PHOTON MAPPING AND BINARY SPACE PARTITIONING TREES

Acceleration Structure

- Binary space partitioning trees
- Restriction : Axis aligned cuts. Also called kd-trees
- Performance measurement using intersection tests per ray

Naive Intersection

- Bunny Model ~ 70,000 triangles
- Intersection cost - 70k intersections per ray
- Total cost - 70,000 * 512 * 512 (1 samples per pixel)
- Time : A few hours for a 512x512 image

Bunny Model

BSP Trees

- Split space using a plane. In axis aligned case, split bounding volume into two using an axis aligned plane
- Simple Method : Split in middle. works well for dense scenes.
- Choose split planes using a cost function. A famous one is Mcdonald Booth's surface area heursitic. Pick plane that maximizes the surface and minimizes the overall cost.

$$t_t + p_B \sum_{i=1}^{N_b} t_i(b_i) + p_A \sum_{i=1}^{N_a} t_i(a_i)$$

- probability to hit child is surface area child / surface area parent.
- More divisions where geometry density is higher.
- Computationally expensive. Takes longer to build

Tree Traversal

- Front to back traversal
- Using stacks instead of recursion. Much more efficient
- Intersect box. Use tmin, tmax and split position to find tsplit (position where it hit the split plane)
- Must cover all scenarios (ran into weird errors)

Optimization - Mailboxing

- Objects can be present in more than one nodes
- Multiple intersections undesirable
- Use tags to avoid intersecting again. Each object is tagged with an id when intersecting. Next time, tag is checked first
- Saves a LOT of time

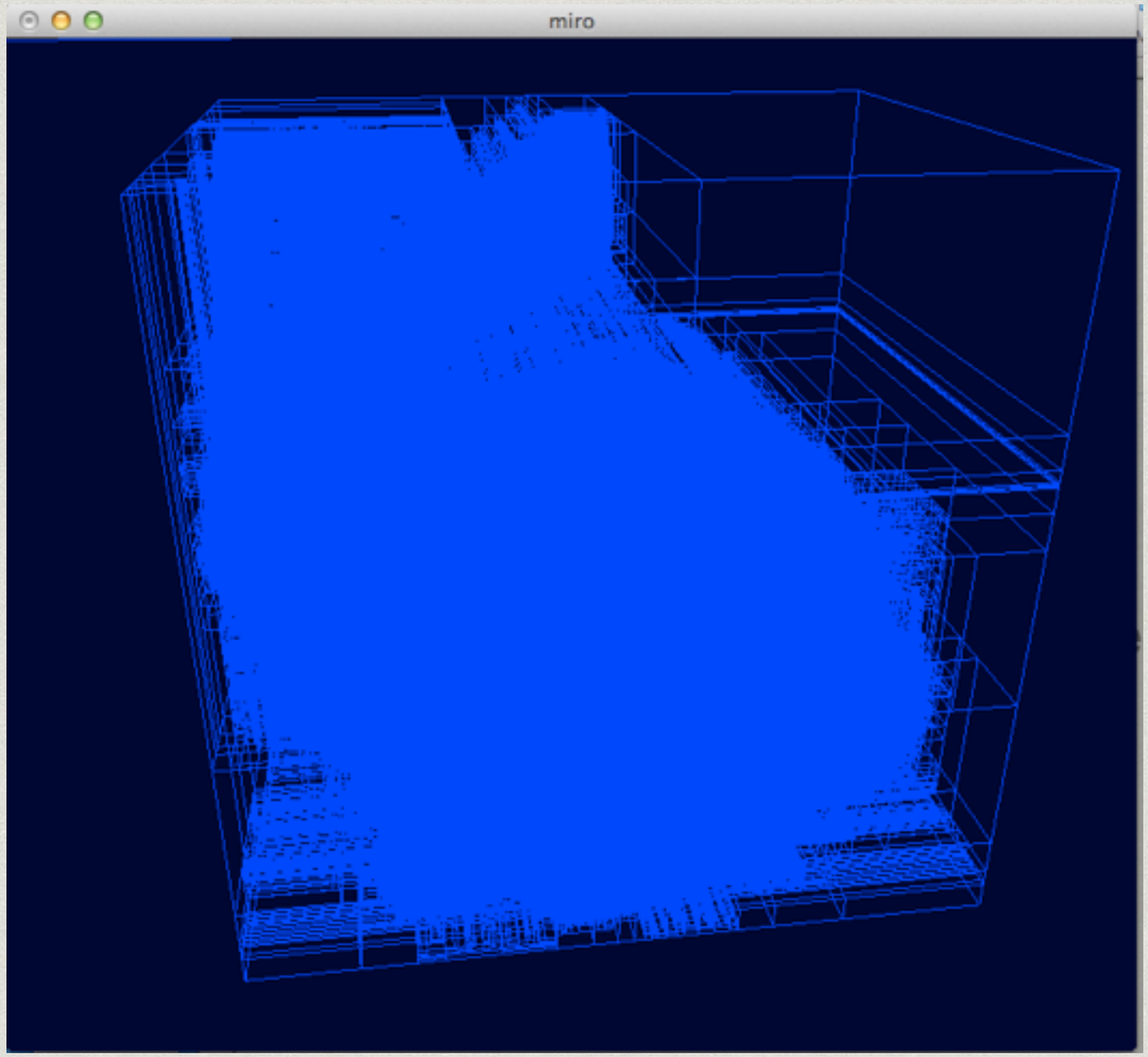
Result

- Construction Time : 5 - 30 secs
- 10,000x speedups
- Bunny Scene - 7 intersections per ray !!!!
- Render time - 8 secs at 9 samples/pixel :)
- Dense model so speedup is huge. Lets try a sparse model

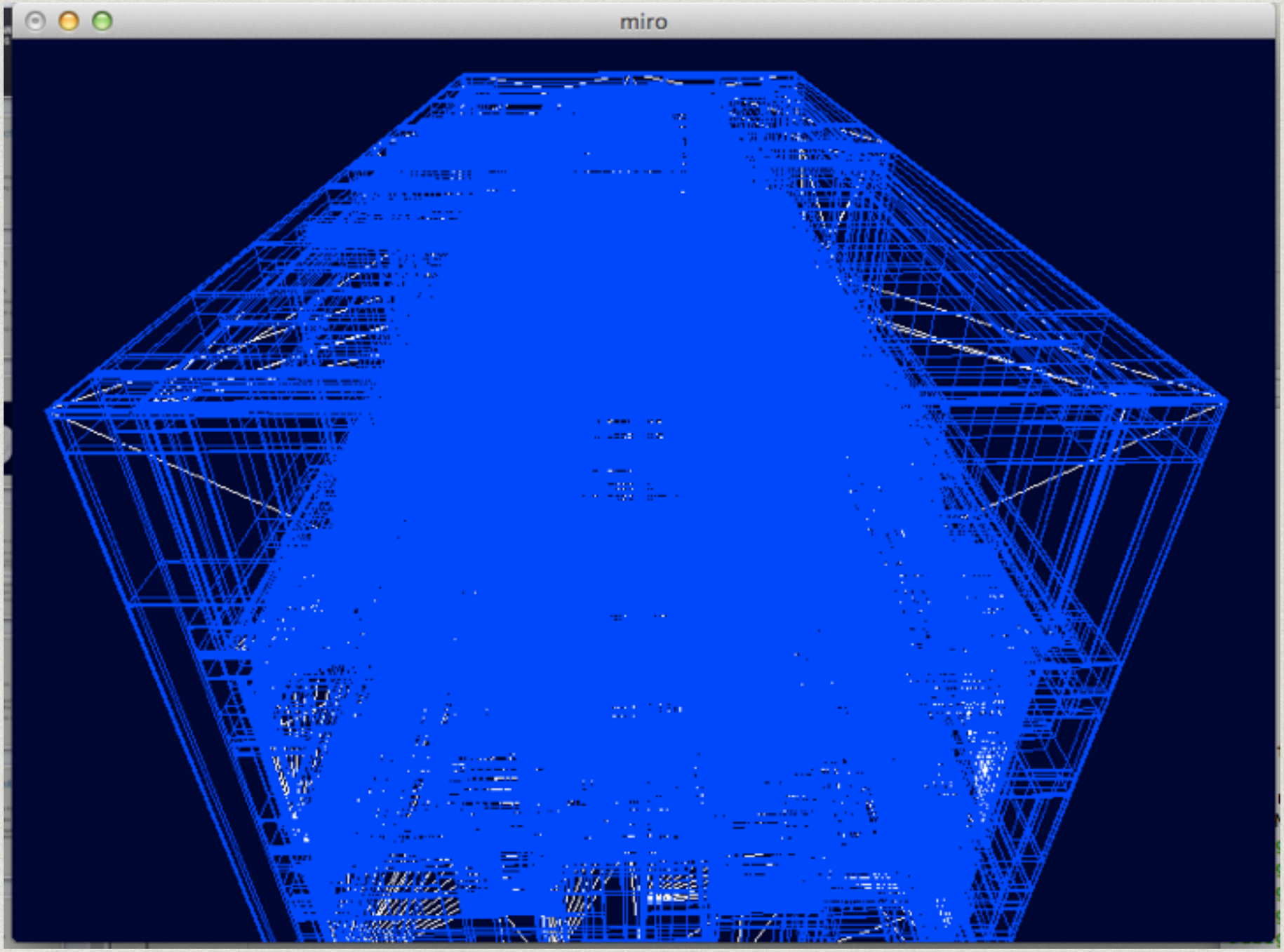
Sponza Atrium

- 66500 triangles
- Render Time - 50 secs at 9 samples/pixel
- Performance - 11 intersections/ray
- Can be further reduced by checking at more split positions. slightly increases build time

Tree for the bunny



Tree for Sponza Atrium



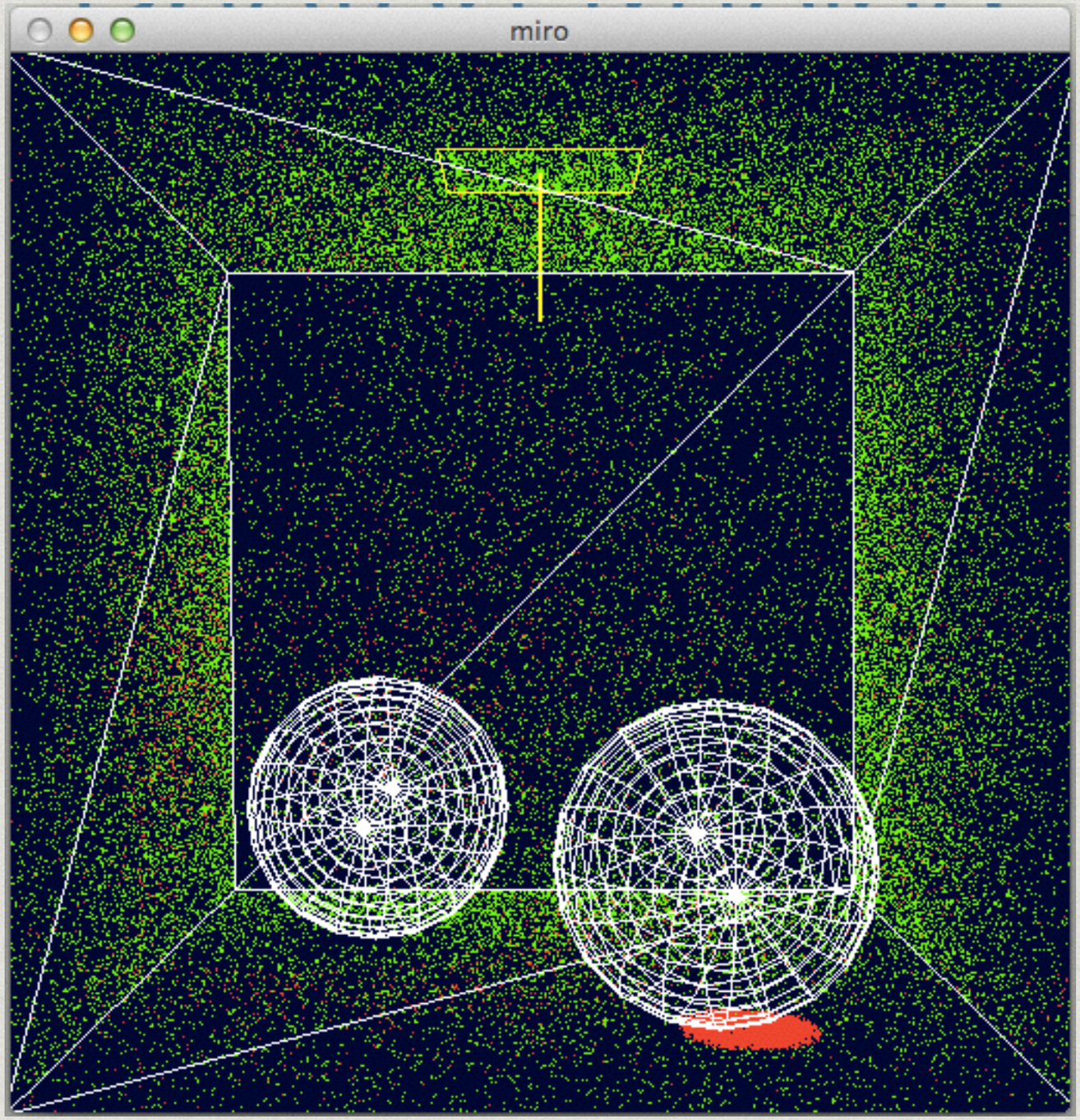
Photon Mapping

- Global Illumination needed for realism
- Biased but consistent method
- Photons carry flux. Map provides irradiance estimate
- Physically based rendering (book) used. All parameters are based on radiometric values
- Lambert BRDF used. Phong is not physically accurate. Still implemented in the code. Can be activated.

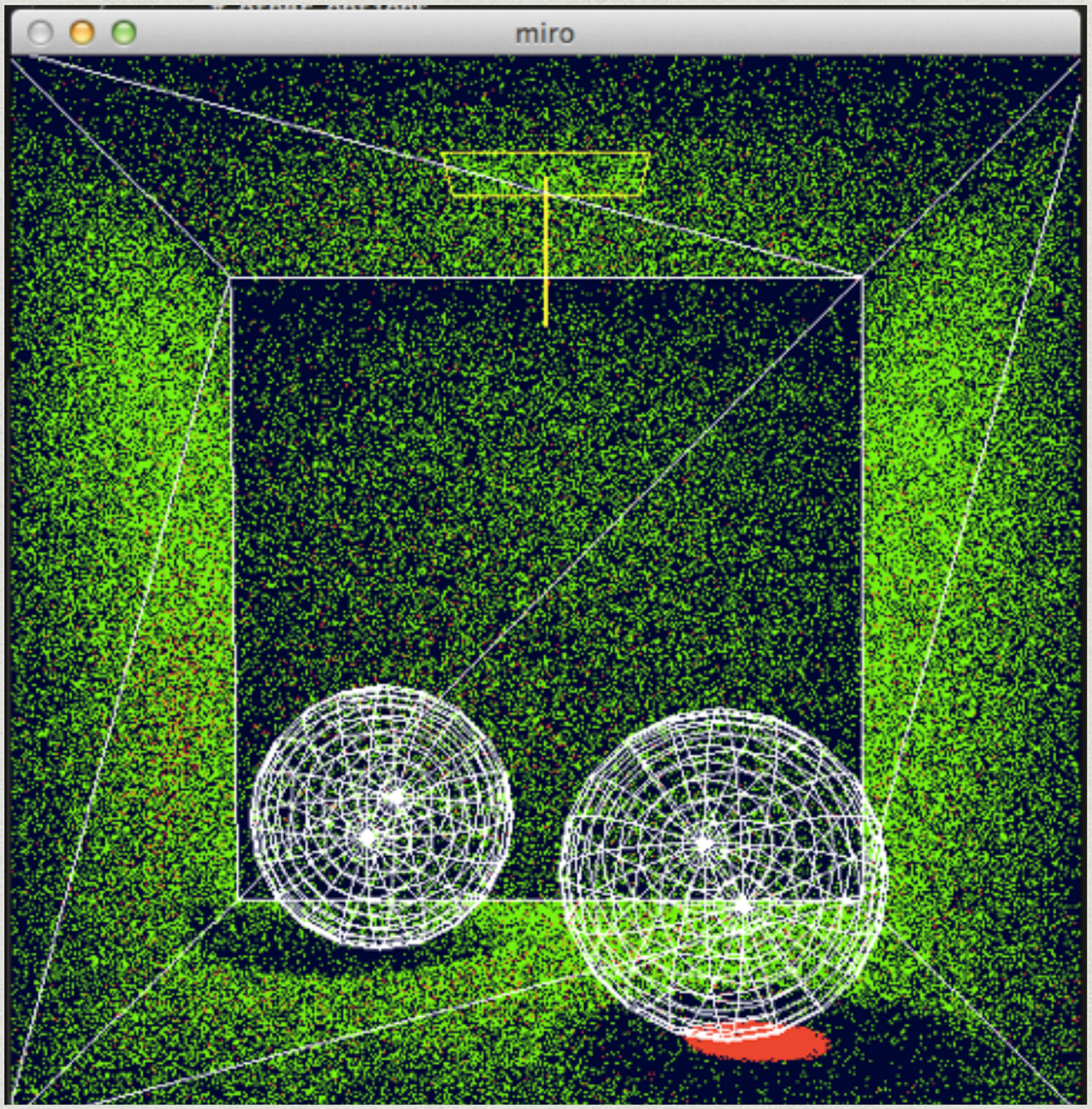
Tracing Photons

- Shoot photons according to power of light source
- Power = L*pi*area (for area light source)
- Power = L*4*pi (diffuse point light)
- Caustic photons stored at first hit at diffuse surface after coming from specular surface
- Indirect OR global photons stored in separate map. Depending on whether final gathering is enabled

Photon tracing (Indirect)



Photon Tracing (global photons for final gather)



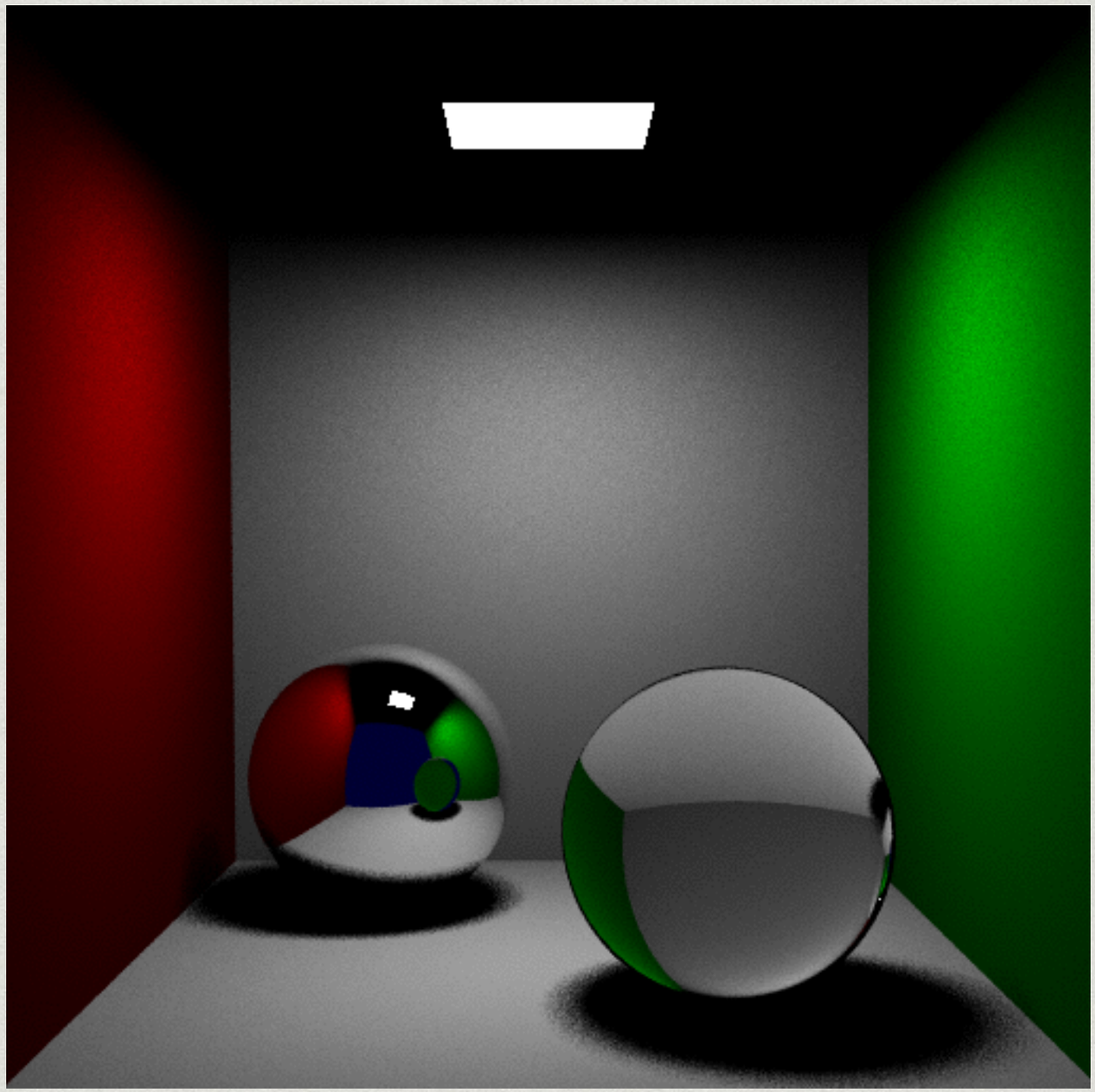
Rendering using photon map

- Direct visualization of caustic map
- Direct visualization of indirect map (option)
- Final Gathering (option)
- Scene description has options for total photons, caustic photons, photons in estimate, caustics photons in estimate and final gather

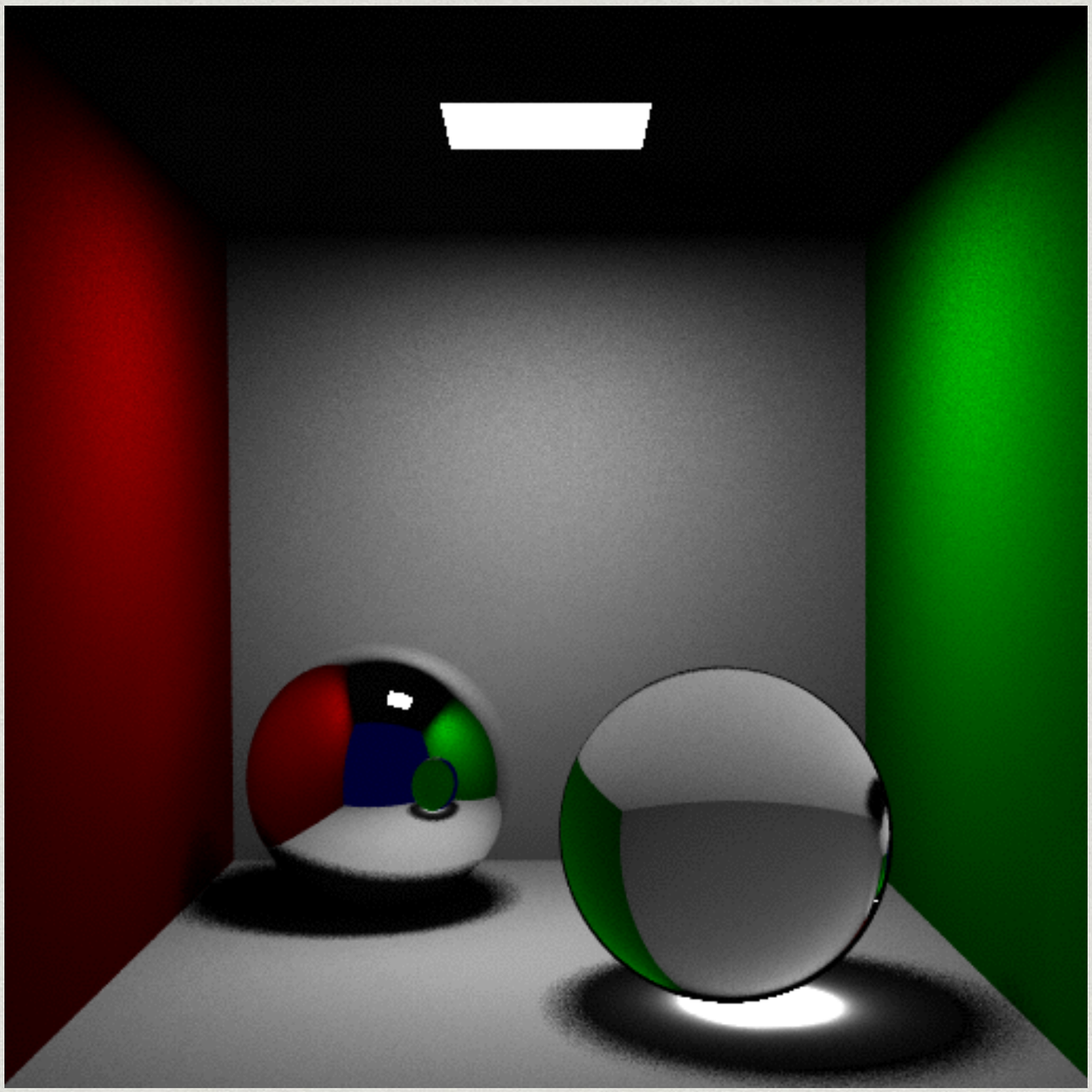
Final Gather

- For Indirect Illumination : L = brdf*irradiance
- For final gather ray : L = brdf*pi*L_reflected (from photon map)
- Derivation given in book and cmu lectures. required to be physically accurate

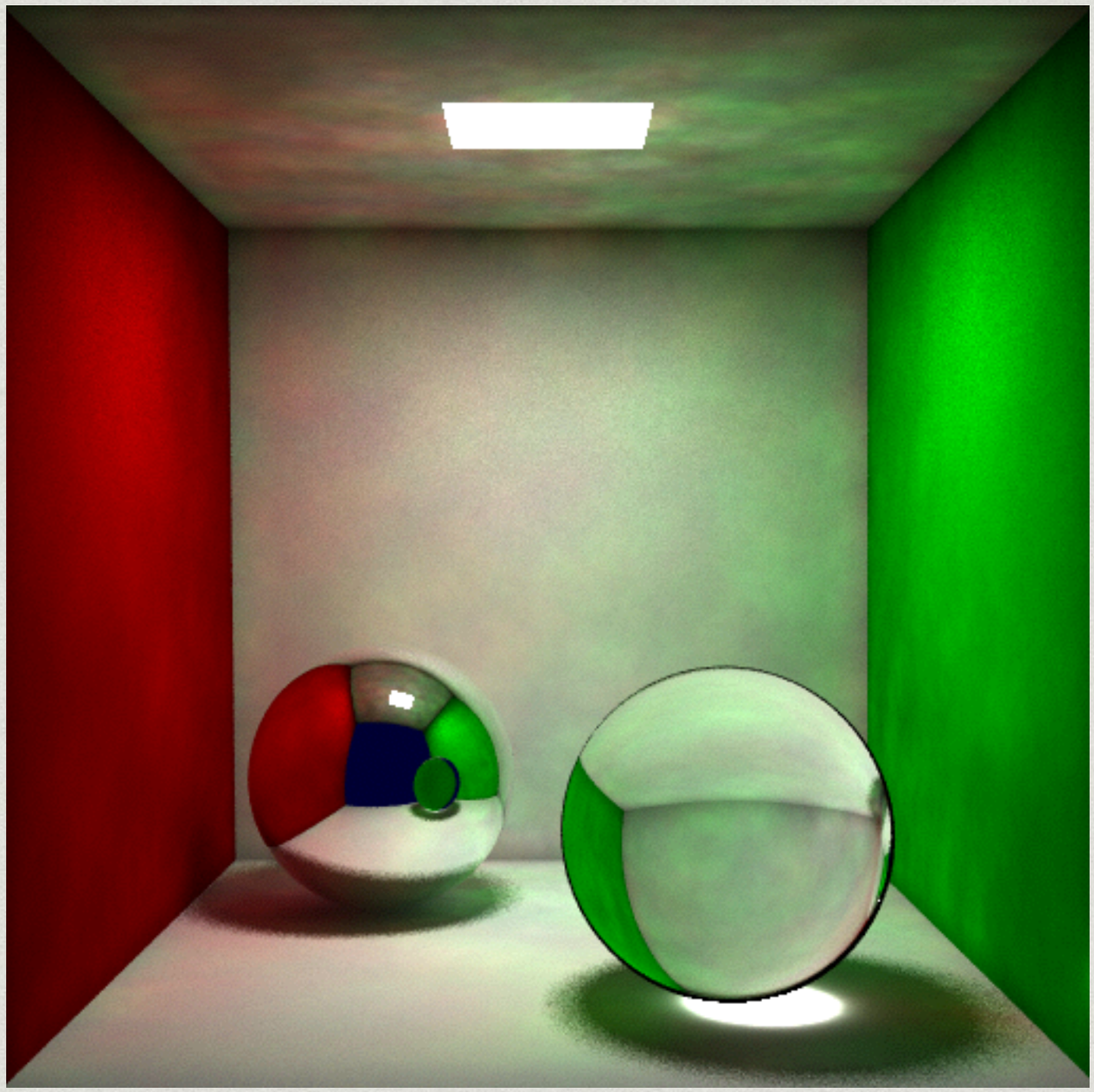
Direct Illumination (Render time - 10 secs



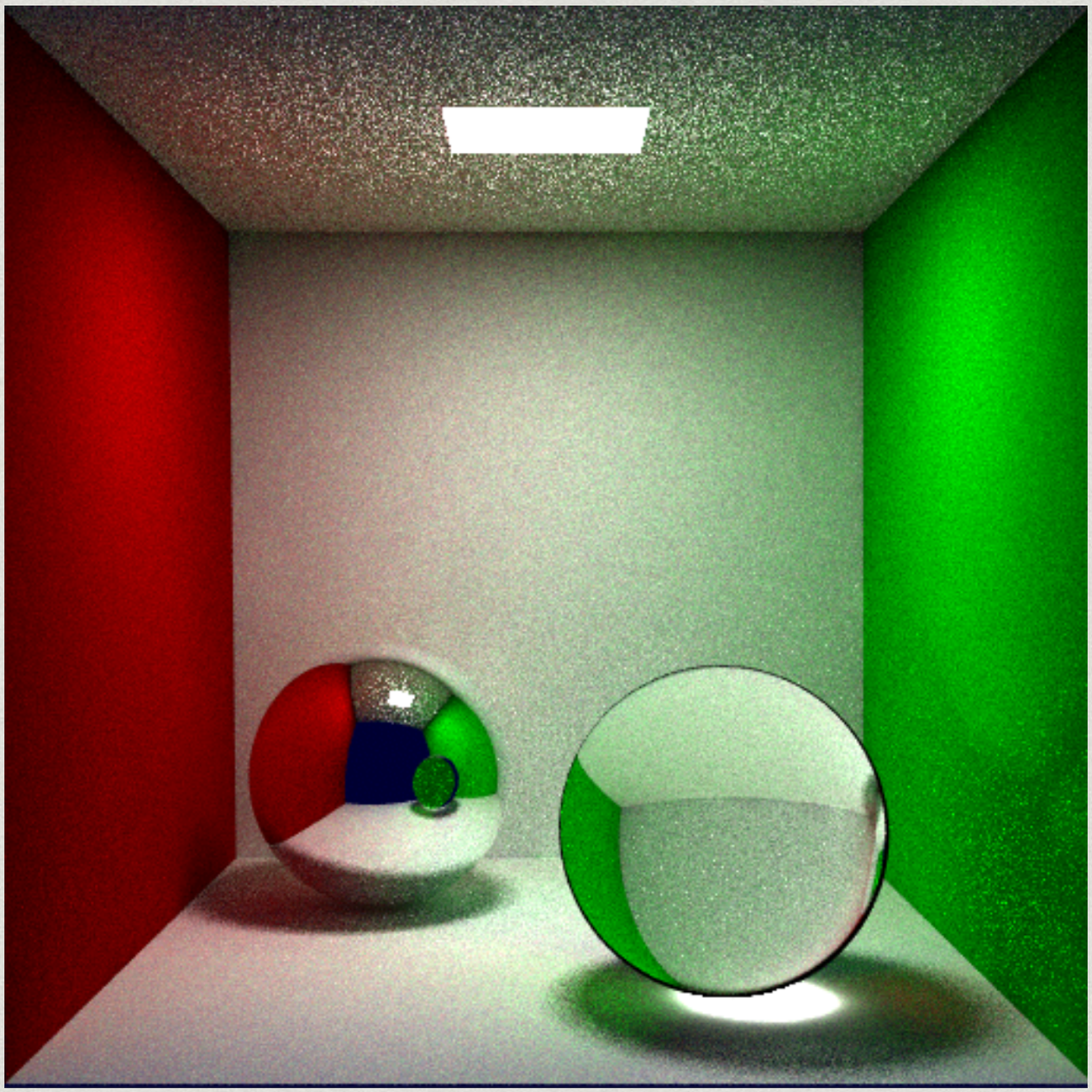
Caustics (render time - 20 secs) (100k photons)



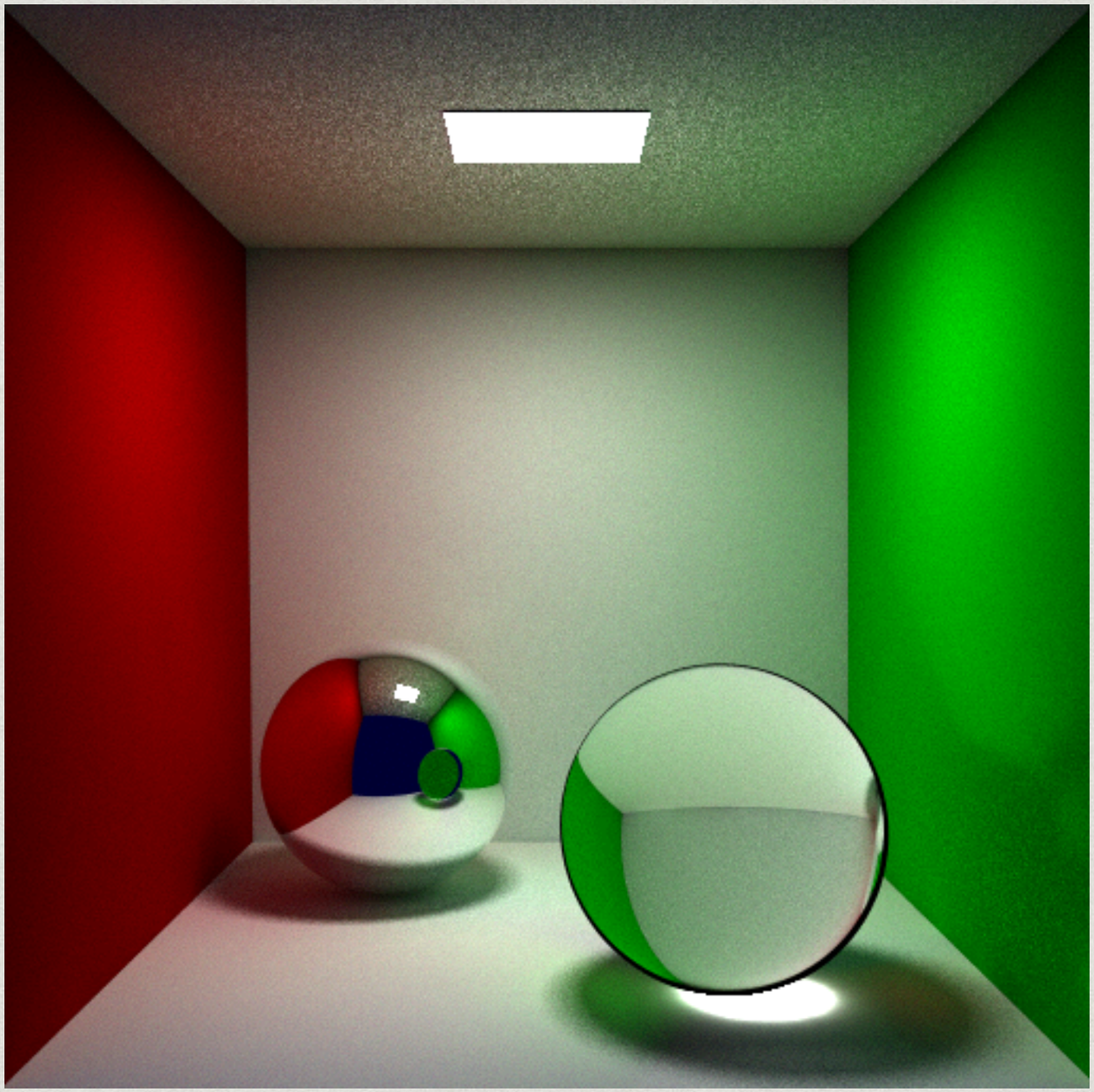
Indirect Illumination (direct visualization) 250k photons (Render time - 1 min 15 secs)



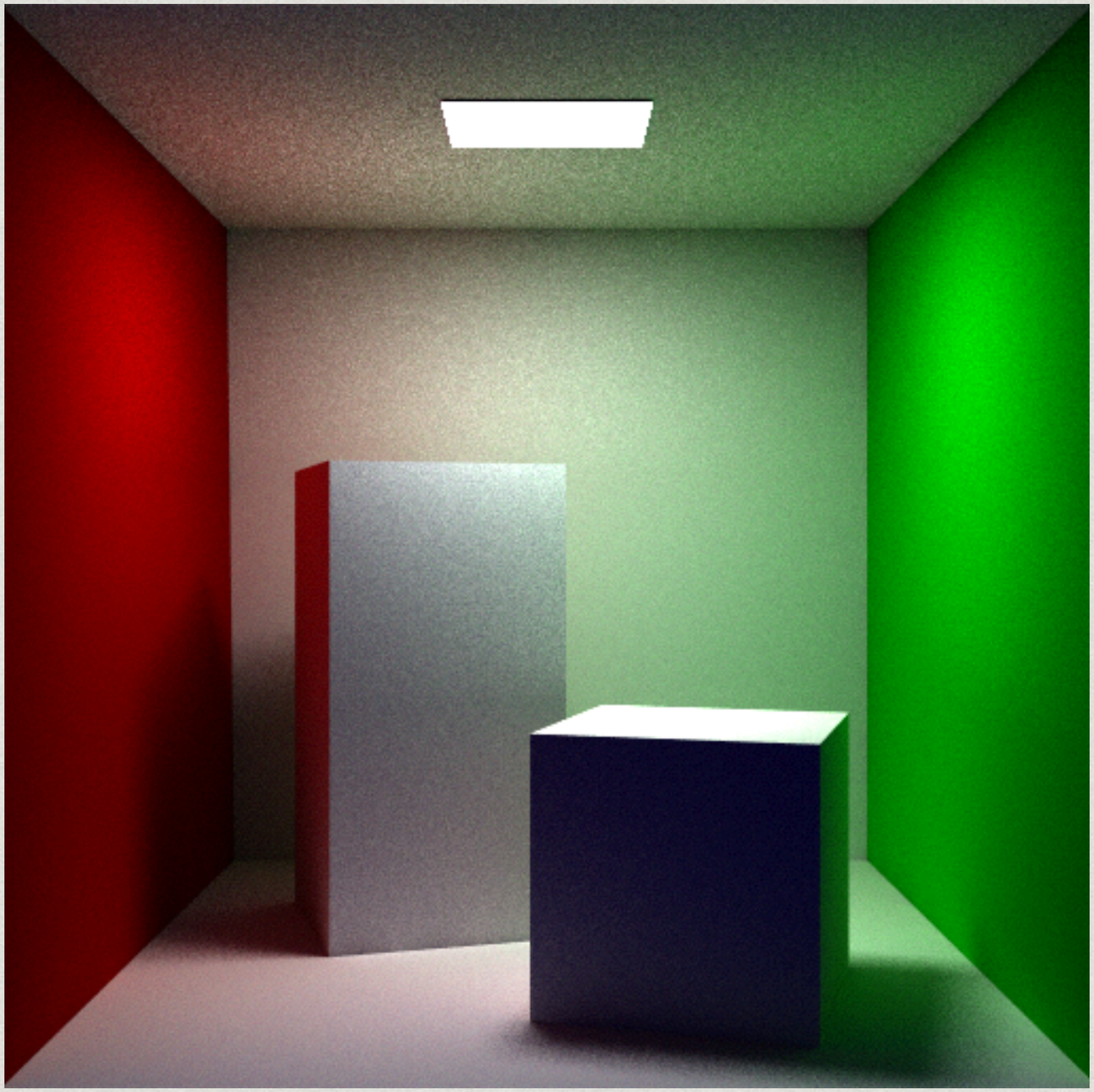
Final Gathering (9 samples/pixel, 3 gather rays, 100k photons) (render time 15 mins)



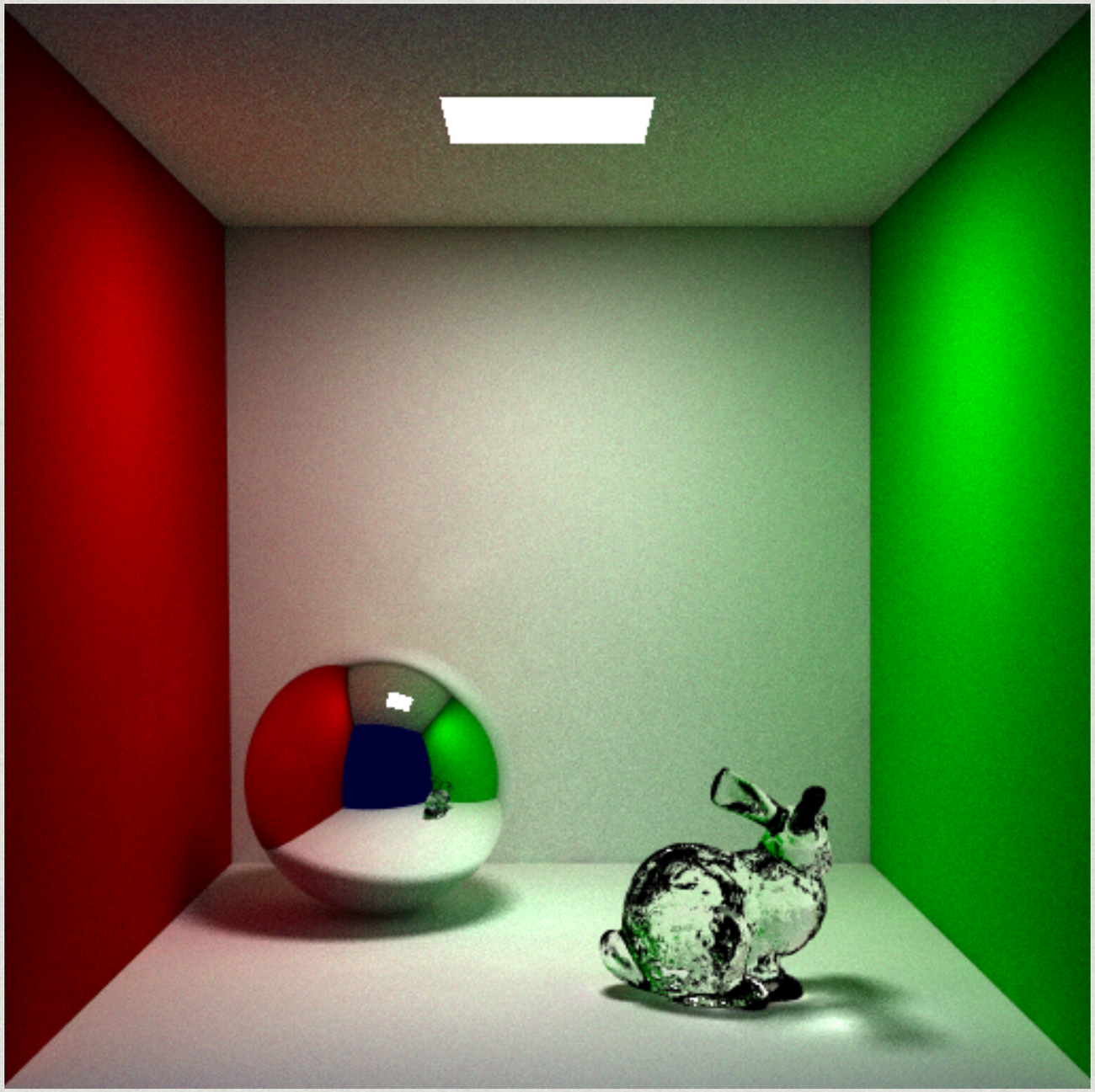
Final Gather (9 samples/pixel, 8 gather rays, 250k photons) (render time - 30 mins)



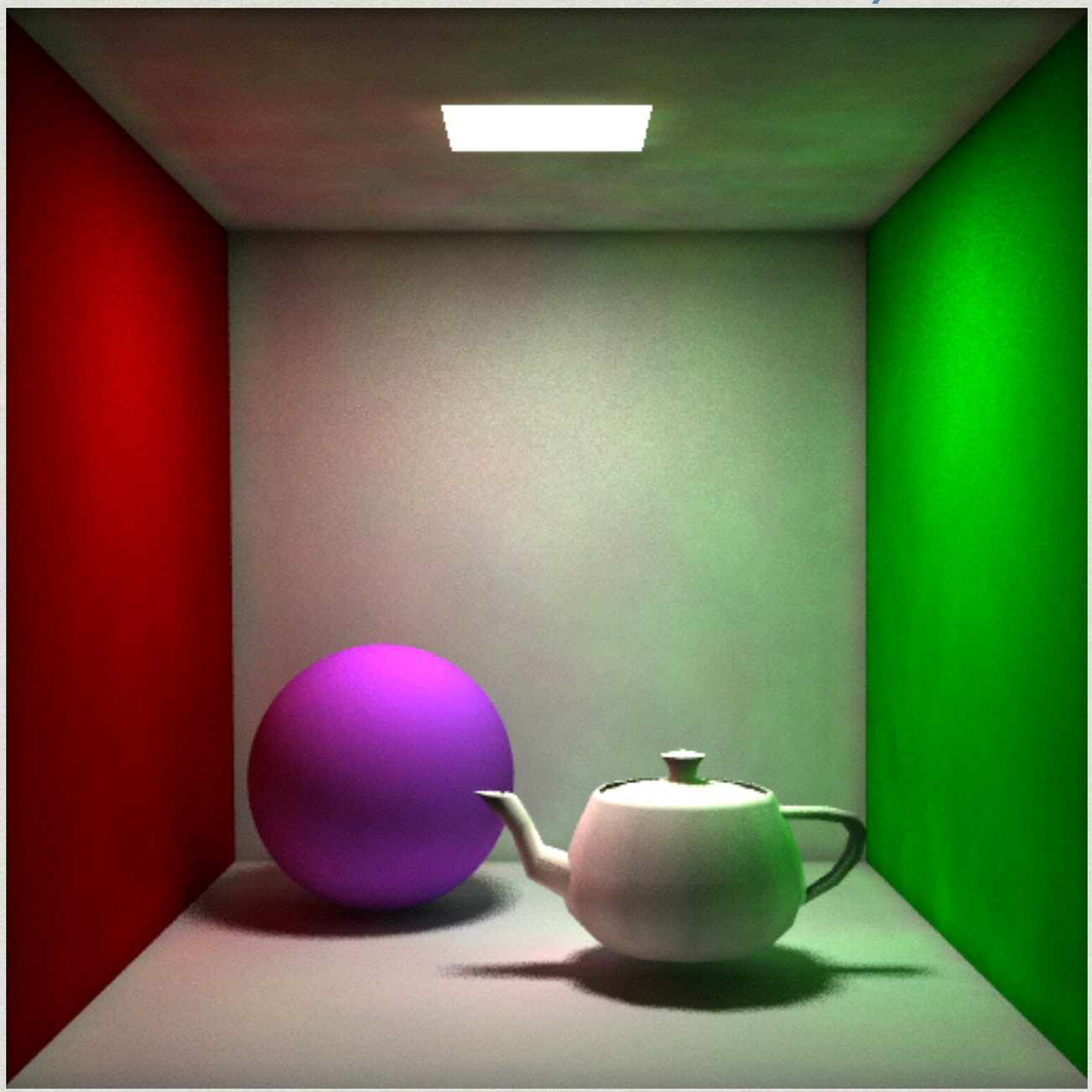
Cornell Box



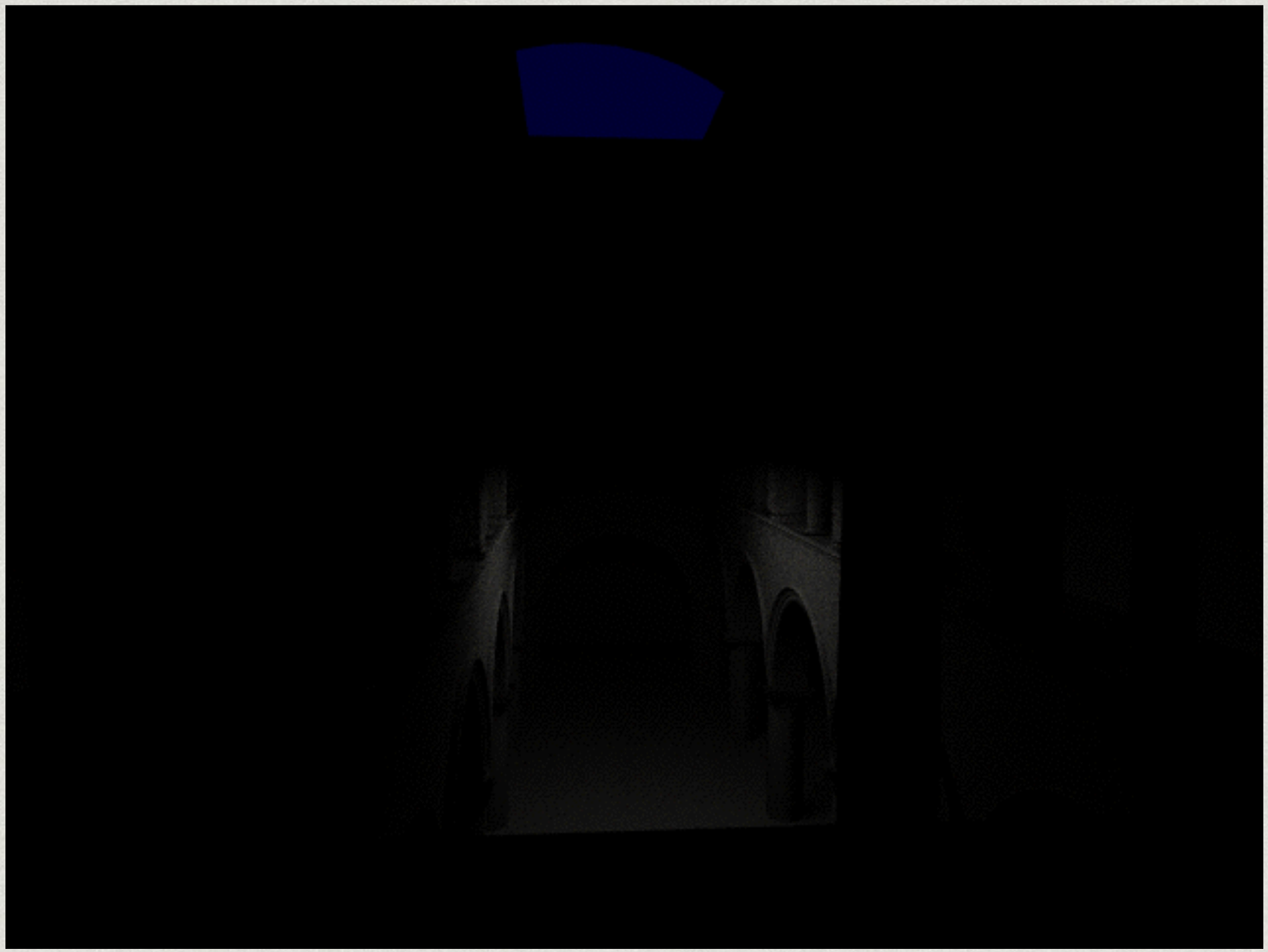
Cornell box with bunny



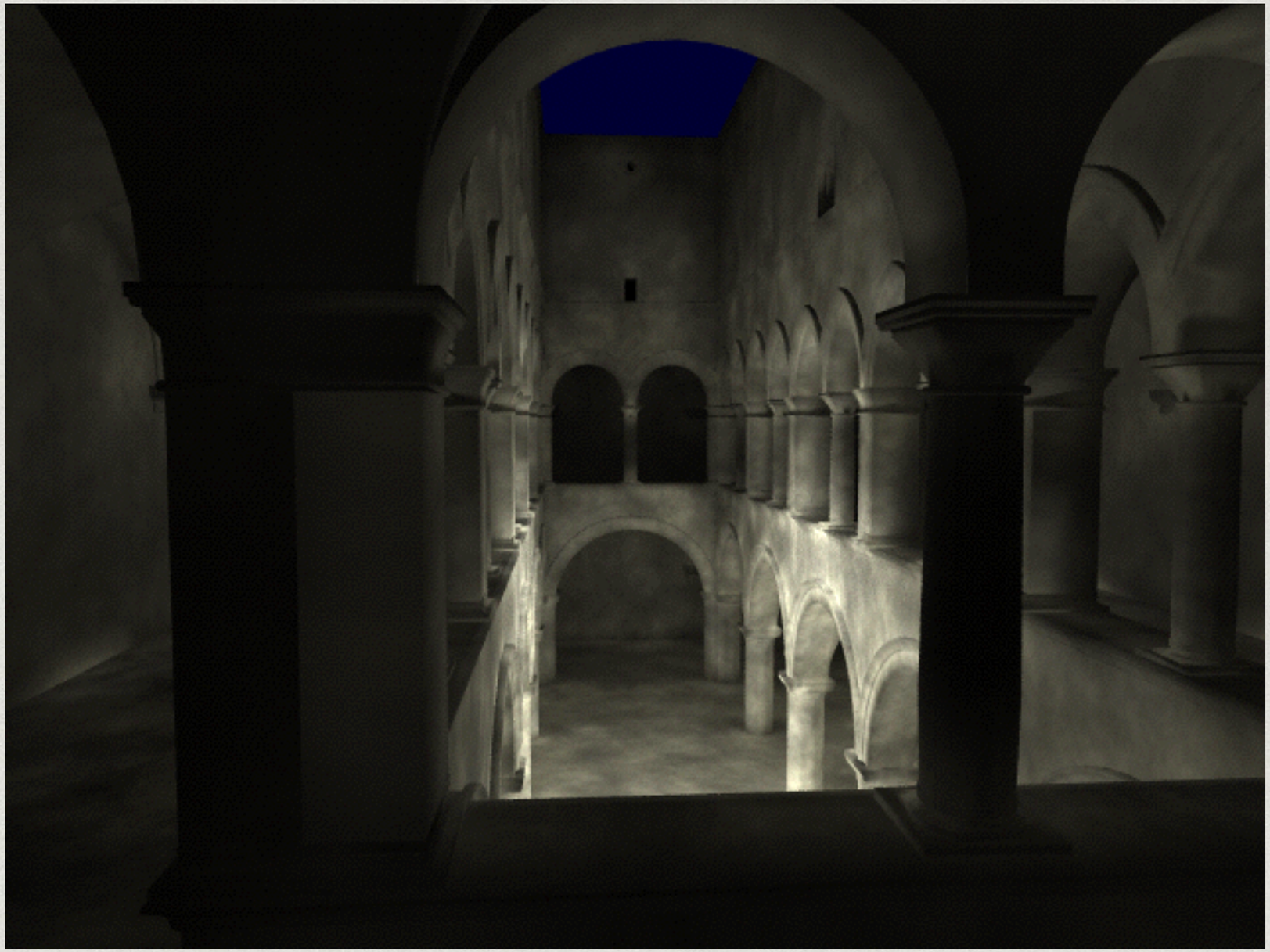
Cornell box with teapot (direct visualization, 750k photons, 750 in estimate, 3 mins 49 secs)



Sponza Atrium (direct illumination)

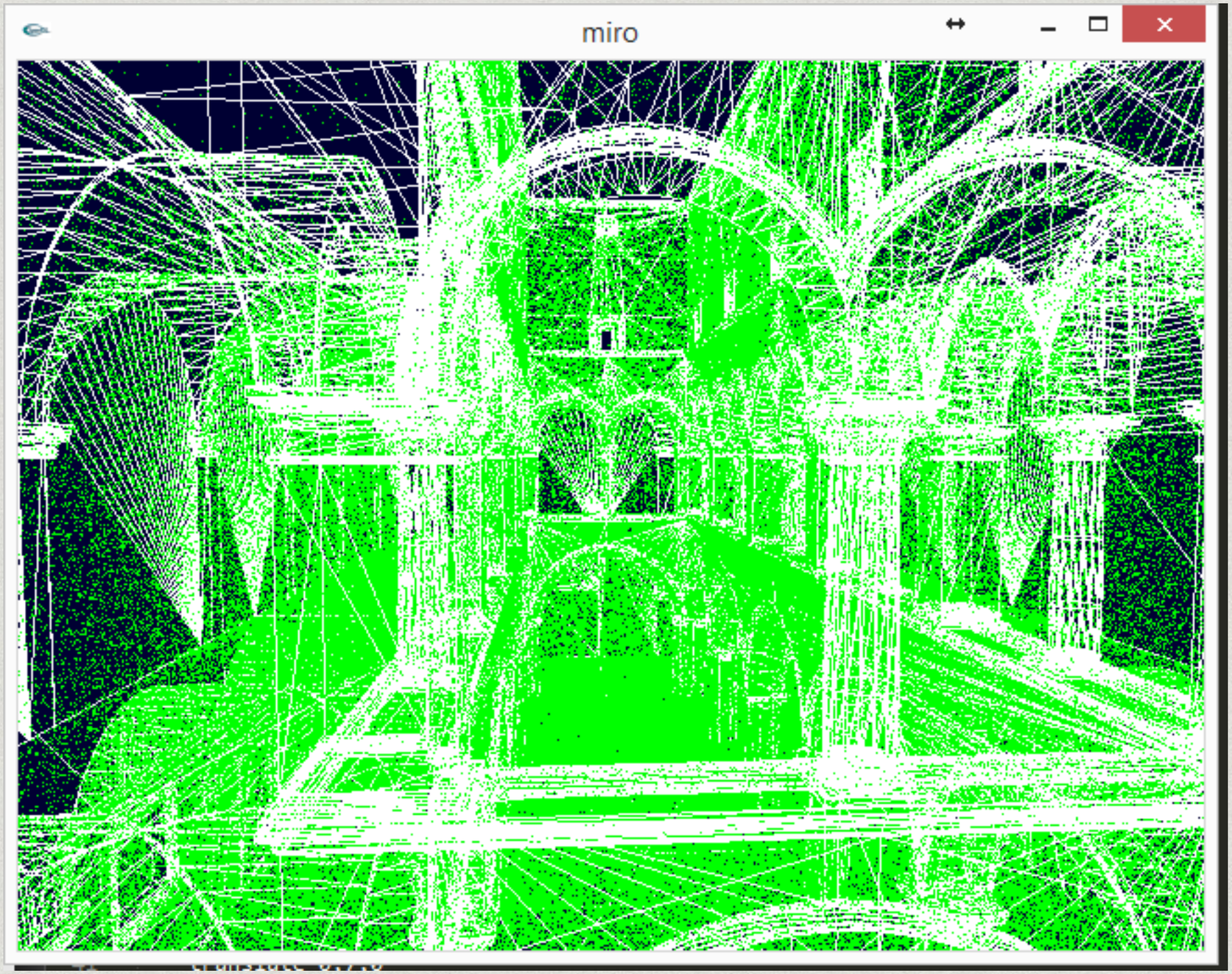


Sponza scene (100k photons)

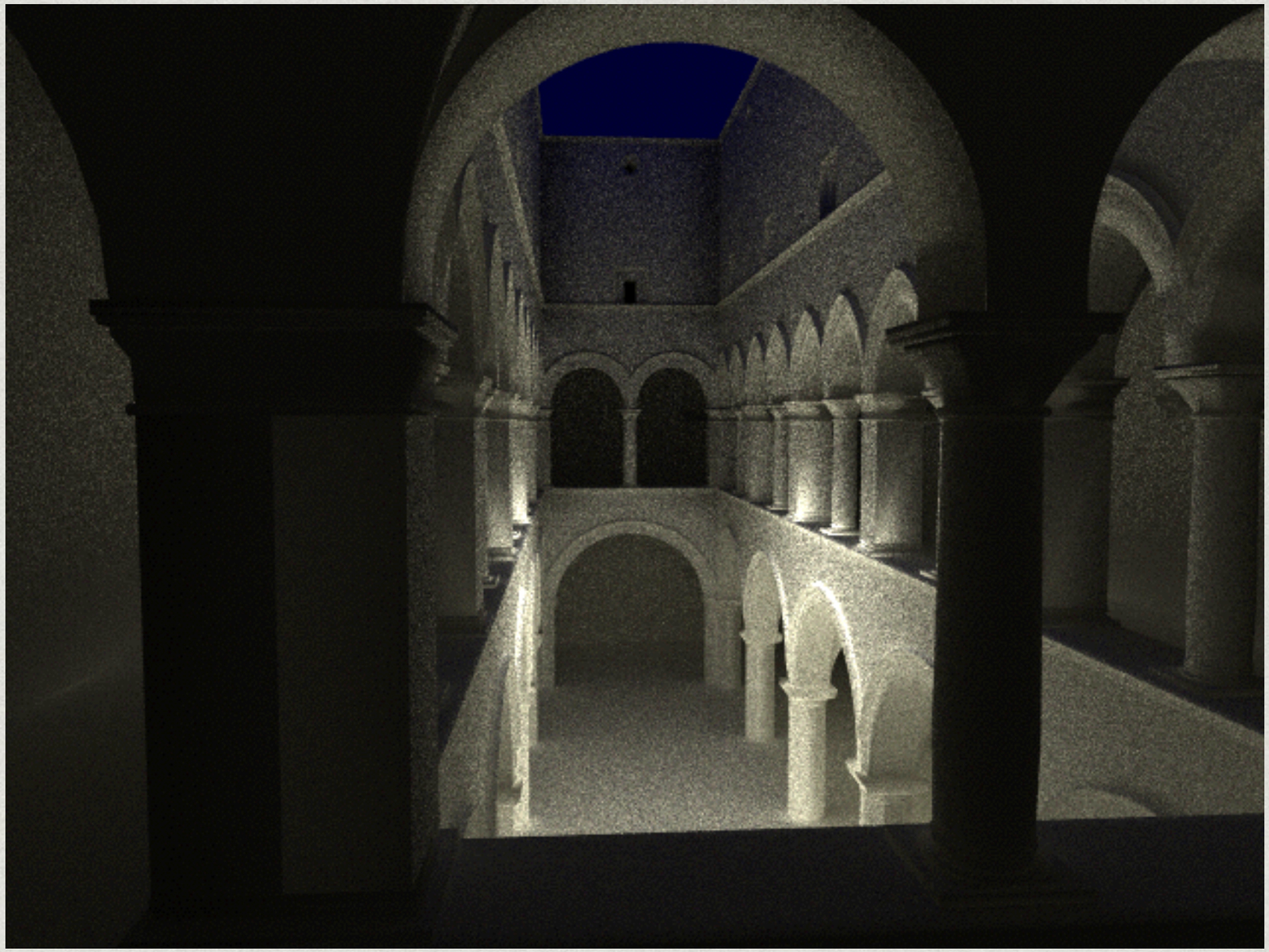


Sponza (500k photons)

Sponza (photons global)

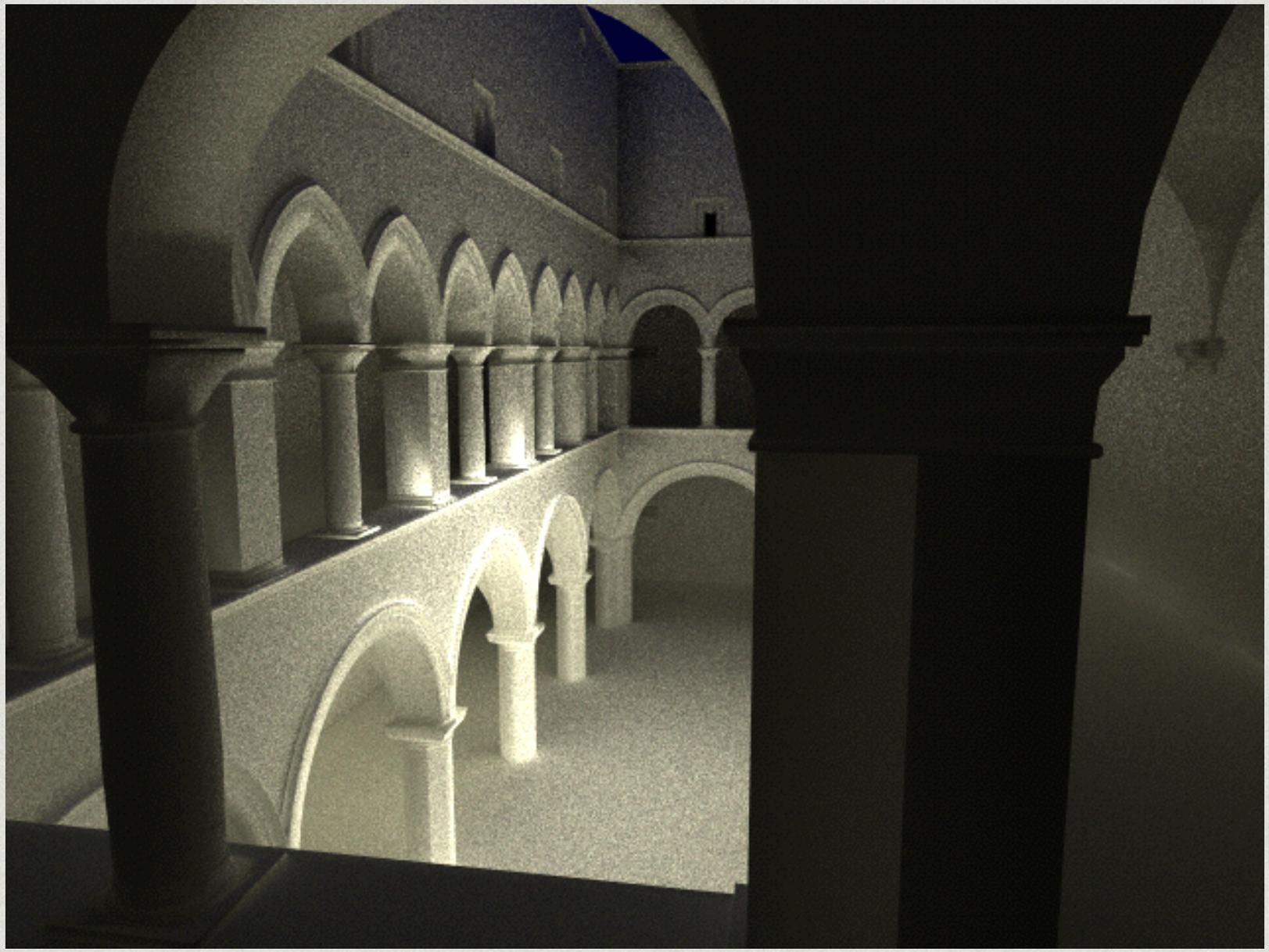


Sponza (final gathering - 3 final gather rays, 9 samples per pixel, 10 mins 24 secs)



Sponza (7 gather rays, 25 rays/pixel, 500k photons, render time - 43 mins)

Sponza (from another view with slightly brighter light)



Sponza Atrium (with two yellow light. render time - 55 mins)

Features to Add

- Textures for realism
- Updated Obj parser with mtl support
- Environment Illumination
- Irradiance Caching (to speed up final gathering stage)ß