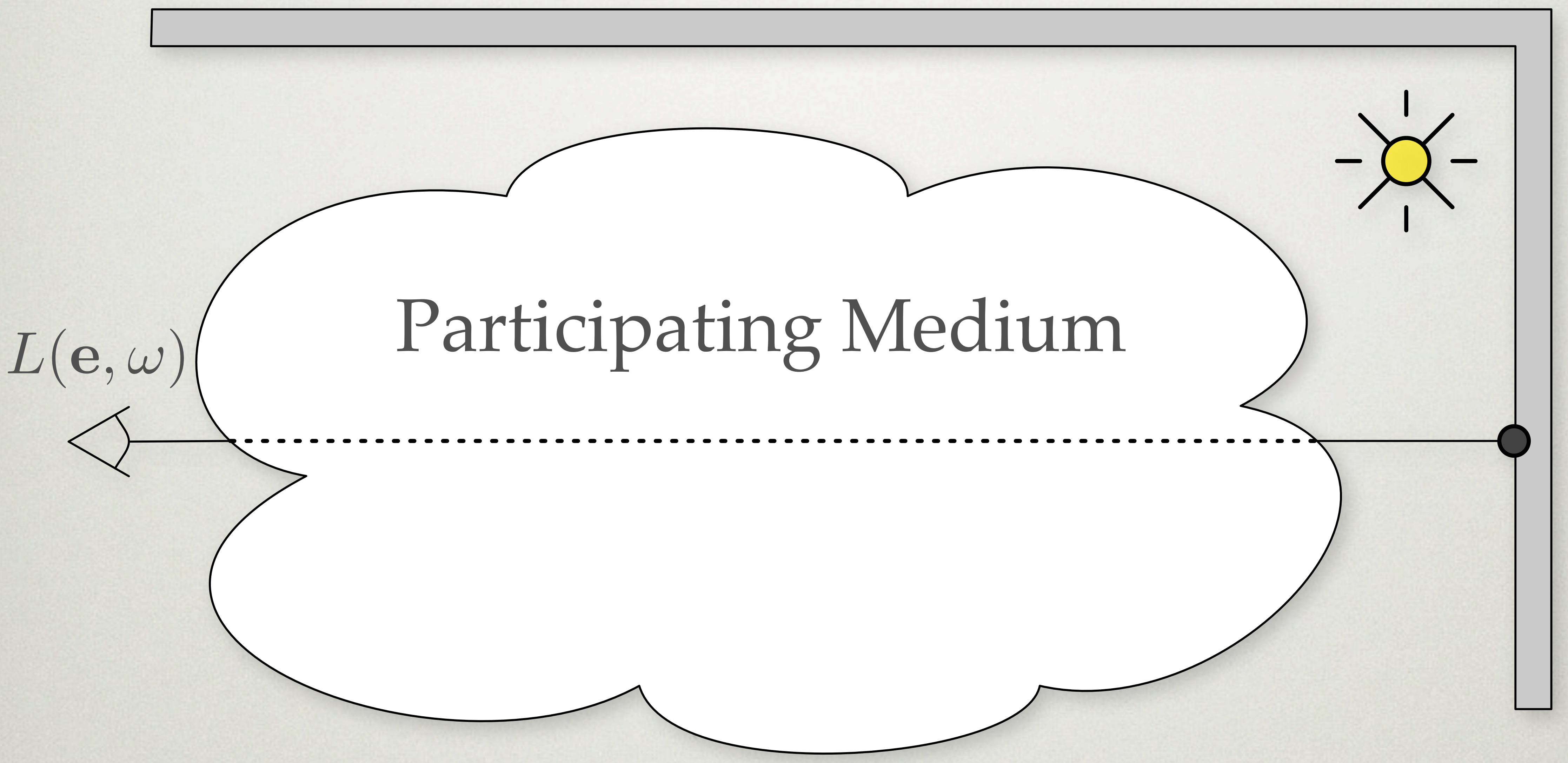
Radiance Caching for Participating Media

Wojciech Jarosz   Craig Donner
Matthias Zwicker   Henrik Wann Jensen

University of California, San Diego
Pixel Lab

* In this talk, we are interested in rendering scene with participating media, or scenes where the volume or medium participates in the lighting interactions.
* Participating media is actually all around us.
* These are just a few example photographs of the type of striking effects that are caused by participating media.

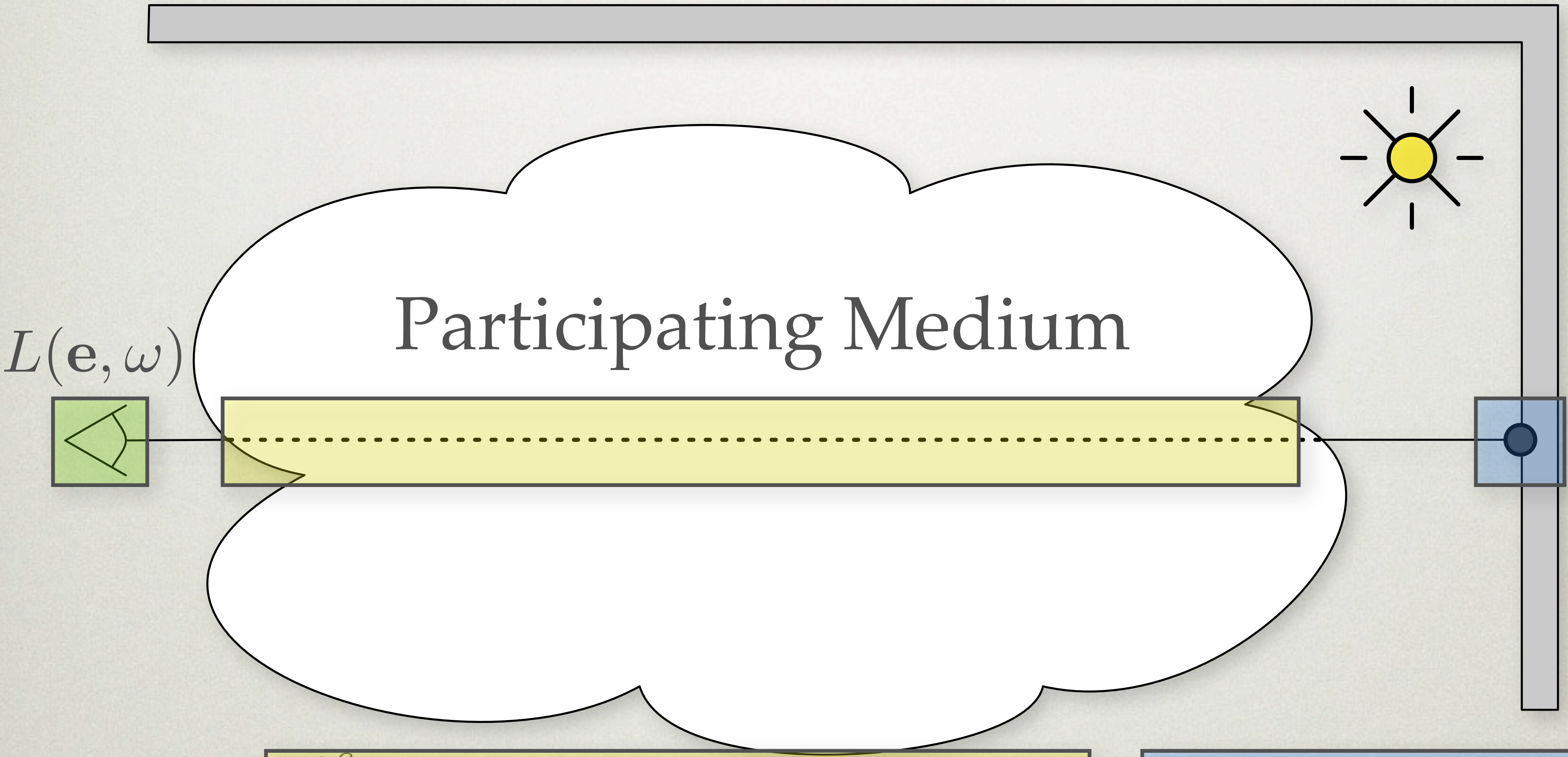
Volume Rendering Eq.



Thursday, 6 September 12

* In order to render images:
* We need to compute the radiance, L, arriving at the eye along a ray in the presence of participating media.
* This can be expressed using the volume rendering equation, which consists of two main terms:
    * The right term incorporates lighting arriving from surfaces
    * and the left term, scattering of light from the medium

Volume Rendering Eq.

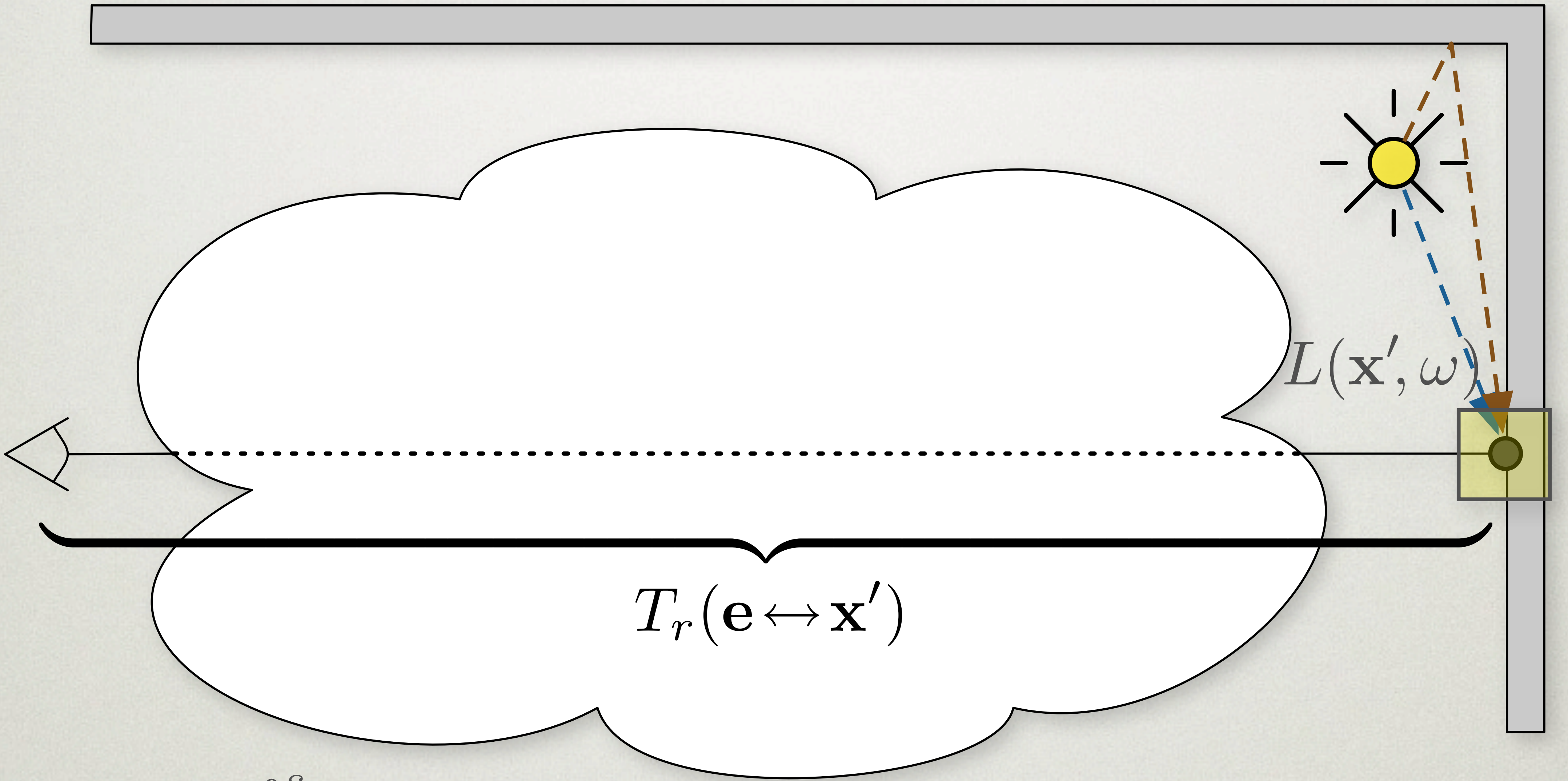


$$L(\mathbf{e},\vec{\omega}) = \int_0^s T_r(\mathbf{e}\leftrightarrow\mathbf{x})\sigma_s(\mathbf{x})L_i(\mathbf{x},\vec{\omega})\,d\mathbf{x} + T_r(\mathbf{e}\leftrightarrow\mathbf{x}')L(\mathbf{x}',\vec{\omega})$$

Thursday, 6 September 12

* In order to render images:
* We need to compute the radiance, L, arriving at the eye along a ray in the presence of participating media.
* This can be expressed using the volume rendering equation, which consists of two main terms:
    * The right term incorporates lighting arriving from surfaces
    * and the left term, scattering of light from the medium

Volume Rendering Eq.

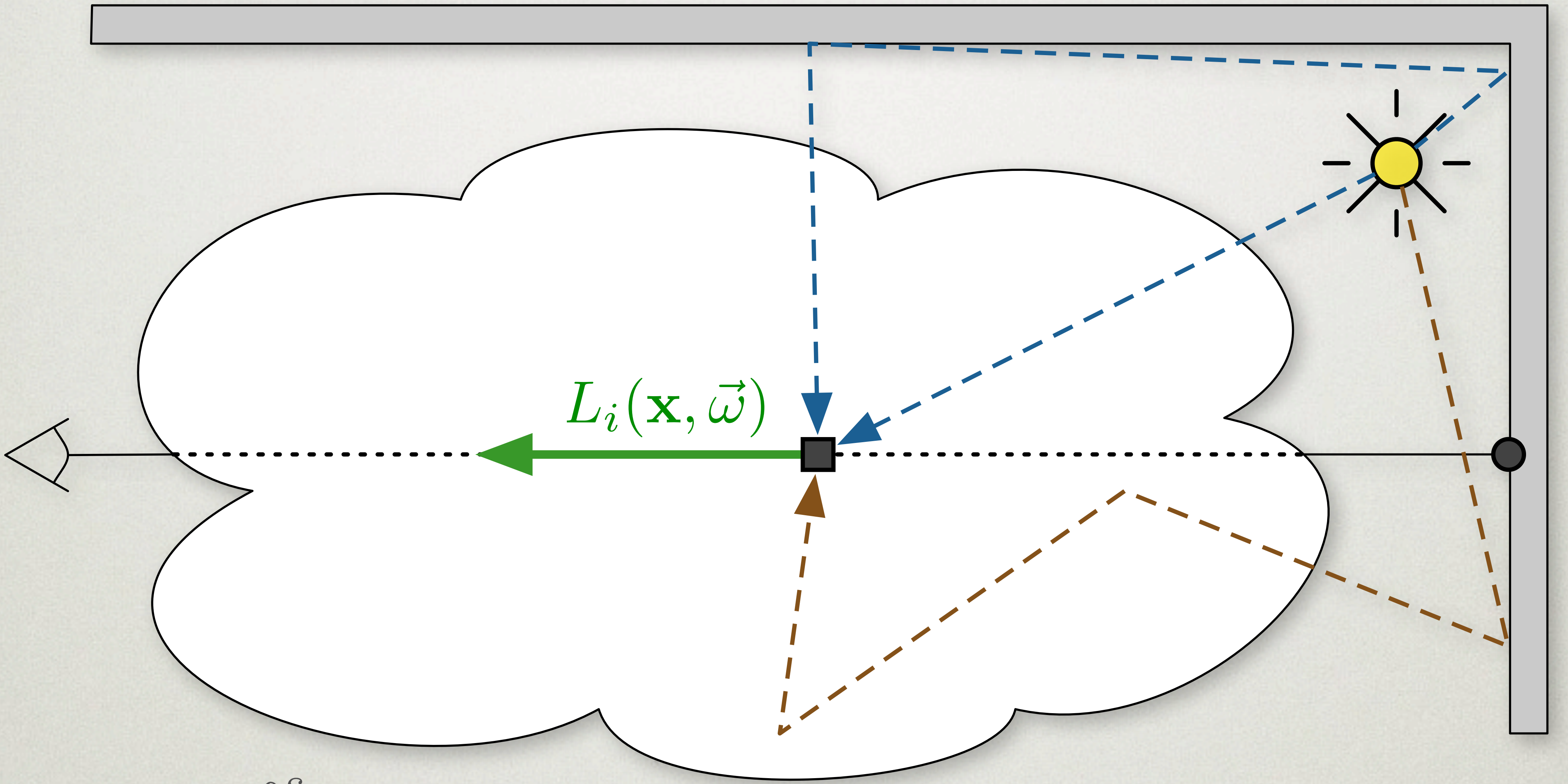


$$L(\mathbf{e}, \vec{\omega}) = \int_0^s T_r(\mathbf{e} \leftrightarrow \mathbf{x}) \sigma_s(\mathbf{x}) L_i(\mathbf{x}, \vec{\omega}) \, d\mathbf{x} + T_r(\mathbf{e} \leftrightarrow \mathbf{x}') L(\mathbf{x}', \vec{\omega})$$

$$\text{Transmittance: } T_r(\mathbf{e} \leftrightarrow \mathbf{x}') = e^{-\int_{\mathbf{e}}^{\mathbf{x}} \sigma(t) \, dt}$$

* This light is then diminished by the transmittance as it travels through the medium towards the eye

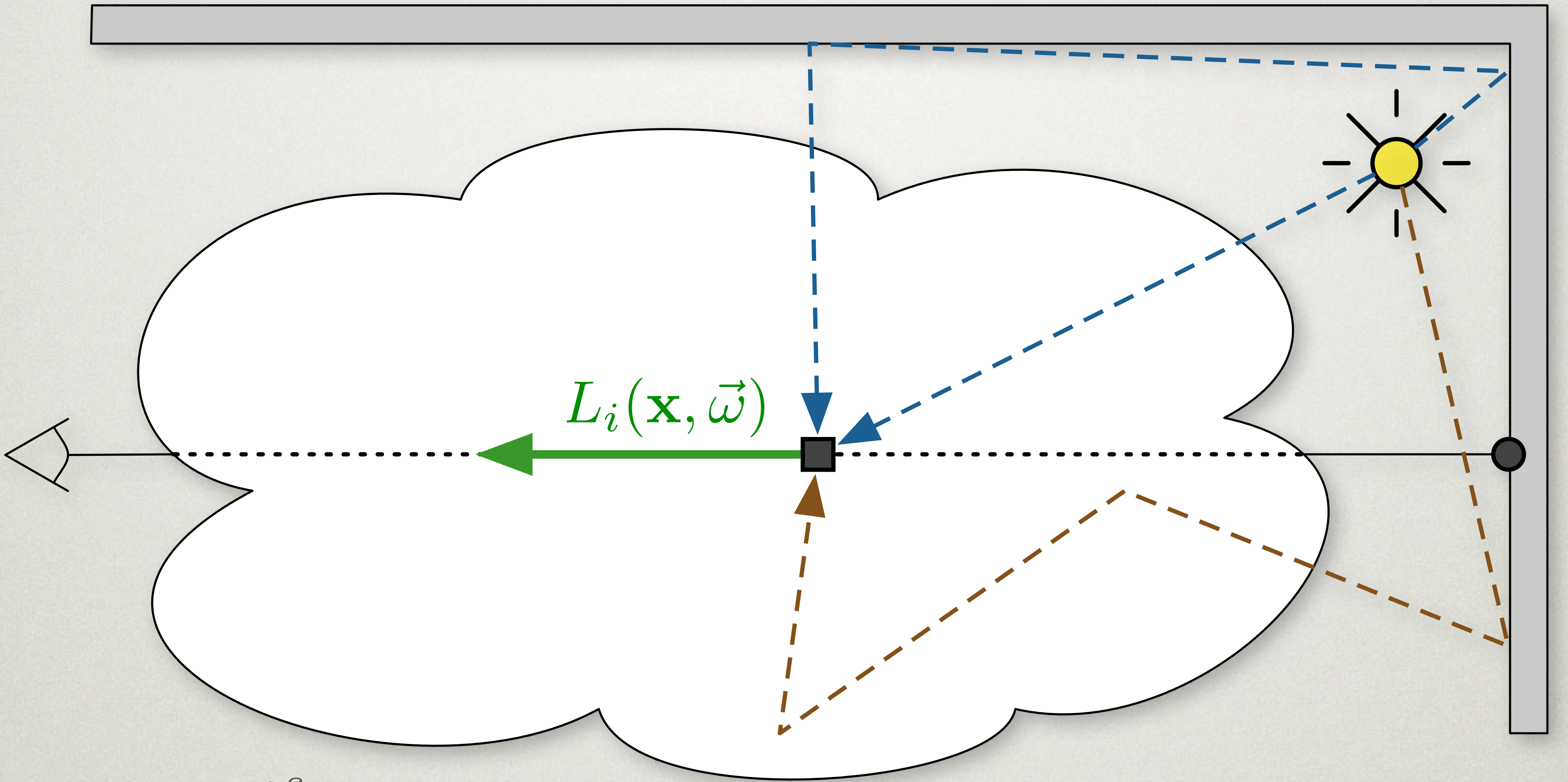
Volume Rendering Eq.



$$L(\mathbf{e}, \vec{\omega}) = \int_0^s T_r(\mathbf{e} \leftrightarrow \mathbf{x}) \sigma_s(\mathbf{x}) L_i(\mathbf{x}, \vec{\omega}) \, d\mathbf{x} + T_r(\mathbf{e} \leftrightarrow \mathbf{x}') L(\mathbf{x}', \vec{\omega})$$

all light arriving at **x** which scatters towards **e**

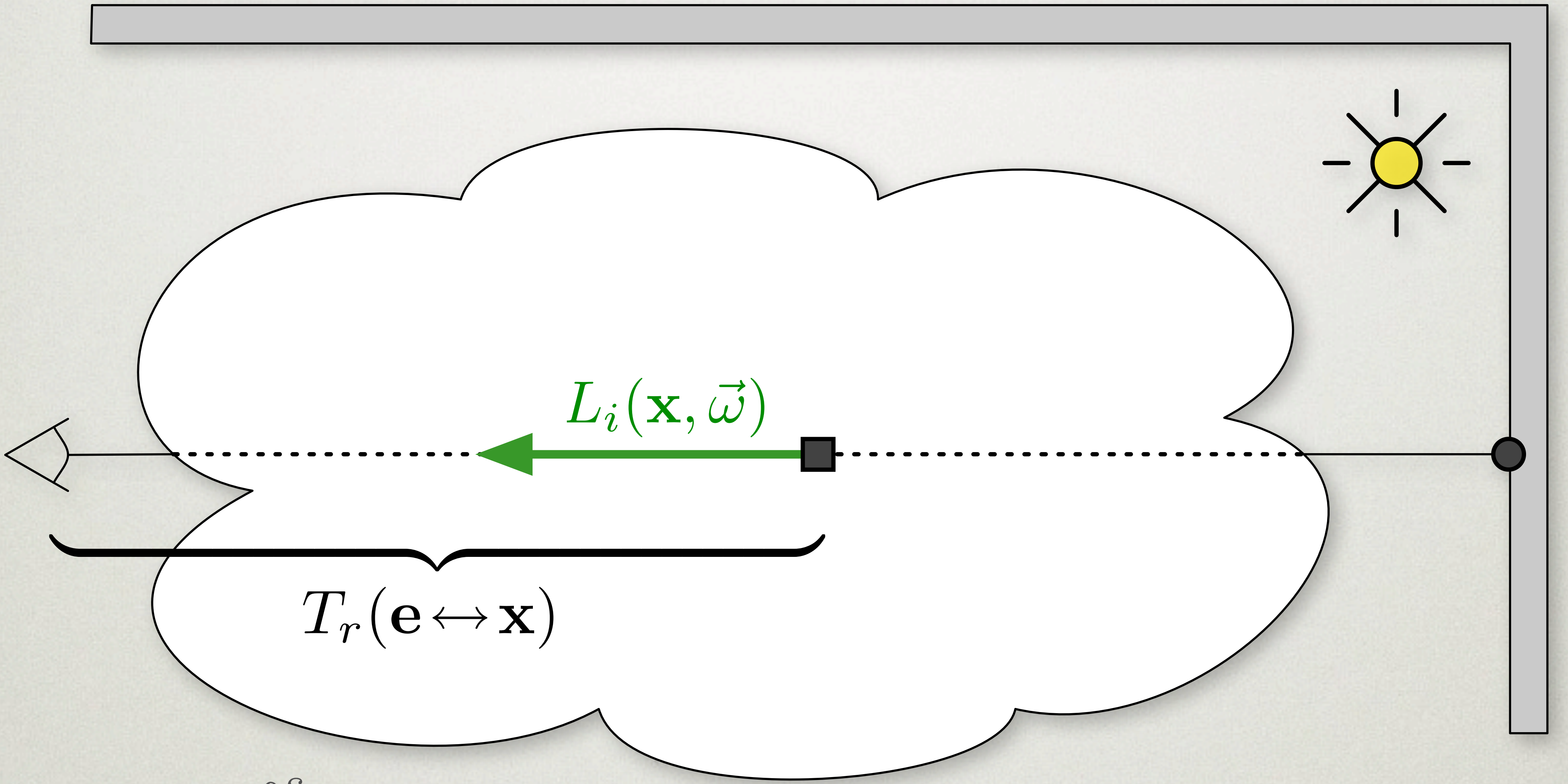
Volume Rendering Eq.



$$L(\mathbf{e}, \vec{\omega}) = \int_0^s T_r(\mathbf{e} \leftrightarrow \mathbf{x}) \sigma_s(\mathbf{x}) L_i(\mathbf{x}, \vec{\omega})\, d\mathbf{x} + T_r(\mathbf{e} \leftrightarrow \mathbf{x}') L(\mathbf{x}', \vec{\omega})$$

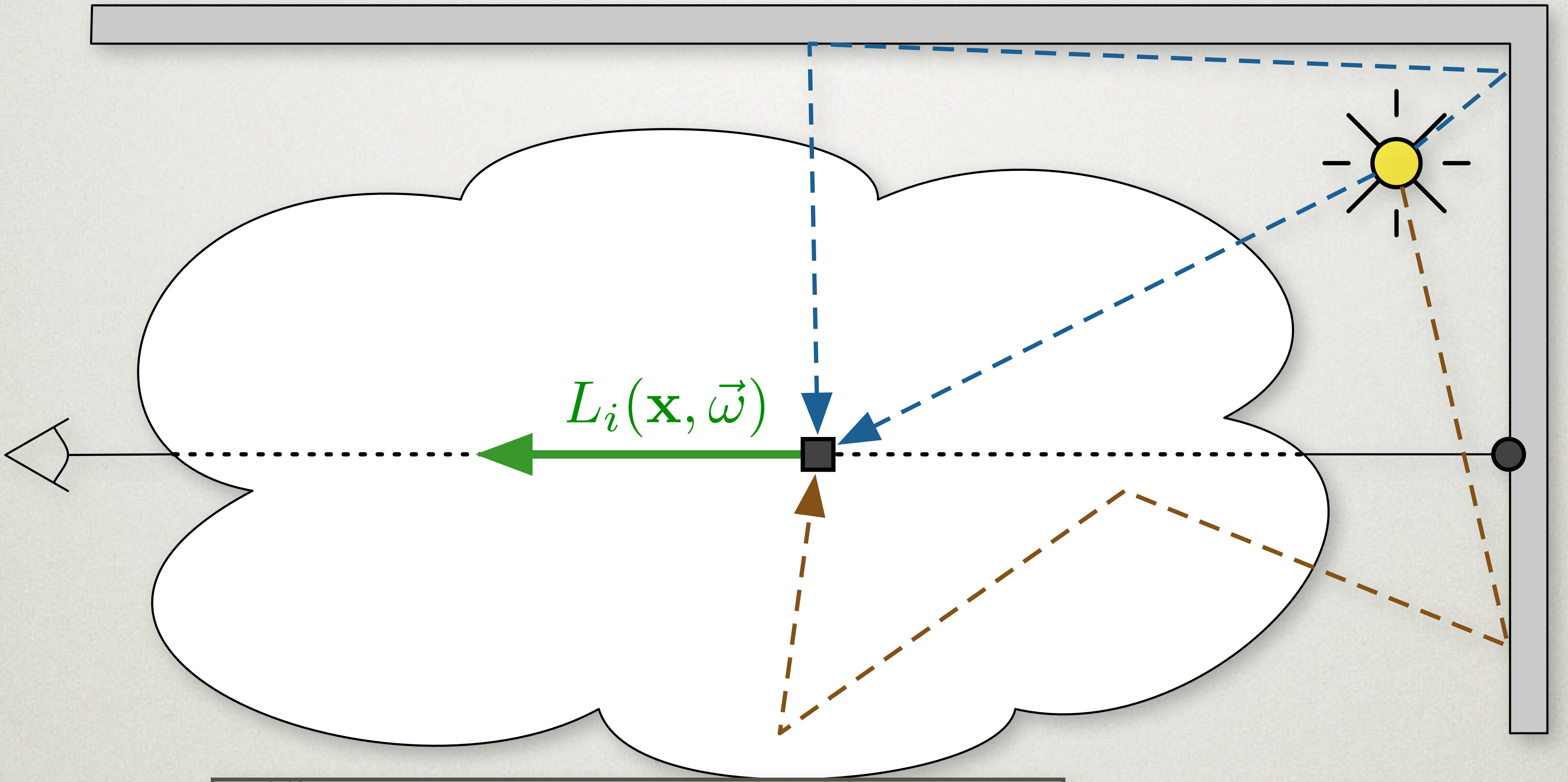
scattering coefficient

Volume Rendering Eq.



$$L(\mathbf{e}, \vec{\omega}) = \int_0^s T_r(\mathbf{e} \leftrightarrow \mathbf{x}) \sigma_s(\mathbf{x}) L_i(\mathbf{x}, \vec{\omega}) \, d\mathbf{x} + T_r(\mathbf{e} \leftrightarrow \mathbf{x}') L(\mathbf{x}', \vec{\omega})$$

Volume Rendering Eq.



$$L(\mathbf{e}, \vec{\omega}) = \int_0^s T_r(\mathbf{e} \leftrightarrow \mathbf{x}) \sigma_s(\mathbf{x}) L_i(\mathbf{x}, \vec{\omega}) \, d\mathbf{x} + T_r(\mathbf{e} \leftrightarrow \mathbf{x}') L(\mathbf{x}', \vec{\omega})$$

Very costly!

PREVIOUS WORK

Participating Media

Path tracing

[Kajiya and Herzen 84, Kajiya 86, Lafortune and Willems 96]

- Slow convergence/noisy results.

Photon mapping

[Jensen and Christensen 1998.]

- Costly for high albedo
- Costly for scenes with large extent

Finite Element

[Rushmeier and Torrance 87]

- Requires discretization

* A number of methods have been developed to handle participating media, but they all have significant limitations.
* this motivates us to develop a new method.

Related Work

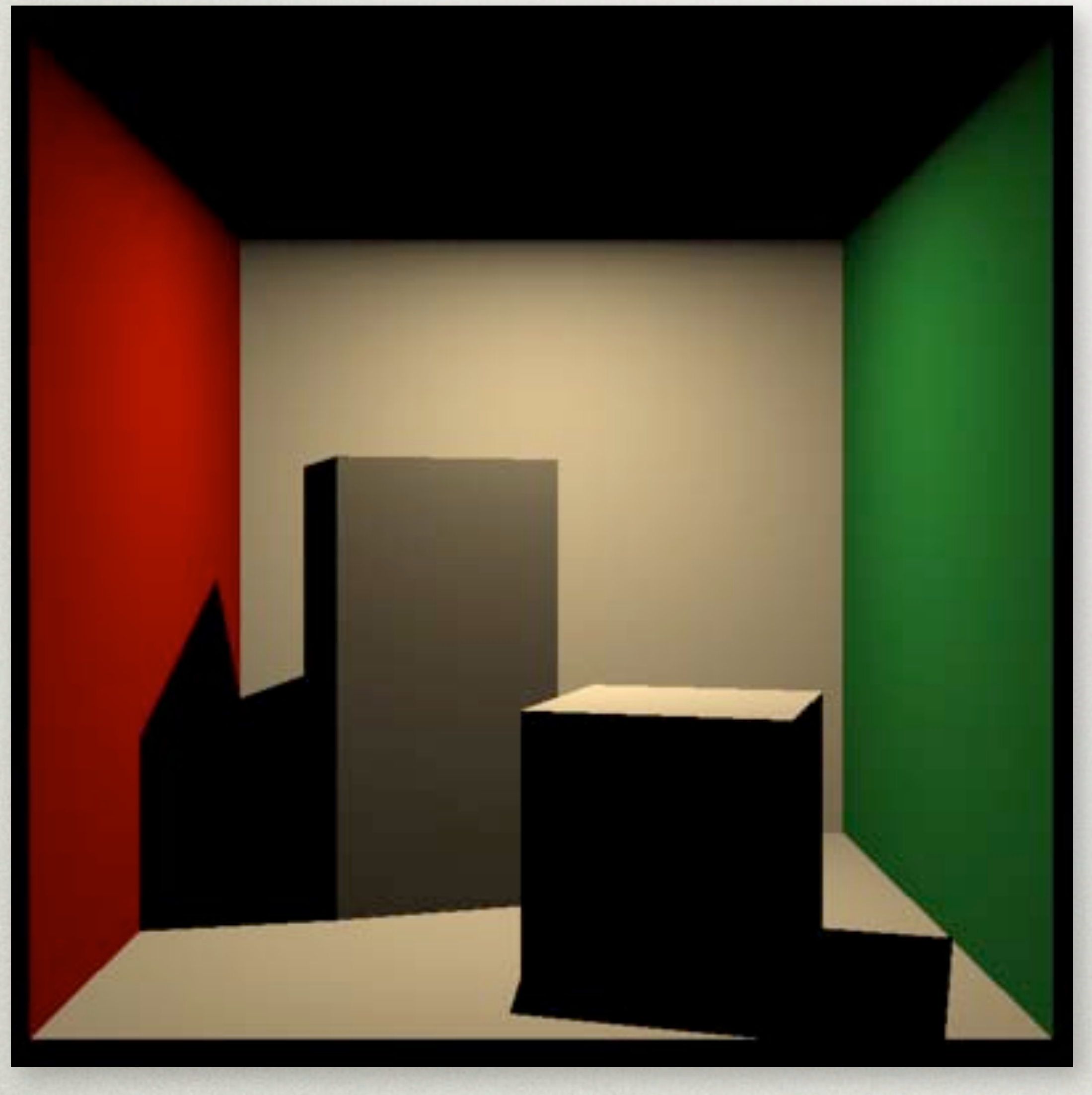
Global Illumination

Caching:

- "A Ray Tracing Solution for Diffuse Interreflection." Ward et al. 1988.
- "Irradiance Gradients." Ward and Heckbert. 1992.
- "Radiance Caching for Efficient Global Illumination Computation." Křivánek et al. '05

* we draw inspiration for our method from irradiance caching methods for surfaces.

Indirect Illumination



Direct Illumination

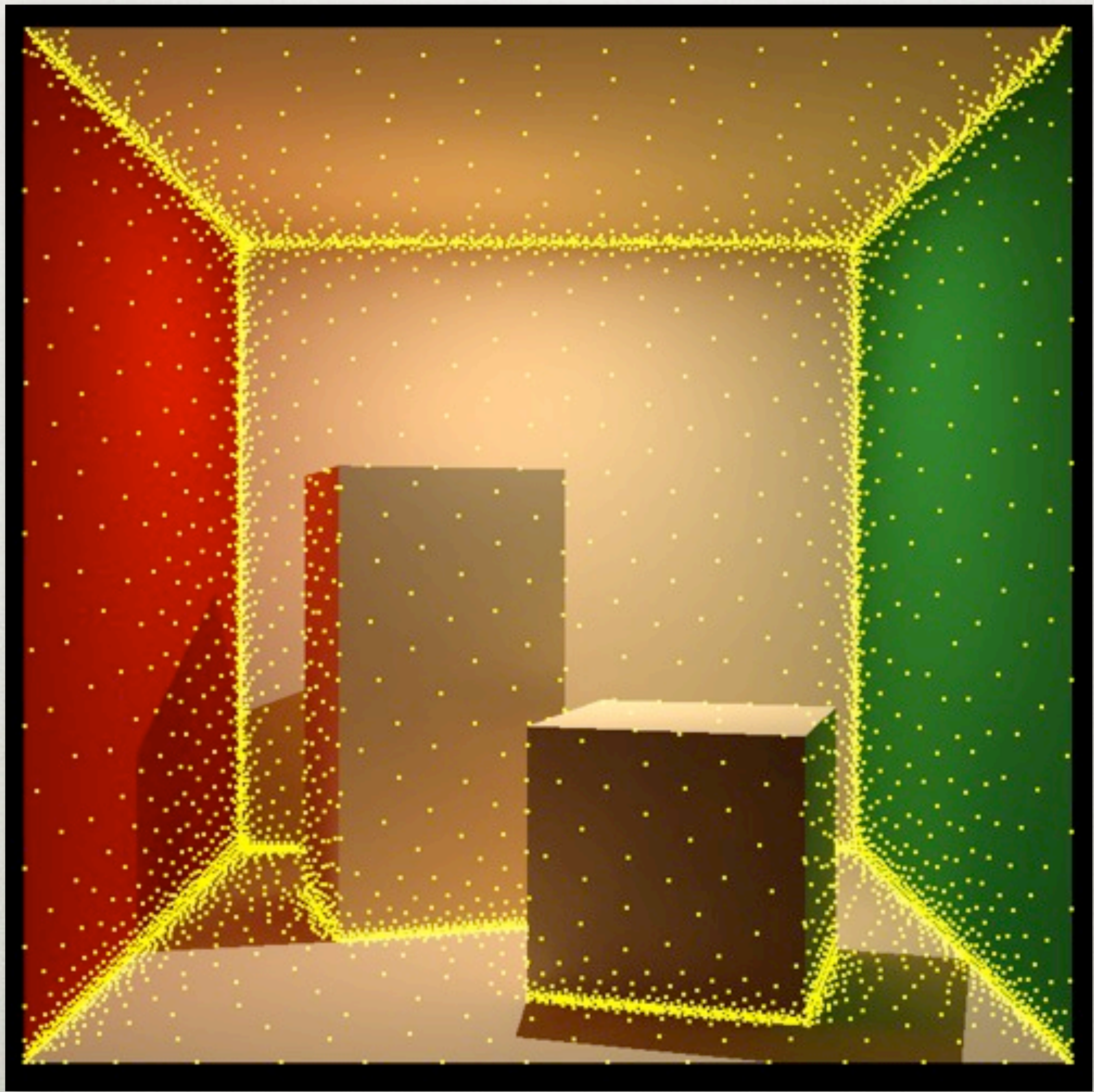
Indirect Illumination

Thursday, 6 September 12

* direct illumination has sharp discontinuities
* Indirect illumination smooth in large regions

Irradiance Caching

Ward et al. '88

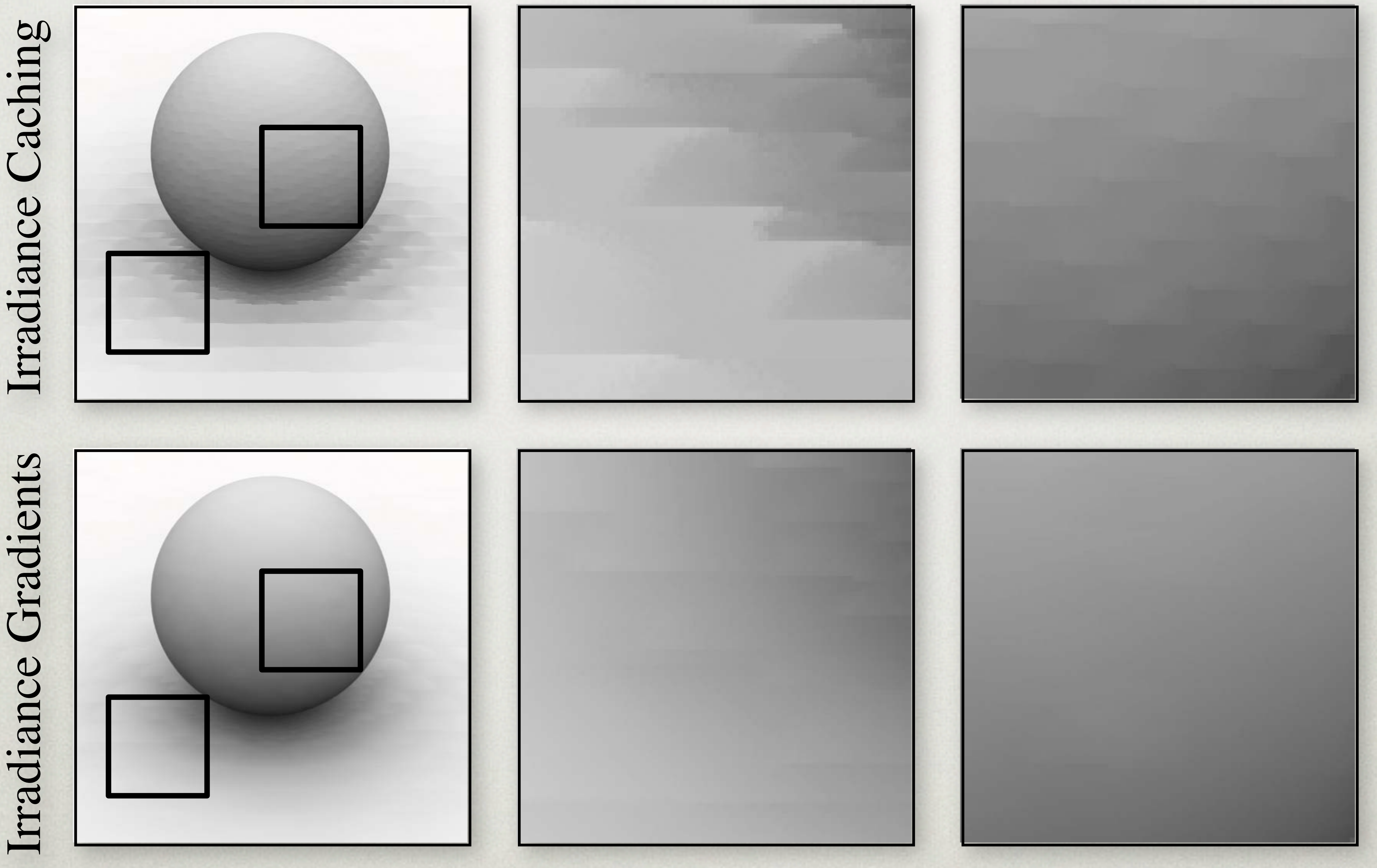


Thursday, 6 September 12

* compute irradiance accurately only at a sparse set of locations (shown in yellow) and interpolate whenever possible.

Irradiance Gradients

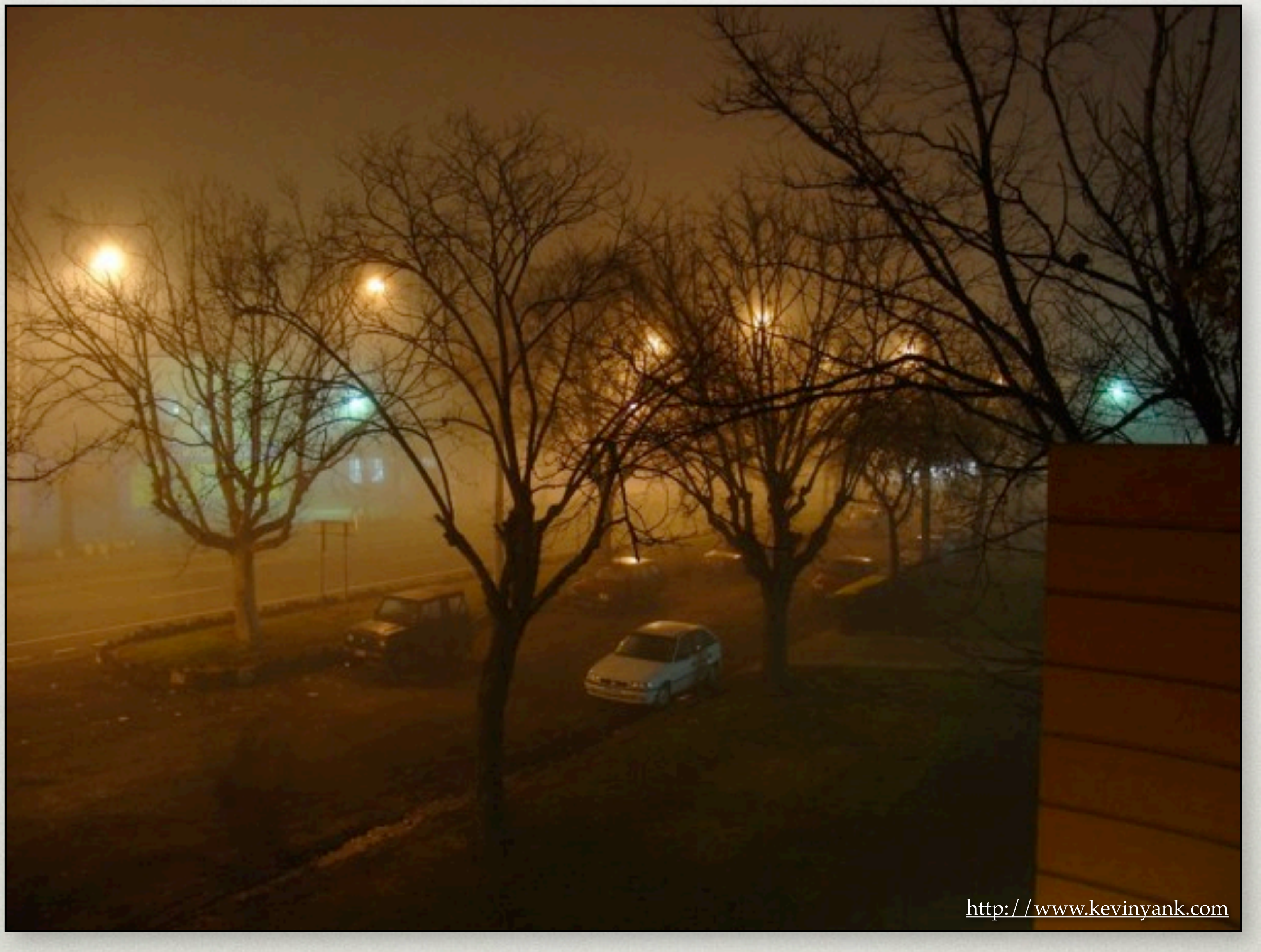
Ward and Heckbert '92



Thursday, 6 September 12

* follow-up work

Observations



Smooth in large portions of the image

Thursday, 6 September 12

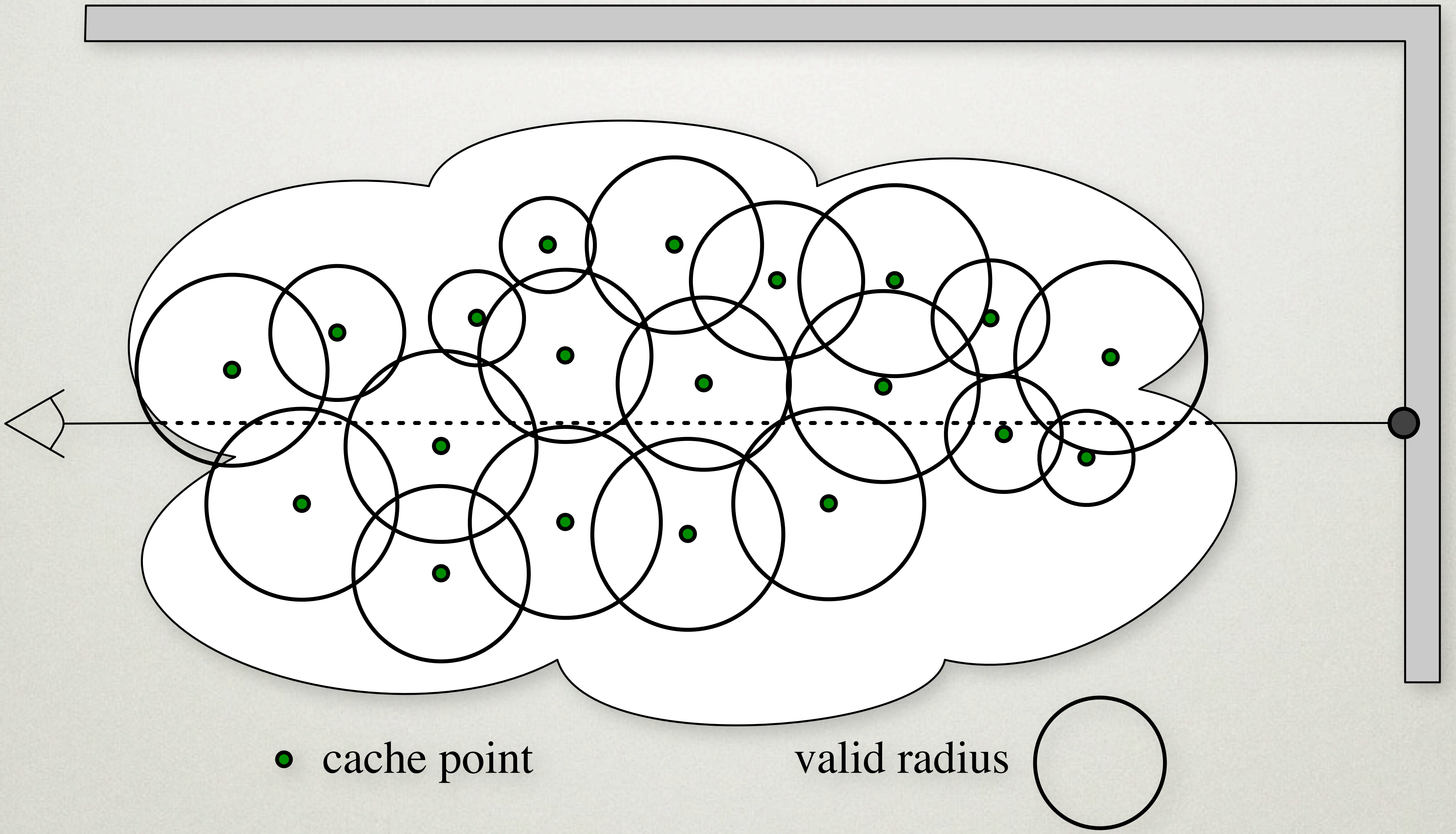
* We make the observation that same property is true for participating media
* computationally very expensive, but very smooth and low frequency in large parts of the image

Goals

- Exploit this property by caching lighting within participating media.
- Develop an efficient but general rendering algorithm which can handle:
  - single, multiple, anisotropic scattering
  - heterogeneous media
  - production quality

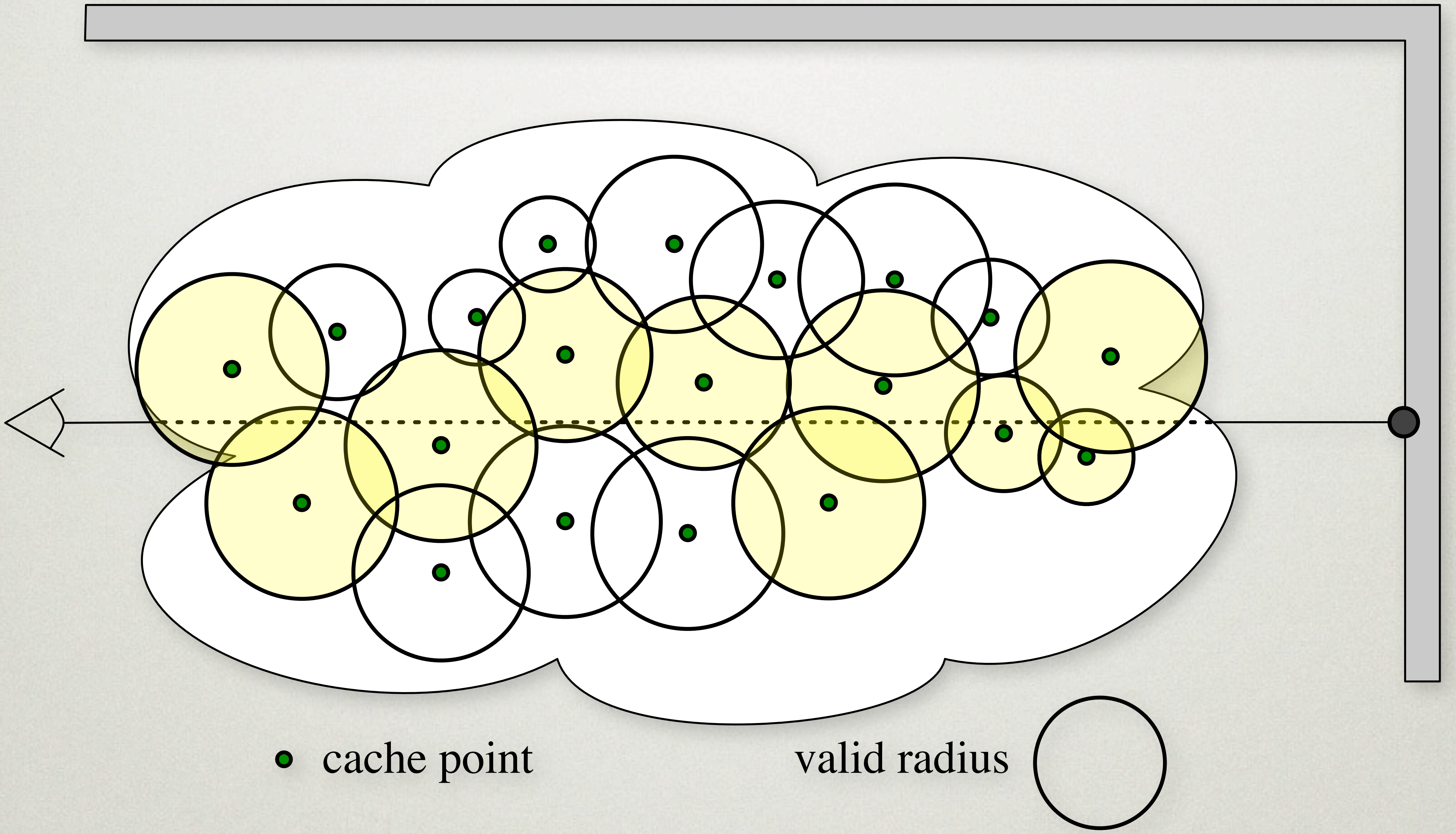
* production quality, physically based

Radiance Caching in Participating Media



* at a high level, radiance caching gains efficiency by caching expensive lighting calculations within the medium.

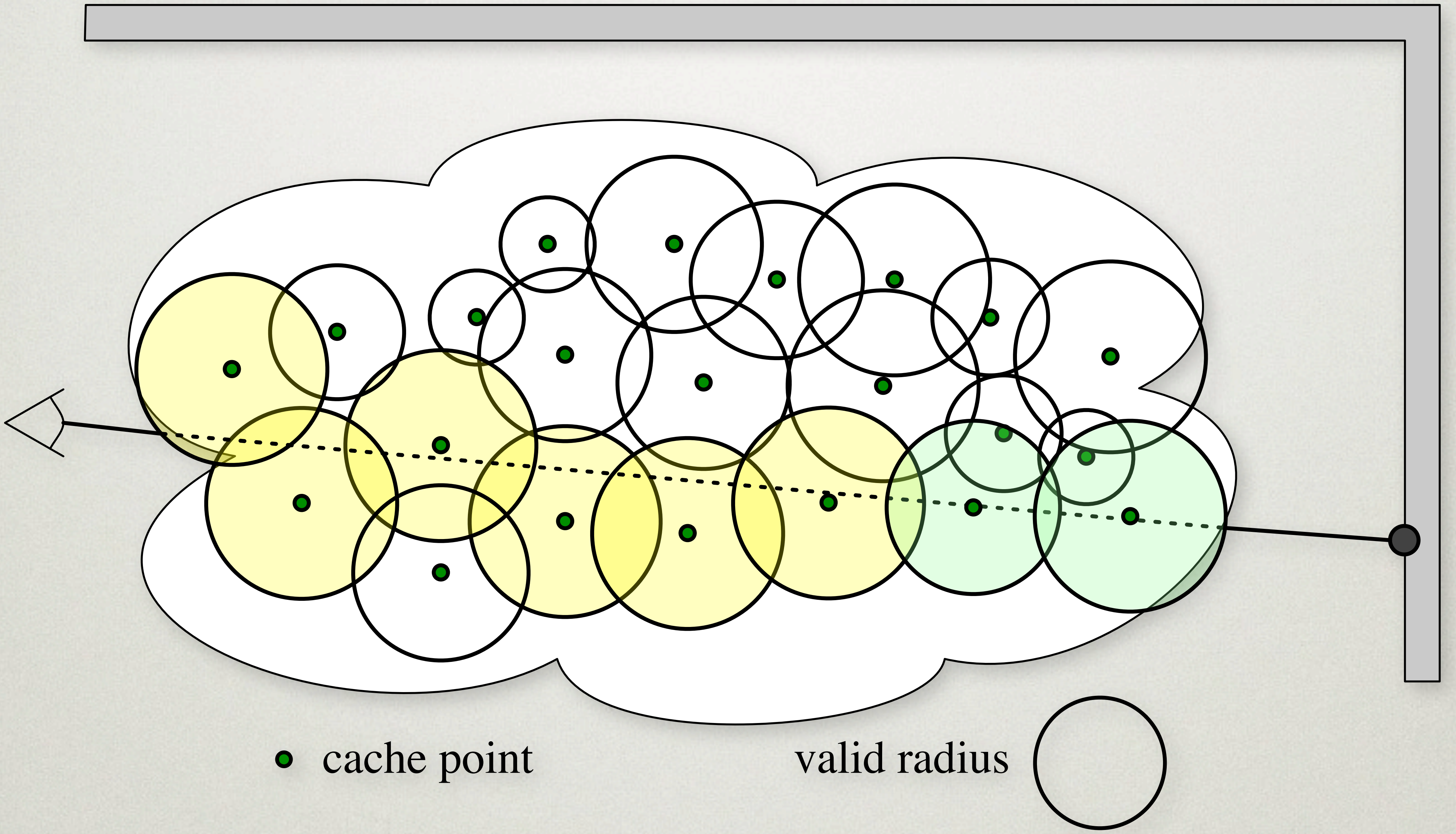
Radiance Caching in Participating Media



Thursday, 6 September 12

* for this ray, since a cache point overlaps with every part of the ray, we can compute the lighting by interpolating the cache points

Radiance Caching in Participating Media



Thursday, 6 September 12

* a neighboring ray can re-use many of the same cache points

Challenges

- What should the cache points store?
- Where to place cache points to minimize visible error?
- How to interpolate cache points accurately?

* Cannot re-use details from irradiance caching directly, since many underlying assumptions are different.
* What is a “good” valid radius?
* How do we interpolate the nearby cached values?

Approach

- Cache inscattered radiance:

$$L(\mathbf{e}, \vec{\omega}) = \int_0^s T_r(\mathbf{e} \leftrightarrow \mathbf{x}) \sigma_s(\mathbf{x}) L_i(\mathbf{x}, \vec{\omega}) \, d\mathbf{x} + T_r(\mathbf{e} \leftrightarrow \mathbf{x}') L(\mathbf{x}', \vec{\omega})$$

* Since the gradient is a local measure of the smoothness of the radiance field, we use it to estimate a valid radius within which it's OK to extrapolate each cache point.

Approach

- Cache inscattered radiance:

$$L(\mathbf{e},\vec{\omega}) = \int_0^s T_r(\mathbf{e}\leftrightarrow\mathbf{x})\sigma_s(\mathbf{x}) L_i(\mathbf{x},\vec{\omega})\, d\mathbf{x} + T_r(\mathbf{e}\leftrightarrow\mathbf{x}')L(\mathbf{x}',\vec{\omega})$$

- Compute gradients due to translation

* Since the gradient is a local measure of the smoothness of the radiance field, we use it to estimate a valid radius within which it’s OK to extrapolate each cache point.

Approach

- Cache inscattered radiance:

$$L(\mathbf{e},\vec{\omega}) = \int_0^s T_r(\mathbf{e}\leftrightarrow\mathbf{x})\sigma_s(\mathbf{x})\boxed{L_i(\mathbf{x},\vec{\omega})}\,d\mathbf{x} + T_r(\mathbf{e}\leftrightarrow\mathbf{x}')L(\mathbf{x}',\vec{\omega})$$

- Compute gradients due to translation
- Use gradients to:
  - Estimate valid radius within which it's OK to extrapolate
  - Provide high quality interpolation

* Since the gradient is a local measure of the smoothness of the radiance field, we use it to estimate a valid radius within which it's OK to extrapolate each cache point.

Radiance Computation

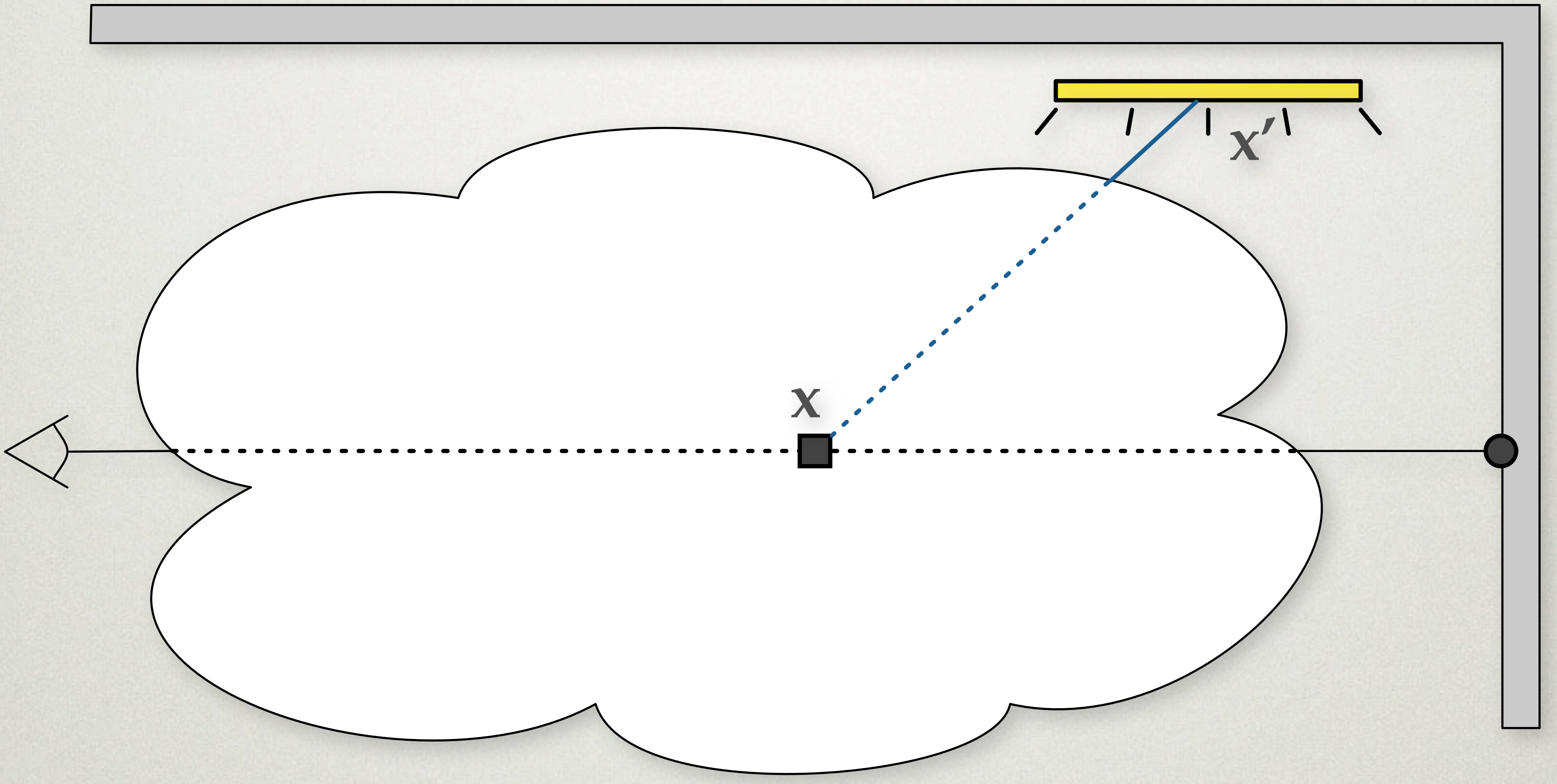
- In order to make gradient derivations more convenient:
  - Split computation into single and multiple scattering components:

$$L_i = L_s + L_m$$

- How do we compute L_s and L_m?
- How do we compute ∇L_s and ∇L_m?

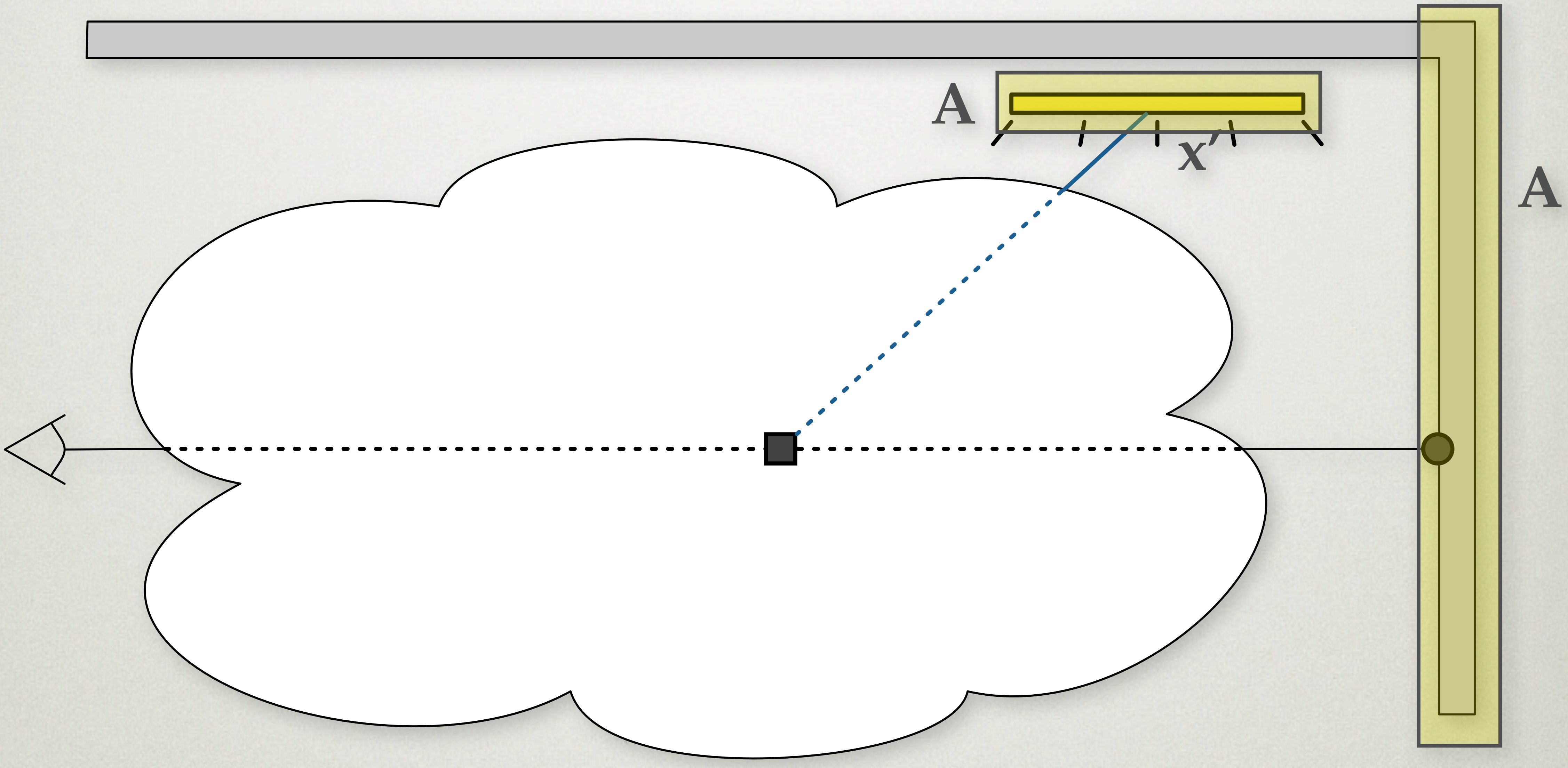
* we cache values of Li
* split because:
  * it makes the derivations more convenient
* single scatter is light that only scatters once in the medium before reaching the eye.
* multiple scattering scatters at least twice

Single Scattering



$$L_s(\mathbf{x},\vec{\omega}) = \int_A p(\vec{\omega},\mathbf{x}'\!\rightarrow\!\mathbf{x})L_r(\mathbf{x}'\!\rightarrow\!\mathbf{x})V(\mathbf{x}'\!\leftrightarrow\!\mathbf{x})H(\mathbf{x}'\!\rightarrow\!\mathbf{x})\;d\mathbf{x}'$$

Single Scattering



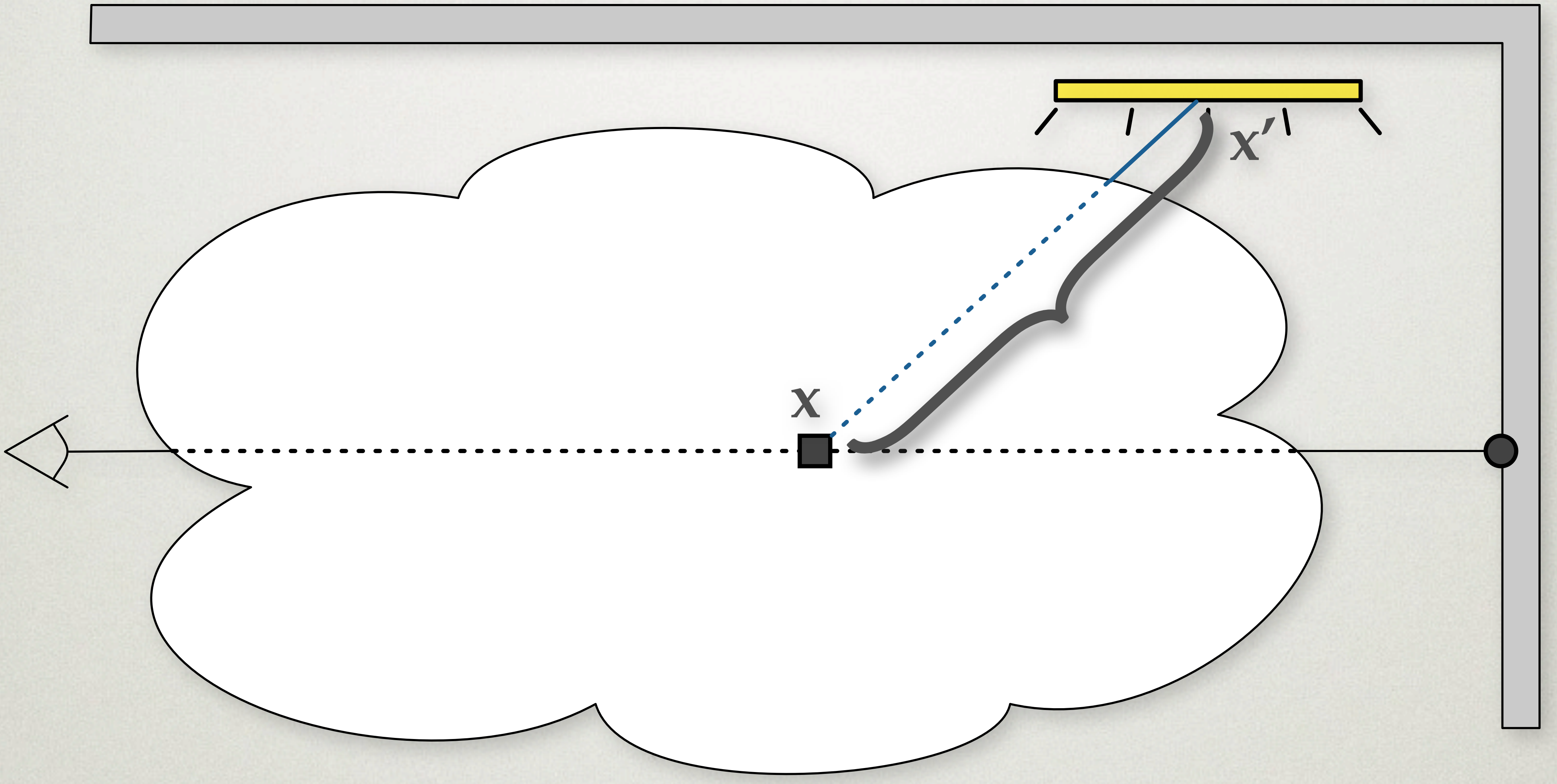
$$L_s(\mathbf{x},\vec{\omega}) = \int_A p(\vec{\omega},\mathbf{x}'\!\rightarrow\!\mathbf{x}) L_r(\mathbf{x}'\!\rightarrow\!\mathbf{x}) V(\mathbf{x}'\!\leftrightarrow\!\mathbf{x}) H(\mathbf{x}'\!\rightarrow\!\mathbf{x})\, d\mathbf{x}'$$

Integration over surface area

Thursday, 6 September 12

* over area of light sources and surfaces
* reduce radiance
* radiance from light, diminished through medium

Single Scattering



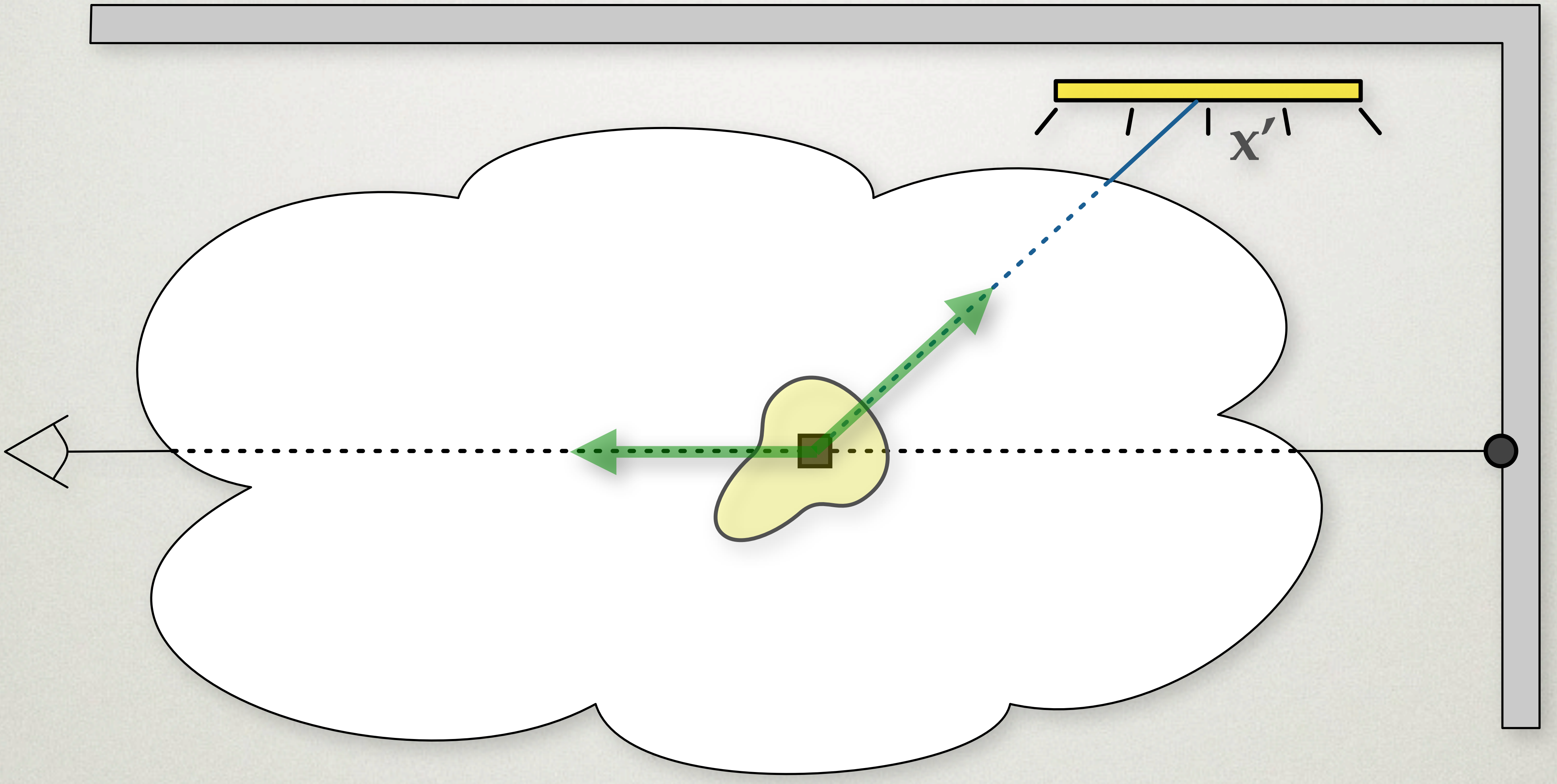
$$L_s(\mathbf{x}, \vec{\omega}) = \int_A p(\vec{\omega}, \mathbf{x}' \!\rightarrow\! \mathbf{x}) L_r(\mathbf{x}' \!\rightarrow\! \mathbf{x}) V(\mathbf{x}' \!\leftrightarrow\! \mathbf{x}) H(\mathbf{x}' \!\rightarrow\! \mathbf{x}) \, d\mathbf{x}'$$

Reduced Radiance: $L_r(\mathbf{x}' \!\rightarrow\! \mathbf{x}) = L(\mathbf{x}' \!\rightarrow\! \mathbf{x}) T_r(\mathbf{x}' \!\leftrightarrow\! \mathbf{x})$

Thursday, 6 September 12

* over area of light sources and surfaces
* reduce radiance
* radiance from light, diminished through medium

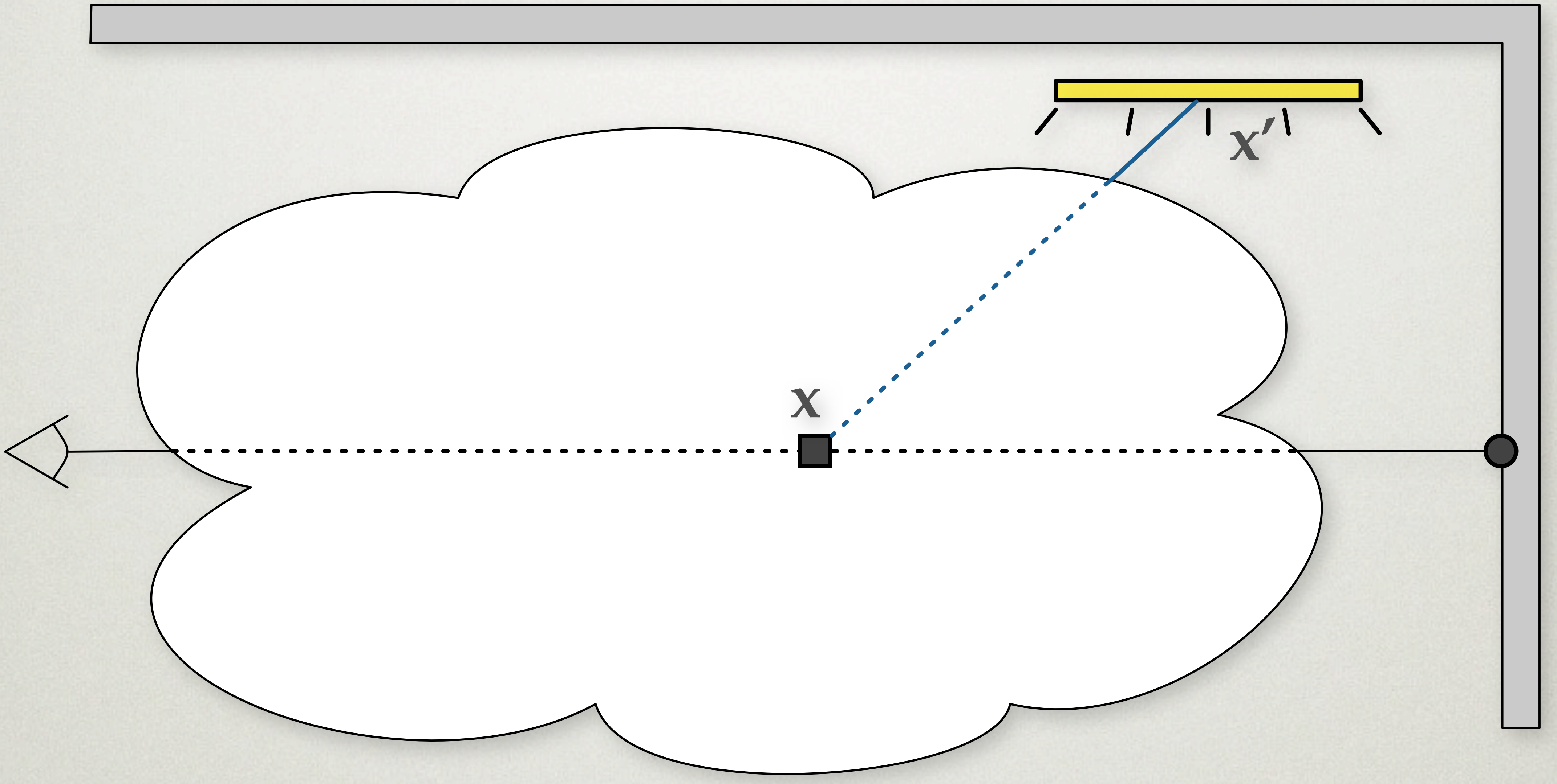
Single Scattering



$$L_s(\mathbf{x}, \vec{\omega}) = \int_A p(\vec{\omega}, \mathbf{x}' \!\rightarrow\! \mathbf{x}) L_r(\mathbf{x}' \!\rightarrow\! \mathbf{x}) V(\mathbf{x}' \!\leftrightarrow\! \mathbf{x}) H(\mathbf{x}' \!\rightarrow\! \mathbf{x})\, d\mathbf{x}'$$

Phase function

Single Scattering



$$L_s(\mathbf{x},\vec{\omega}) = \int_A p(\vec{\omega},\mathbf{x}'\!\rightarrow\!\mathbf{x})L_r(\mathbf{x}'\!\rightarrow\!\mathbf{x})V(\mathbf{x}'\!\leftrightarrow\!\mathbf{x})H(\mathbf{x}'\!\rightarrow\!\mathbf{x})\,d\mathbf{x}'$$

Visibility Function and Geometry Term

* there is also a visibility and geometry function

Gradient Computation

$$L_s(\mathbf{x},\vec{\omega}) = \int_A p(\vec{\omega},\mathbf{x}'\!\rightarrow\!\mathbf{x}) L_r(\mathbf{x}'\!\rightarrow\!\mathbf{x}) V(\mathbf{x}'\!\leftrightarrow\!\mathbf{x}) H(\mathbf{x}'\!\rightarrow\!\mathbf{x})\, d\mathbf{x}'$$

* In order to obtain the gradient, we analytically differentiate the terms in the integrand using the product rule.
* gradient of Lr is most significant:
   * accounts for change in transmission, even in heterogeneous media

Gradient Computation

$$L_s(\mathbf{x},\vec{\omega}) = \int_A p(\vec{\omega},\mathbf{x}'\!\rightarrow\!\mathbf{x}) L_r(\mathbf{x}'\!\rightarrow\!\mathbf{x}) V(\mathbf{x}'\!\leftrightarrow\!\mathbf{x}) H(\mathbf{x}'\!\rightarrow\!\mathbf{x})\, d\mathbf{x}'$$

$$\downarrow$$

$$\nabla L_s(\mathbf{x},\vec{\omega}) = \int_A (\nabla p) L_r V H + p(\nabla L_r) V H + p L_r V(\nabla H)\, d\mathbf{x}'$$

* In order to obtain the gradient, we analytically differentiate the terms in the integrand using the product rule.
* gradient of Lr is most significant:
   * accounts for change in transmission, even in heterogeneous media

Gradient Computation

$$L_s(\mathbf{x},\vec{\omega}) = \int_A p(\vec{\omega},\mathbf{x}'\!\to\!\mathbf{x})L_r(\mathbf{x}'\!\to\!\mathbf{x})V(\mathbf{x}'\!\leftrightarrow\!\mathbf{x})H(\mathbf{x}'\!\to\!\mathbf{x})\;d\mathbf{x}'$$

$$\downarrow$$

$$\nabla L_s(\mathbf{x},\vec{\omega}) = \int_A (\nabla p)L_r V H + p(\nabla L_r)VH + pL_rV(\nabla H)\;d\mathbf{x}'$$

- Assumes constant visibility

* In order to obtain the gradient, we analytically differentiate the terms in the integrand using the product rule.
* gradient of Lr is most significant:
  * accounts for change in transmission, even in heterogeneous media

Gradient Computation

$$L_s(\mathbf{x}, \vec{\omega}) = \int_A p(\vec{\omega}, \mathbf{x}' \to \mathbf{x}) L_r(\mathbf{x}' \to \mathbf{x}) V(\mathbf{x}' \leftrightarrow \mathbf{x}) H(\mathbf{x}' \to \mathbf{x})\, d\mathbf{x}'$$

$$\downarrow$$

$$\nabla L_s(\mathbf{x}, \vec{\omega}) = \int_A (\nabla p) L_r V H + p(\nabla L_r) V H + p L_r V (\nabla H)\, d\mathbf{x}'$$

- Assumes constant visibility
- Evaluated **together** using Monte Carlo integration and ray marching

* In order to obtain the gradient, we analytically differentiate the terms in the integrand using the product rule.
* gradient of Lr is most significant:
  * accounts for change in transmission, even in heterogeneous media

Gradient Computation

$$L_s(\mathbf{x}, \vec{\omega}) = \int_A p(\vec{\omega}, \mathbf{x}' \!\to\! \mathbf{x}) L_r(\mathbf{x}' \!\to\! \mathbf{x}) V(\mathbf{x}' \!\leftrightarrow\! \mathbf{x}) H(\mathbf{x}' \!\to\! \mathbf{x}) \, d\mathbf{x}'$$

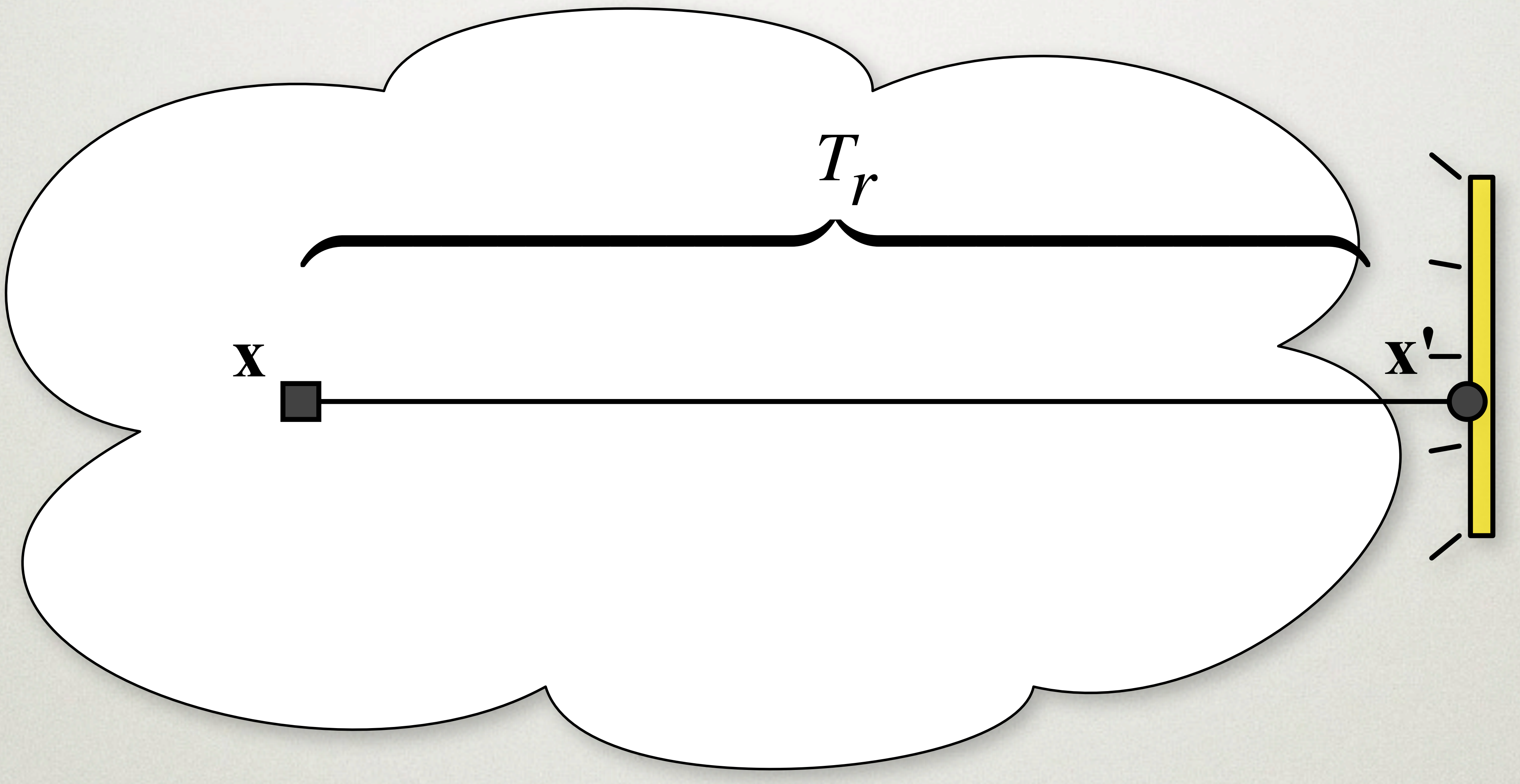
$$\downarrow$$

$$\nabla L_s(\mathbf{x}, \vec{\omega}) = \int_A (\nabla p) L_r V H + p (\nabla L_r) V H + p L_r V (\nabla H) \, d\mathbf{x}'$$

- Assumes constant visibility
- Evaluated **together** using Monte Carlo integration and ray marching
- Gradients take into account changing properties of medium **along the whole ray**

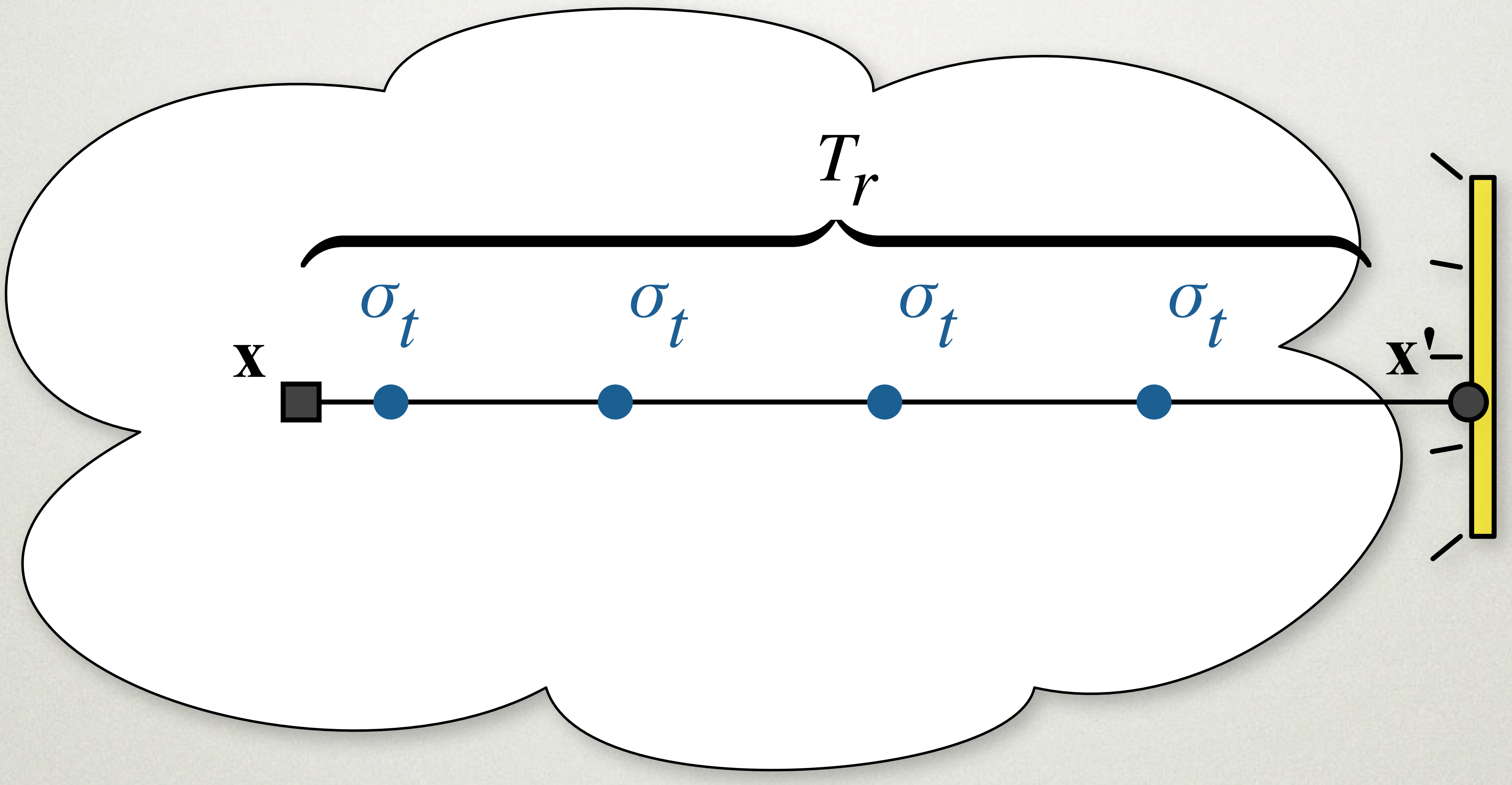
* In order to obtain the gradient, we analytically differentiate the terms in the integrand using the product rule.
* gradient of Lr is most significant:
   * accounts for change in transmission, even in heterogeneous media

Reduced Radiance

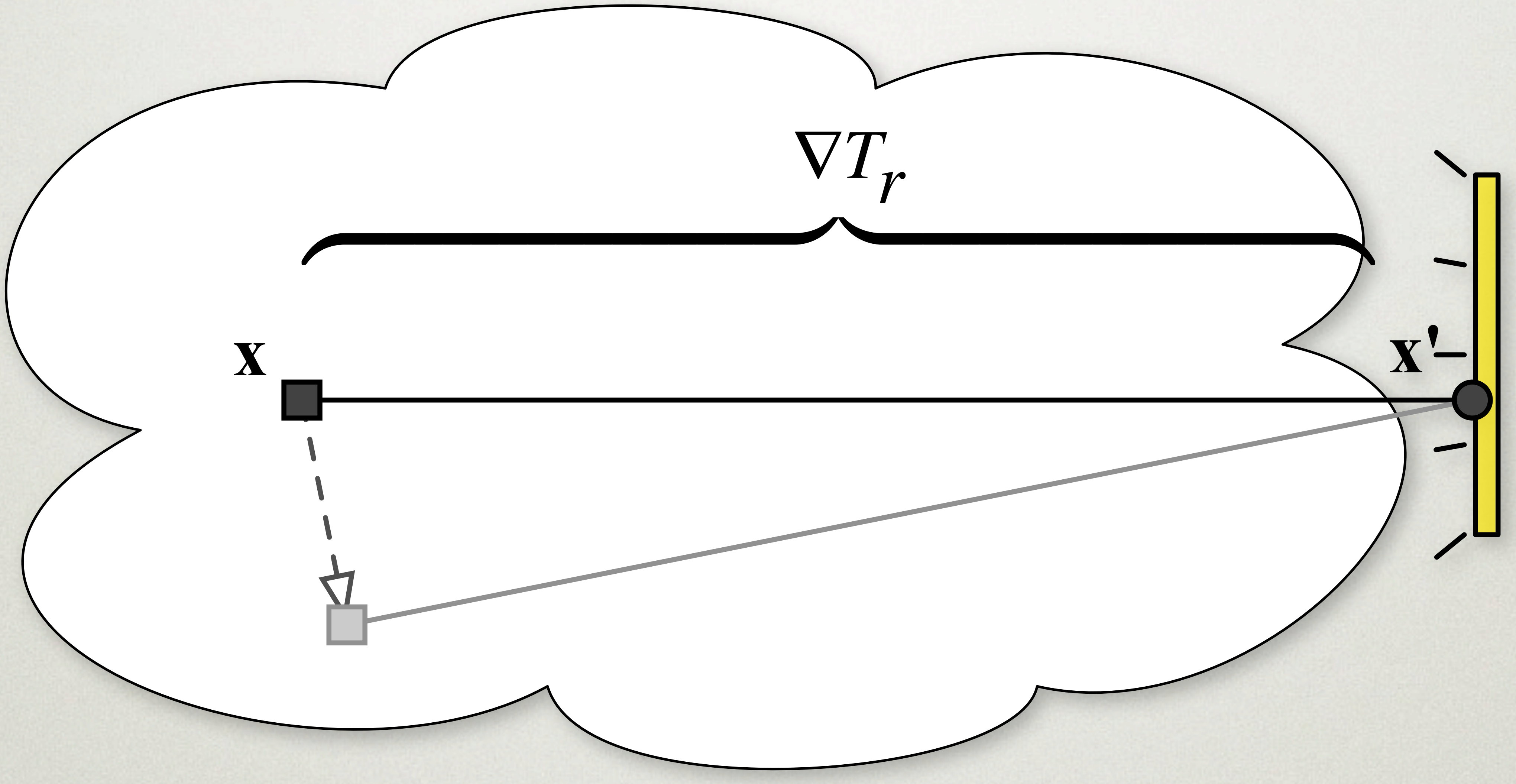


Reduced Radiance: $L_r(\mathbf{x}' \to \mathbf{x}) = L(\mathbf{x}' \to \mathbf{x}) T_r(\mathbf{x}' \leftrightarrow \mathbf{x})$

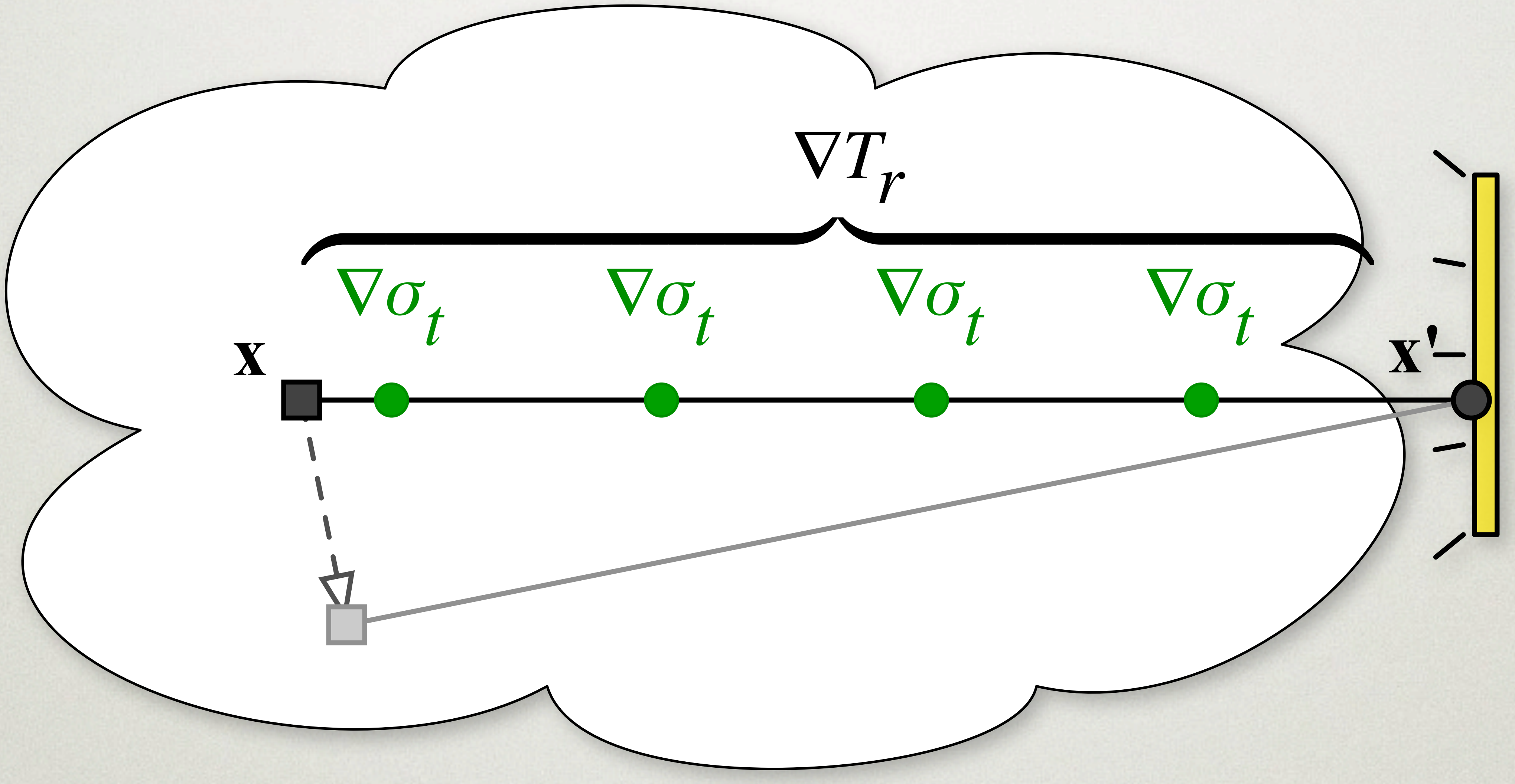
Ray Marching



Transmission Gradient



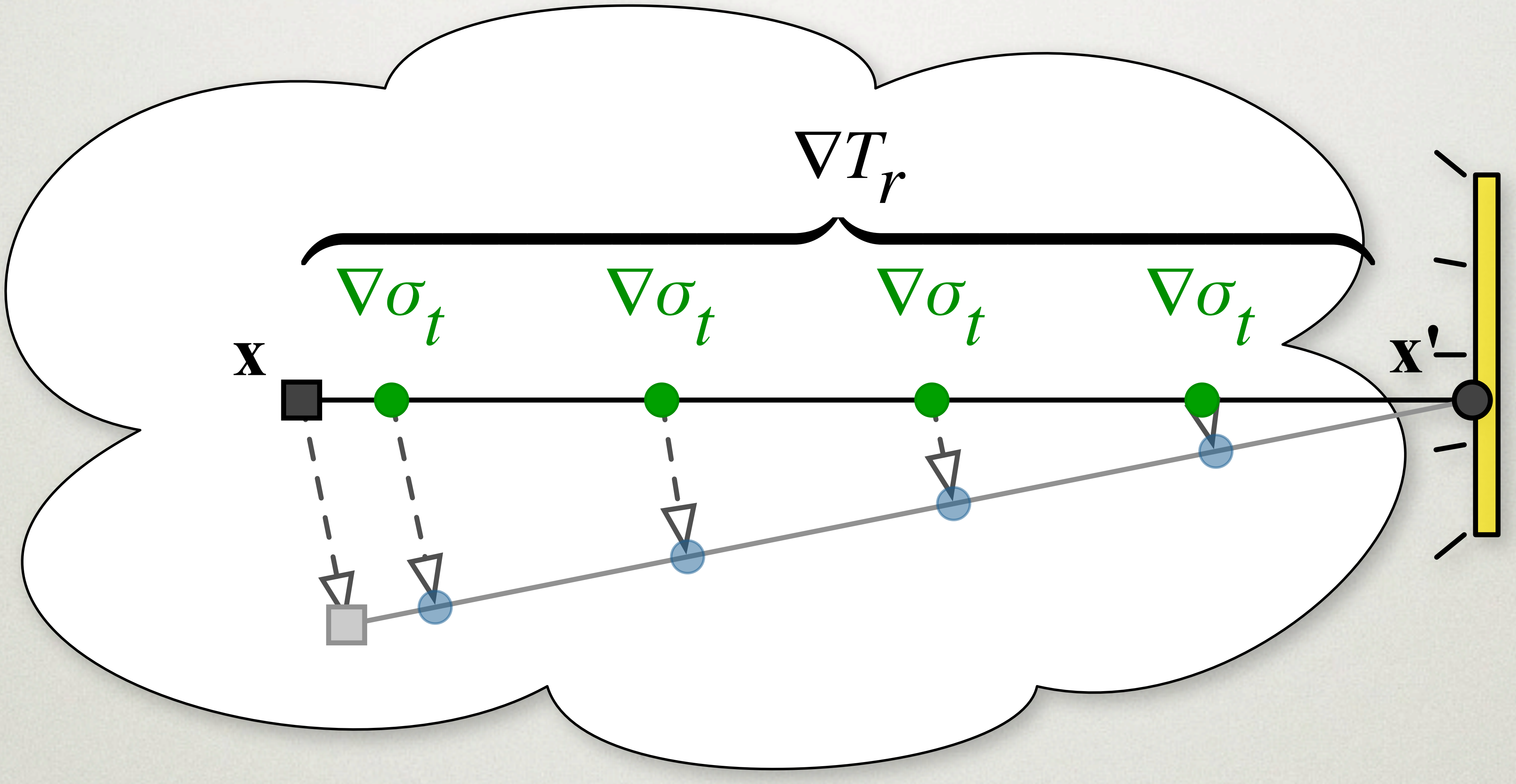
Transmission Gradient



Thursday, 6 September 12

* take into account how the extinction coefficients change along the whole ray segment

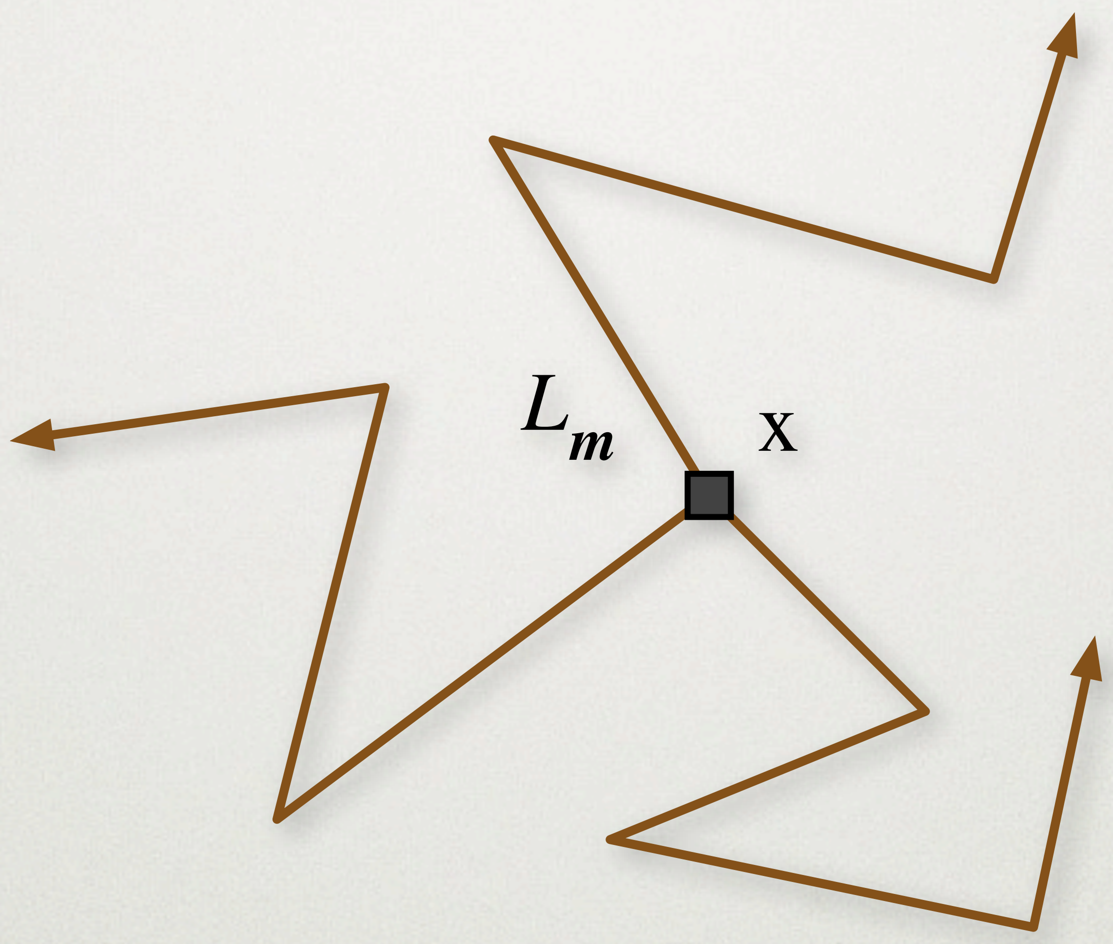
Transmission Gradient



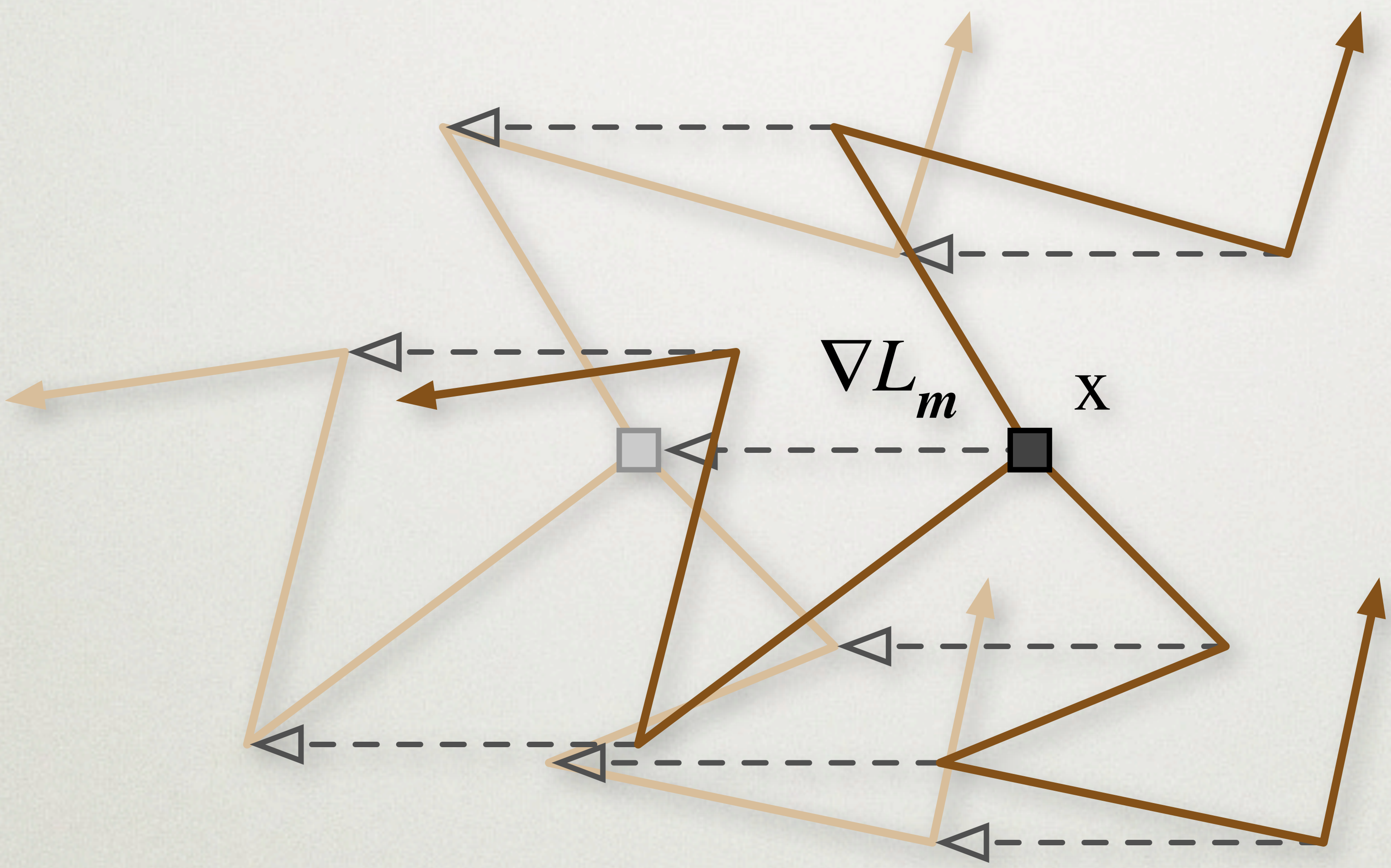
Thursday, 6 September 12

* these changes would induce a different overall transmission when x is translated
* gradients contain meaningful information about how Tr changes as we move x in any direction, even out of the line connecting x to x'

Multiple Scattering



Multiple Scattering Gradient



Cache Storage

- Cached points store:
  - 3D position
  - Value (inscattered radiance)
  - Gradient
  - Valid Radius

* distinct caches for single, surface, and multiple scattering

Cache Storage

Isotropic Media

- Cached points store:
  - 3D position
  - Value
  - Gradient
  - Valid Radius

} ▸ inscattered radiance is a scalar

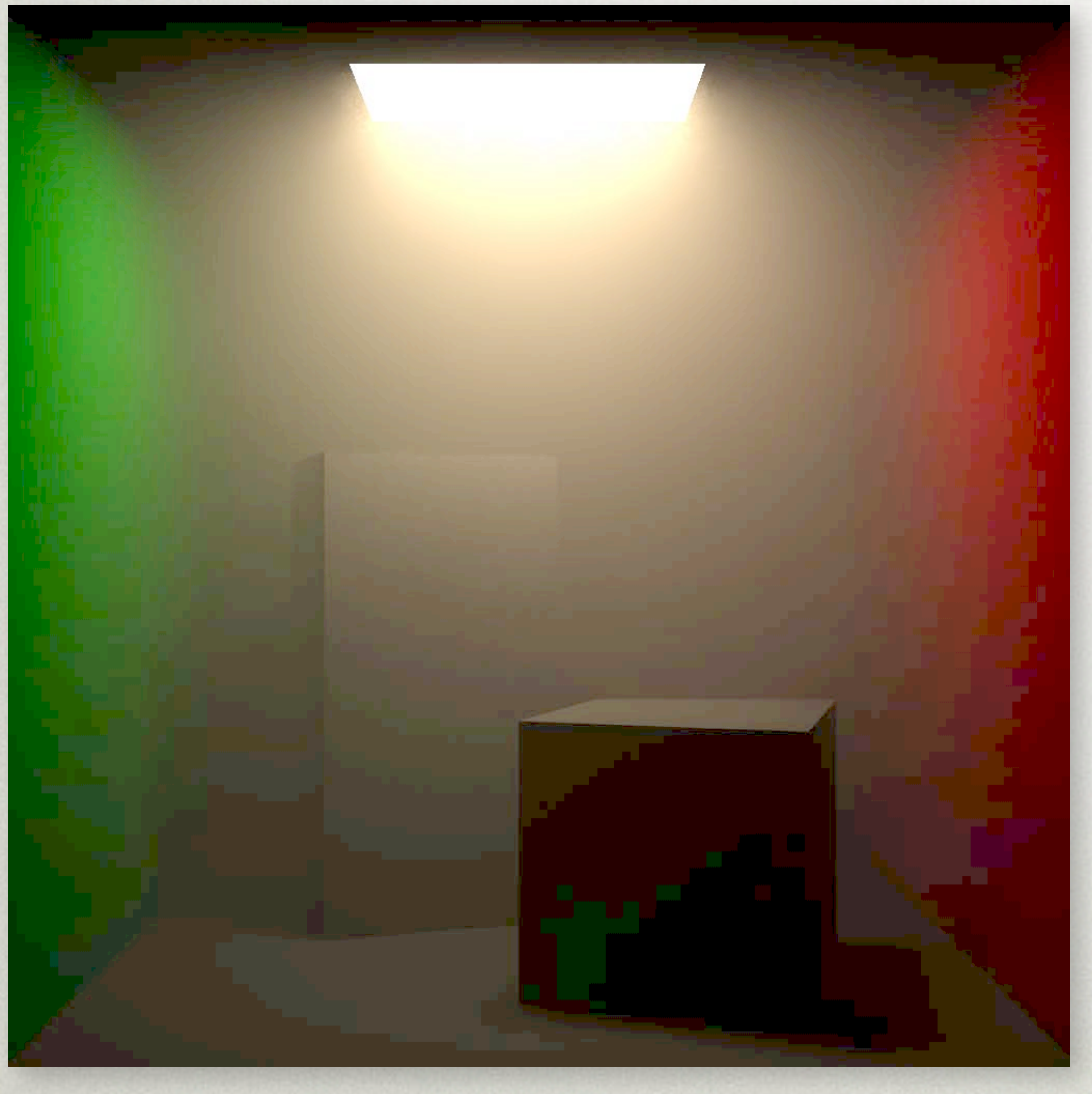
Cache Storage

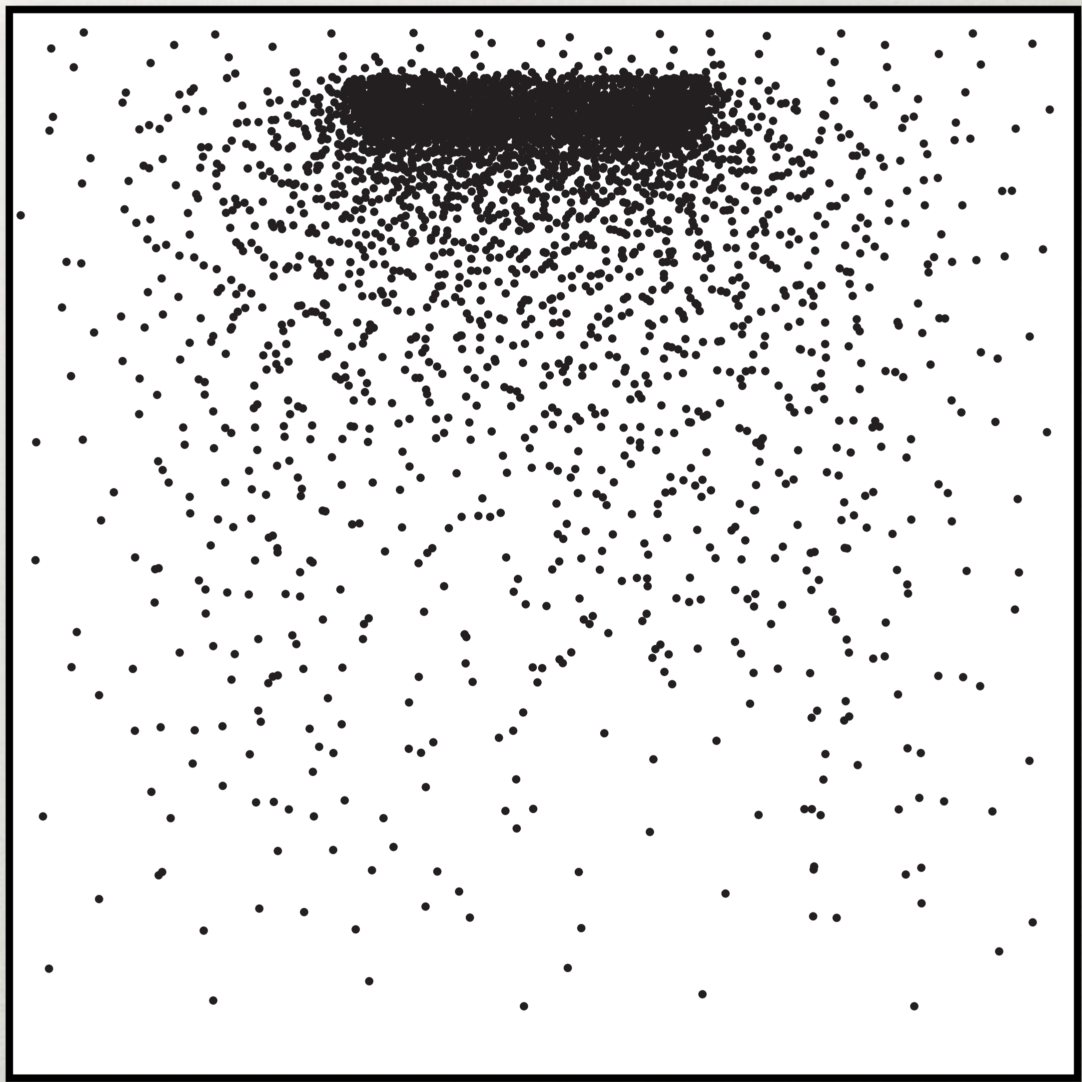
Anisotropic Media

- Cached points store:
  - 3D position
  - Value
  - Gradient
  - Valid Radius

(Value and Gradient:)
- inscattered radiance is a spherical function
- projected onto SH

Valid Radius





Thursday, 6 September 12

Want density of cache points to adapt to the local variation of illumination:

* smooth radiance → large radius, few cache points
* sharp radiance → small radius, many cache points

Optimal Radius

$$A(\mathbf{x}') = \text{actual radiance at } \mathbf{x}'$$

$$E_{\mathbf{x}}(\mathbf{x}') = \text{radiance extrapolated from } \mathbf{x} \text{ to } \mathbf{x}'$$

* Maximum radius such that the total relative error between the extrapolated and actual radiance within the cached region is below some error threshold t.
* using relative error because human vision is sensitive to contrast, not absolute errors

Optimal Radius

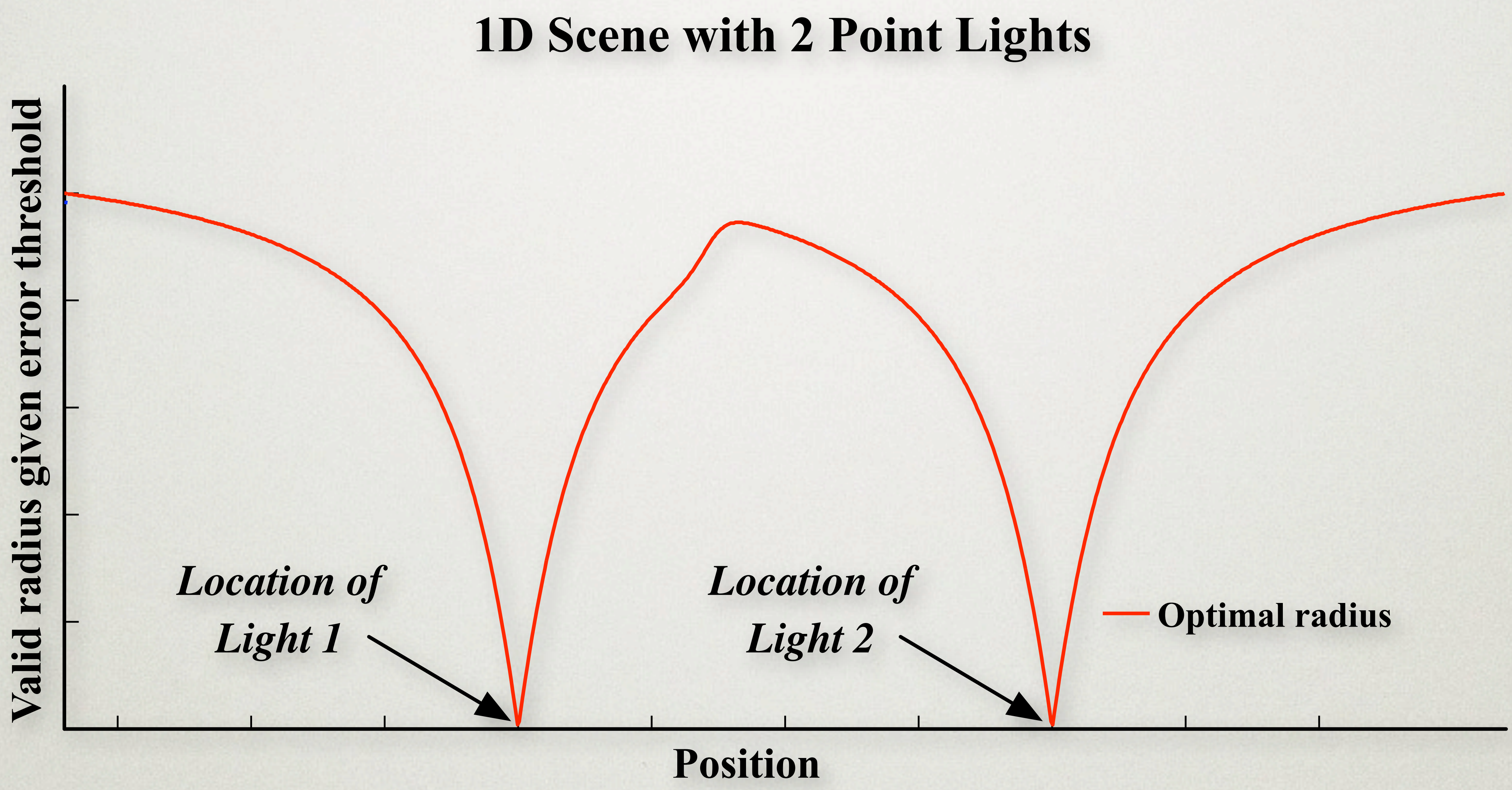
$$A(\mathbf{x}') = \text{actual radiance at } \mathbf{x}'$$

$$E_{\mathbf{x}}(\mathbf{x}') = \text{radiance extrapolated from } \mathbf{x} \text{ to } \mathbf{x}'$$

$$R_{opt}(\mathbf{x}) = \max_r \left(\frac{\int_{\mathbf{x}' \in \Omega_r} |E_{\mathbf{x}}(\mathbf{x}') - A(\mathbf{x}')| \, d\mathbf{x}'}{\int_{\mathbf{x}' \in \Omega_r} A(\mathbf{x}') \, d\mathbf{x}'} < t \right)$$

* Maximum radius such that the total relative error between the extrapolated and actual radiance within the cached region is below some error threshold t.
* using relative error because human vision is sensitive to contrast, not absolute errors

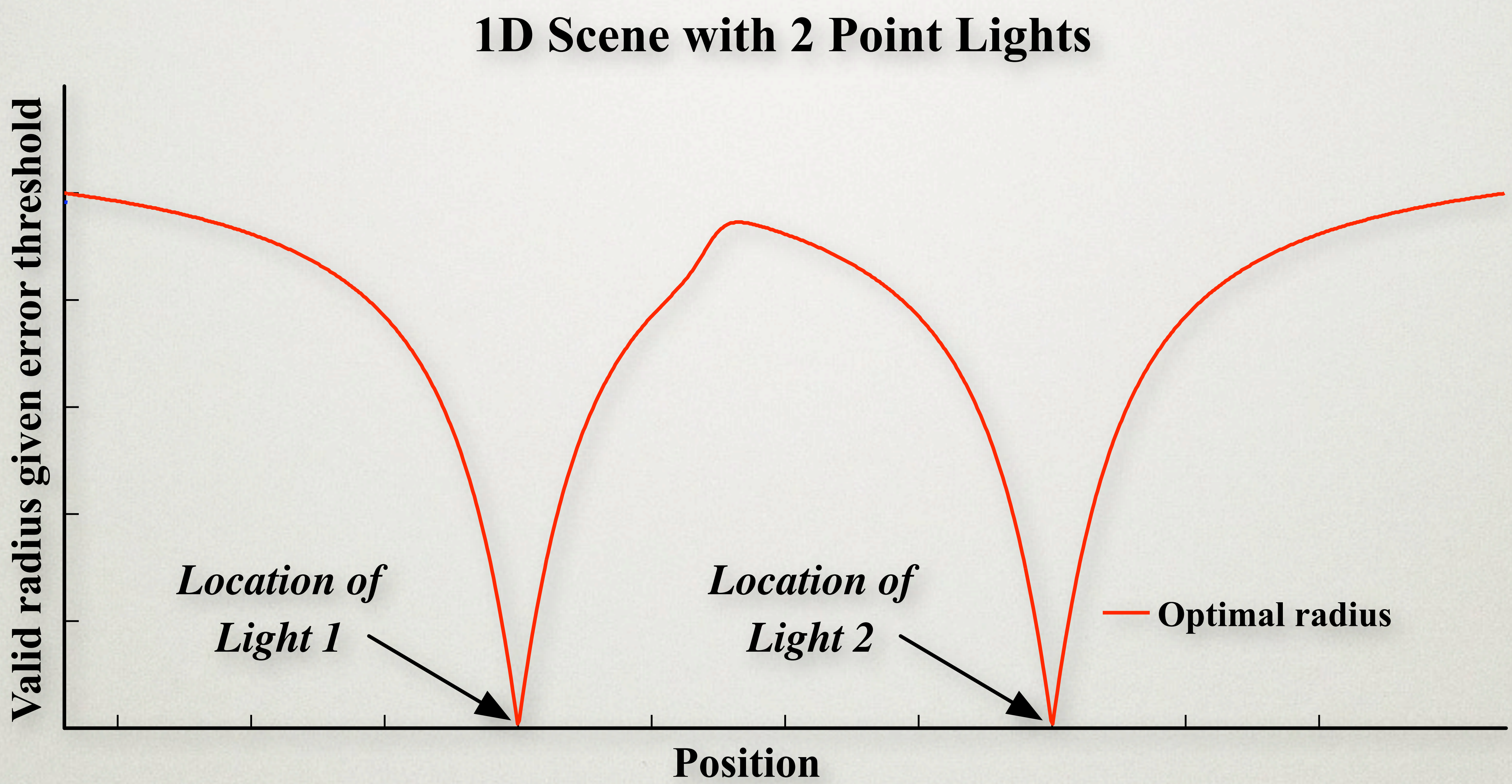
Valid Radius



Thursday, 6 September 12

* numerically computed optimal radius

Valid Radius

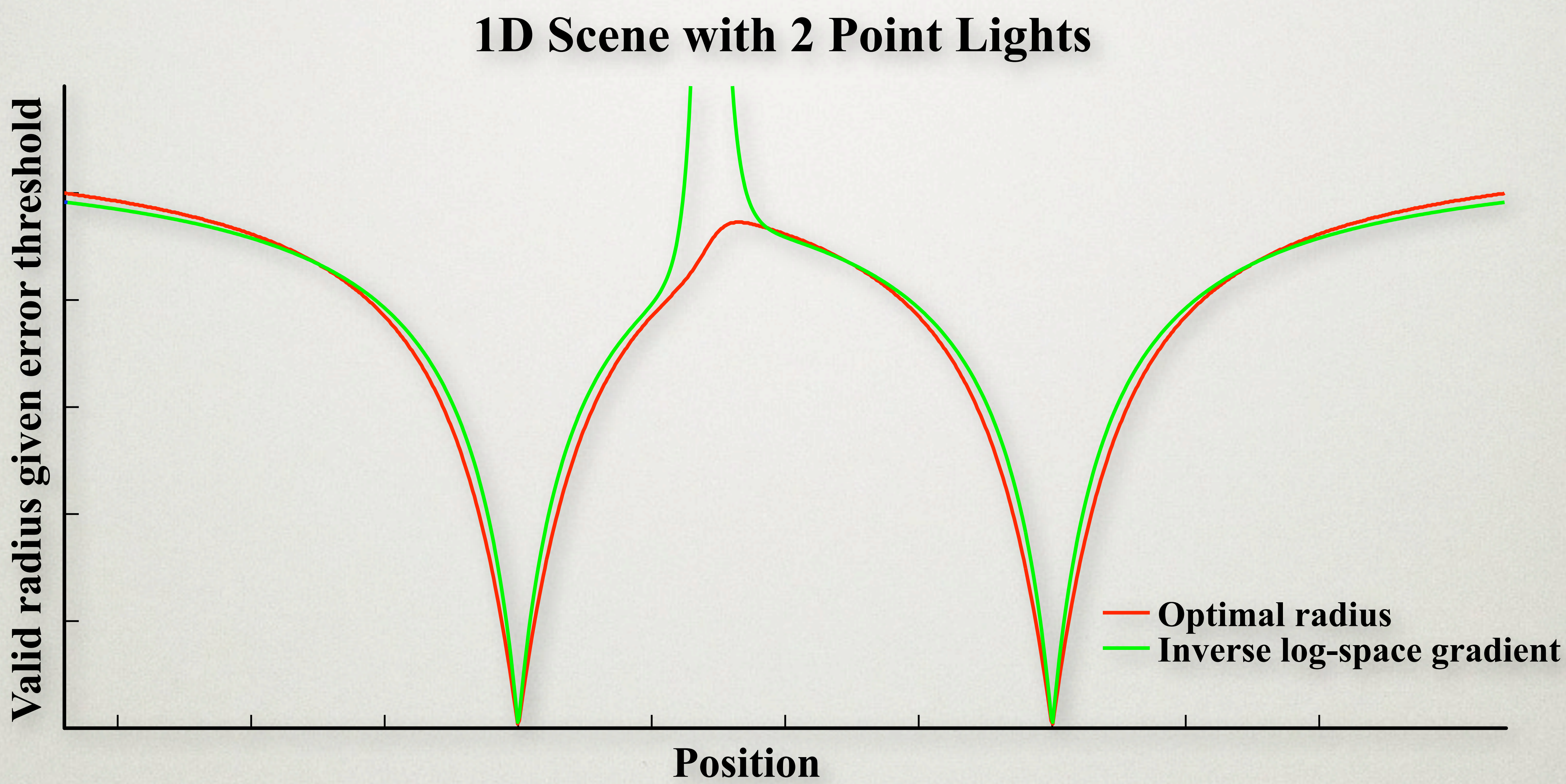


measure of local contrast:

$$\nabla \ln(L) = \frac{\nabla L}{L}$$

* log-space gradient. perceptually motivated: contrast

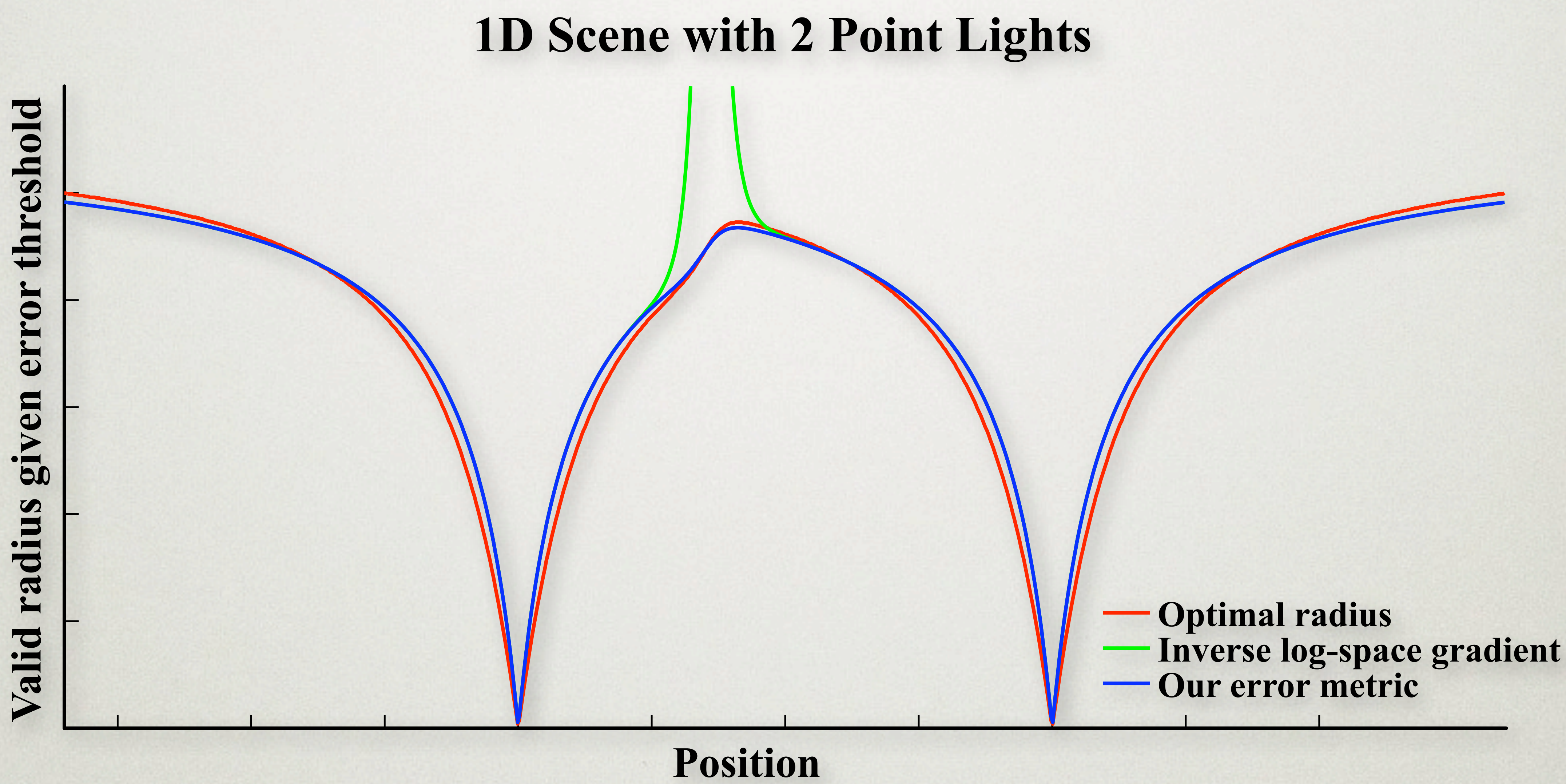
Valid Radius



measure of local contrast:

$$\nabla \ln(L) = \frac{\nabla L}{L} \qquad r \propto \frac{1}{\|\nabla \ln(L)\|} = \frac{\sum_j L_j}{\|\sum_j \nabla L_j\|}$$

Valid Radius



Avoid singularities:

$$r \propto \frac{\sum_j L_j}{\|\sum_j \nabla L_j\|} \longrightarrow r \propto \frac{\sum_j L_j}{\sum_j \|\nabla L_j\|}$$

Interpolation

- Perform a weighted interpolation from nearby cache points.

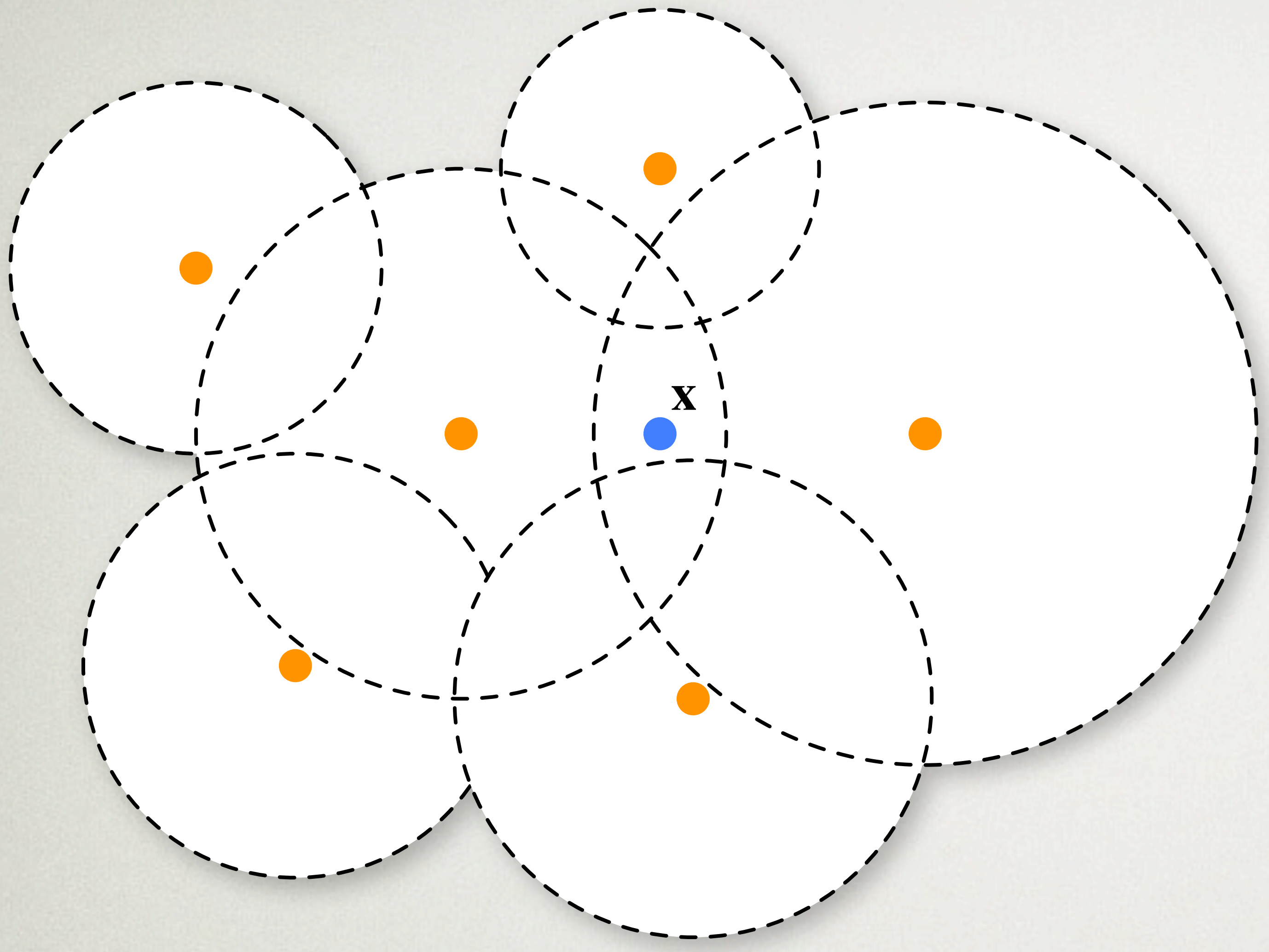
* whenever possible, interpolate from nearby cache points
* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function

Interpolation

- Perform a weighted interpolation from nearby cache points.

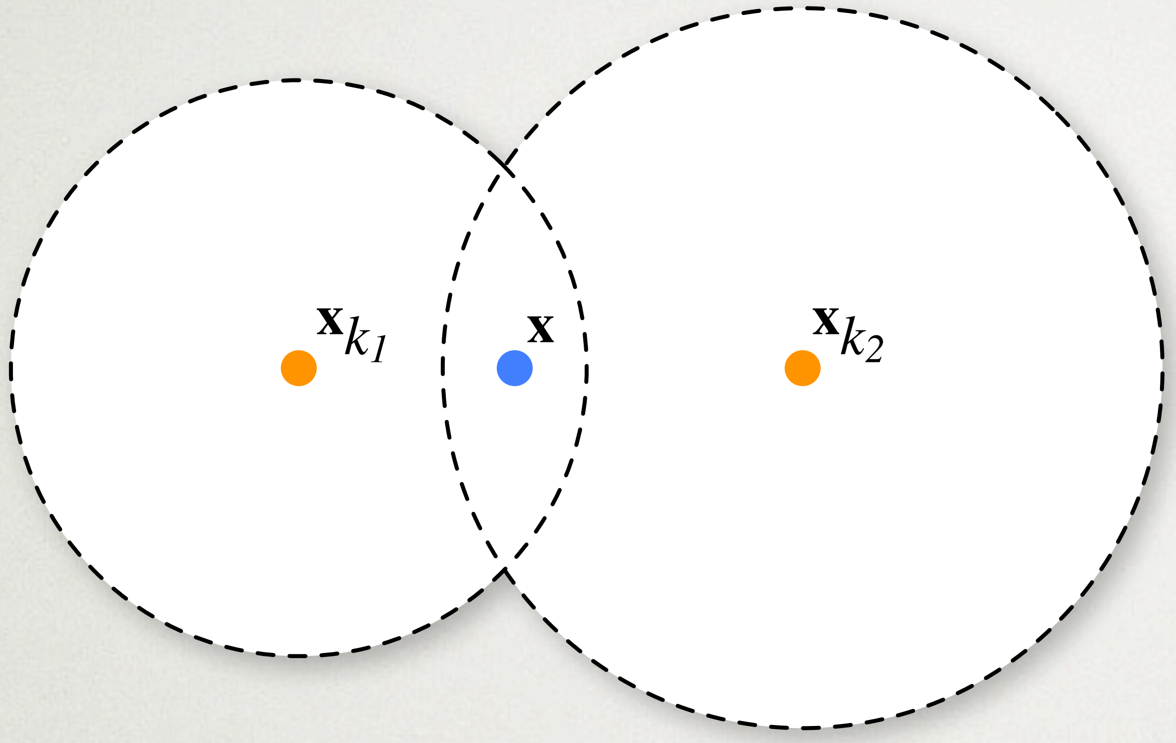
$$L(\mathbf{x}) \approx \exp\left(\frac{\sum_{k\in C}\left(\ln(L_k) + \frac{\nabla L_k}{L_k}\cdot(\mathbf{x}-\mathbf{x}_k)\right) w\left(\|\mathbf{x}-\mathbf{x}_k\|\right)}{\sum_{k\in C} w\left(\|\mathbf{x}-\mathbf{x}_k\|\right)}\right)$$

* whenever possible, interpolate from nearby cache points
* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function



$$L(\mathbf{x}) \approx \exp\left(\frac{\sum_{k\in C}\left(\ln(L_k) + \frac{\nabla L_k}{L_k}\cdot(\mathbf{x}-\mathbf{x}_k)\right) w\left(\|\mathbf{x}-\mathbf{x}_k\|\right)}{\sum_{k\in C} w\left(\|\mathbf{x}-\mathbf{x}_k\|\right)}\right)$$

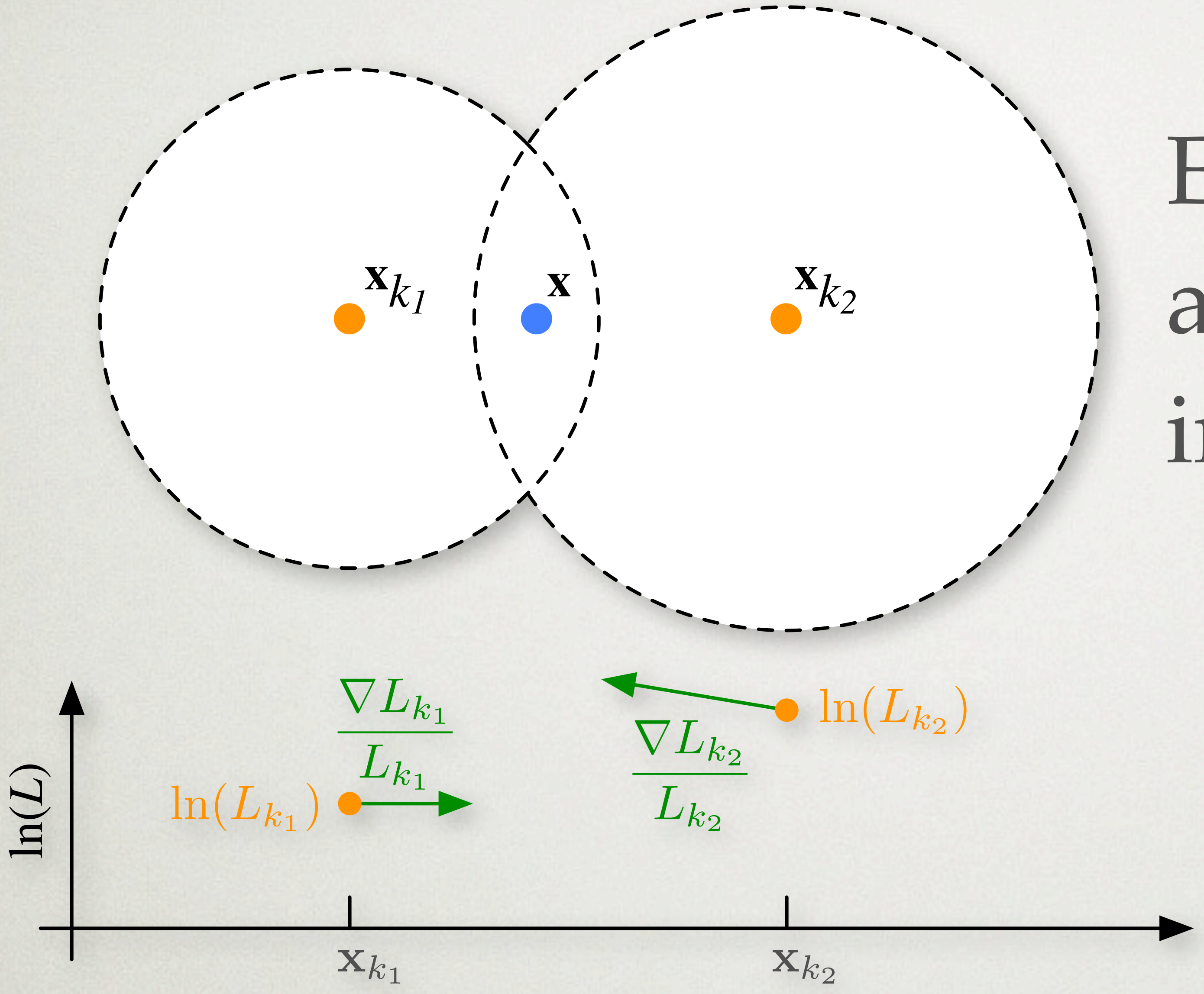
* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function



Find overlapping cache points.

$$L(\mathbf{x}) \approx \exp\left(\frac{\sum_{k\in C} \left(\ln(L_k) + \frac{\nabla L_k}{L_k} \cdot (\mathbf{x} - \mathbf{x}_k) \right) w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)}{\sum_{k\in C} w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)} \right)$$

* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function

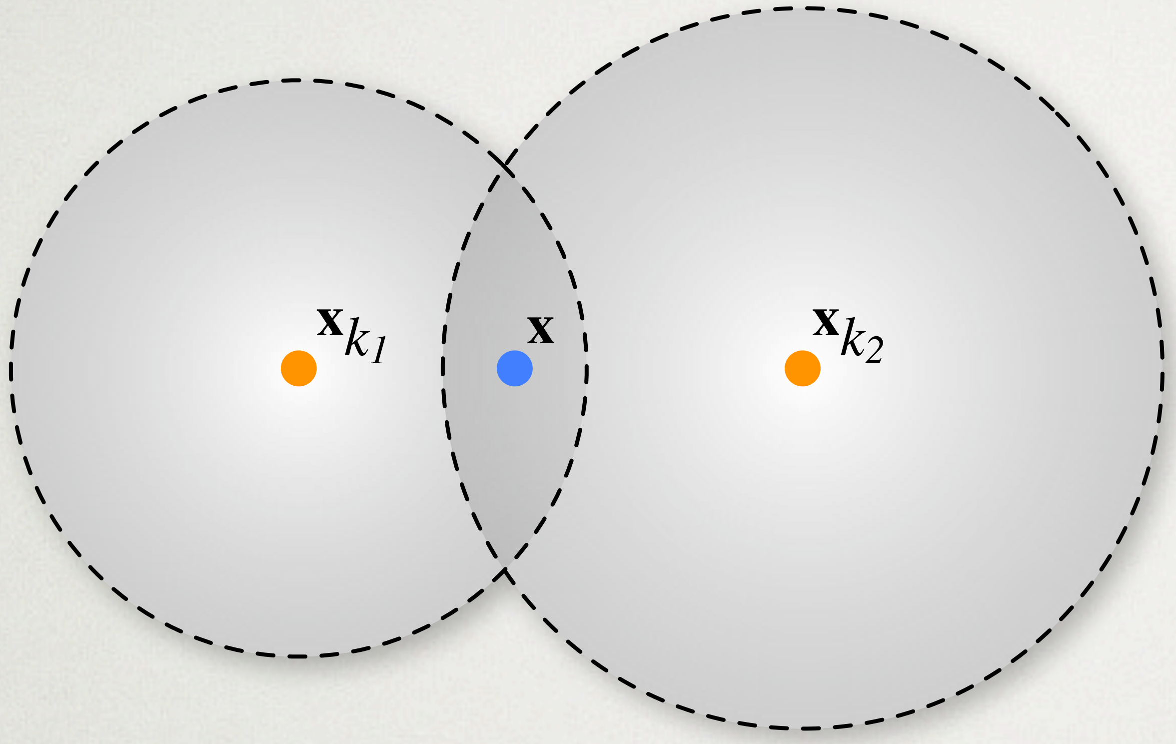


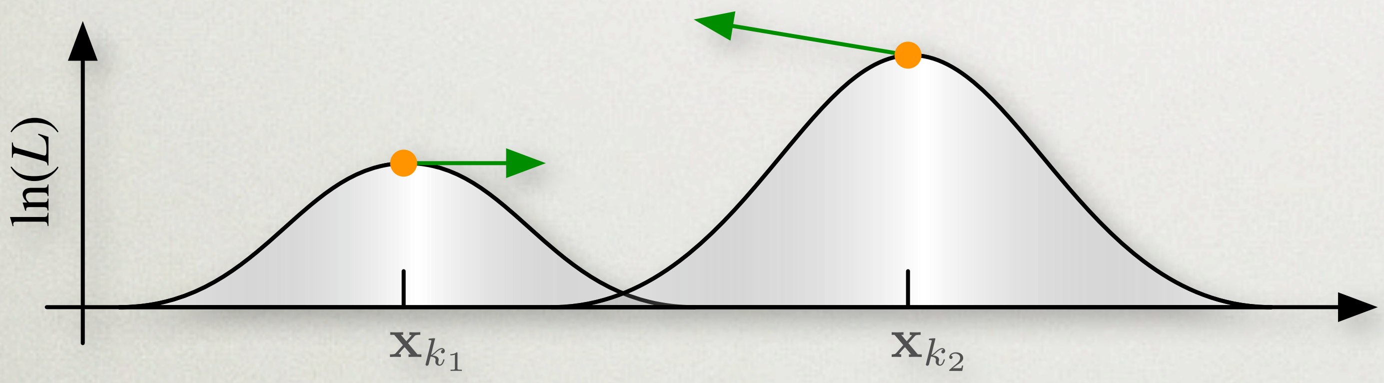
Extrapolate along gradients in log-space.

$$L(\mathbf{x}) \approx \exp\left(\frac{\sum_{k \in C} \left(\ln(L_k) + \frac{\nabla L_k}{L_k} \cdot (\mathbf{x} - \mathbf{x}_k) \right) w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)}{\sum_{k \in C} w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)} \right)$$

* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function

Weight contributions using smooth kernel

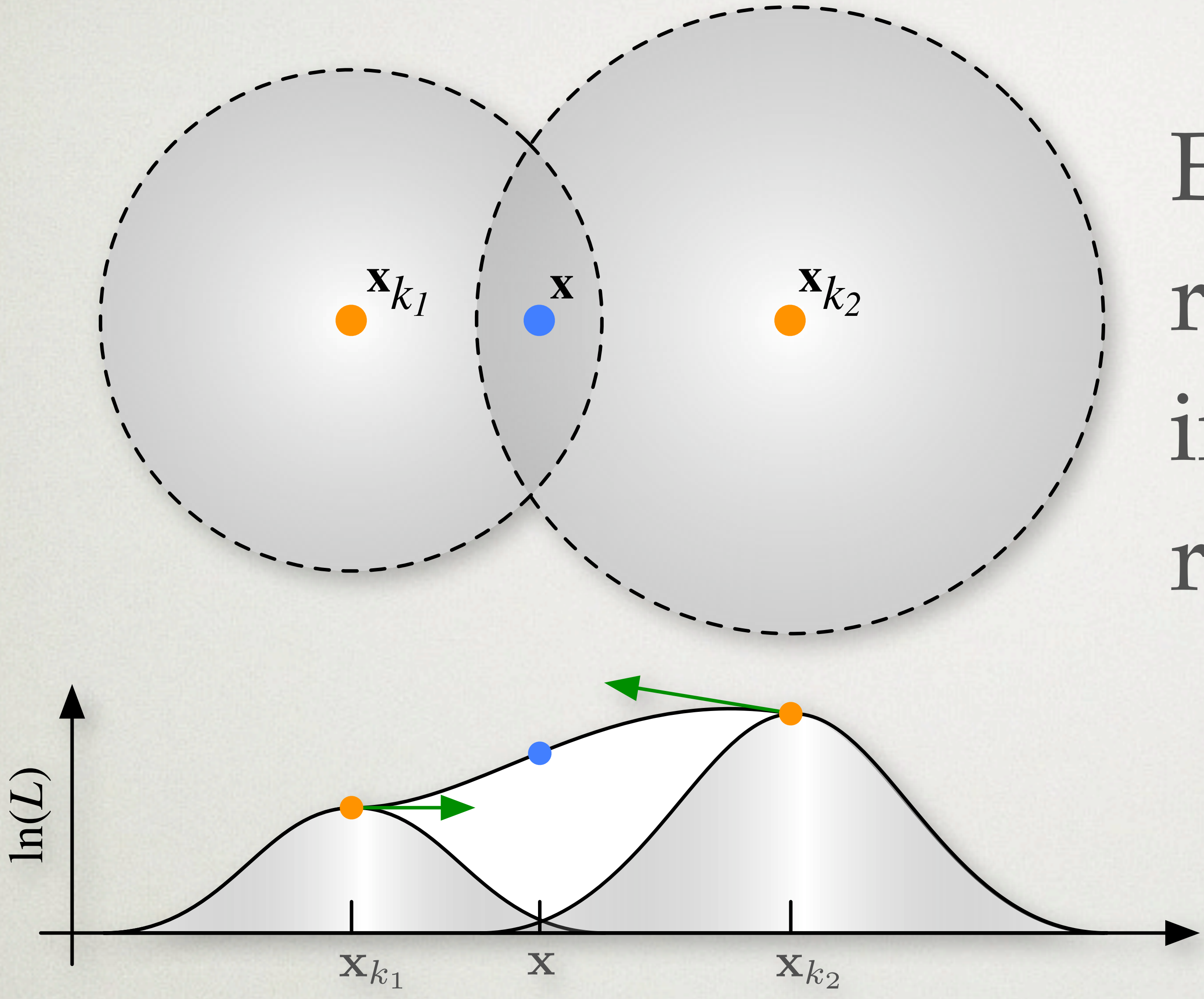




$$L(\mathbf{x}) \approx \exp\left(\frac{\sum_{k \in C} \left(\ln(L_k) + \frac{\nabla L_k}{L_k} \cdot (\mathbf{x} - \mathbf{x}_k) \right) w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)}{\sum_{k \in C} w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)} \right)$$

Thursday, 6 September 12

* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function



Exponentiate result to obtain interpolated radiance.

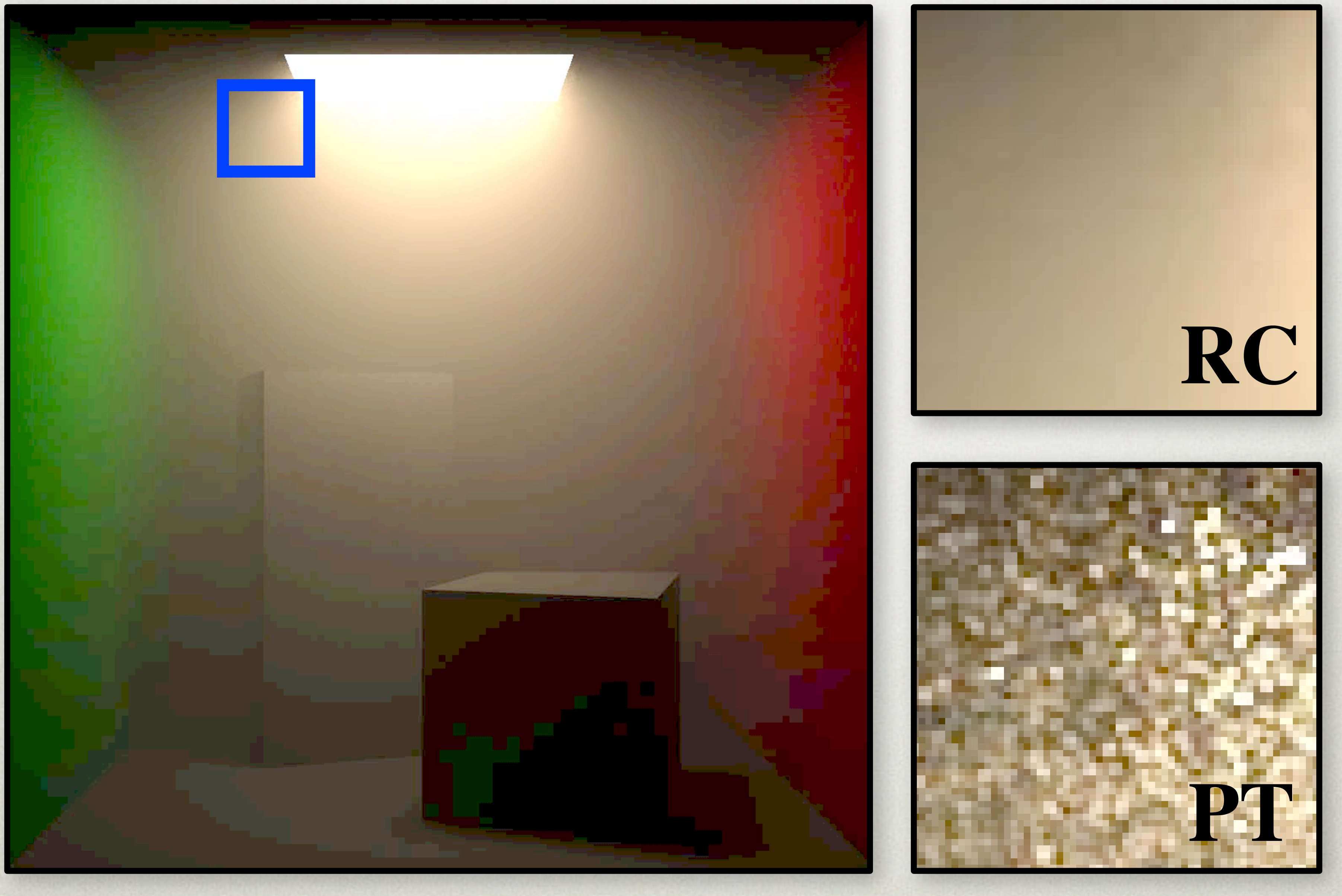
$$L(\mathbf{x}) \approx \exp\left(\frac{\sum_{k \in C} \left(\ln(L_k) + \frac{\nabla L_k}{L_k} \cdot (\mathbf{x} - \mathbf{x}_k) \right) w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)}{\sum_{k \in C} w\left(\|\mathbf{x} - \mathbf{x}_k\|\right)} \right)$$

* given a valid radius for each cache point, in order to compute radiance: interpolate
* weighted average
* smooth weighting function

Results

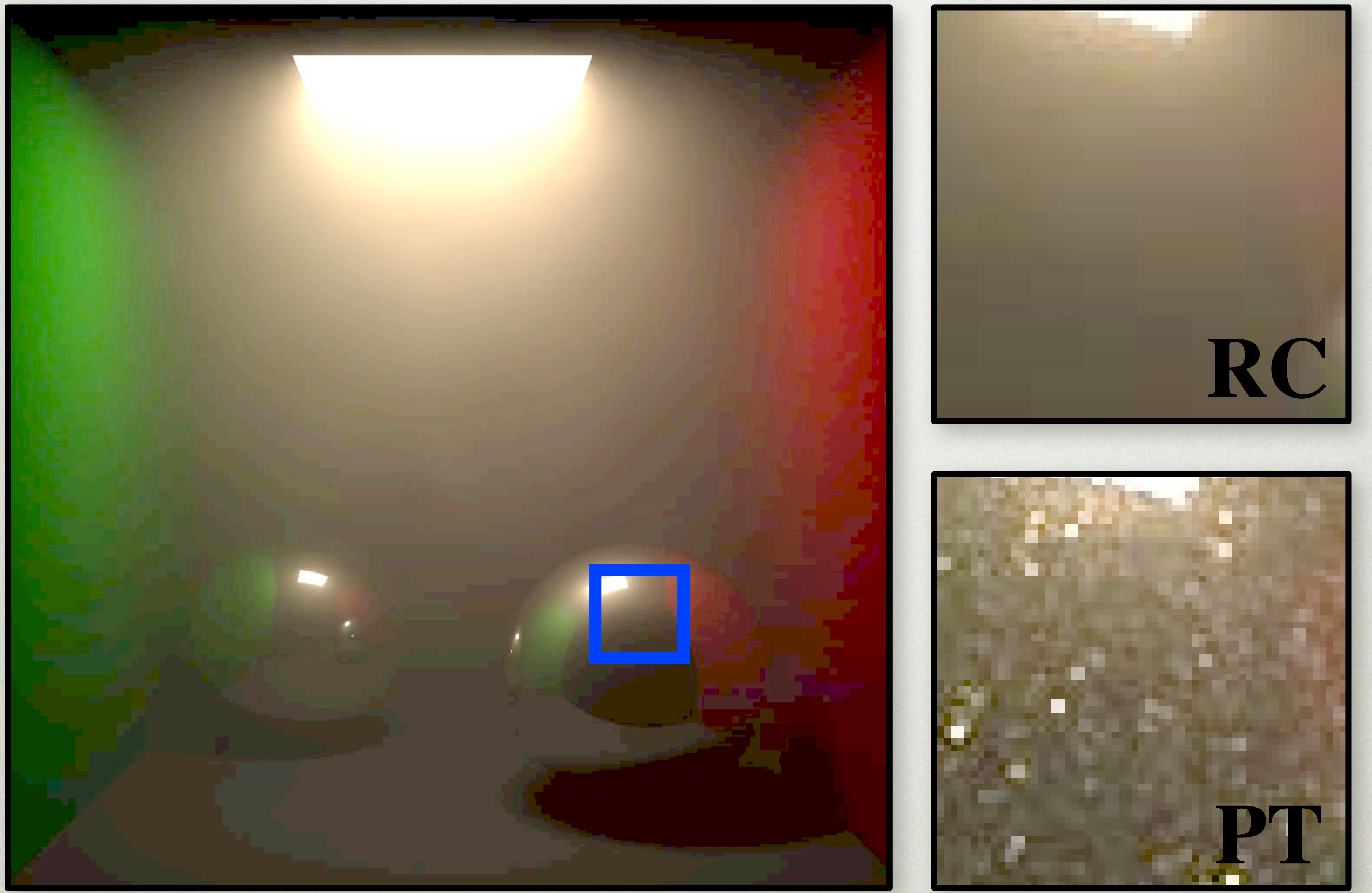
- All results rendered:
  - at 1K horizontal resolution
  - with up to 16 samples per pixel
  - on a Core 2 Duo 2.4 GHz

Results



1.4 minutes

Results

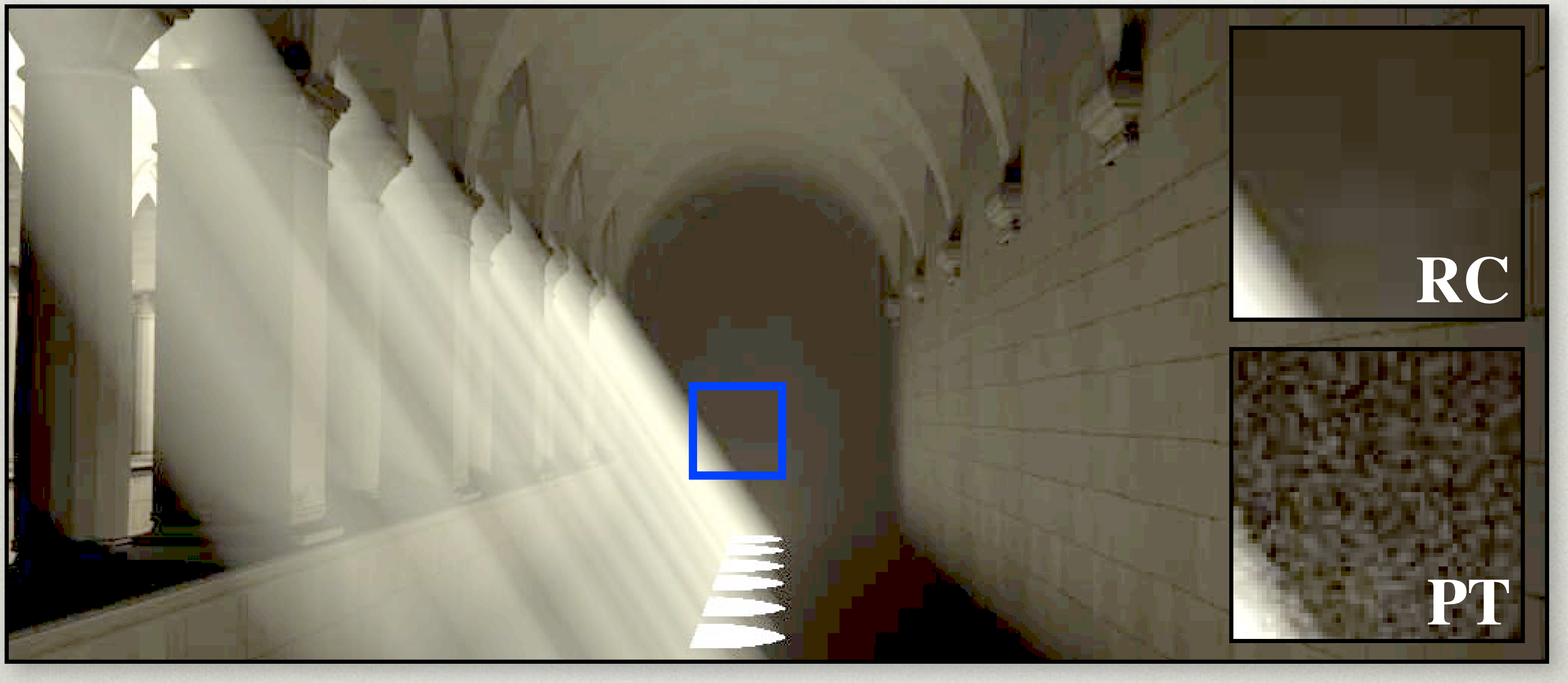


3.6 minutes

Thursday, 6 September 12

* can handle anisotropic media
* project radiance and gradient onto SH
* photon mapping works quite well in contained scenes like this, however...

Results



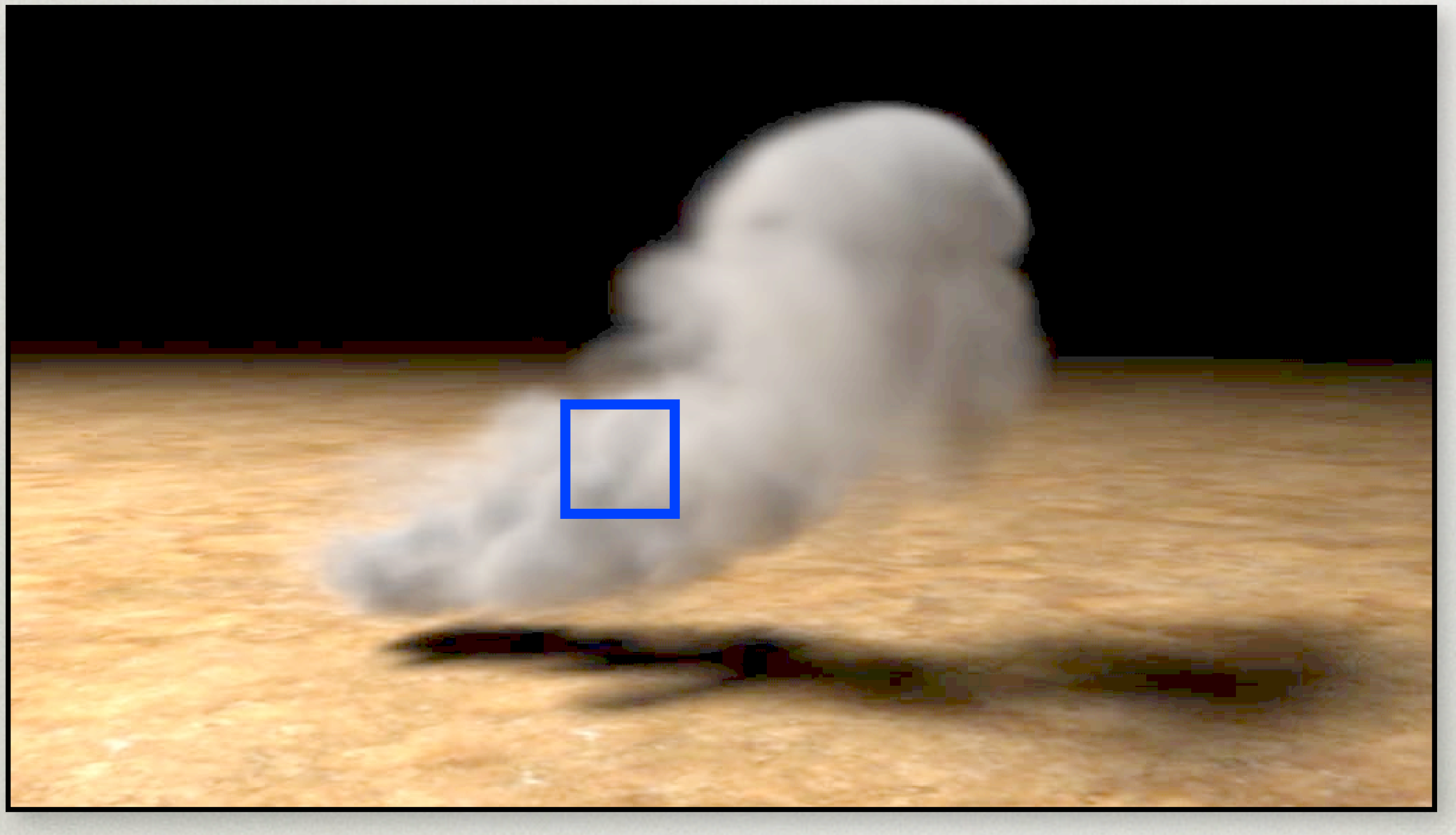
19 minutes

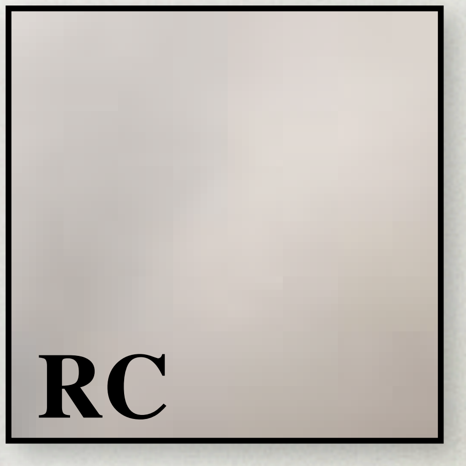
Thursday, 6 September 12

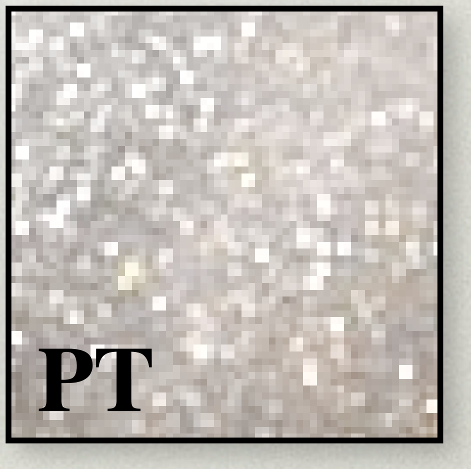
* very difficult for photon mapping
* reuse for walk-through animations

Results

Results





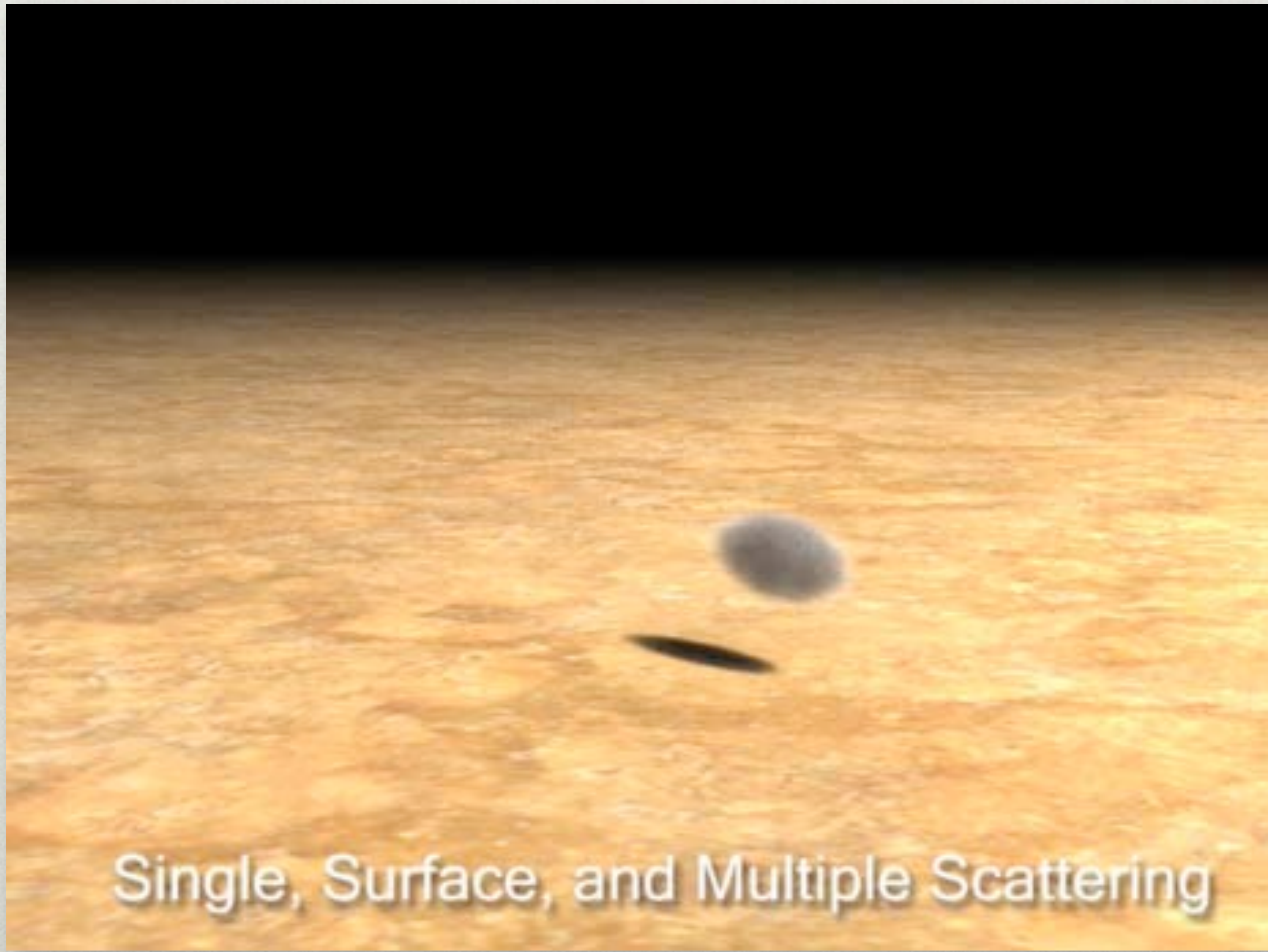


5.8 minutes

Thursday, 6 September 12

* can handle heterogeneous media

Results



Results

20 minutes

Thursday, 6 September 12

* can handle scenes with large extent
* difficult for photon mapping

Results

contrast enhanced

Thursday, 6 September 12

* 8M photons

Contributions

Contributions

- Radiance caching scheme for Part. media:
  - complementary to photon mapping
  - perceptually motivated error metric

Contributions

- Radiance caching scheme for Part. media:
  - complementary to photon mapping
  - perceptually motivated error metric
- Analytic gradient derivations for inscattered radiance:
  - efficient to compute
  - take into account changing properties of medium

Limitations

- Gradient ignores visibility/occlusion changes
- Multiple scattering still costly

Future Work

- Gradient ignores visibility / occlusion changes
  - W. Jarosz, et al. "Irradiance Gradients in the Presence of Participating Media and Occlusions."
- Multiple scattering still costly
  - Terminate recursion using volumetric photon mapping

Thank you