Photon Mapping

Contents

Oihan Abruña

© DigiPen Institute of Technology Europe - Bilbao

Problem

Simulating global illumination is a complex and computationally challenging task. While techniques such as *Monte Carlo* ray tracing can accurately capture both direct and indirect illumination, they often come with significant drawbacks. *Monte Carlo* methods often require a very large number of samples to produce low noise results, making them slow and inefficient for scenes with challenging lighting conditions.

A major difficulty arises when rendering caustics, which are focused patterns of light formed by reflection or refraction though specular surfaces. These phenomena are especially hard to capture efficiently, since their highly concentrated nature requires dense sampling to avoid noise or missing details.

These challenges highlight the need for alternative techniques, such as *Photon Mapping*, or even hybrid approaches that combine the methods to exploit the strengths of both approaches.

Solution

Photon Mapping addresses the challenges of global illumination and caustics by using a two-pass rendering technique, which separates the simulation of light transport from the final image generation.

The primary advantage of *Photon Mapping* over purely *Monte Carlo* based methods is efficiency, especially in scenes dominated by specular interactions and caustics. While *Photon Mapping* requires additional memory to store photon data, it offers faster image generation for these lighting effects. The resulting error tends to be generally less visually disturbing than the high-frequency stochastic noise characteristic of *Monte Carlo* ray tracing.

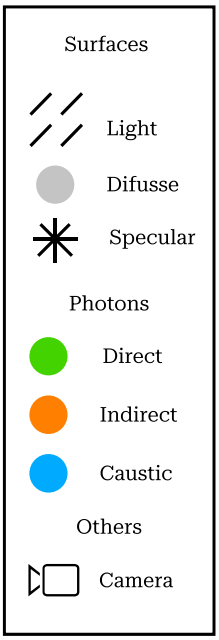


Figure 1: This illustration defines the symbols and icons used to represent light sources, photons, rays, surfaces, and interactions in subsequent diagrams.

© DigiPen Institute of Technology Europe - Bilbao

1. Photon Emission (Light-driven pass)

In the first pass, photons are emitted directly from the light sources and traced through the scene. Their interactions with surfaces are stored in photon maps when they hit a diffuse surface. These maps store information about the position, direction, and energy of each photon, enabling accurate reconstruction of lighting.

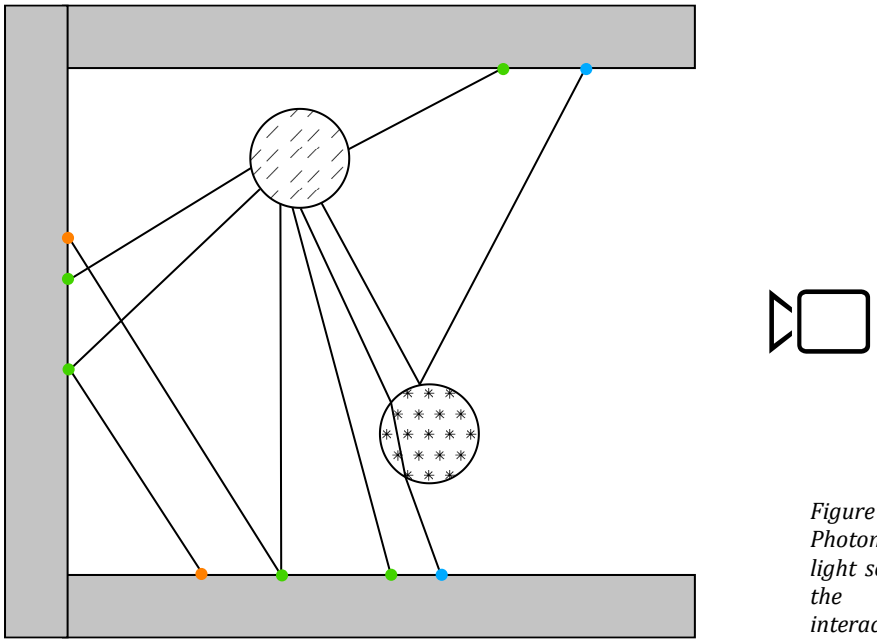


Figure 2: Photon emission pass: Photons are emitted from the light source and traced through the scene, storing their interactions in the photon map.

2. Rendering (Eye-driven pass)

In the second pass, rays are traced from the camera into the scene. At diffuse surfaces, nearby photons are gathered from the photon map to estimate illumination. Reflective and refractive surfaces are handled by bouncing rays using *Monte Carlo* until they reach a diffuse surface for photon gathering.

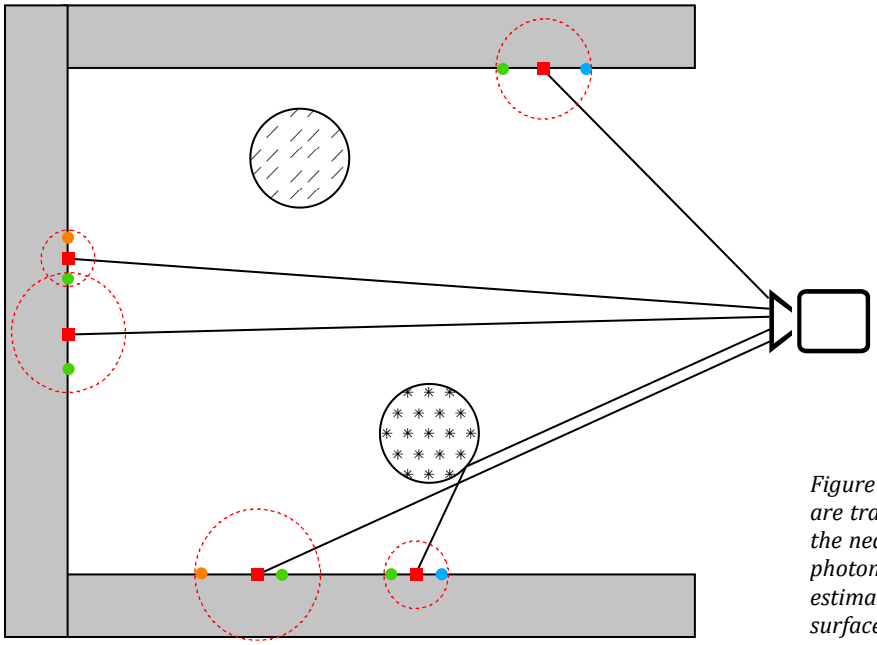


Figure 3: Rendering pass: Rays are traced from the camera, and the nearby photons stored in the photon map are gathered to estimate illumination at each surface point.

© DigiPen Institute of Technology Europe - Bilbao

Implementation

Global illumination

Global illumination can be divided into direct and indirect illumination.

- Direct illumination represents the direct hits of photons from a light source into a surface.
- Indirect illumination represents hits of photons that have been previously bounced.

Direct illumination

In the original paper [1], it suggests computing direct illumination using *shadow rays*. Shadow rays are traced from a surface point directly toward a light source to determine whether the point is illuminated or occluded. This approach works well for point light sources, where a single shadow ray can provide an accurate estimate of direct lighting.

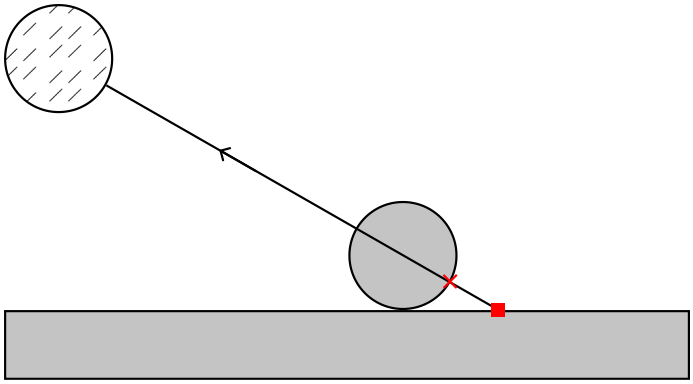


Figure 4: Illustration of shadow rays.

However, for area lights, multiple samples per surface point are required to produce smooth and accurate illumination. In practice, this means casting several shadow rays for every diffuse surface hit, which can significantly increase computation time.

The purpose of this project is to fully explore the capabilities of *Photon Mapping*. For this reason, a dedicated photon map that stores only photons directly hitting diffuse surfaces will be used instead of shadow rays.

© DigiPen Institute of Technology Europe - Bilbao

Photon storing

Photon maps are data structures used to store photons during the first pass of the algorithm, containing information about the illumination of the scene. This information is then used in the second pass to estimate the incoming light at each surface point.

Photon Information

Information stored in photon maps:

- $\vec{x}$: Photon position.
- $\vec{\omega}$: Photon direction.
- ϕ: Photon power.

Multiple Photon Maps

While it is possible to store all photons in a single photon map, this approach limits both accuracy and performance. For a photon map to provide reliable lighting estimates, the photons it contains should have similar power, ensuring a well-distributed representation of light in the scene.

To overcome this limitation, multiple photon maps are used, each corresponding to a different type of illumination. Furthermore, this allows control over the fraction of light power assigned to each type of contribution.

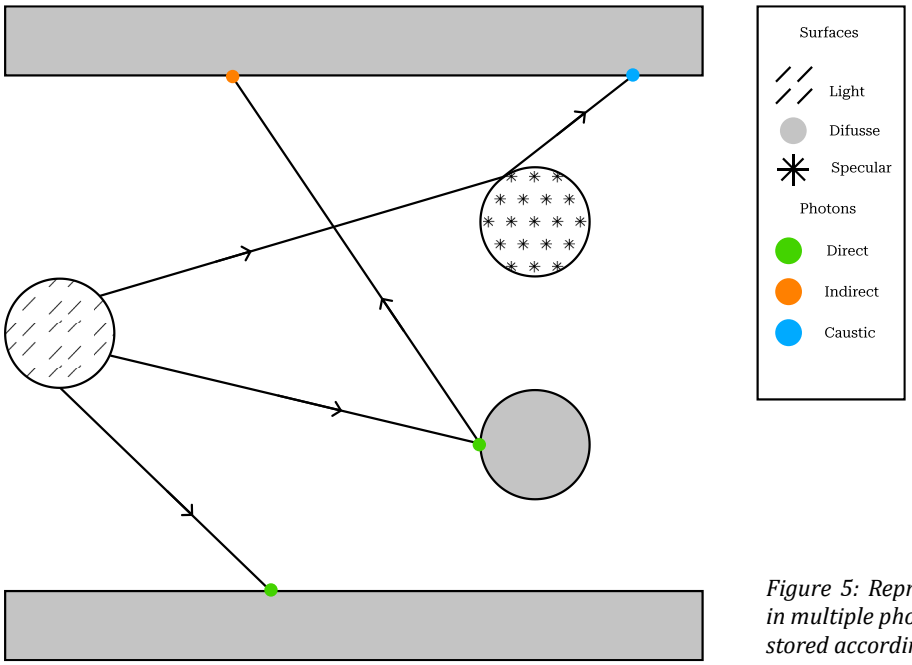


Figure 5: Representation of photons in multiple photon maps. Photons are stored according to their type

© DigiPen Institute of Technology Europe - Bilbao

Direct Photon Map

The direct photon map stores photons that travel directly from a light source to a diffuse surface without undergoing any prior interactions.

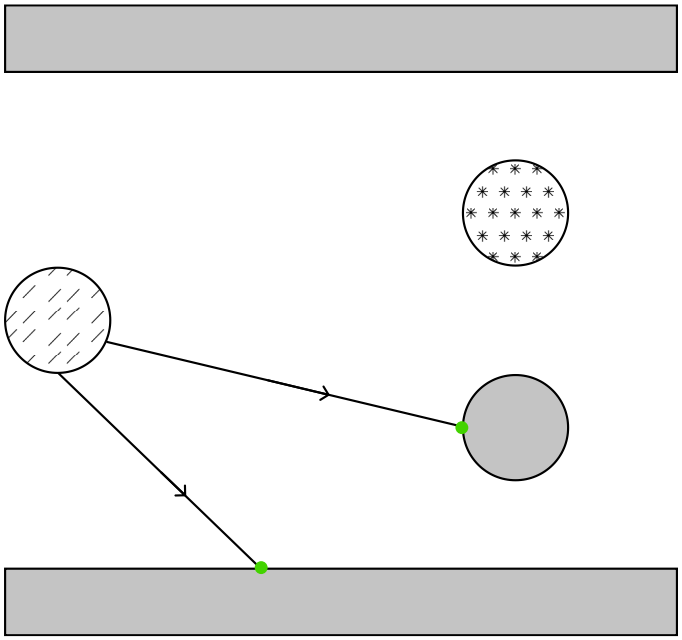


Figure 6: Illustration of direct photons.

The following images show the output obtained by visualizing the direct photon map and estimating the direct illumination using it.

Output:

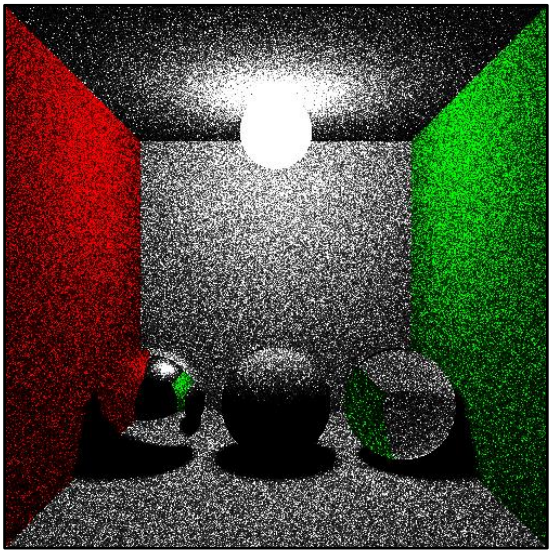


Figure 7: Direct Photon Map visualized directly.

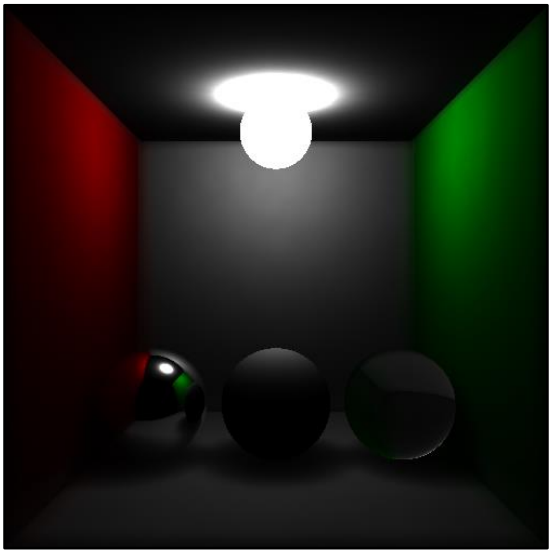


Figure 8: Direct Photon Map used to estimate direct illumination.

© DigiPen Institute of Technology Europe - Bilbao

Indirect Photon Map

This map stores photons that first hit a diffuse surface, scatter, and subsequently reach another diffuse surface.

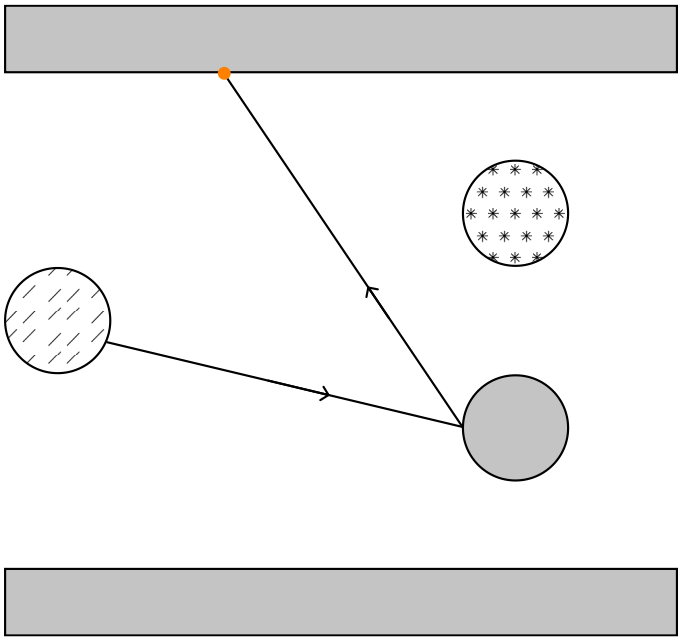


Figure 9: Illustration of indirect photons.

The following images show the output obtained by visualizing the indirect photon map and estimating the indirect illumination using it.

Output:

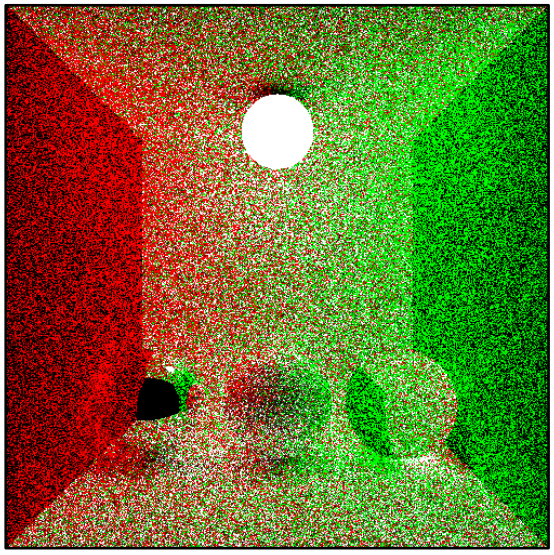


Figure 10: Indirect Photon Map visualized directly.

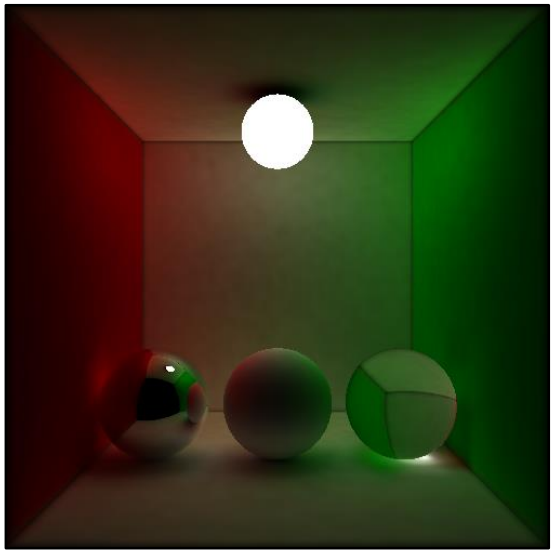


Figure 11: Indirect Photon Map used to estimate indirect illumination.

© DigiPen Institute of Technology Europe - Bilbao

Caustics Photon Map

The caustics photon map stores photons that have interacted exclusively with specular surfaces before hitting a diffuse surface.

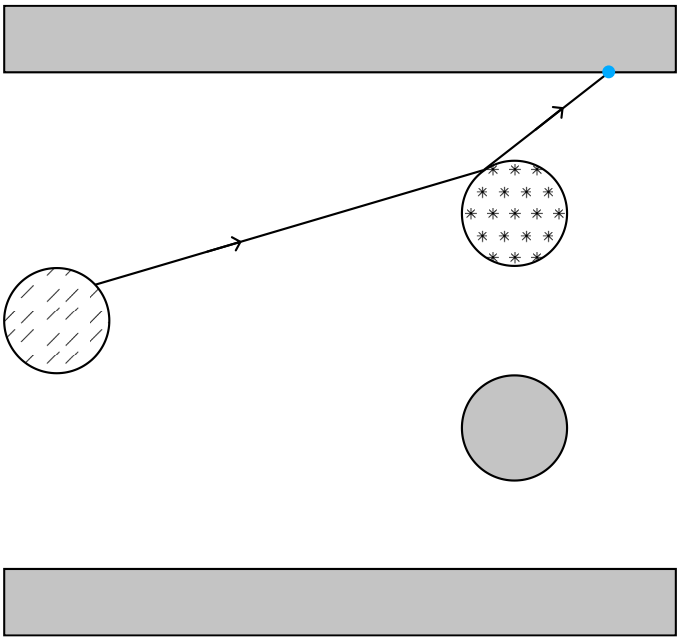


Figure 12: Illustration of caustic photons.

The following images show the output obtained by visualizing the caustics photon map and estimating the caustics using it.

Output:

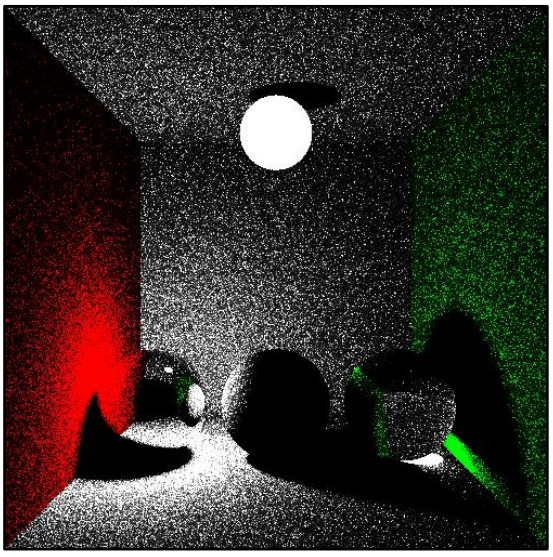


Figure 13: Caustics Photon Map visualized directly.

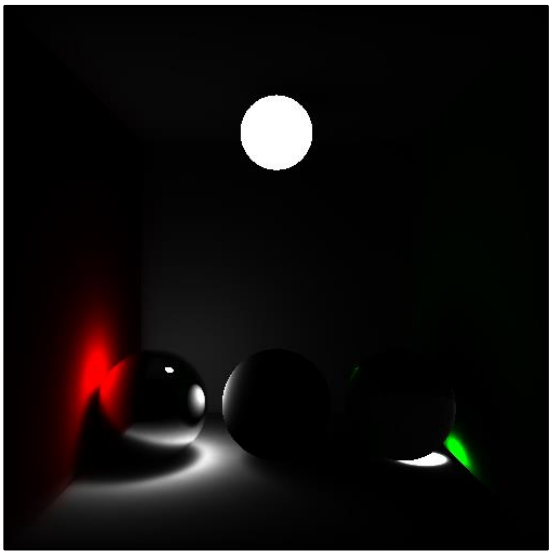


Figure 14: Caustics Photon Map used to estimate caustics.

© DigiPen Institute of Technology Europe - Bilbao

Combined Photon Maps

Once the three photon maps are constructed, they are combined during the rendering pass to reconstruct the full illumination at each surface point.

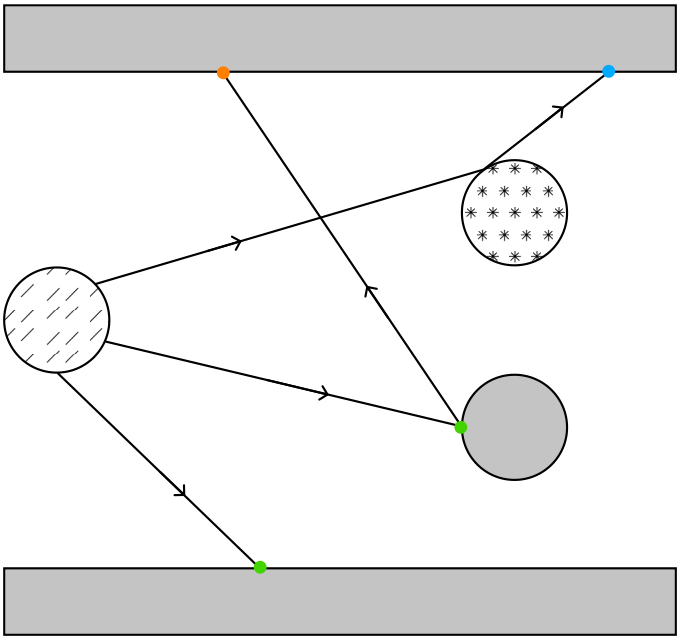


Figure 15: Illustration of all types of photons.

The following images show the output obtained by visualizing all the photon maps and estimating the lighting using them.

Output:

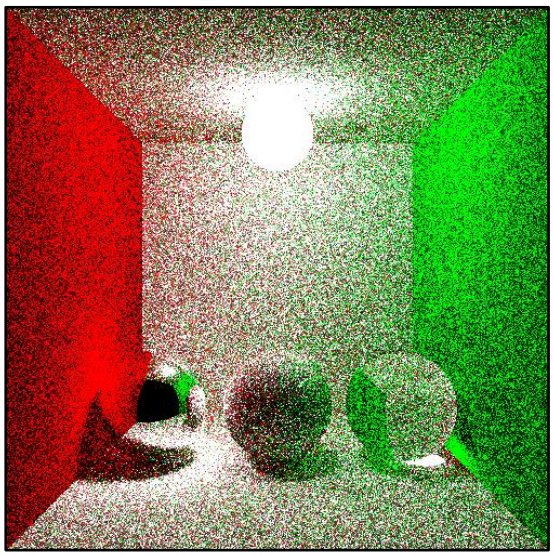


Figure 16: All Photon Maps visualized directly.

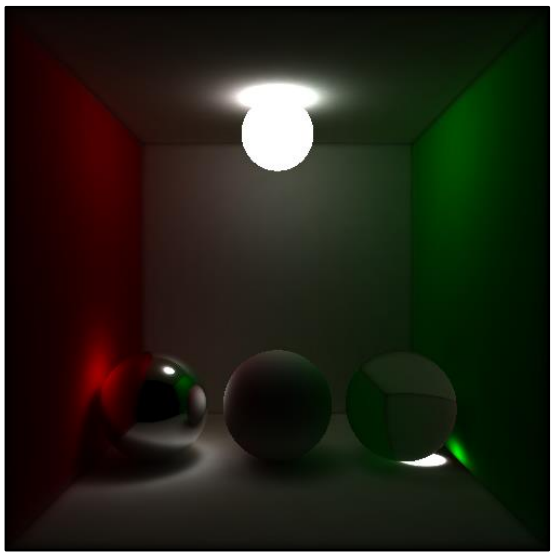


Figure 17: All Photon Maps used to estimate lighting.

© DigiPen Institute of Technology Europe - Bilbao

K-d tree

Photons are stored in a *balanced k-d tree*, a spatial data structure commonly used for organizing points in a 3D space. The k-d tree enables efficient nearest-neighbor queries, which are essential for fast photon gathering during the rendering pass.

Construction

This procedure produces a balanced k-d tree, ensuring that nearest-neighbor searches are efficient and approximately $O(\log n)$ per query.

1. Select Split Axis

For simplicity, the split axis can be selected based on the depth of the node in the tree:

$$axis = depth \bmod num\_dimensions$$

In 3D, this cycles through x, y and z.

2. Select Split Position

For the set of points in the current node, the median point along the selected axis is chosen.

3. Divide points

Points go to one of the children based on if the coordinate is less/greater than the split position.

4. Recurse

Step 1-3 are repeated recursively for each child node until each leaf contains a defined minimum number of points.

Illustration

The construction is illustrated step by step in the following figures. For simplicity, the illustrations are shown in 2D. Each figure corresponds to an iteration in the recursive splitting process:

© DigiPen Institute of Technology Europe - Bilbao

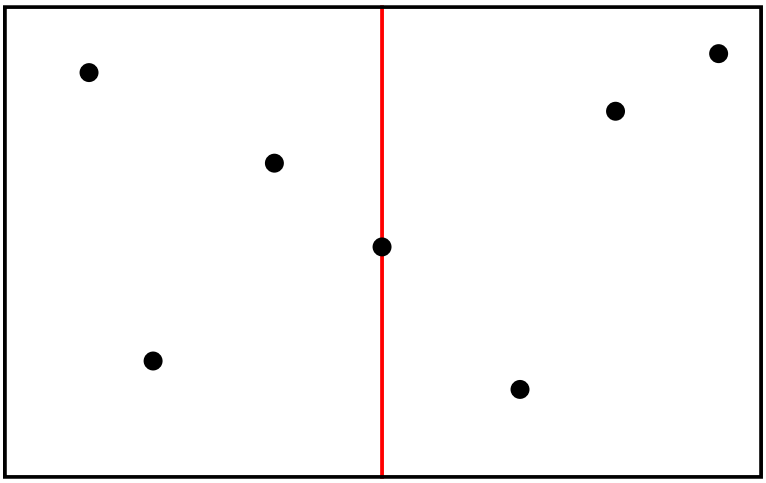


Figure 18: First split along the X-axis.

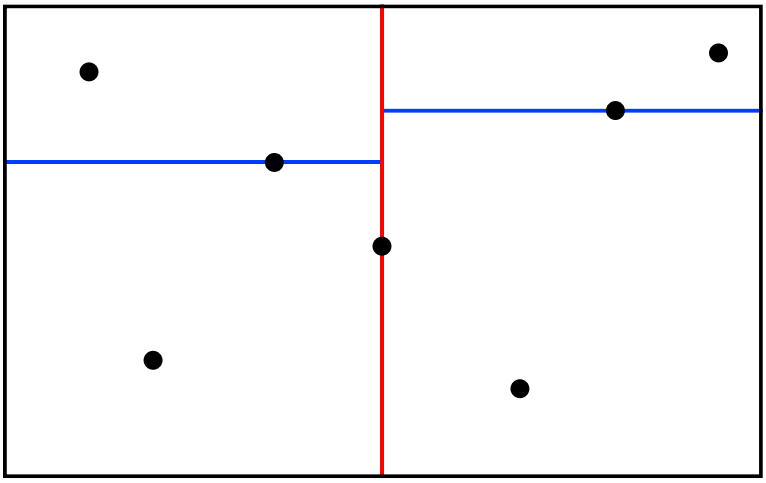


Figure 19: Second split along the Y-axis.

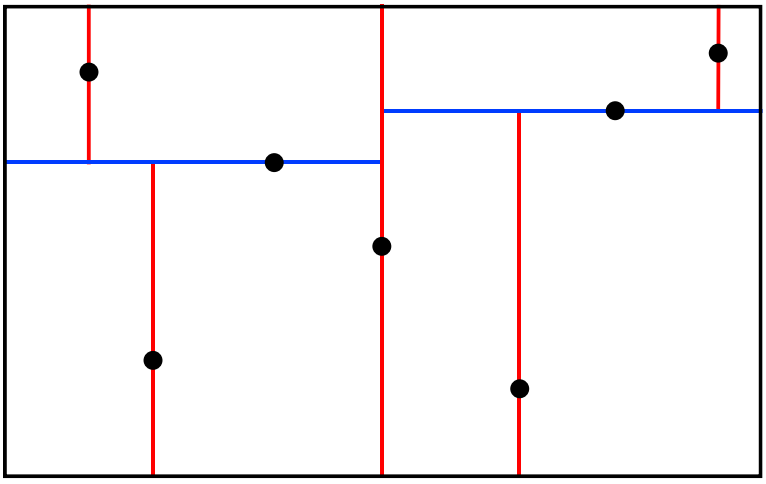


Figure 20: Third split along the X-axis.

© DigiPen Institute of Technology Europe - Bilbao

N-Nearest Neighbor Search

Nearest-neighbor search is used to efficiently find the closest photons to a point in the k-d tree.

1. Select child to explore first

The target point is compared with the current node's splitting axis. The child node that is more likely to contain the nearest neighbor is traversed first.

2. Explore the selected child

The selected child node is recursively traversed and the current nearest neighbors are updated if a close point is found.

3. Check whether the other child must be explored

After exploring the first child, if the distance from the splitting plane to the target point is less than the current best distance or if not enough neighbors have been found yet, the other child may contain closer points and must be explored by repeating step 2 recursively.

4. Return the nearest points

Once all relevant nodes are checked, the closest points found as the nearest neighbors are returned.

Illustration

The nearest-neighbor search is illustrated step by step in the following figures. For simplicity, the illustrations are shown in 2D and with only one nearest neighbor. Each figure corresponds to a step in the recursive search process:

© DigiPen Institute of Technology Europe - Bilbao

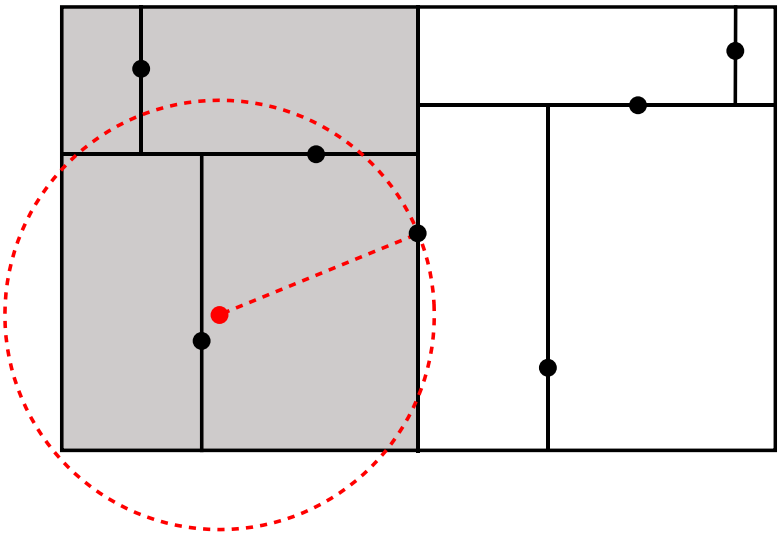


Figure 21: First recursion into the selected child.

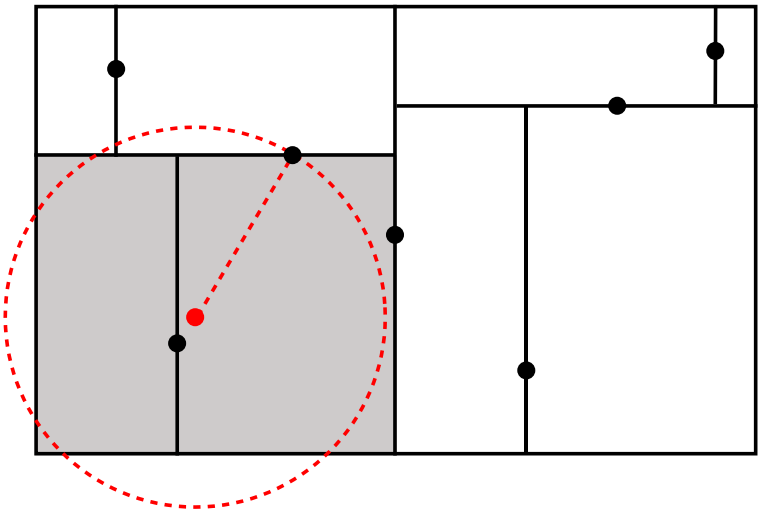


Figure 22: Second recursion into the selected child.

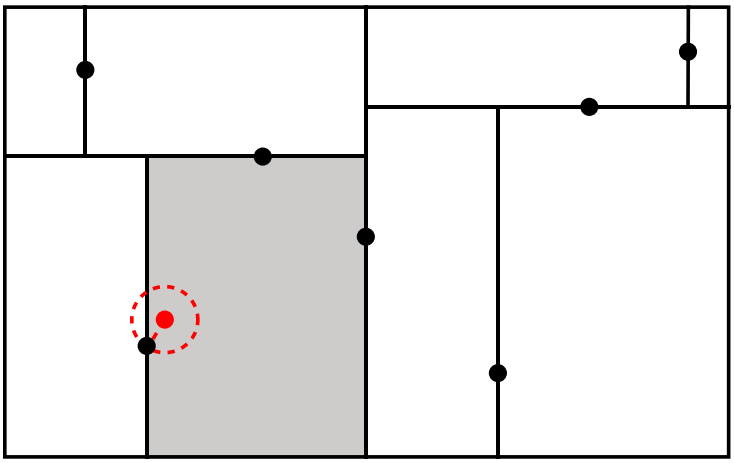


Figure 23: Third recursion into the selected child.

© DigiPen Institute of Technology Europe - Bilbao

Photon emission

Photon emission is the first step in *Photon Mapping*, where photons are emitted from light sources and traced through the scene.

Energy conservation

To conserve energy during *Photon Mapping*, the total light power emitted by a light source must be distributed across all photons. This ensures that the cumulative contribution of all photons matches the source's total emitted power.

The power carried by each photon is given by:

$$\Delta\Phi_p = \frac{\Phi}{N} \qquad \sum_{p=1}^{N} \Delta\Phi_p = \Phi$$

where:

- $\Delta\Phi_p$: Power carried by a single photon.
- Φ: Total power of the light source.
- N: Total number of photons emitted.

Emission direction

Photon maps for global illumination are generated by emitting photons in all directions from the light sources.

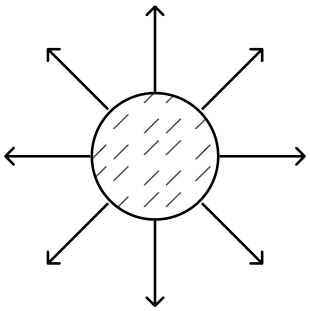


Figure 24: Emission directions for global illumination on a point light.

The photon map for caustics is generated by emitting photons towards specular surfaces from the light sources.

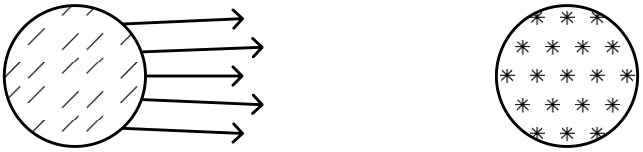


Figure 25: Emission directions for caustics on a spherical light.

© DigiPen Institute of Technology Europe - Bilbao

Area-Weighted Sampling of Specular Surfaces

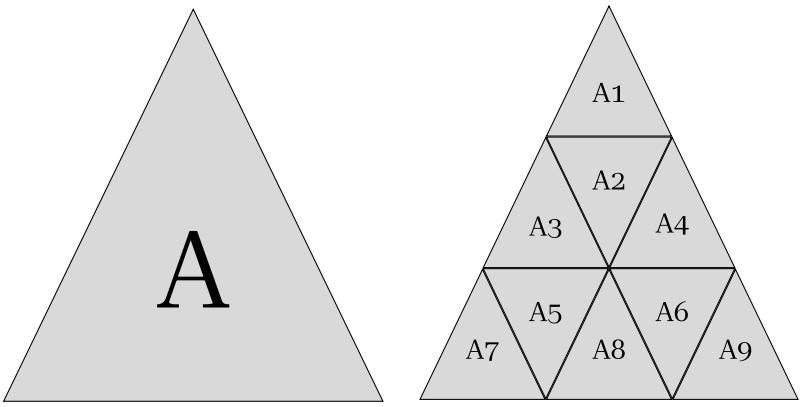
To efficiently generate caustics, photons must be emitted toward specular surfaces, since these are the only surfaces capable of producing concentrated light patterns through reflection or refraction.

Emitting photons uniformly in all directions would result in most photons missing the relevant specular objects, making caustic generation inefficient and noisy.

However, simply choosing a random triangle or mesh element is not sufficient. Consider two specular objects:

- Object A: a single large triangle.
- Object B: a mesh made of many small triangles.

Even if both objects have the same total surface area, randomly selecting one triangle would make Object B far more likely to be chosen, simply because it contains more triangles. But physically, both objects should receive equal photon density, since they contribute the same total area.



A = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9

Figure 26: A large triangle (left) and a subdivided copy formed by smaller triangles (right) have the same total area.

To address this, an area-weighted sampling strategy is used. Each specular object is assigned a probability proportional to its surface area:

$$P(S_i) = \frac{A_i}{\sum_{j=1}^{N} A_j}$$

where:

- A_i: The area of surface S_i.
- N: The total number of specular surfaces.

© DigiPen Institute of Technology Europe - Bilbao

Emission

The following images show the output of a test scene using different numbers of emitted photons.

In all cases, *0.1% of the emitted photons* were gathered per point, allowing a direct comparison of how the number of emitted photons affect the quality and smoothness of the image.

Output:

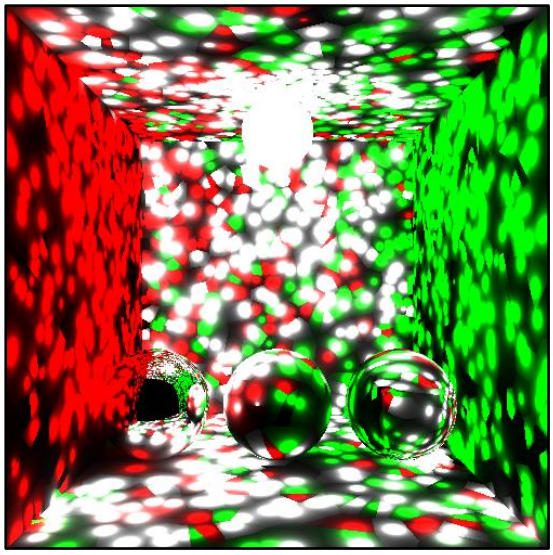


Figure 27: Lighting estimation using 1,000 emitted photons.

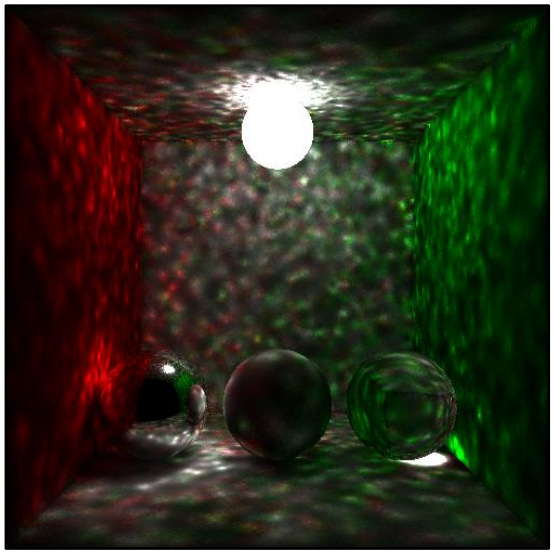


Figure 28: Lighting estimation using 10,000 emitted photons.

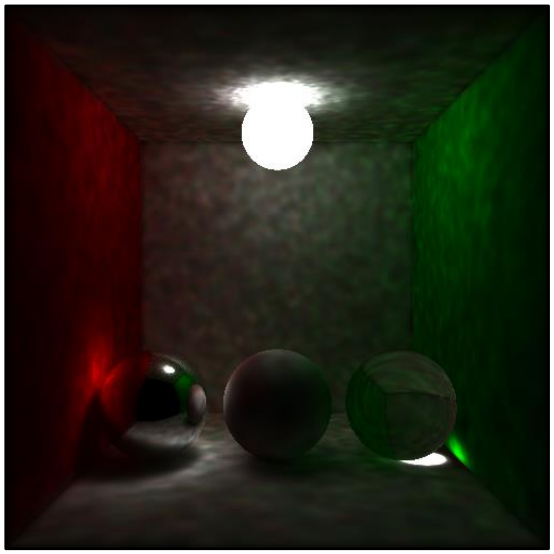


Figure 29: Lighting estimation using 100,000 emitted photons.

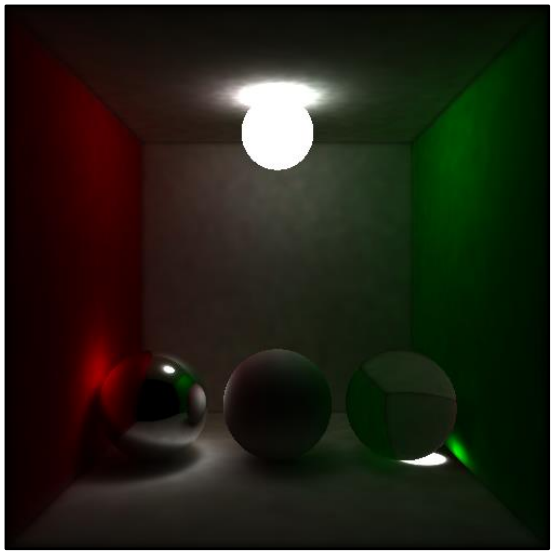


Figure 30: Lighting estimation using 1,000,000 emitted photons.

© DigiPen Institute of Technology Europe - Bilbao

Photon tracing

Once a photon is emitted, it is traced through the scene similarly to a ray. When a photon hits a surface, it may be reflected, transmitted, or absorbed, depending on the material properties at the point of interaction.

Only photons that contribute meaningfully to the lighting simulation should continue to be traced. Therefore, photons whose power become negligible should be terminated to avoid wasting computation on insignificant contributions.

Russian Roulette

In the indirect photon map, photons bounce repeatedly between surfaces. To prevent photons from bouncing indefinitely, a probabilistic termination strategy called *Russian Roulette* introduced in [3] is used. Photons with similar power are preferred in the photon map, as this leads to more stable radiance estimates with fewer stored photons.

Let E be the event that the photon survives a bounce. The survival probability $P(E)$ can be determined in various ways. In this implementation, it is based on the reflective properties of the material, specifically the maximum RGB component of the material color ρ.

$$P(E) = \max(\rho_r, \rho_g, \rho_b)$$

To determine whether a photon path is terminated:

1. Generate a random number $r \in [0, 1]$.
2. If $r > P(E)$, the photon terminates.
3. If $r \leq P(E)$, the photon survives and its power is scaled.

The scaled photon power is computed as:

$$\Delta\Phi'_p = \frac{\Delta\Phi_p}{P(E)}$$

where:

- $\Delta\Phi_p$: The photon's original power.
- $P(E)$: The survival probability.

This method reduces the number of photons stored in the photon map, lowering memory usage and speeding up both construction and nearest-neighbor queries. Although it increases variance, the solution converges to the correct result as enough photons are used.

© DigiPen Institute of Technology Europe - Bilbao

Photon gathering

In the second pass, rays are traced from the camera. For each point intersected by a camera ray, the N closest photons are gathered from the photon maps to estimate the radiance.

Alternatively, a fixed radius gathering could be used, but this approach does not adapt to non-uniform photon density distributions.

Radiance estimation

The *Rendering Equation* describes how light leaving a surface point is generated by incoming illumination and the reflective properties of the material.

$$L_r(x, \vec{v}) = \int_{\Omega} f_r(x, \vec{\imath}) L_i(x, \vec{\imath})(\vec{n} \cdot \vec{\imath}) \; d\omega$$

where:

- x: The position on the surface where radiance is being computed.
- $\vec{v}$: The direction in which light leaves the point.
- $\vec{\imath}$: The direction from which light arrives at the point.
- $\vec{n}$: Normal of the surface.
- Ω: The hemisphere above over which incoming light is integrated.
- $d\omega$: An infinitesimally small solid angle.
- L_i: The incoming radiance.
- f_r: The BRDF (Bidirectional Reflectance Distribution Function).

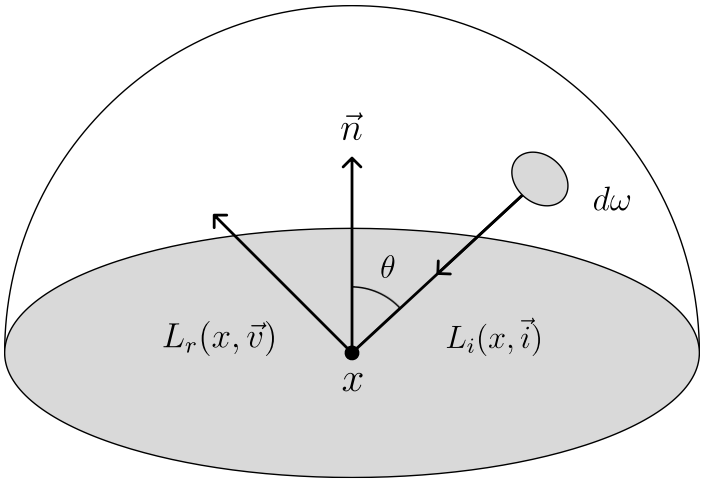


Figure 31: Hemisphere illustrating the rendering equation.

This equation can be split into a sum of several components, each one representing a different type of contribution.

$$L_i = L_{i,d} + L_{i,i} + L_{i,c}$$

where:

© DigiPen Institute of Technology Europe - Bilbao

- $L_{i,d}$: The contribution from direct illumination.
- $L_{i,i}$: The contribution from indirect illumination.
- $L_{i,c}$: The contribution from caustics.

Photon maps store photons that carry a power $\Delta\Phi_p$ and arrive at the surface from all directions. For a small area ΔA around x, the outgoing radiance contribution can be approximated as:

$$L_r(x, \vec{v}) \approx \frac{1}{\Delta A} \sum_{p=1}^{N} f_r(x, \vec{\imath}) \Delta\Phi_p$$

where:

- ΔA: The area around the point used to gather nearby photons. Since photons are gathered based on the distance to the point, this area is taken as a disk of radius r, so $\Delta A = \pi r^2$.
- N: The number of photons gathered within the area.
- $\Delta\Phi_p$: The power of the p-th photon.
- $f_r(x, \vec{\imath})$: The BRDF at the surface.

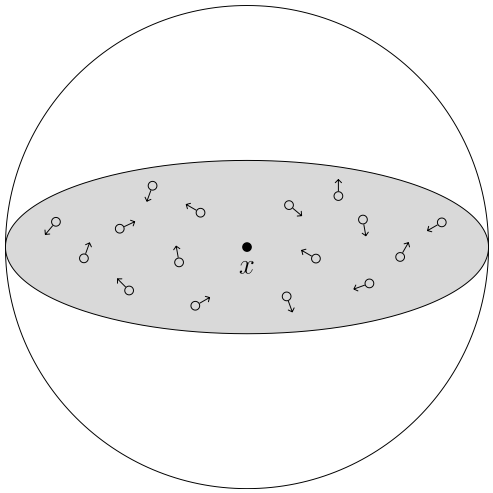


Figure 32: Surface point with nearby photons used for radiance estimation.

© DigiPen Institute of Technology Europe - Bilbao

Lambertian reflectance

In this implementation of *Photon Mapping*, radiance is estimated only for diffuse surfaces. Therefore, the bidirectional reflectance distribution function can be modeled using the Lambertian BRDF, which depends on the surface color ρ:

$$f_r(x, \vec{\imath}) = \frac{\rho}{\pi}$$

The Lambertian model assumes uniform scattering of light in all directions.

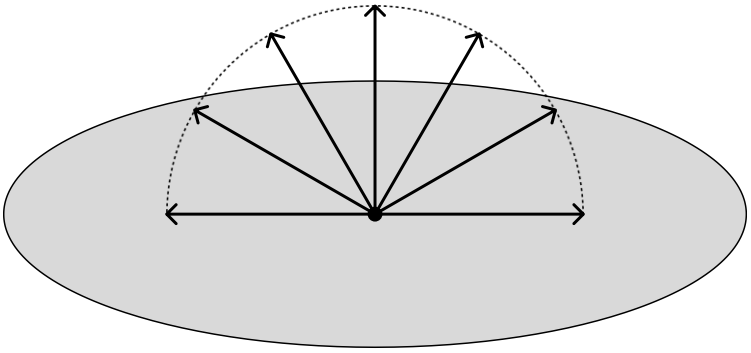


Figure 33: Lambertian surface scattering.

Cone filter

In situations where the density of photons is low, the radiance estimation can produce blurry results. To improve the quality, a weight can be given to each nearby photon based on its distance to the surface point:

$$w_p = \max(0, 1 - \frac{d_p}{kr})$$

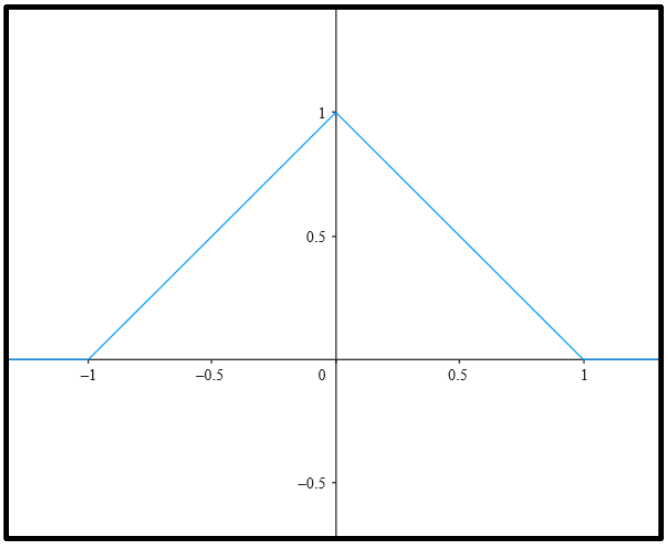


Figure 34: Cone filter values of r = 1 and k = 1.

© DigiPen Institute of Technology Europe - Bilbao

where:

- d_p: The distance between the surface point x and the photon.
- r: The search radius used to gather photons.
- k: A filter constant that controls how sharply the weight falls off with distance.

Although photons are located in a sphere for searching purposes, they are actually stored on 2D surfaces. Therefore, the normalization of the filter is based on a 2D distribution:

$$C = \frac{1}{1 - \frac{2}{3}k}$$

The radiance estimate becomes:

$$L_r(x, \vec{\omega}) \approx \frac{C}{\Delta A} \sum_{p=1}^{N} f_r(x, \vec{i}) \Delta\Phi_p w_p$$

The following images show the output of a test scene generating a cardioid caustic. The comparison illustrates the effect of the cone filter.

The output with the cone filter enabled is sharper and more defined, while the one without the filter appears blurrier.

Output:

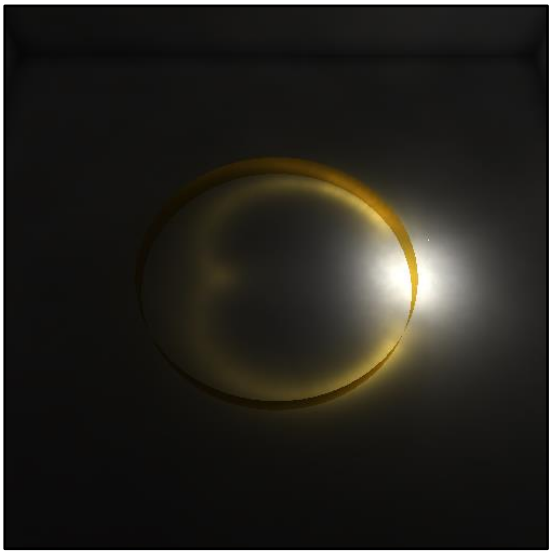


Figure 35: A cardioid caustic without cone filter.

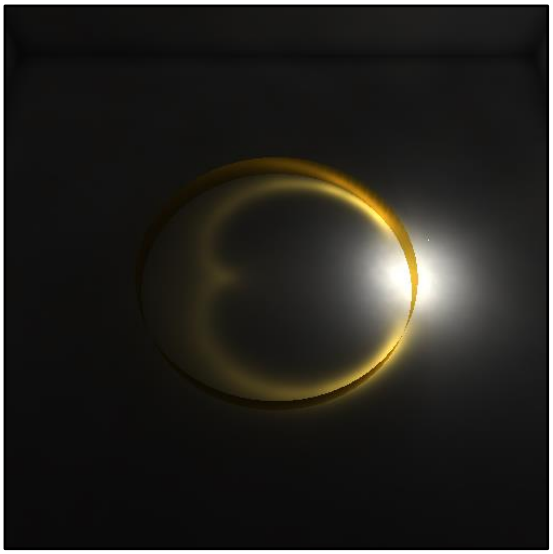


Figure 36: A cardioid caustic with cone filter enabled with k = 1.

© DigiPen Institute of Technology Europe - Bilbao

Gathering

The following images show the output of a test scene using different numbers of gathered photons.

In all cases, the total number of emitted photons was *10 million*, allowing a direct comparison of how the number of gathered photons affects the quality and smoothness of the image.

Output:

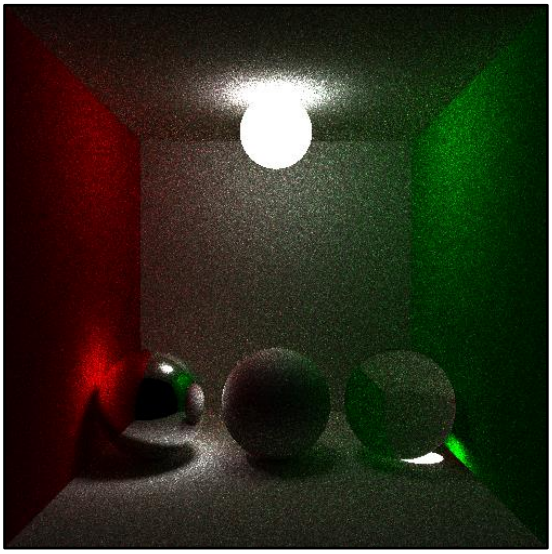


Figure 37: Lighting estimation using 10 nearest photons.

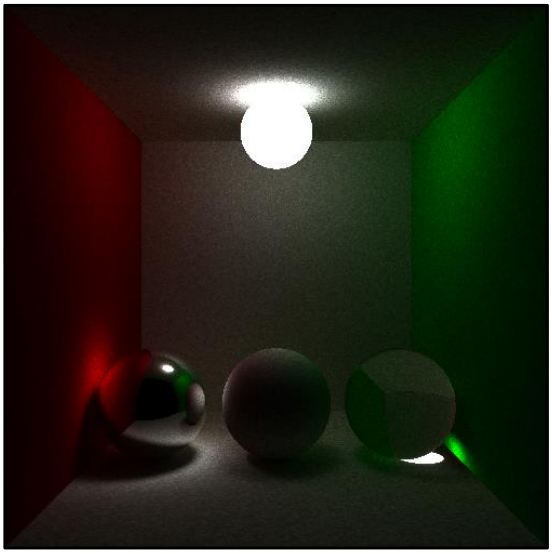


Figure 38: Lighting estimation using 100 nearest photons.

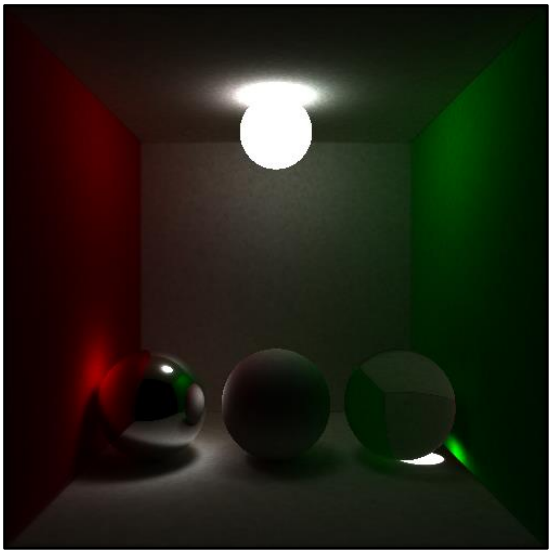


Figure 39: Lighting estimation using 1,000 nearest photons.

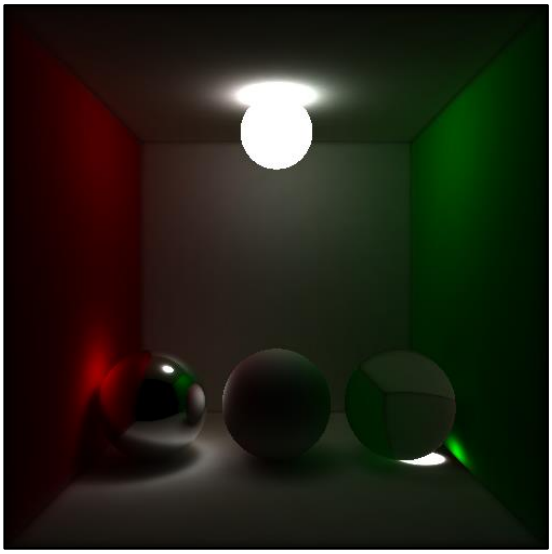


Figure 40: Lighting estimation using 10,000 nearest photons.

© DigiPen Institute of Technology Europe - Bilbao

Testing Scenes

All the output images shown previously were generated using the methods described in this report. In this section, a set of prepared scenes is examined, giving special attention to highlighting the quality of caustics, as reproducing accurate caustics is the primary focus of this project.

Complex Scene

Description:

A box filled with water containing multiple objects, including a metallic object and a diffuse object, is illuminated from above.

Purpose:

To demonstrate caustic formation in a complex environment, highlighting both refractive caustics from the water and reflective caustics from the metallic surface.

Observations:

- Well defined caustics generated by the wavy water surface are visible on the diffuse object and on the bottom of the box.
- The metallic object produces reflective caustics on the surrounding surfaces.

Output:

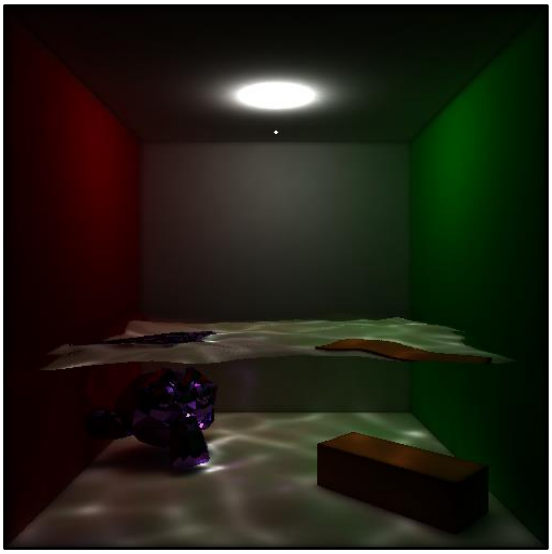


Figure 41: Output of scene containing water surface.

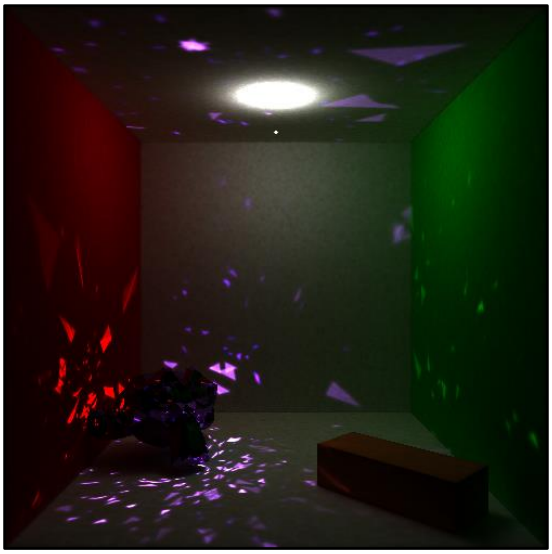


Figure 42: Output of scene containing metallic object.

© DigiPen Institute of Technology Europe - Bilbao

Glass Shapes

Description:

Transparent glass shapes are illuminated to produce caustics on the surfaces beneath them.

Purpose:

To observe the effect of geometry and material properties on the caustics.

Observations:

- Refractions bend light differently depending on the shape.
- Reflections from the glass surfaces contribute subtle highlights.
- Variations in *index of refraction* affect the sharpness and intensity of caustics.

Output:

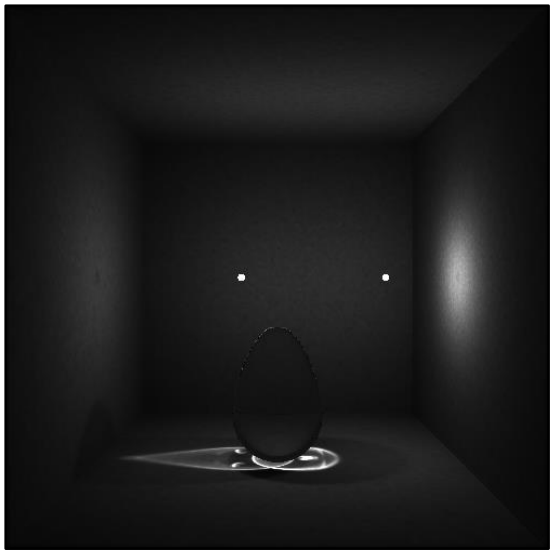


Figure 43: Output of egg-shaped object with IOR = 1.1.

Figure 44: Output of egg-shaped object with IOR = 1.6.

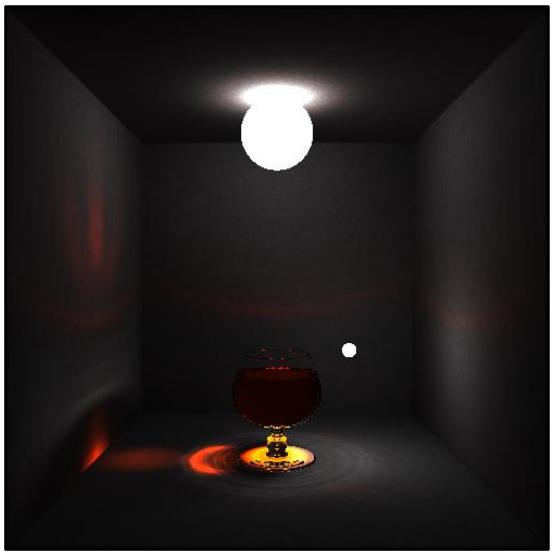


Figure 45: Output of cognac glass.

© DigiPen Institute of Technology Europe - Bilbao

Stanford Dragons

Description:

Two *Stanford Dragon* models are illuminated to generate caustics. One dragon has a metallic material, while the other is refractive.

The scene is rendered with two different point light sizes to compare their effects on caustic formation.

Purpose:

To demonstrate how light source size influences caustic sharpness and definition on complex geometry.

Observations:

- Small spherical light: Produces sharper, more concentrated caustics, highlighting details on both metallic and refractive surfaces.
- Large spherical light: Produces softer, more diffused caustics, reducing the intensity of caustic details.

Output:

Figure 46: Output of Stanford Dragons scene with a large spherical light.

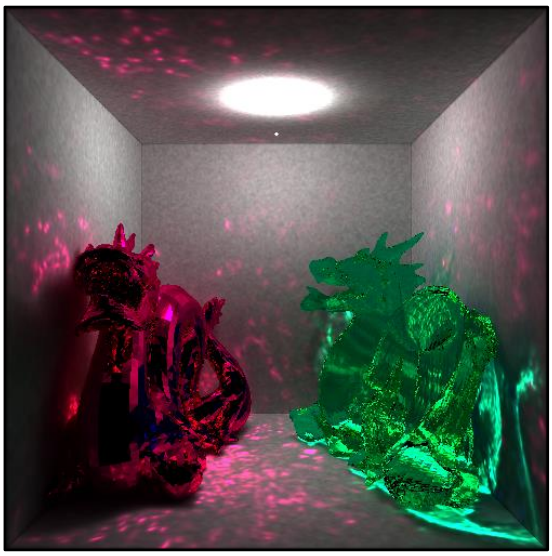


Figure 47: Output of Stanford Dragons scene with a small spherical light.

© DigiPen Institute of Technology Europe - Bilbao

Limitations

Photon Mapping is a powerful technique, but it has several limitations. These problems arise from the discrete nature of photon sampling, data structure requirements, and the handling of complex material interactions.

High Memory Consumption

Storing a large number of photons in a k-d tree can consume significant memory, especially when millions of photons are required for smooth results.

While more compact k-d tree implementations for reducing the per-node data can help mitigate memory usage, high memory consumption remains an unavoidable limitation of *Photon Mapping* for very dense photon maps.

Bias in radiance estimate

The radiance estimated from *Photon Mapping* is biased due to the discrete sampling of photons. This can lead to blotchy artifacts, where some regions appear unnaturally brighter or darker than their true radiance.

Furthermore, photons are collected based on spatial proximity, but not all nearby photons are necessarily relevant to the surface point being shaded. For example, photons originating from objects outside the intended area may be included in the gathering, producing incorrect light contributions.

Figure 48: Demonstration of bias.

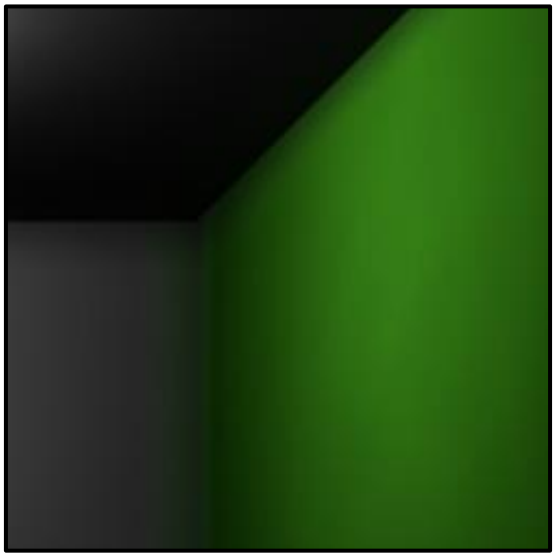


Figure 49: Demonstration of incorrect light contribution for direct illumination.

© DigiPen Institute of Technology Europe - Bilbao

Sampling requirements for complex materials

For diffuse surfaces, multiple samples per pixel are not necessary, since the photons stored in the photon map already incorporate the effects of the material's BRDF.

However, for reflective or refractive surfaces, rays are traced using *Monte Carlo* techniques until they reach a diffuse surface. In the case of rough metals or dielectric materials exhibiting Fresnel effects, multiple samples per pixel are required to capture the combined reflection and refraction accurately, increasing computational cost.

Multiple samples are only needed for rays that do not hit a diffuse surface at first.

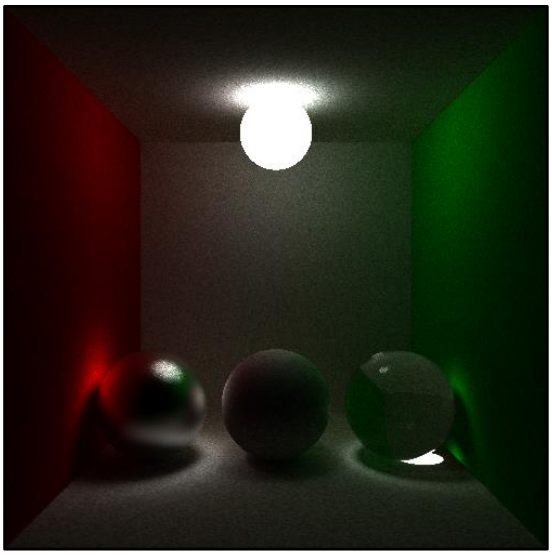


Figure 50: A rough metallic sphere and a dielectric sphere, where photons interact with the surfaces including Fresnel reflection and refraction, requiring multiple samples to capture variations.

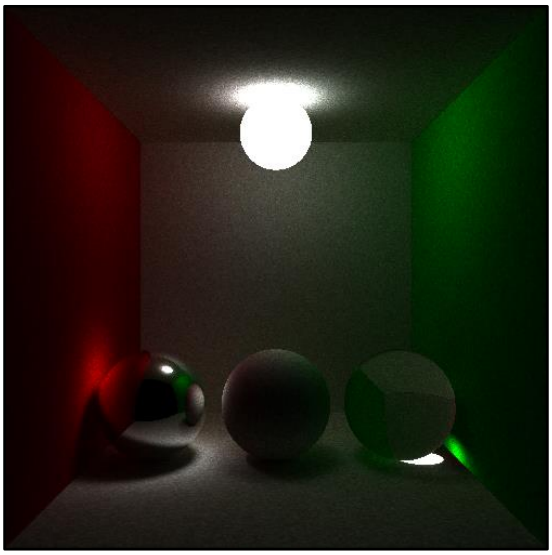


Figure 51: A mirror-like metallic sphere and a refractive sphere without Fresnel, so photons follow a deterministic path per bounce.

© DigiPen Institute of Technology Europe - Bilbao

Alternative Approaches

Global illumination and caustics can be simulated using other rendering techniques different from *Photon Mapping*. One method is *Monte Carlo* ray tracing, which simulates light transport by randomly sampling possible light paths.

The following sections present a comparison of scenes rendered with these two methods, highlighting their respective strengths and limitations.

Photon Mapping vs Monte Carlo Ray Tracing

The scenes used for the comparison were carefully prepared to allow both rendering methods to perform reasonably, to ensure a fair comparison.

Output:

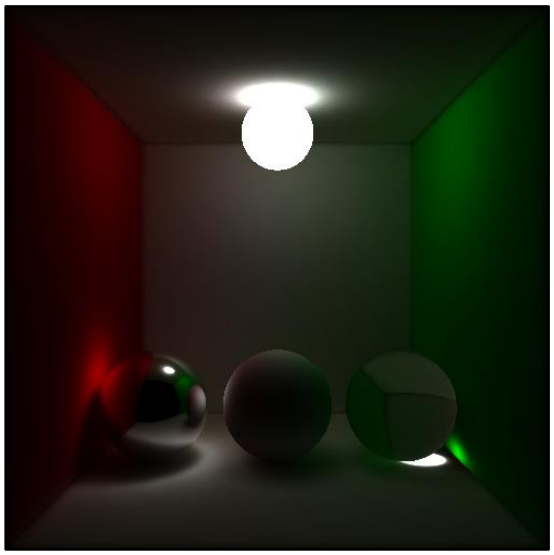


Figure 52: Rendered using Photon Mapping.

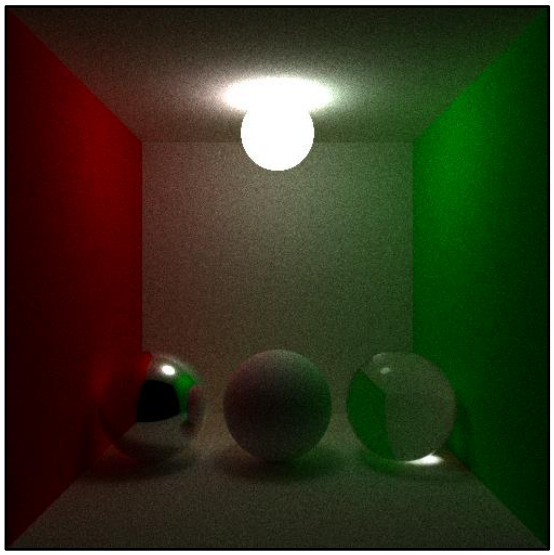


Figure 53: Rendered using Monte Carlo ray tracing.

Figure 54: Rendered using Photon Mapping.

Figure 55: Rendered using Monte Carlo ray tracing.

© DigiPen Institute of Technology Europe - Bilbao

Criterion	Photon Mapping	Monte Carlo Ray Tracing
Global Illumination	- Requires many photons for smooth results. - Slight bias due to discretization.	- Requires many samples for smooth results. - Unbiased light transport.
Caustics	- Sharp, defined and concentrated.	- Often noisy or attenuated.
Efficiency	- High memory usage. - Very efficient for caustics.	- No heavy memory requirements. - Slow convergence.
Noise / Artifacts	- Blotches or uneven regions. - Artifacts if low photon density.	- Stochastic noise. - Noise if low sample count.

Conclusion

Photon Mapping provides better caustic definition and efficiency at the cost of bias. *Monte Carlo* ray tracing offers unbiased global illumination, but achieving comparable caustic quality requires much higher computational effort.

© DigiPen Institute of Technology Europe - Bilbao

Conclusions

Photon Mapping proves to be a powerful and efficient technique for rendering caustics, producing sharp, well-defined light patterns even with complex reflective and refractive surfaces. Its ability to precompute photon interactions allows for accurate radiance estimation on diffuse surfaces with minimal per-pixel sampling.

However, for full global illumination, *Photon Mapping* becomes less efficient. Capturing direct illumination requires emitting a large number of photons, which increases memory usage for storing photon maps, and gathering many photons per shading point can lead to longer rendering times.

To overcome these limitations, hybrid approaches that combine *Photon Mapping* with other global illumination techniques can be explored. Such strategies aim to retain the high-quality caustics provided by *Photon Mapping* while improving efficiency and reducing memory requirements for complete light transport simulation.

© DigiPen Institute of Technology Europe - Bilbao

How to use the Demo

The demo is used in the same way as the ray tracing assignments.

In the configuration file the RENDERMODE should be changed to PHOTONMAP.

Adjust the *Photon Mapping* related parameters as needed.

The configurable parameters are:

- DIRECT_PHOTONS_PER_LIGHT: Number of direct photons emitted per light.

- DIRECT_NEARBY_PHOTONS: Number of nearby direct photons used during gather.

- INDIRECT_PHOTONS_PER_LIGHT: Number of indirect photons emitted per light.

- INDIRECT_NEARBY_PHOTONS: Number of nearby indirect photons used during gather.

- CAUSTIC_PHOTONS_PER_LIGHT: Number of caustic photons emitted per light.

- CAUSTIC_NEARBY_PHOTONS: Number of nearby caustic photons used during gather.

- MULTIPLE_SAMPLES: If true, full statistical sampling is used (roughness, Fresnel, etc.), if false, these effects are simplified or ignored.

- MAX_PHOTON_BOUNCES: Max bounces a photon can do.

- DIRECT_CONE_FILTER_PARAM: Cone filter parameter for direct photons (if < 1.0f, disabled).

- INDIRECT_CONE_FILTER_PARAM: Cone filter parameter for indirect photons (if < 1.0f, disabled).

- CAUSTICS_CONE_FILTER_PARAM: Cone filter parameter for caustic photons (if < 1.0f, disabled).

For more details, refer to the project README.txt.

© DigiPen Institute of Technology Europe - Bilbao

Bibliography

1. Jensen, H. W. (1996). Global illumination using photon maps. In Proceedings of the 7th Eurographics Workshop on Rendering (pp. 21–30). Eurographics Association.

2. Christensen, N. J., & Jensen, H. W. (2000). A Practical Guide to Global Illumination Using Photon Maps. SIGGRAPH 2000 Course 8, July 23, 2000.

3. Arvo, J., & Kirk, D. (1990). Particle transport and image synthesis. Computer Graphics, 24(4), 53–66.

© DigiPen Institute of Technology Europe - Bilbao