Adaptive Caustic Maps Using Deferred Shading

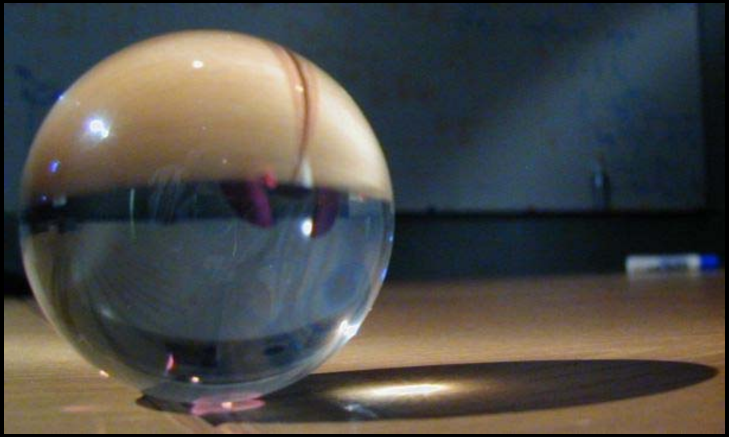
Chris Wyman   Greg Nichols

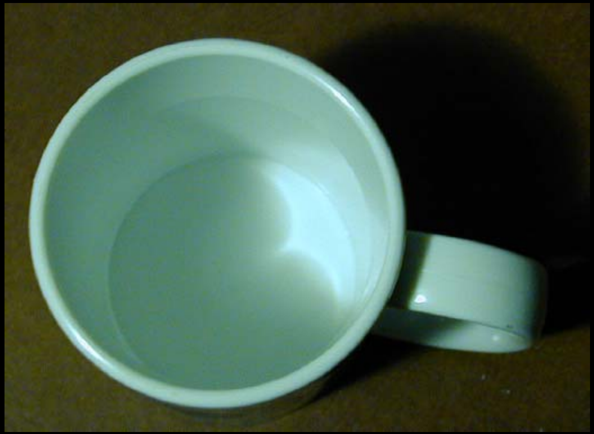
University of Iowa

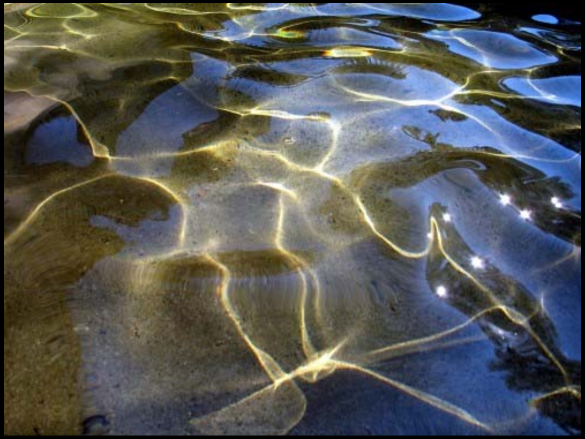
Eurographics 2009

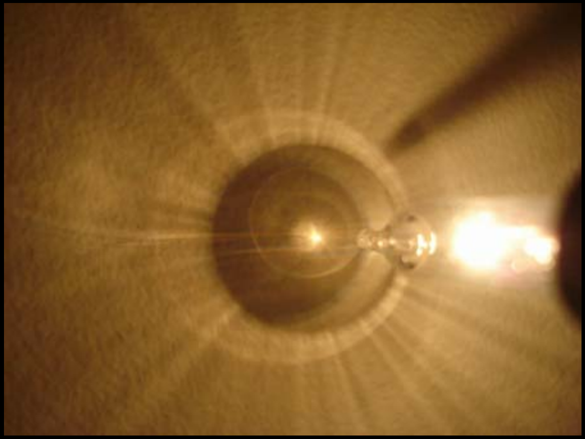
What is a Caustic?

Light focused via reflection or refraction





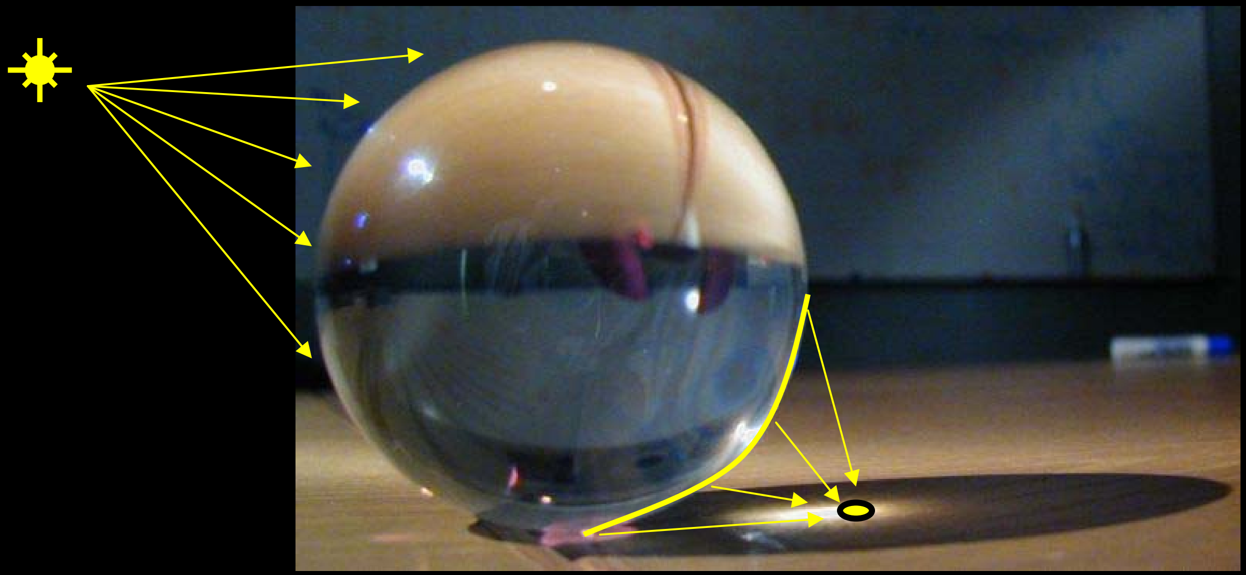




(Some photos used under a Creative Commons License, see Flickr group “Caustics,” users Rob Ireton, mag3737, ccgd, and audreyjm529)

Why Are Interactive Caustics Difficult?

- Depend on accurate specular interactions
  - Difficult in the context of rasterization
- Integrate over specular surfaces
  - Gather incoming illumination at each point
  - Essentially an “area” light – the area of the specular surface



THE UNIVERSITY OF IOWA

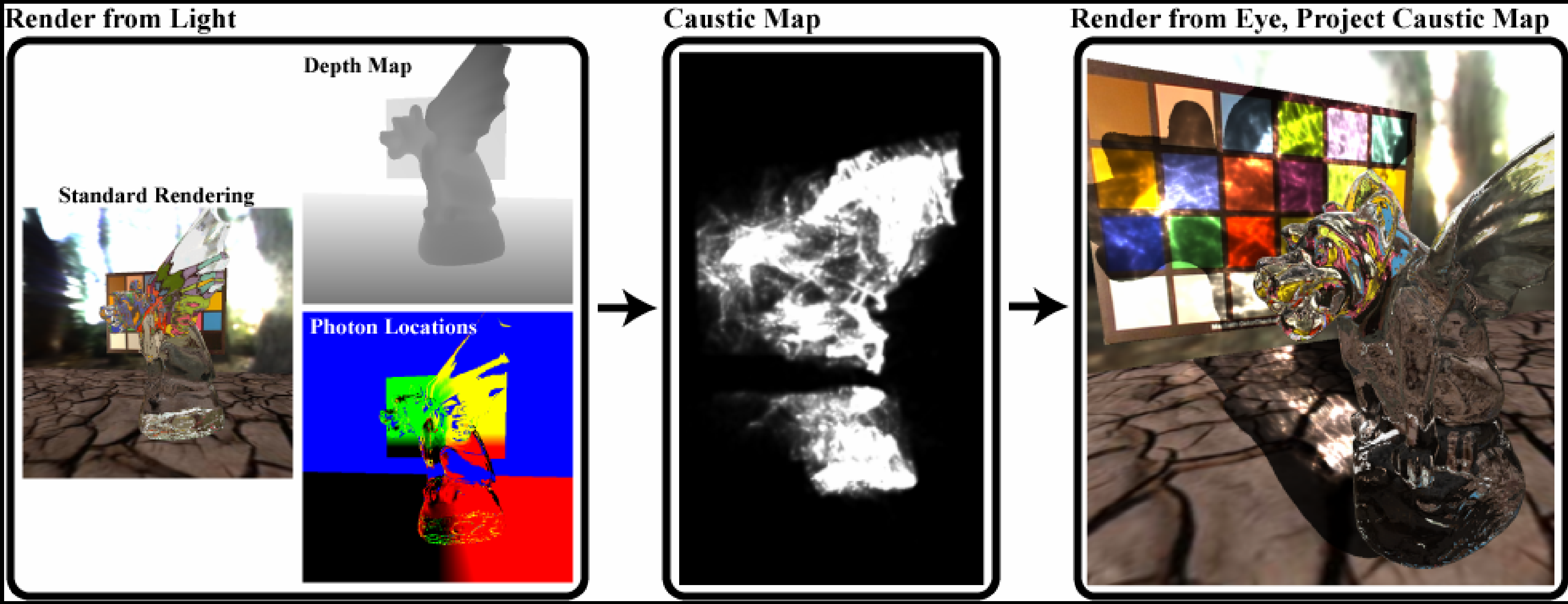
Prior Work

- **Wavefront techniques** *(non-interactive)*
  - First application for caustics [Mitchell 92]
  - March through volumetric scene representation [Ihrke 07]
- **Beam techniques** *(object-space interactive approaches)*
  - Backwards beam tracing [Watt 90]
  - Hardware accelerated approach [Iwasakai 02]
  - Warped volumes that accurately represent regions [Ernst 05]
- **Particle techniques** *(image-space interactive approaches)*
  - Illumination mapping [Arvo 86]
  - Photon mapping [Jensen 01] [Zhou 08]
  - Caustic mapping
    - [Szirmay-Kalos 05] [Wyman 06] [Kruger 06] [Shah 07]

THE UNIVERSITY OF IOWA

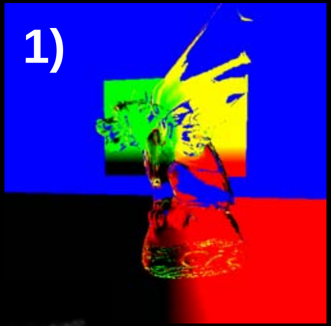
Caustic Mapping

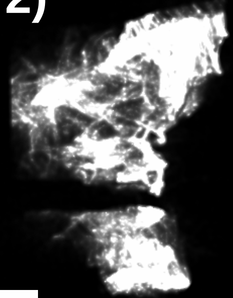
- Idea: Akin to shadow map
  - Image-space technique
  - Shadow map essentially stores binary visibility
  - Caustic map should store integral visibility

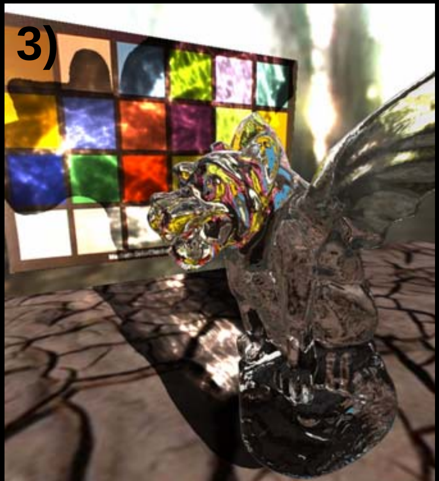


Caustic Mapping

1. Render from light, store photon hits
2. Treat photon hits as geometry
   - Draw into caustic map, with blending
   - Cost depends linearly on number of photons
3. Project caustic & shadow map



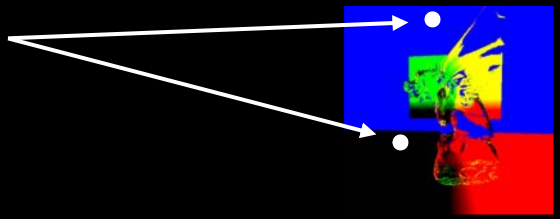




Caustic Mapping Issues

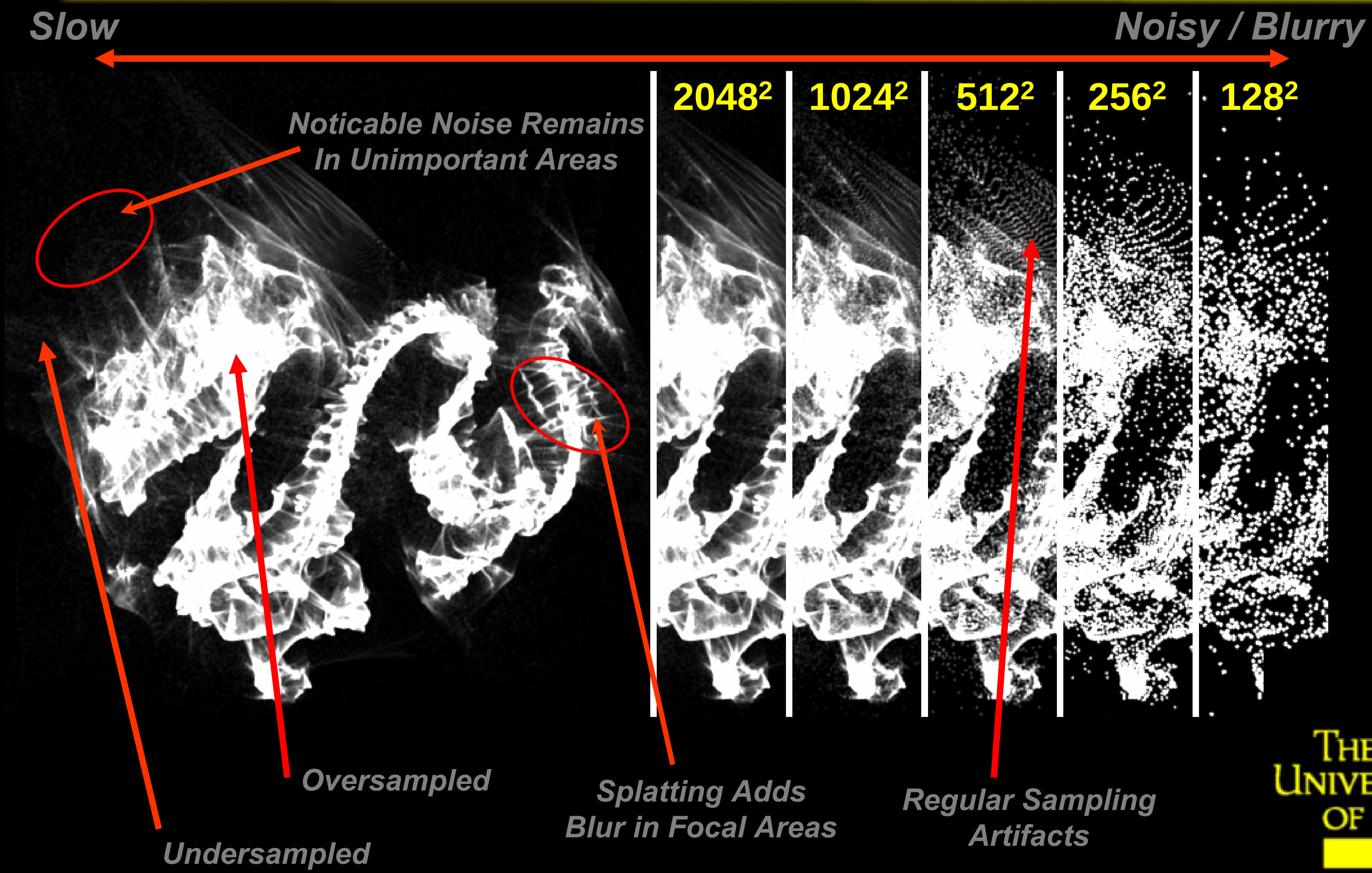
(for rasterization-based caustic maps)

1. "Wasted" photons → no caustic effect
   - These rays miss the specular objects
2. Bottleneck reversal
   - Photon map: emit photons; Caustic map: process photons
3. Regular photon emission
   - Photons "emitted" by rasterization (i.e., on a grid)
4. Over- and undersampling
   - Photons converge and diverge almost arbitrarily
5. Worst undersampling in least important areas
   - Introduces noise where blurring least objectionable
   - Introduces blur in areas with least noise
6. Number of photons limited by texture memory



The University of Iowa

Visual Depiction of Errors


Slow
Noisy / Blurry
2048^2
1024^2
512^2
256^2
128^2
Noticable Noise Remains In Unimportant Areas
Oversampled
Undersampled
Splatting Adds Blur in Focal Areas
Regular Sampling Artifacts

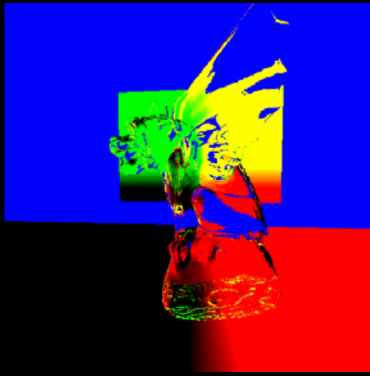
The University of Iowa

Partial Solutions

- Blur the caustic [Wyman06] [Shah07] [Szirmay-Kalos05] [Kruger 06] [Wyman08a]
  - Feels like cheating…. Caustics are *focused* light.
- Avoid point primitives [Wyman06] [Umenhoffer08]
  - E.g., use triangles or other primitives
- Hierarchical photon processing [Wyman08b]
  - Multiresolution blurring to keep details
  - Reduces cost of under or over sampling
- Different photon emission [Shah05]
  - Emit one photon towards vertices on specular surfaces

Adaptive Caustic Maps

- Observe:
  - Rasterization on fixed grid introduces problems
    - Generates extra photons that must be discarded
    - Limits sampling to regular grid, size fixed in advance

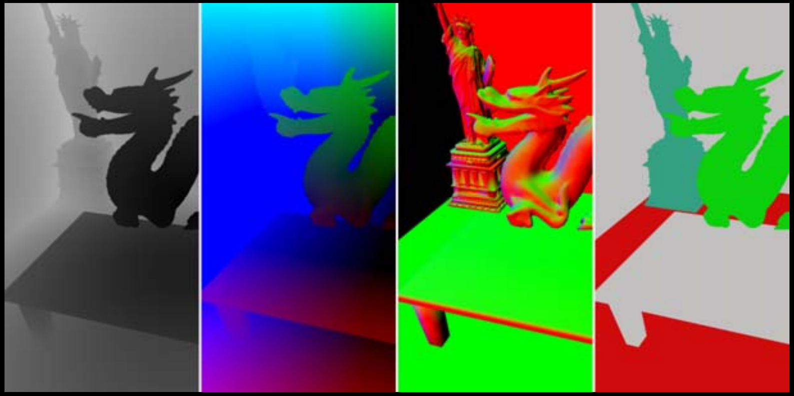


- Ideally:
  - Rasterize adaptively (e.g., on multi-resolution grid)
    - Use hierarchical rasterization [Greene96,Seiler08] ?
    - Use irregular z-buffer [Johnson05] ?
    - But: Still slow and infeasible with current hardware

The University of Iowa

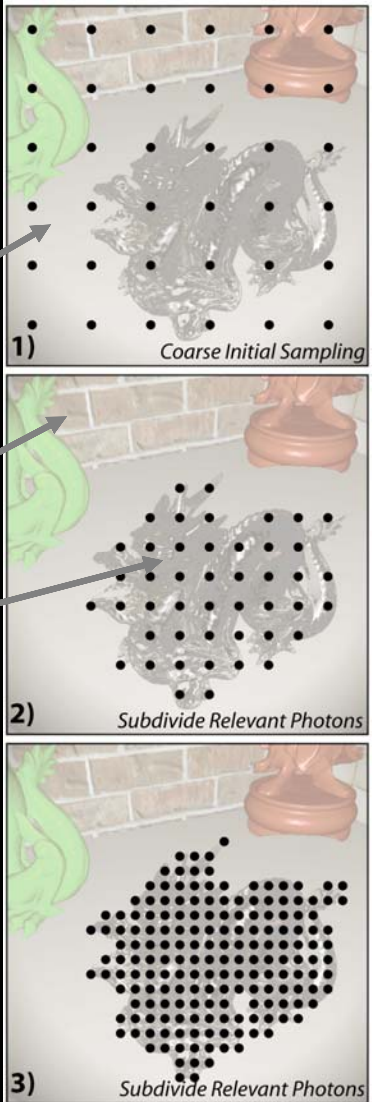
Adaptive Caustic Maps

- Let’s render caustics with *deferred shading*
  - Deferred shading goal:
    - Lighting = most expensive step
    - Avoid shading pixel multiple times using cheap prepass
    - Shade exactly fragments needed
  - How deferred rendering works:
    - First render temporaries → cheap, without shading
    - Render full-screen quad
      - Load data from temporaries, use to shade *exactly* once per pixel



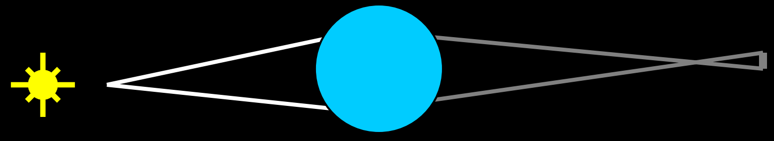
Alternate Deferred Rendering

- Key observation:
  - *Need not draw full-screen quad*
  - *Can perform final shading in any order*
- We choose an adaptive order
  - Start with coarse photon sampling
  - Discard photons that miss specular objects
  - Refine where photons bend to form caustics
- Stop refining when:
  - Reach some maximal refinement level
  - No error remains
  - Exceed a specified number of photons



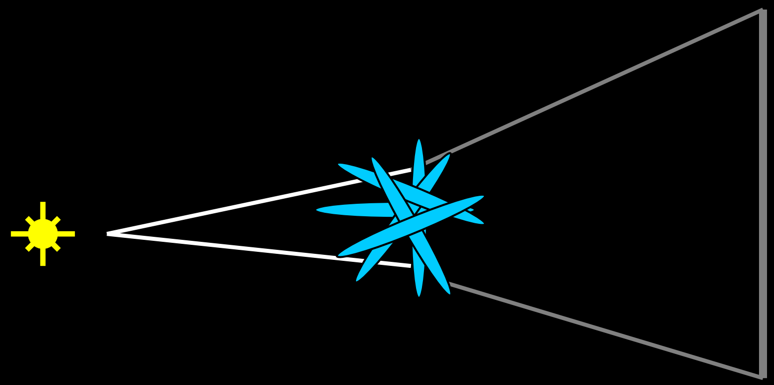
Refinement Criteria

- Need not refine all caustic photons
  - Perhaps some regions converge earlier



Photons converge (sufficiently)

→ *Additional refinement unnecessary*

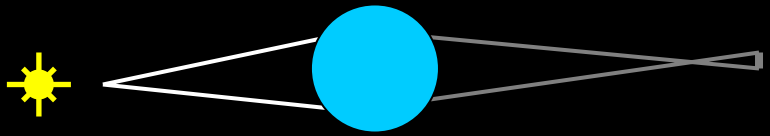


Photons diverge (sufficiently)

→ *Additional refinement needed*

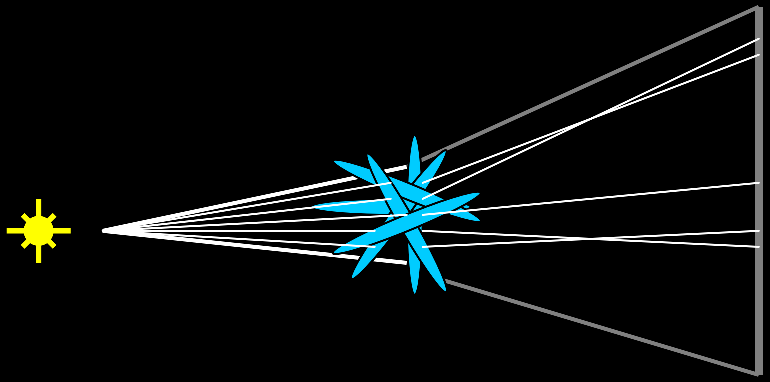
Refinement Criteria

- Need not refine all caustic photons
  - Perhaps some regions converge earlier



Photons converge (sufficiently)

→ Additional refinement unnecessary



Photons diverge (sufficiently)

→ Additional refinement needed

Some photons converge

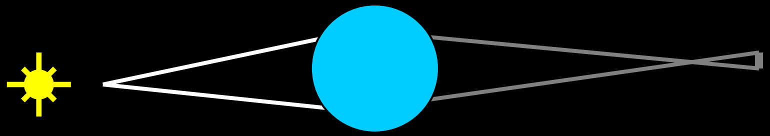
→ Additional refinement unnecessary

Others remain divergent

→ Additional refinement needed

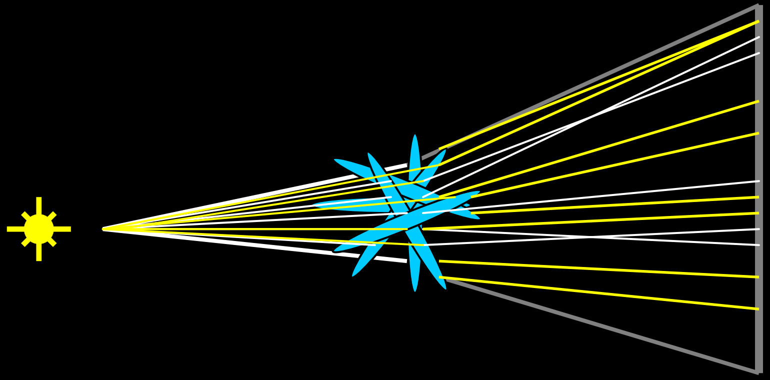
Refinement Criteria

- Need not refine all caustic photons
  - Perhaps some regions converge earlier



Photons converge (sufficiently)

→ *Additional refinement unnecessary*



Photons diverge (sufficiently)

→ *Additional refinement needed*

Some photons converge

→ *Additional refinement unnecessary*

Others remain divergent

→ *Additional refinement needed*

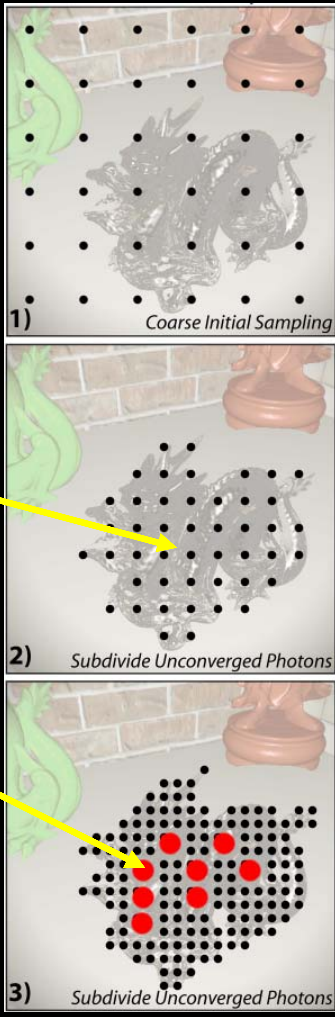
Some photons converge

→ *Perhaps reach max # photons*

→ *Use this set instead of original photons*

Refinement Criteria

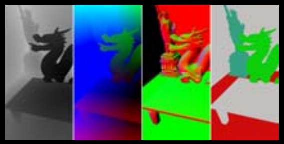
- Only refine if:
  - Maximal refinement level not exceeded
  - Error in neighborhood exceeds threshold



All photons exceed threshold

Error threshold not reached here (i.e., photons converge)

- Caveat:
  - Refinement limited by temp buffer resolutions
  - But, interpolation on positions & normals possible
    - Interpolation on very non-linear photon buffer not



Adaptive Caustic Map Summary

From light view:

- Render G-buffers for refractive objects
  - *Front & back surface normals, fragment depths. Details in paper.*
- Coarsely sample the light view (i.e., “emit photons”)
- For each 2 x 2 cluster of photons, check for convergence
  - If converged → used converged photons
  - If not converged → subdivide more finely
    - *Either recursively or iteratively*
    - *We use a NV_transform_feedback loop to iteratively divide diverging photons (up to a user-specified number of times)*
- Take the list of all photons, render into caustic map

From eye view:

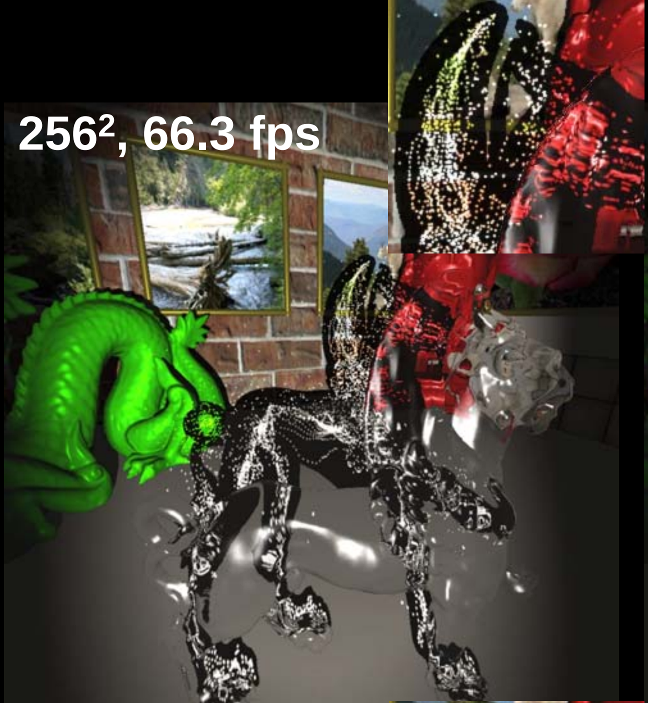
- No change (i.e., project caustic map onto scene)

Effect of Adaptive Approach

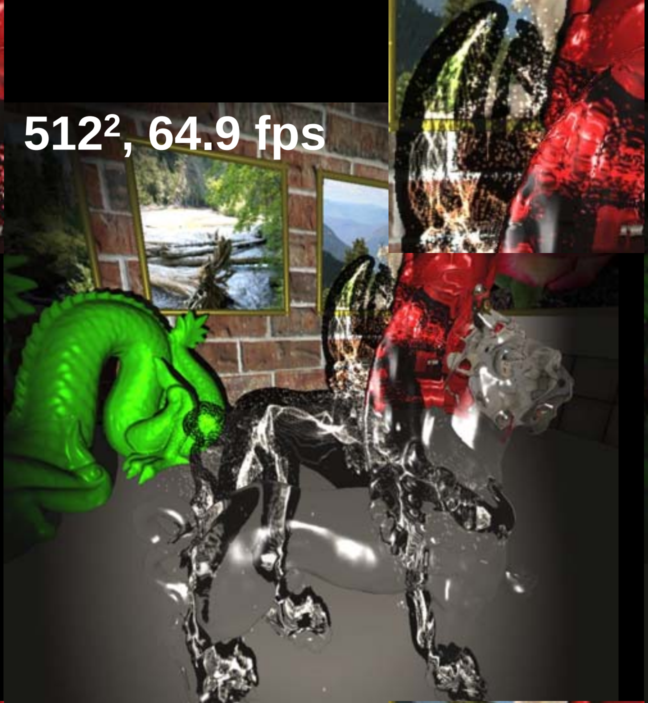
- Avoids explicitly creating the "photon buffer"
  - Generated implicitly in a multiresolution fashion
  - # photons not limited by max texture size on GPU
    - Since the entire buffer is never stored!
  - Never create "wasted" photons

- Adaptively emit photons on GPU
  - Reduces both over- and undersampling
  - Avoids emitting photons at too-fine resolution
  - Complex areas can be refined "arbitrarily"
    - We refined up to 19 levels
      - I.e., equivalent to a 2^{19} x 2^{19} (or $524,288^2$) photon buffer
      - Still "responsive" (running at 0.3 fps)
    - Previous maximum photon buffer was 4096^2

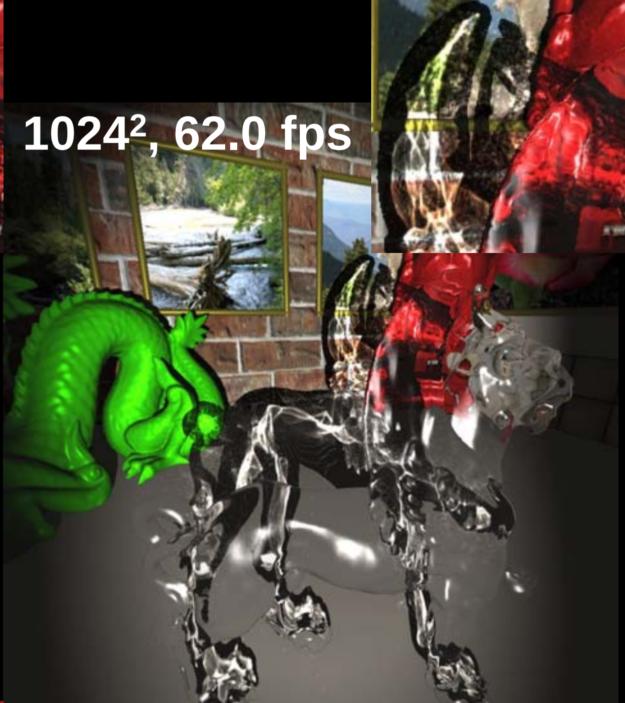
THE UNIVERSITY OF IOWA



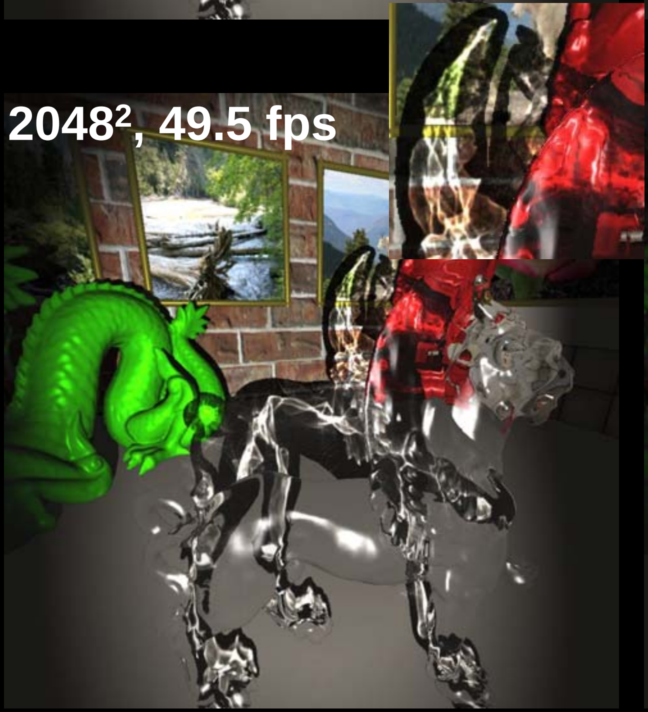
256^2, 66.3 fps



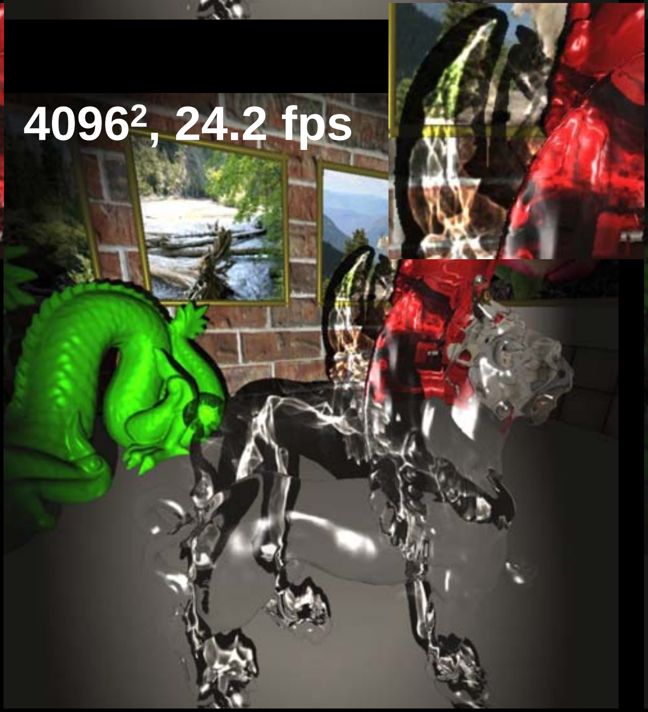
512^2, 64.9 fps



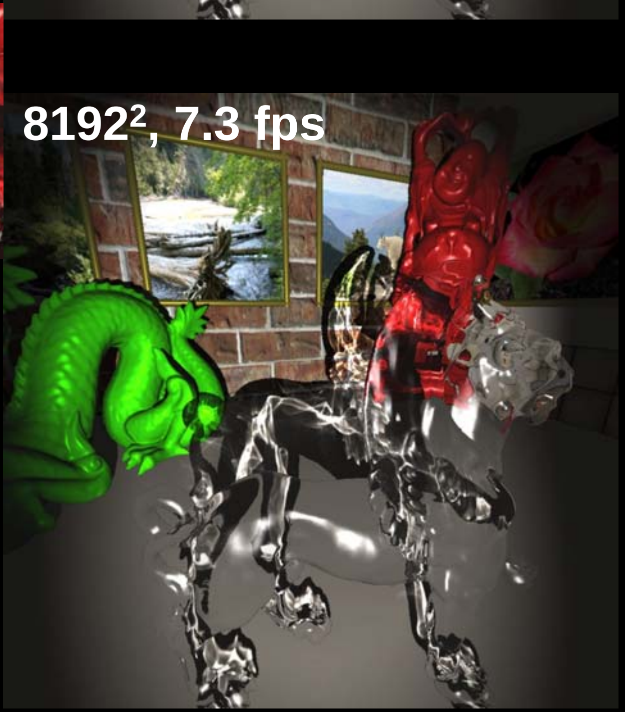
1024^2, 62.0 fps



2048^2, 49.5 fps

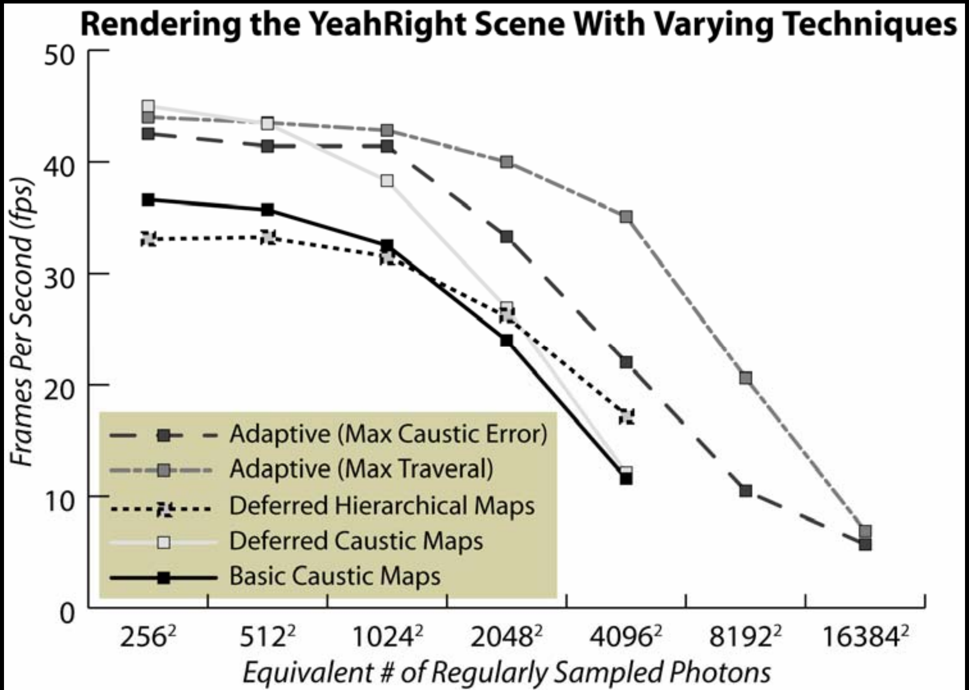


4096^2, 24.2 fps



8192^2, 7.3 fps

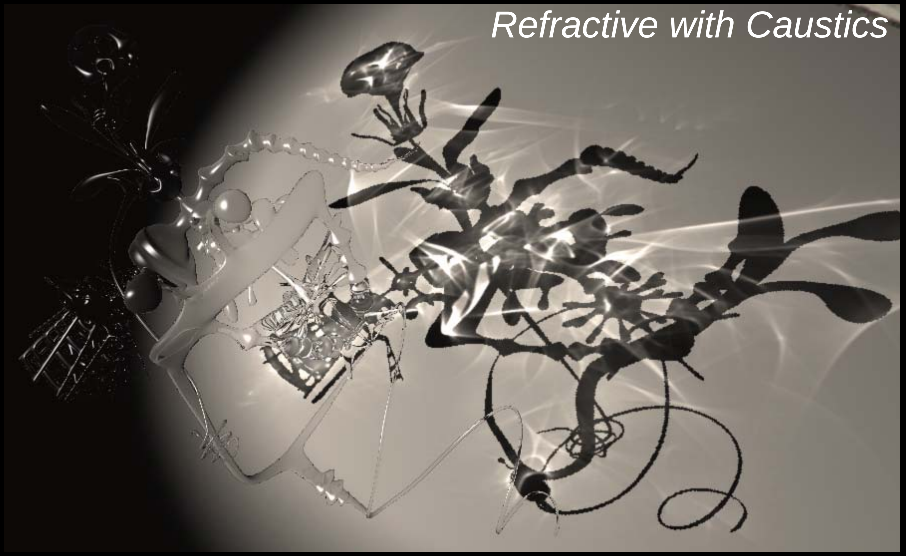
Performance Results



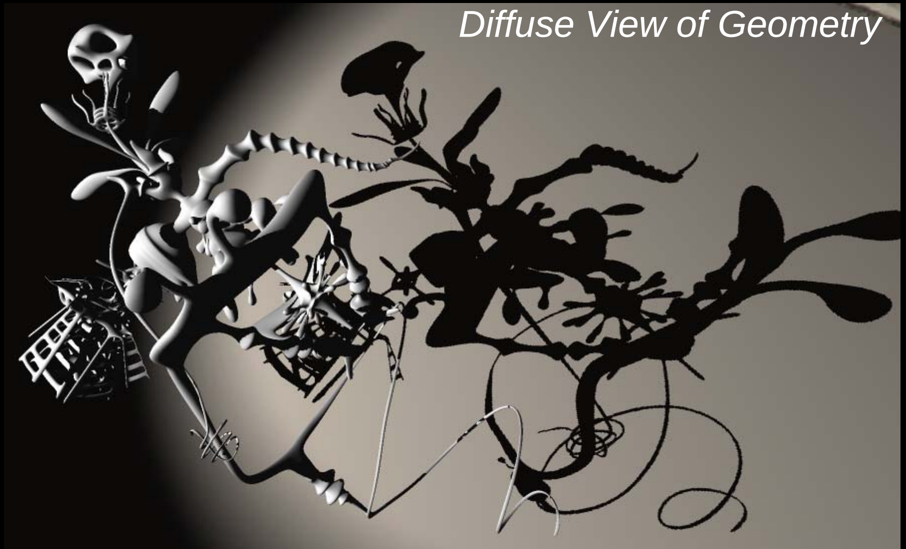
Yeah Right Model:

→ 755,000 triangles

→ Genus 130

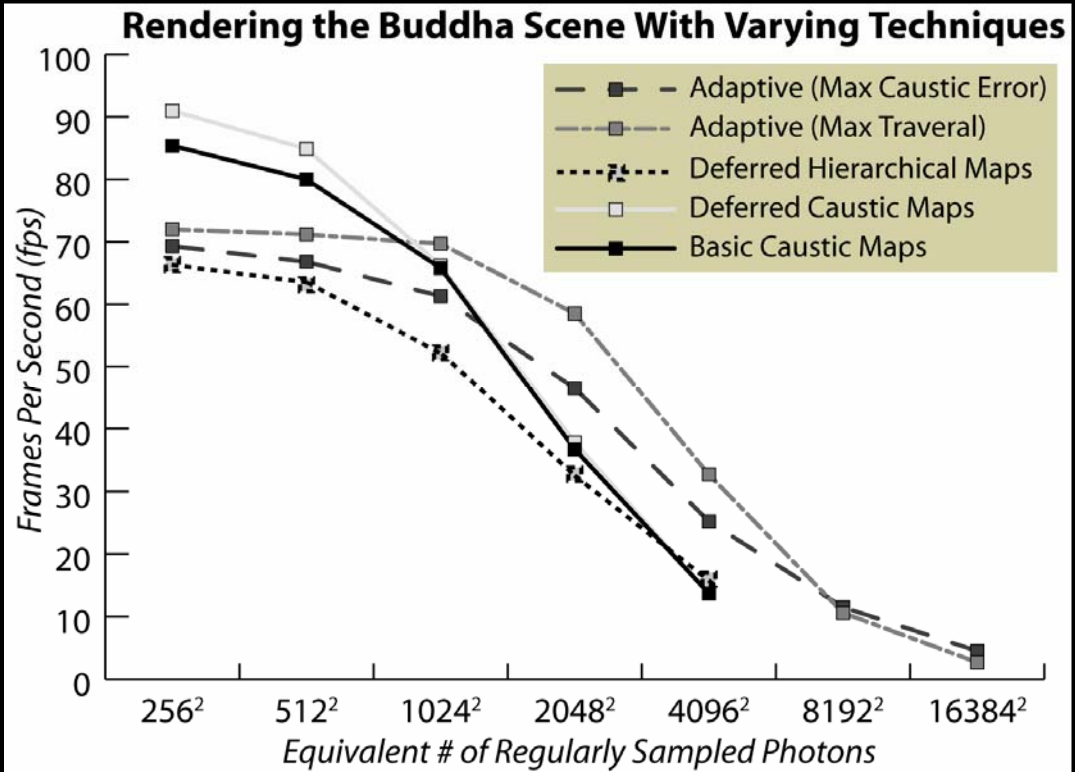


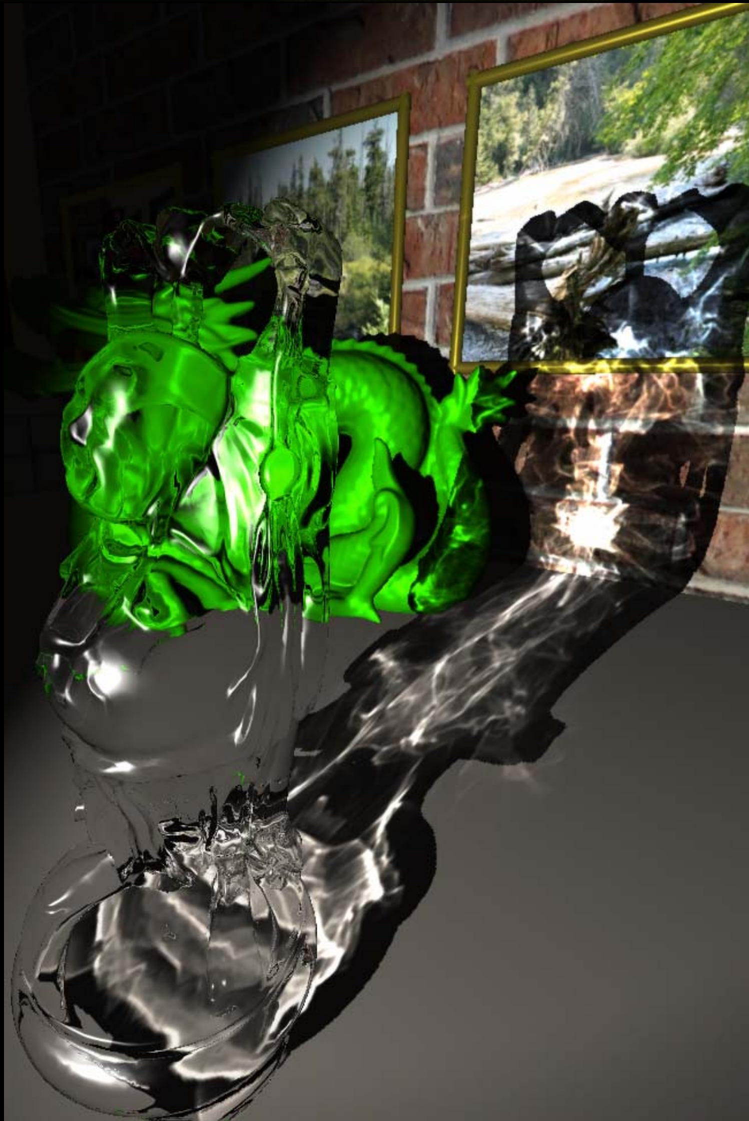
Refractive with Caustics



Diffuse View of Geometry

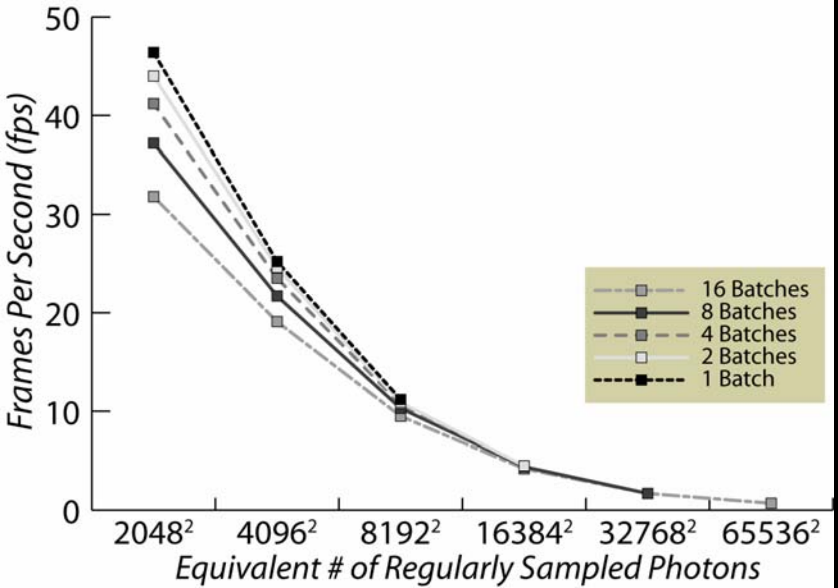
Performance Results

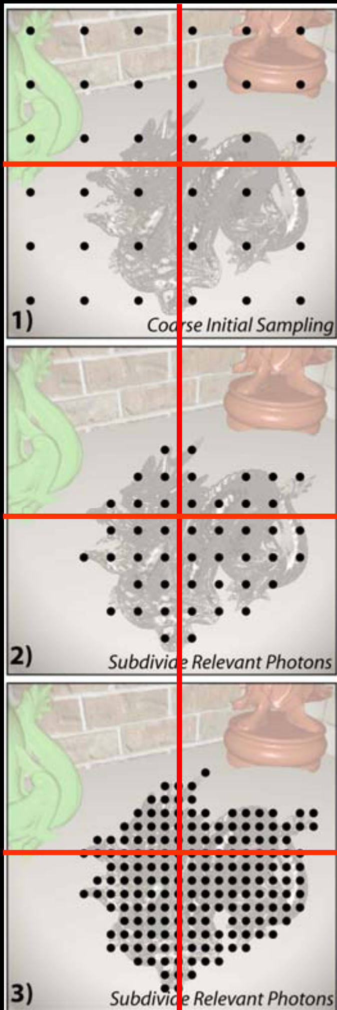




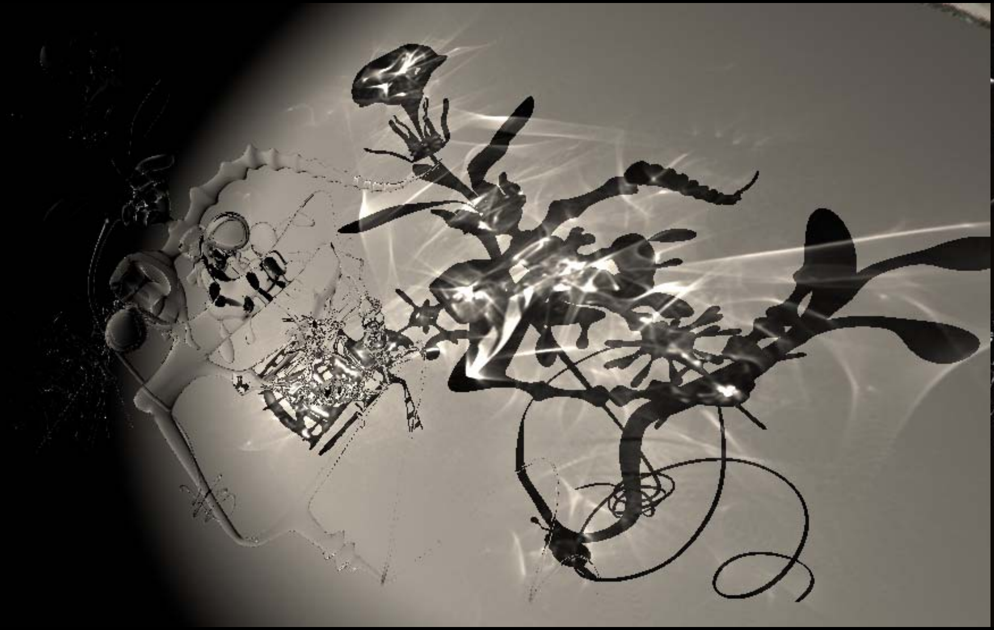
Not Limited By GPU Memory

- Need not perform breadth-first adaptive traversal
  - Split coarse photons into “batches”
    - (Or split after a few traversals)
  - Process one batch at a time
  - Render into caustic map
  - Discard batch memory prior to next batch

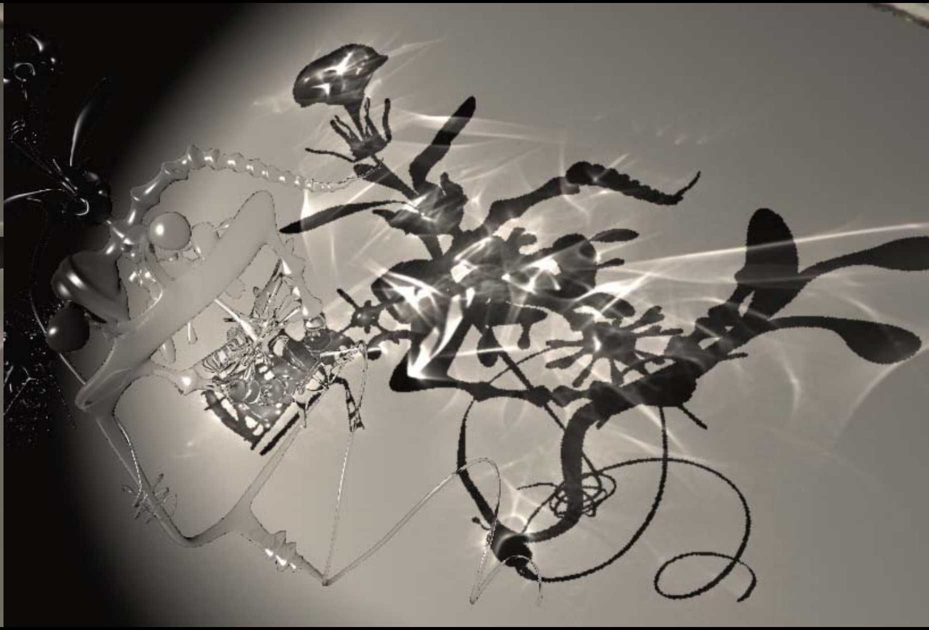




Ground Truth Comparison



Ray Tracing (~5 minutes)



Adaptive Caustic Maps (21 fps)

Video

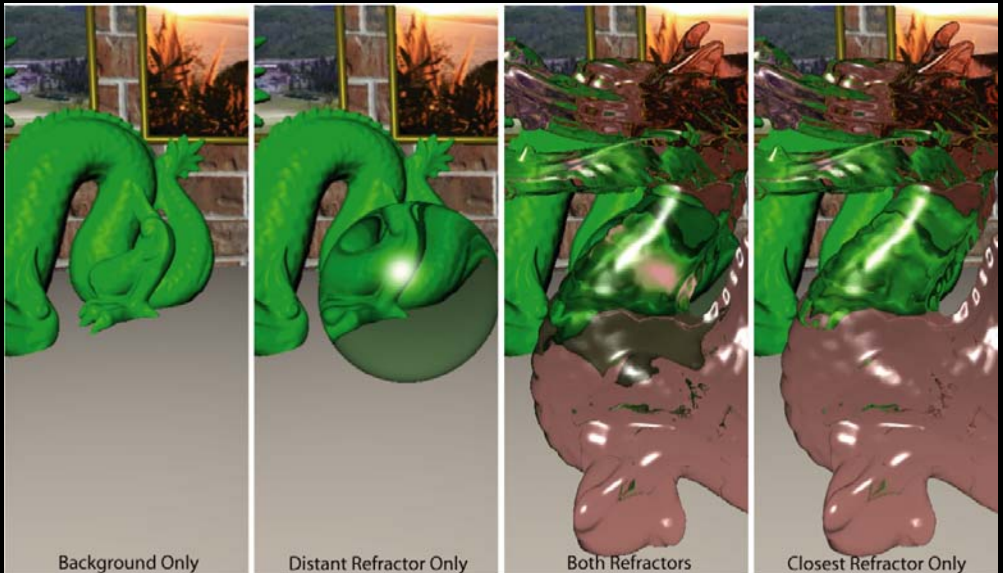
Problems

- Complex refinement metrics poorly supported
  - Simple refinement often faster
    - I.e., “refine all photons to a maximal level”
    - Even though more photons generated & processed
    - Complex refinement criteria hard for GPUs
      - Requires extra passes through photon stream
- Still aliased due to caustic map resolution
- Refraction approximations introduce artifacts
- Still spend lots of time in “unimportant” areas
  - Combining with selective blurring helps

THE UNIVERSITY OF IOWA

Also: Multi-Layer Refraction

- As an extra benefit:
  - Deferred shading allows multiple layers of refractions
  - All data stored in buffers prior to shading
    - No longer have standard rasterization problem of not knowing occluded geometry
  - Paper describes simple approximation extending [Wyman05] to refract through multiple interfaces
    - Very cheap, though secondary refractions are only approximate



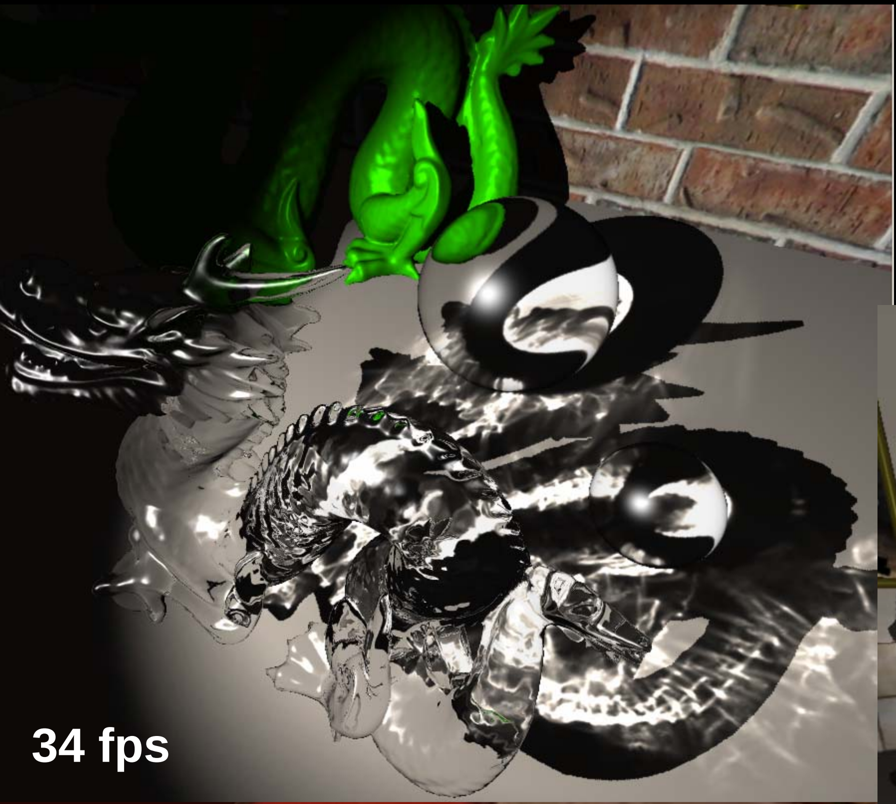
Summary

- Deferred shading for refraction
  - Quickly approximates refraction in deferred renderers
  - Allows multiple layers of refractive surfaces
- Adaptively sampling photons using deferred rendering
  - Deferred shading need not be spawned by full-screen quad
  - Allows adaptive sampling
    - Initial coarse sampling
    - Refine in important areas
    - Some error threshold stops refinement
  - Photons can be processed in batches
    - Overhead v.s. memory tradeoff
  - Avoids creating explicit "photon buffer"
    - HCM avoids processing wasted photons, we avoid creating them

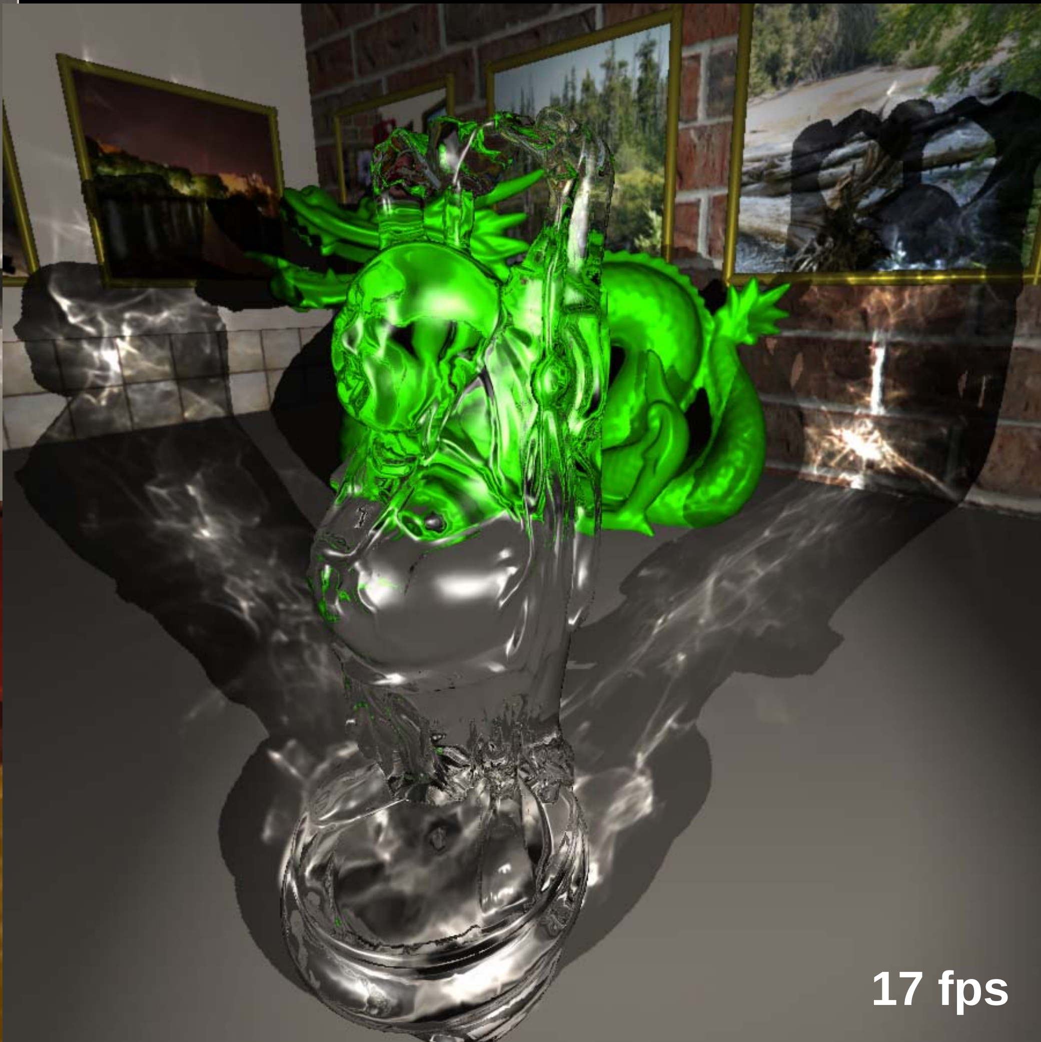
Questions?

Demo, with source code, available:

http://www.cs.uiowa.edu/~cwyman/demos.html


34 fps

21 fps


17 fps