Informatics

Interactive Visualization of Vector Data on Heightfields

Silvana Zechmeister
Visual Computing

TU Wien Informatics
Institute of Visual Computing & Human Centered Technology
Research Unit of Computer Graphics
Supervisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.-Ing. Dr.techn. Daniel Cornel

Problem & Motivation

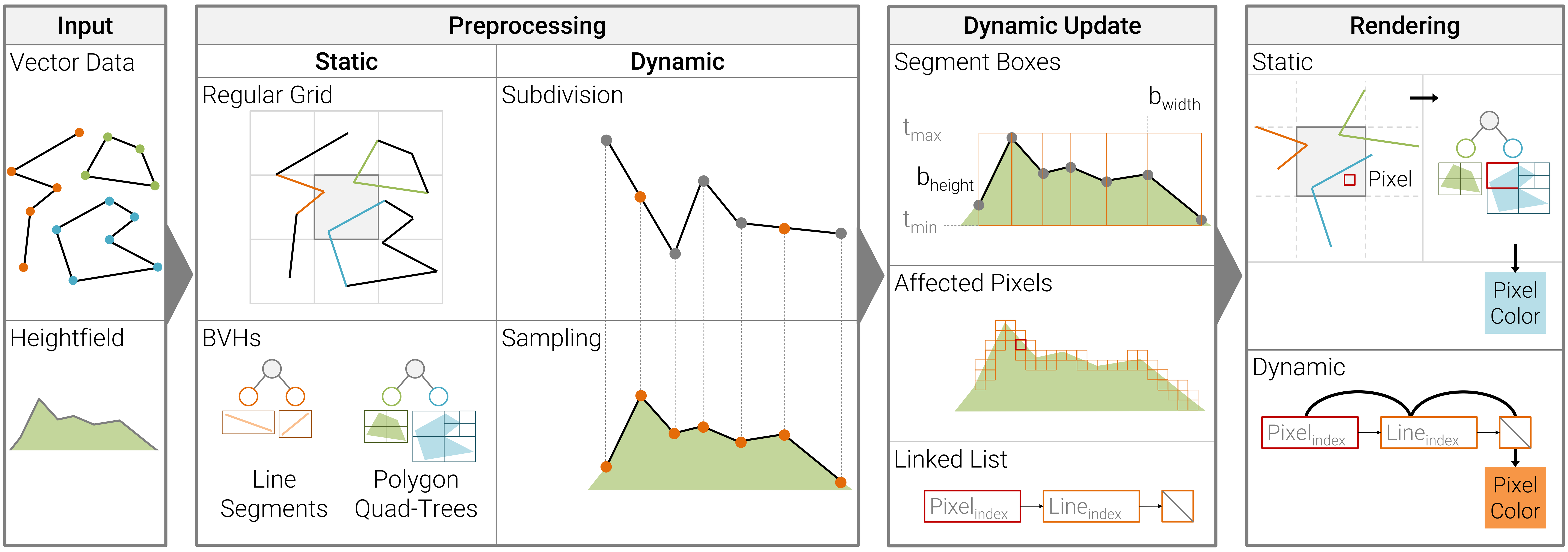
The accurate visualization of huge amounts of georeferenced vector data on heightfields in real-time is a common problem in the field of geographic information systems (GIS). Vector data usually consist of lines and polygons, which represent objects such as roads and parks. The interactive exploration of these vector entities in large-scale virtual 3D environments and the resulting large zoom range pose an additional performance challenge for their visualization. Ensuring clear visibility of all objects of interest in overview and of their details in close-up views is difficult in such large-scale environments.

Contributions

A new screen-based vector data visualization method, which...

- ... is able to render large vector data sets
- ... delivers interactive frame rates
- ... is well suited for large-scale environments
- ... dynamically adapts lines to interactively changing views
- ... supports different polygon and line styles

Visualization Process



The input vector data are lines and polygons defined by **open** and **closed 2D polylines** and a heightfield.

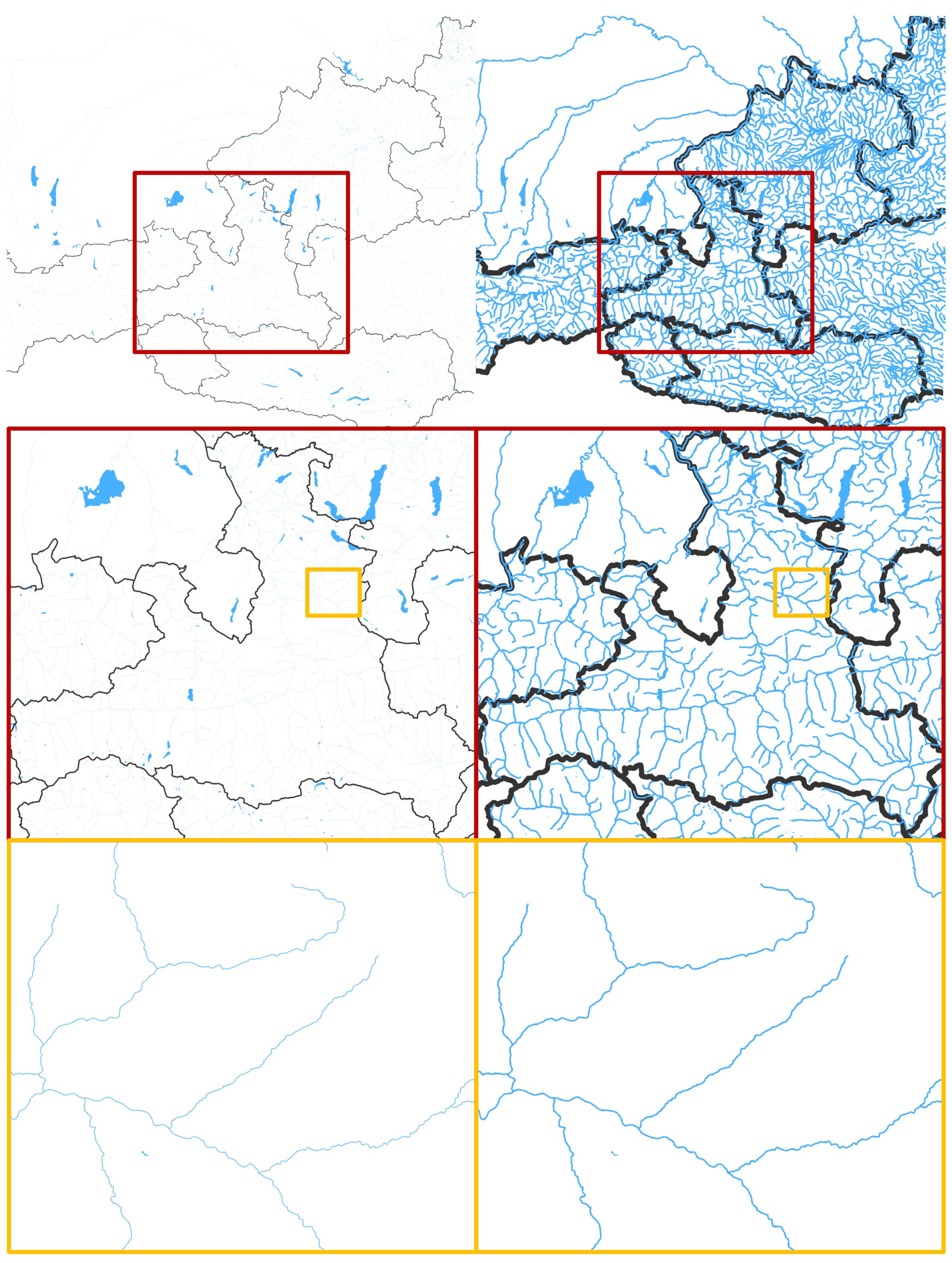
Different **acceleration data structures** are generated during preprocessing for the static and dynamic approach. For **static lines and polygons**, the spatial domain is subdivided by a regular grid. Line segments and polygon quad-trees are assigned to grid cells and organized in bounding volume hierarchies (BVH). **Dynamic lines** are matched to the terrain by subdividing and sampling the line segments on the heightfield to prepare them for dynamic updates.

The line segments from pre-processing are used to create **approximate boxes** to limit the area a line can cover. Accurate **point-in-line tests** determine all lines affecting a pixel, which are stored in linked lists.

A **grid cell**, corresponding **BVH**, and polygon **quad-tree leaf node** is detected by a pixel position to determine the pixel color for the static approach. For dynamic lines, the **per-pixel line list** has to be traversed for the pixel color.

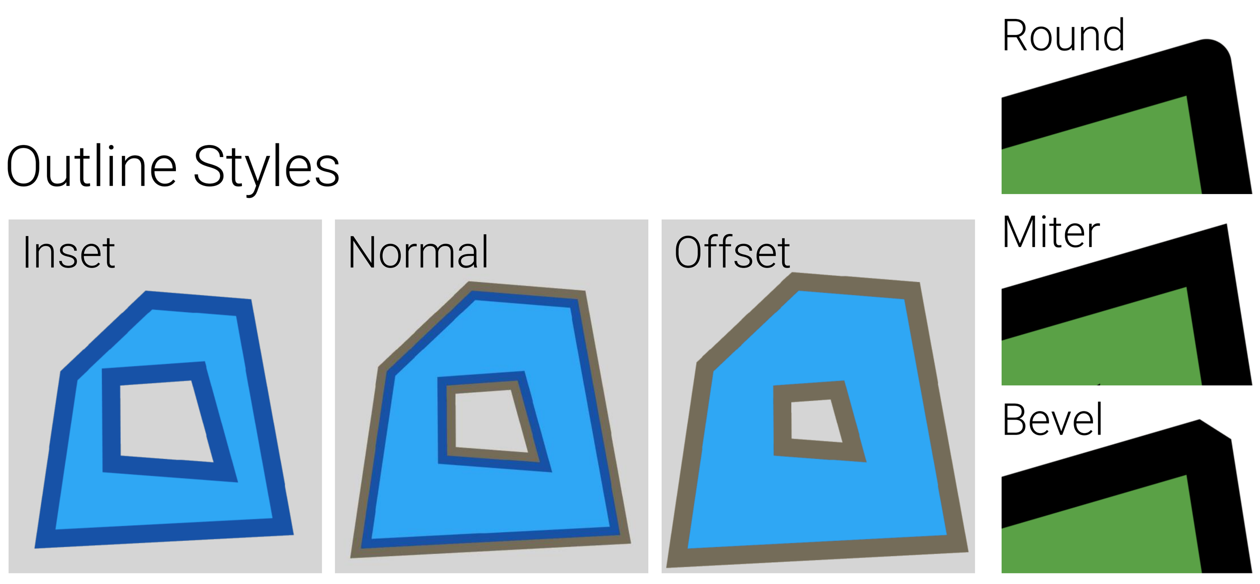
Results

Static Lines — Dynamic Lines



Lines & Polygons with Outlines

Outline Styles



Conclusion

The evaluation results show that both screen-based visualization approaches, static and dynamic, can be applied in real-world use cases of a geospatial information system with large-scale environments and vector data consisting of several millions of vertices and still provide real-time performance.

For large data sets as used in practice, our proposed method significantly outperforms previous approaches.