**Introduction**

In this work we build on top of the recently proposed Street Layout algorithm. This technique can be used for visualizing evolving hierarchical data such as file structures or software systems. Street Layouts represent data as street networks, where each street represents a branch of the hierarchy and buildings around streets represent leaves. We extended the initial idea in various ways to improve stability and compactness in Street Layout visualizations.

**Problem Statement**

The major problem of the Street Layout approach is that the degree of compactness depends on the underlying data. The hierarchy can be unbalanced. As a result, the length of the street in relation to others appears too long, lowering space efficiency and therefore readability significantly.

**Improvements for Visual Stability**

**Local Recalculation of the Layout**

Instead of recalculating the entire layout from scratch, first only that part of the layout where the changes occur is reconfigured. Afterwards, the visualization has to be checked for overlapping regions. If there are any of these regions are recalculated as well.

**Increasing Freespace**

Assigning additional freespace to the elements increases the probability of only having a local recalculation.

**Evaluation and results**

- **Combination of Street Layout and Treemaps**
  - Long streets are substitute with a Treemap representation.

- **Turning the Street**
  - Long streets are subdivided into side streets of the same length which are connected.

- **Turning the Street with Additional Sorting**
  - To increase density when turning the streets, the elements are sorted.

**Example Visualization**

Visualization of the OpenSceneGraph project using the Street Layout. (top), a combination with Treemaps (bottom left) and with additional turning of some streets (bottom right). Color indicates the amount of modification over the recent 500 revisions.