Lab #5 – Photon mapping (part 3)

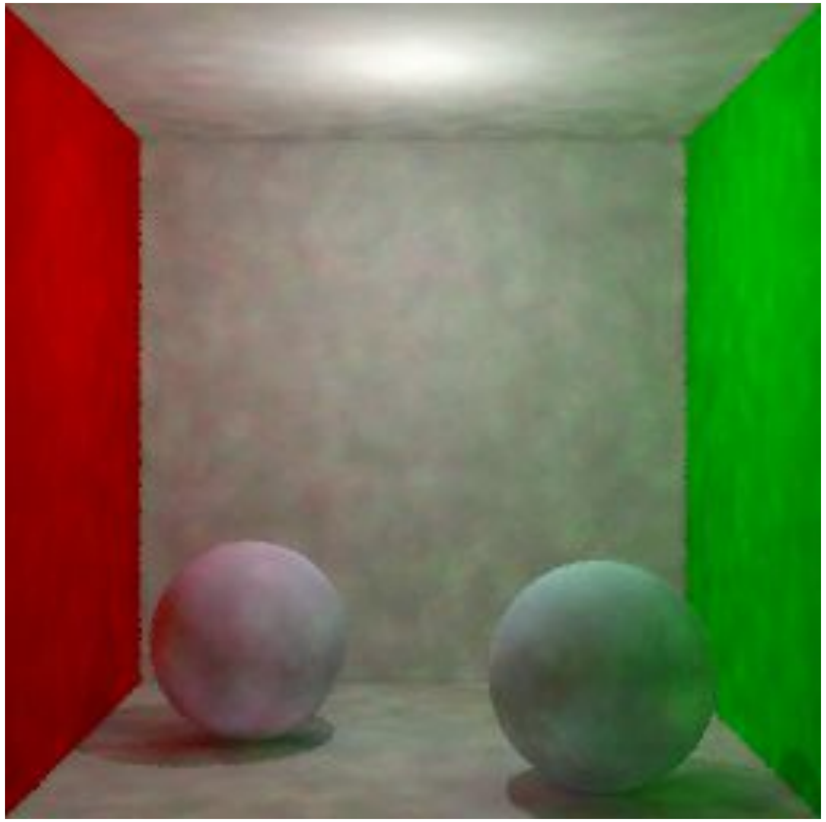
Informática Gráfica

Adolfo Muñoz - Julio Marco
Pablo Luesia - J. Daniel Subías – Óscar Pueyo

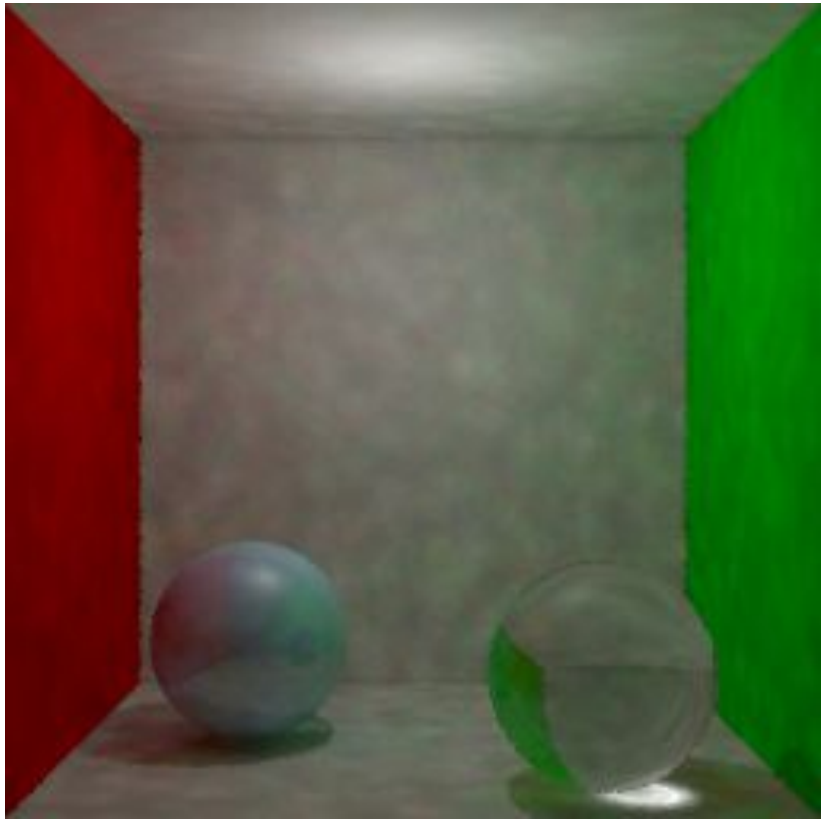
Graphics and Imaging Lab

Before we begin…

- Today: add specular and refractive materials to your photon mapper



Only diffuse



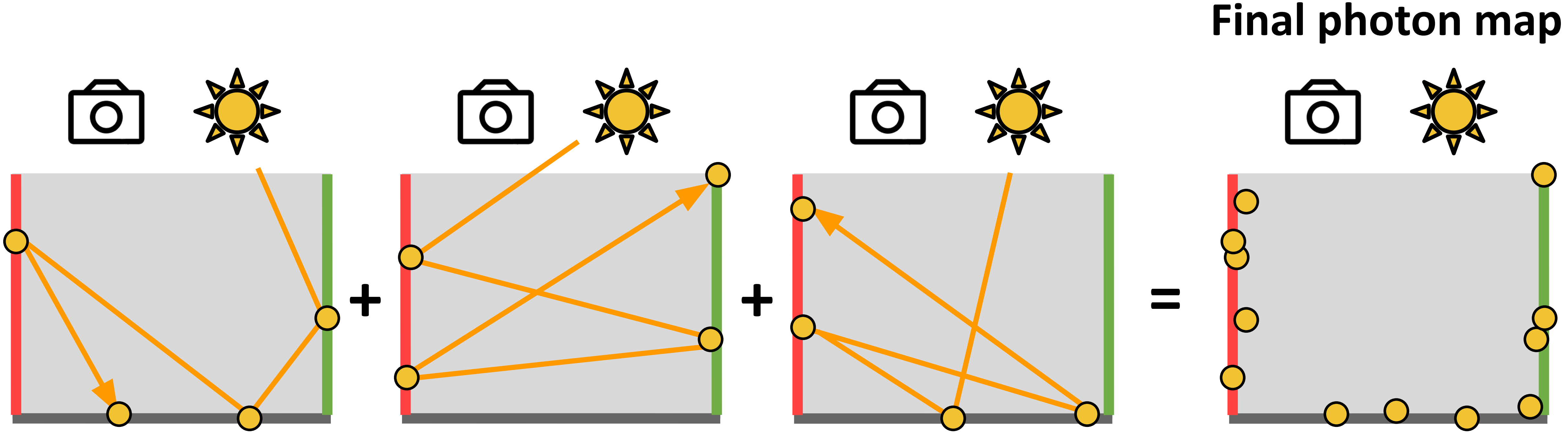
\+ specular and refraction

Before we begin…

- Lab 5 (photon mapping) **is the second submitted work**
  - Recommended deadline: December 4th
  - Moodle: January 11th
- You can probably **reuse most of your code** for this assignment
- Remember: Final work is 80% of the final grade

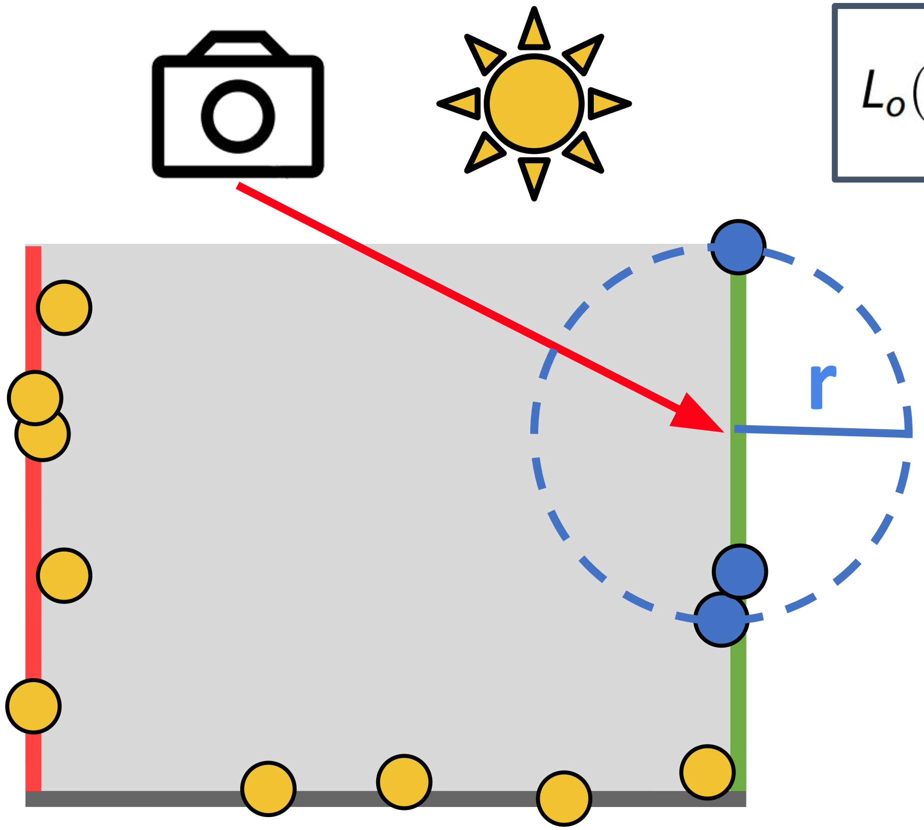
Recap: photon map construction

- Split a path into two:
  - (1) lights write into the photon map
  - (2) camera reads from the photon map
- Photons are sent out from the light sources (**photon random walks**)



Recap: how the photon map is used

- Split a path into two:
  - (1) lights write into the photon map
  - (2) camera reads from the photon map



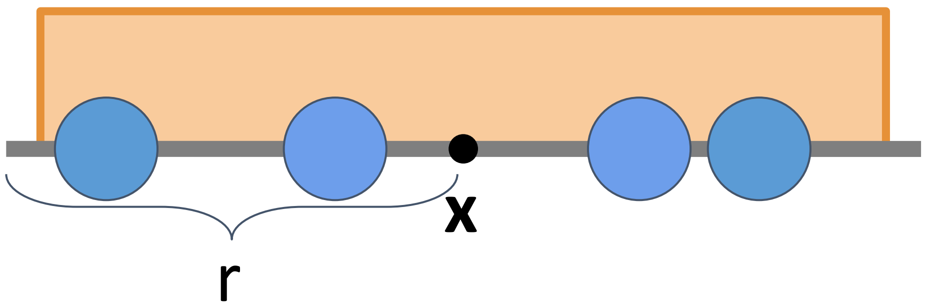
$$L_o(\mathbf{x},\omega_\mathbf{o}) = L_e(\mathbf{x},\omega_\mathbf{o}) + \boxed{\int_\Omega L_i(\mathbf{x},\omega_\mathbf{i}) f_r(\mathbf{x},\omega_\mathbf{i},\omega_\mathbf{o}) |\mathbf{n}\cdot\omega_\mathbf{i}| d\omega_\mathbf{i}}$$

- Idea: use **k nearby** photons (distance < r) to approximate the Render Equation
- The result is the **density estimation** of the three photons

Recap: photon density estimation

Constant density estimation

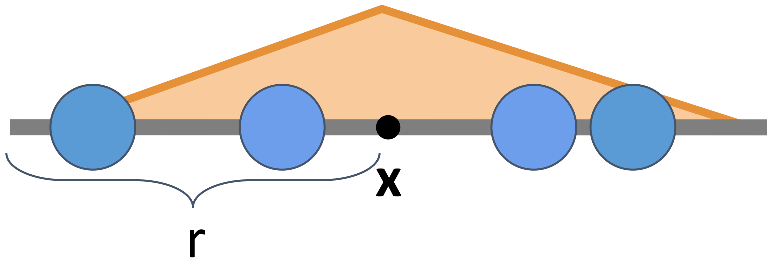
Box kernel



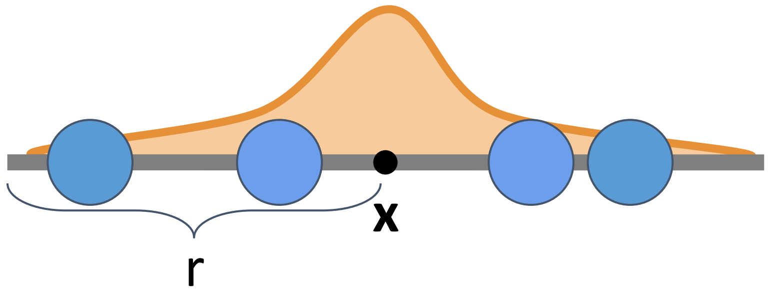
$$L_o(\mathbf{x}, \omega_\mathbf{o}) \approx \sum_{p=1}^{k} f_r(\mathbf{x}, \omega_\mathbf{p}, \omega_\mathbf{o}) \frac{\Phi_p}{\pi r_k^2}$$

Non-constant density estimation

Cone kernel



Gaussian kernel

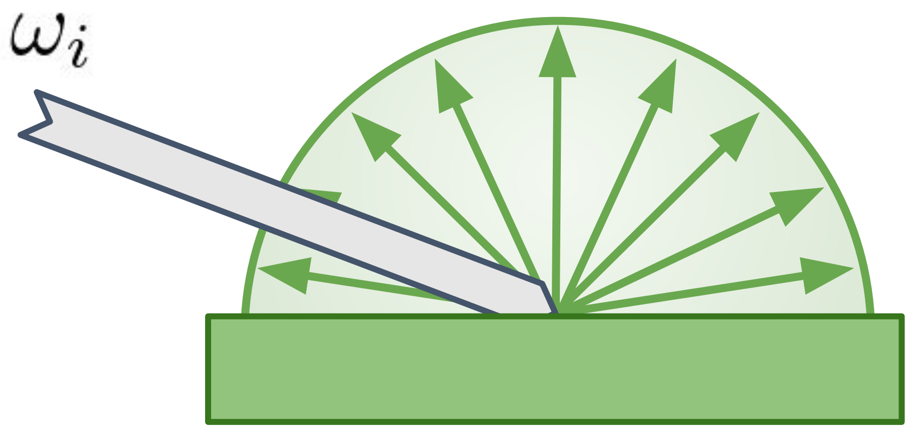


$$L_o(\mathbf{x}, \omega_\mathbf{o}) \approx \sum_{p=1}^{k} f_r(\mathbf{x}, \omega_\mathbf{p}, \omega_\mathbf{o}) \Phi_p K_{2D}(|\mathbf{x} - \mathbf{x_p}|, r_k)$$

Recap: Diffuse and specular BRDFs

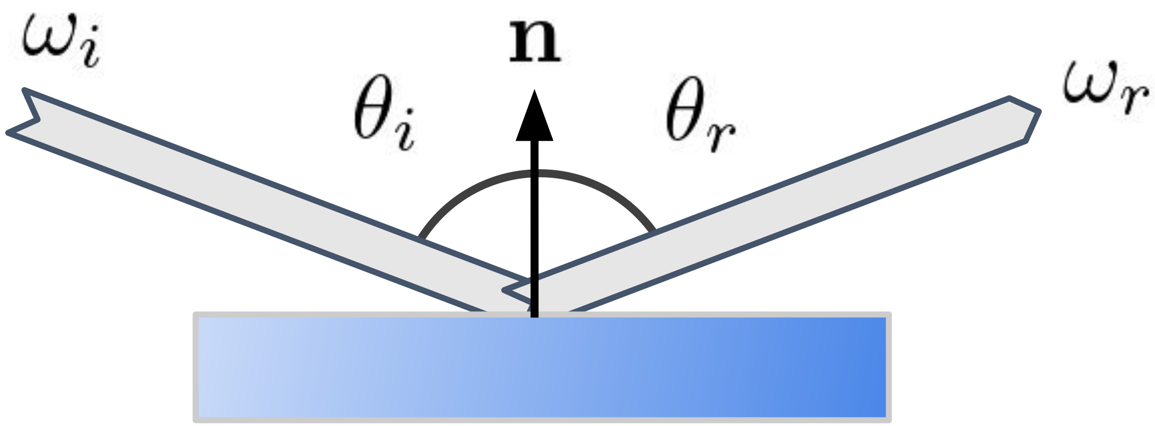
Diffuse material

Light is reflected in all directions equally



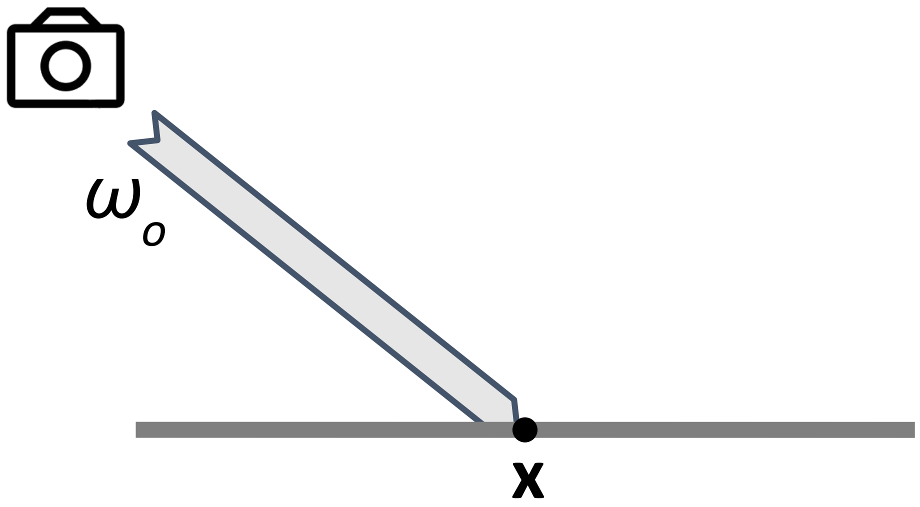
Perfect specular material

All light is (perfectly) reflected towards one direction ω_r



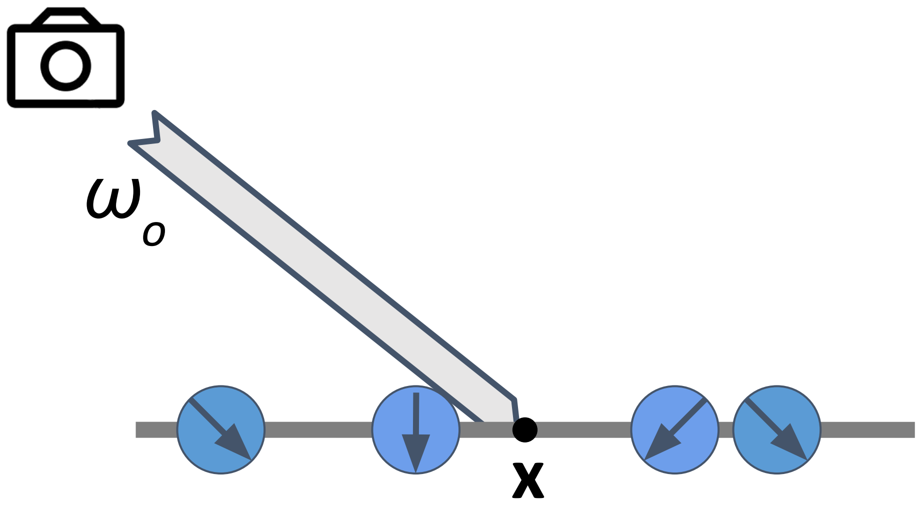
$$\theta_i = \theta_r$$

Only store photons in diffuse surfaces



A ray with direction ω_o intersects in $\mathbf{x}$

Only store photons in diffuse surfaces

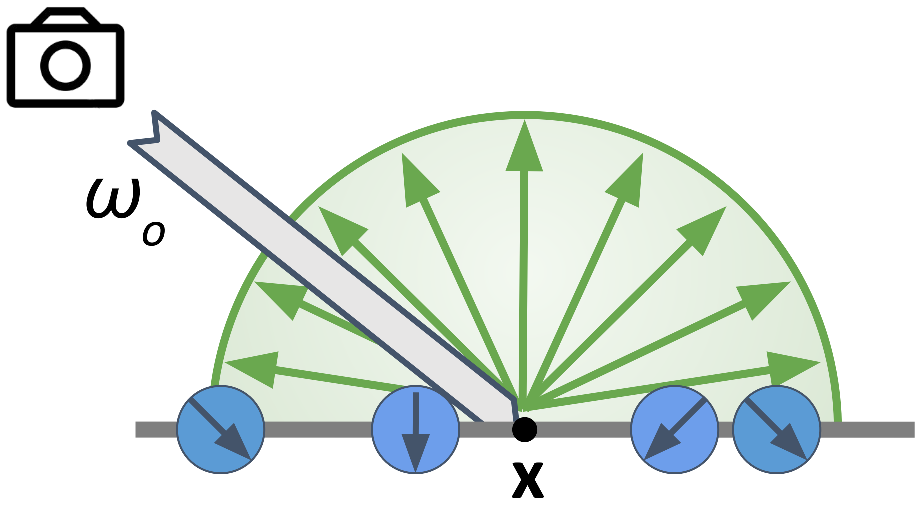


A ray with direction ω_o intersects in $\mathbf{x}$

We obtain a set of near photons, each with:

- Φ_p flux
- ω_p incoming direction
- $\mathbf{x}_p$ position of the photon

Only store photons in diffuse surfaces



Diffuse material

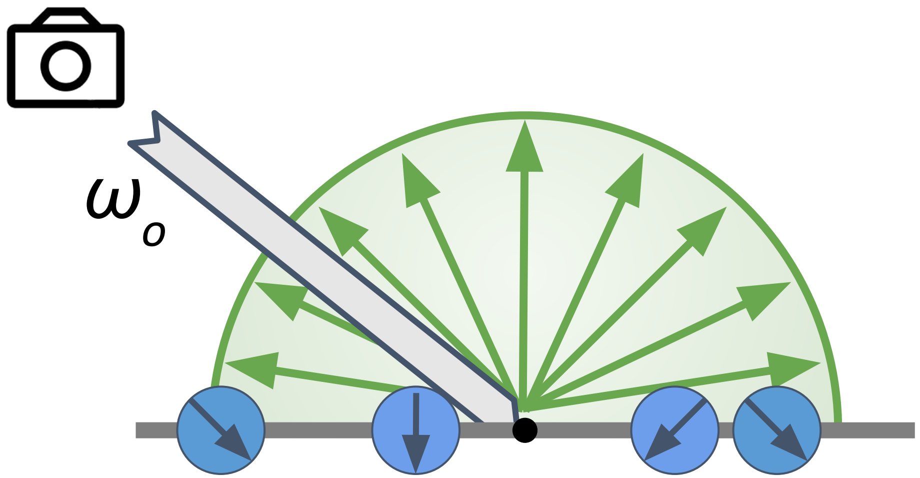
$$L_o(\mathbf{x}, \omega_\mathbf{o}) \approx \sum_{p=1}^{k} f_r(\mathbf{x}, \omega_\mathbf{p}, \omega_\mathbf{o}) \frac{\Phi_p}{\pi r_k^2}$$

A ray with direction ω_o intersects in **x**

We obtain a set of near photons, each with:

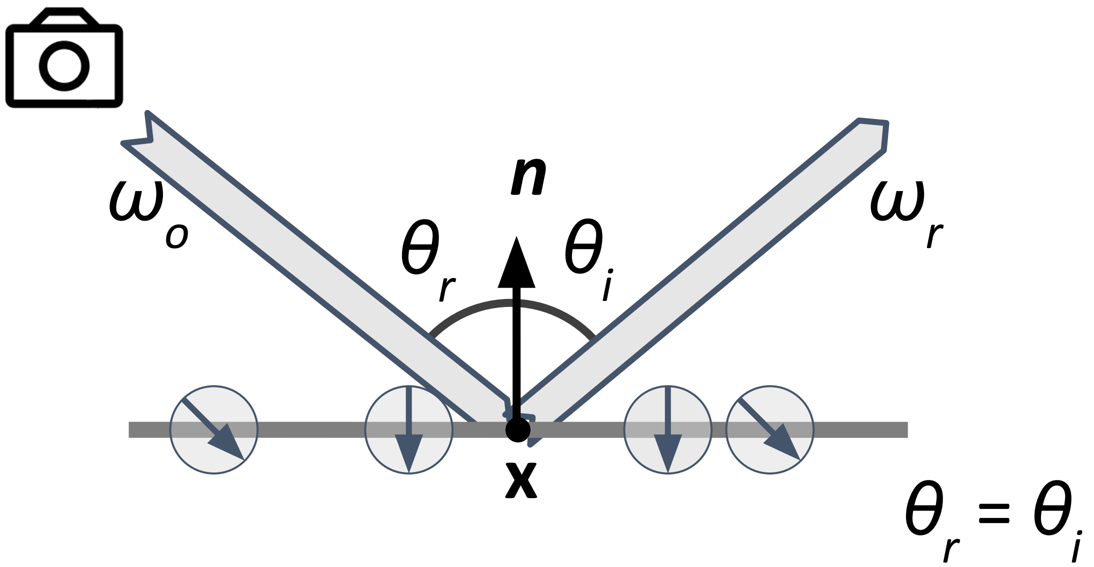
- Φ_p flux
- ω_p incoming direction
- $\mathbf{x}_p$ position of the photon

Only store photons in diffuse surfaces



Diffuse material

$$L_o(\mathbf{x}, \omega_\mathbf{o}) \approx \sum_{p=1}^{k} f_r(\mathbf{x}, \omega_\mathbf{p}, \omega_\mathbf{o}) \frac{\Phi_p}{\pi r_k^2}$$

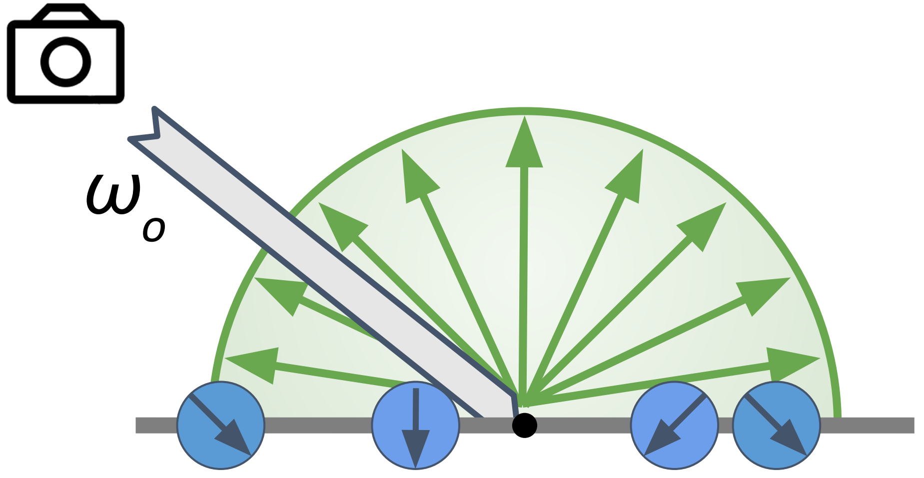


Specular material

$$L_o(\mathbf{x}, \omega_\mathbf{o}) \approx \sum_{p=1}^{k} f_r(\mathbf{x}, \omega_\mathbf{p}, \omega_\mathbf{o}) \frac{\Phi_p}{\pi r_k^2}$$

Only store photons in diffuse surfaces

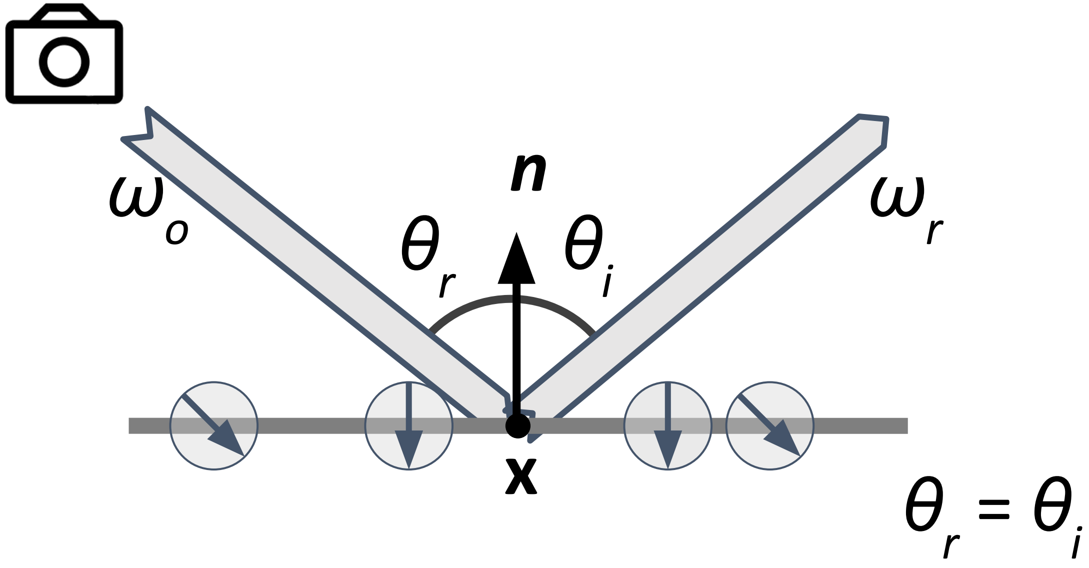
Graphics and Imaging Lab



Diffuse material

$$L_o(\mathbf{x},\omega_o) \approx \sum_{p=1}^{k} \boxed{f_r(\mathbf{x},\omega_\mathbf{p},\omega_\mathbf{o})} \frac{\Phi_p}{\pi r_k^2}$$

$f_r = Kd / \pi \rightarrow L_o > 0$



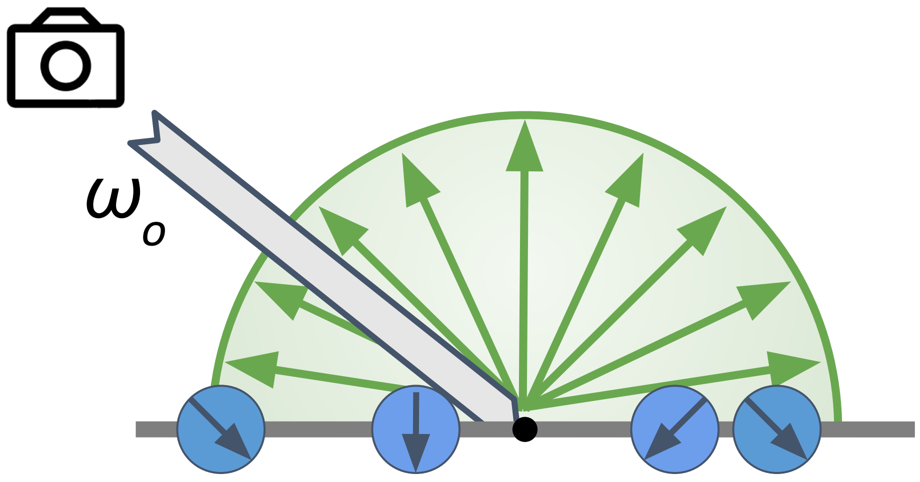
Specular material

$$L_o(\mathbf{x},\omega_o) \approx \sum_{p=1}^{k} \boxed{f_r(\mathbf{x},\omega_\mathbf{p},\omega_\mathbf{o})} \frac{\Phi_p}{\pi r_k^2}$$

$p(\omega_p = -\omega_r) = 0 \rightarrow f_r = 0 \rightarrow L_o = 0$

Only store photons in diffuse surfaces

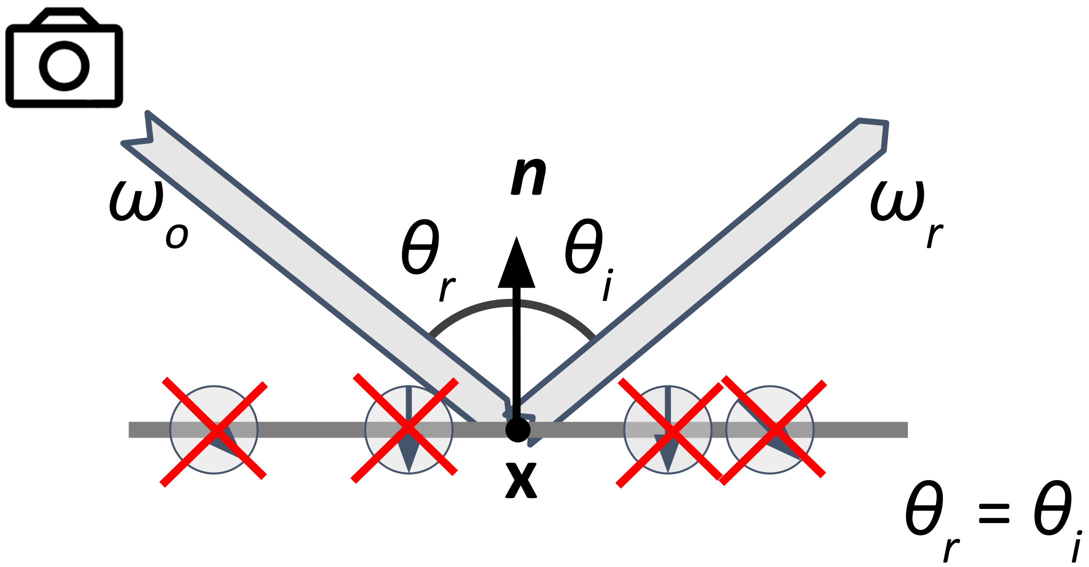
Graphics and Imaging Lab



Diffuse material

$$L_o(\mathbf{x},\omega_\mathbf{o}) \approx \sum_{p=1}^{k} \boxed{f_r(\mathbf{x},\omega_\mathbf{p},\omega_\mathbf{o})} \frac{\Phi_p}{\pi r_k^2}$$

Store photons when intersecting with diffuse surfaces



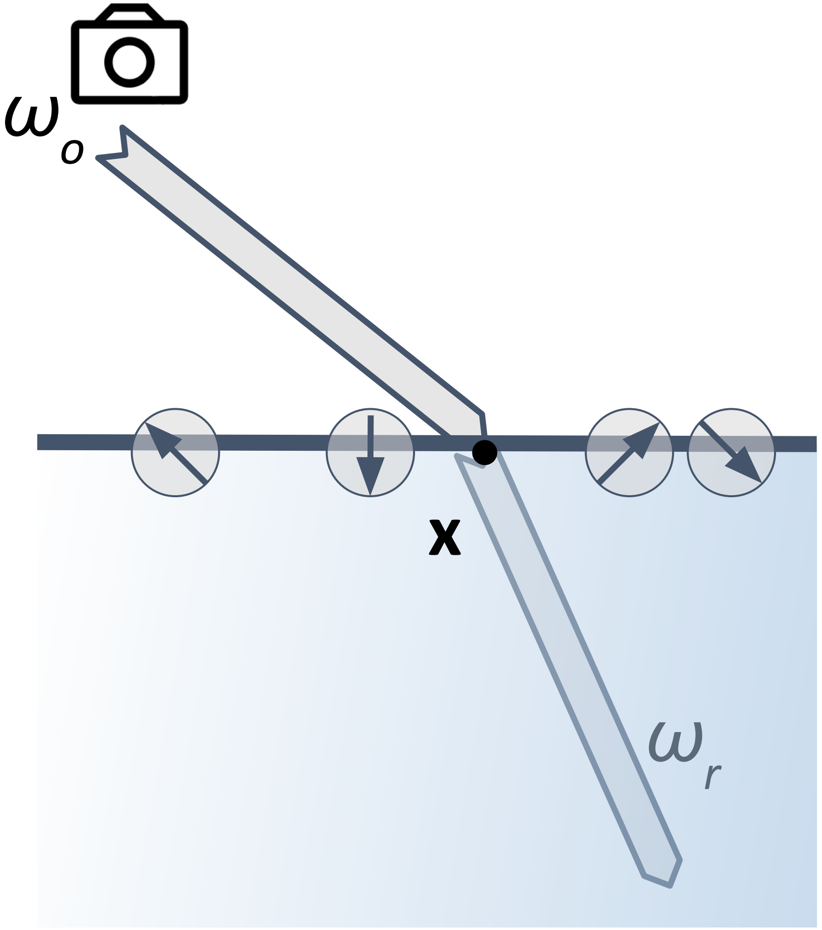
Specular material

$$L_o(\mathbf{x},\omega_\mathbf{o}) \approx \sum_{p=1}^{k} \boxed{f_r(\mathbf{x},\omega_\mathbf{p},\omega_\mathbf{o})} \frac{\Phi_p}{\pi r_k^2}$$

Do NOT store photons when intersecting with specular surfaces

Only store photons in diffuse surfaces

Graphics and Imaging Lab

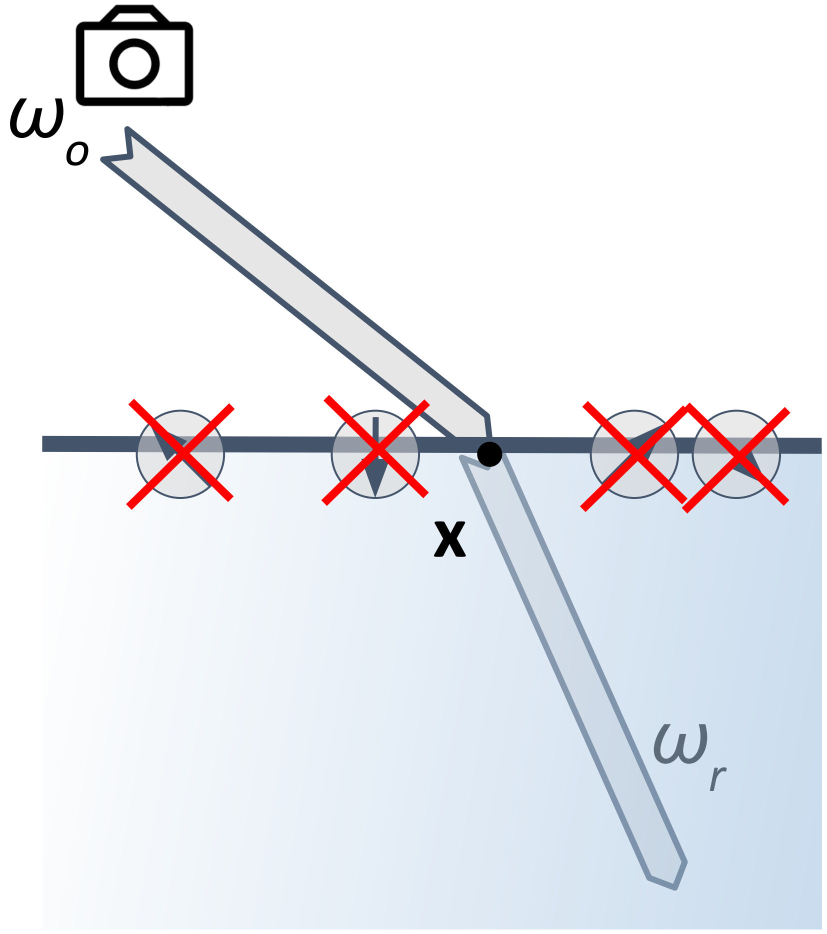


Refractive material

Same happens with refraction

$$p(\omega_p = -\omega_r) = 0 \rightarrow f_r = 0 \rightarrow L_o = 0$$

Only store photons in diffuse surfaces



Refractive material

Same happens with refraction

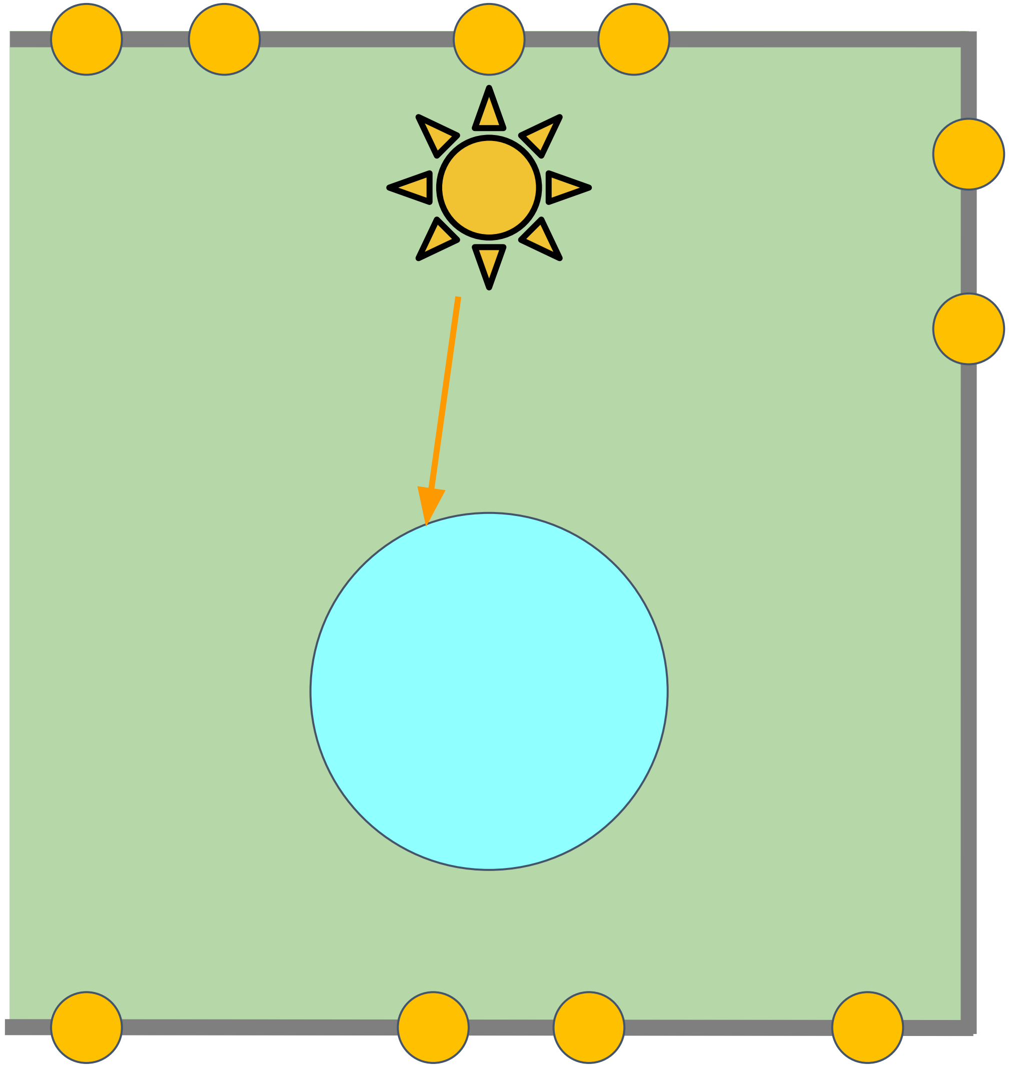
$$p(\omega_p = -\omega_r) = 0 \rightarrow f_r = 0 \rightarrow L_o = 0$$

Do NOT store photons when intersecting with refractive surfaces

What to code in today's session

When a **delta BSDF** is hit:

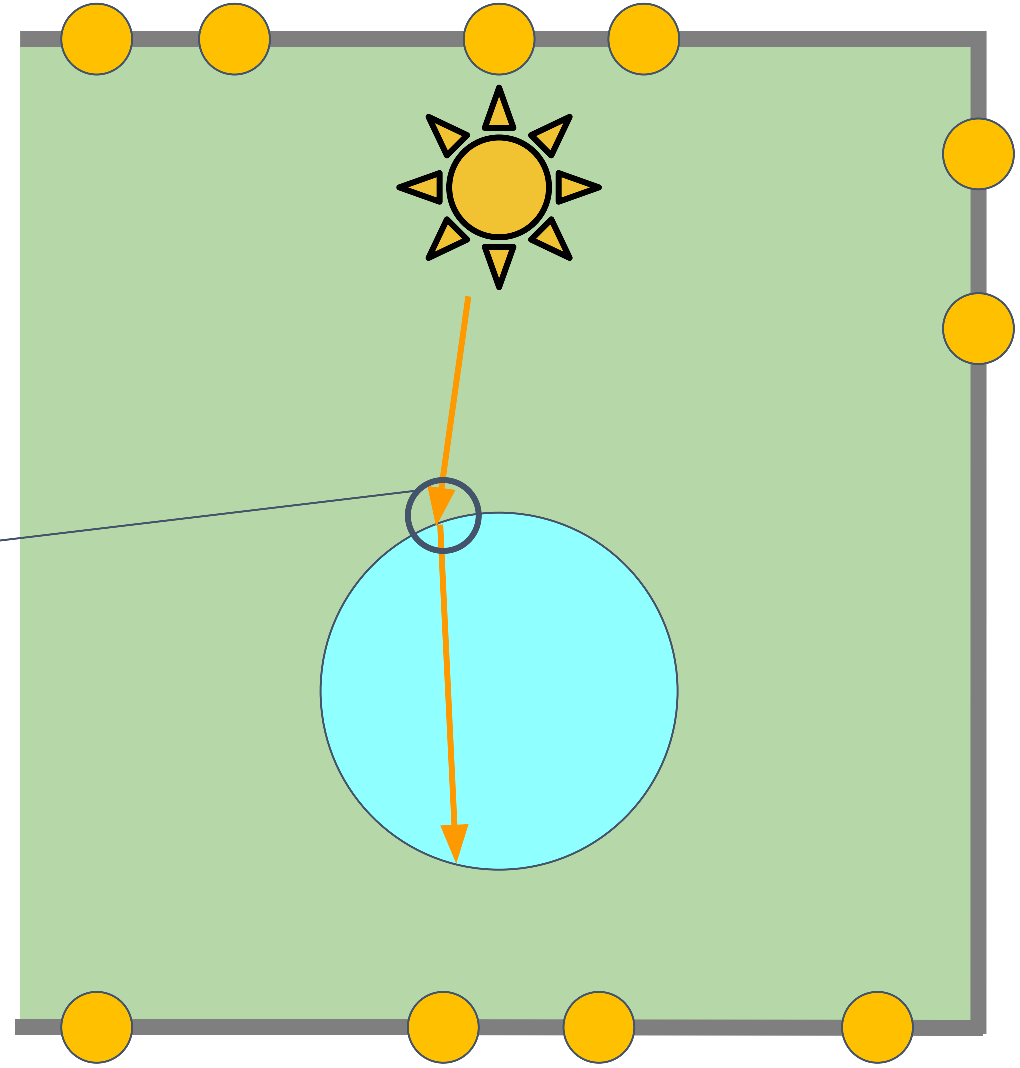
1. Photon map generation: continue the photon walk **without storing a photon**



What to code in today's session

When a **delta BSDF** is hit:

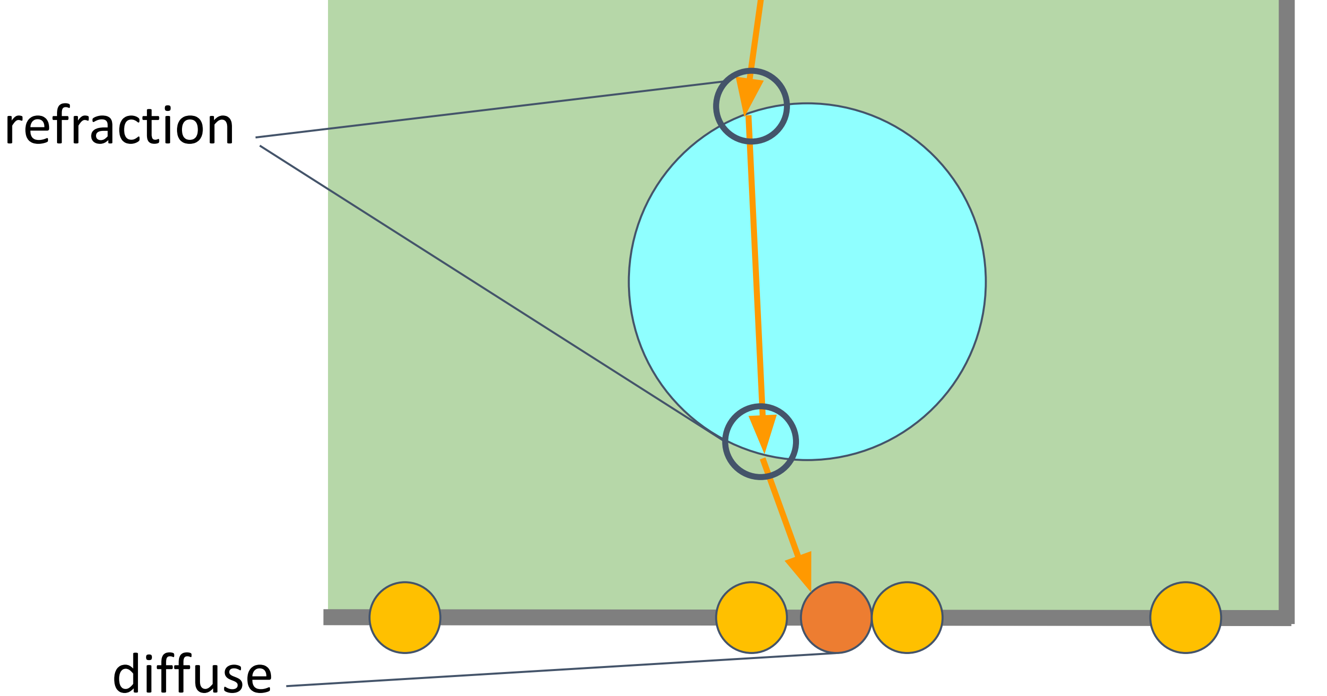
1. Photon map generation: continue the photon walk **without storing a photon**



What to code in today's session

When a **delta BSDF** is hit:

1. Photon map generation: continue the photon walk **without storing a photon**

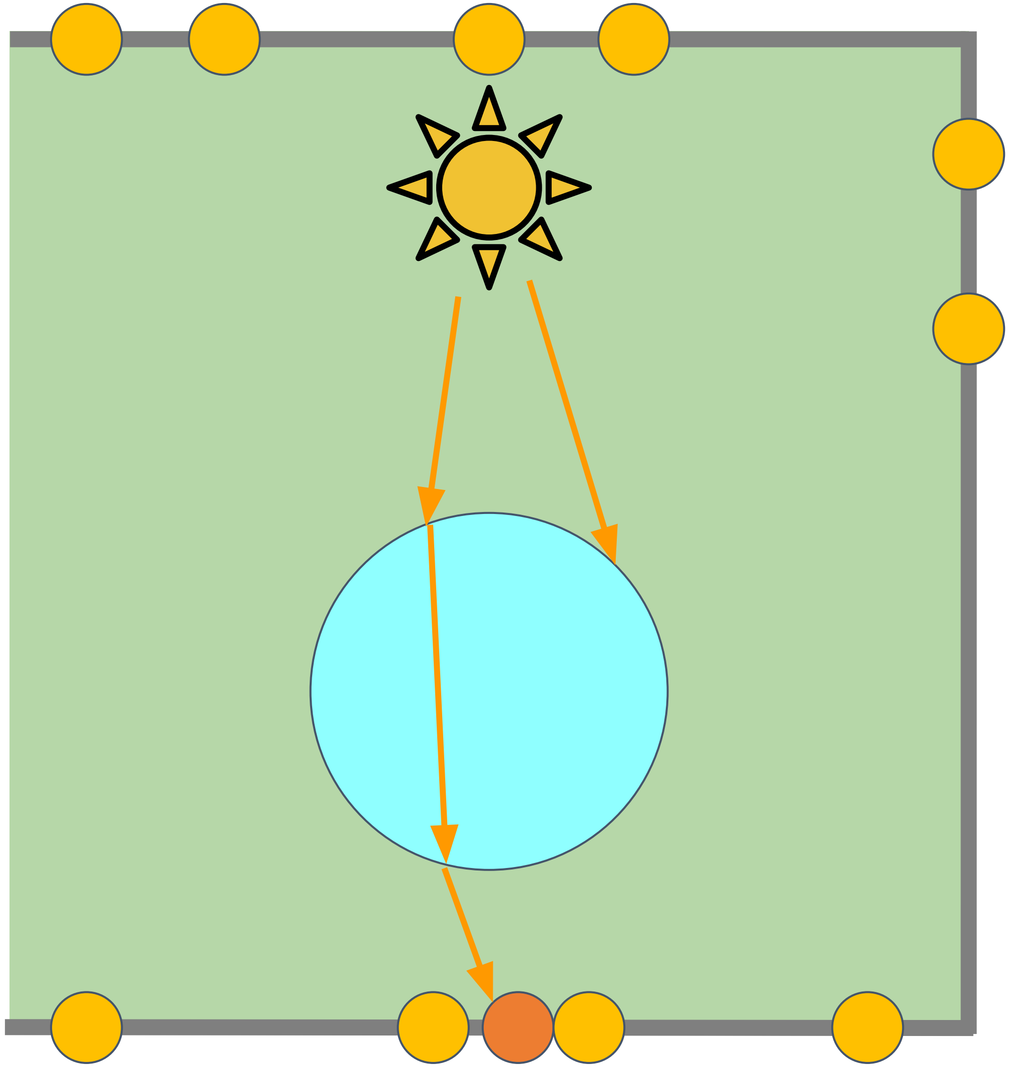


What to code in today's session

Graphics and Imaging Lab

When a **delta BSDF** is hit:

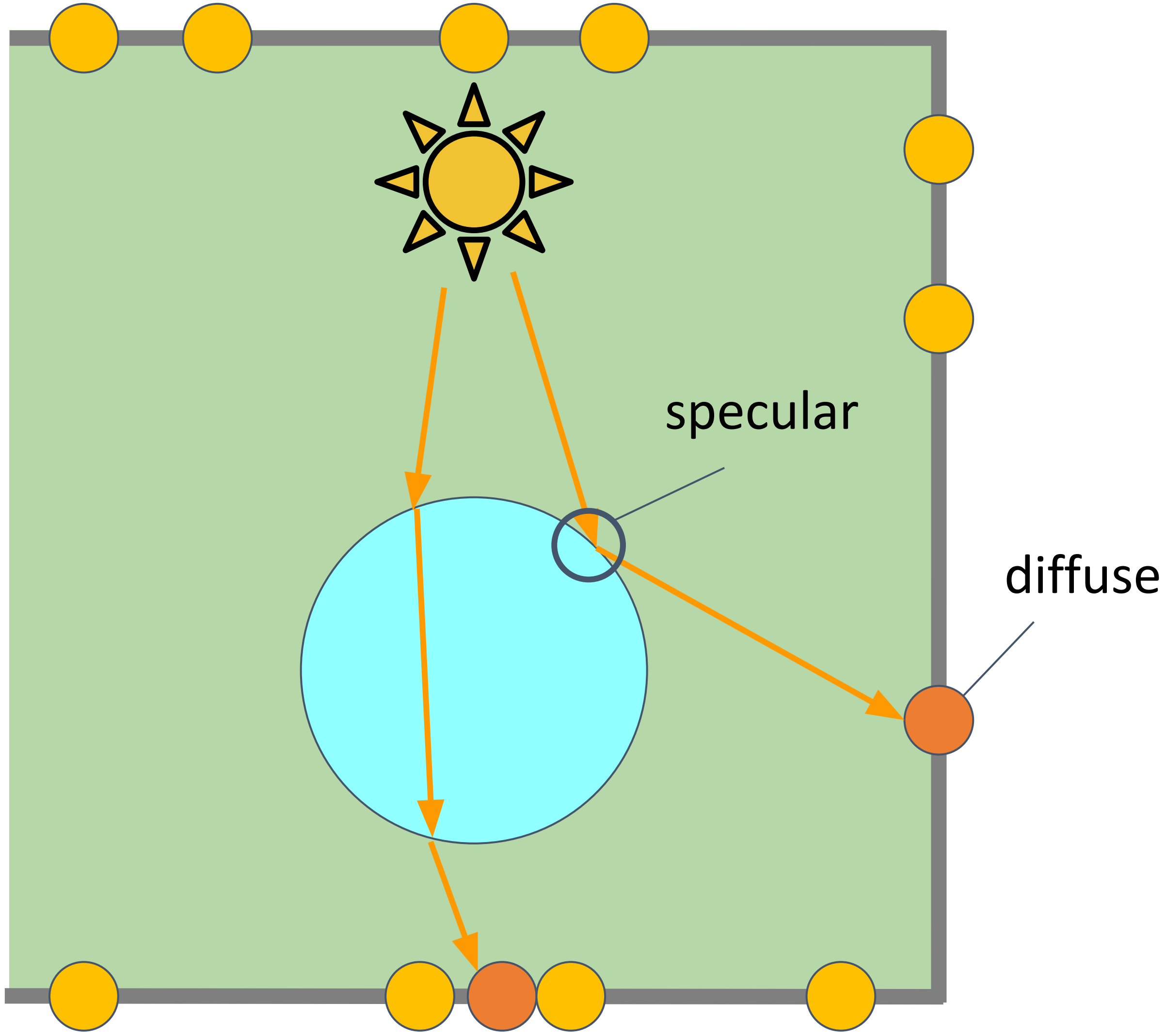
1. Photon map generation: continue the photon walk **without storing a photon**



What to code in today's session

When a **delta BSDF** is hit:

1. Photon map generation: continue the photon walk **without storing a photon**

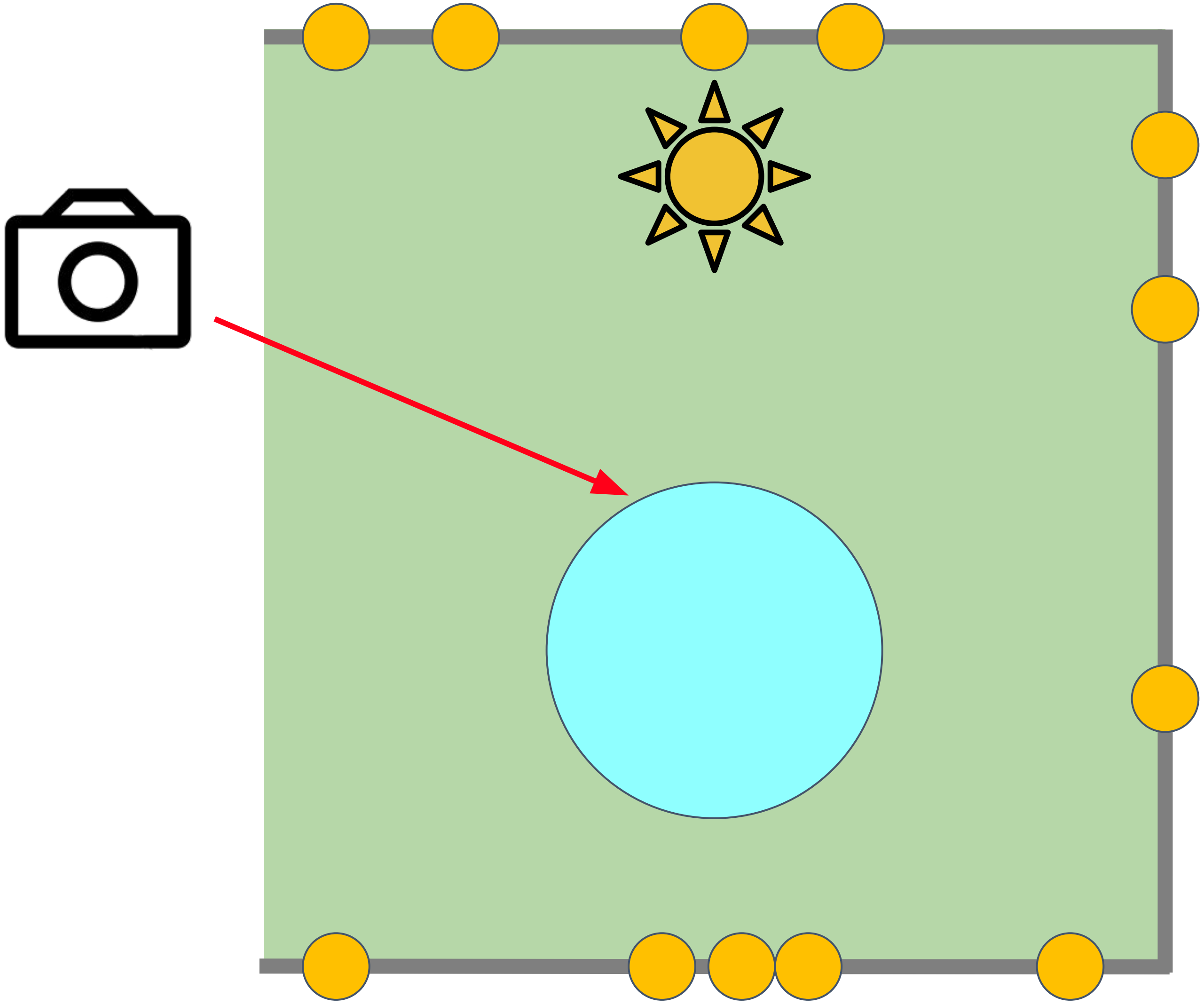


Diffuse, specular and refraction sampling is already implemented (see previous assignments)

What to code in today's session

When a **delta BSDF** is hit:

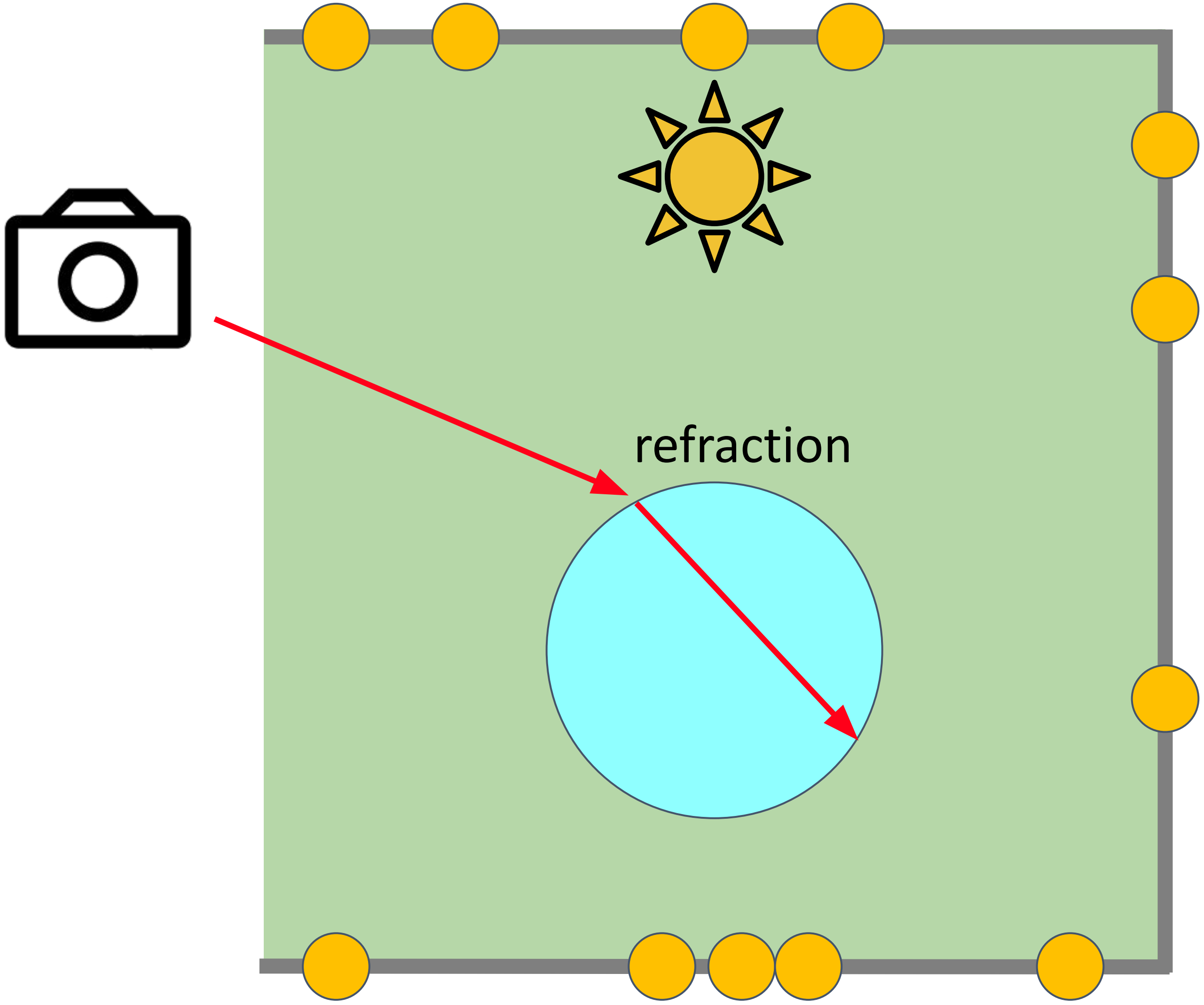
1. Photon map generation: continue the photon walk **without storing a photon**
2. Tracing rays from the camera: follow the perfect specular/refraction directions **instead of estimating radiance**



What to code in today's session

When a **delta BSDF** is hit:

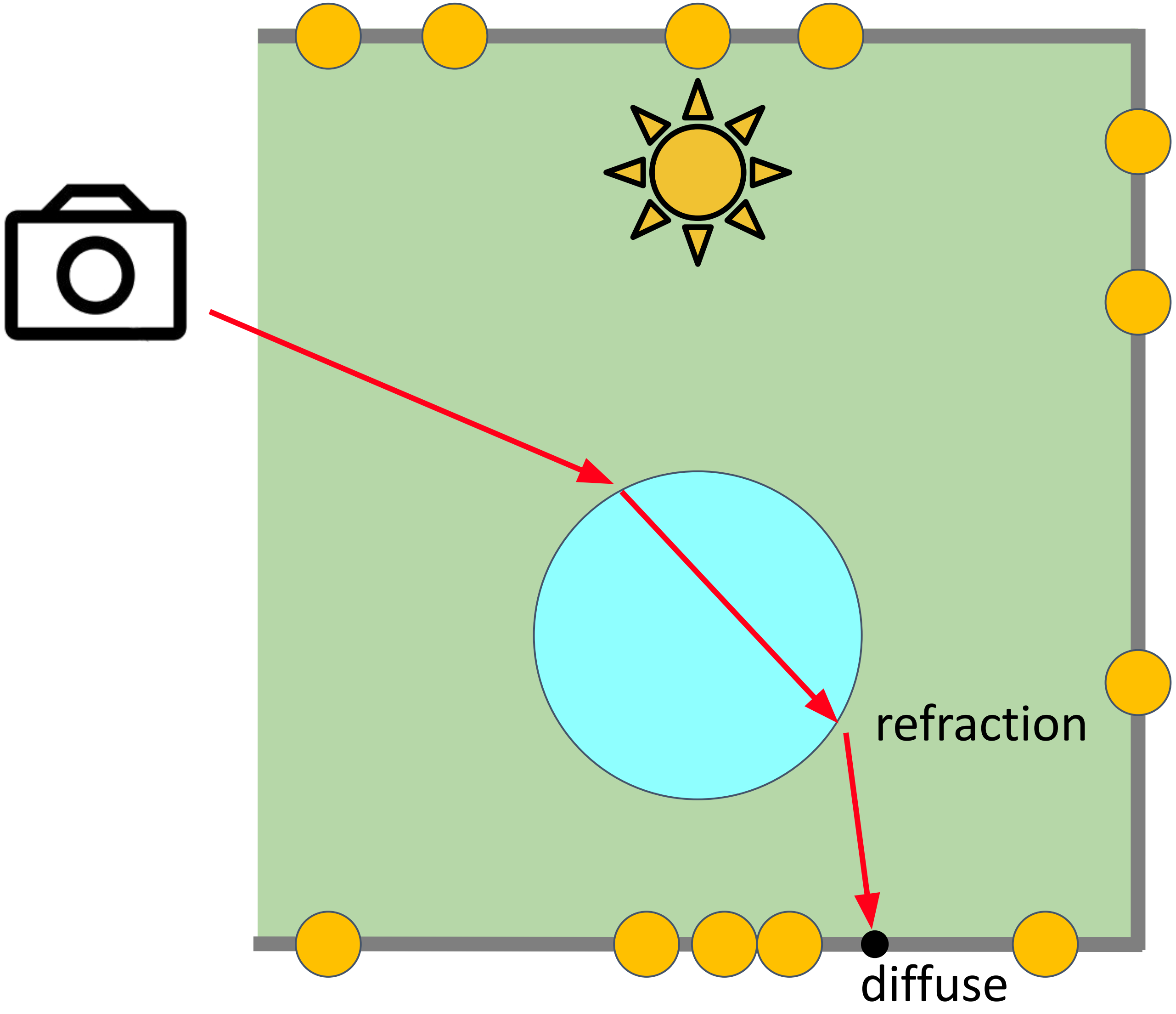
1. Photon map generation: continue the photon walk **without storing a photon**
2. Tracing rays from the camera: follow the perfect specular/refraction directions **instead of estimating radiance**



What to code in today's session

When a **delta BSDF** is hit:

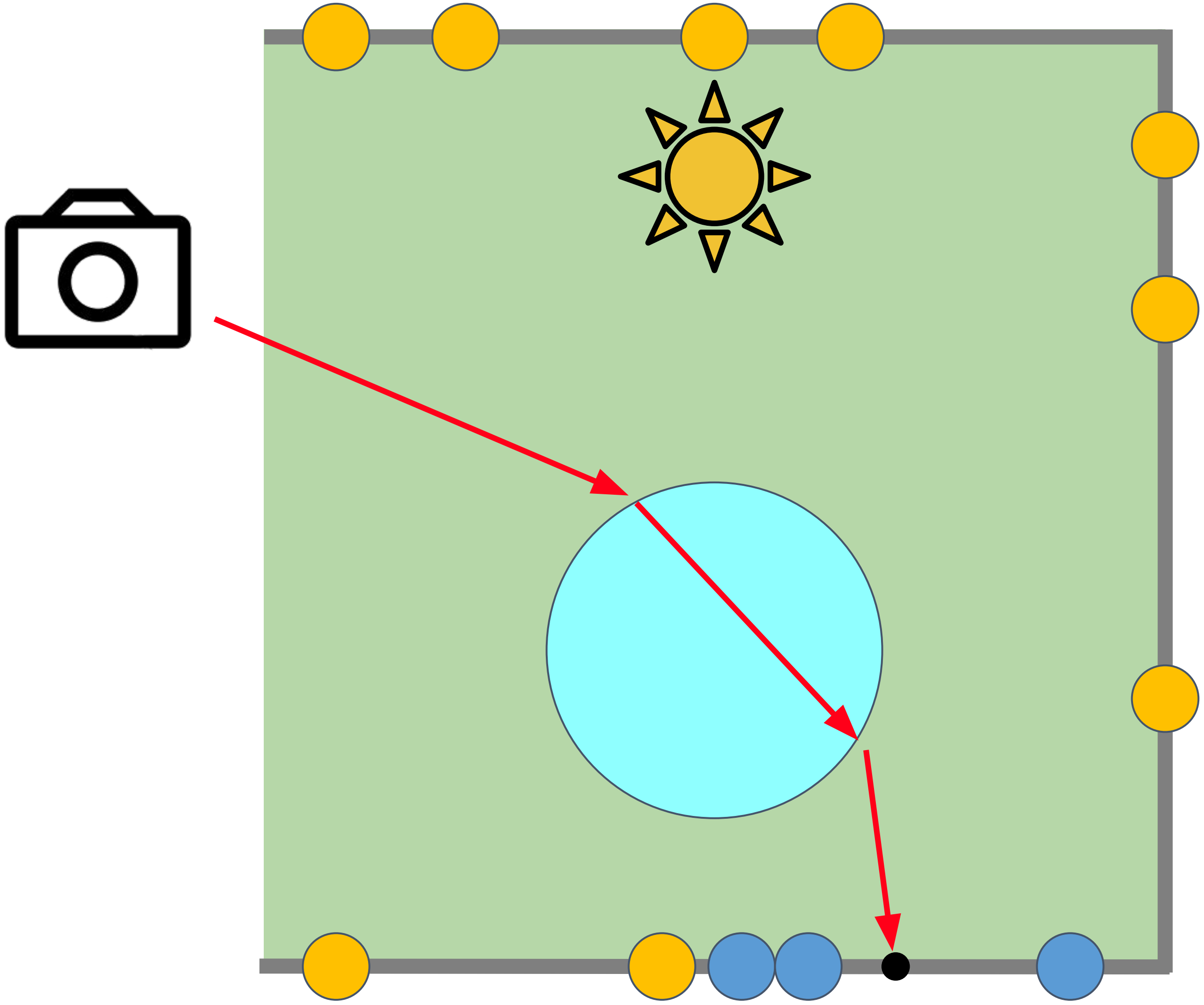
1. Photon map generation: continue the photon walk **without storing a photon**
2. Tracing rays from the camera: follow the perfect specular/refraction directions **instead of estimating radiance**



What to code in today's session

When a **delta BSDF** is hit:

1. Photon map generation: continue the photon walk **without storing a photon**
2. Tracing rays from the camera: follow the perfect specular/refraction directions **instead of estimating radiance**



Russian roulette and materials

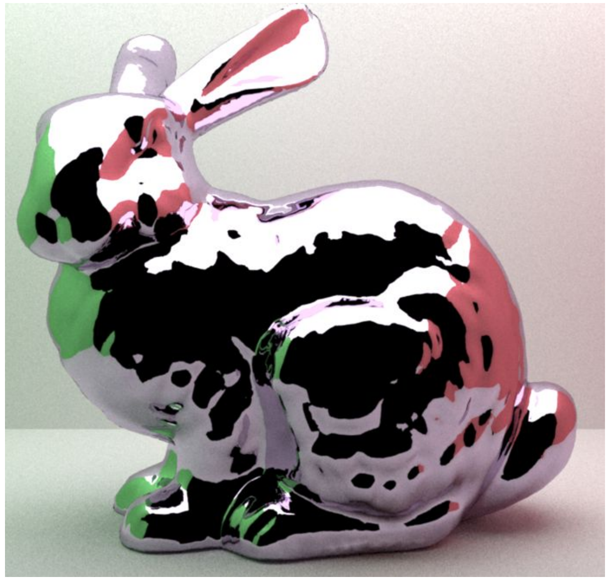
- Recap: you can combine coefficients to get different materials

Diffuse

$k_d > 0$

+

Specular



$k_s > 0$

=

Plastic

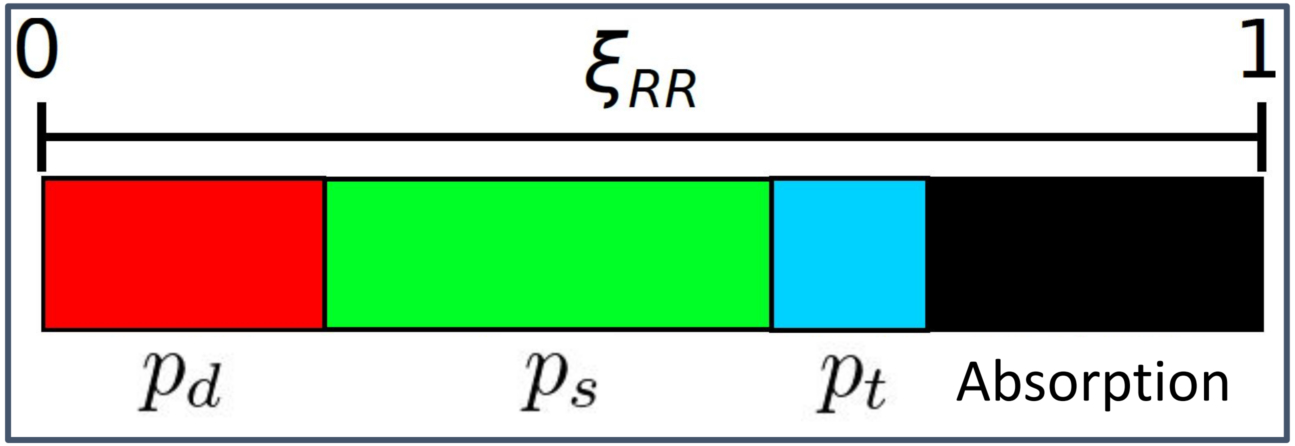
$k_d, k_s > 0$

Russian roulette and materials

Solution:

- **Russian Roulette:** sample one random event (diffuse, specular, refraction or absorption)

Random number $\xi_{RR} \in [0, 1]$:



$$f_r(\mathbf{x}, \omega_\mathbf{i}, \omega_\mathbf{o}) = k_d \frac{1}{\pi} + k_s \frac{\delta_{\omega_\mathrm{r}}(\omega_\mathbf{i})}{\mathbf{n} \cdot \omega_\mathbf{i}} + k_t \frac{\delta_{\omega_\mathrm{t}}(\omega_\mathbf{i})}{\mathbf{n} \cdot \omega_\mathbf{i}}$$

Russian Roulette and materials

Implement Russian Roulette in each part of the photon mapping algorithm:

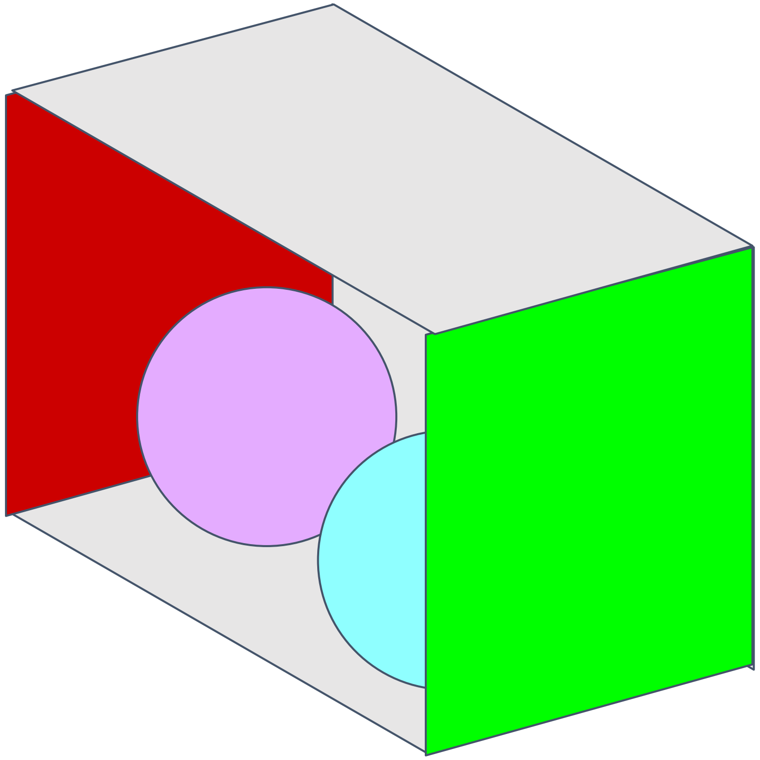
- Photon map generation
- Ray tracing evaluation

Both steps use the same Russian Roulette with four events:

- Diffuse:
  - Store the photon (photon map generation)
  - Use density estimation (tracing rays from the camera)
- Specular/refraction: the path continues in a delta direction instead of what it did before
- Absorption: the path ends (don’t store photons/return no radiance)

Example scene: Cornell Box

- **Geometry**



Planes defined by normal (n) and distance (d)

Left plane	n = (1, 0, 0), d = 1
Right plane	n = (-1, 0, 0), d = 1
Floor plane	n = (0, 1, 0), d = 1
Ceiling plane	n = (0, -1, 0), d = 1
Back plane	n = (0, 0, -1), d = 1

Spheres defined by center (c) and radius (r)

Left sphere c = (-0.5, -0.7, 0.25), r = 0.3

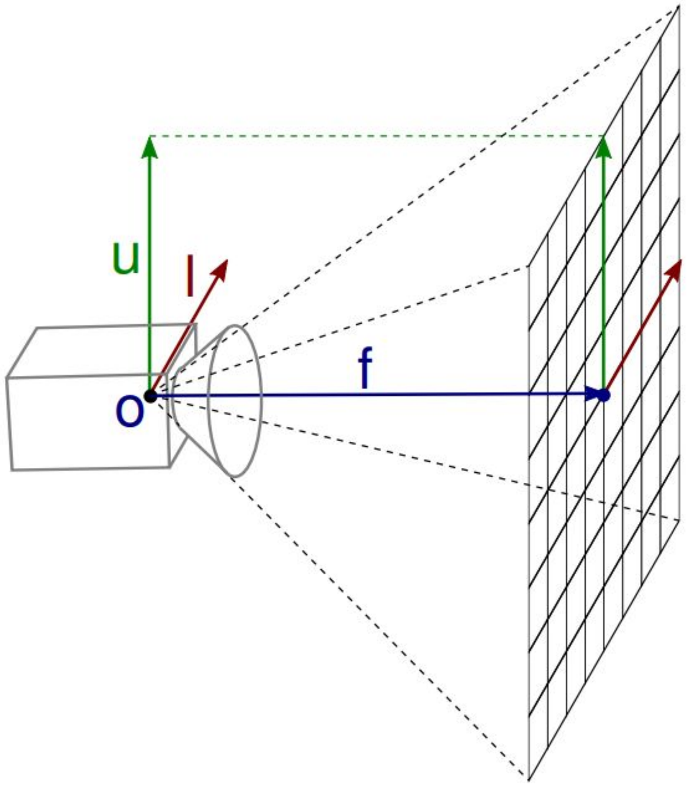
- Mix of blue diffuse + specular

Right sphere c = (0.5, -0.7, -0.25), r = 0.3

- Mix of specular and refraction, $\eta = 1.5$

Example scene: Cornell Box

- **Camera & light sources**

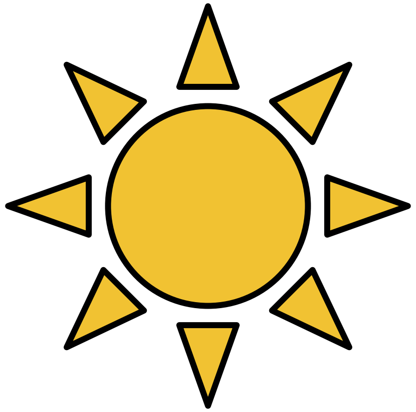


Camera and image plane defined by

Origin	O = (0, 0, -3.5)
Left	L = (-1, 0, 0)
Up	U = (0, 1, 0)
Forward	F = (0, 0, 3)
Size	256x256 pixels

Example scene: Cornell Box

- **Light sources**



Center and power (emission)

Center c = (0, 0.5, 0)

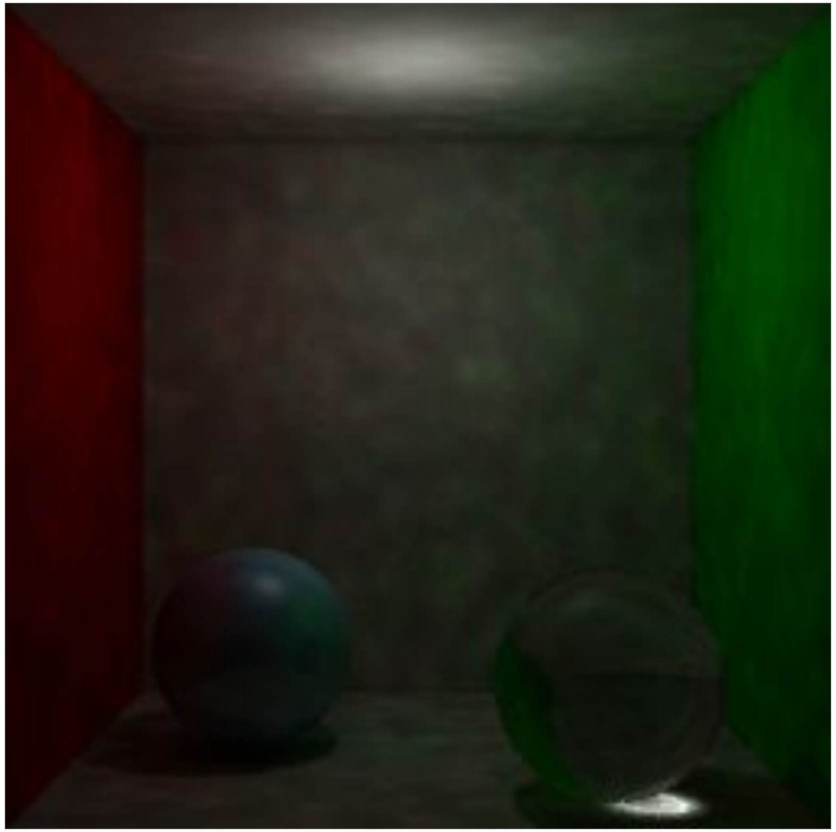
Power can be any number e.g. p = (1, 1, 1)

Just be careful with the #MAX

```
P3
# feep.ppm
#MAX=<maximum of your RGB memory values>
4 4
15
 0  0  0    0  0  0    0  0  0   15  0 15
```

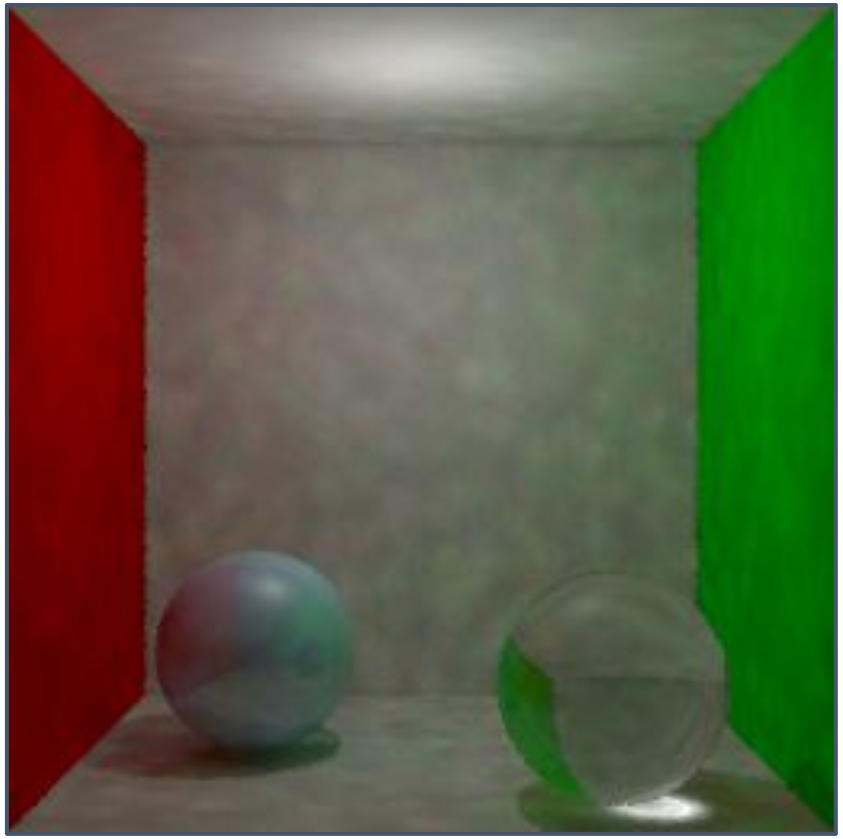
Example scene: Cornell Box

- **Results (direct light is computed using next-event estimation)**



Using a point light

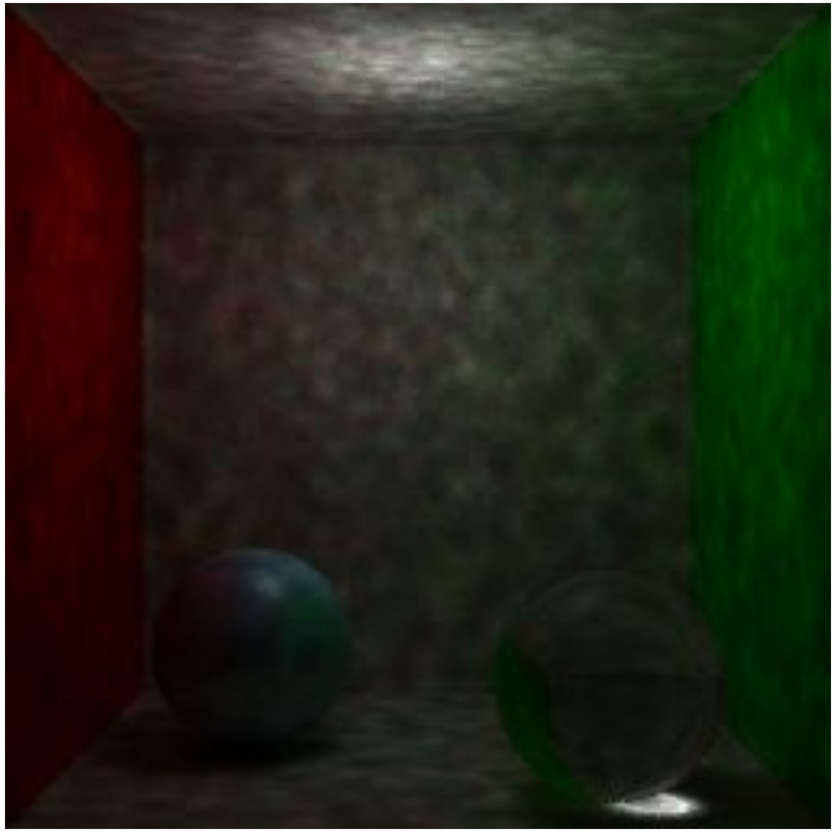
Equalization
Gamma = 2.2



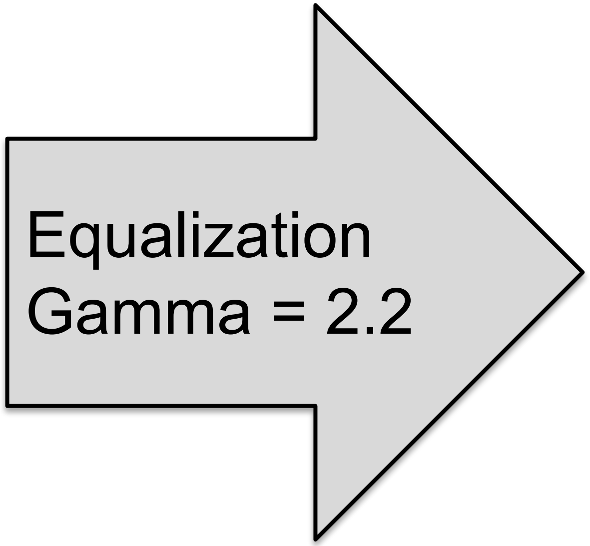
With tone mapping

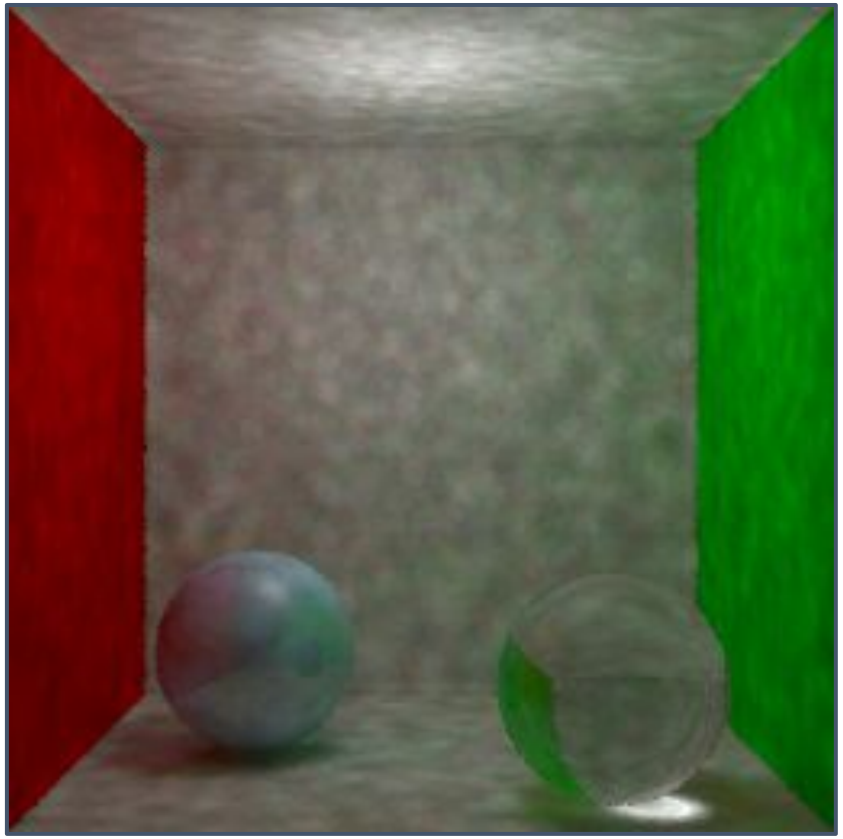
Example scene: Cornell Box

- **Results (direct light is computed using photon mapping)**



Using a point light





With tone mapping

Questions

DO ASK questions, either now or after the lab

But be reasonable, please :)

pluesia@unizar.es | dsubias@unizar.es | o.pueyo@unizar.es

What to expect from this session

In the programming language of your choice implement:

- Perfect specular and refractive materials in photon mapping:
  - Use Russian Roulette to select an event (same as path tracing)
  - **Diffuse event:** same as previous session
  - **Specular/refraction events:** path continues in a delta direction instead of what it did before
  - **Absorption:** path ends (don't store photons, and return zero radiance)
- Recommended deadline: December 4th (moodle: January 11th)
  - Extensions (do not count towards recommended deadline):
    - **Recommended to finish base photon mapper before any optionals**
    - Try **different kernels** (cone, Gaussian, or more sophisticated ones)
    - Use an **adaptive kernel bandwidth** (radius)
    - Others: participating media, transient photon mapping, etc. (talk with us before)