Interactive Walkthroughs using Multiple GPUs

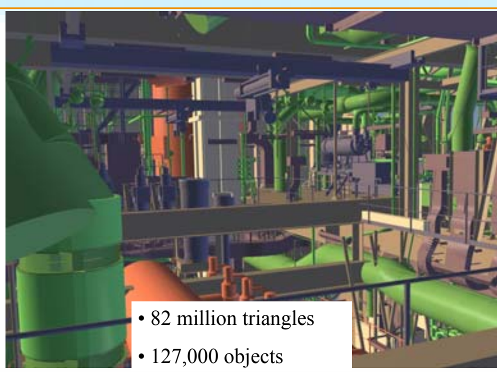
Dinesh Manocha
University of North Carolina-Chapel Hill
http://www.cs.unc.edu/~walk

SIGGRAPH COURSE #11, 2003

Goal

- Interactive Walkthrough of *complex 3D* environments at *high fidelity*
  - Models from CAD, VR
  - High primitive count
  - Heterogeneous geometry
  - Irregular distribution

DoubleEagle Tanker Model



- 82 million triangles
- 127,000 objects

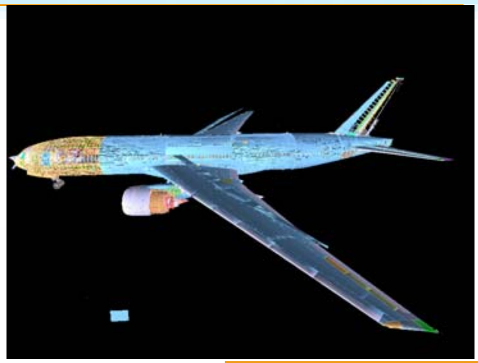
Boeing 777

470 million triangles

Boeing 777

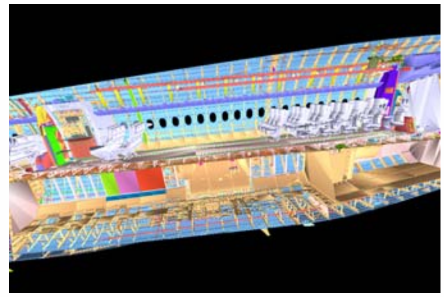
470 million

triangles

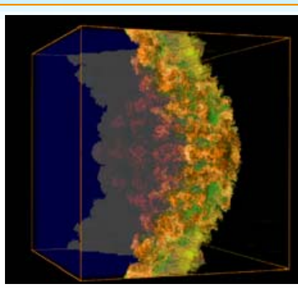


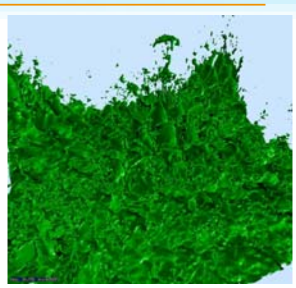
Boeing 777

470 million
triangles:
Zoomed in
View



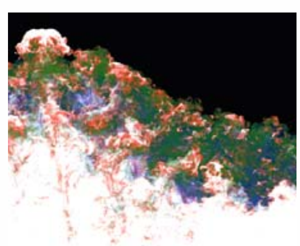
Simulation of Richtmyer-Meshkov instability

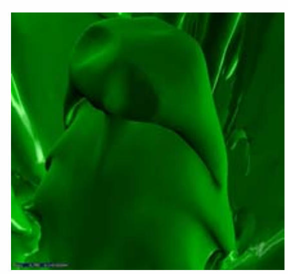




2048 * 2048 * 1920 grid and 27,000 time steps
(5832 processors, IBM SST System):
Over 3 TB of multivariate data generated.
Each time step, ~500 Million triangles for each iso-surface

Simulation of Richtmyer-Meshkov instability





Close view of a small detail
Data: Lawrence Livermore Labs

Commodity Hardware Performance

- Current GPUs have capabilities to render millions of polygons per second
- Require vertex arrays, triangle stripping to achieve these rates
- **Issues**
  - Large polygon count
  - Memory limitations

Commodity Hardware Performance

- Current GPUs have capabilities to render millions of polygons per second
- Require vertex arrays, triangle stripping to achieve these rates
- **Issues**
  - Large polygon count
  - Memory limitations
- **Solution** – Reduce polygon count

Acceleration Techniques

- Geometric simplification
- Occlusion culling
- Hybrid approaches

Geometric Simplification

- Static [Clark 1976; Hoppe 96; Erikson et al. 01; Garland et al. 97]
- View-Dependent [Hoppe 97; Luebke et al. 97]
- Surveyed in LOD book [Luebke et al. 02]

Occlusion Culling

- Cells and portals [Airey et al. 90; Teller 92; Luebke et al. 95]
- Urban datasets composed of large occluders [Coorg et al. 97; Hudson et al. 97; Schaufler et al. 00; Wonka et al. 01]
- Recent survey in [Cohen-Or et al. 02]
- Further classified as
  - Conservative [Greene et al. 93, Zhang et al. 97]
  - Approximate [Bartz et al. 99; Klosowski and Silva 00]

Hybrid Approaches

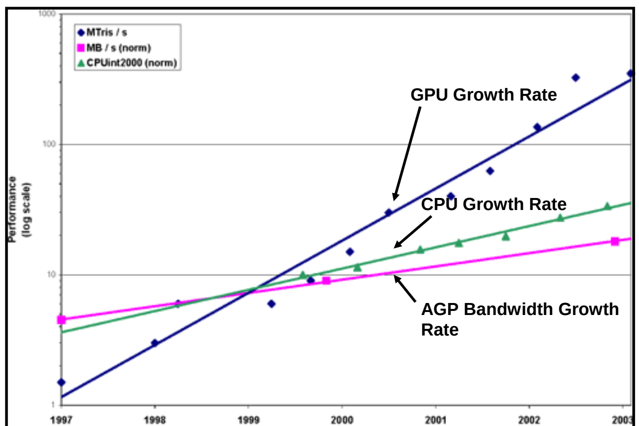
- Combine LOD and Occlusion Culling techniques
  - UC Berkeley Walkthrough [Funkhouser et al. 96]
  - Synthetic convex occluders [Andujar et al. 00]
  - Approximate visibility using prioritized layer projections with view dependent rendering [ElSana et al. 01]
  - UNC MMR system [Aliaga et al. 99]
  - GigaWalk [Baxter et al. 02]

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Motivation

- An algorithm that works on generic models
- Image-based culling algorithms
- Current approaches involve frame buffer readbacks [Greene et al. 93, Zhang et al. 97, Baxter et al. 02]

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL



The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Frame Buffer Readbacks

- GPU growth rate is higher than Moore's law
- Bandwidth growth rate lower than Moore's law
  - Result – Frame buffer readbacks are expensive
  - Also readbacks result in graphics pipeline stalls
- Require a solution that
  - Leverages high GPU power
  - Avoids frame buffer readbacks

Outline

- Parallel algorithm using mulitple GPUs
- Interactive walkthroughs
- Interactive shadow generation

Outline

- **Parallel algorithm using mulitple GPUs**
- Interactive walkthroughs
- Interactive shadow generation

Occlusion-Switch

- Occlusion-switch has 2 GPUs
  - It takes a camera as input
  - The output is PVS and a camera

PVS: Potentially visible set

Occlusion-Switch

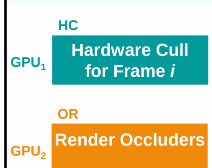
- Parallel image-based culling algorithm
- A novel switching mechanism to eliminate frame-buffer readbacks
- Exploits temporal coherence
  - ♦ Occluder selection
  - ♦ Lower bandwidth requirements
- Integrates with hierarchical LODs

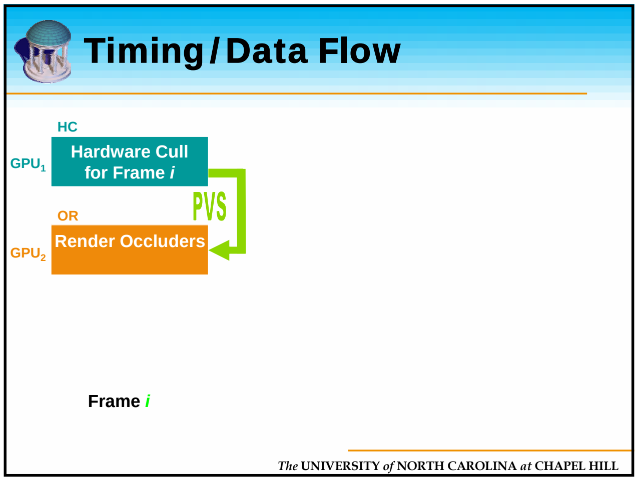
Occlusion Switch : Parallel Algorithm and Architecture

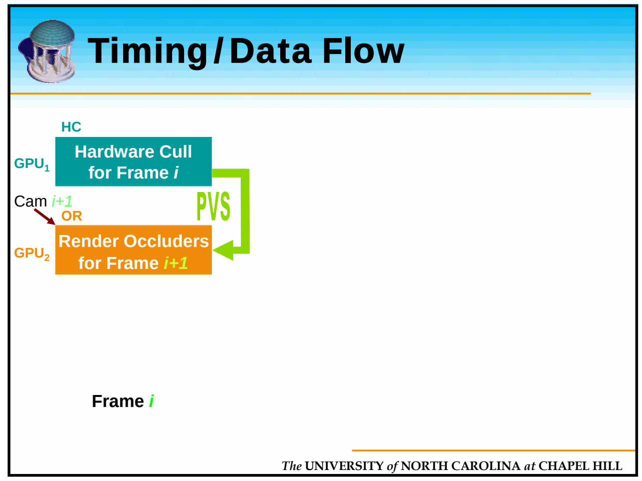
- **Three processes in parallel**
  1. **Occlusion Representation (OR)**
  2. **Hardware Culling (HC)**
  3. **Render Visible Geometry (RVG)**

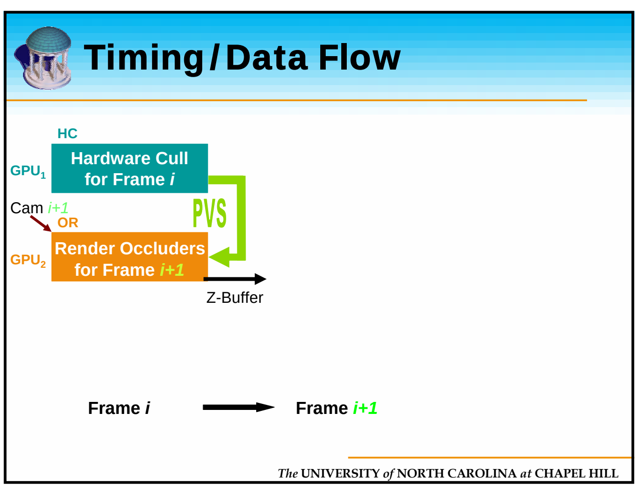
Timing / Data Flow

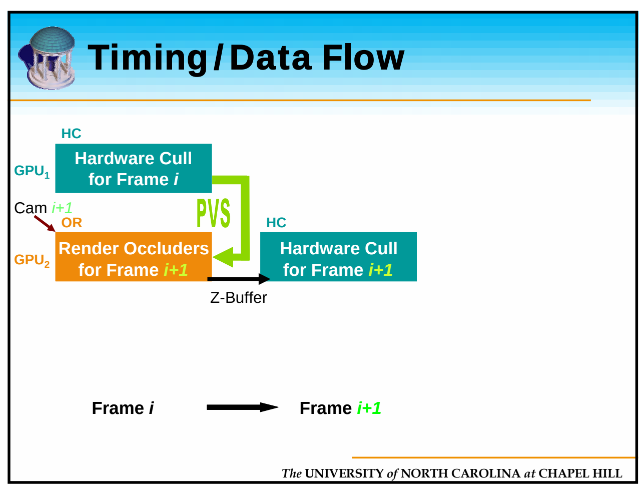
Timing / Data Flow

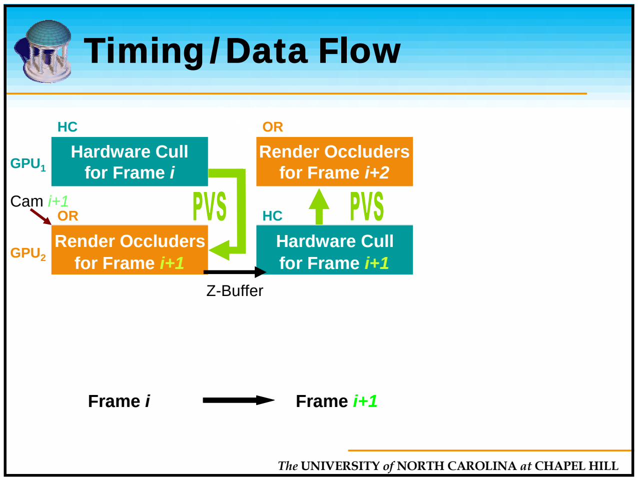


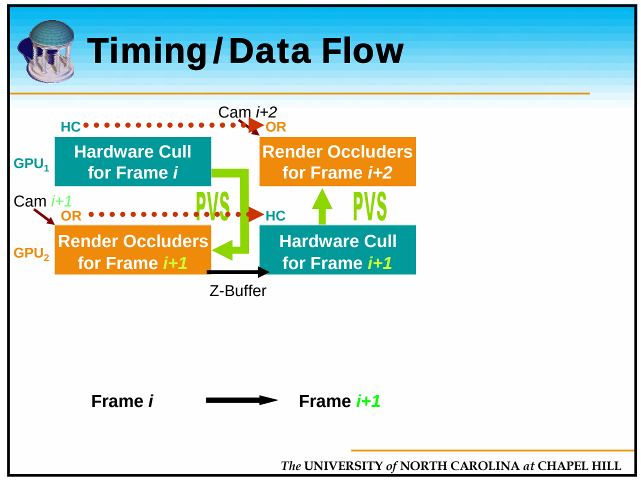

Timing / Data Flow
HC
GPU1
Hardware Cull for Frame i
PVS
OR
GPU2
Render Occluders
Frame i
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

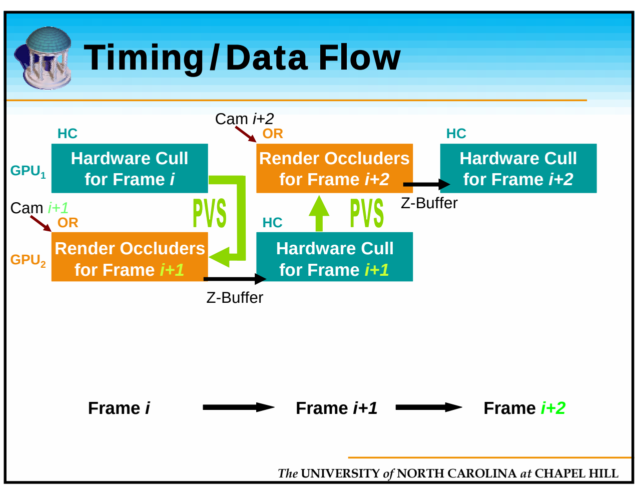

Timing / Data Flow
HC
GPU1
Hardware Cull for Frame i
Cam i+1
PVS
OR
GPU2
Render Occluders for Frame i+1
Frame i
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

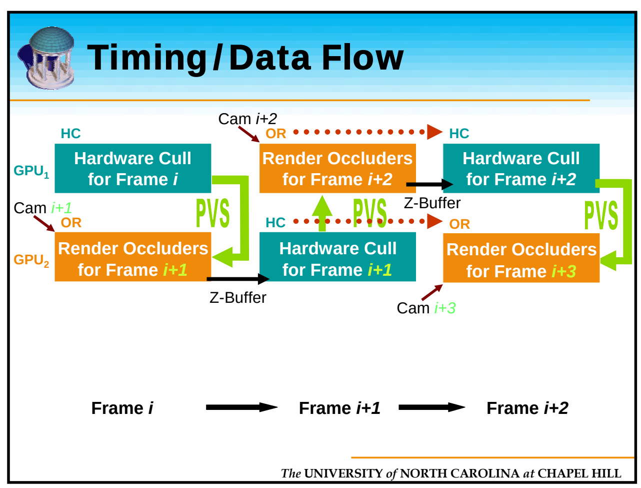

Timing / Data Flow
HC
GPU1
Hardware Cull for Frame i
Cam i+1
PVS
OR
GPU2
Render Occluders for Frame i+1
Z-Buffer
Frame i
Frame i+1
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

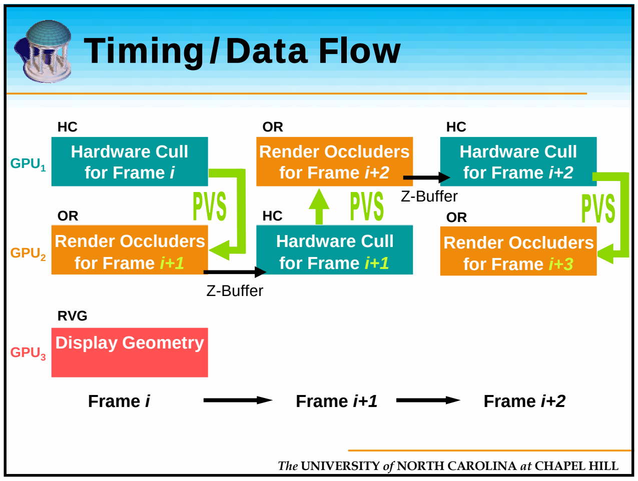

Timing / Data Flow
HC
GPU1
Hardware Cull for Frame i
Cam i+1
PVS
OR
HC
GPU2
Render Occluders for Frame i+1
Hardware Cull for Frame i+1
Z-Buffer
Frame i
Frame i+1
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

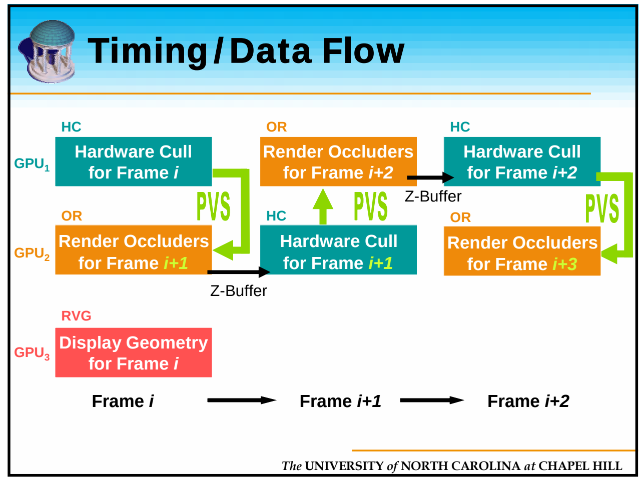

Timing / Data Flow
HC
OR
GPU1
Hardware Cull for Frame i
Render Occluders for Frame i+2
Cam i+1
OR
PVS
HC
PVS
GPU2
Render Occluders for Frame i+1
Hardware Cull for Frame i+1
Z-Buffer
Frame i
Frame i+1
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

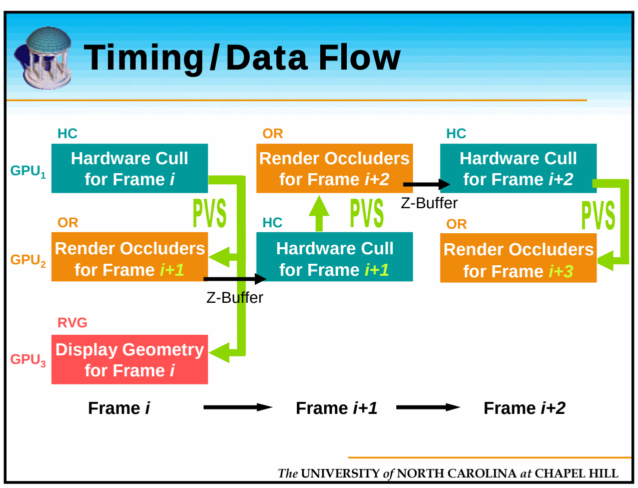

Timing / Data Flow
Cam i+2
HC
OR
GPU1
Hardware Cull for Frame i
Render Occluders for Frame i+2
Cam i+1
OR
PVS
HC
PVS
GPU2
Render Occluders for Frame i+1
Hardware Cull for Frame i+1
Z-Buffer
Frame i
Frame i+1
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

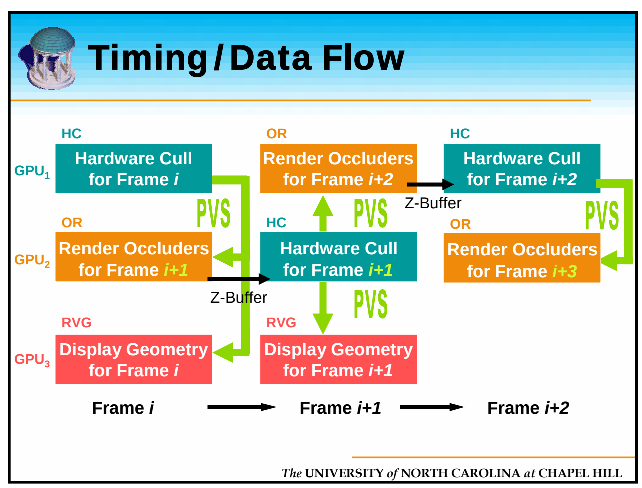

Timing / Data Flow
Cam i+2
HC
OR
HC
GPU1
Hardware Cull for Frame i
Render Occluders for Frame i+2
Hardware Cull for Frame i+2
Z-Buffer
Cam i+1
OR
PVS
HC
PVS
GPU2
Render Occluders for Frame i+1
Hardware Cull for Frame i+1
Z-Buffer
Frame i
Frame i+1
Frame i+2
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


Timing / Data Flow
Cam i+2
HC
OR
HC
GPU1
Hardware Cull for Frame i
Render Occluders for Frame i+2
Hardware Cull for Frame i+2
Z-Buffer
Cam i+1
OR
PVS
HC
PVS
OR
PVS
GPU2
Render Occluders for Frame i+1
Hardware Cull for Frame i+1
Render Occluders for Frame i+3
Z-Buffer
Cam i+3
Frame i
Frame i+1
Frame i+2
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

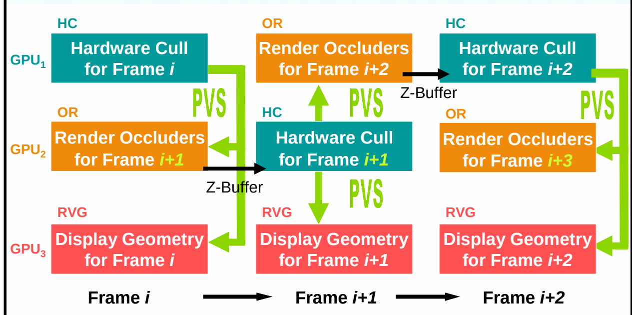

Timing / Data Flow
HC
Hardware Cull for Frame i
OR
Render Occluders for Frame i+2
HC
Hardware Cull for Frame i+2
GPU1
PVS
Z-Buffer
OR
Render Occluders for Frame i+1
HC
Hardware Cull for Frame i+1
OR
Render Occluders for Frame i+3
GPU2
Z-Buffer
RVG
Display Geometry
GPU3
Frame i
Frame i+1
Frame i+2
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


Timing / Data Flow
HC
Hardware Cull for Frame i
OR
Render Occluders for Frame i+2
HC
Hardware Cull for Frame i+2
GPU1
PVS
Z-Buffer
OR
Render Occluders for Frame i+1
HC
Hardware Cull for Frame i+1
OR
Render Occluders for Frame i+3
GPU2
Z-Buffer
RVG
Display Geometry for Frame i
GPU3
Frame i
Frame i+1
Frame i+2
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

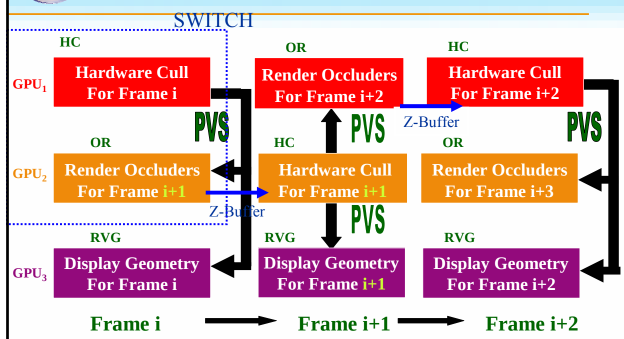

Timing / Data Flow
HC
Hardware Cull for Frame i
OR
Render Occluders for Frame i+2
HC
Hardware Cull for Frame i+2
GPU1
PVS
Z-Buffer
OR
Render Occluders for Frame i+1
HC
Hardware Cull for Frame i+1
OR
Render Occluders for Frame i+3
GPU2
Z-Buffer
RVG
Display Geometry for Frame i
GPU3
Frame i
Frame i+1
Frame i+2
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL


Timing / Data Flow
HC
Hardware Cull for Frame i
OR
Render Occluders for Frame i+2
HC
Hardware Cull for Frame i+2
GPU1
PVS
Z-Buffer
OR
Render Occluders for Frame i+1
HC
Hardware Cull for Frame i+1
OR
Render Occluders for Frame i+3
GPU2
Z-Buffer
RVG
Display Geometry for Frame i
RVG
Display Geometry for Frame i+1
GPU3
Frame i
Frame i+1
Frame i+2
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Timing / Data Flow



Timing / Data Flow



The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

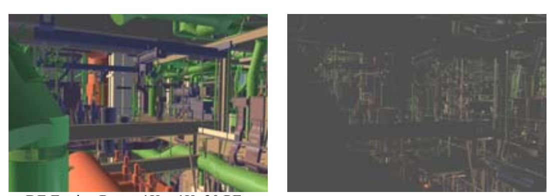
Bandwidth Requirements

- Potential Visible Set (PVS) and camera parameters are transmitted over the network.
- Temporal coherence to minimize bandwidth requirements
  - Each GPU maintains current frame geometry
  - Transmit the differences in visible geometry across successive frames

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Static LOD Selection: Error Metric

- Pixel Error Metric: Max normal deviation of silhouette in image
- Traverse down scene graph till error satisfied
- Upper Bound: Highly conservative



DE Engine Room 1K x 1K, 20 PE

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Hardware Culling

- Use NV_OCCLUSION_QUERY to determine visible pixels
- Traverse scene hierarchy rendering bounding boxes of nodes

Hardware Culling Optimizations

- **Multiple Occlusion Tests**
  - Occlusion Query 'counter' for each node
  - Traverse scene graph breadth first
  - Bunch queries for all nodes at a level
  - 40% faster than testing one node with HP_OCCLUSION_TEST

LOD Selection: Visibility Metric

- **Visibility for LOD selection**
  - Visible pixels of bounding box > visible pixels of geometry
  - No. of visible pixels less than error metric $\vec{}$ provides an early termination condition
  - Provides looser bounds – reduces polygon count

Conservative Occlusion Culling

- Underlying algorithm used for occlusion culling is conservative to image precision
- Exactly same set of LODs is used for both OR and HC stages

Outline

- Parallel algorithm using mulitple GPUs
- **Interactive walkthroughs**
- Interactive shadow generation

Interactive Walkthroughs

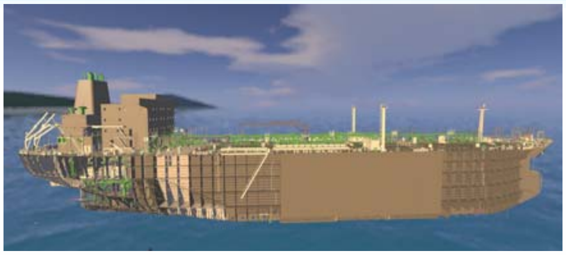
- 3 Dell Precision Workstations with dual 2 GHz Pentium IV CPUs, GeForce 4 GPU, and 2 GB main memory**: Immediate mode rendering**

Test Model: Powerplant

Original
- 0.5 Gigabyte dataset
- 13 Million Polygons
- 1200 objects

Preprocessing
- 7 hours
- 1.2 Gigabytes
- 13 Million Polygons
- 38,000 objects

Test Model: DoubleEagle Tanker



Original
- 4 Gigabyte dataset
- 82 Million polygons
- 127,000 objects

Preprocessing
- 34 hours
- 8 Gigabytes
- 82 Million polygons
- 61,000 objects

Interactive Walkthrough: Powerplant

13M triangles: 3 Dell PCs with NVIDIA GeForce 4 cards (2002)

Interactive Walkthrough: Double Eagle Tanker

82M triangles: 3 Dell PCs with NVIDIA GeForce 4 cards (2002)

Recent Work: Use Single GPU for 3 Processes

- Use vertex arrays
- Perform OR, HC and RVG on the GPU
- Improved support for occlusion queries
- Lower performance as compared to 3 GPU-based solution
- Fewer latency problems

[July 2003]

Live Demo

Dell M50 laptop, 2.4GHz Pentium IV-M CPU, NVIDIA Quadro4 700GoGL GPU, 1GB memory running Windows XP**:** Vertex arrays

Powerplant model (13M triangles)

View-Dependent Rendering

- Use of static LODs leads to popping artifacts
- Compute a view-dependent simplification
  - ♦ Precompute a vertex and cluster hierarchy
  - ♦ Runtime: perform occlusion culling using cluster hierarchy
  - ♦ Runtime: use the vertex hierarchy to compute the view-dependent simplification

View-Dependent Rendering

- Use of static LODs leads to popping artifacts
- Compute a view-dependent simplification
  - ♦ Precompute a vertex and cluster hierarchy
  - ♦ Runtime: perform occlusion culling using cluster hierarchy
  - ♦ Runtime: use the vertex hierarchy to compute the view-dependent simplification

The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Integrated Occlusion Culling with View-Dependent Simplification

Single Dell PC with NVIDIA GeForce FX 5800 Ultra GPU
[Yoon, Salomon and Manocha: IEEE Visualization 2003]

Occlusion-Switch : Advantages

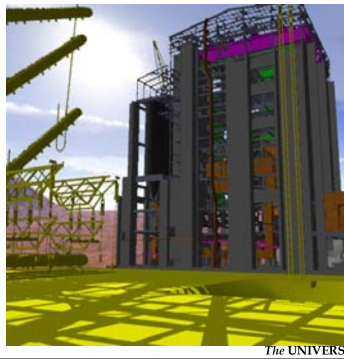
- Generic models
  - ♦ No assumptions on model, distribution
- Conservative Occlusion Culling
- Significant Occlusion Culling
- Low Bandwidth
- Computation performed on GPUs

Outline

- Parallel algorithm using mulitple GPUs
- Interactive walkthroughs
- **Interactive shadow generation**

The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Interactive Shadows

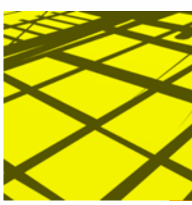


Portions of scene that are occluded from light

The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Goal

- Generate hard-edged shadows for interactive walkthroughs
  - Models with millions of polygons
  - Scenes with a moving light source



Hard-edged shadows

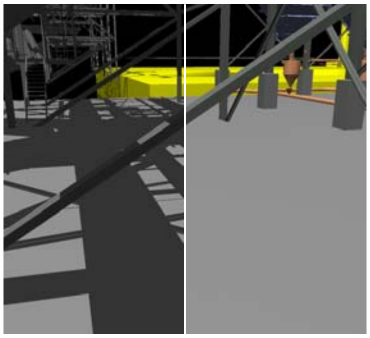
The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Shadows: Importance

- Realism
  - Real world scenes have shadows
- Provide a sense of depth to the user
  - Better navigation
  - Enhance user experience in interactive games, VR applications

The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Shadows: Importance



The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Interactive Shadows

N. Govindaraju et al. "Interactive Shadow Generation in Complex Environments", Proc. of ACM SIGGRAPH 2003

Talk scheduled on Tuesday at 3:40pm (Shadows section)

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Goal

- Generate high fidelity shadows
  - Requires object precision techniques
  - Compute shadow casters and shadow receivers

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Goal

- Interactive performance
  - Use visibility culling and LOD techniques
  - Reduce the number of potential shadow casters and shadow receivers

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

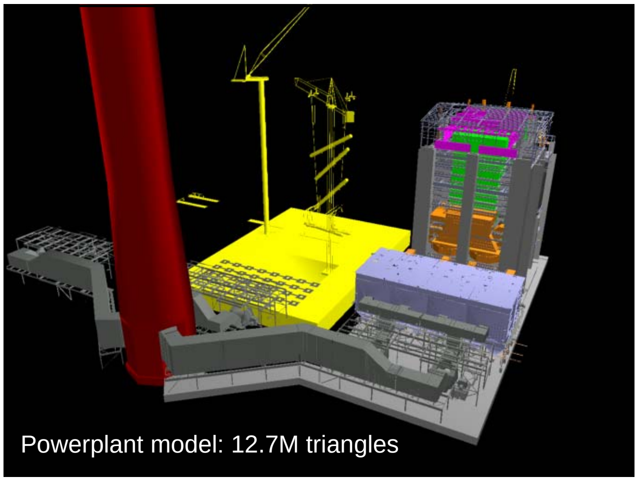
Interactive Shadows

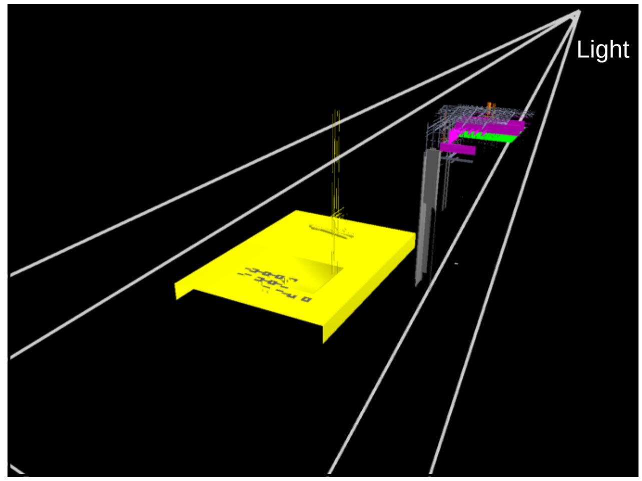
Interactive Shadows

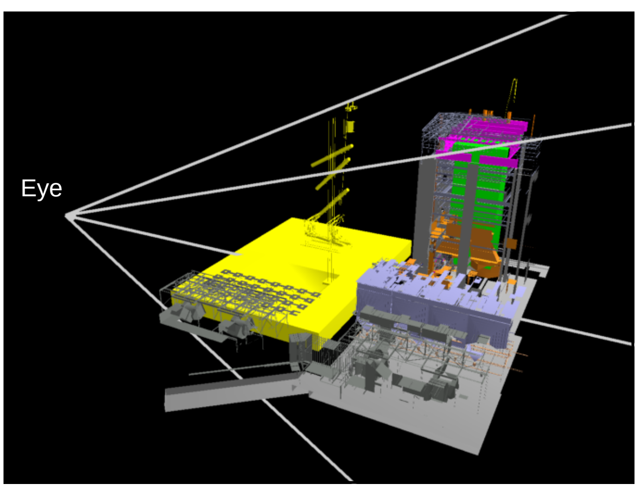
Interactive Shadows

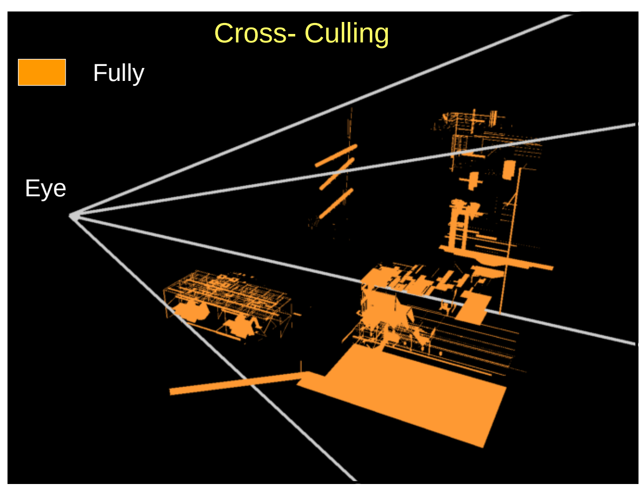
Overview

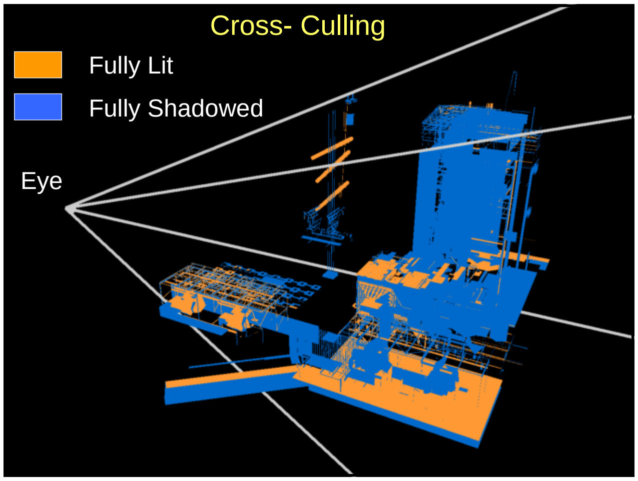
- Algorithm proceeds in three stages
- Stage I: PVS computation
  - Compute potential shadow casters and receivers
- Stage II: Cross Culling
  - Compute shadow casters and receivers that result in aliasing
- Stage III: Hybrid Shadow Generation

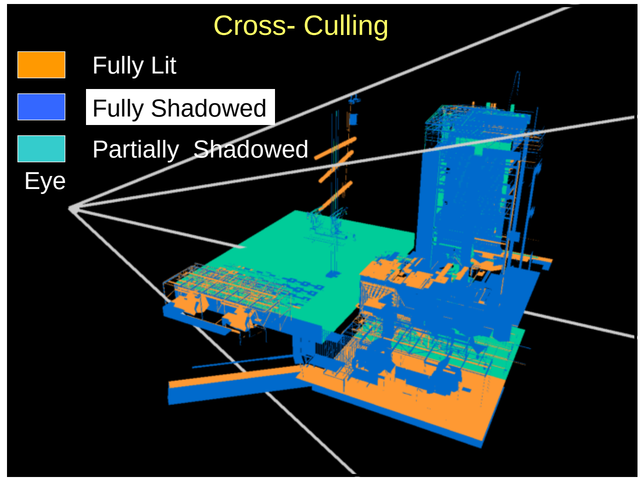

Powerplant model: 12.7M triangles

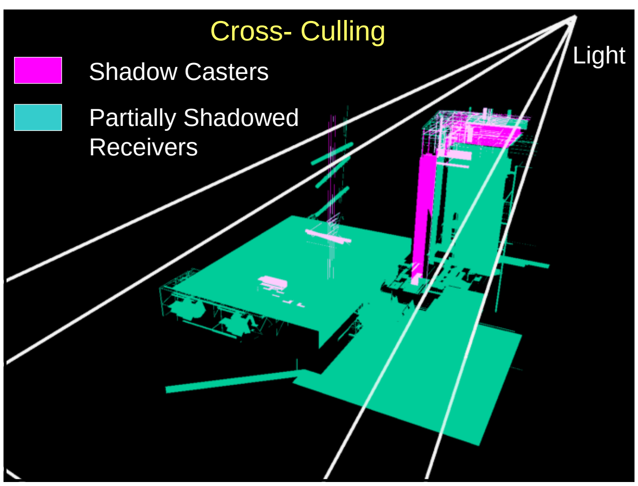

Light

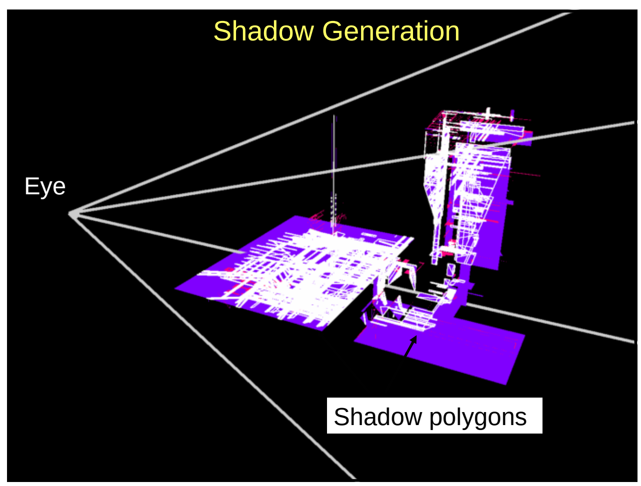

Eye

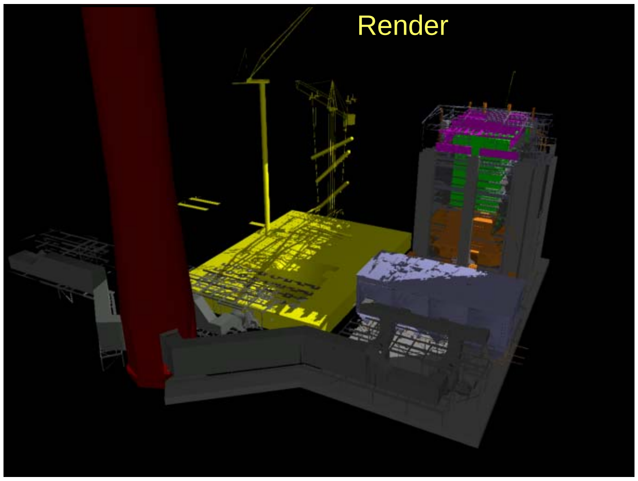

Cross- Culling
Fully
Eye


Cross- Culling
Fully Lit
Fully Shadowed
Eye


Cross- Culling
Fully Lit
Fully Shadowed
Partially Shadowed
Eye


Cross- Culling
Shadow Casters
Partially Shadowed
Receivers
Light


Shadow Generation
Eye
Shadow polygons

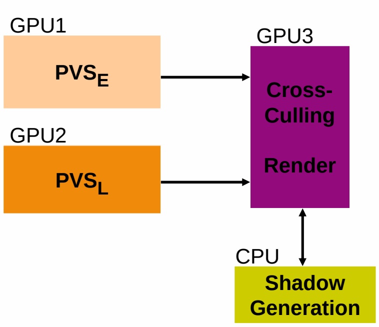


Process-parallel Algorithm

Utilize multiple GPUs to improve performance

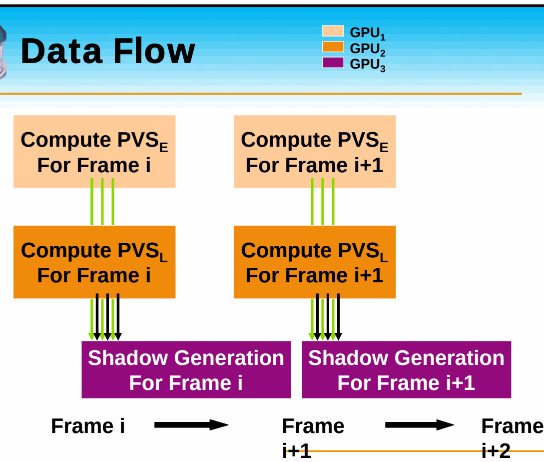
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Architecture



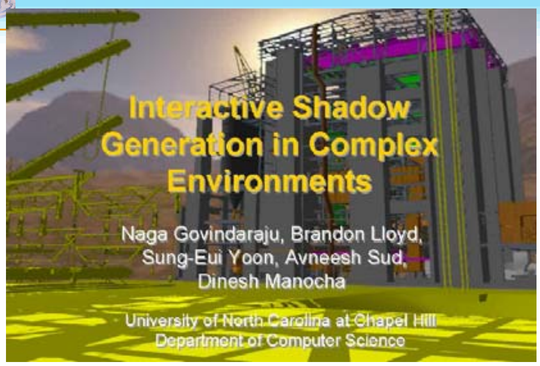
The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Data Flow



The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Interactive Shadows



Conclusions

- Interactive display and shadows
- Massive models
- Use multiple GPUs for visibility computations
- Application to complex models

Collaborators

- Naga Govindaraju
- Brandon Lloyd
- Brian Salomon
- Avneesh Sud
- Sungeui Yoon

http://www.cs.unc.edu/~walk

Acknowledgements

- Army Research Office
- Department of Energy
- Intel Corporation
- Office of Naval Research
- National Science Foundation
- NNS for the DoubleEagle model
- Anonymous donor for Powerplant model
- Boeing Corporation
- NVIDIA Corporation (Occlusion queries)
- UNC Walkthrough group

The End

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL