The pixel color, I, of a scene Φ is,

$$I = \iint f(x, y; \Phi) dxdy \tag{1}$$

where f, the scene function, is the product of the pixel filter, k and the radiance term L, $f(x, y; \Phi) = k(x, y)L(x, y; \Phi)$.

Goal: Compute the gradient with respect to scene parameter, Φ.

$$\boxed{\nabla_\Phi \iint f(x, y; \Phi) dxdy} \tag{2}$$

SML Talk:
Annada Behera, July 11, 2022

T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen, **Differentiable Monte Carlo ray tracing through edge sampling,** ACM Trans. Graph., vol. 37, no. 6, pp. 1–11, Dec. 2018, doi: 10.1145/3272127.3275109.

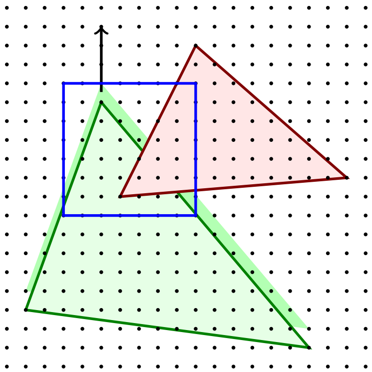
Integrator

Monte-Carlo estimator:

$$F_N = \frac{b-a}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \qquad \mathbb{E}[F_N] = \int dx\, f(x) \tag{3}$$

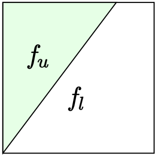
The second part follows from linearity of expectation, provided sampling from $p(X_i)$ is **unbiased.** This discretization is **consistent.**

Uniform sampling



Traditional unbiased uniform area sampling doesn't account for change in area covered.

Triangle discontinuities



The Heaviside function,

$$\mathbb{1}_a(x) = \begin{cases} 1, & x > a \\ 0, & x \leq a \end{cases} \tag{4}$$

It's derivative is the Dirac delta function,

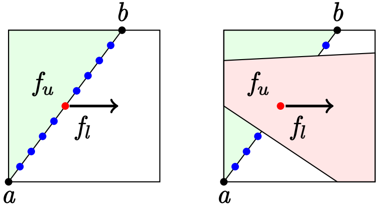
$$\delta(x) = \begin{cases} +\infty, & x = a \\ 0 & x \neq a \end{cases} \quad \text{such that,} \int_{-\infty}^{\infty} dx\, \delta(x) = 1 \tag{5}$$

The discontinuity is then modeled as,

$$\mathbb{1}_\alpha f_u + \mathbb{1}_{-\alpha} f_l, \text{ where, } \alpha = Ax + By + C \tag{6}$$

α is the equation of the edge.

Edge sampling



For two end points a and b, the edge equation is,

$$\alpha(x, y) = (a_y - b_y)x + (b_x - a_x)y + (a_x b_y - b_x a_y) \tag{7}$$

- For the half planes f_u and f_l, if the edge moves right, the green area increases and white decreases.
- Sample a point on the edge and compute the color difference between the half-spaces on the two sides of the edge.
- For occluded points, the color difference is zero. So, no gradient.

Heaviside scene function

The scene function as a summation of Heaviside function with $\{\alpha_i\}$ edges is,

$$\iint f(x, y; \Phi)\, dxdy = \sum_i \iint \mathbb{1}_{\alpha_i} f_i(x, y; \Phi)\, dxdy \tag{8}$$

Then the analytical derivative of the scene function is now,

$$\nabla_\Phi \sum_i \iint \mathbb{1}_{\alpha_i} f_i(x, y; \Phi)\, dxdy \tag{9}$$

$$= \sum_i \iint \nabla_\Phi \mathbb{1}_{\alpha_i} f_i(x, y; \Phi)\, dxdy \tag{10}$$

$$+ \sum_i \iint \mathbb{1}_{\alpha_i} \nabla_\Phi f_i(x, y; \Phi)\, dxdy \tag{11}$$

The easier estimator

The second term in the analytical derivative of the scene function,

$$\iint \sum_i \mathbb{1}_{\alpha_i} \nabla_\Phi f_i(x, y; \Phi)\, dxdy \tag{12}$$

- Bring the summation into the integral.
- The Heaviside function $\mathbb{1}_\alpha$ is zero for almost all triangles.
- For a very few triangles (in the order of tens only), compute the values as before and compute the gradient.
- Use automatic differentiation to compute the gradient.

The hard part

$$
\begin{aligned}
\sum_i \iint \nabla_\Phi \mathbb{1}_{\alpha_i} f_i(x, y; \Phi)\, dxdy &= \sum_i \iint \delta(\alpha_i) \nabla_\Phi \alpha_i f_i(x, y; \Phi)\, dxdy \\
&= \sum_i \int_{\alpha=0} \frac{\nabla_\Phi \alpha_i}{\|\nabla_{xy} \alpha_i\|} f_i(x, y; \Phi)\, d\sigma(x, y)
\end{aligned}
$$

The gradient of the edge equation is,

$$
\|\nabla_{x,y}\alpha\| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} \tag{13}
$$

$$
\frac{\partial \alpha}{\partial a_x} = b_y - y, \quad \frac{\partial \alpha}{\partial a_y} = x - b_y \tag{14}
$$

$$
\frac{\partial \alpha}{\partial b_x} = y - a_y, \quad \frac{\partial \alpha}{\partial b_y} = a_x - x \tag{15}
$$

$$
\frac{\partial \alpha}{\partial x} = a_y - b_y, \quad \frac{\partial \alpha}{\partial y} = b_x - a_x \tag{16}
$$

Bonus: Screen space gradient.

Monte carlo sampling with Dirac integral

For other scene parameters p, (like camera position, vertex color, etc), there is no discontinuities. So, their derivatives are a simple application of chain rule,

$$\frac{\partial \alpha}{\partial \Phi} = \sum_k \frac{\partial \alpha}{\partial a_k}\frac{\partial a_k}{\partial \Phi} + \frac{\partial \alpha}{\partial b_k}\frac{\partial b_k}{\partial \Phi} \tag{17}$$

And the Monte carlo estimator is,

$$\frac{1}{N}\sum_{j=1}^{N} \frac{\|E\|\nabla_{\Phi}\alpha_i(f_u - f_l)}{\Pr[E]\|\nabla_{xy}\alpha_i\|} \tag{18}$$

where, $\|E\|$ length of edge E and $\Pr[E]$ is the probability of selecting edge E. This only for the *silhouette* edges, where the gradient contribution is non-zero.

Sampling an edge and sampling on edge

- There are millions of triangles in the scene.
- Sample an edge which contributes to the gradient, i.e, *silhouette* edge.
- Project all the edges to the screen space (in the pre-processing step), select the silhouette and visible edges and discard the other edges.
- The selected edges are proportional to the length on the screen space. (importance sampling)
- Uniformly select a point on the selected edge.

Limitations

- Secondary visibility is not considered.
- Computationally expensive for larger samples.
- Other light transport phenomenon are not considered.
- Not suitable for dynamic scenes.
- Shader discontinuities are not considered.