Parallel Visualization on Display Wall

Pacific Rim Applications and PRAGMA Institute

Outline

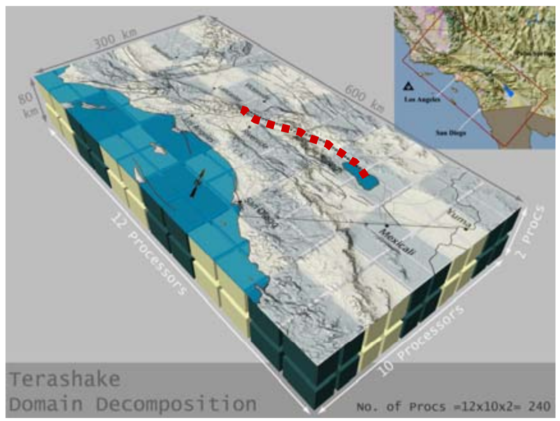
- Problem Statement & Motivation
- Large-Scale Rendering
  - Sorting classification (sort-first, sort-middle, sort-last)
  - Sort-first rendering (Visualization on HIPerWall)
  - Sort-last rendering (Distributed Parallel Terrain Rendering)
- Conclusions

Problem Statement

- Size of datasets are growing rapidly
  - scientific simulations
  - medical imaging
  - hundreds of millions of data points and polygons

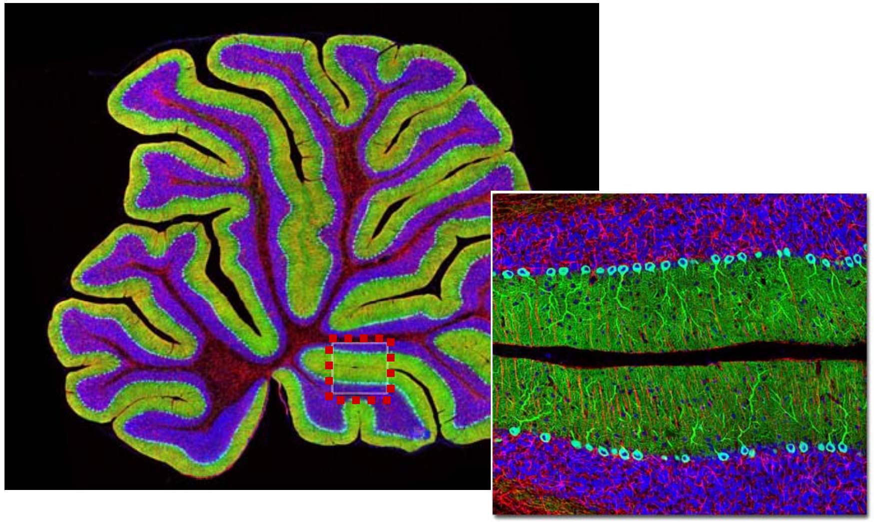
Large Data Example - Terashake

- Simulate ruptured a magnitude 7.7 earthquake
- Simulate 230 km section of the San Andreas fault
- Simulate 600x300x80 km region in southern CA
- 3000x1500x400 mesh (1.8 billion cubes) (200 meters resolution)
- Simulate 3 minutes, 20,000 time steps, Δt = 0.011 sec
- 240 processors on SDSC DataStar
- 5 days, 20,000 CPU hours
- Yielding 47 TB of data



http://users.sdsc.edu/~amit/web/viz/terashake_quake

Rat Cerebellum (300 Megapixel Image)



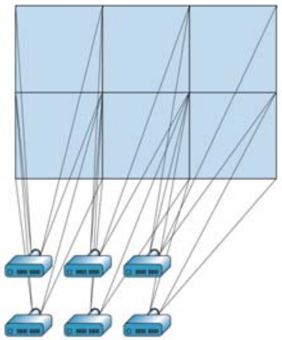
The National Center for Microscopy and Imaging Research http://www.ncmir.ucsd.edu

Motivation

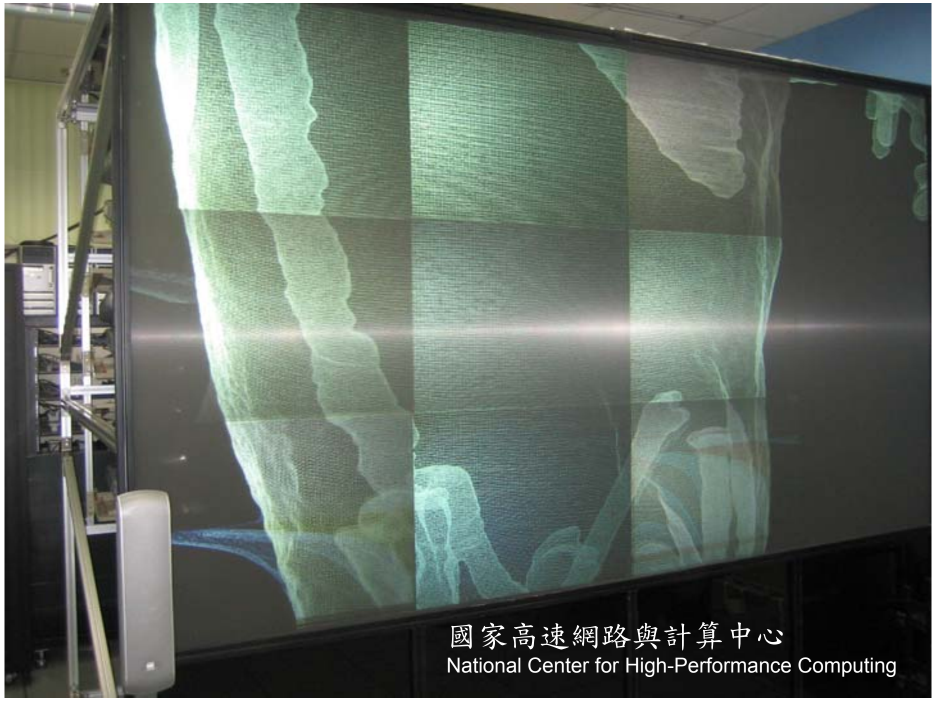
- Large datasets presents challenges
  - How to render this many polygons when the highest resolution of a 30-inch LCD is only 4 mega-pixel (2560x1600)?
  - How to render them interactively?
  - Need to render large amount of geometry faster than high-end graphics system

Solution to Display-Resolution Problem

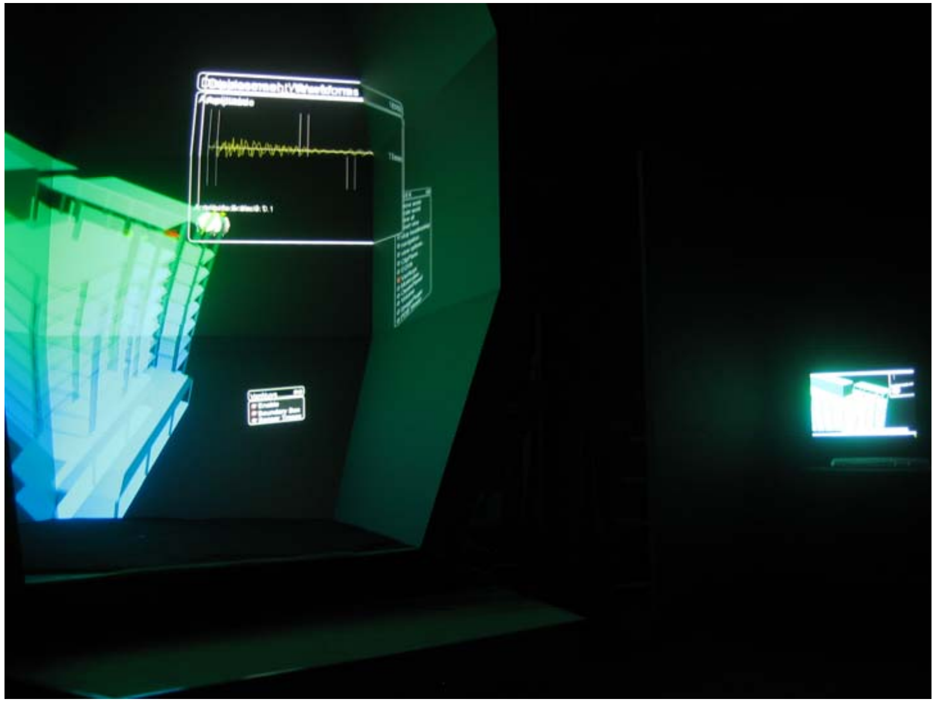
- Build a large display wall with:
  - an array of projectors (projectors suffer from edge light taper problem)
  - an array of LCD panels (LCDs suffer from small separations between panels, windowing effects)

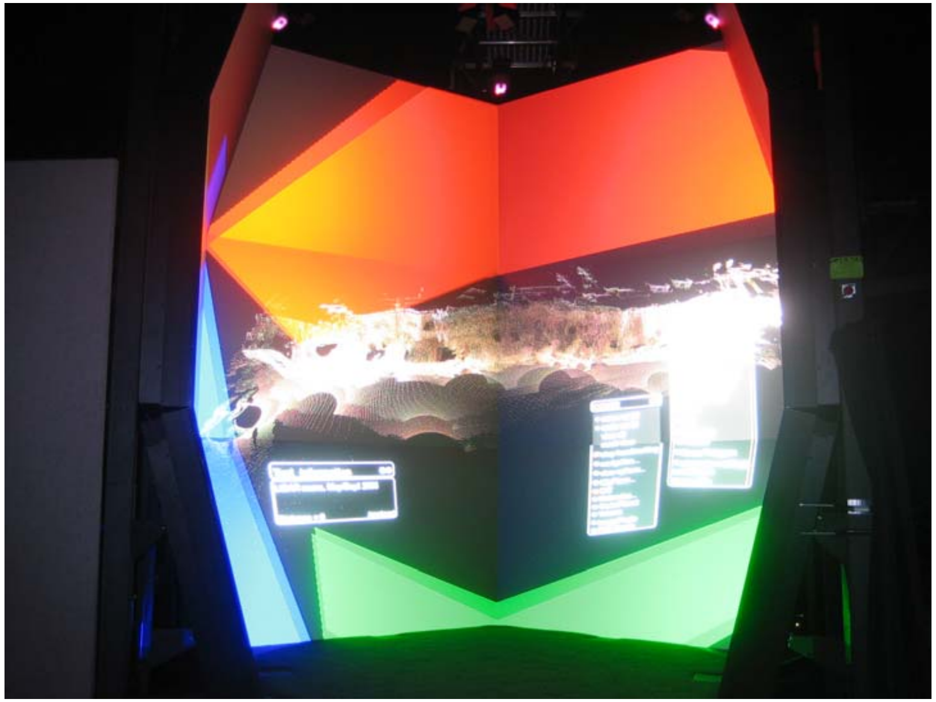


[Angel 2006]


國家高速網路與計算中心
National Center for High-Performance Computing


StarCAVE (Cave Automated Virtual Environment)
[Calit2 UCSD]





Advantage of LCDs

- Advantage
  - color correction is easier
  - less expensive
  - easy to setup, take less space, flexibility layout
  - offer higher resolution in unit area than projectors
- Disadvantage
  - has borders between each tile
  - the highest resolution of a 30-inch LCD is 2560x1600

Advantage of Projectors

- Advantage
  - highest resolution projector (SONY SRX-R105) is 4096x2160
  - large area display, achieve fully immersive
  - seamless
- Disadvantage
  - expensive and high maintaining cost
  - high power consumption
  - noise

Solution to Display-Resolution Problem

- Use clusters of computers
  - connected with network
  - each computer has its graphics hardware
  - advantage is low cost
  - achieve high-performance computing
- There are multiple ways to distribute the work that must be done to render a scene among the processors

Outline

- Problem Statement & Motivation
- Large-Scale Rendering
  - Sorting classification (sort-first, sort-middle, sort-last)
  - Sort-first rendering (Visualization on HIPerWall)
  - Sort-last rendering (Distributed Parallel Terrain Rendering)
- Conclusions

High-Level View of the Graphics Process

- Input: 3D vertices
- Output: 2D pixels

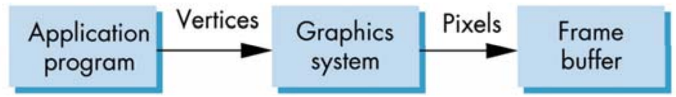


Fig: Graphics Process. (Source: [Angle 2006])

Tasks of Graphics System

- A commodity card with a single GPU as a combination of one geometry processor and one raster processor

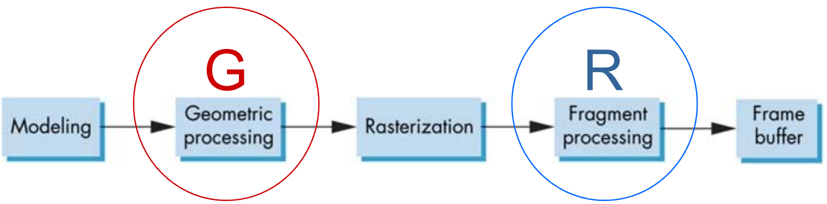


Fig: Graphics Process. (Source: [Angle 2006])

- Transformations
- Polygon clipping
- Backface culling
- Shading
- Viewport mapping

- Scan conversion
- Texture
- Fog

Visual Description of Graphics Pipeline

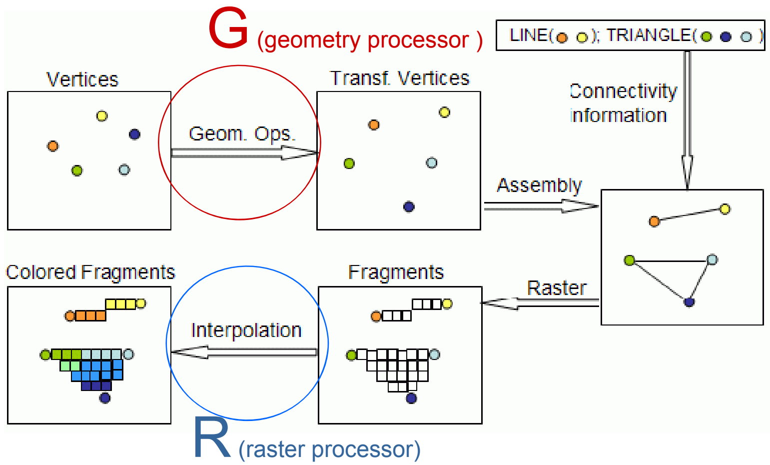


Fig: Visual description of the pipeline stages of a graphics system. (Source: [Lighthouse3D])

Three Possibilities to Distribute Jobs

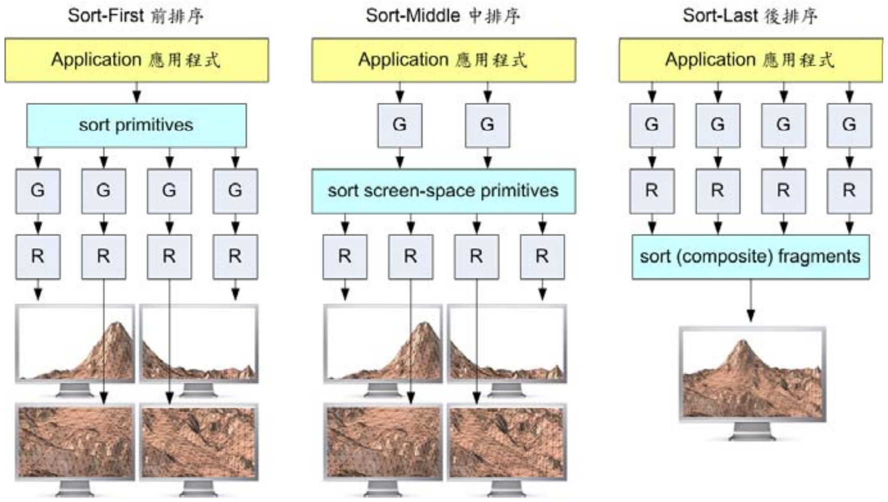
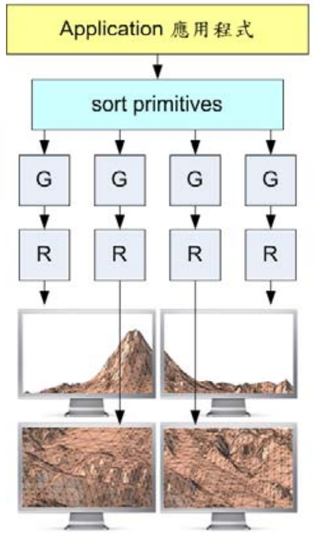
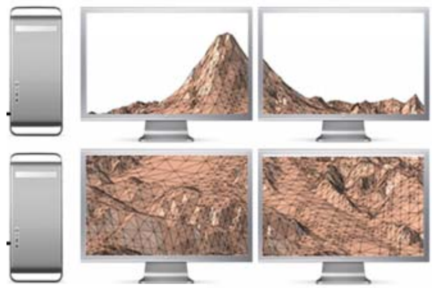


Fig: Sorting classification. [Molnar et al. 1994]

Sort-First Rendering

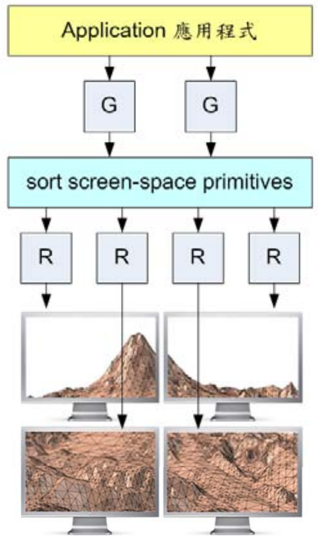




Sort-First Rendering

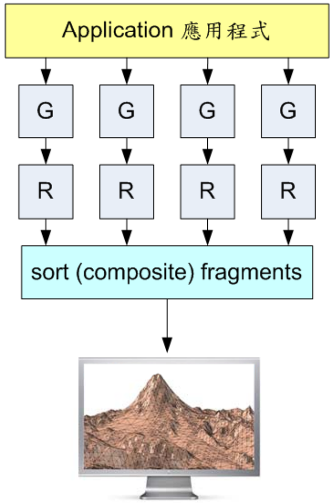
- Pair geometric and raster processors and use standard PCs with standard graphics cards
- Assign a separate portion of the display to each PC
- Front-end sort to make assignment as to which primitives go to which PC
- If a primitive straddles more than one region of the display, it can be sent to multiple geometry processors
- Load-balancing is not addressed
- It is ideally suited for generating high-resolution displays

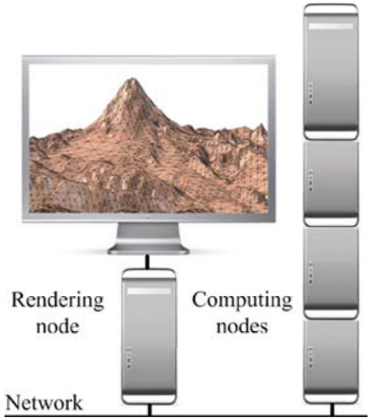
Sort-Middle Rendering



- High-end graphics workstations with special hardware and fast internal buses
- An application generates a large number of geometric primitives
- Sort the outputs of the geometry processors and assign primitives to the correct raster processors
- Load balancing
  - Assign each raster processors to a different region of the frame buffer

Sort-Last Rendering





Sort-Last Rendering

- Each geometry processor is connected to its own raster processor (as standard PCs, each with its own graphics card)
- Each raster processor must have a frame buffer that is the full size of the display
- Each pair produces a correct hidden-surface-removed image for part of the geometry
- Combine the partial images with a compositing step
- Need both information in the color buffers and the depth-buffer

Example of Sort-Last Rendering

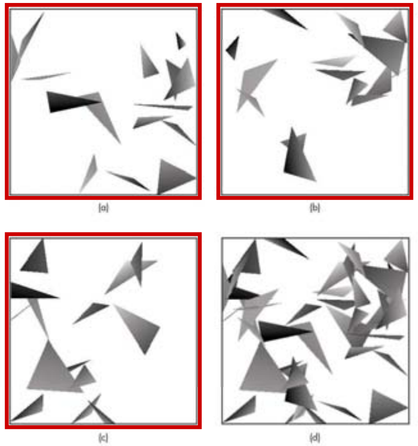


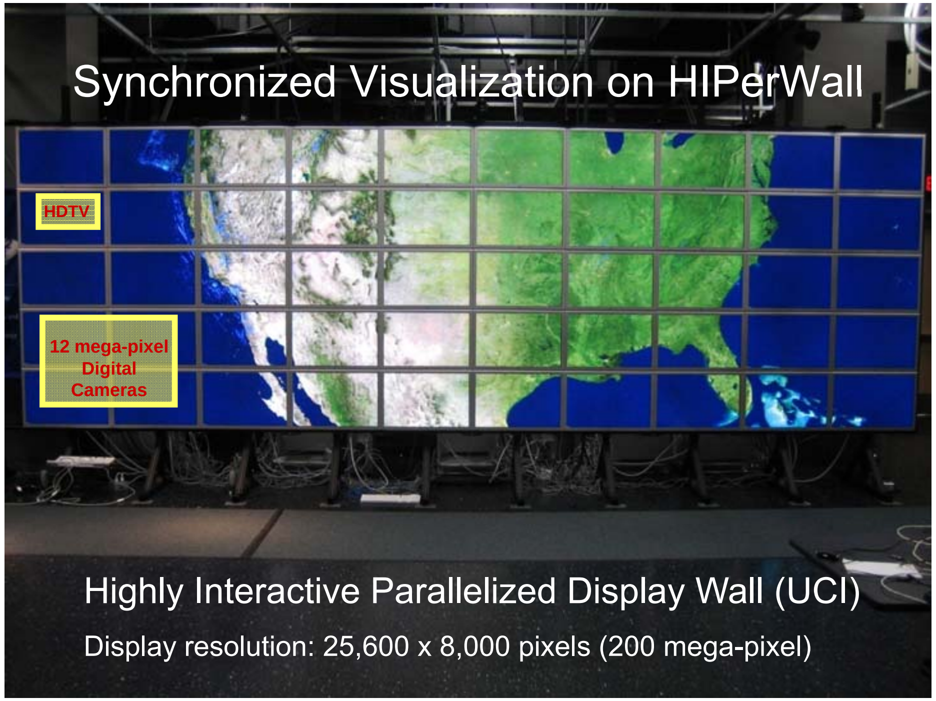
Fig: (a)–(c) Partial renderings, each of which has a correct hidden-surface-removed image for part of the geometry . (d) Composited image. [Angel, 2006]

Outline

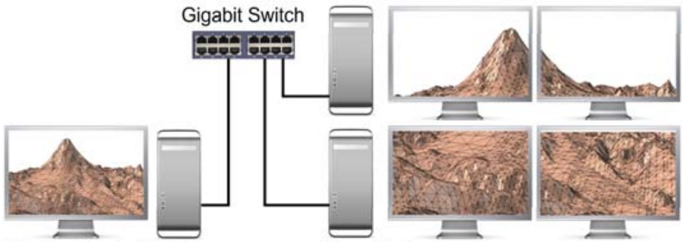
- Problem Statement & Motivation
- Large-Scale Rendering
  - Sorting classification (sort-first, sort-middle, sort-last)
  - Sort-first rendering (Visualization on HIPerWall)
  - Sort-last rendering (Distributed Parallel Terrain Rendering)
- Conclusions

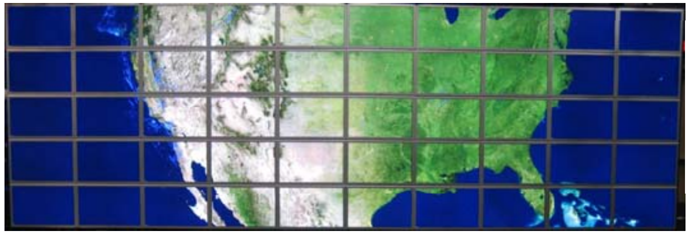
Display Wall

- HIPerWall
  - Number of tiles: 50 (30-inch LCDs)
  - Resolution: 25,600 x 8,000 pixels (200 mega-pixel)
- HIPerSpace
  - Number of tiles: 70 (30-inch LCDs)
  - Resolution: 35,840 x 8,000 pixels (286 mega-pixel)
- HIPerDisplay
  - Number of tiles: 20 (24-inch LCDs)
  - Resolution: 9,600 x 4,800 pixels (46 mega-pixel)

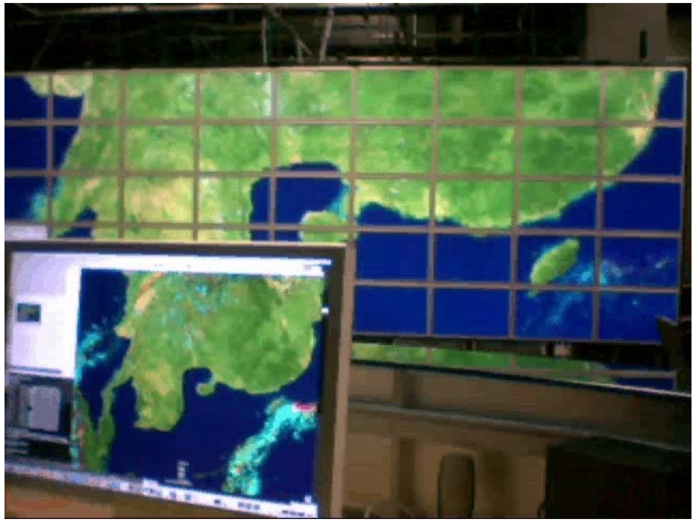


Visualization on HIPerWall





Visualization on HIPerWall

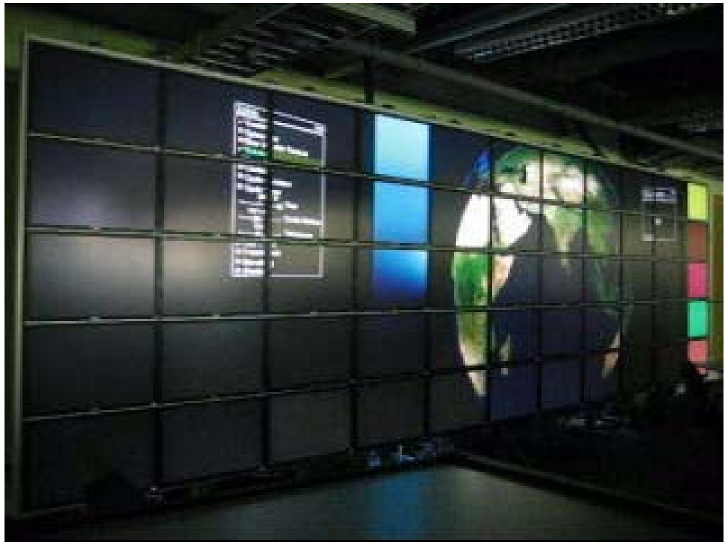


Synchronized Visualization on HIPerSpace

Highly Interactive Parallelized Display Space (UCSD)

Display resolution: 35,840 x 8,000 pixels (286 mega-pixel)

Visualization on HIPerSpace

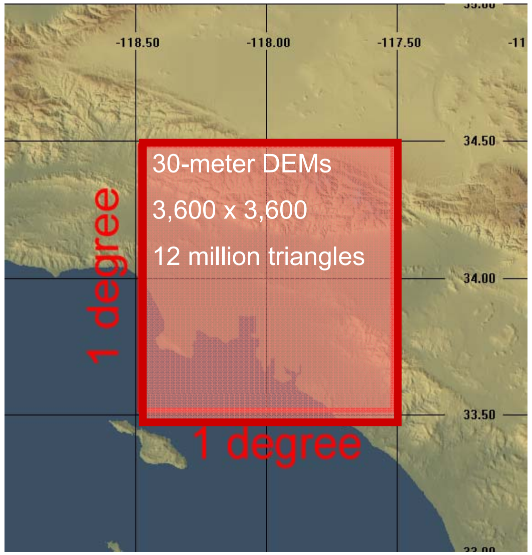


HiPerDisplay

Outline

- ☐ Problem Statement & Motivation
- ☐ Large-Scale Rendering
  - Sorting classification (sort-last, sort-middle, sort-first)
  - Sort-first rendering (Visualization on HIPerWall)
  - Sort-last rendering (Distributed Parallel Terrain Rendering)
- ☐ Conclusions

Digital Elevation Models (DEMs)



Entire United States

216,000 x 84,000

180 billion triangles

Traditional Methods Do not Scale

- ☐ Out-of-core processing
  - reorganize data layout, however, size increased dramatically
- ☐ Memory is the bottleneck
  - block size must be 2^n+1 x 2^n+1, n = 0,1,2,3,…
  - limitation: up to 4,097 x 4,097

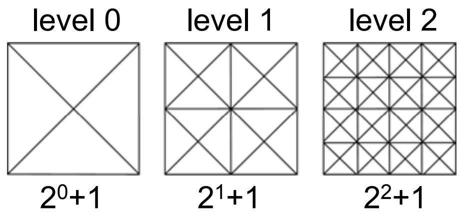
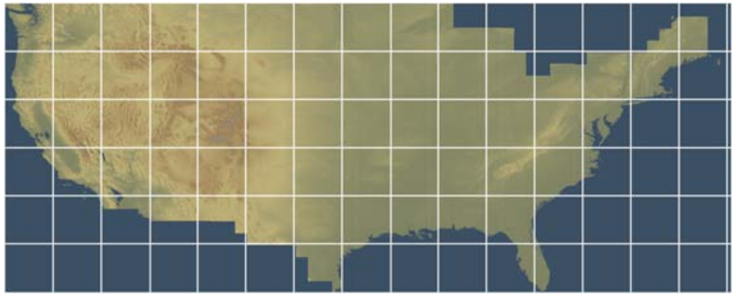


Table: Required memory size.

n	Dimension	Memory Size
11	2049x2049	214 (MB)
12	4097x4097	854
13	8193x8193	3,414

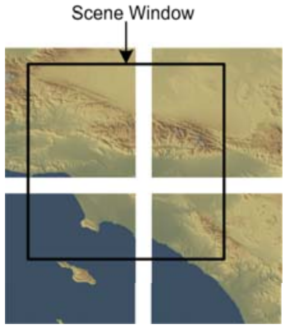
Divide Terrain into Blocks

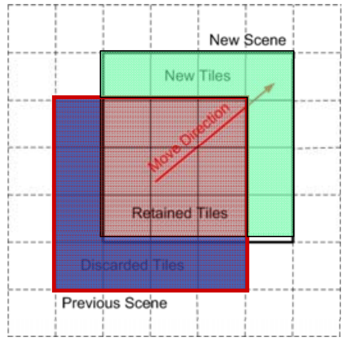
- USGS 1-degree DEMs
  - 72,000x28,800
  - divide into 90 blocks
  - each block is 4,097x4,097



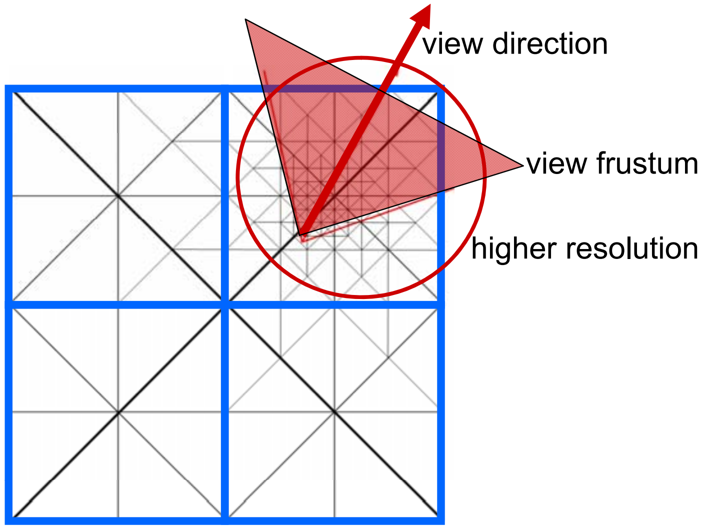
Dynamic Block Management

- Windowing of visible scene [Gross 1995]
  - do not load entire terrain, discard invisible blocks



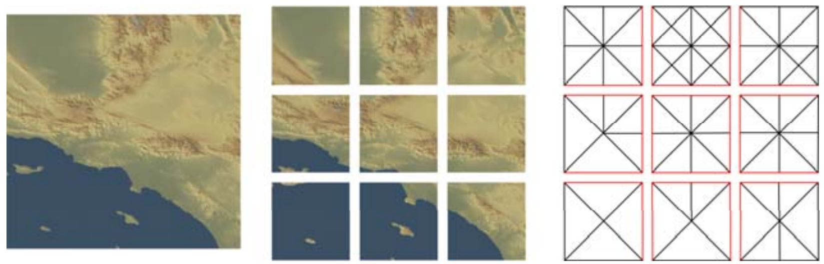


View-Dependent Mesh Refinement

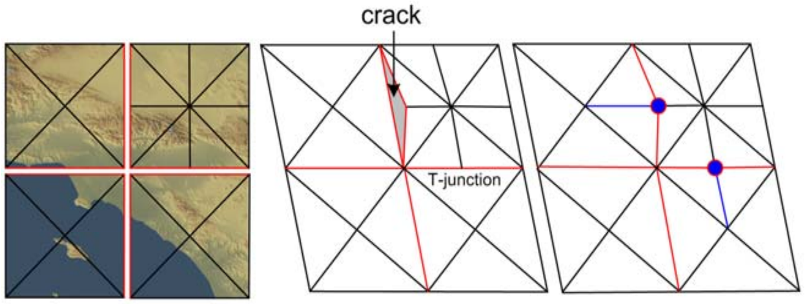


Block is Represented by a Mesh

- Blocks may have different level-of-detail
- Discontinuity must be solved !

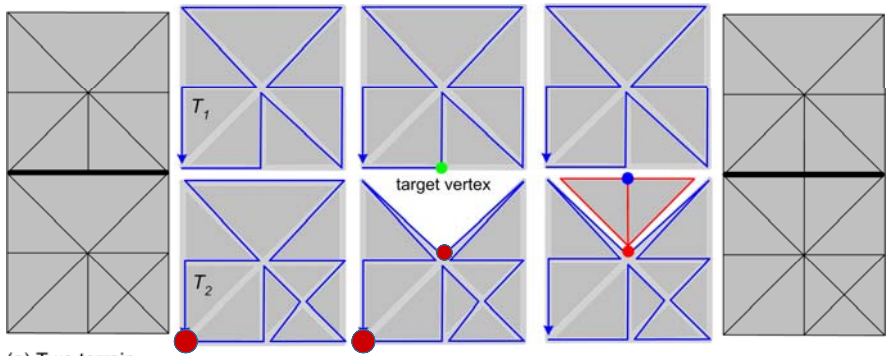


Crack and T-Junction



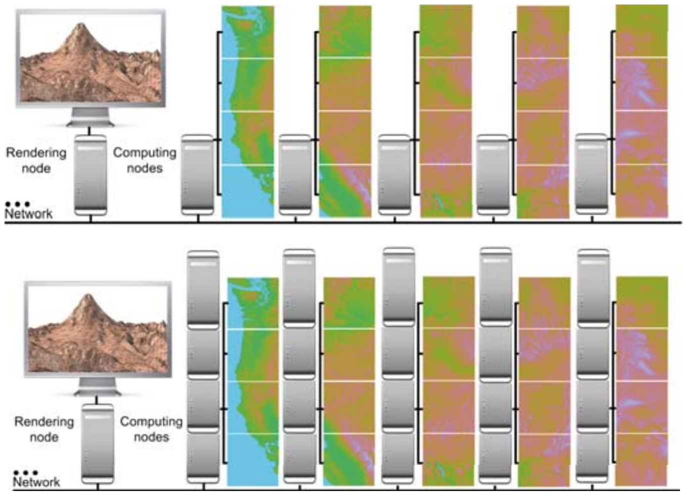
(a) Terrain blocks. (b) Crack / T-junction between blocks. (c) Crack / T-junction removed.

Crack Removal Algorithm

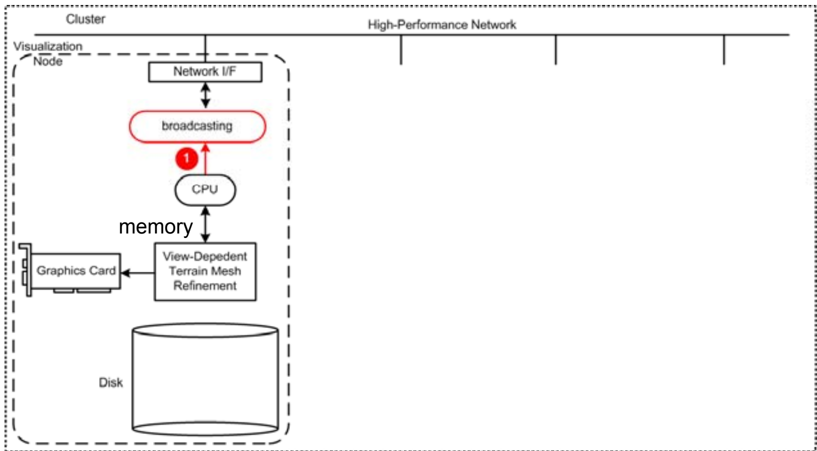


(a) Two terrain meshes (b) Two Triangle strips: T_1 & T_2 (c) Remove a boundary triangle (d) Fill in two triangles (e) Final result

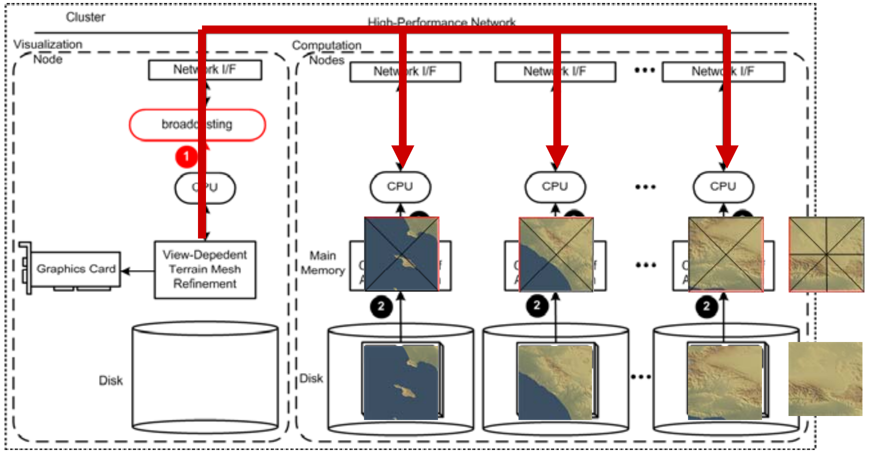
5 servers vs. 20 servers



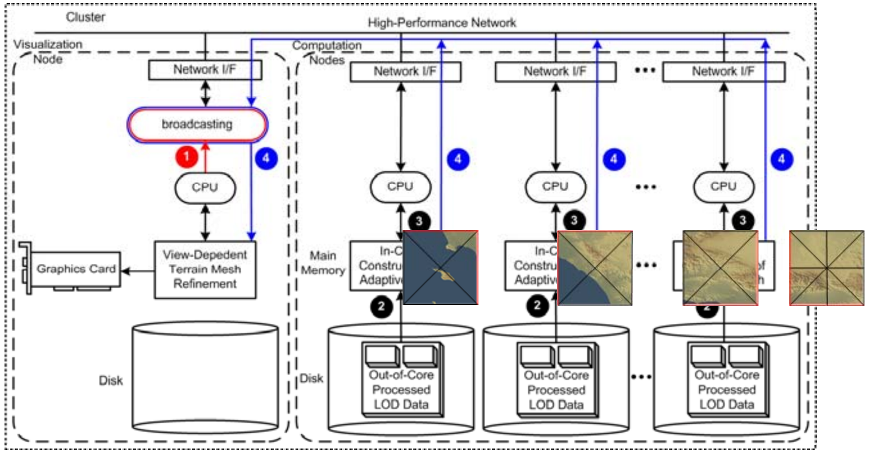
Data Flow



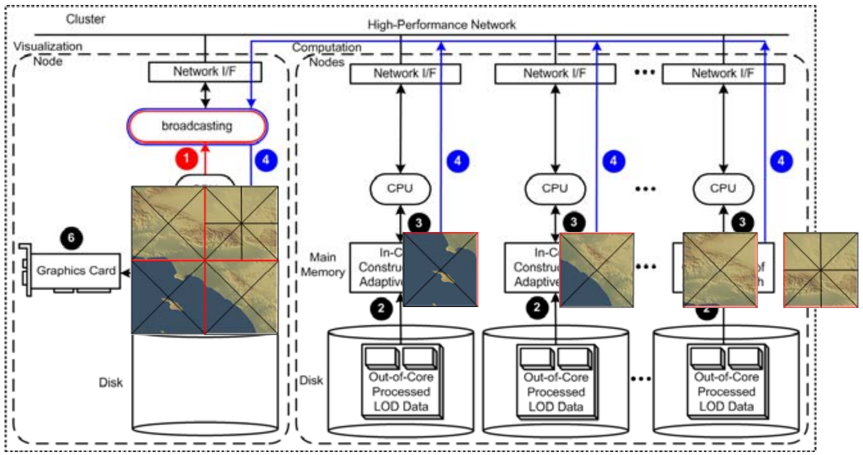
Computing Nodes Construct Mesh



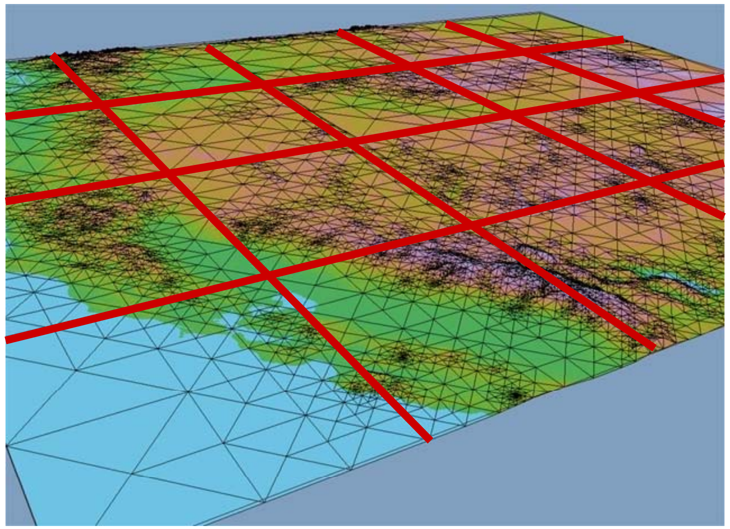
Meshes Sent to Rendering Node



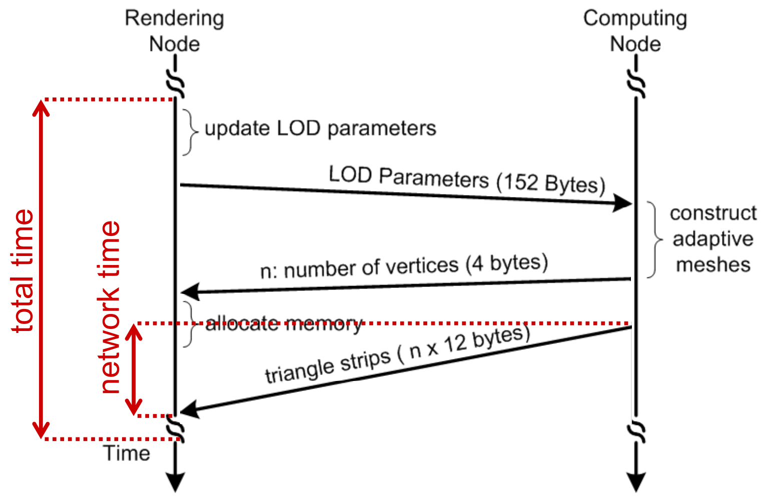
Combine Meshes & Crack Removal



Visualization of 20 Terrain Blocks



Socket Communication (TCP/IP)



Outline

- Problem Statement & Motivation
- Large-Scale Rendering
  - Sorting classification (sort-last, sort-middle, sort-first)
  - Sort-first rendering (Visualization on HIPerWall)
  - Sort-last rendering (Distributed Parallel Terrain Rendering)
- Conclusions

Conclusions

- Sort-first rendering
  - A tiled display system on HIPerWall based on socket messages
- Sort-last rendering
  - A distributed parallel terrain rendering method that outperforms old methods in rendering capacity
- Interactive visualization
  - is made possible with distributed parallel processing, out-of-core management, level-of-detail refinement
  - aids in interpreting complex large-scale datasets
  - highlights characteristics otherwise difficult to pinpoint

References

- Jan P. Springer, Christoph Sladeczek, Martin Scheffler, Jan Hochstrate, Bernd Frohlich, Frank Melchior, 2006, “A Survey of Large High-Resolution Display Technologies, Techniques, and Applications”, *Proceedings of the IEEE Virtual Reality Conference (VR 2006)* , 223-236.
- S. Molnar,M. Cox, D. Ellsworth, H. Fuchs, 1994, “A Sorting Classification of Parallel Rendering”, *IEEE Computer Graphics and Applications*, 14(4), 23-32.
- Ed Angel, 2006, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*, Fourth Edition, Addison-Wesley 2006
- Amit Chourasia, Steve Cutchin, Yifeng Cui, Reagan W. Moore, Kim Olsen, Steven M. Day, J. Bernard Minster, Philip Maechling, and Thomas H. Jordan, 2007, “Visual Insights into High-Resolution Earthquake Simulations”, *IEEE Computer Graphics and Applications*, 27(5), 28-34
- Lighthouse3D, http://www.lighthouse3d.com/
- DeFanti, T.A., Dawe, G., Sandin, D.J., Schulze, J.P., Otto, P., Girado, J., Kuester, F., Smarr, L., Rao, R. (2009). TheStarCAVE, a Third-Generation CAVE and Virtual Reality OptIPortal. To appear in Future Generation Computer Systems, Volume 25(2), Elsevier, February 2009, pp. 169-178. (pdf)