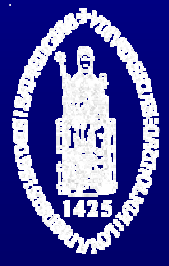
Course 38: A Practical Guide to Global Illumination using Photon Mapping

(A Doug James modified version of…)

Visual Importance and the Photon Map

Frank Suykens

K.U.Leuven, Belgium



What is Visual Importance ?

- Parts of the scene important for a certain view

View

Illumination (photon map)

Importance

Why use Visual Importance?

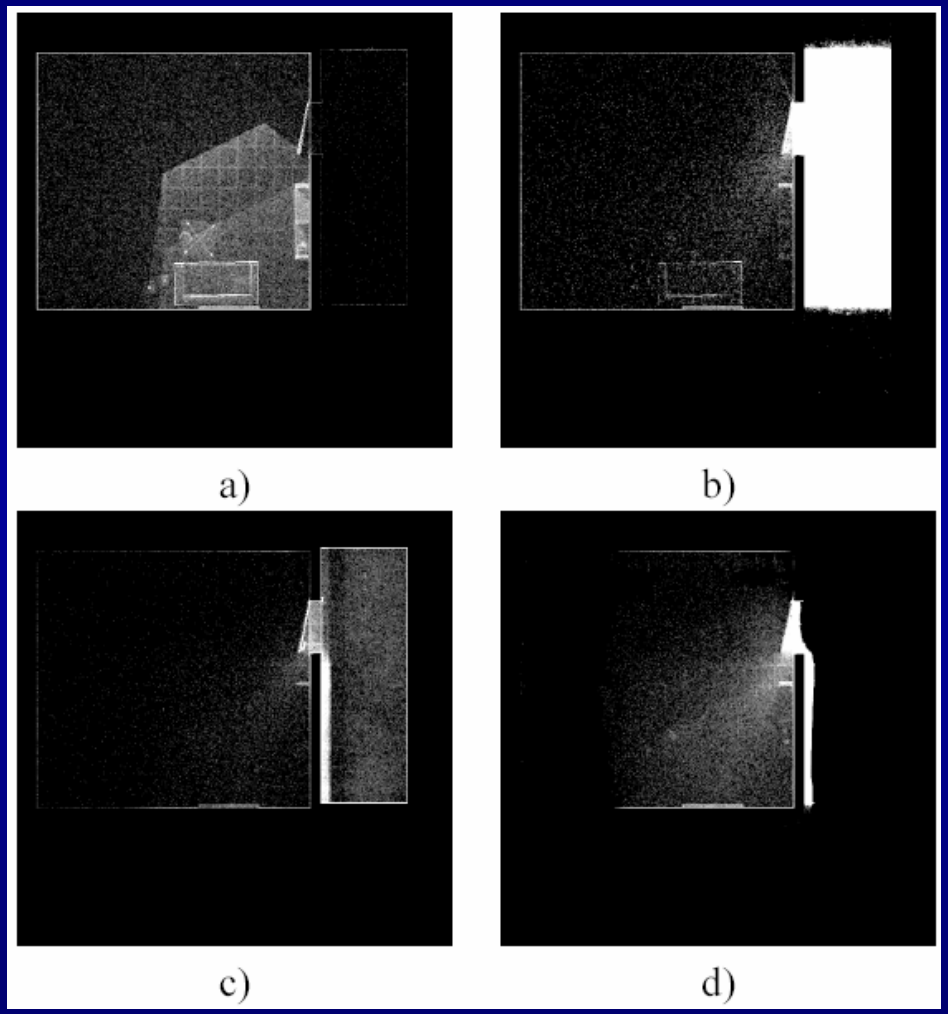


Figure 4: The very problematic setting of a room being lit by a door slit seen from atop. The room on the left has no light sources but the camera within; the room on the right contains one light source. a) The importance distribution, b) the standard photon map, c) the unbiased importance driven photon map, and the biased version using a minimum cutoff probability ϵ.

Keller & Wald 2000

Goal

- Importance driven algorithms:

Use importance to optimize storage and computation (view dependent)

- Previous work:
  - Peter ’98, Suykens ’00, Keller ’00, Christensen ‘01

Overview

- Algorithm:

```
- compute importance maps & required density
- while (photons to trace)
      for each photon hit :
        if (current density(pos) < required(pos))
              store photon
        else
              distribute photon power
- rendering pass ⇒ image
```

Overview

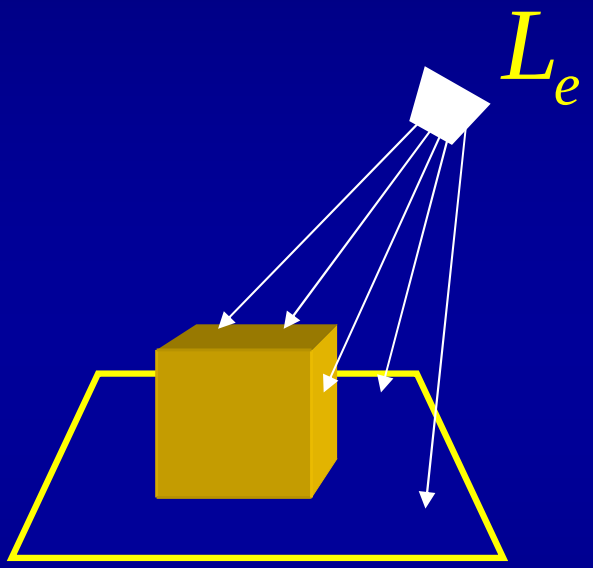
- Algorithm:

- compute importance maps & required density
- *while* (photons to trace)
  - for each p
    - *if* (curre … s))
      - stor
    - *else*
      - distribute photon power
- rendering pass ⇒ image

- Importance math & physics
- Importance maps
- Required density
- Alternative: Path Differentials

Importance: Math & Physics

- Importance = **dual of light**

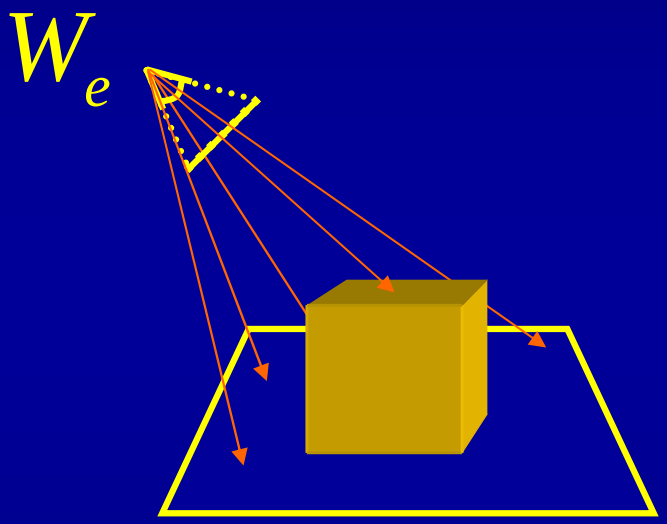


Radiance:

$$L(x,\vec{\omega}) = L_e(x,\vec{\omega}) + \int_{\Omega_x} L_i(x,\vec{\omega}')\, f_r(x,\vec{\omega},\vec{\omega}')\cos(n_x,\vec{\omega}')d\omega'$$

Importance: Math & Physics

- Importance = **dual of light**



'Potential' (adjoint equation):

$$W(x,\vec{\omega}) = W_e(x,\vec{\omega}) + \int_{\Omega_x} W_i(x,\vec{\omega}')\, f_r(x,\vec{\omega},\vec{\omega}') \cos(n_x,\vec{\omega}')\, d\omega'$$

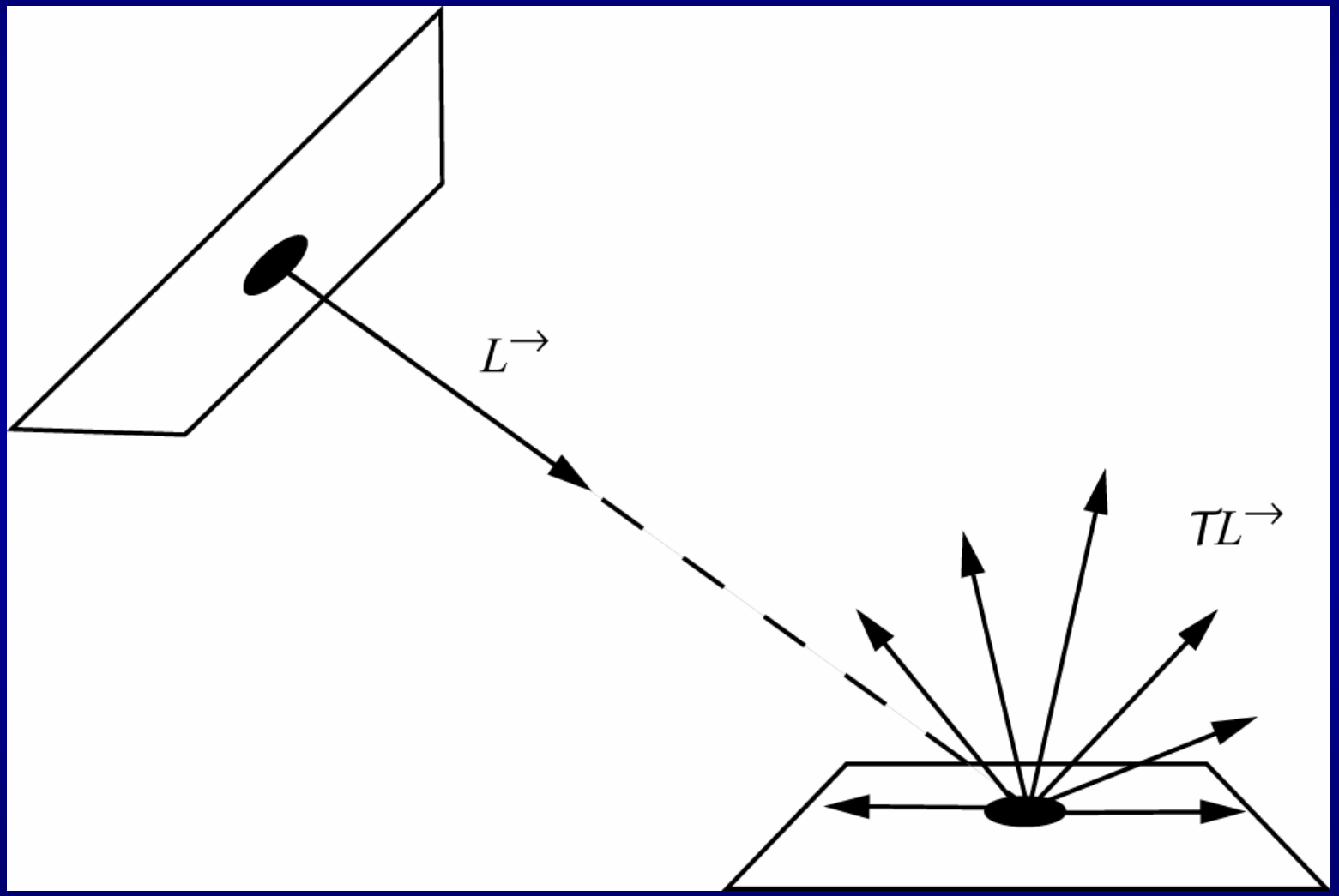
Importance: Math & Physics

- Importance transport == Light transport

Quantities	Importance	Light
	Potential ‘*W*’	Radiance ‘L’
	(Incoming) Importance ‘Γ’	Irradiance ‘E’
	Importance Flux ‘ψ’	Flux/Power ‘Φ’

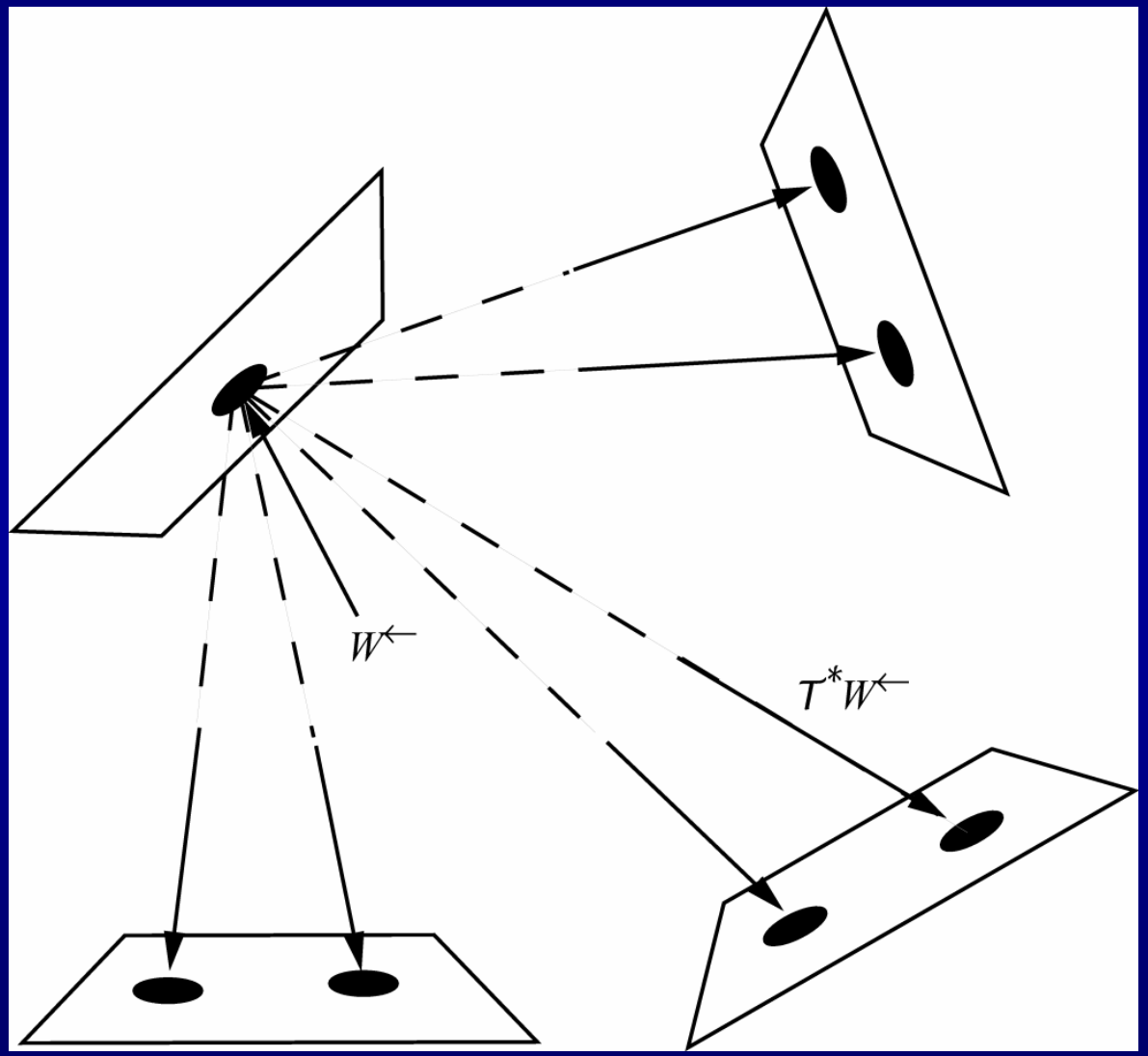
Propagation of an exitant function

(from [Dutre et al.])

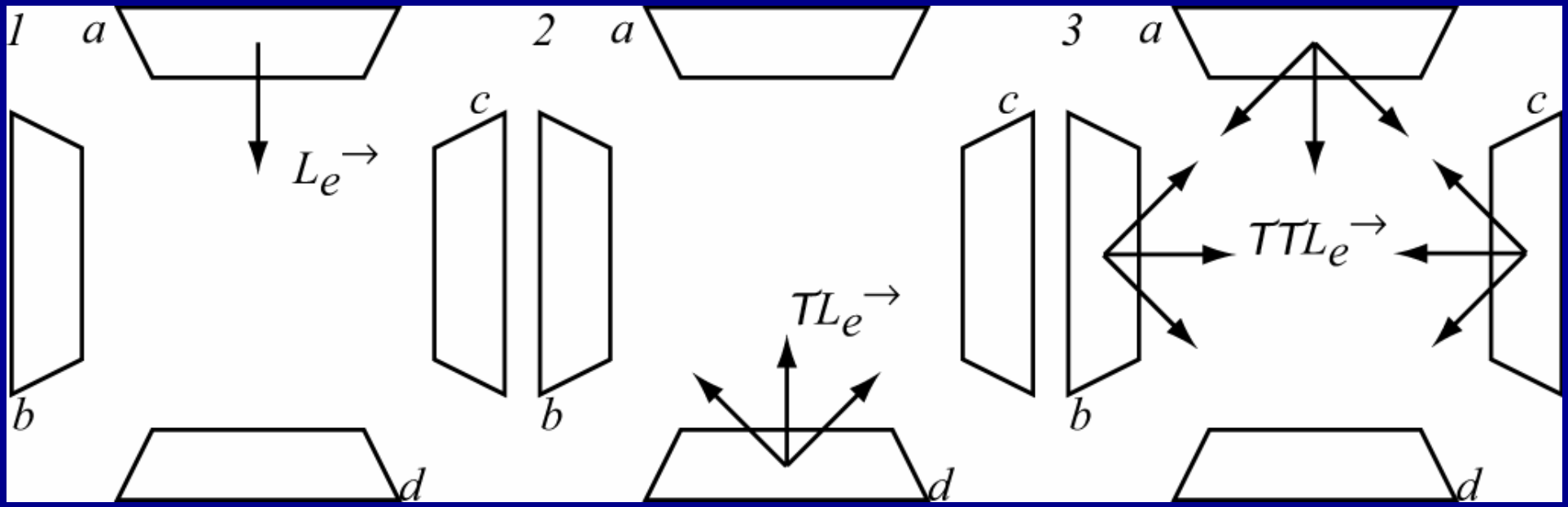


Propagation of an incident function

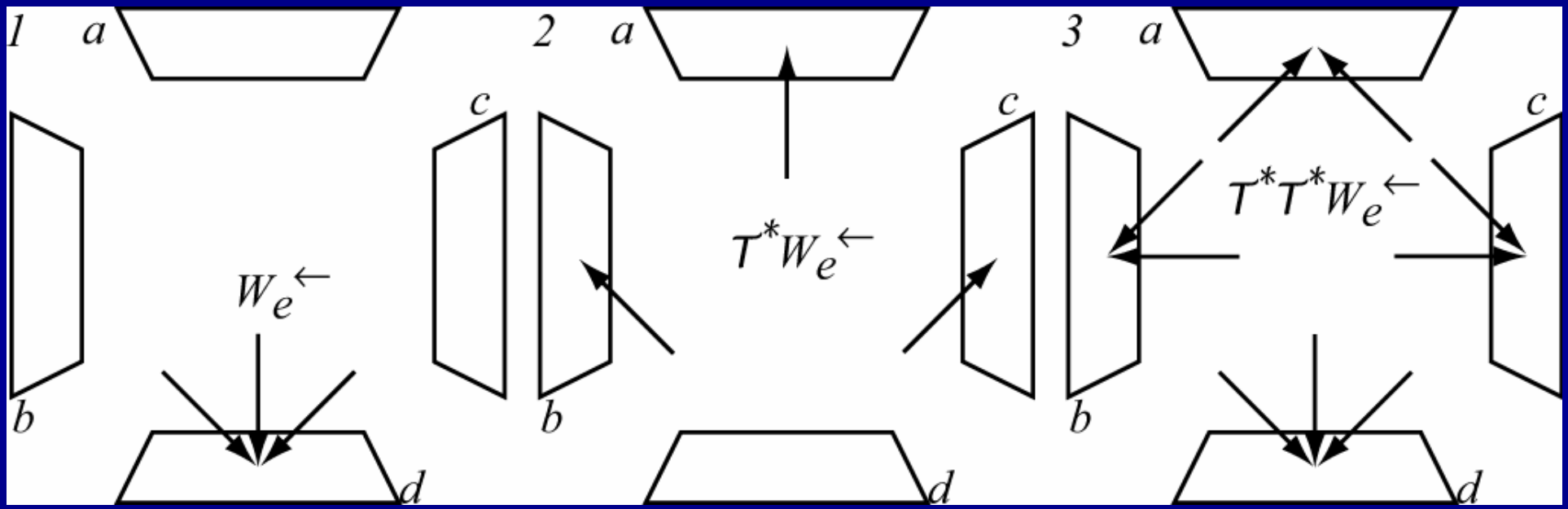
(from [Dutre et al.])



Propagation of an initial radiance distribution (from [Dutre et al.])

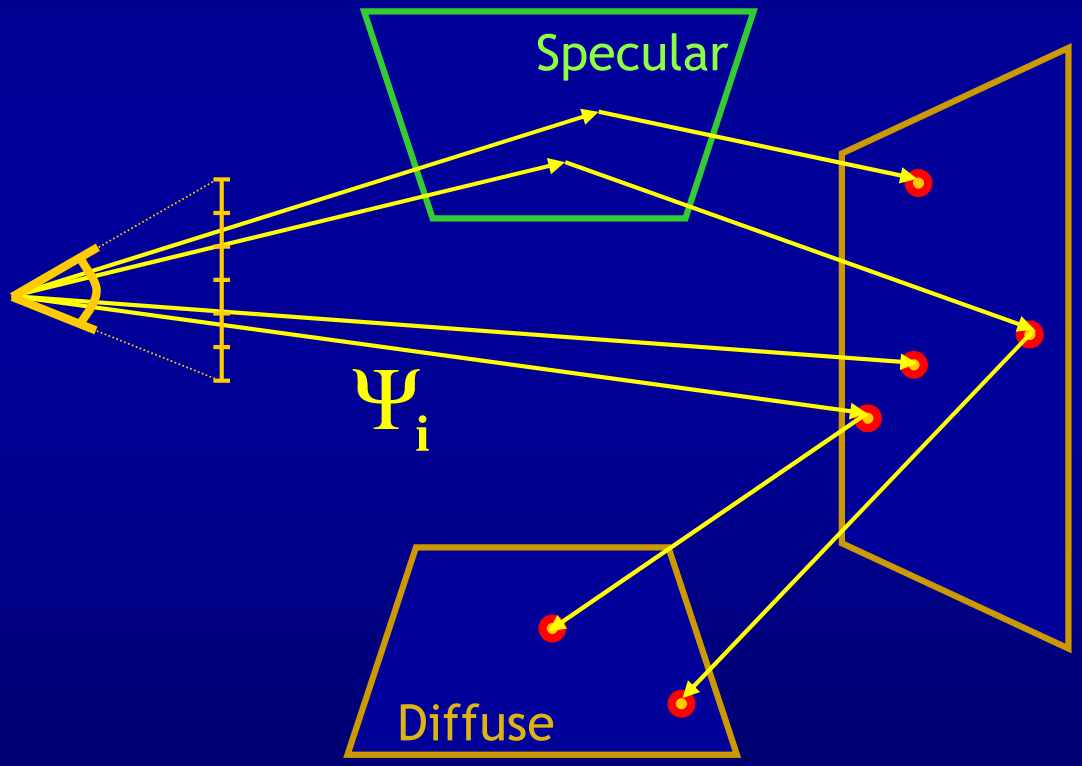


Propagation of an initial importance distribution (from [Dutre et al.])



Importance Maps

- Very similar to constructing photon maps
- Shoot ‘importons’, store on non-specular objects

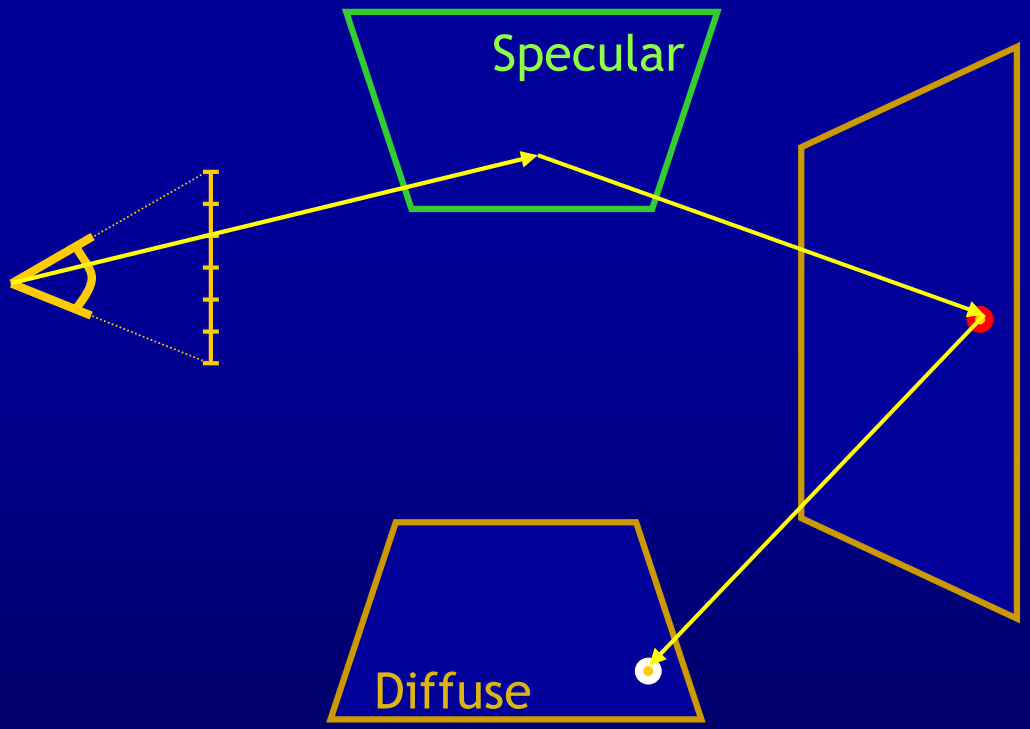


Importon flux:

$$\Psi_i = \frac{\Psi_{screen}}{N_{importons}}$$

Importance Maps

- Two photon maps ⇒ Two importance maps
  - Caustic map : direct visualisation
  - Global map : indirect visualisation (final gather)



Would read caustic map ⇒

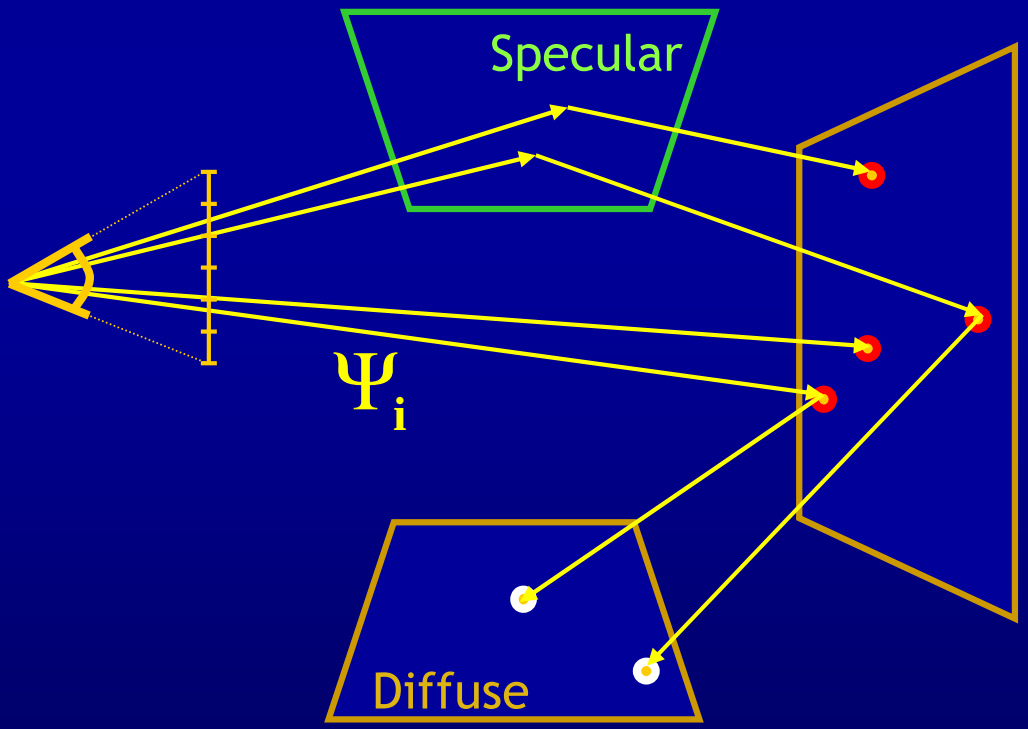
Caustic importance map

Would read global map ⇒

Global importance map

Importance Maps

- Two photon maps ⇒ Two importance maps
  - Caustic map : direct visualisation
  - Global map : indirect visualisation (final gather)



Caustic Importance map

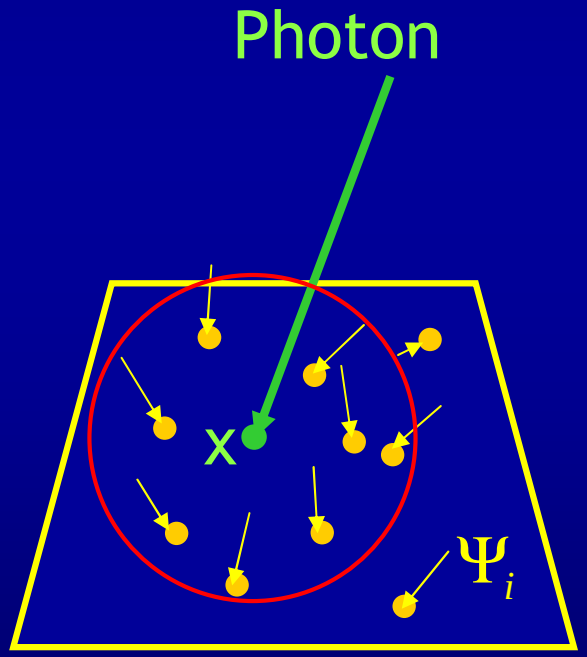
$E(S^*)D$

Global Importance map

$E(S^*)D(S^*)D$

Importance Maps

- Importance reconstruction: Γ

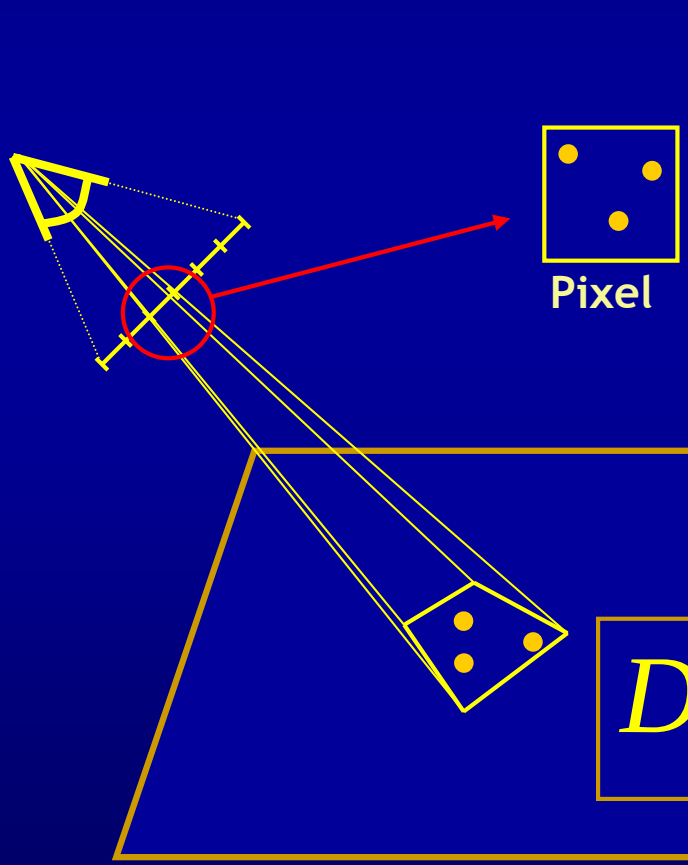


Find M nearest importons :

$$\Gamma(x) \approx \frac{\sum_{i=1}^{M} \Psi_i}{\pi r_M^2(x)}$$

Required Density

- High importance ⇒ High density D_r
- Heuristic : Linear relationship ($D_r = c.\Gamma$)
- Choose density per pixel ⇒ Density in scene



$$\Gamma(x_{pix}) = 1$$

$$D_r(x_{pix}) = M_r / A_{pix}$$

User choice

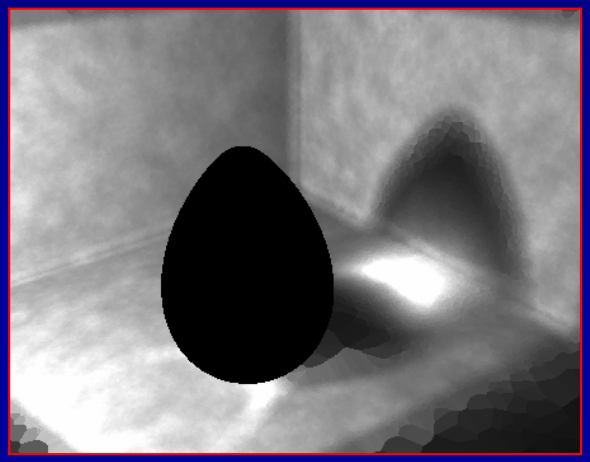
$$D_r(x) = \Gamma(x) M_r / A_{pix}$$

Required Density

$$D_r(x) = \Gamma(x) M_r / A_{pix}$$

- Caustic Map: M_r = 10 – 30
- Global Map: M_r = 1 – 2

- $\Gamma(x)$ requires importance map lookup: balanced kd-tree



Caustic Importance map

Overview

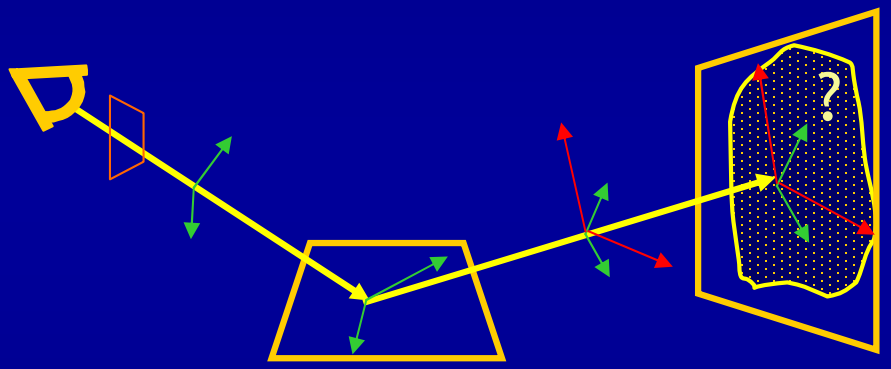
- Algorithm:

- compute importance maps & required density
- *while* (photons to trace)
  - for each p
    - *if* (curre … s))
      - stor
    - *else*
      - distribute photon power
- rendering pass ⇒ image

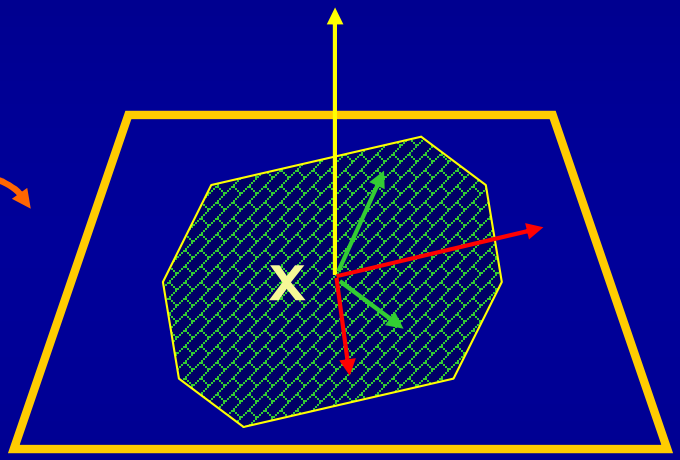
- Importance math & physics
- Importance maps
- Required density
- Alternative: Path Differentials

Importance Map Alternative

- Path differentials : Trace ‘footprint’ of a pixel (or region of influence of a path)



Partial derivatives
(pixel, BRDF sampling)

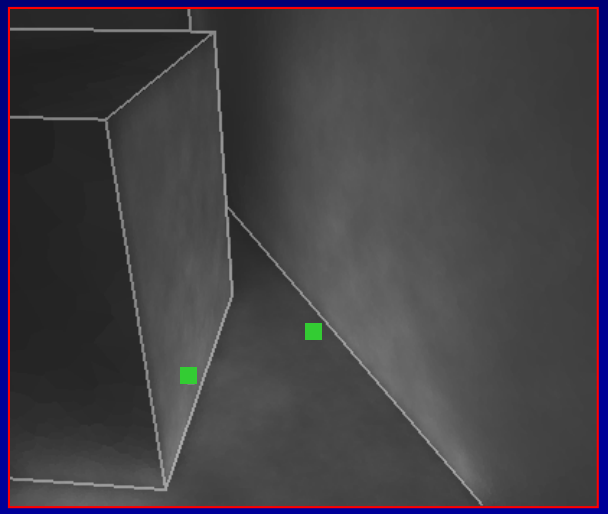


Derivatives to footprint

$$\Gamma_{pix}(x) \sim \frac{1}{A_{footprint}}$$

- Suykens, EGWR ’01
- Igehy, Siggraph ‘99

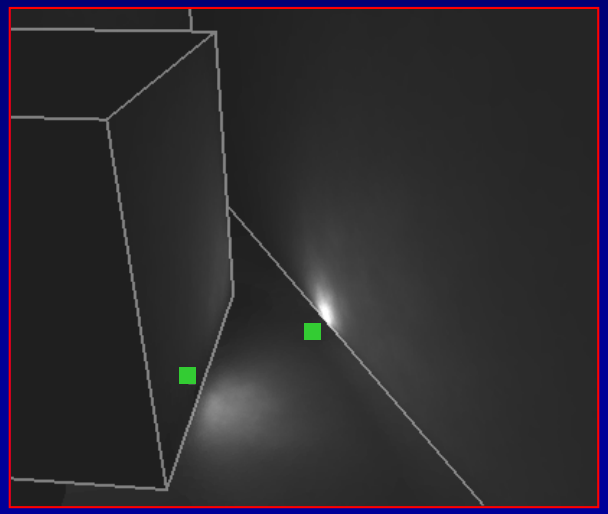
Pixel vs. Screen Importance



Screen importance

= importance map

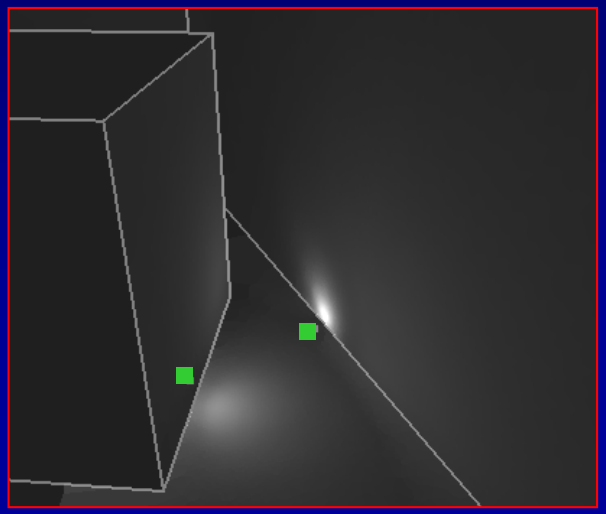
Bound on screen error



Pixel importance

= importance map per pixel

Bound on pixel error



Path differentials

= importance from a single path

!

Overview

- Algorithm:

```
- compute importance maps & required density
- while (photons to trace)
      for each photon hit :
        if (current density(pos) < required(pos))
              store photon
        else
              distribute photon power
- rendering pass ⇒ image
```

Photon Map Construction

- Shoot photons as usual
- For each hit :

  compare current & required density

$$D_{current}(x) < D_{required}(x)\,?$$

Yes → Store

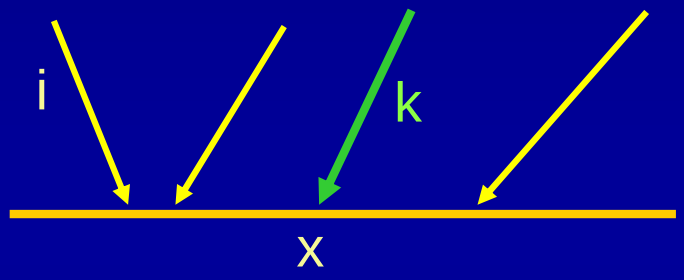
No → ***Redistribute***

$$D_c(x) = \frac{M}{\pi r_M^2(x)}$$

Photon Map lookup: unbalanced kd-tree

Power redistribution

Photon ‘k’ arrives at ‘x’:

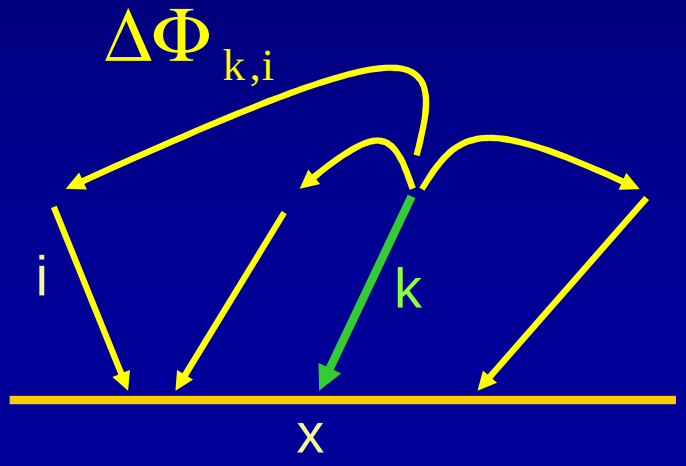


Reconstruction in ‘x’, M nearest photons + photon ‘k’:

$$\tilde{L}_r(x,\omega) = \frac{\sum_{i=1}^{M} \Phi_i \cdot f_r(x,\omega_i,\omega) + \boxed{\Phi_k \cdot f_r(x,\omega_k,\omega)}}{\pi r_{M+1}^2(x)}$$

Power redistribution

Photon not stored:



Reconstruction in ‘x’ after distribution:

$$\tilde{L}_r(x,\omega) = \frac{\sum_{i=1}^{M} f_r(x,\omega_i,\omega)(\Phi_i + \Delta\Phi_{k,i})}{\pi r_M^2(x)}$$

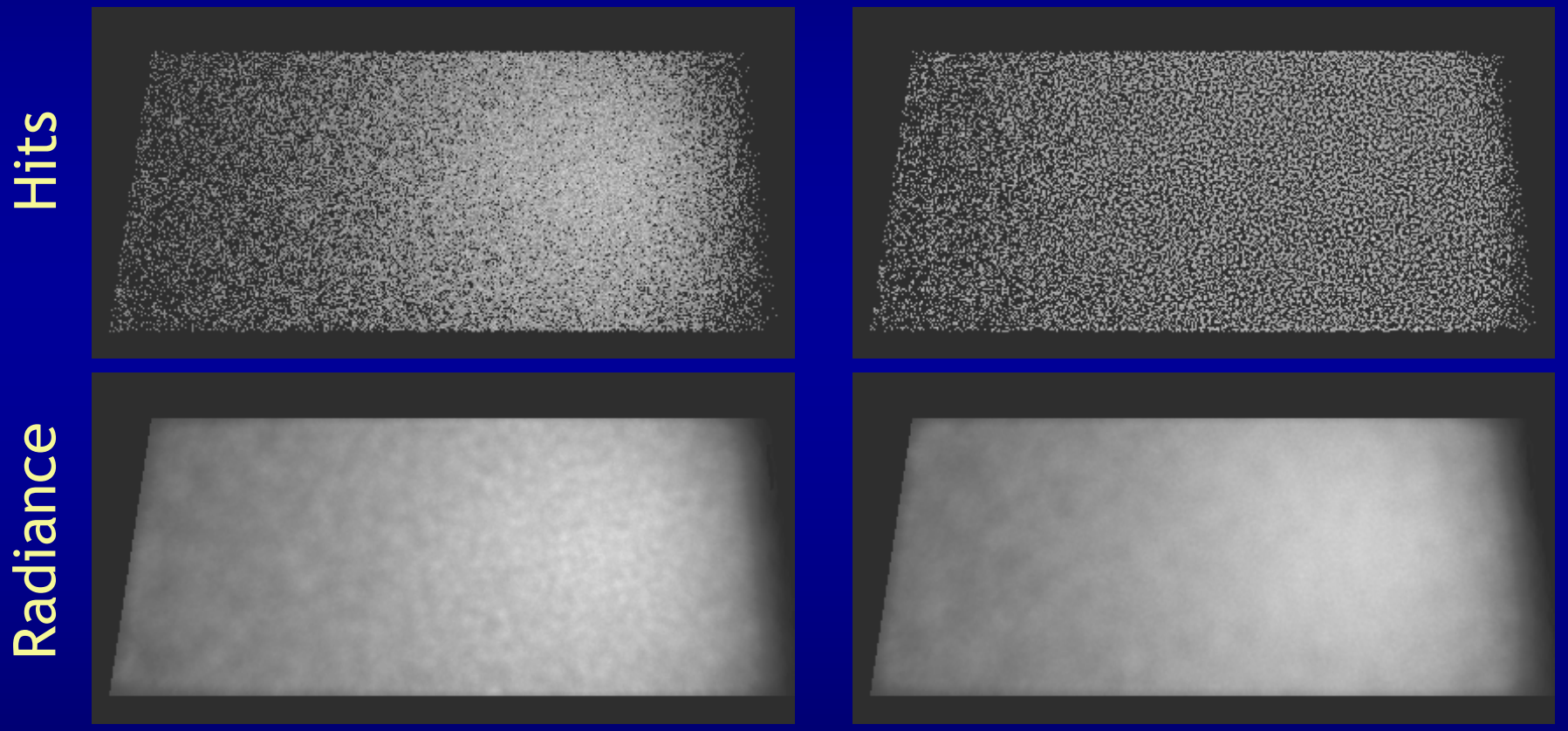
Power redistribution

- Choice for $\Delta\Phi_{k,i}$ can be based on:
  - distance to ‘x’
  - photon direction
- $\Delta\Phi_{k,i} = \Phi_k \;/\; M'$ for M’ photons i that contribute in x (cosine w. normal > 0)

  (Diffuse: equal reconstruction in ‘x’)
- Extra bias (splatting), but current density high enough + M small (± 20)

Power redistribution

- Result (D_r constant)



Overview

- Algorithm:

- compute importance maps & required density
- *while* (photons to trace)

    for each photon hit :
      if (current density(pos) < required(pos))
          store photon
      else
          distribute photon power

- rendering pass ⇒ image

Rendering pass

- Redistribution maintains energy balance
- Photon powers may differ, but
  - Gradual change (homogeneous map)

$\Rightarrow$ **No change in rendering pass**

- But could use importance ?!?!

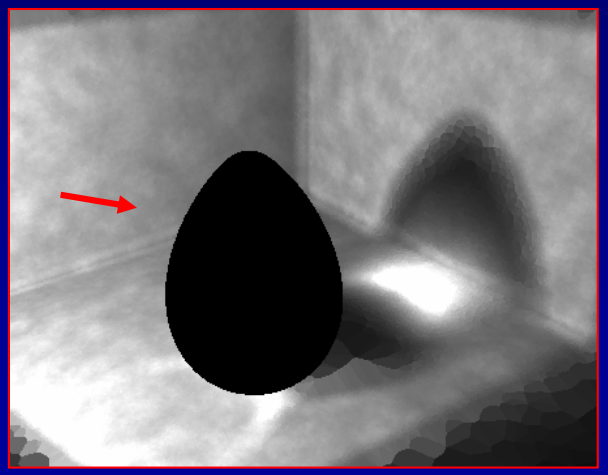
Results

Results: Caustic Map

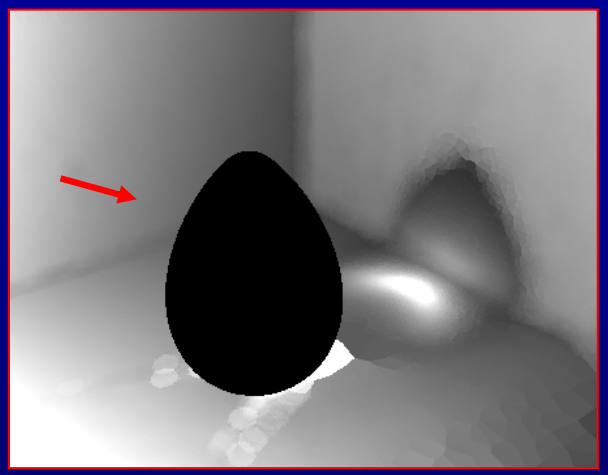
Final Image

Results: Caustic Map

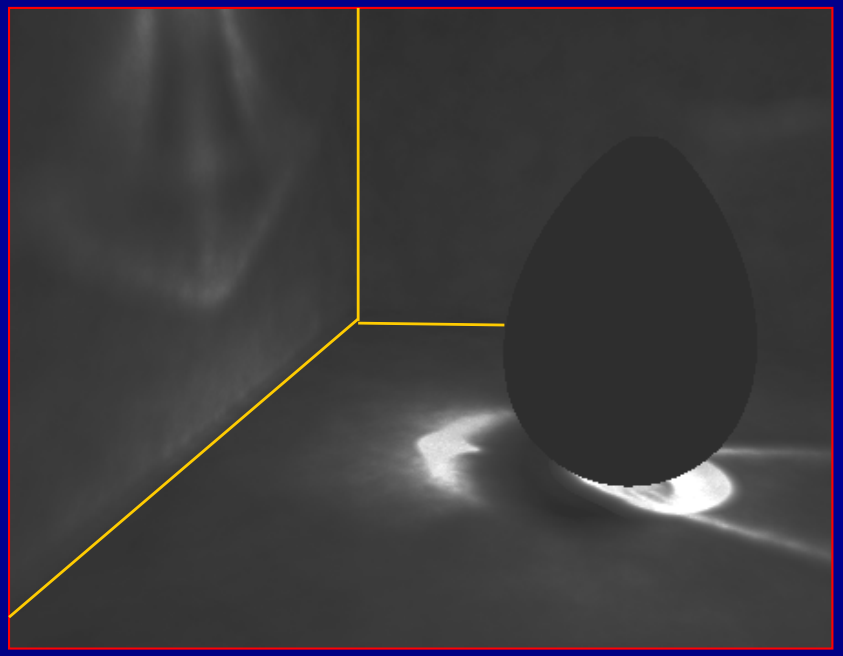
Required (maximum) density



Importons



Path differentials



Caustic map density

(200k vs. 400k photons)

Results: Global Map

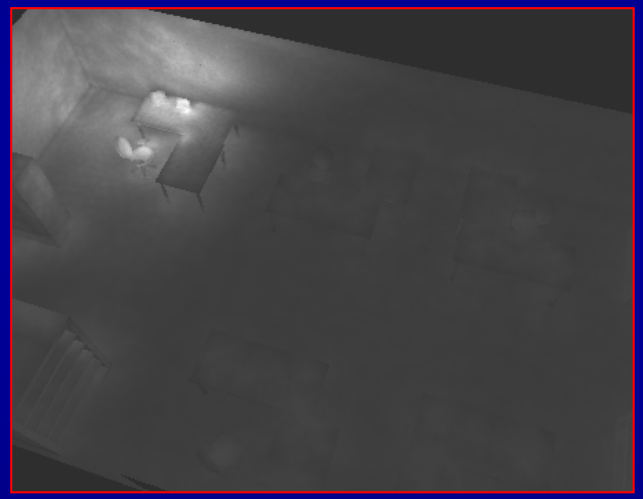
80.000 importons

Required density

400.000 photons

Photon map density (normal)

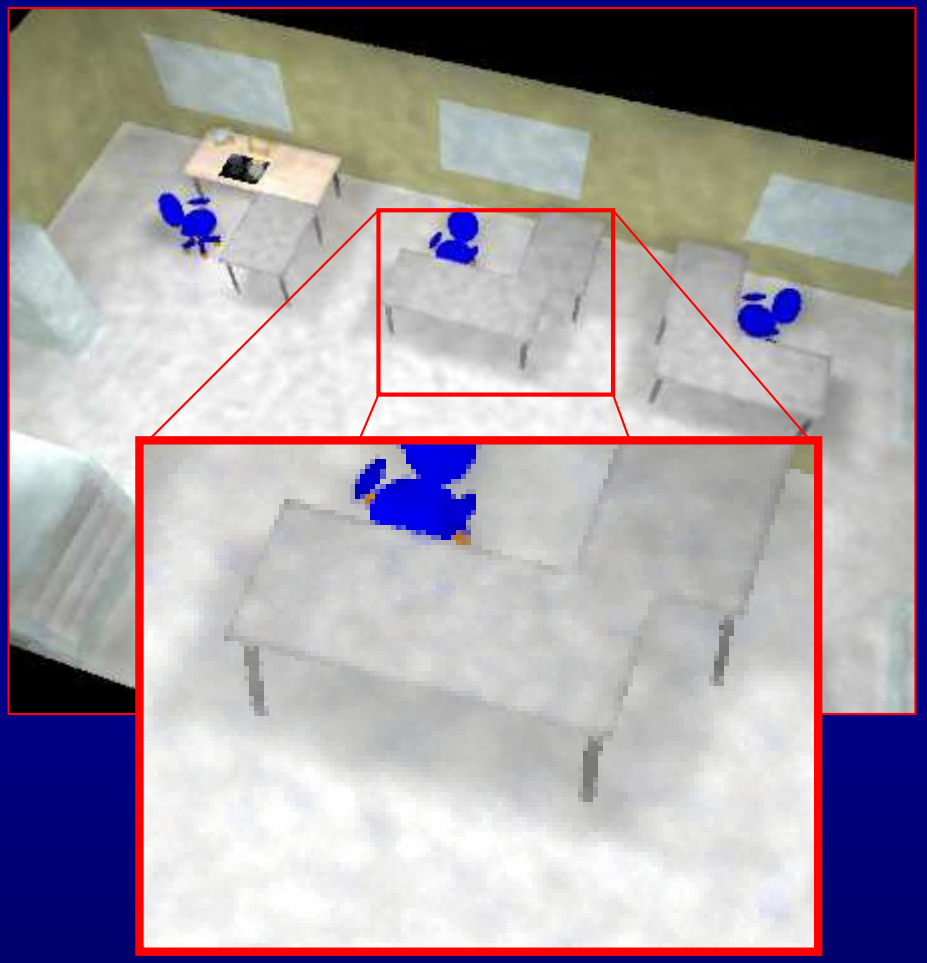
57.000 photons



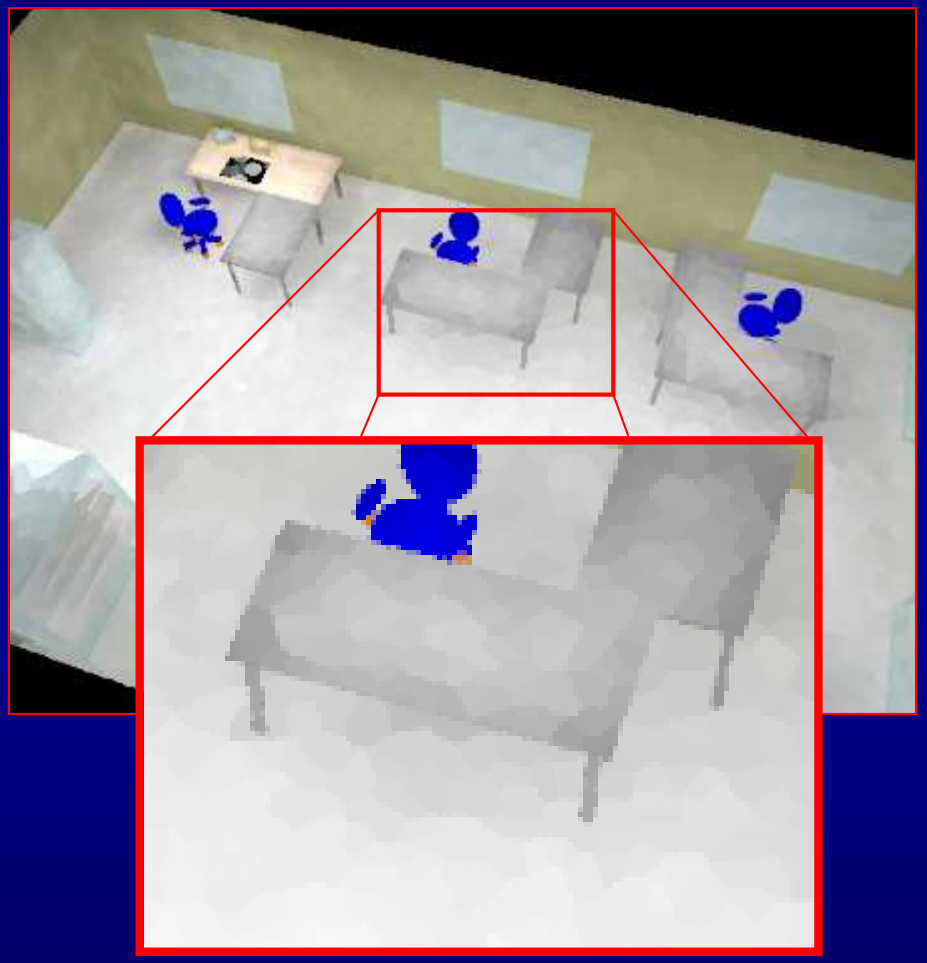
Photon map density (importance driven)

Results: Global Map

Global map, direct visualisation (no imp)



Global map, direct visualisation (imp)



Results: Global Map

Standard

Importance driven

Conclusions

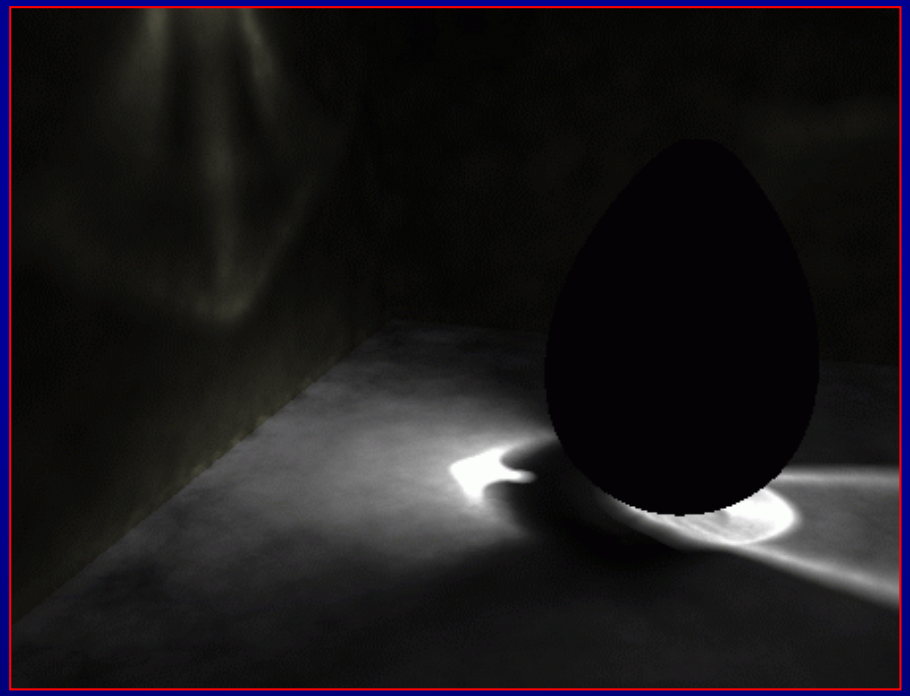
- Visual importance

  $\Rightarrow$ More compact photon map
- Redistribution: number of photons limited
  - Trace until difficult region ok
  - (Arbitrary memory gain)
- Steps towards automatic ‘error control’

- BUT: Still a lot to find out…

Future work

- Required density: dependent on ‘other’ illumination ?

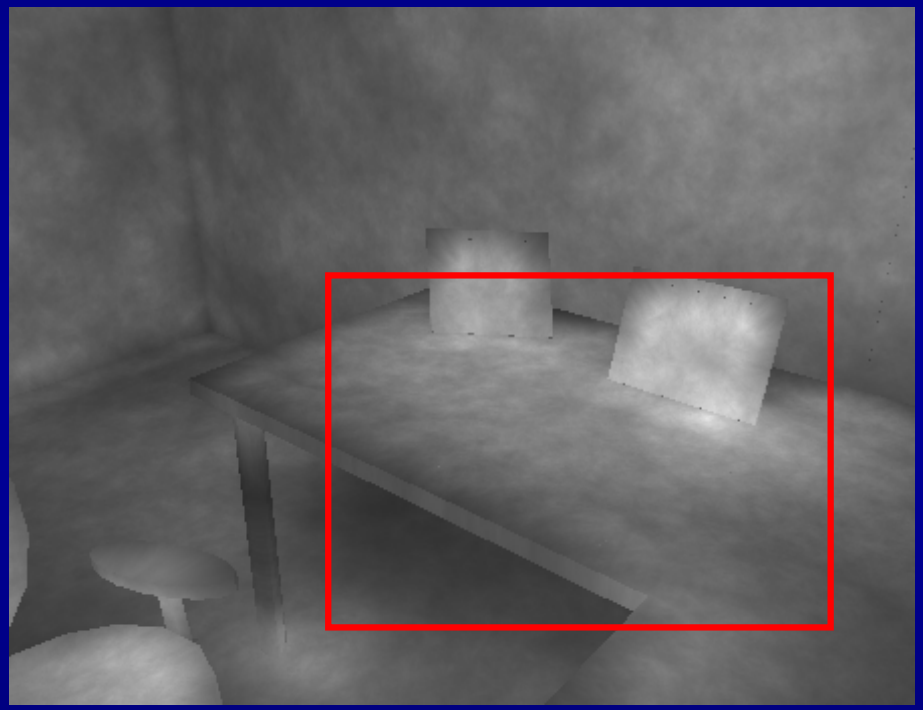


Caustic Map

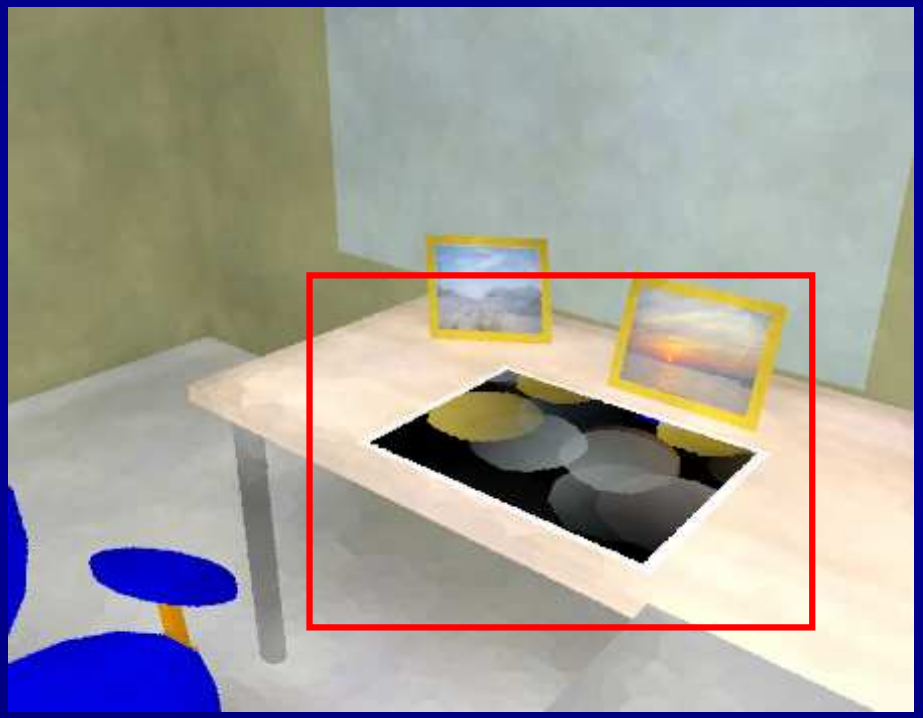
All illumination

Future work

- Required density: take glossiness into account



Global map density



Global map radiance

Directional importance ?

Future work

- *Shoot* fewer photons (homogeneous map!)
  - Per Christensen
- How many nearest photons ?
- Participating media
- …