



Second-Order Occlusion-Aware Volumetric Radiance Caching

Julio Marco¹

Adrian Jarabo¹

Wojciech Jarosz²

Diego Gutierrez¹

¹ Universidad de Zaragoza, I3A

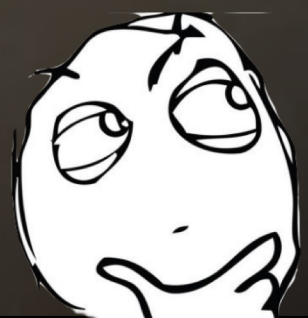
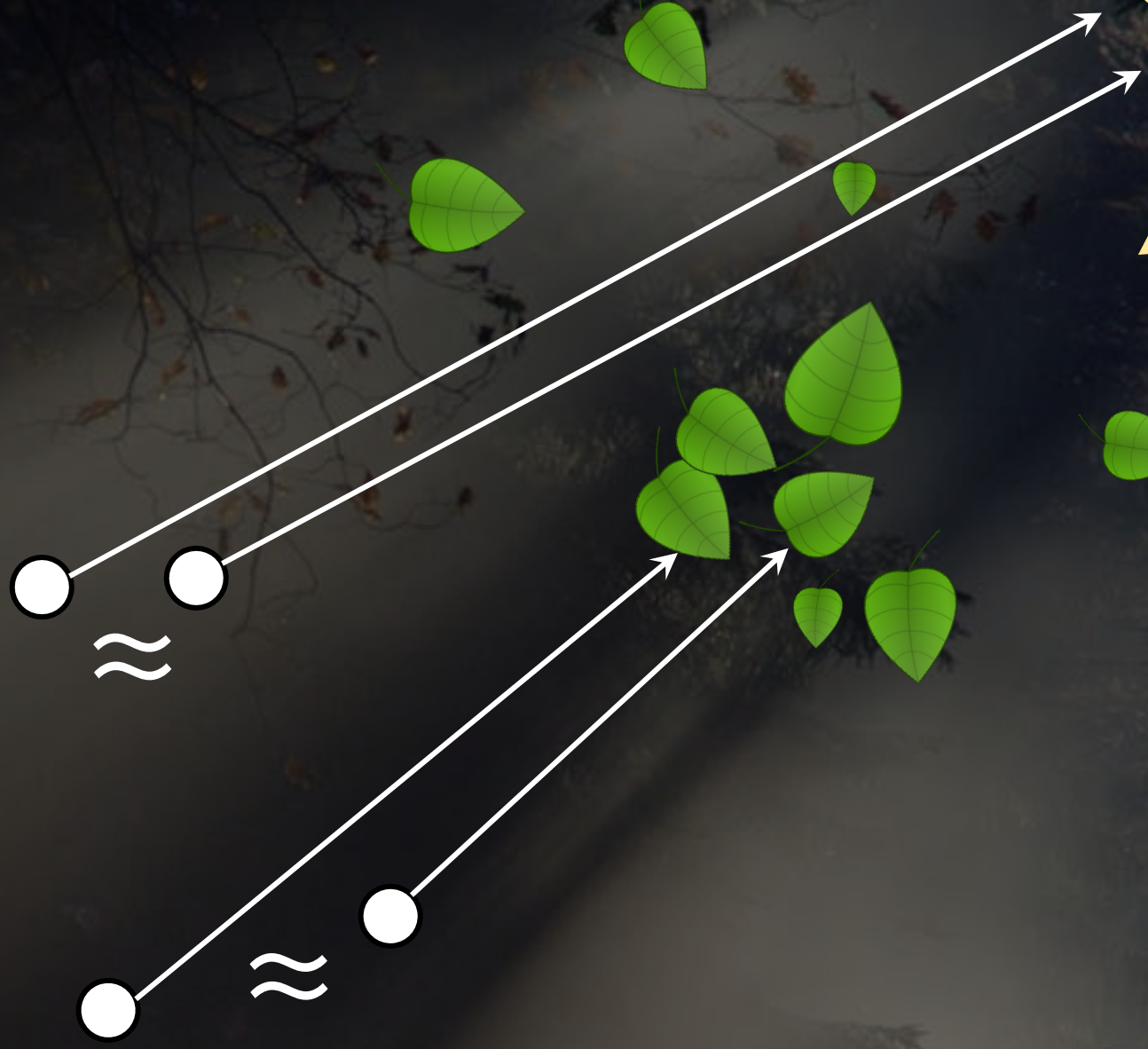
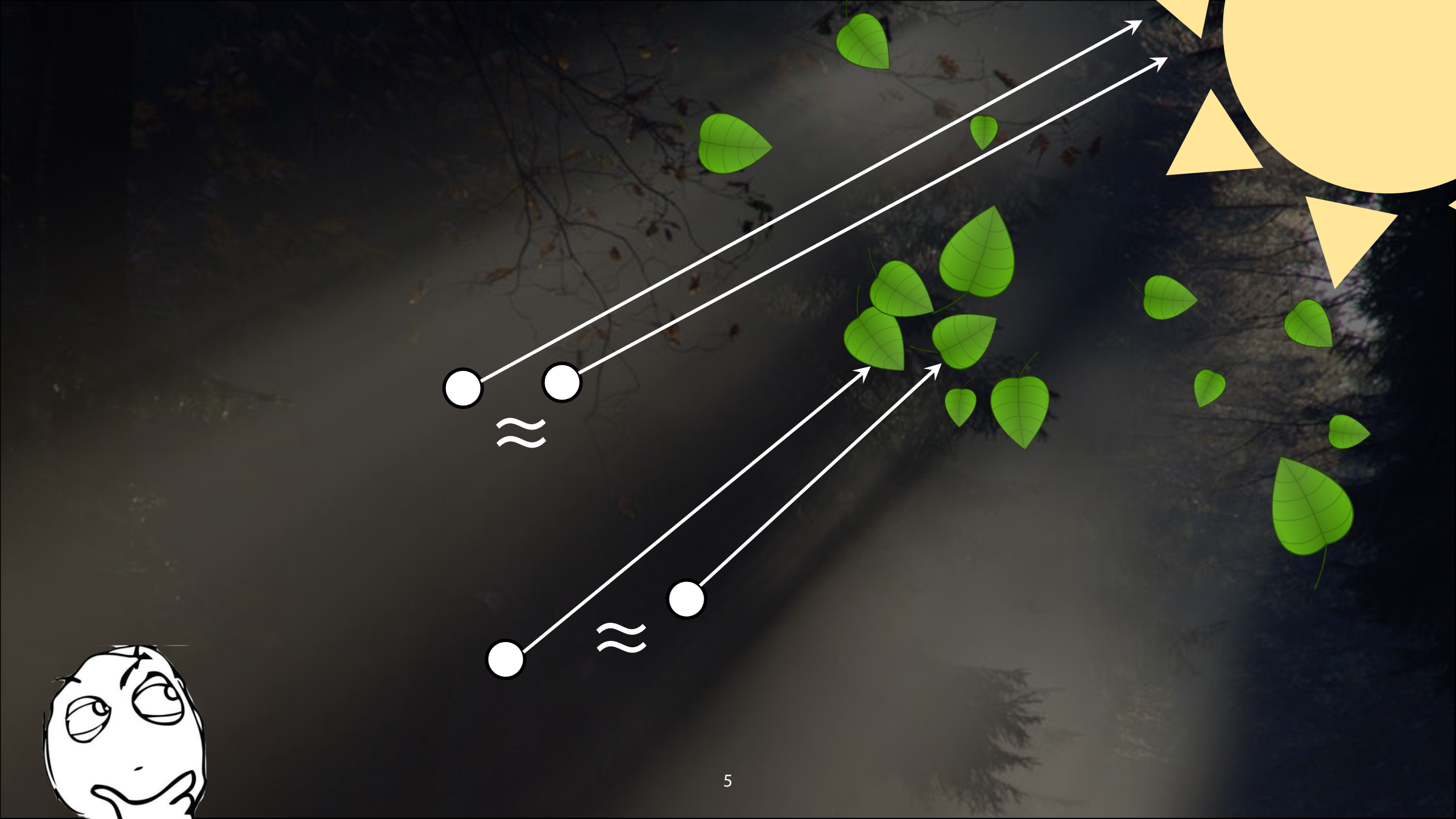
² Dartmouth College





?



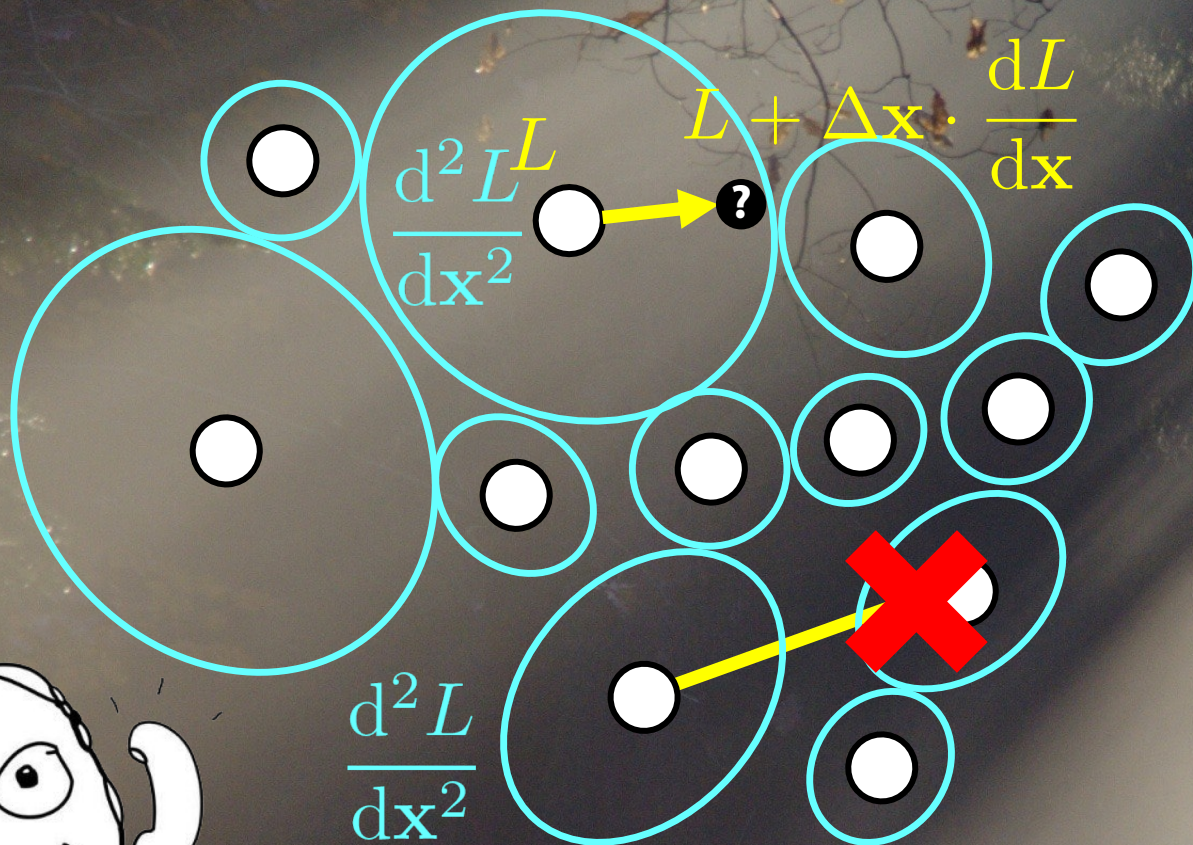




[Schwarzhaupt et al. 2012]



OUR METHOD



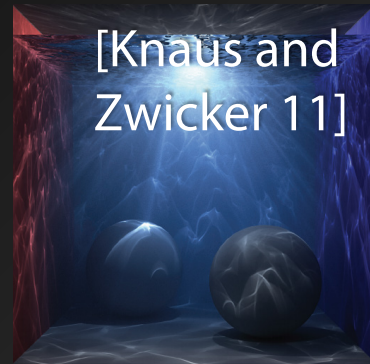


Related Work

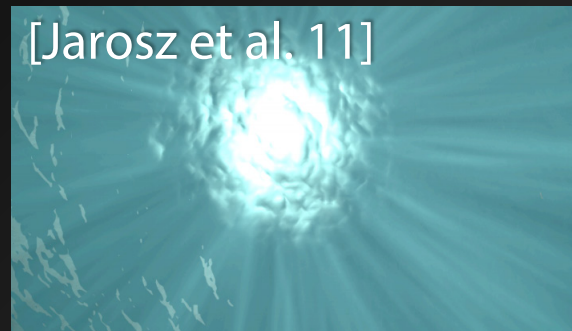
- Photon-based methods



POINTS



Higher-dimensional



BEAMS



POINTS + BEAMS + PATHS



Related Work

Frequency and gradient-domain

- Frequency and first-order analysis
[Durand et al. 2005, Ramamoorthi et al. 2007]
- Media frequency analysis
[Belcour et al. 2014]
- Image-space gradients for MLT
[Lehtinen et al. 2013, Manzi et al. 2014]
- Gradient-domain path tracing methods
[Kettunen et al. 2015, Manzi et al. 2015]



Related Work

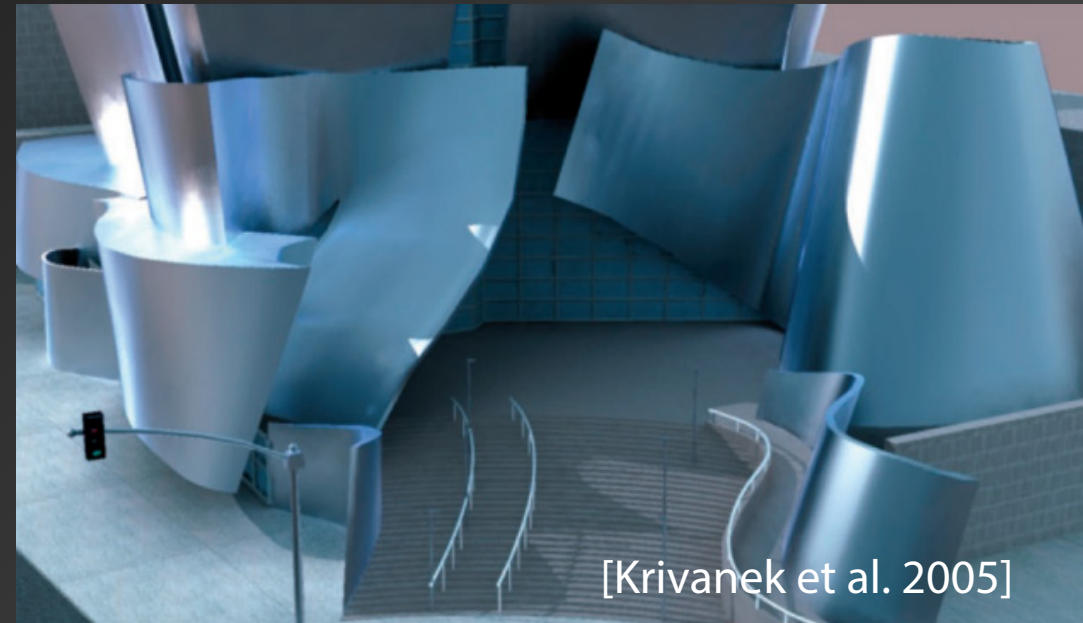
- Irradiance/radiance caching methods





Related Work

- Irradiance/radiance caching methods





Related Work

- Irradiance/radiance caching methods

[Jarosz et al. 2008]

- Media gradients → **YES**
- Occlusions → **NO**
- Higher-order → **NO**





Related Work

[Jarosz et al. 2008]





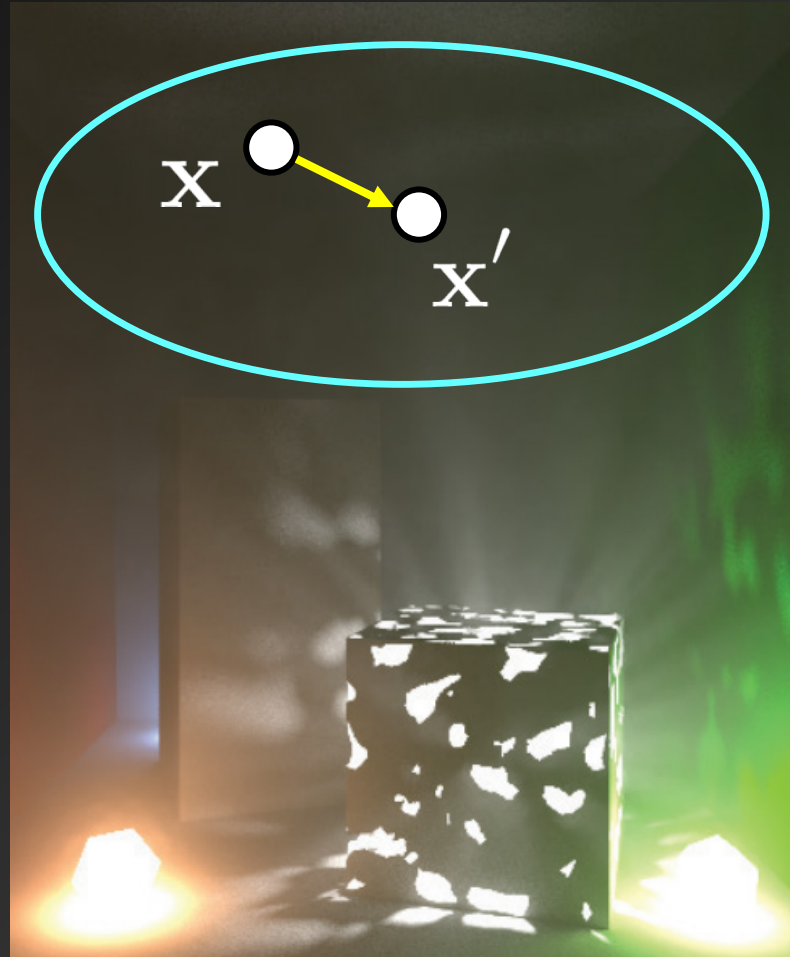
Related Work



OURS



Radiance extrapolation

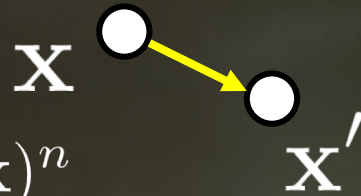




Radiance extrapolation

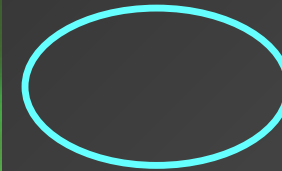
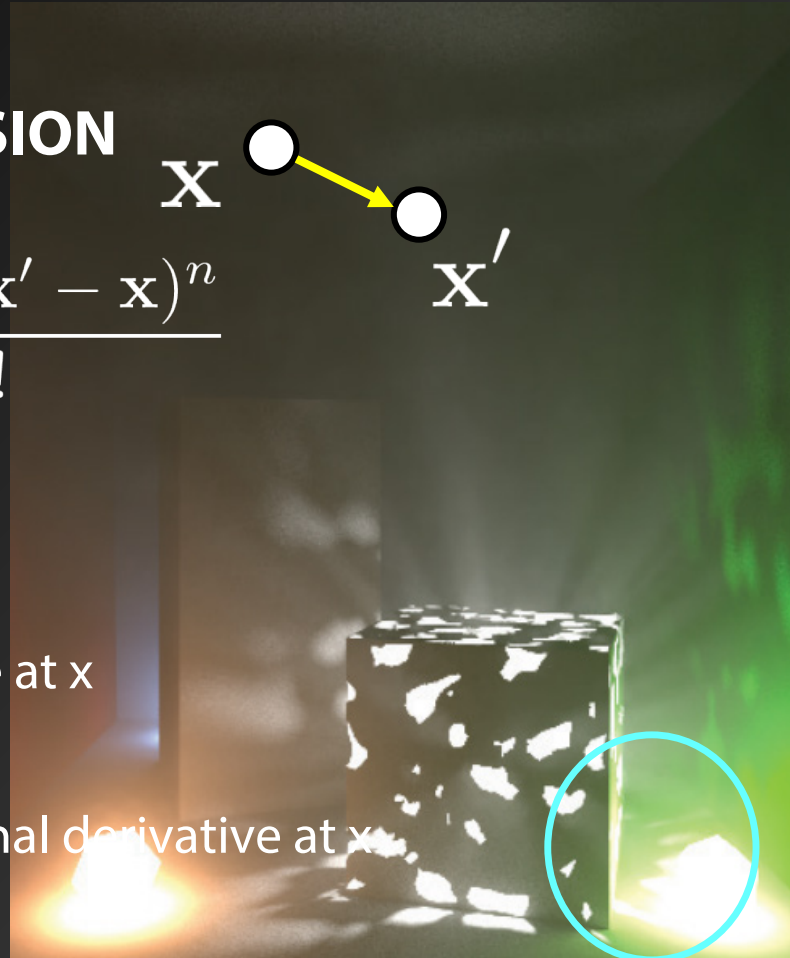
TAYLOR EXPANSION

$$L(\mathbf{x}') = \sum_{n=0}^{\infty} \frac{L^{(n)}(\mathbf{x})(\mathbf{x}' - \mathbf{x})^n}{n!}$$



$L^{(0)}(\mathbf{x})$ Radiance value at \mathbf{x}

$L^{(n)}(\mathbf{x})$ n-th translational derivative at \mathbf{x}

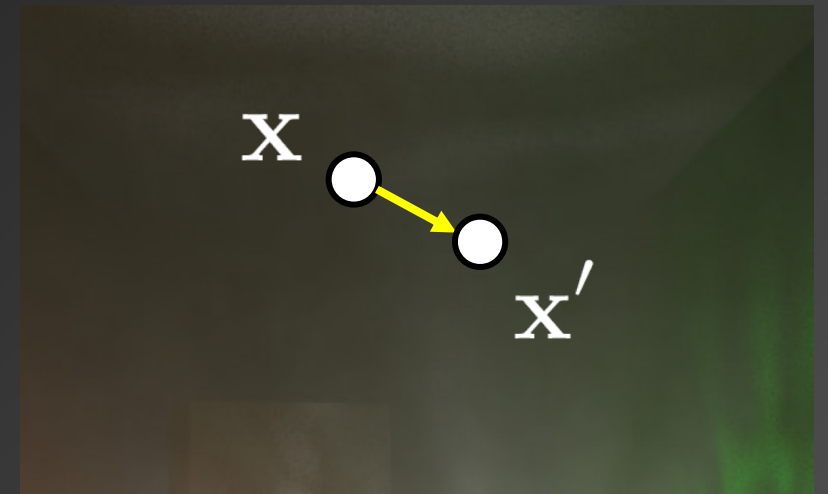




Radiance extrapolation

TRUNCATED TAYLOR EXPANSION

$$\underbrace{L(\mathbf{x}')}_{\text{Actual value}} = \underbrace{L(\mathbf{x}) + \nabla L(\mathbf{x})(\mathbf{x}' - \mathbf{x})}_{\text{Extrapolated value at } \mathbf{x}' \text{ (first order)}} + \underbrace{\mathcal{M}_1}_{\text{Error}}$$



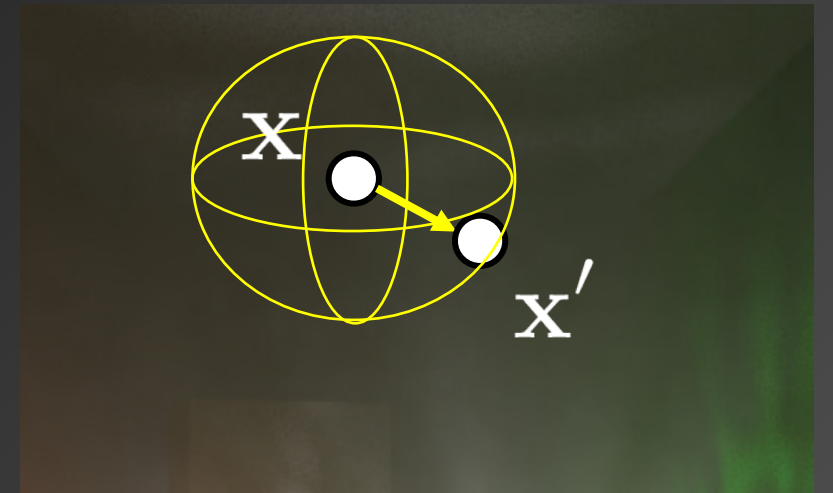
$$\mathcal{M}_1 = \sum_{n=2}^{\infty} \frac{L^{(n)}(x)}{n!} (\mathbf{x}' - \mathbf{x})^n$$



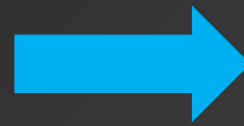
Radiance extrapolation

TRUNCATED TAYLOR EXPANSION

$$\underbrace{L(\mathbf{x}')}_{\text{Actual value}} = \underbrace{L(\mathbf{x}) + \nabla L(\mathbf{x})(\mathbf{x}' - \mathbf{x})}_{\text{Extrapolated value at } \mathbf{x}' \text{ (first order)}} + \underbrace{\mathcal{M}_1}_{\text{Error}}$$



$$\mathcal{M}_1 = \sum_{n=2}^{\infty} \frac{L^{(n)}(x)}{n!} (\mathbf{x}' - \mathbf{x})^n$$



$$\mathcal{M}_1 \approx \frac{\mathbf{H}L(\mathbf{x})}{2} (\mathbf{x}' - \mathbf{x})^2$$



Radiance extrapolation

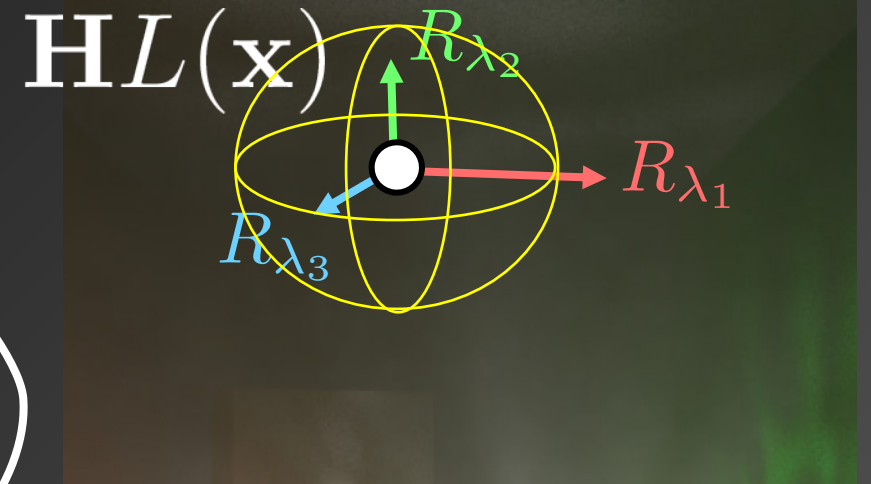
↓
 $\{\lambda_1, \lambda_2, \lambda_3\}$

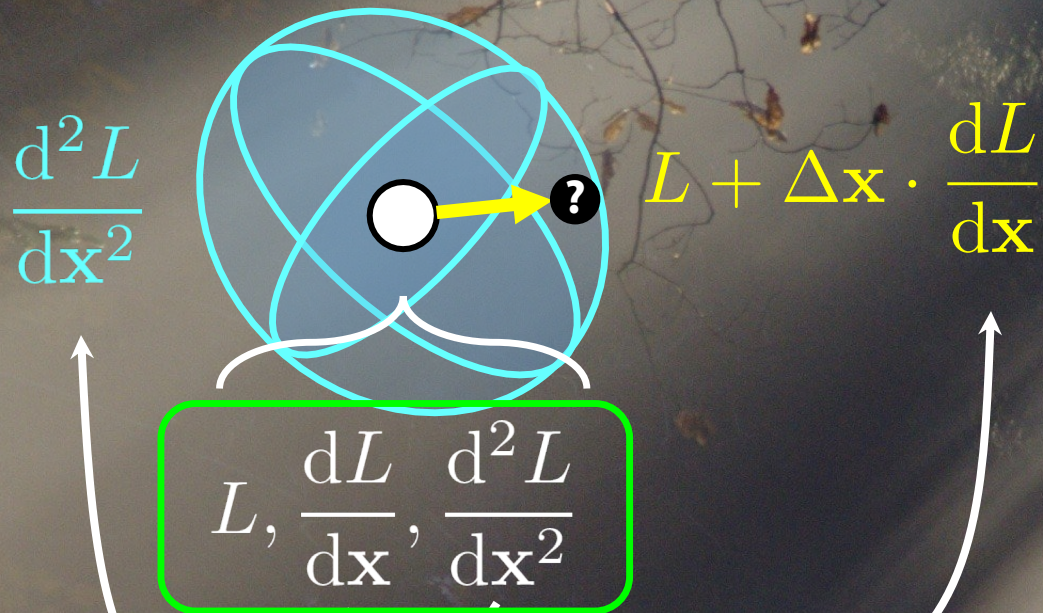
Eigenvalues
of Hessian



$$R_{\lambda_i} = \sqrt[5]{\frac{15L(\mathbf{x})\varepsilon}{4\pi|\lambda_i|}}$$

Error threshold

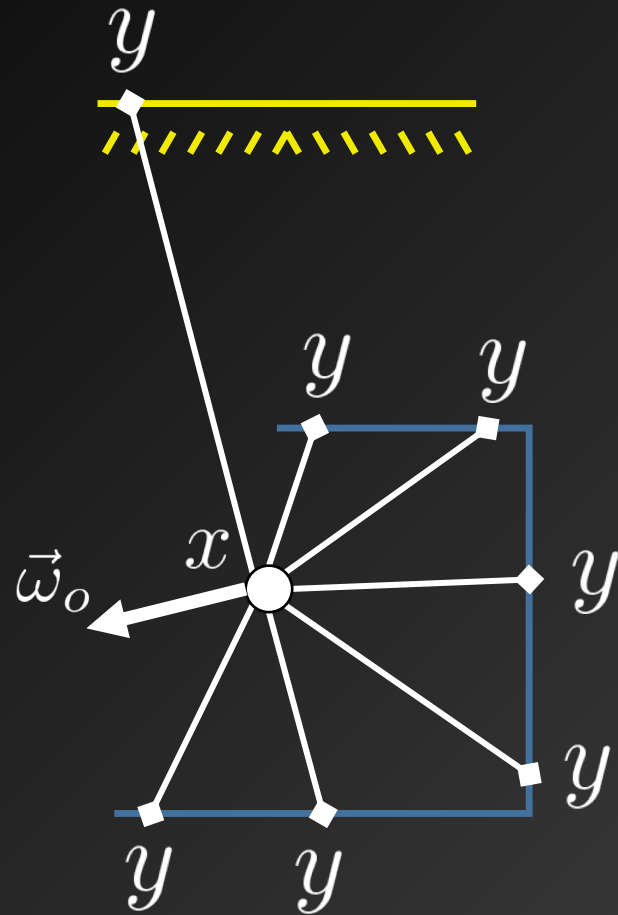




How to compute them?



Derivative computation



$$L(x) = \int_S f(T_r) G V L(y) dy$$

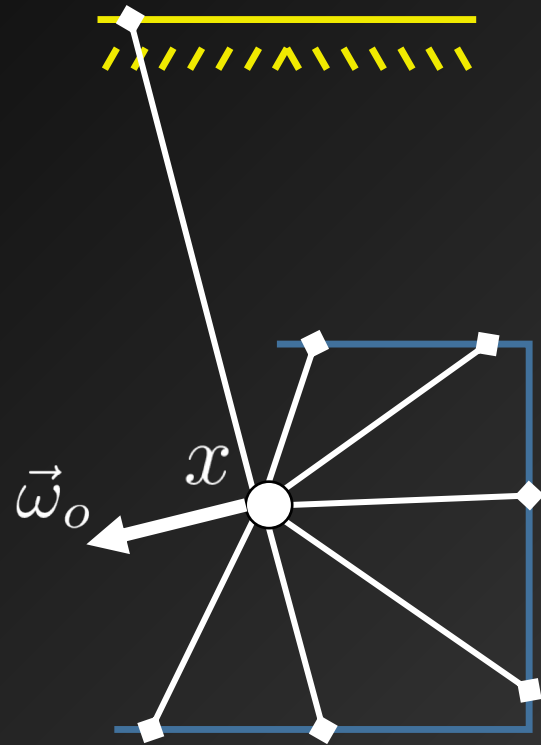
↑ ↑ ↑ ↑ ↑

TRANSMISSION FUNCTION SCATTERING COEFFICIENT TRANSMITTANCE TRANSMITTANCE INCOMING RADIANCE



Derivative computation

[Jarosz et al. 2008]



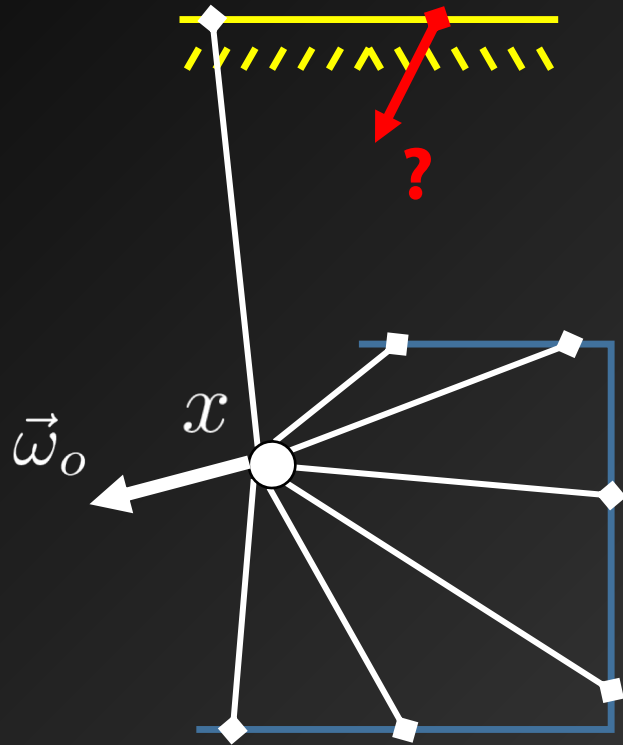
$$L(x) = \int_S f T_r G \mathbf{V} L(y) dy$$

Ignored in gradient computation



Derivative computation

[Jarosz et al. 2008]



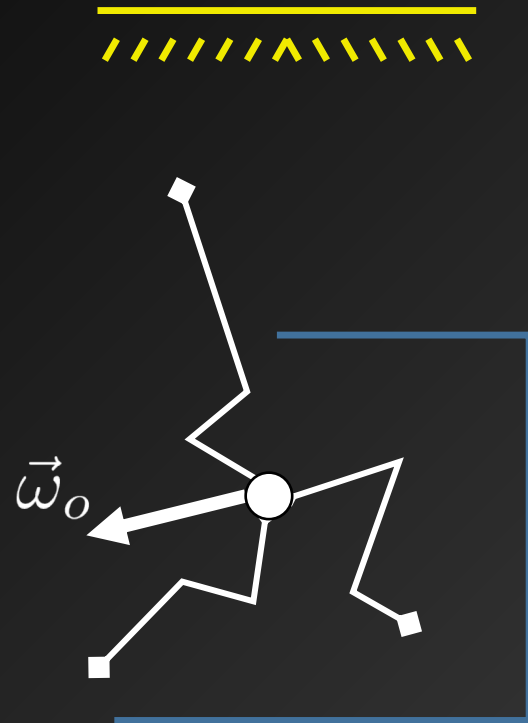
$$L(x) = \int_S f T_r G \mathbf{V} L(y) dy$$

Ignored in gradient
computation



Derivative computation

[Jarosz et al. 2008]



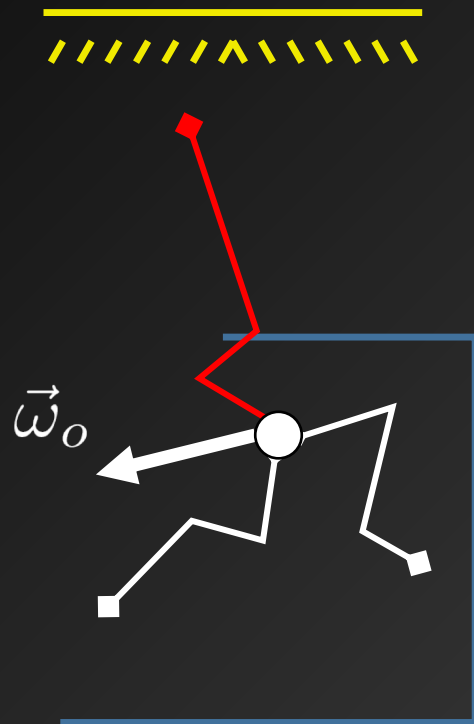
$$L(x) = \int_S f T_r G \mathbf{V} L(y) dy$$

Ignored in gradient
computation



Derivative computation

[Jarosz et al. 2008]



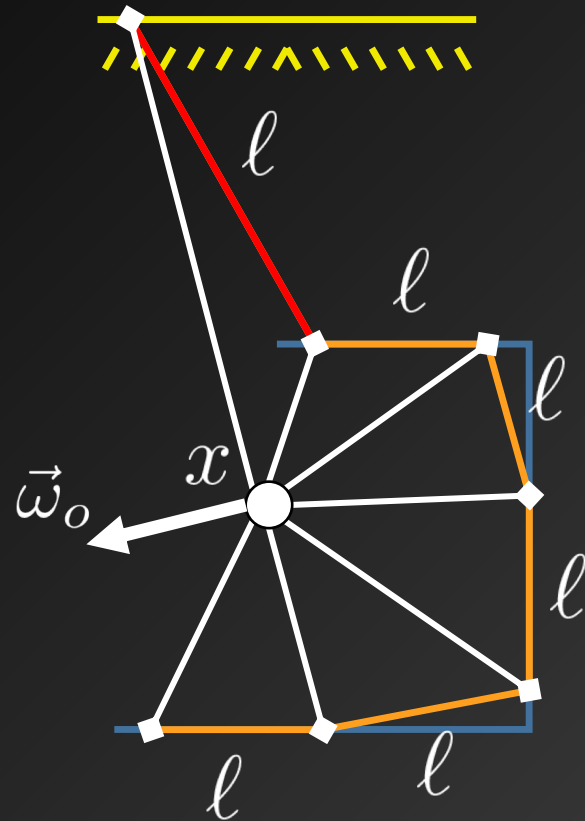
$$L(x) = \int_S f T_r G \mathbf{V} L(y) dy$$

Ignored in gradient computation



Derivative computation

Our method



$$L(x) = \int_S f T_r G V L(y) dy$$



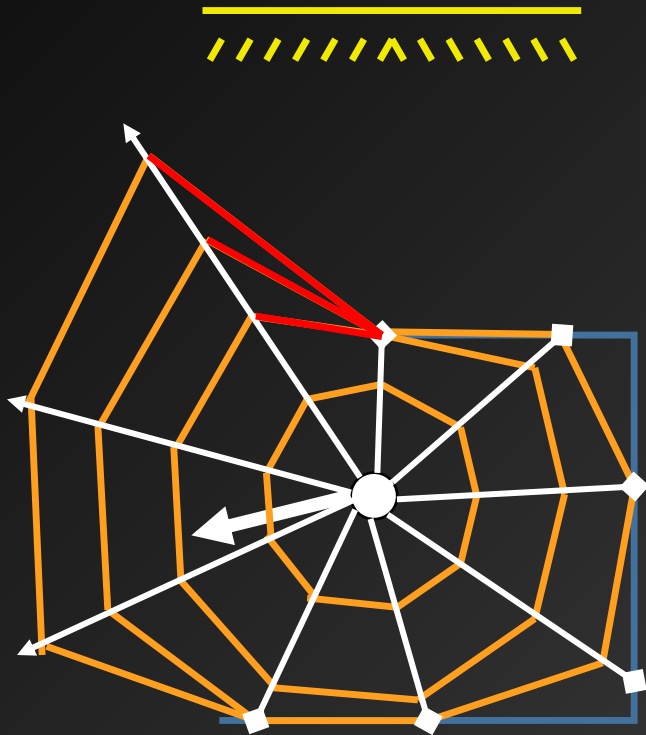
$$L(x) = \sum_{\ell_j} \int_{\ell_j} f T_r G \cancel{V} L(y) dy$$

Gone!



Derivative computation

Our method



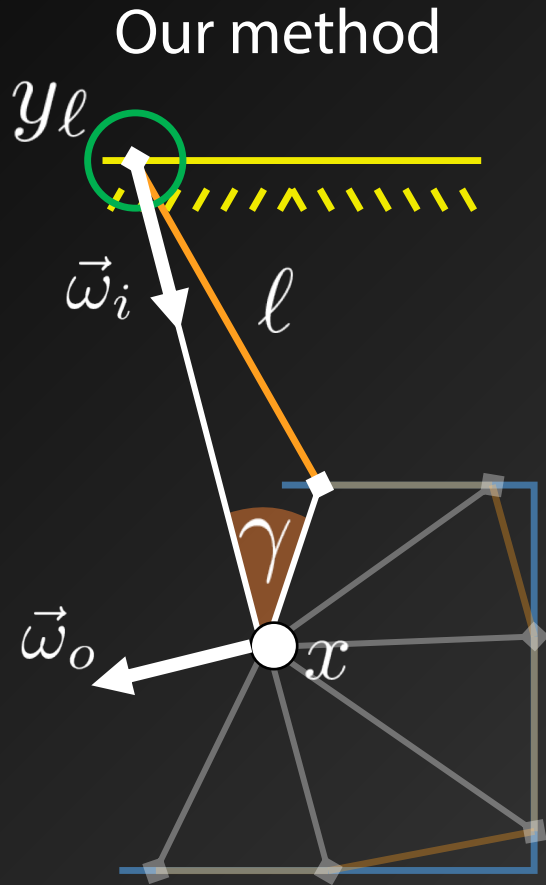
$$L(x) = \int_M f T_r G V L(y) dy$$

$$L(x) = \sum_{r_i} \sum_{\ell_j} \frac{\int_{\ell_j} f T_r G L(y) dy}{\text{pdf}(r_i)}$$

RAY MARCHING



Derivative computation



For a single triangle

$$L(x) = \int_{\ell} f \underbrace{T_r(x, y)}_{\text{Triangle-to-medium form factor}}(dy)$$

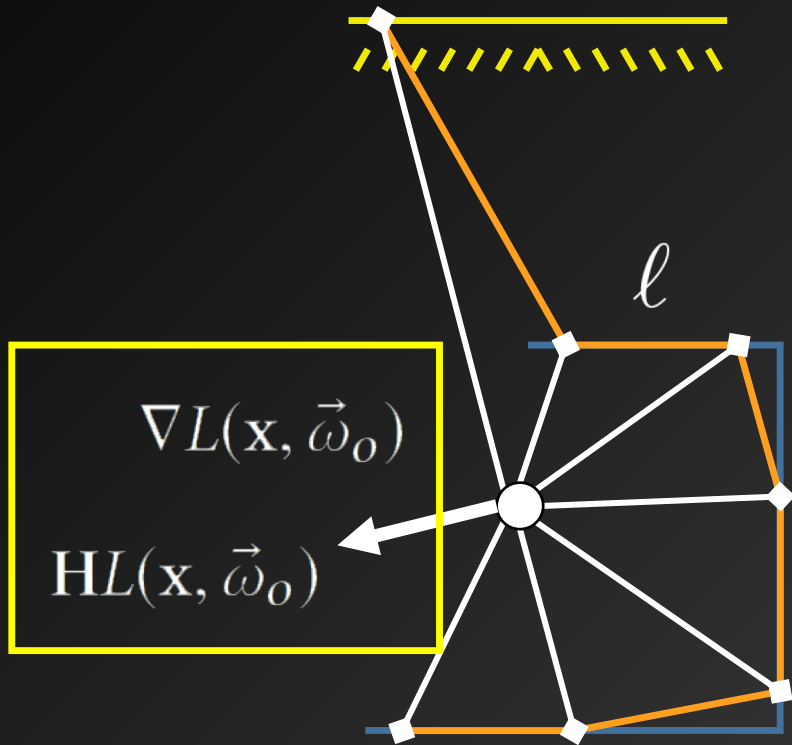
Triangle-to-medium
form factor

$$\gamma \rightarrow 0 \begin{cases} f(x, \omega_i, \omega_o) \rightarrow f_{\ell}(x, \omega_o) \\ \text{Constant } T_r(x, y) \text{ and } L(y) \end{cases}$$



Derivative computation

Our method



$$L(x) = f_{\ell} T_r L(y_{\ell}) F_{\ell}(x)$$

$$\nabla L(x) = \sum_{r_i} \sum_{\ell_j} \frac{\nabla L_{\ell_j}(x)}{\text{pdf}(r_i)}$$

$$HL(x) = \sum_{r_i} \sum_{\ell_j} \frac{HL_{\ell_j}(x)}{\text{pdf}(r_i)}$$



Derivative computation

merged together the phase function $f_s(\mathbf{x}, \hat{\omega}_i, \hat{\omega}_0)$ and scattering coefficient $\mu_s(\hat{\omega})$ as a directional scattering function $f(\mathbf{x}, \hat{\omega}_i, \hat{\omega}_0) = \mu_s(\hat{\omega})f_s(\mathbf{x}, \hat{\omega}_i, \hat{\omega}_0)$, to make the following derivations simpler.

$$\nabla L(\mathbf{x}, \hat{\omega}_0) \approx \sum_{r \in R} \sum_{\ell \in L_r} \frac{\nabla L_f(\mathbf{x}, \hat{\omega}_0)}{\text{pdf}(r_\ell)}, \quad (8)$$

$$\text{HL}(\mathbf{x}, \hat{\omega}_0) \approx \sum_{r \in R} \sum_{\ell \in L_r} \frac{\text{HL}_f(\mathbf{x}, \hat{\omega}_0)}{\text{pdf}(r_\ell)}. \quad (9)$$

which in turn require differentiating the radiance from each segment.

Unfortunately, we cannot compute Equation (7) and its derivatives analytically in closed-form, while computing it numerically would be prohibitively expensive. We instead introduce a set of assumptions to build a closed-form approximation:

- For a sufficiently fine subdivision the angle γ tends to 0, so $\hat{\omega}_i$ can be regarded as constant for the whole segment, and $f(\mathbf{x}, \hat{\omega}_i, \hat{\omega}_0) \approx f(\mathbf{x}, \hat{\omega}_r, \hat{\omega}_0)$, with $\hat{\omega}_r$ a fixed direction from \mathbf{x} to a point in segment ℓ .
- For all $\gamma \in \Gamma$, we assume constant $T_r(\mathbf{x}, \mathbf{y}) = T_r(\mathbf{x}, \gamma_r)$, and $L(\mathbf{y}, \hat{\omega}_i) = L(\gamma_r, \hat{\omega}_i)$. Following existing approaches for surface irradiance, we choose γ_r as the furthest point in the segment ℓ , which will be not occluded/unoccluded.

These assumptions allow us to significantly simplify the integral in Equation (7) to:

$$L_f(\mathbf{x}, \hat{\omega}_0) \approx f(\mathbf{x}, \hat{\omega}_r, \hat{\omega}_0) T_r(\mathbf{x}, \gamma_r) L(\gamma_r, \hat{\omega}_i) \int_{\ell} G(\mathbf{x}, \mathbf{y}) d\mathbf{y} \\ = f(\mathbf{x}, \hat{\omega}_r, \hat{\omega}_0) T_r(\mathbf{x}, \gamma_r) L(\gamma_r, \hat{\omega}_i) F_{\ell}(\mathbf{x}). \quad (10)$$

which now admits a closed-form solution in both 2D and 3D (see Appendices B and C). More importantly, this allows us to approximate the derivatives of L_f in closed form as:

$$\nabla L_f = L_f \nabla f + \nabla L_f f, \quad (11)$$

$$\text{HL}_f = L_f \mathbf{H}f + \nabla L_f \nabla^T f + \nabla f \nabla^T L_f + \text{HL}_f f, \quad (12)$$

where

$$\nabla L_f = L_f \nabla F_r + \nabla L_r F_r, \quad (13)$$

$$\text{HL}_f = L_r \mathbf{H}F_r + \nabla L_r \nabla^T F_r + \nabla F_r \nabla^T L_r + \text{HL}_r F_r, \quad (14)$$

$$\nabla L_r = L \nabla T_r + \nabla L T_r, \quad (15)$$

$$\text{HL}_r = L \nabla H_r + \nabla L \nabla^T T_r + \nabla T_r \nabla^T L_r + \text{HL}_r T_r. \quad (16)$$

For brevity we have omitted function parameters, and we express gradients and Hessians in terms of the scaled radiance $L_r = F_r L_r$, and the reduced radiance $L_r = L T_r$. While Equations (10–16) are general, we restrict our work to Lambertian surfaces and isotropic, homogeneous media (in Section 7 we discuss how to extend it to anisotropic and heterogeneous media). This means that both L and f are constant, and therefore their derivatives cancel out as $\nabla L = \text{HL} = \nabla f = \mathbf{H}f = 0$, removing directional dependences; this allows us to simplify Equations (11) and (12) to:

$$\nabla L_f \approx L_f (T_r \nabla F_r + \nabla T_r F_r), \quad (17)$$

$$\text{HL}_f \approx L_f (T_r \mathbf{H}F_r + \nabla T_r \nabla^T F_r + \nabla F_r \nabla^T T_r + \text{HL}_r F_r). \quad (18)$$

We refer to Appendices A, B and C for all the terms.

By construction, our formulation in Equation (6) and its derivatives (Equations (8) and (9)) are biased but consistent estimators of $L(\mathbf{x}, \hat{\omega}_0)$, $\nabla L(\mathbf{x}, \hat{\omega}_0)$, and $\text{HL}(\mathbf{x}, \hat{\omega}_0)$ respectively. In addition the assumptions imposed in Equation (10) introduce some additional bias due to the piecewise assumption in the scattering f , transmittance T_r , and radiance terms L . However, as shown in Figure 3 our formulation converges accurately to the actual derivatives. Note that we use this biased but consistent approximation only to compute first- and second-order derivatives of media radiance (Equations (8) and (9)), while computing actual radiance values (Equation (1)) using the standard unbiased Monte Carlo estimator. In the following, we describe how to use the derivatives in Equations (8) and (9) for interpolating radiance from a set of cache points, and define an error metric for such interpolation.

5 SECOND-ORDER ERROR CONTROL FOR MEDIA RADIANCE EXTRAPOLATION

The error in radiance caching is controlled by a tolerance value ϵ , and depends both on how radiance is extrapolated, and on the radiance moments at where a point \mathbf{x}^* is being used for extrapolation. We provide here the key ideas and resulting equations for the valid regions in the context of 2D and 3D participating media and provide detailed derivations in the supplementary material.

Existing work on radiance caching for participating media estimates the relative error using radiance gradients at \mathbf{x} . However, ignoring higher-order derivatives creates suboptimal cache distributions that often oversample regions near surfaces and light sources. Given the radiance and the first n derivatives at a media point \mathbf{x} , we can approximate radiance at point $\mathbf{x}^* \in \mathcal{N}$ using an n^{th} -order Taylor expansion. Following previous work [Schwarzaupt et al. 2012] we truncate to order one, approximating $L(\mathbf{x}^*, \hat{\omega}_0)$ as:

$$L(\mathbf{x}^*, \hat{\omega}_0) \approx L(\mathbf{x}, \hat{\omega}_0) + \nabla L(\mathbf{x}, \hat{\omega}_0) \Delta \mathbf{x}. \quad (19)$$

Since we focus on isotropic media, we remove the directional dependence in the following derivations to simplify notation. By using a second order expansion of $L(\mathbf{x})$ as our oracle, we can approximate the relative error $e^l(\mathbf{x}^*)$ of the extrapolation as:

$$e^l(\mathbf{x}^*) \approx \frac{|\Delta \mathbf{x}^T \text{HL}(\mathbf{x}) \Delta \mathbf{x}|}{2L(\mathbf{x})}. \quad (20)$$

with $\text{HL}(\mathbf{x})$ the Hessian matrix of $L(\mathbf{x})$. This expression is similar to the second-order error metric proposed by Jarosz et al. [2012] and follow-up work by Schwarzaupt et al. [2012], although these works dealt with surfaces only.

By integrating Equation (20) in the neighborhood of \mathbf{x} for a given error threshold ϵ , we can express the valid region in two-dimensional media as an ellipse with principal radii $R_{2D}^{\lambda_i}$ (see Equations (S.9)–(S.12) in the supplemental for the complete derivation):

$$R_{2D}^{\lambda_i} = \sqrt{\frac{2L(\mathbf{x})\epsilon}{\lambda_i |\mathbf{H}|}}. \quad (21)$$

where λ_i is the i -th eigenvalue of the radiance Hessian $\text{HL}(\mathbf{x})$. This formula is analogous to the relative error metric presented by Schwarzaupt and colleagues [2012] for surfaces, but here the

Jaakko Lehtinen, Tero Karras, Samuli Laine, Mika Aittala, Fredo Durand, and Timo Aila. 2013. Gradient-Domain Metropolis Light Transport. *ACM Trans. Graph.* 32, 4 (2013).

Marco Masini, Markus Kettunen, Mika Aittala, Jaakko Lehtinen, Fredo Durand, and Matthias Zwicker. 2015. Gradient-Domain Bidirectional Path Tracing. In *Proc. of EGSR*.

Marco Masini, Fabrice Rousselet, Markus Kettunen, Jaakko Lehtinen, and Matthias Zwicker. 2014. Improved Sampling for Gradient-Domain Metropolis Light Transport. *ACM Trans. Graph.* 33, 4, Article 178 (2014).

Suham Uday Mehra, Brandon Wang, Ravi Ramamoorthi, and Fredo Durand. 2013. Axis-aligned Filtering for Interactive Physically-based Diffuse Indirect Lighting. *ACM Trans. Graph.* 32, 4, Article 96 (July 2013), 12 pages. <https://doi.org/10.1145/2641912.2641947>

Adriello Milazzo. 2014. Higher Order Ray Marching. *Computer Graphics Forum* 33, 8 (2014), 1653–1716. <https://doi.org/10.1111/cgf.12424>

Rachel Orel, Stephane Riviere, Fredo Durand, and Claude Puech. 1996. Radiosity for dynamic scenes in flatland with the visibility complex. In *Computer Graphics Forum*, Vol. 15, 237–248.

Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. 2007. A First-order Analysis of Lighting, Shading and Shadows. *ACM Trans. Graph.* 26, 1, Article 2 (Jun. 2007). <https://doi.org/10.1145/1189762.1189764>

Mickaël Ribardière, Samuel Corré, and Karl Bouatouch. 2011. Adaptive records for volume radiance caching. *The Visual Computer* 27, 4 (2011), 455–464.

Fabrice Rousselet, Wojciech Jarosz, and Jan Novák. 2016. Image-space Control Variables for Rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35, 6 (December 2016), 1601–1612. <https://doi.org/10.1145/2969179.2962443>

Lars Schödl, Jeppe Revall Frisvad, Kenan Erdem, and Jon Sporring. 2007. Photon Differentials. In *Proceedings of the 9th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia (GRAPEE '07)*. ACM, New York, USA, 179–186. <https://doi.org/10.1145/1321261.1321293>

Jorge Schwarzhaupt, Henrik Wann Jensen, and Wojciech Jarosz. 2012. Practical Hessian-Based Error Control for Irradiance Caching. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 31, 6 (Nov. 2012). <https://doi.org/10.1145/216145.2166212>

Frank Sottos and Yves D. Williamson. 2004. Path differentials and applications. In *Rendering Techniques 2004 (Proceedings of the Eurographics Workshop on Rendering)*. Eurographics Association, London, United Kingdom, 257–268. https://doi.org/10.1007/978-3-7091-6242-2_24

Adriaan Van Oosterom and Jan Strackee. 1983. The Solid Angle of a Plane Triangle. *IEEE Transactions on Biomedical Engineering* BME-30, 2 (1983), 125–126. <https://doi.org/10.1109/1078.1983.325207>

Ingo Wald, Sven Woop, Carsten Beathia, Gregory S Johnson, and Manfred Eder. 2014. Embree: a kernel framework for efficient CPU ray tracing. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 143.

Gregory J. Ward and Paul S. Heckbert. 1992. Irradiance Gradients. *1992 Eurographics Workshop on Rendering*, 85–98. <https://scholarship.cs.cmu.edu/papers/92/wpaper.html>

Gregory J. Ward, Francis M. Rubinstein, and Robert D. Christ. 1988. A Ray Tracing Solution for Diffuse Interreflection. In *Proceedings of the 21th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '88)*. ACM, New York, NY, USA, 85–92. <https://doi.org/10.1145/54812.378490>

Ling-Qi Yan, Mithal Hashim, Wenzel Jakob, Jason Lawrence, Steve Marschner, and Ravi Ramamoorthi. 2014. Rendering Clints on High-Resolution Normal-Mapped Specular Surfaces. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014).

Ling-Qi Yan, Mithal Hashim, Steve Marschner, and Ravi Ramamoorthi. 2016. Position-Normal Distributions for Efficient Rendering of Specular Microstructure. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35, 4 (2016).

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bohang Moon, Ravi Ramamoorthi, Fabrice Rousselet, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Computer Graphics Forum (Proceedings of Eurographics)* 34, 2 (May 2015), 667–681. <https://doi.org/10.1111/cgf.12592>

APPENDICES

In the following we summarize 2D and 3D expressions of translational derivatives of transmittance and form factors needed for our method. We box all relevant final expressions that to the best of our knowledge are new to the literature. We define column vectors as \mathbf{v} and row vectors as $\bar{\mathbf{v}}$. Expressions such as $\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_2$ denote dot (inner) products, while expressions such as $\bar{\mathbf{r}}_1 \bar{\mathbf{r}}_2^T$, $\bar{\mathbf{v}} \cdot (\dots)^T \bar{\mathbf{v}}$, and $(\dots) \cdot (\dots)^T$ denote outer products.

A HOMOGENEOUS TRANSMITTANCE DERIVATIVES

Homogeneous transmittance is modeled by the exponential decay due to extinction,

$$T_r = e^{-\mu_i \|\bar{\mathbf{y}}\|} \quad (24)$$

where $\|\bar{\mathbf{y}}\|$ denotes distance between source \mathbf{y} and shaded point \mathbf{x} . Its gradient and Hessian with respect to a translation of \mathbf{x} are

$$\nabla T_r = -\mu_i \frac{\bar{\mathbf{y}}}{r}, \quad (25)$$

$$\mathbf{H}T_r = -\mu_i \left(\frac{J(\bar{\mathbf{y}})}{r} - \frac{1}{r^2} \bar{\mathbf{y}} \bar{\mathbf{y}}^T - \frac{\mu_i}{r^2} \bar{\mathbf{y}} \bar{\mathbf{y}}^T \bar{\mathbf{y}} \right) T_r. \quad (26)$$

B 2D SEGMENT-MEDIA FORM FACTOR DERIVATIVES

The form factor between a 2D segment ℓ and a media point \mathbf{x} (Figure 15, left) is defined as the integrated curve-media geometry term along all segment points. This is equivalent to the angular ratio covered by ℓ as seen from \mathbf{x}

$$F_{\ell}(\mathbf{x}) = \frac{1}{2\pi} \int_{y \in \ell} \frac{\cos \theta'_y}{\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\|^2} d\ell(\mathbf{y}) = \frac{1}{2\pi} \arccos \left(\frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_y}{r_1} \right).$$

where $r_1 = \|\bar{\mathbf{x}}\|$. The form factor gradient and Hessian become

$$\nabla F_{\ell}(\mathbf{x}) = \frac{1}{2\pi} \frac{\nabla \cos \theta'}{\sqrt{1 - \cos^2 \theta'}} \quad (27)$$

$$\mathbf{H}F_{\ell}(\mathbf{x}) = -\frac{1}{2\pi} \left(\frac{J(\nabla \cos \theta')}{\sqrt{1 - \cos^2 \theta'}} + \frac{\cos \theta'}{(1 - \cos^2 \theta')^{3/2}} \nabla \cos \theta' \nabla^T \cos \theta' \right) \quad (28)$$

where J is the Jacobian operator, and

$$\nabla \cos \theta' = \frac{\cos \theta'}{r_0} \bar{\mathbf{x}} \bar{\mathbf{y}} + \frac{\cos \theta'}{r_1} \bar{\mathbf{x}} \bar{\mathbf{y}}_1 \quad (29)$$

$$= \frac{1}{r_0 r_1} (\bar{\mathbf{x}} \bar{\mathbf{y}}_0 + \bar{\mathbf{x}} \bar{\mathbf{y}}_1) \quad (30)$$

$$J(\nabla \cos \theta') = -J \left(\frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_0}{r_0 r_1} \right) - J \left(\frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_1}{r_0 r_1} \right) \\ + J \left(\frac{\cos \theta'}{r_0} \bar{\mathbf{x}} \bar{\mathbf{y}}_0 \right) + J \left(\frac{\cos \theta'}{r_1} \bar{\mathbf{x}} \bar{\mathbf{y}}_1 \right). \quad (31)$$

$$J \left(\frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_0}{r_0 r_1} \right) = \frac{J(\bar{\mathbf{x}} \bar{\mathbf{y}}_0)}{r_0 r_1} + \frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_0 \bar{\mathbf{x}} \bar{\mathbf{y}}_0^T}{r_0^2 r_1} + \frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_0 \bar{\mathbf{x}} \bar{\mathbf{y}}_1^T}{r_0 r_1^2} \\ + \frac{2 \cos \theta'}{r_1} \bar{\mathbf{x}} \bar{\mathbf{y}}_0 \bar{\mathbf{x}} \bar{\mathbf{y}}_1^T. \quad (32)$$

$$J \left(\frac{\cos \theta'}{r_1} \bar{\mathbf{x}} \bar{\mathbf{y}}_1 \right) = \frac{\cos \theta'}{r_1} J(\bar{\mathbf{x}} \bar{\mathbf{y}}_1) + \frac{\bar{\mathbf{x}} \bar{\mathbf{y}}_1 \bar{\mathbf{x}} \bar{\mathbf{y}}_1^T}{r_1^2} \\ + 2 \cos \theta' \bar{\mathbf{x}} \bar{\mathbf{y}}_1 \bar{\mathbf{x}} \bar{\mathbf{y}}_1^T. \quad (33)$$

C 3D TRIANGLE-MEDIA FORM FACTOR DERIVATIVES

The form factor between a 3D triangular face Δ , and a media point \mathbf{x} (see Figure 15, right) is defined as the integrated surface-media geometry term along all points in the triangle. Analogous to 2D,

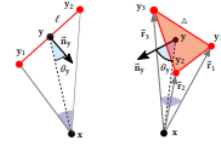


Fig. 15. Setup for segment-to-media (2D, left) and triangle-to-media (3D, right) form factors.

this has analytical solution equal to the ratio of solid angle covered by the triangle as seen from \mathbf{x} ,

$$F_{\Delta}(\mathbf{x}) = \frac{1}{4\pi} \int_{y \in \Delta} \frac{\cos \theta'_y}{\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\|^2} d\Omega(\mathbf{y}) = \frac{\Omega}{4\pi} \quad (34)$$

Solid angle Ω of a triangle can be computed as [Van Oosterom and Strackee 1983]

$$\Omega = 2 \arctan \frac{|A|}{B} \quad (35)$$

with

$$A = \bar{\mathbf{r}}_1 \cdot (\bar{\mathbf{r}}_2 \times \bar{\mathbf{r}}_3) \quad (36)$$

$$B = r_1 r_2 r_3 + (\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_2) r_3 + (\bar{\mathbf{r}}_2 \cdot \bar{\mathbf{r}}_3) r_1 + (\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_3) r_2 \quad (37)$$

where $\bar{\mathbf{r}}_i = \bar{\mathbf{x}} \bar{\mathbf{y}}_i$, and $r_i = \|\bar{\mathbf{r}}_i\|$ (see Figure 15, right). Note that the numerator A requires an absolute value to ensure positive vector order (i.e. triangle winding) with respect to \mathbf{x} . Also, when obtaining negative arctangent values, π must be added to the obtained solid angle.

The gradient of the form factor with respect to a translation of \mathbf{x} becomes

$$\nabla F_{\Delta}(\mathbf{x}) = \frac{1}{2\pi} \nabla \arctan \frac{|A|}{B} \quad (38)$$

$$= \frac{1}{2\pi} \frac{A \nabla |A| - |A| \nabla B}{|A|^2 + B^2}, \quad (39)$$

and its Hessian yields

$$\mathbf{H}F_{\Delta}(\mathbf{x}) = \frac{1}{2\pi} \left(\frac{\nabla(|A|) \nabla^T B - \nabla B \nabla^T (|A|)}{|A|^2 + B^2} + \frac{B \nabla(|A|) - |A| \nabla B}{|A|^2 + B^2} \right. \\ \left. - \frac{\nabla(|A|) \cdot |A| \nabla B}{(|A|^2 + B^2)^2} (\nabla(|A|^2) + \nabla(B^2))^T \right) \quad (40)$$

Note that for computing the terms $\nabla(|A|)$ and $J(\nabla|A|)$, we can apply the derivatives of the absolute value of a vector function:

$$\nabla(|A|) = \frac{A}{|A|} \nabla A, \quad (41)$$

$$J(\nabla|A|) = \frac{A \nabla(|A|) - |A| \nabla A}{|A|} - \frac{A^2 (\nabla A \nabla^T A)}{|A|^3}. \quad (42)$$

The gradient of A becomes

$$\nabla A = J(\bar{\mathbf{r}}_2 \times \bar{\mathbf{r}}_3) \bar{\mathbf{r}}_1 + J(\bar{\mathbf{r}}_1 \times \bar{\mathbf{r}}_3) \bar{\mathbf{r}}_2 + J(\bar{\mathbf{r}}_1 \times \bar{\mathbf{r}}_2) \bar{\mathbf{r}}_3. \quad (43)$$

By the Jacobi identity we have that

$$J(\bar{\mathbf{r}}_2 \times \bar{\mathbf{r}}_3) = \bar{\mathbf{r}}_2 \times J(\bar{\mathbf{r}}_3) - \bar{\mathbf{r}}_3 \times J(\bar{\mathbf{r}}_2) \quad (44)$$

where any vector-matrix cross product $\bar{\mathbf{v}} \times J(\mathbf{e})$ can be expressed by means of the matrix multiplication form

$$\bar{\mathbf{v}} \times J(\mathbf{e}) = (\bar{\mathbf{v}}) J(\mathbf{e}) \quad (45)$$

$$\bar{\mathbf{v}} = \begin{pmatrix} v_{10} \\ v_{20} \\ v_{30} \end{pmatrix}, \quad (\bar{\mathbf{v}}) = \begin{pmatrix} 0 & -v_{10} & v_{20} \\ v_{20} & 0 & -v_{30} \\ -v_{30} & v_{10} & 0 \end{pmatrix} \quad (46)$$

Since $J(\bar{\mathbf{r}}_1) = J(\bar{\mathbf{r}}_2) = J(\bar{\mathbf{r}}_3) = -J$, we have that

$$\nabla A = ((\bar{\mathbf{r}}_2) J(\bar{\mathbf{r}}_3) - (\bar{\mathbf{r}}_3) J(\bar{\mathbf{r}}_2)) \bar{\mathbf{r}}_1 - (\bar{\mathbf{r}}_2 \times \bar{\mathbf{r}}_3) \\ = (\bar{\mathbf{r}}_3 - \bar{\mathbf{r}}_2) \bar{\mathbf{r}}_1 - (\bar{\mathbf{r}}_2 \times \bar{\mathbf{r}}_3). \quad (47)$$

Note that $(\bar{\mathbf{r}}_3 - \bar{\mathbf{r}}_2) = (\mathbf{y}_3 - \mathbf{y}_2)$ and therefore does not depend on \mathbf{x} , and $(\bar{\mathbf{v}})^T = -(\bar{\mathbf{v}})$ (see Equation (46)). As a result, the Jacobian of ∇A becomes a zero matrix

$$J(\nabla A) = J((\bar{\mathbf{r}}_3 - \bar{\mathbf{r}}_2) \bar{\mathbf{r}}_1) - J(\bar{\mathbf{r}}_2 \times \bar{\mathbf{r}}_3) \\ = (\bar{\mathbf{r}}_3 - \bar{\mathbf{r}}_2) - (\bar{\mathbf{r}}_3 - \bar{\mathbf{r}}_2) \\ = 0. \quad (48)$$

The gradient of B becomes

$$\nabla B = \nabla(r_1 r_2 r_3) + \nabla((\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_2) r_3) \\ + \nabla((\bar{\mathbf{r}}_2 \cdot \bar{\mathbf{r}}_3) r_1) + \nabla((\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_3) r_2) \quad (49)$$

where

$$\nabla(r_1 r_2 r_3) = r_2 r_3 \nabla r_1 + r_1 r_3 \nabla r_2 + r_1 r_2 \nabla r_3 \quad (50)$$

$$\nabla((\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_2) r_3) = (\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_2) \nabla r_3 - r_3 (\bar{\mathbf{r}}_1 + \bar{\mathbf{r}}_2) \quad (51)$$

$$\nabla r_i = \frac{\bar{\mathbf{r}}_i}{r_i}. \quad (52)$$

Jacobian of ∇B yields

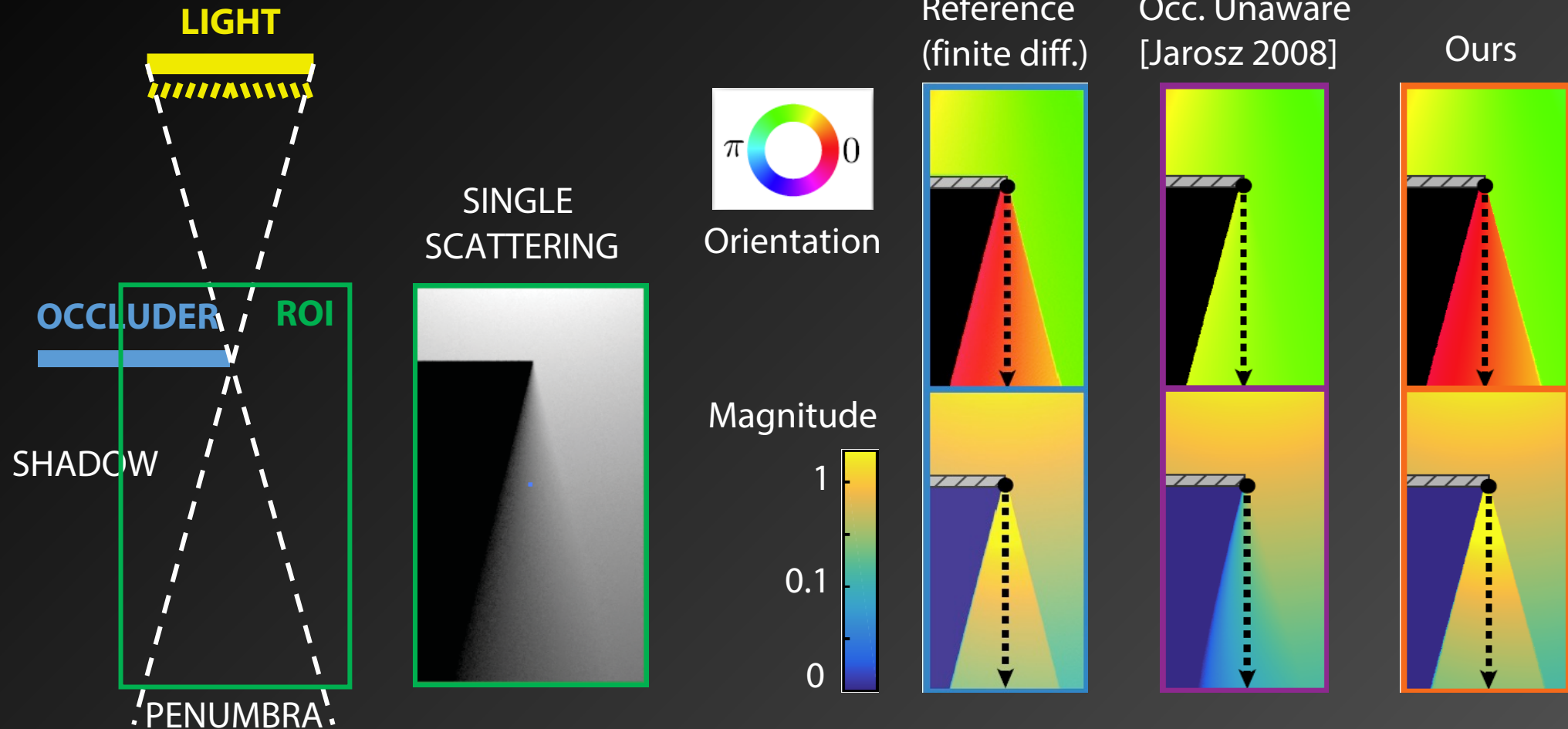
$$J(\nabla B) = J(\nabla(r_1 r_2 r_3)) + J(\nabla((\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_2) r_3)) \\ + J(\nabla((\bar{\mathbf{r}}_2 \cdot \bar{\mathbf{r}}_3) r_1)) + J(\nabla((\bar{\mathbf{r}}_1 \cdot \bar{\mathbf{r}}_3) r_2)) \quad (53)$$

where

$$J(\nabla(r_1 r_2 r_3)) = r_2 r_3 J(\nabla r_1) + r_1 r_3 J(\nabla r_2) + r_1 r_2 J(\nabla r_3) \\ + r_1 r_3 J(\nabla r_2) + r_2 r_3 J(\nabla r_1) + r_1 r_2 J(\nabla r_3) \\ + r_1 r_3 J(\nabla r_2) + r_2 r_3 J(\nabla r_1) + r_1 r_2 J(\nabla r_3) \quad (54)$$

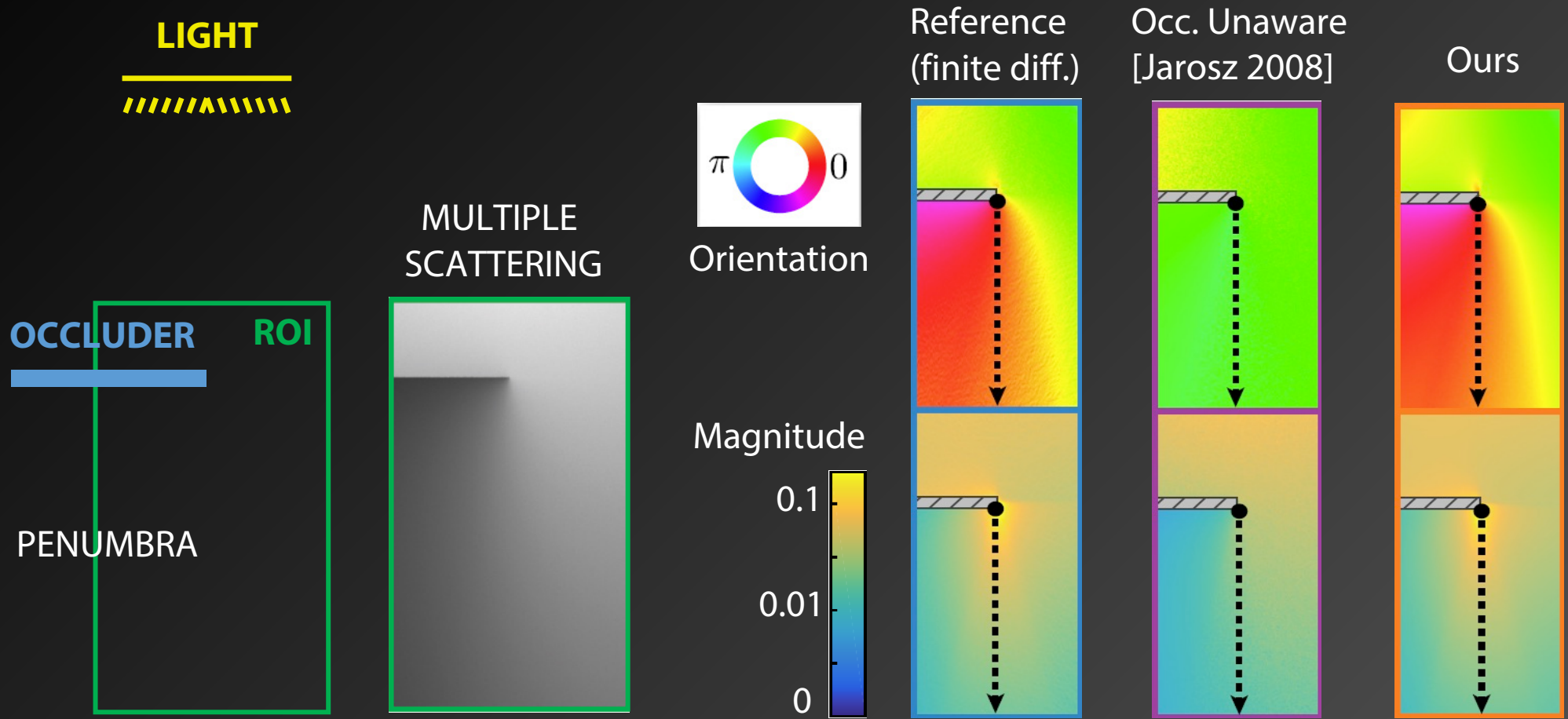


2D visualization of derivatives





2D visualization of derivatives





Results



Statues – Render comparison



Reference

[Jarosz et al. 2008]

Ours



Statues – Render comparison



Reference



[Jarosz et al. 2008]



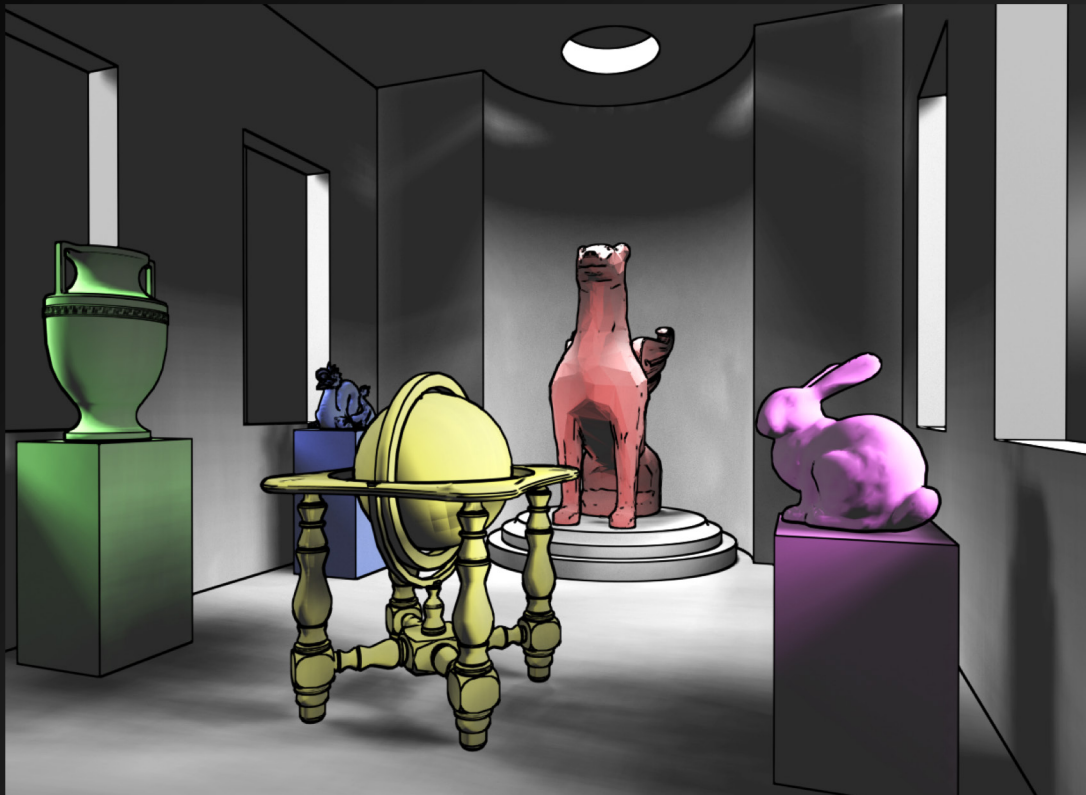
Ours



Statues – Cache distribution

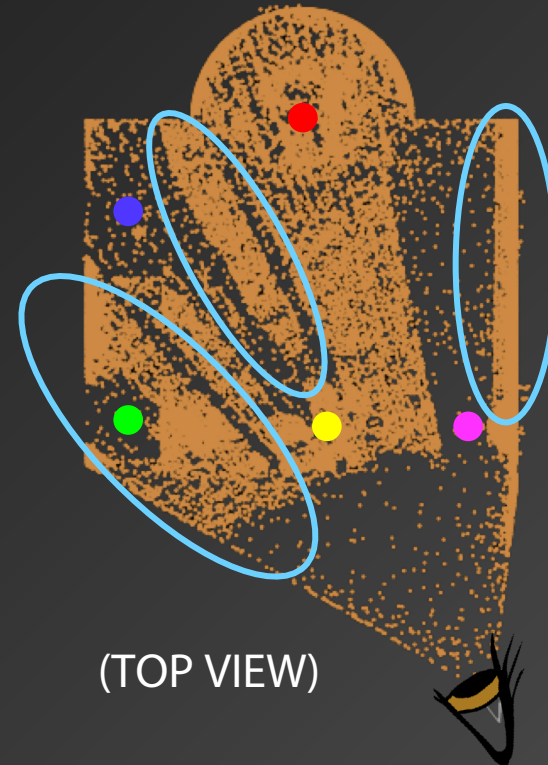
[Jarosz et al. 2008]

Occlusion-unaware, 1st order metric

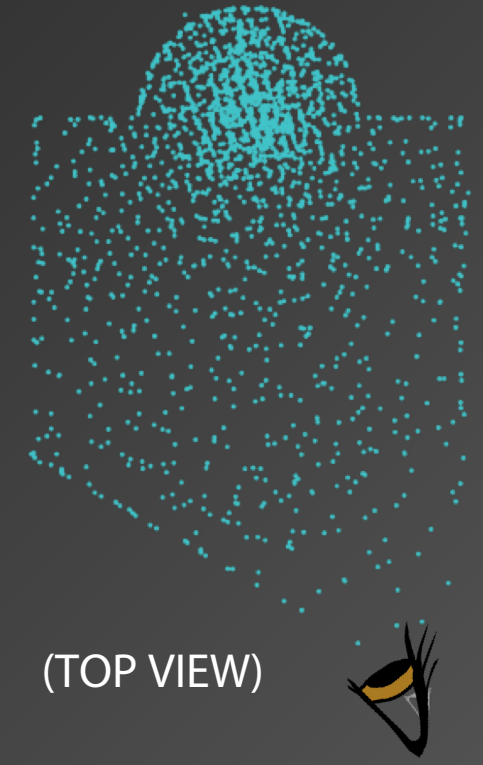


Single scattering

Multiple scattering



(TOP VIEW)



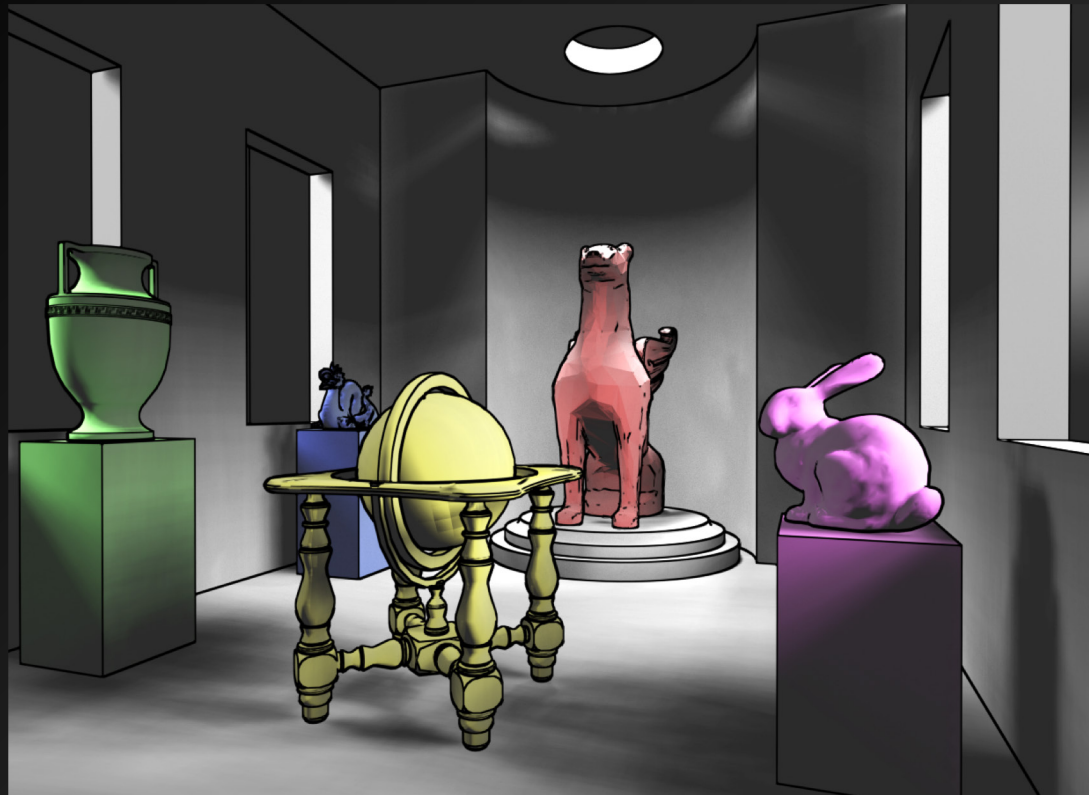
(TOP VIEW)



Statues – Cache distribution

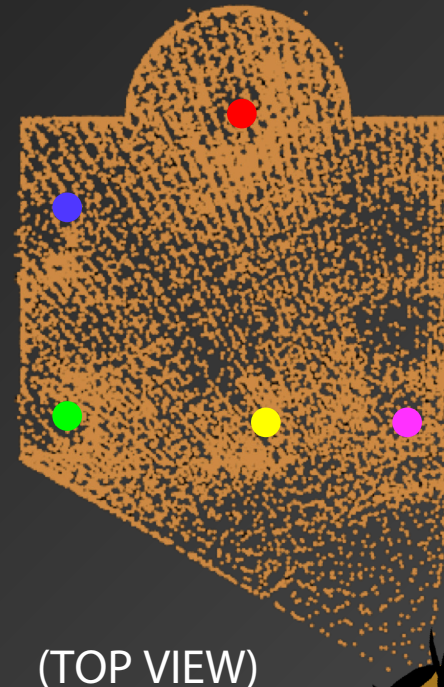
Ours

Occlusion-aware, 2nd-order metric

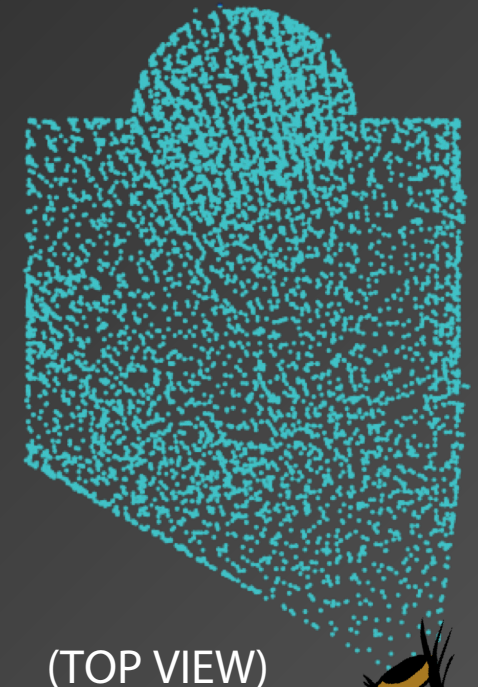


Single scattering

Multiple scattering



(TOP VIEW)



(TOP VIEW)



Patio – Time performance

Our computational overhead

- Triangulation
 - Hessian computation
- } **+9%**

$$\mathbf{H}F_{\Delta}(\mathbf{x}) = \frac{1}{2\pi} \left(\frac{\nabla(|A|)\nabla^{\top}B - \nabla B\nabla^{\top}(|A|)}{|A|^2 + B^2} + \frac{B\mathbf{J}(\nabla(|A|)) - |A|\mathbf{J}(\nabla B)}{|A|^2 + B^2} - \frac{(B\nabla(|A|) - |A|\nabla B)(\nabla(|A|^2) + \nabla(B^2))^{\top}}{(|A|^2 + B^2)^2} \right)$$

$$\begin{aligned} \nabla A &= (\langle \vec{r}_2, \mathbf{J}(\vec{r}_3) \rangle - \langle \vec{r}_3, \mathbf{J}(\vec{r}_2) \rangle) \vec{r}_1 - (\vec{r}_2 \times \vec{r}_3) \\ &= \langle \vec{r}_3 - \vec{r}_2, \vec{r}_1 \rangle - (\vec{r}_2 \times \vec{r}_3). \end{aligned}$$

$$\begin{aligned} \nabla B &= \nabla(r_1 r_2 r_3) + \nabla((\vec{r}_1 \cdot \vec{r}_2) r_3) \\ &\quad + \nabla((\vec{r}_2 \cdot \vec{r}_3) r_1) + \nabla((\vec{r}_1 \cdot \vec{r}_3) r_2) \end{aligned}$$

$$\begin{aligned} \mathbf{J}(\nabla B) &= \mathbf{J}(\nabla(r_1 r_2 r_3)) + \mathbf{J}(\nabla((\vec{r}_1 \cdot \vec{r}_2) r_3)) \\ &\quad + \mathbf{J}(\nabla((\vec{r}_2 \cdot \vec{r}_3) r_1)) + \mathbf{J}(\nabla((\vec{r}_1 \cdot \vec{r}_3) r_2)) \end{aligned}$$



Patio – Time performance

Our computational overhead

- Triangulation
 - Hessian computation
- } +9%

Equal-time



Ours (iso. cache)
135 min., 32k points

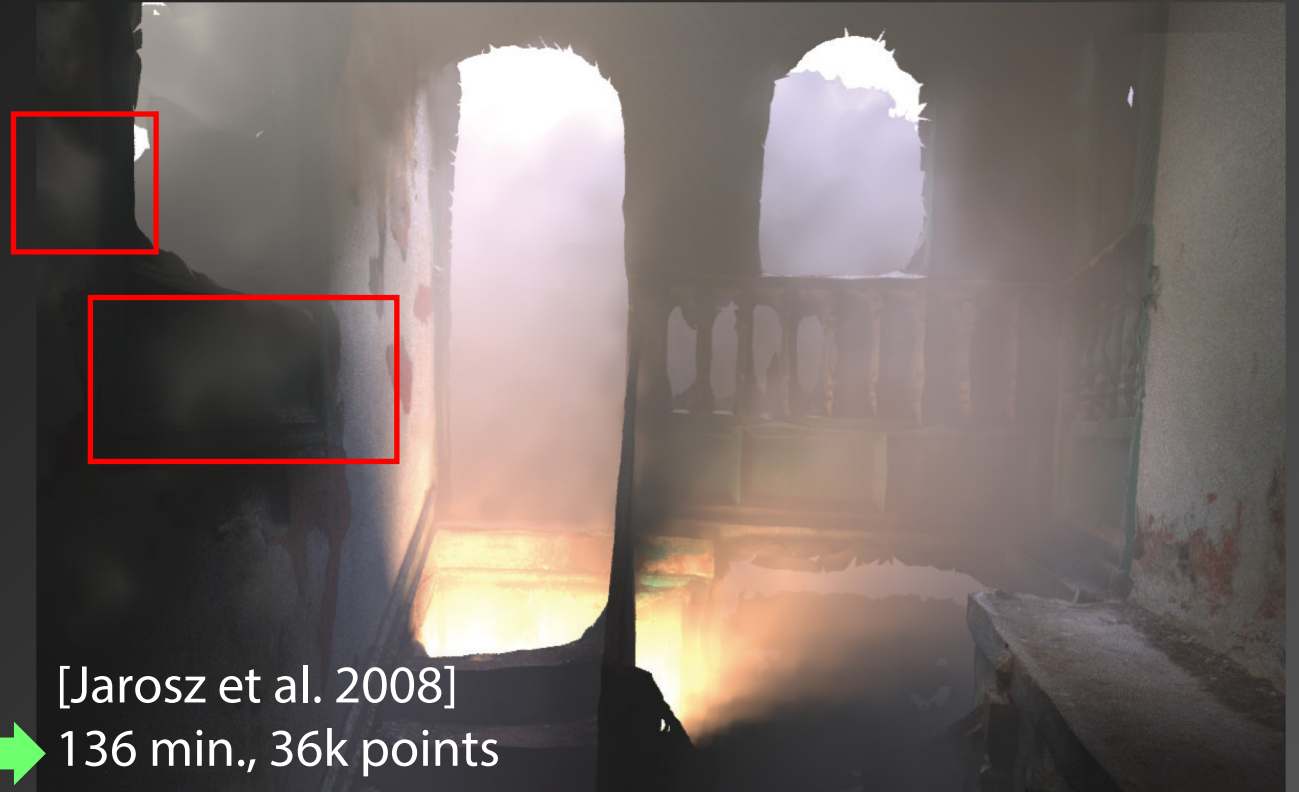




Patio – Time performance

Our computational overhead

- Triangulation
 - Hessian computation
- } +9%



Equal-time



[Jarosz et al. 2008]
136 min., 36k points



Patio – Time performance

Our computational overhead

- Triangulation
 - Hessian computation
- } +9%

Same error threshold
30% faster



Ours (aniso. cache)
94 min., 21k points

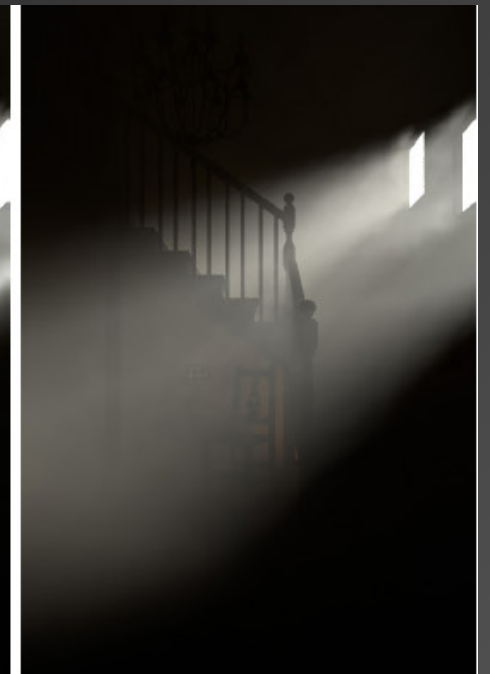
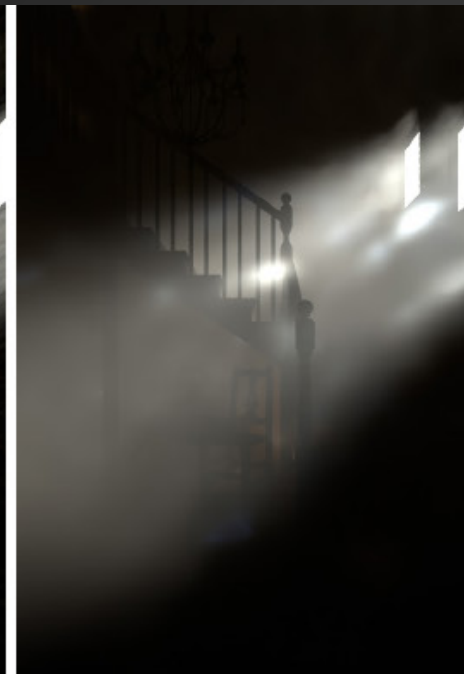




Equal-time comparison

Reference

EQUAL TIME



Path tracing

[Jarosz et al. 2011]
Progressive photon
beams

[Jarosz et al. 2008]
Occlusion-unaware
first-order

Ours
Second-order
occlusion-aware



Future work

- Extend to support scattering from glossy materials
- Limited to finite light sources
- Extend to anisotropic media and heterogeneous materials



Conclusions

- Computation of occlusion-aware media derivatives
- Second-order error metric for volumetric radiance caching
- ... radiance derivatives useful for other applications!



Thanks!