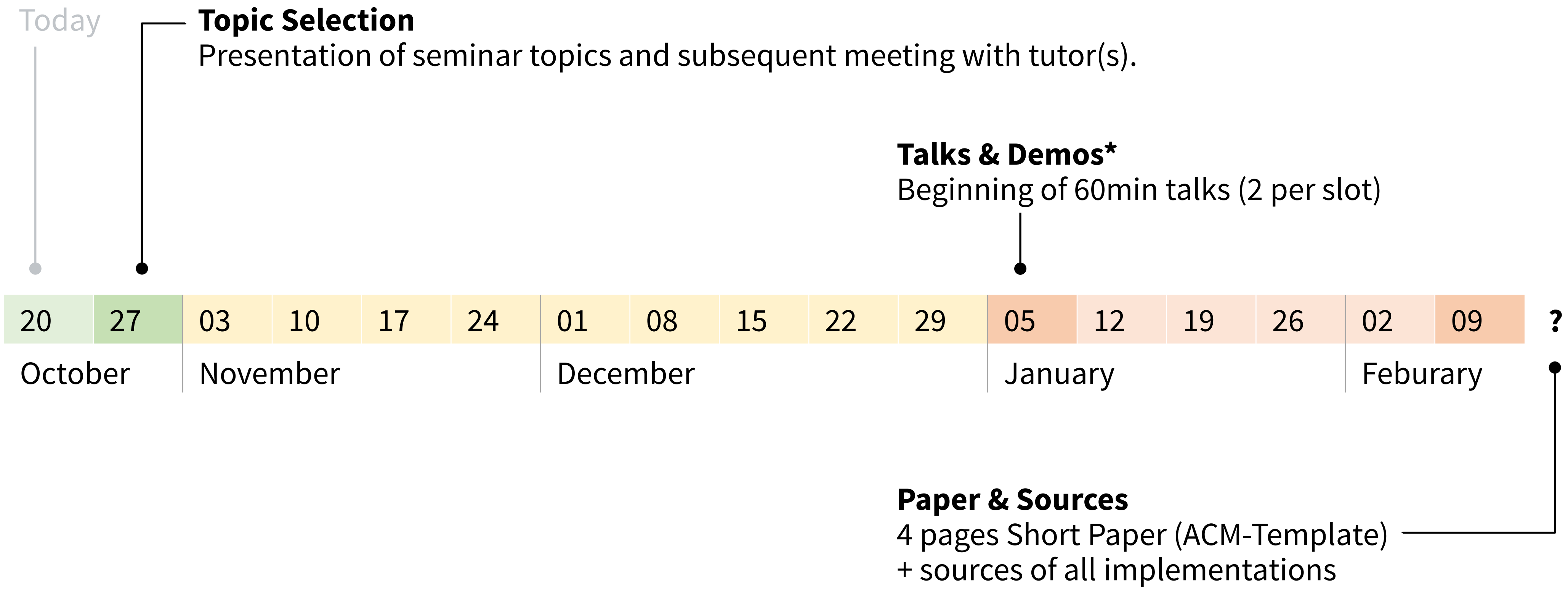
Erfolgreiche Teilnahme erfordert

- konsequente semesterbegleitende Bearbeitung des jeweiligen Themas
- verlässliche Kommunikation mit den Tutoren
- Mitwirkung bei den Präsentationen

Leistungspunkte werden erreicht, wenn die Studierenden

- erfolgreich die zu bearbeitenden Arbeitspakete nachvollziehbar analysieren und konzipieren sowie die wesentlichen Ergebnisse in einem ca. 60-minütigen Vortrag zu präsentieren (1/3);
- erfolgreich eines dem Thema entsprechende Softwareentwicklungsprojekt planen und durchführen (1/3);
- die Ergebnisse erfolgreich in einer schriftlichen Ausarbeitung (ca. 4 Seiten als Short Paper, ACM-Template) zusammenfassen (1/3)

Schedule



* Depending on number of assigned topics and preferred last appointment (2nd or 9th February…)

Resources

https://moodle.hpi3d.de/ – course material, enroll in course using pw: shegalkin

https://gitlab.hpi3d.de/ – tutors will provide you an account and read access to the projects

Use it for everything related to this seminar: source code, slide sets, related work, notes, further reading, paper, etc.

http://jenkins.hpi3d.de/ – bullying authors of non-cross-platform code (no account required :P)

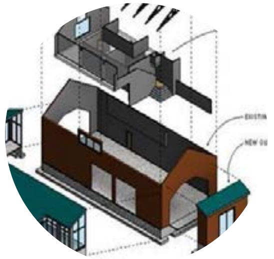
Take Action

Until October 28, 2016

Send an E-Mail to benjamin.hagedorn@hpi.de **AND** daniel.limberger@hpi.de with your choice of at least **TWO** topics.

Feel free to visit/contact us beforehand if you have any questions.

VS**bimlod**

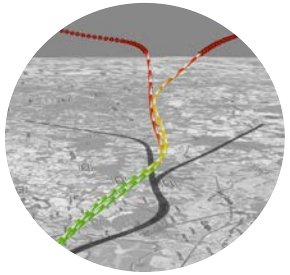


VS**explode**

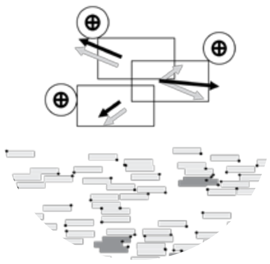
BH**ifctree**

BH**symbols**

JK**geolayout**



SB**link**



DL**cartolabel**

SH**compare**

WS**landscape**

WS**vulkan**

DL**evallabel**

DL**evalaggr**

References

[1] Limberger, D., Tausche, K., Linke, J., and Döllner, J.. Progressive rendering using multi-frame sampling. In *GPU Pro 7*. 2016

[2] Semmo, A., Limberger, D., Kyprianidis, J.E., Döllner, J.. Image Stylization by Oil Paint Filtering Using Color Palettes. In: Proc. CAe. The Eurographics Association; 2015, p. 149–158.

[3] Kyprianidis, J.E., Döllner, J.. Image Abstraction by Structure Adaptive Filtering. In: Proc. EG UK TPCG. The Eurographics Association; 2008, p. 51–58. doi:10.2312/LocalChapterEvents/TPCG/TPCG08/051-058.

[4] Hagedorn, B., Maass, S., and Döllner, J.. Chaining geoinformation services for the visualization and annotation of 3d geovirtual environments. In 4th International Symposium on LBS and Telecartography; 2007

[5] Scheibel, W., Trapp, M., & Döllner, J. Interactive Revision Exploration using Small Multiples of Software Maps. 7th International Conference on Information Visualization Theory and Applications (IVAPP 2016); 2016

[6] Trapp, M., Schmechel, S., and Döllner, J.. Interactive rendering of complex 3d-treemaps. In Proc. of GRAPP 2013.; 2013, p. 165–175.