ECS 275A Advanced Computer Graphics

Photon Mapping with Caustics

Name: Shu-Wei Hsu

Student ID: 998569417

Introduction

We apply Photon Mapping technique to render global illumination. We added glossy objects, such as glass ball, metal ball, and diamond to the scene. These objects are created by combining objects instantiated from existing polygon object we already have in the code.

The direct illumination is done by tracing rays from the camera to each pixel of the image. It is mostly based on conventional Phong Shading without ambient colors. When the ray hits a glass objects, it is reflected and refracted by Snell's Law and Fresnel Equation.

The Photon Mapping for global illumination is separated to two phases. In Pass 1, we generate two photon maps: global photon map and caustic photon map, by tracing photons from the lights. We find the nearby photons by radius search of kd-tree to compute radiance at a surface. In Pass 2, we trace rays from the camera for a final gathering of photons.

The indirect illumination is also done by tracing rays from the camera. Nevertheless, it uses Monte-Carlo's method to integrate over the hemisphere on where it hits. On the second pass, photons are sampled from the global photon map created in pass 1.

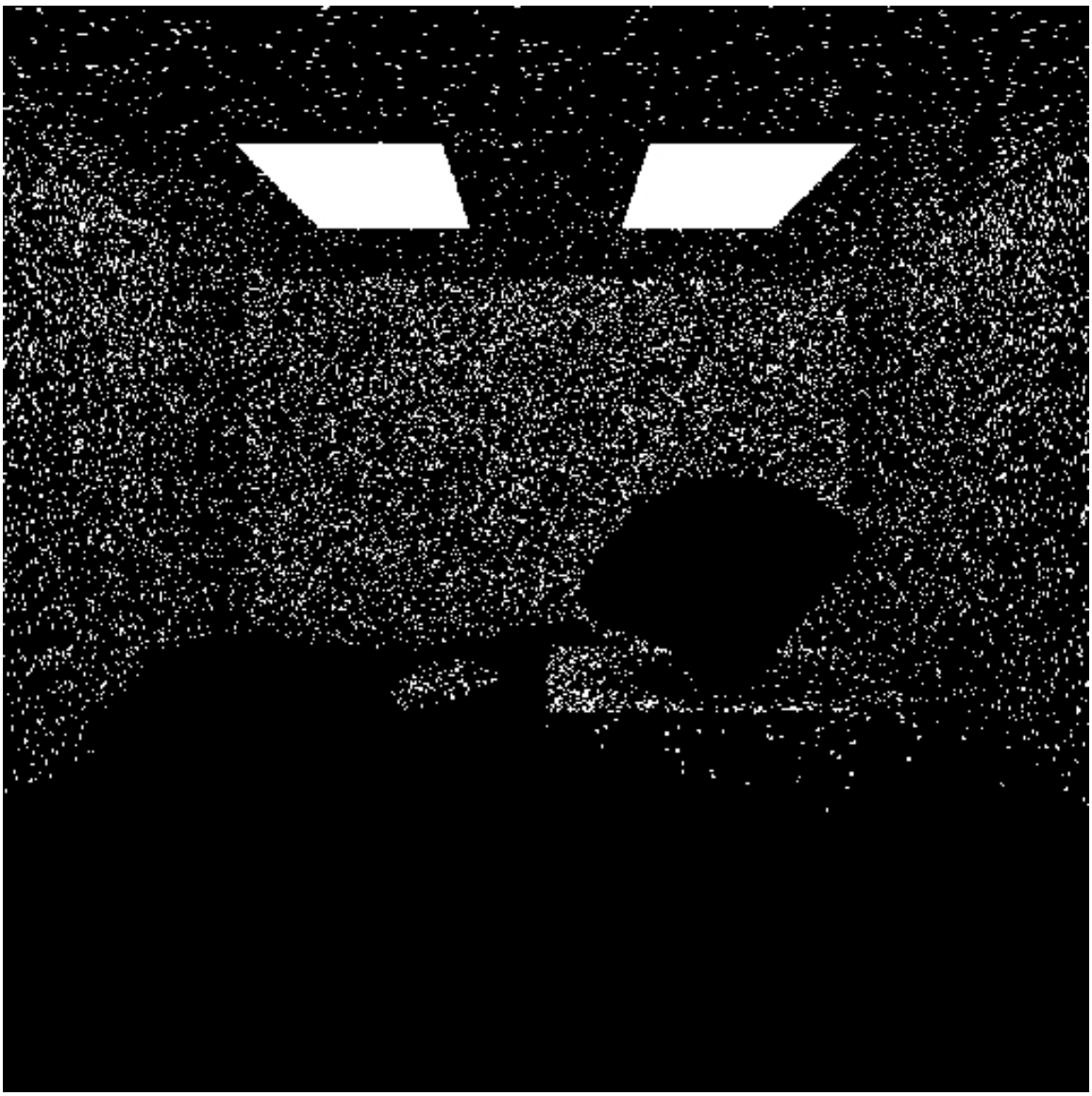


Figure 1. Indirect Illumination with 50,000 Photons Sampling (54.7 seconds)

The caustics are generated by sampling photons from the caustic photon map on the bounces of the rays traced from camera. In the end, we will accumulate direct illumination, indirect illumination, and caustics to create a more spectacular and real picture than conventional ray tracing and path tracing.

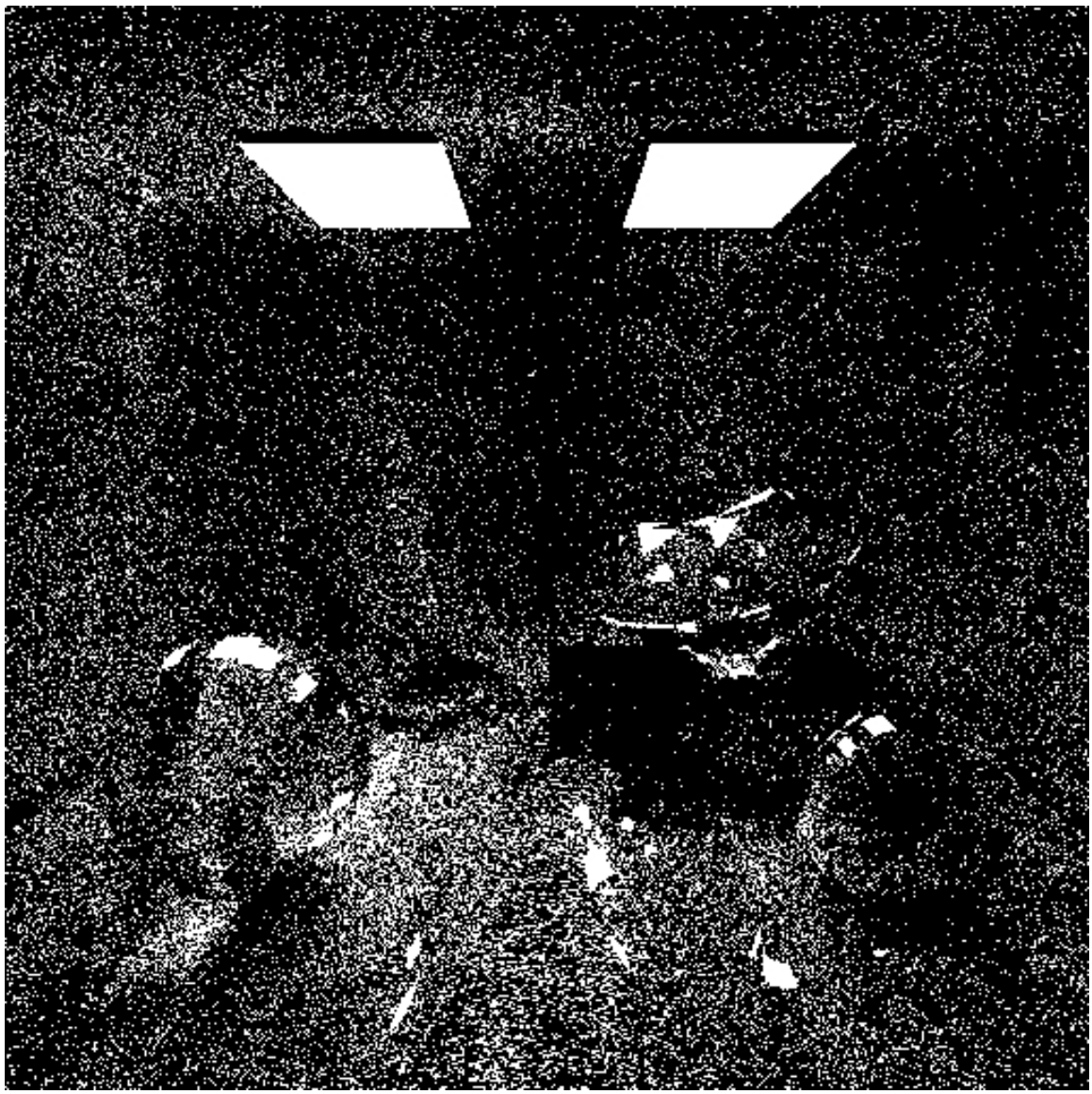


Figure 2. Caustics with 100,000 Photons Sampling (198 seconds)

Implementation

Direct Illumination

The direct illumination is done by tracing rays from the camera to each pixel of the image. based on conventional Phong Shading without ambient colors. Since some objects with lots of facets (e.g. Diamond) and more sophisticate intersection and normal computations take more time to render, we start implementing photon mapping and caustics with simple objects like spheres, cubes, tetrahedrons.

However, our goal is to build glossy objects, such as glass ball and metal ball. The metal ball can be simply implemented by high refection factor. However, glassy objects need both reflection and refraction rays when a ray hit their surfaces. We implement glassy objects based on Snell's Law with critical angle constraint conditional statements.

We use jitter sampling with 36 sampling times for direct illuminations. However, we start with only 1 sampling time for debugging efficiency. The other benefit of starting simple object first is simplicity of arranging the scheme. We can find the most obvious effects on specific objects. Slight refinement is required after gathering all the objects. The rendering computation intensity can be told by the ratio of rendering times.

The Diamond object is created by importing .obj file from Computer Graphics course website of MIT Computer Science And Artificial Intelligence Laboratory – Computer Graphics Group [1] using our own polygon class using Pluker coordinates [2] to find the intersection points. The normal is defined by counter-clockwise declared points of polygons. Two area lights are added for better shadow penumbra

effect. We create a watersink class implementing SINC function on the y-axis with Newton's method to find the intersection point. The initial value is defined by the hit point with considering the water as a flat surface.

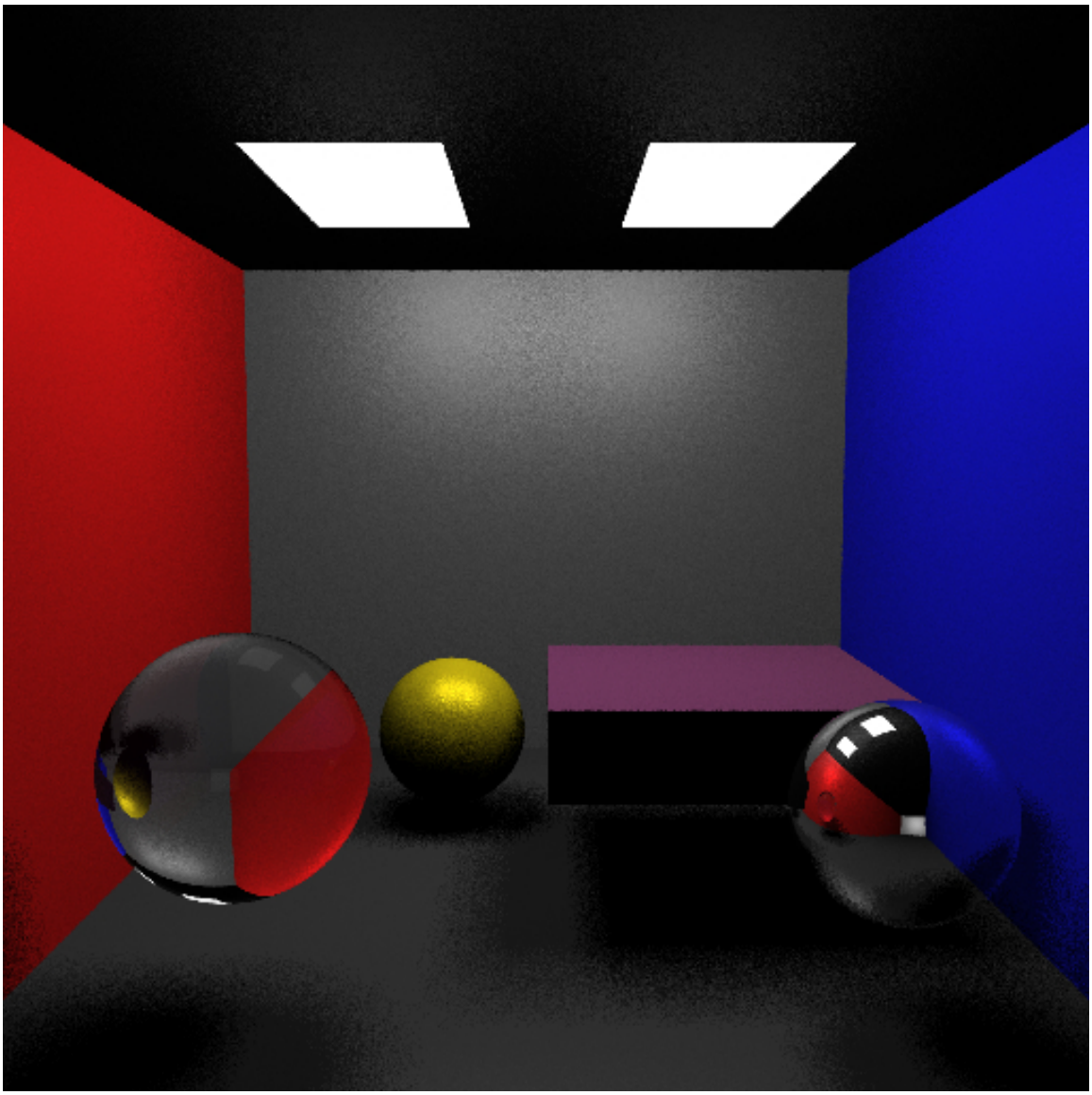


Figure 3. Direct Illumination with Simple Objects (8.39 seconds)

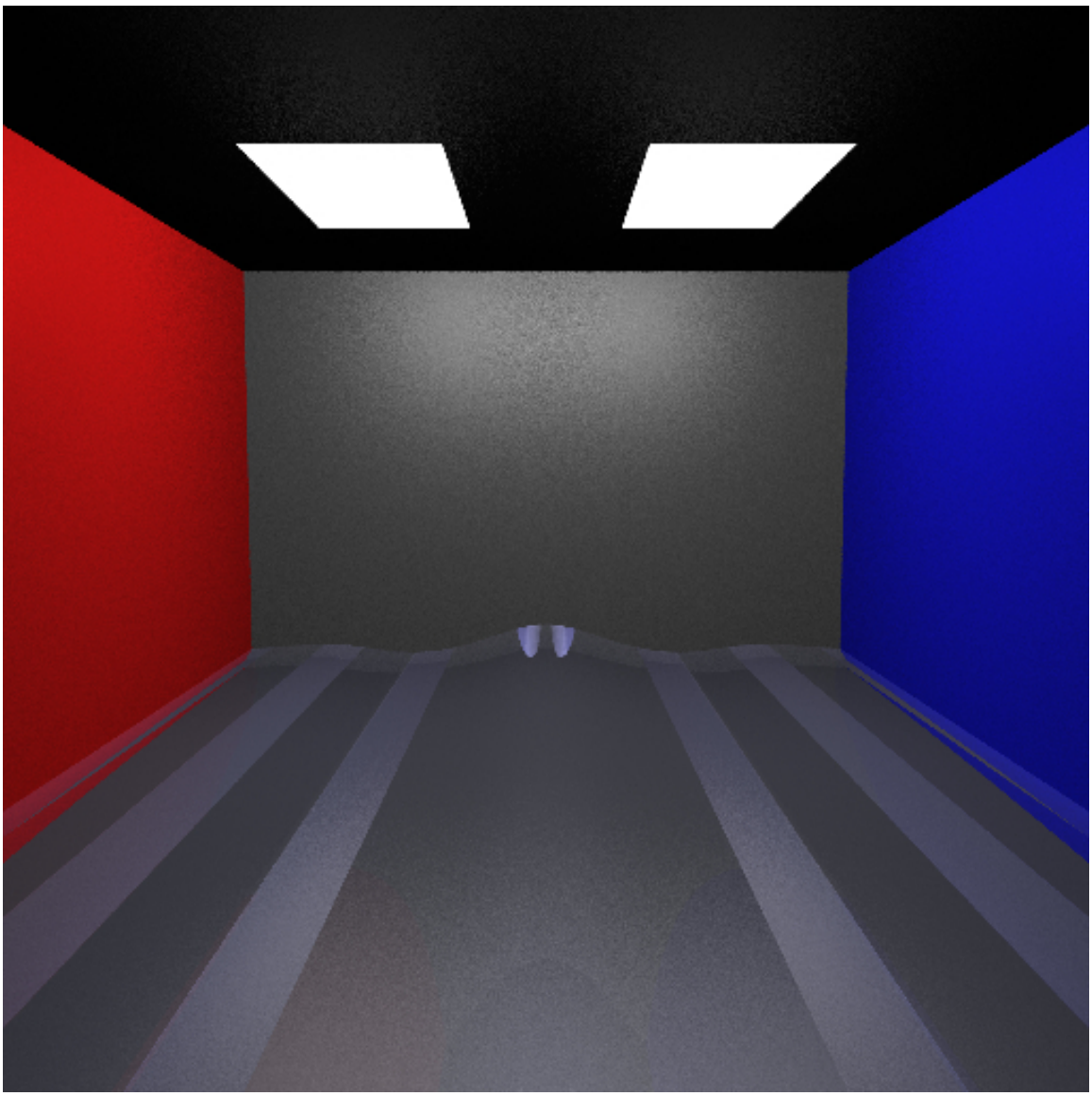


Figure 4. Direct Illumination with Water (9.17 seconds)

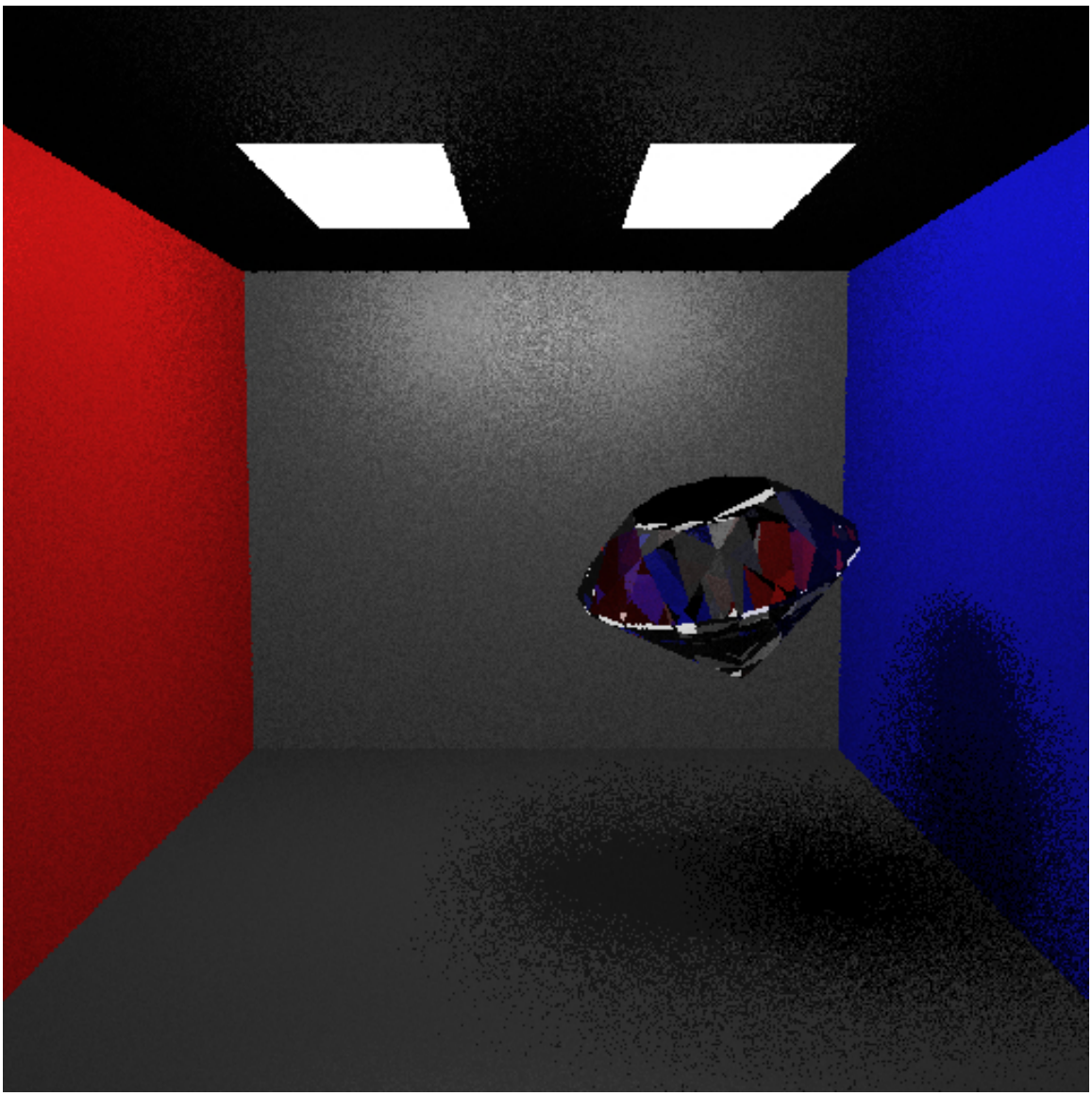


Figure 5. Direct Illumination with Diamond (110 seconds)

Indirection Illumination

For pass 1, we generate two photon maps: global photon map and caustic photon map, by tracing photons from the lights. We find the nearby photons by radius search of kd-tree to compute radiance at a surface. In Pass 2, we trace rays from the camera for a final gathering of photons.

For 50,000 photons are emitted from each light source (total 100,000 photons) with radius search of kd-tree created by John Tsiombikas [4] with radius 3.0. Jitter sampling is also applied to both photon mapping and later caustics effect. Color bleeding on surfaces reflected by nearby objects is shown in indirect illumination images.

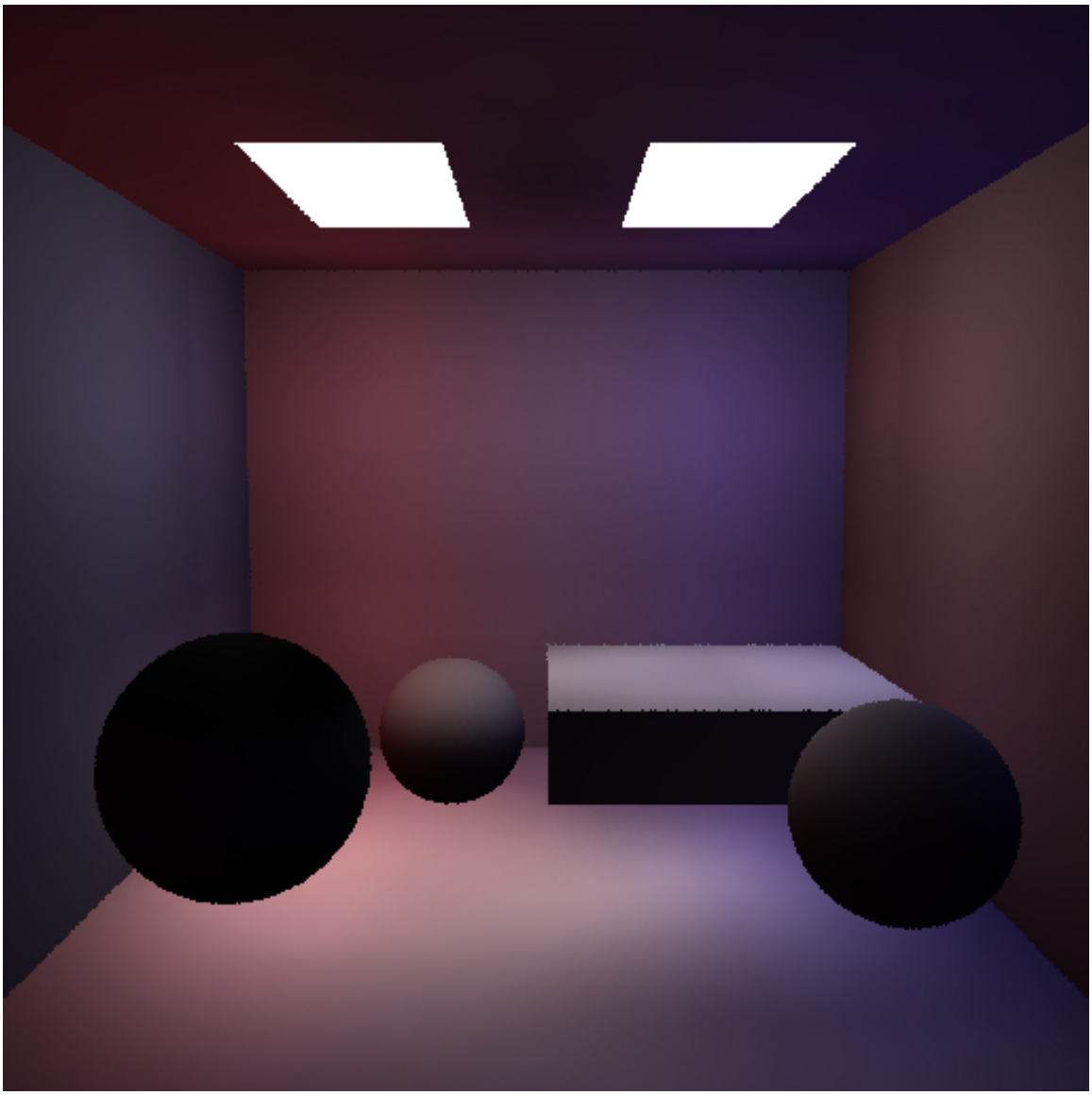


Figure 6. Indirect Illumination with Simple Objects (445 seconds)

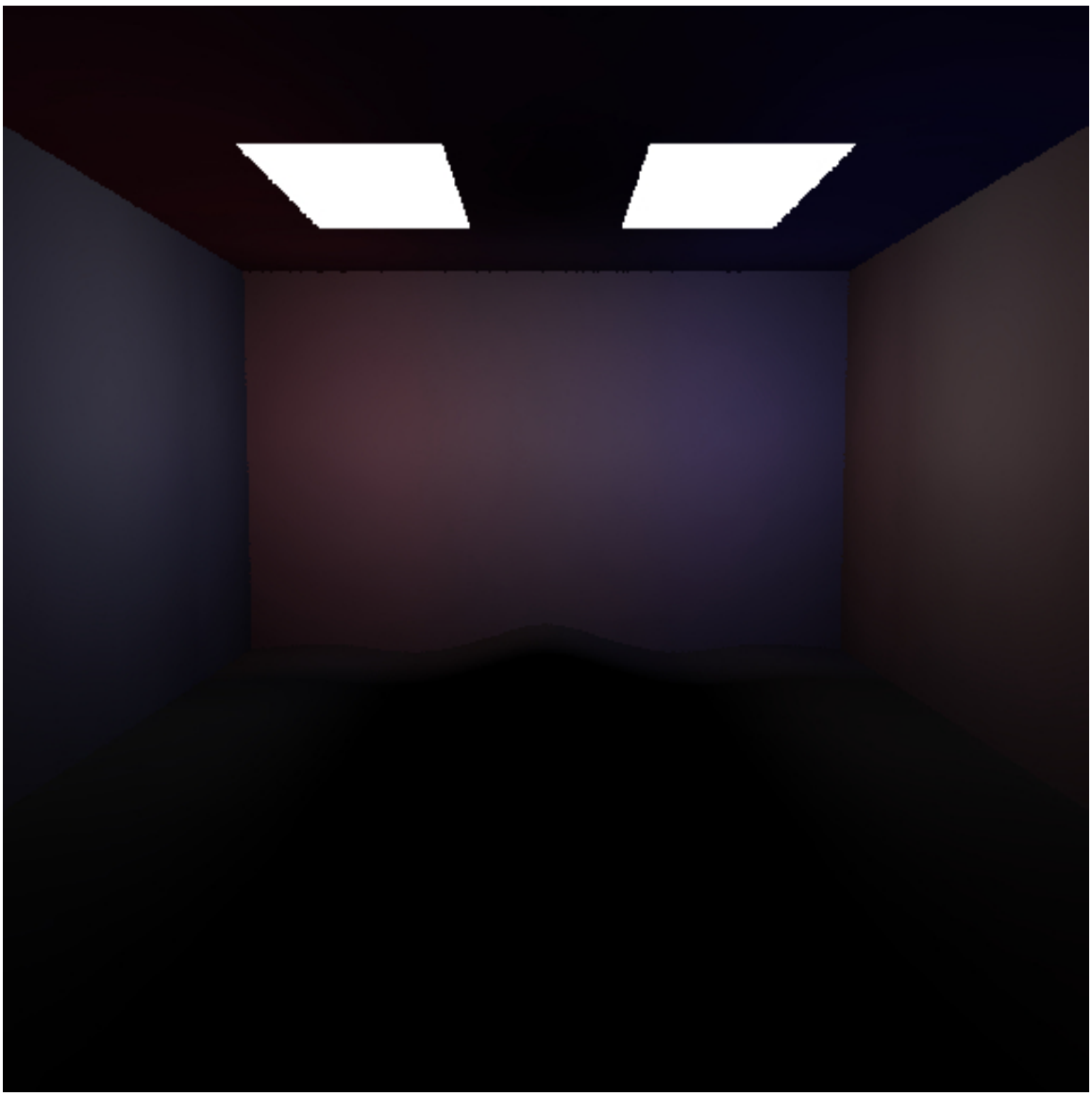


Figure 7. Indirect Illumination with Water (142 seconds)

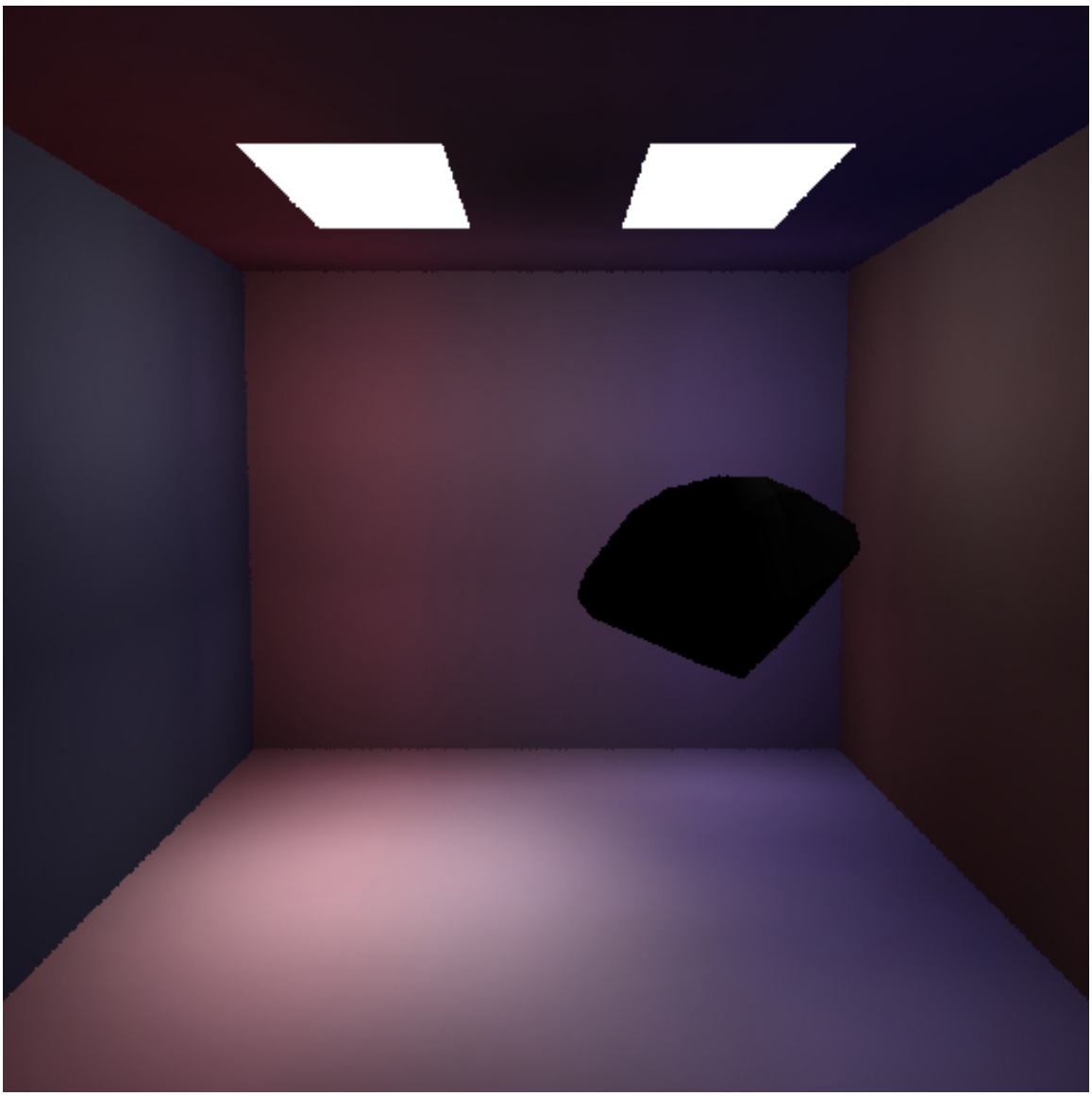


Figure 8. Indirect Illumination with Diamond (413 seconds)

Caustics

The caustics are generated by sampling photons from the caustic photon map on the bounces of the rays traced from camera. We created another independent kd-tree of photon mapping for caustics. More than photon mapping for indirect illumination, 100,000 photons are emitted from each area lights to emphasize

caustics effect. Moreover, to exactly show the caustics effects of diamond, we use radius of 0.5 instead of 2.0 for radius search of kd-tree. We collect the photons not only the surface of first hit of the ray from the camera to the scene, but also accumulate the color with the possible reflected and refracted rays.

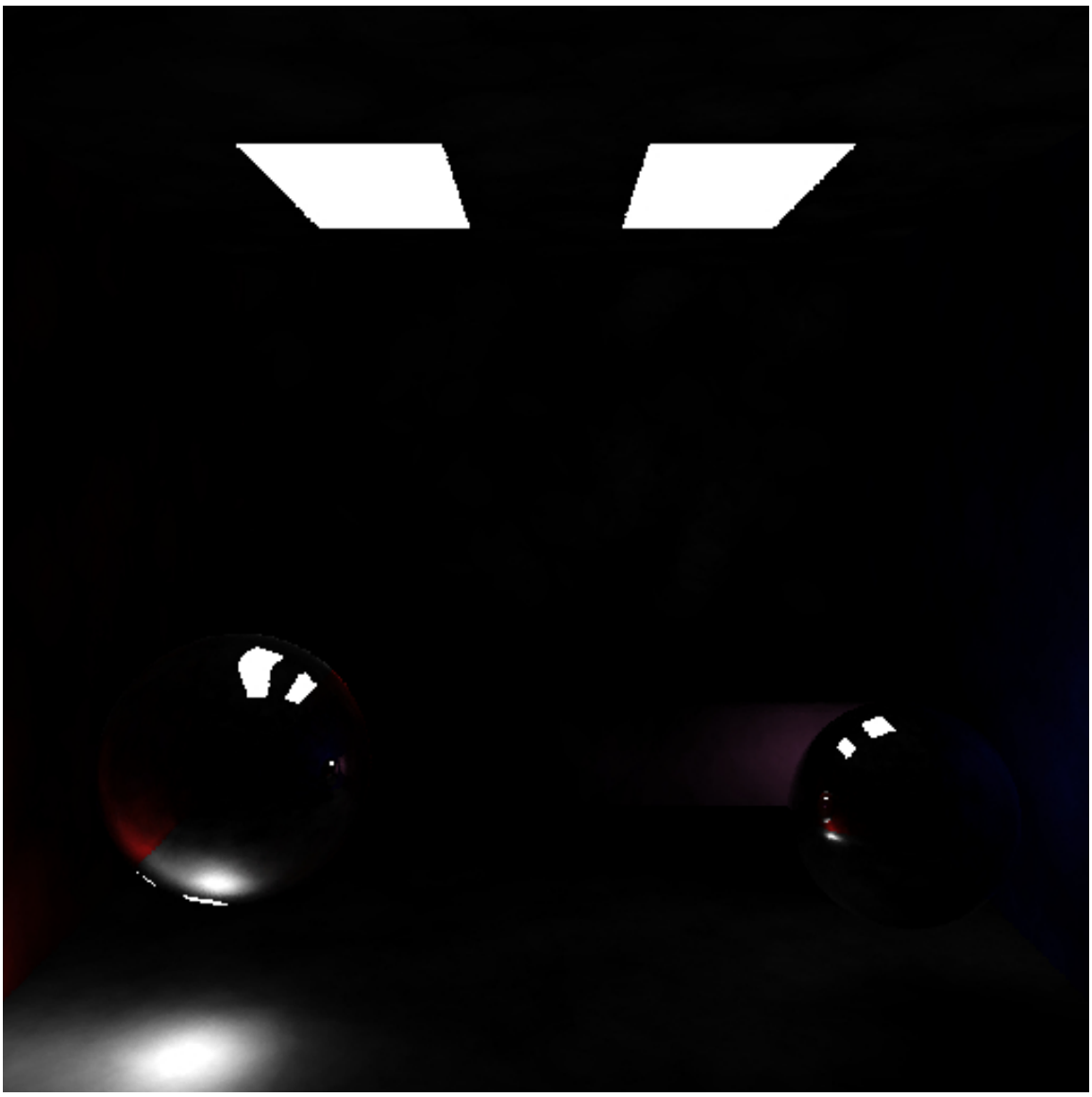


Figure 9. Caustics with Simple Objects (4.4 seconds)

Figure 10. Caustics with Water (70.2 seconds)

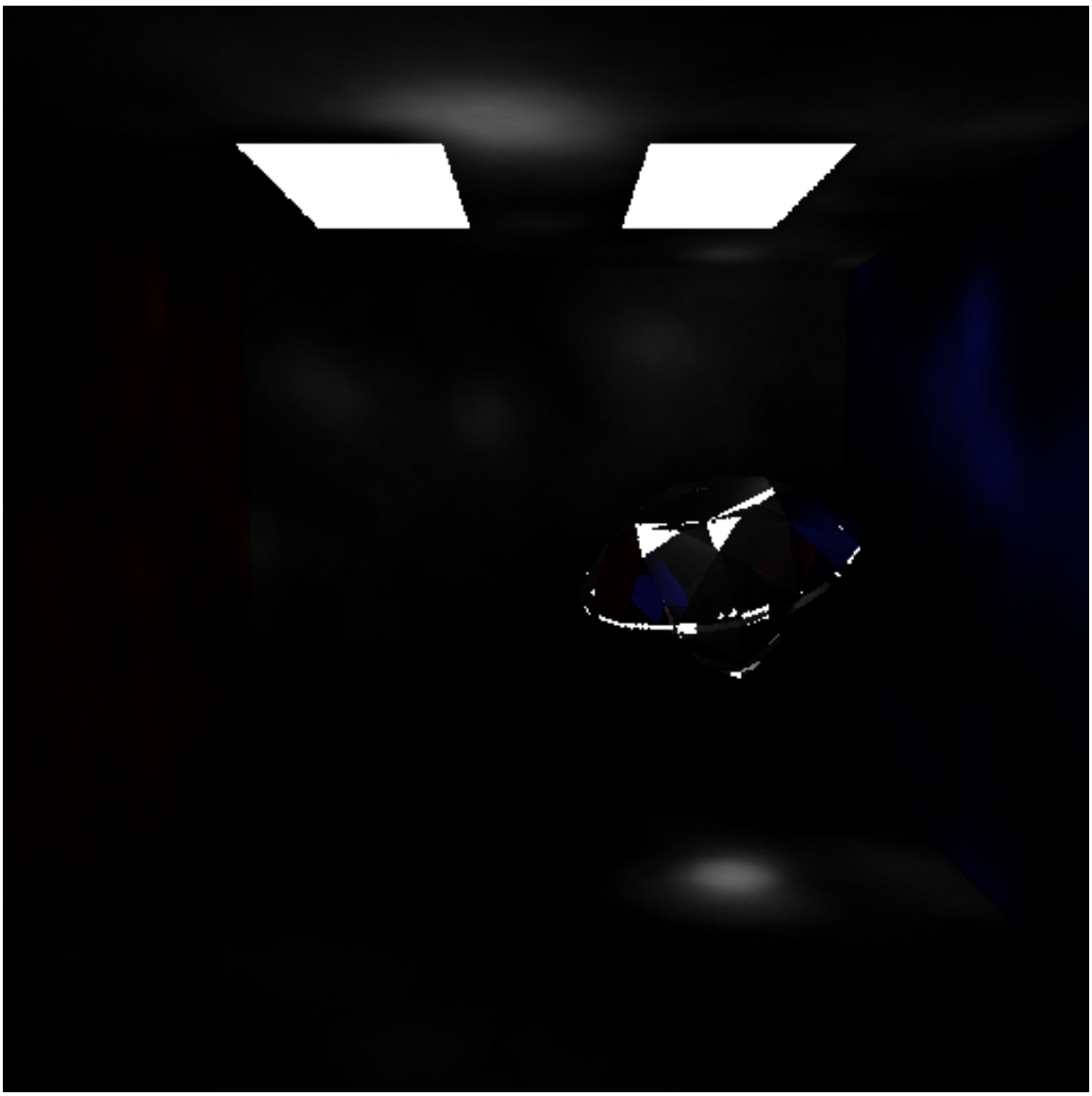
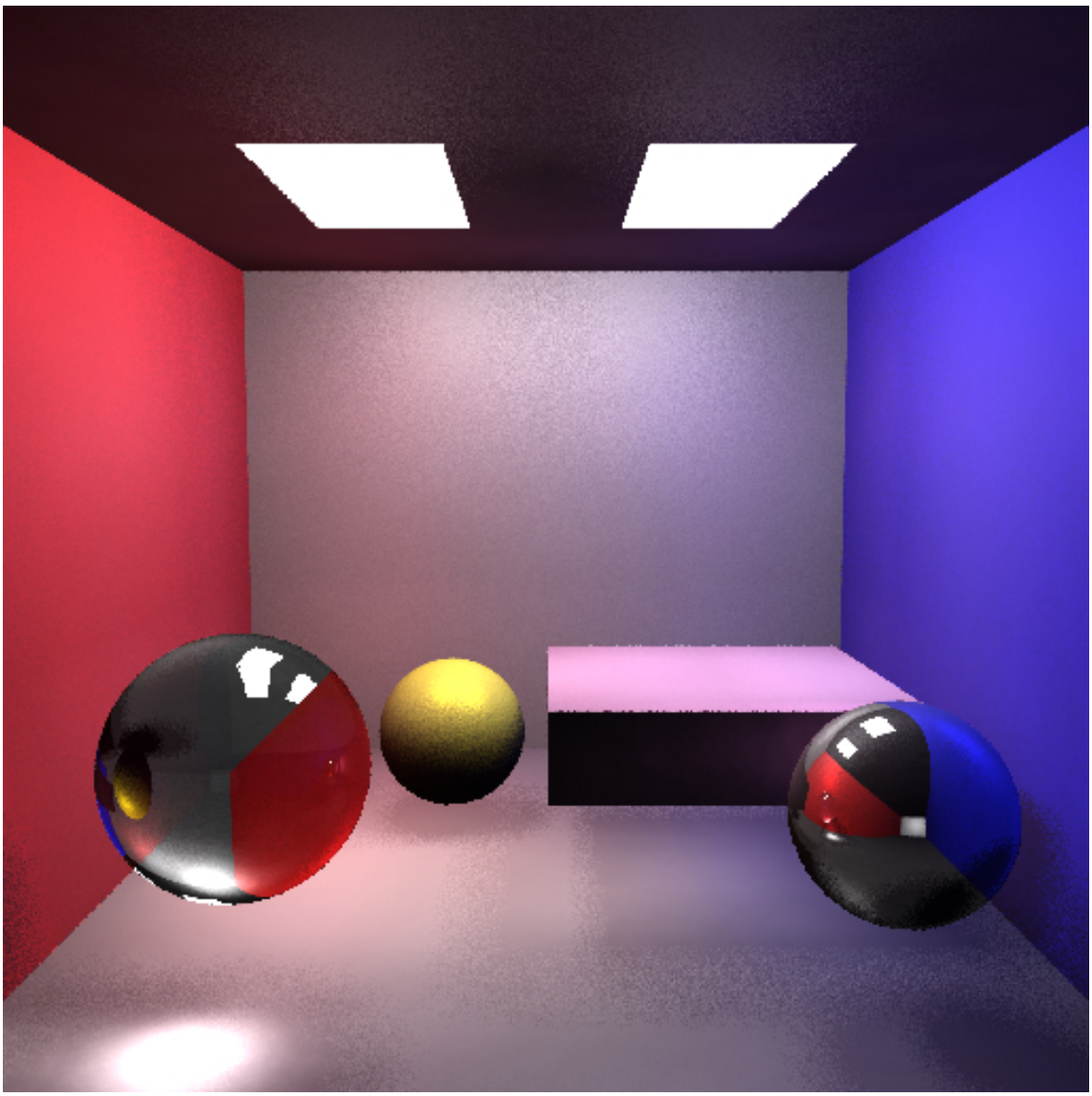


Figure 11. Caustics with Diamond (85 seconds)

All Effects

In the end, we will accumulate direct illumination, indirect illumination, and caustics with more jitter sampling times for anti-aliasing to create a more spectacular and real picture than conventional ray tracing and path tracing.



FIugre 12. All Three Effects with Simple Objects (1.03e+03 seconds)

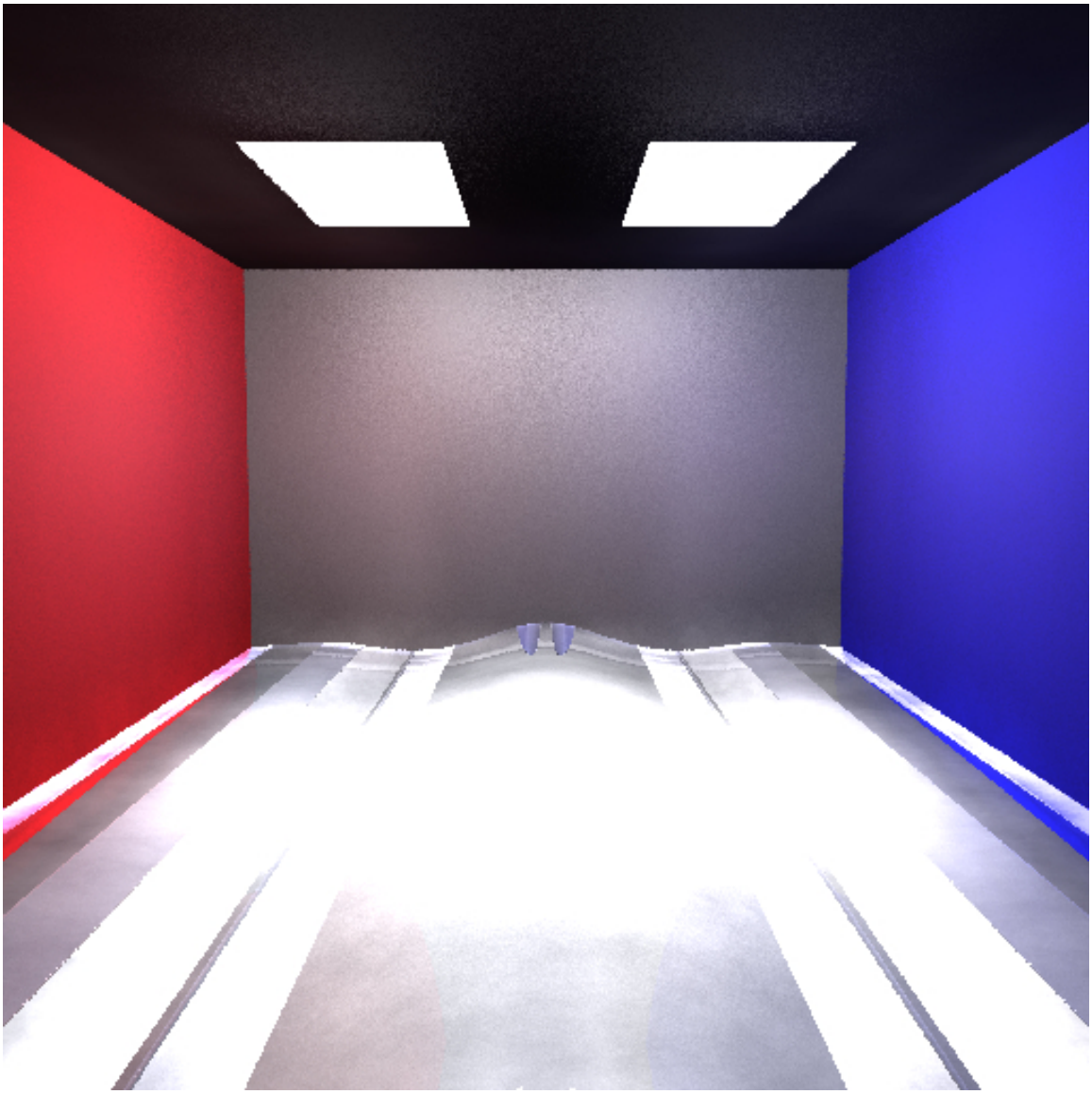
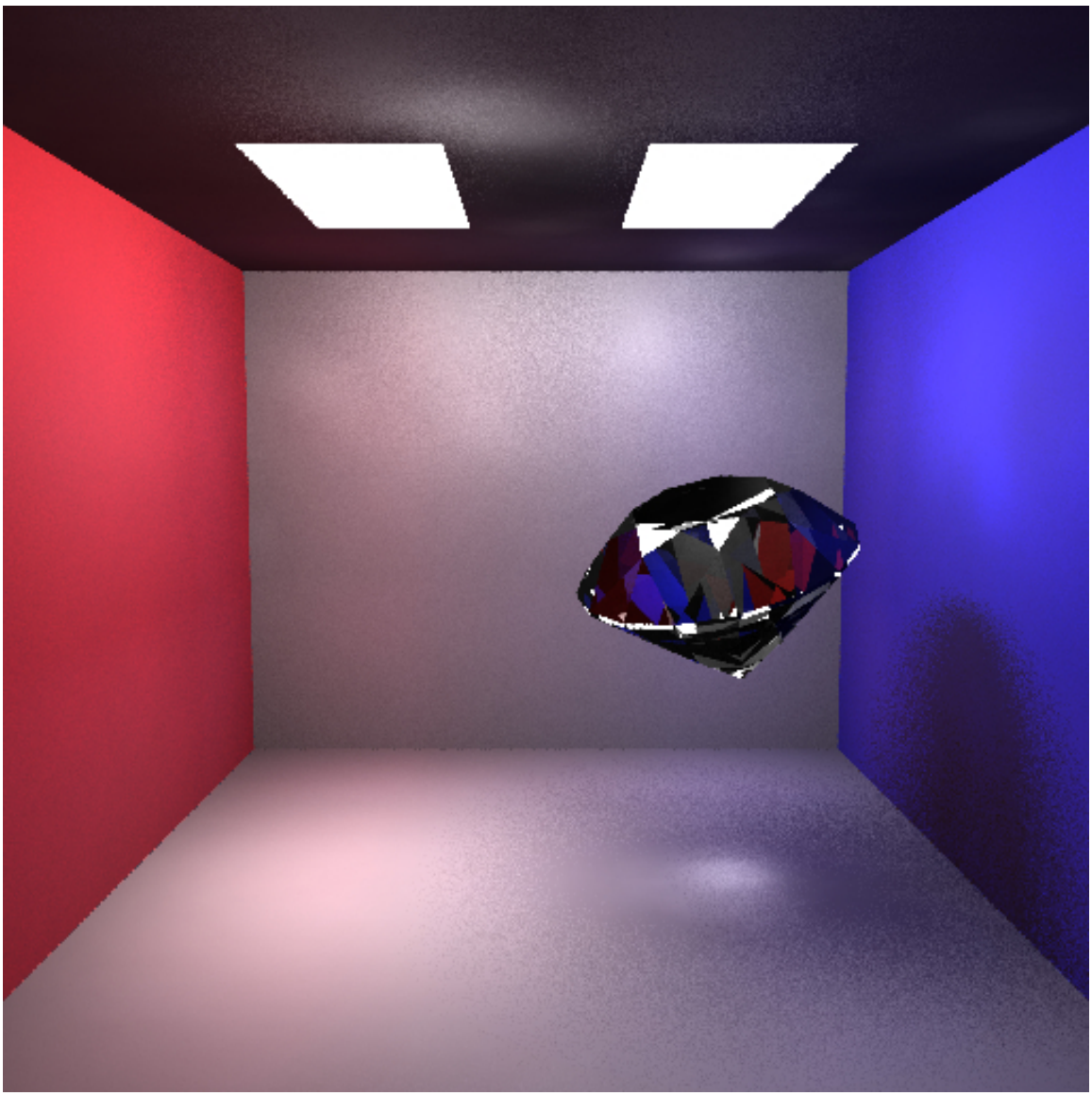


Figure 13. All Three Effects with Water (232 seconds)



Fiugre 14. All Three Effects with Diamond (232 seconds)

Results - Final Scene

We import all the objects and slightly refine all the effects by adjusting the positions of objects. This work is quite inefficient since single image containing all the objects is much more computation intensive because of more reflection and refractions. We use one single image to show the real heights of positions of all the

objects, since the water looks shallower due to the refraction of surface between two different mediums. All the images are with 512 x 512 resolutions except the final one (1024 x 1024). They are with 36, 4, 4 jitter sampling times for direct illumination, indirect illumination, and caustics, respectively.

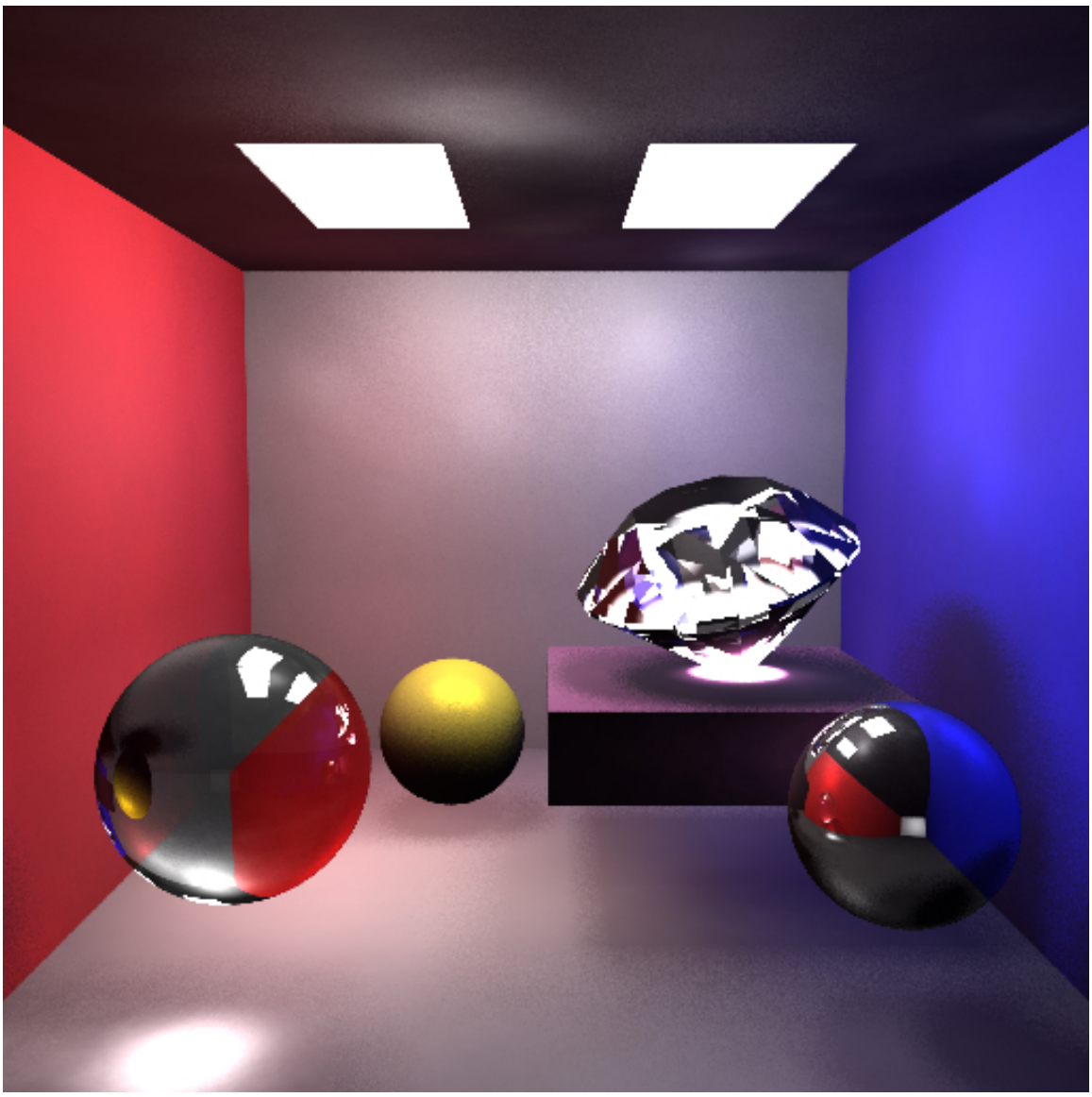


Figure 15. All three effects with all objects without Water (4.08e+03 seconds)

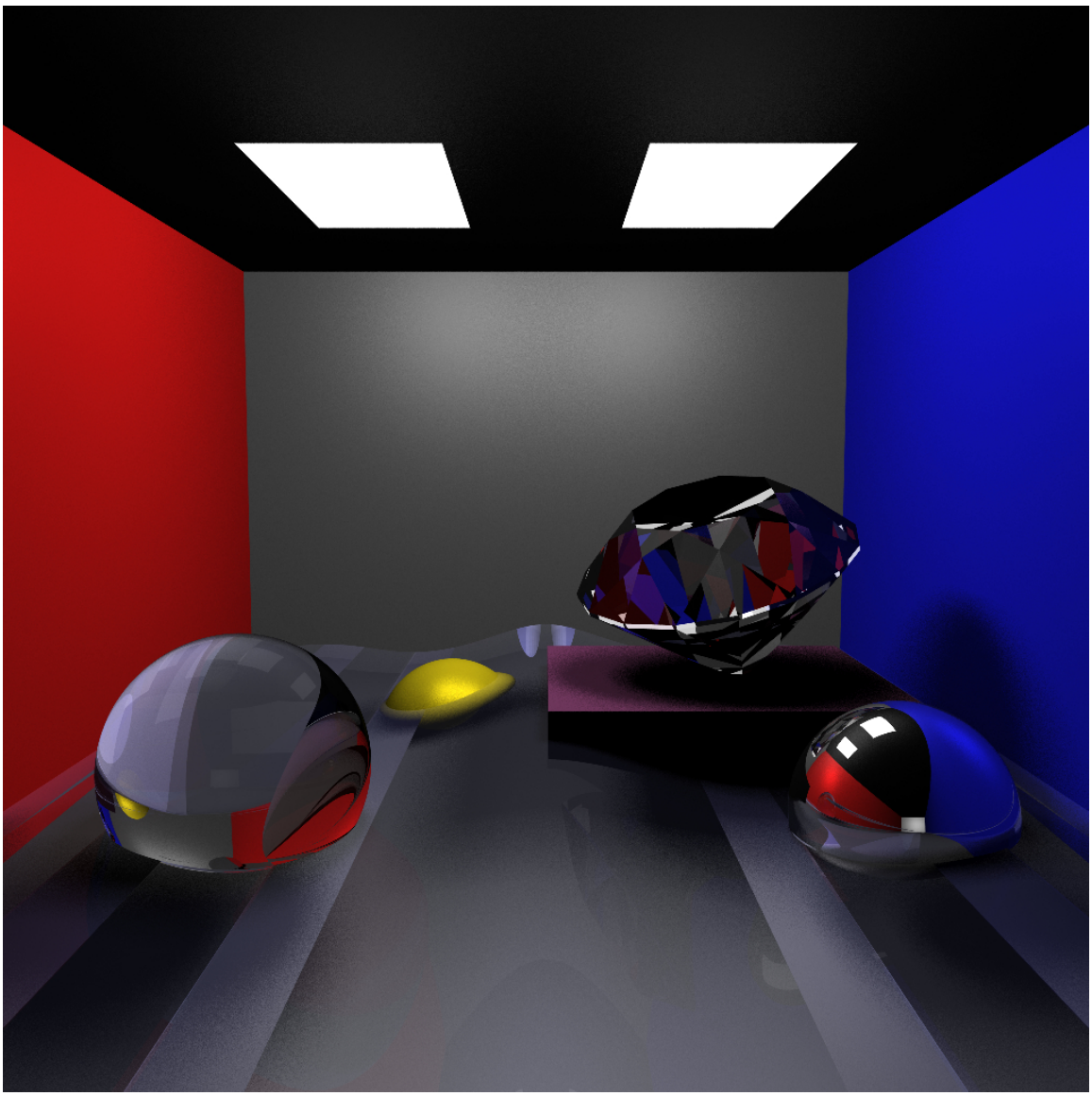
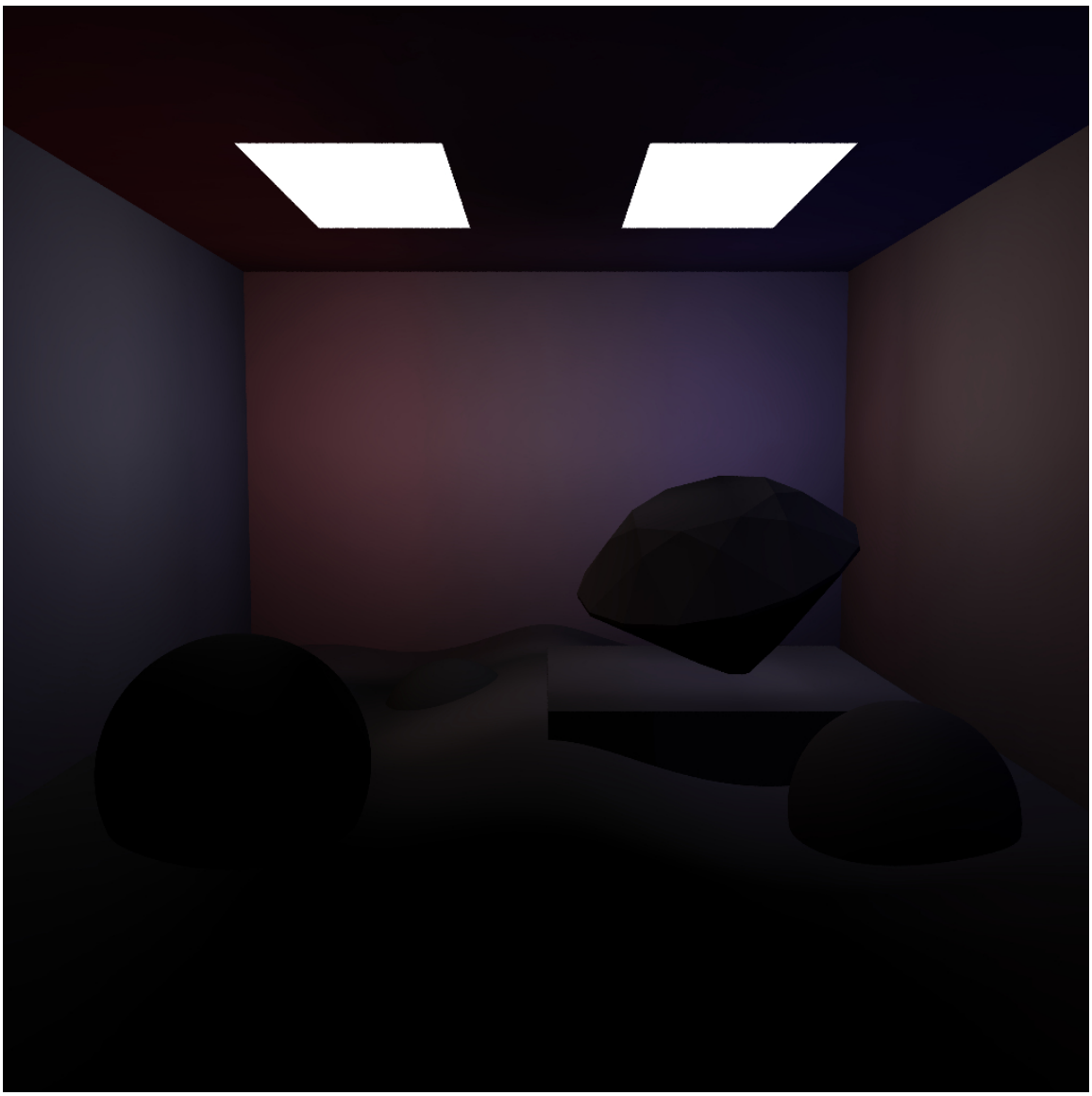
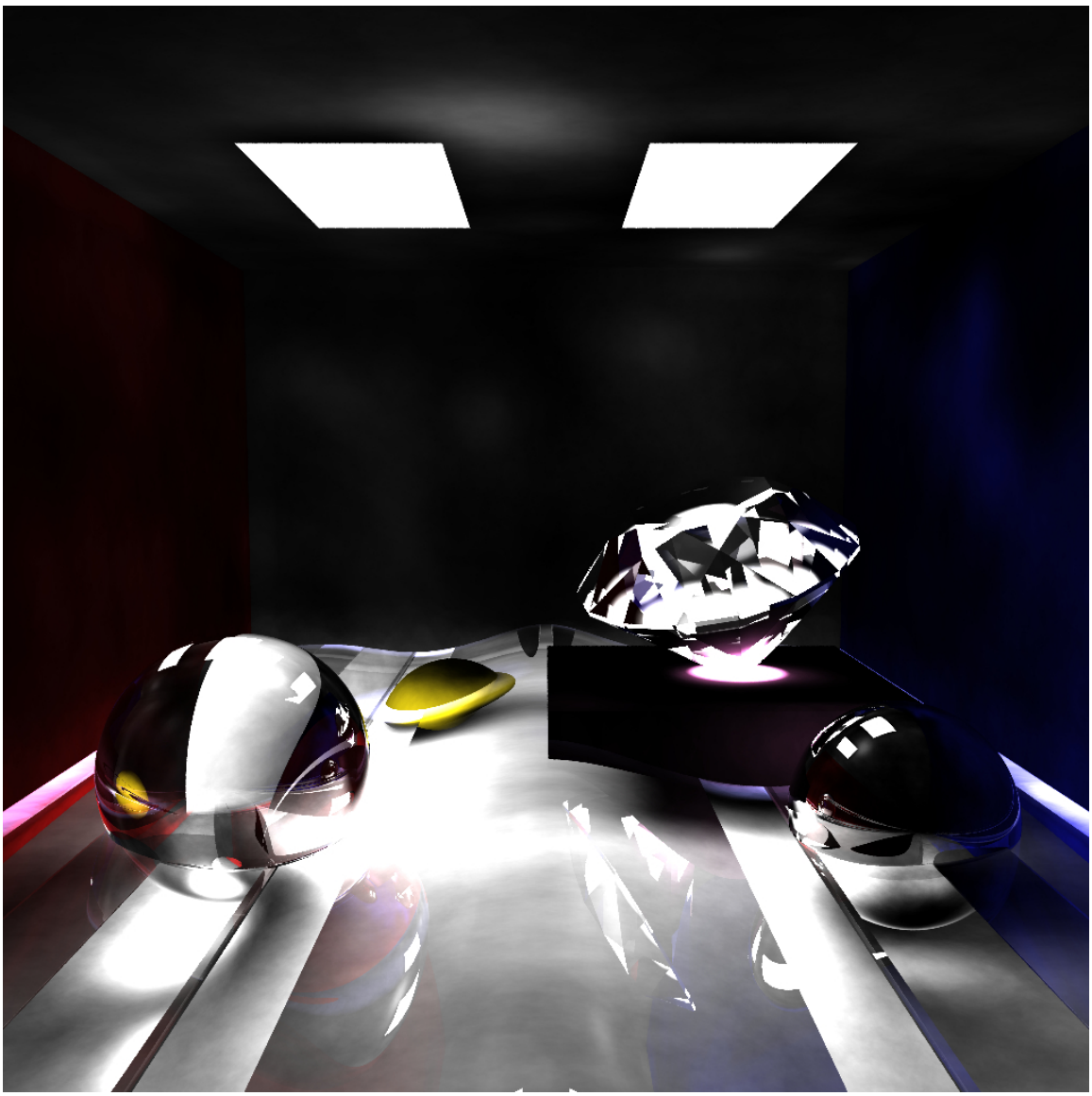


Figure 16. Direct Illumination with all objects (1.58e+04 seconds). 1024x1024 with 36, 4, 4 jitter sampling times for direct illumination, indirect illumination, and caustics, respectively.



FIugre 17. Indirect Illumination with all objects (3.11e+03 seconds). 1024x1024 with 36, 4, 4 jitter sampling times for direct illumination, indirect illumination, and caustics, respectively.



Fiugre 18. Caustics with all objects (2.97e+03 seconds). 1024x1024 with 36, 4, 4 jitter sampling times for direct illumination, indirect illumination, and caustics, respectively.

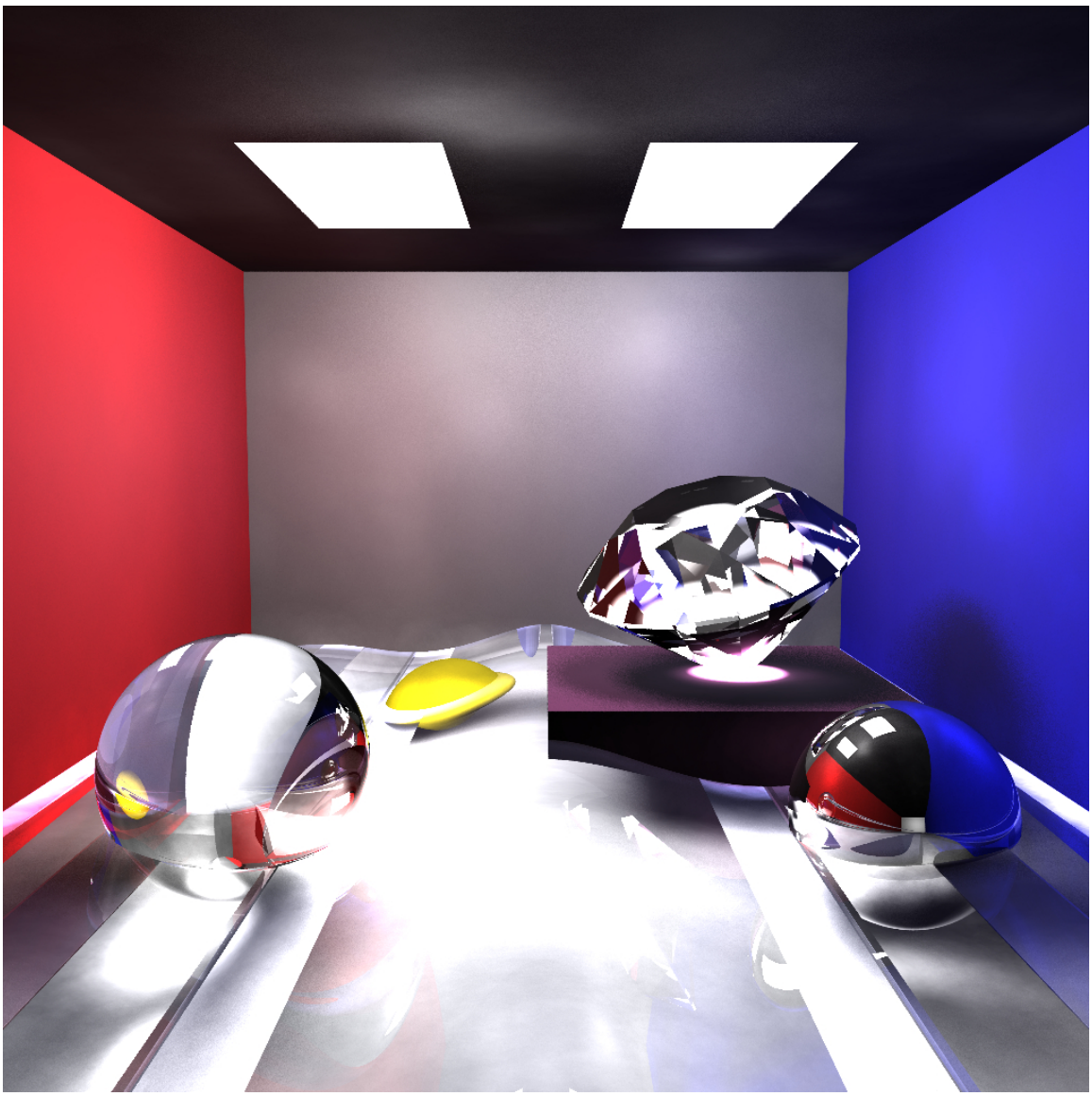


Figure19. All Three Effects with all objects (2.1e+04 seconds). 1024x1024 with 36, 4, 4 jitter sampling times for direct illumination, indirect illumination, and caustics, respectively.

References

[1] Computer Graphics course website of MIT Computer Science And Artificial Intelligence Laboratory – Computer Graphics Group
http://groups.csail.mit.edu/graphics/classes/6.837/F04/assignments/assignment4/diamond.obj

[2] Pluker coordinates
http://en.wikipedia.org/wiki/Pl%C3%BCcker_coordinates

[3] Intersection Algorithm for polygons
http://graphics.di.uoa.gr/Downloads/papers/journals/p19.pdf

[4] John Tsiombikas' website http://nuclear.mutantstargoat.com/

[5] Photon mapping on Wikipedia
http://en.wikipedia.org/wiki/Photon_mapping

[6] Photon mapping algorithm
http://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/photon_mapping/PhotonMapping.html