Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning

Shichen Liu, Tianye Li, Weikai Chen,
Hao Li

University of South California

Presenter:
Siddhesh Kalekar (IIT Delhi)

COL877 : Course Evaluation
September 19, 2023

Introduction

- We're addressing the problem of getting 3D definitions from 2D images
- Problem in the pipeline is Rasterization and Z-buffering are non-differentiable
- Thus, we can't pass the gradients back to variables

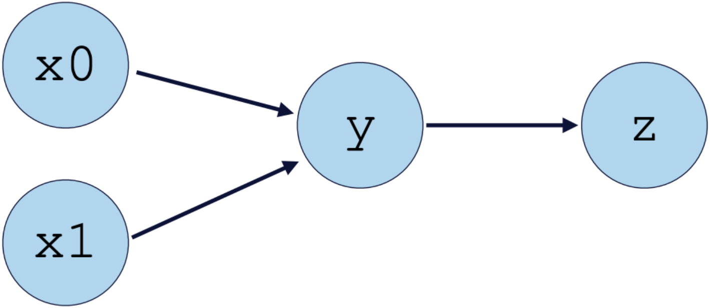
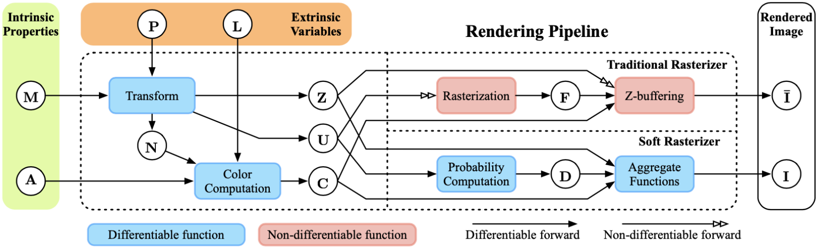


Figure: $\frac{dz}{dx_0} = \frac{dz}{dy} * \frac{dy}{dx_0}$

Proposed Idea : Change the rendering pipeline



- Variables : Mesh (M), Appearance (A), Camera (P), Light (L)
- Normals (N), Depths (Z), Image-space (U), Vertex colours (C)

Rasterization (XY discontinuity) and Z-buffering (Z discontinuity) aren't differentiable functions, hence we replace them with equivalent with differentiable functions "Probability Computation" and "Aggregation function"

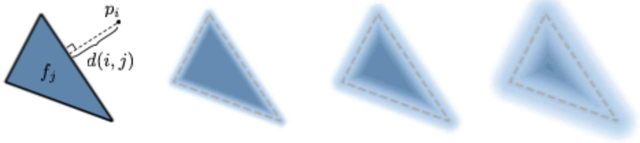
Probability Map

Define, probability map D_j for every triangle f_j and get the influence triangle f_j onto pixel p_i i.e every triangle has some infuence on every pixel

D_j at pixel p_i is defined as follows :

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

where, $sigmoid(x) = \frac{1}{1+e^{-x}}$, $\delta_j^i = +1$ if $p_i \in f_j$ else -1
$d(i,j)$ is closest distance from p_i to f_j's edges, $\sigma > 0$ sharpness scalar



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

Aggregate Function

Color map C_j for every pixel p_i is defined using weighted average of barycentric coordinates for triangle f_j if inside else 0.

Aggregate function is a function of $\{D_j\}$, $\{C_j\}$ and depth $\{z_j\}$, function output at pixel i is given by,

$$I^i = A(\{D_j\}, \{C_j\}, \{z_j\}) = (\Sigma_j w_j^i C_j^i) + w_b^i C_b$$

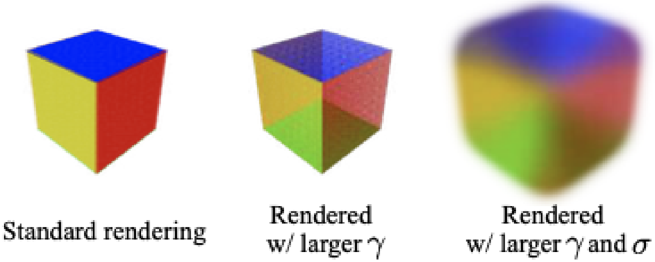
where, $w_b^i + \Sigma_j w_j^i = 1$ and w_j^i is defined as

$$w_j^i = \frac{D_j^i * \exp^{-z_j^i/\gamma}}{\exp^{\epsilon/\gamma} + \Sigma_k D_k^i * \exp^{-z_k^i/\gamma}}$$

Higher weights are assigned to closer triangles, as γ is sharpness constant and ϵ is background color dependent constant

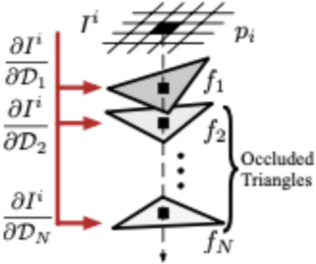
Every triangle is contributing to the color of all pixels

Rendered example



Comparison

- We can pass gradients from pixels to triangles now as shown in figure below - (Z-discontinuity solved)
- Screen space gradients now exists because of probability map of triangles - (XY-discontinuity solved)



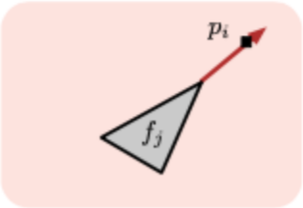
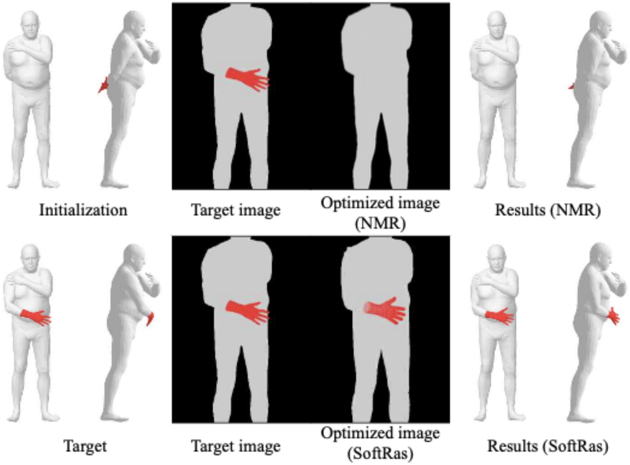


Image based shape fitting

- Occlusion Awareness : Gradients must pass through



Mesh Reconstruction

Minimise $L = L_s + \lambda L_c + \mu L_g$ where

- L_s is silhouette loss, $1 - \frac{||I_s \bigotimes \hat{I}_s||_1}{||I_s \bigoplus \hat{I}_s - I_s \bigotimes \hat{I}_s||_1}$, silhoutte is binary image
- L_c is coloured loss, $||I_c - \hat{I}_c||_1$
- L_g is geomteric loss (Laplacian of shape and colour)

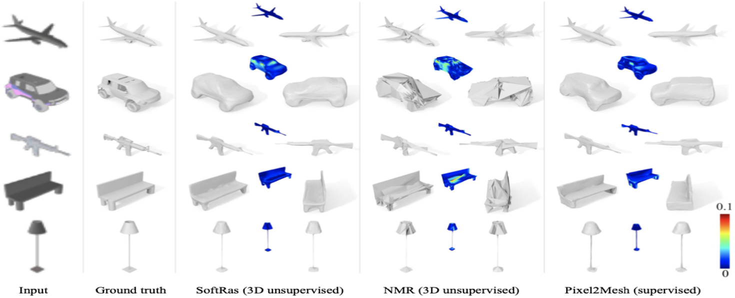


Figure: 64 input images, $\lambda = 1$, $\mu = 10^{-3}$

Summary

1. Change of rendering pipeline by replacing rasterization and z-buffering with probability maps and Aggregator function to form **Soft Rasterizer**
2. This makes the functions differentiable thus differentiable rendering is possible
3. Performance-comparison of new pipeline with other differentiable rendering algorithms with applications in
   - Shape fitting
   - Mesh reconstruction

More details can be found in : *Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning*

Thank You !!!