University of BRISTOL

COMS 30115

Photon Mapping

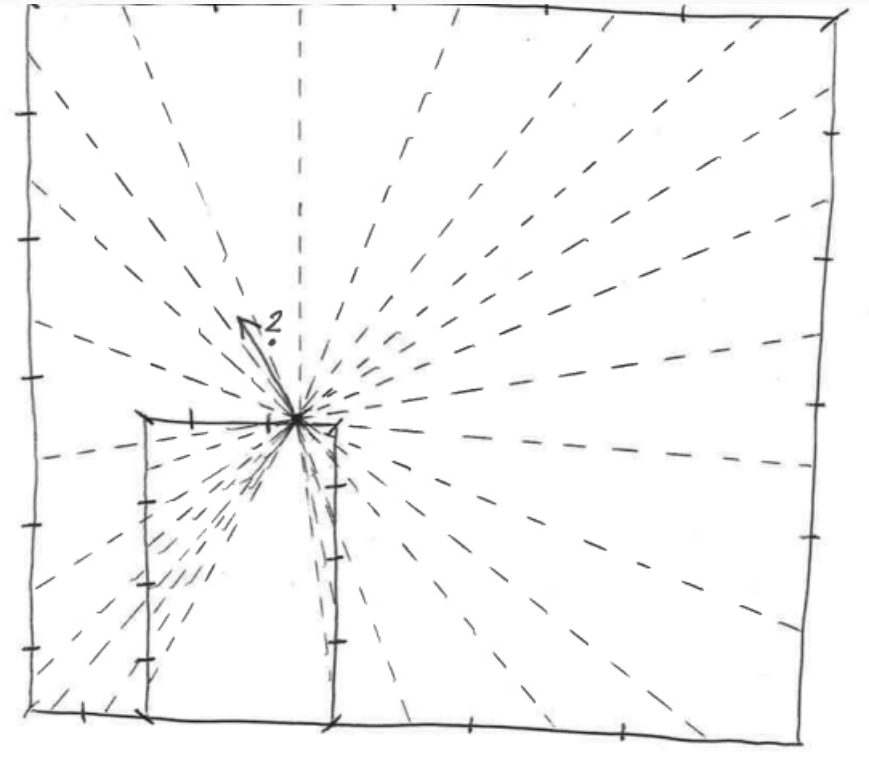
Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

April 1st, 2019

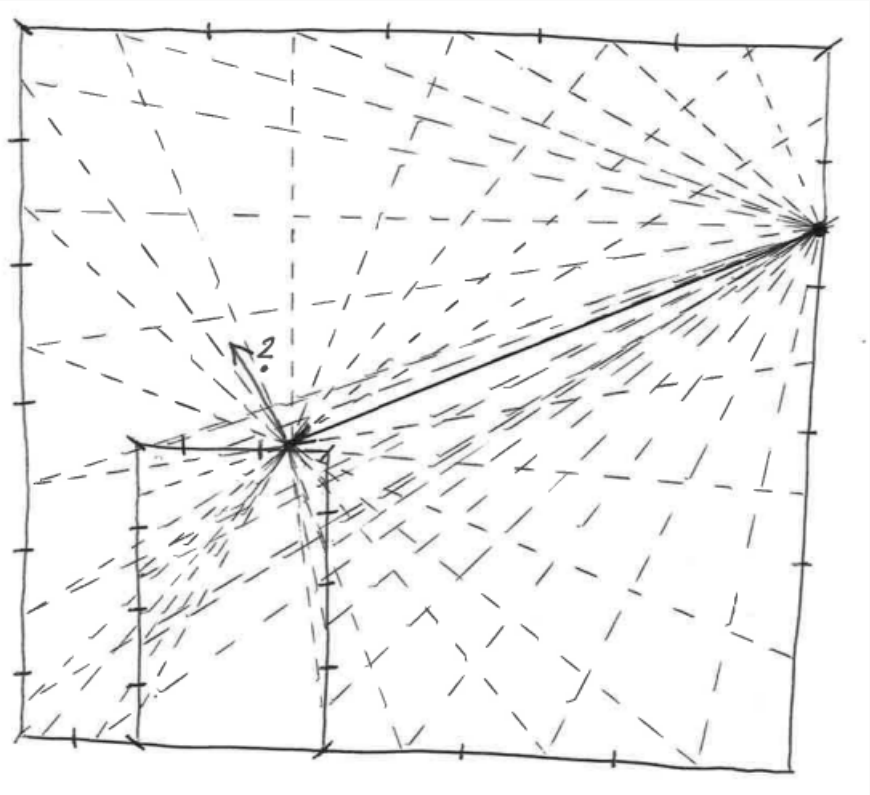
http://www.carlhenrik.com

Introduction

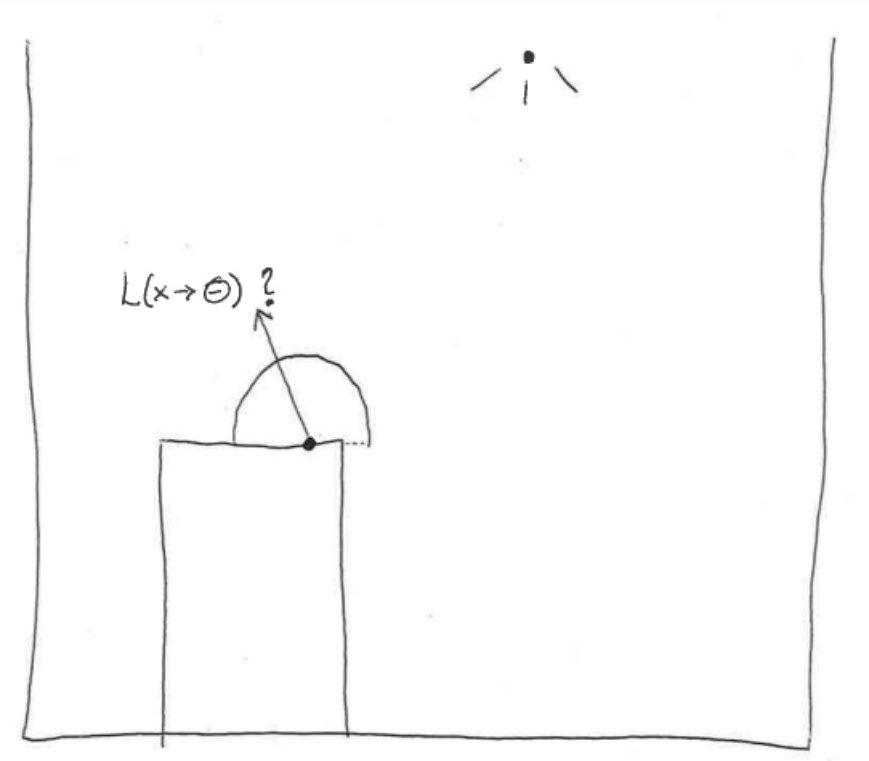
Transport of Light



Transport of Light

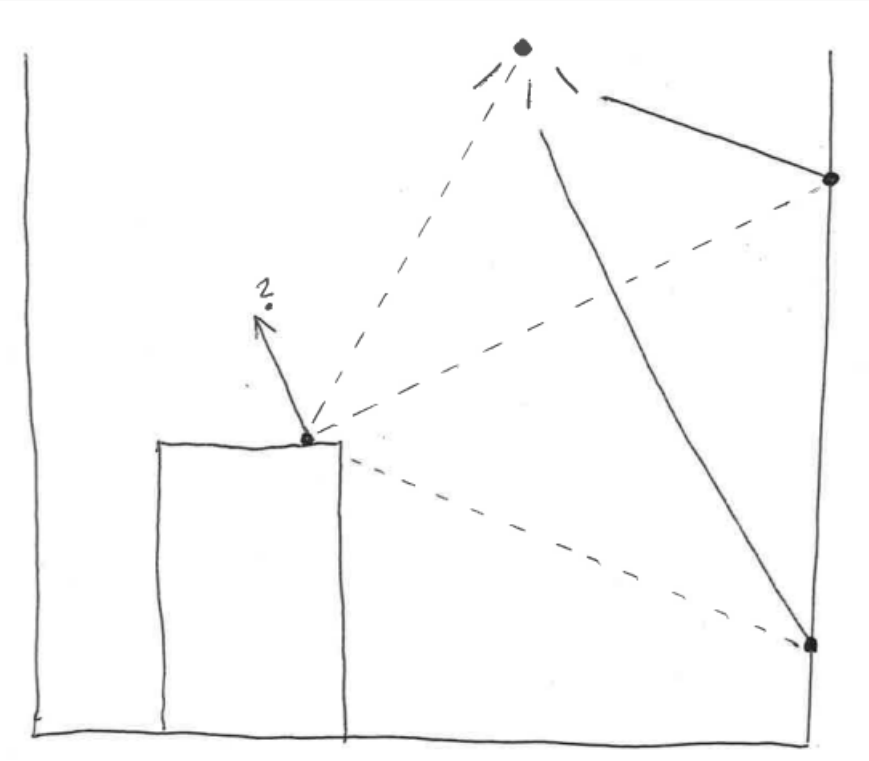


Transport of Light



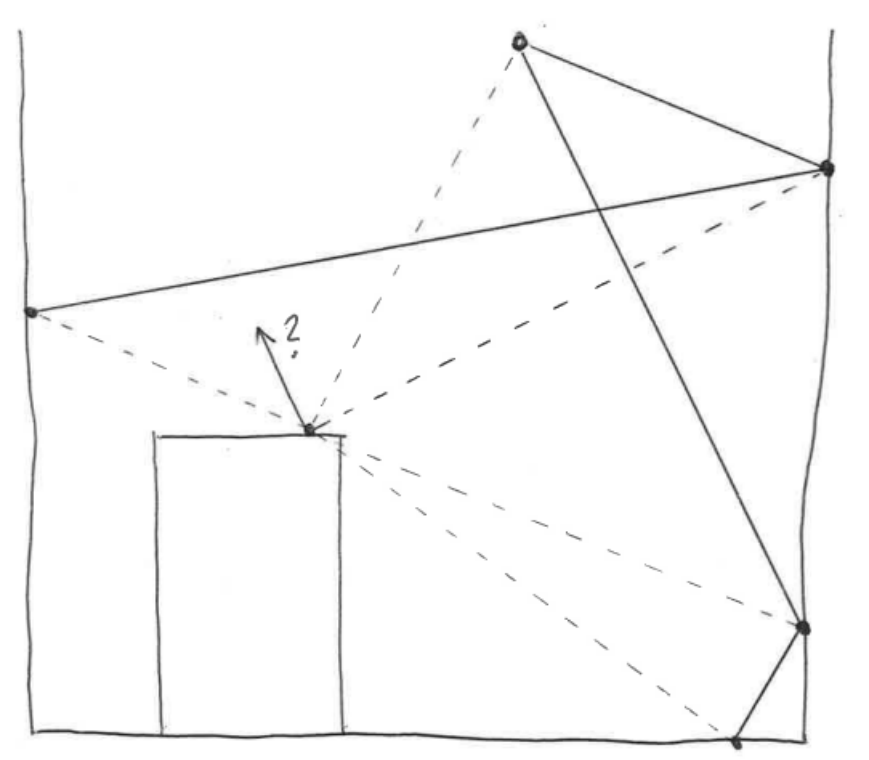
$$L(x \to \Theta) = L_e(x \to \Theta) + \dots$$

Transport of Light



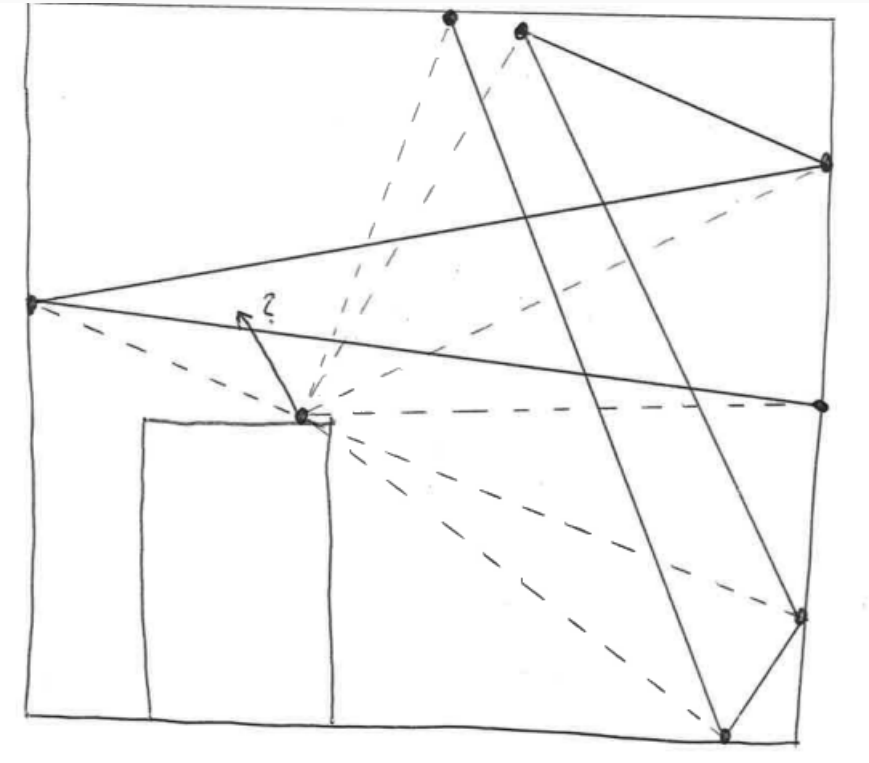
$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \dots$$

Transport of Light



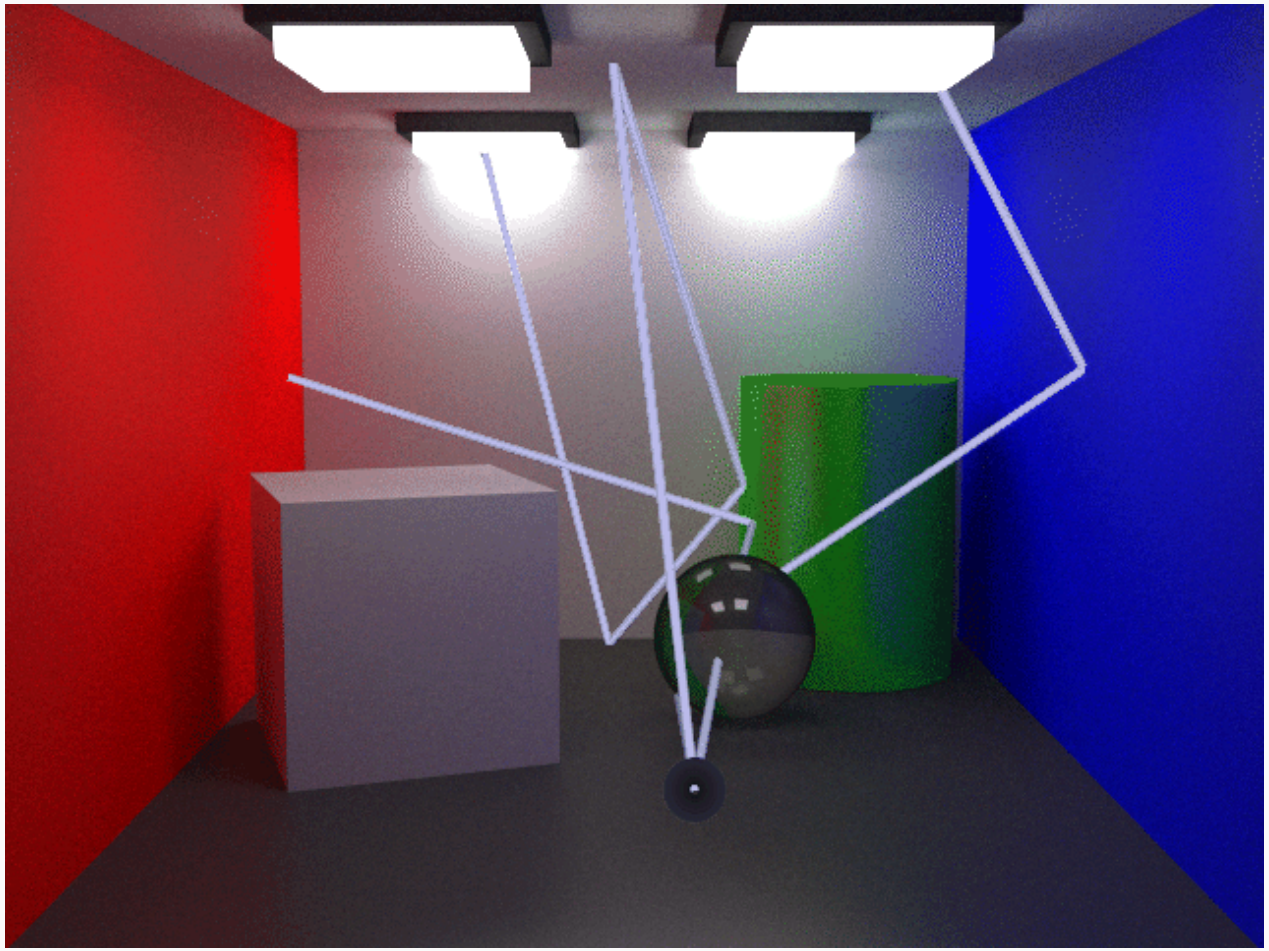
$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \langle T, TL_e \rangle + \dots$$

Transport of Light



$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \langle T, TL_e \rangle + \langle T, TTL_e \rangle + \dots$$

Paths



Pathtracing

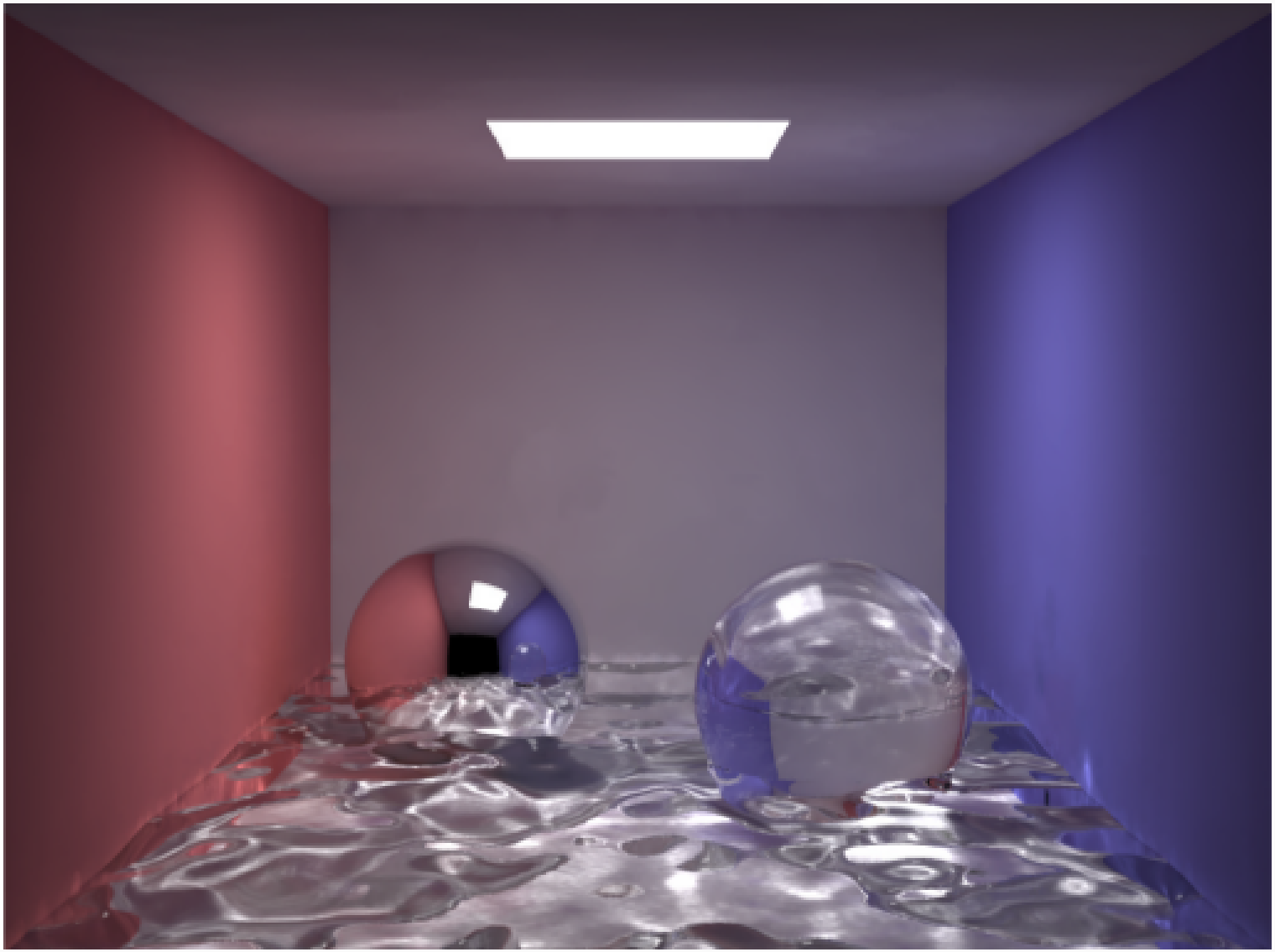
The Good

- Will be correct in the limit
- Simple to implement
- Lots of theoretical work in sampling
- Unbiased

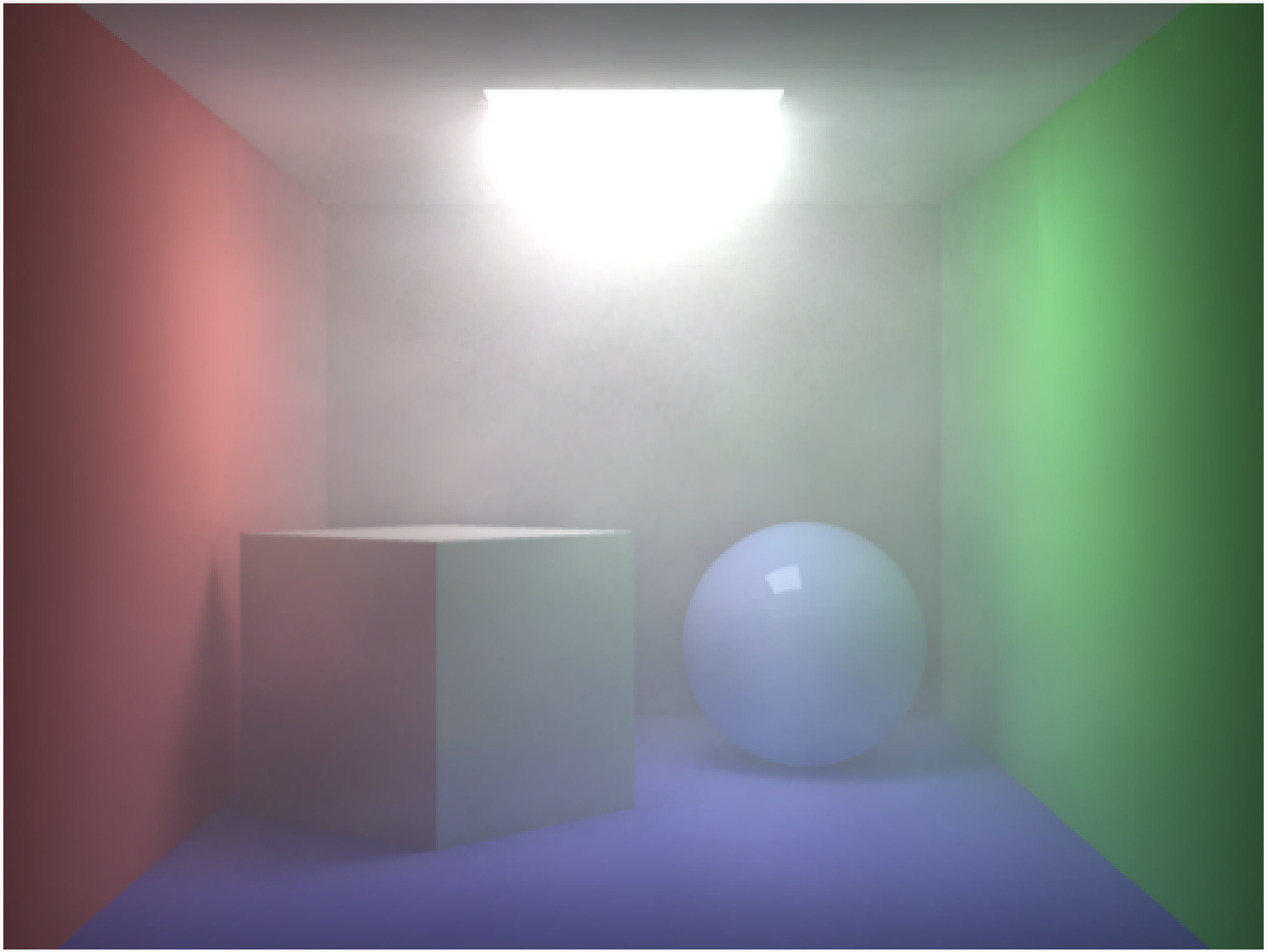
The Bad

- Slow, very slow
- high frequent noise
- hard to simulate light concentrations like caustics
- Unbiased

Today



Today



Material

- thesis
- Equation Compendium
- Global Illumination Resources
- Henrik Jensen Siggraph Tutorial

Photon Mapping

Photon Mapping

- Developed by Henrik Jensen TU Copenhagen
- First proposed 1993
- Propsed as a means to speed up path tracing
- Path tracing often results in high-frequent noise which is very apparent while photon mapping results in low-frequent noise

Photon Mapping

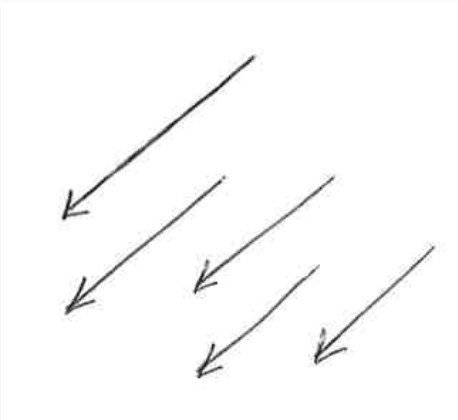
Pass 1 photon shooting stage

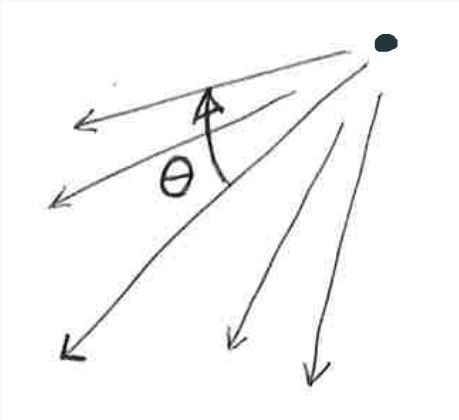
- From each lightsource distribute photons around the scene
- Forward raytracer

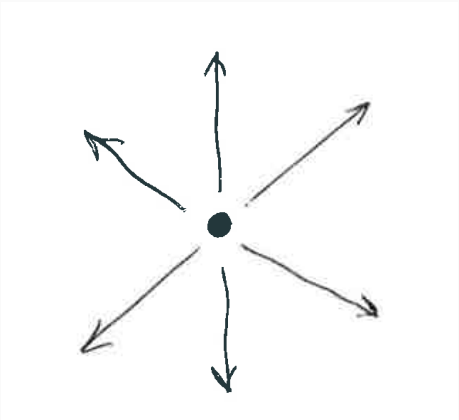
Pass 2 gathering stage

- shoot camera rays and gather the light that has been distributed
- Backward raytracer

Photon Mapping

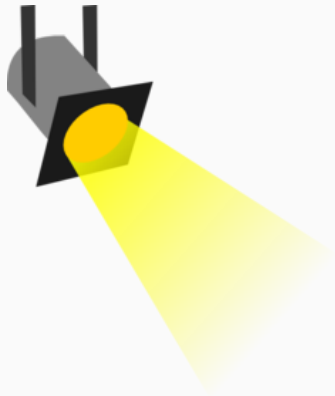






$$\Phi_{\text{photon}} = \frac{\Phi_{\text{light}}}{N}$$

Shooting Photons



Shooting photons is exactly the same as shooting rays, however, we now shoot flux while in the path tracer we gathered radiance

Photon Mapping

- When a photon hits an object it can either be,
  1. Reflected
  2. Refracted
  3. Absorbed
- To determine what happens we play . . .

Russian Roulette

Russian Roulette

- Associate each material with a reflection coefficient
  - s - specular reflection
  - d - diffuse reflection

Russian Roulette

- Associate each material with a reflection coefficient
  - s - specular reflection
  - d - diffuse reflection
- Draw a random number $\eta \in [0, 1]$
  - $\eta \in [0, d] \rightarrow$ diffuse reflection
  - $\eta \in [d, d + s] \rightarrow$ specular reflection
  - $\eta \in [d + s, 1] \rightarrow$ absorption

RGB

$$P_r = \max(d_r + s_r, d_g + s_g, d_b + s_b)$$
$$P_a = 1 - P_r$$

- Diffuse reflection

$$P_d = \frac{d_r + d_g + d_b}{d_r + d_g + d_b + s_r + s_g + s_b} P_r$$

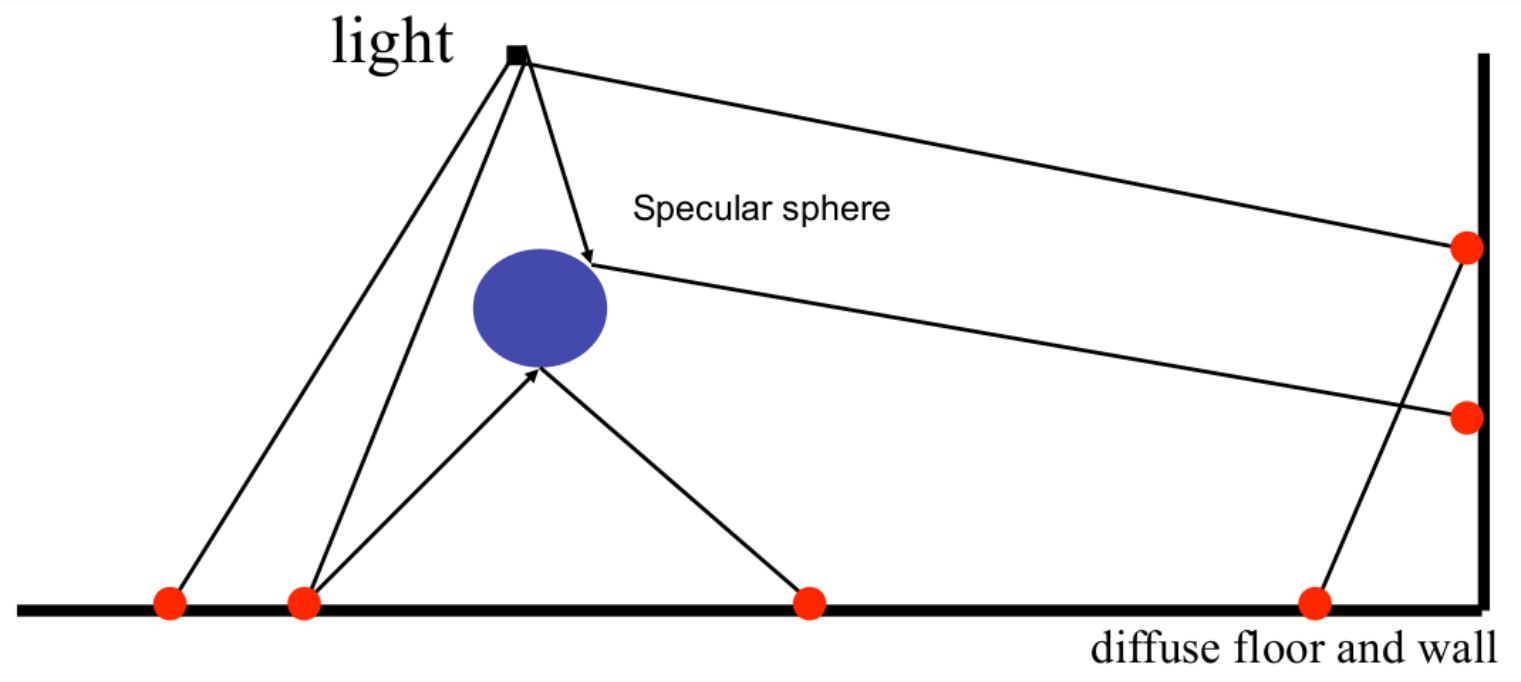
- Specular reflection

$$P_s = \frac{s_r + s_g + s_b}{d_r + d_g + d_b + s_r + s_g + s_b} P_r$$

RGB

$$\eta \in [0, P_d] \rightarrow \text{diffuse reflection}$$
$$\eta \in [P_d, P_s + P_d] \rightarrow \text{specular reflection}$$
$$\eta \in [P_s + P_d, 1] \rightarrow \text{absorption}$$

Photon Maps



- We keep tracing the photon until it gets absorbed
- This map is called the photon map
- Only store at non-specular locations

Projection Maps

$$\Phi_{\text{photon}} = \frac{\Phi_{\text{light}}}{N} \cdot \frac{\text{cells with objects}}{\text{total number of cells}}$$

- Rather than shooting photons blindly we can be a bit clever about it
- Create a map which contains the world as seen from the lightsource
- Discritisise the map into cells
- Use this as a mask to avoid performing intersection checks

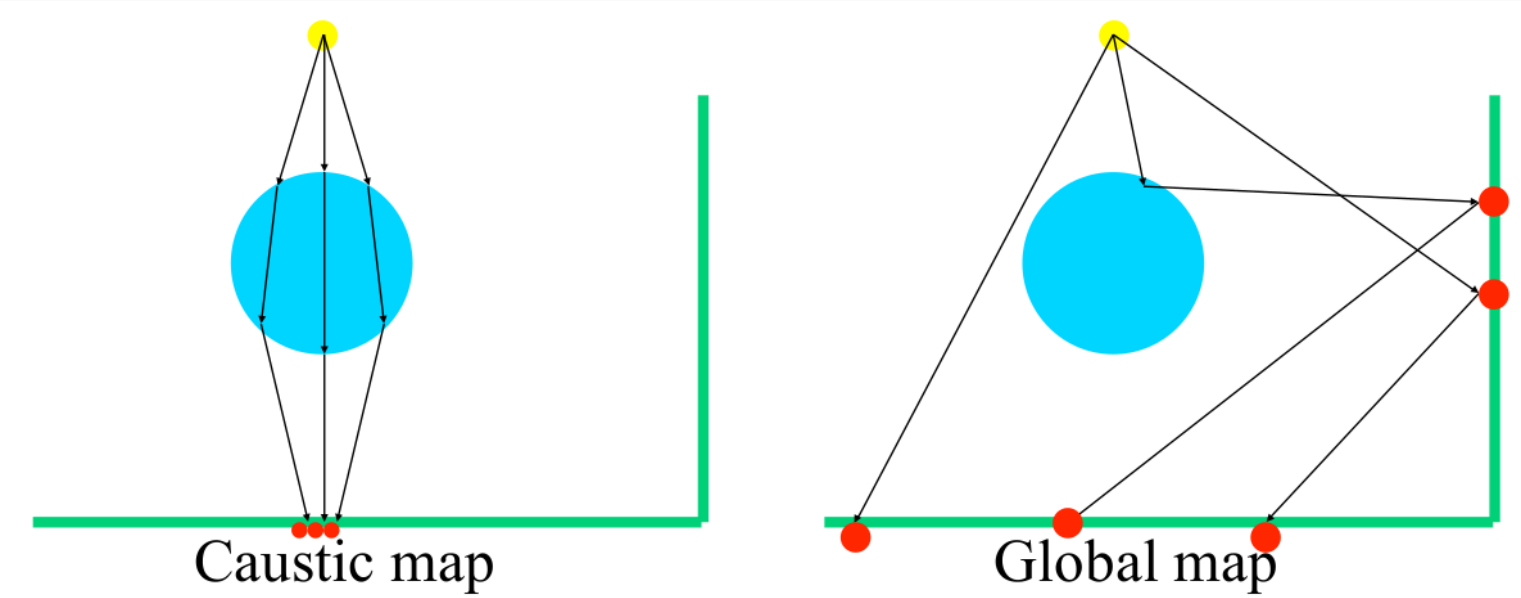
Photon Storage

Caustic map Create a projection map that specifically "targets" areas which could generate caustics

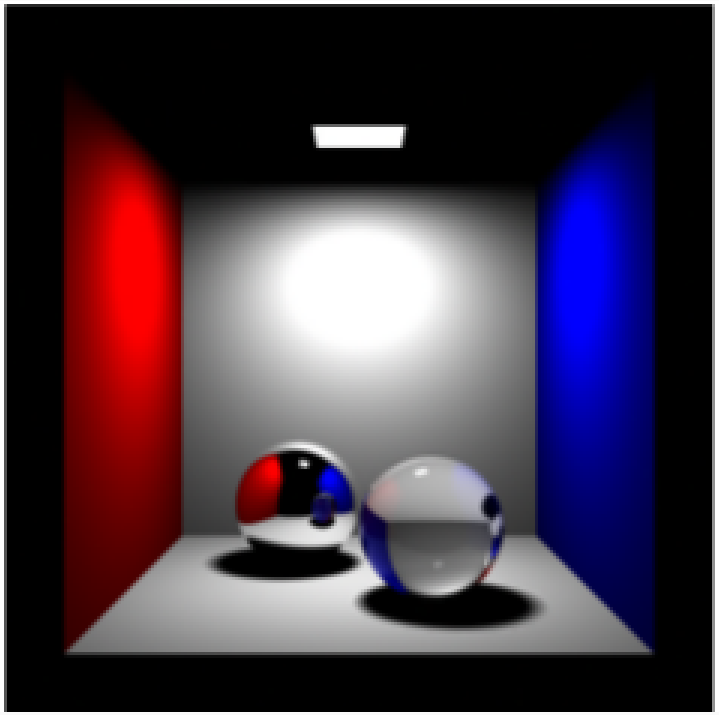
Volume map Create a projection map for participating media where interactions happens randomly

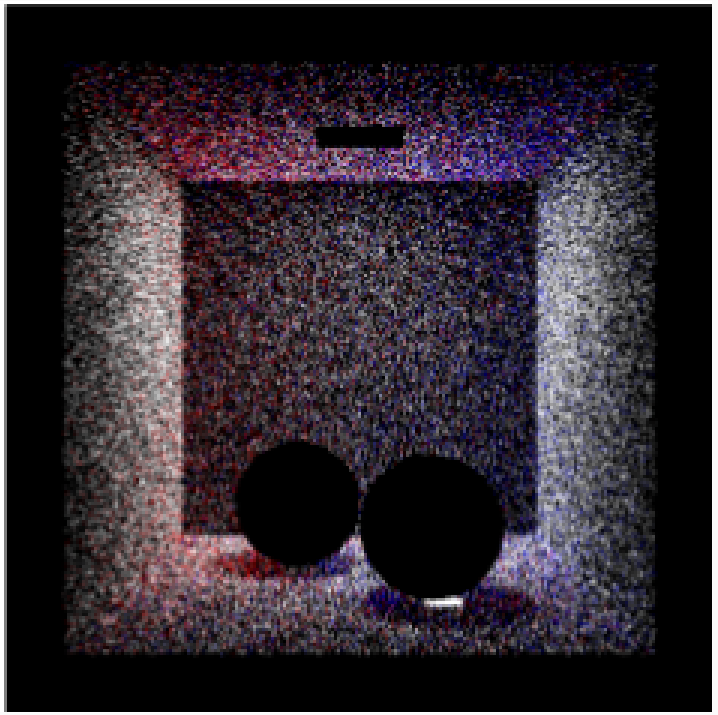
Global map A projection map that contains any geometry

Photon Maps



Photon Maps[1]





[1] A Practical Guide to Global Illumination using Photon Maps - Jensen

Pass II

Factorisation

BRDF

$$f_r(x, \Psi \rightarrow \Theta) = f_{r,s}(x, \Psi \rightarrow \Theta) + f_{r,d}(x, \Psi \rightarrow \Theta)$$

- factorise BRDF in specular and diffuse components

Light

$$L(x \leftarrow \Psi) = L_l(x \leftarrow \Psi) + L_c(x \leftarrow \Psi) + L_d(x \leftarrow \Psi)$$

- factorise *incoming* radiance in direct, caustic and indirect

Rendering Equation

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi =$$

Rendering Equation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi =$$

$$= L_e(x \to \Theta) + \int_{\Omega_x} (f_{r,s} + f_{r,d}) L_l cos(\cdot) d\omega_\Psi + \int_{\Omega_x} f_{r,s}(L_c + L_d) cos(\cdot) d\omega_\Psi +$$

$$\int_{\Omega_x} f_{r,d} L_c cos(\cdot) d\omega_\Psi + \int_{\Omega_x} f_{r,d} L_d cos(\cdot) d\omega_\Psi$$

Rendering

$$\int_{\Omega_x} (f_{r,s} + f_{r,d}) L_I cos(\cdot) d\omega_\Psi$$

- Direct illumination, just as in bog-standard raytracer

$$\int_{\Omega_x} f_{r,s} (L_c + L_d) cos(\cdot) d\omega_\Psi$$

- Specular/glossy reflection very "peaked" so use normal path tracing

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

Rendering

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \rightarrow \Theta)}$$

$$L(x \leftarrow \Psi) = \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA}$$

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

$$L_r(x \to \Theta) = \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA} cos(\mathbf{n}_x, \Psi) d\Psi$$

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

$$\begin{aligned} L_r(x \to \Theta) &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA} cos(\mathbf{n}_x, \Psi) d\Psi \\ &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{dA} \end{aligned}$$

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

$$\begin{aligned} L_r(x \to \Theta) &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA} cos(\mathbf{n}_x, \Psi) d\Psi \\ &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{dA} \\ &\approx \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta\Phi(x, \Psi)}{\Delta A} \end{aligned}$$

Rendering

$$L(x \to \Theta) \approx L_e(x \to \Theta) + \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta \Phi_p(x, \Psi)}{\Delta A}$$

- If we assume that each photon have flux $\Delta \Phi_p$ we can "approximate" the sum above by finding the N closest photons to x and summing their flux
- *Think of this as expanding a sphere around x*

Rendering

$$L(x \to \Theta) \approx L_e(x \to \Theta) + \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta \Phi_p(x, \Psi)}{\Delta A}$$

$$= L_e(x \to \Theta) + \frac{1}{\pi r^2} \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \Delta \Phi_p(x, \Psi)$$

- If we assume that each photon have flux $\Delta \Phi_p$ we can "approximate" the sum above by finding the N closest photons to x and summing their flux
- *Think of this as expanding a sphere around x*

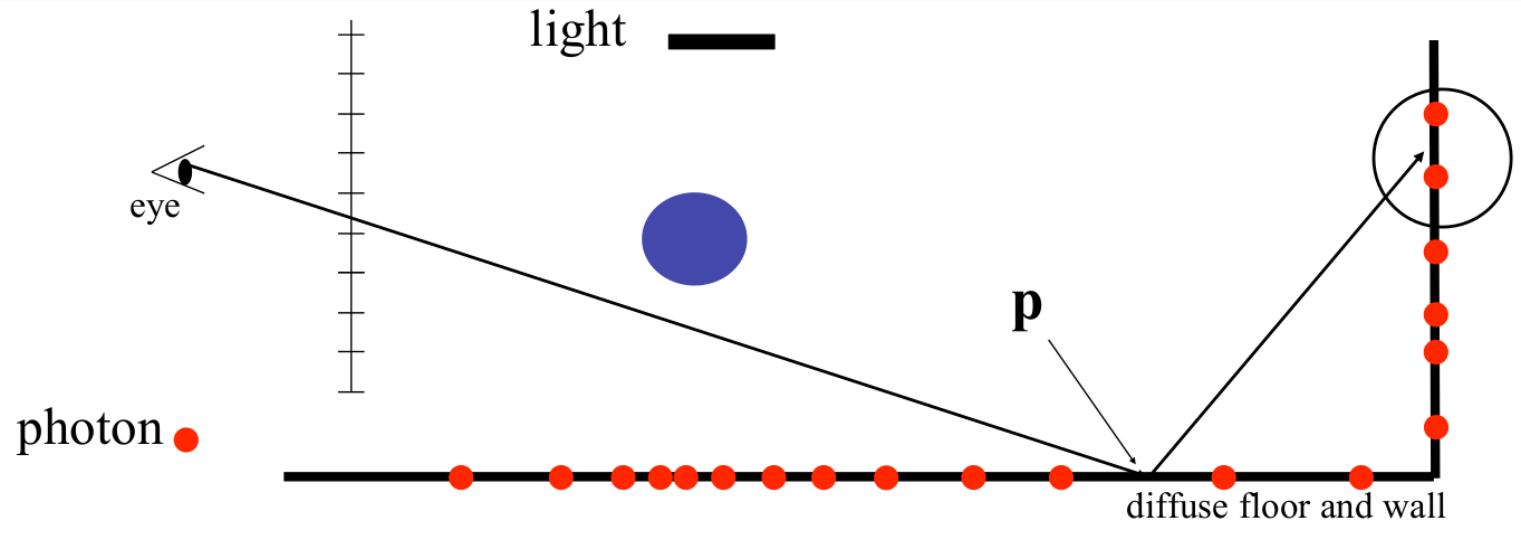
Rendering

$$L(x \to \Theta) \approx L_e(x \to \Theta) + \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta \Phi_p(x, \Psi)}{\Delta A}$$

$$= L_e(x \to \Theta) + \frac{1}{\pi r^2} \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \Delta \Phi_p(x, \Psi)$$

- If we assume that each photon have flux $\Delta \Phi_p$ we can "approximate" the sum above by finding the N closest photons to x and summing their flux
- *Think of this as expanding a sphere around x*
- This will be very wrong if the area of accumulation is not flat

Rendering



Rendering

$$\int_{\Omega_x} f_{r,d} L_c cos(\cdot) d\omega_\Psi = \frac{1}{\pi r^2} \sum_{p=1}^{N} f_{r,d}(x, \Psi \rightarrow \Theta) \Delta\Phi_p^c(x, \Psi)$$

- Caustics will be evaluated with the Caustic Photon Map

$$\int_{\Omega_x} f_{r,d} L_d cos(\cdot) d\omega_\Psi = \frac{1}{\pi r^2} \sum_{p=1}^{N} f_{r,d}(x, \Psi \rightarrow \Theta) \Delta\Phi_p^g(x, \Psi)$$

- Diffuse light will be evaluated with the Global Photon Map

Optimisations

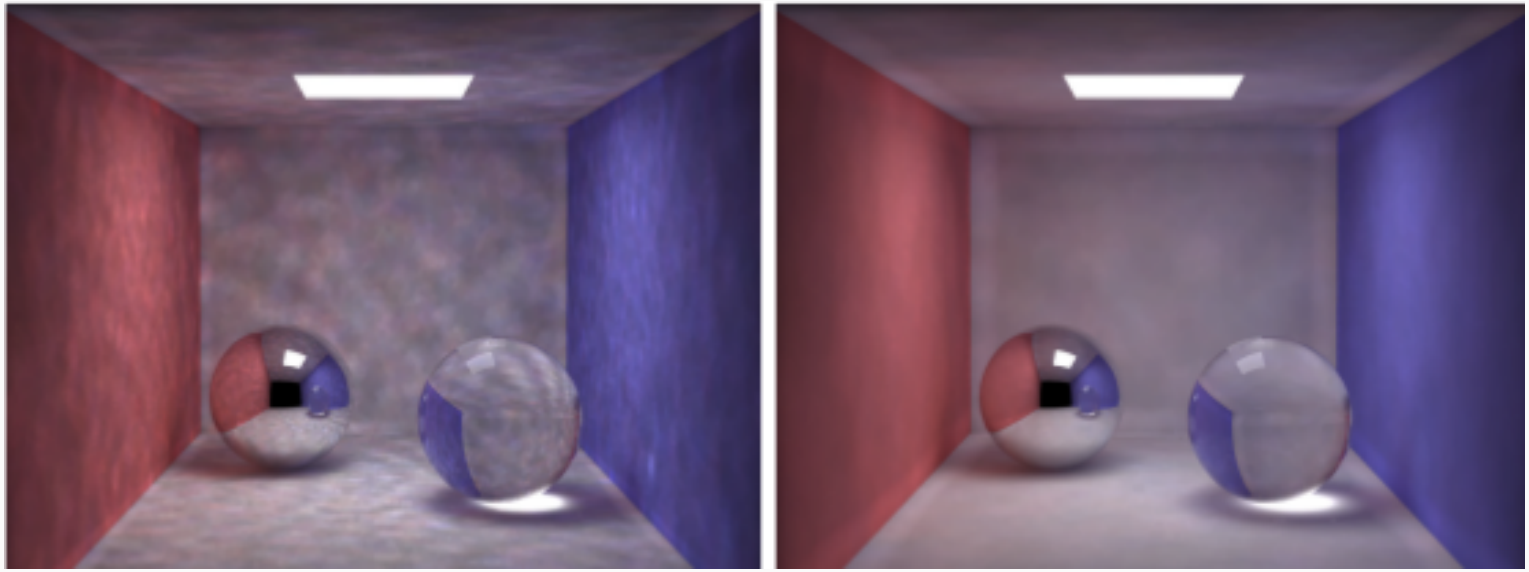
Datastructures

- When we render the scene we compute a lot of distances
- Store the photon-maps in a tree to save look-up time
- This is where the big speed-up can be done and is worth looking into

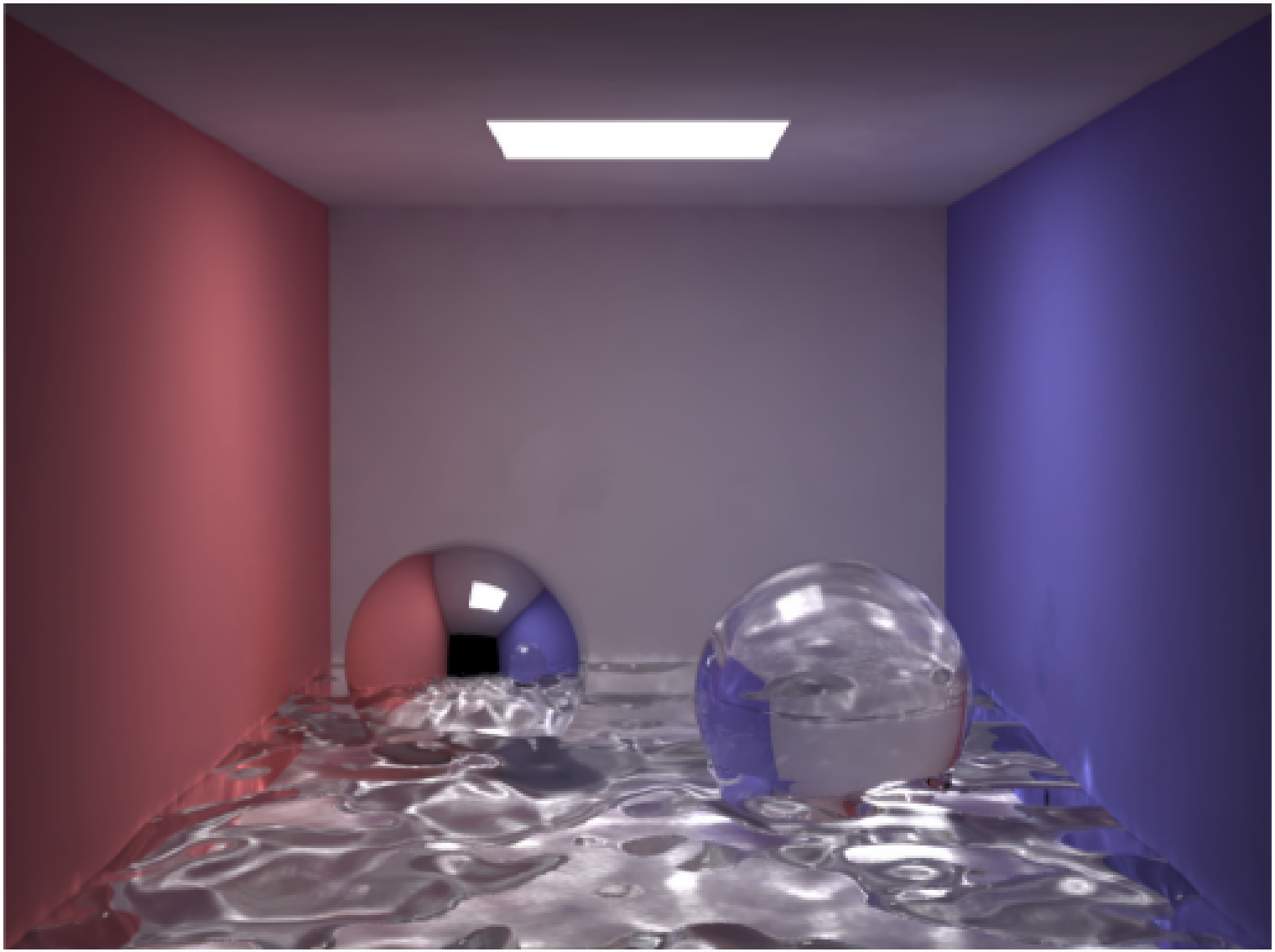
Irradiance Caching

- Indirect light is more likely to be "smooth"
- Cache computations for certain points in the scene
- Interpolate out the missing values

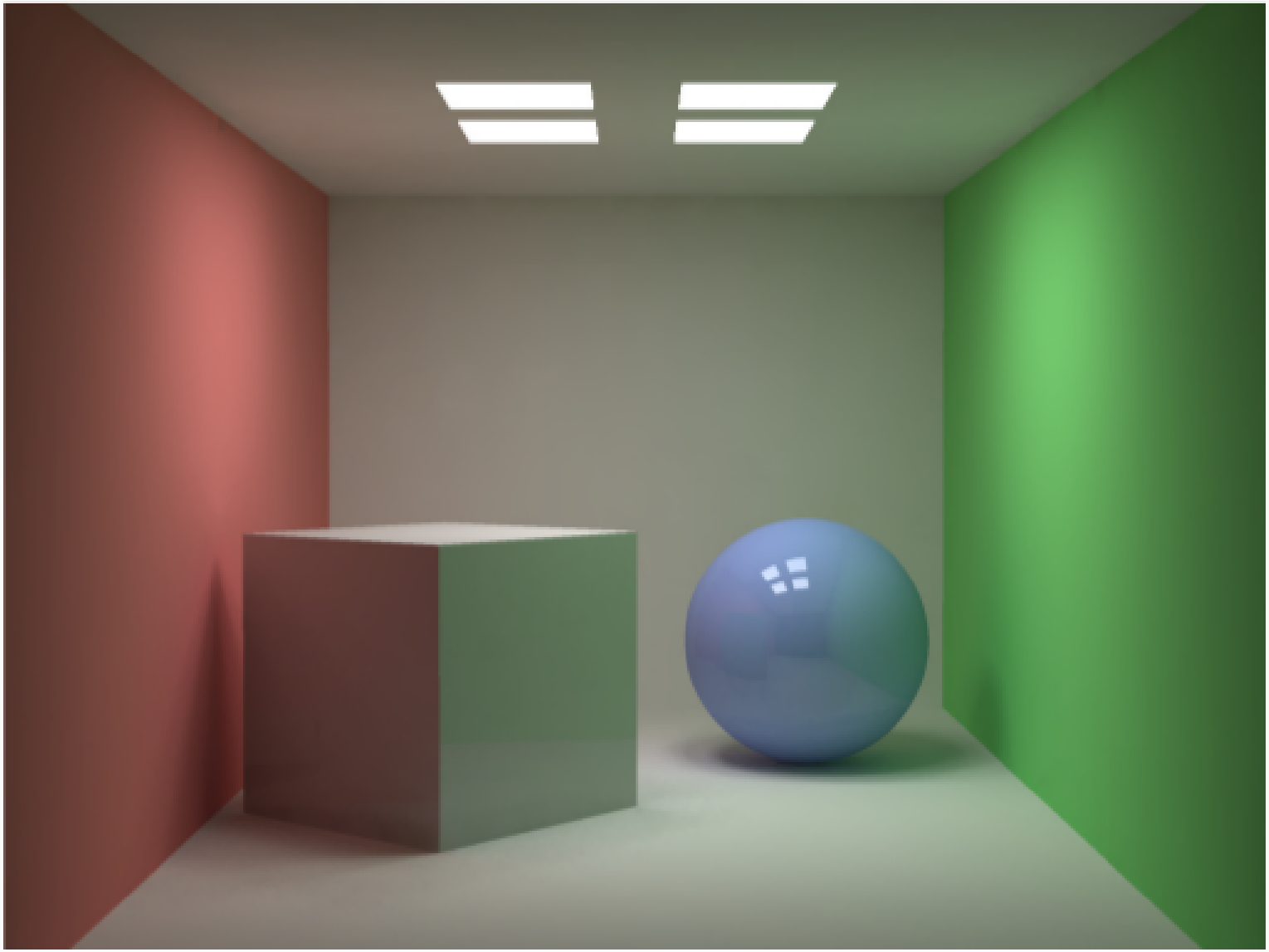
Results[1]



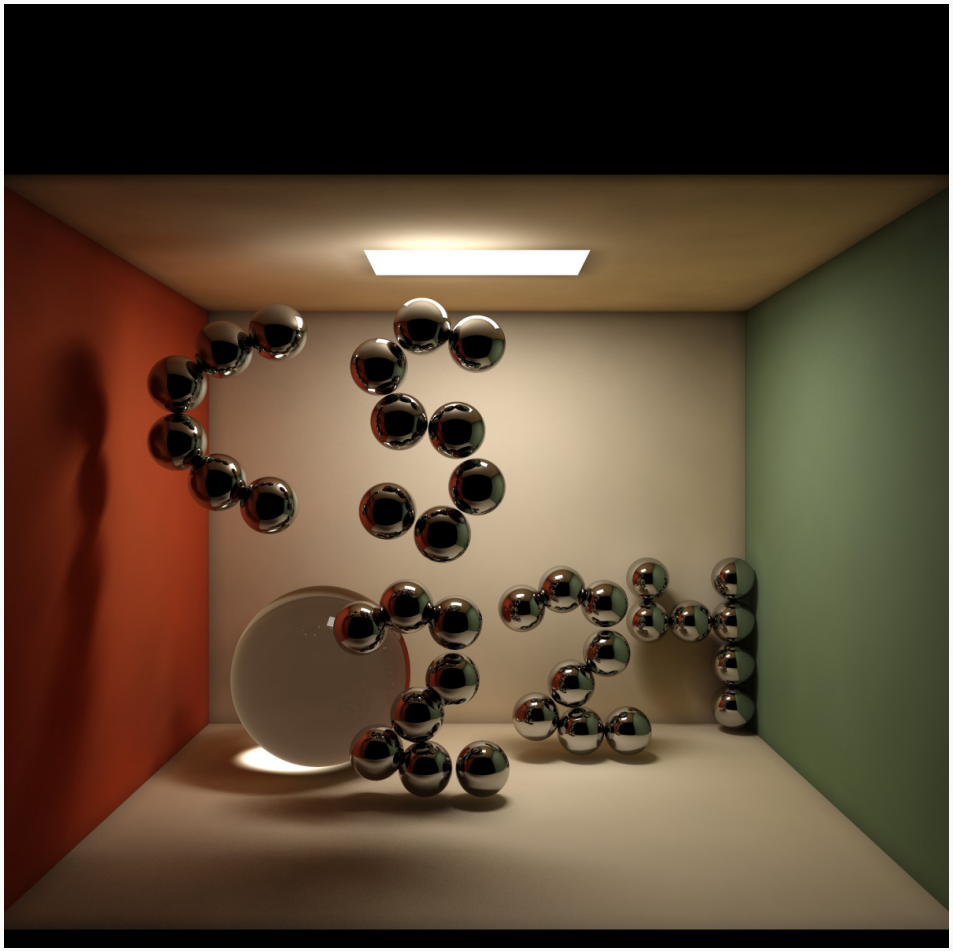
Results[1]



Results[1]

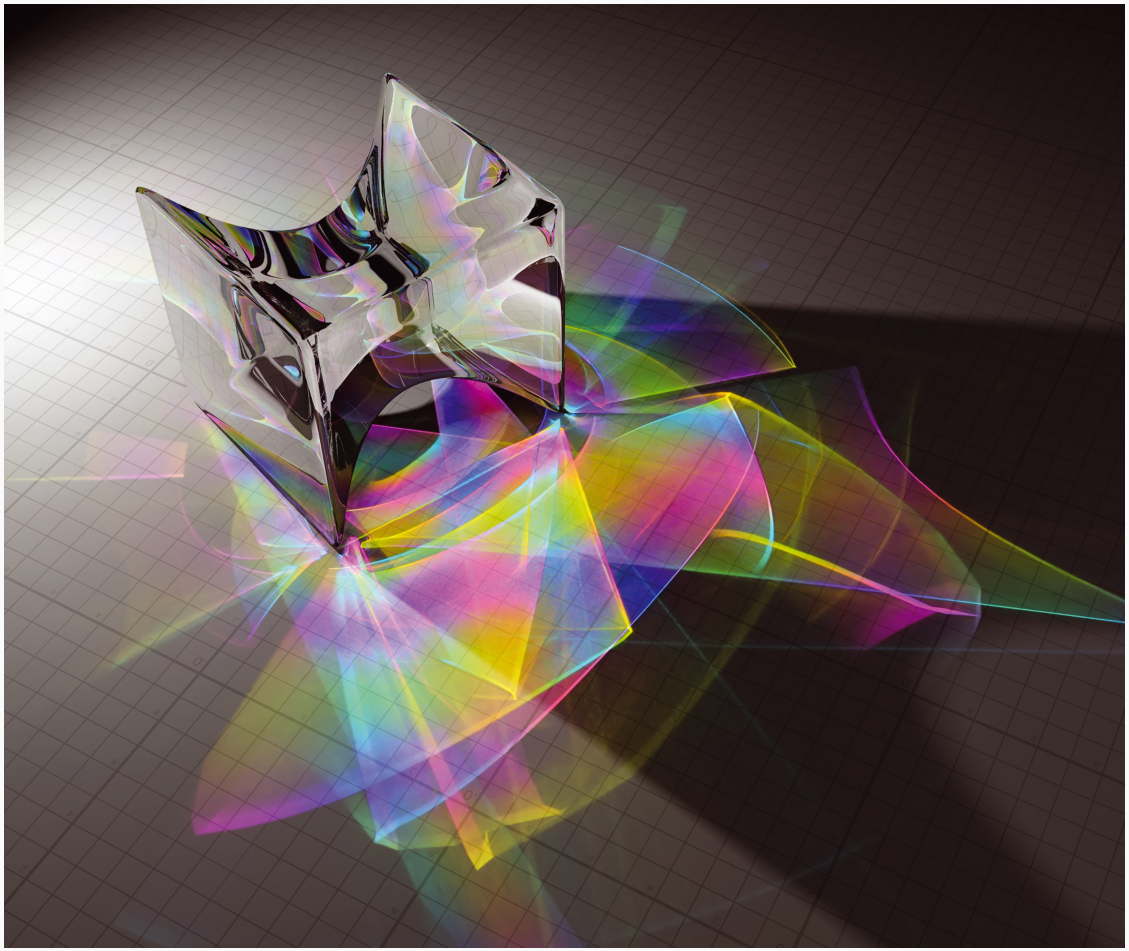


Results[2]



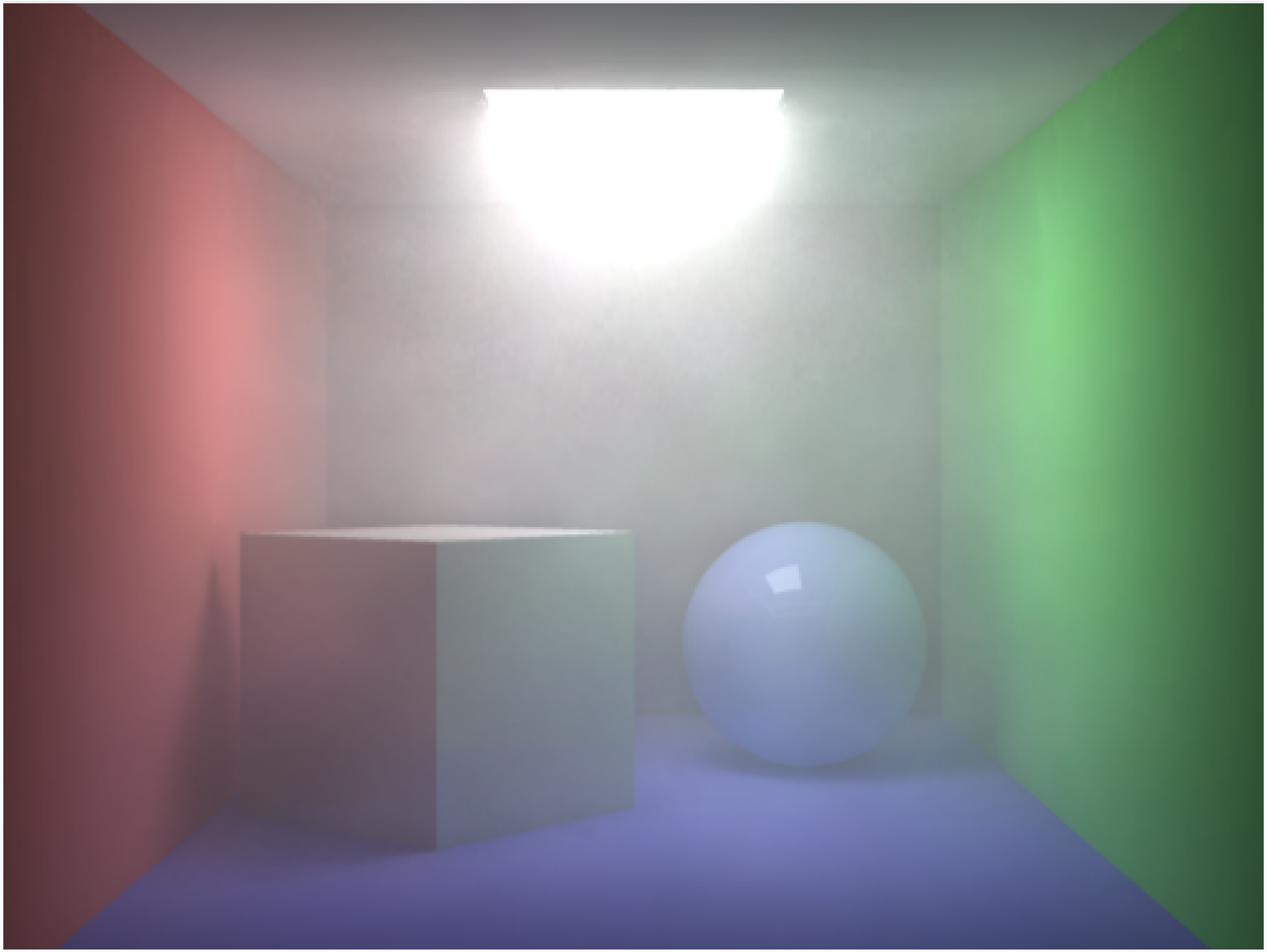
[2] http://marctenbosch.com/photon/

Results[3]



[3] University of Swansea - Visual Computing

Results[1]



Summary

- Two-pass rendering strategy
- Allows to capture things like caustics
- More "hacky" compared to path tracing or radiosity
- Uses several different approaches together

Summary

Global Illumination

Radiosity Completely diffuse surfaces

- Does colour bleeding really well
- Equation system
- View point independent but requires discretisation

Path Tracing Keep tracing light paths in reverse directing

- Correct but slow
- High frequent noise

Photon Mapping Two pass, forward and backward tracing

- Capable of doing caustics easy
- Low frequent noise
- "less" correct

Coursework

Deadlines

Raytracer 3rd of May

- Submit to SAFE

Rasteriser 3rd of May

- Submit to SAFE

Report

Report Submission

- Bulleted list of what you have done

Report Submission

- Bulleted list of what you have done
- Interesting things that you want me to know about

Report Submission

- Bulleted list of what you have done
- Interesting things that you want me to know about
- no fancy formatting

Report Submission

- Bulleted list of what you have done
- Interesting things that you want me to know about
- no fancy formatting
    - in ASCII or `org-mode`

Report Submission

- Bulleted list of what you have done
- Interesting things that you want me to know about
- no fancy formatting
  - in ASCII or `org-mode`
- Each members CANDIDATE NUMBER

Report Submission

- Bulleted list of what you have done
- Interesting things that you want me to know about
- no fancy formatting
  - in ASCII or `org-mode`
- Each members CANDIDATE NUMBER
- If it takes you more than 1h to do you are doing too much

Code

- One person in each group submits the work

Code

- One person in each group submits the work
- The other person submits ..

Code

- One person in each group submits the work
- The other person submits ..
- The basic part should run on the lab machines

Code

- One person in each group submits the work
- The other person submits ..
- The basic part should run on the lab machines
- `README` file

Code

- One person in each group submits the work
- The other person submits ..
- The basic part should run on the lab machines
- `README` file
    - how to compile extensions etc.

Code

- One person in each group submits the work
- The other person submits ..
- The basic part should run on the lab machines
- `README` file
  - how to compile extensions etc.
  - other things I need to know

Makefiles

Code

```
Build: $(OBJ) Makefile
        $(CC) $(LN_OPTS) -o
        $(EXEC) $(OBJ) $(SDL_LDFLAGS)

clean:
        rm -f $(B_DIR)/*

extension_a:

extension_b:
```

Viva

Viva

- tick that you have done the work!

Viva

- tick that you have done the work!
- make sure that I have understood what you have done

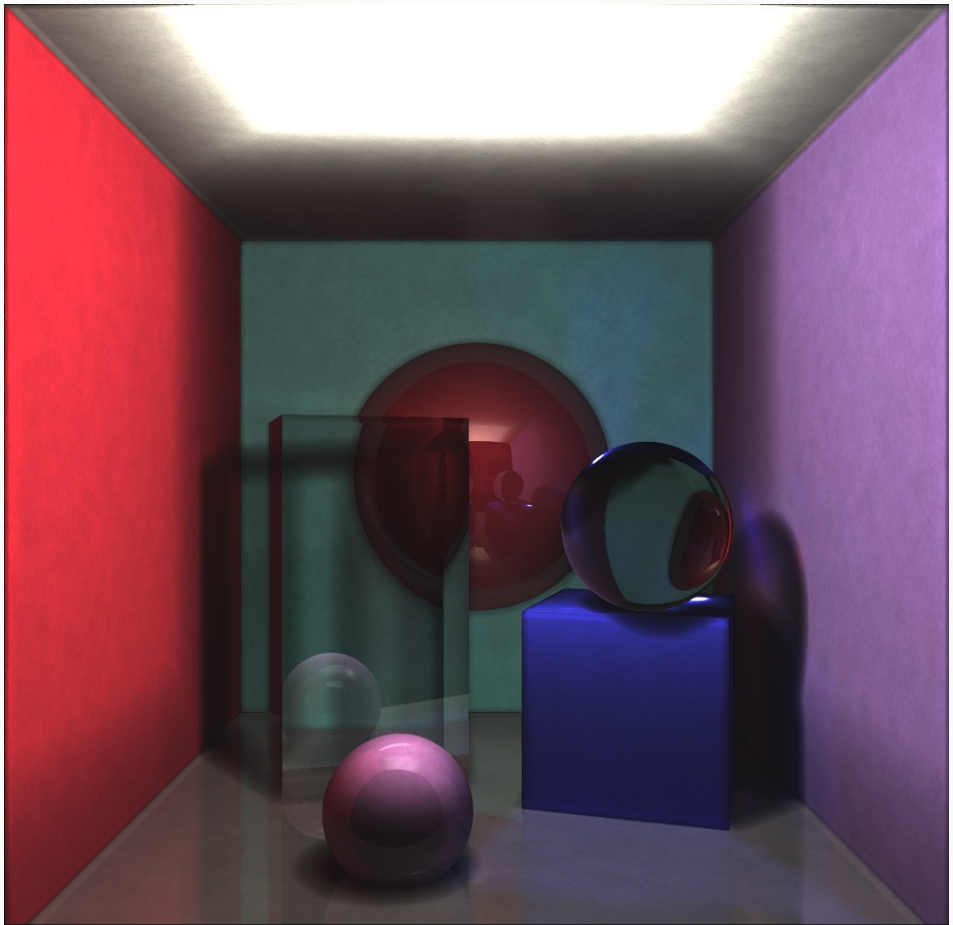
Viva

- tick that you have done the work!
- make sure that I have understood what you have done
- get feedback on your coursework

Viva

- tick that you have done the work!
- make sure that I have understood what you have done
- get feedback on your coursework
- get a mark

Pretty Pictures



Callum Pearce & James Tait: 15 million traced photons, 2500 gathered, 1000x1000

Next Time

Lecture Friday 5th of April

- Simon will talk about Geometry

Lecture Friday 10th of May

- Final lecture
- What to do next
- Thesis work in Computer Graphics
- Work in Computer Graphics

eof