Stochastic Gradient Estimation for Higher-order Differentiable Rendering

Zican Wang[1] Michael Fischer[1,2] Tobias Ritschel[1]

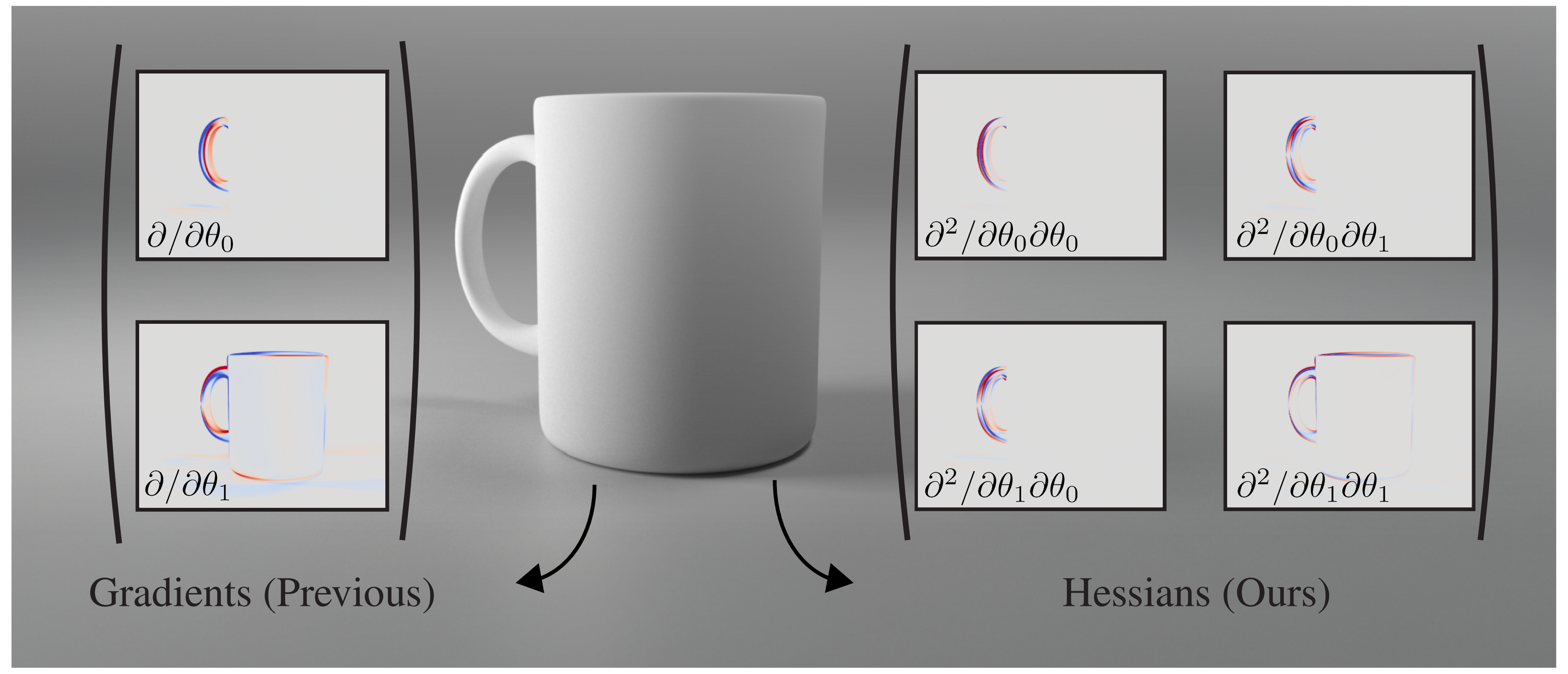
[1]University College London [2]Adobe Research

Project page & code

ICCV HONOLULU HAWAII OCT 19-23, 2025

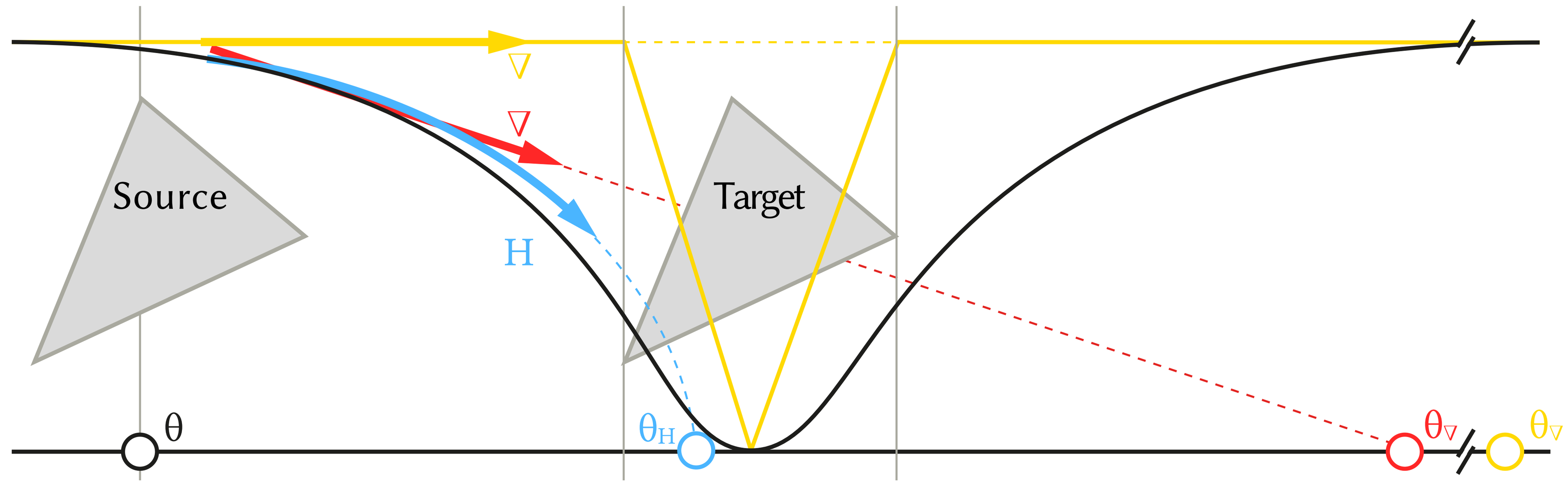
Motivation

- Differentiable renderers provide local first-order gradients ∂f/∂θ (∇)
- Previous work (FR22) convolves loss function with Gaussian.
- Reduces plateaus and supports black-box renderers.



• We propose estimators for second-order information (H) to enable larger, more reliable update steps to improve convergence speed.

• We also propose estimator for Hessian vector product (HVP) and an aggregated sampling method to improve sampling speed.v



Derivation

Key idea: convolve rendering equation L with Gaussian kernel κ and differentiate with different operator D.

$$L(\mathbf{x},\omega_\mathrm{o};\boldsymbol{\theta}) = \int_\Omega \underbrace{f_\mathrm{r}(\omega_\mathrm{i},\omega_\mathrm{o})L(\mathbf{y},\omega_\mathrm{i};\boldsymbol{\theta})}_{R(\omega_\mathrm{i};\boldsymbol{\theta})}\,\mathrm{d}\omega_\mathrm{i},$$

Convolution

$$\kappa * L(\mathbf{x},\omega_\mathrm{o};\boldsymbol{\theta}) = \bar{L}(\mathbf{x},\omega_\mathrm{o};\boldsymbol{\theta}) = \int_\Omega\int_\Theta \kappa(\boldsymbol{\tau})R(\omega_\mathrm{i};\boldsymbol{\theta}-\boldsymbol{\tau})\mathrm{d}\boldsymbol{\tau}\mathrm{d}\omega_\mathrm{i}.$$

Differentiation

$$\mathrm{D}\,\bar{L}(\mathbf{x},\omega_\mathrm{o};\boldsymbol{\theta}) = \int_\Omega\int_\Theta \mathrm{D}\,\kappa(\boldsymbol{\tau})R(\omega_\mathrm{i};\boldsymbol{\theta}-\boldsymbol{\tau})\mathrm{d}\boldsymbol{\tau}\mathrm{d}\omega_\mathrm{i}.$$

$$\mathrm{D}\,\kappa(\boldsymbol{\tau})$$

$\mathrm{D}^\mathrm{G} = \nabla = \partial/\partial\tau_i$

$\mathrm{D}^\mathrm{H} = \nabla^2 = \partial^2/\partial\tau_i\partial\tau_j$

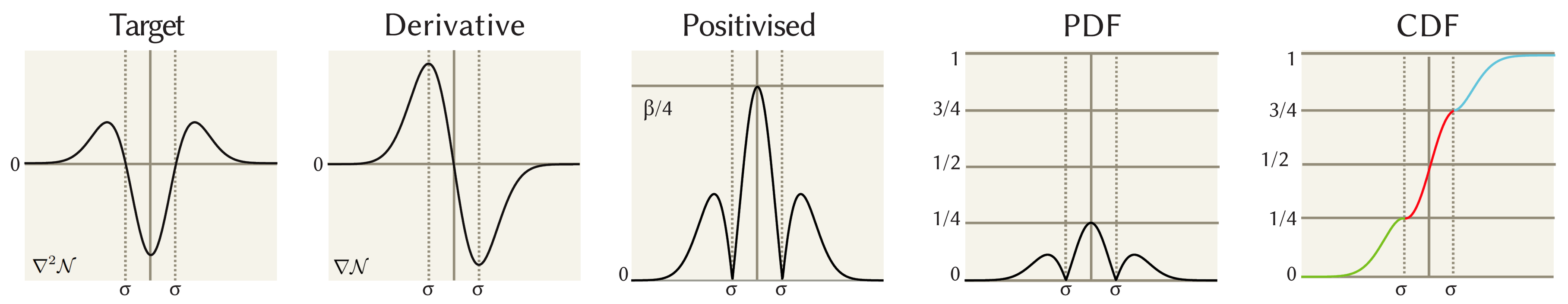
$\mathrm{D}^\mathrm{HVP} = \nabla^2\mathbf{v} = \partial^2/\partial\tau_i\partial\boldsymbol{\tau}\cdot\partial/\partial\boldsymbol{\tau}.$

$$\mathrm{D}^\mathrm{G}\,\kappa(\boldsymbol{\tau}) = -\frac{\tau_i}{\sigma^2}\cdot\mathcal{N}(\boldsymbol{\tau},\sigma).$$

$$\mathrm{D}^\mathrm{HVP}\,\kappa(\boldsymbol{\tau}) = \frac{\mathrm{D}^\mathrm{G}\,\kappa(\boldsymbol{\tau}-\varepsilon\mathbf{v}) - \mathrm{D}^\mathrm{G}\,\kappa(\boldsymbol{\tau}+\varepsilon\mathbf{v})}{2\varepsilon}.$$

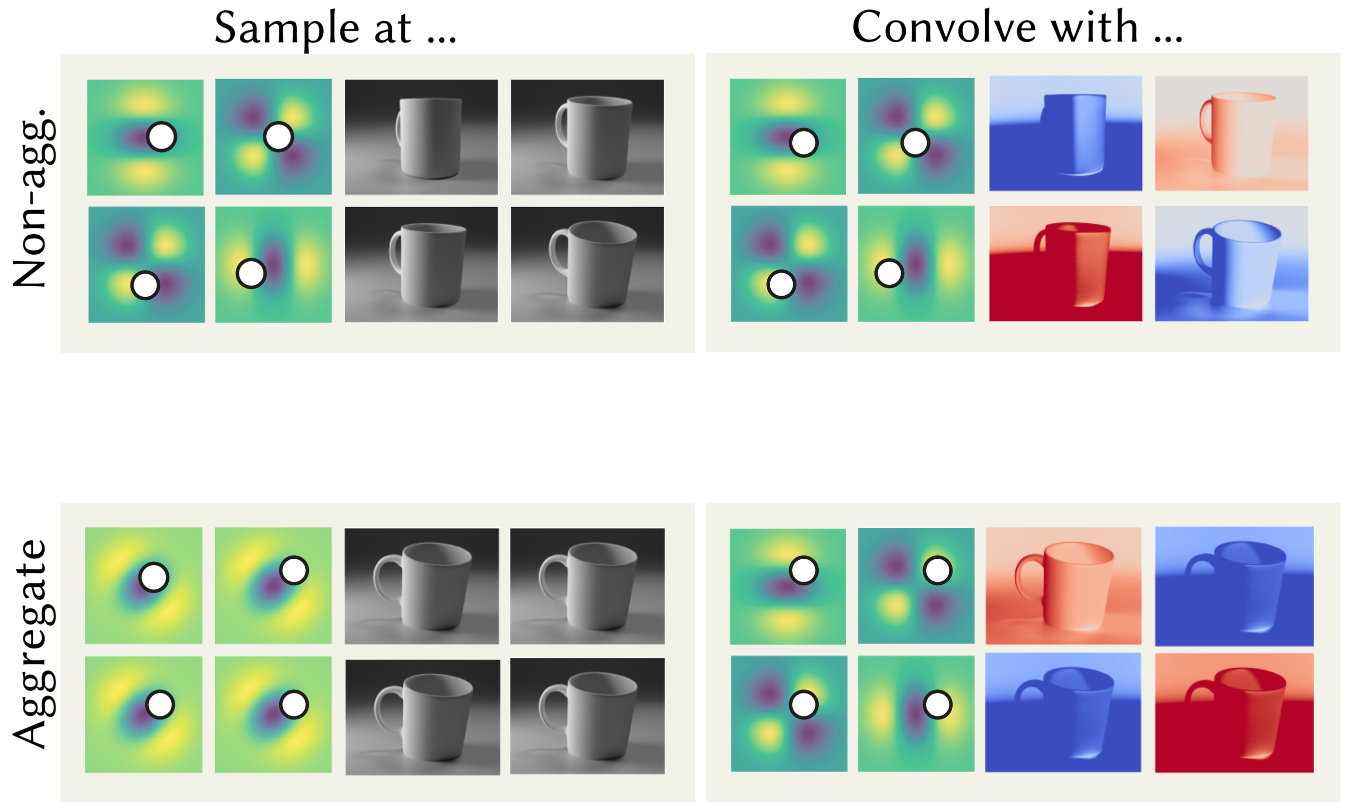
$$\mathrm{D}^\mathrm{H}\,\kappa_{i,j}(\boldsymbol{\tau}) = \begin{cases} \left(-\frac{1}{\sigma^2}+\frac{\tau_i^2}{\sigma^4}\right)\cdot\mathcal{N}(\boldsymbol{\tau},\sigma) & \texttt{if } i=j, \\ \frac{\tau_i\tau_j}{\sigma^4}\cdot\mathcal{N}(\boldsymbol{\tau},\sigma) & \texttt{else.} \end{cases}$$

Positivisation and normalisation



Aggregate Sampling

Problem:
All dimensions need to be sampled:
Costly for Hessian (n^2)
Solution:
Get average convolution, then sample.



Results