Differentiable Rendering

Visgraf Vision and Graphics Laboratory

3D Graphics Systems | AI Graphics - Theory and Practice

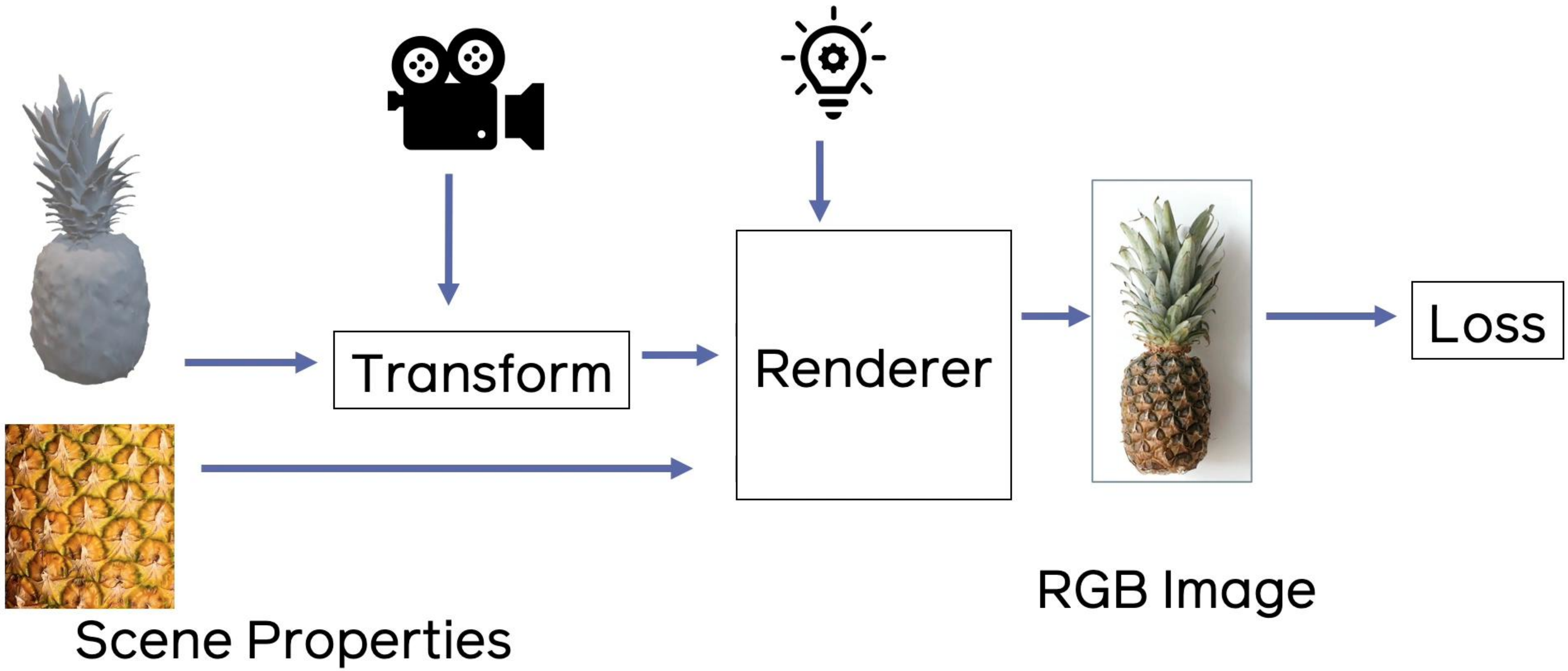
Hallison Paz

IMPA, June 7th, 2023

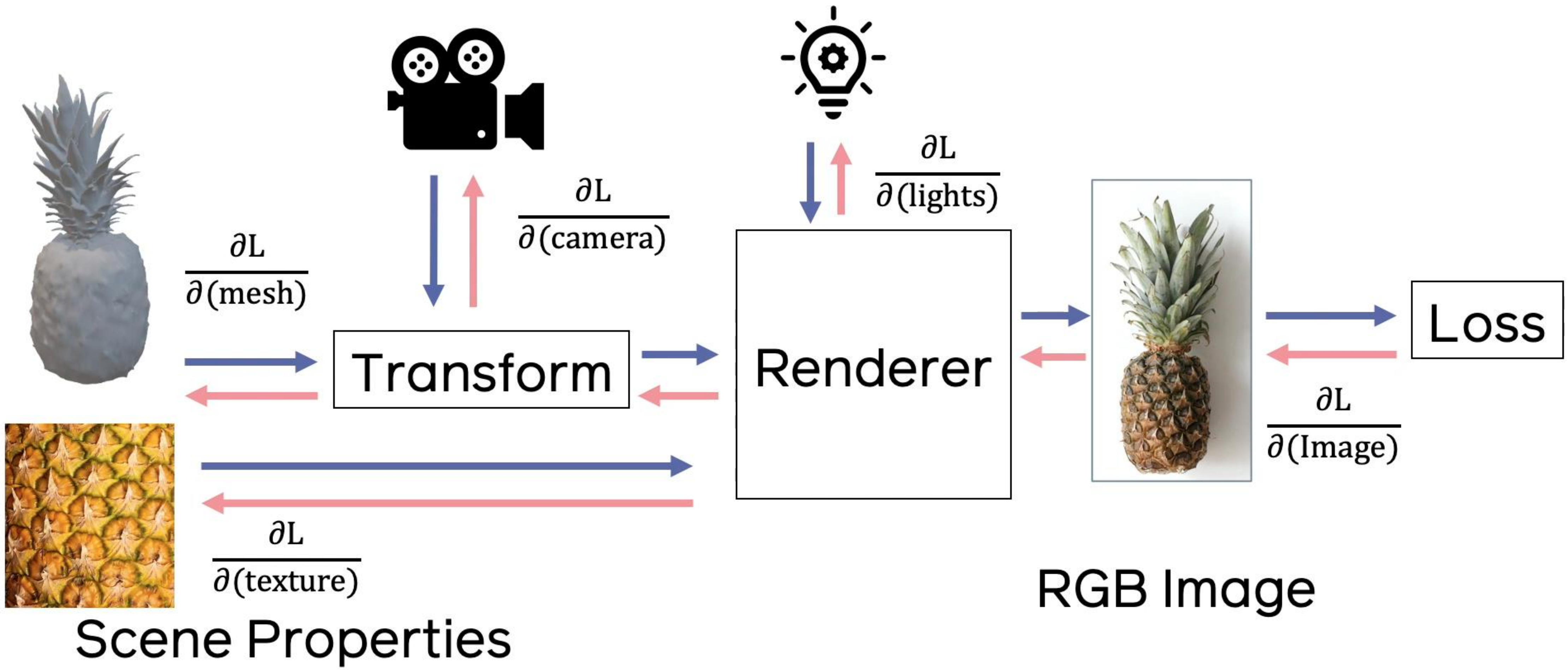
Why Differentiable Rendering?

- Relate 2D pixels to 3D properties
- Image based 3D reasoning

Differentiable Rendering



Differentiable Rendering

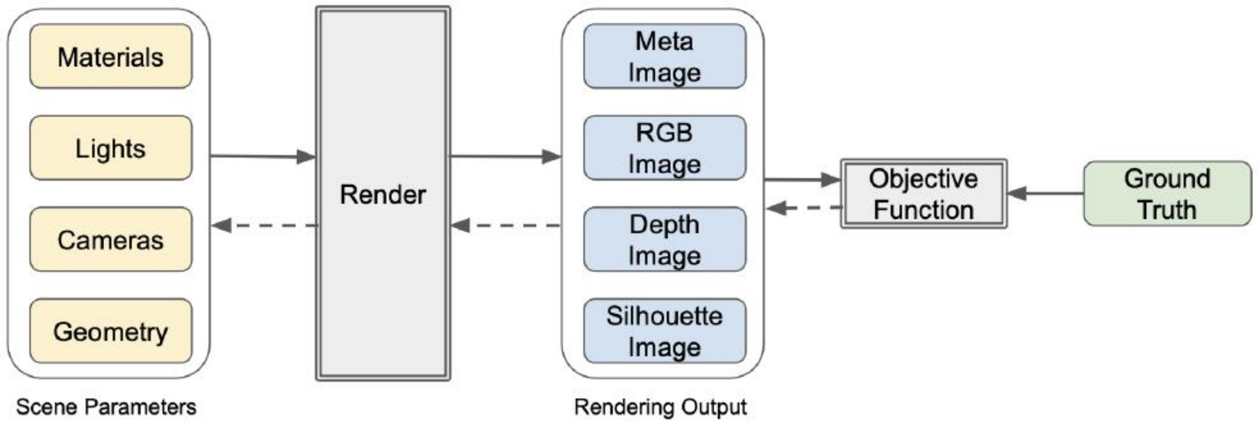


Bridging the Gap: Graphics <> Vision

Scene parameters: θ

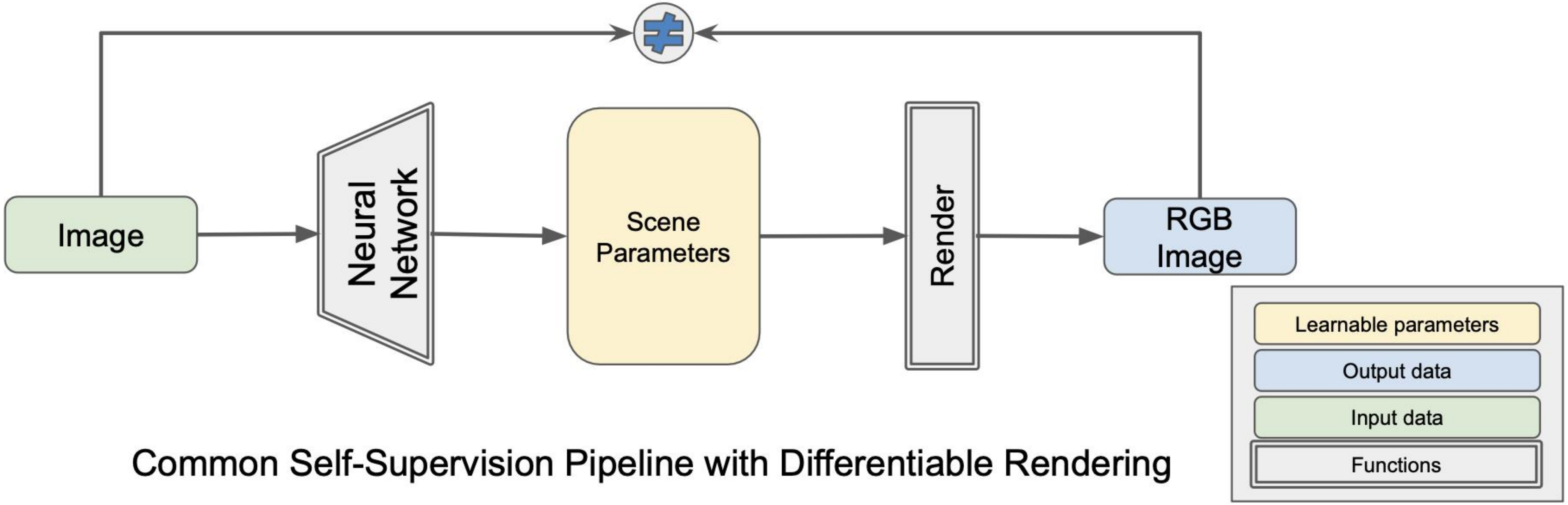
$$f(\theta) = I$$

$$E(\theta) = ||f(\theta) - I||^2$$



Optimization using a Differentiable Renderer

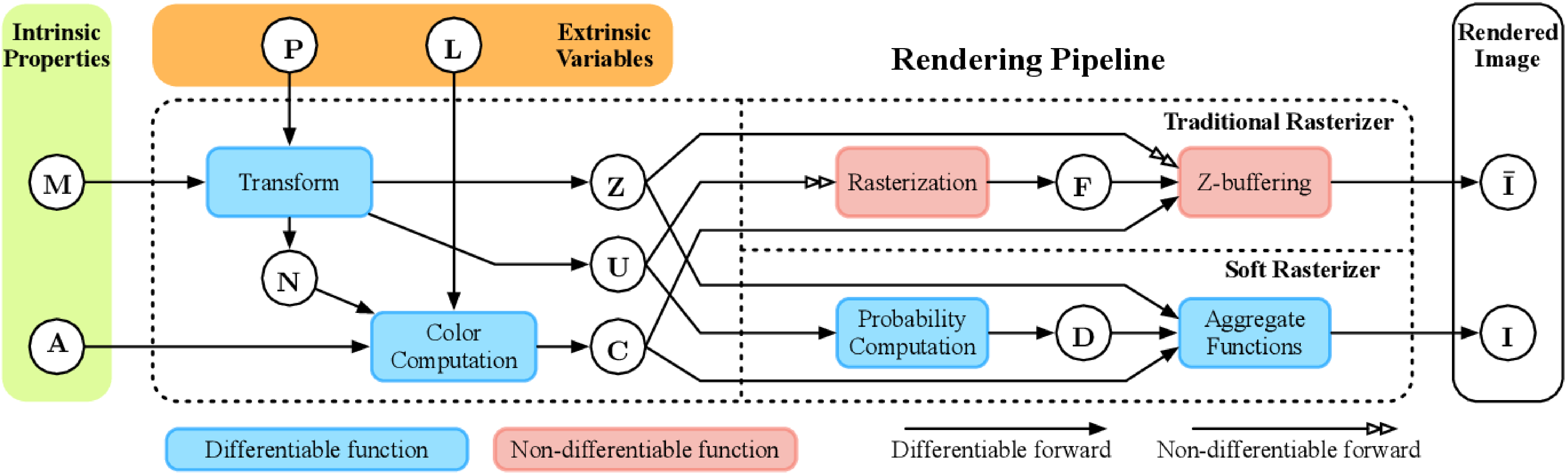
Self supervised learning



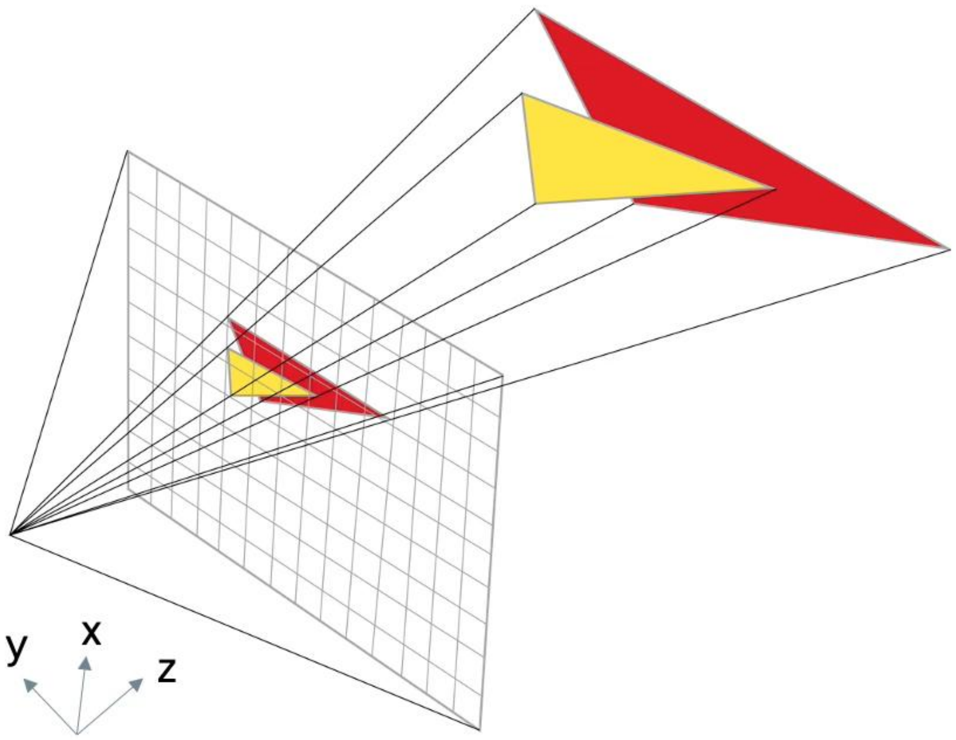
Common Self-Supervision Pipeline with Differentiable Rendering

Challenges

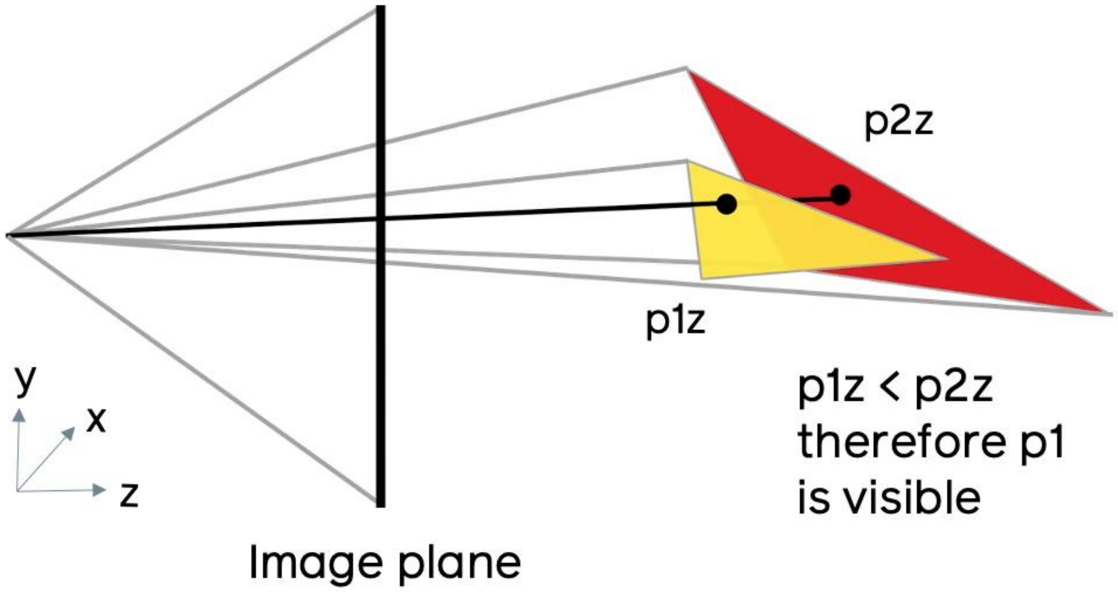
Rasterization Pipeline



Rendering = rasterization + shading



Rasterization

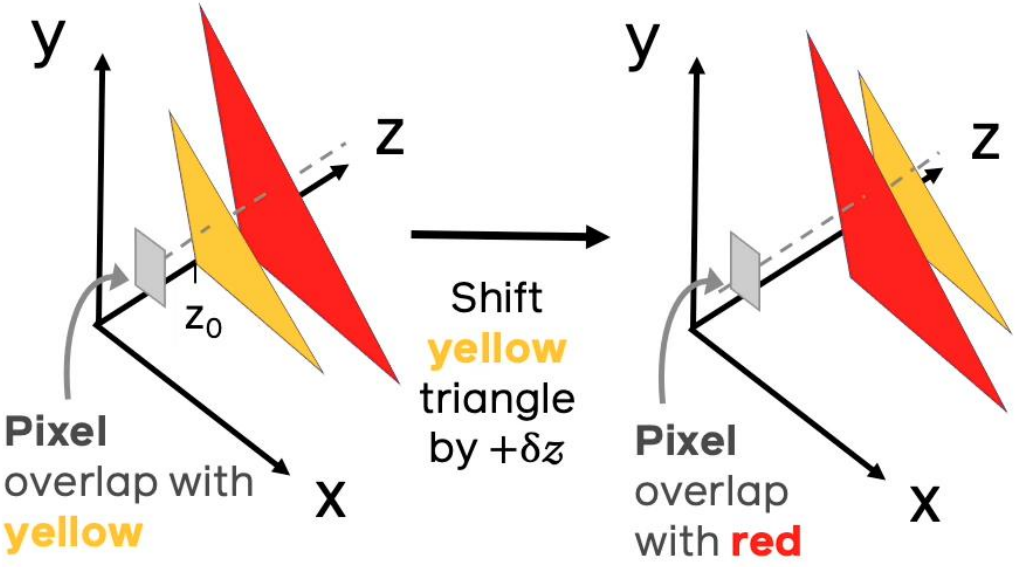


Z buffering

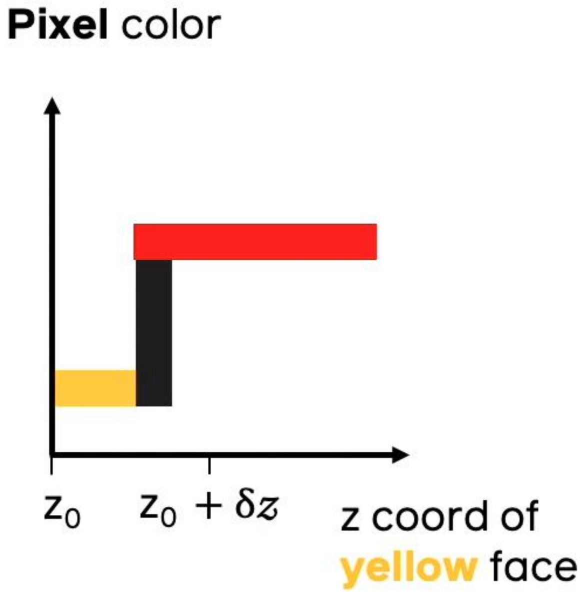
96

Traditional rasterization problem 1

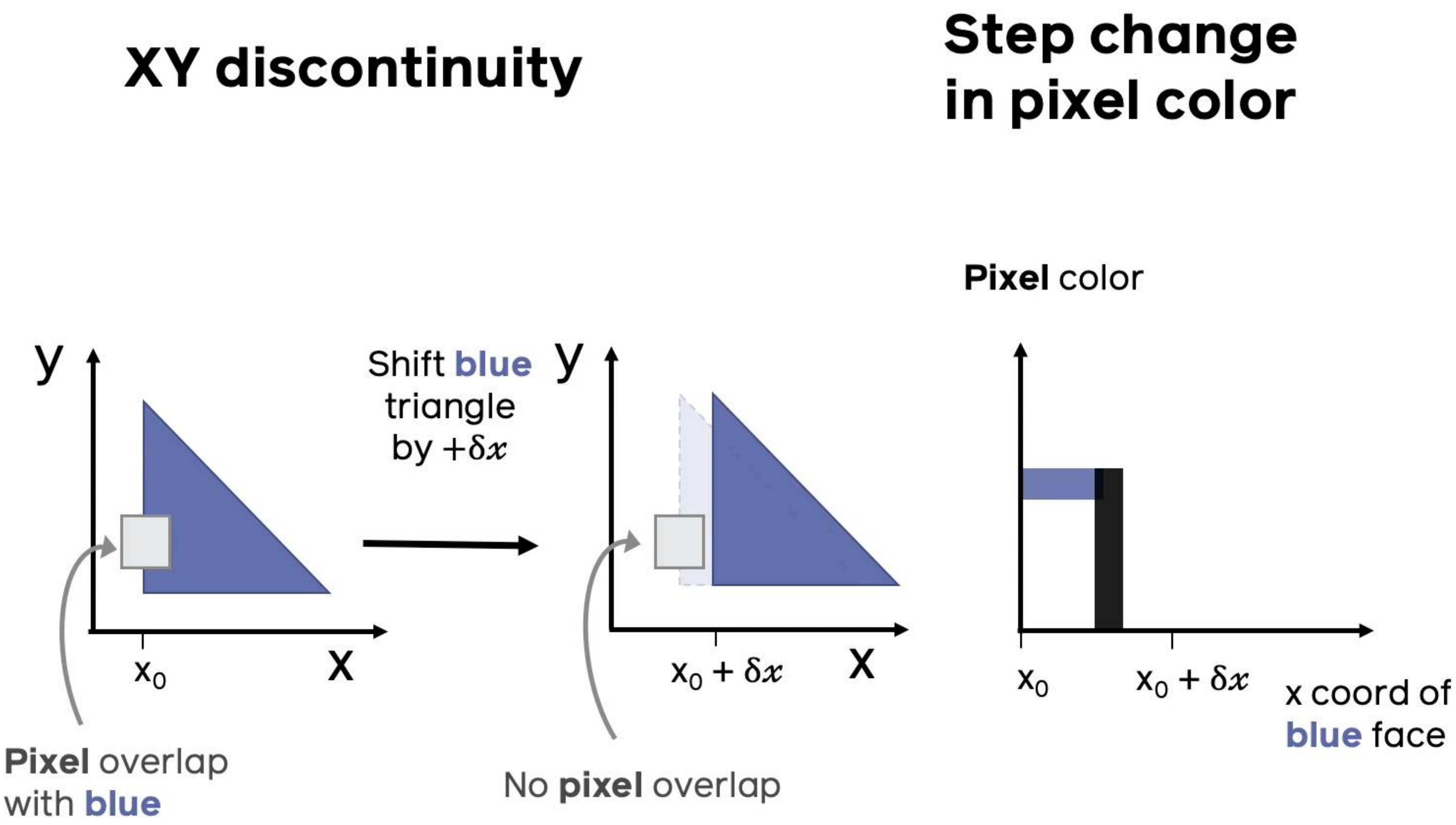
Z discontinuity



Step change in pixel color



Traditional rasterization problem 2



119

Commom approaches

Approximated gradients

- Open DR
- Neural 3D Mesh renderer

Approximated rendering

- Soft Rasterizer

for a more complete list, see: Kato, Hiroharu & Beker, Deniz & Morariu, Mihai & Ando, Takahiro & Matsuoka, Toru & Kehl, Wadim & Gaidon, Adrien. (2020). Differentiable Rendering: A Survey.

Open DR

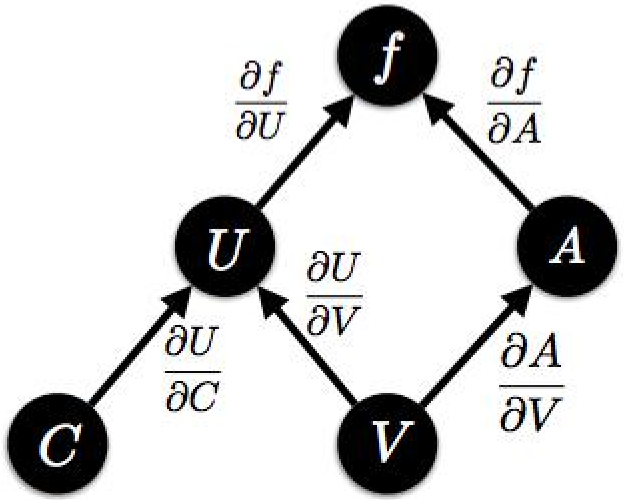
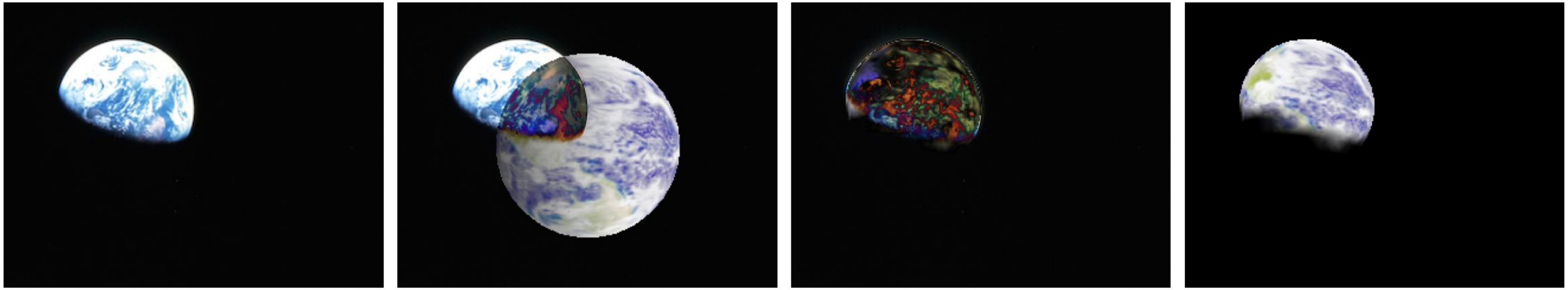


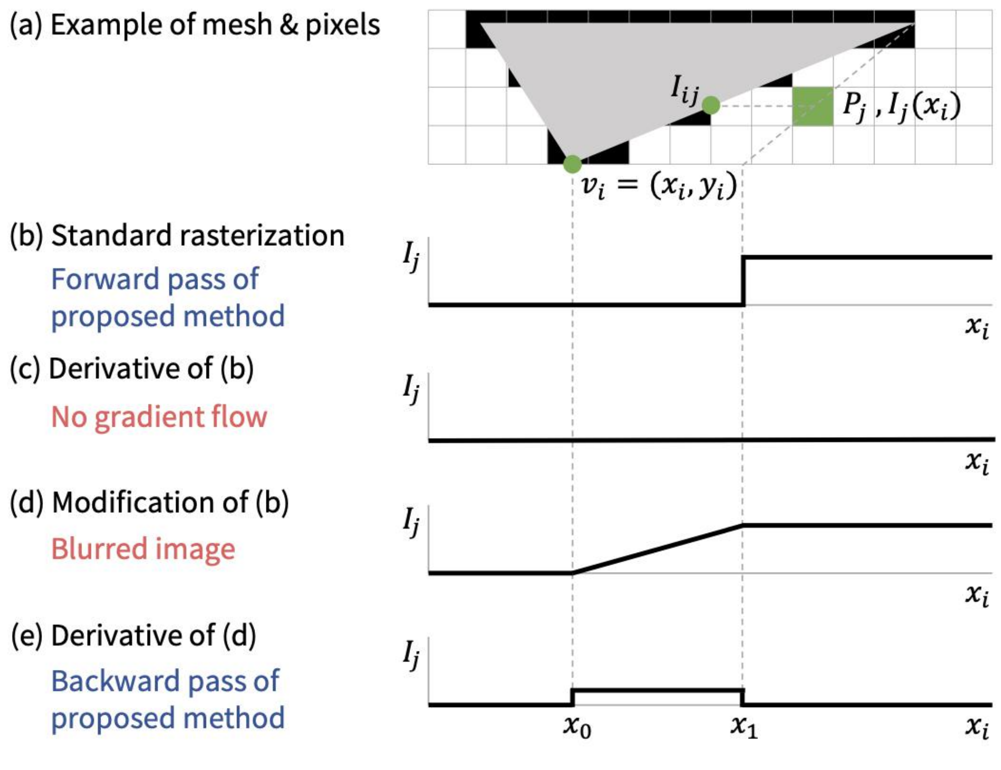
Fig. 1. Partial derivative structure of the renderer.

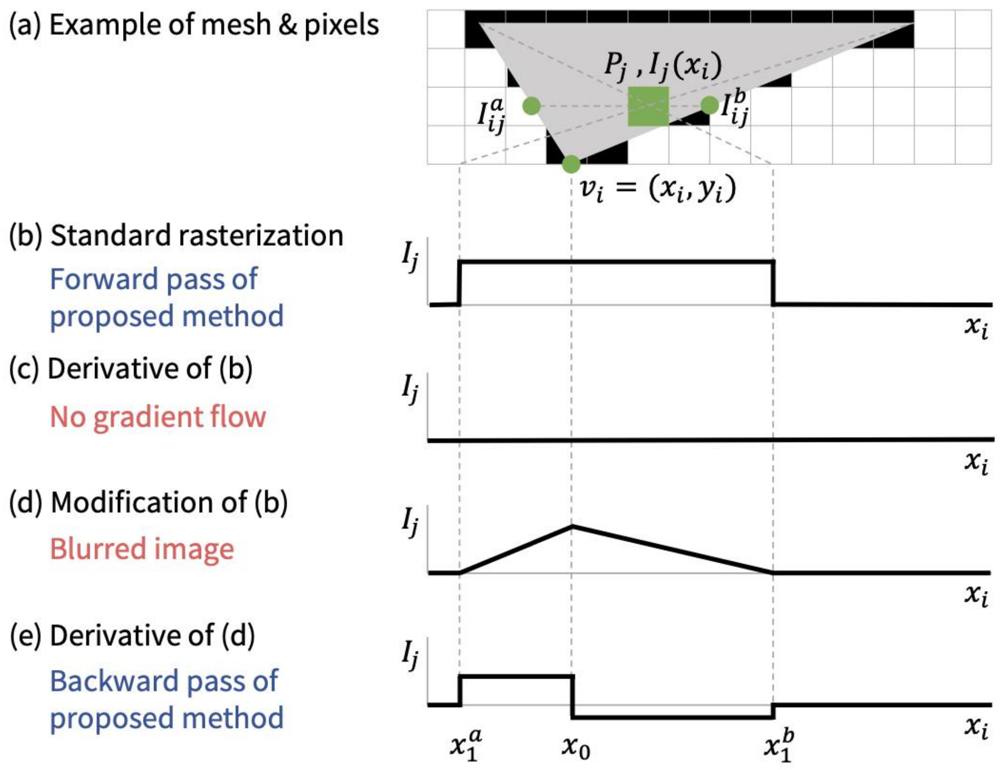
Open DR

```
# Minimize the energy
light_parms = A.components
ch.minimize(E, x0=[translation])
ch.minimize(E, x0=[translation, rotation, light_parms])
```



Neural 3D Mesh Renderer





Soft Rasterizer

Approximated rendering

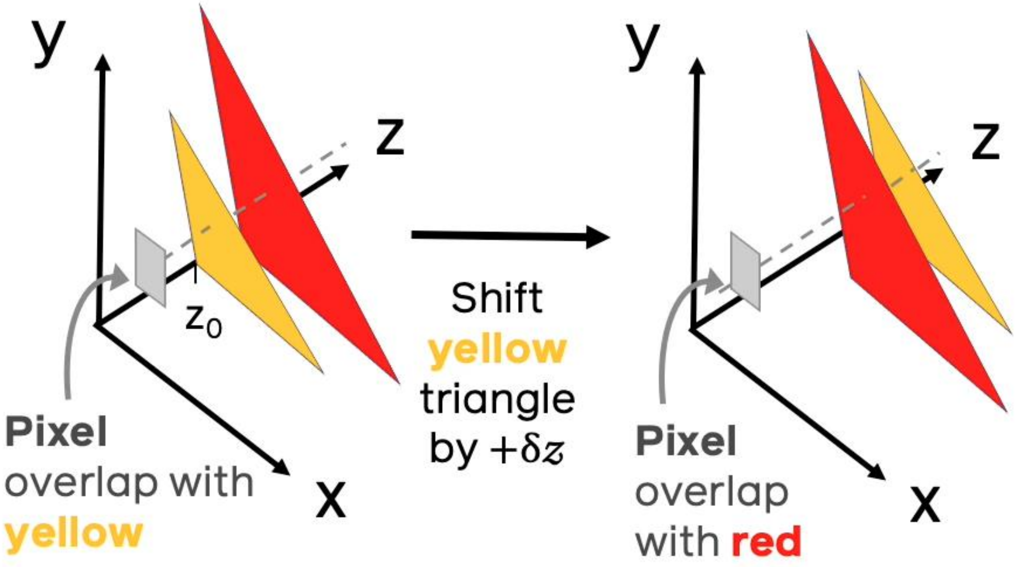
Standard rendering

Rendered w/ larger γ

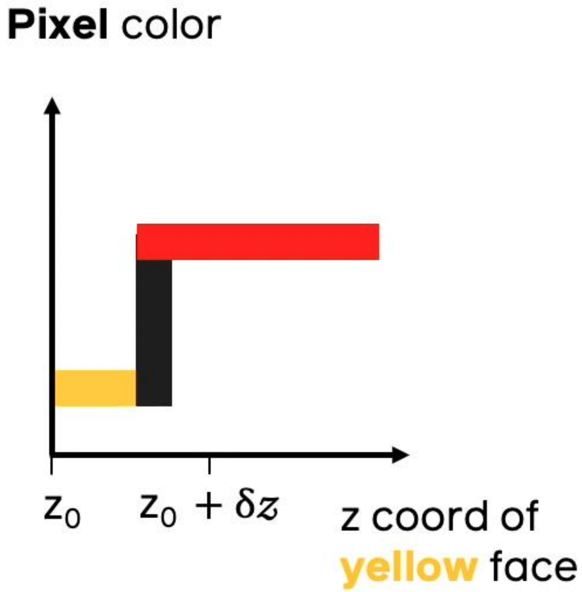
Rendered w/ larger γ and σ

Traditional rasterization problem 1

Z discontinuity

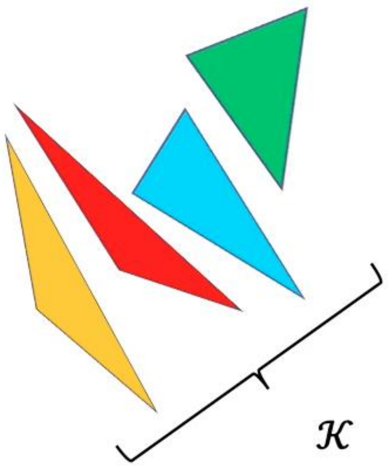


Step change in pixel color



Solution using soft aggregation

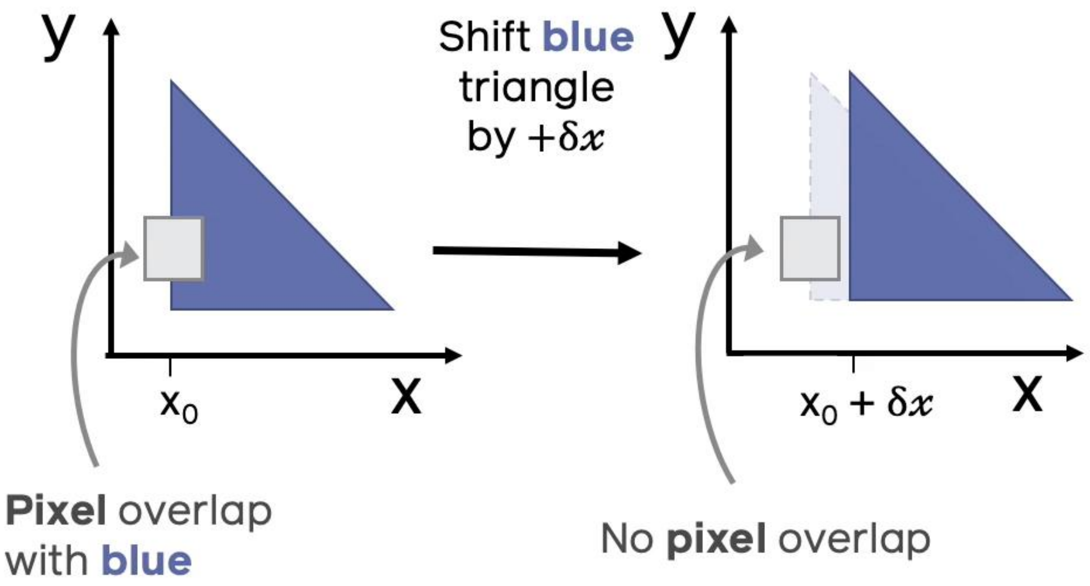
Blend closest K faces in the Z direction



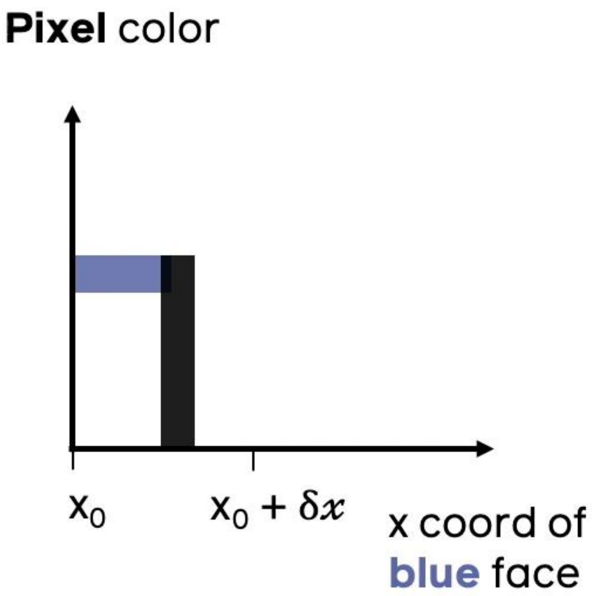
114

Traditional rasterization problem 2

XY discontinuity

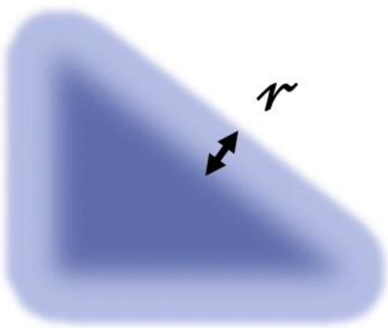


Step change in pixel color



Solution using soft aggregation

Consider faces which fall within a blur radius



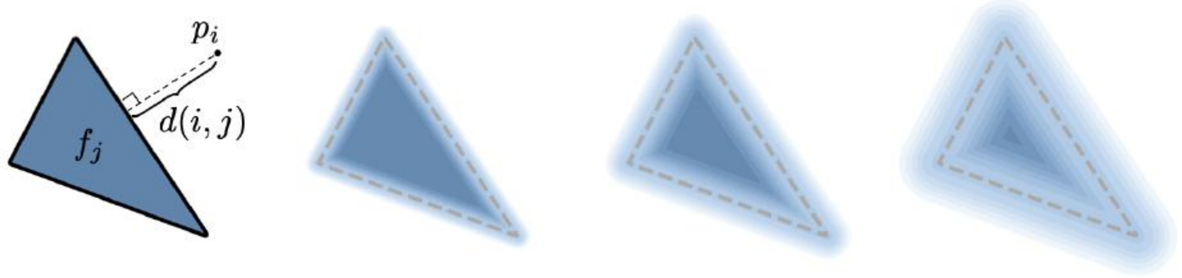
120

Fuzzy geometry

Probability map D_j at pixel p_i:

$$D_j^i = sigmoid(\delta_j^i * \frac{d^2(i,j)}{\sigma})$$

$$\delta_j^i = \{+1, \text{if } p_i \in f_j; -1, \text{otherwise}\}$$



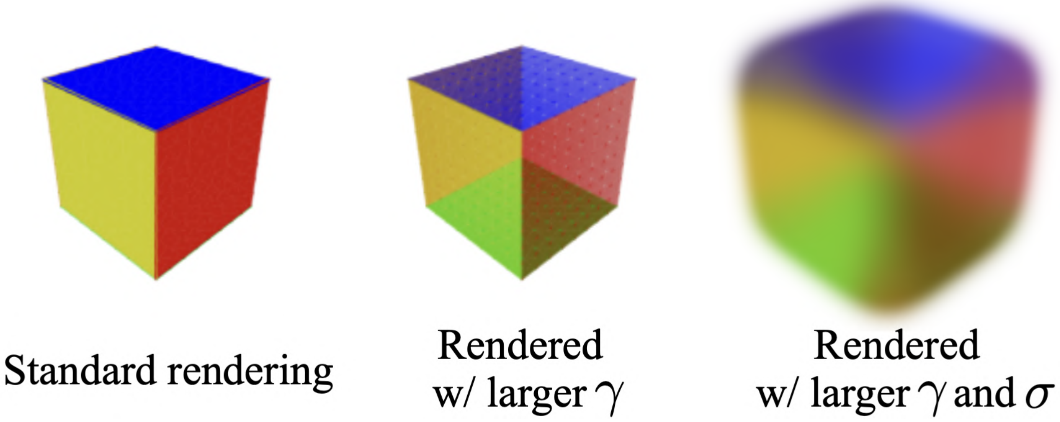
(a) ground truth (b) $\sigma = 0.003$ (c) $\sigma = 0.01$ (d) $\sigma = 0.03$

Aggregation

$$I^i = A_S(\{C_j\}) = \sum_j w_j^i C_j^i + w_b^i C_b$$

$$w_j^i = \frac{D_j^i exp(z_j^i/\gamma)}{\sum_k D_k^i exp(z_k^i/\gamma) + exp(\epsilon/\gamma)}$$

- z_j^i - normalized inverse depth
- ϵ (for background)
- γ - sharpness of the aggregate function.



PyTorch3D