Advanced Computer Graphics
CS 563: *Adaptive Caustic Maps Using Deferred Shading*

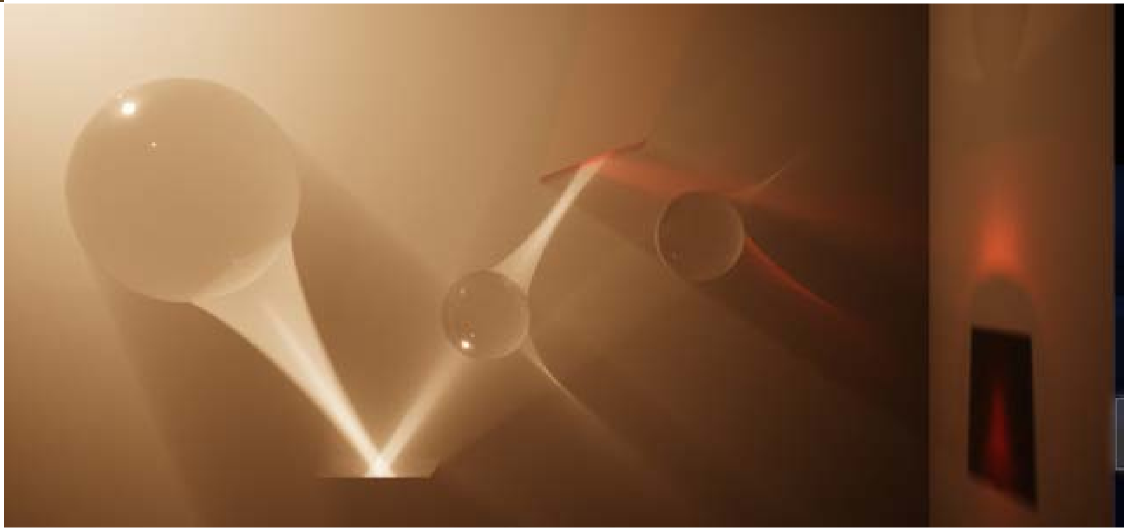
Frederik Clinckemaillie

Computer Science Dept.

Worcester Polytechnic Institute (WPI)

Introduction: Caustics

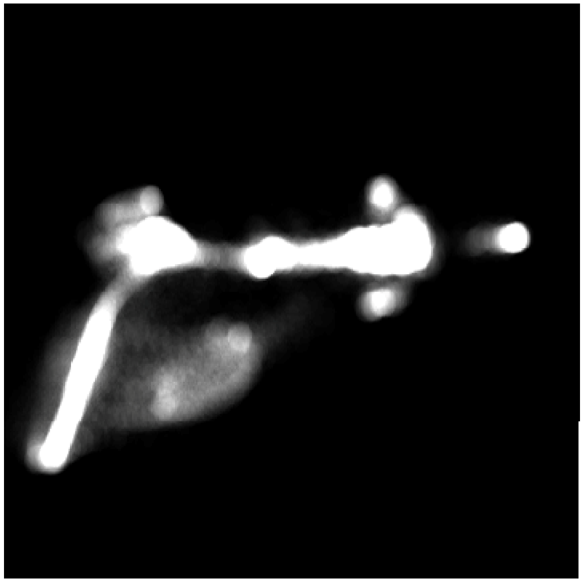
Reflective Caustics

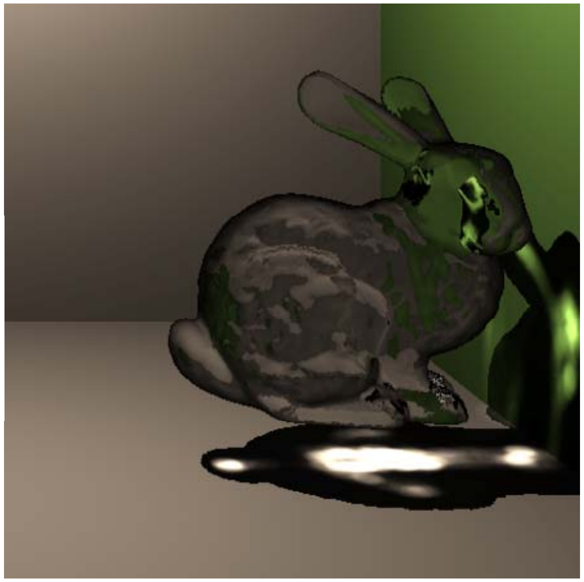


Refractive Caustics

Introduction: Caustic Mapping

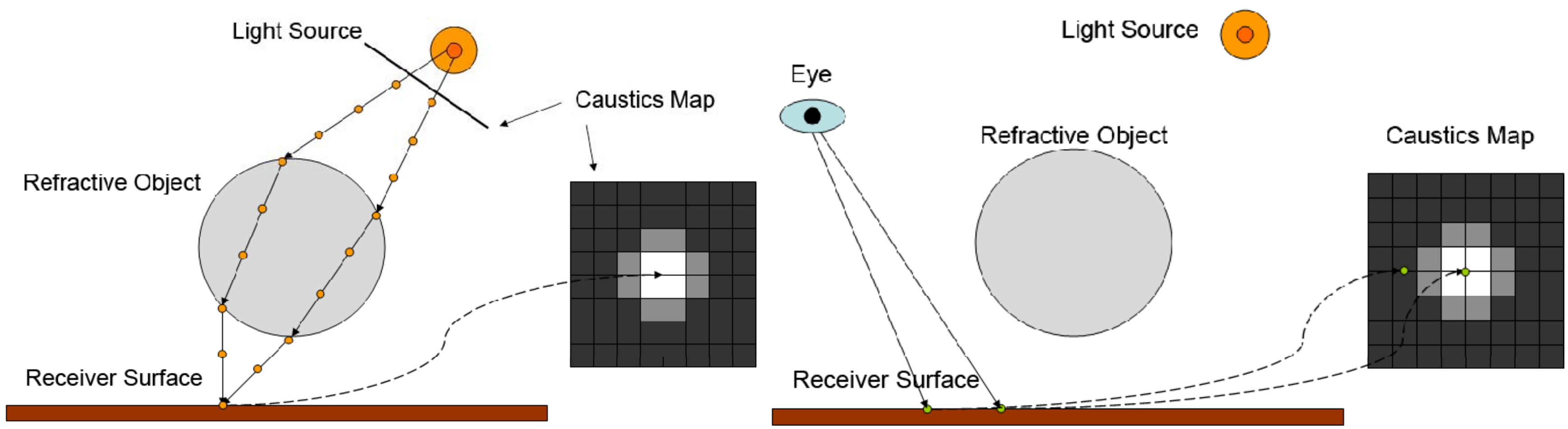
- Much faster than path tracing algorithms
- Two-pass process
  - Similar to photon mapping
- Creates a caustic intensity map





Caustic Mapping

- Three Step Process
  1) Photon Emission
  2) Rearrangement into Caustic Map
  3) Caustic Map Projection



Caustic Mapping

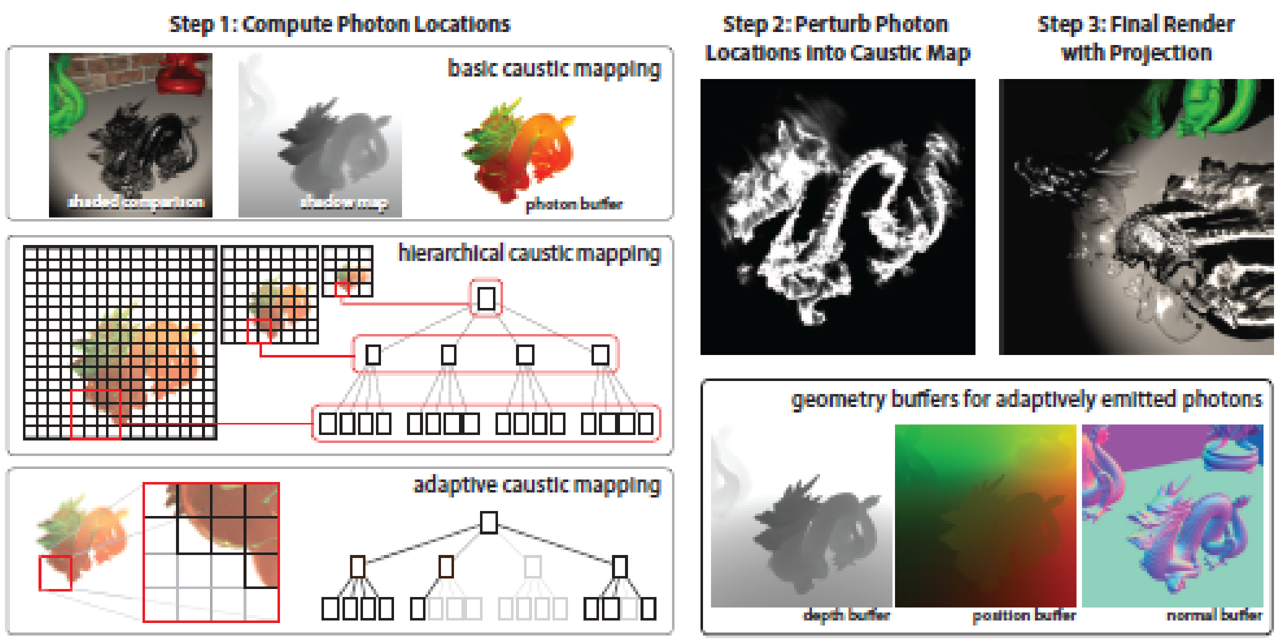


Figure 2: *Basic caustic mapping occurs in three steps. The first step creates a photon buffer storing the final hitpoints for photons interacting with specular surfaces; we provide a shaded light view and shadow map for comparison. The second step rearranges these photons by splatting into a caustic map. Finally, the caustic map is projected onto the scene. Hierarchical caustic maps improve performance by creating a mipmap-like hierarchy and processing in a top-down manner that avoids processing the entire photon buffer. We introduce a new, adaptive approach that never creates an explicit photon buffer. Instead, an adaptive deferred shading pass that point-samples the geometry buffers allows us to emit photons adaptively. This not only avoids processing the entire photon buffer, it never generates unused photons.*

Photon Emission

- Rasterizes from light view to generate grid of photons
- Creates photon buffer
  - 2D image storing final photon hit points
  - In conjunction with shadow maps
  - Allows quick lookups to determine Indirect lighting from caustics

Caustic Map Creation

- Controls lighting quality and cost
- Splatting photons into caustic map becomes bottleneck
- Crisp noise-free images require millions of photons
  - Not feasible in interactive time
- Hierarchical caustic maps: discard unimportant parts of photon buffer to improve speed
  - Uses multi-resolution caustic map to reduce splatting costs

Problems

- Poor photon sampling due to rasterization leads to under and over sampling
  - Proper sampling resolution cannot be determined
- Millions of photons are required for high-quality caustics. Processing each is too expensive
- Photon sampling location can change between frames, leading to coherency problems
  - Photon sampling is not dynamic

Deferred Shading

- Good sampling rates cannot be computed
- Ideally, number of photons is determined adaptively
- Hierarchical Caustic Map(HCM)
  - Has a maximum number of photons, not all are processed
- Photon Emission and Caustic Map Generation should be coupled

Deferred Shading

- Postpones final illumination computations until visible fragments are identified
- HCMs generates grid of photons and only processes relevant ones
- Deferred Shading never generates irrelevant photons.

Image-Space Refraction

- Image-space refraction requires four passes
  - Opaque geometry behind the refractor is rendered
  - Stores surface normals and depth for the backside of the refractor
  - Rasterizes the refractor
    - Approximates doubly refracted ray at each fragment
  - Refractor is combined with opaque geometry
- If refractor has depth complexity greater than two, extraneous shader executions occur for some pixels

Deferred Shading for Refraction

- Refraction Passes to store geometry buffers for both front and back refractor surfaces
  - 1. Render color and depth of geometry behind the refractor.
  - 2. Render back of refractor, storing normals and depth.
  - 3. Render front of refractor, storing normals and depth.
  - 4. Render a full screen quad, approximating refraction if the if the pixel lies on refractor
- Hidden Fragments are not shaded
- Step 2 & 3 can be combined in a single step

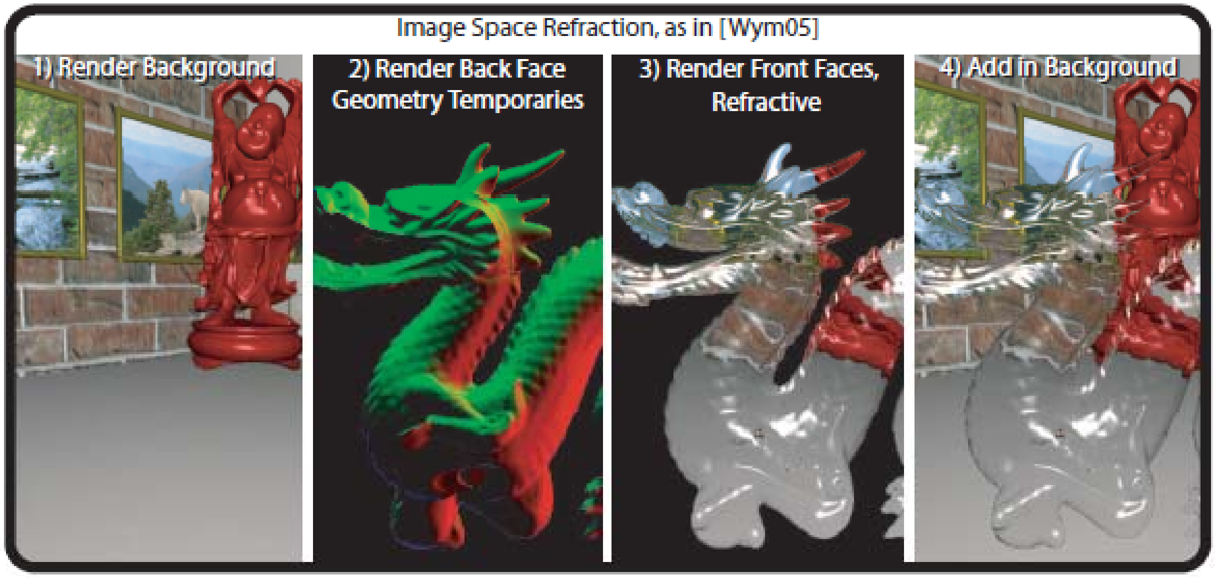
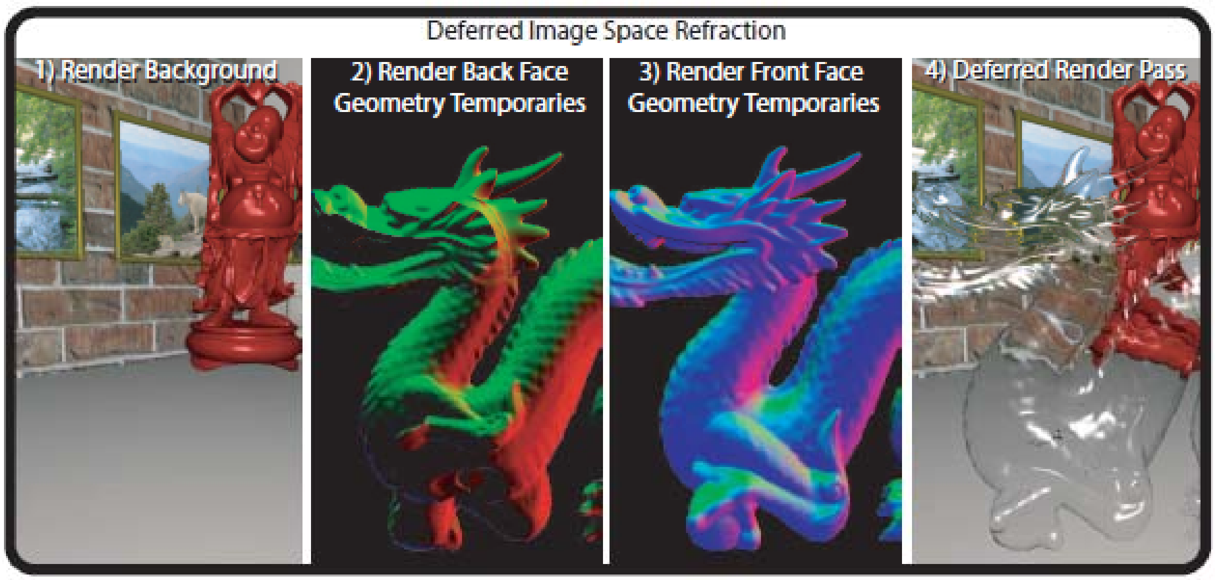


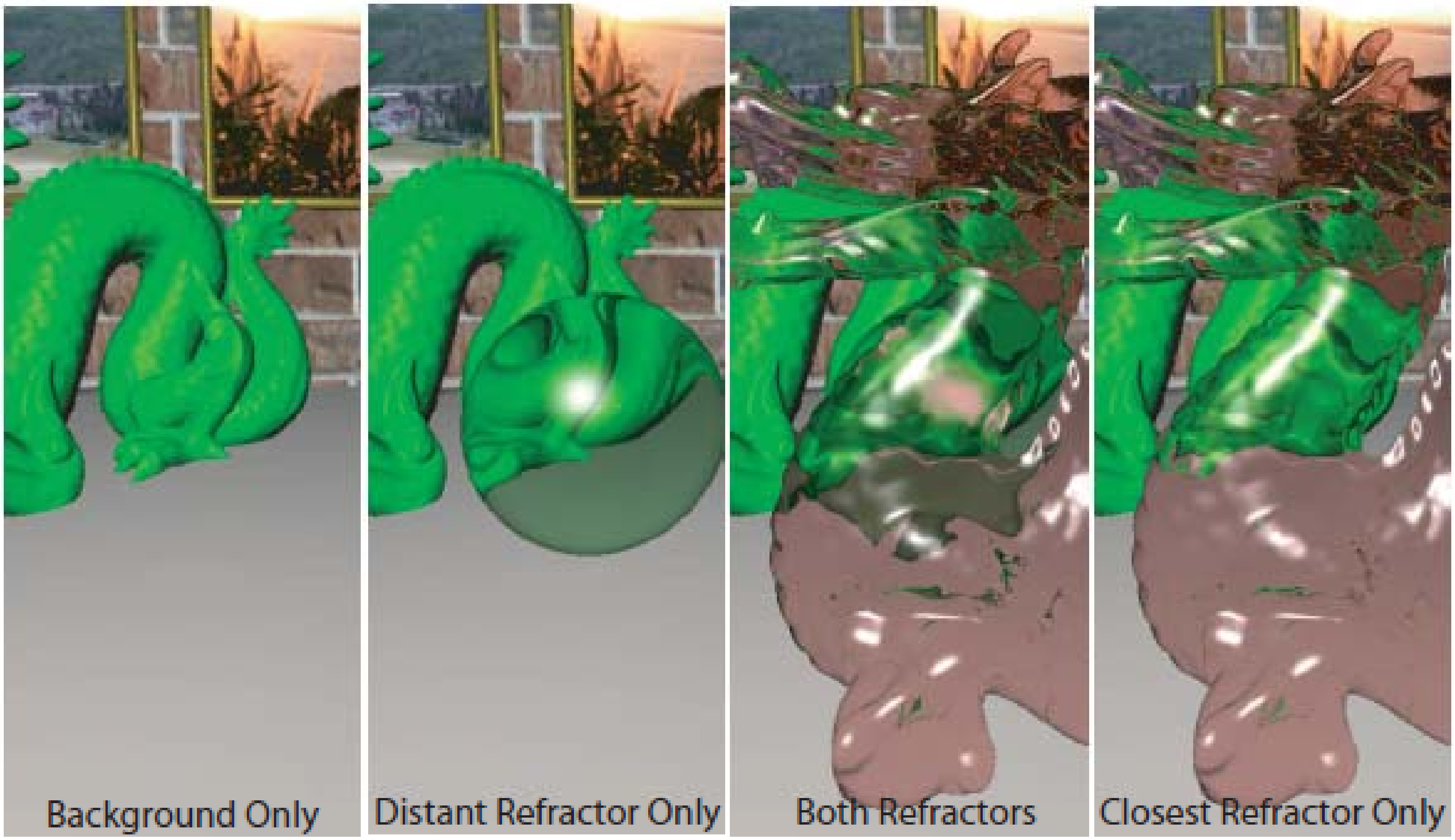
Image Space Refraction, as in [Wym05]
1) Render Background
2) Render Back Face Geometry Temporaries
3) Render Front Faces, Refractive
4) Add in Background


Deferred Image Space Refraction
1) Render Background
2) Render Back Face Geometry Temporaries
3) Render Front Face Geometry Temporaries
4) Deferred Render Pass

Multi-Layer Deferred Refraction

- Deferred Shading allows for rendering multiple refractors
- Processed via deferred shading from back to front:
  - Render Background
  - Render furthest refractor's geometry buffers
  - Render full screen quad to display furthest refractor
  - Render closer refractor geometry buffers
  - Render full screen quad to display final result, using step 3 result as background
- Refraction angles is incorrect at interfaces beyond the second

Multi-Layer Refraction



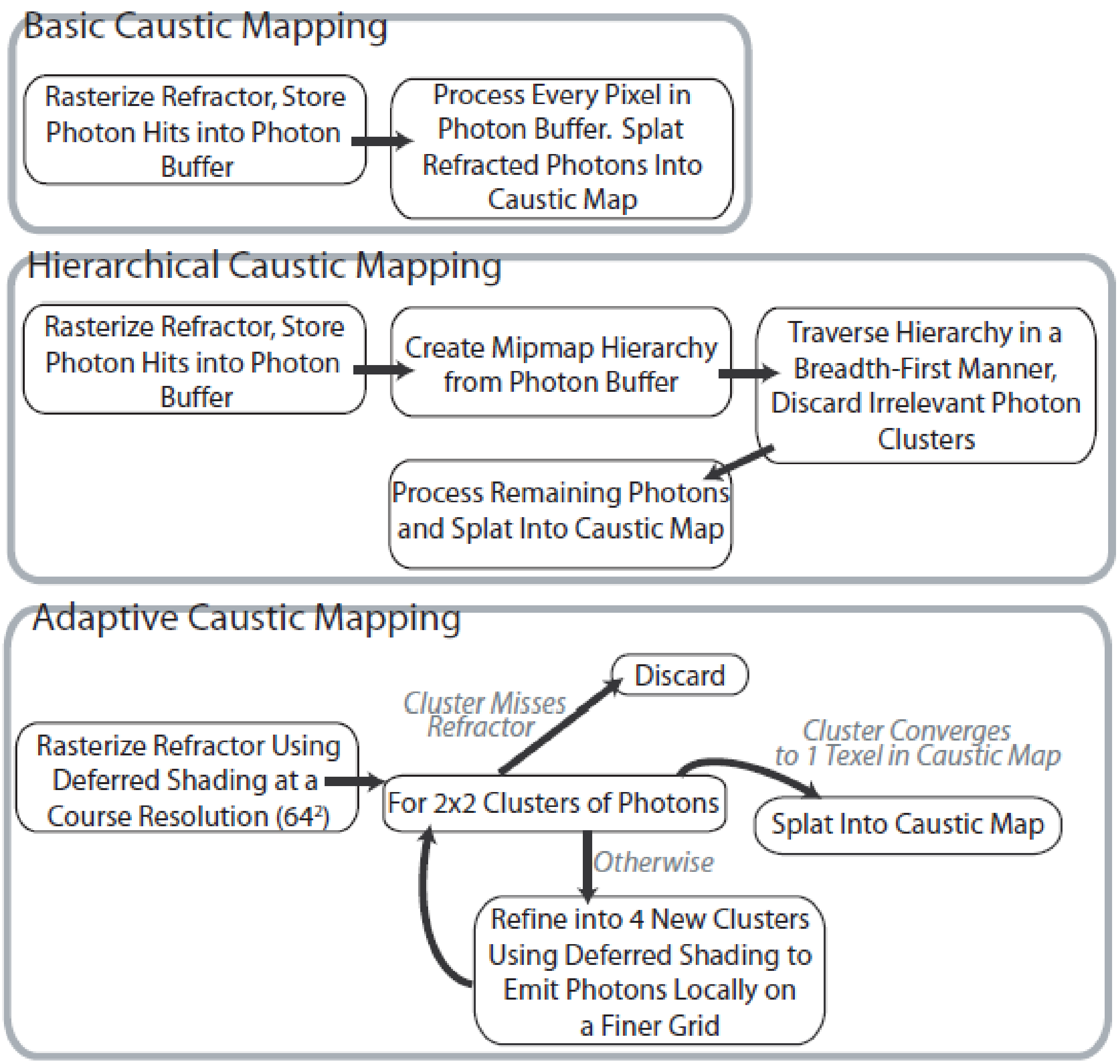
Deferred Shading for Caustic Rendering

- Pixels can be filled in any order in DS
  - All information is pre-computed
  - Allows us to create a photon buffer adaptively
  - Avoids creating irrelevant photons.
- Prior Algorithms approximate refraction by rasterization
  - In eye space, fixed-size image is final rendering
  - In light space, fixed-size image is photon buffer
- Deferred Shading avoids this limitation

Deferred Shading for Caustic Rendering (cont.)

- Start with a 64^2 grid of photons
- For each 2x2 cluster of photons
  - Discard the photons if they miss refractor
  - Splat onto caustic map if termination criteria is met
  - Refine cluster into four new cluster by generating new photons

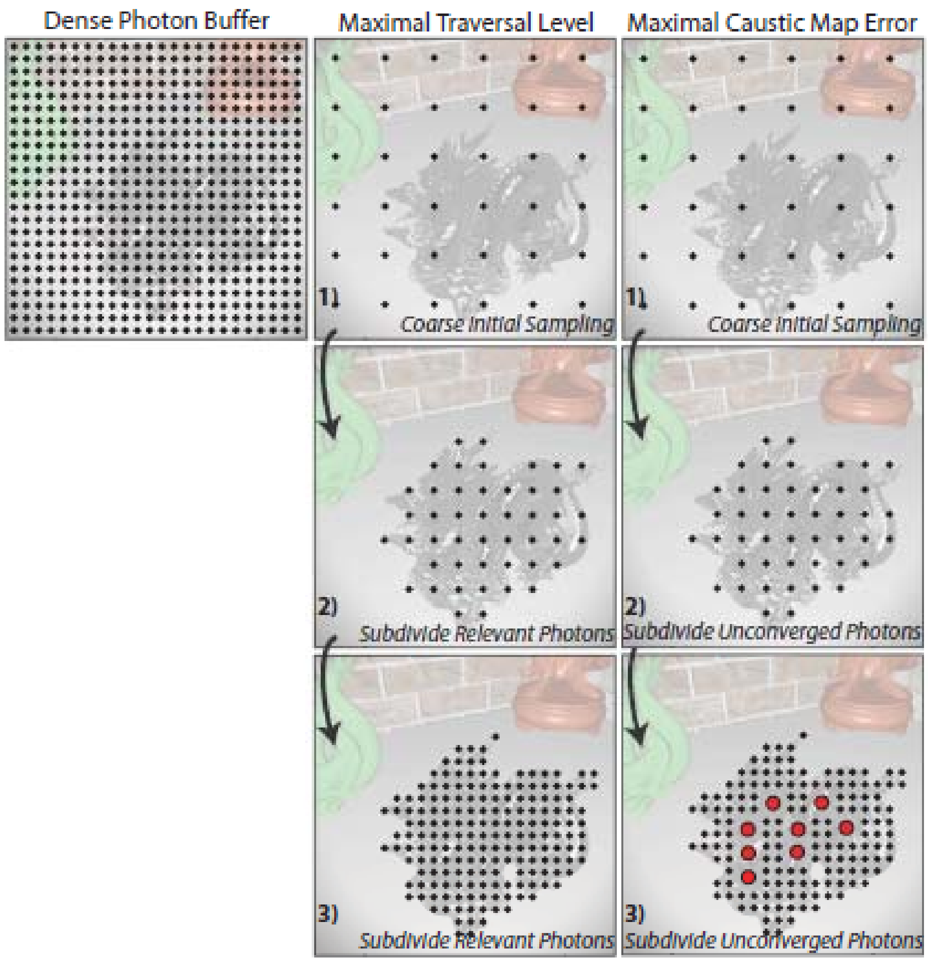
Comparison of Caustic Mapping Algorithms



Termination Criteria

- Metric: Maximal Traversal Level:
  - 1. If all photons in the current 2×2 cluster miss the refractor, none are output.
  - 2. When sampling has reached some maximal subdivision level, all remaining photons are output.
- Metric: Maximal Caustic Map Error:
  - 1. If all photons in the current 2×2 cluster miss the refractor, none are output.
  - 2. When all photons in the cluster converge to a single caustic map texel, one photon is output with intensity based upon the solid angle of all cluster photons

Termination Criteria Results



Maximal Caustic Map Error

- Additional Condition: Add maximal traversal level
  - Even after 14 levels (16384^2 grid), some photons remain above error threshold
  - Lists of photons often exceeds memory available on graphics accelerator
- Noise is eliminated by rendering to a multi-resolution caustic map
  - Lower resolution map for diverging photons

Error Metric Implementation

- Maximal Traversal Metric runs faster than Maximal Error Metric until buffer is larger than 8192^2 is used
  - Even though MTM generates more photons
  - Due to MEM requiring three kernels per traversal step
    - Computes photon hit positions
    - Identifies converged photons and outputs
    - Identifies unconverged photons and subdivides them

Avoiding Serial and Extraneous Processing

- Does not start with a single photon.
  - Begin with a 64^2 regularly sampled grid
- Use maximal Traversal Metric for first traversal steps
  - Coarsely sampled photons rarely converge to a single caustic map texel
  - Check convergence at 512^2 subdivision level

Lowering Memory Usage with Photon Batching

- Memory issues for large photon buffers
  - 8192^2 photon buffer requires 512 MB
  - relevant photons (10% of photons) requires top 50 MB
- Use Batches to reduce memory costs
  - Once user-defined memory limit is reached, photons are split into batches
  - Batches are processed one at a time
- Memory reduction sufficient to use grid of 65536^2
  - Using 16 batches, memory requirements are 16MB and 32MB temporary vs 250MB and 500MB temp.

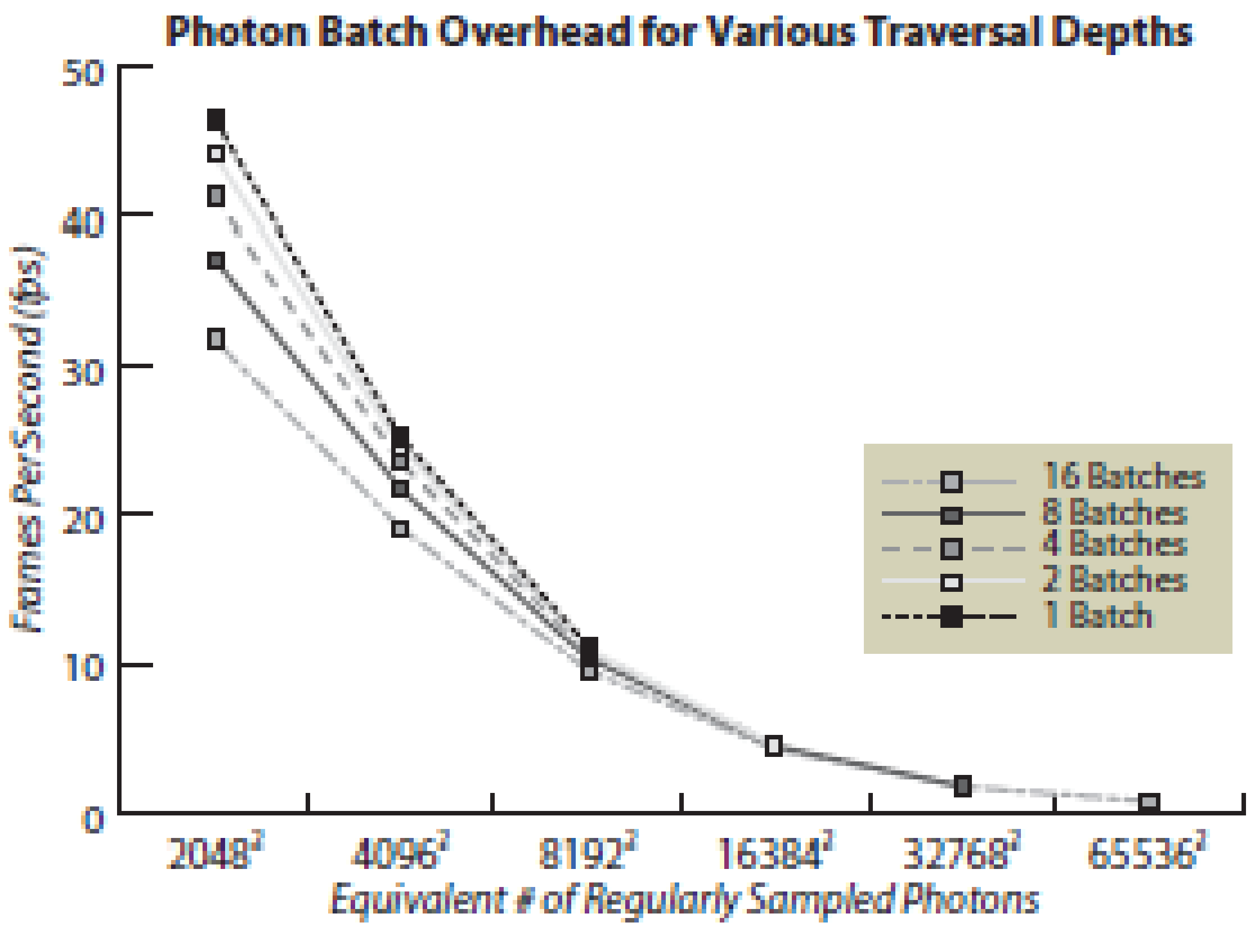


Figure 8: *Overhead from multiple photon batches for the Buddha scene. Without photon batching, traversing 14 levels (16384^2) in the photon hierarchy was impossible.*

Results and Discussion

- Generally, deferred rendering speeds refraction by 5-25%
  - Low polygon objects can perform 10% worse caused by temporary buffer

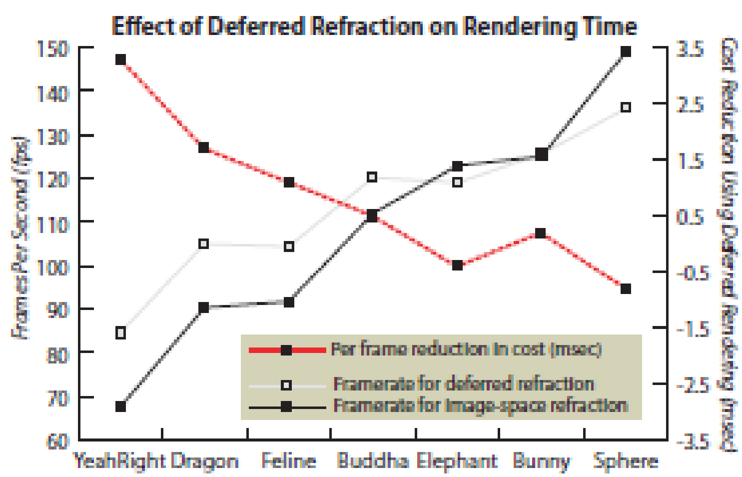


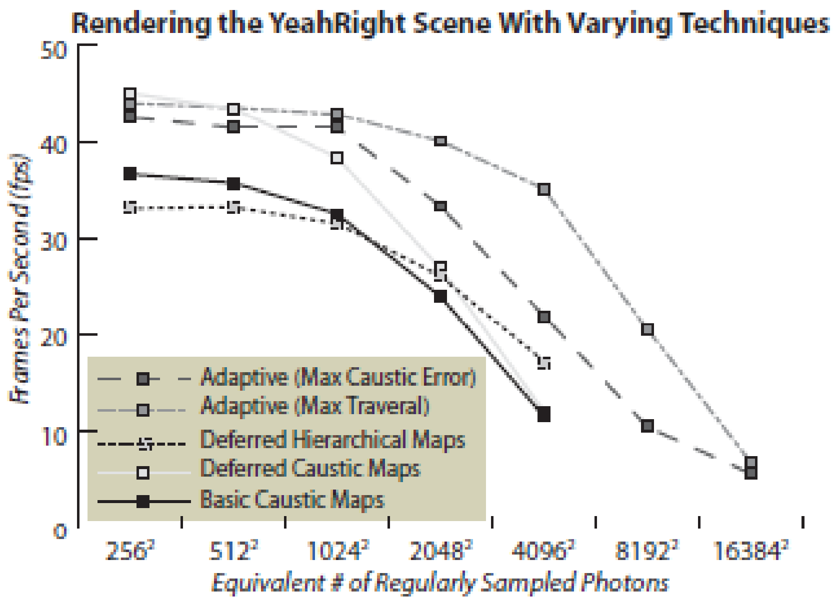
Figure 9: *Graph comparing costs of previous image-space refraction and deferred shading. Performance on low-polygon models actually degrades, as repeatedly rendering these models costs less than the overhead to defer shading.*

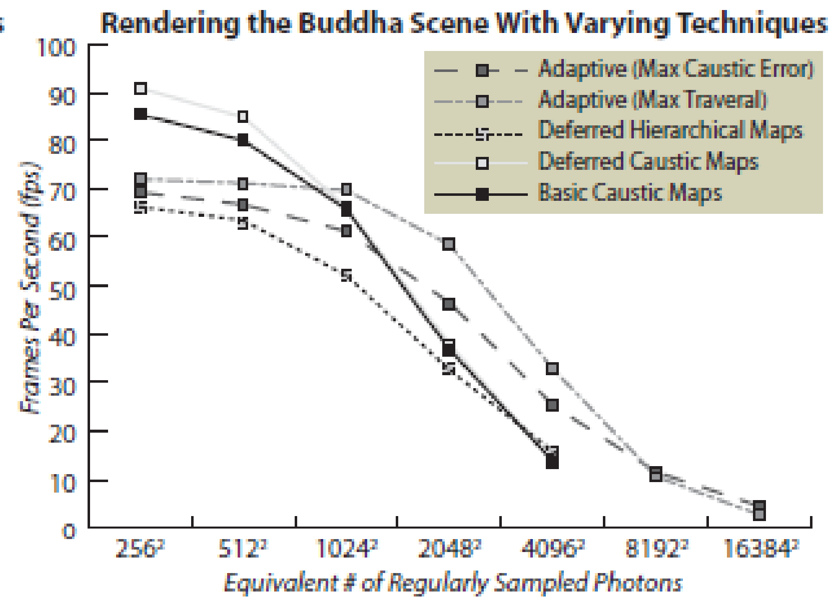
Results and Discussion (cont.)

Photon Buffer Res.	Photons Processed With Various Rendering Methods			
	Caustic Maps	Hier. Caustic Maps	Adaptive with Max Traversal	Adaptive with Max Map Error
256^2	65,656	4,803	5,092	4,556
512^2	262,144	19,861	20,160	19,380
1024^2	1,048,576	78,538	81,268	78,320
2048^2	4,194,304	286,298	326,896	266,768
4096^2	16,777,216	947,559	1,312,524	816,888
8192^2	67,108,864	N/A	5,210,492	2,182,704
16384^2	268,435,456	N/A	12,582,912	5,398,900

Results and Discussion (cont.)

- Adaptive Caustic Mapping runs faster with at least 1024^2 photons
  - Savings overcome traversal overhead





Results and Discussion (cont.)

- Multi-Layer refractions

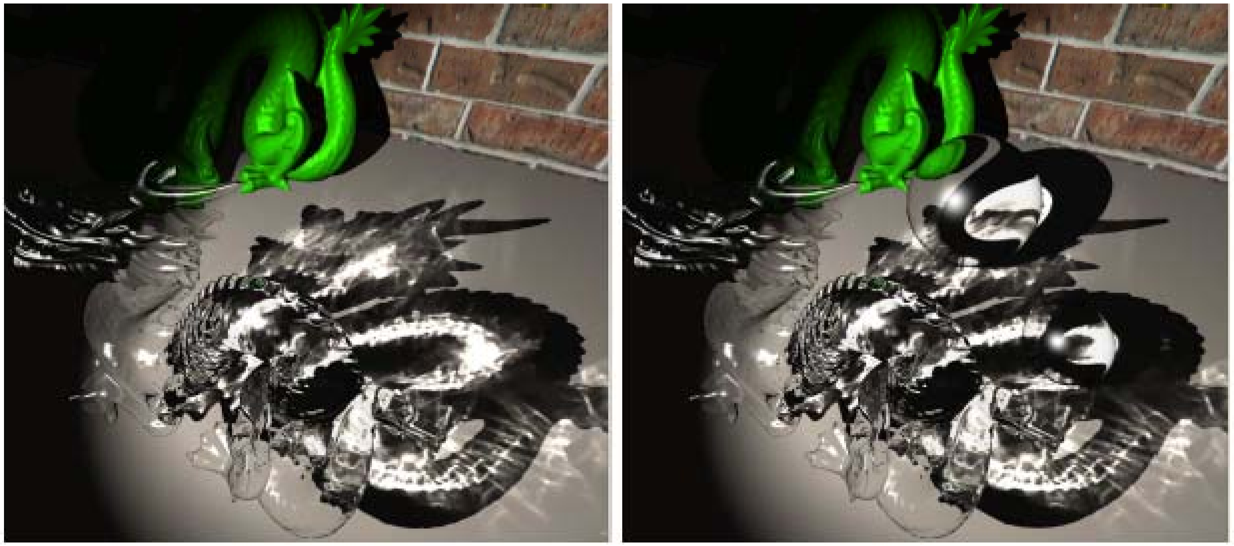


Figure 13: *Caustics from a refractive dragon. (Left) With just the dragon the scene runs at 35 fps. (Right) Adding two occluded spheres slows rendering to 34 fps.*

Results and Discussion (cont.)

- Comparison to Ground Truth

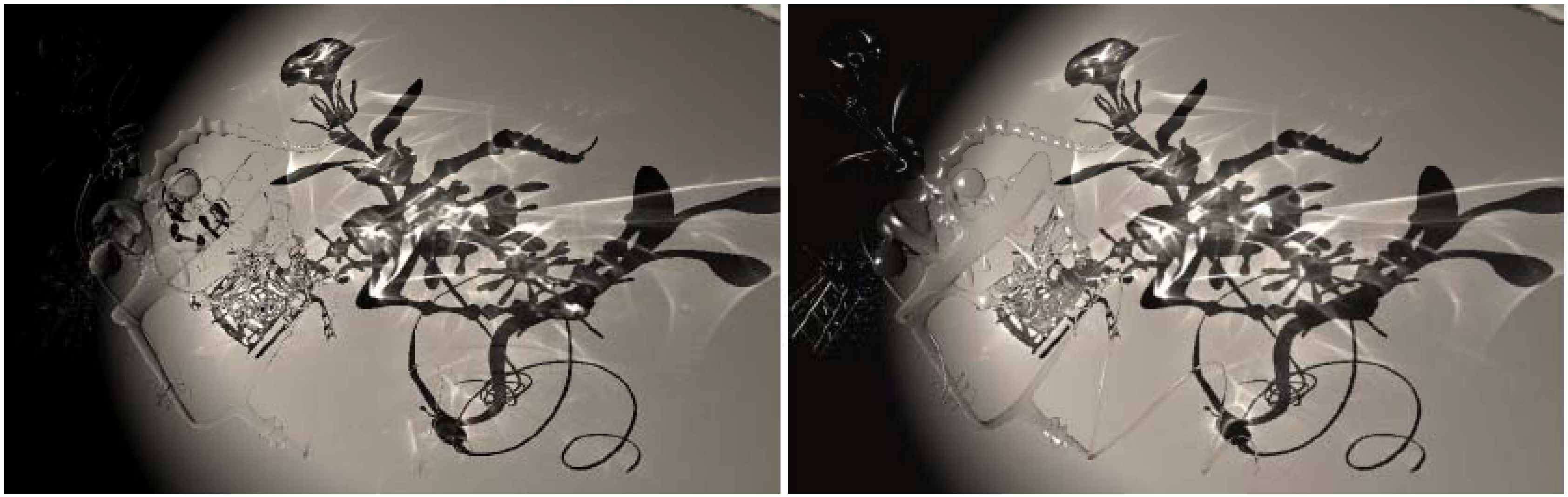


Figure 14: *(Left) Ground truth ray traced rendering of the "YeahRight" model using* 8196^2 *photons sampled on a regular grid. (Right) Interactive rendering at 21 fps using our adaptive technique and the same number of photons.*

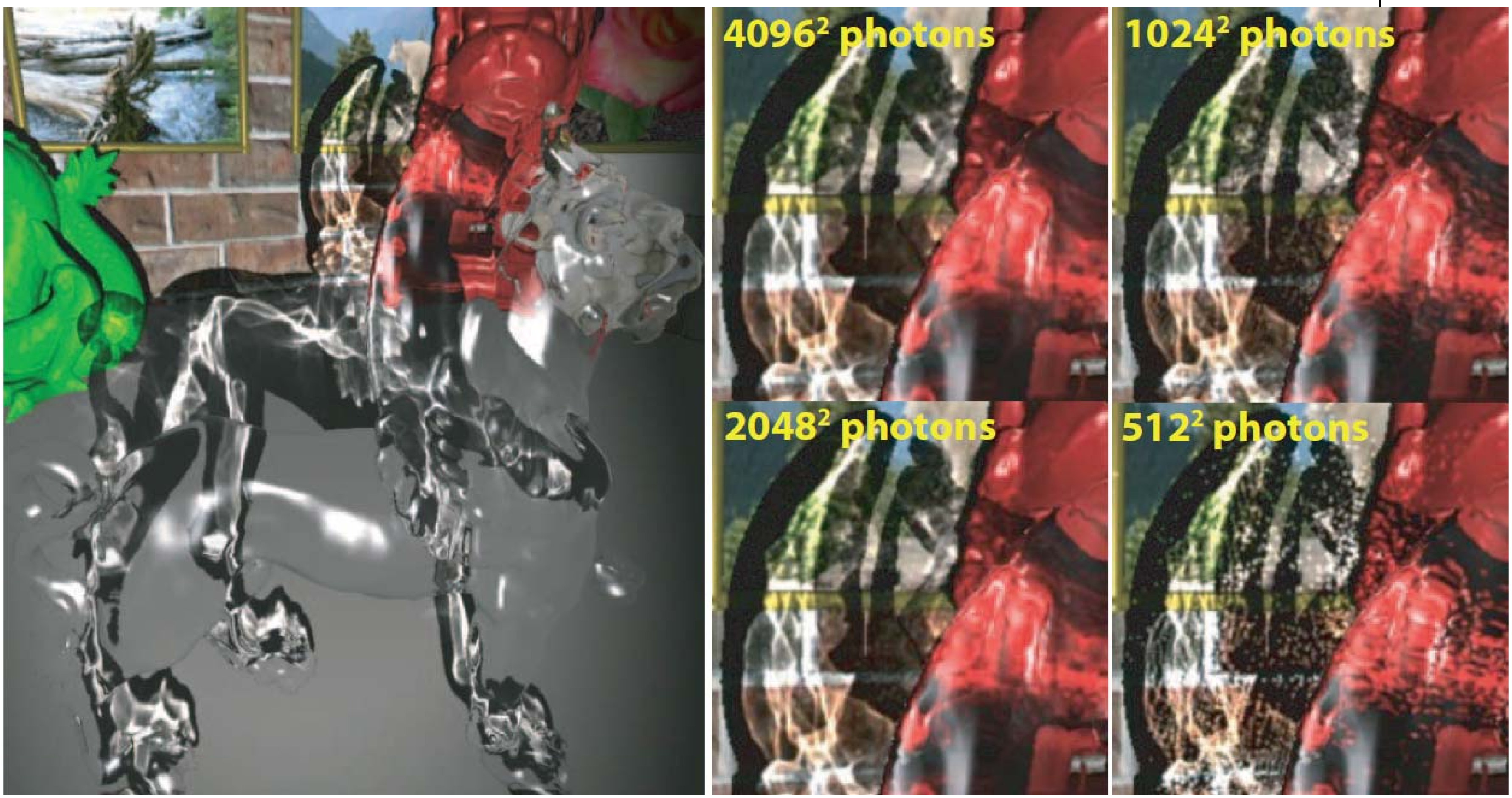
  - Interactive rendering only approximates true refraction
  - Ray traced appears noisier in unconverged regions due to final gathers

Benefits

- Render high quality caustics in interactive frame rates
- Photon count instead of refractor complexity controls performance
- Deferred shadowing avoids rasterizing extra geometries and allows multi-layer refractions
- Large number of photons can be simulated
  - Prototype handles equivalent of 524288^2 while responsive to user input

Limitations

- Decreases performance for small photon counts (below 1024^2)
- Maximal error metric underperforms Maximal traversal level for most reasonable sampling rates
- Only shown on refractive caustics



4096² photons
1024² photons
2048² photons
512² photons

References

- Wyman, C. and Nichols, G. (2009), Adaptive Caustic Maps Using Deferred Shading. Computer Graphics Forum, 28: 309–318. doi: 10.1111/j.1467-8659.2009.01370.x
- *graphics.cs.ucf.edu/gpuseminar/gamasutra_**caustics**.doc*