Interactive Rendering of Massive Geometric Models

Enrico Gobbetti
CRS4 Visual Computing

(CRS4 Visual Computing Group)
- Staff
  - 6+ people
- RTD
  - Geometry processing / rendering
  - Scientific visualization
  - Haptics
  - VR & Simulation
- Service
  - Sci Viz + Post production

Interactive Rendering of Massive Geometric Models

Enrico Gobbetti
CRS4 Visual Computing

(CRS4 in one slide)
- Interdisciplinary research center focused on computational sciences
  - No-profit consortium
    - RAS(C21), IBM, STM, UHCA, UniSS, Saras, Tikal
  - Operational since 1992
- RTD staff of ~80 people
- Turnover of ~7M Euro, of which ~50% from external funding
  - EU/National research project
  - Industrial contracts

Goal and Motivation
Accurate interactive inspection of very large models on PC platforms...

Application domains / data sources
- Many important application domains
- Models exceed
  - O(10^5–10^9) samples
  - O(10^10) bytes
- Varying
  - Dimensionality
  - Topology
  - Sampling distribution

Enrico Gobbetti, February 17th, 2005

UniSS, Saras, Tiscali

Dense regular sampling
Accurate interactive inspection of very large models on PC platforms...

Local Terrain Models
2.5D – Path – Dense regular sampling

Planetary terrain models
2.5D – Spherical – Dense regular sampling

Laser scanned models
3D – Naturaltopology – low depth complexity – dense

CAD models
3D – complex topology – high depth complexity – structured

Natural objects / Simulation results
3D – complex topology + high depth complexity + unstructured / high frequency details

Xeon 2.4GHz / 1GB RAM / 70GB SCSI 330 disk / NVIDIA 6800GT

Enrico Gobbetti, February 17th, 2005
Interactive rendering constraints

- Frequency, latency, resolution should match human capabilities.
- ... or at least output device's ones!
- On today's displays
  - Frequency: 10-100Hz
  - Latency: ~0.1s
  - Resolution: O(10^9-10^10) pixels

Tiled high resolution displays
- ~10-100M pixels, tiling

Holographic displays
- ~10-100M pixels, tiling

Why large scale model visualization research? (1/2)

- ... because large scale models are too large for brute force approaches in interactive applications!

Size matters! Or does it? (1/15)

Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)

Size matters! Or does it? (2/15)

Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)

Size matters! Or does it? (3/15)

Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)

Multiresolution + ...
Size matters! Or does it? (4/15)
Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)
Multiresolution + View dependent LOD selection + ...

Size matters! Or does it? (5/15)
Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)
Multiresolution + View dependent LOD selection + View culling + ...

Size matters! Or does it? (6/15)
Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)
Multiresolution + View dependent LOD selection + View culling + Occlusion culling + ...

Size matters! Or does it? (7/15)
Out-of-core output-sensitive techniques
Goal: Time/Memory Complexity = O(N) (independent of K)
Multiresolution + View dependent LOD selection + View culling + Occlusion culling + External memory management

Size matters! Or does it? (8/15)
Out-of-core output-sensitive techniques
• At preprocessing time: build MR hierarchy

Size matters! Or does it? (9/15)
Out-of-core output-sensitive techniques
• At preprocessing time: build MR hierarchy
• At run time: selective view-dependent refinement
  - Stop when node accurate, out-of-view, or occluded
Size matters! Or does it? (10/15)
Out-of-core output-sensitive techniques

- At preprocessing time: build MR hierarchy
- At run time: selective view-dependent refinement
  - Stop when node accurate, out-of-view, or occluded
  - Use dependencies to maintain structure consistent

Size matters! Or does it? (11/15)
Out-of-core output-sensitive techniques

- At preprocessing time: build MR hierarchy
- At run time: selective view-dependent refinement
  - Stop when node accurate, out-of-view, or occluded
  - Use dependencies to maintain structure consistent

Size matters! Or does it? (12/15)
Out-of-core output-sensitive techniques

- At preprocessing time: build MR hierarchy
- At run time: selective view-dependent refinement
  - Stop when node accurate, out-of-view, or occluded
  - Use dependencies to maintain structure consistent
- Keep hierarchy cut in-core, load data on demand
  - Reduce/avoid 1/0 latency by
    - Rendering data
    - Compressing data
    - Predict data misses (prefetching)

Size matters! Or does it? (13/15)
Out-of-core output-sensitive techniques

- Many (many!) data structure/algorithm variations on this theme:
  - Hierarchies/DAGs
    - Evolutionary models
      - Vertex split/edge collapse
      - Vertex insertion/deletion
      - Vertex contraction
    - Octree
      - Adaptive octree
      - Coarser grids
      - Voxel grids
    - Nested models for 2.5D datasets
      - Mimesis models
        - graveyard
  - Granularity = point/triangle/vertex
  - Occlusion culling independent of LOD construction/selection
    - Space partitioning
      - On-line (from page)
      - Off-line (from region)
    - Granularity = cell/region

Size matters! Or does it? (14/15)
Out-of-core view-dependent simplification

- Build point / vertex hierarchy, refine it at run-time
  - ElSana2000, Rus2000, Lin2003,
- CPU bound
  - High per-primitive selection and culling costs
  - Hard to use preferential data paths
  - Hard to build and maintain optimized graphics representations
- Hard to combine with visibility culling methods

Size matters! Or does it? (15/15)
Out-of-core chunk-based techniques

- Partition model into chunks, simplify each chunk independently, build LOD hierarchy
  - Erik2001, Var2002
- GPU friendly
  - Each chunk is an independent mesh
  - LOD selection costs amortized on many primitives
- Hierarchical partitioning useful for visibility culling
- Problems at block boundaries
  - Cracks / costly CPU updates / low simplification quality
Our contributions
GPU-friendly output-sensitive techniques

- Underlying ideas
  - Chunk-based multiresolution structures
  - Seamless combination of surface chunks
- Complex rendering primitives
  - GPU programming features
  - Complex patches, view-dependent variables
- Chunk-based memory management
  - Compression/decompression, block transfers, caching

Our contributions
GPU-friendly output-sensitive techniques

- Adaptive TetraPuzzles
  - High performance visualization of dense 3D meshes
  - Two-level multiresolution model based on volumetric decomposition

Our contributions
Adaptive TetraPuzzles – Dense 3D meshes

Construction

Target = k triangles/chunk
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

• Construction

...
Interactive Rendering of Massive Geometric Models

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- Construction

Adaptive Tetrapuzzles – Dense 3D meshes

- Construction

(6 tetra / diamond)
(4 tetra / diamond)
(8 tetra / diamond)

Adaptive Tetrapuzzles – Dense 3D meshes

- Construction

Adaptive Tetrapuzzles – Dense 3D meshes

- Construction
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

• Construction

Adaptive Tetrapuzzles – Dense 3D meshes

• Construction

k triangles/chunk

Adaptive Tetrapuzzles – Dense 3D meshes

• Construction

Adaptive Tetrapuzzles – Dense 3D meshes

• Construction

Diamond external boundary
Diamond internal boundary
Child tetrahedra boundary
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- Construction

Diamond external boundary
Diamond internal boundary
Child tetrahedra boundary

Interactive Rendering of Massive Geometric Models
E. Gobbetti, February 17th, 2005

NO CRACKS / NO GLOBALLY LOCKED BOUNDARY!
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- Independent diamond processing
- For each mesh chunk: Simplify + stripify + compress + eval bounds/error
- Out-of-core + parallel
- Out-of-core cull+refine traversal
- GPU cached optimized meshes

SEE PAPER FOR DETAILS

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- Linux/MPI Construction
- OpenGL renderer
- VBO
- Prefetch
- mincore/mmap interface

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- 1-14 Athlon 2200+ CPU, 3 x 70GB ATA 133 Disk (IDE+NFS)
- 3-30K triangles/sec
  - Scales well, limited by slow disk I/O for large meshes
- 96-144 bits/triangle (~lossless)
  - Comparable to other view-dependent simplification methods

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- Xeon 2.4GHz, 70GB SCSI 320 Disk, NVIDIA GeForce FX5800U
- GPU bound
  - ~70M-100M triangles/sec
  - >60Hz when rendering at ± 2px tolerance on a 800x600 window with 4x FSAA
- Resident set size limited to ~150MB

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- Tested on a number of large data sets
  - Bonsai CT / David 2mm / David 1mm / St. Matthew 0.25mm
- Tested in a number of situations
  - Single processor / cluster construction
  - Workstation viewing, large scale display

Interactive Rendering of Massive Geometric Models
E. Gobbetti, February 17th, 2005

Copyright 2005
www.crs4.it/vic/
**Our contributions**

**Adaptive TetraPuzzles – Dense 3D meshes**

- **Adaptive TetraPuzzles**: High performance visualization of dense 3D meshes
  - Two-level multiresolution model based on volumetric decomposition

**Our contributions**

**P-BDAM – Planetary terrain models**

- **P-BDAM**: High performance planetary terrain visualization technique
  - Handles planet curvature
  - The only accelerated technique with sub-meter global accuracy on entire Earth
  - Parallel construction method

**Our contributions**

**GPU-friendly output-sensitive techniques**

- **BDAM - Local Terrain Models**
  - Gobbets/Martin (2005)
  - Cignoni/Giacomelli/Ponsio/Alessandri (2005)

- **P-BDAM - Planetary terrain models**
  - Gobbets/Martin (2005)
  - Cignoni/Giacomelli/Ponsio/Alessandri (2005)

- **Adaptive TetraPuzzles – dense mesh models**
  - Gobbets/Martin (2005)
  - Cignoni/Giacomelli/Ponsio/Alessandri (2005)

- **Layered Point Clouds – dense point clouds**
  - Gobbets/Martin (2005)
  - UMR J2AV / Computer A-Graphs J2AV

- **Far Volumes – General**
  - Gobbets/Martin (2005)
  - [Under review – Stay tuned]
Our contributions

GPU-friendly output-sensitive techniques
- BDAM - Local Terrain Models
  Gobbetti/Marton (CRS4), Cignoni/Garanelli/Porta/Scopigno (ISTI-CNR) 2002
- Far Voxels - General 3D models
  Gobbetti/Marton (CRS4), Cignoni/Garanelli/Porta/Scopigno (ISTI-CNR) 2002

Far Voxels - General 3D models
- Classic multisresolution models
  - Error measures on boundary surfaces
  - Visibility culling decoupled from multisolution
- Hard to apply to models with high detail and complex topology and high depth complexity!

Our contributions

Far Voxels - General 3D models
- Far Voxels: High performance visualization of arbitrary 3D models
  - Mixed model
  - Seamless integration of occlusion culling with out-of-core data management and multiresolution rendering
    - ... work in progress
**Our contributions**

**Far Voxels – General 3D models**

- **Off-line Reconstruction**
  - Sampling
  - Raycasting
  - Occlusion culling
    - Sample from distance dictated by maximum possible projected voxel size
- **Fitting**
  - Choose best voxel representation among selected parameterized shaders
  - Error minimization
- **On-line Rendering**
  - Refine until projected voxel size < desired accuracy
  - Exploit GPU for shader evaluation and on-line occlusion culling

---

**Conclusions**

- Many high performance models
  - BDAM/P-BDAM: Terrains
  - LPC: Dense point sampled models
  - ATP: Dense triangle meshes
  - FARVOX: General 3D models
- Current/Future work: a lot
  - Generalize mesh-based framework
  - Improve quality of volumetric framework
  - Improved voxel shaders
  - Fragment-based volumetric renderer
  - Introduce (limited) interactive manipulation features
  - Compression + Streaming + Next generation displays

---

**So many things, so little time...**

- More info:
  - http://www.crs4.it/vic/
  - http://vcg.isti.cnr.it/
- Models courtesy of Stanford Graphics Group /NASA MOLA / ISTAR / The Boeing Company
- Q&A: Your turn...