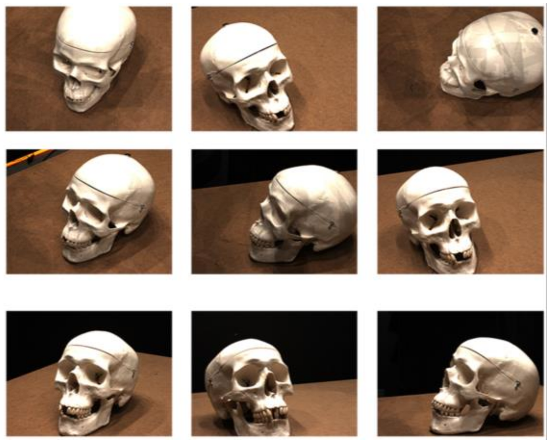
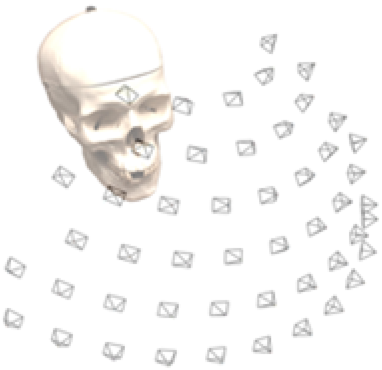
Differentiable rendering
For Multi-View 3D Reconstruction

January 23rd, 2023

Multi-View 3D Reconstruction



$I_1, \ldots, I_n$



$P_1, \ldots, P_n$

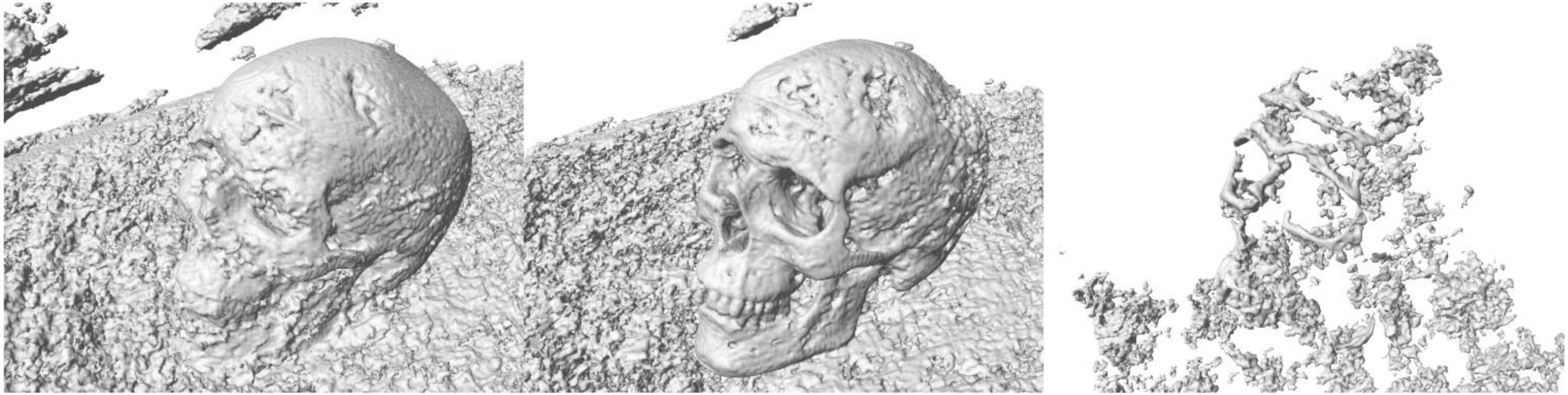
$P_i = K_i[R_i|t_i]$

Dense 3D reconstruction
(Mesh, Voxels, Implicit)

Classical methods

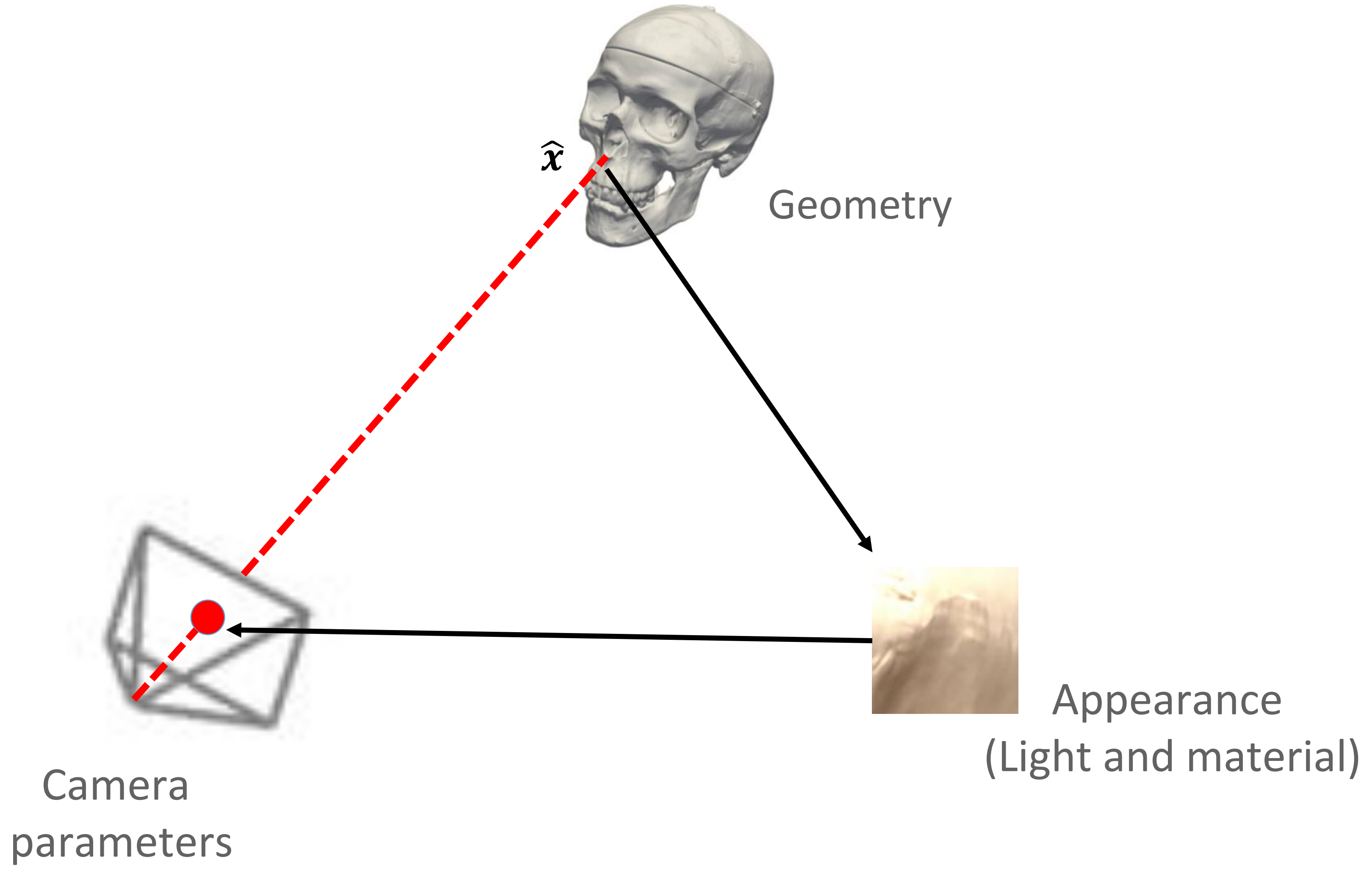
COLMAP, Schonberger et. al. (2016)

NeRF – Volume rendering

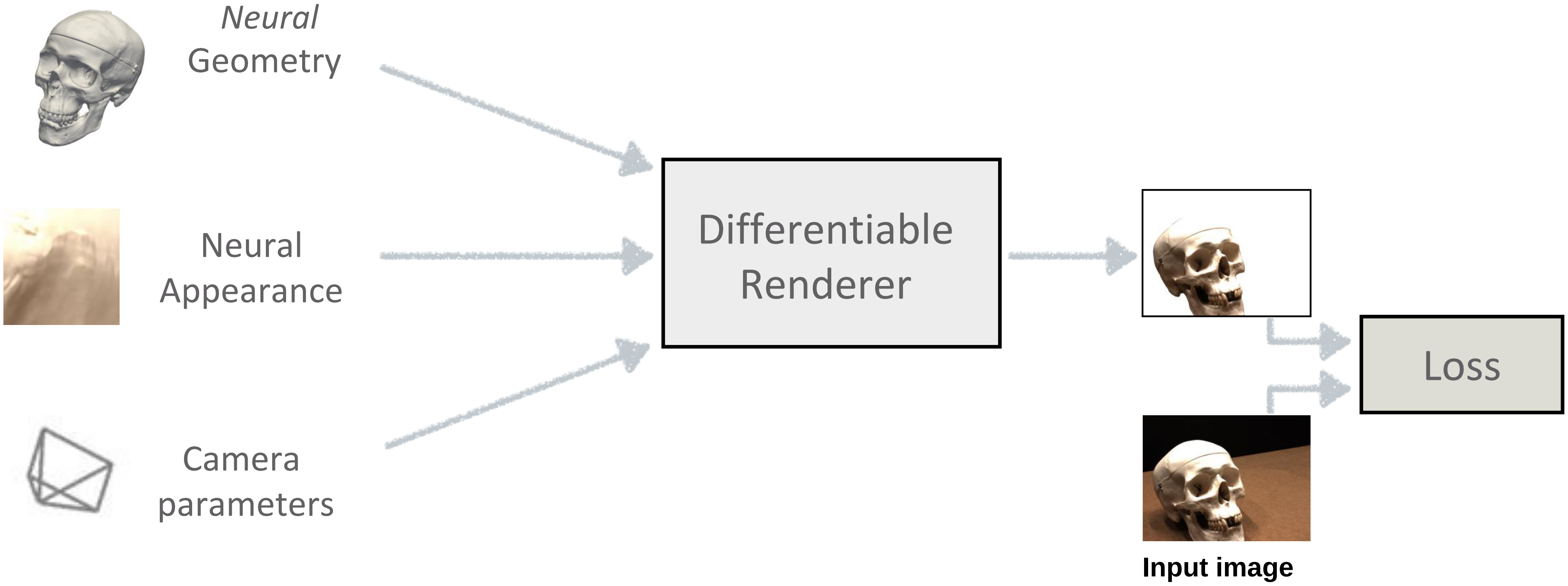


$\sigma = 1$ $\sigma = 50$ $\sigma = 500$

Differentiable **surface** rendering

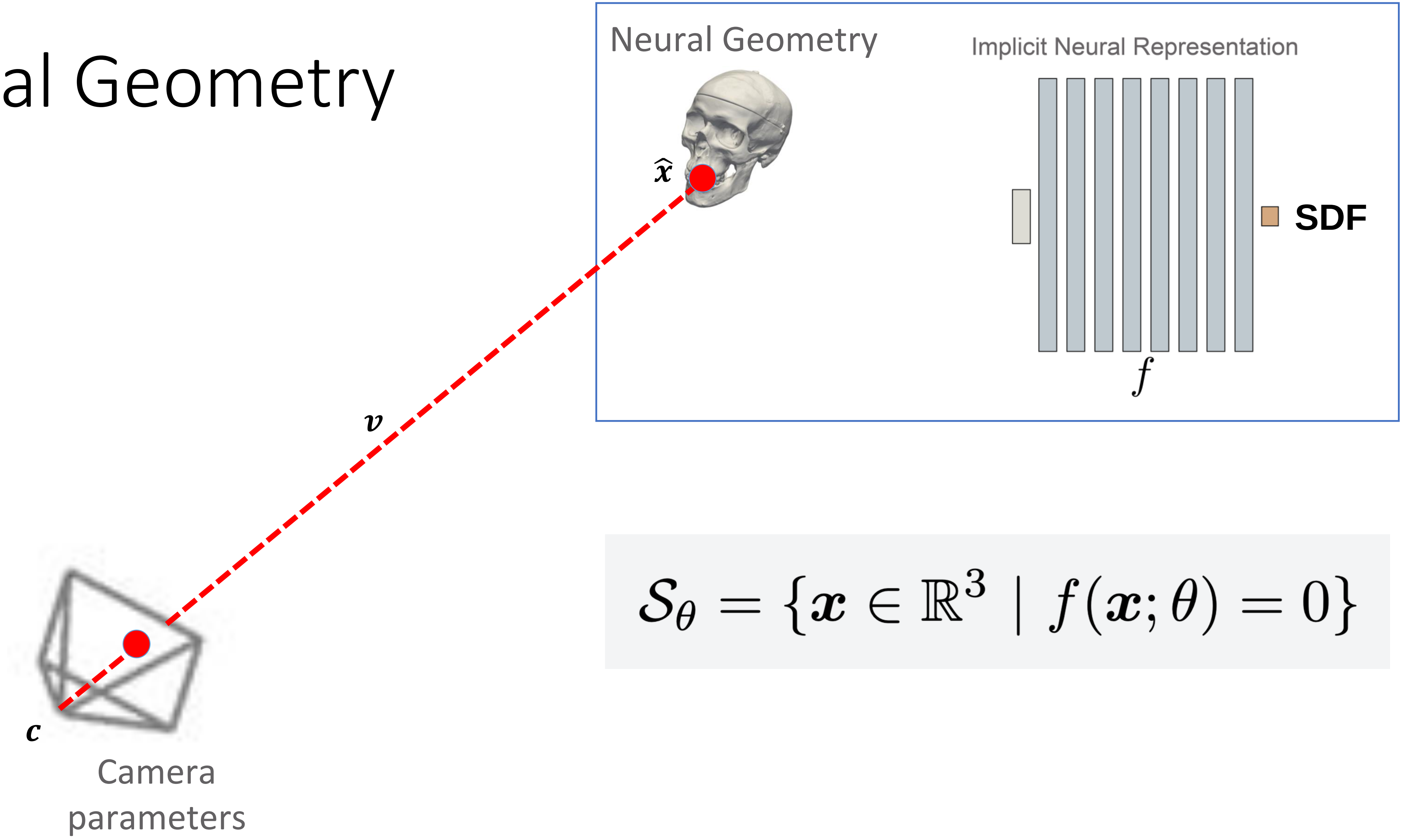


Implicit Differentiable Renderer (IDR)



Multiview neural surface reconstruction by disentangling geometry and appearance (IDR), Yariv et. al., NeurIPS 2020

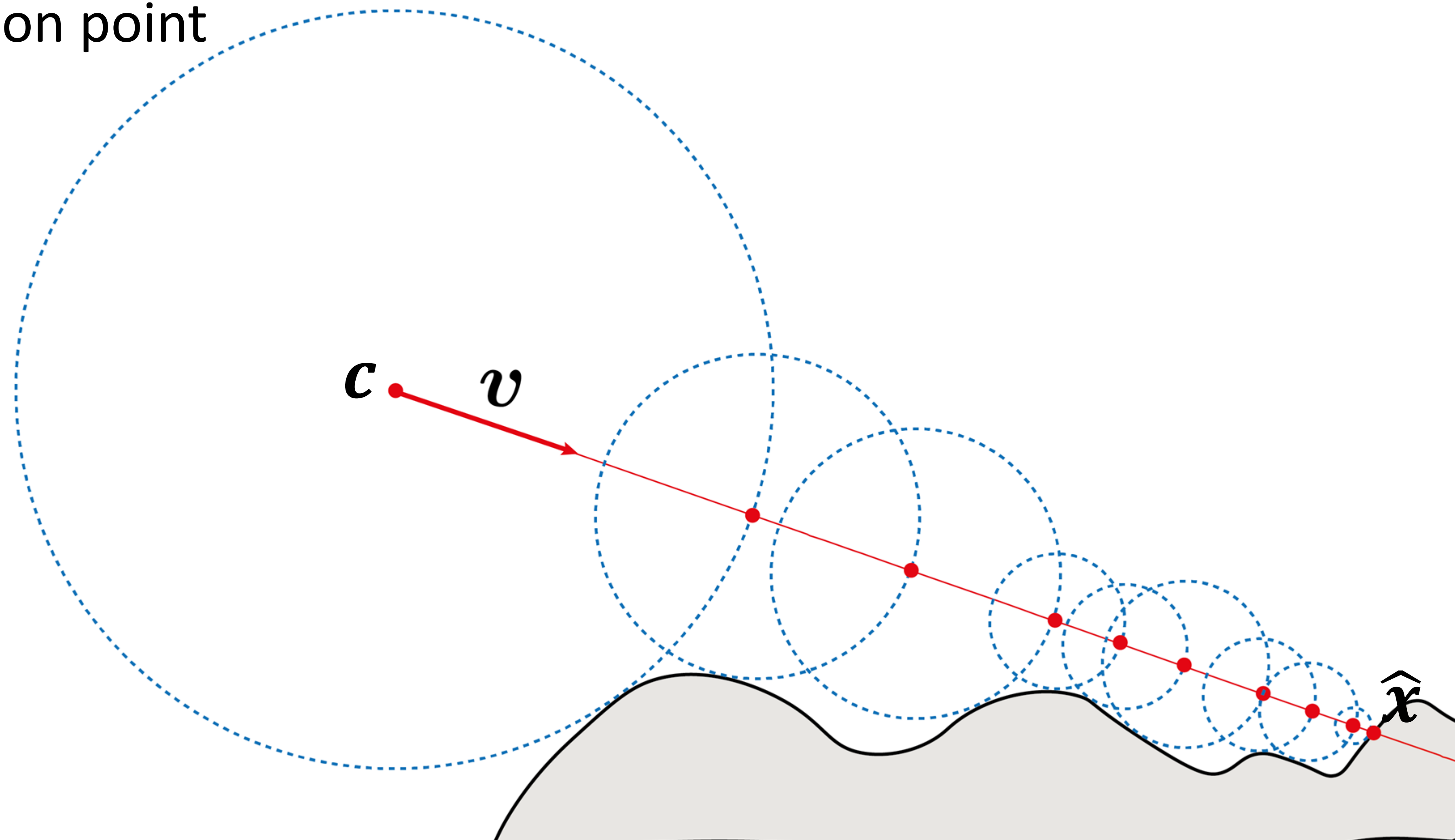
Neural Geometry



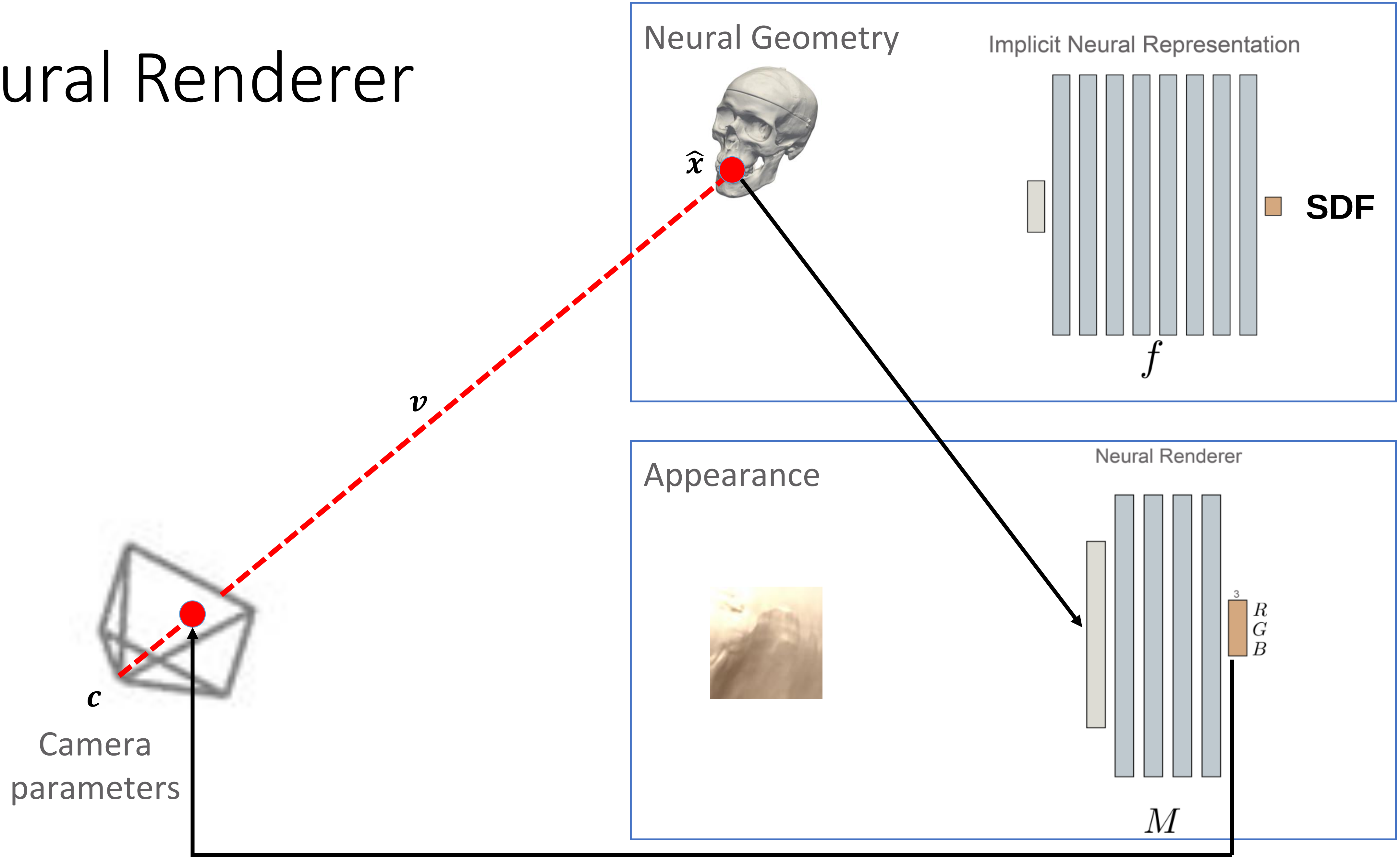
$$S_\theta = \{\boldsymbol{x} \in \mathbb{R}^3 \mid f(\boldsymbol{x};\theta) = 0\}$$

Sphere tracing

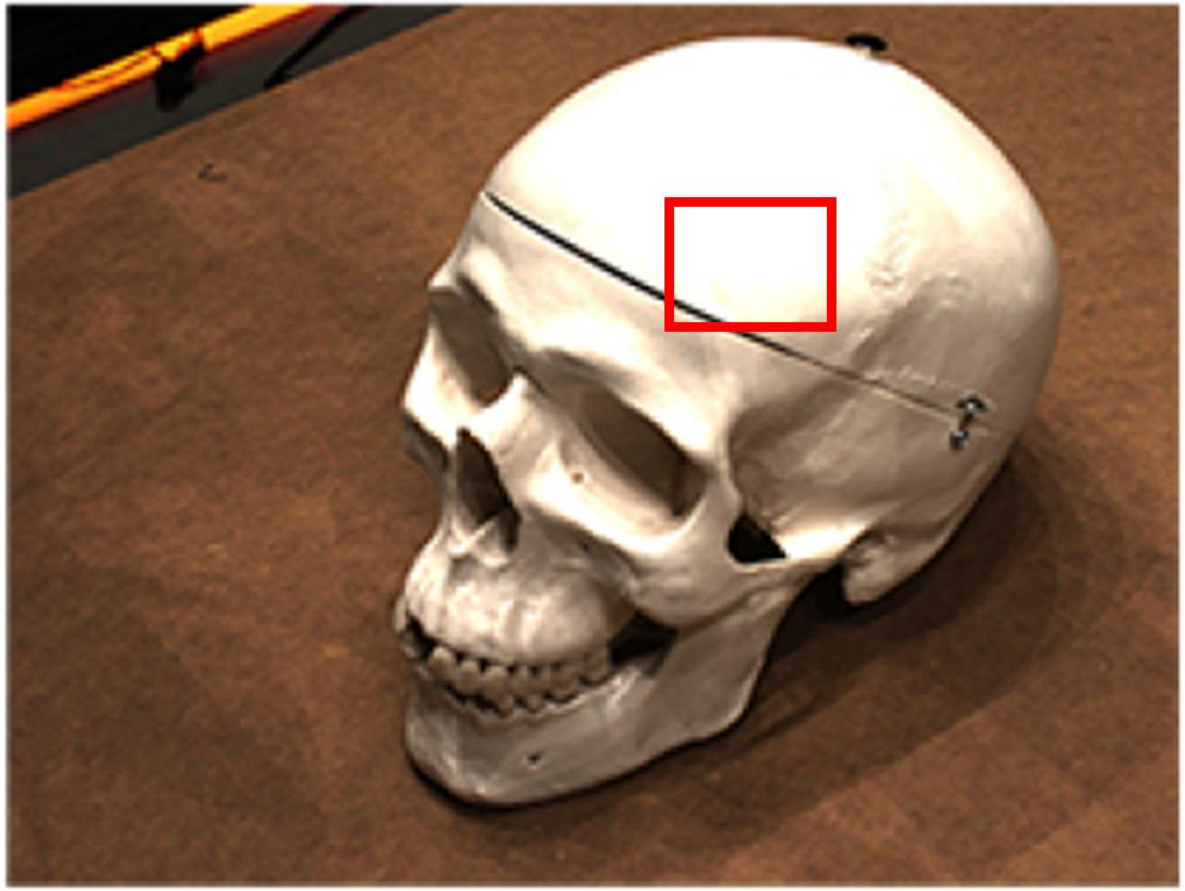
- Finding intersection point

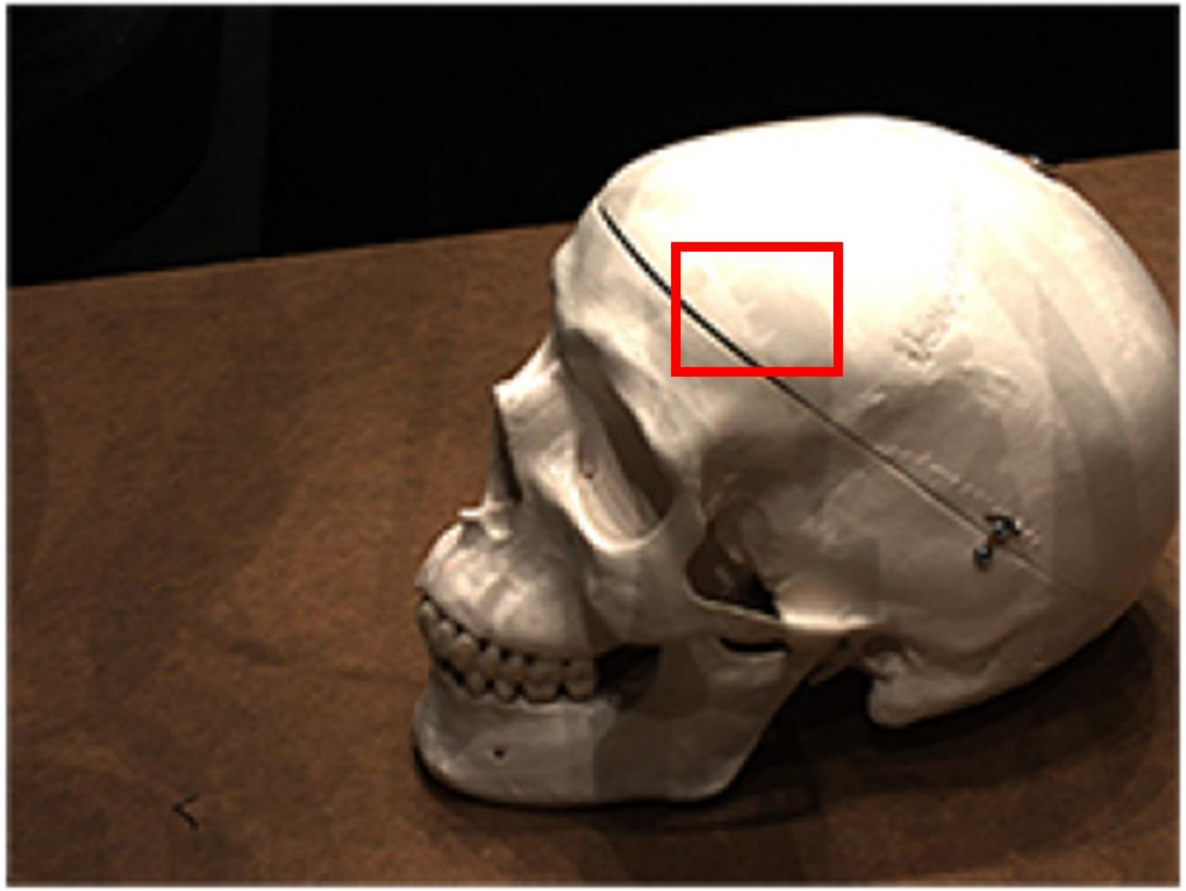


Neural Renderer

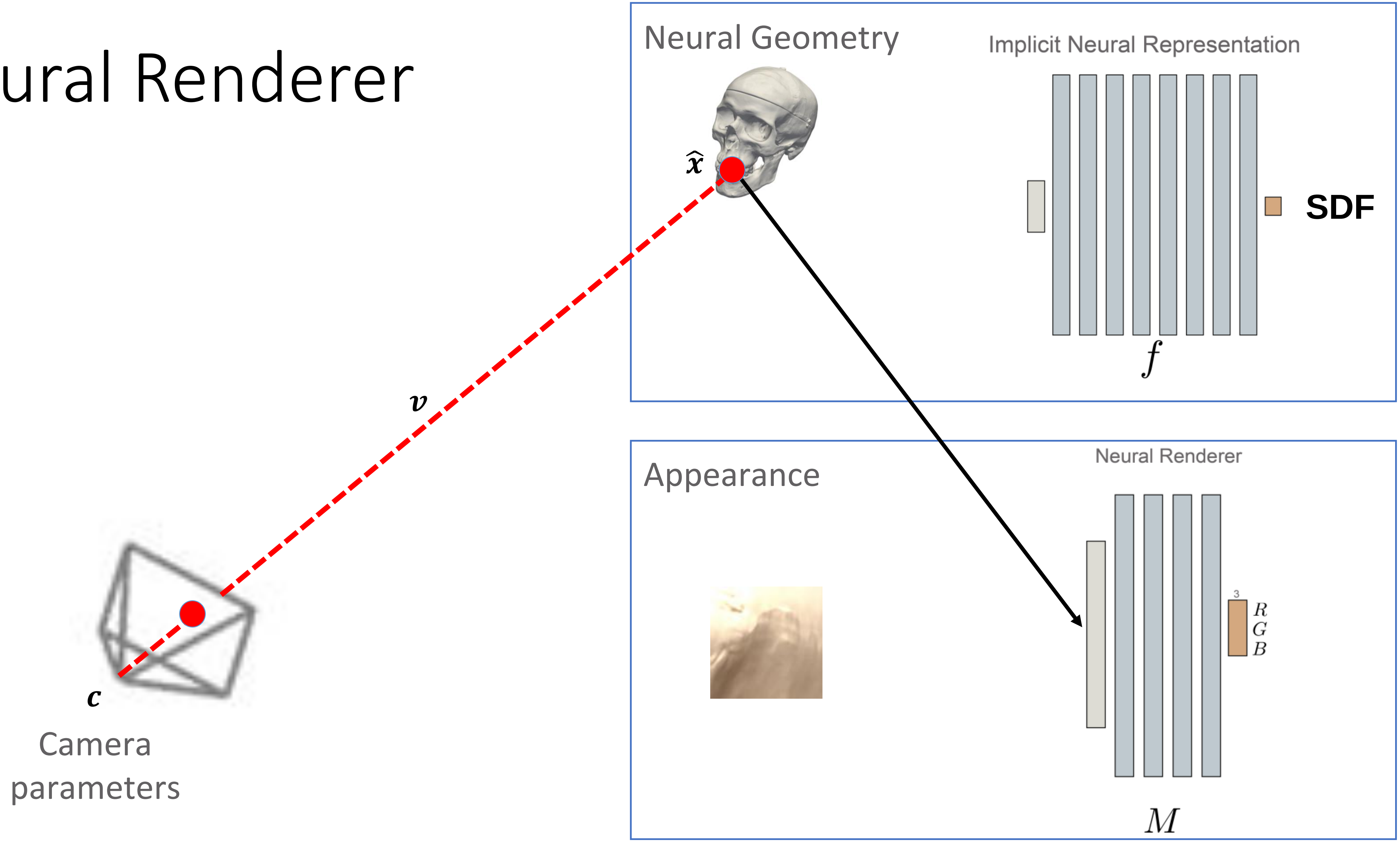


View dependent color

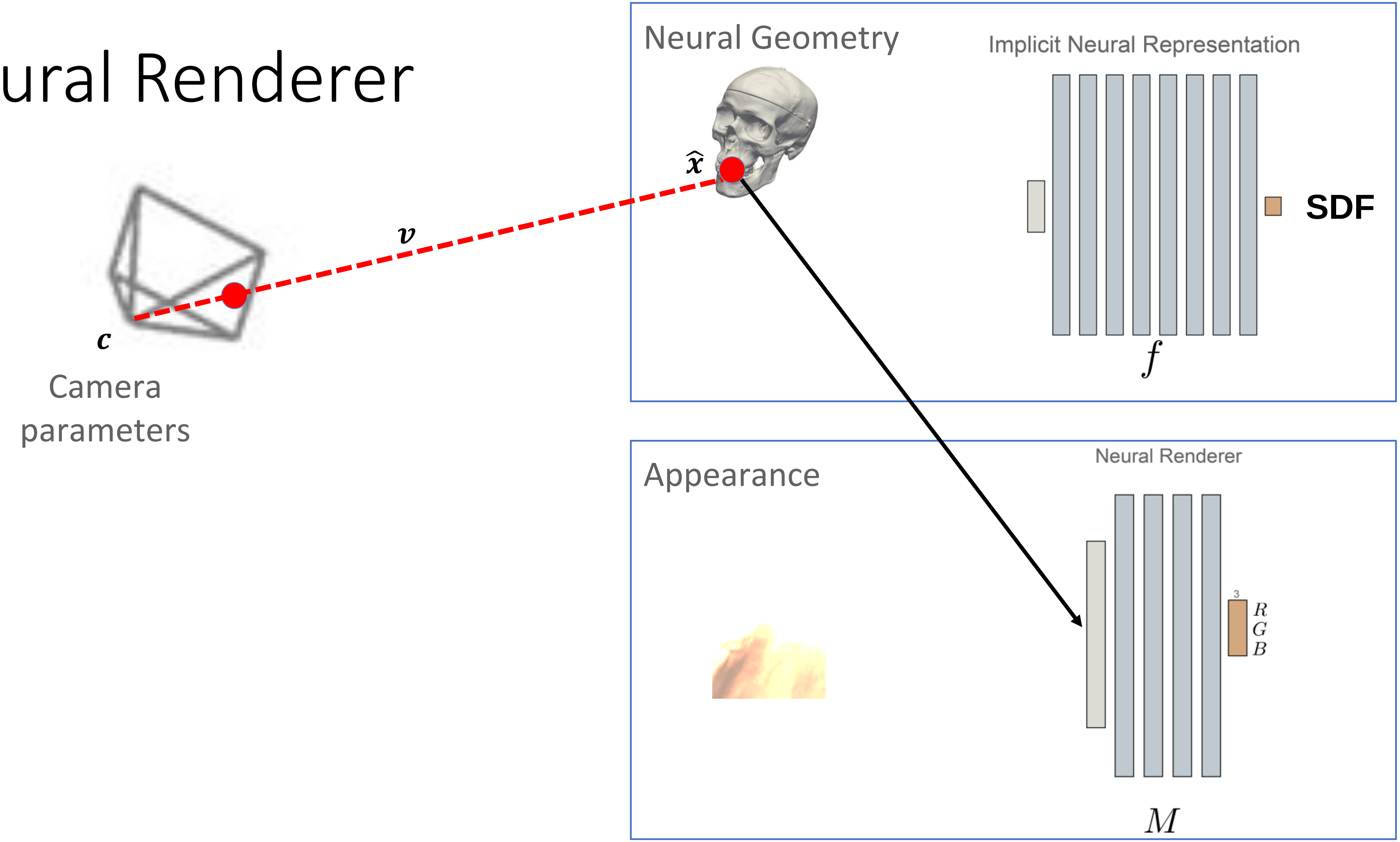




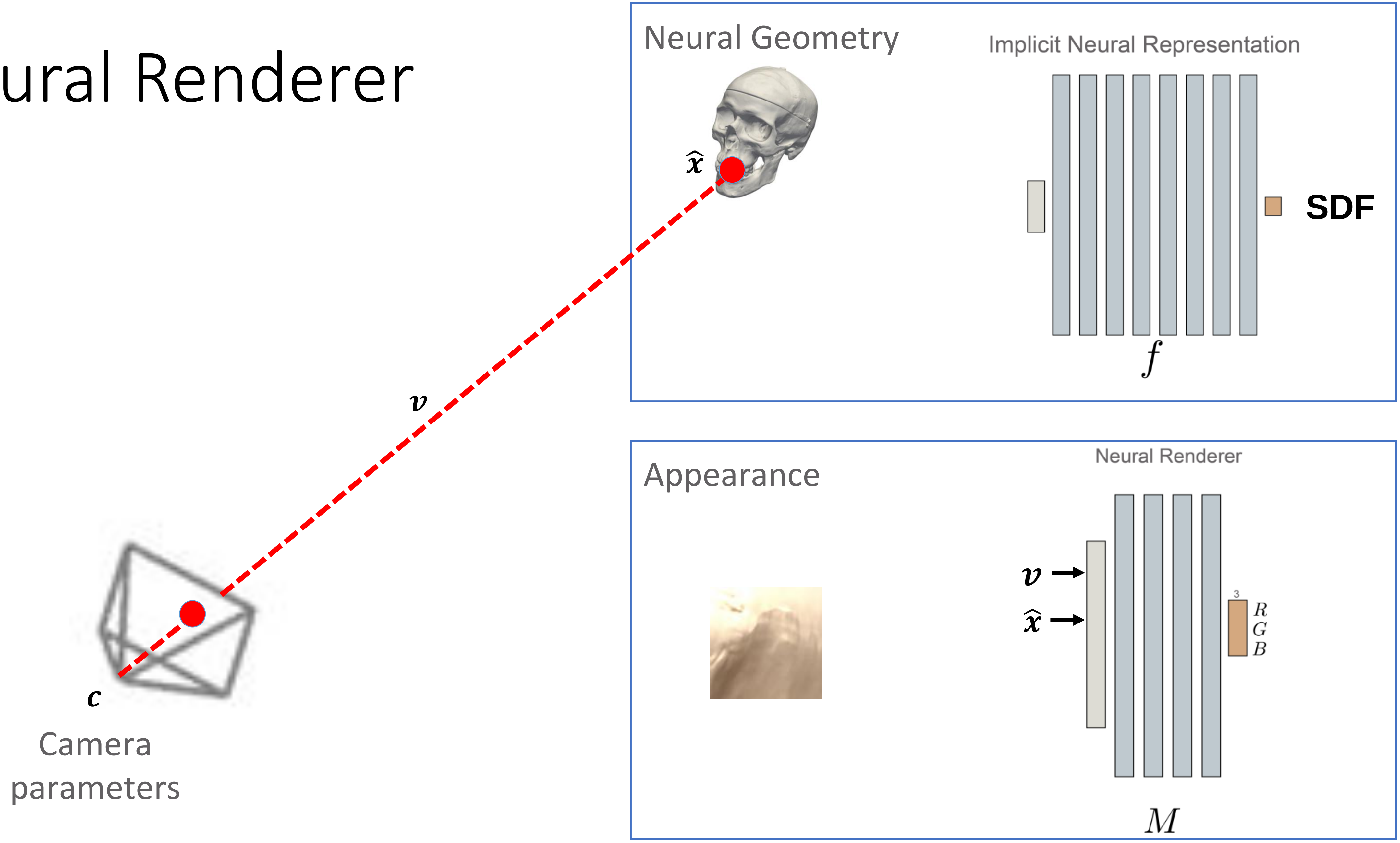
Neural Renderer



Neural Renderer

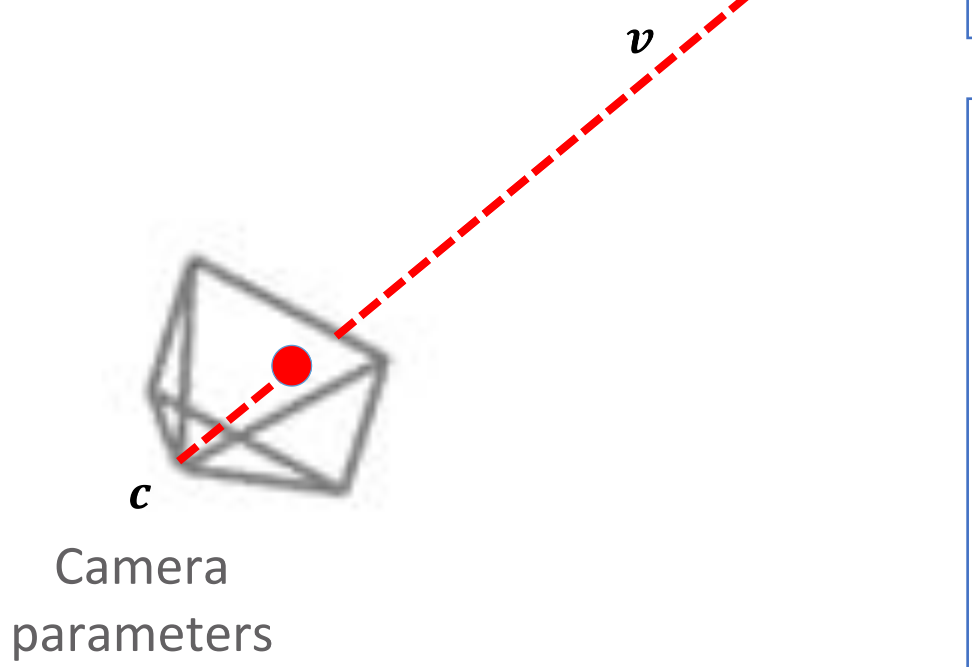


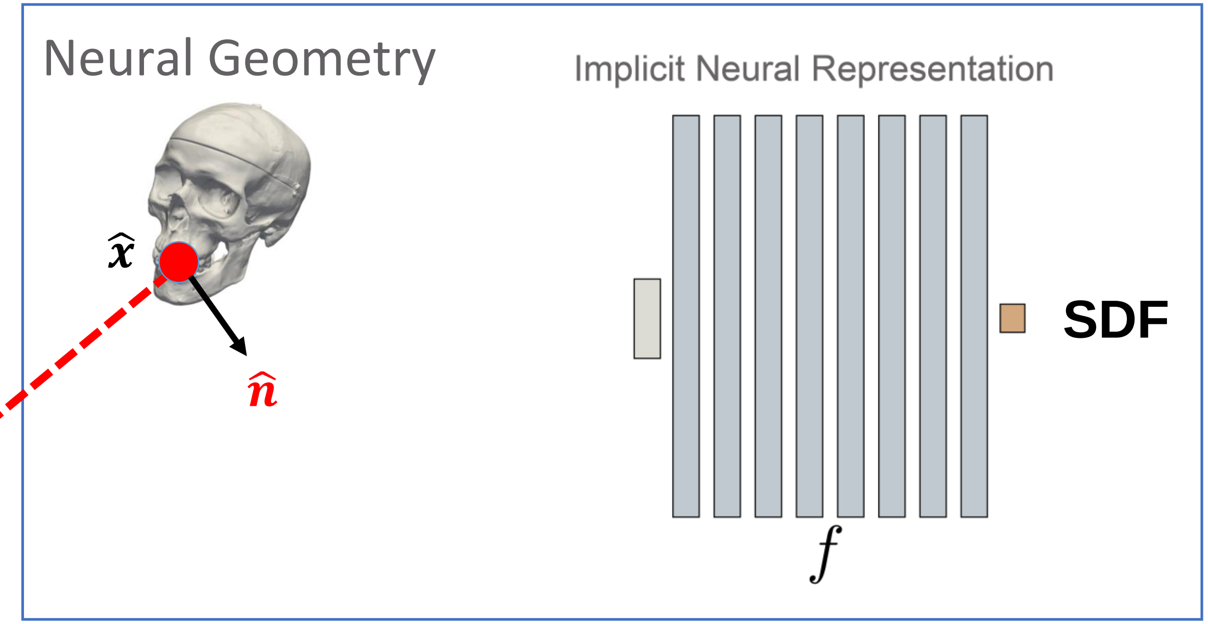
Neural Renderer

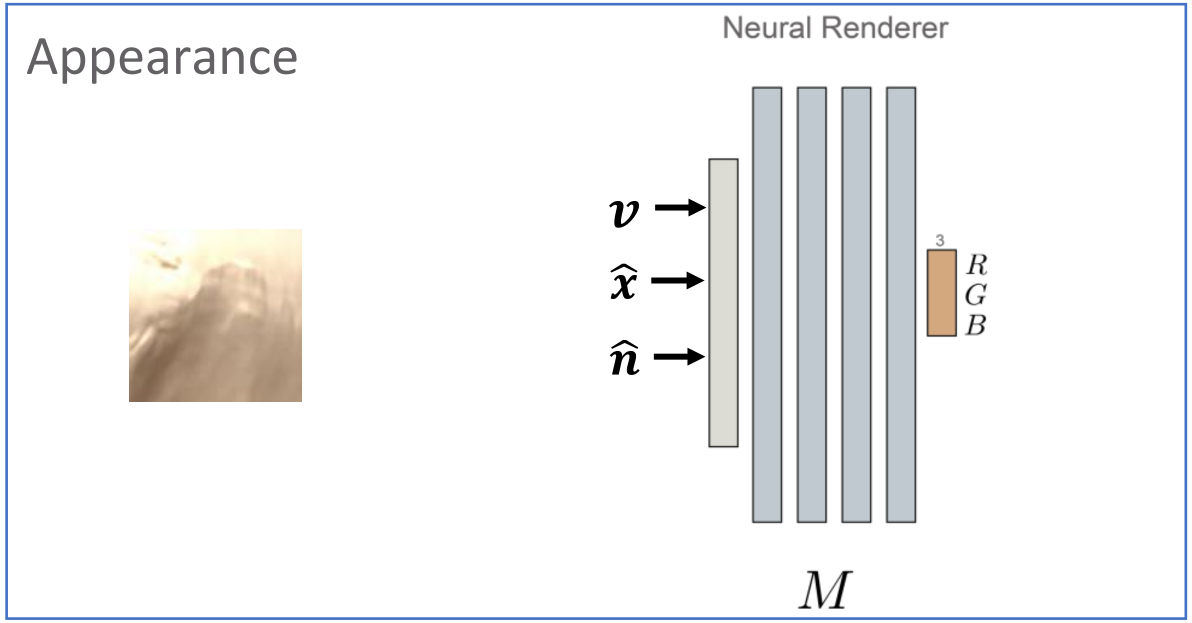


Neural Renderer

- Can we render a different geometry with the same renderer?
- What kind of input can "encourage" the renderer to generalize?

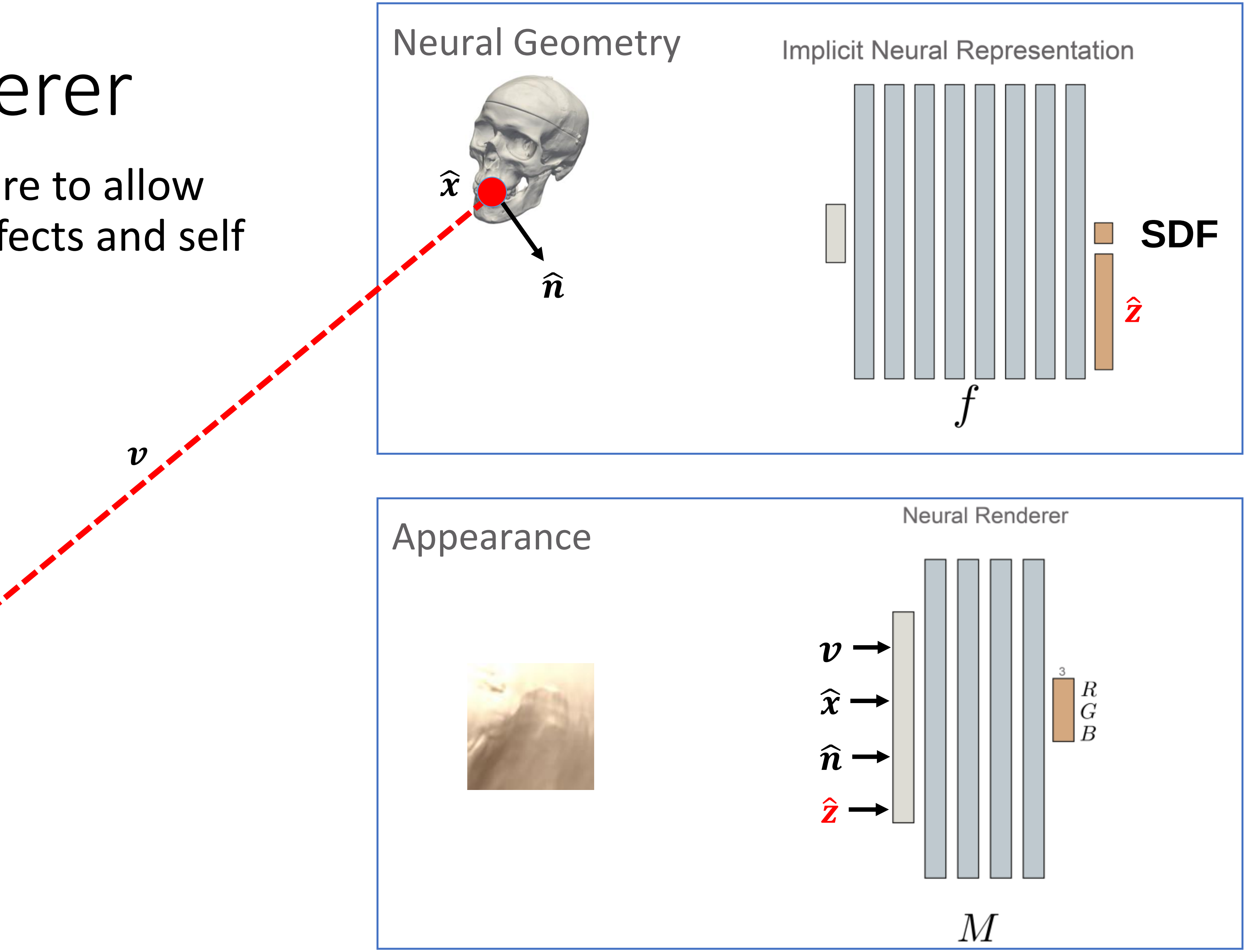




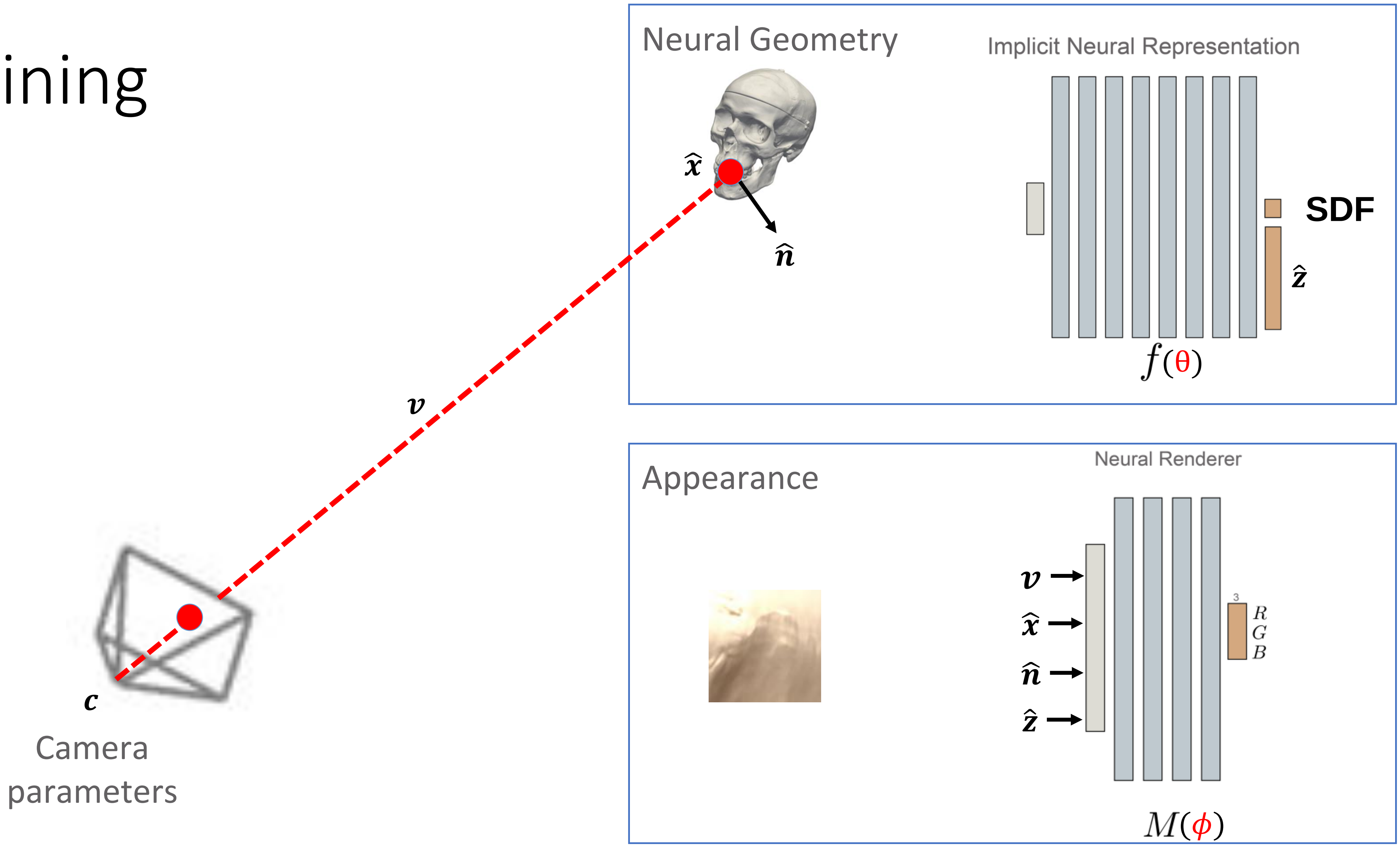


Neural Renderer

- Adding a global feature to allow secondary lighting effects and self shadows



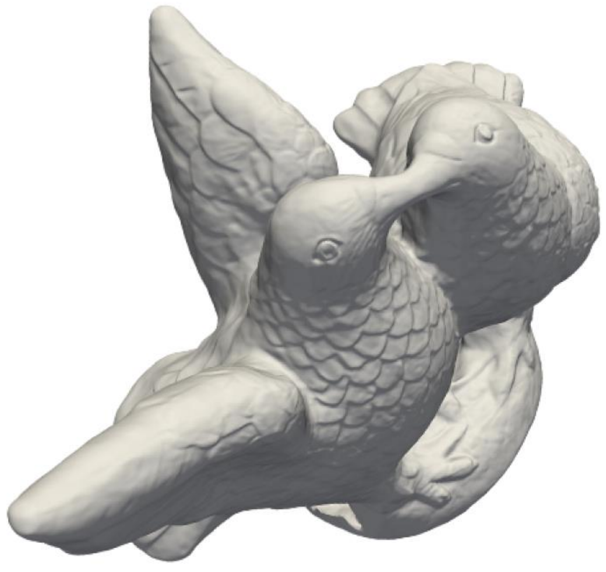
Training



Neural Geometry
Implicit Neural Representation
$\hat{x}$
$\hat{n}$
SDF
$\hat{z}$
$f(\theta)$
v
c
Camera parameters
Appearance
Neural Renderer
v
$\hat{x}$
$\hat{n}$
$\hat{z}$
3
R G B
$M(\phi)$

Positional encoding

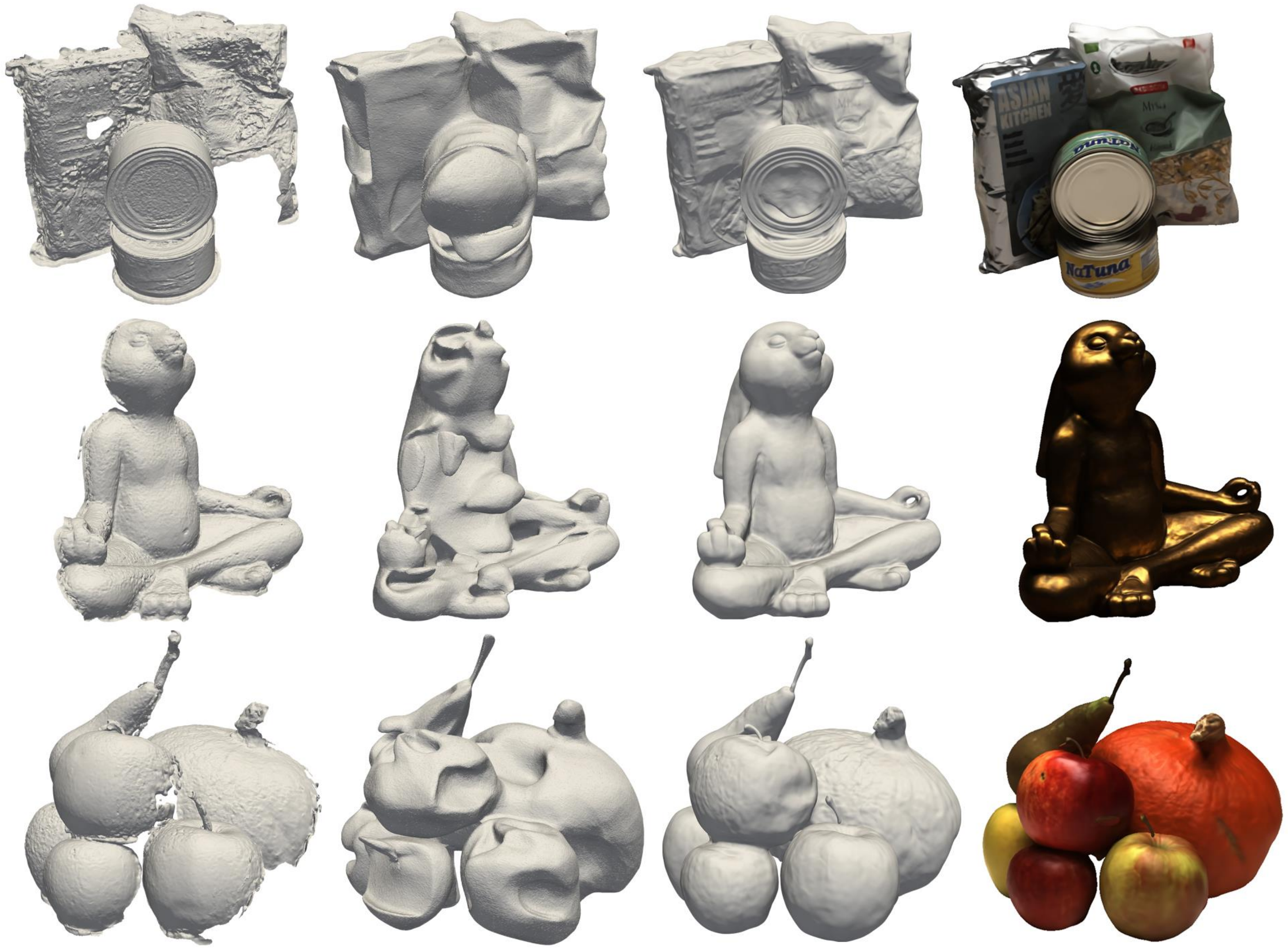
No PE, 5000 epochs



2000 epochs

Results

Results - Comparison



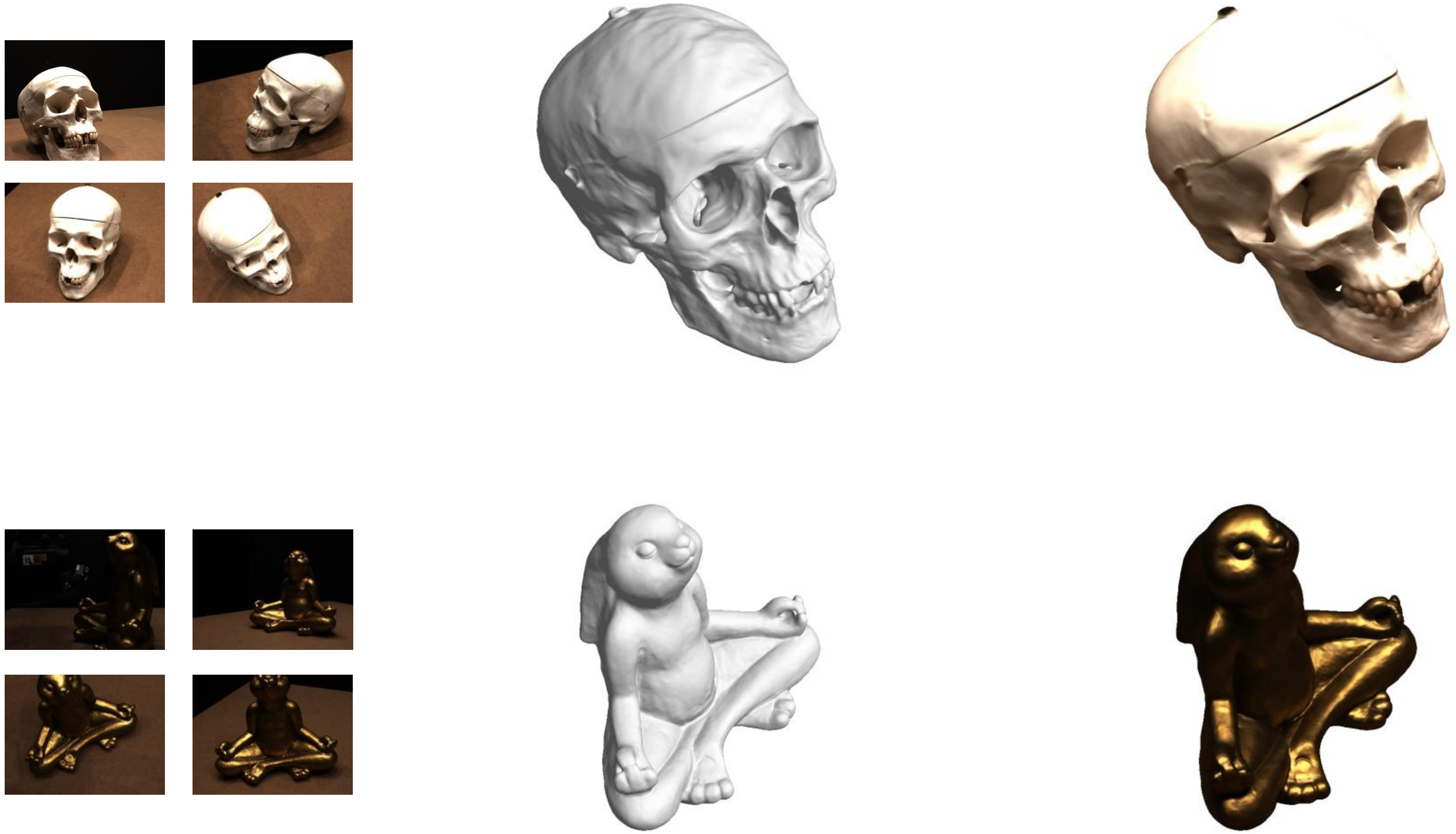
COLMAP | DVR($\hat{x}$, ~~$\hat{n}$~~, ~~v~~) | IDR | IDR rendering

DTU, Jensen et. al., 2020

Ablation study

$M(\hat{\boldsymbol{x}}, \cancel{\hat{\boldsymbol{n}}}, \boldsymbol{v}, \boldsymbol{z})$ $M(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \cancel{\boldsymbol{v}}, \boldsymbol{z})$ $M(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{v}, \cancel{\boldsymbol{z}})$ $M(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{v}, \boldsymbol{z})$

Disentangling geometry and appearance



Input images Geometry Rendering

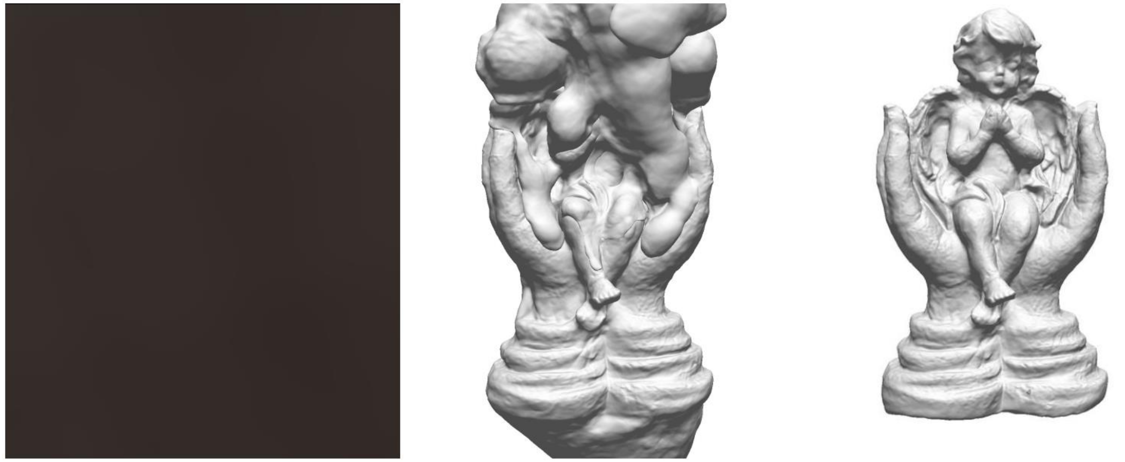
IDR

Pros:

- Simultaneously learns geometry and appearance
- Highly accurate 3D reconstruction

Cons:

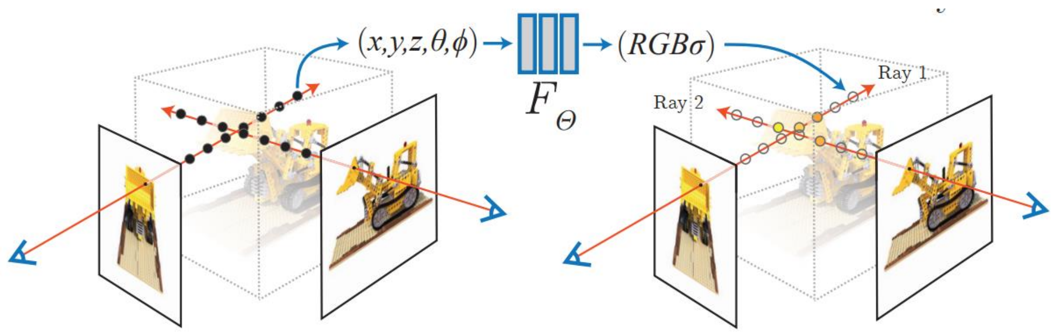
- Requires initialization
- Requires object masks



Random Init. No Masks | Sphere Init. No Masks[1] | Sphere Init. With Masks

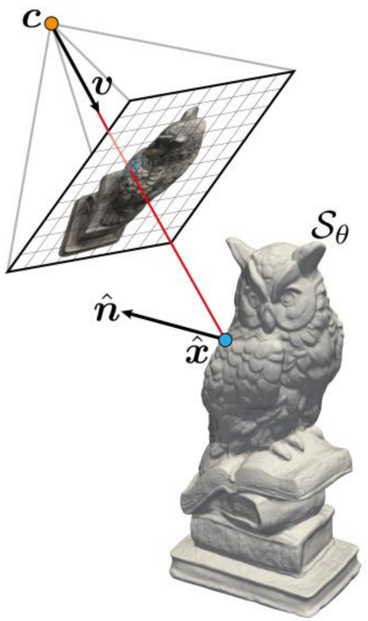
Nerf – Volume Rendering

- Poor geometry



IDR – Surface Rendering

- Requires masks
- Requires initialization

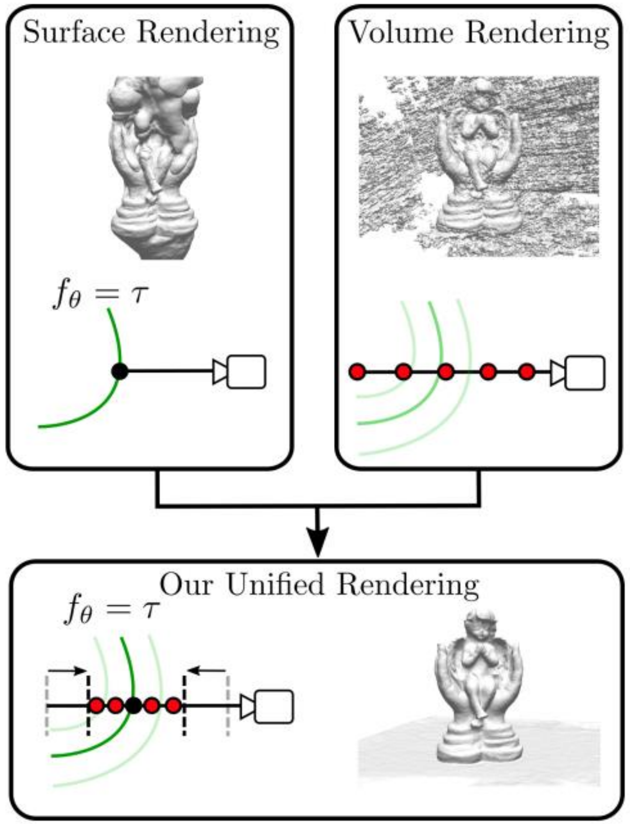


Density **Vs** Geometry

Sampling **Vs** Intersection point

UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction

- Unified framework for implicit surfaces and radiance fields
- Reconstructing **solid** objects from a set of RGB images without masks

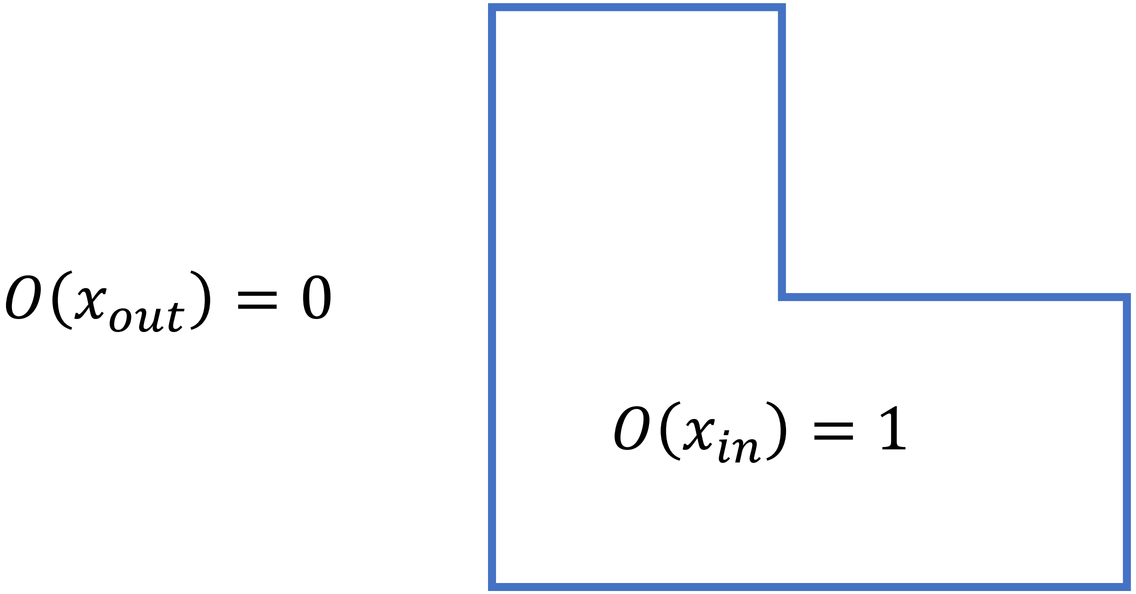


Oechsle et al.

Density to occupancy

- Occupancy:

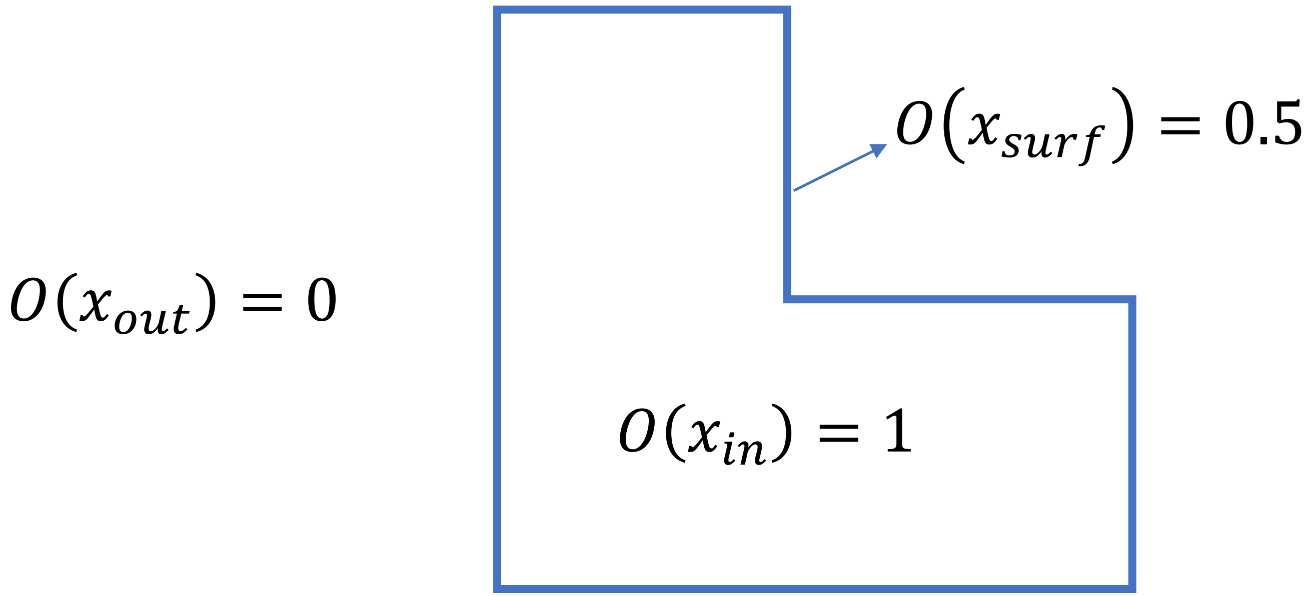
$$O: \mathbb{R}^3 \rightarrow \{0,1\}$$



Density to occupancy

- Occupancy:

$$O_\theta : \mathbb{R}^3 \rightarrow [0,1]$$



Density to occupancy

- NeRF

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_\theta(\mathbf{x}_i)\,\delta_i\right)\right) c_\theta(\mathbf{x}_i, \mathbf{d})$$

$$T_i = \exp\left(-\sum_{j<i} \sigma_\theta(\mathbf{x}_j)\,\delta_j\right)$$

$\sigma(x): R^3 \rightarrow R^+$ is the density function.
$c(x, d)$ is the color function.

- Can be written as:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i(\mathbf{x}_i) \prod_{j<i} \left(1 - \alpha_j(\mathbf{x}_j)\right) c(\mathbf{x}_i, \mathbf{d})$$

$$\alpha_i(\mathbf{x}) = 1 - \exp\left(-\sigma(\mathbf{x})\,\delta_i\right)$$

$\alpha \in [0,1]$ become an occupancy indicator

Density to occupancy

- NeRF

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_\theta(\mathbf{x}_i)\,\delta_i\right)\right) c_\theta(\mathbf{x}_i, \mathbf{d})$$

$$T_i = \exp\left(-\sum_{j<i} \sigma_\theta(\mathbf{x}_j)\,\delta_j\right)$$

$\sigma(x): R^3 \rightarrow R^+$ is the density function.
$c(x, d)$ is the color function.

- Can be written as:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} o(\mathbf{x}_i) \prod_{j<i} \left(1 - o(\mathbf{x}_j)\right) c(\mathbf{x}_i, \mathbf{d})$$

$o(x): R^3 \rightarrow [0,1]$ is the occupancy function

Rendering formula

- Volume rendering

$$\hat{C}_v(\mathbf{r}) = \sum_{i=1}^{N} o_\theta(\mathbf{x}_i) \prod_{j<i} \big(1 - o_\theta(\mathbf{x}_j)\big) c_\theta(\mathbf{x}_i, \mathbf{n}_i, \mathbf{h}_i, \mathbf{d})$$

x is a 3D points
n is the surface normal
h is a feature vector
d is the viewing direction

- Surface rendering

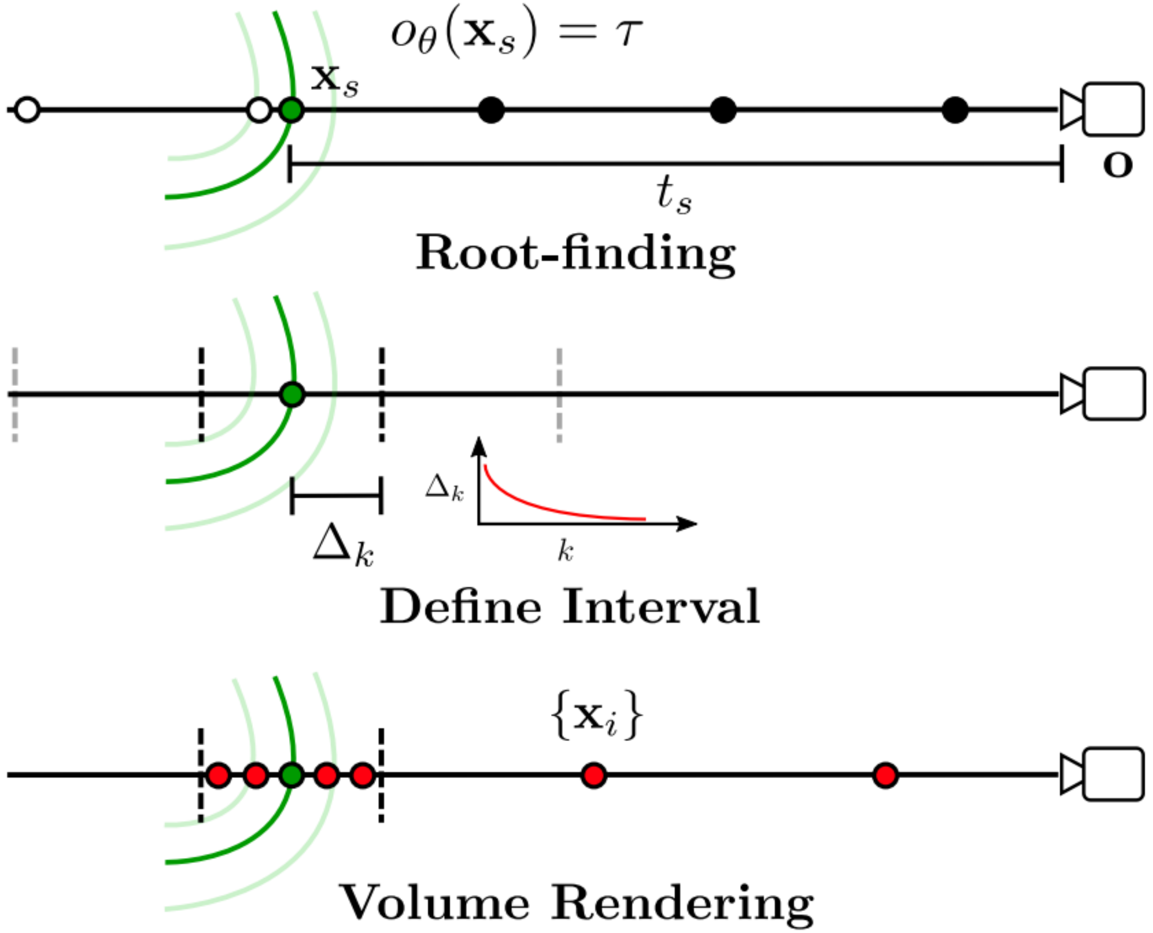
$$\hat{C}_s(\mathbf{r}) = c_\theta(\mathbf{x}_s, \mathbf{n}_s, \mathbf{h}_s, \mathbf{d})$$

Sampling Vs surface point

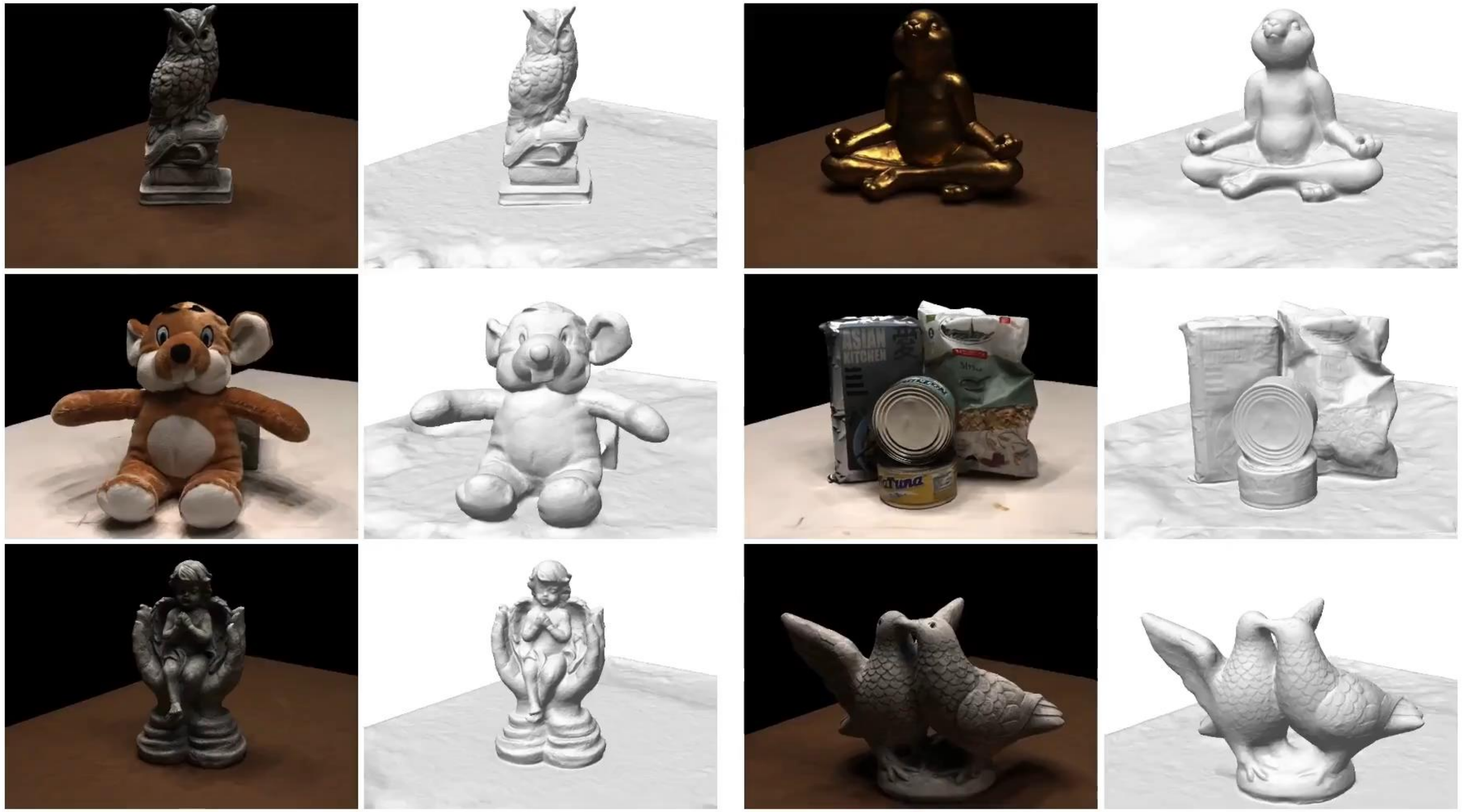
1. Find surface point
2. Define intervals:

$$\Delta_k = \max(\Delta_{\text{max}} \exp(-k\,\beta), \Delta_{\text{min}})$$

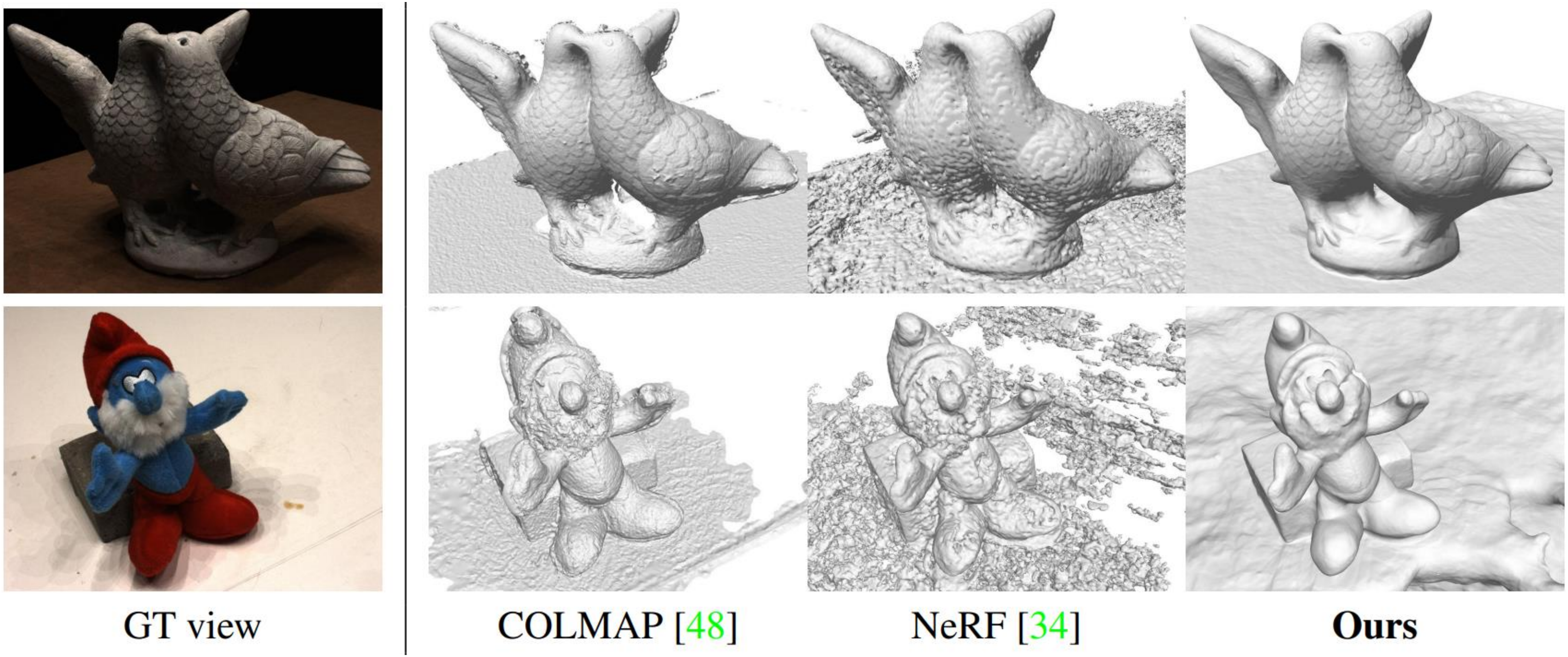
3. Sample ray



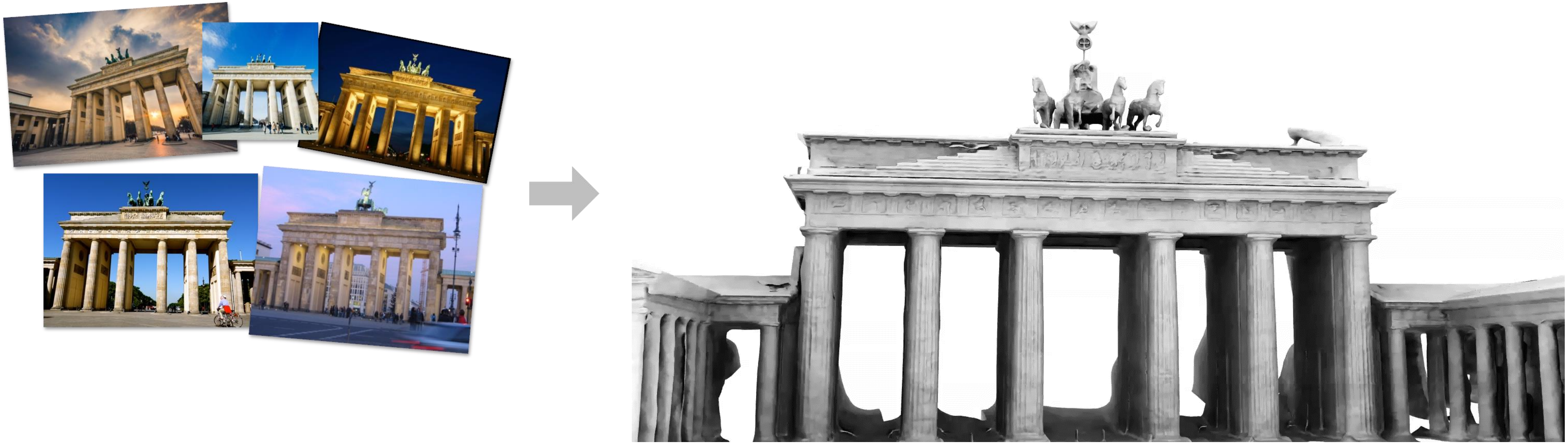
Results



Results



Neural 3D Reconstruction in the Wild



Sun et. al., SIGGRAPH 2022

Summary

- The underlying geometry, obtained by volume renderers is usually non-smooth and contains artifacts.
- Differentiable surface renderers produce highly accurate 3D reconstructions. However, they depend on object masks.
- Unifying surface and volume rendering is possible!

Questions?