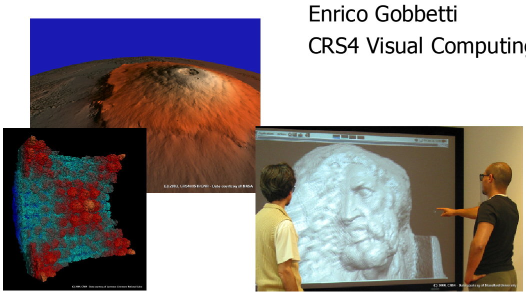
Interactive Rendering of Massive Geometric Models

Enrico Gobbetti
CRS4 Visual Computing

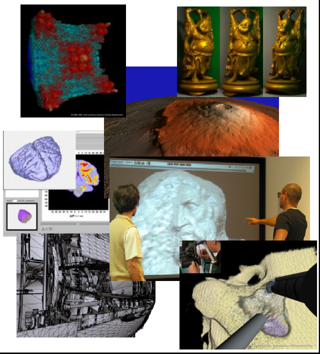


(CRS4 in one slide)

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

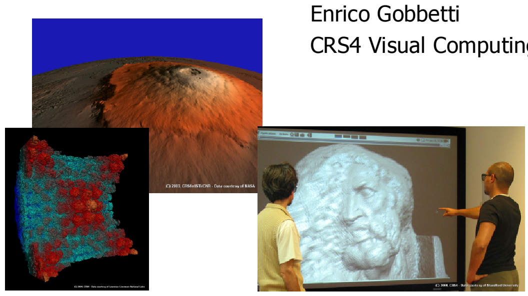
(CRS4 Visual Computing Group)

- Staff
  - 6+ people
- RTD
  - Geometry processing / rendering
  - Scientific visualization
  - Haptics
  - VR & Simulation
- Service
  - Sci Viz + Post production



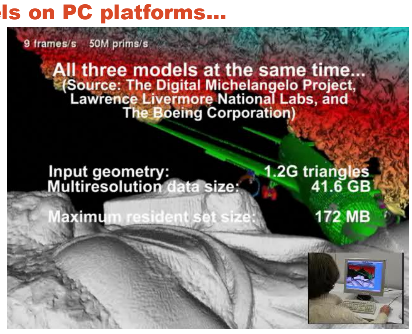
Interactive Rendering of Massive Geometric Models

Enrico Gobbetti
CRS4 Visual Computing



Goal and Motivation

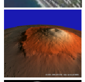
Accurate interactive inspection of very large models on PC platforms…



Xeon 2.4GHz / 1GB RAM / 70GB SCSI 320 Disk / NVIDIA 6800GTS

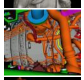
Application domains / data sources

Local Terrain Models
2.5D – Flat – Dense regular sampling

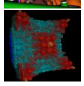


Planetary terrain models
2.5D – Spherical – Dense regular sampling

Laser scanned models
3D – Moderately simple topology – low depth complexity - dense



CAD models
3D – complex topology – high depth complexity – structured - 'ugly' mesh



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

- Many important application domains
- Models exceed
  - $O(10^8\text{-}10^9)$ samples
  - $O(10^9)$ bytes
- Varying
  - Dimensionality
  - Topology
  - Sampling distribution

Interactive rendering constraints

Regular desktop displays
~1M pixels

Geowall-type displays
~1-10M pixels, stereo

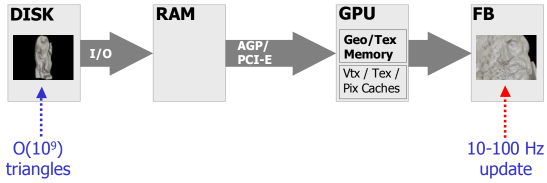
Tiled high resolution displays
~10-100M pixels

Holographic displays
~10-100M pixels, holo

- 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: $O(10^6\text{-}10^7)$ px

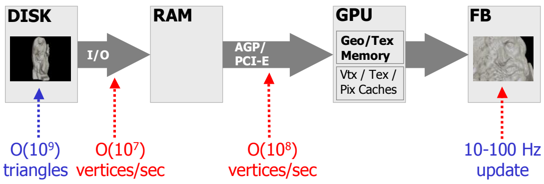
Why large scale model visualization research? (1/2)

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



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

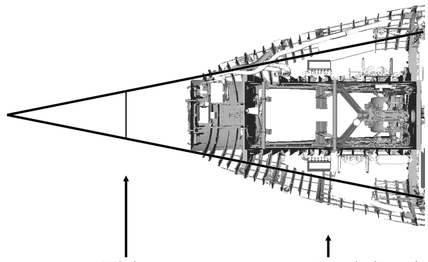
- Real-time rendering needs to rapidly move data from disk (to RAM) to GPU
  - => out-of-core data management
  - => adaptive techniques to reduce data transfers



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

Out-of-core output-sensitive techniques

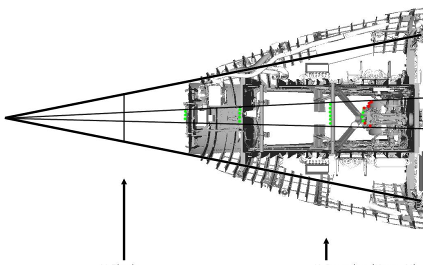
Goal: Time/Memory Complexity = O(N) (independent of K)



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

Out-of-core output-sensitive techniques

Goal: Time/Memory Complexity = O(N) (independent of K)

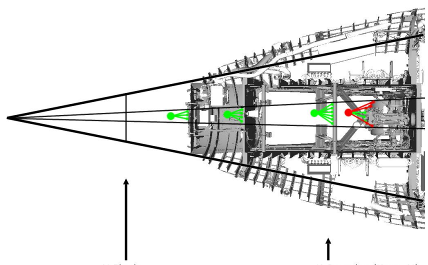


Size matters! Or does it? (3/15)

Out-of-core output-sensitive techniques

Goal: Time/Memory Complexity = O(N) (independent of K)

Multiresolution + …

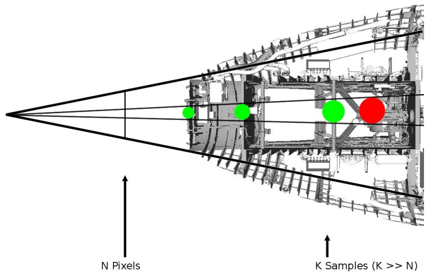


Size matters! Or does it? (4/15)

Out-of-core output-sensitive techniques

Goal: Time/Memory Complexity = O(N) (independent of K)

Multiresolution + View dependent LOD selection + …

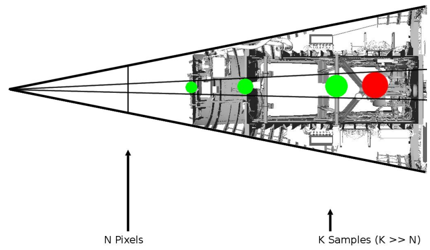


Size matters! Or does it? (5/15)

Out-of-core output-sensitive techniques

Goal: Time/Memory Complexity = O(N) (independent of K)

Multiresolution + View dependent LOD selection + View culling + …

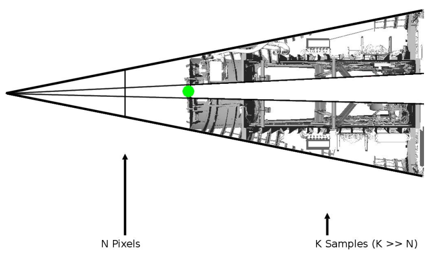


Size matters! Or does it? (6/15)

Out-of-core output-sensitive techniques

Goal: Time/Memory Complexity = O(N) (independent of K)

Multiresolution + View dependent LOD selection + View culling + Occlusion culling + …

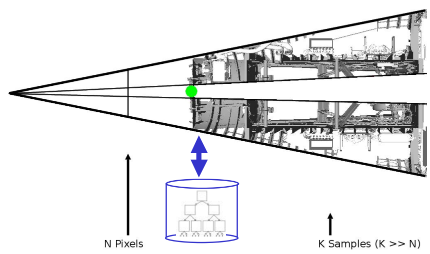


Size matters! Or does it? (7/15)

Out-of-core output-sensitive techniques

Goal: Time/Memory Complexity = O(N) (independent of K)

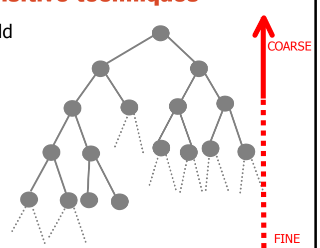
Multiresolution + View dependent LOD selection + View culling + Occlusion culling + External memory management



Size matters! Or does it? (8/15)

Out-of-core output-sensitive techniques

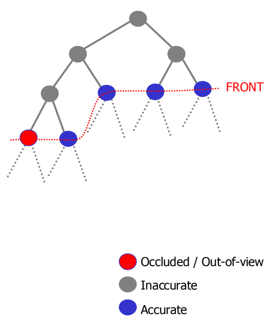
- At preprocessing time: build MR hierarchy



Size matters! Or does it? (9/15)

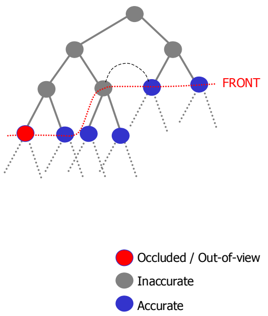
Out-of-core output-sensitive techniques

- At preprocessing time: build MR hierarchy
- At run time: selective view-dependent refinement
  - Stop when node accurate, out-of-view, or occluded



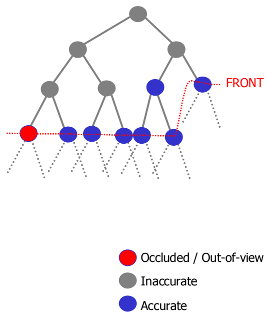
Size matters! Or does it? (10/15)
Out-of-core output-sensitive techniques

- 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



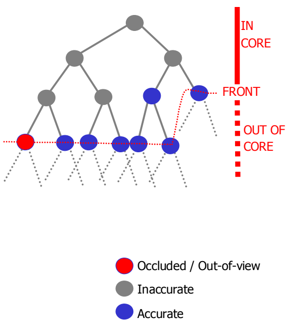
Size matters! Or does it? (11/15)
Out-of-core output-sensitive techniques

- 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 I/O latency by
    - Reordering data
    - Compressing data
    - Predict data misses (prefetching)

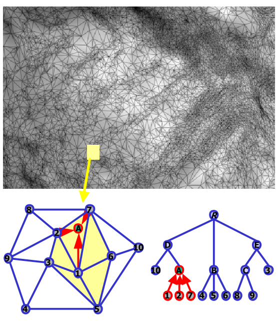


Size matters! Or does it? (13/15)
Out-of-core output-sensitive techniques

- Many (many!) data structure/algorithm variations on this theme:
  - Hierarchies/DAGs
    - Evolutionary models
      - Vertex split/edge collapse
        - » Hoppe1996/97/98, Xia1996/97, Maheswari1997, Gueziec1998, Kobbelt1998, …
      - Vertex insertion/decimation
        - » De Floriani1989, deBerg1995, Cignoni1995/97, Brown1996/97, Klein1996, DeFloriani1996/97/98, …
    - Nested models for 2.5D datasets
      - VonHerzen1987, Gross1996, …
    - Meshless models
      - Rusinkiewicz2000, …
  - Granularity = point/triangle/vertex
- Occlusion culling independent of LOD construction/selection
  - Space partitioning
    - On-line (from point)
    - Off-line (from region)
  - Granularity = cell/region

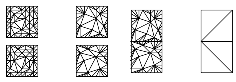
Size matters! Or does it? (14/15)
Out-of-core view-dependent simplification

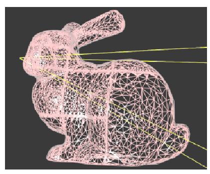
- Build point / vertex hierarchy, refine it at run-time
  - ElSana2000, Rus2000, Lin2003, …
- CPU bound
  - High per-primitive selection and culling costs
  - Hard to use preferential data paths
  - Hard to build and maintain optimized graphics representations
- Hard to combine with visibility culling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

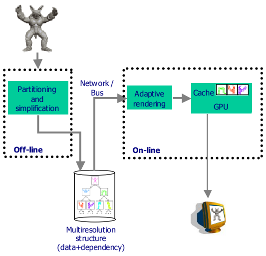




HLOD View-Dependent Rendering (Erikson et al., 2001)

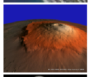
Our contributions
GPU-friendly output-sensitive techniques

- Underlying ideas
  - **Chunk-based multiresolution structures**
    - Combine space partitioning + level of detail
    - Same structure used for visibility and detail culling
  - **Seamless combination of surface chunks**
    - Dependencies ensure consistency at the level of chunks
  - **Complex rendering primitives**
    - GPU programming features
    - Curvilinear patches, view-dependent voxels, …
  - **Chunk-based external memory management**
    - Compression/decompression, block transfers, caching



Our contributions
GPU-friendly output-sensitive techniques

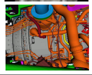
BDAM - Local Terrain Models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
EUROGRAPHICS 2003



P-BDAM - Planetary terrain models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
IEEE Visualization 2003

Adaptive Tetrapuzzles – Dense mesh models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
SIGGRAPH 2004

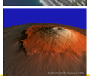
Layered Point Clouds – Dense point clouds
Gobbetti/Marton (CRS4)
SPBG 2004 / Computers & Graphics 2004



Far Voxels – General
Gobbetti/Marton (CRS4)
[Under review – Stay tuned]

Our contributions
GPU-friendly output-sensitive techniques

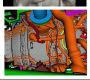
BDAM - Local Terrain Models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
EUROGRAPHICS 2003



P-BDAM - Planetary terrain models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
IEEE Visualization 2003

Adaptive Tetrapuzzles – Dense mesh models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
SIGGRAPH 2004

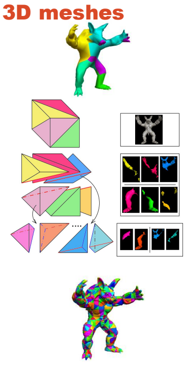
Layered Point Clouds – Dense point clouds
Gobbetti/Marton (CRS4)
SPBG 2004 / Computers & Graphics 2004



Far Voxels – General
Gobbetti/Marton (CRS4)
[Under review – Stay tuned]

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

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



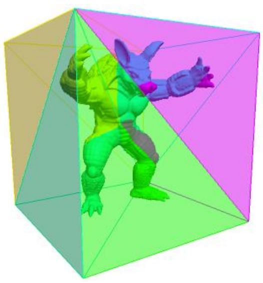
Cignoni, Ganovelli, Gobbetti, Marton, Ponchio, and Scopigno.
Adaptive TetraPuzzles - Efficient Out-of-core Construction and Visualization of Gigantic Polygonal Models.
ACM Transactions on Graphics, 23(3), August 2004
(Proc. SIGGRAPH 2004).

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

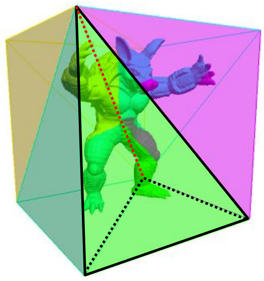
- **Construction**



Target = *k* triangles/chunk

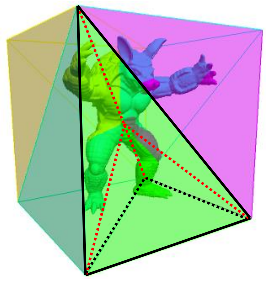
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



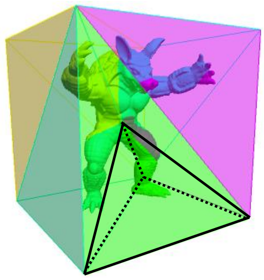
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



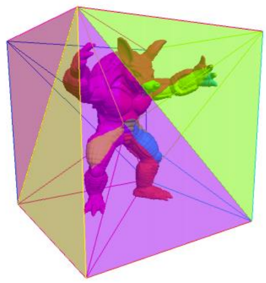
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



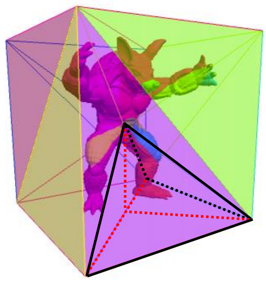
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



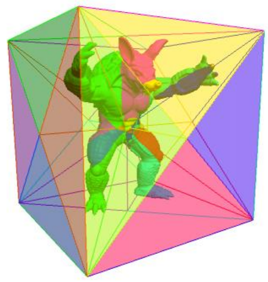
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



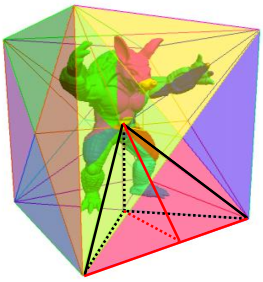
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



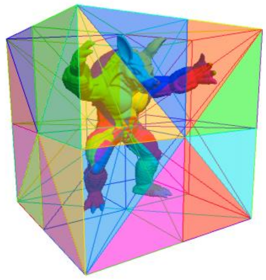
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



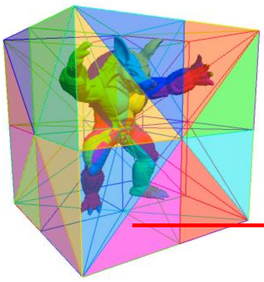
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

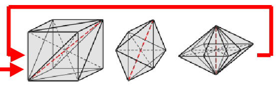
- **Construction**



Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**

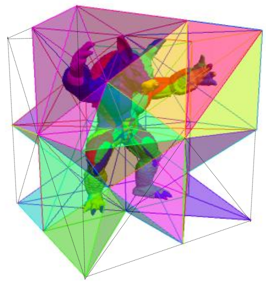




(6 tetra / diamond) (4 tetra / diamond) (8 tetra / diamond)

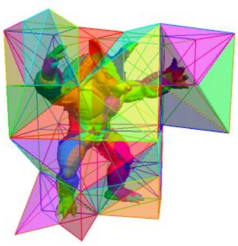
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



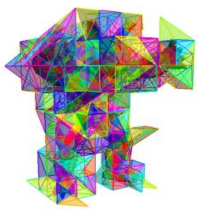
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**

k triangles/chunk

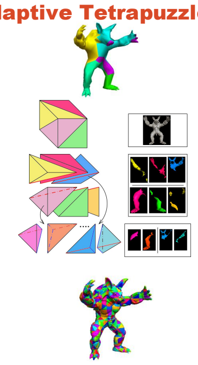
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**

k triangles/chunk

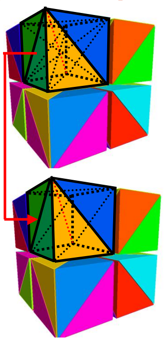
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



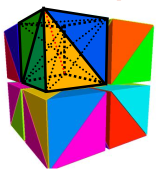
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

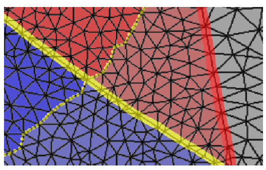
- **Construction**



Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

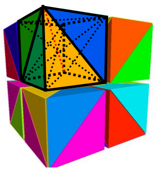
- **Construction**

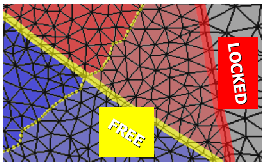




Diamond external boundary
Diamond internal boundary
Child tetrahedra boundary

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

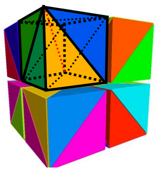


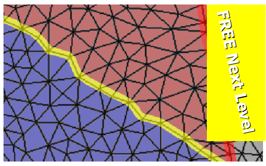


- **Construction**

Diamond external boundary
Diamond internal boundary
Child tetrahedra boundary

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes



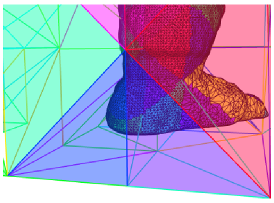


- **Construction**

Diamond external boundary
Diamond internal boundary

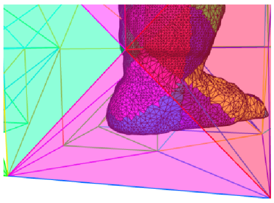
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



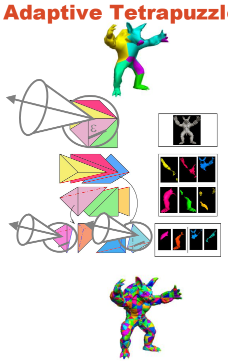
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- **Construction**



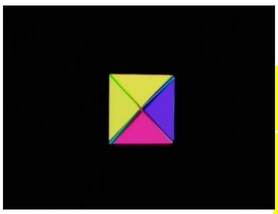
NO CRACKS / NO GLOBALLY LOCKED BOUNDARY!

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes



- **Construction**

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes



View dependent mesh refinement

- **Construction**

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

Independent diamond processing

For each mesh chunk: Simplify + stripify + compress + eval bounds/error

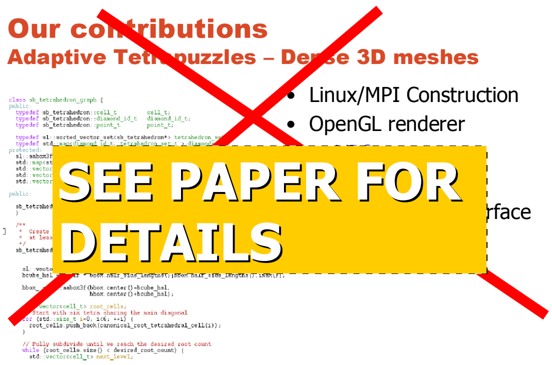
Out-of-core + parallel

Out-of-core cull+refine traversal / GPU cached optimized meshes

Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

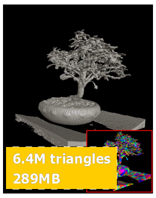
- Linux/MPI Construction
- OpenGL renderer
  - VBO
  - Prefetch
- mincore/mmap interface

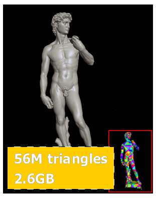
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

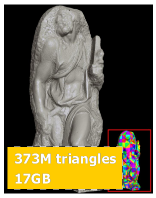


Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- 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





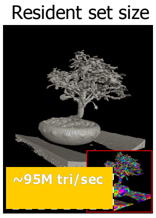


Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- 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

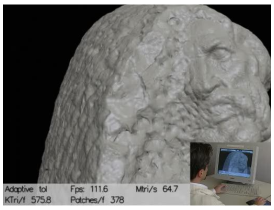
Our contributions
Adaptive Tetrapuzzles – Dense 3D meshes

- 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

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

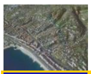


Cignoni, Ganovelli, Gobbetti, Marton, Ponchio, and Scopigno.
Adaptive TetraPuzzles - Efficient Out-of-core Construction and Visualization of Gigantic Polygonal Models.
ACM Transactions on Graphics, 23(3), August 2004
(Proc. SIGGRAPH 2004).

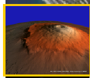
Michelangelo's St. Matthew
Source: Digital Michelangelo Project
Data: 374M triangles

Intel Xeon 2.4GHz 1GB
GeForce FX 5800U AGP8X

Our contributions
GPU-friendly output-sensitive techniques



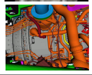
BDAM - Local Terrain Models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
EUROGRAPHICS 2003



P-BDAM - Planetary terrain models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
IEEE Visualization 2003

Adaptive Tetrapuzzles – Dense mesh models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
SIGGRAPH 2004

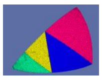
Layered Point Clouds – Dense point clouds
Gobbetti/Marton (CRS4)
SPBG 2004 / Computers & Graphics 2004

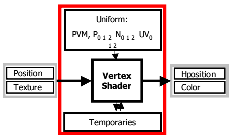


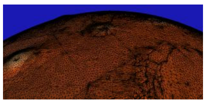
Far Voxels – General
Gobbetti/Marton (CRS4)
[Under review – Stay tuned]

Our contributions
P-BDAM – Planetary terrain models

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



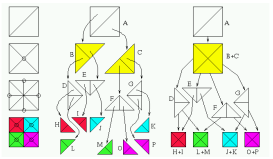


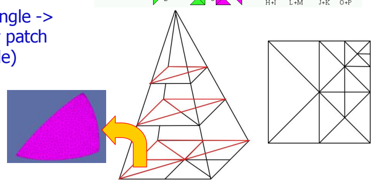


Cignoni, Ganovelli, Gobbetti, Marton, Ponchio, and Scopigno.
Planet-Sized Batched Dynamic Adaptive Meshes (P-BDAM).
In Proceedings IEEE Visualization. Pages 147-155., October 2003.

Our contributions
P-BDAM – Planetary terrain models

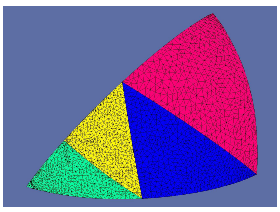
- Geometry multiresolution data structure
  - Pair of triangle bintrees
    - Each triangle is recursively bisected by splitting it along its longest edge
  - Base domain triangle -> curved triangular patch (displaced triangle)

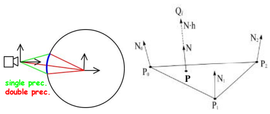




Our contributions
P-BDAM – Planetary terrain models

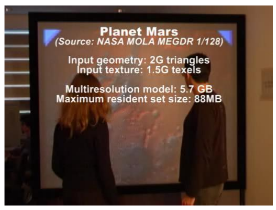
- Displaced triangle
  - General triangle mesh hi-quality adaptively simplified + stripified during preprocessing
  - Takes into account planet curvature / size
    - 3 double precision corner coordinate
    - Mesh vertex positions computed on GPU from parametric coords
  - Preserves connectivity among adjacent levels using matched triangulations
- Global continuity, compression, submetric accuracy on Earth





Our contributions
P-BDAM – Planetary terrain models

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



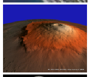
Cignoni, Ganovelli, Gobbetti, Marton, Ponchio, and Scopigno.
Planet-Sized Batched Dynamic Adaptive Meshes (P-BDAM).
In Proceedings IEEE Visualization. Pages 147-155., October 2003.

Planet Mars
Source: NASA MOLA MEGDR
Elevation: 44Kx22K / Color: 44Kx22K

Rendering: 2x1024x768 @ 1px accuracy
Intel Xeon 2.4GHz 1GB
GeForce 6800GT AGP8X

Our contributions
GPU-friendly output-sensitive techniques

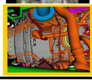
BDAM - Local Terrain Models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
EUROGRAPHICS 2003



P-BDAM - Planetary terrain models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
IEEE Visualization 2003

Adaptive Tetrapuzzles – Dense mesh models
Gobbetti/Marton (CRS4), Cignoni/Ganovelli/Ponchio/Scopigno (ISTI-CNR)
SIGGRAPH 2004

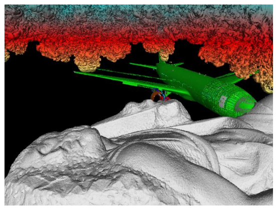
Layered Point Clouds – Dense point clouds
Gobbetti/Marton (CRS4)
SPBG 2004 / Computers & Graphics 2004



Far Voxels – General
Gobbetti/Marton (CRS4)
[Under review – Stay tuned]

Our contributions
Far Voxels – General 3D models

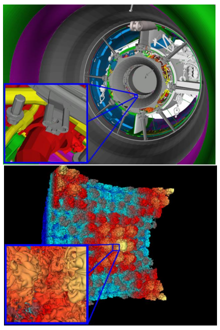
- Far Voxels: High performance visualization of arbitrary 3D models
  - Mixed model
  - Seamless integration of occlusion culling with out-of-core data management and multiresolution rendering
  - … work in progress



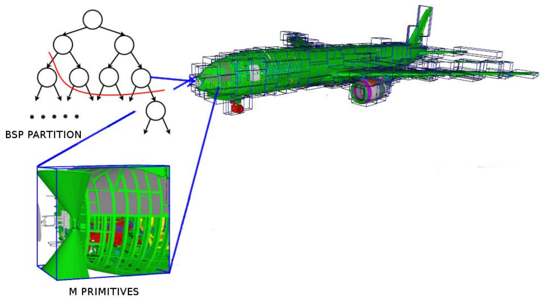
Gobbetti, Marton.
Far Voxels.
Under review (2005).

Our contributions
Far Voxels – General 3D models

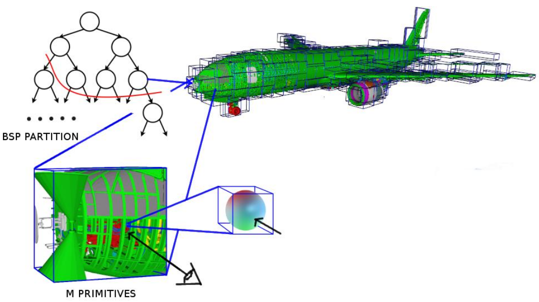
- Classic multiresolution models
  - Error measures on boundary surfaces
  - Visibility culling decoupled from multiresolution
- Hard to apply to models with high detail and complex topology and high depth complexity!



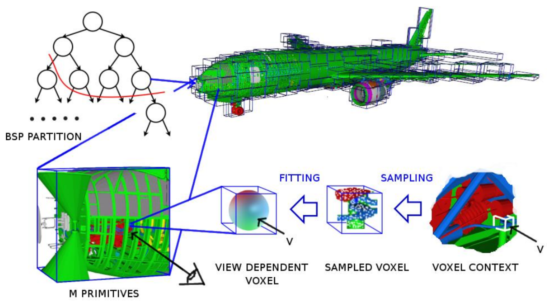
Our contributions
Far Voxels – General 3D models



Our contributions
Far Voxels – General 3D models



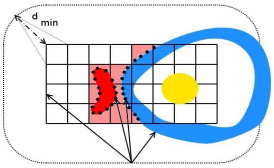
Our contributions
Far Voxels – General 3D models

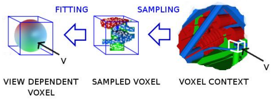


Our contributions

Far Voxels – General 3D models

- Off-line: Reconstruction = sampling + fitting
  - Sampling
    - Raycasting
      - extract n,BRDF = f(ray)
    - Occlusion culling!
      - Sample from distance d_{min} 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



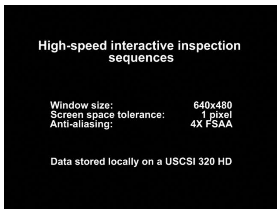


$$Shader_i(v,l) = BRDF_i(v,l)(n(v).l)_+$$

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



Intel Xeon 2.4GHz 1GB
GeForce 6800GT AGP8X

Gobbetti, Marton.
Far Voxels.
Under review (2005).

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**
    - Multi-triangulations
  - **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…