Theory, Analysis, and Applications of 2D Global Illumination

Wojciech Jarosz[1,2] Volker Schönefeld[2,3,4]
Leif Kobbelt[4] Henrik Wann Jensen[2]

1 Disney Research, Zurich 2 UC San Diego 3 LIMBIC 4 RWTH AACHEN UNIVERSITY

Association for Computing Machinery
ACM Transactions on Graphics
SIGGRAPH 2012

Wednesday, 5 September 12

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination

Global Illumination

Wednesday, 5 September 12

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination
- Which is applicable in a wide range of disciplines, ranging from architecture and industrial design, to obviously the entertainment industry

Global Illumination

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination
- Which is applicable in a wide range of disciplines, ranging from architecture and industrial design, to obviously the entertainment industry
- One of the core challenges with global illumination however, is that its complicated
- Our goal in this paper is to help understand it, teach it, and make it computationally more tractable

Global Illumination

Architecture/Industrial Design

Wednesday, 5 September 12

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination
- Which is applicable in a wide range of disciplines, ranging from architecture and industrial design, to obviously the entertainment industry
- One of the core challenges with global illumination however, is that its complicated
- Our goal in this paper is to help understand it, teach it, and make it computationally more tractable

Global Illumination

Architecture/Industrial Design

Entertainment Industry


[Tabellion & Lamorlette 04]

Wednesday, 5 September 12

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination
- Which is applicable in a wide range of disciplines, ranging from architecture and industrial design, to obviously the entertainment industry
- One of the core challenges with global illumination however, is that its complicated
- Our goal in this paper is to help understand it, teach it, and make it computationally more tractable

Global Illumination

Architecture/Industrial Design

Entertainment Industry

- Complex!

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination
- Which is applicable in a wide range of disciplines, ranging from architecture and industrial design, to obviously the entertainment industry
- One of the core challenges with global illumination however, is that its complicated
- Our goal in this paper is to help understand it, teach it, and make it computationally more tractable

Global Illumination

Architecture/Industrial Design

Entertainment Industry

- Complex!
- Understand it, teach it, make it faster

- Thanks for the introduction, and thank you all for coming to my talk
- As the title implies, this paper deals with global illumination
- Which is applicable in a wide range of disciplines, ranging from architecture and industrial design, to obviously the entertainment industry
- One of the core challenges with global illumination however, is that its complicated
- Our goal in this paper is to help understand it, teach it, and make it computationally more tractable

Overview: Theory of 2D Global Illumination

- Simplify by going to 2D

- To deal with this complexity, we simplify the problem by going to 2D
- We derive a full theory of light transport in 2D, resulting in a 2D rendering equation
- There are a number of benefits to this:
  - Firstly, visualizing quantities related to GI becomes significantly easier, since we can often just plot them as 1D graphs
  - Secondly, rendering become significantly faster in 2D, making rapid prototyping and experimentation possible
  - Also, since the problem becomes simplified it makes it easier to analyze mathematically
  - And finally, teaching global illumination in 2D is simpler, providing a stepping-stone for teaching full 3D global illumination

Overview: Theory of 2D Global Illumination

- Simplify by going to 2D
  - 2D rendering equation

- To deal with this complexity, we simplify the problem by going to 2D
- We derive a full theory of light transport in 2D, resulting in a 2D rendering equation
- There are a number of benefits to this:
  - Firstly, visualizing quantities related to GI becomes significantly easier, since we can often just plot them as 1D graphs
  - Secondly, rendering become significantly faster in 2D, making rapid prototyping and experimentation possible
  - Also, since the problem becomes simplified it makes it easier to analyze mathematically
  - And finally, teaching global illumination in 2D is simpler, providing a stepping-stone for teaching full 3D global illumination

Overview: Theory of 2D Global Illumination

- Simplify by going to 2D
  - 2D rendering equation
  - Benefits:
    - Visualizing

- To deal with this complexity, we simplify the problem by going to 2D
- We derive a full theory of light transport in 2D, resulting in a 2D rendering equation
- There are a number of benefits to this:
  - Firstly, visualizing quantities related to GI becomes significantly easier, since we can often just plot them as 1D graphs
  - Secondly, rendering become significantly faster in 2D, making rapid prototyping and experimentation possible
  - Also, since the problem becomes simplified it makes it easier to analyze mathematically
  - And finally, teaching global illumination in 2D is simpler, providing a stepping-stone for teaching full 3D global illumination

Overview: Theory of 2D Global Illumination

- Simplify by going to 2D
  - 2D rendering equation
  - Benefits:
    - Visualizing
    - Experimenting

- To deal with this complexity, we simplify the problem by going to 2D
- We derive a full theory of light transport in 2D, resulting in a 2D rendering equation
- There are a number of benefits to this:
  - Firstly, visualizing quantities related to GI becomes significantly easier, since we can often just plot them as 1D graphs
  - Secondly, rendering become significantly faster in 2D, making rapid prototyping and experimentation possible
  - Also, since the problem becomes simplified it makes it easier to analyze mathematically
  - And finally, teaching global illumination in 2D is simpler, providing a stepping-stone for teaching full 3D global illumination

Overview: Theory of 2D Global Illumination

- Simplify by going to 2D
  - 2D rendering equation
  - Benefits:
    - Visualizing
    - Experimenting
    - Deriving

- To deal with this complexity, we simplify the problem by going to 2D
- We derive a full theory of light transport in 2D, resulting in a 2D rendering equation
- There are a number of benefits to this:
  - Firstly, visualizing quantities related to GI becomes significantly easier, since we can often just plot them as 1D graphs
  - Secondly, rendering become significantly faster in 2D, making rapid prototyping and experimentation possible
  - Also, since the problem becomes simplified it makes it easier to analyze mathematically
  - And finally, teaching global illumination in 2D is simpler, providing a stepping-stone for teaching full 3D global illumination

Overview: Theory of 2D Global Illumination

- Simplify by going to 2D
  - 2D rendering equation
  - Benefits:
    - Visualizing
    - Experimenting
    - Deriving
    - Teaching

- To deal with this complexity, we simplify the problem by going to 2D
- We derive a full theory of light transport in 2D, resulting in a 2D rendering equation
- There are a number of benefits to this:
  - Firstly, visualizing quantities related to GI becomes significantly easier, since we can often just plot them as 1D graphs
  - Secondly, rendering become significantly faster in 2D, making rapid prototyping and experimentation possible
  - Also, since the problem becomes simplified it makes it easier to analyze mathematically
  - And finally, teaching global illumination in 2D is simpler, providing a stepping-stone for teaching full 3D global illumination

Overview: Analysis & Applications to 3D

- Analyze rendering algorithms

- To demonstrate the benefit of a 2D theory, show how to easily analyze algorithms like photon mapping in 2D
- Also, we perform an in-depth 2nd order analysis of global illumination in 2D
- And we show how the insights gained can lead to practical improvements for 3D rendering using irradiance caching

Overview: Analysis & Applications to 3D

- Analyze rendering algorithms
- Second-order analysis of global illumination in 2D

- To demonstrate the benefit of a 2D theory, show how to easily analyze algorithms like photon mapping in 2D
- Also, we perform an in-depth 2nd order analysis of global illumination in 2D
- And we show how the insights gained can lead to practical improvements for 3D rendering using irradiance caching

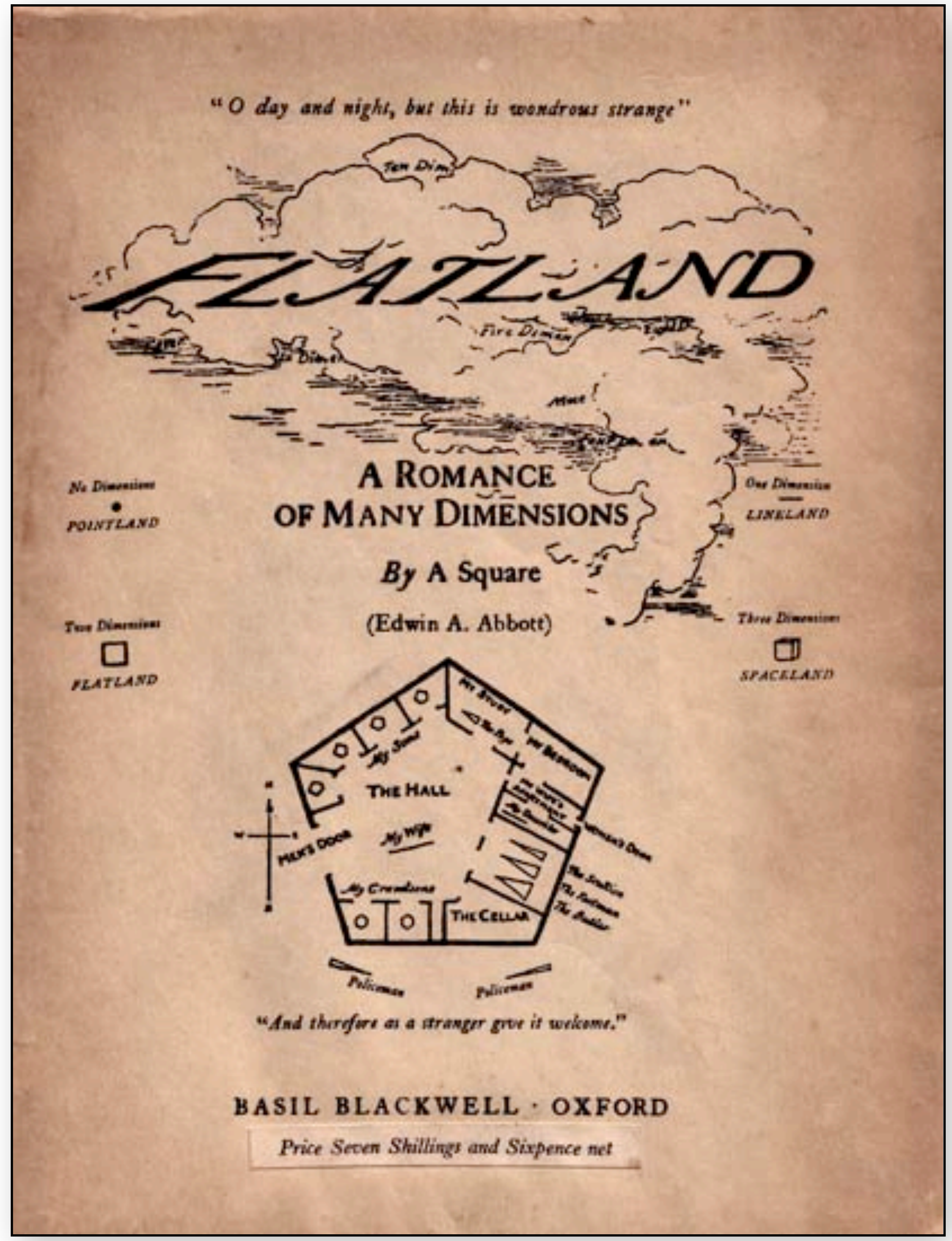
Overview: Analysis & Applications to 3D

- Analyze rendering algorithms
- Second-order analysis of global illumination in 2D
- Apply lessons learned to 3D
  - Improve irradiance caching

- To demonstrate the benefit of a 2D theory, show how to easily analyze algorithms like photon mapping in 2D
- Also, we perform an in-depth 2nd order analysis of global illumination in 2D
- And we show how the insights gained can lead to practical improvements for 3D rendering using irradiance caching

Previous Work (1) - 2D World

- Flatland: A Romance of Many Dimensions [Abbot 1884]



- The use of 2D is not new. An early detailed description of a 2D world was provided in Abbot in the late 1800s in his novella: Flatland. This term was later adopted by graphics researchers when analyzing algorithms in 2D

Previous Work (2) - 2D Ray Casting

- Wolfenstein 3-D [1992]
  - 2D ray casting

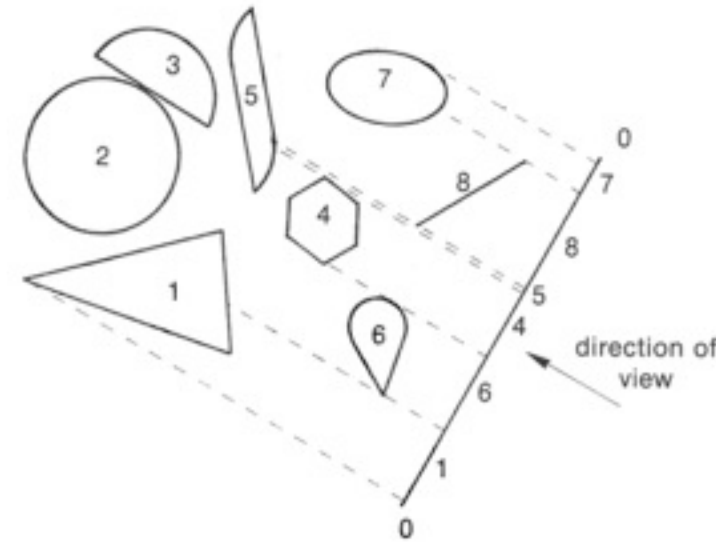
- 2D simplifications have also been applied in practical contexts. An early example of this in the game industry was with Wolfenstein 3D, which used a 2D ray casting algorithm to render its pseudo-3D world.

Previous Work (3) - 2D Light Transport

- 2D light transport has also been considered in isolated cases within academia
- Researchers have used this simplified domain to analyze (click) hidden surface elimination, (click) Radiosity, (click) and perform frequency and gradient space analyses.
- We are inspired by this line of research, but, while these methods considers 2D for isolated problems, we wish to provide a more holistic description of 2D light transport by deriving and analyzing a full 2D rendering equation

Previous Work (3) - 2D Light Transport

- Hidden surface removal
  [Edelsbrunner et al. 83] [Pocchiola 90]

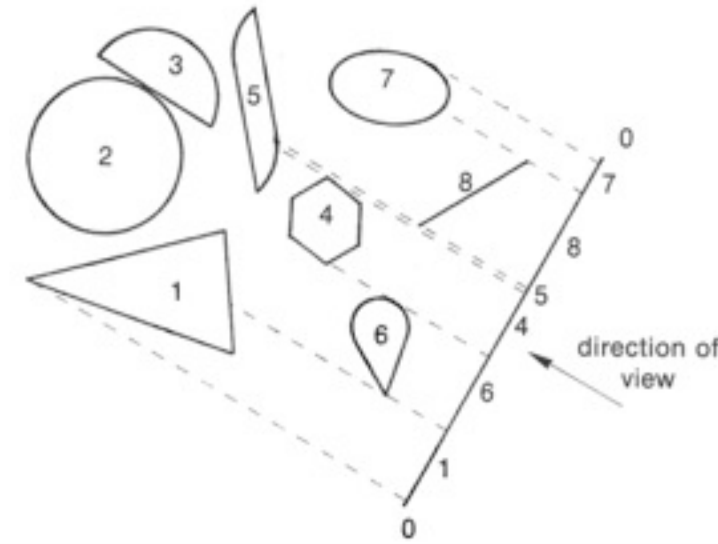


Wednesday, 5 September 12

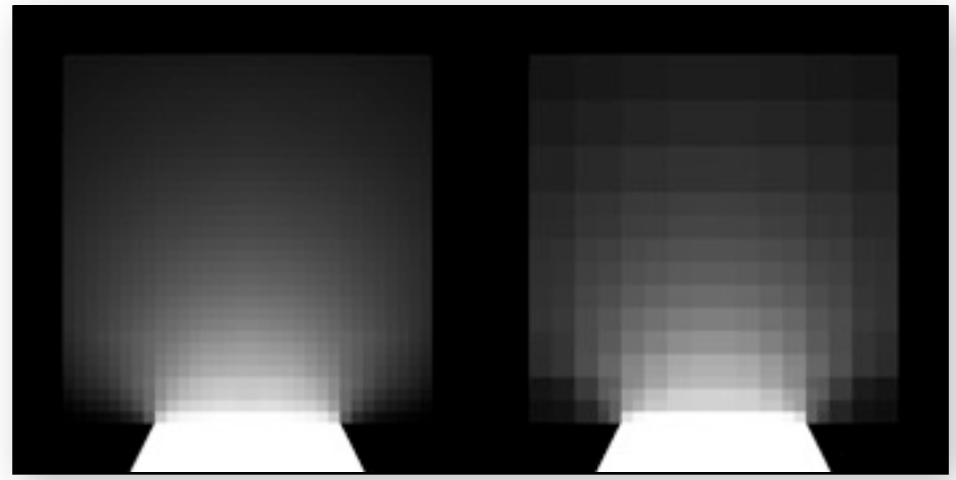
- 2D light transport has also been considered in isolated cases within academia
- Researchers have used this simplified domain to analyze (click) hidden surface elimination, (click) Radiosity, (click) and perform frequency and gradient space analyses.
- We are inspired by this line of research, but, while these methods considers 2D for isolated problems, we wish to provide a more holistic description of 2D light transport by deriving and analyzing a full 2D rendering equation

Previous Work (3) - 2D Light Transport

- Hidden surface removal
  [Edelsbrunner et al. 83] [Pocchiola 90]



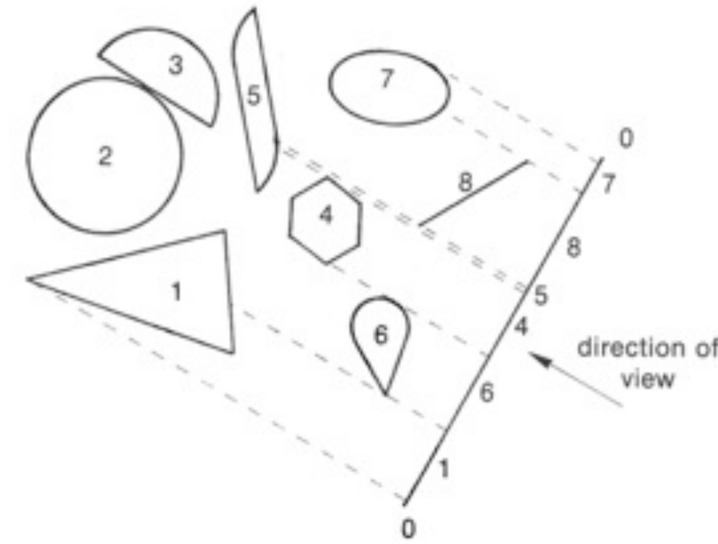
- Radiosity
  [Heckbert 92] [Gortler et al. 93]
  [Orti et al. 96] [Durand et al. 96]



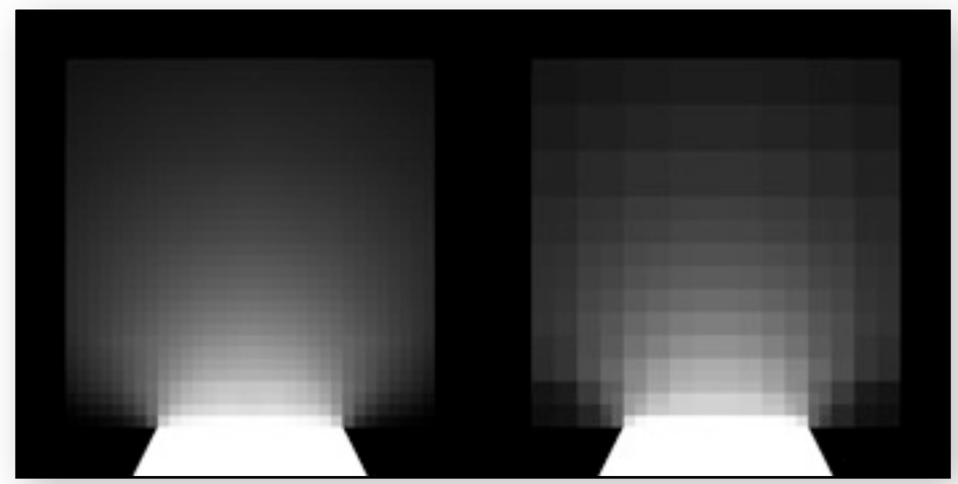
- 2D light transport has also been considered in isolated cases within academia
- Researchers have used this simplified domain to analyze (click) hidden surface elimination, (click) Radiosity, (click) and perform frequency and gradient space analyses.
- We are inspired by this line of research, but, while these methods considers 2D for isolated problems, we wish to provide a more holistic description of 2D light transport by deriving and analyzing a full 2D rendering equation

Previous Work (3) - 2D Light Transport

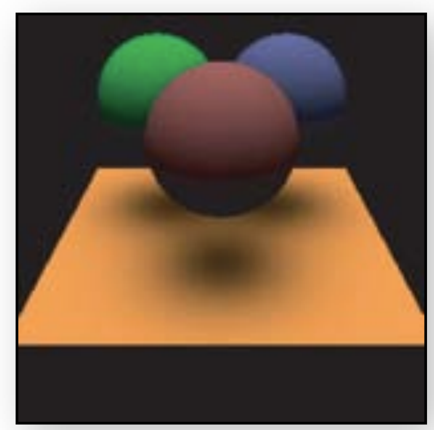
- Hidden surface removal
  [Edelsbrunner et al. 83] [Pocchiola 90]

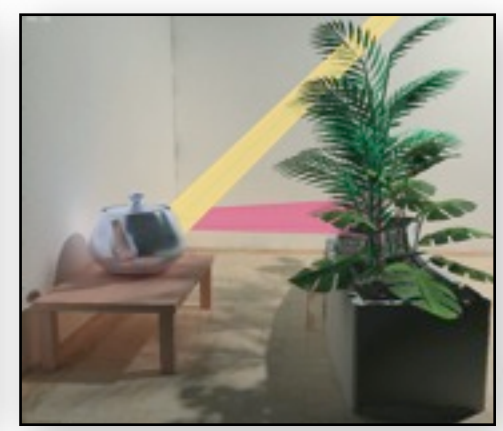


- Radiosity
  [Heckbert 92] [Gortler et al. 93]
  [Orti et al. 96] [Durand et al. 96]



- Frequency/Gradient-space analysis
  [Durand et al. 05]
  [Ramamoorthi et al. 07]





- 2D light transport has also been considered in isolated cases within academia
- Researchers have used this simplified domain to analyze (click) hidden surface elimination, (click) Radiosity, (click) and perform frequency and gradient space analyses.
- We are inspired by this line of research, but, while these methods considers 2D for isolated problems, we wish to provide a more holistic description of 2D light transport by deriving and analyzing a full 2D rendering equation

Theory of 2D Light Transport

- **Intrinsic 2D Model**:
  - self-contained 2D world, composed of curves

- Before we begin, we need to define our 2D world
- We assume a true 2D intrinsic model where the world is composed of curves, and all light is emitted, reflected and absorbed within the 2D world.
- We want the final theory to be highly analogous to 3D so that is can provide practical insights for rendering. We therefore derive it in analogy to 3D, and not from first-principles

Theory of 2D Light Transport

- **Intrinsic 2D Model**:
  - self-contained 2D world, composed of curves
- Not derived from first-principles, but in *analogy* to 3D model

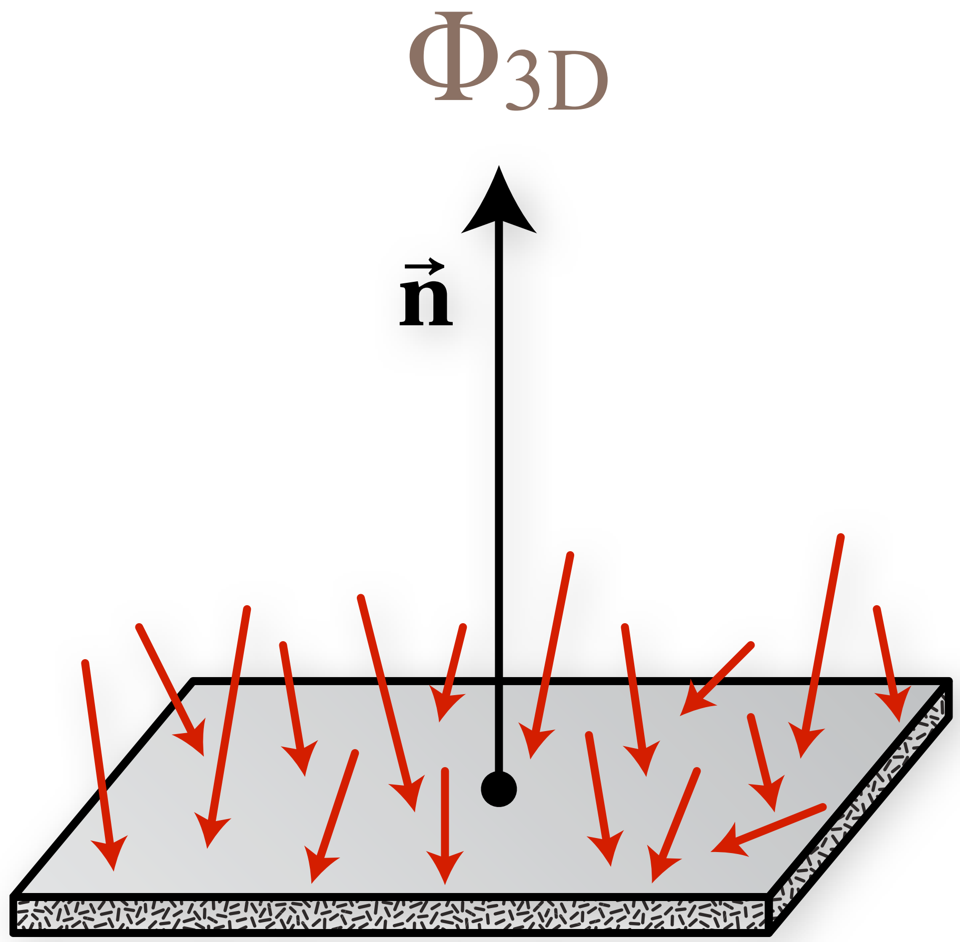
- Before we begin, we need to define our 2D world
- We assume a true 2D intrinsic model where the world is composed of curves, and all light is emitted, reflected and absorbed within the 2D world.
- We want the final theory to be highly analogous to 3D so that is can provide practical insights for rendering. We therefore derive it in analogy to 3D, and not from first-principles

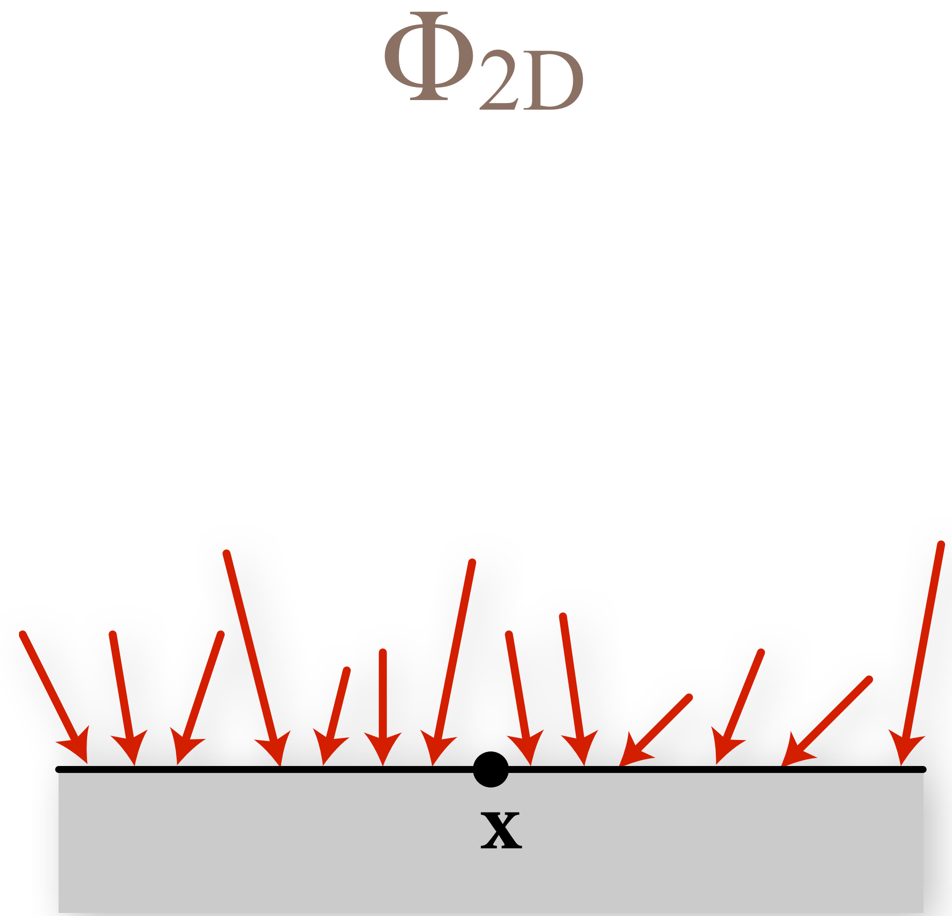
Radiometry

- Assume light consists of photons
- Define basic quantities by “counting photons”

- We start by assuming that light consists of photons and define 2D equivalents to common radiometric quantities

Flux (Power)



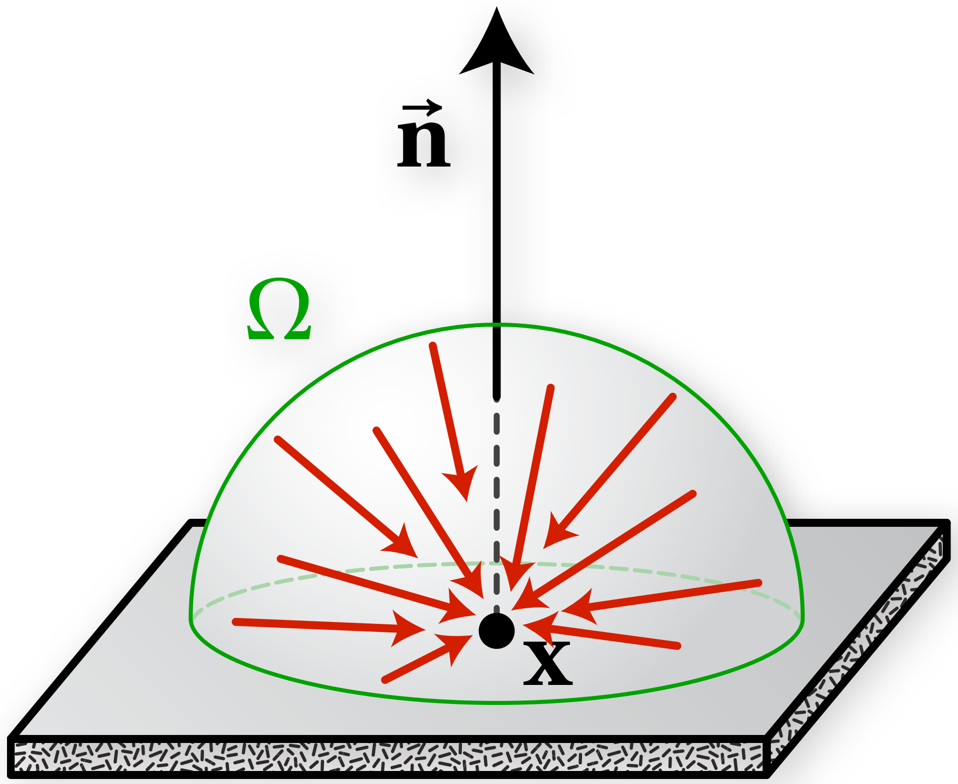


Units: [W = J / s]

- Flux is the total amount of energy passing through a surface (3D) or curve (2D) per unit time
- In both the 3D and 2D world it has units of Watts (Joules per second) since it effectively counts the number of photons hitting a wall per second

Irradiance

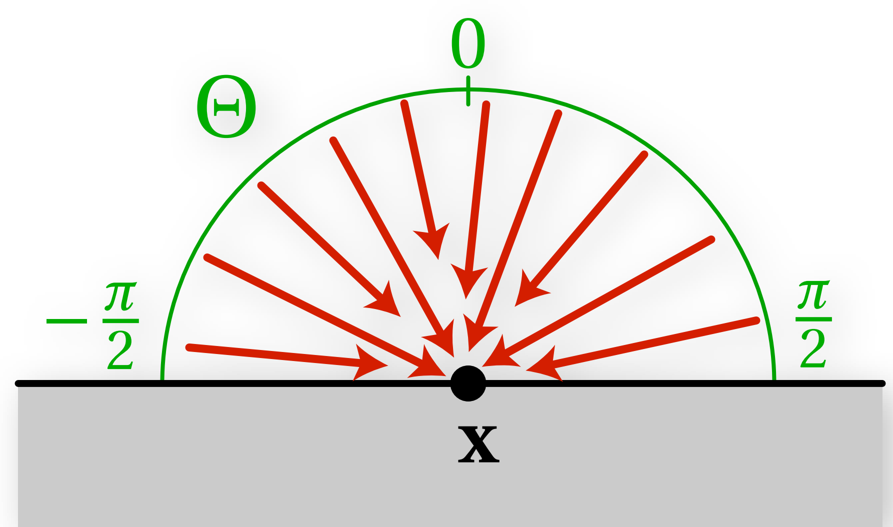
$E_{3D}(\mathbf{x})$



Units: [W / m^2]

- flux per unit ***area*** arriving at a ***surface***

$E_{2D}(\mathbf{x})$



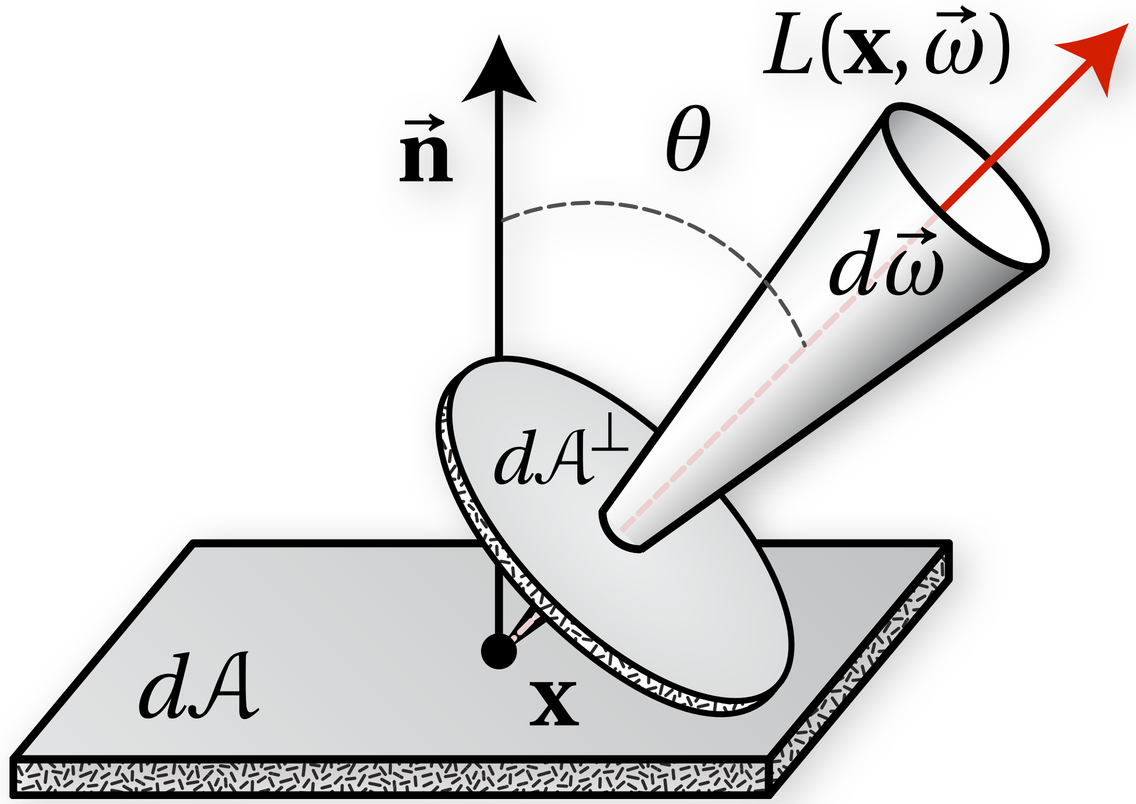
Units: [W / m]

- flux per unit ***arc length*** arriving at a ***curve***

- In 3D, irradiance is the flux density per unit surface area
- In 2D, surface area turns into arc length, so irradiance is the flux per unit arc length arriving at a curve
- This changes the units of irradiance
- In both cases, irradiance effectively counts the photons that arrive at an infinitesimal patch on a wall, from all directions

Irradiance

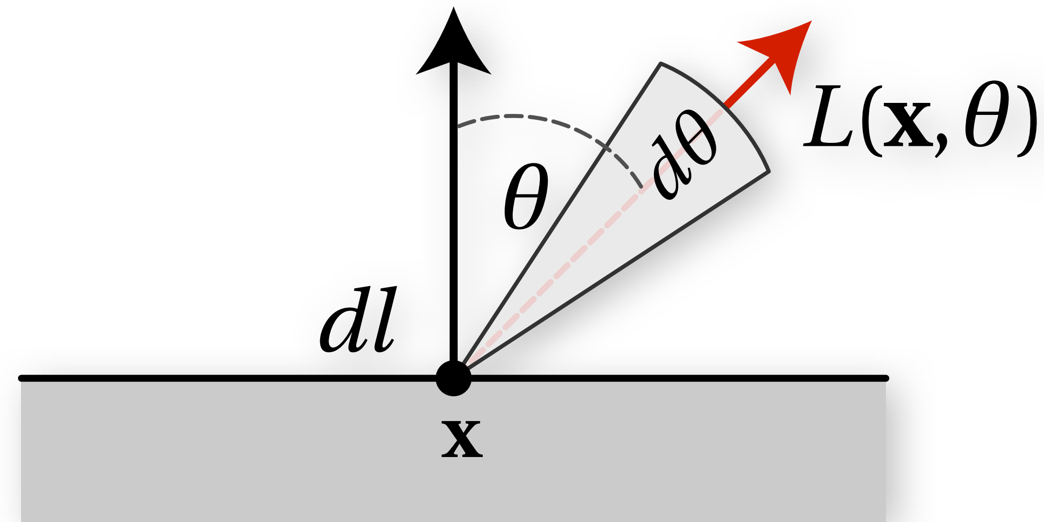
$L_{3D}(\mathbf{x},\omega)$



Units: [W / sr / m^2]

- flux density per unit **solid angle**, per perp. unit **area**

$L_{2D}(\mathbf{x},\theta)$



Units: [W / rad / m]

- flux density per unit **angle**, per perp. unit **arc length**

- Radiance restricts this even further, and only considers photons from a certain differential set of directions
- In 3D, it has units of W / sr / m2, since it is the flux density per unit solid angle per perpendicular unit area
- In 2D, solid angles completely disappear, giving us the flux density per unit angle, per perpendicular arc length

Radiance Integrals

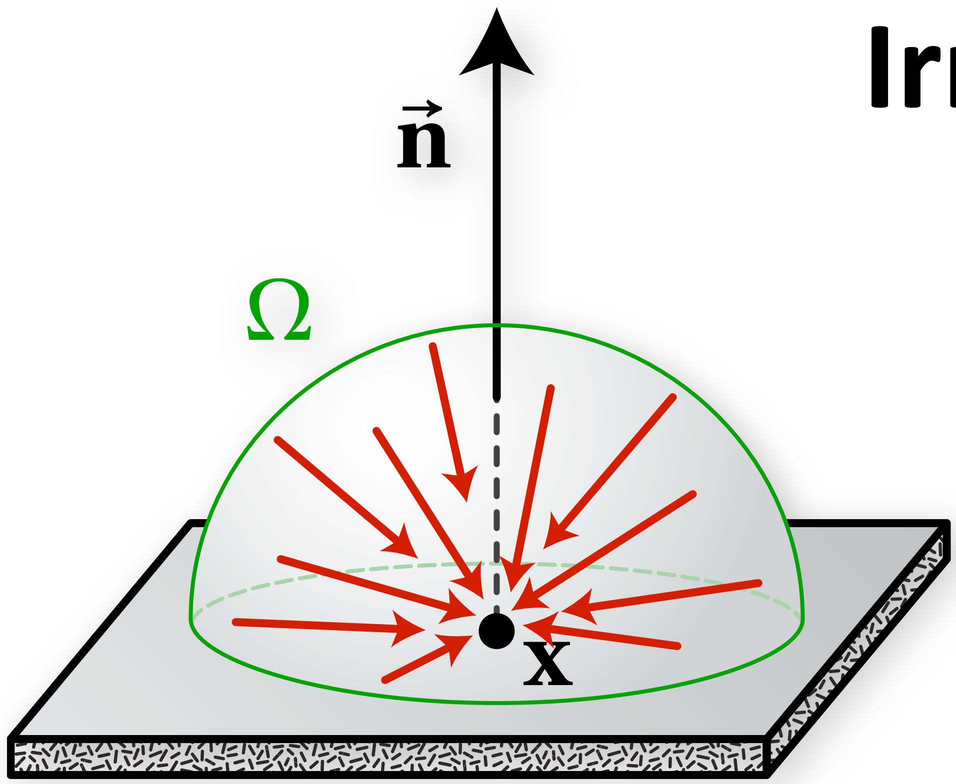
- Other radiometric quantities can be expressed in terms of radiance

- Just like in 3D, we can derive other radiometric quantities from Radiance
- For instance, in 3D irradiance is the 2D integral of the cosine-weighted radiance over the hemisphere, while in 2D this is a 1D integral over the hemicircle

Radiance Integrals

- Other radiometric quantities can be expressed in terms of radiance

Irradiance



$$E_{3D}(\mathbf{x}) = \int_{\boxed{\Omega}} L_{3D}(\mathbf{x} \leftarrow \vec{\omega}) \cos\theta \boxed{d\vec{\omega}}$$

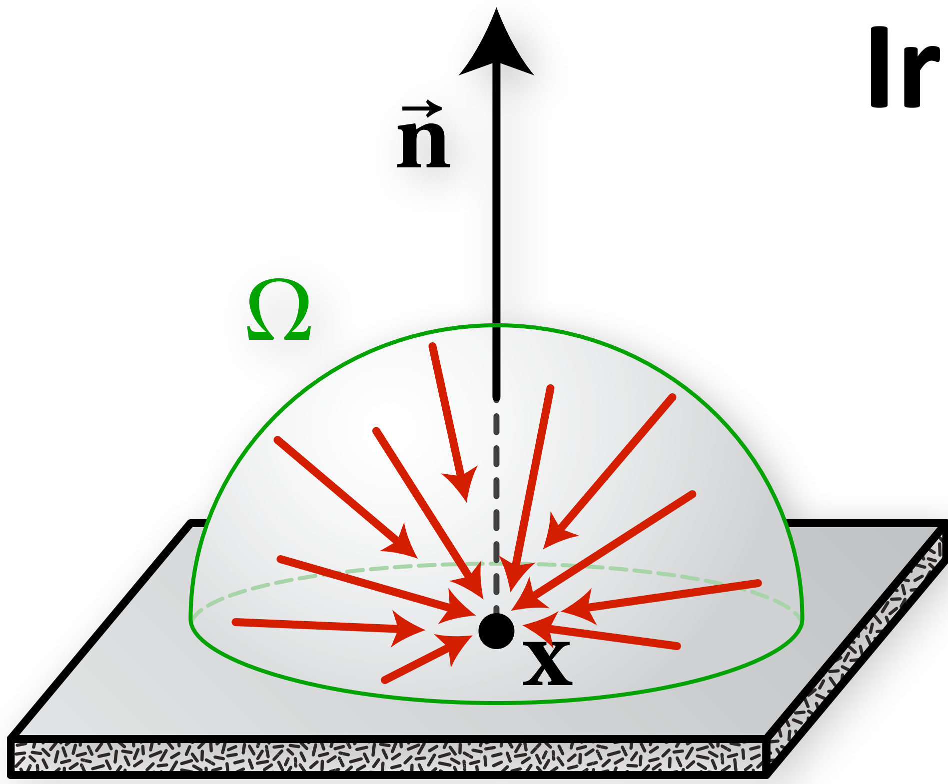
integrate over **hemisphere**

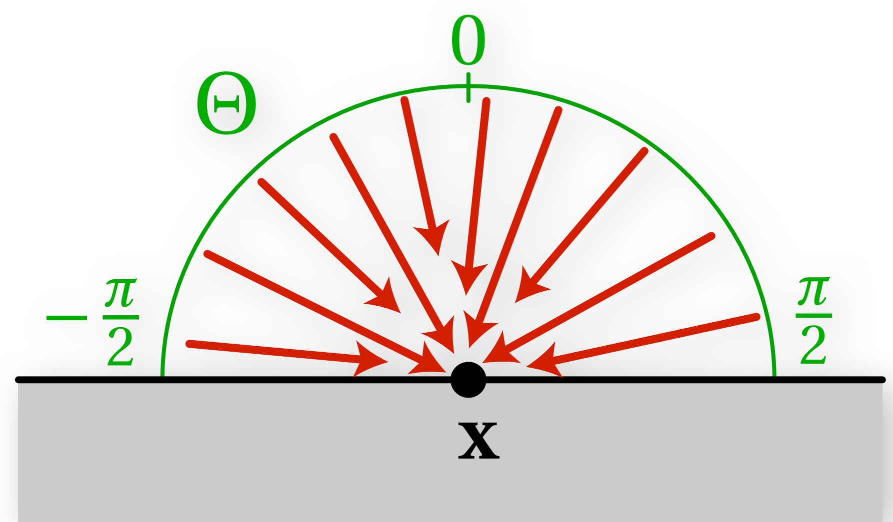
- Just like in 3D, we can derive other radiometric quantities from Radiance
- For instance, in 3D irradiance is the 2D integral of the cosine-weighted radiance over the hemisphere, while in 2D this is a 1D integral over the hemicircle

Radiance Integrals

- Other radiometric quantities can be expressed in terms of radiance

Irradiance





$$E_{3D}(\mathbf{x}) = \int_{\boxed{\Omega}} L_{3D}(\mathbf{x} \leftarrow \vec{\omega}) \cos\theta \boxed{d\vec{\omega}}$$

integrate over **hemisphere**

$$E_{2D}(\mathbf{x}) = \int_{\boxed{\Theta}} L_{2D}(\mathbf{x} \leftarrow \theta) \cos\theta \boxed{d\,\theta}$$

integrate over **hemicircle**

- Just like in 3D, we can derive other radiometric quantities from Radiance
- For instance, in 3D irradiance is the 2D integral of the cosine-weighted radiance over the hemisphere, while in 2D this is a 1D integral over the hemicircle

Radiance Discussion

- Different complexity:

- An important difference in 2D and 3D is in the complexity of the resulting radiometric functions
- For instance, in a 3D world, radiance is a 5D function, 3 for position, and 2 for direction
- By just moving down 1 dimension to a 2D world, radiance simplifies to a 3D function (2 position, 1 direction)
- This reduction has a significant impact on the convergence of rendering algorithms

Radiance Discussion

- Different complexity:
  - Radiance in 3D is a 5D function

3D position

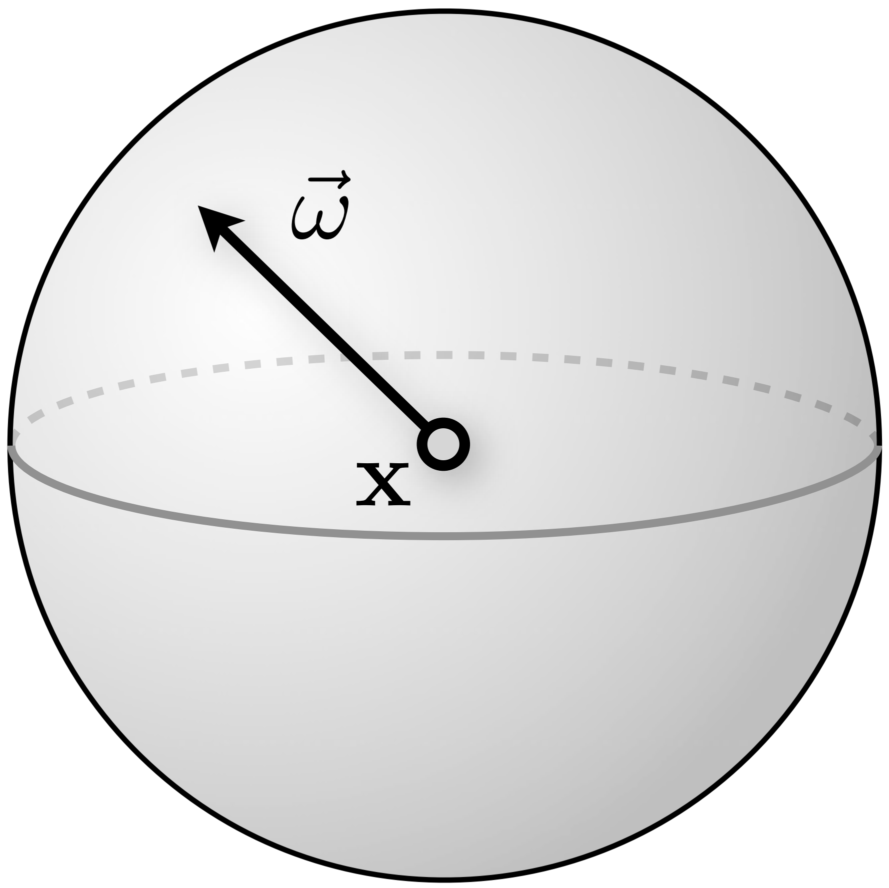
$\mathbf{x}$

- An important difference in 2D and 3D is in the complexity of the resulting radiometric functions
- For instance, in a 3D world, radiance is a 5D function, 3 for position, and 2 for direction
- By just moving down 1 dimension to a 2D world, radiance simplifies to a 3D function (2 position, 1 direction)
- This reduction has a significant impact on the convergence of rendering algorithms

Radiance Discussion

- Different complexity:
  - Radiance in 3D is a 5D function

3D position
2D direction



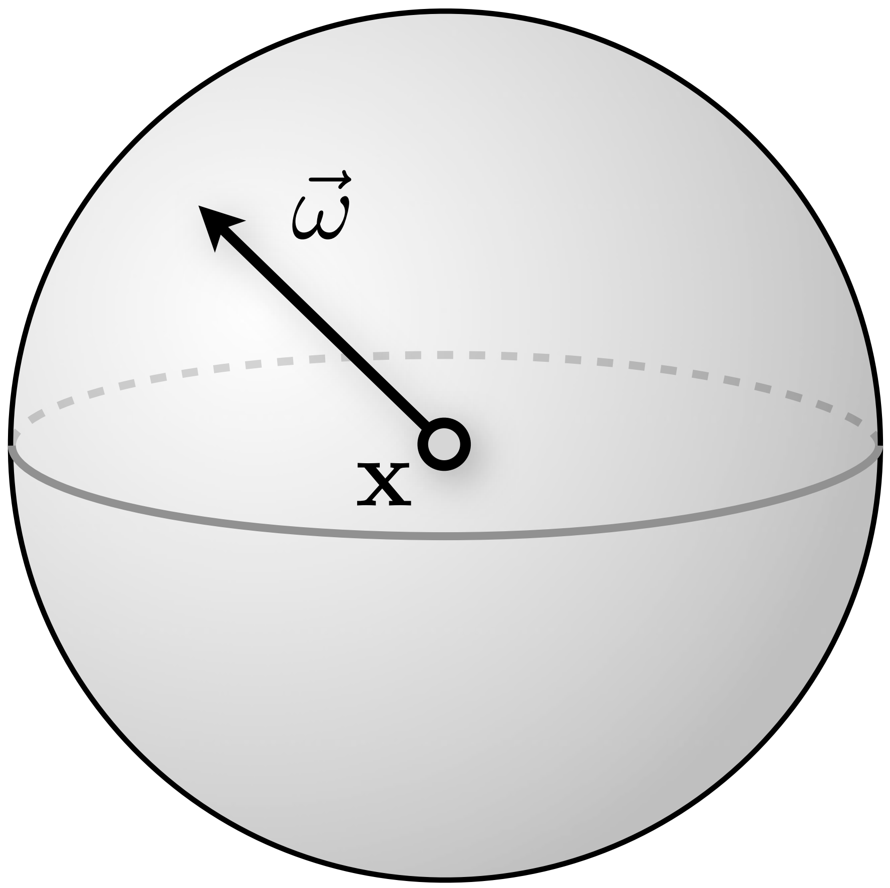
Wednesday, 5 September 12

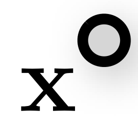
- An important difference in 2D and 3D is in the complexity of the resulting radiometric functions
- For instance, in a 3D world, radiance is a 5D function, 3 for position, and 2 for direction
- By just moving down 1 dimension to a 2D world, radiance simplifies to a 3D function (2 position, 1 direction)
- This reduction has a significant impact on the convergence of rendering algorithms

Radiance Discussion

- Different complexity:
  - Radiance in 3D is a 5D function
  - Radiance in 2D is only a 3D function

3D position
2D direction





2D position

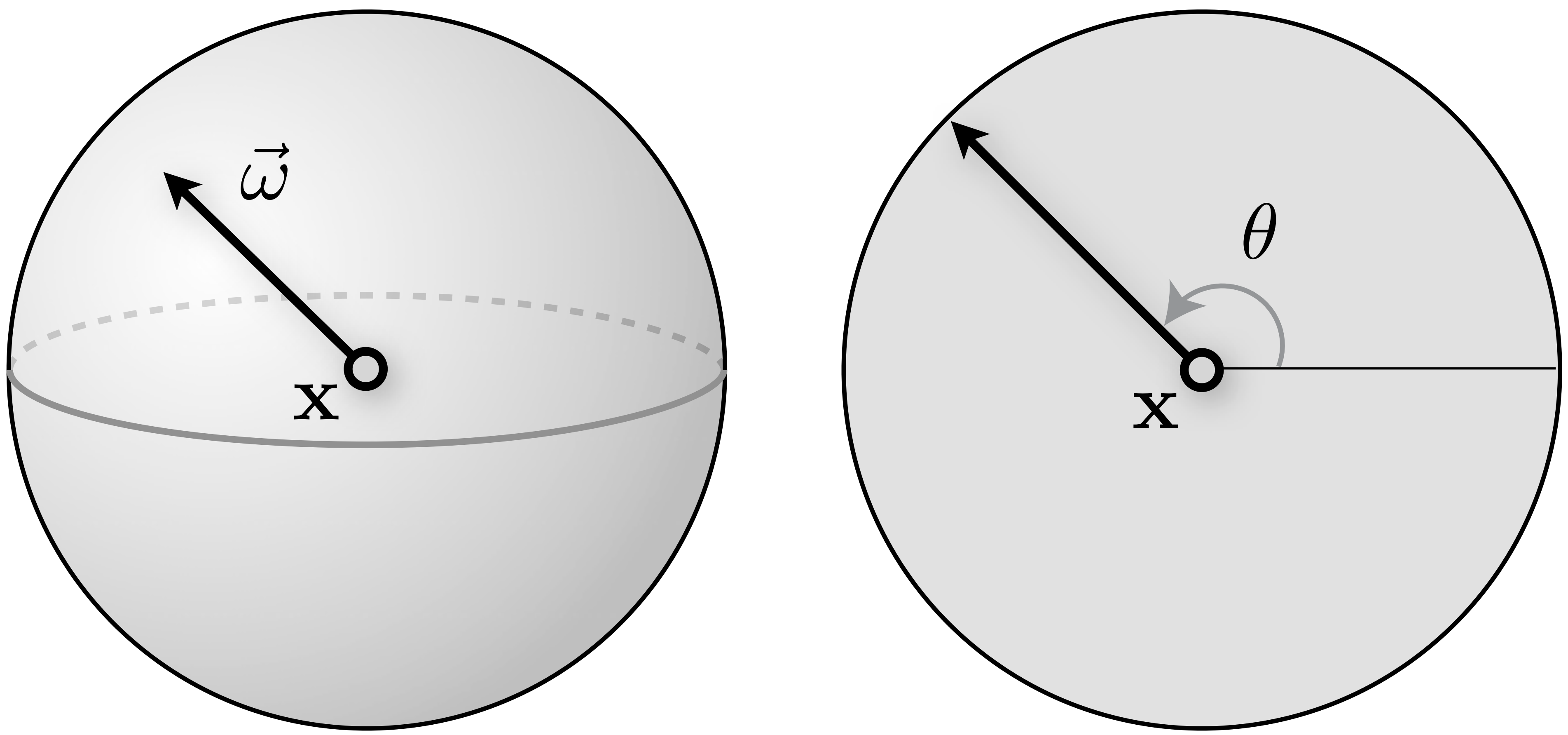
Wednesday, 5 September 12

- An important difference in 2D and 3D is in the complexity of the resulting radiometric functions
- For instance, in a 3D world, radiance is a 5D function, 3 for position, and 2 for direction
- By just moving down 1 dimension to a 2D world, radiance simplifies to a 3D function (2 position, 1 direction)
- This reduction has a significant impact on the convergence of rendering algorithms

Radiance Discussion

- Different complexity:
  - Radiance in 3D is a 5D function
  - Radiance in 2D is only a 3D function

3D position
2D direction



2D position
1D direction

Wednesday, 5 September 12

- An important difference in 2D and 3D is in the complexity of the resulting radiometric functions
- For instance, in a 3D world, radiance is a 5D function, 3 for position, and 2 for direction
- By just moving down 1 dimension to a 2D world, radiance simplifies to a 3D function (2 position, 1 direction)
- This reduction has a significant impact on the convergence of rendering algorithms

The BRDF

- Notation: $f_{3D}(\mathbf{x}, \vec{\omega} \to \vec{\omega}')$ or $f_{2D}(\mathbf{x}, \theta \to \theta')$
- Conceptually like in 3D, but with **important differences**

- To complete our theory we also need an analogy to the BRDF, but there are some important differences

The BRDF

- **Domain:**
  - **3D**: six-dimensional function (2 pos, 2 in-dir, 2 out-dir)
  - **2D**: three-dimensional function (1 pos, 1 in-dir, 1 out-dir)

- Just as with the radiance function, the BRDF becomes significantly simplified in a 2D world.
- It goes from being a 6D function to just a 3D function
- Other than this, the BRDF effectively works analogously to 3D

The BRDF

- **Domain:**
  - **3D**: six-dimensional function (2 pos, 2 in-dir, 2 out-dir)
  - **2D**: three-dimensional function (1 pos, 1 in-dir, 1 out-dir)
- **Range:** $[0, \infty)$
- **Reciprocity**
- **Energy Conservation**
- **Specular interactions**: Snell/Fresnel/mirror unchanged

- Just as with the radiance function, the BRDF becomes significantly simplified in a 2D world.
- It goes from being a 6D function to just a 3D function
- Other than this, the BRDF effectively works analogously to 3D with reciprocity, energy conservation, etc.

Reflection & Rendering Equation

- Relation between incident/reflected light

- The BRDF also allows us to define the relationship between the incident and reflect light, giving us the reflection & ultimately rendering equations
- In both 2D and 3D, to compute reflected radiance, we simply integrate the BRDF, light, visibility function, and geometry term over all points in the scene (be that over curves or over surfaces)
- However, there is an important conceptual difference is in the geometry term
- In 3D we have the familiar inverse-squared falloff; whereas in 2D this just inverse falloff
- This is because the light's wavefront expands along the surface area of a sphere in 3D, but along the perimeter of a circle in 2D

Reflection & Rendering Equation

- Relation between incident/reflected light
- Surface-area / arc-length formulation:

$$L^r_{3D}(\mathbf{x}\to\mathbf{e}) = \int_A f_{3D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\, L_{3D}(\mathbf{x}\leftarrow\mathbf{y}) V_{3D}(\mathbf{x}\leftrightarrow\mathbf{y})\, G_{3D}(\mathbf{x}\leftrightarrow\mathbf{y})\; da(\mathbf{y}),$$

$$L^r_{2D}(\mathbf{x}\to\mathbf{e}) = \int_{\mathfrak{L}} f_{2D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\, L_{2D}(\mathbf{x}\leftarrow\mathbf{y}) V_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})\, G_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})\; dl(\mathbf{y}),$$

- The BRDF also allows us to define the relationship between the incident and reflect light, giving us the reflection & ultimately rendering equations
- In both 2D and 3D, to compute reflected radiance, we simply integrate the BRDF, light, visibility function, and geometry term over all points in the scene (be that over curves or over surfaces)
- However, there is an important conceptual difference is in the geometry term
- In 3D we have the familiar inverse-squared falloff; whereas in 2D this just inverse falloff
- This is because the light's wavefront expands along the surface area of a sphere in 3D, but along the perimeter of a circle in 2D

Reflection & Rendering Equation

- Relation between incident/reflected light
- Surface-area / arc-length formulation:

$$L^r_{3D}(\mathbf{x}\rightarrow\mathbf{e}) = \int_A f_{3D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\, L_{3D}(\mathbf{x}\leftarrow\mathbf{y}) V_{3D}(\mathbf{x}\leftrightarrow\mathbf{y}) \boxed{G_{3D}(\mathbf{x}\leftrightarrow\mathbf{y})}\, da(\mathbf{y}),$$

$$L^r_{2D}(\mathbf{x}\rightarrow\mathbf{e}) = \int_{\mathfrak{L}} f_{2D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\, L_{2D}(\mathbf{x}\leftarrow\mathbf{y}) V_{2D}(\mathbf{x}\leftrightarrow\mathbf{y}) \boxed{G_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})}\, dl(\mathbf{y}),$$

- The BRDF also allows us to define the relationship between the incident and reflect light, giving us the reflection & ultimately rendering equations
- In both 2D and 3D, to compute reflected radiance, we simply integrate the BRDF, light, visibility function, and geometry term over all points in the scene (be that over curves or over surfaces)
- However, there is an important conceptual difference is in the geometry term
- In 3D we have the familiar inverse-squared falloff; whereas in 2D this just inverse falloff
- This is because the light's wavefront expands along the surface area of a sphere in 3D, but along the perimeter of a circle in 2D

Reflection & Rendering Equation

- Relation between incident/reflected light
- Surface-area / arc-length formulation:

$$L^r_{3D}(\mathbf{x}\to\mathbf{e}) = \int_A f_{3D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\, L_{3D}(\mathbf{x}\leftarrow\mathbf{y}) V_{3D}(\mathbf{x}\leftrightarrow\mathbf{y}) \boxed{G_{3D}(\mathbf{x}\leftrightarrow\mathbf{y})}\, da(\mathbf{y}),$$

$$L^r_{2D}(\mathbf{x}\to\mathbf{e}) = \int_{\mathfrak{L}} f_{2D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\, L_{2D}(\mathbf{x}\leftarrow\mathbf{y}) V_{2D}(\mathbf{x}\leftrightarrow\mathbf{y}) \boxed{G_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})}\, dl(\mathbf{y}),$$

$$G_{3D}(\mathbf{x}\leftrightarrow\mathbf{y}) = \frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{\boxed{\|\mathbf{x}-\mathbf{y}\|^2}} \qquad G_{2D}(\mathbf{x}\leftrightarrow\mathbf{y}) = \frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{\boxed{\|\mathbf{x}-\mathbf{y}\|}}$$

- The BRDF also allows us to define the relationship between the incident and reflect light, giving us the reflection & ultimately rendering equations
- In both 2D and 3D, to compute reflected radiance, we simply integrate the BRDF, light, visibility function, and geometry term over all points in the scene (be that over curves or over surfaces)
- However, there is an important conceptual difference is in the geometry term
- In 3D we have the familiar inverse-squared falloff; whereas in 2D this just inverse falloff
- This is because the light's wavefront expands along the surface area of a sphere in 3D, but along the perimeter of a circle in 2D

2D Rendering Algorithms & Results

- Framework for experimentation/analysis

- Having a full 2D theory of light transport provides us a framework for easy experimentation and analysis
- We have many examples in the paper to demonstrate this, but due to time constraints I'll only focus on two here

2D Rendering Algorithms & Results

- Framework for experimentation/analysis
  - Ray tracing
  - Path tracing
  - Photon mapping
  - Irradiance caching

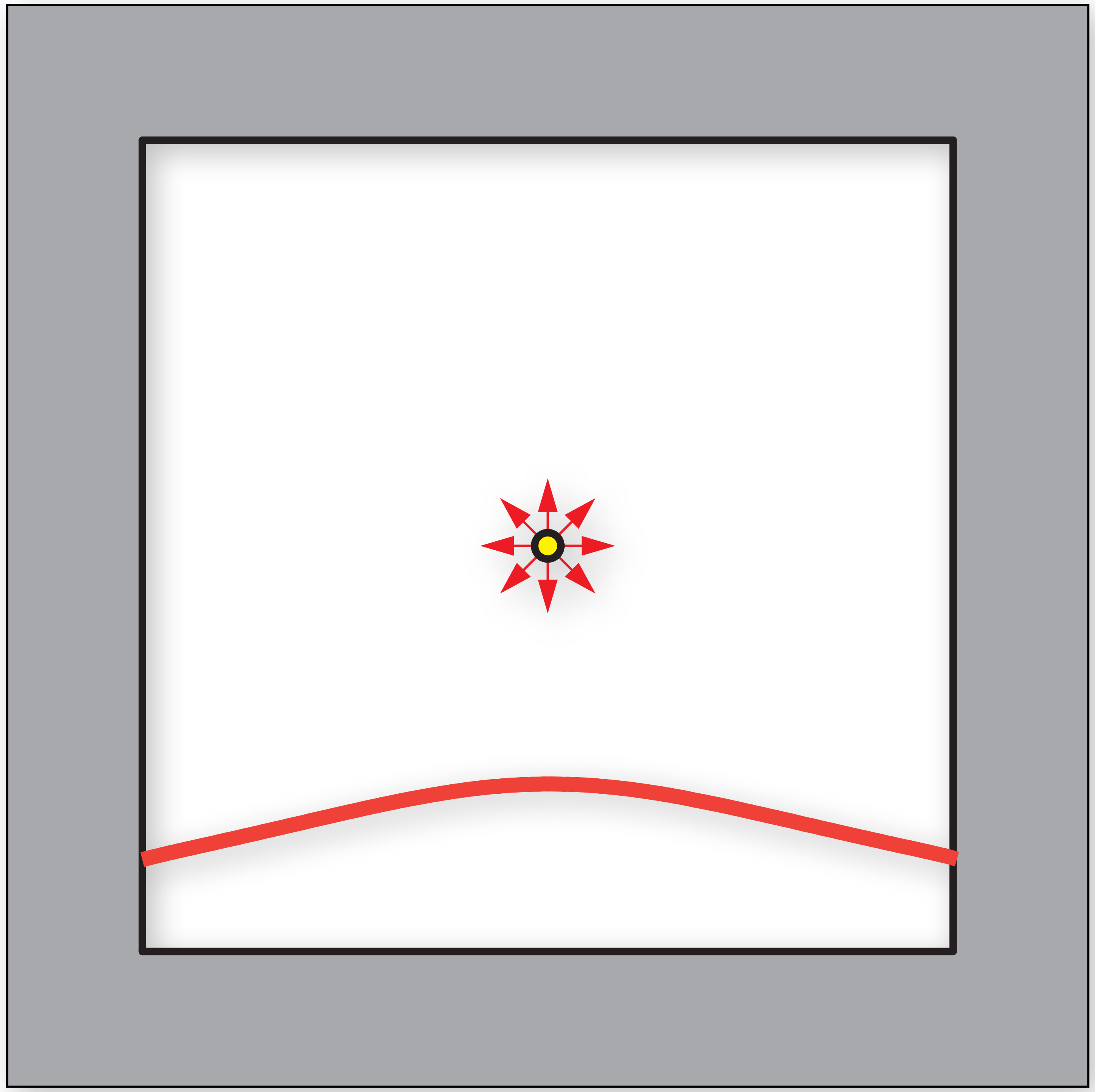
- Having a full 2D theory of light transport provides us a framework for easy experimentation and analysis
- We have many examples in the paper to demonstrate this, but due to time constraints I'll only focus on two here

2D Rendering Algorithms & Results

- Framework for experimentation/analysis
  - Ray tracing
  - Path tracing
  - **Photon mapping**
  - **Irradiance caching**

- Having a full 2D theory of light transport provides us a framework for easy experimentation and analysis
- We have many examples in the paper to demonstrate this, but due to time constraints I'll only focus on two here

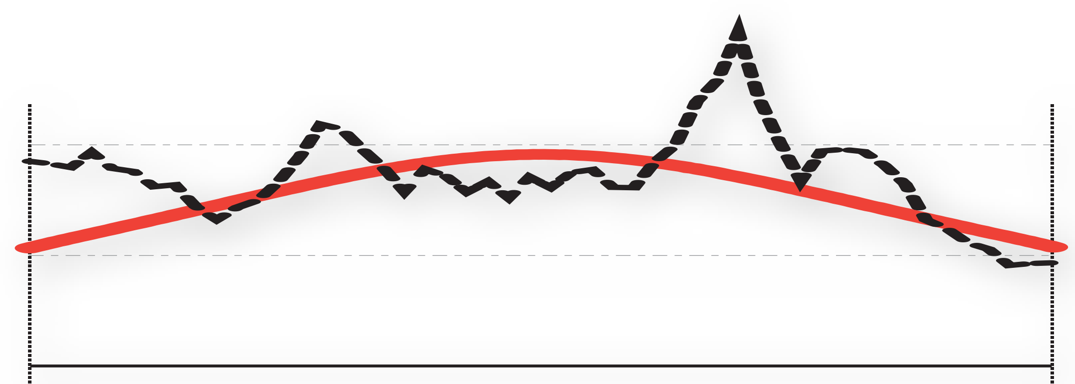
Photon Mapping



True Irradiance

- We can for instance analyze photon mapping in this simple 2D scene where I'm plotting the ground truth irradiance in red along the floor
- And this is the result of photon mapping with 100 photons, we can see the quality is quite poor
- And here we can see the impact of applying final gather to the photon map, we see that even with this low photon count, a single final gather pass can dramatically improve the quality

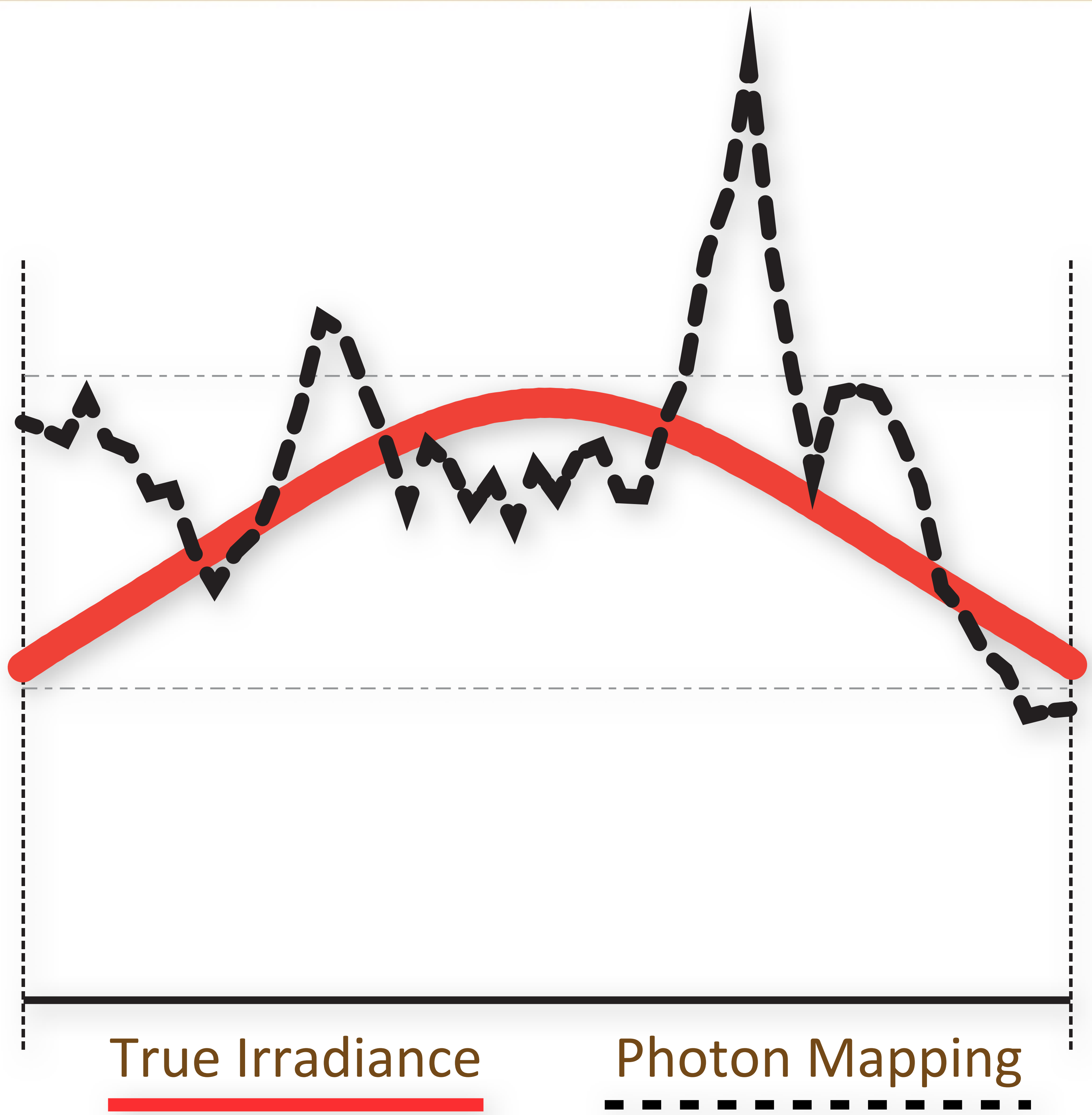
Photon Mapping



True Irradiance Photon Mapping

- We can for instance analyze photon mapping in this simple 2D scene where I'm plotting the ground truth irradiance in red along the floor
- And this is the result of photon mapping with 100 photons, we can see the quality is quite poor
- And here we can see the impact of applying final gather to the photon map, we see that even with this low photon count, a single final gather pass can dramatically improve the quality

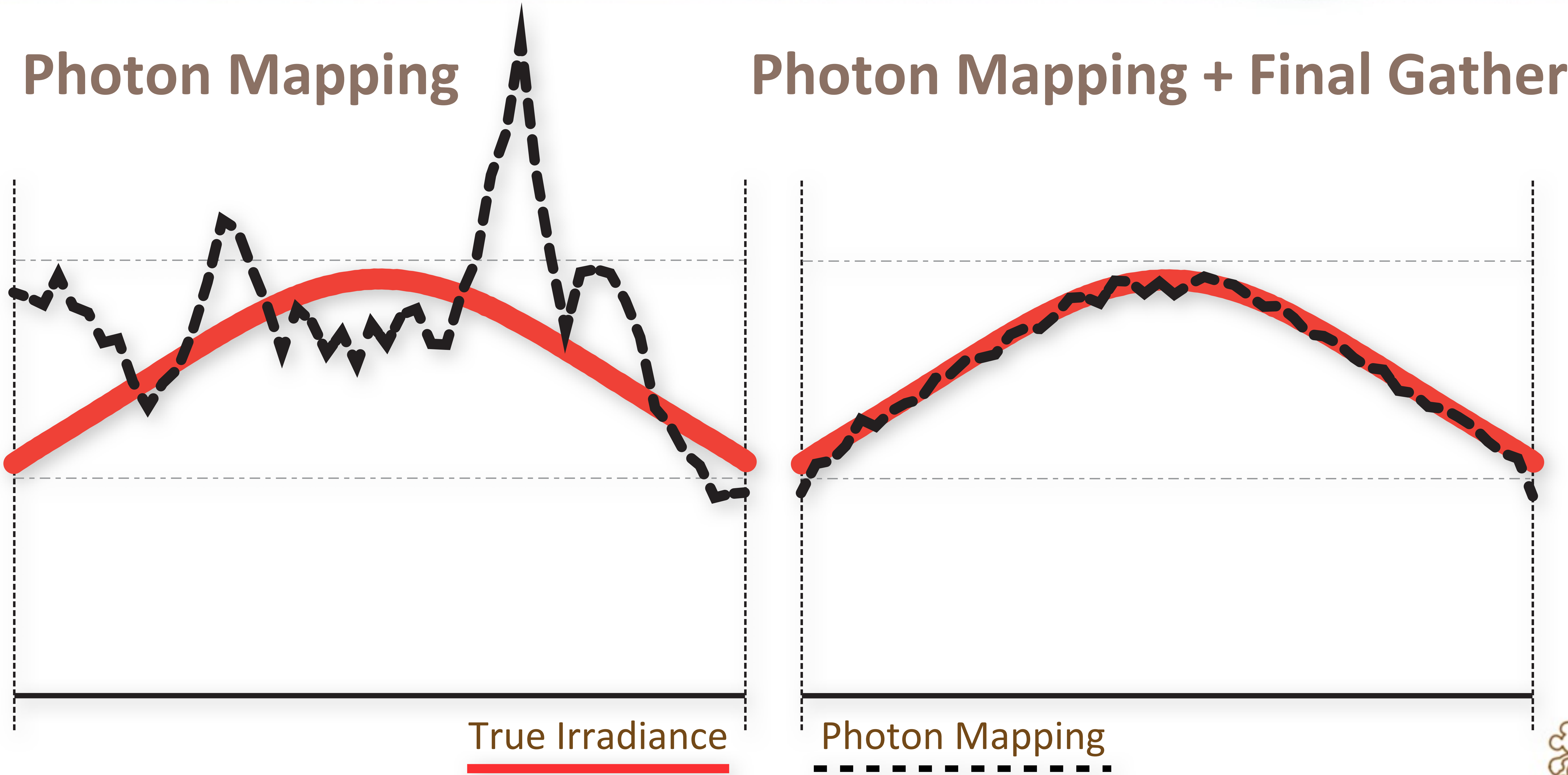
100 Photons



Wednesday, 5 September 12

- We can for instance analyze photon mapping in this simple 2D scene where I'm plotting the ground truth irradiance in red along the floor
- And this is the result of photon mapping with 100 photons, we can see the quality is quite poor
- And here we can see the impact of applying final gather to the photon map, we see that even with this low photon count, a single final gather pass can dramatically improve the quality

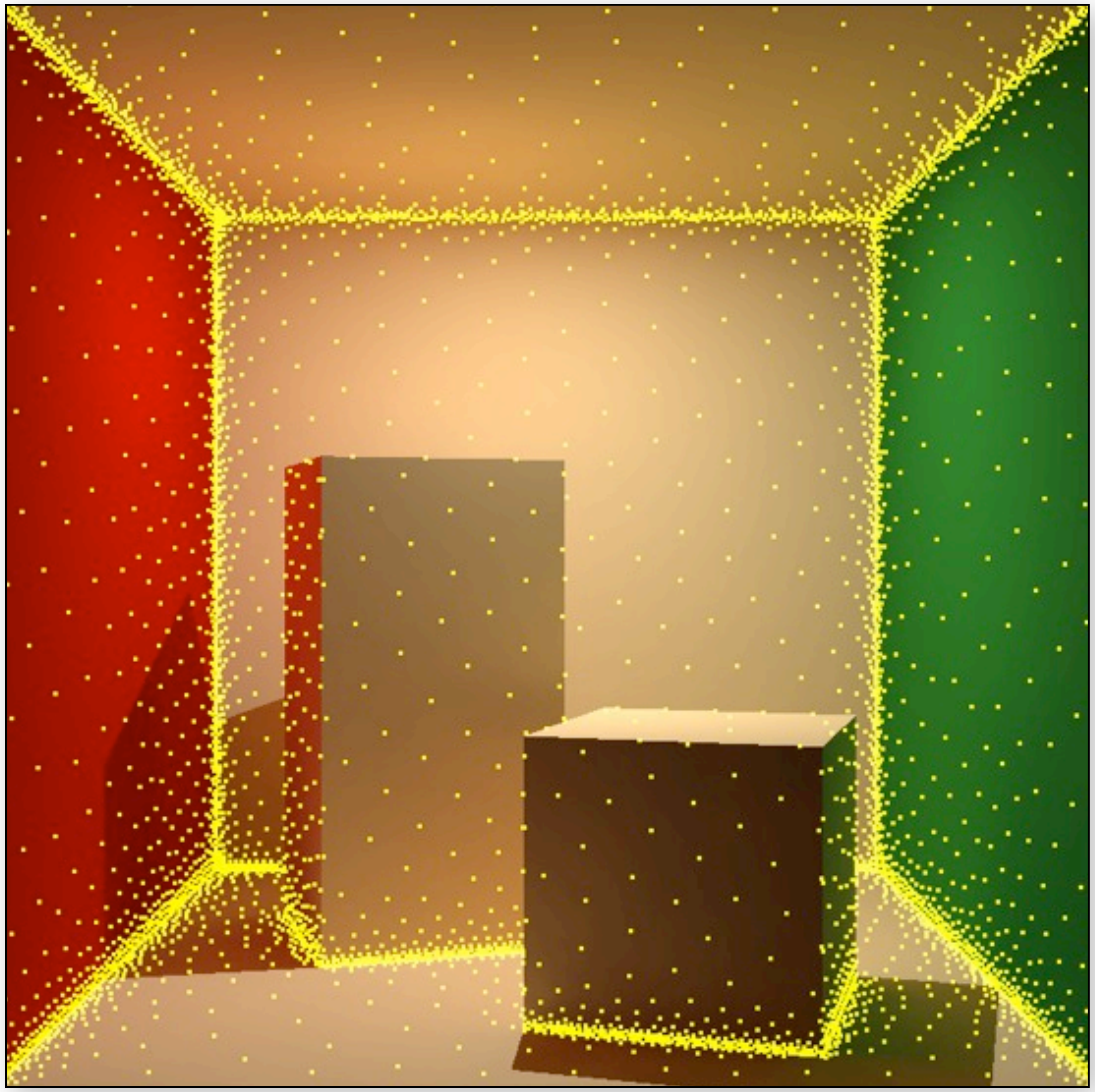
100 Photons



- We can for instance analyze photon mapping in this simple 2D scene where I’m plotting the ground truth irradiance in red along the floor
- And this is the result of photon mapping with 100 photons, we can see the quality is quite poor
- And here we can see the impact of applying final gather to the photon map, we see that even with this low photon count, a single final gather pass can dramatically improve the quality

Irradiance Caching

[Ward et al. in 1988]



Wednesday, 5 September 12

- We also perform a more in-depth analysis of irradiance caching.
- Ward and colleagues main insight was that in Lambertian scenes, the indirect irradiance changes slowly over surfaces, making it the perfect candidate for sparse sampling and interpolation

Irradiance Caching

- Major questions:
  - **Interpolation**: how to interpolate/extrapolate values
  - **Error control**: how to determine if values are “nearby”

- The key questions in irradiance caching are:
  - How do we interpolate/extrapolate from the cache values, and
  - How do we determine how far away we can keep re-using values
- It turns out that being able to quickly compute accurate irradiance derivatives can significantly improve both of these steps
- So we will use our 2D framework to perform an in-depth gradient-space analysis of 2D irradiance to see how it could improve irradiance caching

Irradiance Caching

- Major questions:
  - **Interpolation**: how to interpolate/extrapolate values
  - **Error control**: how to determine if values are “nearby”
- Accurate irradiance derivatives needed

Wednesday, 5 September 12

- The key questions in irradiance caching are:
  - How do we interpolate/extrapolate from the cache values, and
  - How do we determine how far away we can keep re-using values
- It turns out that being able to quickly compute accurate irradiance derivatives can significantly improve both of these steps
- So we will use our 2D framework to perform an in-depth gradient-space analysis of 2D irradiance to see how it could improve irradiance caching

Irradiance Caching

- Major questions:
  - **Interpolation**: how to interpolate/extrapolate values
  - **Error control**: how to determine if values are "nearby"
- Accurate irradiance derivatives needed
- Gradient analysis in 2D

- The key questions in irradiance caching are:
  - How do we interpolate/extrapolate from the cache values, and
  - How do we determine how far away we can keep re-using values
- It turns out that being able to quickly compute accurate irradiance derivatives can significantly improve both of these steps
- So we will use our 2D framework to perform an in-depth gradient-space analysis of 2D irradiance to see how it could improve irradiance caching

Gradient Analysis of 2D Irradiance

- Illumination Gradients
  [Ward and Heckbert 92] [Arvo 94]
  [Holzschuch et al. 95, 96, 98]
  [Annen t al. 04] [Krivanek et al. 05b]
  [Ramamoorthi et al. 07]
  [Jarosz et al. 08a, 08b]

- There has actually been a considerable amount of research on illumination gradients, and we will see how some of these can be re-derived much more easily in a 2D setting
- Furthermore, we will go a step further, and perform a 2nd-order analysis, and apply this to irradiance caching

Gradient Analysis of 2D Irradiance

- Illumination Gradients
  [Ward and Heckbert 92] [Arvo 94]
  [Holzschuch et al. 95, 96, 98]
  [Annen t al. 04] [Krivanek et al. 05b]
  [Ramamoorthi et al. 07]
  [Jarosz et al. 08a, 08b]

- Second order (Hessian) analysis of irradiance

- There has actually been a considerable amount of research on illumination gradients, and we will see how some of these can be re-derived much more easily in a 2D setting
- Furthermore, we will go a step further, and perform a 2nd-order analysis, and apply this to irradiance caching

Gradients (Arc-Length Formulation)

- Differentiate Arc-Length formulation of irradiance:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) = \nabla_{\mathbf{x}} \int_{\mathfrak{L}} L_{2D}(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x}, \mathbf{y})\, G_{2D}(\mathbf{x}, \mathbf{y})\, dl(\mathbf{y})$$

- The simplest approach to derive an irradiance gradient is to simply take the arc-length form of the irradiance integral and apply the gradient operator
- By distributing it within the integral and applying the product rule we are left with three gradients
- Since we are dealing with Lambertian scenes, then the first term drops out
- The second term is the gradient of the visibility function
- We make a simplifying assumption (as in previous work) that the visibility gradient is zero
- This leaves just this final term, which is simply the gradient of the analytic geometric term

Gradients (Arc-Length Formulation)

- Differentiate Arc-Length formulation of irradiance:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) = \nabla_{\mathbf{x}} \int_{\mathfrak{L}} L_{2D}(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x}, \mathbf{y})\, G_{2D}(\mathbf{x}, \mathbf{y})\; dl(\mathbf{y})$$

$$= \int_{\mathfrak{L}} \nabla_{\mathbf{x}} L\, V\, G \;+\; L\, \nabla_{\mathbf{x}} V\, G \;+\; L\, V\, \nabla_{\mathbf{x}} G\; dl(\mathbf{y})$$

- The simplest approach to derive an irradiance gradient is to simply take the arc-length form of the irradiance integral and apply the gradient operator
- By distributing it within the integral and applying the product rule we are left with three gradients
- Since we are dealing with Lambertian scenes, then the first term drops out
- The second term is the gradient of the visibility function
- We make a simplifying assumption (as in previous work) that the visibility gradient is zero
- This leaves just this final term, which is simply the gradient of the analytic geometric term

Gradients (Arc-Length Formulation)

- Differentiate Arc-Length formulation of irradiance:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) = \nabla_{\mathbf{x}} \int_{\mathfrak{L}} L_{2D}(\mathbf{x}\leftarrow\mathbf{y})\, V_{2D}(\mathbf{x},\mathbf{y})\, G_{2D}(\mathbf{x},\mathbf{y})\; dl(\mathbf{y})$$

$$= \int_{\mathfrak{L}} \cancel{\nabla_{\mathbf{x}} L\, V\, G} + L\, \nabla_{\mathbf{x}} V\, G + L\, V\, \nabla_{\mathbf{x}} G\; dl(\mathbf{y})$$

- The simplest approach to derive an irradiance gradient is to simply take the arc-length form of the irradiance integral and apply the gradient operator
- By distributing it within the integral and applying the product rule we are left with three gradients
- Since we are dealing with Lambertian scenes, then the first term drops out
- The second term is the gradient of the visibility function
- We make a simplifying assumption (as in previous work) that the visibility gradient is zero
- This leaves just this final term, which is simply the gradient of the analytic geometric term

Gradients (Arc-Length Formulation)

- Differentiate Arc-Length formulation of irradiance:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) = \nabla_{\mathbf{x}} \int_{\mathfrak{L}} L_{2D}(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x}, \mathbf{y})\, G_{2D}(\mathbf{x}, \mathbf{y})\, dl(\mathbf{y})$$

$$= \int_{\mathfrak{L}} \cancel{\nabla_{\mathbf{x}} L\, V}\, G + L\, \cancel{\nabla_{\mathbf{x}} V}\, G + L\, V\, \nabla_{\mathbf{x}} G\, dl(\mathbf{y})$$

- The simplest approach to derive an irradiance gradient is to simply take the arc-length form of the irradiance integral and apply the gradient operator
- By distributing it within the integral and applying the product rule we are left with three gradients
- Since we are dealing with Lambertian scenes, then the first term drops out
- The second term is the gradient of the visibility function
- We make a simplifying assumption (as in previous work) that the visibility gradient is zero
- This leaves just this final term, which is simply the gradient of the analytic geometric term

Gradients (Arc-Length Formulation)

- Differentiate Arc-Length formulation of irradiance:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) = \nabla_{\mathbf{x}} \int_{\mathfrak{L}} L_{2D}(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x}, \mathbf{y})\, G_{2D}(\mathbf{x}, \mathbf{y})\, dl(\mathbf{y})$$

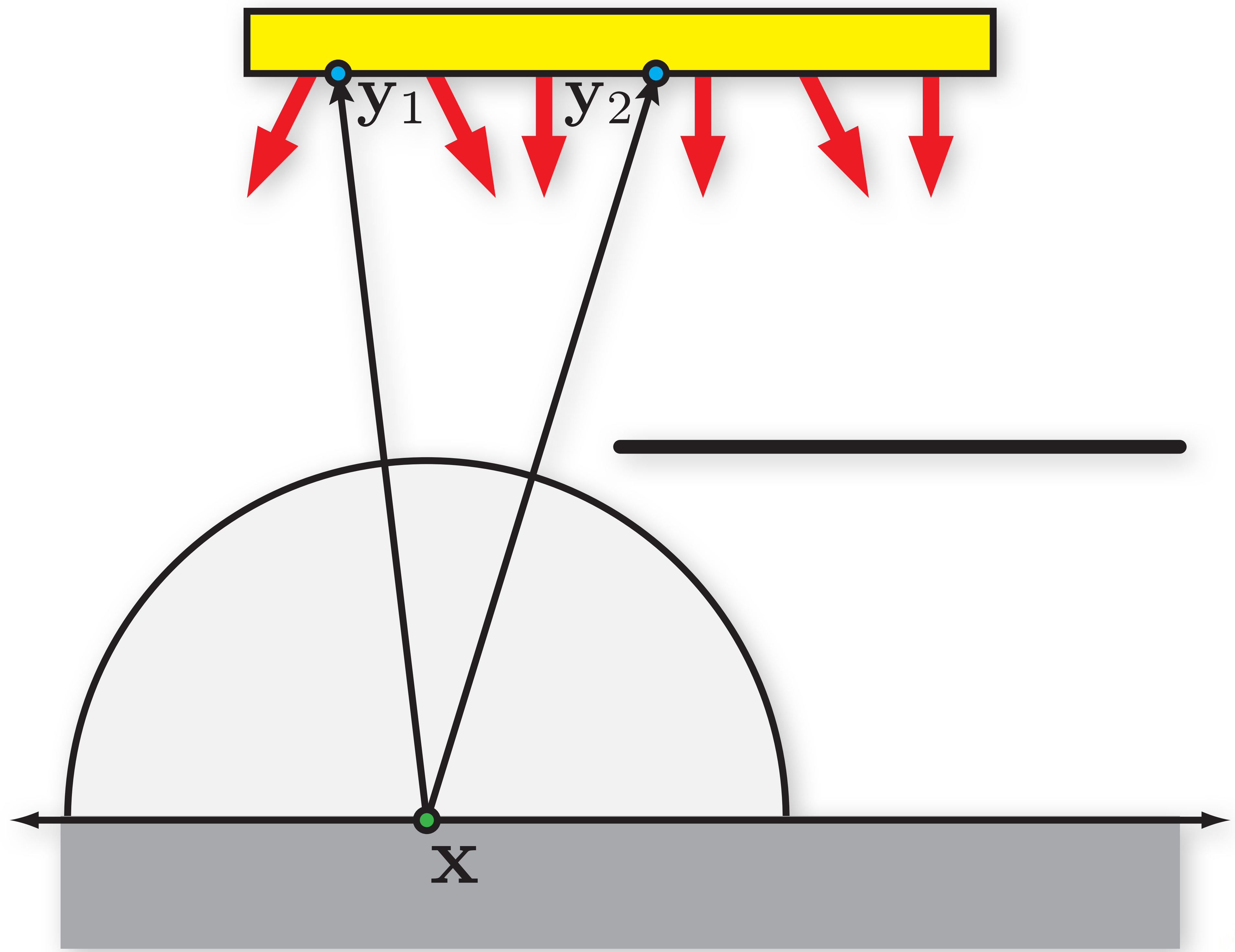
$$= \int_{\mathfrak{L}} \cancel{\nabla_{\mathbf{x}} L\, V\, G} + \cancel{L\, \nabla_{\mathbf{x}} V\, G} + \boxed{L\, V\, \nabla_{\mathbf{x}} G}\, dl(\mathbf{y})$$

$$\approx \int_{\mathfrak{L}} L_{2D}(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x}, \mathbf{y})\, \nabla_{\mathbf{x}} G_{2D}(\mathbf{x}, \mathbf{y})\, dl(\mathbf{y})$$

Wednesday, 5 September 12

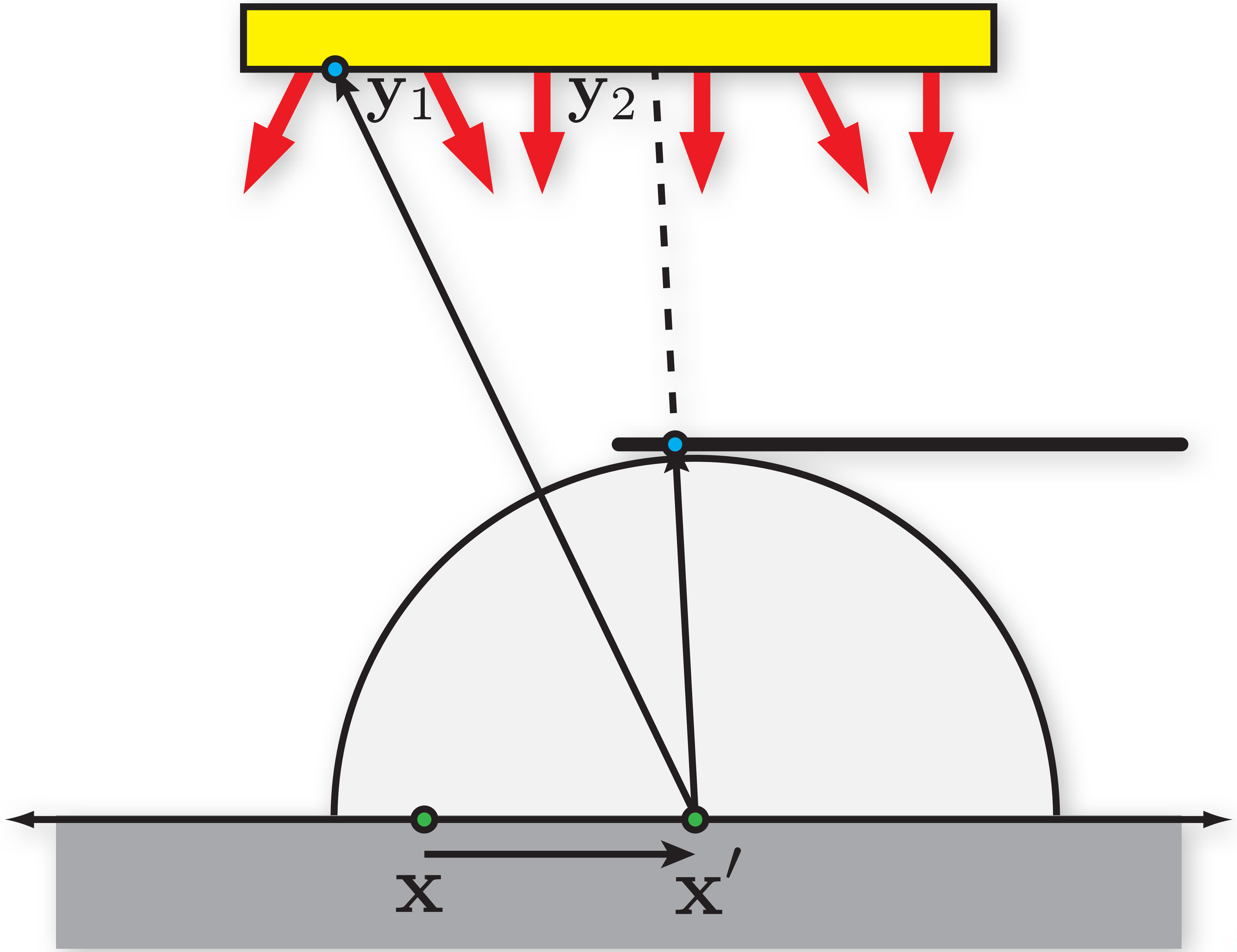
- The simplest approach to derive an irradiance gradient is to simply take the arc-length form of the irradiance integral and apply the gradient operator
- By distributing it within the integral and applying the product rule we are left with three gradients
- Since we are dealing with Lambertian scenes, then the first term drops out
- The second term is the gradient of the visibility function
- We make a simplifying assumption (as in previous work) that the visibility gradient is zero
- This leaves just this final term, which is simply the gradient of the analytic geometric term

Gradients (Arc-Length Formulation)



- To see how this works, we imagine shooting a number rays over the hemicircle, which hit other surfaces
- We are now interested in how the irradiance changes as we translate the evaluation location x

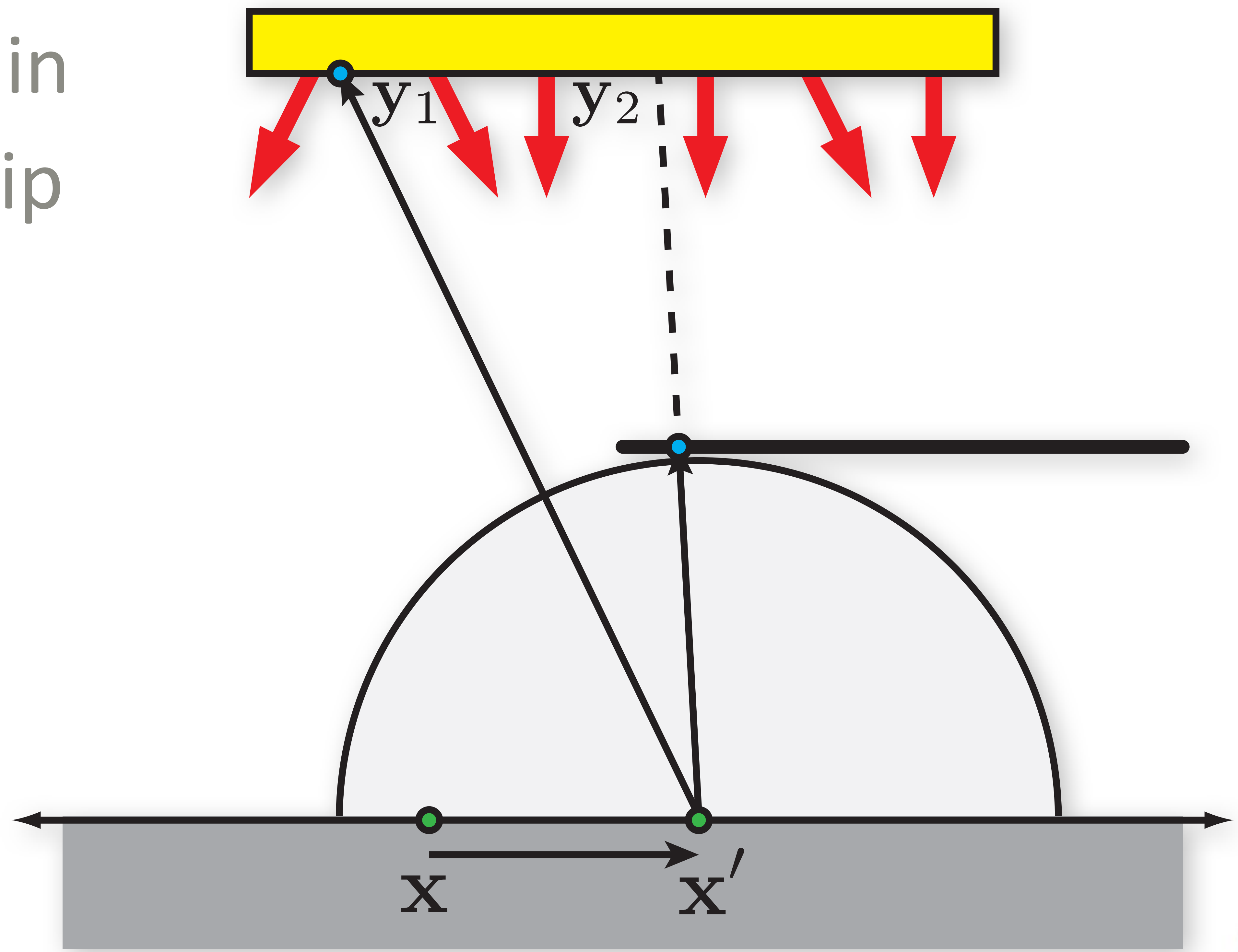
Gradients (Arc-Length Formulation)



- The arc-length formulation accounts for the change in the geometric relationship between x and the hitpoints y
- However, it ignores changes due to occlusions

Gradients (Arc-Length Formulation)

- Accounts for change in geometric relationship between $\mathbf{x}$ & $\mathbf{y}$

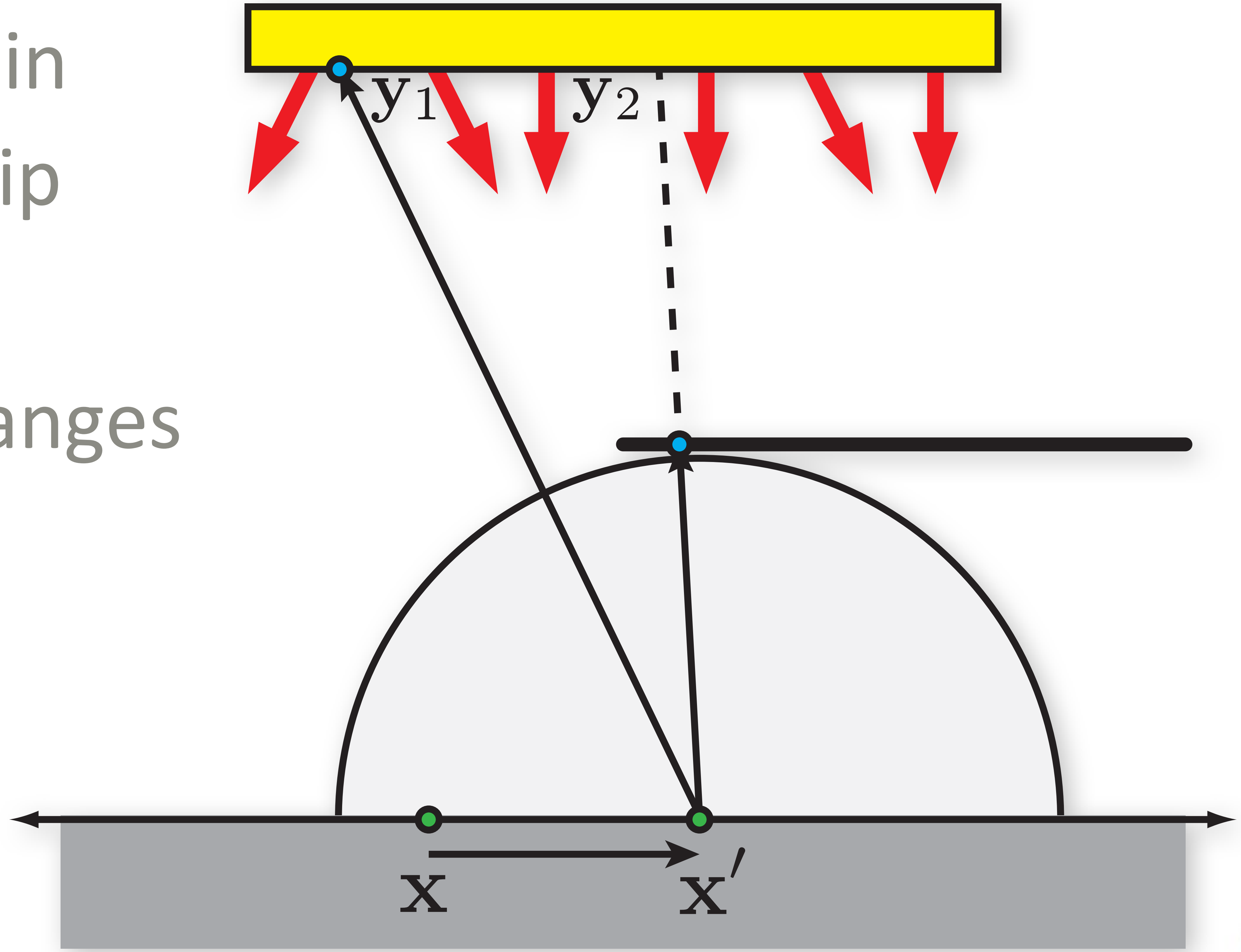


Wednesday, 5 September 12

- The arc-length formulation accounts for the change in the geometric relationship between x and the hitpoints y
- However, it ignores changes due to occlusions

Gradients (Arc-Length Formulation)

- Accounts for change in geometric relationship between $\mathbf{x}$ & $\mathbf{y}$
- Ignores occlusion changes

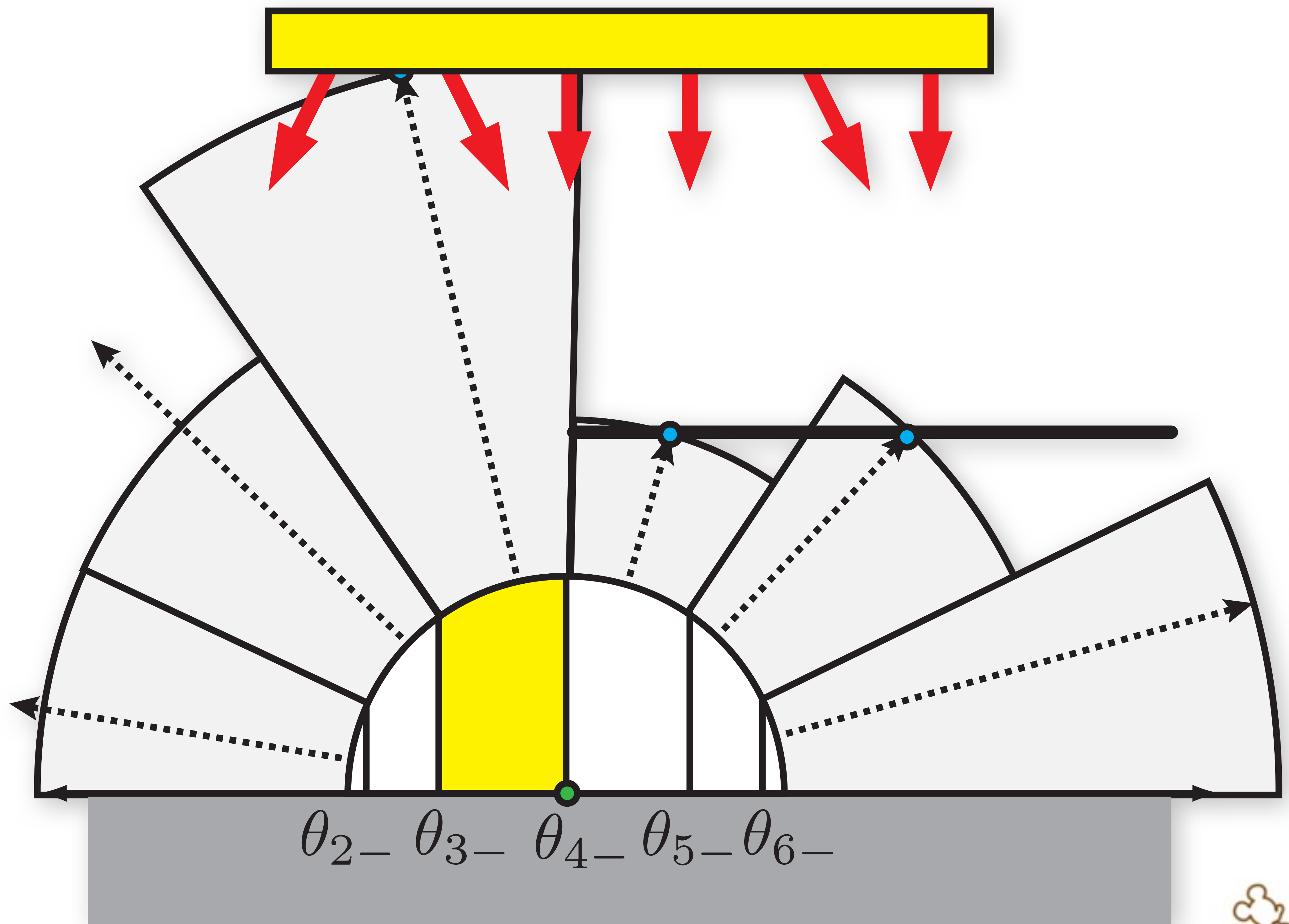


Wednesday, 5 September 12

- The arc-length formulation accounts for the change in the geometric relationship between x and the hitpoints y
- However, it ignores changes due to occlusions

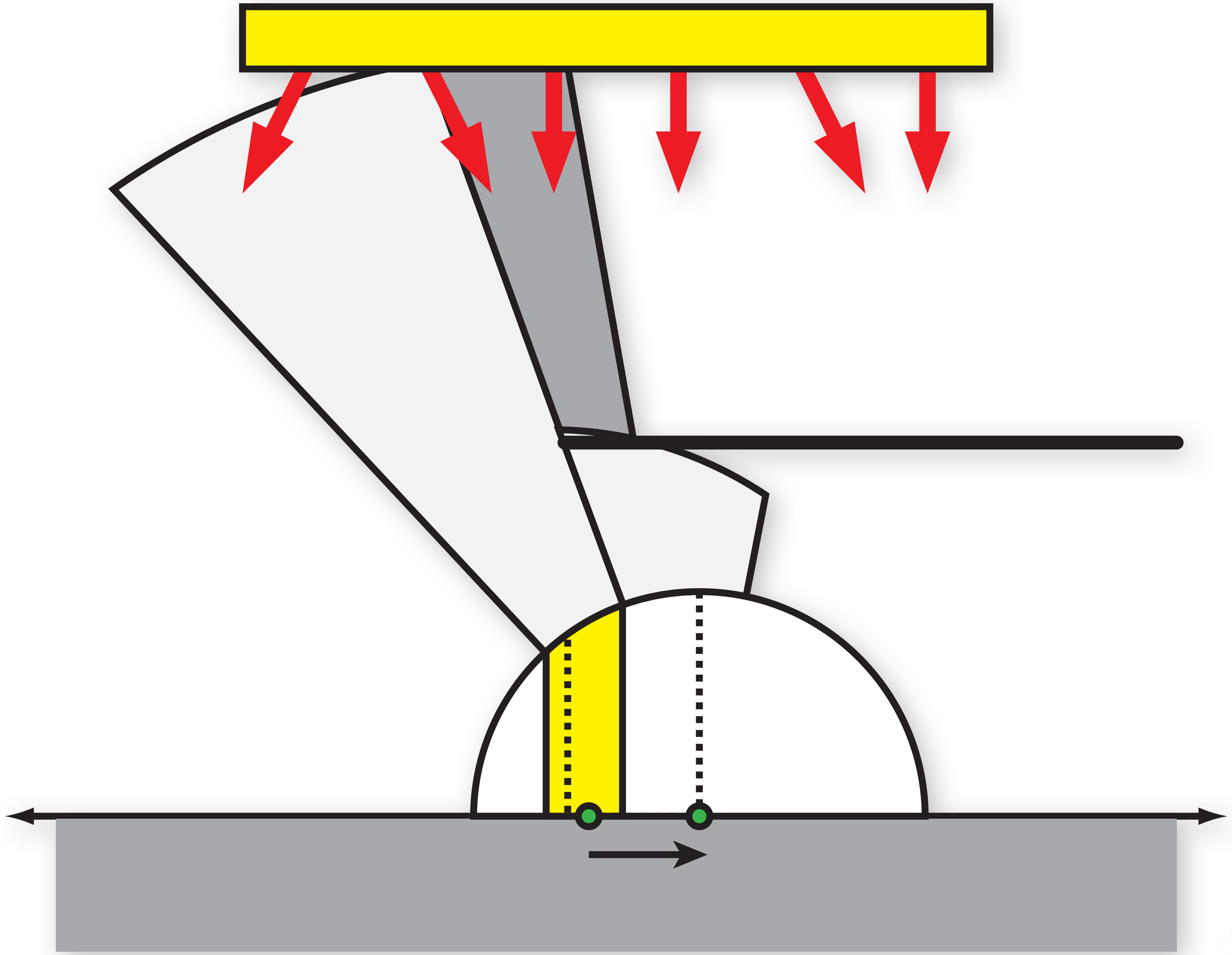
Gradients (Stratified Formulation)

[Ward and Heckbert 92]



- A more sophisticated method was proposed by Ward and Heckbert, which we can again analyze and re-derive in 2D
- This method stratifies the direction form of the irradiance integral, and shoots a ray in each stratum
- The gradient computation then tries to consider how the sizes of the strata would change as we moved the center of project

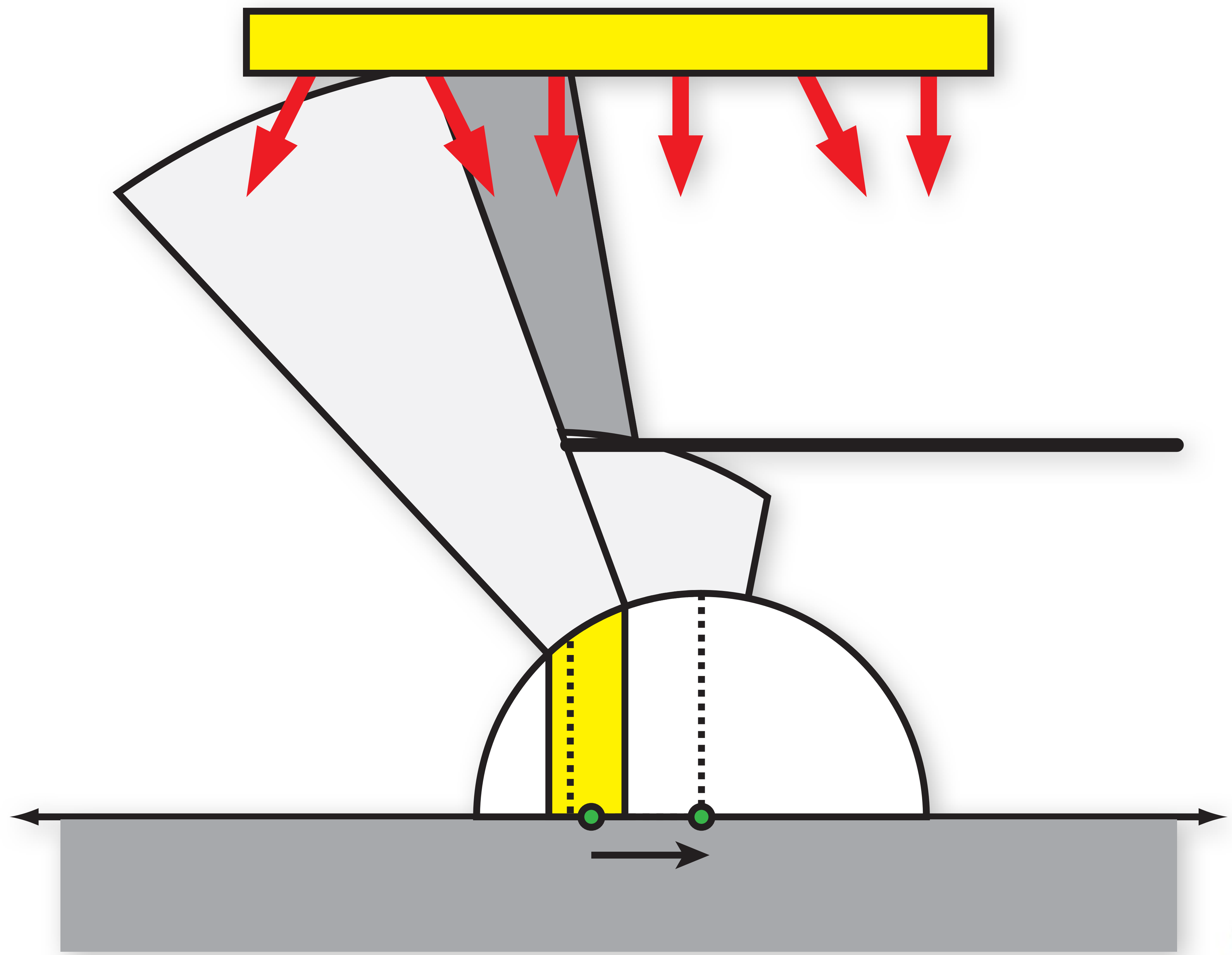
Gradients (Stratified Formulation)



- And the big benefit of enforcing a stratification, is that due to neighbor relationships we can account for occlusion changes during translation

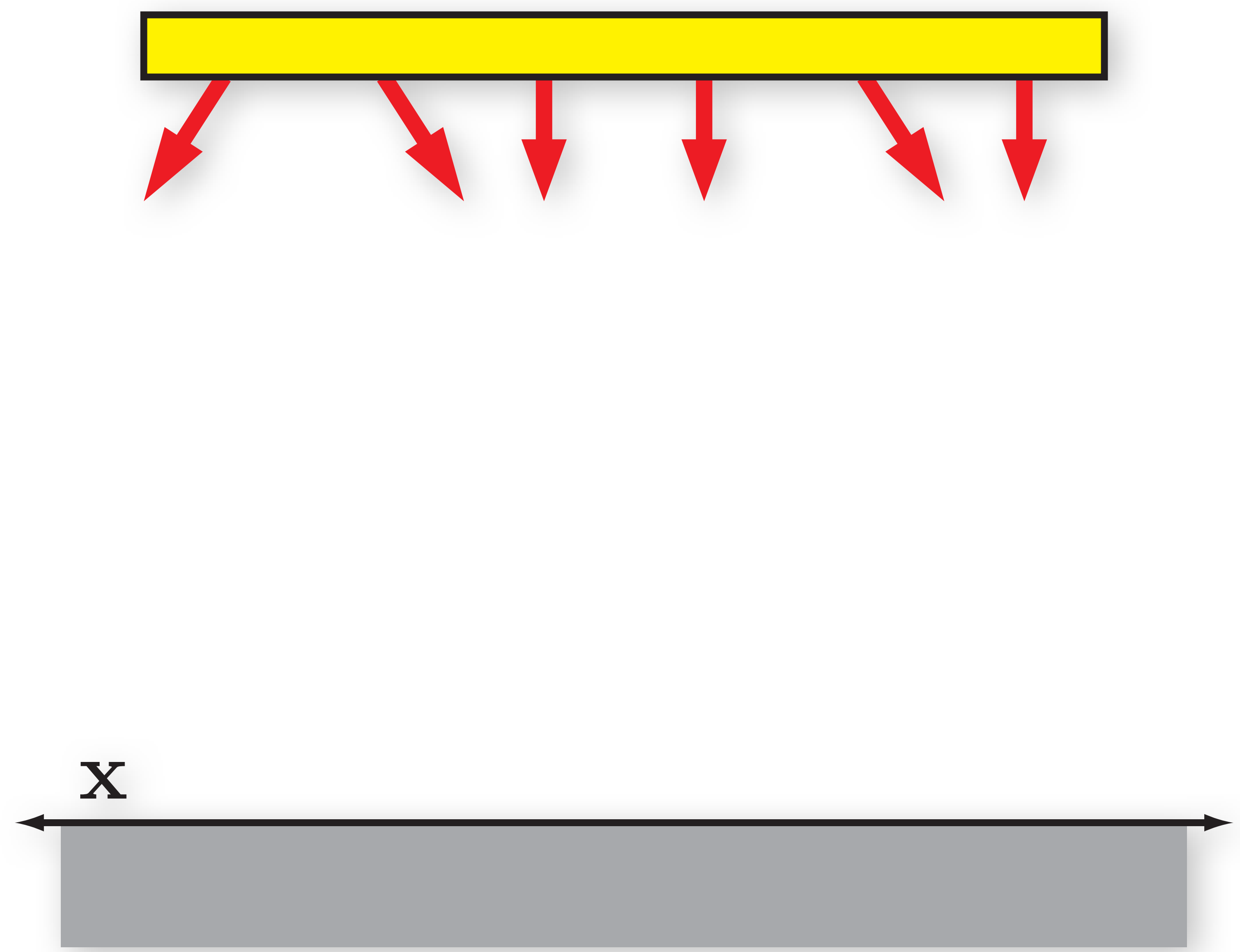
Gradients (Stratified Formulation)

- Considers occlusion changes



- And the big benefit of enforcing a stratification, is that due to neighbor relationships we can account for occlusion changes during translation

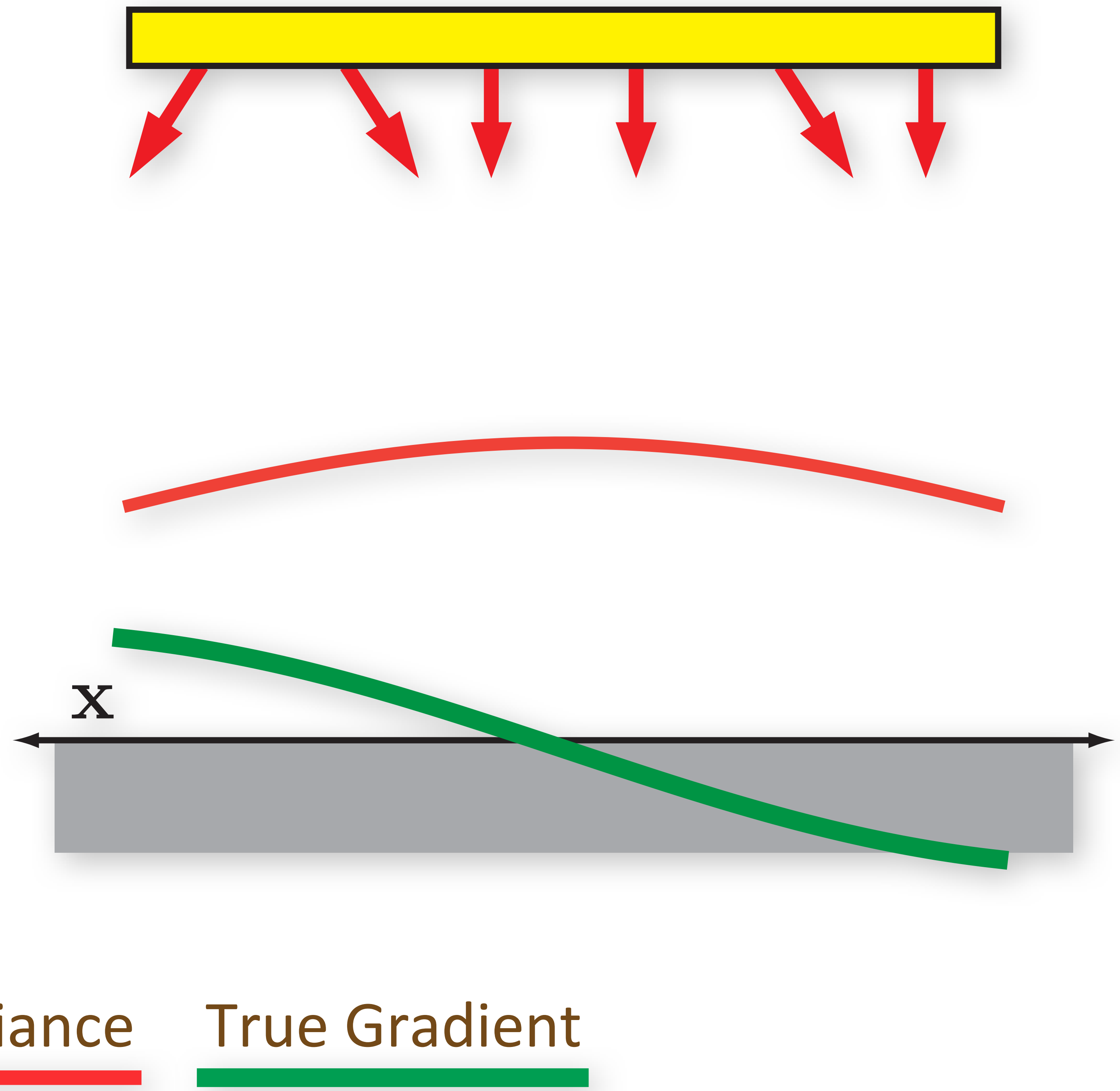
Irradiance Gradients Comparison



Wednesday, 5 September 12

- We can easily compare these two approaches in 2D on a simple scene where we have a light at the top
- In 2D, we can actually compute analytic solutions for the irradiance, as well as the gradient, which we plot as red and green curves along the floor
- In a scene without occluders both numerical methods return the correct solution
- But when we add an occluder, we see that the stratified formulation (dotted-black) retains the correct answers, while the arc-length formulation (orange) gives the wrong results because it ignores the gradient of the visibility function

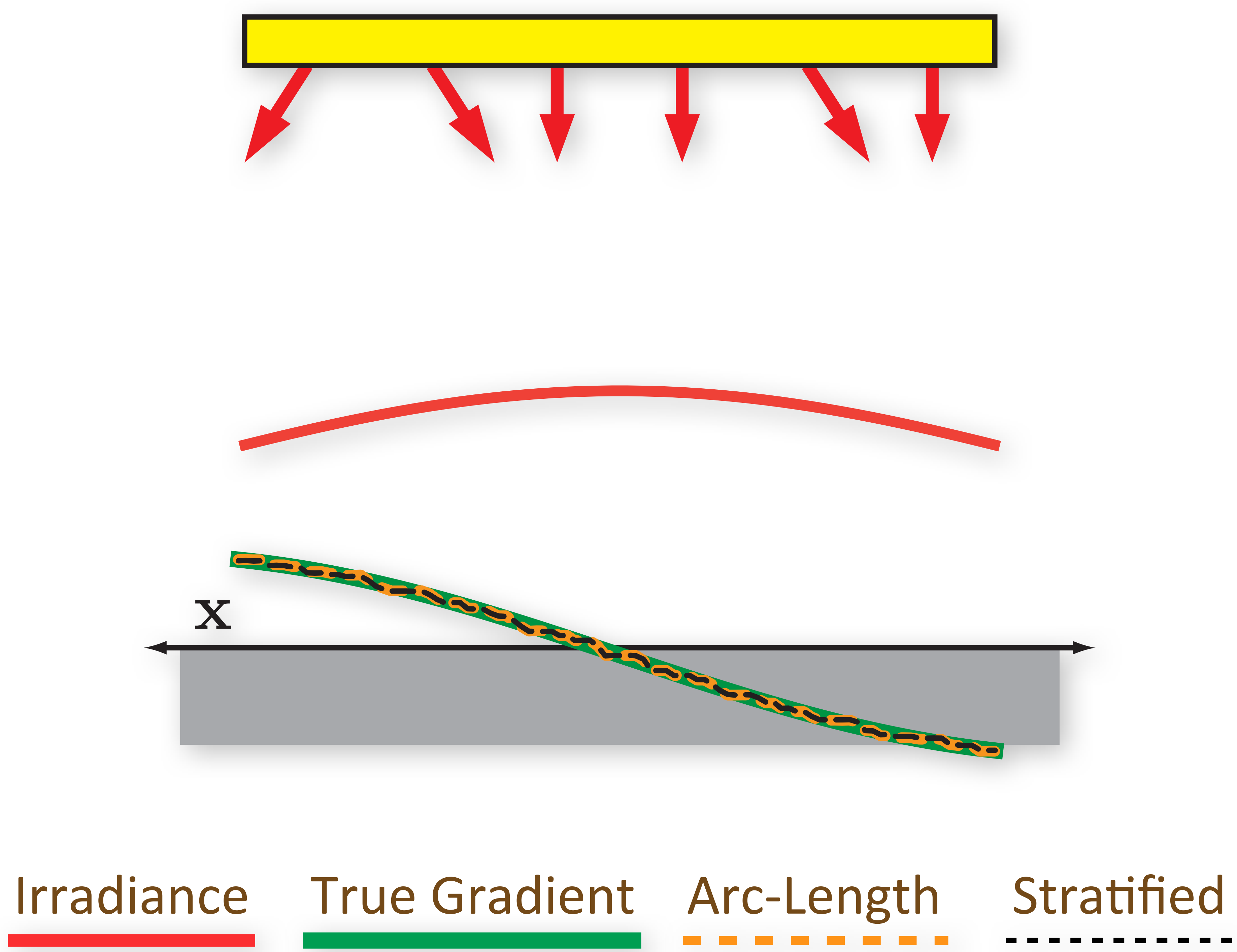
Irradiance Gradients Comparison



Wednesday, 5 September 12

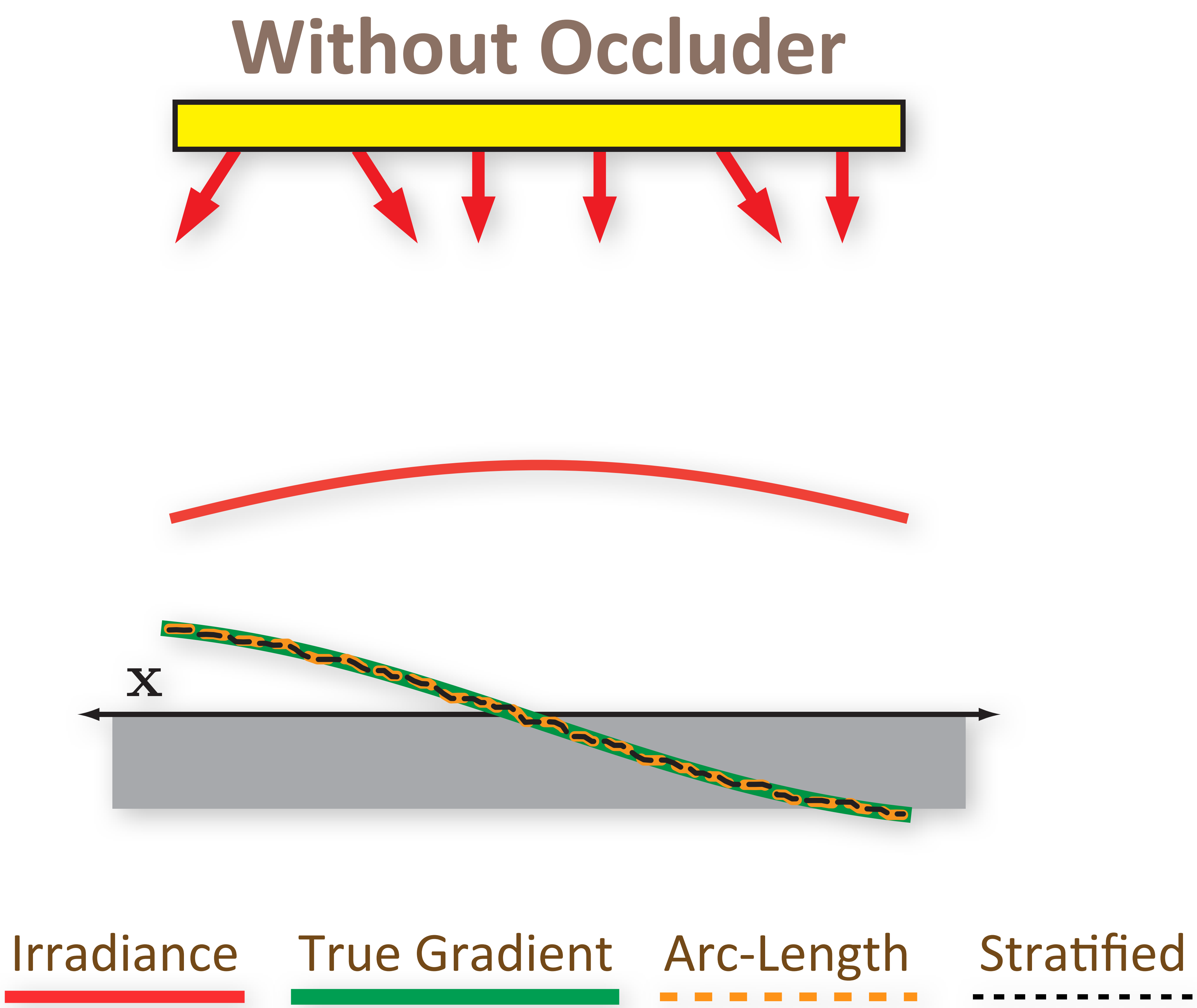
- We can easily compare these two approaches in 2D on a simple scene where we have a light at the top
- In 2D, we can actually compute analytic solutions for the irradiance, as well as the gradient, which we plot as red and green curves along the floor
- In a scene without occluders both numerical methods return the correct solution
- But when we add an occluder, we see that the stratified formulation (dotted-black) retains the correct answers, while the arc-length formulation (orange) gives the wrong results because it ignores the gradient of the visibility function

Irradiance Gradients Comparison



- We can easily compare these two approaches in 2D on a simple scene where we have a light at the top
- In 2D, we can actually compute analytic solutions for the irradiance, as well as the gradient, which we plot as red and green curves along the floor
- In a scene without occluders both numerical methods return the correct solution
- But when we add an occluder, we see that the stratified formulation (dotted-black) retains the correct answers, while the arc-length formulation (orange) gives the wrong results because it ignores the gradient of the visibility function

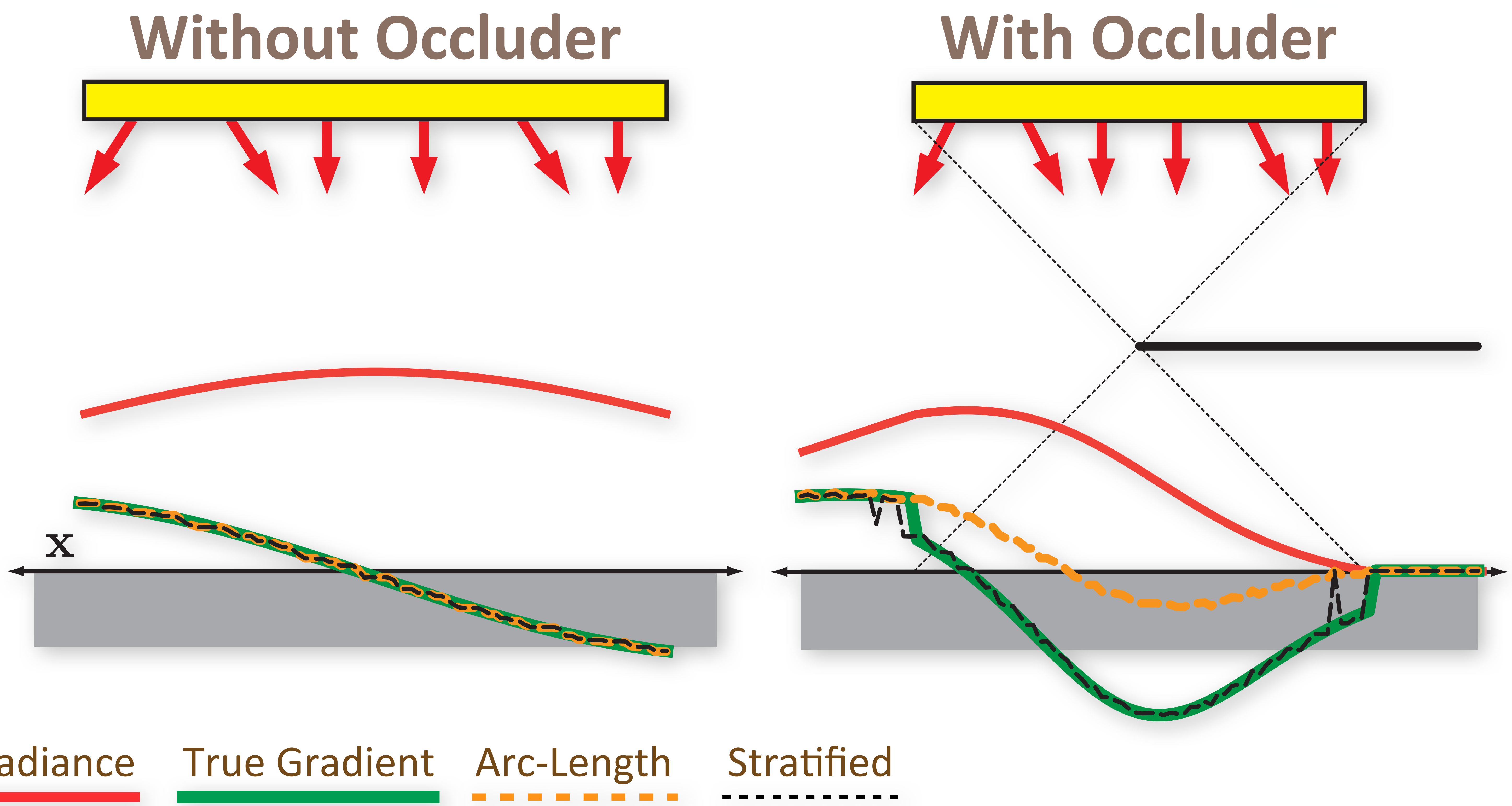
Irradiance Gradients Comparison



Wednesday, 5 September 12

- We can easily compare these two approaches in 2D on a simple scene where we have a light at the top
- In 2D, we can actually compute analytic solutions for the irradiance, as well as the gradient, which we plot as red and green curves along the floor
- In a scene without occluders both numerical methods return the correct solution
- But when we add an occluder, we see that the stratified formulation (dotted-black) retains the correct answers, while the arc-length formulation (orange) gives the wrong results because it ignores the gradient of the visibility function

Irradiance Gradients Comparison



Wednesday, 5 September 12

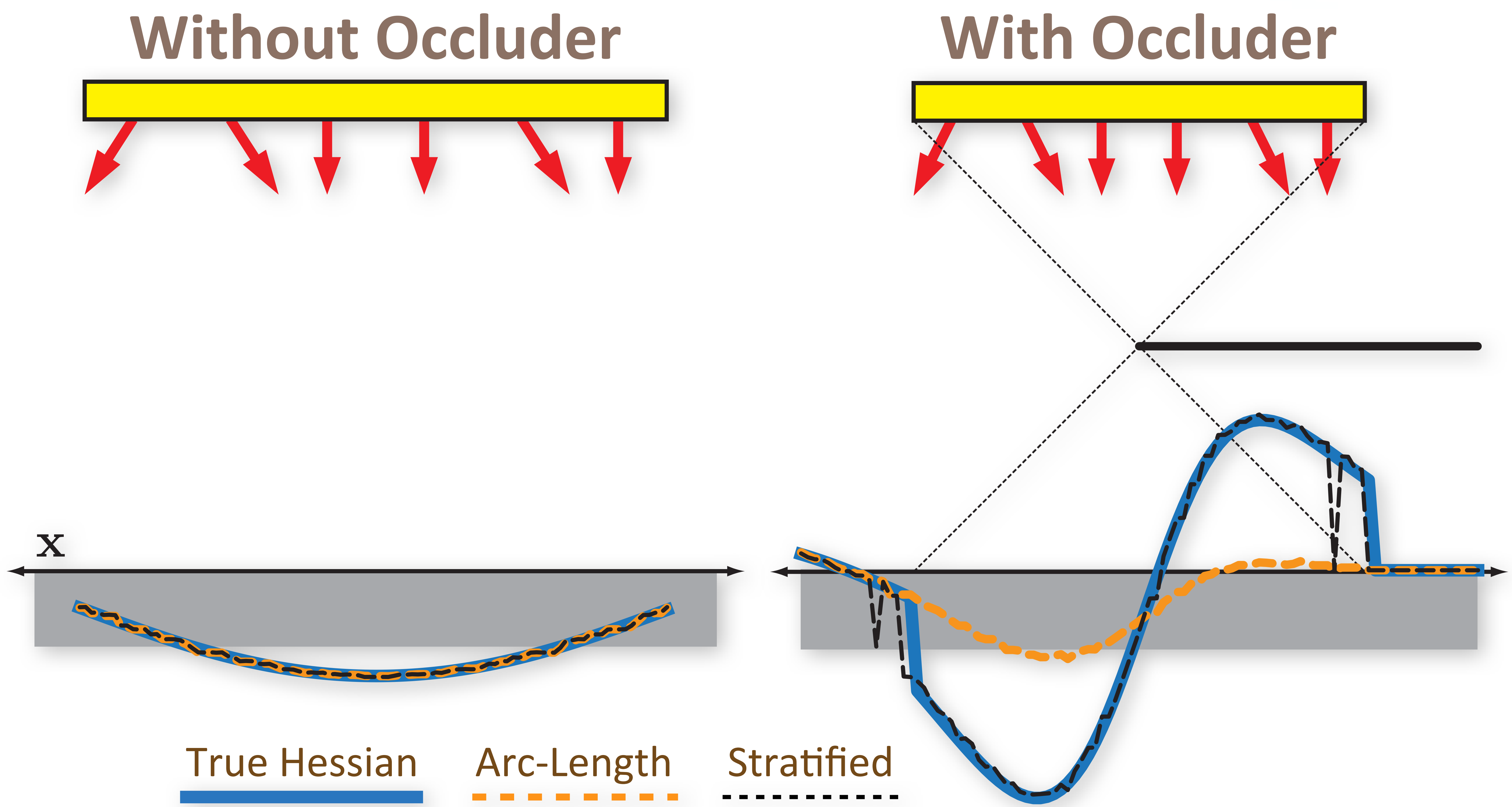
- We can easily compare these two approaches in 2D on a simple scene where we have a light at the top
- In 2D, we can actually compute analytic solutions for the irradiance, as well as the gradient, which we plot as red and green curves along the floor
- In a scene without occluders both numerical methods return the correct solution
- But when we add an occluder, we see that the stratified formulation (dotted-black) retains the correct answers, while the arc-length formulation (orange) gives the wrong results because it ignores the gradient of the visibility function

Beyond Previous Work

- Second-order analysis

- We can also easily go beyond what has been done in previous work, and extend both formulations to 2nd derivatives, or irradiance Hessians. The details are in the paper

Irradiance Hessian Comparison



Wednesday, 5 September 12

- Again, in this case the stratified formulation properly accounts for occlusion changes, whereas both are accurate when no occlusions are present

Moving to 3D

- To make practical use of this analysis, we need to generalize to 3D.
- We can easily generalize both gradient formulations to 3D, and our resulting 3D gradients have some minor, but practical benefits over previously published derivations
- We can also easily generalize the arc-length Hessian to 3D, and we will show that even though this formulation ignores occlusions, it can lead to practical benefits for 3D irradiance caching

Moving to 3D

- Gradient formulations easy to generalize to 3D
  - minor benefits over previous 3D derivations

- To make practical use of this analysis, we need to generalize to 3D.
- We can easily generalize both gradient formulations to 3D, and our resulting 3D gradients have some minor, but practical benefits over previously published derivations
- We can also easily generalize the arc-length Hessian to 3D, and we will show that even though this formulation ignores occlusions, it can lead to practical benefits for 3D irradiance caching

Moving to 3D

- Gradient formulations easy to generalize to 3D
  - minor benefits over previous 3D derivations
- Arc-Length Hessian easy to generalize to 3D (ignores occlusions)

- To make practical use of this analysis, we need to generalize to 3D.
- We can easily generalize both gradient formulations to 3D, and our resulting 3D gradients have some minor, but practical benefits over previously published derivations
- We can also easily generalize the arc-length Hessian to 3D, and we will show that even though this formulation ignores occlusions, it can lead to practical benefits for 3D irradiance caching

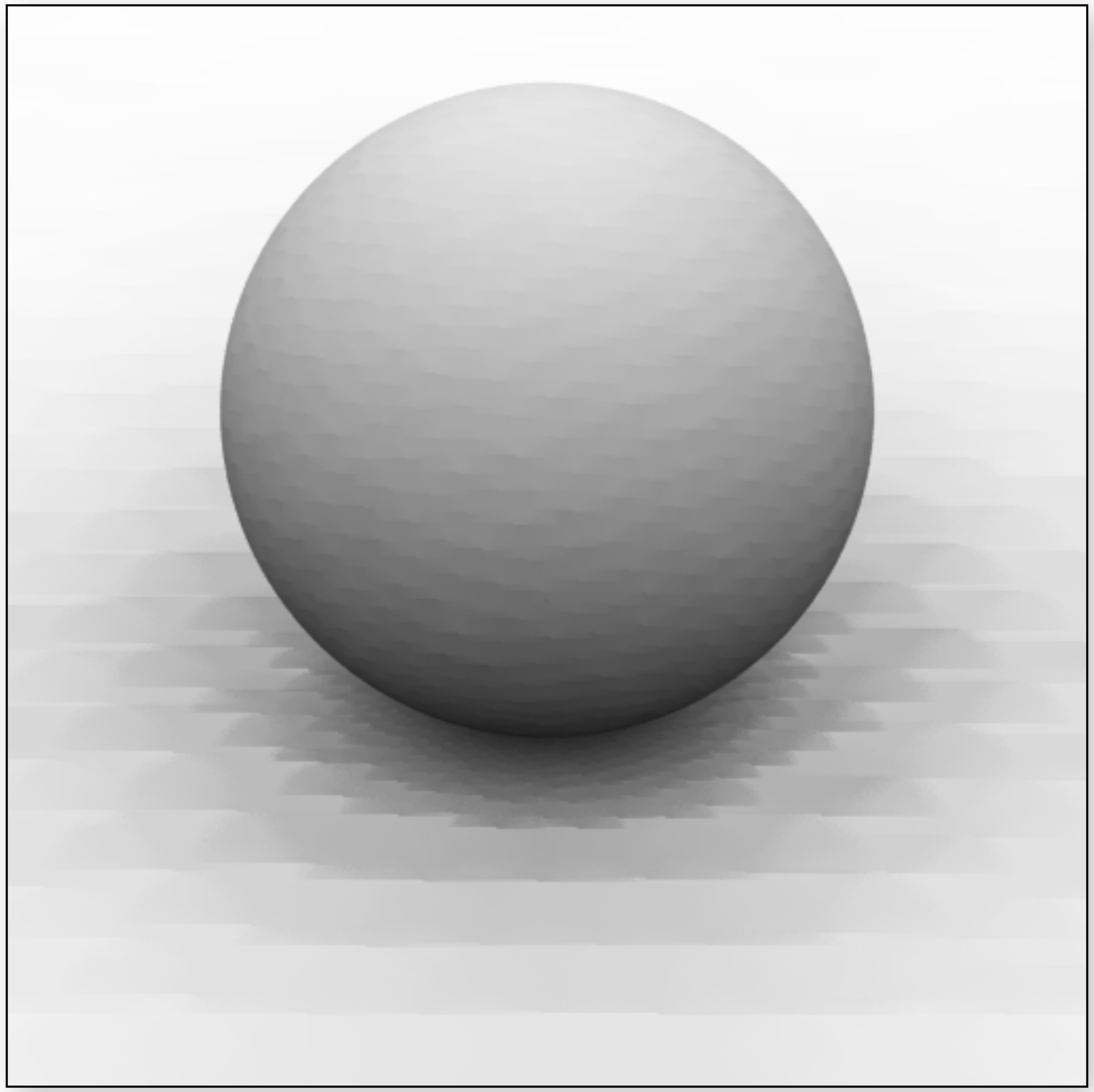
Irradiance Caching

- Apply gradient analysis to:
  - **Interpolation/Extrapolation**
  - **Error control**

- We now apply our gradient analysis to the two key parts of irradiance caching: extrapolation and error control

Irradiance Extrapolation

Constant

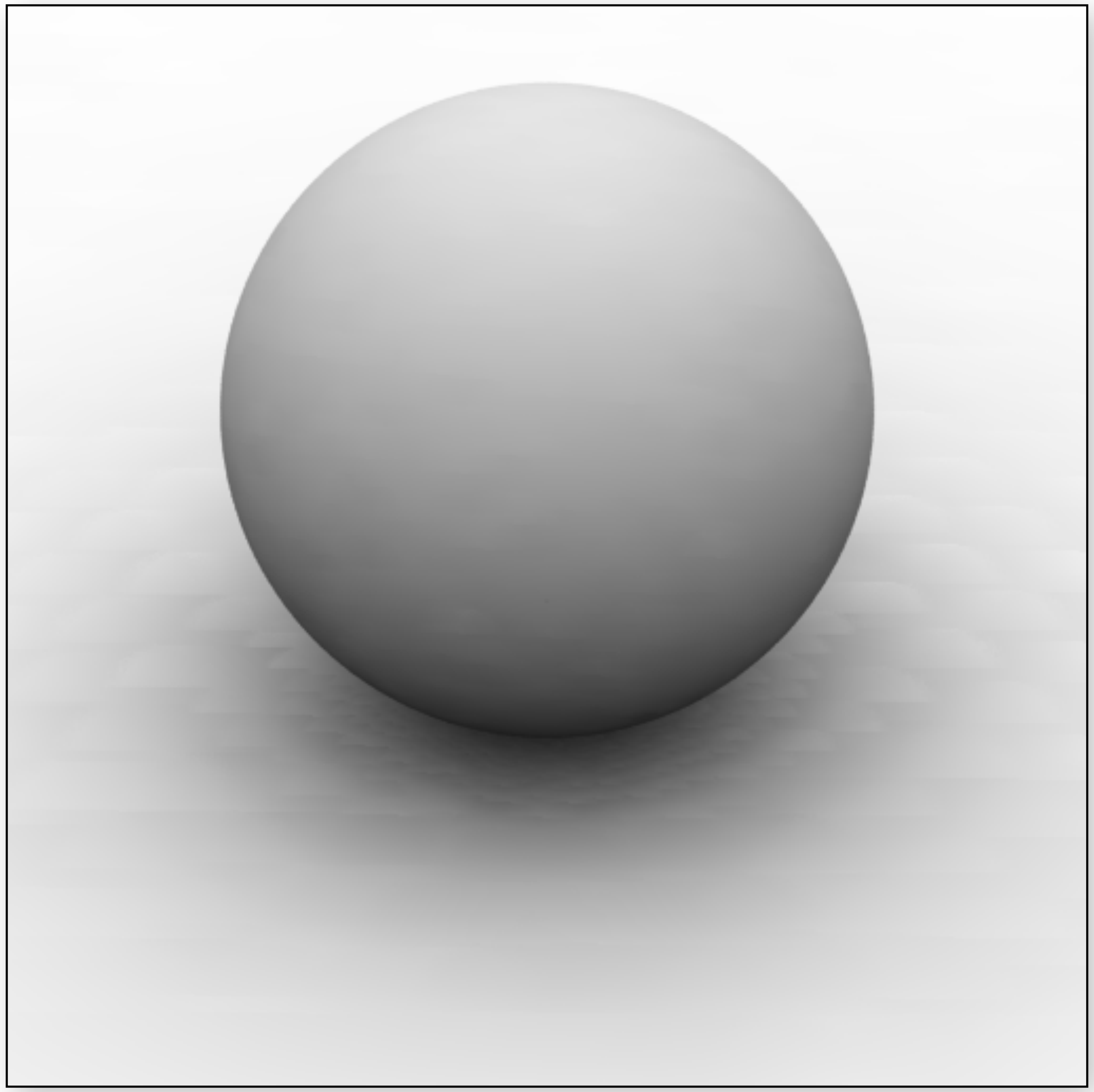


[Ward et al. in 1988]

- For cache point extrapolation, Ward initially proposed to simply re-use cache values using constant extrapolation

Irradiance Extrapolation

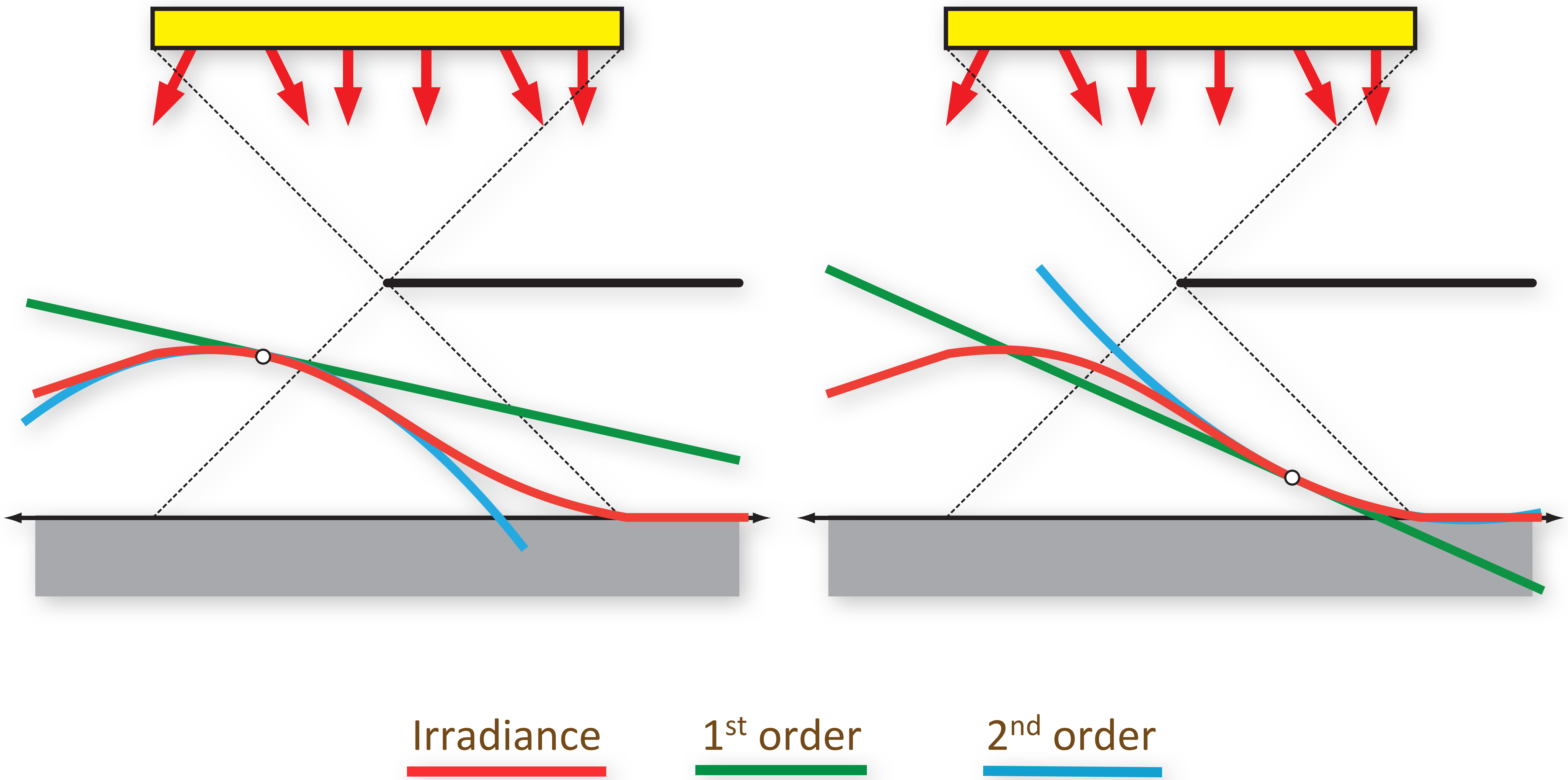
Gradient (Linear)



[Ward and Heckbert 92]

- Later, Ward and Heckbert linearly extrapolated the cached values along the irradiance gradient, which significantly improved reconstruction quality

1st & 2nd Order Extrapolation

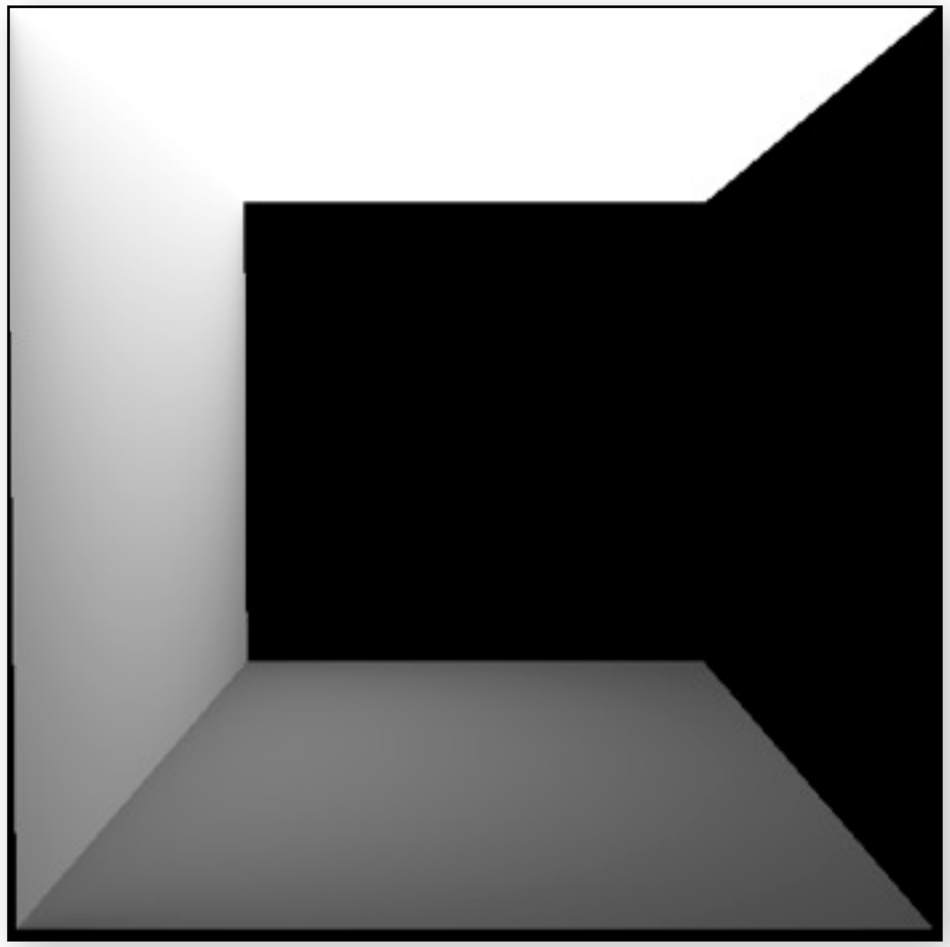


Wednesday, 5 September 12

- Given our irradiance Hessian derivations, we can now take this a step further
- Here we compare for two cache point locations a first-order extrapolation, and a second-order extrapolation, and we can see that by exploiting the information in the Hessian, we can more faithfully reconstruct the irradiance in the neighborhood of the cache point

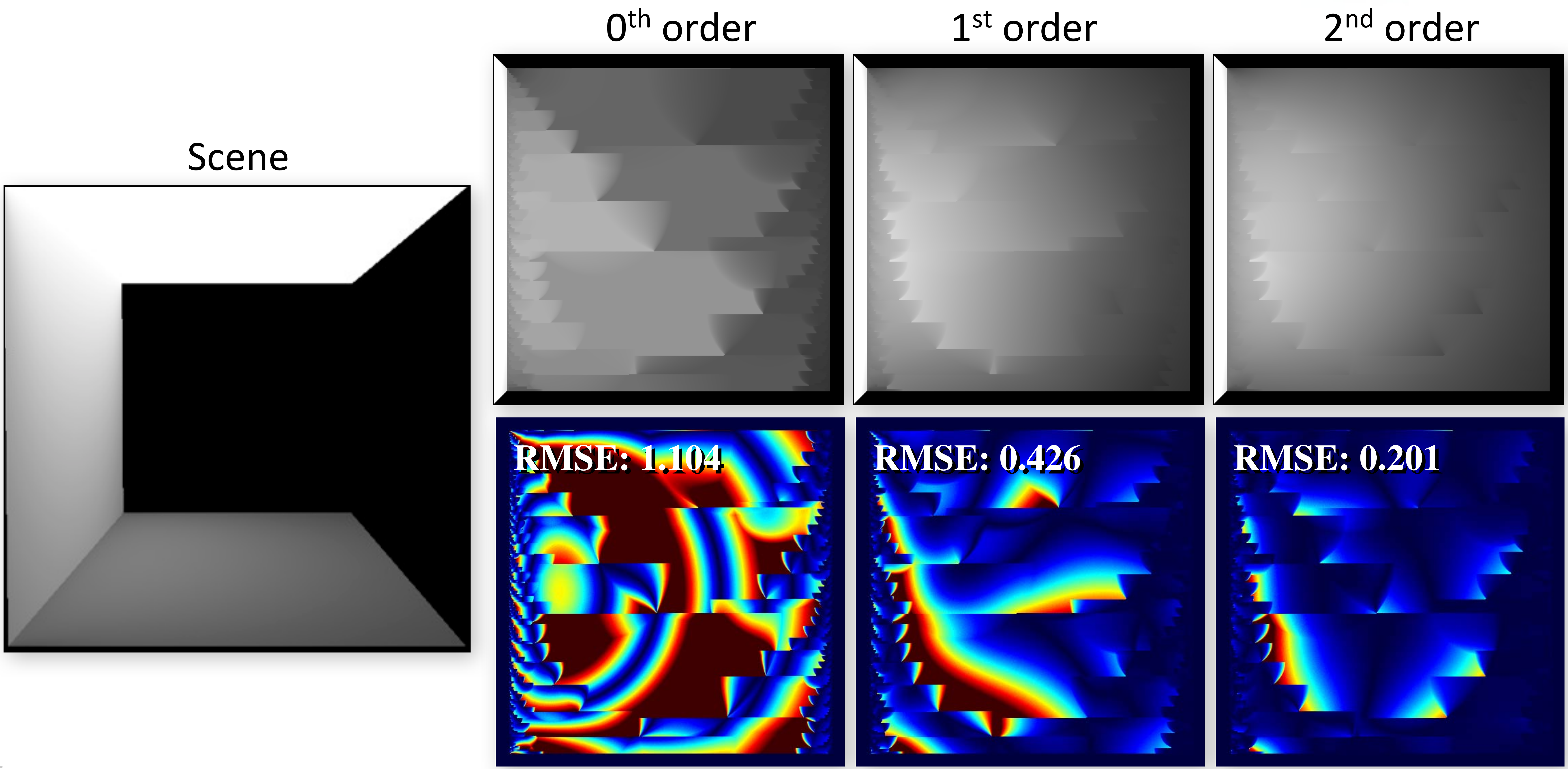
Taylor Extrapolation Comparison

Scene



- We can also apply this idea in 3D
- Here we visualize the indirect irradiance on the ground plane of this simple box scene (as viewed from above)
- We can see that as we perform higher-order taylor extrapolations, we improve the reconstruction quality, and reduce the RMS error

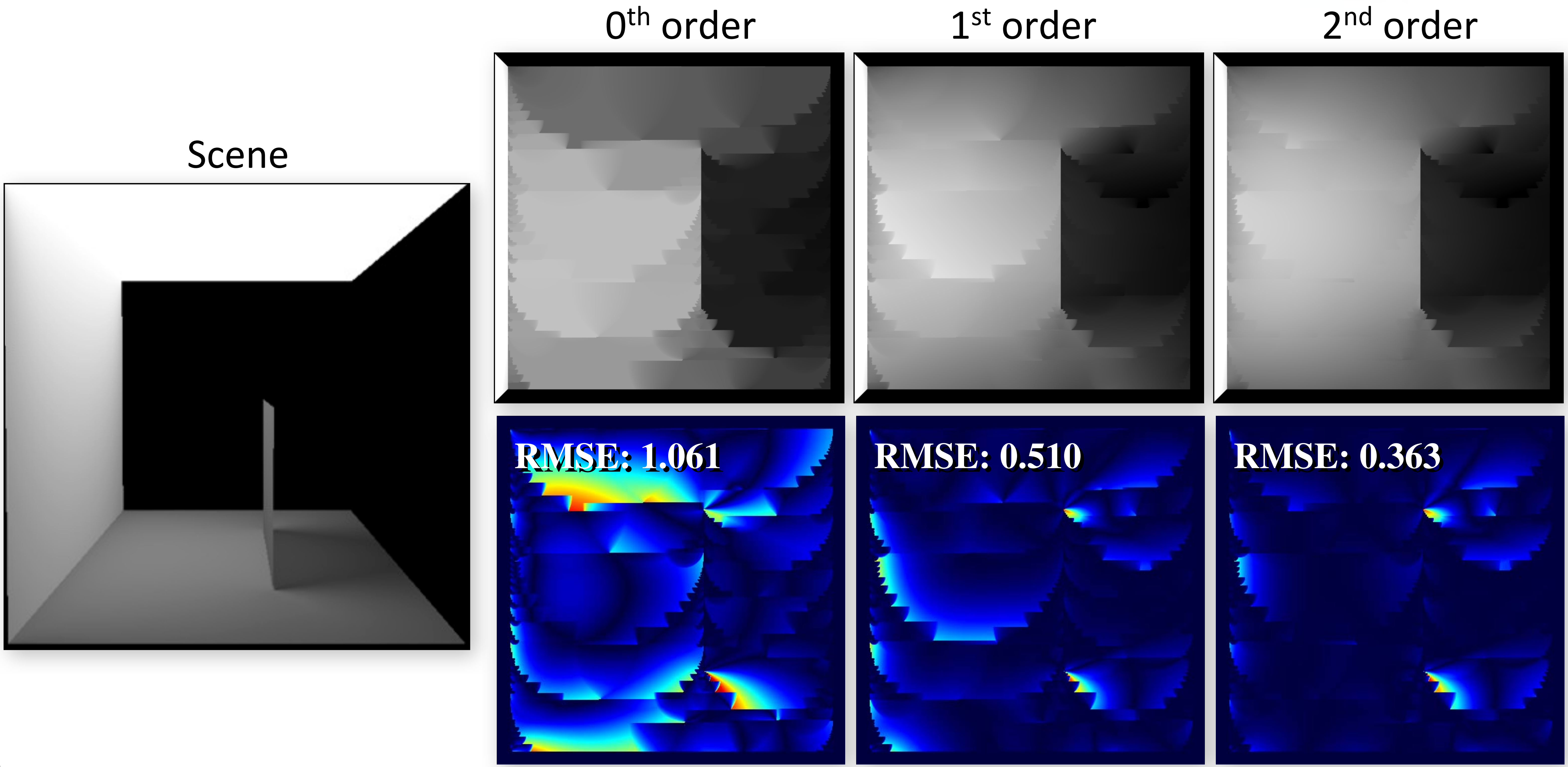
Taylor Extrapolation Comparison



Wednesday, 5 September 12

- We can also apply this idea in 3D
- Here we visualize the indirect irradiance on the ground plane of this simple box scene (as viewed from above)
- We can see that as we perform higher-order taylor extrapolations, we improve the reconstruction quality, and reduce the RMS error

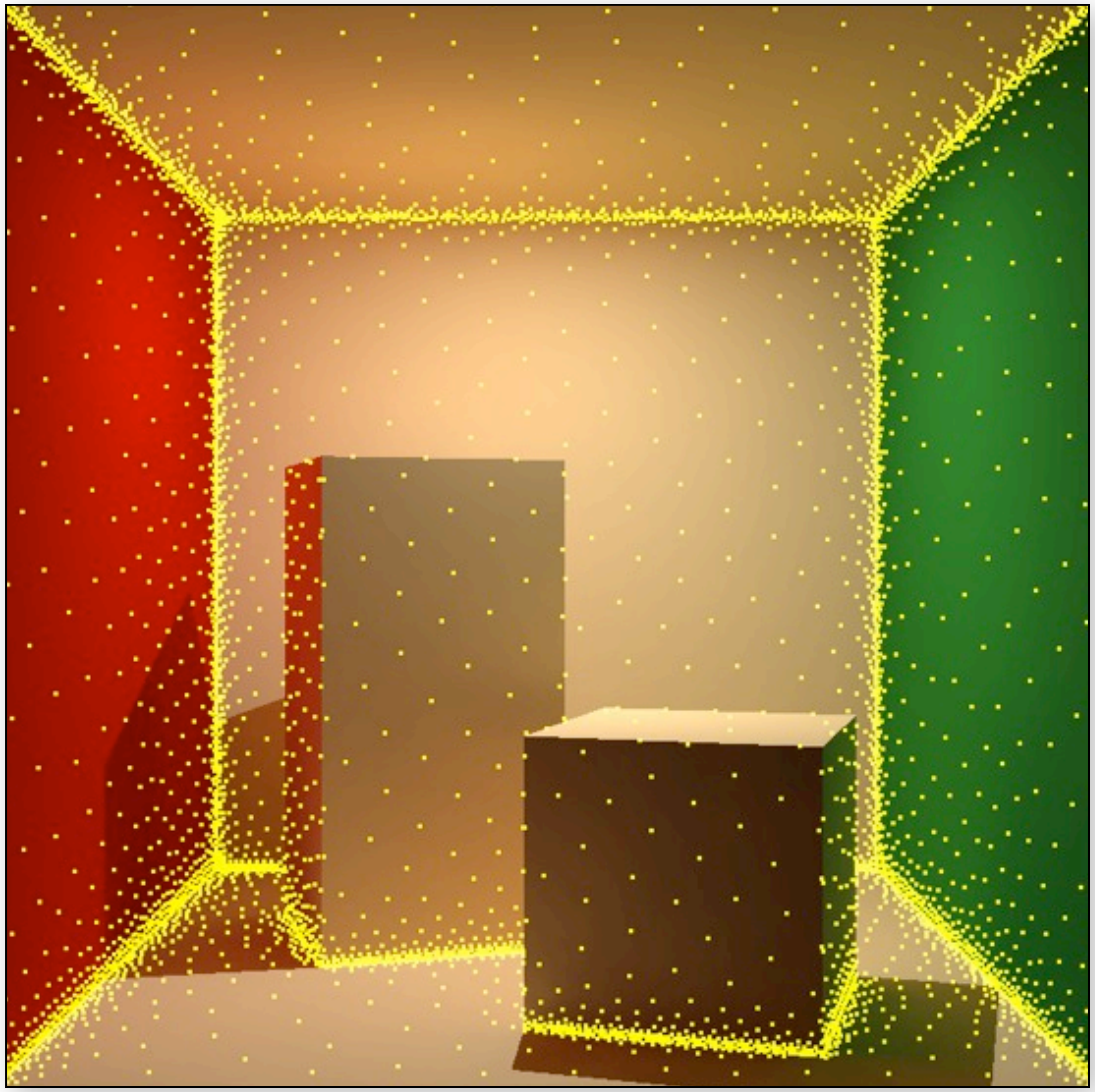
Taylor Extrapolation Comparison



Wednesday, 5 September 12

- And here we added a simple occluder to the scene, which introduces visibility changes
- We can see that even though our Hessian formulation in 3D ignores visibility changes, we can still obtain higher quality reconstruction and reduced RMS error

Error Control

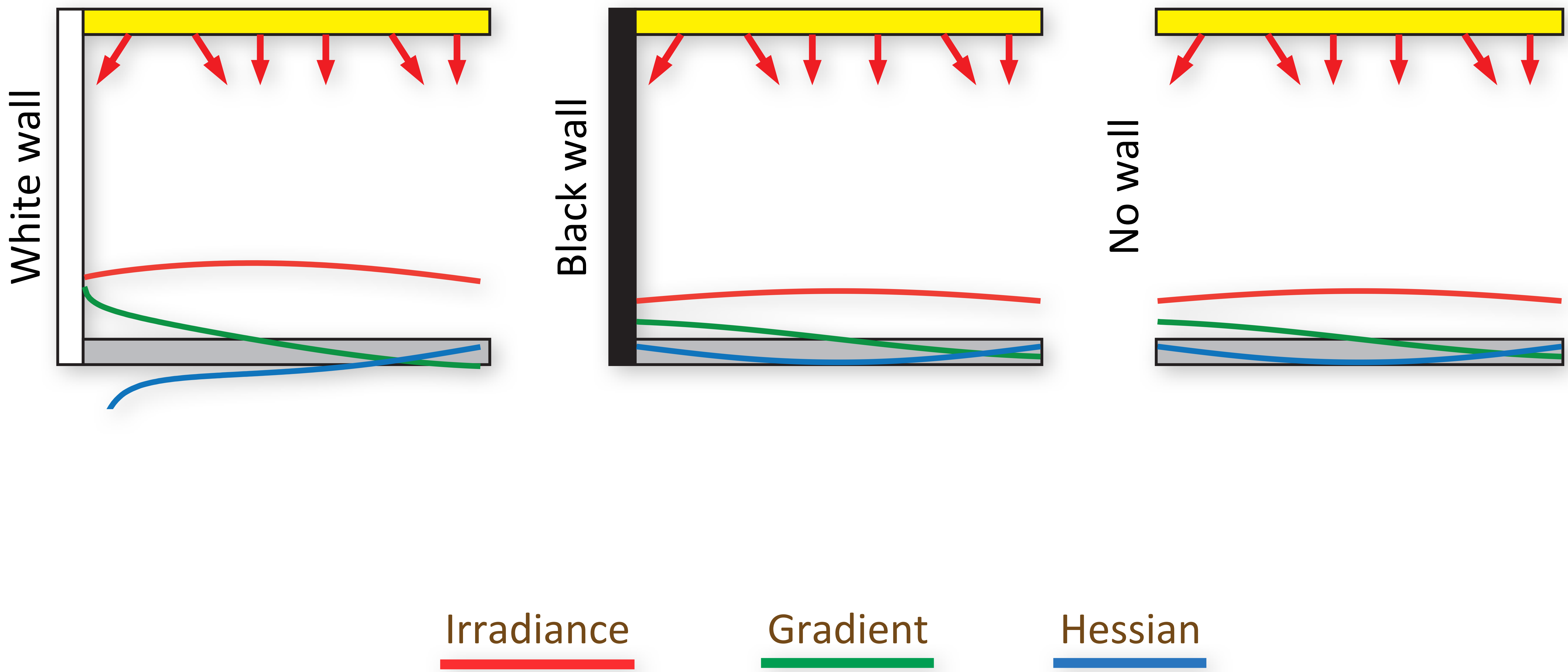


- "Split-sphere" heuristic
  - irradiance changes rapidly near objects
  - radius proportional to reciprocal "average" ray distance

Wednesday, 5 September 12

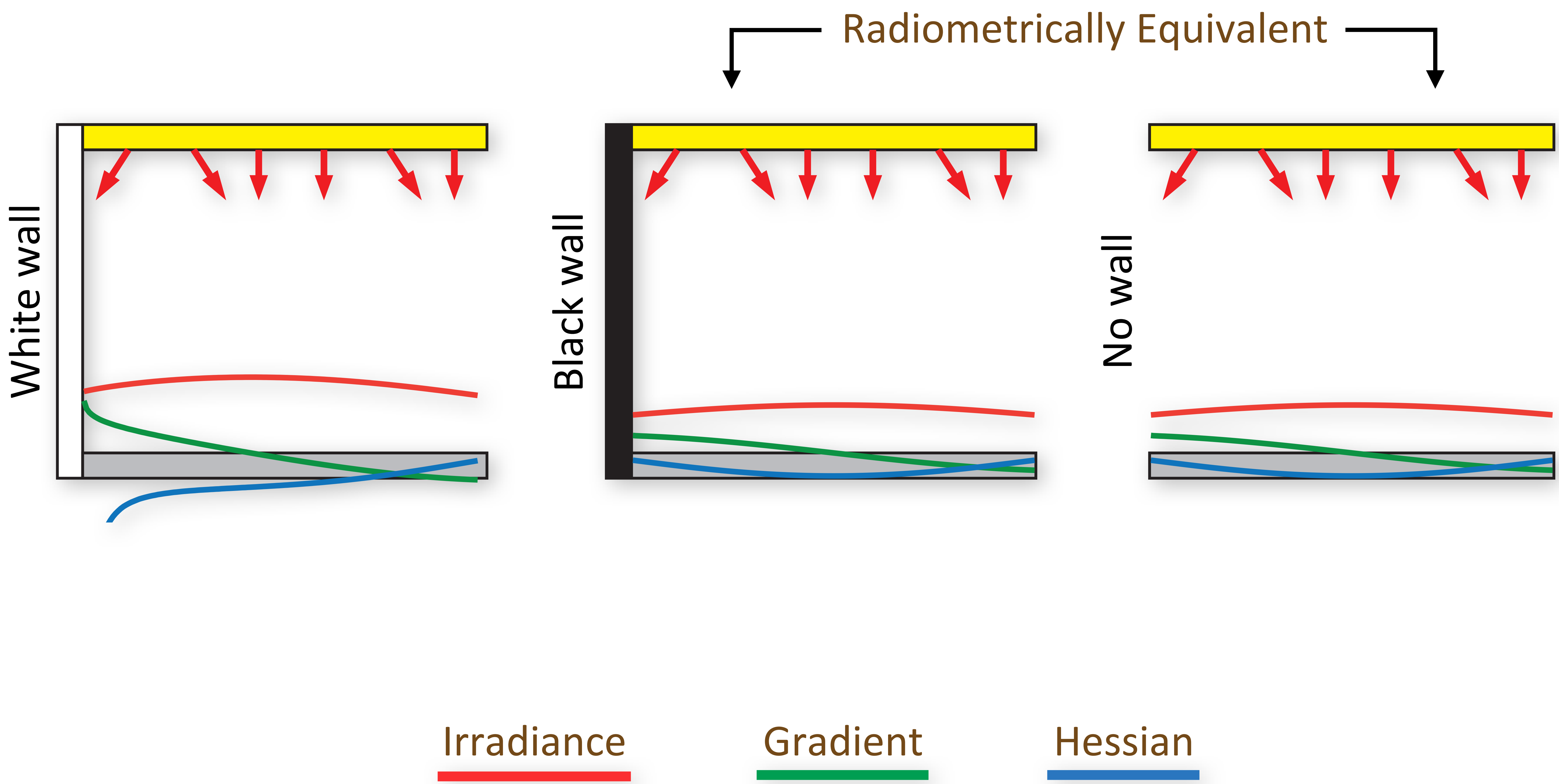
- The other major component of irradiance caching is the so-called split-sphere heuristic which dictates the location and density of cache points in the scene
- It sets the radius of cache points inversely proportional to the average distance to nearby objects.
- In essence: near corners and edges, the irradiance is expected to change more rapidly so the radii are small, increasing the caching density in those regions.

Irradiance Caching Test Scenes



- Lets see how this behaves in 2D.
- Here we have an area light at the top, and on the right side we have either a white wall, or a black wall, or no wall at all
- We reconstruct the irradiance on the ground
- Note that the middle and right scene are actually radiometrically identical: having the same irradiance and all derivatives along the floor

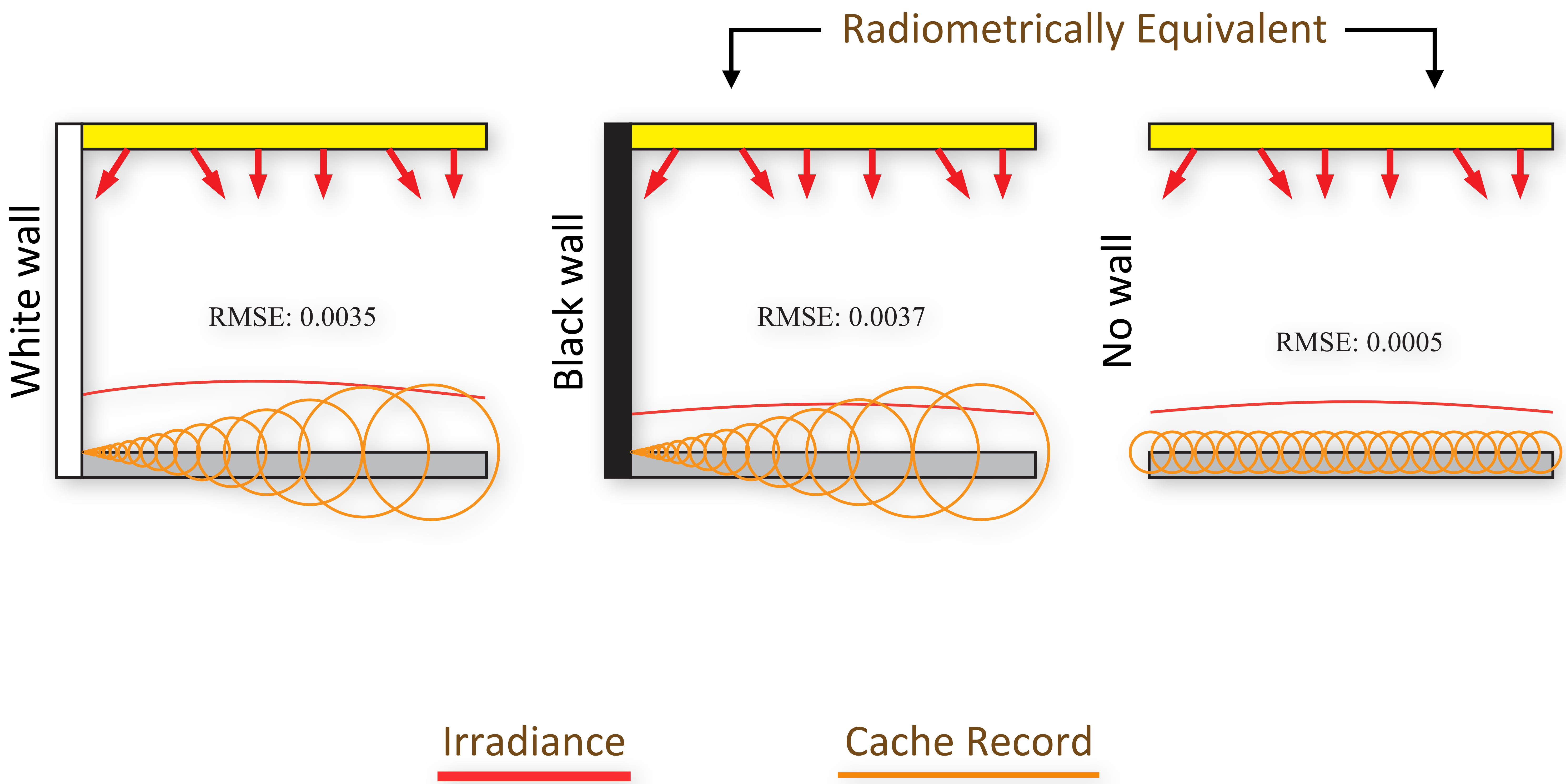
Irradiance Caching Test Scenes



Wednesday, 5 September 12

- Lets see how this behaves in 2D.
- Here we have an area light at the top, and on the right side we have either a white wall, or a black wall, or no wall at all
- We reconstruct the irradiance on the ground
- Note that the middle and right scene are actually radiometrically identical: having the same irradiance and all derivatives along the floor

Split-Sphere



Wednesday, 5 September 12

- When applying irradiance caching with the split-sphere, one thing we immediately notice is that the two equivalent scenes actually get totally different cache point distributions
- Also, the split-sphere is generally too conservative and therefore dedicates far too many cache points in corners and edges, resulting in high reconstruction error

Better Error Control

- This is why many papers have tried to apply fix ups to the split-sphere, but these add more parameters, and don't ultimately solve the underlying problem
- Instead, our goal was to use our 2D theory to come up with a more principled metric from the ground up

Better Error Control

- Many fix-ups possible, but increase complexity

- This is why many papers have tried to apply fix ups to the split-sphere, but these add more parameters, and don't ultimately solve the underlying problem
- Instead, our goal was to use our 2D theory to come up with a more principled metric from the ground up

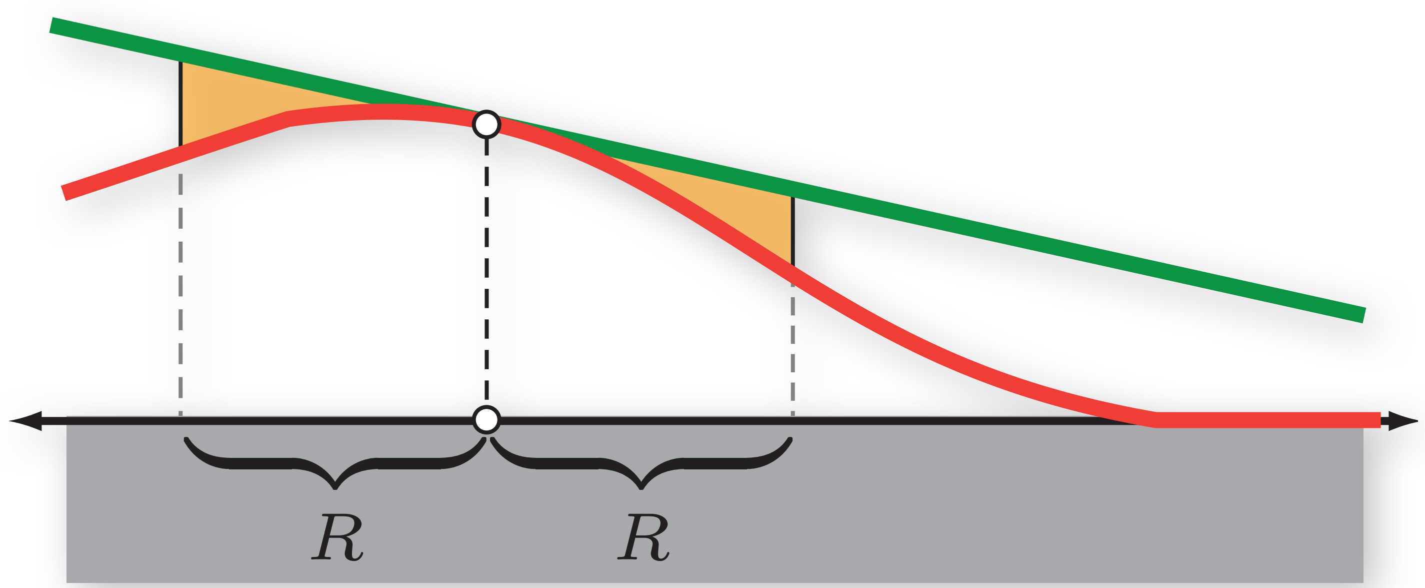
Better Error Control

- Many fix-ups possible, but increase complexity
- **Goal**: create a more principled metric from scratch

- This is why many papers have tried to apply fix ups to the split-sphere, but these add more parameters, and don't ultimately solve the underlying problem
- Instead, our goal was to use our 2D theory to come up with a more principled metric from the ground up

Better Error Control

- **total error** ϵ^t = integrated difference between extrapolated and correct irradiance

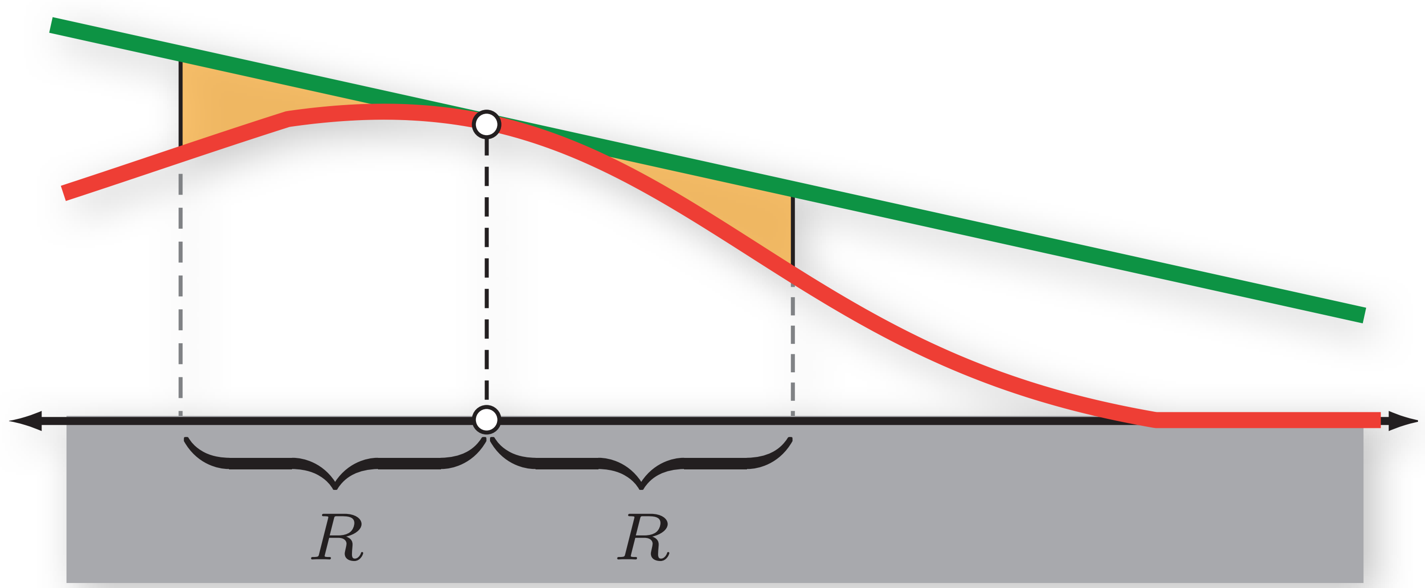


Wednesday, 5 September 12

- We therefore imagined what would be the ideal radius for a cache point.
- Ultimately we are interested in minimizing the error introduced by each cache point to the rendered image
- We can express this mathematically as the integrated difference between the extrapolated and correct irradiance

Better Error Control

- **total error ϵ^t** = integrated difference between extrapolated and correct irradiance



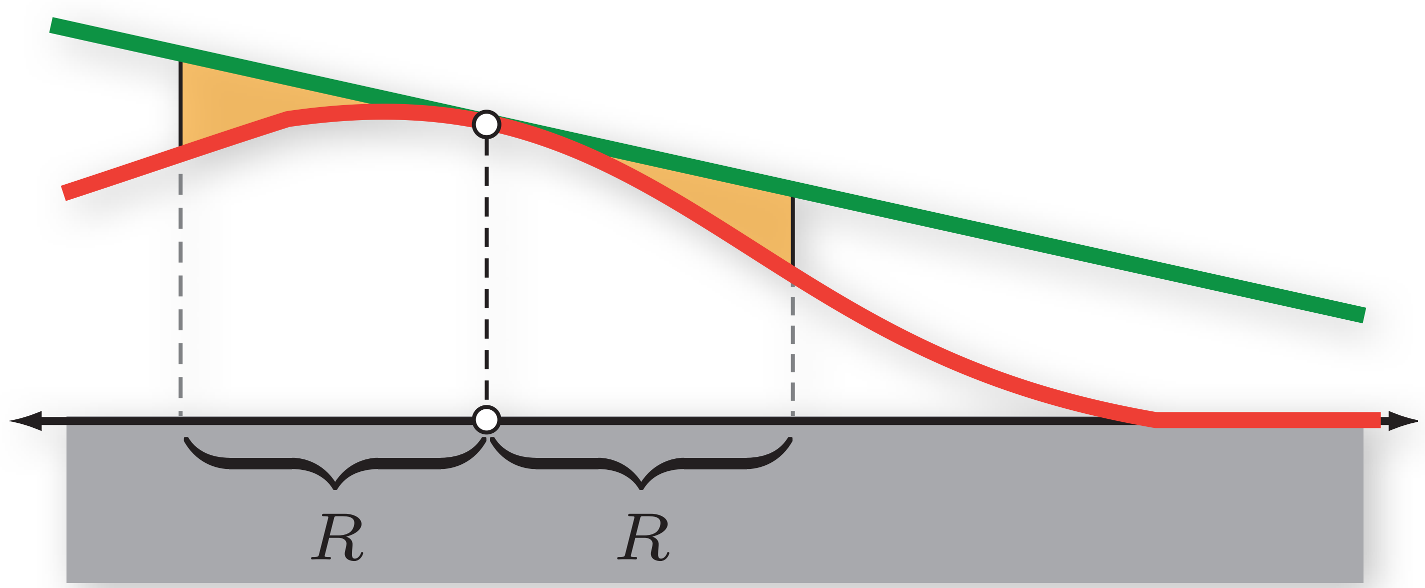
$$\epsilon^t = \int_{-R_i}^{R_i} \left| E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x) \right| \, dx$$

Wednesday, 5 September 12

- We therefore imagined what would be the ideal radius for a cache point.
- Ultimately we are interested in minimizing the error introduced by each cache point to the rendered image
- We can express this mathematically as the integrated difference between the extrapolated and correct irradiance

Better Error Control

- *E'* is 1st-order Taylor extrapolation

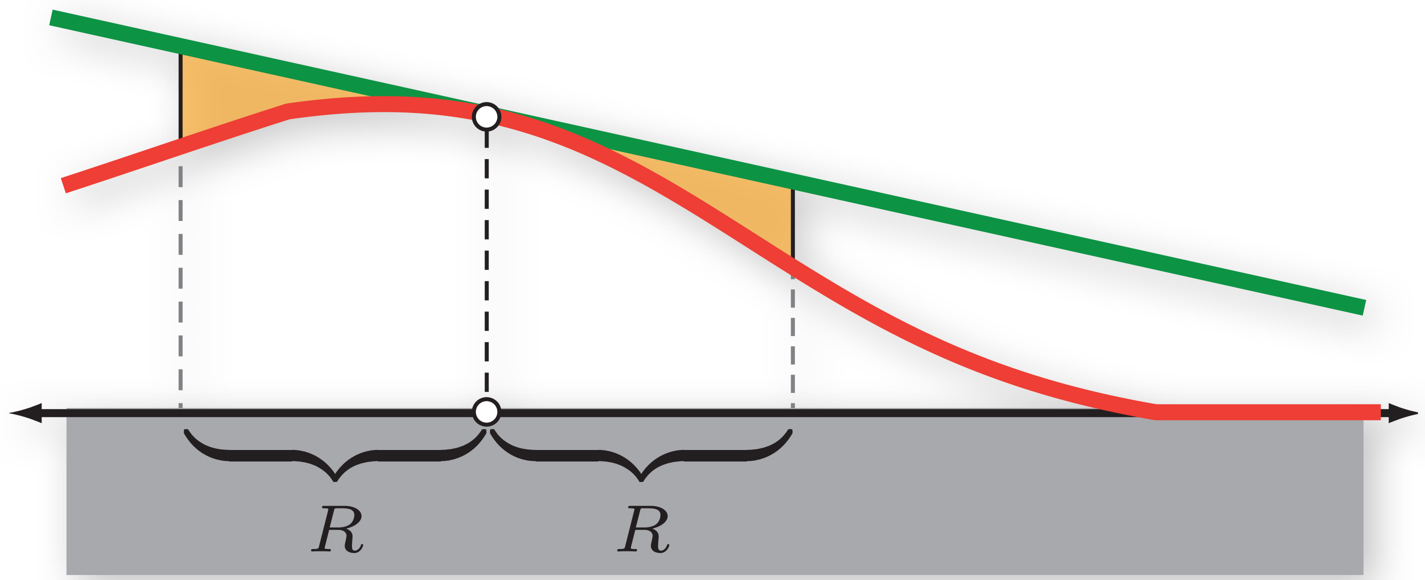


$$\epsilon^t = \int_{-R_i}^{R_i} \left| E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x) \right| dx$$

- If we are using gradient extrapolation, then E' is simply the 1st order taylor extrapolation from the cache point
- However, the true irradiance E, is unknown since this is the quantity we are trying to avoid computing in the first place!

Better Error Control

- *E'* is 1st-order Taylor extrapolation
- *E* is unknown!

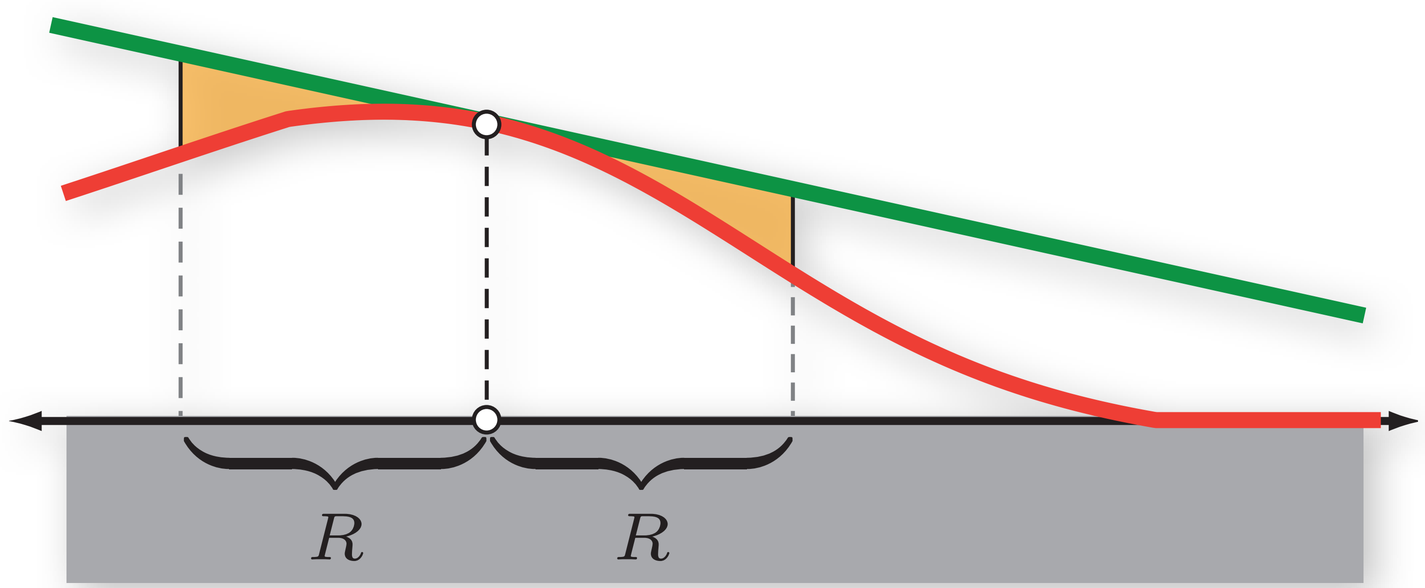


$$\epsilon^t = \int_{-R_i}^{R_i} \left| E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x) \right| \, dx$$

- If we are using gradient extrapolation, then E' is simply the 1st order taylor extrapolation from the cache point
- However, the true irradiance E, is unknown since this is the quantity we are trying to avoid computing in the first place!

Better Error Control

- E' is 1st-order Taylor extrapolation
- E is unknown!

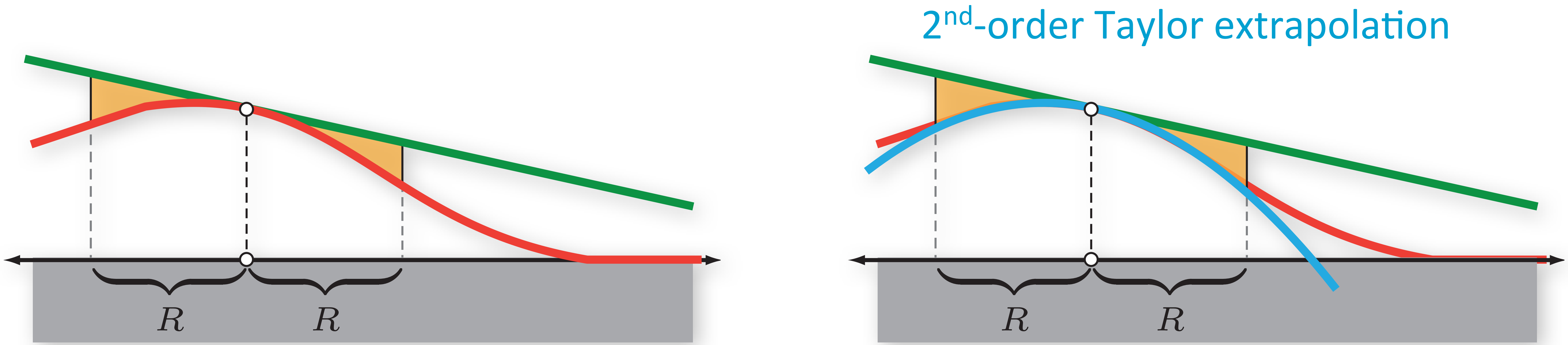


$$\epsilon^t = \int_{-R_i}^{R_i} \left| E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x) \right| \, dx$$

- If we are using gradient extrapolation, then E’ is simply the 1st order taylor extrapolation from the cache point
- However, the true irradiance E, is unknown since this is the quantity we are trying to avoid computing in the first place!

Better Error Control

- E' is 1st-order Taylor extrapolation
- E is unknown!



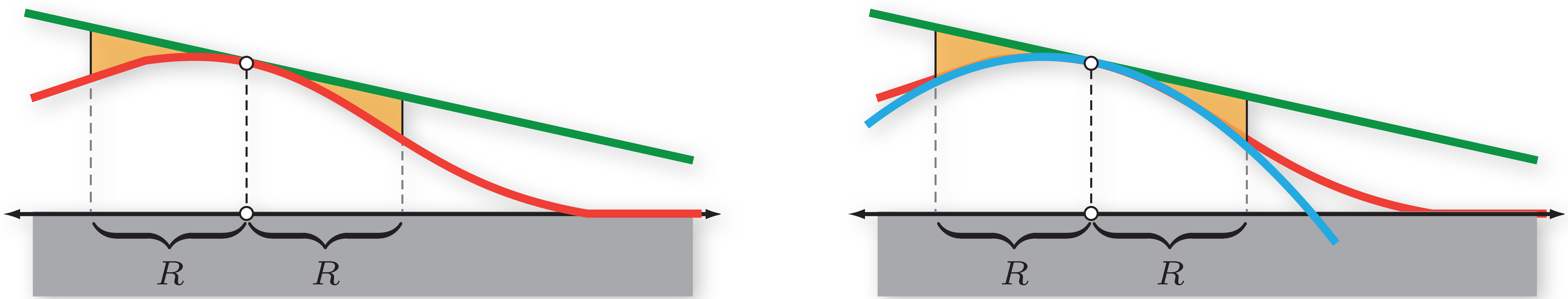
$$\epsilon^t = \int_{-R_i}^{R_i} \left| E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x) \right| dx$$

Wednesday, 5 September 12

- However, we can compute a second derivative at the cache point, and we have already seen that the 2nd-order taylor extrapolation is significantly more accurate.

Hessian-based Error Control

- E' is 1st-order Taylor extrapolation
- 2nd-order Taylor extrapolation approximates E



$$\epsilon^t = \int_{-R_i}^{R_i} \left| E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x) \right| \, dx \quad \approx \quad \hat{\epsilon}^t = \frac{1}{2} \int_{-R_i}^{R_i} \left| x \, \mathbf{H}_{\mathbf{x}}(E_i) \, x \right| \, dx$$

Wednesday, 5 September 12

- We therefore propose to use the 2nd order Taylor extrapolation as an oracle for the true irradiance in the local region
- We can see that the integrated orange regions look quite similar on the left and right, but on the right this is completely defined by the irradiance Hessian at the cache point

Hessian-based Error Control

$$\hat{\epsilon}^t = \frac{1}{2} \int_{-R_i}^{R_i} |x \, \mathbf{H_x}(E_i) \, x| \; dx$$

- We can therefore easily compute this integral since its just a simple polynomial, where hx here is just the scalar second derivative in 2D
- By enforcing a certain error threshold and solving this equation for the radius, we see that the radius should be related to the cube root of the reciprocal second derivative

Hessian-based Error Control

$$\hat{\epsilon}^t = \frac{1}{2}\int_{-R_i}^{R_i} |x\,\mathbf{H}_{\mathbf{x}}(E_i)\,x|\;dx \;=\frac{1}{3}|h_{\mathbf{x}}(E_i)|\,R_i^3$$

- We can therefore easily compute this integral since its just a simple polynomial, where hx here is just the scalar second derivative in 2D
- By enforcing a certain error threshold and solving this equation for the radius, we see that the radius should be related to the cube root of the reciprocal second derivative

Hessian-based Error Control

$$\hat{\epsilon}^t = \frac{1}{2}\int_{-R_i}^{R_i} |x\,\mathbf{H}_{\mathbf{x}}(E_i)\,x|\;dx \;=\frac{1}{3}\boxed{|h_{\mathbf{x}}(E_i)|}R_i^3$$

second derivative

- We can therefore easily compute this integral since its just a simple polynomial, where hx here is just the scalar second derivative in 2D
- By enforcing a certain error threshold and solving this equation for the radius, we see that the radius should be related to the cube root of the reciprocal second derivative

Hessian-based Error Control

$$\hat{\epsilon}^t = \frac{1}{2}\int_{-R_i}^{R_i} |x\,\mathbf{H}_{\mathbf{x}}(E_i)\,x|\;dx \;=\; \frac{1}{3}\boxed{|h_{\mathbf{x}}(E_i)|}R_i^3$$

second derivative

- fix ϵ^t, solve for R_i

- We can therefore easily compute this integral since its just a simple polynomial, where hx here is just the scalar second derivative in 2D
- By enforcing a certain error threshold and solving this equation for the radius, we see that the radius should be related to the cube root of the reciprocal second derivative

Hessian-based Error Control

$$\hat{\epsilon}^t = \frac{1}{2}\int_{-R_i}^{R_i} |x\,\mathbf{H}_{\mathbf{x}}(E_i)\,x|\;dx \;=\; \frac{1}{3}\boxed{|h_{\mathbf{x}}(E_i)|}R_i^3$$

second derivative

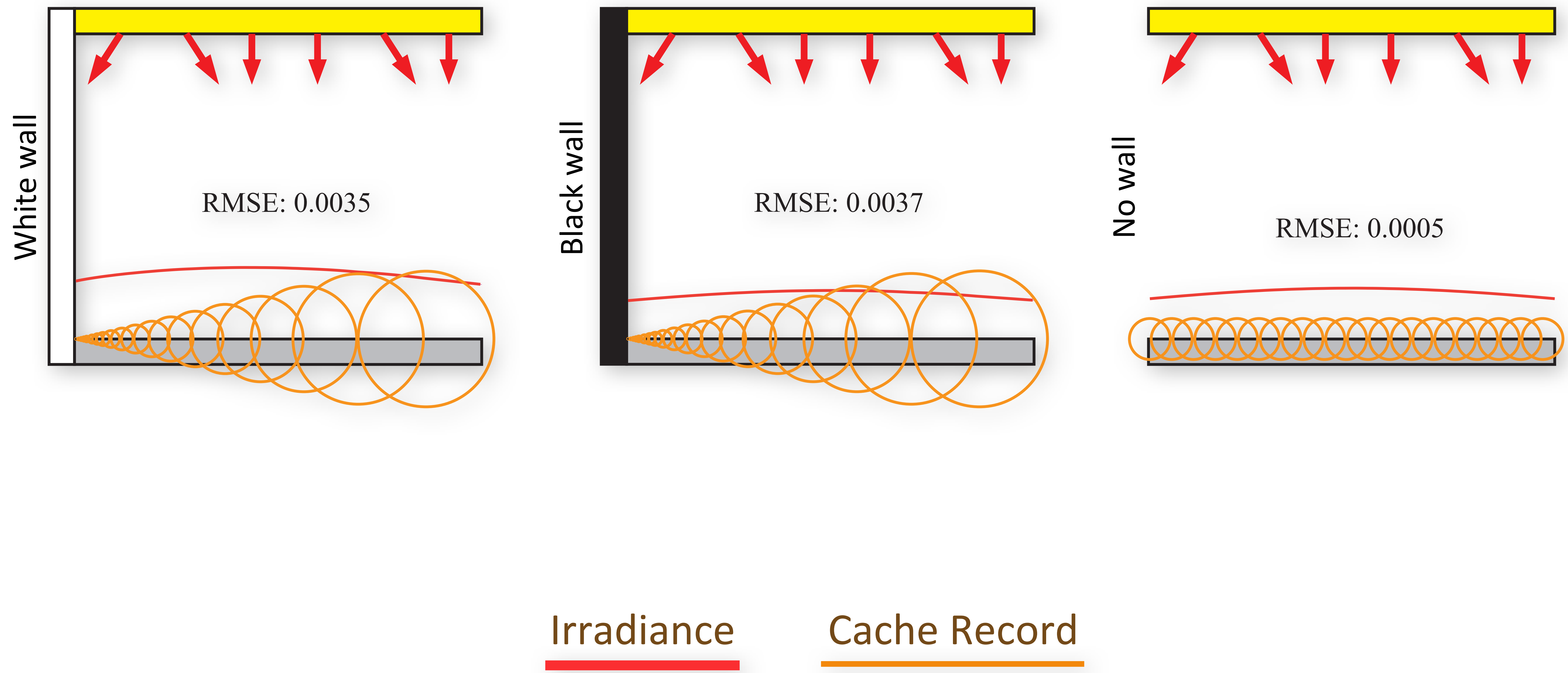
- fix ϵ^t, solve for R_i

$$\boxed{R_i = \sqrt[3]{\frac{3\hat{\epsilon}^t}{|h_{\mathbf{x}}(E_i)|}}}$$

Wednesday, 5 September 12

- We can therefore easily compute this integral since its just a simple polynomial, where hx here is just the scalar second derivative in 2D
- By enforcing a certain error threshold and solving this equation for the radius, we see that the radius should be related to the cube root of the reciprocal second derivative

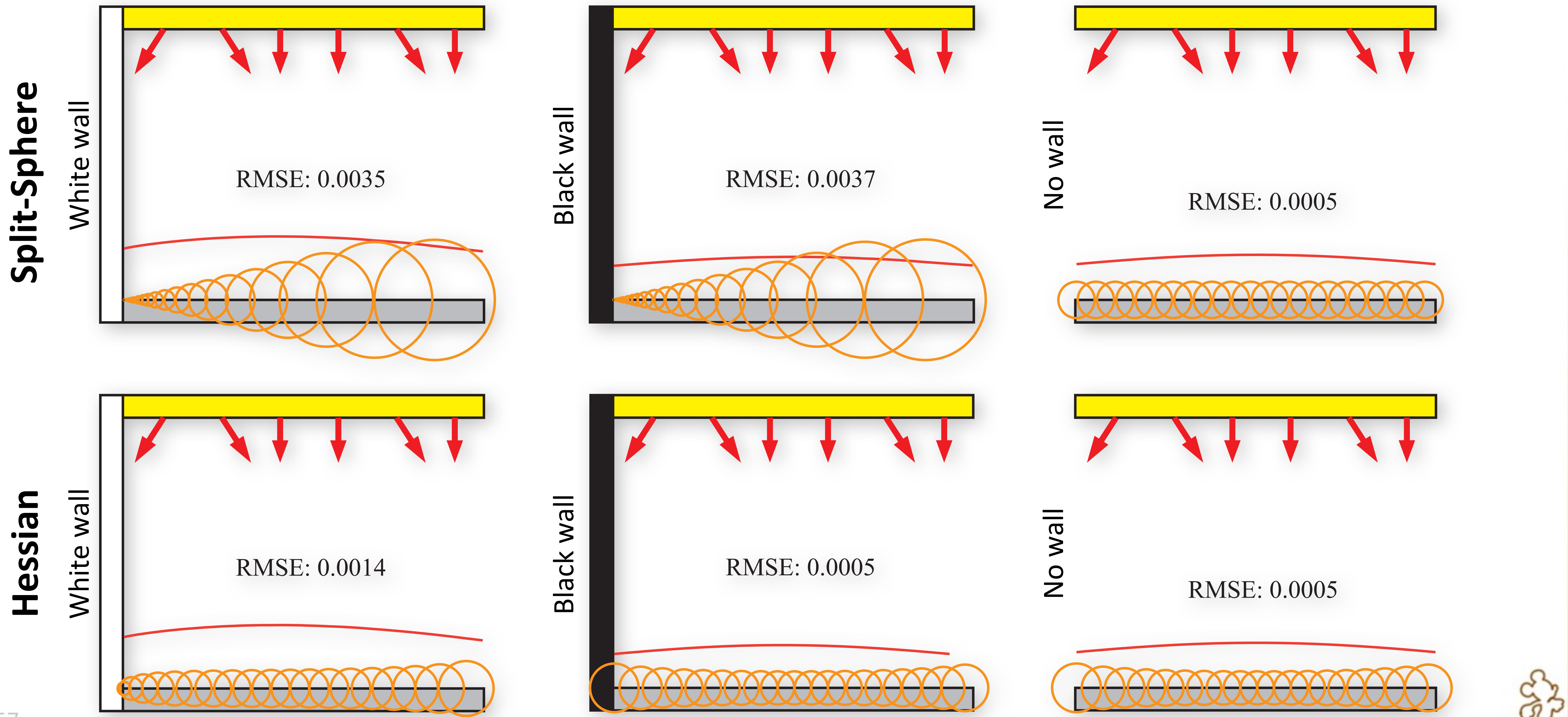
Split-Sphere



Wednesday, 5 September 12

- We can therefore use this expression instead of the split-sphere for our 2D scene

Split-Sphere vs Hessian



Wednesday, 5 September 12

- And we can see that many of the problems with the split-sphere are eliminated
- Firstly, the two radiometrically-identical scenes now have identical cache point distributions
- Also, notice that the RMS error in the reconstruction has gone down by as much as a factor of 7

Hessian-based Error Control

2D

$$R_i = \sqrt[3]{\frac{3\hat{\epsilon}^t}{|h_{\mathbf{x}}(E_i)|}}$$

- By just following through with the same derivations, we can generalize this idea to 3D, by using the 3D irradiance Hessian
- Where now there is a forth root, and lamba 1 is simply the maximum eigenvalue of the irradiance Hessian matrix

Hessian-based Error Control

2D

$$R_i = \sqrt[3]{\frac{3\hat{\epsilon}^t}{|h_{\mathbf{x}}(E_i)|}}$$

$\longrightarrow$

3D

$$R_i = \sqrt[4]{\frac{4\hat{\epsilon}^t}{\pi\lambda_1}}$$

- By just following through with the same derivations, we can generalize this idea to 3D, by using the 3D irradiance Hessian
- Where now there is a forth root, and lamba 1 is simply the maximum eigenvalue of the irradiance Hessian matrix

Hessian-based Error Control

2D

$$R_i = \sqrt[3]{\frac{3\hat{\epsilon}^t}{|h_{\mathbf{x}}(E_i)|}}$$

$\longrightarrow$

3D

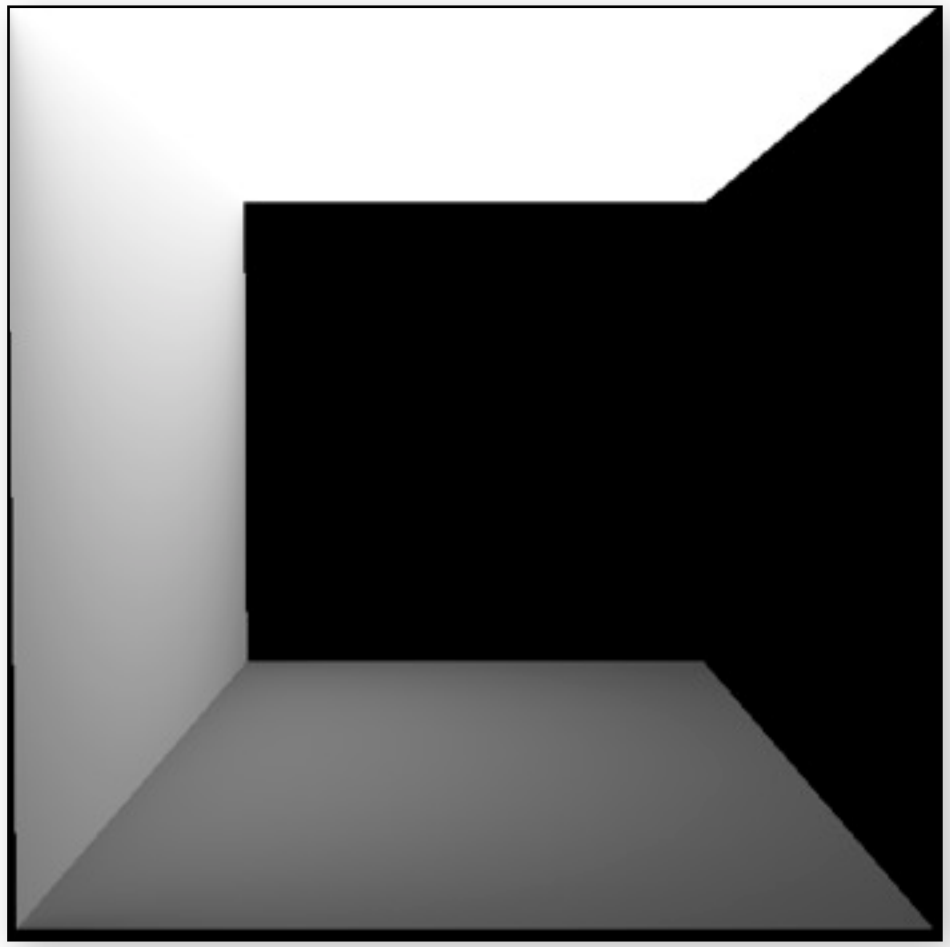
$$R_i = \sqrt[4]{\frac{4\hat{\epsilon}^t}{\pi\boxed{\lambda_1}}}$$

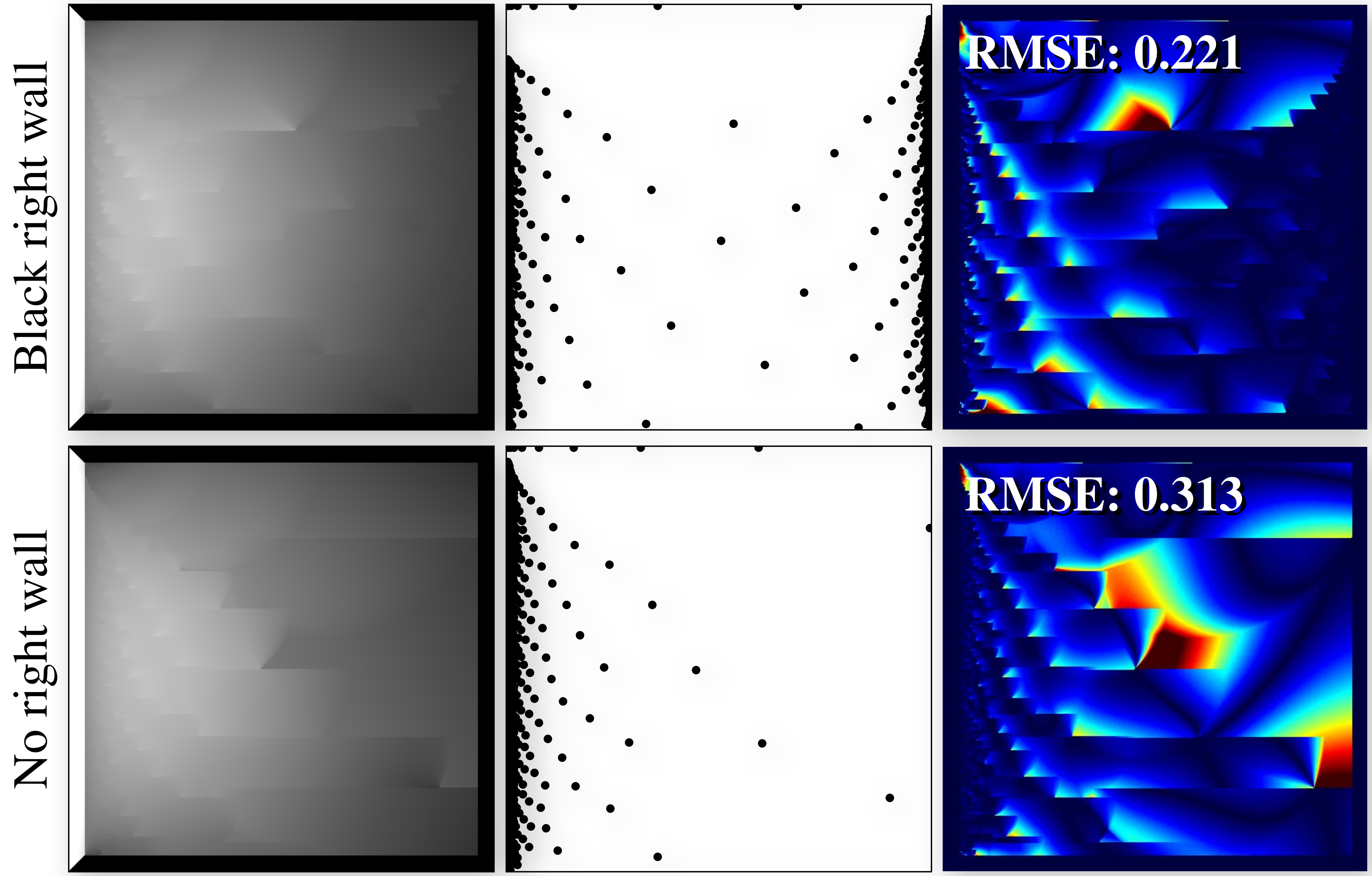
max eigenvalue of Hessian

- By just following through with the same derivations, we can generalize this idea to 3D, by using the 3D irradiance Hessian
- Where now there is a forth root, and lamba 1 is simply the maximum eigenvalue of the irradiance Hessian matrix

Split-Sphere

Scene

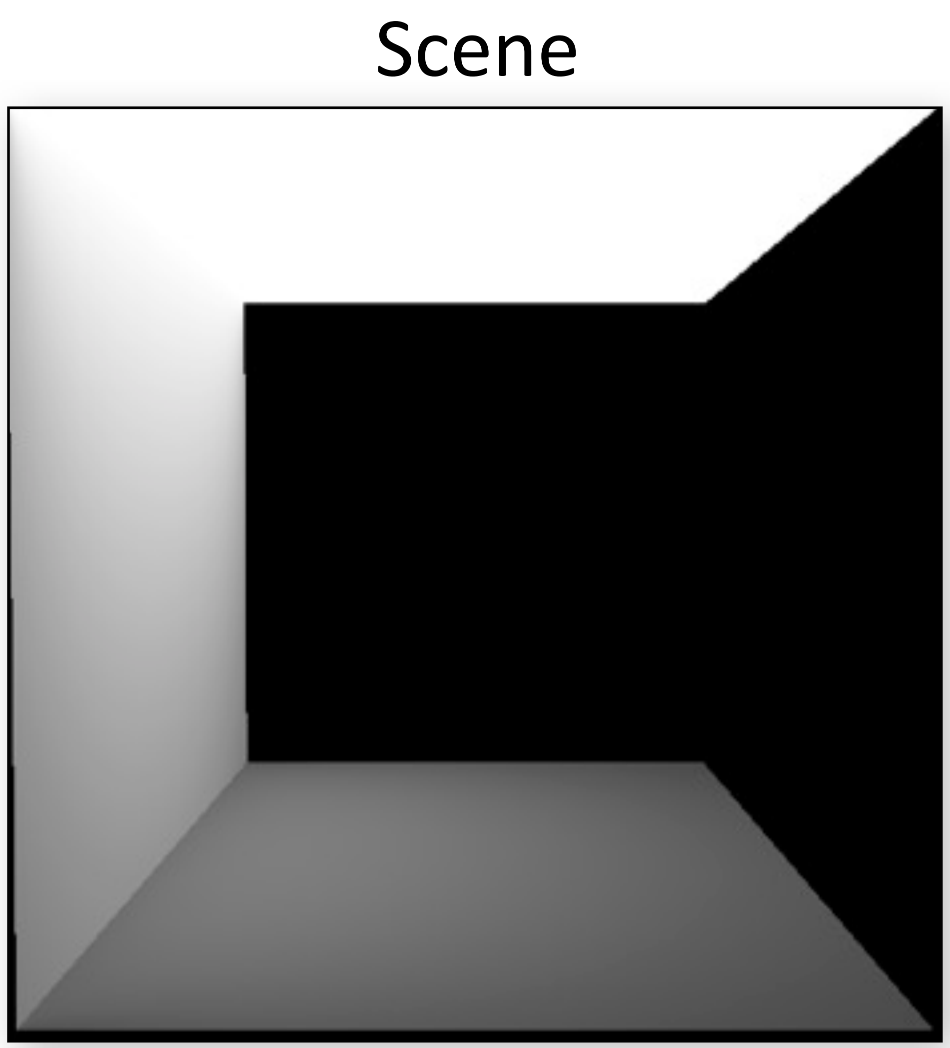


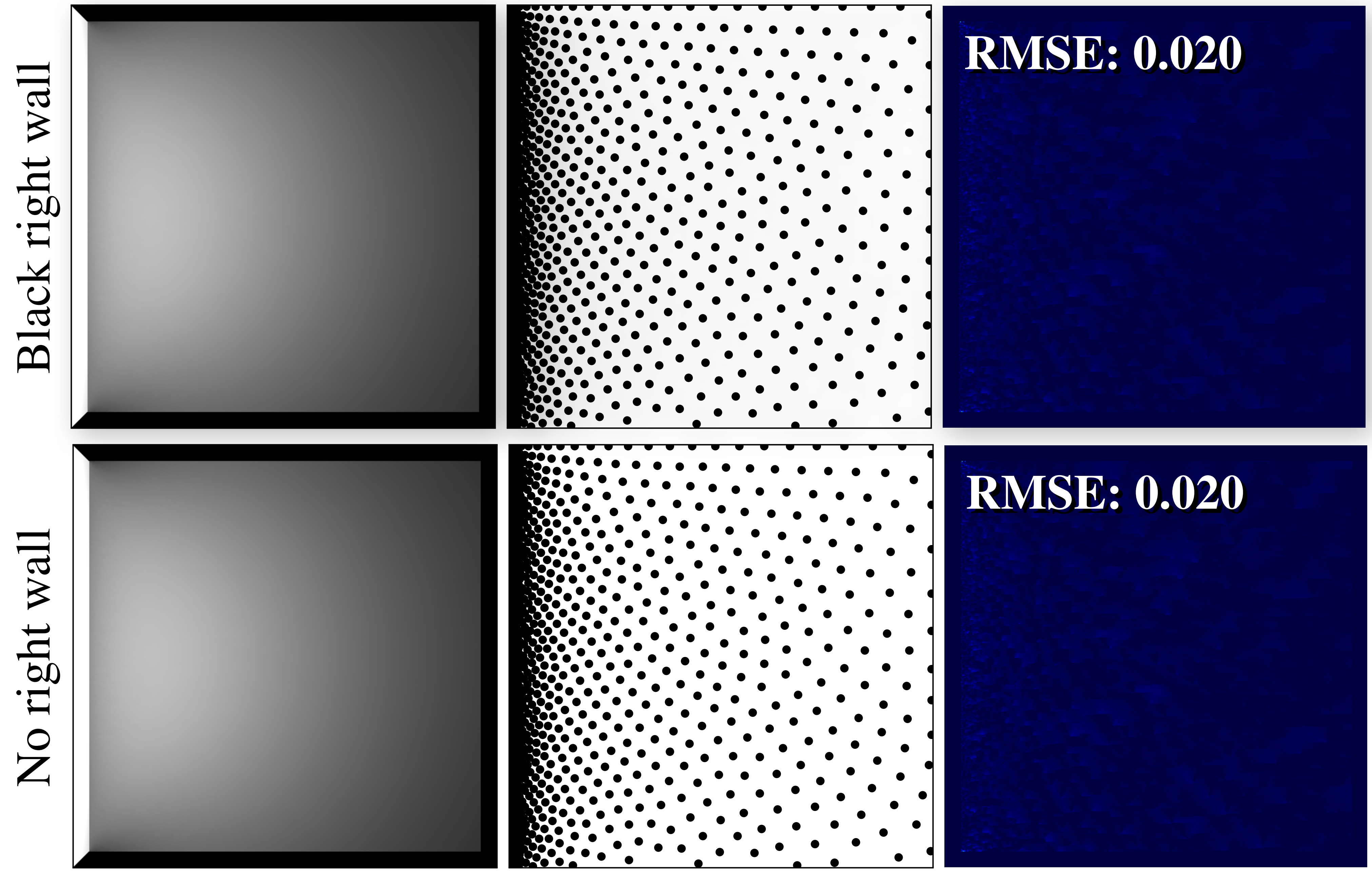


Wednesday, 5 September 12

- Again, in a 3D scene, the split-sphere shows the aggressive behavior at edges

Radiometric Hessian

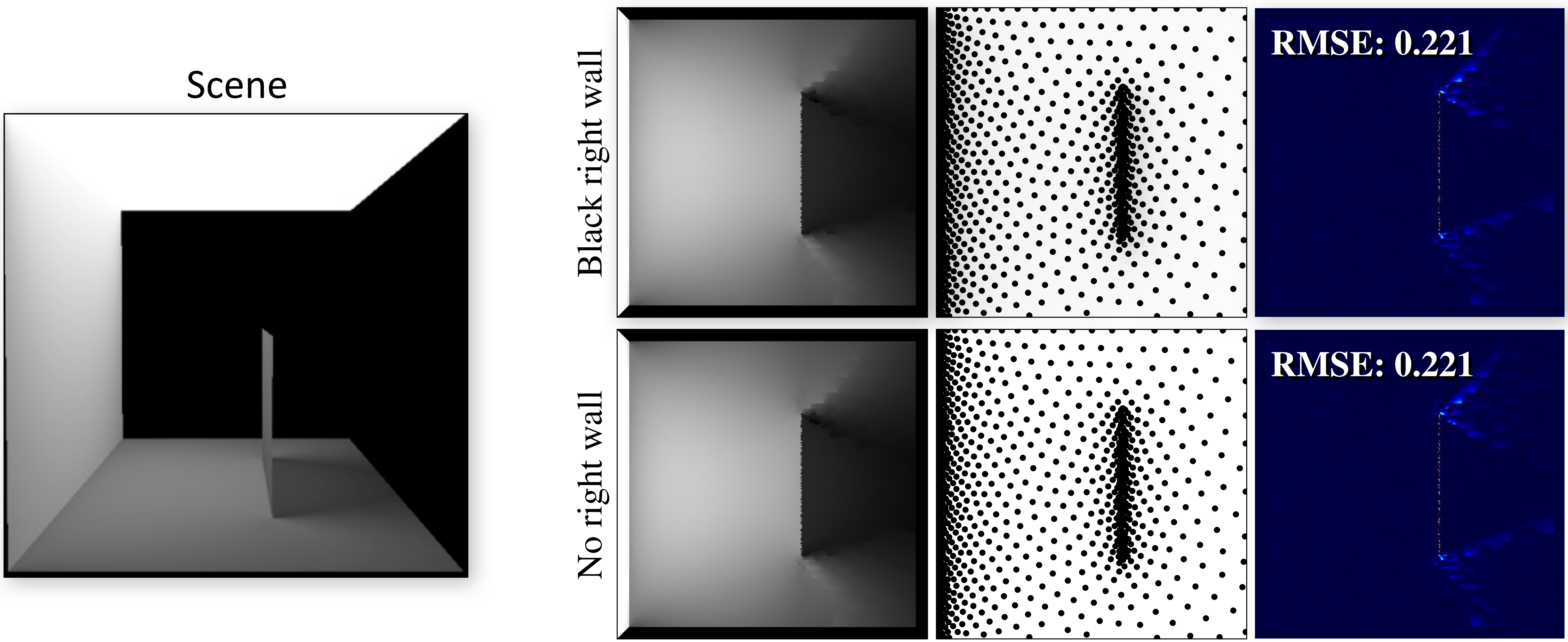




Wednesday, 5 September 12

- Whereas, with the same number of cache points, we can obtain a much nicer distribution with the Hessian approach, which reduces the reconstructed error by an order of magnitude

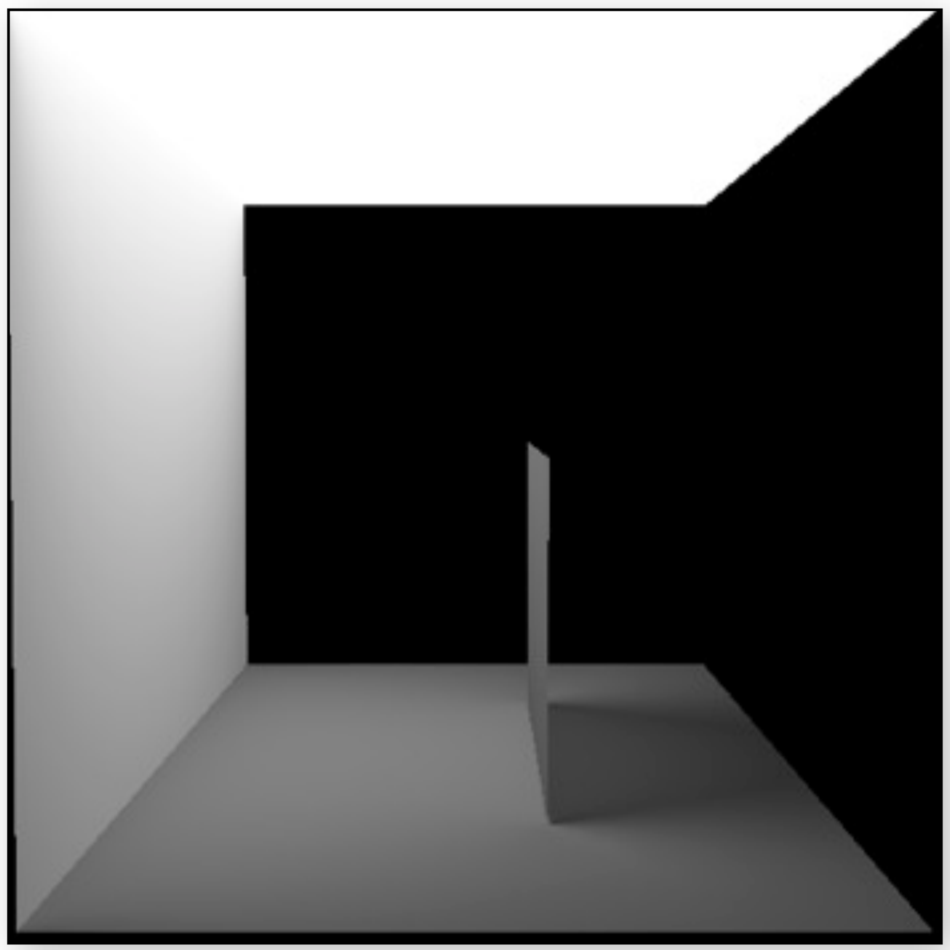
Radiometric Hessian

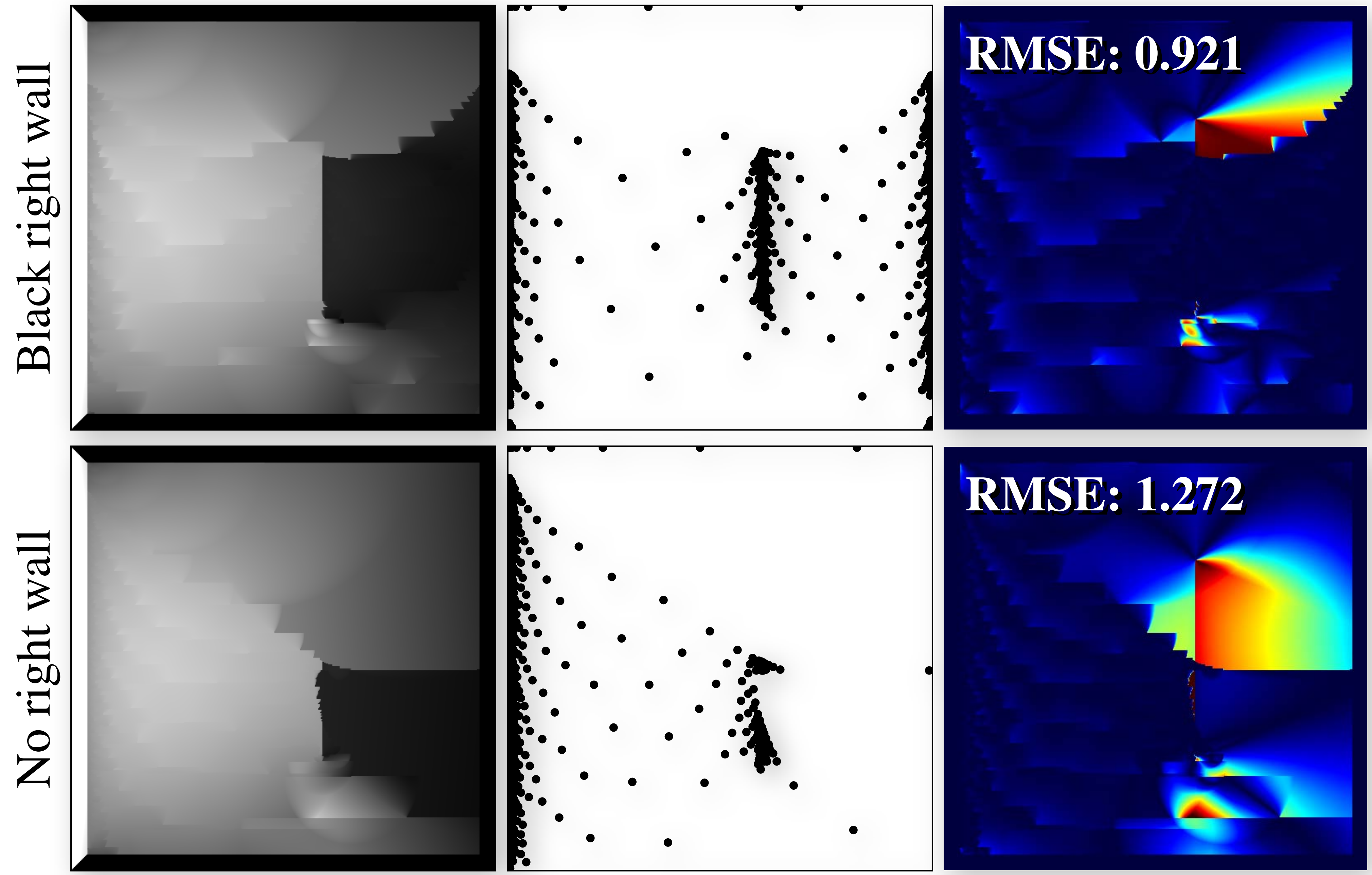


Wednesday, 5 September 12

- Here is a simple modification where an occluder has been introduce
- And even though our arc-length 3D Hessian ignores visibility changes, when used as a error control method it still significantly out-performs the split-sphere without having to enforce minimum radii

Split-Sphere





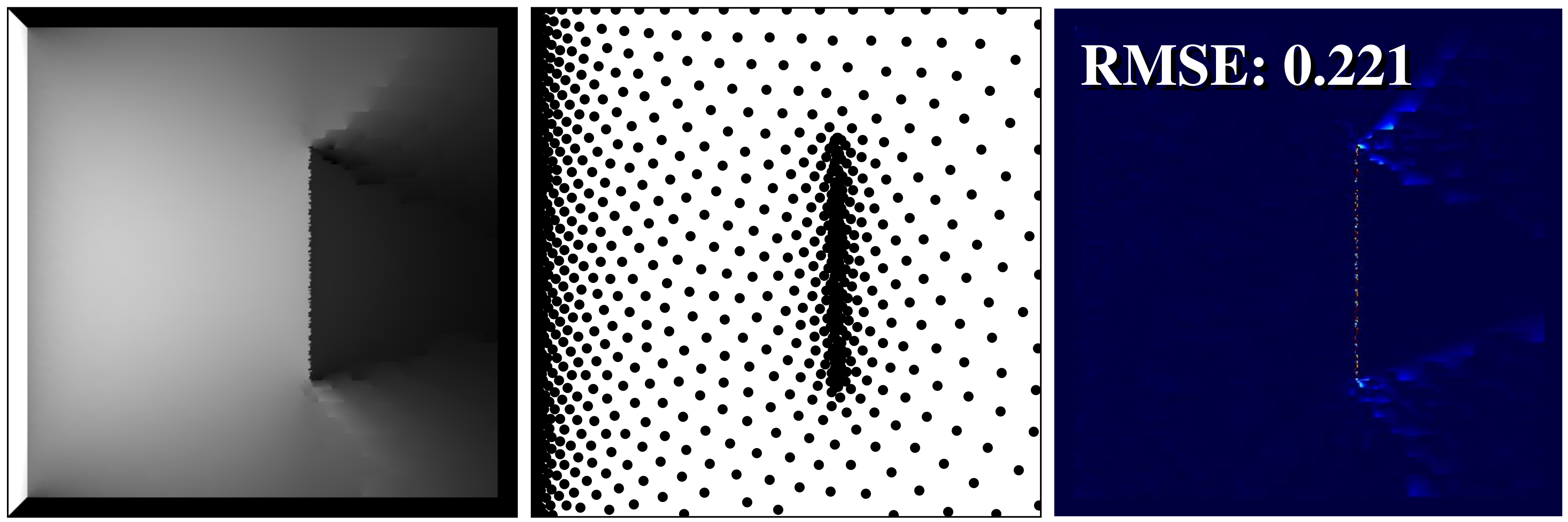
Wednesday, 5 September 12

- And even though our arc-length 3D Hessian ignores visibility changes, when used as a error control method it still significantly out-performs the split-sphere without having to enforce minimum radii

Radiometric Hessian (Failure Case)

- Purely radiometric approach can fail

grey occluder



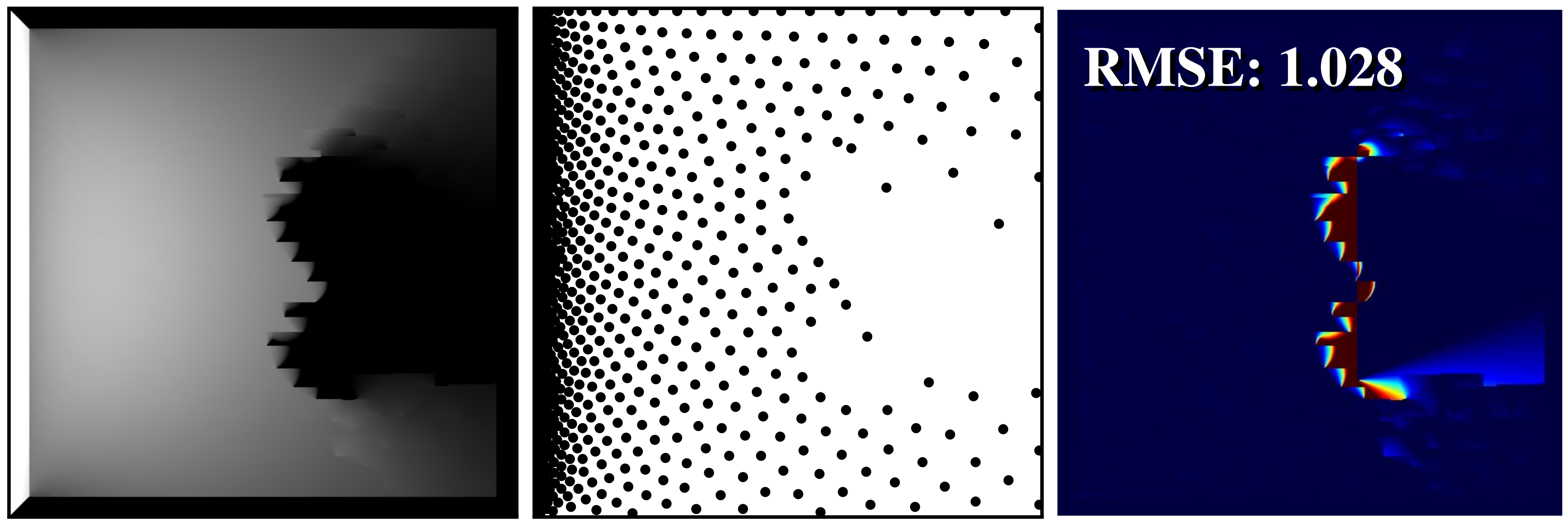
- However, a purely radiometric approach will always have common failure cases, for instance if we simply change the blocker to be perfectly black, we can get in situations where the irradiance and all its derivatives are 0, leading to infinite radii
- The solution we propose for this is to use a conservative lower bound on the radiance returned by each final gather ray when computing the Hessian.
- And, because this ends up being proportional to the integrated hessian of the geometry term, we call this the Geometric Hessian

Radiometric Hessian (Failure Case)

- Purely radiometric approach can fail

$$\mathbf{H}_{\mathbf{x}}(E_{3D}) \approx \int_A L\,V\,\mathbf{H}_{\mathbf{x}}(G_{3D})\,da(\mathbf{y})$$

black occluder



- However, a purely radiometric approach will always have common failure cases, for instance if we simply change the blocker to be perfectly black, we can get in situations where the irradiance and all its derivatives are 0, leading to infinite radii
- The solution we propose for this is to use a conservative lower bound on the radiance returned by each final gather ray when computing the Hessian.
- And, because this ends up being proportional to the integrated hessian of the geometry term, we call this the Geometric Hessian

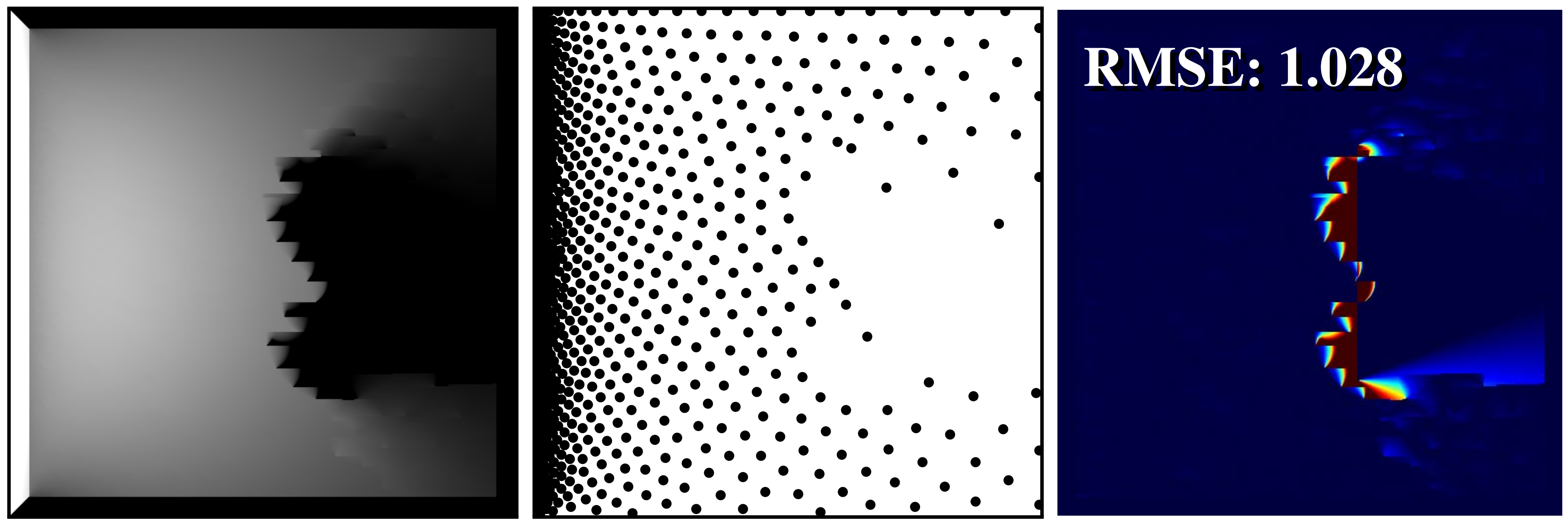
Radiometric Hessian (Failure Case)

- Purely radiometric approach can fail

$$\mathbf{H}_{\mathbf{x}}(E_{3D}) \approx \int_A \boxed{L\,V}\,\mathbf{H}_{\mathbf{x}}(G_{3D})\,da(\mathbf{y})$$

could be zero!

black occluder



Wednesday, 5 September 12

- However, a purely radiometric approach will always have common failure cases, for instance if we simply change the blocker to be perfectly black, we can get in situations where the irradiance and all its derivatives are 0, leading to infinite radii
- The solution we propose for this is to use a conservative lower bound on the radiance returned by each final gather ray when computing the Hessian.
- And, because this ends up being proportional to the integrated hessian of the geometry term, we call this the Geometric Hessian

Geometric Hessian

- Purely radiometric approach can fail

$$\mathbf{H}_{\mathbf{x}}(E_{3D}) \approx \int_A L\,V\,\mathbf{H}_{\mathbf{x}}(G_{3D})\;da(\mathbf{y})$$

- Solution:

- However, a purely radiometric approach will always have common failure cases, for instance if we simply change the blocker to be perfectly black, we can get in situations where the irradiance and all its derivatives are 0, leading to infinite radii
- The solution we propose for this is to use a conservative lower bound on the radiance returned by each final gather ray when computing the Hessian.
- And, because this ends up being proportional to the integrated hessian of the geometry term, we call this the Geometric Hessian

Geometric Hessian

- Purely radiometric approach can fail

$$\mathbf{H_x}(E_{3D}) \approx \int_A L\,V\,\mathbf{H_x}(G_{3D})\,da(\mathbf{y})$$

$$\downarrow$$

- Solution:

$$\mathbf{H_x}(E_{3D}^{\max}) = \int_A L^{\max}\,\mathbf{H_x}(G_{3D})\,da(\mathbf{y})$$

- However, a purely radiometric approach will always have common failure cases, for instance if we simply change the blocker to be perfectly black, we can get in situations where the irradiance and all its derivatives are 0, leading to infinite radii
- The solution we propose for this is to use a conservative lower bound on the radiance returned by each final gather ray when computing the Hessian.
- And, because this ends up being proportional to the integrated hessian of the geometry term, we call this the Geometric Hessian

Geometric Hessian

- Purely radiometric approach can fail

$$\mathbf{H}_{\mathbf{x}}(E_{3D}) \approx \int_A L\,V\,\mathbf{H}_{\mathbf{x}}(G_{3D})\,da(\mathbf{y})$$

$$\downarrow$$

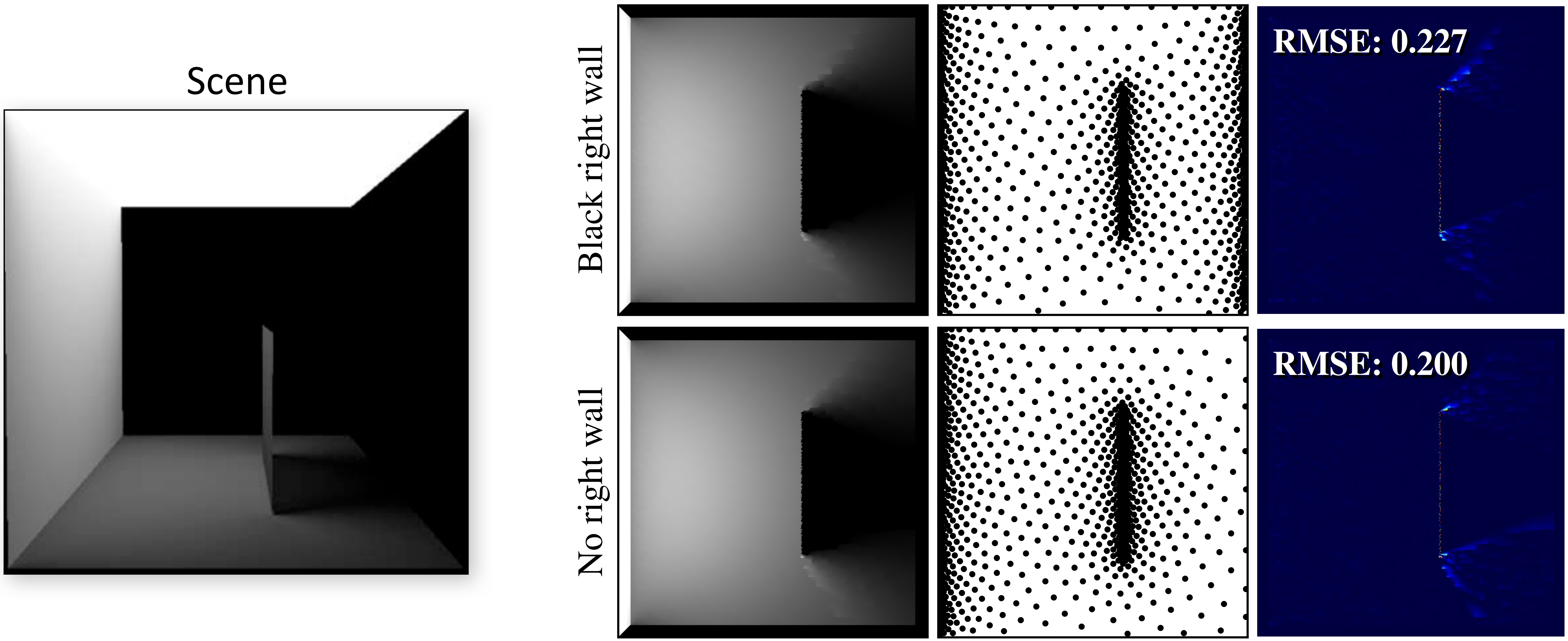
- Solution:

$$\mathbf{H}_{\mathbf{x}}(E_{3D}^{\max}) = \int_A L^{\max}\,\mathbf{H}_{\mathbf{x}}(G_{3D})\,da(\mathbf{y})$$

$$\propto \int_A \mathbf{H}_{\mathbf{x}}(G_{3D})\,da(\mathbf{y})$$

- However, a purely radiometric approach will always have common failure cases, for instance if we simply change the blocker to be perfectly black, we can get in situations where the irradiance and all its derivatives are 0, leading to infinite radii
- The solution we propose for this is to use a conservative lower bound on the radiance returned by each final gather ray when computing the Hessian.
- And, because this ends up being proportional to the integrated hessian of the geometry term, we call this the Geometric Hessian

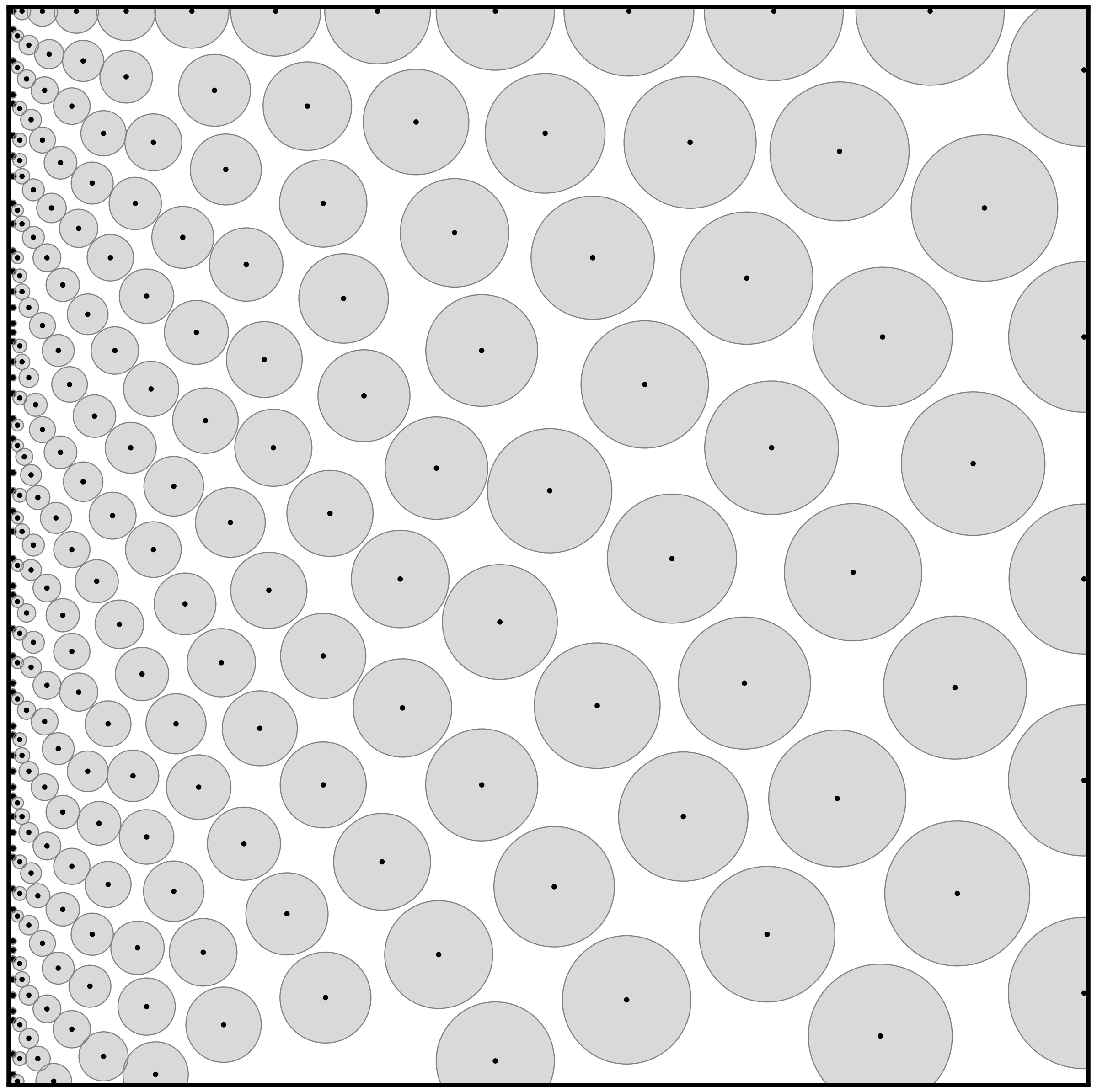
Geometric Hessian

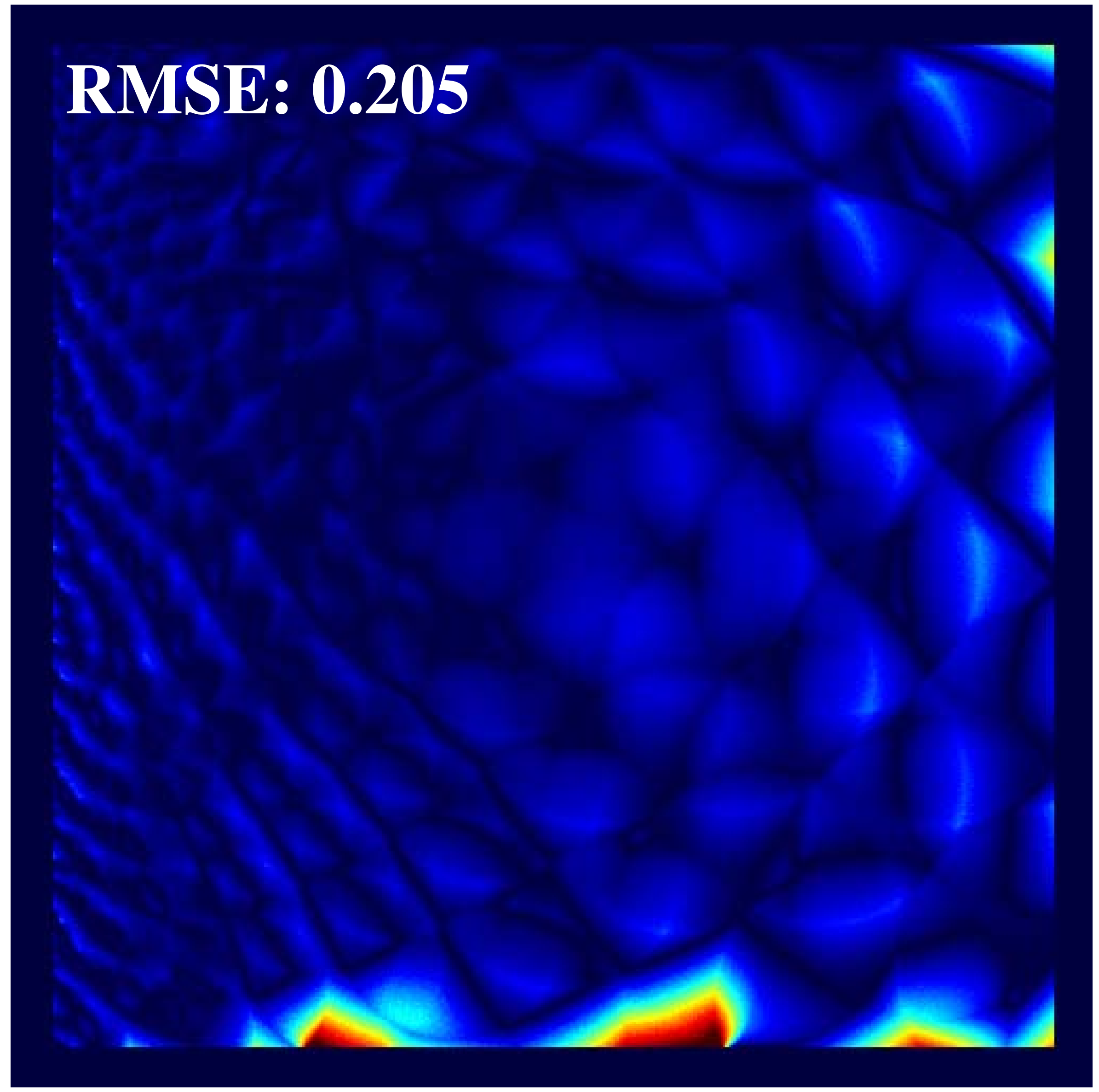


Wednesday, 5 September 12

- Using this instead of the radiometric Hessian eliminates these problems, and provides nice sample distributions even for these failure cases

Isotropic Error Control

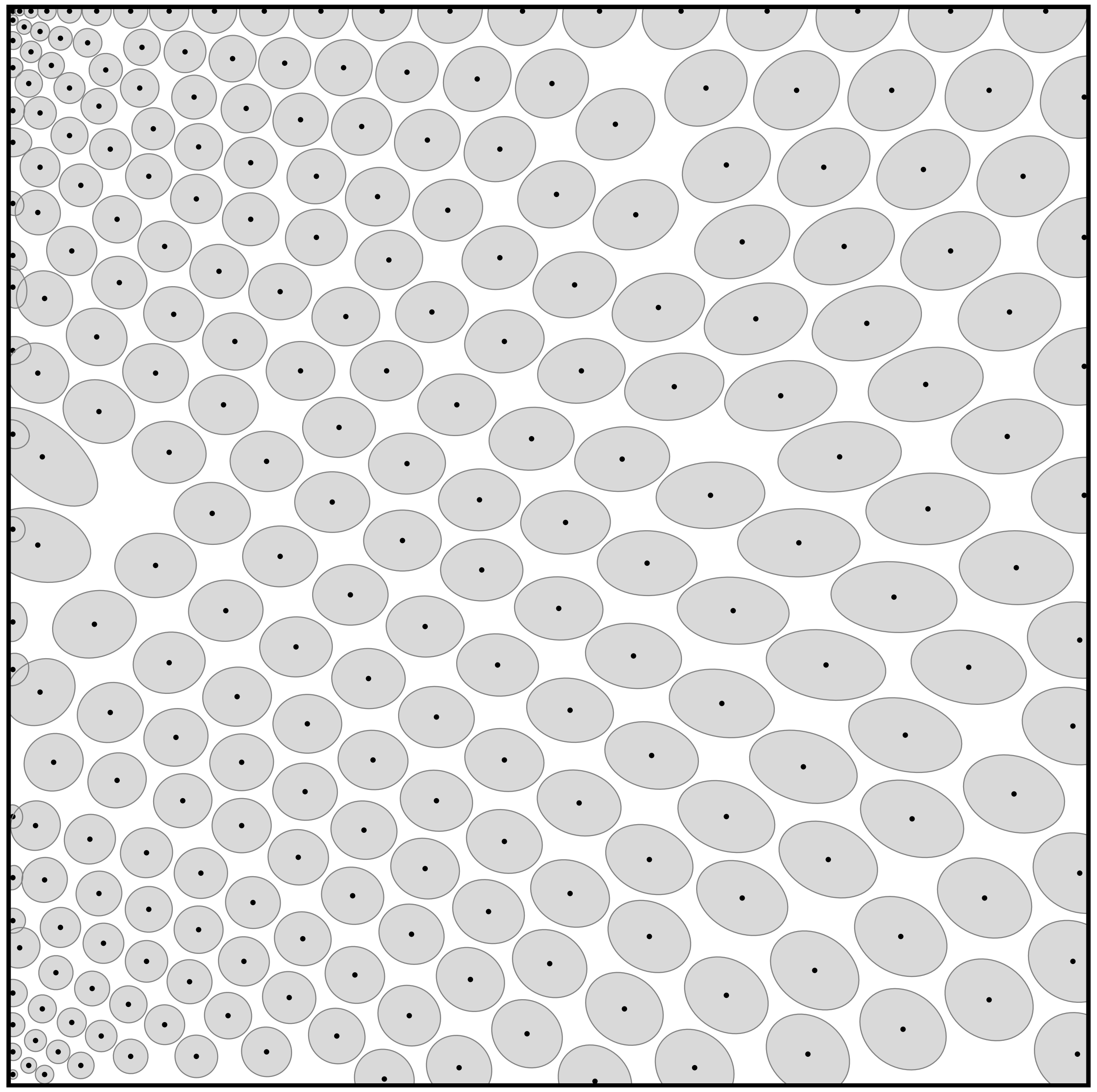


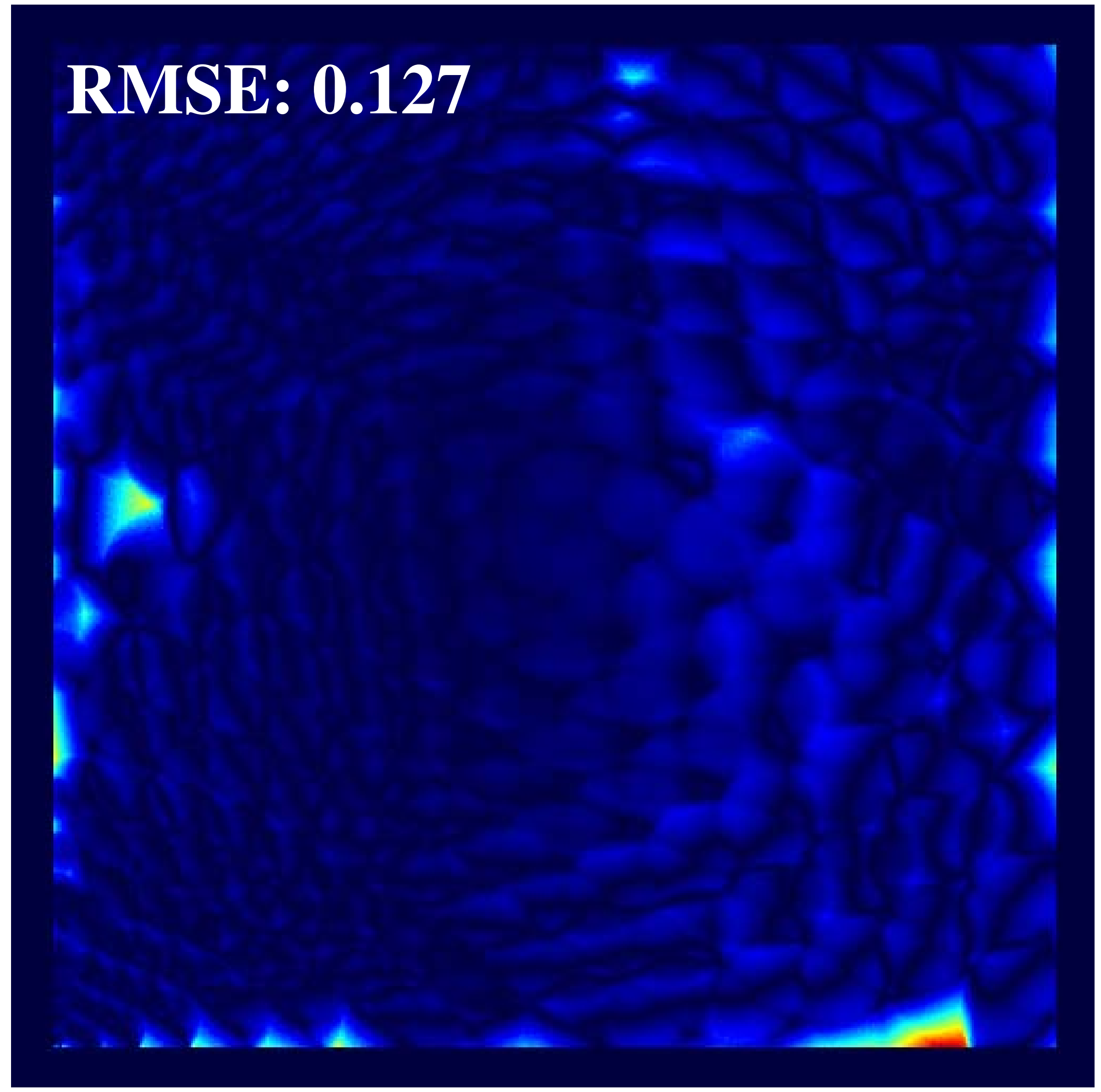


$$R_i = \sqrt[4]{\frac{4\hat{\epsilon}^t}{\pi\lambda_1}}$$

- Finally, since our error is based on the Hessian, in 3D the Hessian retains anisotropic structure.
- We can therefore easily replace our circular cache points, with elliptical cache points, where the ellipse radii are determined by the two eigenvalues of the Hessian and the major and minor axes are the eigenvectors
- This improves the reconstruction quality

Anisotropic Error Control





$$\left(R_i^{\lambda_1},\ R_i^{\lambda_2}\right) = \sqrt[4]{\frac{4\hat{\epsilon}^t}{\pi}} \left(\sqrt[4]{\frac{1}{\boxed{\lambda_1}}},\ \sqrt[4]{\frac{1}{\boxed{\lambda_2}}} \right)$$

- Finally, since our error is based on the Hessian, in 3D the Hessian retains anisotropic structure.
- We can therefore easily replace our circular cache points, with elliptical cache points, where the ellipse radii are determined by the two eigenvalues of the Hessian and the major and minor axes are the eigenvectors
- This improves the reconstruction quality

Summary

- 2D Light Transport is useful
  - Second-order Analysis of 2D GI
  - Hessian enhancements for Irradiance Caching
  - **More examples in the paper**

- So I hope these examples convince you that 2D light transport theory can provide practical insights for 3D rendering
- I do encourage you to read the paper, which covers several more examples including: recursive monte carlo ray tracing, path tracing, and more

Limitations / Future Work

- There are still many things to consider in future work
- Firstly, though we have received positive anecdotal feedback when using our 2D theory for teaching a rendering class, a full user study would really be needed to gauge this benefit
- Also, our theory currently ignores participating media but it would be possible to derive a 2D volume rendering equation for similar benefits
- Finally, our proposed hessian-error control shows promise, but this was just a proof-of-concept. More validations are needed on complex scenes, and, account for visibility in the Hessian is still an open problem

Limitations / Future Work

- Full user study to evaluate benefit for teaching

- There are still many things to consider in future work
- Firstly, though we have received positive anecdotal feedback when using our 2D theory for teaching a rendering class, a full user study would really be needed to gauge this benefit
- Also, our theory currently ignores participating media but it would be possible to derive a 2D volume rendering equation for similar benefits
- Finally, our proposed hessian-error control shows promise, but this was just a proof-of-concept. More validations are needed on complex scenes, and, account for visibility in the Hessian is still an open problem

Limitations / Future Work

- Full user study to evaluate benefit for teaching
- 2D Theory ignores participating media
  - 2D volume rendering equation?

- There are still many things to consider in future work
- Firstly, though we have received positive anecdotal feedback when using our 2D theory for teaching a rendering class, a full user study would really be needed to gauge this benefit
- Also, our theory currently ignores participating media but it would be possible to derive a 2D volume rendering equation for similar benefits
- Finally, our proposed hessian-error control shows promise, but this was just a proof-of-concept. More validations are needed on complex scenes, and, account for visibility in the Hessian is still an open problem

Limitations / Future Work

- Full user study to evaluate benefit for teaching
- 2D Theory ignores participating media
  - 2D volume rendering equation?
- Hessian-error control:
  - Proof-of-concept/more validation needed
  - 3D occlusion-aware Hessian still needed

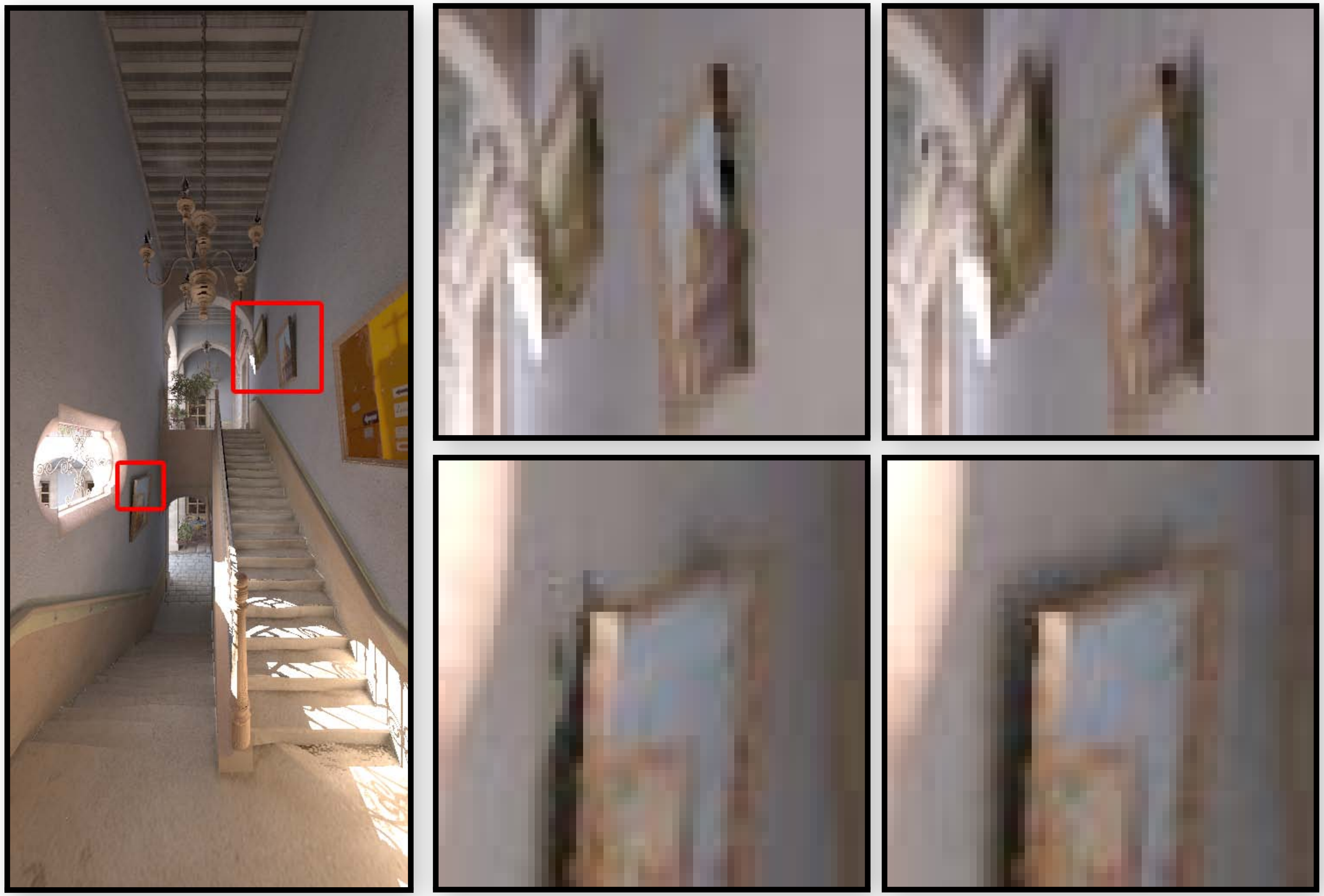
- There are still many things to consider in future work
- Firstly, though we have received positive anecdotal feedback when using our 2D theory for teaching a rendering class, a full user study would really be needed to gauge this benefit
- Also, our theory currently ignores participating media but it would be possible to derive a 2D volume rendering equation for similar benefits
- Finally, our proposed hessian-error control shows promise, but this was just a proof-of-concept. More validations are needed on complex scenes, and, account for visibility in the Hessian is still an open problem

Limitations / Future Work

- Full user study to evaluate benefit for teaching
- 2D Theory ignores participating media
  - 2D volume rendering equation?
- Hessian-error control:
  - Proof-of-concept/more validation needed
  - 3D occlusion-aware Hessian still needed

SIGGRAPH Asia 2012 paper

- There are still many things to consider in future work
- Firstly, though we have received positive anecdotal feedback when using our 2D theory for teaching a rendering class, a full user study would really be needed to gauge this benefit
- Also, our theory currently ignores participating media but it would be possible to derive a 2D volume rendering equation for similar benefits
- Finally, our proposed hessian-error control shows promise, but this was just a proof-of-concept. More validations are needed on complex scenes, and, account for visibility in the Hessian is still an open problem



Reference Bounded Split-Sphere Occlusion Hessian

SIGGRAPH Asia 2012
(to appear)

Wednesday, 5 September 12

- Here is just a quick teaser, which shows that using an occlusion-aware version of our method, with some further enhancements, we can handle complex scenes like this, and resolve indirect illumination much more robustly than the split-sphere.

Acknowledgements

- Derek Nowrouzezahrai
- Peter-Pike Sloan
- DFG, IRTG 1328

- I'd like to thank the following people and grants, and also you for your attention.
- I'd be happy to answer questions

Questions