Photon Mapping and GPU

Towards Real Time ?

Fabien Lavignotte et Mathias Paulin

Équipe Synthèse d'Images et Réalité Virtuelle

IRIT-UPS-CNRS UMR 5505

Université Paul Sabatier

Toulouse - France

Real Time Lighting Simulation

- Software/cluster approach :
  - Massively parallel implementation.
  - Wald et al., 2003, Interactive Global Illumination in Complex and Highly Occluded Environments

Real Time Lighting Simulation

- Software/cluster approach :
  - Massively parallel implementation.
  - Wald et al., 2003, Interactive Global Illumination in Complex and Highly Occluded Environments
- Hardware approach :
  - Specialized hardware : Schmittler et al., 2002, SaarCOR - A Hardware Architecture for Ray Tracing
  - Using GPU : Purcell et al., 2003, Photon Mapping on Programmable Graphics Hardware

Real Time Lighting Simulation

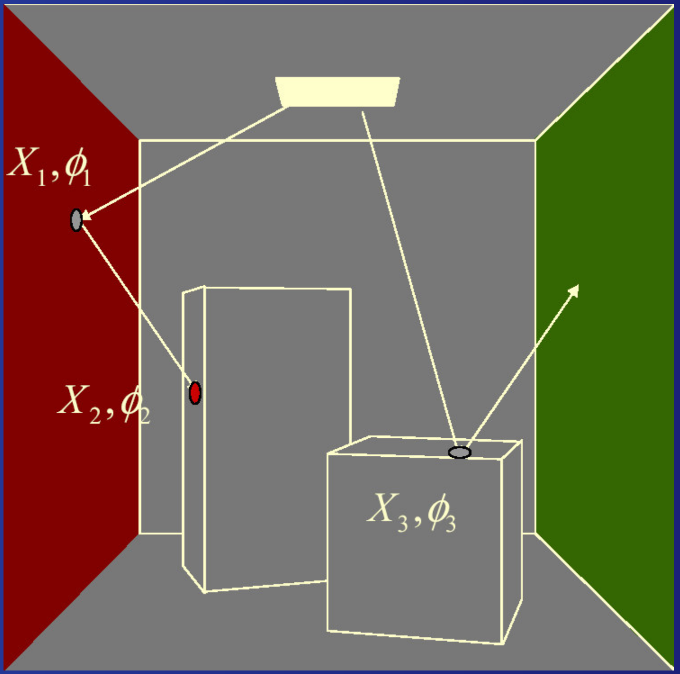
- Our approach : Photon Mapping and IBR
  - Photon tracing
    - Monte Carlo numerical integration
    - Error as variance/bias
    - Error control simple

Real Time Lighting Simulation

- Our approach : Photon Mapping and IBR
  - Photon tracing
    - Monte Carlo numerical integration
    - Error as variance/bias
    - Error control simple
  - "Image Based" Rendering
    - Low scene complexity dependence.
    - CPU-GPU Parallelism.
    - Goal : Real Time.

Photon Mapping

Density estimation and global illumination :



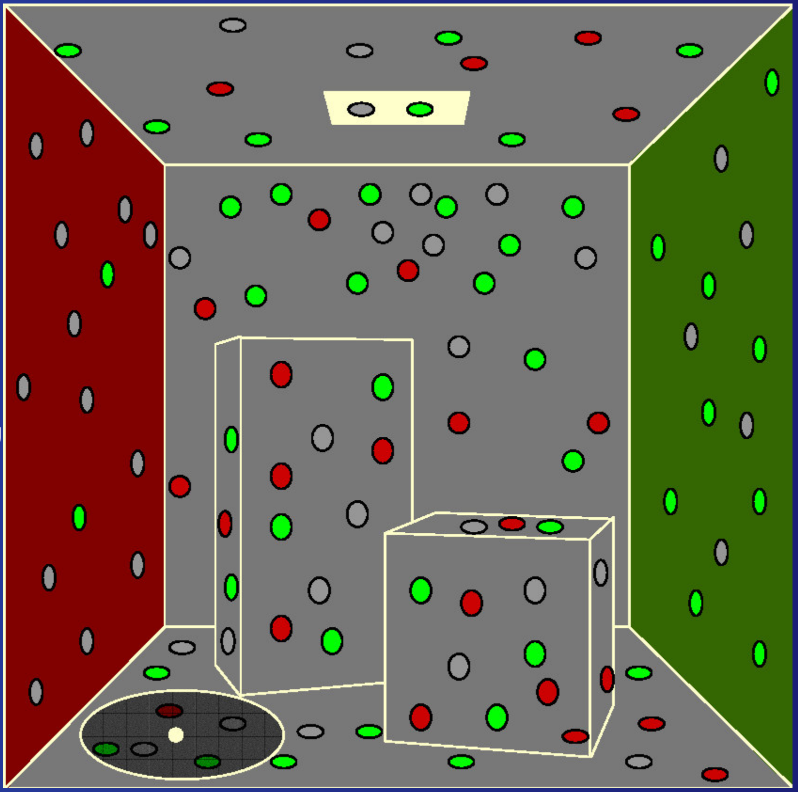
- Stochastic process
  - Path tracing
- Observed data
  - Photons
- *Photons*
  - Density proportional to irradiance.
  - $E(x) = \Phi_T f(x) \Rightarrow \widehat{E}(x) = \sum_1^n \phi_i K_h(x - X_i)$

Irradiance estimation

- Direct method : per measurement point
  - k photons h-nearest neighbors

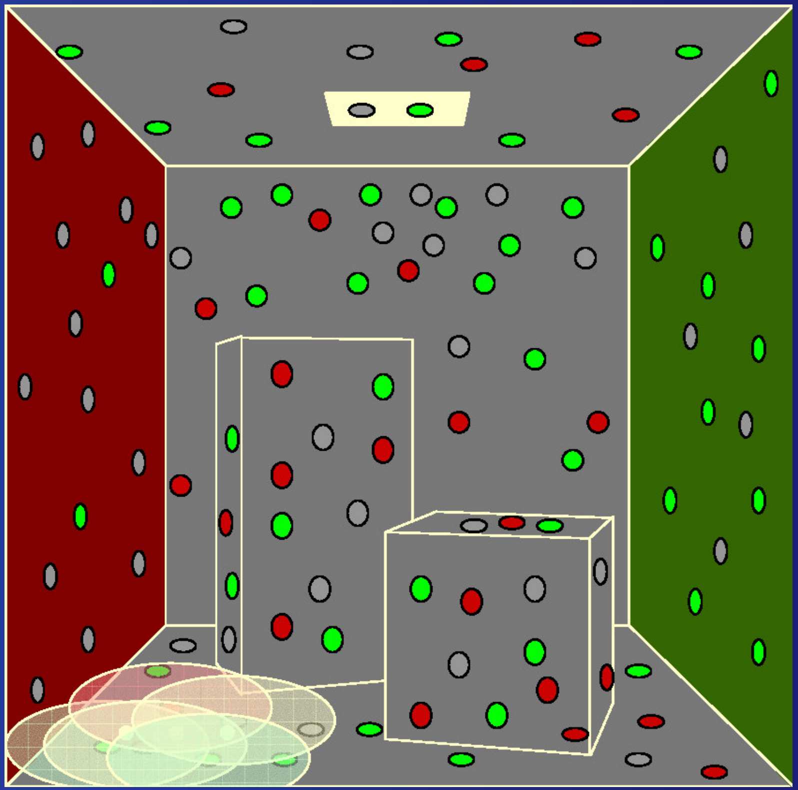
$$\widehat{E}(x) = \sum_{1}^{k} \phi_i K_h(x - X_i)$$

Random access to photons, low performance.



Irradiance estimation

- Dual approach : by photons
  - k measurement point h-nearest neighbors



$$C_i(x) += \phi_i K_h(x - X_i)$$

Sequential access,

high performance.

Photon Splatting

- Image based density estimation.
  - Hardware assisted estimation.
  - Dual approach by pixel.

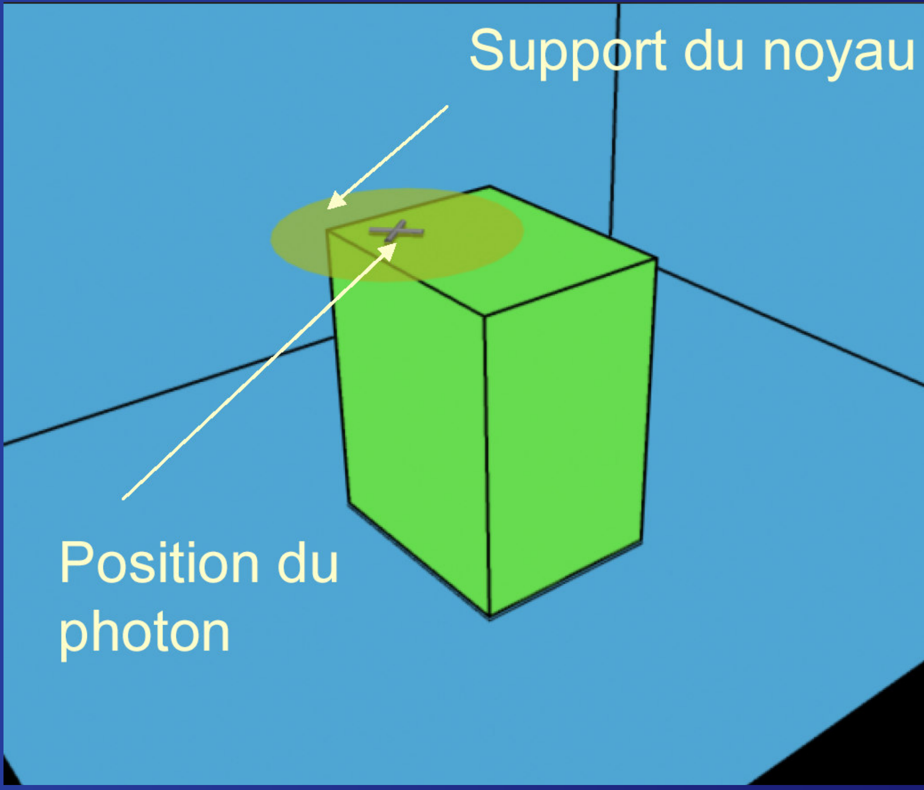
Photon Splatting

- Image based density estimation.
  - Hardware assisted estimation.
  - Dual approach by pixel.

1. Surface identifiers buffer construction
   - Contributions restricted by surface.
2. Photon splatting
   - Contributions accumulated by pixel.

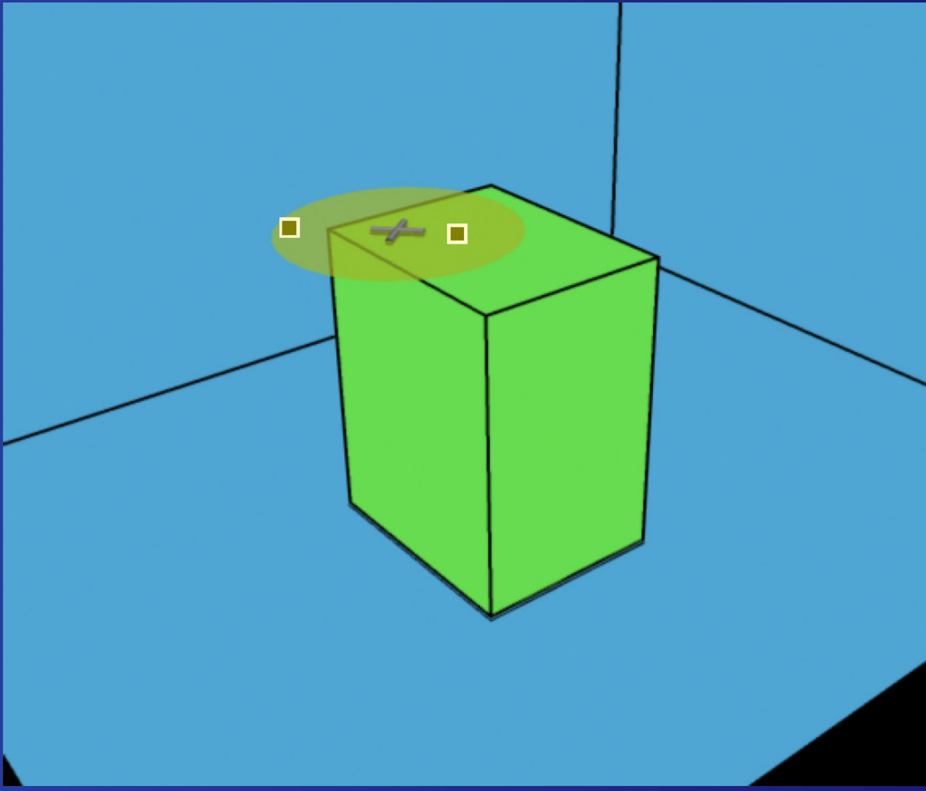
Photon Splatting

- Contributions restricted by surface.
  - Influenced pixels localization.
  - Support of the estimation kernel : disc.



Photon Splatting

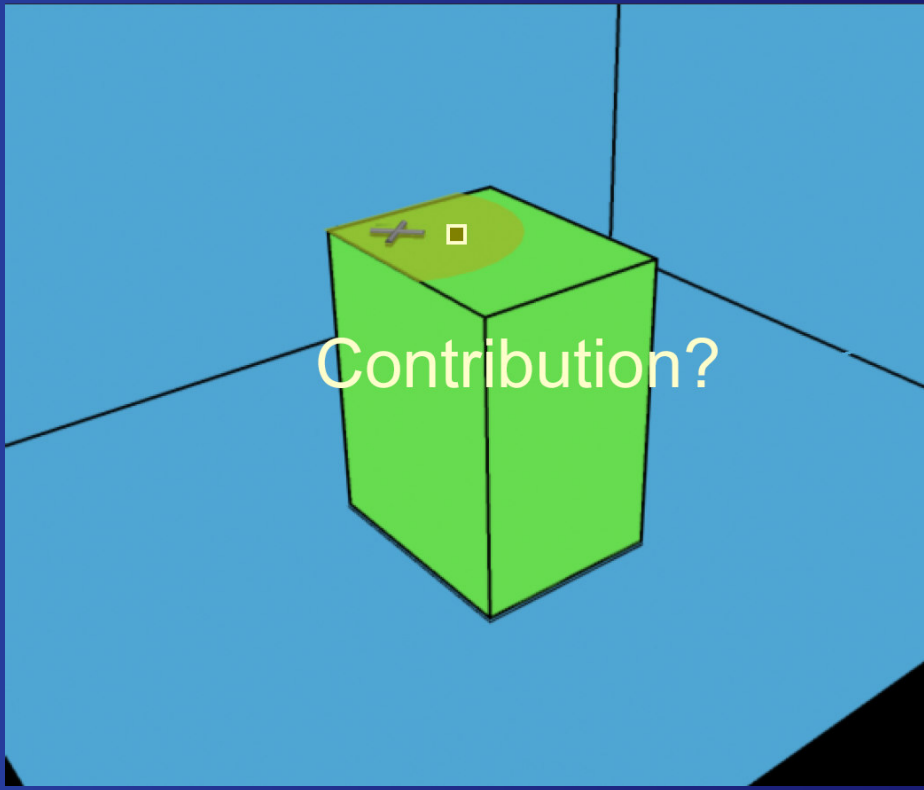
- Contributions restricted by surface.
  - Influenced pixels localization.
  - Support of the estimation kernel : disc.
  - Limit the rendering of disc.



Photon Splatting

- By pixel contributions accumulation.

$$C_i(x) = \phi_i \frac{1}{h^2} K(\frac{x - X_i}{h})$$

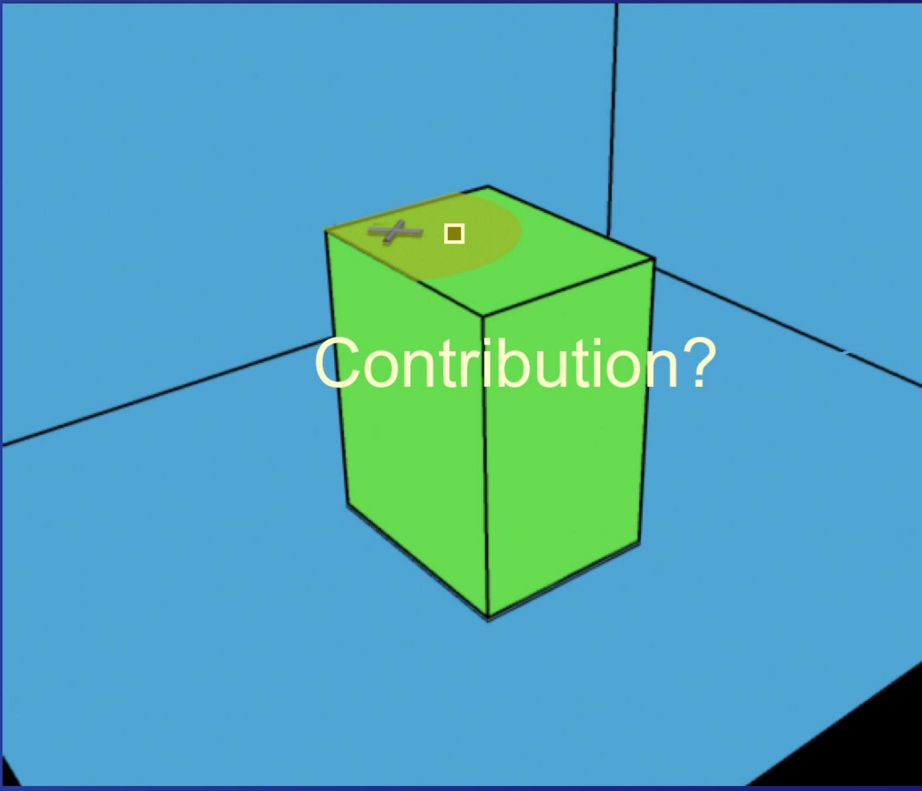


Photon Splatting

- By pixel contributions accumulation.

$$C_i(x) = \phi_i \frac{1}{h^2} K(\frac{x - X_i}{h})$$

  - contributions

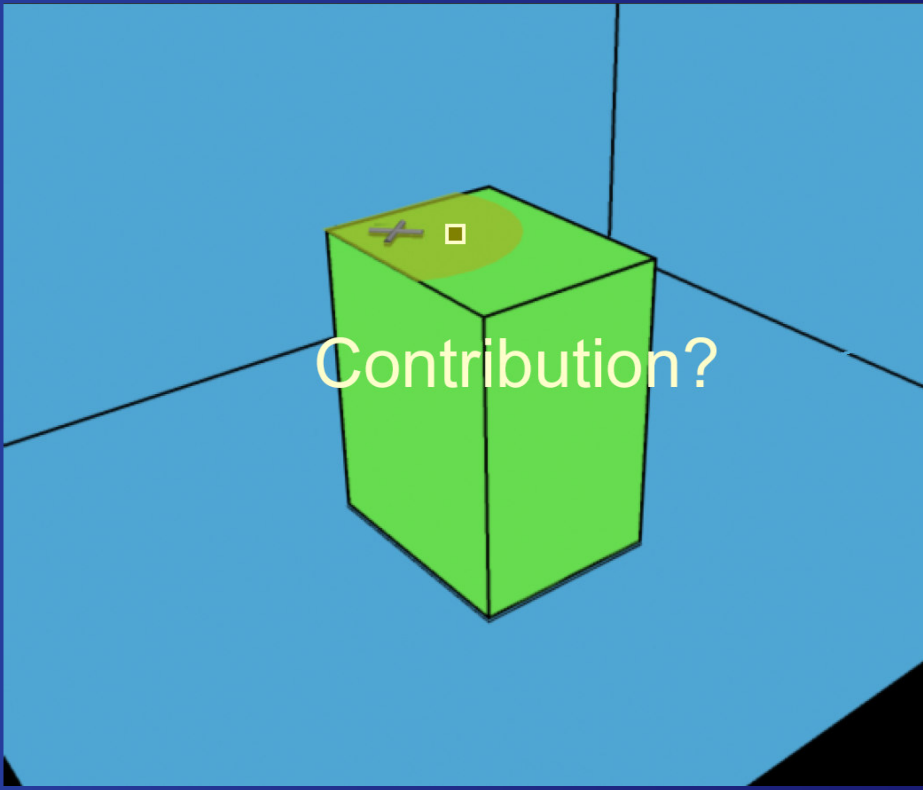


Photon Splatting

- By pixel contibutions accumulation.

$$C_i(x) = \phi_i \frac{1}{h^2} K\left(\frac{x - X_i}{h}\right)$$

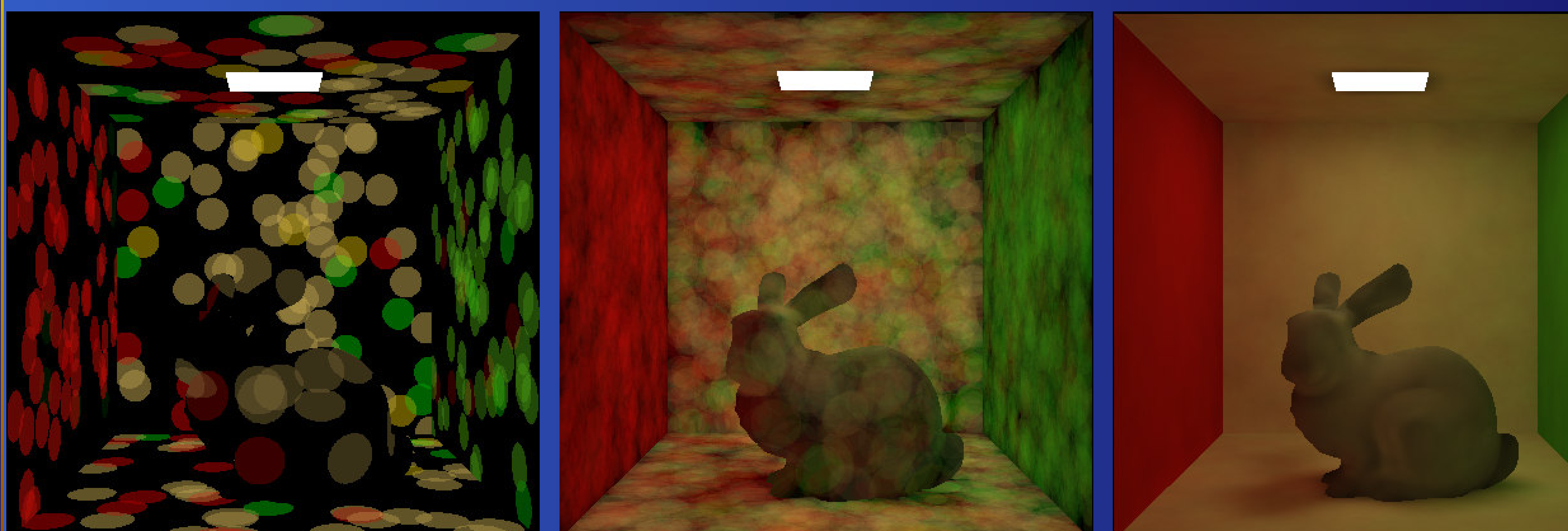
- Disc color.
- Texture.



Photon Splatting

- Image based density estimation
  1 000 000 photons.

0.05% 1% 100%



Photon Splatting

- Image based density estimation.
  - Boundary bias removal.

Origin :

Kernel support not included in the surface.

Photon Splatting

- Image based density estimation.
  - Boundary bias removal.

Solution :

$$C_i'(x) = \frac{\pi h^2}{A(S_p \cap D_{x,h})} C_i(x)$$

Photon Splatting

- Image based density estimation.
  - Boundary bias removal.

Solution :

$$C_i'(x) = \frac{\pi h^2}{A(S_p \cap D_{x,h})} C_i(x)$$

By pixel computation.

GPU constraints

- Surface identification.
  - Using stencil buffer ?
    + Simple test
    − 8 bits identifiers is insufficient
- Rendering surface identifier in a texture
  + 32 bits identifiers
  − fragment program based comparison

GPU constraints

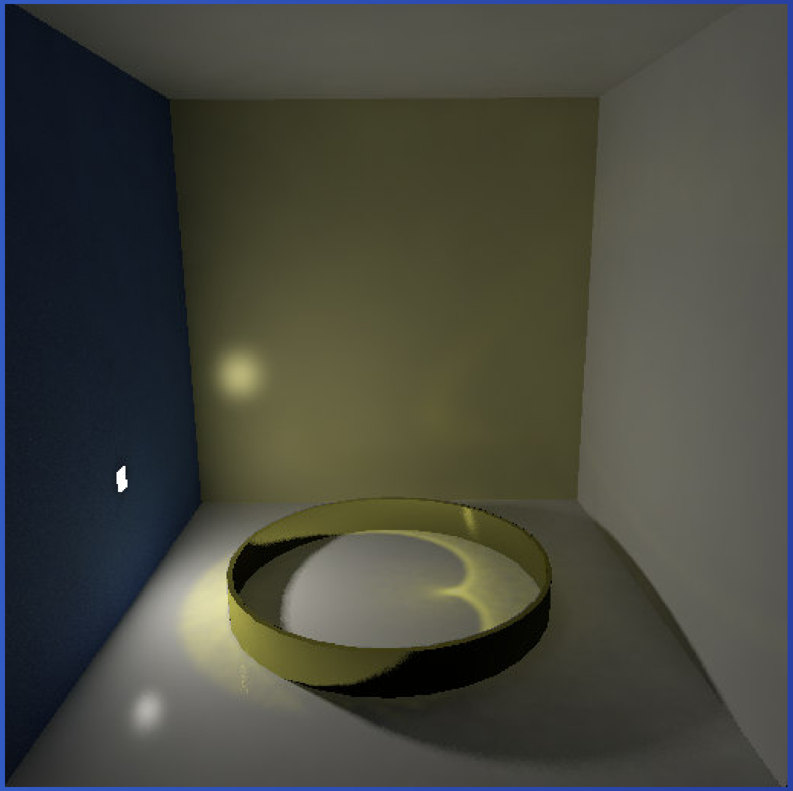
- By pixel accumulation.
  - Using blending ?
    - 8 bits precision is insufficient
    - No 32 bits blending
- Transform the problem to a photon counting one
    - Constant energy by photon
    - To be managed by pdf

GPU constraints

- boundary bias removal.
  - Triangle/disc intersection computation
    - By pixel expensive computation
    - No 32 bits blending
    - No cache beetween different fragments
- Not GPU implementable for the moment
    - Software implementation

Results

- Athlon XP2600+, GeForceFX 5800.



1500 triangles.

Path tracing :

2 M in 4.45s

Photon splatting :

0.74s

Results

- Athlon XP2600+, GeForceFX 5800.

100 000 triangles.

Path tracing :

2 M in 26.68s

Photon splatting :

0.76s

Real Time Photon splatting

- Efficient lighting model ...
  - Quantifiable error, variance/bias compromise ...

Real Time Photon splatting

- Efficient lighting model ...
  - Quantifiable error, variance/bias compromise ...
- Efficient algorithms ...
  - Low complexity.
  - Implementation quite easy.

Real Time Photon splatting

- Efficient lighting model ...
  - Quantifiable error, variance/bias compromise ...
- Efficient algorithms ...
  - Low complexity.
  - Implementation quite easy.
- Hardware still limiting !

References

Fabien Lavignotte PhD
Presented on July 2003
www.irit.fr/recherches/ESIRV/VIS/Photons