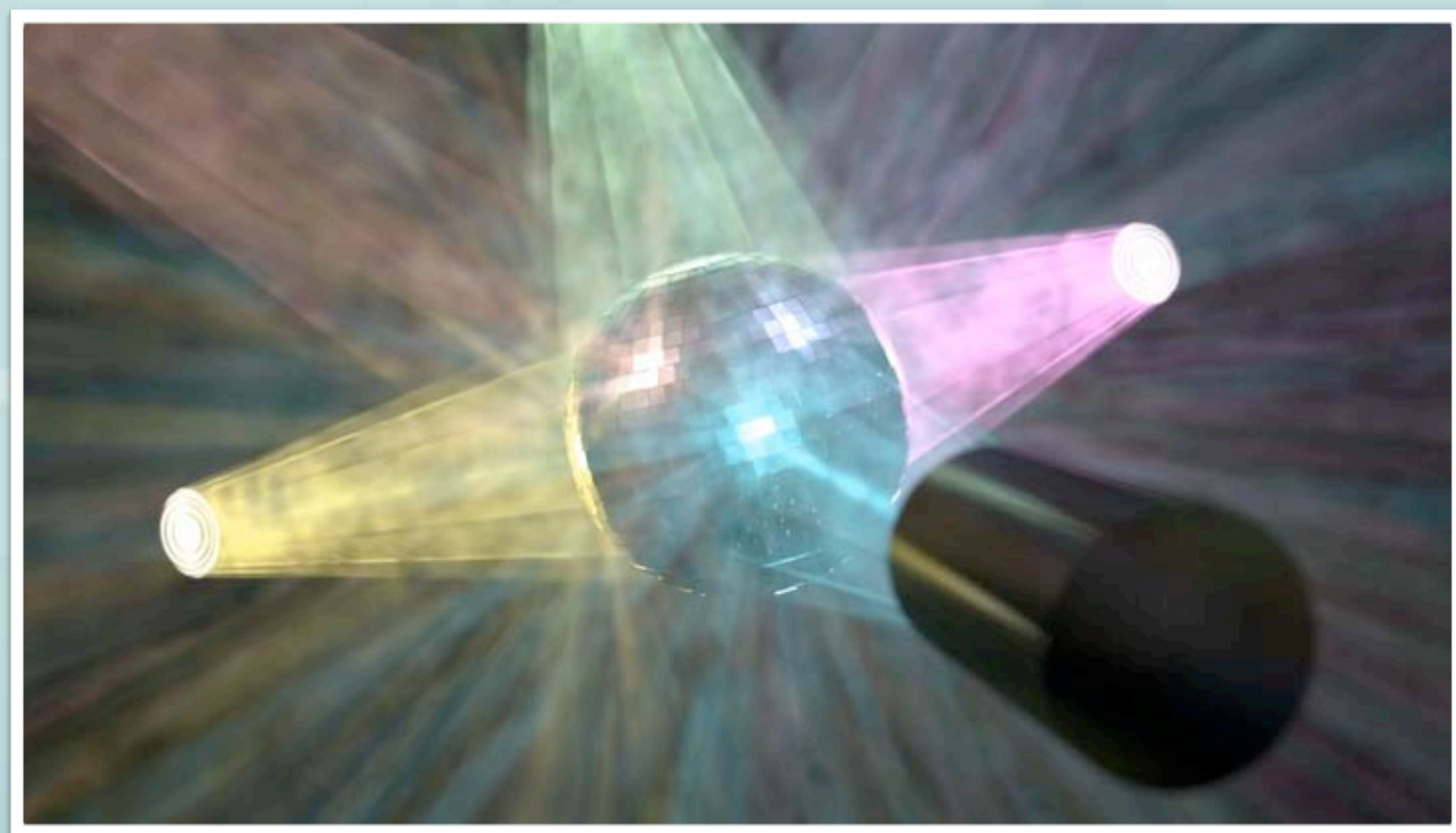
Progressive Photon Beams



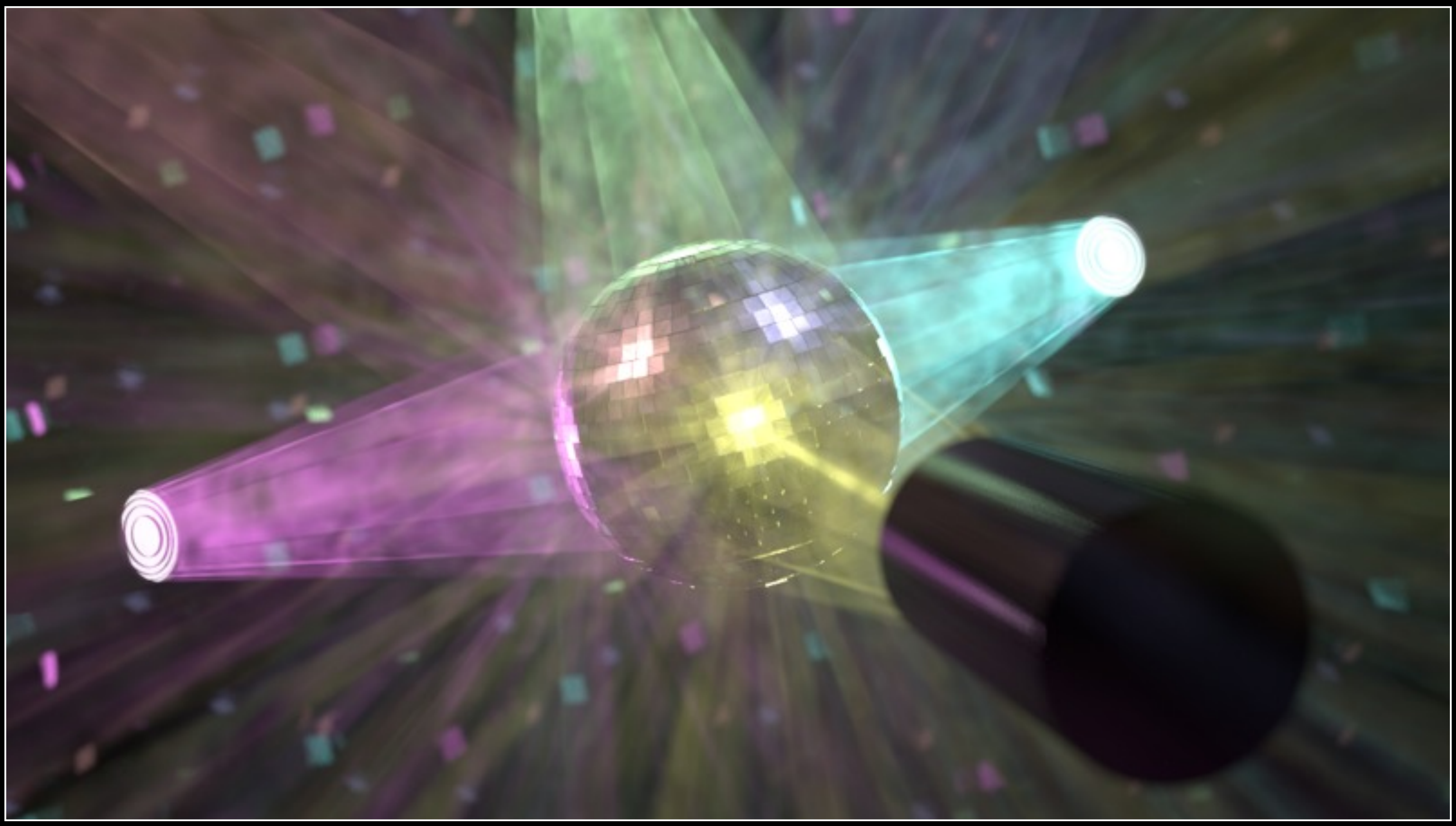
Wojciech Jarosz[1] Derek Nowrouzezahrai[1]
Robert Thomas[1] Peter-Pike Sloan[2] Matthias Zwicker[3]

[1]Disney Research Zürich [2]Disney Interactive Studios [2]Universität Bern

Association for Computing Machinery

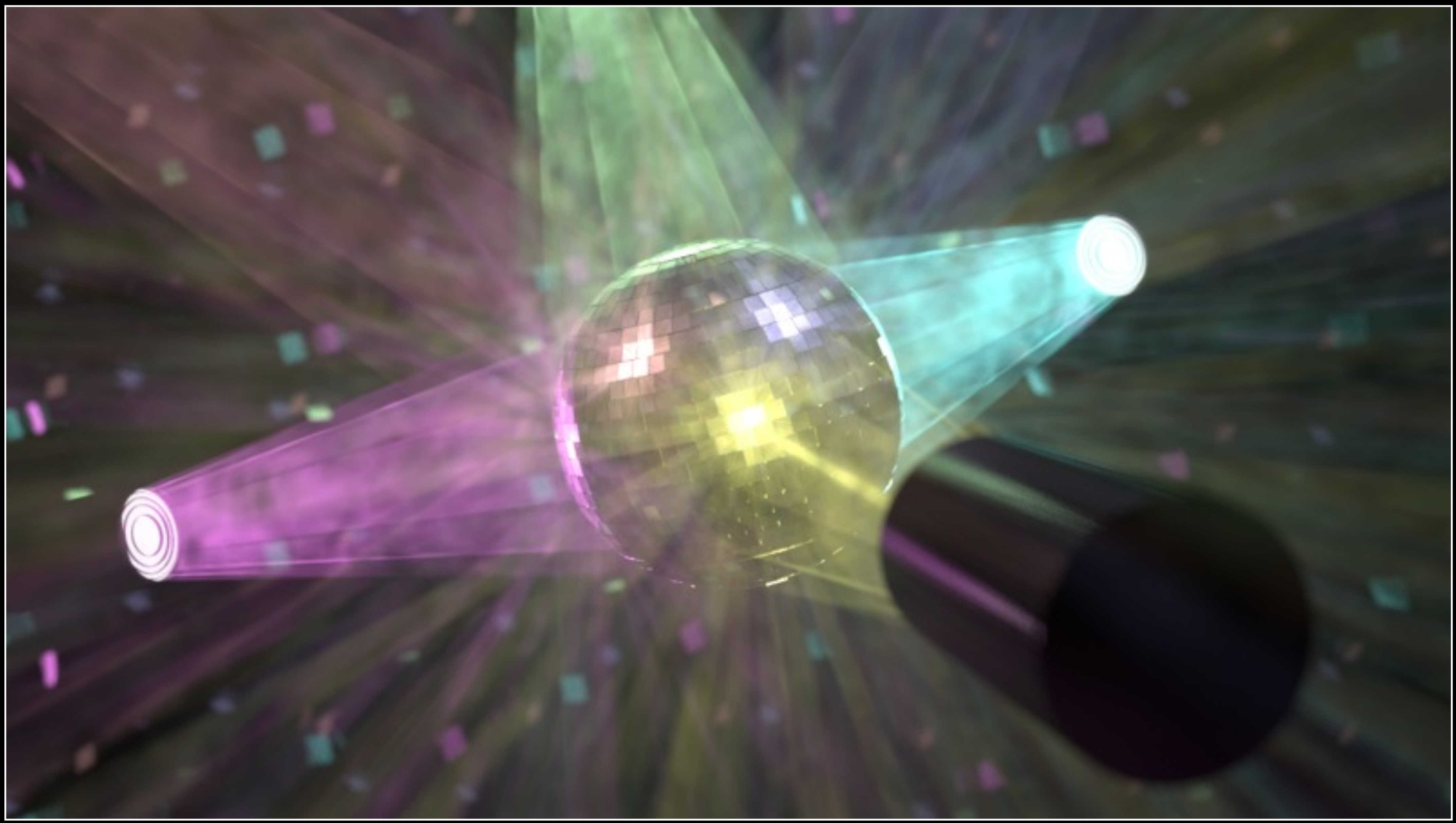
ACMSIGGRAPH

- Thanks for the introduction, and thank you all for attending



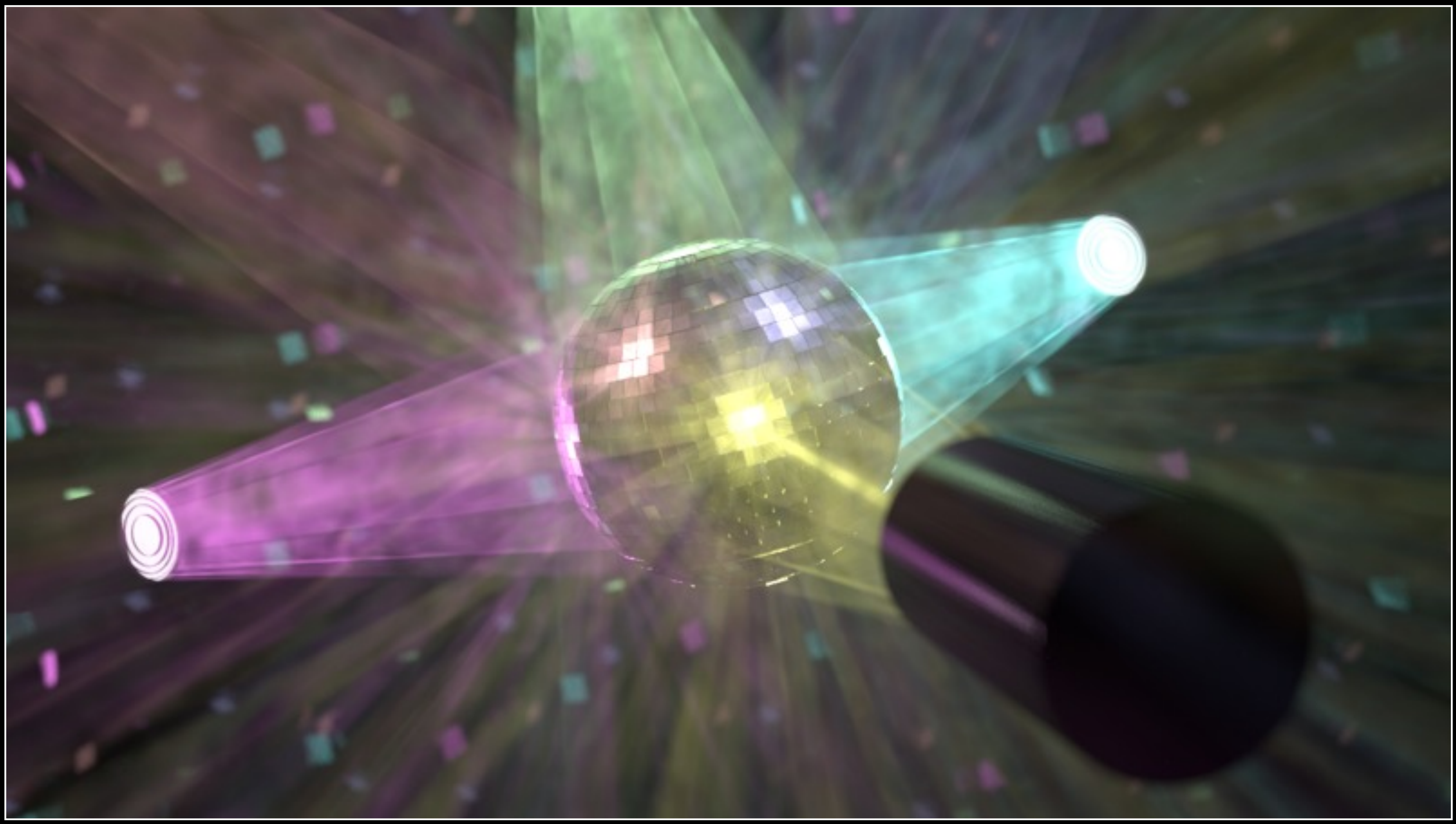
Thursday, 6 September 12

- In this talk we are interesting in rendering scenes like the one shown here
- Unfortunately, most previous rendering techniques have difficulty with this type of scene [click] due to the extremely complex light paths involved. [click]
- In particular, each light source is modeled realistically using a collection of mirrors and lenses
- Hence, all the illumination you see is actually caustics [click]
- Furthermore, we have a heterogeneous medium. [click]
- I'll describe our new Progressive Photon Beams approach which, by combining the efficiency of photon beams with the convergence guarantees of progressive photon mapping, is able to handle scenes like this quickly and accurately



- Complex light paths

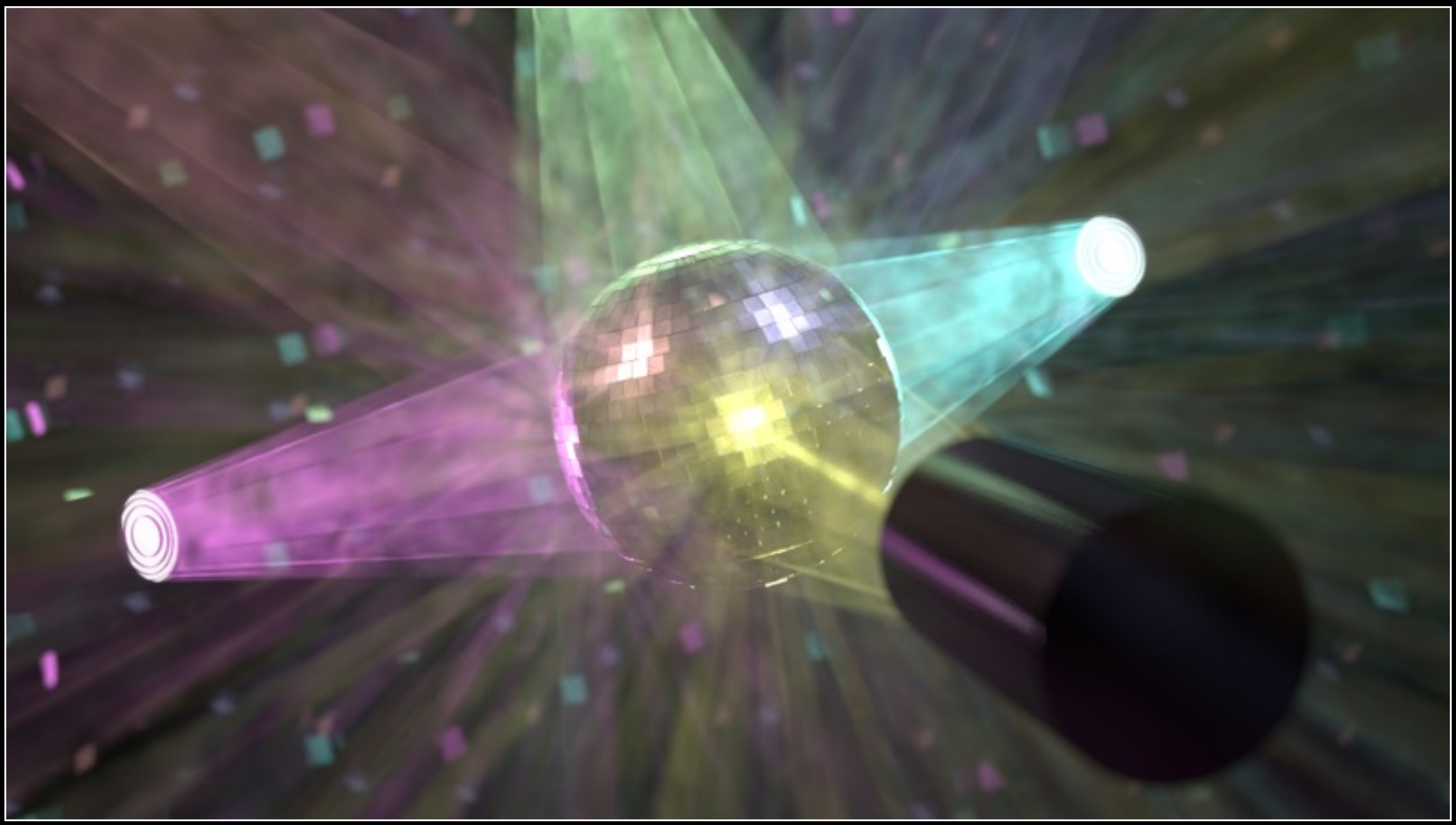
- In this talk we are interesting in rendering scenes like the one shown here
- Unfortunately, most previous rendering techniques have difficulty with this type of scene [click] due to the extremely complex light paths involved. [click]
- In particular, each light source is modeled realistically using a collection of mirrors and lenses
- Hence, all the illumination you see is actually caustics [click]
- Furthermore, we have a heterogeneous medium. [click]
- I'll describe our new Progressive Photon Beams approach which, by combining the efficiency of photon beams with the convergence guarantees of progressive photon mapping, is able to handle scenes like this quickly and accurately



- Complex light paths
- Realistic light sources (encased in glass: caustics)

Thursday, 6 September 12

- In this talk we are interesting in rendering scenes like the one shown here
- Unfortunately, most previous rendering techniques have difficulty with this type of scene [click] due to the extremely complex light paths involved. [click]
- In particular, each light source is modeled realistically using a collection of mirrors and lenses
- Hence, all the illumination you see is actually caustics [click]
- Furthermore, we have a heterogeneous medium. [click]
- I'll describe our new Progressive Photon Beams approach which, by combining the efficiency of photon beams with the convergence guarantees of progressive photon mapping, is able to handle scenes like this quickly and accurately

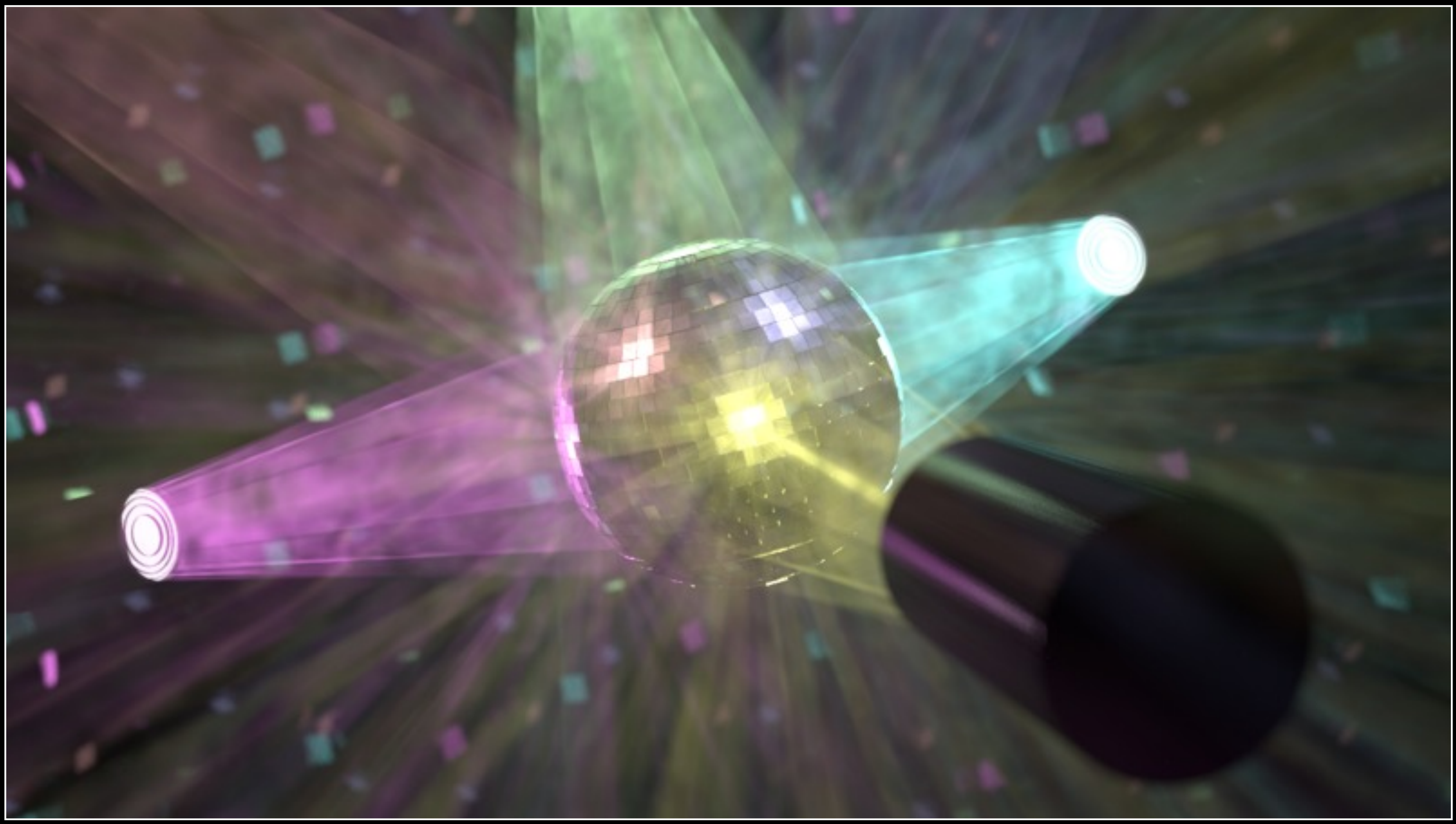


- Complex light paths
- Realistic light sources (encased in glass: caustics)
- Heterogeneous media

- In this talk we are interesting in rendering scenes like the one shown here
- Unfortunately, most previous rendering techniques have difficulty with this type of scene [click] due to the extremely complex light paths involved. [click]
- In particular, each light source is modeled realistically using a collection of mirrors and lenses
- Hence, all the illumination you see is actually caustics [click]
- Furthermore, we have a heterogeneous medium. [click]
- I'll describe our new Progressive Photon Beams approach which, by combining the efficiency of photon beams with the convergence guarantees of progressive photon mapping, is able to handle scenes like this quickly and accurately

Progressive Photon Beams

= ***progressive*** photon mapping using ***photon beams***



- Complex light paths
- Realistic light sources (encased in glass: caustics)
- Heterogeneous media

Thursday, 6 September 12

- In this talk we are interesting in rendering scenes like the one shown here
- Unfortunately, most previous rendering techniques have difficulty with this type of scene [click] due to the extremely complex light paths involved. [click]
- In particular, each light source is modeled realistically using a collection of mirrors and lenses
- Hence, all the illumination you see is actually caustics [click]
- Furthermore, we have a heterogeneous medium. [click]
- I'll describe our new Progressive Photon Beams approach which, by combining the efficiency of photon beams with the convergence guarantees of progressive photon mapping, is able to handle scenes like this quickly and accurately

Previous Work: Unbiased Methods

- Path tracing
  - [Kajiya 86]
  - [Veach and Guibas 94]
  - [Lafortune and Willems 96]

- The gold standard for rendering these types of scene is arguably computing unbiased, noise-free images.
- Several algorithms exist, which are essentially variants of brute-force path tracing or Metropolis light transport
- Though these methods are unbiased, [click] they are notoriously slow to converge to noise-free images
- Another problem with these approaches [click] is their inability to handle certain types of light paths robustly, in particular caustics or specular reflections/refractions of caustics. Hence, they would fall apart on this disco scene

Previous Work: Unbiased Methods

- Path tracing
  - [Kajiya 86]
  - [Veach and Guibas 94]
  - [Lafortune and Willems 96]
- Metropolis light transport
  - [Veach and Guibas 97]
  - [Pauly et al. 00]

✓ unbiased

Thursday, 6 September 12

- The gold standard for rendering these types of scene is arguably computing unbiased, noise-free images.
- Several algorithms exist, which are essentially variants of brute-force path tracing or Metropolis light transport
- Though these methods are unbiased, [click] they are notoriously slow to converge to noise-free images
- Another problem with these approaches [click] is their inability to handle certain types of light paths robustly, in particular caustics or specular reflections/refractions of caustics. Hence, they would fall apart on this disco scene

Previous Work: Unbiased Methods

- Path tracing
  - [Kajiya 86]
  - [Veach and Guibas 94]
  - [Lafortune and Willems 96]
- Metropolis light transport
  - [Veach and Guibas 97]
  - [Pauly et al. 00]

✓ unbiased

– slow

Thursday, 6 September 12

- The gold standard for rendering these types of scene is arguably computing unbiased, noise-free images.
- Several algorithms exist, which are essentially variants of brute-force path tracing or Metropolis light transport
- Though these methods are unbiased, [click] they are notoriously slow to converge to noise-free images
- Another problem with these approaches [click] is their inability to handle certain types of light paths robustly, in particular caustics or specular reflections/refractions of caustics. Hence, they would fall apart on this disco scene

Previous Work: Unbiased Methods

- Path tracing
  - [Kajiya 86]
  - [Veach and Guibas 94]
  - [Lafortune and Willems 96]
- Metropolis light transport
  - [Veach and Guibas 97]
  - [Pauly et al. 00]

✓ unbiased

– slow

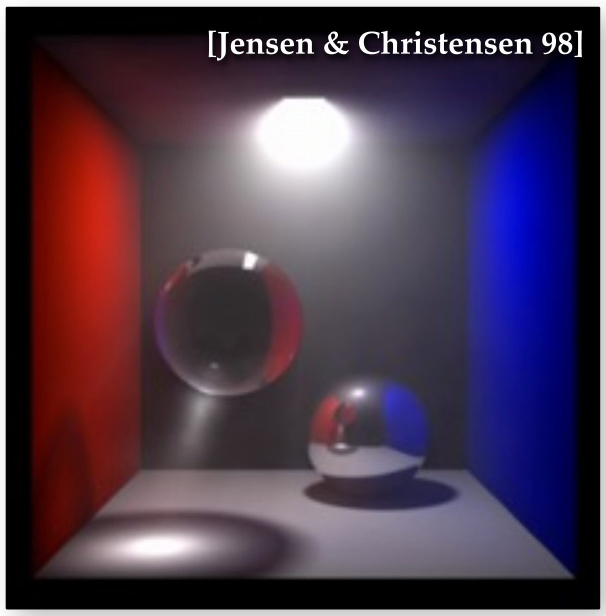
– not robust to caustics

Thursday, 6 September 12

- The gold standard for rendering these types of scene is arguably computing unbiased, noise-free images.
- Several algorithms exist, which are essentially variants of brute-force path tracing or Metropolis light transport
- Though these methods are unbiased, [click] they are notoriously slow to converge to noise-free images
- Another problem with these approaches [click] is their inability to handle certain types of light paths robustly, in particular caustics or specular reflections/refractions of caustics. Hence, they would fall apart on this disco scene

Previous Work: Volumetric PM

- Volumetric Photon Mapping
  - [Jensen & Christensen 98]
  - [Jarosz et al. 08]

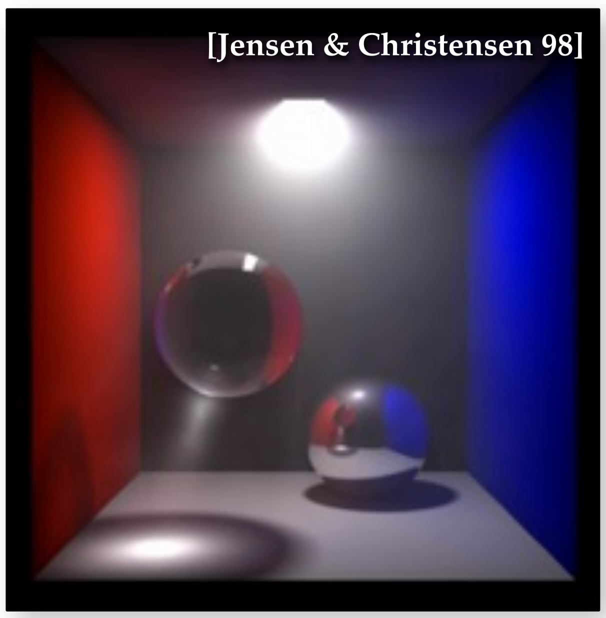


Thursday, 6 September 12

- Methods such as volumetric photon mapping do not suffer from these problems [click]
- They typically produce high-quality results much faster than unbiased techniques [click]
- and is one of the few algorithms that is robust to caustic paths [click]
- However, photon mapping introduces bias.
- It is consistent, though, which means that if we use an infinite number of photons, we will get the correct solution.
- but, this is of little practical value since we obviously cannot store an unlimited number of photons.

Previous Work: Volumetric PM

- Volumetric Photon Mapping
  - [Jensen & Christensen 98]
  - [Jarosz et al. 08]

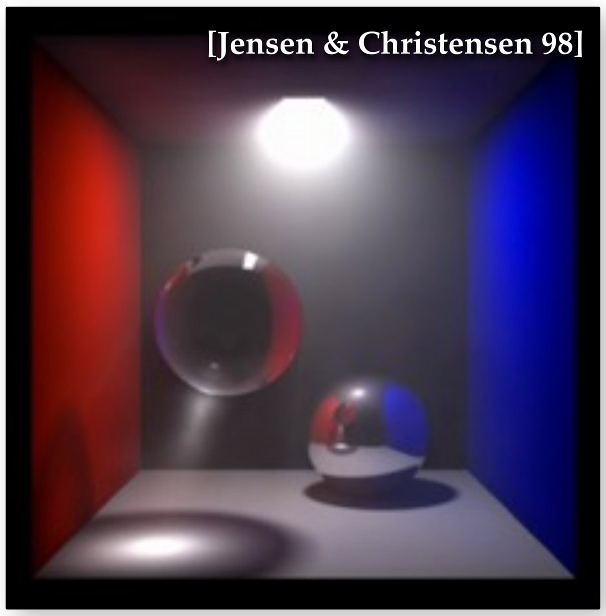


✓ generally produce high-quality images faster

- Methods such as volumetric photon mapping do not suffer from these problems [click]
- They typically produce high-quality results much faster than unbiased techniques [click]
- and is one of the few algorithms that is robust to caustic paths [click]
- However, photon mapping introduces bias.
- It is consistent, though, which means that if we use an infinite number of photons, we will get the correct solution.
- but, this is of little practical value since we obviously cannot store an unlimited number of photons.

Previous Work: Volumetric PM

- Volumetric Photon Mapping
  - [Jensen & Christensen 98]
  - [Jarosz et al. 08]



✓ generally produce high-quality images faster

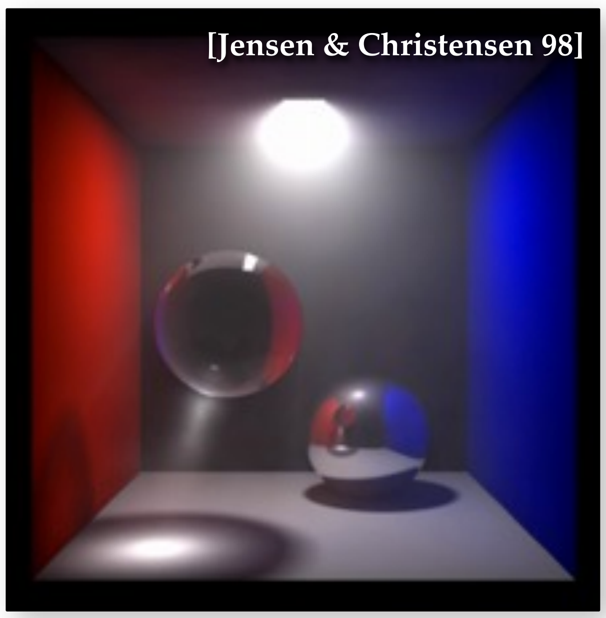
✓ robust to caustic paths

Thursday, 6 September 12

- Methods such as volumetric photon mapping do not suffer from these problems [click]
- They typically produce high-quality results much faster than unbiased techniques [click]
- and is one of the few algorithms that is robust to caustic paths [click]
- However, photon mapping introduces bias.
- It is consistent, though, which means that if we use an infinite number of photons, we will get the correct solution.
- but, this is of little practical value since we obviously cannot store an unlimited number of photons.

Previous Work: Volumetric PM

- Volumetric Photon Mapping
  - [Jensen & Christensen 98]
  - [Jarosz et al. 08]



✓ generally produce high-quality images faster

✓ robust to caustic paths

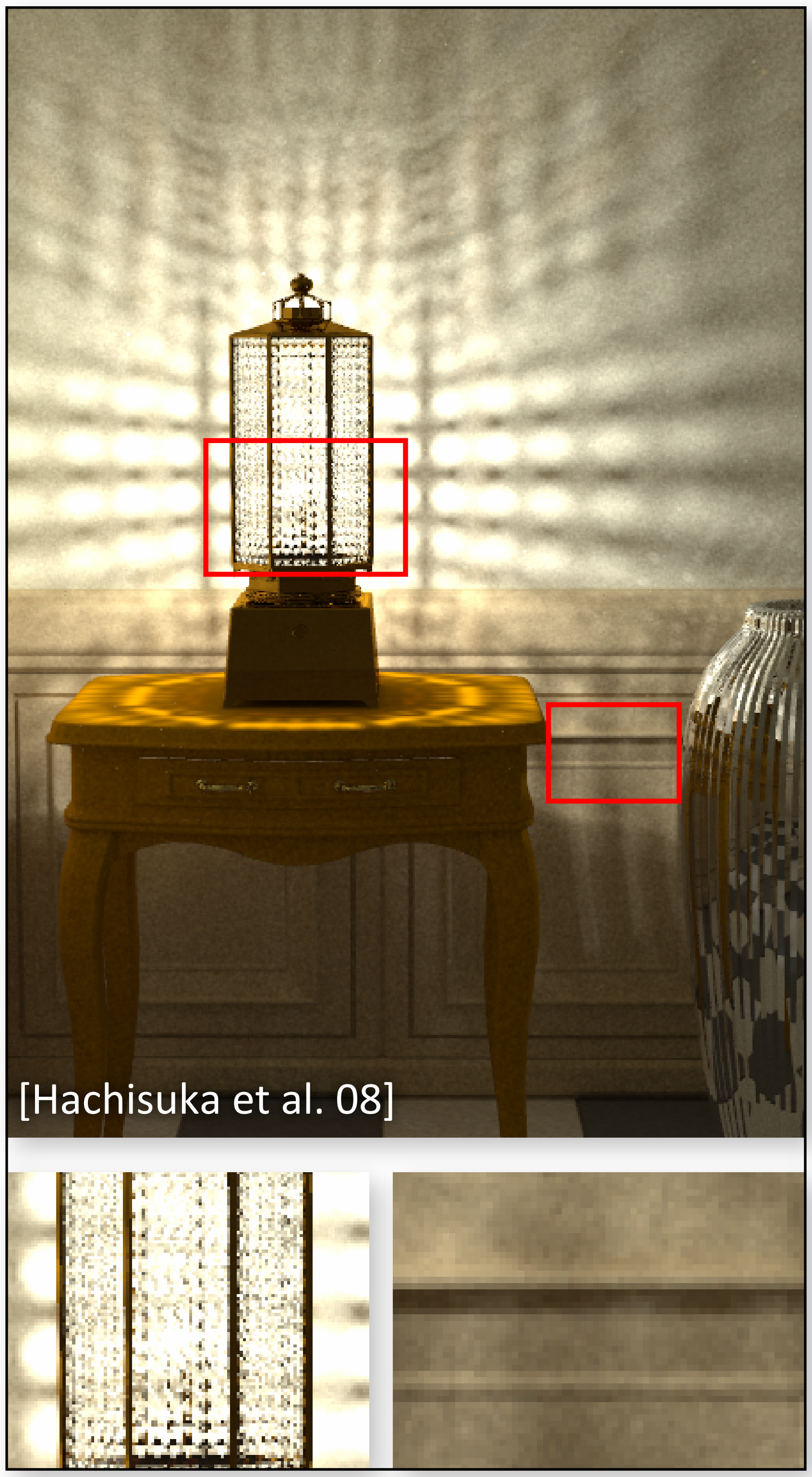
– biased (but consistent)

Thursday, 6 September 12

- Methods such as volumetric photon mapping do not suffer from these problems [click]
- They typically produce high-quality results much faster than unbiased techniques [click]
- and is one of the few algorithms that is robust to caustic paths [click]
- However, photon mapping introduces bias.
- It is consistent, though, which means that if we use an infinite number of photons, we will get the correct solution.
- but, this is of little practical value since we obviously cannot store an unlimited number of photons.

Previous Work: Progressive PM

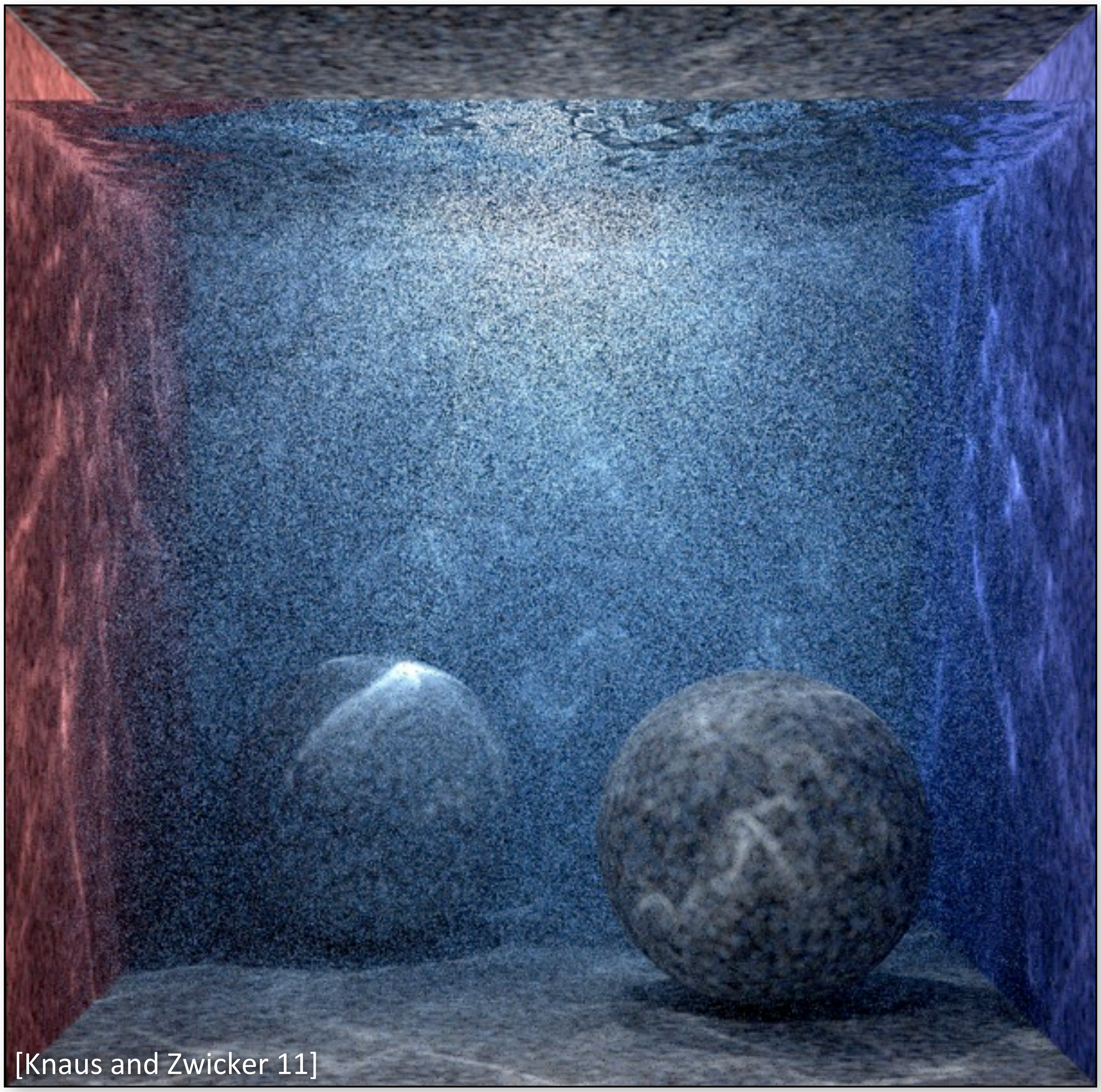
- Progressive photon mapping
  - [Hachisuka et al. 08]
  - [Hachisuka and Jensen 09]
  - [Knaus and Zwicker 11]



Thursday, 6 September 12

- Hachisuka et al. introduced a practical way to alleviate this memory constraint
- They introduced progressive photon mapping, which shows how to eliminate bias and noise simultaneously in photon mapping without having to storing an unlimited number of photons

Previous Work: Progressive PM



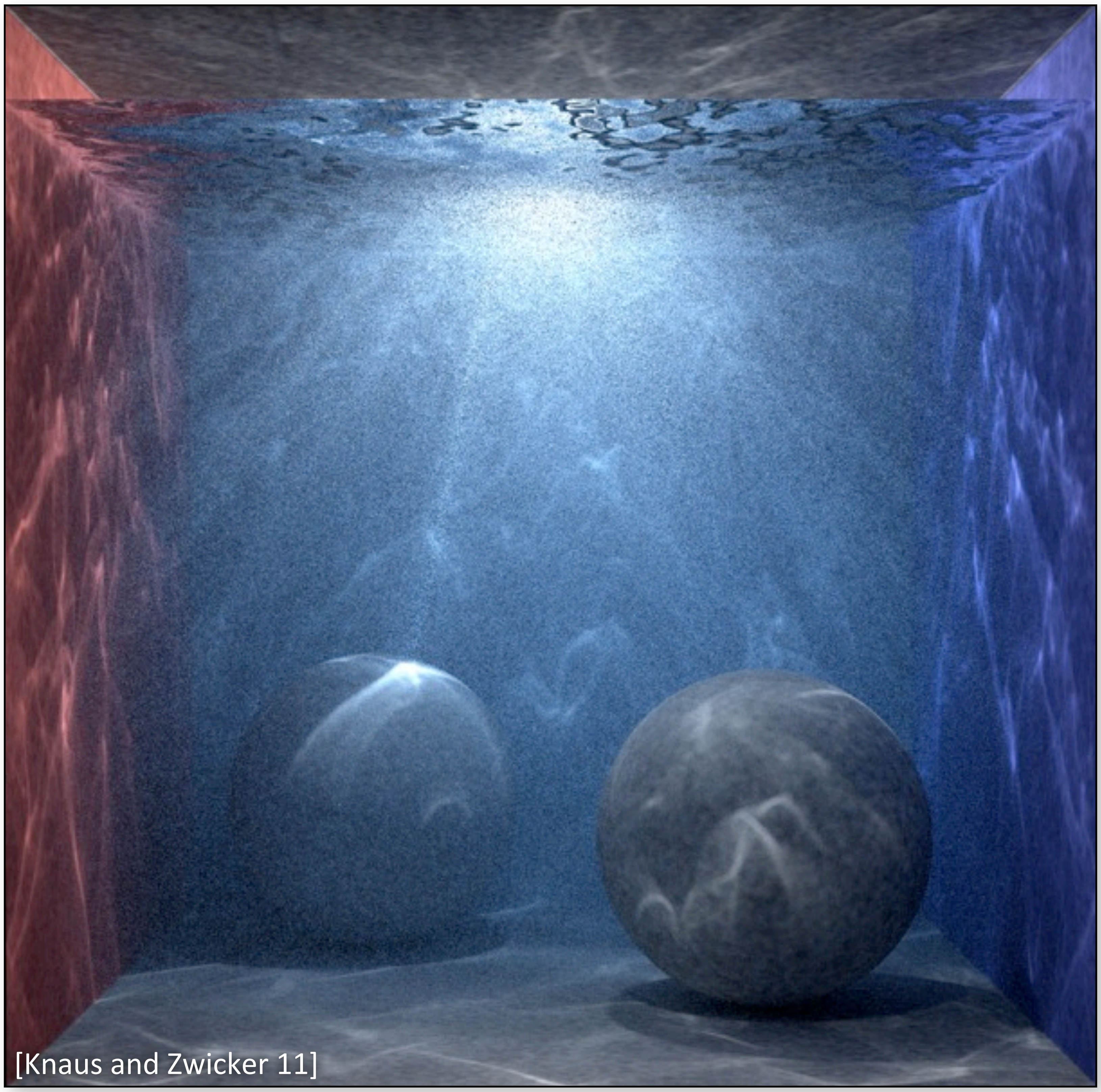
1 iteration

2 million photons

Thursday, 6 September 12

- Instead of storing all photons needed to obtain a converged result [click]
- Photons are traced and discarded progressively [click]
- the rendered image is updated after each photon tracing pass [click]
- in such a way that the approximation converges to the correct solution in the limit.

Previous Work: Progressive PM

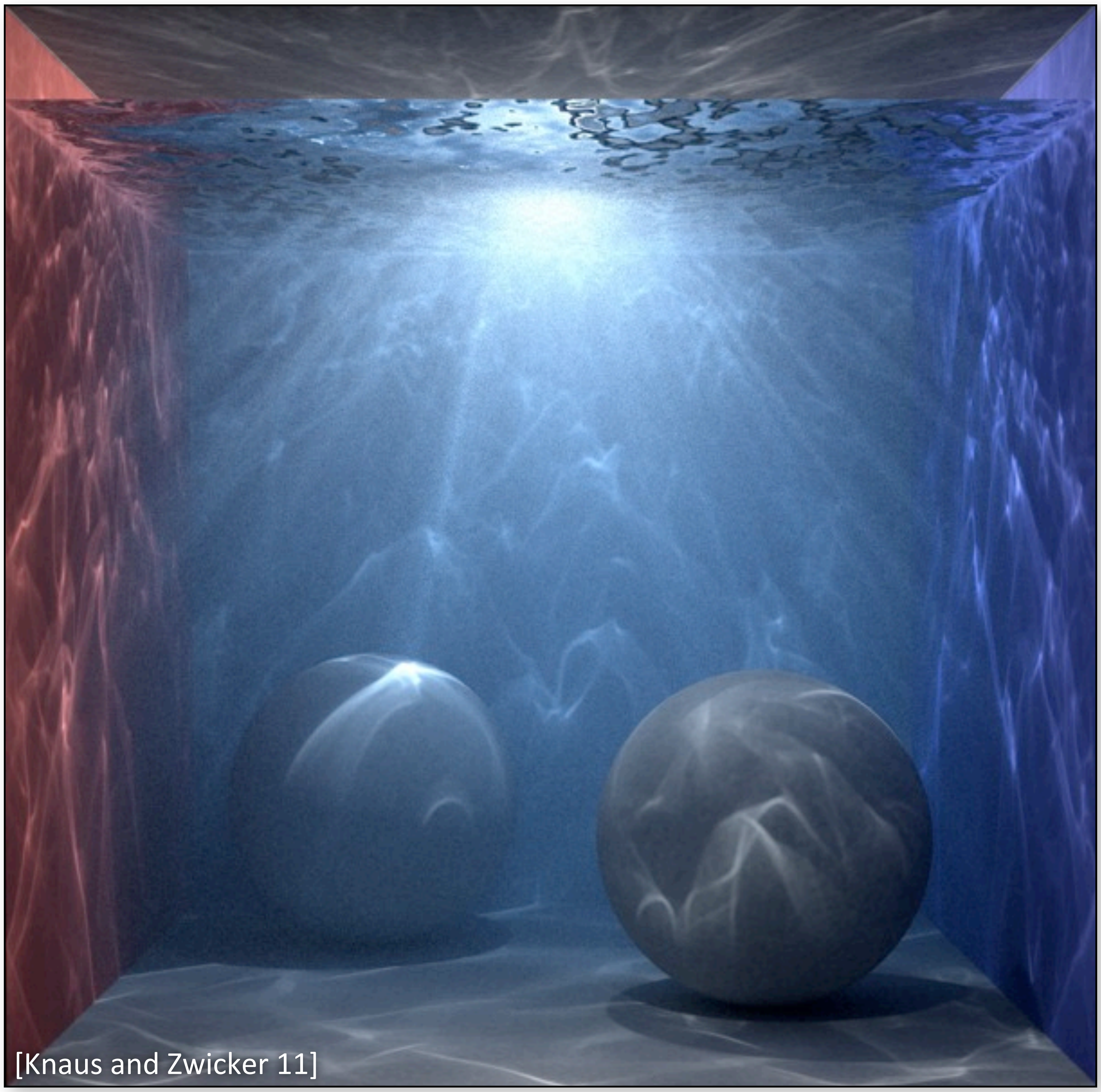


10 iterations

20 million photons

- Instead of storing all photons needed to obtain a converged result [click]
- Photons are traced and discarded progressively [click]
- the rendered image is updated after each photon tracing pass [click]
- in such a way that the approximation converges to the correct solution in the limit.

Previous Work: Progressive PM

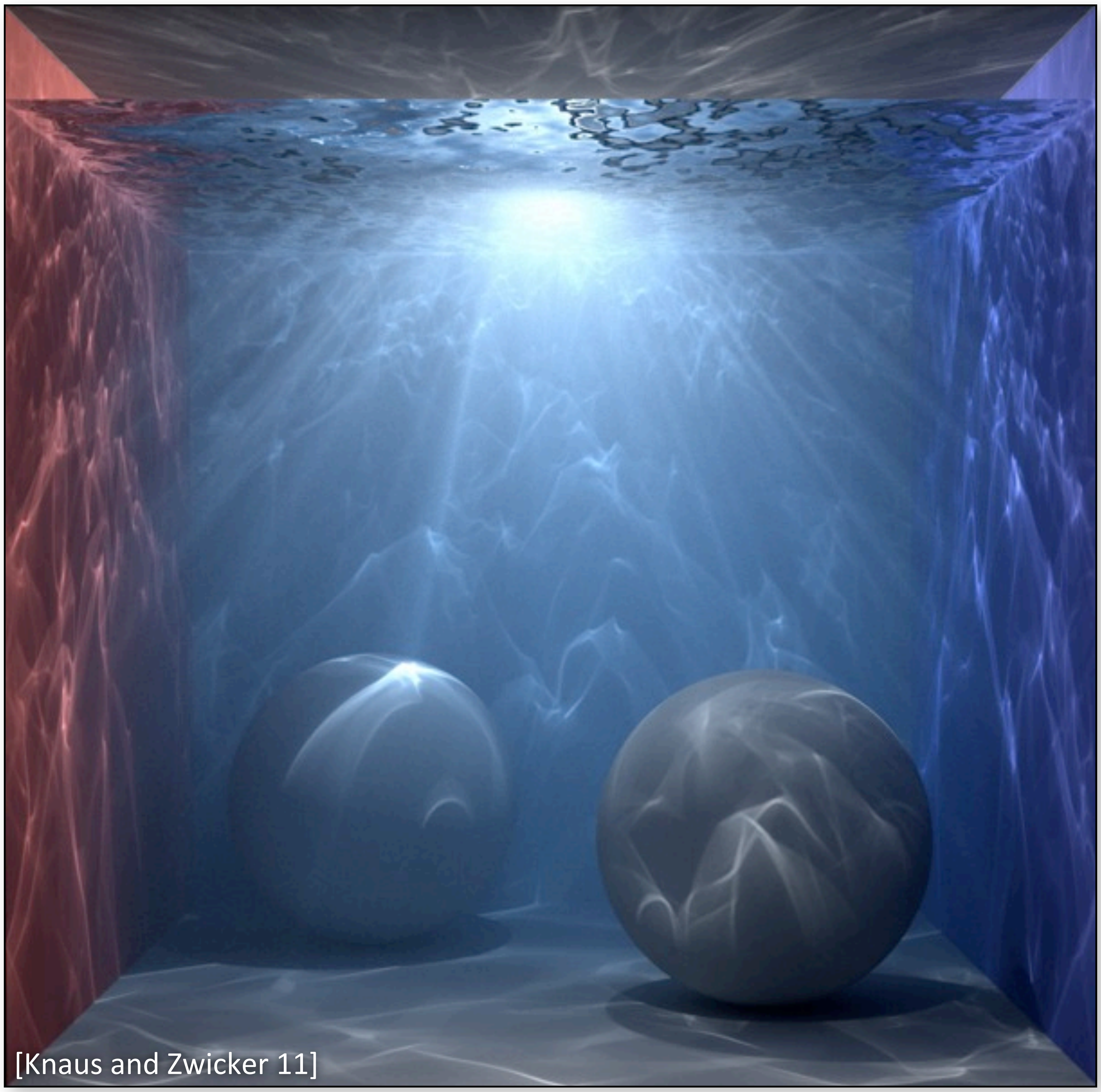


100 iterations

200 million photons

- Instead of storing all photons needed to obtain a converged result [click]
- Photons are traced and discarded progressively [click]
- the rendered image is updated after each photon tracing pass [click]
- in such a way that the approximation converges to the correct solution in the limit.

Previous Work: Progressive PM



1000 iterations

2 billion photons

- Instead of storing all photons needed to obtain a converged result [click]
- Photons are traced and discarded progressively [click]
- the rendered image is updated after each photon tracing pass [click]
- in such a way that the approximation converges to the correct solution in the limit.

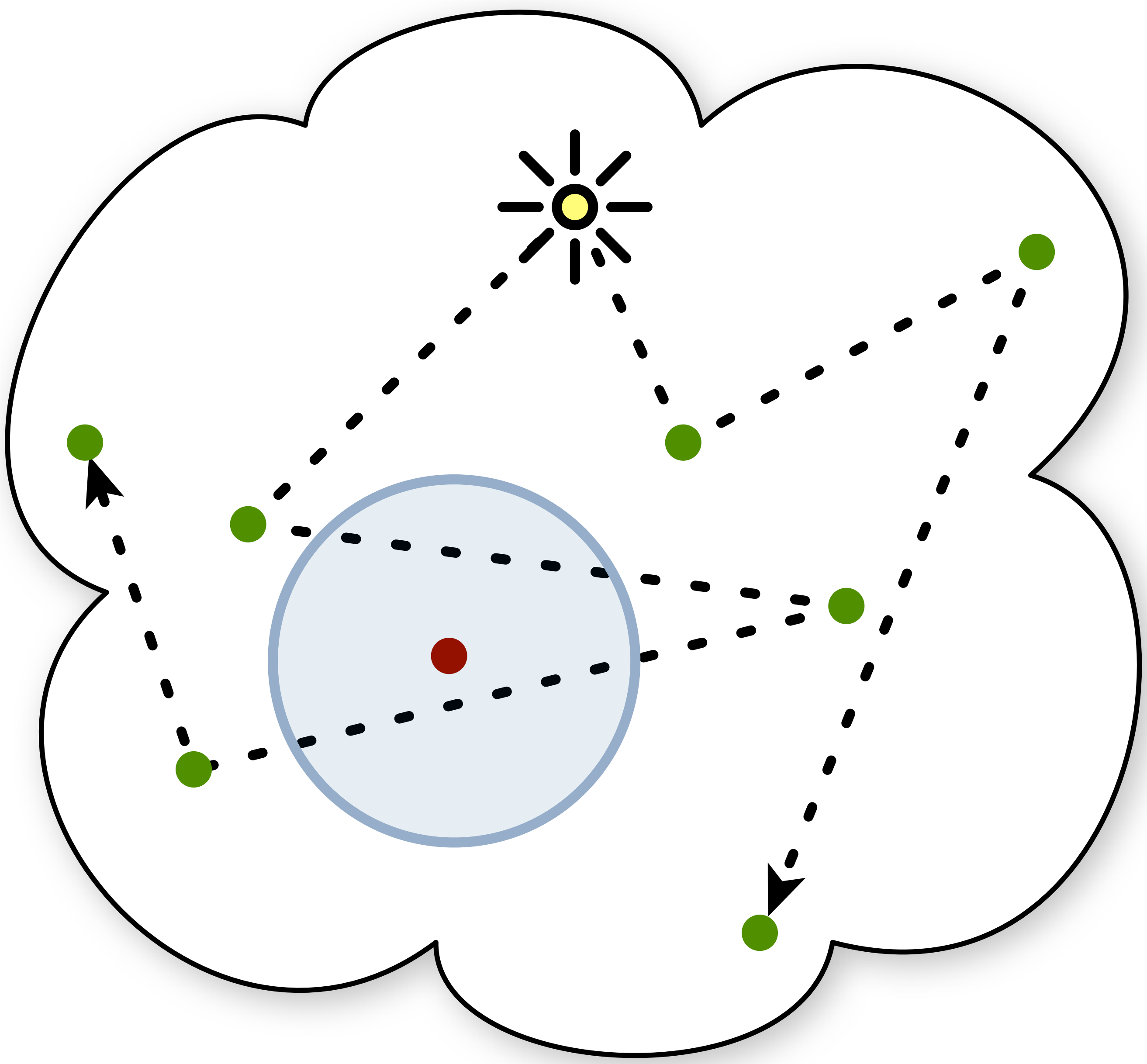
Previous Work: Photon Beams

- Photon Beams [Jarosz et al. 11]

- This past SIGGRAPH, we introduced a new technique for rendering participating media called photon beams
- The central observation that we made is that volumetric photon mapping throws away a lot of potentially useful information between the shooting stage and the density estimation stage
- In particular, photon mapping traces random-walk paths from the light, and then stores the vertices of these paths as photons
- We made the observation that if we stored the entire path of the photons, and not just the scattering locations, then we get a higher sampling density within the medium, and obtain higher quality renderings at virtually no extra cost.

Previous Work: Photon Beams

- Photon Beams [Jarosz et al. 11]

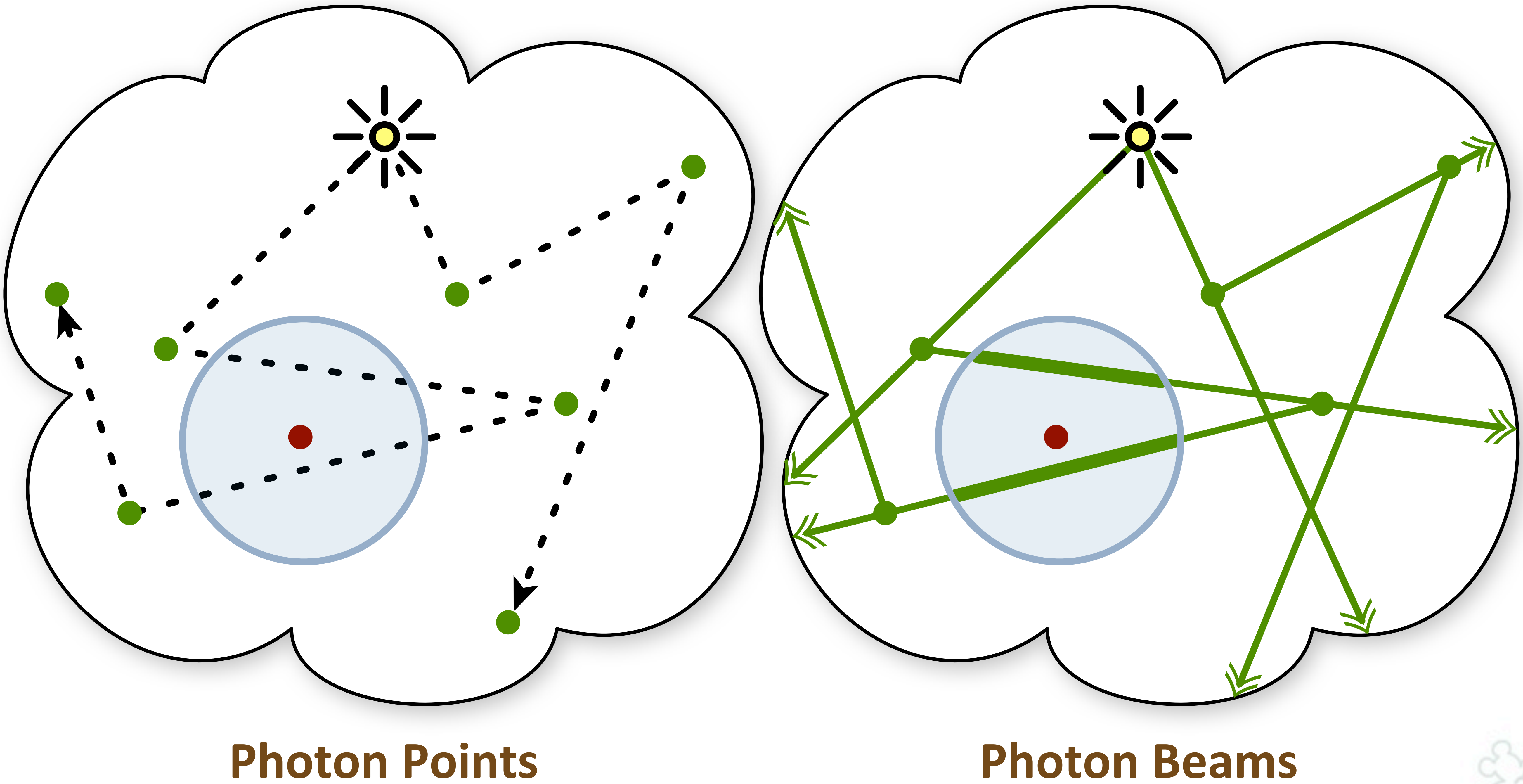


Photon Points

- This past SIGGRAPH, we introduced a new technique for rendering participating media called photon beams
- The central observation that we made is that volumetric photon mapping throws away a lot of potentially useful information between the shooting stage and the density estimation stage
- In particular, photon mapping traces random-walk paths from the light, and then stores the vertices of these paths as photons
- We made the observation that if we stored the entire path of the photons, and not just the scattering locations, then we get a higher sampling density within the medium, and obtain higher quality renderings at virtually no extra cost.

Previous Work: Photon Beams

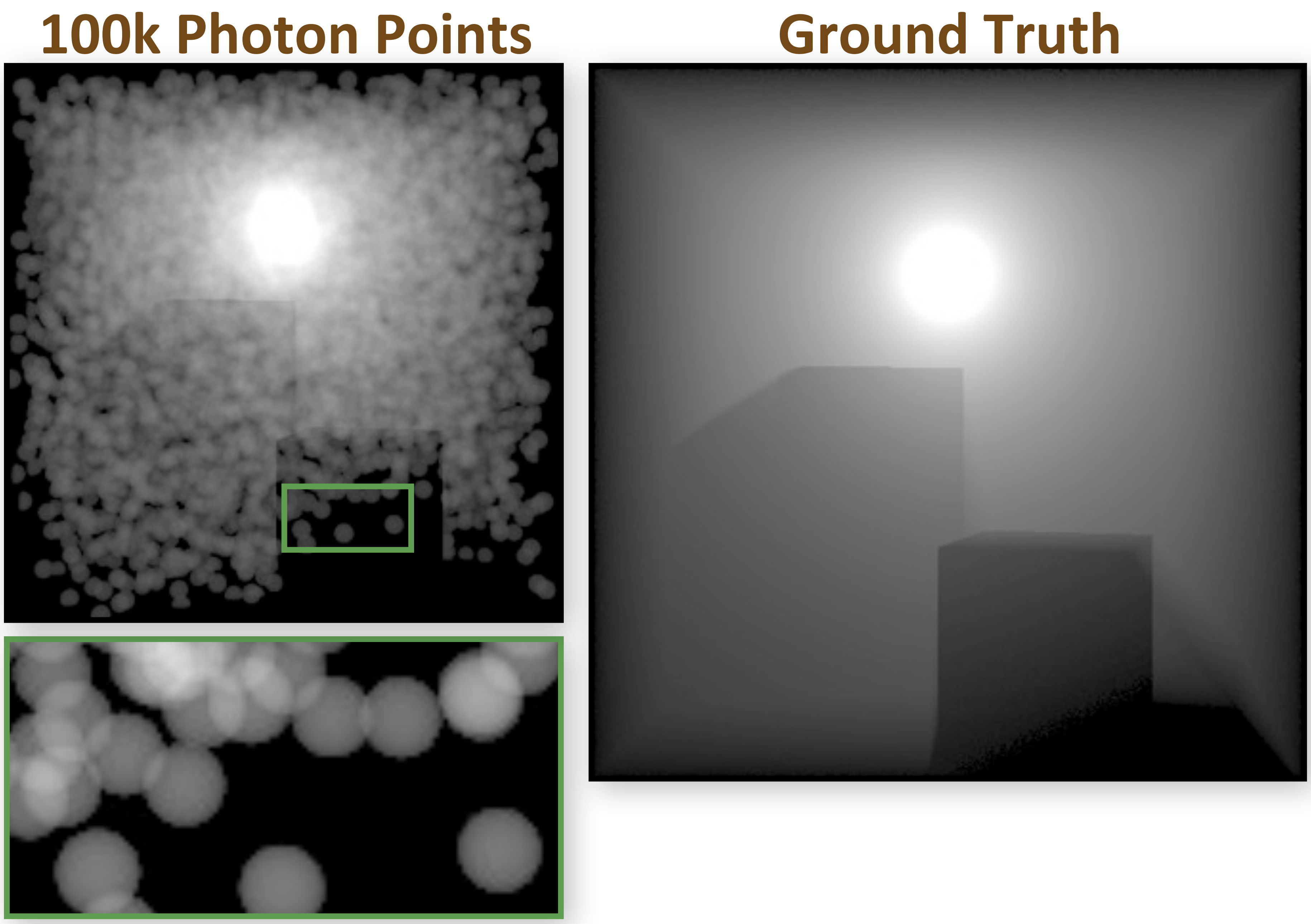
- Photon Beams [Jarosz et al. 11]



Thursday, 6 September 12

- This past SIGGRAPH, we introduced a new technique for rendering participating media called photon beams
- The central observation that we made is that volumetric photon mapping throws away a lot of potentially useful information between the shooting stage and the density estimation stage
- In particular, photon mapping traces random-walk paths from the light, and then stores the vertices of these paths as photons
- We made the observation that if we stored the entire path of the photons, and not just the scattering locations, then we get a higher sampling density within the medium, and obtain higher quality renderings at virtually no extra cost.

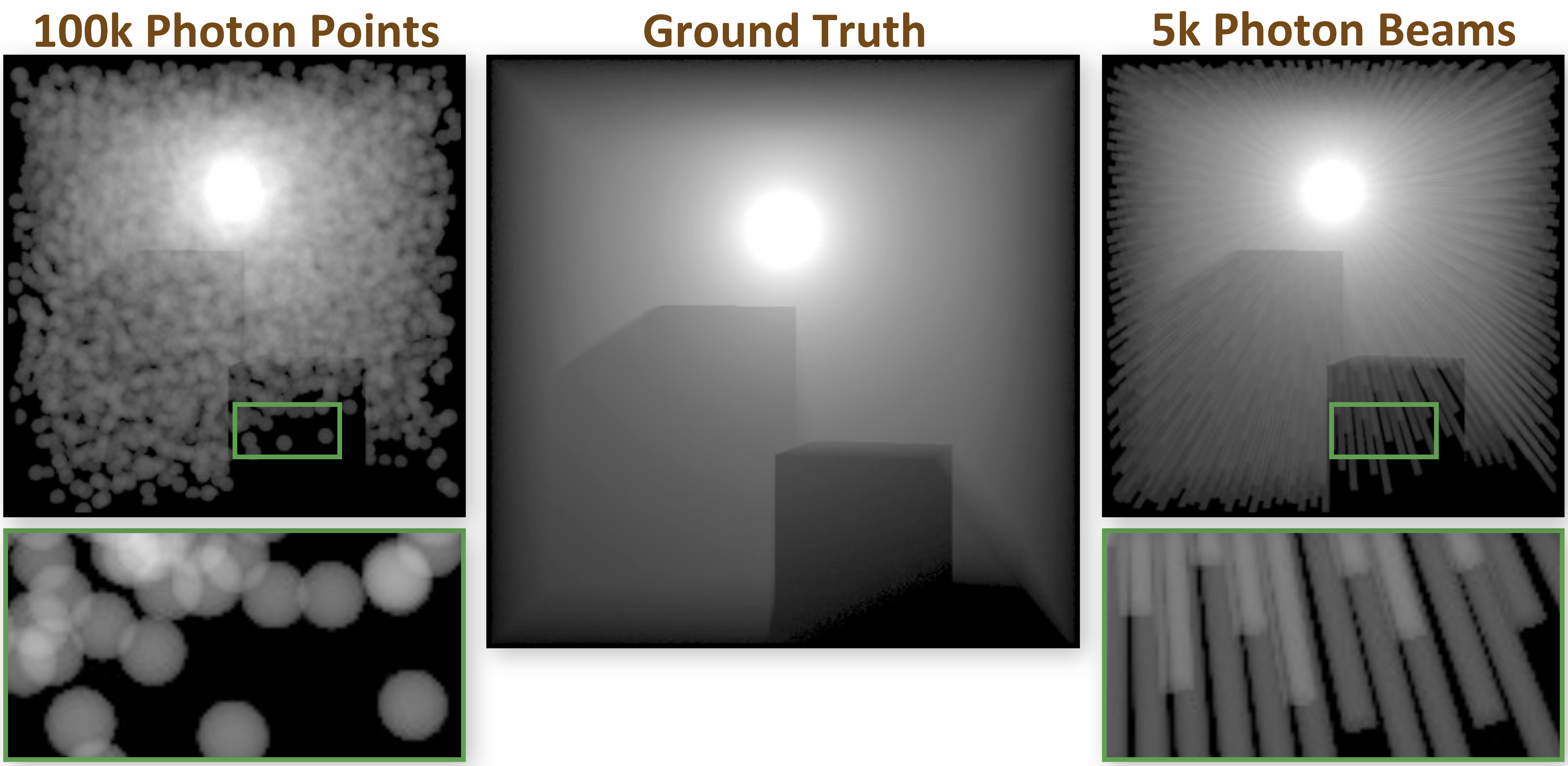
Photon Points vs. Photon Beams



Thursday, 6 September 12

- In practice, photon mapping effectively blurs each photon point into a small disc.
- On the other hand, [click] photon beams blurs each photon path into a thick line segment
- Now, even though beams produce higher quality results using less photons, the generated images are still biased, since each beam is blurred with a finite width

Photon Points vs. Photon Beams



Thursday, 6 September 12

- In practice, photon mapping effectively blurs each photon point into a small disc.
- On the other hand, [click] photon beams blurs each photon path into a thick line segment
- Now, even though beams produce higher quality results using less photons, the generated images are still biased, since each beam is blurred with a finite width

Goals

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

- The goal of our paper is to combine the benefits of photon beams and progressive photon mapping
- We would like to use the more efficient photon beams approach but formulate a progressive algorithm that will converge to the correct result [click]
- We call this combination: Progressive Photon Beams
- The end result is an efficient algorithm which: [click]
  - is robust to complex light paths [click]
  - converges to unbiased result in finite memory [click]
  - handles heterogeneous media, and [click]
  - supports progressive updates for interactive preview

Goals

- Combine benefits of:
  - photon beams
  - progressive photon mapping
- Progressive Photon Beams

- The goal of our paper is to combine the benefits of photon beams and progressive photon mapping
- We would like to use the more efficient photon beams approach but formulate a progressive algorithm that will converge to the correct result [click]
- We call this combination: Progressive Photon Beams
- The end result is an efficient algorithm which: [click]
  - is robust to complex light paths [click]
  - converges to unbiased result in finite memory [click]
  - handles heterogeneous media, and [click]
  - supports progressive updates for interactive preview

Goals

- Combine benefits of:
  - photon beams
  - progressive photon mapping
- **Progressive Photon Beams**
  - **robust** to complex light paths

- The goal of our paper is to combine the benefits of photon beams and progressive photon mapping
- We would like to use the more efficient photon beams approach but formulate a progressive algorithm that will converge to the correct result [click]
- We call this combination: Progressive Photon Beams
- The end result is an efficient algorithm which: [click]
  - is robust to complex light paths [click]
  - converges to unbiased result in finite memory [click]
  - handles heterogeneous media, and [click]
  - supports progressive updates for interactive preview

Goals

- Combine benefits of:
  - photon beams
  - progressive photon mapping
- **Progressive Photon Beams**
  - **robust** to complex light paths
  - converges to **unbiased** result in **finite** memory

Thursday, 6 September 12

- The goal of our paper is to combine the benefits of photon beams and progressive photon mapping
- We would like to use the more efficient photon beams approach but formulate a progressive algorithm that will converge to the correct result [click]
- We call this combination: Progressive Photon Beams
- The end result is an efficient algorithm which: [click]
  - is robust to complex light paths [click]
  - converges to unbiased result in finite memory [click]
  - handles heterogeneous media, and [click]
  - supports progressive updates for interactive preview

Goals

- Combine benefits of:
  - photon beams
  - progressive photon mapping
- **Progressive Photon Beams**
  - **robust** to complex light paths
  - converges to **unbiased** result in **finite** memory
  - handles **heterogeneous** media

- The goal of our paper is to combine the benefits of photon beams and progressive photon mapping
- We would like to use the more efficient photon beams approach but formulate a progressive algorithm that will converge to the correct result [click]
- We call this combination: Progressive Photon Beams
- The end result is an efficient algorithm which: [click]
  - is robust to complex light paths [click]
  - converges to unbiased result in finite memory [click]
  - handles heterogeneous media, and [click]
  - supports progressive updates for interactive preview

Goals

- Combine benefits of:
  - photon beams
  - progressive photon mapping
- **Progressive Photon Beams**
  - **robust** to complex light paths
  - converges to **unbiased** result in **finite** memory
  - handles **heterogeneous** media
  - supports **progressive/interactive** updates

- The goal of our paper is to combine the benefits of photon beams and progressive photon mapping
- We would like to use the more efficient photon beams approach but formulate a progressive algorithm that will converge to the correct result [click]
- We call this combination: Progressive Photon Beams
- The end result is an efficient algorithm which: [click]
  - is robust to complex light paths [click]
  - converges to unbiased result in finite memory [click]
  - handles heterogeneous media, and [click]
  - supports progressive updates for interactive preview

Progressive Photon Beams

- At a high-level our approach proceeds similarly as previous PPM techniques.
- The main idea is to generate a sequence of render passes (one of which is shown here), where each pass uses an independent collection of photon beams. The output of our algorithm is a running average of the passes so far (which I'll show on the right).
- Since photon beams are biased, by definition, if we just averaged several independent passes, the variance would diminish, but the final result would still be biased [click]

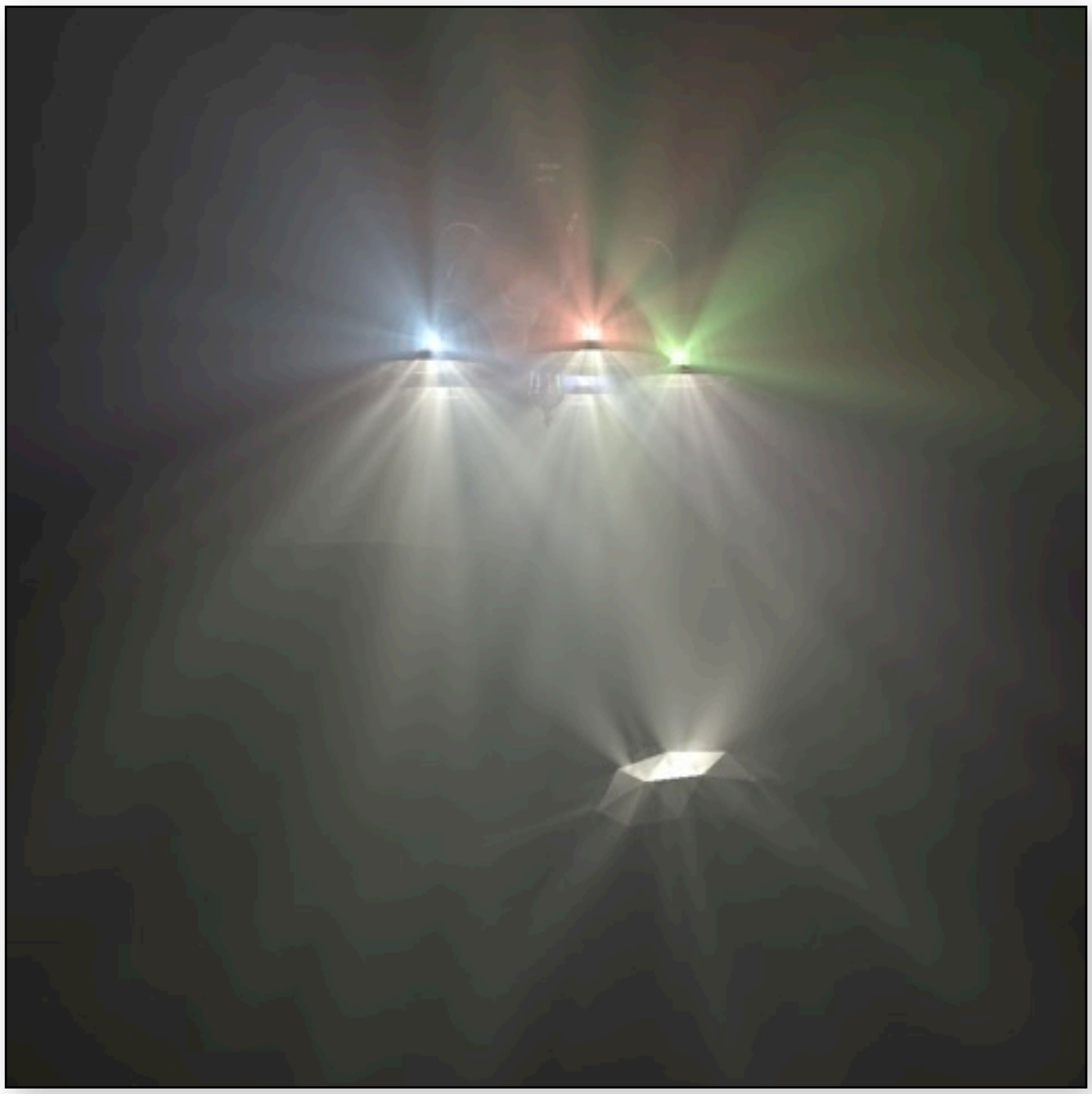
Progressive Photon Beams

Pass 1

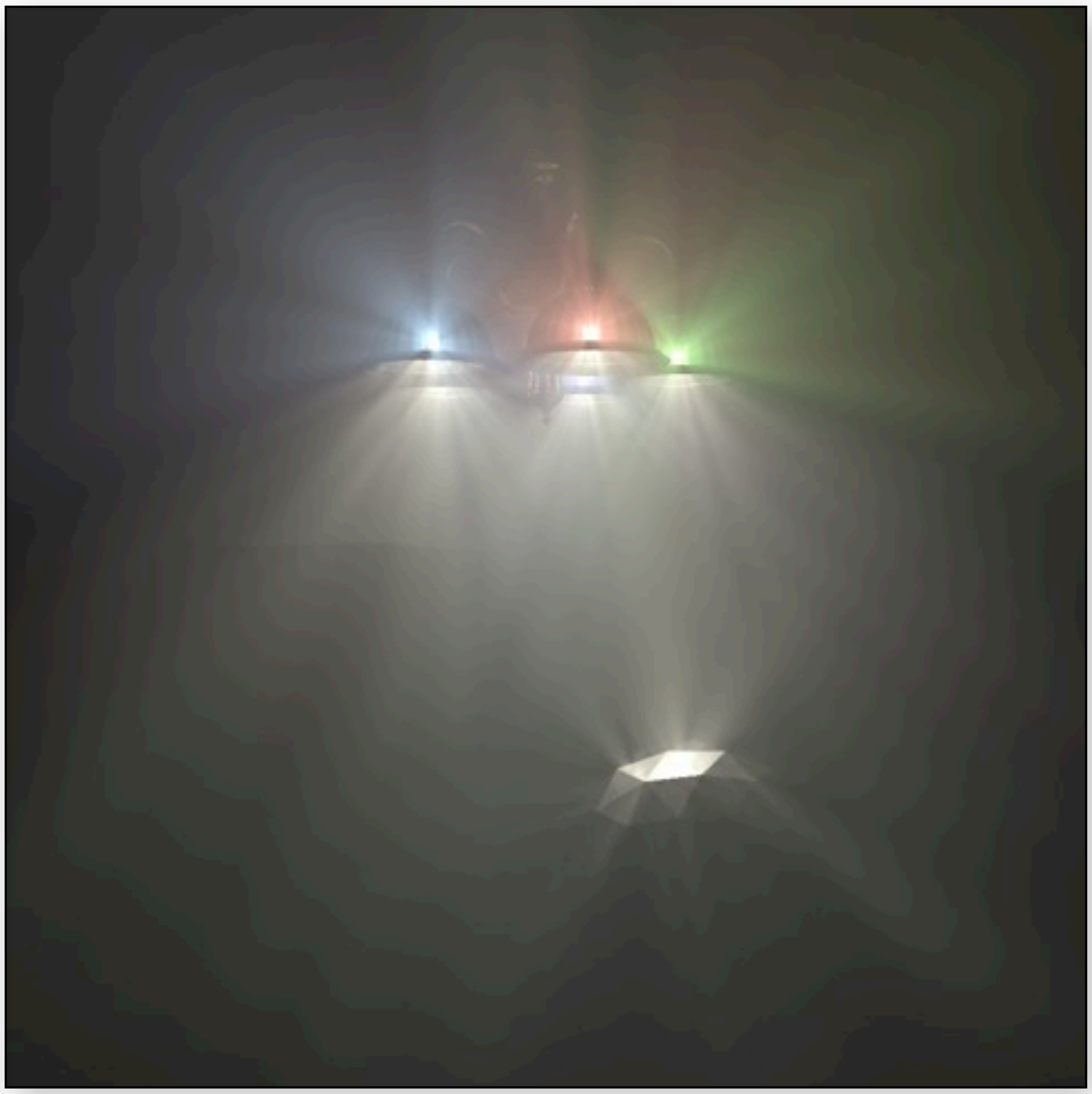
Thursday, 6 September 12

- At a high-level our approach proceeds similarly as previous PPM techniques.
- The main idea is to generate a sequence of render passes (one of which is shown here), where each pass uses an independent collection of photon beams. The output of our algorithm is a running average of the passes so far (which I'll show on the right).
- Since photon beams are biased, by definition, if we just averaged several independent passes, the variance would diminish, but the final result would still be biased [click]

Progressive Photon Beams



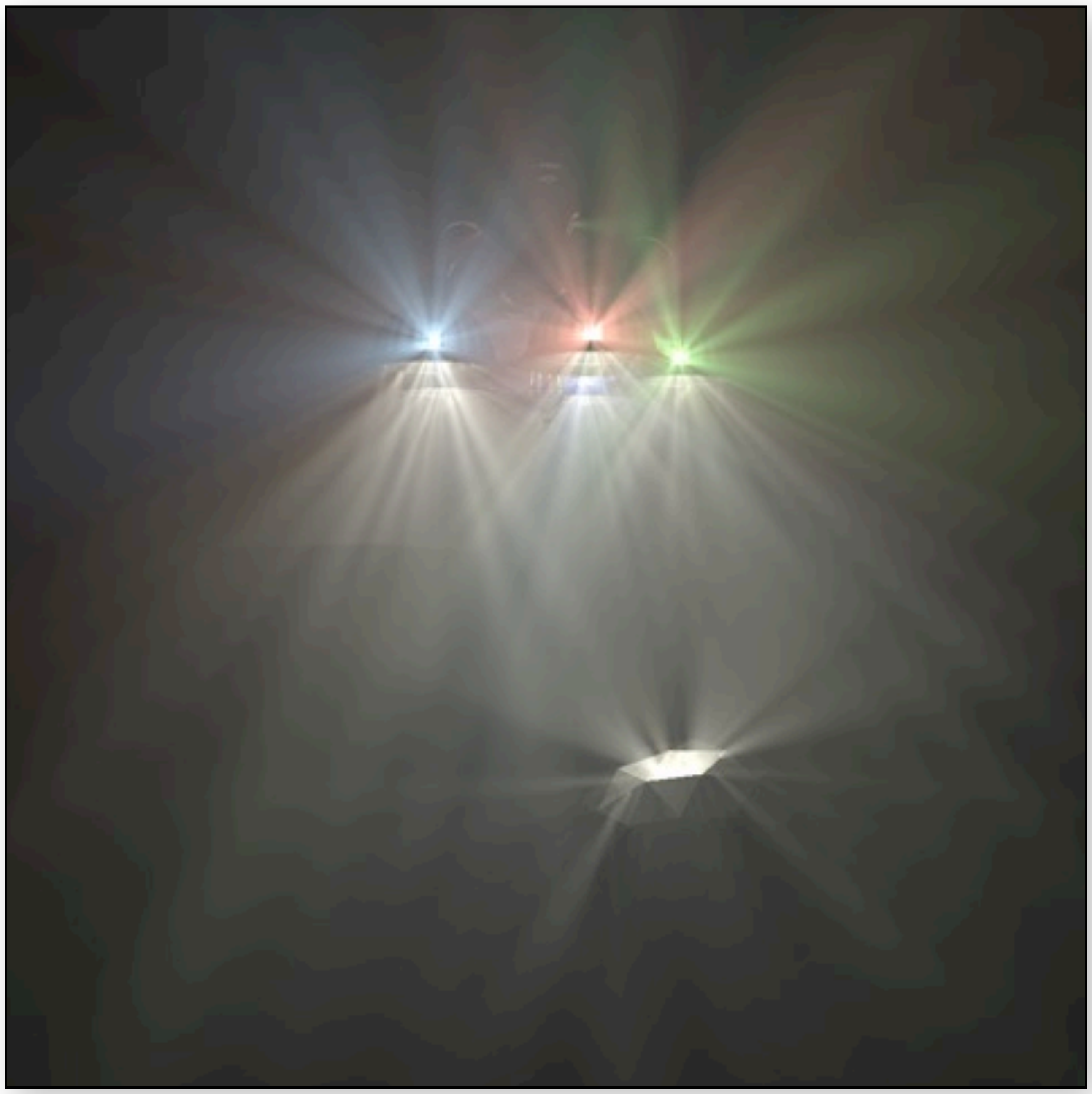
Pass 2



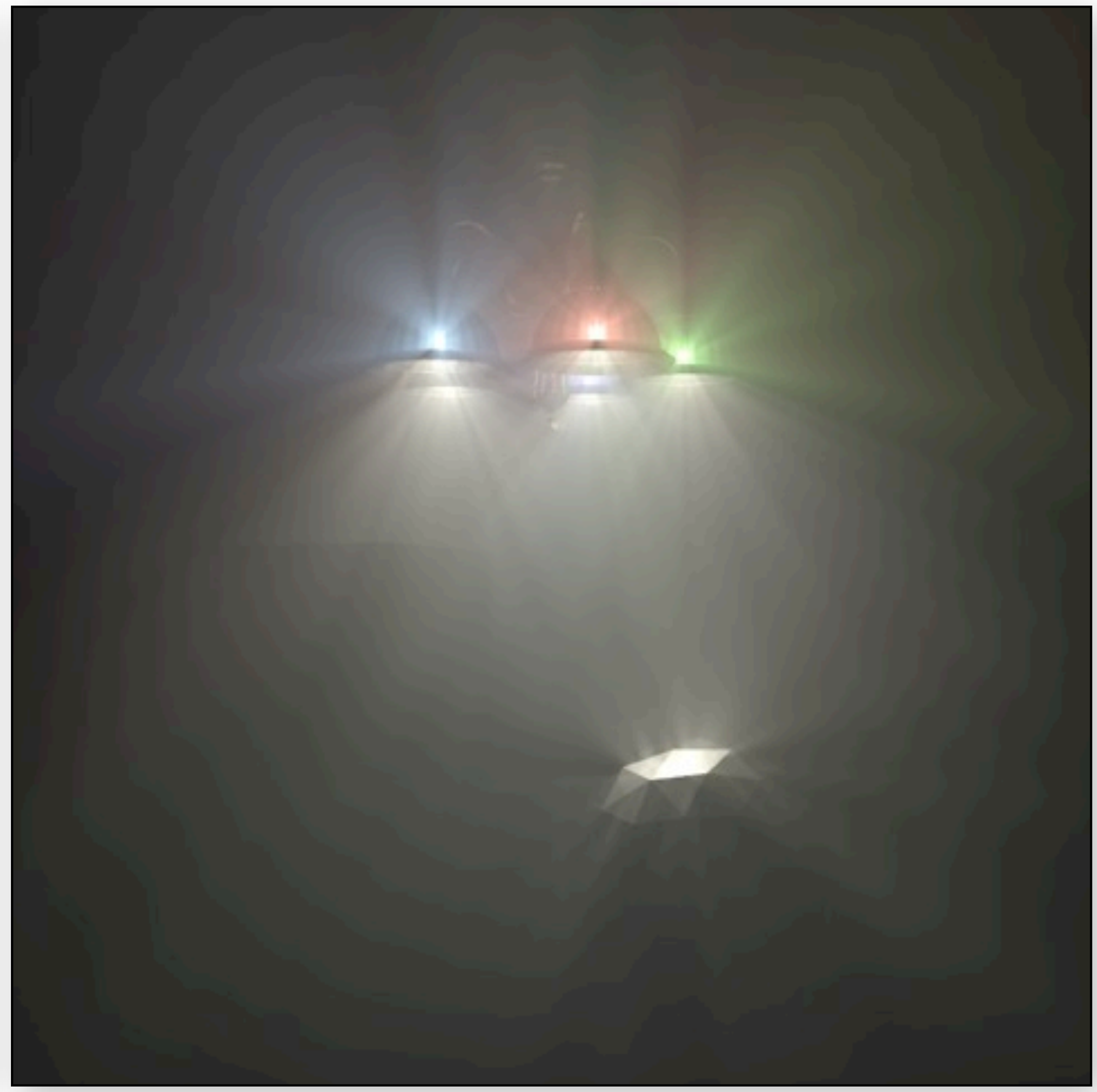
Average of Passes 1..2

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



Pass 4

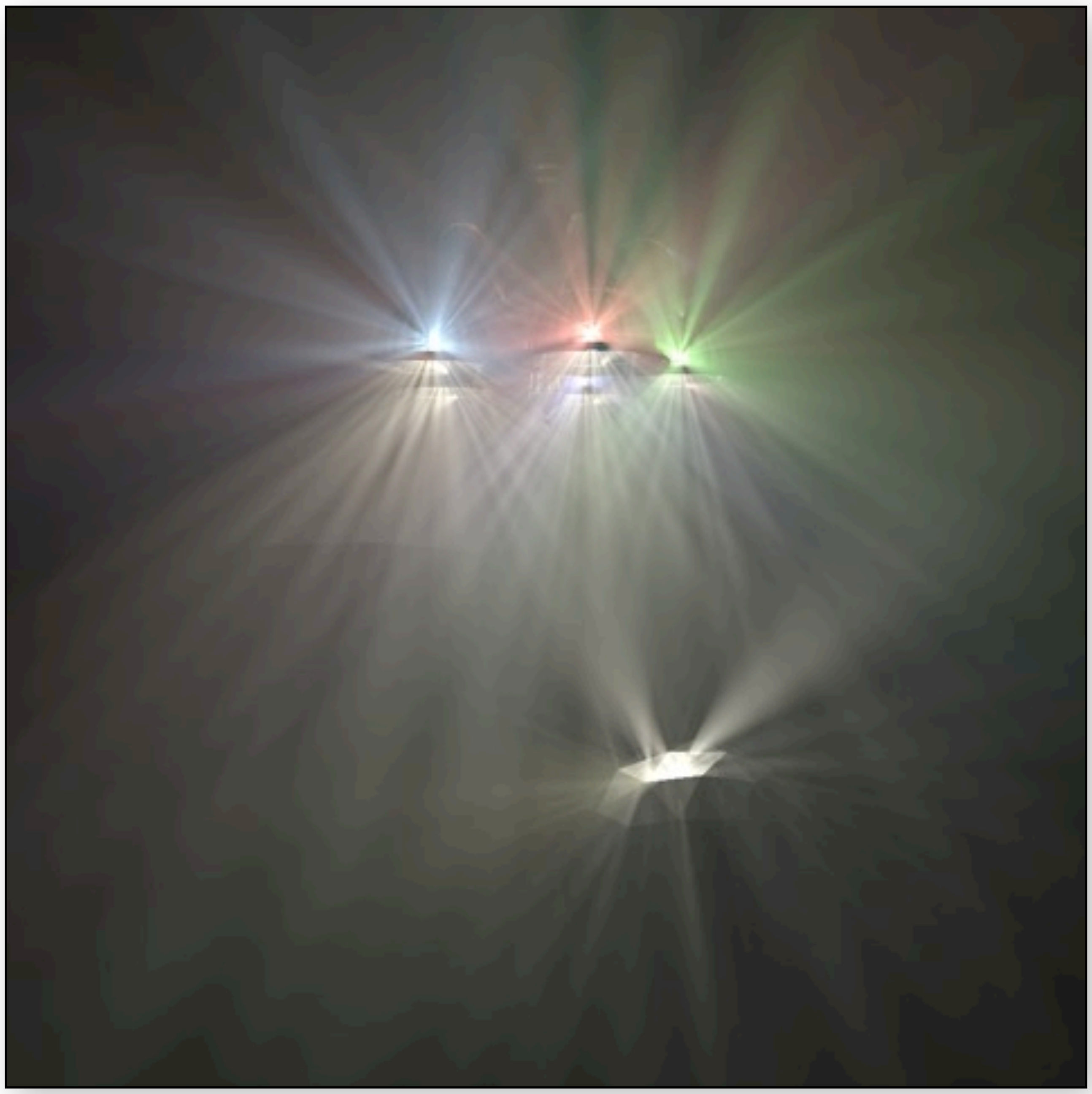


Average of Passes 1..4

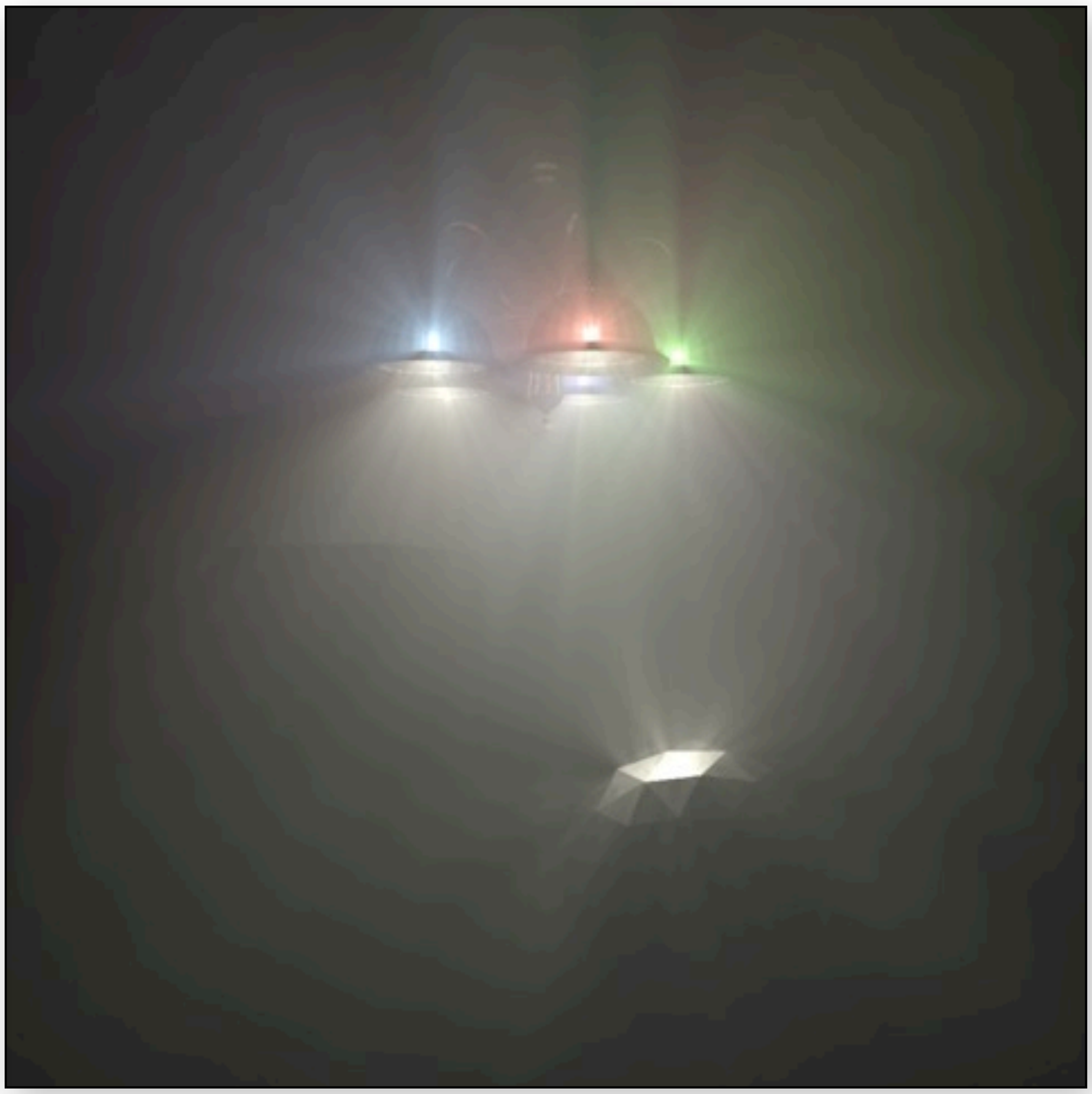
Thursday, 6 September 12

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



Pass 8

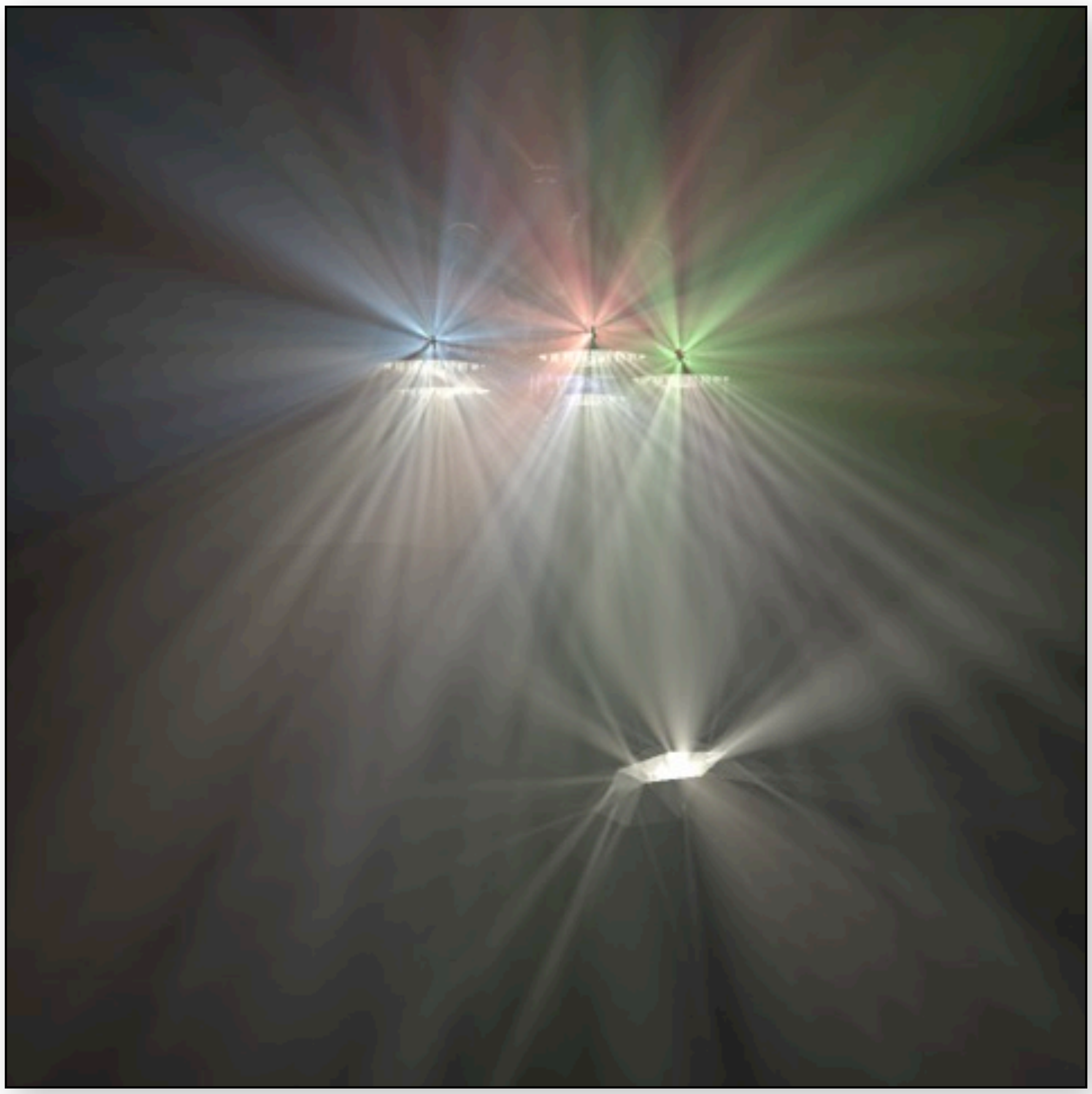


Average of Passes 1..8

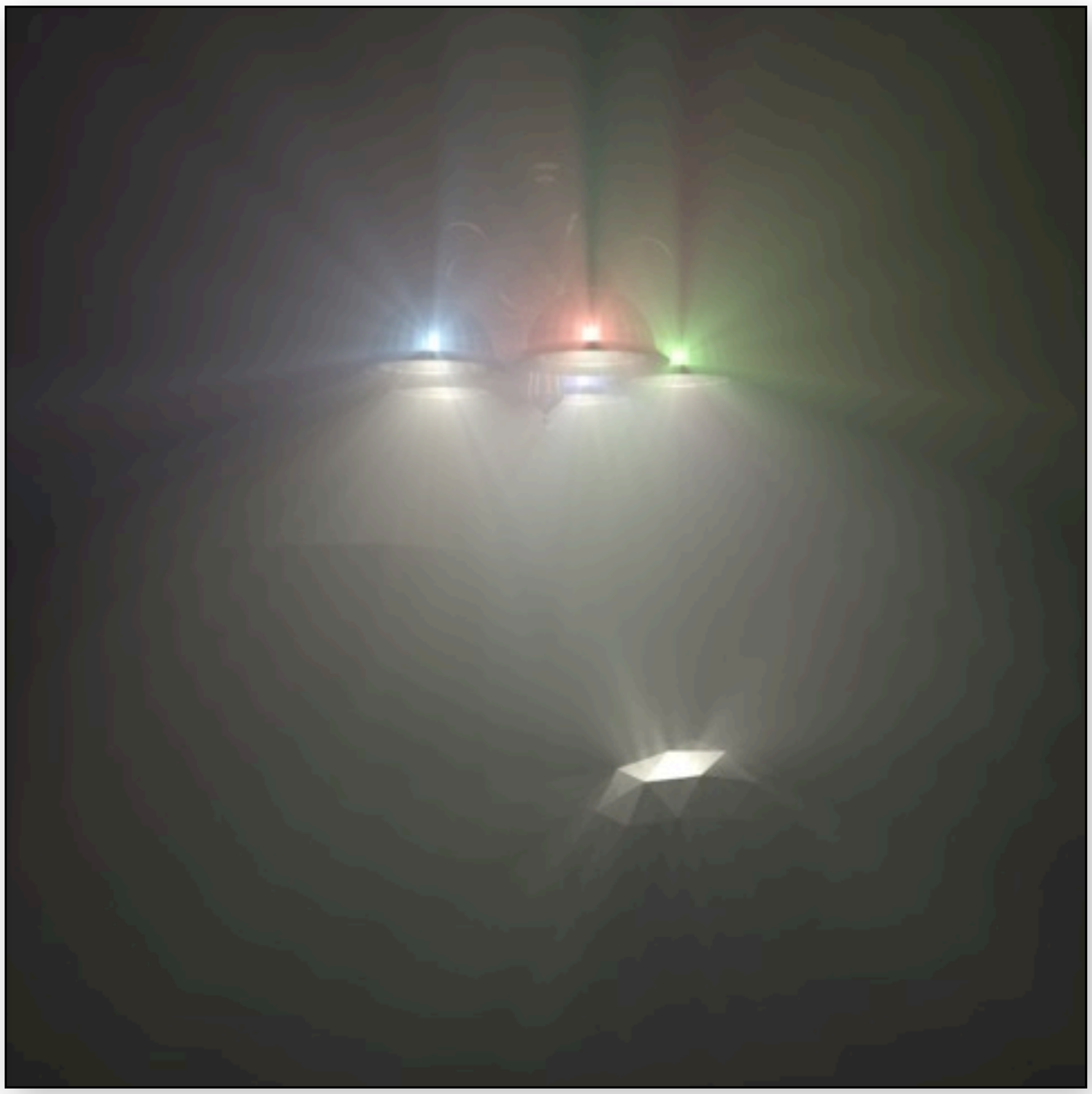
Thursday, 6 September 12

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



Pass 16

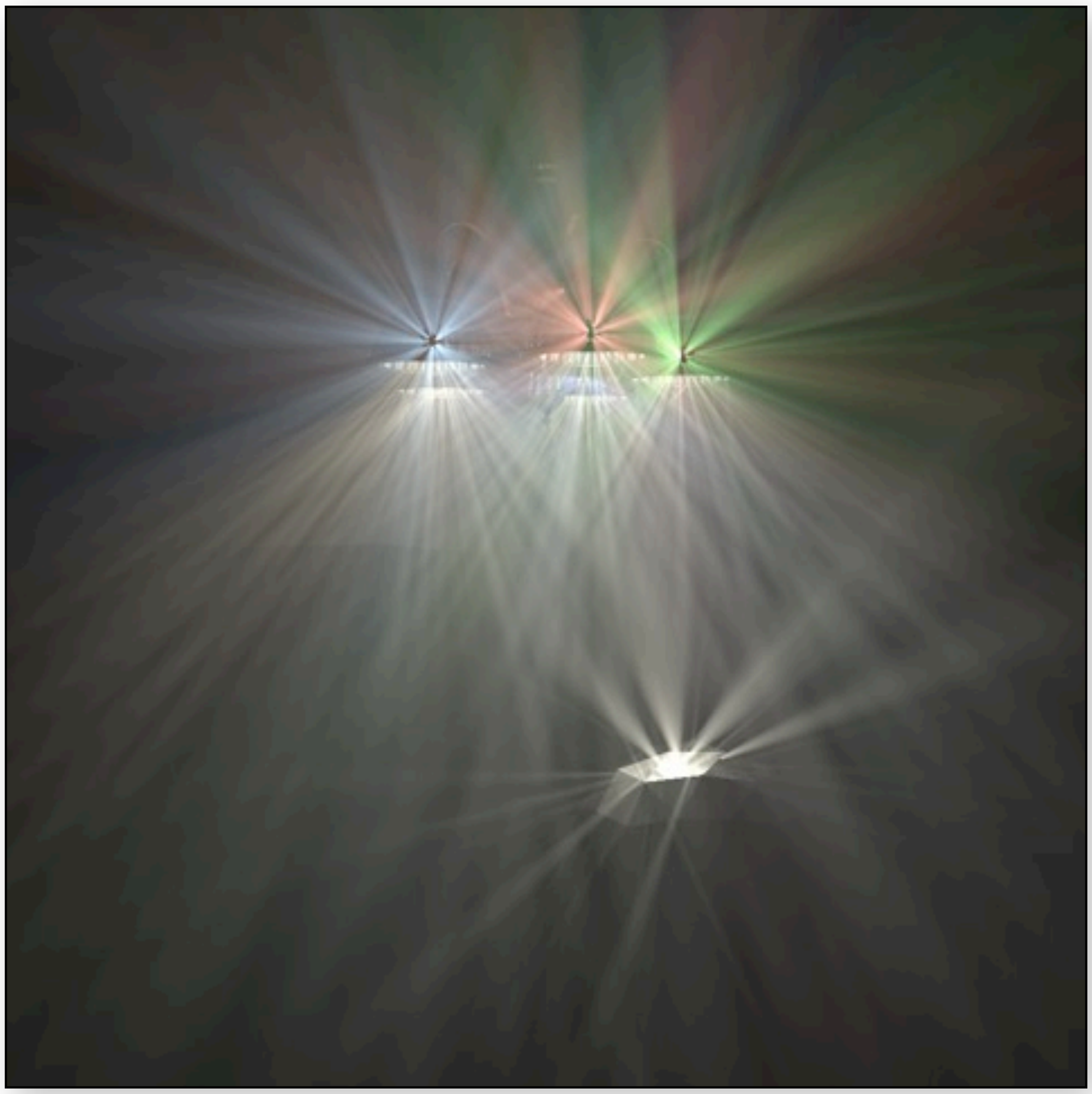


Average of Passes 1..16

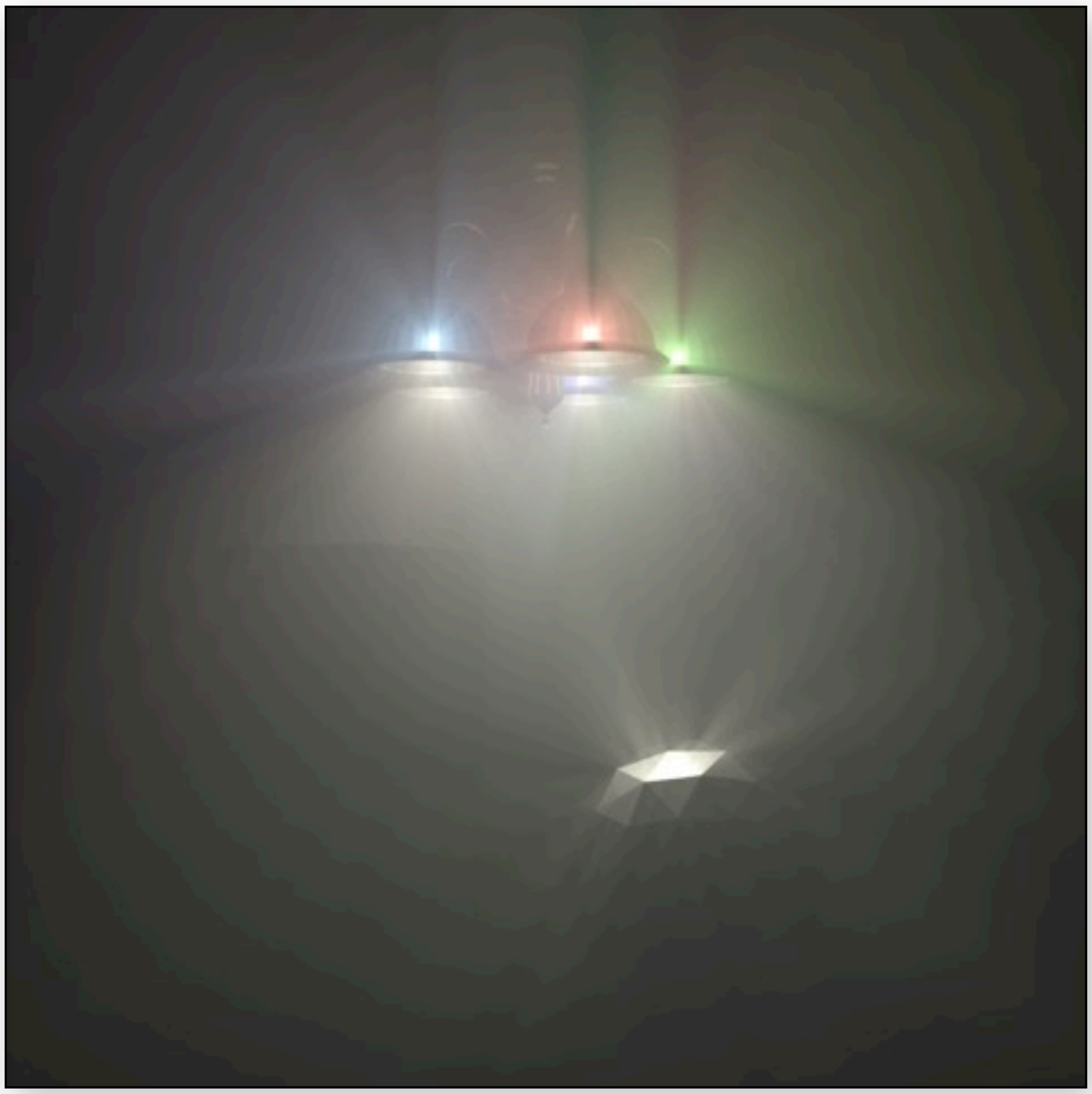
Thursday, 6 September 12

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



Pass 32

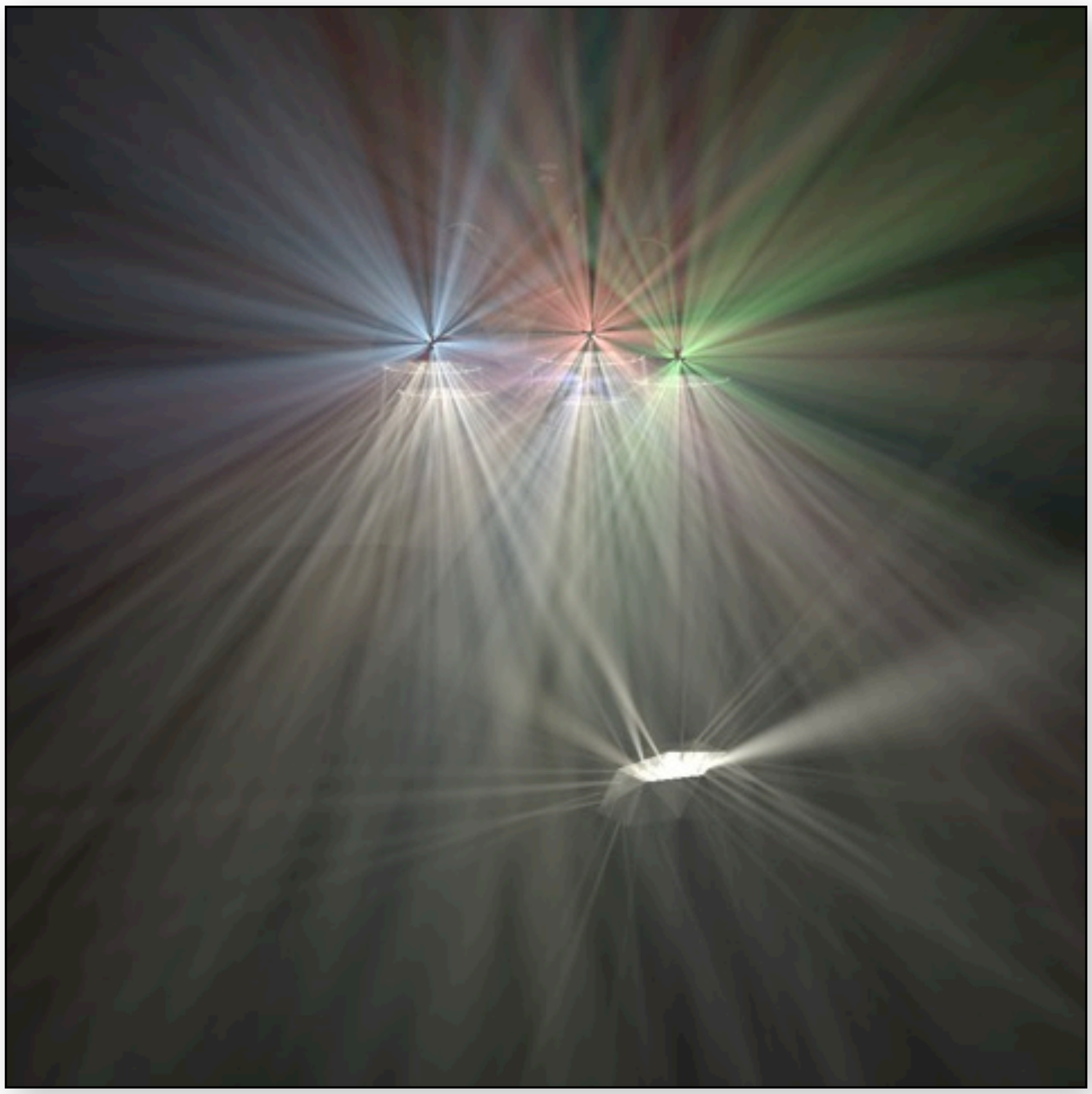


Average of Passes 1..32

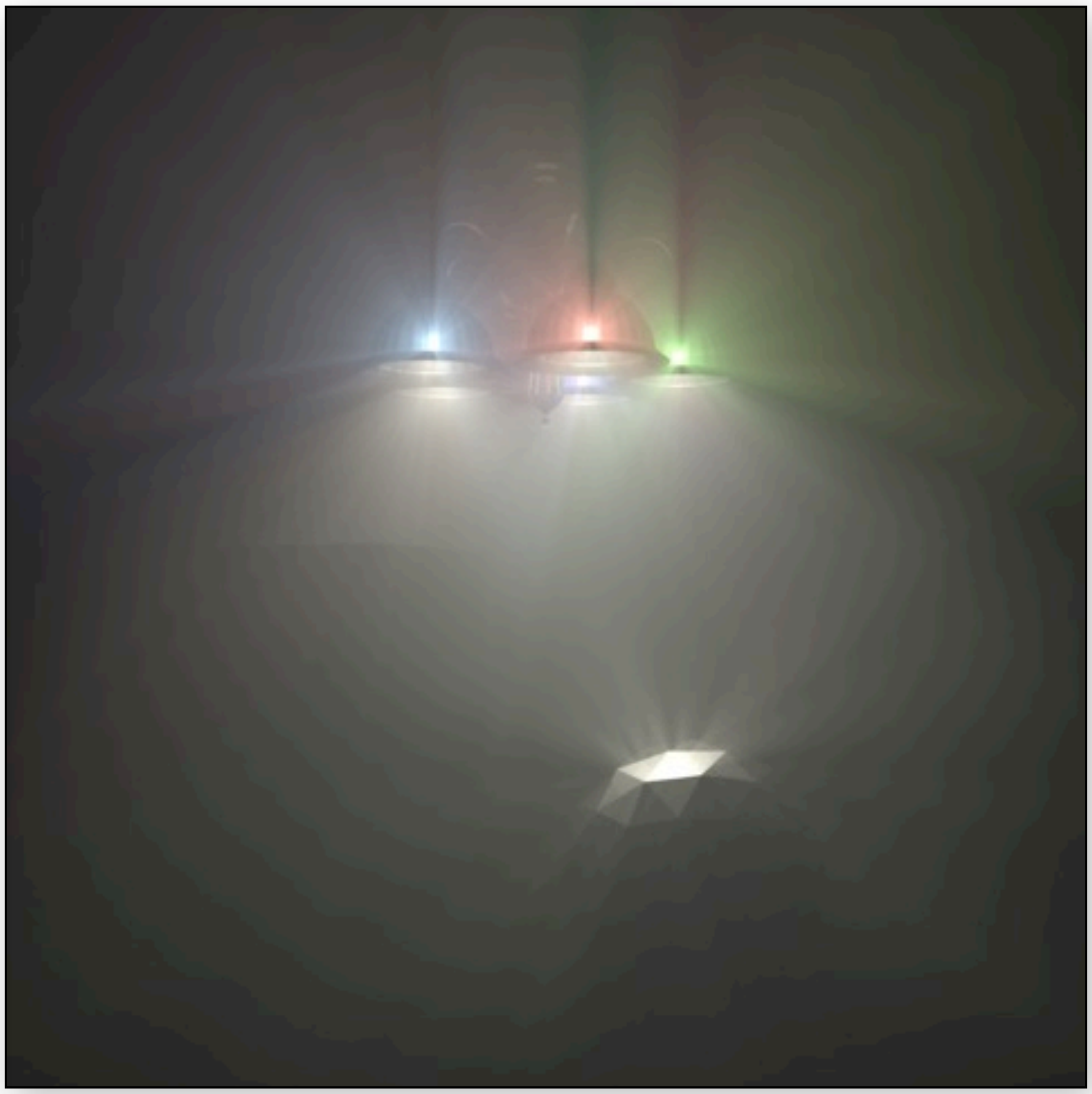
Thursday, 6 September 12

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



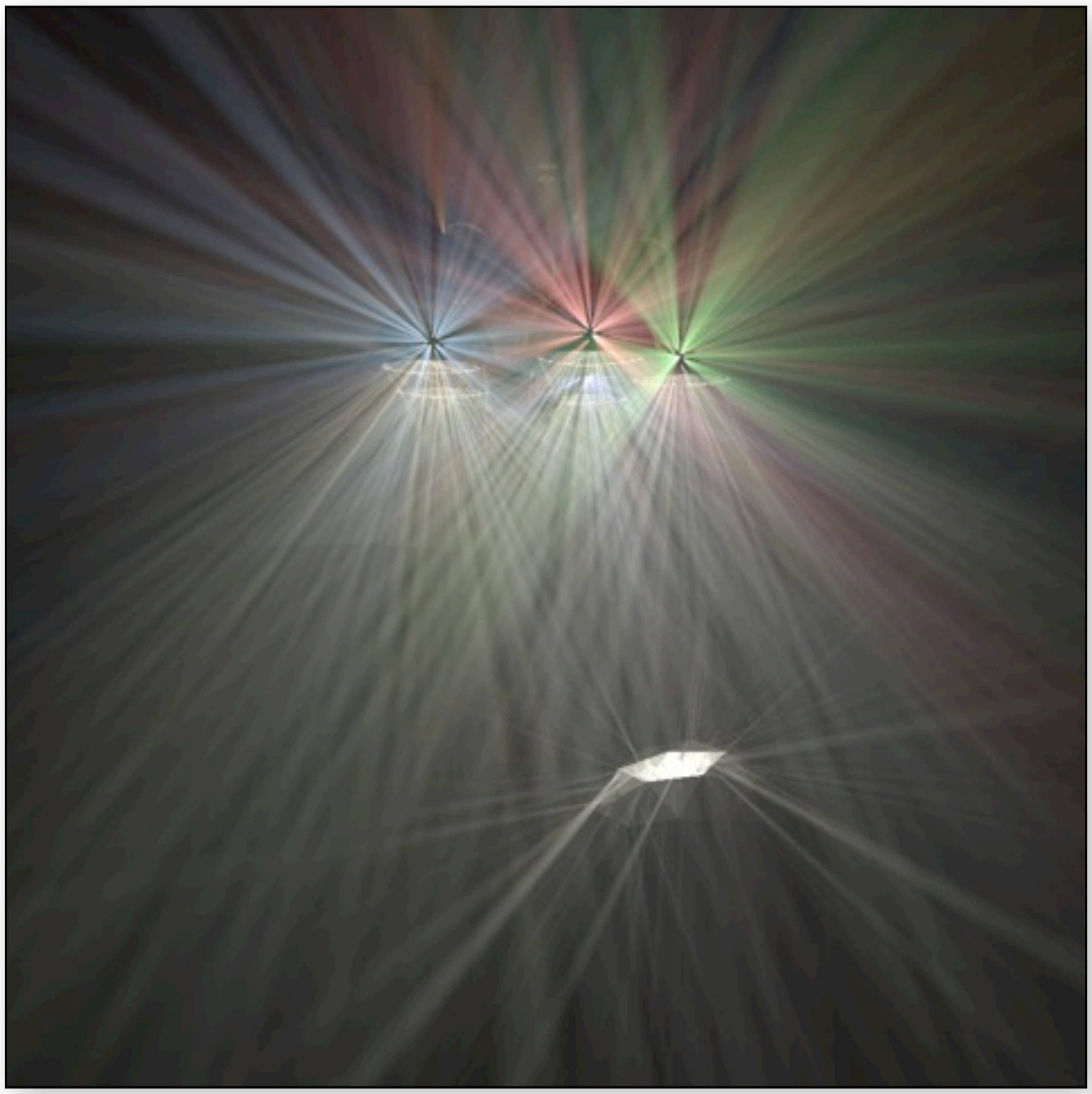
Pass 64



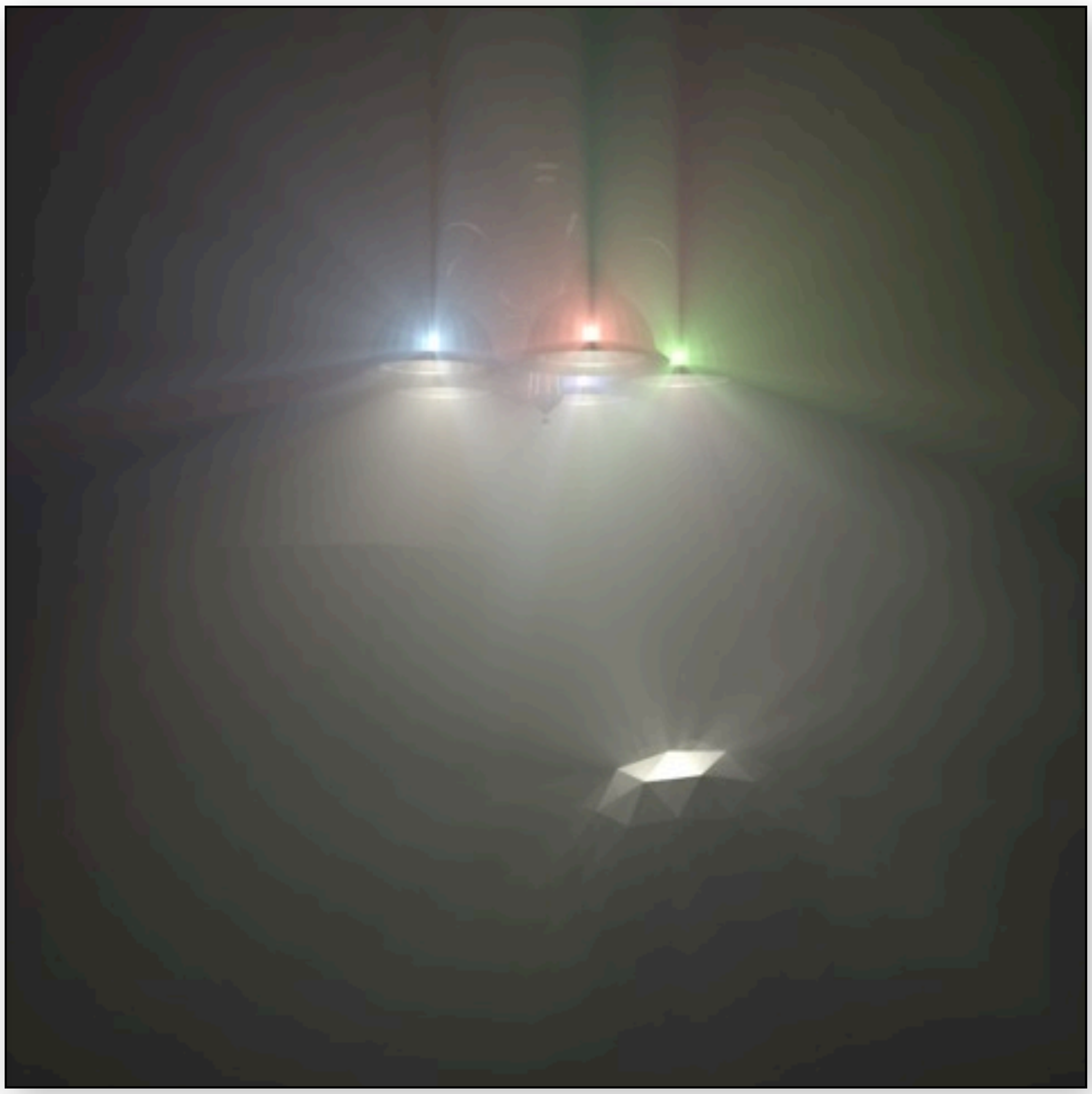
Average of Passes 1..64

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



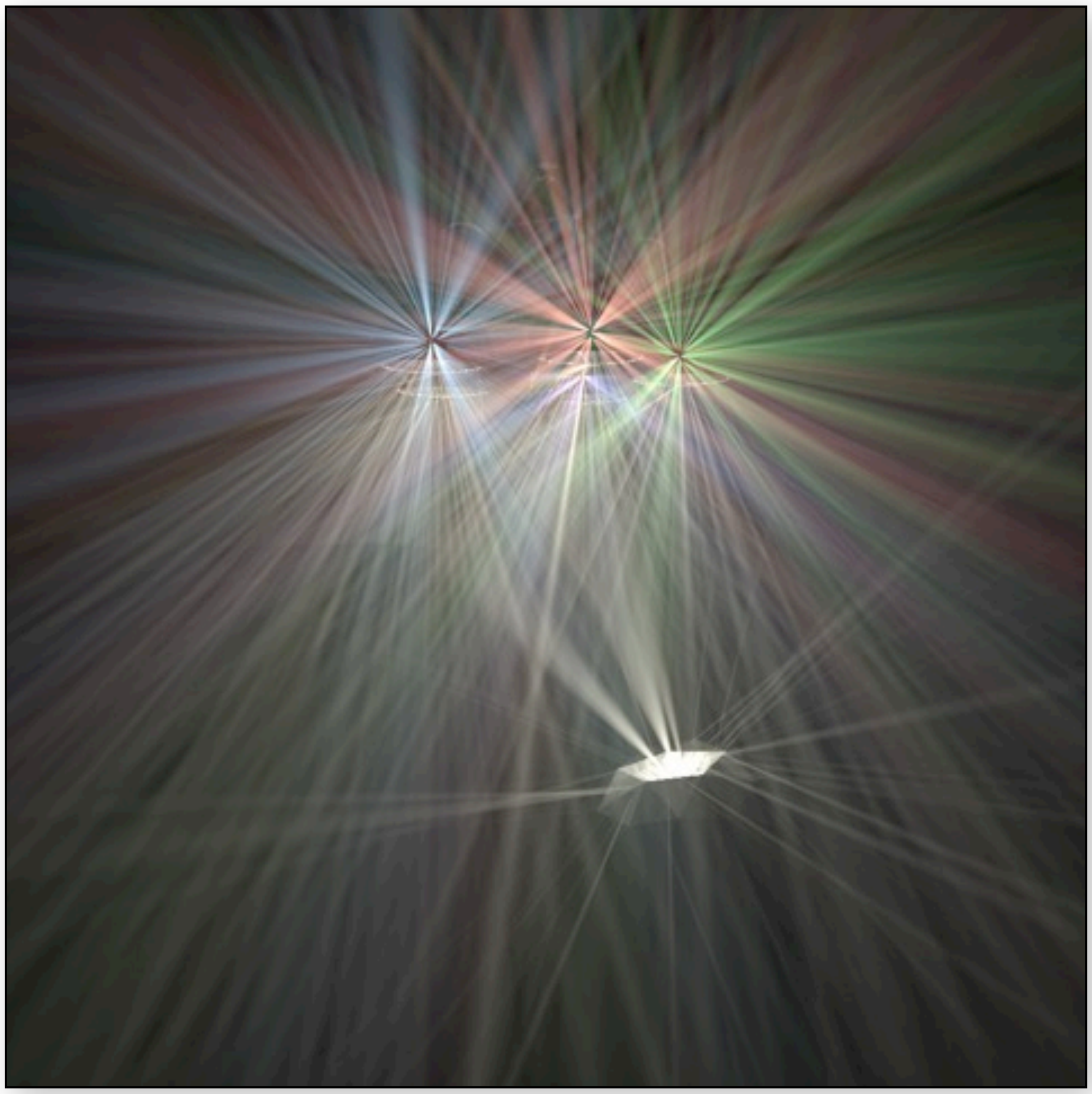
Pass 128



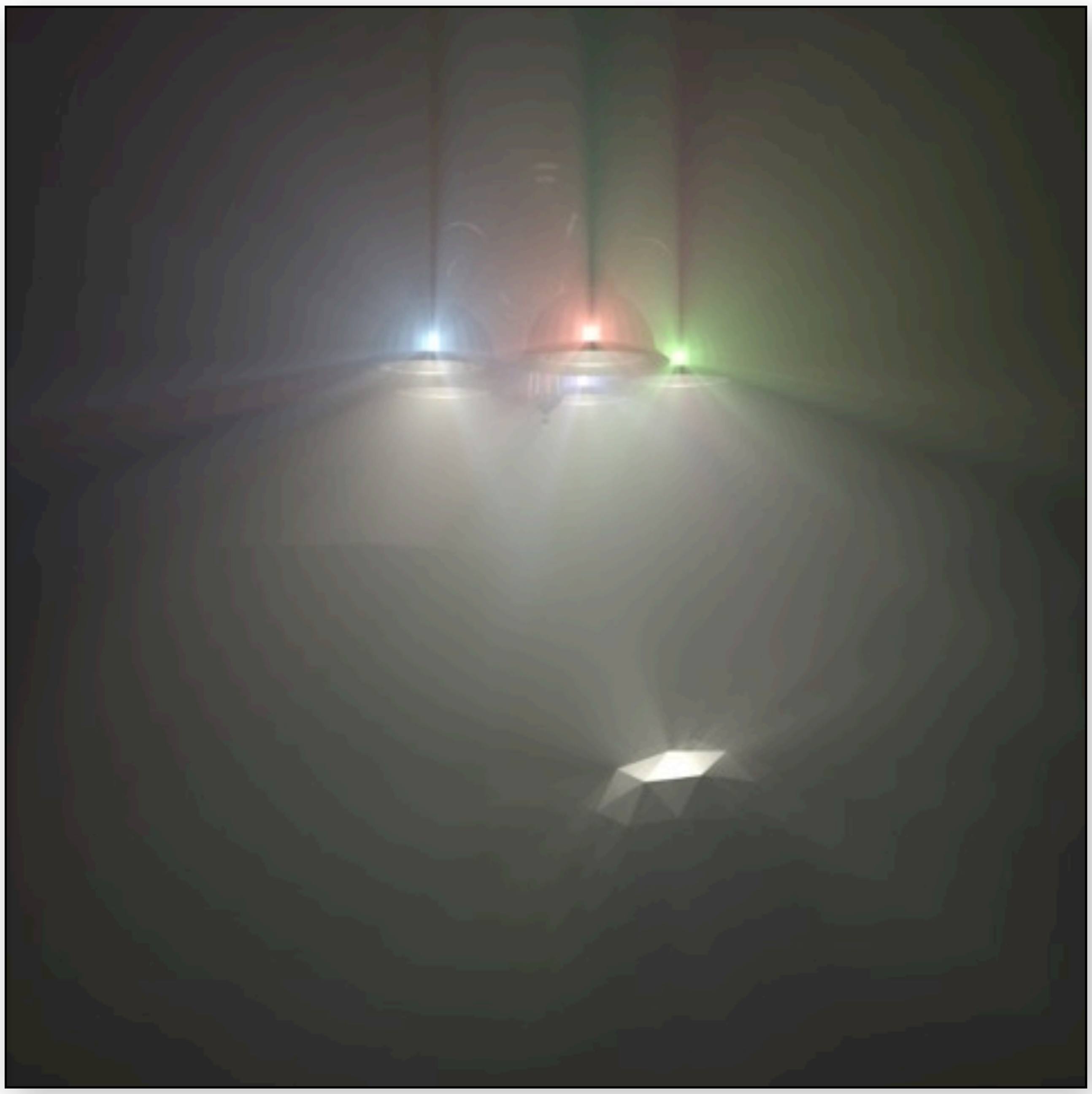
Average of Passes 1..128

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



Pass 256

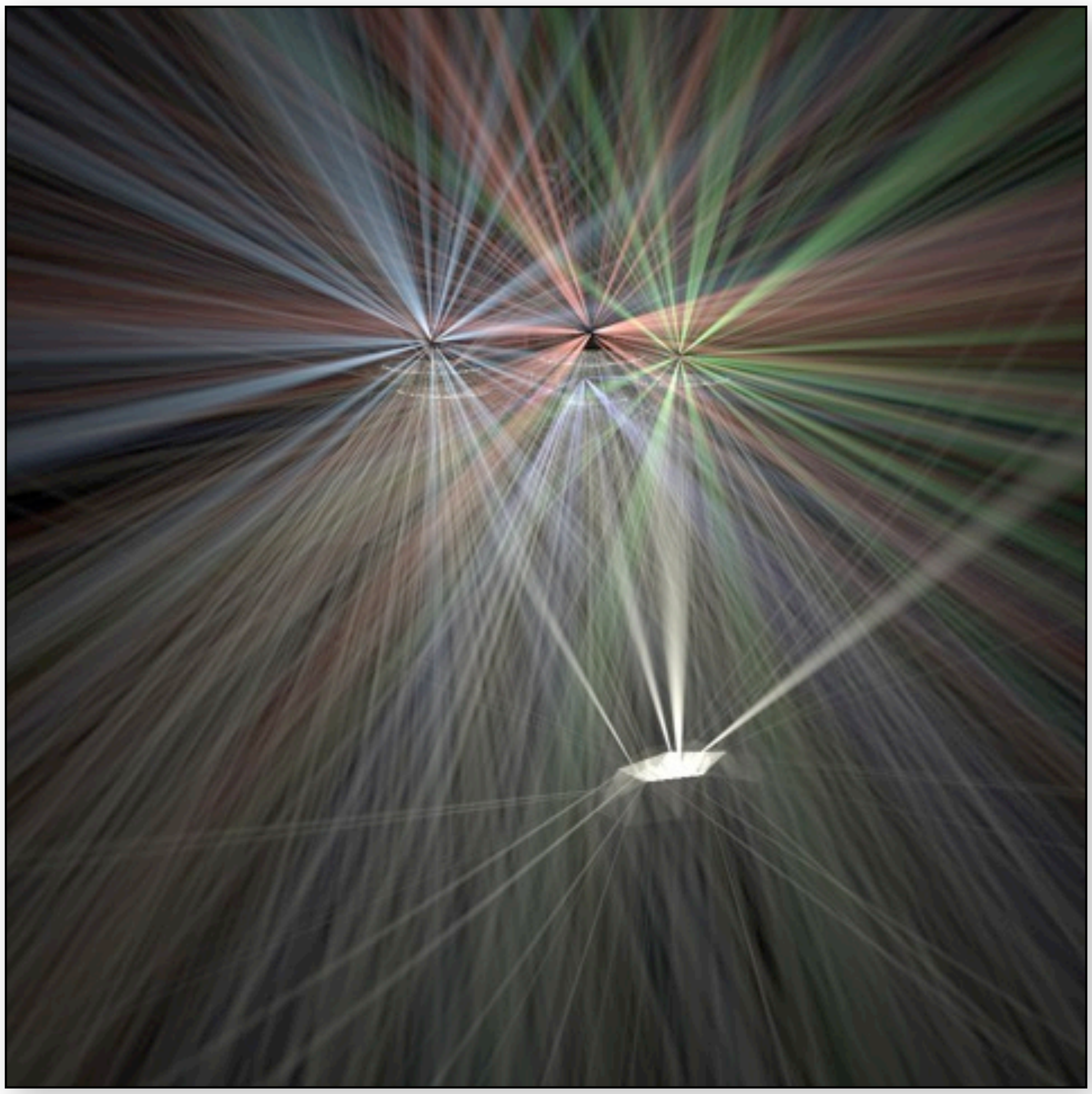


Average of Passes 1..256

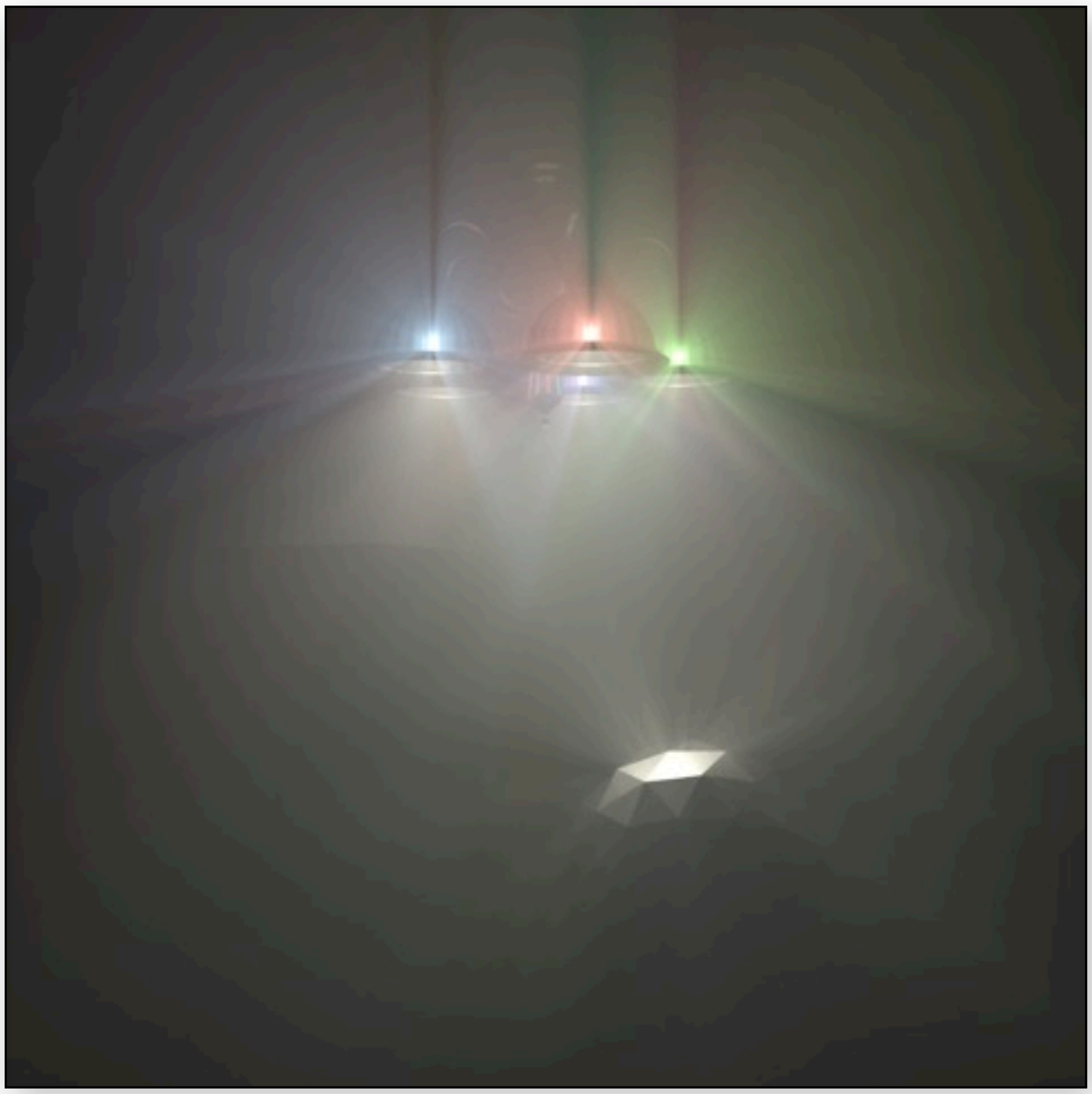
Thursday, 6 September 12

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



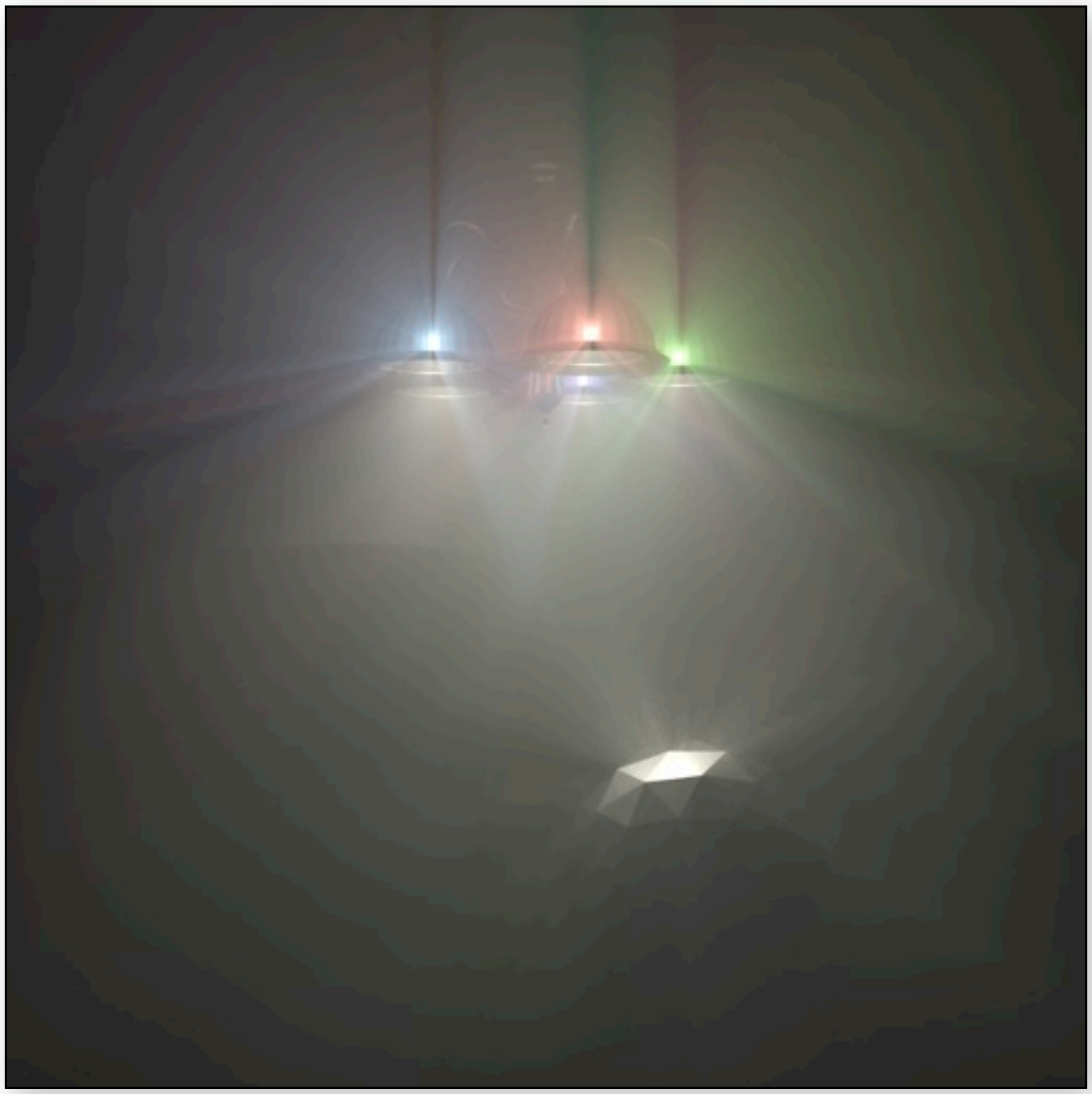
Pass 512



Average of Passes 1..512

- As a key step, we reduce the radii of the photon beams after each pass. Therefore each subsequent image on the left has less bias but slightly more noise. We reduce the radii in such a way, however, that as we add more passes the running average of these images converges to the correct solution.
- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams



100K beams per pass
51.2M beams total

Thursday, 6 September 12

- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams

100K beams per pass
51.2M beams total
+ progressive surface photon mapping

- The question is, how quickly should we reduce the radii?
- A main contribution of this paper is to determine exactly how quickly we need to reduce the beams radii so that we obtain convergence.

Progressive Photon Beams

- In the remainder of the talk I will describe how we arrive at our algorithm [click]
- One of the key steps is to derive the necessary conditions for statistical convergence [click]
- Also, we introduce an efficient and unbiased new way to handle heterogeneous media with beams [click]
- Finally, I'll discuss how our formulation allows for efficient implementation on either a CPU or GPU [click]
- We also propose some usability improvements applicable to all ppm techniques and I encourage you to look in the paper for these details.

Progressive Photon Beams

- Statistical convergence

- In the remainder of the talk I will describe how we arrive at our algorithm [click]
- One of the key steps is to derive the necessary conditions for statistical convergence [click]
- Also, we introduce an efficient and unbiased new way to handle heterogeneous media with beams [click]
- Finally, I'll discuss how our formulation allows for efficient implementation on either a CPU or GPU [click]
- We also propose some usability improvements applicable to all ppm techniques and I encourage you to look in the paper for these details.

Progressive Photon Beams

- Statistical convergence
- Heterogeneous media

- In the remainder of the talk I will describe how we arrive at our algorithm [click]
- One of the key steps is to derive the necessary conditions for statistical convergence [click]
- Also, we introduce an efficient and unbiased new way to handle heterogeneous media with beams [click]
- Finally, I'll discuss how our formulation allows for efficient implementation on either a CPU or GPU [click]
- We also propose some usability improvements applicable to all ppm techniques and I encourage you to look in the paper for these details.

Progressive Photon Beams

- Statistical convergence
- Heterogeneous media
- Efficient implementation

- In the remainder of the talk I will describe how we arrive at our algorithm [click]
- One of the key steps is to derive the necessary conditions for statistical convergence [click]
- Also, we introduce an efficient and unbiased new way to handle heterogeneous media with beams [click]
- Finally, I'll discuss how our formulation allows for efficient implementation on either a CPU or GPU [click]
- We also propose some usability improvements applicable to all ppm techniques and I encourage you to look in the paper for these details.

Progressive Photon Beams

- Statistical convergence
- Heterogeneous media
- Efficient implementation
- Usability improvements

- In the remainder of the talk I will describe how we arrive at our algorithm [click]
- One of the key steps is to derive the necessary conditions for statistical convergence [click]
- Also, we introduce an efficient and unbiased new way to handle heterogeneous media with beams [click]
- Finally, I'll discuss how our formulation allows for efficient implementation on either a CPU or GPU [click]
- We also propose some usability improvements applicable to all ppm techniques and I encourage you to look in the paper for these details.

Statistical Convergence

- Previous derivations not directly applicable

- Even though photon beams are a generalization of photon mapping, we cannot directly re-use the convergence analyses from progressive photon mapping directly for photon beams [click]
- The reason is that density estimation using beams is mathematically quite different than density estimation using points
- We need to analyze the necessary conditions for convergence in this more complicated case

Statistical Convergence

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

- Even though photon beams are a generalization of photon mapping, we cannot directly re-use the convergence analyses from progressive photon mapping directly for photon beams [click]
- The reason is that density estimation using beams is mathematically quite different than density estimation using points
- We need to analyze the necessary conditions for convergence in this more complicated case

Statistical Framework

- We build on Knaus and Zwicker approach

- To accomplish this, we build off of the probabilistic framework developed by Knaus and Zwicker
- They showed that convergence in such a progressive algorithm can be achieved [click] by enforcing a ratio of the variance between passes.
- Alpha is a user parameter between 0 and 1 which influences the bias/variance (as we will see later)
- Our task is to enforce such a variance ratio when the images are generated using photon beams
- Ultimately we are interested in determining how the radii of beams should shrink, so we need some way to relate the variance of each pass to the radius of the beams

Statistical Framework

- We build on Knaus and Zwicker approach
- Need to enforce variance ratio:

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$

- To accomplish this, we build off of the probabilistic framework developed by Knaus and Zwicker
- They showed that convergence in such a progressive algorithm can be achieved [click] by enforcing a ratio of the variance between passes.
- Alpha is a user parameter between 0 and 1 which influences the bias/variance (as we will see later)
- Our task is to enforce such a variance ratio when the images are generated using photon beams
- Ultimately we are interested in determining how the radii of beams should shrink, so we need some way to relate the variance of each pass to the radius of the beams

Rendering Using One Beam

- First, consider rendering each pass using a single beam

- Lets first consider render each pass using a single stochastically-generated beam, we will generalize this later

Rendering Using One Beam

- We show that:

- In the paper, we show that under some reasonable assumptions: [click]
- The variance is inversely proportional to the radius of the beam. [click]
- More concretely, the variance increases linearly as we reduce the kernel radius r [click]
- On the other hand, we show that the opposite is true for bias: it is linearly proportional to the radius.

Rendering Using One Beam

- We show that:
  - $\mathrm{Variance}[\mathrm{Pass}_i] \propto \dfrac{1}{r_i}$

- In the paper, we show that under some reasonable assumptions: [click]
- The variance is inversely proportional to the radius of the beam. [click]
- More concretely, the variance increases linearly as we reduce the kernel radius r [click]
- On the other hand, we show that the opposite is true for bias: it is linearly proportional to the radius.

Rendering Using One Beam

- We show that:
  - $\text{Variance}[\text{Pass}_i] \propto \dfrac{1}{r_i}$
    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.

- In the paper, we show that under some reasonable assumptions: [click]
- The variance is inversely proportional to the radius of the beam. [click]
- More concretely, the variance increases linearly as we reduce the kernel radius r [click]
- On the other hand, we show that the opposite is true for bias: it is linearly proportional to the radius.

Rendering Using One Beam

- We show that:
  - $\text{Variance}[\text{Pass}_i] \propto \dfrac{1}{r_i}$
    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.
  - $\text{Bias}[\text{Pass}_i] \propto r_i$
    - The **bias *decreases* linearly** as we ***reduce*** the kernel radius r.

- In the paper, we show that under some reasonable assumptions: [click]
- The variance is inversely proportional to the radius of the beam. [click]
- More concretely, the variance increases linearly as we reduce the kernel radius r [click]
- On the other hand, we show that the opposite is true for bias: it is linearly proportional to the radius.

Rendering Using Many Beams

- We show that:
  - $\text{Variance}[\text{Pass}_i] \propto \dfrac{1}{r_i}$
    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.
  - $\text{Bias}[\text{Pass}_i] \propto r_i$
    - The **bias *decreases* linearly** as we ***reduce*** the kernel radius r.

Thursday, 6 September 12

- In practice, each rendered pass will use many beams, each with their own radius.
- In the original photon beams method, the radii are computed using photon differentials to adapt the blurring across the scene
- In our paper we show that we can easily generalize the one-beam case to allow for multiple beams.
- The only change that is necessary is that, for variance, the beam radius is replaced by the harmonic mean of the beam radii
- For bias, the radius gets replaced by the arithmetic mean of the beam radii
- The high-level story remains the same, that variance is inversely proportional to the beam radii, and bias is directly proportional to the radii

Rendering Using **Many Beams**

- We show that:
  - $\mathrm{Variance}[\mathrm{Pass}_i] \propto \frac{1}{\boxed{r_i}}$
    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.
  - $\mathrm{Bias}[\mathrm{Pass}_i] \propto r_i$
    - The **bias *decreases* linearly** as we ***reduce*** the kernel radius r.

- In practice, each rendered pass will use many beams, each with their own radius.
- In the original photon beams method, the radii are computed using photon differentials to adapt the blurring across the scene
- In our paper we show that we can easily generalize the one-beam case to allow for multiple beams.
- The only change that is necessary is that, for variance, the beam radius is replaced by the harmonic mean of the beam radii
- For bias, the radius gets replaced by the arithmetic mean of the beam radii
- The high-level story remains the same, that variance is inversely proportional to the beam radii, and bias is directly proportional to the radii

Rendering Using Many Beams

- We show that:
  - $\mathrm{Variance}[\mathrm{Pass}_i] \propto \frac{1}{\boxed{r_i}}$

    harmonic mean of beam radii
    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.
  - $\mathrm{Bias}[\mathrm{Pass}_i] \propto r_i$
    - The **bias *decreases* linearly** as we ***reduce*** the kernel radius r.

Thursday, 6 September 12

- In practice, each rendered pass will use many beams, each with their own radius.
- In the original photon beams method, the radii are computed using photon differentials to adapt the blurring across the scene
- In our paper we show that we can easily generalize the one-beam case to allow for multiple beams.
- The only change that is necessary is that, for variance, the beam radius is replaced by the harmonic mean of the beam radii
- For bias, the radius gets replaced by the arithmetic mean of the beam radii
- The high-level story remains the same, that variance is inversely proportional to the beam radii, and bias is directly proportional to the radii

Rendering Using Many Beams

- We show that:
  - $\text{Variance}[\text{Pass}_i] \propto \frac{1}{\boxed{r_i}}$

    harmonic mean of beam radii

    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.
  - $\text{Bias}[\text{Pass}_i] \propto \boxed{r_i}$
    - The **bias *decreases* linearly** as we ***reduce*** the kernel radius r.

- In practice, each rendered pass will use many beams, each with their own radius.
- In the original photon beams method, the radii are computed using photon differentials to adapt the blurring across the scene
- In our paper we show that we can easily generalize the one-beam case to allow for multiple beams.
- The only change that is necessary is that, for variance, the beam radius is replaced by the harmonic mean of the beam radii
- For bias, the radius gets replaced by the arithmetic mean of the beam radii
- The high-level story remains the same, that variance is inversely proportional to the beam radii, and bias is directly proportional to the radii

Rendering Using Many Beams

- We show that:
  - $\text{Variance}[\text{Pass}_i] \propto \frac{1}{\boxed{r_i}}$

    harmonic mean of beam radii

    - The **variance *increases* linearly** as we ***reduce*** the kernel radius r.
  - $\text{Bias}[\text{Pass}_i] \propto \boxed{r_i}$

    arithmetic mean of beam radii

    - The **bias *decreases* linearly** as we ***reduce*** the kernel radius r.

- In practice, each rendered pass will use many beams, each with their own radius.
- In the original photon beams method, the radii are computed using photon differentials to adapt the blurring across the scene
- In our paper we show that we can easily generalize the one-beam case to allow for multiple beams.
- The only change that is necessary is that, for variance, the beam radius is replaced by the harmonic mean of the beam radii
- For bias, the radius gets replaced by the arithmetic mean of the beam radii
- The high-level story remains the same, that variance is inversely proportional to the beam radii, and bias is directly proportional to the radii

Radius Reduction Sequence

- Global radius reduction factor R_i which scales all beam radii

- Our ultimate goal is to derive a global radius reduction factor, which I'll denote with capital R
- R will start out at 1 in the first pass, and in each subsequent pass, the photon beams radii will be scaled by this factor [click]
- We can combine our desired variance ratio [click], with our relation between variance and radius to obtain an expression [click] which dictates how the global scaling factor should change across passes in order to obtain a convergent algorithm.
- This radius reduction factor is what allows our algorithm to, in theory, produce convergent results across multiple passes.

Radius Reduction Sequence

- Global radius reduction factor R_i which scales all beam radii

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$

- Our ultimate goal is to derive a global radius reduction factor, which I'll denote with capital R
- R will start out at 1 in the first pass, and in each subsequent pass, the photon beams radii will be scaled by this factor [click]
- We can combine our desired variance ratio [click], with our relation between variance and radius to obtain an expression [click] which dictates how the global scaling factor should change across passes in order to obtain a convergent algorithm.
- This radius reduction factor is what allows our algorithm to, in theory, produce convergent results across multiple passes.

Radius Reduction Sequence

- Global radius reduction factor R_i which scales all beam radii

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha} \quad \& \quad \text{Variance}[\text{Pass}_i] \propto \frac{1}{r_H}$$

Thursday, 6 September 12

- Our ultimate goal is to derive a global radius reduction factor, which I'll denote with capital R
- R will start out at 1 in the first pass, and in each subsequent pass, the photon beams radii will be scaled by this factor [click]
- We can combine our desired variance ratio [click], with our relation between variance and radius to obtain an expression [click] which dictates how the global scaling factor should change across passes in order to obtain a convergent algorithm.
- This radius reduction factor is what allows our algorithm to, in theory, produce convergent results across multiple passes.

Radius Reduction Sequence

- Global radius reduction factor R_i which scales all beam radii

$$\frac{\mathrm{Variance}[\mathrm{Pass}_{i+1}]}{\mathrm{Variance}[\mathrm{Pass}_i]} = \frac{i+1}{i+\alpha} \quad \& \quad \mathrm{Variance}[\mathrm{Pass}_i] \propto \frac{1}{r_H}$$

$$\Downarrow$$

$$\frac{R_{i+1}}{R_i} = \frac{i+\alpha}{i+1}$$

Thursday, 6 September 12

- Our ultimate goal is to derive a global radius reduction factor, which I'll denote with capital R
- R will start out at 1 in the first pass, and in each subsequent pass, the photon beams radii will be scaled by this factor [click]
- We can combine our desired variance ratio [click], with our relation between variance and radius to obtain an expression [click] which dictates how the global scaling factor should change across passes in order to obtain a convergent algorithm.
- This radius reduction factor is what allows our algorithm to, in theory, produce convergent results across multiple passes.

Radius Reduction Sequence

- Global radius reduction factor R_i which scales all beam radii

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha} \quad \& \quad \text{Variance}[\text{Pass}_i] \propto \frac{1}{r_H}$$

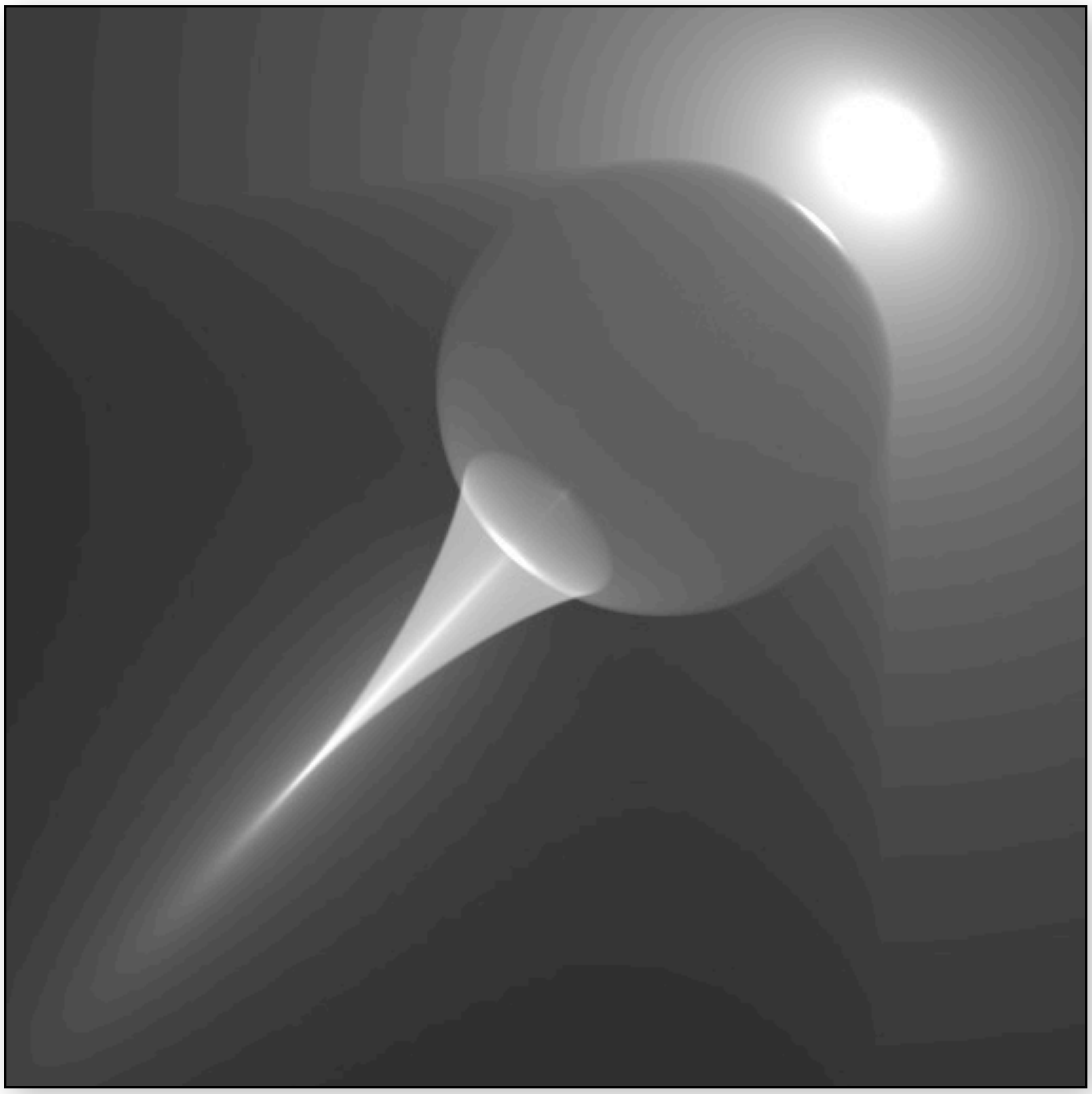
$$\Downarrow$$

$$\boxed{\frac{R_{i+1}}{R_i} = \frac{i+\alpha}{i+1}}$$

Thursday, 6 September 12

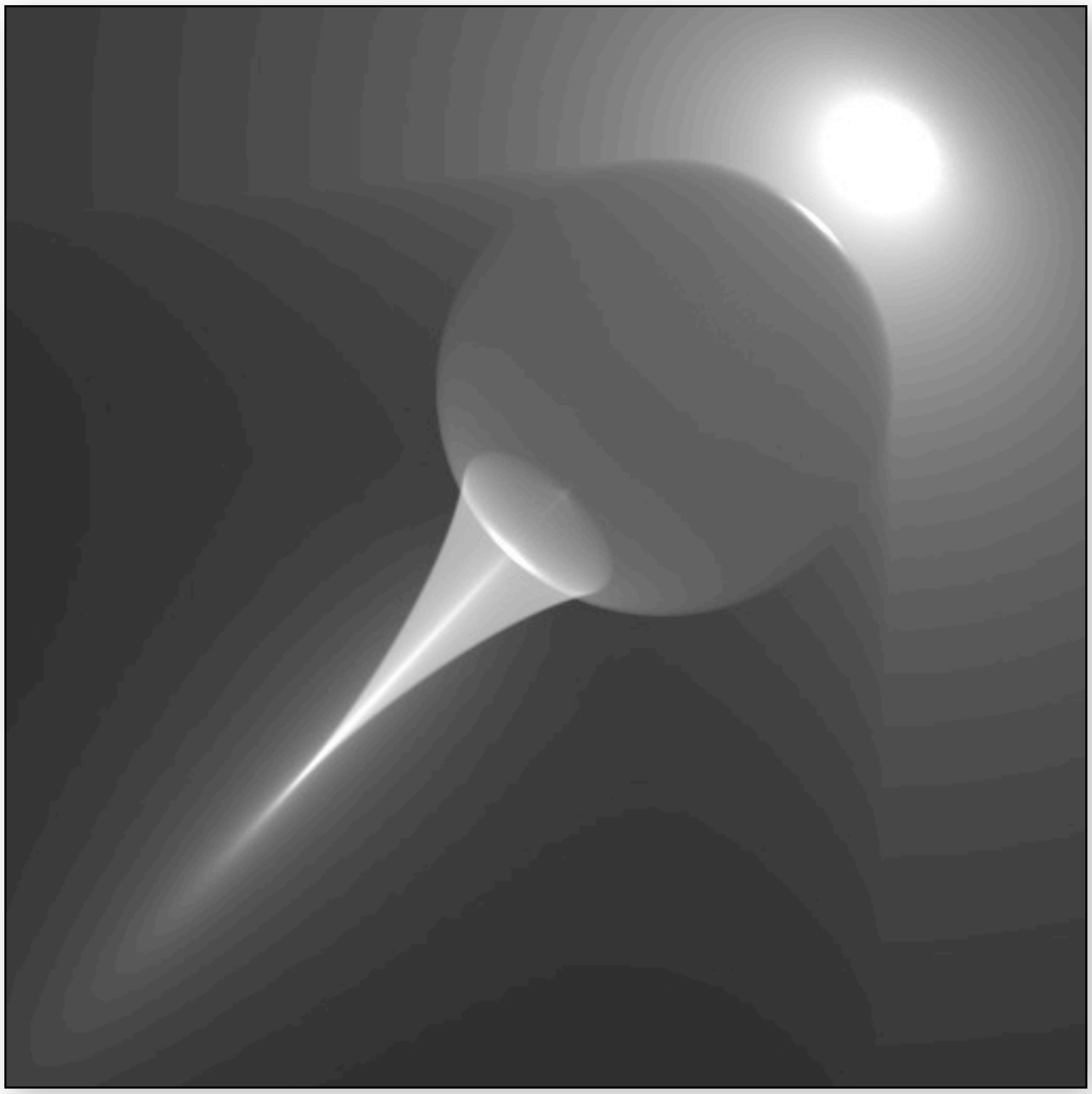
- Our ultimate goal is to derive a global radius reduction factor, which I'll denote with capital R
- R will start out at 1 in the first pass, and in each subsequent pass, the photon beams radii will be scaled by this factor [click]
- We can combine our desired variance ratio [click], with our relation between variance and radius to obtain an expression [click] which dictates how the global scaling factor should change across passes in order to obtain a convergent algorithm.
- This radius reduction factor is what allows our algorithm to, in theory, produce convergent results across multiple passes.

Empirical Validation



- To empirically validate this result, we ran the following experiment on this sphere caustic scene
- We first render a high-quality ground truth image
- Then, we run 1 thousand passes of our algorithm with three different settings for alpha
- We repeat this 10 thousand times
- Given these 30 million measurements, we can compute and plot the sample variance and bias at the highlighted point A for each pass

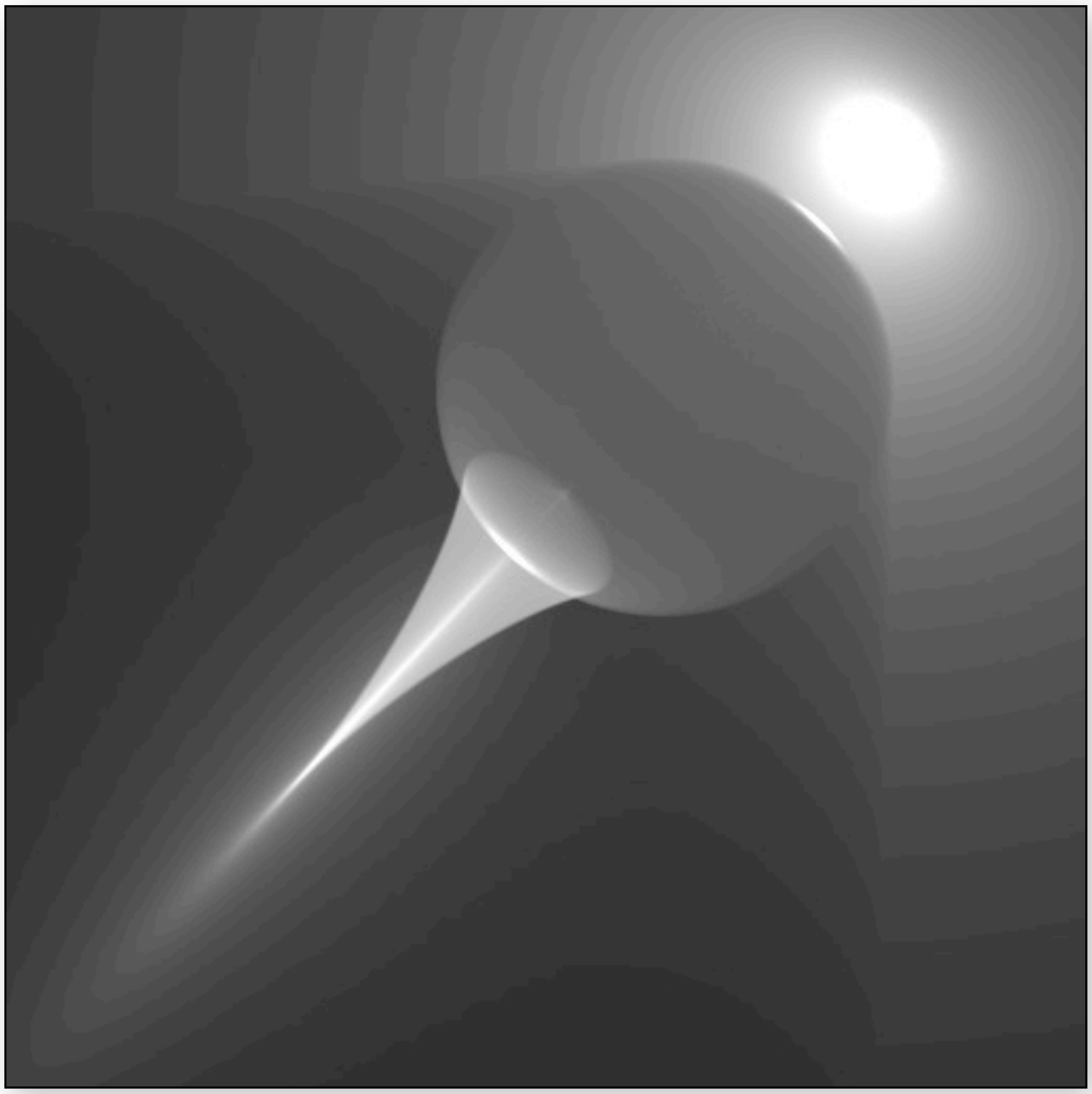
Empirical Validation



- Render reference

- To empirically validate this result, we ran the following experiment on this sphere caustic scene
- We first render a high-quality ground truth image
- Then, we run 1 thousand passes of our algorithm with three different settings for alpha
- We repeat this 10 thousand times
- Given these 30 million measurements, we can compute and plot the sample variance and bias at the highlighted point A for each pass

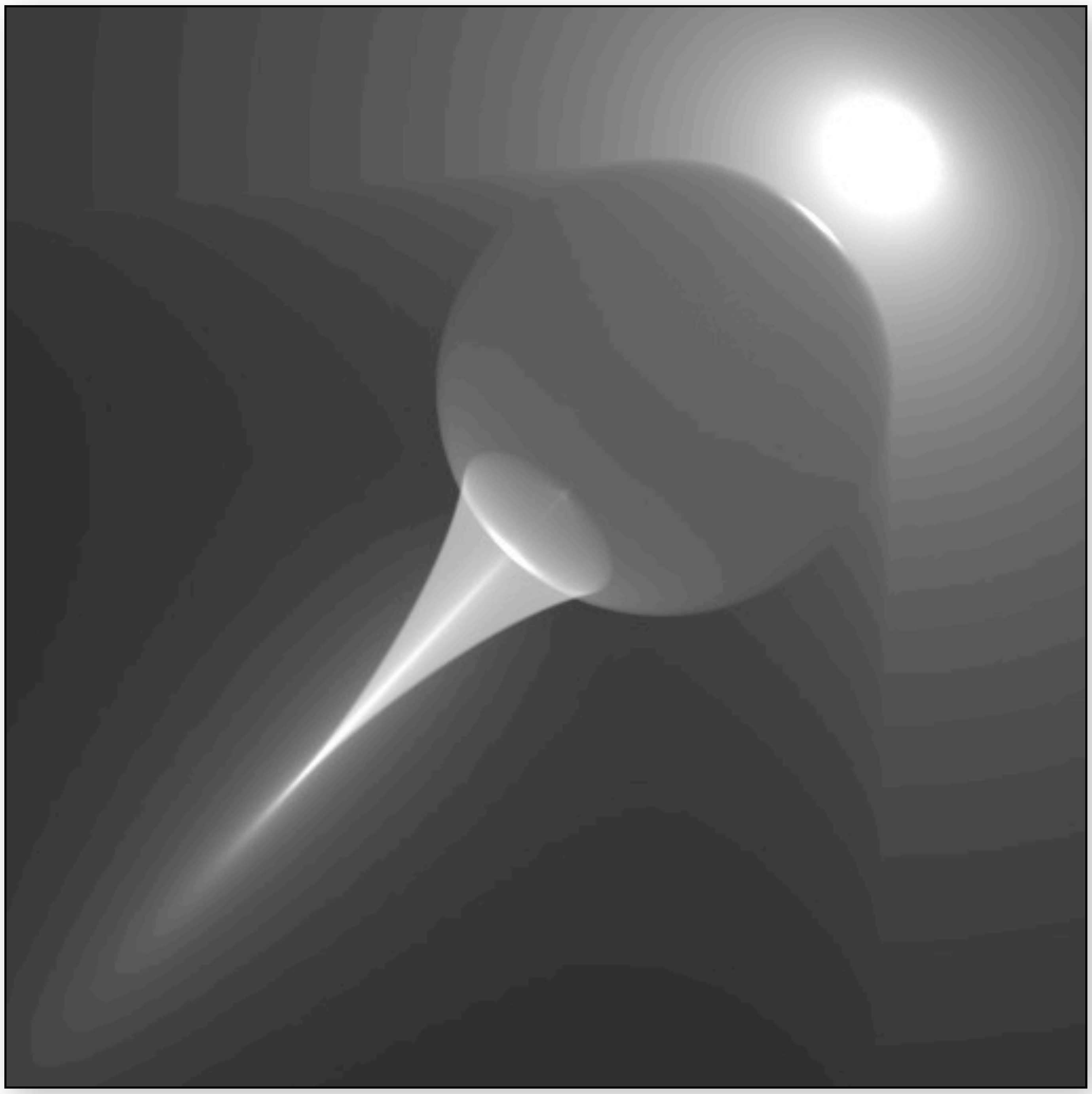
Empirical Validation



- Render reference
- Run 1024 passes of PPB with various α

- To empirically validate this result, we ran the following experiment on this sphere caustic scene
- We first render a high-quality ground truth image
- Then, we run 1 thousand passes of our algorithm with three different settings for alpha
- We repeat this 10 thousand times
- Given these 30 million measurements, we can compute and plot the sample variance and bias at the highlighted point A for each pass

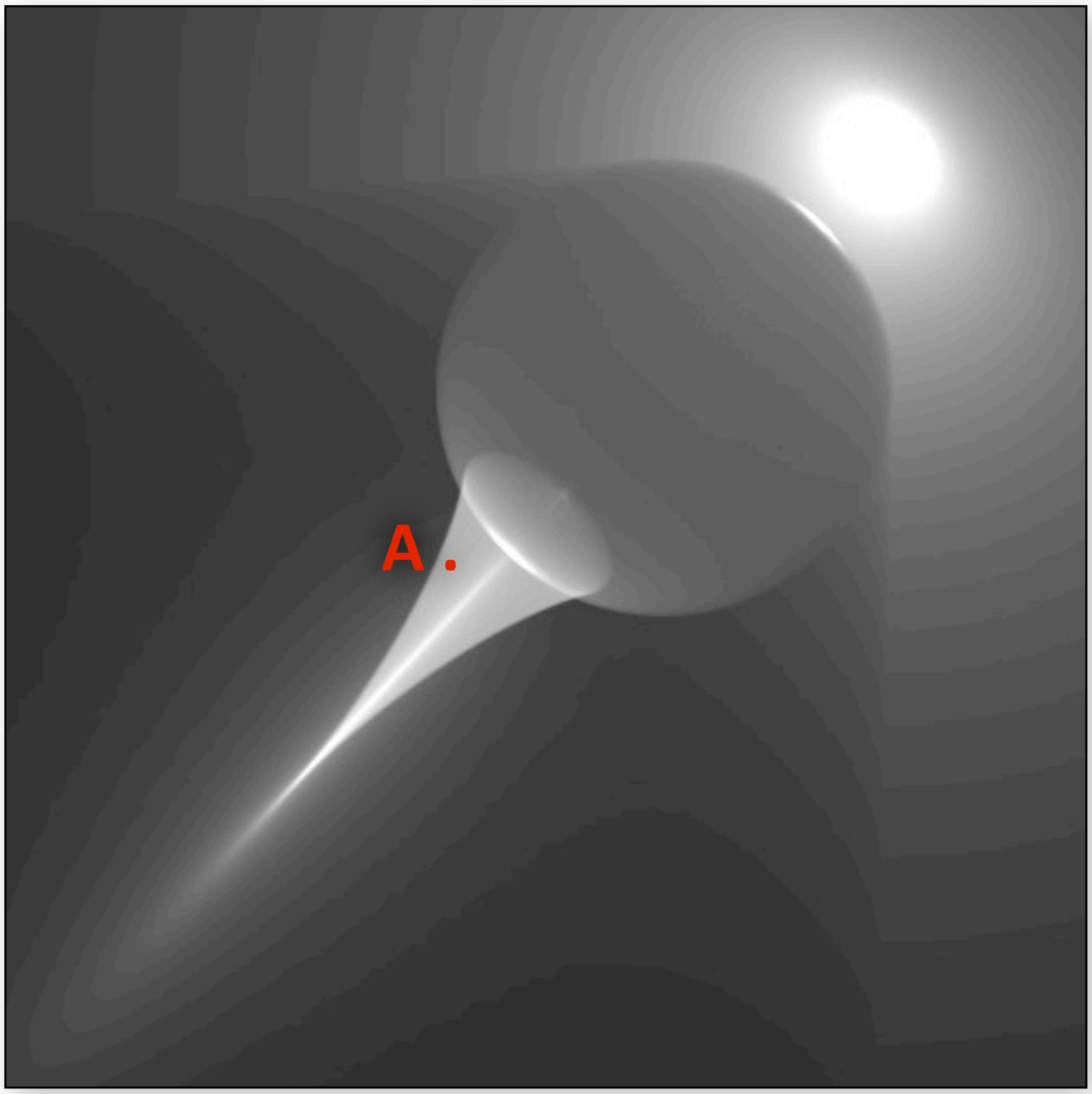
Empirical Validation



- Render reference
- Run 1024 passes of PPB with various α
  - Repeat 10K times

- To empirically validate this result, we ran the following experiment on this sphere caustic scene
- We first render a high-quality ground truth image
- Then, we run 1 thousand passes of our algorithm with three different settings for alpha
- We repeat this 10 thousand times
- Given these 30 million measurements, we can compute and plot the sample variance and bias at the highlighted point A for each pass

Empirical Validation



- Render reference
- Run 1024 passes of PPB with various α
  - Repeat 10K times
- Measure variance and bias at point **A**

Thursday, 6 September 12

- To empirically validate this result, we ran the following experiment on this sphere caustic scene
- We first render a high-quality ground truth image
- Then, we run 1 thousand passes of our algorithm with three different settings for alpha
- We repeat this 10 thousand times
- Given these 30 million measurements, we can compute and plot the sample variance and bias at the highlighted point A for each pass

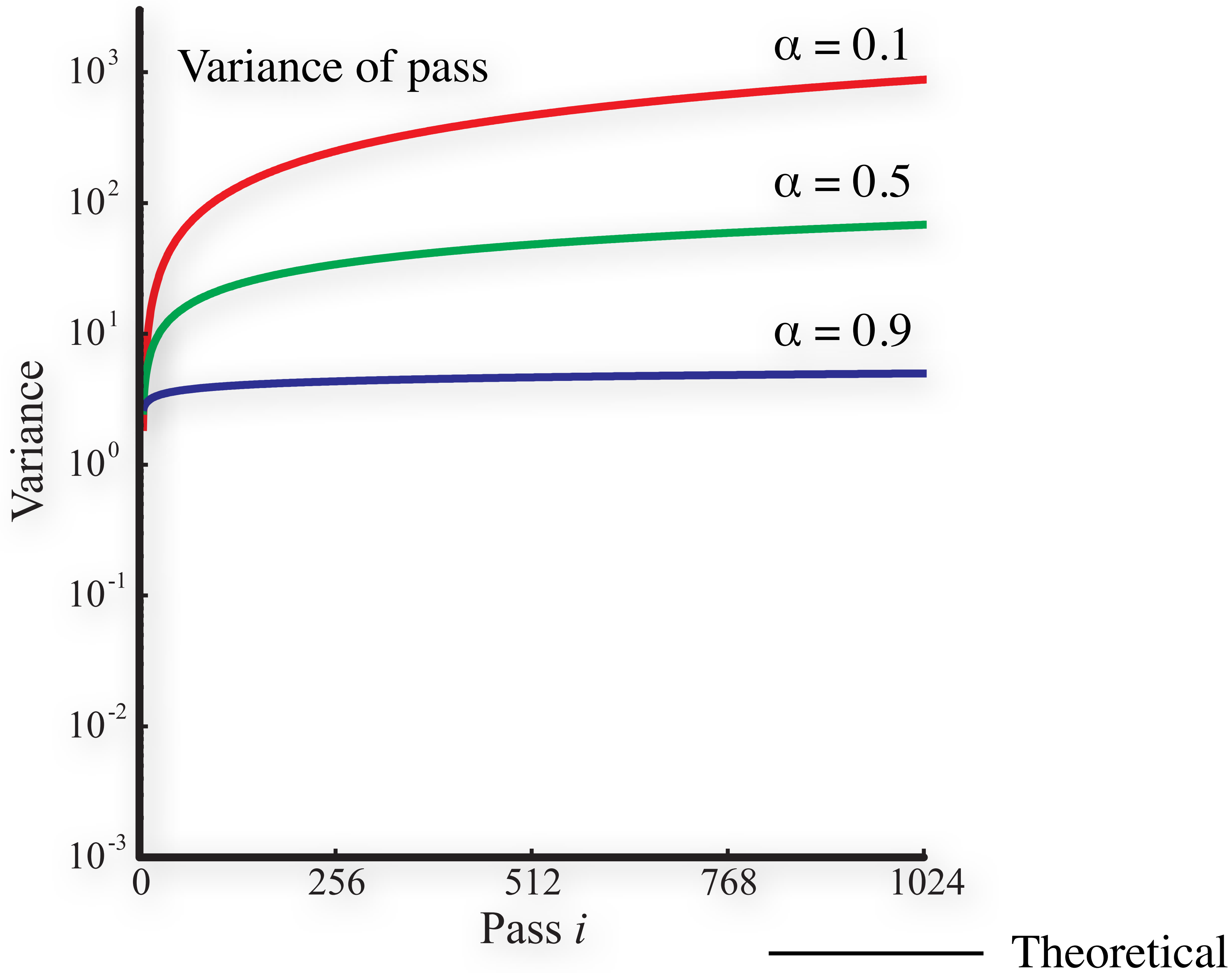
Empirical Validation

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$

- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Empirical Validation

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$

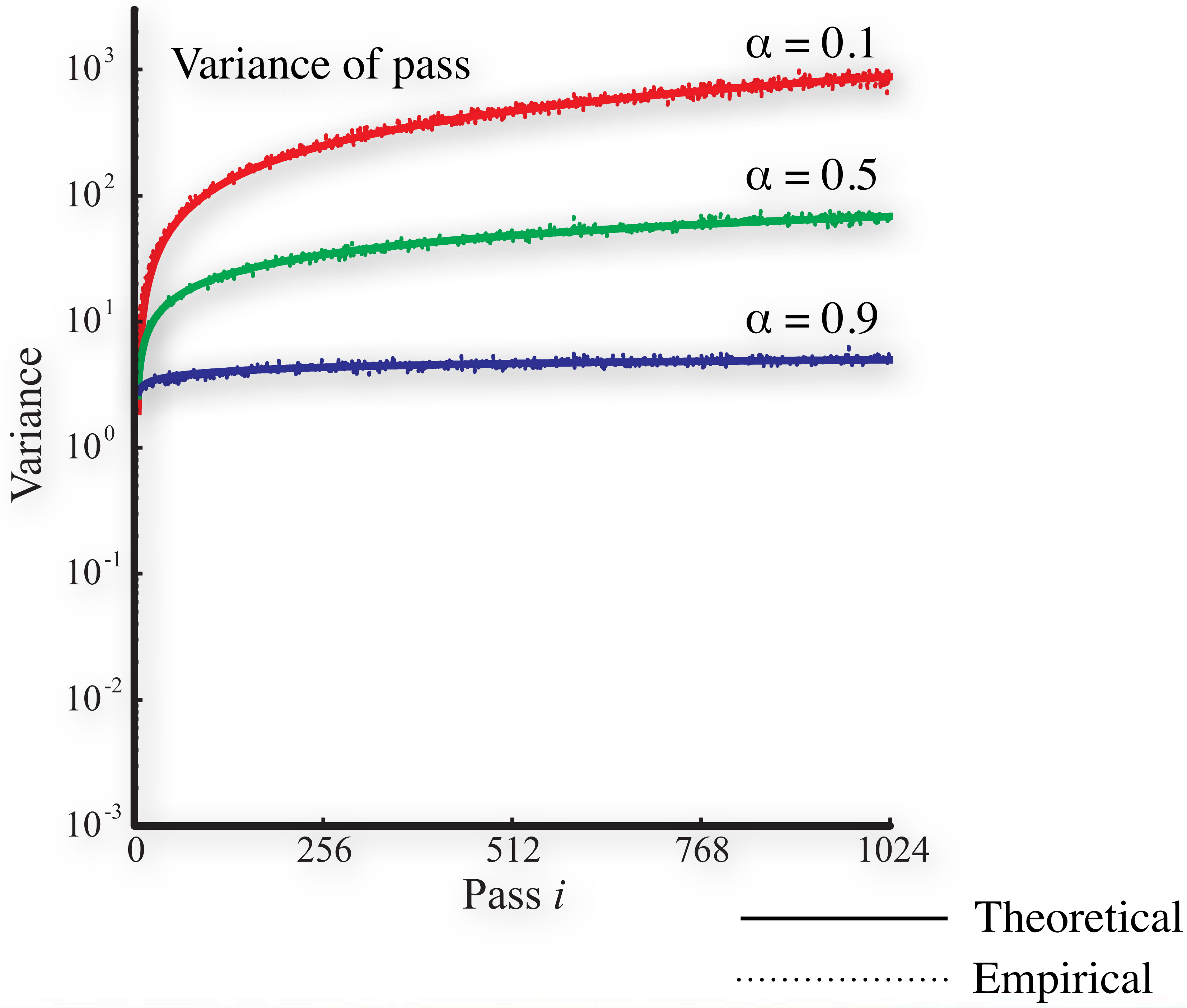


Thursday, 6 September 12

- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Empirical Validation

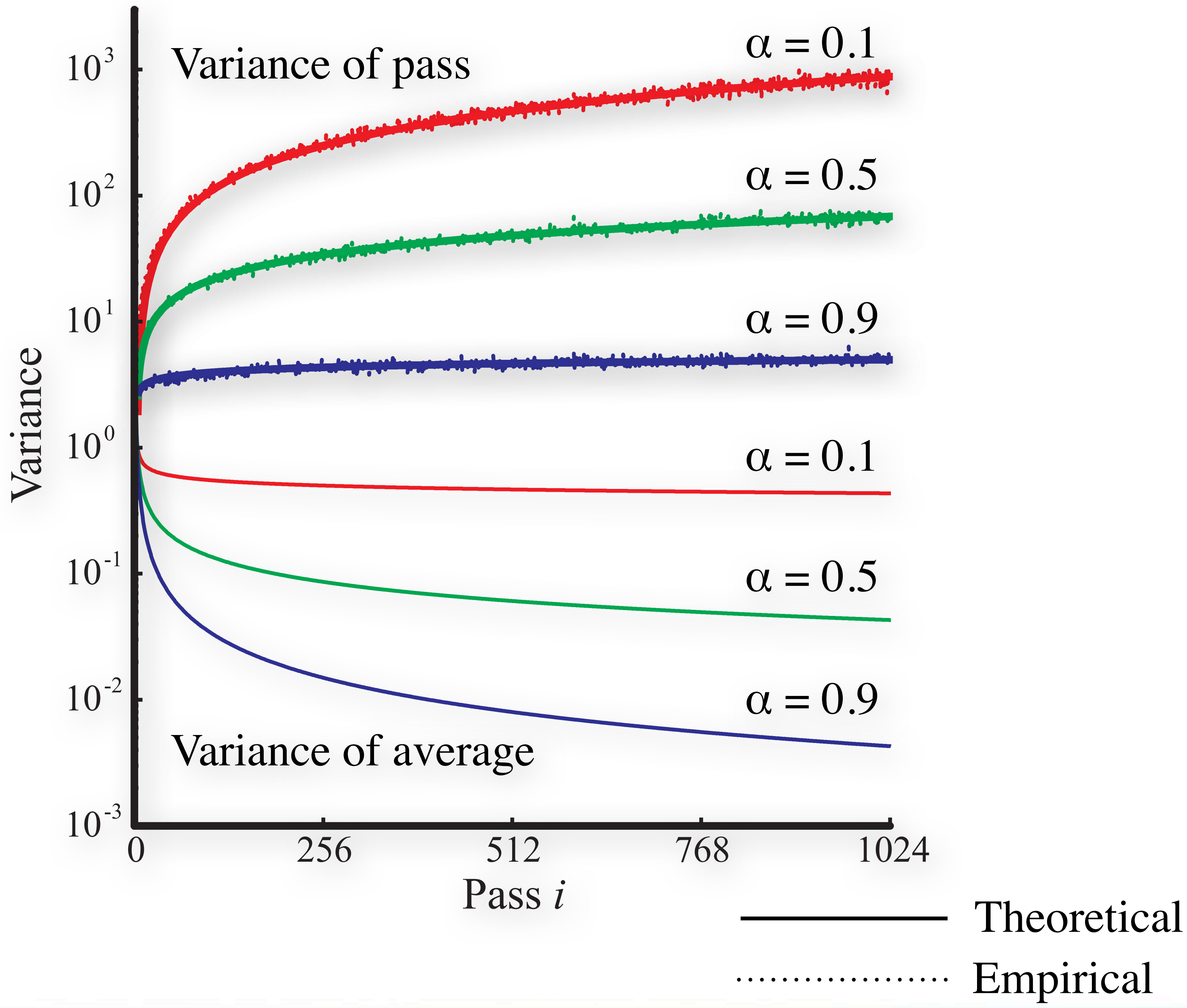
$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$



- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Empirical Validation

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$

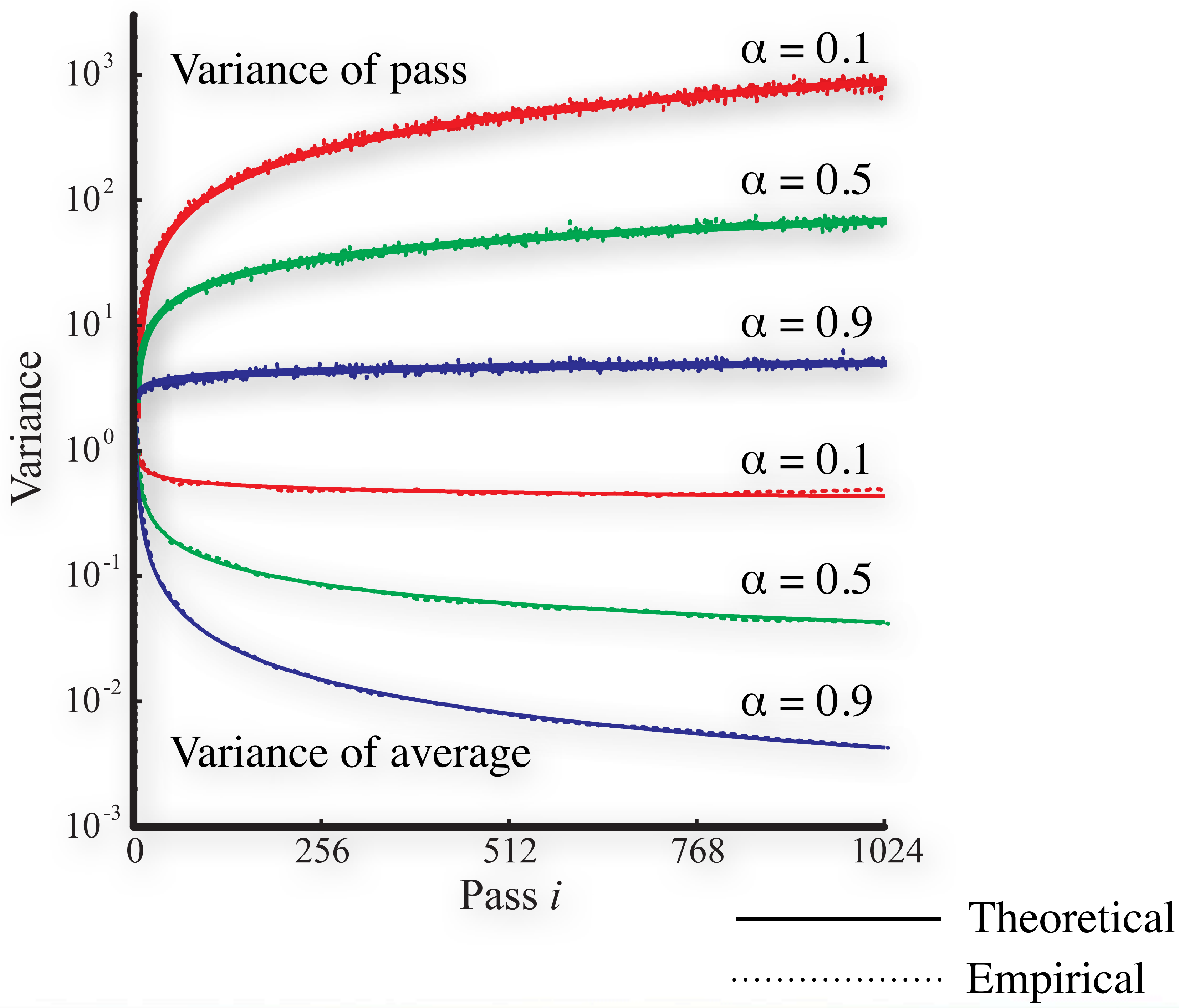


Thursday, 6 September 12

- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Empirical Validation

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$

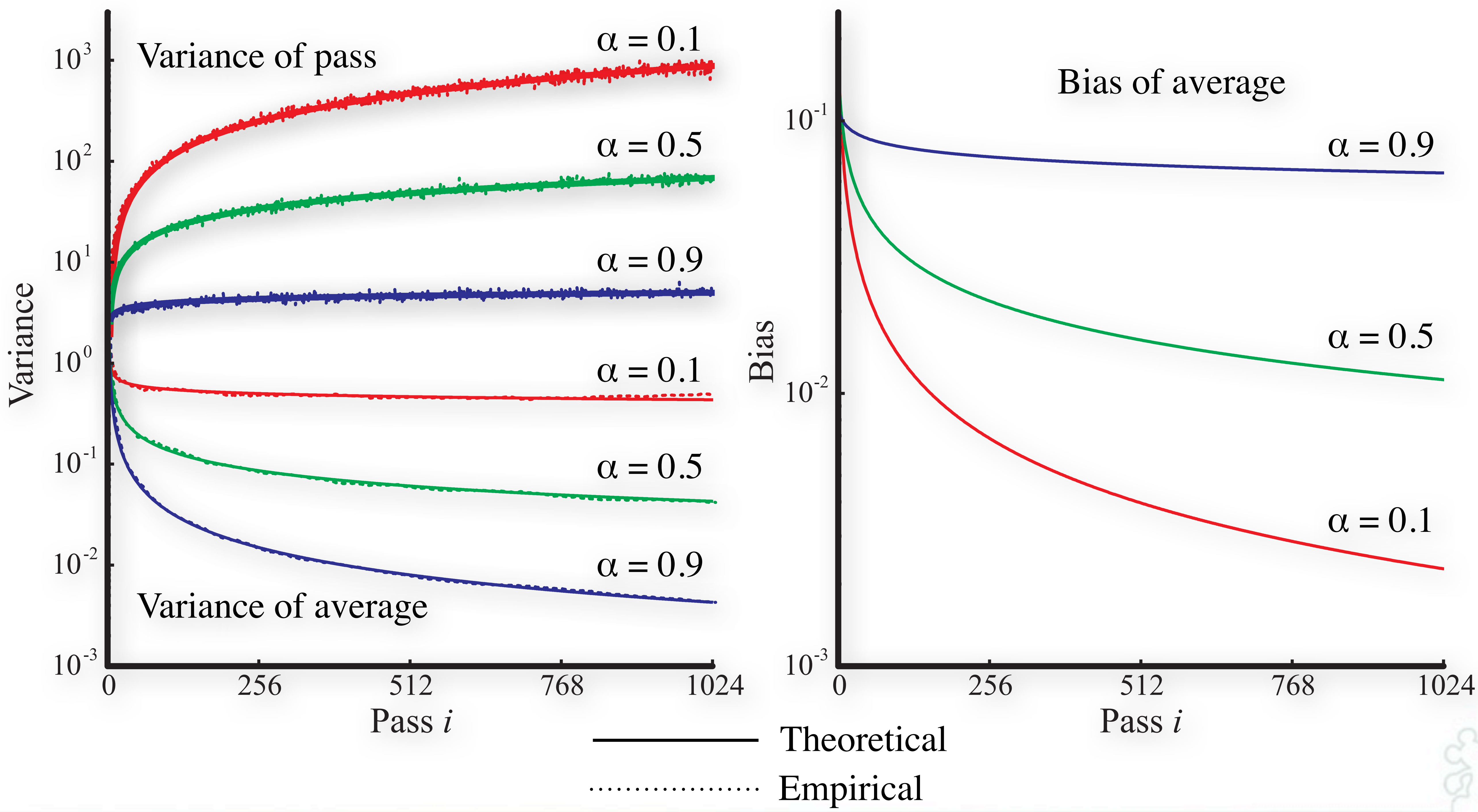


Thursday, 6 September 12

- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Empirical Validation

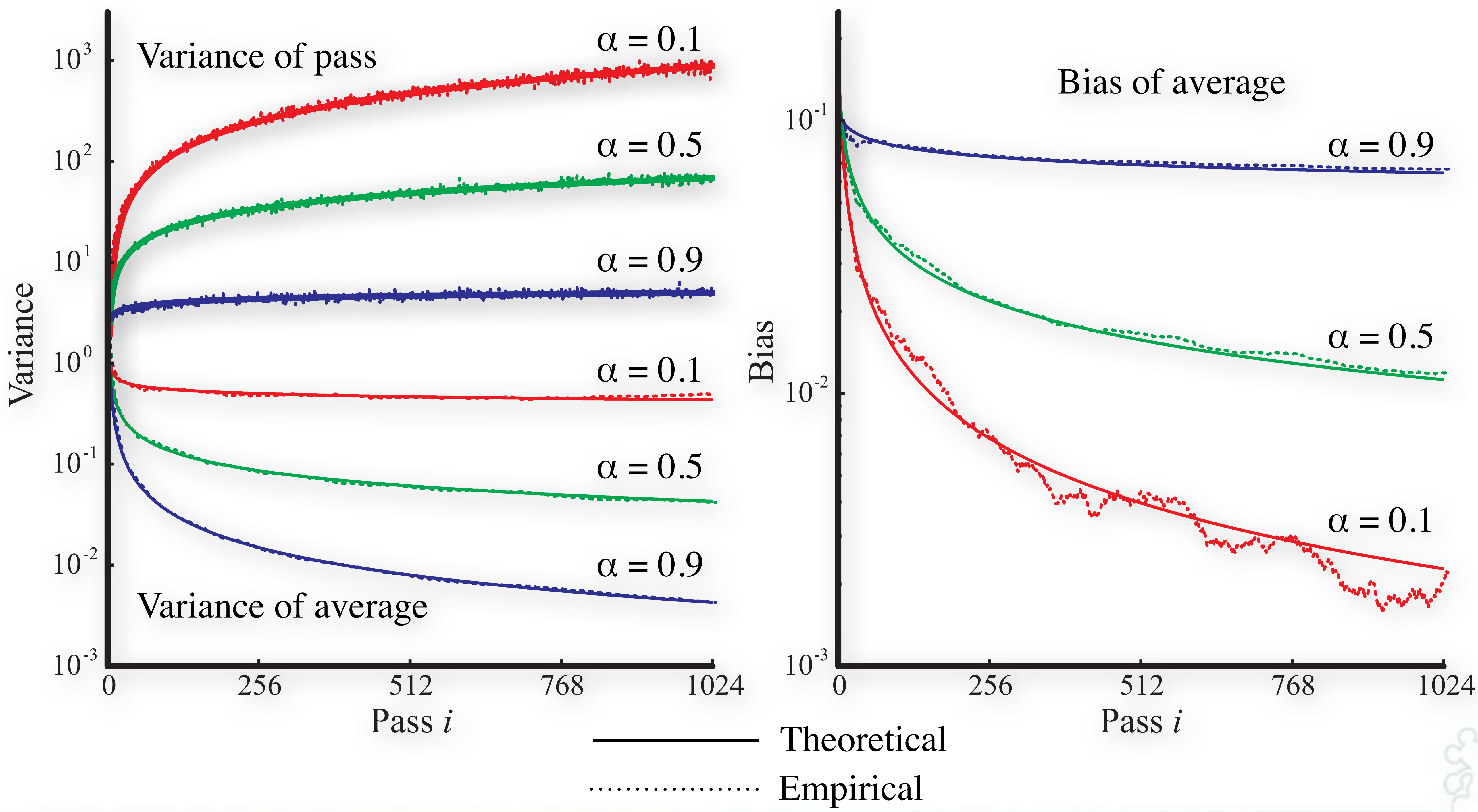
$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$



- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Empirical Validation

$$\frac{\text{Variance}[\text{Pass}_{i+1}]}{\text{Variance}[\text{Pass}_i]} = \frac{i+1}{i+\alpha}$$



Thursday, 6 September 12

- Given our theoretical variance ratio, the variance of each pass should in theory looks like this.
- When alpha is set to be low, the variance of each pass increases quickly. This corresponds to reducing the radii more rapidly
- In contrast, when alpha is set high, the variance of each pass should increase slowly.
- When we plot the sample variance from the data we gathered in our experiment, we see, by using our radius reduction factor, we method strongly agrees with the theoretical prediction.
- More importantly, we are interested in the variance of the running average of the passes
- The theoretical model predicts the following behavior
- Which again, our empirical results confirm.
- Lastly, our theoretical model can also predict the bias of each pass, as well as the bias of the average.
- In theory, the bias should also converge to zero as we increase the number of passes, and we see here that our empirical data strongly agrees with this prediction.
- We see that when alpha is between 0 and 1 both the variance and the bias of the running average converge to zero as predicted by our model. Also, the alpha parameter allows the user to control the relative reduction of bias vs reduction of variance in the average.

Algorithm

Thursday, 6 September 12

- We can now use our algorithm to render participating media in a progressive fashion [click]
- The first step is just standard photon beam tracing, the only change is that we scale the beam radii by the global factor Ri, which starts at 1 [click]
- In the second step, we render the image by tracing random paths through each pixel, and evaluating the radius along each ray using beams [click]
- We then average the image with our previous results, and display this running average [click]
- Finally, we reduce the scaling factor, and repeat
- Note that this is just a simple loop around standard photon beams
- Also, each iteration is in fact independent (it only needs knowledge of the pass number), [click] so we can trivially parallelize this by farming out each pass to a different render node in a large cluster, and average all the resulting images

Algorithm

Step 1:

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

- We can now use our algorithm to render participating media in a progressive fashion [click]
- The first step is just standard photon beam tracing, the only change is that we scale the beam radii by the global factor Ri, which starts at 1 [click]
- In the second step, we render the image by tracing random paths through each pixel, and evaluating the radius along each ray using beams [click]
- We then average the image with our previous results, and display this running average [click]
- Finally, we reduce the scaling factor, and repeat
- Note that this is just a simple loop around standard photon beams
- Also, each iteration is in fact independent (it only needs knowledge of the pass number), [click] so we can trivially parallelize this by farming out each pass to a different render node in a large cluster, and average all the resulting images

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

- We can now use our algorithm to render participating media in a progressive fashion [click]
- The first step is just standard photon beam tracing, the only change is that we scale the beam radii by the global factor Ri, which starts at 1 [click]
- In the second step, we render the image by tracing random paths through each pixel, and evaluating the radius along each ray using beams [click]
- We then average the image with our previous results, and display this running average [click]
- Finally, we reduce the scaling factor, and repeat
- Note that this is just a simple loop around standard photon beams
- Also, each iteration is in fact independent (it only needs knowledge of the pass number), [click] so we can trivially parallelize this by farming out each pass to a different render node in a large cluster, and average all the resulting images

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

- We can now use our algorithm to render participating media in a progressive fashion [click]
- The first step is just standard photon beam tracing, the only change is that we scale the beam radii by the global factor Ri, which starts at 1 [click]
- In the second step, we render the image by tracing random paths through each pixel, and evaluating the radius along each ray using beams [click]
- We then average the image with our previous results, and display this running average [click]
- Finally, we reduce the scaling factor, and repeat
- Note that this is just a simple loop around standard photon beams
- Also, each iteration is in fact independent (it only needs knowledge of the pass number), [click] so we can trivially parallelize this by farming out each pass to a different render node in a large cluster, and average all the resulting images

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*

- We can now use our algorithm to render participating media in a progressive fashion [click]
- The first step is just standard photon beam tracing, the only change is that we scale the beam radii by the global factor Ri, which starts at 1 [click]
- In the second step, we render the image by tracing random paths through each pixel, and evaluating the radius along each ray using beams [click]
- We then average the image with our previous results, and display this running average [click]
- Finally, we reduce the scaling factor, and repeat
- Note that this is just a simple loop around standard photon beams
- Also, each iteration is in fact independent (it only needs knowledge of the pass number), [click] so we can trivially parallelize this by farming out each pass to a different render node in a large cluster, and average all the resulting images

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*

Trivially Parallelizable

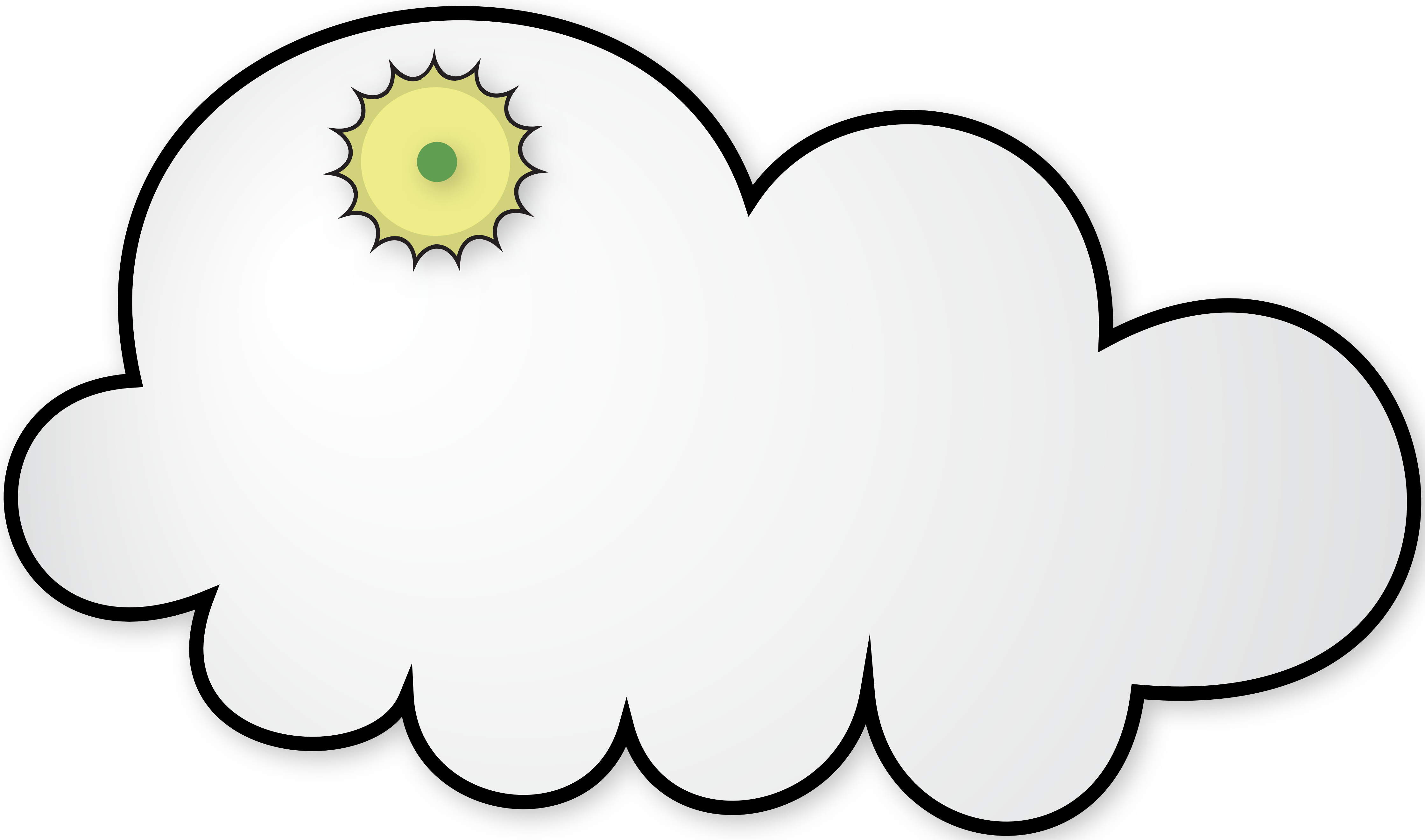
- We can now use our algorithm to render participating media in a progressive fashion [click]
- The first step is just standard photon beam tracing, the only change is that we scale the beam radii by the global factor Ri, which starts at 1 [click]
- In the second step, we render the image by tracing random paths through each pixel, and evaluating the radius along each ray using beams [click]
- We then average the image with our previous results, and display this running average [click]
- Finally, we reduce the scaling factor, and repeat
- Note that this is just a simple loop around standard photon beams
- Also, each iteration is in fact independent (it only needs knowledge of the pass number), [click] so we can trivially parallelize this by farming out each pass to a different render node in a large cluster, and average all the resulting images

Evaluating the Transmittance

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

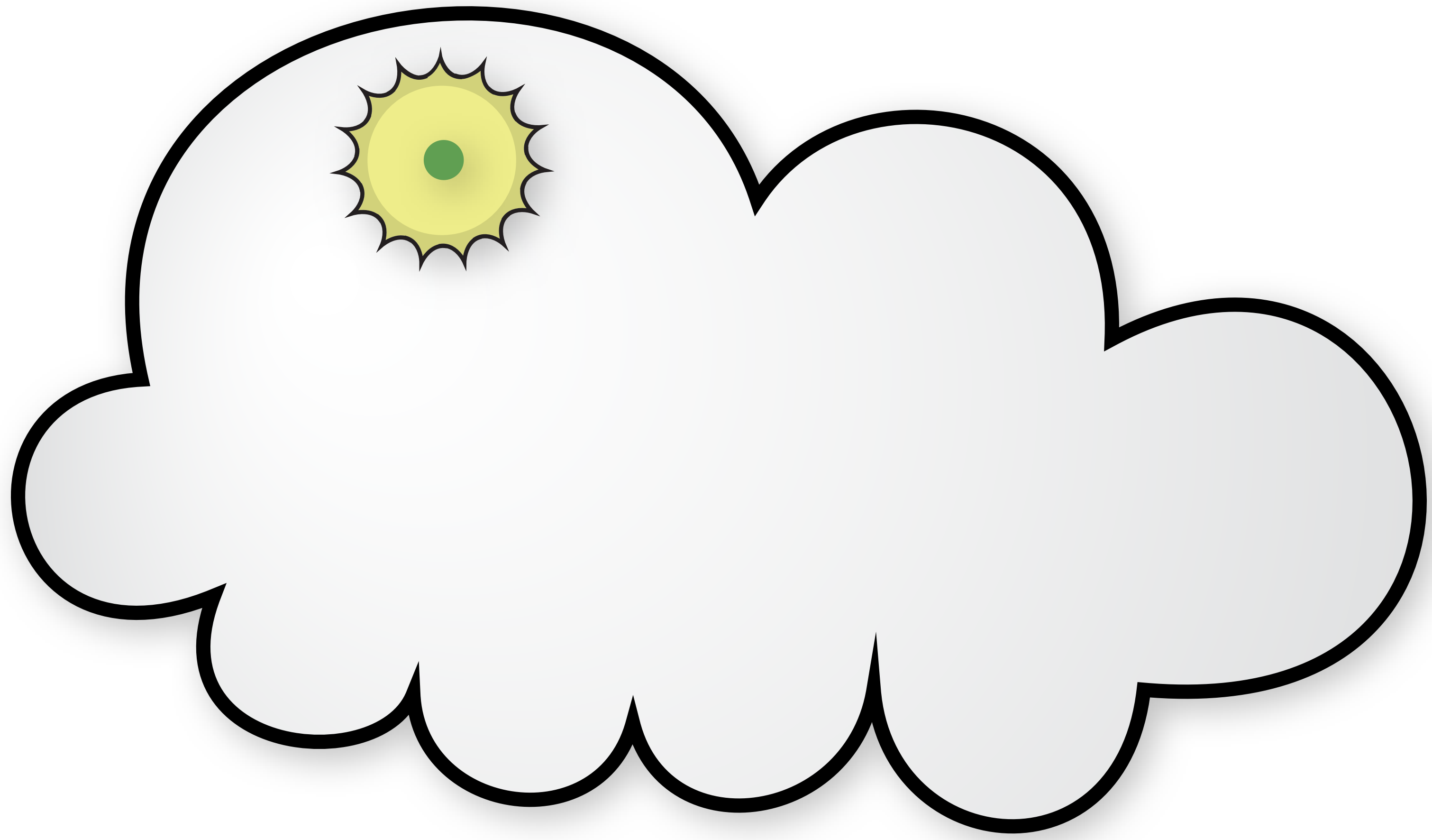
- Now, lets turn to some practical details
- During rendering, to compute the contribution of each beam, we need to compute the transmittance along the beam, as well as along the camera ray.

Homogeneous Media



- In homogeneous media, transmittance can be computed analytically, and so the intensity of each beam can be described using this exponential attenuation curve

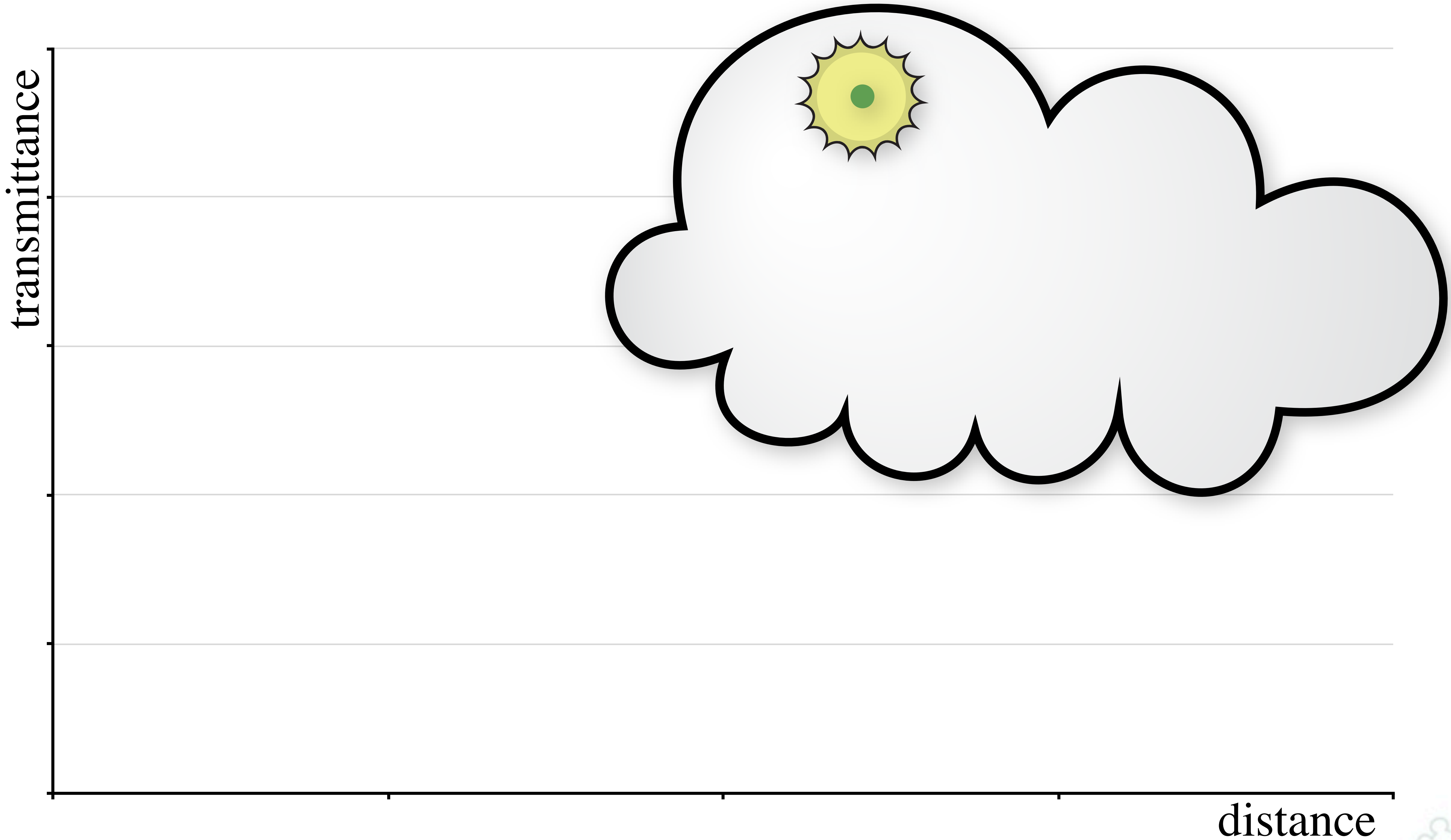
Homogeneous Media



Thursday, 6 September 12

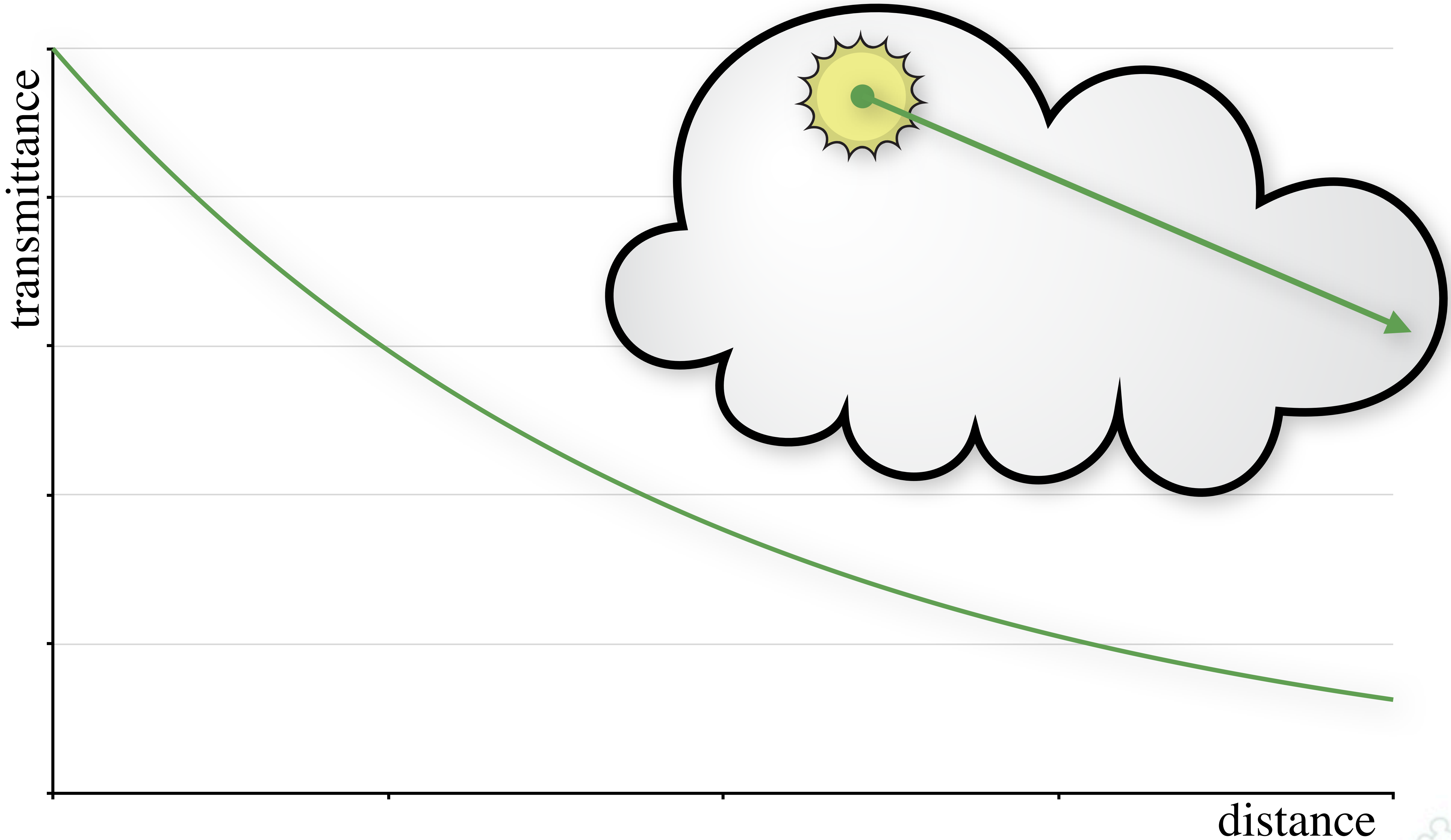
- In homogeneous media, transmittance can be computed analytically, and so the intensity of each beam can be described using this exponential attenuation curve

Homogeneous Media



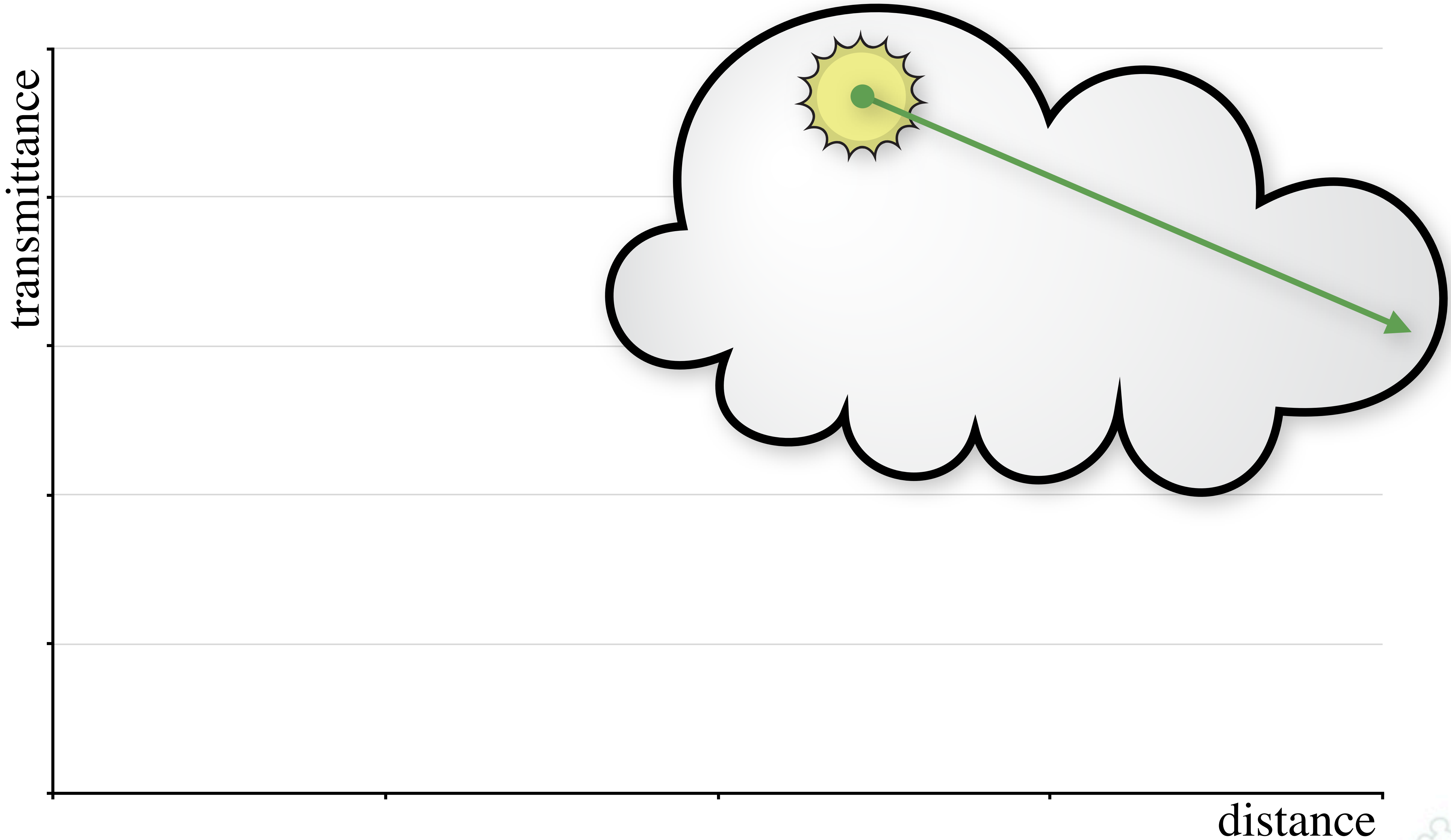
- In homogeneous media, transmittance can be computed analytically, and so the intensity of each beam can be described using this exponential attenuation curve

Homogeneous Media



- In homogeneous media, transmittance can be computed analytically, and so the intensity of each beam can be described using this exponential attenuation curve

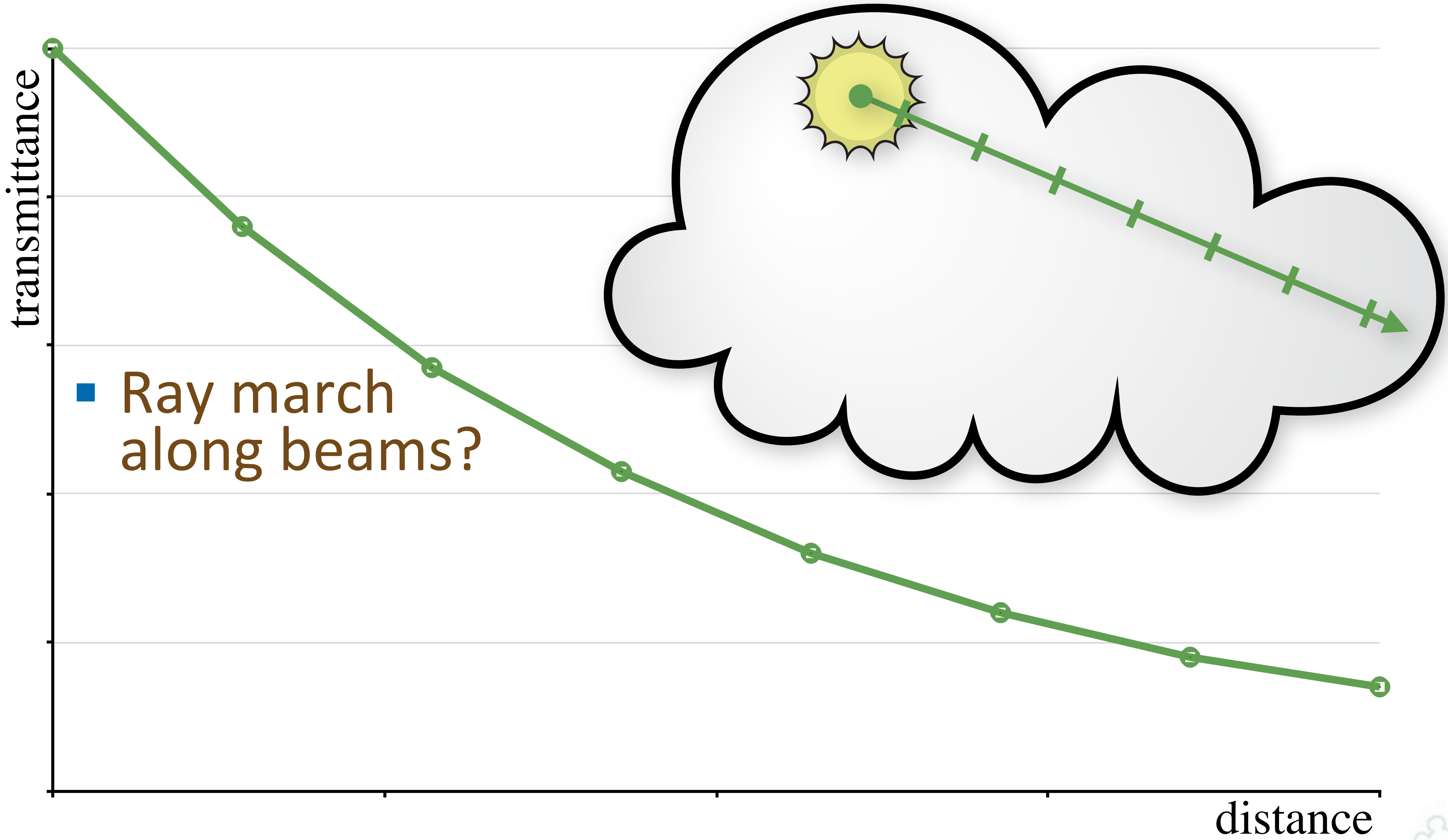
Heterogeneous Media



Thursday, 6 September 12

- The solution we proposed in the original photon beams paper was to use ray marching to handle heterogeneous media [click]
- This is efficient since we can just ray march along each beam in the preprocess, cache this piecewise linear transmittance approximation, and in constant-time evaluate the transmittance for any camera ray/beam intersection during rendering [click]
- However, ray marching introduces additional bias!
- We unfortunately cannot use our progressive theory with ray-marched transmittance since this additional bias is not considered in our error analysis. [click]
- To obtain convergence in heterogeneous media, we need an unbiased transmittance estimator

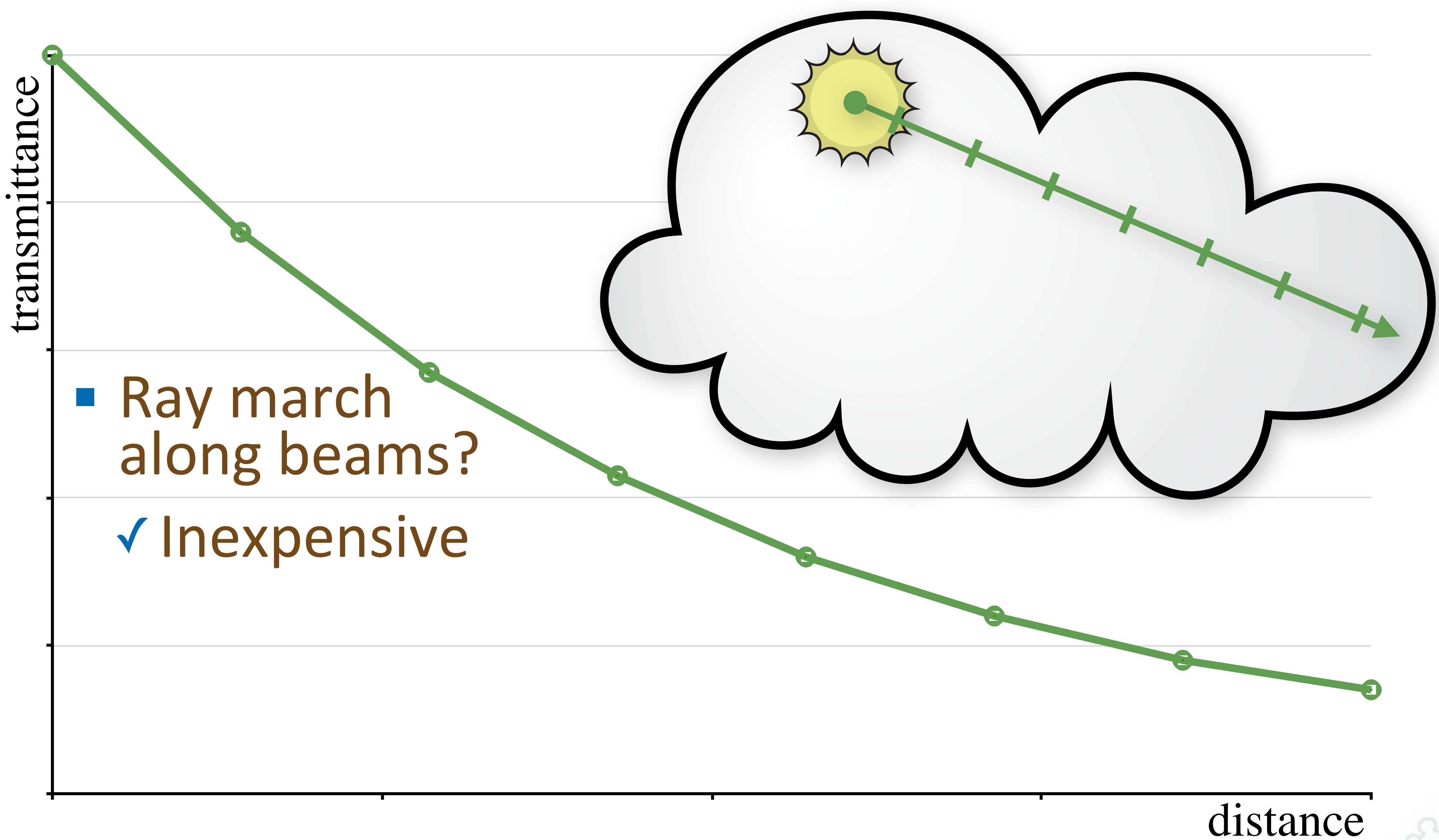
Heterogeneous Media



Thursday, 6 September 12

- The solution we proposed in the original photon beams paper was to use ray marching to handle heterogeneous media [click]
- This is efficient since we can just ray march along each beam in the preprocess, cache this piecewise linear transmittance approximation, and in constant-time evaluate the transmittance for any camera ray/beam intersection during rendering [click]
- However, ray marching introduces additional bias!
- We unfortunately cannot use our progressive theory with ray-marched transmittance since this additional bias is not considered in our error analysis. [click]
- To obtain convergence in heterogeneous media, we need an unbiased transmittance estimator

Heterogeneous Media

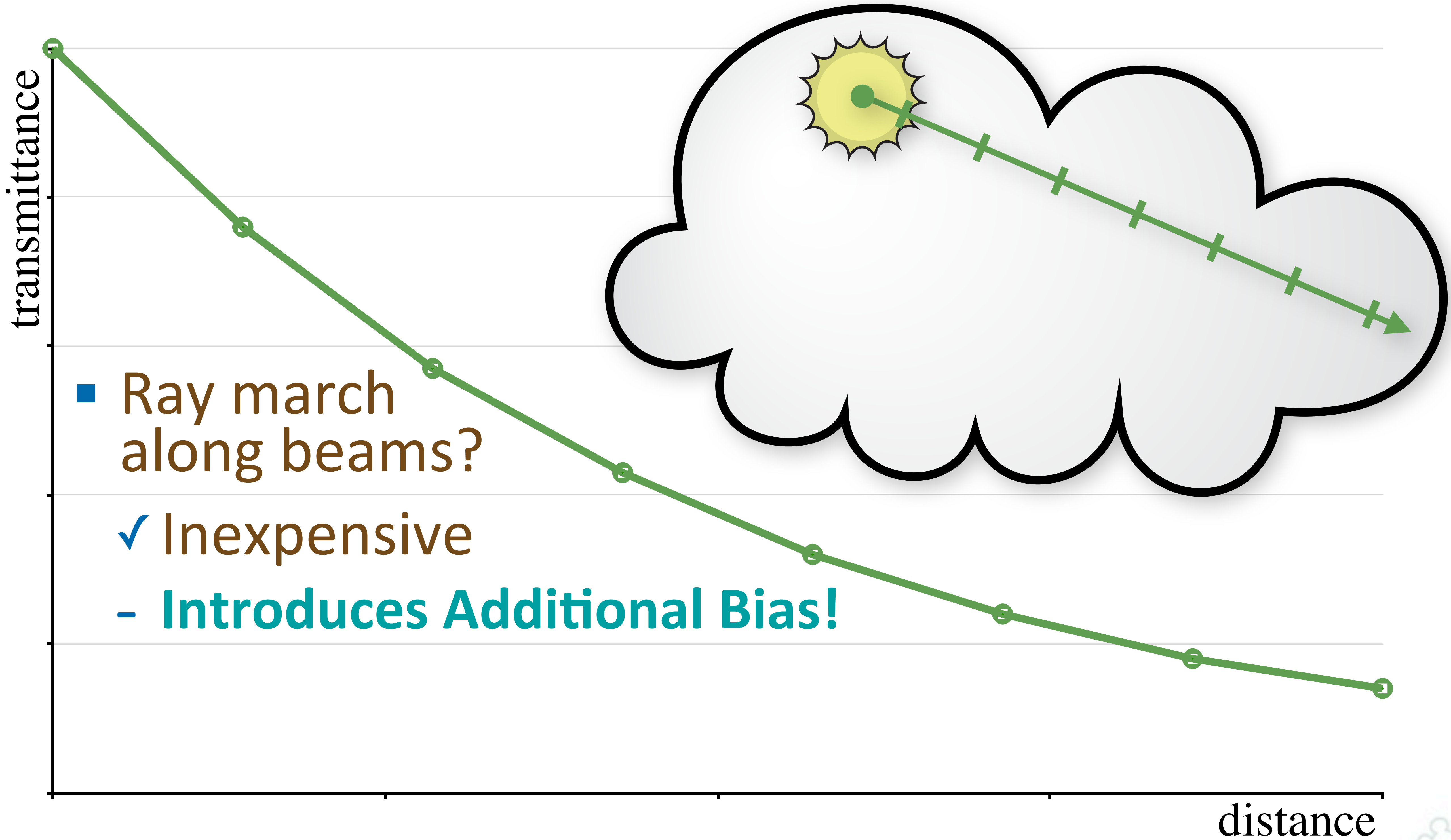


- Ray march along beams?
  - ✓ Inexpensive

Thursday, 6 September 12

- The solution we proposed in the original photon beams paper was to use ray marching to handle heterogeneous media [click]
- This is efficient since we can just ray march along each beam in the preprocess, cache this piecewise linear transmittance approximation, and in constant-time evaluate the transmittance for any camera ray/beam intersection during rendering [click]
- However, ray marching introduces additional bias!
- We unfortunately cannot use our progressive theory with ray-marched transmittance since this additional bias is not considered in our error analysis. [click]
- To obtain convergence in heterogeneous media, we need an unbiased transmittance estimator

Heterogeneous Media

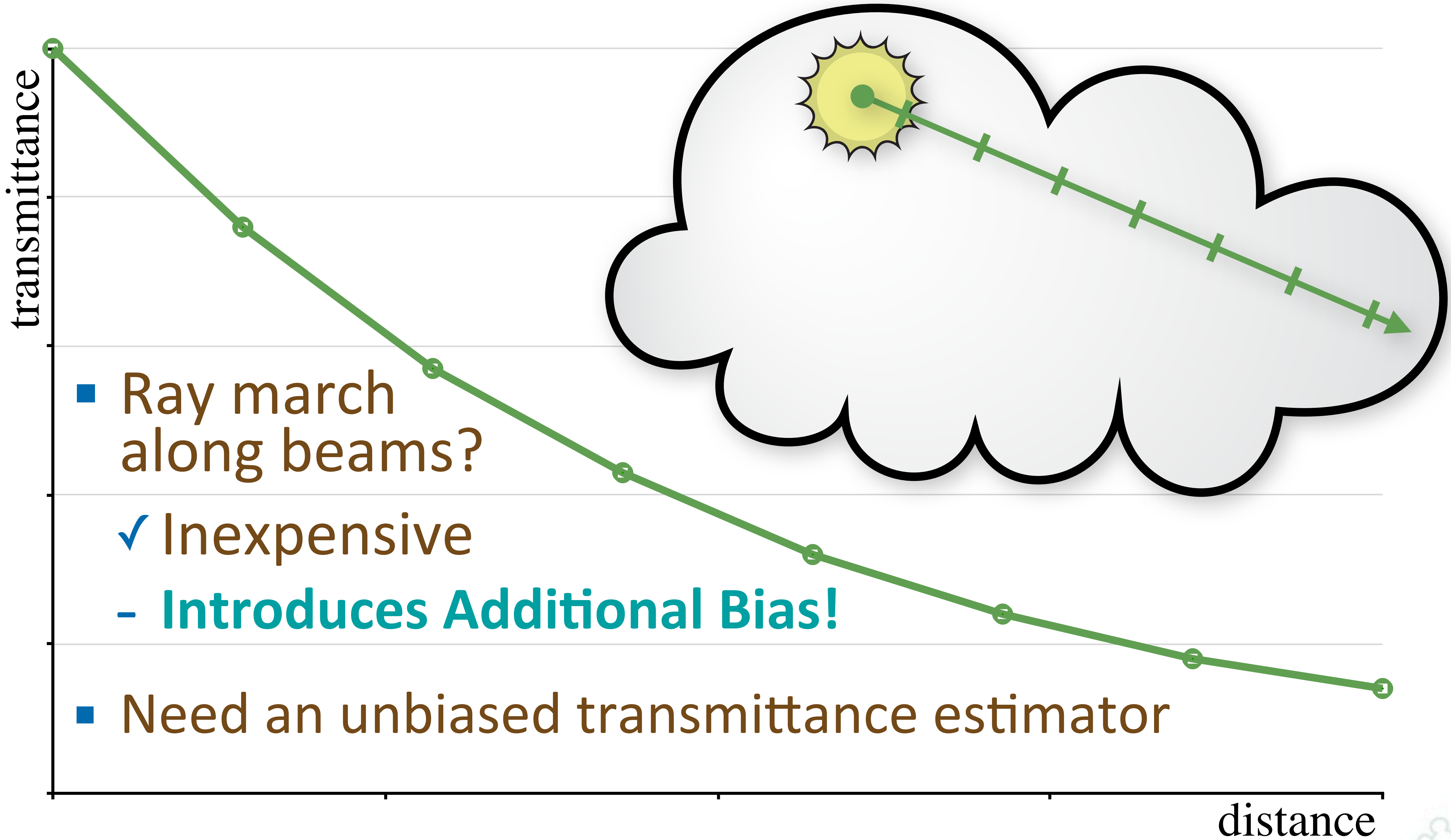


- Ray march along beams?
  - ✓ Inexpensive
  - – **Introduces Additional Bias!**

Thursday, 6 September 12

- The solution we proposed in the original photon beams paper was to use ray marching to handle heterogeneous media [click]
- This is efficient since we can just ray march along each beam in the preprocess, cache this piecewise linear transmittance approximation, and in constant-time evaluate the transmittance for any camera ray/beam intersection during rendering [click]
- However, ray marching introduces additional bias!
- We unfortunately cannot use our progressive theory with ray-marched transmittance since this additional bias is not considered in our error analysis. [click]
- To obtain convergence in heterogeneous media, we need an unbiased transmittance estimator

Heterogeneous Media



- Ray march along beams?
  - ✓ Inexpensive
  - – **Introduces Additional Bias!**
- Need an unbiased transmittance estimator

Thursday, 6 September 12

- The solution we proposed in the original photon beams paper was to use ray marching to handle heterogeneous media [click]
- This is efficient since we can just ray march along each beam in the preprocess, cache this piecewise linear transmittance approximation, and in constant-time evaluate the transmittance for any camera ray/beam intersection during rendering [click]
- However, ray marching introduces additional bias!
- We unfortunately cannot use our progressive theory with ray-marched transmittance since this additional bias is not considered in our error analysis. [click]
- To obtain convergence in heterogeneous media, we need an unbiased transmittance estimator

Unbiased Heterogeneous Transmittance

- Unbiased transmittance estimators exist
  - [Woodcock et al. 1965]
  - [Raab et al. 2008]
  - [Yue et al. 2010]
  - [Szirmay-Kalos et al. 2011]

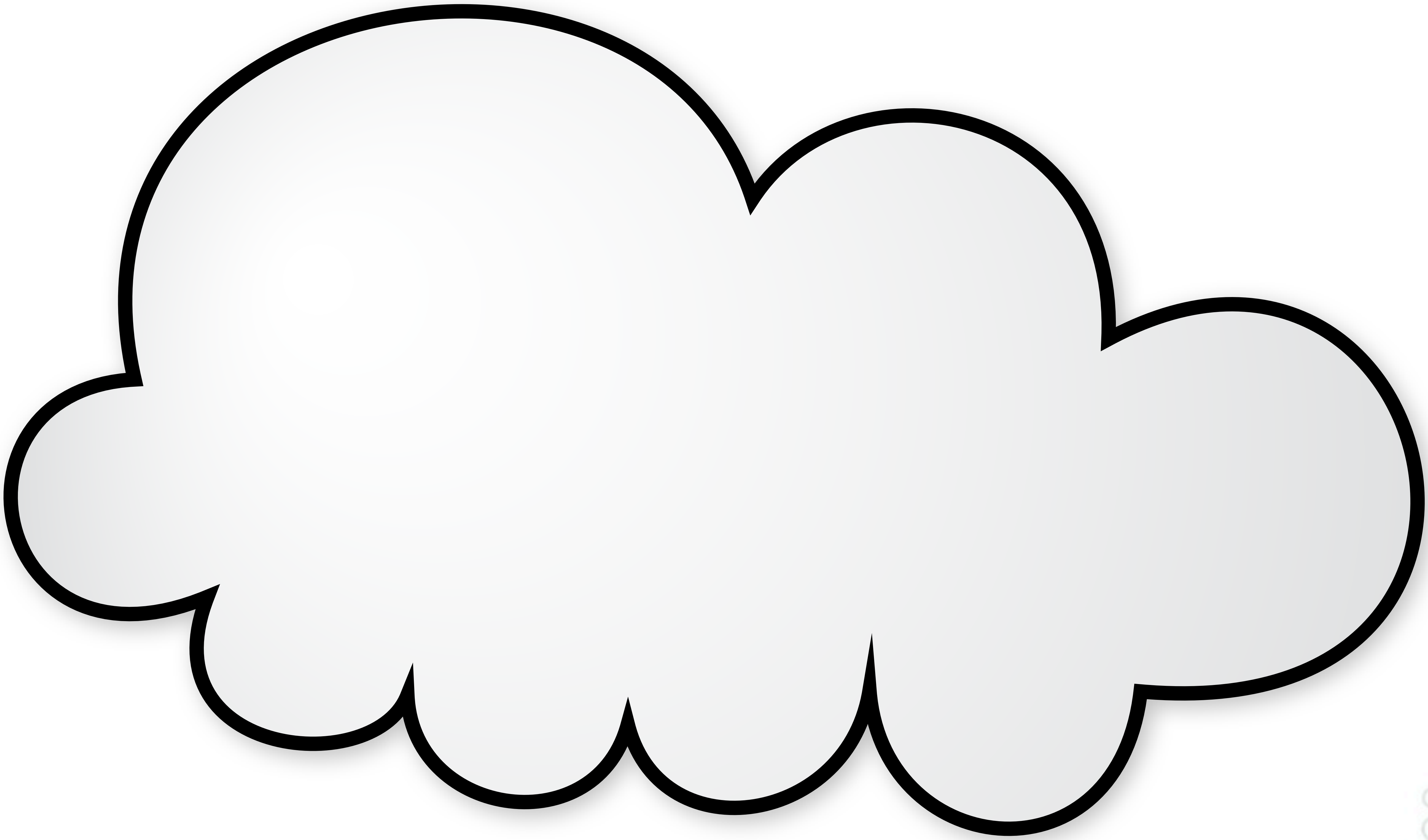
- Unbiased transmittance estimators do in fact exist [click]
- The problem is that they are slow and noisy
- and, they are not cache-able. Meaning, we cannot store some representation along each beam, and quickly re-evaluate during rendering like we did with ray marching

Unbiased Heterogeneous Transmittance

- Unbiased transmittance estimators exist
  - [Woodcock et al. 1965]
  - [Raab et al. 2008]
  - [Yue et al. 2010]
  - [Szirmay-Kalos et al. 2011]
  - slow/noisy
  - not cache-able

- Unbiased transmittance estimators do in fact exist [click]
- The problem is that they are slow and noisy
- and, they are not cache-able. Meaning, we cannot store some representation along each beam, and quickly re-evaluate during rendering like we did with ray marching

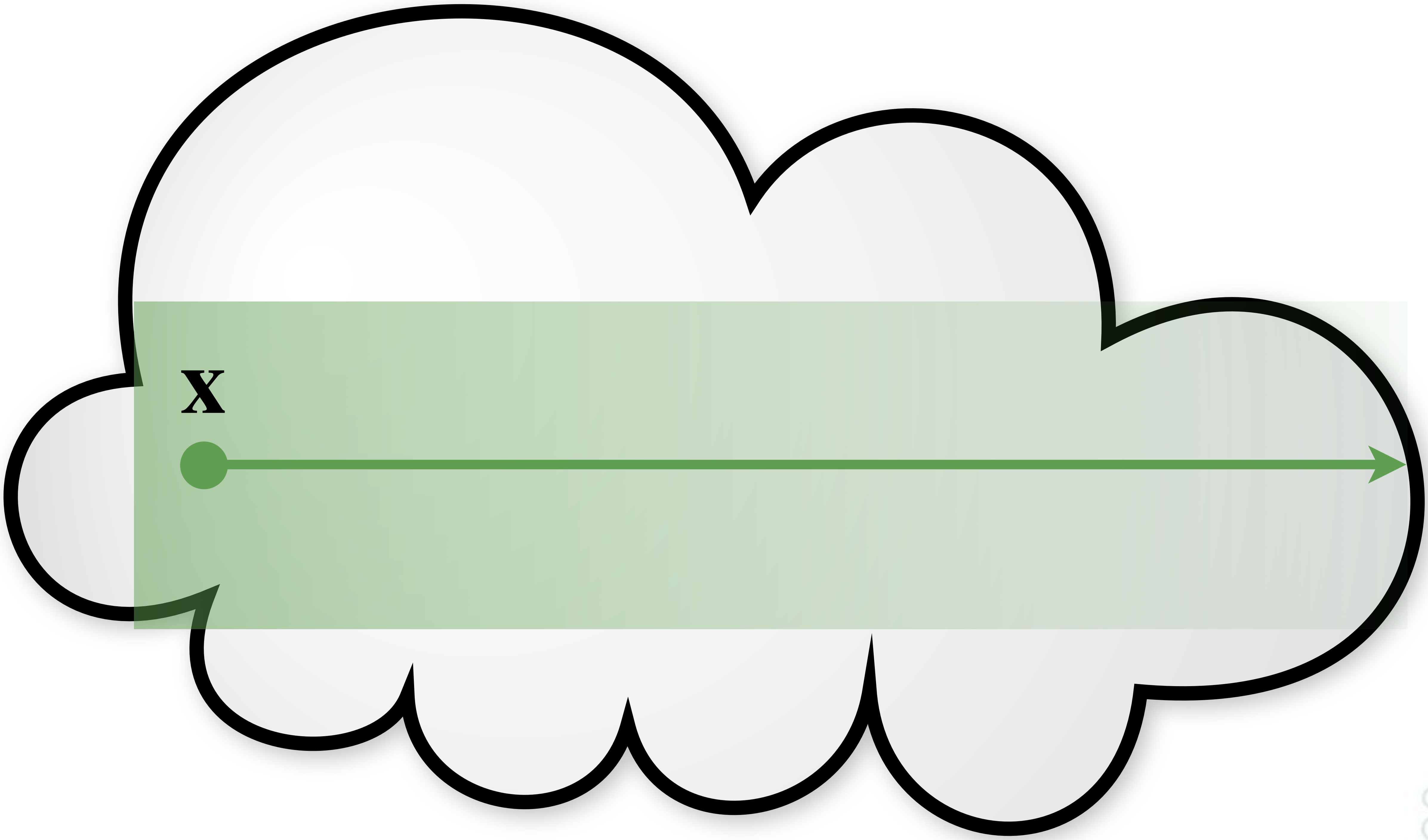
Transmittance using Free-flight Distance



Thursday, 6 September 12

- If we want to compute the transmittance along the beam to some location y [click]
- The standard way to do this in an unbiased way is to generate a random free-flight propagation distance [click], d (this can be done for both homogeneous media, and heterogeneous media using something like woodcock tracking) [click]
- we can then estimate transmittance by comparing whether this distance is greater than or less than the distance from x to y [click]
- Depending on where our random distance lands, we estimate transmittance as either 0 or 1
- Now, this is unbiased, but as I said, its also extremely noisy

Transmittance using Free-flight Distance

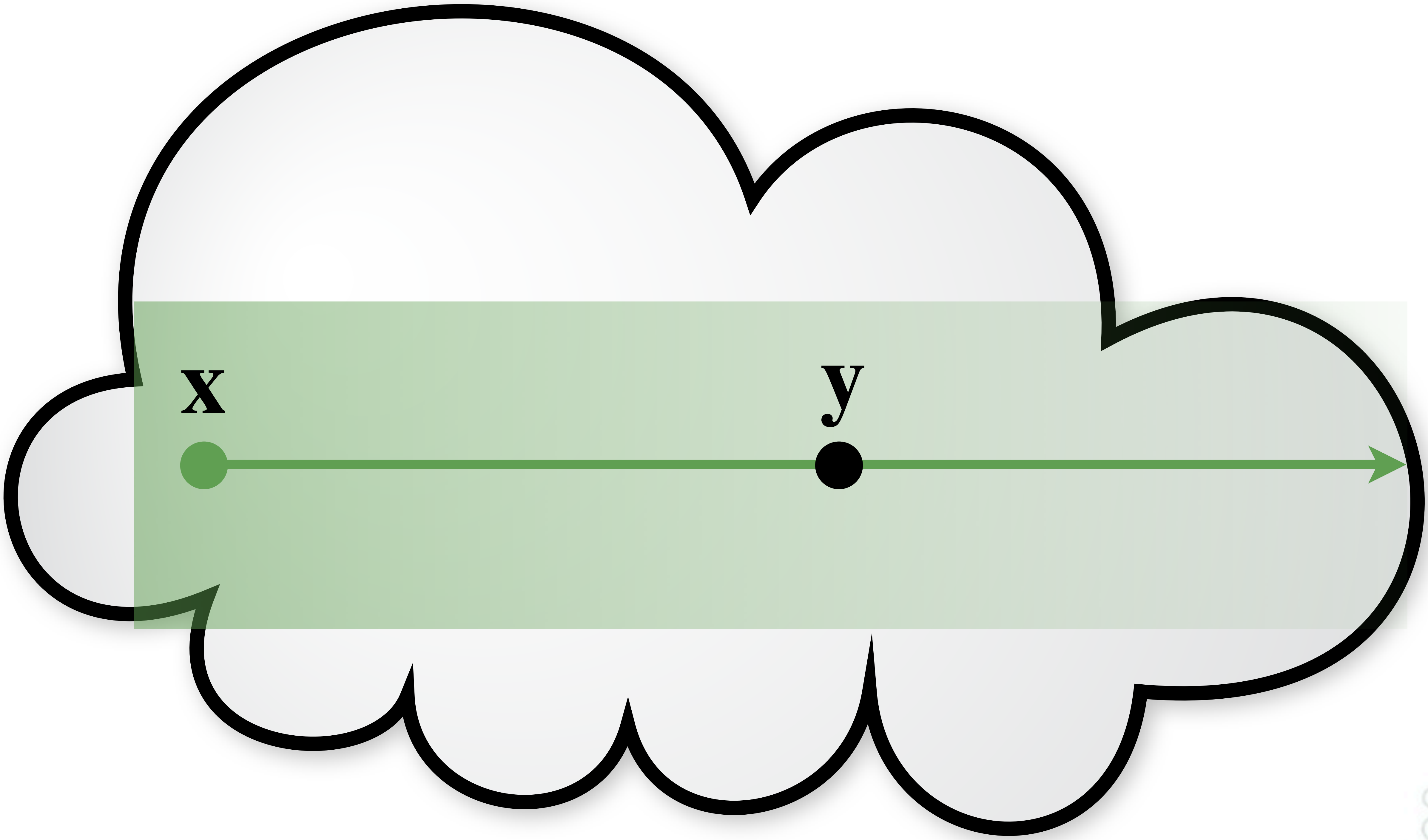


Thursday, 6 September 12

- If we want to compute the transmittance along the beam to some location y [click]
- The standard way to do this in an unbiased way is to generate a random free-flight propagation distance [click], d (this can be done for both homogeneous media, and heterogeneous media using something like woodcock tracking) [click]
- we can then estimate transmittance by comparing whether this distance is greater than or less than the distance from x to y [click]
- Depending on where our random distance lands, we estimate transmittance as either 0 or 1
- Now, this is unbiased, but as I said, its also extremely noisy

Transmittance using Free-flight Distance

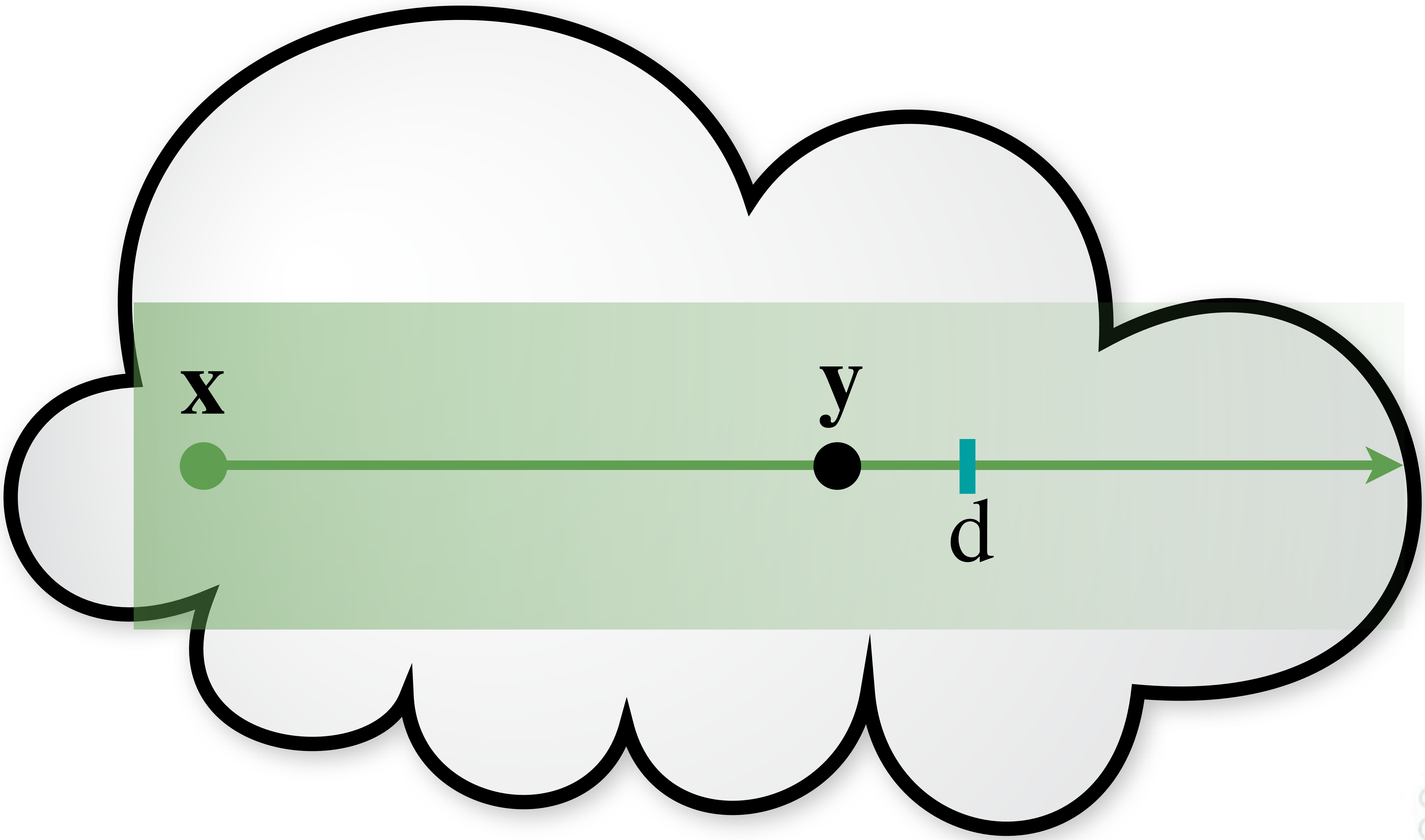
$$T_r(\mathbf{x}, \mathbf{y}) = ?$$



- If we want to compute the transmittance along the beam to some location y [click]
- The standard way to do this in an unbiased way is to generate a random free-flight propagation distance [click], d (this can be done for both homogeneous media, and heterogeneous media using something like woodcock tracking) [click]
- we can then estimate transmittance by comparing whether this distance is greater than or less than the distance from x to y [click]
- Depending on where our random distance lands, we estimate transmittance as either 0 or 1
- Now, this is unbiased, but as I said, its also extremely noisy

Transmittance using Free-flight Distance

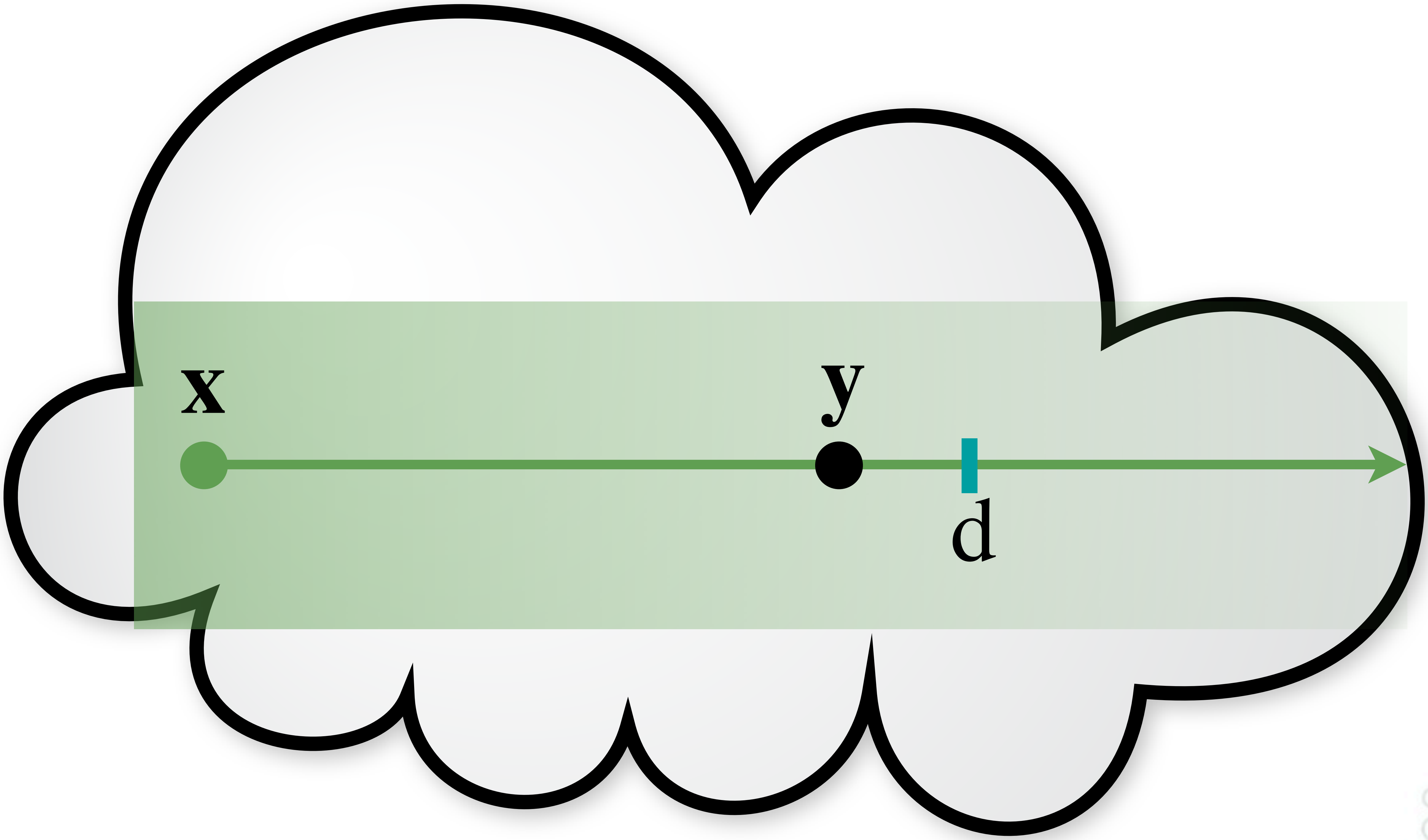
$$T_r(\mathbf{x}, \mathbf{y}) = ?$$



- If we want to compute the transmittance along the beam to some location y [click]
- The standard way to do this in an unbiased way is to generate a random free-flight propagation distance [click], d (this can be done for both homogeneous media, and heterogeneous media using something like woodcock tracking) [click]
- we can then estimate transmittance by comparing whether this distance is greater than or less than the distance from x to y [click]
- Depending on where our random distance lands, we estimate transmittance as either 0 or 1
- Now, this is unbiased, but as I said, its also extremely noisy

Transmittance using Free-flight Distance

$$T_r(\mathbf{x}, \mathbf{y}) \approx \|\mathbf{y} - \mathbf{x}\| < \mathrm{d}$$

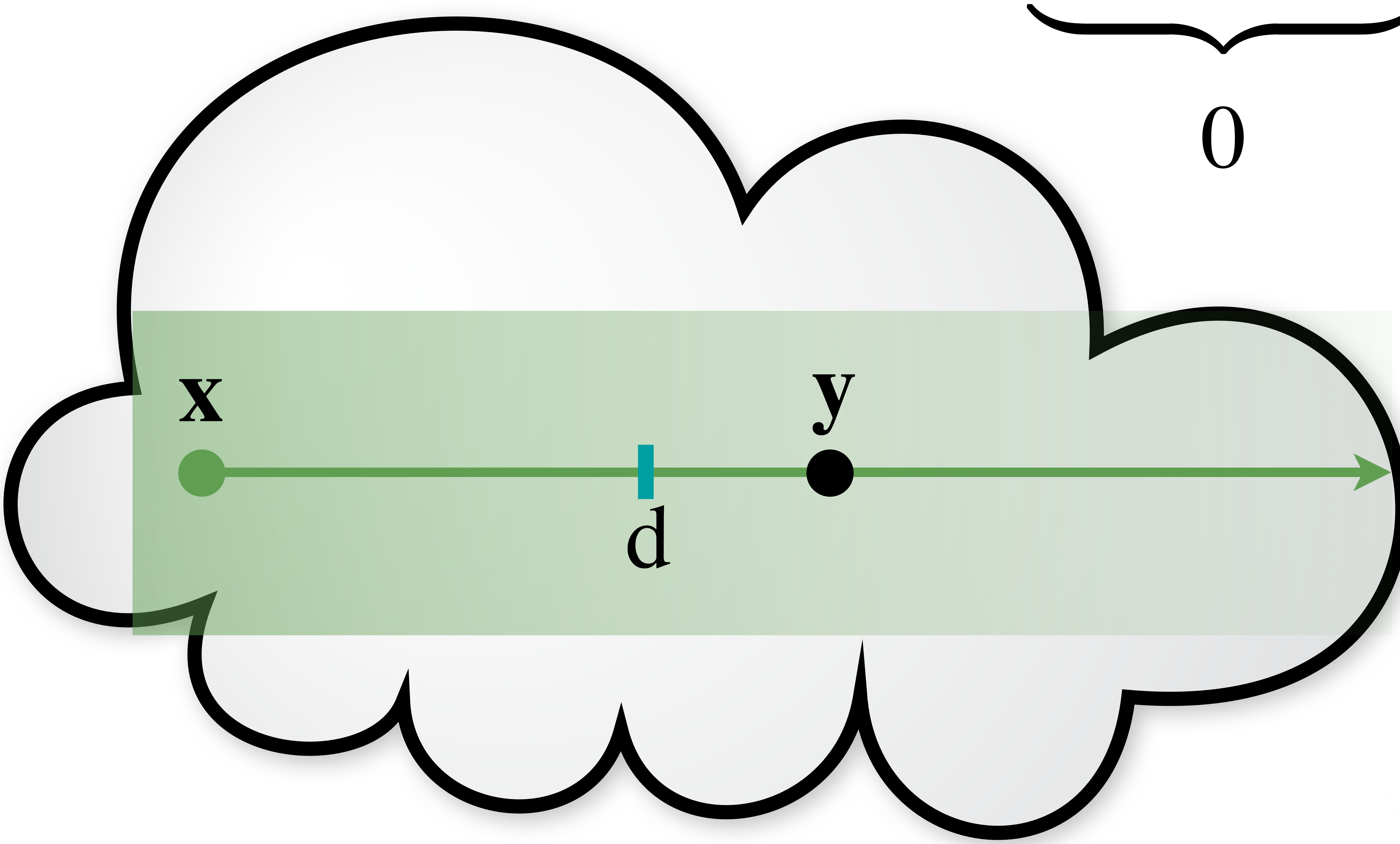


Thursday, 6 September 12

- If we want to compute the transmittance along the beam to some location y [click]
- The standard way to do this in an unbiased way is to generate a random free-flight propagation distance [click], d (this can be done for both homogeneous media, and heterogeneous media using something like woodcock tracking) [click]
- we can then estimate transmittance by comparing whether this distance is greater than or less than the distance from x to y [click]
- Depending on where our random distance lands, we estimate transmittance as either 0 or 1
- Now, this is unbiased, but as I said, its also extremely noisy

Transmittance using Free-flight Distance

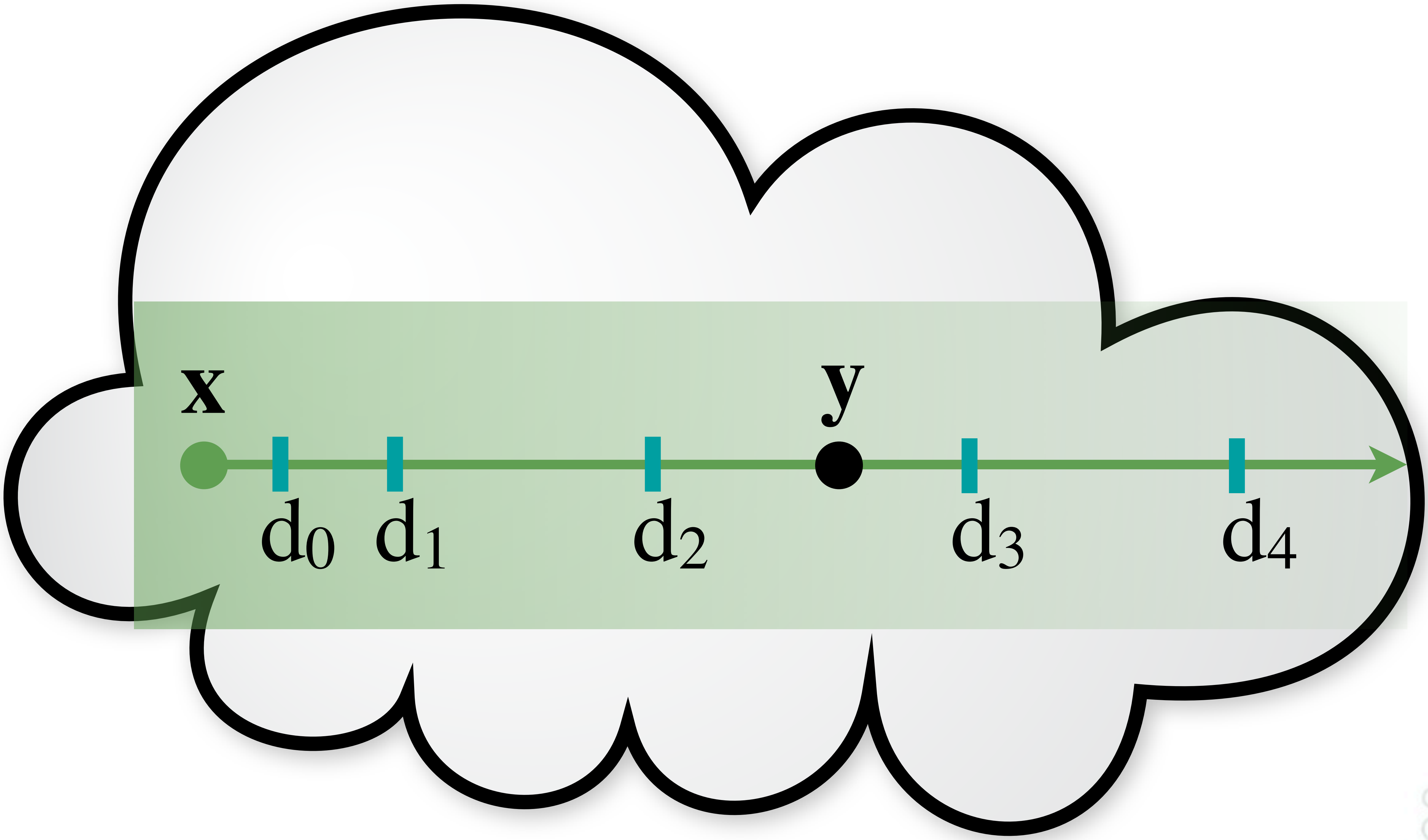
$$T_r(\mathbf{x}, \mathbf{y}) \approx \underbrace{\|\mathbf{y} - \mathbf{x}\| < \mathrm{d}}_{0}$$



- If we want to compute the transmittance along the beam to some location y [click]
- The standard way to do this in an unbiased way is to generate a random free-flight propagation distance [click], d (this can be done for both homogeneous media, and heterogeneous media using something like woodcock tracking) [click]
- we can then estimate transmittance by comparing whether this distance is greater than or less than the distance from x to y [click]
- Depending on where our random distance lands, we estimate transmittance as either 0 or 1
- Now, this is unbiased, but as I said, its also extremely noisy

Transmittance using Free-flight Distance

- Compute n distances, average: $T_r(\mathbf{x}, \mathbf{y}) \approx \frac{2}{5}$

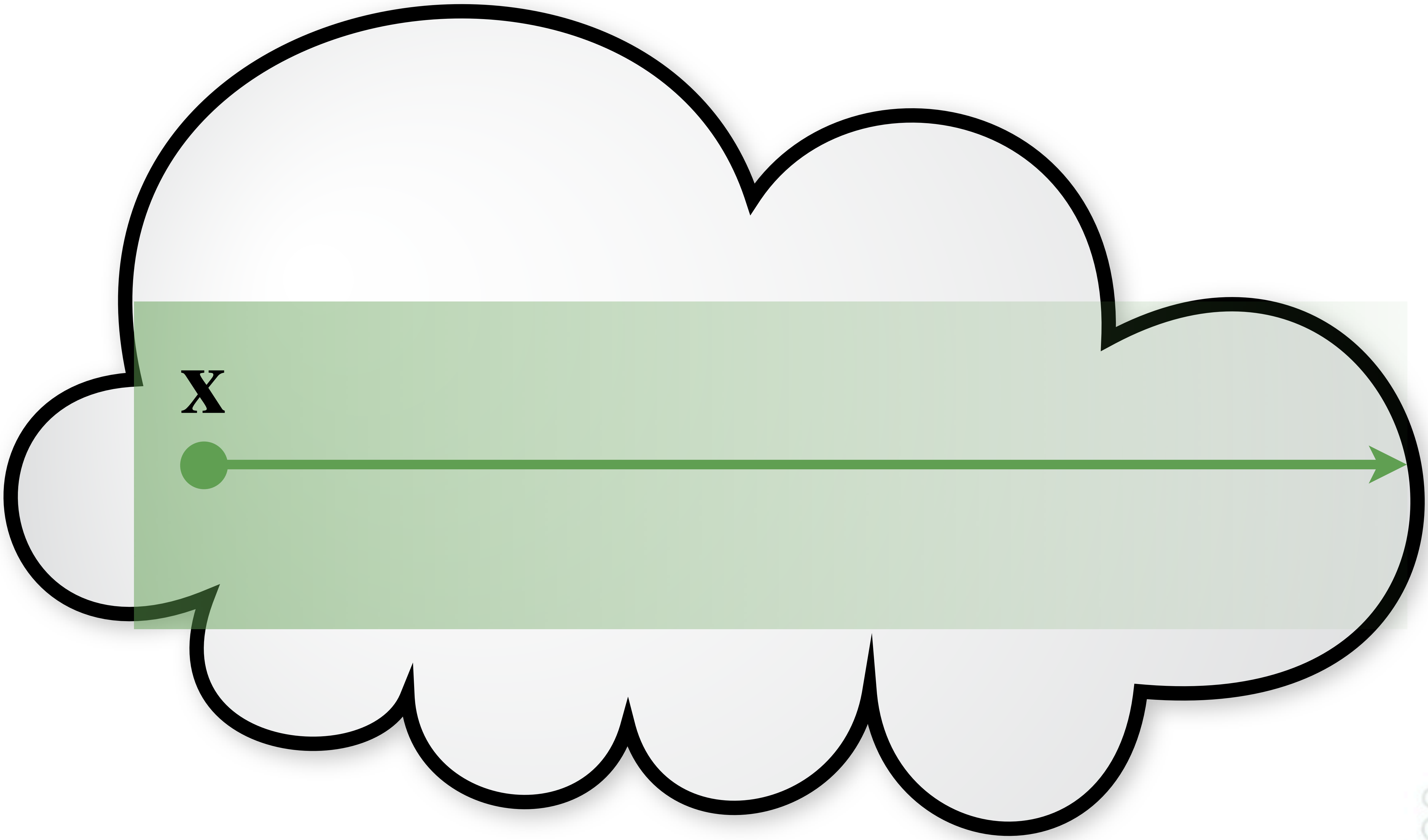


Thursday, 6 September 12

- To improve the quality, you can simply repeat this a number of times: compute several distances (in this case I'm using 5) and then just count how many of those distances propagated past our point y

Transmittance using Free-flight Distance

- Perform this for each ray/beam intersection?

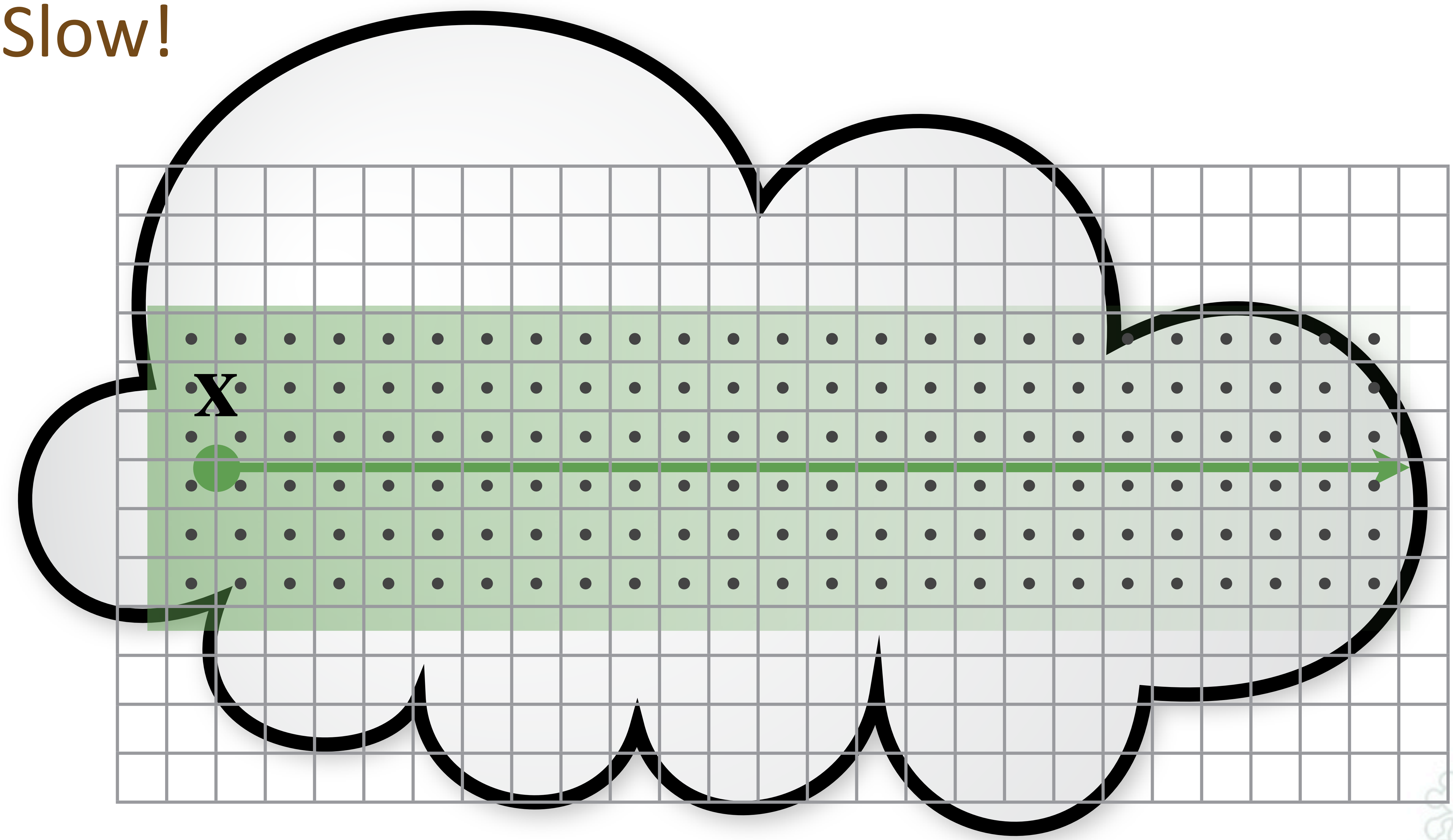


Thursday, 6 September 12

- The problem is that we have to do this for every ray/beam intersection, and there may be [click] thousands of these for a single beam, so repeating this for each intersection would be completely impractical

Transmittance using Free-flight Distance

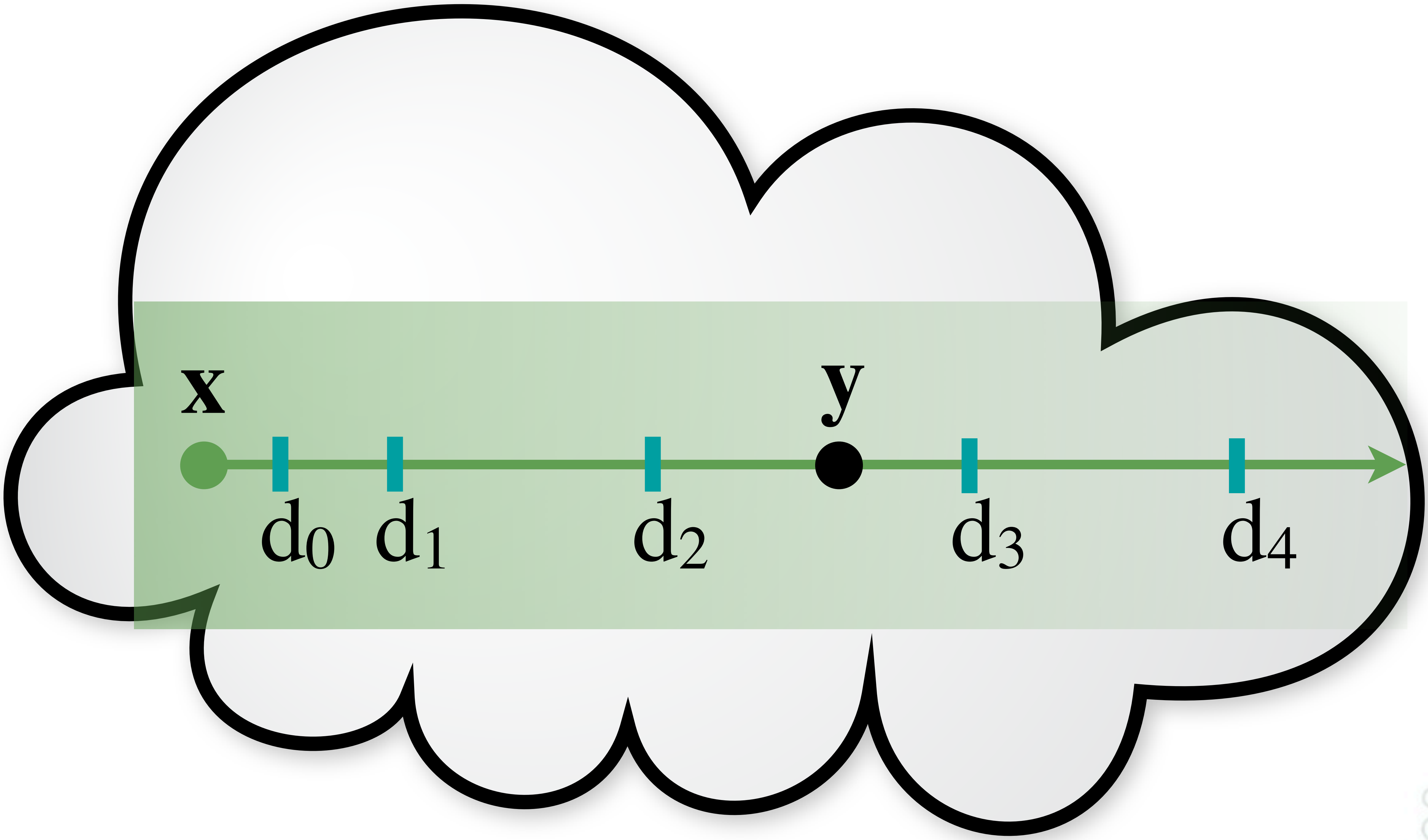
- Perform this for each ray/beam intersection?
- Slow!



- The problem is that we have to do this for every ray/beam intersection, and there may be [click] thousands of these for a single beam, so repeating this for each intersection would be completely impractical

Transmittance using Free-flight Distance

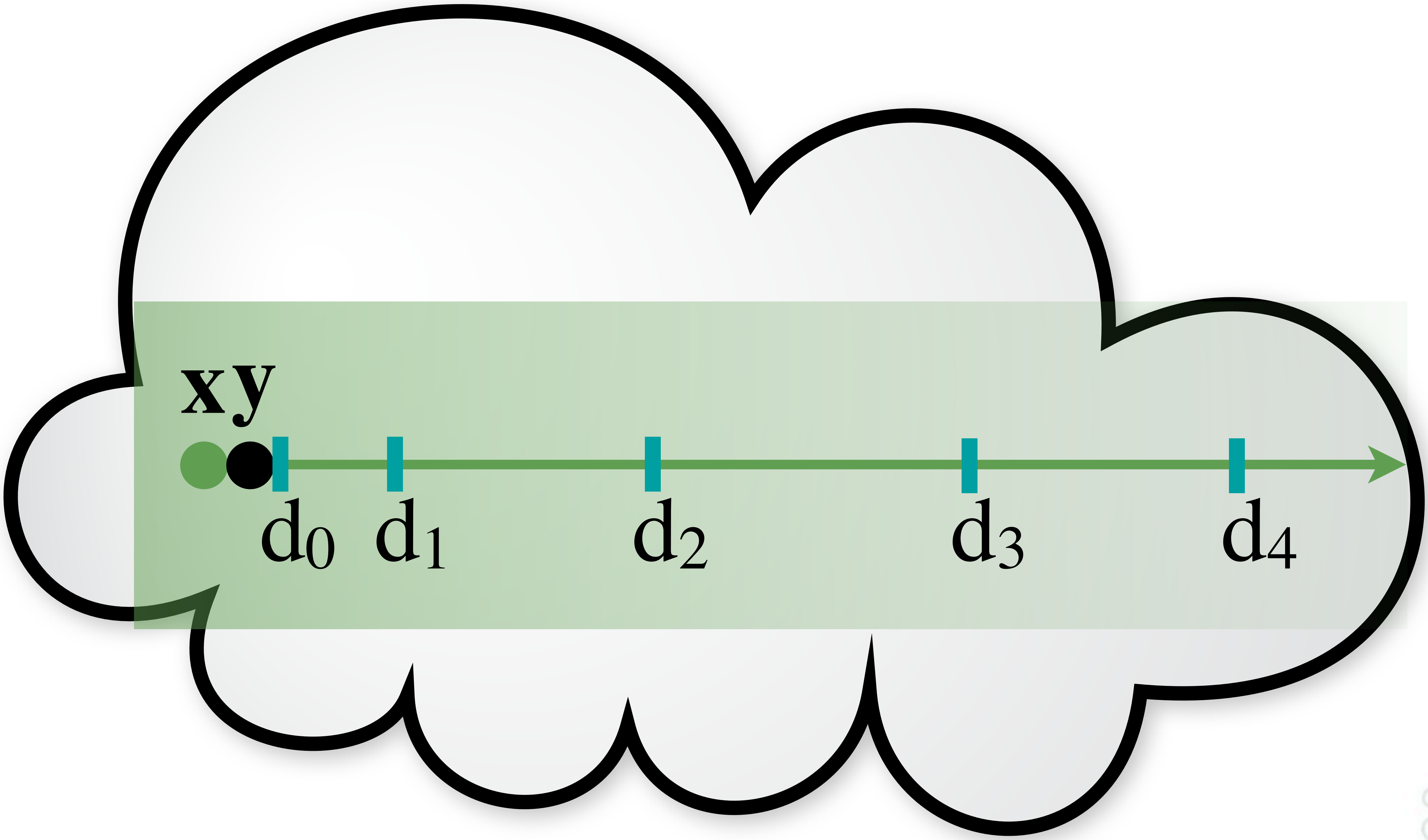
- Store n distances, re-evaluate: $T_r(\mathbf{x}, \mathbf{y}) \approx \frac{2}{5}$



- To arrive at our efficient solution, we make the observation that once we have computed several random distances for some location y, we can also re-evaluate the transmittance at arbitrary locations along the beam, by simply counting how many samples fell before and how many after our evaluation location y
- Hence, we can compute these propagation distances once for each beam, and cache them with the beams for evaluation during rendering

Transmittance using Free-flight Distance

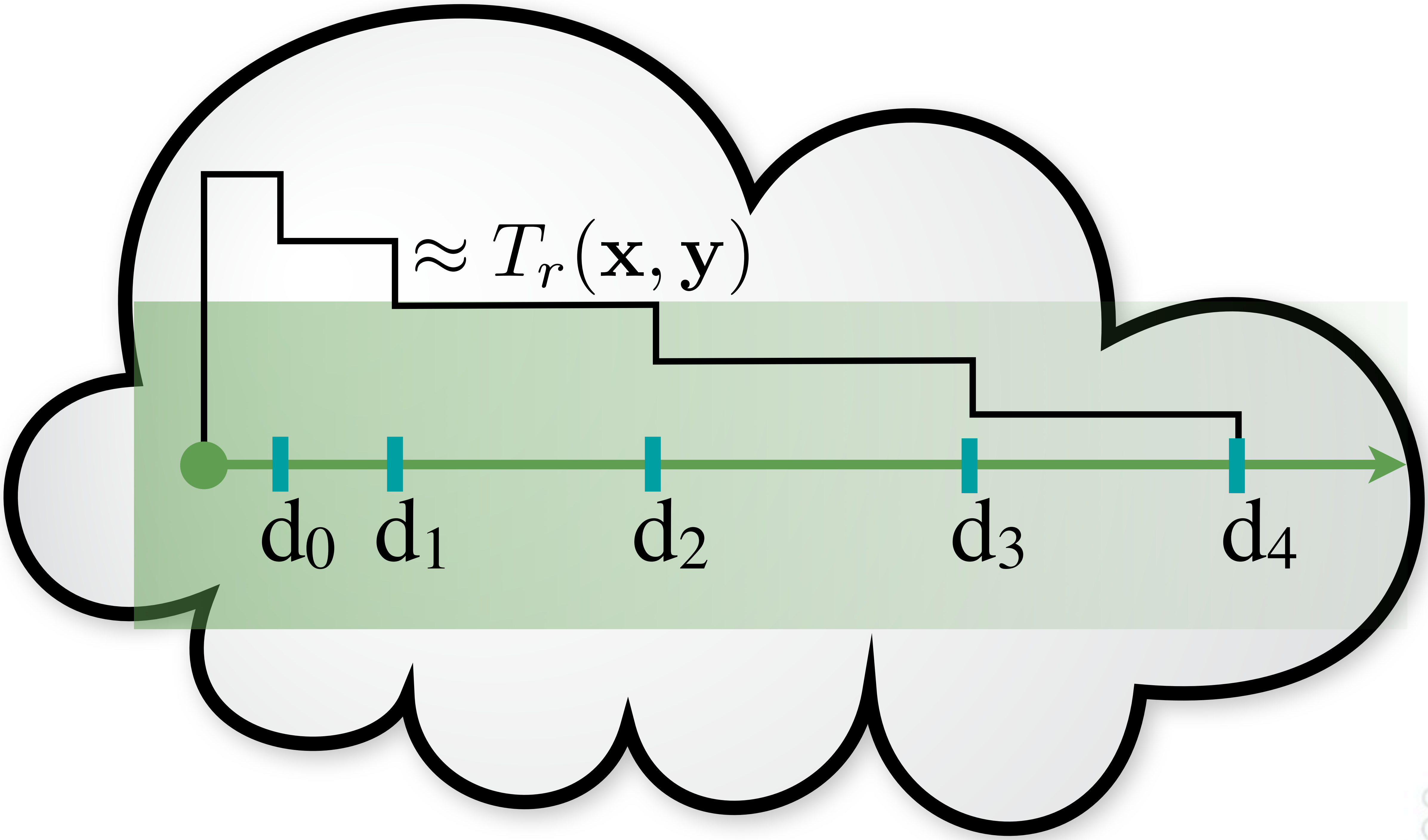
- Store n distances, re-evaluate: $T_r(\mathbf{x}, \mathbf{y}) \approx \frac{5}{5}$



- To arrive at our efficient solution, we make the observation that once we have computed several random distances for some location y, we can also re-evaluate the transmittance at arbitrary locations along the beam, by simply counting how many samples fell before and how many after our evaluation location y
- Hence, we can compute these propagation distances once for each beam, and cache them with the beams for evaluation during rendering

Transmittance using Free-flight Distance

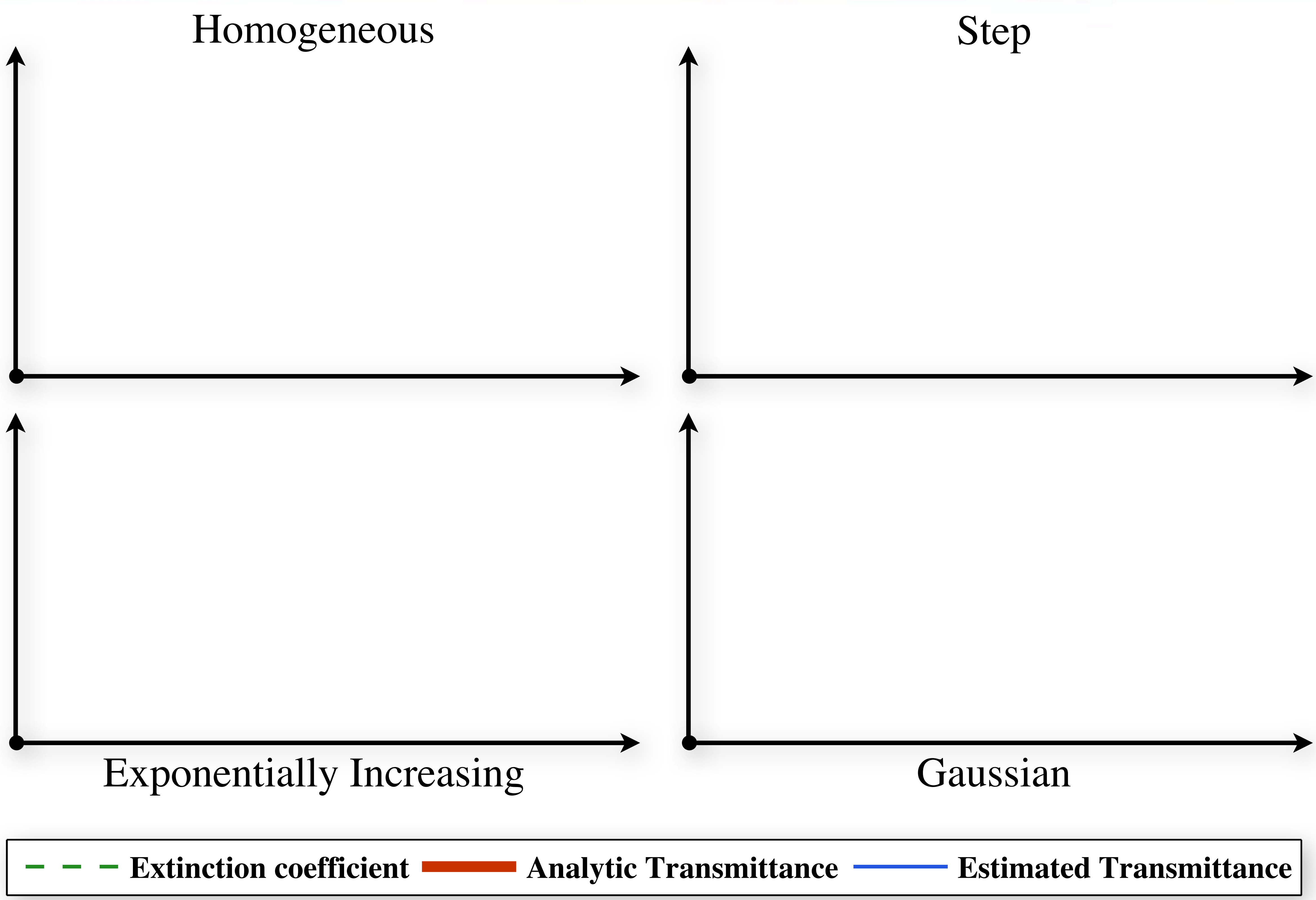
- Step-approximation to transmittance



Thursday, 6 September 12

- This effectively stores a piece-wise constant step-approximation to the transmittance with each beam
- This has the convenience of the ray marching solution, since we can compactly cache it and quickly re-evaluate, but additionally it is unbiased.

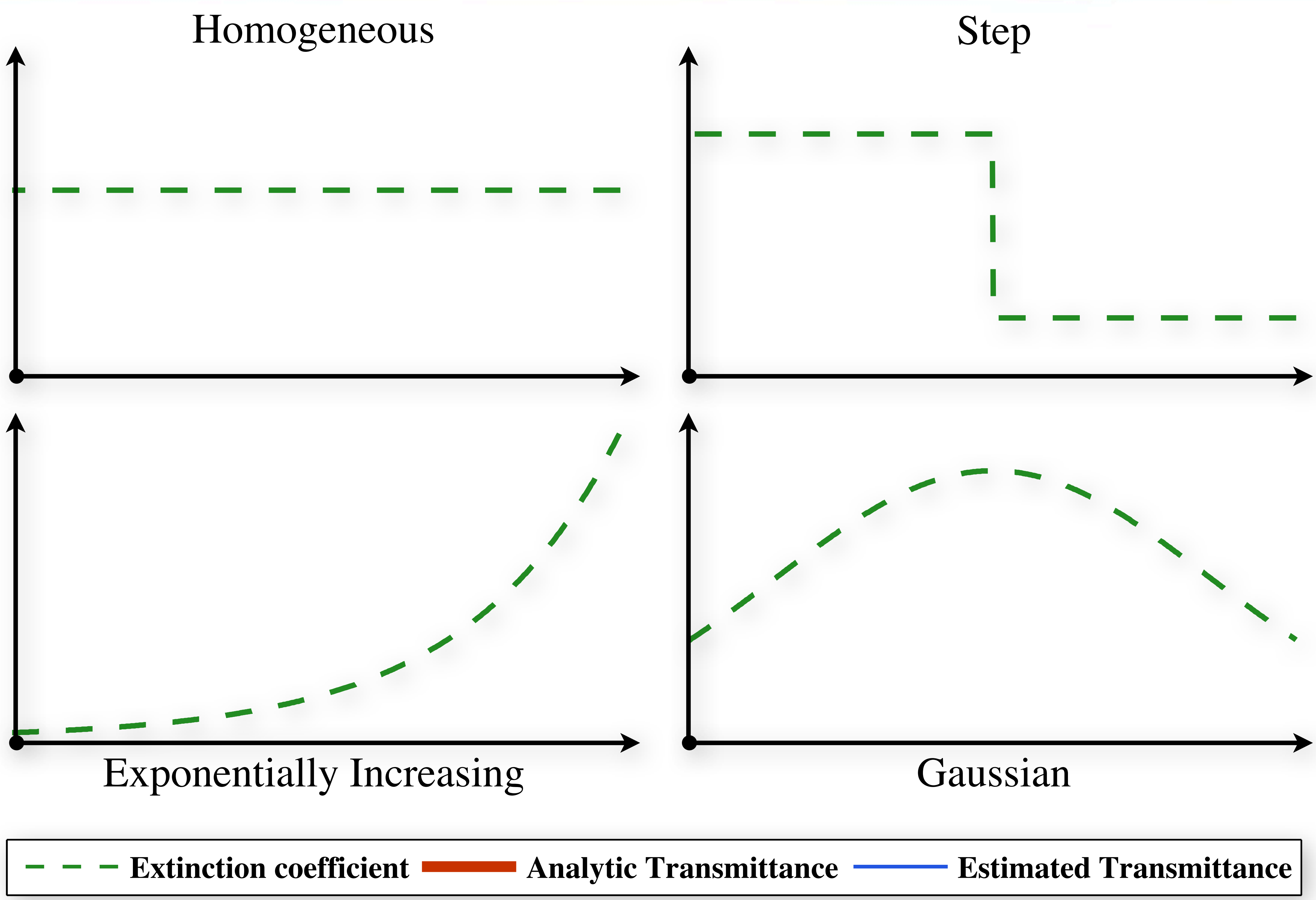
Transmittance Validation



Thursday, 6 September 12

- To confirm that this [click]
- we used this technique to compute transmittance for a number of media configurations [click]
- where the transmittance can be computed analytically

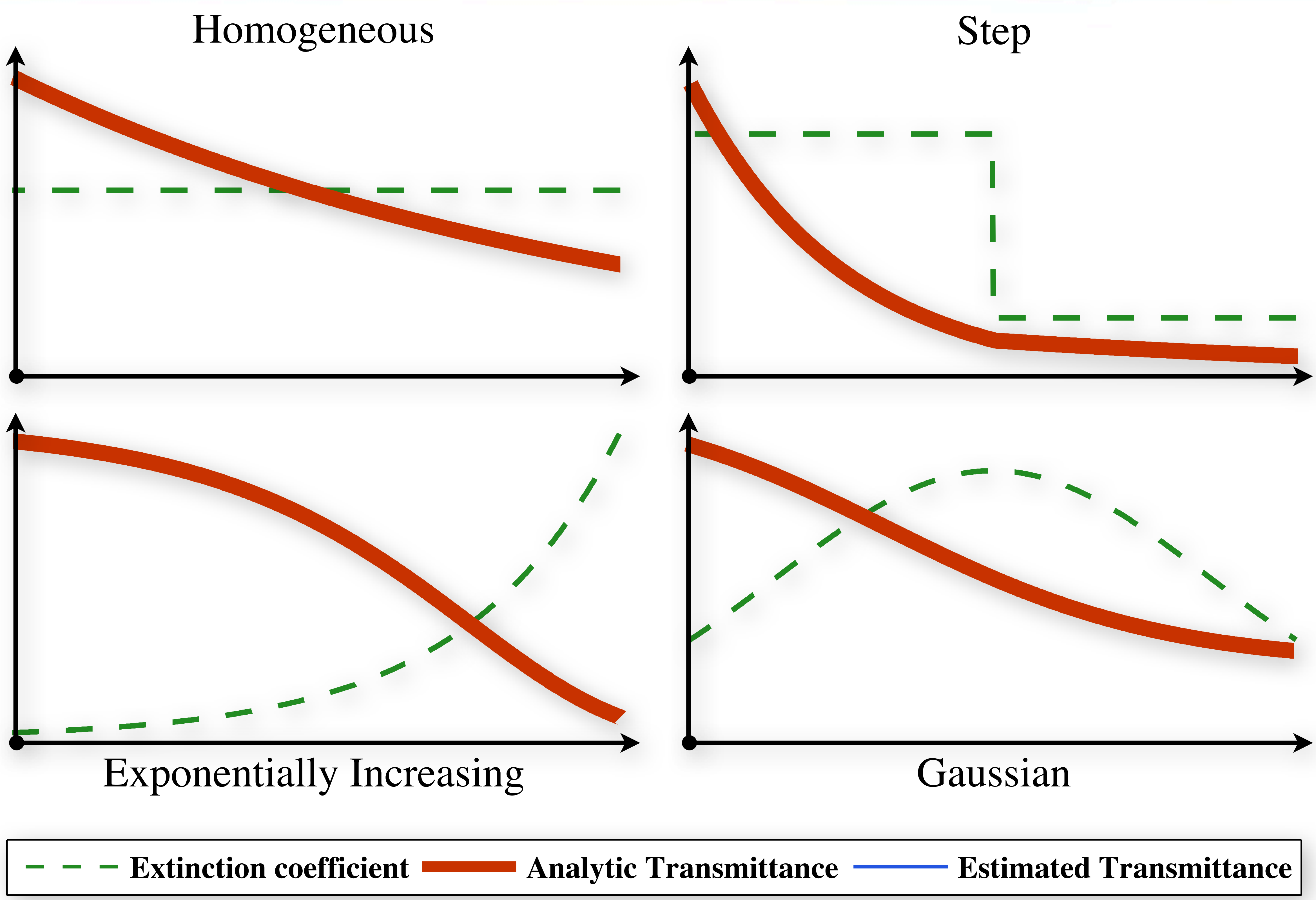
Transmittance Validation



Thursday, 6 September 12

- To confirm that this [click]
- we used this technique to compute transmittance for a number of media configurations [click]
- where the transmittance can be computed analytically

Transmittance Validation

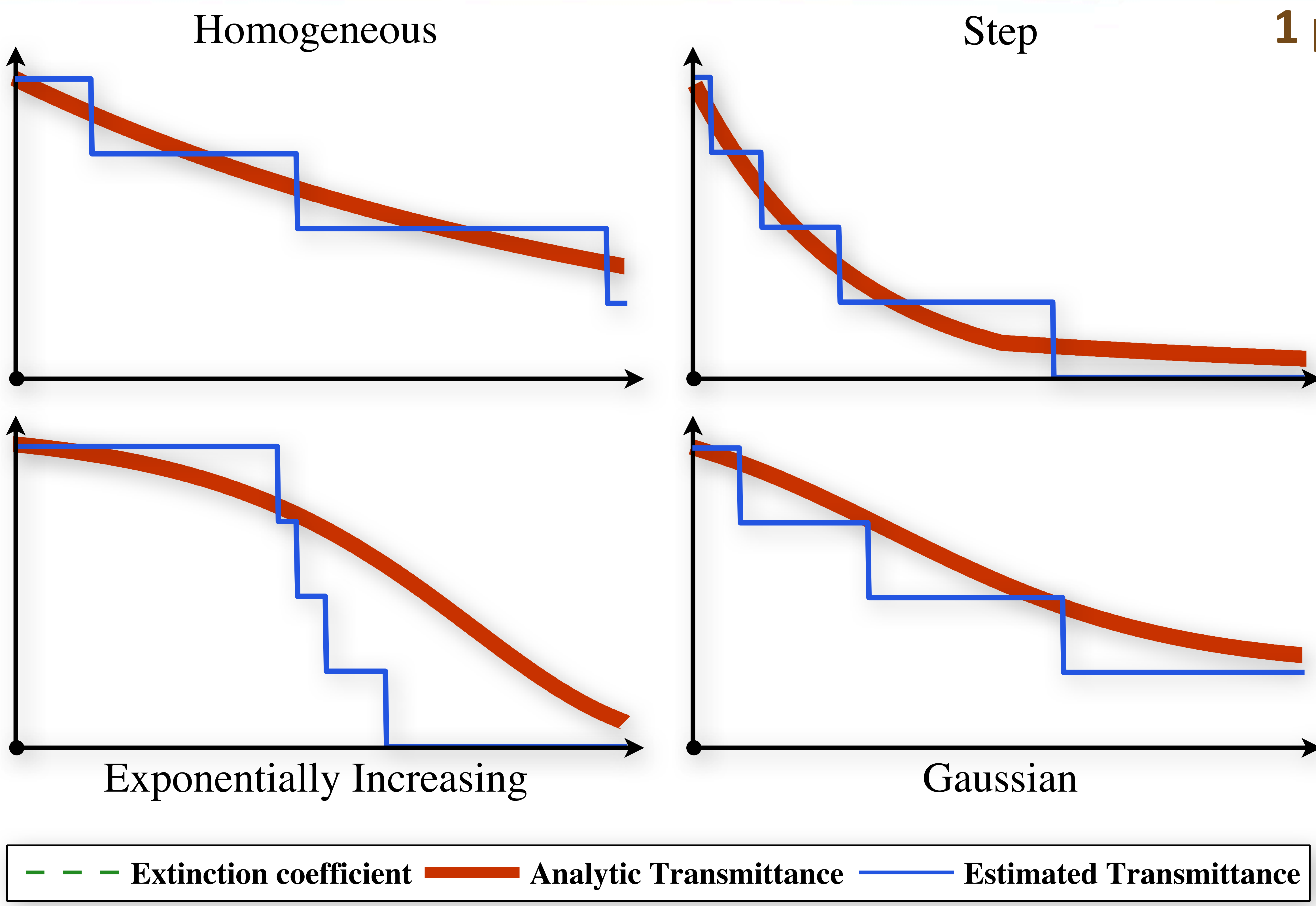


Thursday, 6 September 12

- To confirm that this [click]
- we used this technique to compute transmittance for a number of media configurations [click]
- where the transmittance can be computed analytically

Transmittance Validation

1 pass

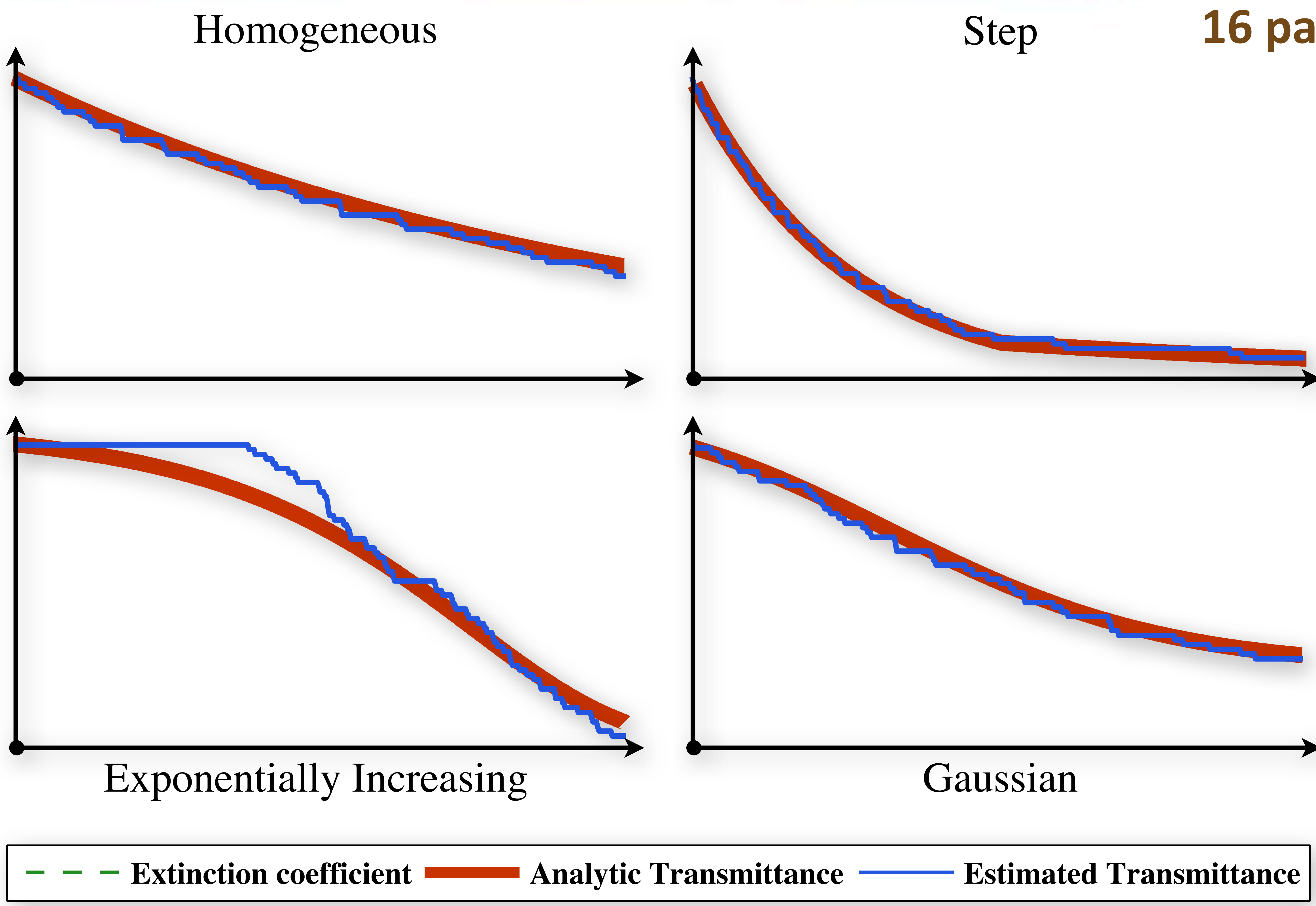


Thursday, 6 September 12

- If we use just 4 propagation distances, we get a pretty course approximation of the transmittance

Transmittance Validation

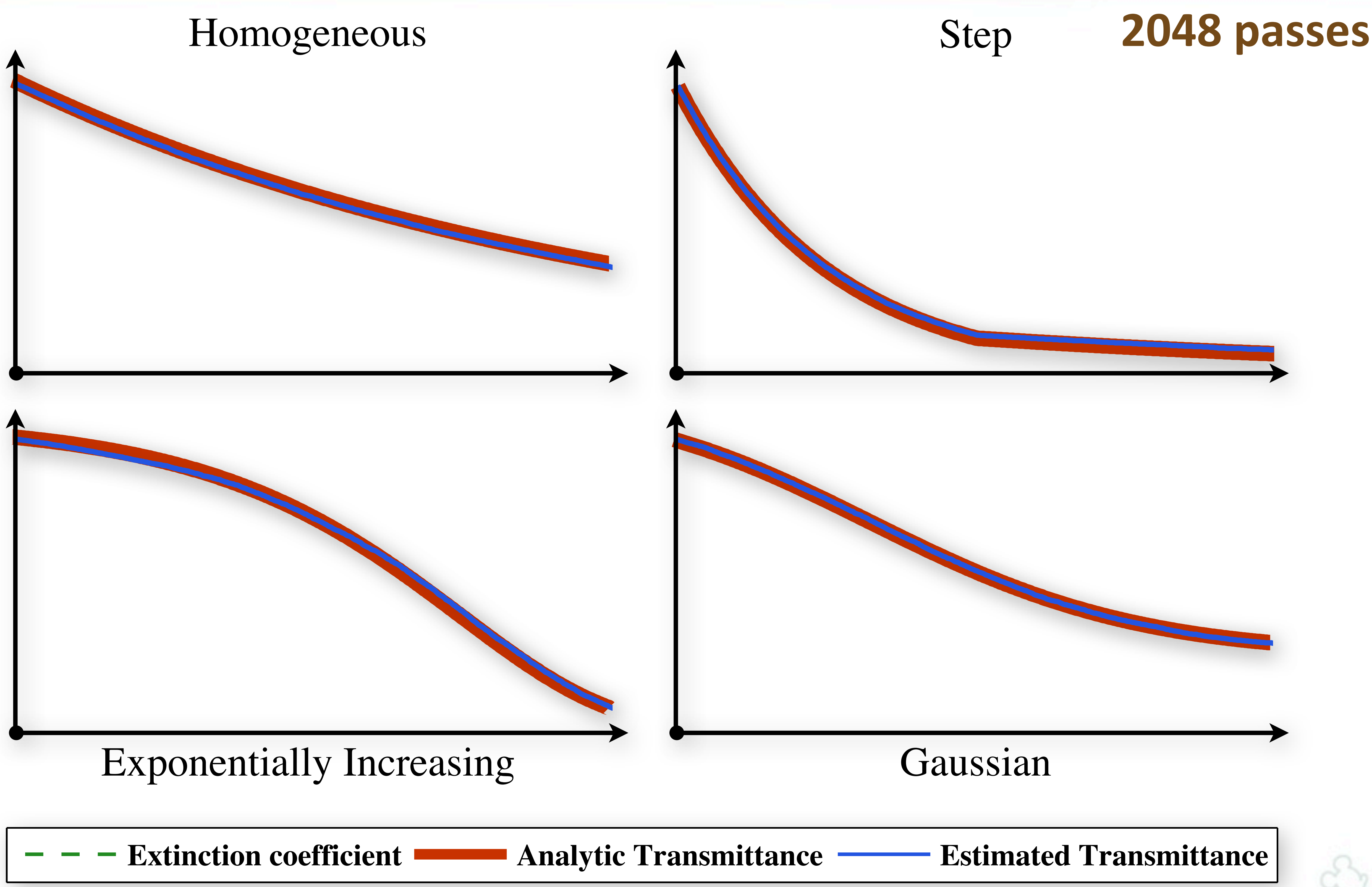
16 passes



Thursday, 6 September 12

- However, since the method is unbiased, by just performing this independently for each pass, we can reduce the error of the approximation

Transmittance Validation



Thursday, 6 September 12

- which is guaranteed to converge to the correct solution with more passes

Algorithm (Heterogeneous)

Thursday, 6 September 12

- We use this approach to handle heterogeneous media
- This requires modifying our base algorithm in two simple ways: [click]
- During photon tracing, we compute and store several propagation distances with each beam [click]
- Similarly, before we start rendering, we compute and store several propagation distances for each pixel in the image
- We repeat these extra steps with different random numbers in each pass to ensure convergence

Algorithm (Heterogeneous)

Step 1:

+ Compute/store n-step transmittance with each **beam**

- We use this approach to handle heterogeneous media
- This requires modifying our base algorithm in two simple ways: [click]
- During photon tracing, we compute and store several propagation distances with each beam [click]
- Similarly, before we start rendering, we compute and store several propagation distances for each pixel in the image
- We repeat these extra steps with different random numbers in each pass to ensure convergence

Algorithm (Heterogeneous)

Step 1:

+ Compute/store n-step transmittance with each **beam**

Step 2:

+ Compute/store n-step transmittance for each **pixel**

- We use this approach to handle heterogeneous media
- This requires modifying our base algorithm in two simple ways: [click]
- During photon tracing, we compute and store several propagation distances with each beam [click]
- Similarly, before we start rendering, we compute and store several propagation distances for each pixel in the image
- We repeat these extra steps with different random numbers in each pass to ensure convergence

Results & Implementation

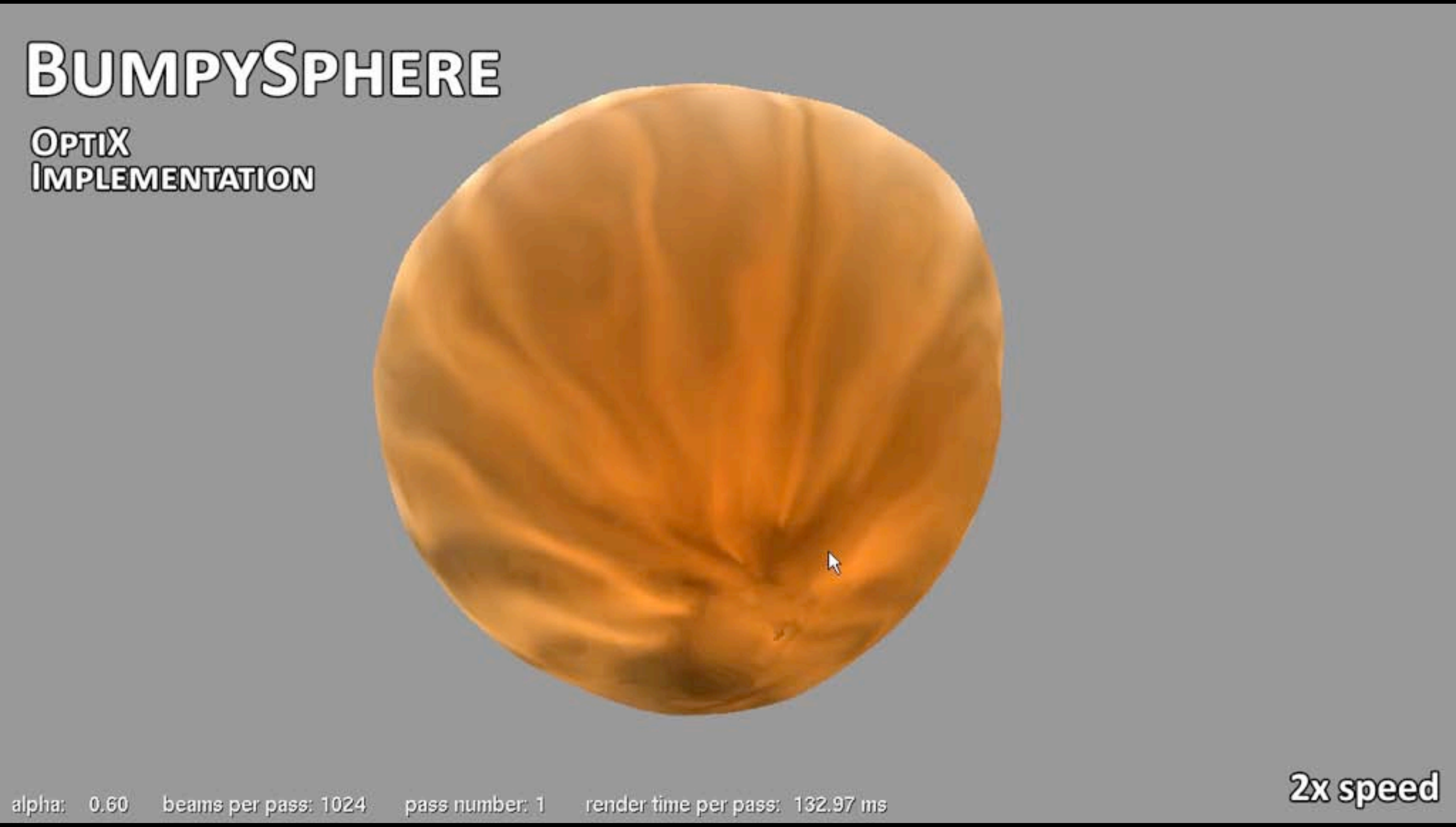
- 3 implementations:
  - GPU-only OptiX ray-tracer
  - GPU-only rasterization
  - General: Hybrid CPU/GPU

- To show that our approach can easily be applied to different computing platforms, we demonstrate our results using three different implementations.

Results & Implementation

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

- Firstly, we implemented the technique on the GPU using NVIDIA's OptiX ray-tracing framework



scene courtesy of Bruce Walter

Thursday, 6 September 12

- We ran our algorithm on the BumpySphere scene which we used in our original photon beams paper.
- This is a deformed refracting sphere filled with a homogeneous medium, such as amber for instance
- With a thousand beams per pass the scene renders interactively, and quickly converges to a crisp solution in a matter of seconds
- Here the user can manipulating the light source direction or change the camera's view and the algorithm quickly converges

Results & Implementation

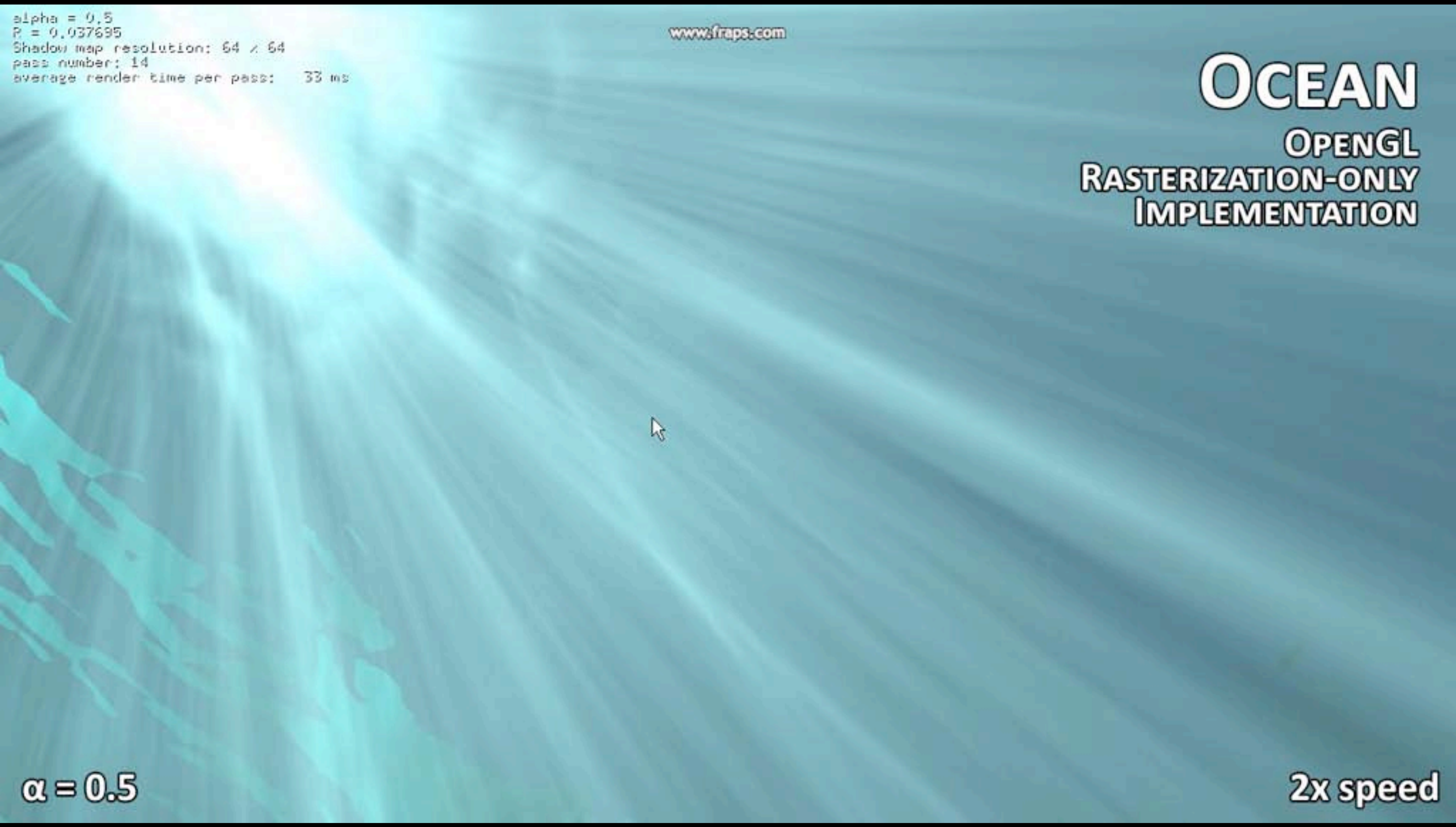
- 3 implementations:
  - GPU-only OptiX **ray-tracer**
  - GPU-only **rasterization**
  - General: Hybrid CPU/GPU

- In a ray-tracing framework, we need to intersect all camera rays with all beams
- This is easily the most expensive part of the algorithm

Results & Implementation

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

- In a ray-tracing framework, we need to intersect all camera rays with all beams
- This is easily the most expensive part of the algorithm
- To make this more efficient, we can render the directly-visible beams as axial-billboards using rasterization on the GPU.



- In this ocean scene, beams are generated, refracted at the ocean surface, and rasterized entirely on the GPU
- If we limit the number of beams per pass we can easily scale from real-time results, to interactive results, all the way to reference quality results if we let the algorithm converge over the course of a few seconds.

Results & Implementation

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

- Our most complex results are rendered in a general implementation which uses a combination of the CPU and GPU.
- We use the rasterization optimization I just mentioned to offload directly-visible media computation onto the GPU, while handling secondary reflections and refractions off of surfaces using ray tracing on the CPU

SOCCER

scene courtesy of Xin Sun

512x512

Line-space Gathering

[Sun et al. 2010]

Thursday, 6 September 12

- Xin Sun kindly shared their Soccer scene with us, rendered here using their line-space gathering algorithm
- For comparison, we also ran our algorithm on this scene, and additionally simulate multiple scattering, which line-space gather does not support.

SOCCER

scene courtesy of Xin Sun

512x512

Line-space Gathering

[Sun et al. 2010]

Our Method

16 passes
7.5 seconds CPU+GPU

- We can obtain fast preview quality results from 16 passes, after about 7 and a half seconds

SOCCER

scene courtesy of Xin Sun

512x512

Line-space Gathering

Our Method

[Sun et al. 2010]
73 min (CPU); 6.5 min (GPU)

512 passes
61 seconds CPU+GPU

✳ evaluated on similar, but not identical systems

- After 512 passes and about a minute of computation, our hybrid CPU+GPU renderer produces noise-free results
- In comparison, the performance reported by the line-space gathering paper is 73 minutes on the CPU or 6.5 minutes on the GPU.

CARS

1280x720, Depth-of-Field

Pass 1

Homogeneous

Heterogeneous

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 2 | Average of Passes 1..2

Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 4 | Average of Passes 1..4

Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

Cars

1280x720, Depth-of-Field

Pass 8 | Average of Passes 1..8

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 16 | Average of Passes 1..16

Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 32 | Average of Passes 1..32

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 64 | Average of Passes 1..64

Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 128 | Average of Passes 1..128

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 256 | Average of Passes 1..256

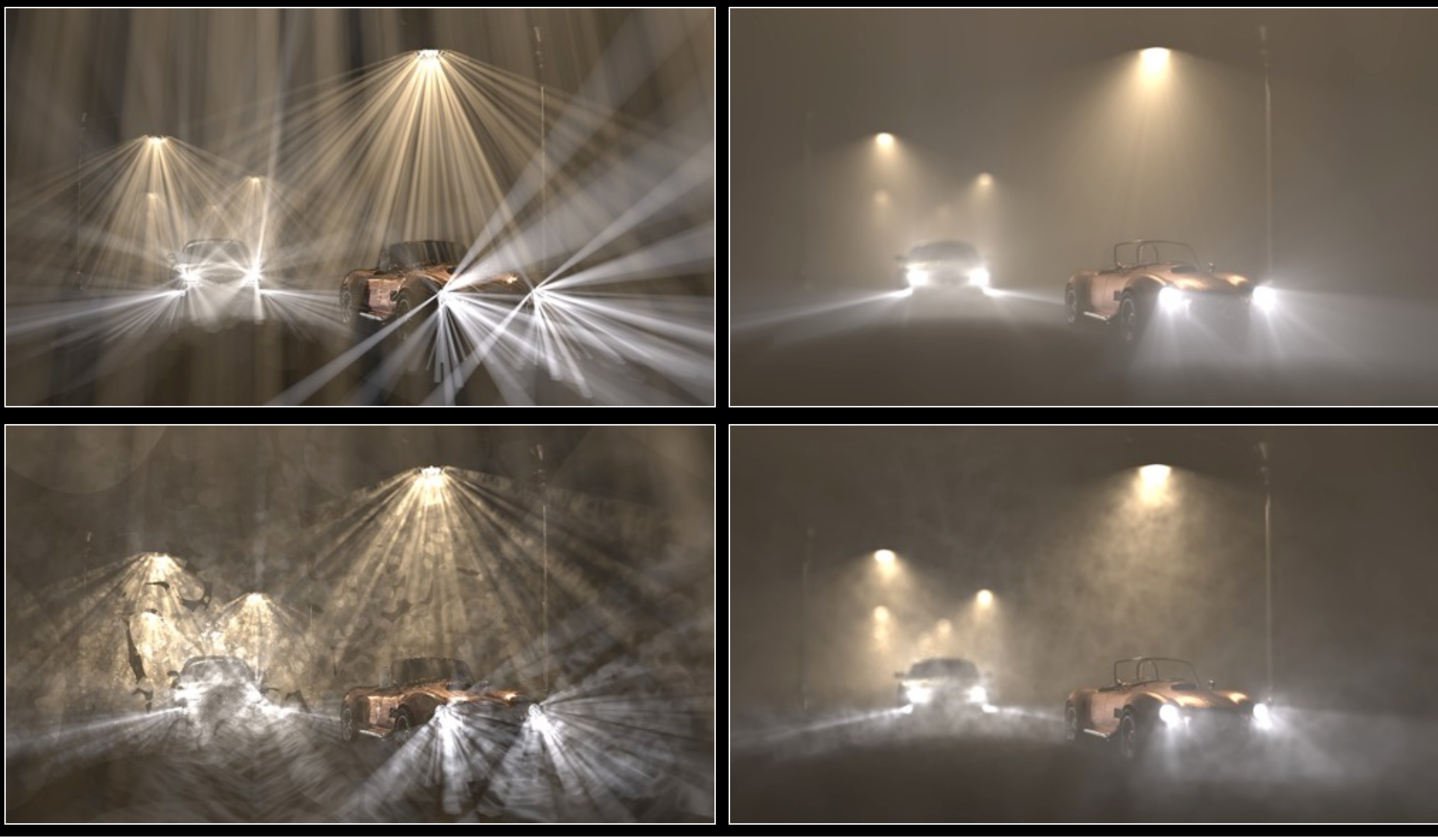
Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

Cars

1280x720, Depth-of-Field

Pass 512 | Average of Passes 1..512



Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

CARS

1280x720, Depth-of-Field

Pass 1024 | Average of Passes 1..1024

Thursday, 6 September 12

- In the last three results I show the heterogeneous version on the bottom, and homogeneous version at the top
- In all these scenes, the light sources (such as the street lights and the headlights of the cars) are encased in glass, which produce interesting and realistic angular variation in the lighting
- Note that the individual passes of heterogeneous media on the left show this swiss-cheese effect, which is due to our piecewise-step transmittance representation, but this quickly converges to a smooth result when averaged across passes.

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

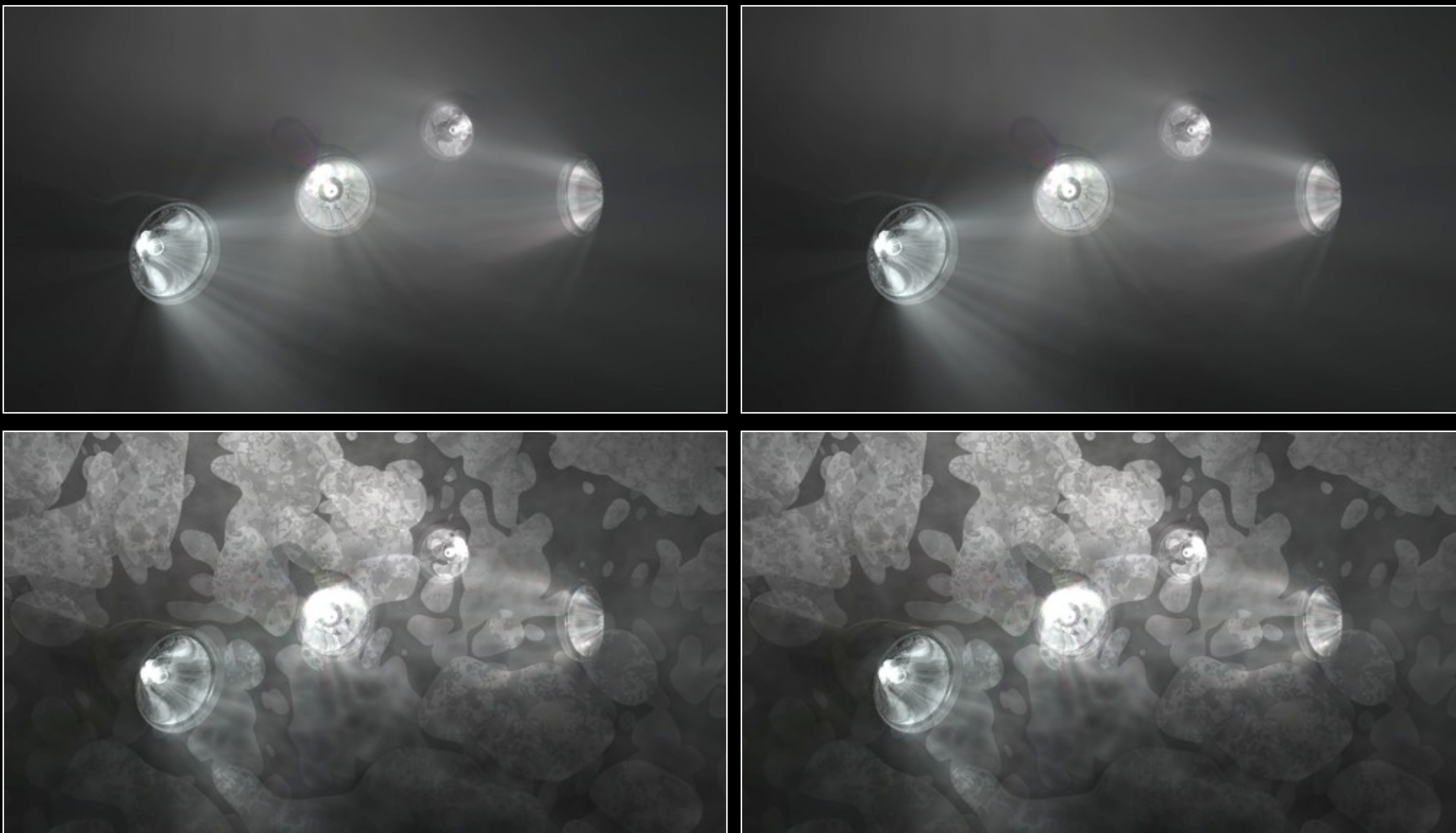
15.04M Photon Beams

16.8 minutes

FlashLights

1280x720, Depth-of-Field

Pass 1 | Average of Passes 1..1



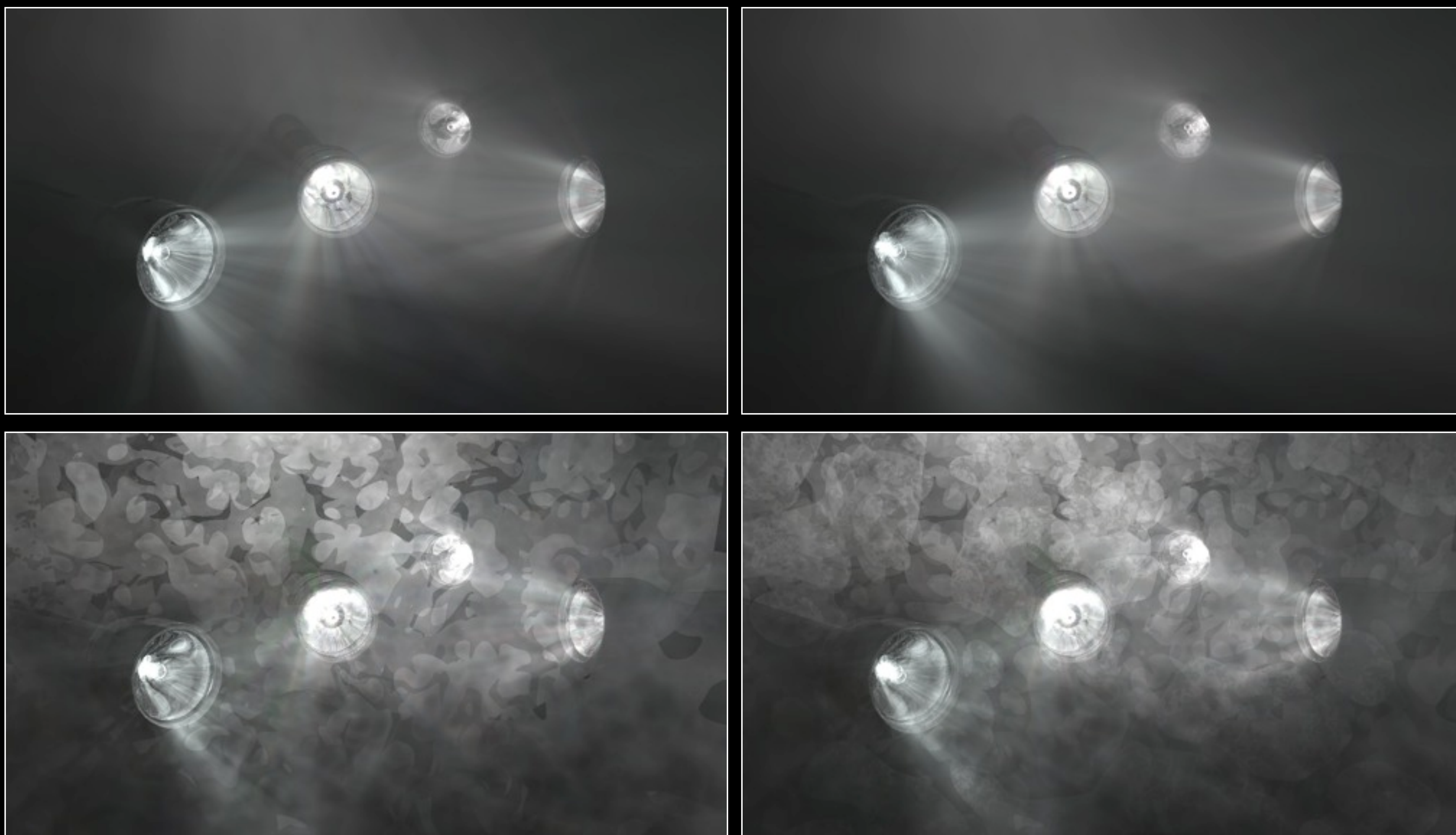
Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FLASHLIGHTS

1280x720, Depth-of-Field

Pass 2 | Average of Passes 1..2



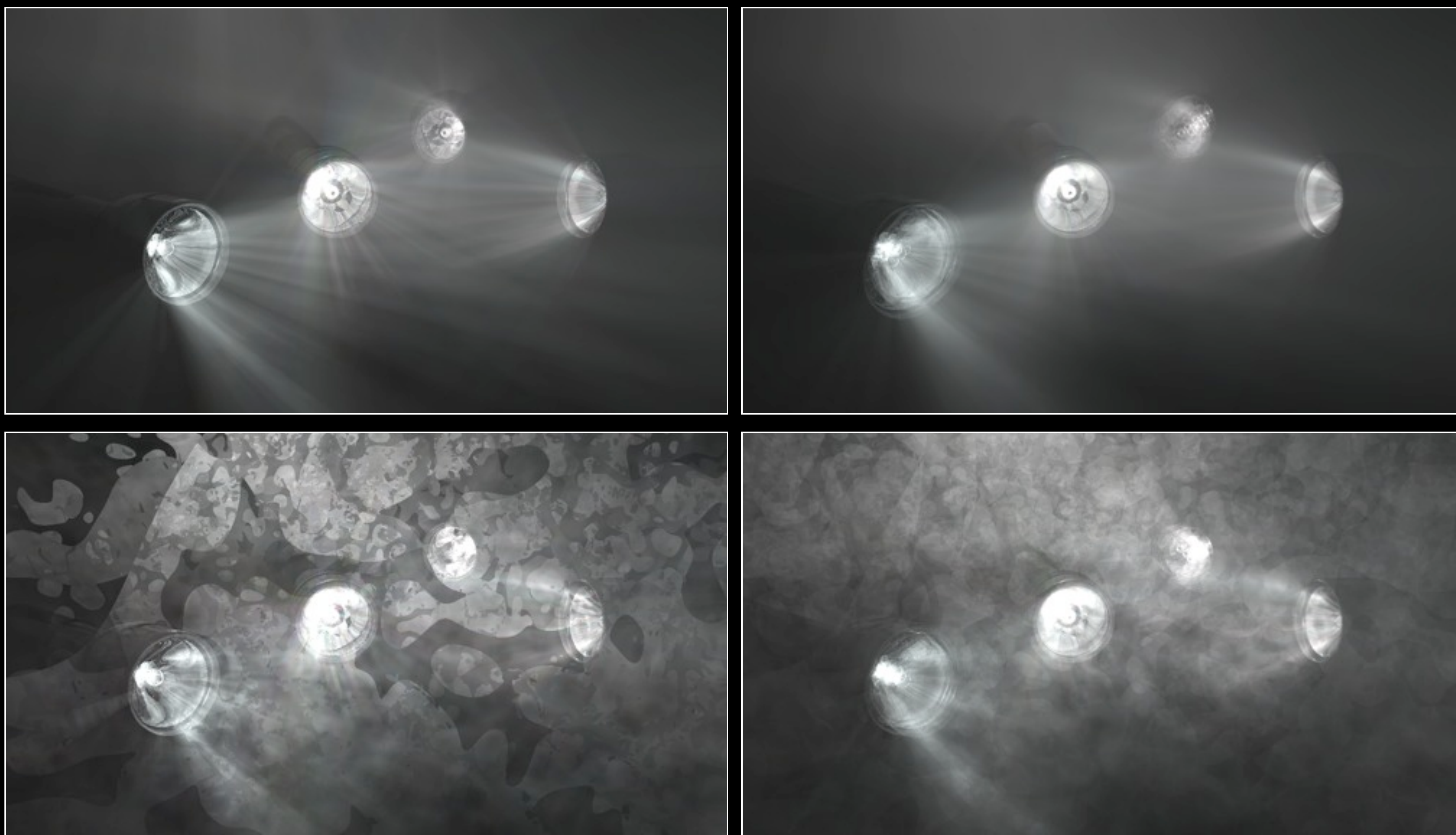
Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FLASHLIGHTS

1280x720, Depth-of-Field

Pass 4 | Average of Passes 1..4

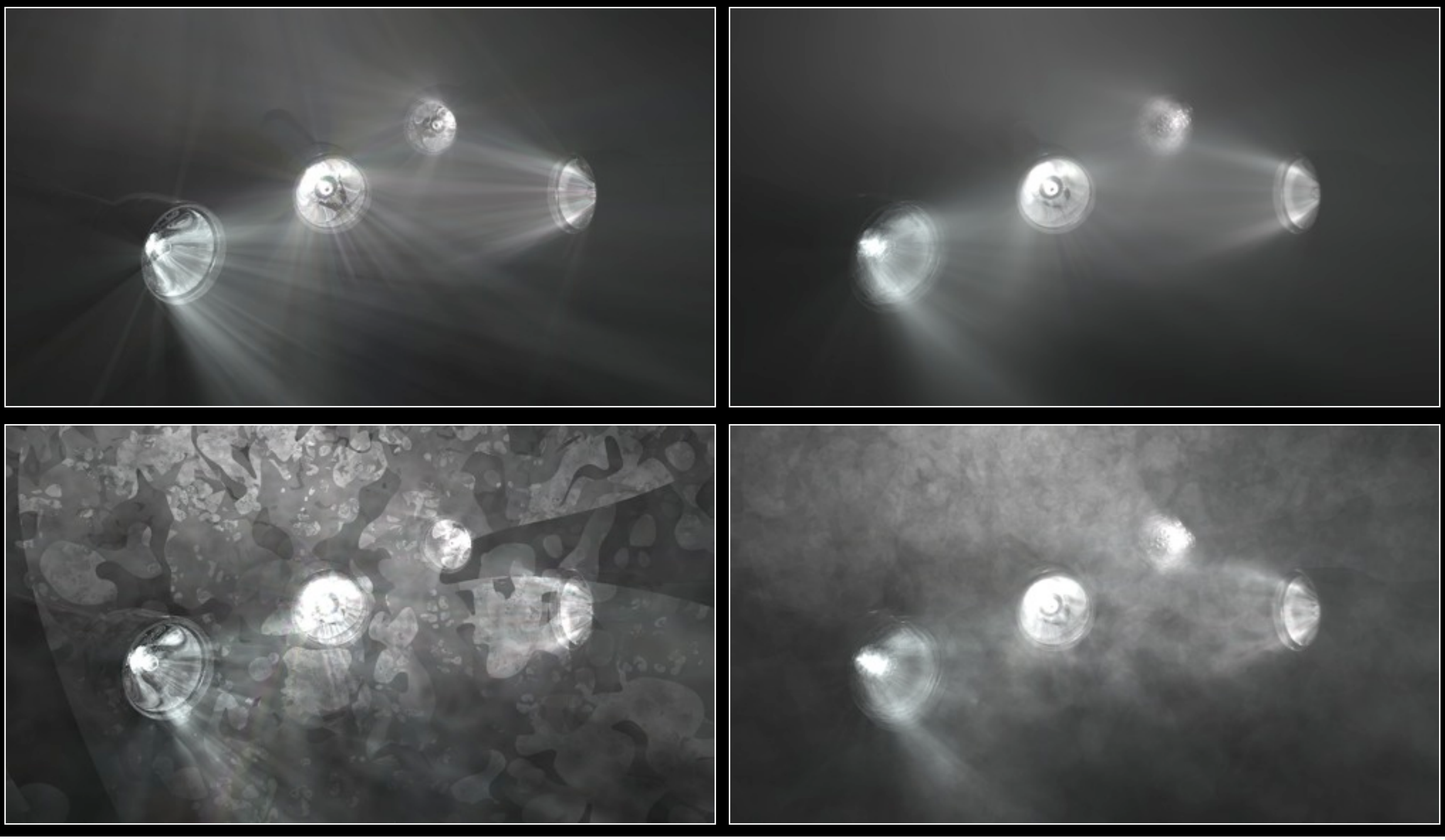


- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FlashLights

1280x720, Depth-of-Field

Pass 8 | Average of Passes 1..8



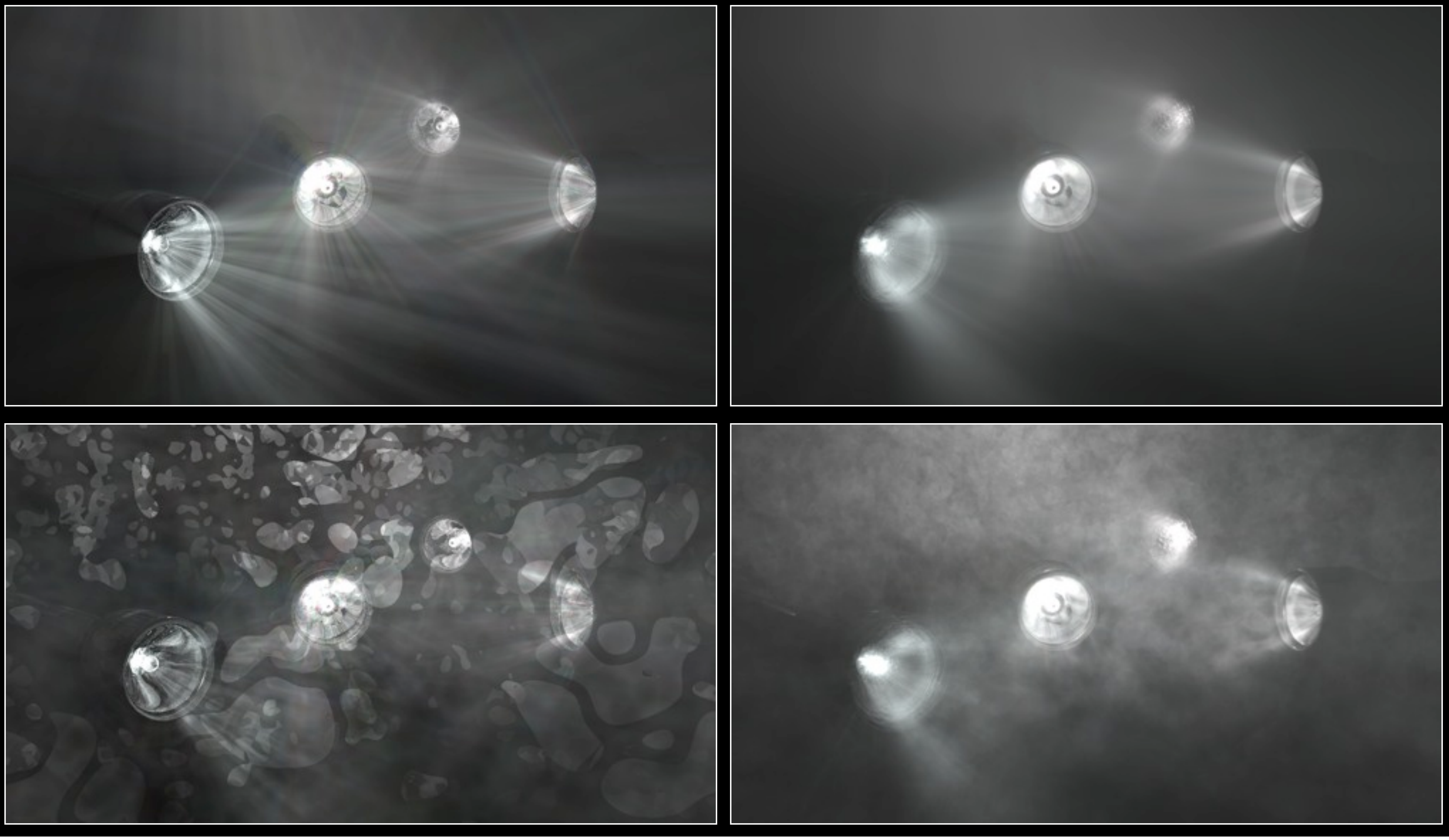
Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FLASHLIGHTS

1280x720, Depth-of-Field

Pass 16 | Average of Passes 1..16



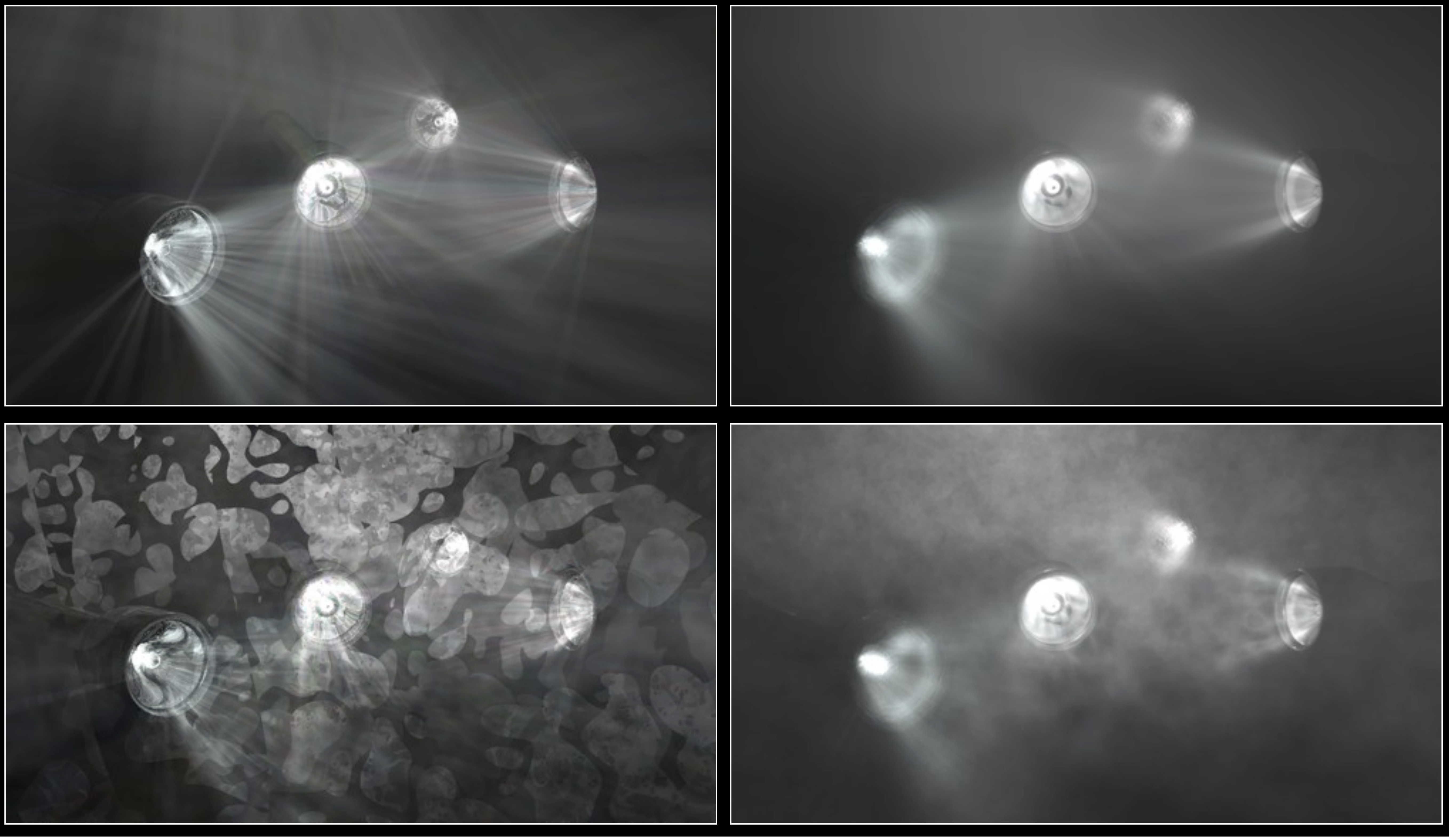
Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FlashLights

1280x720, Depth-of-Field

Pass 32 | Average of Passes 1..32



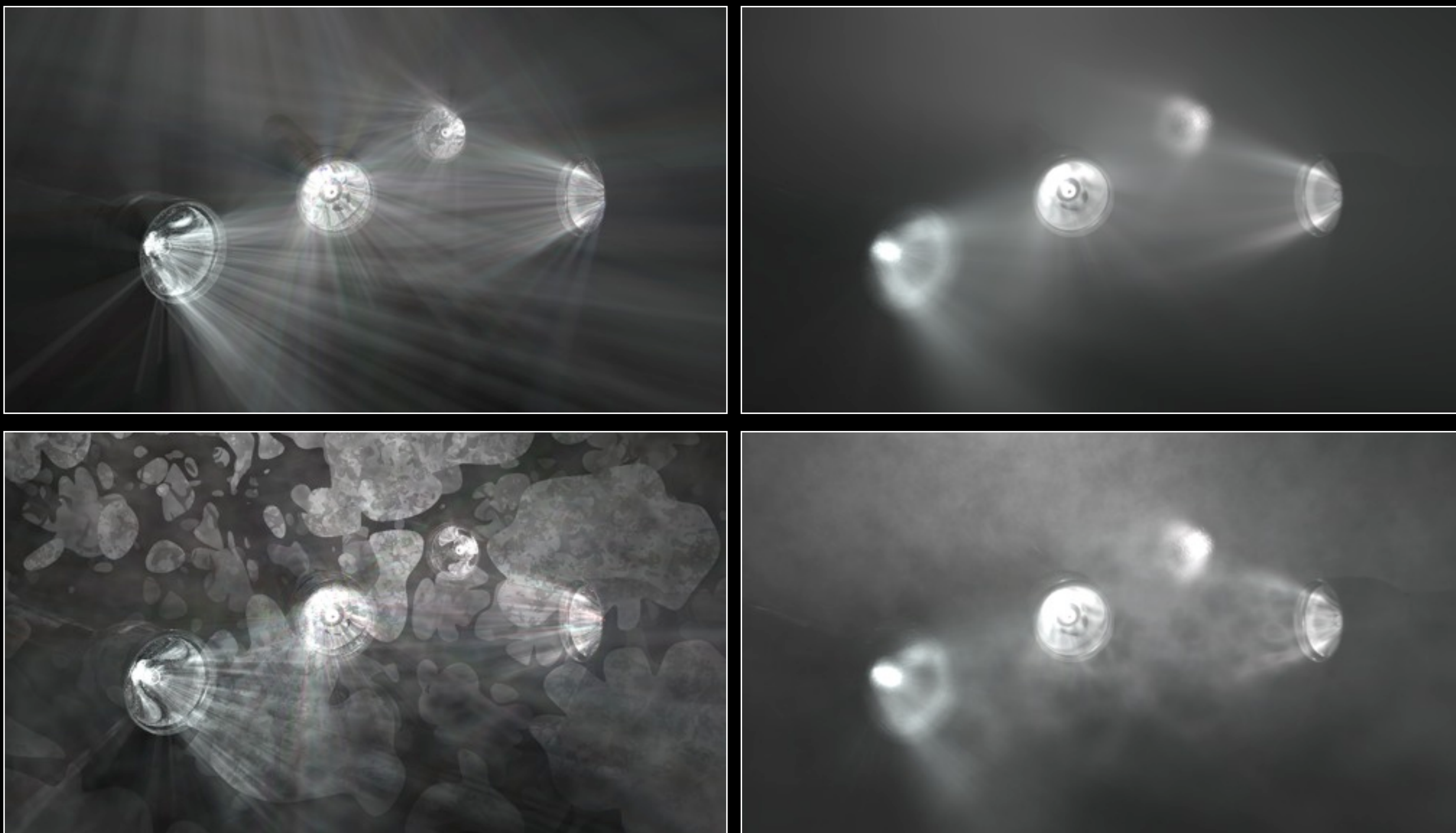
Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FLASHLIGHTS

1280x720, Depth-of-Field

Pass 64 | Average of Passes 1..64

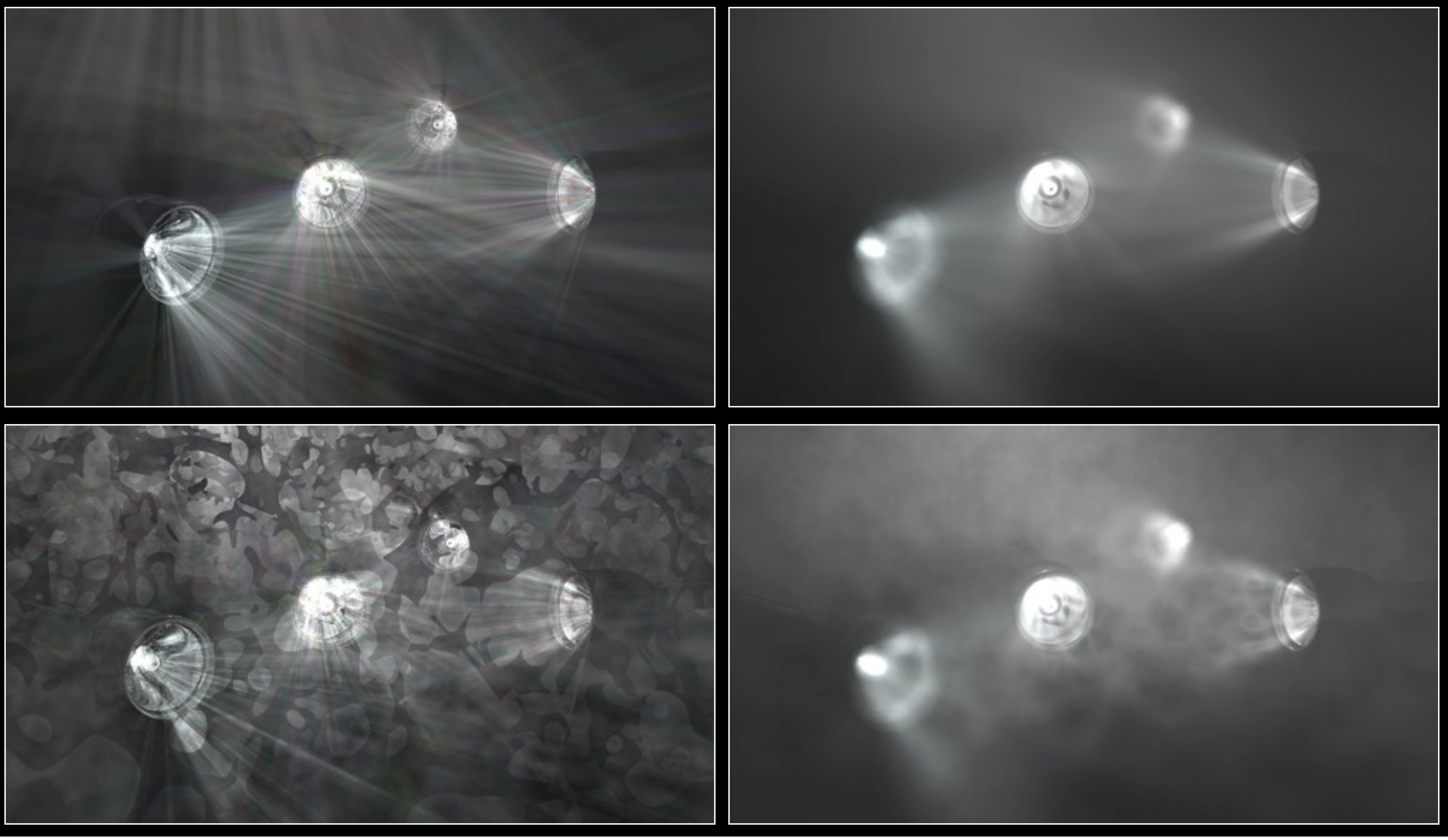


- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FLASHLIGHTS

1280x720, Depth-of-Field

Pass 128 | Average of Passes 1..128



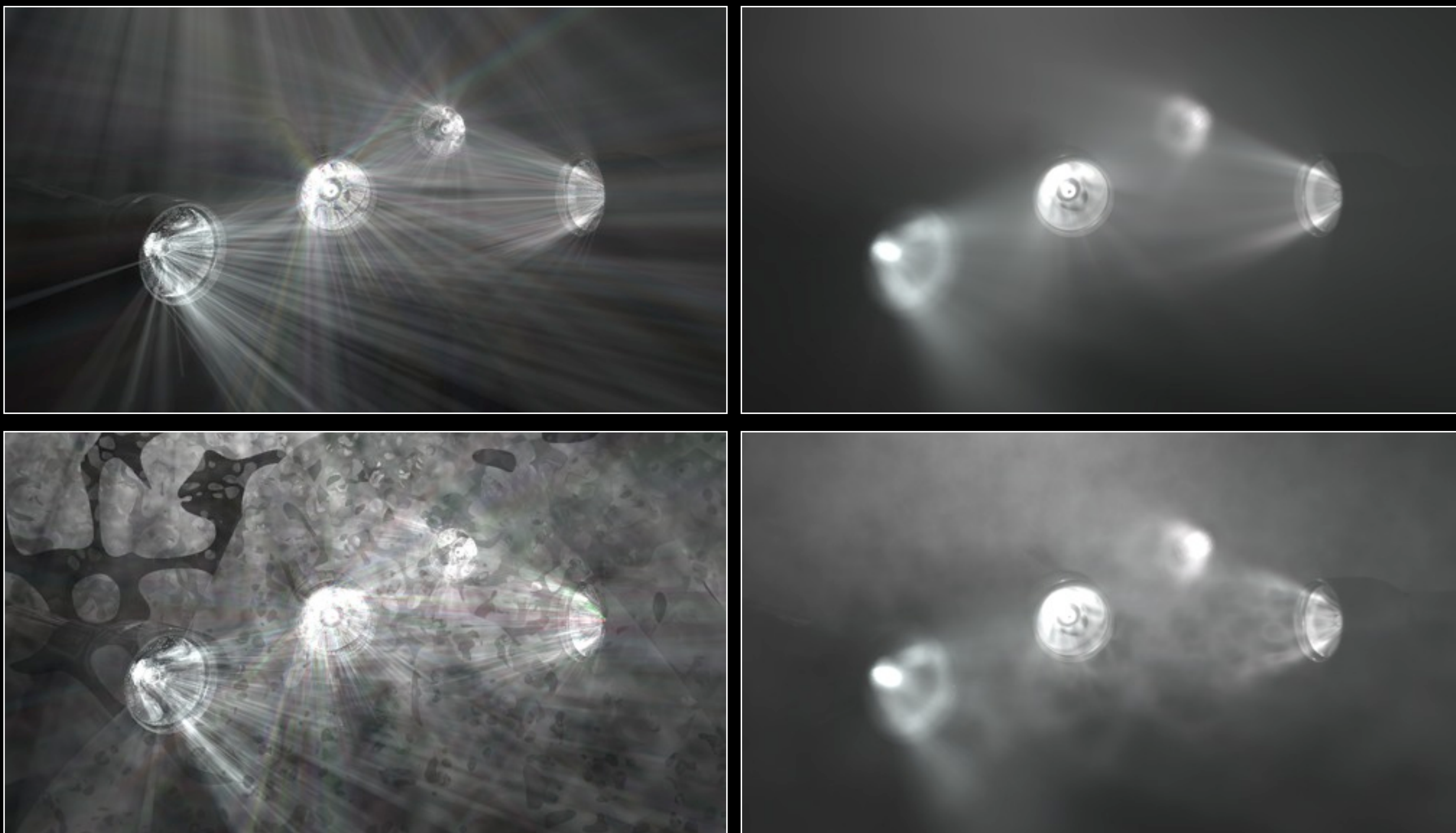
Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FlashLights

1280x720, Depth-of-Field

Pass 256 | Average of Passes 1..256

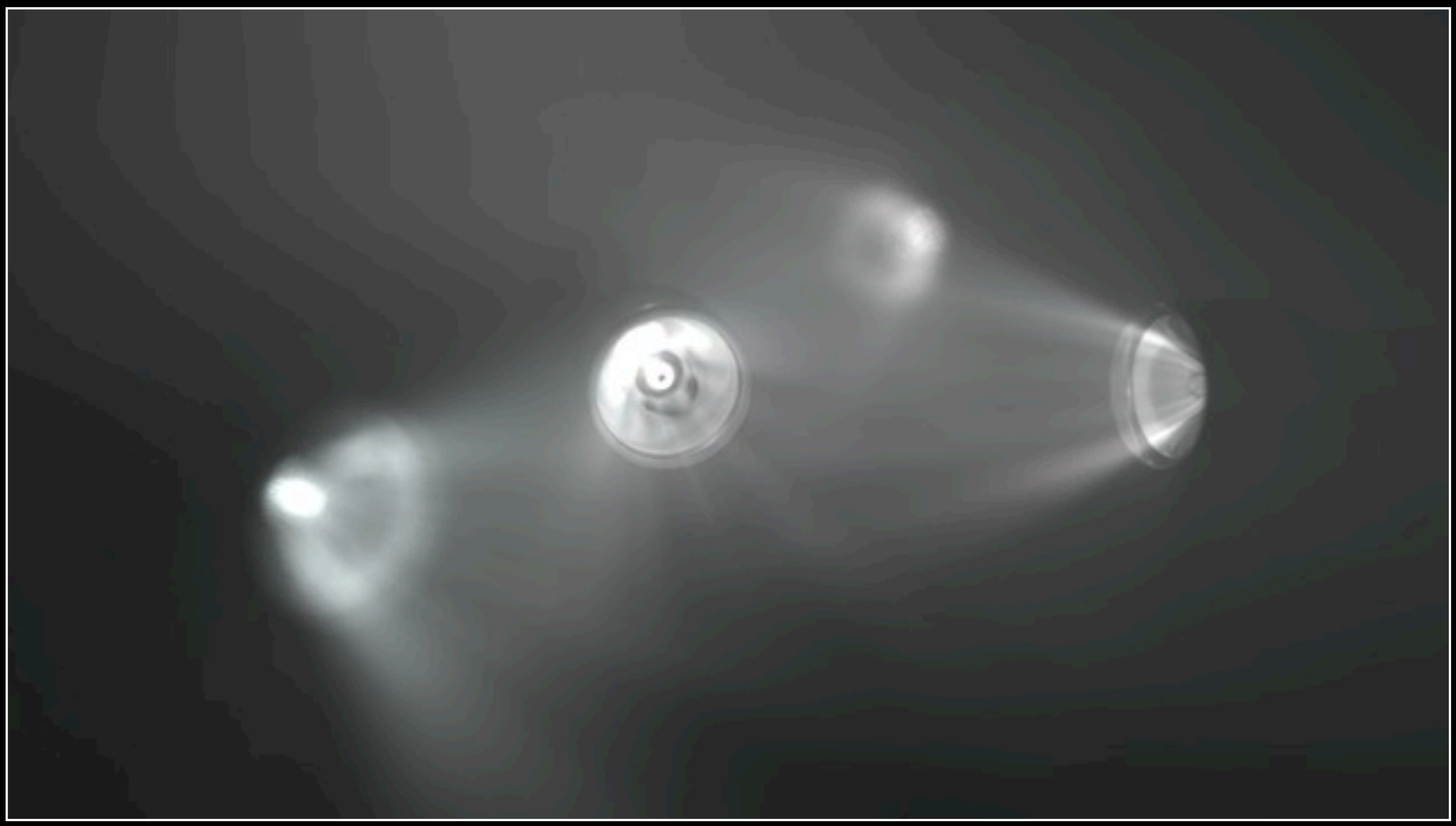


Thursday, 6 September 12

- In this flashlight scene, we again have light sources encased in glass and mirror elements
- Also note that we can trivially support depth-of-field since we are averaging over multiple passes

FLASHLIGHTS

1280x720, Depth-of-Field



Homogeneous

2.1M Photon Beams

8 minutes

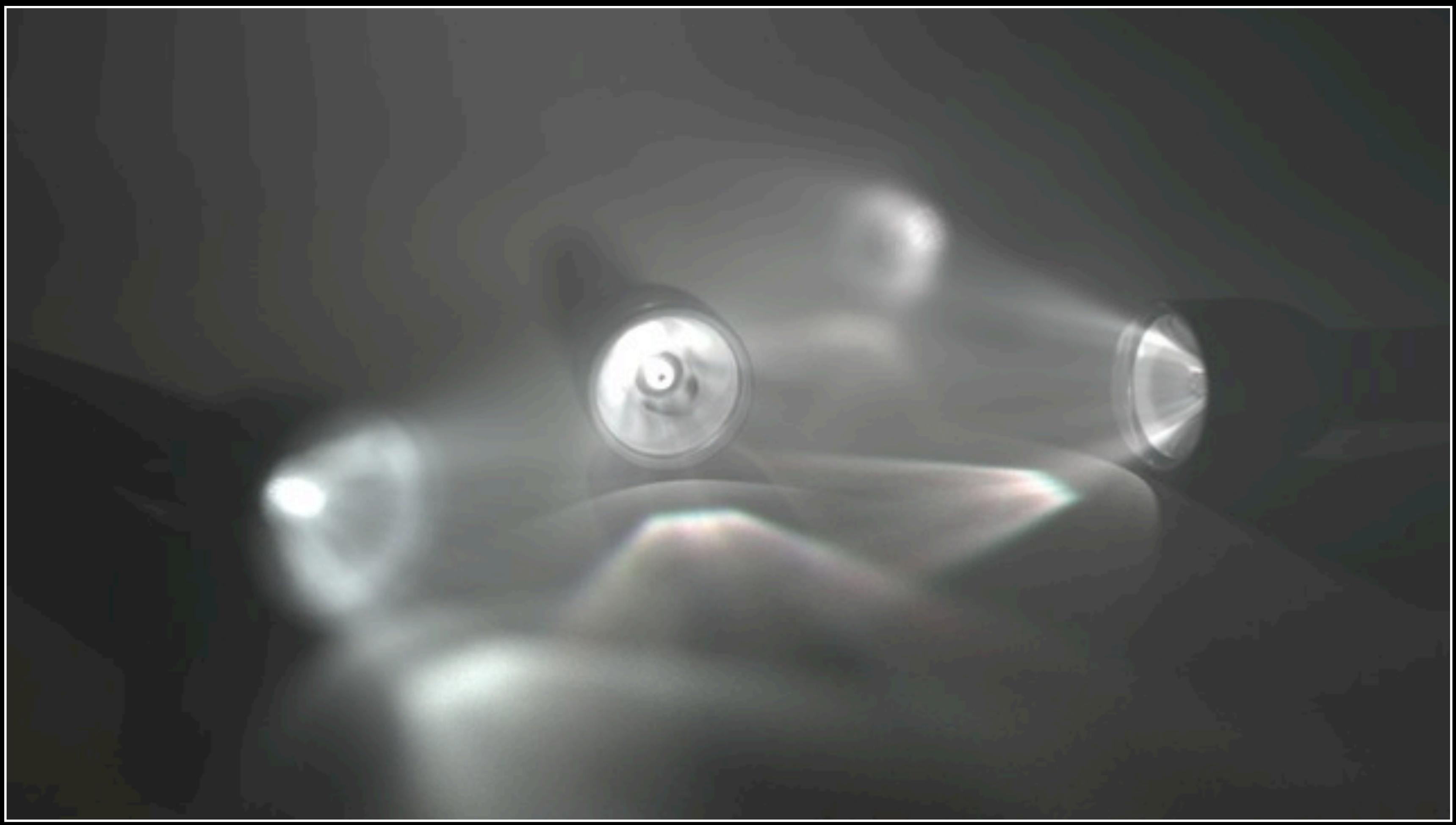
Heterogeneous

2.1M Photon Beams

10.8 minutes

FLASHLIGHTS

1280x720, Depth-of-Field



Homogeneous

2.1M Photon Beams

8 minutes

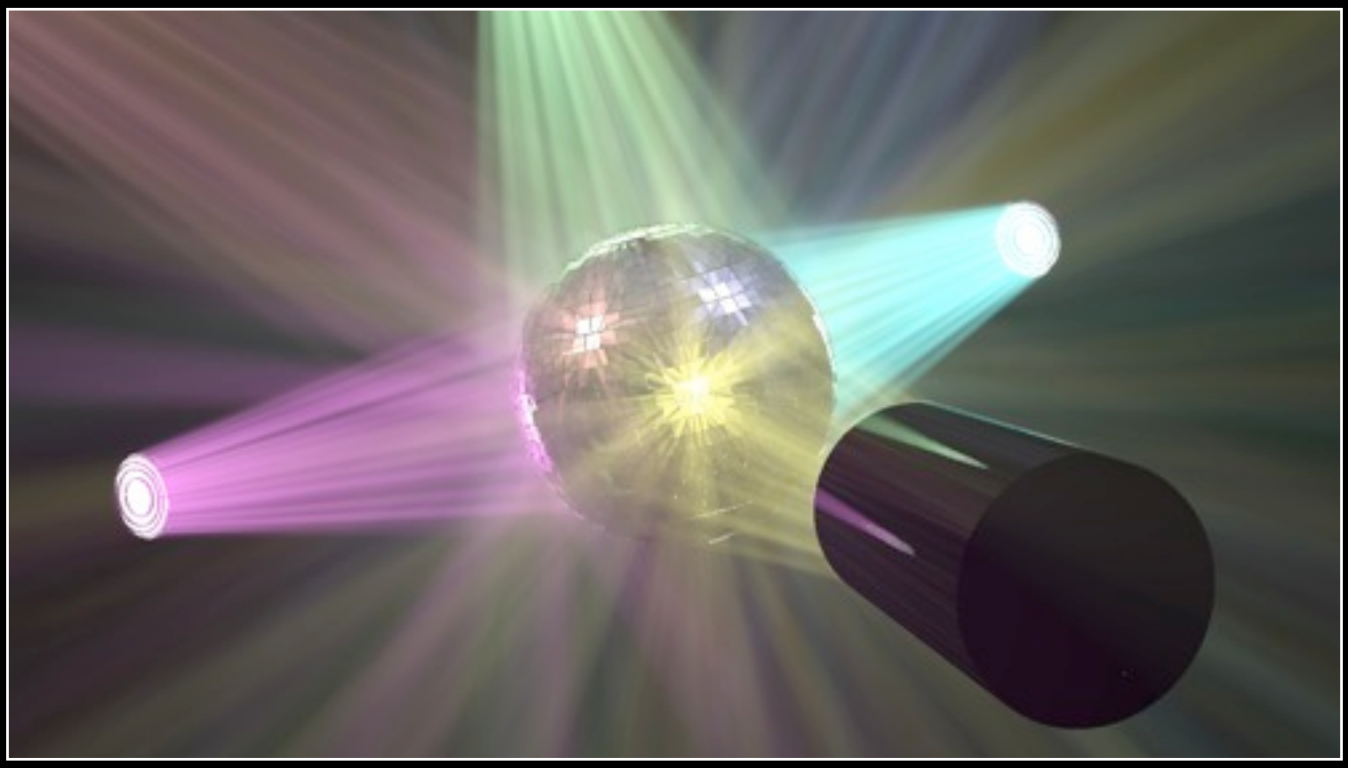
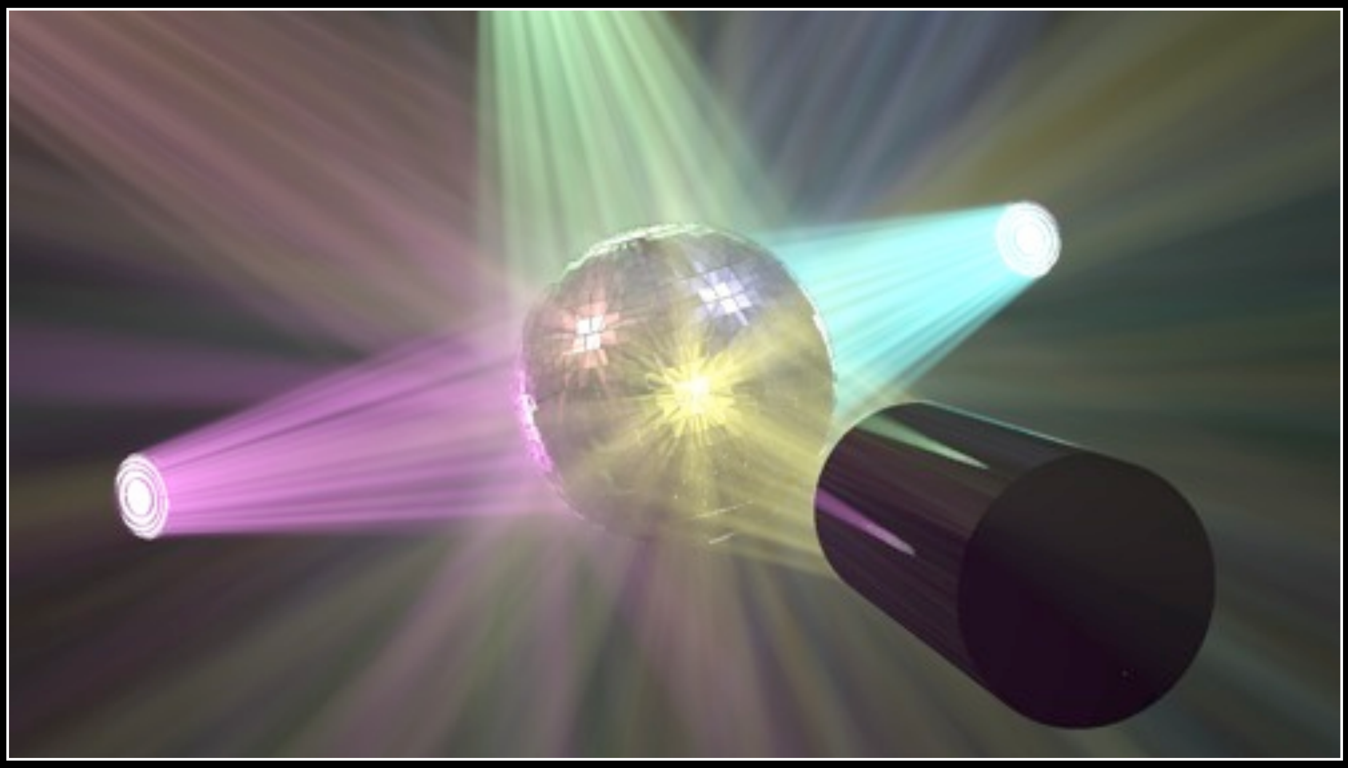
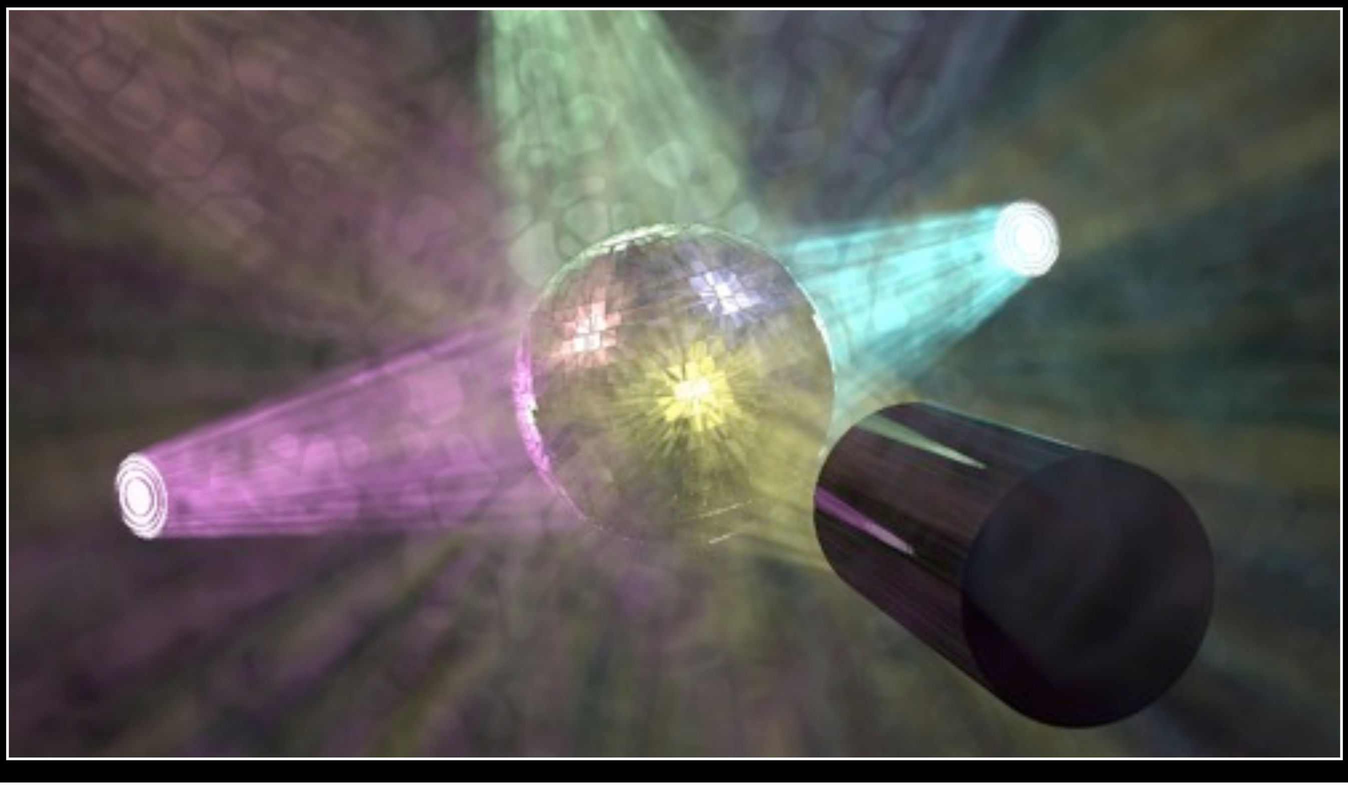
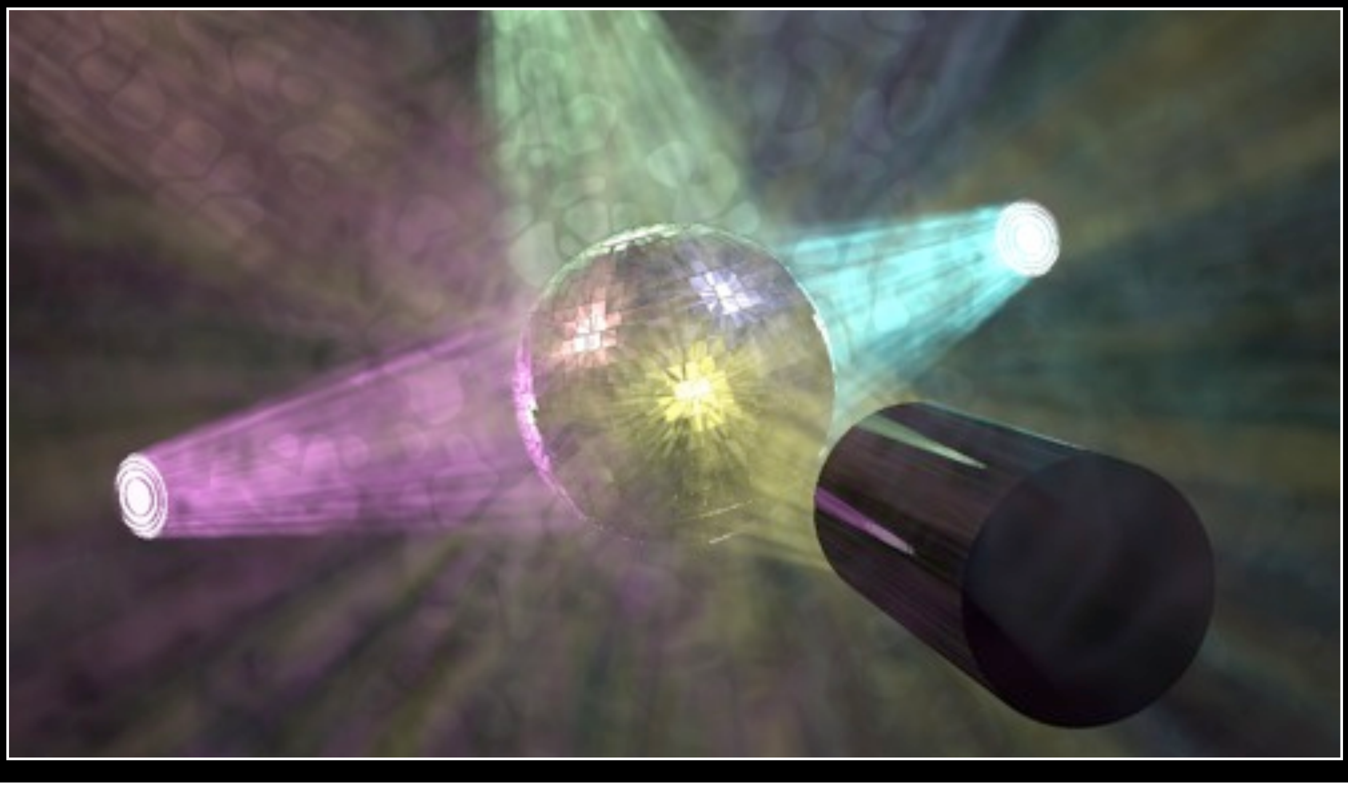
Heterogeneous

2.1M Photon Beams

10.8 minutes

Disco

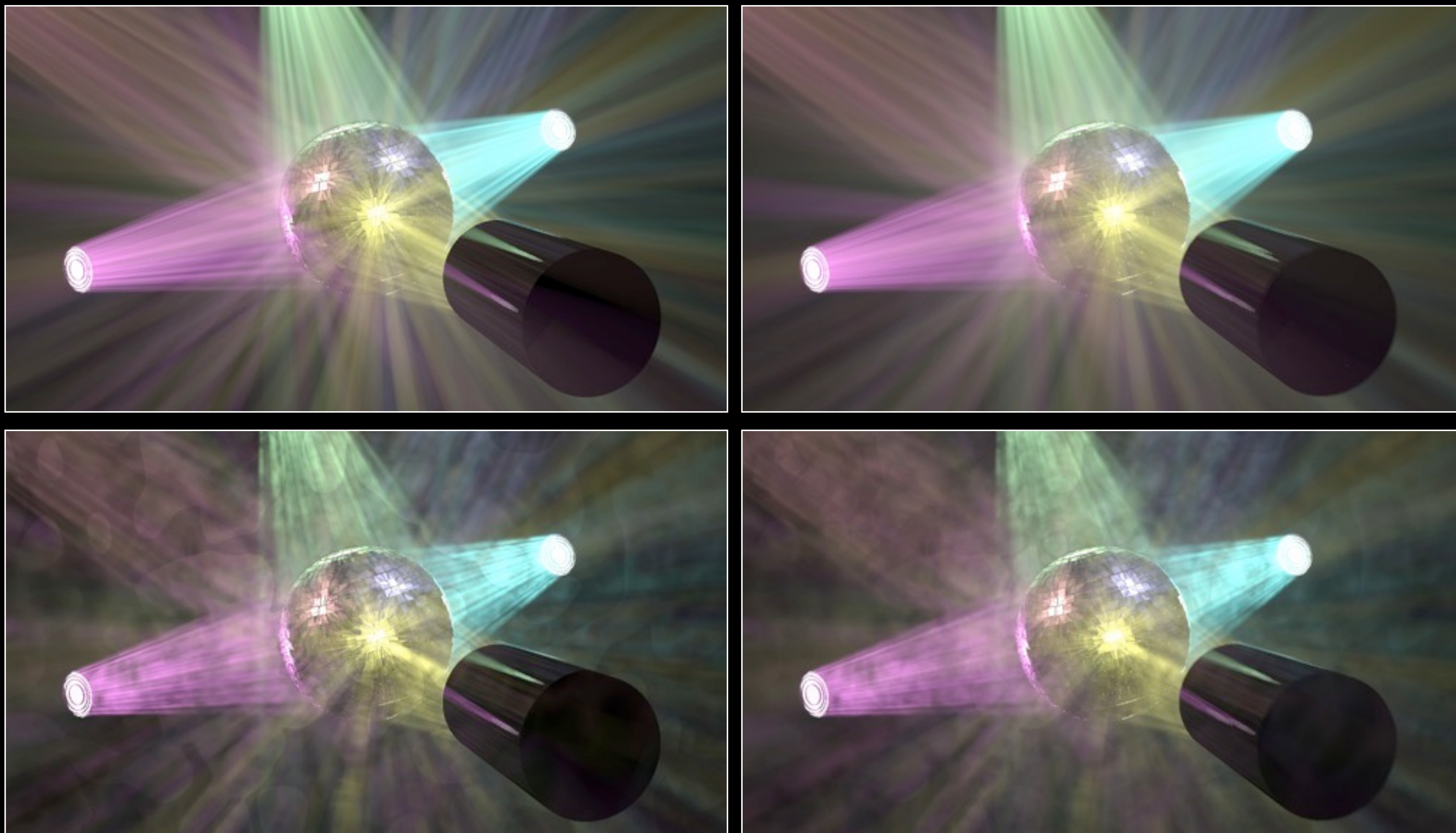
1280x720, Depth-of-Field

Pass 1	Average of Passes 1..1
	
	

Disco

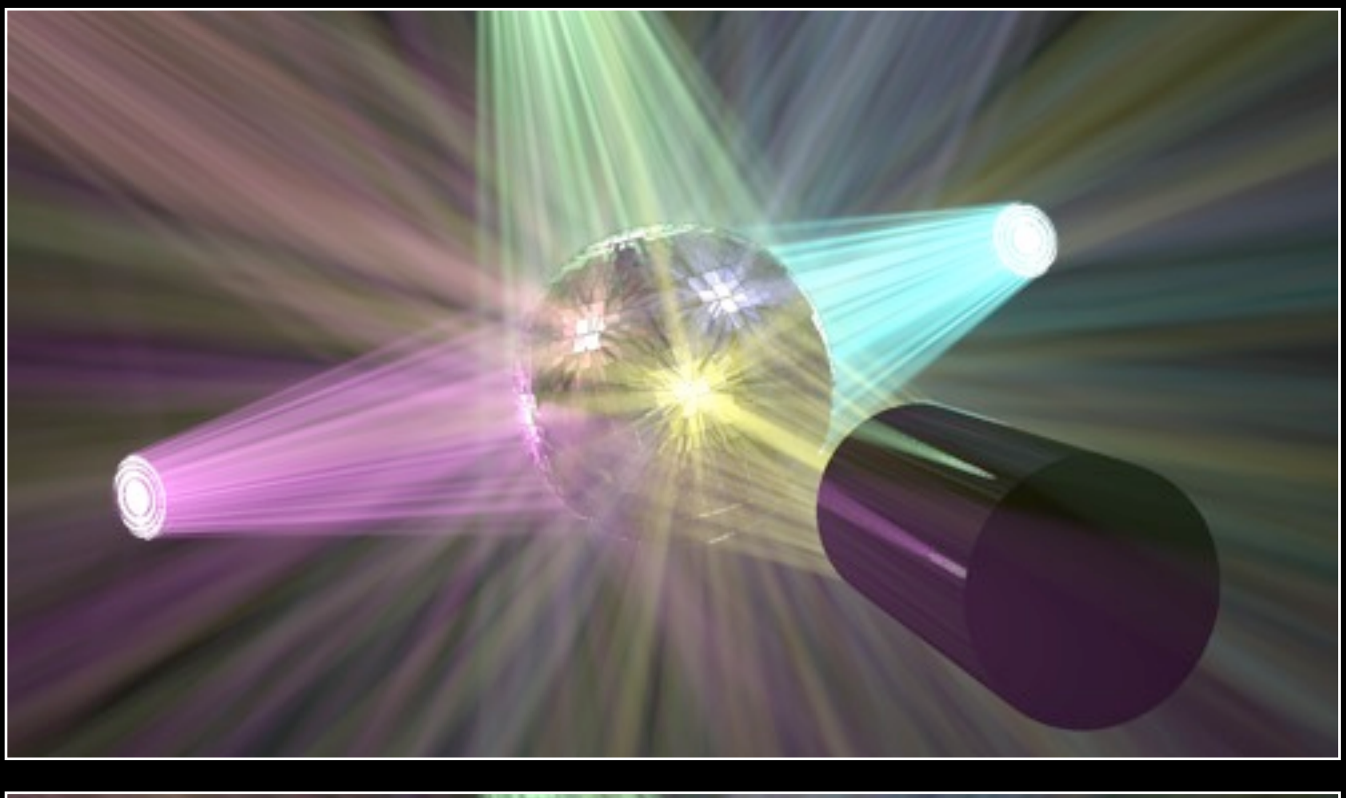
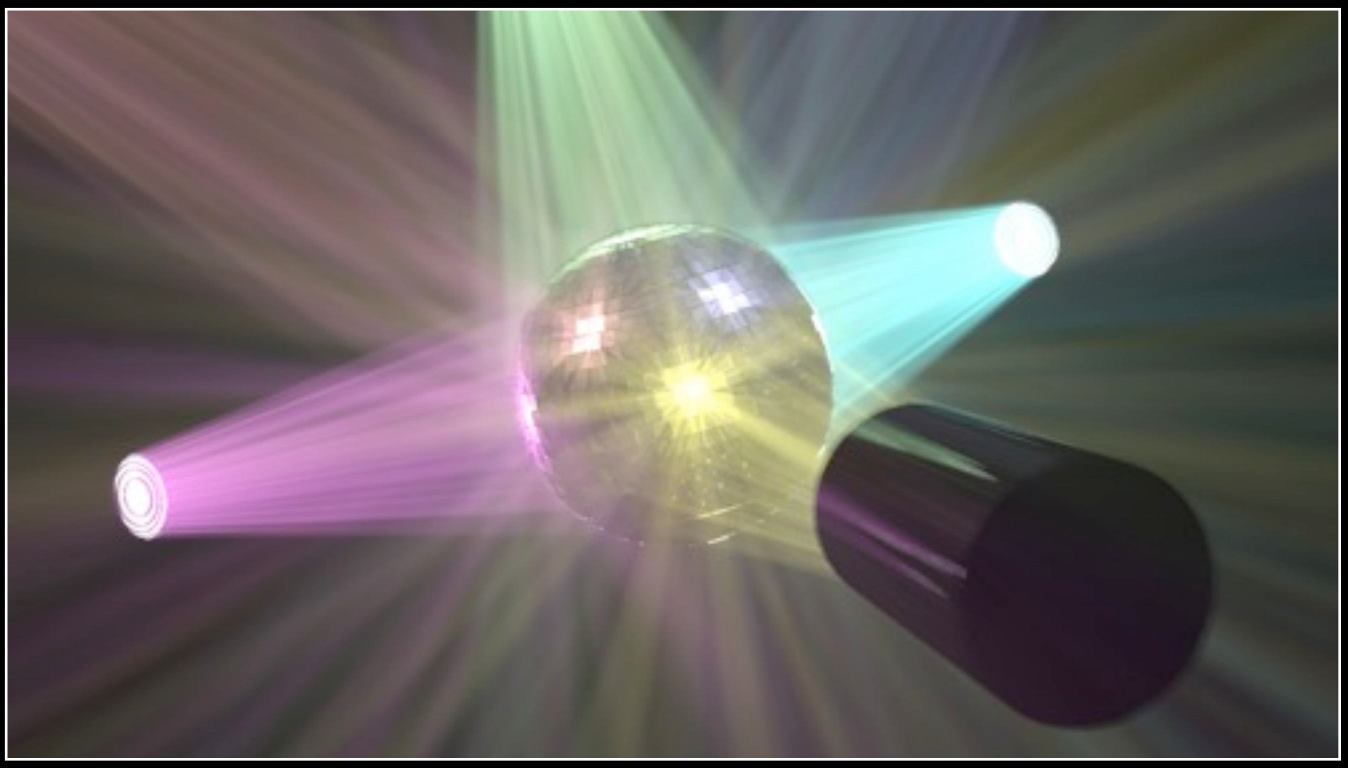
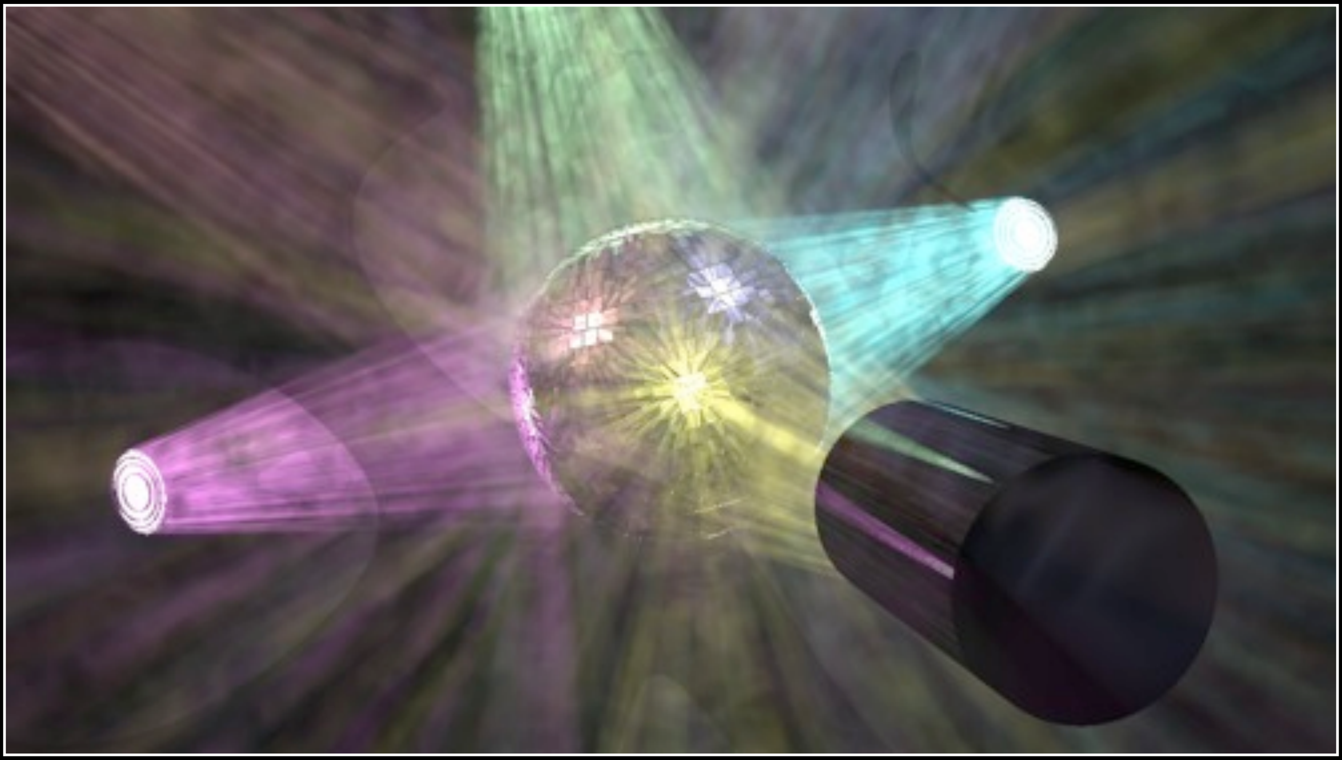
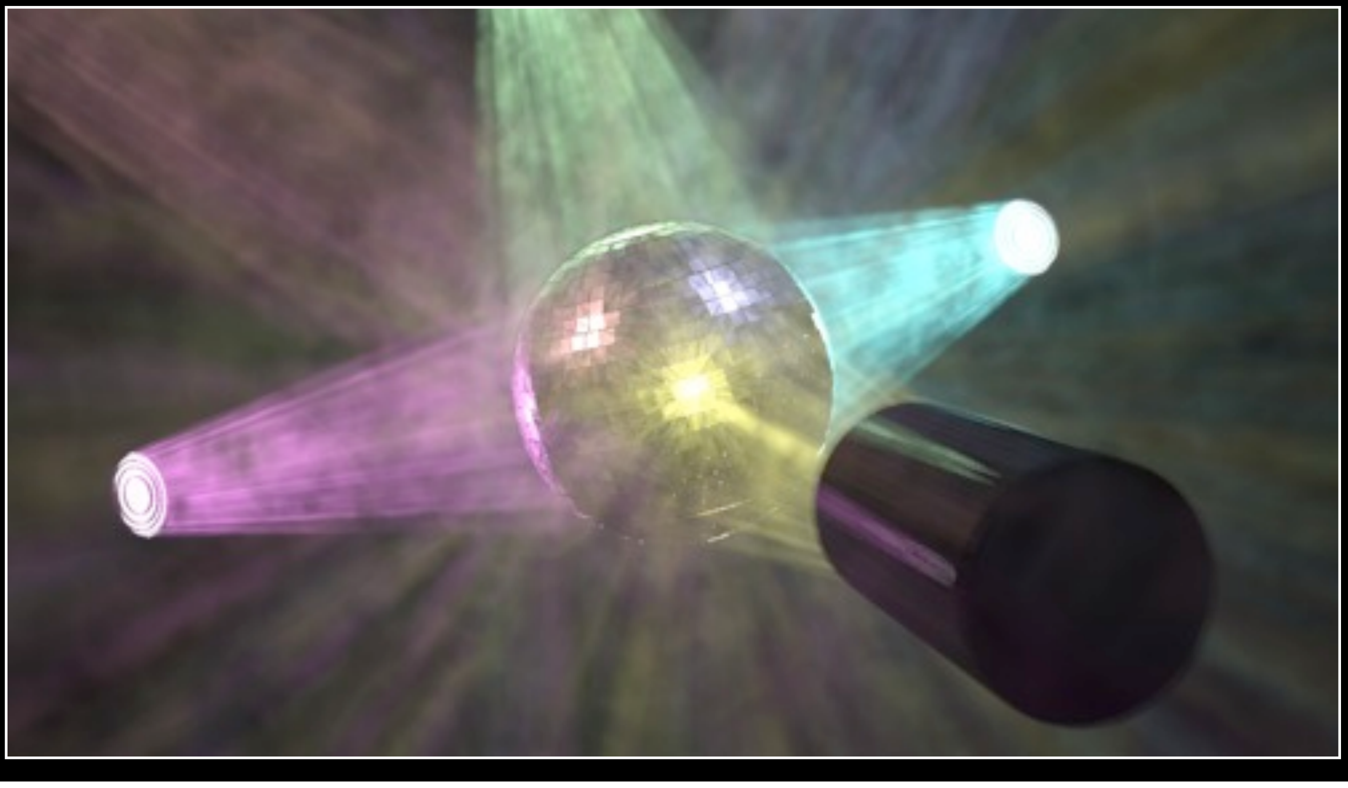
1280x720, Depth-of-Field

Pass 2 | Average of Passes 1..2



Disco

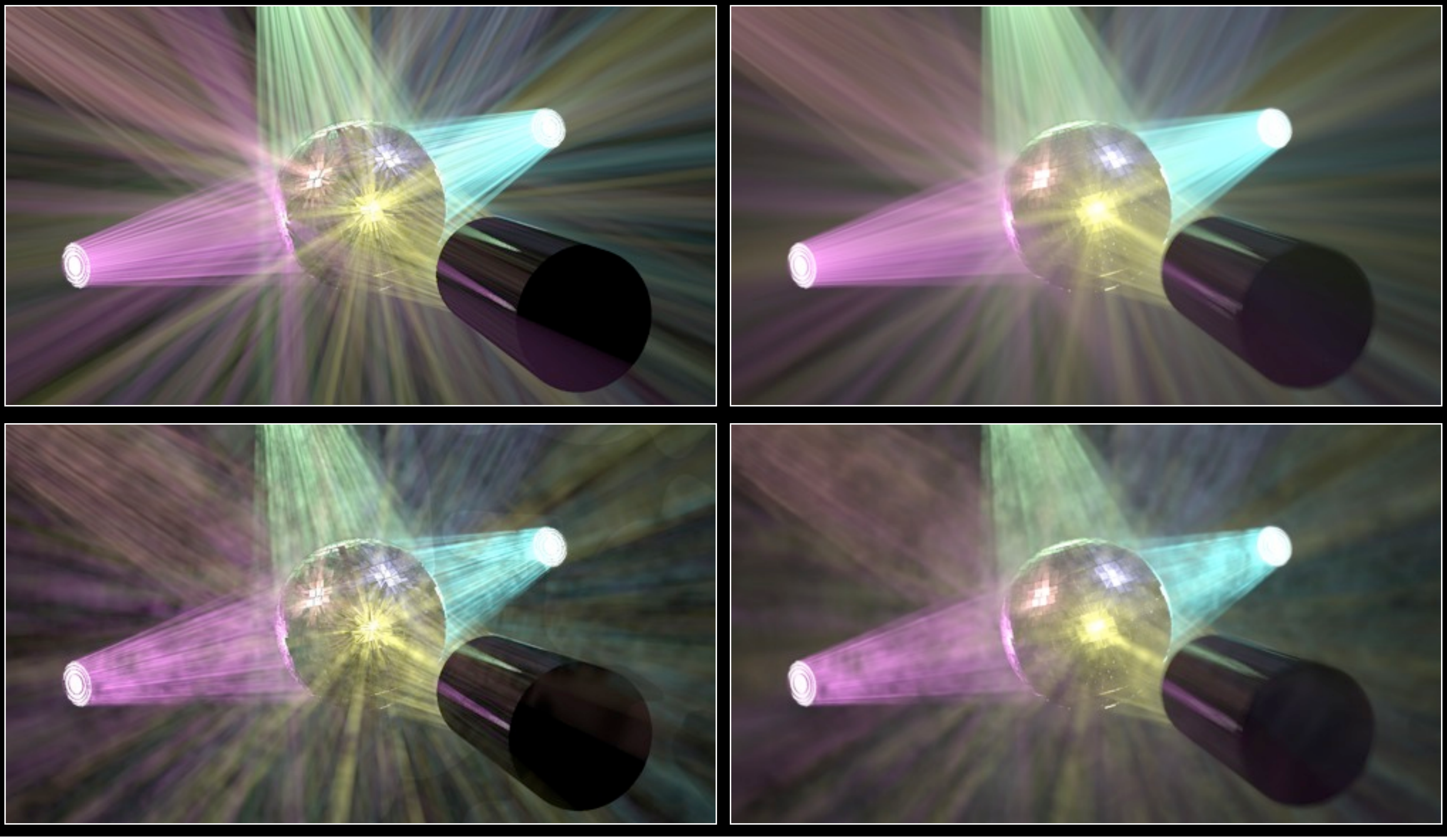
1280x720, Depth-of-Field

Pass 4	Average of Passes 1..4
	
	

Disco

1280x720, Depth-of-Field

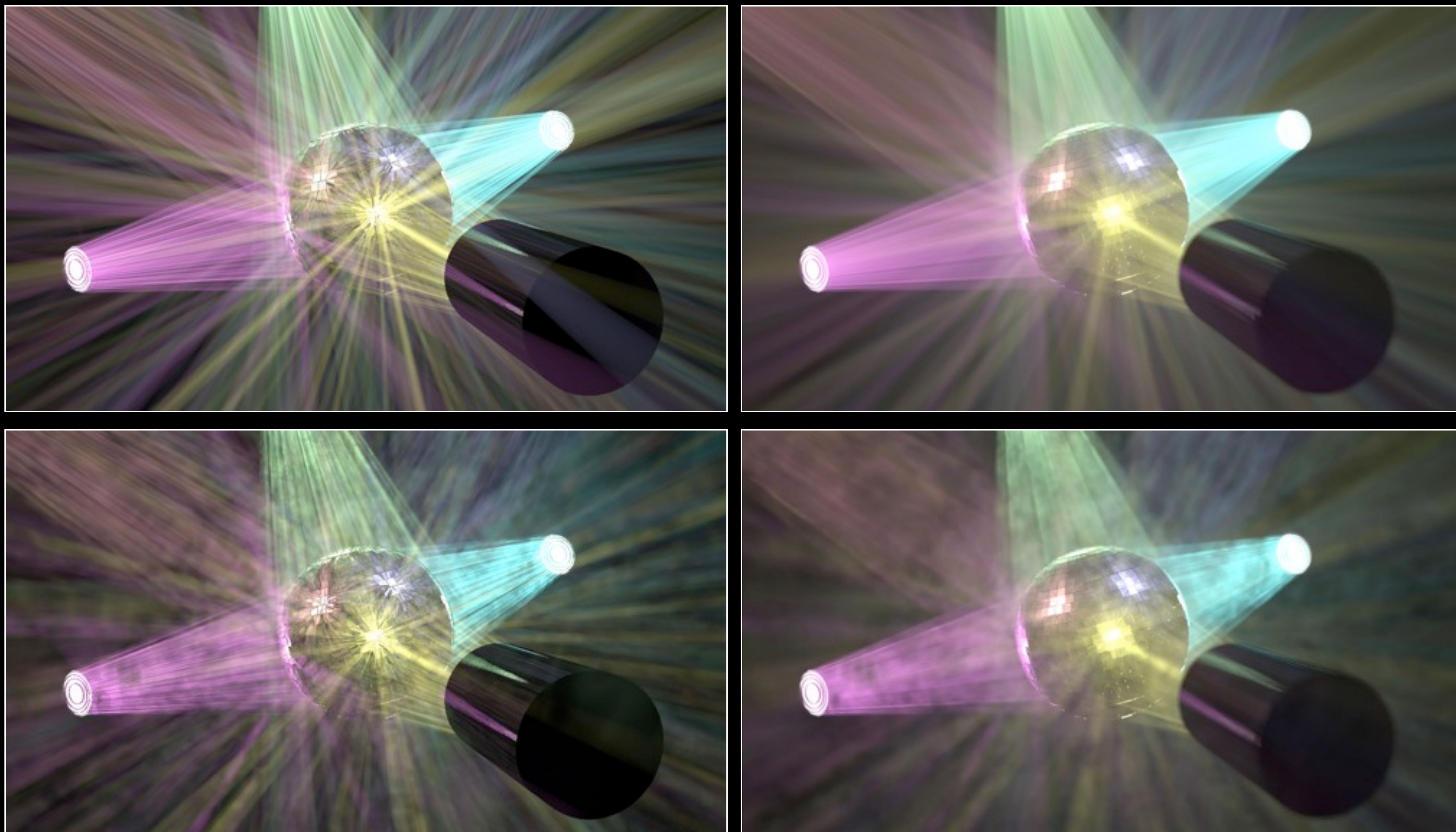
Pass 8	Average of Passes 1..8



Disco

1280x720, Depth-of-Field

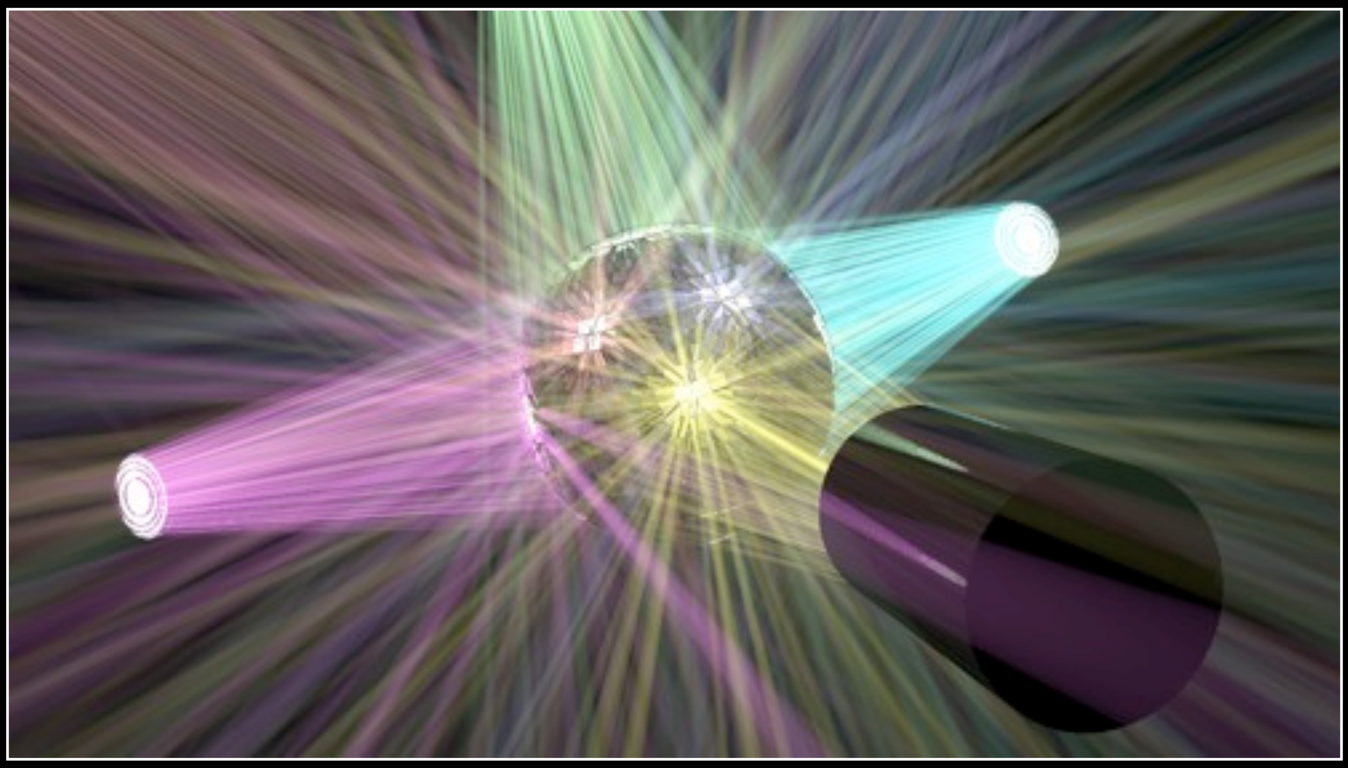
Pass 16 | Average of Passes 1..16

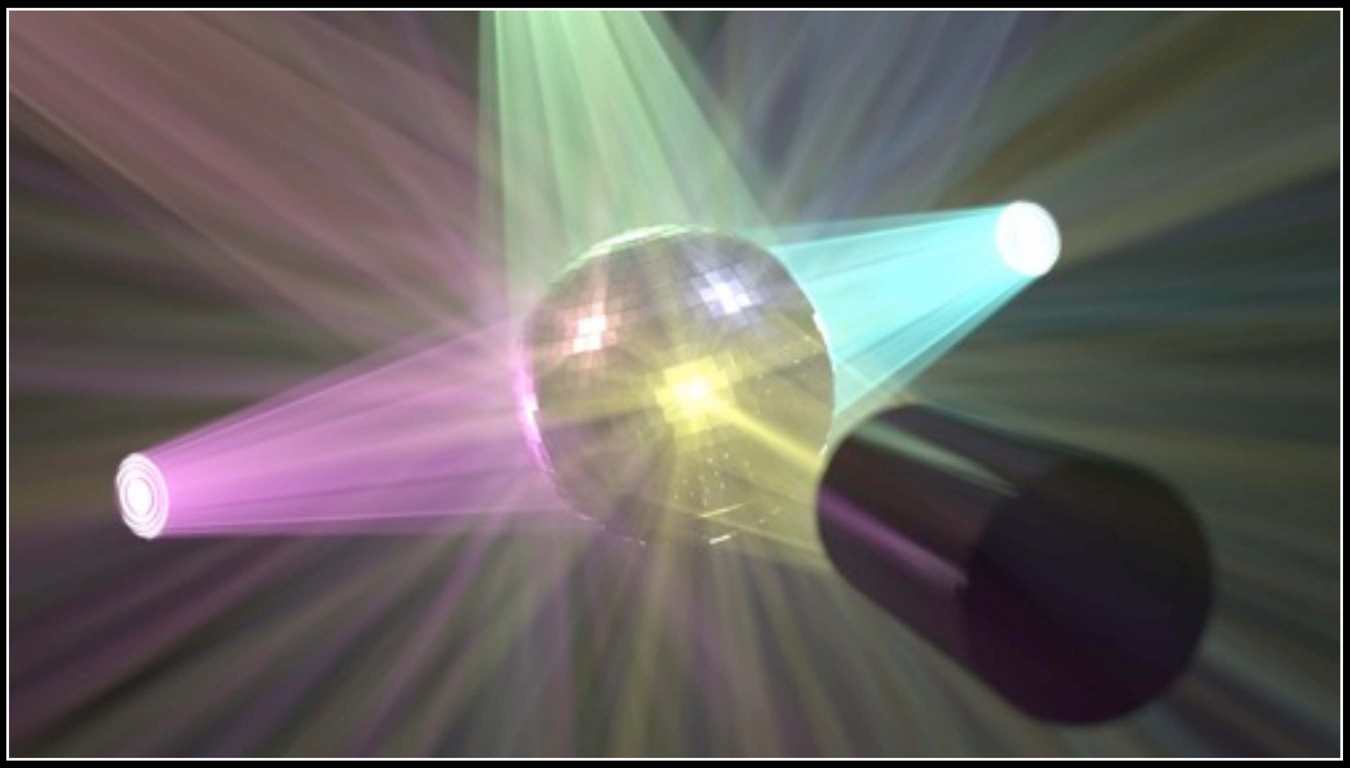


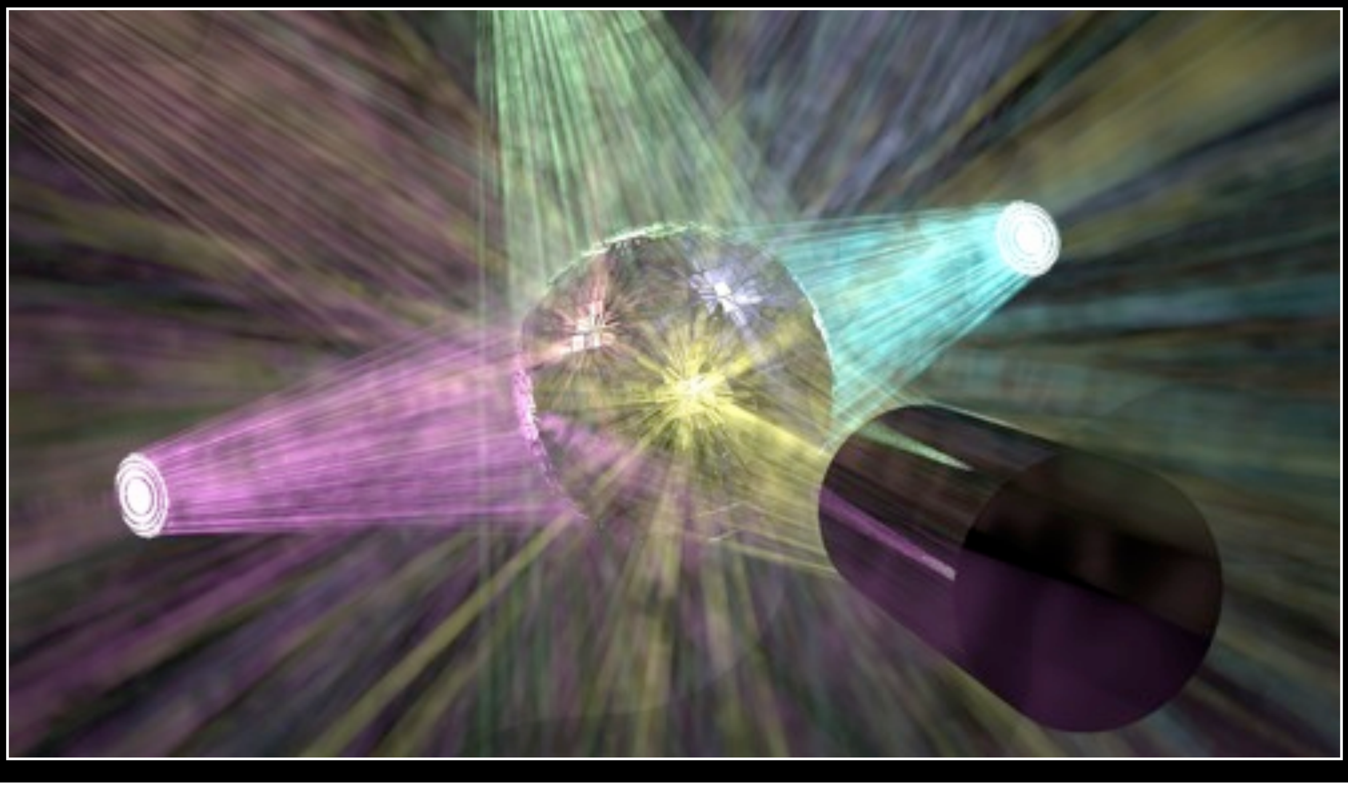
Disco

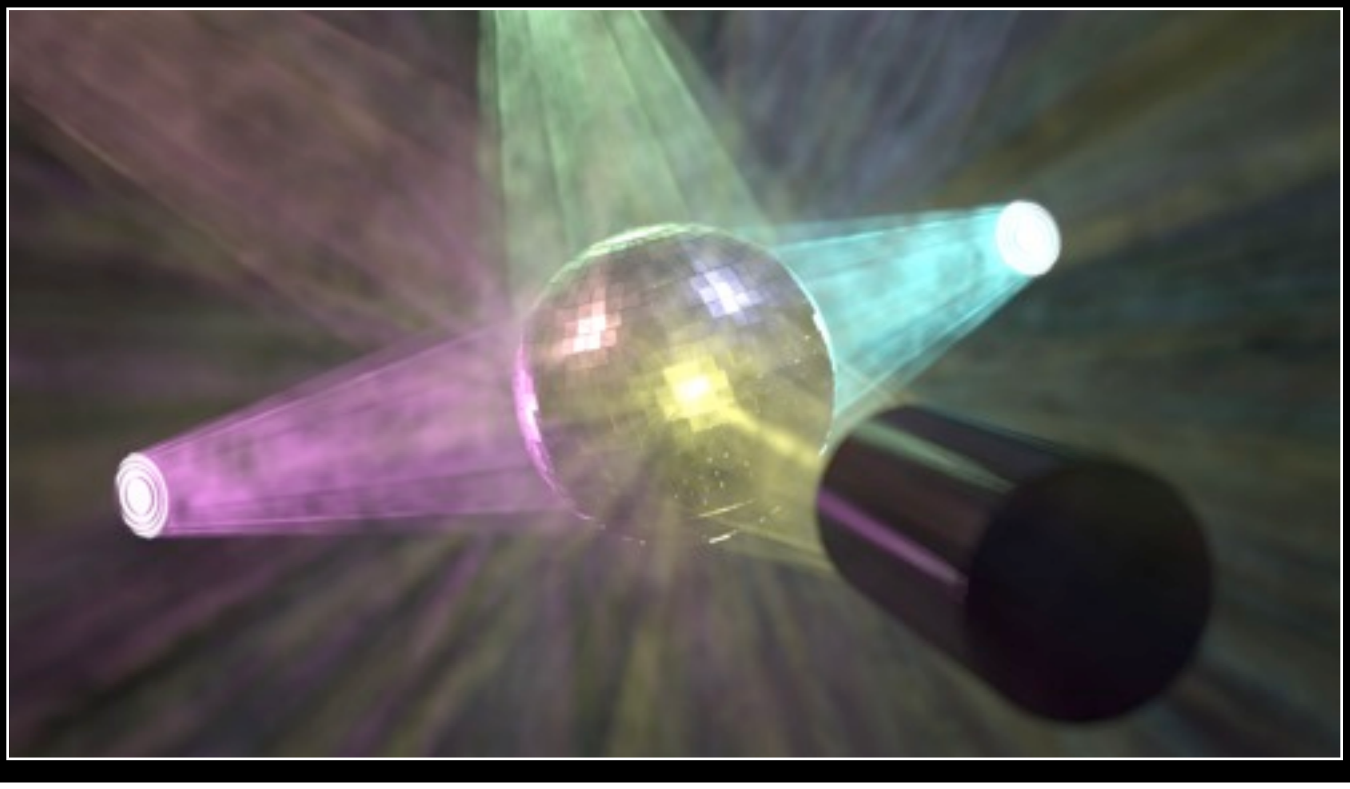
1280x720, Depth-of-Field

Pass 32 | Average of Passes 1..32





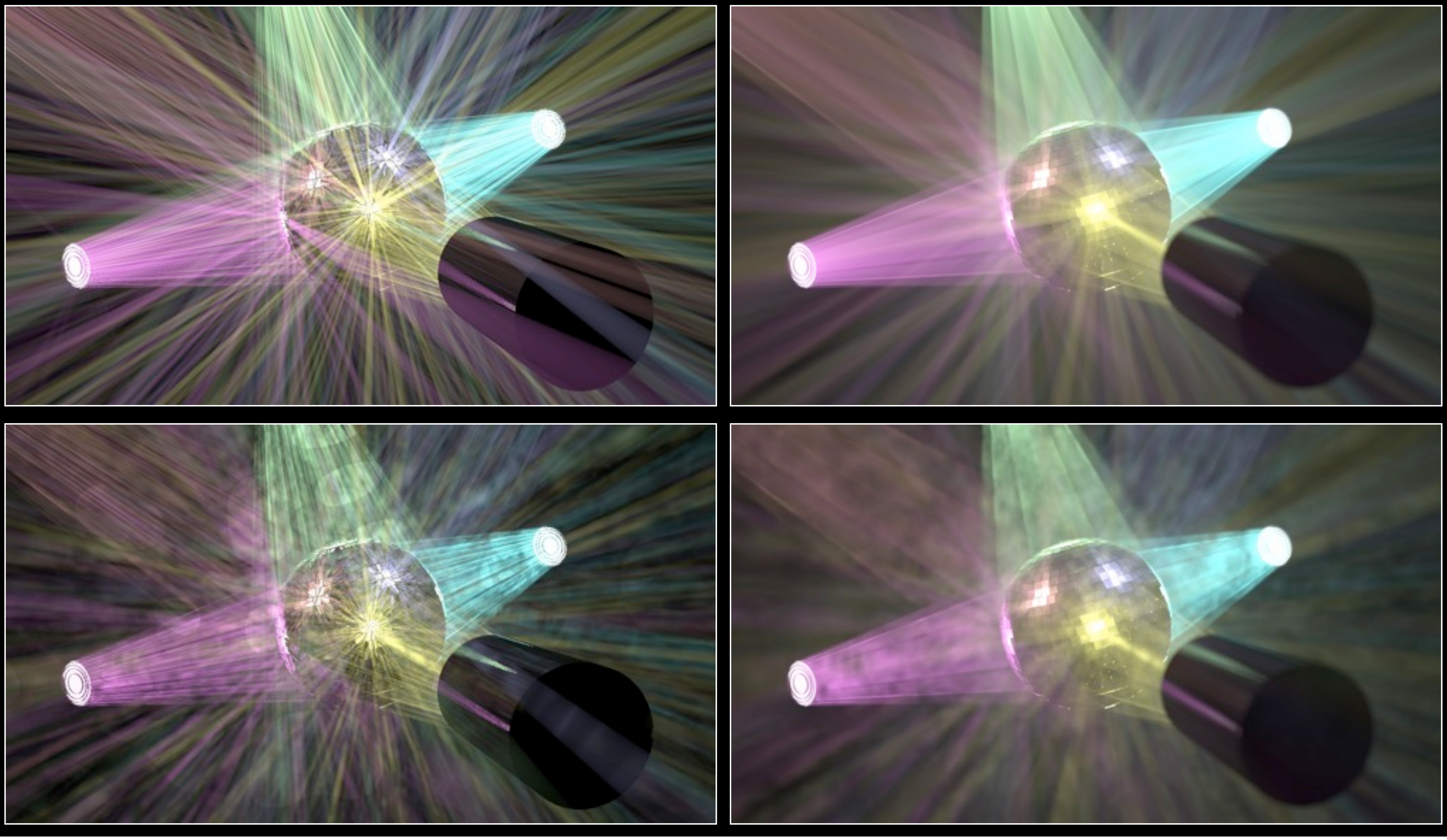




Disco

1280x720, Depth-of-Field

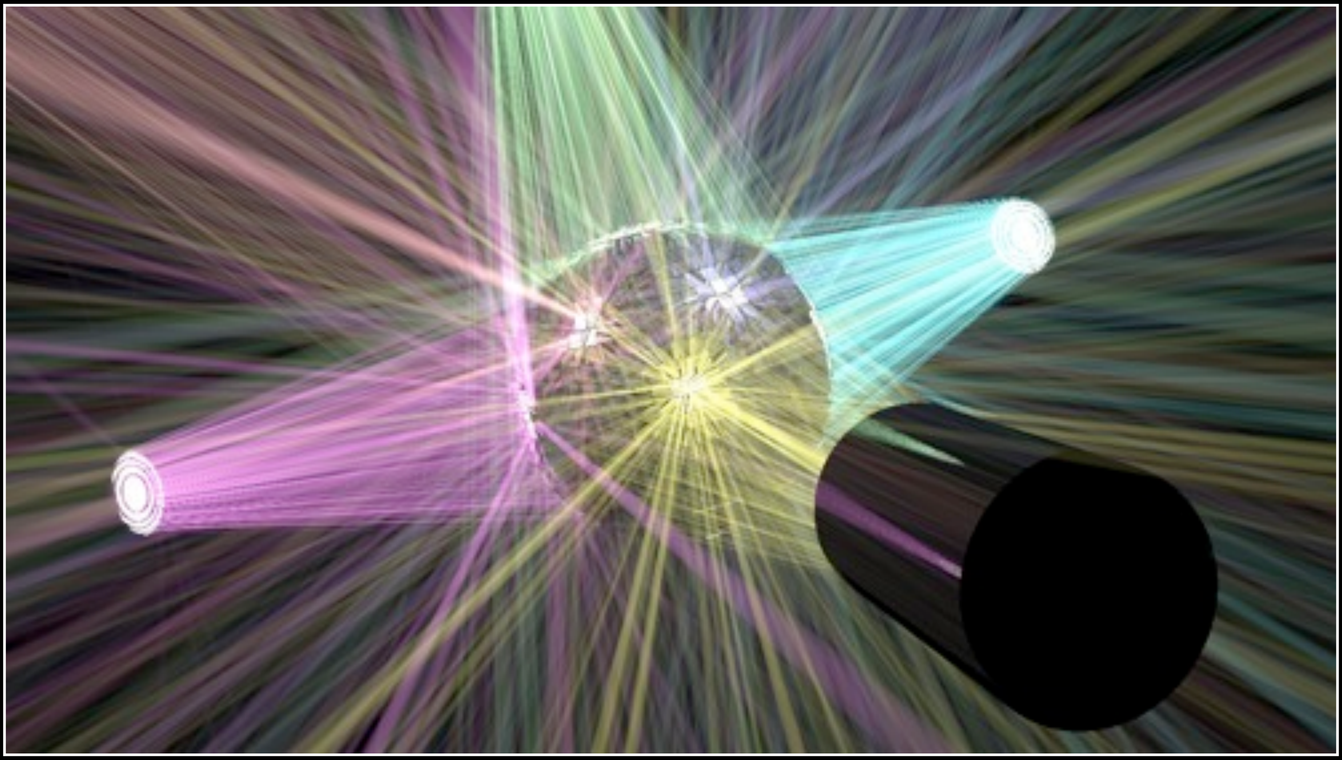
Pass 64 | Average of Passes 1..64

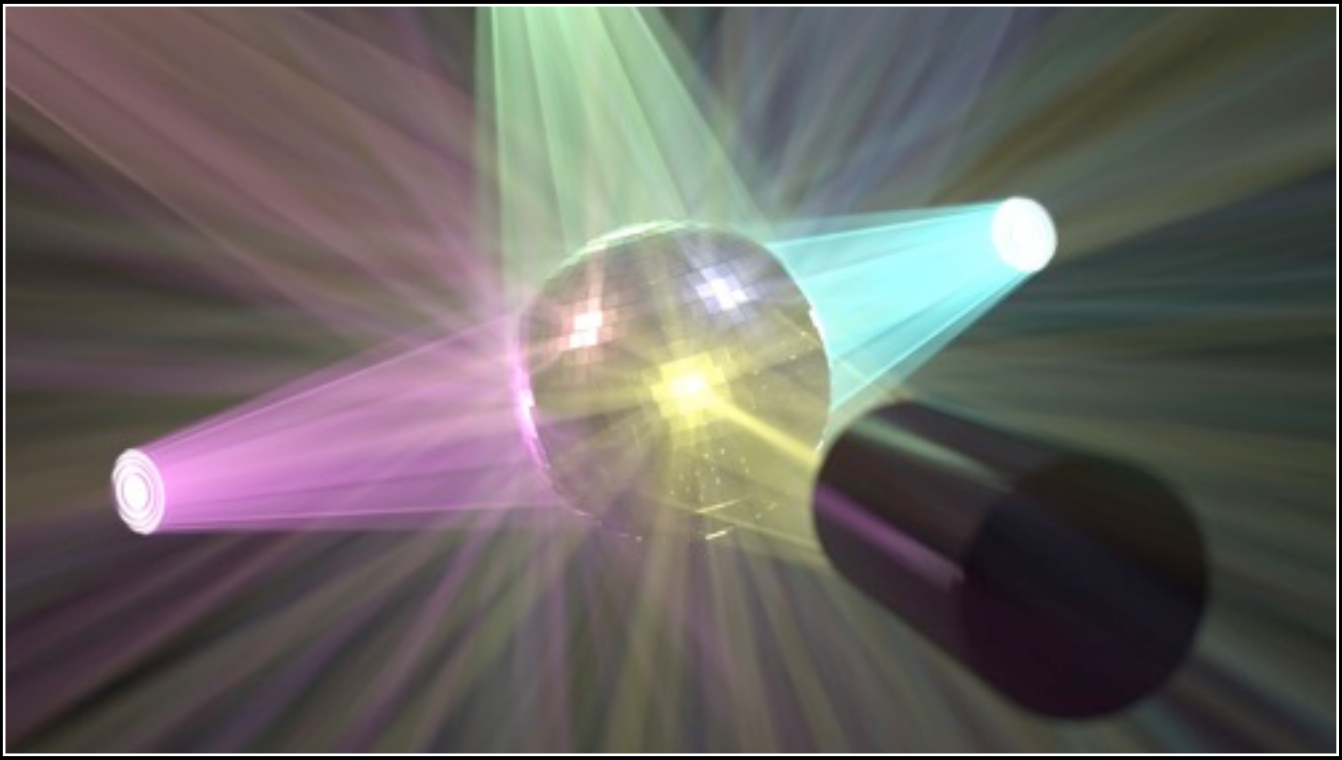


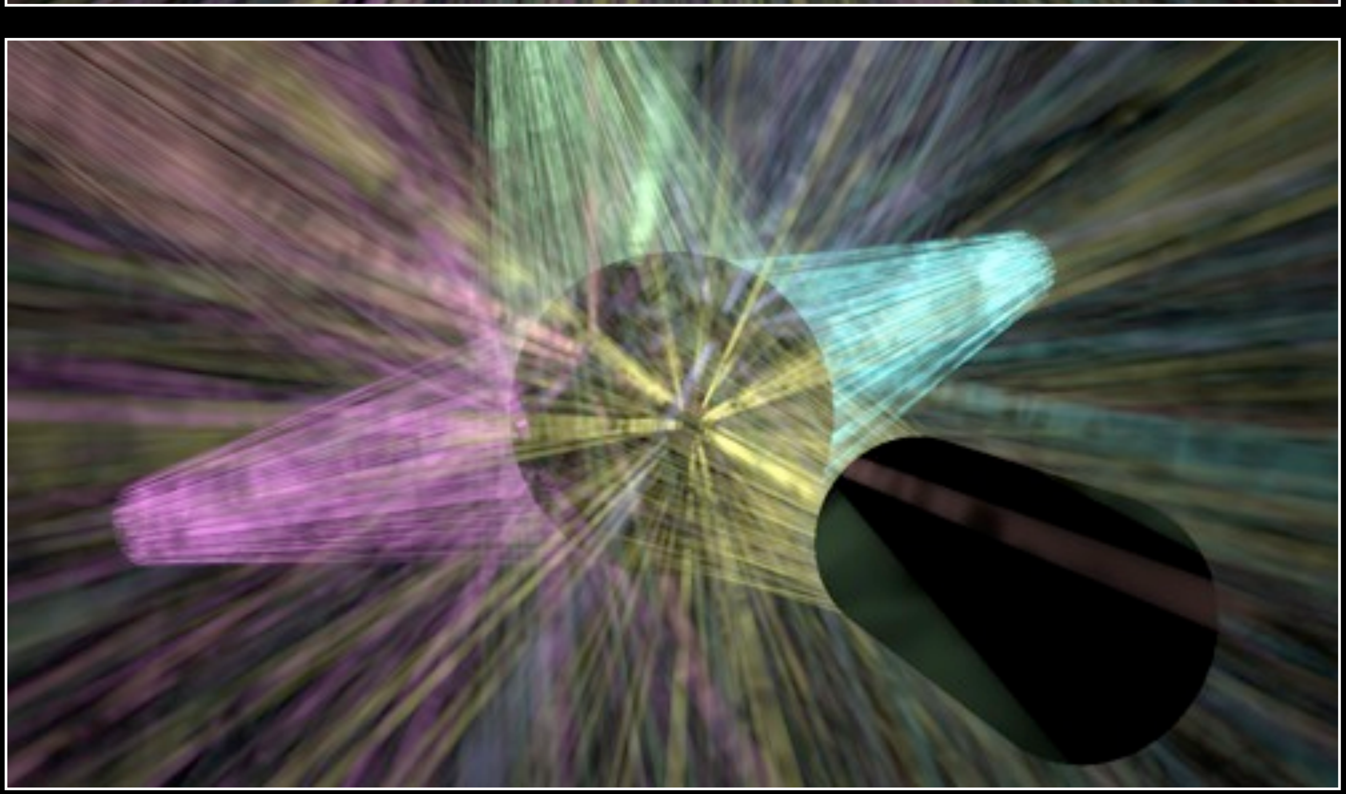
DISCO

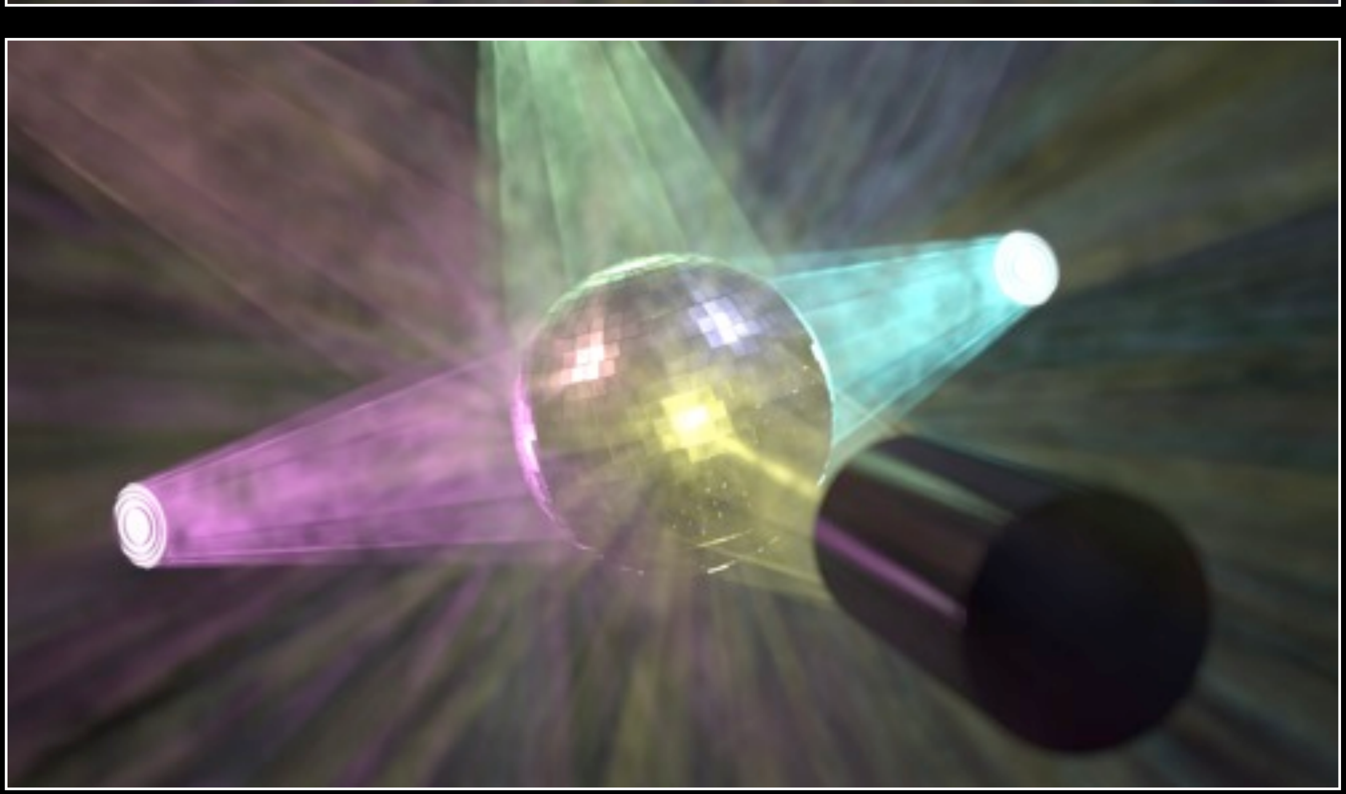
1280x720, Depth-of-Field

Pass 128 | Average of Passes 1..128





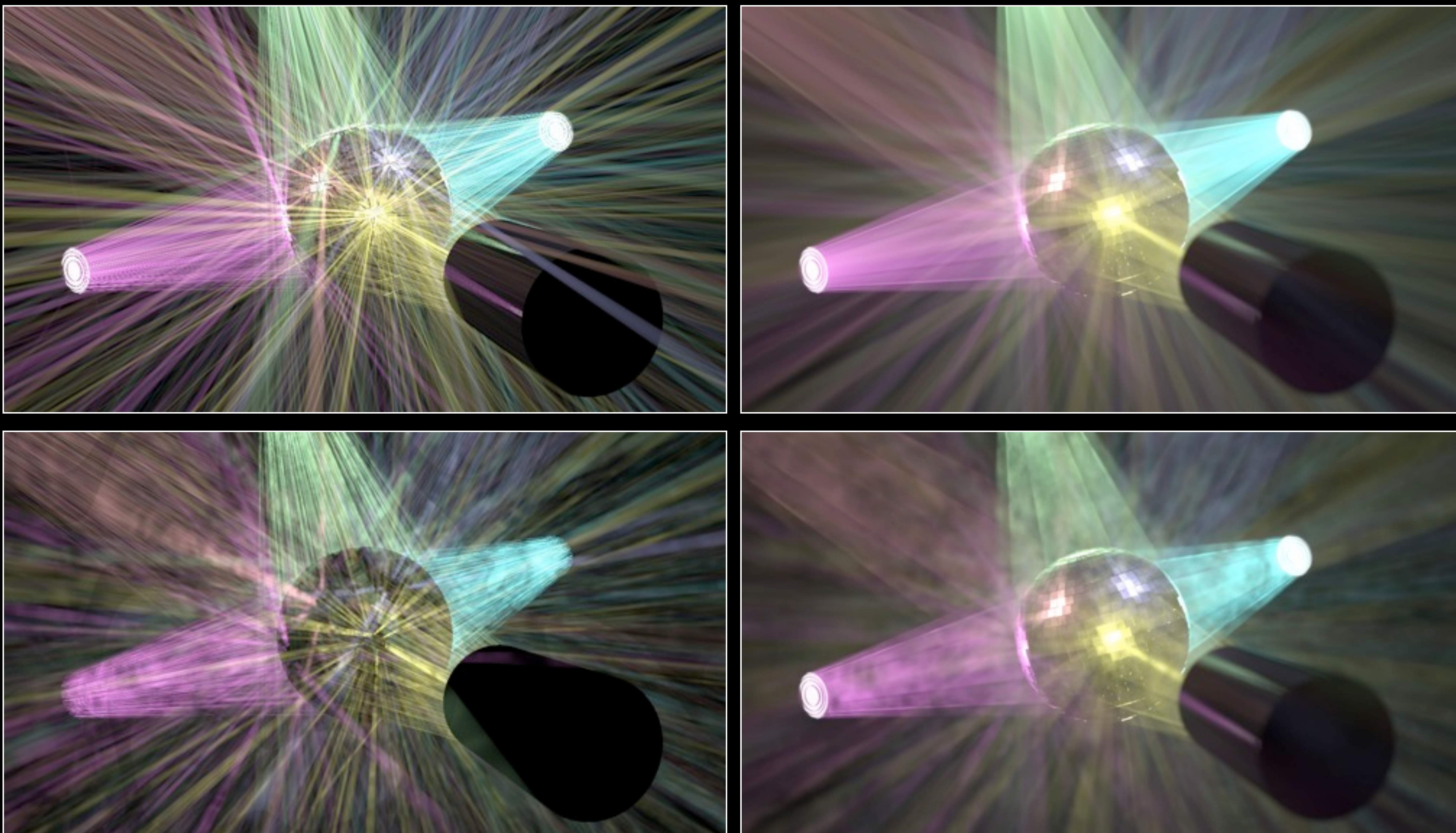




Disco

1280x720, Depth-of-Field

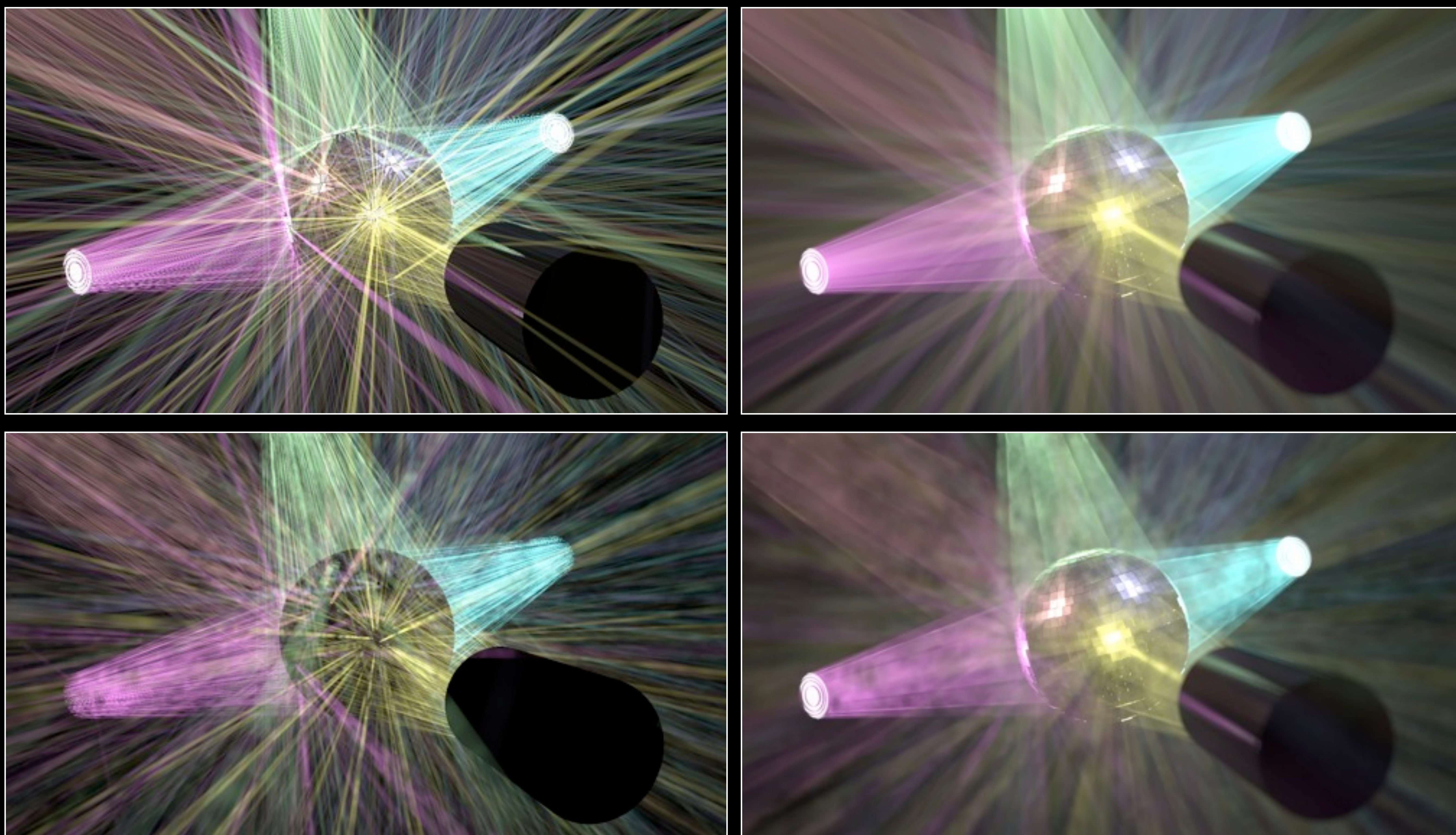
Pass 256 | Average of Passes 1..256



Disco

1280x720, Depth-of-Field

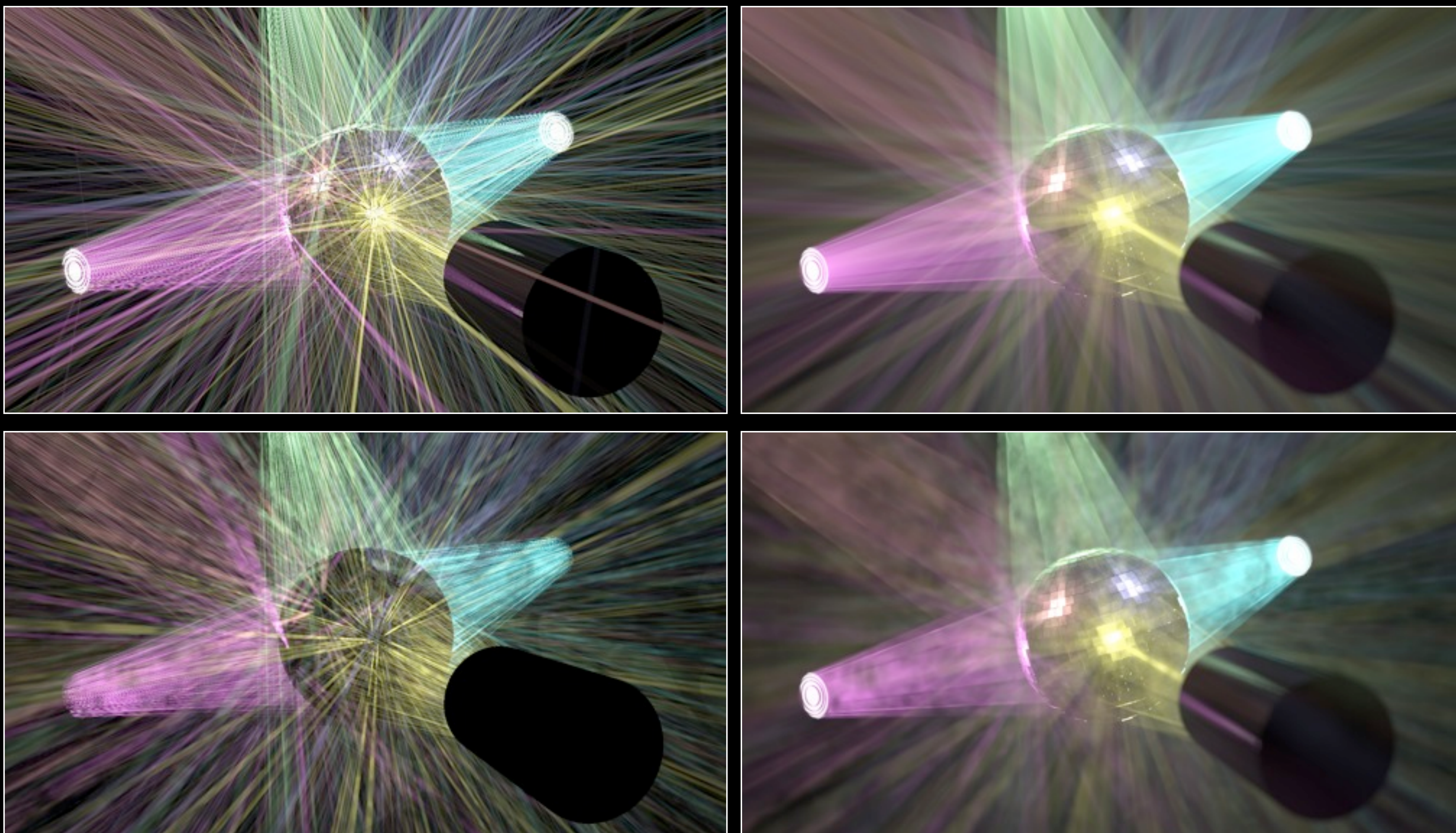
Pass 512 | Average of Passes 1..512



DISCO

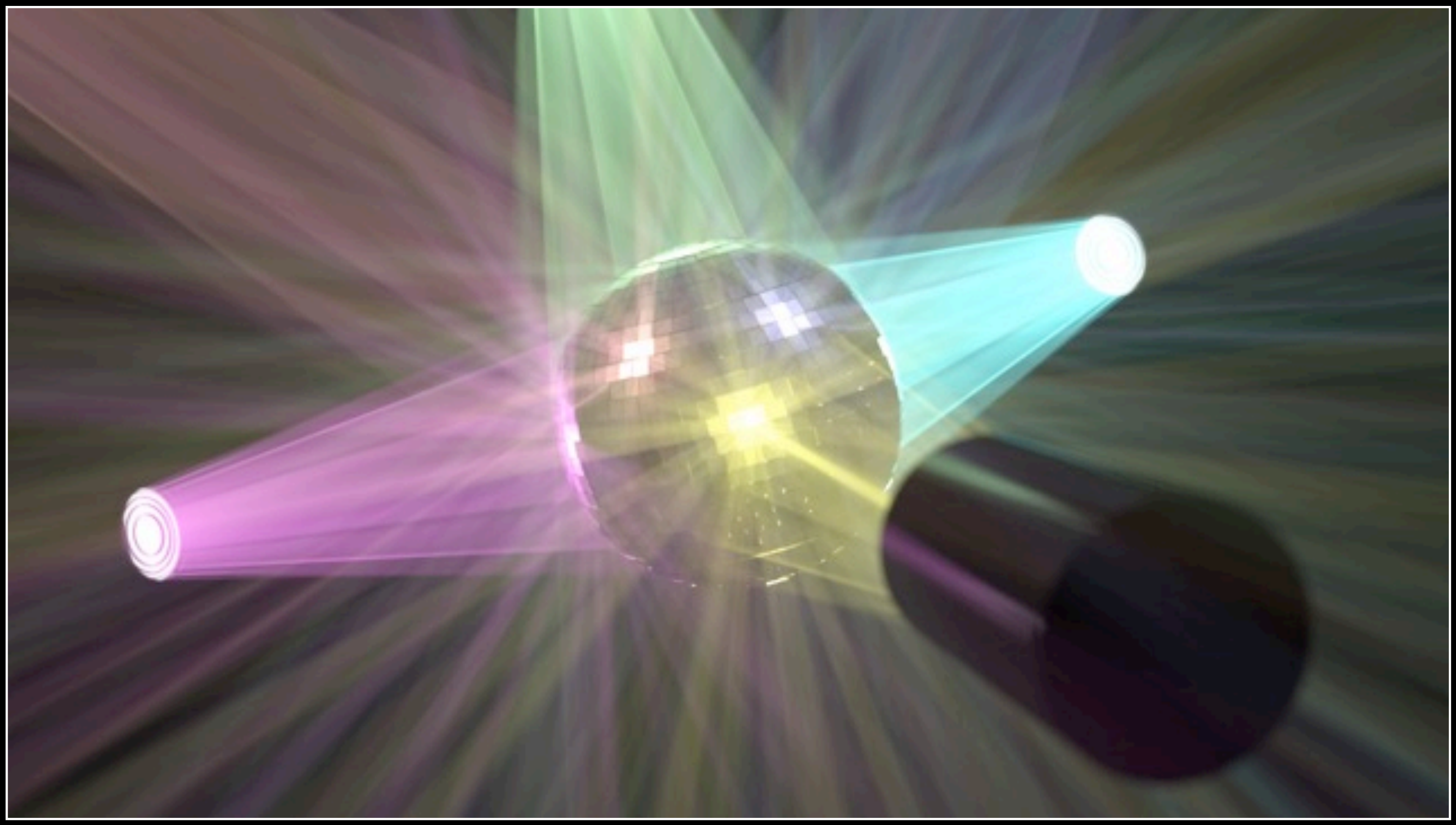
1280x720, Depth-of-Field

Pass 1024 | Average of Passes 1..1024



DISCO

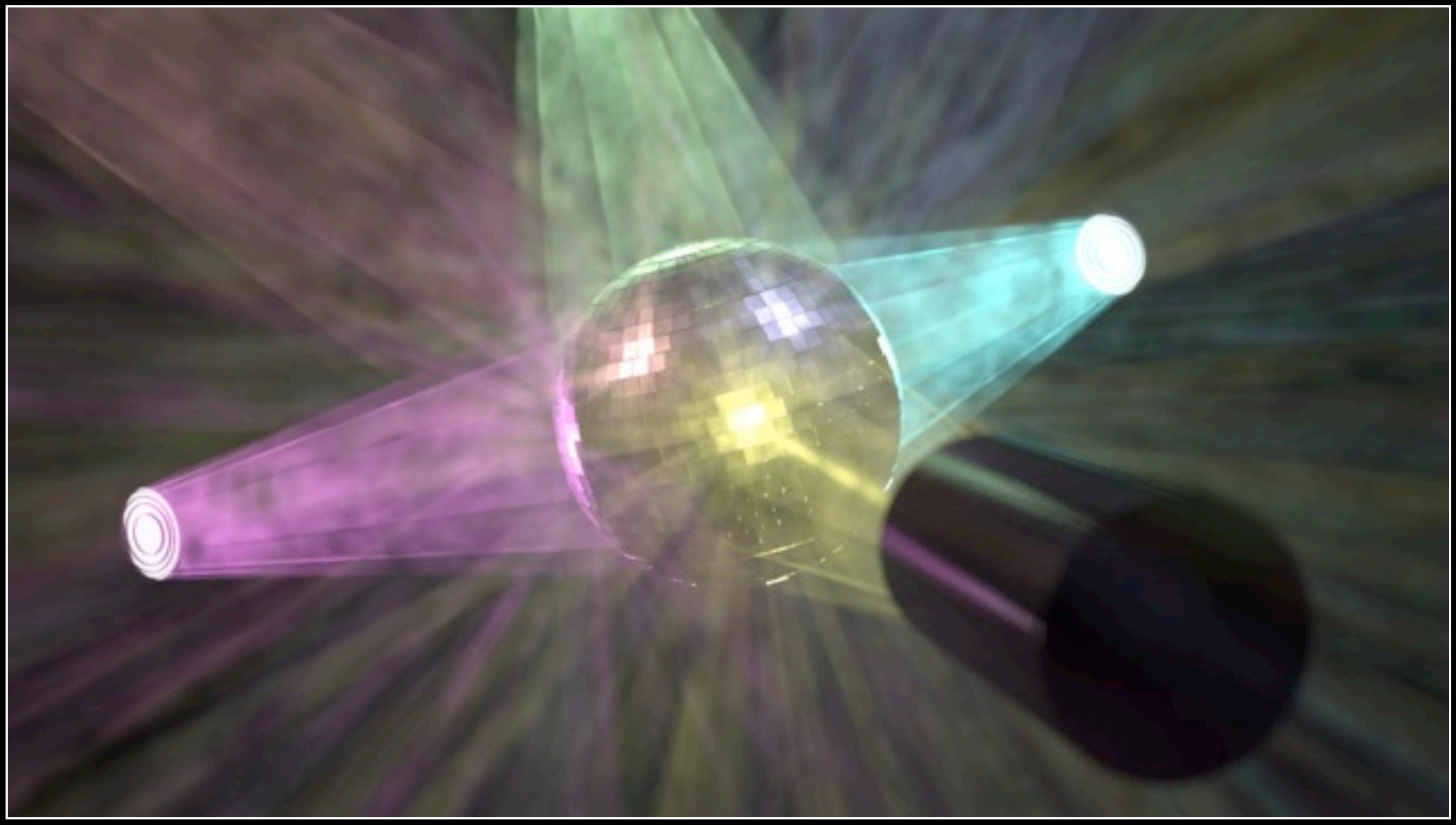
1280x720, Depth-of-Field



Homogeneous

19.67M Photon Beams

3 minutes



Heterogeneous

16.19M Photon Beams

5.7 minutes

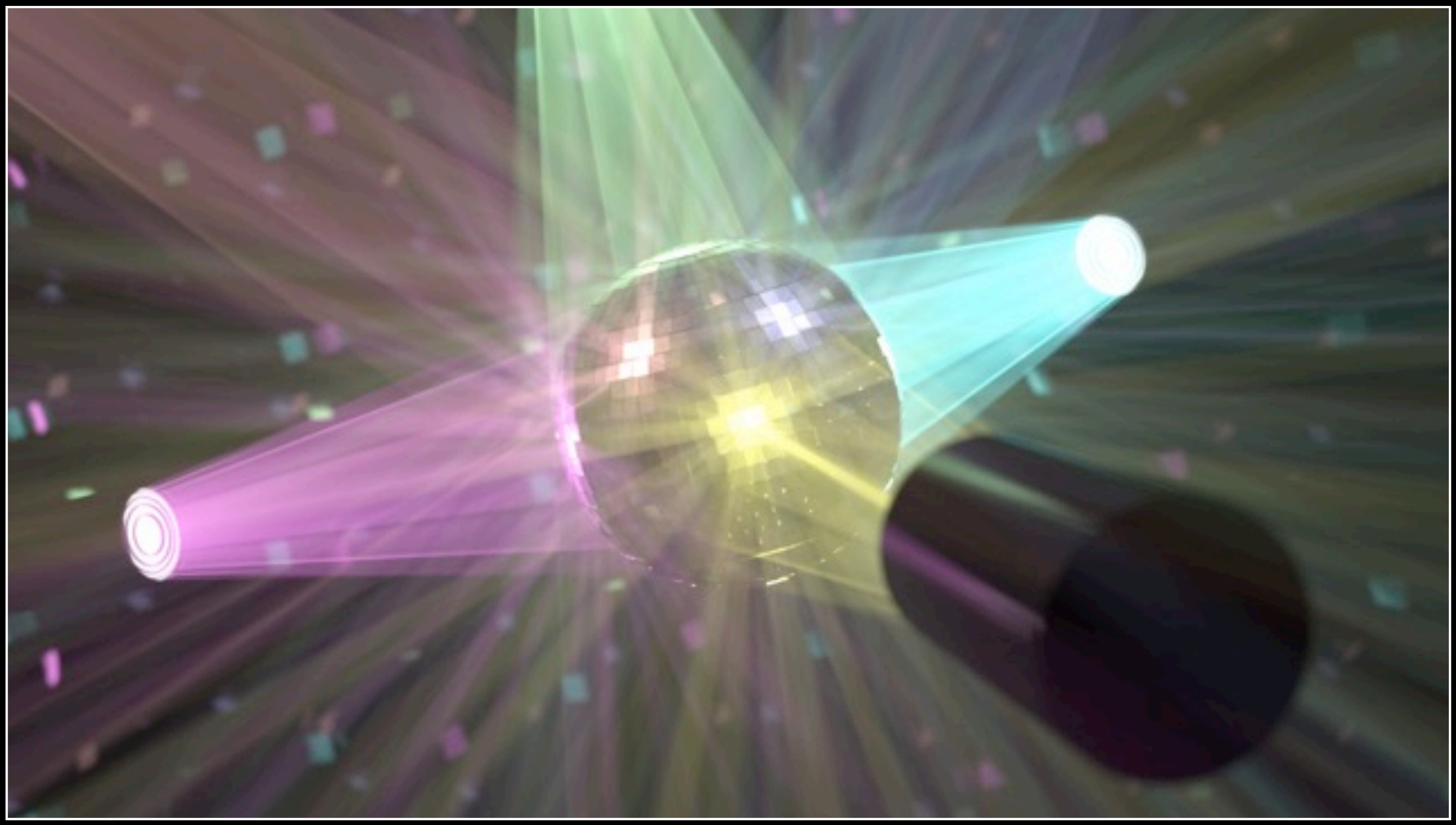
Disco

1280x720, Depth-of-Field

Homogeneous

19.67M Photon Beams

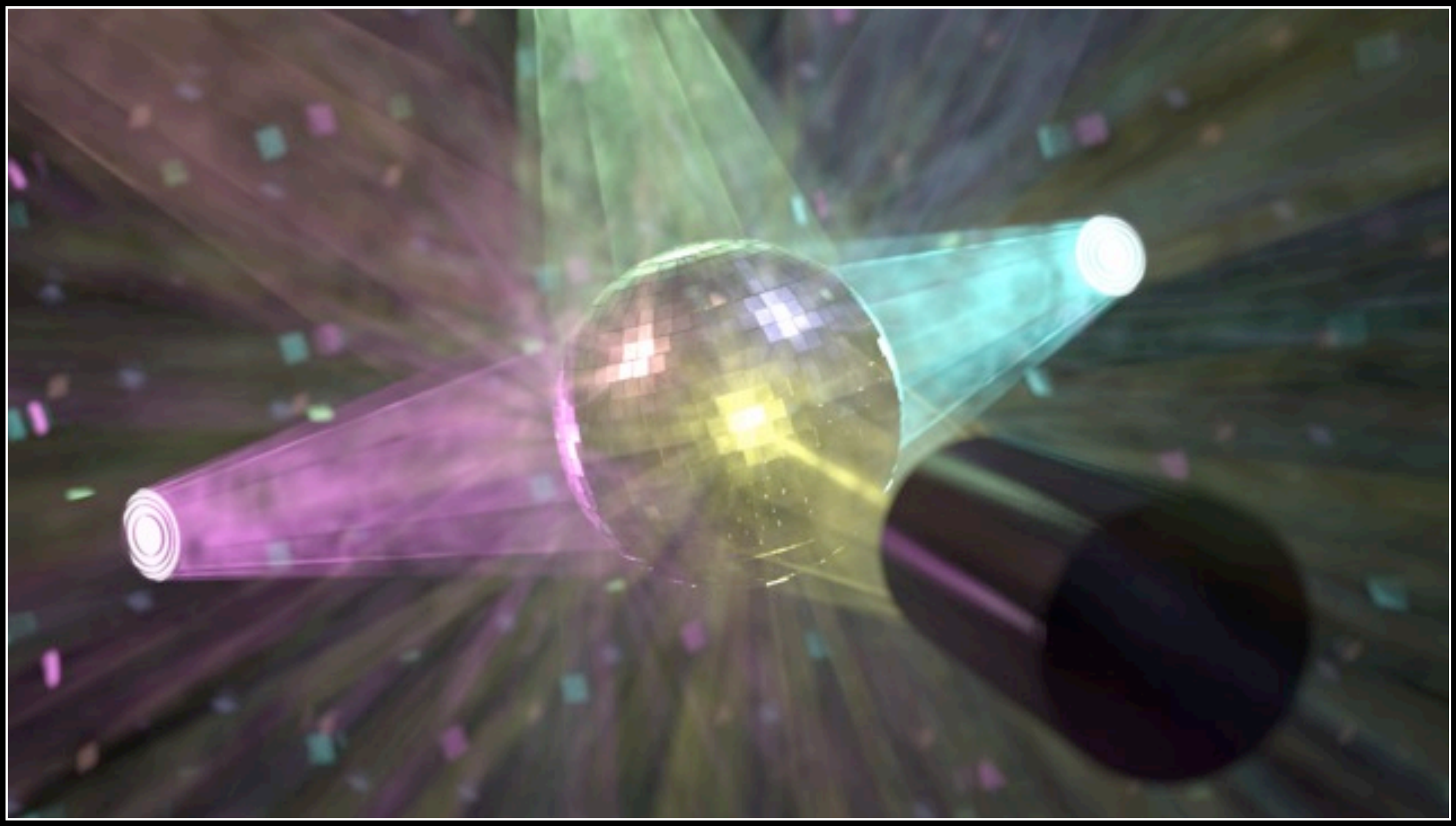
3 minutes

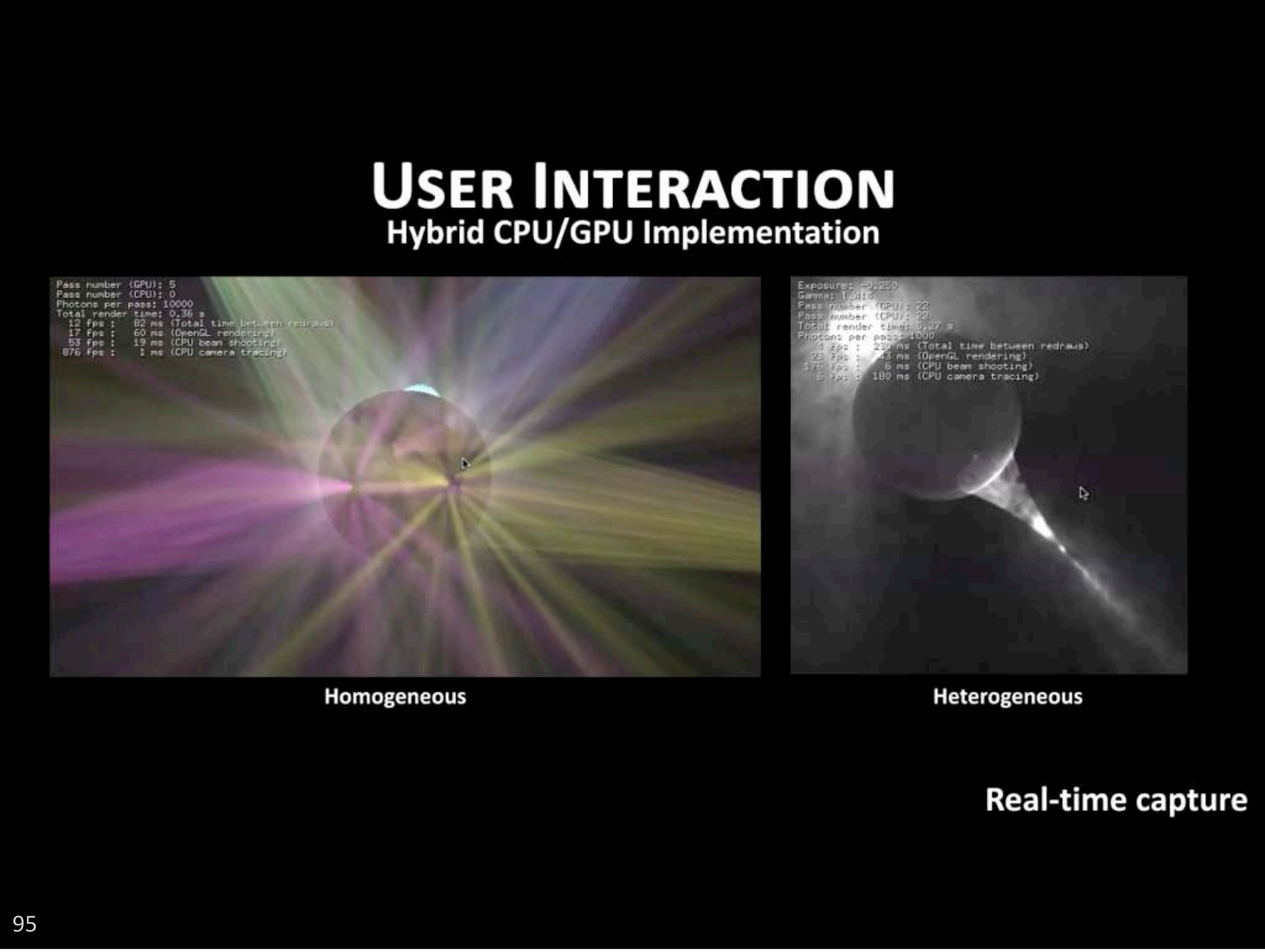


Heterogeneous

16.19M Photon Beams

5.7 minutes





Thursday, 6 September 12

- An important aspect of our algorithm is that it provides a quick interactive preview when manipulating the scene, while rapidly converging to the ground-truth solution when the user lets go of the mouse.

Limitations & Future Work

- There are of course some limitations
- Firstly, photon beams are fantastic at reconstructing caustic light paths, but multiple scattering effects can still be quite costly
- Also, though our approach is convergent with any alpha between 0 and 1, this is fixed for the entire image, it might be possible to adapt alpha in different regions of the image to accelerate convergence
- Our theory tells you how to reduce the radii if you want an unbiased image in the limit. However, an interesting practical question is, if you know you only have a certain time budget, would this affect the optimal reduction factor?
- Finally, though our transmittance estimator is unbiased, it also increases variance, especially in dense media. It would be interesting to see if an unbiased but smoother estimator for transmittance is possible.

Limitations & Future Work

- Multiple scattering (relatively) costly

- There are of course some limitations
- Firstly, photon beams are fantastic at reconstructing caustic light paths, but multiple scattering effects can still be quite costly
- Also, though our approach is convergent with any alpha between 0 and 1, this is fixed for the entire image, it might be possible to adapt alpha in different regions of the image to accelerate convergence
- Our theory tells you how to reduce the radii if you want an unbiased image in the limit. However, an interesting practical question is, if you know you only have a certain time budget, would this affect the optimal reduction factor?
- Finally, though our transmittance estimator is unbiased, it also increases variance, especially in dense media. It would be interesting to see if an unbiased but smoother estimator for transmittance is possible.

Limitations & Future Work

- Multiple scattering (relatively) costly
- Single α not optimal for entire image
  - adaptive α possible?

- There are of course some limitations
- Firstly, photon beams are fantastic at reconstructing caustic light paths, but multiple scattering effects can still be quite costly
- Also, though our approach is convergent with any alpha between 0 and 1, this is fixed for the entire image, it might be possible to adapt alpha in different regions of the image to accelerate convergence
- Our theory tells you how to reduce the radii if you want an unbiased image in the limit. However, an interesting practical question is, if you know you only have a certain time budget, would this affect the optimal reduction factor?
- Finally, though our transmittance estimator is unbiased, it also increases variance, especially in dense media. It would be interesting to see if an unbiased but smoother estimator for transmittance is possible.

Limitations & Future Work

- Multiple scattering (relatively) costly
- Single α not optimal for entire image
  - adaptive α possible?
- Radius reduction for finite time budget?

- There are of course some limitations
- Firstly, photon beams are fantastic at reconstructing caustic light paths, but multiple scattering effects can still be quite costly
- Also, though our approach is convergent with any alpha between 0 and 1, this is fixed for the entire image, it might be possible to adapt alpha in different regions of the image to accelerate convergence
- Our theory tells you how to reduce the radii if you want an unbiased image in the limit. However, an interesting practical question is, if you know you only have a certain time budget, would this affect the optimal reduction factor?
- Finally, though our transmittance estimator is unbiased, it also increases variance, especially in dense media. It would be interesting to see if an unbiased but smoother estimator for transmittance is possible.

Limitations & Future Work

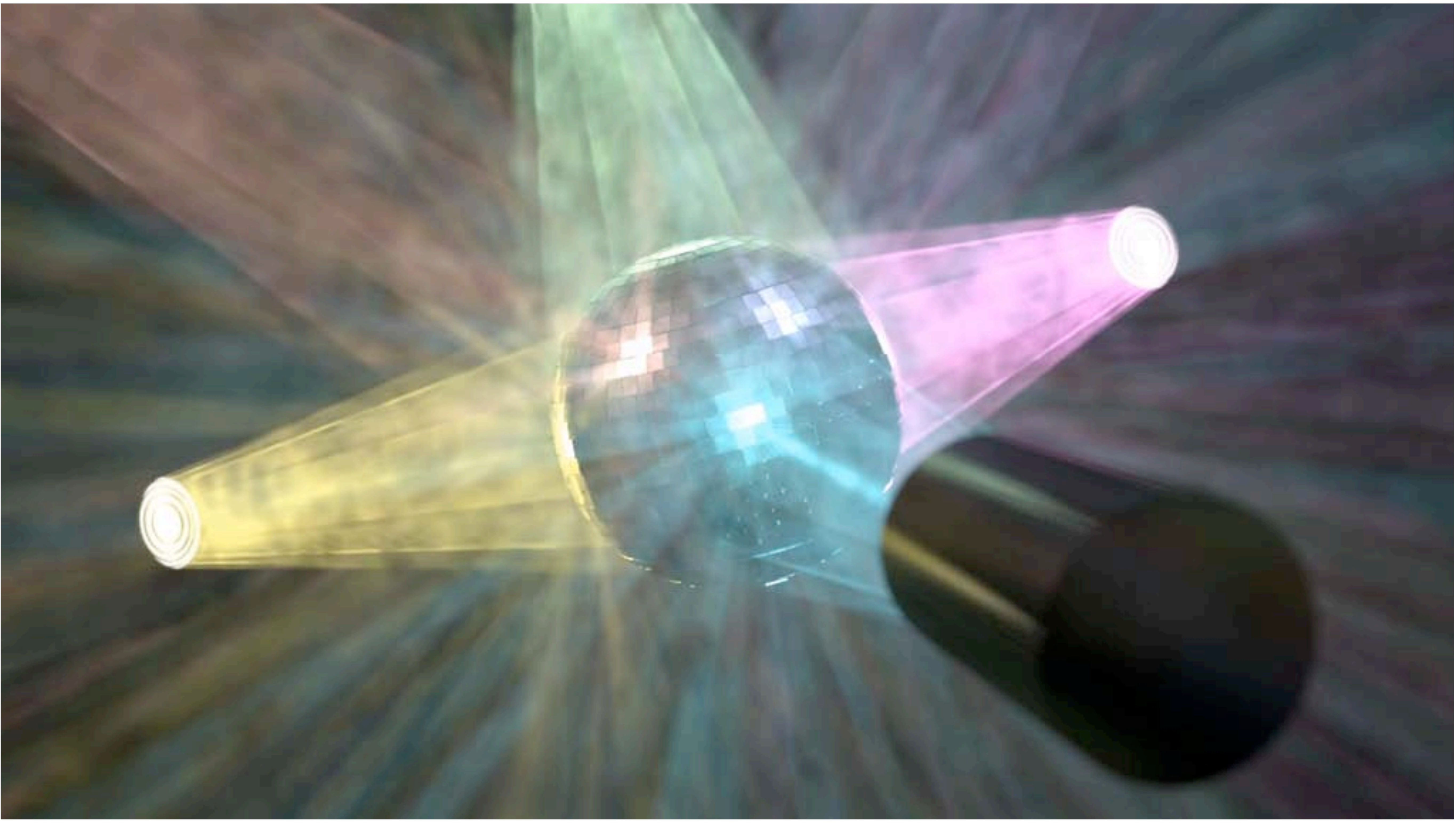
- Multiple scattering (relatively) costly
- Single α not optimal for entire image
  - adaptive α possible?
- Radius reduction for finite time budget?
- Unbiased transmittance noisier (dense media)
  - smooth and unbiased transmittance?

- There are of course some limitations
- Firstly, photon beams are fantastic at reconstructing caustic light paths, but multiple scattering effects can still be quite costly
- Also, though our approach is convergent with any alpha between 0 and 1, this is fixed for the entire image, it might be possible to adapt alpha in different regions of the image to accelerate convergence
- Our theory tells you how to reduce the radii if you want an unbiased image in the limit. However, an interesting practical question is, if you know you only have a certain time budget, would this affect the optimal reduction factor?
- Finally, though our transmittance estimator is unbiased, it also increases variance, especially in dense media. It would be interesting to see if an unbiased but smoother estimator for transmittance is possible.

Acknowledgements

- Anonymous SIGGRAPH reviewers
- ETH/DRZ internal reviewers
- Xin Sun
- Bruce Walter
- Wenzel Jakob
- Derek Bradley

Thank You



Practical Improvements: User Parameters

- Goal: single user parameter to control bias/variance

- Now, on the more practical side, our goal was to have a single parameter to control the bias/variance tradeoff (as seen in the previous graphs) [click]
- However, in practice, the rate of convergence is influenced both by alpha, as well as the number of photons shot per pass. Let me illustrate this

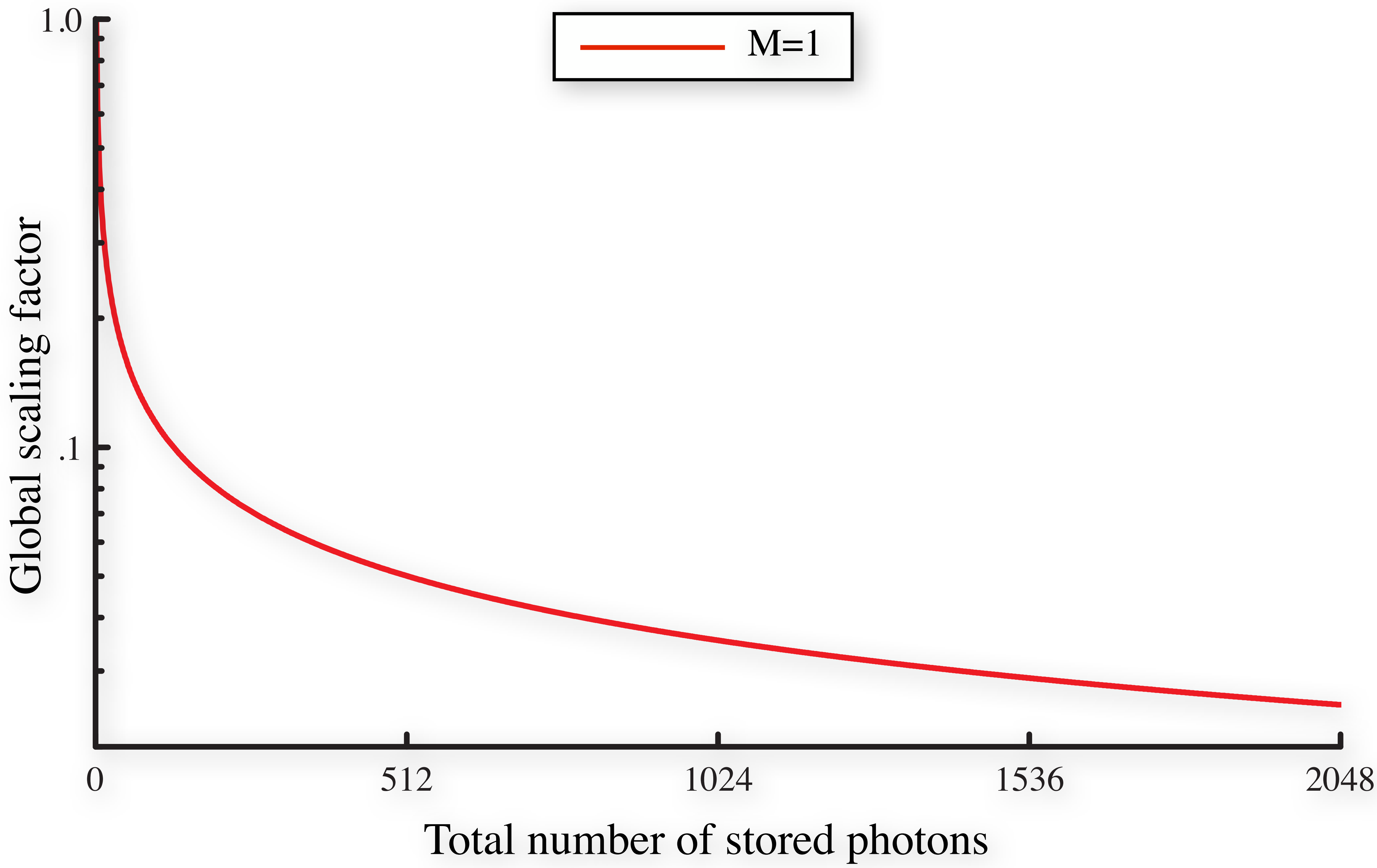
Practical Improvements: User Parameters

- Goal: single user parameter to control bias/variance
- α parameter and M (number of photons per pass) interdependent!

Thursday, 6 September 12

- Now, on the more practical side, our goal was to have a single parameter to control the bias/variance tradeoff (as seen in the previous graphs) [click]
- However, in practice, the rate of convergence is influenced both by alpha, as well as the number of photons shot per pass. Let me illustrate this

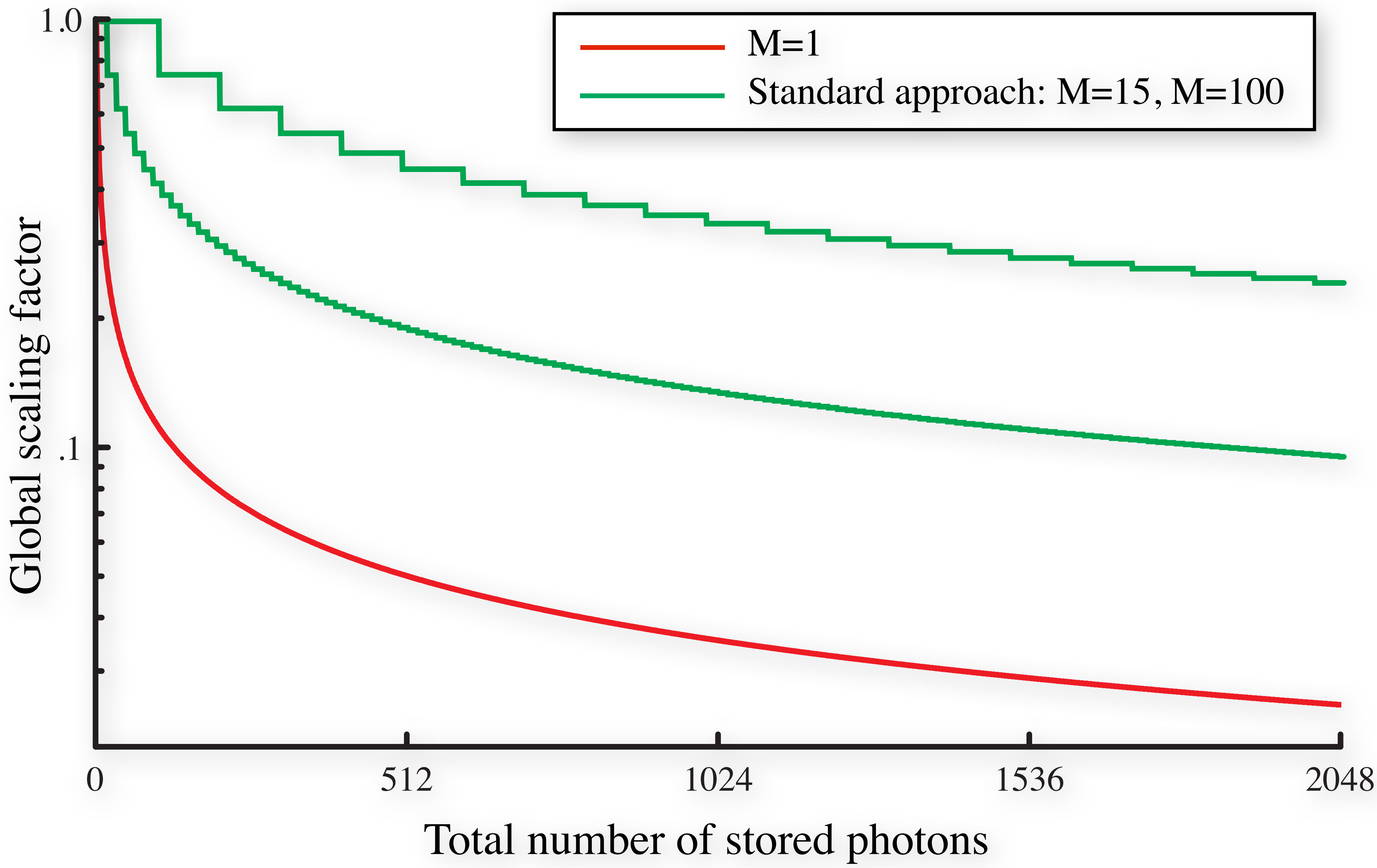
User Parameters



Thursday, 6 September 12

- If we shoot one photon beam per pass, this graph shows the radius reduction rate as a function of the total number of stored photons
- Now, if we don't change alpha and we store the same total number of beams, we would hope to obtain basically identical results.
- With the standard approach, used by all previous PPM techniques, this is not the case

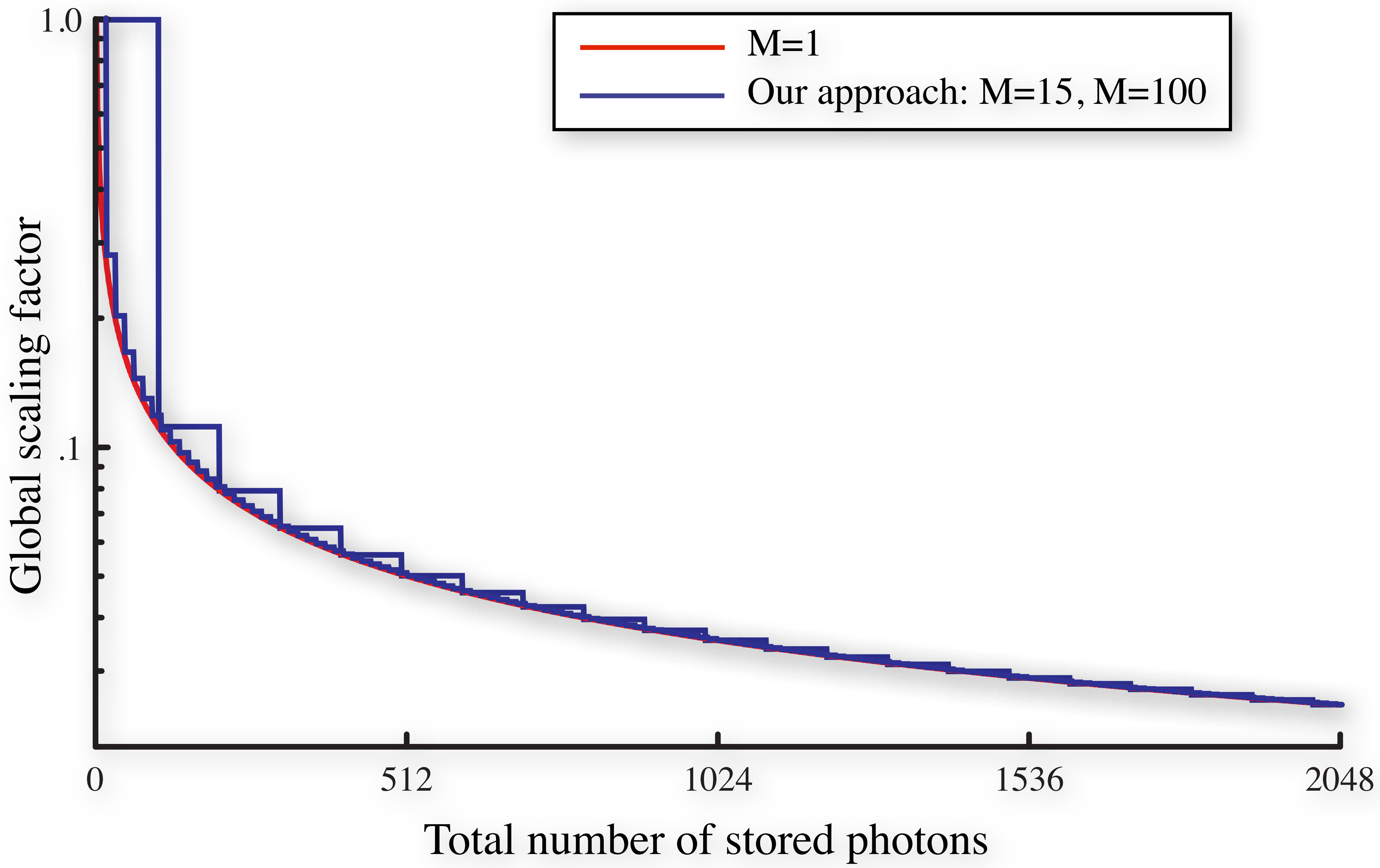
User Parameters



Thursday, 6 September 12

- Since the radius reduction factor is applied at the granularity of the passes, the final image will look significantly different if we decided to show incremental updates every 1 beam, every 10 beams, or every 100 beams.
- We make a very simple modifications which eliminates this problem

User Parameters



Thursday, 6 September 12

- At the end of each pass, we apply a radius update for every stored beam.
- This means that regardless of the number of beams per pass, we obtain very similar results, so we can choose the display frequency without worrying about modifying the other parameters.
- This makes it much more intuitive to scale the algorithm to different interactivity settings.