max planck institut informatik

EG 2007, 3.9.-7.9. Prague

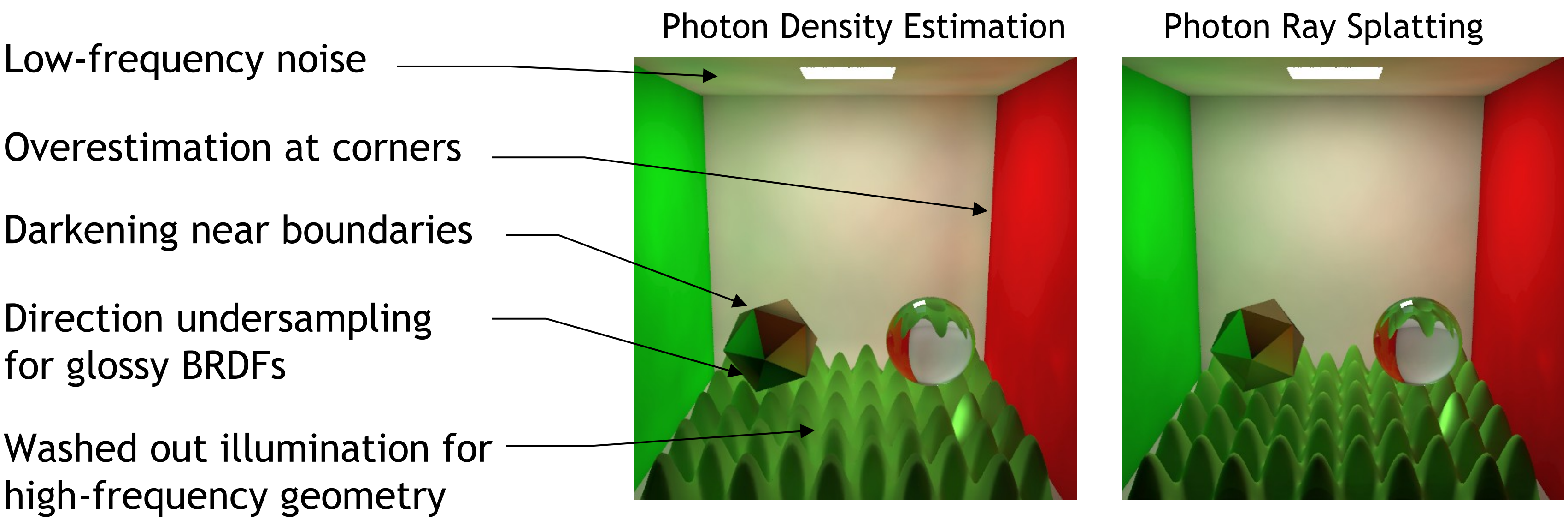
Global Illumination using Photon Ray Splatting

Robert Herzog* Vlastimil Havran‡ S. Kinuwaki*
K. Myszkowski* H.-P. Seidel*

*MPI Informatik, Germany ‡Czech Technical University Prague, Czech Republic

Motivation:

Filling the gap between photon density estimation and final gathering

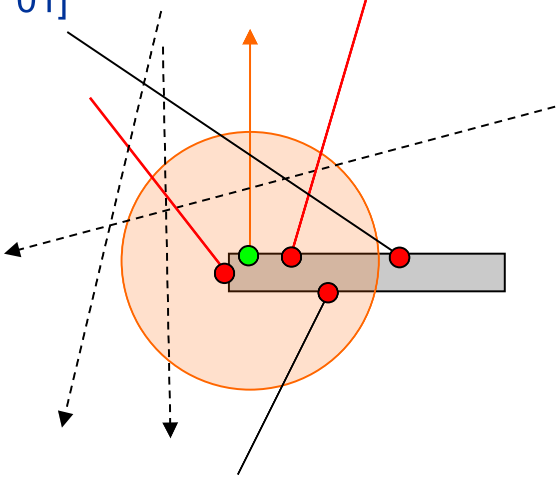


Related Work

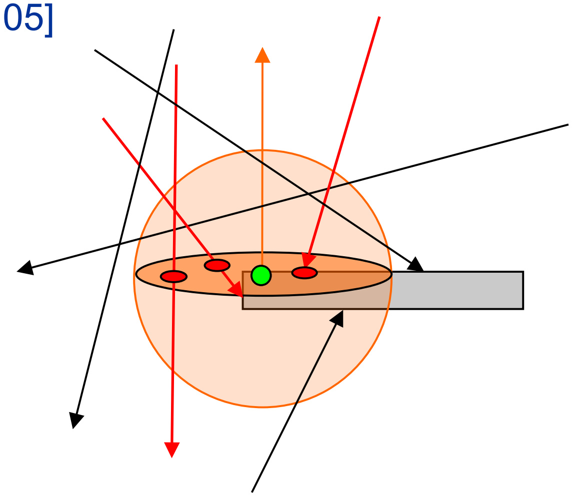
- Ray Maps for Global Illumination, **[Havran et al., EGSR 2005]**
  - Gathering of photon rays for a disc in the tangent plane
  - \+ Eliminates boundary bias
  - \- Complex and memory demanding search data structures
- Scalable Photon Splatting for Global Illumination, **[Lavignotte el al., Graphite 2003]**
  - \+ Efficient image based photon splatting approach with adaptive bandwidth selection
  - \- Restricted to triangle meshes and only diffuse light transport towards camera
- Radiance Caching for Efficient Global Illumination Computation, **[Křivánek et al., IEEE TVCG 2005]**
  - State-of-the-art for (ir)radiance caching on diffuse and glossy surface

Density Estimation Metrics

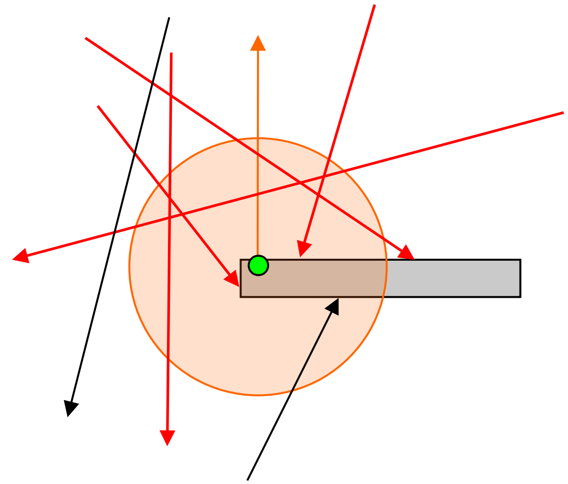
Standard photon density estimation on surfaces [Photon-maps, H.W. Jensen ‘01]



Photon density estimation for a disc in the tangent plane [Ray-maps, Havran et al. EGSR ’05]

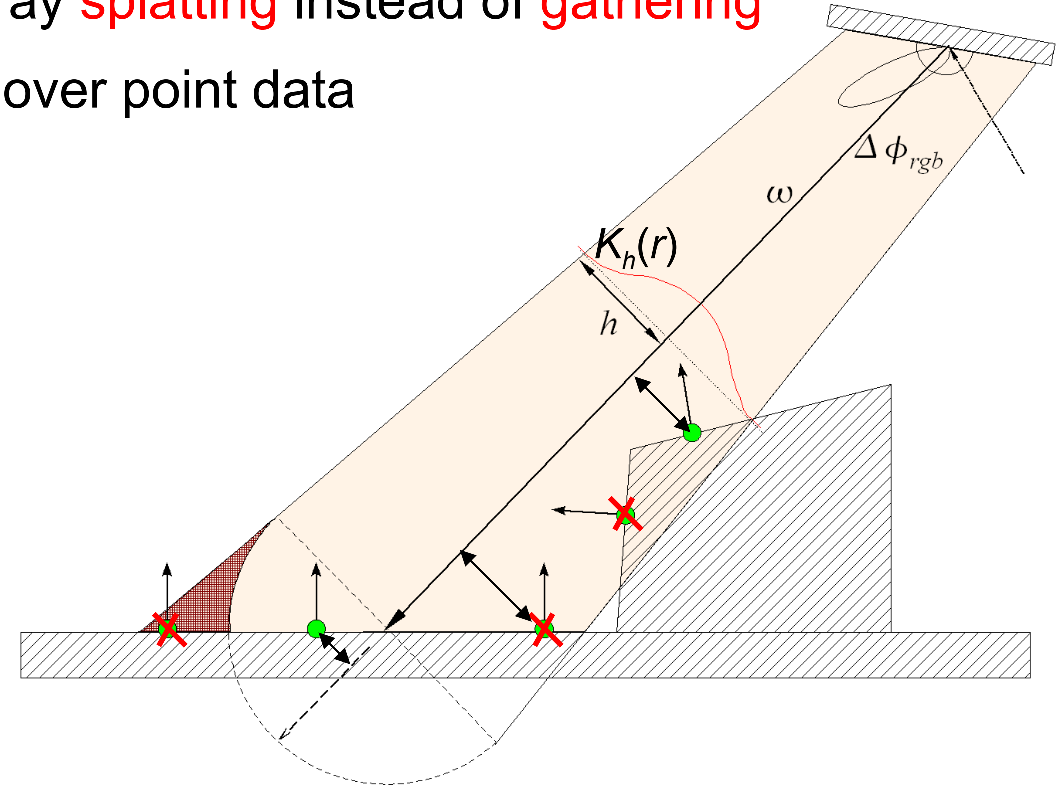


Cosine-weighted photon density estimation in ray-space [Eurographics ’07]



Ray Splatting instead of Gathering

- Efficient gathering of photon rays intersecting a sphere needs complex search data structures (e.g. ray maps **[Havran et al. 05]**)
- *Our solution*: photon ray splatting instead of gathering
- Boils down to search over point data



Choosing the Bandwidth (Splat-radius)

- Photon density estimation needs *adaptive* bandwidth selection
- Various statistical methods for bandwidth selection exist – though often too complex to be used for efficient photon density estimation
- Common k-nearest-neighbor (k-nn) method is computational intensive
  - → Either costly search for each density estimation query (*gathering* methods)
  - → Or costly precomputation of the k-nn radius per photon (*splatting* methods)

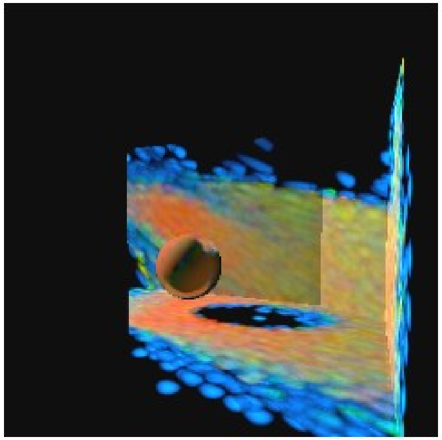
- Remedy: exploit knowledge from photon sampling, i.e. choose a photon's splatting radius according to *photon path probability density*
- We get this almost for free !!!

Adaptive Bandwidth Selection

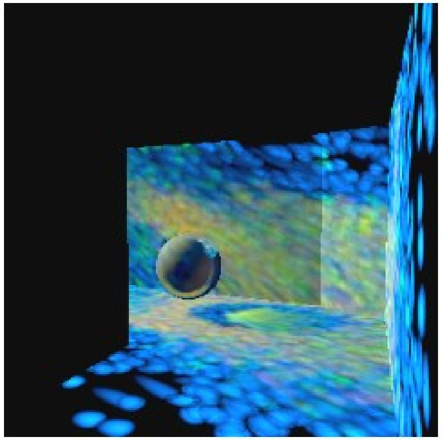
max planck institut informatik

- Choose the photon bandwidth inversely proportional to its path probability density
- Reliable estimate only for a small number of bounces (1 to 2)
  - → Need to slightly modify formula to become robust
  - → See paper for details!

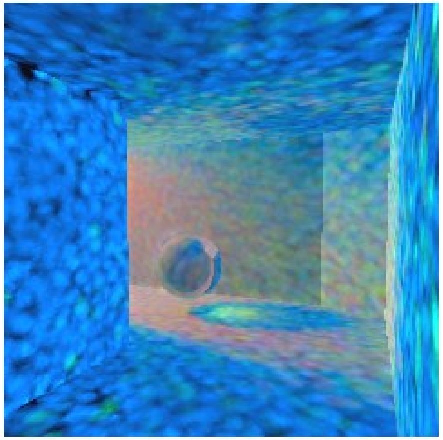
Color-coded visualization of the bandwidth (shown as photon splats)



2. bounce



4. bounce



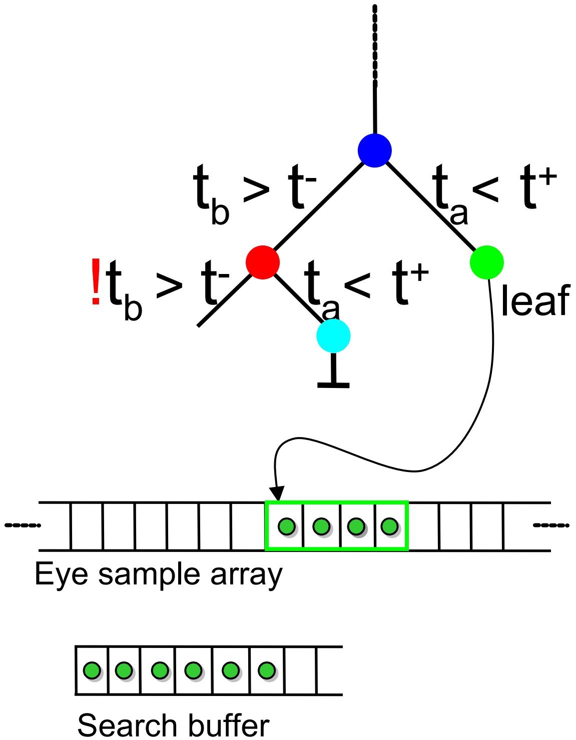
1-6 bounces

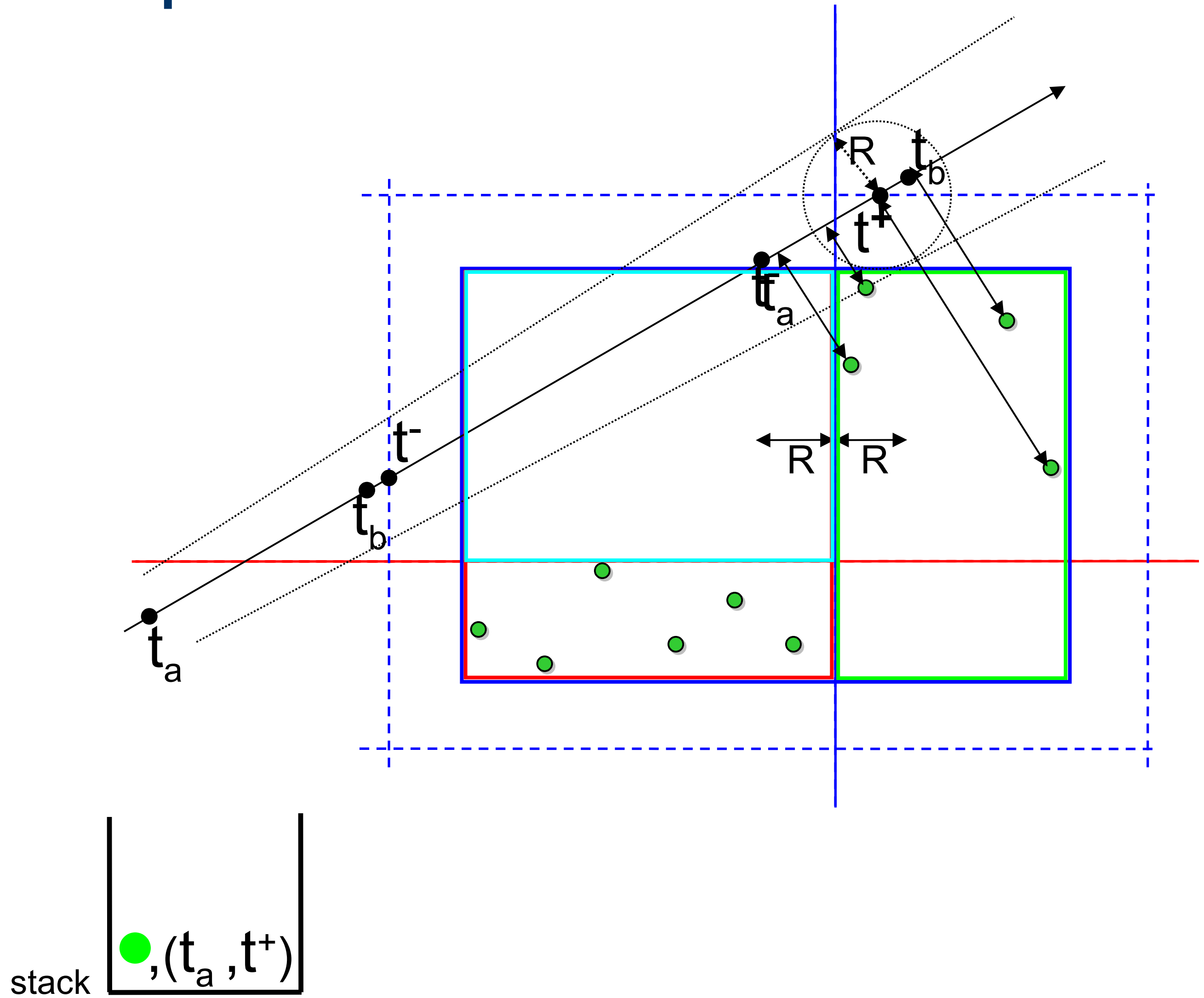
Indirect Illumination

Efficient Nearest Neighbor Search

- Nearest neighbor search for ray splatting is simpler than for ray gathering (e.g. ray maps **[Havran et al. EGSR05]**)
- Boils down to a search for point samples in a cylindrical bounding volume
- We use a standard kd-tree constructed on top of the eye samples with axis-aligned spatial splitting planes

Search in the Spatial kD-Tree





(Ir)radiance Caching

- Computation of (ir)radiance for each pixel is still too slow
- However, irradiance function is really smooth due to low-pass filtering via density estimation
- → sparse sampling of the (ir)radiance is sufficient
- We propose an algorithm similar to *cache splatting* **[Gautron et al., EGSR ’05]** in the spherical harmonics basis
- Cache interpolation gives additional filtering of noisy photon splats → reduce overall splat radius → more efficient ray splatting

Radiance Cache Splatting

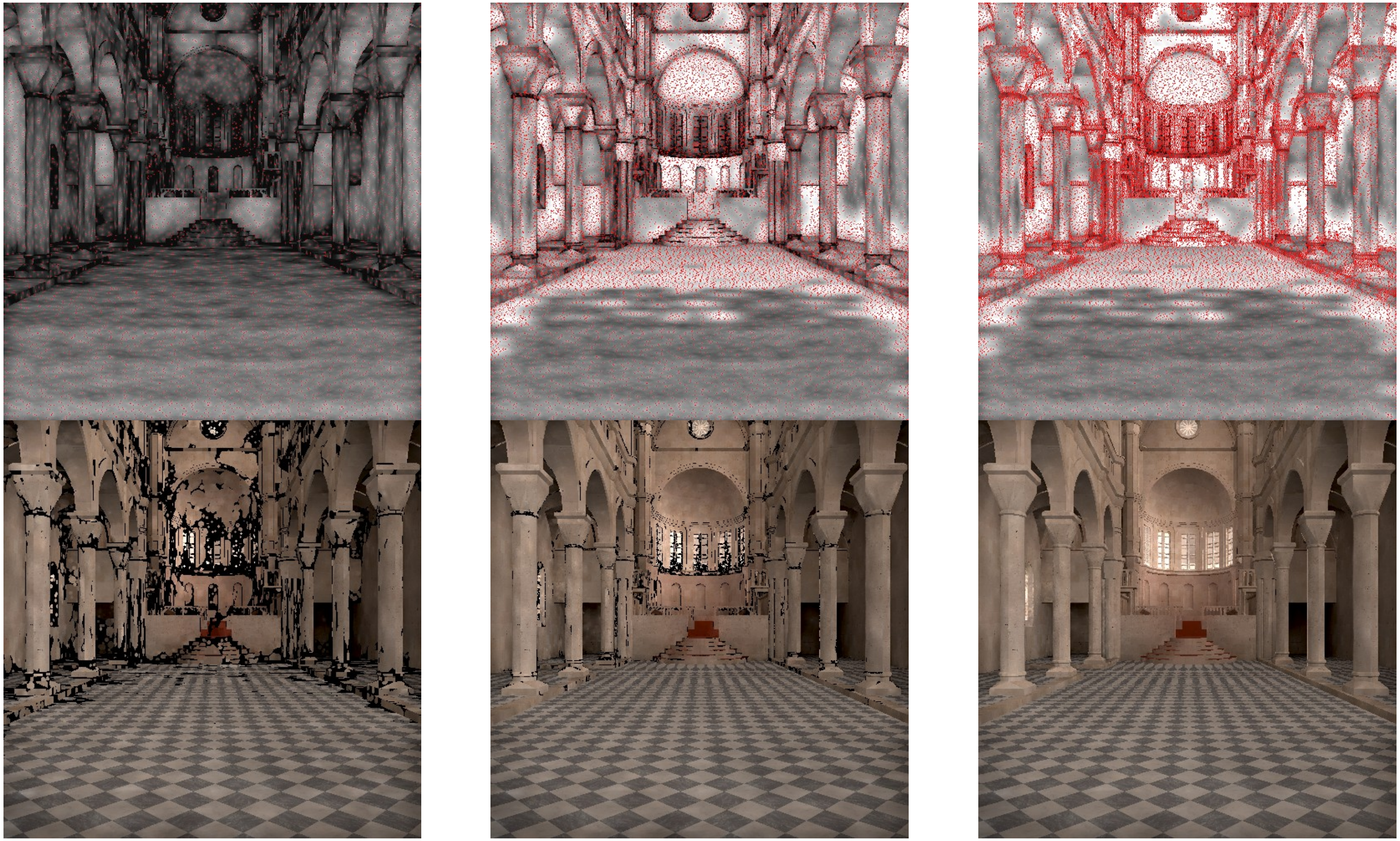
Noisy input from ray splatting → Adaptive radiance cache splatting → BRDF modulated and filtered result

Multi-Pass Radiance Cache Splatting

max planck institut informatik

Cache locations (red) + cache weights (~grey shade)

Cache splatting result



Pass 1 | Pass 2 | Pass 3

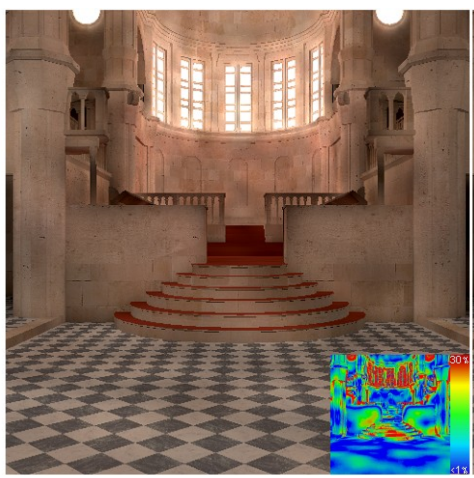
Results: Diffuse Sibenik Scene

- 500,000 indirect diffuse photons
- 512 x 512 image resolution
- Compared against reference (photon mapping + final gathering)

Final gathering (1200 rays per pixel)

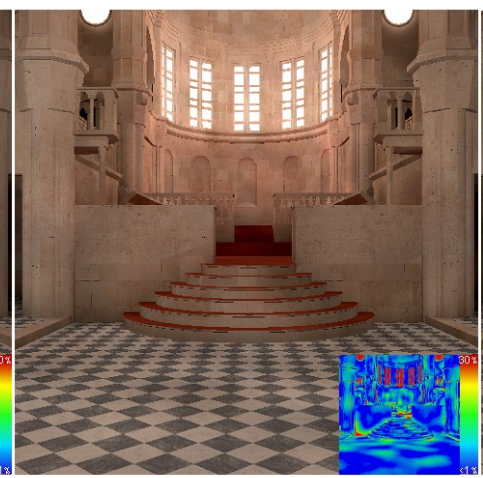
~4000 sec

Photon density estimation (500 k-nn)



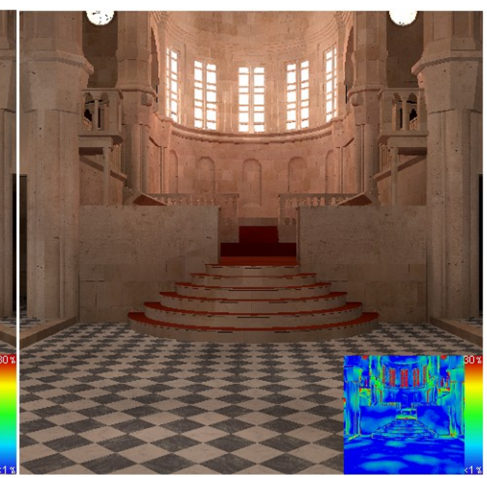
29 sec

Photon ray splatting



35 sec

Ray splatting with caching and filtering

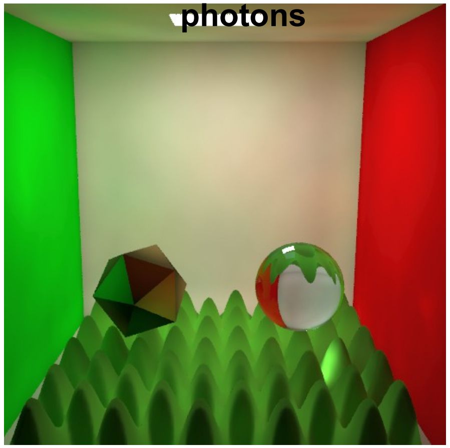


20 sec

Results

1000 × 1000 pixel, 200.000 photons | **500 × 500 pixel, 500.000 photons** | **1000 × 1000 pixel, 300.000**

K-nn Photon Density Estimation

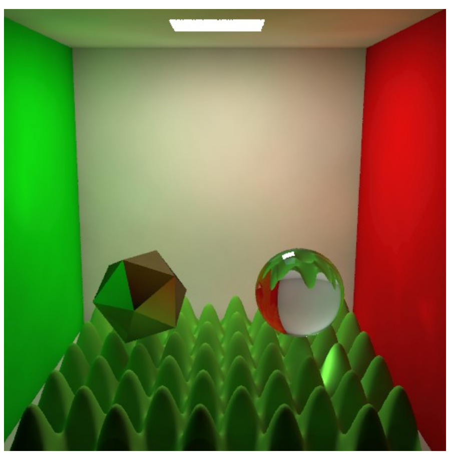


69 sec (400 k-nn)

66 sec (600 k-nn)

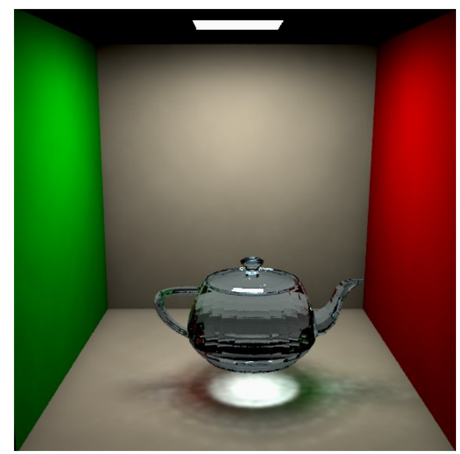
92 sec (400/ 250 k-nn)

Photon Ray Splatting (with radiance caching)



63 sec (16 sec)

64 sec (20 sec)

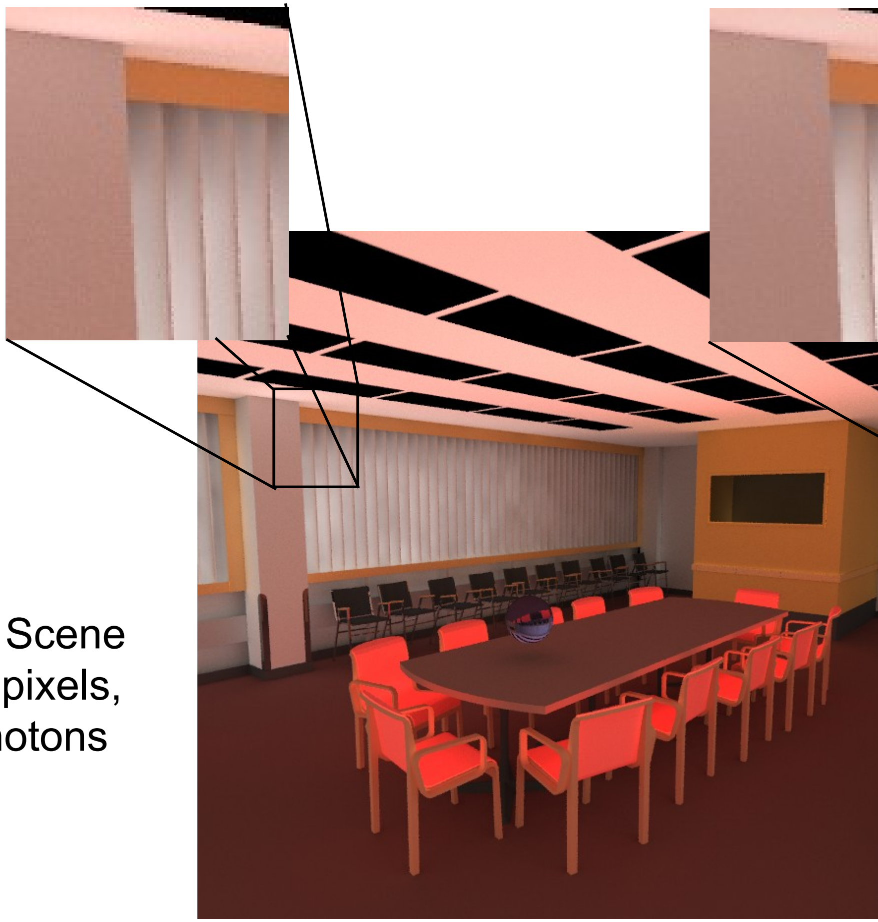


52 sec

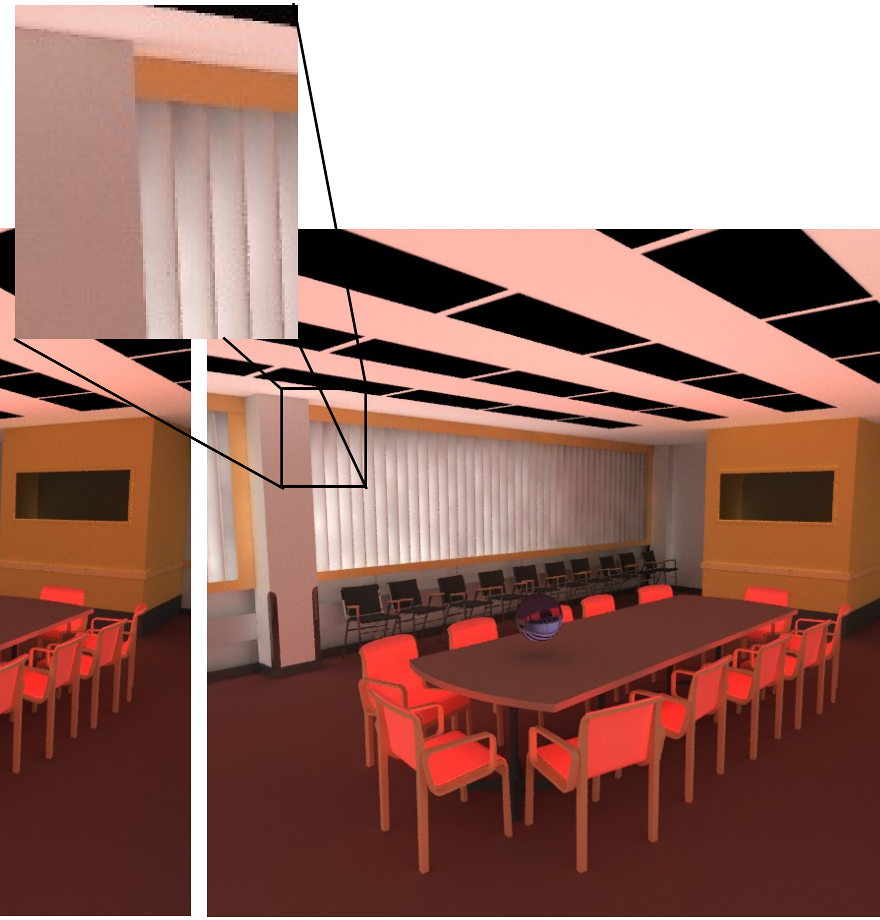
Two-Pass Final Gathering

Path-traced
(2000 rays)

Conference Scene
4×700×700 pixels,
160.000 photons



Ray splatting (600 rays)
~ 64 min

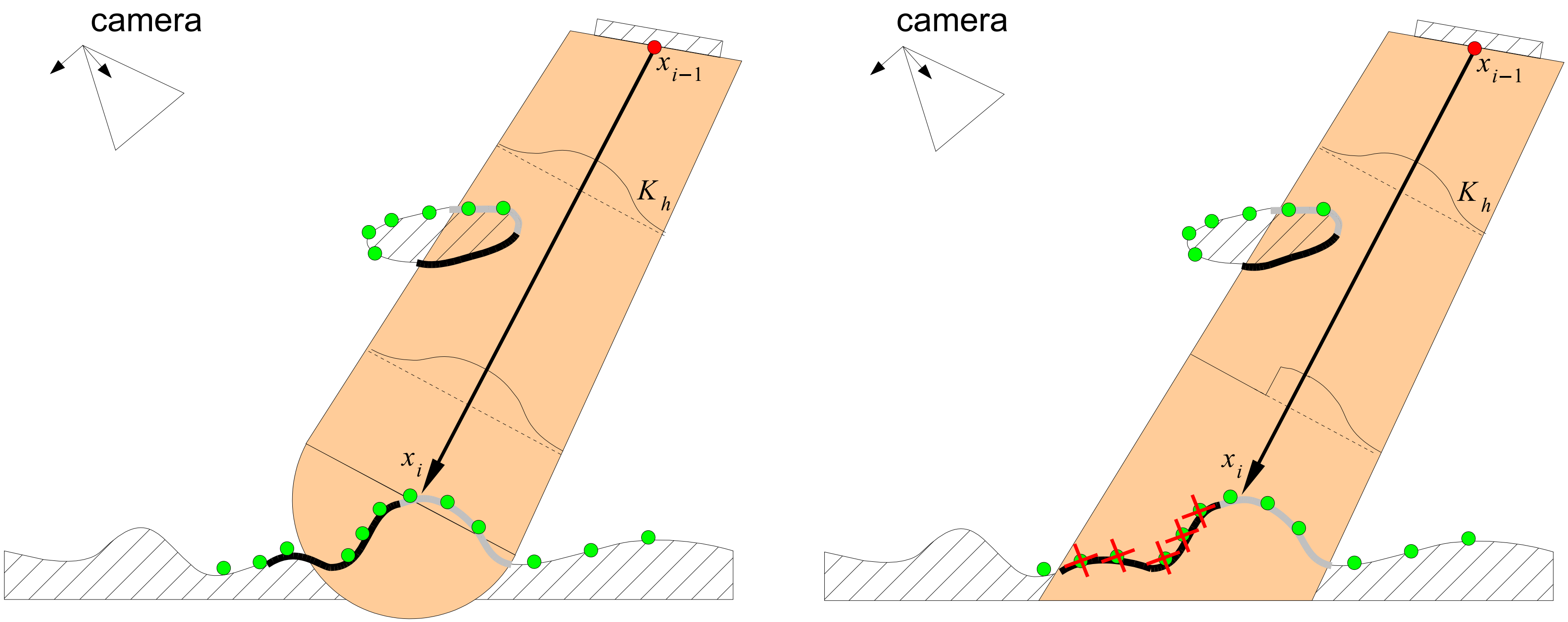


Photon mapping (70 k-nn
600 rays) ~ 85 min

Conclusions

- Ray density estimation as trade-off between photon density estimation and final gathering
- \+ No boundary bias, no topological bias
- \+ Works well with (ir)radiance caching
- \- However, still problems with proximity bias (light leaking)
  - → requires occlusion testing (future work)
    (**see [Herzog, Pacific Graphics '07]**)
  - → only low-frequency (indirect) illumination can be rendered so far
- More details provided in a technical report (coming soon)

Occlusion Testing



Current splatting neglects occlusion in the footprint

Correctly masking the density estimation kernel