

Continuous and Progressive Point Cloud Rendering Methods

Markus Schütz

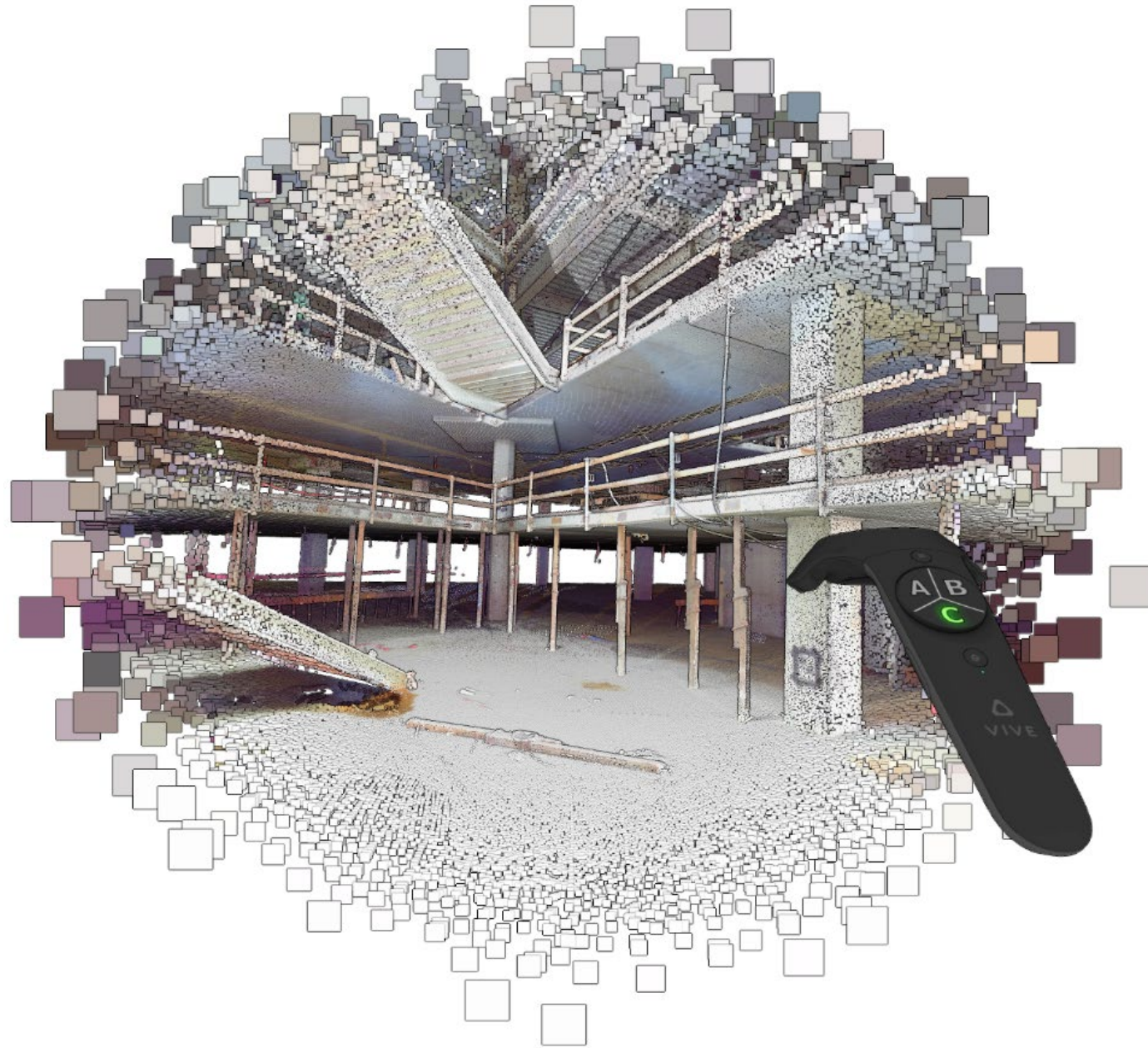
Research Division of Computer Graphics
Institute of Visual Computing & Human-Centered Technology
TU Wien, Austria



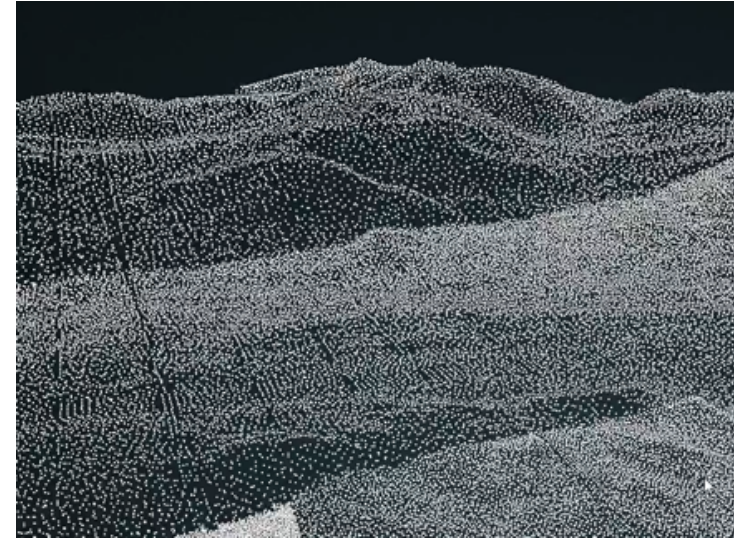
- Continuous Level of Detail
- Progressive Rendering
- New Converter Prototype
- Future Work



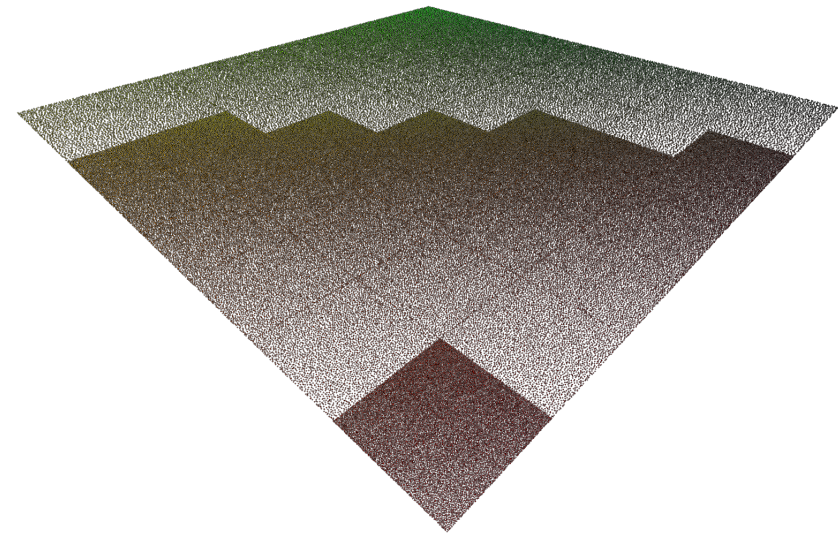
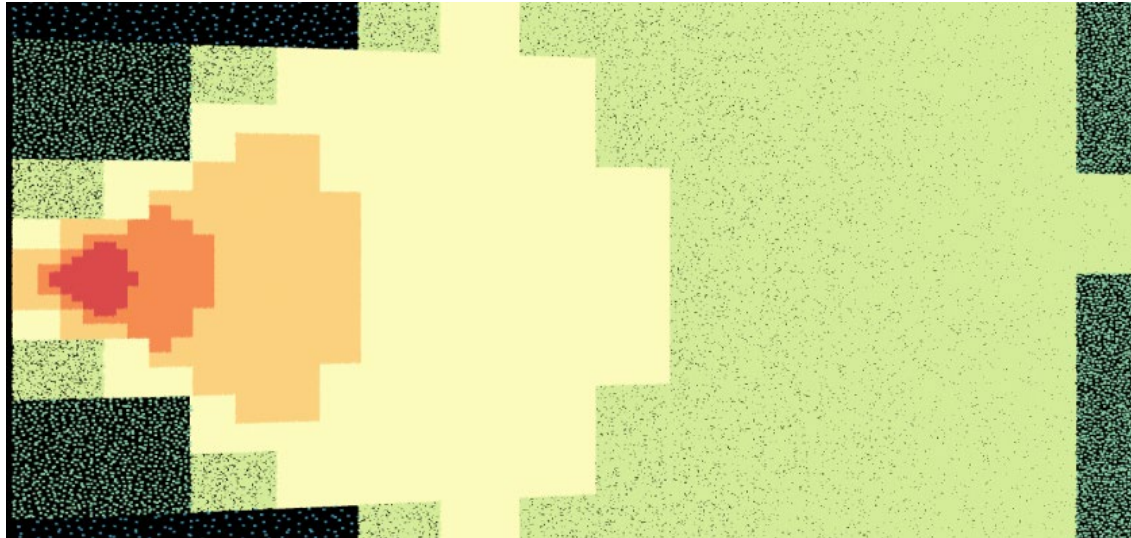
Continuous Level of Detail



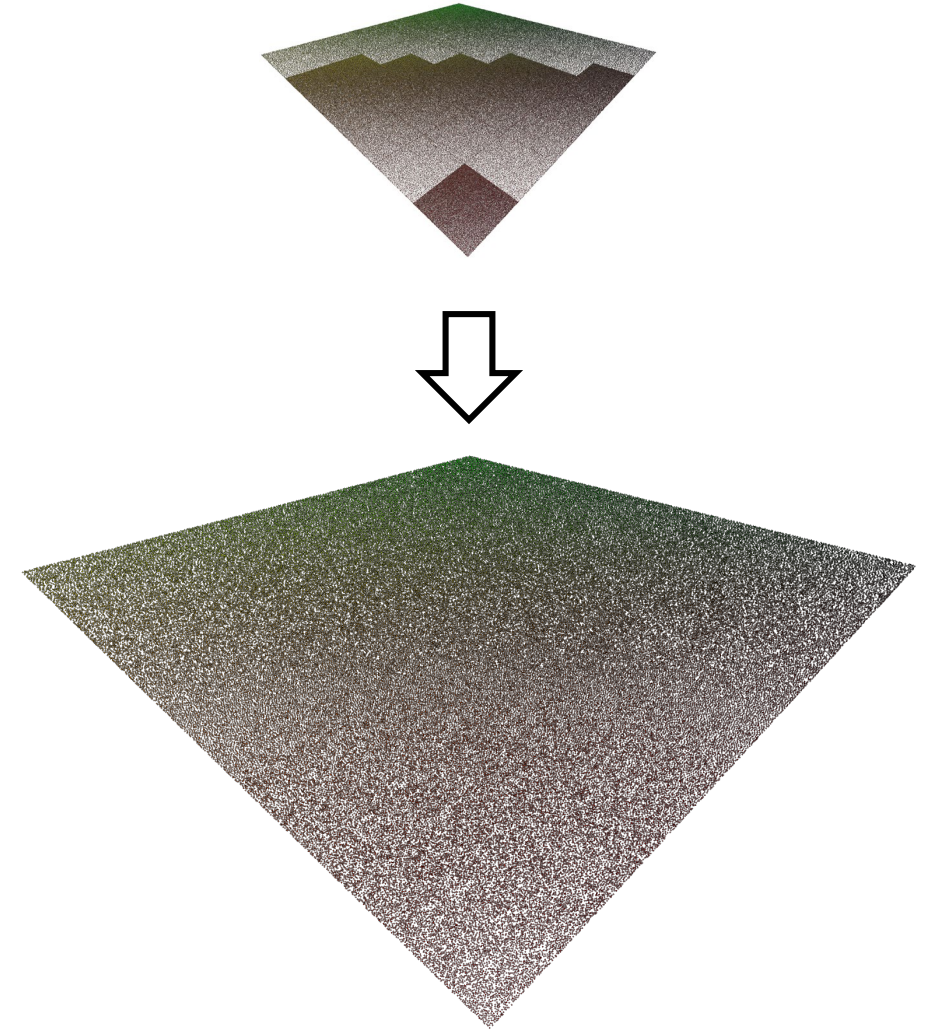
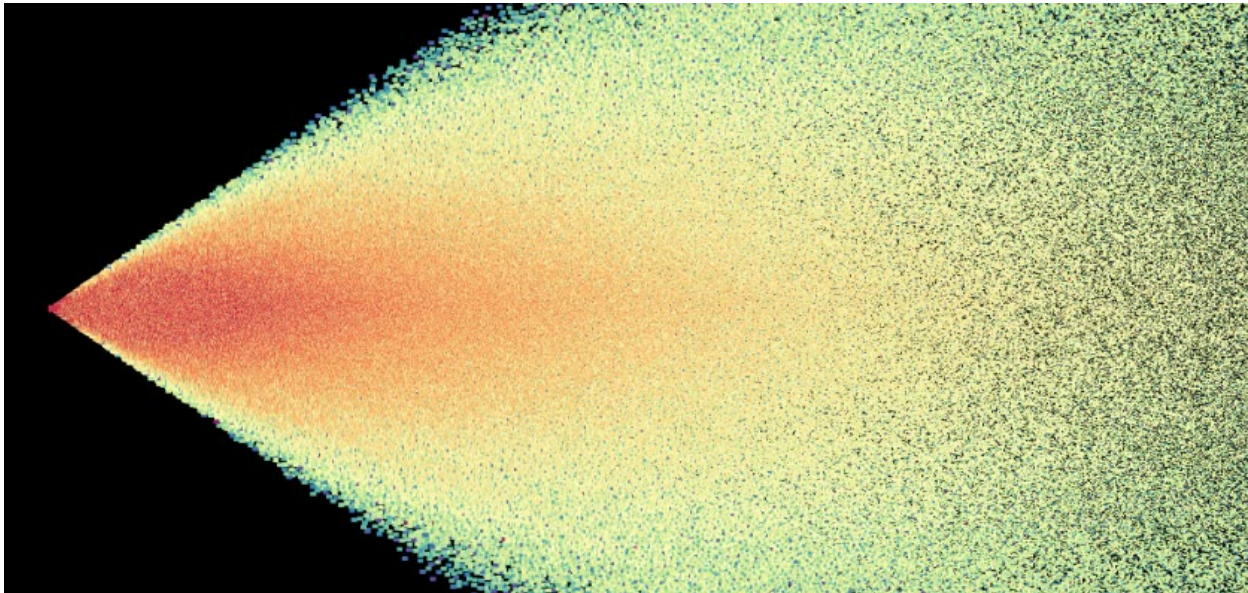
- Discrete LOD
 - Sudden drops in density
 - Popping during motion
 - Especially noticeable at low LOD
- Low LOD in VR due to performance



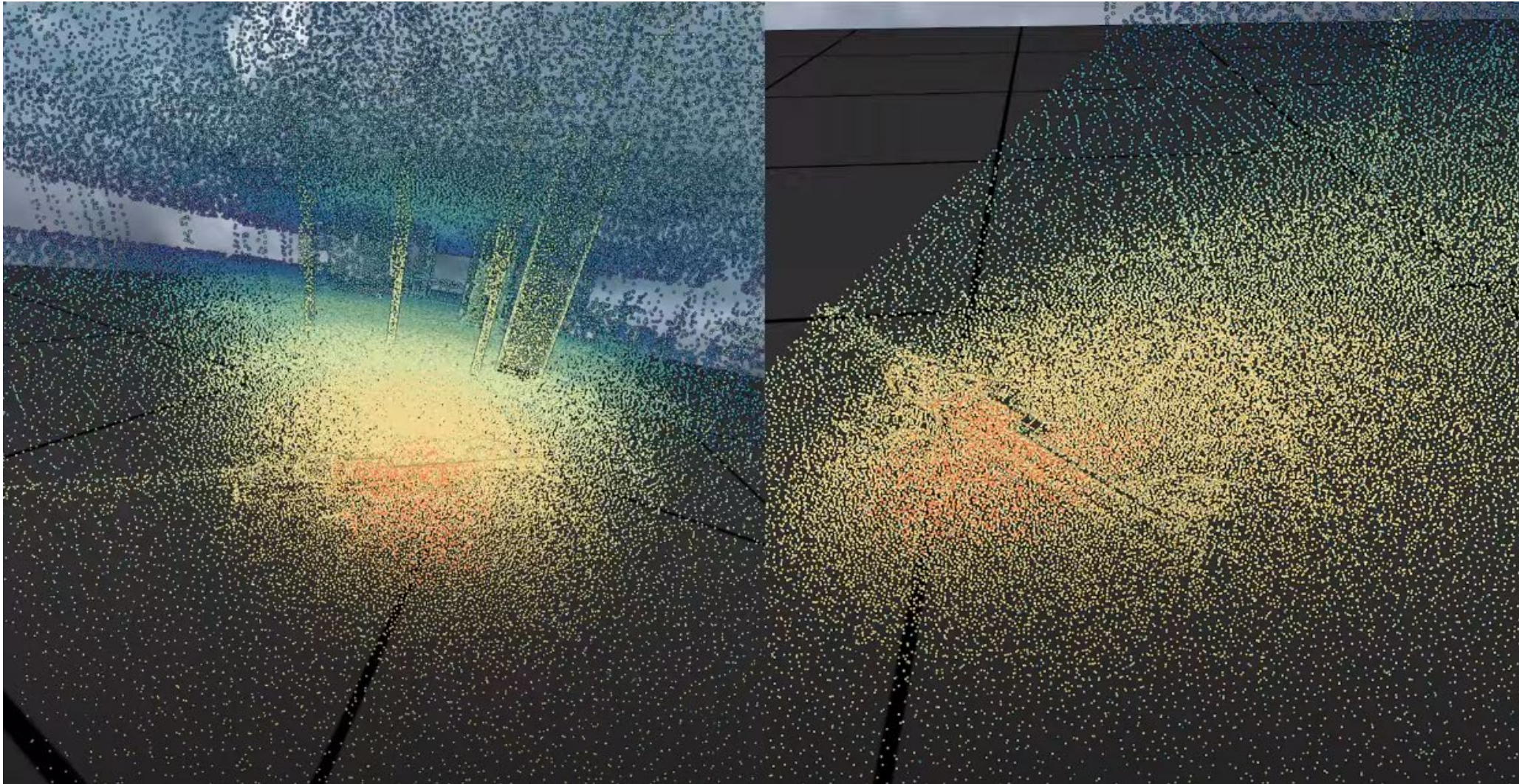
<https://youtu.be/sd-PhImDJic>



- Continuous LOD
 - Continuous transition
 - No visible chunks
 - Subtle point-wise fading



Downsampling



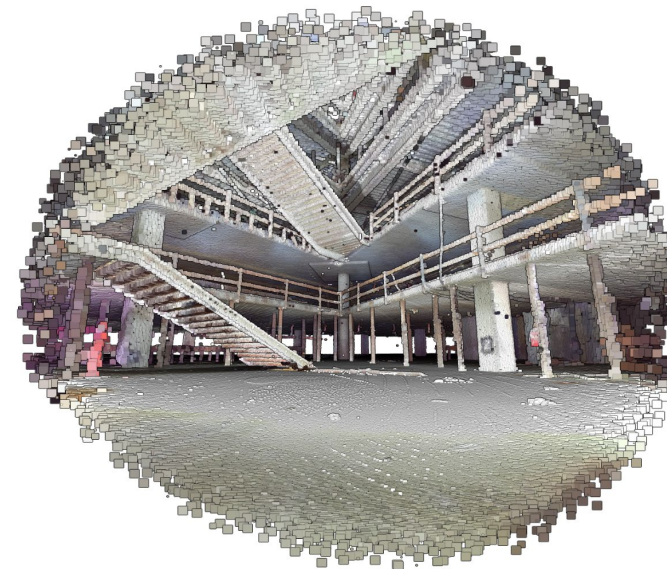
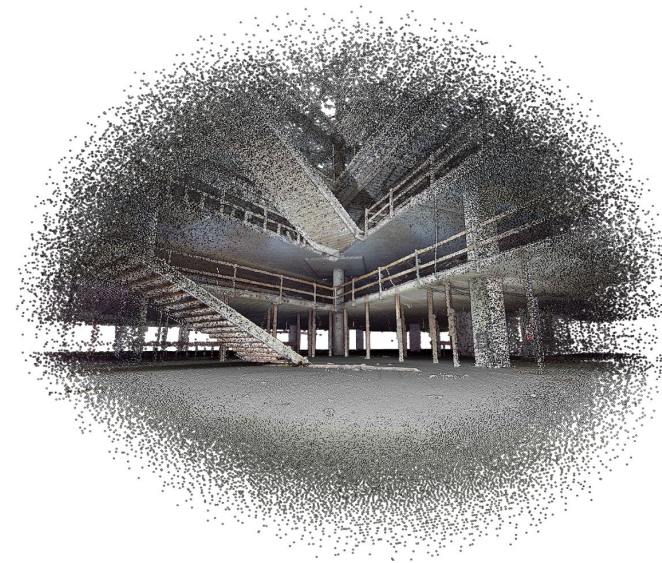
<https://youtu.be/jFIQOda9rxQ>



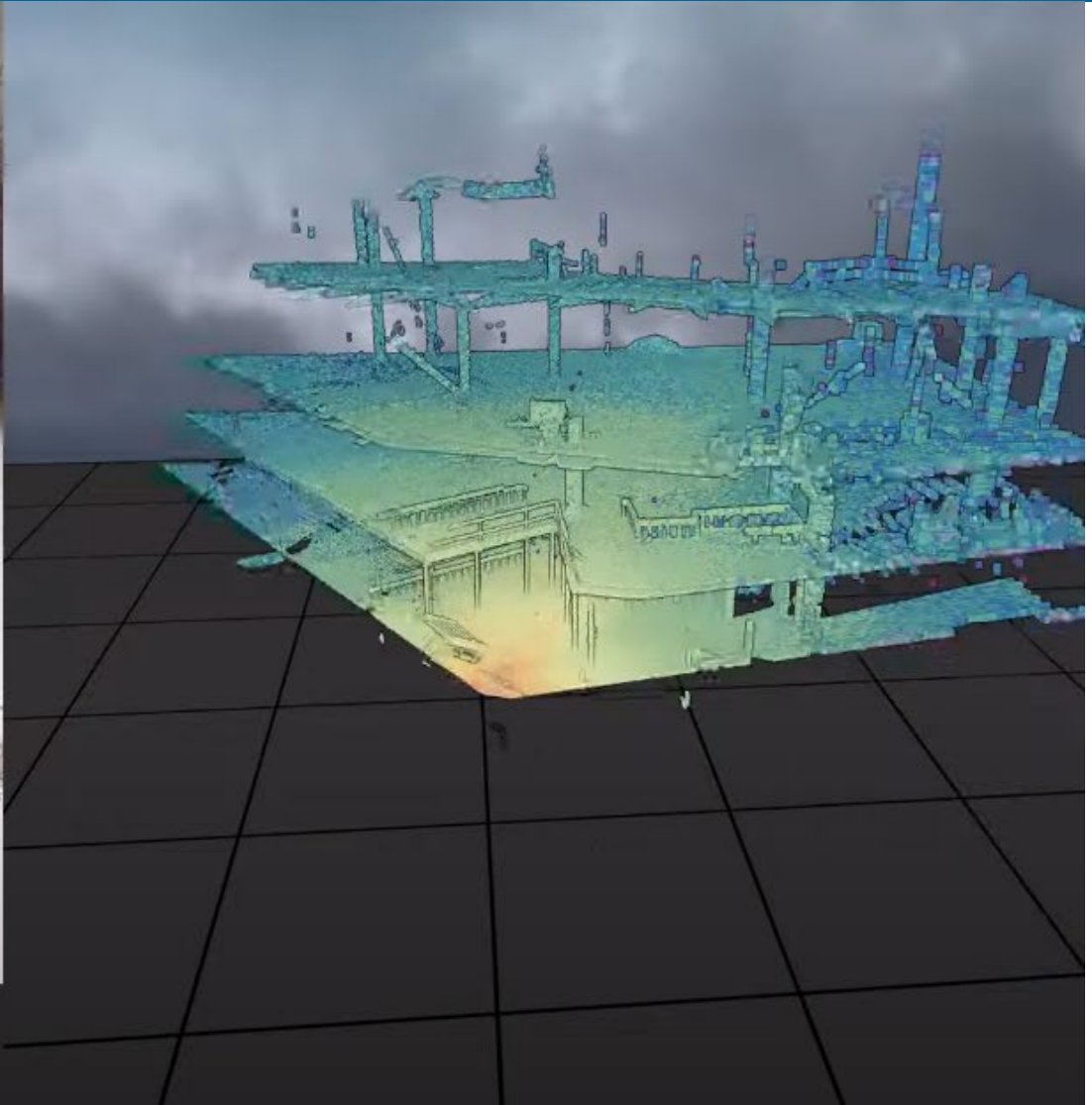
- Target-spacing based on:
 - Distance to viewer
 - Distance to center of screen (VR)
- Point-size = target-spacing
 - But: Fade-In to avoid popping



<https://youtu.be/YygAhAPRXLM>



Results



<https://youtu.be/00mldos05WM>



Progressive Rendering



- Problem: LOD structures take time to generate
- Goal: Render any point cloud that fits in memory in real-time



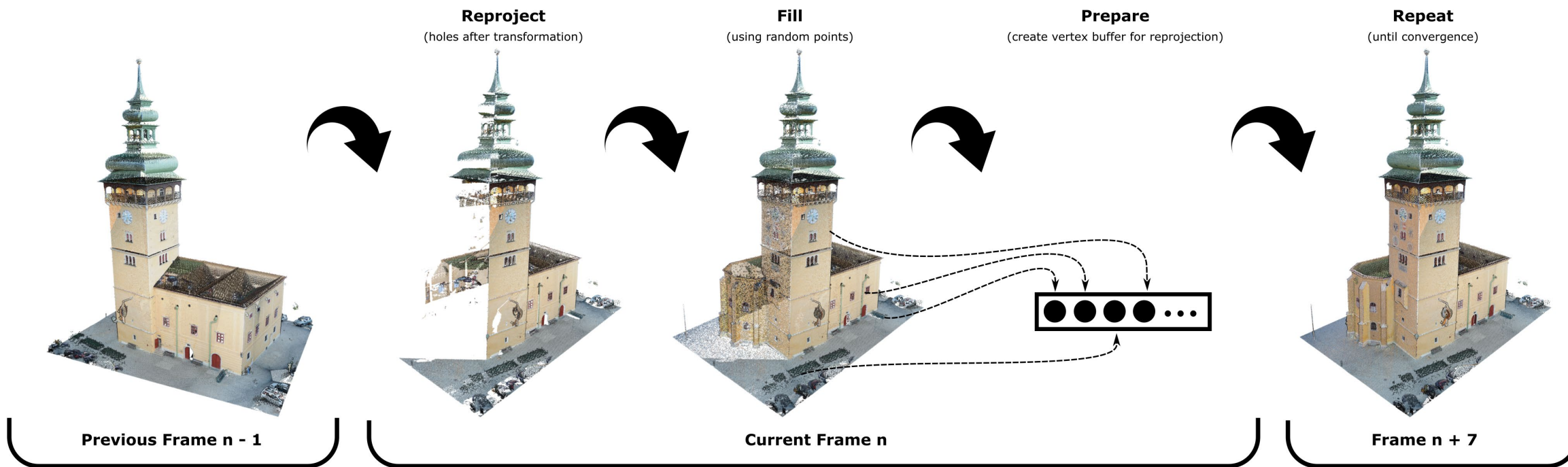
Progressive Rendering



<https://youtu.be/ySDnh0gLqBk>



Progressive Rendering



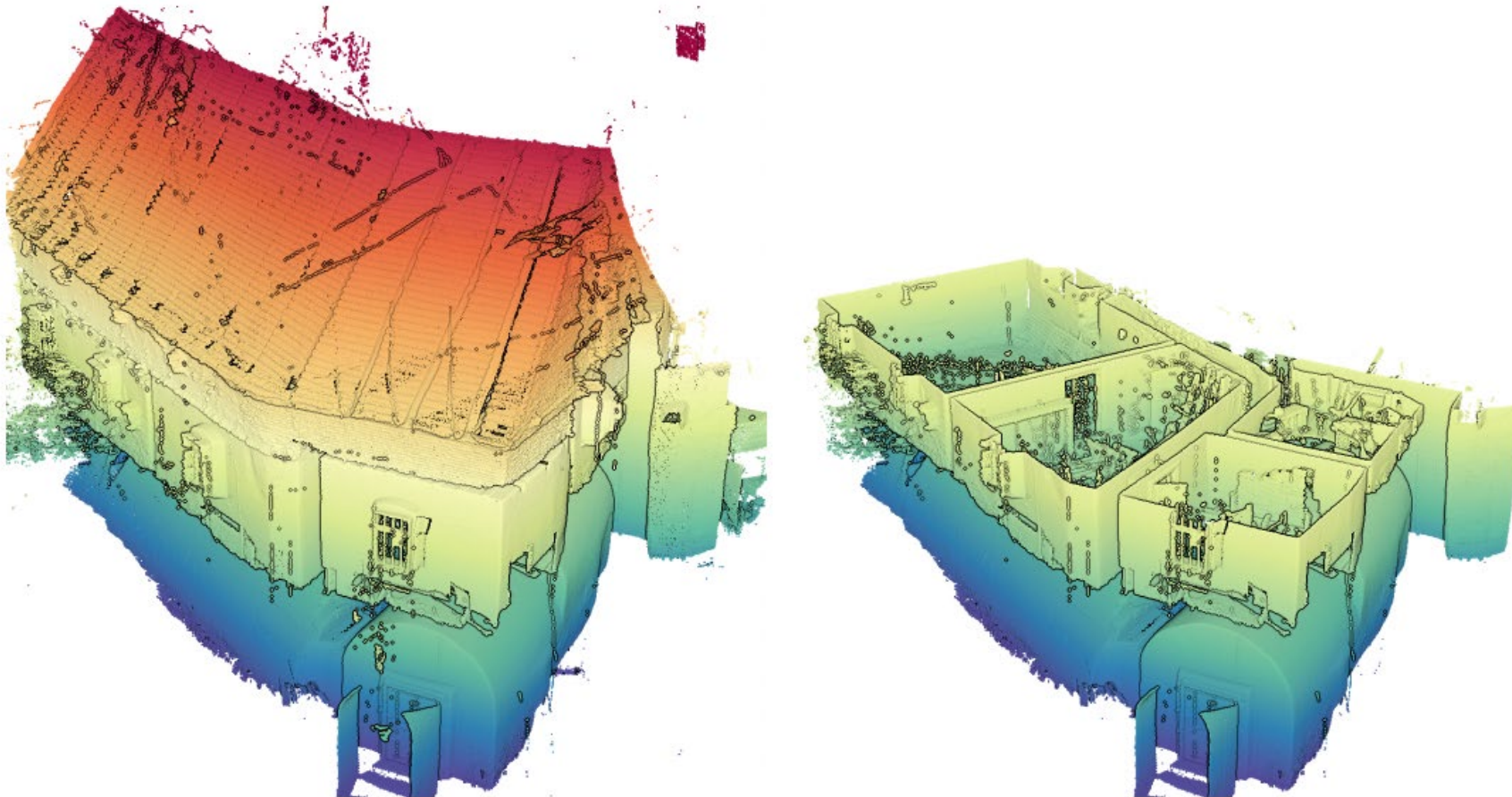
Progressive Rendering



<https://youtu.be/A0Va8zmxB3s>

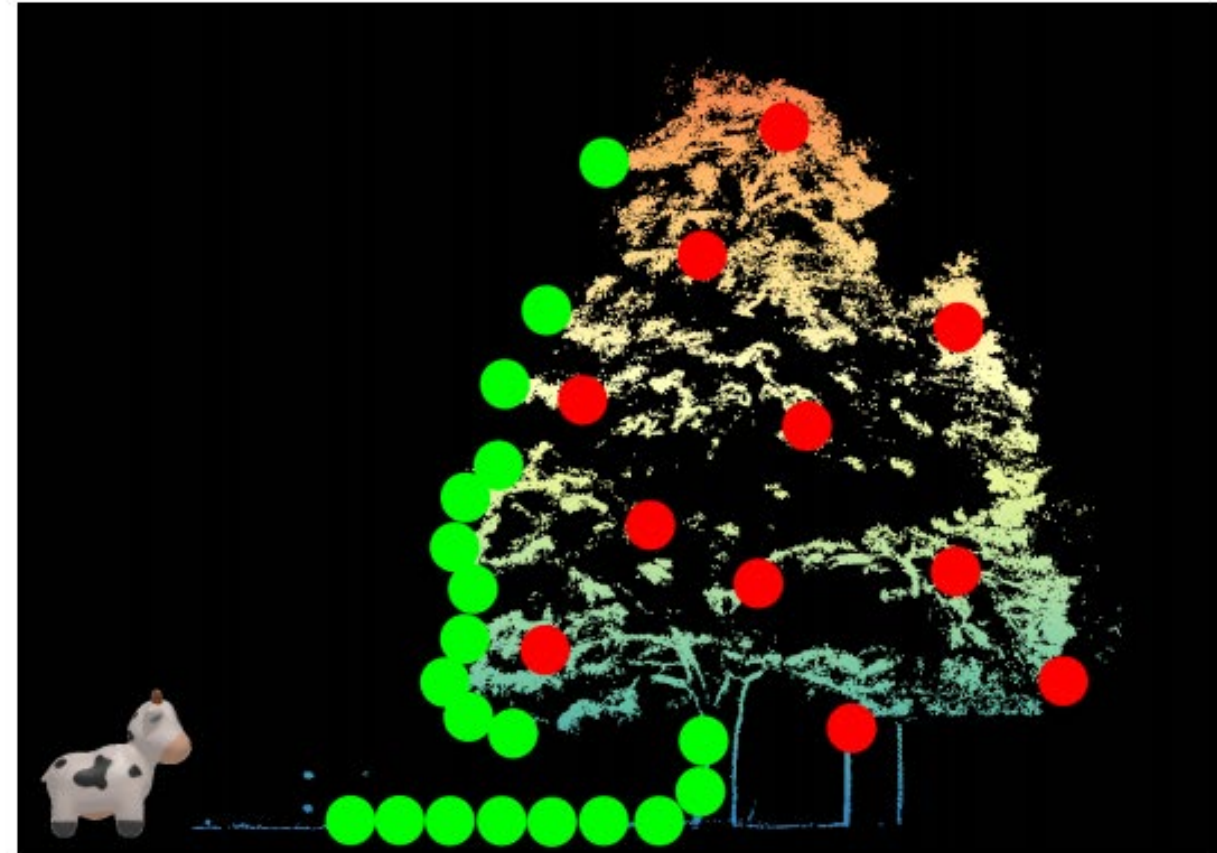
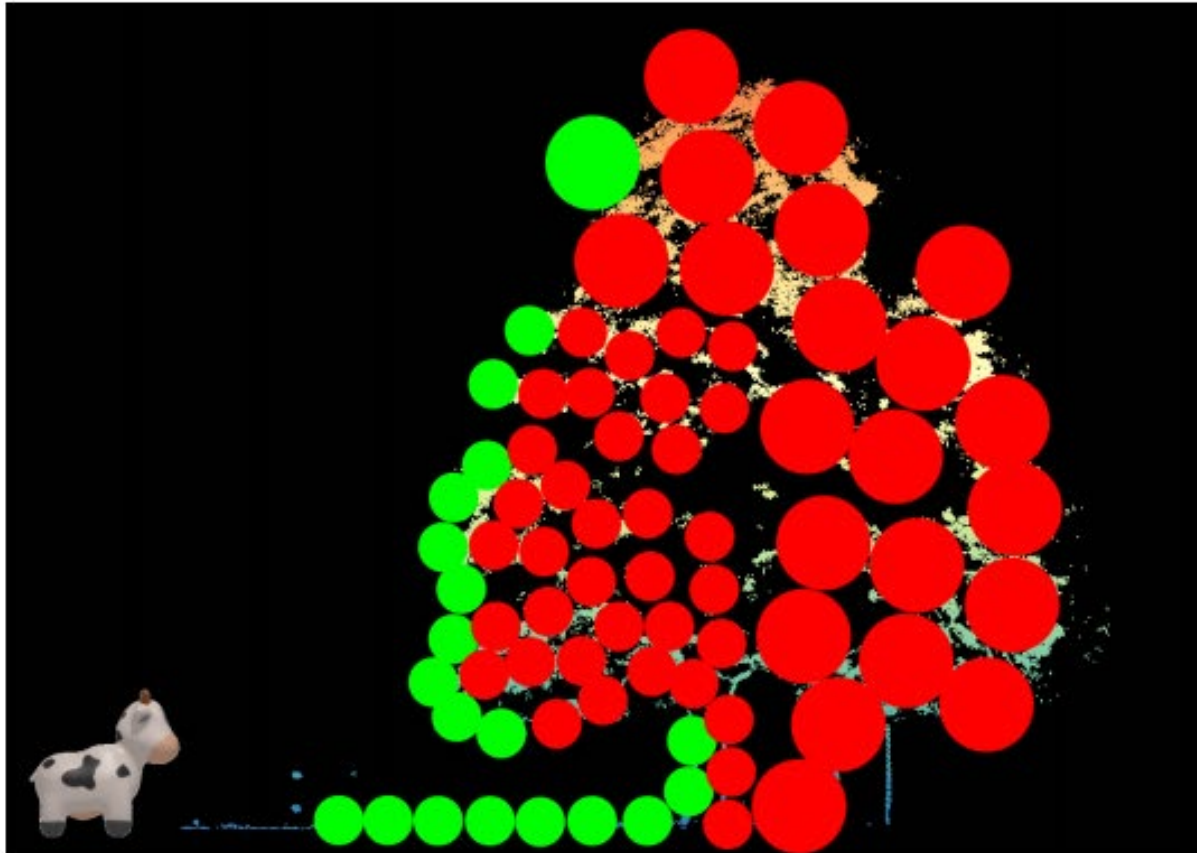


- Depth-Complexity:
The amount of hidden surfaces/layers/points



Depth Complexity

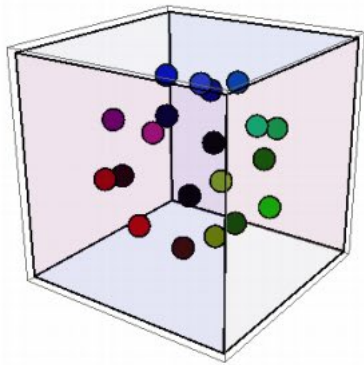
- LOD: Complexity reduces framerates and/or details
- Progressive: Always 60fps & full detail; longer convergence times



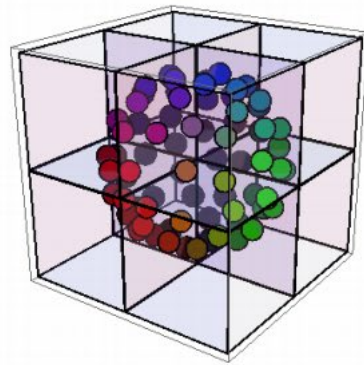
- Dataset: Vienna, 276M points
- **Brute-Force: 85ms / frame (~12 fps)**
- **Progressive:**
 - Budget: 10M points / frame
 - **5.68ms / frame (~176 fps)**
 - Converges in 157ms



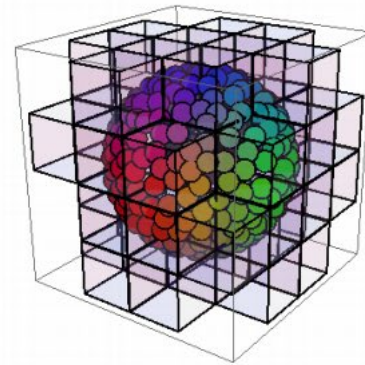
New Converter Prototype



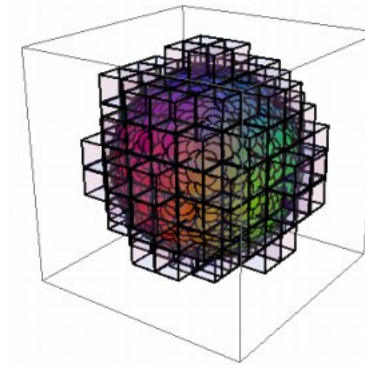
(a) Level 0



(b) Level 1



(c) Level 2



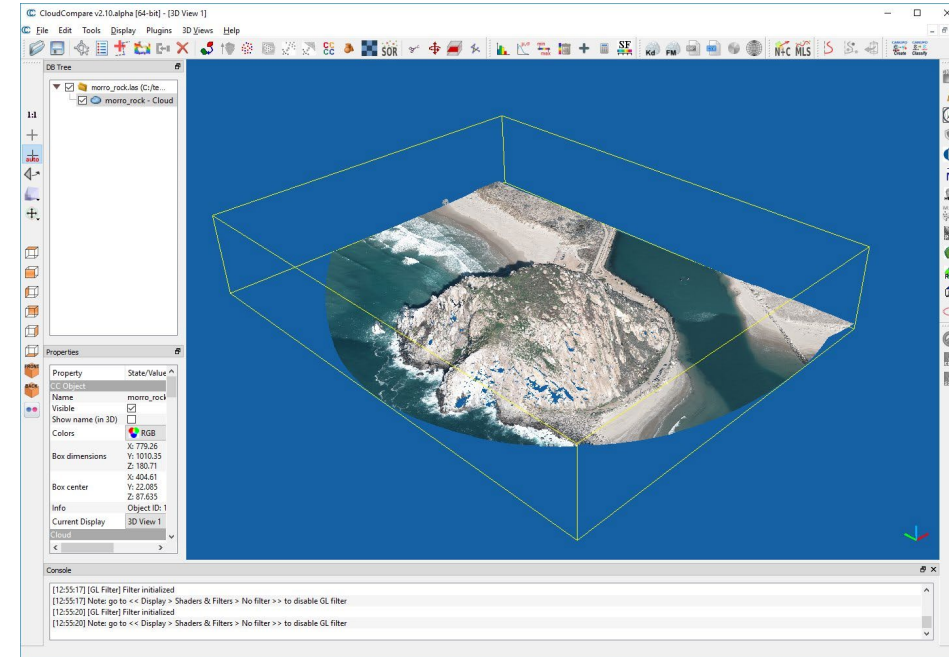
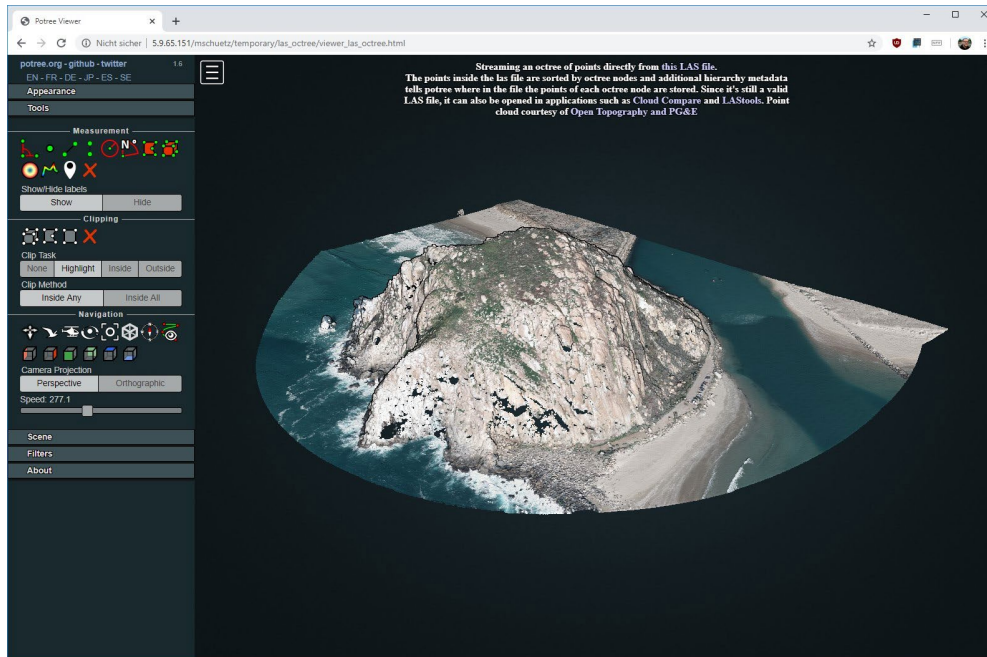
(d) Level 3

- PotreeConverter 1.6 issues:
 - One file per node => millions of files
 - Very slow, less than 1M points / sec
 - Doesn't take advantage of SSD performance
- New Prototype:
 - Octree in a single file
 - ~5M points / second
 - Less memory
 - Parallel processing



New Converter Prototype

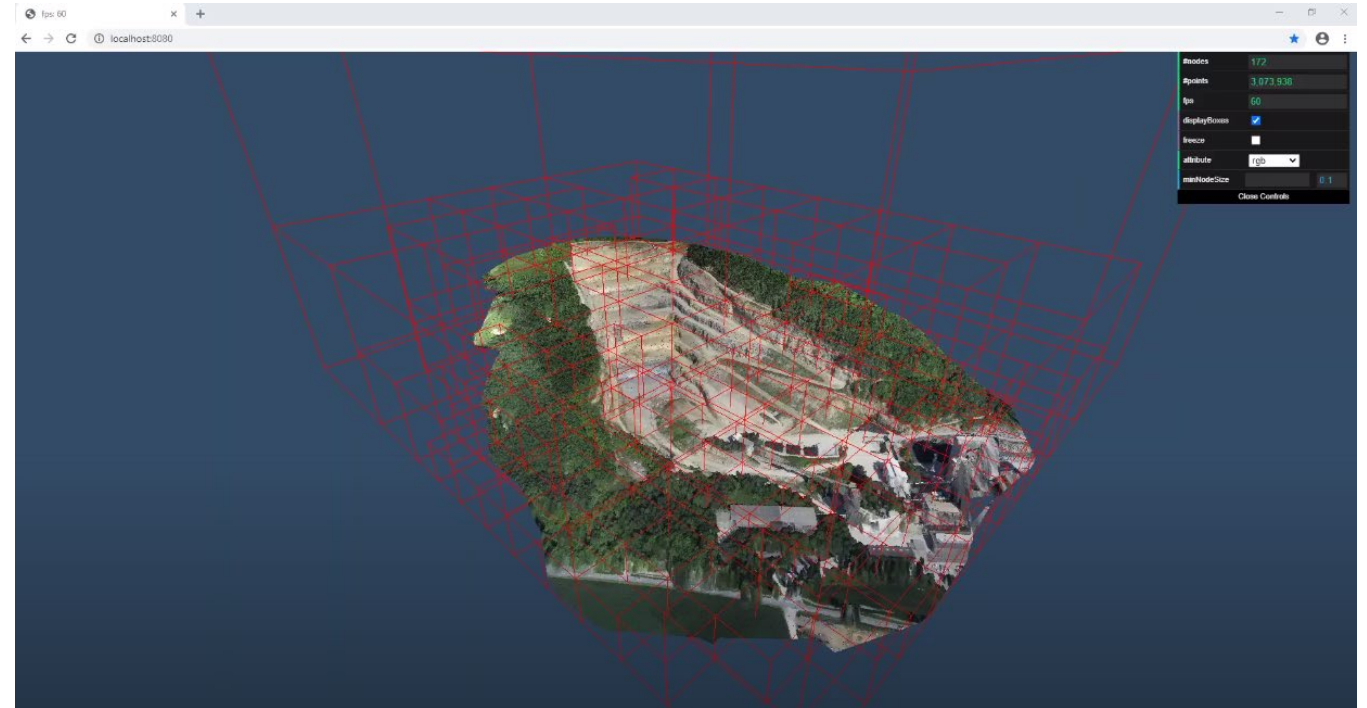
- Experiment: Store octree in a LAS file
 - Stream octree from LAS in Potree
 - Same file usable in Cloud Compare, LAStools, ...
 - http://5.9.65.151/mschuetz/temporary/las_octree/viewer_las_octree.html



Future Work



- Rewrite Potree in WebGPU
 - Modern Graphics API
 - Compute Shaders! Needed for
 - Continuous LOD
 - Progressive rendering
 - Custom rasterization
 - Etc



<https://youtu.be/IWV183LdZGk>



- Simultaneous conversion & viewing
 - Quick preview at 50M points / sec with progressive rendering
 - Full LOD with 10M points / sec
 - View while it's still converting
 - Prioritize conversion of high LODs according to viewer!
 - Applied for research funding to do this, we hope we'll get it!



Thanks for your attention!

