The Beam Radiance Estimate for Volumetric Photon Mapping

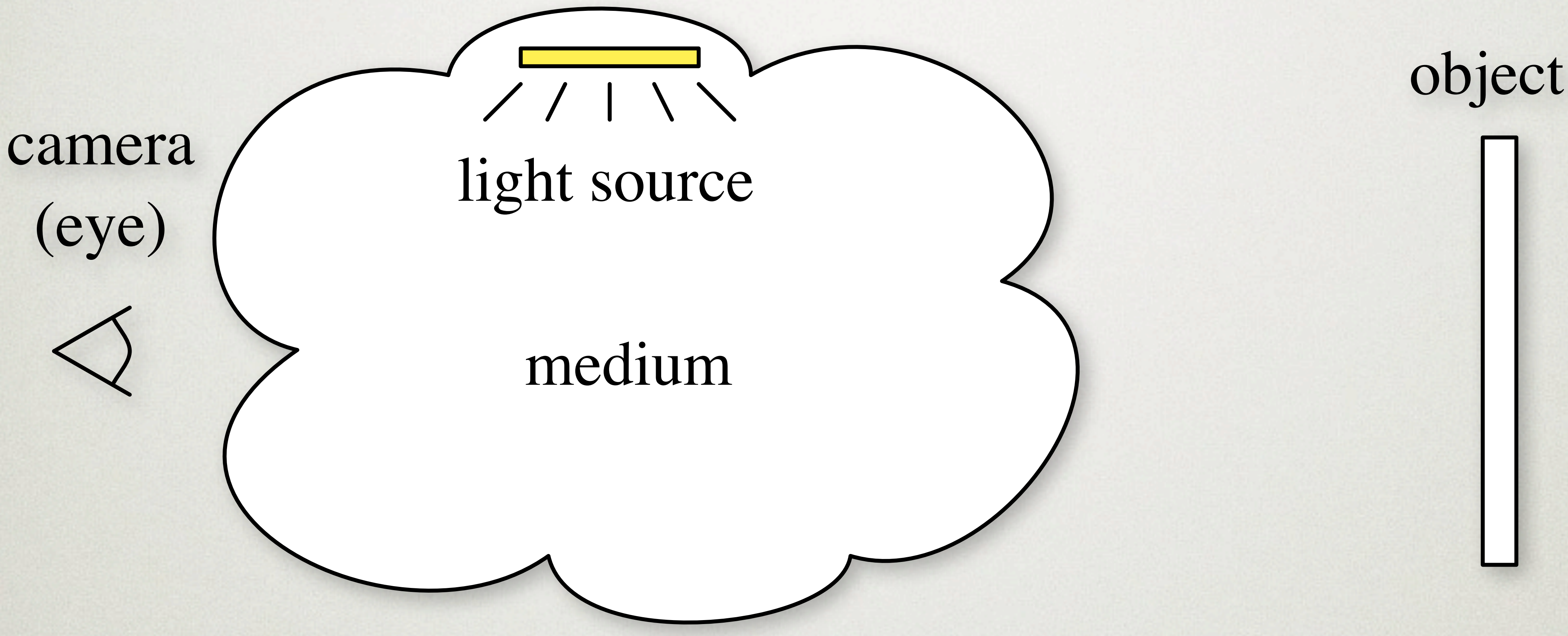
Wojciech Jarosz
in collaboration with
Matthias Zwicker and Henrik Wann Jensen

University of California, San Diego
April 17, 2008

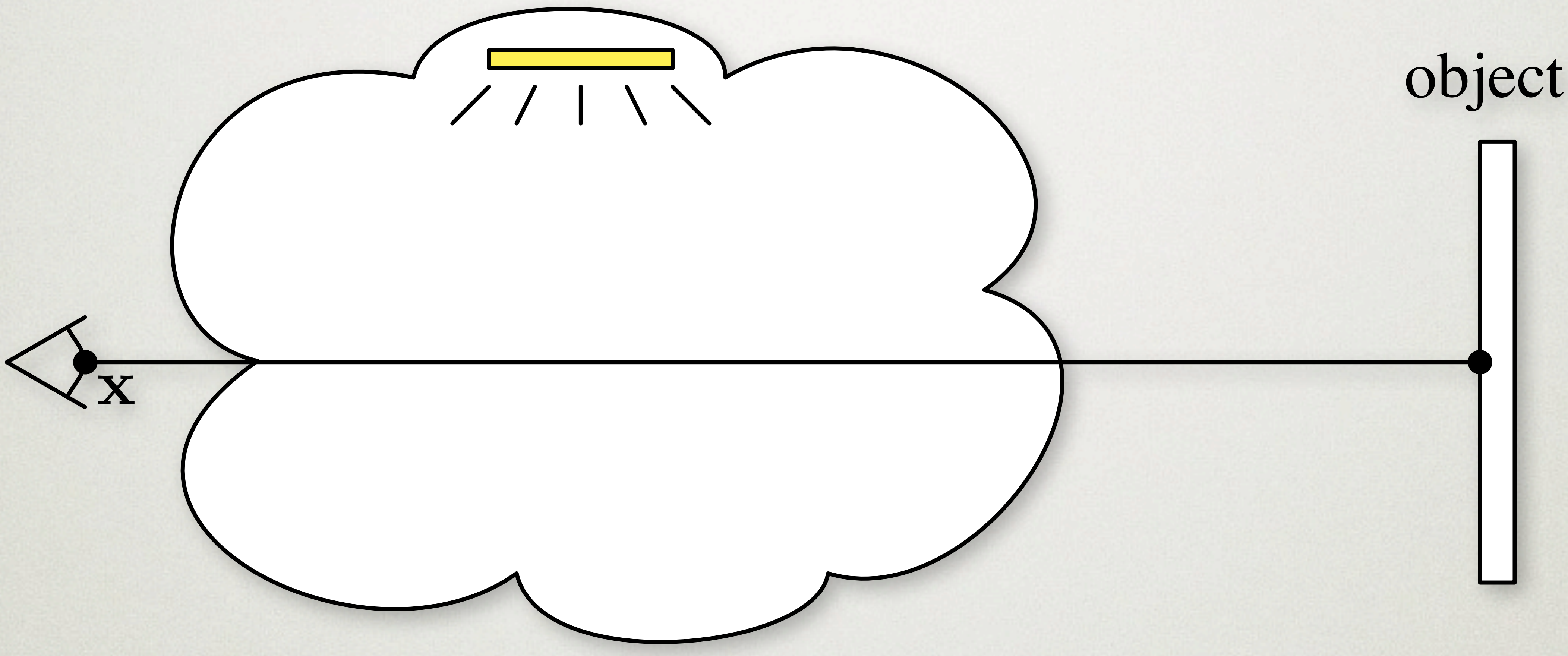
Motivation

* In this talk, we are interested in rendering scene with participating media, or scenes where the volume or medium participates in the lighting interactions.
* These are just a few example photographs of the types of effects that are caused by participating media.

Theoretical Background



Volume Rendering Equation

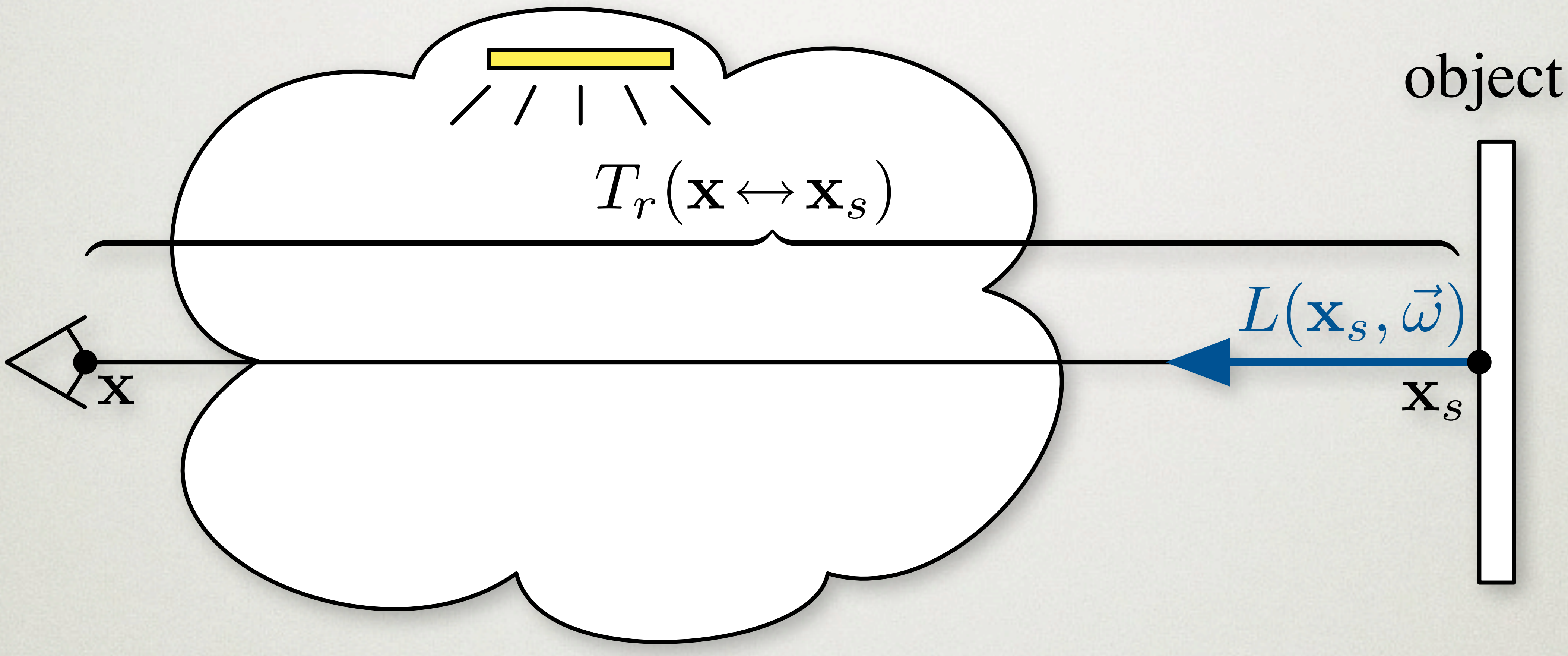


$$L(\mathbf{x},\vec{\omega}) = \int_0^s T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\,dt + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

Thursday, 6 September 12

* The radiance, L, arriving at the eye along a ray can be expressed using the volume rendering equation.
* Now this may seem like a very intimidating and complex equation, and that’s because it is (at least computationally)
* but at a high-level the meaning is pretty simple.
* the radiance arriving at the eye is the sum of two terms:

Volume Rendering Equation

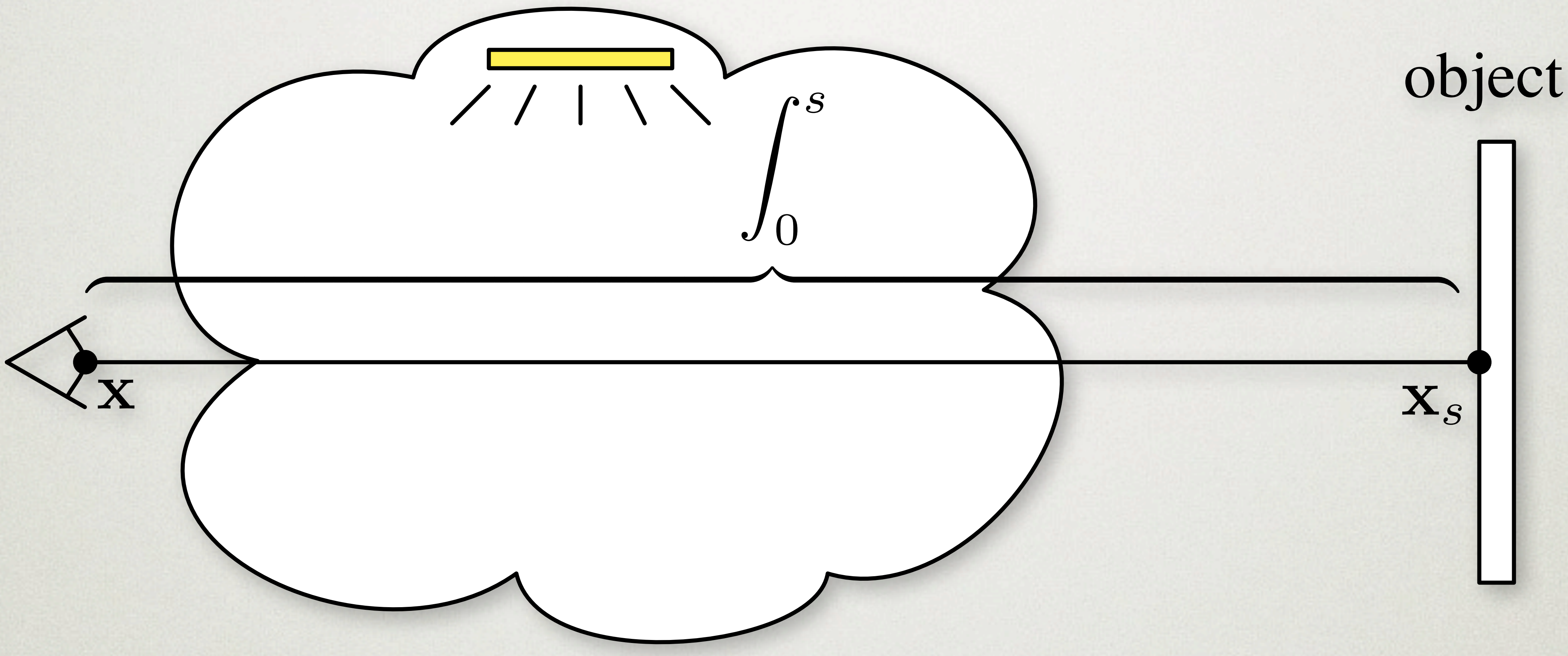


$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

Thursday, 6 September 12

* the right-hand term incorporates lighting arriving from a surface
* before reaching the eye, this radiance must travel through the medium and so is attenuated by a transmission term

Volume Rendering Equation

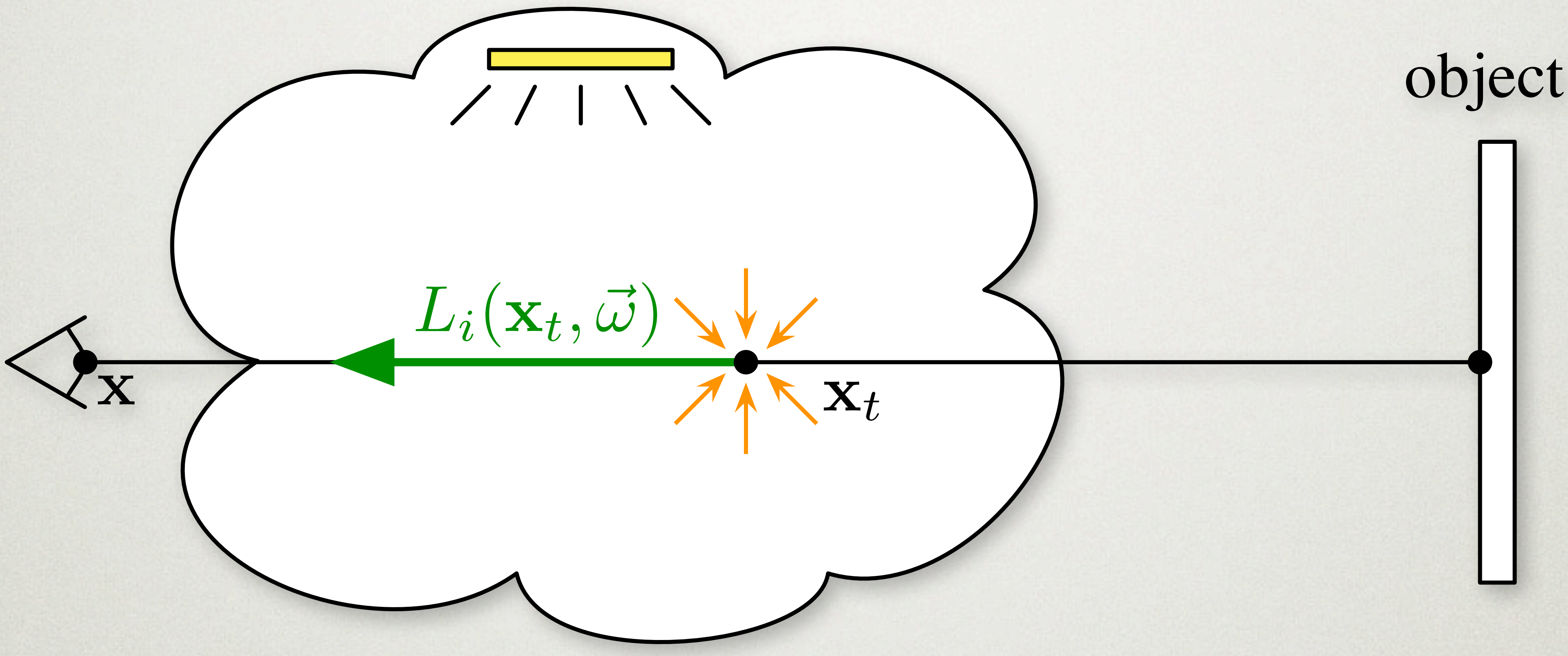


$$L(\mathbf{x}, \vec{\omega}) = \boxed{\int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt} + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

Thursday, 6 September 12

* the left-hand term integrates the scattering of light from the medium along the whole length of the ray

Volume Rendering Equation

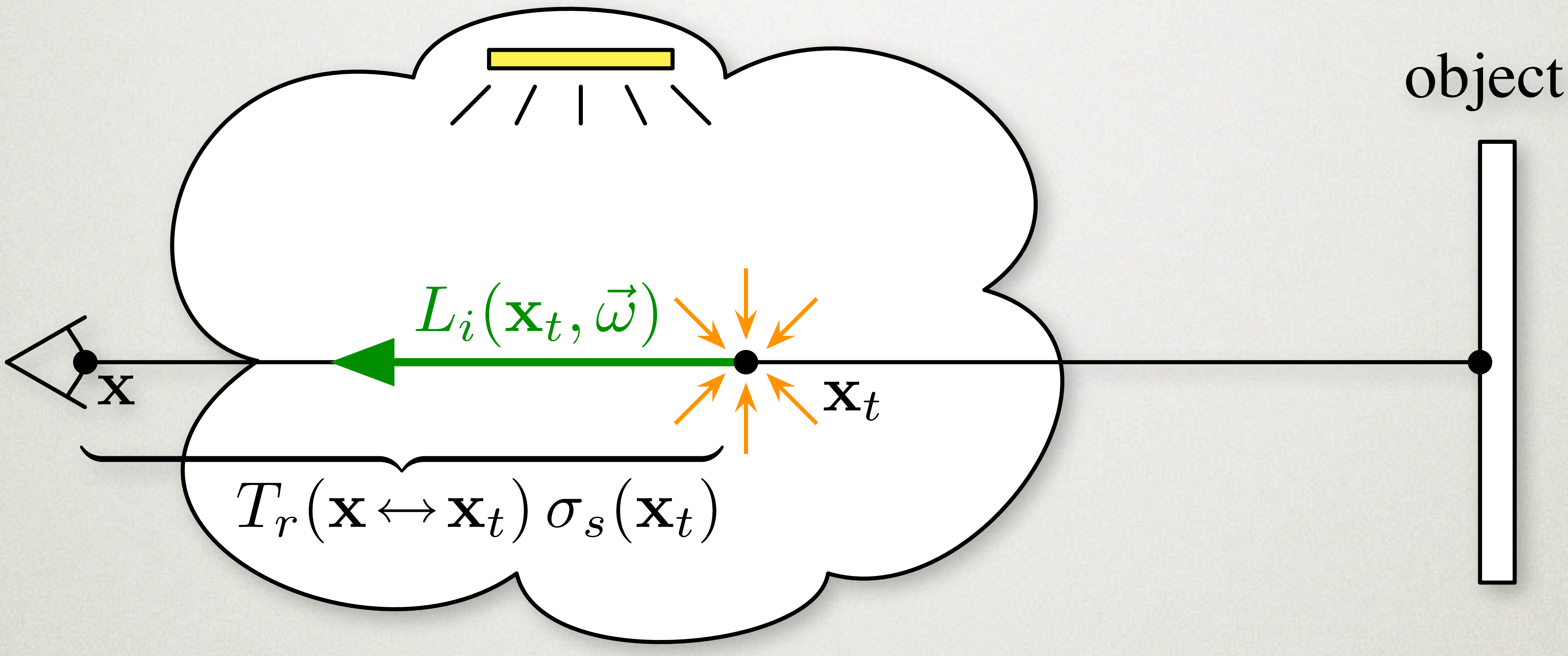


$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t) \boxed{L_i(\mathbf{x}_t, \vec{\omega})}\, dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega})$$

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

* the main quantity that is integrated, Li, is inscattered radiance
* Li itself is an integral. it represents the amount of light that reaches some point in the volume (from any other location in the scene), and then subsequently scatterers towards the eye
* Li this brings about a recursive nature of the volume rendering equation and is extremely expensive to compute

Volume Rendering Equation



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

* as this scattered light travels towards the eye it is also dissipated by extinction through the medium
* this computation is very expensive and there has been a lot of work on how to solve this problem efficiently

Previous Work

Finite Element

- Zonal Method [Rushmeier and Torrance 87; Bhate and Tokuta 92]
- Diffusion [Stam 95]

\- Requires discretization

Monte Carlo

- Path tracing [Kajiya and Herzen 84; Kajiya 86; Lafortune and Willems 96]
- Photon mapping [Jensen and Christensen 1998]
- Metropolis [Pauly, Kollig, and Keller 00]
- Path Integration [Premože 03]

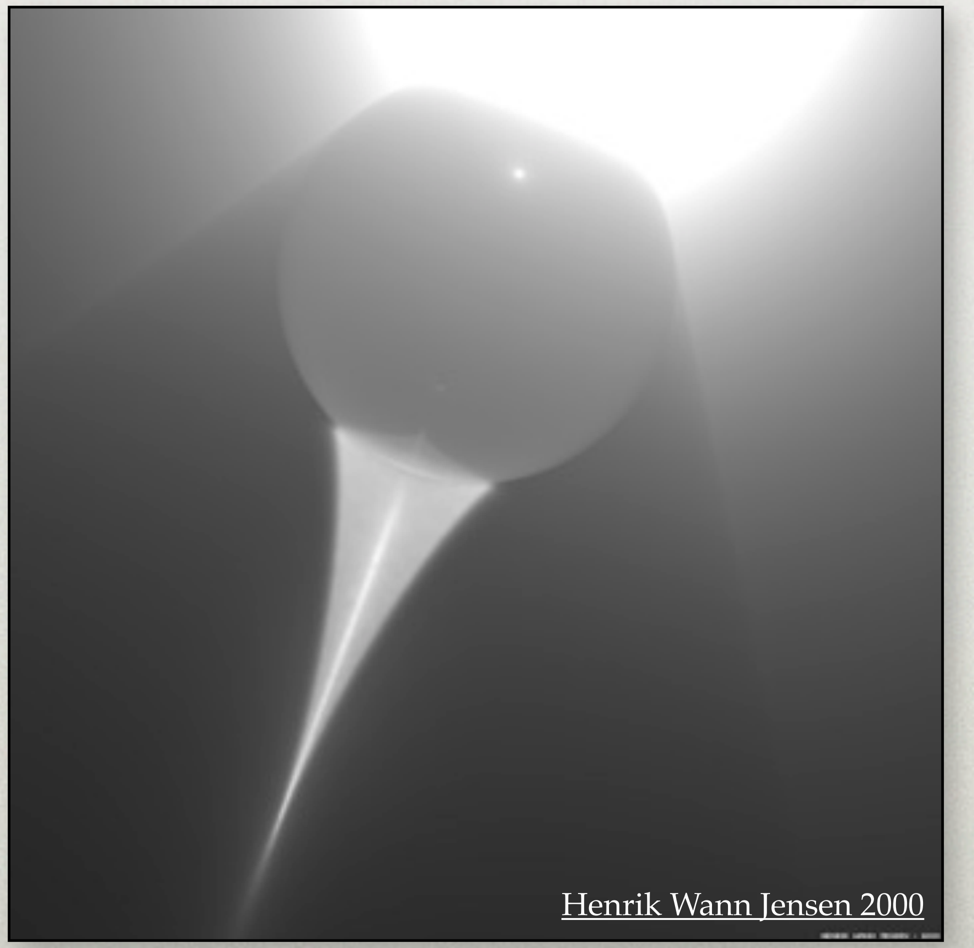
\- Slow convergence/noisy results.

* previous methods can roughly be split up into two main categories.

Previous Work

Monte Carlo

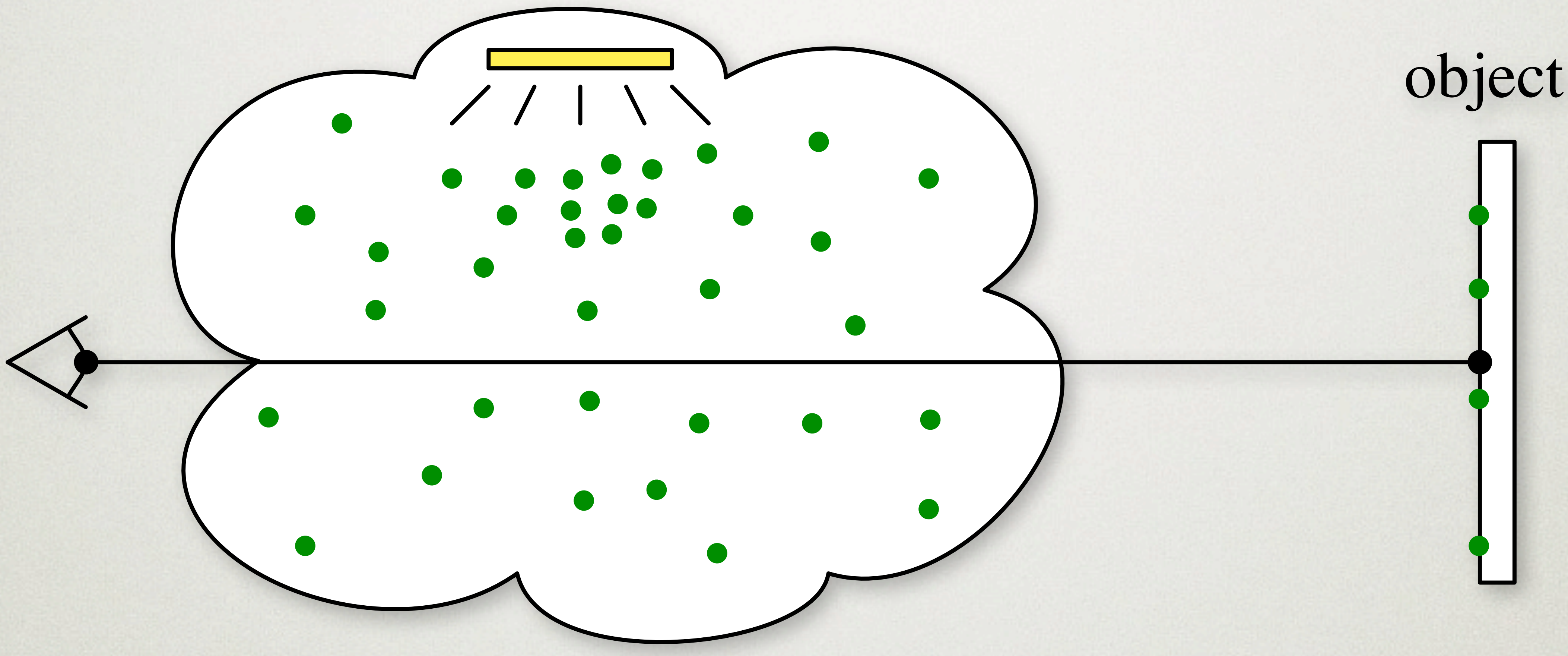
- Photon mapping [Jensen and Christensen 1998]
  - Costly for scenes with large extent



Thursday, 6 September 12

* One of the techniques that has proven more popular is photon mapping

Volumetric Photon Mapping

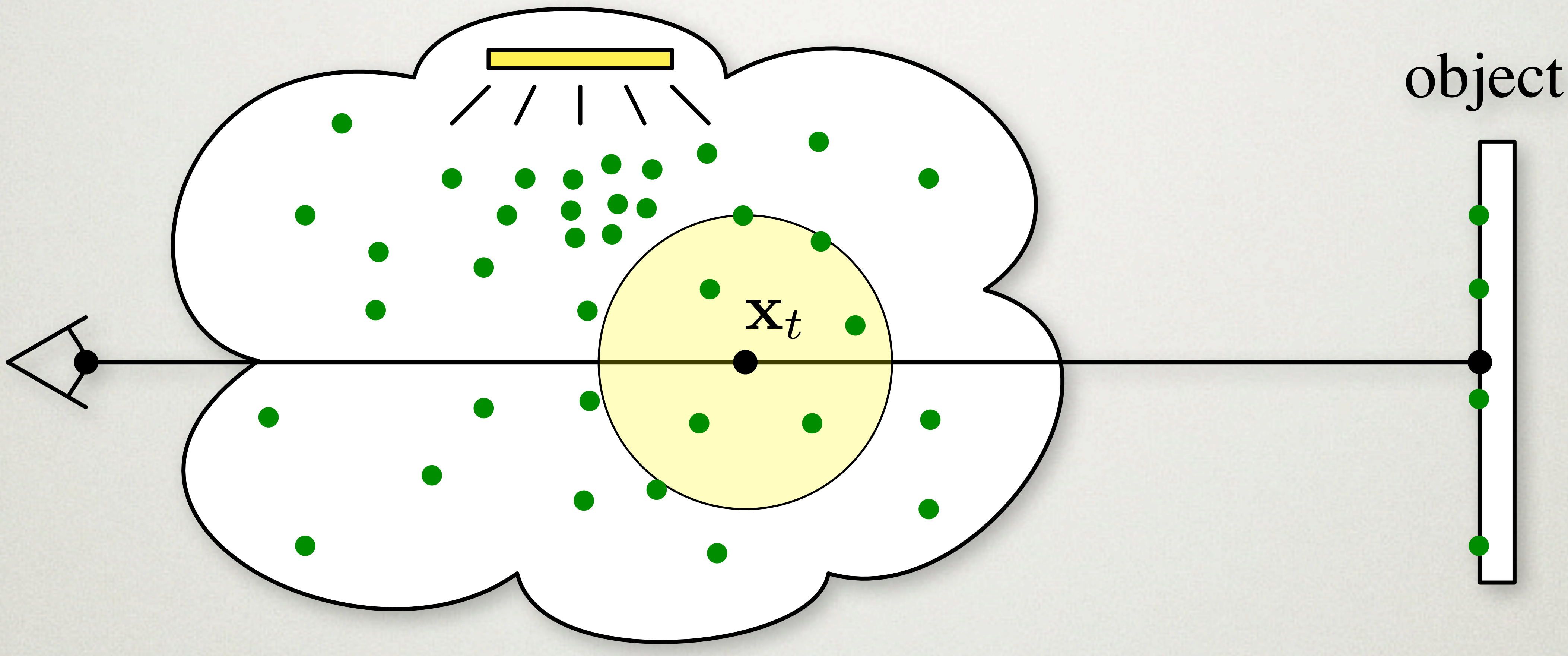


$$L(\mathbf{x},\vec{\omega}) = \int_0^s T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\,dt + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

Thursday, 6 September 12

* volumetric photon mapping starts by shooting photons from light sources
* these photons carry energy and are deposited at surfaces and within the volume at scattering events
* after the photon tracing stage the photon density represents the distribution of radiance within the scene.

Volumetric Photon Mapping

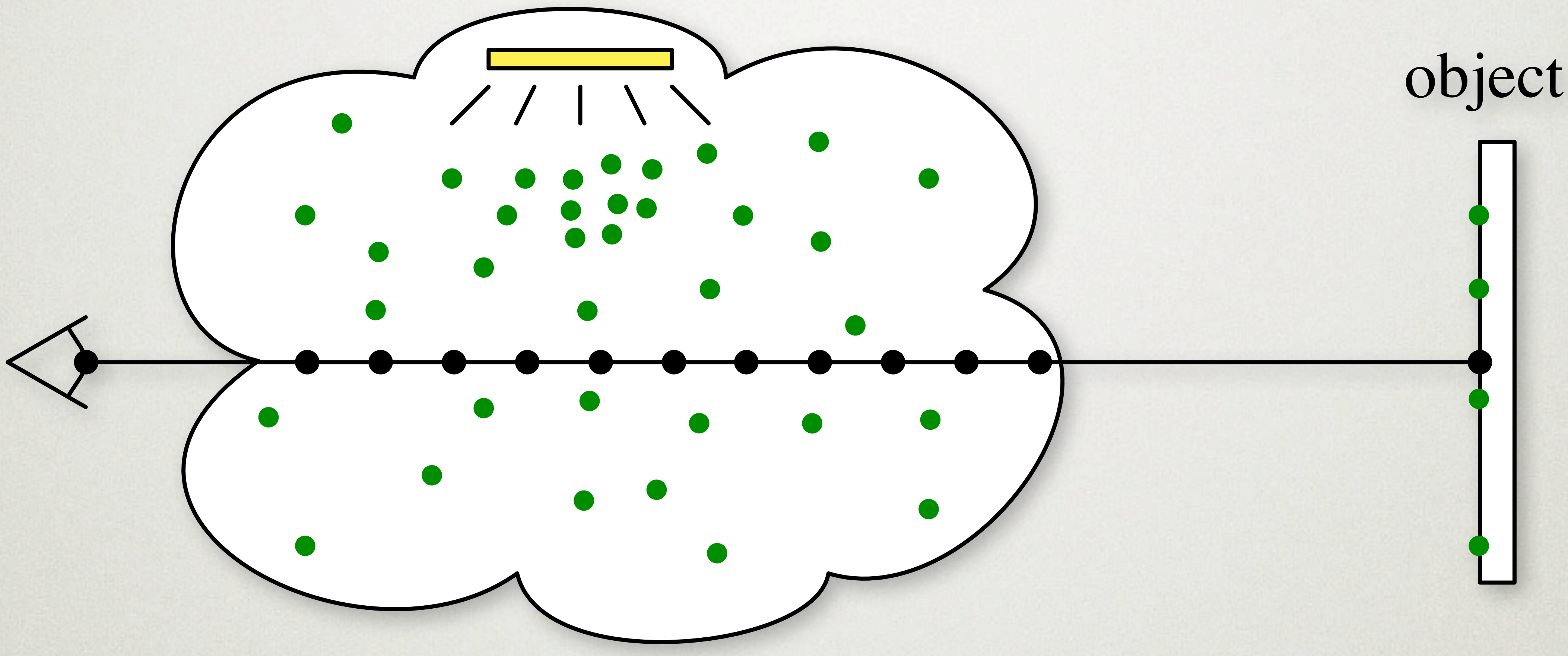


$$L(\mathbf{x},\vec{\omega}) = \int_0^s T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)\boxed{L_i(\mathbf{x}_t,\vec{\omega})}\,dt + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

$L_i(\mathbf{x}_t,\vec{\omega})$ approximated using photon map

* the local information in the photon map is used to efficiently estimate values of inscattered radiance
* at any location within the medium inscattered radiance is computed by taking a local average of the photon energy.
* efficiency is gained by reusing a relatively small collection of photons to compute inscattered radiance at all locations in the image (no new rays need to be traced to compute Li)
* by reusing photons during this process, the lighting is blurred or smoothed out, which reduces high frequency noise, but introduces bias

Volumetric Photon Mapping (Ray Marching)

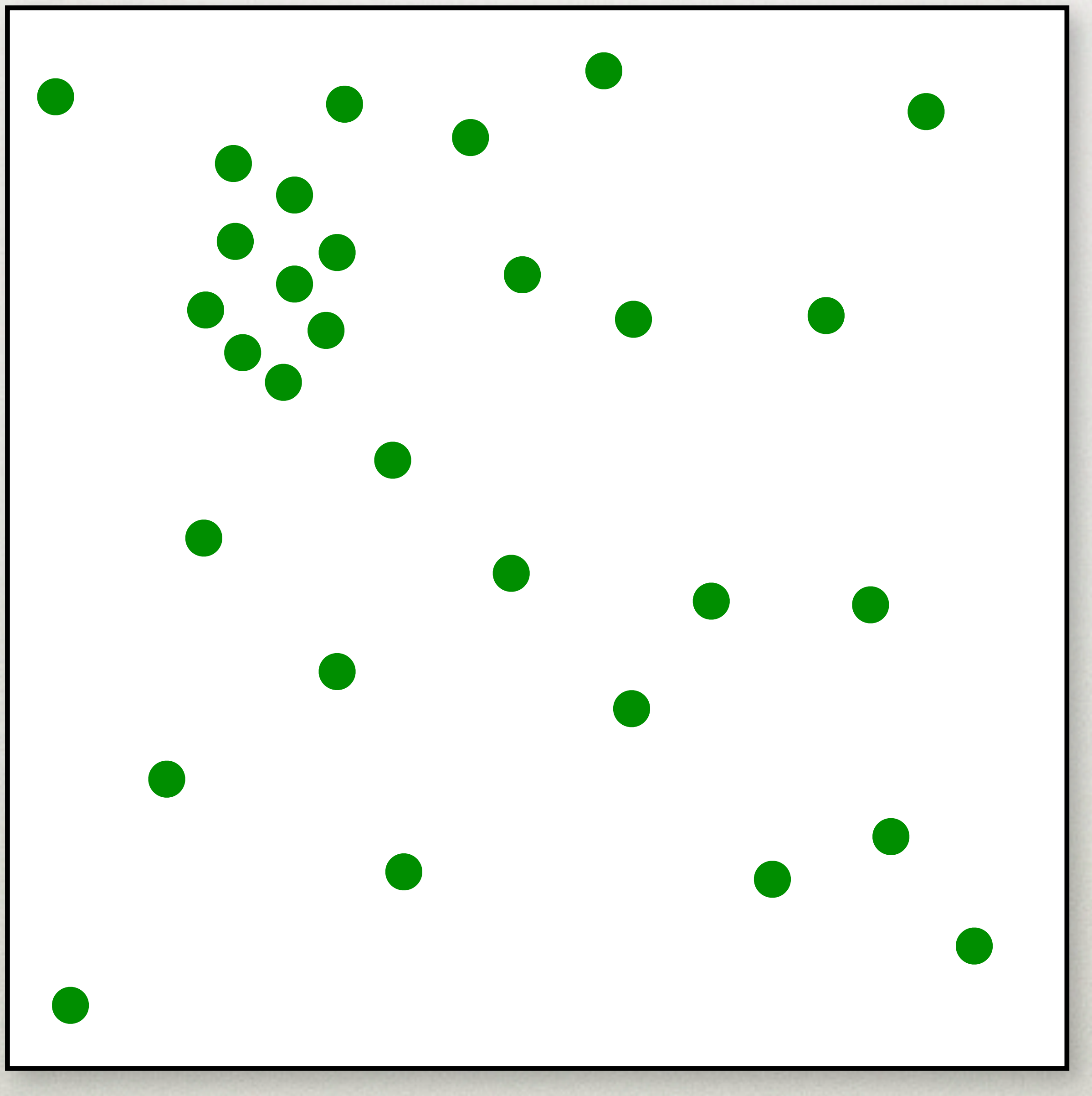


$$L(\mathbf{x},\vec{\omega}) \approx \sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

* However, in order to approximate the integral along the ray, photon mapping uses ray marching.
* ray marching is a 1D numerical integration technique which is computed by taking small steps along the ray and evaluating the inscattered radiance at each discrete step.

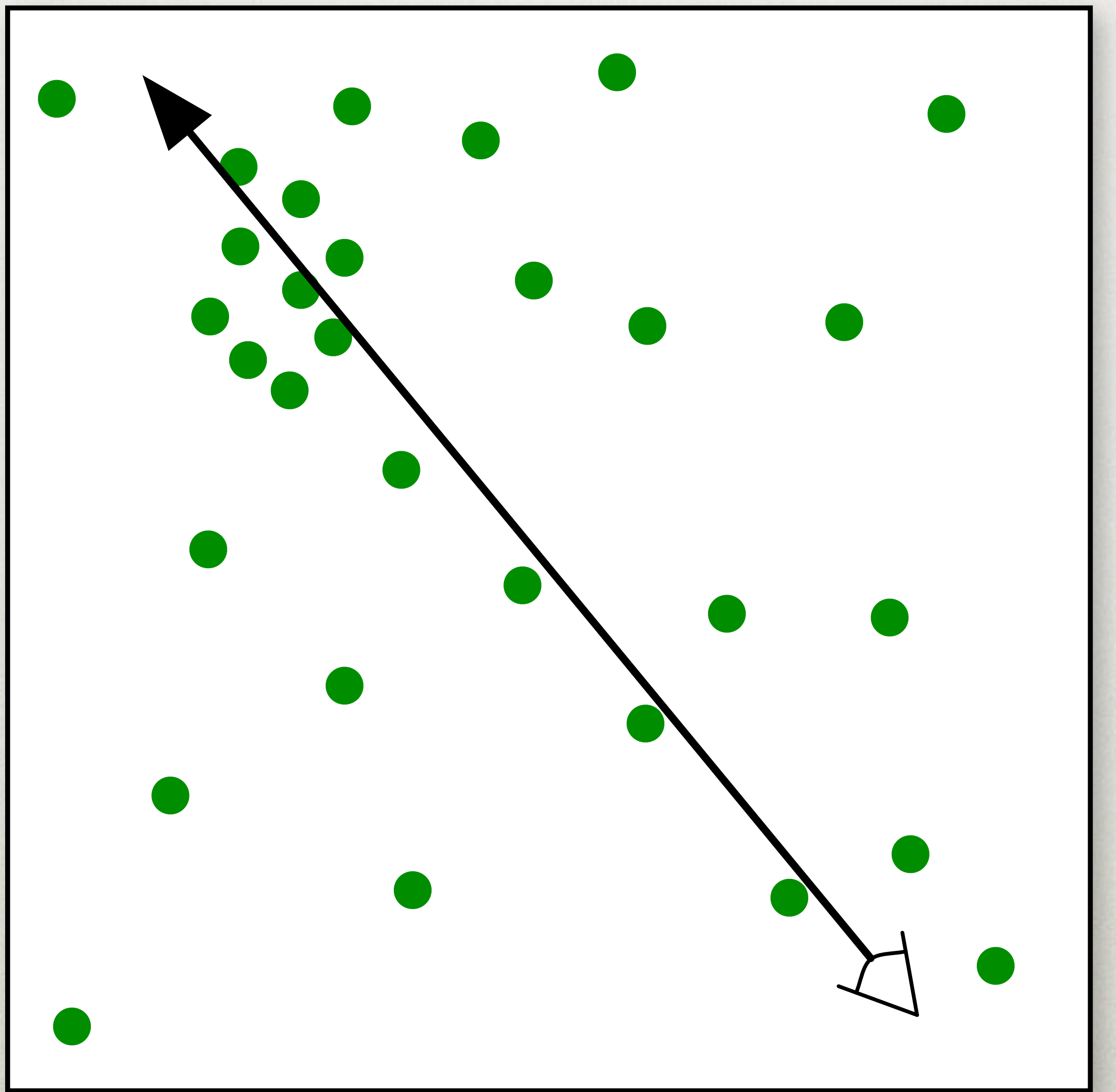
Volumetric Photon Mapping

Conventional Radiance Estimate



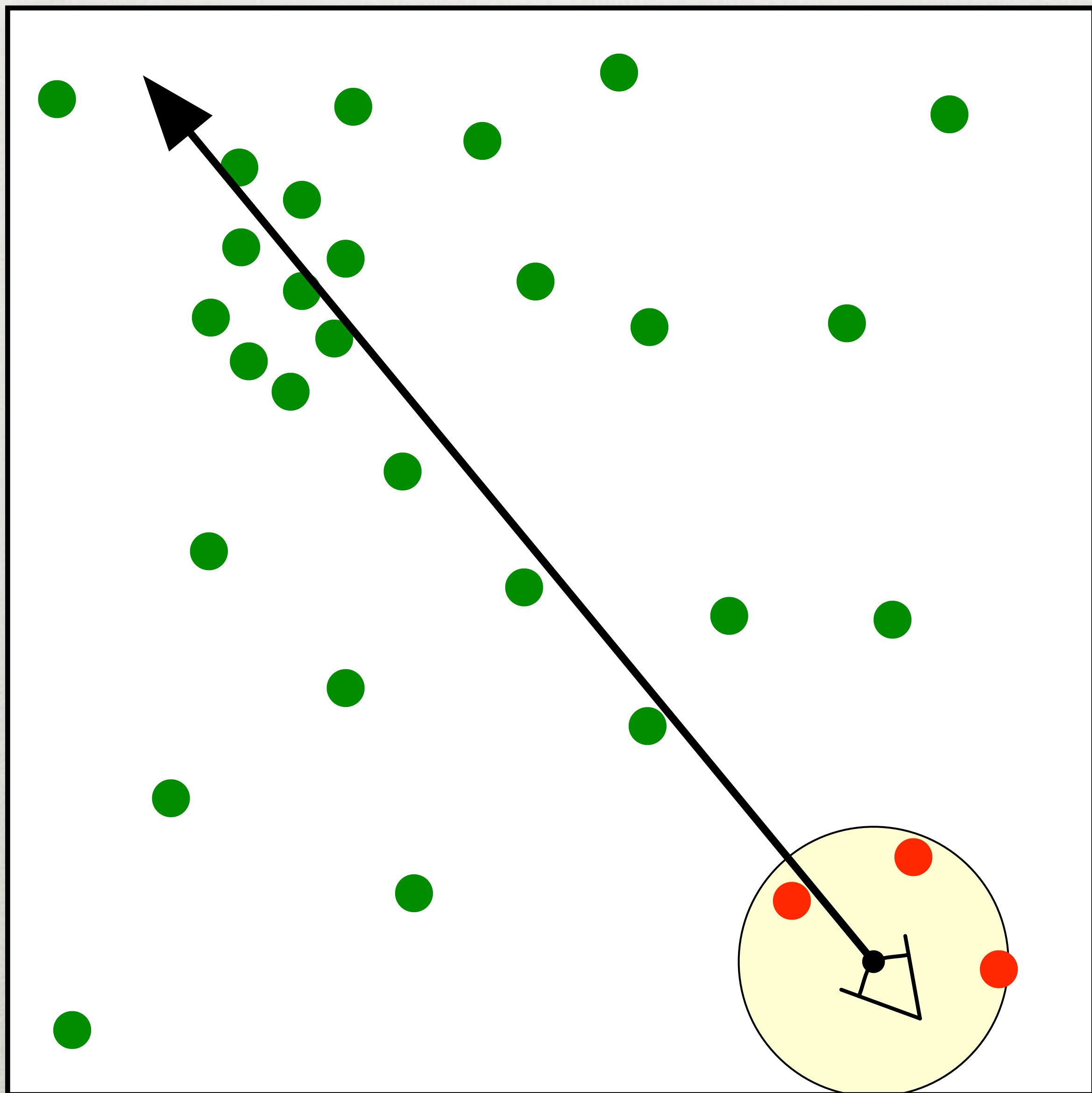
Volumetric Photon Mapping

Conventional Radiance Estimate



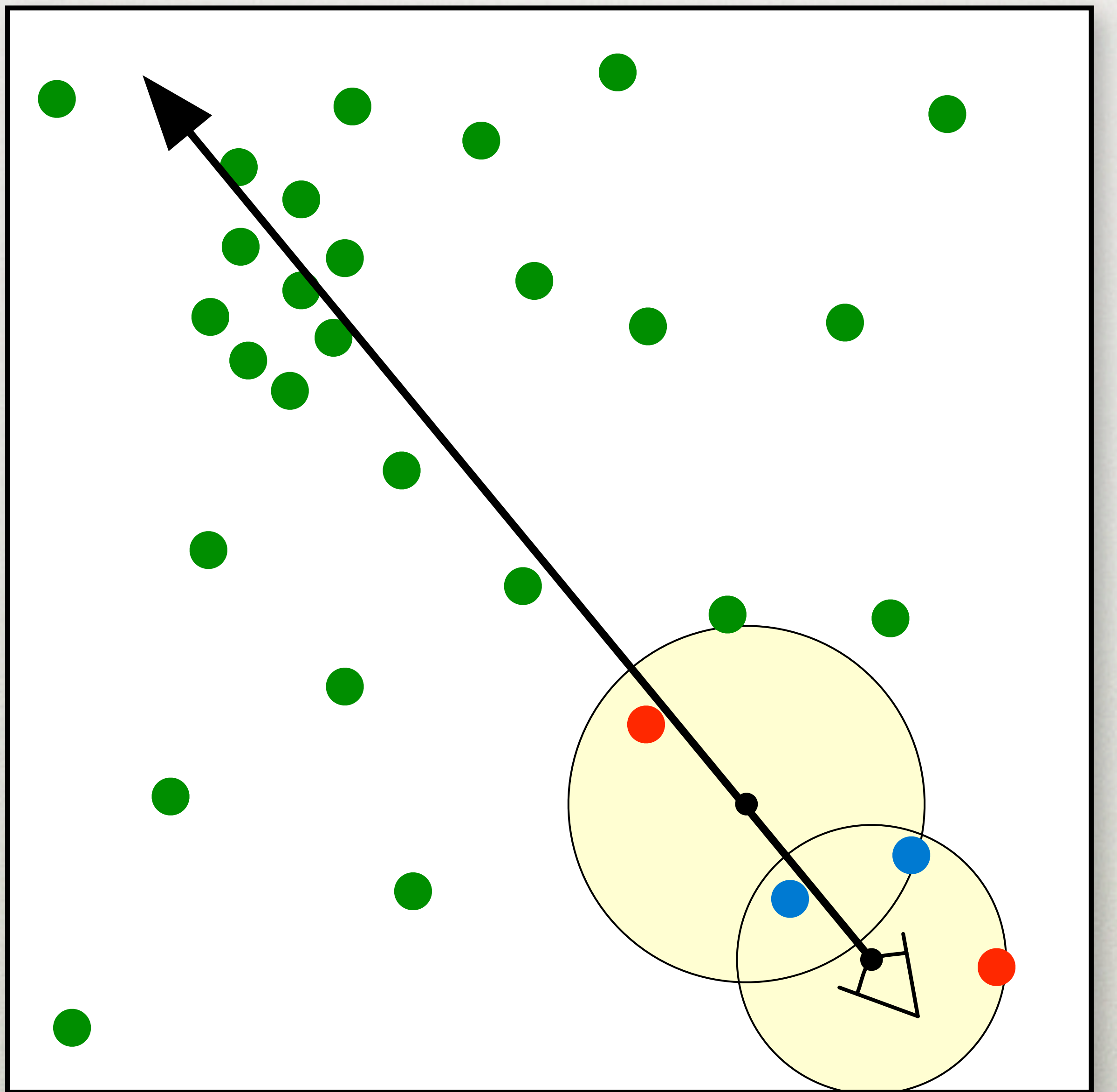
Volumetric Photon Mapping

Conventional Radiance Estimate



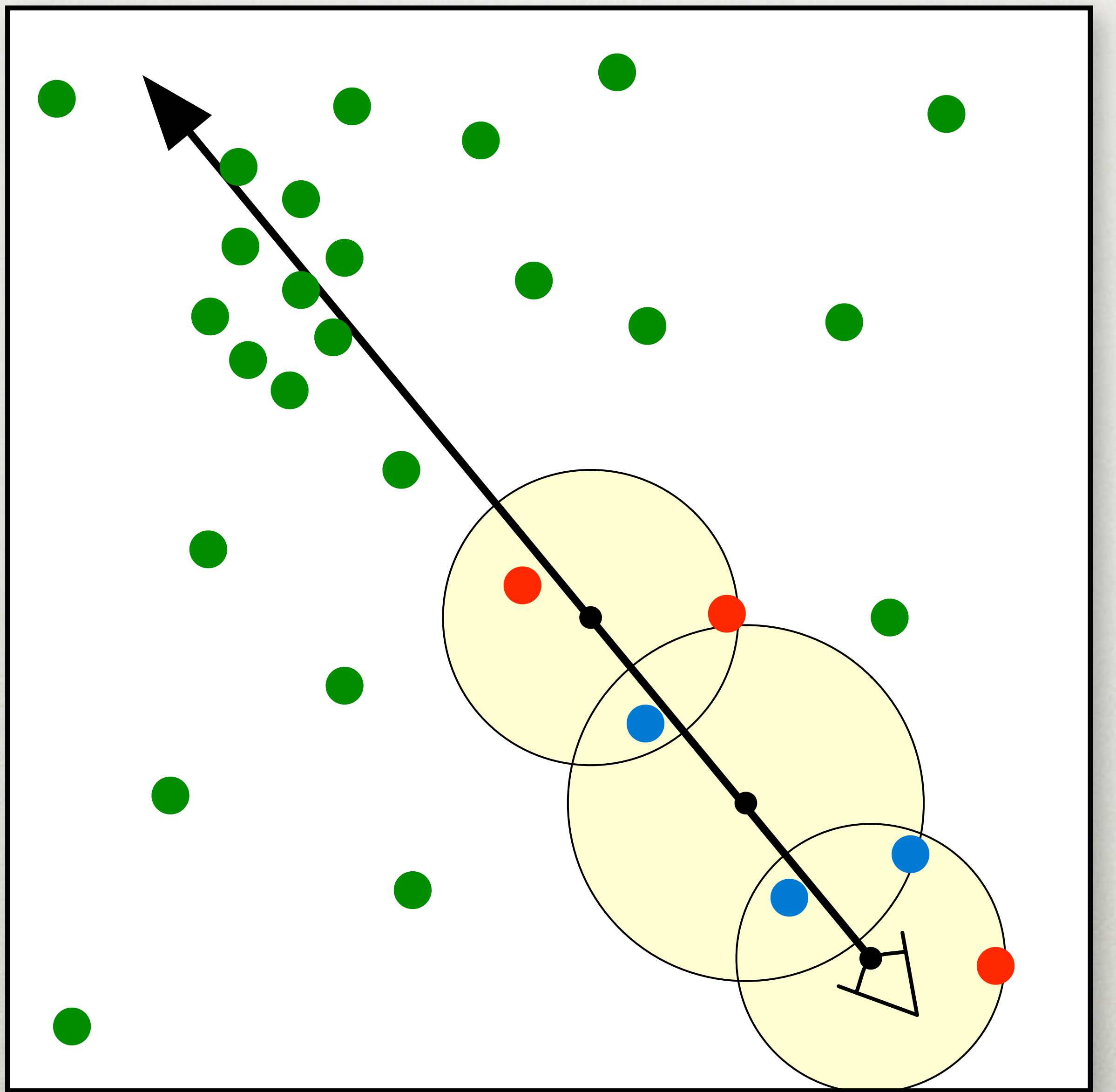
Volumetric Photon Mapping

Conventional Radiance Estimate



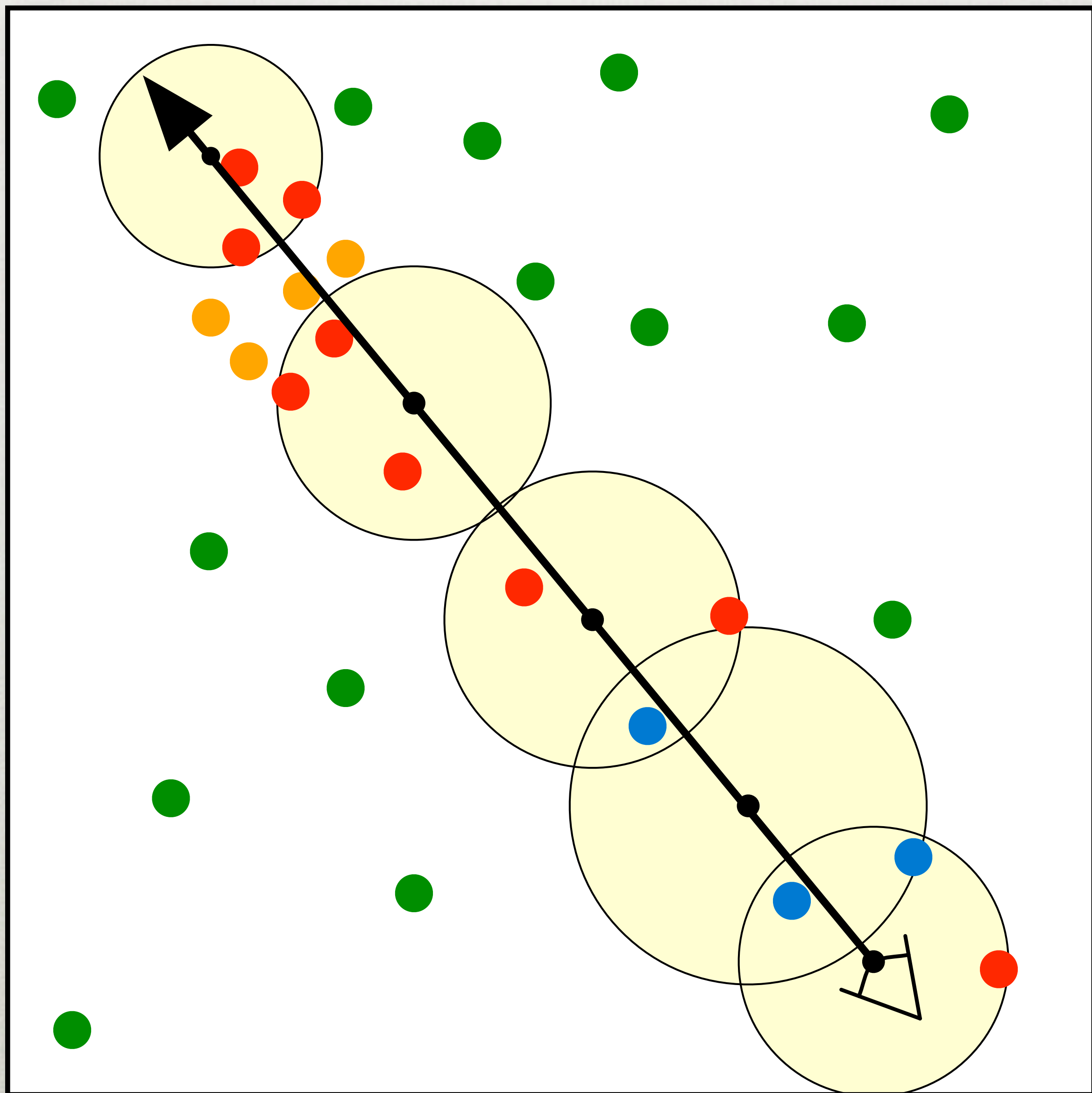
Volumetric Photon Mapping

Conventional Radiance Estimate

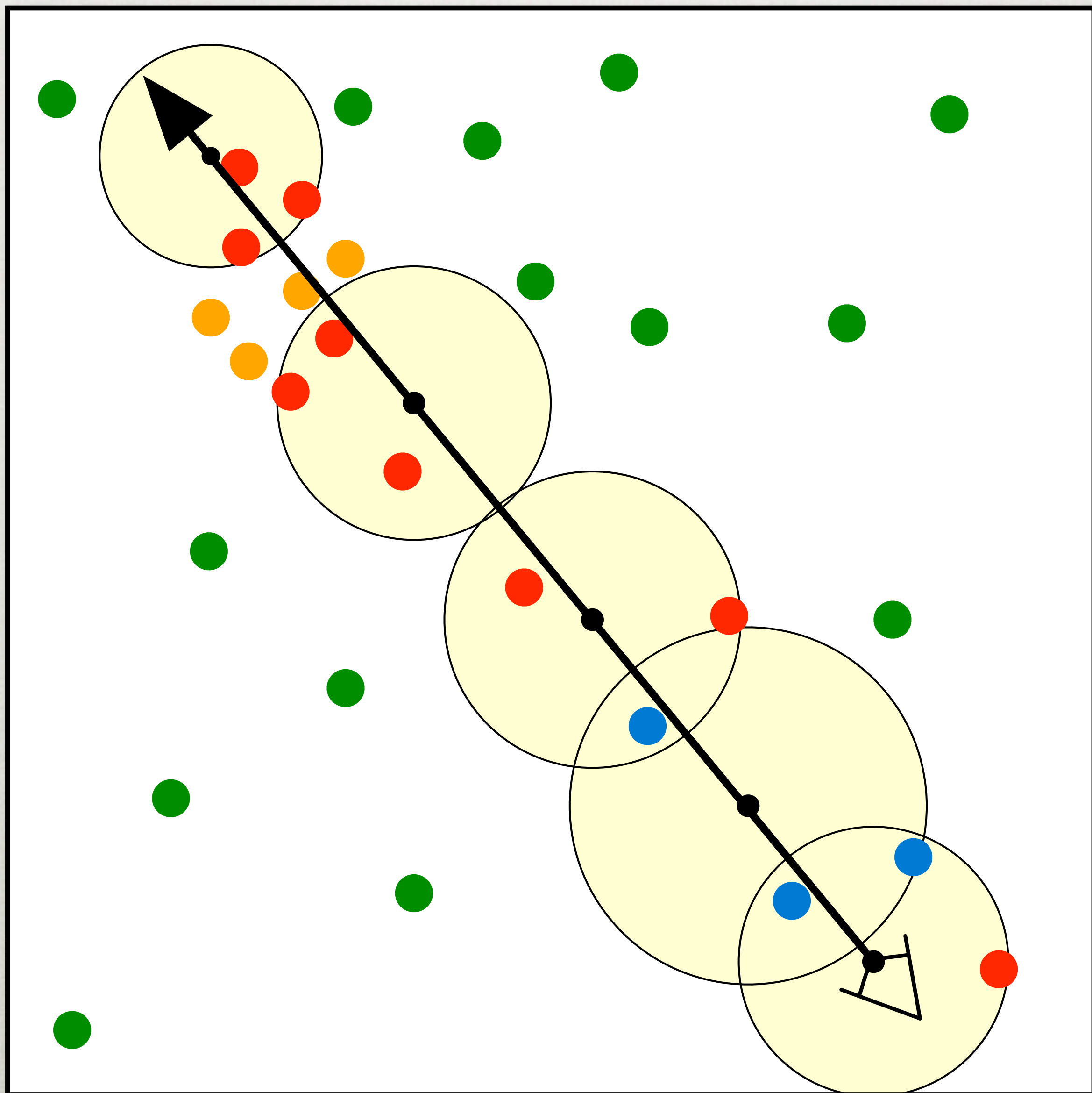


Volumetric Photon Mapping

Conventional Radiance Estimate



Drawbacks

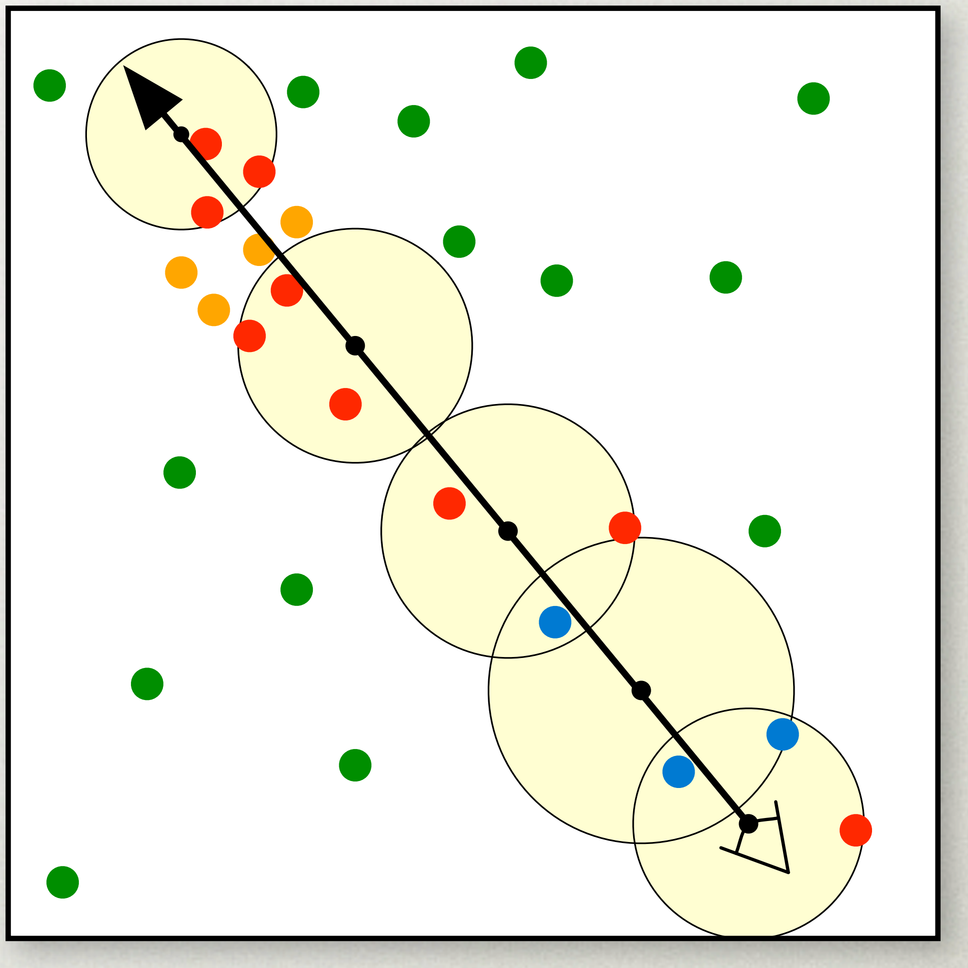


Thursday, 6 September 12

* if the step size is too small, then we may find the same photons multiple times (shown in blue)
* if the step size is too big, we miss features (as shown in orange).

Drawbacks

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



Thursday, 6 September 12

* if the step size is too small, then we may find the same photons multiple times (shown in blue)
* if the step size is too big, we miss features (as shown in orange).

Drawbacks

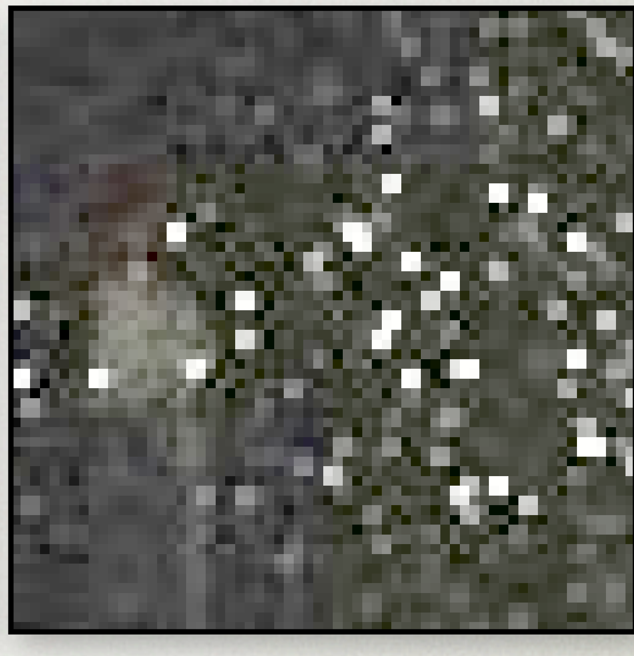
Large Step-size

Thursday, 6 September 12

* the way this manifests itself in renderings is high-frequency noise
* with a large step-size we may completely jump over the narrow lighthouse beam
* there is a tension between efficiency and noise in setting this parameters

Drawbacks

Large Step-size

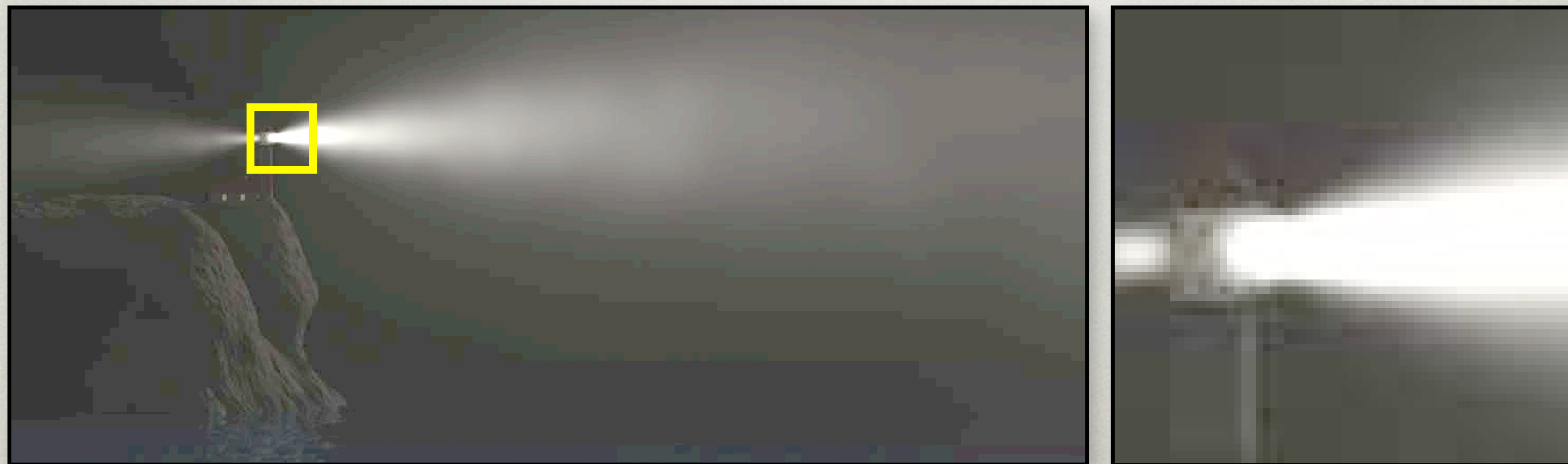


Thursday, 6 September 12

* the way this manifests itself in renderings is high-frequency noise
* with a large step-size we may completely jump over the narrow lighthouse beam
* there is a tension between efficiency and noise in setting this parameters

Drawbacks

Large Step-size



Very Small Step-size

Thursday, 6 September 12

* the way this manifests itself in renderings is high-frequency noise
* with a large step-size we may completely jump over the narrow lighthouse beam
* there is a tension between efficiency and noise in setting this parameters

Goal

- Render high-quality, noise-free images using photon mapping, ***faster***.

Goal

- Render high-quality, noise-free images using photon mapping, ***faster***.
  - Eliminate ray marching by finding ***all*** photons which contribute to the ***entire*** length of a ray.

Goal

- Need to solve:
  - How do we *find* photons?
  - How do we *use* photons?

* Find all photons which contribute to the entire length of a ray.
* Given all photons along ray, how do we use them to compute a radiance estimate?

Outline

- Need to solve:

  2) How do we *find* photons?

  1) How do we *use* photons?

* I'll cover these in reverse order.
* This will require some equations (to convince you that I didn't just make this up)

Outline

- Need to solve:

  2) How do we *find* photons?

  1) How do we *use* photons?

3) Render pretty pictures! (quickly)

* I'll cover these in reverse order.
* This will require some equations (to convince you that I didn't just make this up)

Outline

- Need to solve:

  2) How do we *find* photons?

  1) How do we *use* photons?

3) Render pretty pictures! (quickly)

* I’ll cover these in reverse order.
* This will require some equations (to convince you that I didn’t just make this up)

Approach

- High level description of photon mapping is intuitive, but difficult to generalize
- Theoretical reformulation of volumetric photon mapping
  - Using the Measurement Equation
  - More flexible

* this reformulation allows us to mathematically express higher level radiometric quantities
* for instance, if we don't just want the radiance at a point, but want the total flux on a surface, or the accumulate radiance along a line.
* and it shows us how to estimate these values using the photon map.

Measurement Equation

- Radiance, $L(\mathbf{x}, \vec{\omega})$, is a 5D function over position, $\mathbf{x}$, and direction, $\vec{\omega}$.

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

- Radiance, $L(\mathbf{x}, \vec{\omega})$, is a 5D function over position, $\mathbf{x}$, and direction, $\vec{\omega}$.
- A ***measurement*** is a weighted integral of radiance:

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x})$$

Thursday, 6 September 12

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

- Radiance, $L(\mathbf{x}, \vec{\omega})$, is a 5D function over position, $\mathbf{x}$, and direction, $\vec{\omega}$.
- A ***measurement*** is a weighted integral of radiance:

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x}) = \langle W_e, L \rangle$$

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x}) = \langle W_e, L \rangle$$

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x}) = \langle W_e, L \rangle$$

- Many global illumination algorithms can be expressed this way

Thursday, 6 September 12

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x}) = \langle W_e, L \rangle$$

- Many global illumination algorithms can be expressed this way
  - path tracing

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x}) = \langle W_e, L \rangle$$

- Many global illumination algorithms can be expressed this way
  - path tracing
  - radiosity

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Measurement Equation

$$I = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}, \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, d\vec{\omega} \, d\mathcal{V}(\mathbf{x}) = \langle W_e, L \rangle$$

- Many global illumination algorithms can be expressed this way
  - path tracing
  - radiosity
  - particle tracing [Veach98]

* concisely written as an inner product between the radiance field and a weighting function
* the weighting function is typically non-zero only within a small region of the whole domain

Photon Mapping as a Measurement

- Photon tracing generates N weighted sample rays, or photons $(\alpha_i, \mathbf{x}_i, \vec{\omega}_i)$
  - $(\mathbf{x}_i, \vec{\omega}_i)$: ray
  - α_i: corresponding weight

Thursday, 6 September 12

* Veach showed that given certain constraints on how the photons are distributed, unbiased measurements can be estimated as a weighted sum
* Veach showed this for particle tracing on surfaces, and we extend his derivation to include participating media
* Arbitrary measurements can be computed using the photon map

Photon Mapping as a Measurement

- Photon tracing generates N weighted sample rays, or photons $(\alpha_i, \mathbf{x}_i, \vec{\omega}_i)$
  - $(\mathbf{x}_i, \vec{\omega}_i)$: ray
  - α_i: corresponding weight
- ***Unbiased*** measurements can be estimated as a weighted sum of photons:

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

* Veach showed that given certain constraints on how the photons are distributed, unbiased measurements can be estimated as a weighted sum
* Veach showed this for particle tracing on surfaces, and we extend his derivation to include participating media
* Arbitrary measurements can be computed using the photon map

Photon Mapping as a Measurement

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

* Veach showed that given certain constraints on how the photons are distributed, unbiased measurements can be estimated as a weighted sum
* Veach showed this for particle tracing on surfaces, and we extend his derivation to include participating media
* Arbitrary measurements can be computed using the photon map

Conventional Radiance Estimate

- Veach showed that the conventional radiance estimate (for surfaces) is a measurement, where $W_e(\mathbf{x}_i, \vec{\omega}_i)$ blurs photon contributions across surfaces.
  - Also true for conventional *volumetric* radiance estimate, but blurs within volume.

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Arbitrary Measurements Using the Photon Map

- However, any **arbitrary** weighting function $W_e(\mathbf{x}_i, \vec{\omega}_i)$ can be used to compute a different **measurement**.

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

* if we can represent the quantity we want to compute as a measurement, then we can compute estimates of that quantity using the measurement equation

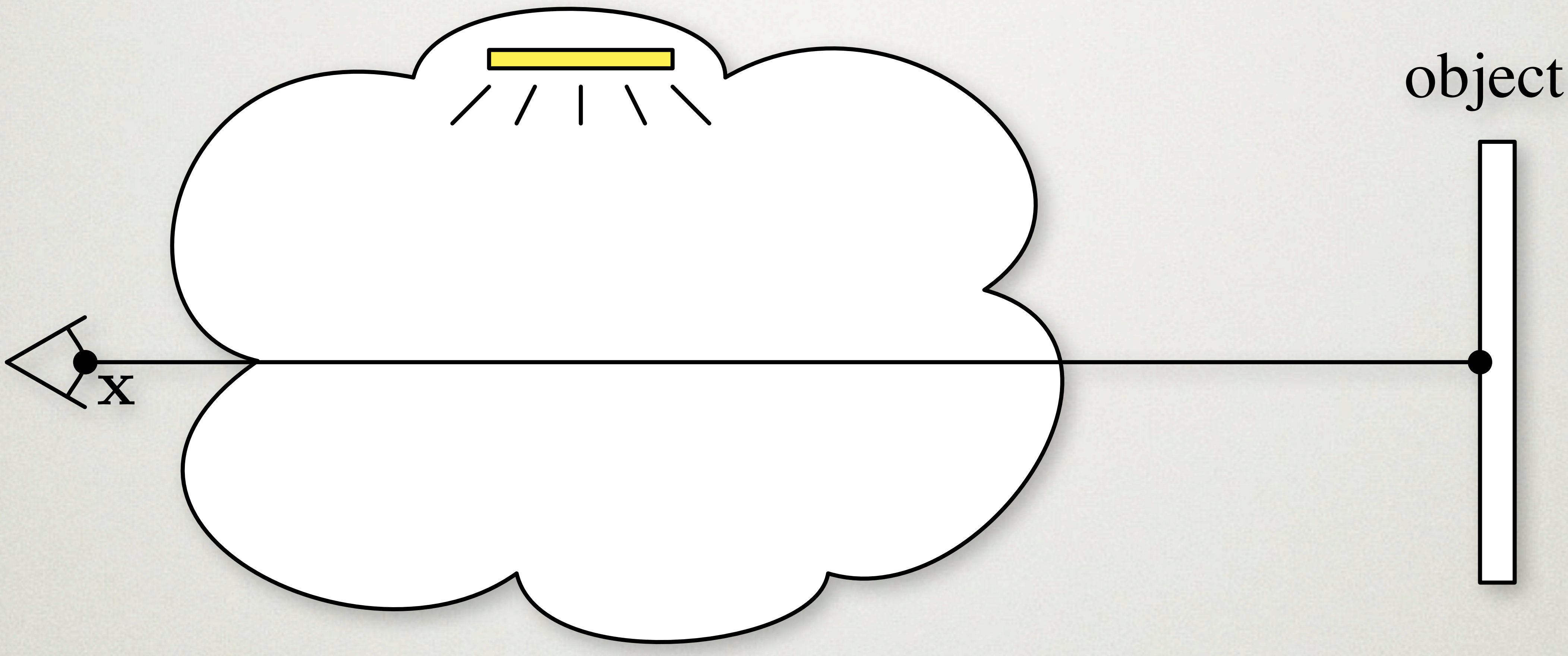
Arbitrary Measurements Using the Photon Map

- However, any **arbitrary** weighting function $W_e(\mathbf{x}_i, \vec{\omega}_i)$ can be used to compute a different **measurement**.
- **If we can express our problem as a measurement, we can estimate it using the photon map.**

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

* if we can represent the quantity we want to compute as a measurement, then we can compute estimates of that quantity using the measurement equation

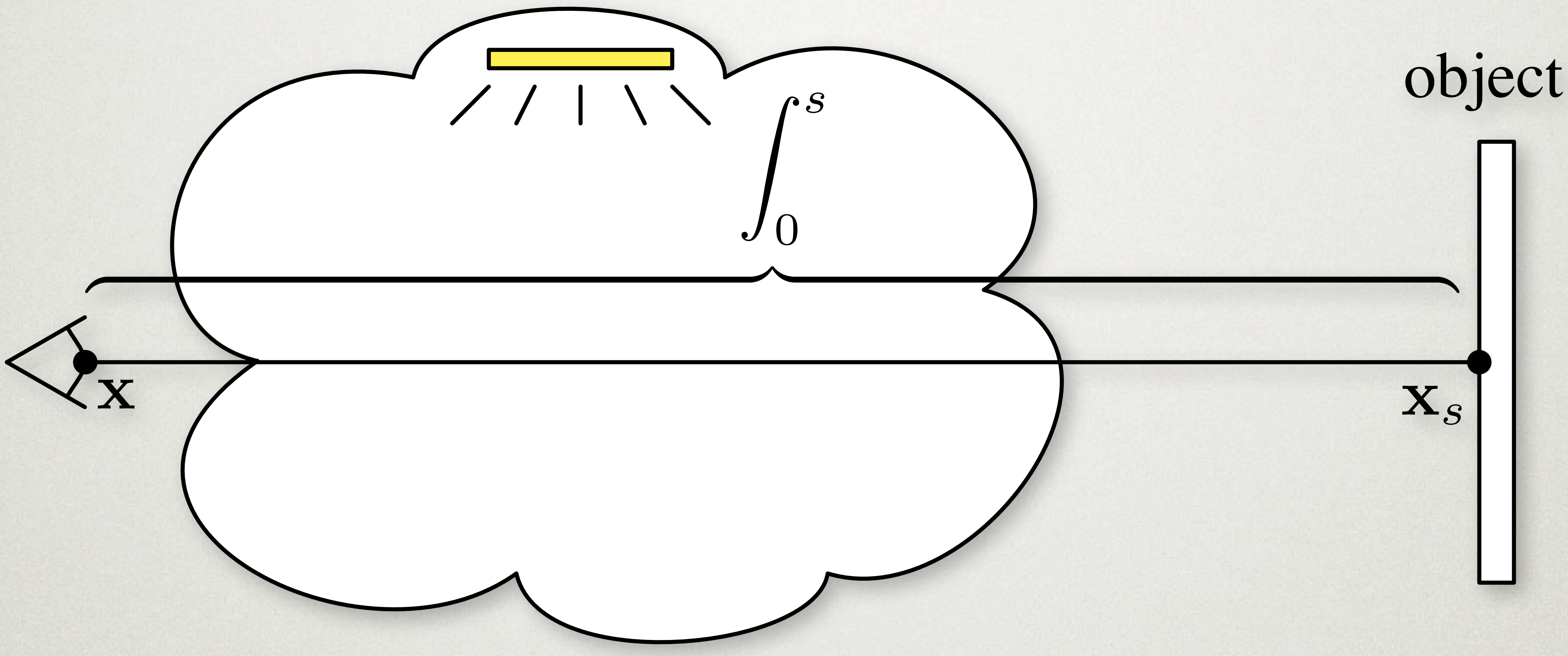
Volume Rendering Equation



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

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Volume Rendering Equation



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

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i,\vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Volume Rendering Equation

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

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Volume Rendering Equation

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

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Volume Rendering Equation

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

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

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i,\vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Volume Rendering Equation

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

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

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)p(\mathbf{x}_t,\vec{\omega},\vec{\omega}_t)L(\mathbf{x}_t,\vec{\omega}_t)\, d\omega_t\, dt$$

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i,\vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Beam Radiance

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t) \; d\omega_t \; dt$$

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

Beam Radiance

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t)L(\mathbf{x}_t, \vec{\omega}_t)\; d\omega_t\; dt$$

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L\rangle$$

Beam Radiance

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t) \; d\omega_t \; dt$$

$$\langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}_t, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t) \; d\vec{\omega}_t \; d\mathcal{V}(\mathbf{x}_t)$$

$$E\left[\frac{1}{N} \sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i) \alpha_i \right] = \langle W_e, L \rangle$$

Beam Radiance

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\; d\omega_t\; dt$$

$\updownarrow$

$$\langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}_t, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\; d\vec{\omega}_t\; d\mathcal{V}(\mathbf{x}_t)$$

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

Beam Radiance

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\, d\omega_t\, dt$$

$$\updownarrow$$

$$\langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}_t, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\, d\vec{\omega}_t\, d\mathcal{V}(\mathbf{x}_t)$$

Change integration over *t* into integration over *V*.

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

Beam Radiance is a Measurement

$$\int_{\mathcal{V}} \int_{\Omega_{4\pi}} \delta\, T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t) p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\; d\omega_t \; d\mathcal{V}(\mathbf{x}_t)$$

$$\updownarrow$$

$$\langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}_t, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\; d\vec{\omega}_t \; d\mathcal{V}(\mathbf{x}_t)$$

Change integration over *t* into integration over *V*.
Use delta function, δ, to limit integration to line.

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

* delta function means we only get a useable estimate if a photon falls directly on the line

Beam Radiance (Bias)

$$\int_{\mathcal{V}} \int_{\Omega_{4\pi}} K T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t) \, d\omega_t \, d\mathcal{V}(\mathbf{x}_t)$$

$$\updownarrow$$

$$\langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}_t, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t) \, d\vec{\omega}_t \, d\mathcal{V}(\mathbf{x}_t)$$

In practice, use cylindrical blurring kernel, K.

$$E\left[\frac{1}{N} \sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i) \alpha_i \right] = \langle W_e, L \rangle$$

* so in practice we replace the delta function with a blurring kernel which blurs radiance from the line into a cylinder.
* the kernel allows photons that are not directly on the line to be used in the estimate
* we have the freedom to choose the exact form of this blurring kernel

Beam Radiance (Bias)

$$\int_{\mathcal{V}} \int_{\Omega_{4\pi}} K T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) p(\mathbf{x}_t, \vec{\omega}, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\, d\omega_t\, d\mathcal{V}(\mathbf{x}_t)$$

$$\langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x}_t, \vec{\omega}_t) L(\mathbf{x}_t, \vec{\omega}_t)\, d\vec{\omega}_t\, d\mathcal{V}(\mathbf{x}_t)$$

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i, \vec{\omega}_i)\alpha_i\right] = \langle W_e, L \rangle$$

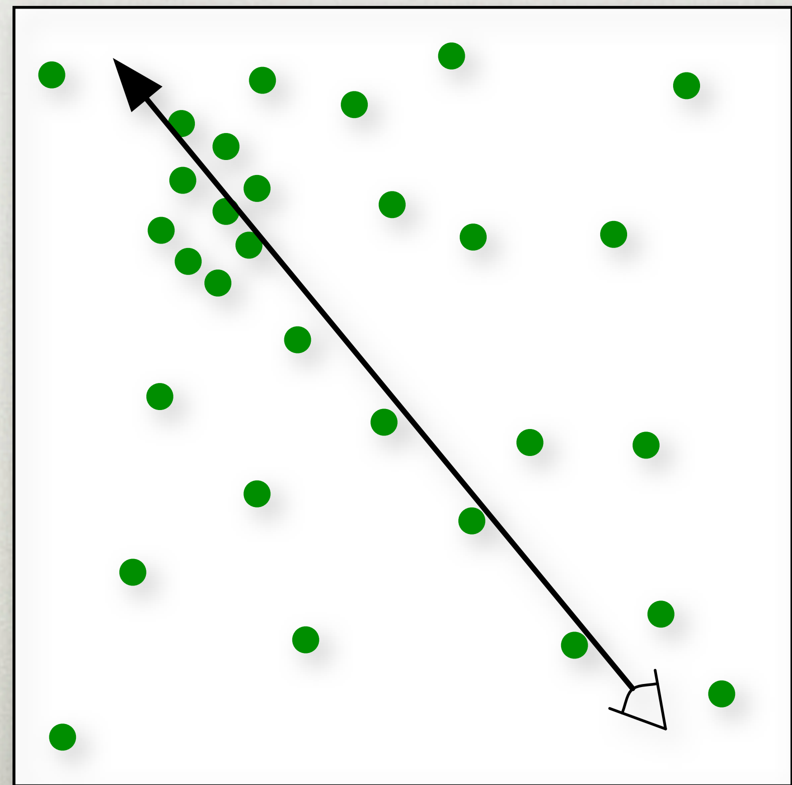
Thursday, 6 September 12

* so in practice we replace the delta function with a blurring kernel which blurs radiance from the line into a cylinder.
* the kernel allows photons that are not directly on the line to be used in the estimate
* we have the freedom to choose the exact form of this blurring kernel

Volumetric Photon Mapping

Conventional Radiance Estimate

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

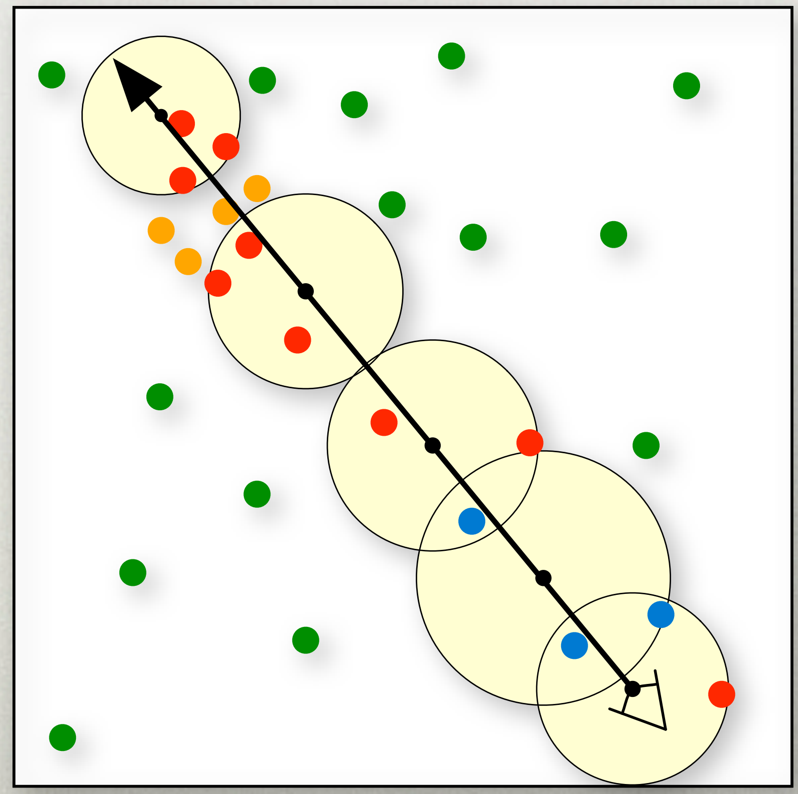


Thursday, 6 September 12

Volumetric Photon Mapping

Conventional Radiance Estimate

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

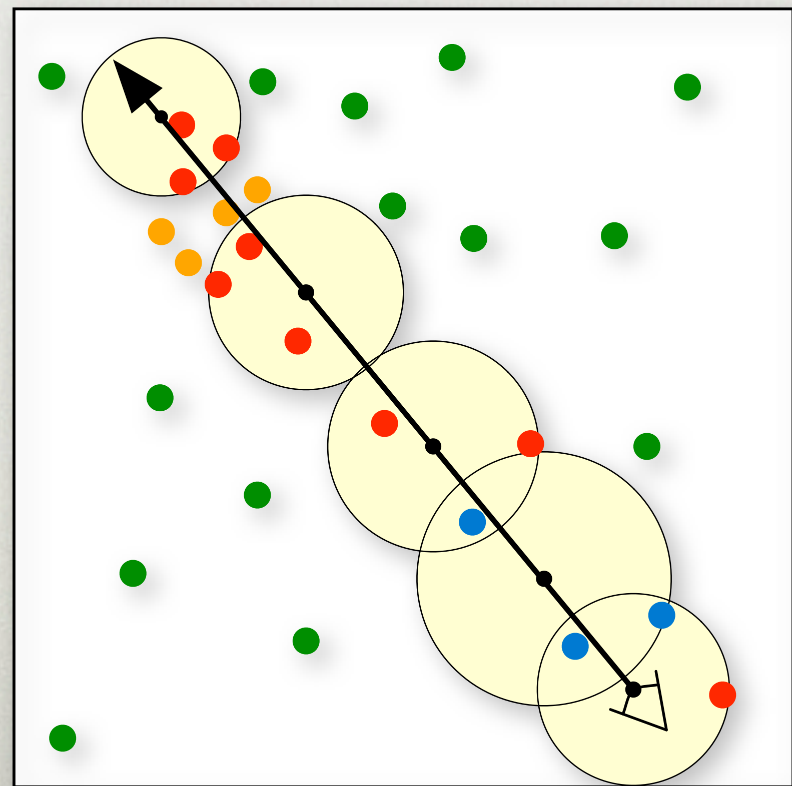


Volumetric Photon Mapping

Conventional Radiance Estimate

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

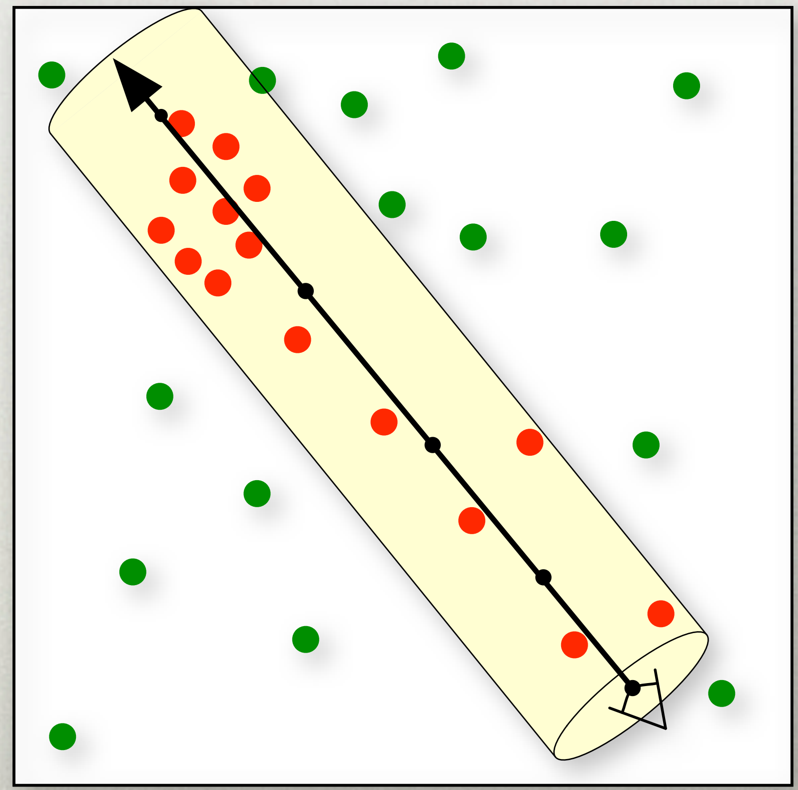
$$\left(\sum_{t=0}^{S-1} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\Delta_t\right)$$



Volumetric Photon Mapping

Beam Radiance Estimate

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

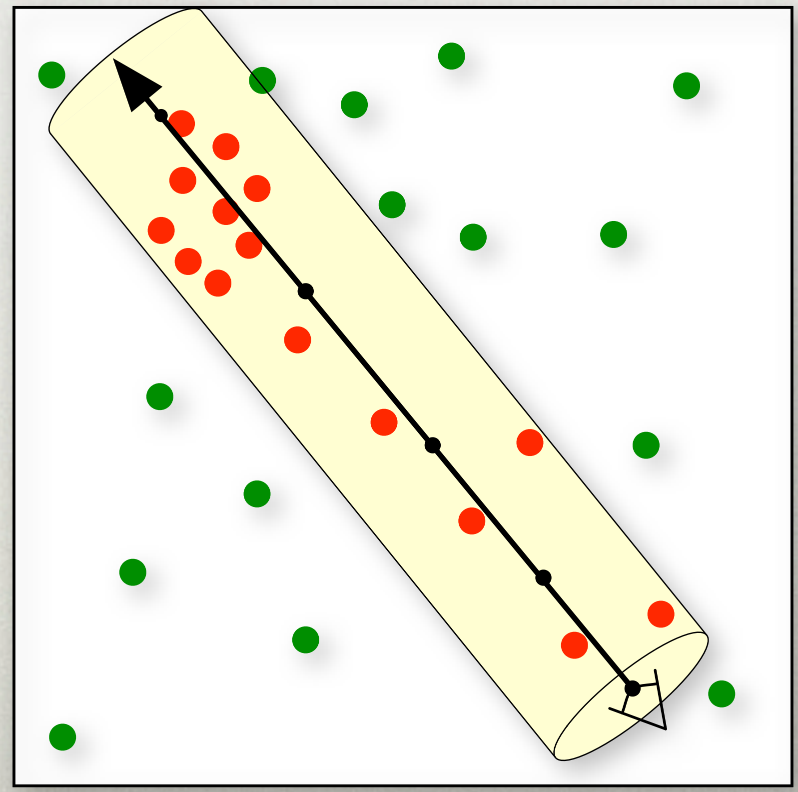


Volumetric Photon Mapping

Beam Radiance Estimate

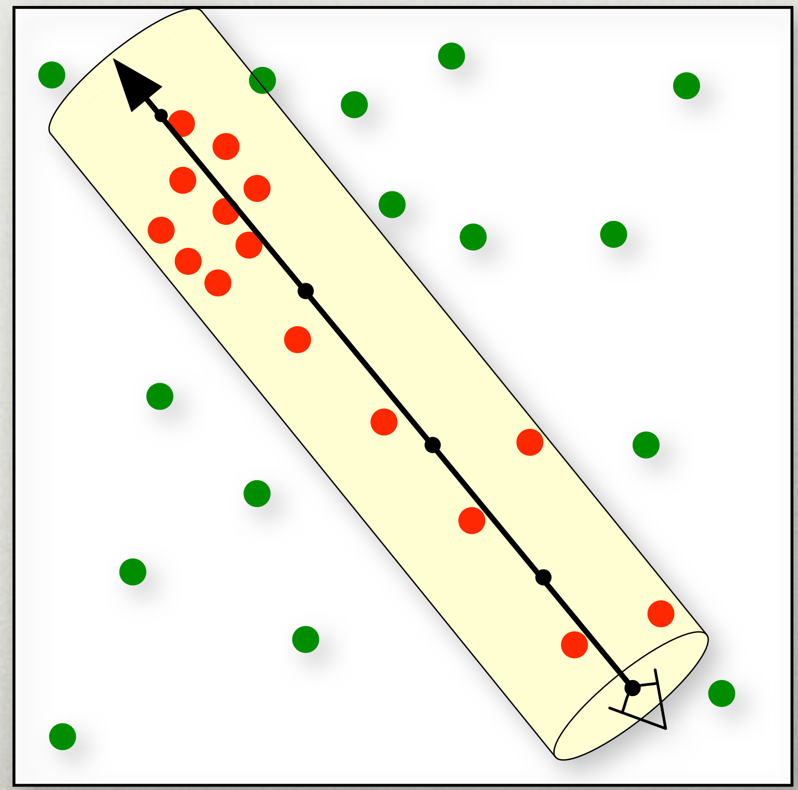
$$L(\mathbf{x}, \vec{\omega}) \approx T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, \vec{\omega}) + L_b(\mathbf{x}, \vec{\omega})$$

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



WHAT IS K_i?

- A fixed-size kernel results in a uniform blur of the photon map.
- In this case, we need to find photons in fixed-radius cylinder about ray.



Fixed Radius Comparison

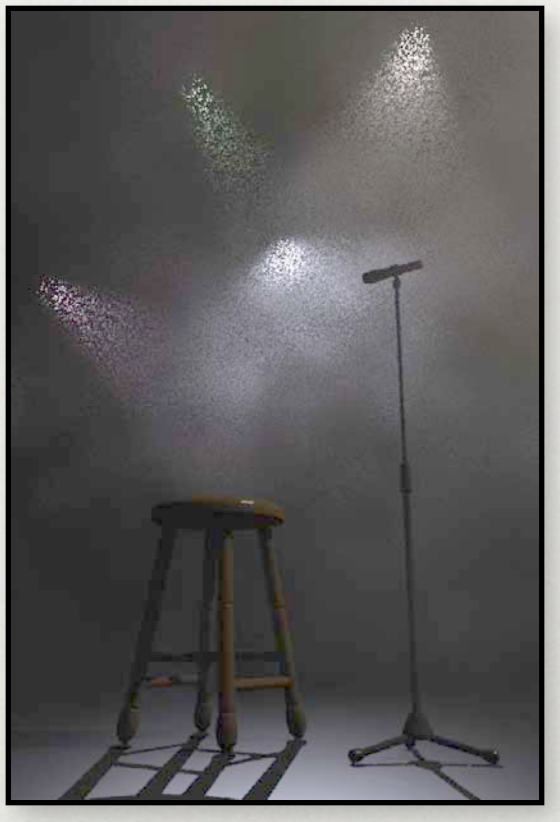
Beam Estimate

Thursday, 6 September 12

* When using a constant blurring radius, in the limit the conventional and beam radiance estimates are equivalent.
* uses exactly the same photon map

Fixed Radius Comparison

Conv. Estimate Beam Estimate

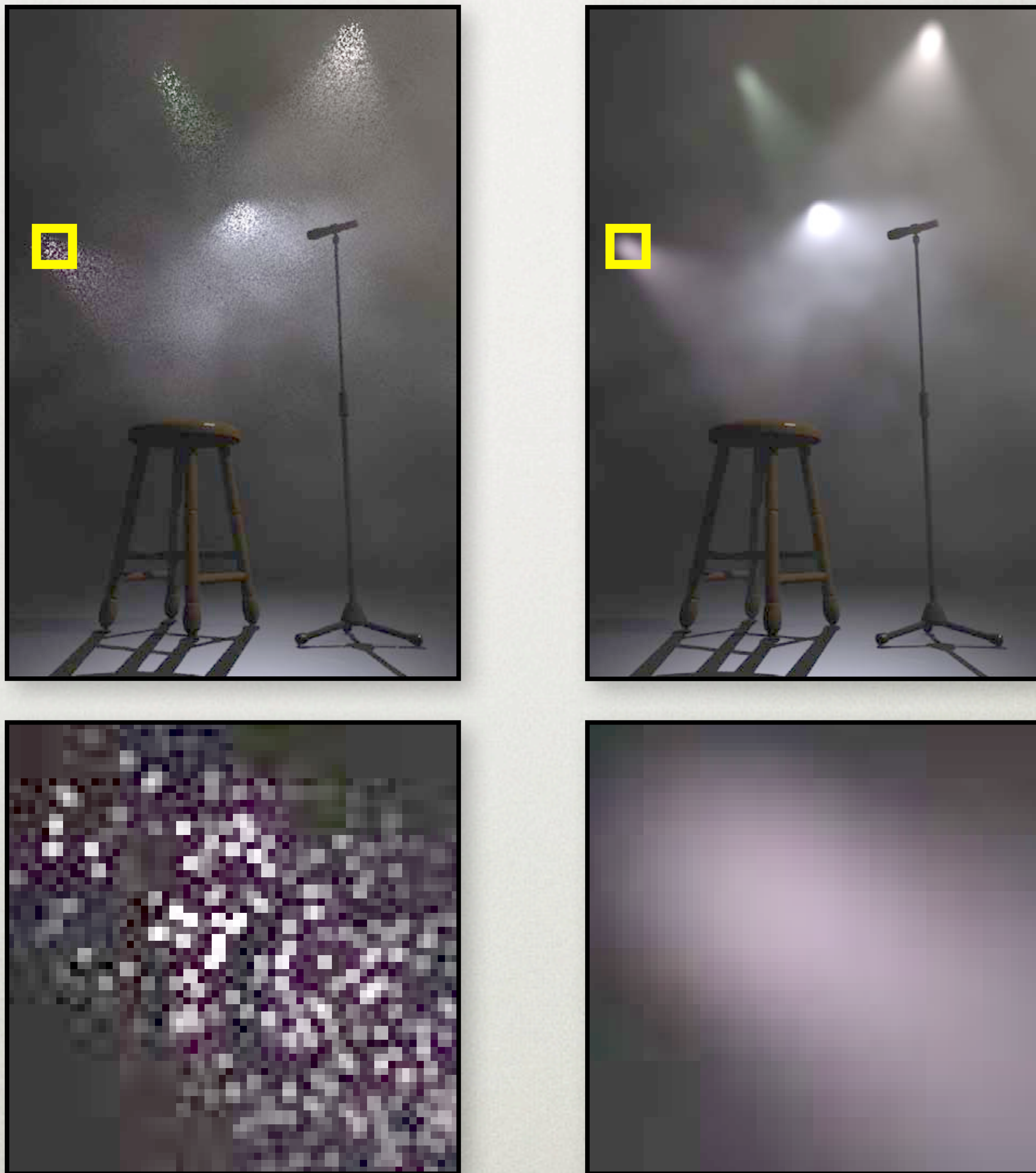


Thursday, 6 September 12

* When using a constant blurring radius, in the limit the conventional and beam radiance estimates are equivalent.
* uses exactly the same photon map

Fixed Radius Comparison

Conv. Estimate Beam Estimate

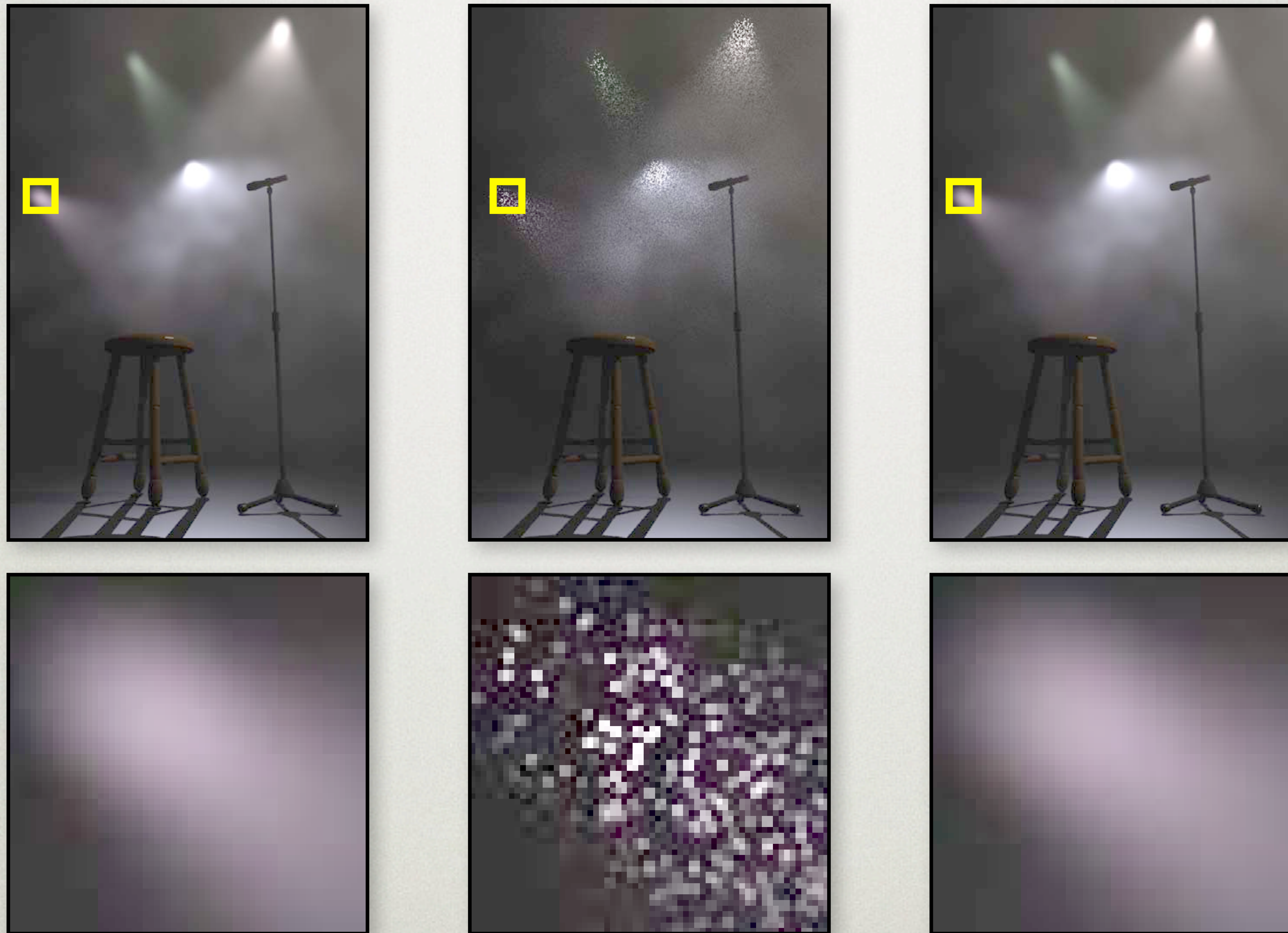


(4:21) (4:15)

* When using a constant blurring radius, in the limit the conventional and beam radiance estimates are equivalent.
* uses exactly the same photon map

Fixed Radius Comparison

Conv. Estimate | Conv. Estimate | Beam Estimate



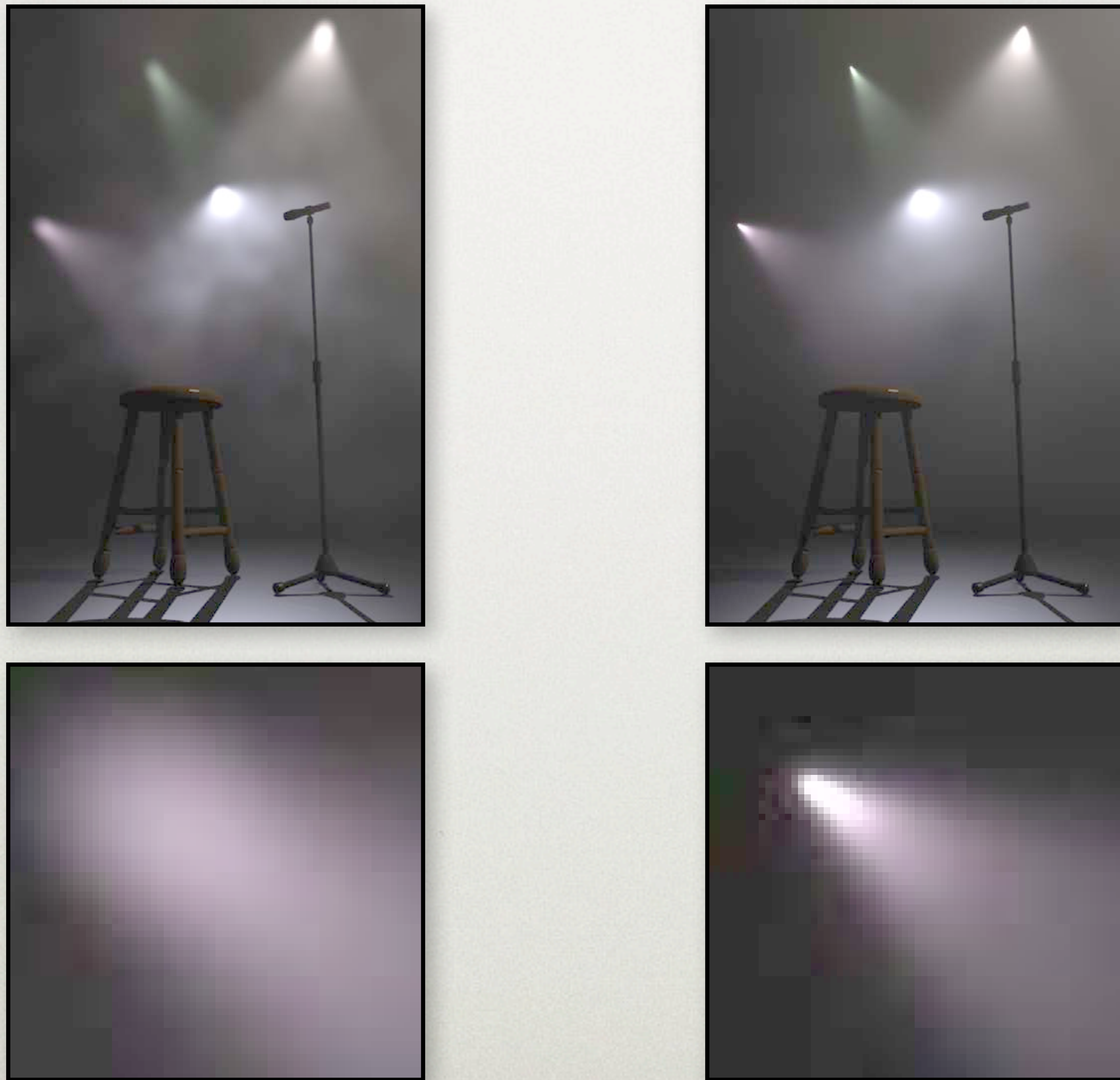
(∞) | (4:21) | (4:15)

Thursday, 6 September 12

* When using a constant blurring radius, in the limit the conventional and beam radiance estimates are equivalent.
* uses exactly the same photon map

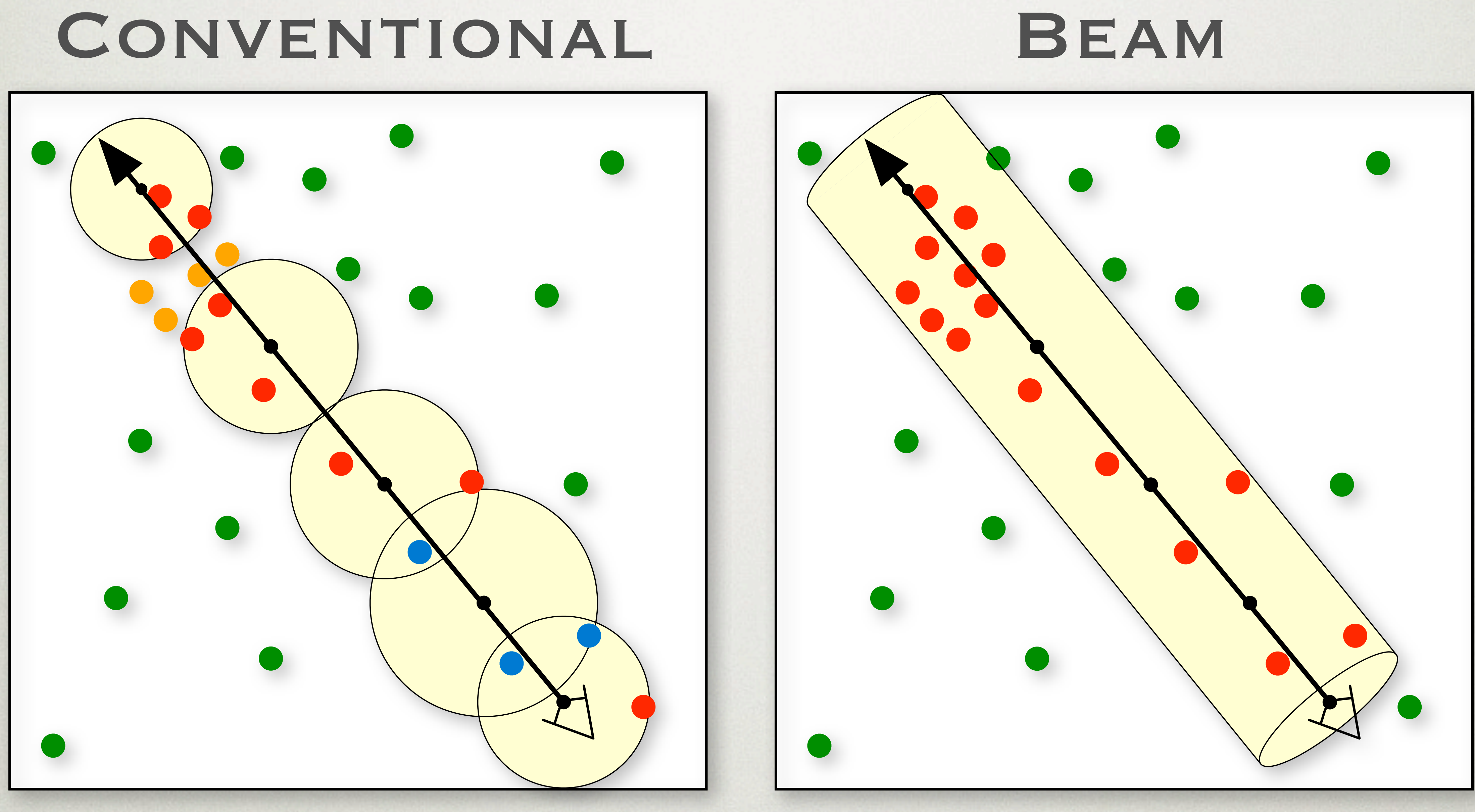
Conventional Radiance Estimate

Fixed Radius **Nearest Neighbor**



* however, in practice a fixed radius is rarely used, and the nearest neighbors method is used to adapt the radius to the local density of photons

Adaptive K_i?

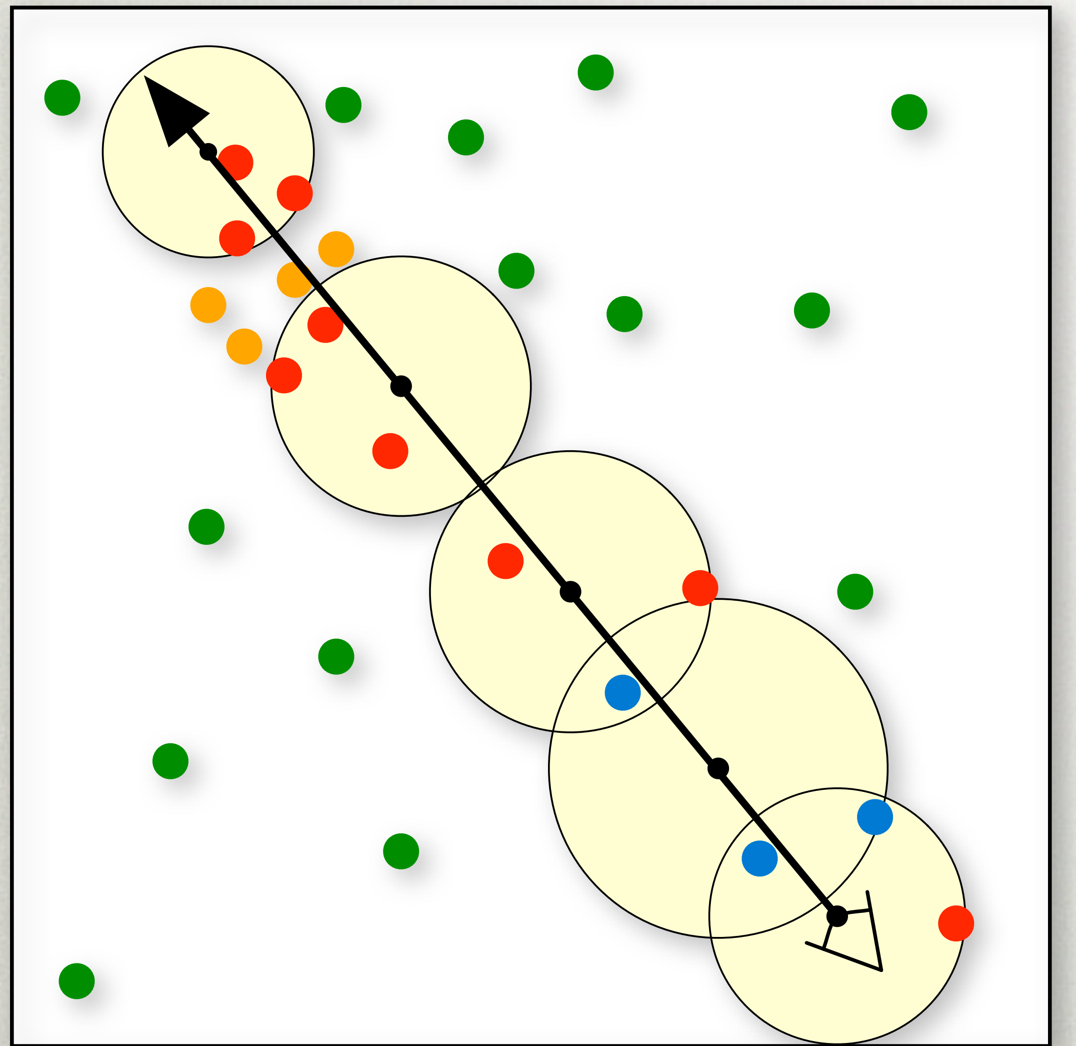


Thursday, 6 September 12

* The conventional radiance estimate uses the k-nearest neighbor method at a point.
* How can we generalize this along a line?

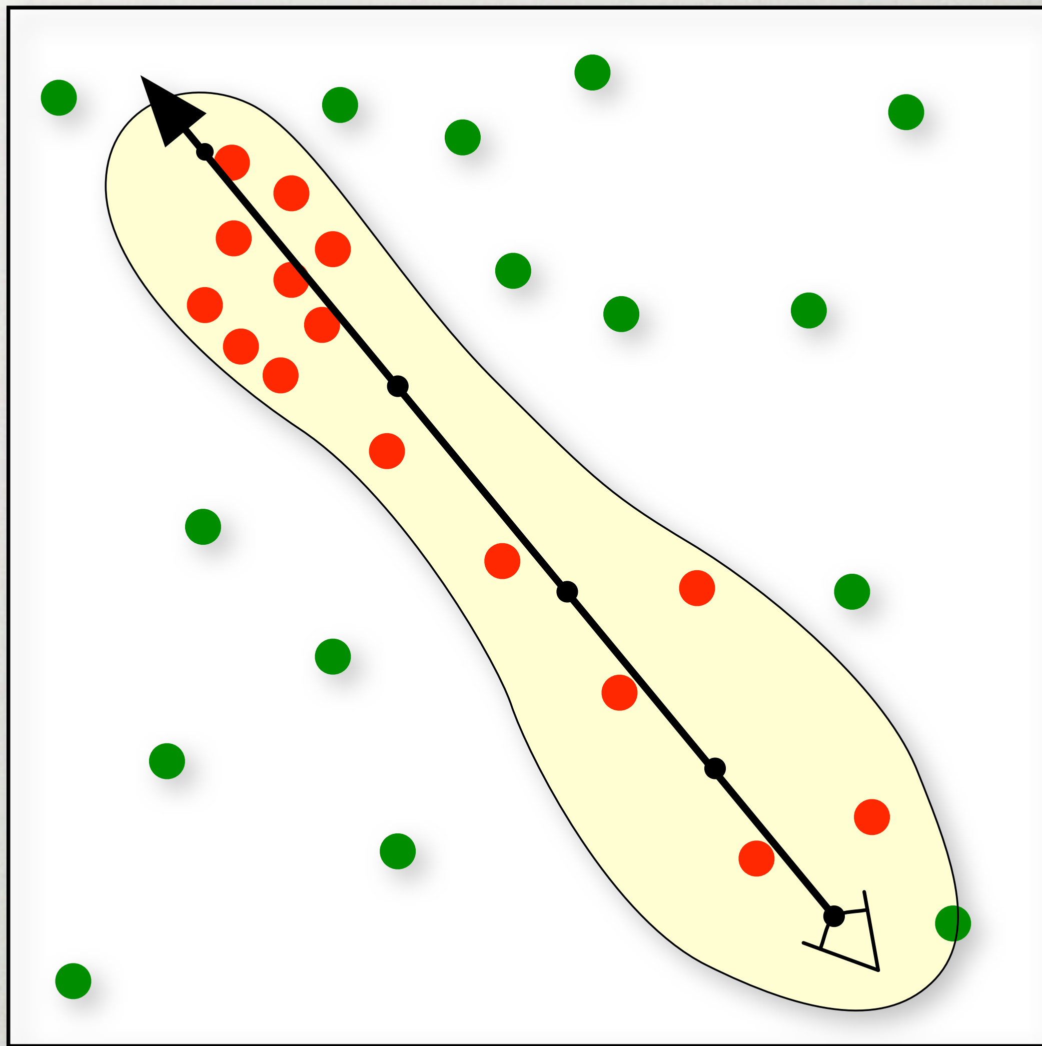
ADAPTIVE K_i?

CONVENTIONAL



Adaptive radius

BEAM

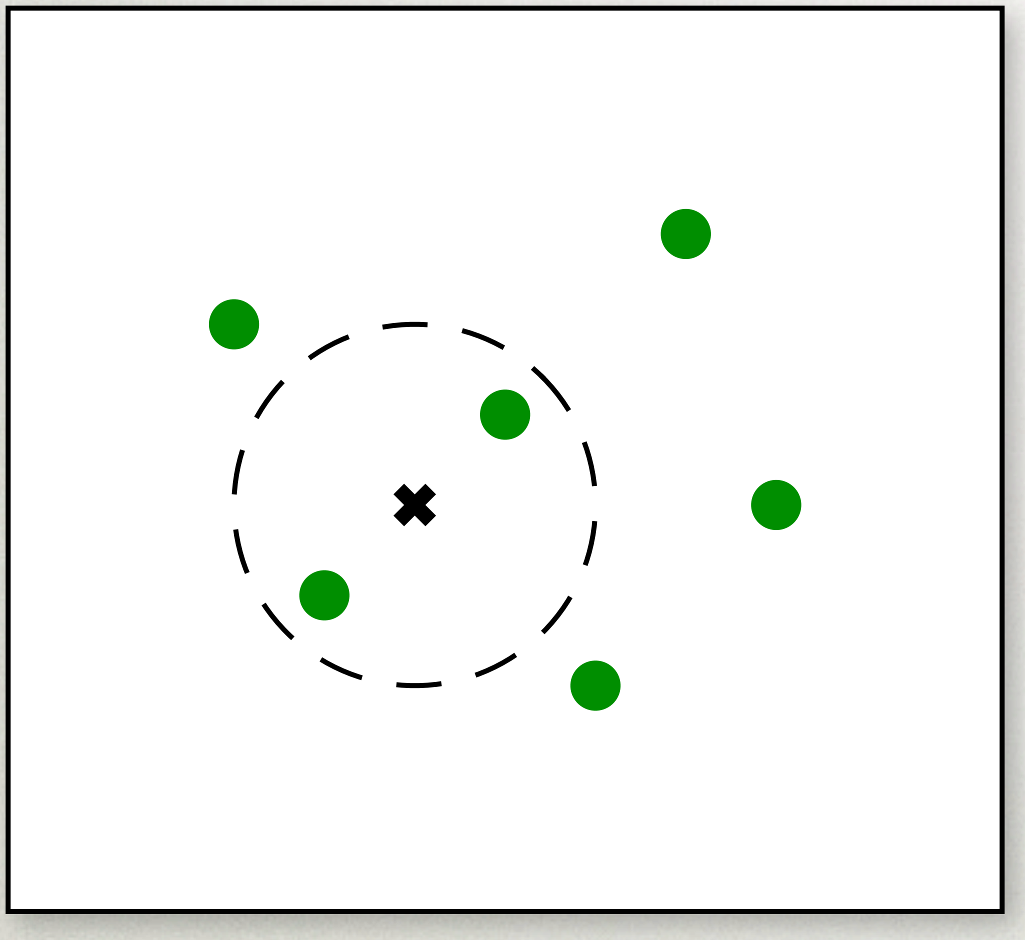


?

* The conventional radiance estimate uses the k-nearest neighbor method at a point.
* How can we generalize this along a line?

Primal vs. Dual

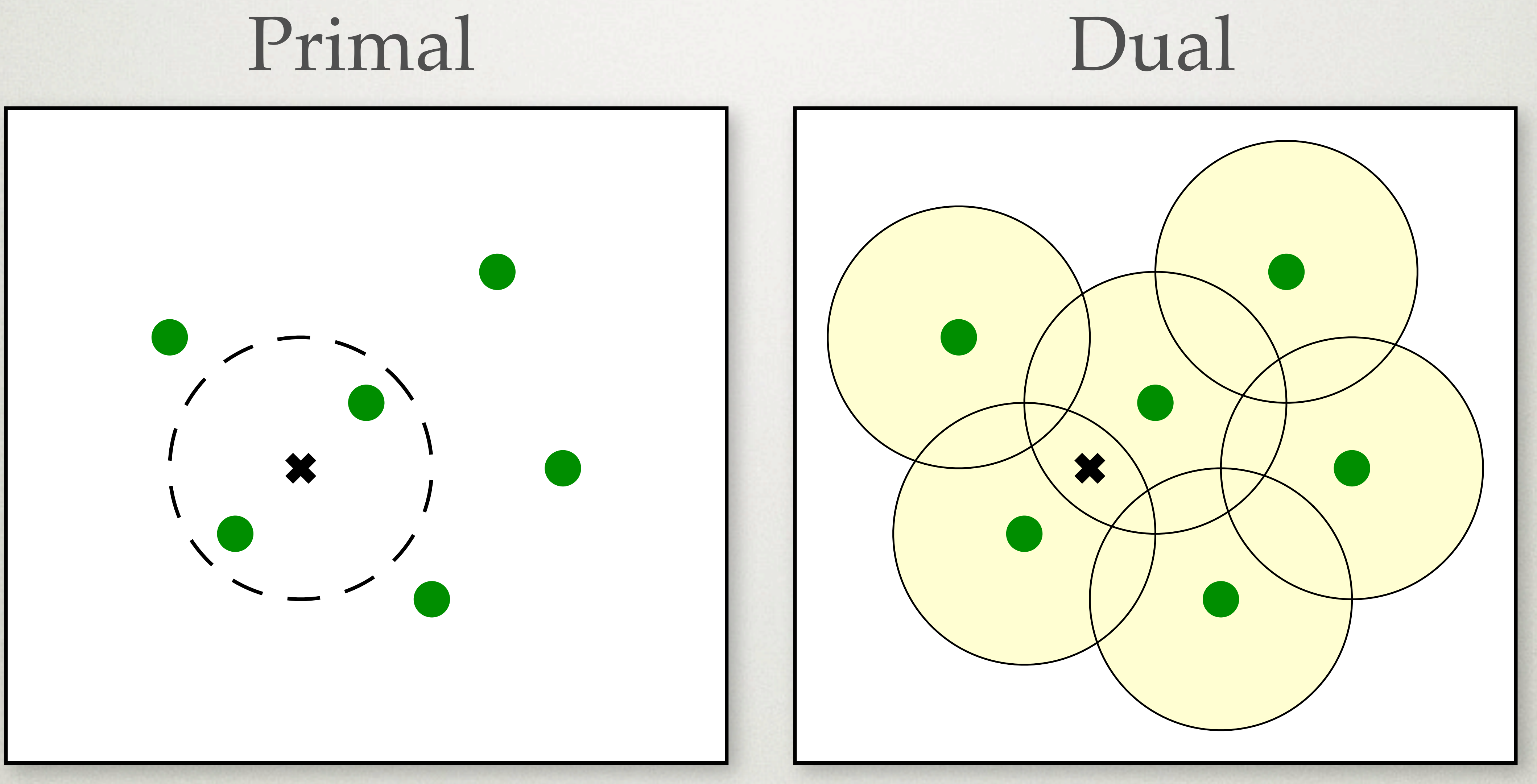
Primal



Thursday, 6 September 12

* in order to address this we turn to the primal vs. dual interpretation of density estimation
* two different interpretations of density estimation
* exactly equivalent for fixed-radius searches

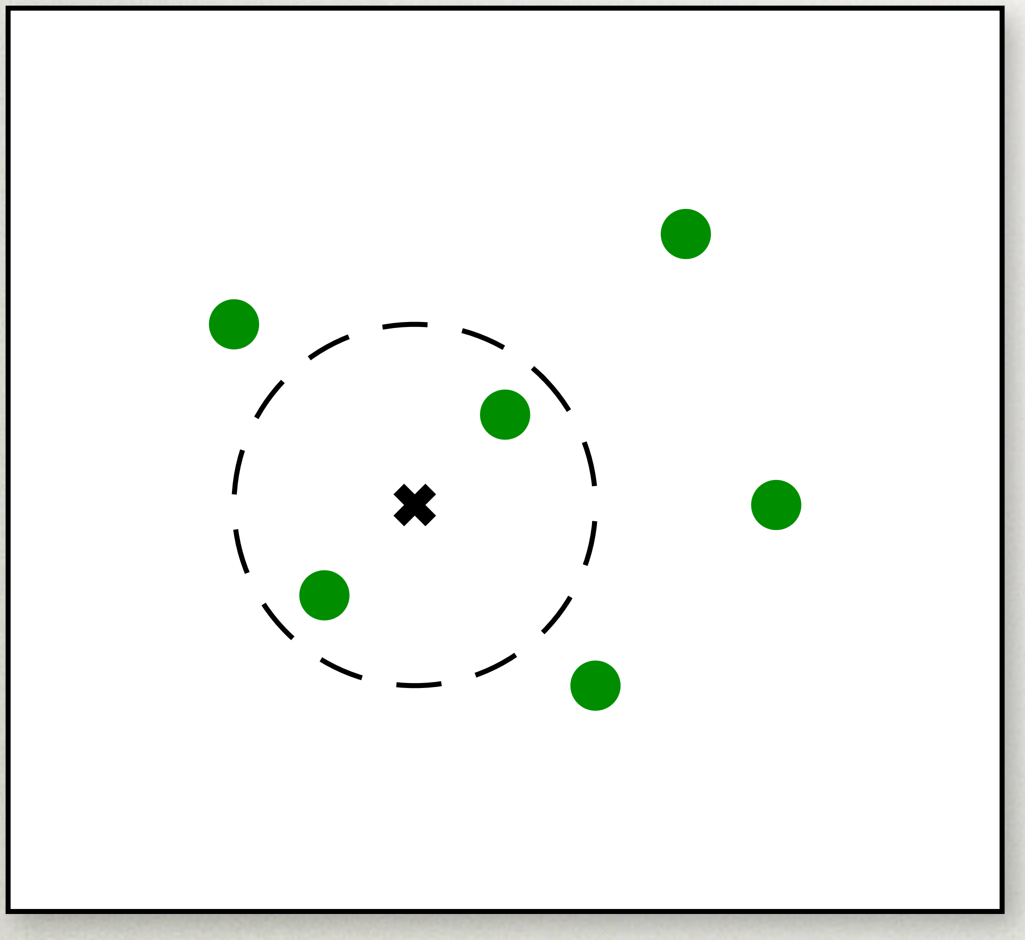
Primal vs. Dual



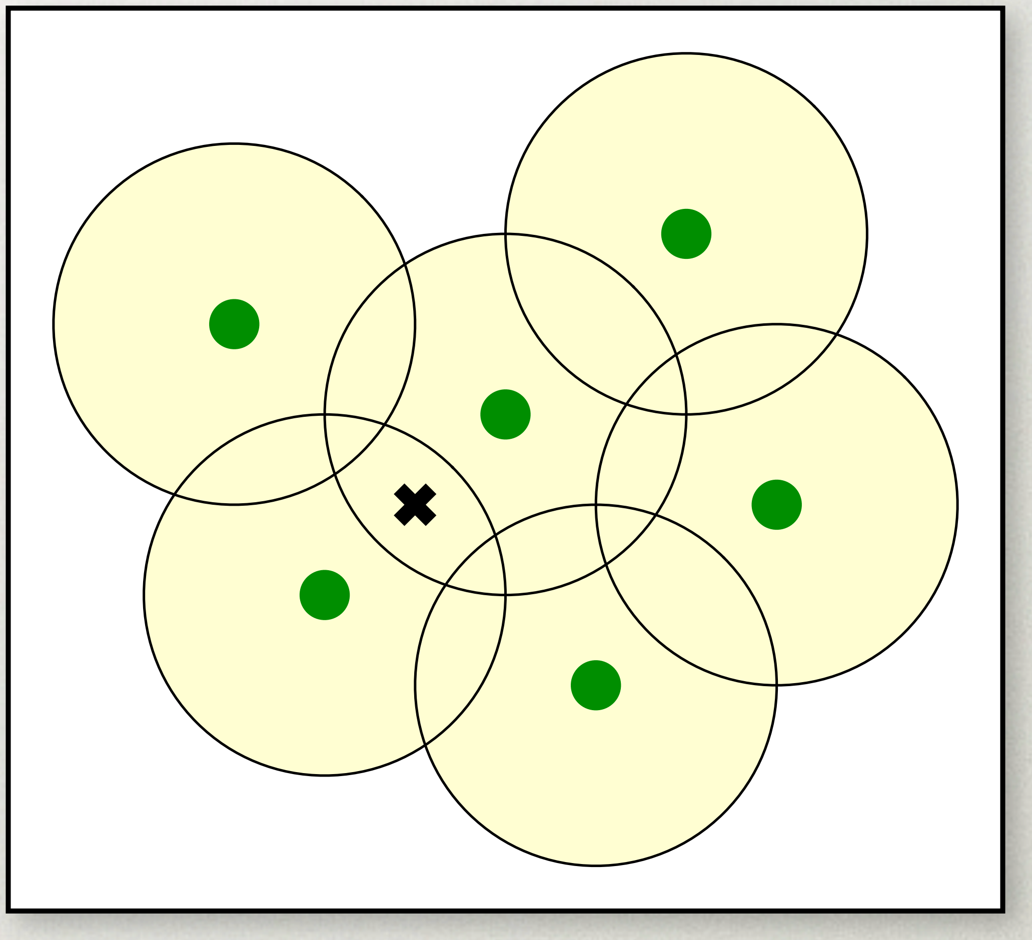
* in order to address this we turn to the primal vs. dual interpretation of density estimation
* two different interpretations of density estimation
* exactly equivalent for fixed-radius searches

Primal vs. Dual

Primal



Dual



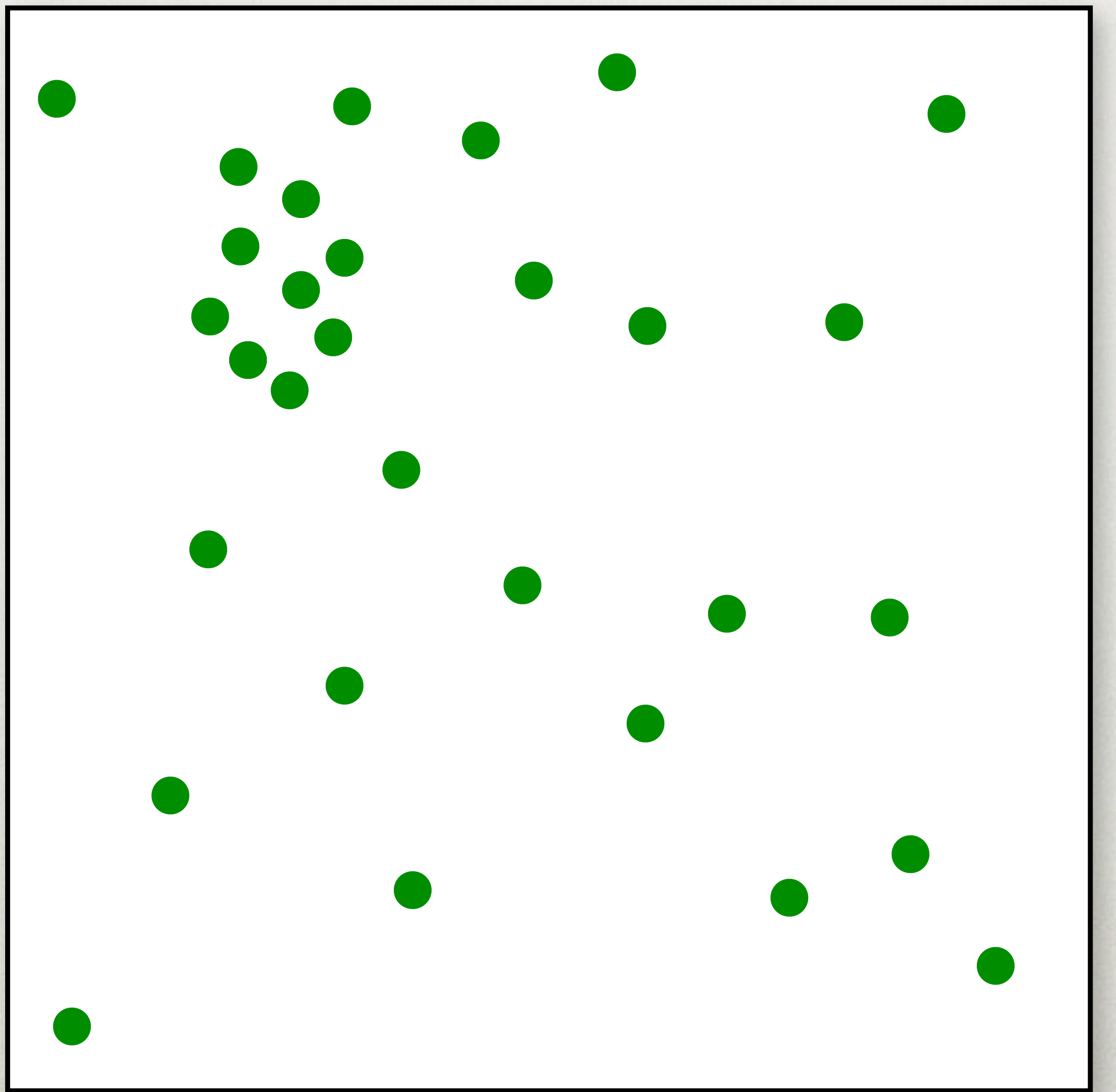
allow radius to vary:
adaptive kernel method

Thursday, 6 September 12

* in order to address this we turn to the primal vs. dual interpretation of density estimation
* two different interpretations of density estimation
* exactly equivalent for fixed-radius searches

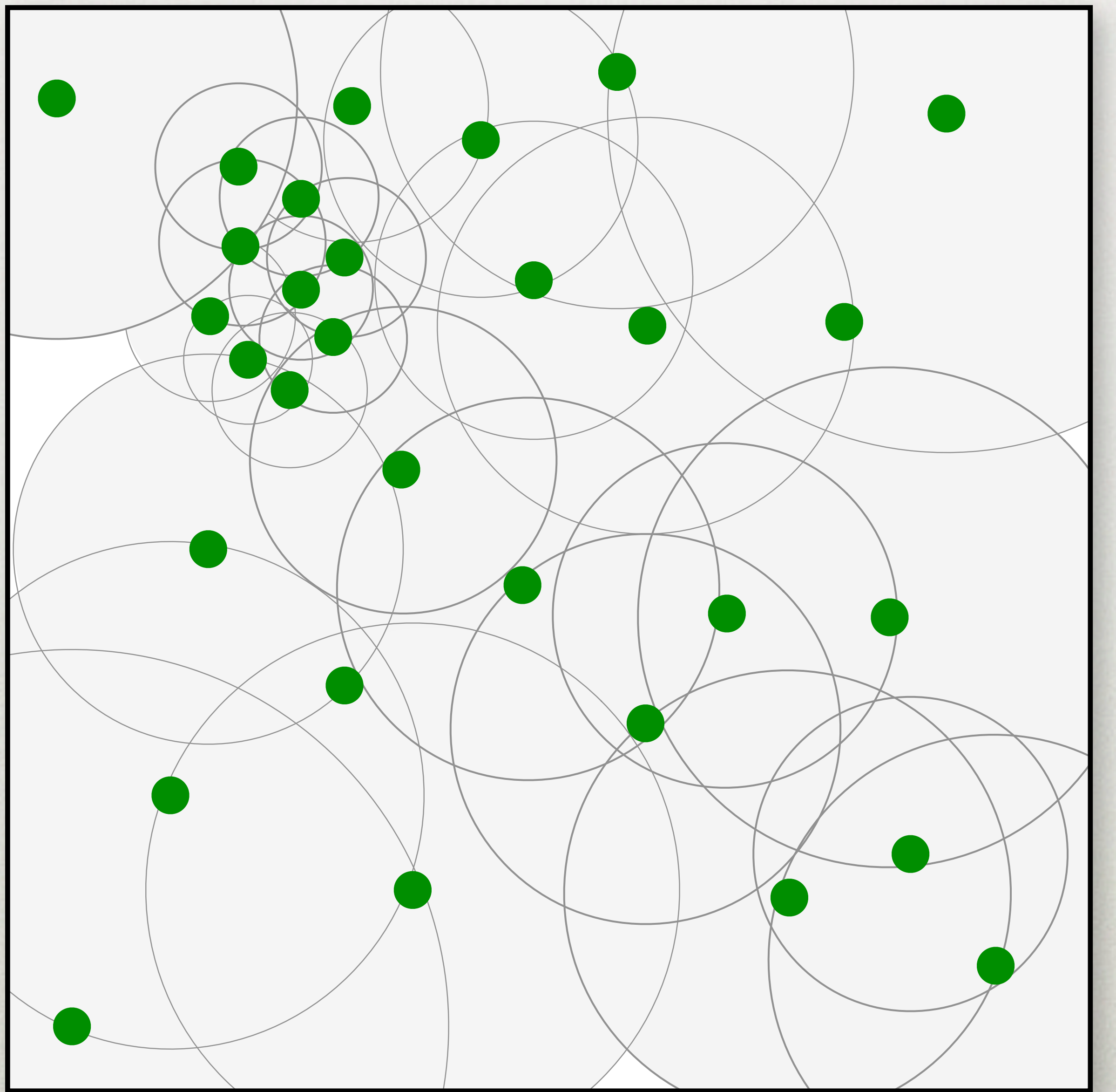
Volumetric Photon Mapping

Beam Radiance Estimate



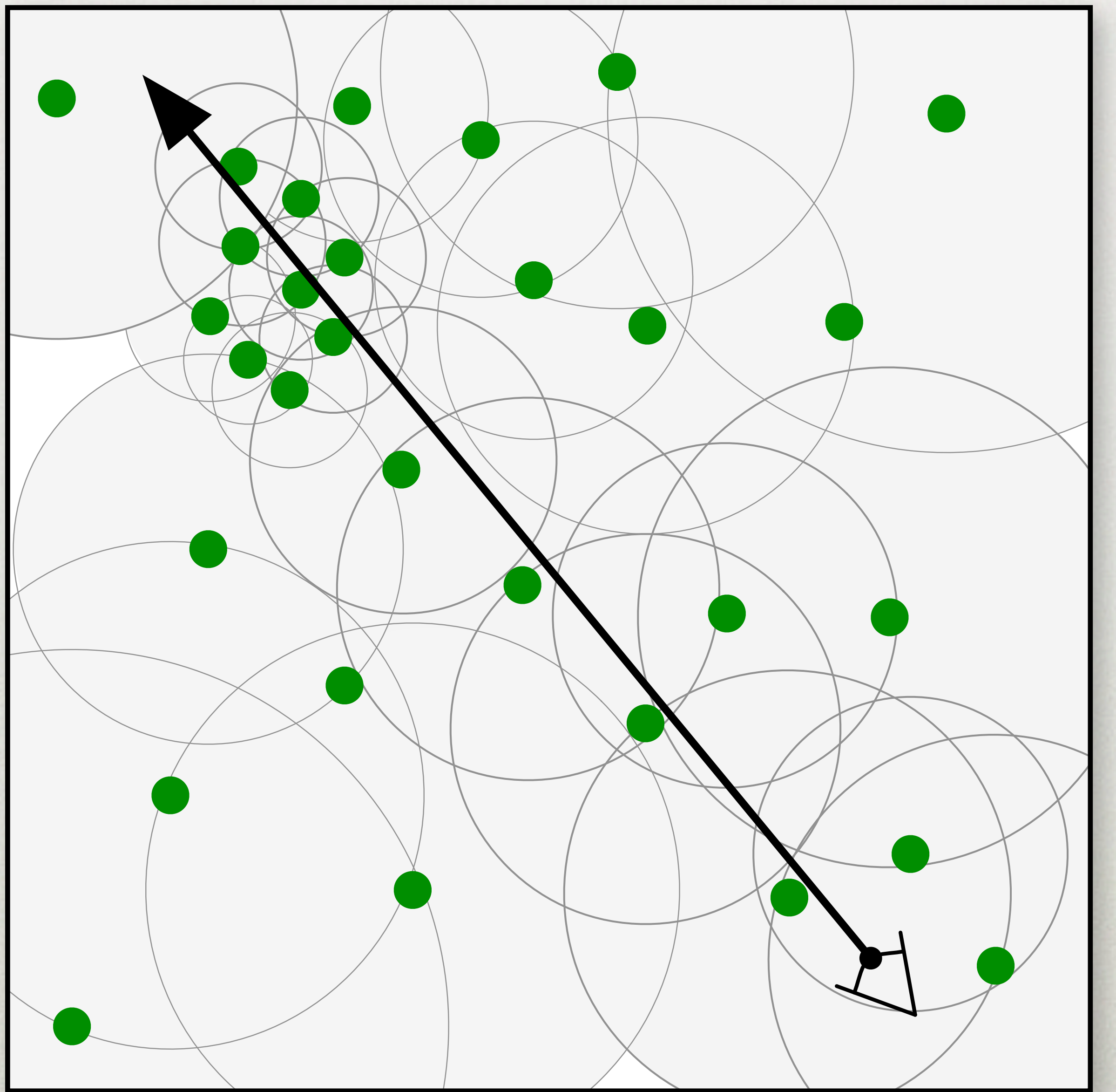
Volumetric Photon Mapping

Beam Radiance Estimate



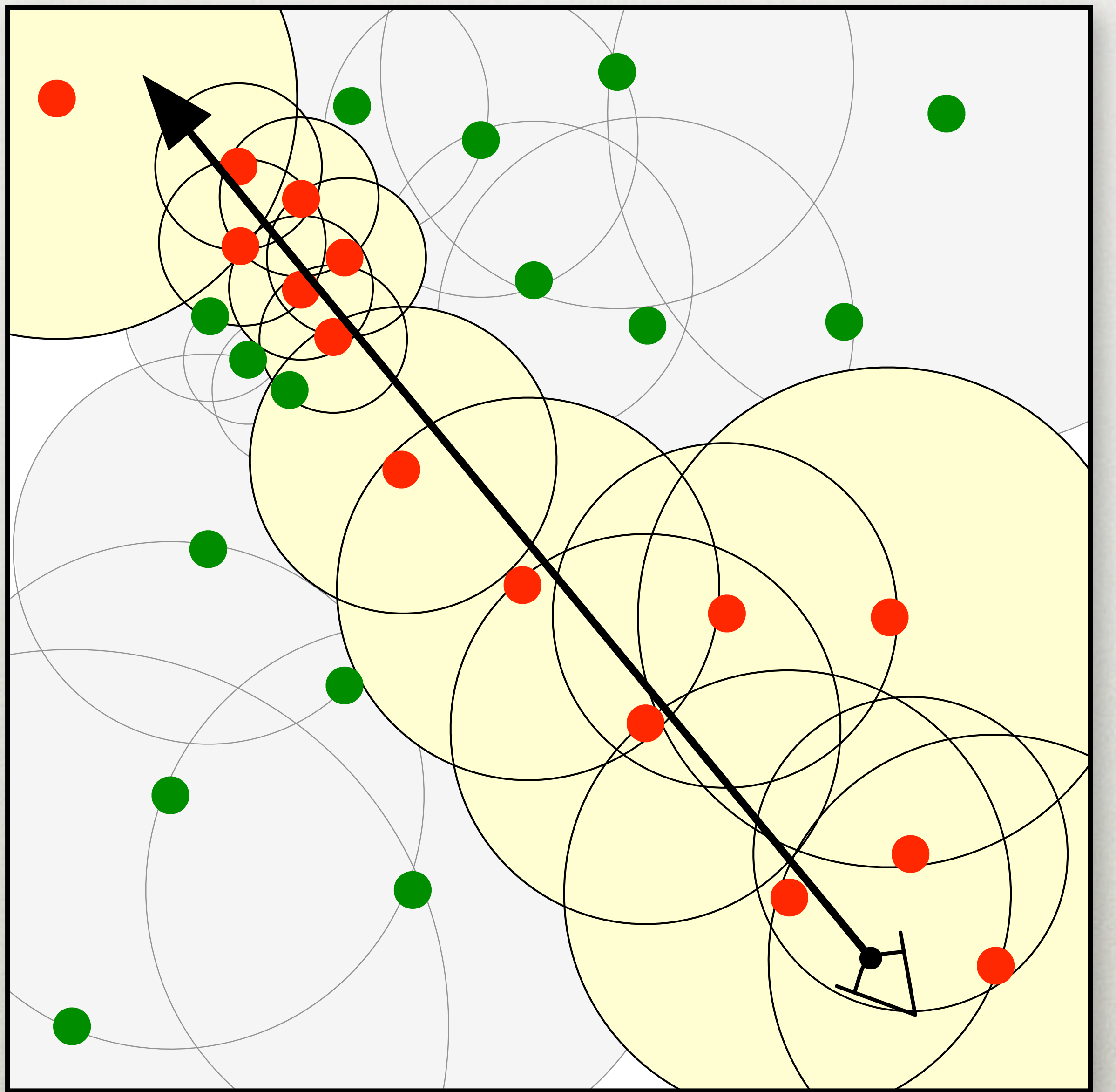
Volumetric Photon Mapping

Beam Radiance Estimate



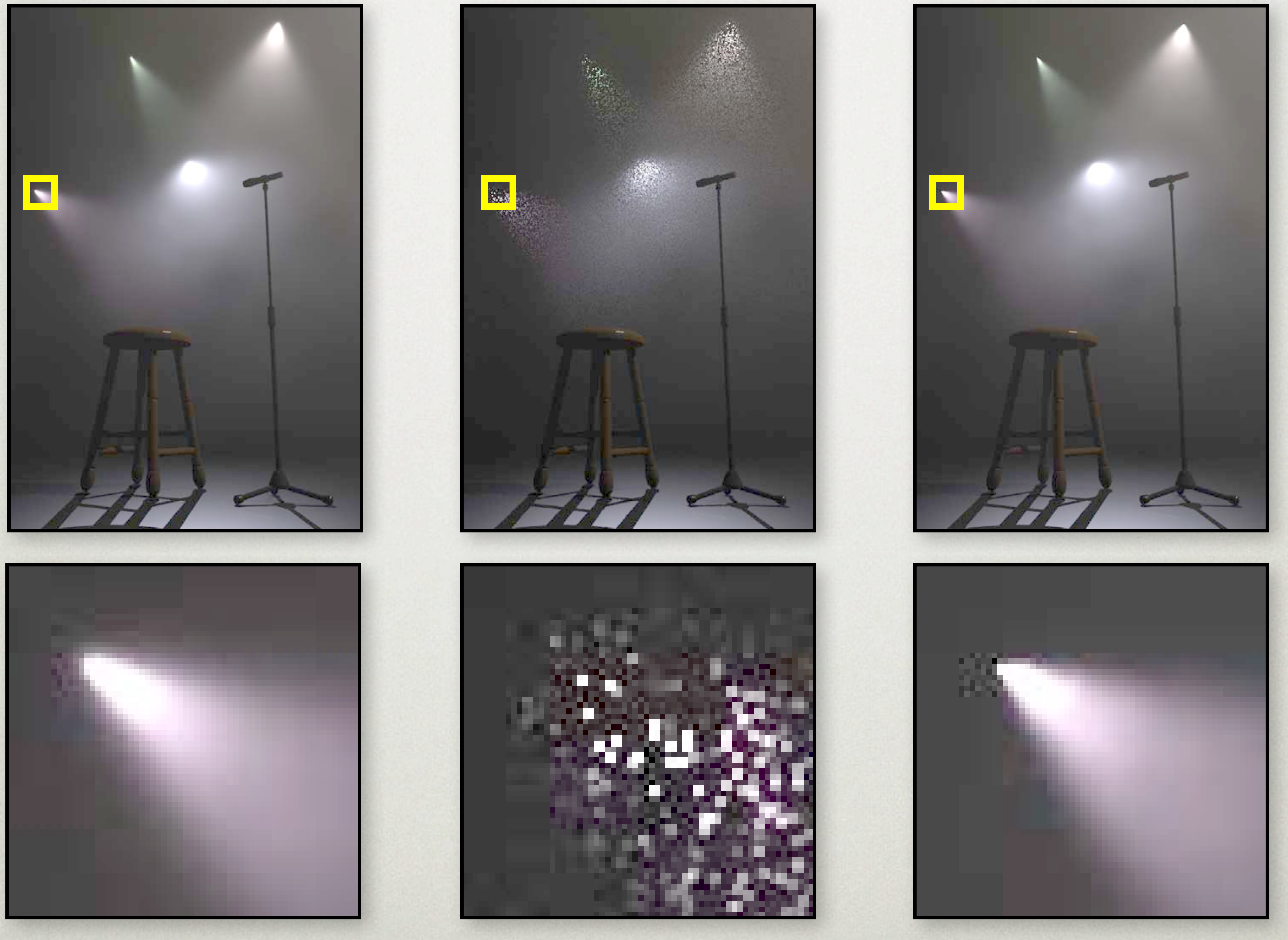
Volumetric Photon Mapping

Beam Radiance Estimate



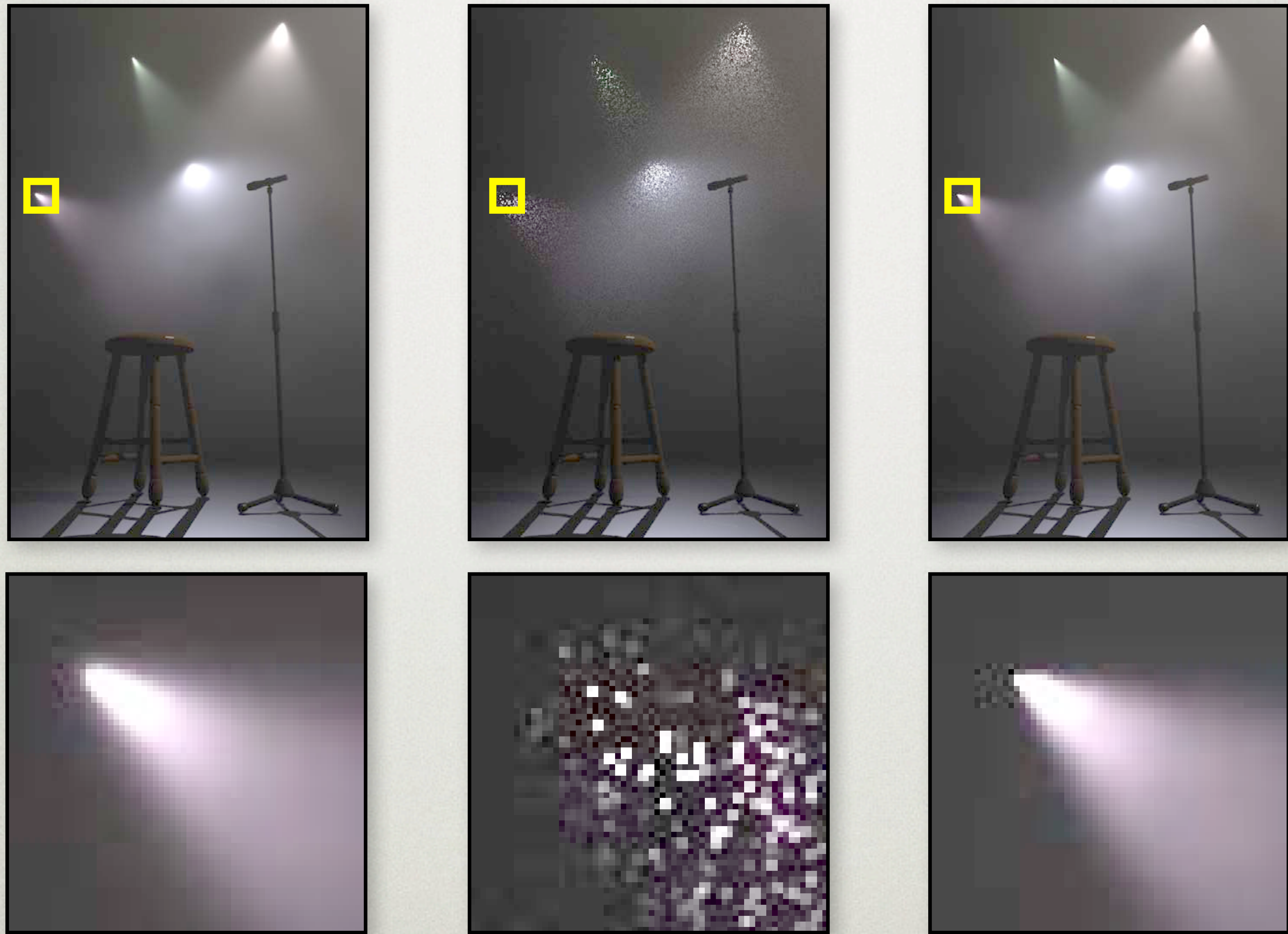
Adaptive Radius Comparison

Conv. Estimate | **Conv. Estimate** | **Beam Estimate**



Adaptive Radius Comparison

Conv. Estimate | **Conv. Estimate** | **Beam Estimate**



(∞) (6:38) (6:22)

Algorithm

1) Shoot photons from light sources.
2) Construct a balanced kD-tree for the photons.
3) Assign a radius for each photon (*photon-discs*).
4) Create acceleration structure over photon-discs.
5) Render:
    - For each ray through the medium, accumulate all *photon-discs* that intersect ray.

* first two steps identical to regular photon mapping

Algorithm

Same as Regular Photon Mapping

1) Shoot photons from light sources.
2) Construct a balanced kD-tree for the photons.
3) Assign a radius for each photon (*photon-discs*).
4) Create acceleration structure over photon-discs.
5) Render:
    - For each ray through the medium, accumulate all *photon-discs* that intersect ray.

* first two steps identical to regular photon mapping

Algorithm

Same as Regular Photon Mapping

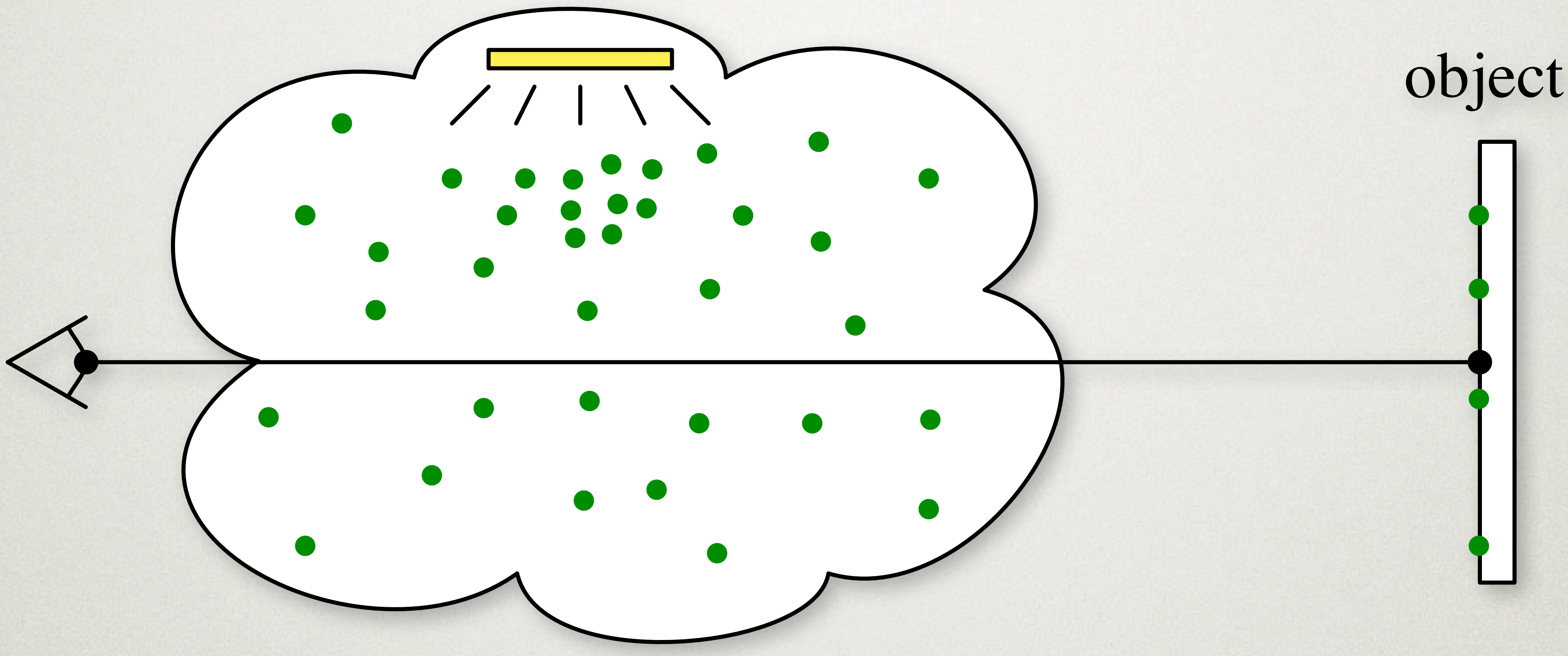
1) Shoot photons from light sources.
2) Construct a balanced kD-tree for the photons.
3) Assign a radius for each photon (*photon-discs*).
4) Create acceleration structure over photon-discs.
5) Render:
   - For each ray through the medium, accumulate all *photon-discs* that intersect ray.

Our Method

Thursday, 6 September 12

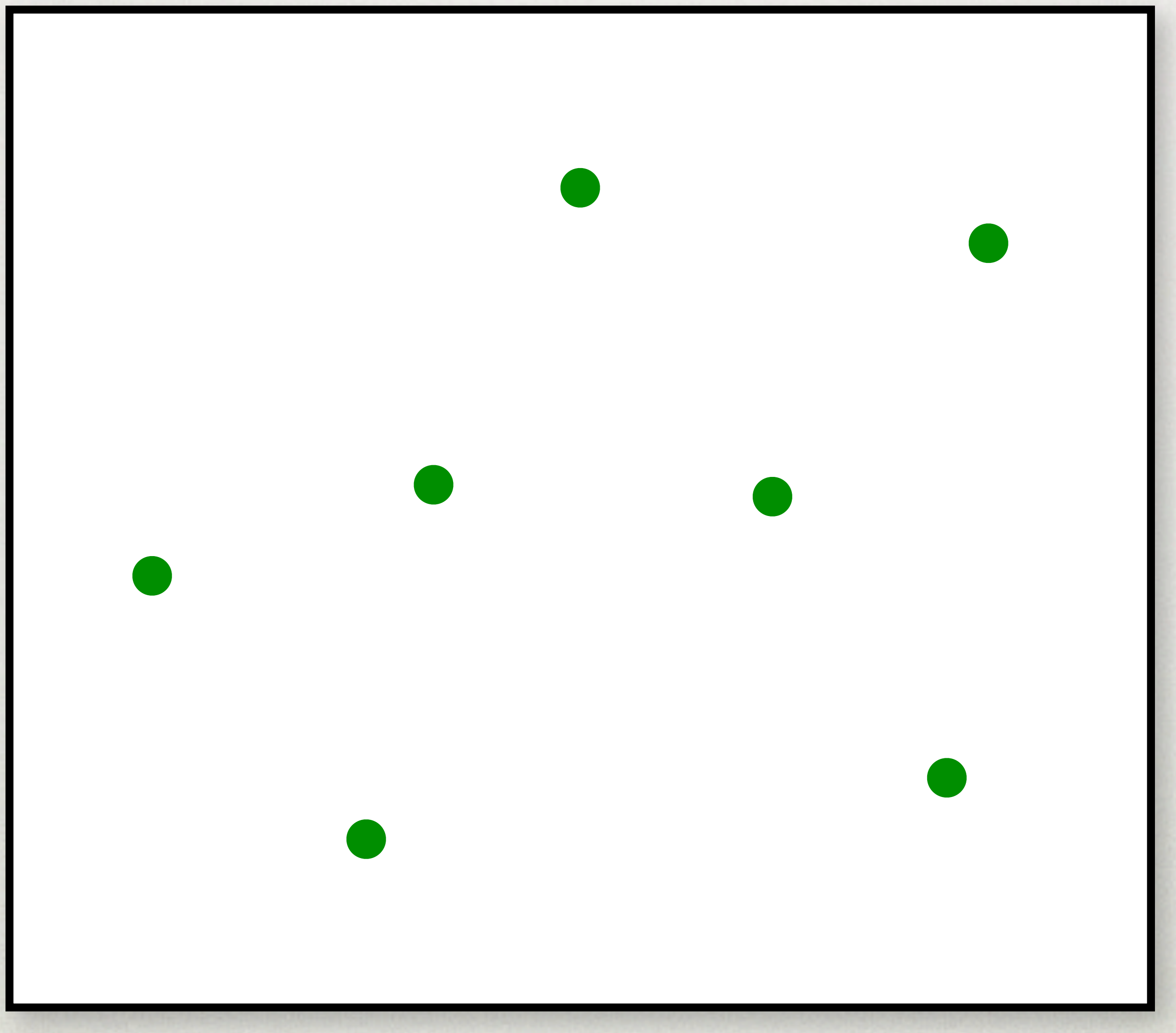
* first two steps identical to regular photon mapping

Algorithm



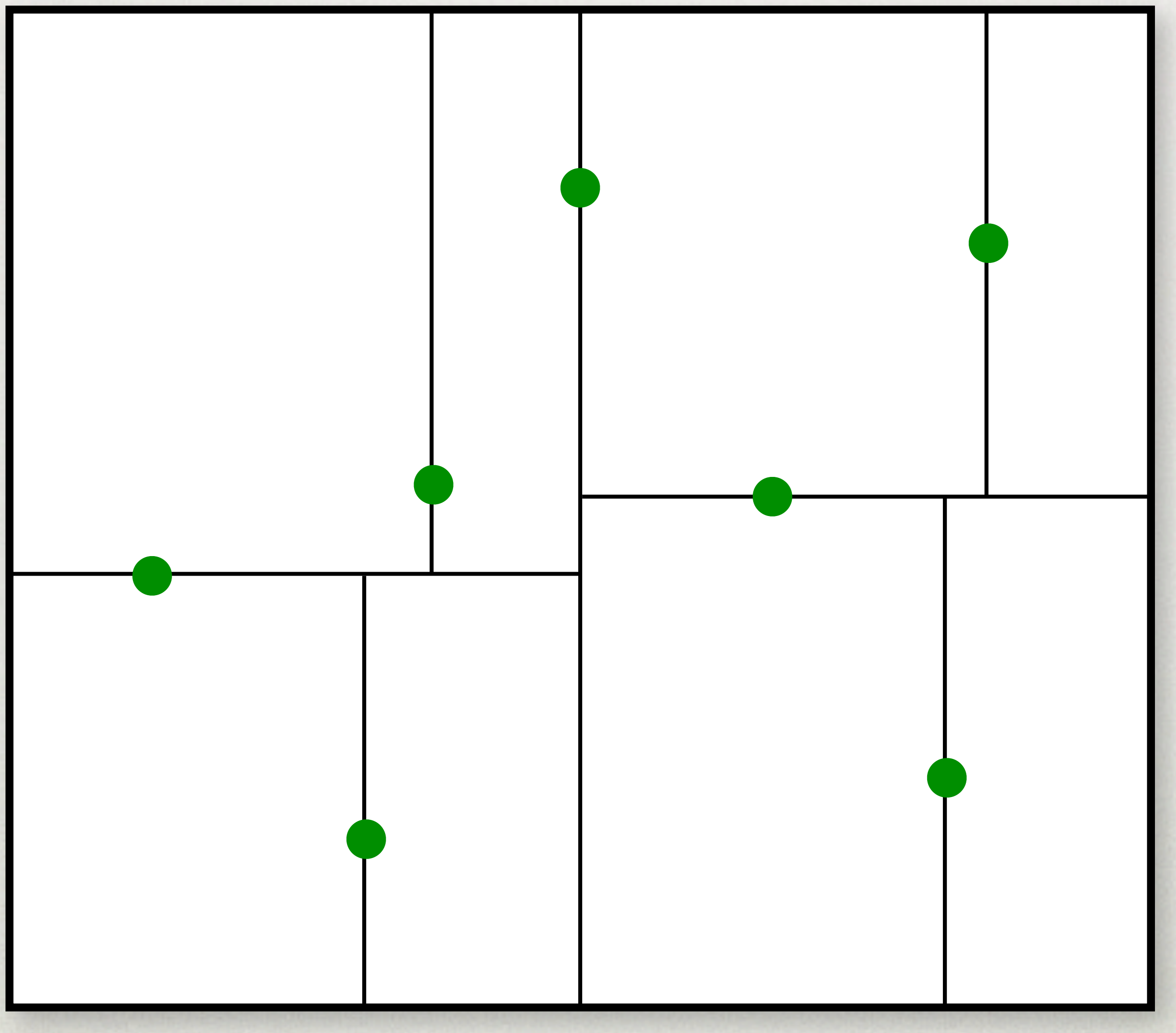
1) Shoot photons from light sources.

Algorithm



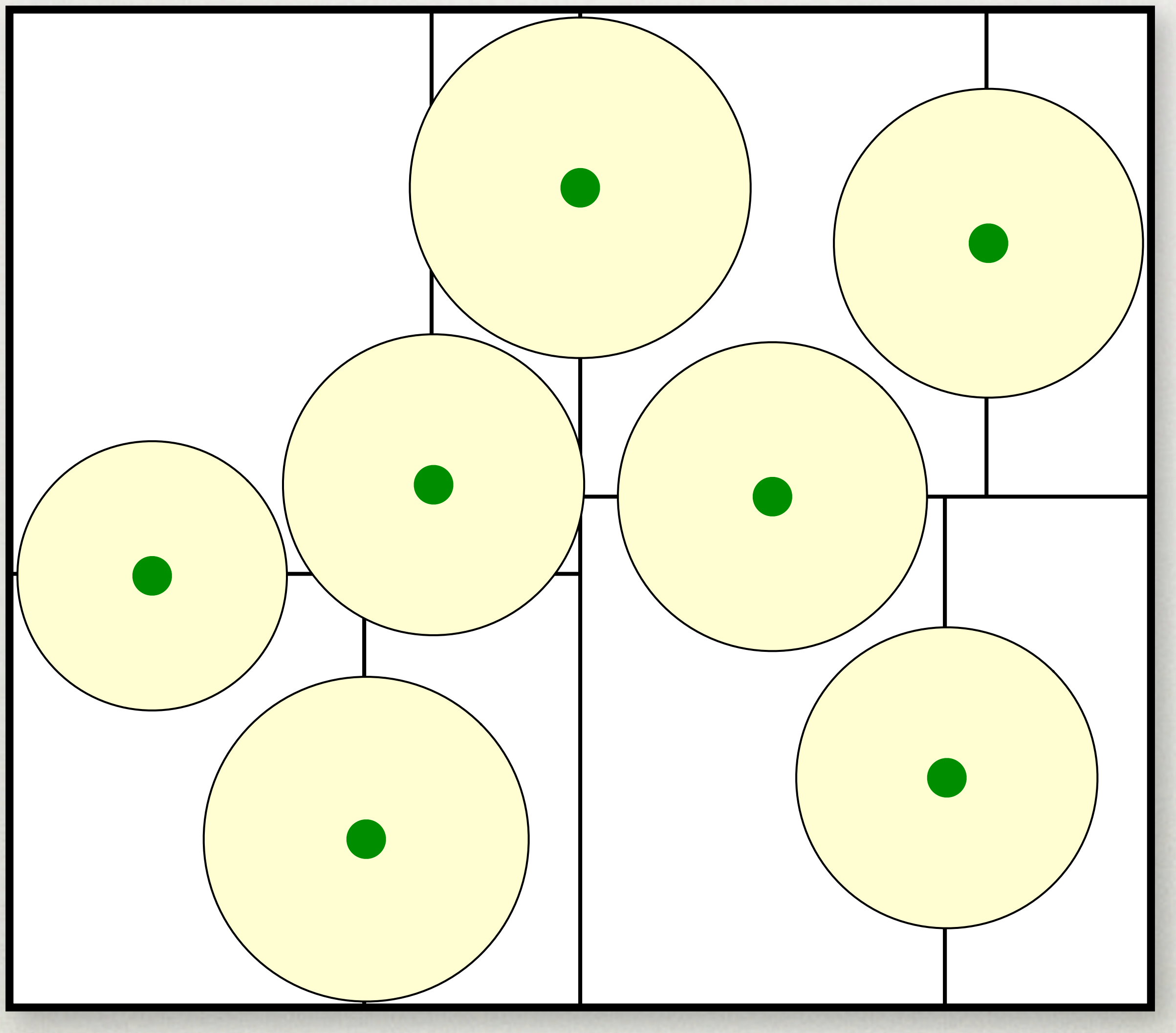
2) Construct a balanced kD-tree for the photons.

Algorithm



2) Construct a balanced kD-tree for the photons.

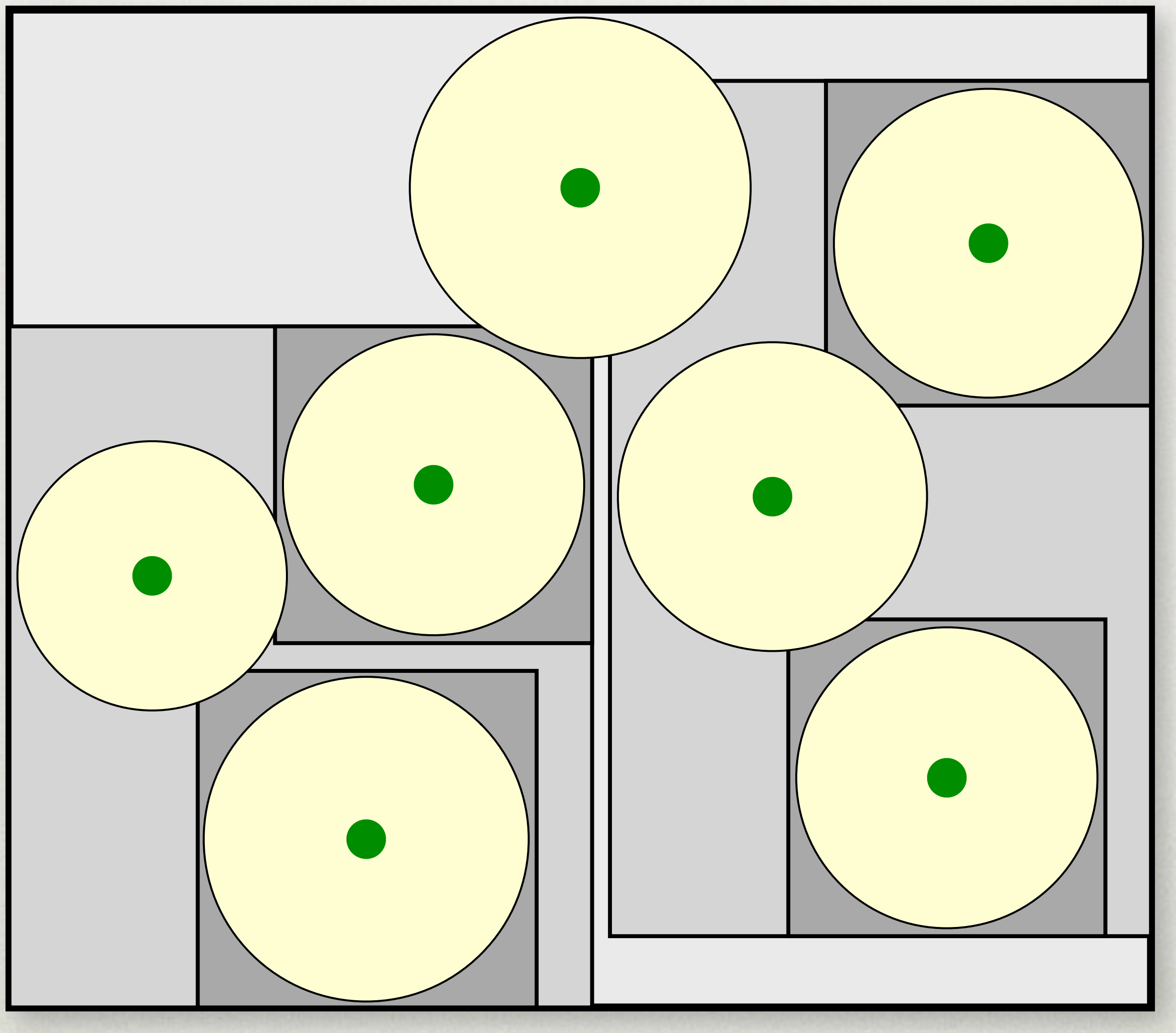
Algorithm



3) Assign a radius for each photon (*photon-discs*).
Adaptive: perform k-NN search at each photon

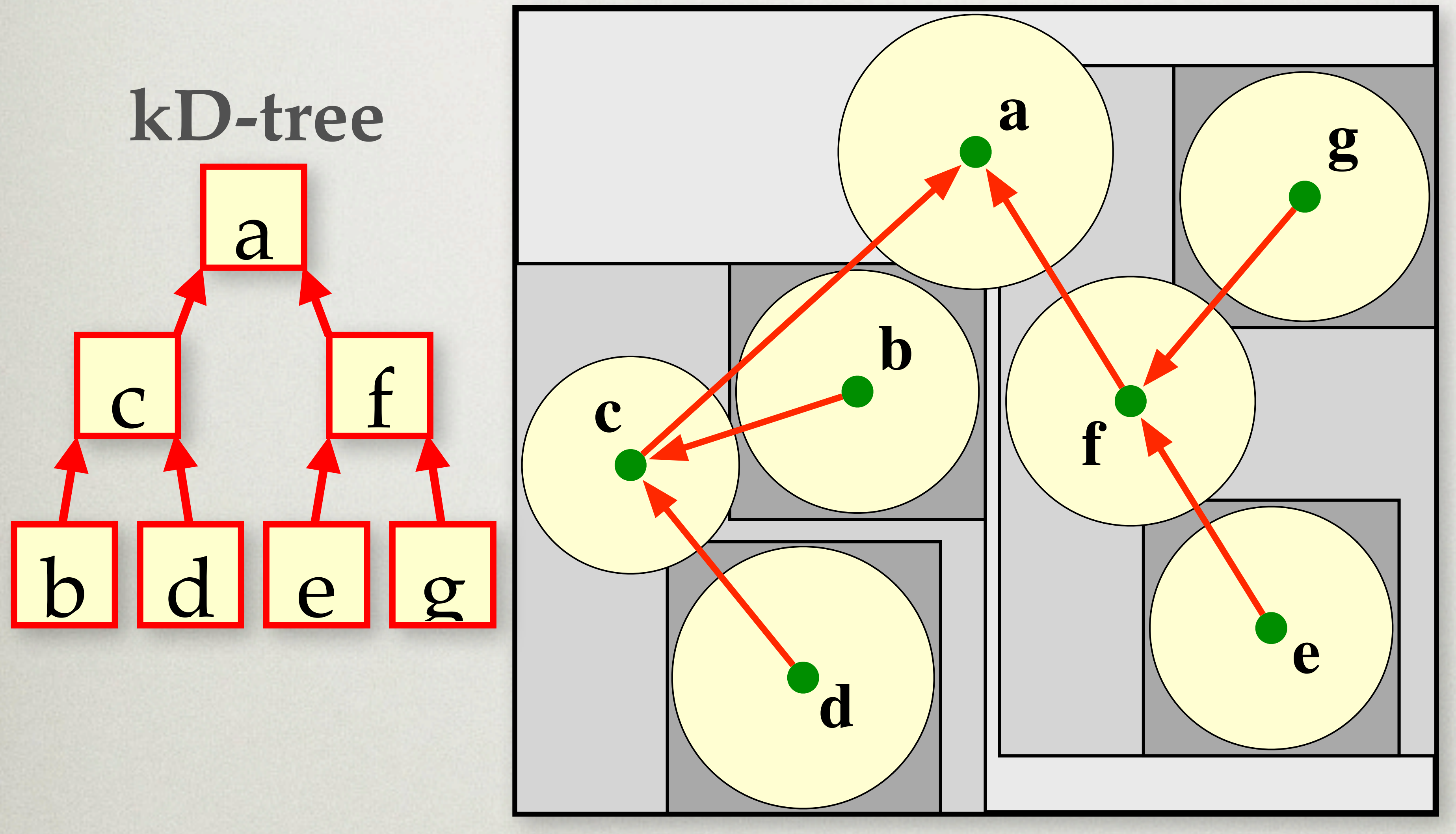
* if we use a fixed kernel, then each radius is the same, otherwise the radius is computed from the local density of each photon

Algorithm



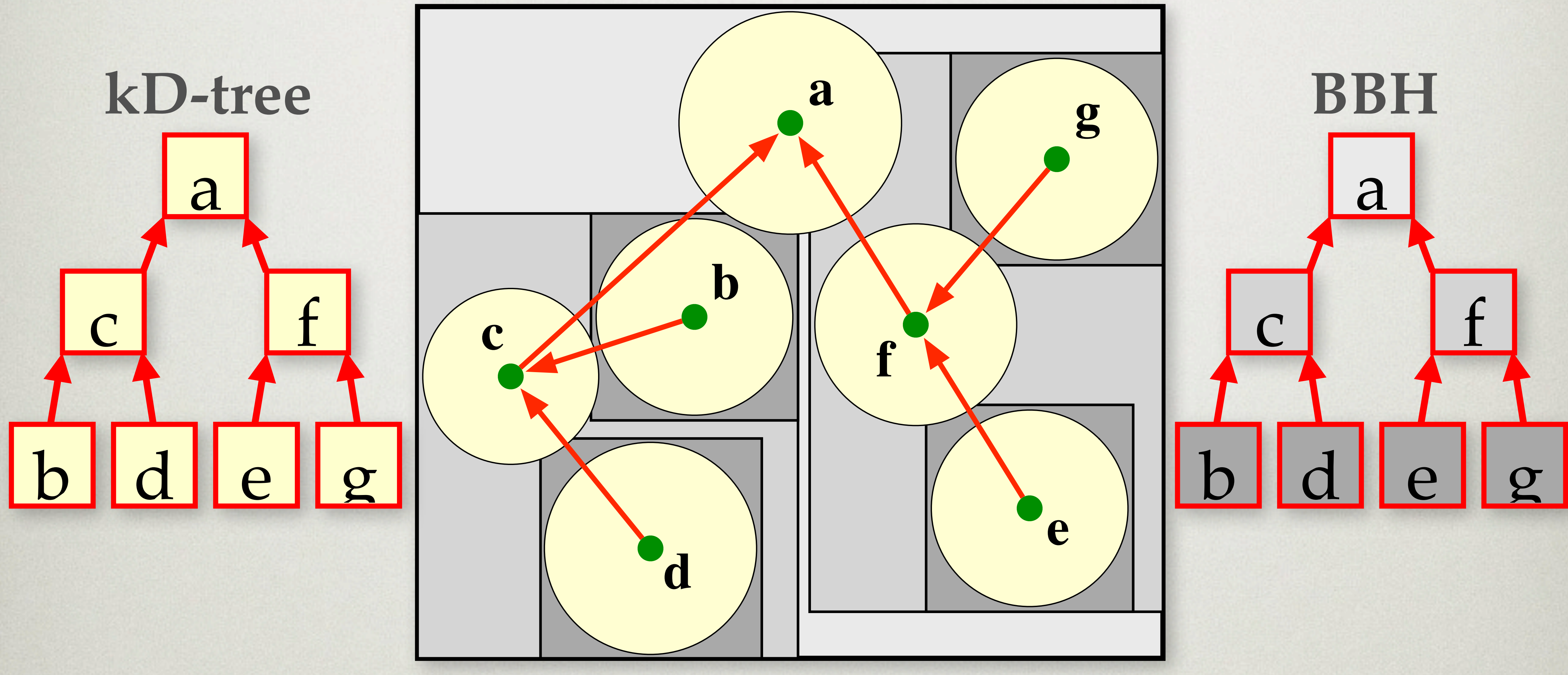
4) Create a bounding-box hierarchy over photon-discs

Algorithm



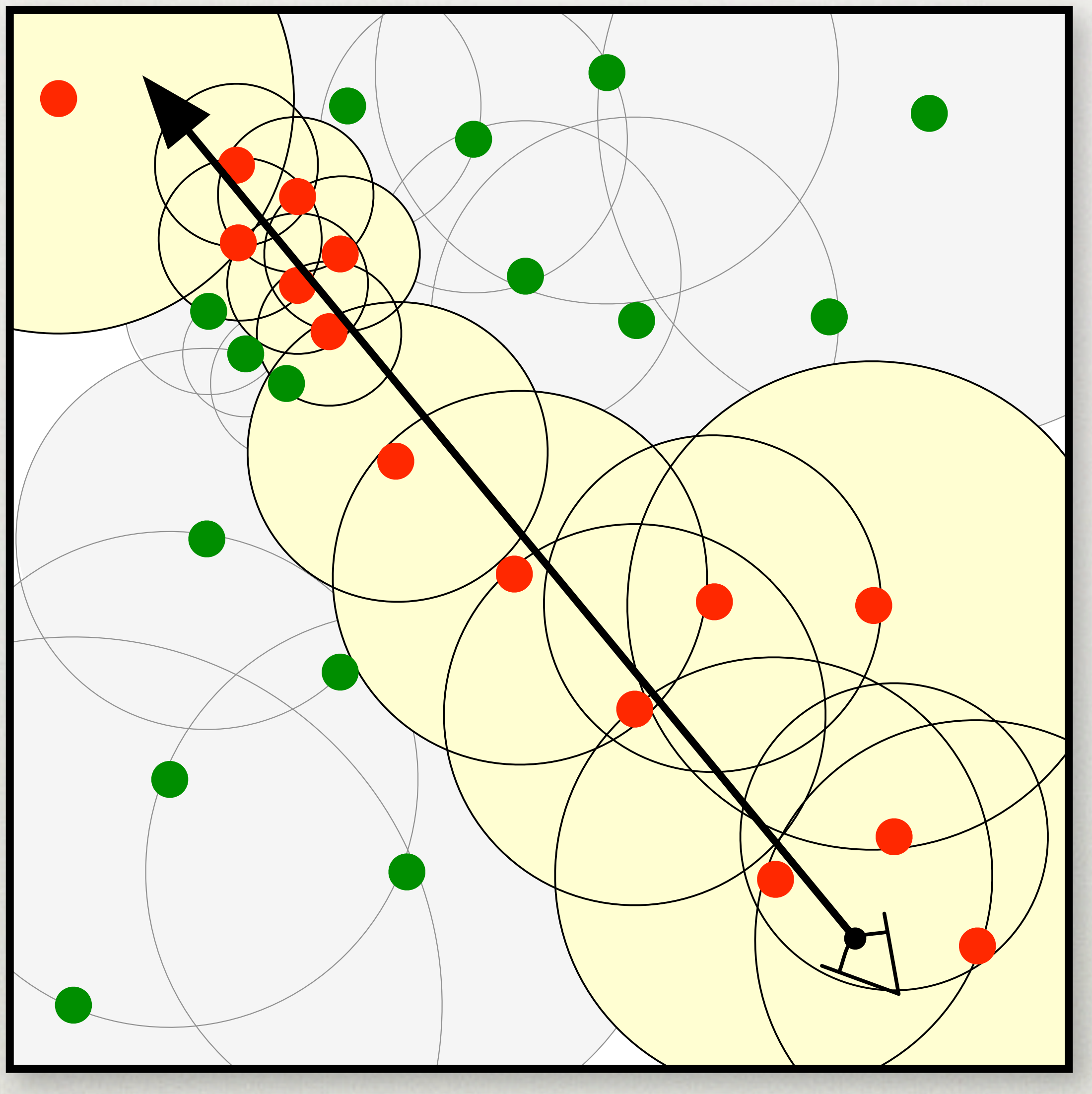
4) Create a bounding-box hierarchy over photon-discs

Algorithm



4) Create a bounding-box hierarchy over photon-discs
reuse hierarchical structure of kD-tree

Algorithm



5) Render: For each ray through the medium, accumulate all photon-discs that intersect ray.

Results

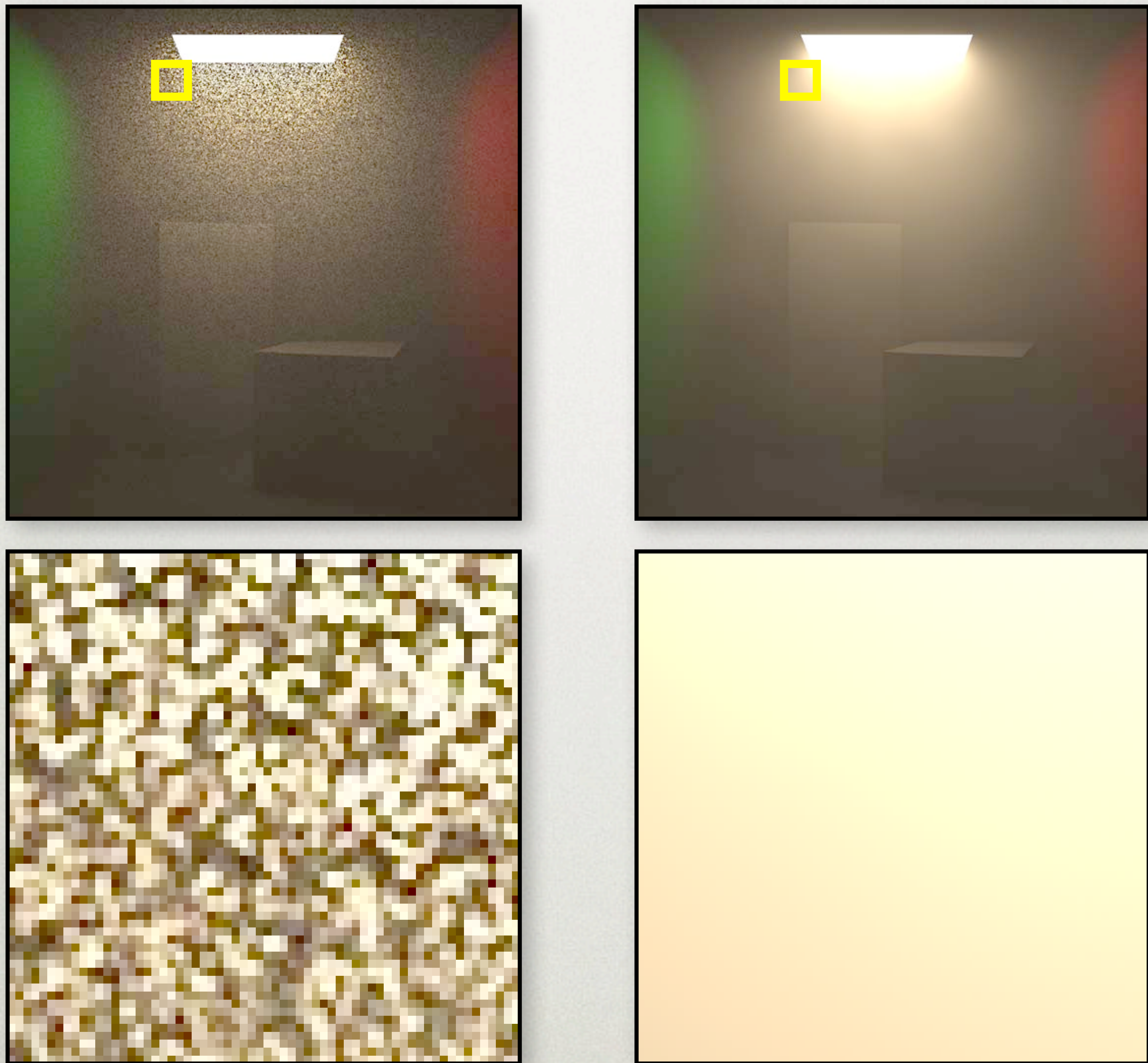
- 1K horizontal resolution
- 2.4 GHz Core 2 Duo (using one Core)
- Comparing ***identical*** photon maps

Smoky Cornell Box

Conv. Estimate | **Beam Estimate**

Smoky Cornell Box

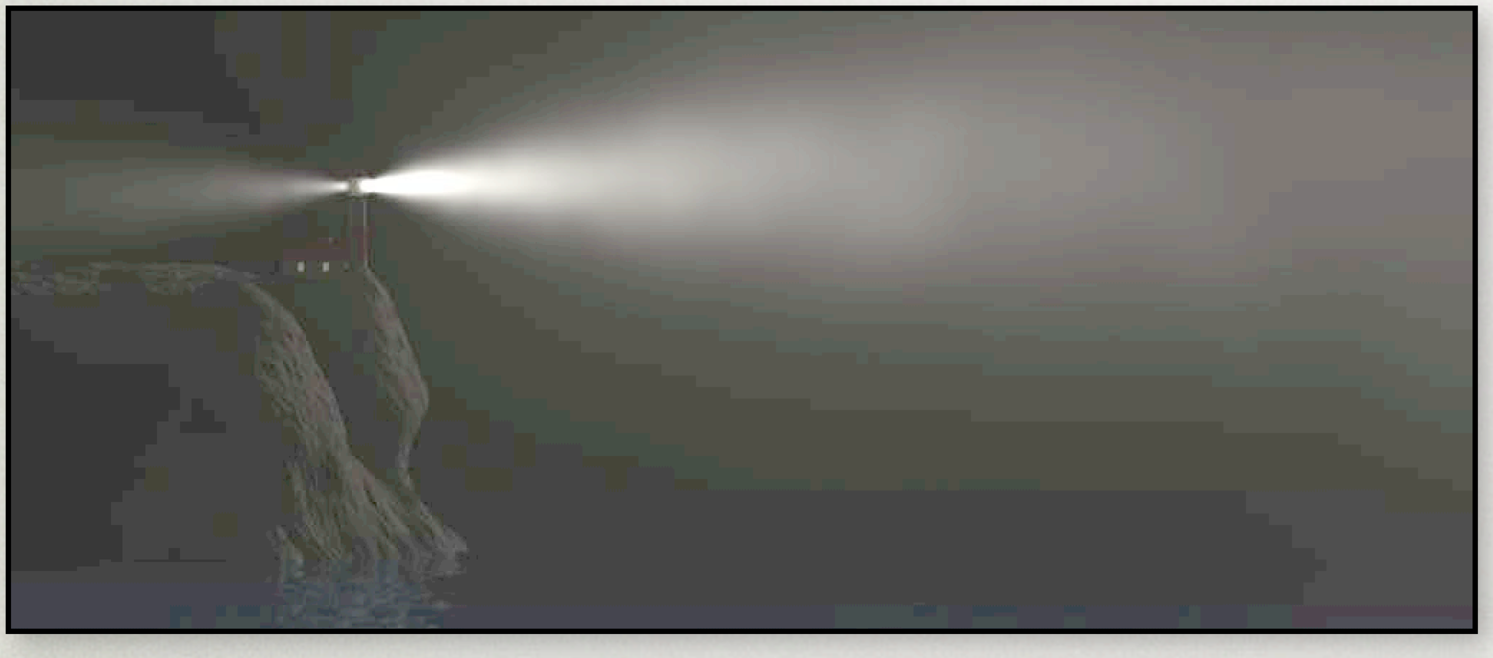
Conv. Estimate | **Beam Estimate**



(4:03) (3:35)

Lighthouse

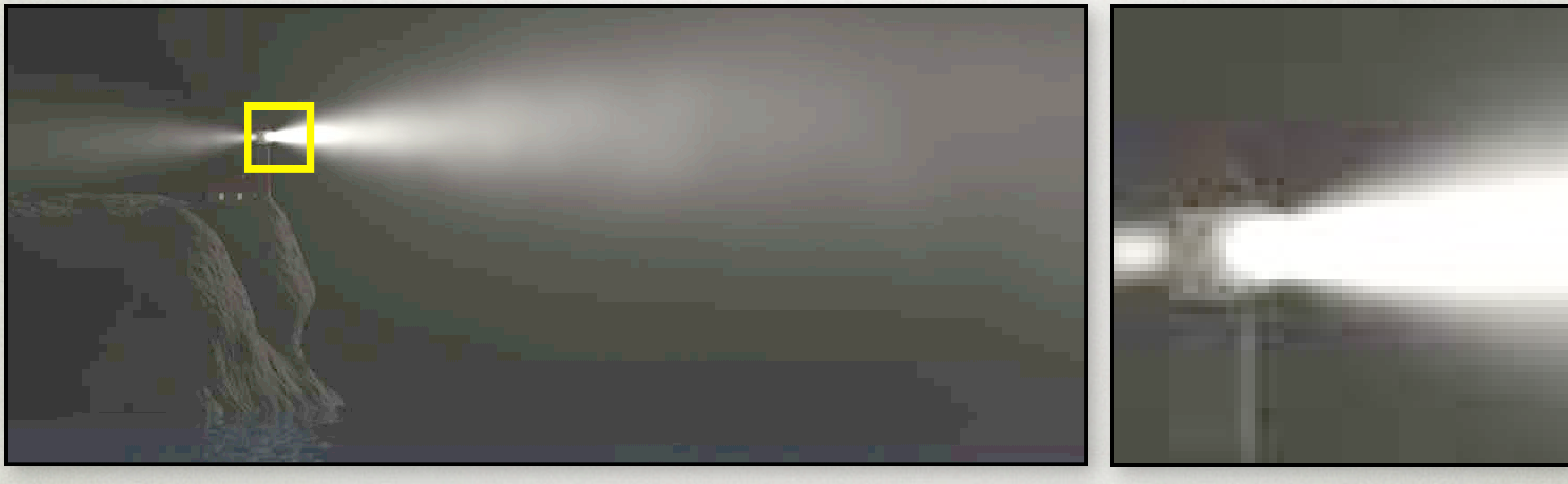
Beam Estimate

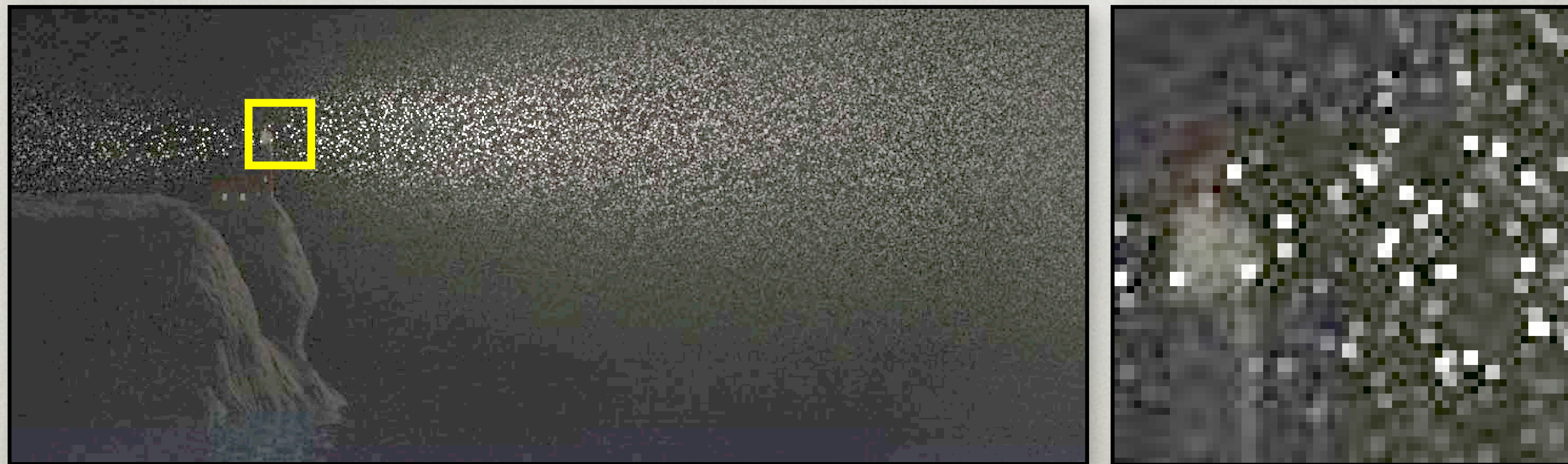


Conventional Estimate

Lighthouse

Beam Estimate (1:05)





Conventional Estimate (1:12)

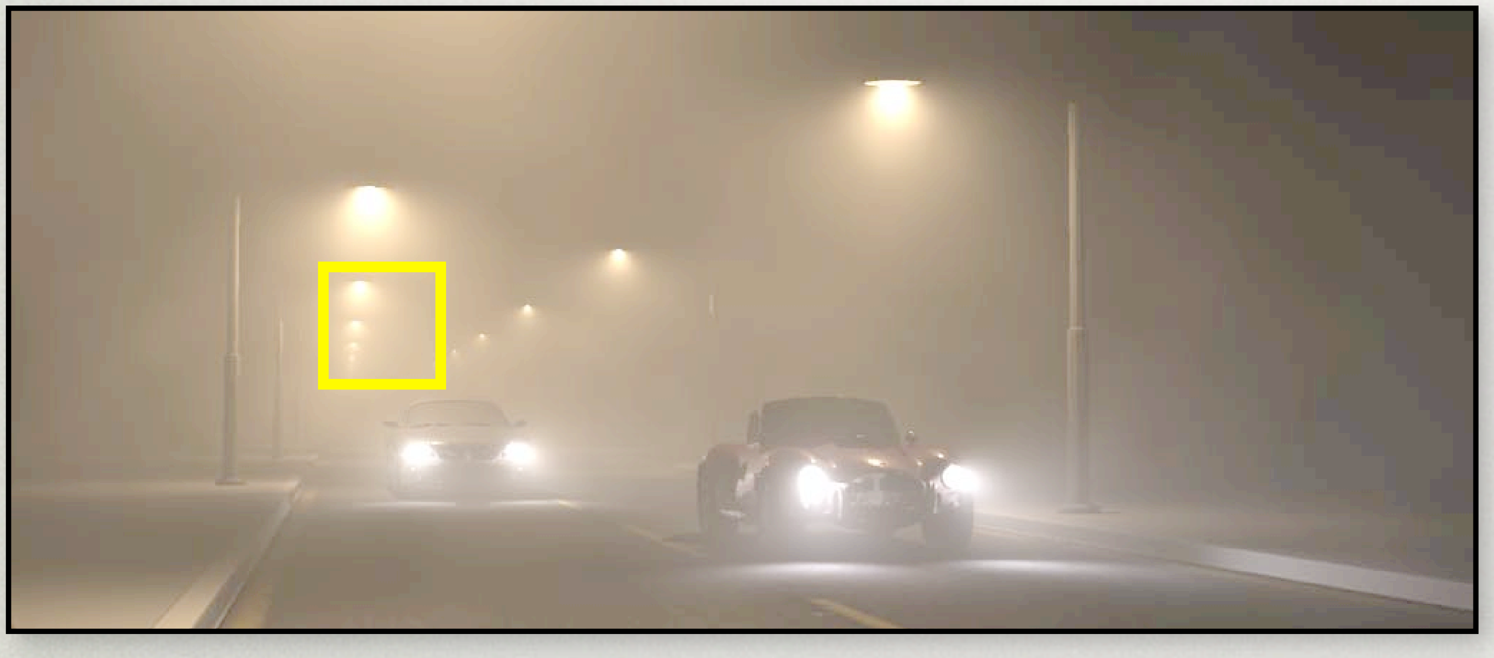
Cars on Foggy Street

Beam Estimate

Conventional Estimate

Cars on Foggy Street

Beam Estimate (1:53)



Conventional Estimate (2:02)

Summary

- Theoretical reformulation of PM (*measurement equation*)
- Beam radiance estimate
  - Eliminates ray-marching (and all high-frequency noise) in PM
    - Same photon map as conv. PM
    - Can handle adaptive search radius

Questions?