Differentiable Monte Carlo Ray Tracing through Edge Sampling,

SIGGRAPH Asia 2018 [1]

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen

Presenter: Doyeon Kim

Agenda

1. Differentiable Rendering
2. A Pixel Example
3. Mathematical Formulation
4. Results

Differentiable Rendering

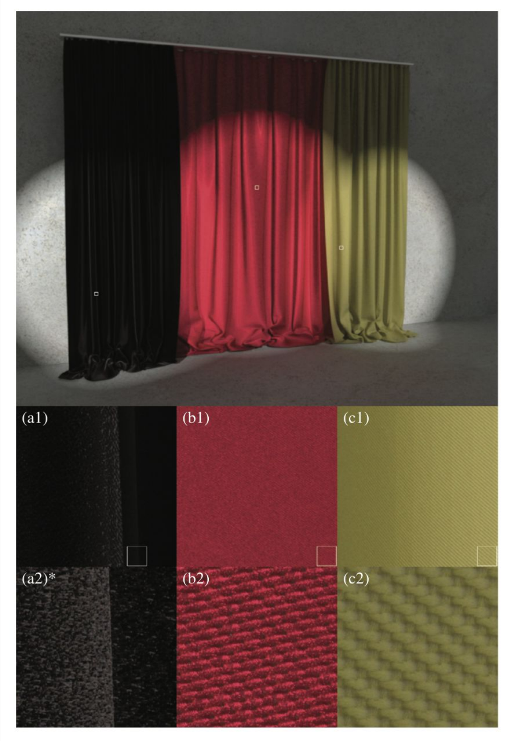
- Allows the gradients of 3D objects to be calculated and propagated through images [2]
- Crucial to optimization, inverse problem, and deep learning
- Gradient w.r.t camera parameters, light sources, scene geometry, material appearance

Why is it challenging?

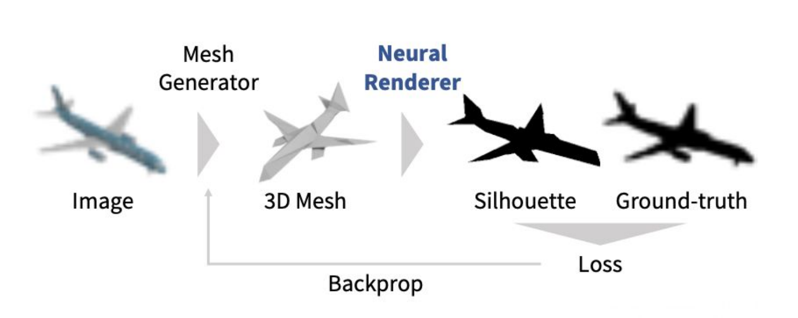
$$L_r(x \to \Theta) = \int_A L(y \to -\Psi) f_r(x, \Psi \to \Theta) \frac{cos\theta_x cos\theta_y}{r_{xy}^2} \boxed{V(x, y)} dA$$

Discontinuous!

Previous works



1. Specialized solver for fabric [3]



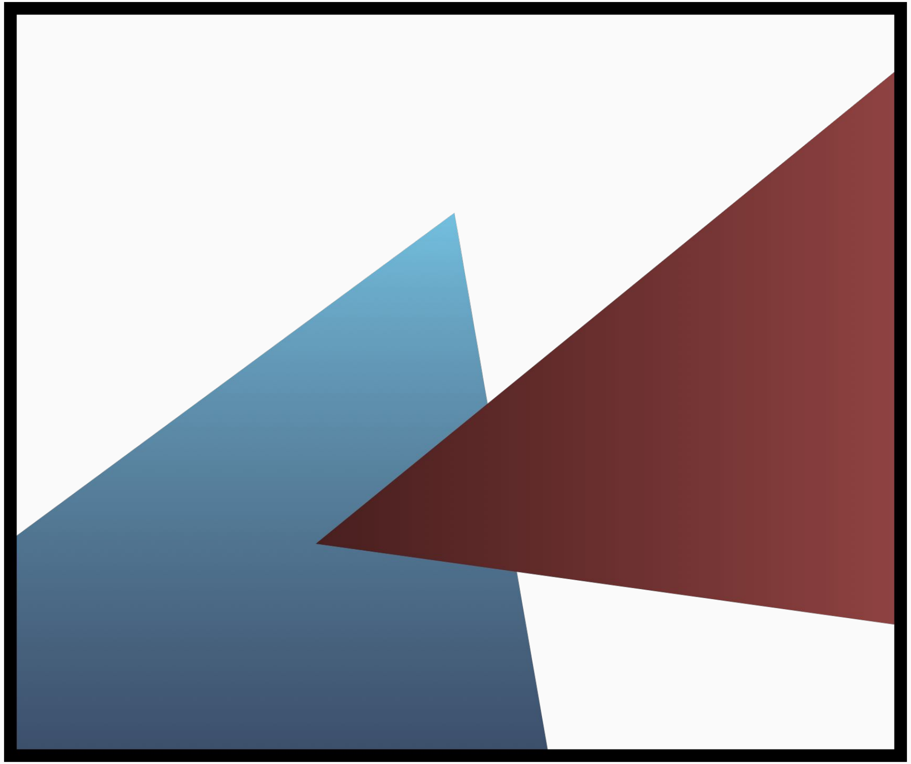
2. General renderer with limitations [4]

Contributions

- The first comprehensive solution to derivatives w.r.t arbitrary scene parameters
- Unbiased gradients and supports arbitrary materials, shadow, and global illumination.

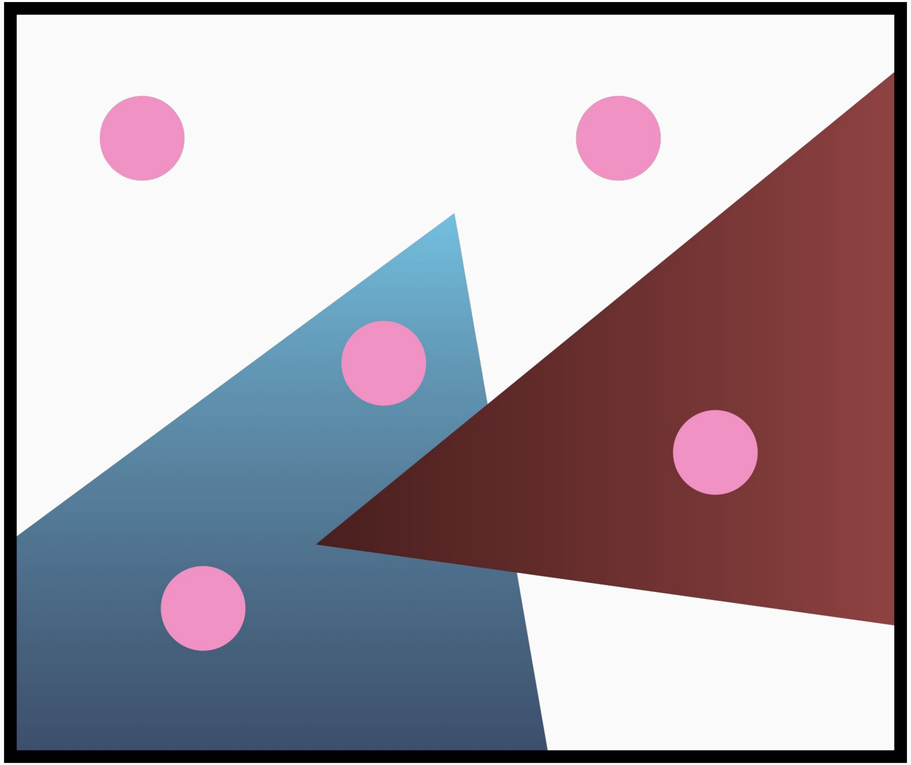
Idea - A Pixel Example

A Pixel Example



From Tzu-Mao Li's slides

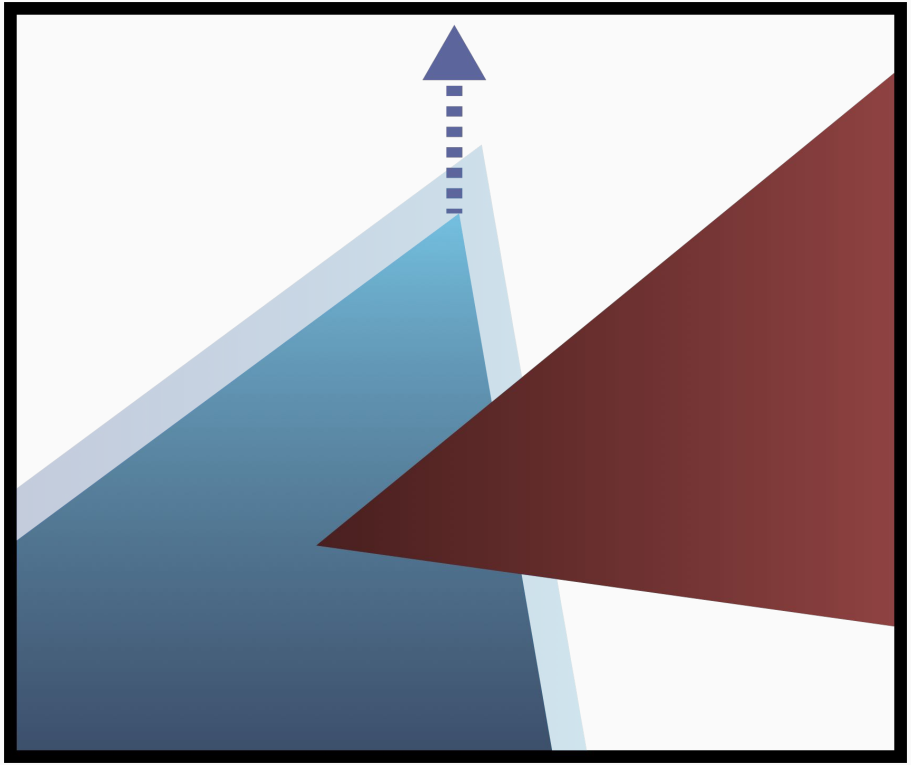
A Pixel Example



From Tzu-Mao Li's slides

A Pixel Example

Q. Color change when blue triangle moves up?

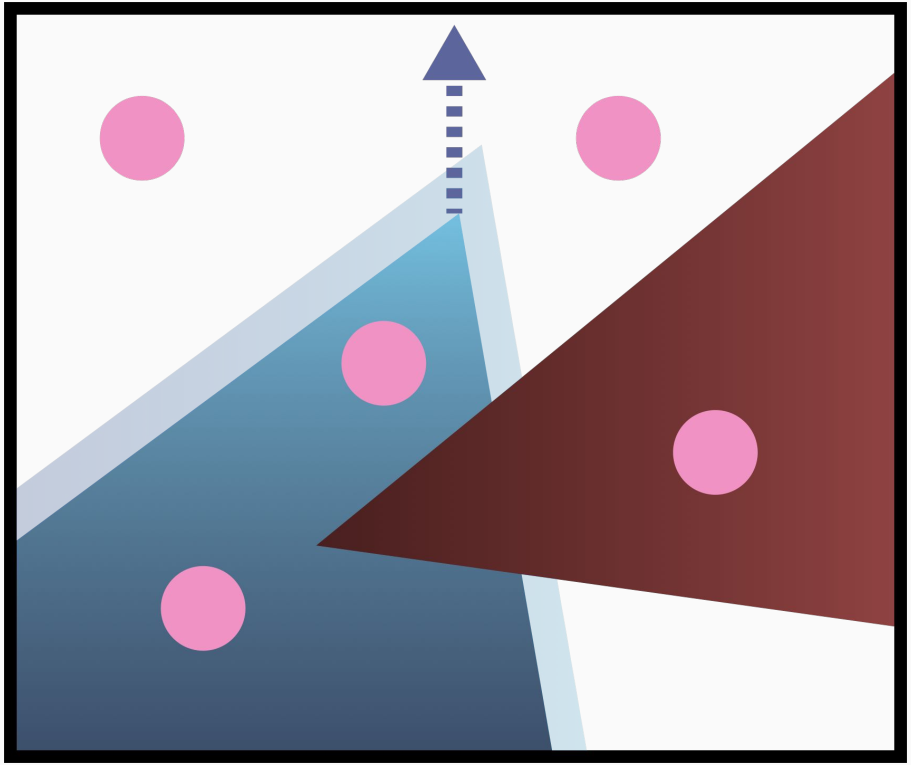


A. More blue less white

From Tzu-Mao Li's slides

A Pixel Example

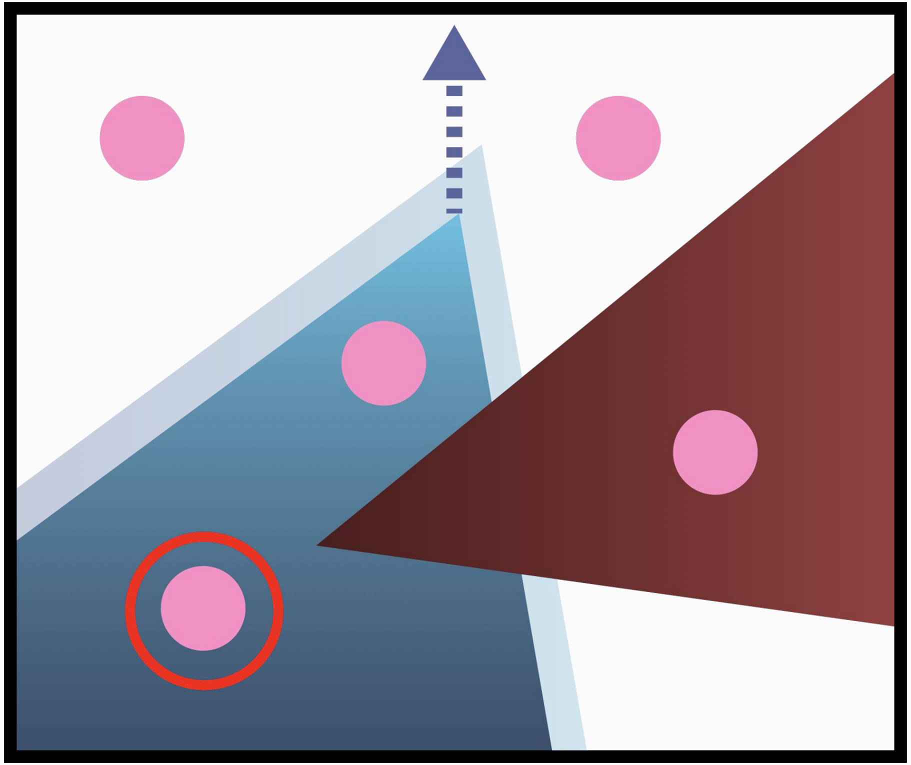
Q. Color change when blue triangle moves up?



From Tzu-Mao Li's slides

A Pixel Example

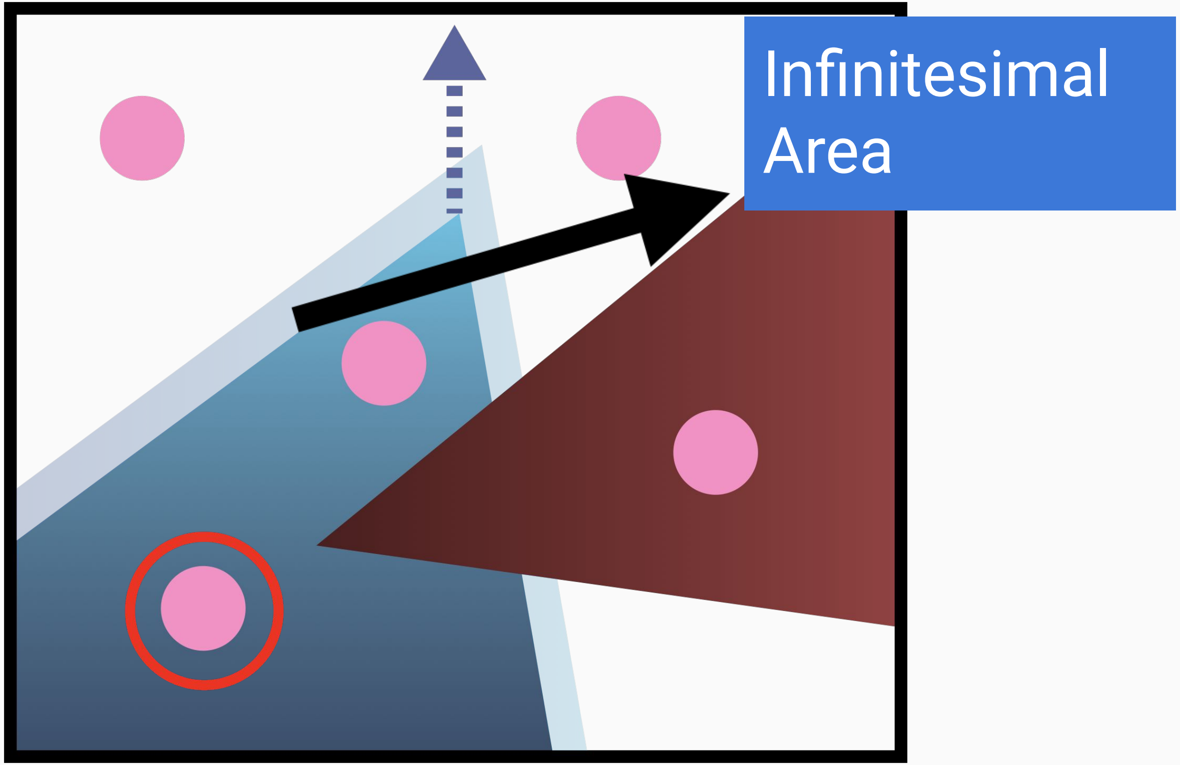
Q. Color change when blue triangle moves up?



From Tzu-Mao Li's slides

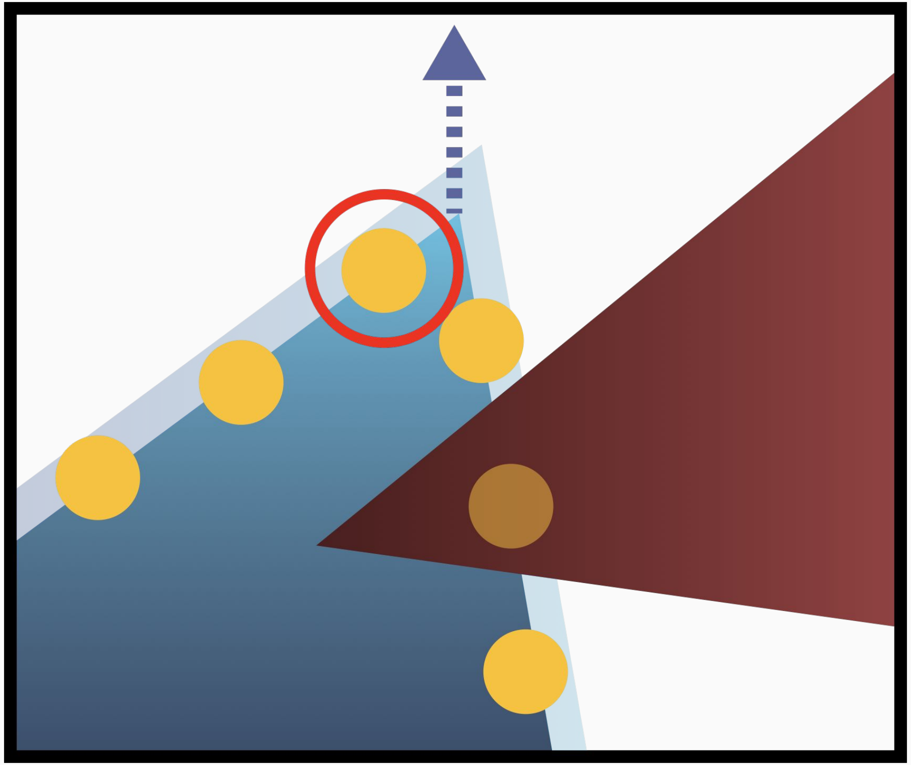
A Pixel Example

Q. Color change when blue triangle moves up?



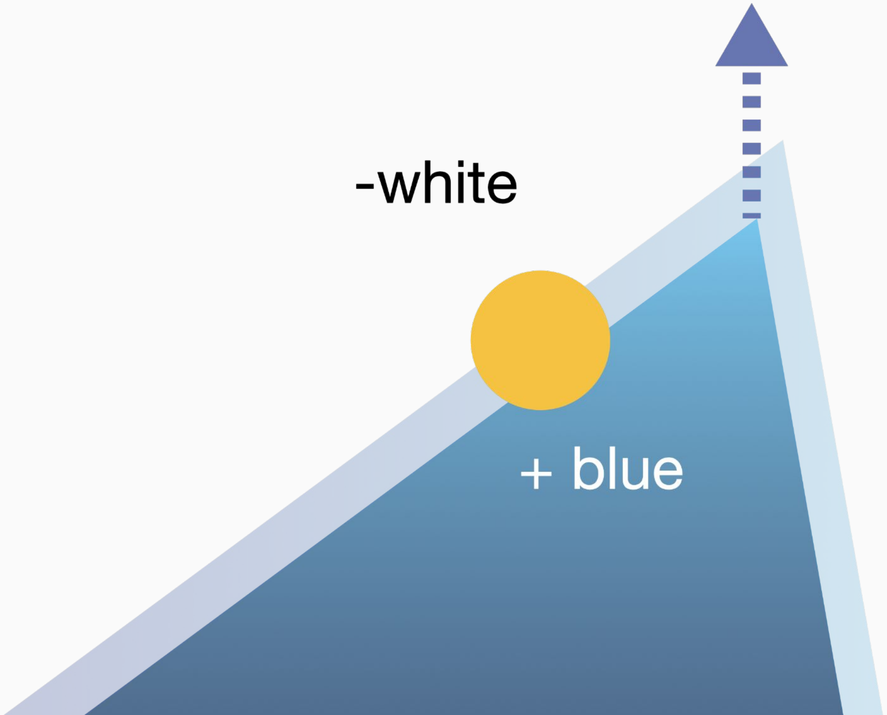
From Tzu-Mao Li's slides

A Pixel Example



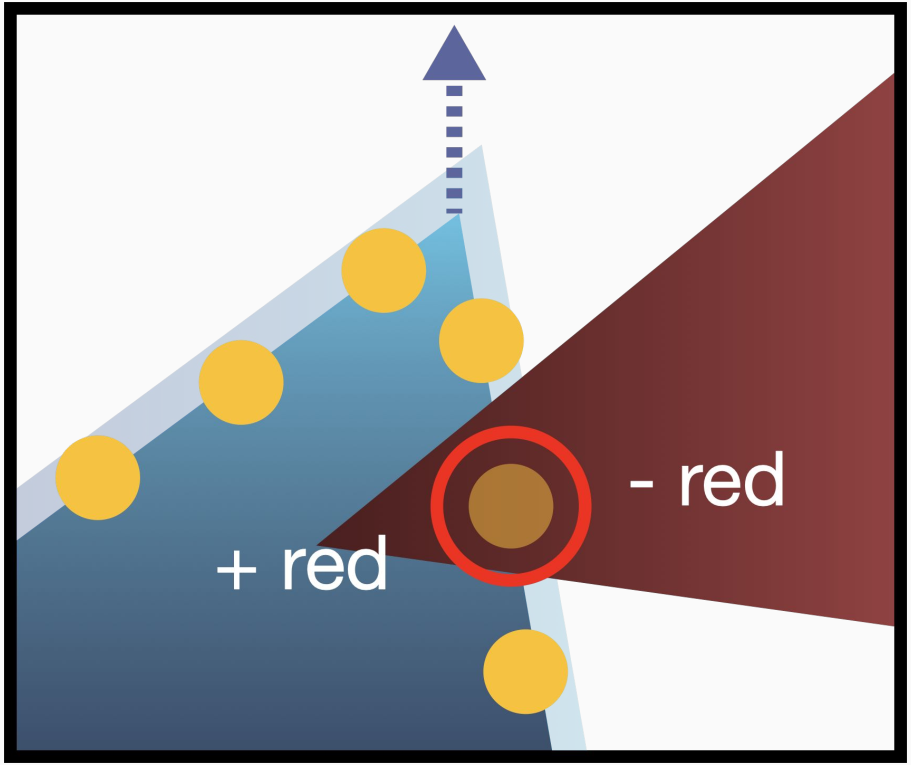
From Tzu-Mao Li's slides

A Pixel Example



From Tzu-Mao Li's slides

A Pixel Example



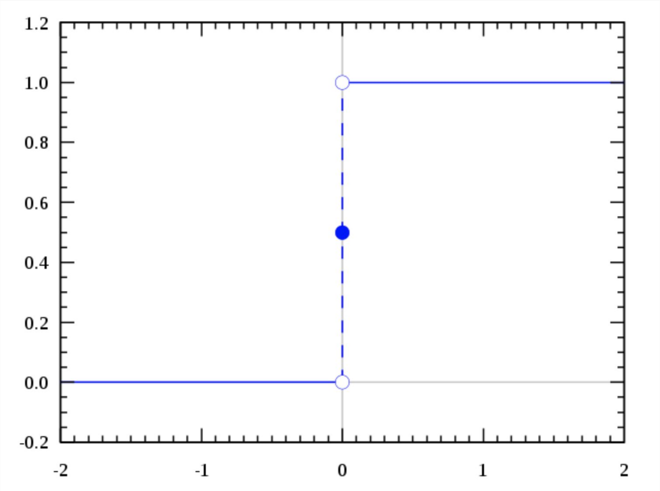
From Tzu-Mao Li's slides

A Pixel Example

- Both primary and secondary visibility can be modeled using this idea.
- Correctly models the global illumination effects.

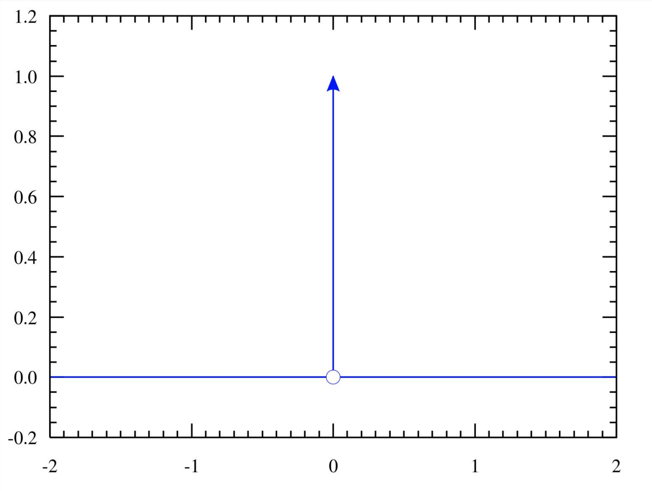
Mathematical Formulation

Heaviside Step Function



$\theta(x)$: Heaviside step function [5]

$$\frac{d}{dx}$$



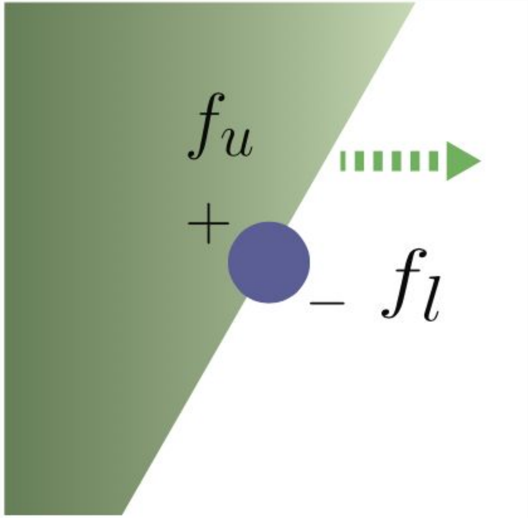
$\delta(x)$: Dirac delta function [6]

Mathematical Formulation

f : Product of radiance and visibility term

Intensity of a pixel: $$I = \int \int f(x, y) dx dy$$

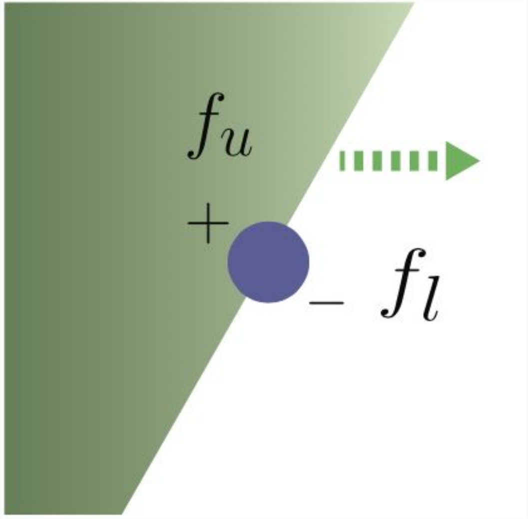
Mathematical Formulation



$$\alpha(x, y) = Ax + By + C$$

: Equation of an edge

Mathematical Formulation



$$f(x, y)$$

$$\downarrow$$

$$\theta(\alpha(x, y)) f_u(x, y) + \theta(-\alpha(x, y)) f_l(x, y)$$

$$= \begin{cases} f_u(x, y), & if\ \alpha(x, y) > 0 \\ f_l(x, y), & otherwise \end{cases}$$

Mathematical Formulation

- We can generalize the idea to n edges in a pixel

$$I = \int\int f(x,y)dxdy$$

$$= \sum_i \int\int \theta(\alpha_i(x,y))f_i(x,y)dxdy$$

Mathematical Formulation

- We can analytically differentiate each term in the sum.
- Estimate the gradient using two Monte Carlo estimators

$$\nabla \int \int \theta(\alpha_i(x,y)) f_i(x,y) dx dy$$

$$= \int \int \delta(\alpha_i(x,y)) \nabla \alpha_i(x,y) f_i(x,y) dx dy$$

$$+ \int \int \theta(\alpha_i(x,y)) \nabla f_i(x,y) dx dy$$

Gradients on edge samples (yellow sample points)

Gradients on area samples (pink sample points)

Importance sampling the edges

- Only few edges are important for a given viewport.
- Hierarchical sampling for silhouette edges
- Several factors (eg. distance, material response, …)

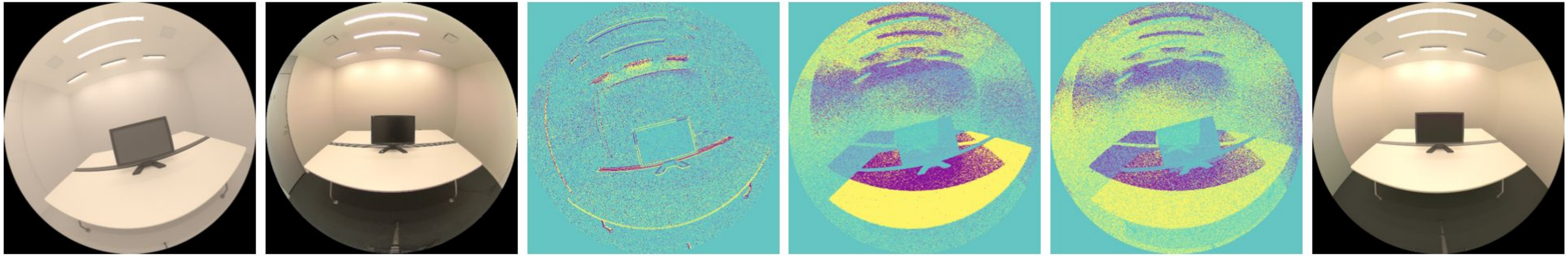
Recap

- Model edge using step function
- Analytical differentiation -> two Monte Carlo estimators
- Importance sampling of edges

Results

Results

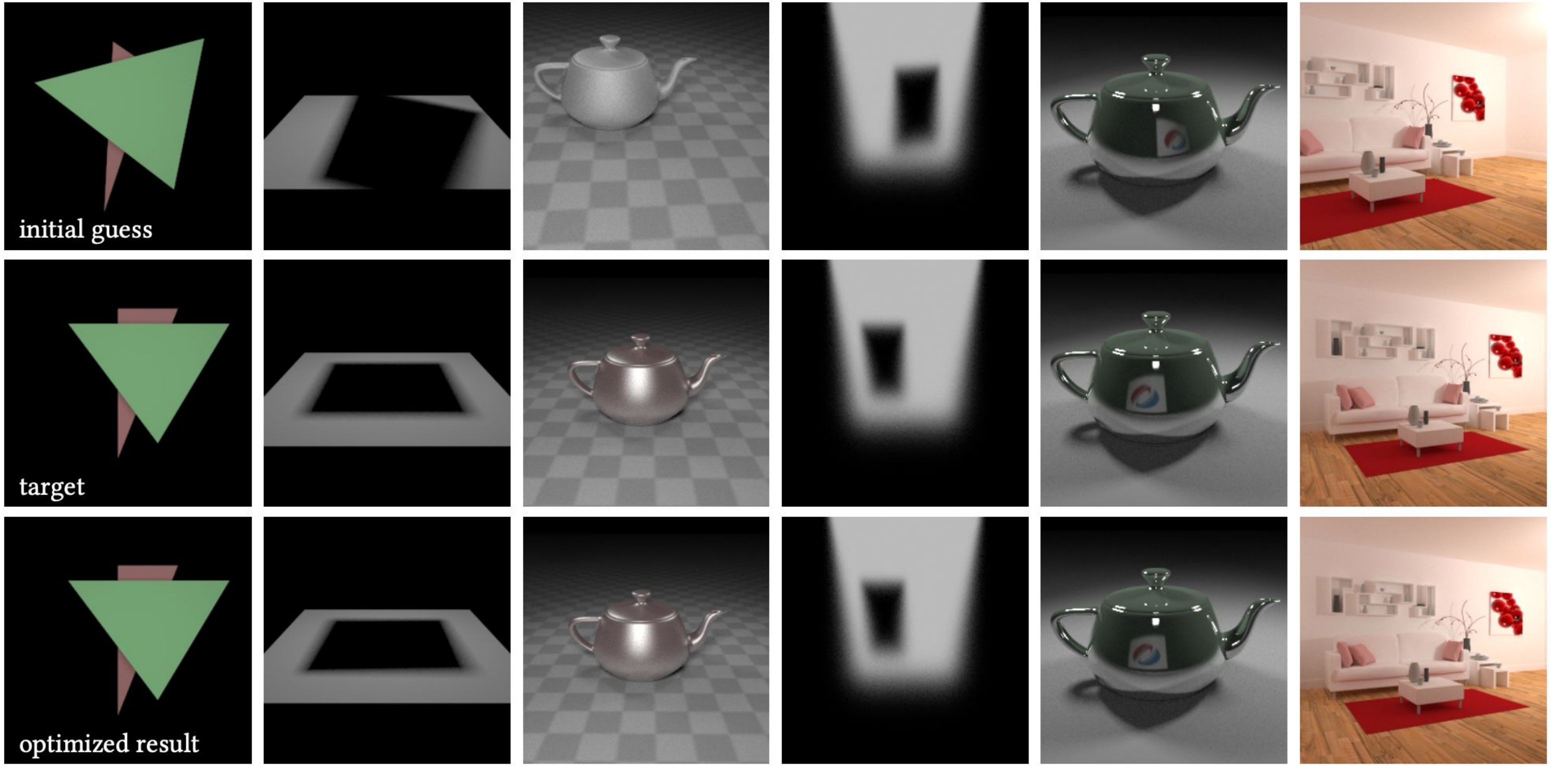
- Gradients w.r.t. various scene parameters



(a) initial guess (b) real photograph (c) camera gradient (per-pixel contribution) (d) table albedo gradient (per-pixel contribution) (e) light gradient (per-pixel contribution) (f) our fitted result

Results

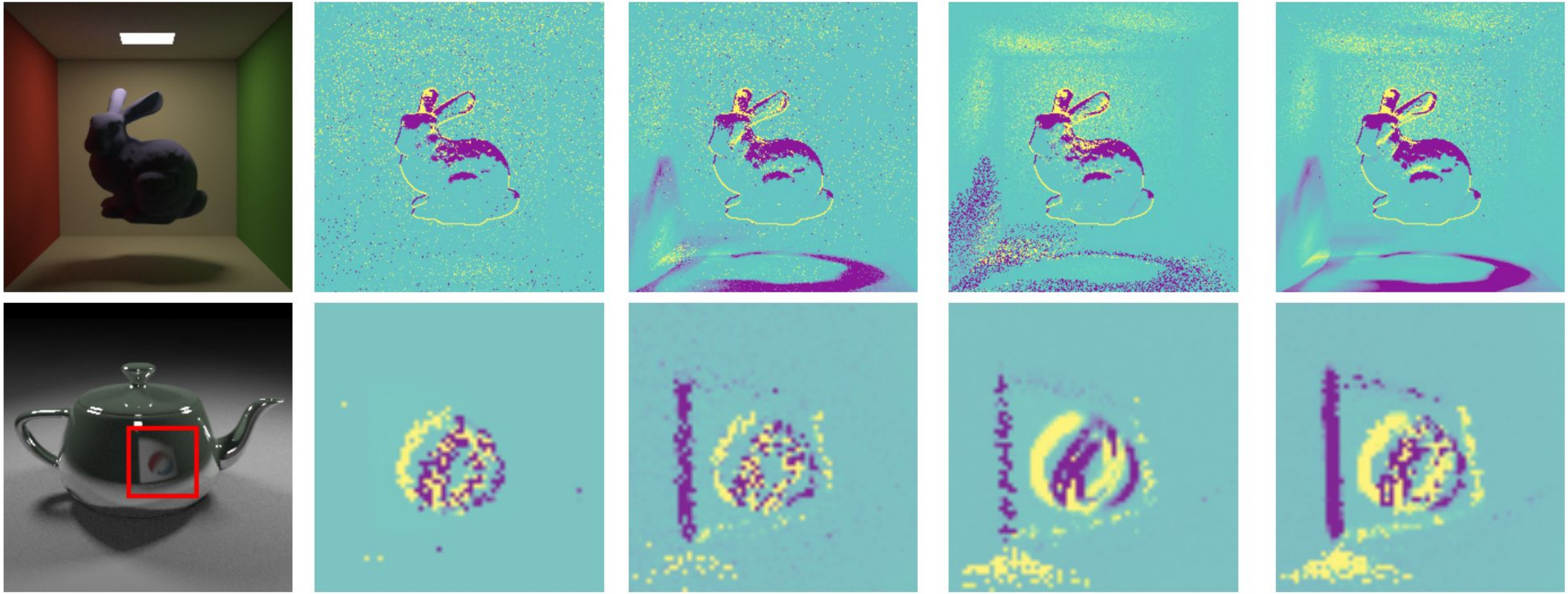
- Various materials and global illumination effects



(a) primary occlusion (b) shadow (c) camera & glossy (d) glossy receiver (e) near-specular (f) global illumination

Results

- Importance sampling of edges reduces variance



scenes | 10s, w/o importance samp. | 10s, w/ importance samp. | 350s, w/o importance samp. | 350s, w/ importance samp.

Contributions

- The first comprehensive solution to compute derivatives of scalar functions over a rendered image w.r.t arbitrary scene parameters
- Unbiased gradients and supports arbitrary materials, shadow, and global illumination.

Limitations

- Performance
- Other light transport phenomena (e.g. motion blur)
- Interpenetrating geometries

References

[1] Li, Tzu-Mao, et al. "Differentiable monte carlo ray tracing through edge sampling." ACM Transactions on Graphics (TOG) 37.6 (2018): 1-11.

[2] Kato, Hiroharu, et al. "Differentiable rendering: A survey." arXiv preprint arXiv:2006.12057 (2020).

[3] Khungurn, Pramook, et al. "Matching Real Fabrics with Micro-Appearance Models." ACM Trans. Graph. 35.1 (2015): 1-1.

[4] Kato, Hiroharu, Yoshitaka Ushiku, and Tatsuya Harada. "Neural 3d mesh renderer." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[5] Wikipedia contributors. "Heaviside step function." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 6 Nov. 2021. Web. 6 Nov. 2021.

[6] Wikipedia contributors. "Dirac delta function." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2 Nov. 2021. Web. 6 Nov. 2021.

Thank you! :)