The POP Buffer: Rapid Progressive Clustering by Geometry Quantization

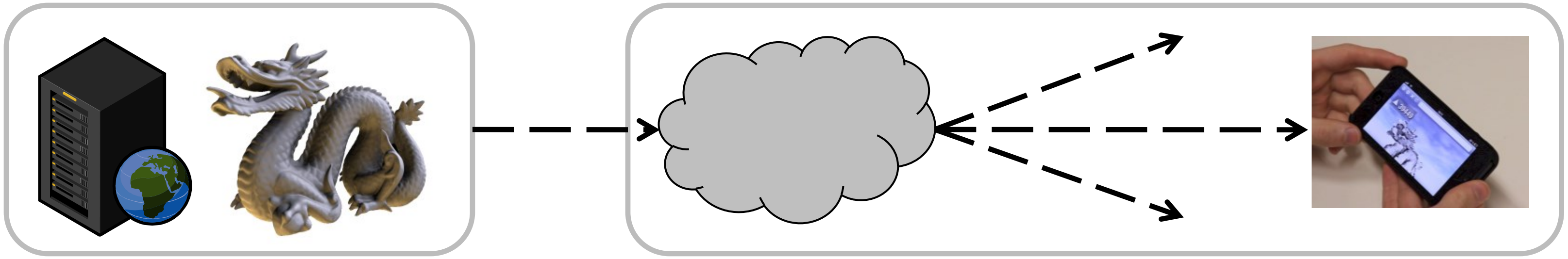
Max Limper[1,2] Yvonne Jung[2] Johannes Behr[2] Marc Alexa[3]

[1]TU Darmstadt [2]Fraunhofer IGD, Darmstadt [3]TU Berlin

Pacific Graphics 2013, Singapore

October 8, 2013

Motivation:
Progressive Mesh Transmission



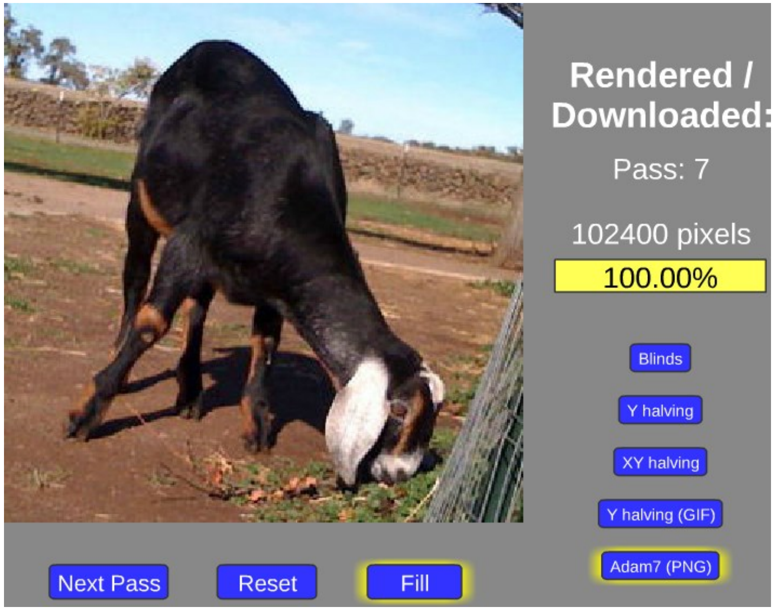
- Progressive Meshes? (Hoppe ‘96, Alliez/Desbrun ‘01, Peng ‘05, …)
  - Most don’t encode **general** 3D meshes, just 2-manifolds
  - Tradeoff: **Bandwidth Usage** vs. **Decode Time**
  - Web & Mobile clients: Limited computational power

- Simple Web-capable Implementation, as known from Images?

based on 09-08-30-js4_PPT_Master_IGD

Progressive Transmission: Interlacing Schemes

Images (PNG, Adam7)



Rendered / Downloaded:

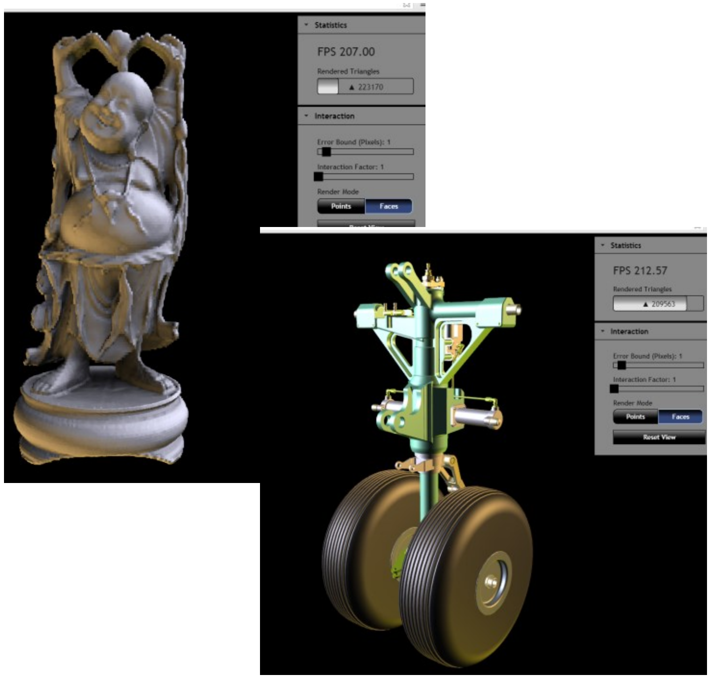
Pass: 7

102400 pixels

100.00%

http://beej.us/blog/data/image-interlacing/

Meshes (POP Buffer)



Statistics

FPS 207.00

Rendered Triangles ▲ 223170

Interaction

Error Bound (Pixels): 1

Interaction Factor: 1

Render Mode: Faces

Statistics

FPS 212.57

Rendered Triangles ▲ 209563

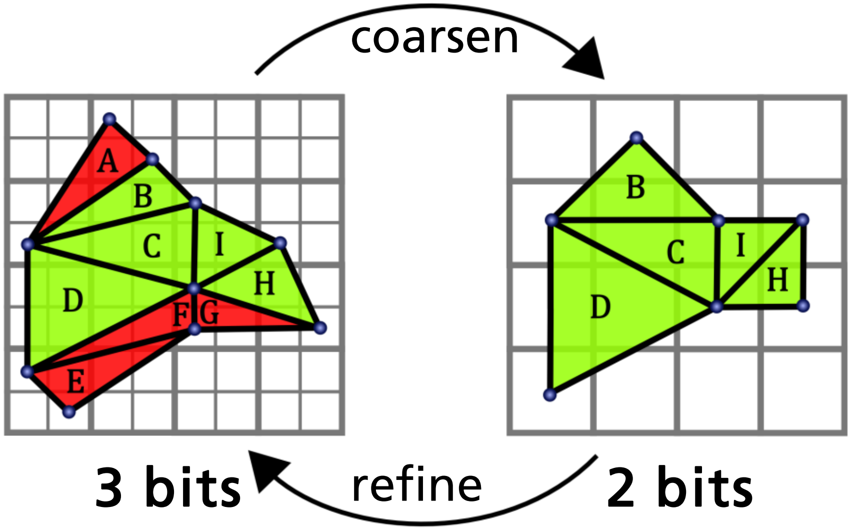
Interaction

Error Bound (Pixels): 1

Interaction Factor: 1

Render Mode: Faces

Ordering Triangles: Finding degenerated geometry



coarsen

refine

3 bits

2 bits

Non-degenerate triangles:

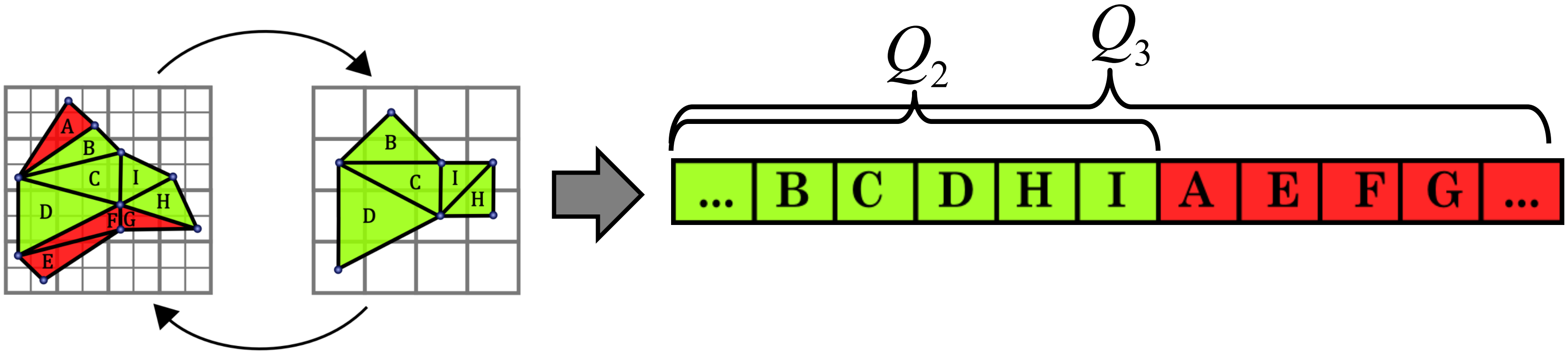
…

$$Q_2 = \{B, C, D, H, I\}$$

$$Q_3 = \{A, B, C, D, E, F, G, H, I\}$$

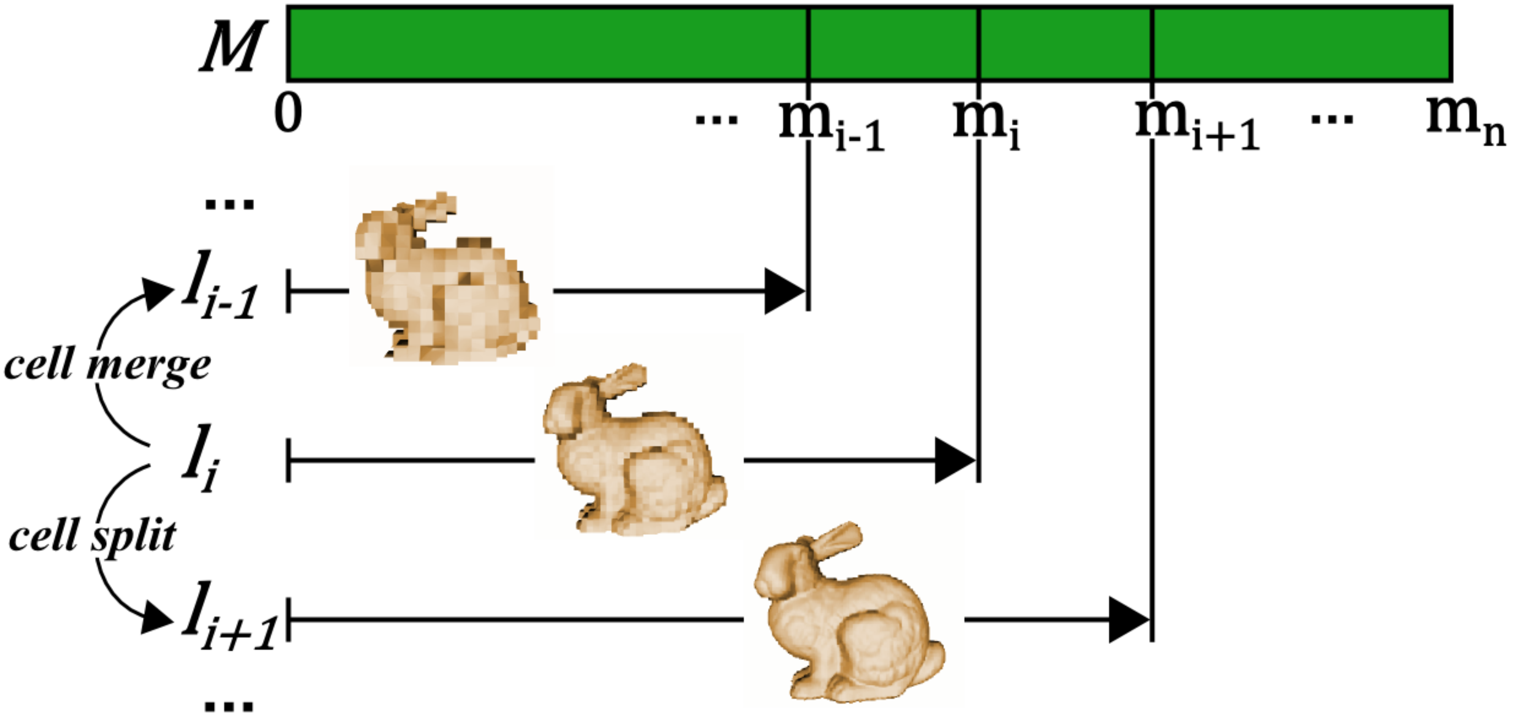
…

Ordering Triangles: Sorting

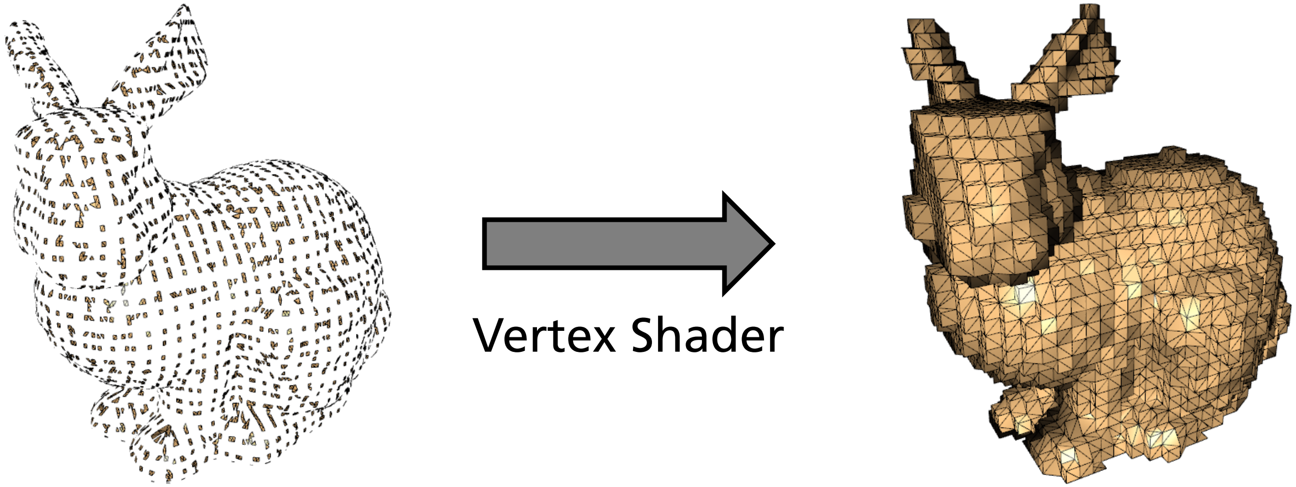


- Classify triangles, sort them
  - Order within sets can be freely chosen
  - Fast sorting (linear time), ~5-6 Mio. Δ / s

Progressively Ordered Primitive (POP) Buffer

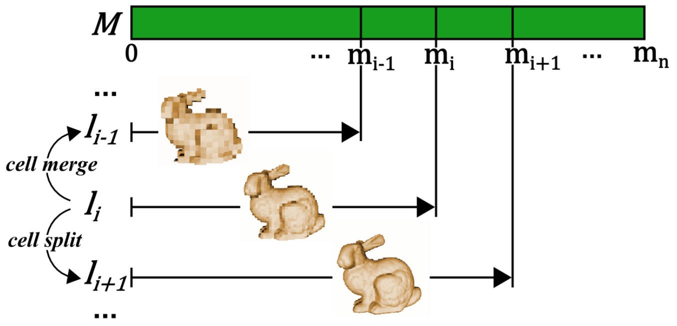


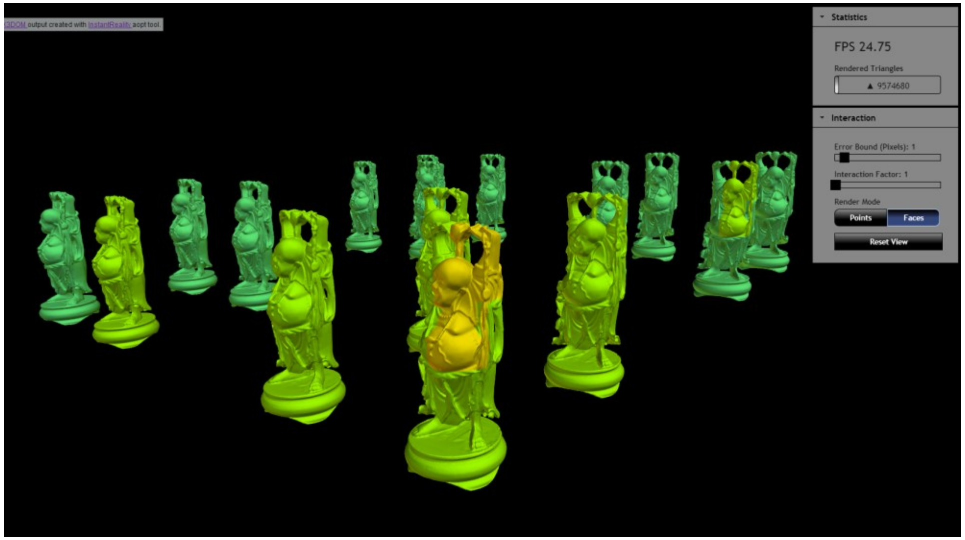
Progressive Transmission & Rendering


Vertex Shader

- Don‘t send less bits/vertex, just send less vertices!
  - Also applies for index data, normals, texture coordinates
  - No additional decoding, guarantees **stateless** property

The Stateless POP Buffer




X3DOM output created with InstantReality aopt tool.

FPS 24.75

Rendered Triangles: ▲ 9574680

Error Bound (Pixels): 1

Interaction Factor: 1

Render Mode: Points [✓ Faces]

- One static buffer for all detail levels
- One static buffer for all instances

The POP Buffer: Discussion: Limitations

- Limited compression capabilities (R/D-performance)
- Limited LOD efficiency
- Reordering limits cache optimization strategies
- Non-rigid mesh animations not possible

The POP Buffer: Discussion: Advantages

- Handles arbitrary topology
- Conversion at interactive rates
- No CPU-based decoding steps
- Stateless structure (less GPU traffic, Instancing)
- No GPU memory overhead
- Straightforward implementation (WebGL)

Try it yourself

- Paper video and demos available at www.x3dom.org/pop

- Demos need WebGL-capable browser

based on 09-08-30-js4_PPT_Master_IGD

Thank you for listening!

Questions?

View-Dependent LOD: Closing Cracks

- Partition mesh for view-dependent LOD
- Problem: Cracks between sub-meshes
- Simple solution: Protected vertices
  - Move border vertices to begin of buffer
  - Always render those with full precision
  - Can be used for shape / feature preservation