Physics based rendering differentiably [CVPR 2021 Tutorial]

1) forward rendering function : $I = R(\theta)$

$\theta \in \mathbb{R}^n$ (n: num of param)

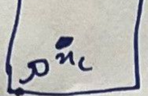
$I \in \mathbb{R}^m$ (m: num of pixel)

1) Grad mat $\frac{dR}{d\theta}(x) \in \mathbb{R}^{m \times n}$

1) Pixel level antialiasing matters so it gives you a continuous texture instead of a binary one which causes issues with differentiability.

Pixel val $= I(x_c)$

binary



$$\frac{1}{|P|}\int_P I(x)\,dx$$

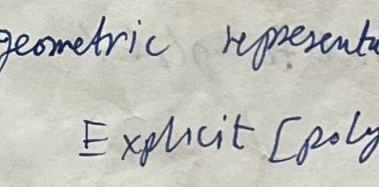
Perfect antialiasing

1) geometric representations

Explicit [polygonal meshes]   Implicit [SDF]

1) because initial guess for grad descent is important considering how complicated BRDF, illumination are, usually before using diff rendering we use D.L to initialise those variables and then allow for diff rendering to work

1)



$$I = \underbrace{\int_{H^2}}_{\text{unit hemisphere}} \underbrace{\overbrace{f_r(w_i, w_o)}^{\text{BRDF}}\ \overbrace{L_i(w_i)}^{\text{Incident radiance}}\ \overbrace{(n \cdot w_i)}^{\text{shading wrt normal } n}\ d\sigma(w_i)}_{r}$$

sample random directions w_i from pdf $p(w_i)$

$$I \approx \frac{r^{s}}{p(w_i^s)}$$

$s \rightarrow$ sampled

Monte Carlo

1) Differential radiance x

$$\frac{dI}{d\pi} = \frac{d}{d\pi}\int_{H^2} fr(w_i, w_o)\, Li(w_i)\,(n\cdot w_i)\, d\sigma(w_i)$$

π: Local para

1) BRDF
2) shading normal
3) illumination brightness

They don't introduce discontinues - simple form of leibniz rule applying

then

$$\int_{H^2} \frac{d}{d\pi} \text{———}$$

π: global para

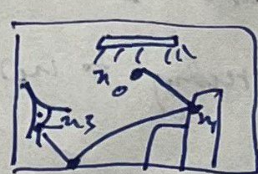
1) shape and pose of different scene elements [camera, sources, objects]

we cannot use the simple leibniz rule for global para

☒ 1:09:00 — 1:15:00 discontinuities in type Imp

1) Path Integral for Global illumination [Alternate to Rendering Eqn]

$$I = \underset{\text{Path space}}{\int_{\Omega}} \overbrace{f(\bar{x})}^{\text{Measurement contrib}}\, \underbrace{d\mu(\bar{x})}_{\text{area product}}$$



light path $\bar{x} = (x_0, x_1, x_2, x_3)$

.) Captures surface as well as volumetric scattering

Monte Carlo $\langle I \rangle = \dfrac{f(\bar{x})}{p(\bar{x})}$ ⟶ How to design a low variance sampling

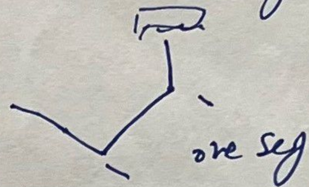
$p(\bar{x})$: probability density for sampling path $\bar{x}$

1) Path space differentiable rendering

$$\frac{d}{d\theta}\left(\int_{\Omega} f(\bar{x})\, d\mu(\bar{x})\right) = \underbrace{\int_{\Omega} \dot{f}(\bar{x})\, d\mu(\bar{x})}_{\text{Interior Integral}} + \underbrace{\int_{\partial\Omega} g(\bar{x})\, d\mu'(\bar{x})}_{\text{Boundary Integral}}$$

$\dot{f} = \frac{\partial f}{\partial \theta}$ => can be done with A.D but problem being
  1) e.g 10^6 path integrals [per pixel]
  1) Can be 10^6 parameters

1) Inorder to estimate correct gradients we need to correctly Calculate "Boundary Integral" [boundary light path + One boundary segment]



1) sharp edges = shape edges
1) Silhouette edges = view dependent [occluding edges]

1) Edge sampling for rendering boundaries [li et.]
  1) trace main paths to estimate interior integral [same as forward path tracing]
  2) Trace additional side paths for boundary integral, where each side path begins with a boundary segment obtained with edge sampling

1) ~~?~~ study [Zhang et al 201] path space