Patch-Trees for Fast Level-of-Detail Synthesis

Hermann Birkholz
University of Rostock
Germany

Outline

- Motivation
- Patch-Tree Construction
- Patch-Tree Rendering
- Results

Motivation

- Large 3d meshes are widely available
  - Laser scan, simulation results, …
- Very powerful GPUs
- CPU has to supply GPU
  - limited bandwidth
- High CPU load with triangle based LOD approaches such as „edge collapse“

Motivation

- Increase LOD granularity from few triangles to patches of triangles
- Geometry of patches can be stored in GPU memory
- Synthesize approximation by stitching patches of different LOD together

Patch-Tree Construction

- Create initial patches
- Straighten initial patch borders
- Create balanced tree from initial patches

Patch-Tree Construction

Initial patch contruction

- Treat triangles as patches
- Iterativley merge adjacent patches, ordered by the resulting shape
  - Priority depends on area and perimeter
- Stop at a given number of patches

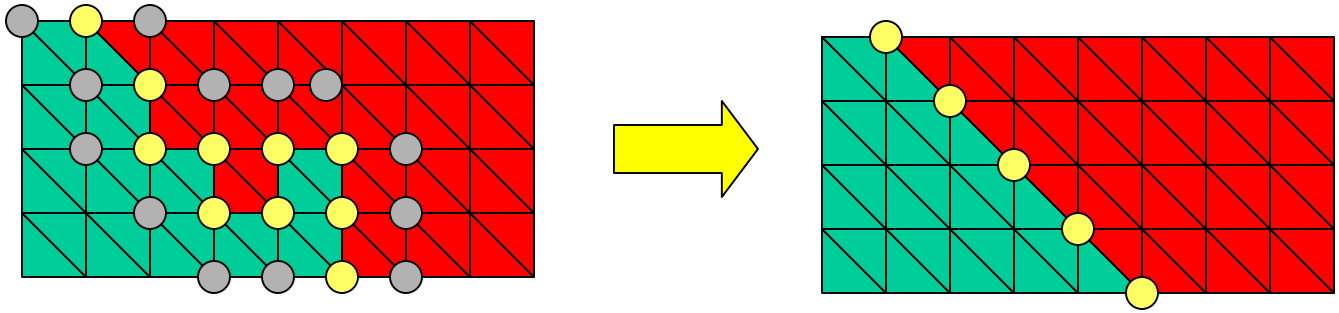
Patch-Tree Construction

Straighten patch borders

- Locally optimize length of borders between adjacent patches
  - Mark optimization corridor
  - Optimize path length
  - Repeat until no further optimization possible

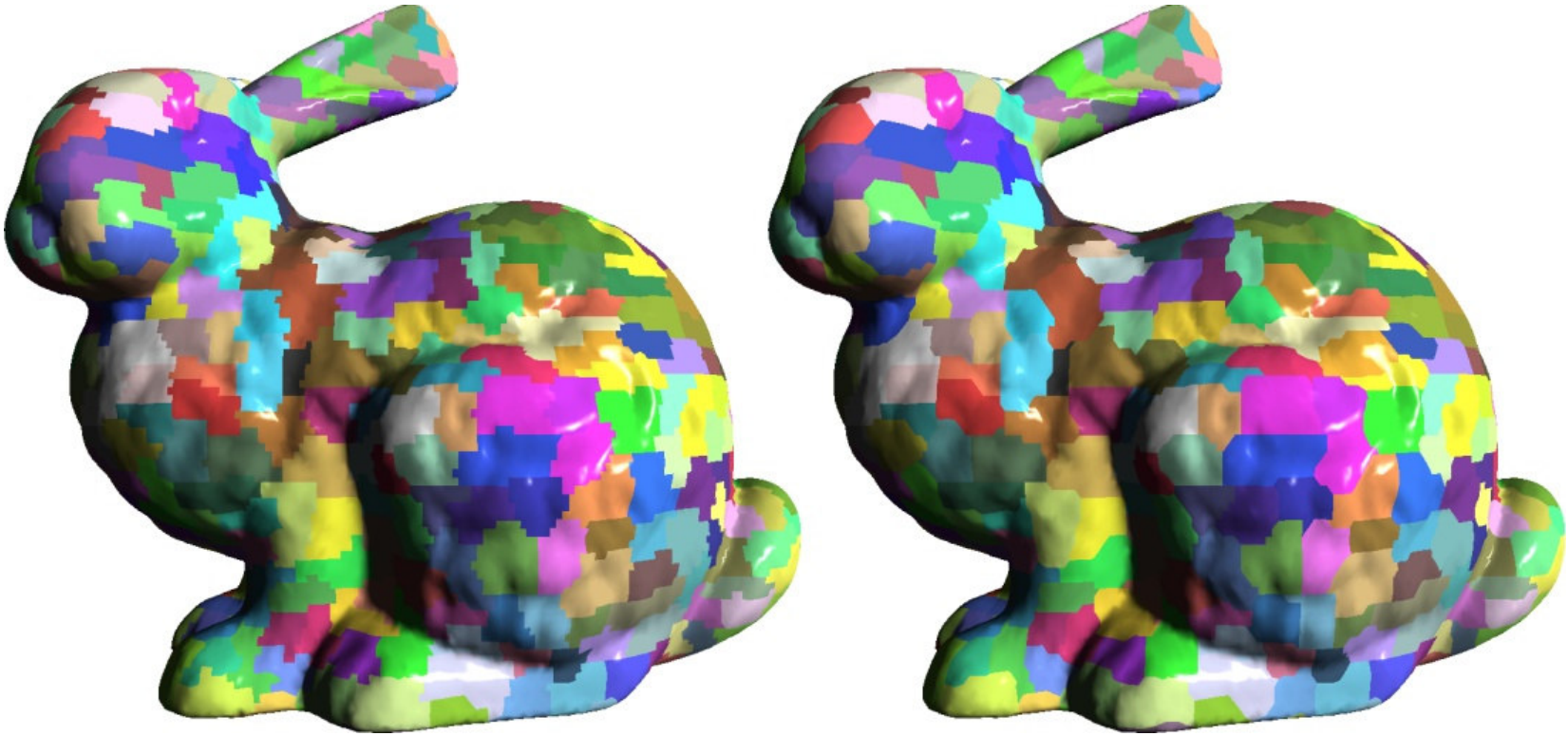
Patch-Tree Construction

Optimize path length



Patch-Tree Construction

Smoothed patch borders

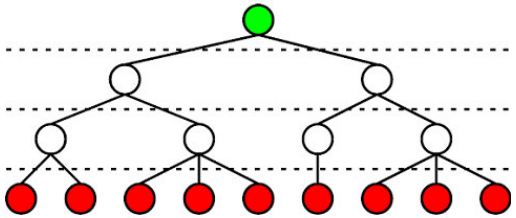


Patch-Tree Construction

Create balanced tree from initial patches

- Iteratively merge adjacent patches
  - Initialize leaf level with initial patches
  - Whenever the number of patches halves, create new tree level
- Simplify geometry in the tree levels
  - Number of triangles halves in each level
  - Compute approximation error per patch

Patch-Tree Construction

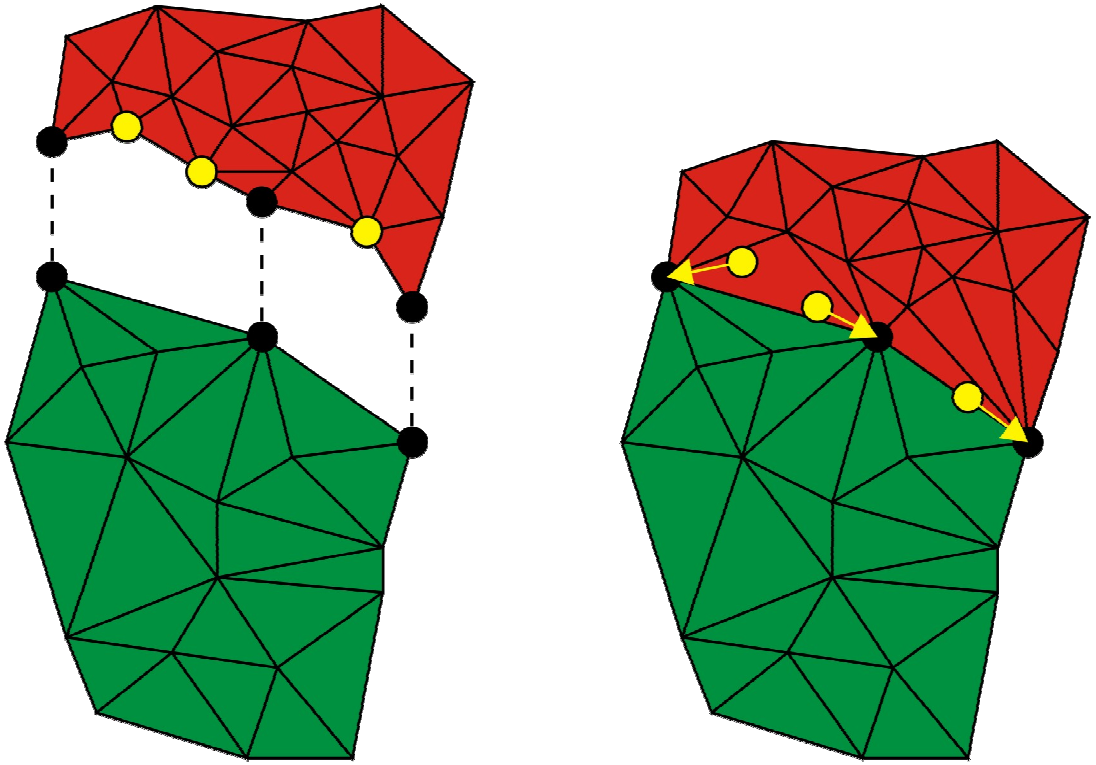


Patch-Tree Rendering

- Initialize approximation with root patch
- Substitute patches with high approximation errors by their childs
  - Keep level difference between adjacent patches below 1 (-> forced splits)
- Repeat until given number of triangles is reached
- Stitch patches together

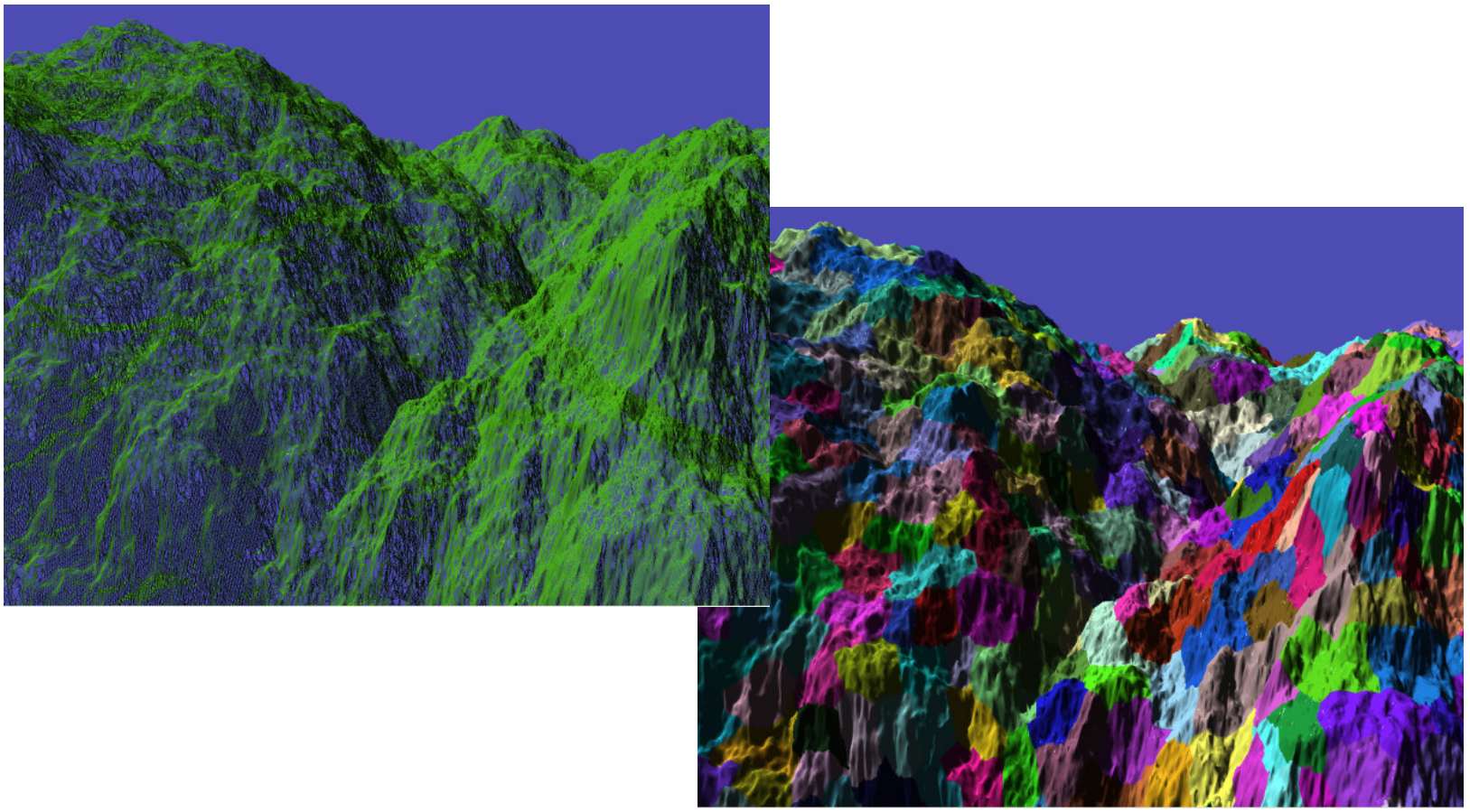
Patch-Tree Rendering

Stitching patches together (precomputed)



Results

- Lower CPU and higher GPU load
- Approx. 50 fps with a 500,000 triangle approximation on a 2GHz Pentium IV with a Radeon 9800 Pro
  - Average patch size 1,000 triangles
- Minor “popping artifacts” at patch substitution



Thanks for your attention!