State of the Art in Photon Density Estimation:

From Photons to Beams

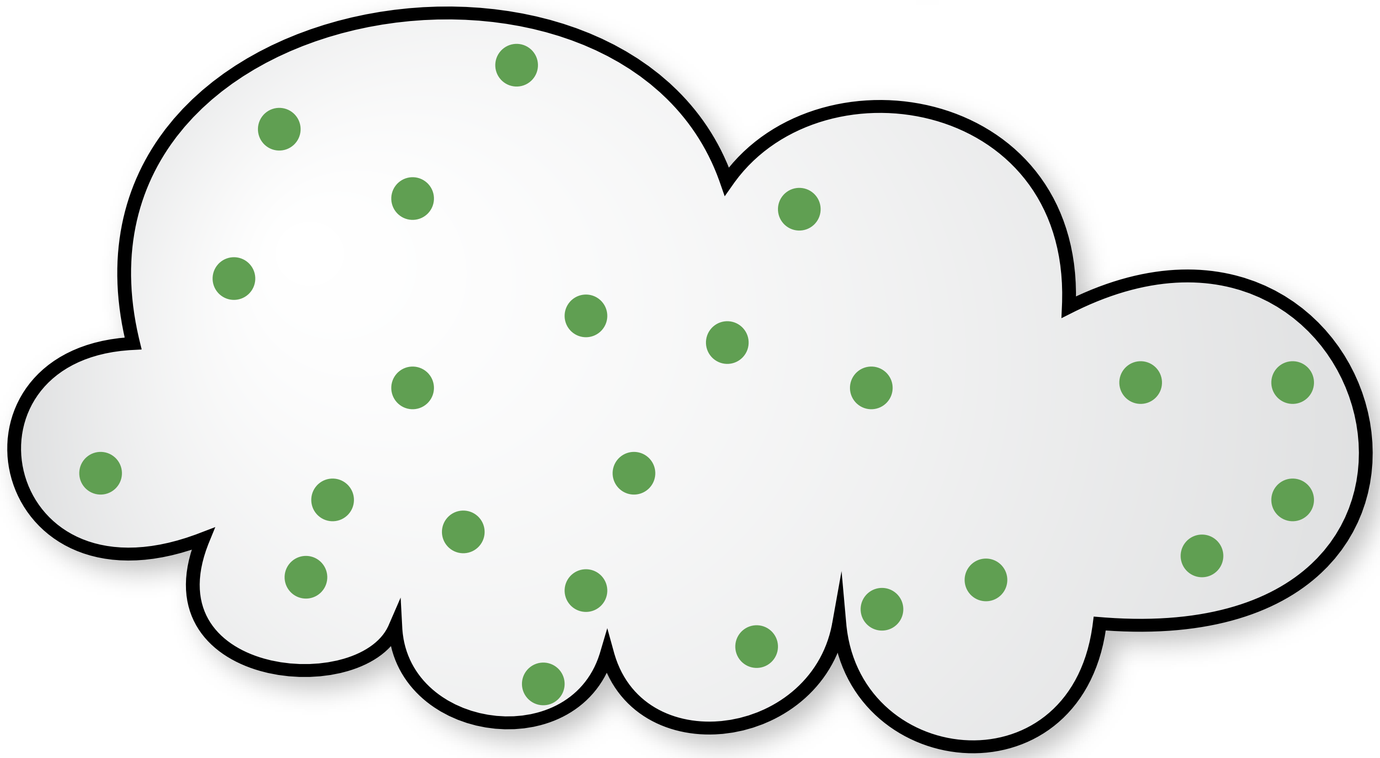
Wojciech Jarosz

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

So Far...

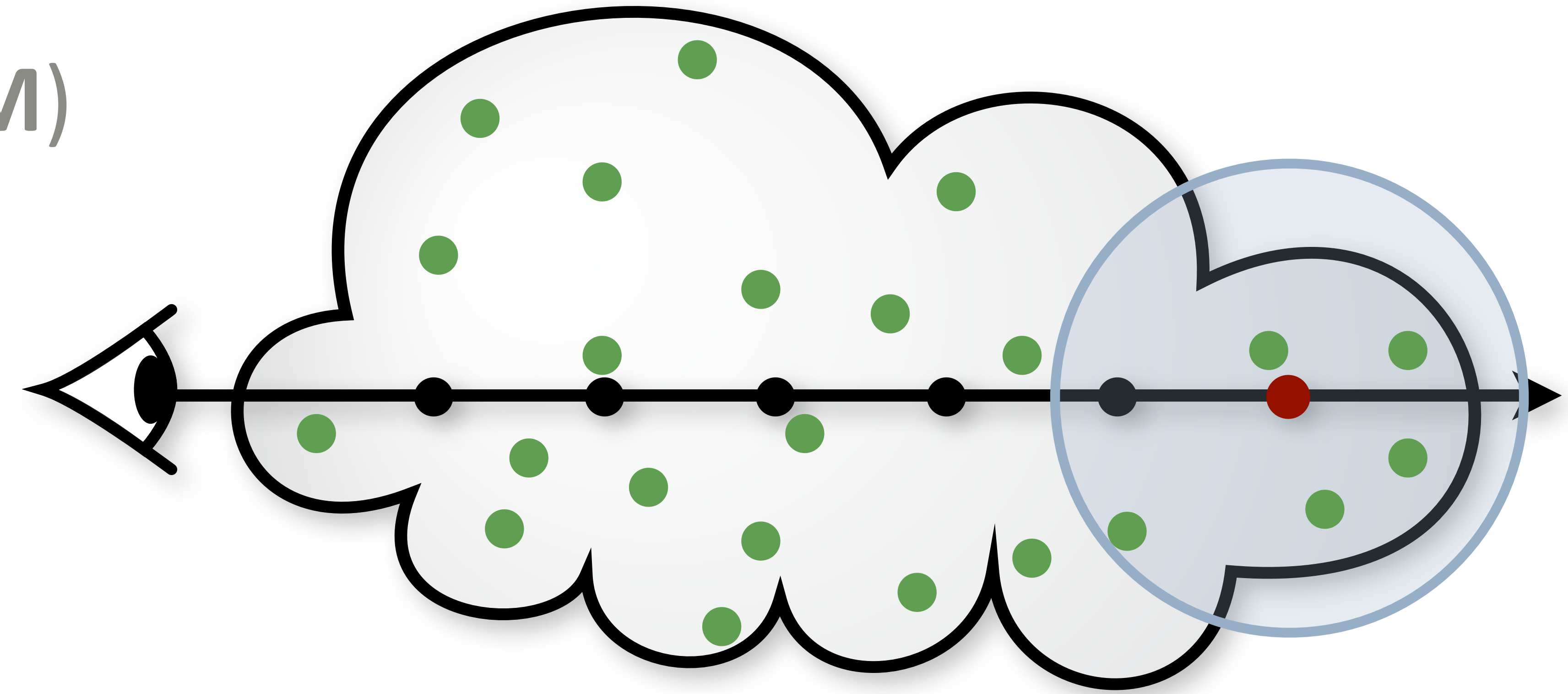
- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]

- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]



So Far...

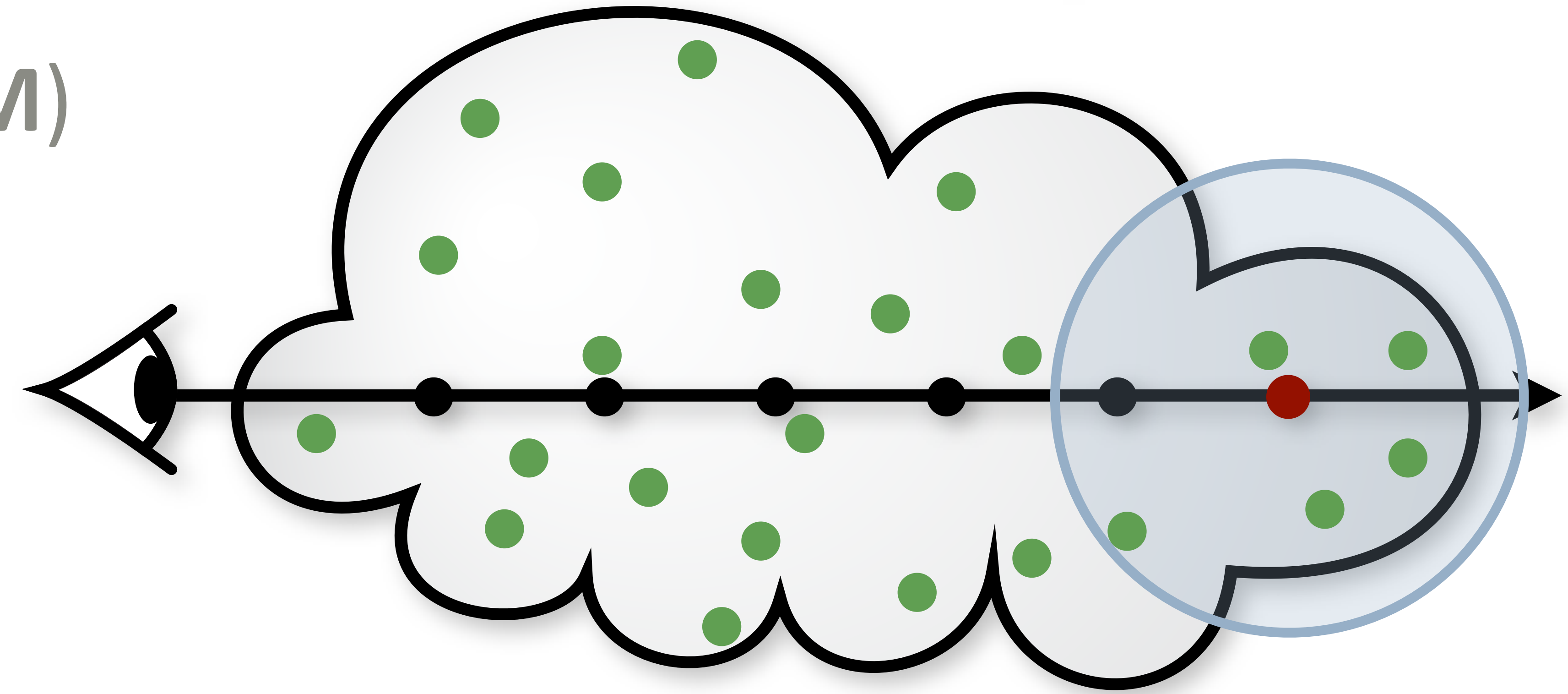
- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]



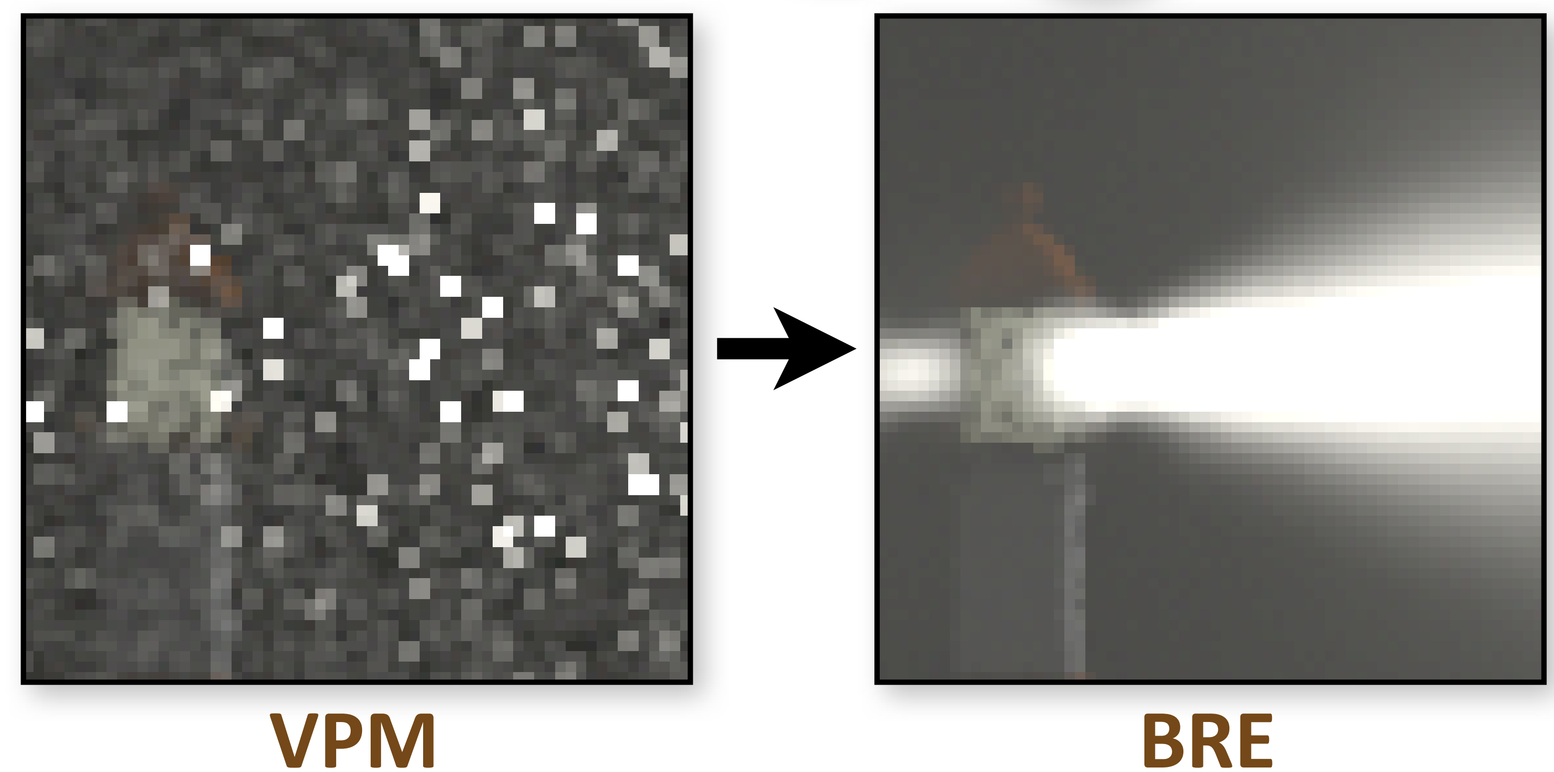
- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]

So Far...

- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]

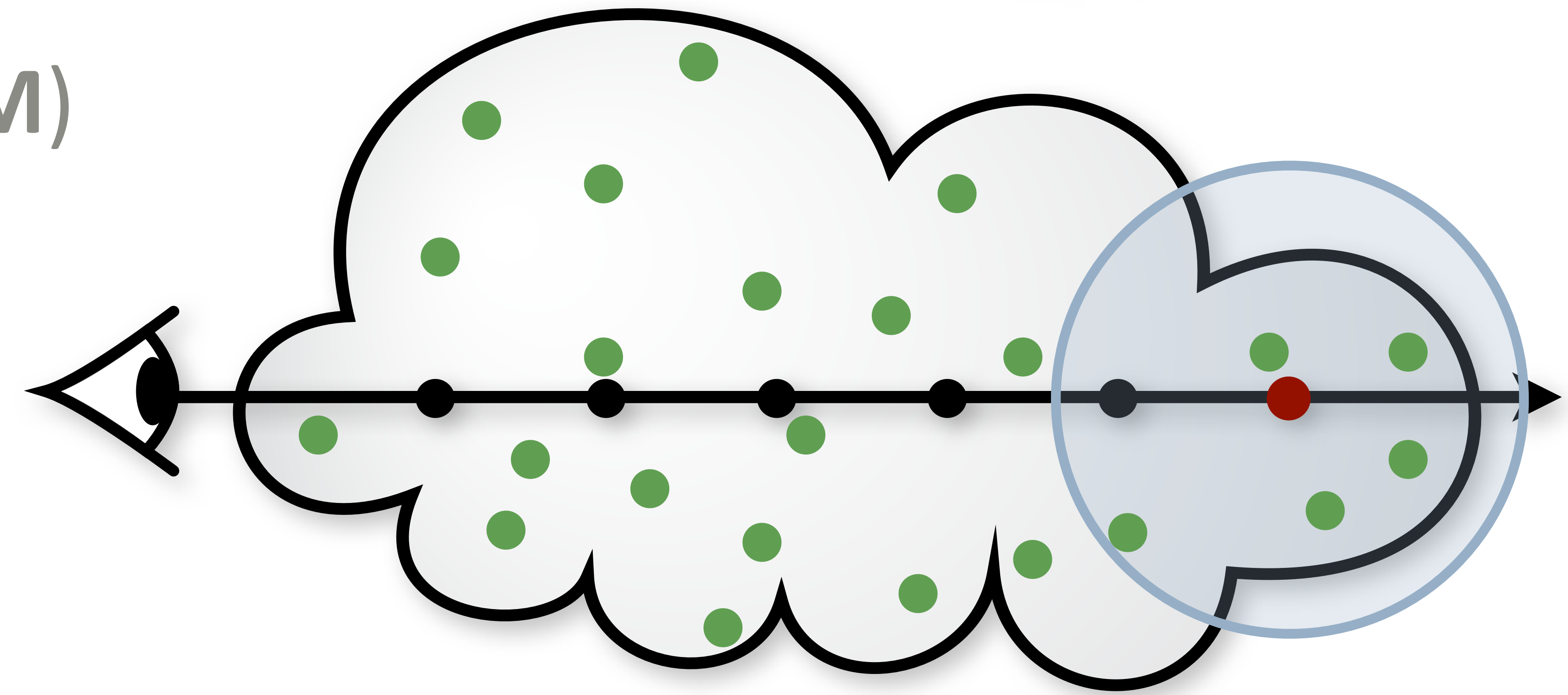


- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]

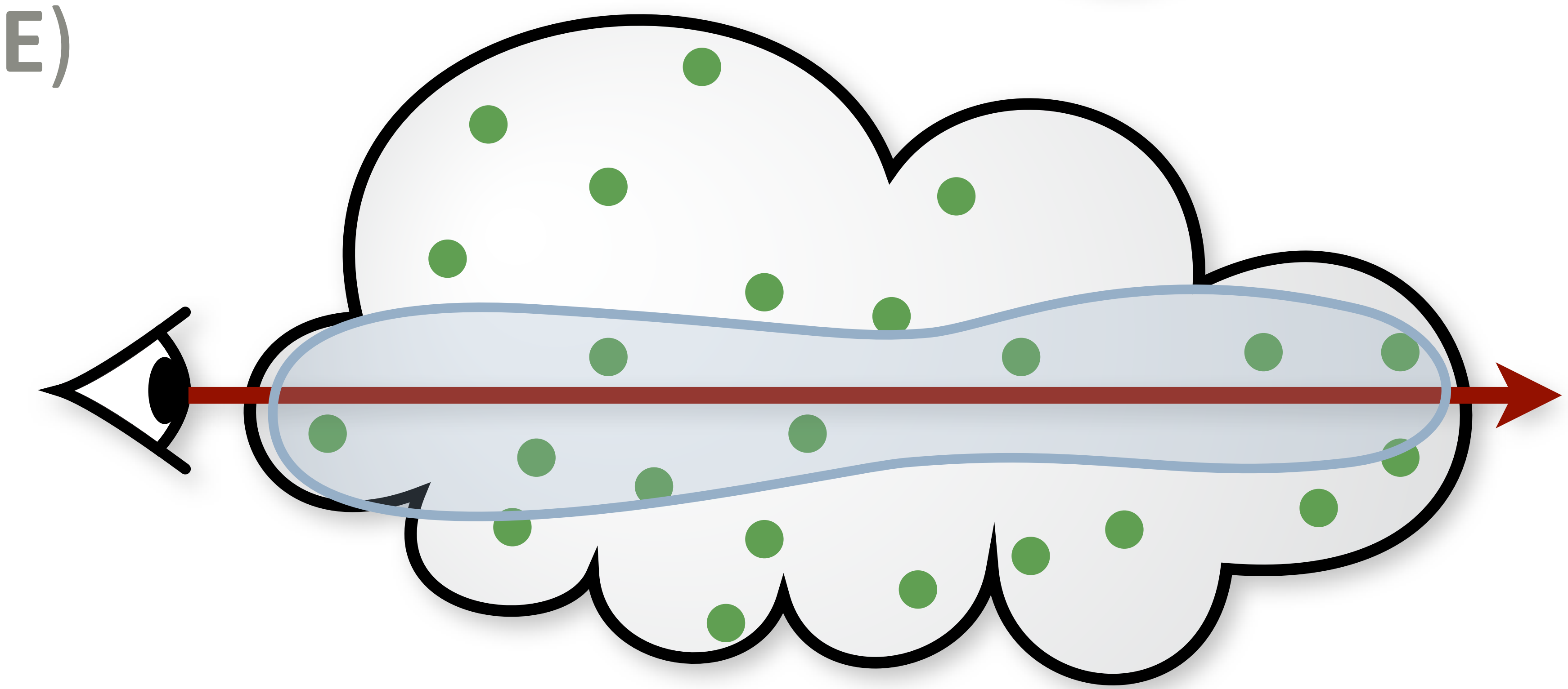


So Far...

- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]



- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]

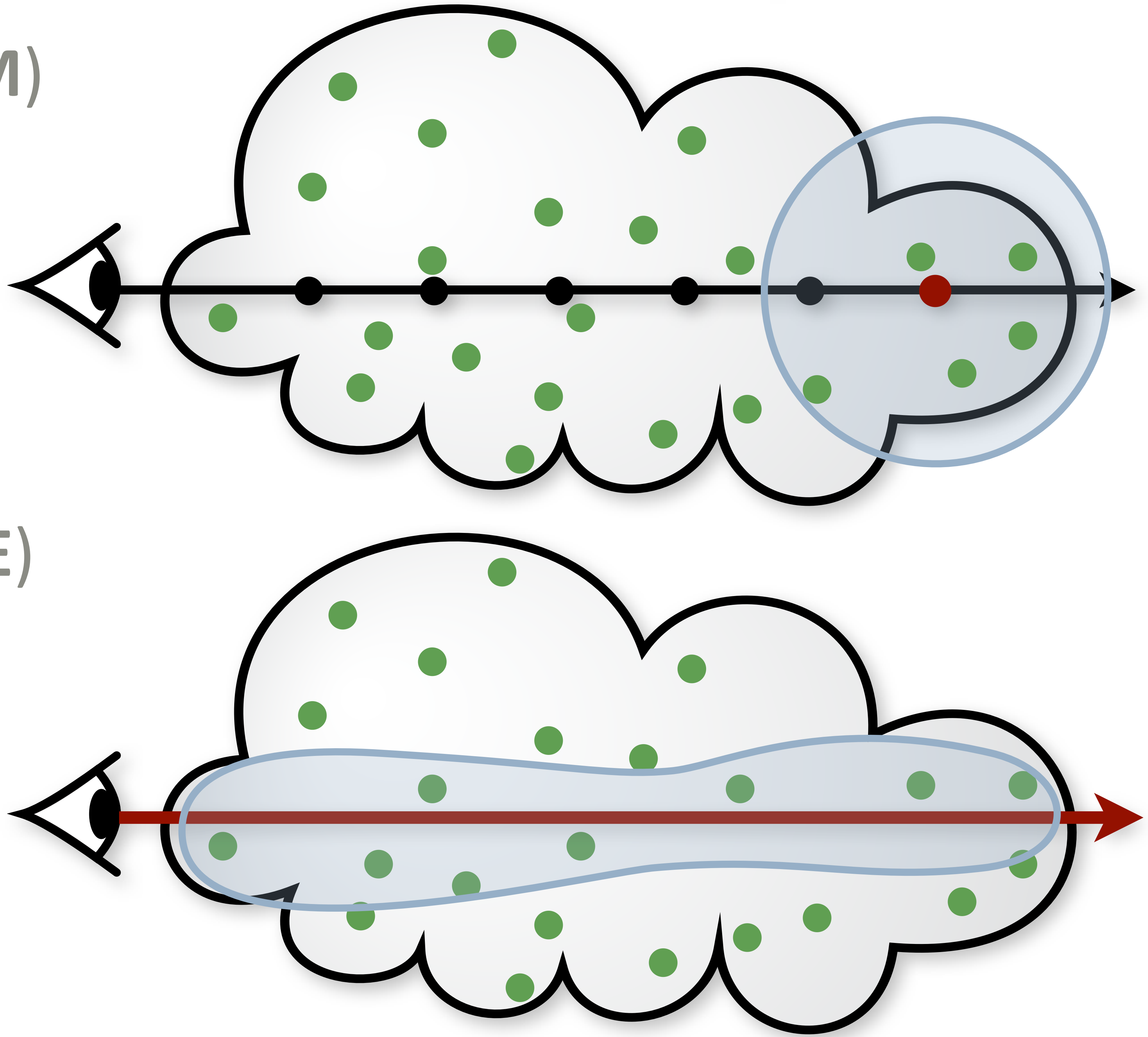


So Far...

- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]

Query
Point

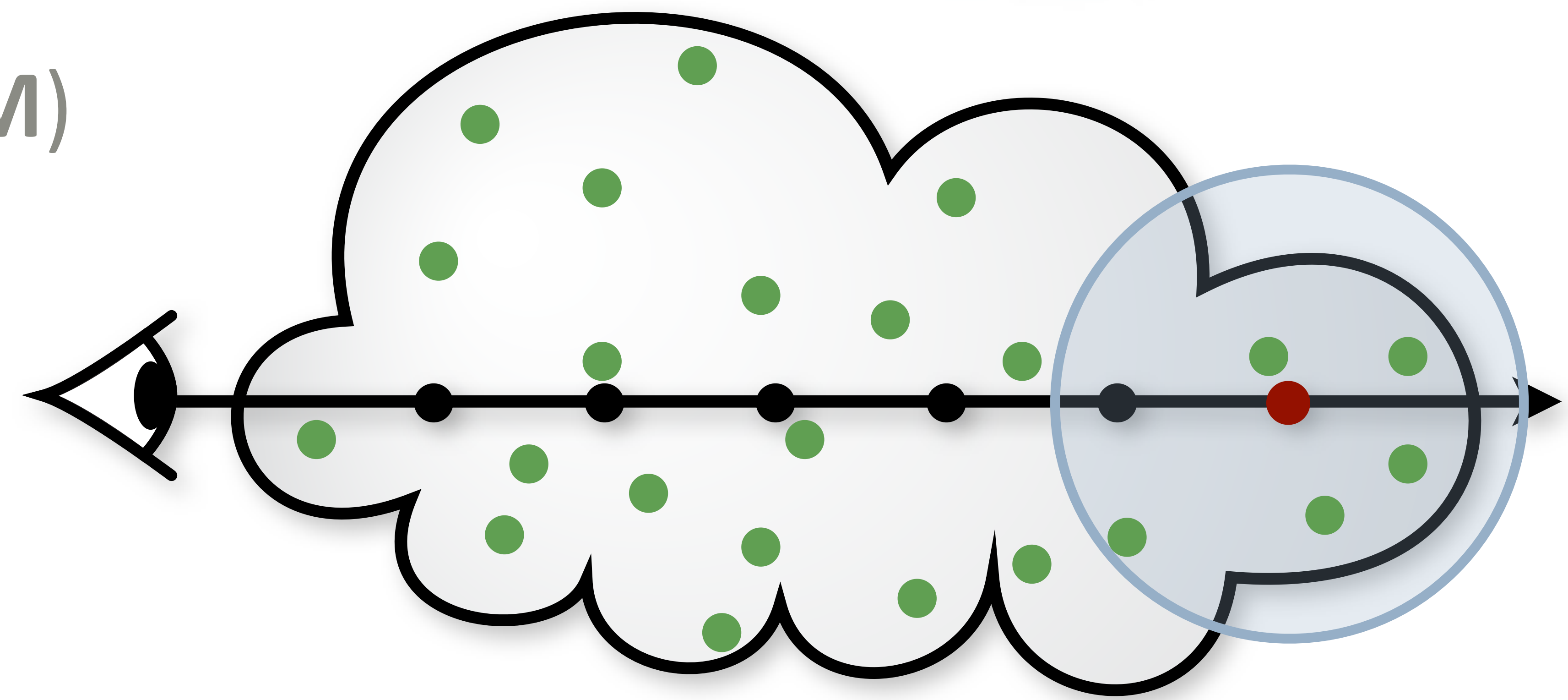
- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]



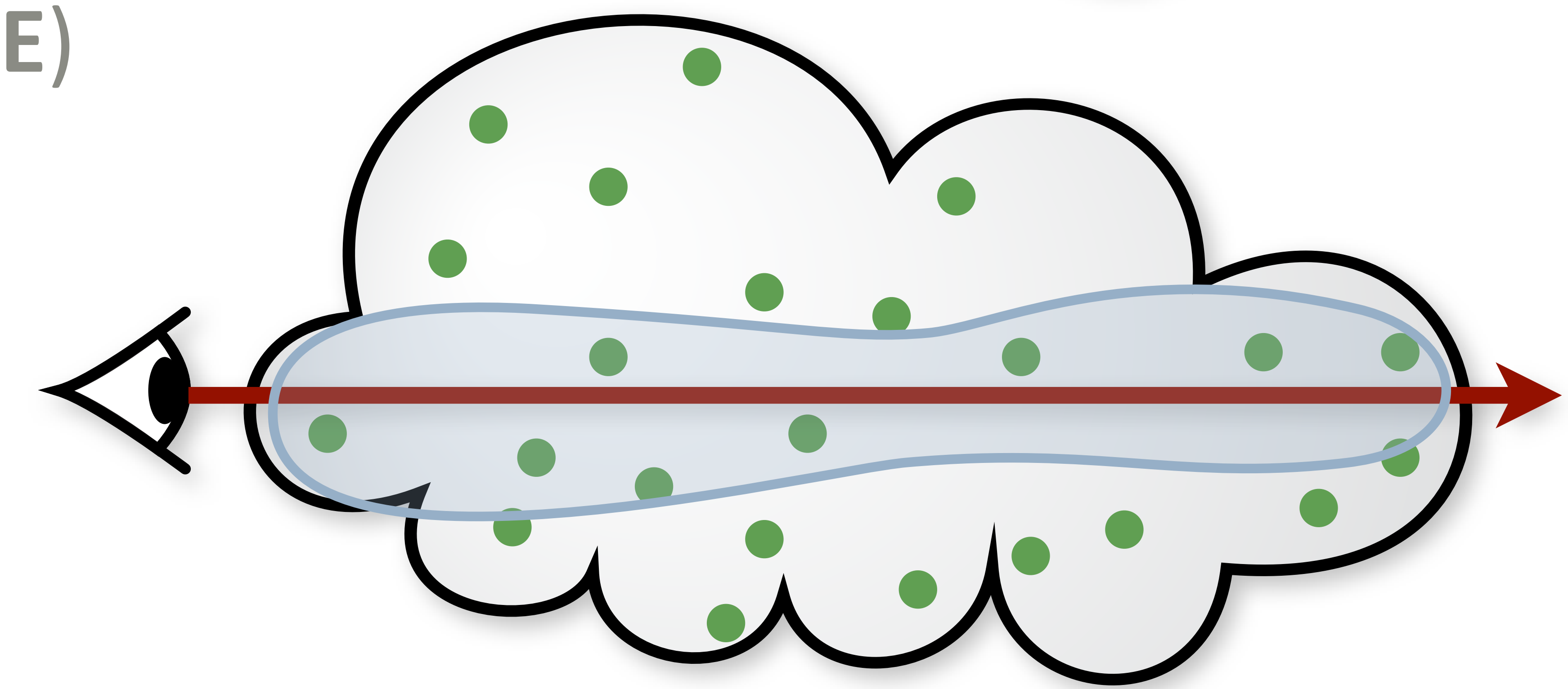
So Far...

- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]

Query	x	Data
Point	x	Point



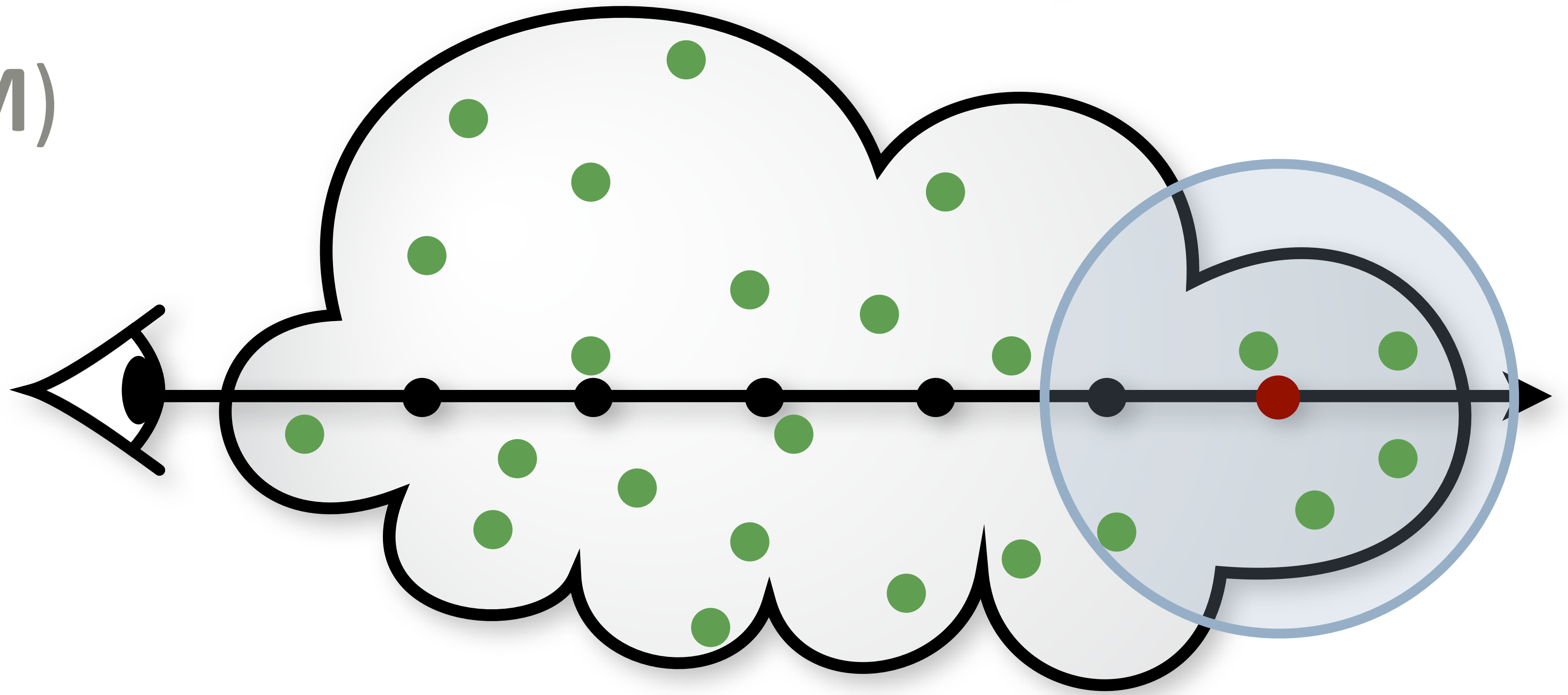
- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]



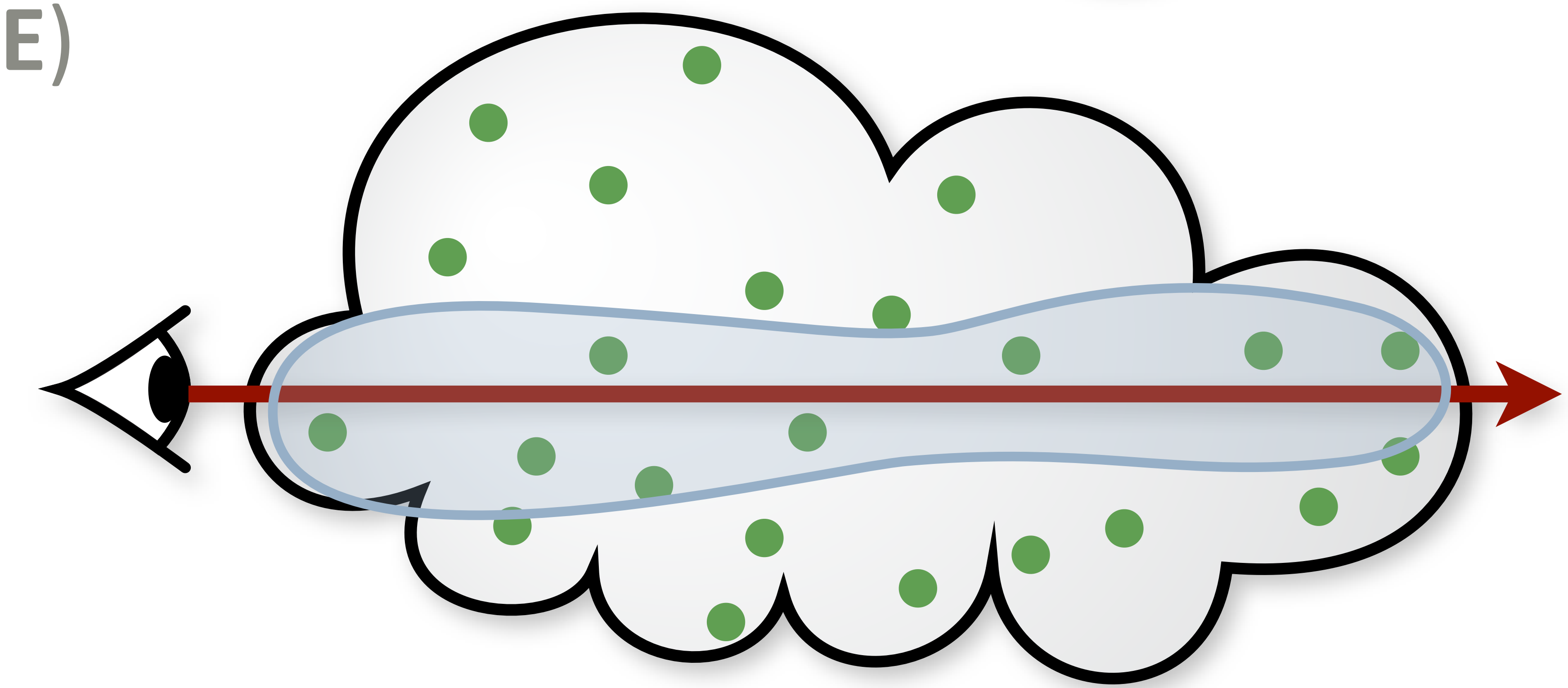
So Far...

- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)



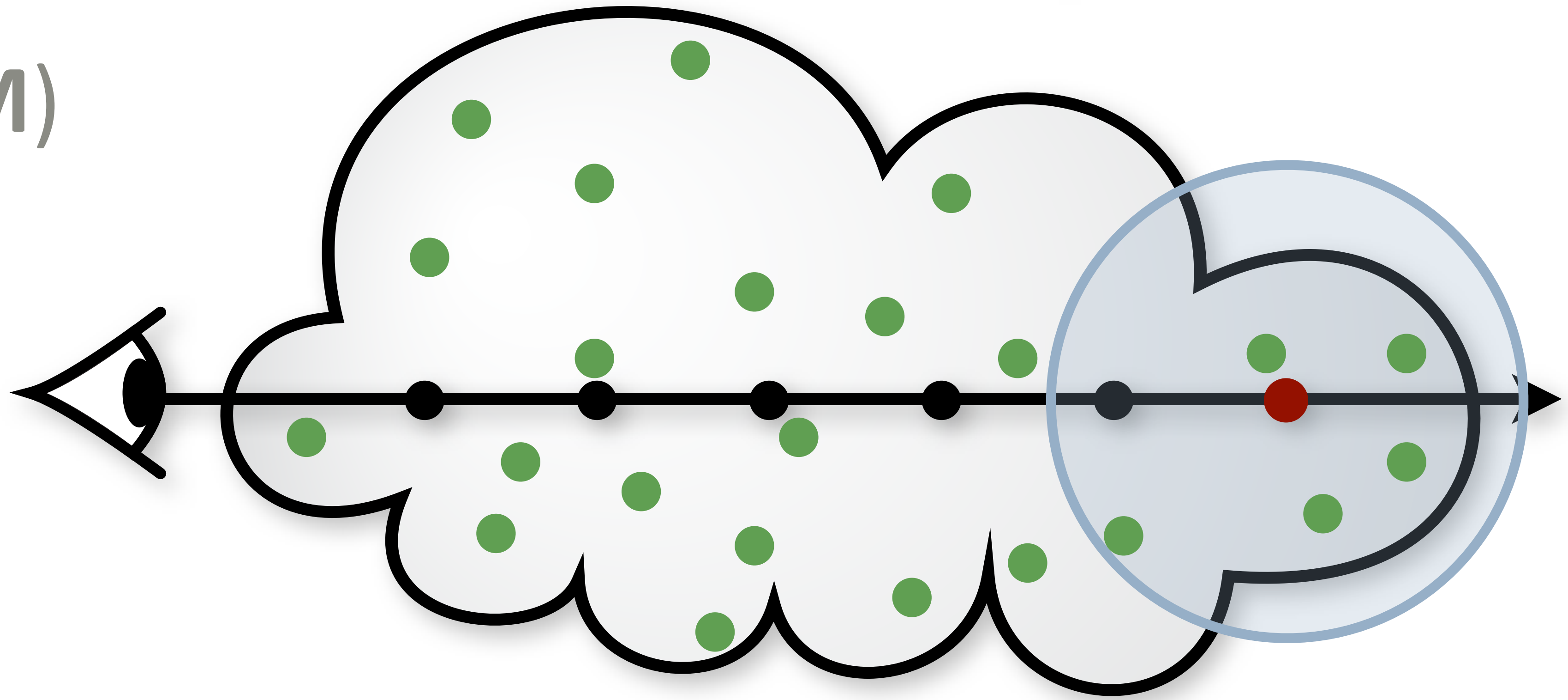
- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]



So Far...

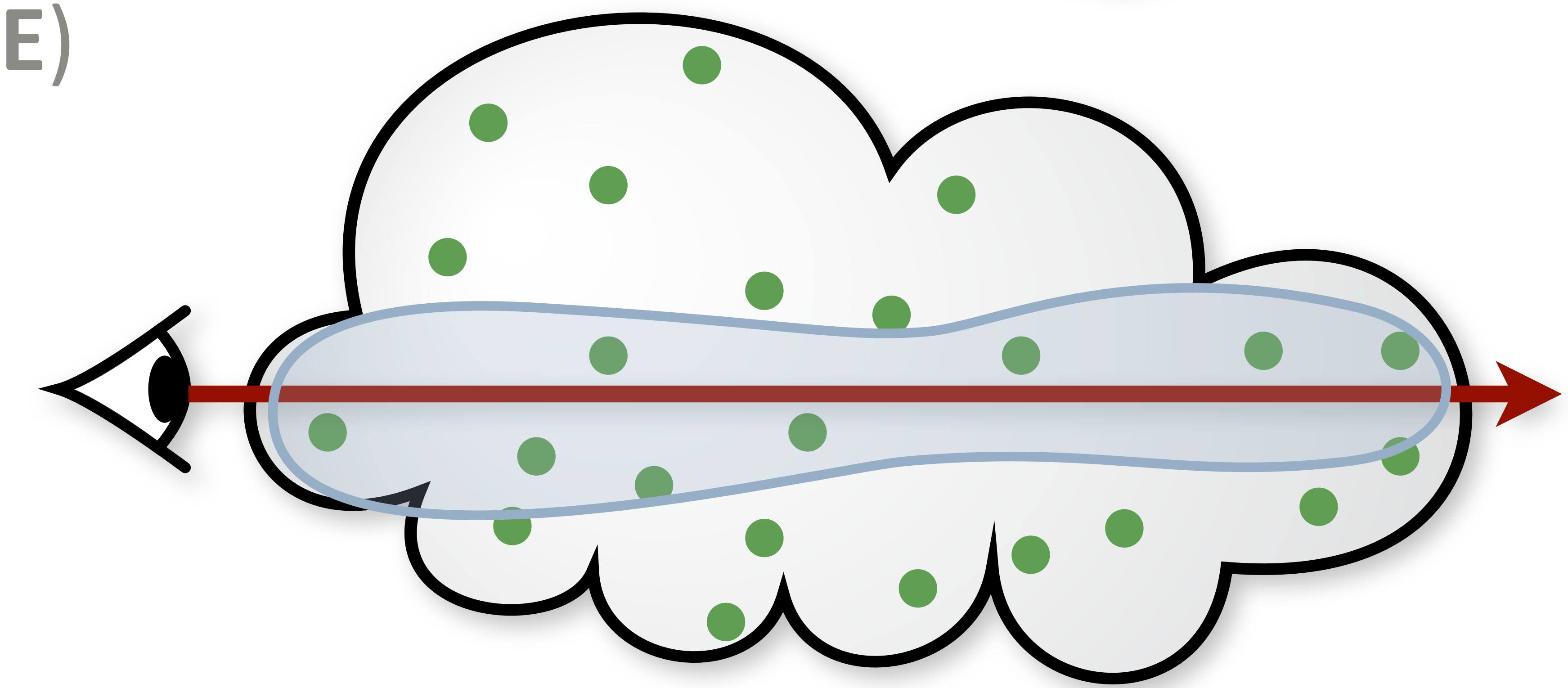
- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)



- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]

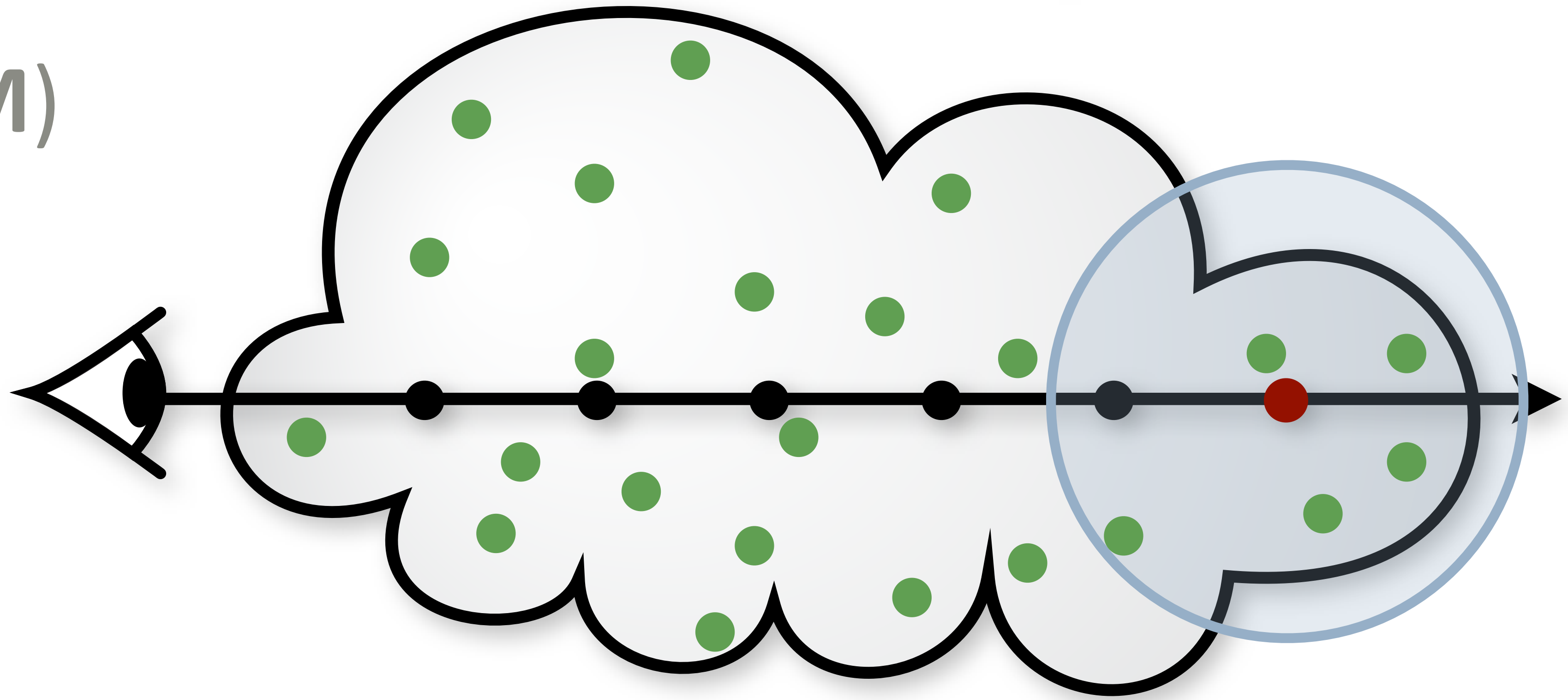
Query
Beam



So Far...

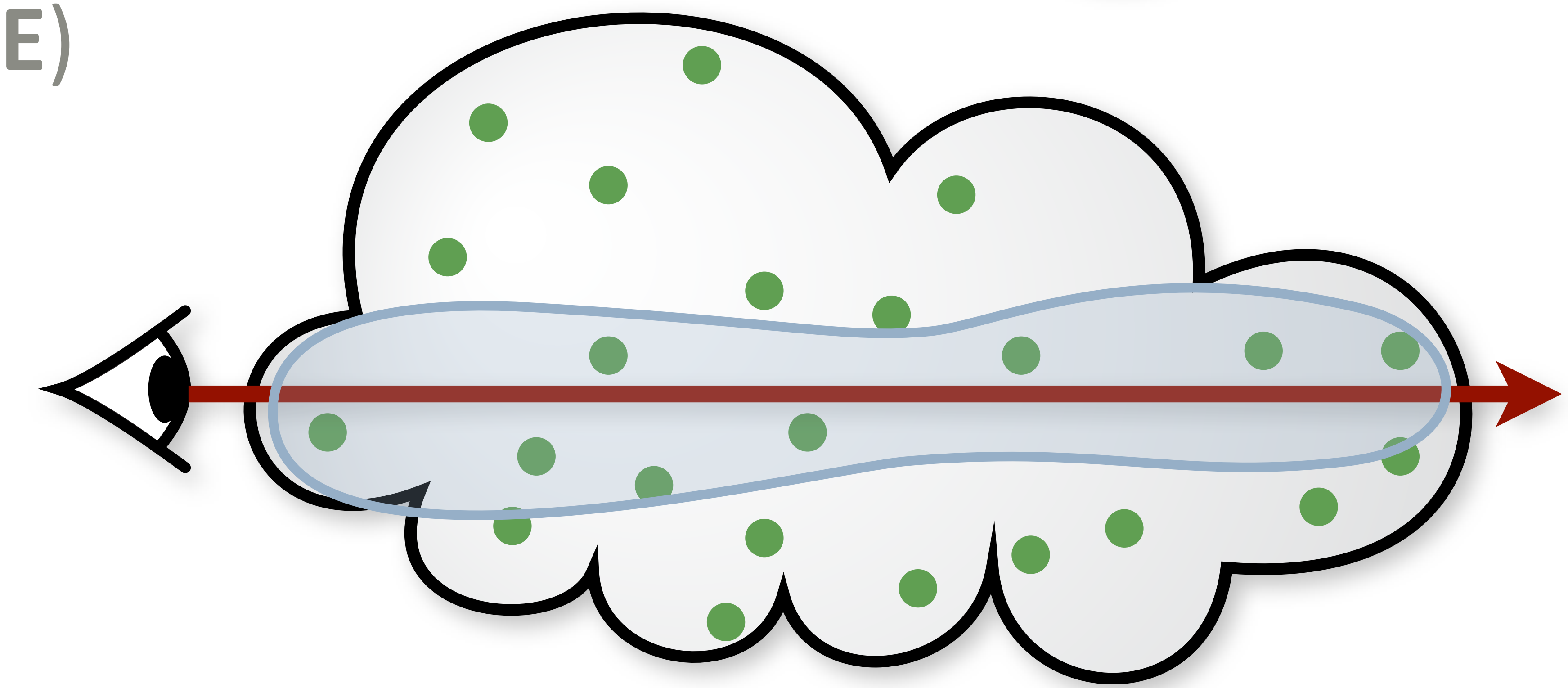
- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)



- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]

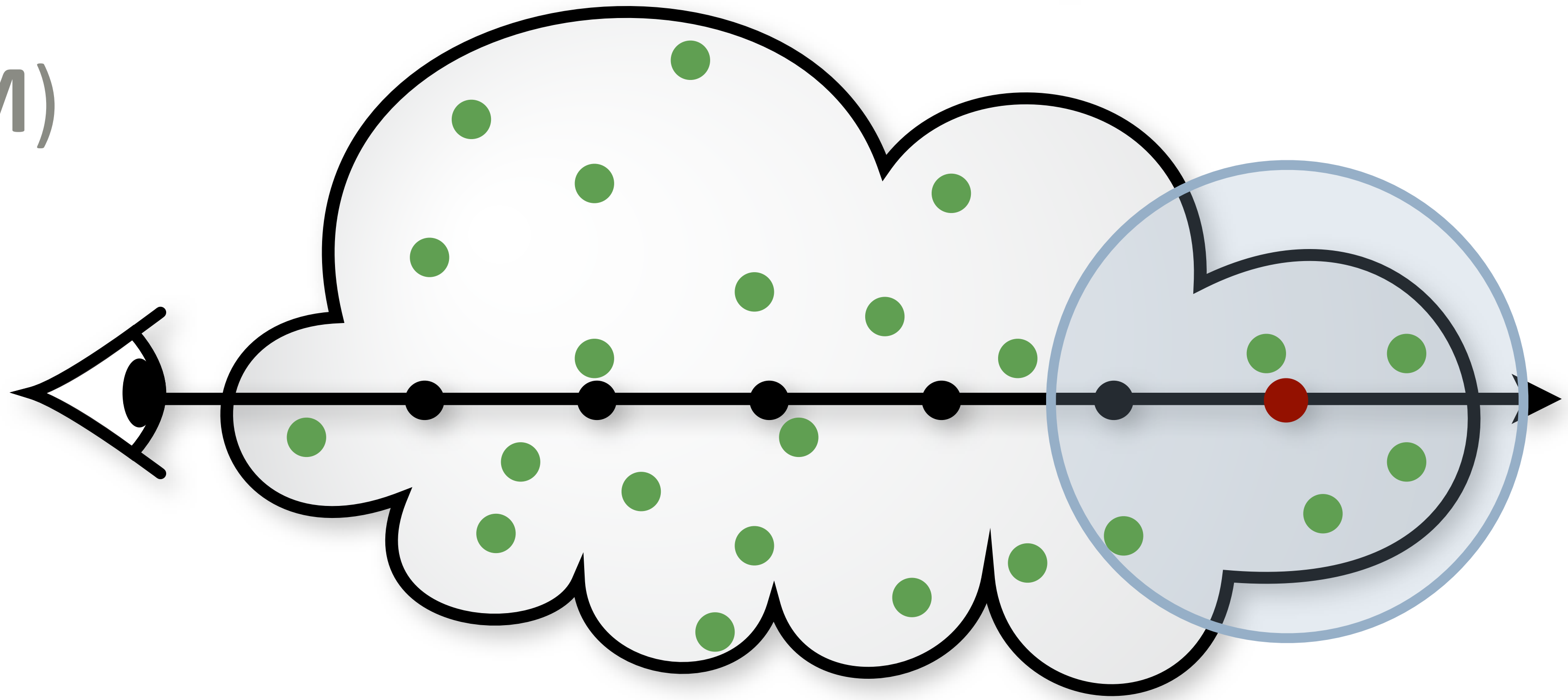
Query	x	Data
Beam	x	Point



So Far...

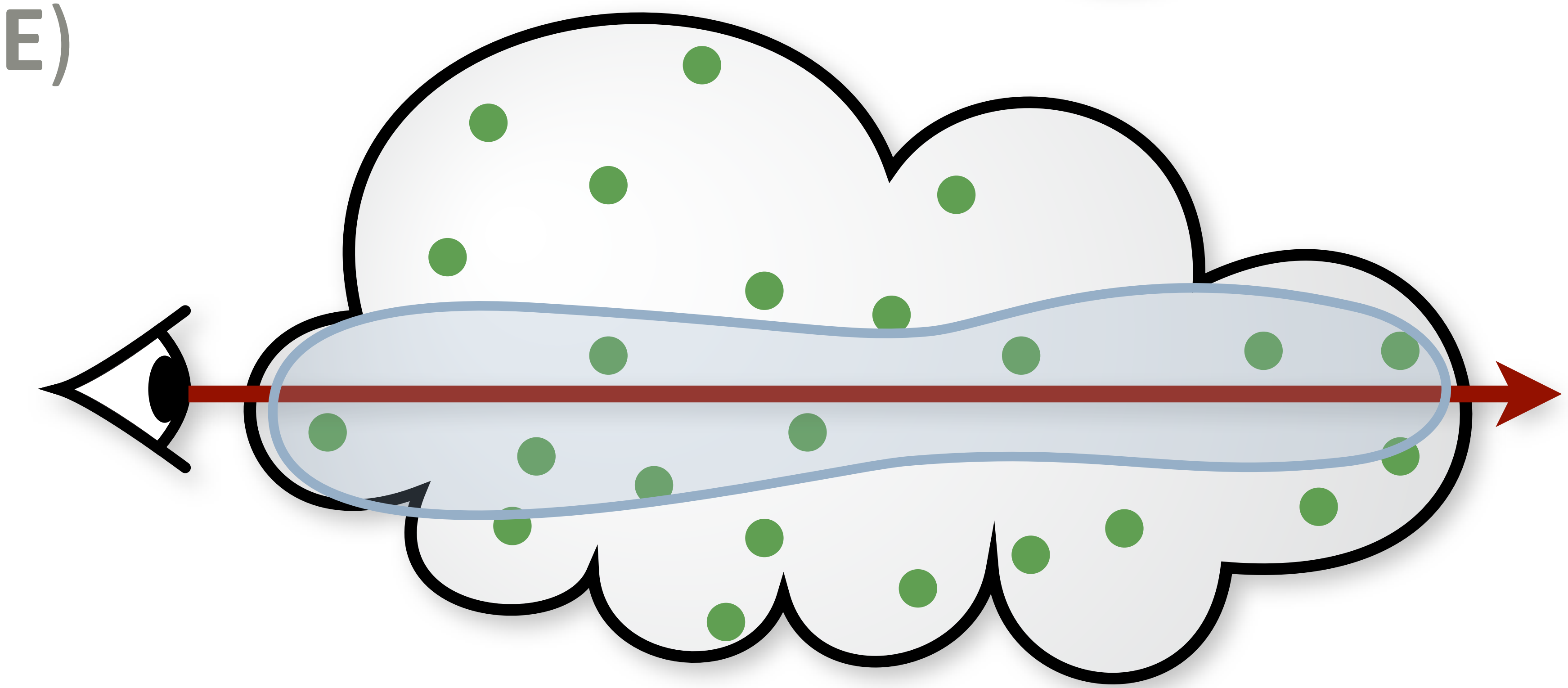
- Volumetric Photon Mapping (**VPM**)
  [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)



- The Beam Radiance Estimate (**BRE**)
  [Jarosz et al. 08]

Query	x	Data	Blur
Beam	x	Point	**(2D)**



So Far...

- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)

- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]

Query	x	Data	Blur
Beam	x	Point	**(2D)**

- Beyond Photon Points:

So Far...

- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)

- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]

Query	x	Data	Blur
Beam	x	Point	**(2D)**

- Beyond Photon Points:

Query
Point/Beam

So Far...

- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)

- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]

Query	x	Data	Blur
Beam	x	Point	**(2D)**

- Beyond Photon Points:

Query	Data
Point/Beam	**Point/Beam**

So Far...

- Volumetric Photon Mapping (**VPM**) [Jensen & Christensen 98]

Query	x	Data	Blur
Point	x	Point	(3D)

- The Beam Radiance Estimate (**BRE**) [Jarosz et al. 08]

Query	x	Data	Blur
Beam	x	Point	**(2D)**

- Beyond Photon Points:

Query	Data	Blur
Point/Beam	**Point/Beam**	**1D/2D/3D**

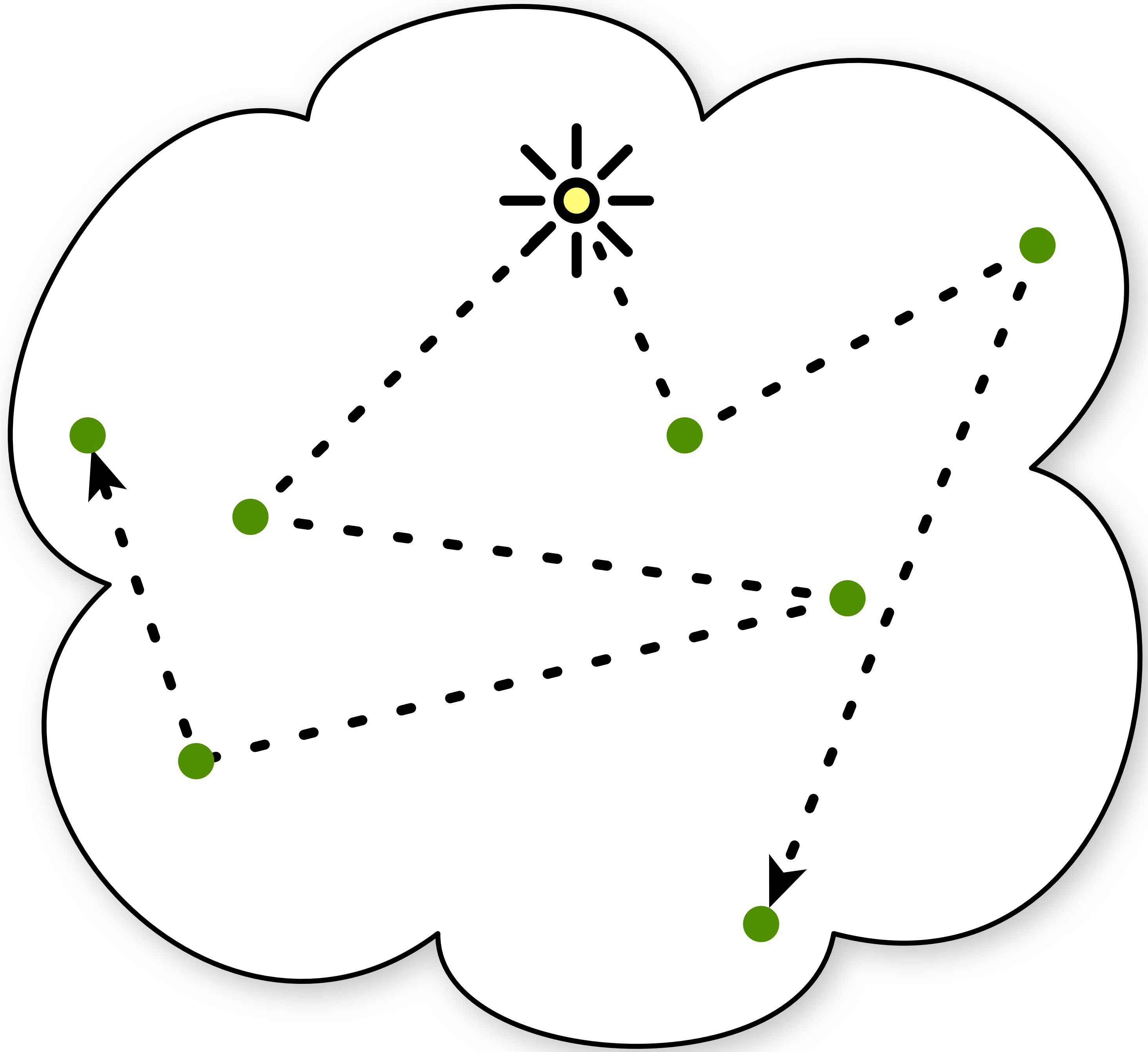
Density Estimator Options

Query	x	Data	Blur
Point	x	Point	(3D)
Beam	x	Point	(2D)

Density Estimator Options

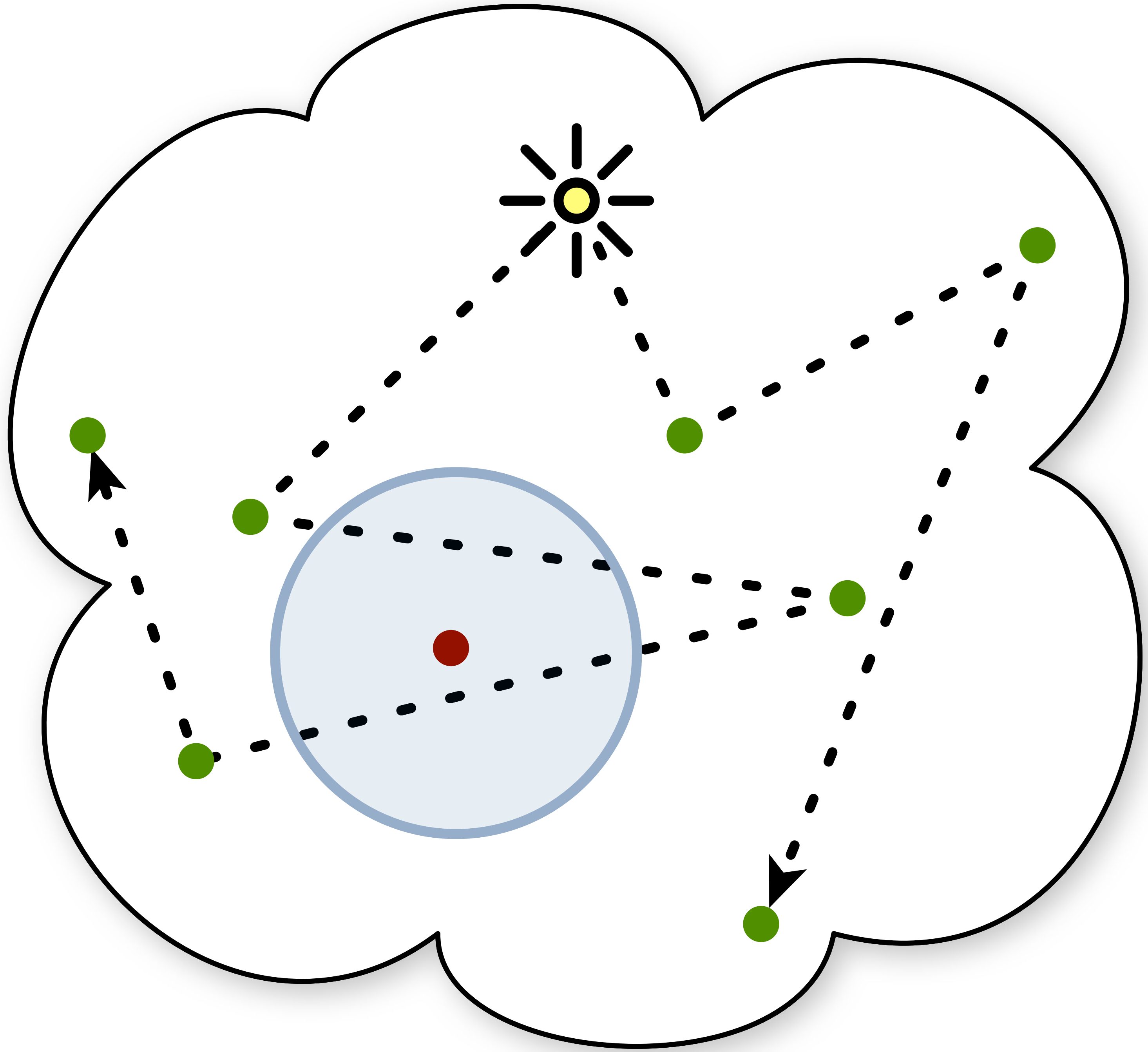
Query	x	Data	Blur
Point	x	Point	(3D)
Beam	x	Point	(2D)
Beam	**x**	**Point**	**(3D)**
Point	**x**	**Beam**	**(3D)**
Point	**x**	**Beam**	**(2D)**
Beam	**x**	**Beam**	**(3D)**
Beam	**x**	**Beam**	**$(2D)_1$**
Beam	**x**	**Beam**	**$(2D)_2$**
Beam	**x**	**Beam**	**(1D)**

Volumetric Photon Mapping



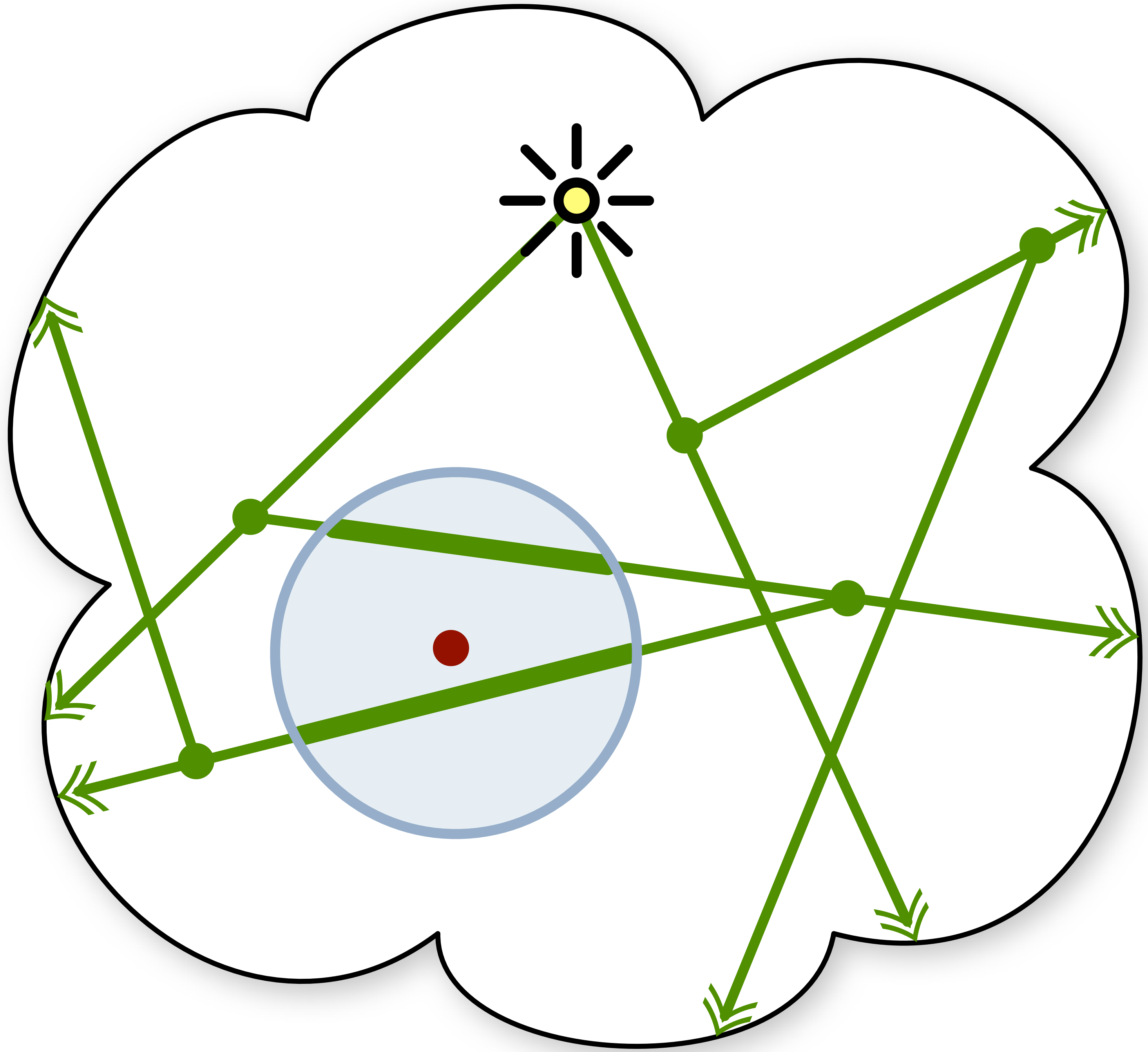
Photon Points

Volumetric Photon Mapping

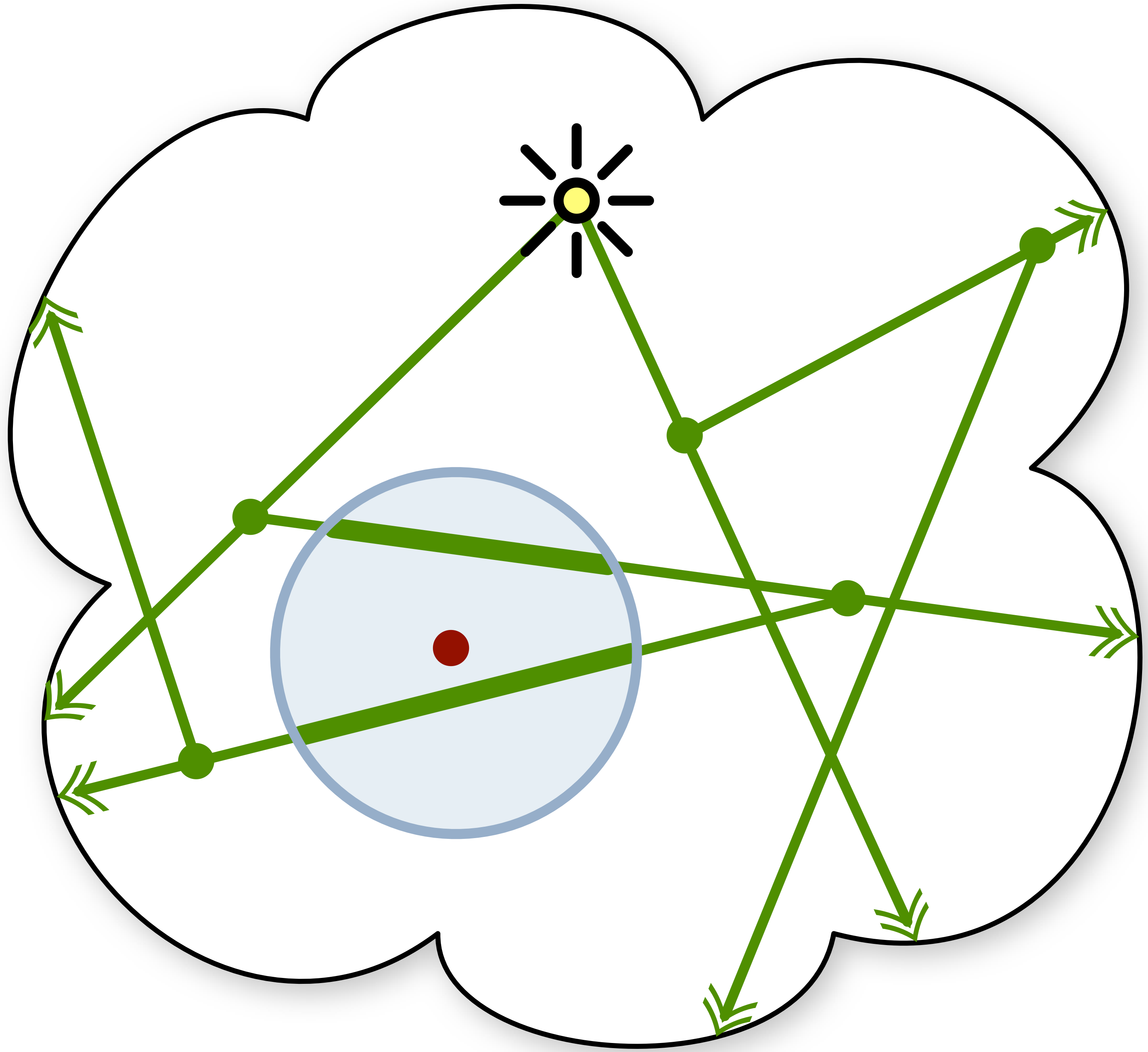


Photon Points

Volumetric Photon Mapping



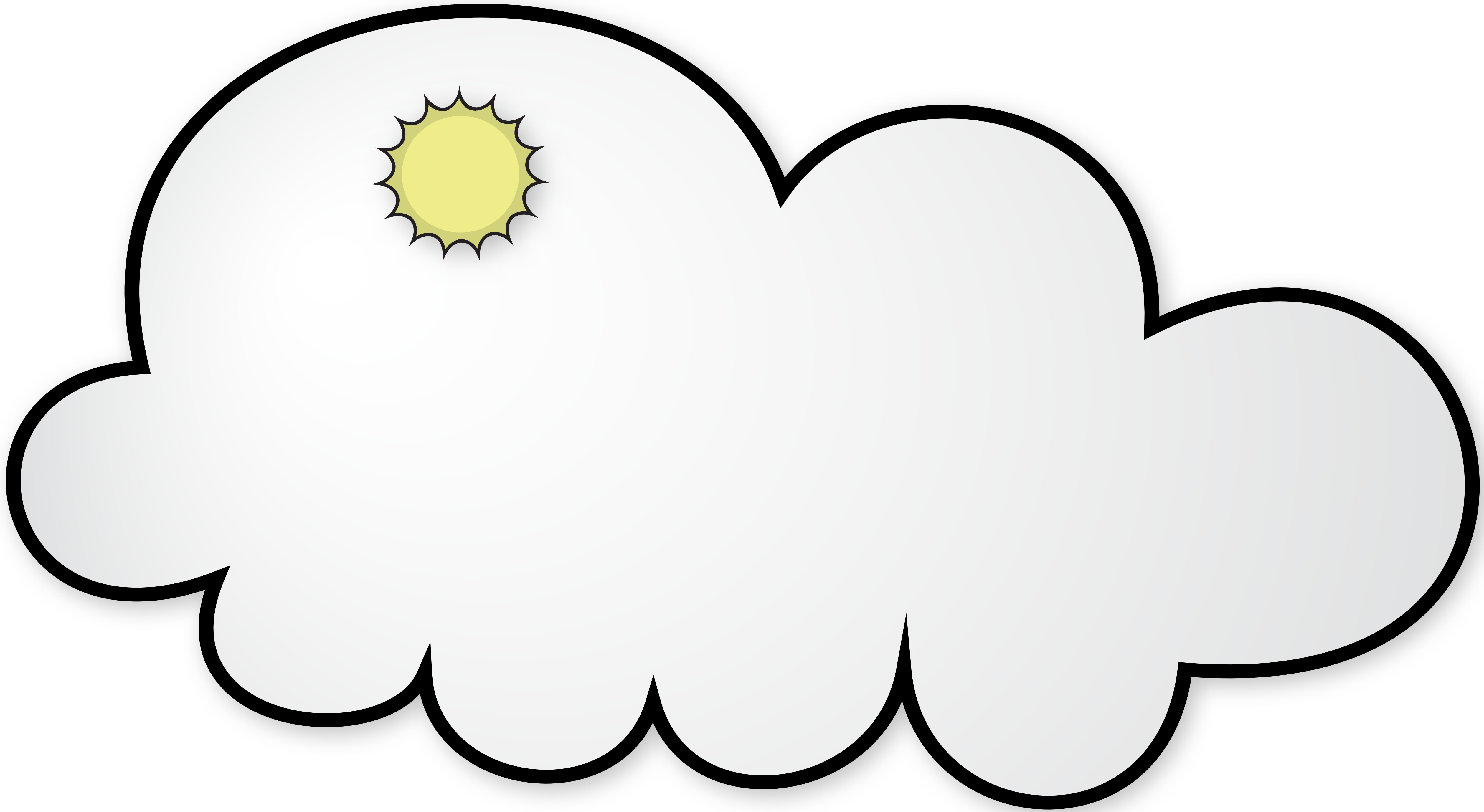
Volumetric Photon Mapping



Photon Beams

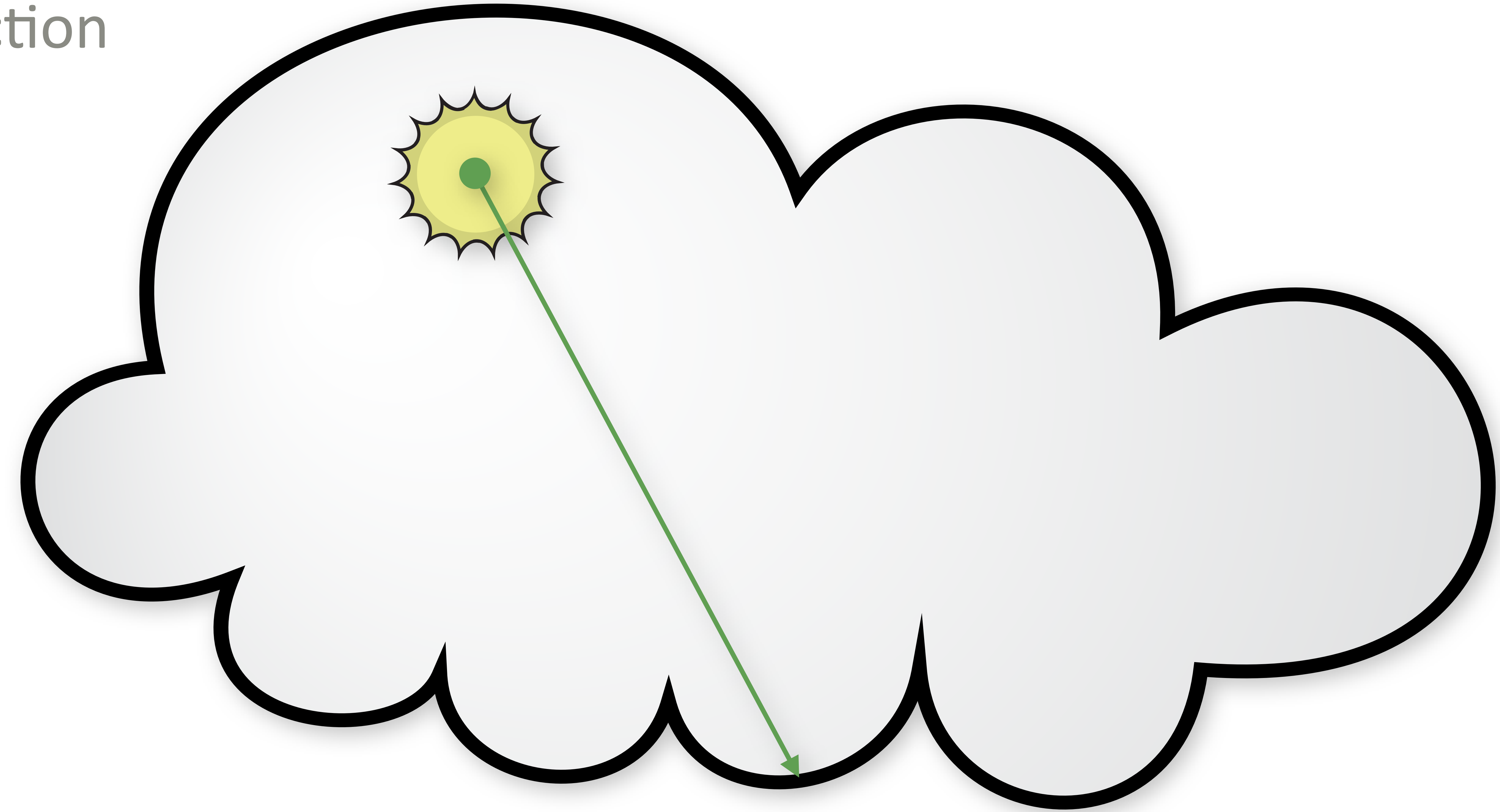
Photon Beams

Traditional Photon Tracing



Traditional Photon Tracing

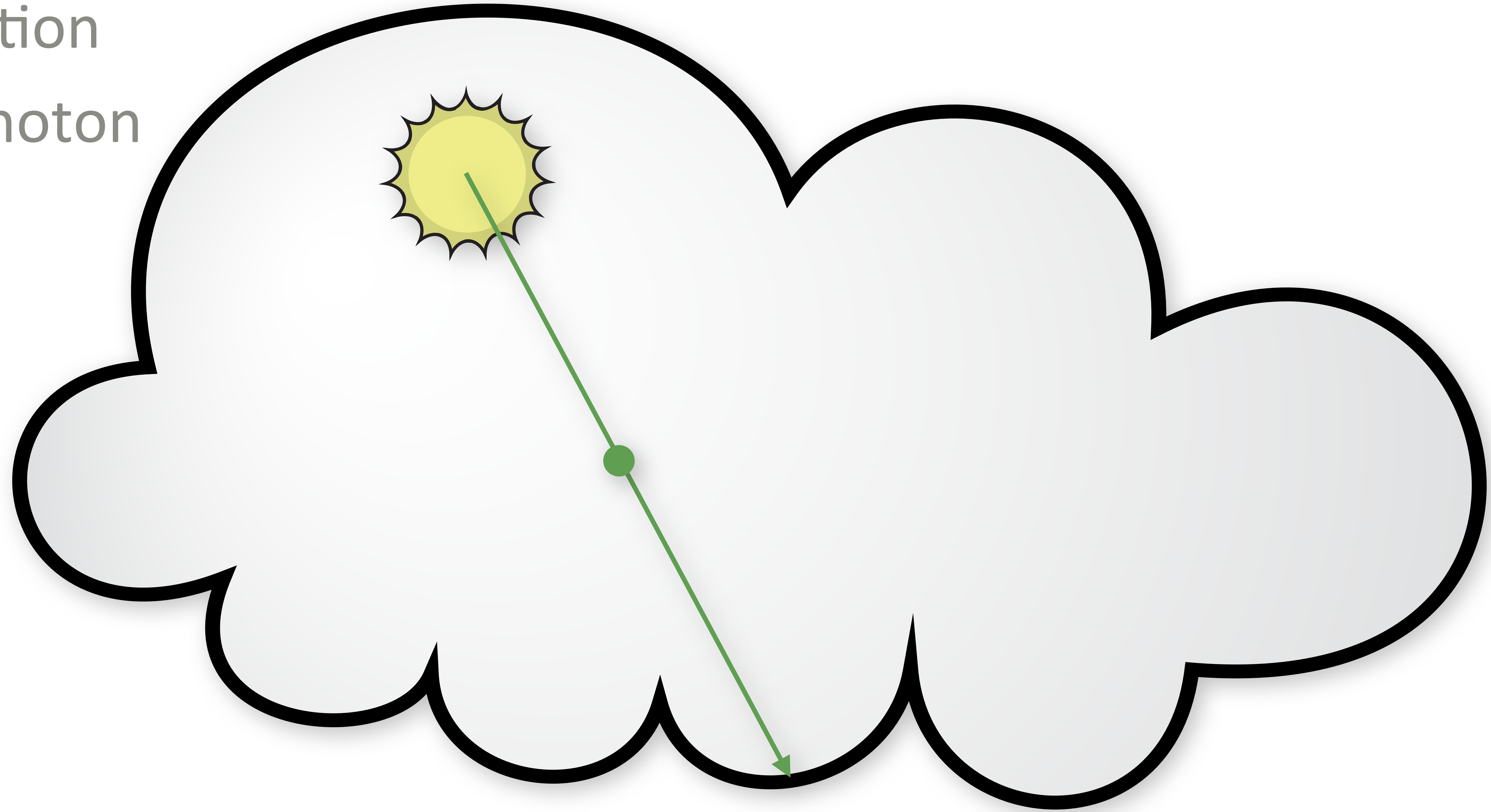
1) choose direction



Traditional Photon Tracing

1) choose direction

2) propagate photon

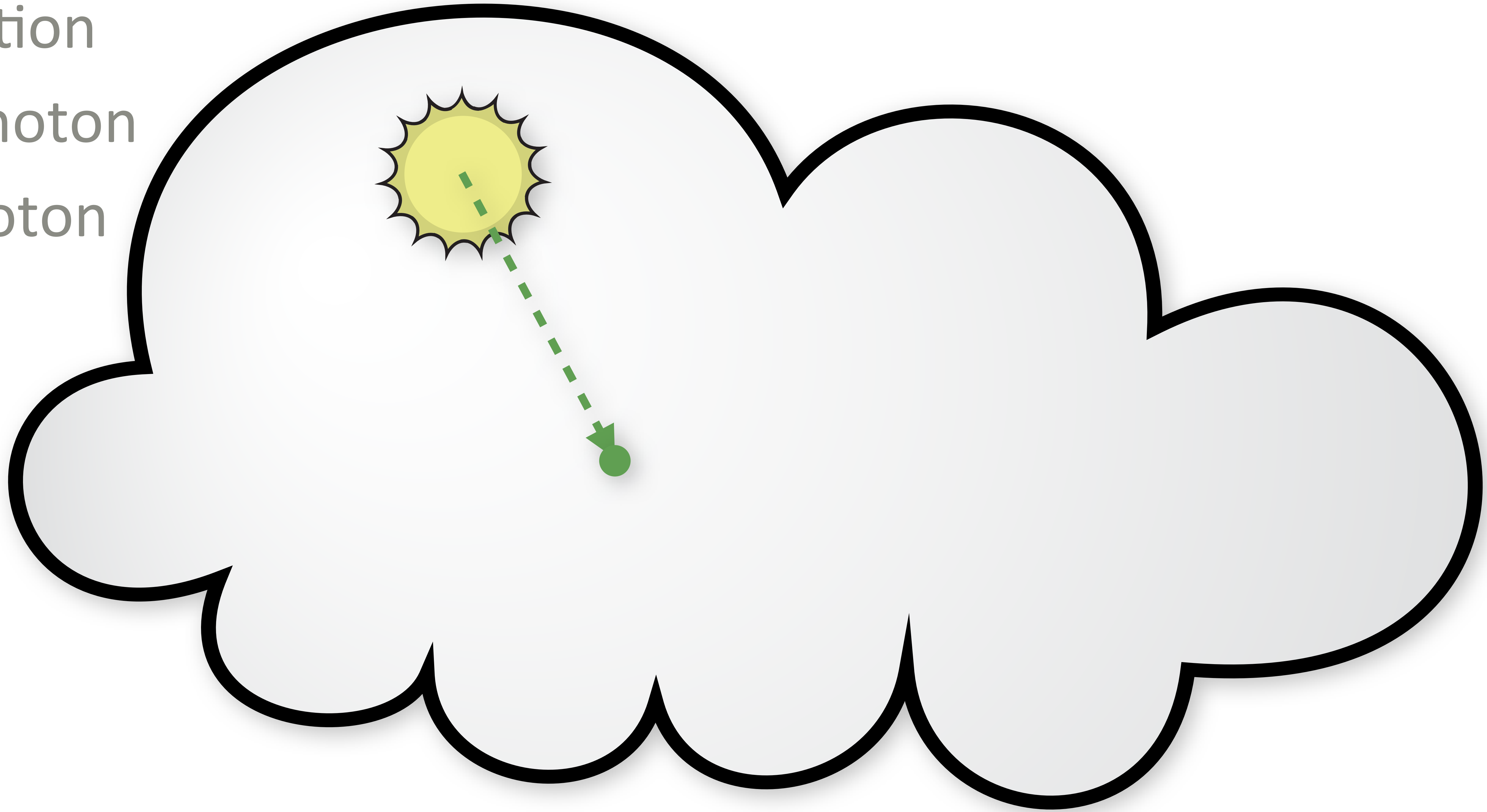


Traditional Photon Tracing

1) choose direction

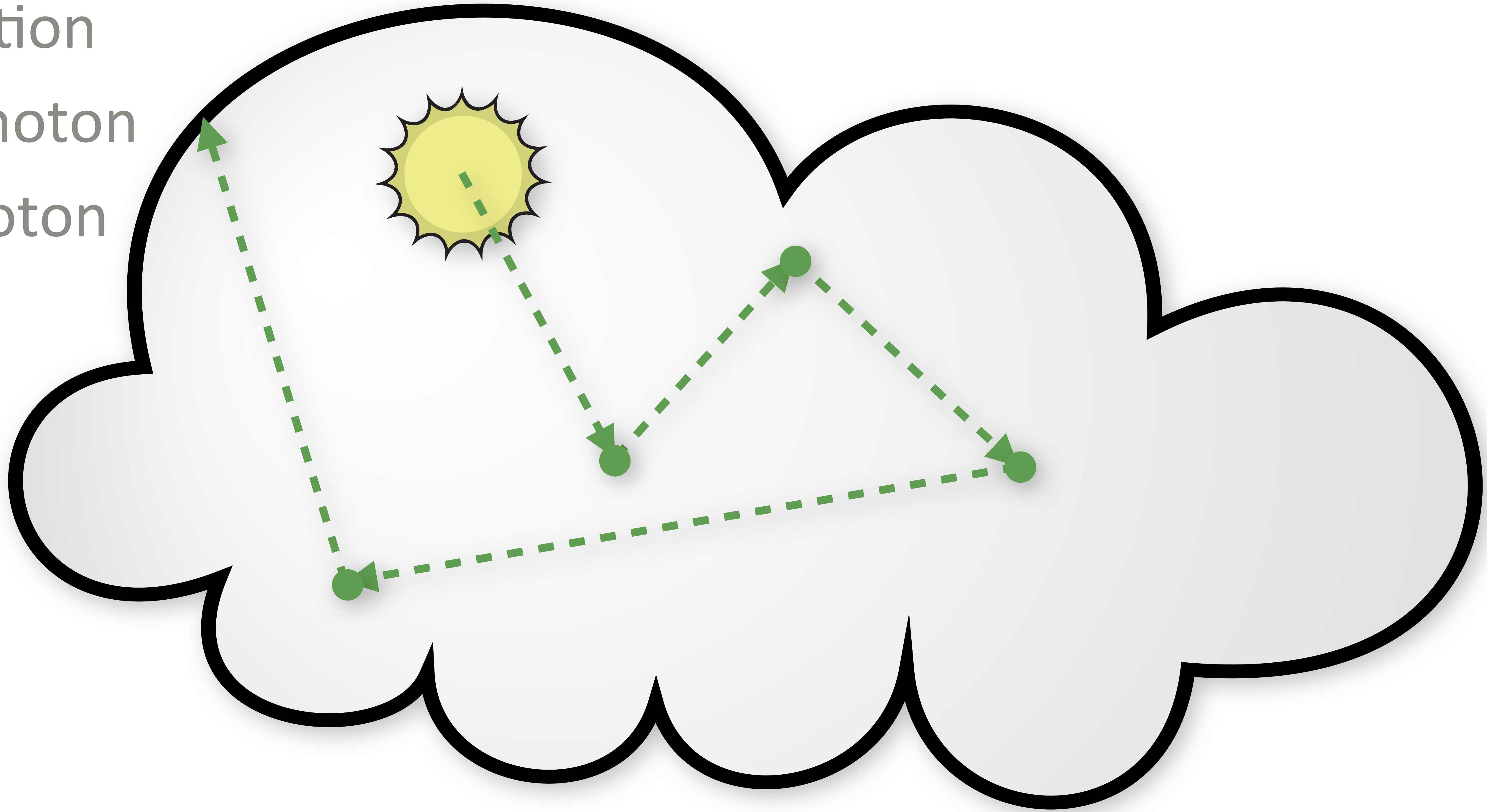
2) propagate photon

3) deposit a photon



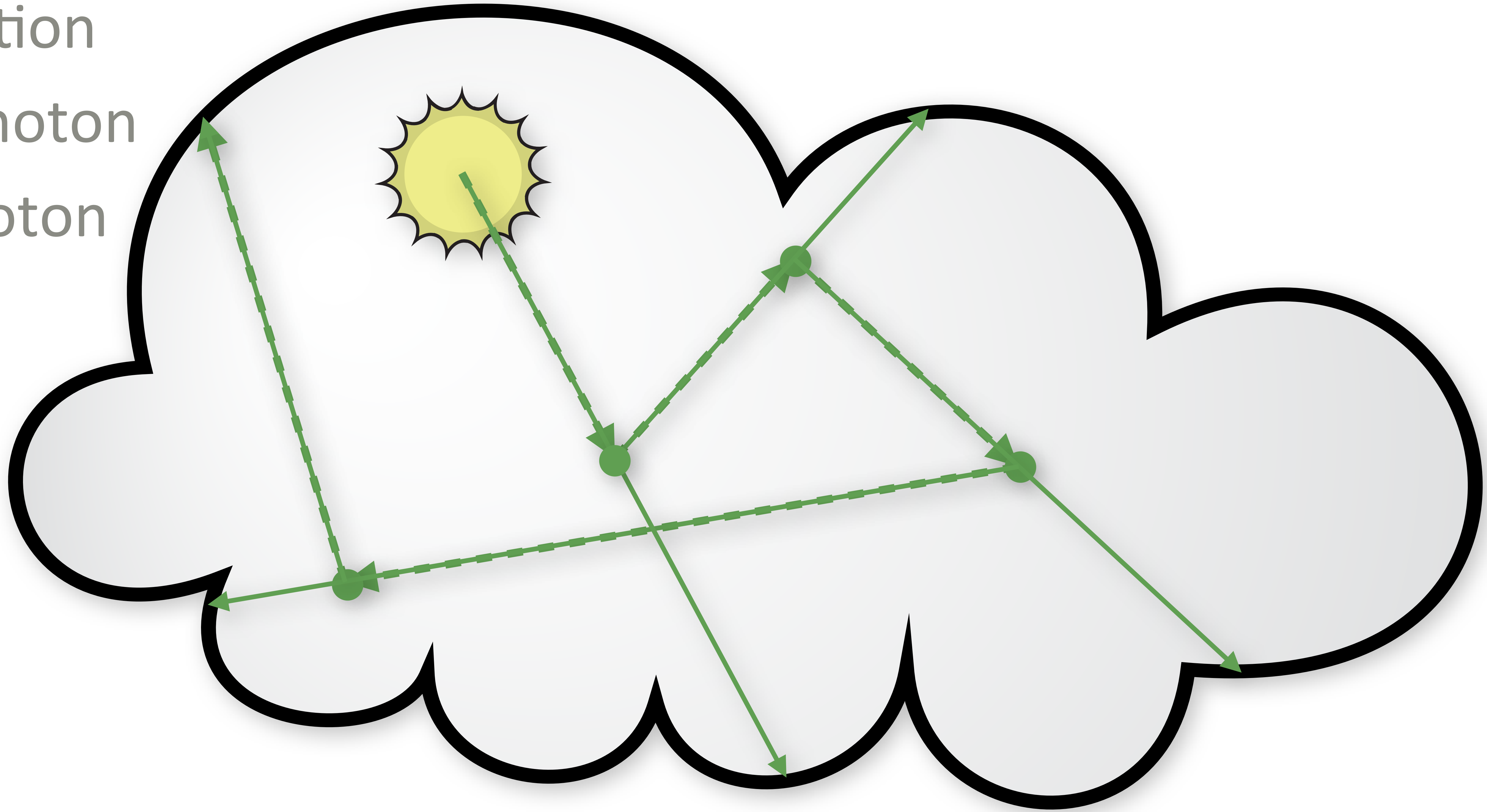
Traditional Photon Tracing

1) choose direction
2) propagate photon
3) deposit a photon
4) repeat



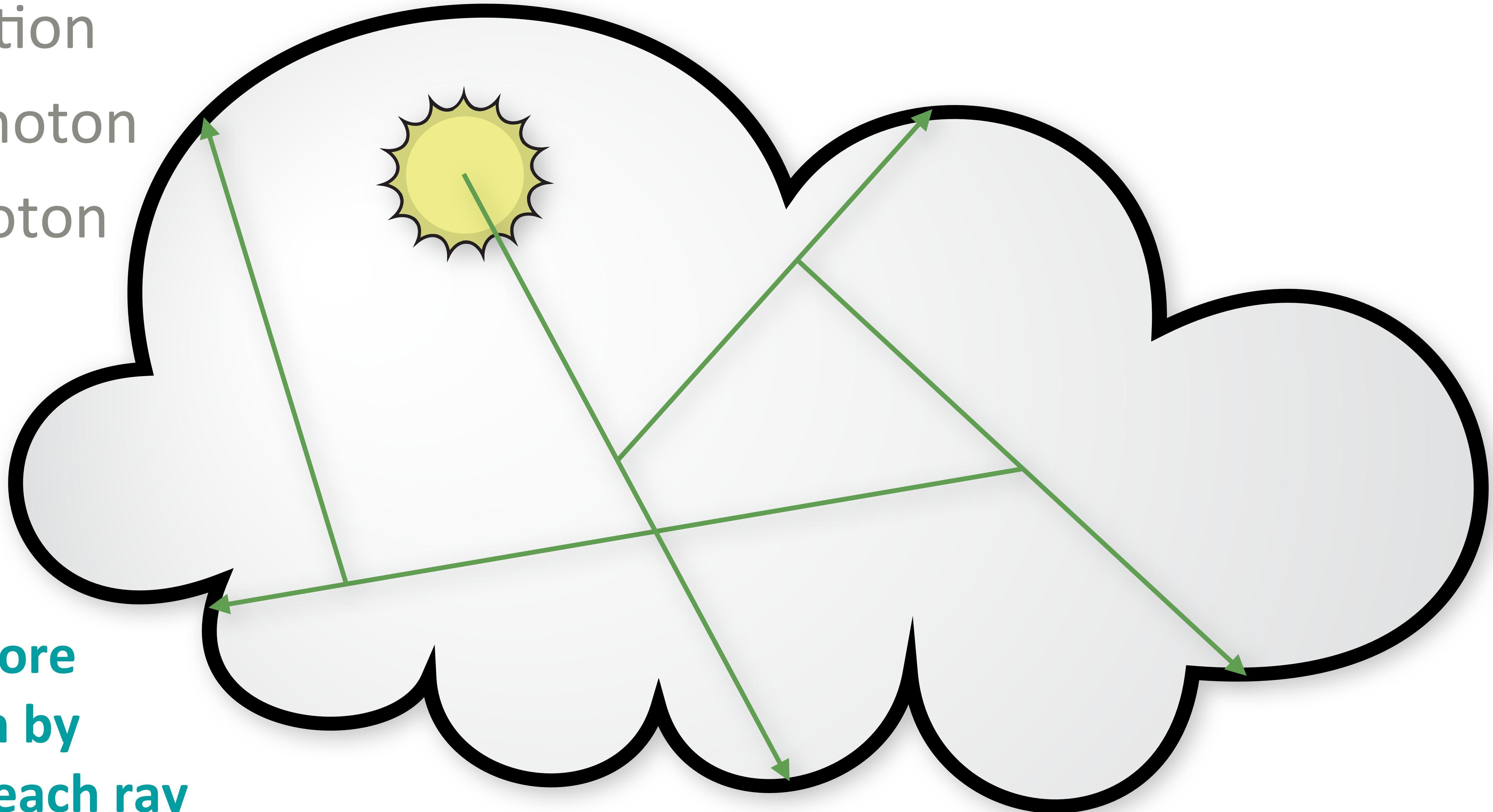
Traditional Photon Tracing

1) choose direction
2) propagate photon
3) deposit a photon
4) repeat



"Photon Marching"

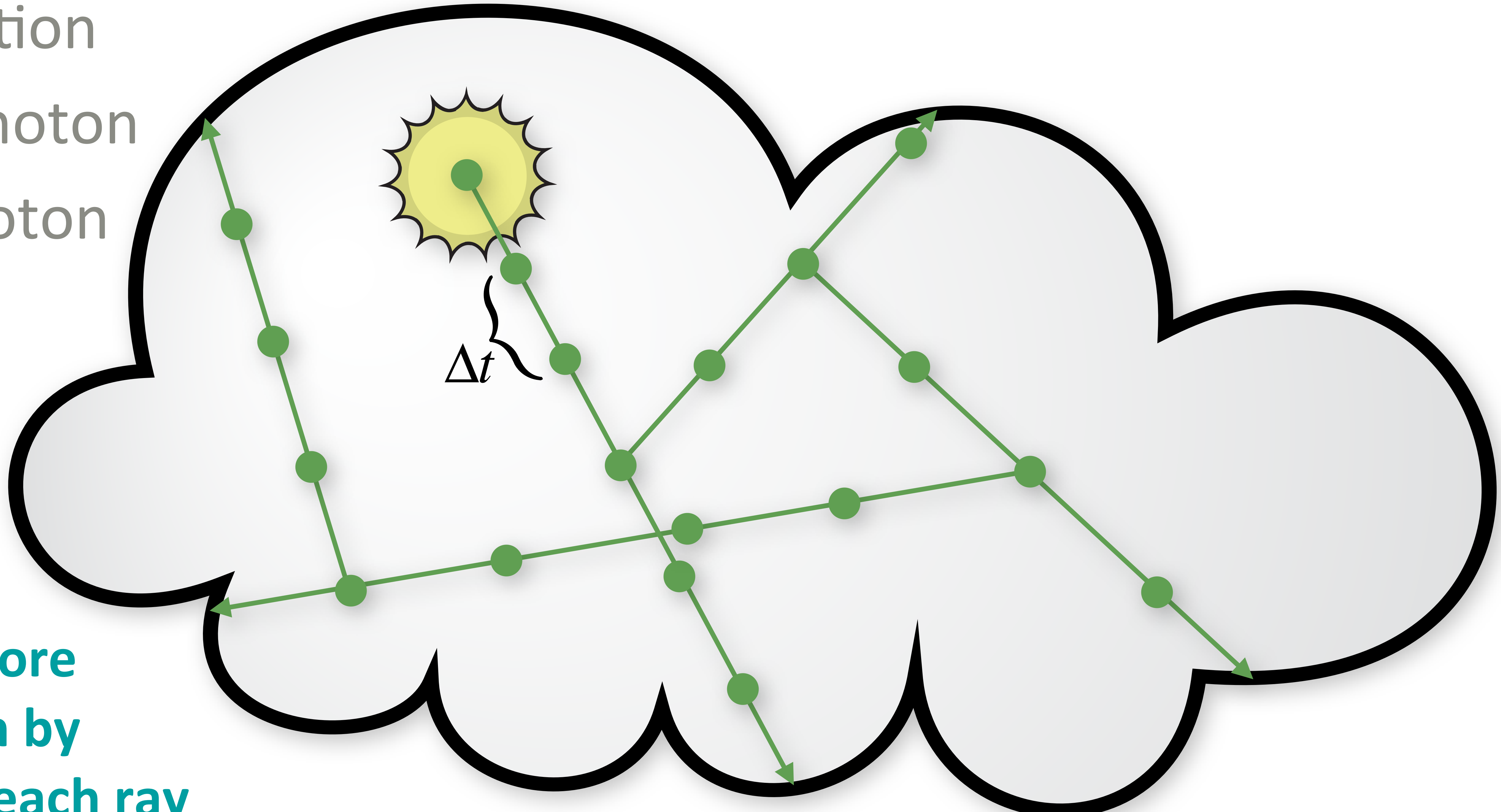
1) choose direction
2) propagate photon
3) deposit a photon
4) repeat



Could deposit more than one photon by marching along each ray

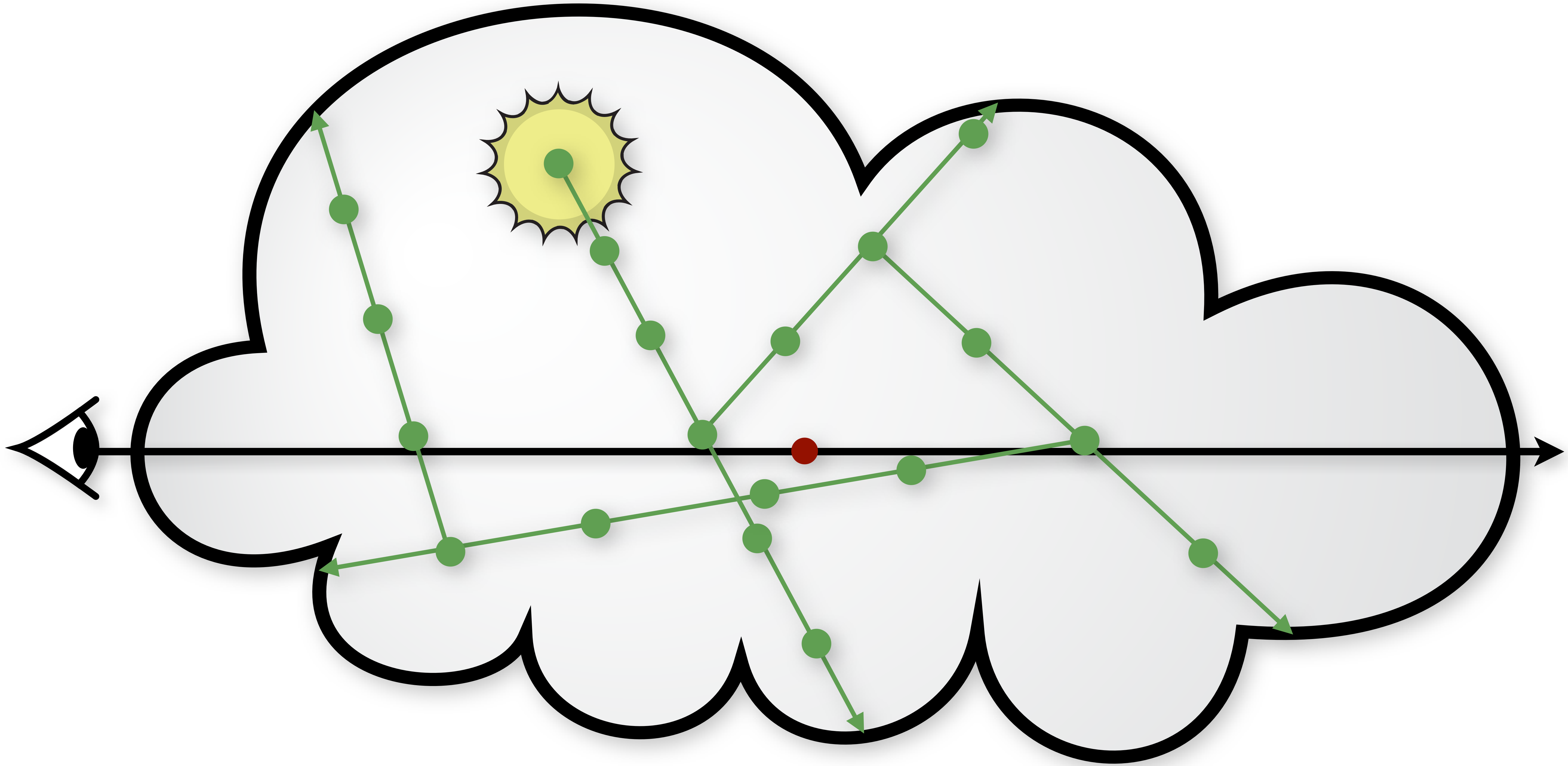
“Photon Marching”

1) choose direction
2) propagate photon
3) deposit a photon
4) repeat



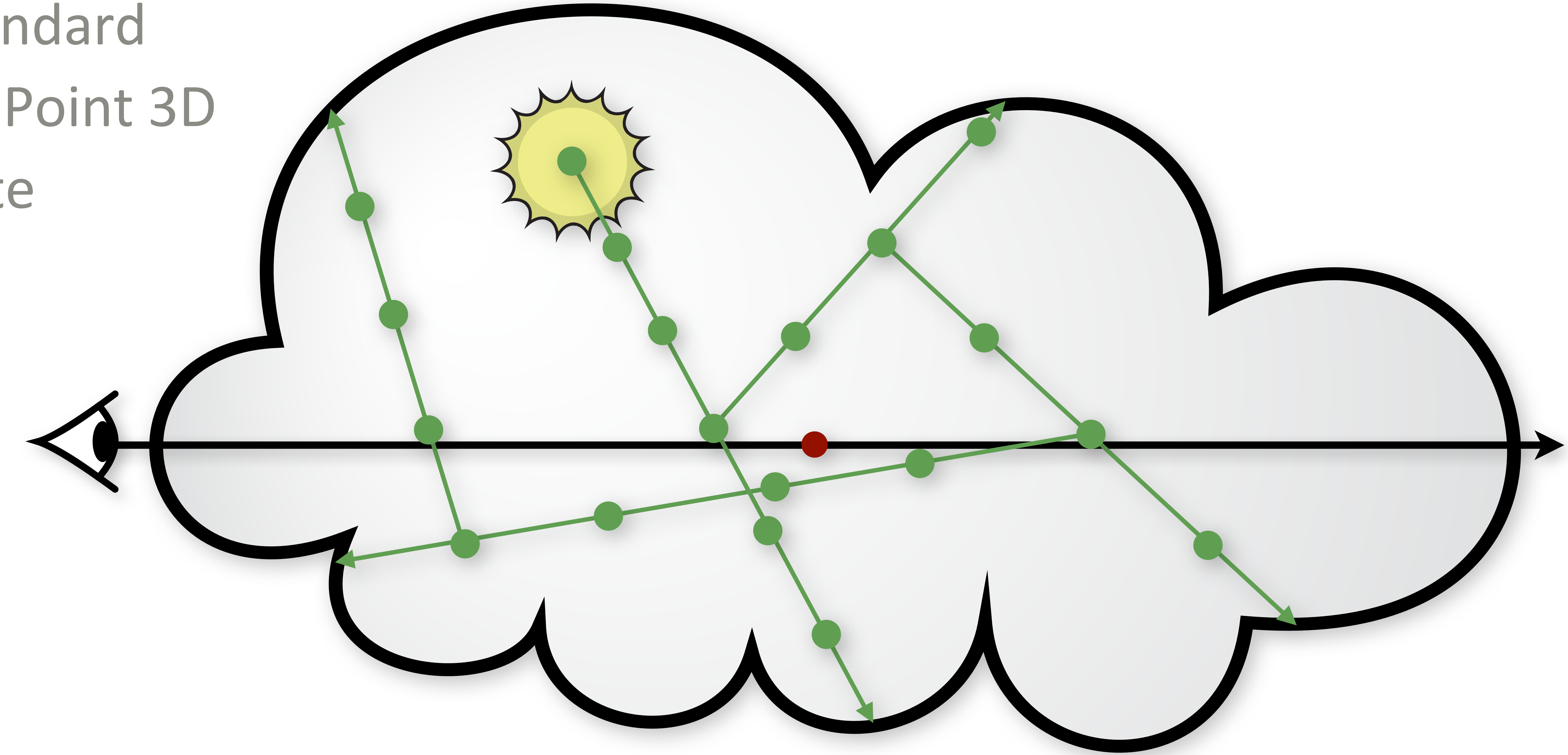
Could deposit more than one photon by marching along each ray

Radiance Estimation using “Discrete Photon Beams”



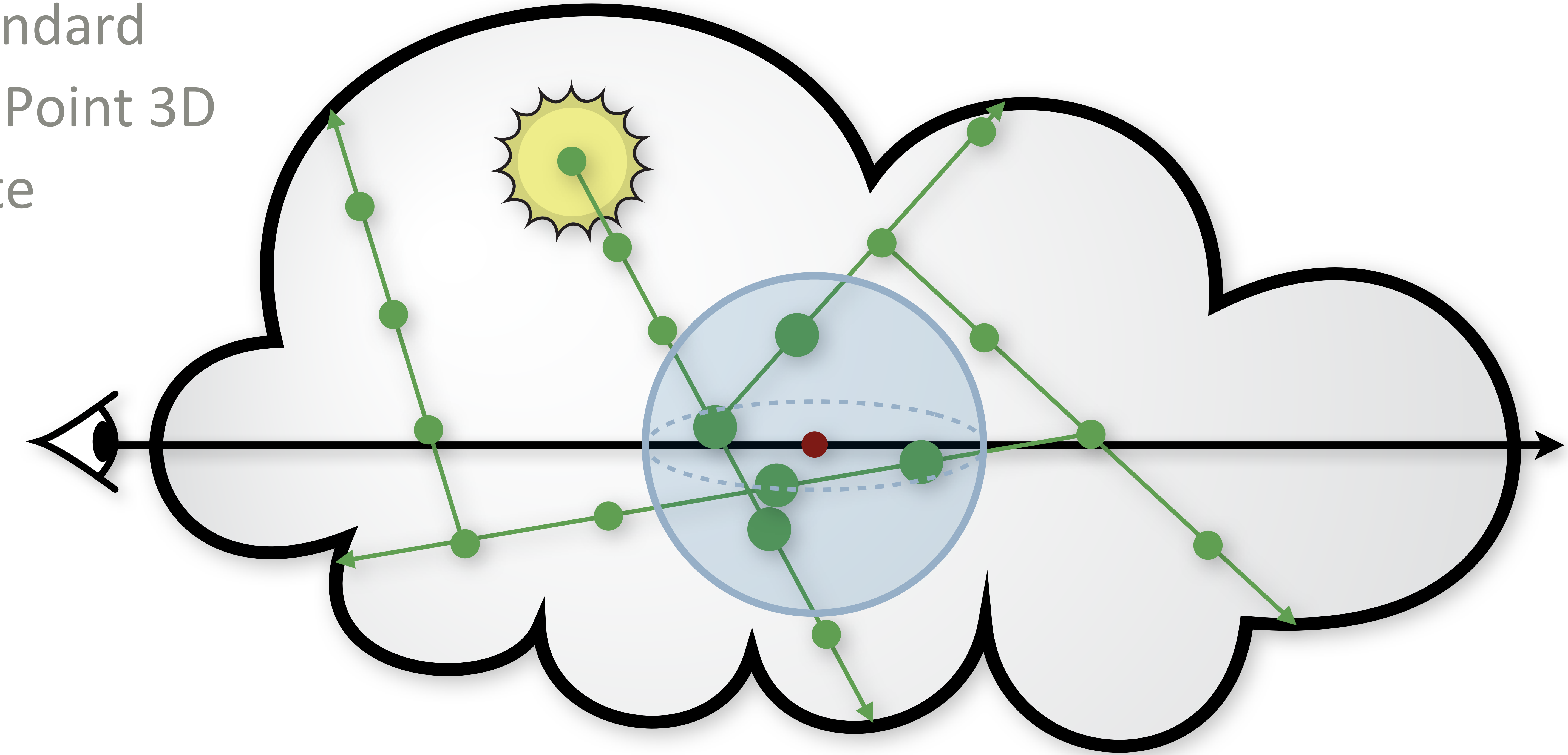
Radiance Estimation using "Discrete Photon Beams"

- Use standard Point x Point 3D estimate



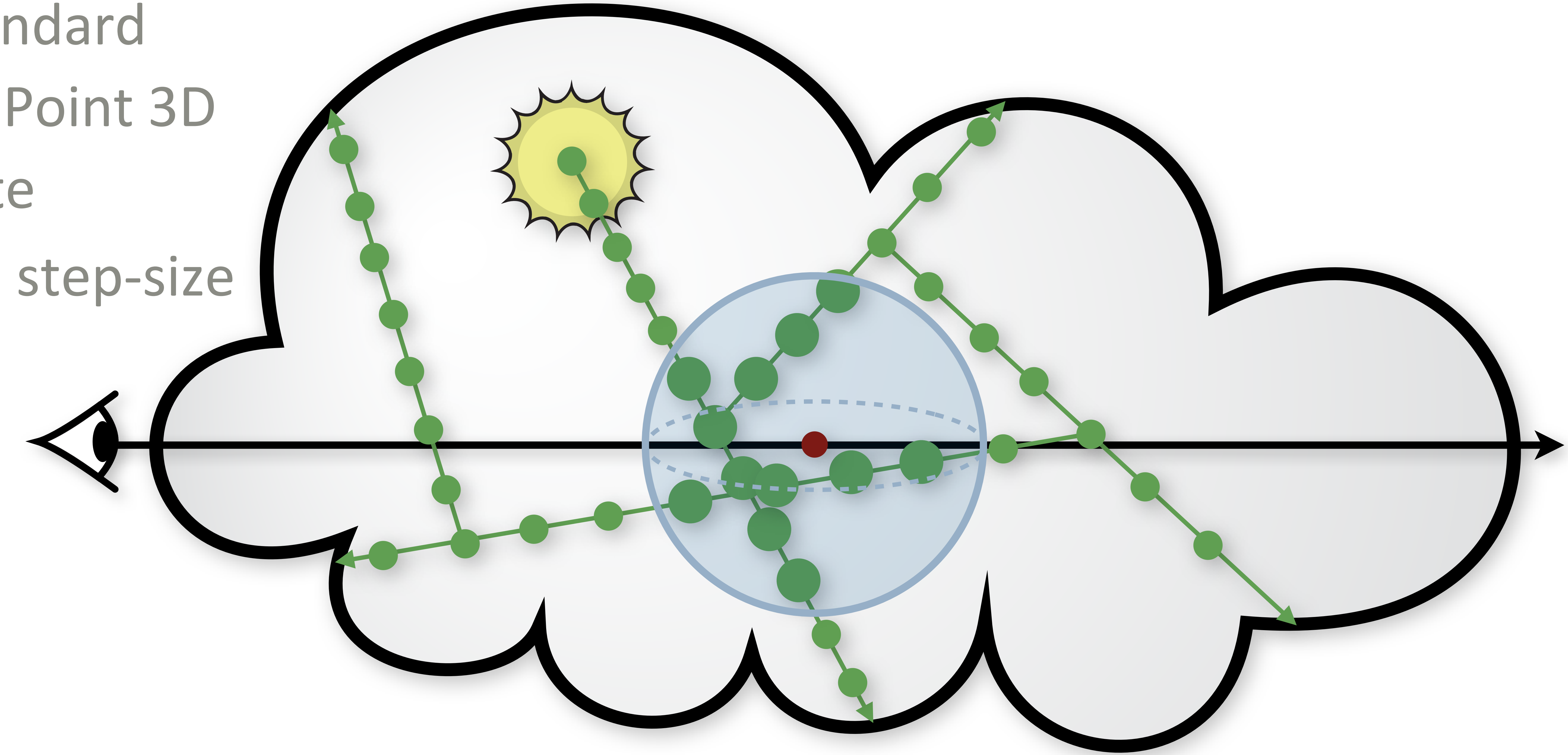
Radiance Estimation using "Discrete Photon Beams"

- Use standard Point x Point 3D estimate



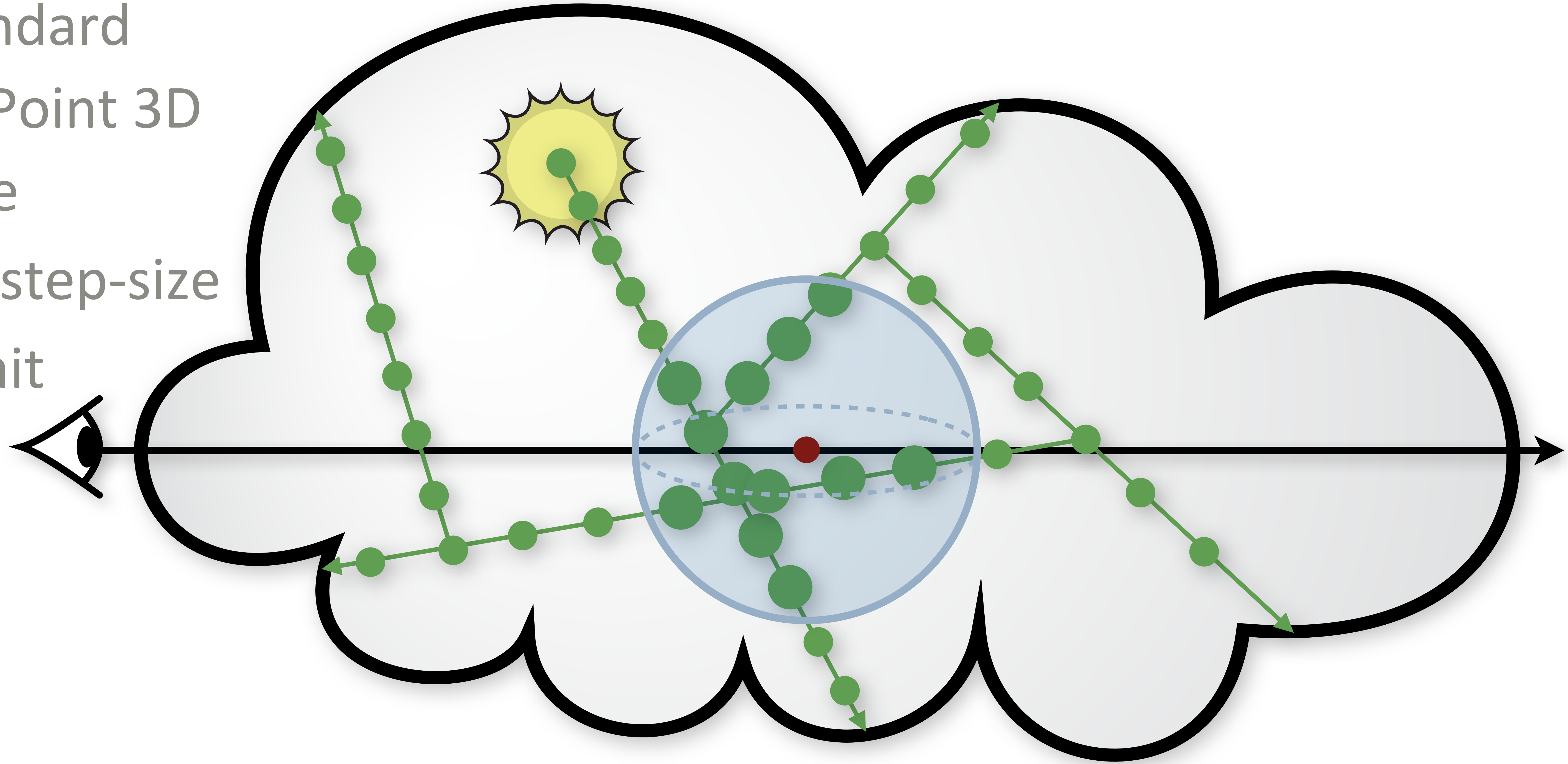
Radiance Estimation using "Discrete Photon Beams"

- Use standard Point x Point 3D estimate
- Reduce step-size



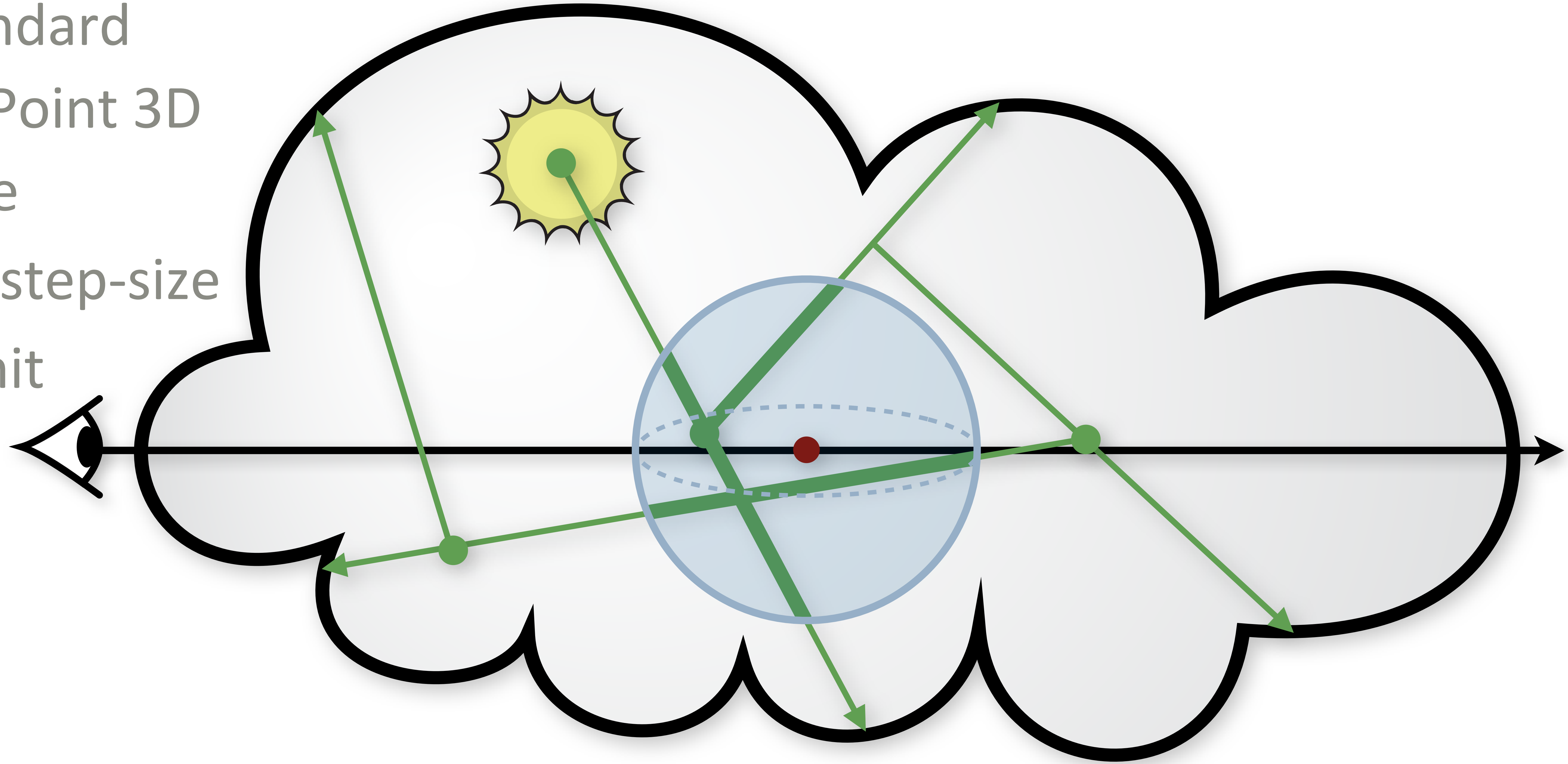
Radiance Estimation using "Discrete Photon Beams"

- Use standard Point x Point 3D estimate
- Reduce step-size
- Take limit

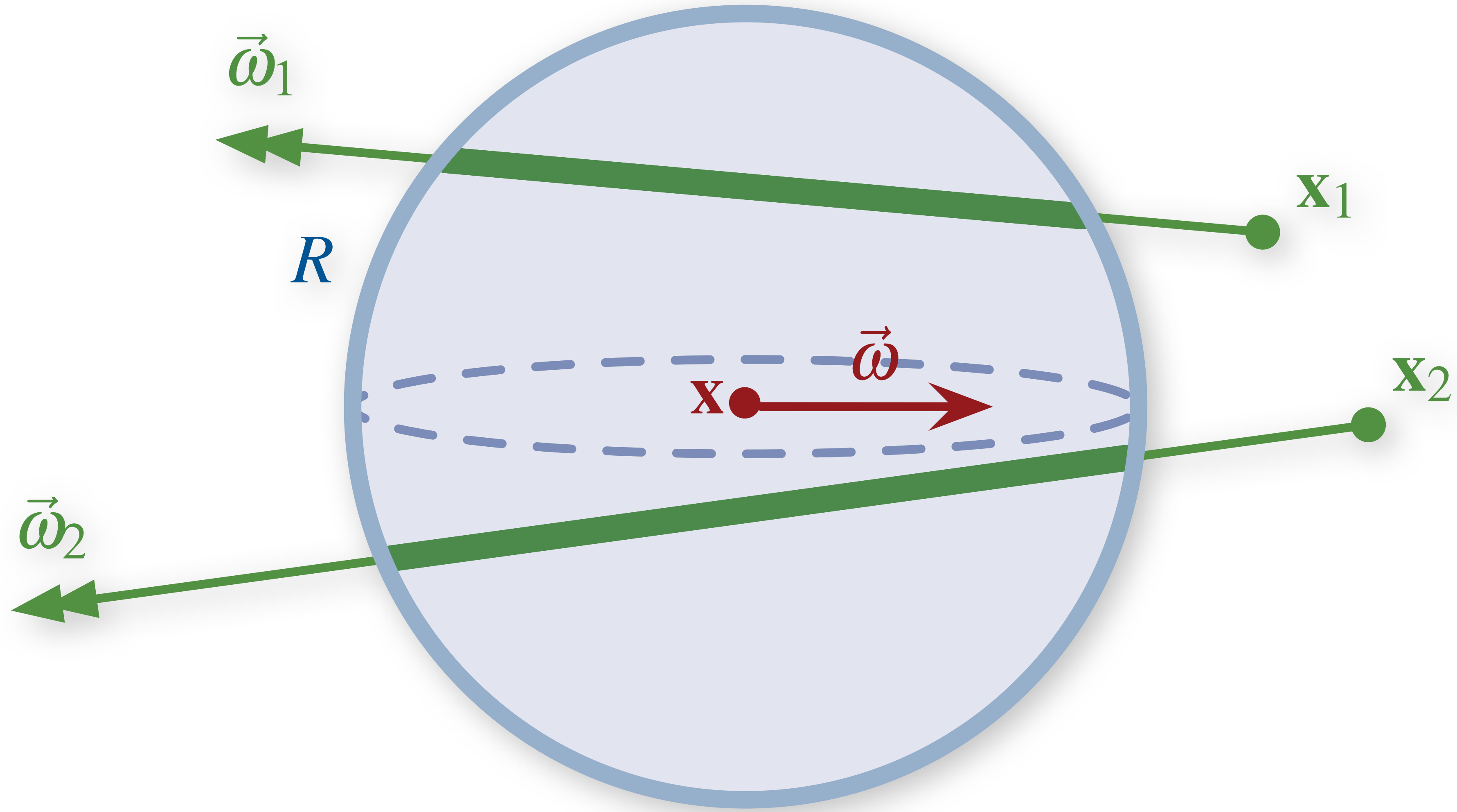


Radiance Estimation using "Discrete Photon Beams"

- Use standard Point x Point 3D estimate
- Reduce step-size
- Take limit

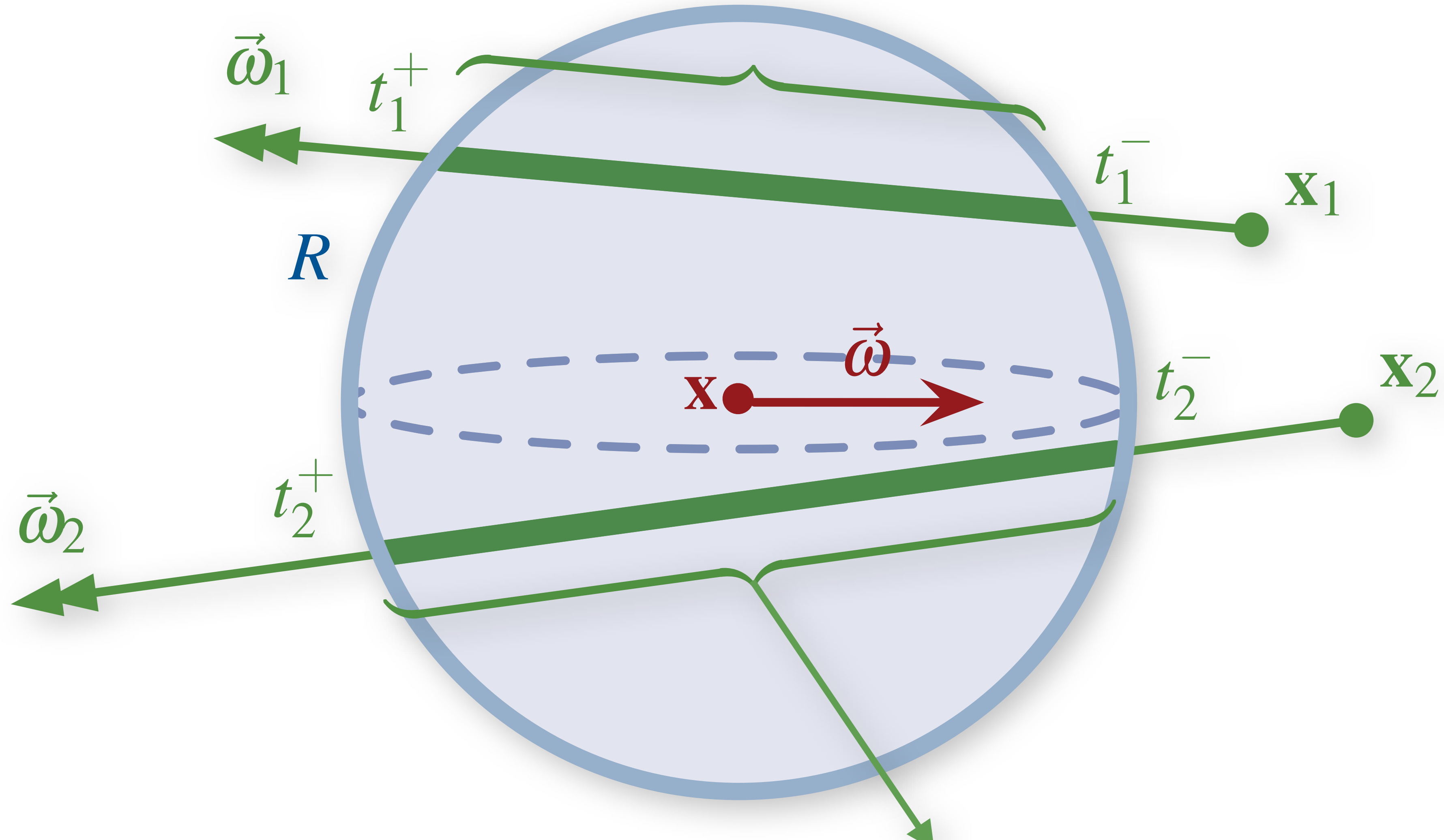


Point Query × Beam Data (3D blur)



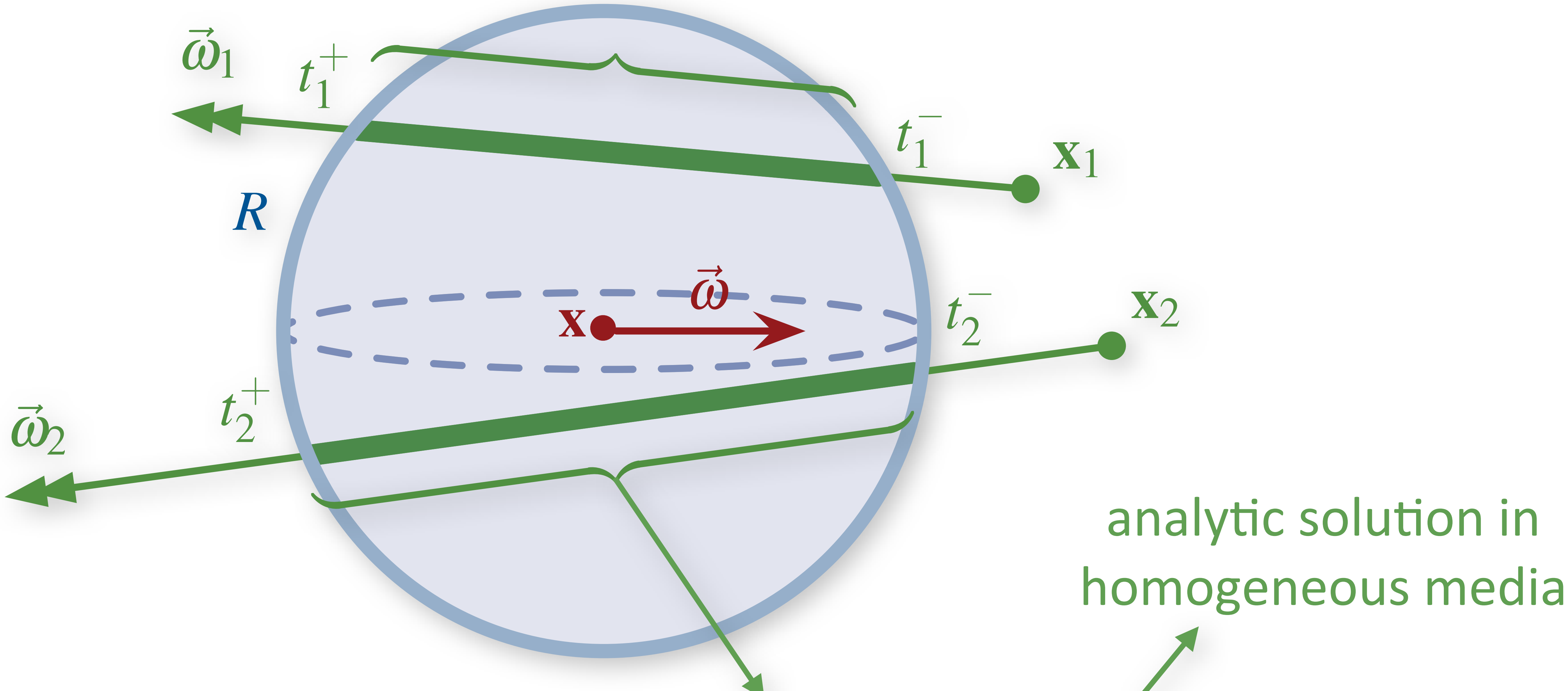
$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i)\,\Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t}\, dt$$

Point Query × Beam Data (3D blur)



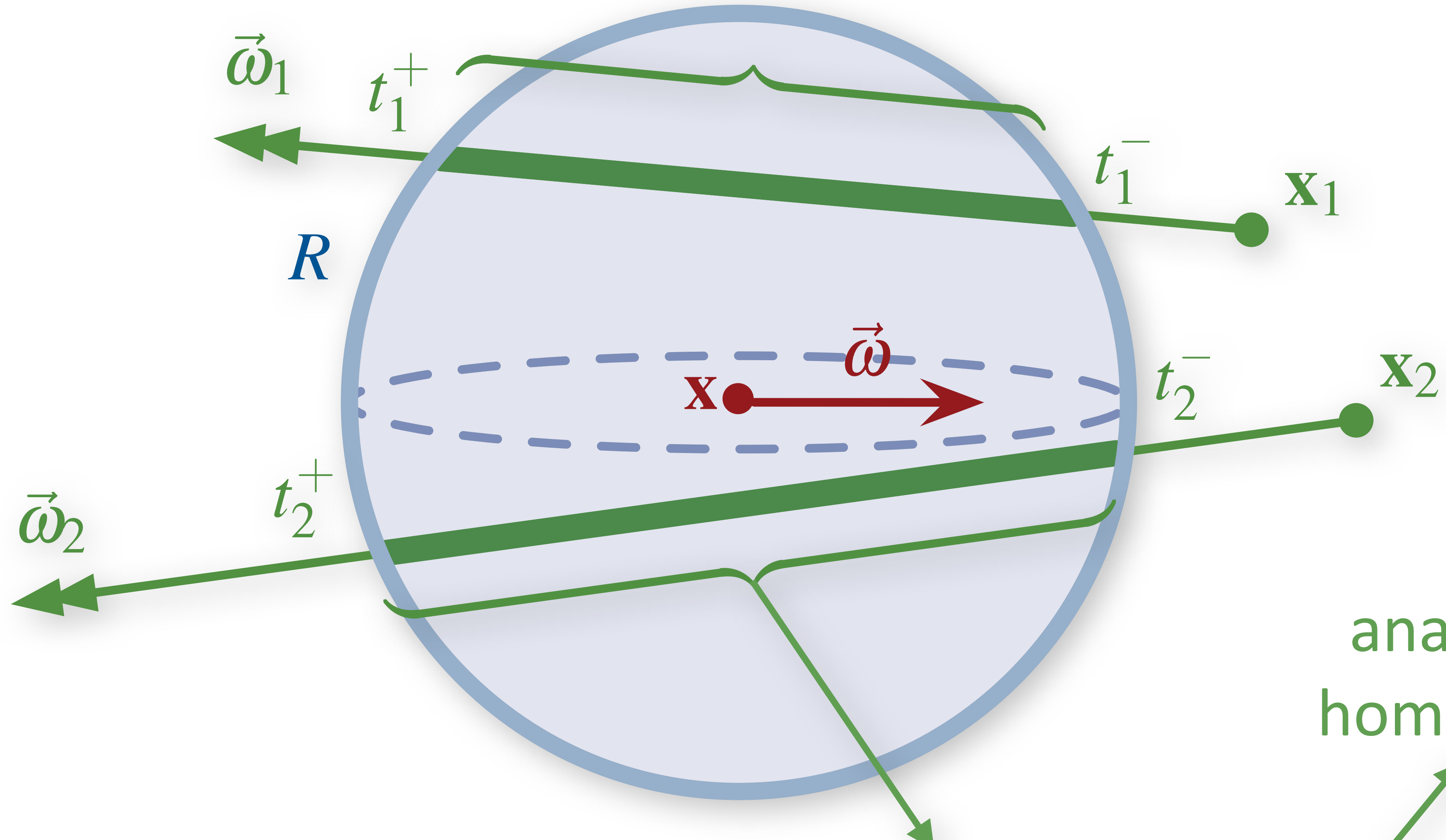
$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i) \boxed{\Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t}\, dt}$$

Point Query × Beam Data (3D blur)



$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i) \boxed{\Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t} dt}$$

Point Query × Beam Data (3D blur)

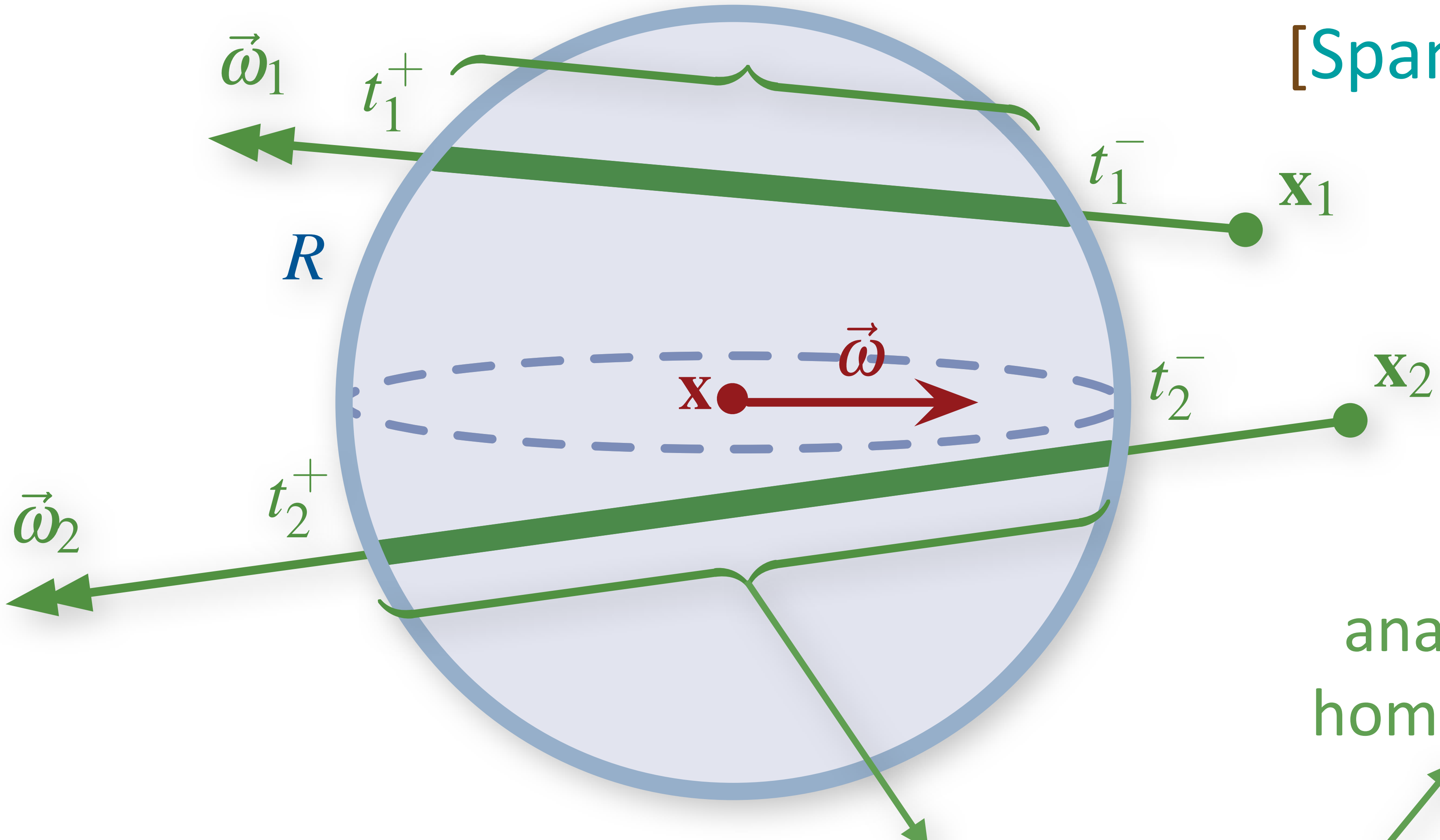


analytic solution in homogeneous media

$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i) \Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t} dt$$

Point Query × Beam Data (3D blur)

related to: "track length" estimators
[Spanier and Gelbard 69]

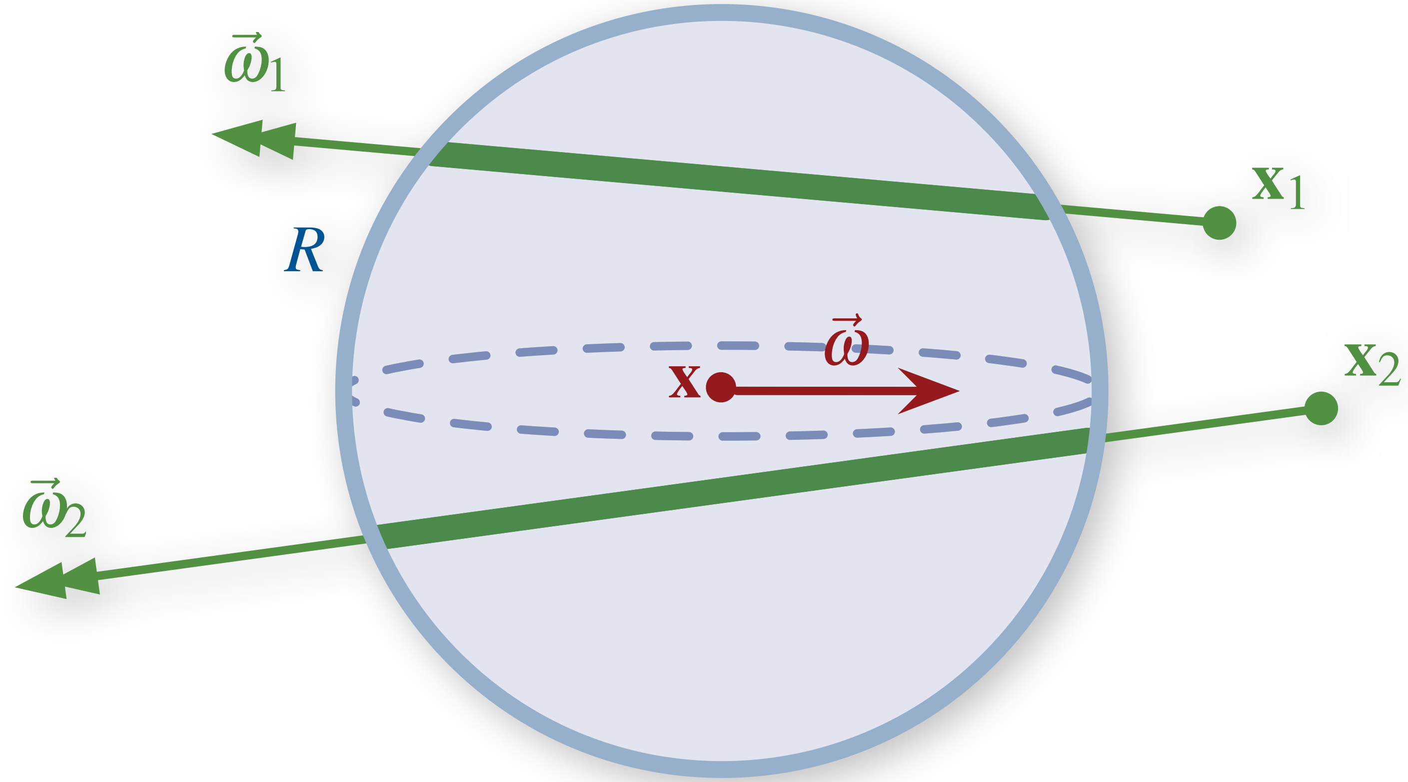


analytic solution in
homogeneous media

$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i) \Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t} dt$$

Reducing Blur Dimensionality

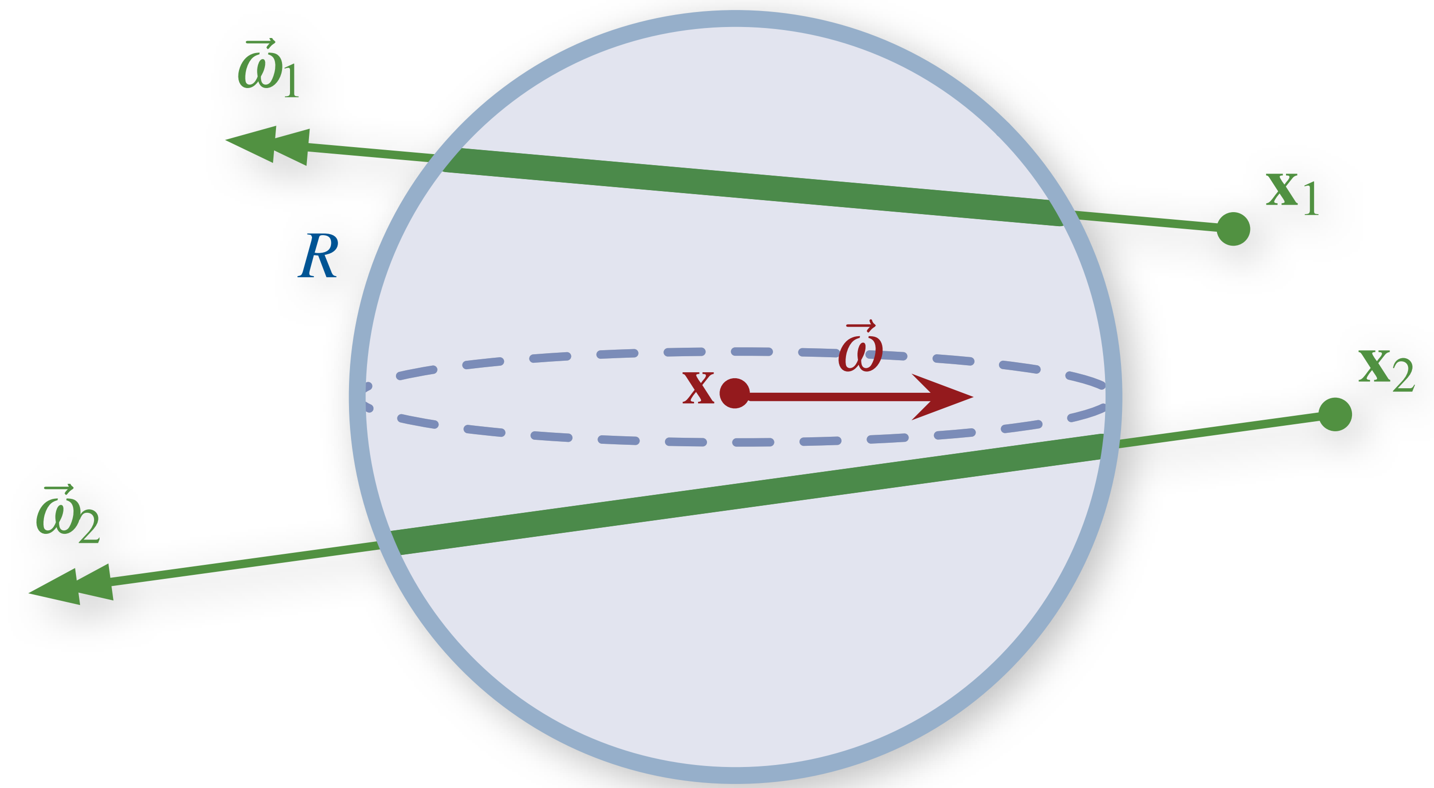
Point Query x Beam Data (3D blur)



$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i)\,\Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t}\,dt$$

Reducing Blur Dimensionality

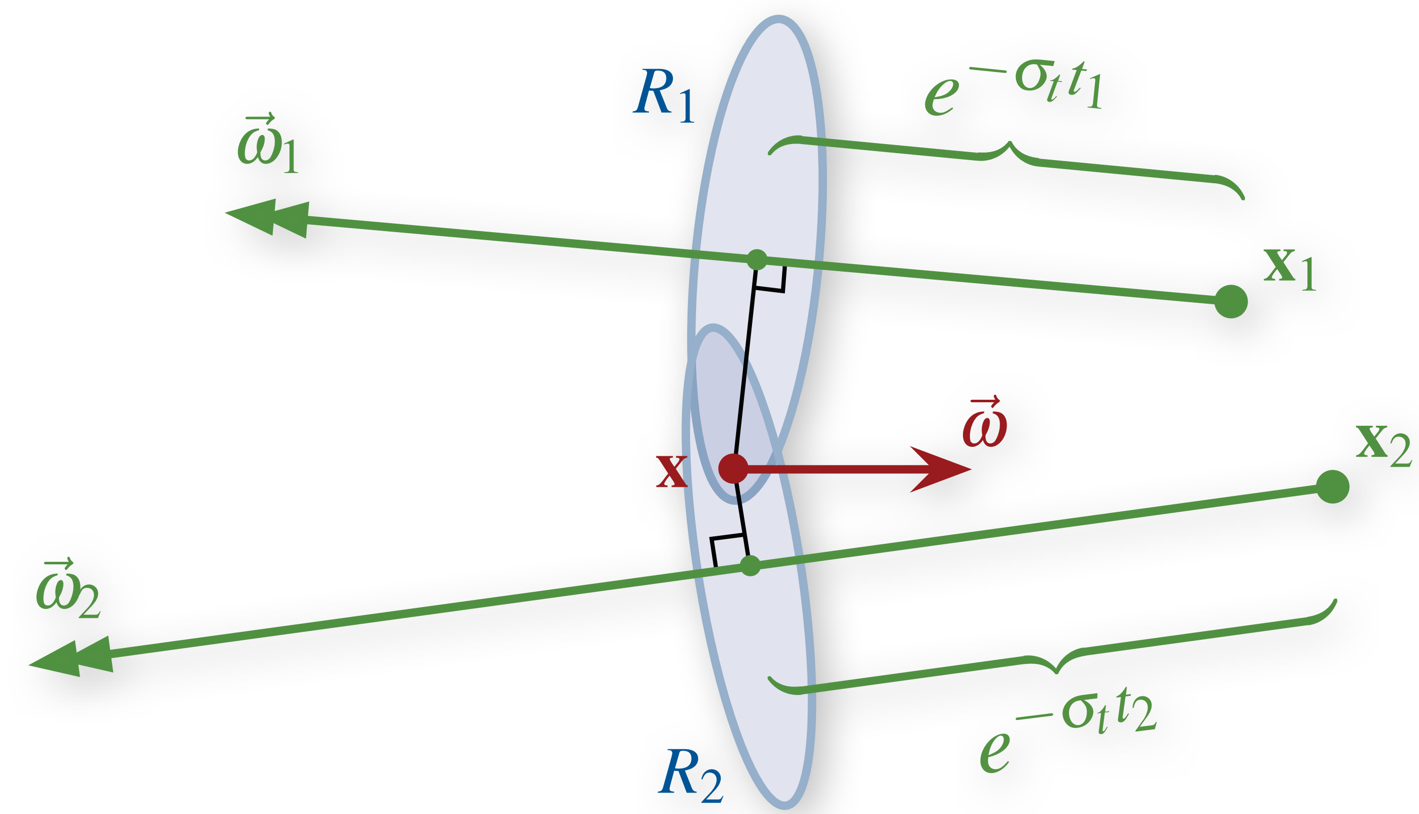
Point Query x Beam Data (3D blur)



$$L \approx \frac{1}{\mu_R(r^3)} \sum_i f(\theta_i) \Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t}\, dt$$

Reducing Blur Dimensionality

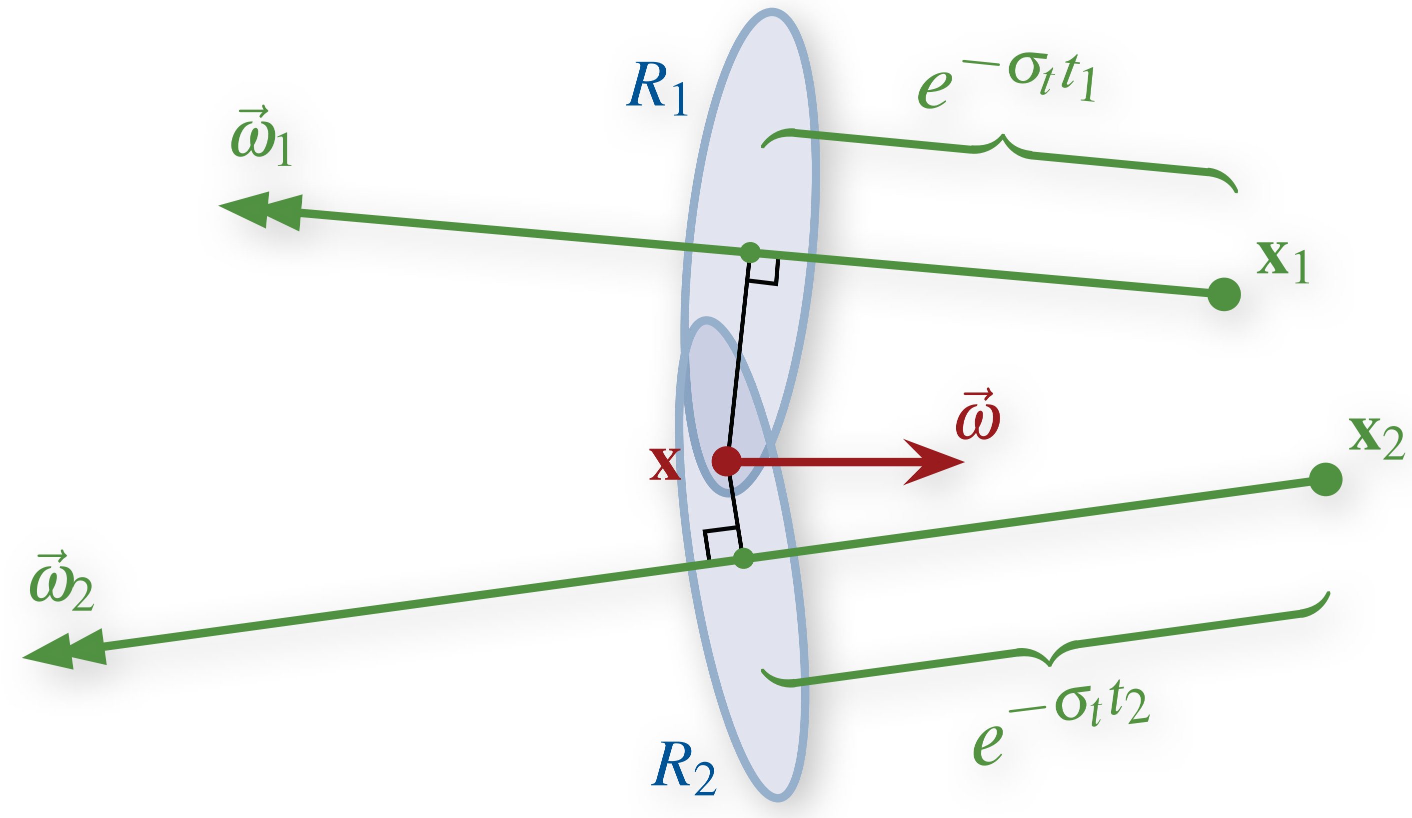
Point Query x Beam Data (2D blur)



$$L \approx \frac{1}{\mu_R(r^2)} \sum_i f(\theta_i) \Phi_i e^{-\sigma_t t_i}$$

Reducing Blur Dimensionality

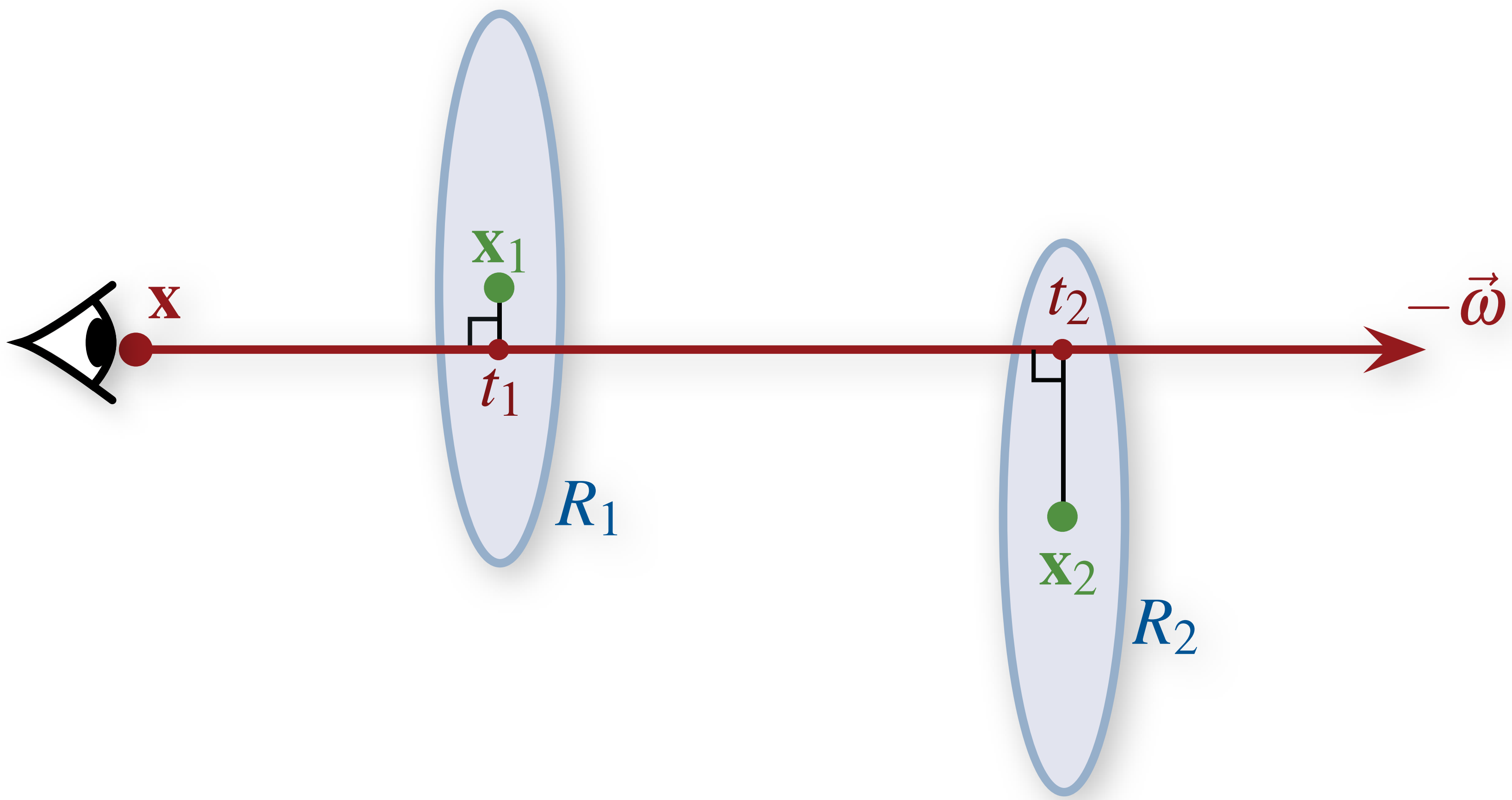
Point Query x Beam Data (2D blur)



$$L \approx \frac{1}{\mu_R(r^2)} \sum_i f(\theta_i)\, \Phi_i\, e^{-\sigma_t t_i}$$

Radiometric Duality

Beam Query x Point Data (2D blur)

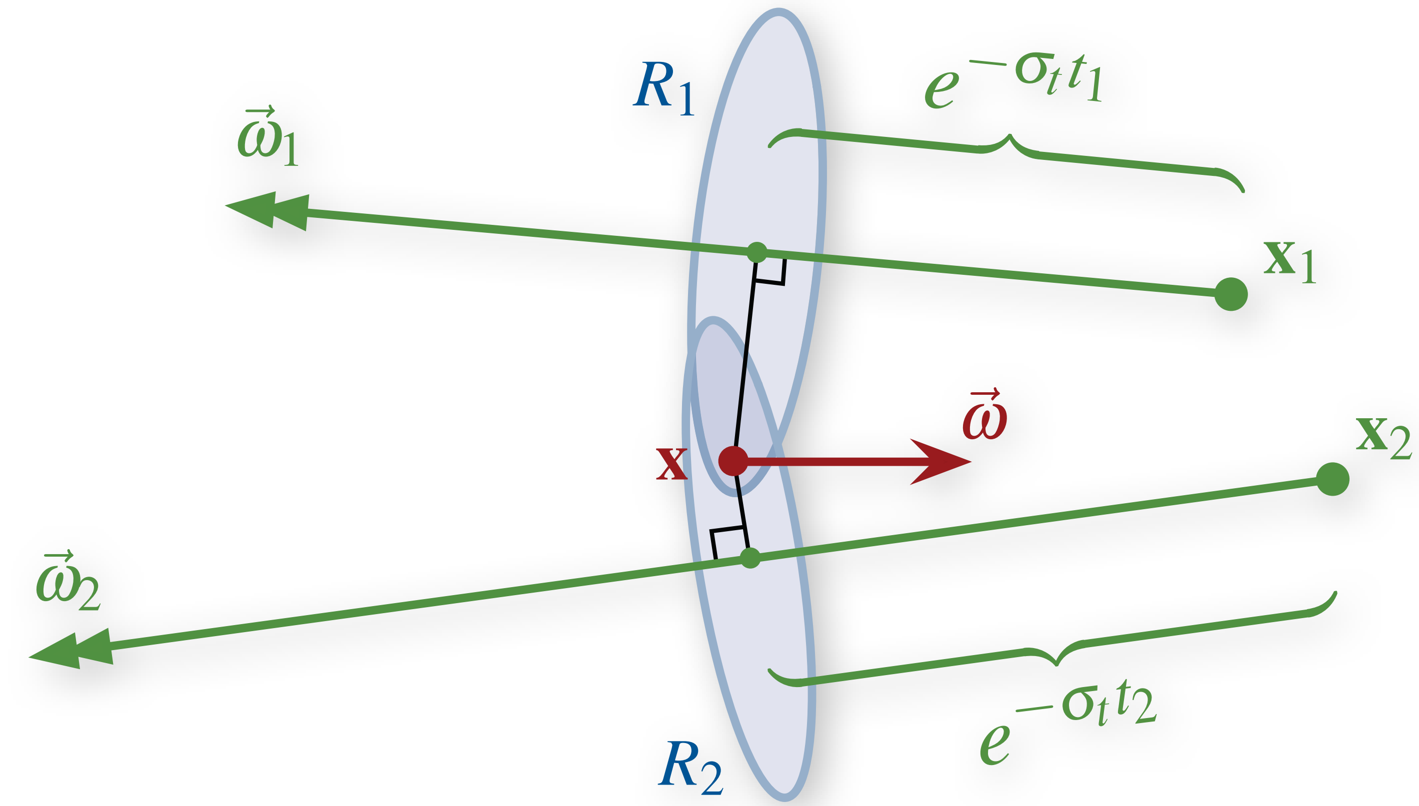


$$L \approx \frac{1}{\mu_R(r^2)} \sum_i f(\theta_i)\, \Phi_i\, e^{-\sigma_t t_i}$$

"Beam Radiance Estimate"

[Jarosz et al. 08]

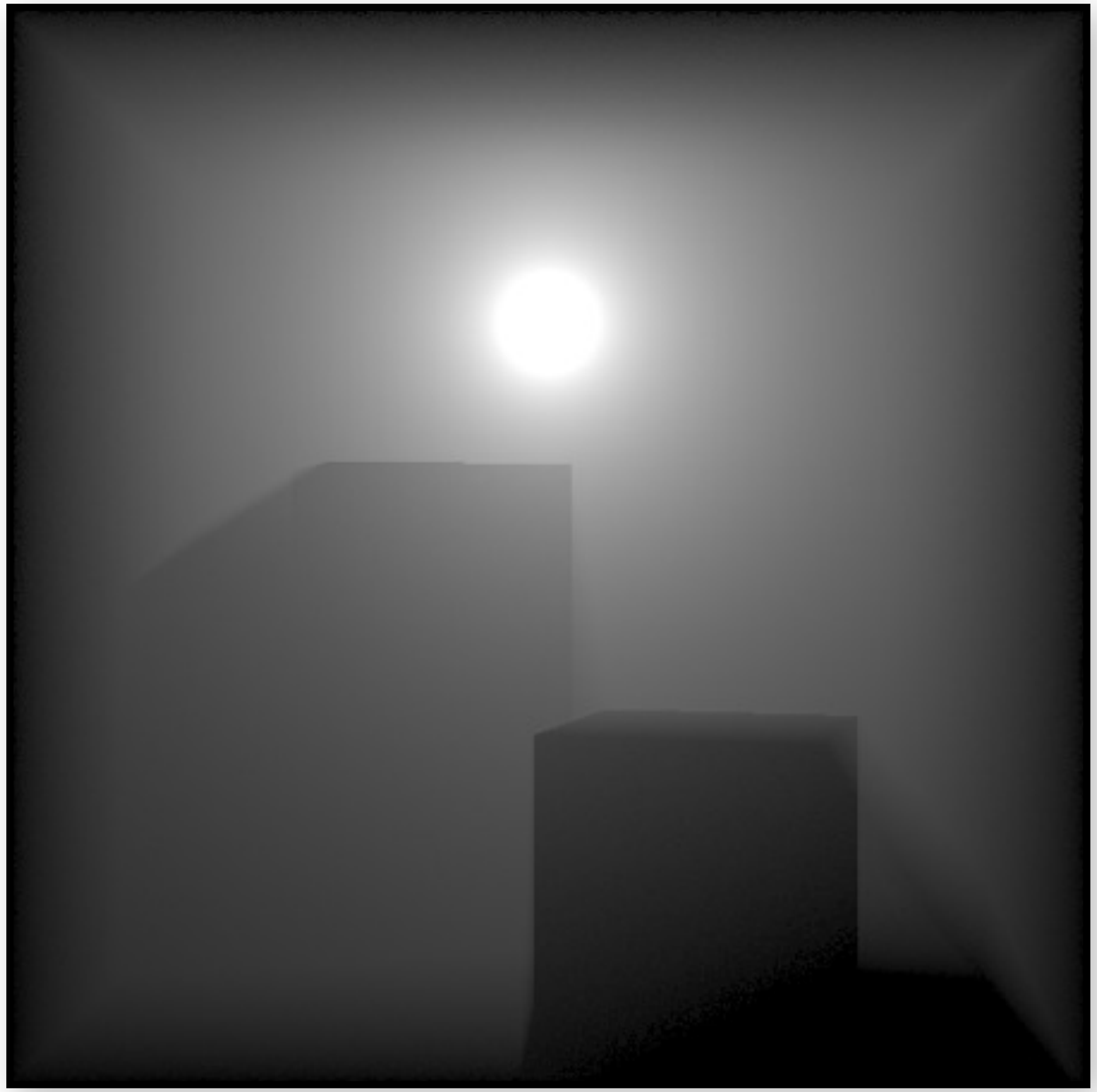
Point Query x Beam Data (2D blur)



$$L \approx \frac{1}{\mu_R(r^2)} \sum_i f(\theta_i)\, \Phi_i\, e^{-\sigma_t t_i}$$

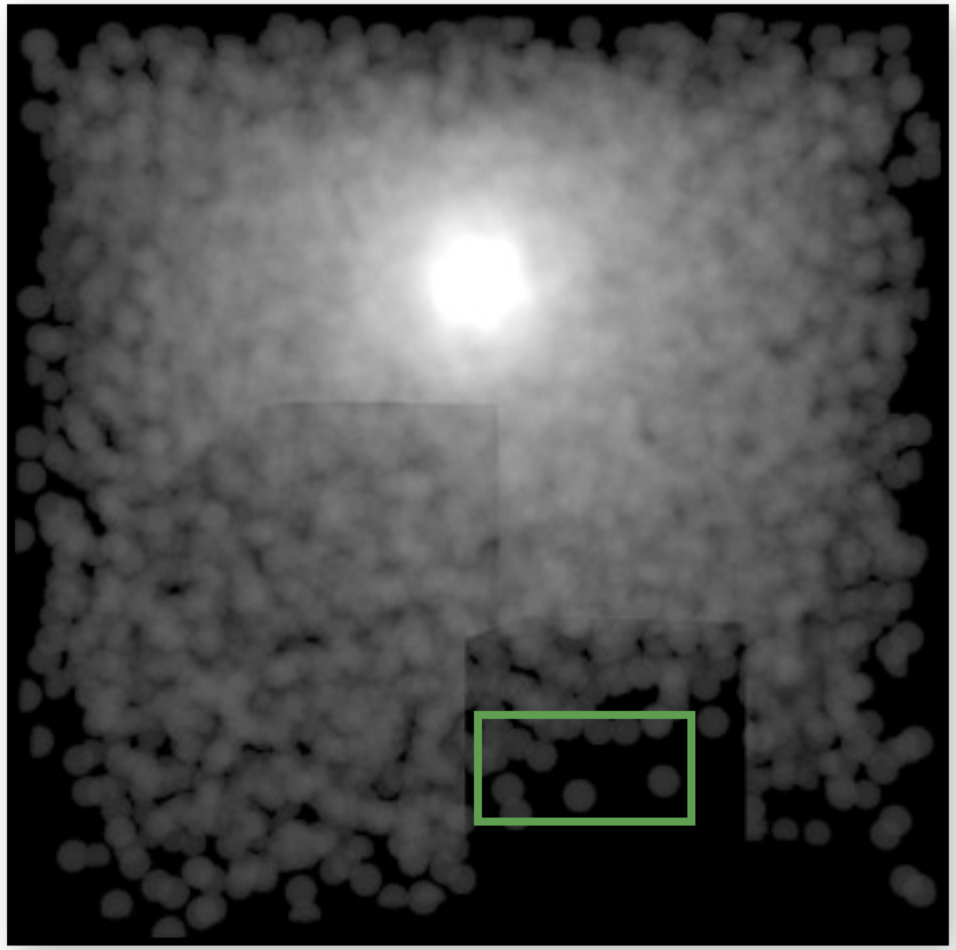
Photon Points vs. Photon Beams

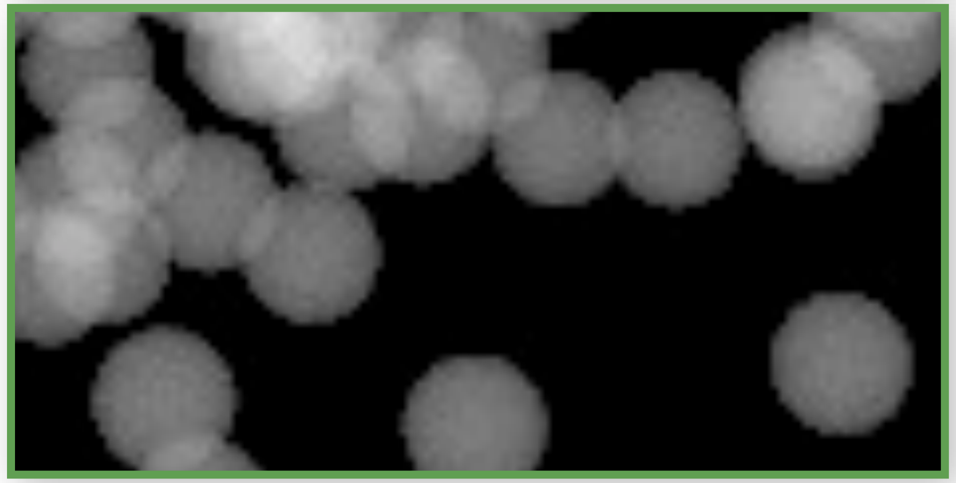
Ground Truth



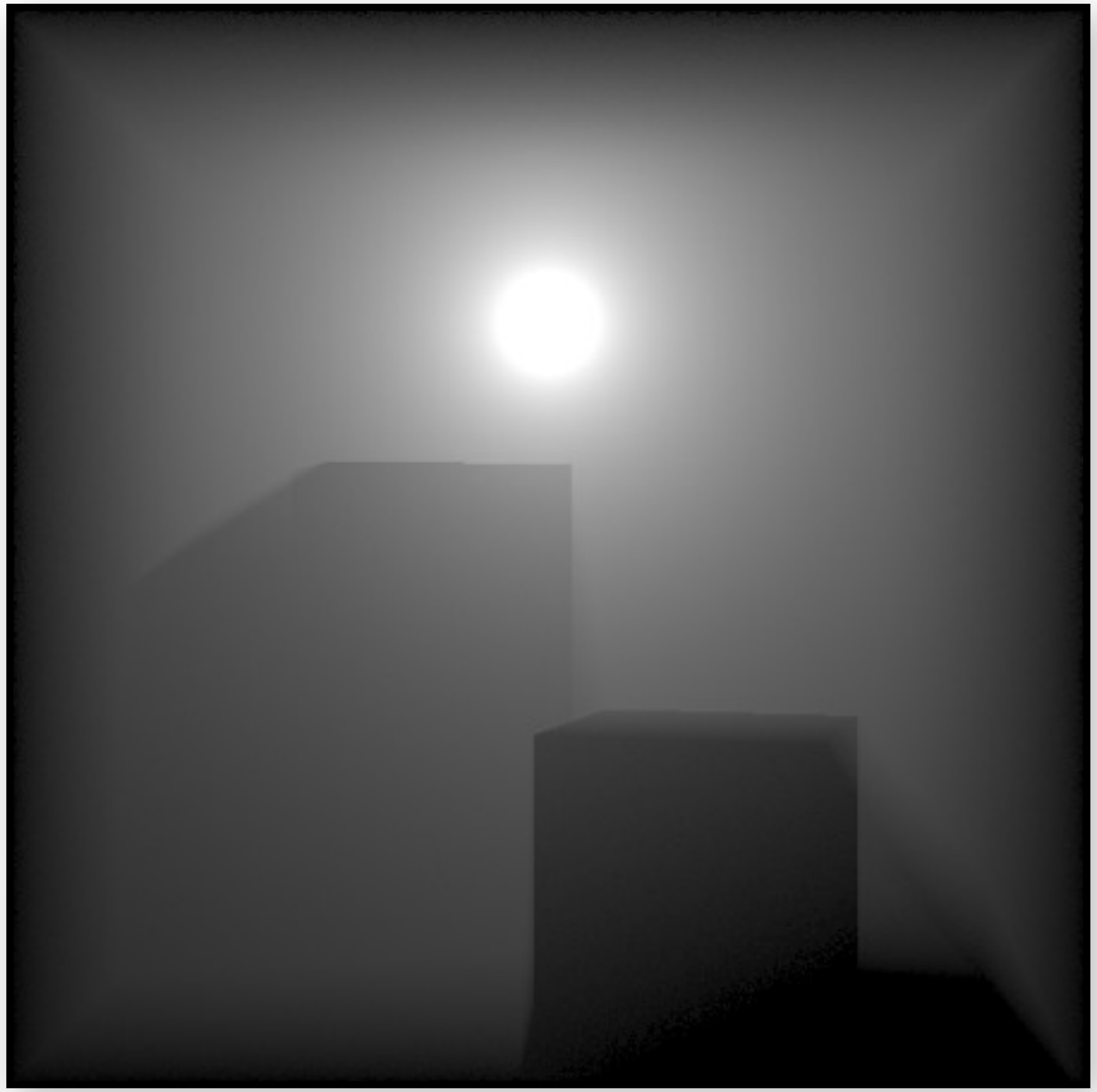
Photon Points vs. Photon Beams

100k Photon Points



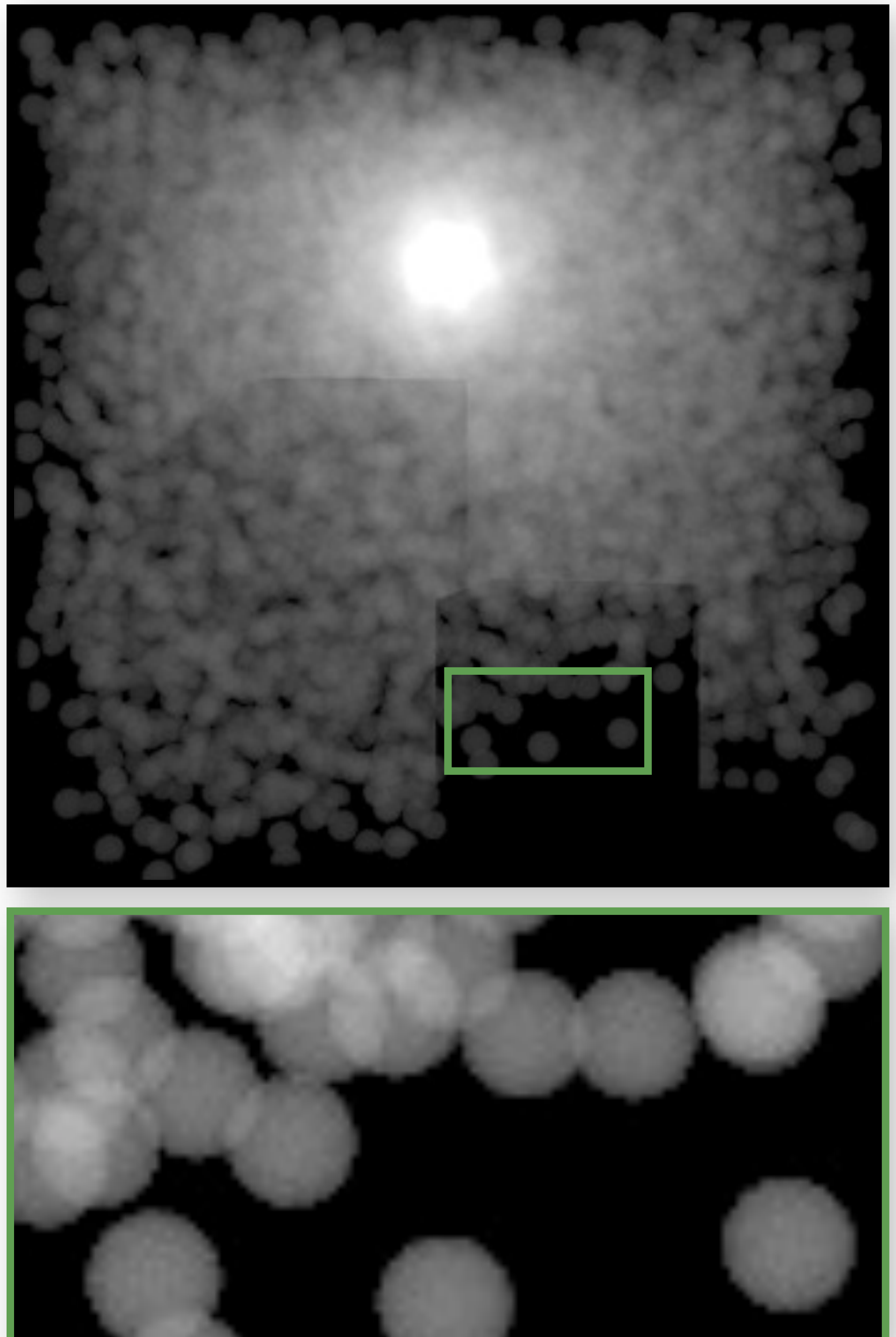


Ground Truth

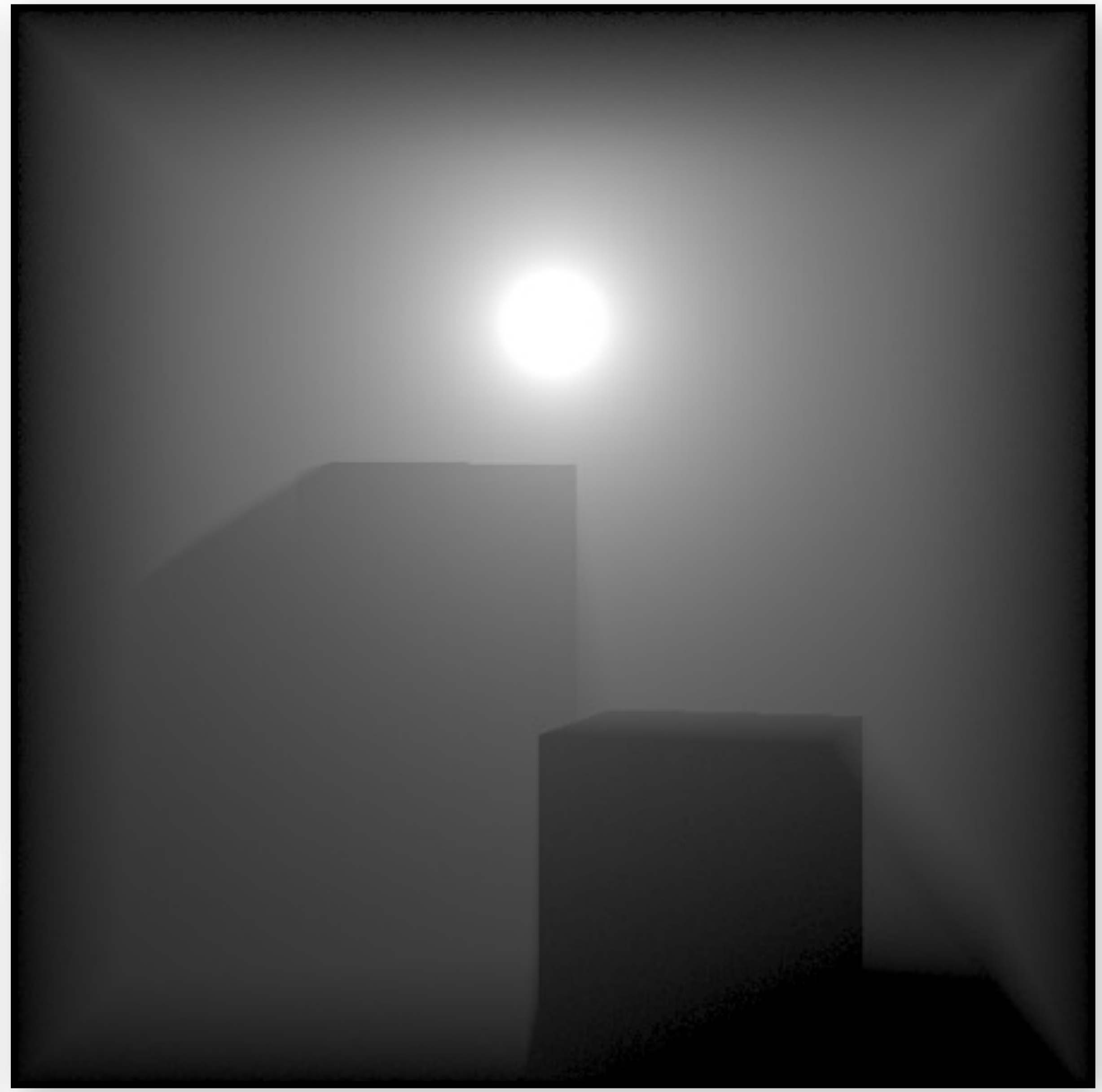


Photon Points vs. Photon Beams

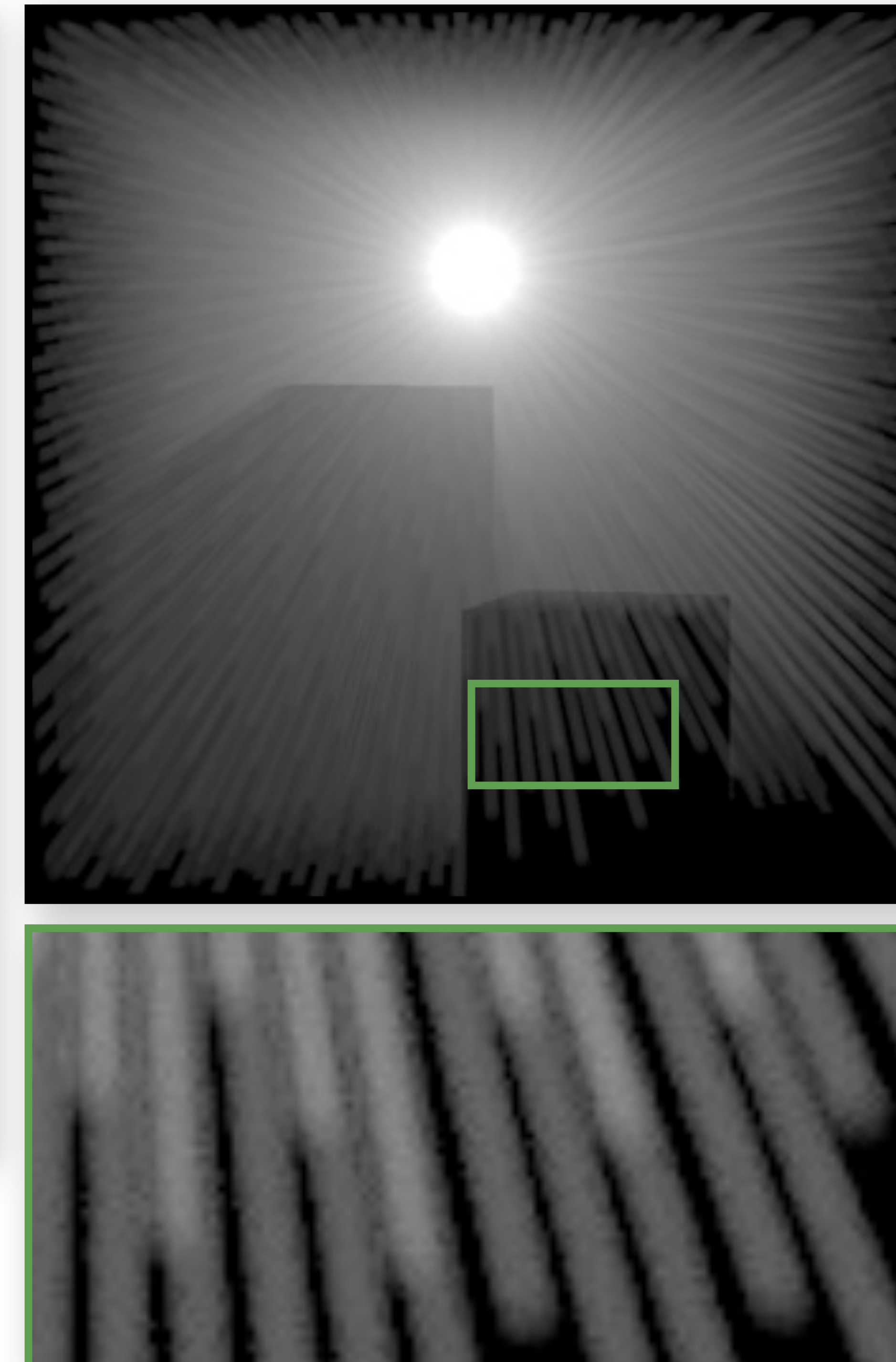
100k Photon Points



Ground Truth



5k Photon Beams

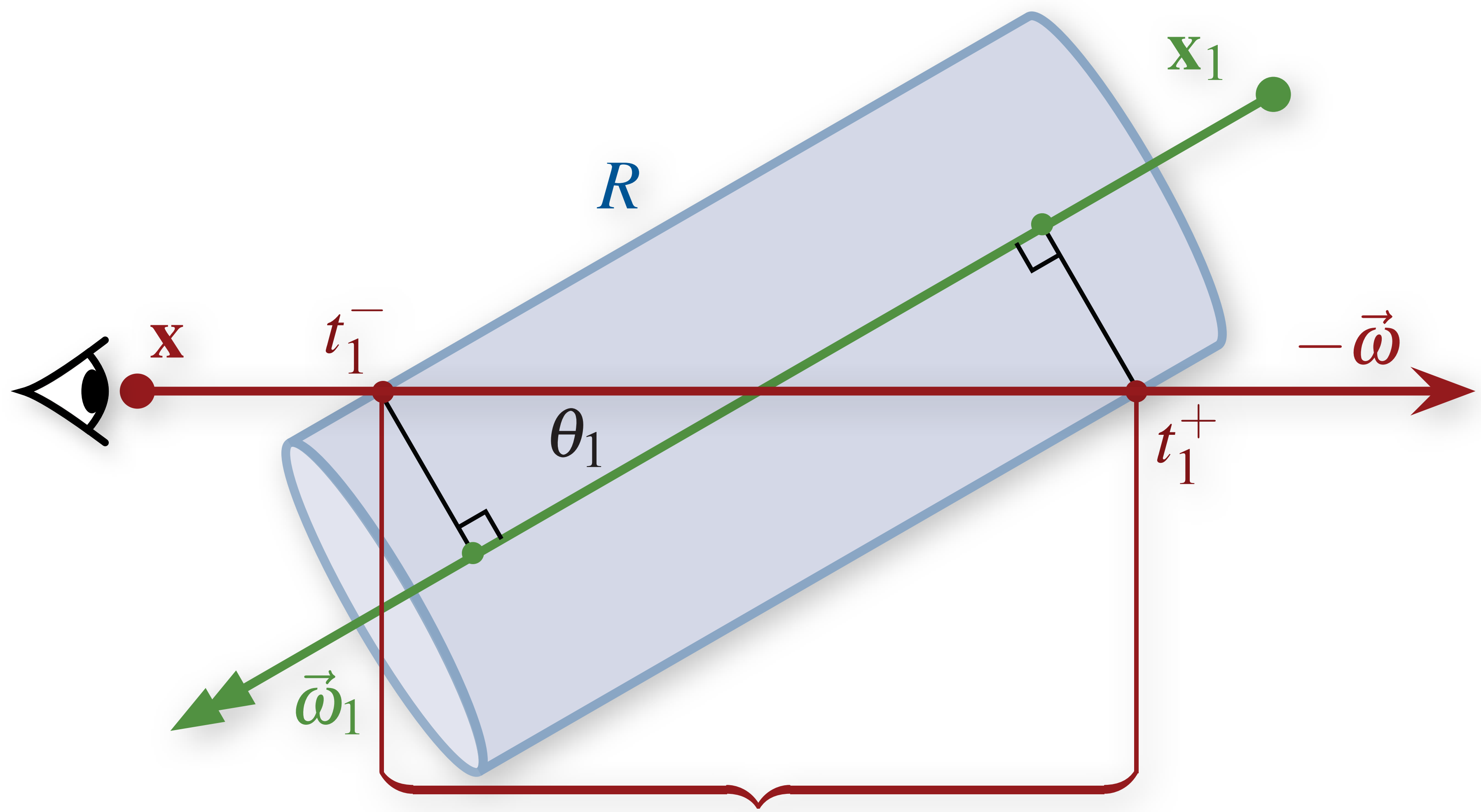


Beam Queries with Photon Beams

- Beam Query x Beam Data (3D)
- Beam Query x Beam Data $(2D)_1$
- Beam Query x Beam Data $(2D)_2$
- Beam Query x Beam Data (1D)

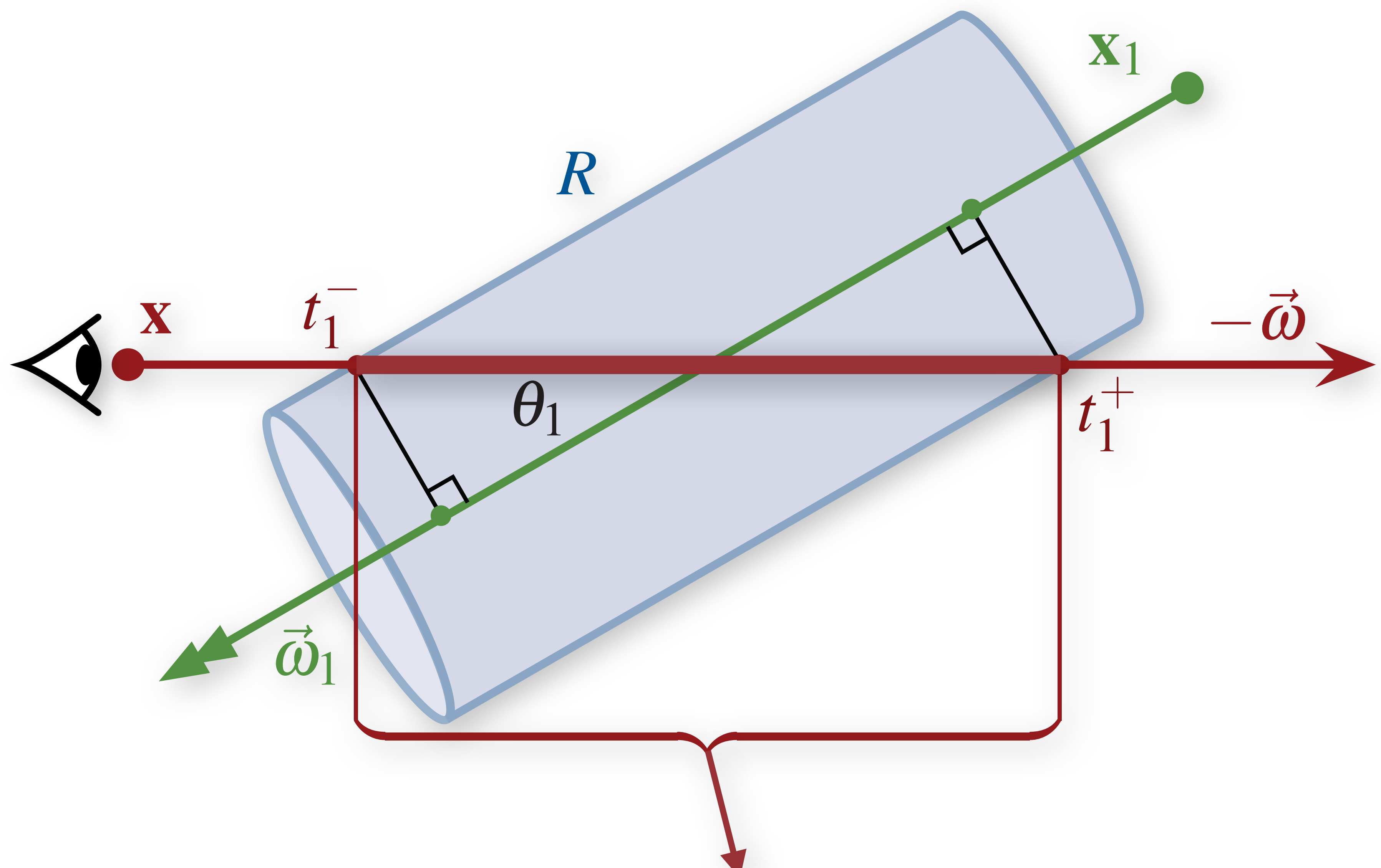
Beam Query × Beam Data (2D blur)

Beam Query × Beam Data (2D blur)



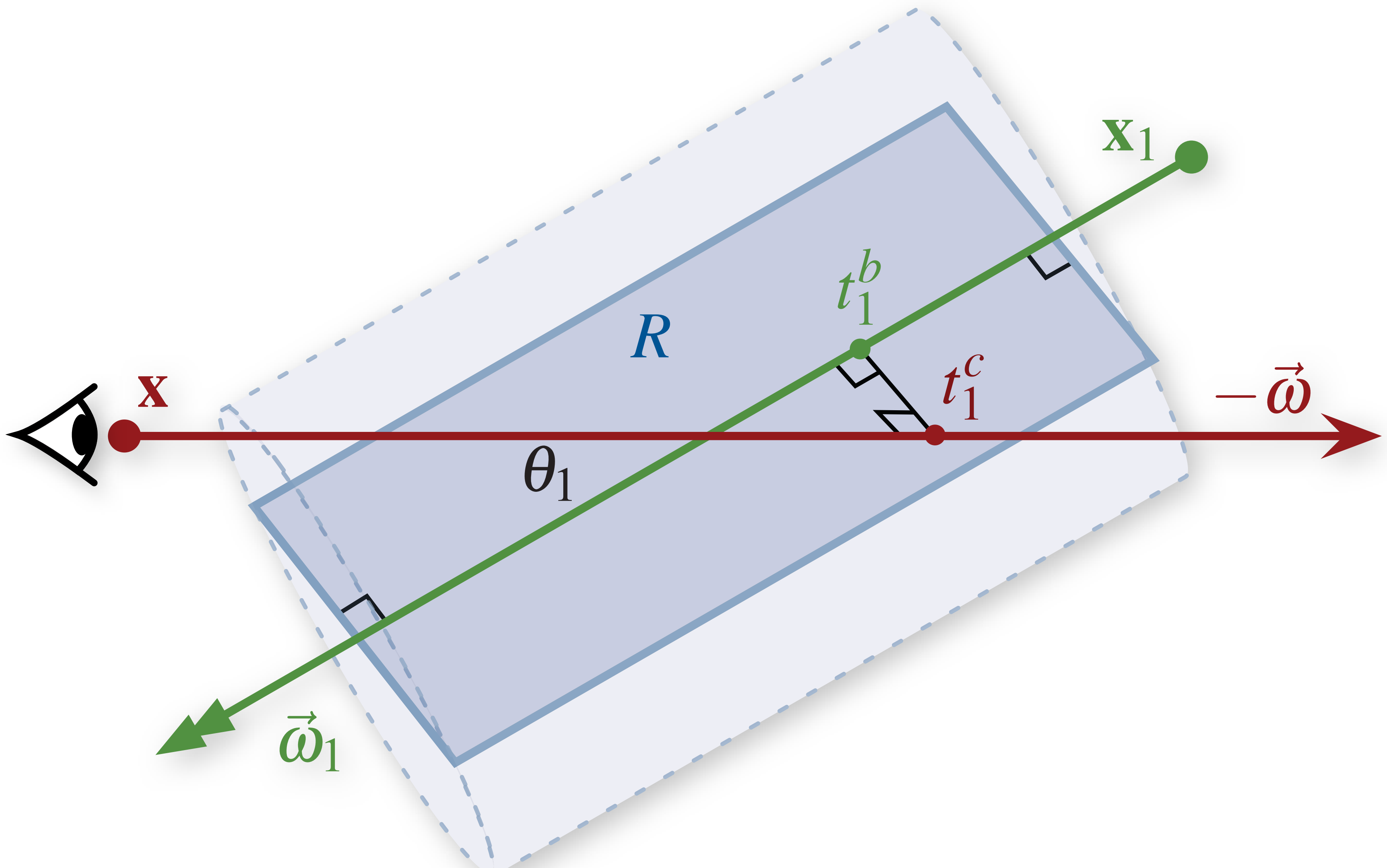
$$L \approx \frac{\sigma_s}{\mu_R(r^2)} \sum_i f(\theta_i)\,\Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t_c} e^{-\sigma_t t_b}\, dt_c$$

Beam Query × Beam Data (2D blur)



$$L \approx \frac{\sigma_s}{\mu_R(r^2)} \sum_i f(\theta_i)\, \Phi_i \int_{t_i^-}^{t_i^+} e^{-\sigma_t t_c} e^{-\sigma_t t_b}\, dt_c$$

Beam Query × Beam Data (1D blur)



$$L \approx \frac{\sigma_s}{\mu_R(r)} \sum_i \frac{f(\theta_i)\, \Phi_i\, e^{-\sigma_t t_i^c}\, e^{-\sigma_t t_i^b}}{\sin\theta_i}$$

Radiance Estimator Summary

Radiance Estimator Summary

- Beam queries remove ray marching

Radiance Estimator Summary

- Beam queries remove ray marching
- Beam data increases data density

Radiance Estimator Summary

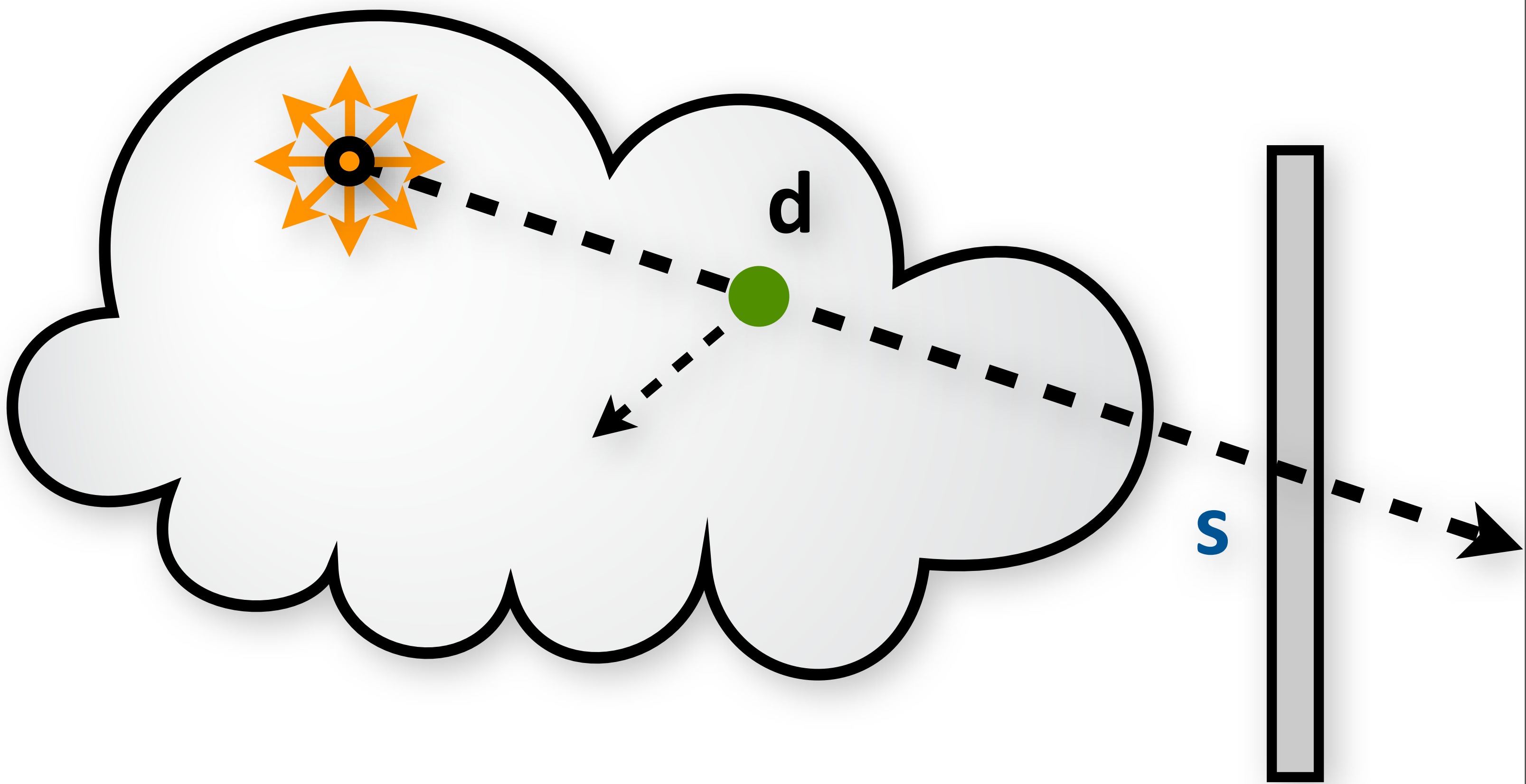
- Beam queries remove ray marching
- Beam data increases data density
- Lower blur dimension reduces bias and computation

Radiance Estimator Summary

- Beam queries remove ray marching
- Beam data increases data density
- Lower blur dimension reduces bias and computation
- **use: Beam Query x Beam Data (1D)**

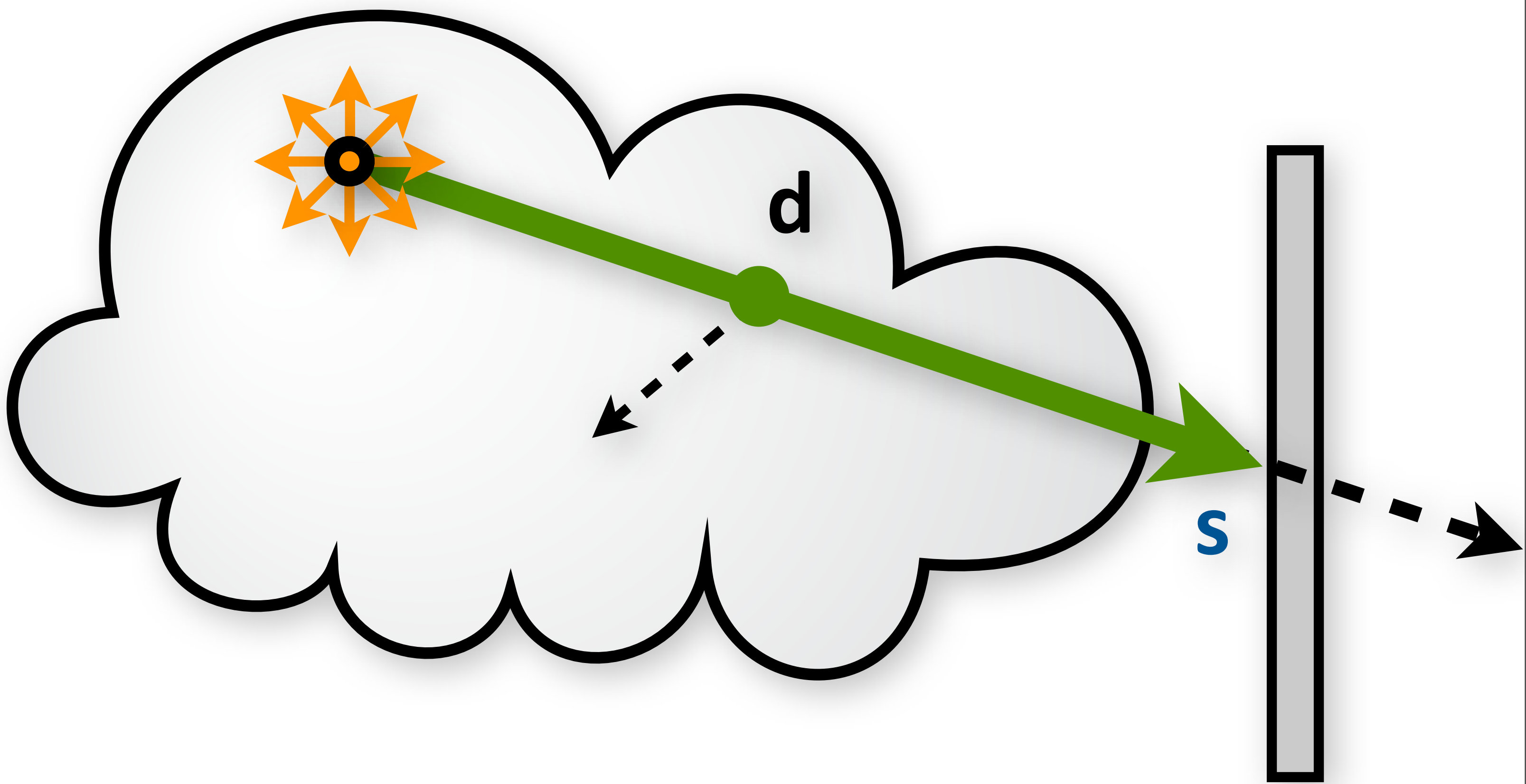
Basic Volumetric Photon Tracer

```
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)
```



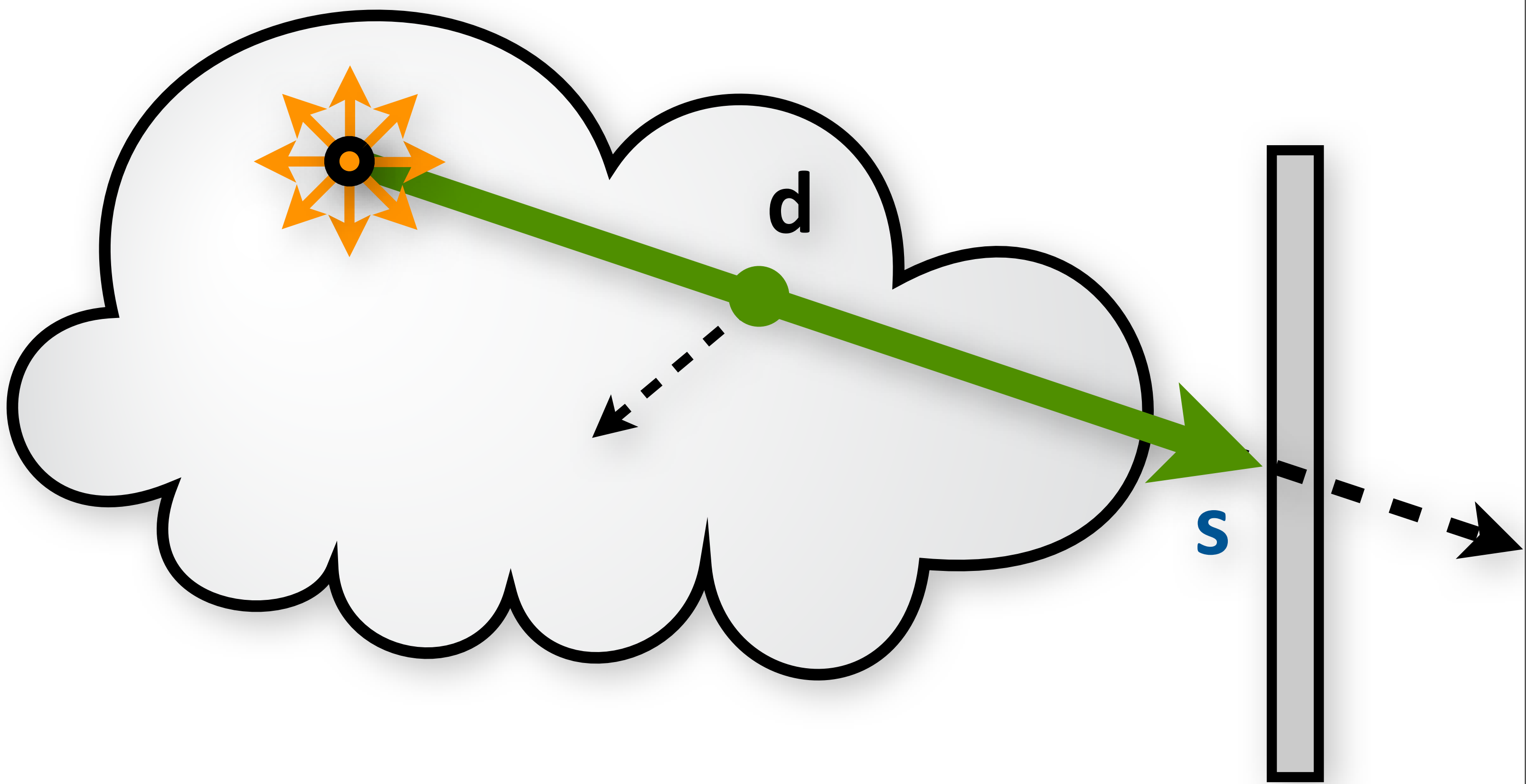
Basic Volumetric Photon Tracer

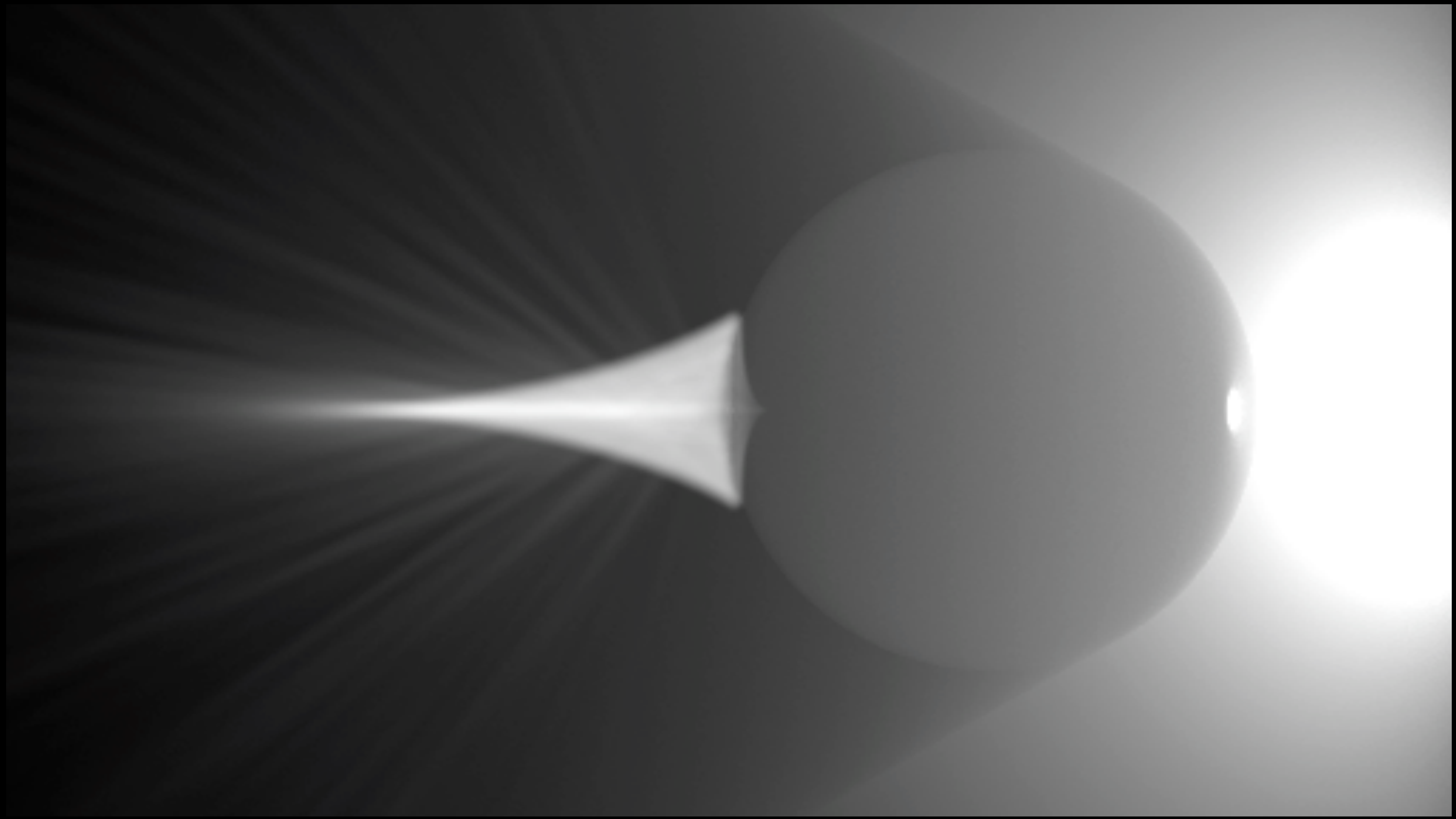
```
void vPT(o, ω, Φ)
    s = nearestSurfaceHit(o, ω)
    storeVolumePhoton(o, ω, Φ)
    d = freeFlightDistance(o, ω)
    if (d < s)          // media scattering
        o += d*ω        // propagate photon
        return vPT(o, samplePF(), Φ * σs / σt)
    else                // surface scattering
 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

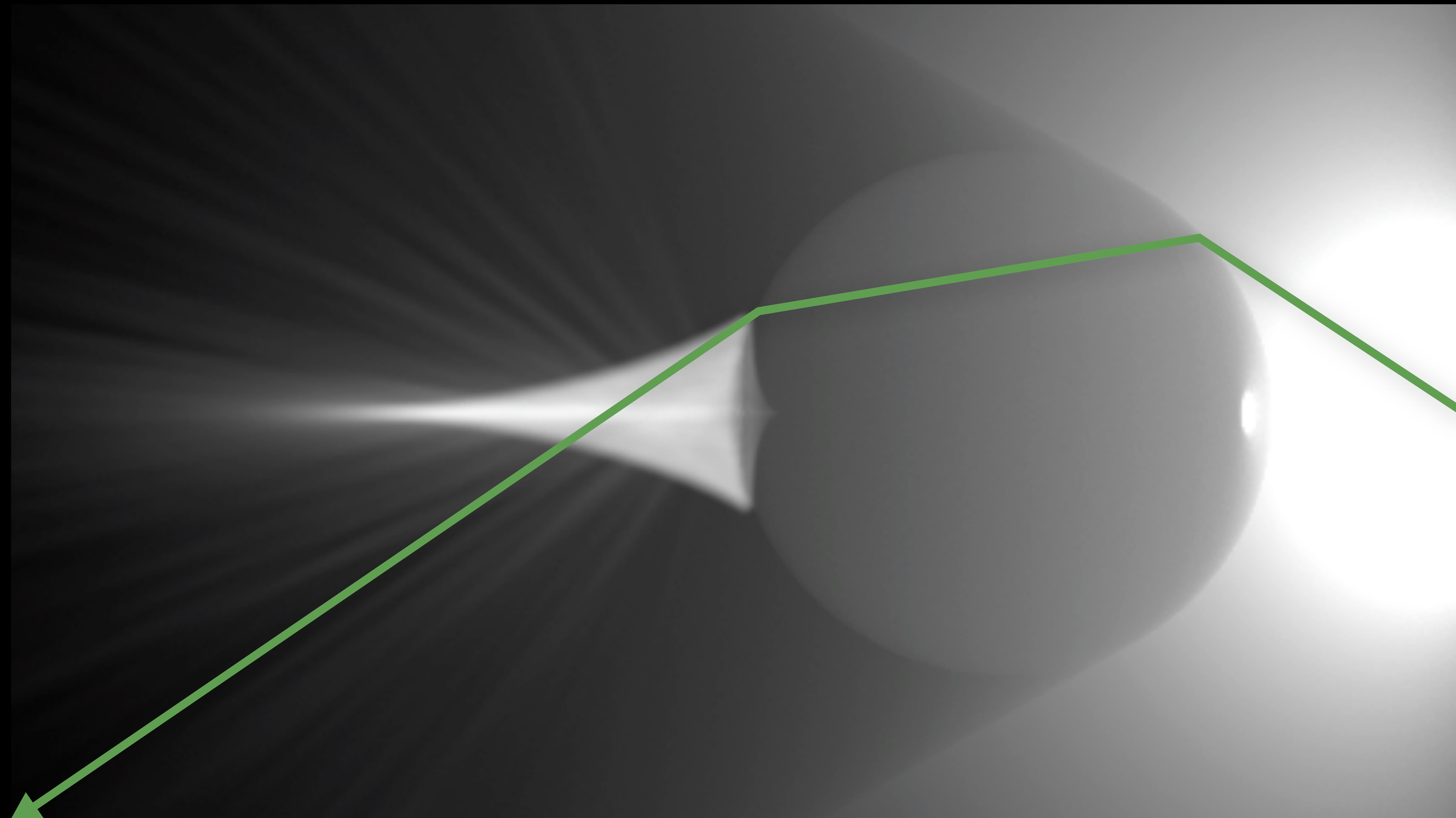


Basic Volumetric Photon Tracer

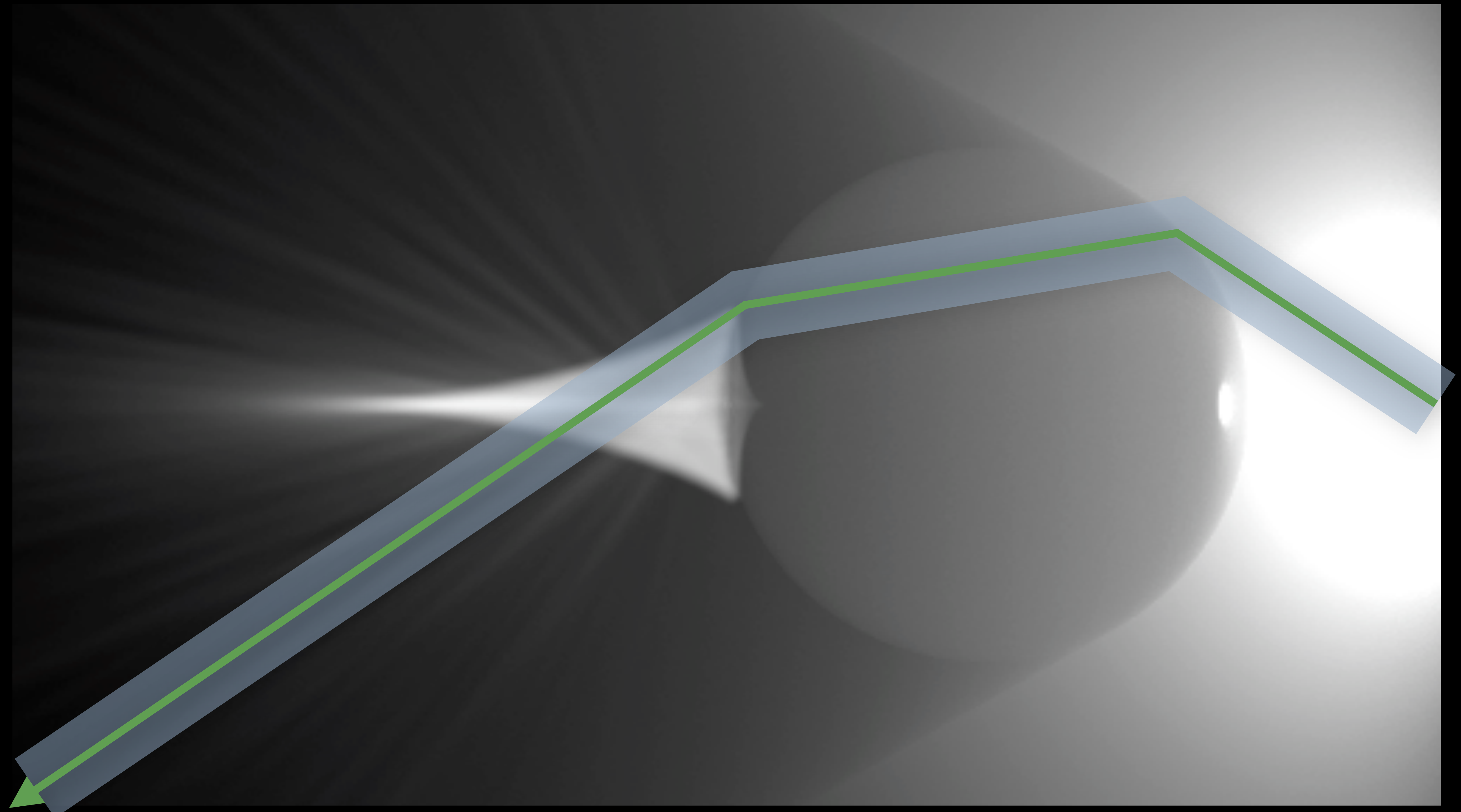
```
void vPT(o, ω, Φ)
    s = nearestSurfaceHit(o, ω)
    storePhotonBeam(o, ω, s, Φ)
    d = freeFlightDistance(o, ω)
    if (d < s)          // media scattering
        o += d*ω        // propagate photon
        return vPT(o, samplePF(), Φ * σs / σt)
    else                // surface scattering
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



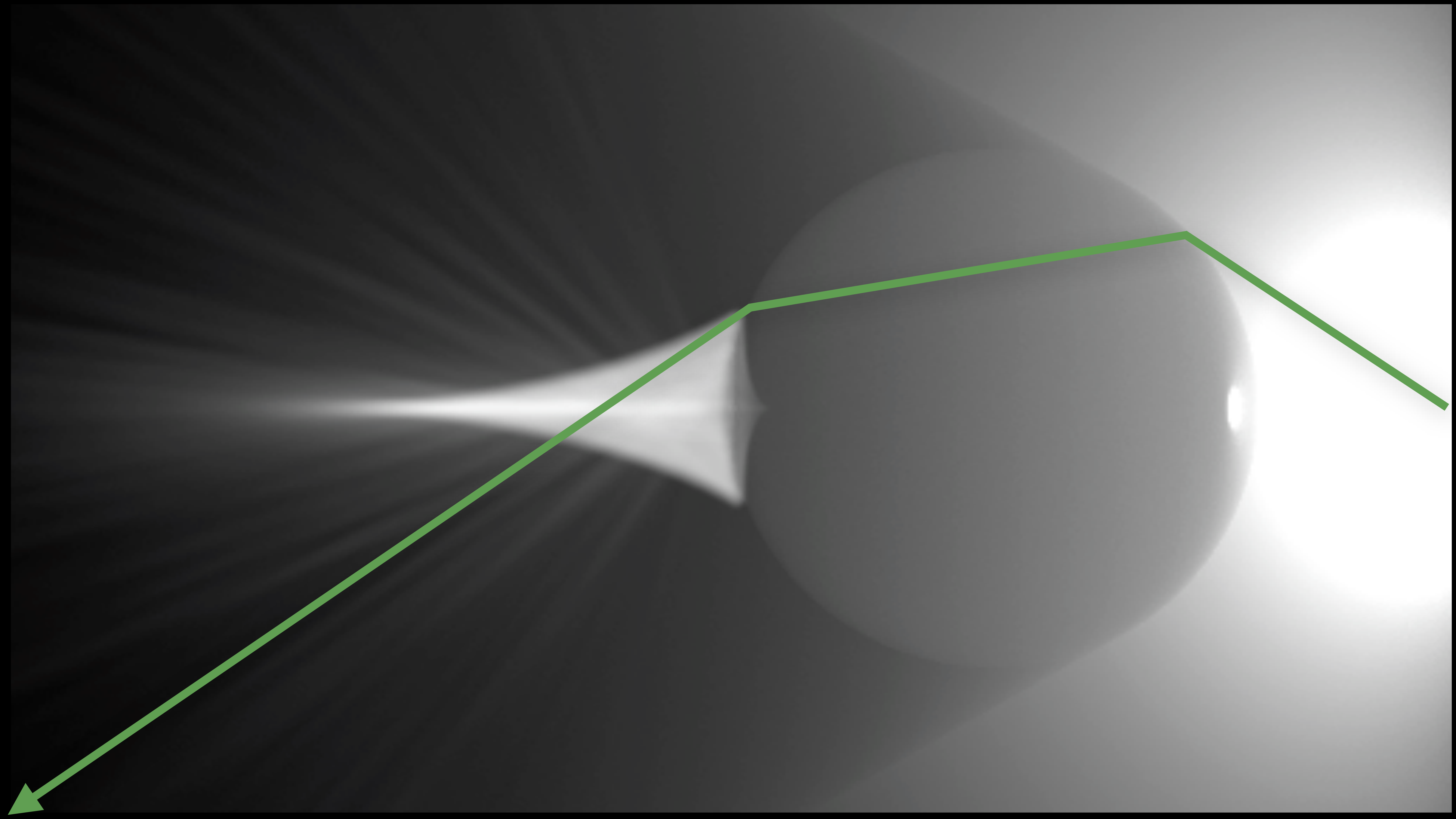




Fixed-width Beams

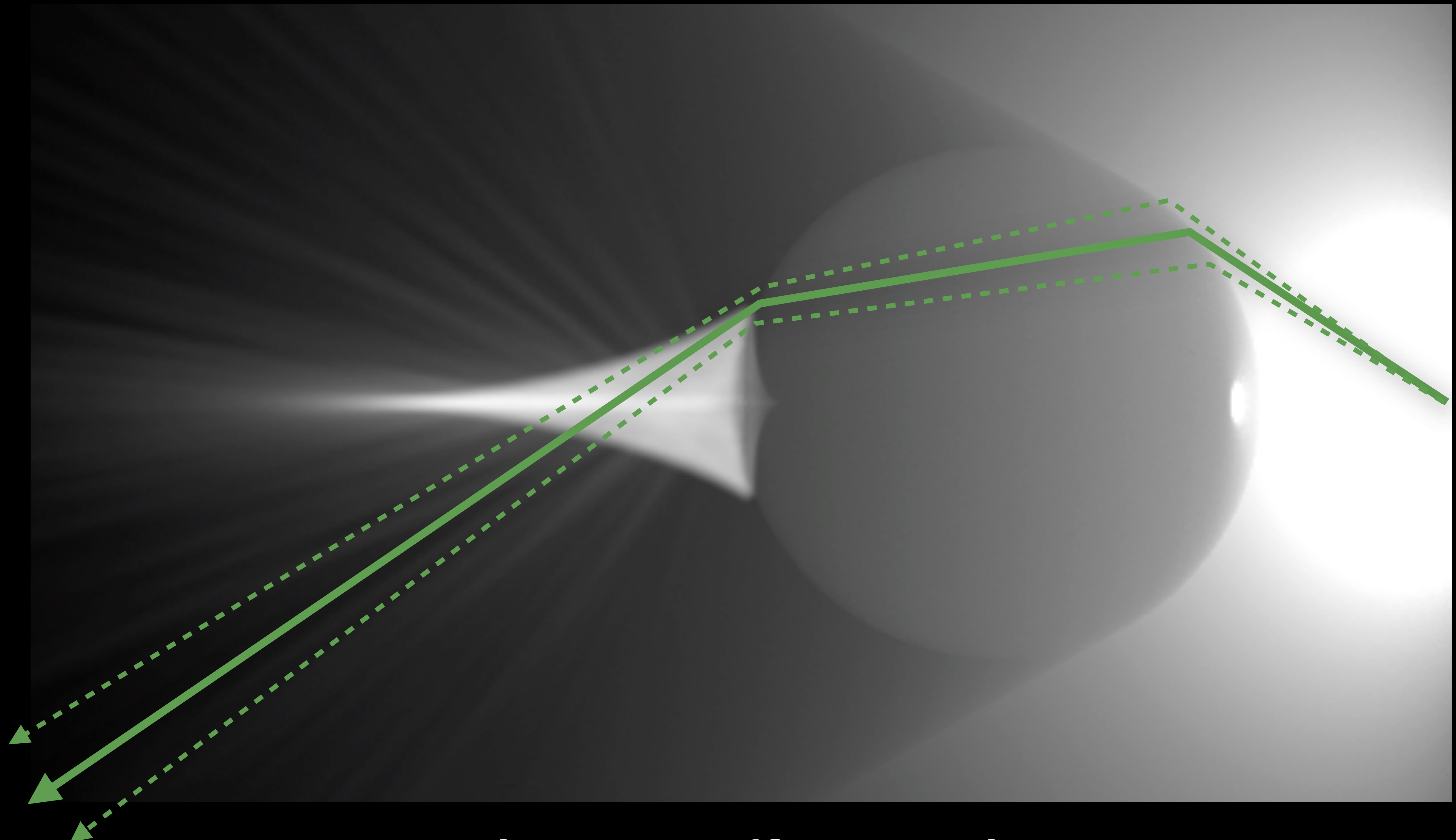


Fixed-width Beams



Photon Differentials
[Igehy 99, Schjøth et al. 07]

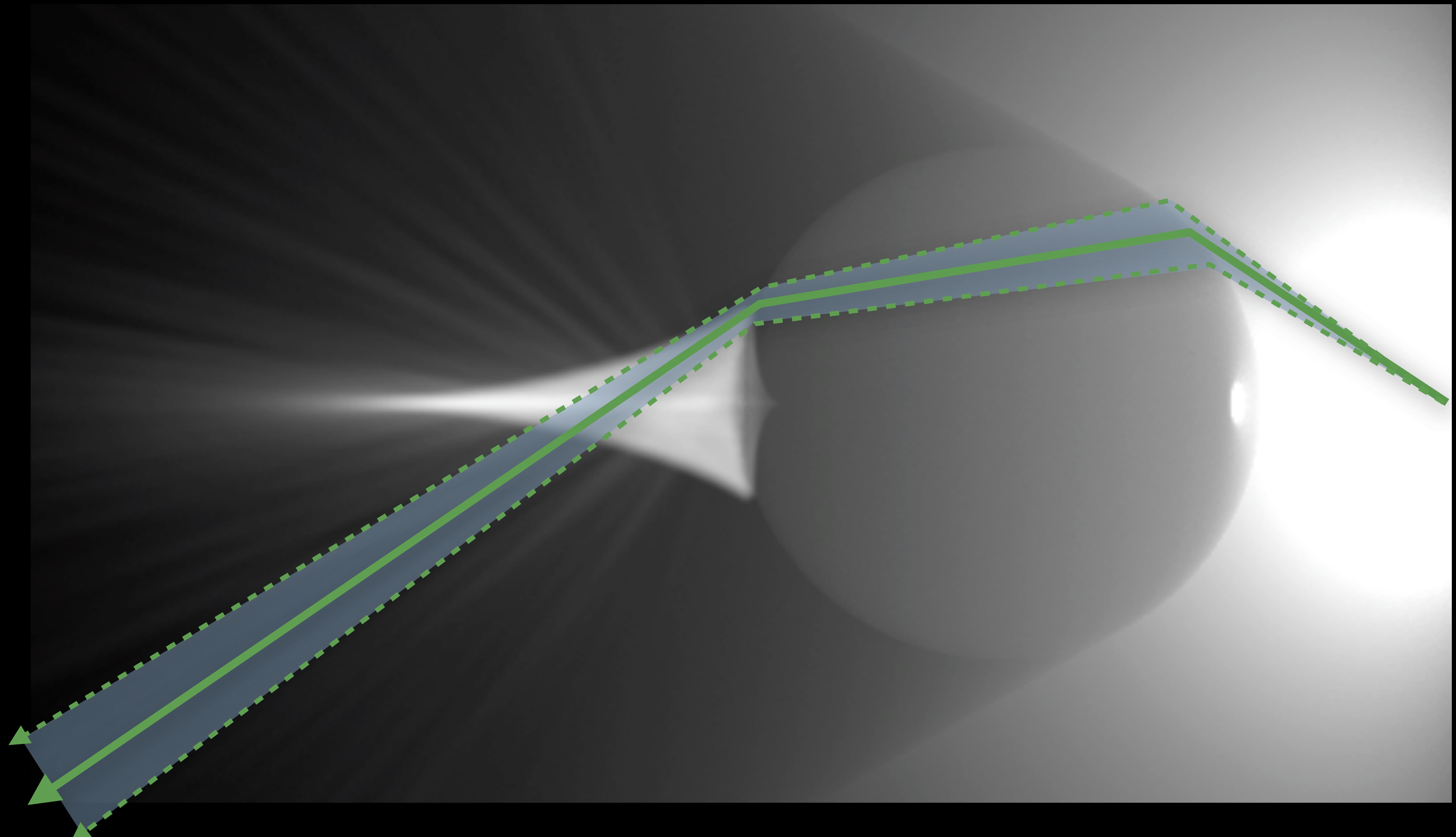
Fixed-width Beams



Photon Differentials

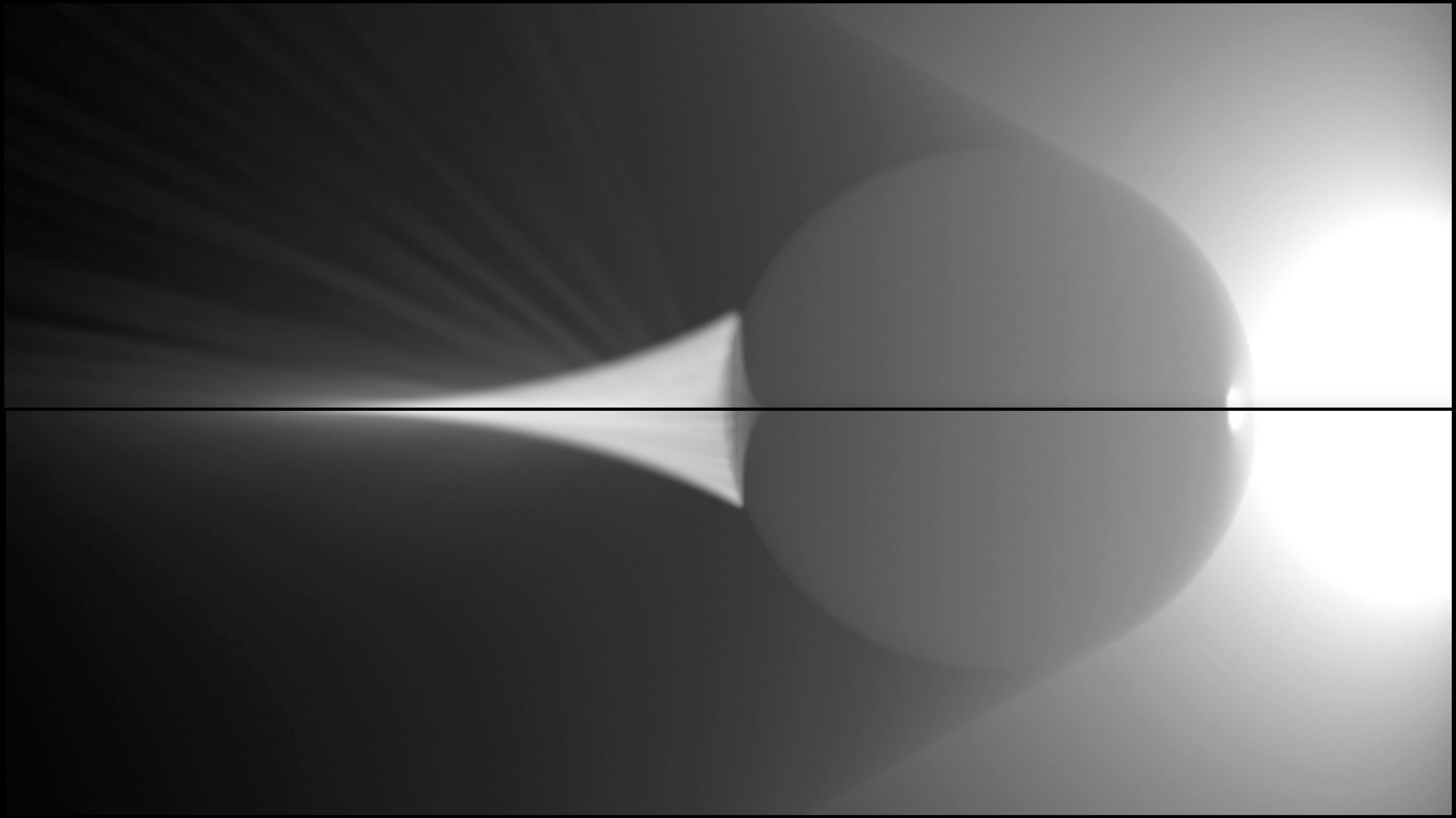
[Igehy 99, Schjøth et al. 07]

Fixed-width Beams



Photon Differentials
[Igehy 99, Schjøth et al. 07]

Fixed-width Beams



Adaptive-width Beams

Rendering

Rendering

- Need to intersect each ray with all photon beams (expensive!)

Rendering

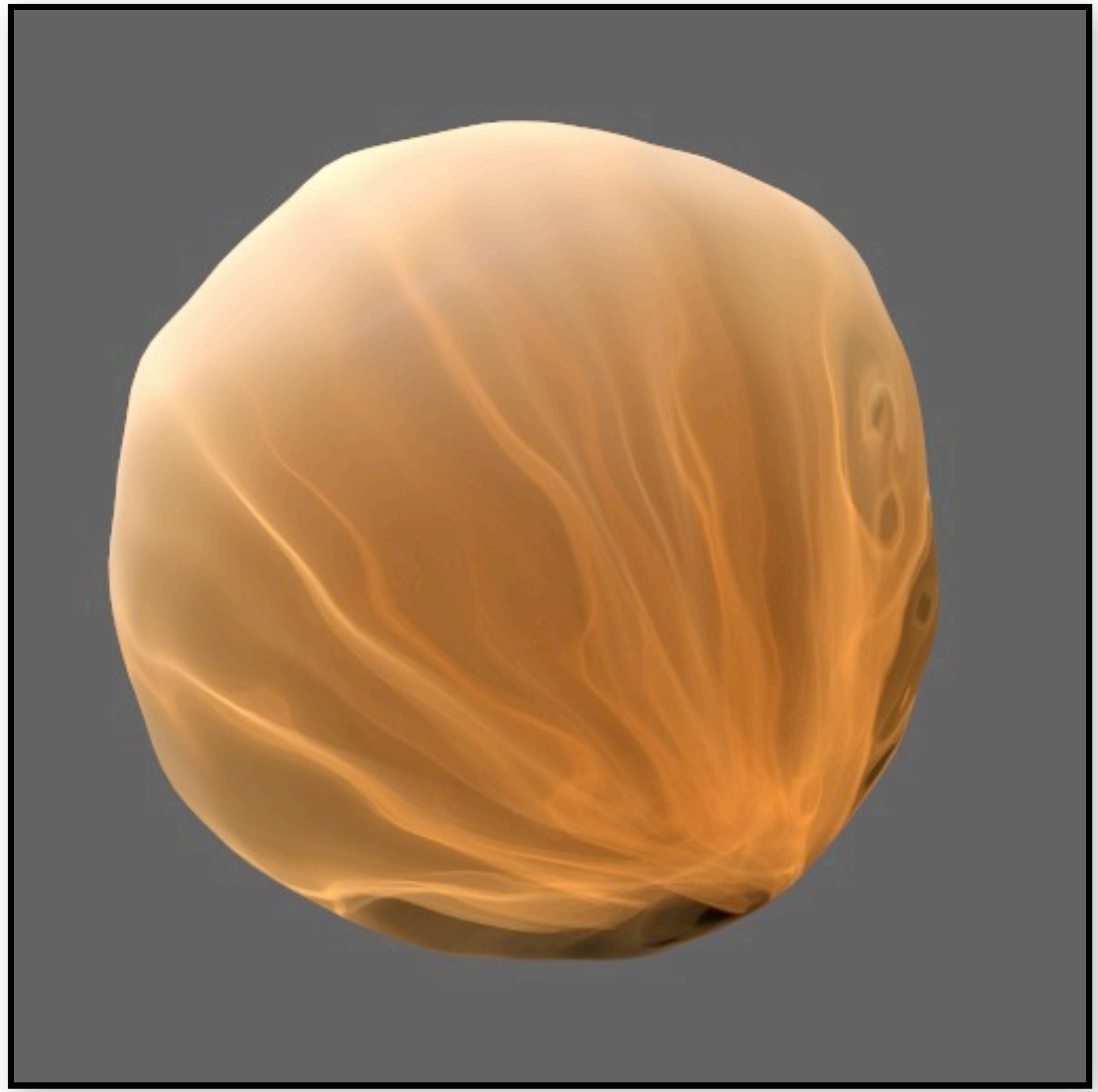
- Need to intersect each ray with all photon beams (expensive!)
- Place photon beams in an acceleration structure

Rendering

- Need to intersect each ray with all photon beams (expensive!)
- Place photon beams in an acceleration structure
- Rasterization (beams are just axial billboards!)

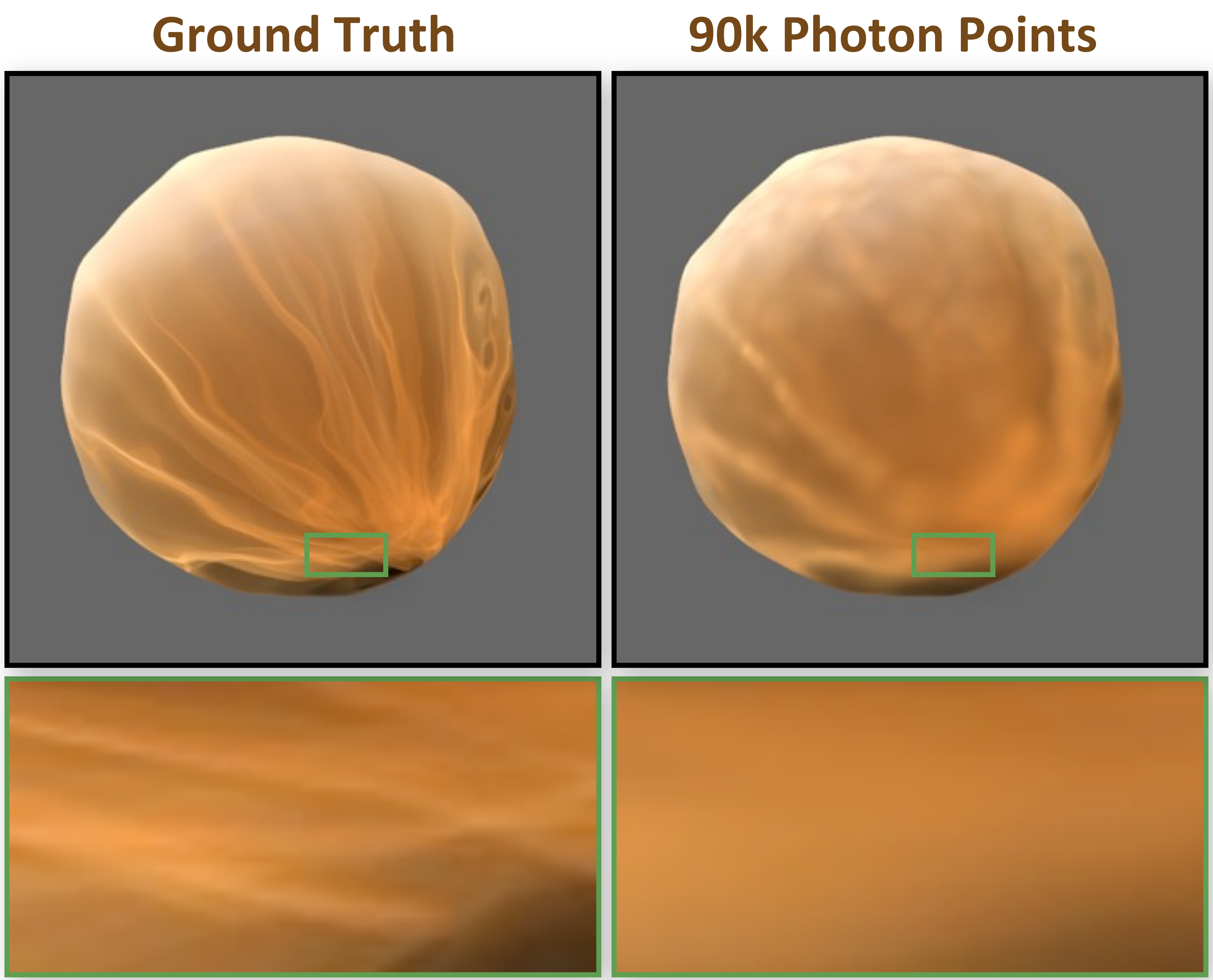
Bumpy Sphere

courtesy of Bruce Walter



Bumpy Sphere

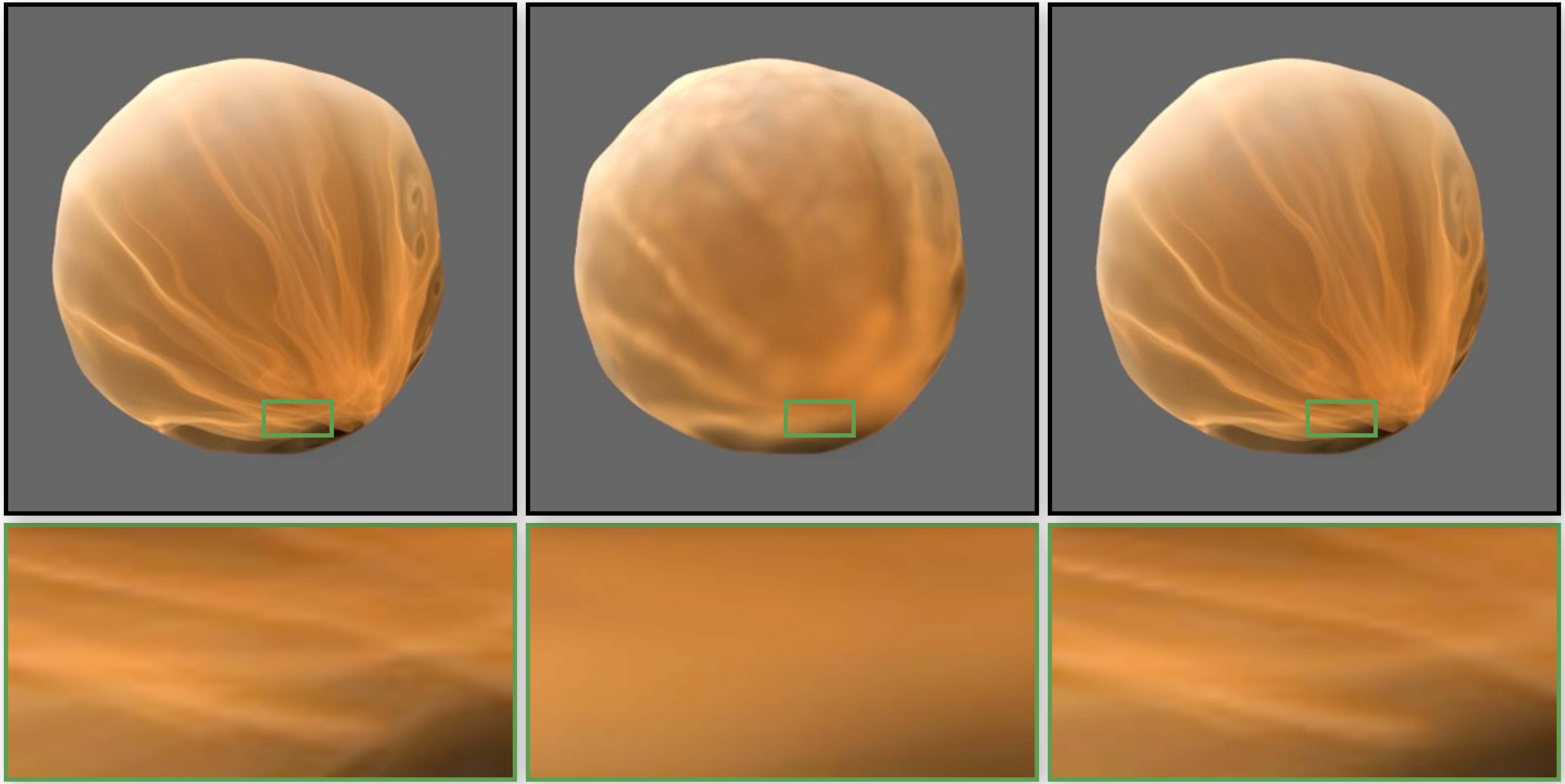
courtesy of Bruce Walter



Bumpy Sphere

courtesy of Bruce Walter

Ground Truth	90k Photon Points	90k Photon Beams

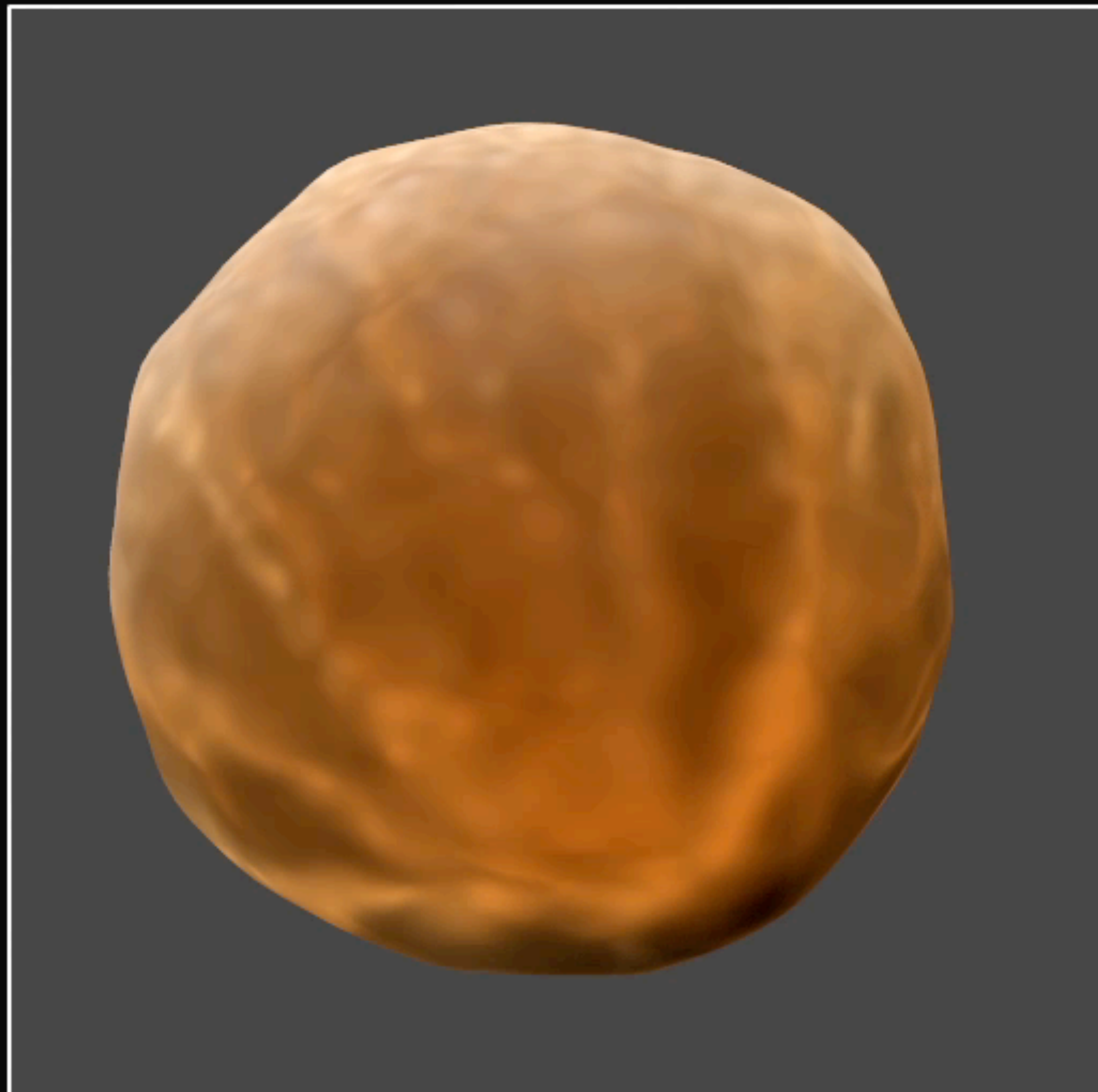


Bumpy Sphere

Rendered at 512x512 with up to 16 samples/pixel

Equal Photon Count

Photon Points



90K Photon **Points**
~ 40 seconds/frame

Photon Beams

90K Photon **Beams**
~ 103 seconds/frame

Equal Render Time

Photon Points

Photon Beams

1.3M Photon **Points**
~ 101 seconds/frame

90K Photon **Beams**
~ 103 seconds/frame

Lighthouse

Photon Points

10K Photon **Points**
~ 31 seconds/frame

Photon Beams

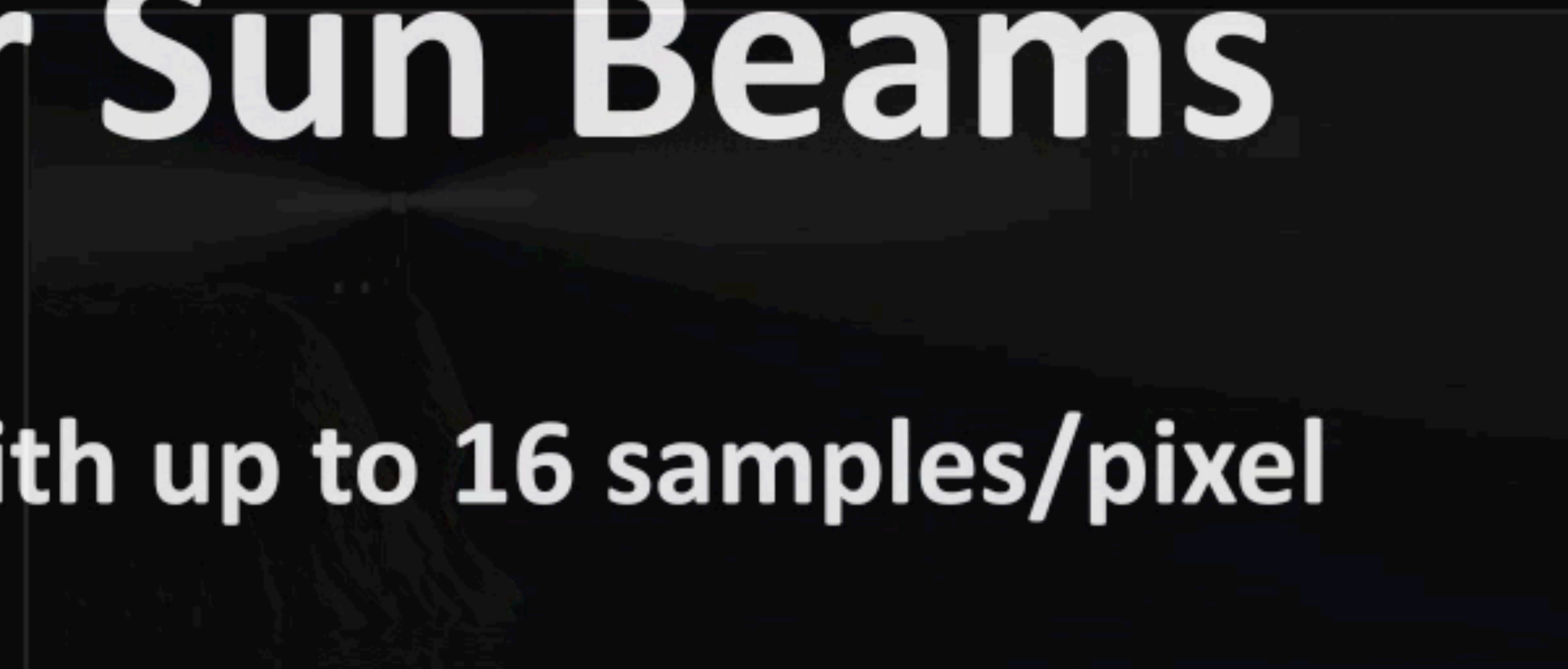
700 Photon **Beams**
~ 25 seconds/frame

Roughly Equal Time

Lighthouse

Jarosz et al. '08

Our Method



Underwater Sun Beams

Rendered at 1024x576 with up to 16 samples/pixel

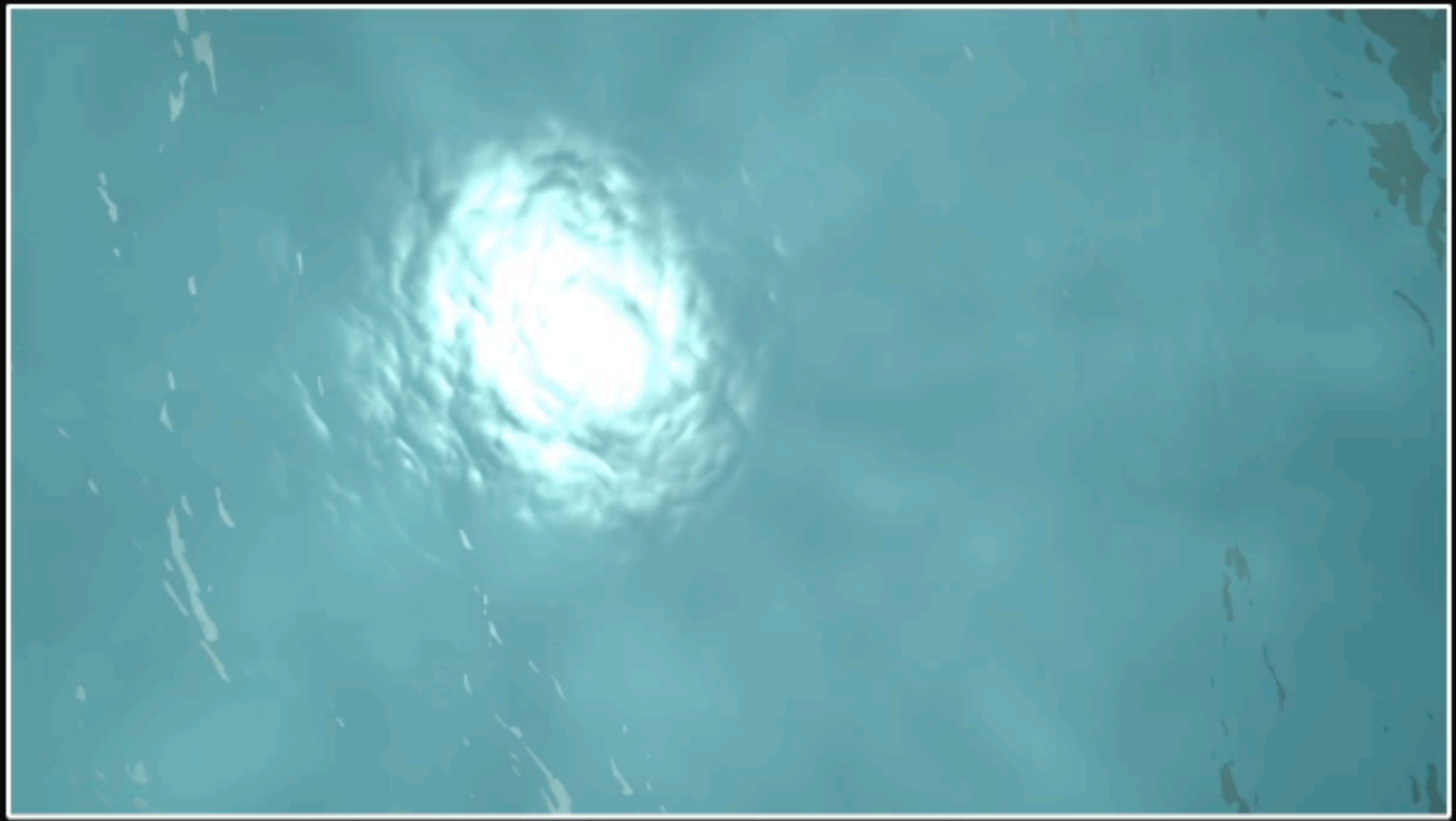
1M Photon **Points**
~ 226 seconds/frame

9x Render Time

700 Photon **Beams**
~ 25 seconds/frame

Underwater Sun Beams

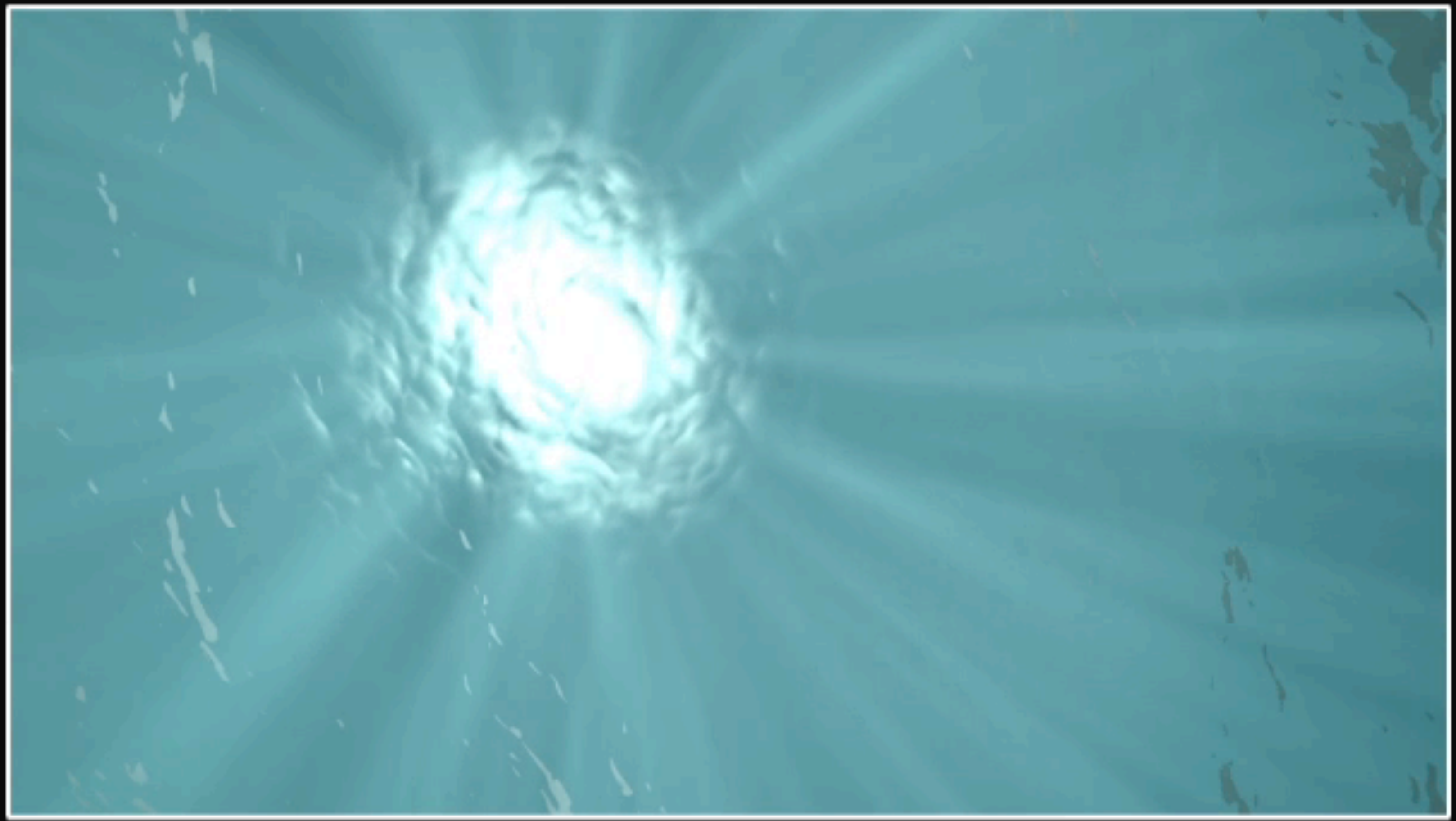
Photon Points



100K Photon **Points**
~ 204 seconds/frame

Roughly Equal Time

Photon Beams



25K Photon **Beams**
~ 200 seconds/frame

Progressive Photon Beams

Progressive Photon Beams

- Combine benefits of:

Progressive Photon Beams

- Combine benefits of:
  - photon beams

Progressive Photon Beams

- Combine benefits of:
  - photon beams
  - progressive photon mapping

Statistical Convergence

- Previous derivations not directly applicable
  - beam density vs. point density

Statistical Convergence

- Previous derivations not directly applicable
  - beam density vs. point density
- Reduction factor: $f_i = \dfrac{i+\alpha}{i+1}$

Statistical Convergence

- Previous derivations not directly applicable
  - beam density vs. point density
- Reduction factor: $f_i = \frac{i + \alpha}{i + 1}$
- *Application* of factor depends on blur dimensionality
  - Surfaces (2D): $r_{i+1}^2 = f_i \cdot r_i^2$

Statistical Convergence

- Previous derivations not directly applicable
  - beam density vs. point density
- Reduction factor: $f_i = \frac{i + \alpha}{i + 1}$
- *Application* of factor depends on blur dimensionality
  - Surfaces (2D): $r_{i+1}^2 = f_i \cdot r_i^2$
  - Volumetric photon mapping (3D): $r_{i+1}^3 = f_i \cdot r_i^3$

Statistical Convergence

- Previous derivations not directly applicable
  - beam density vs. point density
- Reduction factor: $f_i = \frac{i + \alpha}{i + 1}$
- *Application* of factor depends on blur dimensionality
  - Surfaces (2D): $r_{i+1}^2 = f_i \cdot r_i^2$
  - Volumetric photon mapping (3D): $r_{i+1}^3 = f_i \cdot r_i^3$
  - Beam × Beam (1D): $r_{i+1} = f_i \cdot r_i$

Algorithm

Algorithm

Step 1:

- Photon tracing: emit, scatter, store beams
- Scale beam widths by global factor r_i

Algorithm

Step 1:

- Photon tracing: emit, scatter, store beams
- Scale beam widths by global factor r_i

Step 2:

- Trace random camera path, evaluate radiance estimate along each ray using beams

Algorithm

Step 1:

- Photon tracing: emit, scatter, store beams
- Scale beam widths by global factor r_i

Step 2:

- Trace random camera path, evaluate radiance estimate along each ray using beams

- Display running average

Algorithm

Step 1:

- Photon tracing: emit, scatter, store beams
- Scale beam widths by global factor r_i

Step 2:

- Trace random camera path, evaluate radiance estimate along each ray using beams

- Display running average
- *Reduce* global factor r_i and *repeat*

Algorithm

Step 1:

- Photon tracing: emit, scatter, store beams
- Scale beam widths by global factor r_i

Step 2:

Trivially Parallelizable

- Trace random camera path, evaluate radiance estimate along each ray using beams

- Display running average
- *Reduce* global factor r_i and *repeat*

Evaluating the Transmittance

- Need to compute transmittance: along photon beam, along camera ray

Evaluating the Transmittance

- Need to compute transmittance: along photon beam, along camera ray
- Homogeneous: analytic

Evaluating the Transmittance

- Need to compute transmittance: along photon beam, along camera ray
- Homogeneous: analytic
- Heterogeneous: use progressive deep shadow maps

Results & Implementation

- 3 implementations:
  - GPU-only OptiX ray-tracer
  - GPU-only rasterization
  - General: Hybrid CPU/GPU

Results & Implementation

- 3 implementations:
  - GPU-only OptiX ray-tracer
  - GPU-only rasterization
  - General: Hybrid CPU/GPU

BumpySphere

scene courtesy of Bruce Walter

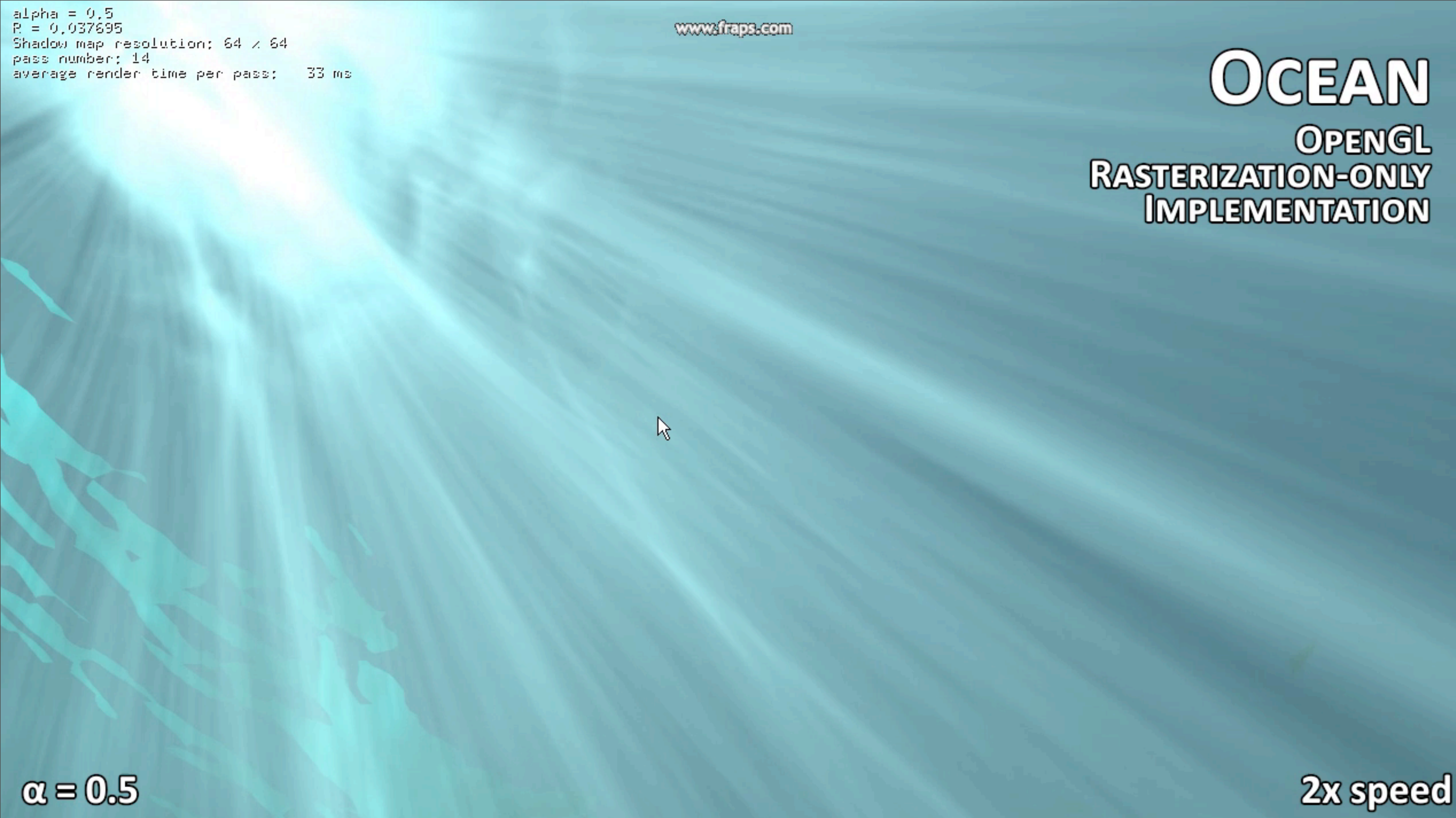
OptiX Implementation

alpha: 0.60 beams per pass: 1024 pass number: 1 render time per pass: 132.97 ms

2x speed

Results & Implementation

- 3 implementations:
  - GPU-only OptiX **ray-tracer**
  - GPU-only **rasterization**
  - General: Hybrid CPU/GPU


alpha = 0.5
R = 0.037695
Shadow map resolution: 64 x 64
pass number: 14
average render time per pass: 33 ms
www.fraps.com
OCEAN
OPENGL
RASTERIZATION-ONLY
IMPLEMENTATION
α = 0.5
2x speed

Results & Implementation

- 3 implementations:
  - GPU-only OptiX ray-tracer
  - GPU-only rasterization
  - General: Hybrid CPU/GPU

Cars

1280x720, Depth-of-Field

Pass 1

Homogeneous

Heterogeneous

Pass 1

Average of Passes 1..1

Pass 2 | Average of Passes 1..2

Pass 4 | Average of Passes 1..4

Pass 8 | Average of Passes 1..8

Pass 16

Average of Passes 1..16

Thursday, August 23, 12

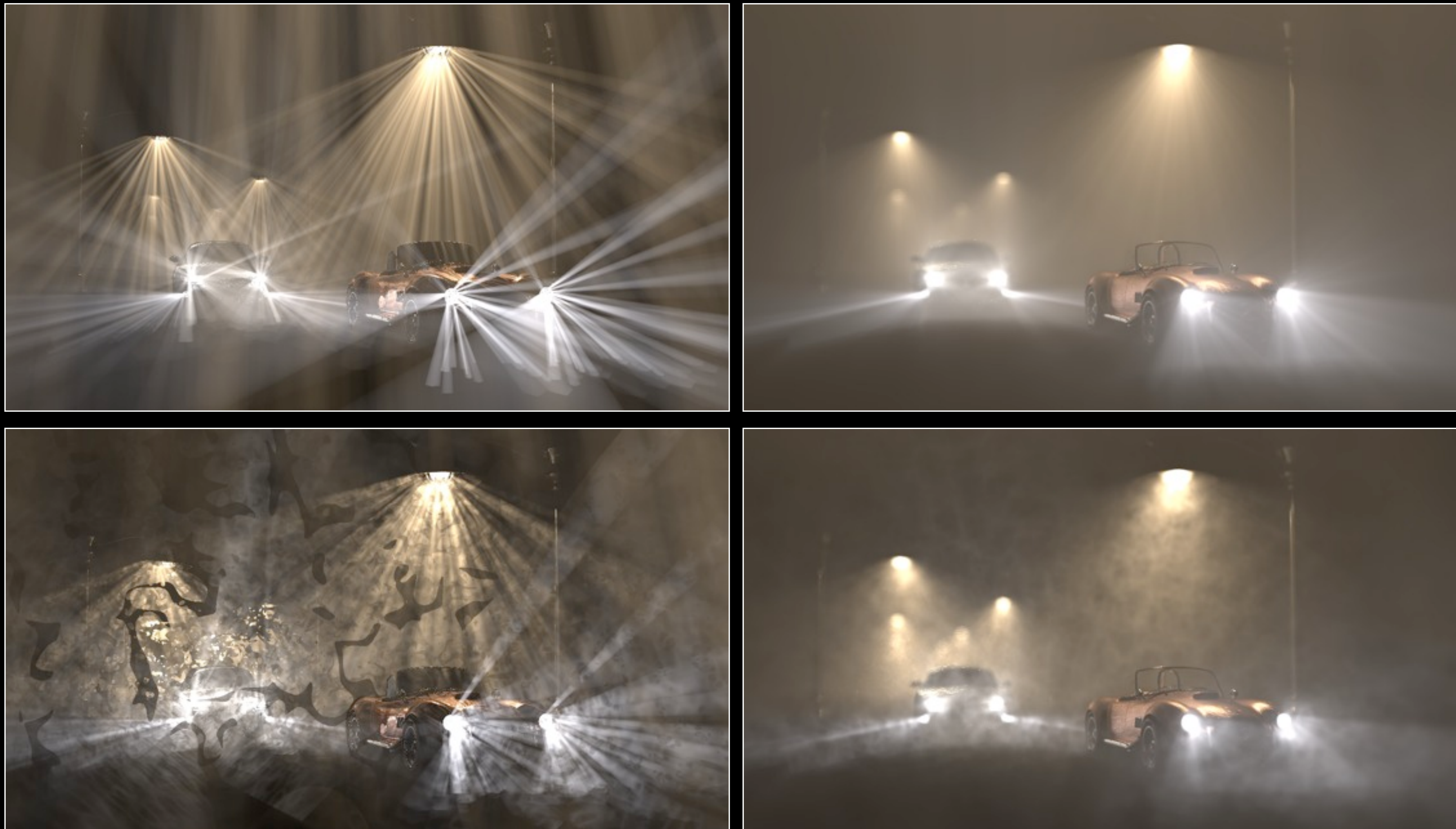
Pass 32	Average of Passes 1..32

Pass 64	Average of Passes 1..64

Pass 128 | Average of Passes 1..128

Pass 256

Average of Passes 1..256



Pass 512

Average of Passes 1..512

Pass 1024 | Average of Passes 1..1024

CARS

1280x720, Depth-of-Field

Homogeneous

14.55M Photon Beams

9.5 minutes

Heterogeneous

15.04M Photon Beams

16.8 minutes

CARS

1280x720, Depth-of-Field

Homogeneous

14.55M Photon Beams

9.5 minutes

Heterogeneous

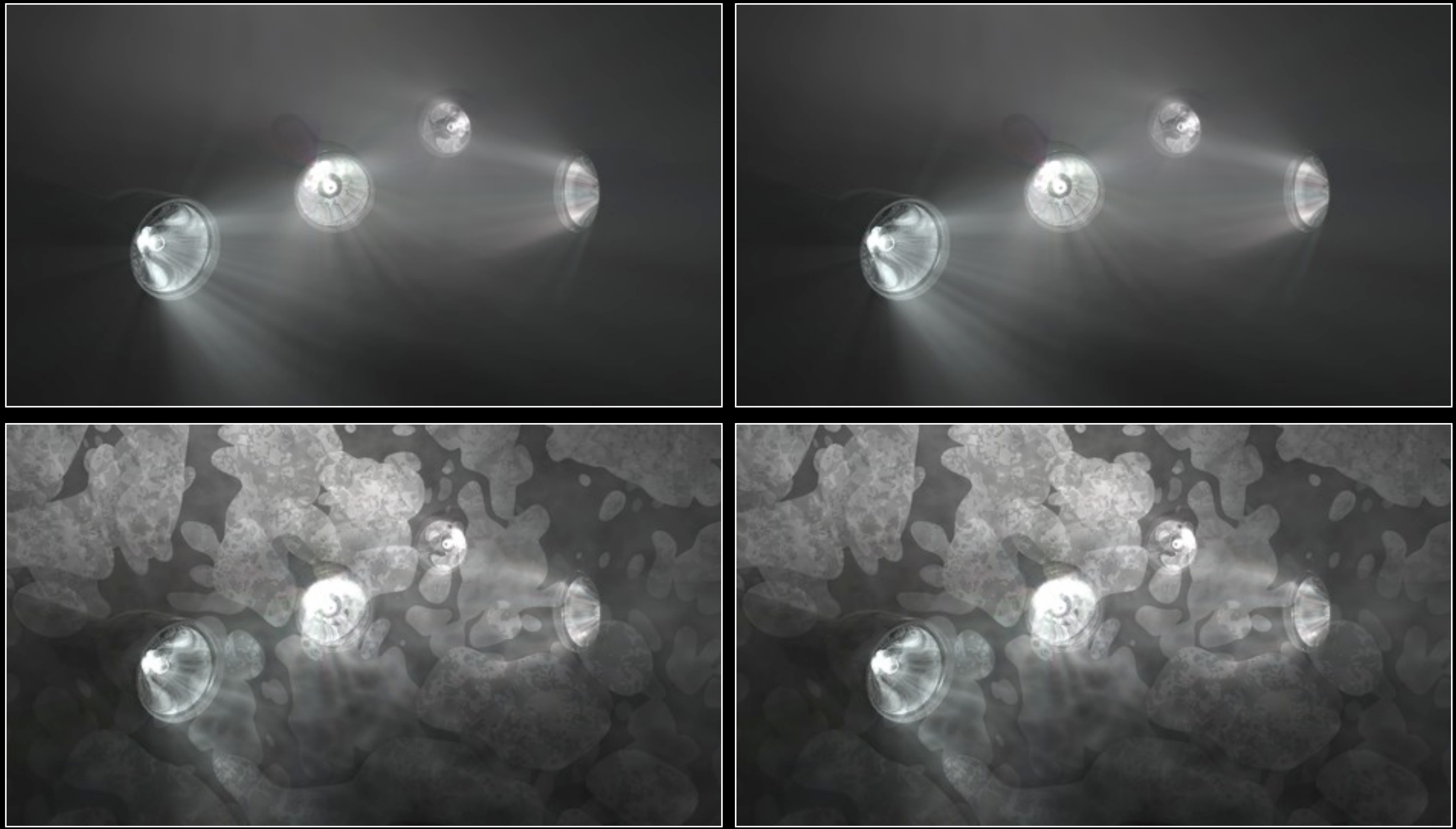
15.04M Photon Beams

16.8 minutes

FlashLights

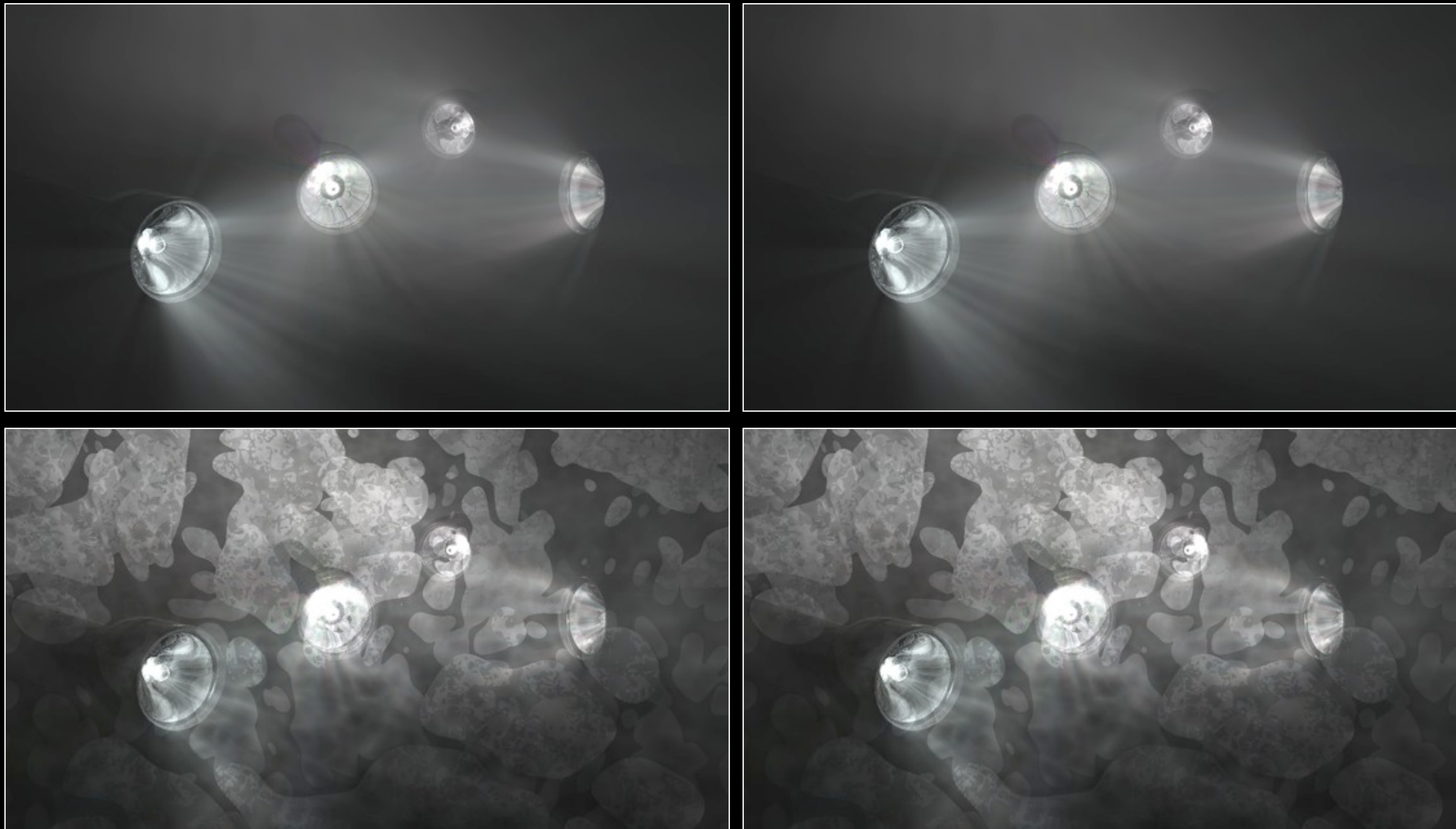
1280x720, Depth-of-Field

Pass 1 | Average of Passes 1..1

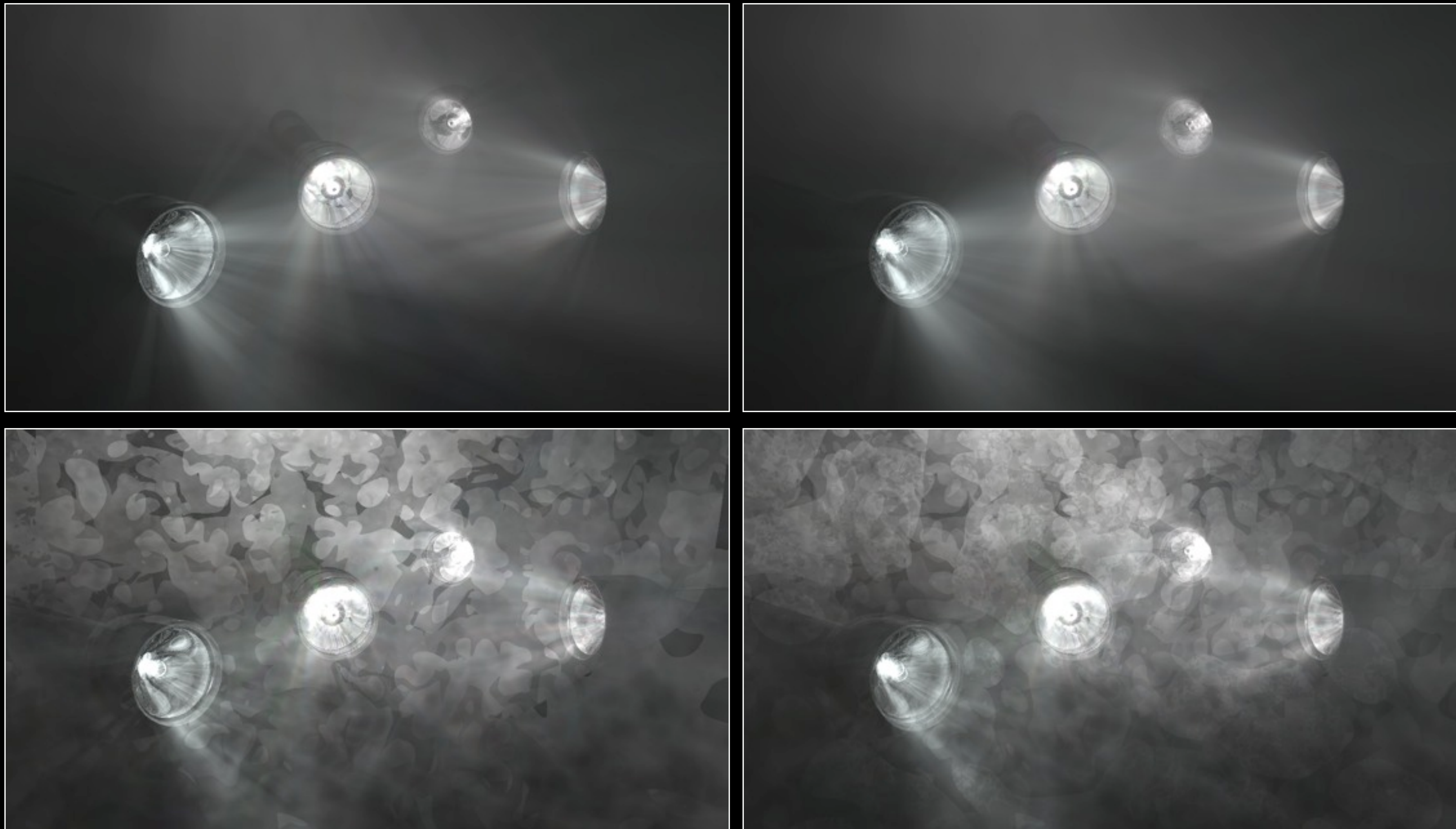


Pass 1

Average of Passes 1..1

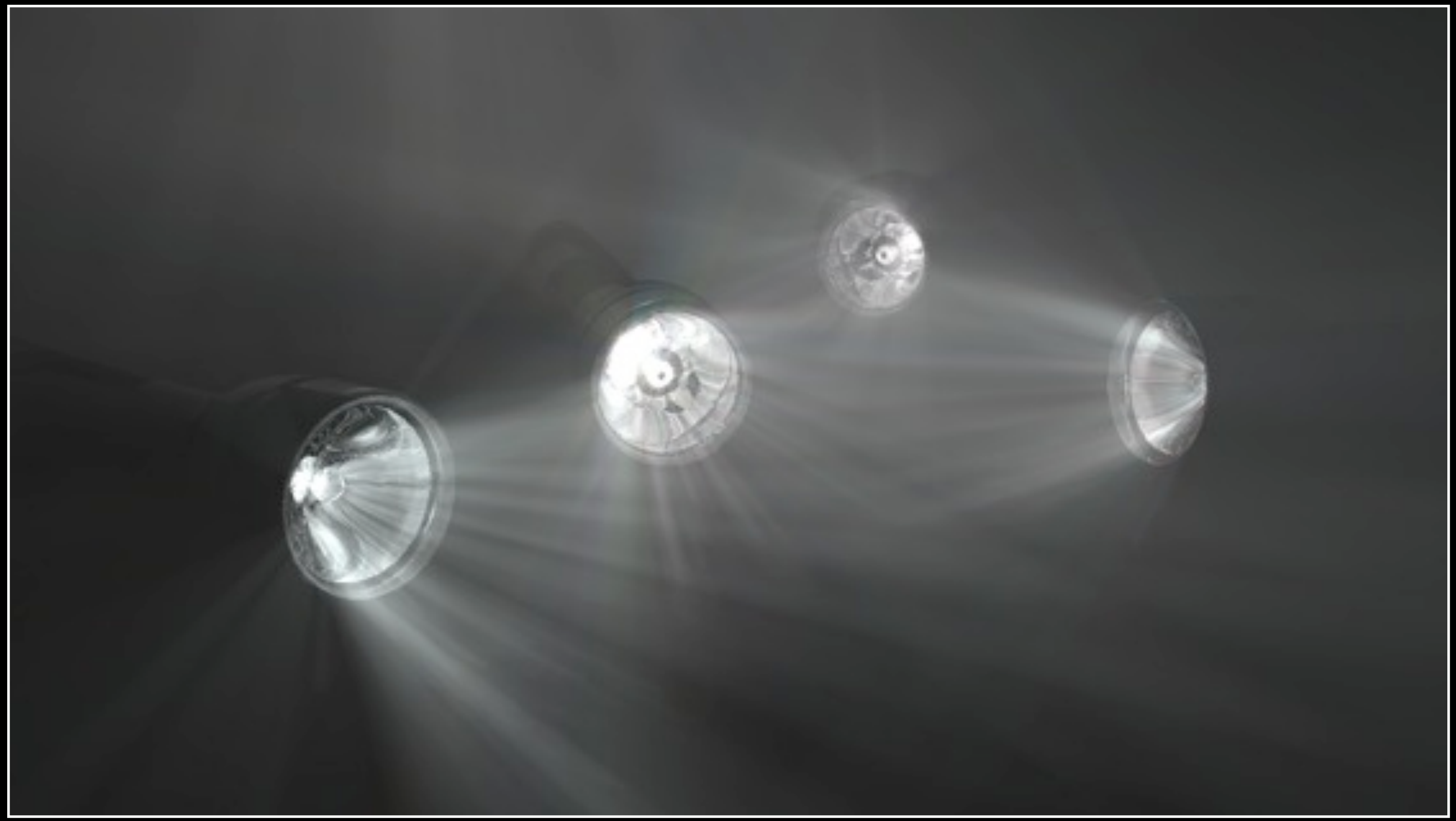


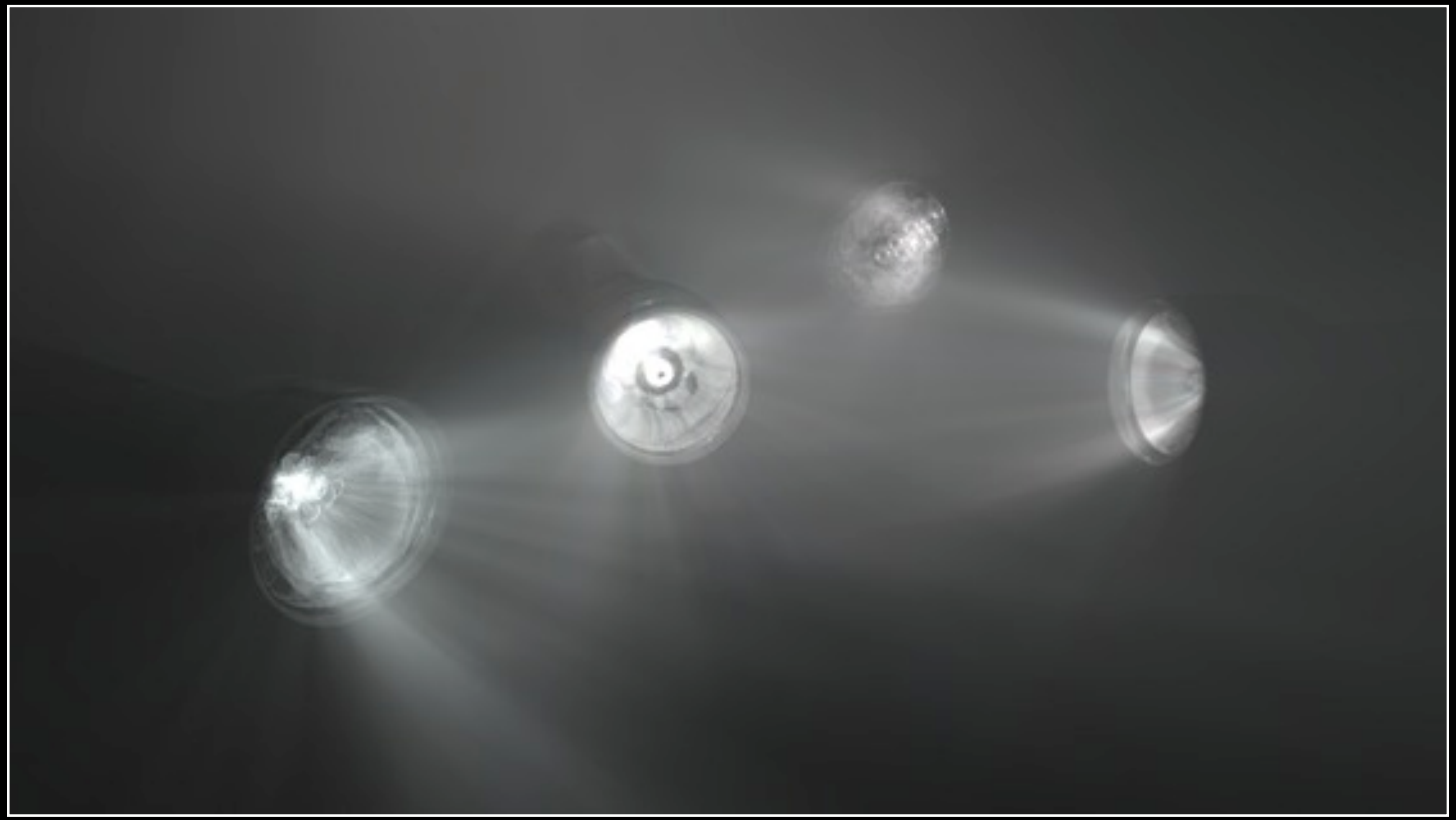
Pass 2 | Average of Passes 1..2

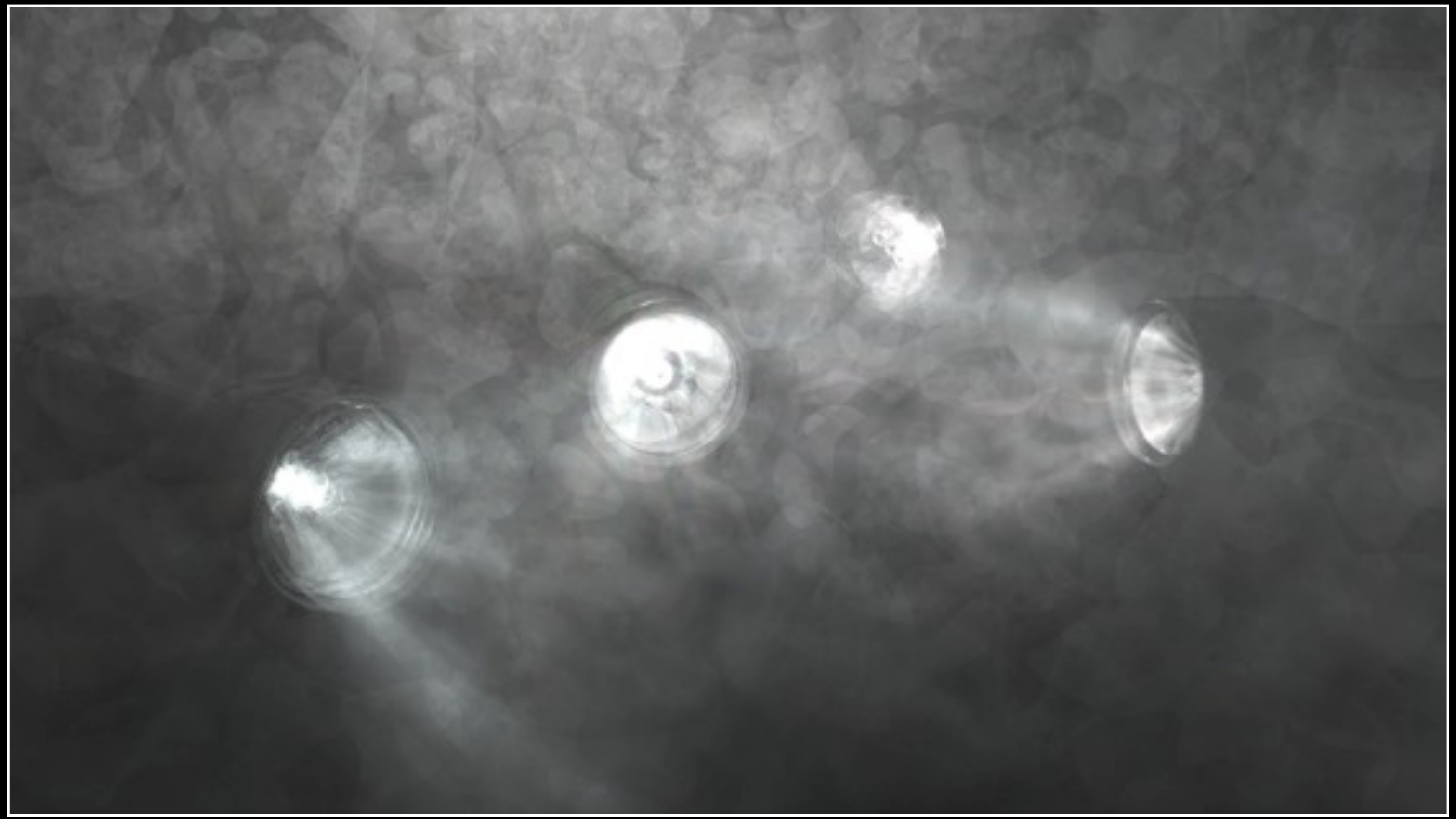


Pass 4

Average of Passes 1..4

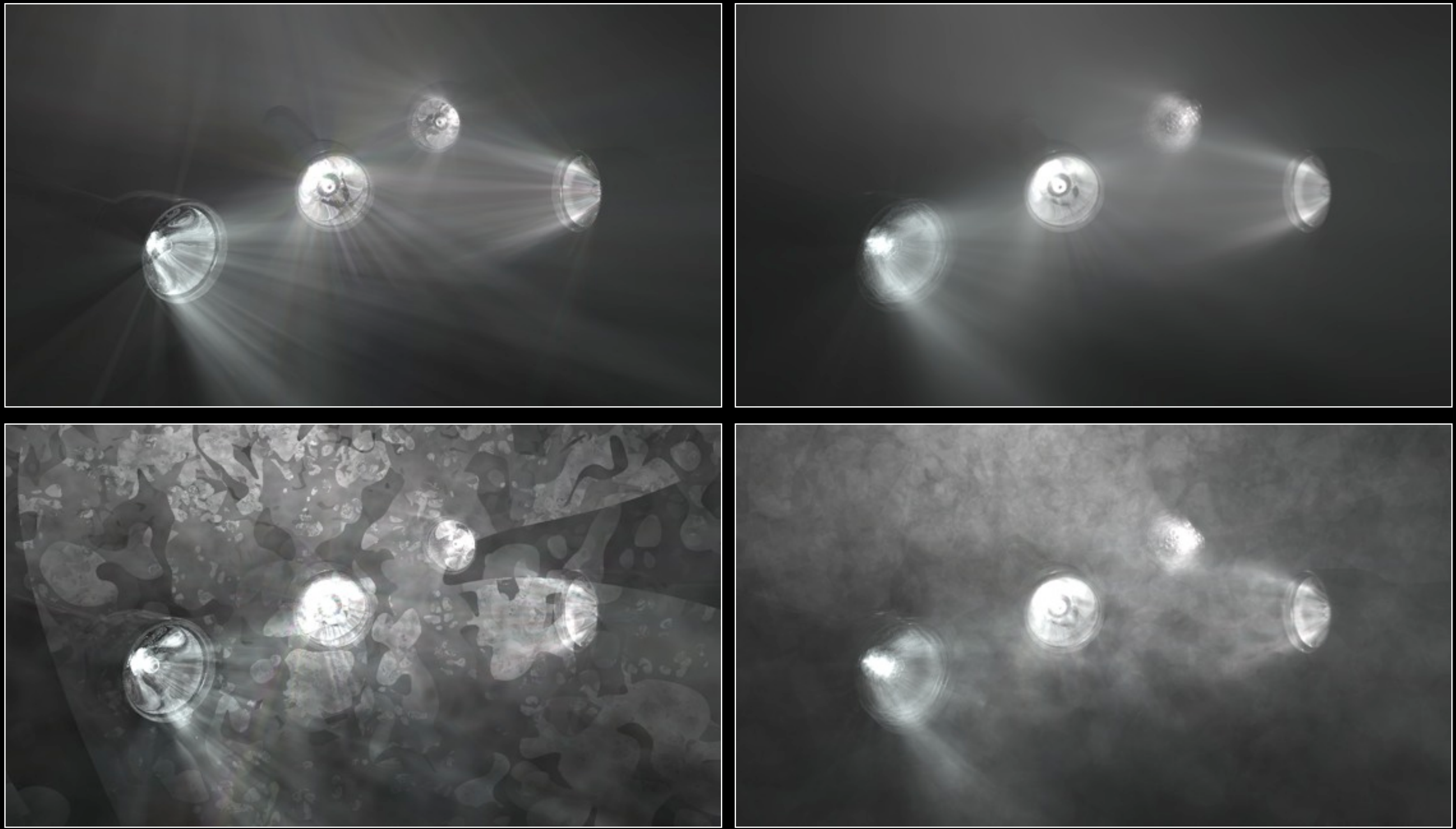




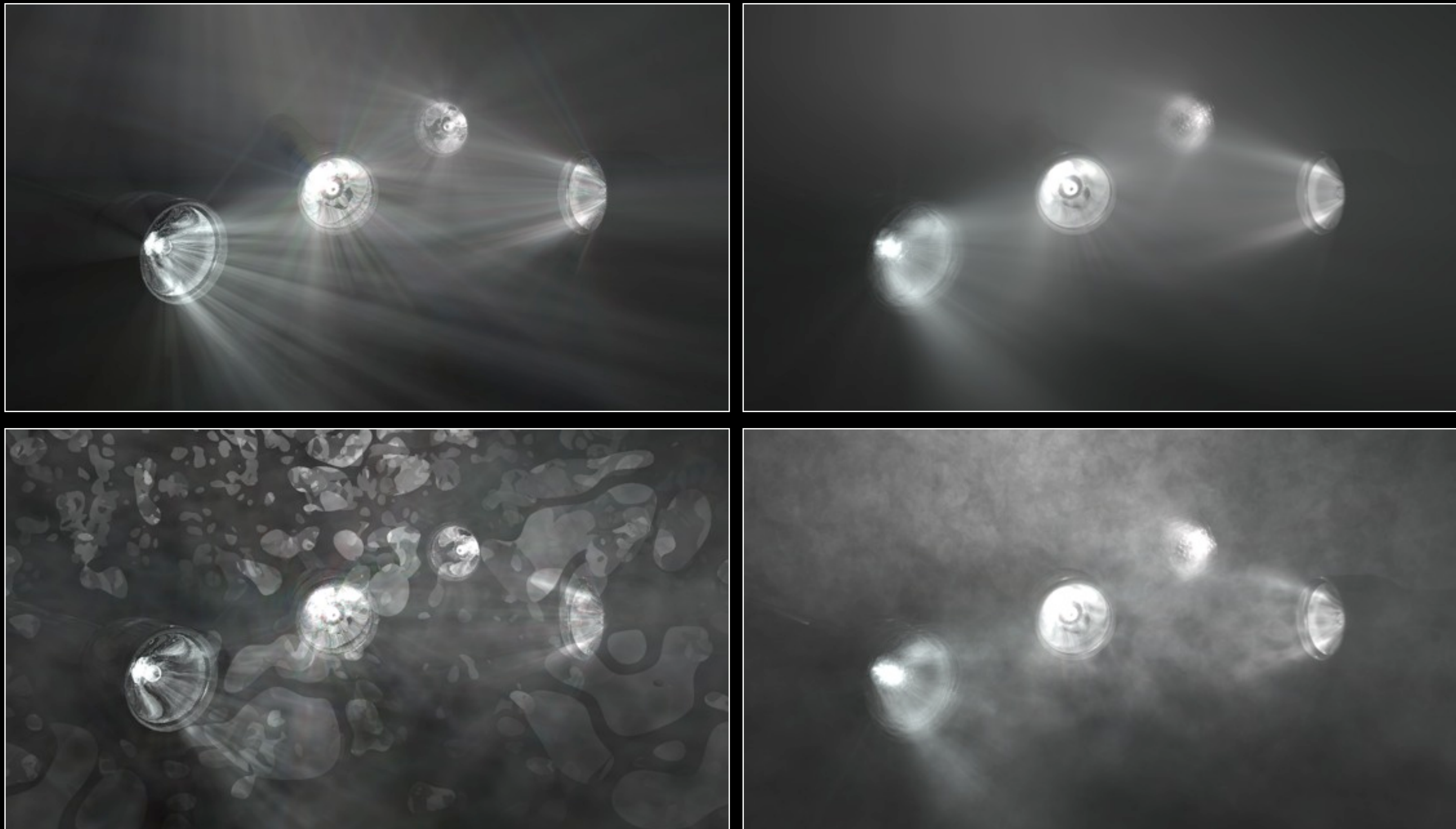


Pass 8

Average of Passes 1..8

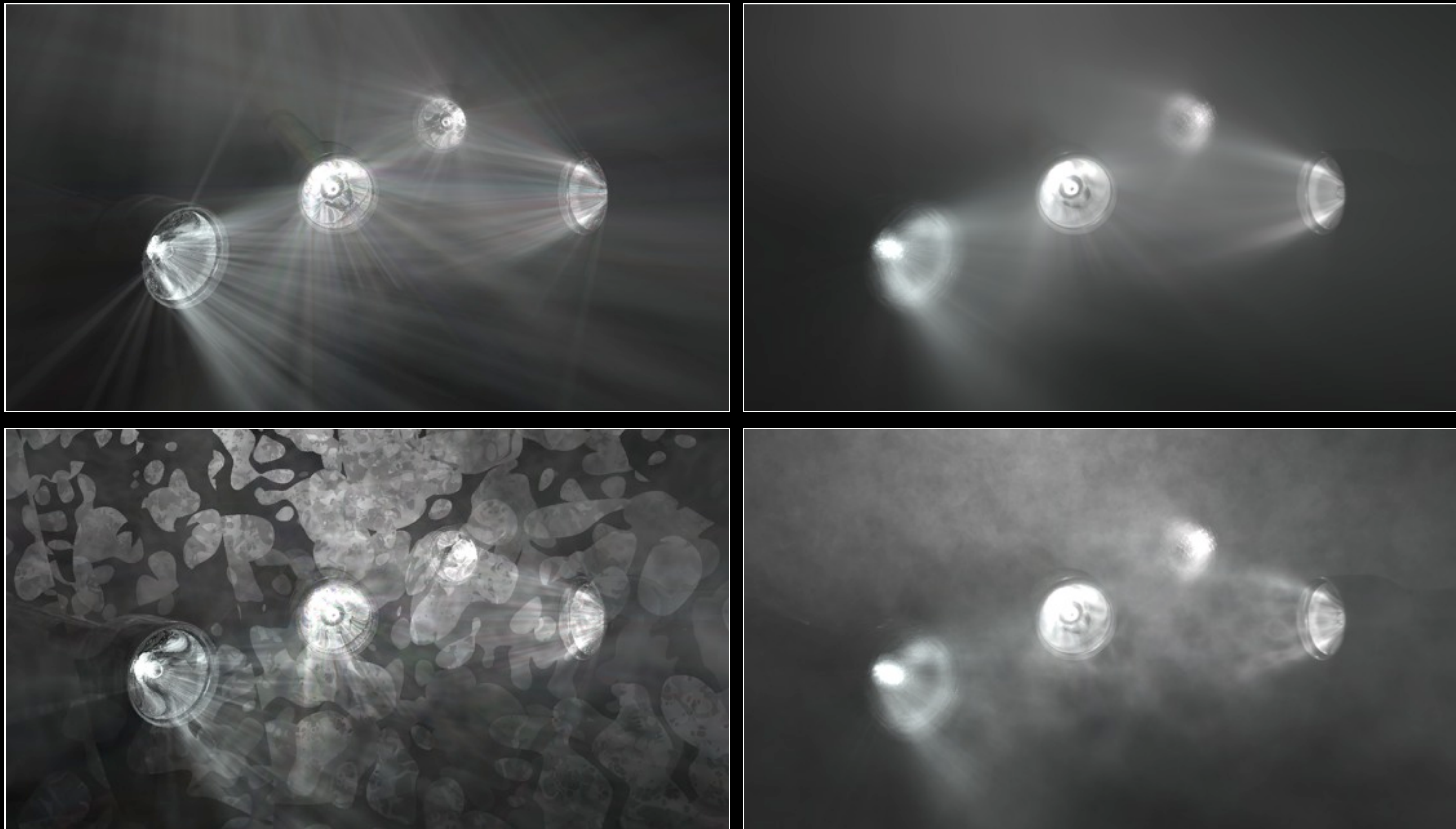


Pass 16	Average of Passes 1..16

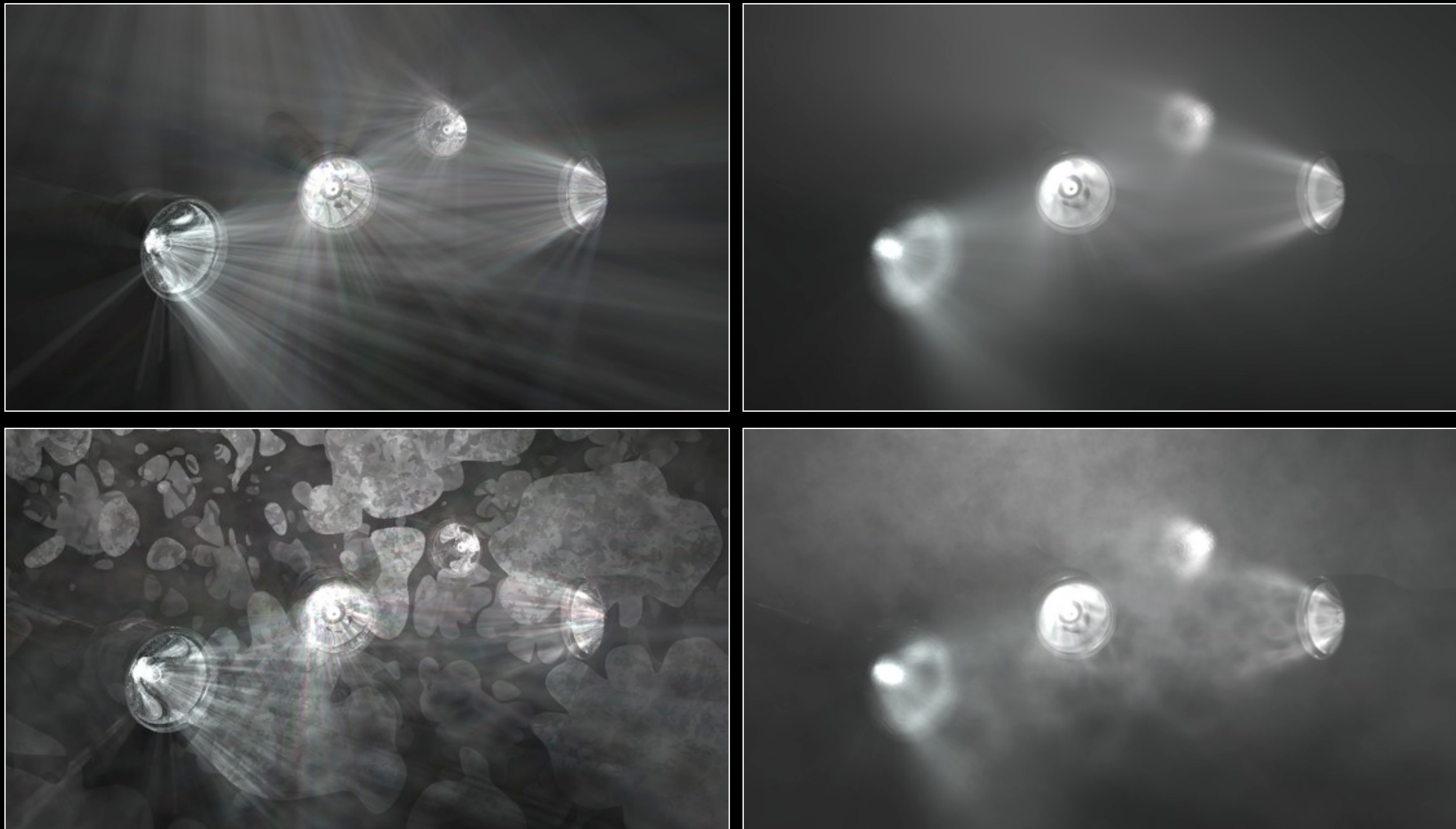


Pass 32

Average of Passes 1..32

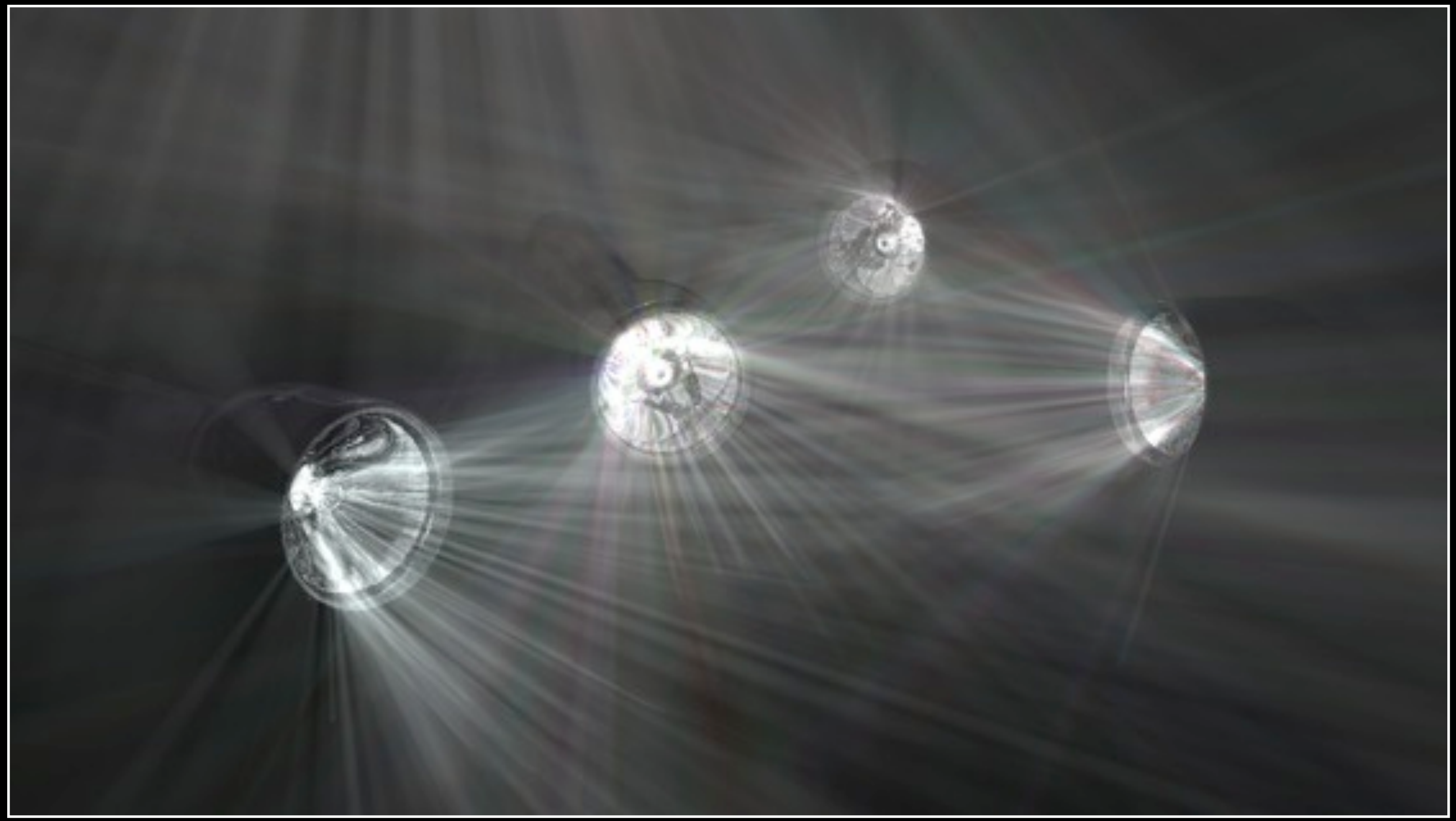


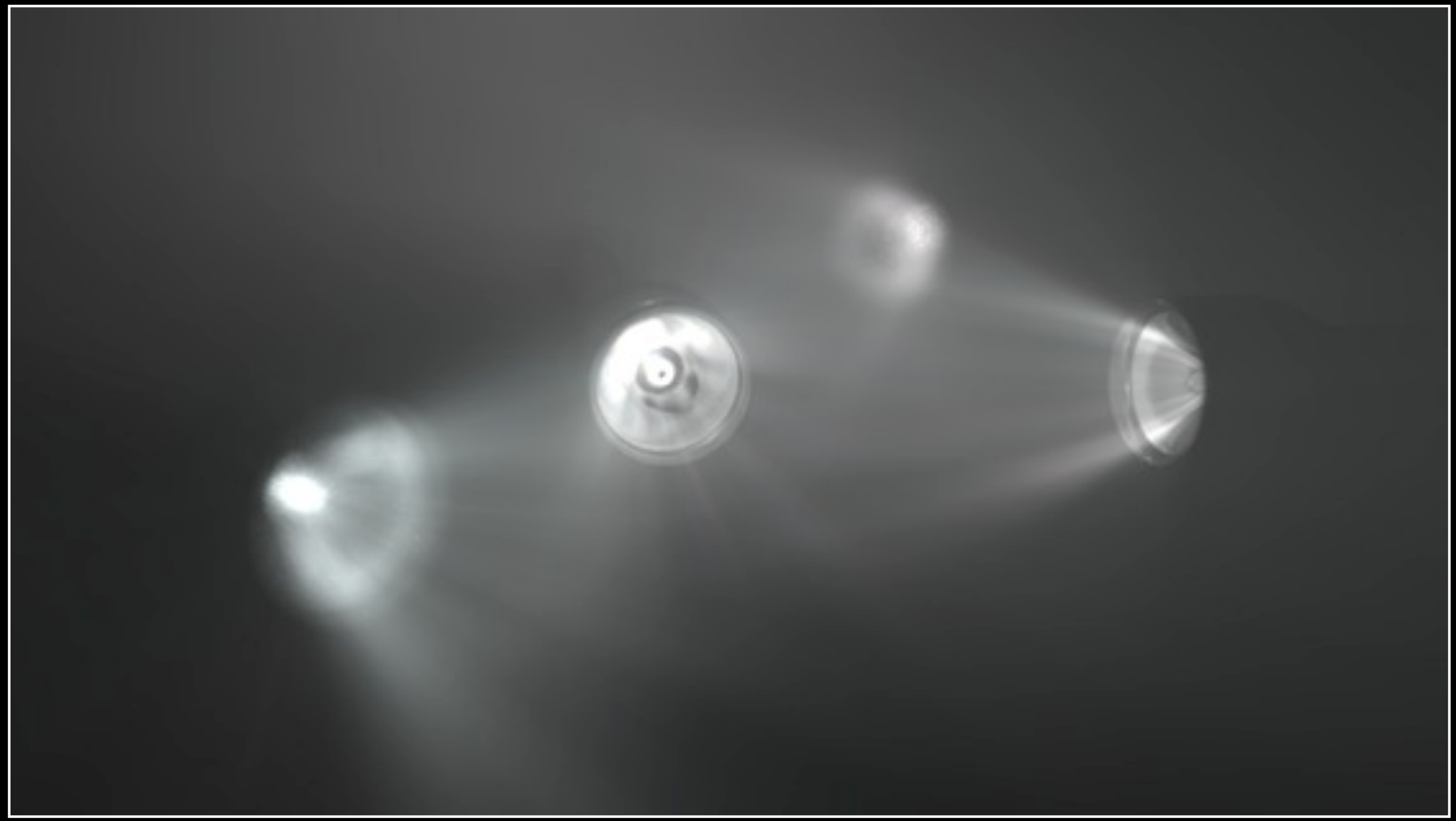
Pass 64 | Average of Passes 1..64



Pass 128

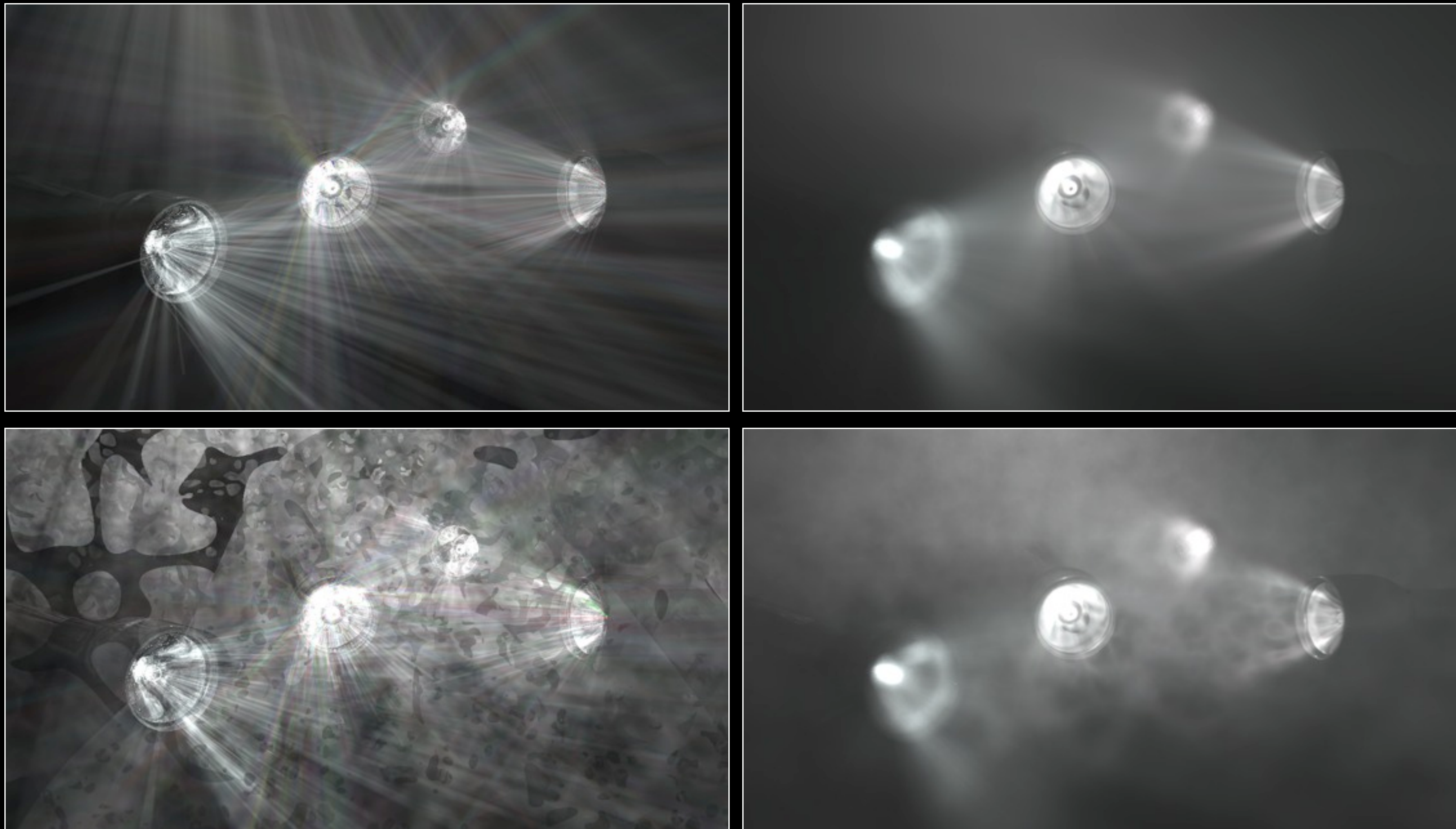
Average of Passes 1..128





Pass 256

Average of Passes 1..256



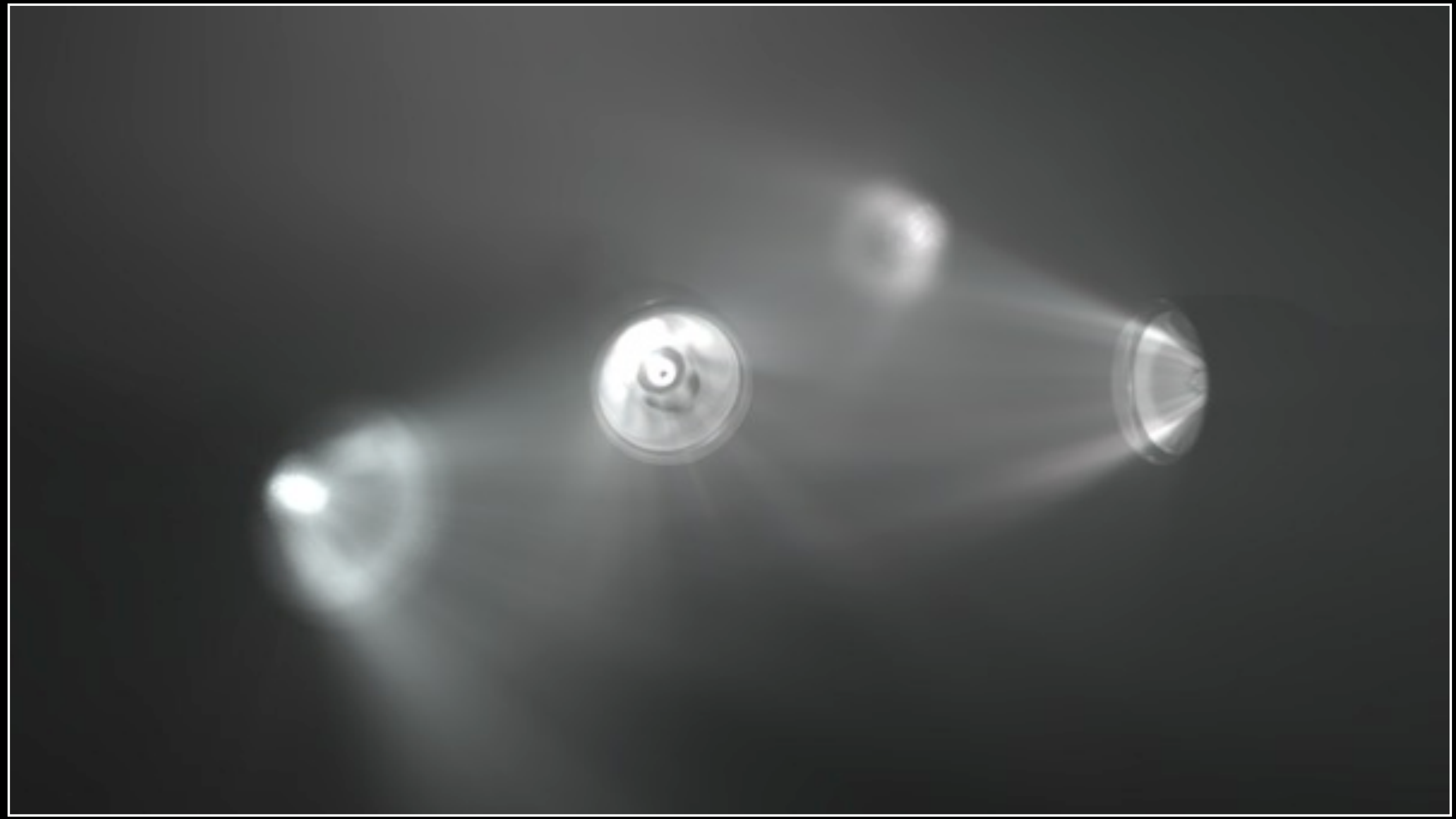
FLASHLIGHTS

1280x720, Depth-of-Field

Homogeneous

2.1M Photon Beams

8 minutes



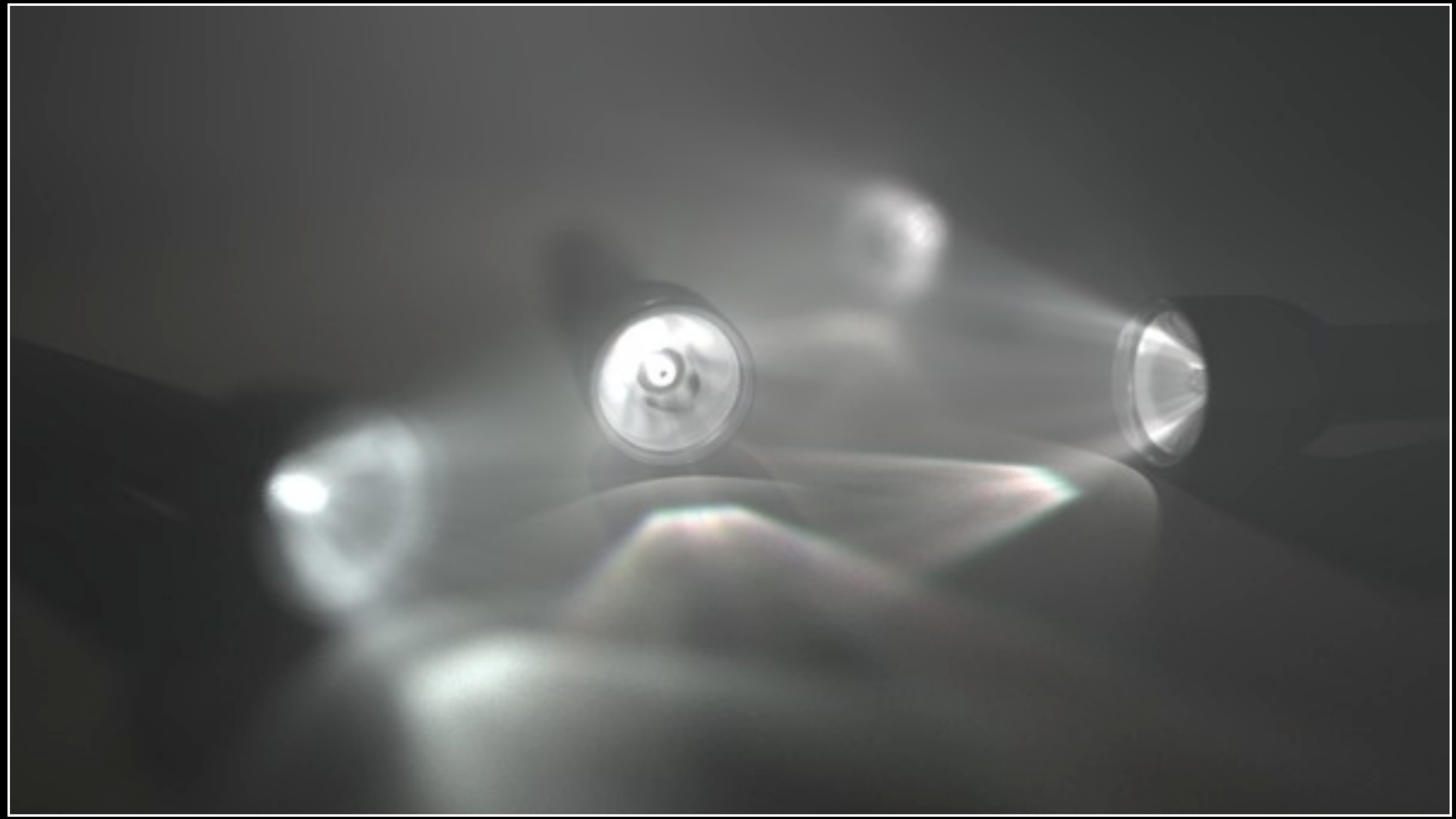
Heterogeneous

2.1M Photon Beams

10.8 minutes

FLASHLIGHTS

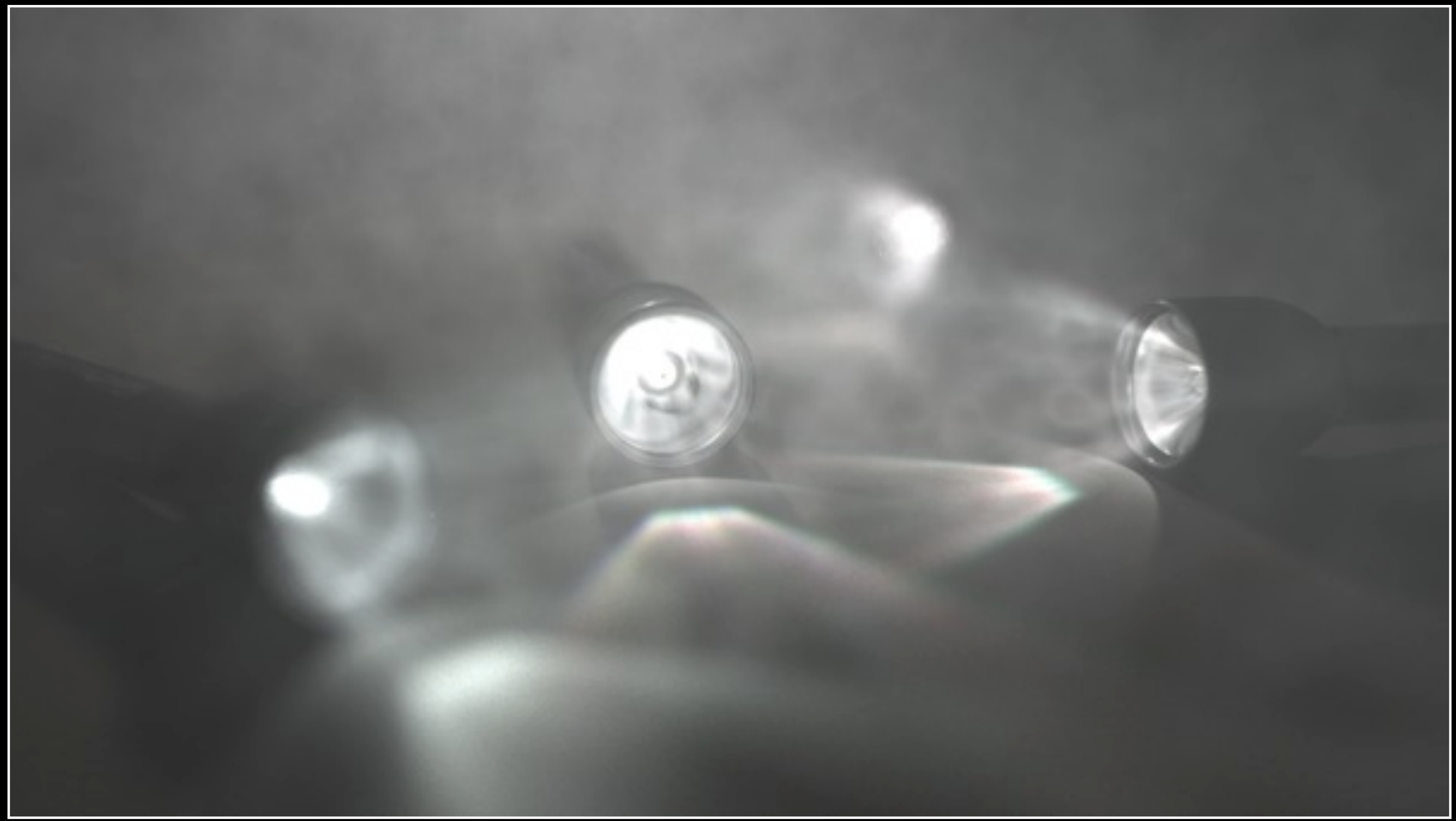
1280x720, Depth-of-Field



Homogeneous

2.1M Photon Beams

8 minutes



Heterogeneous

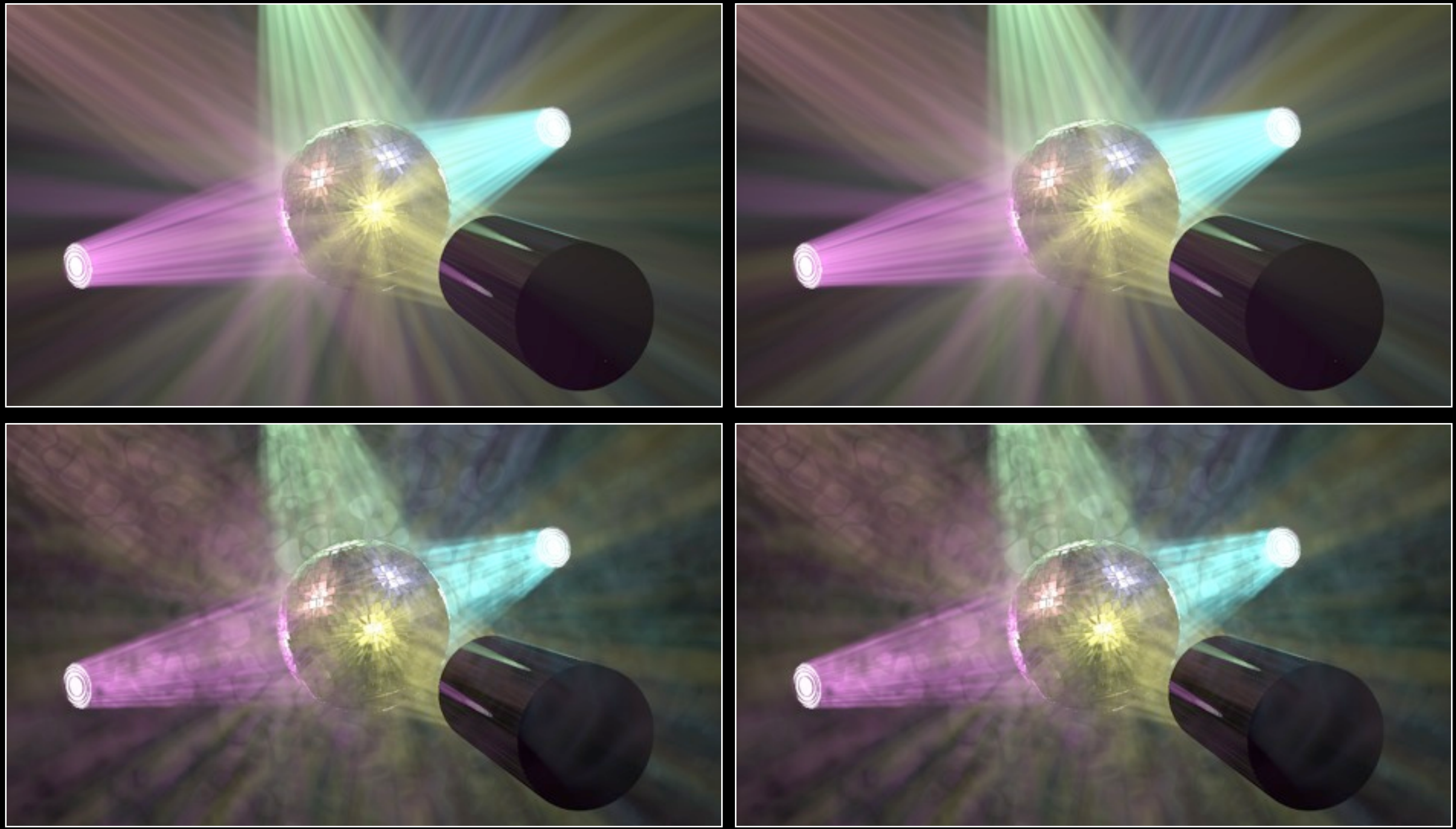
2.1M Photon Beams

10.8 minutes

DISCO

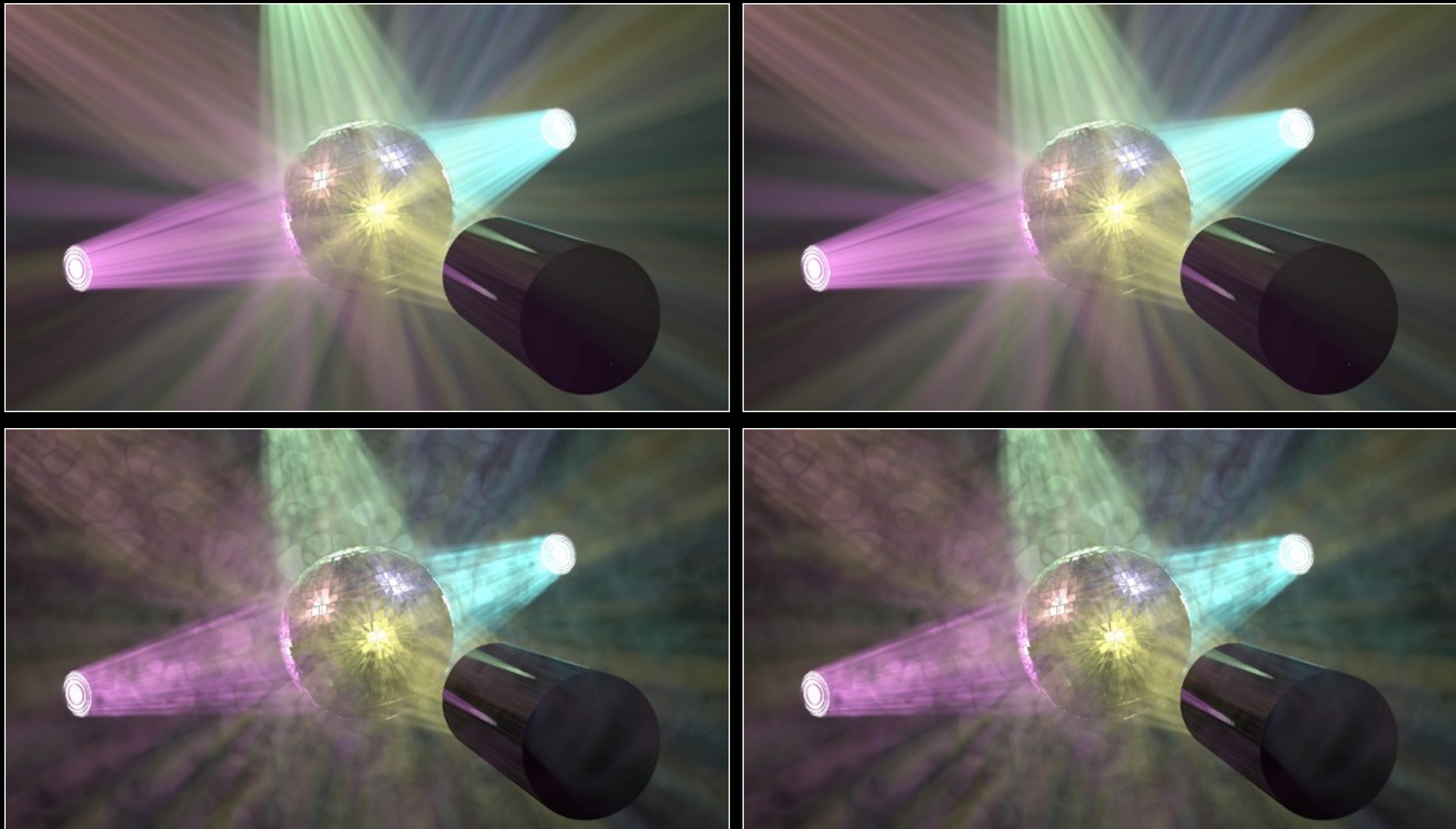
1280x720, Depth-of-Field

Pass 1 | Average of Passes 1..1



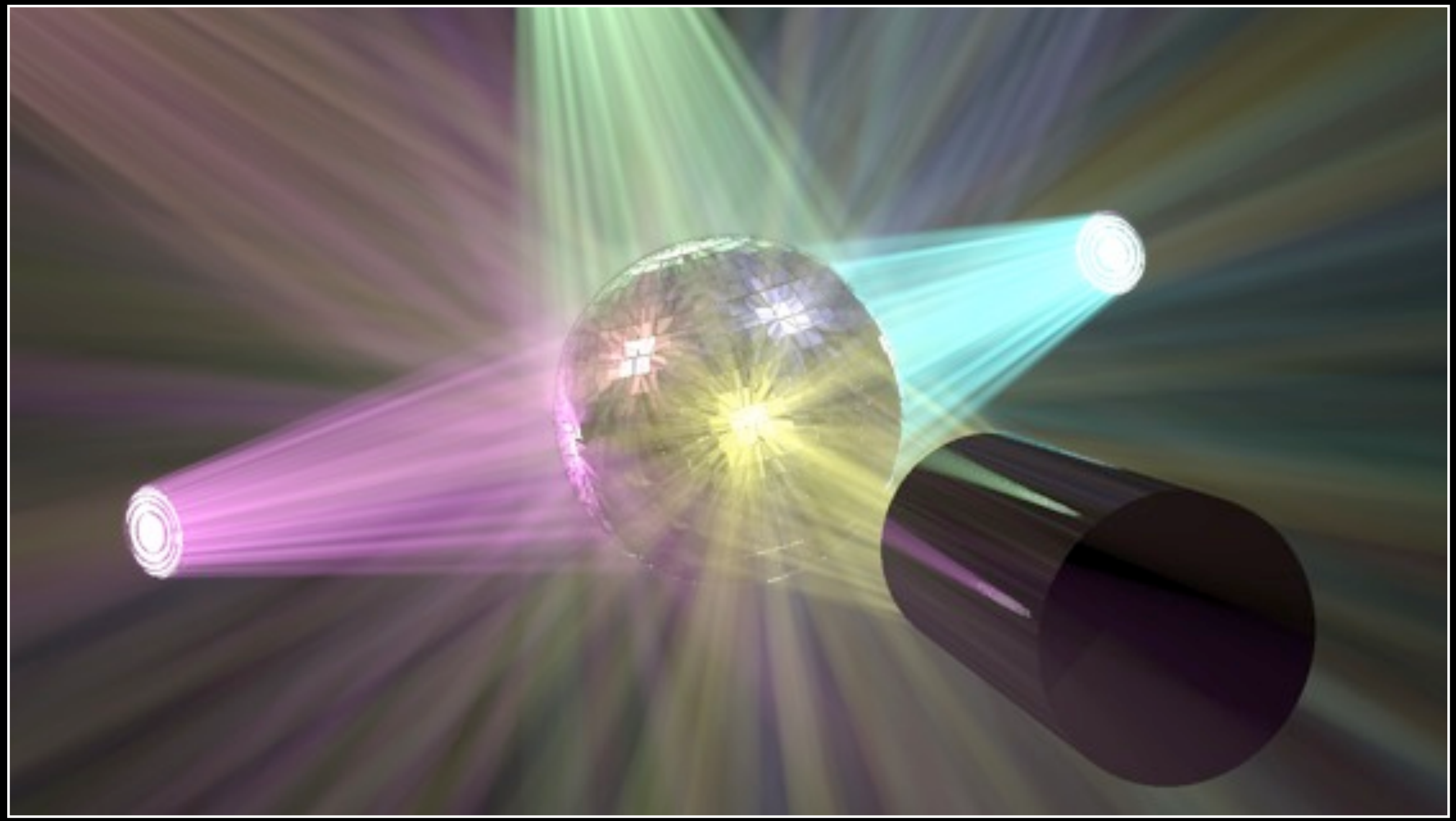
Pass 1

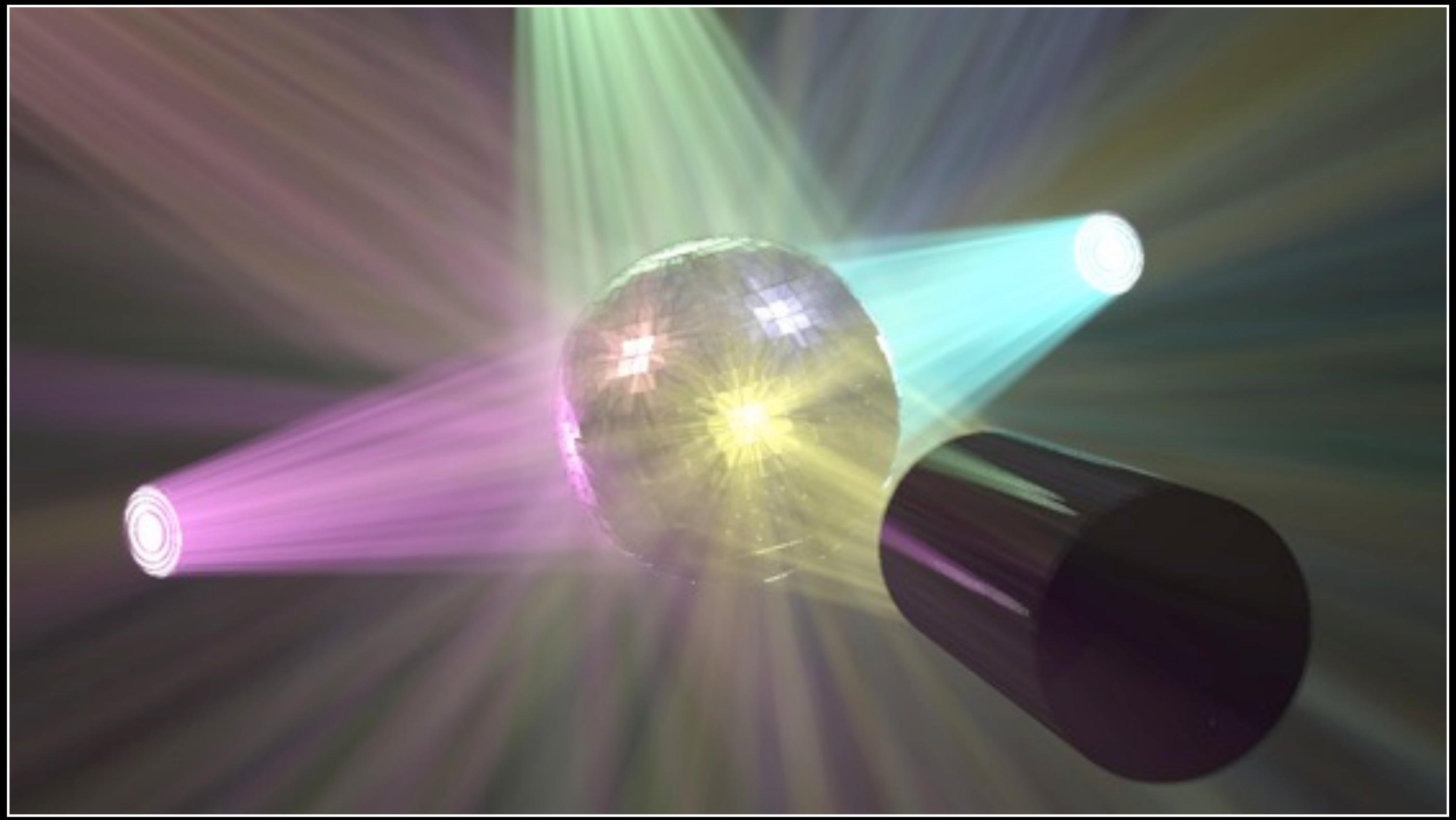
Average of Passes 1..1

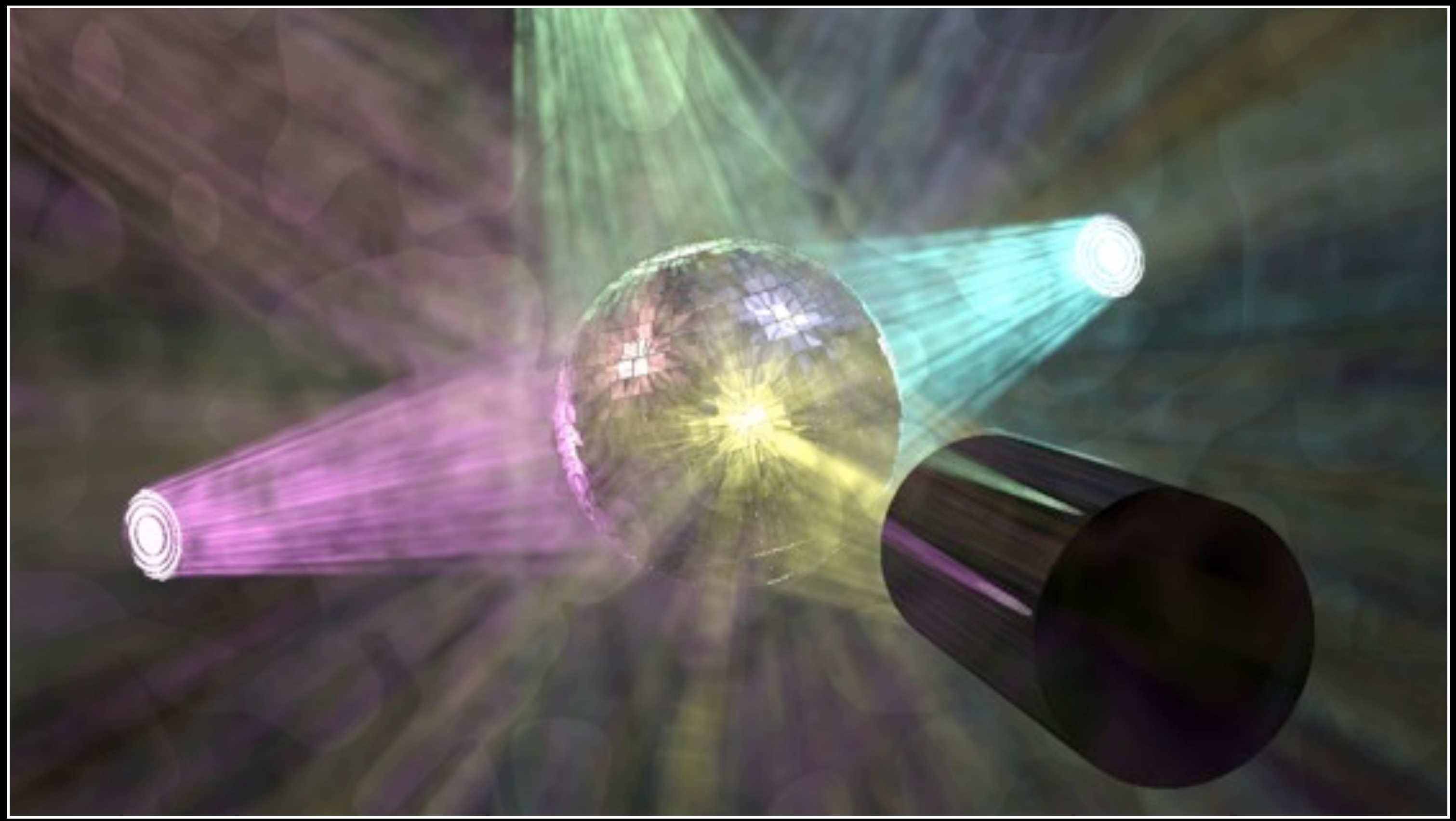


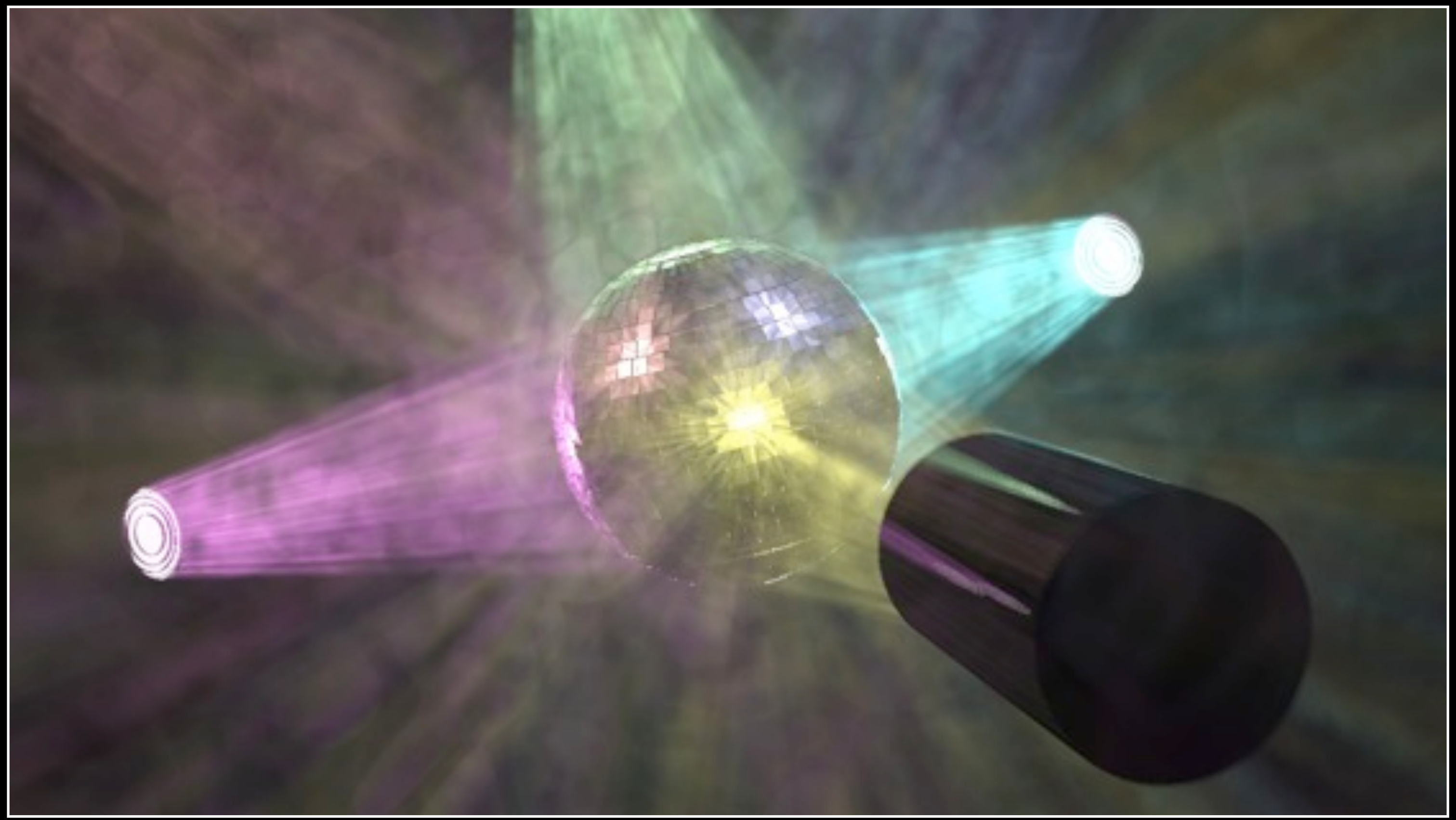
Pass 2

Average of Passes 1..2

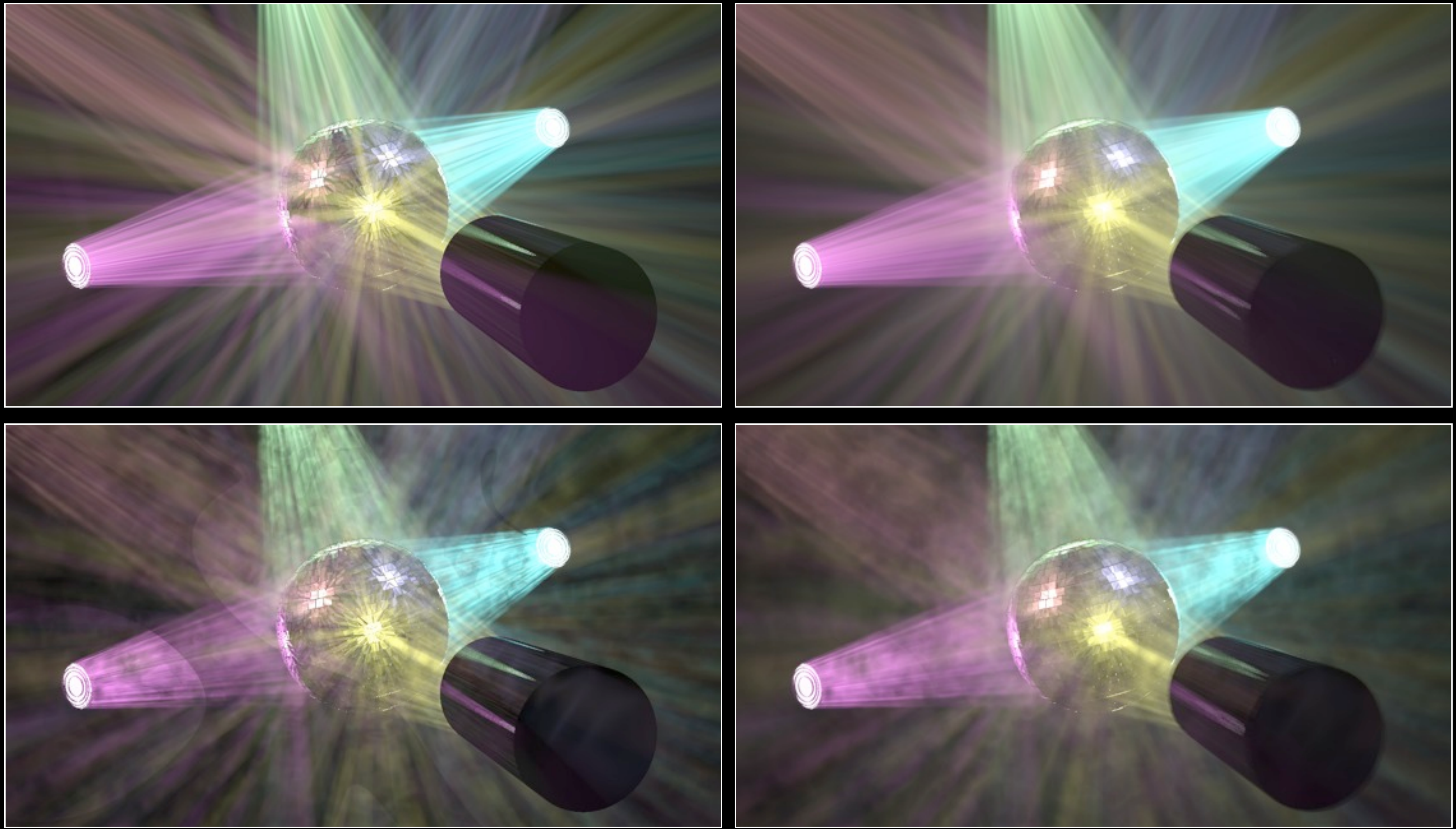




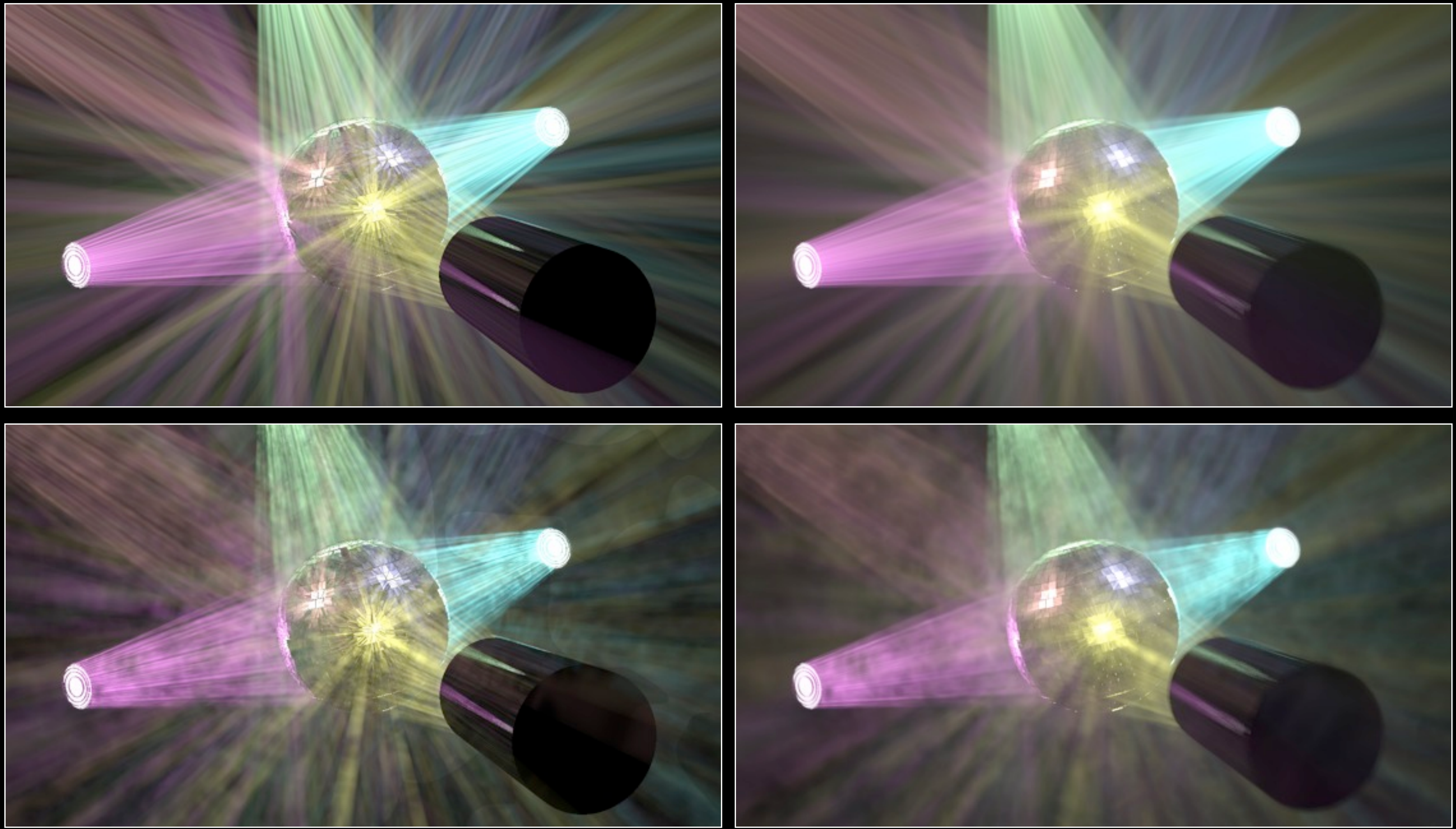




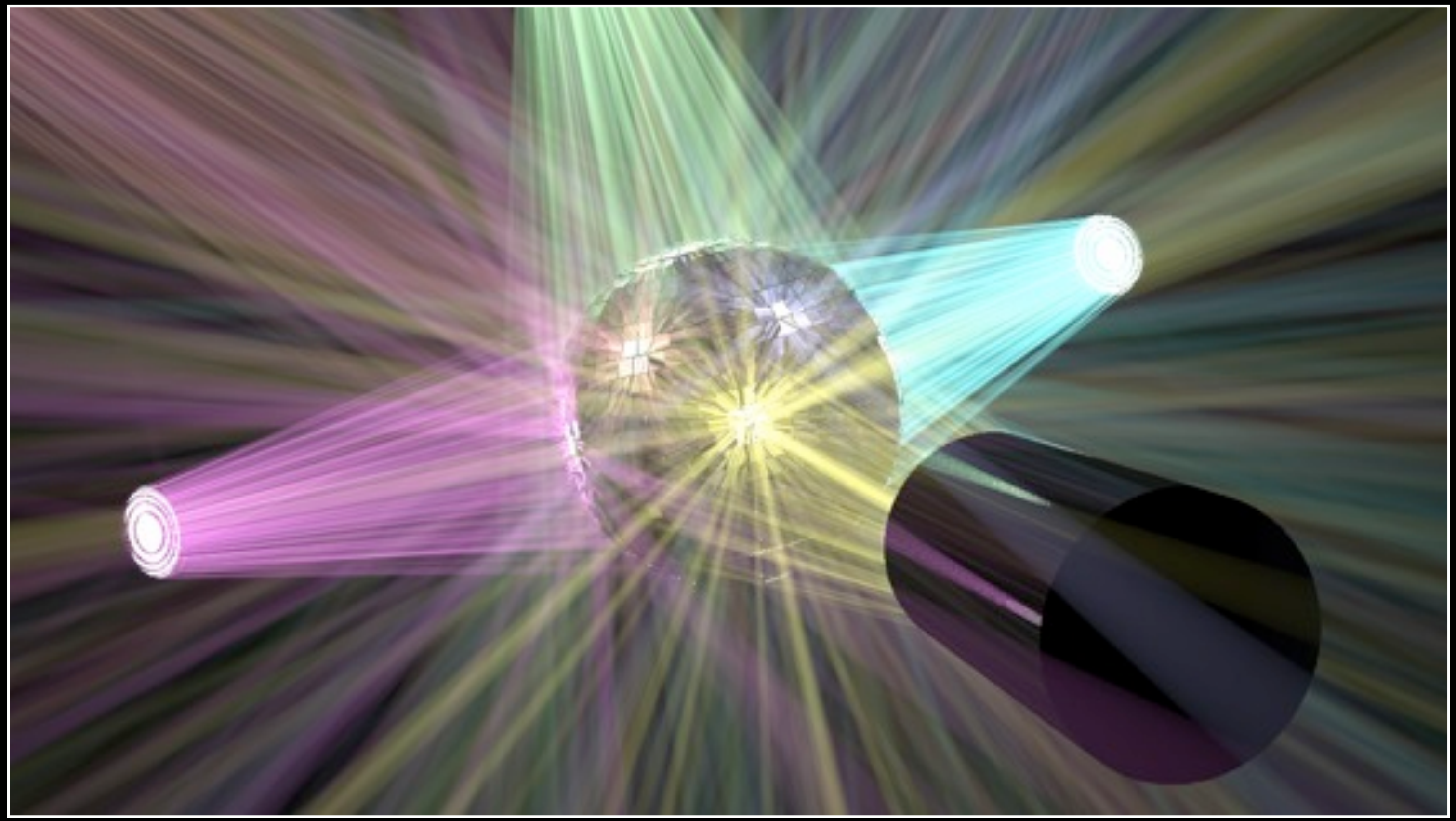
Pass 4	Average of Passes 1..4

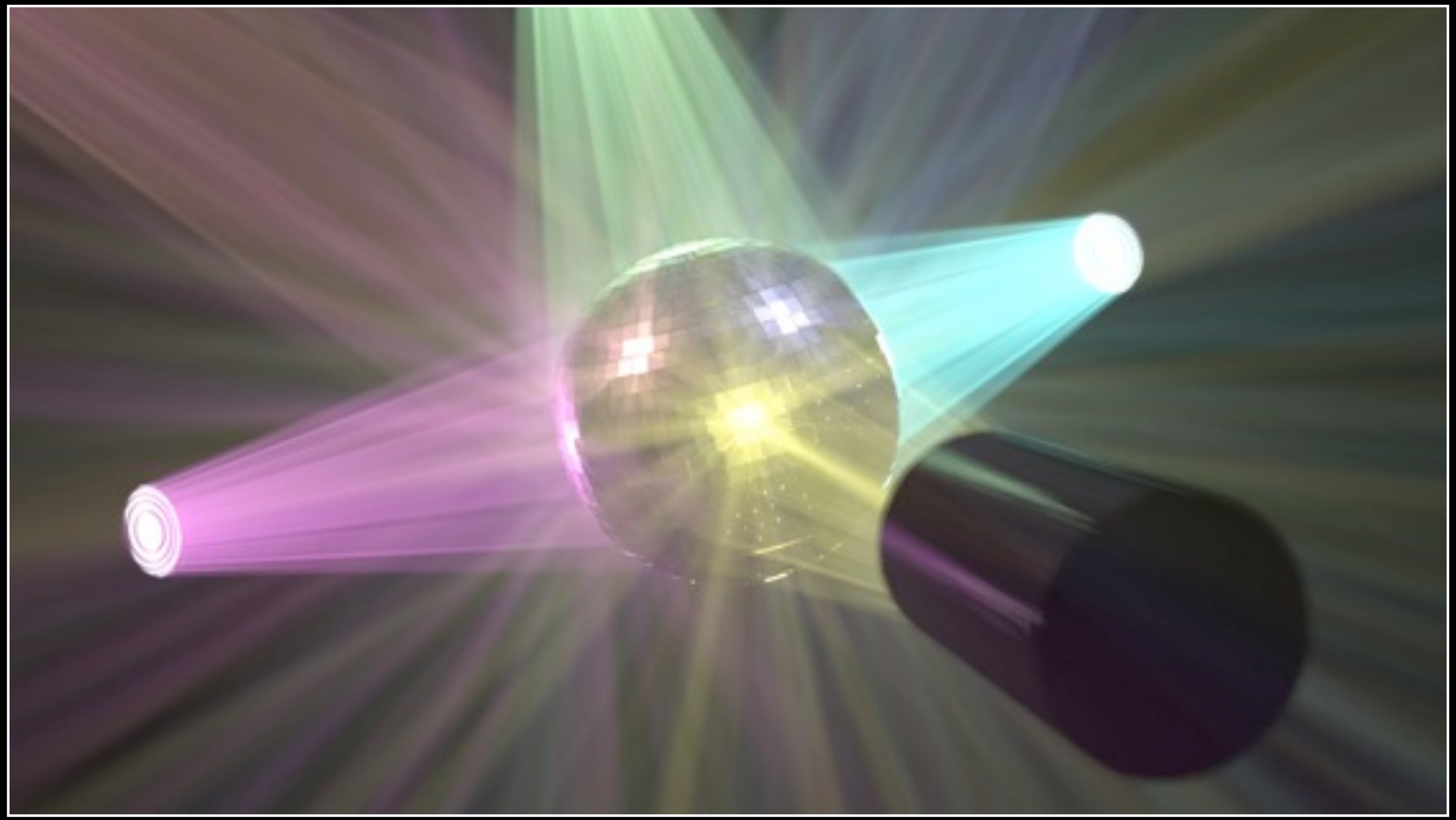


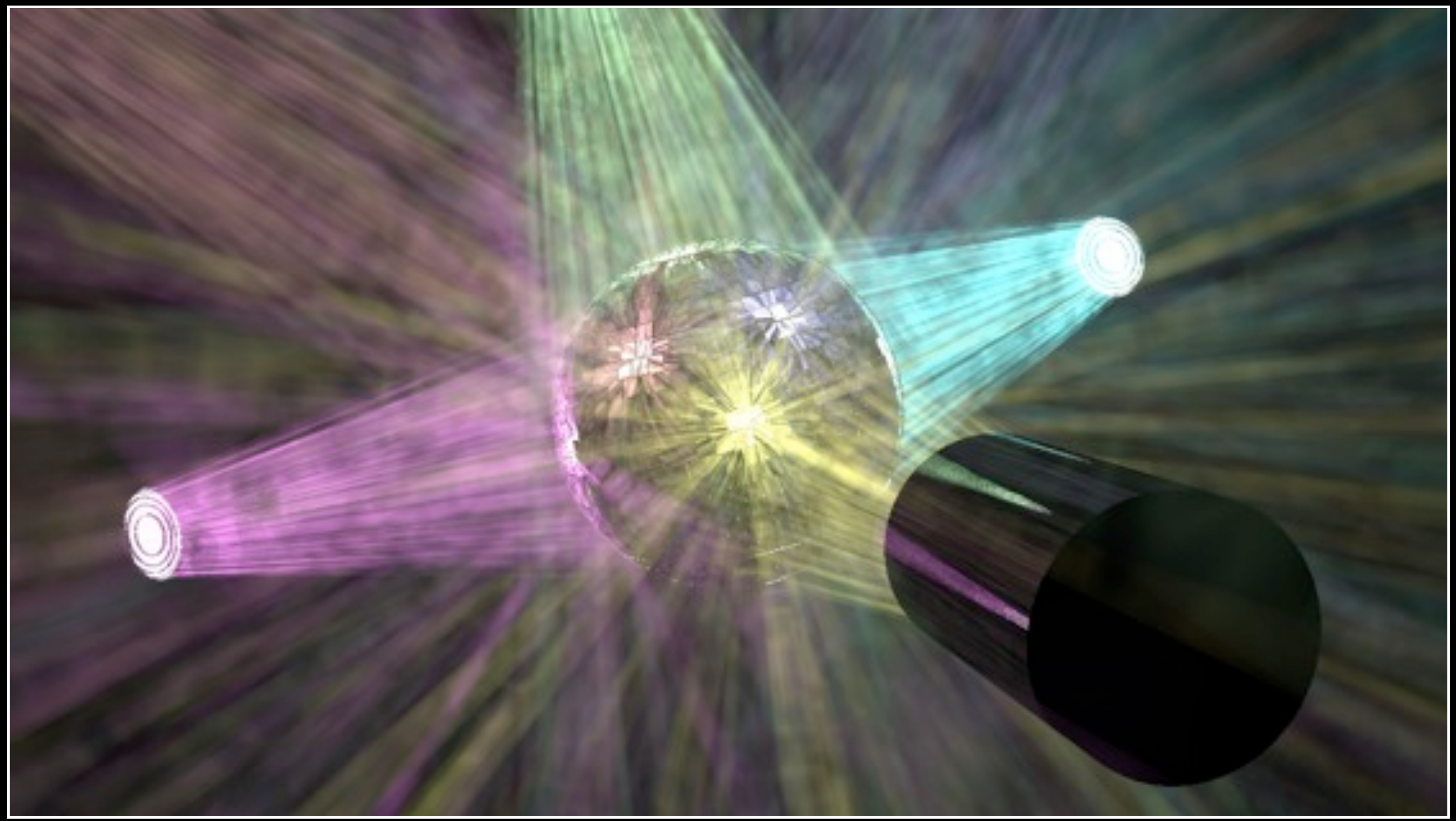
Pass 8 | Average of Passes 1..8

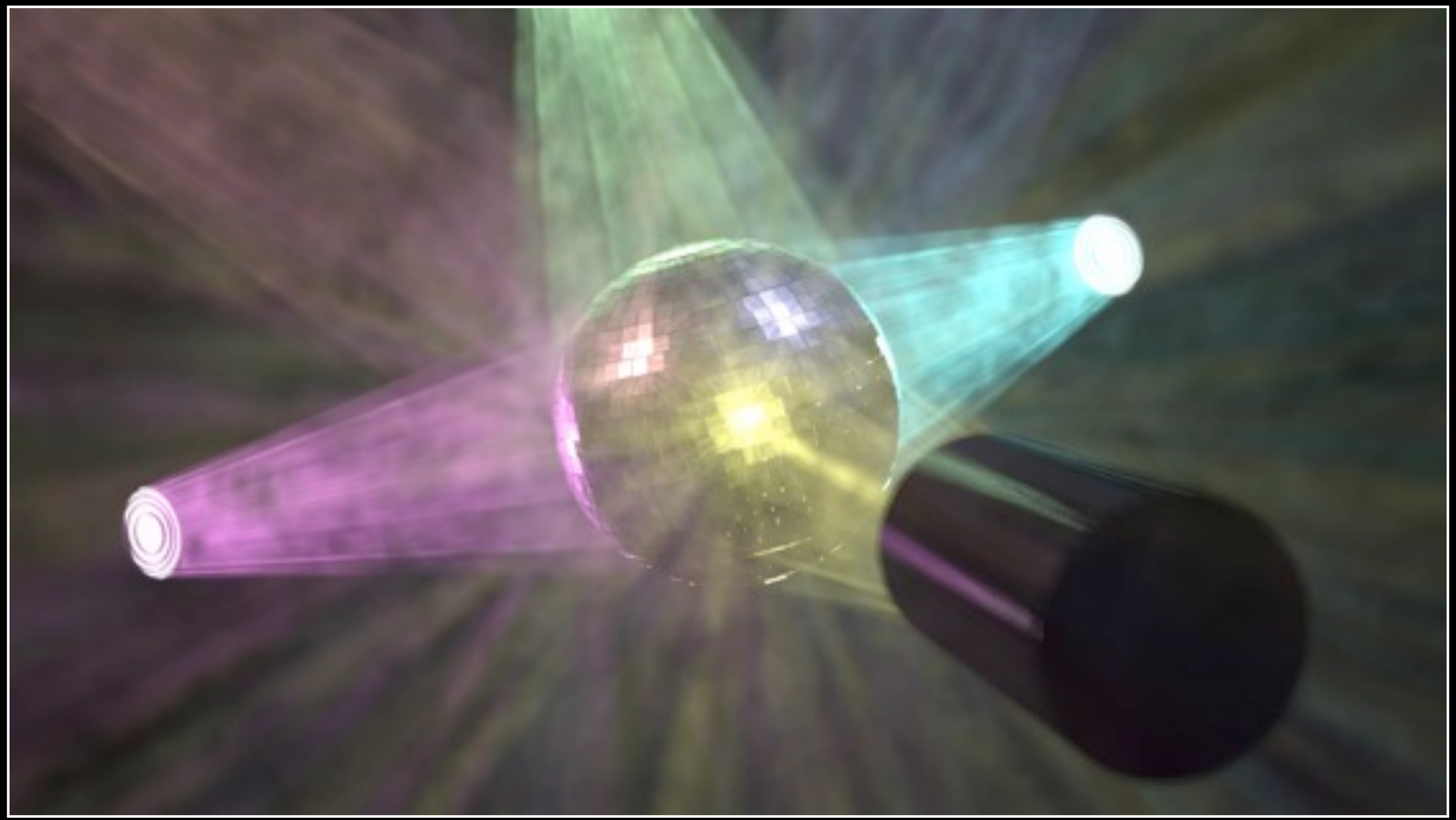


Pass 16 | Average of Passes 1..16

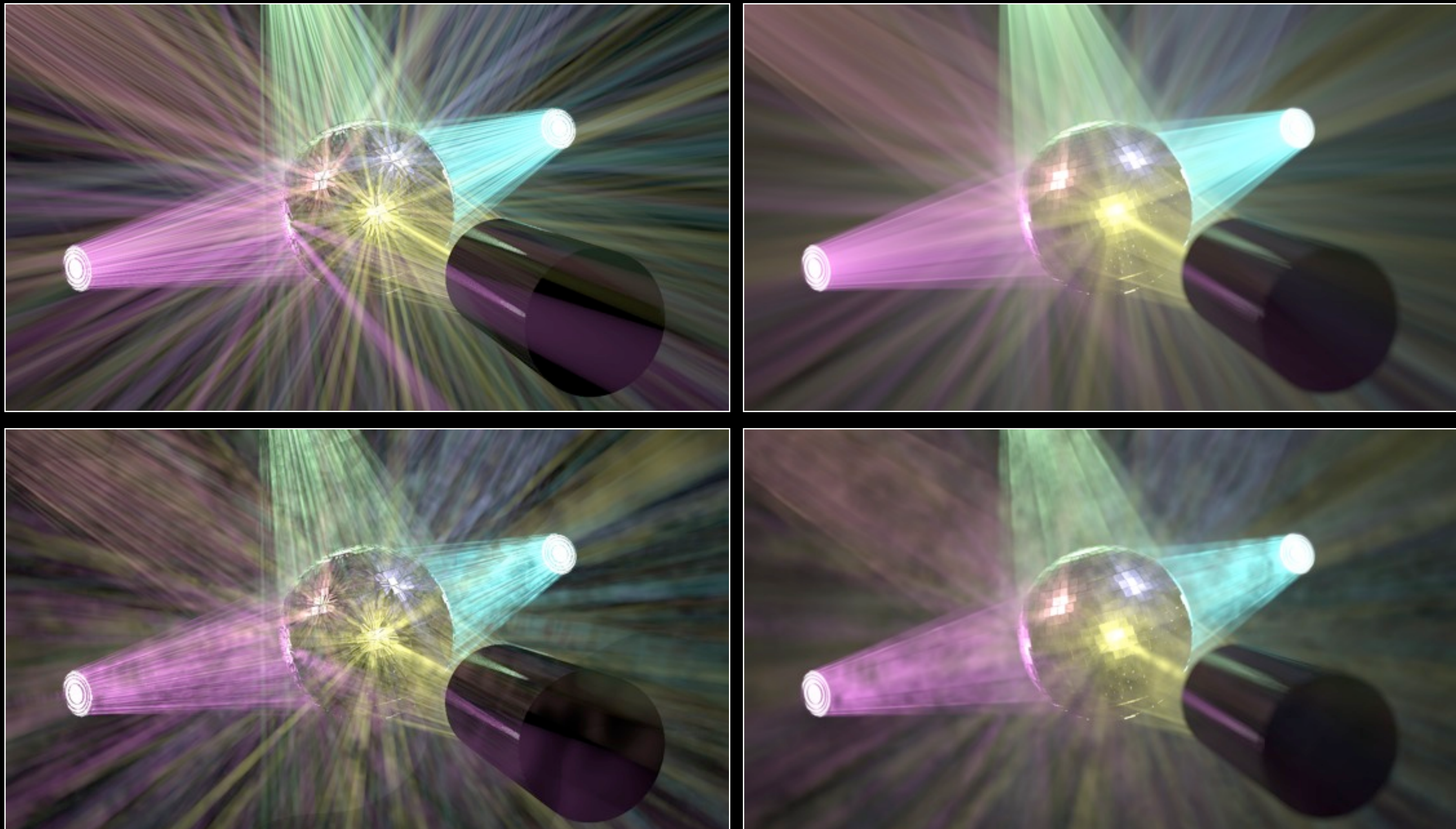




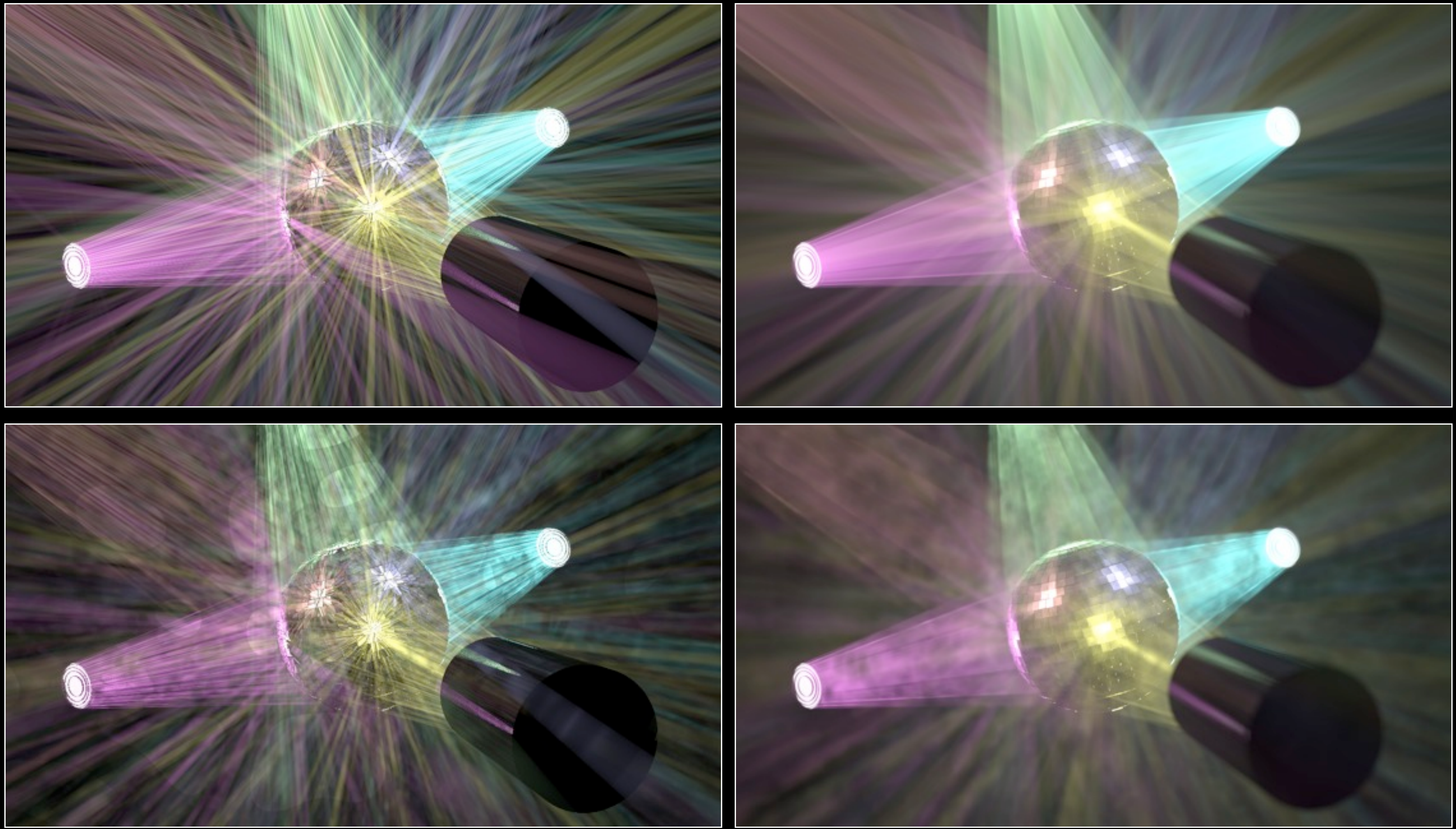




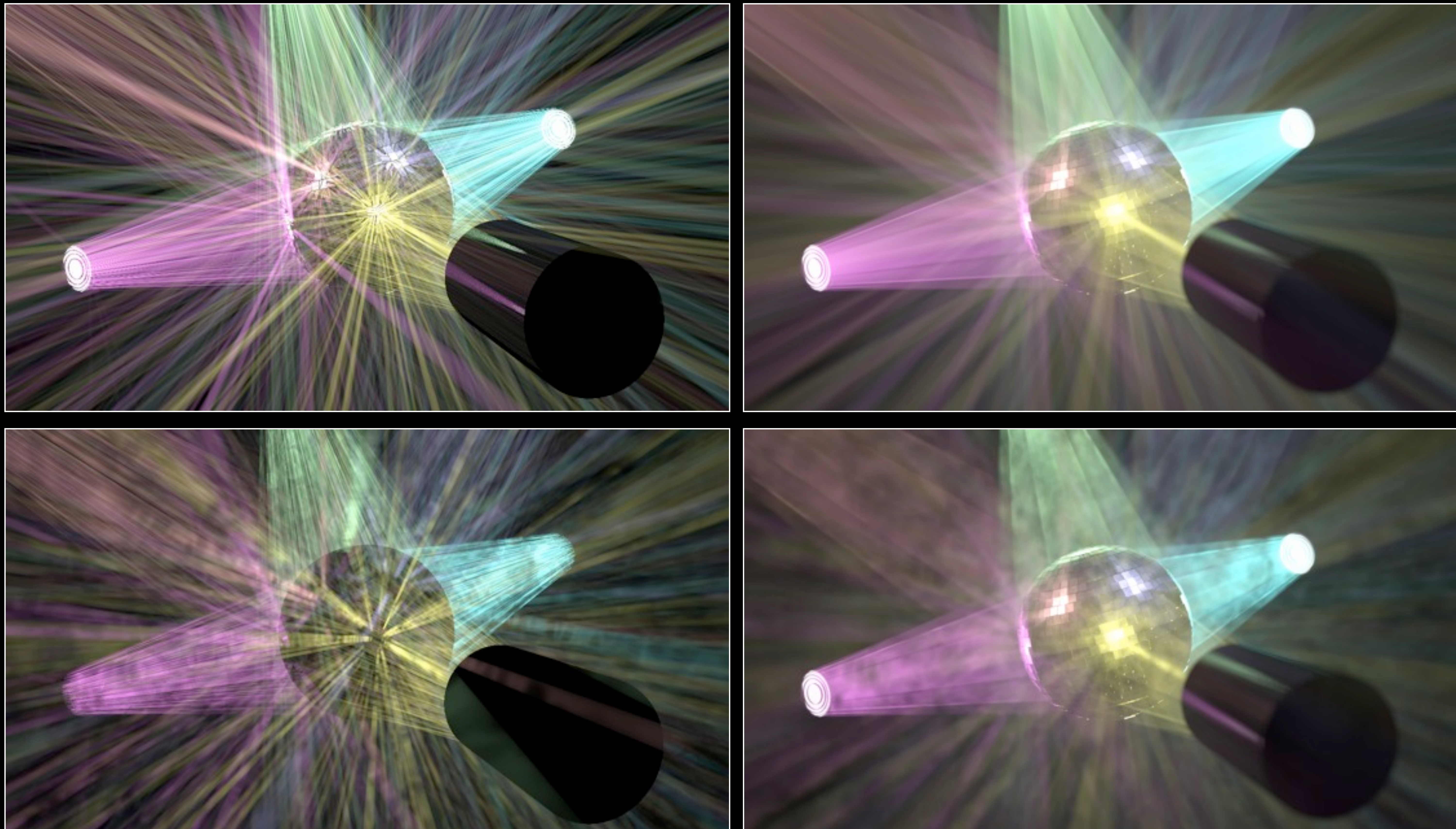
Pass 32 | Average of Passes 1..32



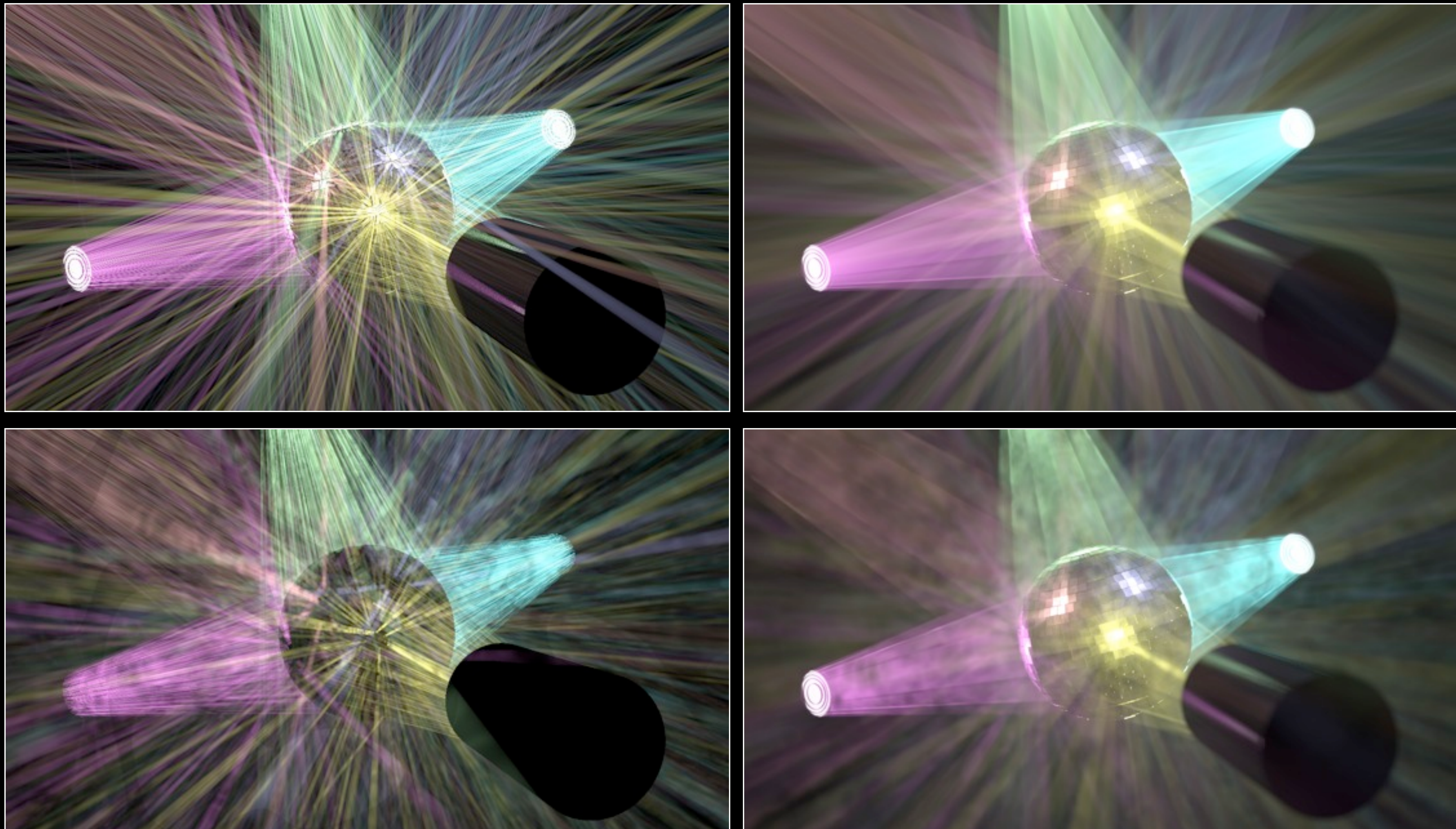
Pass 64	Average of Passes 1..64



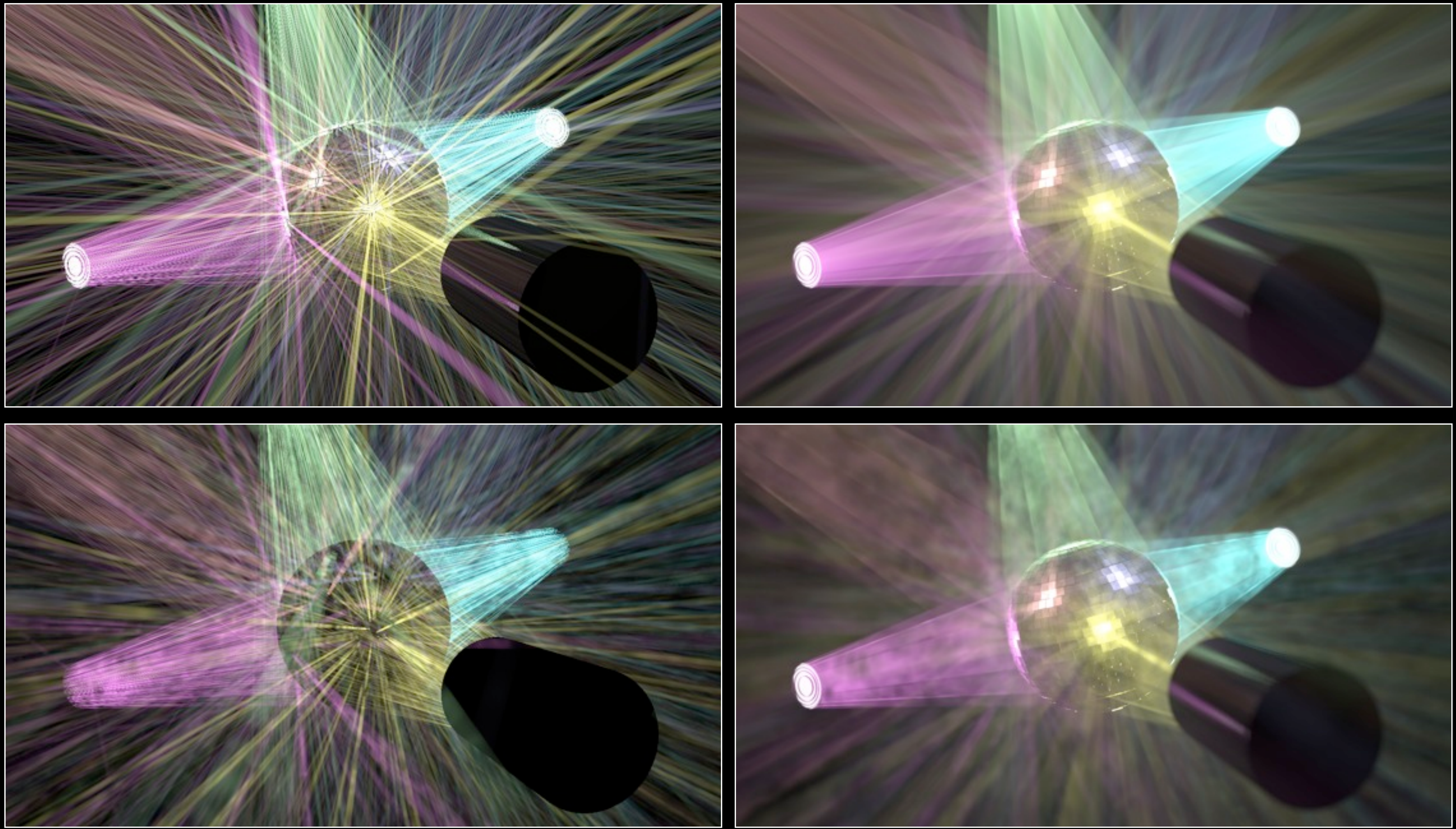
Pass 128 | Average of Passes 1..128



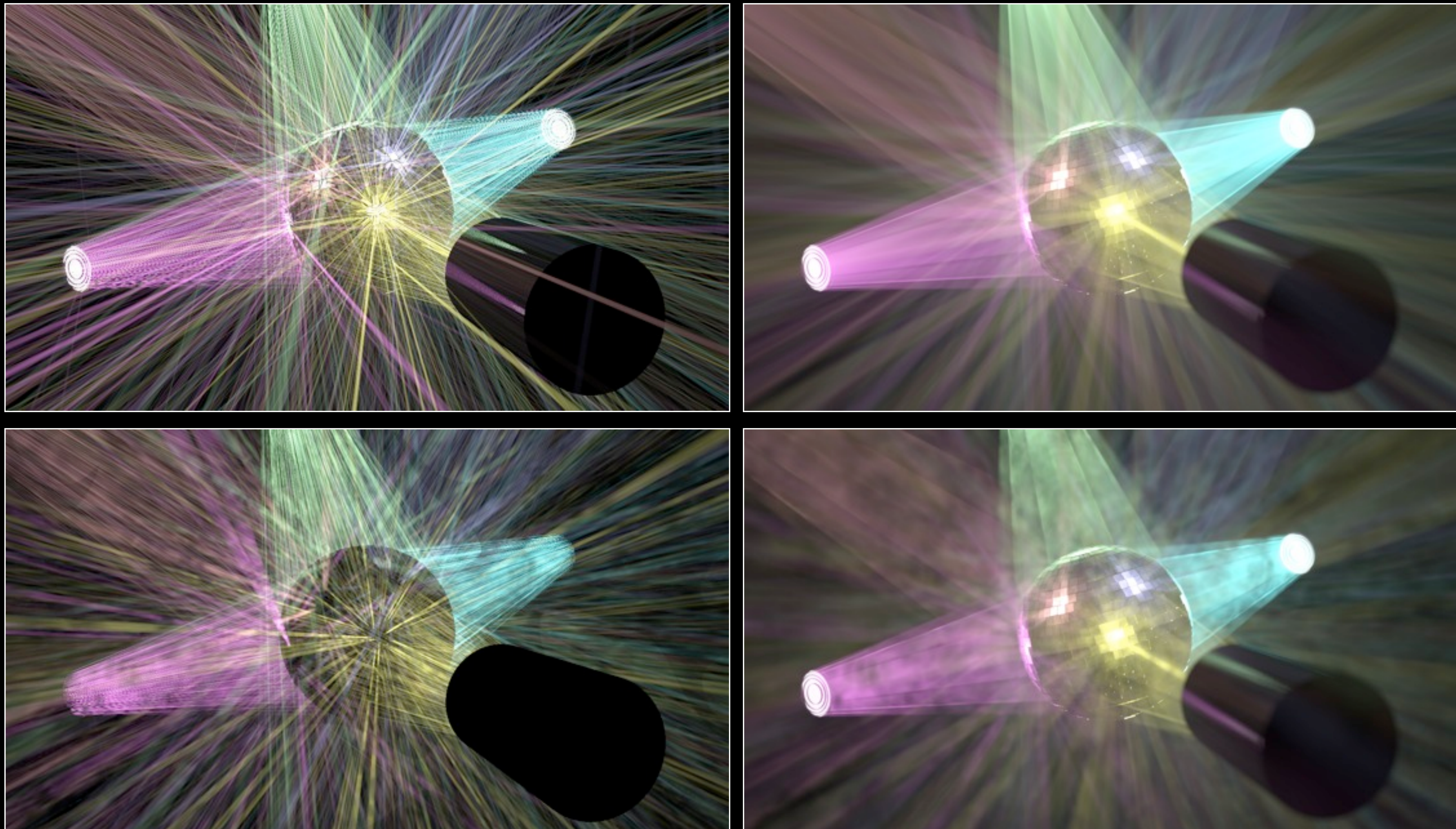
Pass 256 | Average of Passes 1..256



Pass 512 | Average of Passes 1..512



Pass 1024 | Average of Passes 1..1024



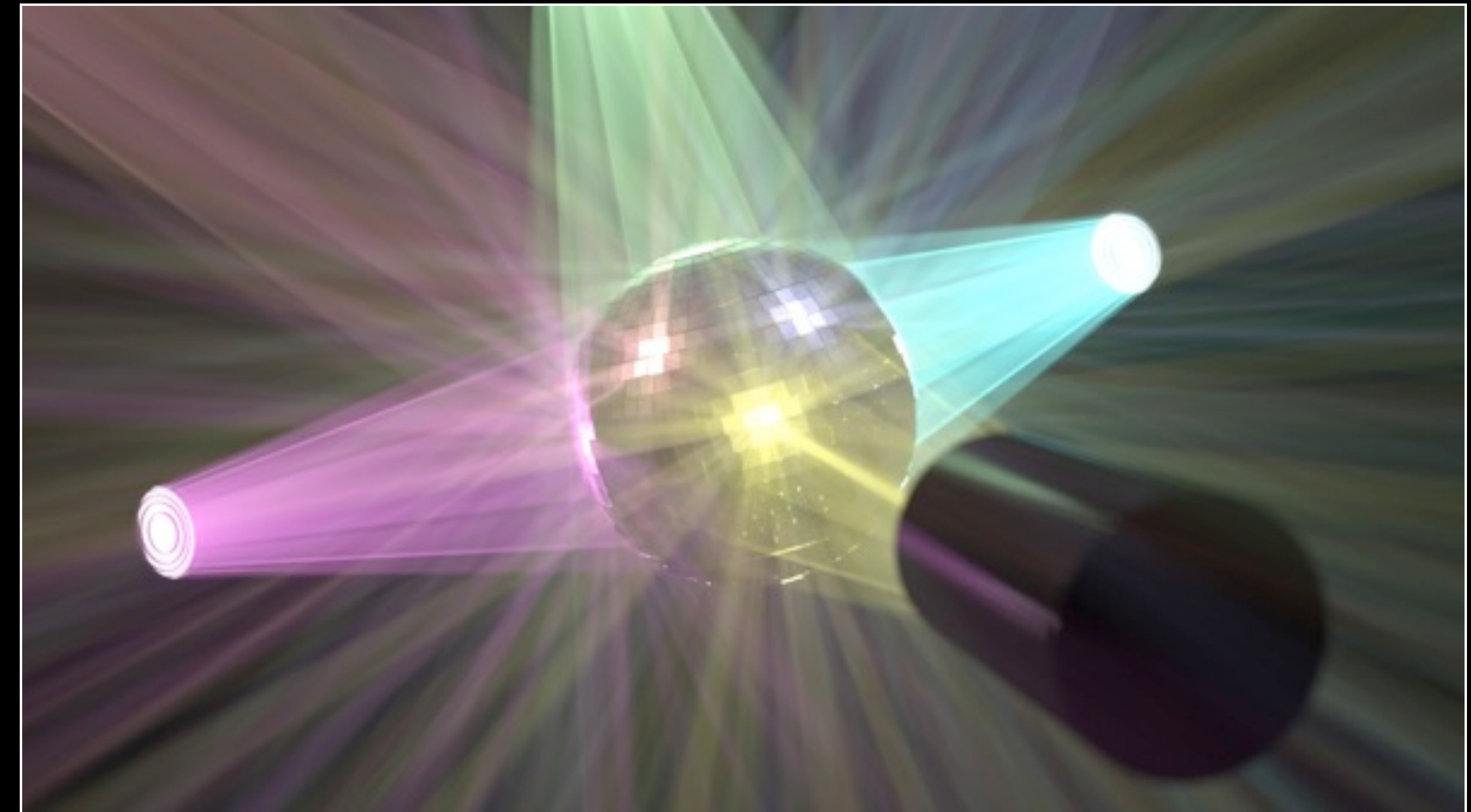
Disco

1280x720, Depth-of-Field

Homogeneous

19.67M Photon Beams

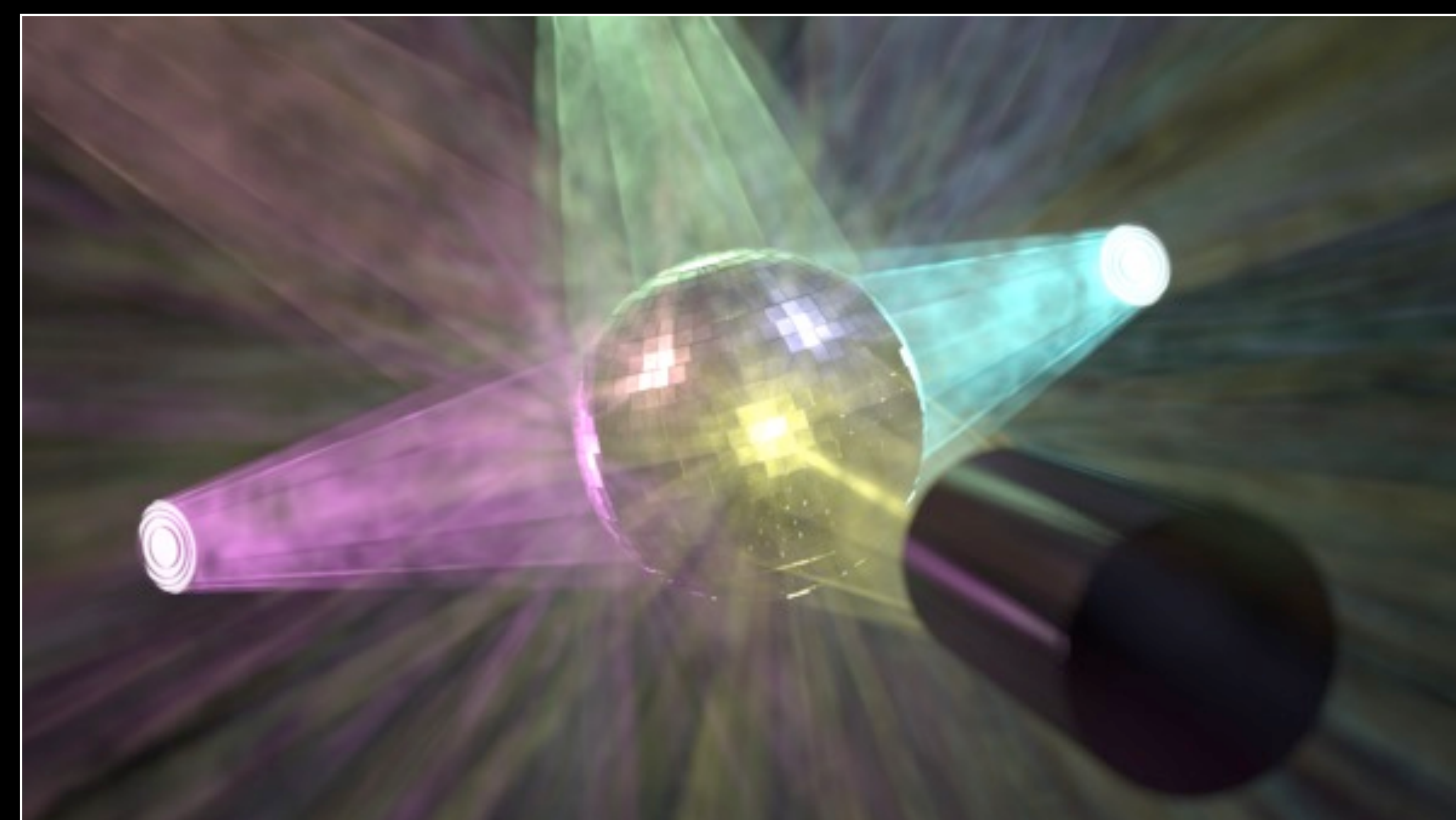
3 minutes



Heterogeneous

16.19M Photon Beams

5.7 minutes



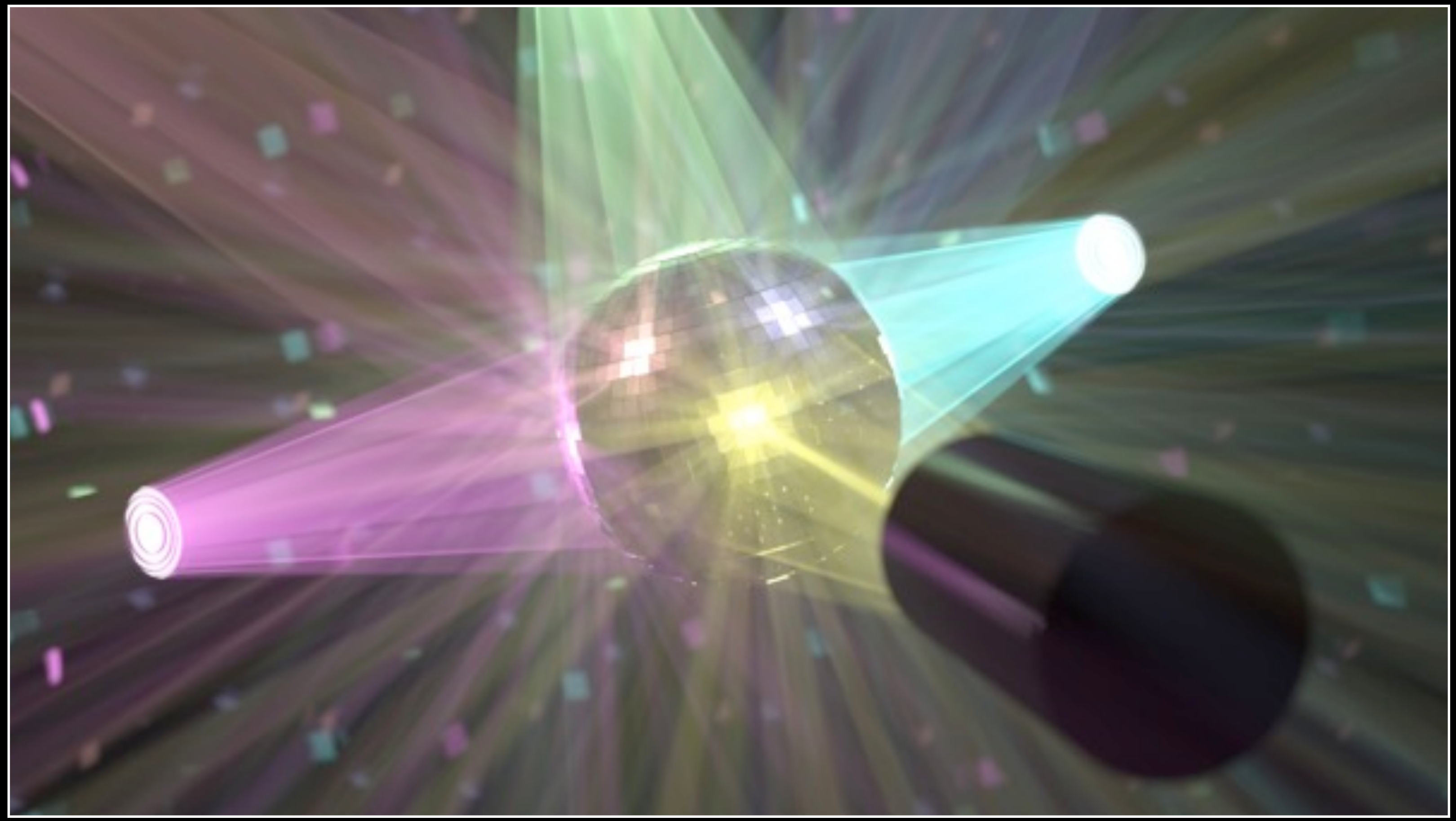
Disco

1280x720, Depth-of-Field

Homogeneous

19.67M Photon Beams

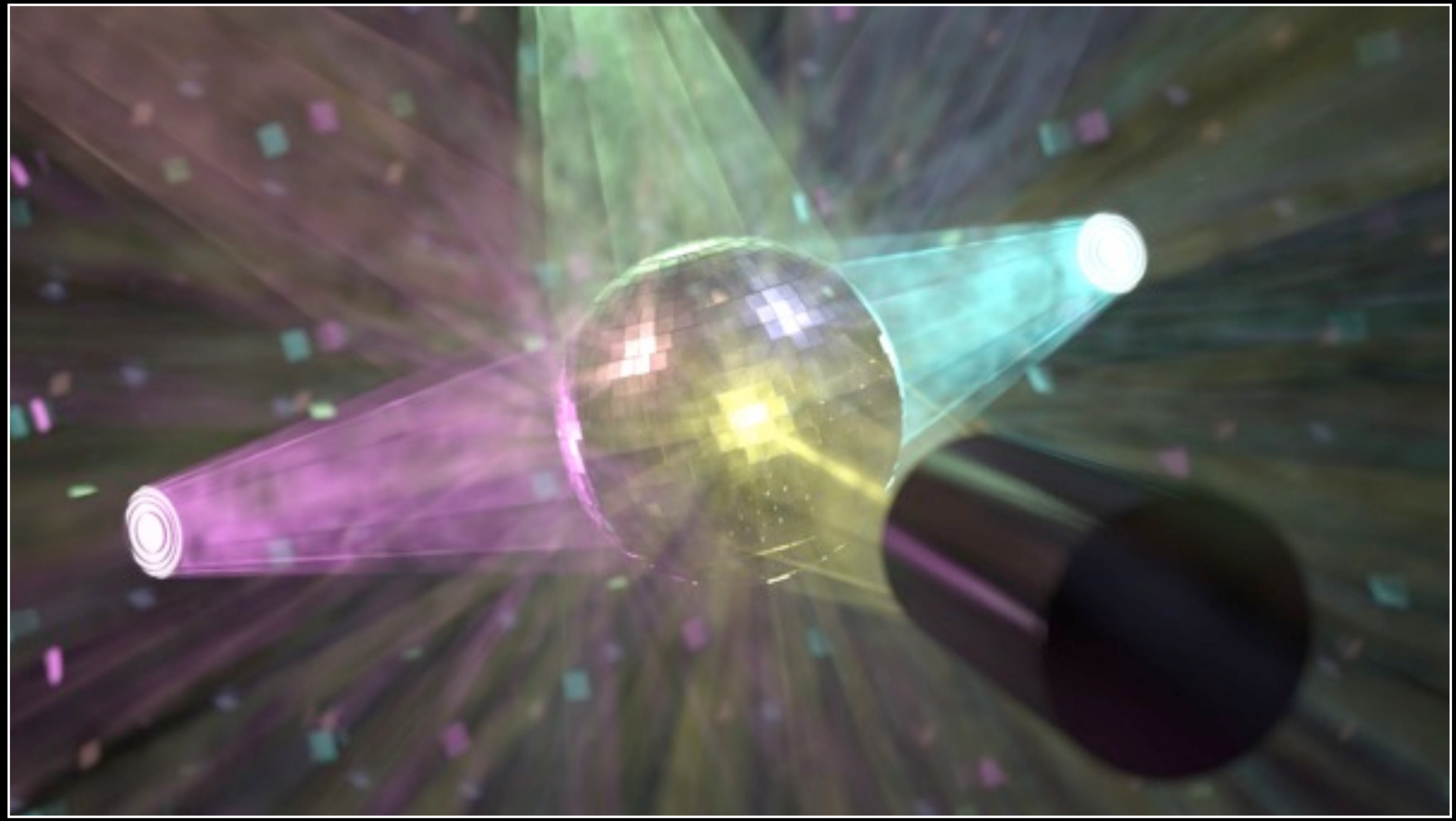
3 minutes



Heterogeneous

16.19M Photon Beams

5.7 minutes



USER INTERACTION

Hybrid CPU/GPU Implementation

Homogeneous

Heterogeneous

Real-time capture

Questions?