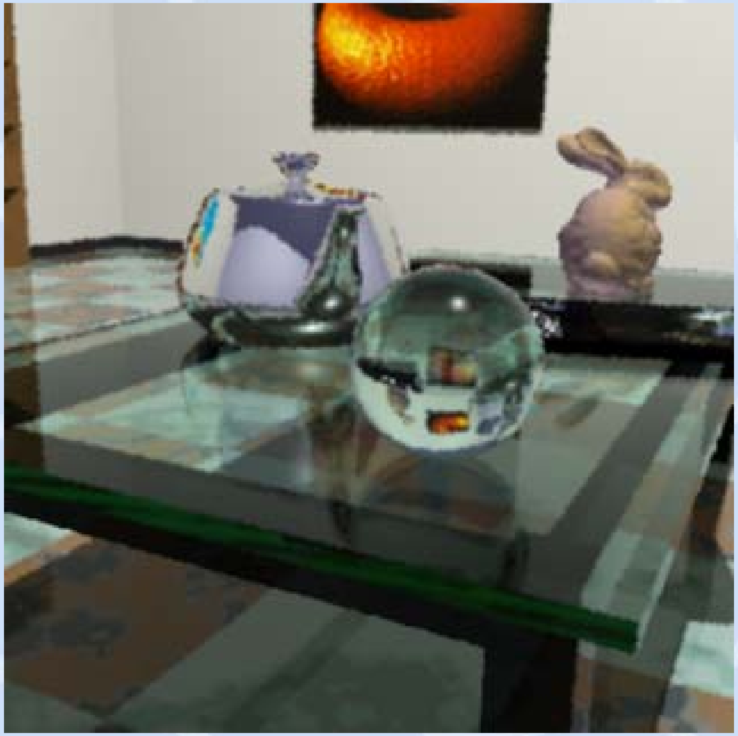
Reusing Pixels for Interactive Global Illumination

Bruce Walter
Cornell University

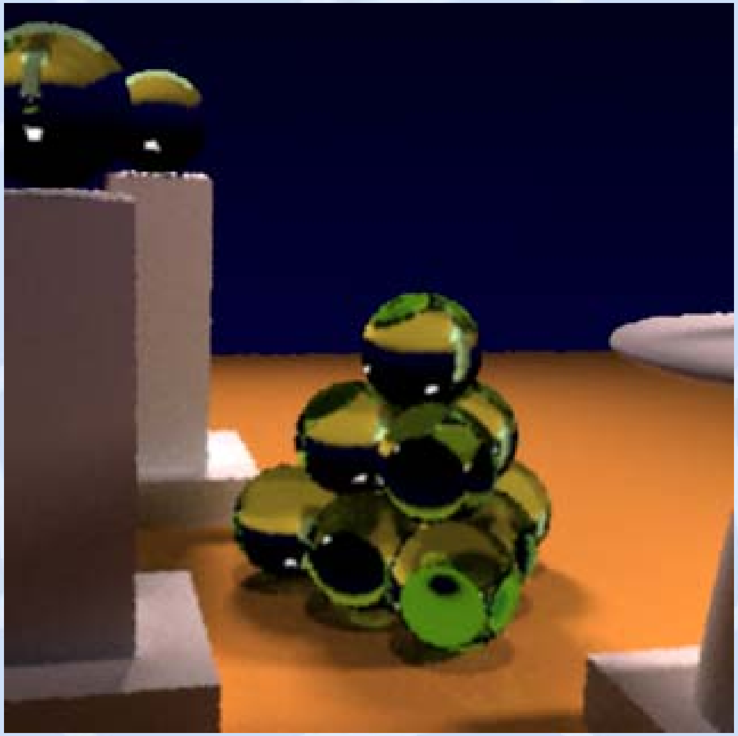
SIGGRAPH 2003
SAN DIEGO

Interactive Global Illumination

- Problem: global illumination methods are too slow for interactive use



Ray tracing



Path tracing

Interactive Global Illumination

- Do we need to render every pixel of every frame?

Frame 10

Frame 20

Frame 30

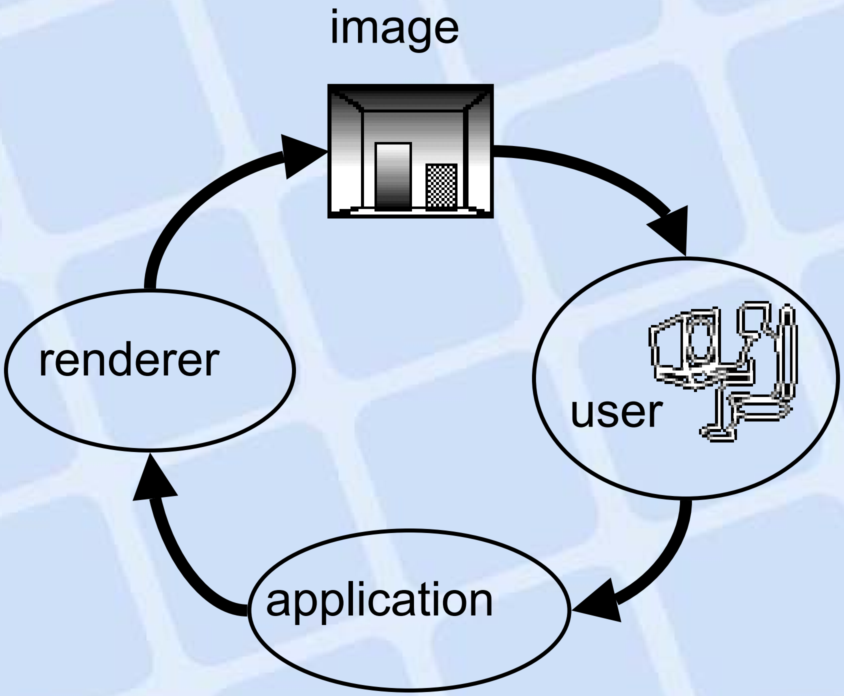
Display Process

SIGGRAPH 2003
SAN DIEGO

- Automatically exploit spatial and temporal coherence
- Layered on top of an existing (slow) global illumination renderer
- Provide interactive performance

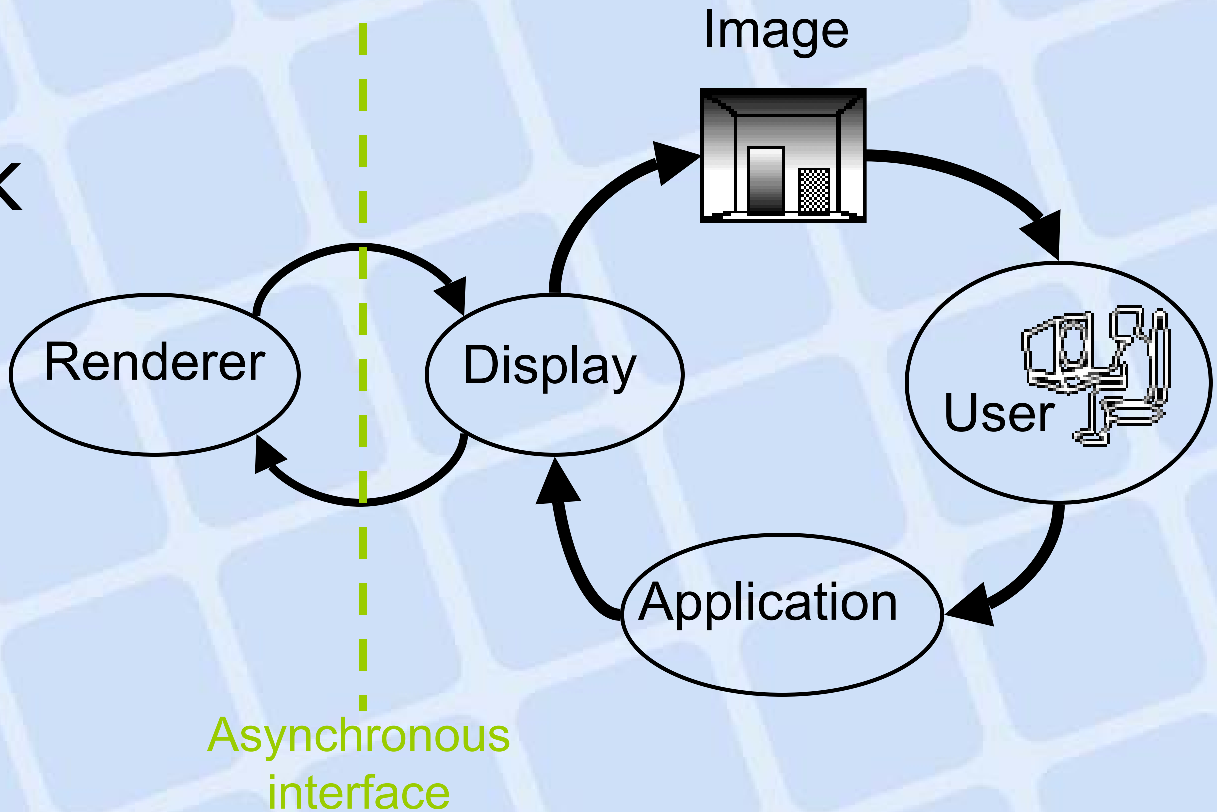
Interactive Global Illumination

- Standard visual feedback loop
  - Entirely synchronous
  - Framerate is limited by the renderer



Interactive Global Illumination

- Modified visual feedback loop



Interactive Global Illumination

SIGGRAPH 2003 SAN DIEGO

- Interactive requirements
  - Image quality
  - Responsiveness
    - Don’t make the user wait
    - Provide rapid user feedback
  - Consistency
    - Don’t surprise or distract the user
    - Avoid sudden changes if possible
      - Eg, in quality, frame rate, popping, etc.

Issues

SIGGRAPH 2003
SAN DIEGO

- How to cache results for reuse?
  - Interpolate images from sparse data
  - Handle camera motion between frames
    - Handle occlusion changes
- Which samples should be rendered?
  - Prioritize for maximum benefit
- What if scene or shading changes?
  - Detect and discard data that is no longer valid

Approaches

- Image based
  - Post-rendering Warp *(Mark97)*
  - Corrective texturing *(Stamminger00)*
- Point based
  - Render Cache *(Walter99,02)*
- Mesh based
  - Holodeck *(Ward98,99)*
  - Tapestry *(Simmons00)*
  - Shading Cache *(Tole02)*

Post-Rendering 3D Warp

- Render subset of frames
  - E.g, every 6th frame is rendered
- Use standard image warping techniques to compute the other frames

Post-Rendering 3D Warp

SIGGRAPH 2003
SAN DIEGO

- Problem: holes and missing data

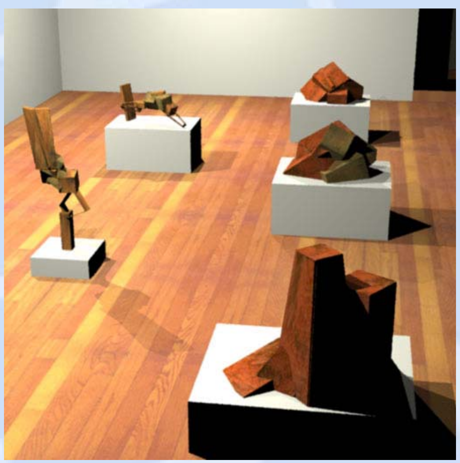
Reference frame

Warped frame

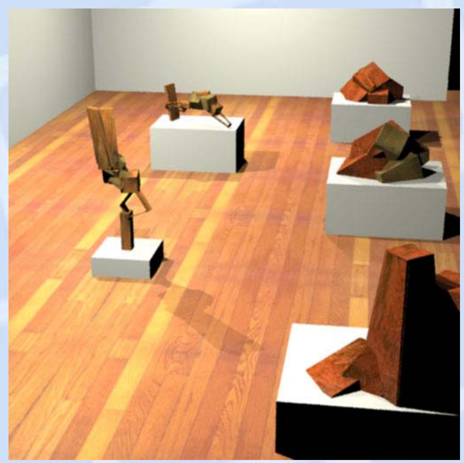
The camera is moving to the left in this example.

Post-Rendering 3D Warp

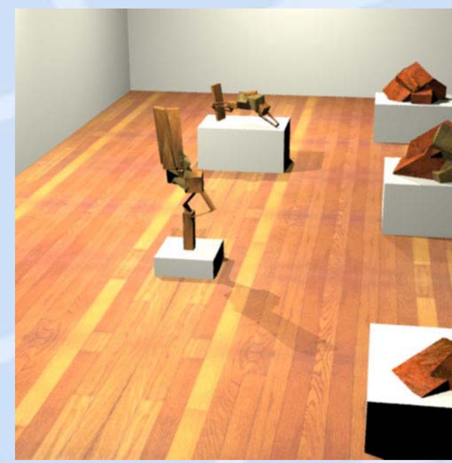
- Warp from both past and future reference frames
  - Heuristics for combining pixel results



Prior reference



Warped frame



Next reference

Post-Rendering 3D Warp

- Must predict the locations of future frames
  - Longer predictions become rapidly less accurate

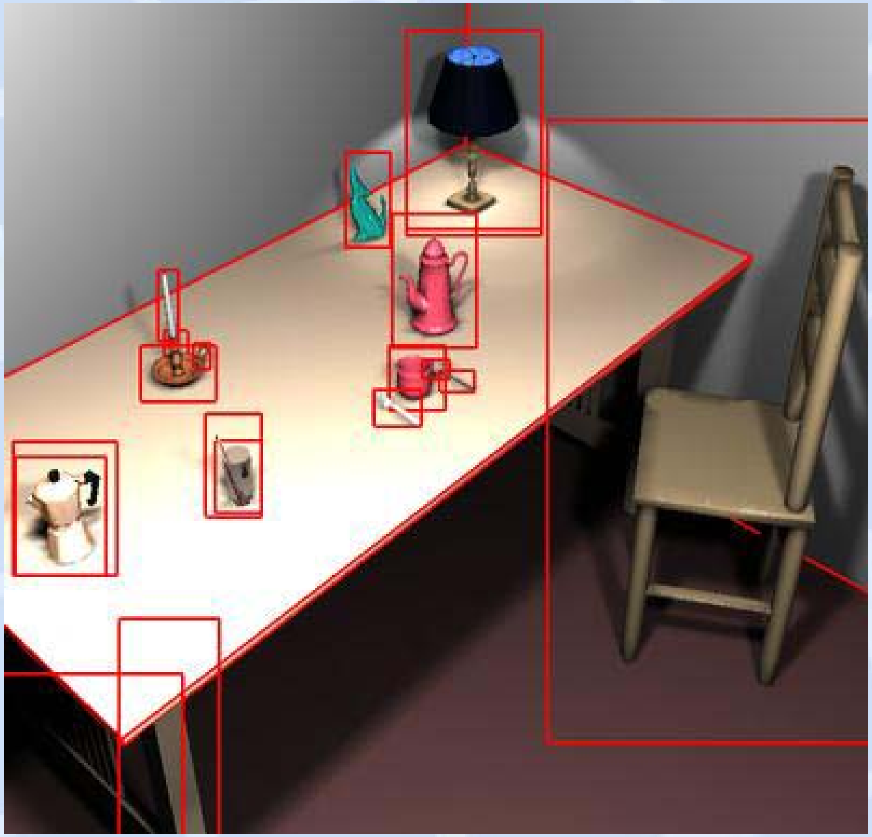
Corrective Texturing

SIGGRAPH 2003
SAN DIEGO

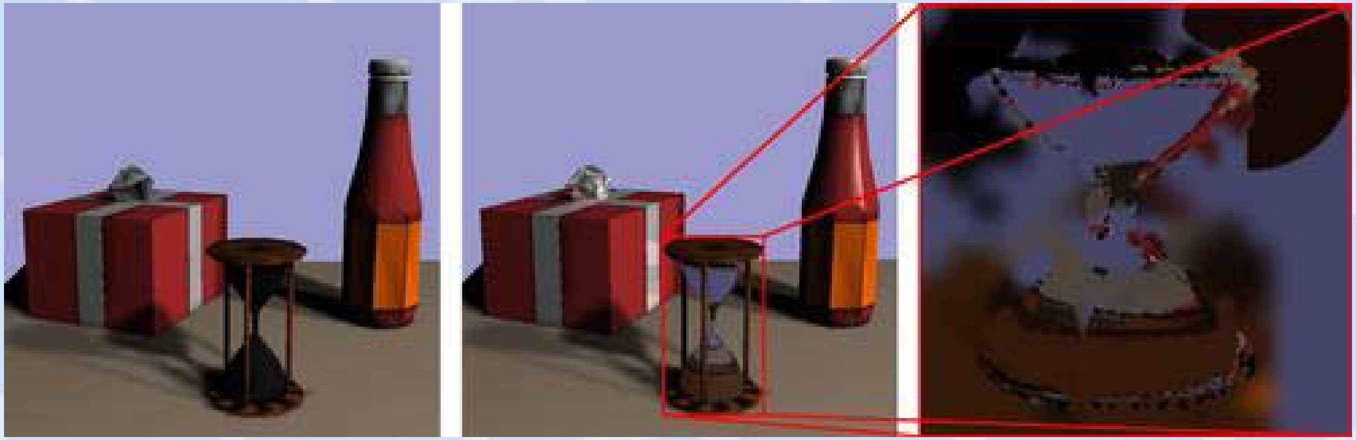
- Start with a standard hardware rendering of scene
  - Graphics hardware very good at interactive display
  - Start with a radiosity solution
- Compare to underlying renderer
  - Apply corrections where they differ
  - Corrections applied as projective textures

Corrective Texturing

- Corrective textures are dynamically assigned to objects



Corrective Texturing



Radiosity solution

Corrected image

Corrective texture

Corrective Texturing

SIGGRAPH 2003
SAN DIEGO

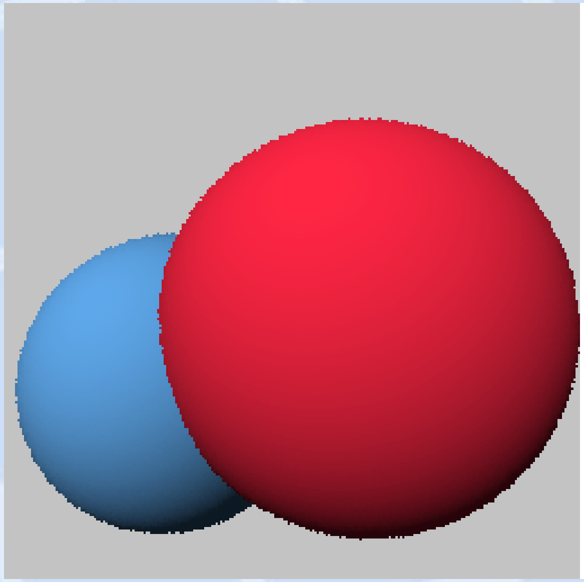
- Sparse rendered samples compared to hardware displayed results
  - Differences splatted into textures
  - More samples generated near points which had large differences
  - Samples which are likely to have changed are deprecated so that can be overwritten by future results

Render Cache

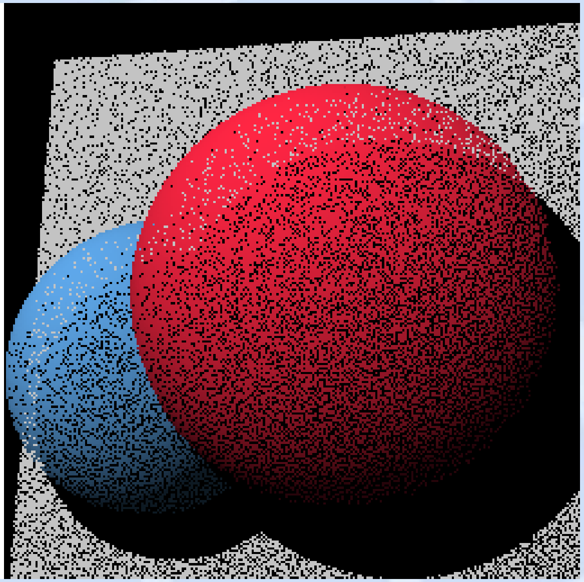
- Results stored as cloud of unordered points with:
  - 3D position (located on surfaces)
  - Color
  - Age
  - Object identifier

Render Cache

- Reproject points into current frame
  - Occlusion errors
  - Holes in data



Initial view

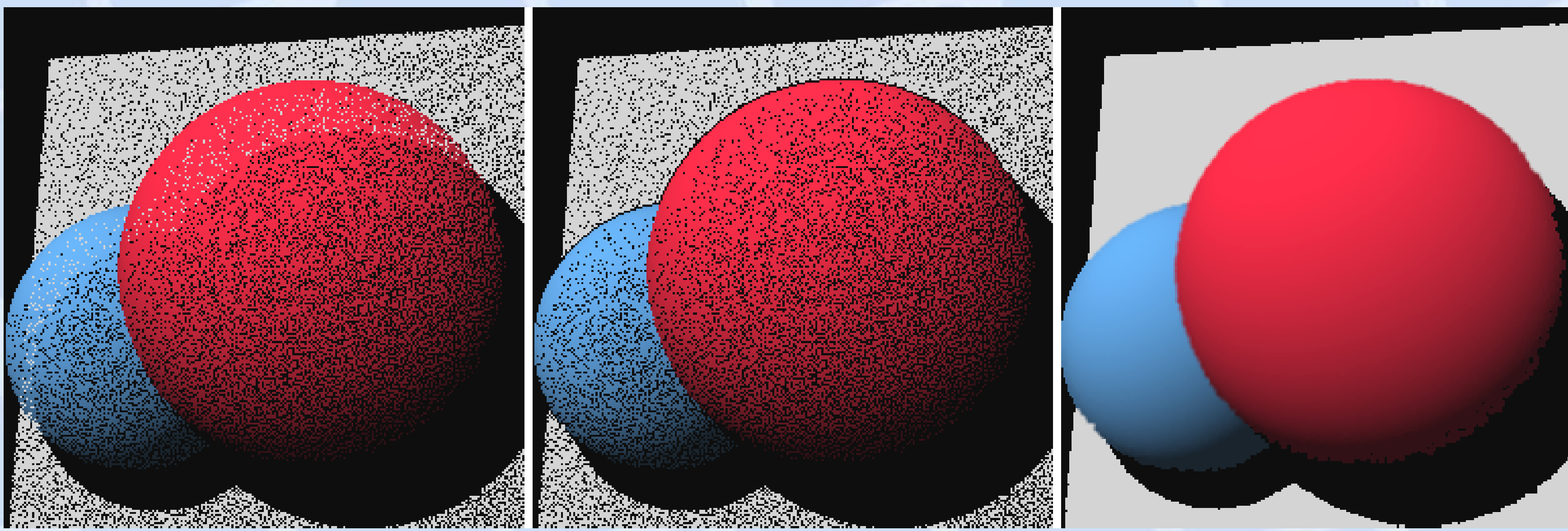


After reprojection

Render Cache

SIGGRAPH 2003
SAN DIEGO

- Use occlusion culling heuristic
- Interpolation to fill holes
  - Fixed size kernels, 3x3 and 7x7

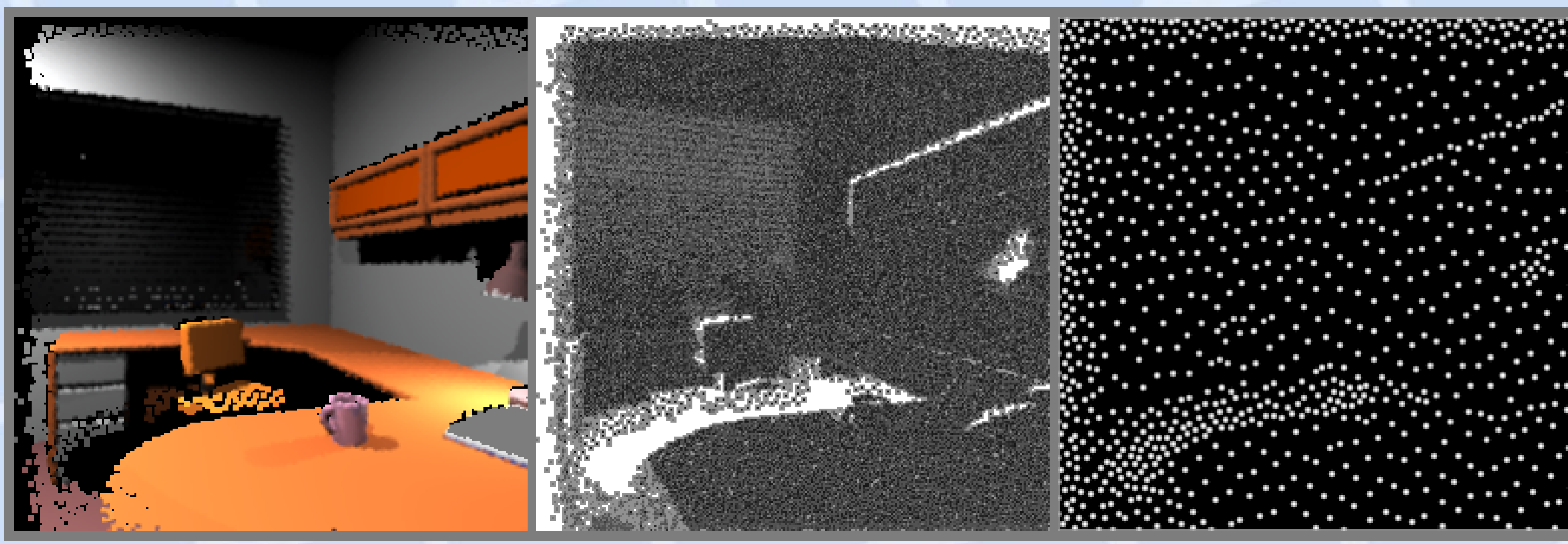


Reprojection Occlusion cull Interpolation

Render Cache

- Priority image for sampling
  - High priority for sparse regions
  - High priority for old points
- Convert priority image to sparse set of locations to be rendered
  - Uses error-diffusion dither
- Also uses predictive sampling
  - Try to sample new regions just before they become visible

Render Cache



Displayed image Priority image Requested pixels

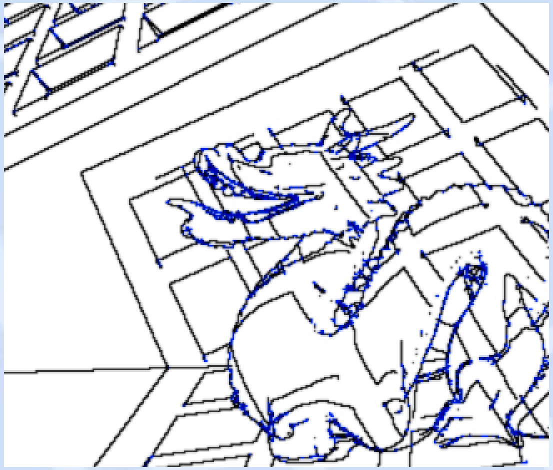
Render Cache

SIGGRAPH 2003
SAN DIEGO

- Recomputes old samples to detect changes
  - Nearby points are aged to raise priority and cause point invalidation
- Object motion
  - Associated points can be transformed along with the object

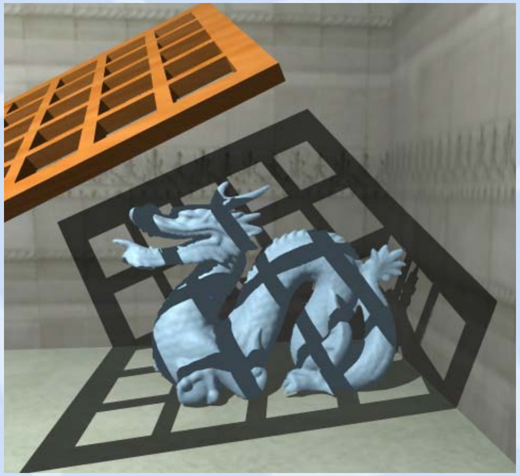
Point & Edge Rendering

- Tracks and uses discontinuity edges
- SIGGRAPH 2003 paper
  - Presentation: Wednesday 1:45pm



Edges

Edges and samples



Output image

Holodeck

SIGGRAPH 2003
SAN DIEGO

- Uses Radiance as its renderer
- Rendered samples stored in a 4D data structure
  - Similar to Light Field or Lumigraph
  - Can be very large
    - Paged to disk if necessary
  - Lazily evaluated
    - Samples generated near current viewpoint
  - Position and other parameters are specified by the user

Holodeck

- Uses Gouraud-shaded triangle mesh
  - Get samples near current viewpoint
  - Samples become vertices in a mesh
  - Delaunay triangulation of samples in direction space about a center of projection
  - Hardware provides fast display including interpolation between samples

Holodeck

- Mesh construction
  - Choose center point
  - Construct Delaunay triangulation
    - Based on sample point's projection onto a sphere about the center point
  - Display mesh using hardware
  - Update incrementally with new samples
- If user moves too far, then must choose new center and rebuild mesh

Holodeck

SIGGRAPH 2003
SAN DIEGO

- Depth heuristic to reduce occlusion errors
- Special techniques for designated moving objects

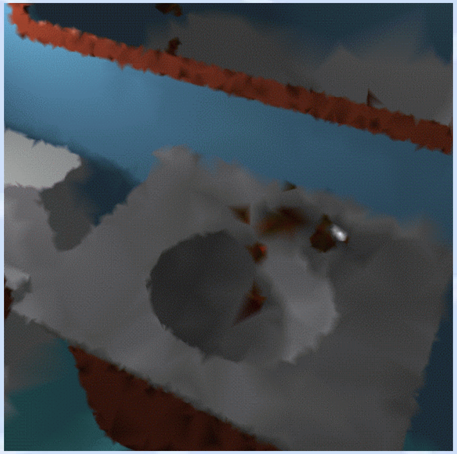
Tapestry

- Based on Holodeck system with several enhancements:
  - Prioritized sampling
  - Incremental “recentering” of spherical Delaunay mesh as viewpoint moves
  - Fixed cache size
    - Max vertices = pixels
  - Sample invalidation
    - Occlusion and color change heuristics

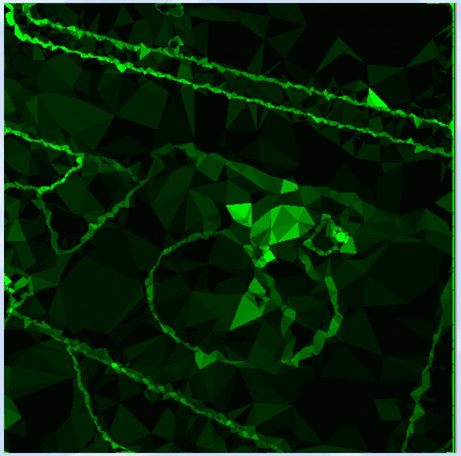
Tapestry

SIGGRAPH 2003
SAN DIEGO

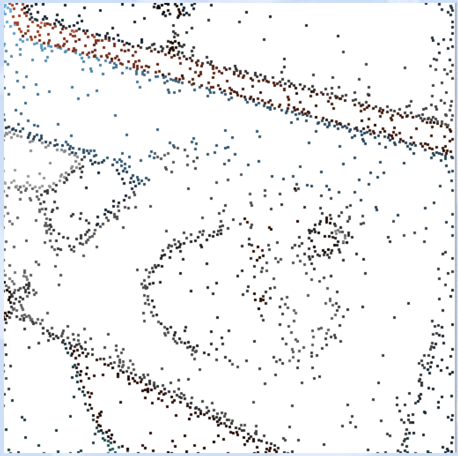
- Each triangle assigned a priority
  - Color & depth differences and age
  - Rasterize priority using hardware
  - Quasi-random sampling with rejection



Image



Priority



Samples

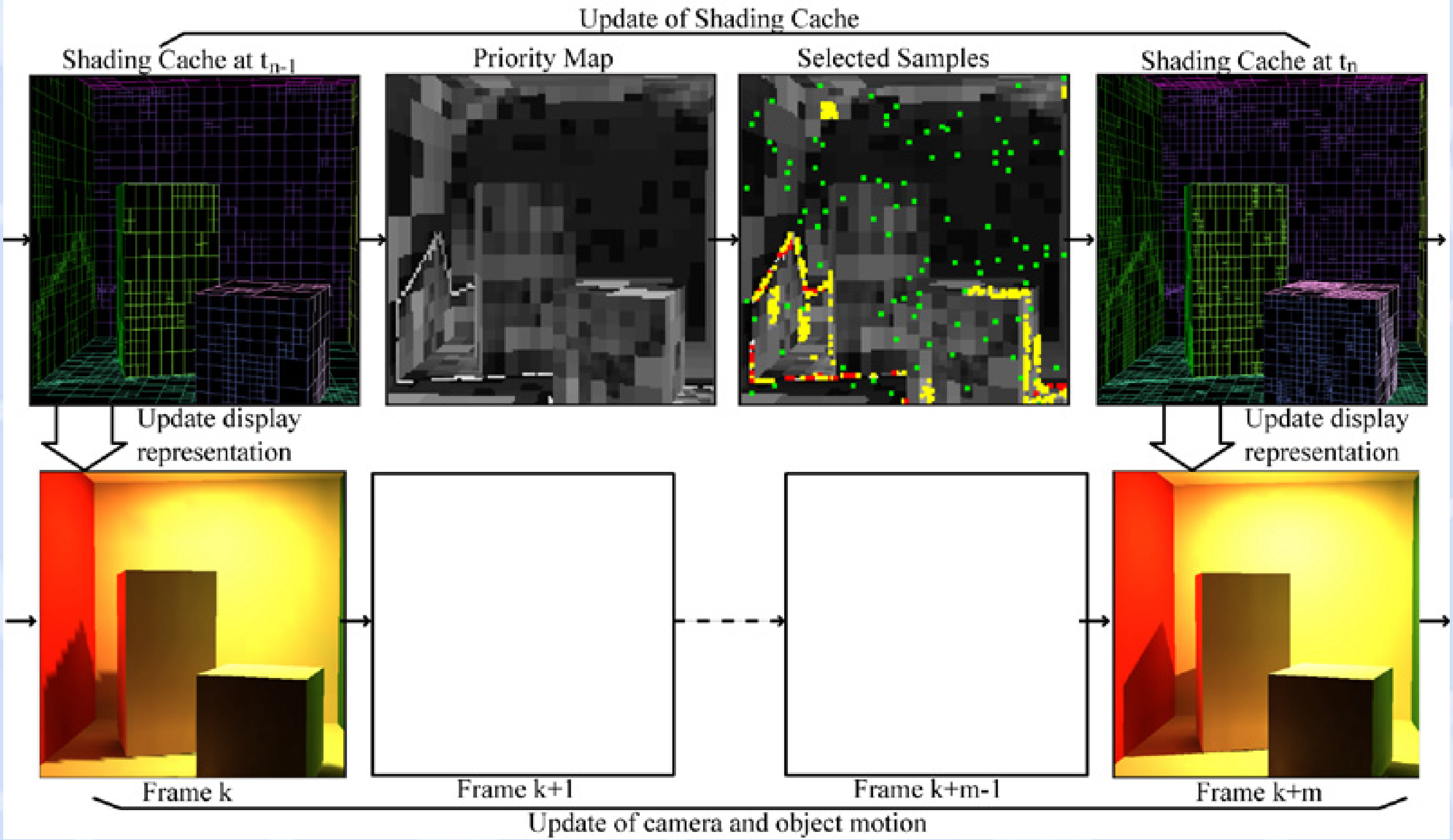
Shading Cache

- Display mesh is refinement of original scene mesh
  - No occlusion errors
  - Hardware handles textures
  - Display mesh >= original mesh
  - Easier to handle moving objects
- Decouples frame update from mesh update

Shading Cache

- Adds flood-fill heuristic for sampling
  - Discontinuities require locally dense subdivision
- Mesh de-refinement
  - If not recently visible
  - If denser than pixel spacing
  - If color changes are detected

Shading Cache



Comparison

	Target renderer speed	Sparseness ratio	Typical frame rates
Warp	< 1s	4 - 10	20 - 60 fps
Corrective Tex.	20 - 200s	250 - 1000	5 - 10 fps
Render Cache	.5 - 10s	8 - 100	10 - 20 fps
Holodeck	200 - 1000s	NA	NA
Tapestry	500 - 1000s	2000 - 8000	3 - 10 fps
Shading Cache	50 - 1000s	1000 - 20000	30 - 60 fps

Comparison

	Hardware accelerated	Independent of scene complexity	Moving objects
Warp	No	Yes	No
Corrective Tex.	Yes	No	No
Render Cache	No	Yes	Limited
Holodeck	Yes	No	Special
Tapestry	Yes	Yes	No
Shading Cache	Yes	No	Yes

Downloadable Versions

- Render Cache
  - http://www.graphics.cornell.edu/research/interactive/rendercache/
- Holodeck
  - http://radsite.lbl.gov/radiance

Bibliography

- W. Mark, L. McMillan, and G. Bishop, “Post-Rendering Warping”, *Proceedings of 1997 Symposium on Interactive 3D Graphics*, 1997.
  - http://www.cs.utexas.edu/users/billmark/i3dwww/i3d.html
- M. Stamminger, J. Haber, H. Schirmacher, H.-P. Seidel, “Walkthroughs with Corrective Texturing”, *Rendering Techniques 2000* (Proc. Eurographics Workshop on Rendering), 2000
  - http://www-sop.inria.fr/reves/personnel/Marc.Stamminger/corrtex/

Bibliography

- B. Walter, G. Drettakis, S. Parker, “Interactive Rendering Using the Render Cache”, *Rendering Techniques '99*, (Proc. 10th Eurographics Workshop on Rendering), 1999
- B. Walter, G. Drettakis, D. P. Greenberg, “Enhancing and Optimizing the Render Cache”, *Rendering Techniques ‘02*, (Proc. 13th Eurographics Workshop on Rendering), 2002
  - http://www.graphics.cornell.edu/research/interactive/rendercache/

Bibliography

- G. W. Larson, “The Holodeck: A Parallel Ray-caching Rendering System'’, *Proceedings of the Second Eurographics Workshop on Parallel Graphics and Visualisation*, September 1998.
- G. Ward, and M. Simmons, “The Holodeck Ray Cache: An Interactive Rendering System for Global Illumination in Nondiffuse Environments,'' *ACM Transactions on Graphics*, 18(4):361-98, October 1999.

Bibliography

- M. Simmons and C. H. SÈquin "Tapestry: A Dynamic Mesh-based Display Representation for Interactive Rendering”, *Rendering Techniques 2000*, (Proc. of the 11th Eurographics Workshop on Rendering), 2000
  - http://www.cs.berkeley.edu/~simmons/
- P. Tole, F. Pellacini, B. Walter, and D. P. Greenberg, ”Interactive global illumination in dynamic scenes”, *SIGGRAPH 2002 Conference Proceedings*, 2002.

Updates

- For the latest version of this talk, go to:
  - http://www.graphics.cornell.edu/~bjw/IDPCourse/