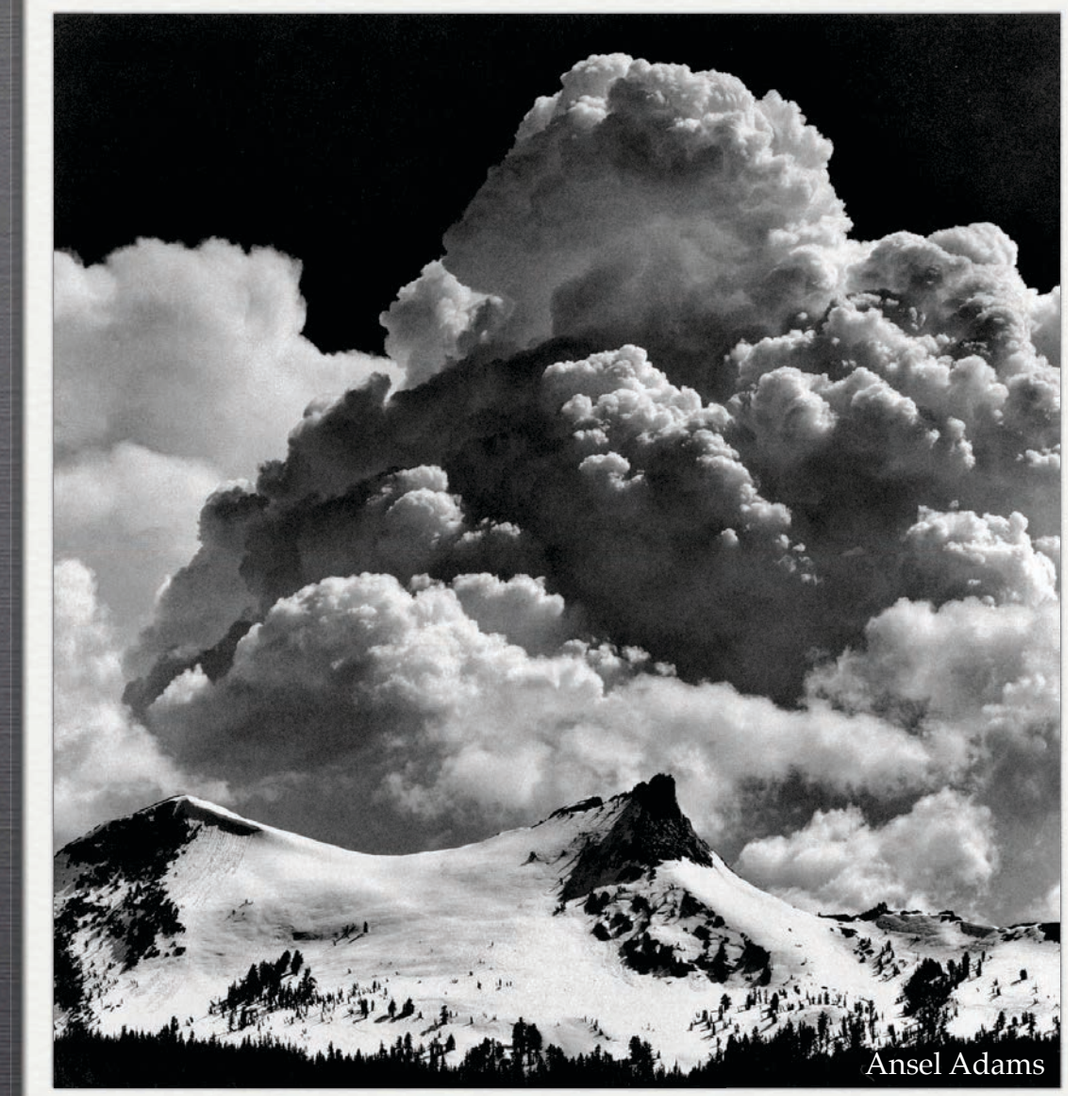
# Rendering Participating Media





## Wojciech Jarosz

Disney Research, Zürich

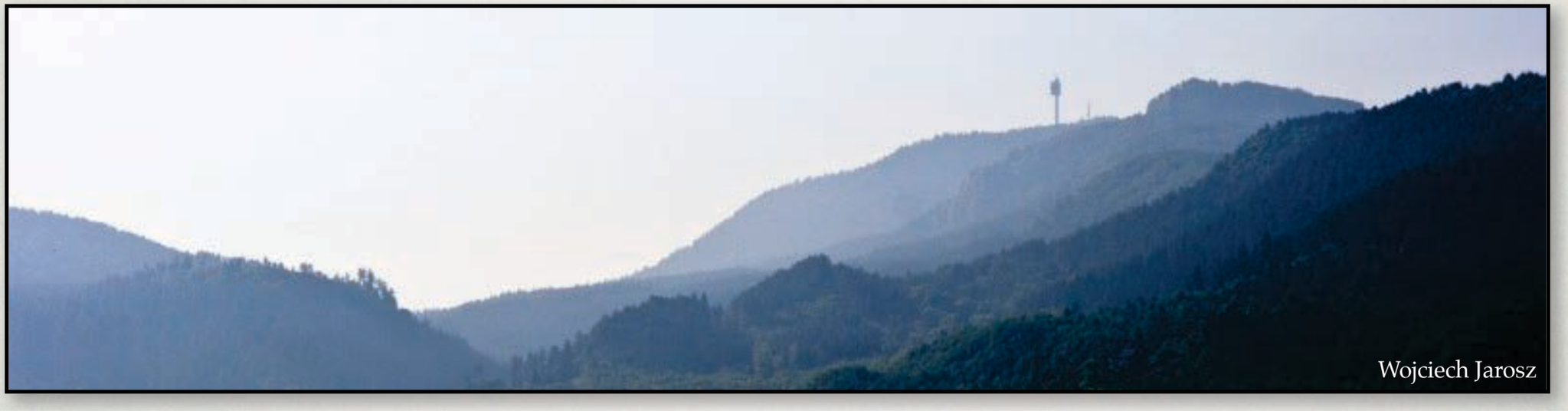

SIGGRAPH 2009 Course: Scattering

Thursday, August 6th, 12:45 - 16:30, Room 265-266

# Fog

# Clouds & Crespucular rays

# Aerial (Atmospheric)Perspective




# Leonardo da Vinci (1480)

"Thus, if one is to be five times as distant, make it five times bluer."
—*Treatise on Painting*, Leonardo Da Vinci, pp 295, circa 1480.

# Nebula

T.A.Rector (NOAO/AURA/NSF) and the Hubble Heritage Team (STScI/AURA/NASA)

# Outline

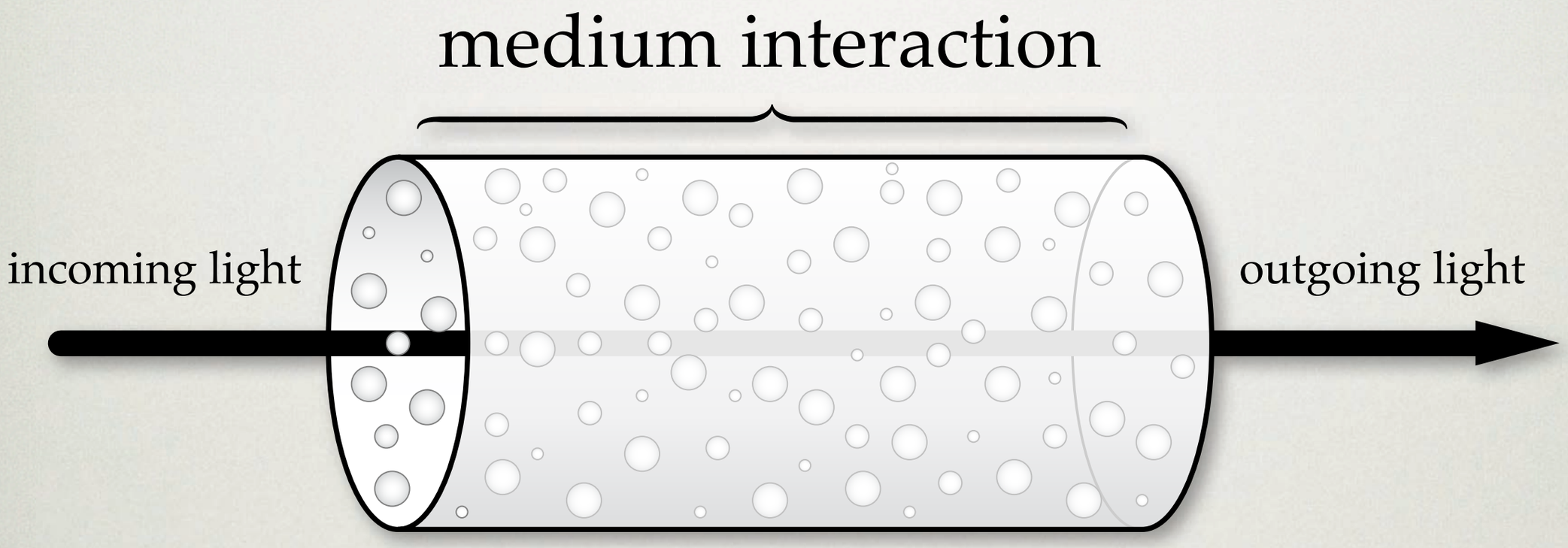
- Theoretical background
- Methods for rendering participating media

# Radiance

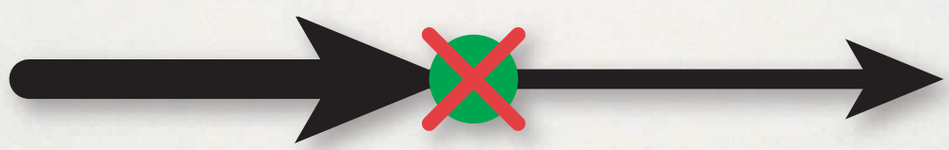
- The main quantity we are interested in for rendering is radiance:

$L(\mathbf{x} \rightarrow \vec{\omega})$, $L(\mathbf{x}, \vec{\omega})$: radiance, or "light"

# Participating Media



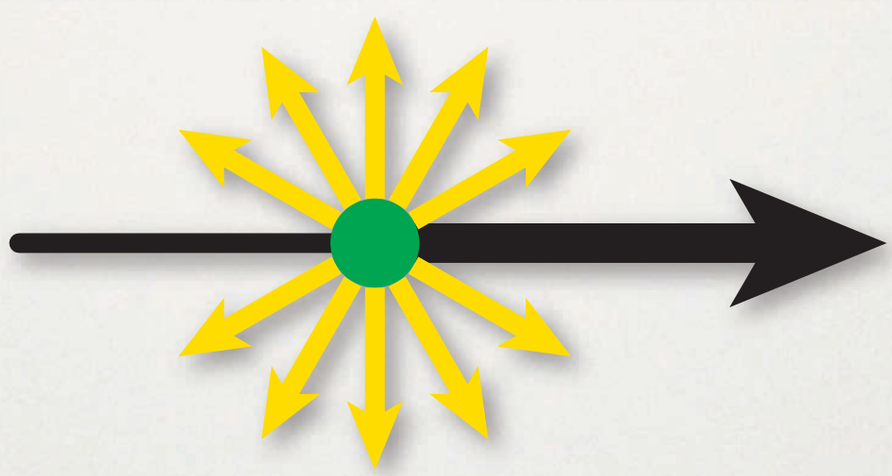
# Absorption



$$(\vec{\omega} \cdot \nabla_a)\, L(\mathbf{x} \rightarrow \vec{\omega}) = -\sigma_a(\mathbf{x})\, L(\mathbf{x} \rightarrow \vec{\omega})$$

$\sigma_a(\mathbf{x})$: absorption coefficient $[1/\mathrm{m}]$

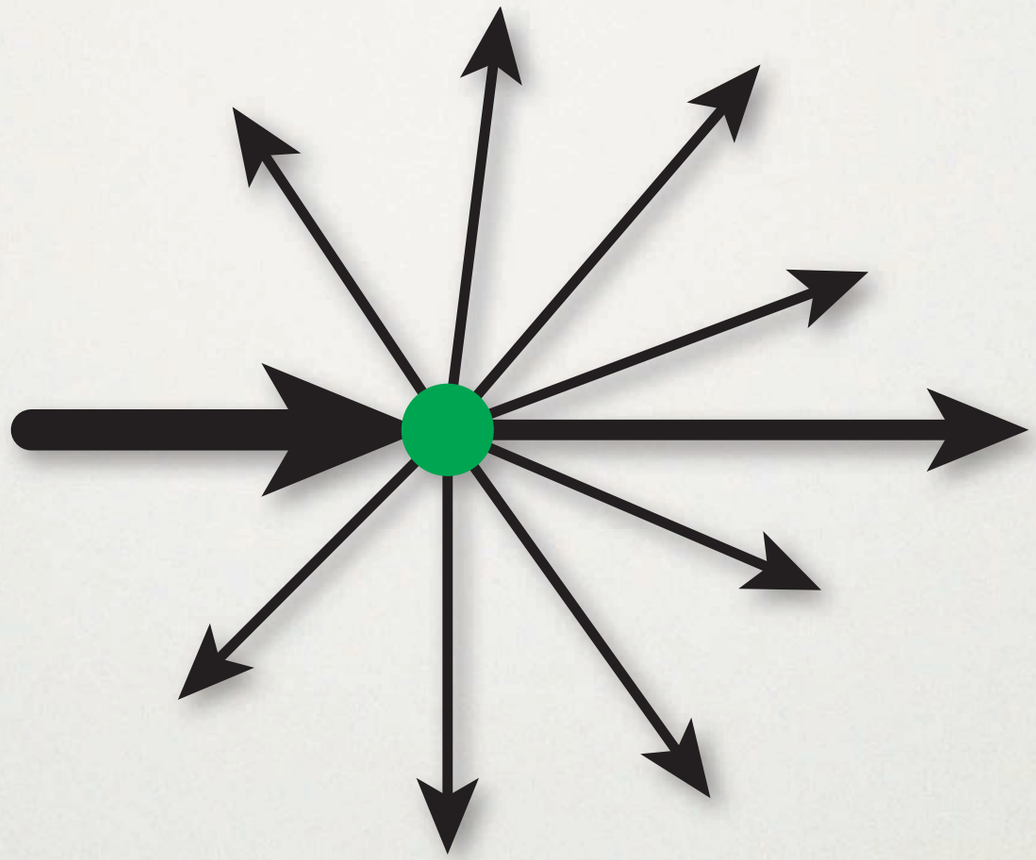
# Emission



$$(\vec{\omega} \cdot \nabla_e) L(\mathbf{x} \rightarrow \vec{\omega}) = \sigma_a(\mathbf{x}) L_e(\mathbf{x} \rightarrow \vec{\omega})$$

$\sigma_a(\mathbf{x})$: absorption coefficient [1/m]

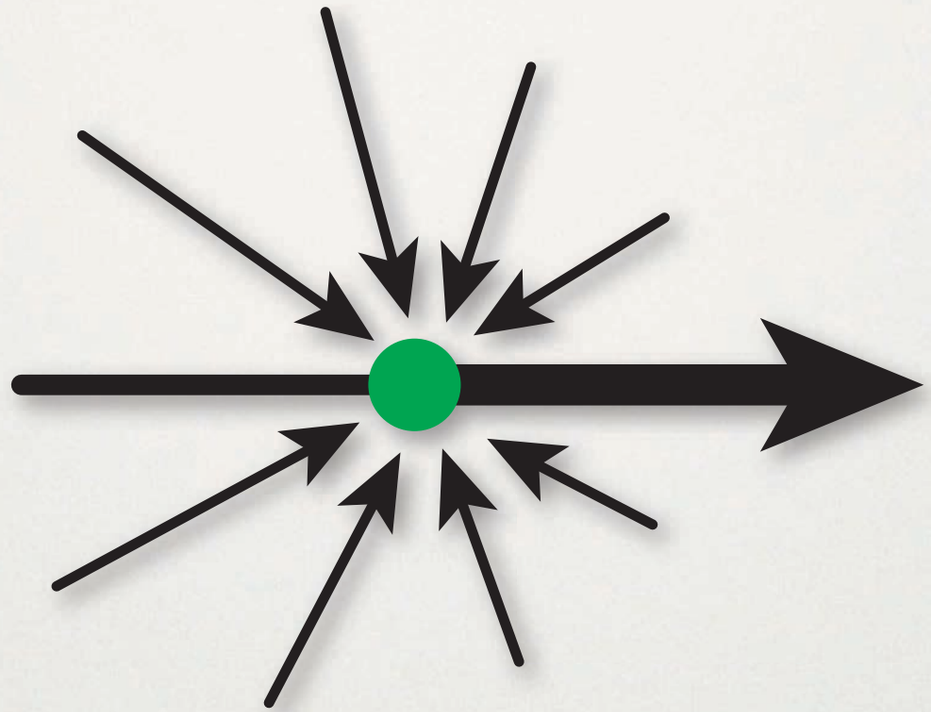
# Out-Scattering



$$(\vec{\omega} \cdot \nabla_o) L(\mathbf{x} \rightarrow \vec{\omega}) = -\sigma_s(\mathbf{x})\, L(\mathbf{x} \rightarrow \vec{\omega})$$

$\sigma_s(\mathbf{x})$: scattering coefficient $[1/\mathrm{m}]$

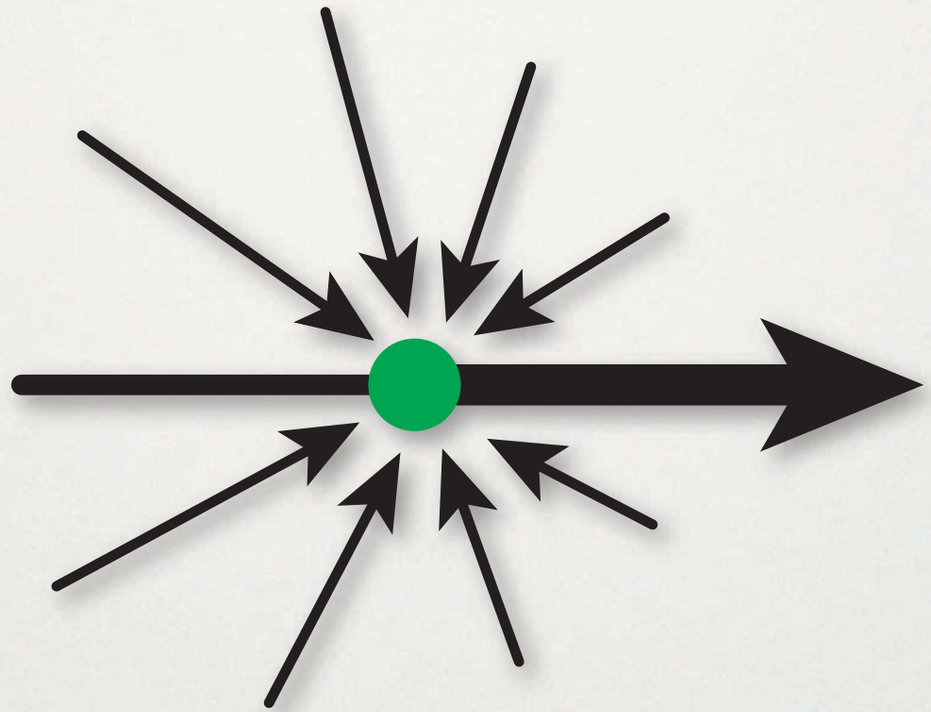
# In-Scattering



$$(\vec{\omega} \cdot \nabla_i) L(\mathbf{x} \rightarrow \vec{\omega}) = \sigma_s(\mathbf{x})\, L_i(\mathbf{x} \rightarrow \vec{\omega})$$

$\sigma_s(\mathbf{x})$: scattering coefficient $[1/\mathrm{m}]$

# In-Scattering



$$(\vec{\omega} \cdot \nabla_i) L(\mathbf{x} \rightarrow \vec{\omega}) = \sigma_s(\mathbf{x})\, L_i(\mathbf{x} \rightarrow \vec{\omega})$$

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) = \int_{\Omega_{4\pi}} p(\mathbf{x}, \vec{\omega}', \vec{\omega})\, L(\mathbf{x} \leftarrow \vec{\omega}')\, d\vec{\omega}'$$

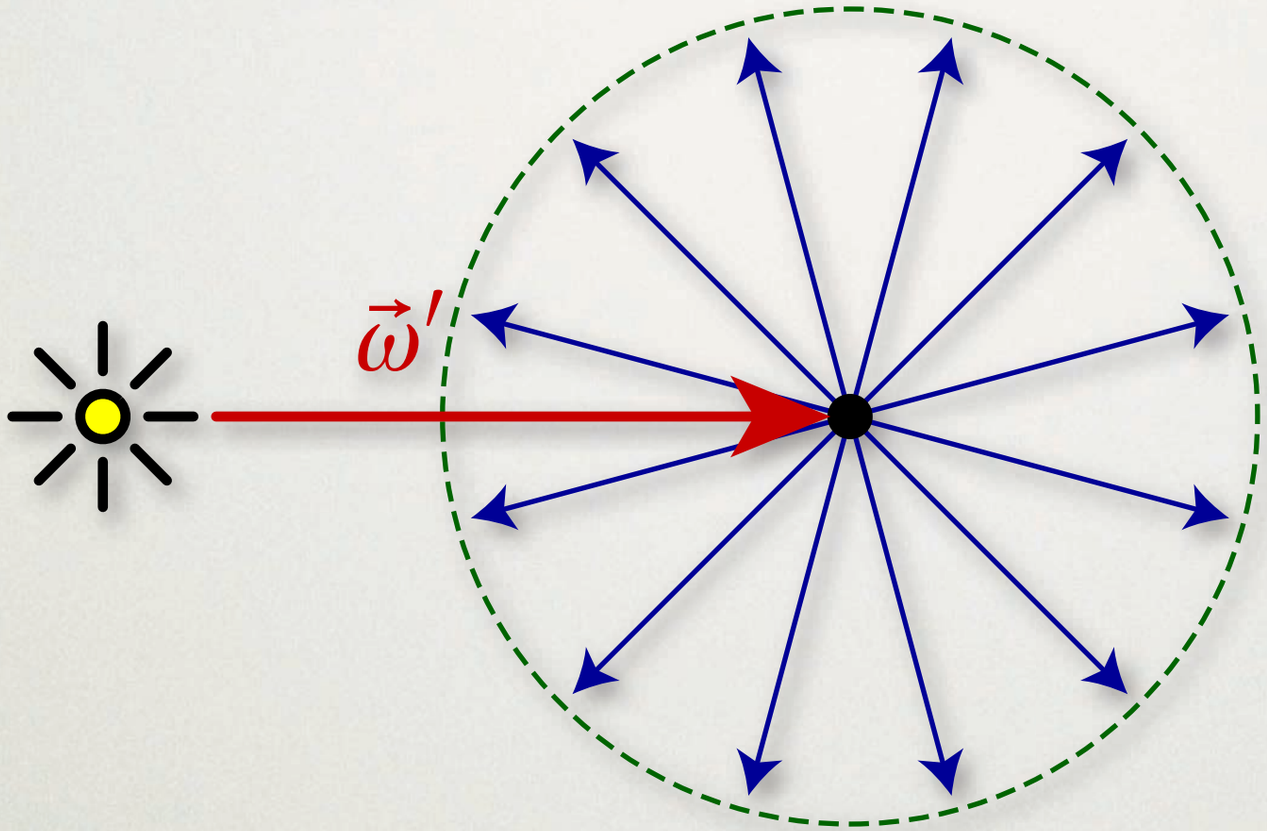
# The Phase Function

$$p(\mathbf{x}, \vec{\omega}' \rightarrow \vec{\omega})$$
or
$$p(\mathbf{x}, \vec{\omega}', \vec{\omega})$$

- Local, directional distribution of scattering
- Integrates to 1 over all directions:

$$\int_{\Omega_{4\pi}} p(\mathbf{x}, \vec{\omega}', \vec{\omega}) d\vec{\omega}' = 1$$

# The Phase Function




isotropic scattering

$$p(\mathbf{x}, \vec{\omega}', \vec{\omega}) = \frac{1}{4\pi}$$

# Anisotropic Scattering

- Anisotropy parameter g (average cosine):

$$g = \int_{\Omega_{4\pi}} p(\mathbf{x}, \vec{\omega}', \vec{\omega}) \cos(\theta) d\vec{\omega}'$$

where,

$$\cos(\theta) = \vec{\omega} \cdot \vec{\omega}'$$

# Anisotropic Scattering

- Anisotropy parameter g (average cosine):

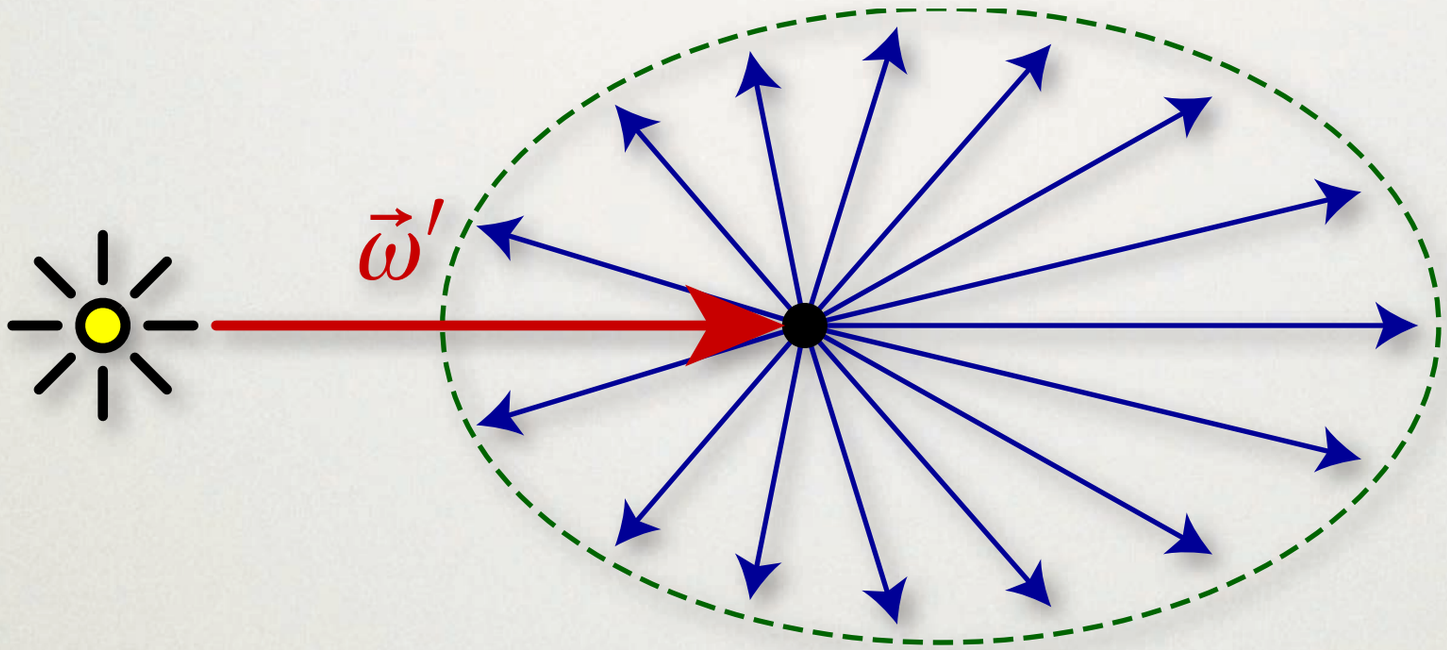
$$g = \int_{\Omega_{4\pi}} p(\mathbf{x}, \vec{\omega}', \vec{\omega}) \cos(\theta) d\vec{\omega}'$$

where,

$$\cos(\theta) = \vec{\omega} \cdot \vec{\omega}'$$

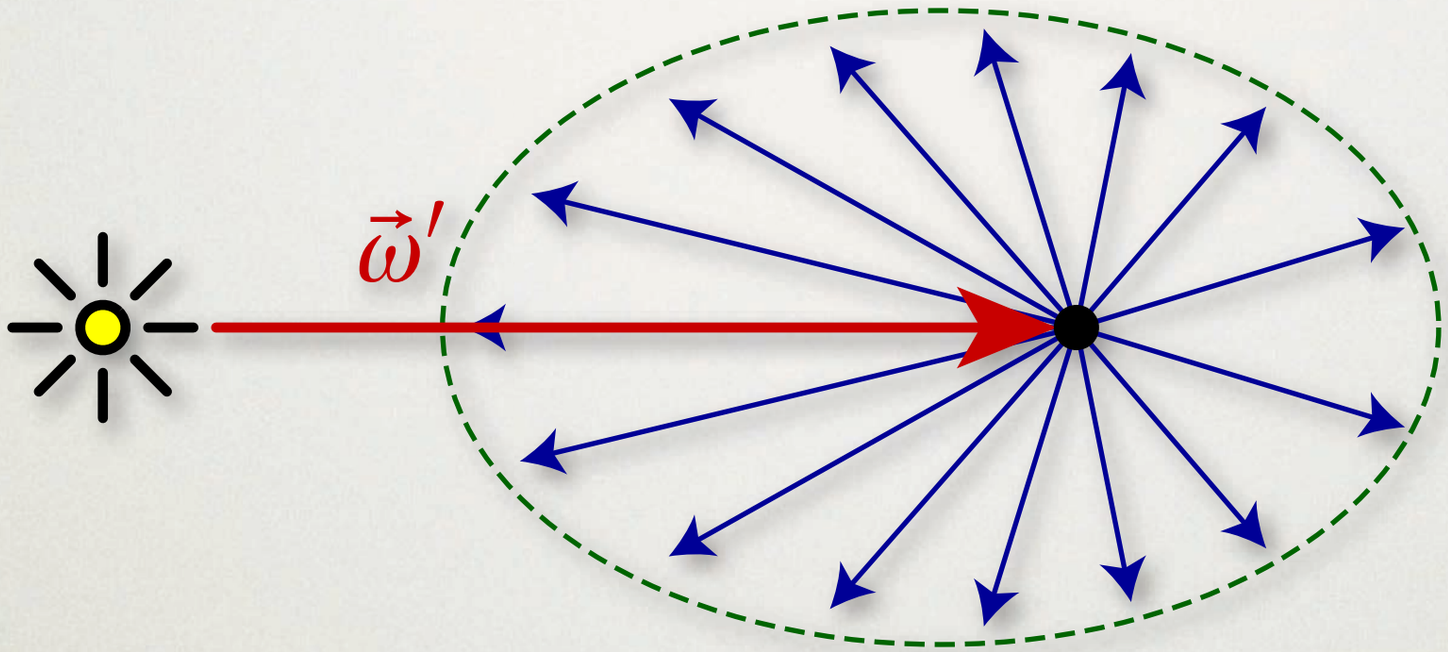
- g=0: isotropic scattering

# The Phase Function




g > 0: forward scattering

# The Phase Function




$g < 0$: backward scattering

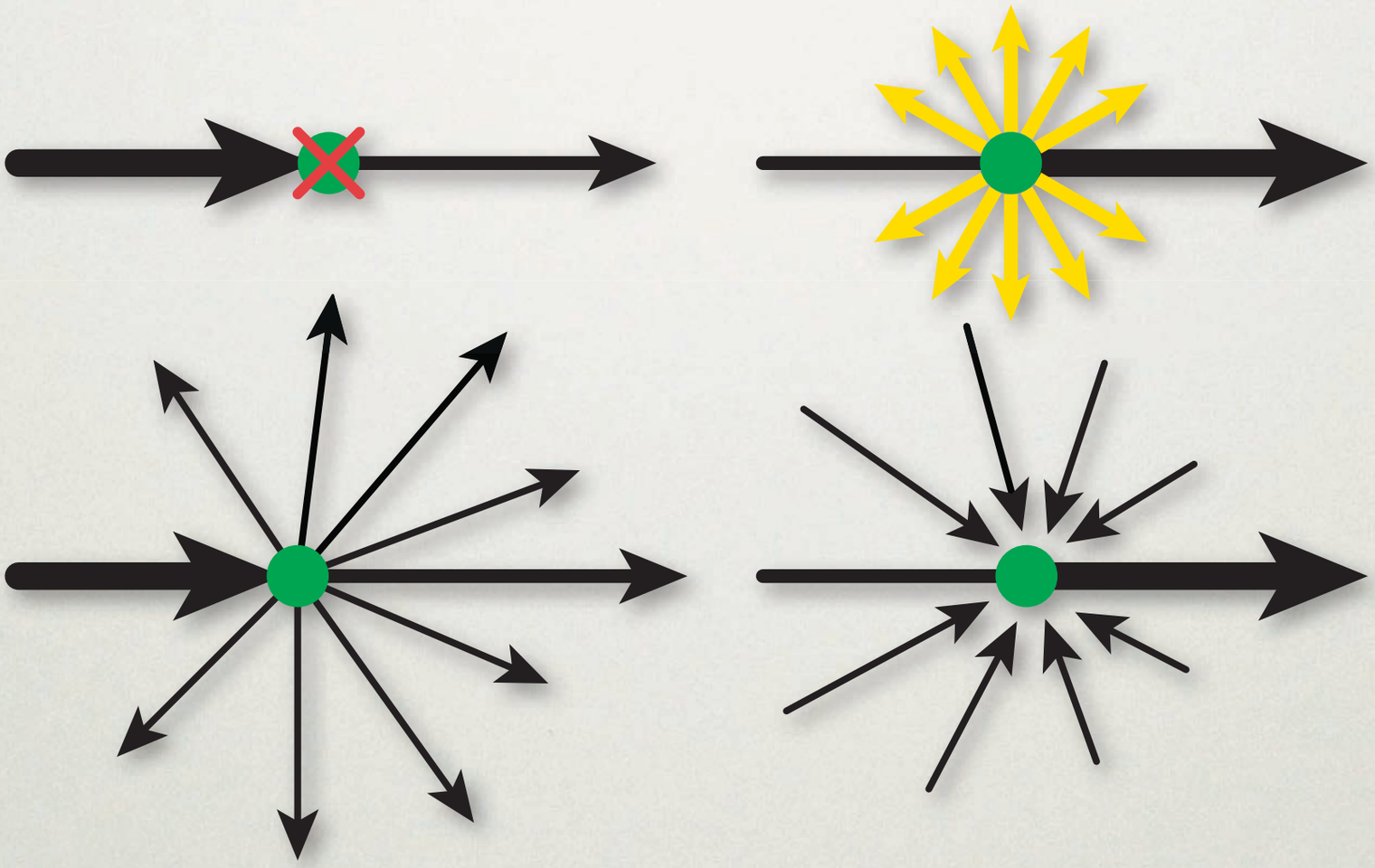
# The 4 Scattering Events

**Absorption**

$(\vec{\omega} \cdot \nabla_a) L(\mathbf{x} \rightarrow \vec{\omega})$

**Emission**

$(\vec{\omega} \cdot \nabla_e) L(\mathbf{x} \rightarrow \vec{\omega})$



**Out-scattering**

$(\vec{\omega} \cdot \nabla_o) L(\mathbf{x} \rightarrow \vec{\omega})$

**In-scattering**

$(\vec{\omega} \cdot \nabla_i) L(\mathbf{x} \rightarrow \vec{\omega})$

# Radiative Transport Eqn

$$(\vec{\omega} \cdot \nabla) L(\mathbf{x} \rightarrow \vec{\omega}) = -\underbrace{\sigma_a(\mathbf{x}) L(\mathbf{x} \rightarrow \vec{\omega})}_{\text{absorption}}$$

# Radiative Transport Eqn

$$(\vec{\omega}\cdot\nabla)L(\mathbf{x}\rightarrow\vec{\omega}) = -\underbrace{\sigma_a(\mathbf{x})L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{absorption}} - \underbrace{\sigma_s(\mathbf{x})\,L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{out-scattering}}$$

# Radiative Transport Eqn

$$(\vec{\omega}\cdot\nabla)L(\mathbf{x}\rightarrow\vec{\omega}) = -\underbrace{\underbrace{\sigma_a(\mathbf{x})L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{absorption}} - \underbrace{\sigma_s(\mathbf{x})\,L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{out-scattering}}}_{\text{extinction}}$$

# Radiative Transport Eqn

$$(\vec{\omega}\cdot\nabla)L(\mathbf{x}\rightarrow\vec{\omega}) = -\underbrace{\underbrace{\sigma_a(\mathbf{x})L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{absorption}} - \underbrace{\sigma_s(\mathbf{x})\,L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{out-scattering}}}_{\text{extinction}} + \underbrace{\sigma_a(\mathbf{x})\,L_e(\mathbf{x}\rightarrow\vec{\omega})}_{\text{emission}}$$

# Radiative Transport Eqn

$$(\vec{\omega}\cdot\nabla)L(\mathbf{x}\rightarrow\vec{\omega}) = -\underbrace{\underbrace{\sigma_a(\mathbf{x})L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{absorption}} - \underbrace{\sigma_s(\mathbf{x})\,L(\mathbf{x}\rightarrow\vec{\omega})}_{\text{out-scattering}}}_{\text{extinction}} +$$

$$\underbrace{\sigma_a(\mathbf{x})\,L_e(\mathbf{x}\rightarrow\vec{\omega})}_{\text{emission}} + \underbrace{\sigma_s(\mathbf{x})\,L_i(\mathbf{x}\rightarrow\vec{\omega})}_{\text{in-scattering}}$$

# Radiative Transport Eqn

$$(\vec{\omega} \cdot \nabla) L(\mathbf{x} \rightarrow \vec{\omega}) = -\underbrace{\sigma_t(\mathbf{x}) L(\mathbf{x} \rightarrow \vec{\omega})}_{\text{extinction}} +$$

$$\underbrace{\sigma_a(\mathbf{x})\, L_e(\mathbf{x} \rightarrow \vec{\omega})}_{\text{emission}} + \underbrace{\sigma_s(\mathbf{x})\, L_i(\mathbf{x} \rightarrow \vec{\omega})}_{\text{in-scattering}}$$

# Radiative Transport Eqn

$$(\vec{\omega} \cdot \nabla) L(\mathbf{x} \rightarrow \vec{\omega}) = -\underbrace{\sigma_t(\mathbf{x}) L(\mathbf{x} \rightarrow \vec{\omega})}_{\text{extinction}} +$$

$$\underbrace{\sigma_a(\mathbf{x})\, L_e(\mathbf{x} \rightarrow \vec{\omega})}_{\text{emission}} + \underbrace{\sigma_s(\mathbf{x})\, L_i(\mathbf{x} \rightarrow \vec{\omega})}_{\text{in-scattering}}$$

$$\sigma_t(\mathbf{x}) = \sigma_a(\mathbf{x}) + \sigma_s(\mathbf{x})$$

extinction coefficient

# Volume Rendering Eqn

$$L(\mathbf{x} \leftarrow \vec{\omega}) = \underbrace{T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)\, L(\mathbf{x}_s \rightarrow -\vec{\omega})}_{\text{reduced surface radiance}} +$$

$$\underbrace{\int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\, \sigma_a(\mathbf{x})\, L_e(\mathbf{x} \rightarrow -\vec{\omega})\, dt}_{\text{accumulated emitted radiance}} +$$

$$\underbrace{\int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\, \sigma_s(\mathbf{x}_t)\, L_i(\mathbf{x}_t \rightarrow -\vec{\omega})\, dt}_{\text{accumulated in-scattered radiance}}$$

# Volume Rendering Eqn

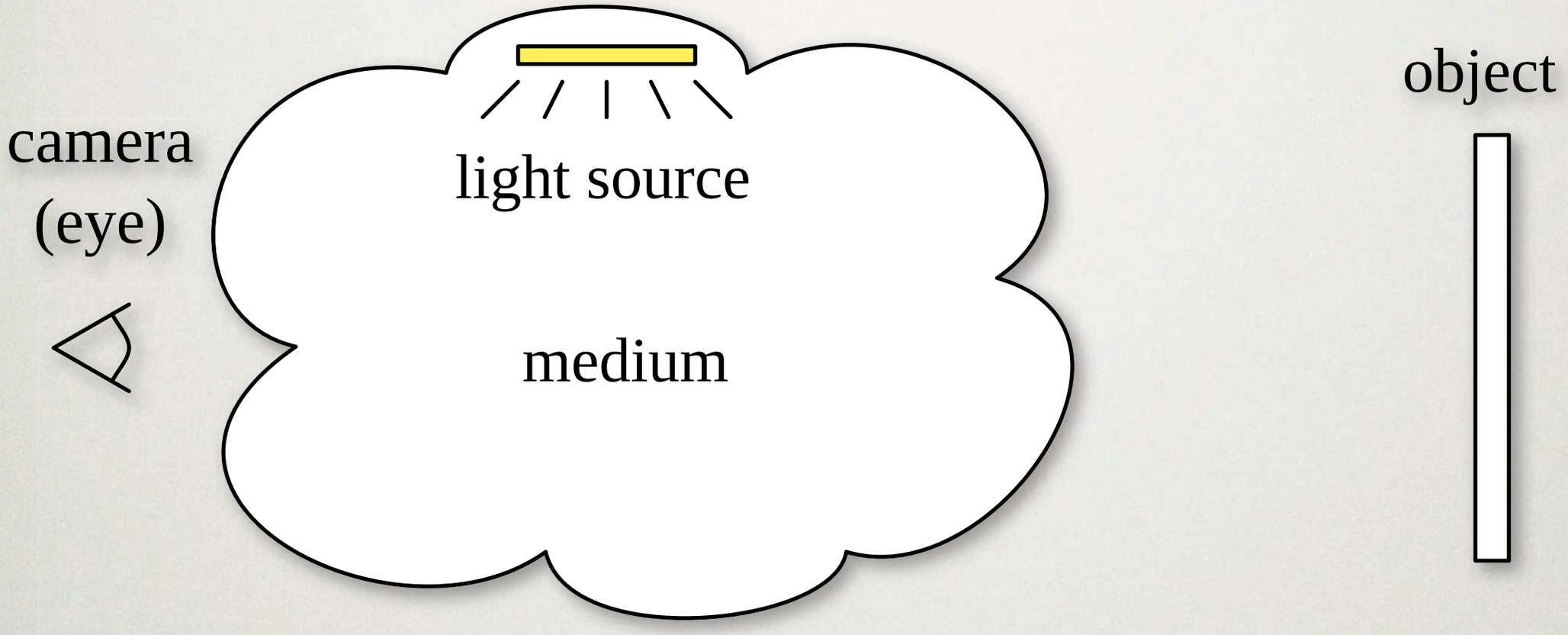
$$L(\mathbf{x} \leftarrow \vec{\omega}) = \underbrace{T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)\, L(\mathbf{x}_s \rightarrow -\vec{\omega})}_{\text{reduced surface radiance}} +$$

$$\underbrace{\int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\, \sigma_s(\mathbf{x}_t)\, L_i(\mathbf{x}_t \rightarrow -\vec{\omega})\, dt}_{\text{accumulated in-scattered radiance}}$$

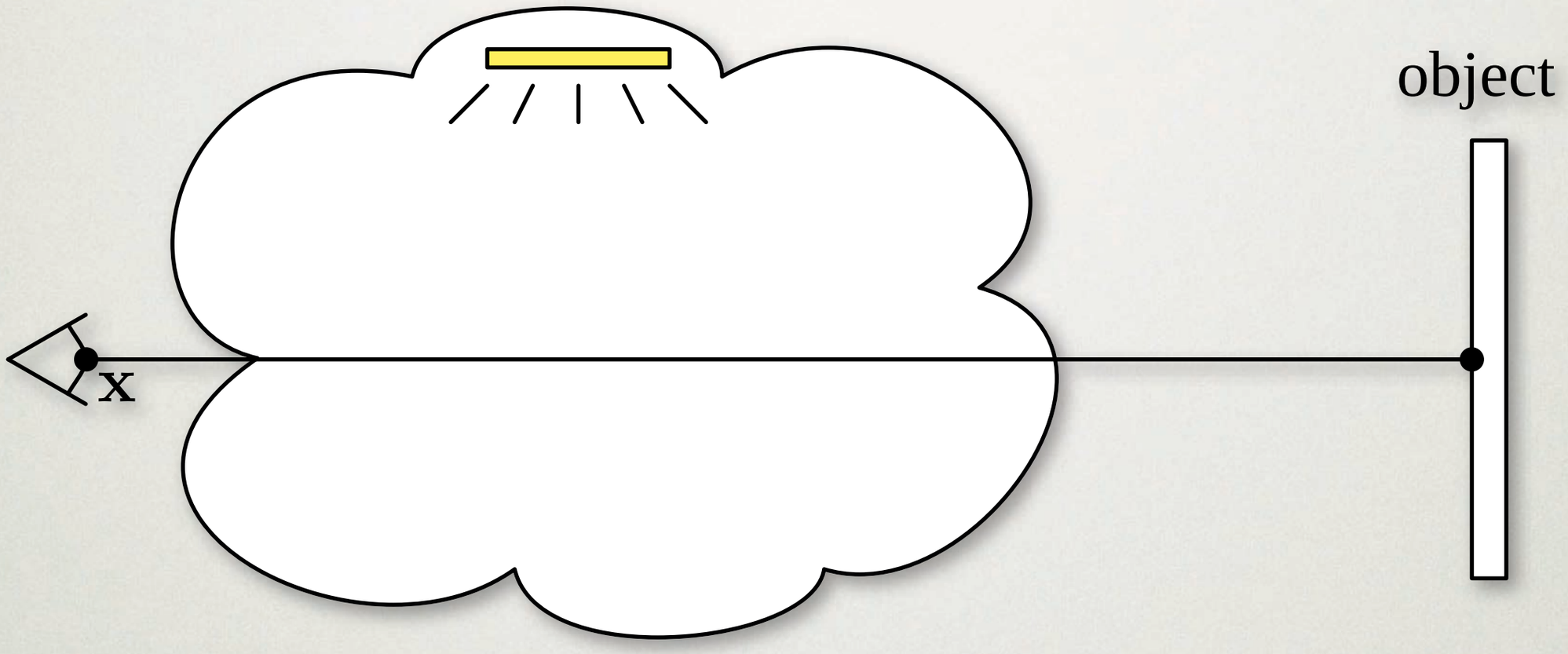
# Volume Rendering Eqn

$$L(\mathbf{x} \leftarrow \vec{\omega}) = \underbrace{T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)\, L(\mathbf{x}_s \rightarrow -\vec{\omega})}_{\text{reduced surface radiance}} + \underbrace{\int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\, \sigma_s(\mathbf{x}_t)\, L_i(\mathbf{x}_t \rightarrow -\vec{\omega})\, dt}_{\text{accumulated in-scattered radiance}}$$

# Scene with Medium

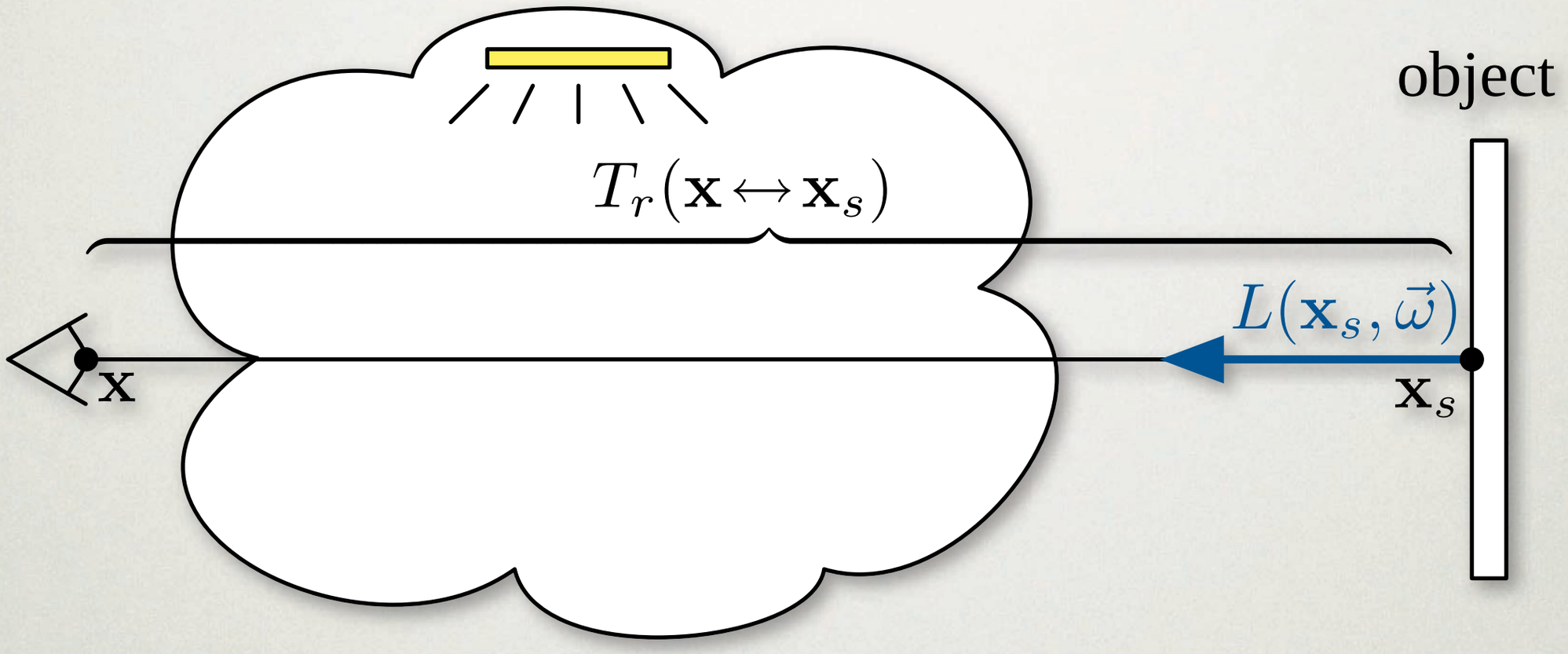


# Volume Rendering Eqn




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

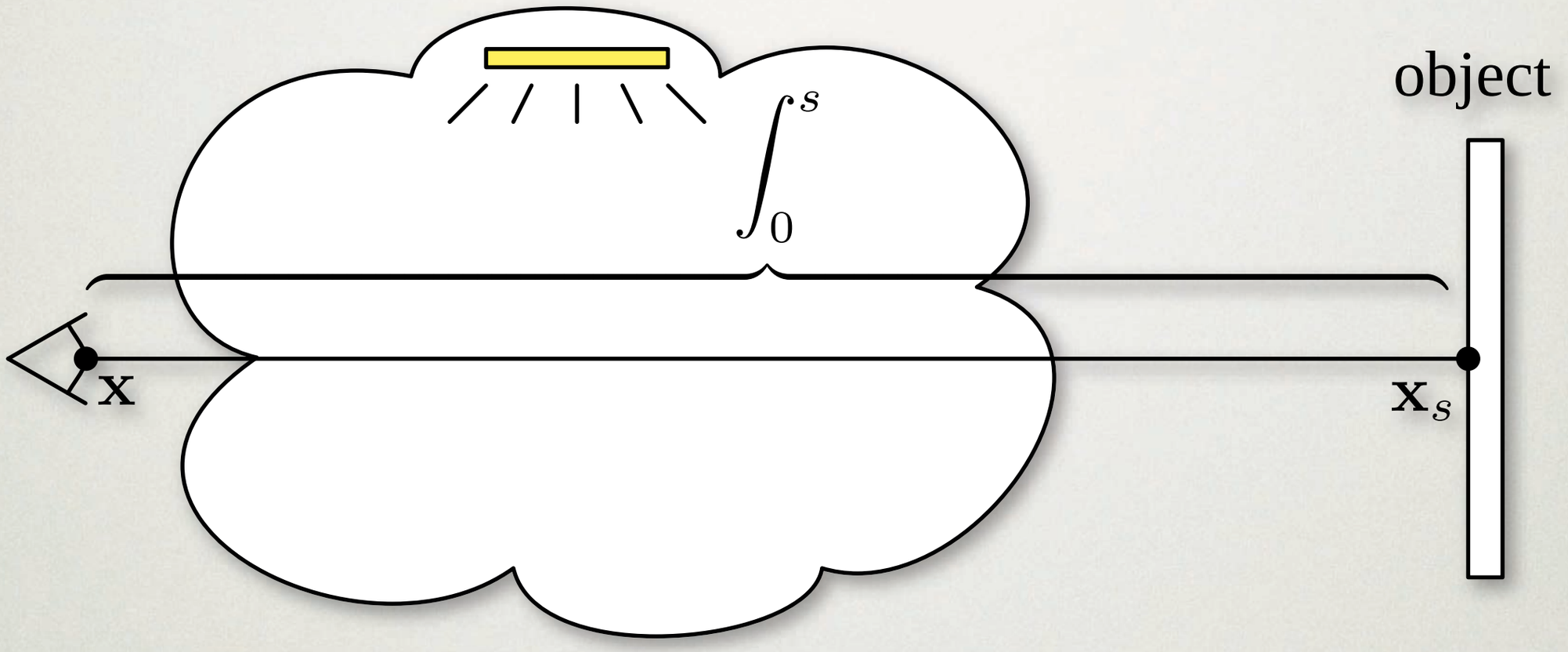
# Volume Rendering Eqn




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

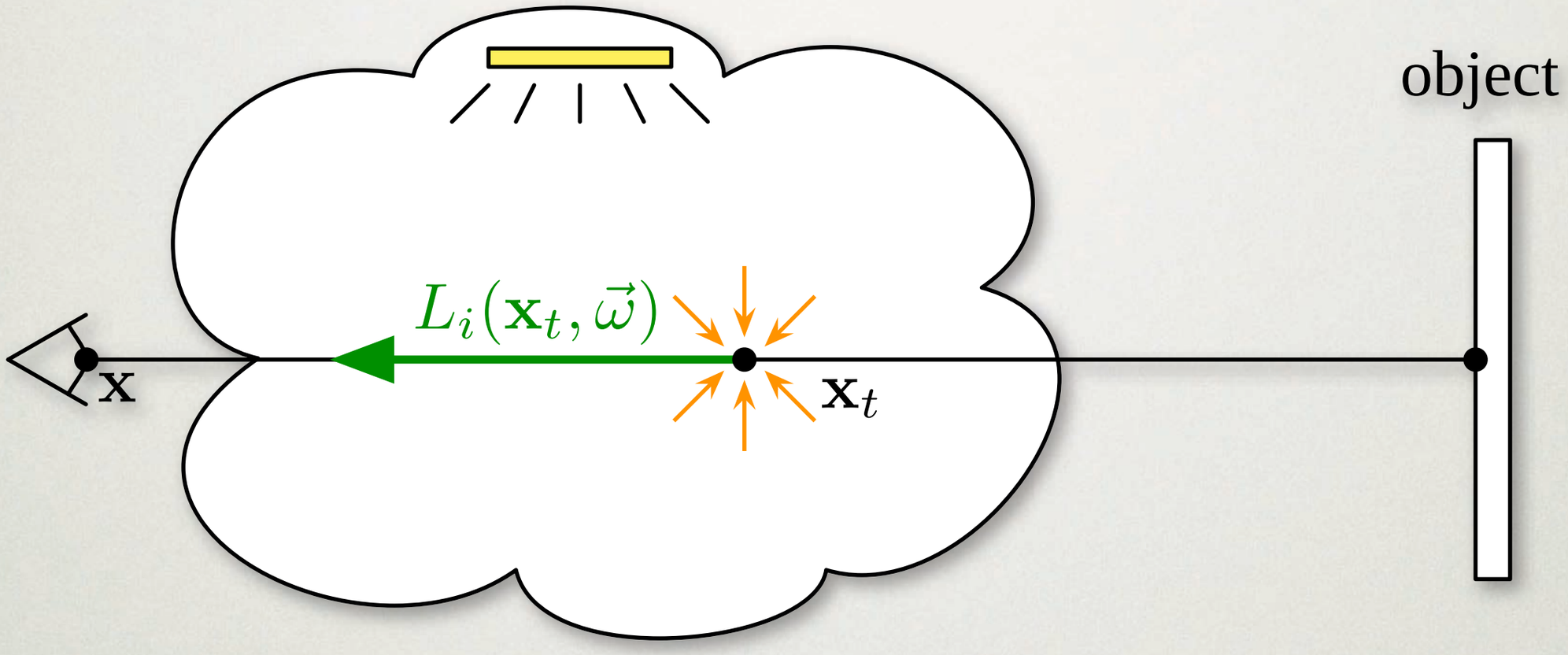
Transmittance: $T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) = \exp\left(-\int_0^s \sigma_t(\mathbf{x} + t\vec{\omega})\, dt\right)$

# Volume Rendering Eqn




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

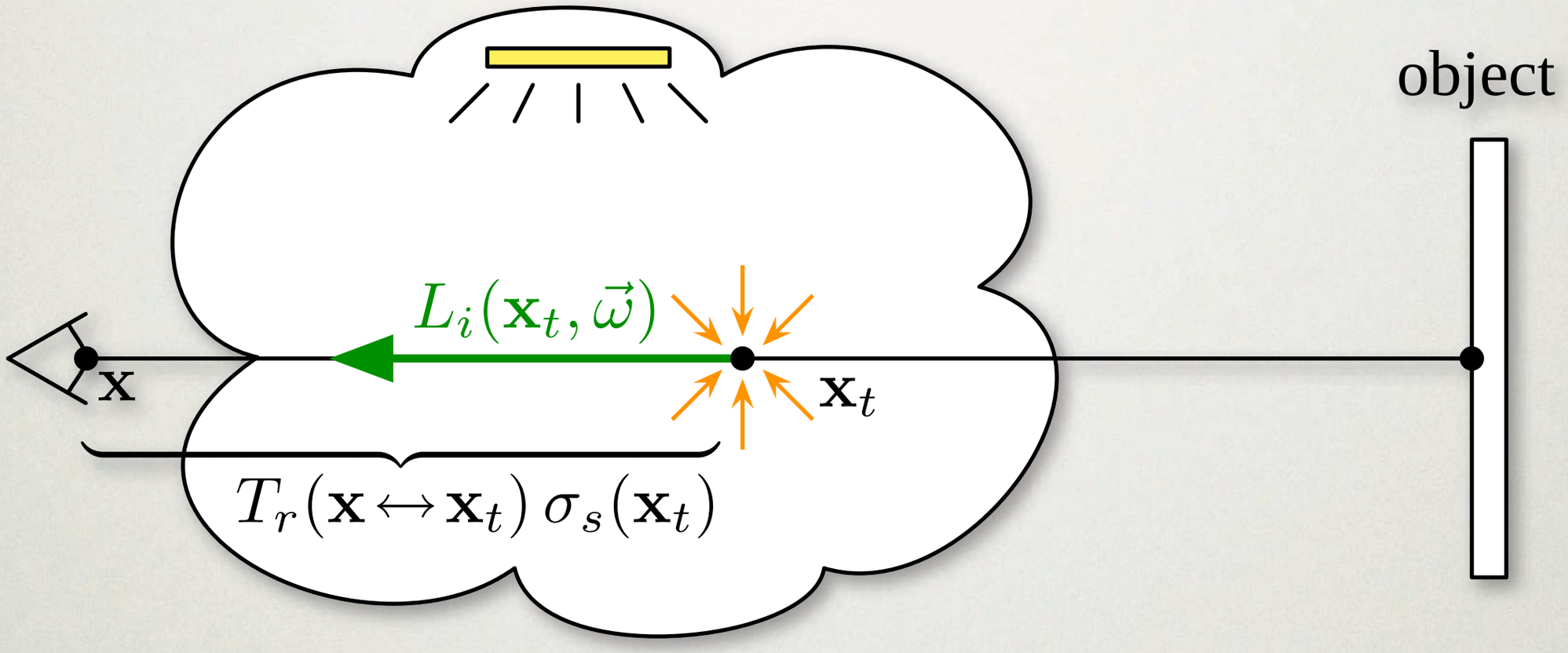
# Volume Rendering Eqn




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\,dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

$$L_i(\mathbf{x}_t, \vec{\omega}) = \int_{\Omega_{4\pi}} p(\mathbf{x}_t, \vec{\omega}_t, \vec{\omega})L(\mathbf{x}_t, \vec{\omega}_t)\; d\omega_t$$

# Volume Rendering Eqn




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

# Media Properties

$\sigma_s(\mathbf{x})$: scattering coefficient $[1/m]$

$\sigma_a(\mathbf{x})$: absorption coefficient $[1/m]$

$p(\mathbf{x}, \vec{\omega}', \vec{\omega})$: phase function $[1/sr]$

# Media Properties

$\sigma_s(\mathbf{x})$: scattering coefficient $[1/m]$

$\sigma_a(\mathbf{x})$: absorption coefficient $[1/m]$

$p(\mathbf{x}, \vec{\omega}', \vec{\omega})$: phase function $[1/sr]$

**Homogeneous**
spatially constant

**Heterogeneous**
spatially varying

# Media Properties

$\sigma_s(\mathbf{x})$: scattering coefficient $[1/m]$

$\sigma_a(\mathbf{x})$: absorption coefficient $[1/m]$

$p(\mathbf{x}, \vec{\omega}', \vec{\omega})$: phase function $[1/sr]$

**Homogeneous**
spatially constant

**Heterogeneous**
spatially varying

**Isotropic**
directionally constant

**Anisotropic**
directionally varying

# Derived Properties

$\sigma_t(\mathbf{x}) = \sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})$: extinction coefficient $[1/m]$

# Derived Properties

$\sigma_t(\mathbf{x}) = \sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})$: extinction coefficient $[1/m]$

$\frac{1}{\sigma_t}$: mean-free path $[m]$

# Derived Properties

$\sigma_t(\mathbf{x}) = \sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})$: extinction coefficient $[1/m]$

$\frac{1}{\sigma_t}$: mean-free path $[m]$

$\frac{\sigma_s}{\sigma_t}$: scattering albedo $[none]$

# Derived Properties

$\sigma_t(\mathbf{x}) = \sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})$: extinction coefficient $[1/m]$

$\frac{1}{\sigma_t}$: mean-free path $[m]$

$\frac{\sigma_s}{\sigma_t}$: scattering albedo $[none]$

$T_r(\mathbf{x}' \leftrightarrow \mathbf{x})$: transmittance $[none]$

$$\exp\left(-\int_0^d \sigma_t(\mathbf{x} + t\vec{\omega})\, dt\right)$$

# Outline

- Theoretical background
- Methods for rendering participating media

# Available Techniques

## Rendering Participating Media

- “Ray Tracing Volume Densities.” Kajiya and Herzen. 1984.
- “The Rendering Equation.” Kajiya. 1986.
- “The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium.” Rushmeier and Torrance. 1987.
- “Efficient Light Propagation for Multiple Anisotropic Volume Scattering.” Max. 1994.
- “Multiple Scattering as a Diffusion Process.” Stam. 1995.
- “Rendering Participating Media with Bidirectional Path Tracing.” Lafortune and Willems. 1996.
- “Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps.” Jensen and Christensen. 1998.
- “Metropolis Light Transport for Participating Media.” Pauly et al. 2000.
- “Practical Rendering of Multiple Scattering Effects in Participating Media.” Premože et al. 2004.
- “Multidimensional Lightcuts.” Walter et al. 2006.
- “Radiance Caching for Participating Media.” Jarosz et al. 2008.
- “The Beam Radiance Estimate for Volumetric Photon Mapping.” Jarosz et al. 2008.

# Available Techniques

## Rendering Participating Media

- Path tracing
- Ray marching
- Metropolis
- Finite element methods (Radiosity)
- Diffusion
- Interpolation methods (Radiance caching)
- Density estimation methods (Photon mapping)
- VPL methods (Lightcut, Instant radiosity)

# Available Techniques

## Rendering Participating Media

- Path tracing
- Ray marching
- Metropolis
- Finite element methods (Radiosity)
- Diffusion
- Interpolation methods (Radiance caching)
- Density estimation methods (Photon mapping)
- VPL methods (Lightcut, Instant radiosity)

# Available Techniques

## Rendering Participating Media

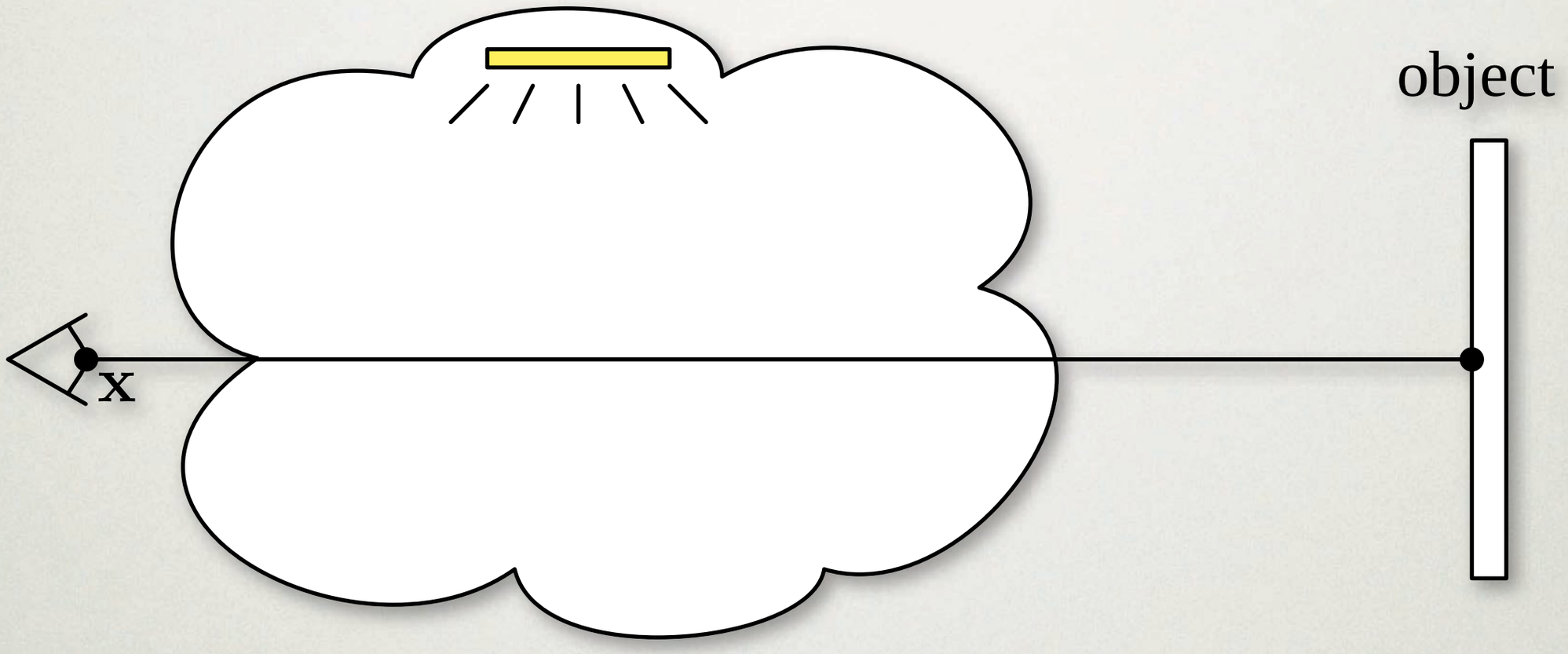
- Path tracing
- Ray marching
- Metropolis
- Finite element methods (Radiosity)
- Diffusion
- Interpolation methods (Radiance caching)
- Density estimation methods (Photon mapping)
- VPL methods (Lightcut, Instant radiosity)

# Available Techniques

## Rendering Participating Media

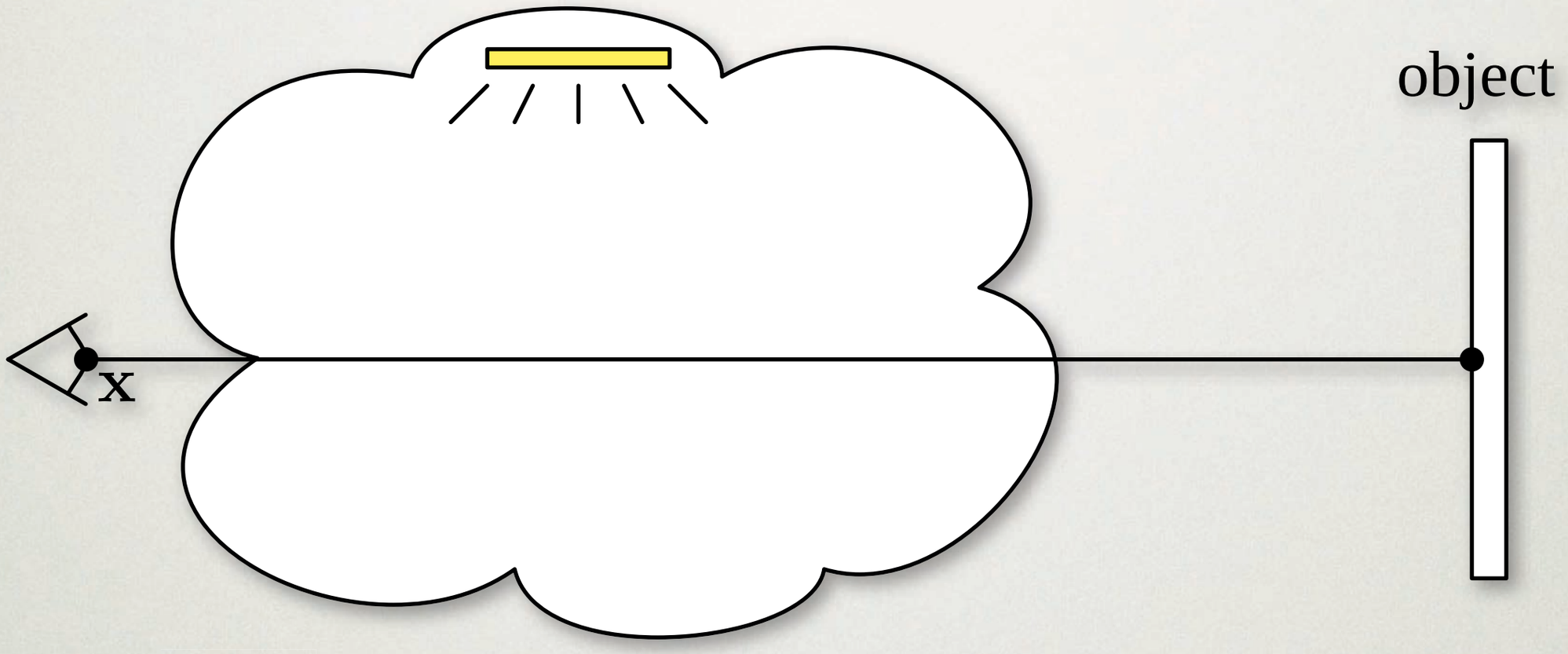
- Path tracing
- Ray marching
- Metropolis
- Finite element methods (Radiosity)
- Diffusion
- Interpolation methods (Radiance caching)
- Density estimation methods (Photon mapping)
- VPL methods (Lightcut, Instant radiosity)

# Ray Marching




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

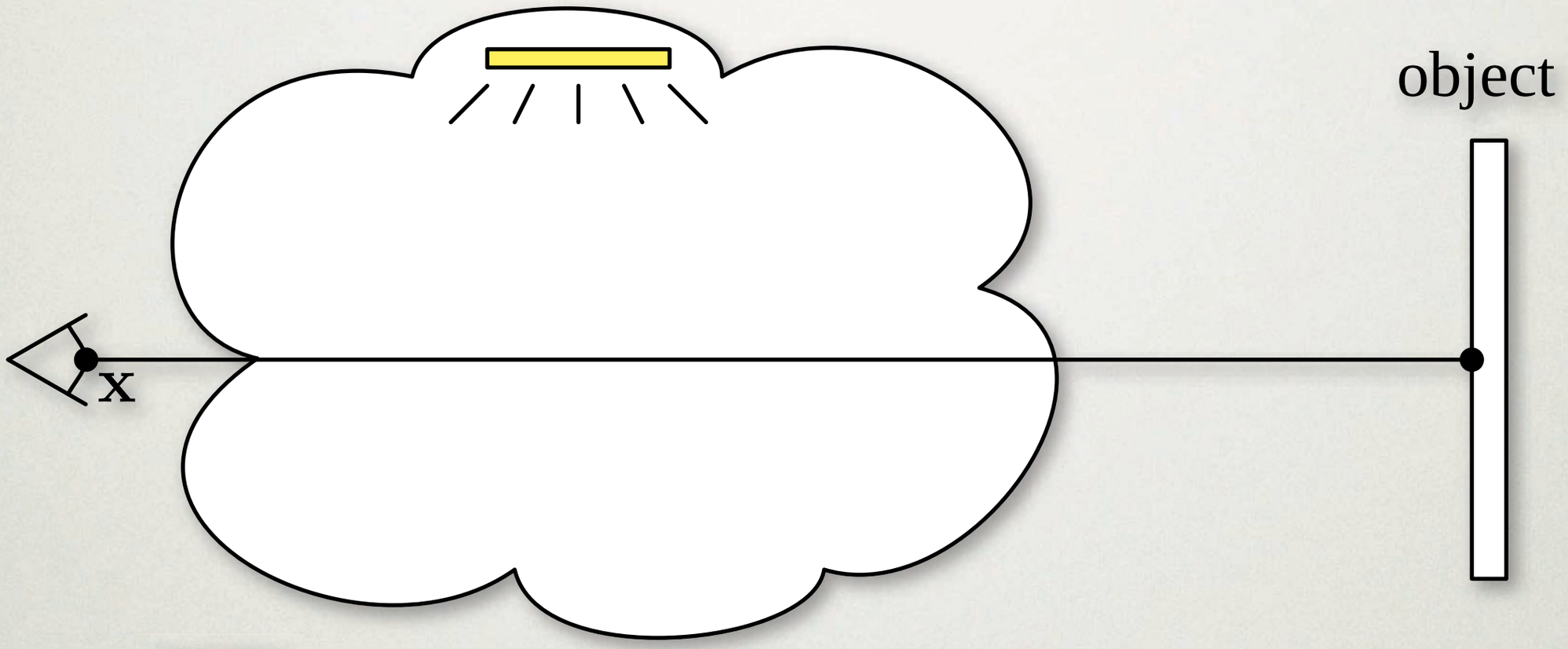
# Ray Marching




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \, dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

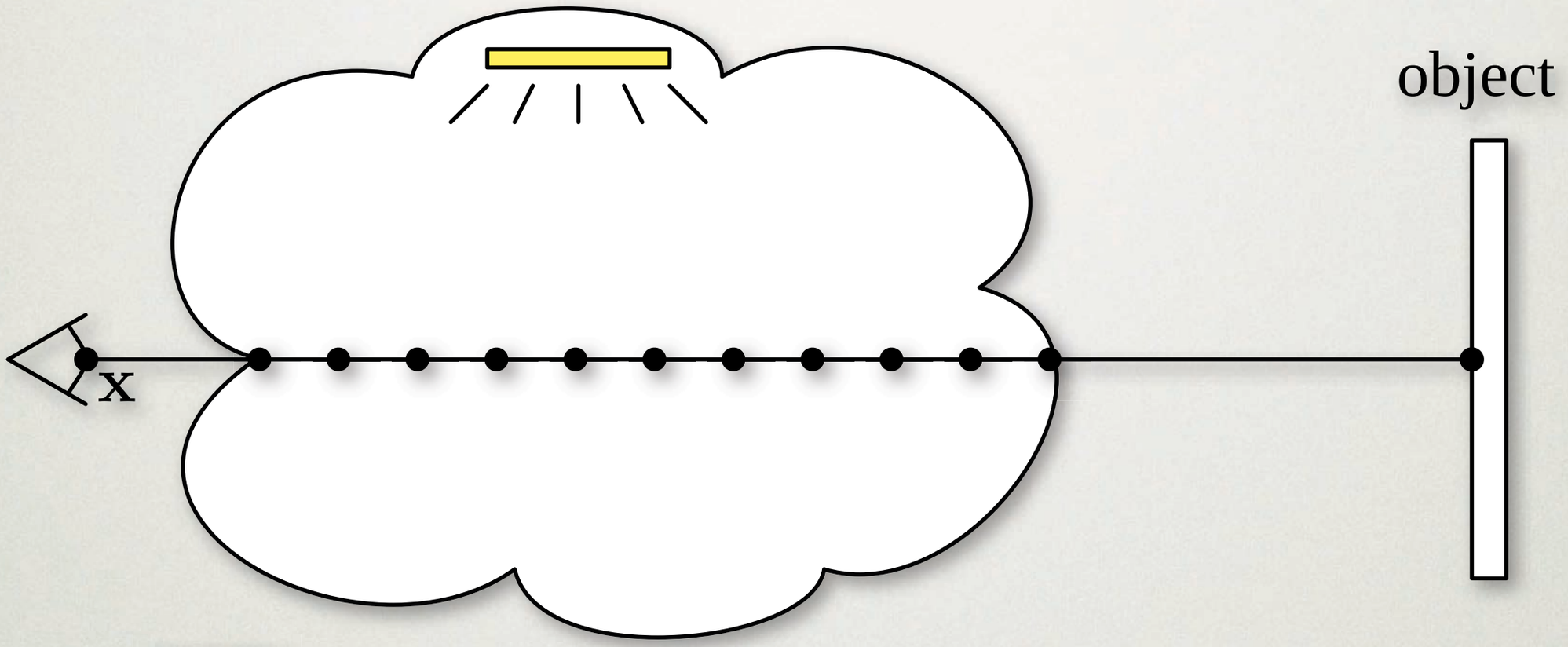
- approximate / compute using Riemann sum

# Ray Marching




$$L(\mathbf{x},\vec{\omega}) \approx \boxed{\sum_{t=0}^{S-1}} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\boxed{\Delta_t} + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

# Ray Marching




$$L(\mathbf{x},\vec{\omega}) \approx \boxed{\sum_{t=0}^{S-1}} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\boxed{\Delta_t} + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

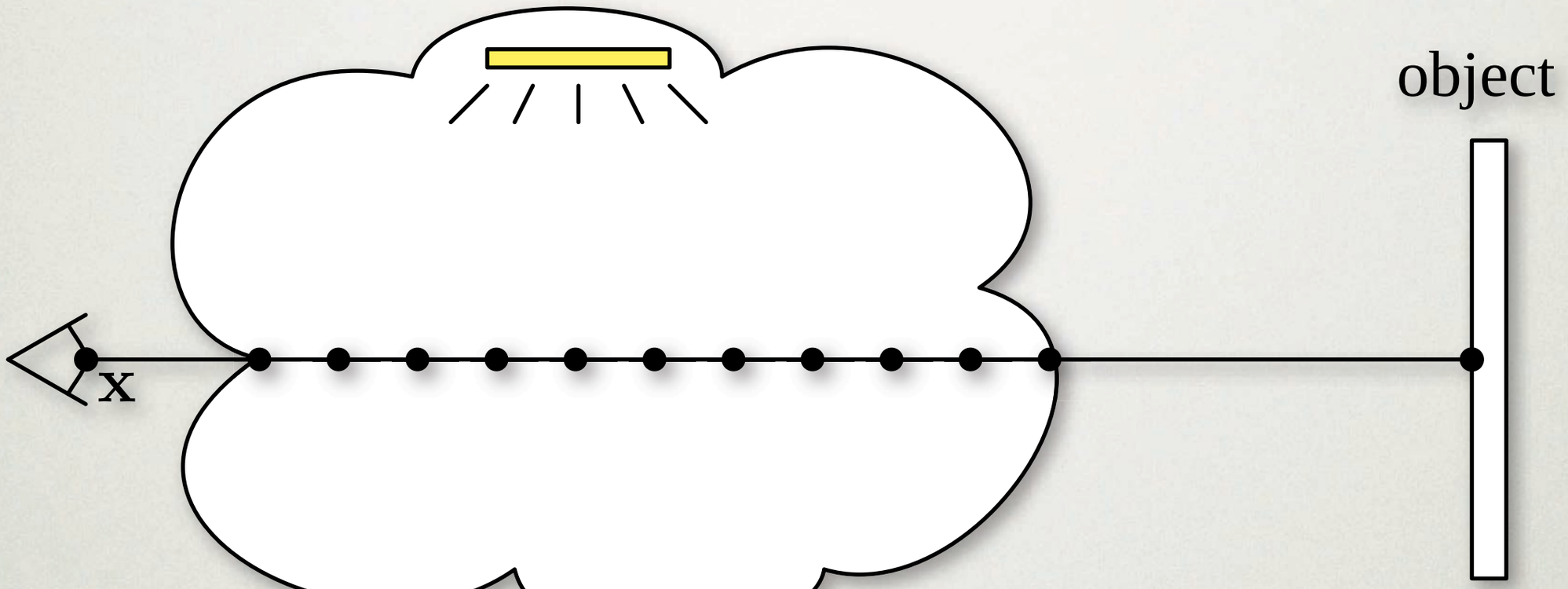
# Computing $T_R$

- In general:

$$T_r(\mathbf{x}' \leftrightarrow \mathbf{x}) = \exp\left(-\int_0^d \sigma_t(\mathbf{x} + t\vec{\omega})\, dt\right)$$

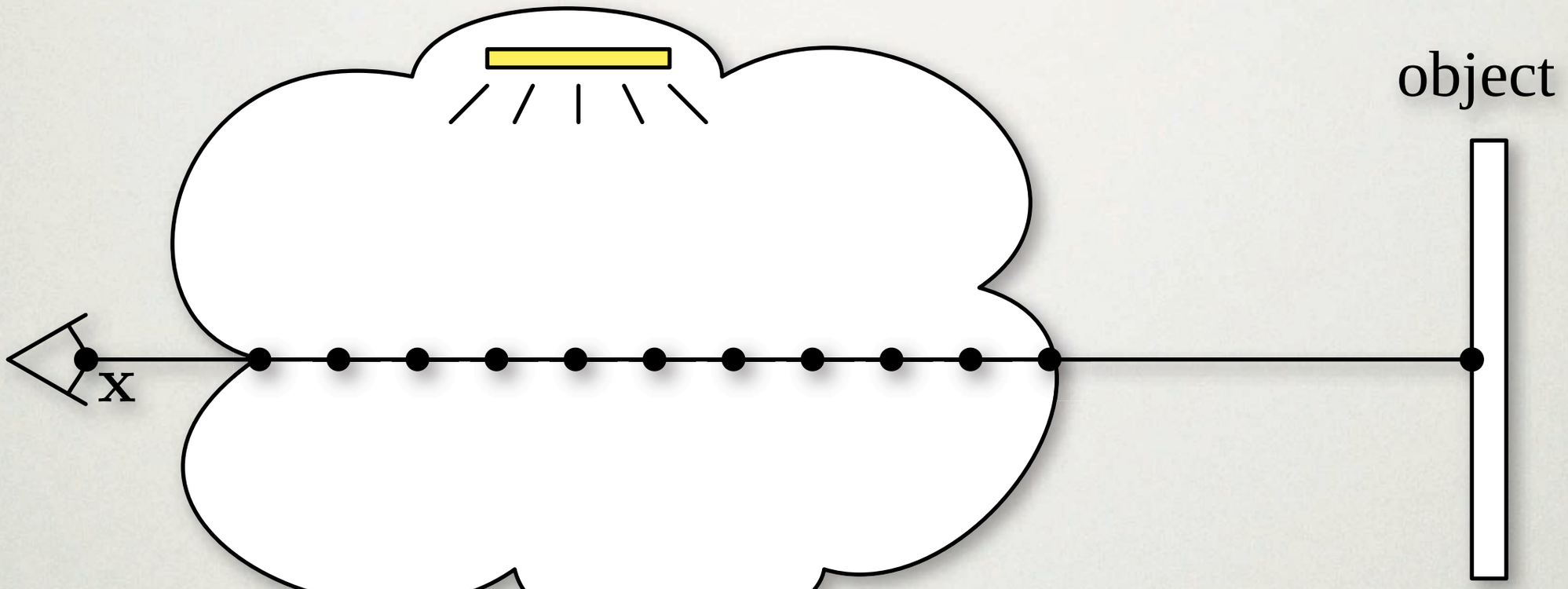
- In homogeneous medium:
- $T_r(\mathbf{x}' \leftrightarrow \mathbf{x}) = e^{\|\mathbf{x}' - \mathbf{x}\|\sigma_t}$

# Ray Marching (Homogeneous Media)




$$L(\mathbf{x}, \vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \Delta_t + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

# Ray Marching (Homogeneous Media)



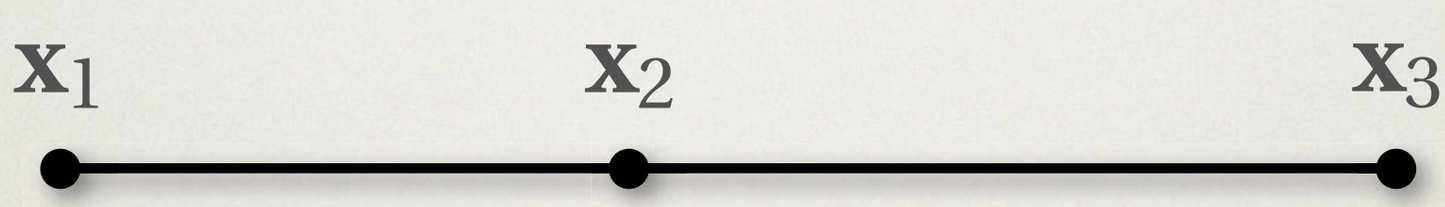

$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

$$\downarrow$$

$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} e^{\|\mathbf{x}_t-\mathbf{x}\|\sigma_t}\sigma_s L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + e^{\|\mathbf{x}_s-\mathbf{x}\|\sigma_t}L(\mathbf{x}_s,\vec{\omega})$$

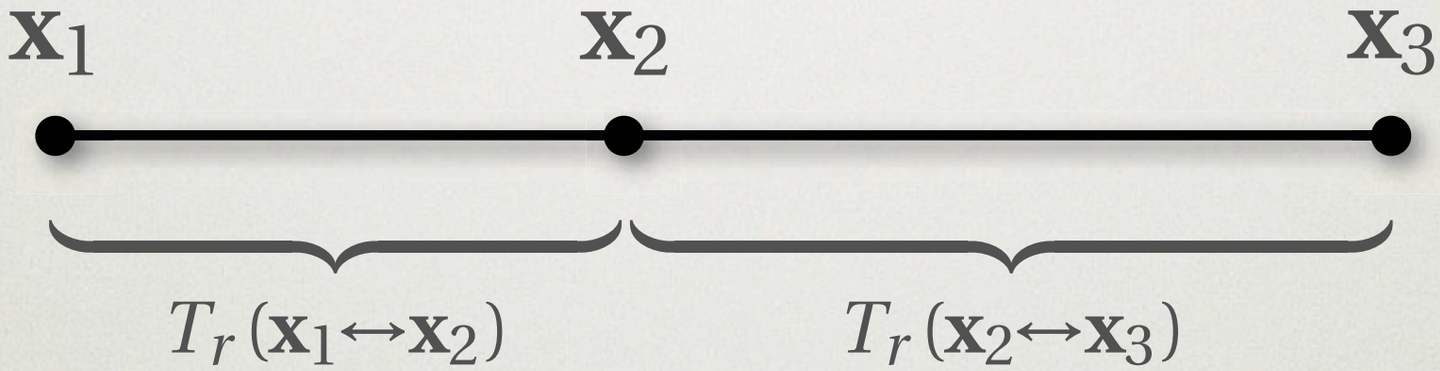
# Computing $T_R$

- In general, if $x_1$, $x_2$ and $x_3$ are collinear,
  - then: $T_r(\mathbf{x}_1 \leftrightarrow \mathbf{x}_3) = T_r(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2)\, T_r(\mathbf{x}_2 \leftrightarrow \mathbf{x}_3)$



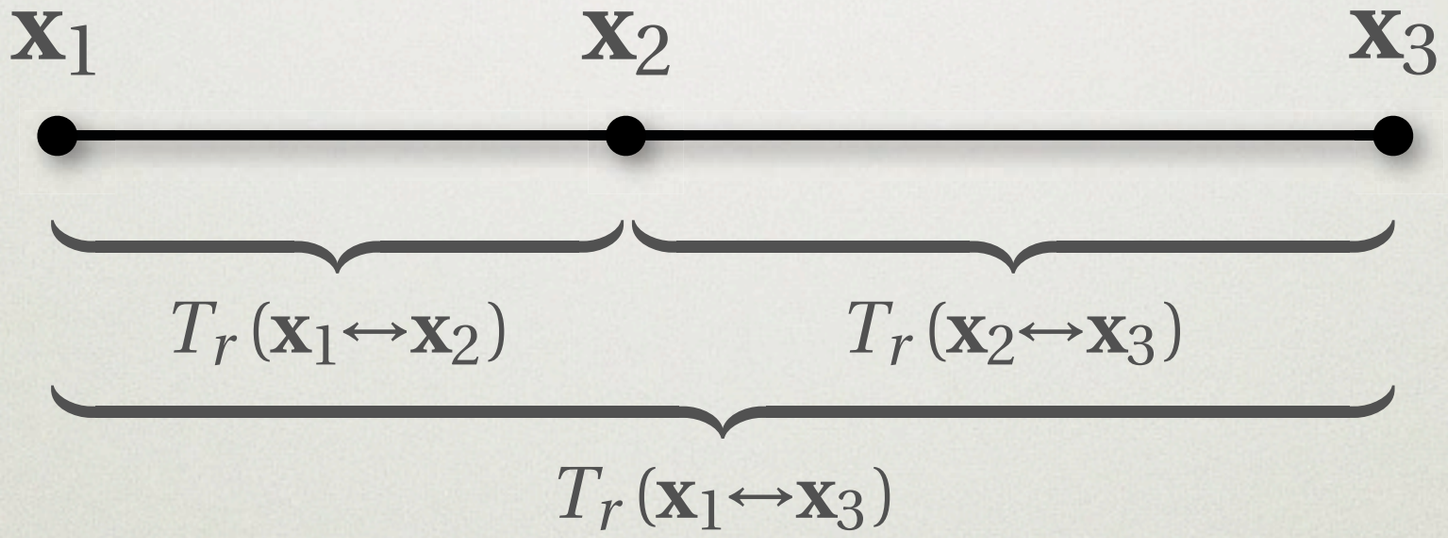
# Computing $T_R$

- In general, if $x_1$, $x_2$ and $x_3$ are collinear,
  - then: $T_r(\mathbf{x}_1 \leftrightarrow \mathbf{x}_3) = T_r(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2)\, T_r(\mathbf{x}_2 \leftrightarrow \mathbf{x}_3)$

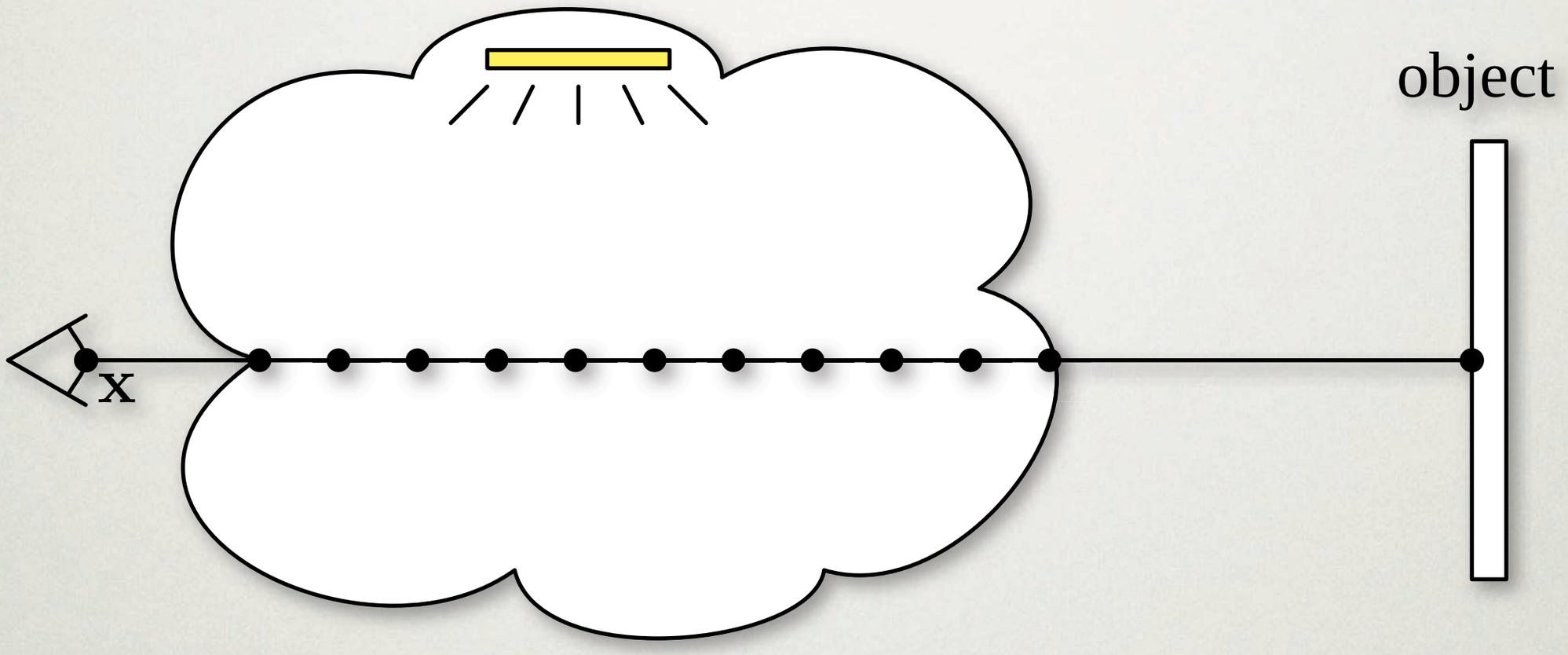


# Computing $T_R$

- In general, if $x_1$, $x_2$ and $x_3$ are collinear,
  - then: $T_r(\mathbf{x}_1 \leftrightarrow \mathbf{x}_3) = T_r(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2)\, T_r(\mathbf{x}_2 \leftrightarrow \mathbf{x}_3)$



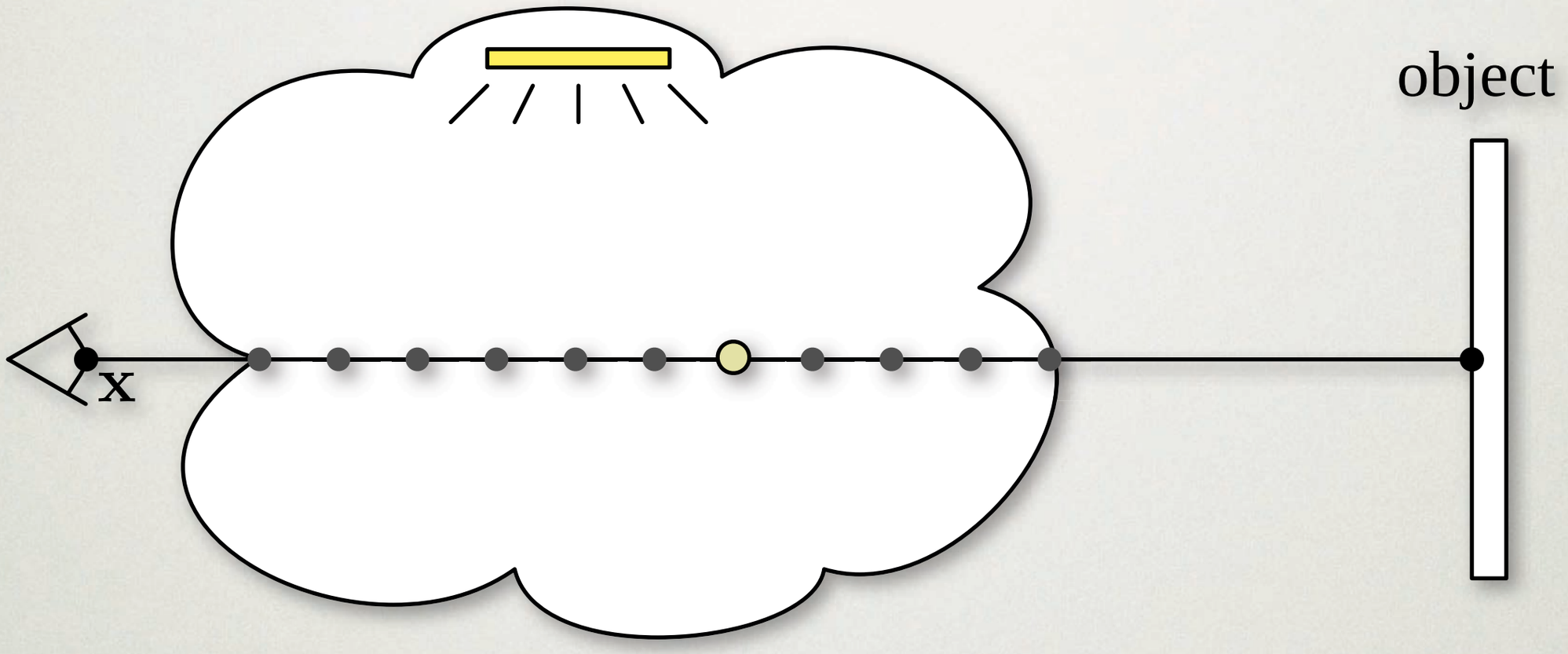
# Ray Marching




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

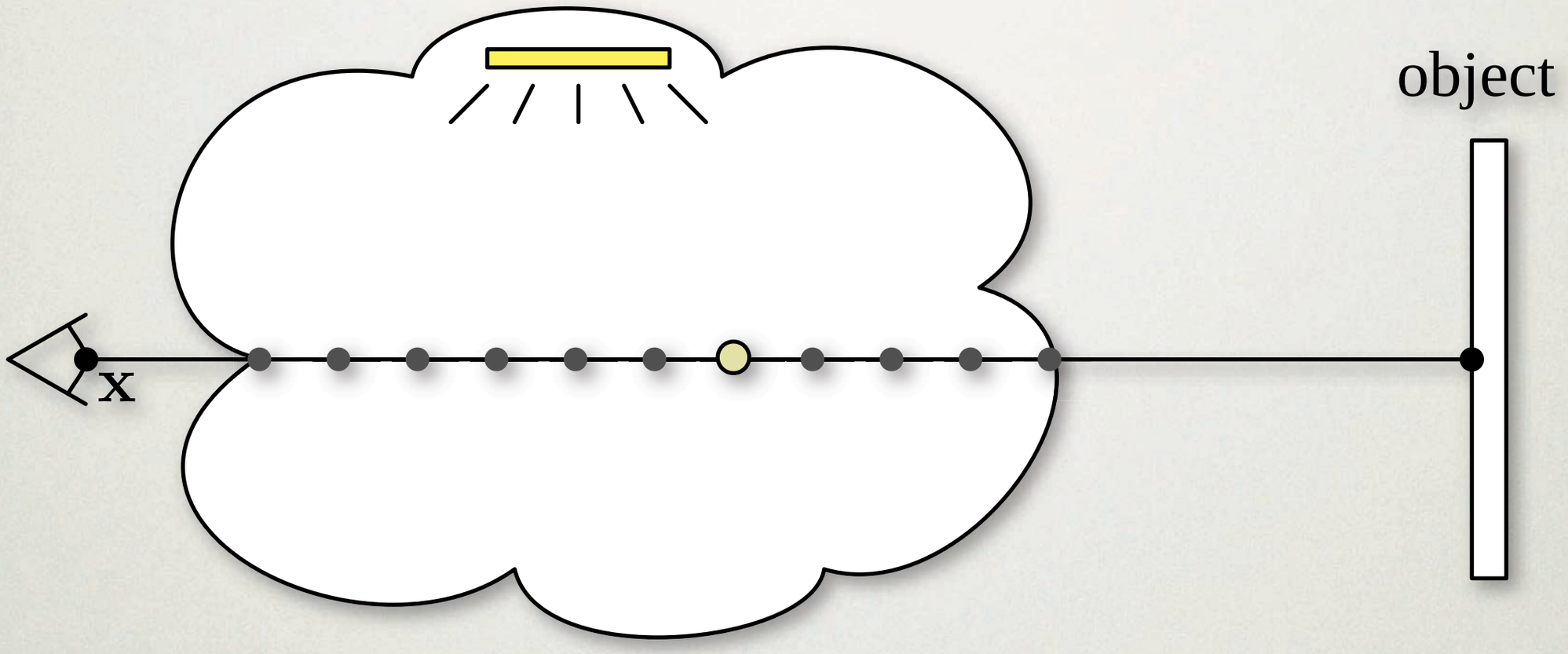
- compute $T_r$ incrementally

# Ray Marching




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

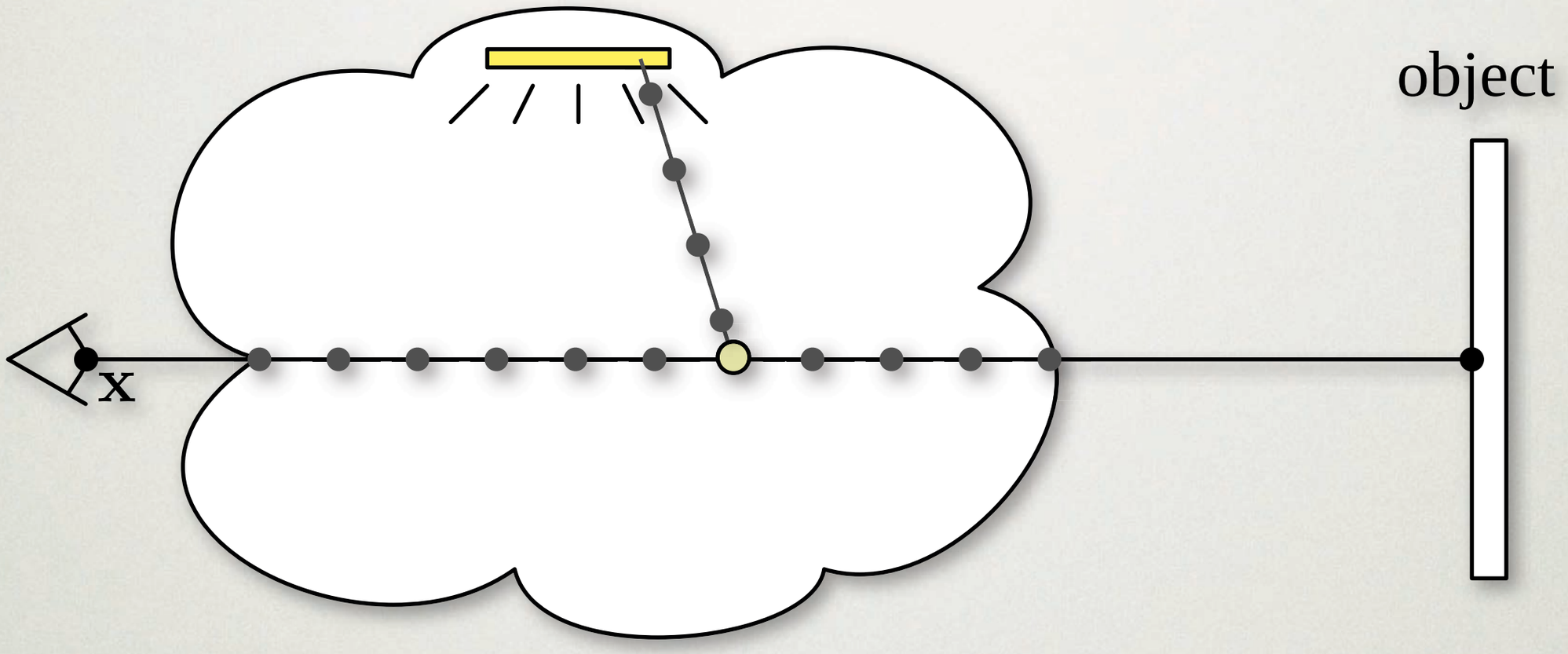
# Ray Marching




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

$$L_i(\mathbf{x}_t,\vec{\omega}) = \int_{\Omega_{4\pi}} p(\mathbf{x}_t,\vec{\omega}_t,\vec{\omega})L(\mathbf{x}_t,\vec{\omega}_t)\ d\omega_t$$

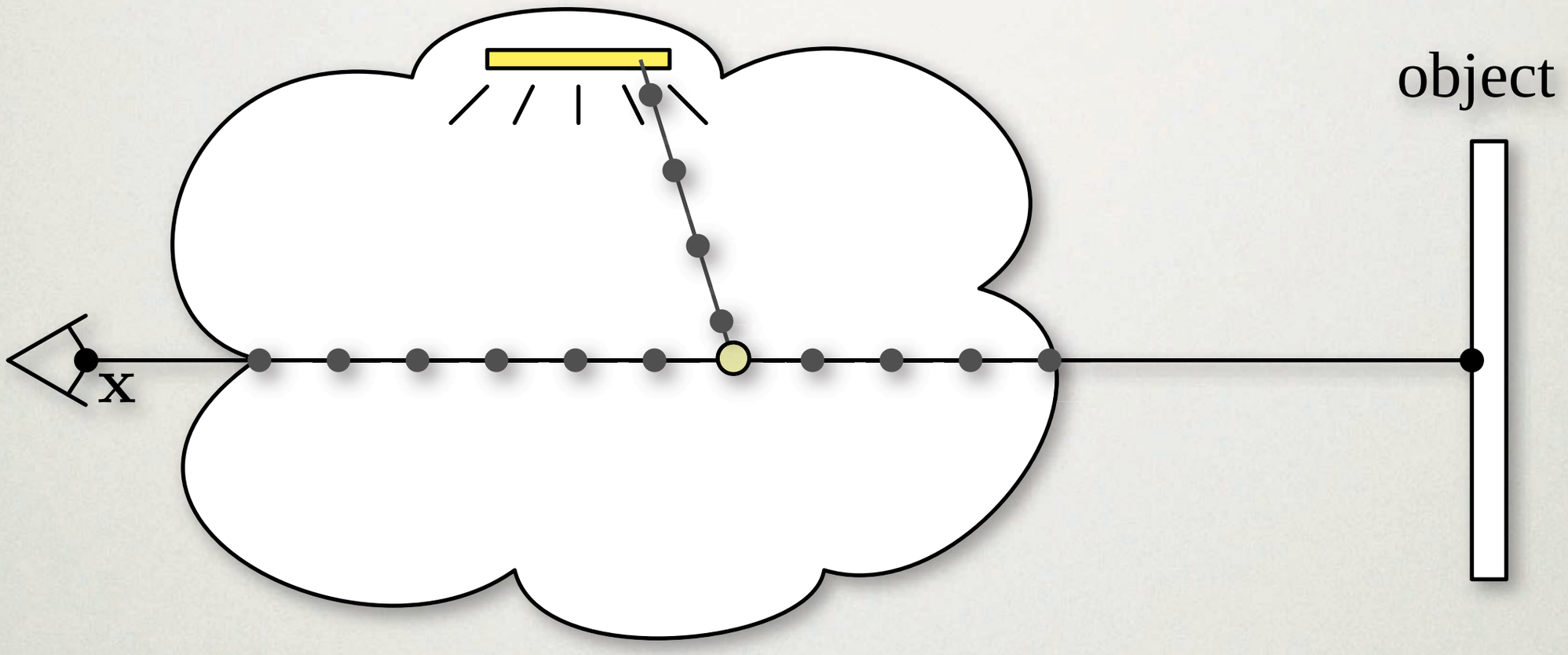
# Ray Marching (Single Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- assume only single scattering (direct lighting)

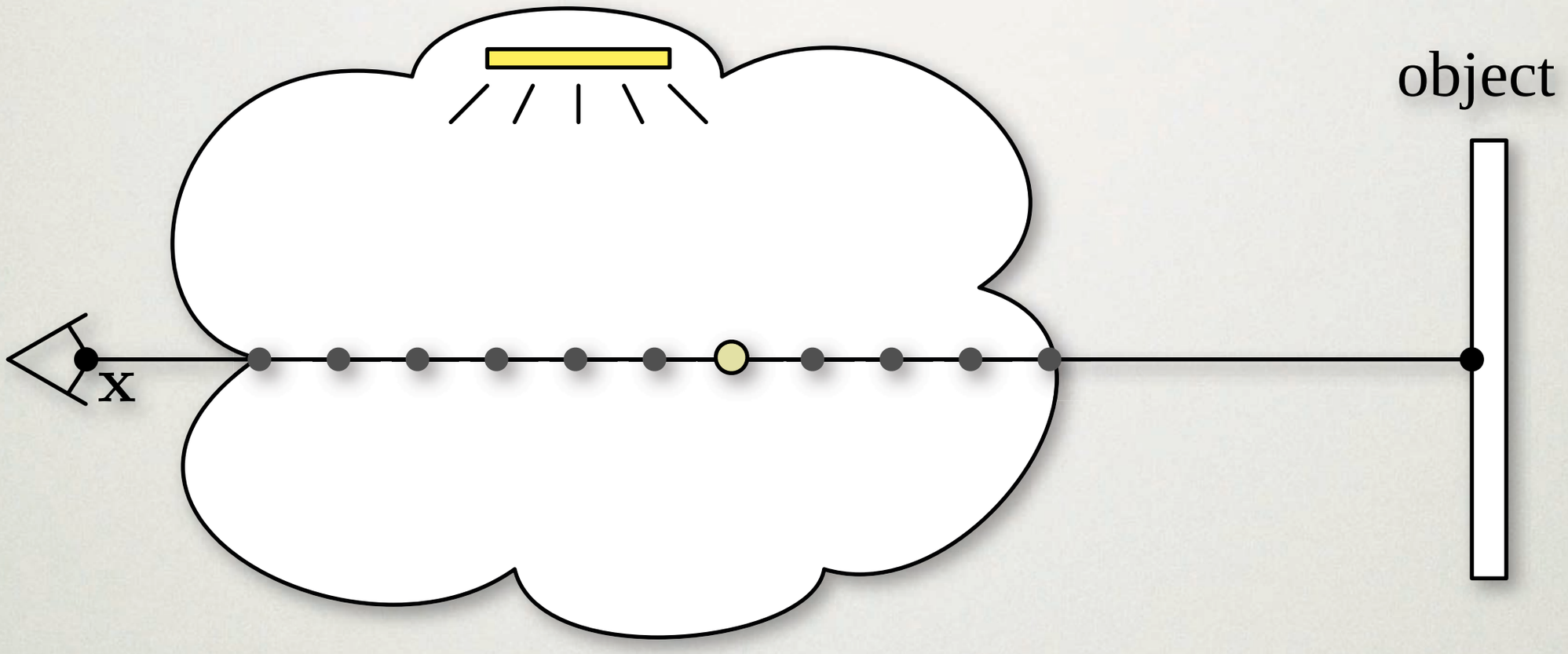
# Ray Marching (Single Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- assume only single scattering (direct lighting)
- trace shadow ray for volumetric shadows

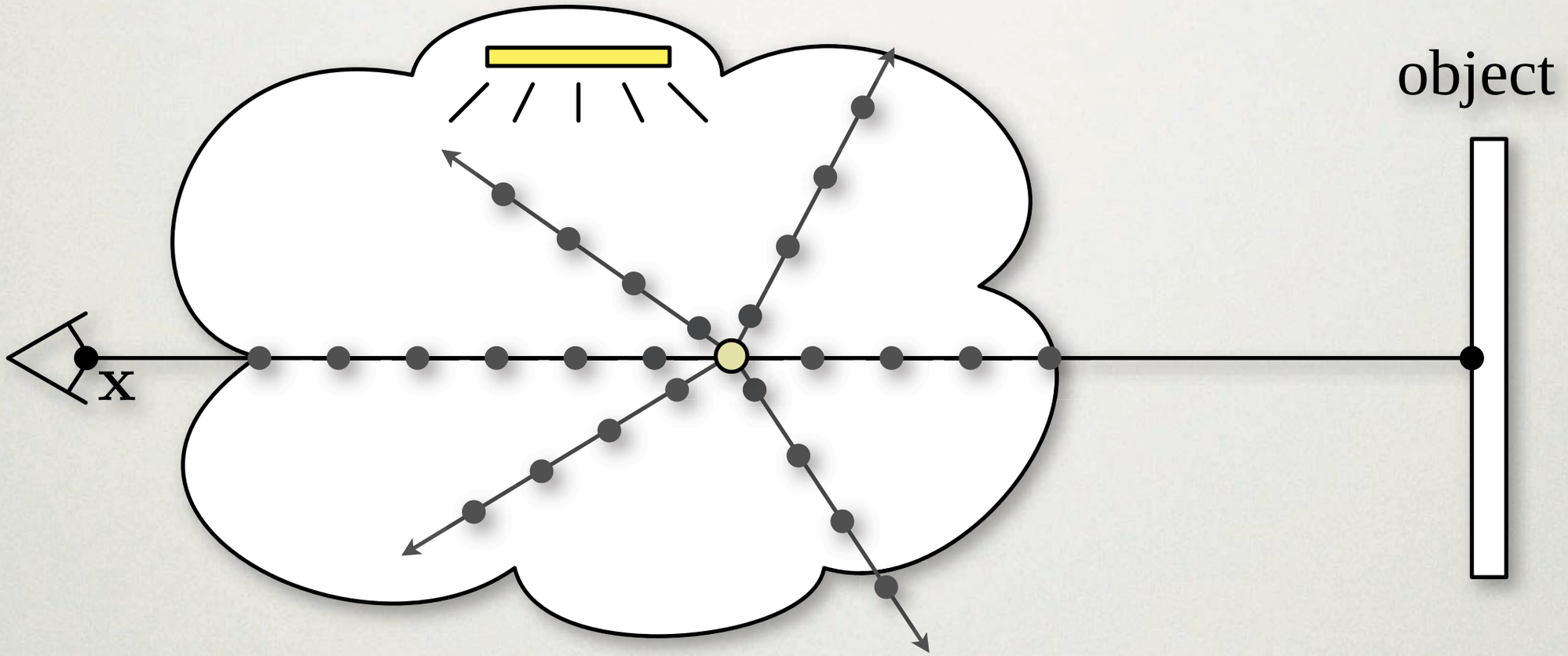
# Ray Marching (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- recursive ray marching
- exponential growth! expensive!

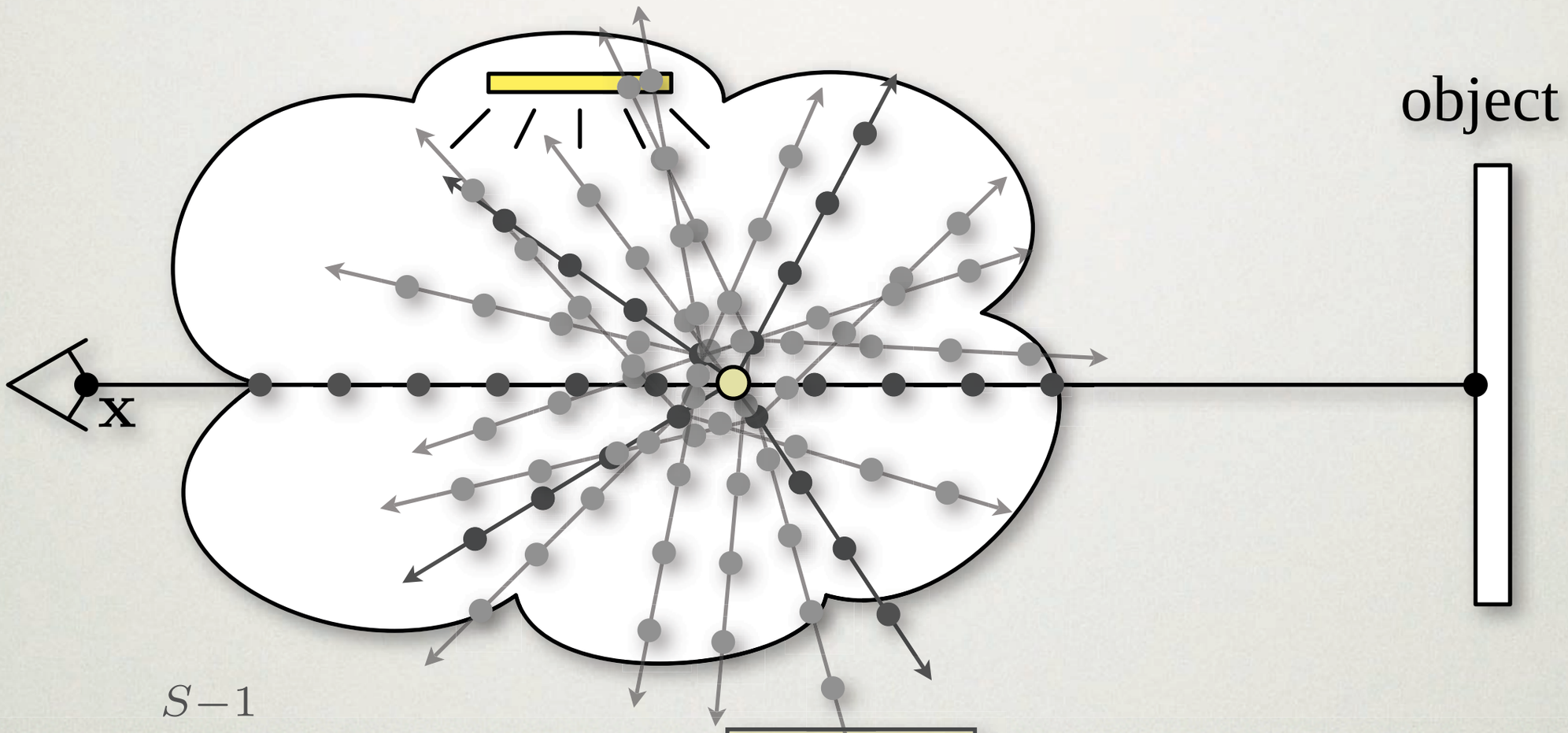
# Ray Marching (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- recursive ray marching
- exponential growth! expensive!

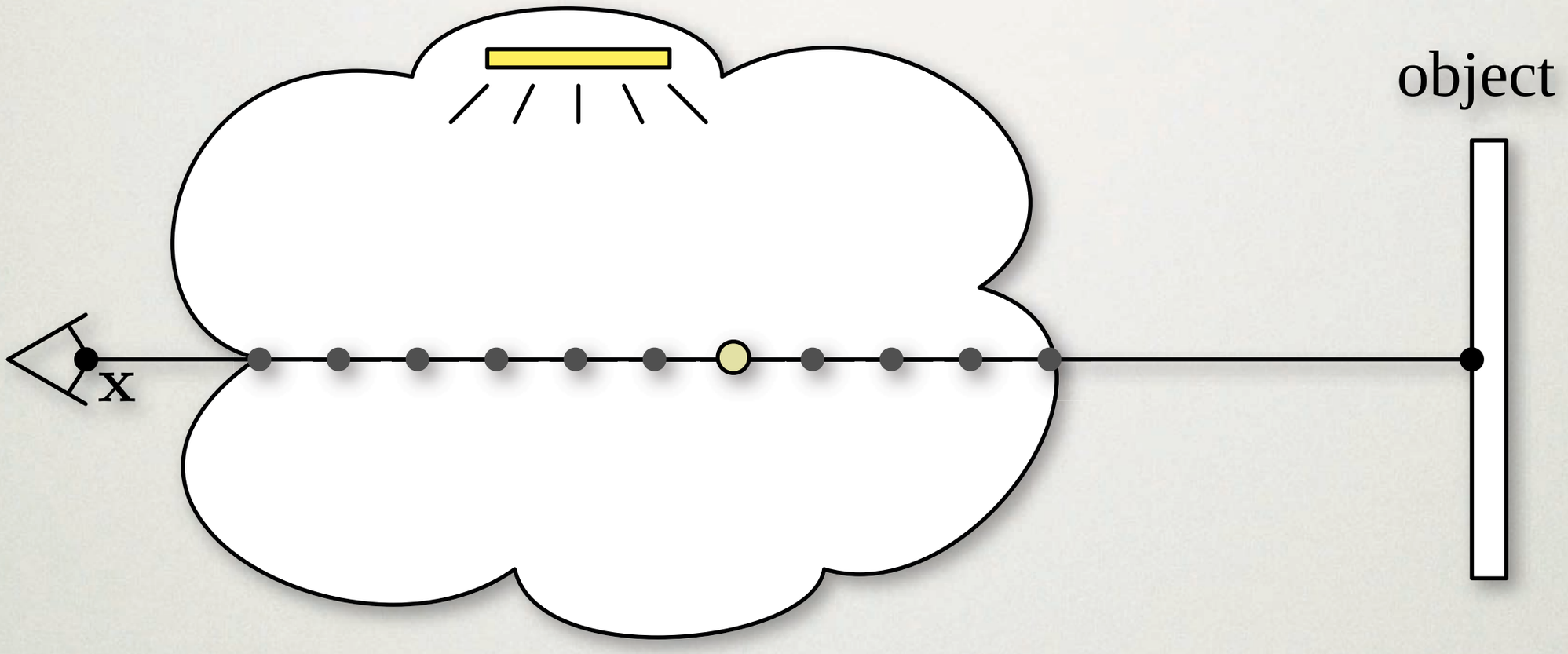
# Ray Marching (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- recursive ray marching
- exponential growth! expensive!

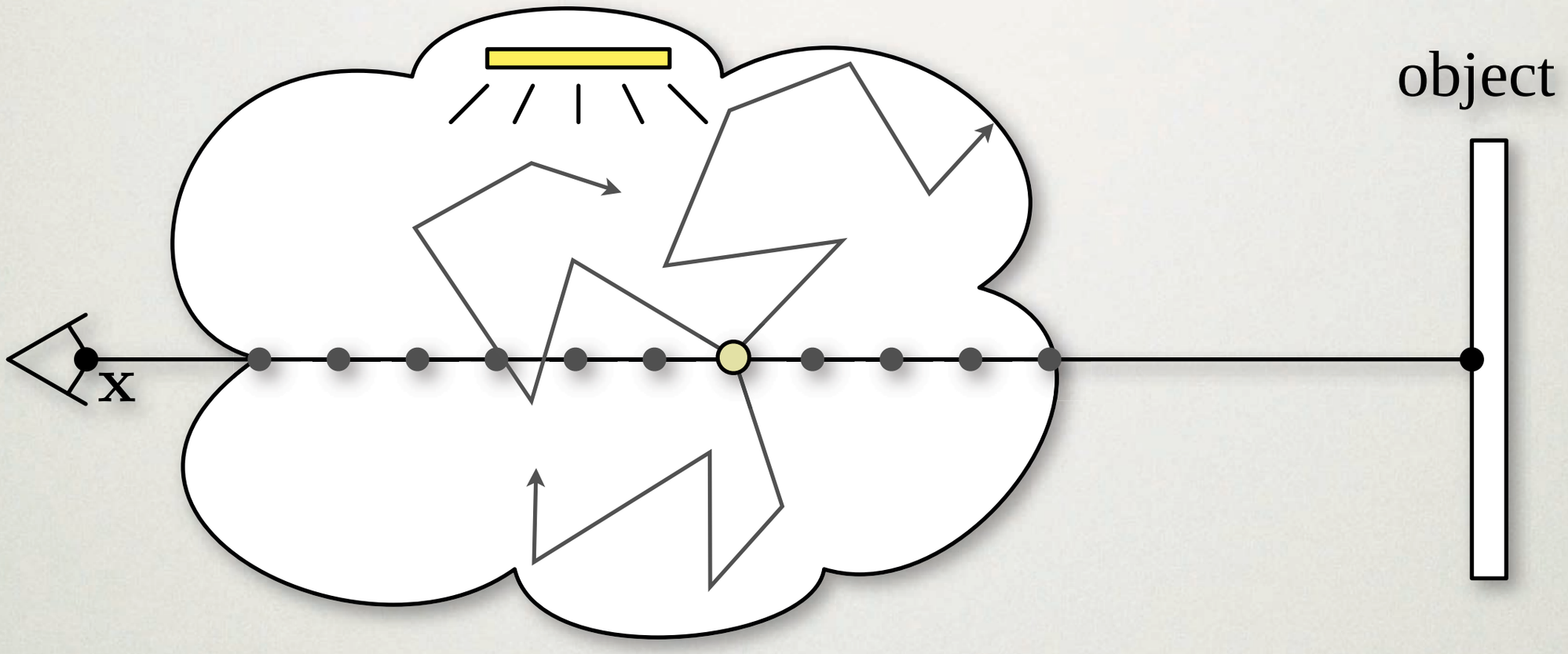
# Ray Marching (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- random walk sampling (path tracing)
- linear growth, but still expensive.

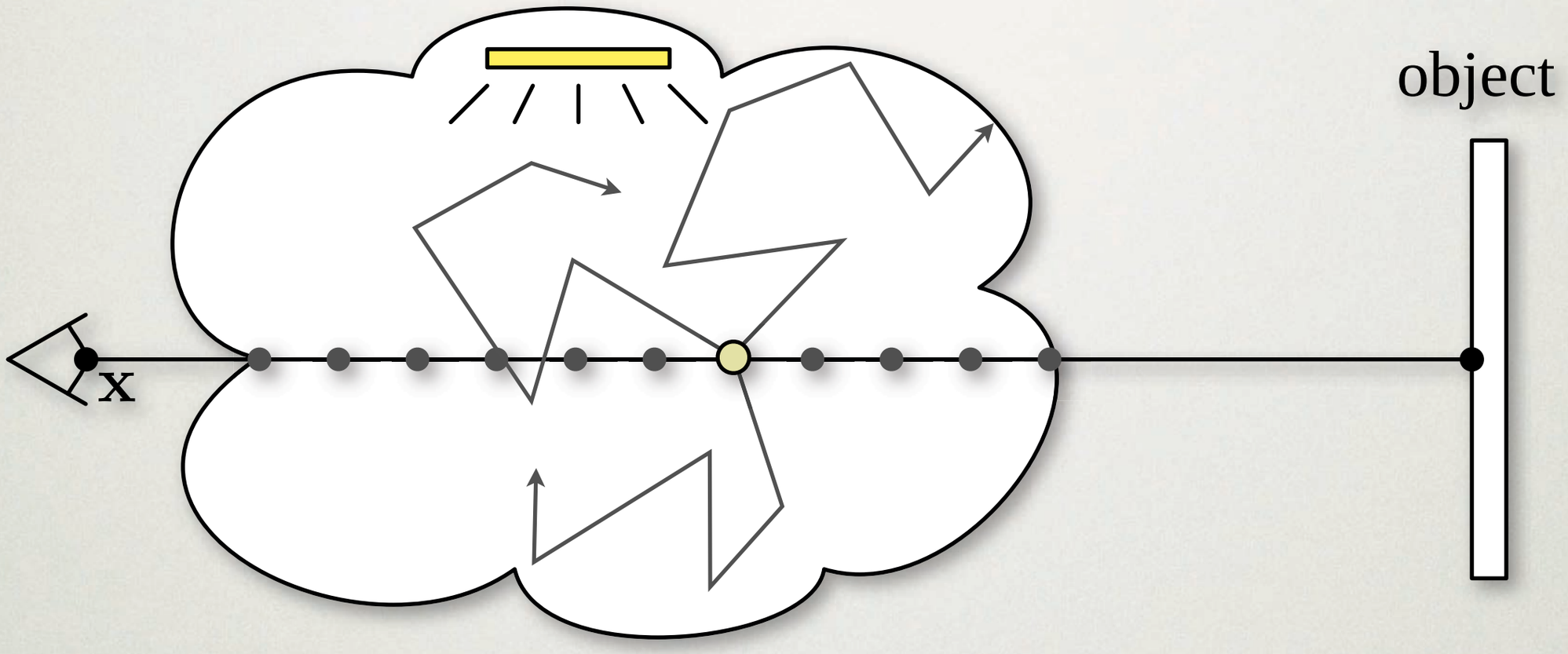
# Ray Marching (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- random walk sampling (path tracing)
- linear growth, but still expensive.

# Ray Marching (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

- random walk sampling (path tracing)
- linear growth, but still expensive.

# So Far

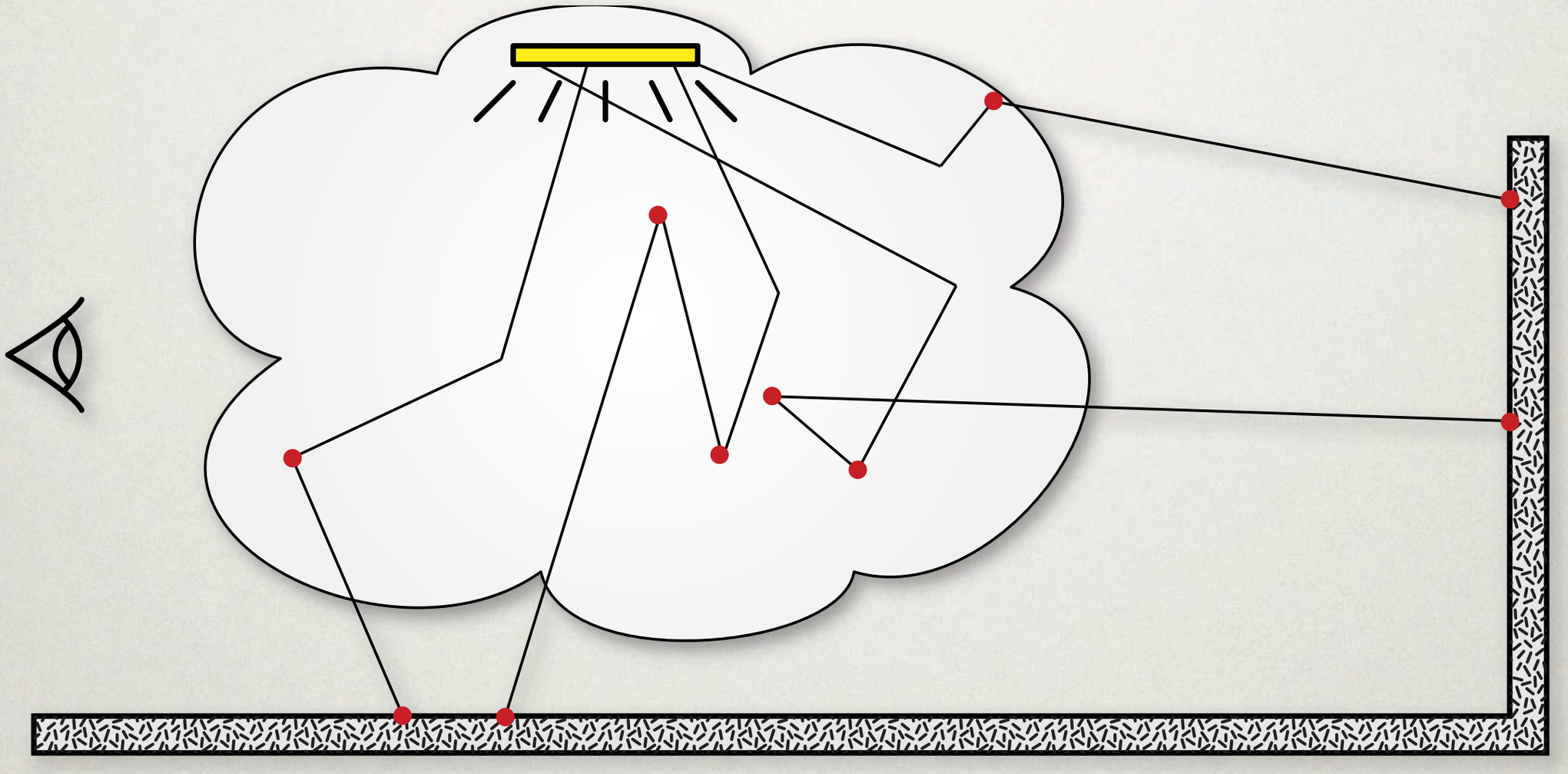
- Single scattering relatively in-expensive
- Multiple scattering **very** expensive

# Volumetric Photon Tracing

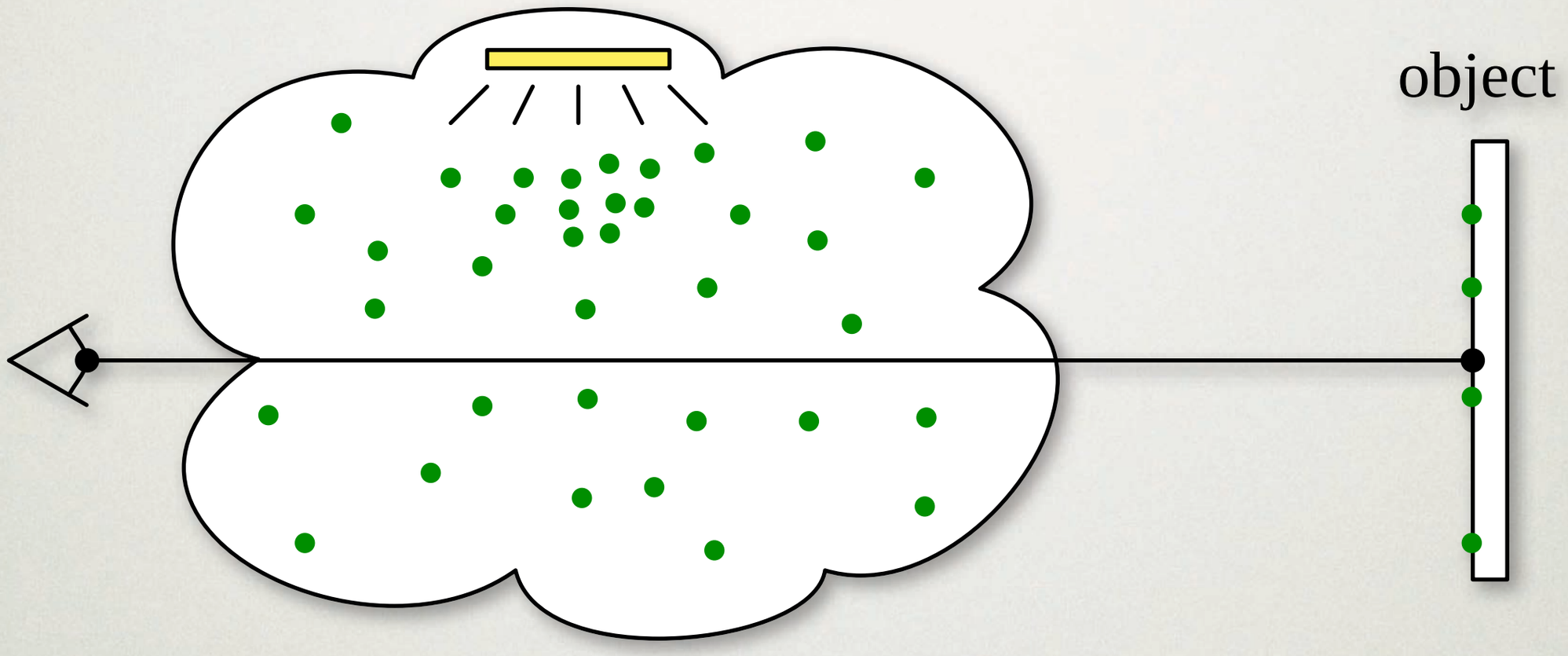
Two-pass algorithm:

1) Photon tracing
   - Simulate the scattering of photons
2) Rendering
   - Reuse the photons to estimate multiple scattering
     - VPL methods or density estimation

# Volumetric Photon Tracing

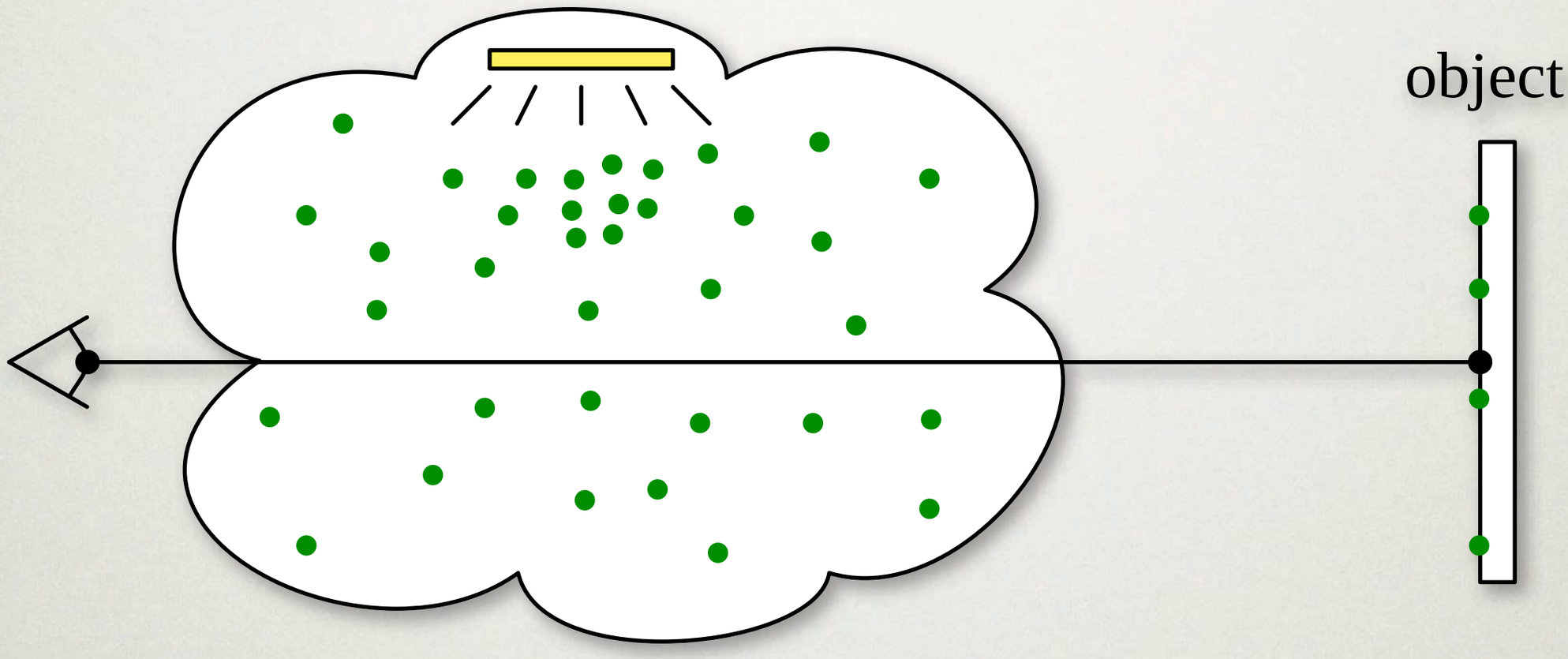


# Volume Photon Map



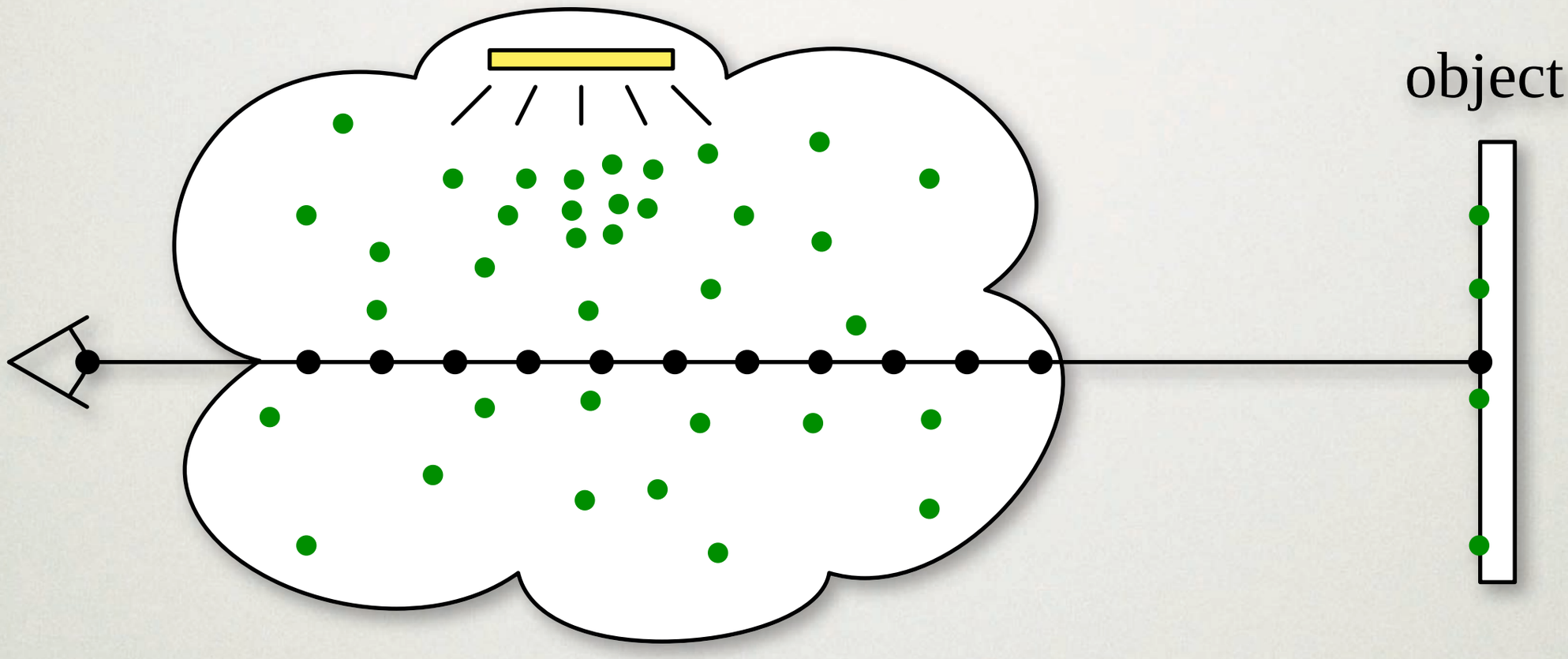

$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \; dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

# VPL (Virtual Point Light) Methods



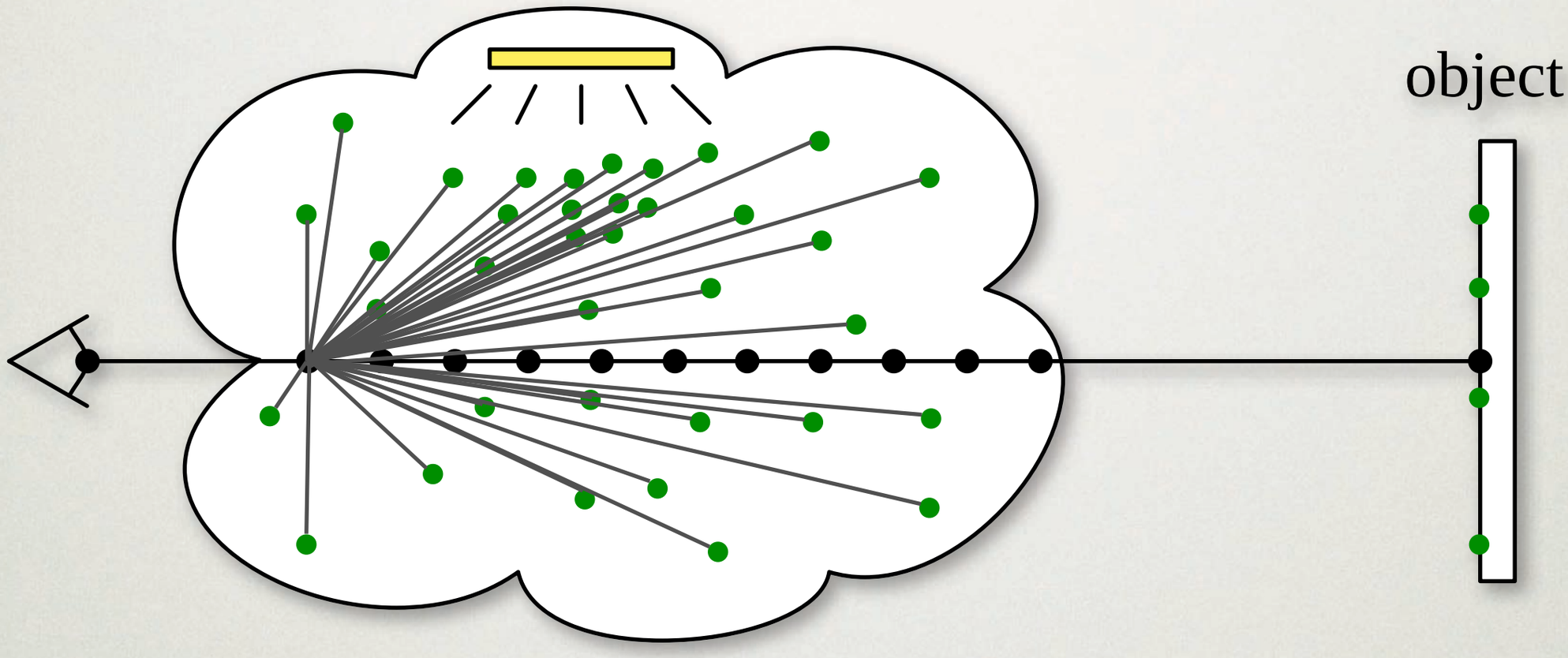

1. Treat each photon as a “virtual point light”

# VPL Methods (Instant Radiosity)



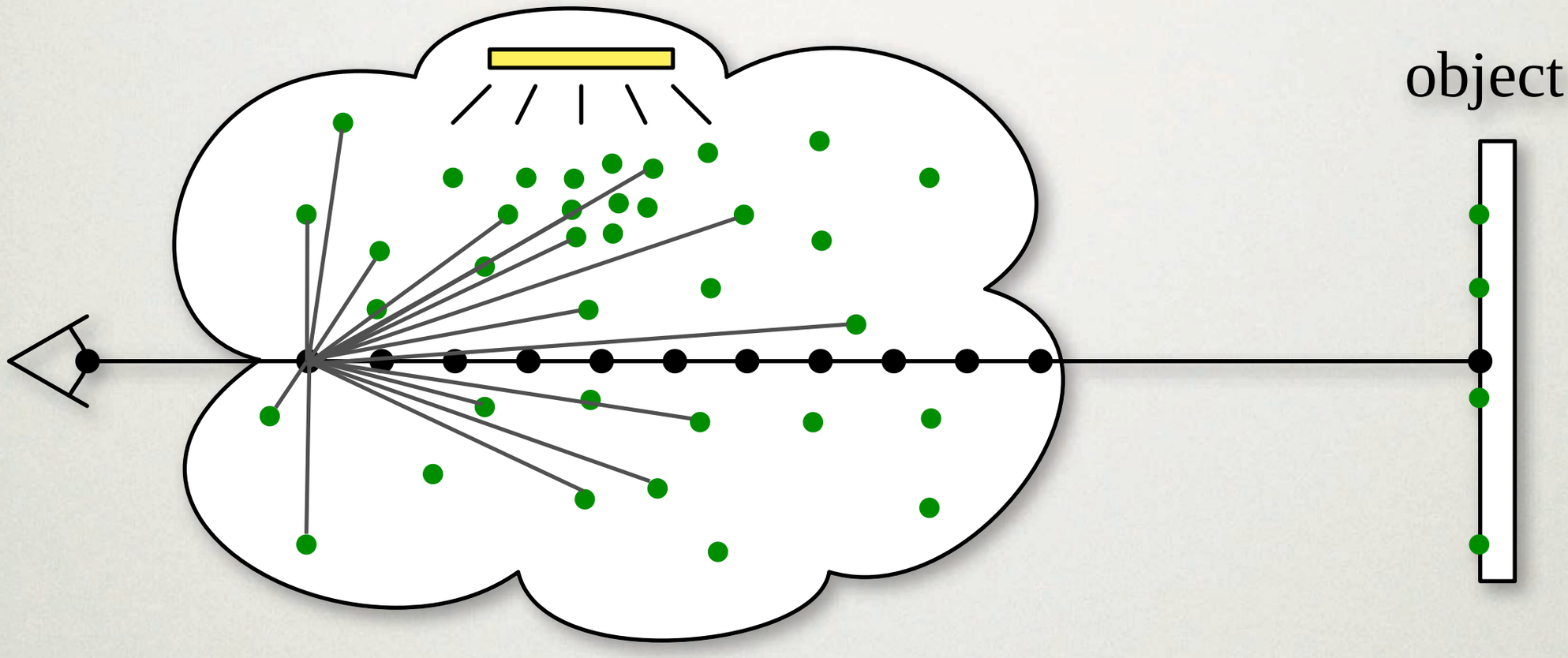

1. Treat each photon as a “virtual point light”
2. Perform ray marching
   - At each step: shoot shadow rays to VPLs

# VPL Methods (Instant Radiosity)



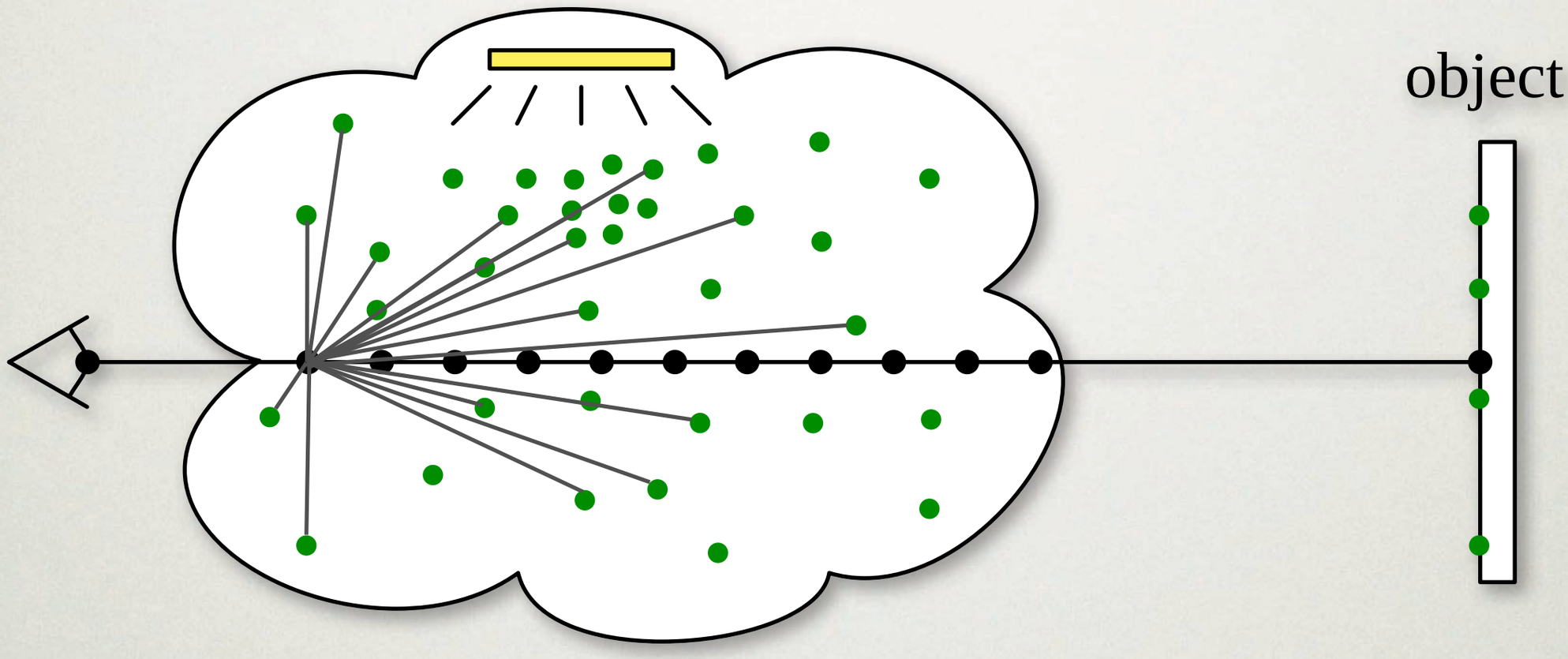

1. Treat each photon as a “virtual point light”
2. Perform ray marching
   - At each step: shoot shadow rays to VPLs

# VPL Methods (Instant Radiosity)



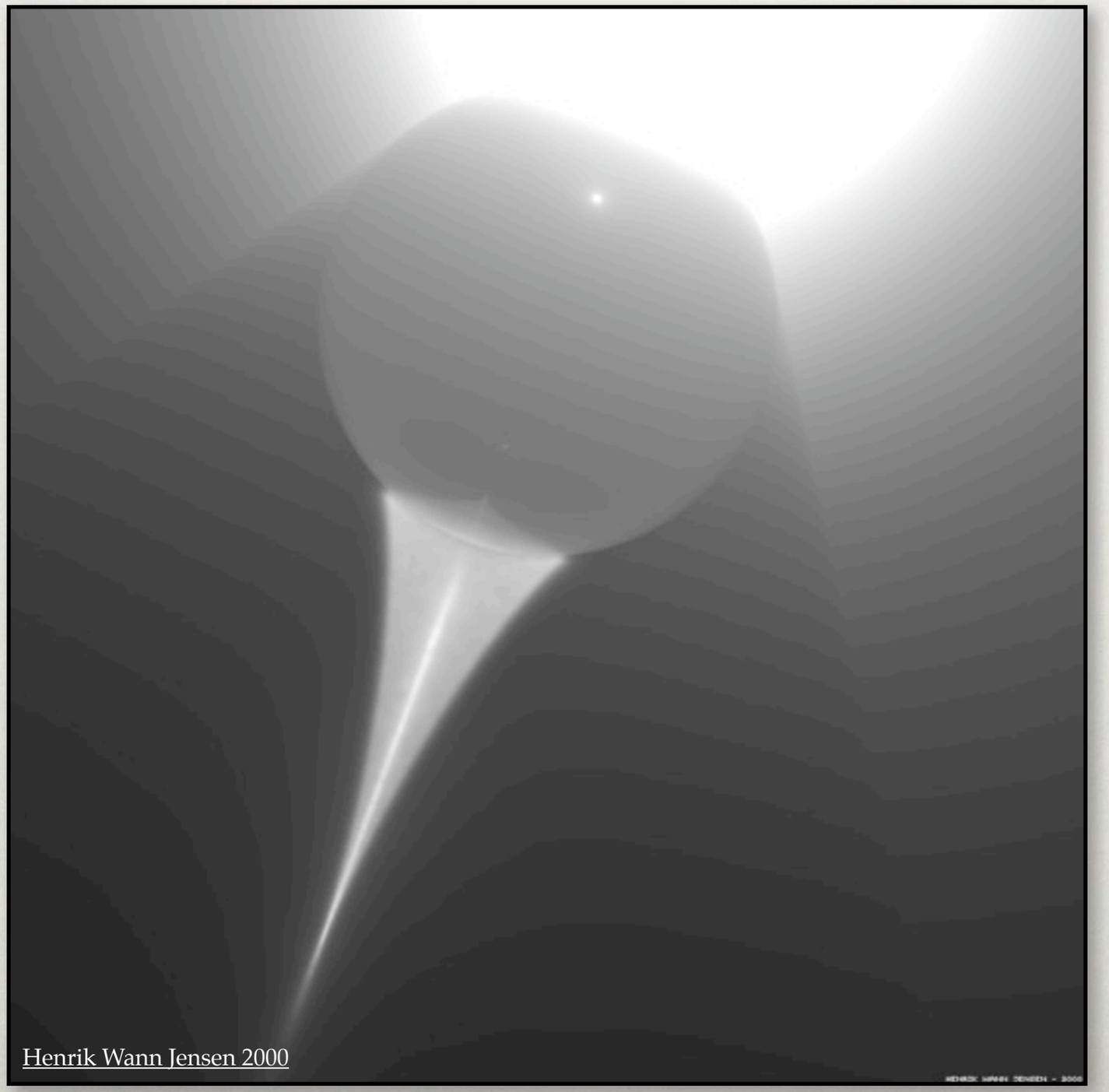

1. Treat each photon as a “virtual point light”
2. Perform ray marching
   - At each step: choose a **subset** of VPLs (faster performance, introduces noise)

# VPL Methods (Lightcuts)



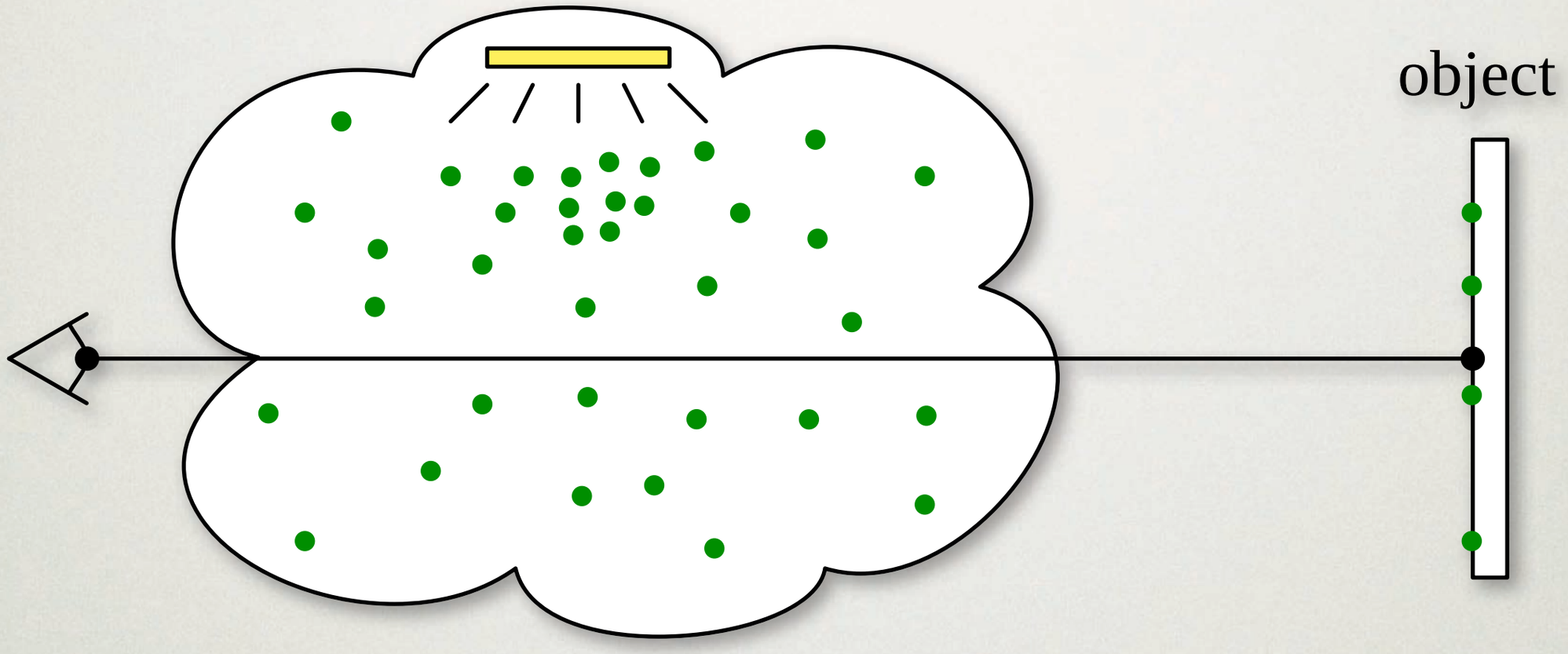

1. Create VPL **hierarchy**
2. Perform ray marching
   - At each step: choose **hierarchical subset** (faster performance, tries to limit noise)

# Caustics



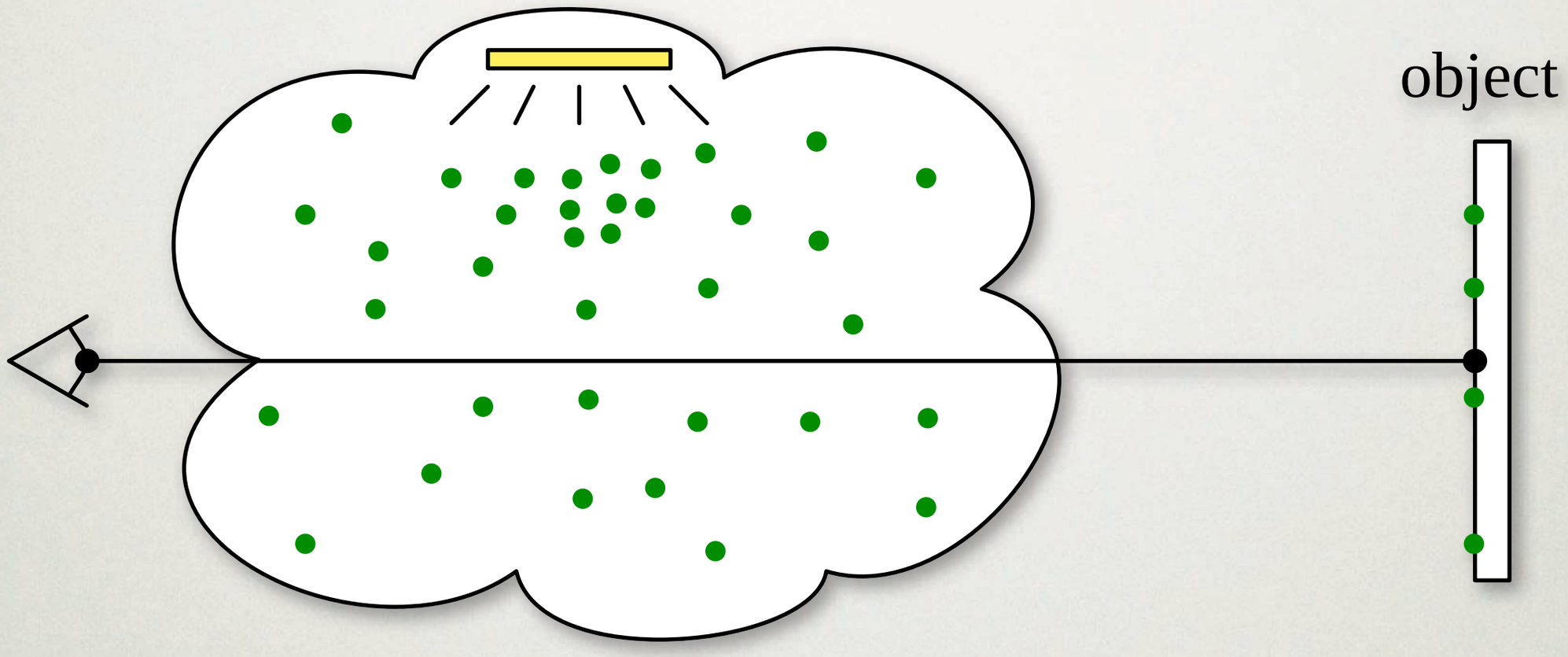
Henrik Wann Jensen 2000

# Volumetric Photon Mapping




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \; dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

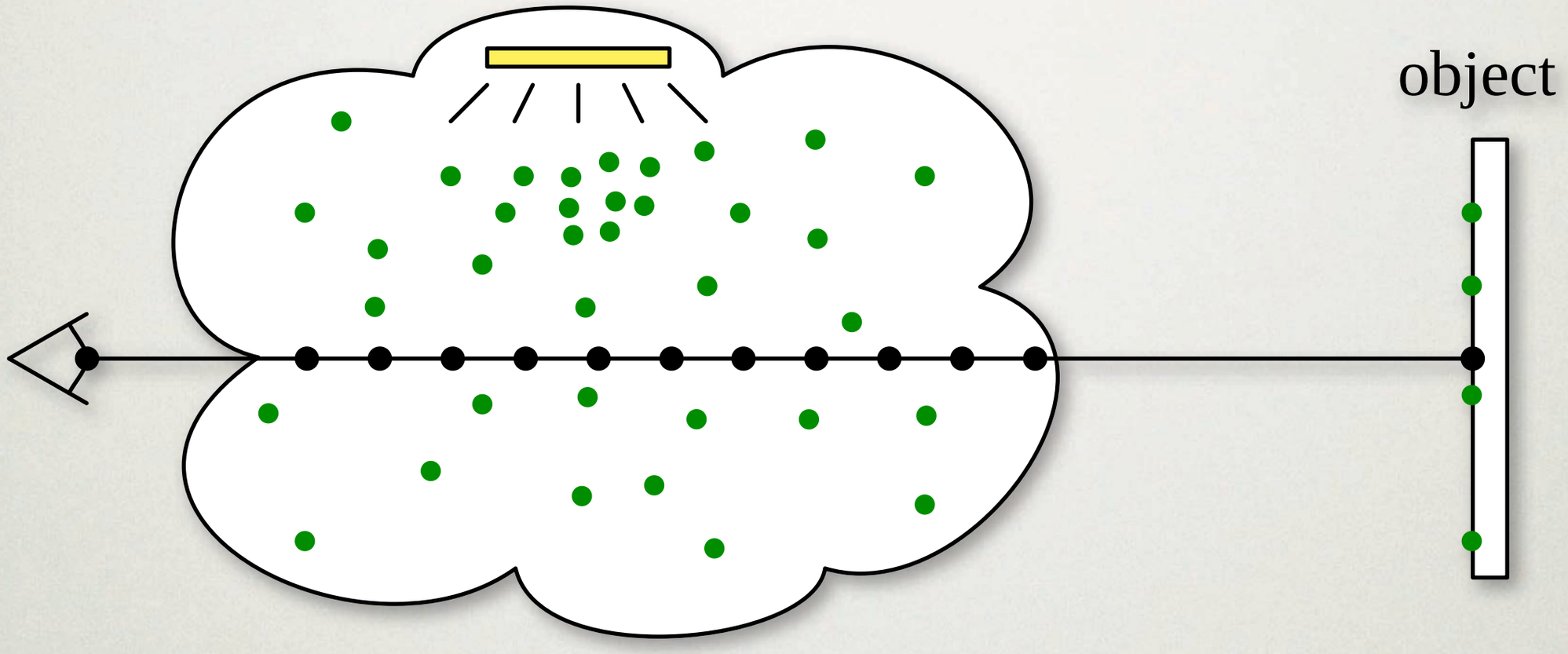
# Volumetric Photon Mapping




$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

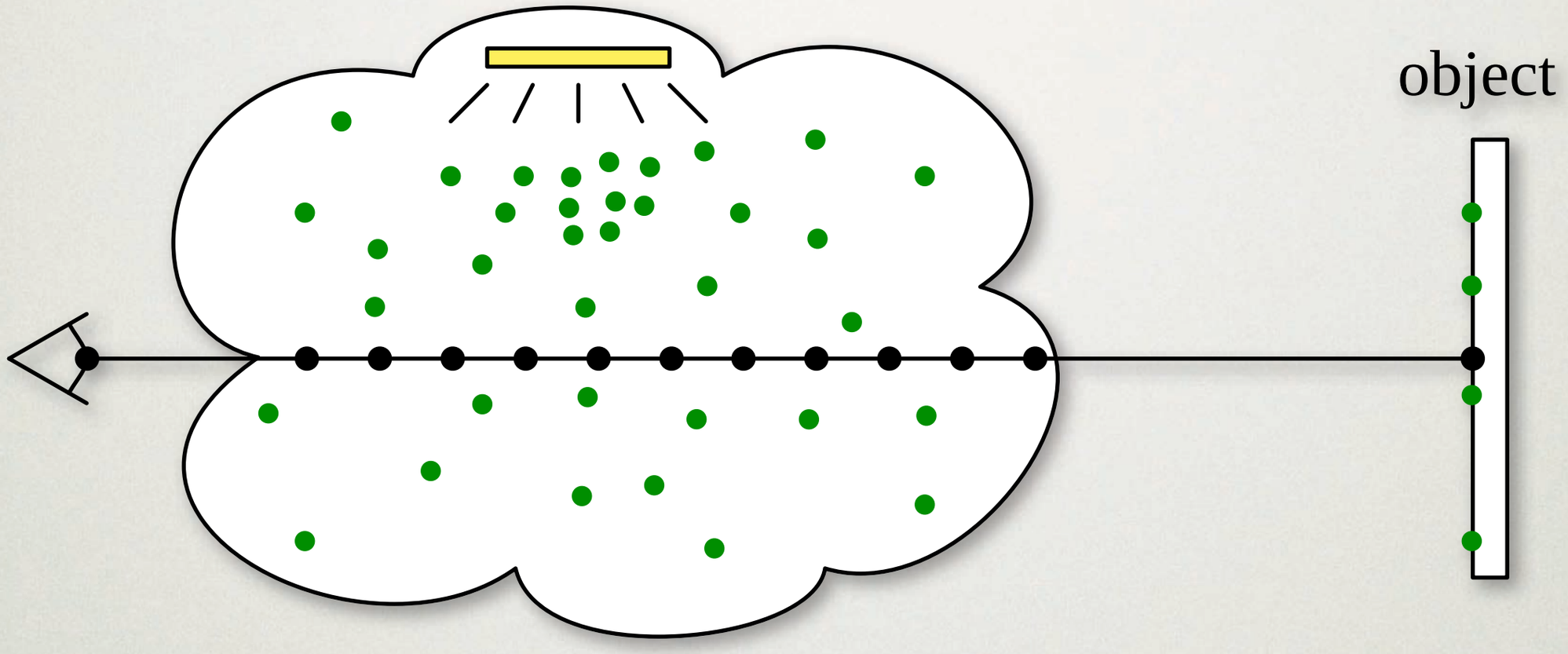
Two approaches: traditional & beam estimation

# Traditional Photon Mapping (Ray Marching)



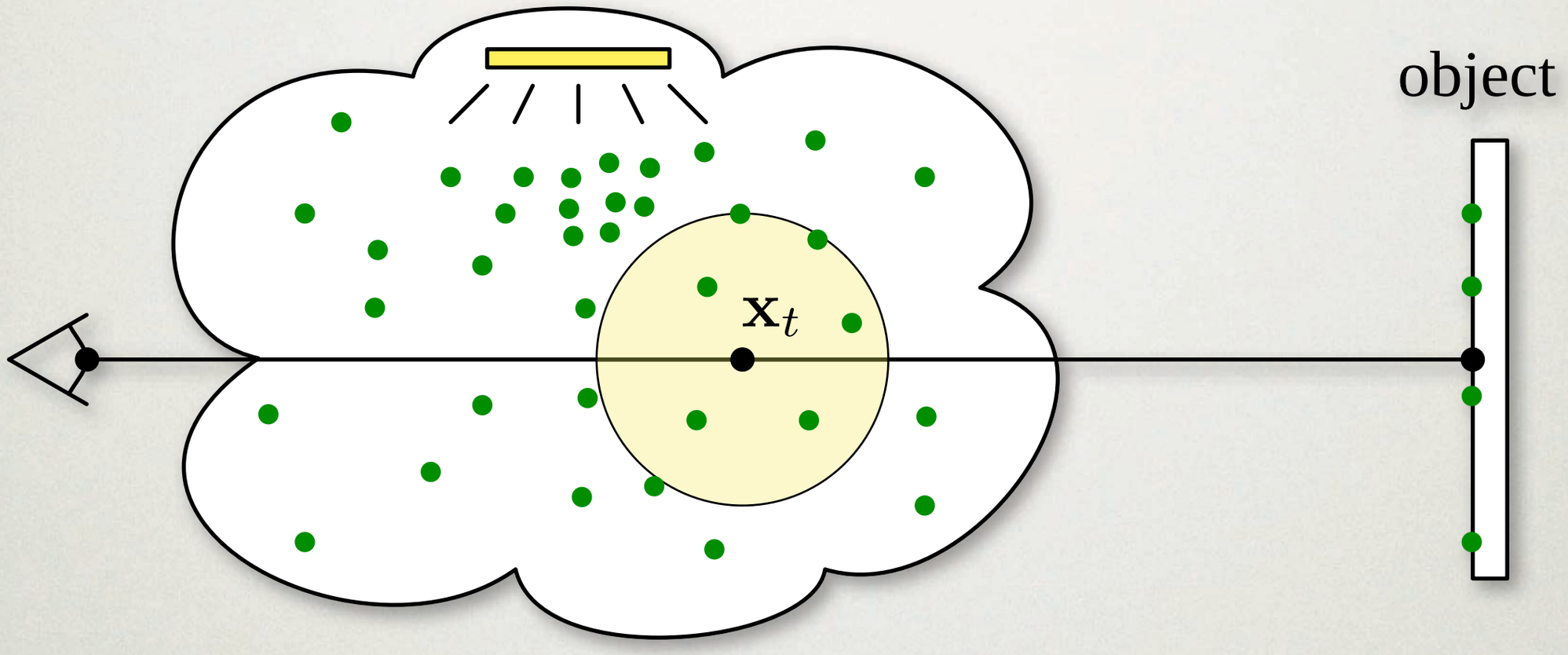

$$L(\mathbf{x}, \vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \Delta_t + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

# Traditional Photon Mapping (Ray Marching)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

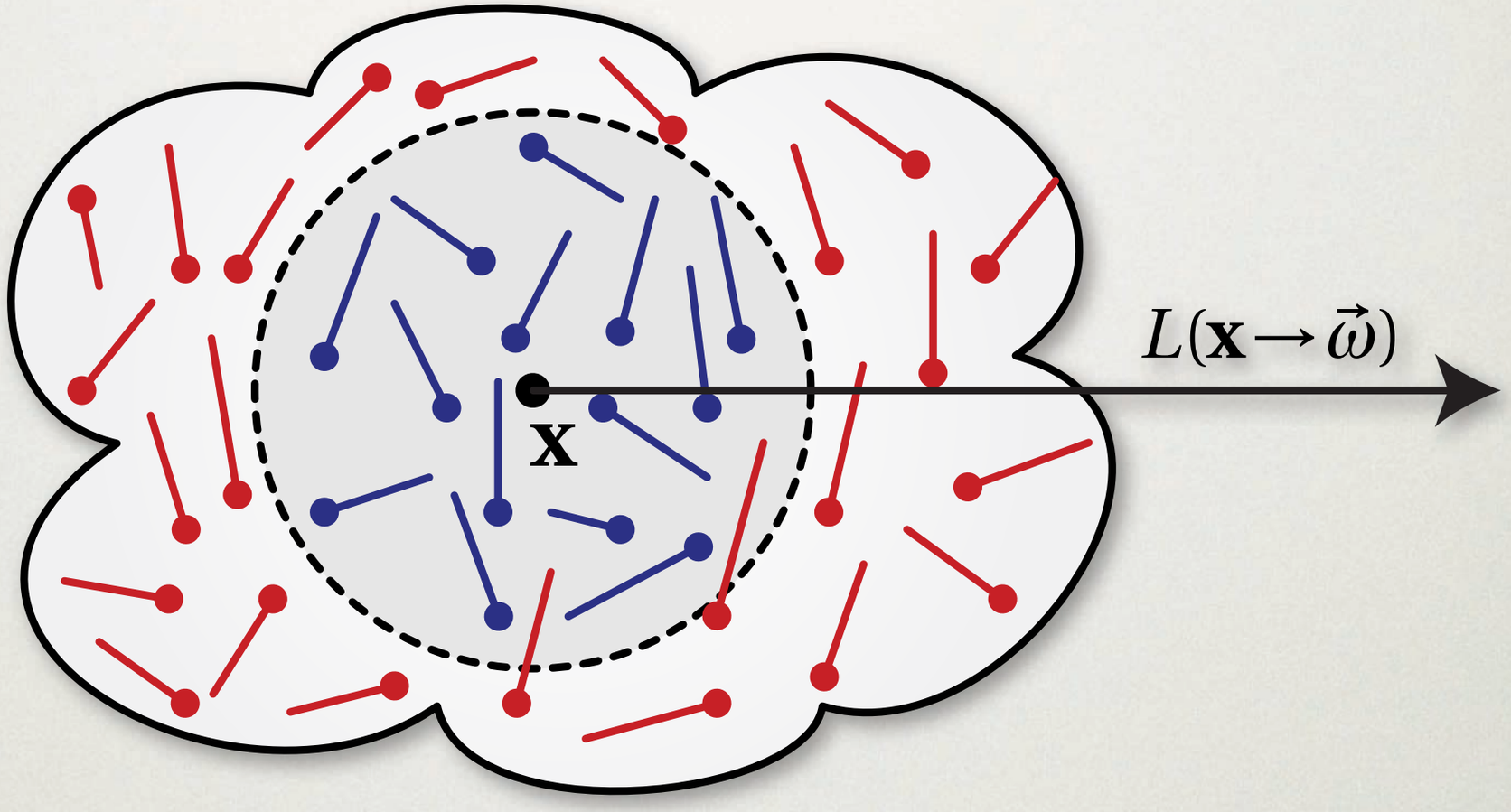
# Traditional Photon Mapping (Multiple Scattering)




$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

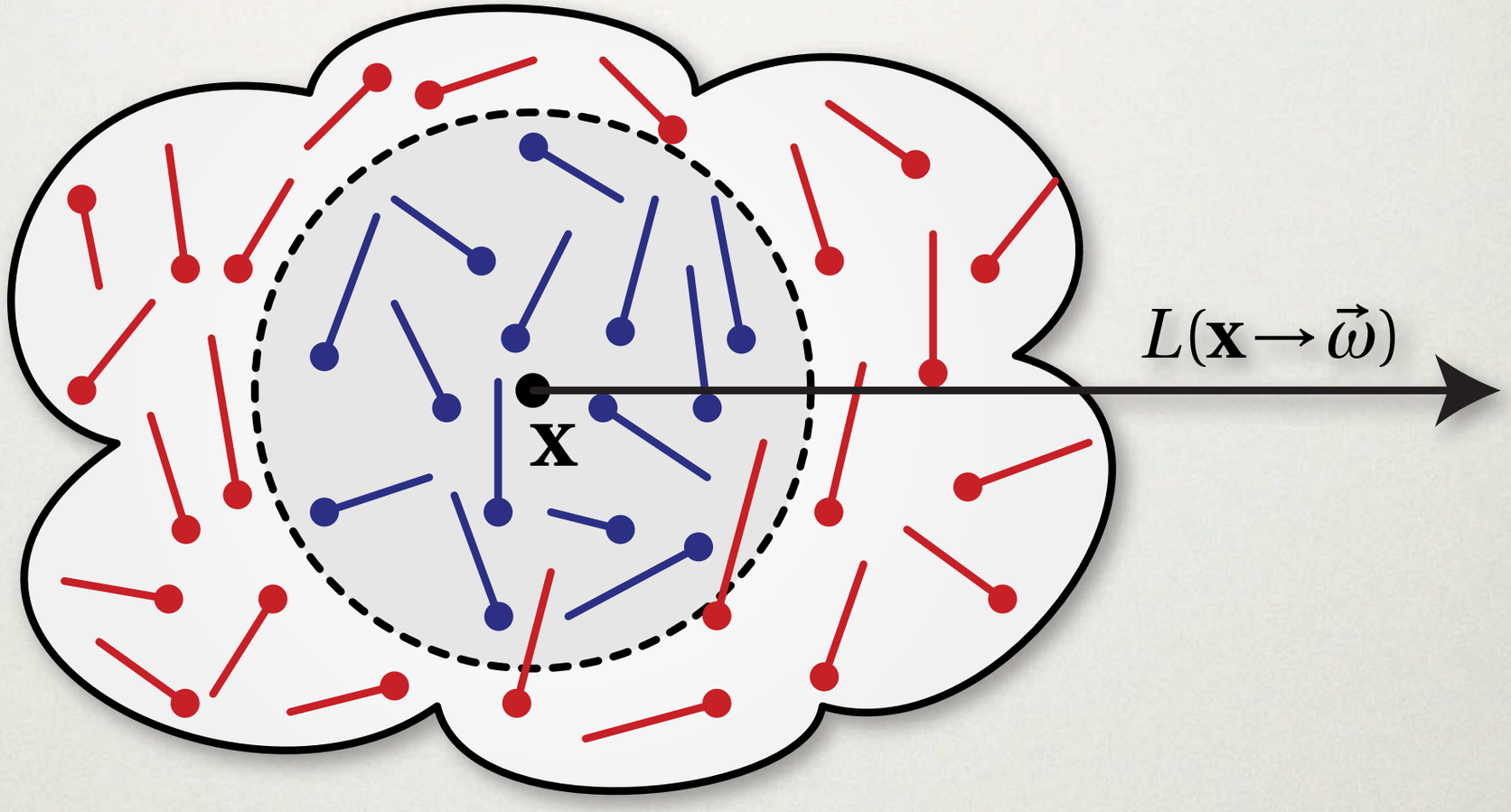
$L_i(\mathbf{x}_t,\vec{\omega})$ photon map for multiple scattering

# Volumetric Radiance Estimate



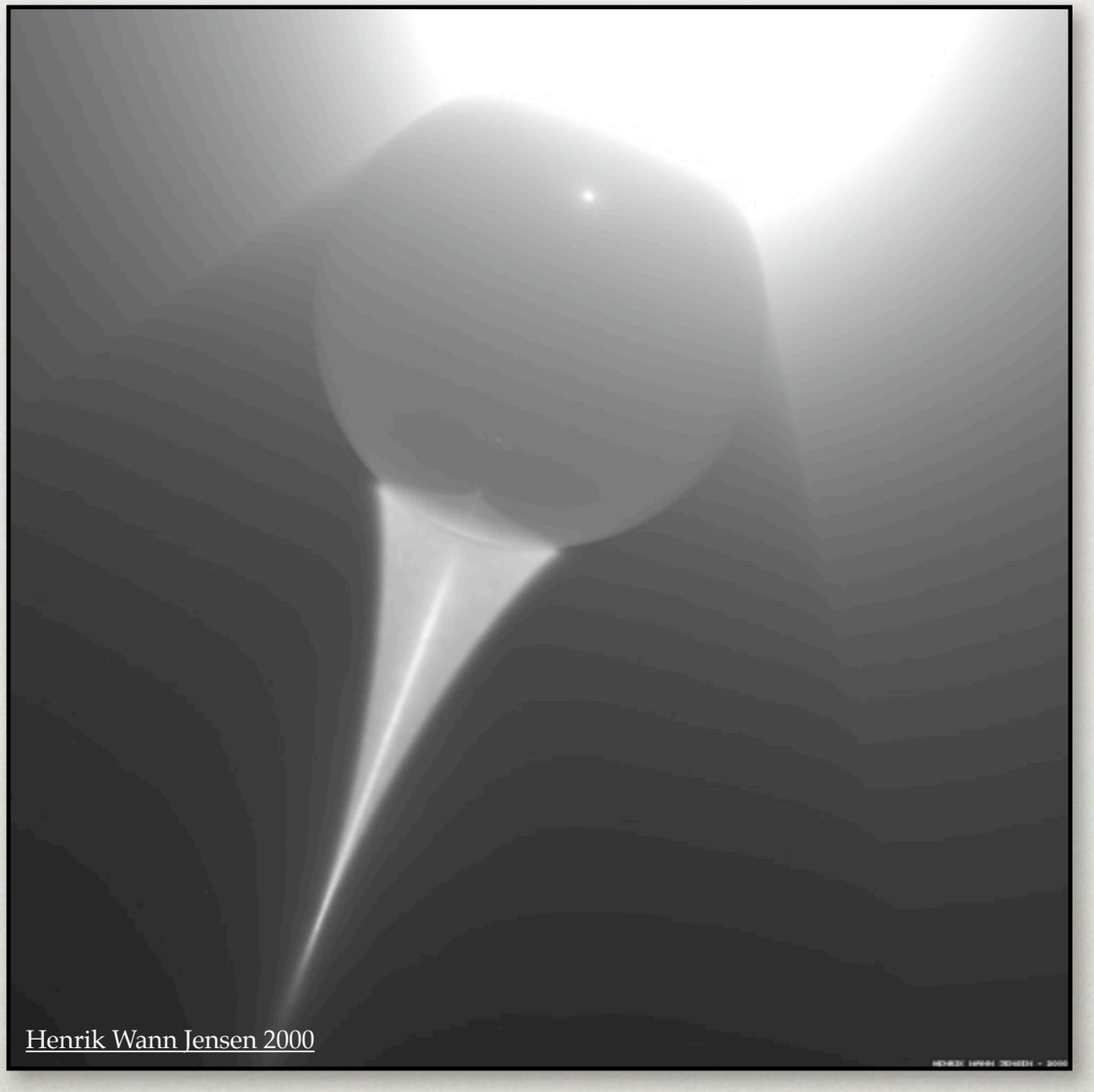

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \sum_{p=1}^{k} p(\mathbf{x}, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_p(\mathbf{x}, \vec{\omega}_p)}{\mathcal{V}(\mathbf{x})}$$

# Volumetric Radiance Estimate




$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \sum_{p=1}^{k} p(\mathbf{x}, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_p(\mathbf{x}, \vec{\omega}_p)}{\frac{4}{3}\pi r(\mathbf{x})^3}$$

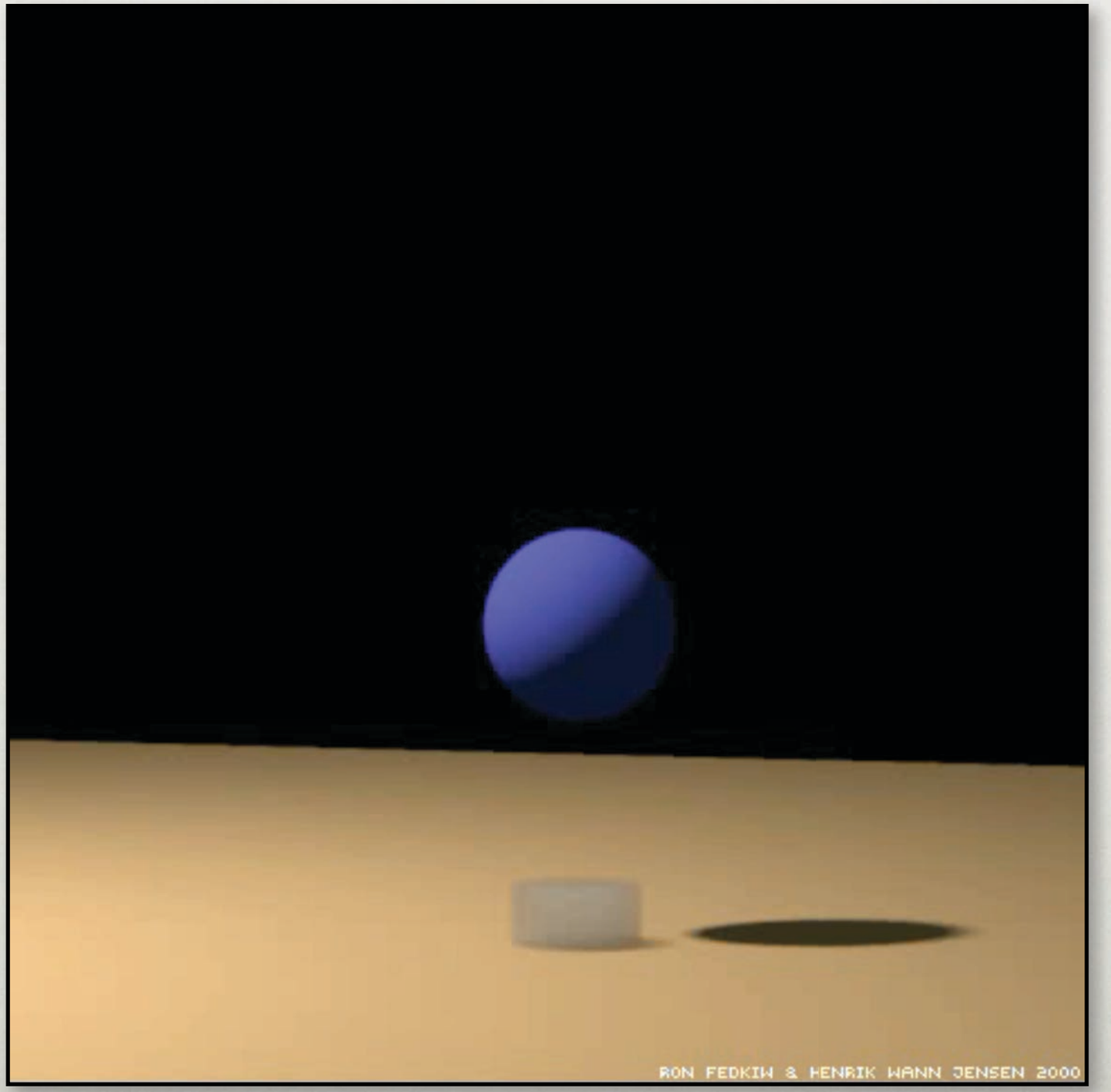
# A Volume Caustic




500,000 photons. 1 minute

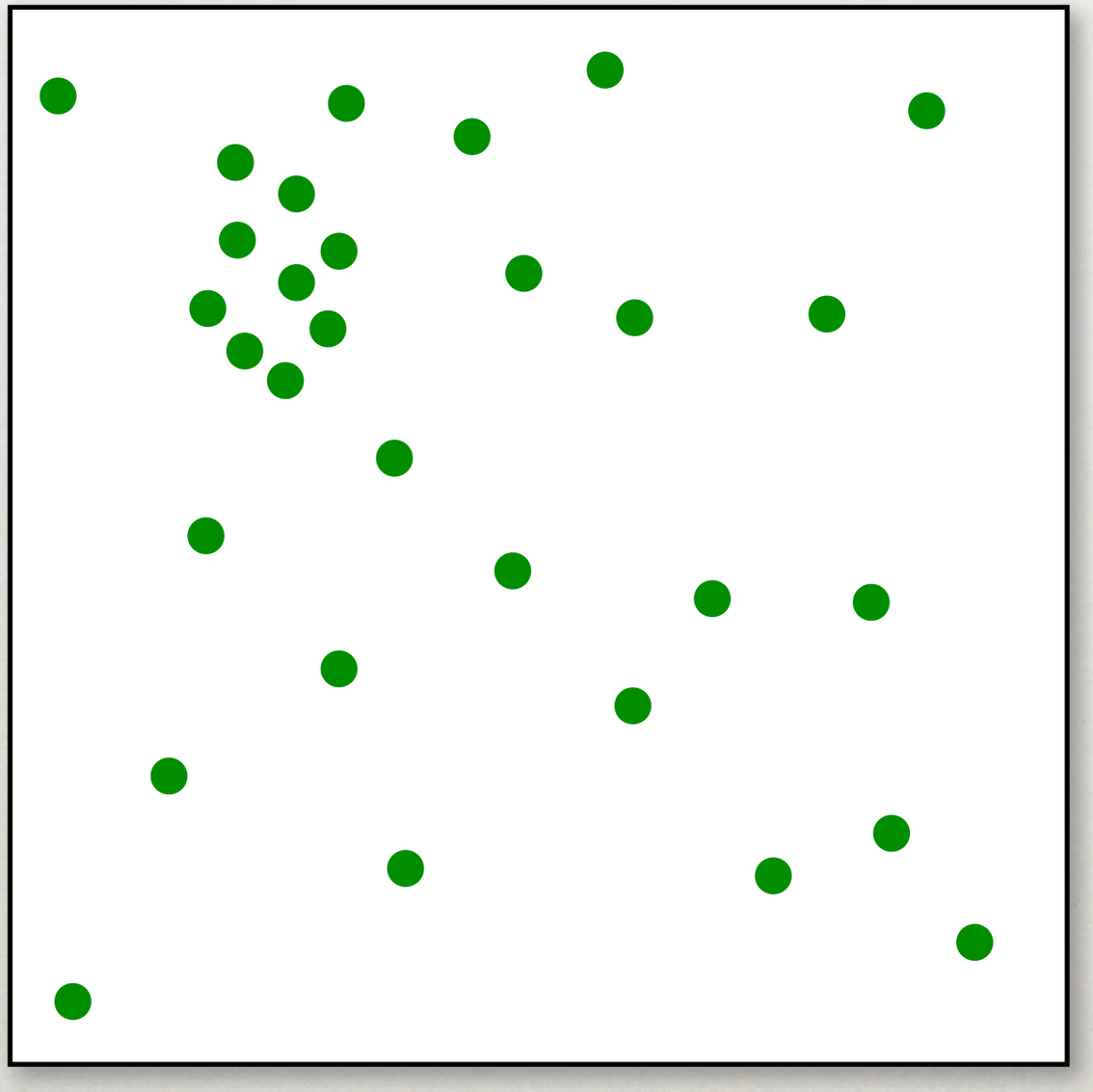
# Rising Smoke

# Smoke flowing past a sphere



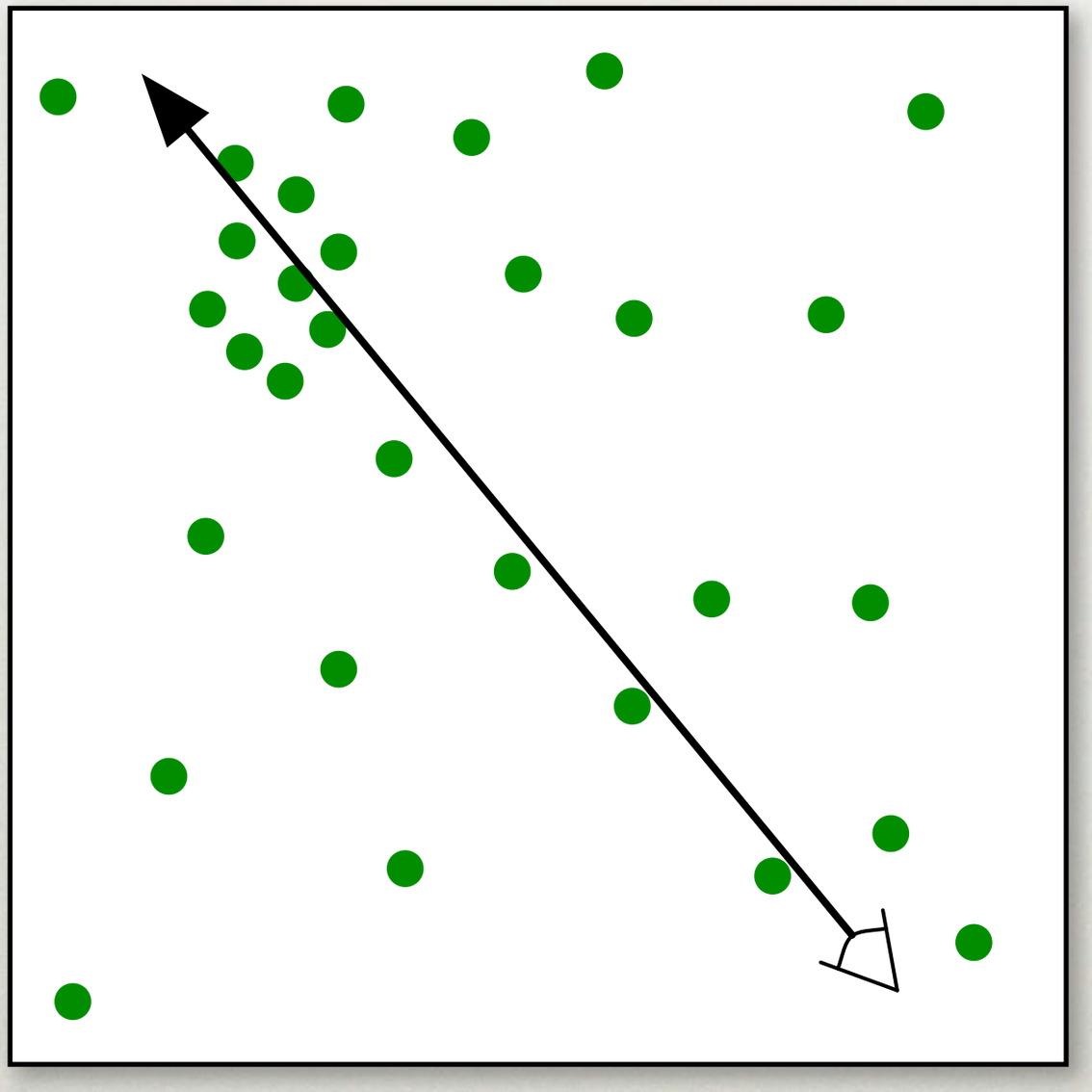
# Volumetric Photon Mapping

## Conventional Radiance Estimate



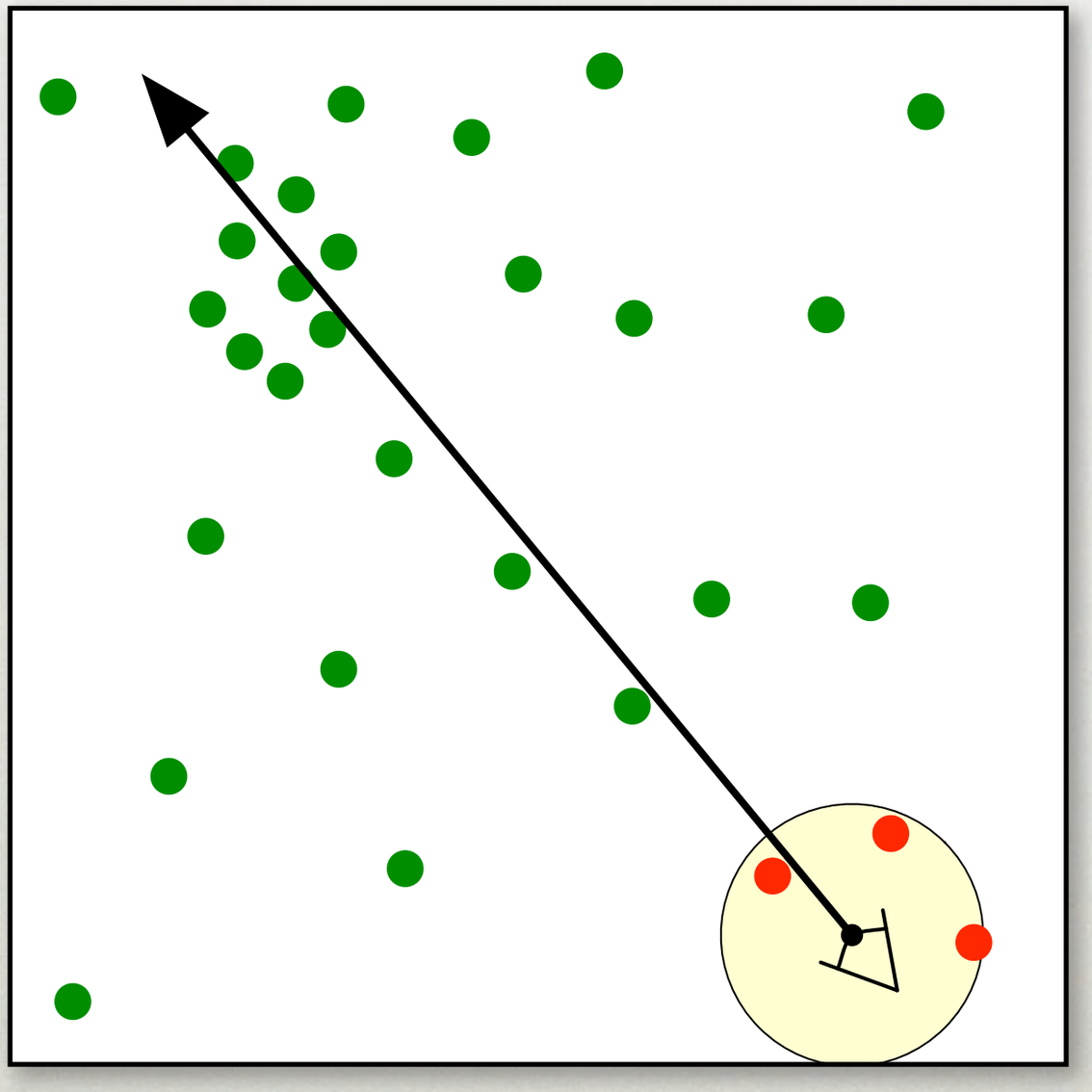
# Volumetric Photon Mapping

## Conventional Radiance Estimate



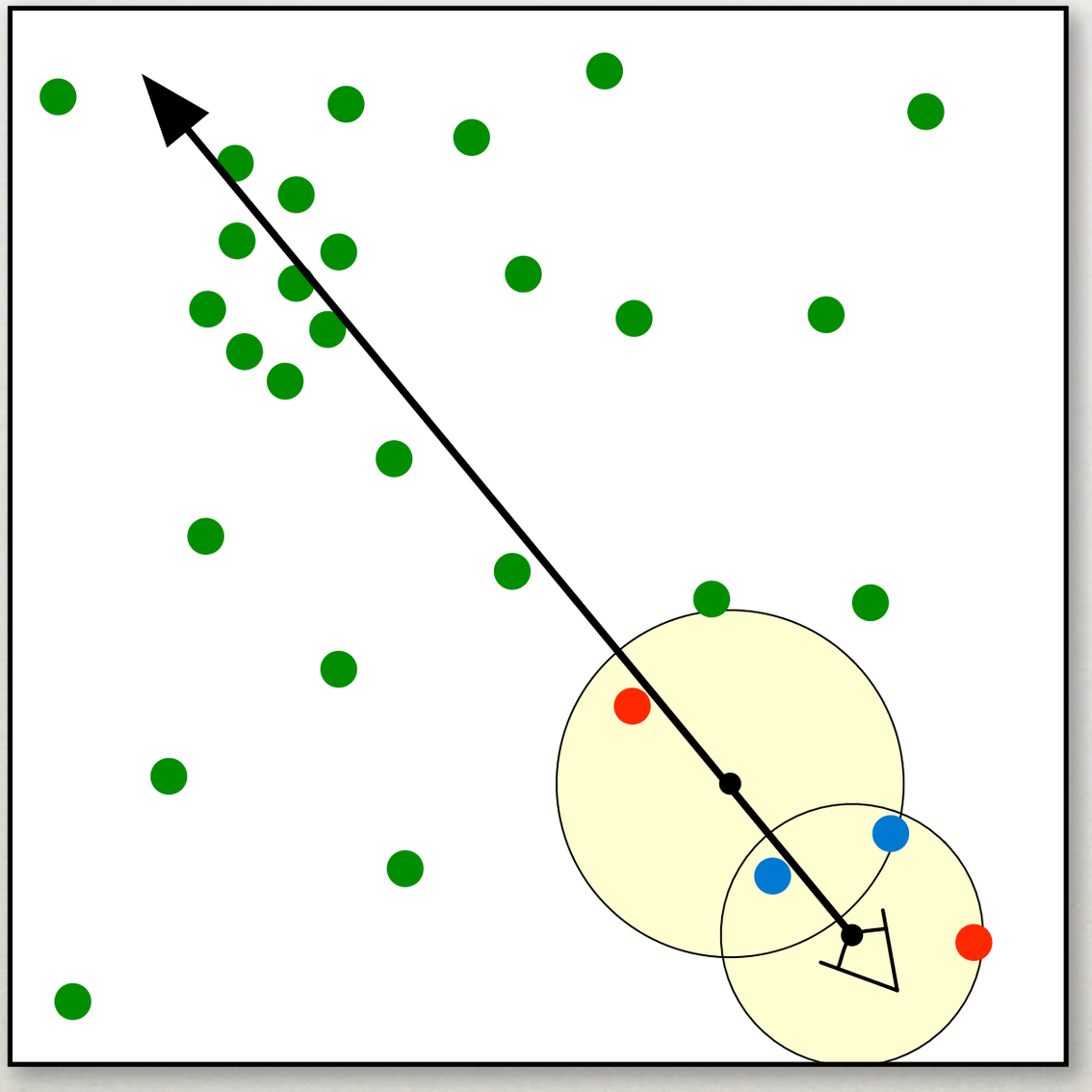
# Volumetric Photon Mapping

## Conventional Radiance Estimate



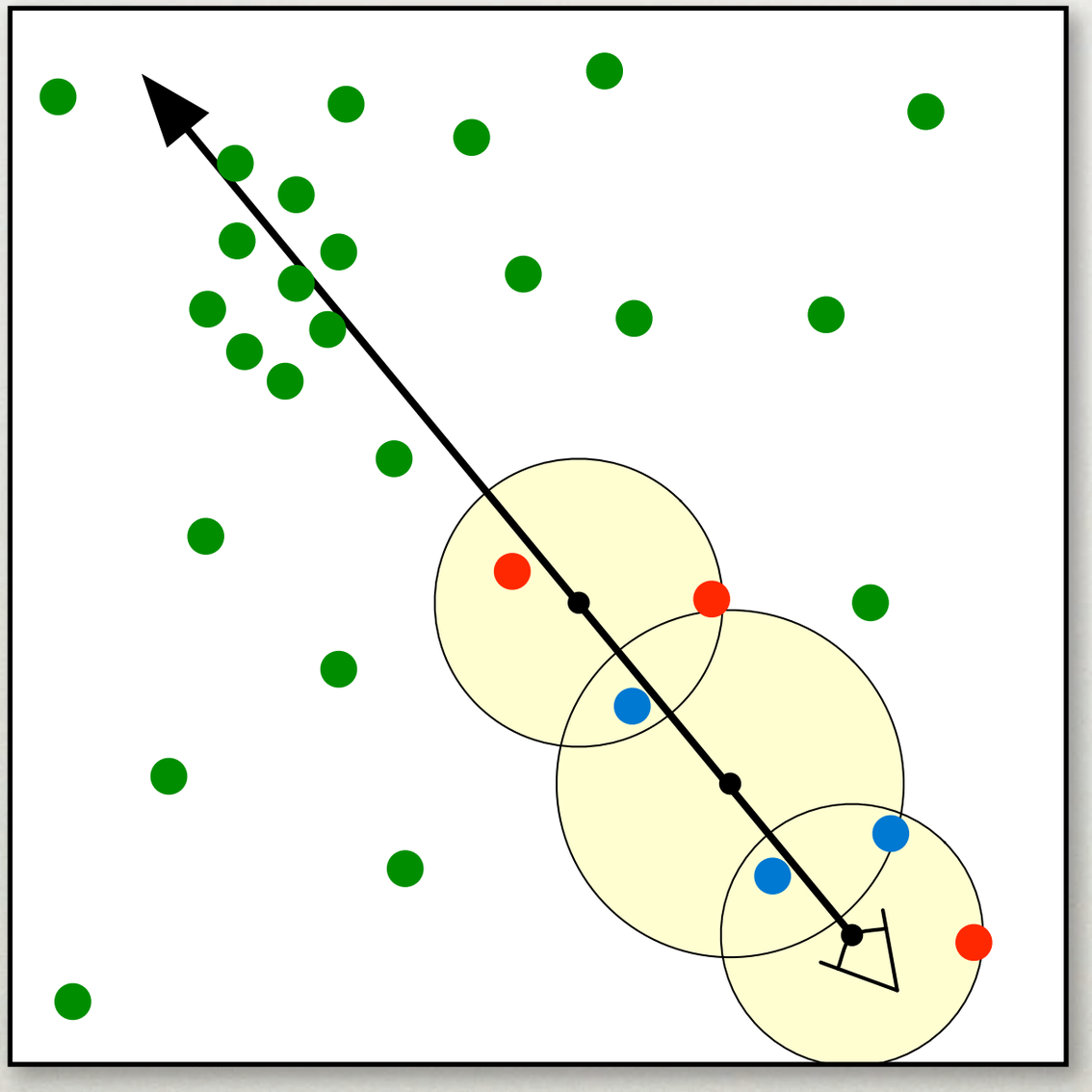
# Volumetric Photon Mapping

## Conventional Radiance Estimate



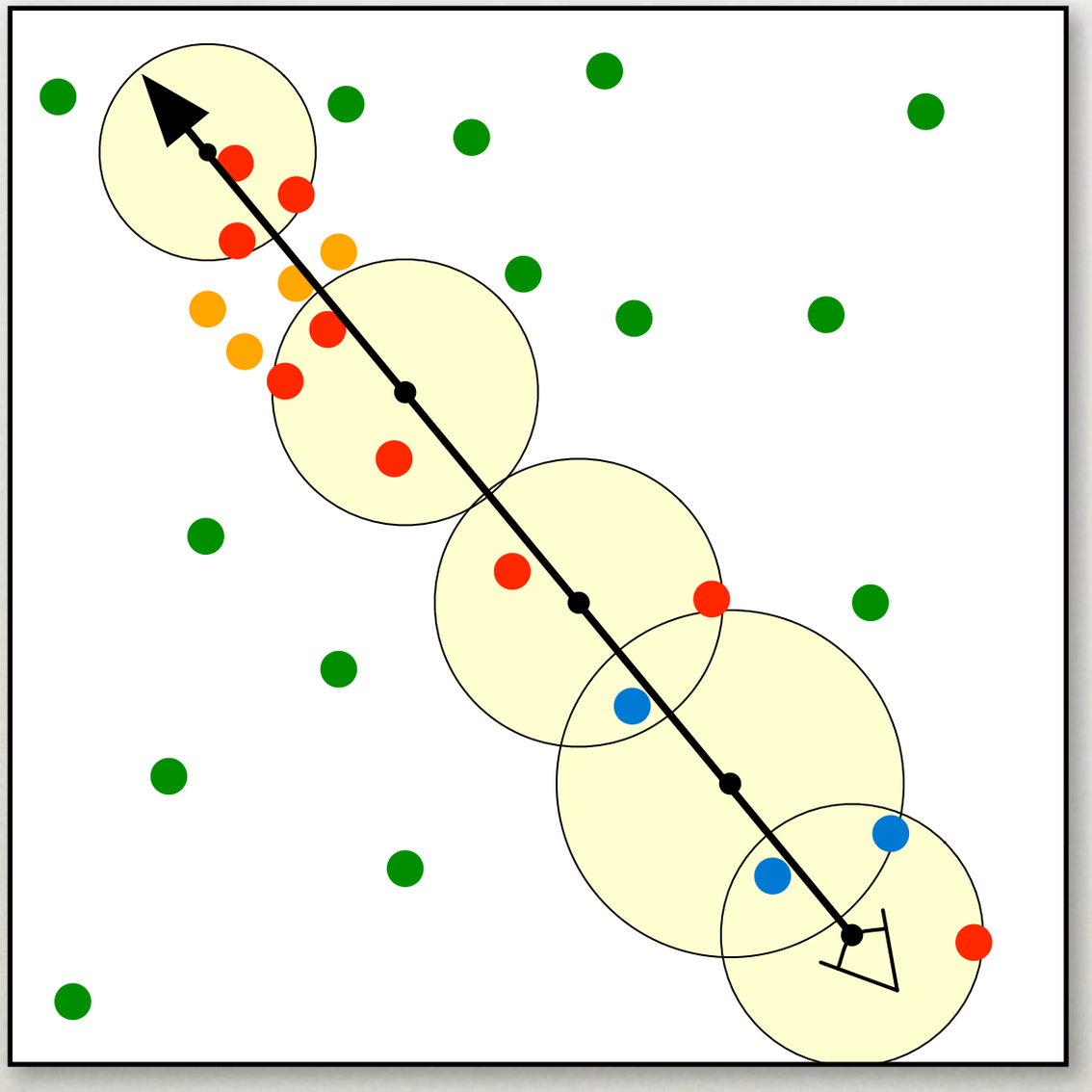
# Volumetric Photon Mapping

## Conventional Radiance Estimate

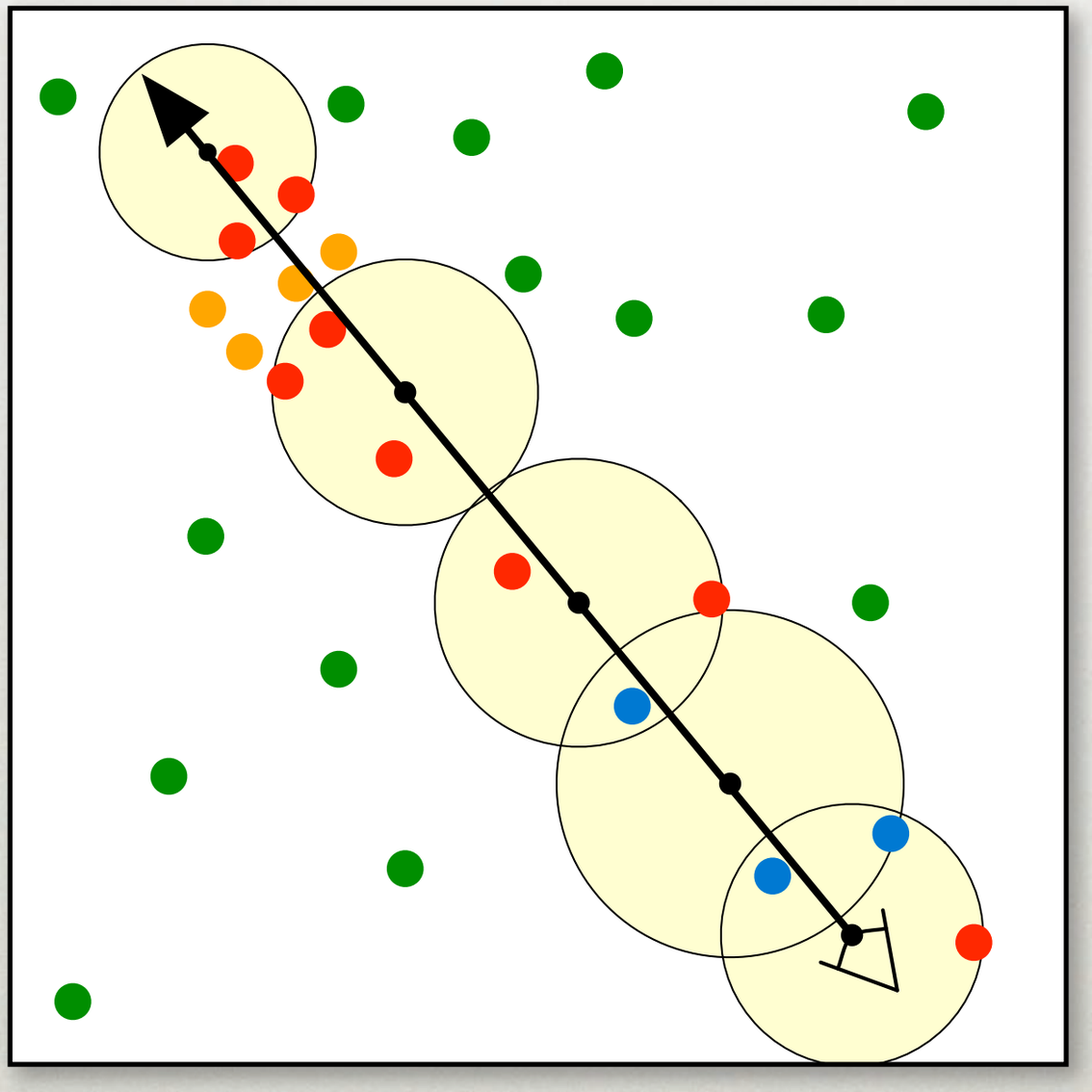


# Volumetric Photon Mapping

## Conventional Radiance Estimate

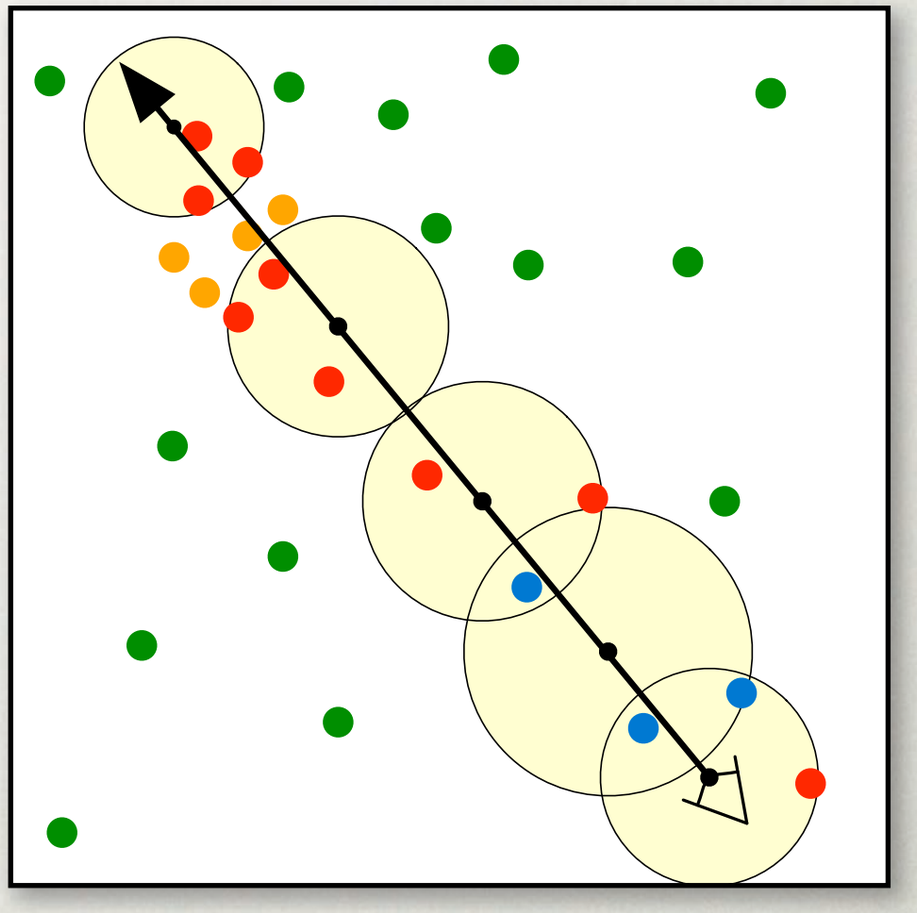


# Drawbacks



# Drawbacks

- Radiance estimation is expensive
  - Requires range search in photon map
- Performed numerous times per ray

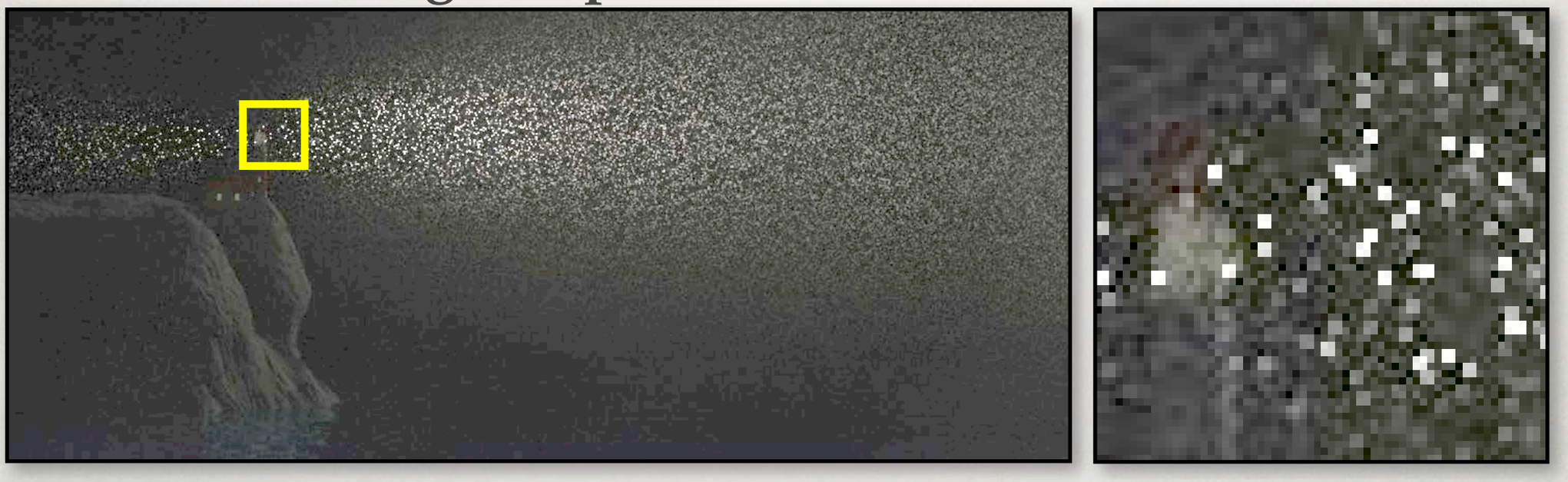


# Drawbacks

## Large Step-size

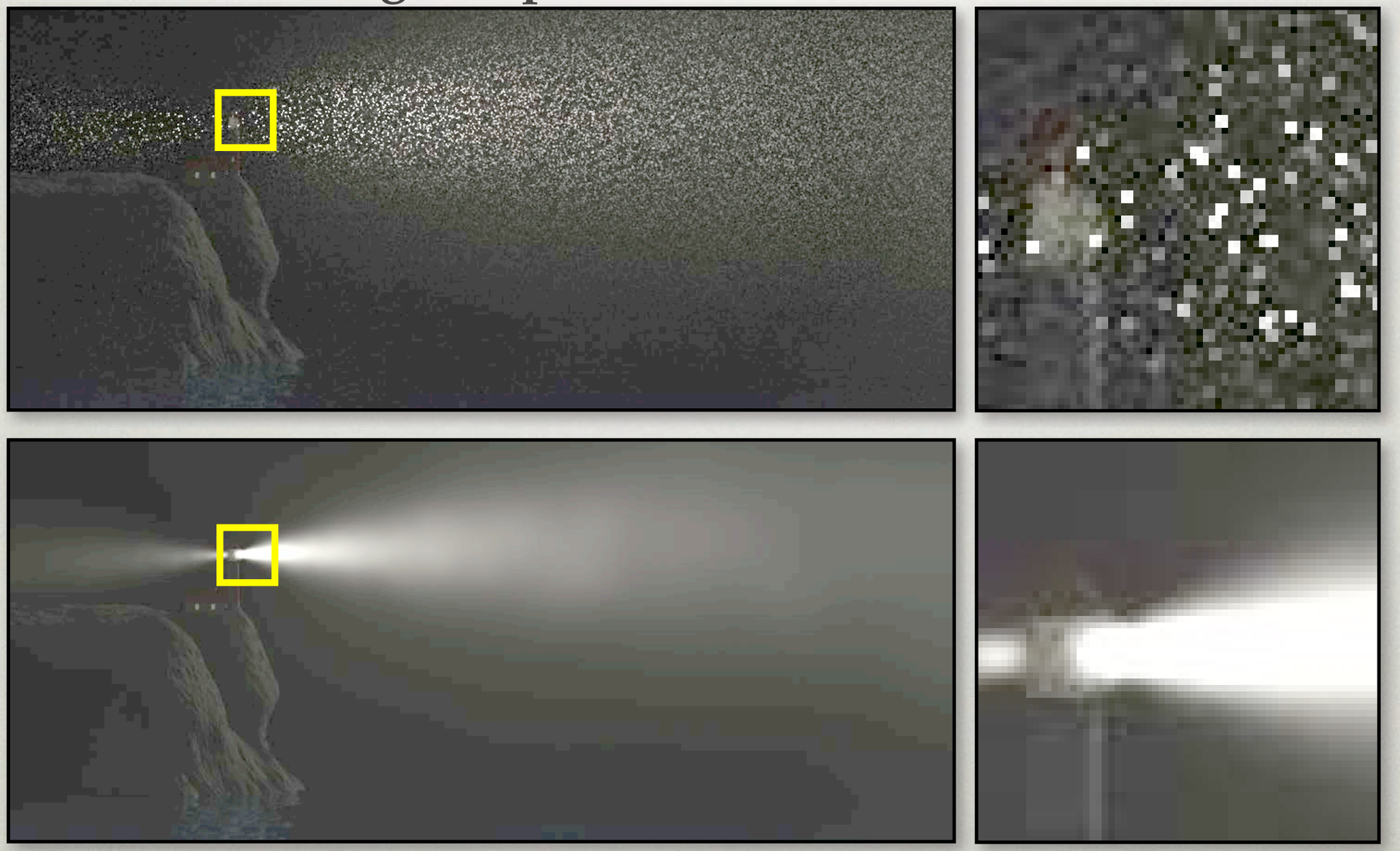
# Drawbacks

## Large Step-size



# Drawbacks

Large Step-size

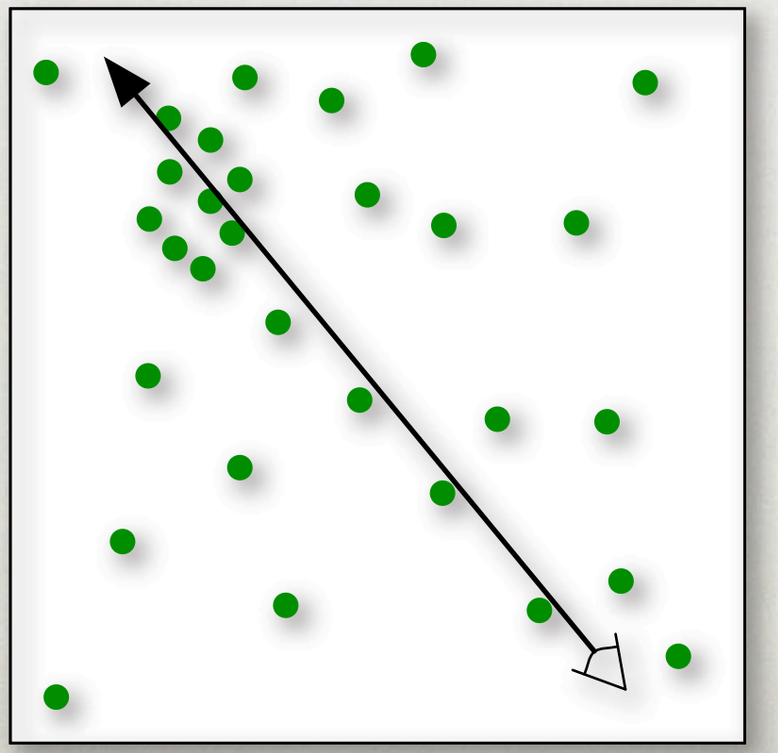


Very Small Step-size

# Volumetric Photon Mapping

## Conventional Radiance Estimate

$$L(\mathbf{x}, \vec{\omega}) \approx T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega}) + \left( \sum_{t=0}^{S-1} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, \vec{\omega}) \Delta_t \right)$$

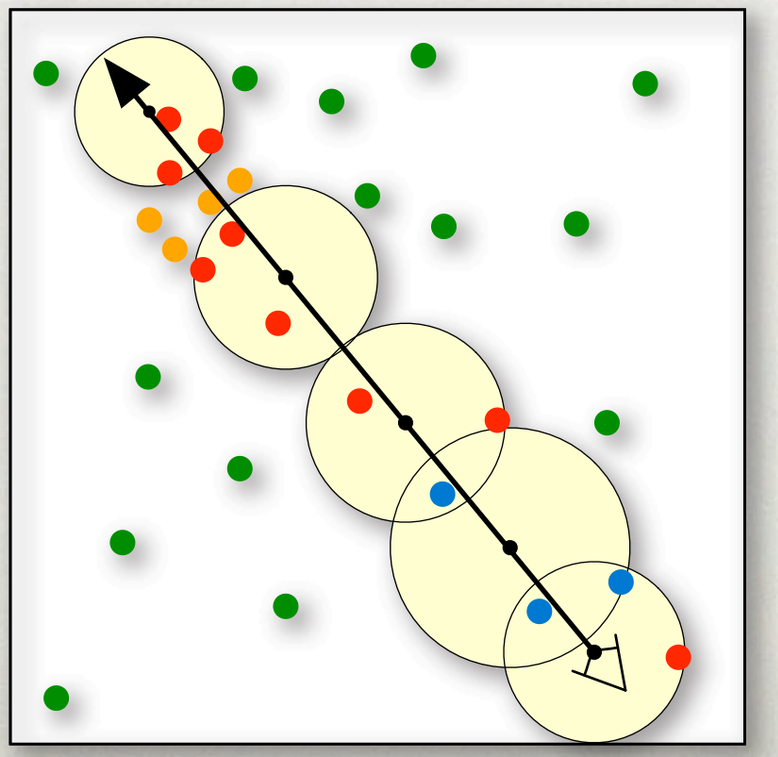


# Volumetric Photon Mapping

## Conventional Radiance Estimate

$$L(\mathbf{x},\vec{\omega}) \approx T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega}) + \left(\sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t\right)$$

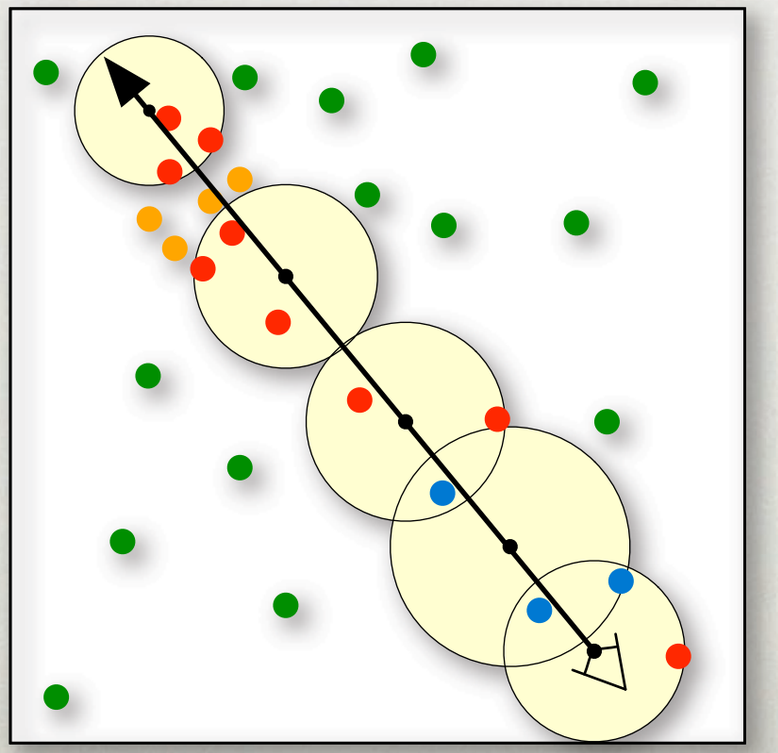
$$L_i(\mathbf{x}_t,\vec{\omega}) \approx \sum_{p=1}^{n} \frac{p(\mathbf{x}_t,\vec{\omega},\vec{\omega}_p)\Delta\Phi_p}{\frac{4}{3}\pi r^3}$$



# Volumetric Photon Mapping

## Conventional Radiance Estimate

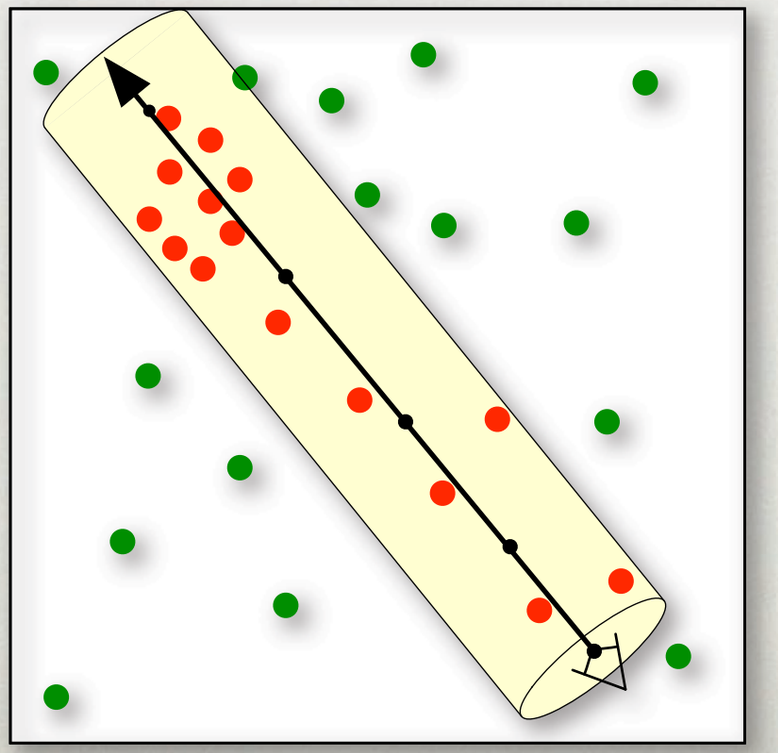
$$L(\mathbf{x},\vec{\omega}) \approx T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega}) + \left(\sum_{t=0}^{s-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\,\rho_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t\right)$$



# Volumetric Photon Mapping

## Beam Radiance Estimate

$$L(\mathbf{x}, \vec{\omega}) \approx T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega}) + L_b(\mathbf{x}, \vec{\omega})$$

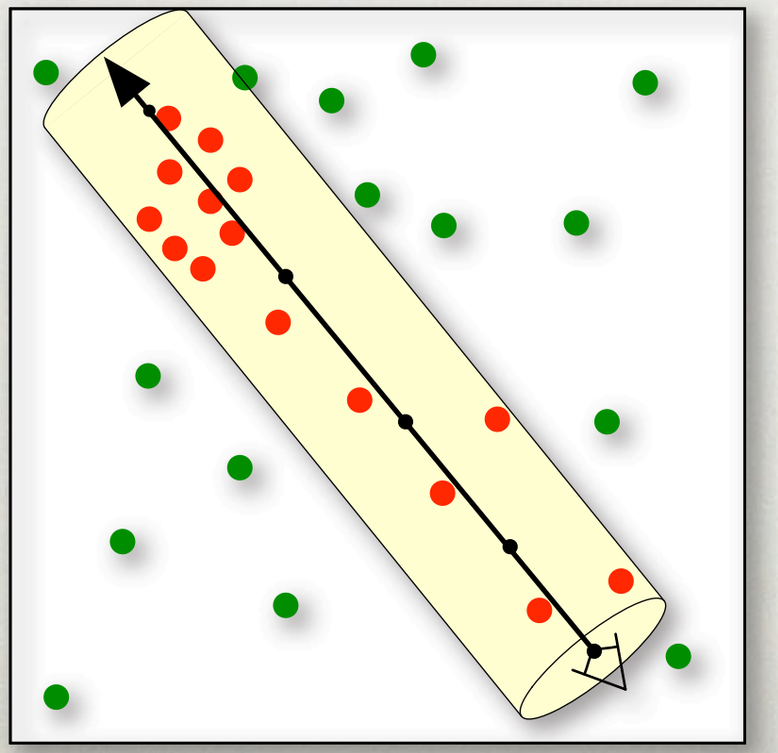


# Volumetric Photon Mapping

## Beam Radiance Estimate

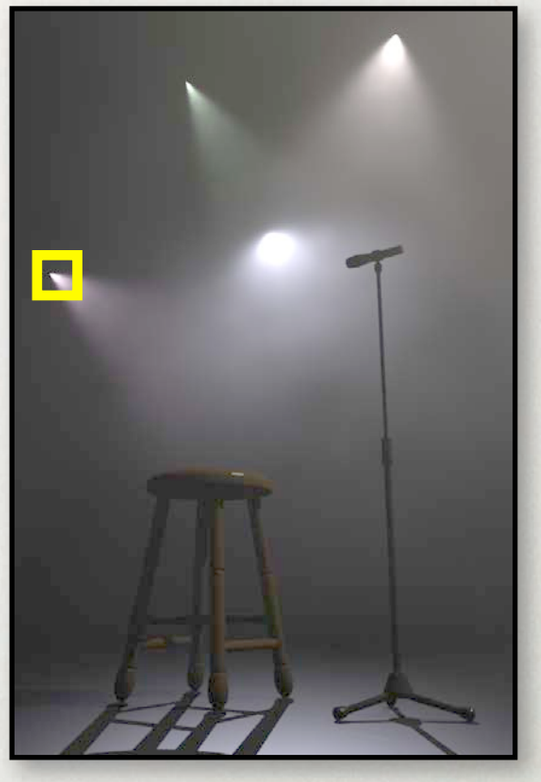
$$L(\mathbf{x}, \vec{\omega}) \approx T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega}) + L_b(\mathbf{x}, \vec{\omega})$$

$$L_b(\mathbf{x}, \vec{\omega}) = \frac{1}{N} \sum_{i=1}^{N} K_i T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \vec{\omega}_i) \alpha_i$$



# Adaptive Radius Comparison

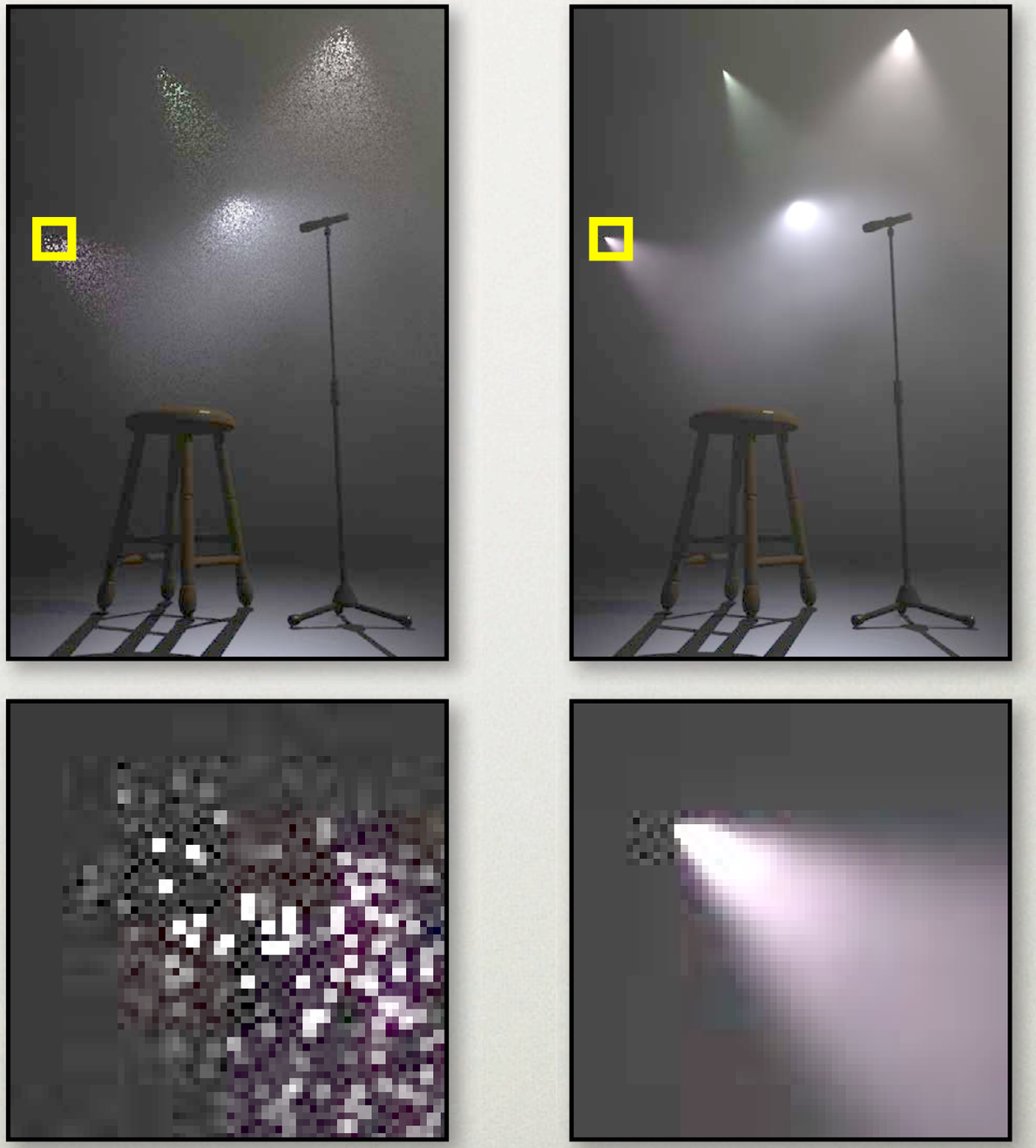
## Beam Estimate



(6:22)

# Adaptive Radius Comparison

Conv. Estimate  Beam Estimate



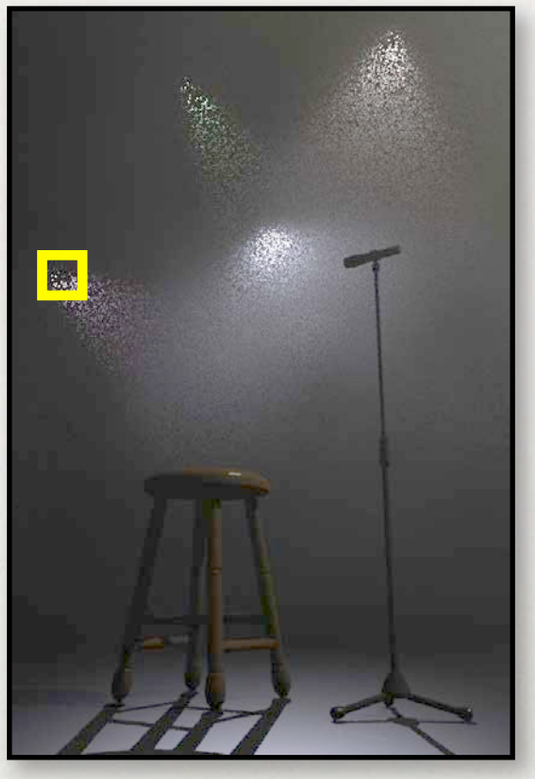
(6:38)  (6:22)

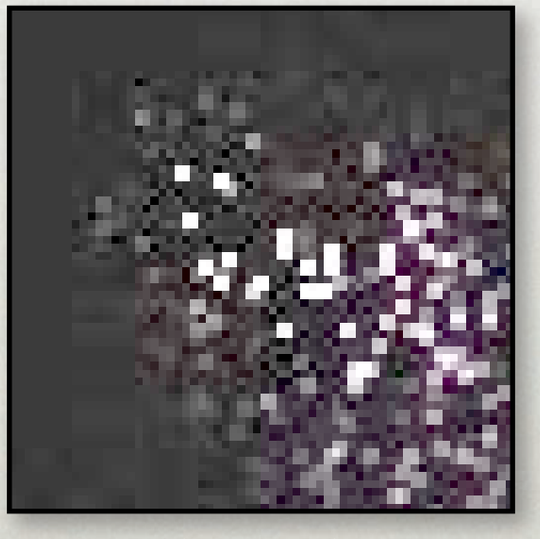
# Adaptive Radius Comparison

Conv. Estimate

(∞)

Conv. Estimate





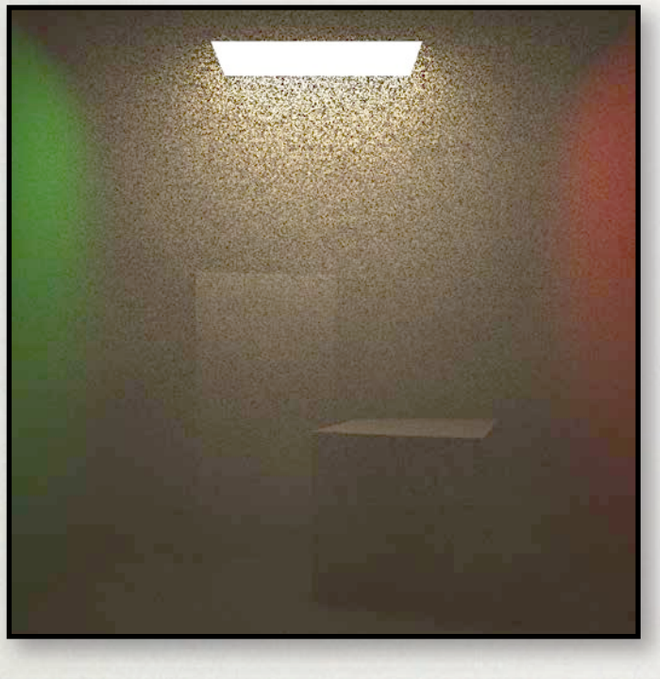
(6:38)

Beam Estimate

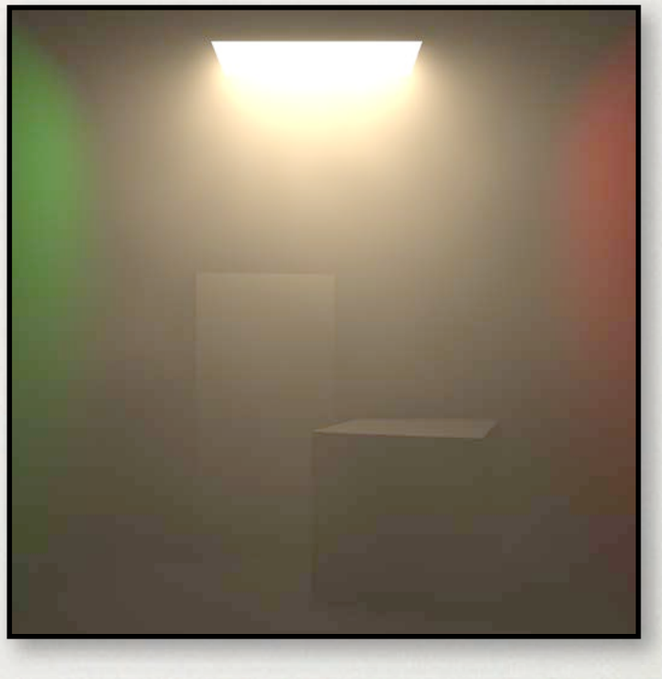
(6:22)

# Smoky Cornell Box

## Conv. Estimate

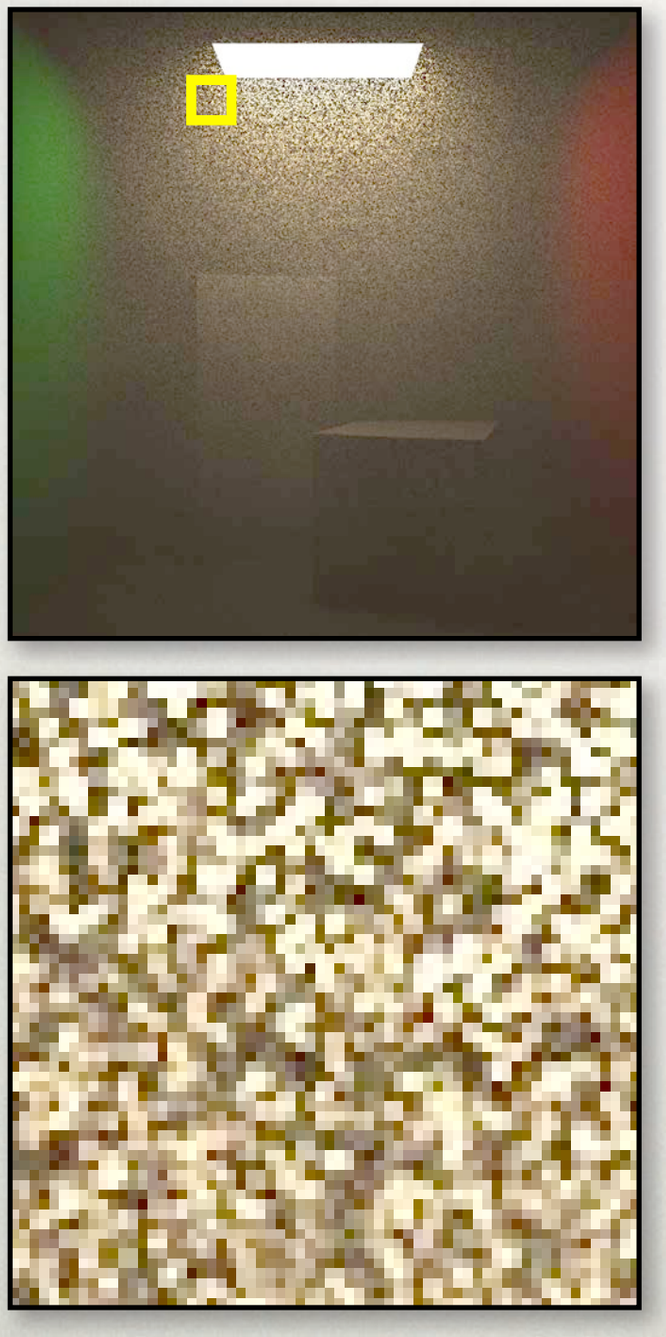


## Beam Estimate



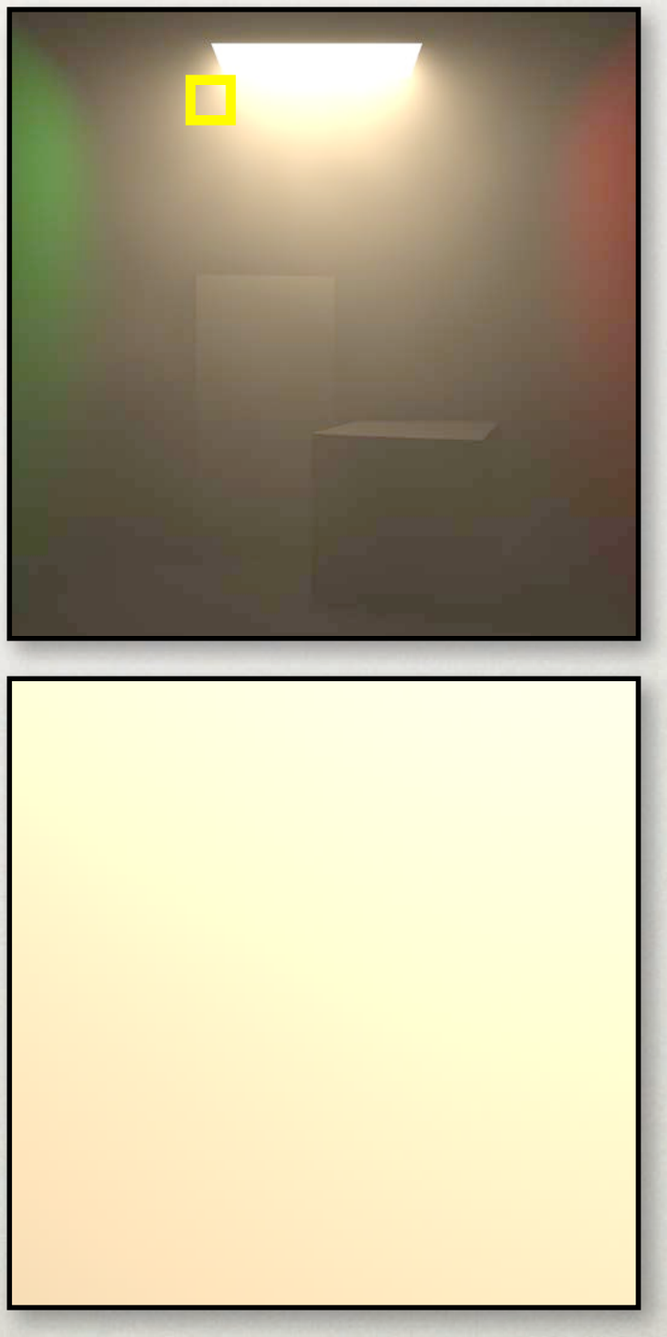
# Smoky Cornell Box

## Conv. Estimate



(4:03)

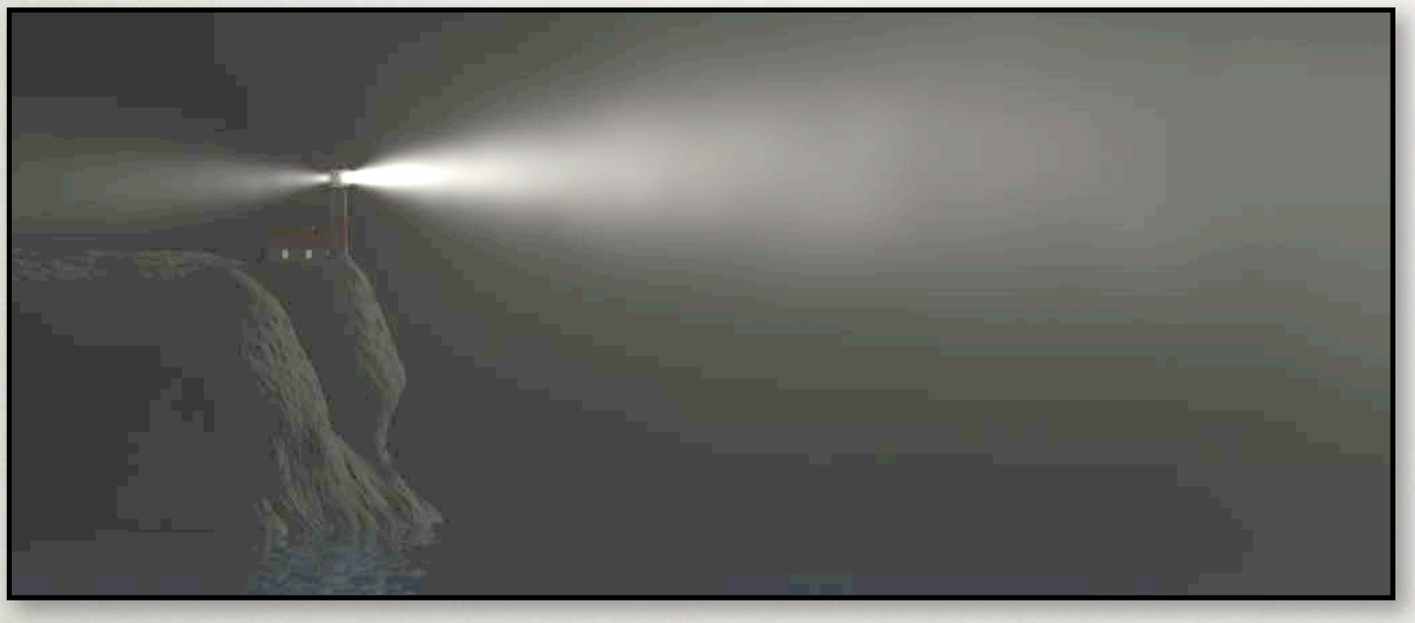
## Beam Estimate



(3:35)

# Lighthouse

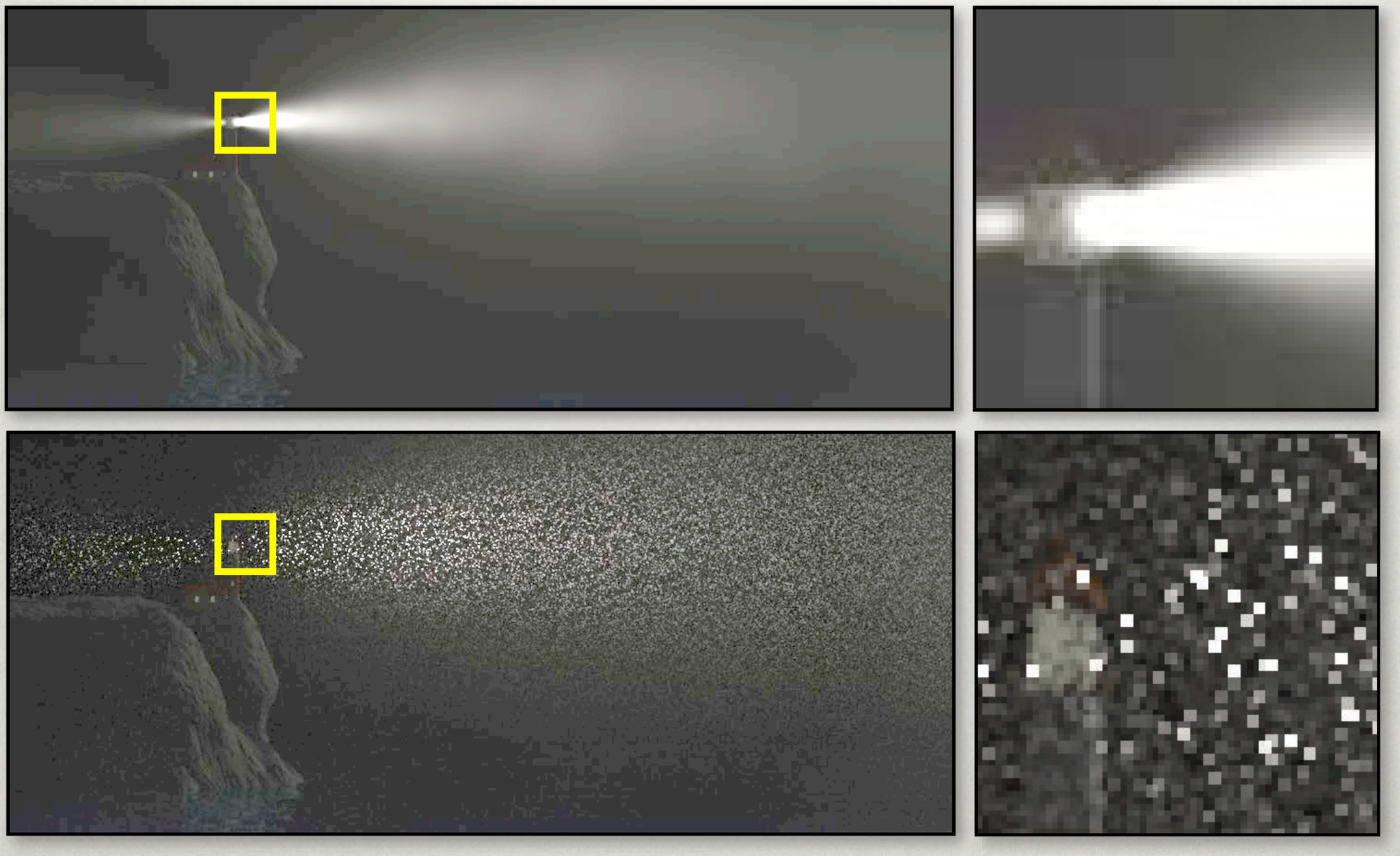
## Beam Estimate



## Conventional Estimate

# Lighthouse

**Beam Estimate** (1:05)



**Conventional Estimate** (1:12)

# Cars on Foggy Street

## Beam Estimate

## Conventional Estimate

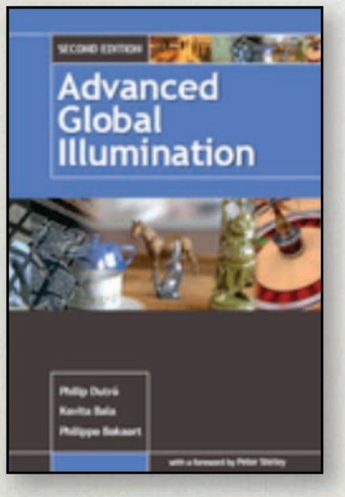
# Cars on Foggy Street

**Beam Estimate** (1:53)

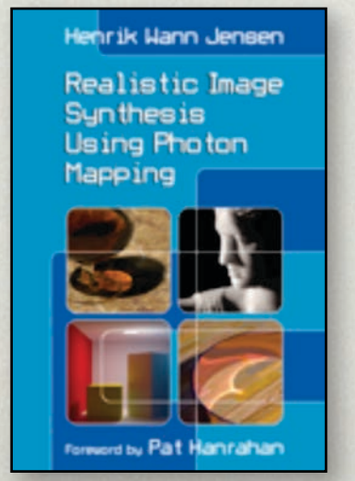
**Conventional Estimate** (2:02)

# For More Information


*Physically Based Rendering.*
Matt Pharr and Greg Humphreys




*Advanced Global Illumination.*
Philip Dutre, Kavita Bala, and Philippe Bekaert




*Realistic Image Synthesis Using Photon Mapping.*
Henrik Wann Jensen

# Questions?

Eurographics Symposium on Rendering (2005)
Kavita Bala, Philip Dutré (Editors)

# Non-Linear Volume Photon Mapping

Diego Gutierrez†, Adolfo Munoz, Oscar Anson and Francisco J. Seron

GIGA, Universidad de Zaragoza, Spain

**Abstract**

*This paper describes a novel extension of the photon mapping algorithm, capable of handling both volume multiple inelastic scattering and curved light paths simultaneously. The extension is based on the Full Radiative Transfer Equation (FRTE) and Fermat's law, and yields physically accurate, high-dynamic data than can be used for image generation or for other simulation purposes, such as driving simulators, underwater vision or lighting studies in architecture. Photons are traced into the participating medium with a varying index of refraction, and their curved trajectories followed (curved paths are the cause of certain atmospheric effects such as mirages or rippling desert images). Every time a photon is absorbed, a Russian roulette algorithm based on the quantum efficiency of the medium determines whether the inelastic scattering event takes place (causing volume fluorescence). The simulation of both underwater and atmospheric effects is shown, providing a global illumination solution without the restrictions of previous approaches.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

## 1. Introduction

Simulation of nature has always been one of the loftiest goals of computer graphics, providing a rich range of visual phenomena. Most of the times, the effect to be reproduced can be faked using a top-down approach, where the final desired result guides the implementation. This usually turns out relatively fast, ad-hoc methods that yield more than acceptable results. However, a physically correct simulation is necessary in certain fields where accuracy is a must. Underwater vision, driving simulators, the military, architectural lighting design etc. are fields where it is not enough to render an image which resembles reality. Predictive algorithms must be developed instead, where the image is the final visualization of the physically correct data generated. A bottom-up approach is then necessary: first, the basic laws of physics that govern the phenomenon need to be described and fed to the rendering system; the phenomenon itself will just be the logical, inevitable output. This approach sacrifices rendering speed in exchange for reliable, physically accurate numerical data that can be used for purposes beyond image generation.

Two of the greatest sources of visually appealing phenomena in nature are participating media and a varying index of refraction. Participating media are the cause of such well-known effects such as fog, clouds or blurry underwater vision, whereas a varying index of refraction yields mirages, rippling images, twinkling stars or some spectacular sunsets. Sources of inelastic scattering in ocean waters can greatly affect visibility and alter its color, whereas distortions caused by temperature differences can further alter the perception of things in such environment. Simulating underwater rescue missions, laying submarine data cables or even the correct interpretation of ancient World Heritage sites can benefit from an accurate description of light that includes an ampler range of phenomena.

We present in this paper a physically-based spectral simulation of light, solving the Full Radiative Transfer Equation (FRTE) and applying Fermat's law, which includes multiple inelastic scattering as well as an accurate description of the non-linear paths followed by the light rays in media with a varying index of refraction. It is based on an extension of the volume photon map algorithm presented by Wann Jensen and Christensen [JC98]. The main contributions are a full global illumination solution which supports non-linear light

† e-mail: diegog@unizar.es

paths and is free of the restrictions of previous works, and the physically-correct simulation of volume fluorescence in participating media, caused by inelastic scattering, including efficient computation of caustics. Atmospheric effects and underwater imagery are simulated as case studies to demonstrate the algorithm. To our knowledge, there is no previous research in computer graphics literature that models together physically-based inelastic scattering in participating, inhomogeneous media where the index of refraction varies continuously. Related previous works therefore span two different categories: inelastic scattering in participating media and non-linear light propagation.

Rendering participating media is not a new field in computer graphics, and an exhaustive review can be found in [PPS97]. There are two types of scattering events in a participating medium: *elastic* scattering, where no transfer of energy occurs between wavelengths, and *inelastic* scattering, where such energy transfers do occur, from shorter to longer wavelengths. Spectral global illumination algorithms that handle participating media only take into account elastic scattering, with the strategy consisting on decoupling the solutions for each sampled wavelength, then adding them to obtain the final image. No interaction between wavelengths is computed. To the authors' knowledge, the only previous work that simulates volume inelastic scattering in participating media is owed to Cerezo and Seron [CS03], using a discrete ordinate method. Unfortunately their method requires both rectangular meshing of the geometry, as well as an angular and spatial discretization which imposes high memory requirements, thus limiting the complexity of the scenes that can be reproduced (the problem is aggravated when simulating highly anisotropic scattering). They also cannot provide a full solution, failing to render caustics. Surface inelastic scattering works include [Gla95b] or [WTP01], but their methods are not extensible to participating media.

With respect to non-linear ray tracing, the first method to deal with non-straight light paths is owed to Berger et al. [BTL90], refracting the ray according to Snell's law in each of a series of flat homogeneous layers, thus achieving a piece-wise linear approximation of a curved path. This was challenged by Musgrave [Mus90], who develops a purely reflective model where rays follow a parabolic path, following the Kuhlar/Fabri physical model [FFLV82]. A more general approach to non-linear ray tracing is proposed by Gröller [Grö95], although the work does not study the influence of the index of refraction in the curvature of the rays, visualizing mathematical and physical systems instead. In the paper by Stam and Languenou [SL96], the authors use geometrical optics to describe how light bends if the index of refraction of the medium varies continuously. They nevertheless fail to provide a physically-based analytical expression for the index of refraction as a function of temperature and wavelength, and solve the equations only for two specific cases, thus losing generality. Seron et al. [SGGC05] implement a method of curved ray tracing capable of simulating the inferior mirage and some sunset effects, although they do not attempt to calculate any lighting, deforming pre-lit textures instead. In [HW01] gravitational light bending is visualized according to the theory of general relativity, whereas other relativity- and physics-related papers include the bending caused by neutron stars or black holes [Nem93], so they cannot (nor pretend to) simulate the phenomena described in this paper. Yngve et al. [YOH00] describe a simple method to simulate the bending of light by interpolating a density field, but they need to exaggerate the variation of the index of refraction tenfold for the effect to be visible.

The paper is organized as follows: section 2 provides the physically-based background, with an overview of inelastic scattering, the FRTE and the Fermat's law. In section 3 we describe our extension of the volume photon map algorithm to include inelastic scattering and curved light paths, with sections 4 and 5 providing case studies of underwater imagery and atmospheric effects respectively. The discussion of the results and some additional images are presented in section 6, to finish the paper in section 7 with the conclusions and future work.

## 2. Physically-based Framework

We now present the physical framework of our work, by first introducing what inelastic scattering is, then deriving the FRTE that needs to be solved to account for it. In order to be able to compute non-linear light paths, we will use Fermat's law to obtain the correct trajectories.

### 2.1. Inelastic scattering

Inelastic scattering implies an energy transfer from wavelength $\lambda'$ to $\lambda$, with $\lambda' < \lambda$ within the visible spectrum, and gives rise to fluorescence and phosphorescence phenomena. Fluorescence occurs when a molecule absorbs a photon of wavelength $\lambda'$ (called excitation wavelength), and re-emits it at a longer wavelength $\lambda$ according to a *fluorescence efficiency function* $P_f(\lambda)$. The time lapse between the two events is $10^{-11}$ to $10^{-8}$ seconds, so for computer graphics it can be taken as an instantaneous process. For pure substances, re-emission is isotropic and the wavelength of the re-emitted photons is independent of the different excitation wavelengths, although the intensity of the re-emission does depend on them. Phosphorescence is a similar process, governed by the *phosphorescence efficiency function*, with the main difference being that the re-emitted energy declines with time according to a function $d(t)$.

### 2.2. Full Radiative Transfer Equation

Usually, participating media algorithms solve the integro-differential Radiative Transfer Equation (RTE), which takes into account emission, absorption and elastic scattering, but

does not yield a solution for inelastic scattering events. Following the notation in [JC98], and reformulating to include wavelength dependencies, the RTE can be written as:

$$\frac{\partial L_\lambda(x,\vec{w})}{\partial x} = \alpha_\lambda(x)L_{e,\lambda}(x,\vec{w}) + \sigma_\lambda(x)L_{i,\lambda}(x,\vec{w}) - \alpha_\lambda(x)L_\lambda(x,\vec{w}) - \sigma_\lambda(x)L_\lambda(x,\vec{w}) \quad (1)$$

where $\frac{\partial L(x,\vec{w})}{\partial_x}$ represents the variation of radiance $L$ at a point $x$ in the direction $\vec{w}$, $\alpha$ and $\sigma$ are the absorption and scattering coefficients, $L_e$ is the emitted radiance and $L_i$ is the in-scattered radiance. Defining the extinction coefficient as $\kappa_\lambda(x) = \alpha_\lambda(x) + \sigma_\lambda(x)$ and integrating $L_{i,\lambda}$ over the sphere $\Omega$ we get:

$$\frac{\partial L_\lambda(x,\vec{w})}{\partial x} = \alpha_\lambda(x)L_{e,\lambda}(x,\vec{w}) + \sigma_\lambda(x)\int_\Omega p_\lambda(x,\vec{w}',\vec{w})L_\lambda(x,\vec{w}')d\vec{w}' - \kappa_\lambda(x)L_\lambda(x,\vec{w}) \quad (2)$$

which is the integro-differential, wavelength-dependent RTE governing the transport of light in participating media, with $p_\lambda(x,\vec{w}',\vec{w})$ being the phase function that defines the re-emission direction. However, this equation does not account for energy transfers between wavelengths, the phenomenon known as inelastic scattering. To be able to compute these inelastic scattering events, we need to develop the RTE equation further, by adding a term that accounts for such energy transfers. This term can be expressed as a double integral over the domains of the solid angle and wavelength:

$$\int_\Omega \int_\lambda \alpha_{\lambda_i}(x) f(x,\lambda_i \to \lambda) L_{\lambda_i}(x,\vec{w}') \frac{p_\lambda(x,\vec{w}'_i,w)}{4\pi} d\vec{w}_i d\lambda_i \quad (3)$$

where $\alpha_{\lambda_i}$ is the absorption coefficient for wavelength $\lambda_i$ (remember there is no inelastic scattering without previous absorption), $f(x,\lambda_i \to \lambda)$ is the function that governs the efficiency of the energy transfer between wavelengths, defined as the probability of a photon of $\lambda_i$ being re-emitted at $\lambda$. For fluorescence and phosphorescence, this phase function is isotropic [Mob94]. Adding this term to the RTE (equation 2) we obtain the FRTE:

$$\frac{\partial L_\lambda(x,\vec{w})}{\partial x} = \alpha_\lambda(x)L_{e,\lambda}(x,\vec{w}) + \sigma_\lambda(x)\int_\Omega p_\lambda(x,\vec{w}',\vec{w})L_\lambda(x,\vec{w}')d\vec{w}' - \kappa_\lambda(x)L_\lambda(x,\vec{w}) + \int_\Omega \int_\lambda \alpha_{\lambda_i}(x) f(x,\lambda_i \to \lambda) L_{\lambda_i}(x,\vec{w}') \frac{p_\lambda(x,\vec{w}'_i,w)}{4\pi} d\vec{w}_i d\lambda_i \quad (4)$$

which is the equation that must be solved to take into account multiple inelastic scattering in participating media, thus being able to render volume fluorescence effects.

### 2.3. Varying index of refraction in inhomogeneous media

A varying index of refraction $n_\lambda$ defines an inhomogeneous medium where light travels in curved paths. These curved paths result in a distorted image, with the mirages being probably the best known manifestation of the effect. To be able to simulate this type of inhomogeneous medium, we therefore need to obtain the curved trajectory of light as it traverses it. The direction $\vec{w}$ in equation 4 therefore needs to be recomputed at each differential step, accounting for the changes in $n_\lambda$. We obtain this corrected direction at each step by solving Fermat's law, which defines how light traverses one given medium.

The following derivation of Fermat's law uses the work of Gutierrez et al. [GSMA04] and is not meant to be exhaustive. As stated in [Gla95a], *a ray of light, when travelling from one point to another, follows a path that corresponds to a stationary value of the optical path length* (OPL). The OPL is defined as the index of refraction times the travelled path (or the distance the light would have travelled in a vacuum during the flight time through the material), and in its differential form it can be formulated as $d(OPL) = ndl$, where $l$ is the path travelled by the light ray. The equation shows how light gets bent towards the areas with a greater index of refraction, as Snell's law also predicts for the boundary of two homogeneous media. A stationary value corresponds to a maximum or a minimum in the function, thus the derivative equals zero. We can therefore write:

$$\delta(OPL) = \delta\int_A^B ndl = \int_A^B \delta n dl + \int_A^B n\delta(dl) = \int_A^B \frac{\delta n}{\delta x_i}\delta x_i dl + \int_A^B n\delta(dl) = 0 \quad (5)$$

where $x_i$ are the vector components of $l$. Considering $dx_i$ as variables and taking increments we get $\delta(dl) = \frac{dx_i}{dl}\delta(dx_i)$. Since light trajectories start and end at the stationary points $A$ and $B$, we get $\delta x_i(A) = 0$ and $\delta x_i(B) = 0$. Equation 5 then results:

$$\delta L = \int_A^B \left[\frac{\partial n}{\partial x_i} - \frac{d}{dl}\left(n\frac{dx_i}{dl}\right)\right]\delta x_i dl = 0 \quad (6)$$

Since this equation must hold for any value of $\delta x_i$, the integrand must equal zero, so we finally come up with the equation that must be solved to obtain the path followed by light while traversing any medium, as a function of the index of refraction at each point:

$$\frac{d}{dl}\left(n\frac{d\vec{r}}{dl}\right) - \nabla n = 0 \Leftrightarrow \frac{d}{dl}\left(n\frac{dx_j}{dl}\right) - \frac{\partial n}{\partial x_j} = 0\,(j = 1,2,3) \quad (7)$$

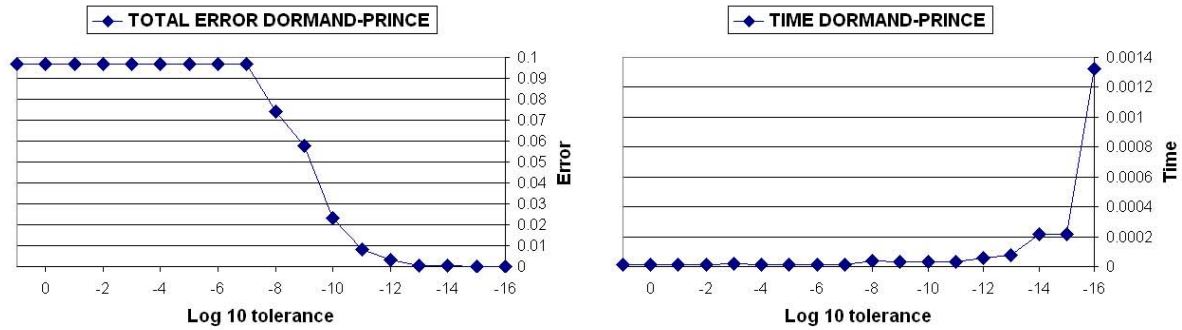



**Figure 1:** *Error and rendering time (secs.) as functions of the error tolerance in the Dormand-Prince RK5(4)7M method for a test scene.*

where $\vec{r} = x_j$ are the coordinates $(x, y, z)$ of each point. This equation cannot be solved analytically, and thus we must apply a numerical method. We now need to rewrite equation 7 in order to solve it in a more efficient way than the Euler method presented in [GSMA04]:

$$\frac{d^2 x_j}{dl^2} = \frac{1}{n}\left(\frac{\partial n}{\partial x_j} - \frac{dn}{dl}\frac{dx_j}{dl}\right) \tag{8}$$

Doing the change of variable $y_j = \frac{dx_j}{dl}$ we obtain:

$$y_j' = \frac{1}{n}\left(\frac{\partial n}{\partial x_j} - \frac{dn}{dl} y_j\right) \tag{9}$$

where $\frac{dn}{dl} = \frac{dn}{dx_j}\frac{dx_j}{dl}$. The change of variable can also be written as:

$$x_j' = y_j \tag{10}$$

Equations 9 and 10 define a system where $x_j$ represents the position and $y_j$ the velocity at a given point in the trajectory, which can be written in matrix form as:

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix}' = \begin{pmatrix} y_j \\ \frac{1}{n}\left(\frac{\partial n}{\partial x_j} - \frac{dn}{dl} y_j\right) \end{pmatrix} \tag{11}$$

This equation 11 has the form $Y' = f(l, Y)$, which defines an Initial Value Problem with $Y(0) = \alpha$. We solve this problem by applying the embedded Runge-Kutta method RK5(4)7M from the Dormand-Prince family. A detailed description of the method and the error tolerance can be found in [DP80].

We have tested the implementation in a simple scene where the index of refraction varies according to the equation $n = 1 + ky$, with $y$ representing height, and $k$ varying from -0.1 to 0.1. This distribution of $n$ can be solved analytically, so we can measure the numerical error against the exact solution. Figure 1 shows the error of the Dormand-Prince RK5(4)7M method as the tolerance is reduced, along with the time it takes to reach the solution. As it can be seen, error tolerances in the range of $10^{-8}$ to $10^{-12}$ yield good results without much of a time penalty. Error tolerances beyond $10^{-14}$ start increasing rendering times considerably.

## 3. Extension of the Volume Photon Mapping Algorithm

Ray tracing techniques involve shooting rays into the scene from the camera and following them to detect hits with the geometry, then shooting shadow rays to the lights to find out direct illumination. With curved light paths this turns out to be highly impractical, though, since finding the ray with the physically-correct curvature which goes from the intersection point to the light is computationally very expensive (or the solution might not even exist). Groeller [Grö95] proposes three solutions: considering shadow rays to follow straight paths, retrieving all lighting information straight from the textures, and finally voxelizing the space and prestoring the approximated incident directions of light sources for each voxel, by launching rays from the light sources into the scene prior to the render pass. The first two are clearly not physically-based, while the third only approximates the solution with a preprocessing step.

In order to obtain a physically-based solution for multiple inelastic scattering in inhomogeneous media with a varying index of refraction $n$, we have extended the volume photon mapping algorithm [JC98] to account both for volume fluorescence and the distortions caused by the changing $n$.

For inelastic scattering, we need to model the possibility of an absorbed photon being re-emitted at a different wavelength. Equation 4 includes a term $f(x, \lambda_i \rightarrow \lambda)$ known as *wavelength redistribution function*, which represents the ef-

ficiency of the energy transfer between wavelengths. It is defined as the quotient between the energy of the emitted wavelength and the energy of the absorbed excitation wavelength, per wavelength unit. Reformulating in terms of photons instead of energy we have the *spectral quantum efficiency function* $\eta(x,\lambda_i \rightarrow \lambda)$, defined as the number of photons emitted at $\lambda$ per wavelength unit, divided by the number of absorbed photons at $\lambda_i$. Both functions are dimensional ($nm^{-1}$), and are related as follows:

$$f(x,\lambda_i \rightarrow \lambda) = \eta(x,\lambda_i \rightarrow \lambda)\frac{\lambda_i}{\lambda} \tag{12}$$

A related dimensionless function that describes inelastic scattering is the *quantum efficiency* $\Gamma$, defined as the total number of photons emitted at all wavelengths divided by the number of photons absorbed at excitation wavelength $\lambda_i$. It is related to the spectral quantum efficiency function by the equation:

$$\Gamma(\lambda_i) = \int_{\lambda} \eta(x,\lambda_i \rightarrow \lambda) d\lambda \tag{13}$$

Our extension to the volume photon mapping algorithm includes a) solving Fermat's law to obtain the curved trajectory of each photon if the index of refraction varies (and also for the eye rays shot during the radiance estimate phase), thus being able to overcome the shadow ray problem presented above and to obtain a full solution including effects such as color bleeding and caustics; and b) the inclusion of the quantum efficiency $\Gamma$ to govern the probability of an inelastic scattering event. As shown in figure 2, once the albedo-based Russian roulette determines that a certain photon has been absorbed by the medium, a second Russian roulette based on the quantum efficiency determines whether an inelastic scattering event takes place, and therefore the photon has to be re-emitted at a different wavelength. This is done by generating a random number $\xi_{in}[0,1]$ so that:

$$\xi_{in}[0,1] \rightarrow \begin{cases} \xi_{in} \leq \Gamma & \text{Photon is re-emitted} \\ \xi_{in} > \Gamma & \text{Photon remains absorbed} \end{cases} \tag{14}$$

If re-emitted, the new wavelength must be obtained, for which we must sample the spectral quantum efficiency function $\eta(x,\lambda_i \rightarrow \lambda)$ for the excitation wavelength $\lambda_i$. This can be simply done by rejection sampling the function, but to increase efficiency we perform importance sampling using the inverse of its cumulative distribution function (cdf). A random number $\psi[0,1]$ therefore yields the new wavelength for the re-emitted photon. Steeper areas of the cdf increase the probability of a photon being re-emitted at the corresponding wavelengths.

Figure 2 shows the basic scheme of the algorithm. The sequence of events in the original volume photon mapping by [JC98] is represented inside the grey area.

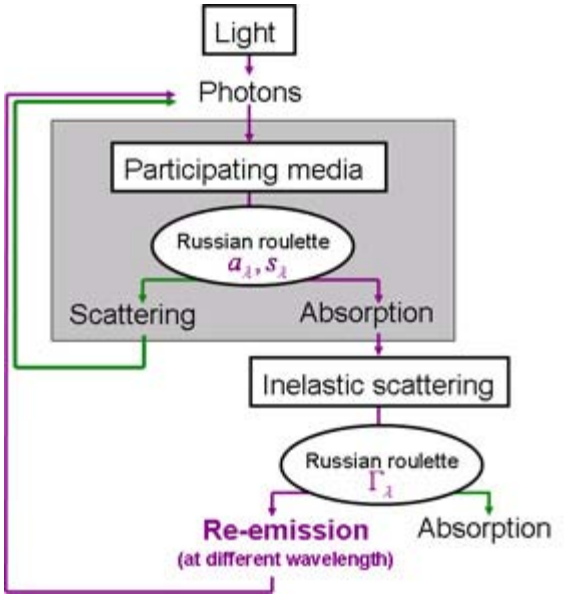



**Figure 2:** *Our extended volume photon mapping algorithm.*

## 4. Case Study: Underwater Imagery

We chose deep ocean waters as our first case study, given its rich range of elastic and inelastic scattering phenomena and the fact that it is a medium well studied by oceanographers. Pure seawater absorbs most wavelengths except for blue: the absorption coefficient peaks at 760 nanometers, and reaches a minimum at 430 nm. The phase function $p$ is modelled as the phase function in pure sea water plus the phase function of the scattering by suspended particles, as proposed in [Mob94] ($p = p_w + p_p$). For pure water we use a phase function similar to Rayleigh's:

$$p_w(\theta) = 0.06225(1 + 0.835cos^2\theta) \tag{15}$$

while the scattering caused by particles is modelled using a Henyey-Greenstein phase function with $g = 0.924$:

$$p_p(\theta,g) = \frac{1-g^2}{(1+g^2-2gcos\theta)^{3/2}} \tag{16}$$

It is very common in ocean waters to see a color shift ranging from greenish to very bright green, or even yellowish. These hue shifts are due to the variation in the concentration and type of the suspended microorganisms, mainly phytoplankton, which presents a maximum absorption at 350 nm. rapidly decreasing to almost zero beyond 500 nm. The

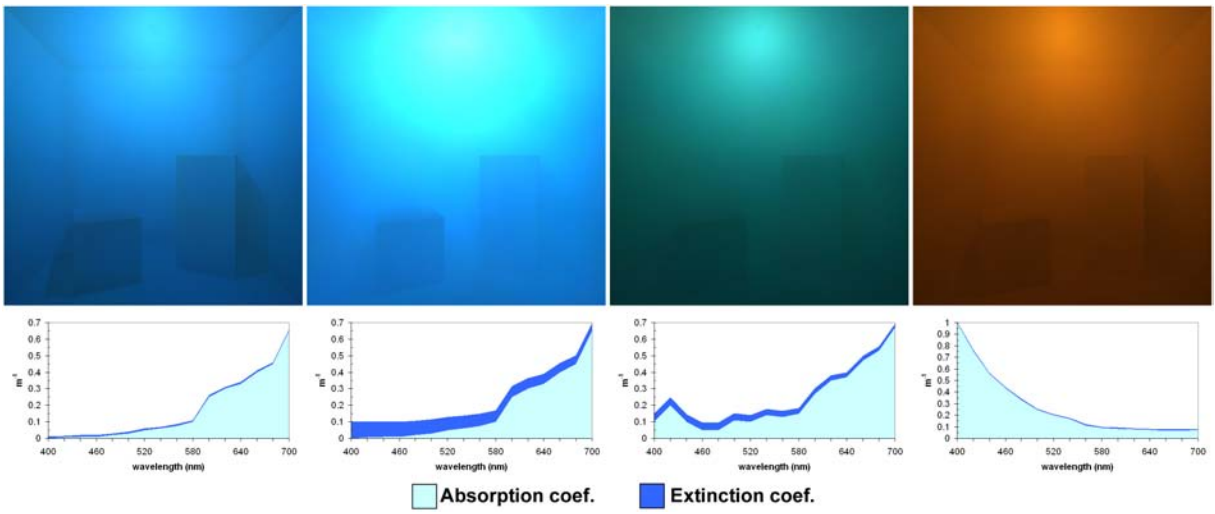



**Figure 3:** *Fluorescent ocean water in Cornell rooms. (a), (b) and (c) show varying concentrations of chlorophyll (*$0.05mg/m^3$, $0.1mg/m^3$ *and* $5mg/m^3$ *respectively). (d) High concentration of yellow matter (*$5mg/m^3$*).*

most important element in the phytoplankton is chlorophyll, which presents spectral absorption peaks in the blue and red ends of the spectrum and is the most important source of volume fluorescence in the waters. For chlorophyll, $\Gamma^c(\lambda_i)$ is wavelength-independent, with values ranging from 0.01 to 0.1 (we use the superscript $c$ for chlorophyll). As with most inelastic scattering event, the re-emission phase function is isotropic.

Another important source of fluorescence is the Color Dissolved Organic Matter (CDOM), also called yellow matter, present in shallow ocean waters and harbors. $\Gamma^y(\lambda_i)$ is also wavelength-independent, with values between 0.005 and 0.025, and re-emission is also isotropic [Haw92].

All the images in the paper have been rendered on a Beowulf system composed of six nodes, each one being a Pentium 4 @ 2.8 GHz. with 1 Gb. of RAM. Figure 3 shows different colorations of ocean water, according to varying chlorophyll and yellow matter concentrations which trigger inelastic scattering events with different probabilities. The images were rendered with 250,000 photons stored in the volume photon map and 200 photons used for the radiance estimate. This high numbers are needed to obtain accurate results, since we use the volume photon map to compute both direct and indirect illumination. Direct illumination in participating media with a varying index of refraction cannot be efficiently computed using ray tracing techniques, as explained at the beginning of section 3. The spectrum was sampled at nine intervals. Below each picture, the resulting absorption and extinction curves (functions of the different concentrations of chlorophyll in the modelled waters) are shown for each case. Image (a) shows little fluorescence (low chlorophyll concentration of $0.05mg/m^3$), and the waters are relatively clear. When chlorophyll concentration increases, fluorescence events become more prominent and the image first gets a milky aspect (b), losing visibility and reaching a characteristic green hue when chlorophyll reaches $5mg/m^3$. Image (d) shows fluorescence owed to yellow matter. The absorption function in this case has been modelled after [Mob94]: $a_y(\lambda) = a_y(440)^{-0.014(\lambda-440)}$ where $a_y(440)$ is the empirical absorption at 440 nm. Rendering times for the images were six minutes.

## 5. Case Study: Atmospheric Phenomena

The images in this section illustrate some of the most relevant effects in nature owed to curved light paths. To achieve physically correct results we have modelled the Earth as a sphere with a radius of 6371 units (one unit equals one kilometer); the atmosphere is another concentric sphere with a thickness of 40 kilometers. Taking the 1976 *USA Standard Atmosphere* (USA76) [USG76], we first obtain a standard temperature and pressure profile of the whole 40 kilometers, with temperature decreasing at an approximate rate of $0.6°$C per 100 meters. In order to curve light correctly according to Fermat's law, we need to obtain the wavelength-dependent index of refraction as a function of both the temperature and pressure given by the USA76. To do so, we follow the method described in [GSMA04], by first obtaining density as a function of temperature $T(h)$ and pressure $P(h)$ using the Perfect Gas law $\rho(h) = \frac{P(h)M}{RT(h)}$, where $M$ and $R$ are constants of values $28.93 \cdot 10^{-3}\, kg/mol$ and $8.3145\, J/mol \cdot K$ respectively. The Gladstone-Dale law [GD58] relates $n(\lambda, h)$ as a function of both $\rho(h)$ and $n(\lambda)$, given by the expression:

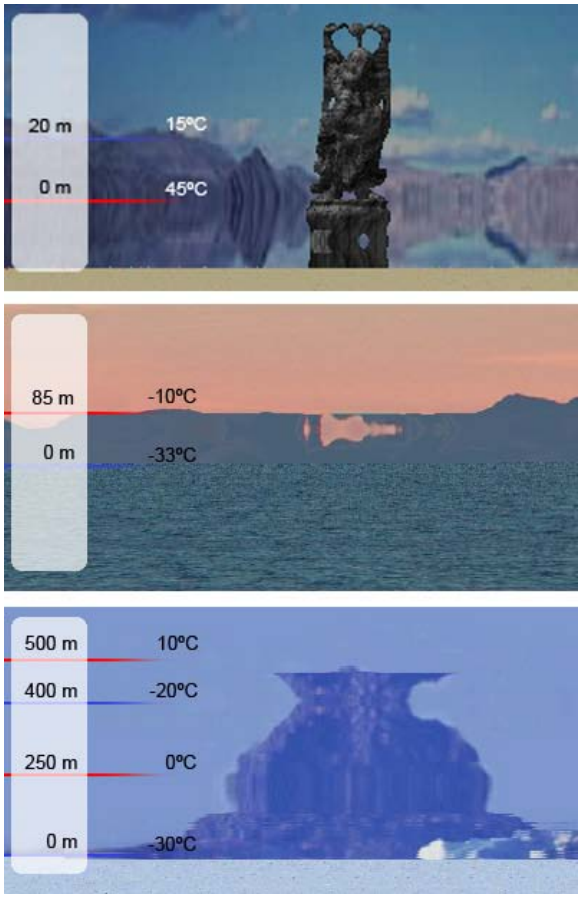



**Figure 4:** *Simulation of several atmospheric phenomena. Top: inferior mirage. Middle: superior mirage. Bottom: Fata Morgana.*

$$n(h,\lambda) = \rho(h) \cdot (n(\lambda) - 1) + 1 \tag{17}$$

The only missing function is now $n(\lambda)$, which we obtain from Cauchy's analytical formula [BW02]:

$$n(\lambda) = a \cdot \left(1 + \frac{b}{\lambda^2}\right) + 1 \tag{18}$$

where $a$ and $b$ depend on the medium considered (for air, their values are $a = 29.79 \cdot 10^{-5}$ and $b = 5.67 \cdot 10^{-5}$). Sellmeier [BW02] provides a slightly more elaborated formula, but we have chosen Cauchy's for efficiency reasons.

Combining equations 17 and 18 we finally obtain our profile for $n(\lambda,h)$, which we can alter at will to obtain the desired effects. To interpolate the complete, altered profiles for the whole 40 km. we use Fermi's distribution, as proposed in [VDWGL00].

The camera in the scenes is placed far from the mirages at a specific height for each effect to be seen (they can only appear if the observer's line of vision forms an angle less than one degree with the horizon). The error tolerance in the Dormand-Prince RK5(4)7M method has been set to $10^{-9}$, and the spectrum has been sampled in three wavelengths. Figure 4 (top) shows our simulation of an inferior mirage, which occurs when the ground is very hot and heats up the air layers right above it, thus creating a steep temperature gradient (30°C in 20 meters). As a consequence, light rays get bent upwards, and an inverted image of the Happy Buddha and the background appears on the ground. The camera is placed 10 meters above the ground. The image took 14 minutes to render.

Inversion layers are caused by an increase of air temperature with height, reversing the standard behavior where temperature decreases as a function of height. This happens most commonly above cold sea waters, and the light rays get bent downward, giving rise to the superior mirage. Figure 4 (middle) shows our simulation, modelling an inversion layer with a temperature gradient of 23°C. The apparent hole in the mountains is actually formed by the superior inverted image of the real mountains. The camera is placed also 10 meters above the ground, and the image took four minutes and 32 seconds to render. The great decrease in rendering time compared to the inferior mirage is owed to the simpler geometry of the scene, since the far away mountains are textured low-resolution objects.

Maybe less known than the two previous examples, the Fata Morgana occurs as a concatenation of both superior and inferior mirages, and is a much rarer phenomenon. Figure 4 (bottom) shows our simulation with two inversion layers with steep temperature gradients. There is an inferior mirage image across the middle of the mountain plus a superior mirage with the inverted image on top. The shape of the mountain gets greatly distorted; the Fata Morgana has historically tricked arctic expeditions, making them believe they were seeing huge mountains that were just a complicated pattern of upright and inverted images of the real, much lower hill (*Fata Morgana* is in fact the name of a fairly enchantress skilled in the art of changing shape, which she learnt from Merlin the Magician). The camera is placed at 300 meters (for the Fata to be visible it needs to be between the inversion layers), and the rendering time was five minutes.

## 6. Discussion

The method described has been implemented in Lucifer, our in-house global illumination renderer. It can handle multiple inelastic scattering in inhomogeneous participating media with a varying index of refraction, thus rendering effects such as mirages or fluorescence in ocean waters with full lighting computation. It deals well with strong anisotropy in the phase functions and the effects of backscattering, since no discretizations of the scene must be performed,

and thus the shortcoming of the only previous work on volume fluorescence [CS03] is overcome. It also supports real light sources, with photometric data input specified in the standard CIBSE TM14 format [CIB88]. This is a must for predictive rendering and for generating physically accurate data. The real light sources are sampled so that photons are emitted proportionally to the distribution of the light, given by its photometry.

Spectral images are calculated in high dynamic range, in order to obtain accurate data from the simulations. For tone reproduction purposes we map luminances to the display based on the work by Ward et al. [LRP97] and Pattanaik et al. [PTYG00]. To increase realism during the visualization of the images, an additional operator has been added which simulates the effects of chromatic adaptation in the human eye. This operator is specially important in the realistic depiction of underwater imagery, where the cones in the human eye might undergo a loss of spectral sensitivity after having been exposed to the same wavelength for a long period of time (underwater imagery being usually blue or green mostly). The complete description of such operator can be found in [GSMA04].

As stated in the introduction, the algorithm implemented is general and physically-based. This allows us to use the radiometric and photometric data obtained from the simulations for any purpose other than rendering, such as professional architectural lighting or accurate simulations of deep underwater vision, given the exact description of the luminaire to be used and the water conditions. This accuracy obviously increases rendering times compared to faked, ad-hoc solutions. To improve efficiency, we impose an early light path termination and an adaptive integration step while solving Fermat's law. Choosing the Dormand-Prince RK5(4)7M numerical method over the more standard Euler method has produced speedups of up to 106.4. We have also used a parallel implementation on a six-PC Beowulf system of our non-linear photon mapping algorithm, achieving additional speedups between 4.2 and 4.8.

The non-linear photon mapping implementation allows us to extend several sunset effects similar to the ones simulated in [GSMA04], by including a thin layer of fog between the observer and the sun. The solar disk gets distorted into different shapes, while light is scattered through the layer of fog, thus achieving a "winter sunset" look (figure 5, left and middle). Figure 5 right shows volume caustics generated by a crystal sphere in a fluorescent medium.

Figure 6 shows several renders obtained with Lucifer. All of them are lit by a Philips SW-type© luminaire, specified according to the CIBSE TM14 format. The only light source is immersed in the medium, so no caustics from the interaction of sunlight with the surface appear. The medium modelled does not emit light, although adding that to the model is straightforward and would allow us to simulate effects such as bioluminiscence in the water. Fluorescence

**Figure 5:** *Sunset effects through a layer of fog. Left: flattened sun. Middle: split sun. Right: Volume caustics in a fluorescent medium.*

owed to inelastic scattering is computed according to the varying concentrations of chlorophyll in each image (between 0.01 and $0.1 mg/m^3$). The volume photon map in all the images contains 500.000 photons, and the radiance estimate used 250. Again, these high numbers are needed since we compute direct lighting with the photon map. The top two images represent a sunken boat along a Happy Buddha in clear, shallow waters (left) or deep underwater with a chlorophyll concentration of $0.05 mg/m^3$(right). For the bottom-left image, we have added a volume temperature field that simulates a heat source outside the image as explained in [SGGC05], deriving the index of refraction using the formula $n = 1 + \frac{T_o}{T}(n_o - 1)$ as proposed by Stam and Languenou [SL96]. The distortions caused by the varying index of refraction are visible, similar to the characteristic rippling in a real desert scene. The bottom-middle image uses a smoke-like medium, modelled as a 3D turbulence function, whereas the last to the right shows the effects of a highly anisotropic medium. The images are 400 pixels wide and took between 30 and 40 minutes to render, without any penalty imposed by the anisotropy in the last image.

## 7. Conclusion and Future Work

We proposed a novel extension of the widely used photon mapping technique, which accounts for multiple inelastic scattering and can provide a full global illumination solution in inhomogeneous media with a varying index of refraction, where light paths are bent. No pre-lit textures are needed in this case, since both direct and indirect lighting is calculated from the photon map. The method is physically-based and yields accurate high-dynamic results that can either be output as an image to a display device (via tone mapping), or used in other fields as raw data. Inelastic scattering is calculated during the photon tracing stage, so the extra cost required is just a second Russian roulette per absorption. The accompanying video shows the feasibility of the approach for animations.

Practically all inelastic scattering effects in the visible range of the spectrum mean a transfer of energy from shorter to longer wavelengths. Nevertheless, the algorithm presented in this work can handle rarer inelastic scattering events where energy gets transferred from longer to shorter wave-

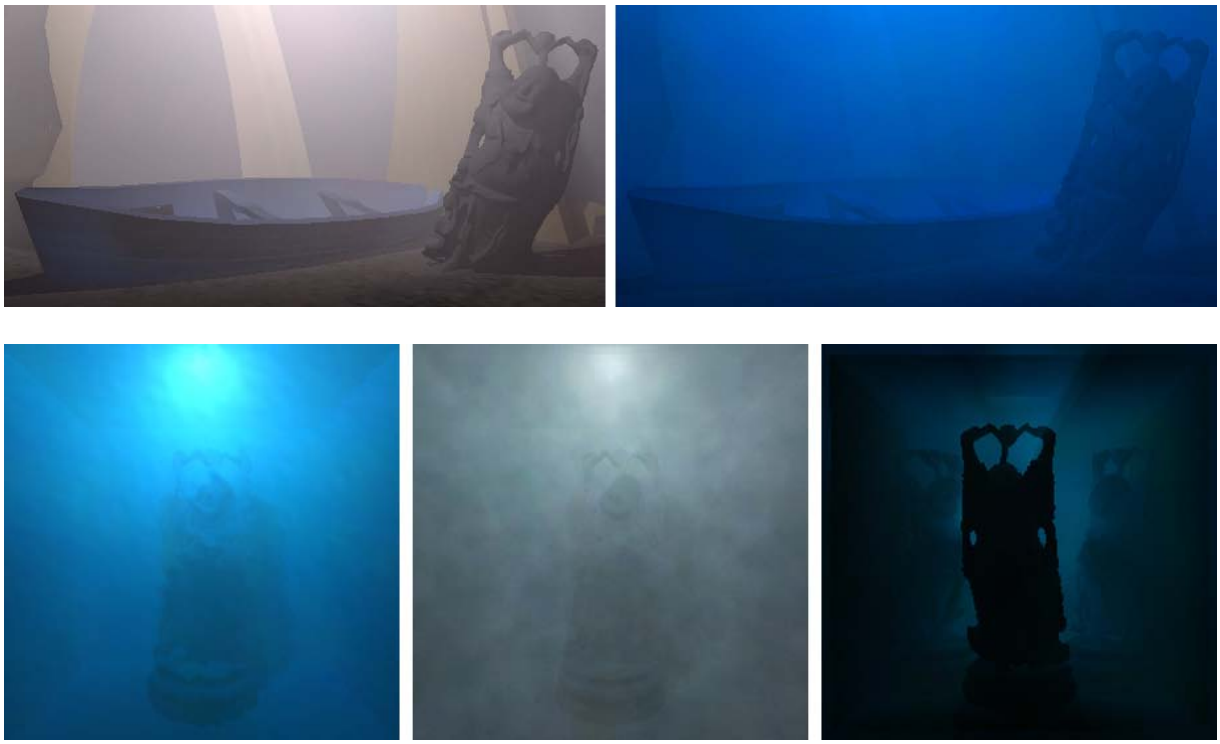


**Figure 6:** *Different images with inelastic scattering in participating media. Top left: very low chlorophyll concentration. Top right: higher concentration yields more inelastic scattering events. Bottom left: distortions caused by a 3D temperature field. Bottom middle: 3D turbulence field simulating smoke. Bottom right: highly anisotropic medium.*

lengths (such as a fraction of the Raman scattering that occurs naturally in several solids, liquids and gases [Mob94]), since it does not follow a cascade, one-way scheme from the blue end to the red end of the spectrum. The application of these type of inelastic scattering to computer graphics is probably just marginal, but the data generated can be very useful to physicists or oceanographers. Adding phosphorescence effects could make use of the work by Cammarano and Wann Jensen [CJ02], although a more straightforward approach would be to use the decay function $d(t)$ in each frame. Any number of light sources can be used in the scene, even with different photometric descriptions.

The bottleneck of the algorithm is solving the paths for each photon and eye-ray using Fermat's law. Although the use of a Dormand-Prince method has drastically reduced rendering times by two orders of magnitude, additional work needs to be done to achieve near real-time frame rates. Importance maps could be used for this purpose, although two other promising fields of research lay ahead: the first one is the implementation of the algorithm on GPUs, as proposed by Purcell et al. [PDC*03]. The second would try to take advantage of temporal coherence of light distribution, as presented by Myszkowski et al. [MTAS01].

## 8. Acknowledgements

This research was partly done under the sponsorship of the Spanish Ministry of Education and Research through the project TIN2004-07672-C03-03. The authors would also like to thank Eduardo Jiménez for his initial contribution to this work.

## References

[BTL90] BERGER M., TROUT T., LEVIT N.: Ray tracing mirages. *IEEE Computer Graphics and Applications 10*, 3 (May 1990), 36–41.

[BW02] BORN M., WOLF E.: *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 2002.

[CIB88] CIBSE: *Standard File Format for Transfer of Luminaire Photometric Data*. The Chartered Institution of Building Services Engineers, 1988.

[CJ02] CAMMARANO M., JENSEN H. W.: Time dependent photon mapping. In *Proceedings of the 13th Eurographics workshop on Rendering* (2002), Eurographics Association, pp. 135–144.

[CS03] CEREZO E., SERON F.: Inelastic scattering in participating media. application to the ocean. In *Proceedings of the Annual Conference of the European Association for Computer Graphics, Eurographics 2003* (2003), pp. CD–ROM.

[DP80] DORMAND J., PRINCE P.: A family of embeded runge-kutta formulae. *Journal of Computational and Applied Mathematics 6(1)* (1980), 19–26.

[FFLV82] FABRI E., FIORZO G., LAZZERI F., VIOLINO P.: Mirage in the laboratory. *Am. J. Physics 50(6)* (1982), 517–521.

[GD58] GLADSTONE J. H., DALE J.: On the influence of temperature on the refraction of light. *Phil. Trans. 148* (1858), 887.

[Gla95a] GLASSNER A.: *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, California, 1995.

[Gla95b] GLASSNER A. S.: A model for fluorescence and phosphorescence. In *Photorealistic Rendering Techniques* (1995), Sakas P. S. G., Müller S., (Eds.), Eurographics, Springer-Verlag Berlin Heidelberg New York, pp. 60–70.

[Grö95] GRÖLLER E.: Nonlinear ray tracing: visualizing strange worlds. *The Visual Computer 11*, 5 (1995), 263–276.

[GSMA04] GUTIERREZ D., SERON F., MUNOZ A., ANSON O.: Chasing the green flash: a global illumination solution for inhomogeneous media. In *Spring Conference on Computer Graphics* (2004), (in cooperation with ACM SIGGRAPH A. P., Eurographics), (Eds.), pp. 95–103.

[Haw92] HAWES S.: *Quantum Fluorescence Efficiencies of Marine Fulvic and Humid Acids*. PhD thesis, Dept. of Marince Science, Univ. of South Florida, 1992.

[HW01] HANSON A. J., WEISKOPF D.: Visualizing relativity. siggraph 2001 course 15, 2001.

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH 98 Conference Proceedings* (jul 1998), Cohen M., (Ed.), Annual Conference Series, ACM SIGGRAPH, Addison Wesley, pp. 311–320. ISBN 0-89791-999-8.

[LRP97] LARSON G. W., RUSHMEIER H., PIATKO C.: A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics 3*, 4 (Oct. 1997), 291–306.

[Mob94] MOBLEY C.: *Light and Water. Radiative Transfer in Natural Waters*. Academic Press, Inc., 1994.

[MTAS01] MYSZKOWKSI K., TAWARA T., AKAMINE A., SEIDEL H. P.: Perception-guided global illumination solution for animation. In *Computer Graphics Proceedings, Annual Conference Series, 2001 (ACM SIGGRAPH 2001 Proceedings)* (Aug. 2001), pp. 221–230.

[Mus90] MUSGRAVE F. K.: A note on ray tracing mirages. *IEEE Computer Graphics and Applications 10*, 6 (Nov. 1990), 10–12.

[Nem93] NEMIROFF R. J.: Visual distortions near a neutron star and black hole. *American Journal of Physics 61(7)* (1993), 619–632.

[PDC*03] PURCELL T. J., DONNER C., CAMMARANO M., JENSEN J., HANRAHAN P.: Photon mapping on programmable graphics hardware. In *SIGGRAPH/Eurographics Workshop on Graphics Hardware* (2003), Eurographics Association, pp. 041–050.

[PPS97] PEREZ F., PUEYO X., SILLION F.: Global illumination techniques for the simulation of participating media. In *Proc. of the Eigth Eurographics Workshop on Rendering* (1997), pp. 16–18.

[PTYG00] PATTANAIK S., TUMBLIN J. E., YEE H., GREENBERG. D. P.: Time-dependent visual adaptation for realistic image display. In *SIGGRAPH 2000, Computer Graphics Proceedings* (2000), Akeley K., (Ed.), Annual Conference Series, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 47–54.

[SGGC05] SERON F., GUTIERREZ D., GUTIERREZ G., CEREZO E.: Implementation of a method of curved ray tracing for inhomogeneous atmospheres. *Computers and Graphics 29(1)* (2005).

[SL96] STAM J., LANGUÉNOU E.: Ray tracing in non-constant media. In *Eurographics Rendering Workshop 1996* (New York City, NY, June 1996), Pueyo X., Schröder P., (Eds.), Eurographics, Springer Wien, pp. 225–234. ISBN 3-211-82883-4.

[USG76] USGPC: *U.S. Standard Atmosphere*. United State Government Printing Office, Washington, D.C., 1976.

[VDWGL00] VAN DER WERF S., GUNTHER G., LEHN W.: Novaya zemlya effects and sunsets. *Applied Optics 42*, 3 (2000).

[WTP01] WILKIE A., TOBLER R., PURGATHOFER W.: Combined rendering of polarization and fluorescence effects. In *Rendering Techniques '01 (Proc. Eurographics Workshop on Rendering 2001)* (2001), Gortler S.J. M. K. e., (Ed.), Eurographics, Springer-Verlag, pp. 197–204.

[YOH00] YNGVE G. D., O'BRIEN J. F., HODGINS H.: Animating explosions. In *Proceedings of the Computer Graphics Conference 2000 (SIGGRAPH-00)* (New York, July 23–28 2000), Hoffmeyer S., (Ed.), ACMPress, pp. 29–36.