02941 Physically Based Rendering

Particle Tracing and Photon Mapping

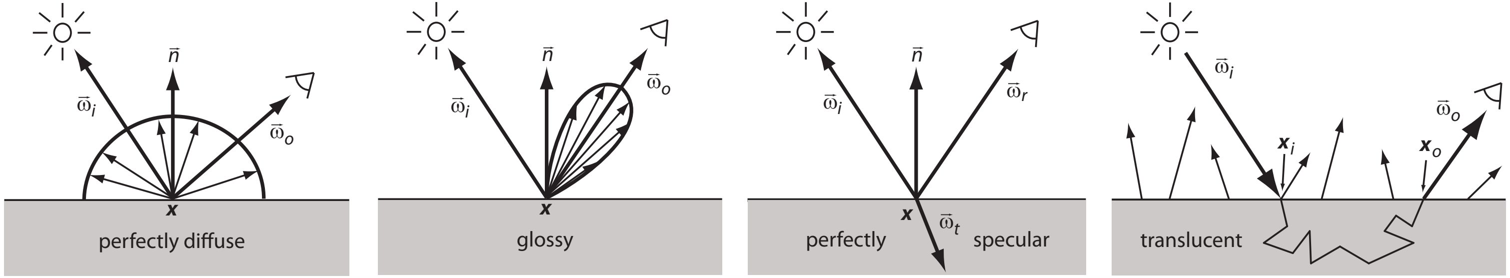
Jeppe Revall Frisvad

June 2020

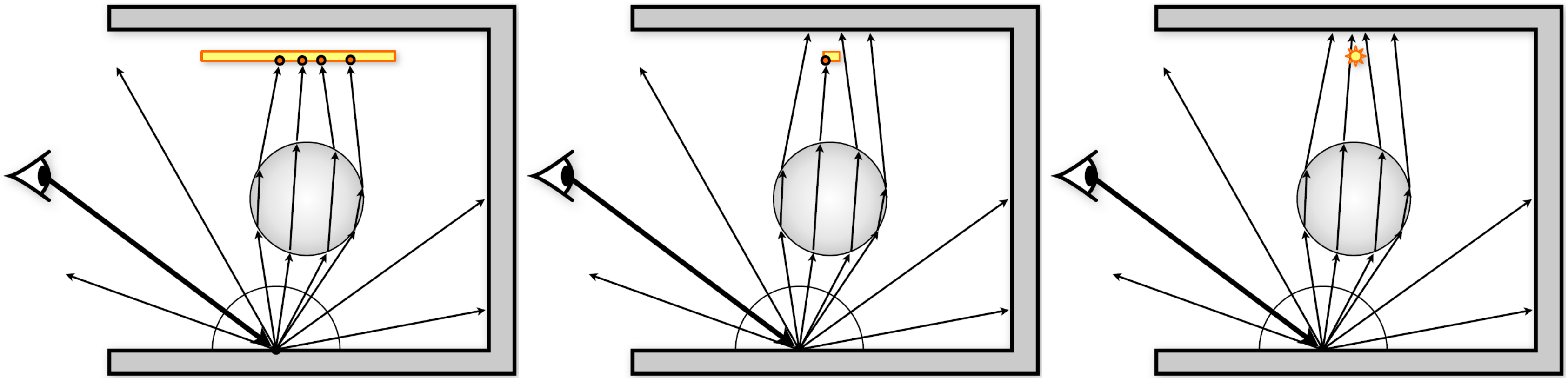
The importance of specular paths

- Light through windows.
- Light in water.
- Glossy materials.
- Translucent materials.

- But occluded specular paths are hard to find.



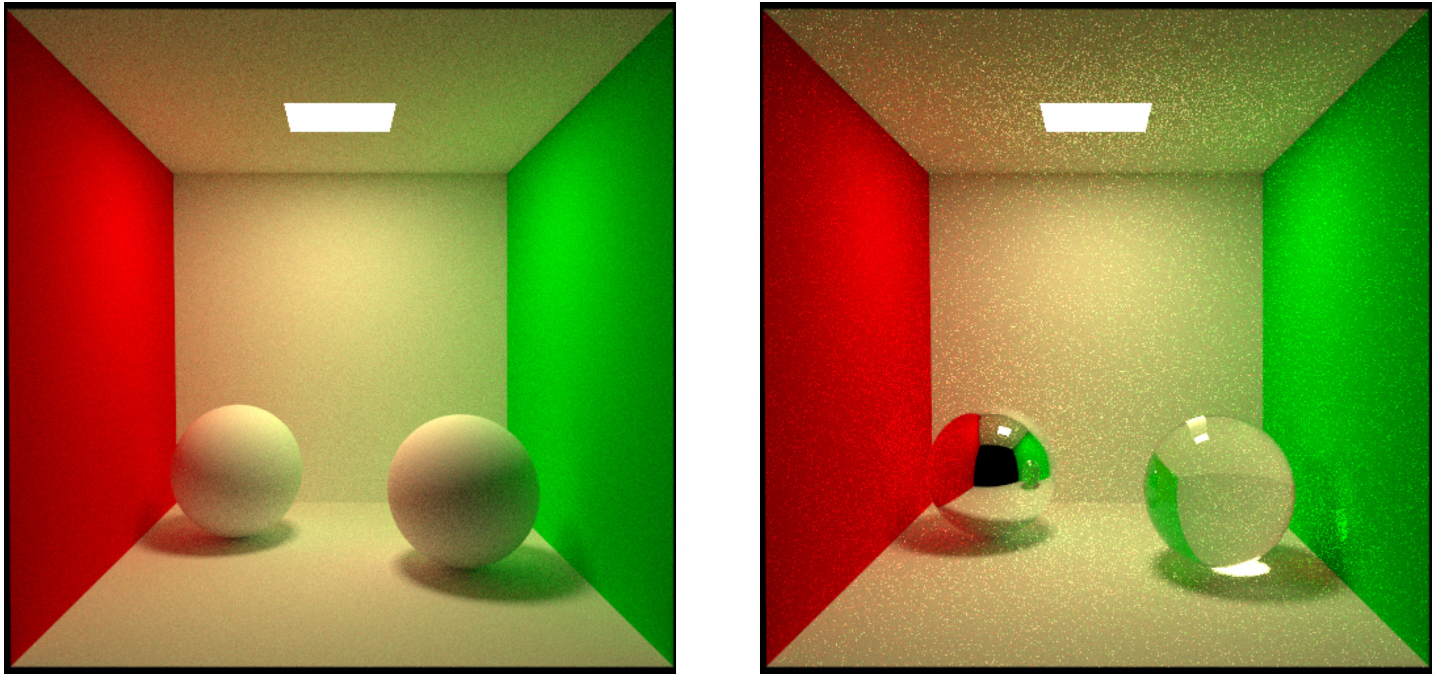
The caustics challenge



[Hachisuka et al. 2013]

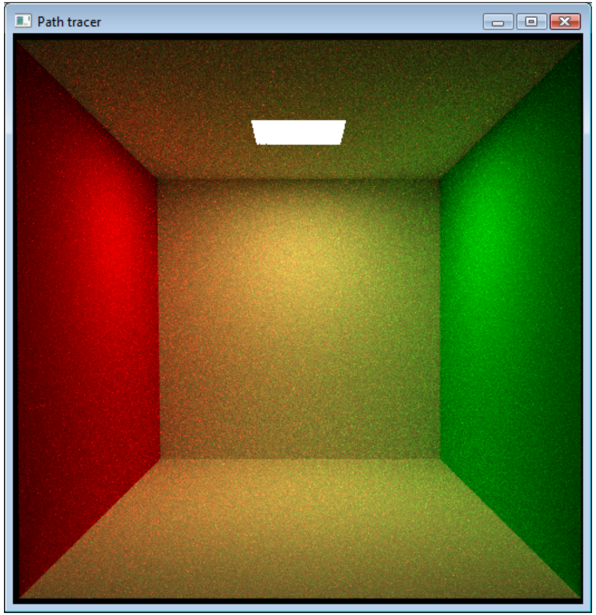
- How to find a small light source occluded by a transparent specular object (glass, water)?
- Translucent objects carry the same challenge.

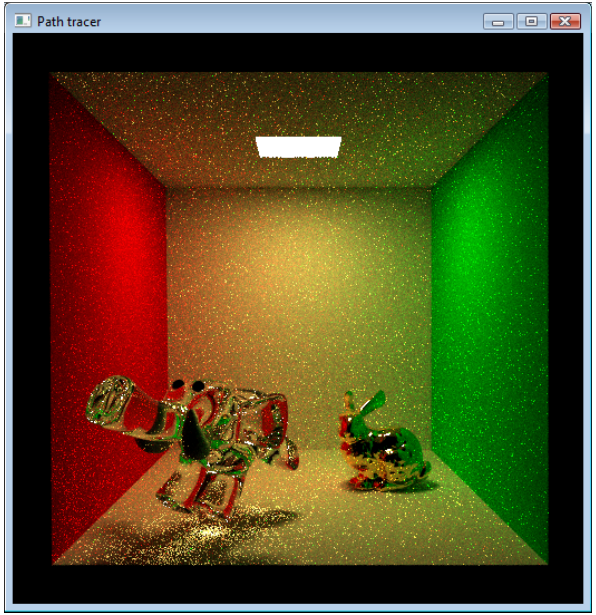
Path-traced caustics are noisy



Regular path tracing, 100 paths per pixel.

What's wrong with path tracing?





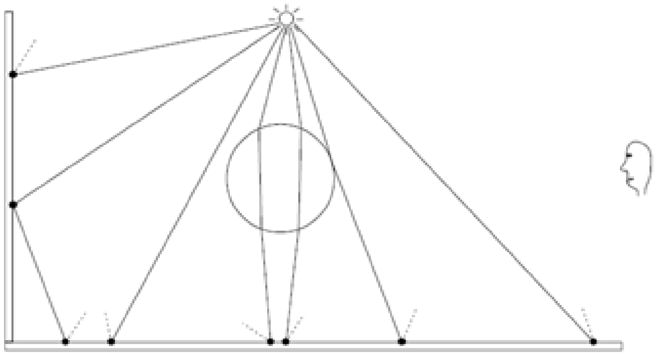
- Low probability paths result in bright dots (fireflies).
- A bright dot is visually unacceptable.
- A bright dot takes a long time to mean out.
- Soft caustics take "forever".

The variance versus bias trade-off

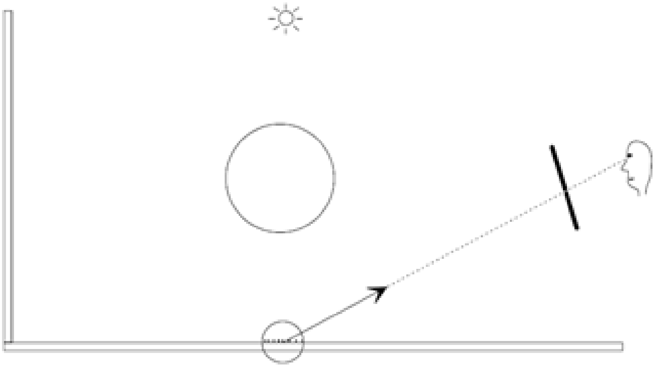
- What if we assume that illumination is always soft?
- Then we can reuse illumination computed at nearby points.
  - Eliminates high frequency noise.
  - Saves computational effort.
- But then the method is *biased*.
  - Error estimates are not well-defined.
  - Images with no visual artifacts may still have substantial error.
  - Artifacts are not eliminated in a predictable way.
- The trade-off is *high frequency* versus *low frequency* noise.

Reusing computations

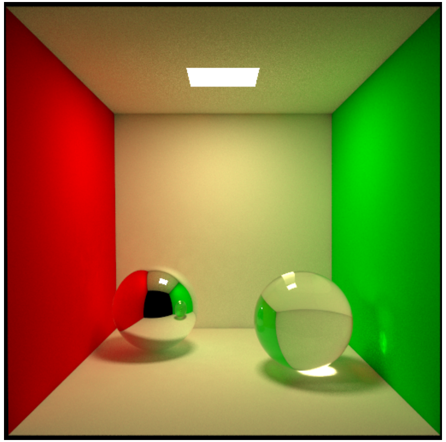
- Store light path vertices (photon mapping):
  - Path reuse vs. memory consumption.



trace and store



flux density estimation



photon mapping

[Jensen and Christensen 2000]

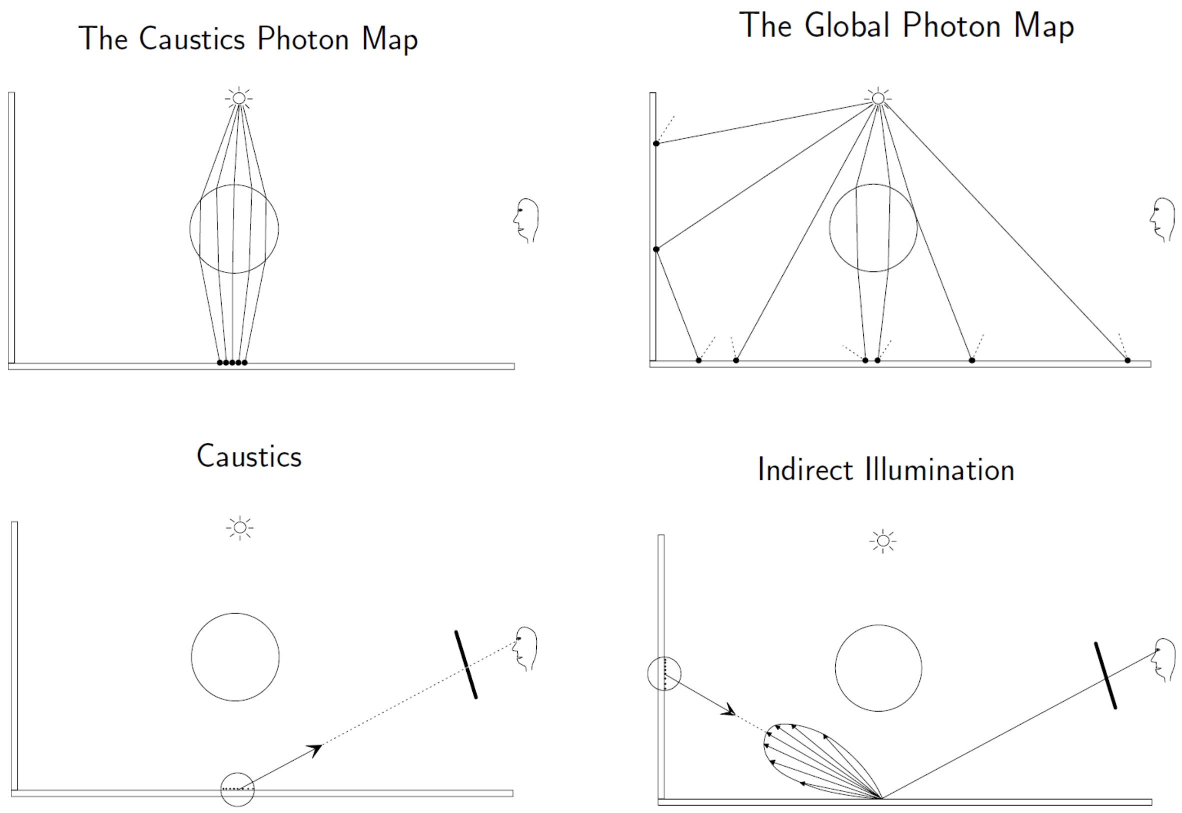
The Light of Mies van der Rohe

- Photon mapping excels at caustics and is thus an excellent choice for architectural lighting simulation

How to pre-compute illumination?

- Sparse tracing from the eye (irradiance caching).
  - Very efficient.
  - Not good for caustics and glossy surfaces.
- Trace rays from the light sources (particle tracing).
  - Rays from the lights are good for finding caustics.
  - Caustics are particularly prone to variance in path tracing.
- A popular particle tracing algorithm: *photon mapping*.
  - Emit.
  - Trace.
  - Store at non-specular surfaces.
  - Use density estimation to reconstruct illumination.

Photon mapping - overview



References

- Jensen, H. W., and Christensen, N. J. A Practical Guide to Global Illumination Using Photon Maps, ACM SIGGRAPH 2000 Course Notes, Course 8, 2000.

Photon emission

- "Photons" in photon mapping are *flux packets*.
- We must ensure that our rays carry *flux*.
- The definition of radiance:

$$L = \frac{\mathrm{d}^2\Phi}{\cos\theta \, \mathrm{d}A \, \mathrm{d}\omega} \quad .$$

- Flux emitted from a source:

$$\Phi = \int_{2\pi}\int_A L_e \cos\theta \, \mathrm{d}A \, \mathrm{d}\omega = \int_{2\pi}\int_A L_e(\boldsymbol{x},\vec{\omega})(\vec{n}\cdot\vec{\omega}) \, \mathrm{d}A \, \mathrm{d}\omega \quad .$$

- A ray is a Monte Carlo sample of this integral.
  - Shooting N rays.
  - Uniformly sampling source area A for origin of ray.
  - Sampling cosine-weighted hemisphere for direction of ray.
  - Assuming homogeneous, diffuse light.

$$\Phi_N = \frac{1}{N}\sum_{p=1}^{N}\Phi_p \quad , \quad \Phi_p = \frac{L_e(\boldsymbol{x},\vec{\omega})(\vec{n}\cdot\vec{\omega})}{\mathrm{pdf}(\boldsymbol{x})\,\mathrm{pdf}(\vec{\omega})} = L_e A\pi \quad .$$

Photon tracing

- Once emitted the flux is confined by the solid angle of the ray.
- Flux carried by a ray changes like radiance upon diffuse and specular reflection.
- Tracing ‘photons’ is like tracing ordinary rays.
- Whenever the ‘photon’ is traced to a non-specular surface:
  - It is stored in a kd-tree.
  - Position is stored.
  - Direction from where it came (opposite ray direction).
  - Flux (Φ_p) is stored.
- Russian roulette is used to stop the recursive tracing.

Photon flux density estimation

- The rendering equation in terms of irradiance $E = \mathrm{d}\Phi/\mathrm{d}A$:

$$
\begin{aligned}
L(\boldsymbol{x},\vec{\omega}) &= L_e(\boldsymbol{x},\vec{\omega}) + \int f_r(\boldsymbol{x},\vec{\omega}',\vec{\omega})\,\mathrm{d}E(\boldsymbol{x},\vec{\omega}') \\
&\approx L_e(\boldsymbol{x},\vec{\omega}) + \sum_{p\in\Delta A} f_r(\boldsymbol{x},\vec{\omega}'_p,\vec{\omega})\,\frac{\Delta\Phi_p(\boldsymbol{x},\vec{\omega}'_p)}{\Delta A}
\end{aligned}
$$

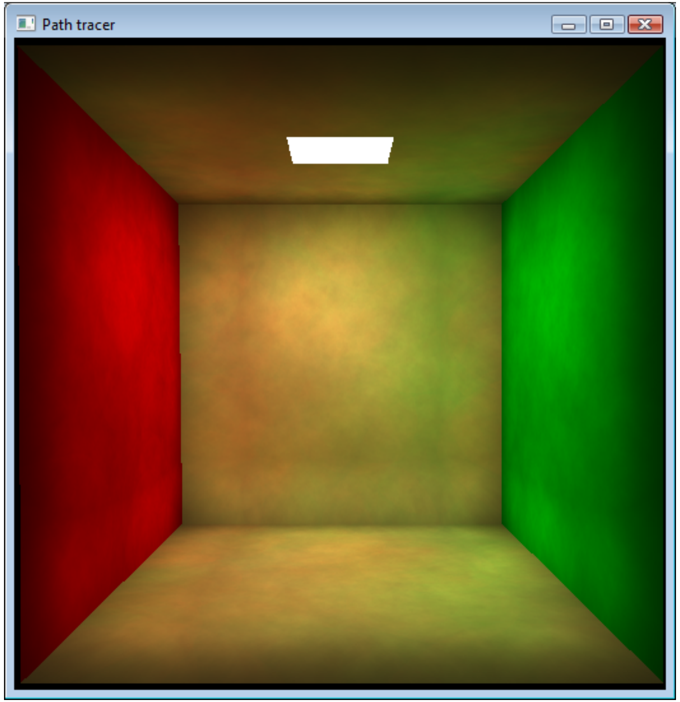
- The Δ-term is called the *irradiance estimate*.
- It is computed using a *kernel method*.
  - Consider a circular surface area $\Delta A = \pi r^2$
  - Then the power contributed by the 'photon' $p \in \Delta A$ is

    $$\Delta\Phi_p(\boldsymbol{x},\vec{\omega}'_p) = \Phi_p\,\pi K\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_p\|}{r}\right)$$

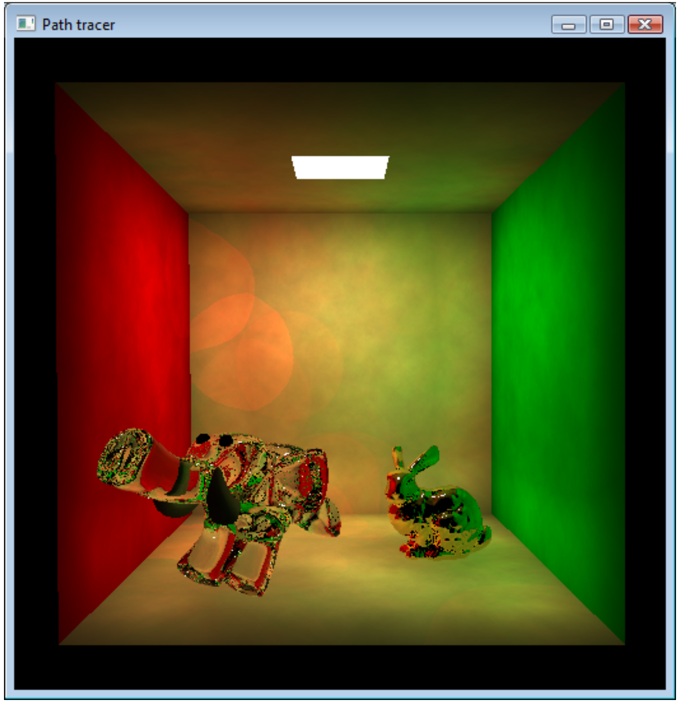
    where K is a filter kernel.
  - Simplest choice (constant kernel)

    $$K(x) = \begin{cases} 1/\pi & \text{for } x^2 < 1 \\ 0 & \text{otherwise} \end{cases}.$$

Photon mapping results (radiance estimate)



c. 10 s



c. 10 s

Final Gathering

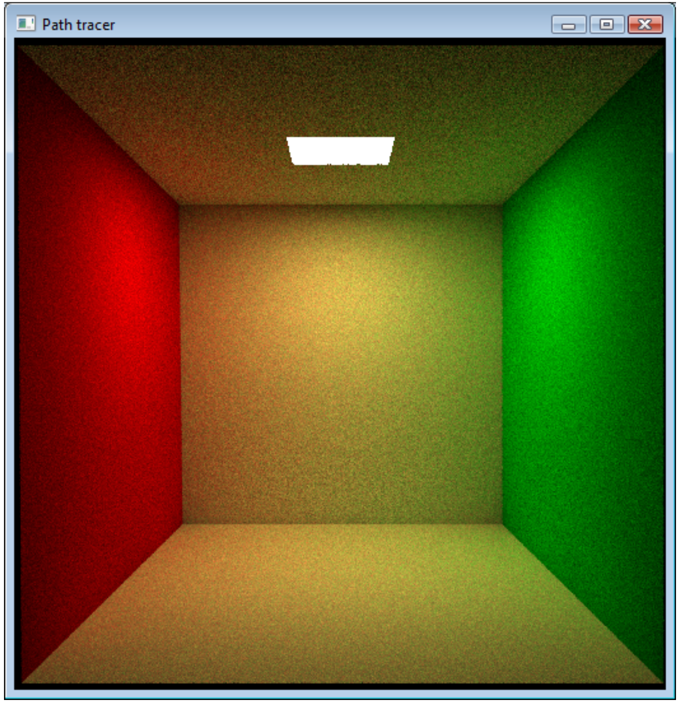
- How to eliminate the worst low frequency noise:

- Use two photon maps:
  - A global map (all photons: L(S|D)*D)
  - A caustics map (only caustics photons: LS+D)

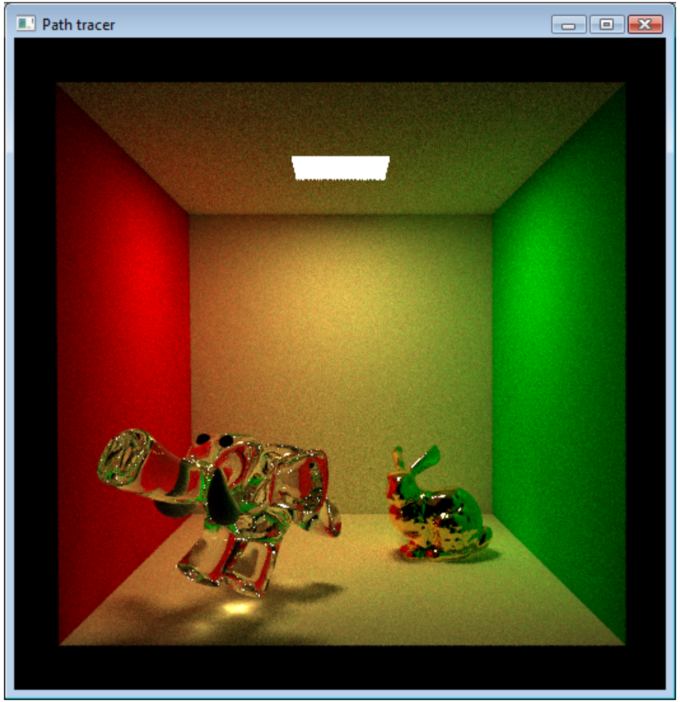
- At the first non-specular surface reached from the eye:
  - Do a final gathering (sample the hemisphere)
  - Add the radiance estimate from the caustics map

- At subsequent non-specular surfaces:
  - Use the radiance estimate from the global map

Photon mapping results (final gathering)



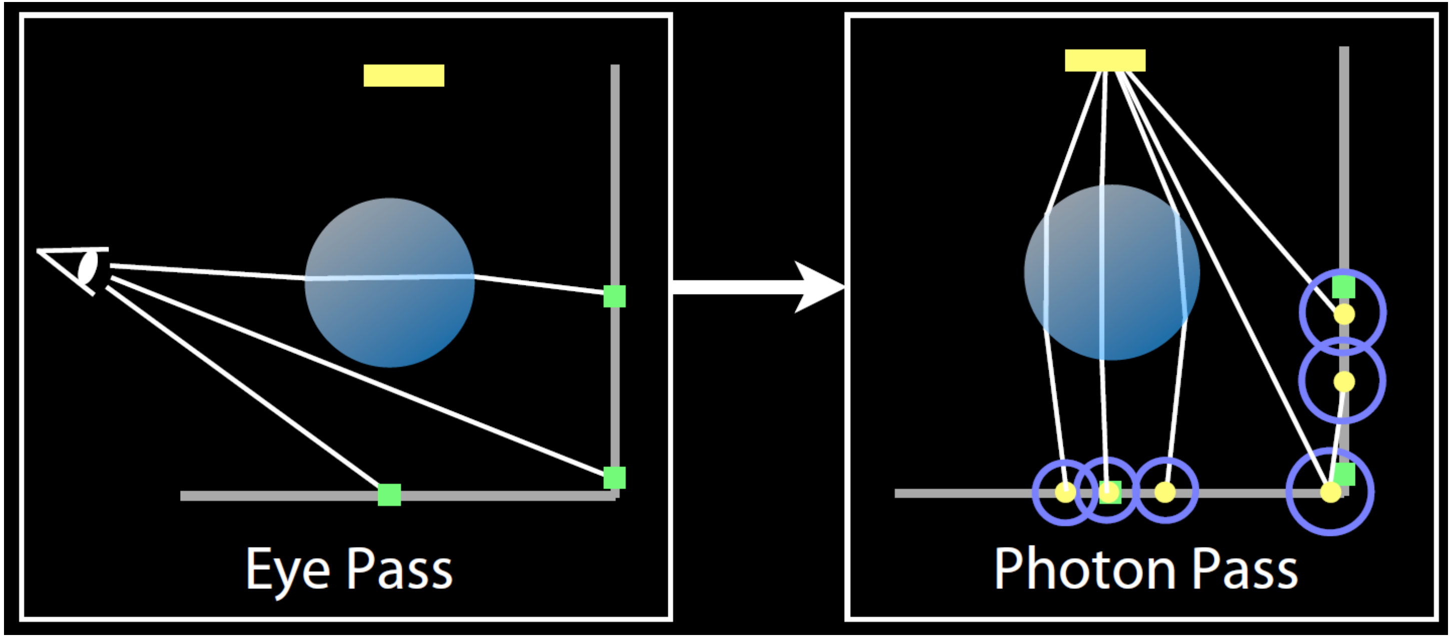
c. 3 min



c. 5 min and 20 s

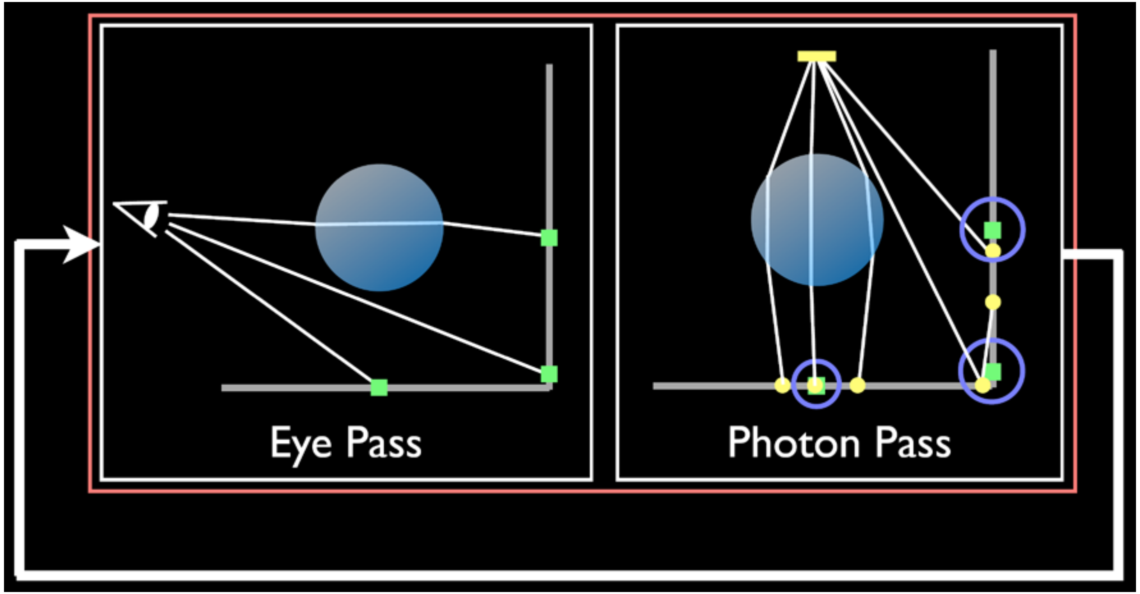
Reusing computations

- Storing eye path vertices (photon splatting)
  - Path reuse vs. accuracy in density estimation



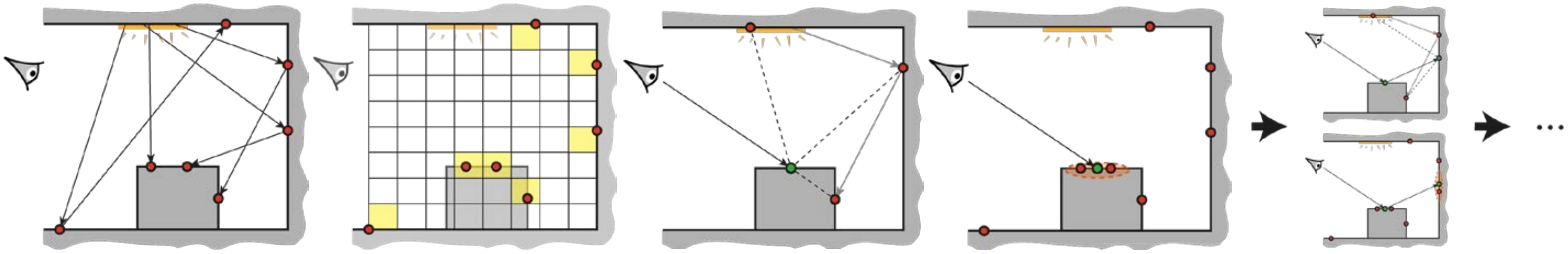
Reusing computations

- Store all path vertices and refine progressively (stochastic progressive photon mapping)
  - Path reuse, little memory, but slow convergence



[Hachisuka et al. 2012]

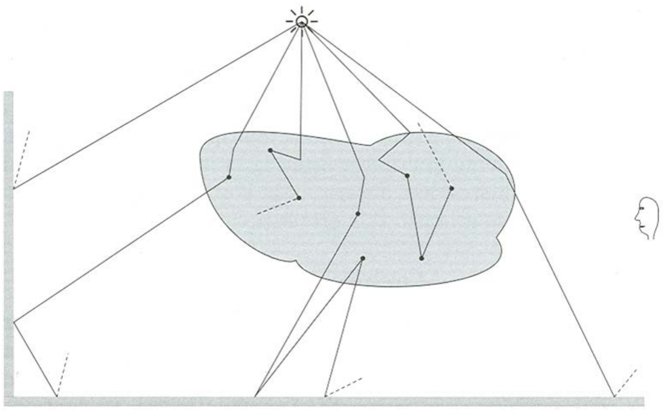
Unified framework

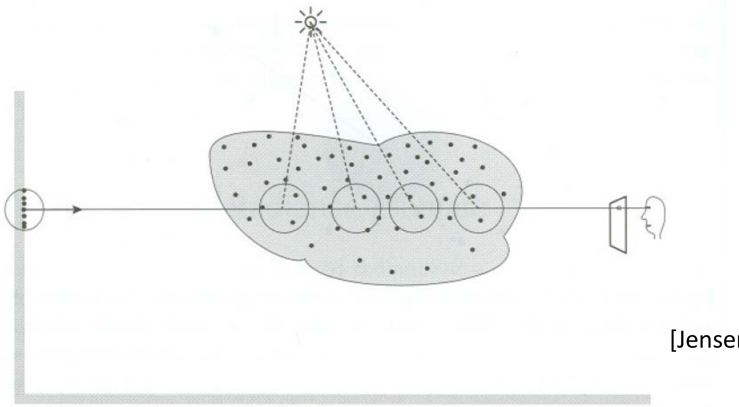


[Hachisuka et al. 2013]

- Bidirectional path tracing (vertex connection)
- Progressive photon mapping (vertex merging)
- Same framework, different integrators

Volumetric effects





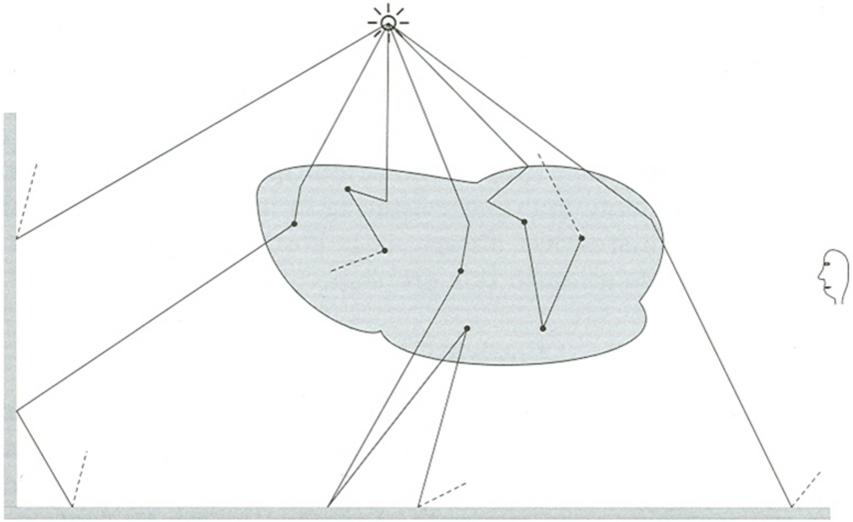
[Jensen 2001]

[Jensen and Christensen 1998]

[Jensen and Christensen 2000]

[Jarosz et al. 2008]

Volume photon mapping

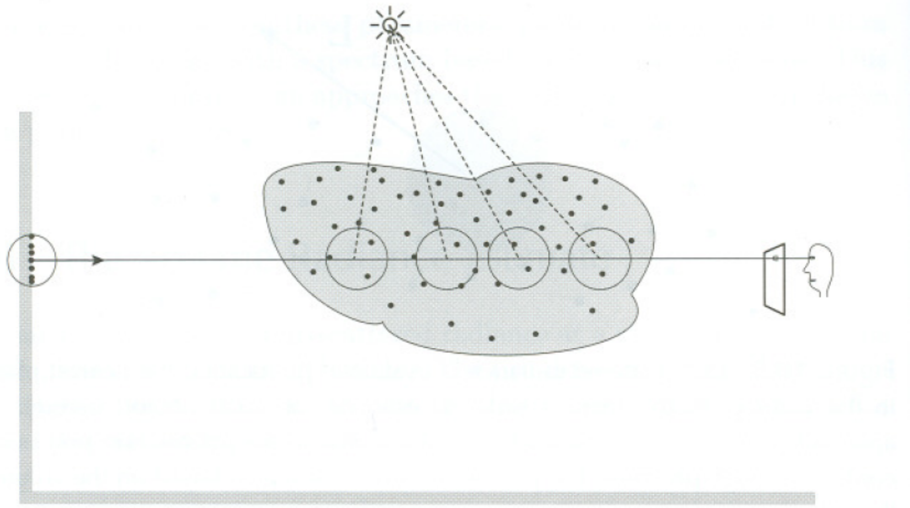


- Trace photons in the same way as we trace eye rays when path tracing volumes (see slides on volume rendering).
- Store photons whenever they interact with the medium (both at scattering and absorption events).

References

- Jensen, H. W. Realistic Image Synthesis Using Photon Mapping. A K Peters, 2001.

Ray marching with the photon map



- Stepping backward along the ray

$$L_n(\boldsymbol{x}, \vec{\omega}) = J(\boldsymbol{x}, \vec{\omega})\sigma_s(\boldsymbol{x})\Delta x + e^{-\sigma_t(\boldsymbol{x})\Delta x} L_{n-1}(\boldsymbol{x} + \vec{\omega}\Delta x, \vec{\omega}) \ ,$$

  where $J(\boldsymbol{x}, \vec{\omega})$ is the source function and $\Delta x = -\ln(\xi)/\sigma_t(\boldsymbol{x})$.
- L_0 is the radiance entering the volume at the backside.

Volumetric radiance

- Radiance incident or exitent at a surface location:
$$L = \frac{\mathrm{d}^2\Phi}{\cos\theta\,\mathrm{d}A\,\mathrm{d}\omega} \ .$$
- How does it work in a volume? What is the projected area?
- The scattering coefficient is the total scattering cross section $\mathrm{d}A_s$ in an element of volume $\mathrm{d}V$ around a point $\boldsymbol{x}$
$$\sigma_s(\boldsymbol{x}) = \frac{\mathrm{d}A_s(\boldsymbol{x})}{\mathrm{d}V} = \int_0^\infty C_s(r)N(\boldsymbol{x}, r)\,\mathrm{d}r \ ,$$
where
  - r is the radius of a particle,
  - C_s is the scattering cross section of the particle,
  - N is the number of these particles in $\mathrm{d}V$.
- Let us use $\mathrm{d}A_s$ in place of projected area $\cos\theta\,\mathrm{d}A$ to define radiance in a volume. Then
$$L = \frac{\mathrm{d}^2\Phi}{\sigma_s\,\mathrm{d}V\,\mathrm{d}\omega} \ .$$

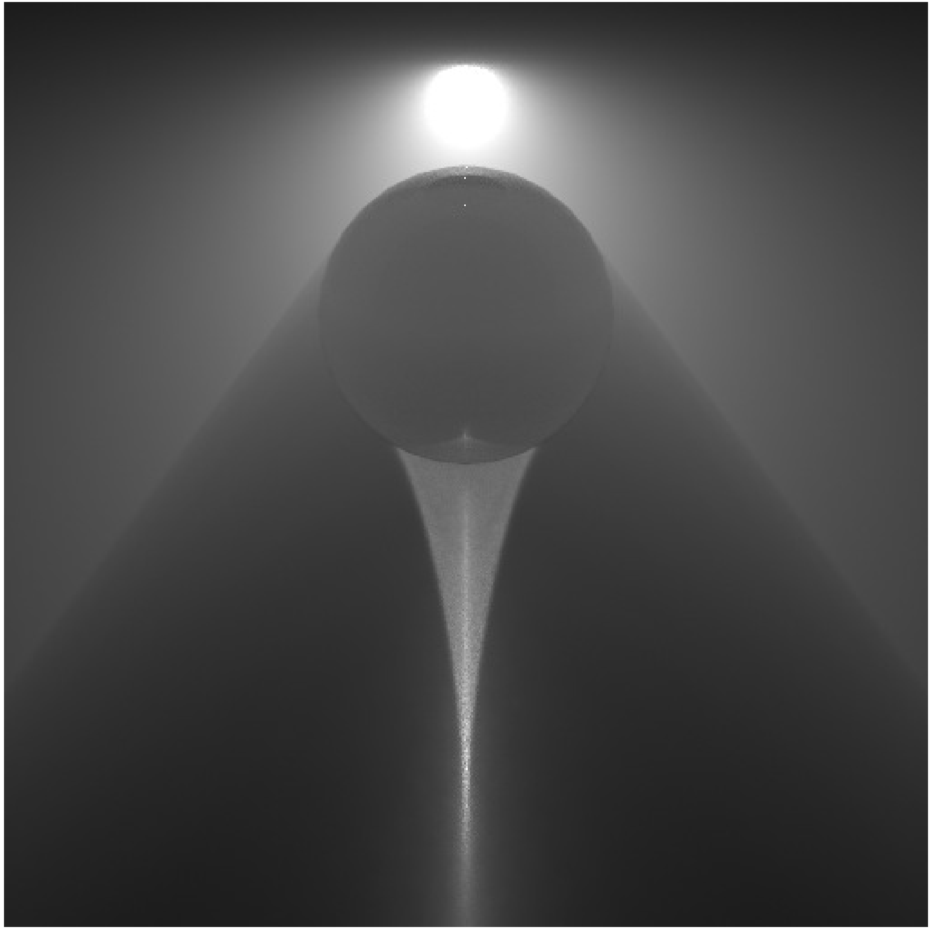
The volume radiance estimate

- Using the definition of volumetric radiance: $L = \dfrac{\mathrm{d}^2\Phi}{\sigma_s\,\mathrm{d}V\,\mathrm{d}\omega}$, we can estimate radiance in a volume using the photon map.
- The source function is

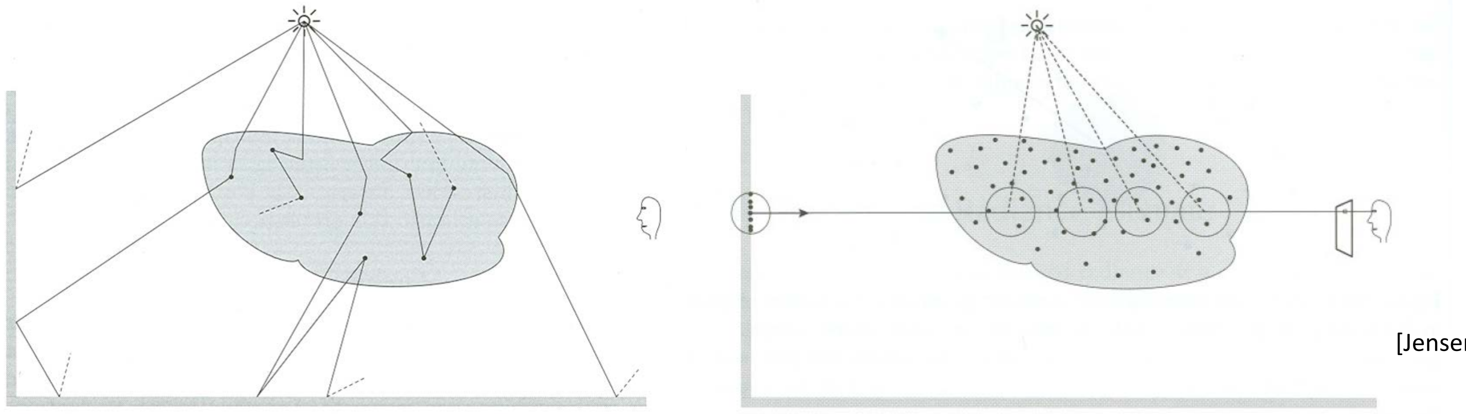
$$
\begin{aligned}
J(\boldsymbol{x},\vec{\omega}) &= \int_{4\pi} p(\boldsymbol{x},\vec{\omega}\,',\vec{\omega})L(\boldsymbol{x},\vec{\omega}')\mathrm{d}\omega' \\
&= \int_{4\pi} p(\boldsymbol{x},\vec{\omega}',\vec{\omega})\frac{\mathrm{d}^2\Phi}{\sigma_s(\boldsymbol{x})\,\mathrm{d}V\,\mathrm{d}\omega'}\mathrm{d}\omega' \\
&= \frac{1}{\sigma_s(\boldsymbol{x})}\int_{4\pi} p(\boldsymbol{x},\vec{\omega}',\vec{\omega})\frac{\mathrm{d}^2\Phi}{\mathrm{d}V} \\
&\approx \frac{1}{\sigma_s(\boldsymbol{x})}\sum_{p\in\Delta V} p(\boldsymbol{x},\vec{\omega}'_p,\vec{\omega})\frac{\Delta\Phi_p(\boldsymbol{x},\vec{\omega}'_p)}{\Delta V} \quad ,
\end{aligned}
$$

where we consider a spherical volume $\Delta V = \frac{4}{3}\pi r^3$.

Participating medium



Volumetric effects



[Jensen 2001]

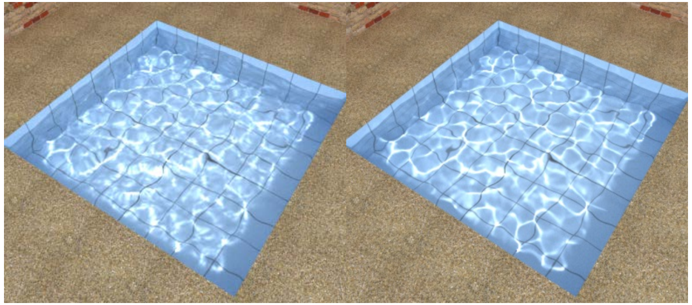
[Jensen and Christensen 1998]

[Jensen and Christensen 2000]

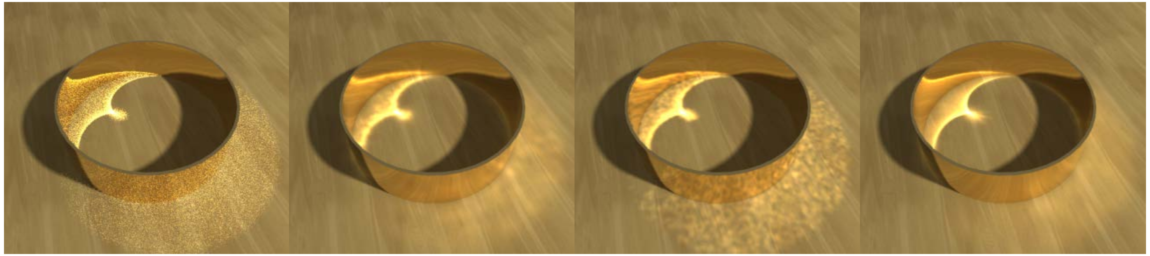
[Jarosz et al. 2008]

Improving density estimation

- Photon differentials for surfaces and volumes

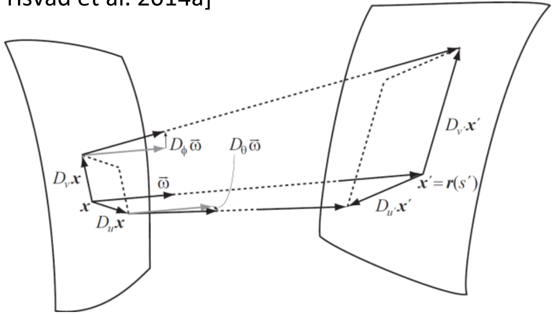


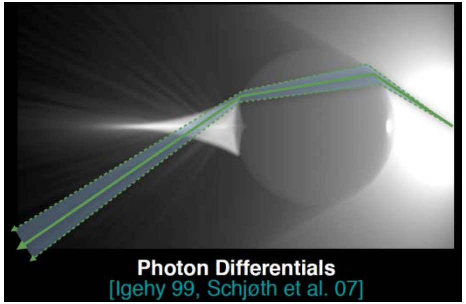
photon mapping photon differentials



path tracing mapping splatting differentials

[Frisvad et al. 2014a]





First-order approximation of the photon path derivative

[Jarosz et al. 2011]

Photon diffusion

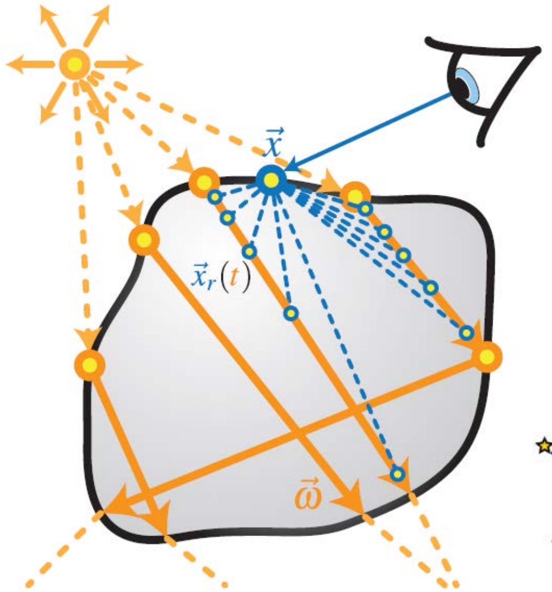
- Diffusion-based analytical models ease computation of subsurface scattering

path tracing
(days)
still noisy

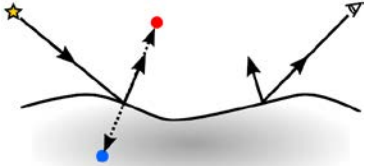
[Donner and Jensen 2007]

diffusion models (minutes)
no noise

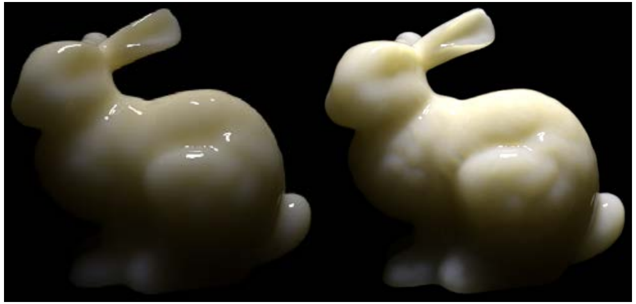
[Frisvad et al. 2014b]

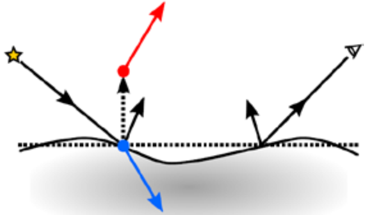


photon (beam) diffusion
[Habel et al. 2013]



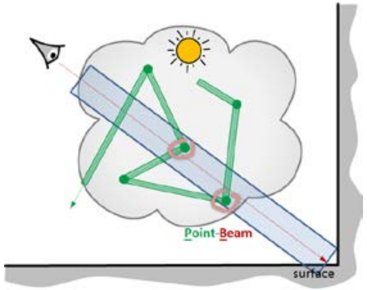
dipole model
[Jensen et al. 2001]



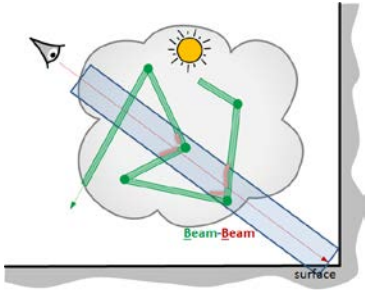


directional
dipole model

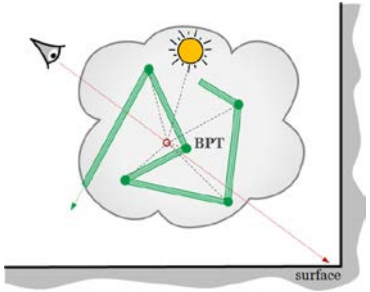
Unified framework with volumes



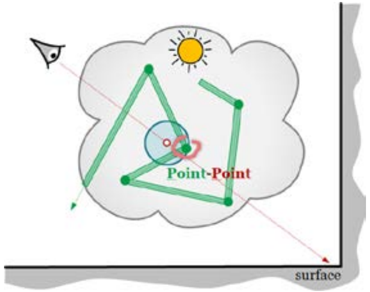
eye beams



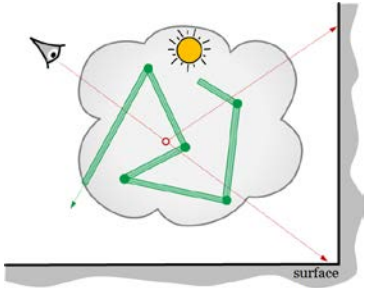
photon beams



path tracing



photon mapping



continue …

[Křivánek et al. 2014]