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
Related Work

• Photon-based methods

[Jensen and Christensen 98] [Knaus and Zwicker 11]

POINTS

[Jarosz et al. 11] [Jarosz et al. 11]

BEAMS

[Bitterli and Jarosz 17]

Higher-dimensional

[Krivanek et al. 14]

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

[Ward et al. 88]

[Ward and Heckbert 91]
Related Work

• Irradiance/radiance caching methods

[Krivanek et al. 2005]

[Krivanek et al. 2006]
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

[Image: Related Work diagram with text]

Related Work in the context of surfaces and media, as per Jarosz et al. 2008, Schwarzhaupt et al. 2012, and our own approach.
Radiance extrapolation
Radiance extrapolation

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

- \( L^{(0)}(x) \): Radiance value at \( x \)
- \( L^{(n)}(x) \): \( n \)-th translational derivative at \( x \)
Radiance extrapolation

**TRUNCATED TAYLOR EXPANSION**

\[
L(x') = L(x) + \nabla L(x)(x' - x) + M_1
\]

- **Actual value**
- **Extrapolated value at x' (first order)**
- **Error**

\[
M_1 = \sum_{n=2}^{\infty} \frac{L^{(n)}(x)}{n!} (x' - x)^n
\]
Radiance extrapolation

**TRUNCATED TAYLOR EXPANSION**

\[ L(x') = L(x) + \nabla L(x)(x' - x) + \mathcal{M}_1 \]

- Actual value
- Extrapolated value at \( x' \) (first order)
- Error

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

\[ \mathcal{M}_1 \approx \frac{HL(x)}{2} (x' - x)^2 \]
Radiance extrapolation

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

Eigenvalues of Hessian

Error threshold
How to compute them?
Derivative computation

\[ L(x) = \int_S f(T, G(V, L(y)) dy \]

- \( f(T, G(V, L) \) is the incoming radiance function.
- \( T \) and \( G \) are parameters.
- \( V \) is the viewing direction.
- The integral is taken over the surface \( S \) of the scene.

\[ x \]
\[ y \]
\[ \omega_0 \]
Derivative computation

\[ L(x) = \int_S f \cdot T_{\mathbf{r}} \mathbf{G} \mathbf{V} L(y) \, dy \]

Ignored in gradient computation

[Jarosz et al. 2008]
Derivative computation

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

Ignored in gradient computation

[Jarosz et al. 2008]
Derivative computation

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

[Jarosz et al. 2008]

Ignored in gradient computation
Derivative computation

\[ L(x) = \int_S f \cdot T_rG \nabla L(y) \, dy \]

[Jarosz et al. 2008]

Ignored in gradient computation
Derivative computation

\[ L(x) = \sum \int_{l_j} f T_r G V L(y) \, dy \]

Our method

Gone!
Derivative computation

Our method

\[ L(x) = \int_{M} f T_r G V L(y) dy \]

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

RAY MARCHING
Derivative computation

Triangle-to-medium form factor

For a single triangle

\[ L(x) = \int_{\ell} \int_{\gamma} f T_r(\Omega \Omega)(E_y)(dy) \]

Triangle-to-medium form factor

\[ f(x, \omega_i, \omega_o) \rightarrow f_{\ell}(x, \omega_o) \]

\[ \gamma \rightarrow 0 \]

Constant \( T_r(x, y) \) and \( L(y) \)
Derivative computation

\[ \nabla L(x, \vec{\omega}_o) \]

\[ \text{HL}(x, \vec{\omega}_o) \]

\[ L(x) = f_\ell T_r L(y_\ell) F_\ell(x) \]

\[ \nabla L(x) = \sum \sum \frac{\nabla L_{\ell_j}(x)}{\text{pdf}(r_i)} \]

\[ \text{HL}(x) = \sum \sum \frac{\text{HL}_{\ell_j}(x)}{\text{pdf}(r_i)} \]
Derivative computation

We refer to Appendices A, B, C, and D for all the terms. For constructing our formulas for the second-order derivatives (Equation 51), we develop the linear approximations for both sides of the surface. To do this, we work from the boundary curves. We start from the initial conditions of the problem and follow the usual method of solving the equations for the derivatives. We then use these solutions to construct the formulas for the second-order derivatives.

APPENDIX

In the following sections, we compare 2D and 3D derivatives of the function. We will use a fixed grid for this in the next section. We define the second derivative as:

\[ \nabla^2 f = \nabla \cdot (\nabla f) \]

where \( \nabla f \) is the gradient of the function and \( \nabla \) is the del operator. The gradient is defined as:

\[ \nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \]

where \( f(x, y, z) \) is the function and \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \) are the partial derivatives of the function with respect to the variables \( x, y, \) and \( z \), respectively.

REFERENCES


2D visualization of derivatives

- Reference (finite diff.)
- Occ. Unaware [Jarosz 2008]
- Ours

Orientation
Magnitude
2D visualization of derivatives

LIGHT

MULTIPLE SCATTERING

MAGNITUDE

ORIENTATION

Reference (finite diff.)

Occ. Unaware [Jarosz 2008]

Ours
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)

Sponsored by
**Patio – Time performance**

Our computational overhead

- Triangulation
- Hessian computation $\{ +9\% \}$

\[
H_{P_{\nu}}(x) = \frac{1}{2\pi} \left( \frac{\nabla |A|^2 - \nabla |B|^2}{|A|^2 + |B|^2} \right)
+ \frac{|\nabla A| - |\nabla B|}{|A|^2 + |B|^2} \left( \frac{(\delta |A|^2 - |A|^2)}{|A|^2 + |B|^2} \right)^T
\]

\[
\nabla A = (\hat{c}_2 \hat{c}_1 - \hat{c}_3 \hat{c}_2) \hat{r}_1 - (\hat{c}_2 \times \hat{r}_1),
\]

\[
\nabla B = \nabla \hat{c}_1 \hat{r}_2 \hat{r}_3 + \nabla (\hat{c}_1 \hat{c}_2 \hat{r}_3)
+ \nabla (\hat{c}_2 \hat{c}_1 \hat{r}_3) + \nabla (\hat{c}_1 \hat{c}_3 \hat{r}_2)
\]

\[
J(\nabla \hat{A}) = J(\nabla (\hat{c}_1 \hat{c}_2 \hat{r}_3)) + J(\nabla (\hat{r}_1 \hat{c}_2 \hat{r}_3))
+ J(\nabla (\hat{c}_2 \hat{r}_3 \hat{r}_1)) + J(\nabla (\hat{r}_1 \hat{r}_3 \hat{c}_2))
\]
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

\[ +9\% \]

[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

Reference

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!