State of the Art in Photon Density Estimation:

Participating Media Basics

Wojciech Jarosz

THURSDAY, 9 AUGUST 2:00 PM - 5:15 PM | Room 408B

Fog

Thursday, August 23, 12

- In this portion of the class we are interested in understanding how to render scattering media.
- so far we have assumed that photons can travel unobstructed between objects.
- In reality, the “solid objects” in our world are embedded within a medium.
- In fog for instance, the interaction of light with the tiny water particles in the air can create stunning effects such as the volumetric shadow beams emanating from the trees in this image.

Clouds & Crepuscular rays

Thursday, August 23, 12

- clouds are another example of this, and since we see them much further away, we can very easily see their heterogeneous nature

Surface or Volume?

- In fact, the concept of “solid objects” is really a simplification, and the boundary between media and surfaces is actually not very easy to define.
- for instance, this iceberg may appear as a solid object, but really the light penetrates through it and you could also think of it as a medium

Surface or Volume?

Thursday, August 23, 12

- In reality its more accurate to think of solid objects as just the boundary between different media, in this case air and jade.
- And by generalizing photon mapping to account for this, we can get effects such as subsurface scattering of light past the boundary

Antelope Canyon, Az.

- Its also important to point out that we cannot consider surface illumination and media illumination in isolation.
- In this photograph almost all illumination on the walls is indirect light that has either bounced off of the small illuminated patches on the ground, or off of tiny dust particles in the air.
- There is a coupling between these two, which is quite easy to account for with photon mapping.

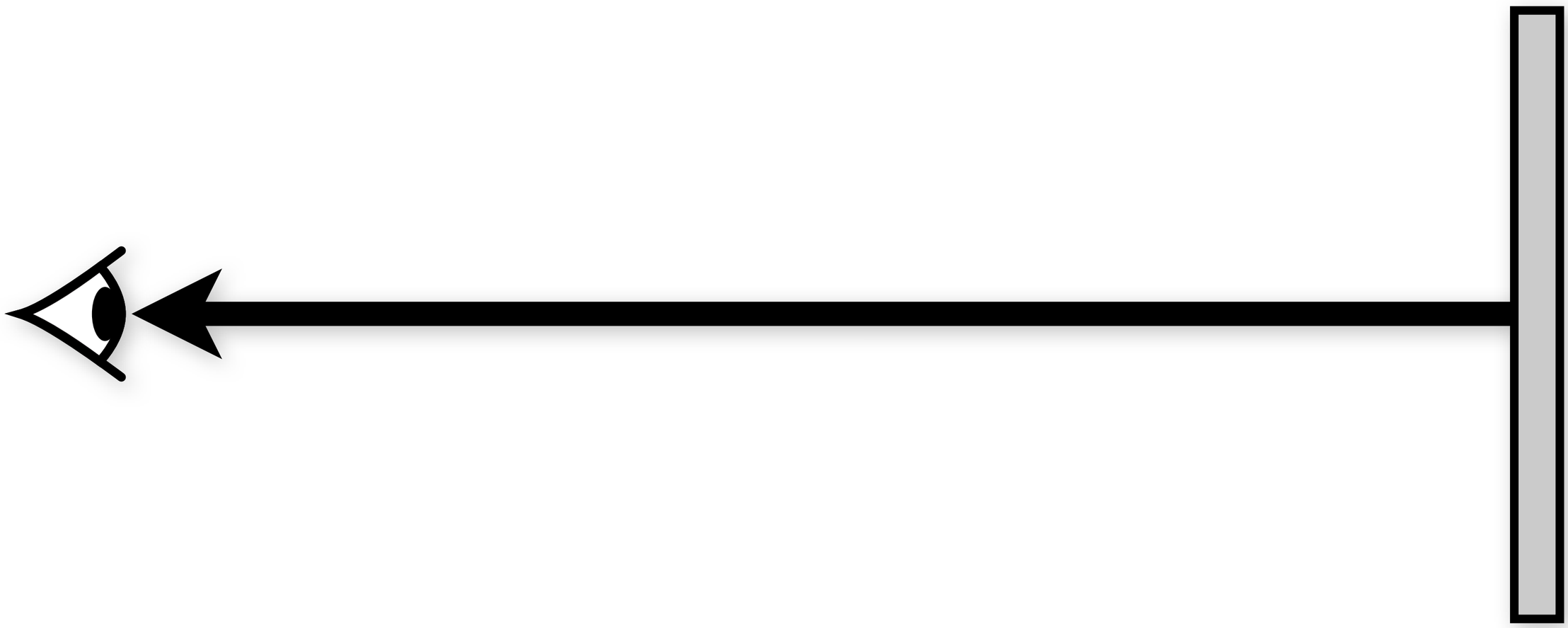
Outline

- Theoretical background
- Extending photon mapping to media

Participating Media

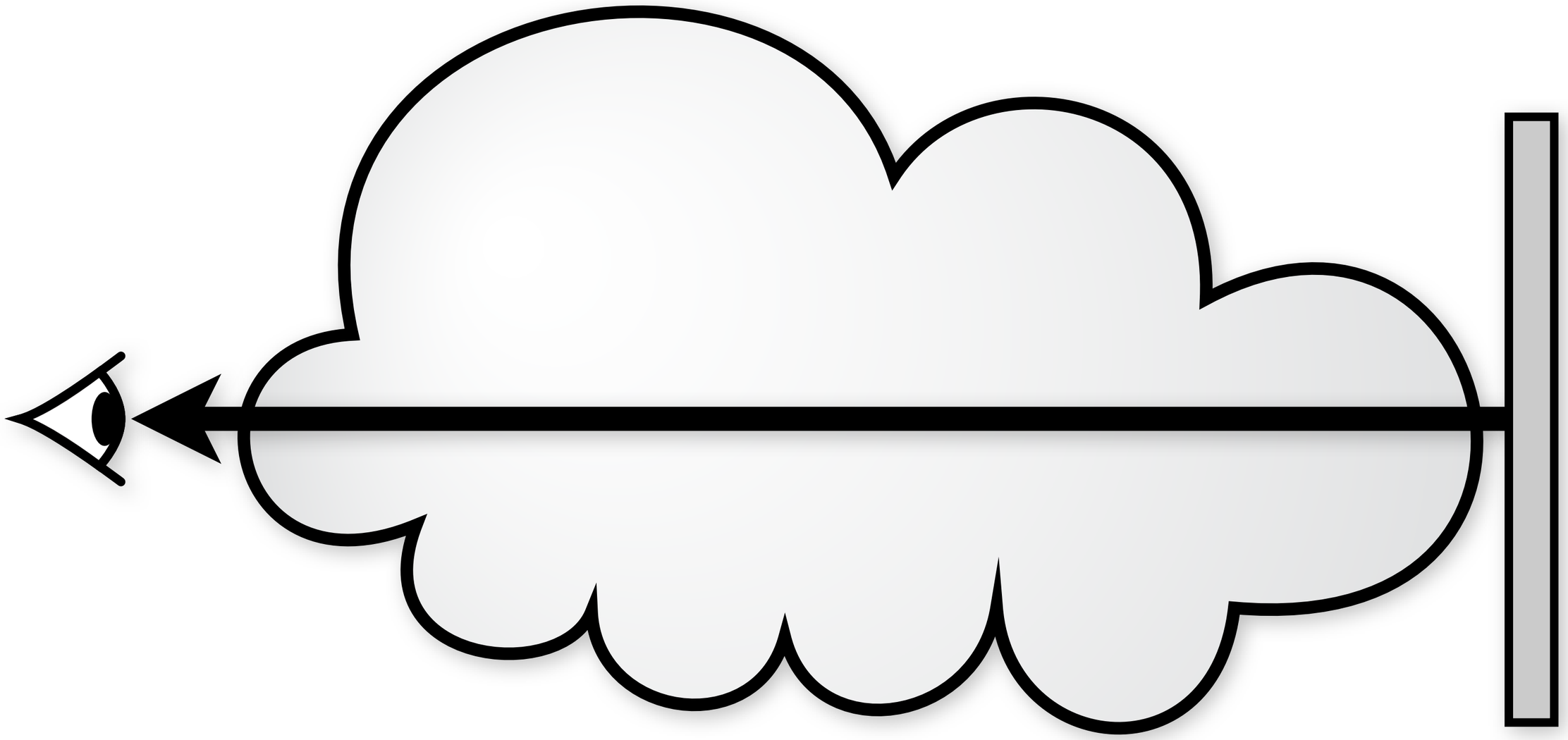
- Part. media can be thought of as a collection of particles suspended in a vacuum.
- As a photon travel through the scene, it may interact with the medium by hitting one of these particles.
- We will not model each of these particles individually, since this would be impractical, but instead assume the particles are small and statistically model their aggregate properties.
- If we consider an infinitesimal segment of media, when a photon travels through this segment, a number of possible interactions might occur.

Participating Media



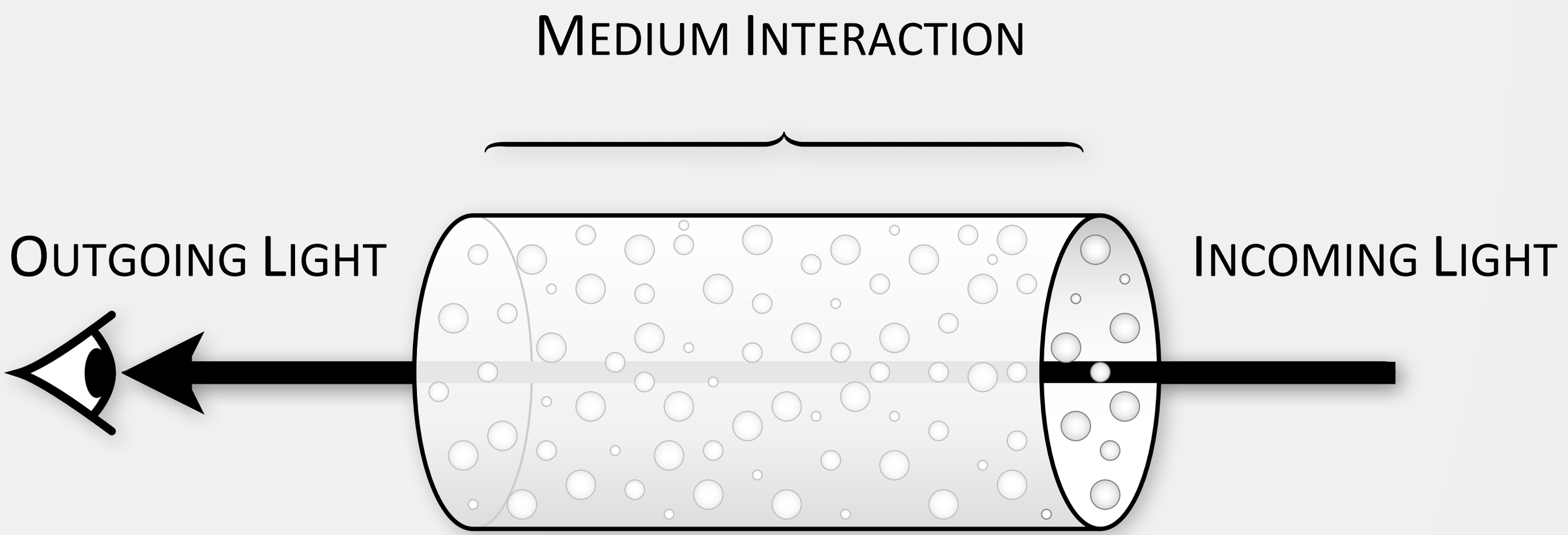
- Part. media can be thought of as a collection of particles suspended in a vacuum.
- As a photon travel through the scene, it may interact with the medium by hitting one of these particles.
- We will not model each of these particles individually, since this would be impractical, but instead assume the particles are small and statistically model their aggregate properties.
- If we consider an infinitesimal segment of media, when a photon travels through this segment, a number of possible interactions might occur.

Participating Media



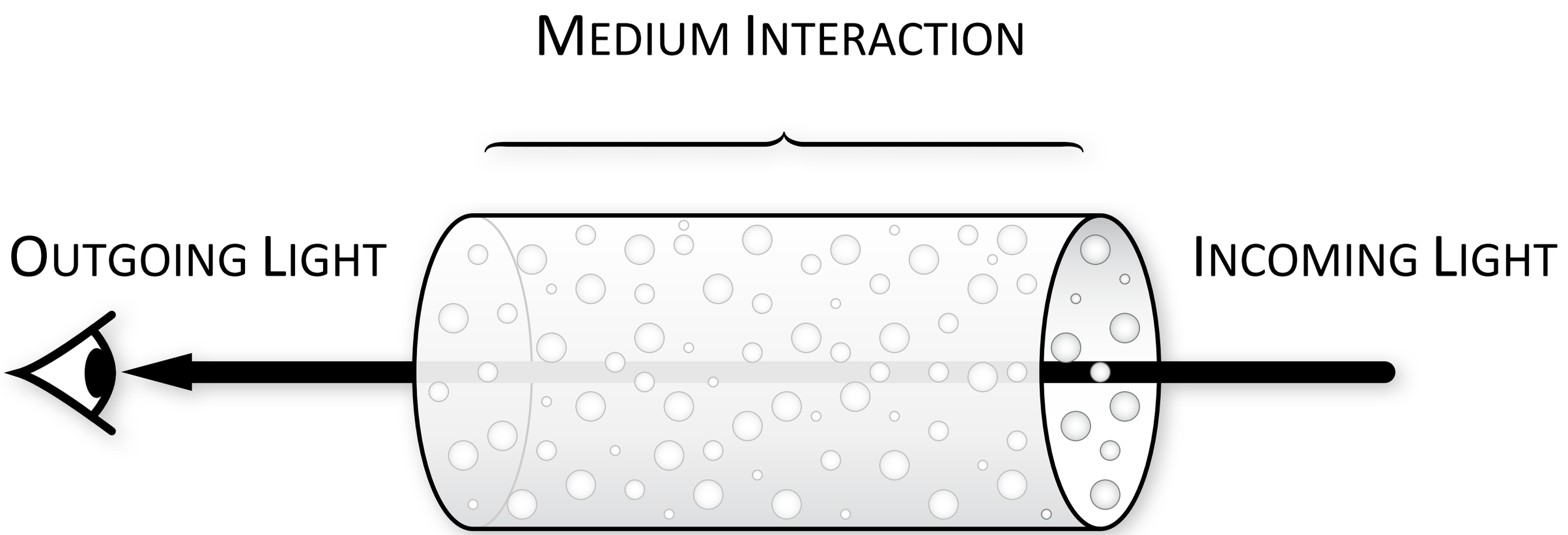
- Part. media can be thought of as a collection of particles suspended in a vacuum.
- As a photon travel through the scene, it may interact with the medium by hitting one of these particles.
- We will not model each of these particles individually, since this would be impractical, but instead assume the particles are small and statistically model their aggregate properties.
- If we consider an infinitesimal segment of media, when a photon travels through this segment, a number of possible interactions might occur.

Participating Media



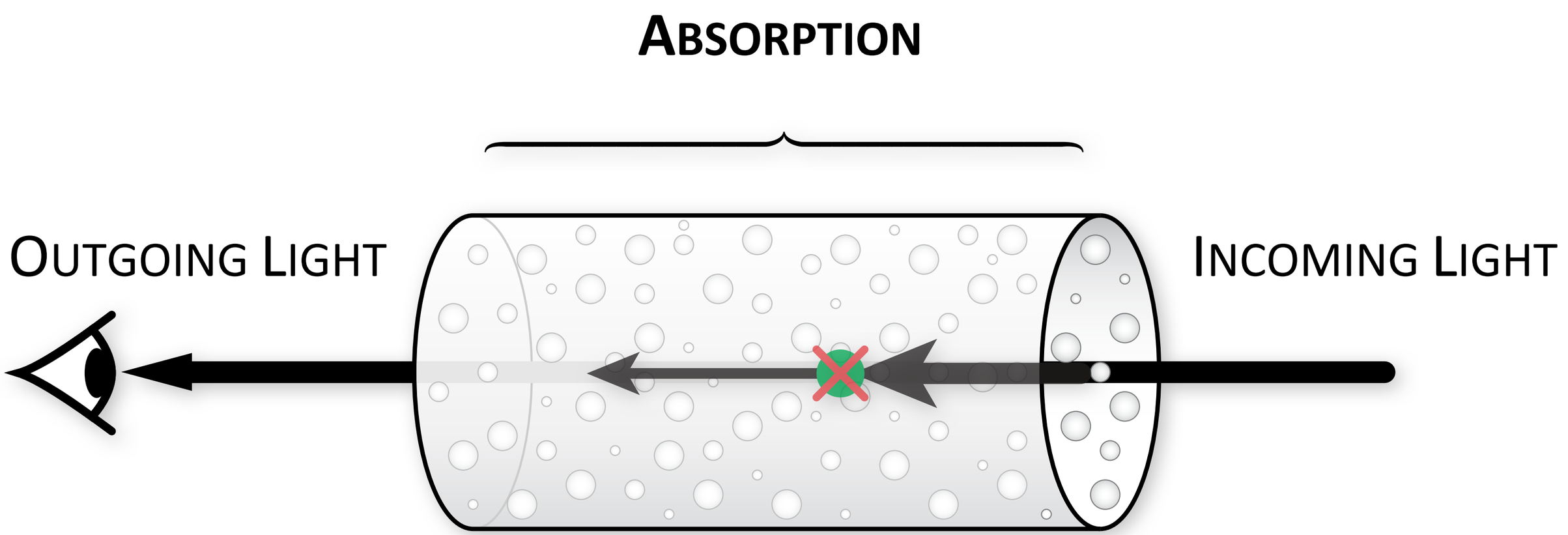
- Part. media can be thought of as a collection of particles suspended in a vacuum.
- As a photon travel through the scene, it may interact with the medium by hitting one of these particles.
- We will not model each of these particles individually, since this would be impractical, but instead assume the particles are small and statistically model their aggregate properties.
- If we consider an infinitesimal segment of media, when a photon travels through this segment, a number of possible interactions might occur.

Participating Media



- Part. media can be thought of as a collection of particles suspended in a vacuum.
- As a photon travel through the scene, it may interact with the medium by hitting one of these particles.
- We will not model each of these particles individually, since this would be impractical, but instead assume the particles are small and statistically model their aggregate properties.
- If we consider an infinitesimal segment of media, when a photon travels through this segment, a number of possible interactions might occur.

Participating Media

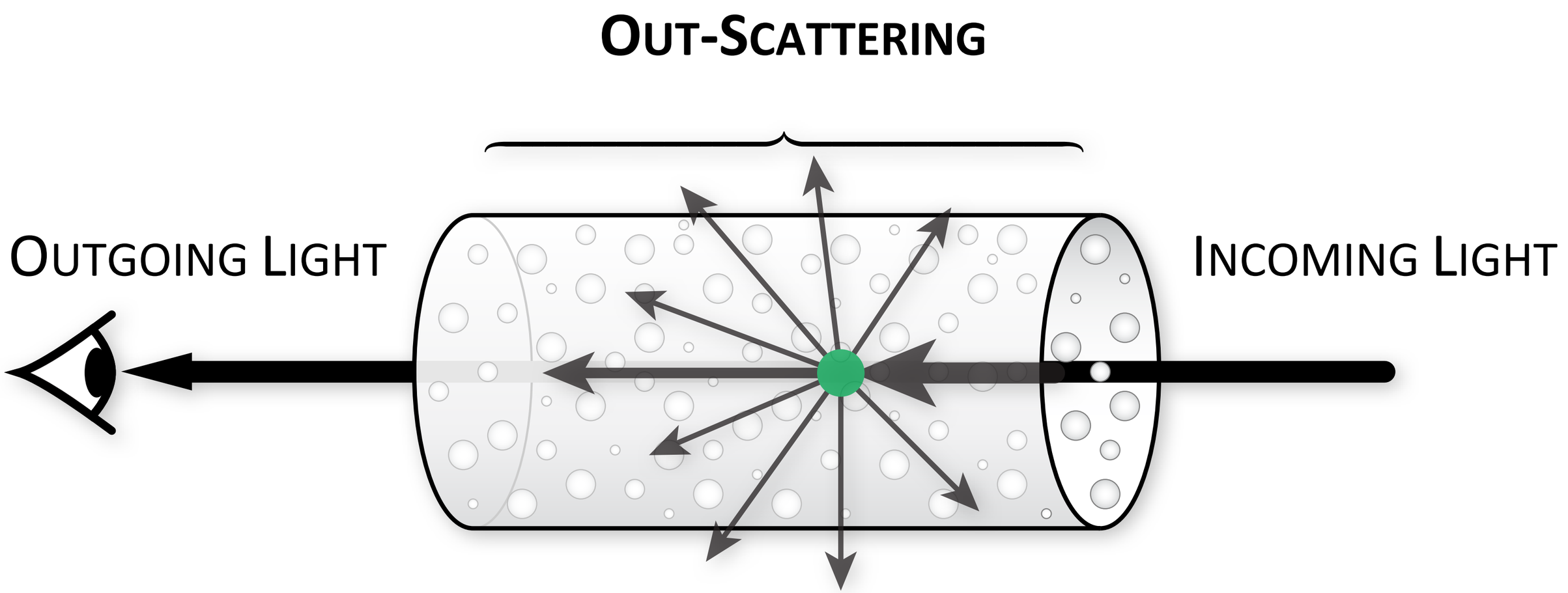


$\sigma_a(\mathbf{x})$: absorption coefficient [1/m]

Thursday, August 23, 12

- One possibility is an absorption event.
- This means that some portion of the photons that enter the infinitesimal segment become absorbed by the particles, which means that less photons leave the segment.
- The amount of radiance loss depends on the absorption coefficient of the medium. This can be any non-negative value. If this coefficient is 0, then no absorption happens, and the radiance remains unchanged. The larger the value, the denser and darker the medium becomes.
- Media such as black smoke can be well approximated using only absorption events.

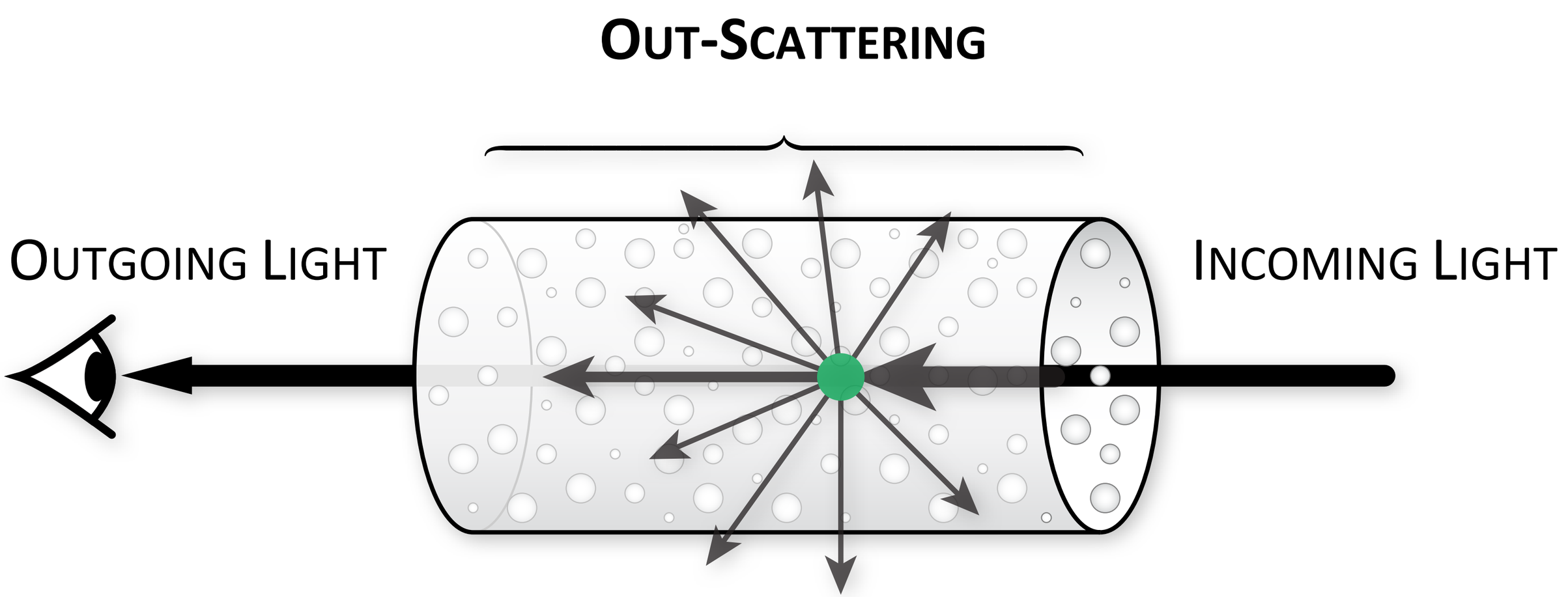
Participating Media



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

- Another thing that can happen is that the photons entering the medium segment are scattered into some other directions after hitting one of the particles.
- Out-scattering produces a net loss in radiance along the original direction.
- The amount of loss is determined by the scattering coefficient of the medium. As with the absorption coefficient it can take on any non-negative value, and if it is 0, no scattering happens.
- The directional distribution of scattering is described by the phase function, which is basically the analog of the BRDF, but for media

Participating Media



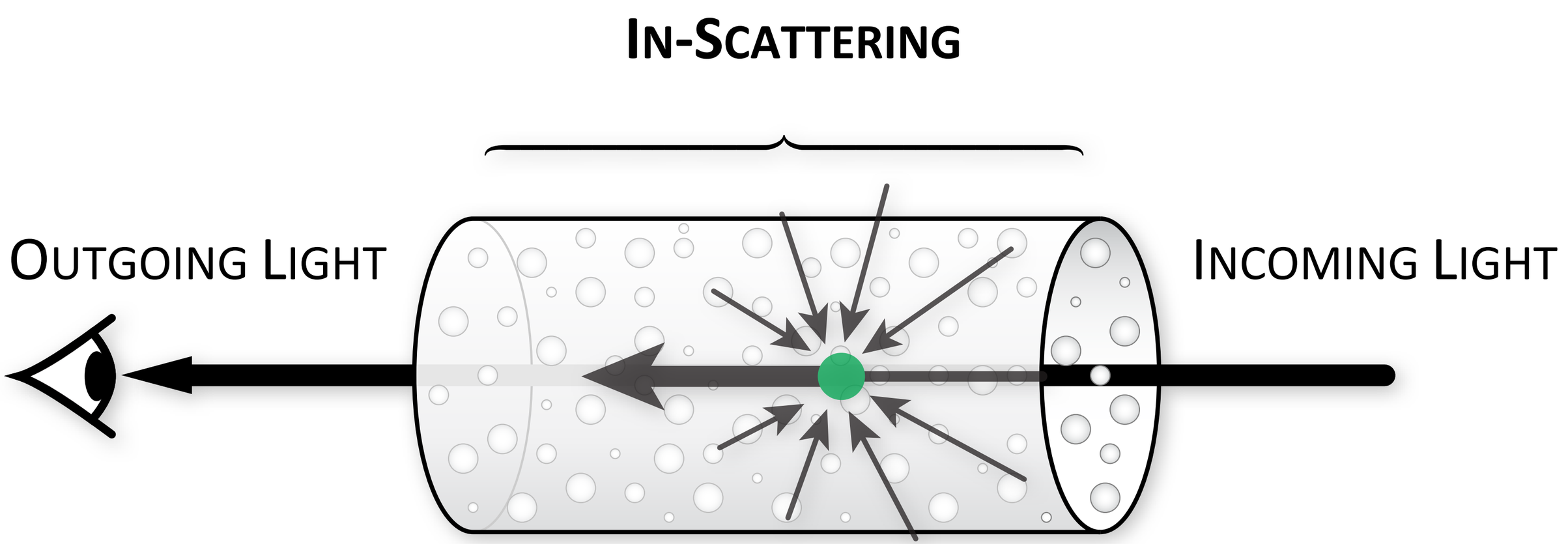
$\sigma_s(\mathbf{x})$: scattering coefficient [1/m]

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

Thursday, August 23, 12

- Another thing that can happen is that the photons entering the medium segment are scattered into some other directions after hitting one of the particles.
- Out-scattering produces a net loss in radiance along the original direction.
- The amount of loss is determined by the scattering coefficient of the medium. As with the absorption coefficient it can take on any non-negative value, and if it is 0, no scattering happens.
- The directional distribution of scattering is described by the phase function, which is basically the analog of the BRDF, but for media

Participating Media



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

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

- It is also possible for a photon originally traveling in a different direction to get scattered into the direction we are considering.
- This is called in-scattering and produces a net increase in radiance along the ray.
- Out-scattering and in-scattering are really just two sides of the same coin. A photon that out-scatters, reduces radiance in the original direction, but becomes the in-scattered photon in the new direction.
- We will see that this in-scattering is really the most complex and interesting lighting interaction to account for, and photon mapping will make this efficient

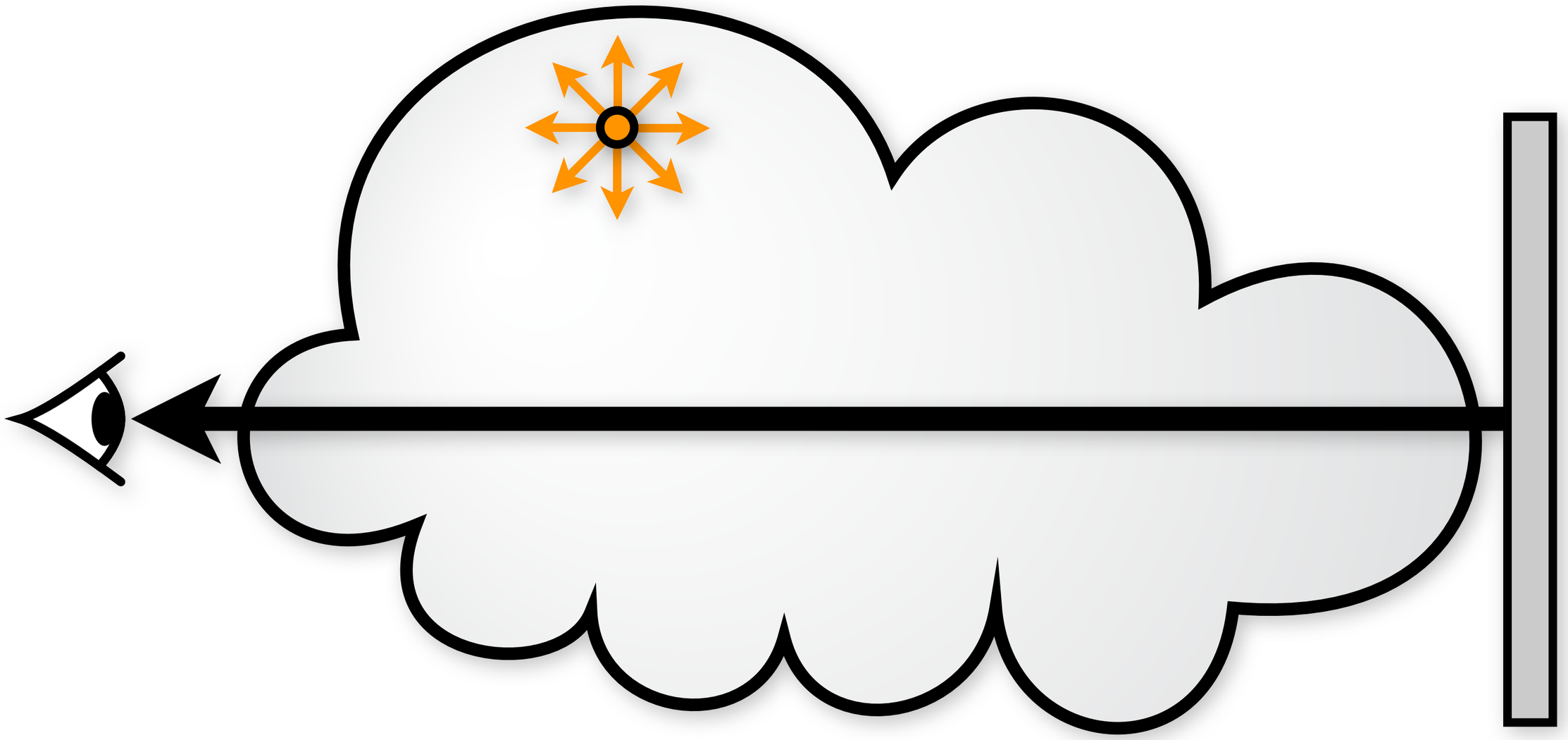
Photon Mapping

1) Photon tracing

2) Rendering / Radiance Estimation

- To account for media in the photon mapping algorithm, we will need to modify both the photon tracing stage, and also the rendering or radiance estimation stage

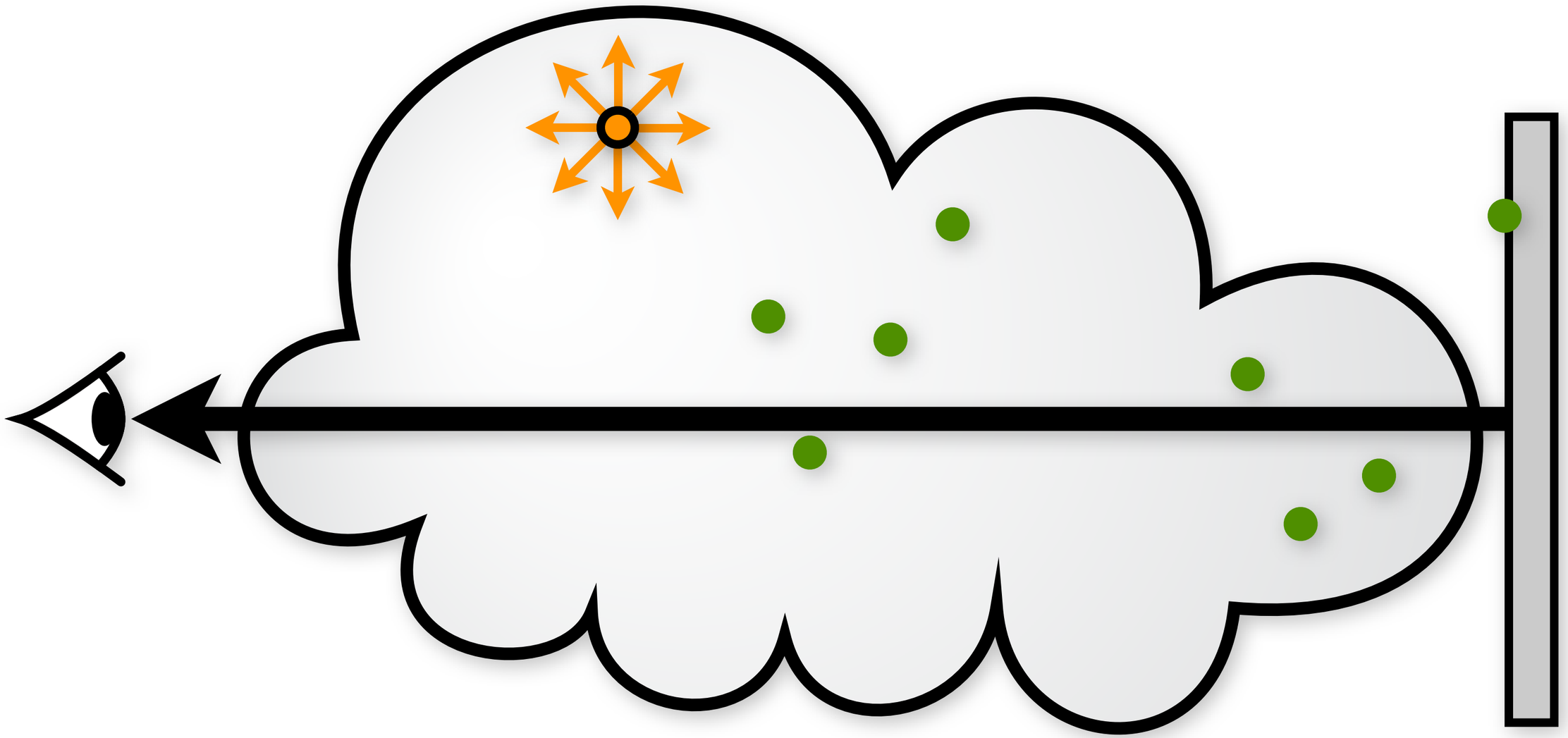
Photon Tracing in Participating Media



Thursday, August 23, 12

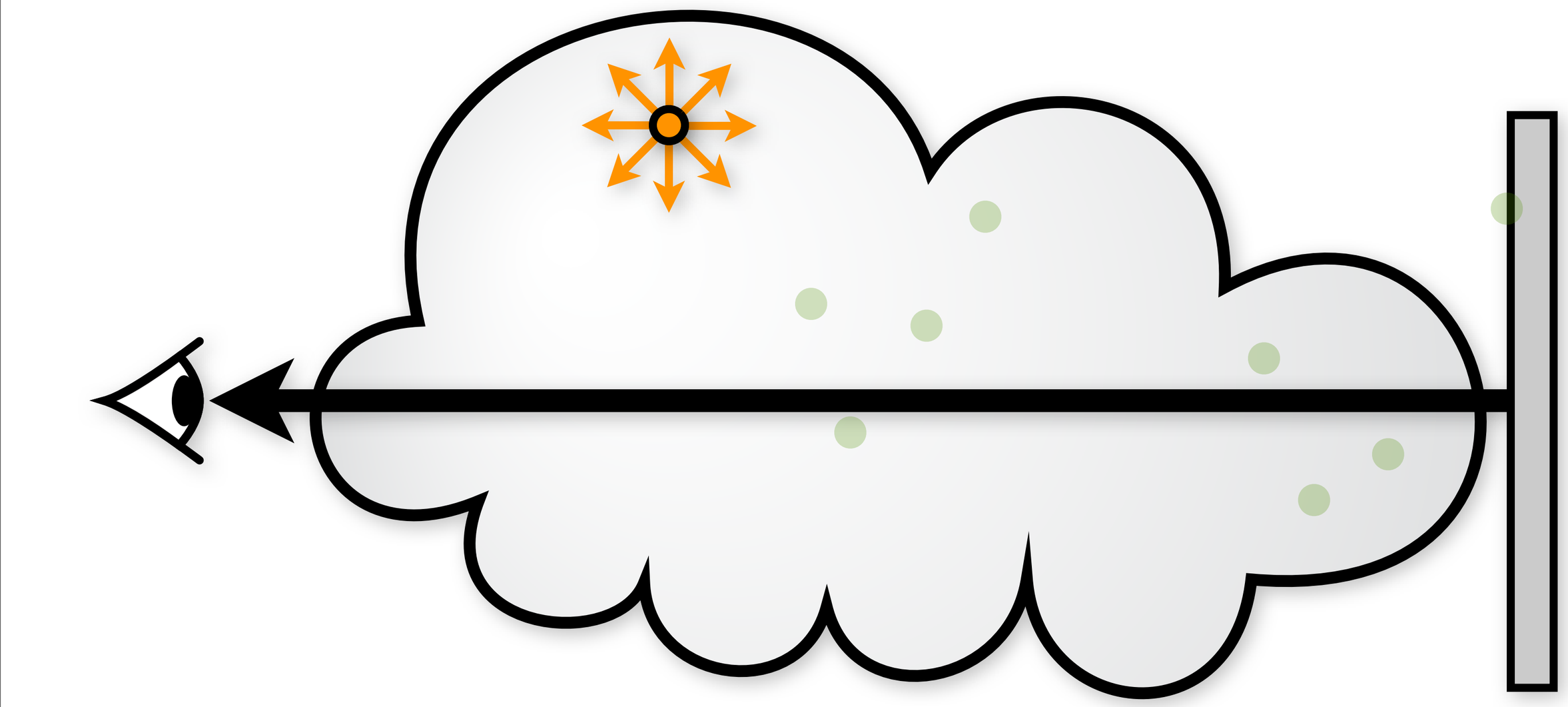
- For photon tracing, instead of emitted photons traveling directly to the surfaces, these photons now have some probability of scattering or being absorbed by the medium along their path.
- So we now trace paths that may have vertices within the medium, and at some point these paths will be probabilistically terminated when absorption happens.

Photon Tracing in Participating Media



- For photon tracing, instead of emitted photons traveling directly to the surfaces, these photons now have some probability of scattering or being absorbed by the medium along their path.
- So we now trace paths that may have vertices within the medium, and at some point these paths will be probabilistically terminated when absorption happens.

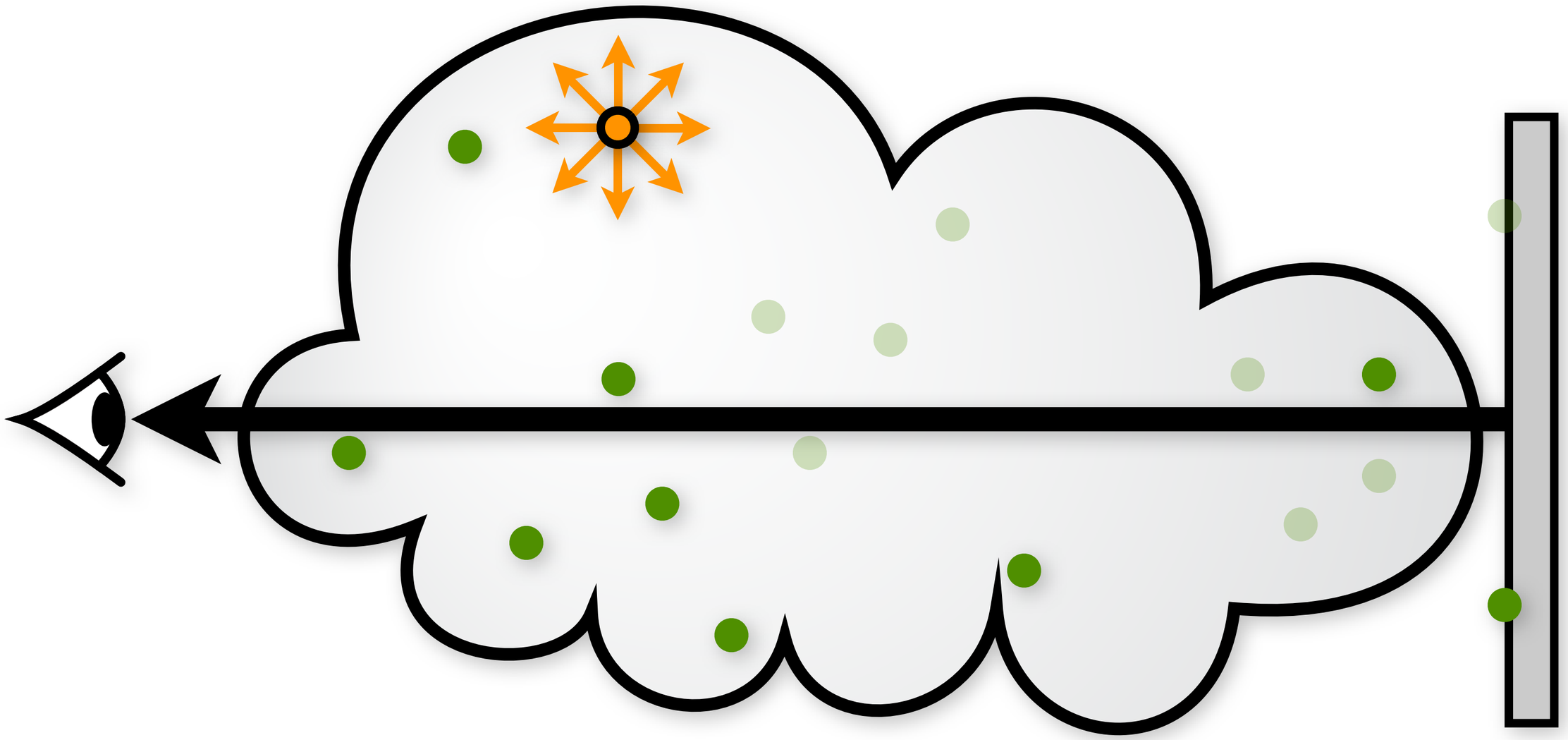
Photon Tracing in Participating Media



Thursday, August 23, 12

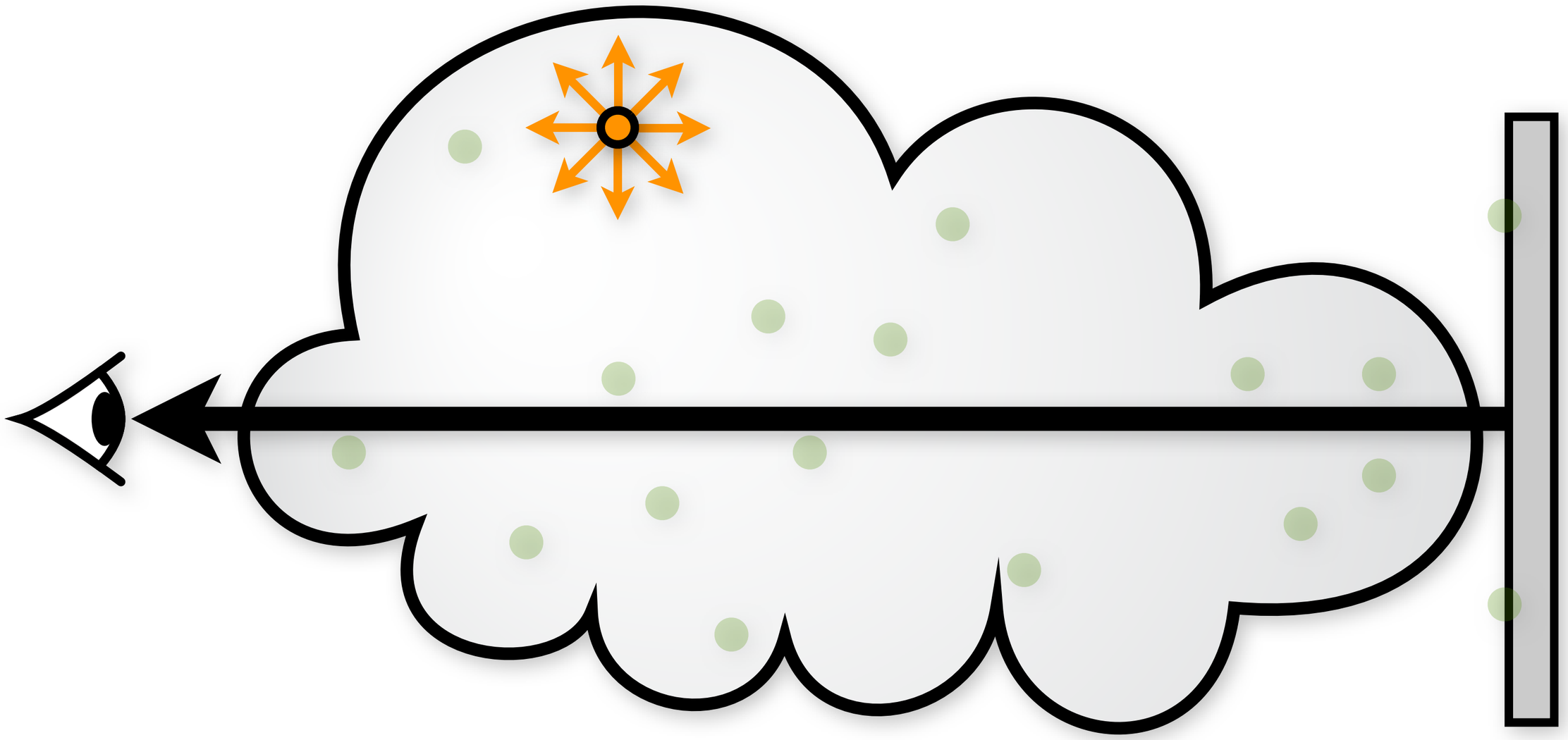
- For photon tracing, instead of emitted photons traveling directly to the surfaces, these photons now have some probability of scattering or being absorbed by the medium along their path.
- So we now trace paths that may have vertices within the medium, and at some point these paths will be probabilistically terminated when absorption happens.

Photon Tracing in Participating Media



- For photon tracing, instead of emitted photons traveling directly to the surfaces, these photons now have some probability of scattering or being absorbed by the medium along their path.
- So we now trace paths that may have vertices within the medium, and at some point these paths will be probabilistically terminated when absorption happens.

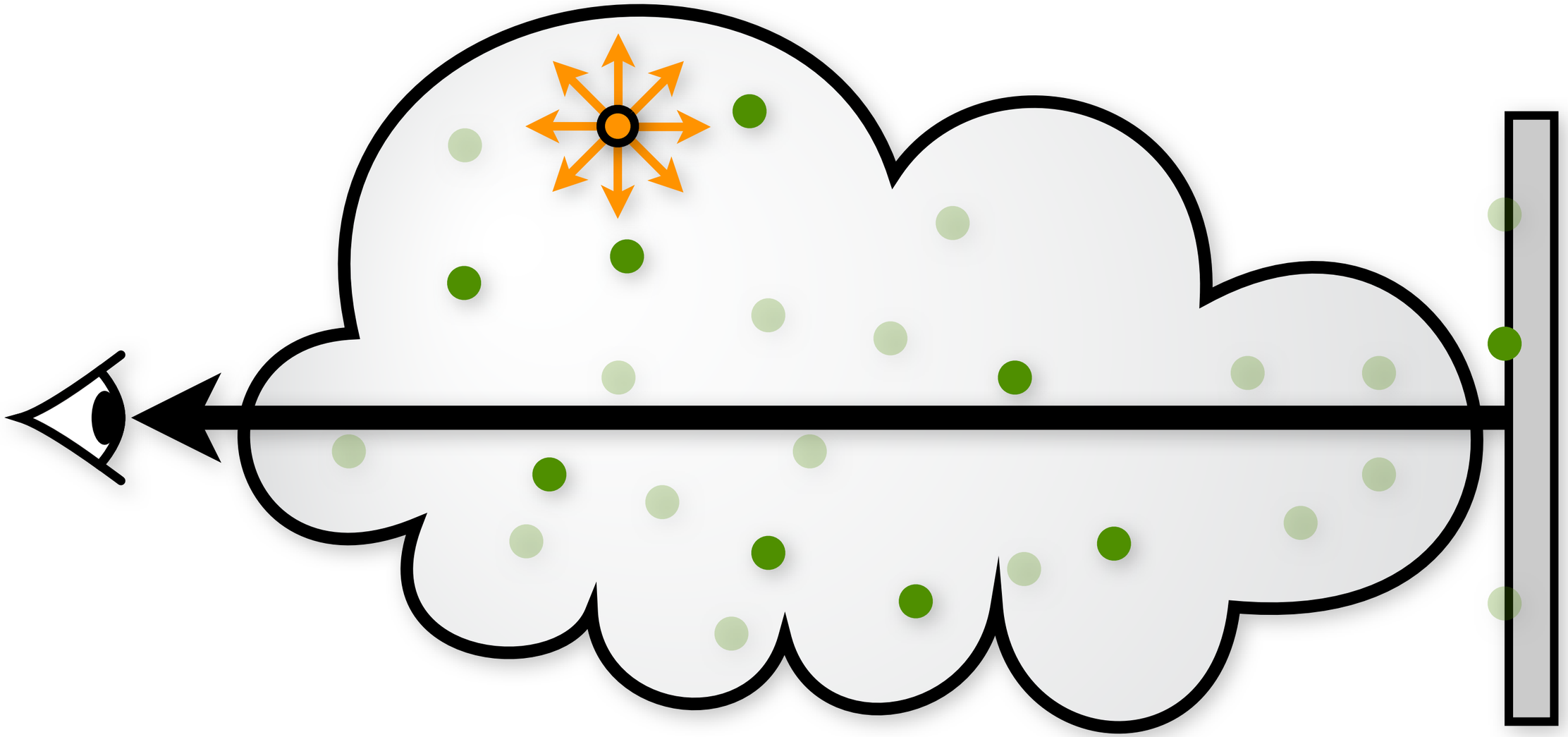
Photon Tracing in Participating Media



Thursday, August 23, 12

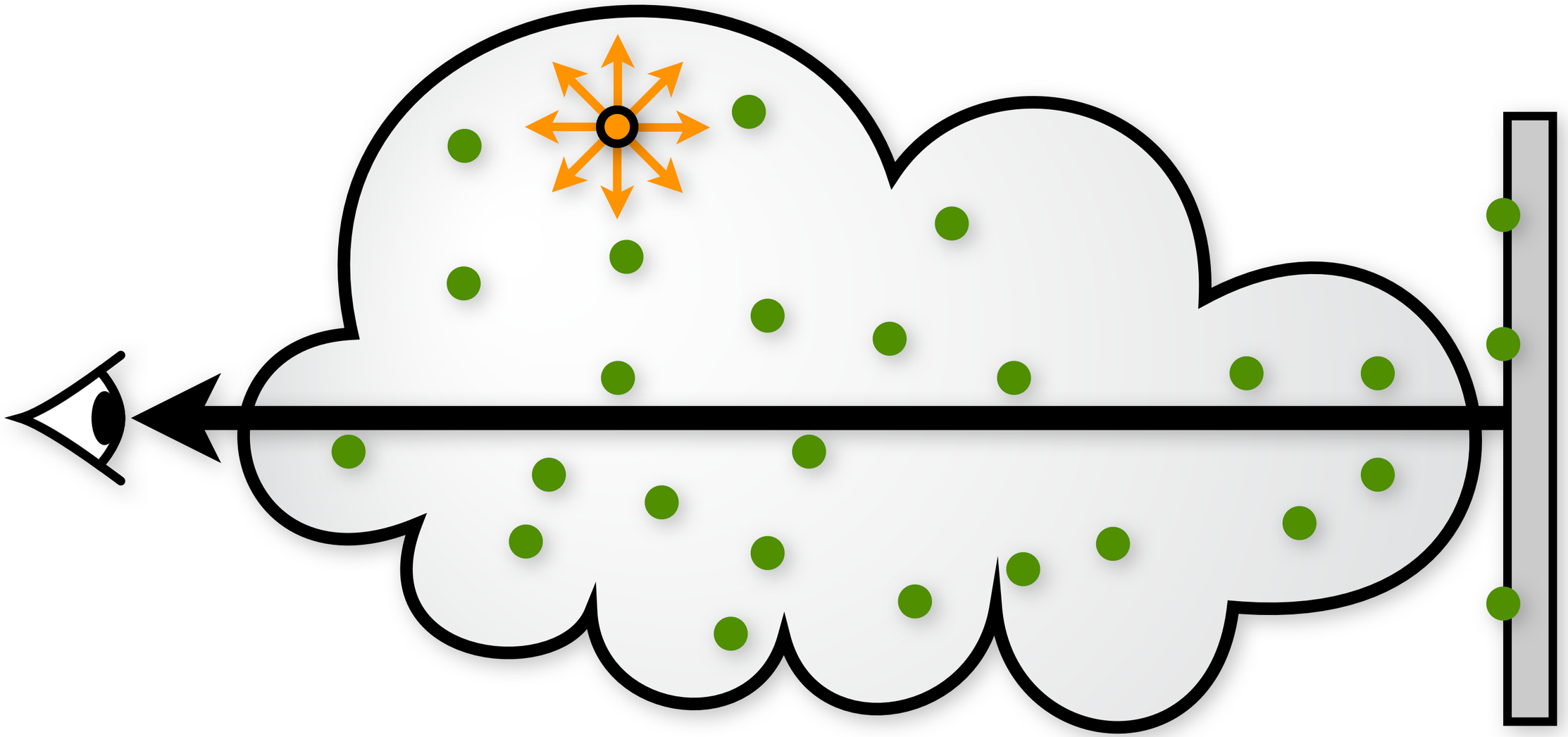
- For photon tracing, instead of emitted photons traveling directly to the surfaces, these photons now have some probability of scattering or being absorbed by the medium along their path.
- So we now trace paths that may have vertices within the medium, and at some point these paths will be probabilistically terminated when absorption happens.

Photon Tracing in Participating Media



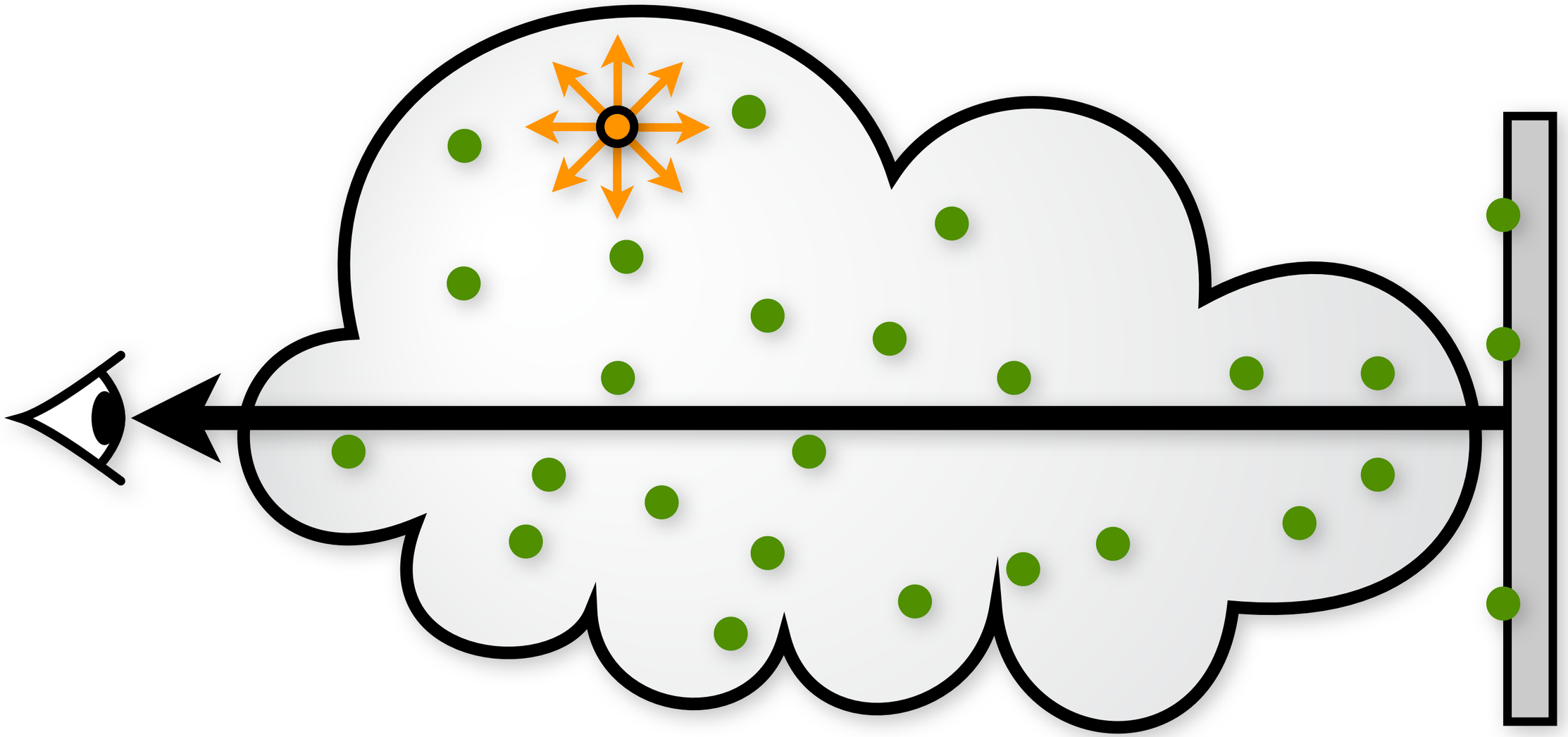
- For photon tracing, instead of emitted photons traveling directly to the surfaces, these photons now have some probability of scattering or being absorbed by the medium along their path.
- So we now trace paths that may have vertices within the medium, and at some point these paths will be probabilistically terminated when absorption happens.

Photon Tracing in Participating Media



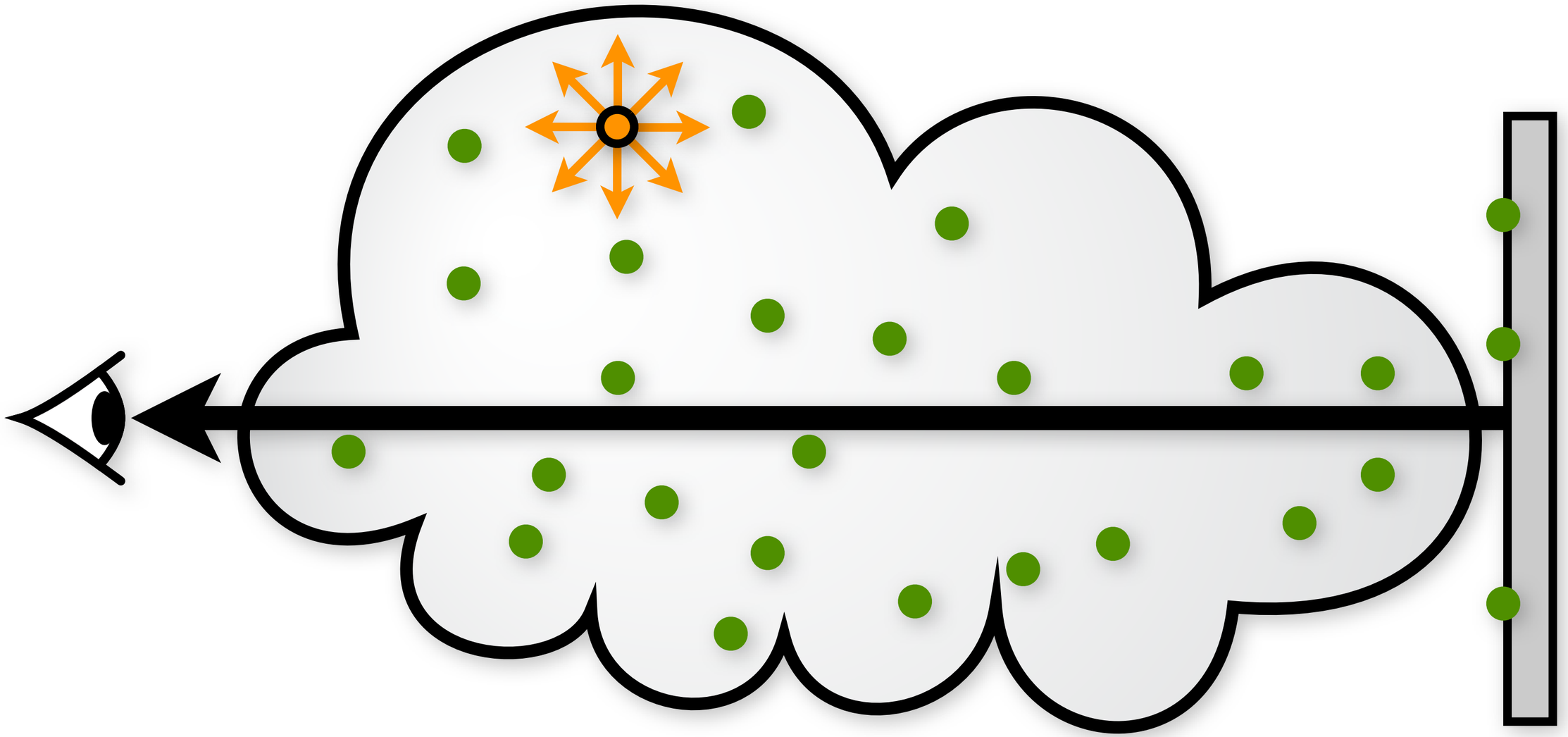
- At the end of this process we will have a collection of photons both on surfaces, and also some suspended in the medium

Photon Tracing in Participating Media



- At the end of this process we will have a collection of photons both on surfaces, and also some suspended in the medium

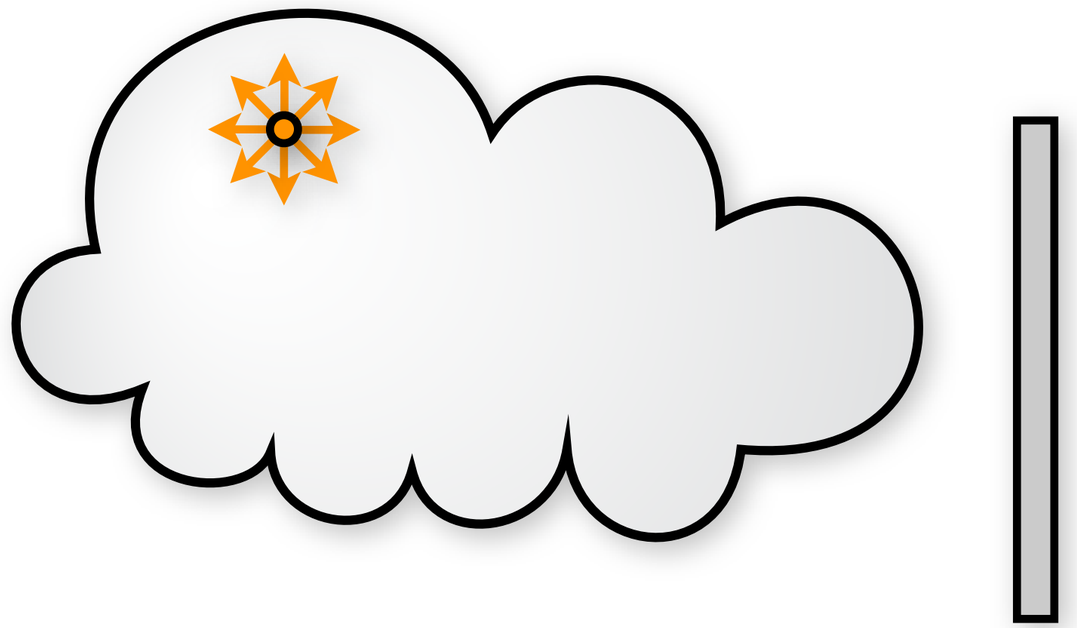
Photon Tracing in Participating Media



- At the end of this process we will have a collection of photons both on surfaces, and also some suspended in the medium

Basic Surface Photon Tracer

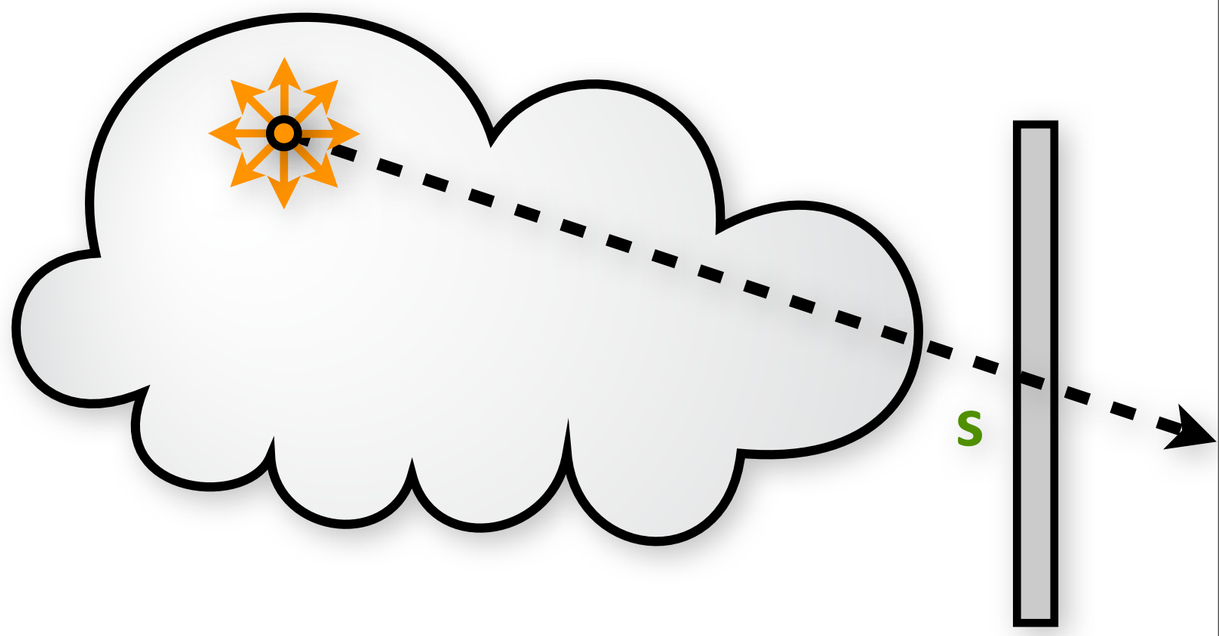
```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
```



- In the most basic form, a surface photon tracing algorithm might look something like this.
- We trace a ray to figure out the nearest surface hit in that direction, we then propagate the photon to the surface, store it, and recursively scattering based on the BRDF

Basic Surface Photon Tracer

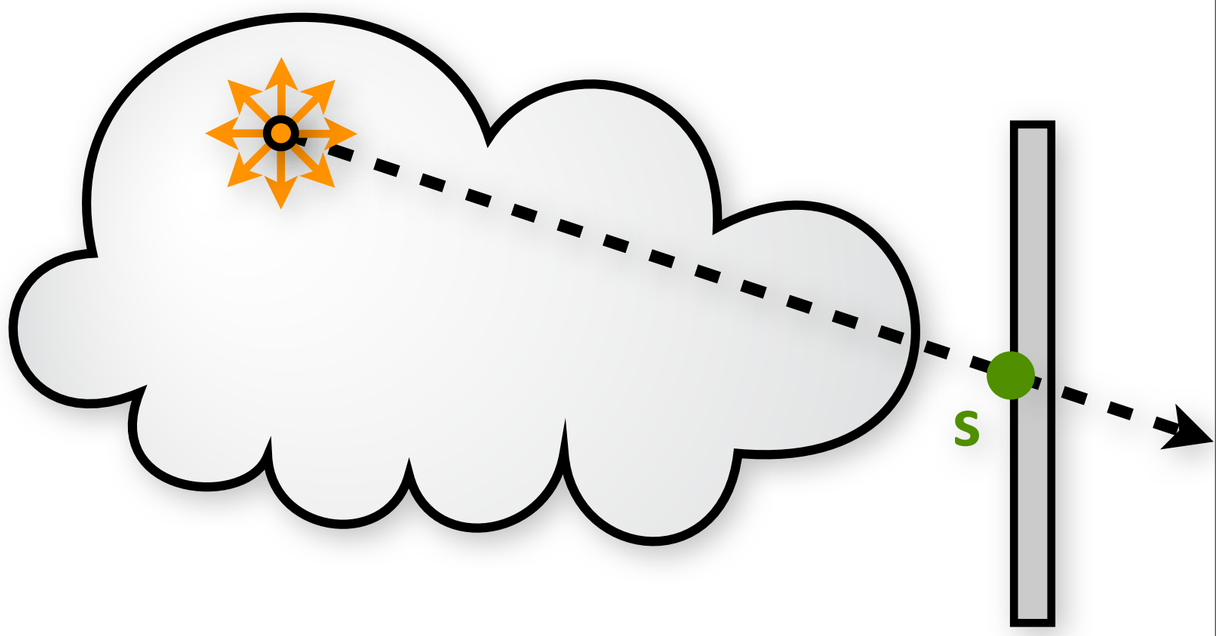
```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
```



- In the most basic form, a surface photon tracing algorithm might look something like this.
- We trace a ray to figure out the nearest surface hit in that direction, we then propagate the photon to the surface, store it, and recursively scattering based on the BRDF

Basic Surface Photon Tracer

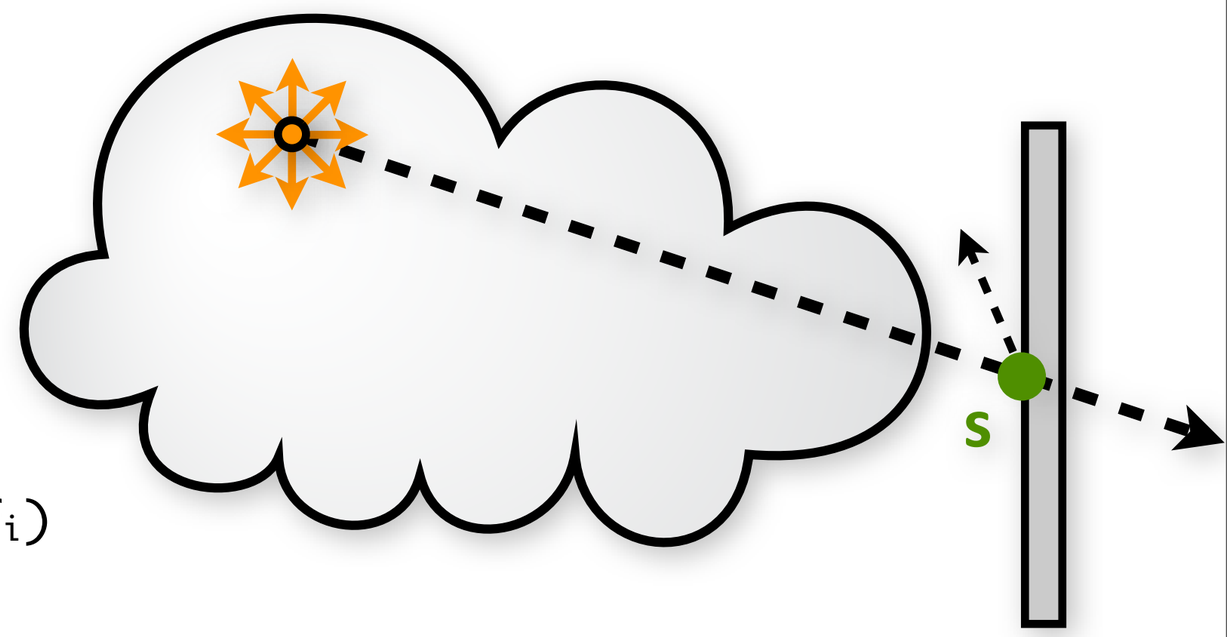
```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
    o += s*ω        // propagate photon
    storeSurfacePhoton(o, ω, ϕ)
```



- In the most basic form, a surface photon tracing algorithm might look something like this.
- We trace a ray to figure out the nearest surface hit in that direction, we then propagate the photon to the surface, store it, and recursively scattering based on the BRDF

Basic Surface Photon Tracer

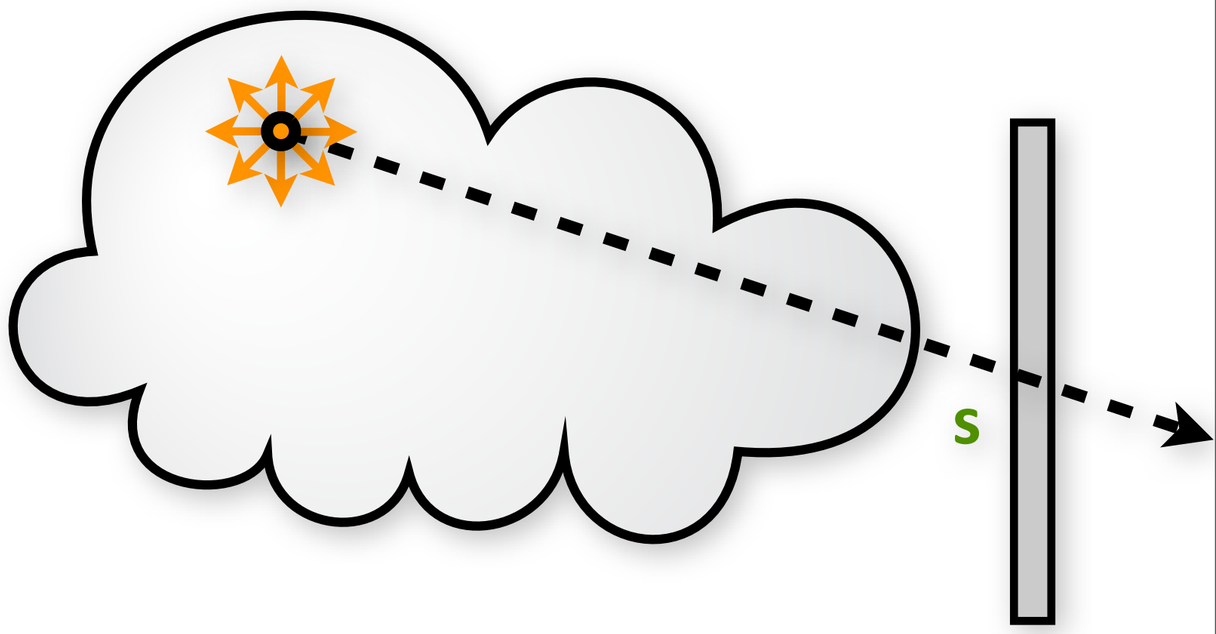
```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
    o += s*ω        // propagate photon
    storeSurfacePhoton(o, ω, ϕ)
    (ωi, pdfi) = sampleBRDF(o, ω)
    return vPT(o, ωi, ϕ * BRDF(o,ω,ωi) / pdfi)
```



- In the most basic form, a surface photon tracing algorithm might look something like this.
- We trace a ray to figure out the nearest surface hit in that direction, we then propagate the photon to the surface, store it, and recursively scattering based on the BRDF

Basic Volumetric Photon Tracer

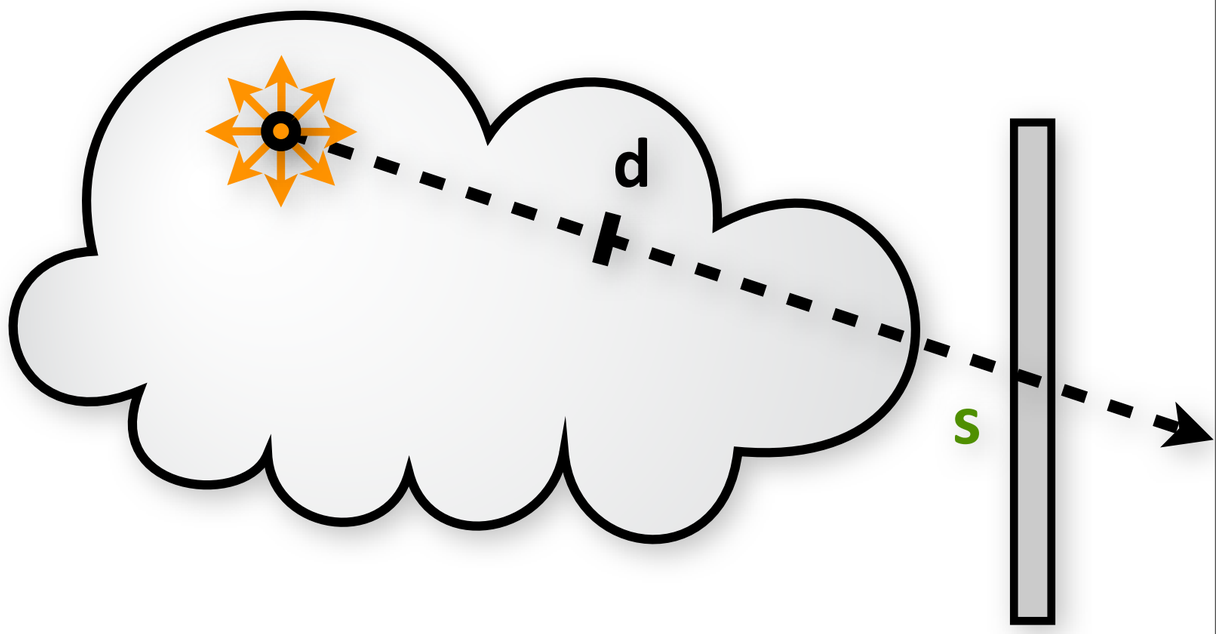
```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
    o += s*ω        // propagate photon
    storeSurfacePhoton(o, ω, ϕ)
    (ωi, pdfi) = sampleBRDF(o, ω)
    return vPT(o, ωi, ϕ * BRDF(o,ω,ωi) / pdfi)
```



- We can pretty easily generalize this to a volumetric photon tracer
- We will still need to perform the surface computation, but first, we will compute whats called the free flight distance.
- This is the random distance that a photon will travel before probabilistically interacting with the medium
- Now, if this distance d is less than the distance s to the nearest surface, then we have a medium event, and we propagate the photon, and then simulate a recursive medium scattering event
- Only if the distance d is greater than s will the photon actually reach the surface, which is where we use our surface photon scattering code

Basic Volumetric Photon Tracer

```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
    d = freeFlightDistance(o, ω)
    o += s*ω          // propagate photon
    storeSurfacePhoton(o, ω, ϕ)
    (ωi, pdfi) = sampleBRDF(o, ω)
    return vPT(o, ωi, ϕ * BRDF(o,ω,ωi) / pdfi)
```



Thursday, August 23, 12

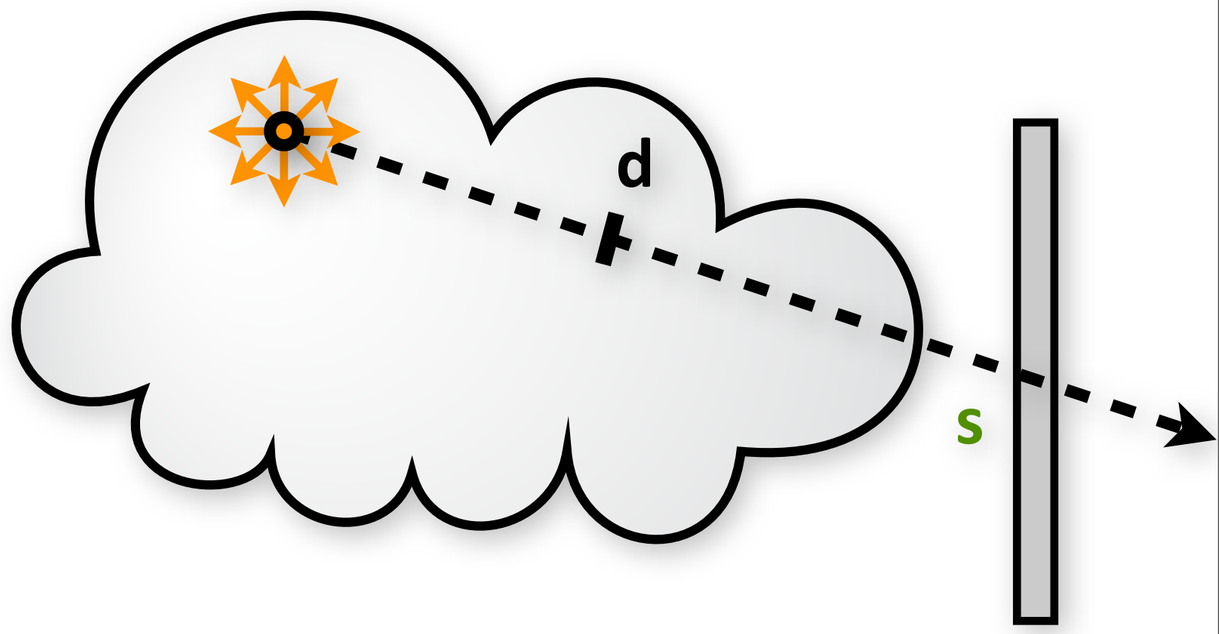
- We can pretty easily generalize this to a volumetric photon tracer
- We will still need to perform the surface computation, but first, we will compute whats called the free flight distance.
- This is the random distance that a photon will travel before probabilistically interacting with the medium
- Now, if this distance d is less than the distance s to the nearest surface, then we have a medium event, and we propagate the photon, and then simulate a recursive medium scattering event
- Only if the distance d is greater than s will the photon actually reach the surface, which is where we use our surface photon scattering code

Basic Volumetric Photon Tracer

```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
    d = freeFlightDistance(o, ω)
    if (d < s)          // media scattering




    else                // surface scattering
        o += s*ω        // propagate photon
        storeSurfacePhoton(o, ω, ϕ)
        (ωi, pdfi) = sampleBRDF(o, ω)
        return vPT(o, ωi, ϕ * BRDF(o,ω,ωi) / pdfi)
```

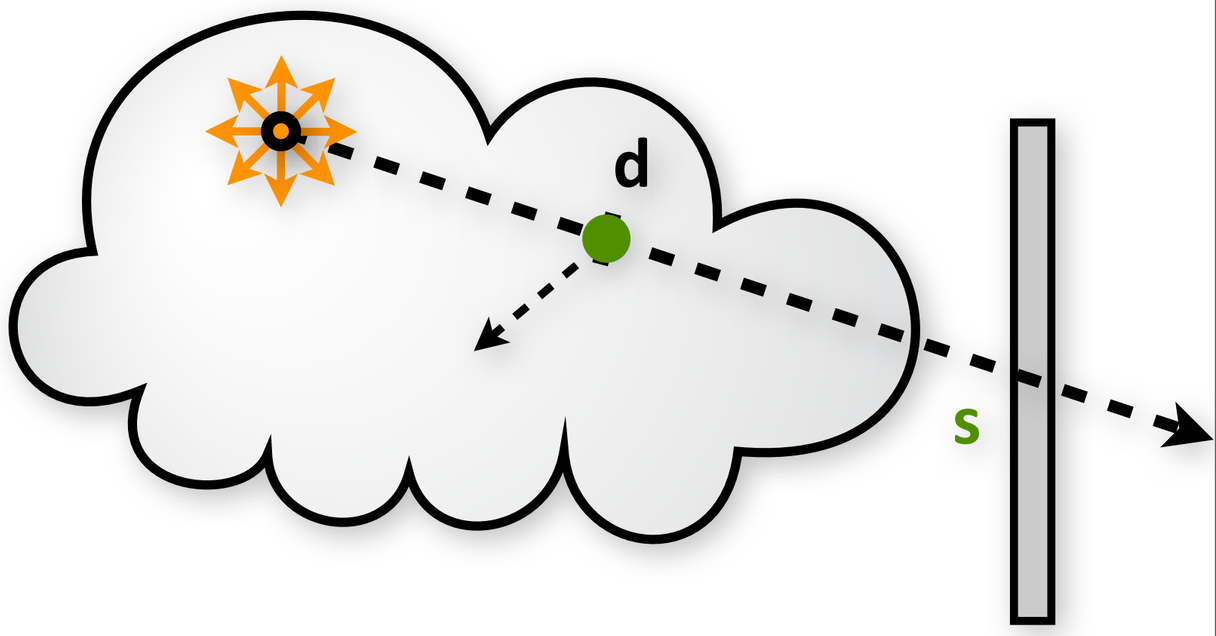


Thursday, August 23, 12

- We can pretty easily generalize this to a volumetric photon tracer
- We will still need to perform the surface computation, but first, we will compute whats called the free flight distance.
- This is the random distance that a photon will travel before probabilistically interacting with the medium
- Now, if this distance d is less than the distance s to the nearest surface, then we have a medium event, and we propagate the photon, and then simulate a recursive medium scattering event
- Only if the distance d is greater than s will the photon actually reach the surface, which is where we use our surface photon scattering code

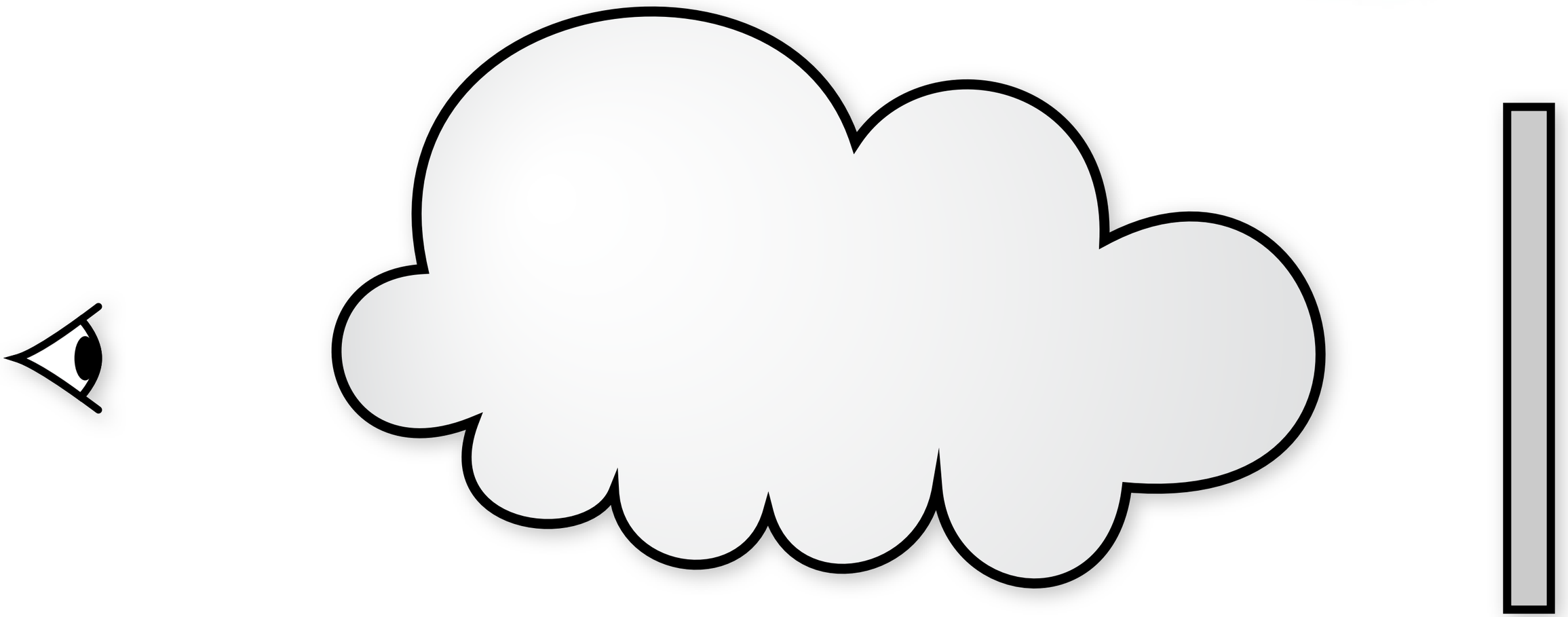
Basic Volumetric Photon Tracer

```
void vPT(o, ω, ϕ)
    s = nearestSurfaceHit(o, ω)
    d = freeFlightDistance(o, ω)
    if (d < s)              // media scattering
        o += d*ω            // propagate photon
        storeVolumePhoton(o, ω, ϕ)
        return vPT(o, samplePF(), ϕ * σs / σt)
    else                    // surface scattering
        o += s*ω            // propagate photon
        storeSurfacePhoton(o, ω, ϕ)
        (ωi, pdfi) = sampleBRDF(o, ω)
        return vPT(o, ωi, ϕ * BRDF(o,ω,ωi) / pdfi)
```



- We can pretty easily generalize this to a volumetric photon tracer
- We will still need to perform the surface computation, but first, we will compute whats called the free flight distance.
- This is the random distance that a photon will travel before probabilistically interacting with the medium
- Now, if this distance d is less than the distance s to the nearest surface, then we have a medium event, and we propagate the photon, and then simulate a recursive medium scattering event
- Only if the distance d is greater than s will the photon actually reach the surface, which is where we use our surface photon scattering code

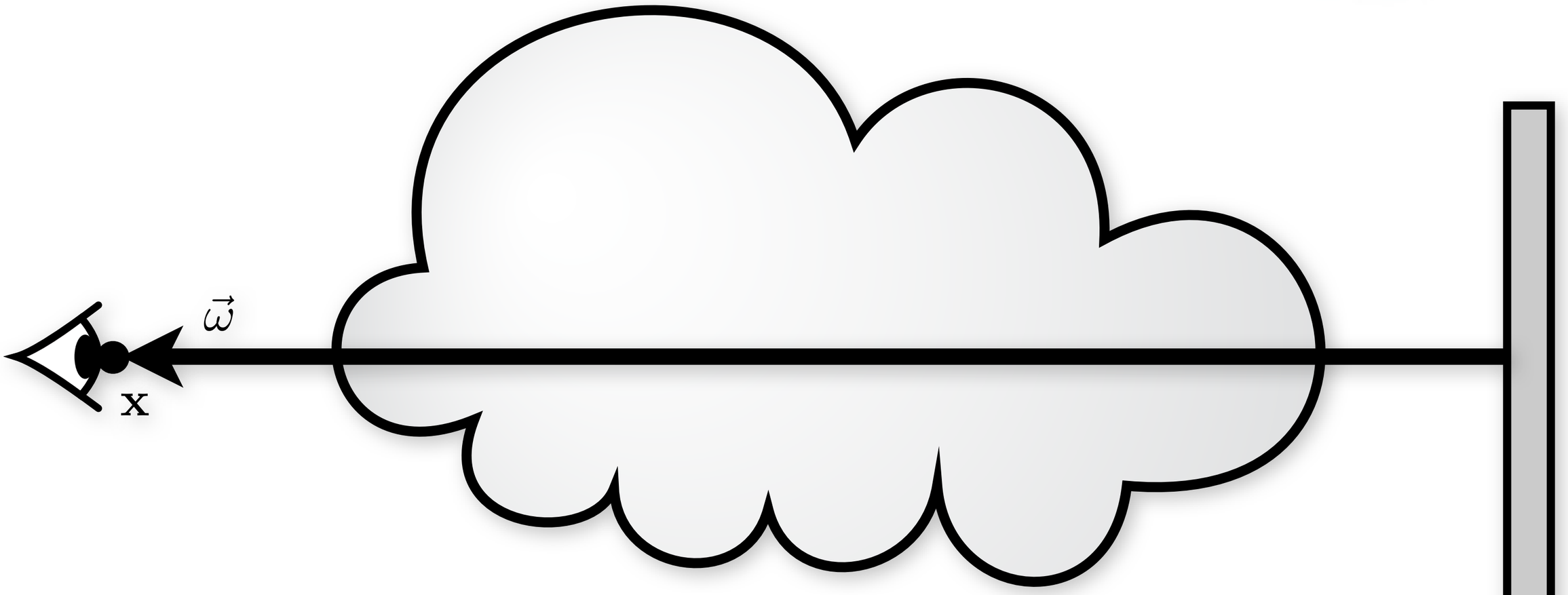
Volume Rendering Equation



Thursday, August 23, 12

- For rendering the final image, we also need to consider not just the rendering equation, but the volume rendering equation
- Which is shown below
- And this is really just the sum to two main terms

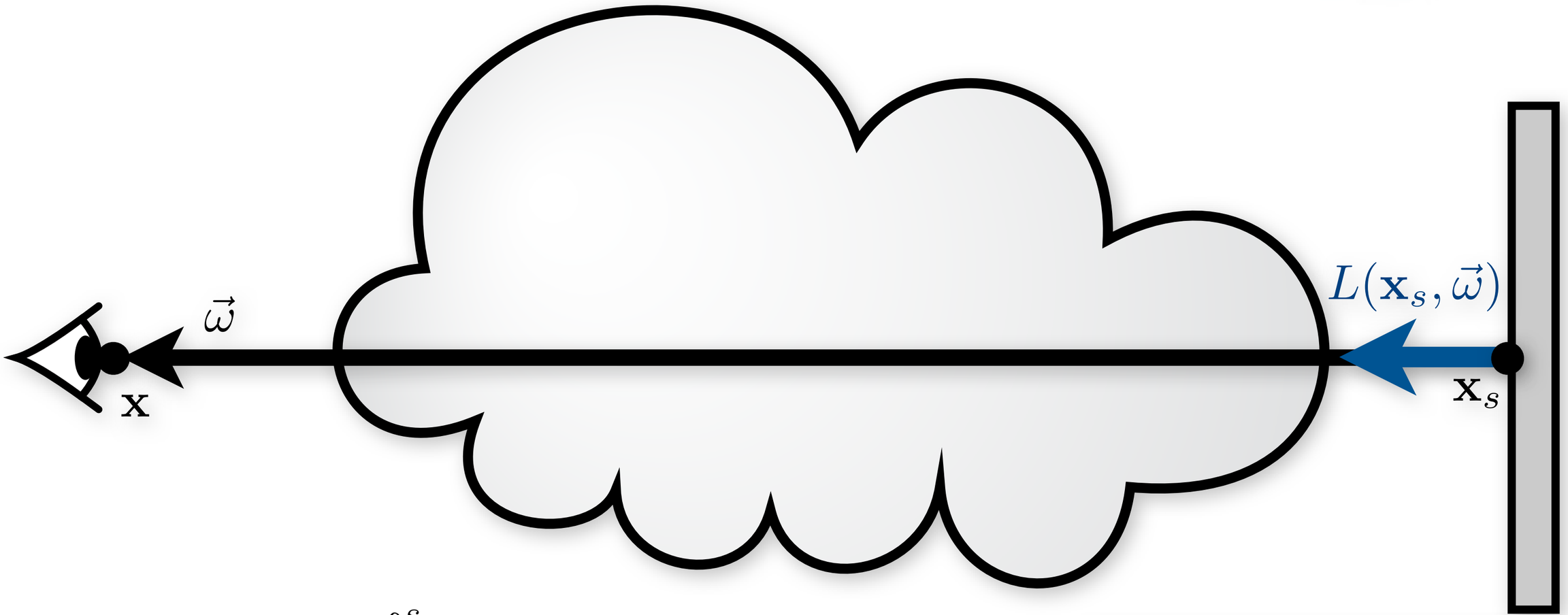
Volume Rendering Equation



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

- For rendering the final image, we also need to consider not just the rendering equation, but the volume rendering equation
- Which is shown below
- And this is really just the sum to two main terms

Volume Rendering Equation

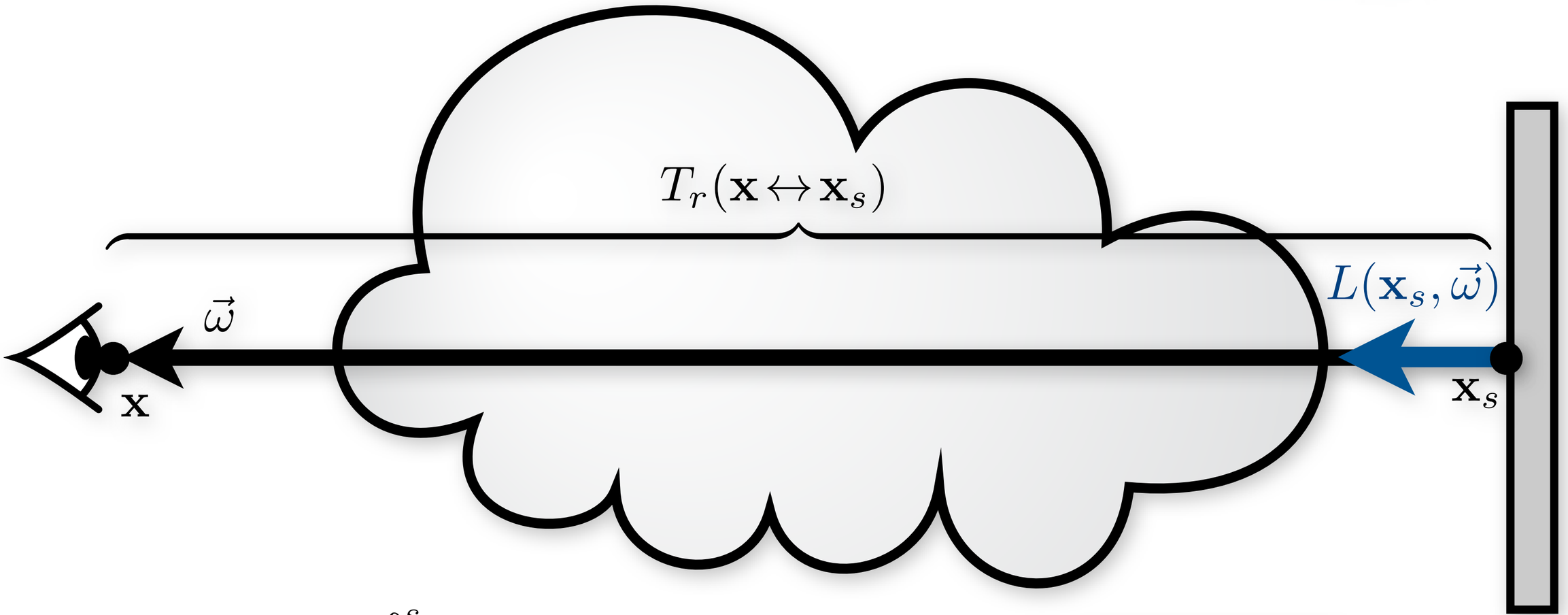


$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

Reduced surface radiance

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

Volume Rendering Equation

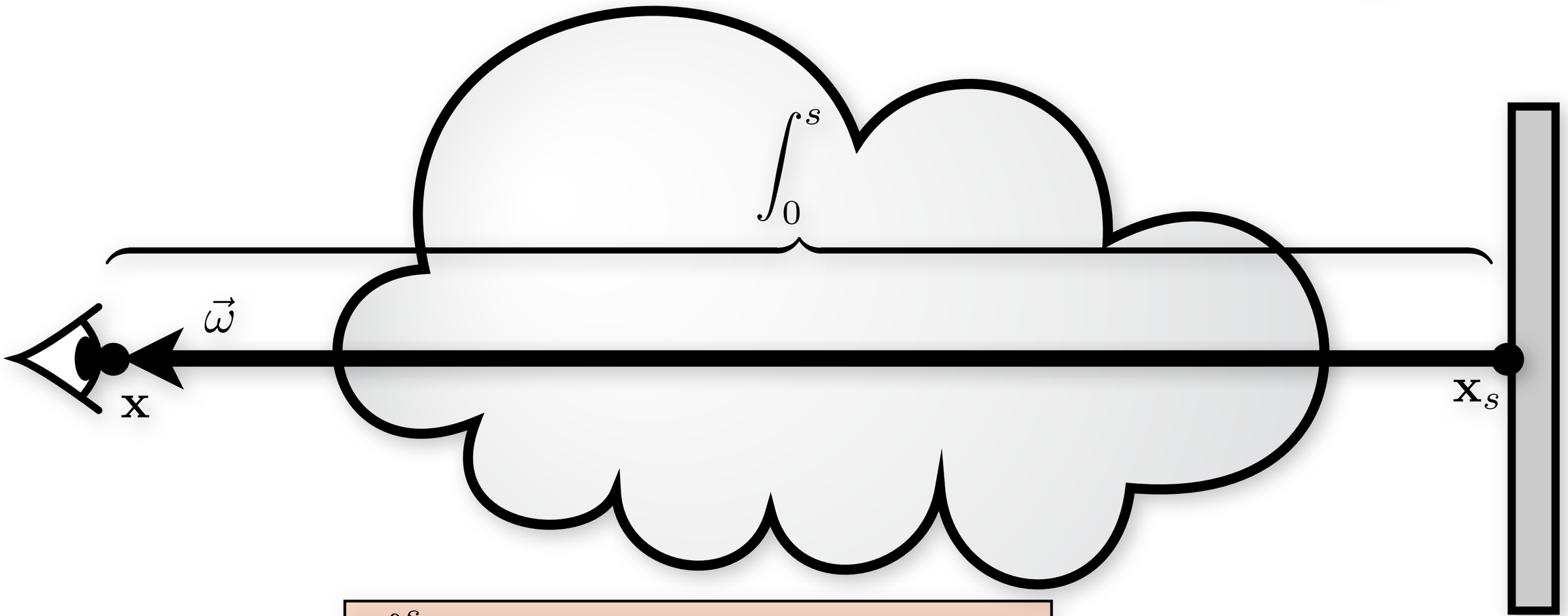


$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\ dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

Transmittance: fractional visibility between two points

- The right-hand term incorporates lighting arriving from a surface
- However, before reaching the eye, this radiance must travel through the medium and so is attenuated by a transmission term, which is really just a fractional visibility between the two points

Volume Rendering Equation

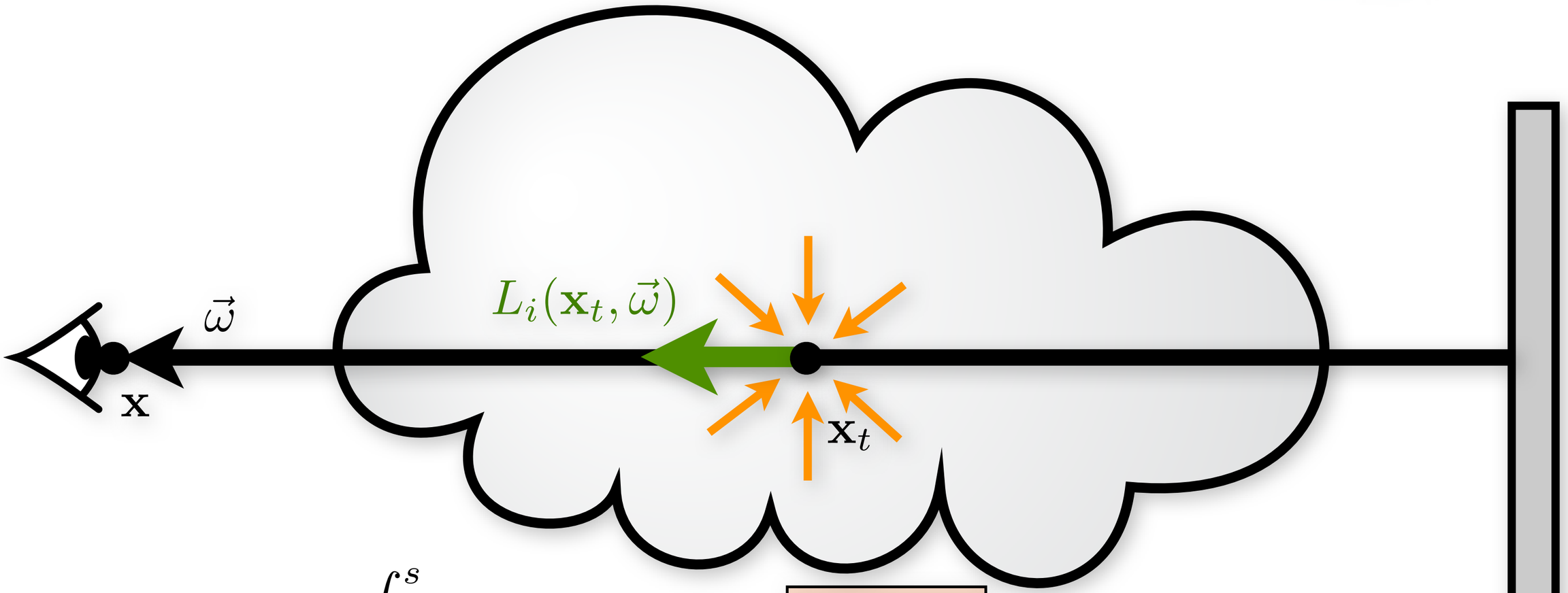


$$L(\mathbf{x},\vec{\omega}) = \int_0^s T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\ dt + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

Accumulated in-scattered radiance

- However, light may also actually come from the medium
- The left-hand term integrates the scattering of light from the medium along the whole length of the ray

Volume Rendering Equation

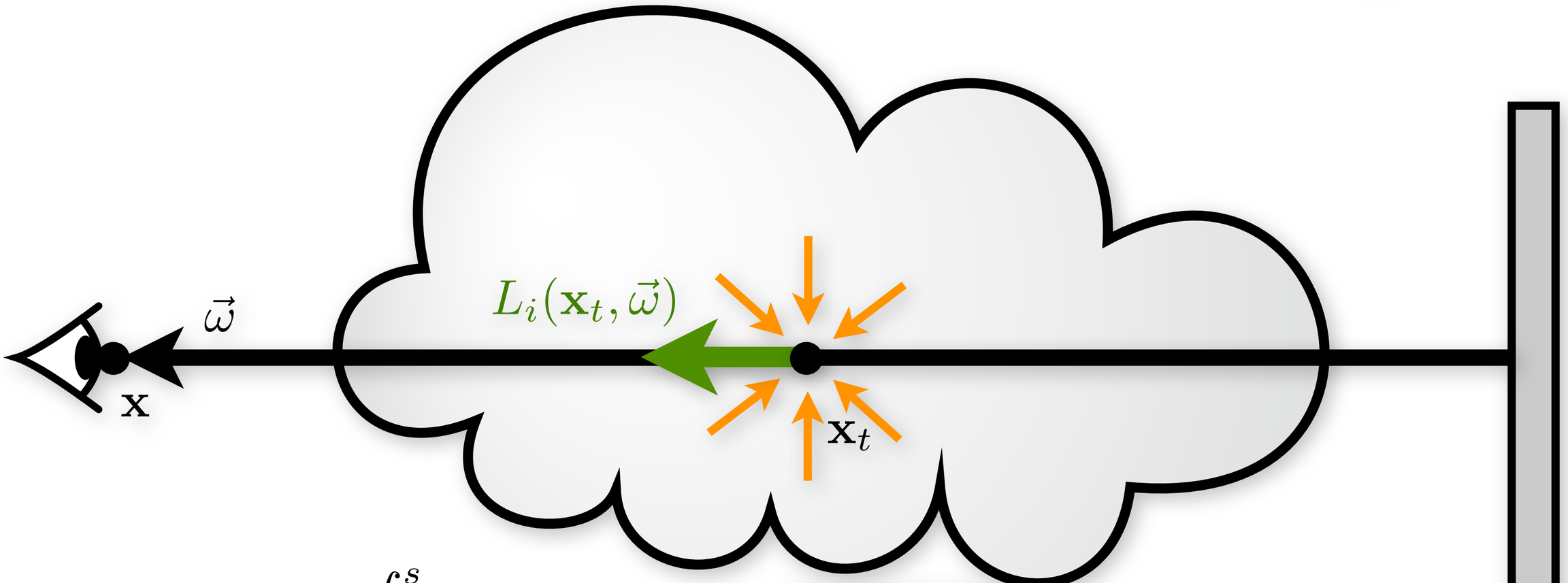


$$L(\mathbf{x}, \vec{\omega}) = \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t, \vec{\omega})\,dt + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)L(\mathbf{x}_s, \vec{\omega})$$

Thursday, August 23, 12

- And 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 scatters towards the eye.
- This is similar to the reflection equation on surfaces, but using the phase function
- Computing this inscattered radiance is really what makes volume rendering expensive

Volume Rendering Equation



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

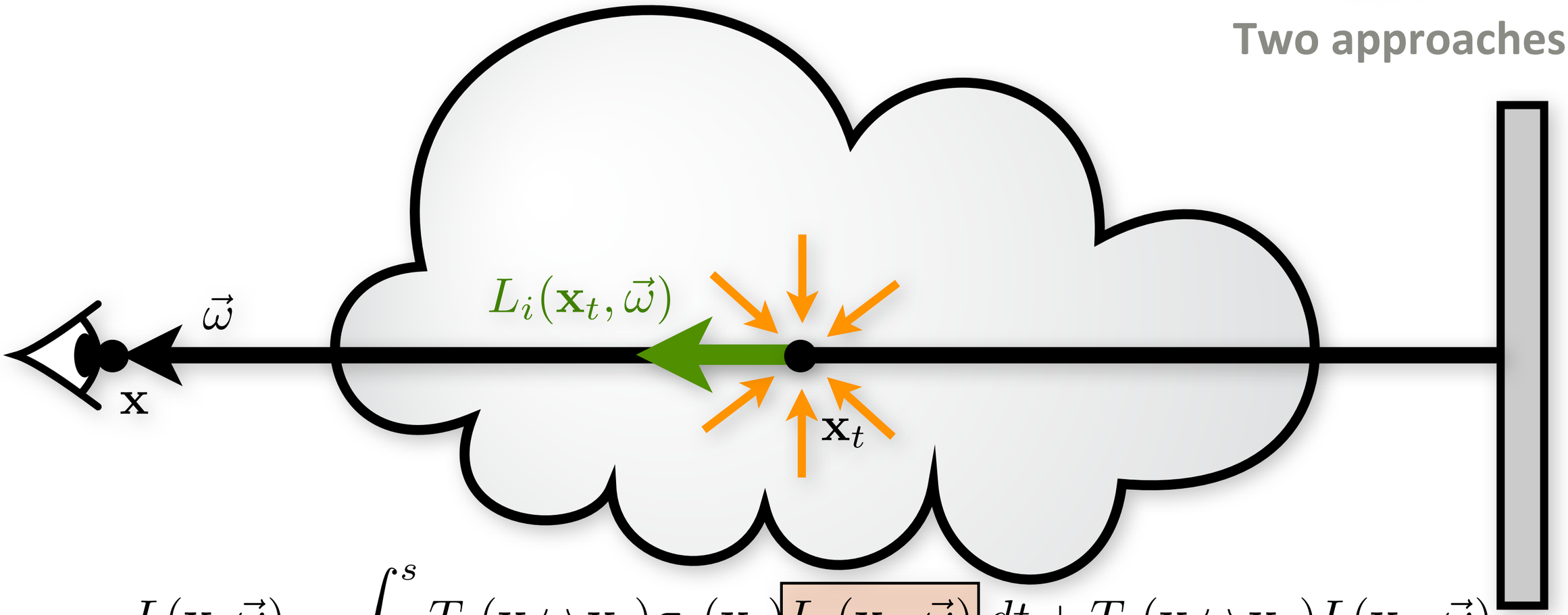
$$\boxed{L_i(\mathbf{x}_t, \vec{\omega})} = \int_{\Omega_{4\pi}} p(\mathbf{x}_t, \vec{\omega}_t, \vec{\omega})L(\mathbf{x}_t, \vec{\omega}_t)\; d\omega_t$$

Thursday, August 23, 12

- And 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 scatters towards the eye.
- This is similar to the reflection equation on surfaces, but using the phase function
- Computing this inscattered radiance is really what makes volume rendering expensive

Volume Rendering Equation

Two approaches



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

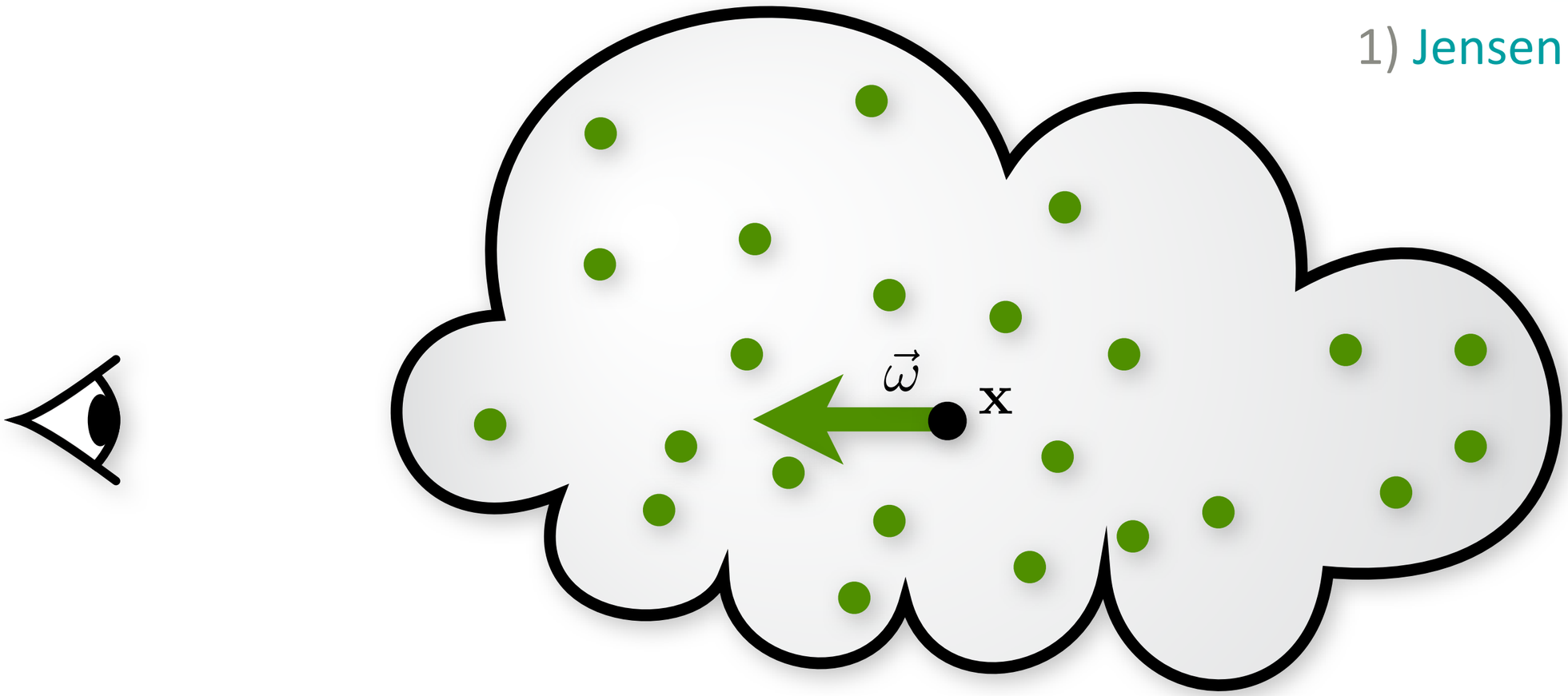
$$\boxed{L_i(\mathbf{x}_t, \vec{\omega})} = \int_{\Omega_{4\pi}} p(\mathbf{x}_t, \vec{\omega}_t, \vec{\omega})L(\mathbf{x}_t, \vec{\omega}_t)\; d\omega_t$$

Thursday, August 23, 12

- And 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 scatters towards the eye.
- This is similar to the reflection equation on surfaces, but using the phase function
- Computing this inscattered radiance is really what makes volume rendering expensive

Volumetric Radiance Estimate

1) Jensen & Christensen 98

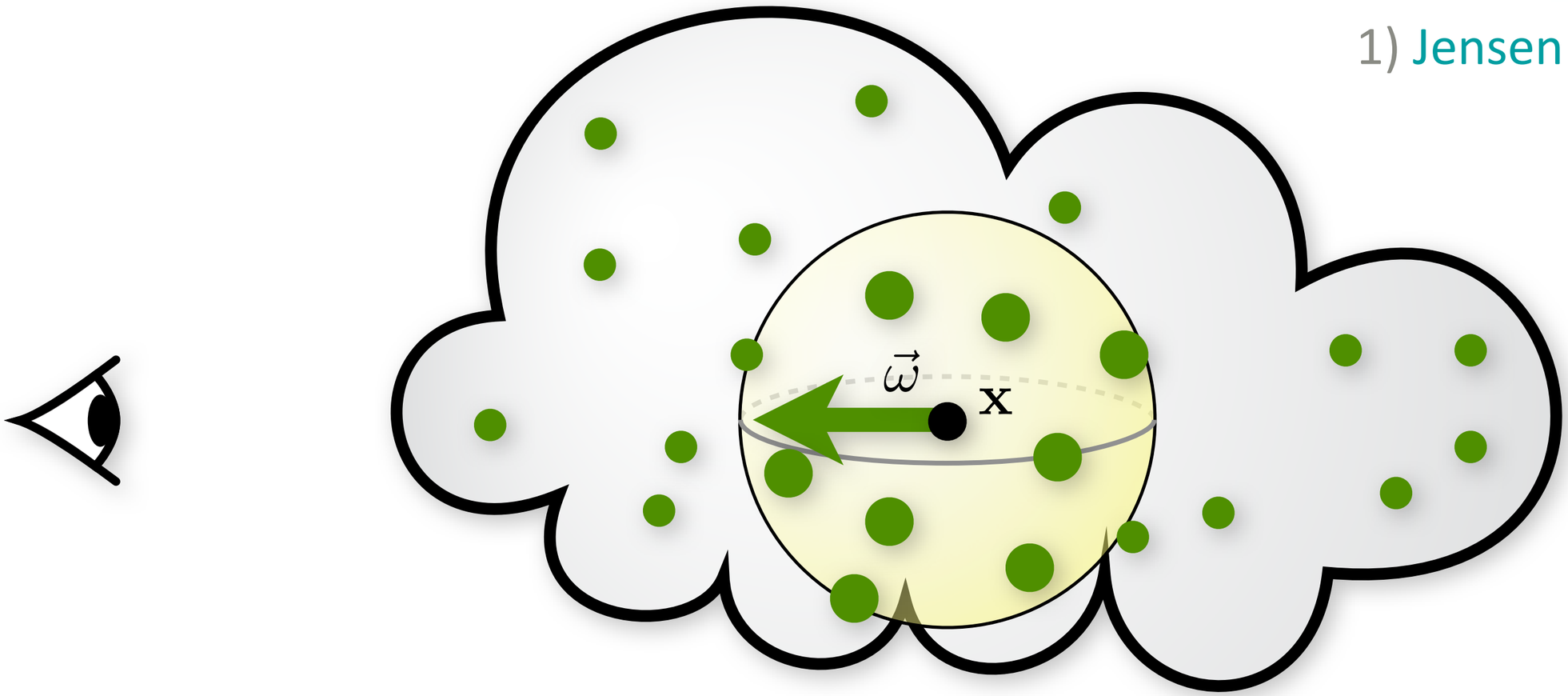


$$L_i(\mathbf{x}_t, \vec{\omega}) = \int_{\Omega_{4\pi}} p(\mathbf{x}_t, \vec{\omega}_t, \vec{\omega}) L(\mathbf{x}_t, \vec{\omega}_t)\ d\omega_t$$

- But we have the photon map, and we can simply find photons within the local region, just like we did at surfaces
- However, instead of computing a density on a surface, we compute a volume density, so we divide by the volume of the sphere

Volumetric Radiance Estimate

1) Jensen & Christensen 98

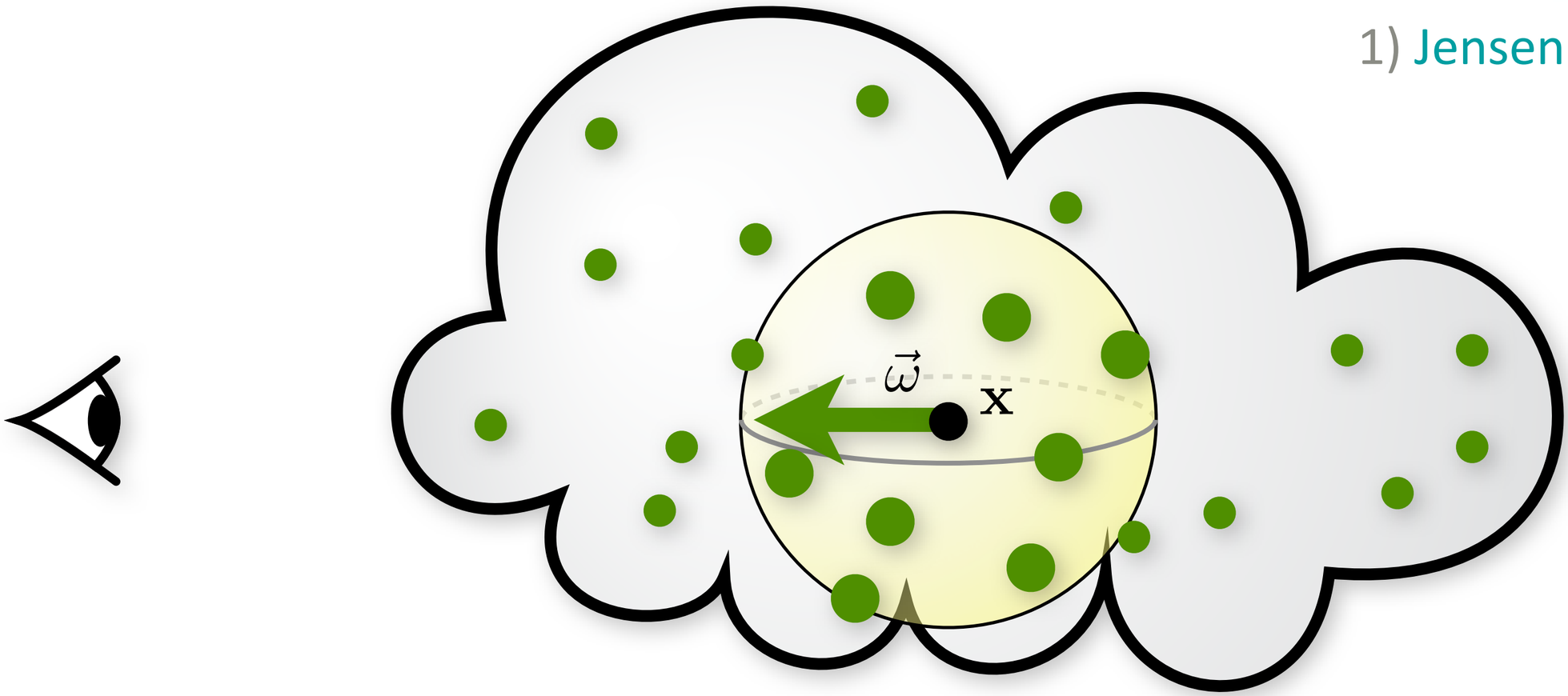


$$L_i(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{k} p(\mathbf{x}, \vec{\omega}_i, \vec{\omega}) \frac{\Phi_i(\mathbf{x}, \vec{\omega}_i)}{\mathcal{V}(\mathbf{x})}$$

- But we have the photon map, and we can simply find photons within the local region, just like we did at surfaces
- However, instead of computing a density on a surface, we compute a volume density, so we divide by the volume of the sphere

Volumetric Radiance Estimate

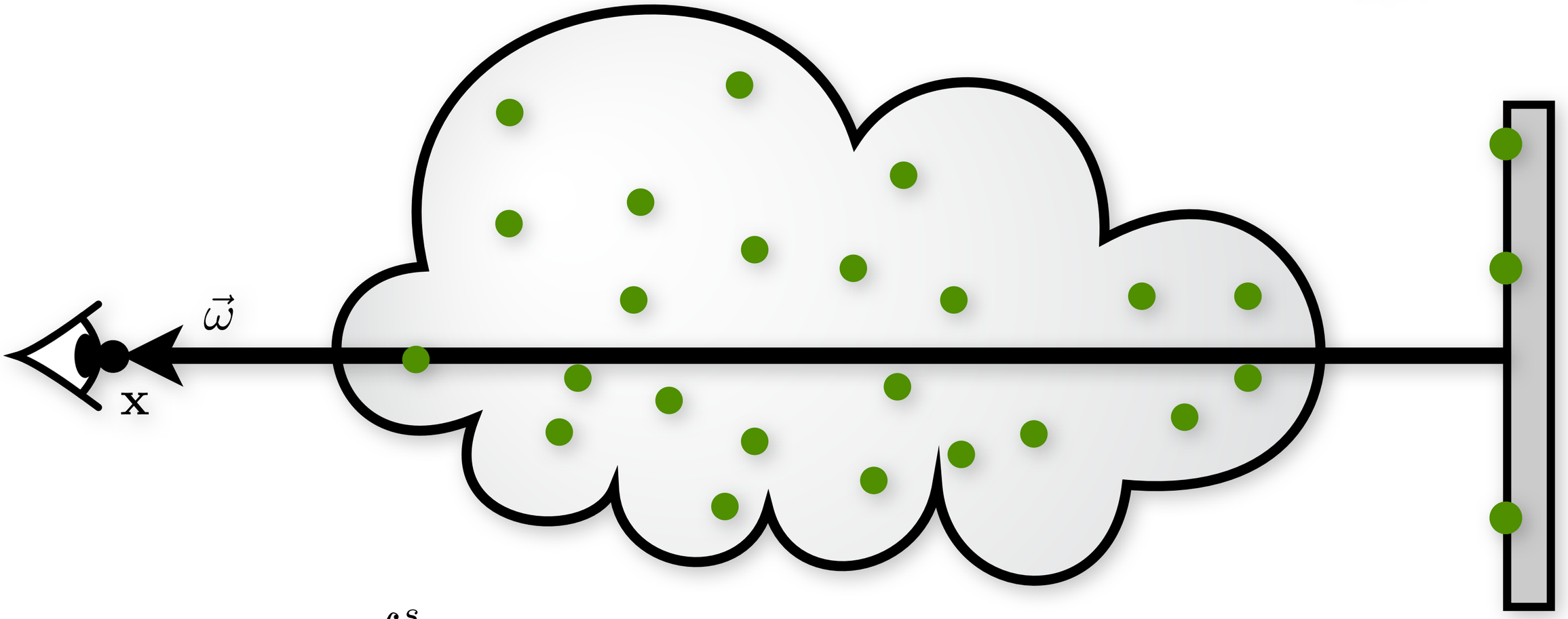
1) Jensen & Christensen 98



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

- But we have the photon map, and we can simply find photons within the local region, just like we did at surfaces
- However, instead of computing a density on a surface, we compute a volume density, so we divide by the volume of the sphere

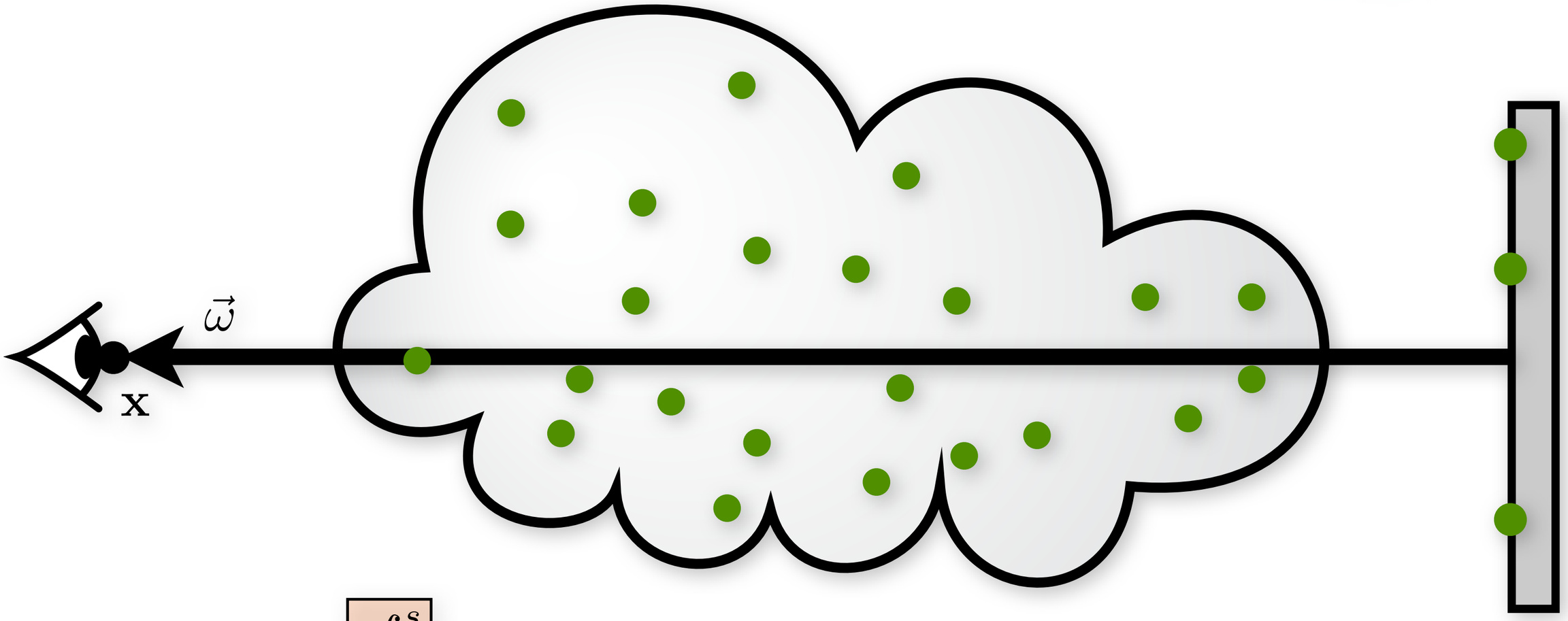
Volume Rendering Equation



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

- However, we still need to compute the outer integral
- And we can do this by using a simple Riemann sum, which results in ray marching

Volume Rendering Equation



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

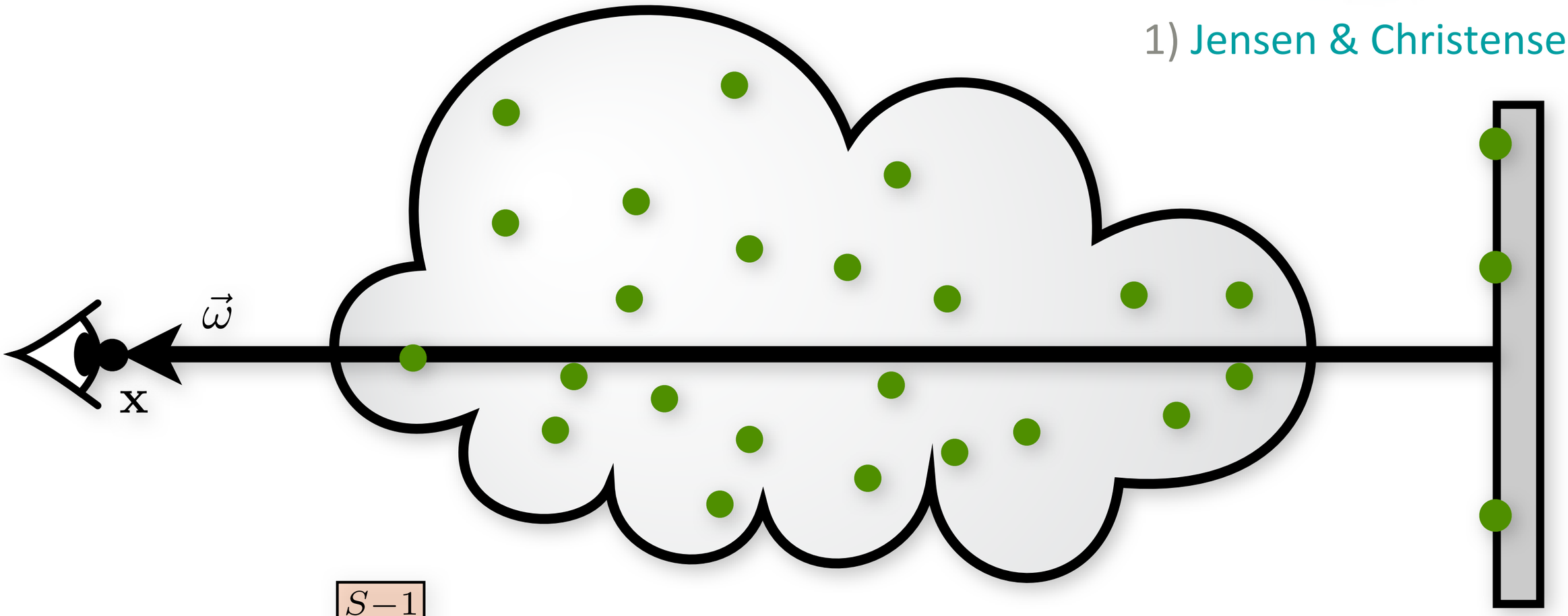
Approximate/compute using Riemann sum

Thursday, August 23, 12

- However, we still need to compute the outer integral
- And we can do this by using a simple Riemann sum, which results in ray marching

Ray Marching

1) Jensen & Christensen 98



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

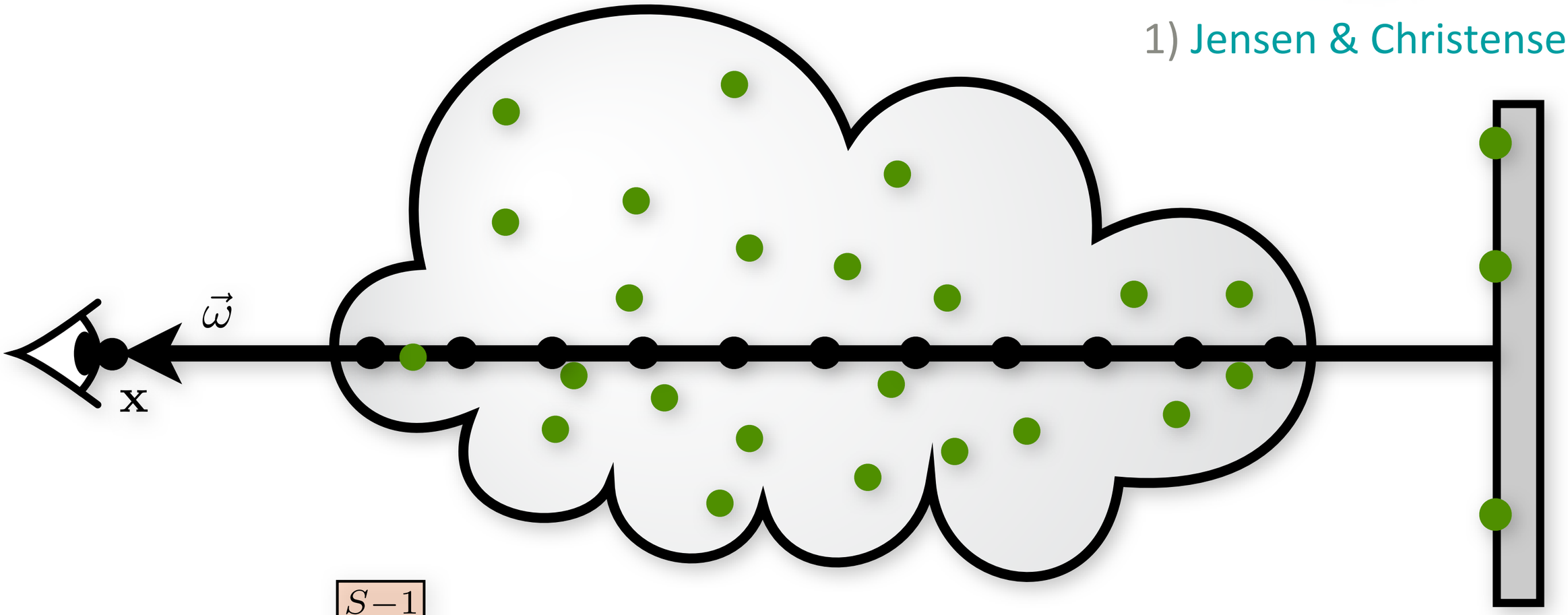
Approximate/compute using Riemann sum

Thursday, August 23, 12

- However, we still need to compute the outer integral
- And we can do this by using a simple Riemann sum, which results in ray marching
- Where we effectively march along the ray, and at each point we will estimate the integrand, and just add all these evaluations together.

Ray Marching

1) Jensen & Christensen 98



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

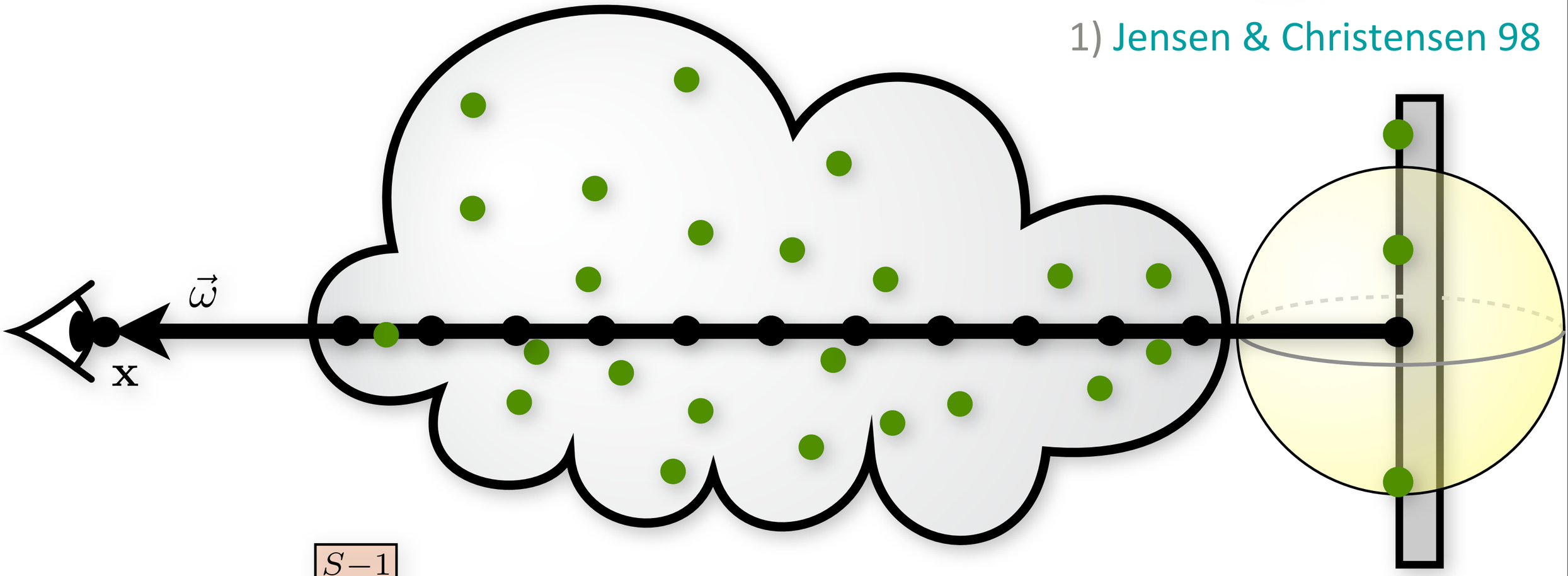
Approximate/compute using Riemann sum

Thursday, August 23, 12

- However, we still need to compute the outer integral
- And we can do this by using a simple Riemann sum, which results in ray marching
- Where we effectively march along the ray, and at each point we will estimate the integrand, and just add all these evaluations together.

Ray Marching

1) Jensen & Christensen 98



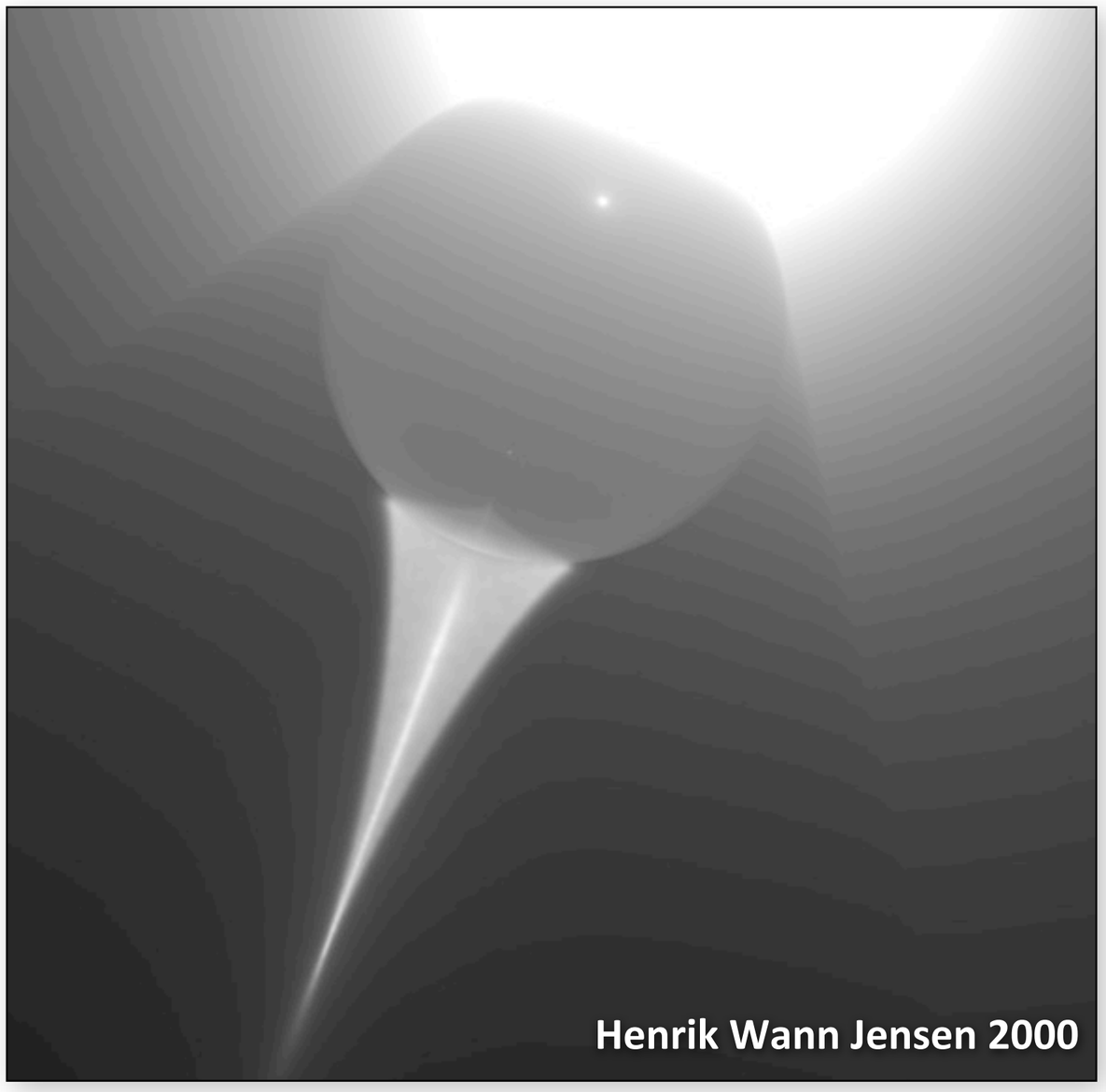
$$L(\mathbf{x},\vec{\omega}) \approx \boxed{\sum_{t=0}^{S-1}} T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\sigma_s(\mathbf{x}_t)L_i(\mathbf{x}_t,\vec{\omega})\boxed{\Delta_t} + T_r(\mathbf{x}\leftrightarrow\mathbf{x}_s)L(\mathbf{x}_s,\vec{\omega})$$

Approximate/compute using Riemann sum

Thursday, August 23, 12

- However, we still need to compute the outer integral
- And we can do this by using a simple Riemann sum, which results in ray marching
- Where we effectively march along the ray, and at each point we will estimate the integrand, and just add all these evaluations together.

Volume Caustics



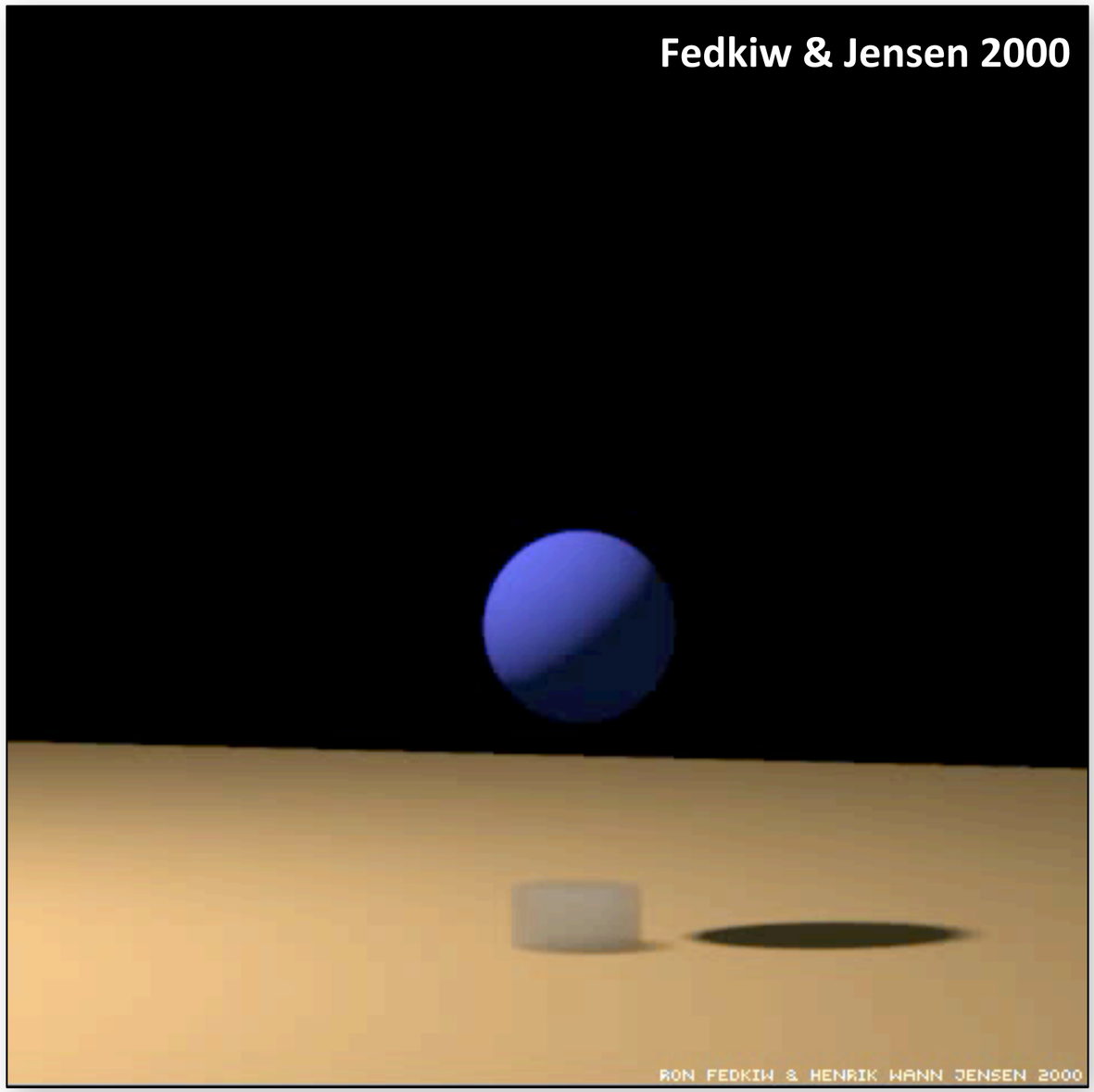
Thursday, August 23, 12

- And by doing this, we can get really complex lighting effects like this volume caustic

Subsurface Scattering

- Or, if we imagine the surface of this statue to just be the boundary of some participating medium, we get soft natural subsurface scattering within this marble bust

Rising Smoke

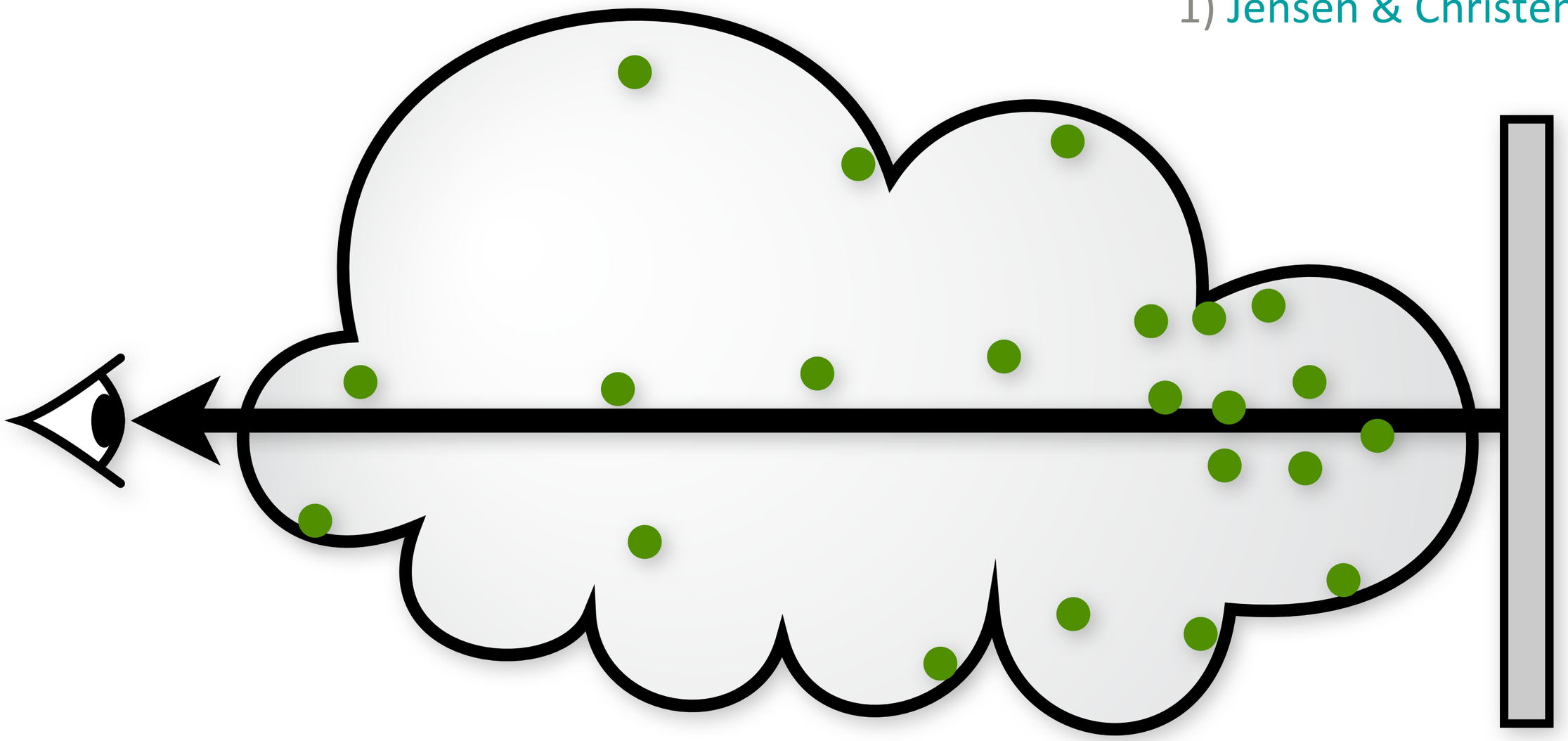


Thursday, August 23, 12

- We can also simulate smoke, and we can see that

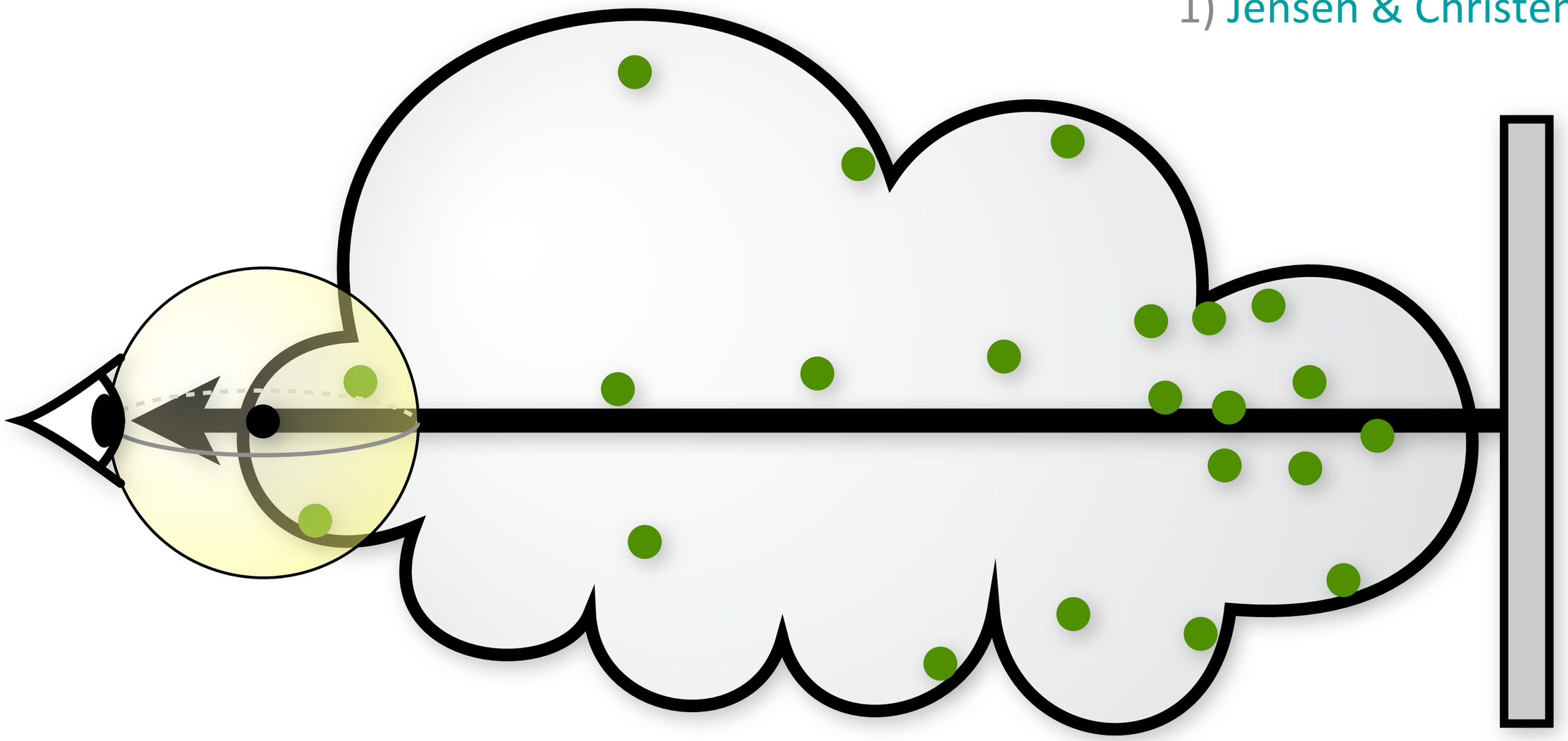
Volumetric Photon Mapping

1) Jensen & Christensen 98



Volumetric Photon Mapping

1) Jensen & Christensen 98



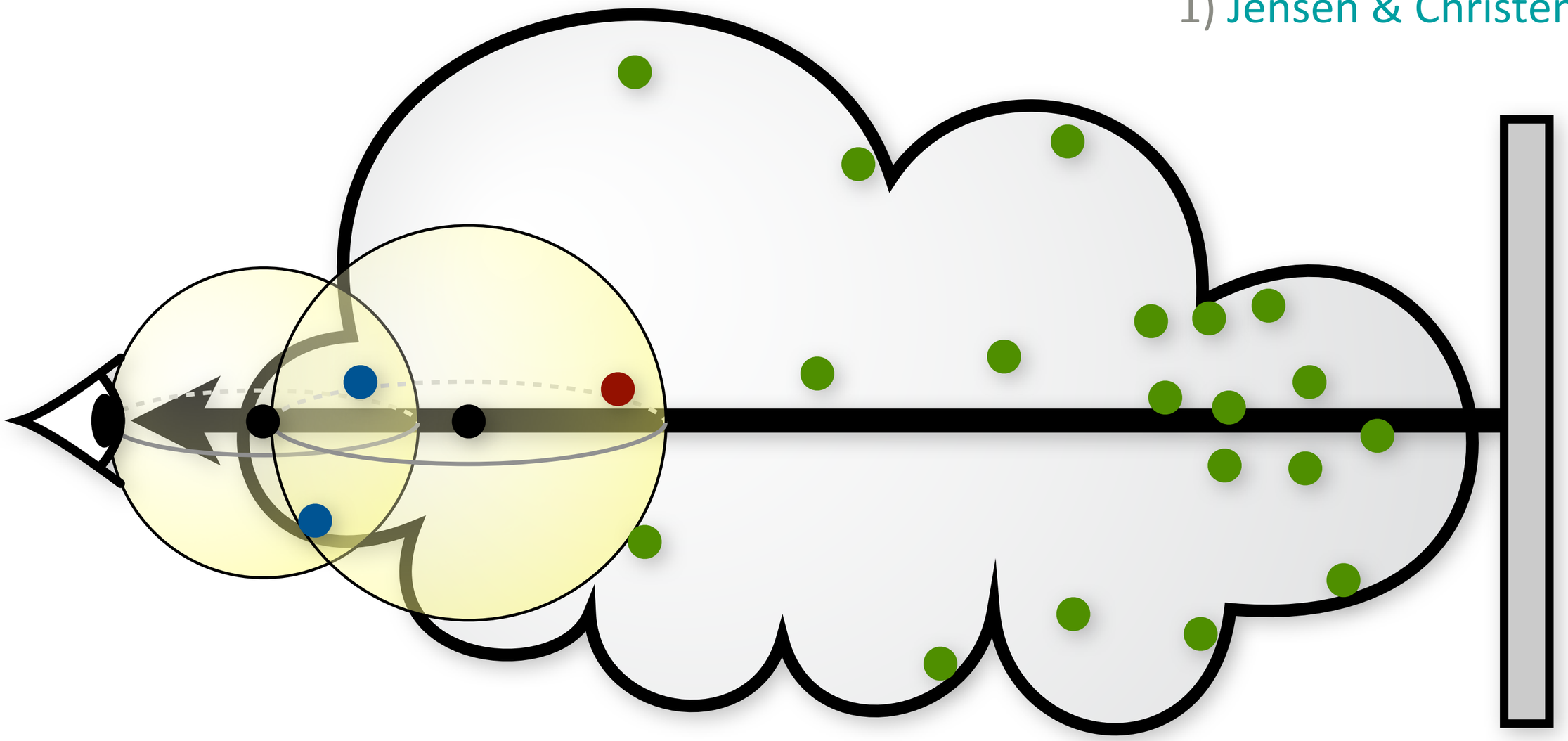
Volumetric Photon Mapping

1) Jensen & Christensen 98

found

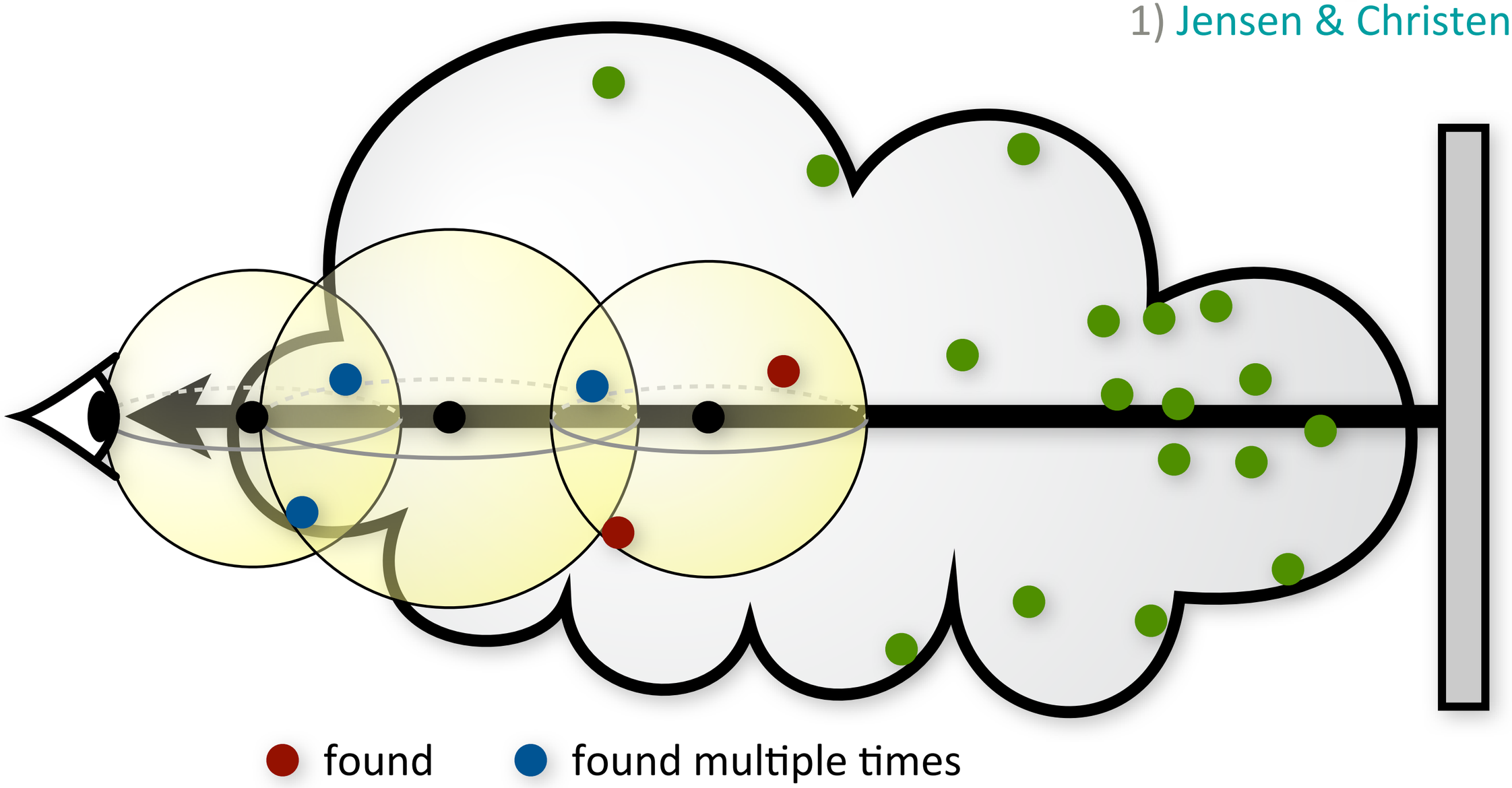
Volumetric Photon Mapping

1) Jensen & Christensen 98



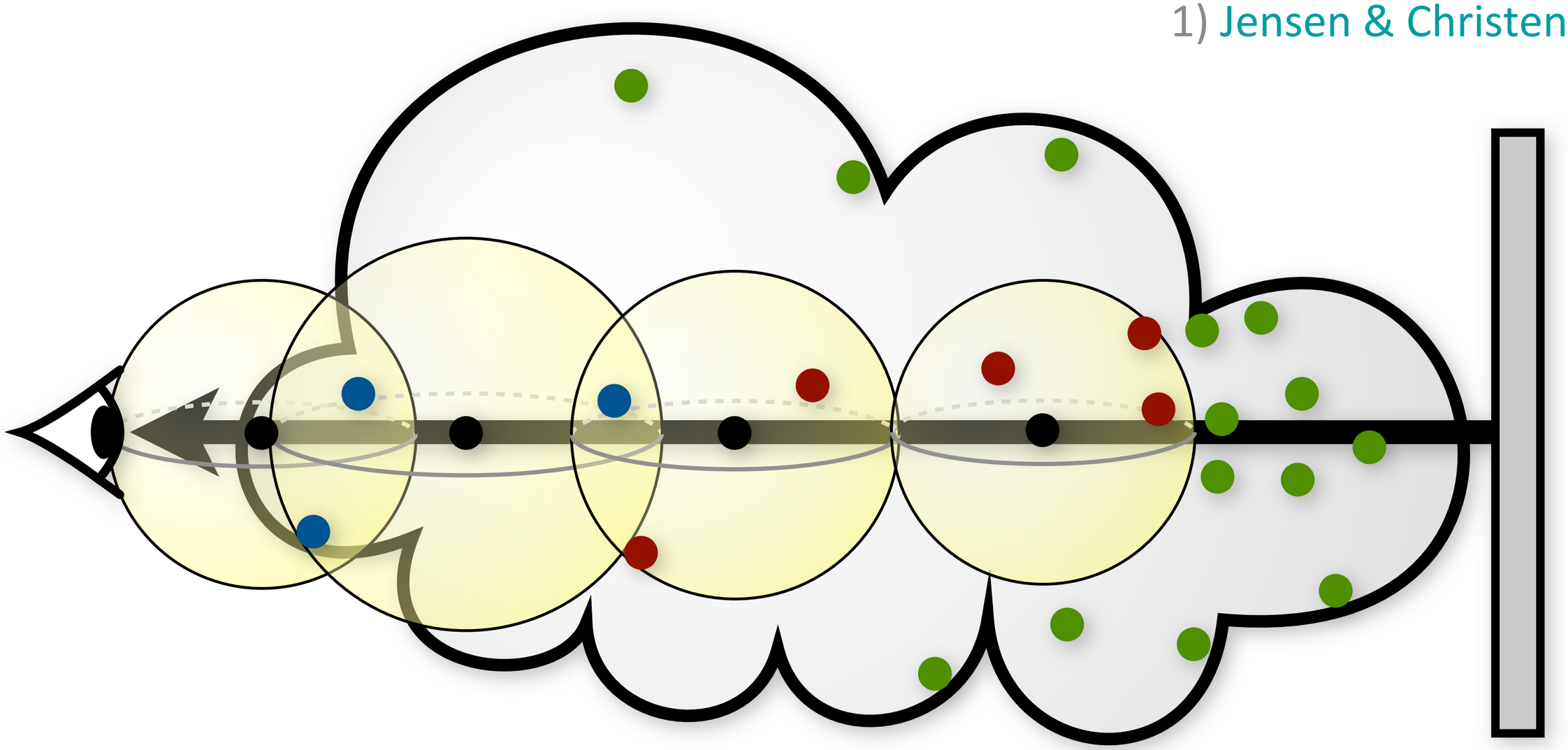
Volumetric Photon Mapping

1) Jensen & Christensen 98



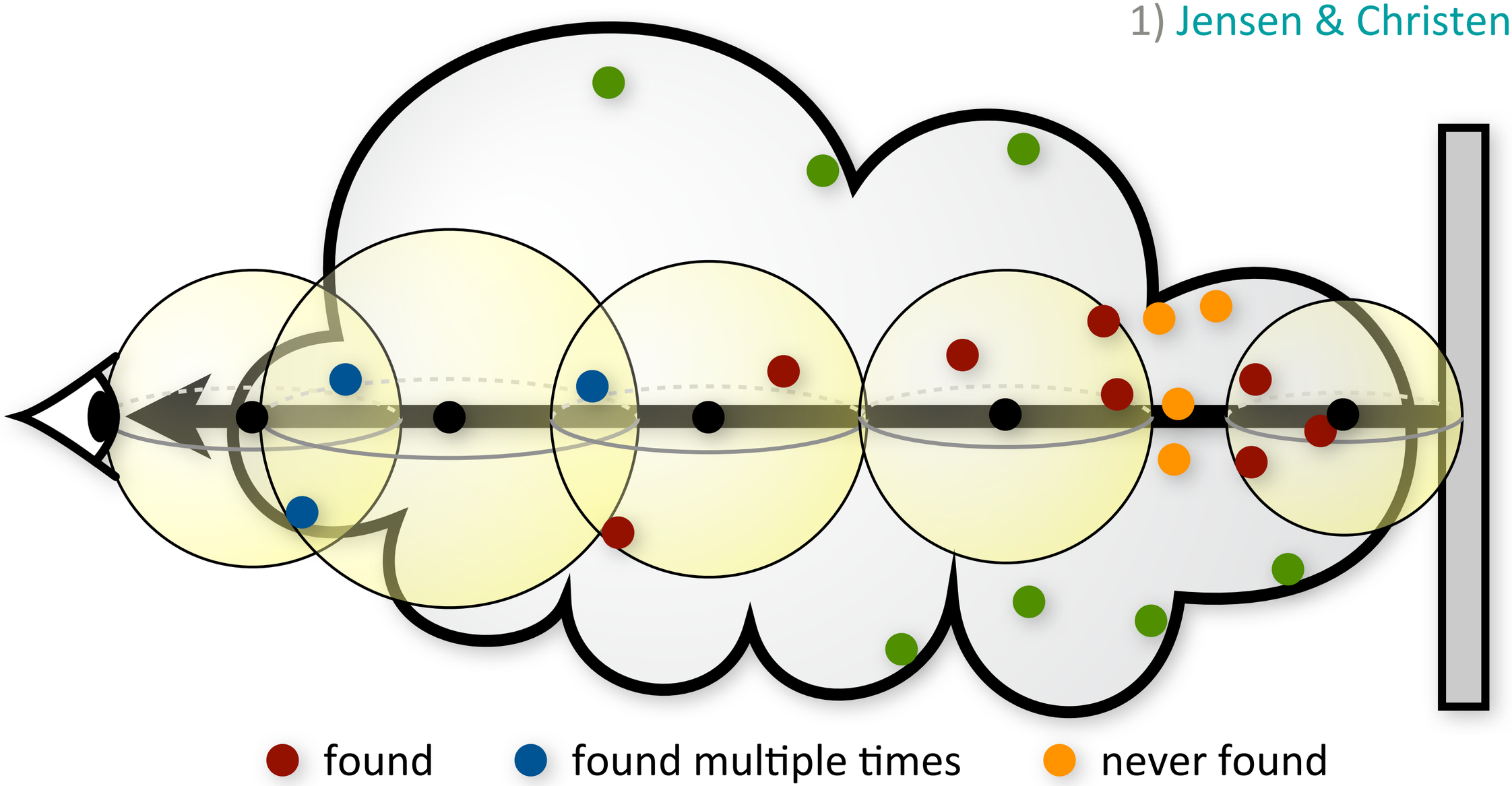
Volumetric Photon Mapping

1) Jensen & Christensen 98



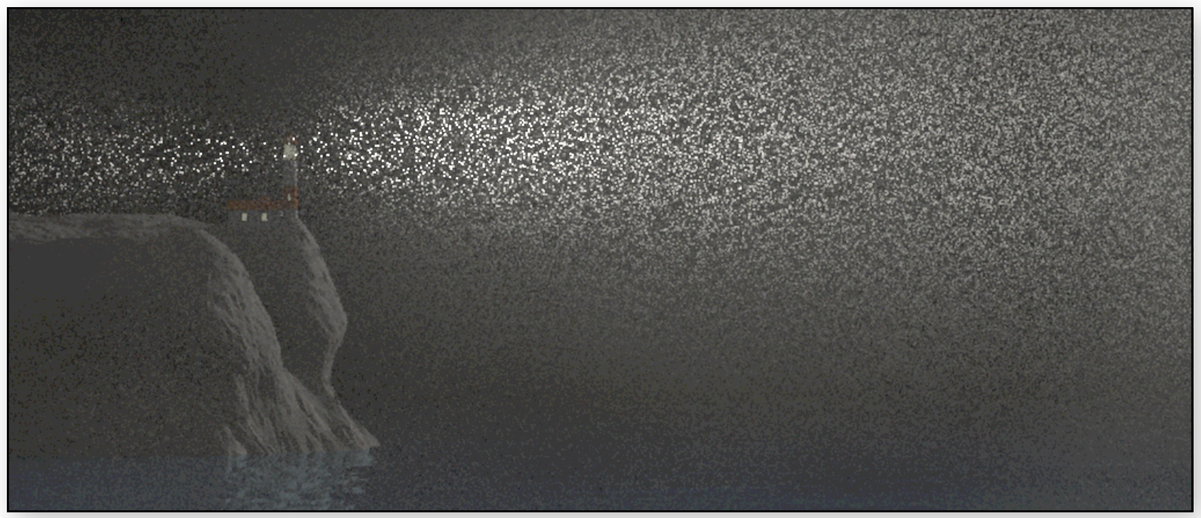
Volumetric Photon Mapping

1) Jensen & Christensen 98



Drawbacks

Large Step-size

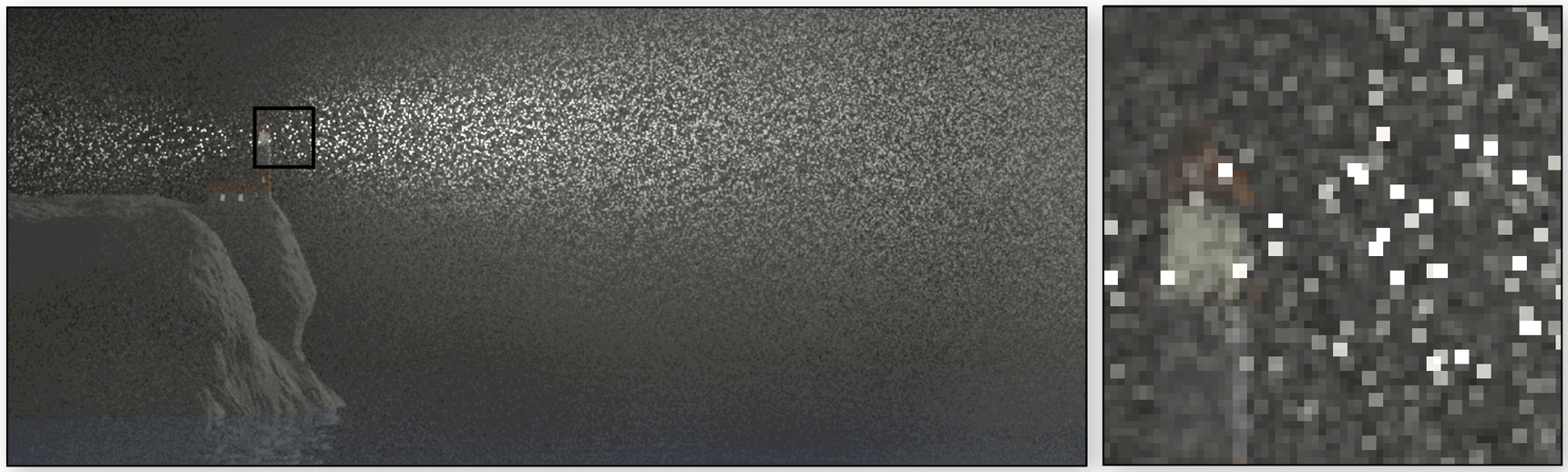


1) Jensen & Christensen 98

Drawbacks

Large Step-size

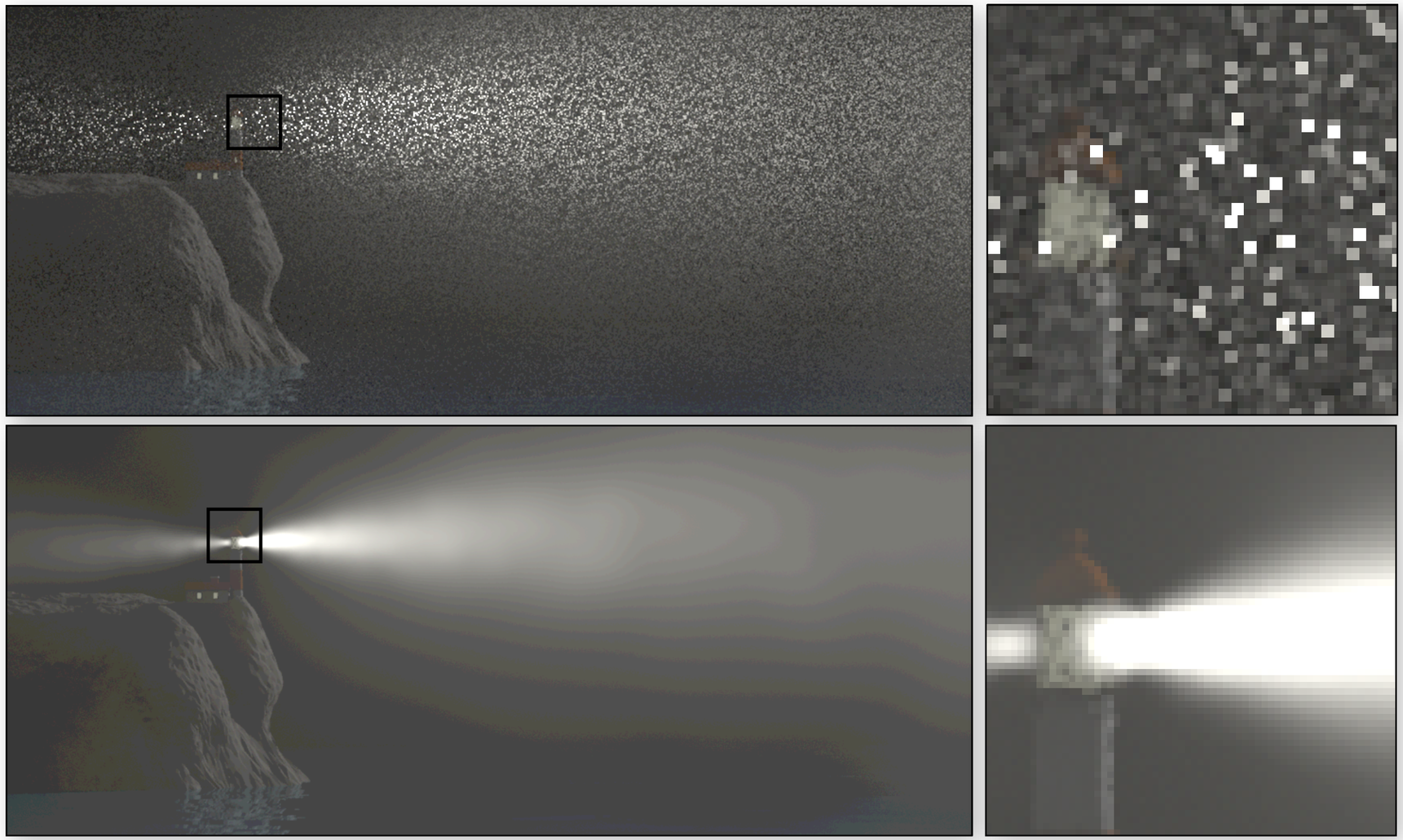
1) Jensen & Christensen 98



Drawbacks

Large Step-size

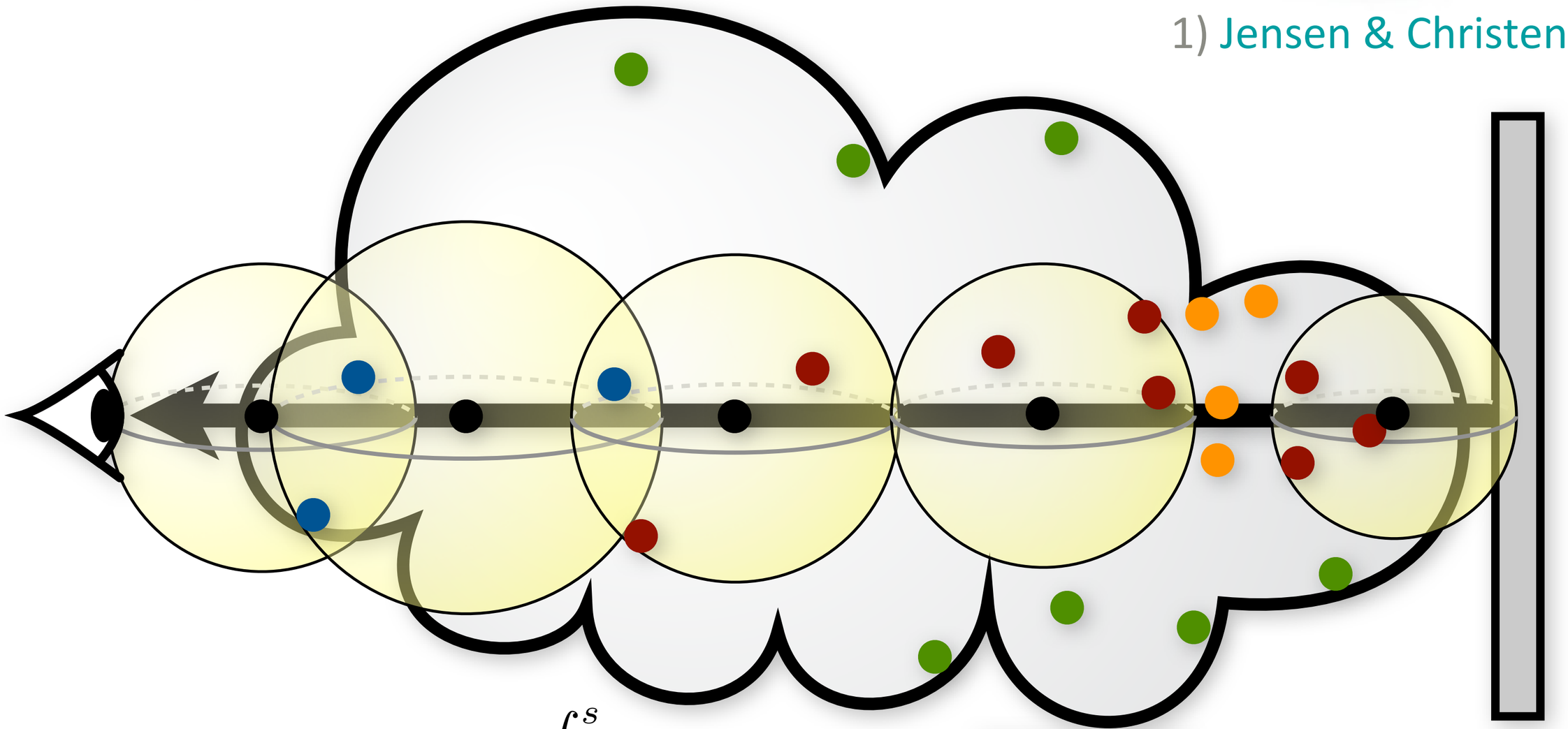
1) Jensen & Christensen 98



Very Small Step-size

Volumetric Photon Mapping

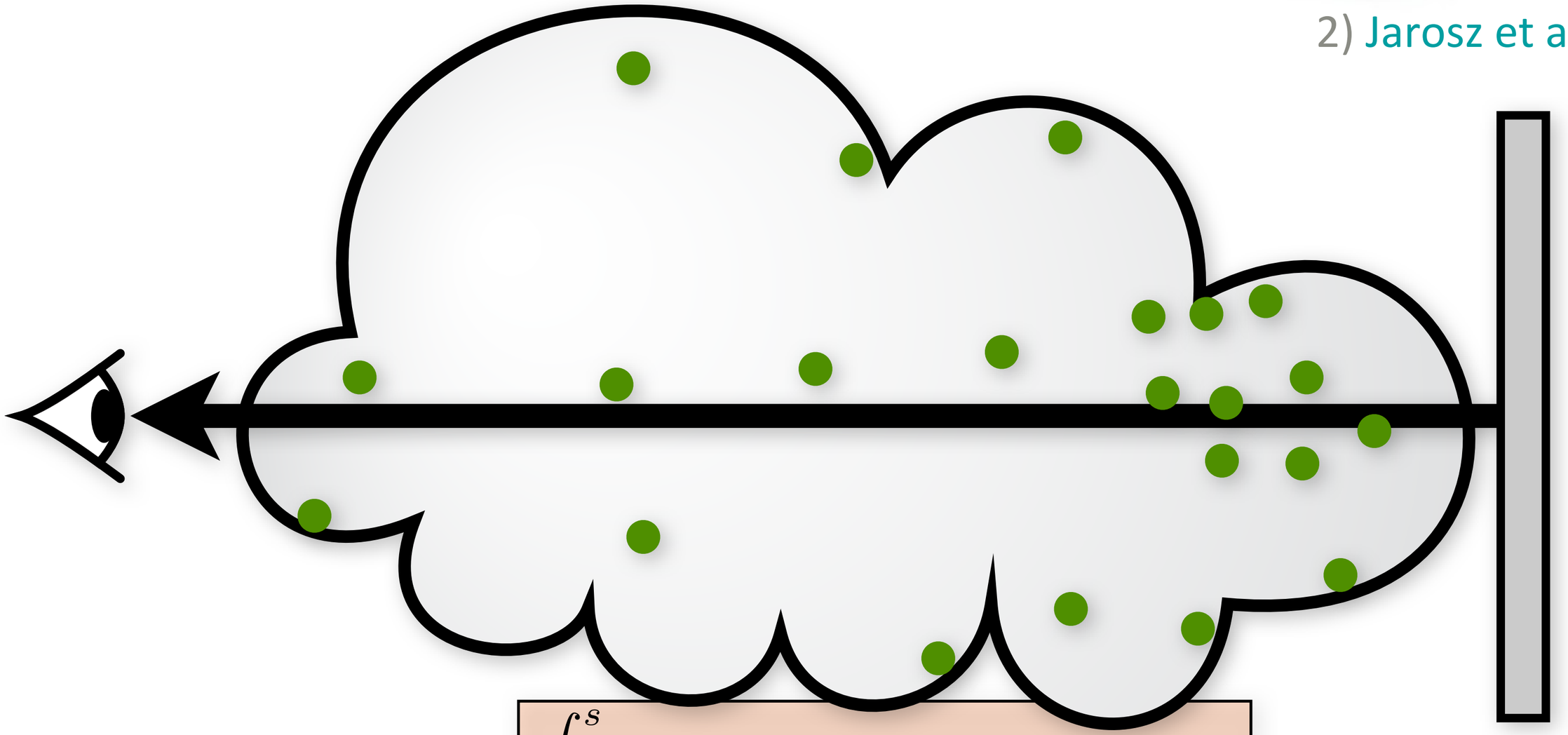
1) Jensen & Christensen 98



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

Beam Radiance Integral

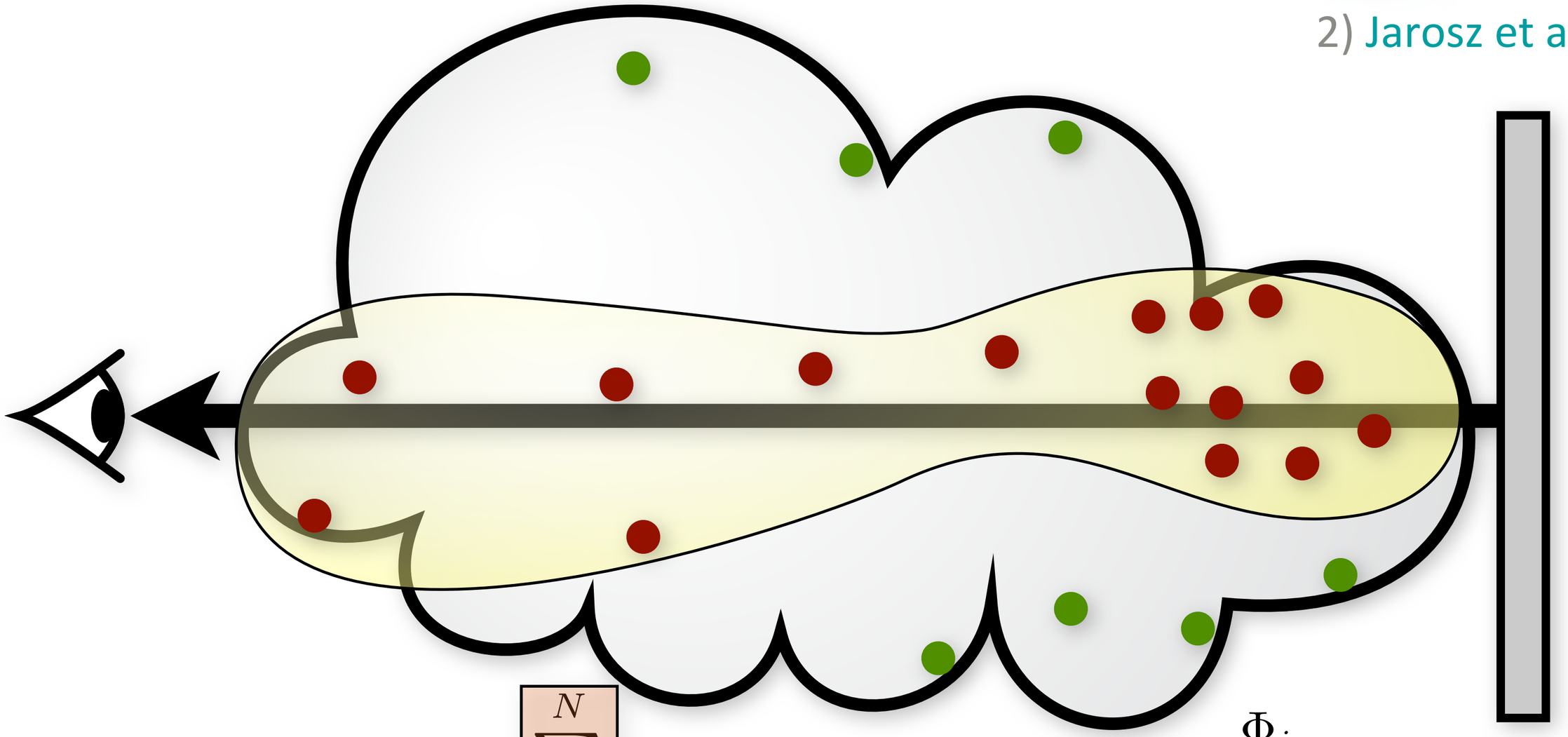
2) Jarosz et al. 2008



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

Beam Radiance Estimation

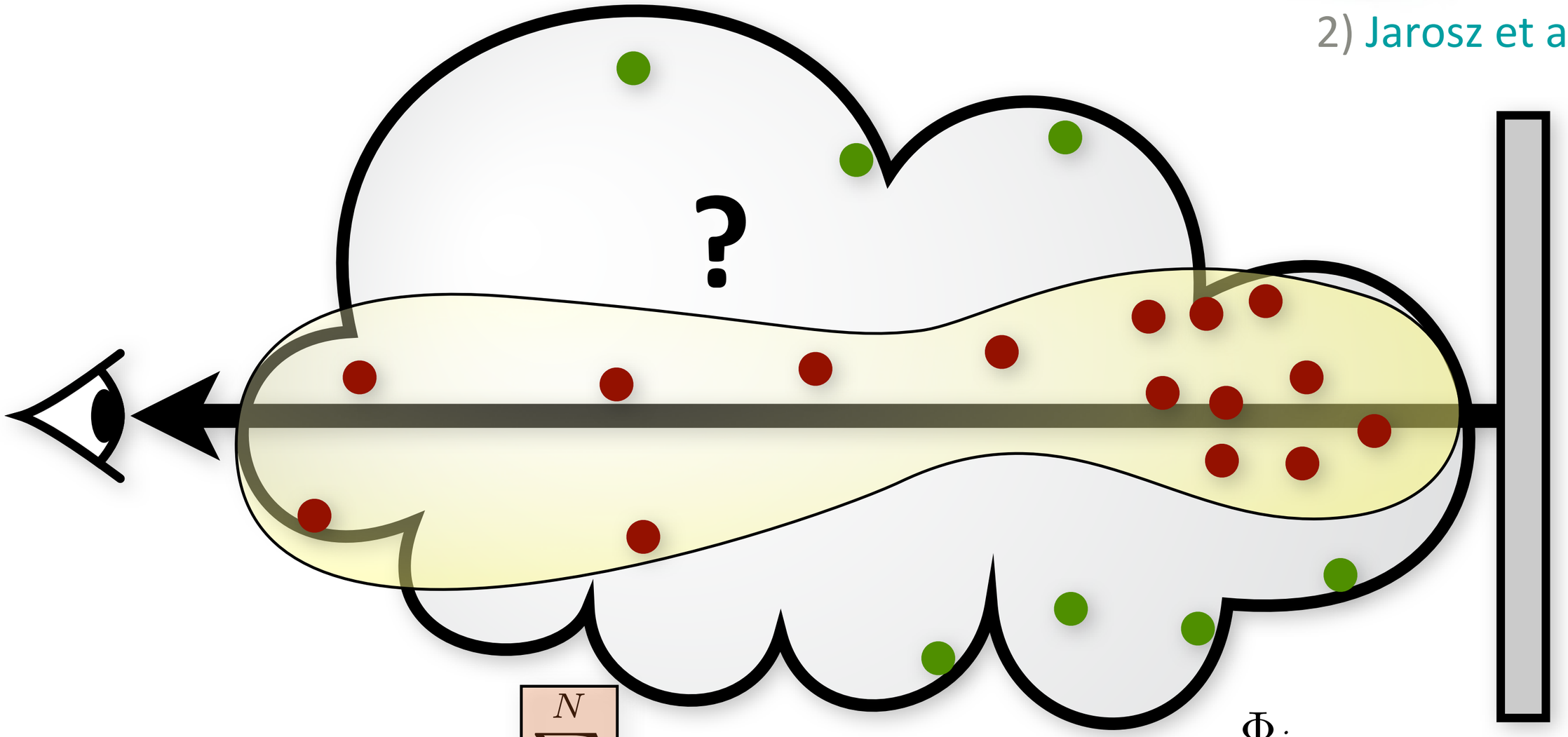
2) Jarosz et al. 2008



$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \vec{\omega}_i) \frac{\Phi_i}{\pi r_i^2}$$

Beam Radiance Estimation

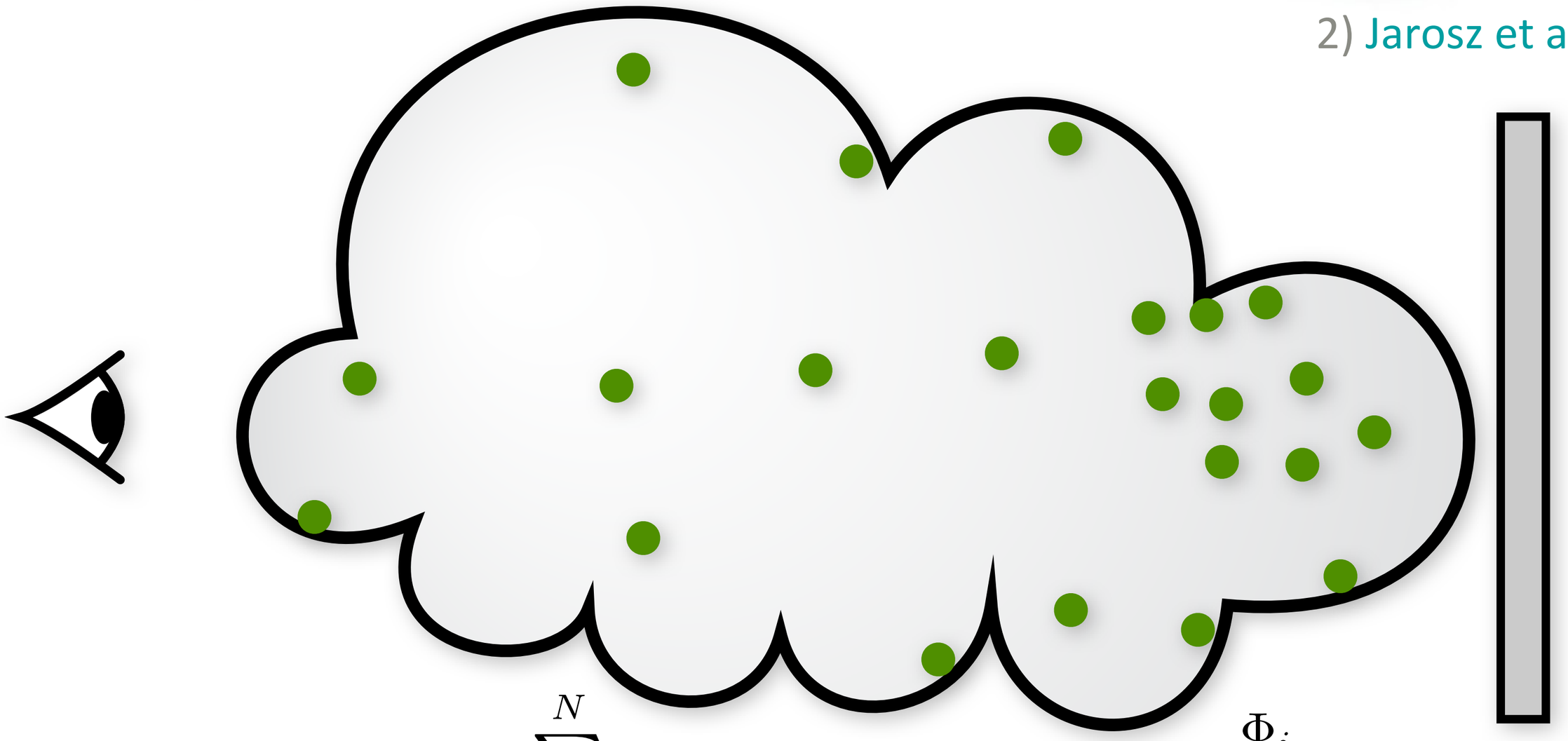
2) Jarosz et al. 2008



$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \vec{\omega}_i) \frac{\Phi_i}{\pi r_i^2}$$

Photon Radius Estimation

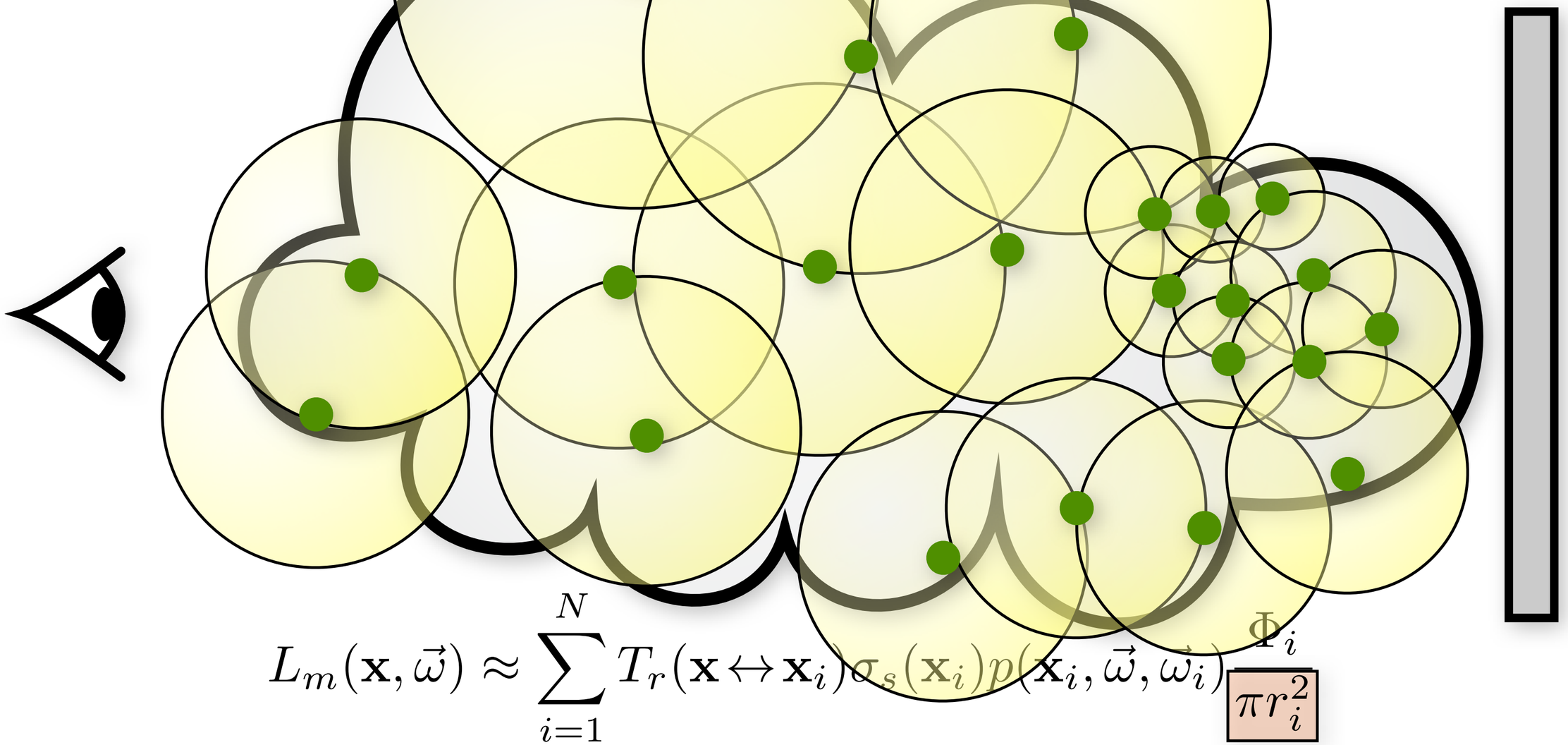
2) Jarosz et al. 2008



$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \vec{\omega}_i) \frac{\Phi_i}{\pi r_i^2}$$

Photon Radius Estimation

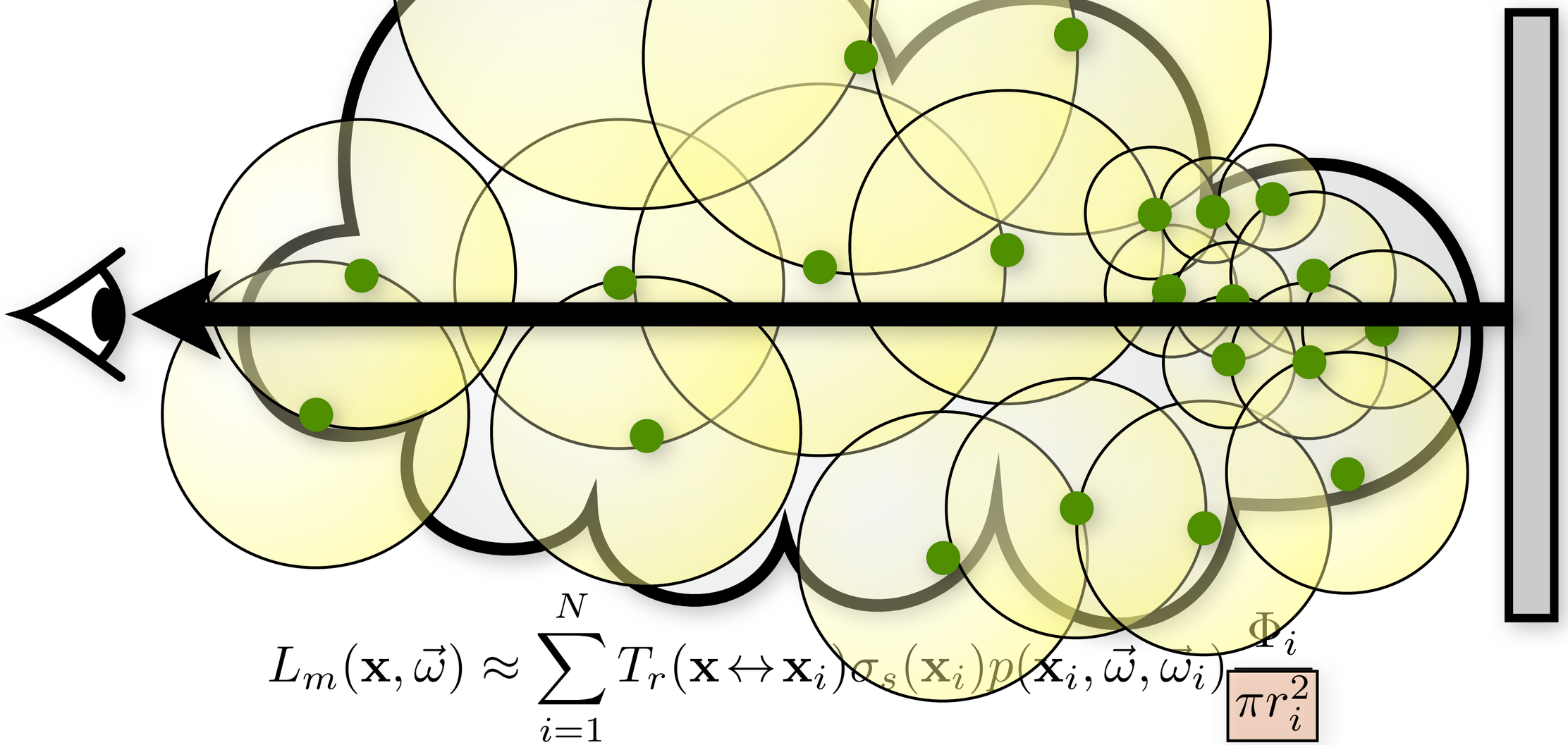
2) Jarosz et al. 2008



$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \omega_i) \frac{\Phi_i}{\pi r_i^2}$$

Photon Radius Estimation

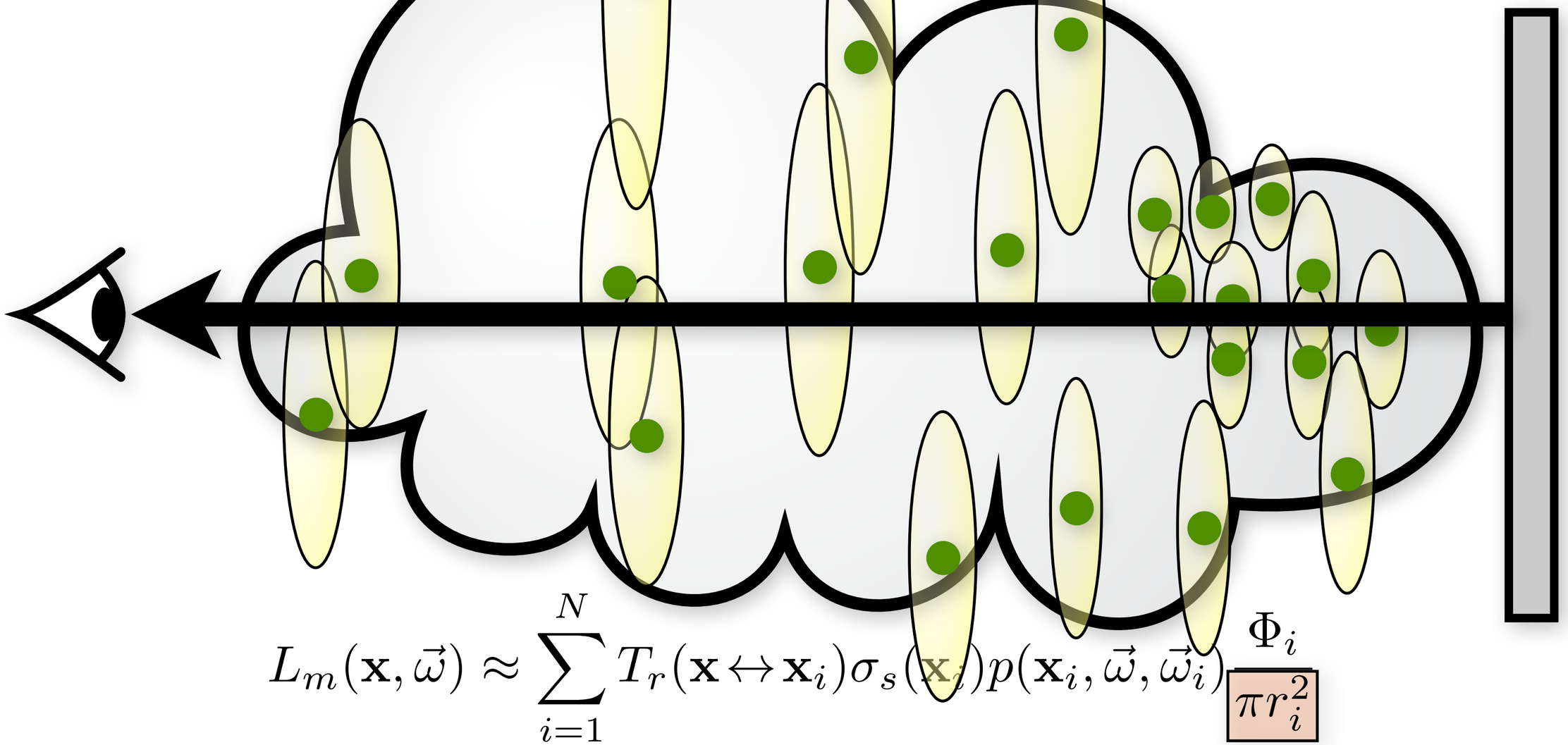
2) Jarosz et al. 2008



$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \omega_i) \frac{\Phi_i}{\pi r_i^2}$$

Photon Radius Estimation

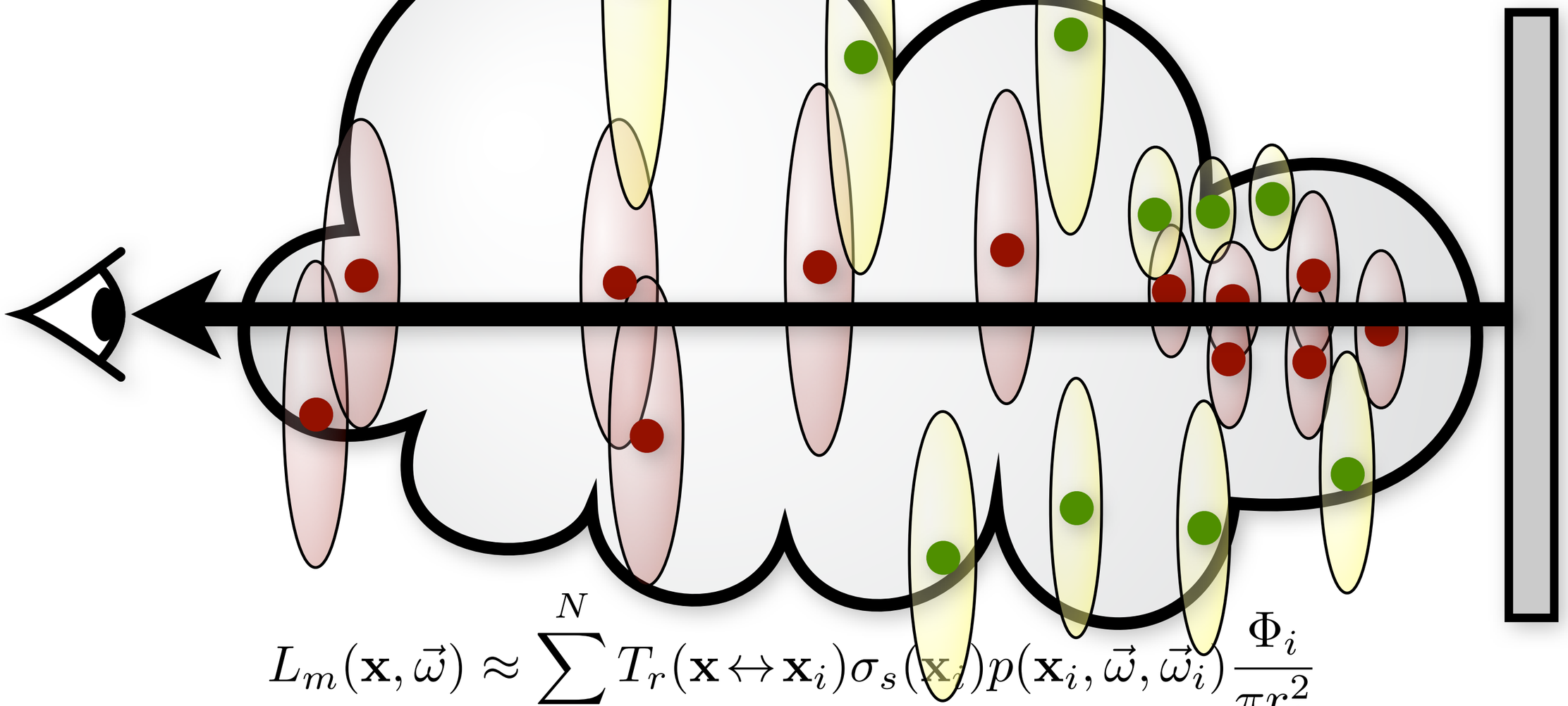
2) Jarosz et al. 2008



$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \vec{\omega}_i) \frac{\Phi_i}{\pi r_i^2}$$

Photon Radius Estimation

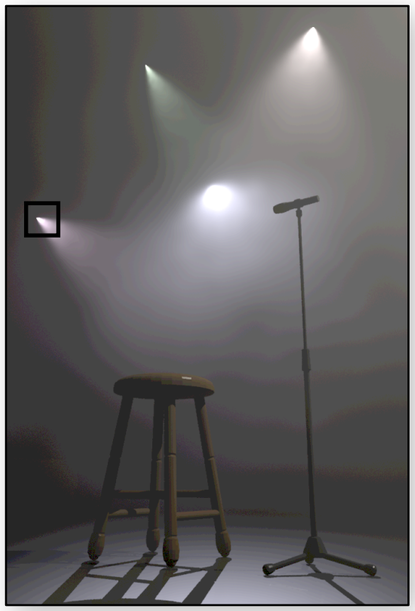
2) Jarosz et al. 2008

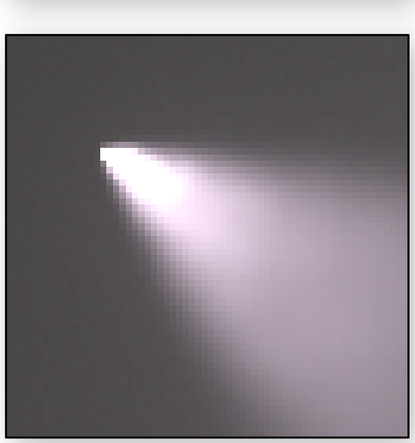


$$L_m(\mathbf{x}, \vec{\omega}) \approx \sum_{i=1}^{N} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i) \sigma_s(\mathbf{x}_i) p(\mathbf{x}_i, \vec{\omega}, \vec{\omega}_i) \frac{\Phi_i}{\pi r_i^2}$$

Concert Stage

Beam Estimate

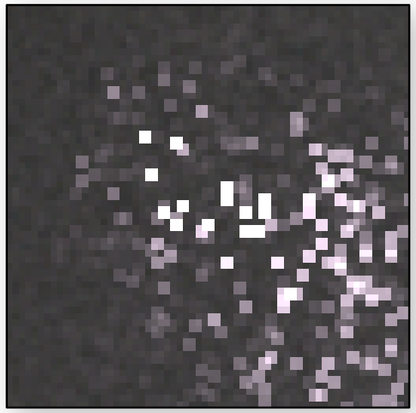




(6:22)

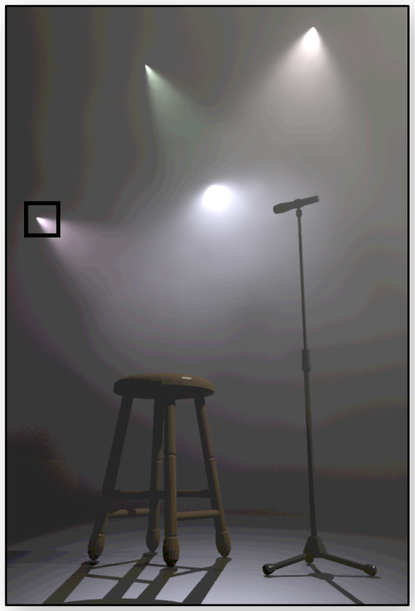
Concert Stage

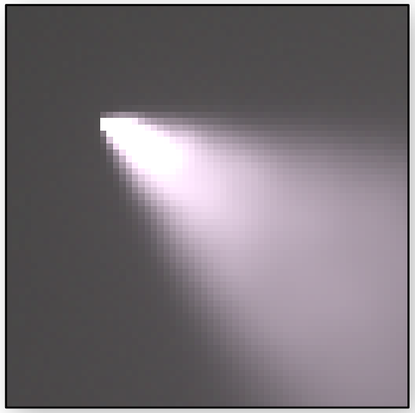
Trad. Estimate



(6:38)

Beam Estimate

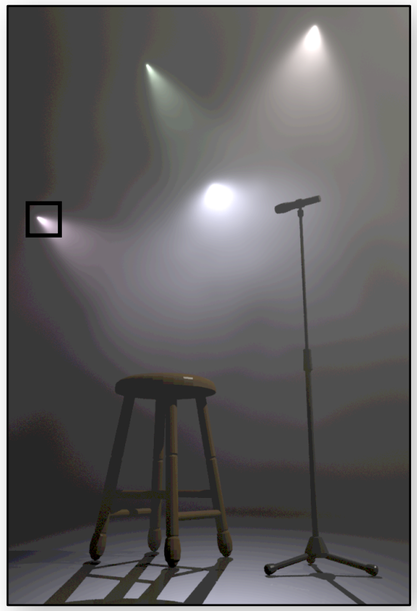




(6:22)

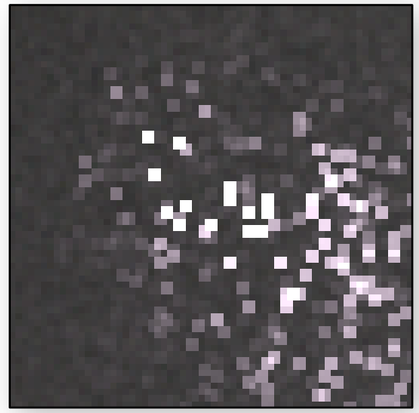
Concert Stage

Trad. Estimate



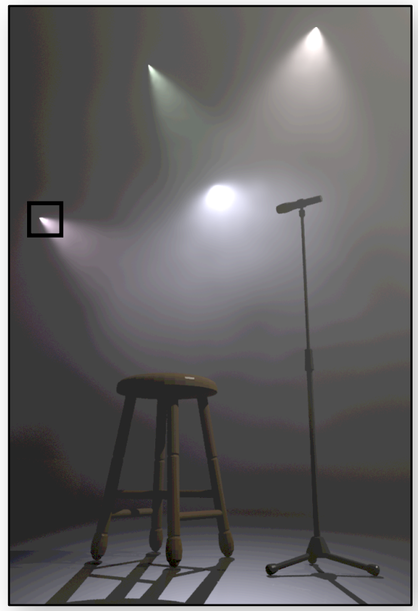
(∞)

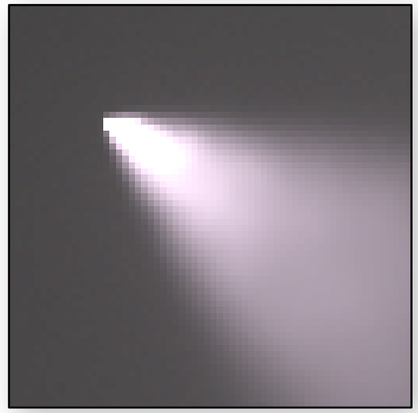
Trad. Estimate



(6:38)

Beam Estimate

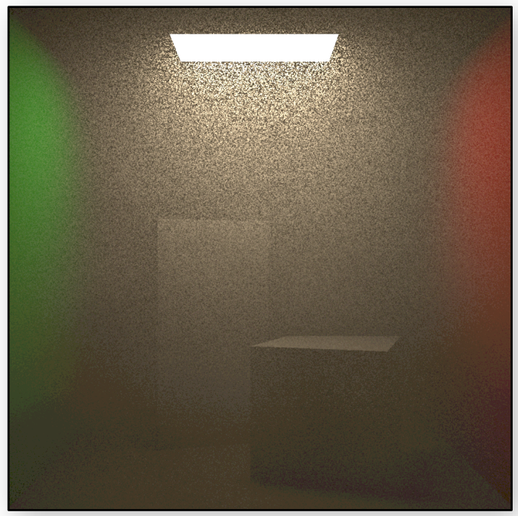




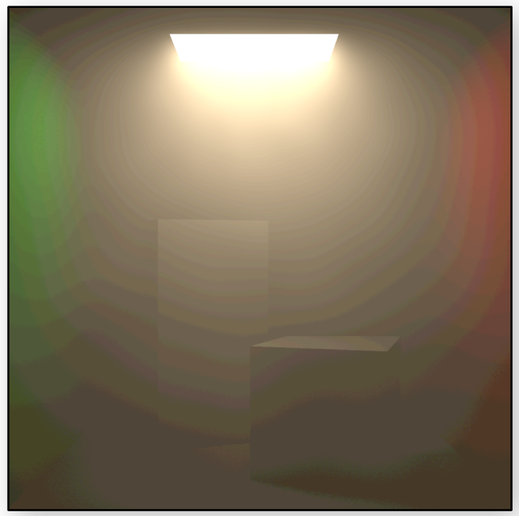
(6:22)

Smoky Cornell Box

Trad. Estimate

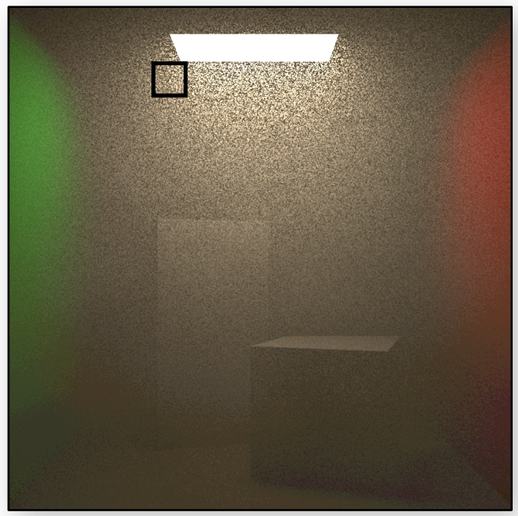


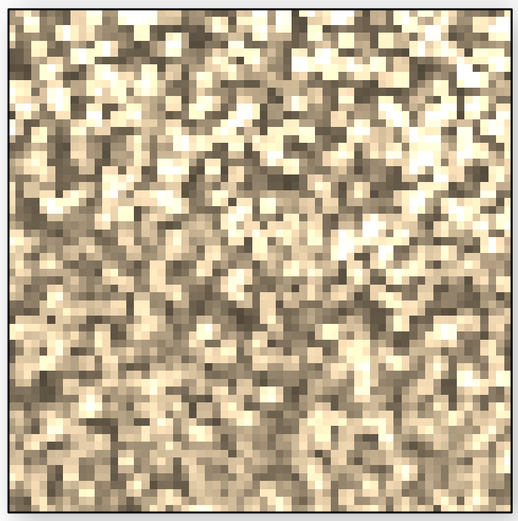
Beam Estimate



Smoky Cornell Box

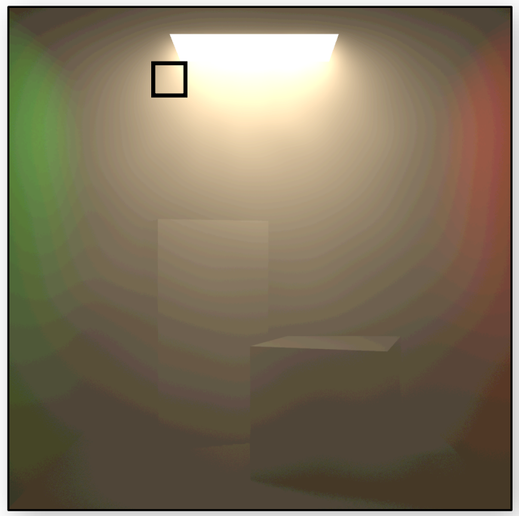
Trad. Estimate

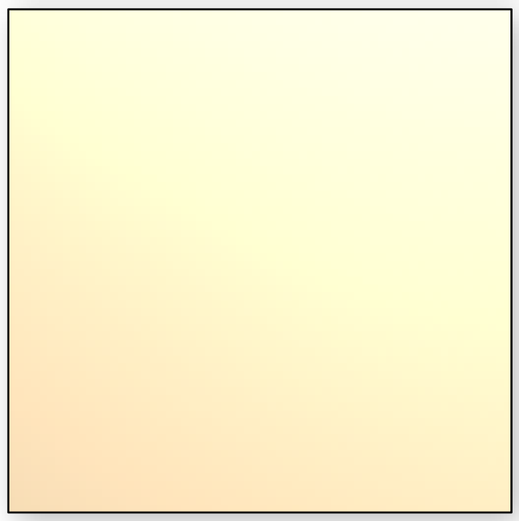




(4:03)

Beam Estimate





(3:35)

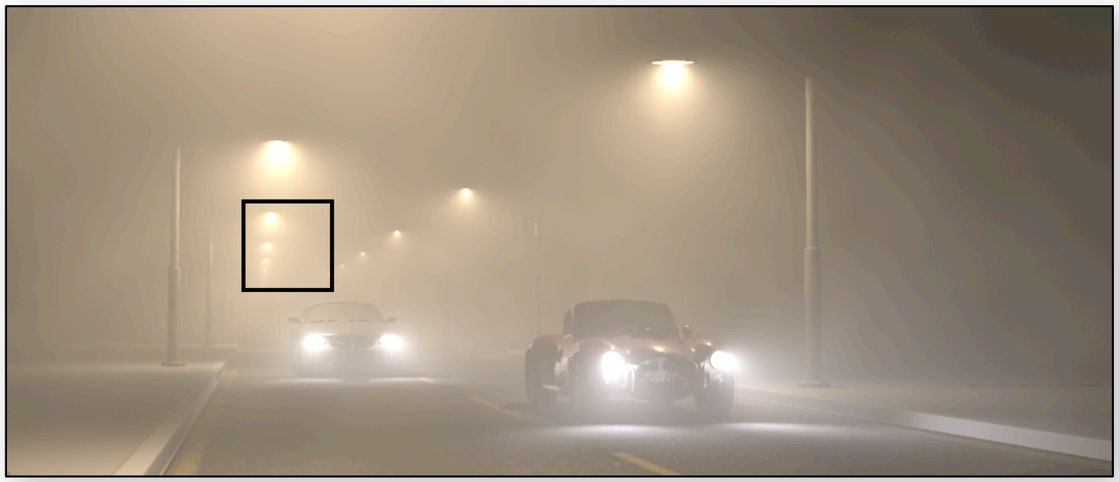
Cars on Foggy Street

Beam Estimate

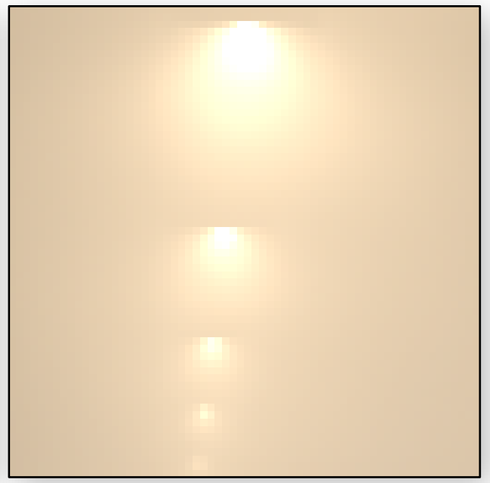
Traditional Estimate

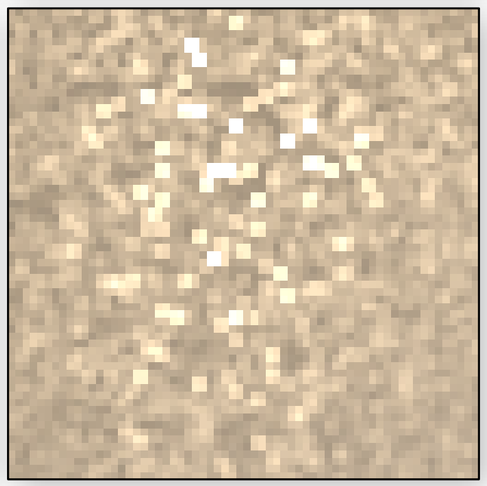
Cars on Foggy Street

Beam Estimate



(1:53)





Traditional Estimate

(2:02)

Questions?

Thursday, August 23, 12

11:20