Point Reprojection and Dynamic Scenes

Erik Reinhard
University of Utah
reinhard@cs.utah.edu

Overview

- Point reprojection
- Animation

Point Reprojection

Brute Force Ray Tracing:

- Enables interactive ray tracing
- Does not allow large image sizes
- Does not scale to scenes with high depth complexity

Solution:

- Exploit temporal coherence
- Re-use results from previous frames

SIGGRAPH 2001

Practical Solutions:

- Tapestry (Simmons et. al. 2000)
  - Focuses on complex lighting simulation
- Render cache (Walter et. al. 1999)
  - Addresses scene complexity issues
  - Explained next
- Parallel render cache (Reinhard et. al. 2000)
  - Builds on Walter's render cache
  - Explained next

SIGGRAPH 2001

Render Cache Algorithm

Basic setup

- One front-end for:
  - Displaying pixels
  - Managing previous results
- Parallel back-end for:
  - Producing new pixels

SIGGRAPH 2001

Render Cache Front-end

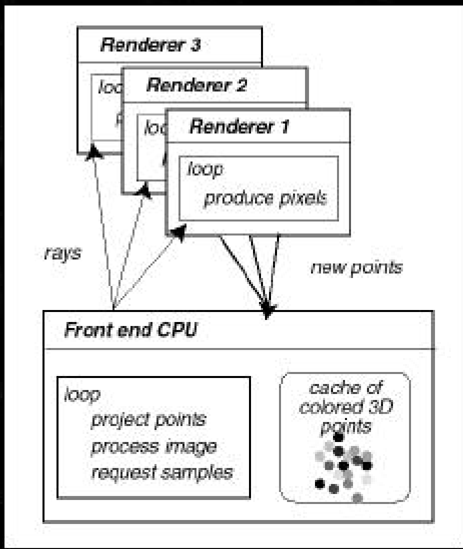
Frame based rendering

For each frame do:

- Project existing points
- Smooth image and display
- Select new rays using heuristics
- Request samples from back-end
- Insert new points into point cloud

SIGGRAPH 2001

Render Cache



SIGGRAPH 2001

Render Cache (2)

- Point reprojection is relatively cheap
- Smooth camera movement for small images
- Does not scale to large images or large numbers of renderers → front-end becomes bottleneck

SIGGRAPH 2001

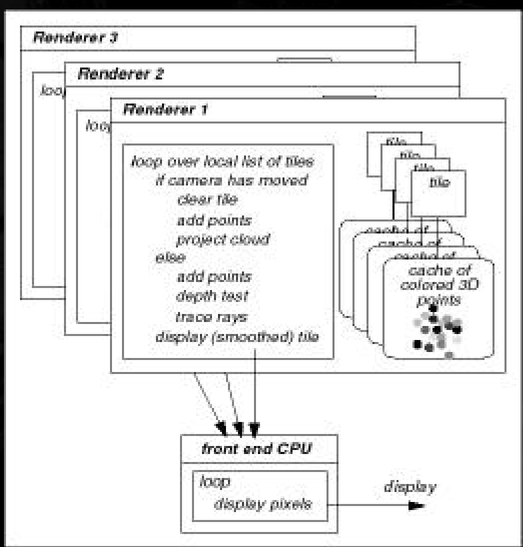
Parallel Render Cache

Aim: remove front-end bottleneck

- Distribute point reprojection functionality
- Integrate point reprojection with renderers
- Front-end only displays results

SIGGRAPH 2001

Parallel Render Cache (2)



SIGGRAPH 2001

Parallel Render Cache (3)

Features:

- Scalable behavior for scene complexity
- Scalable in number of processors
- Allows larger images to be rendered

- Retains artifacts from render cache
- Introduces new artifacts

Artifacts

- Render cache artifacts at tile boundaries
- Image deteriorates during camera movement

These artifacts are deemed more acceptable than loss of smooth camera movement!

Video

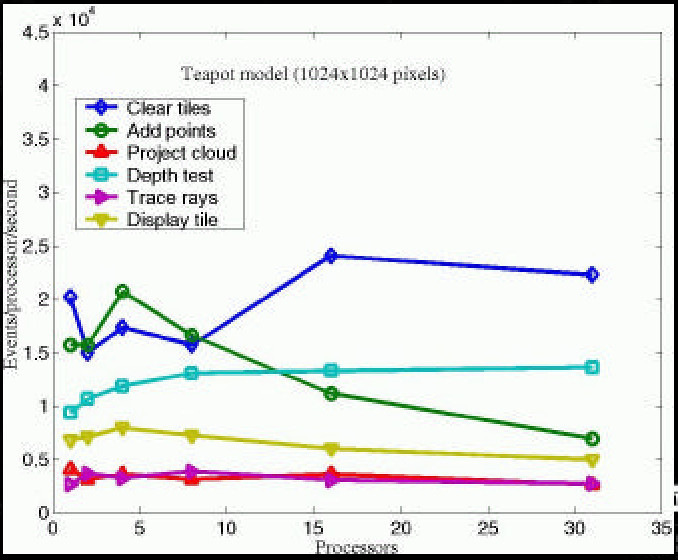
Test Scenes

Results

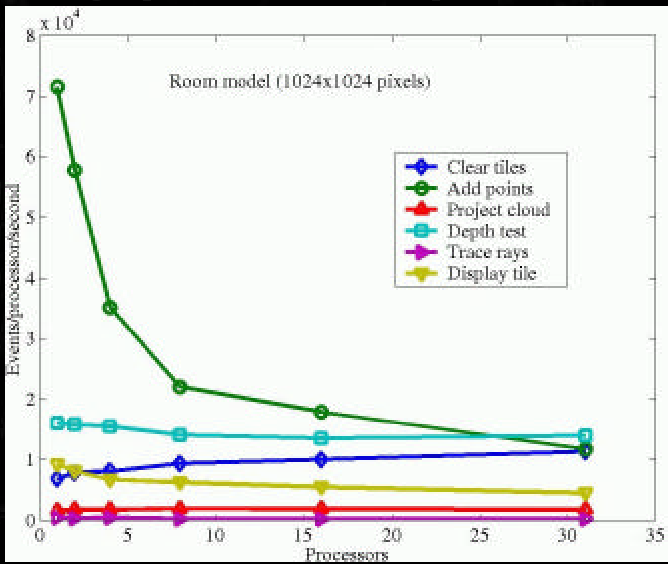
Sub-parts of algorithm measured individually

- Measure time per call to subroutine
- Sum over all processors and all invocations
- Afterwards divide by number of processors and number of invocations
- Results are measured in events per second per processor

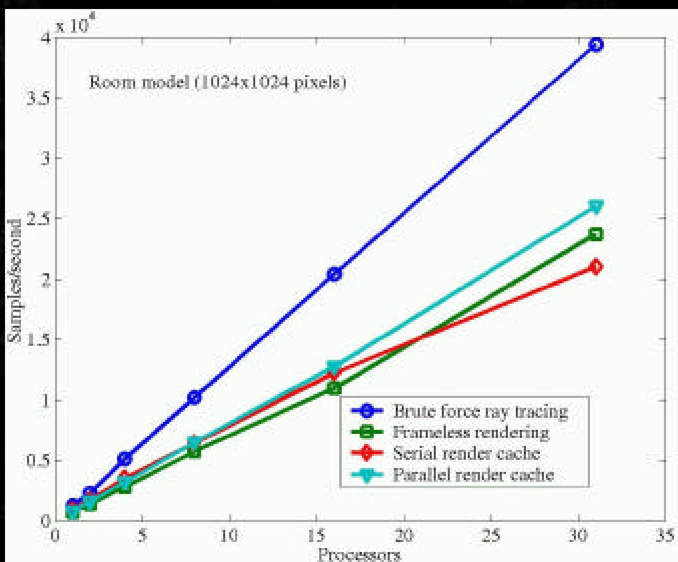
Scalability (Teapot Model)



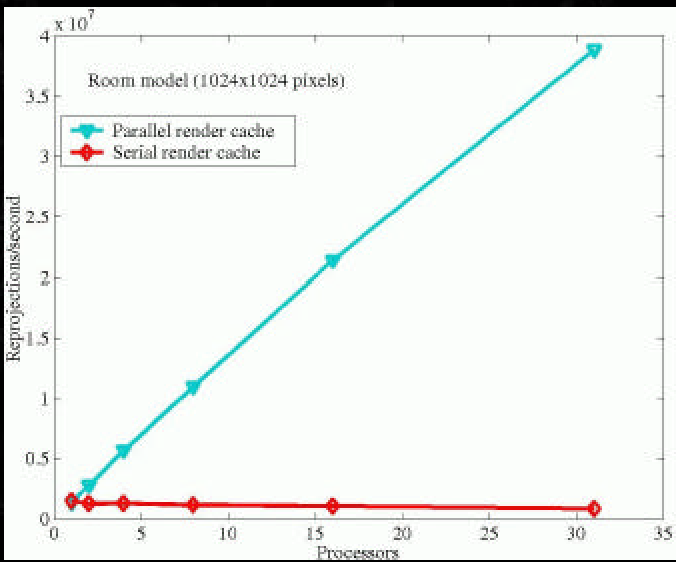
Scalability (Room Model)



Samples Per Second



Reprojections Per Second



Conclusions

- Exploitation of temporal coherence gives significantly smoother results than available with brute force ray tracing alone
- This is at the cost of some artifacts which require further investigation
- (More results available in course notes)

Overview

- Point reprojection
- Animation

SIGGRAPH 2001

Animation

SIGGRAPH 2001

Why Animation

- Once interactive rendering is feasible, walk-through is not enough
- Desire to manipulate the scene interactively
- Render preprogrammed animation paths

SIGGRAPH 2001

Issues to Be Addressed:

What stops us from animating objects?

- Answer: spatial subdivisions
- Acceleration structures normally built during pre-processing
- They assume objects are stationary

SIGGRAPH 2001

Possible Solutions:

Target applications that require a small number of objects to be manipulated/animated

- Render these objects separately
  - Traversal cost will be linear in the number of animated objects
  - Only feasible for extremely small number of objects

SIGGRAPH 2001

Possible Solutions (2)

Target small number of manipulated or animated objects

- Modify existing spatial subdivisions
  - For each frame delete object from data structure
  - Update object's coordinates
  - Re-insert object into data structure
- This is our preferred approach

SIGGRAPH 2001

Spatial Subdivision Should Be Able to Deal With the Following:

- Basic operations such as insertion and deletion of objects should be rapid
- User manipulation can cause the extent of the scene to grow

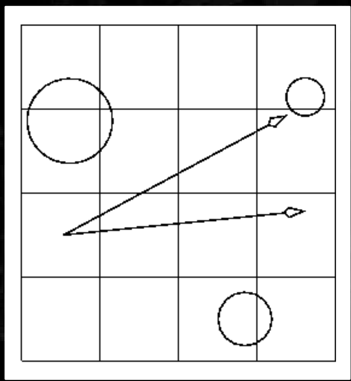
Subdivisions Investigated:

- Regular grid
- Hierarchical grid
  - Borrows from octree spatial subdivision
  - In our case this is a full tree: all leaf nodes are at the same depth

Both acceleration structures are investigated in the next few slides

Regular Grid Data Structure

We assume familiarity with spatial subdivisions!



Object Insertion Into Grid

- Compute bounding box of object
- Compute overlap of bounding box with grid voxels
- Object is inserted into overlapping voxels

- Object deletion works similarly

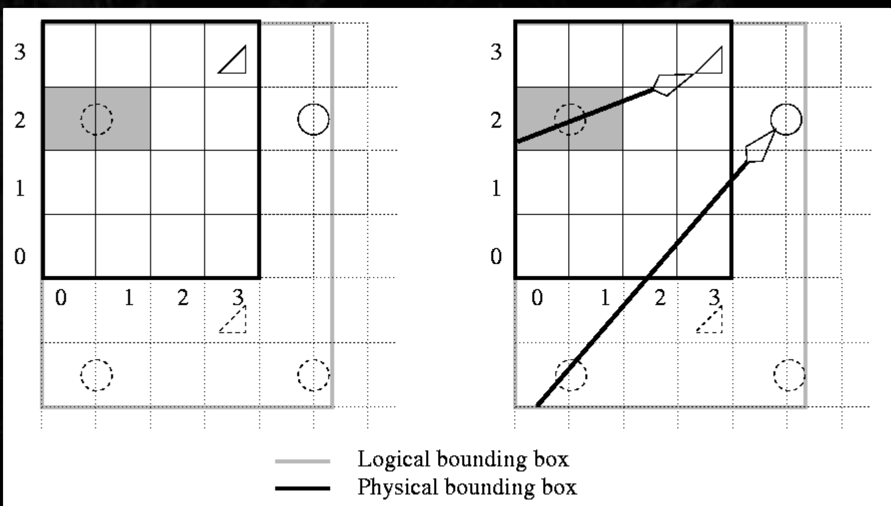
Extensions to Regular Grid

Dealing with expanding scenes requires

- Modifications to object insertion/deletion
- Ray traversal

SIGGRAPH 2001

Extensions to Regular Grid (2)



Features of New Grid Data Structure

We call this an 'Interactive Grid'

- Straightforward object insertion/deletion
- Deals with expanding scenes

- Insertion cost depends on relative object size
- Traversal cost somewhat higher than for regular grid

SIGGRAPH 2001

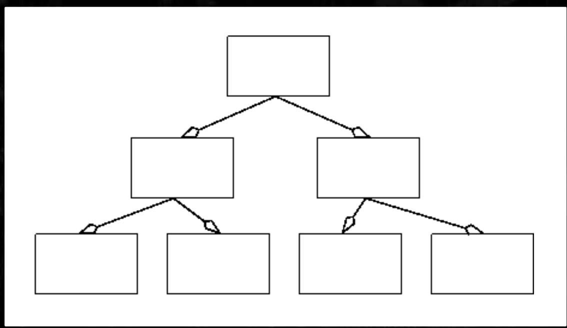
Hierarchical Grid

Objectives

- Reduce insertion/deletion cost for larger objects
- Retain advantages of interactive grid

SIGGRAPH 2001

Hierarchical Grid (2)



Hierarchical Grid (3)

Build full octree with all leaf nodes at the same level

- Allow objects to reside in leaf nodes as well as in nodes higher up in the hierarchy
- Each object can be inserted into one or more voxels of at most one level in the hierarchy
- Small object reside in leaf nodes, large objects reside elsewhere in the hierarchy

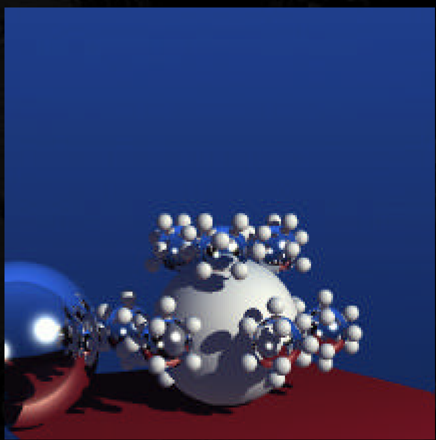
Hierarchical Grid (4)

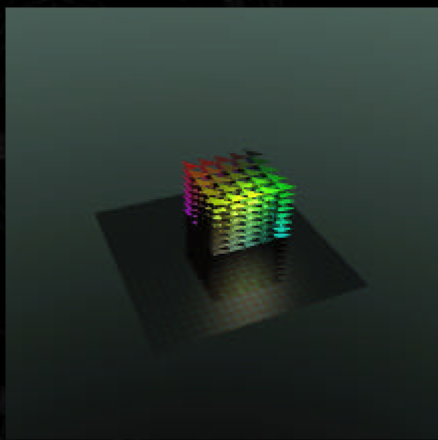
Features:

- Deals with expanding scenes similar to interactive grid
- Reduced insertion/deletion cost
- Traversal cost somewhat higher than interactive grid

SIGGRAPH 2001

Test Scenes





SIGGRAPH 2001

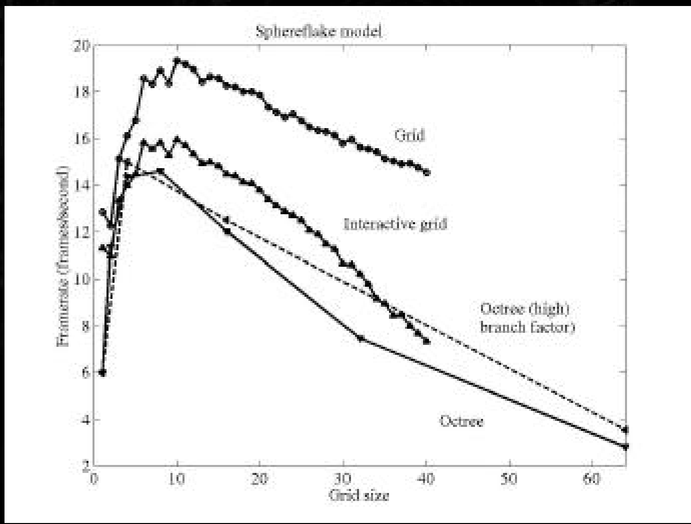
Video

Measurements

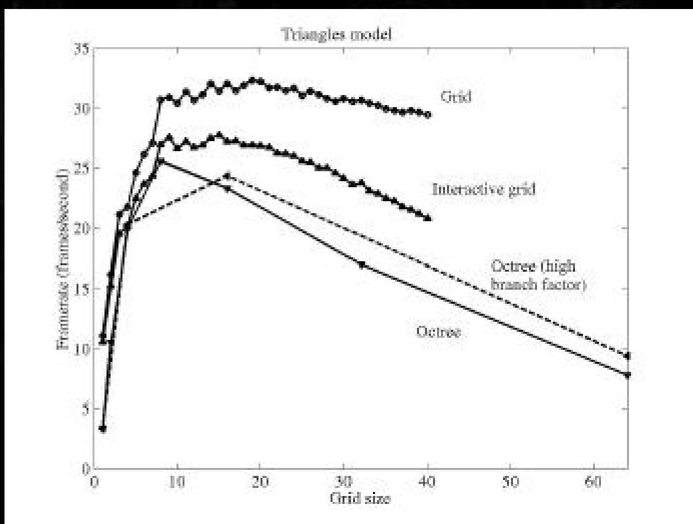
We measure:

- Traversal cost of
  - Interactive grid
  - Hierarchical grid
  - Regular grid
- Object update rates of
  - Interactive grid
  - Hierarchical grid

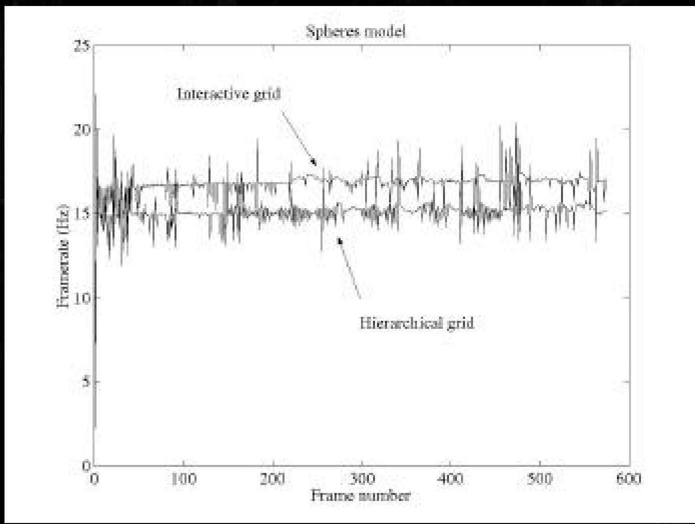
Framerate vs. Grid Size (Sphereflake)



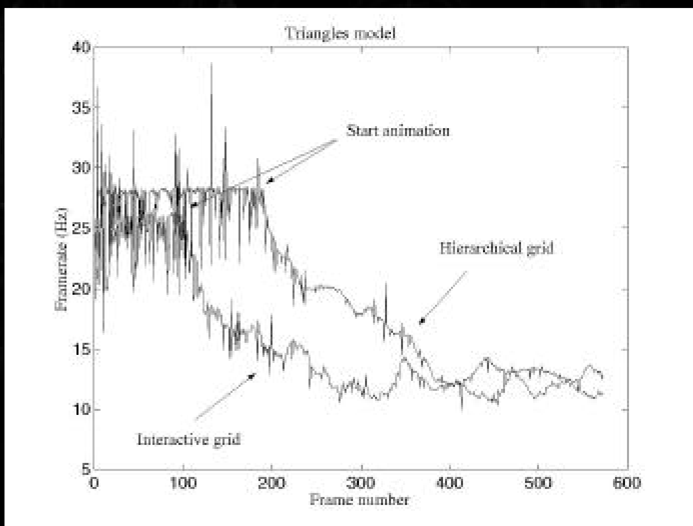
Framerate vs. Grid Size (Triangles model)



Framerate Over Time (Sphereflake Model)



Framerate Over Time (Triangles Model)



Conclusions

- Interactive manipulation of ray traced scenes is both desirable and feasible using these modifications to grid and hierarchical grids
- Slight impact on traversal cost
- (More results available in course notes)

SIGGRAPH 2001

Acknowledgements

Thanks to:

- Brian Smits, Peter Shirley and Charles Hansen for involvement in the animation and parallel point reprojection projects
- Bruce Walter, George Drettakis and Steven Parker for the render cache source code

SIGGRAPH 2001