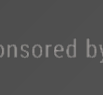
Second-Order Occlusion-Aware Volumetric Radiance Caching

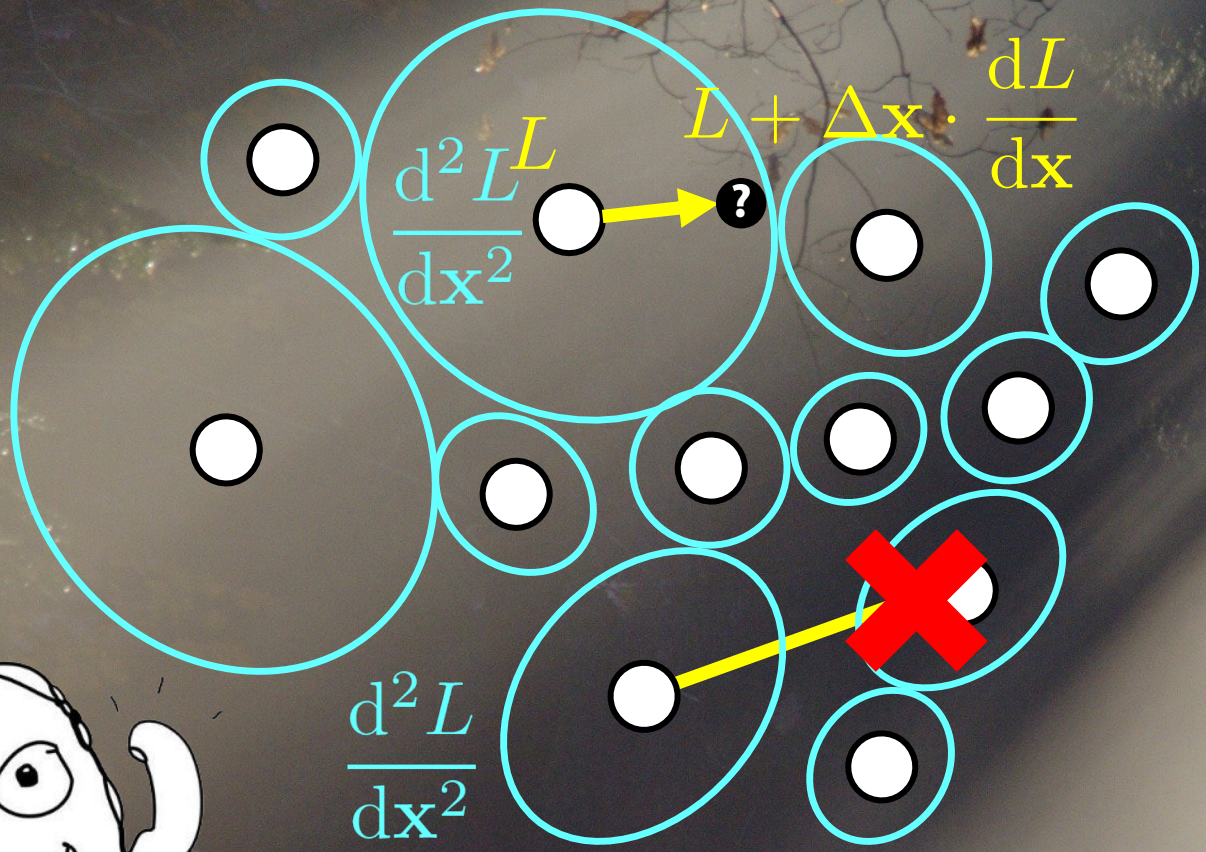
Julio Marco[1] Adrian Jarabo[1] Wojciech Jarosz[2] Diego Gutierrez[1]

[1] Universidad de Zaragoza, I3A [2] Dartmouth College



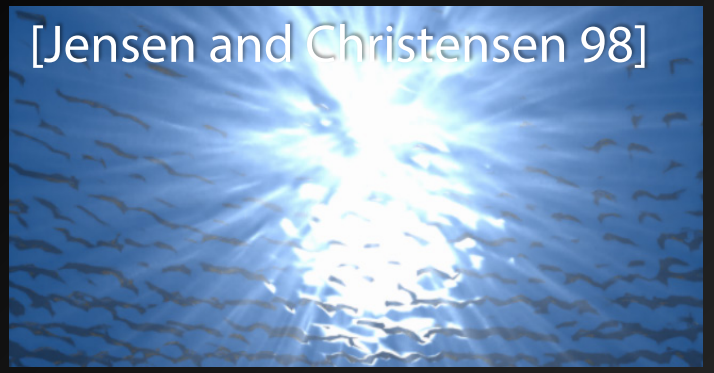

[Schwarzhaupt et al. 2012]

OUR METHOD


L
$\frac{\mathrm{d}^2 L}{\mathrm{d}\mathbf{x}^2}$
?
$L + \Delta\mathbf{x} \cdot \frac{\mathrm{d}L}{\mathrm{d}\mathbf{x}}$
$\frac{\mathrm{d}^2 L}{\mathrm{d}\mathbf{x}^2}$

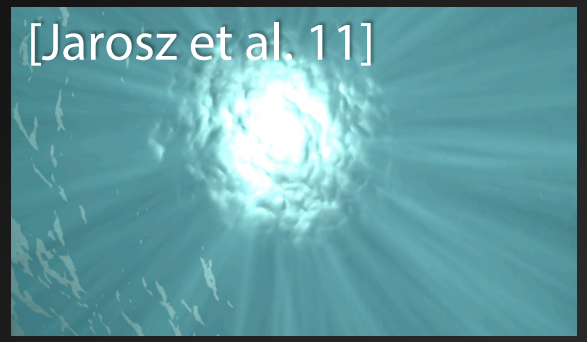
Related Work

- 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]

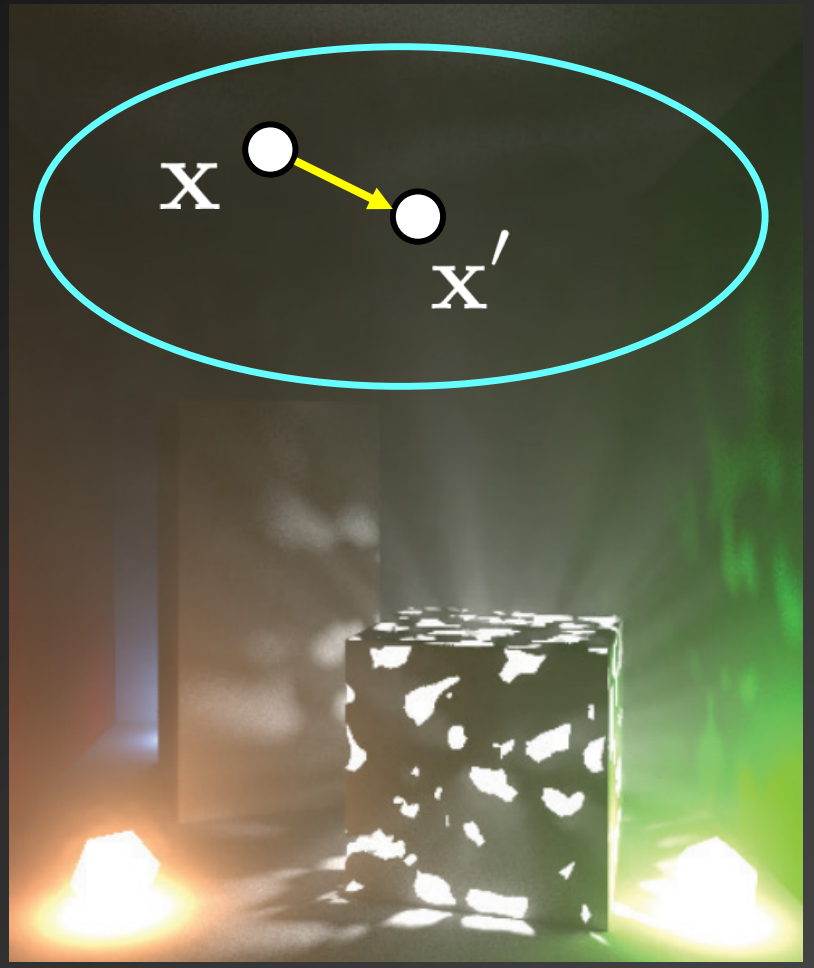
- Media gradients ➔ **YES**
- Occlusions ➔ **NO**
- Higher-order ➔ **NO**

Related Work

[Jarosz et al. 2008]

Related Work

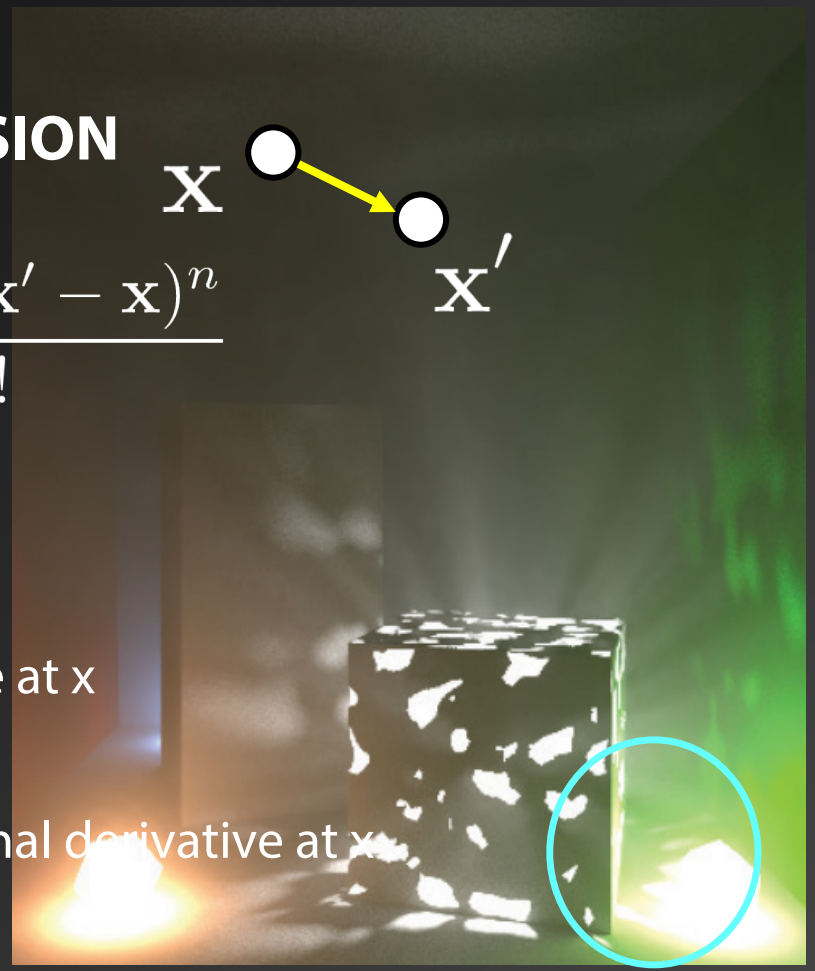
Radiance extrapolation



Radiance extrapolation

TAYLOR EXPANSION

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



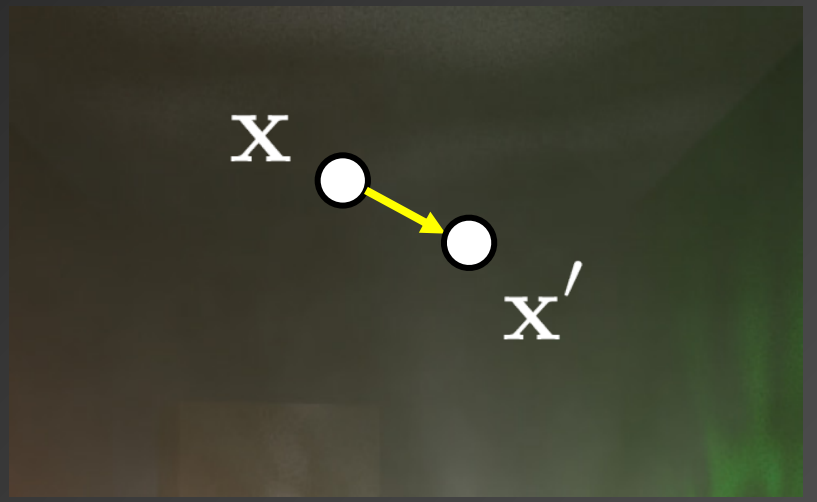
$L^{(0)}(\mathbf{x})$ Radiance value at x

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

Radiance extrapolation

TRUNCATED TAYLOR EXPANSION

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

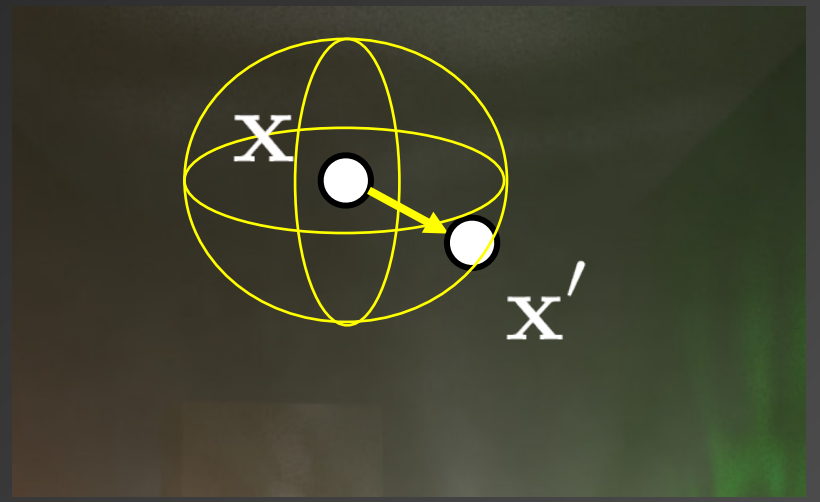


$$\mathcal{M}_1 = \sum_{n=2}^{\infty} \frac{L^{(n)}(x)}{n!}(\mathbf{x}' - \mathbf{x})^n$$

Radiance extrapolation

TRUNCATED TAYLOR EXPANSION

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



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

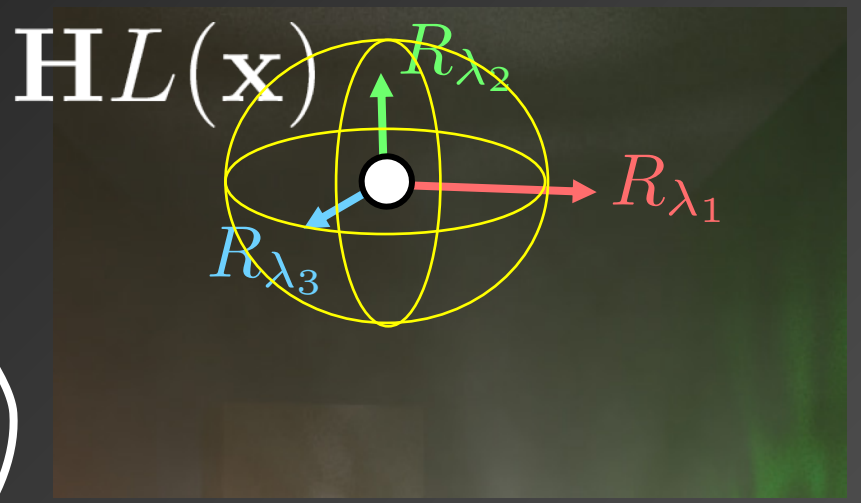
$$\mathcal{M}_1 \approx \frac{\mathbf{H}L(\mathbf{x})}{2}(\mathbf{x}' - \mathbf{x})^2$$

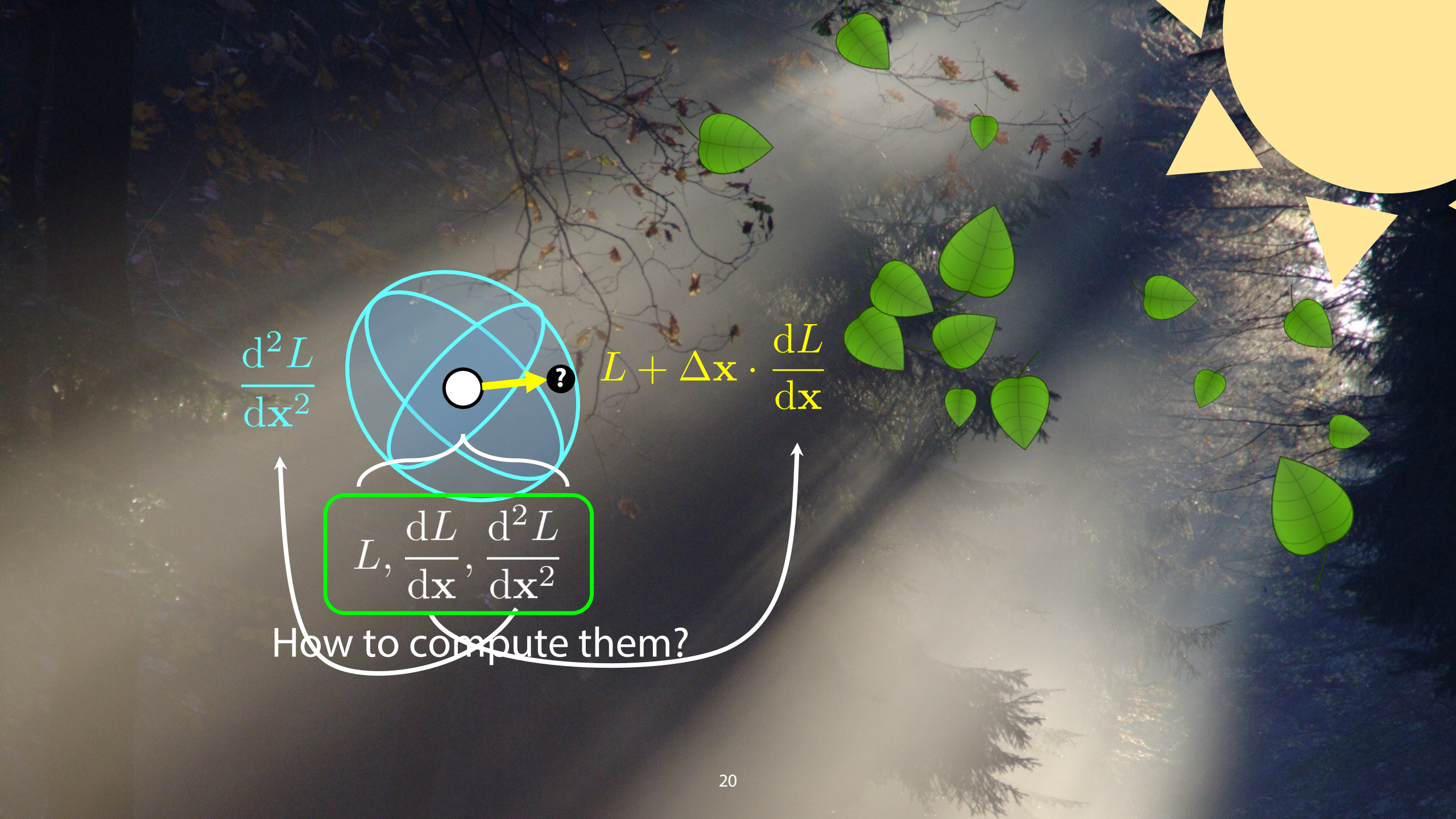
Radiance extrapolation

$$\{\lambda_1, \lambda_2, \lambda_3\} \rightarrow R_{\lambda_i} = \sqrt[5]{\frac{15L(\mathbf{x})\varepsilon}{4\pi|\lambda_i|}}$$

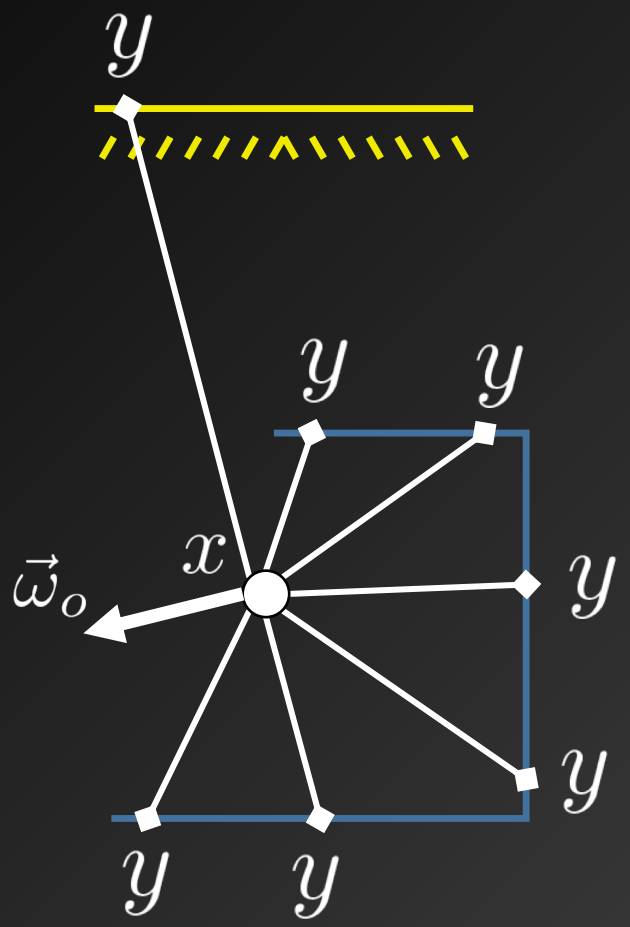
Eigenvalues of Hessian

Error threshold




$\frac{\mathrm{d}^2 L}{\mathrm{dx}^2}$
?
$L + \Delta\mathbf{x} \cdot \frac{\mathrm{d}L}{\mathrm{dx}}$
$L, \frac{\mathrm{d}L}{\mathrm{dx}}, \frac{\mathrm{d}^2 L}{\mathrm{dx}^2}$
How to compute them?

Derivative computation



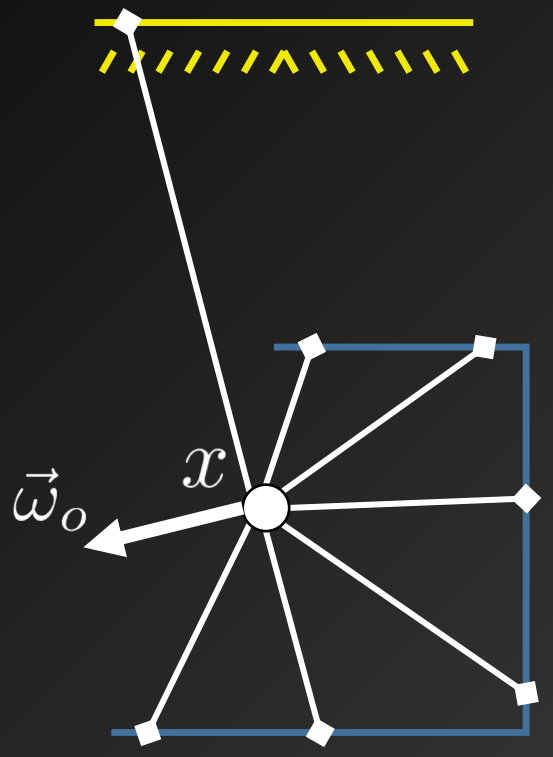
$$L(x) = \int_S f T_x G V L(y) dy$$

TRANSFER FUNCTION

INCOMING RADIANCE

Derivative computation

[Jarosz et al. 2008]

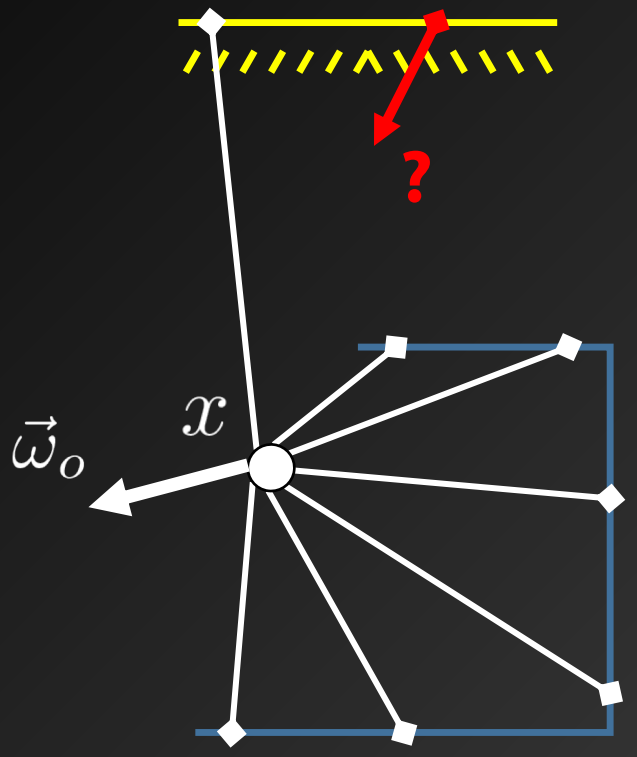


$$L(x) = \int_S f\, T_r G V L(y)\, dy$$

Ignored in gradient computation

Derivative computation

[Jarosz et al. 2008]

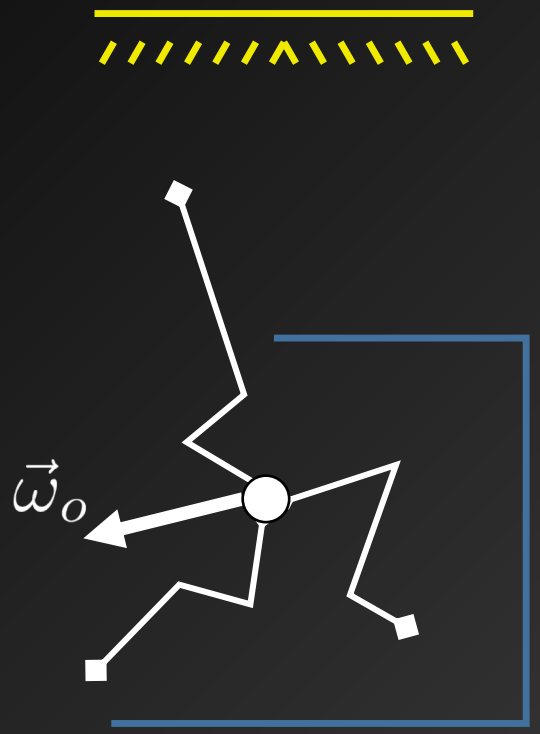


$$L(x) = \int_S f\, T_r G V L(y)\, dy$$

Ignored in gradient computation

Derivative computation

[Jarosz et al. 2008]

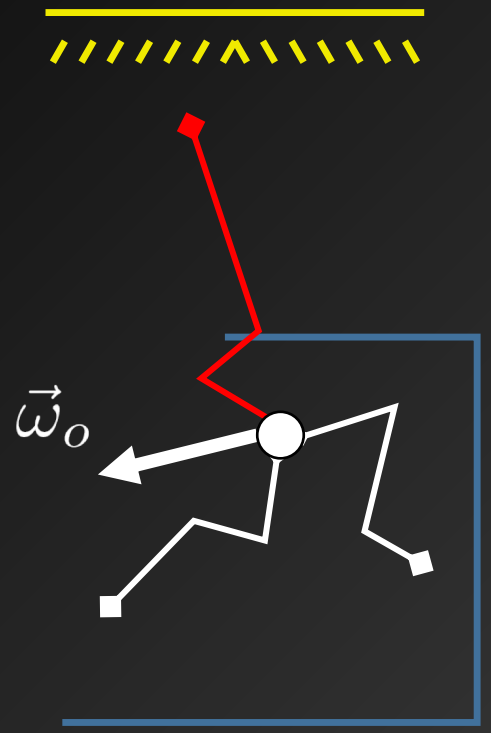


$$L(x) = \int_S f\, T_r G V L(y)\, dy$$

Ignored in gradient computation

Derivative computation

[Jarosz et al. 2008]

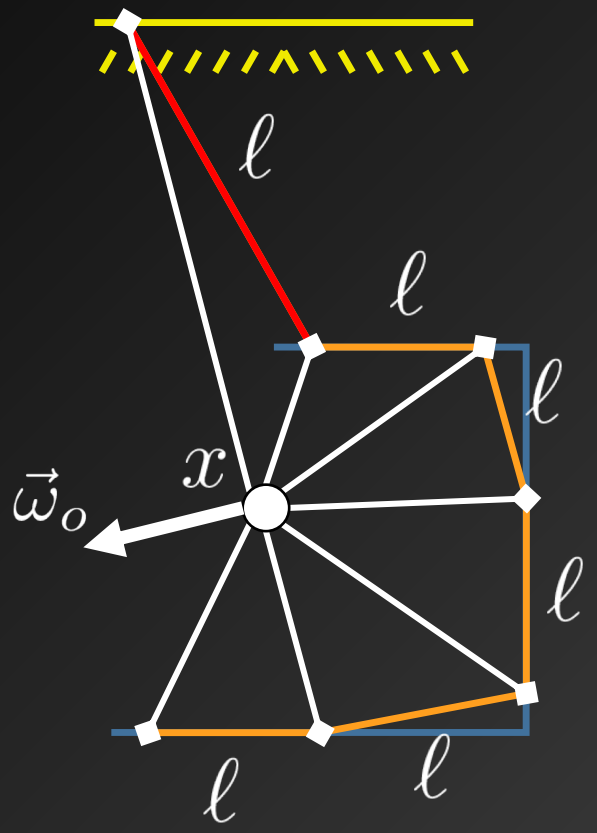


$$L(x) = \int_S f\, T_r G V L(y)\, dy$$

Ignored in gradient computation

Derivative computation

Our method



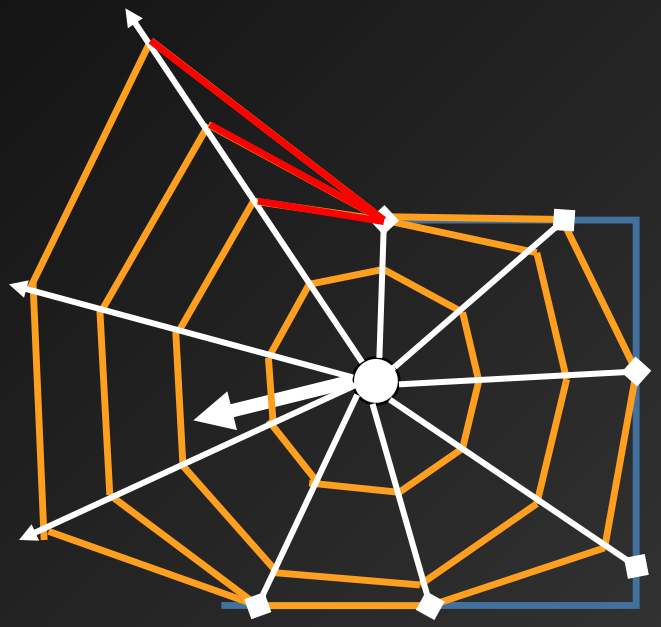
$$L(x) = \int_S f\, T_r G V\, L(y)\, dy$$

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

Gone!

Derivative computation

Our method

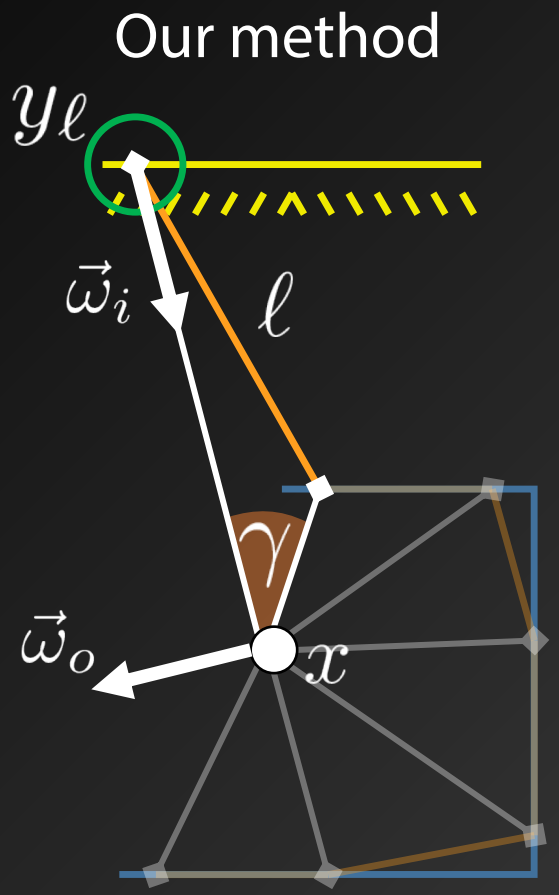


$$L(x) = \int_M f T_r G V L(y)\, dy$$

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

RAY MARCHING

Derivative computation



For a single triangle

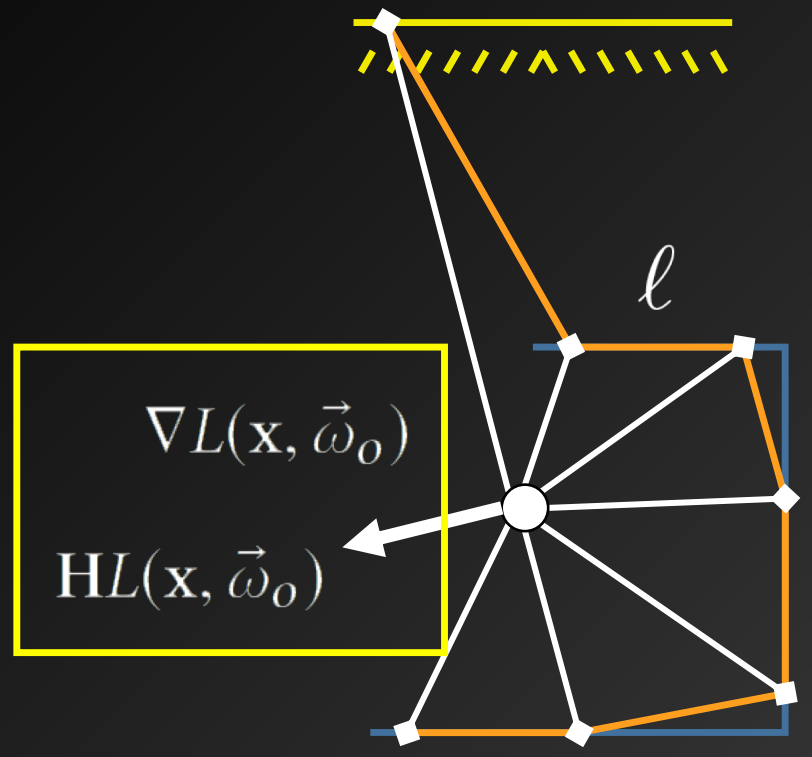
$$L(x) = \int_\ell f \, T_r(G_\ell) \, L(F_\ell) \, (dy)$$

Triangle-to-medium form factor

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

Derivative computation

Our method

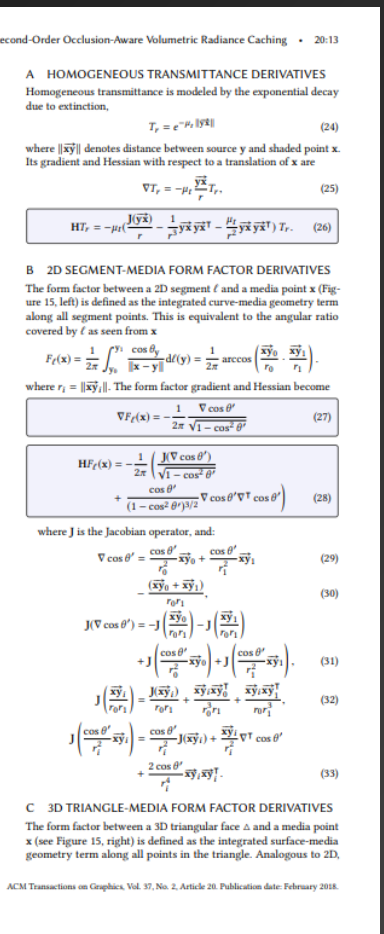


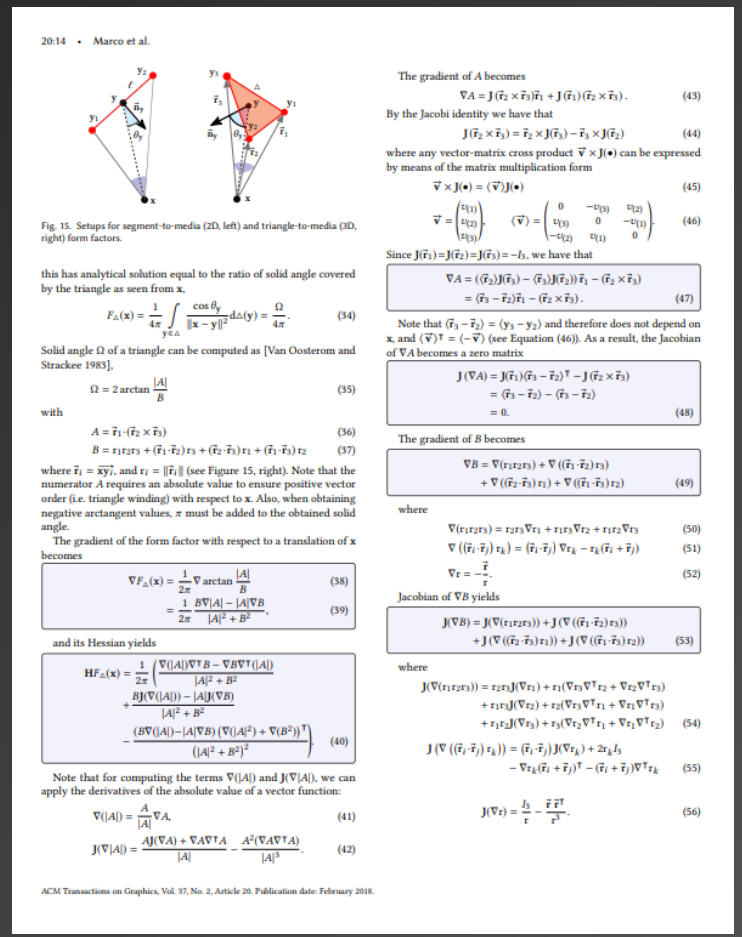
$$L(x) = f_\ell T_r L(y_\ell)\, F_\ell(x)$$

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

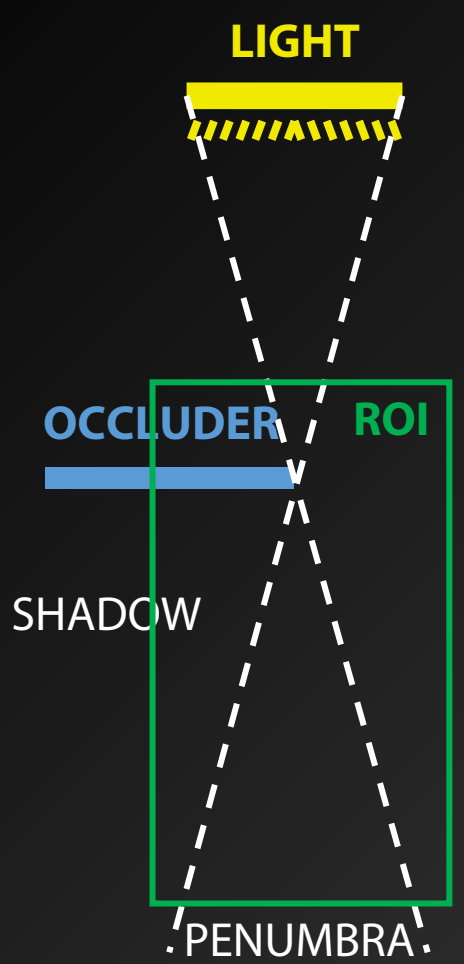
$$\mathrm{H}L(x) = \sum_{r_i} \sum_{\ell_j} \frac{\mathrm{H}L_{\ell_j}(x)}{\mathrm{pdf}(r_i)}$$

Derivative computation

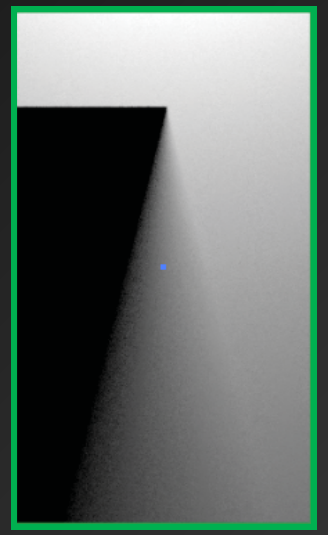




2D visualization of derivatives



SINGLE SCATTERING

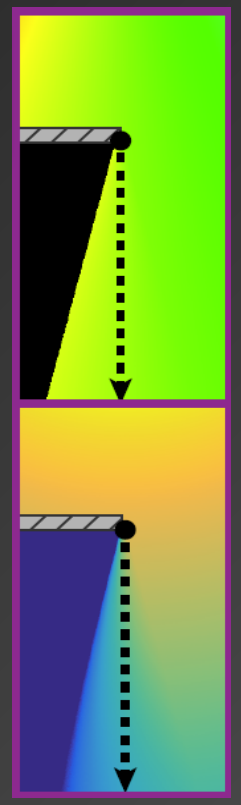


Orientation

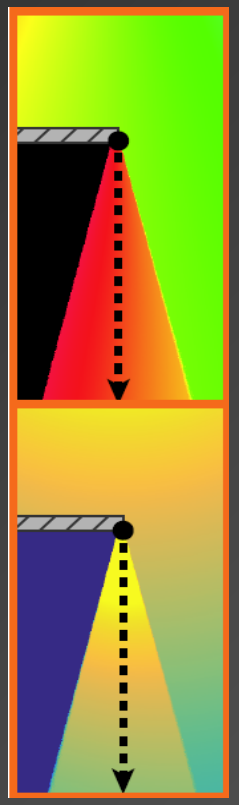
Magnitude

Reference (finite diff.)

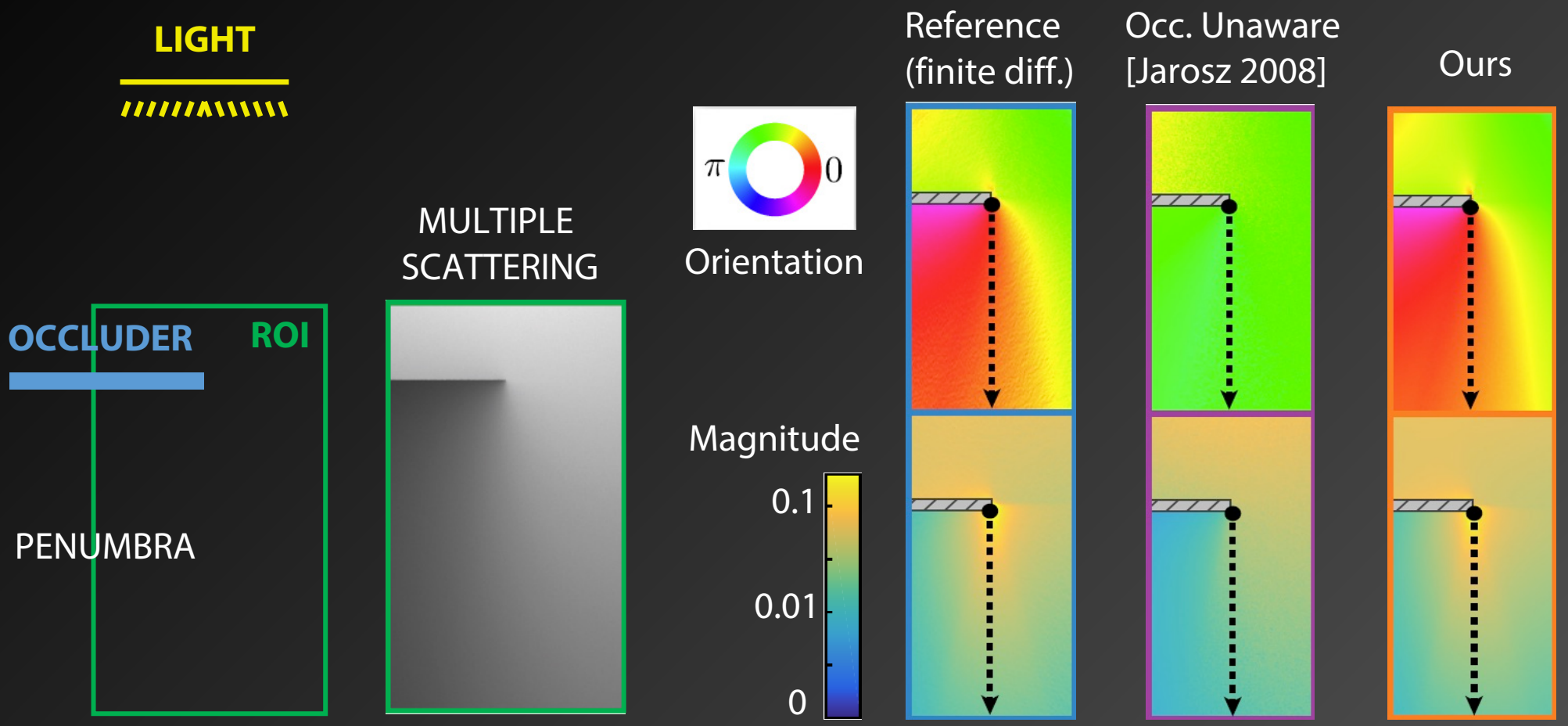
Occ. Unaware [Jarosz 2008]



Ours

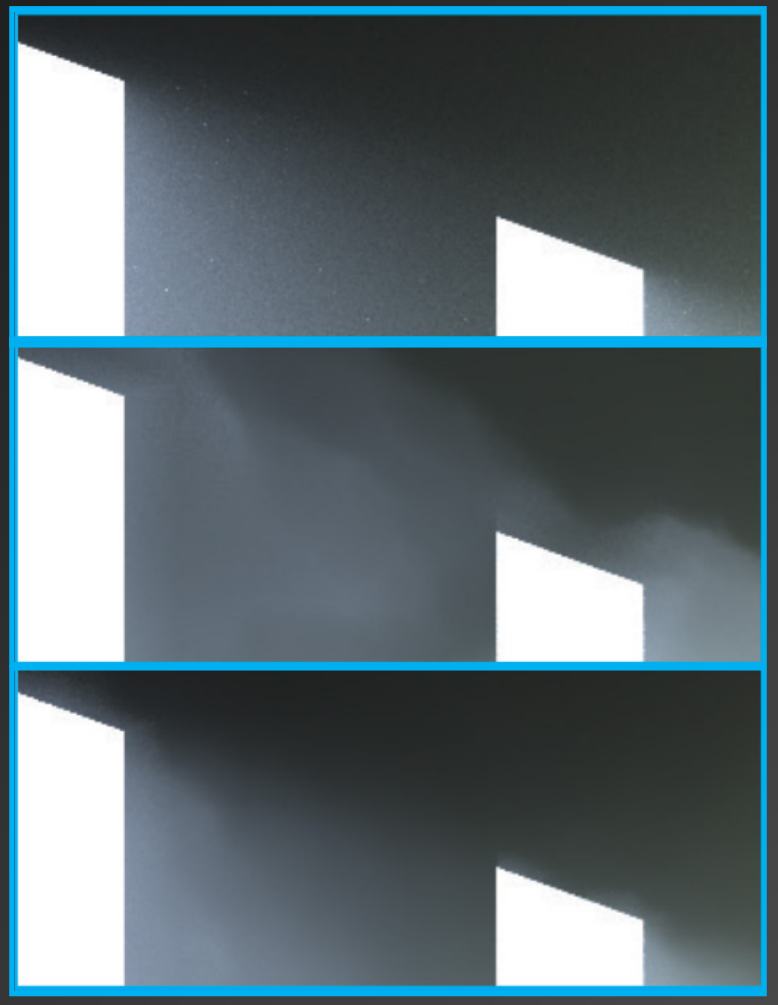


2D visualization of derivatives



Results

Statues – Render comparison

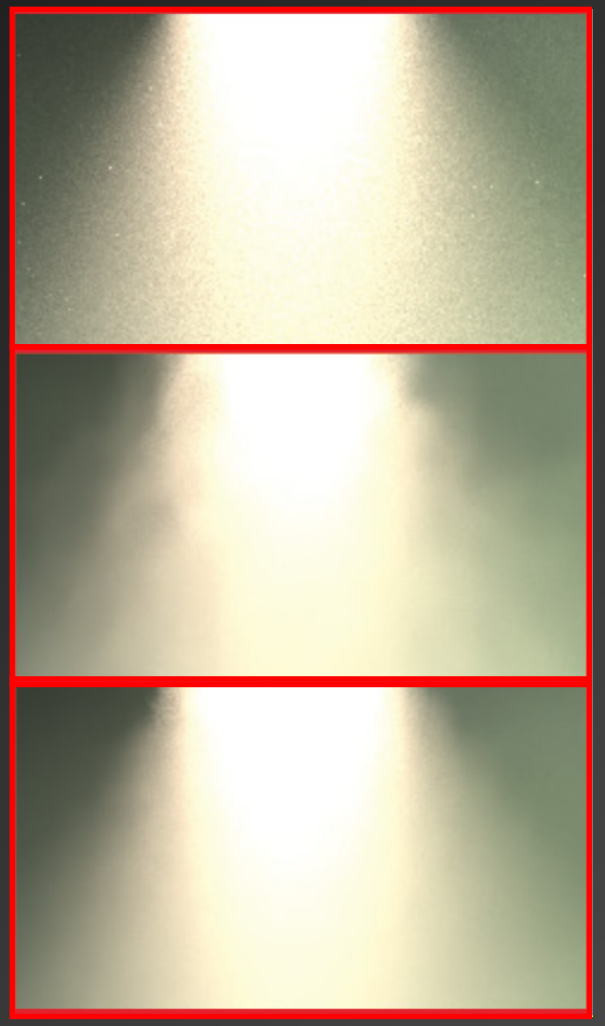


Reference

[Jarosz et al. 2008]

Ours

Statues – Render comparison

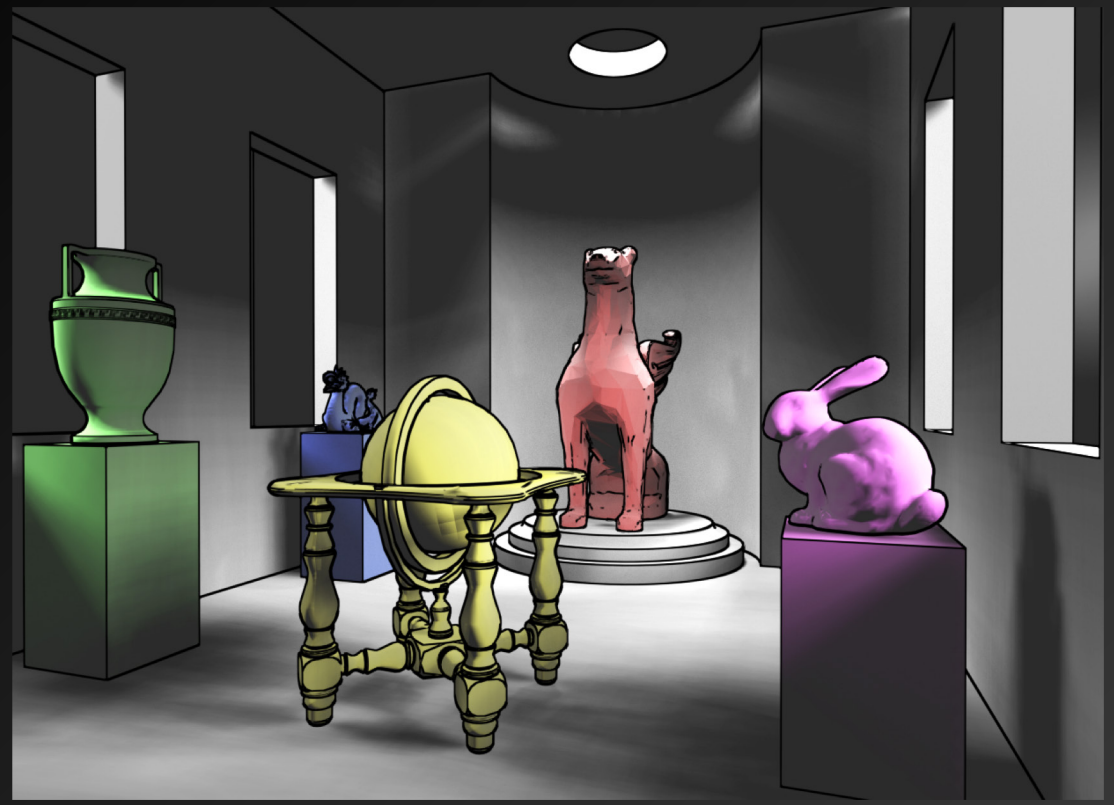


Reference

[Jarosz et al. 2008]

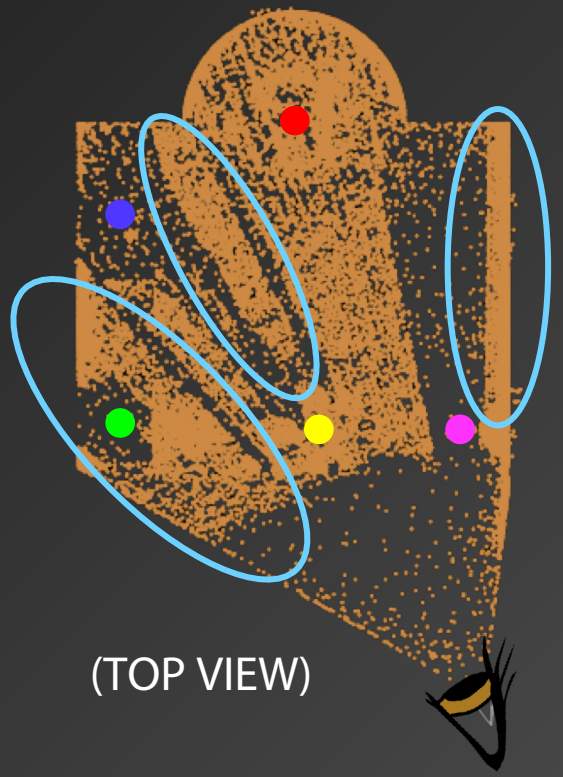
Ours

Statues – Cache distribution



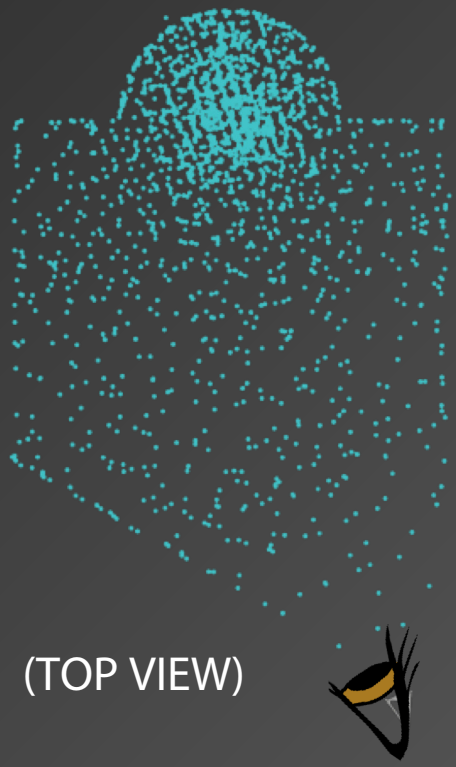
[Jarosz et al. 2008]
Occlusion-unaware, 1st order metric

Single scattering



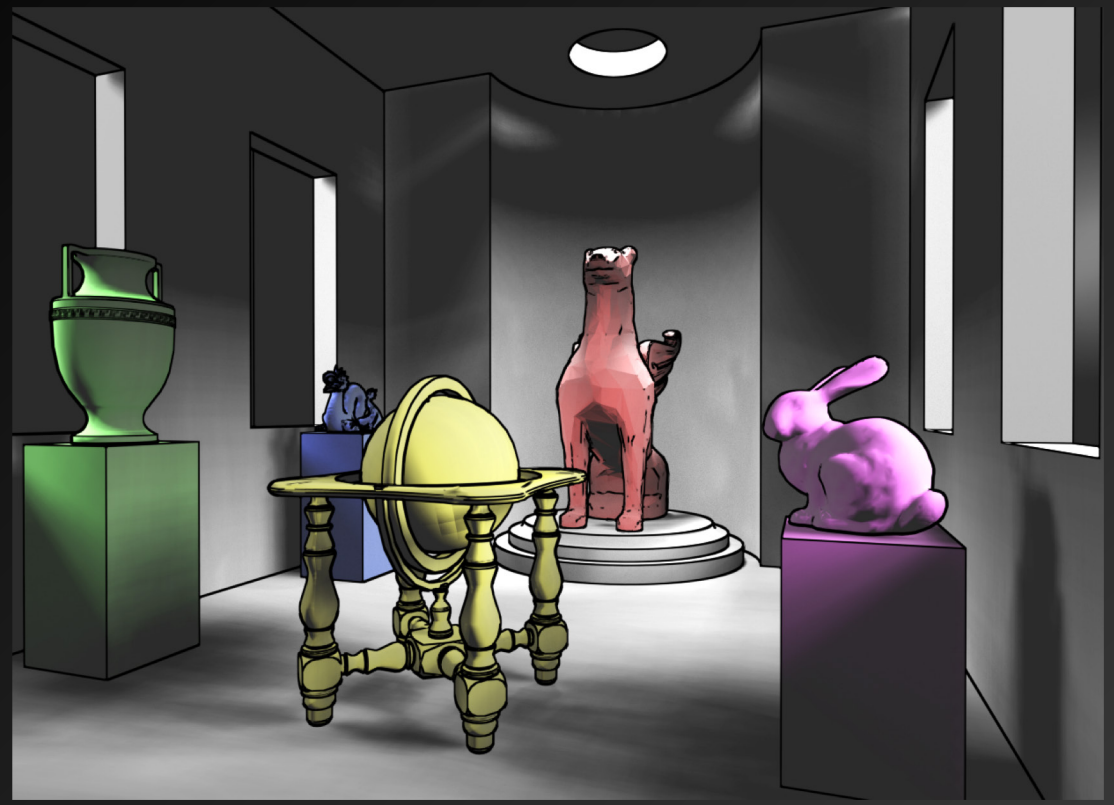
(TOP VIEW)

Multiple scattering



(TOP VIEW)

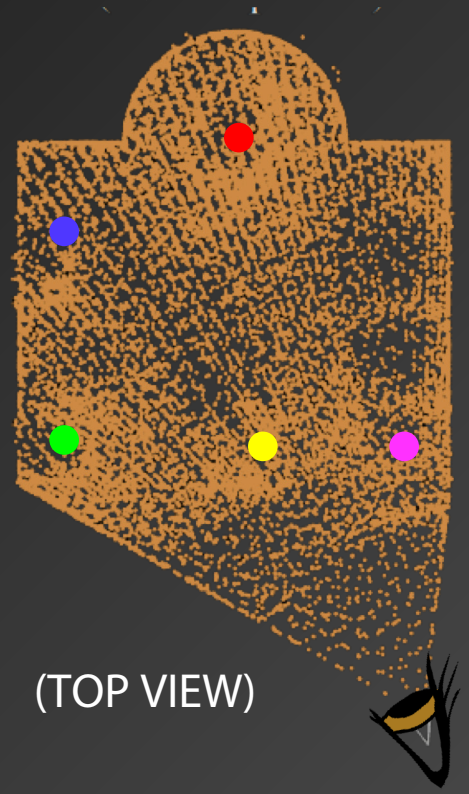
Statues – Cache distribution



Ours

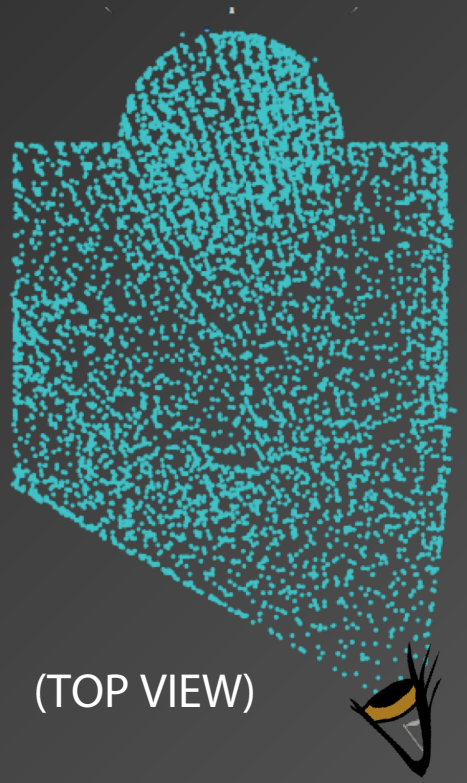
Occlusion-aware, 2nd-order metric

Single scattering



(TOP VIEW)

Multiple scattering



(TOP VIEW)

Patio – Time performance

Our computational overhead

- Triangulation
- Hessian computation

} **+9%**

$$\mathrm{H}F_{\triangle}(\mathrm{x}) = \frac{1}{2\pi}\left(\frac{\nabla(|A|)\nabla^{\mathsf{T}}B - \nabla B\nabla^{\mathsf{T}}(|A|)}{|A|^2+B^2} + \frac{B\mathrm{J}(\nabla(|A|)) - |A|\mathrm{J}(\nabla B)}{|A|^2+B^2} - \frac{(B\nabla(|A|) - |A|\nabla B)\left(\nabla(|A|^2)+\nabla(B^2)\right)^{\mathsf{T}}}{\left(|A|^2+B^2\right)^2}\right).$$

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

$$\begin{aligned}\nabla B &= \nabla(\mathrm{r}_1\mathrm{r}_2\mathrm{r}_3) + \nabla\left((\vec{\mathrm{r}}_1\cdot\vec{\mathrm{r}}_2)\,\mathrm{r}_3\right)\\ &\quad + \nabla\left((\vec{\mathrm{r}}_2\cdot\vec{\mathrm{r}}_3)\,\mathrm{r}_1\right) + \nabla\left((\vec{\mathrm{r}}_1\cdot\vec{\mathrm{r}}_3)\,\mathrm{r}_2\right)\\ \mathrm{J}(\nabla B) &= \mathrm{J}(\nabla(\mathrm{r}_1\mathrm{r}_2\mathrm{r}_3)) + \mathrm{J}\left(\nabla\left((\vec{\mathrm{r}}_1\cdot\vec{\mathrm{r}}_2)\,\mathrm{r}_3\right)\right)\\ &\quad + \mathrm{J}\left(\nabla\left((\vec{\mathrm{r}}_2\cdot\vec{\mathrm{r}}_3)\,\mathrm{r}_1\right)\right) + \mathrm{J}\left(\nabla\left((\vec{\mathrm{r}}_1\cdot\vec{\mathrm{r}}_3)\,\mathrm{r}_2\right)\right)\end{aligned}$$

Patio – Time performance

Our computational overhead

- Triangulation
- Hessian computation

} **+9%**

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

Equal-time ➡

Patio – Time performance

Our computational overhead

- Triangulation
- Hessian computation

} **+9%**

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

Equal-time ➡

Patio – Time performance

Our computational overhead

- Triangulation
- Hessian computation

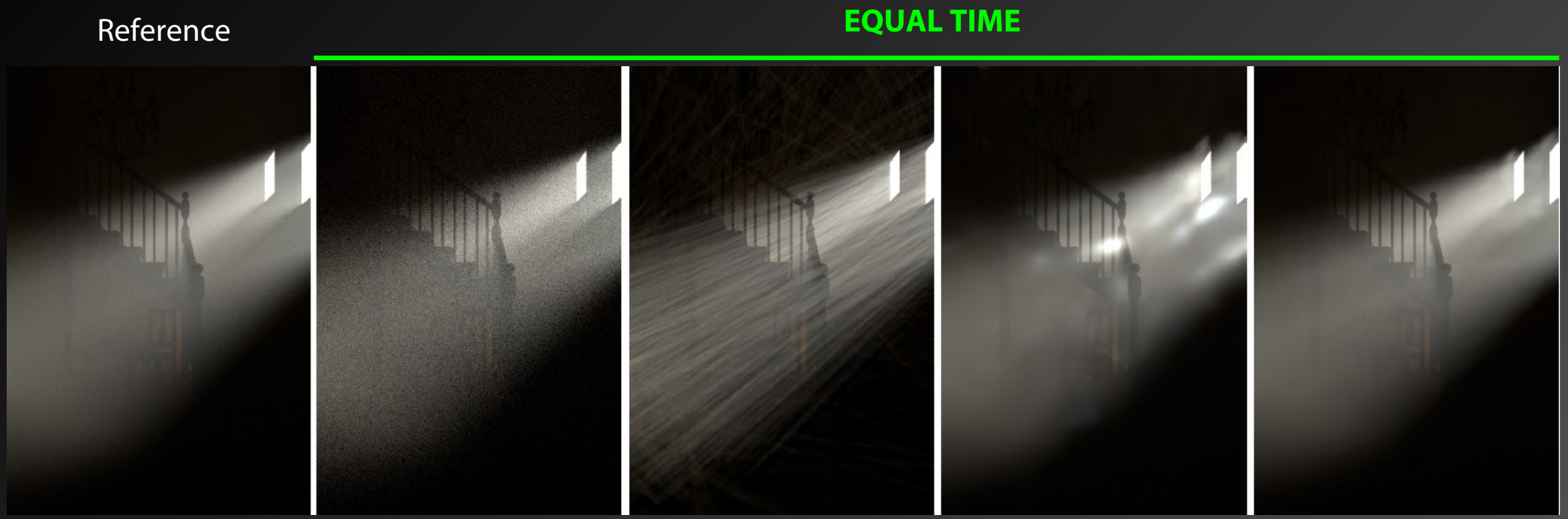
} **+9%**

Same error threshold

30% faster

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

Equal-time comparison



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!