Octree Textures on Graphics Hardware

Joe Kniss, Aaron Lefohn, Robert Strzodka, Shubho Sengupta, John D. Owens
Introduction

• 2D texture mapping is difficult on non-trivial models
  – Complex topology
  – Implicit surfaces

• Large body of work on 3D model parameterization, 3D $\rightarrow$ 2D
  – Seams
  – Distortion
Previous Work

• Octree Textures, Siggraph 2002
• Adaptive Texture Maps, 2002
• Streaming Narrow Band, 2003
• Brick-map Textures, 2004
• All Purpose Texture Sprites, 2004
• Multi-res Mesh Atlas, 2004
Multi-resolution Atlas

- Carr et al.
Octree Textures

- Benson & Davis
- DeBry et al.
- Christensen & Batali
Adaptive Texture Maps

- Kraus & Ertl
Streaming Narrow Band

- Lefohn et al.
Texture Sprites & GPU Octex

- Lefebvre et al.
Glift

- STL like abstraction for CPU/GPU data structures
- Primary abstraction of texture data
  - Virtual Domain
  - Address Translation
  - Physical Domain
- Consolidates common data structures
Glift-Octree

- 3D virtual and physical domains
Glift-Octree

- 3D virtual and physical domains
Gliff-Octree

- 3D virtual and physical domains
Glift-Octree

• 3D virtual and physical domains
Address Translation

1) Get physical page address: page table lookup
2) Calculate offset
3) Add offset to PPA
4) Read texture
Address Translation

1) Get physical page address: page table lookup, PPA & scale
2) Calculate offset
3) Add offset to PPA
4) Read texture
Over-representation

- Node Centered scheme
Over-representation

- Node Centered scheme
Painting

- View-space brush profiles
  - Nearly 2D implementation
  - Most texels not “on” surface
Applying the Paint

- Hit-test: Rasterize tex-coord image
- Memory Allocation: Map physical tile
- Refinement: Interpolate low res $\rightarrow$ high res
- Write texels + Over-rep at tile edges
Limitations

- Occlusion
- Large brush rasterization
- Normal keys
Filtering

• MIP mapping is important
• Difficult
  – Multi-resolution data structure
  – Node-centered scheme
  – Approximate surface method using 3D samples

SIGGRAPH 2005
Filtering

• MIP mapping is important
• Difficult
  – Multi-resolution data structure
  – Node-centered scheme
  – Approximate surface method using 3D samples
Filtering

- MIP mapping is important
- Difficult
  - Multi-resolution data structure
  - Node-centered scheme
  - Approximate surface method using 3D samples
Filtering

- MIP mapping is important
- Difficult
  - Multi-resolution data structure
  - Node-centered scheme
  - Approximate surface method using 3D samples
Filtering

• MIP mapping is important
• Difficult
  – Multi-resolution data structure
  – Node-centered scheme
  – Approximate surface method using 3D samples
Solution

• Weight each sample by resolution level
  – Lower resolution samples contribute more
  – Should be more high-res samples per unit volume
  – Resolution level stored in alpha channel
Filtering Limitations

• Only approximate solution
  – Would be better using 3D rasterization?
  – Need to know area of surface associated with sample
  – Open problem
Results

• Demo
Results

• Frame rates are geometry bound
• Benchmarks = ½ normal texture
• Key characteristics of datastructure:
  – Adaptive & Sparse representation
  – O(N) memory usage
  – O(1) memory access / computation
  – Uniform (SIMD) computation
  – Uses native GPU filtering
• 2048^3 texture resolution
Results

- Cost of indirection
  - RGBA8, Nvidia GeForce 6800 GT, 75.22
Conclusion

• Real-time Octree painting on GPU possible
• Not actually an octree
  – Shallow branching “N-tree”
  – Designed for optimality on GPU
  – Continuum of trees from ours to Lefebvre
  – 2-level “N-tree” should be sufficient

• Open problems:
  – Fast Brushing
  – Correct & Fast filtering