Neural Inverse Rendering

Pratul Srinivasan

ECCV Advances in Image Manipulation Workshop

October 23, 2022

About Me

PhD at UC Berkeley 2014-2020
Advisors: Ravi Ramamoorthi and Ren Ng

Research Scientist at Google Research
Manager: Jon Barron
with Ben Mildenhall and Peter Hedman

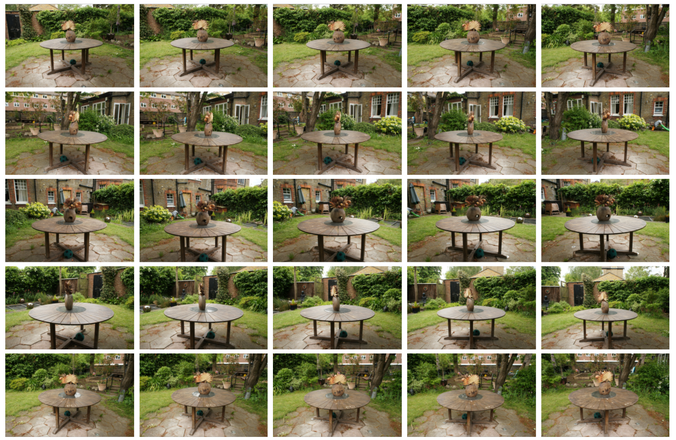



Computer vision as inverse rendering

3D
Scene
Representation

Rendering

Computer vision as inverse rendering

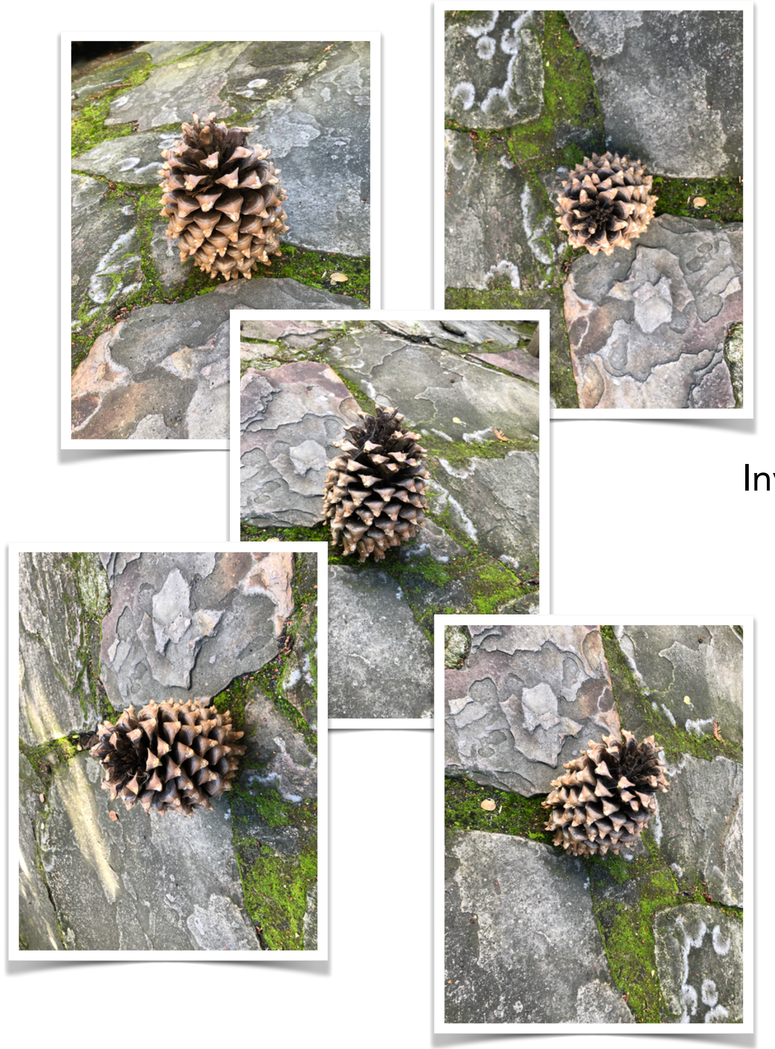


Inverse Rendering

3D
Scene
Representation

Rendering

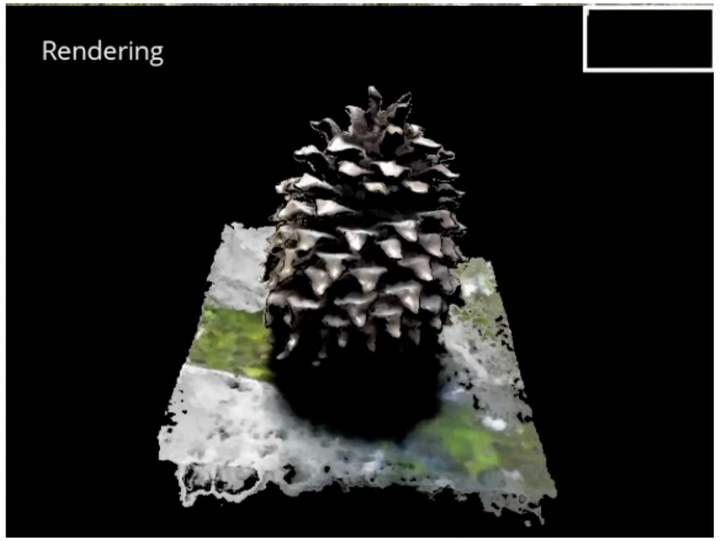
Computer vision as inverse rendering



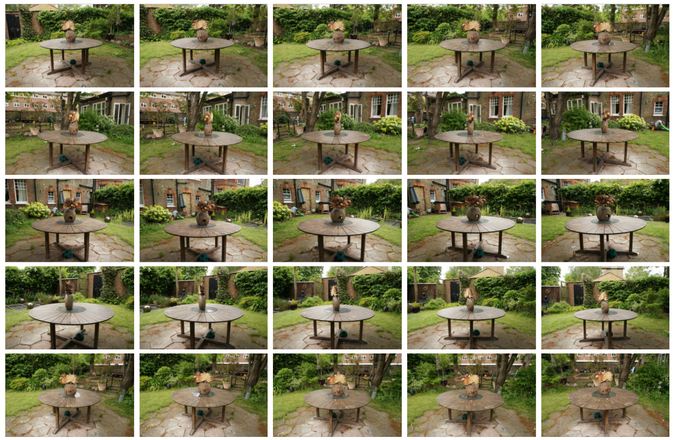
Inverse Rendering

3D
Scene
Representation

Rendering



Approach: “Optimization-based inverse rendering” or “Analysis-by-synthesis” or “Render-and-compare”

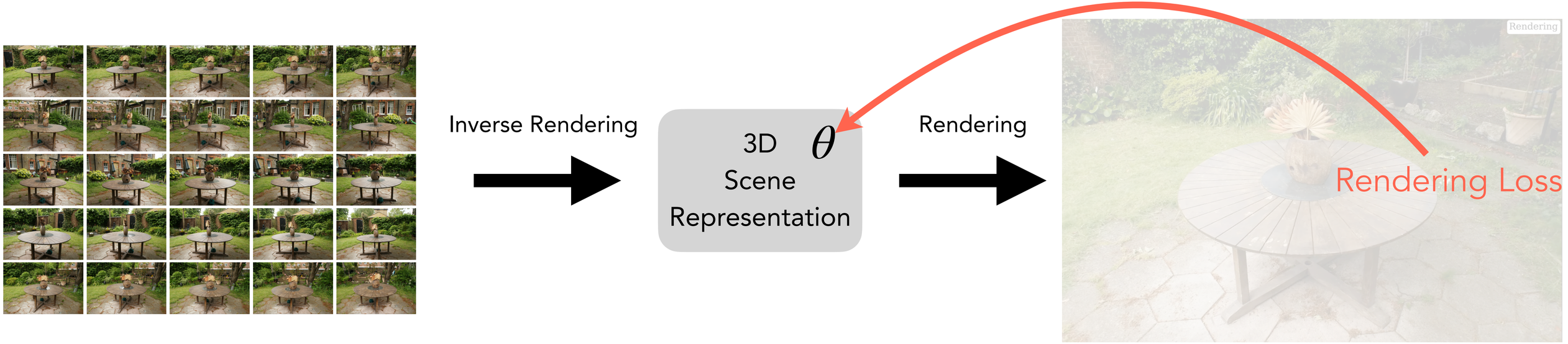


Inverse Rendering

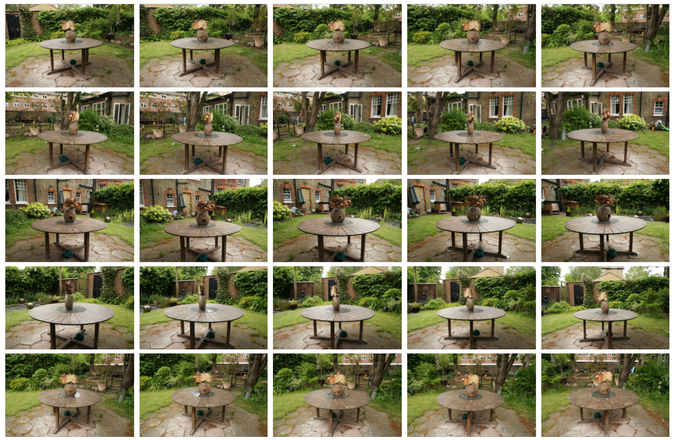
3D Scene Representation θ

Rendering

Approach: “Optimization-based inverse rendering” or “Analysis-by-synthesis” or “Render-and-compare”



Neural Field representations for "computer vision as inverse rendering" tasks

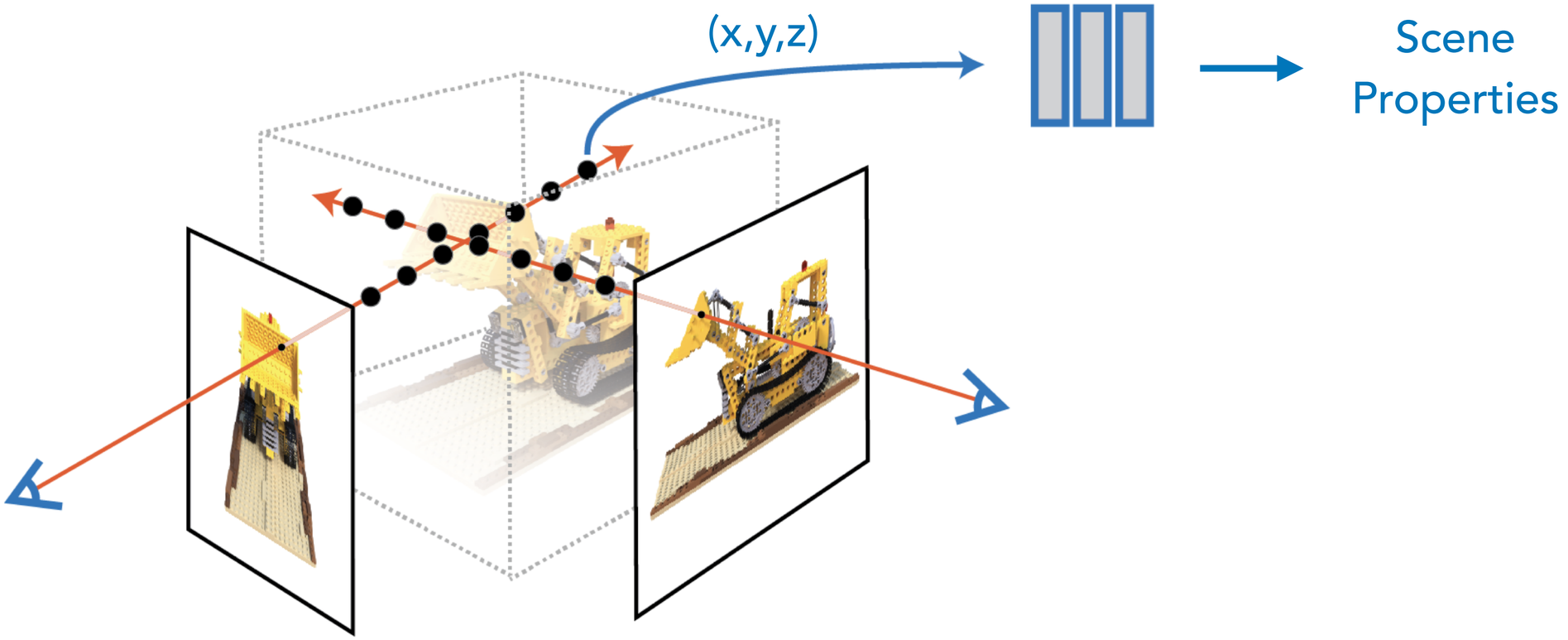


Inverse Rendering

Neural Field θ

Rendering

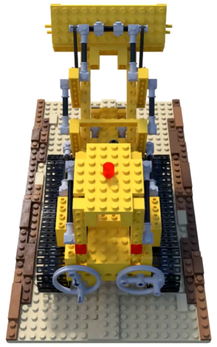
Neural Fields use MLP to represent scene as continuous function

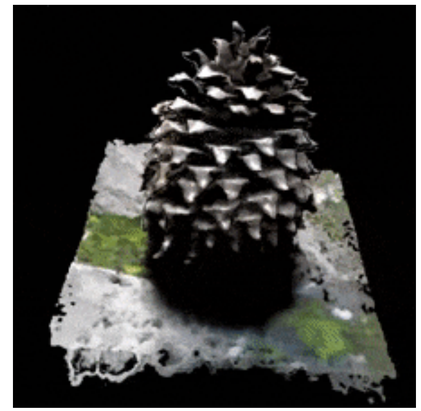


What should the Neural Field store at any point?

What should the Neural Field store at any point?

Depends on what scene parameters you want to vary…



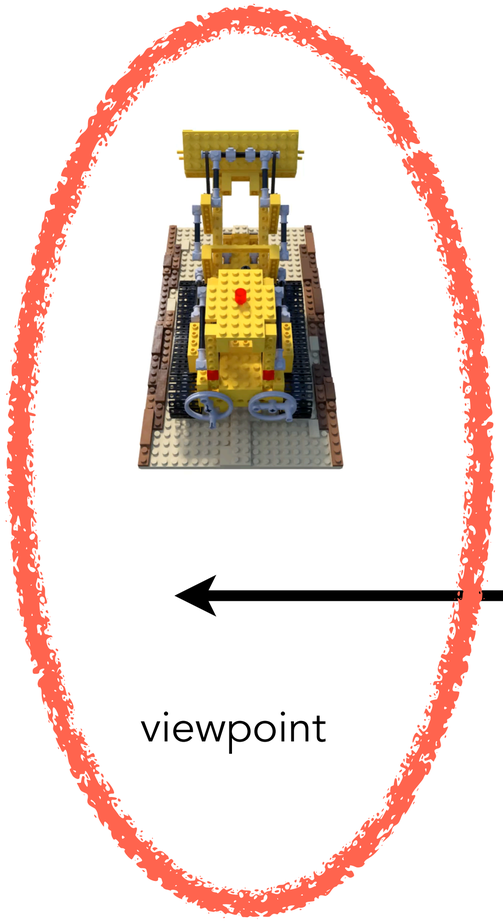


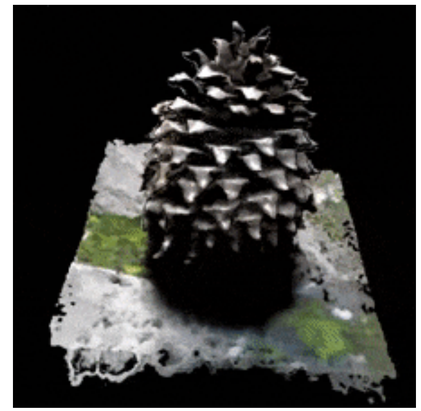
viewpoint

viewpoint,
lighting,
material

Varying just the viewpoint

NeRF stores volume density and outgoing/emitted radiance





viewpoint

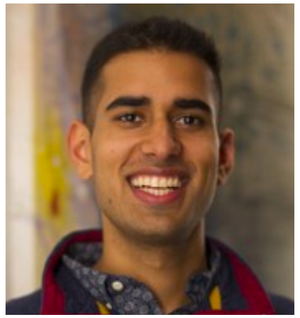
viewpoint,
lighting,
material

NeRF:
Representing Scenes as Neural Radiance Fields for View Synthesis

Ben Mildenhall*

UC Berkeley

Pratul Srinivasan*



UC Berkeley

Matt Tancik*

UC Berkeley

Jon Barron

Google Research

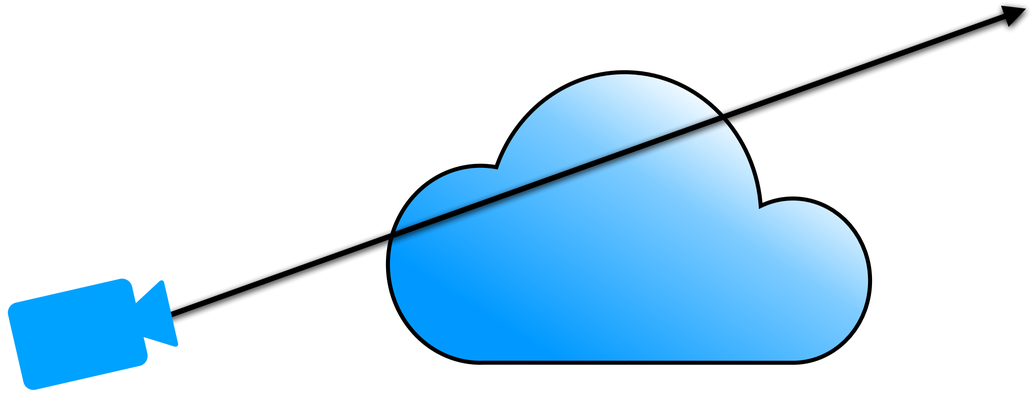
Ravi Ramamoorthi

UC San Diego

Ren Ng

UC Berkeley

Volumetric formulation for NeRF



Scene is a cloud of tiny colored particles

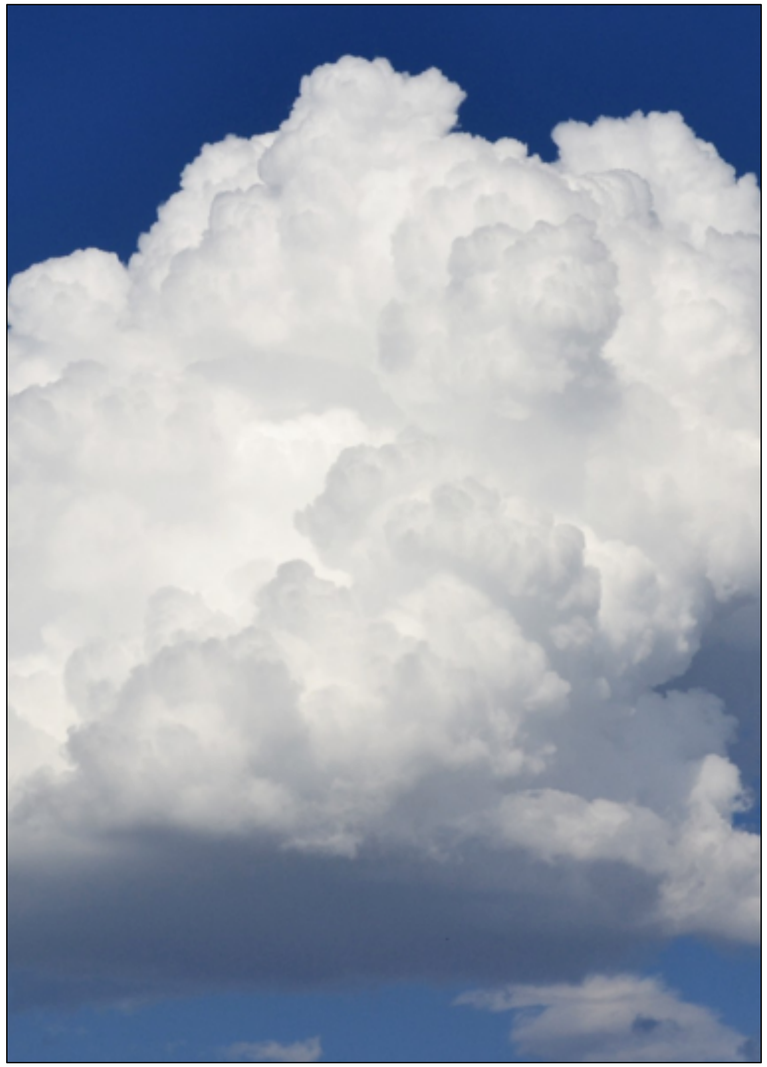
Max and Chen 2010, *Local and Global Illumination in the Volume Rendering Integral*

Volumetric rendering

Absorption

http://commons.wikimedia.org

Scattering



http://coclouds.com

Emission

http://wikipedia.org

Slide credit: Novak et al 2018, *Monte Carlo methods for physically based volume rendering*

Volumetric formulation for NeRF

Absorption

http://commons.wikimedia.org

Scattering

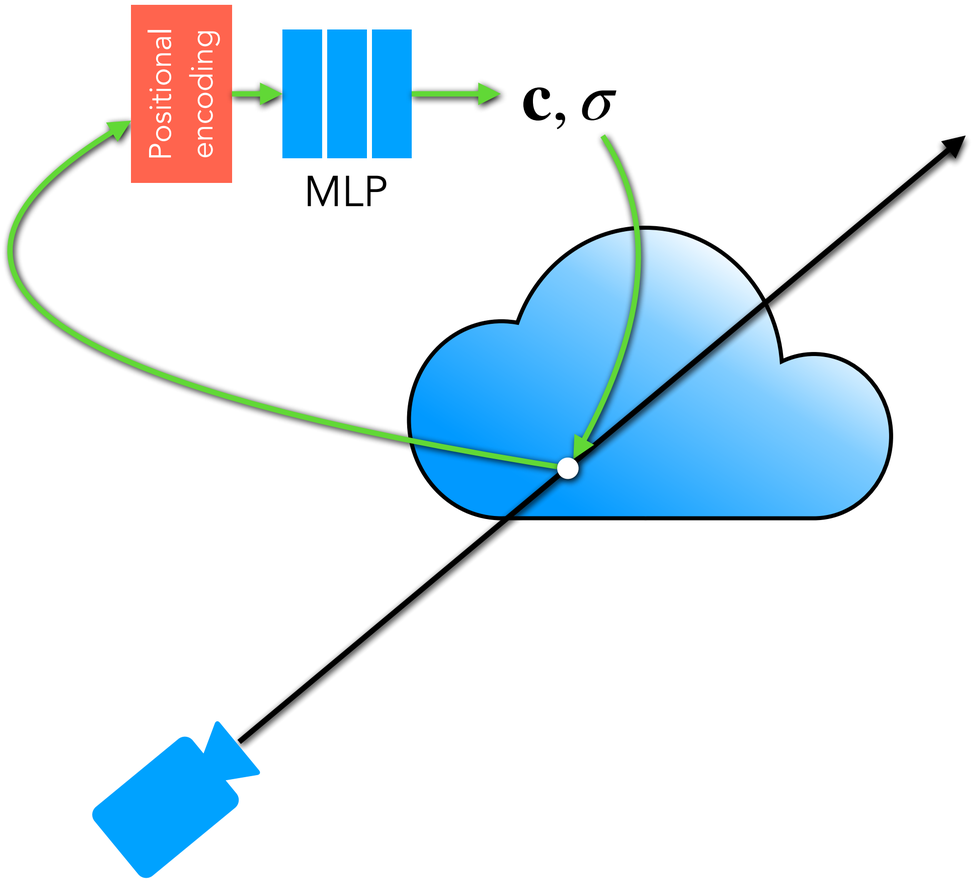
http://coclouds.com

Emission

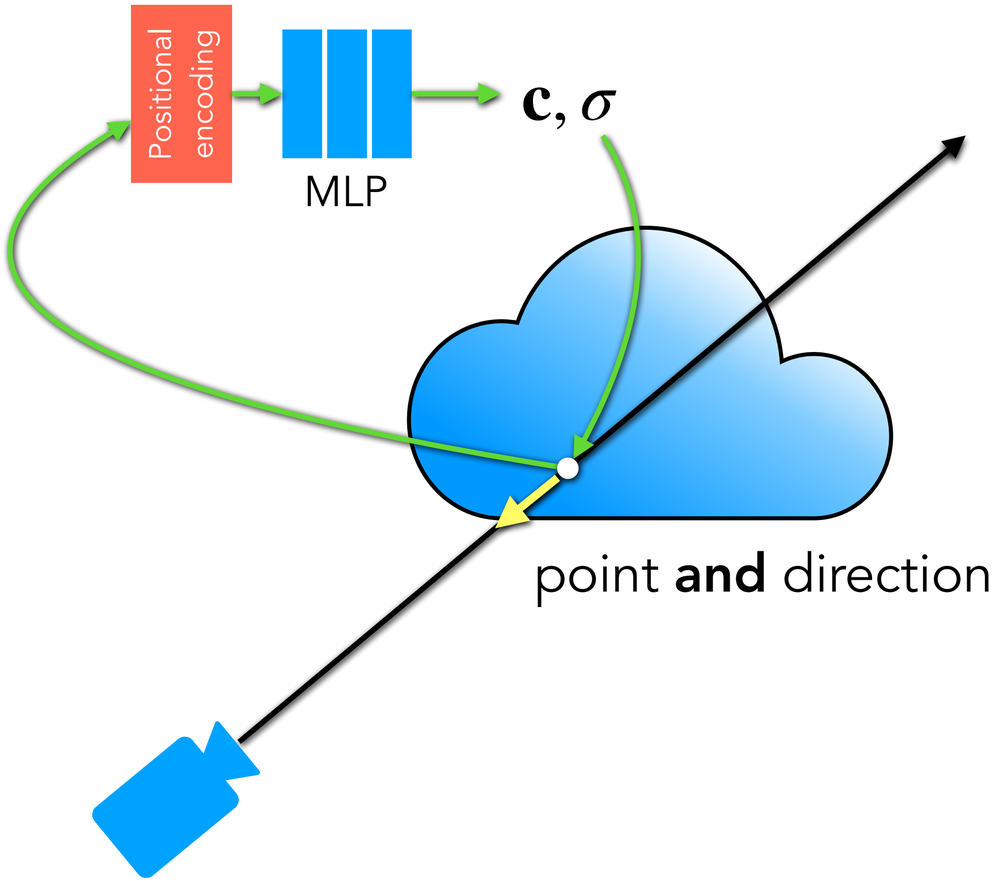
http://wikipedia.org

Slide credit: Novak et al 2018, *Monte Carlo methods for physically based volume rendering*

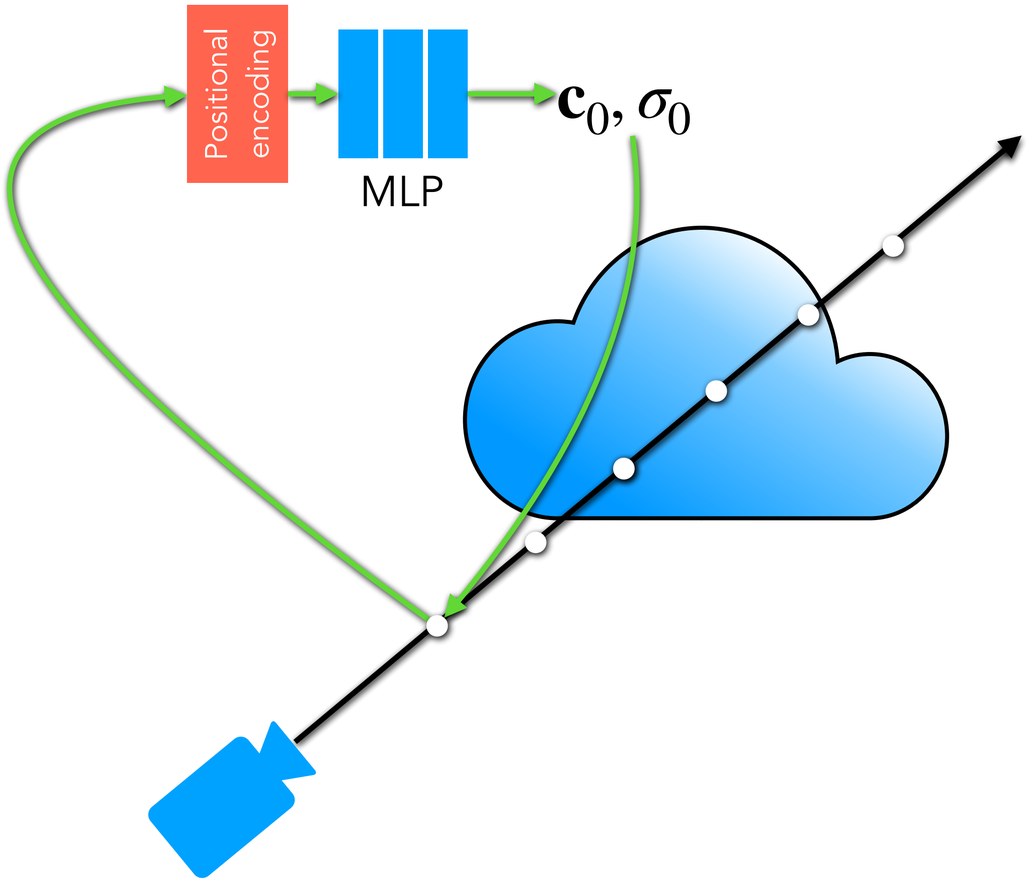
NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



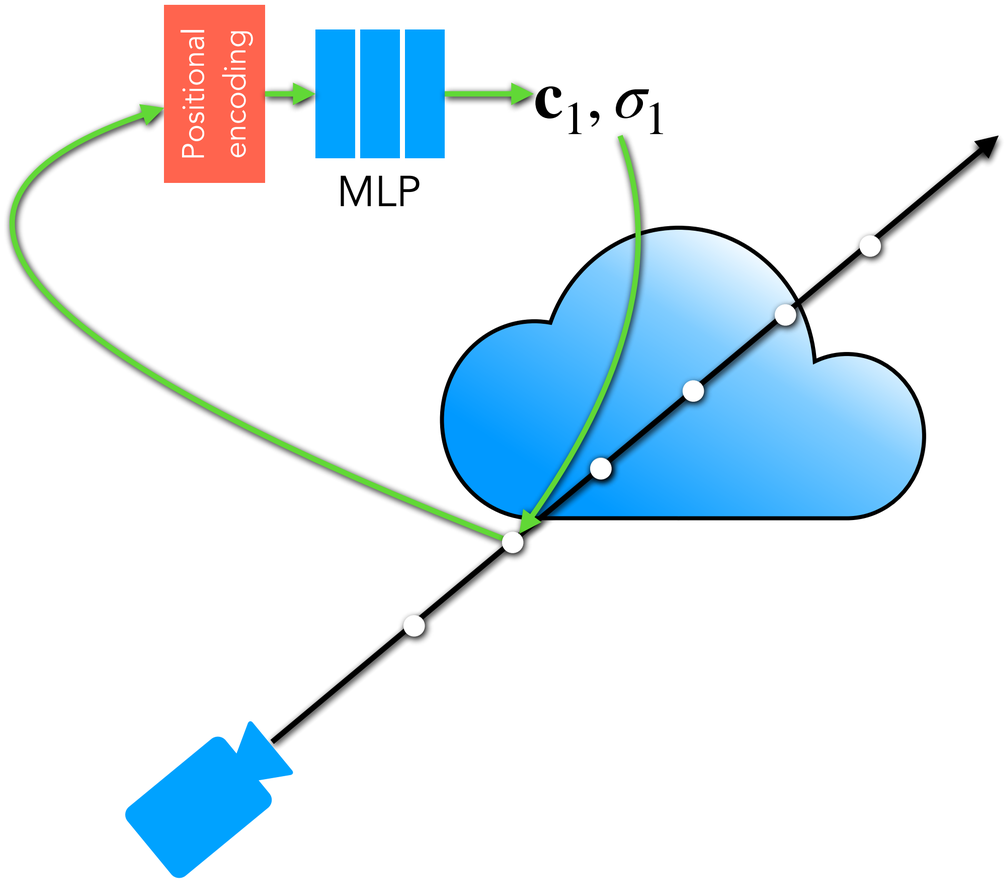
NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



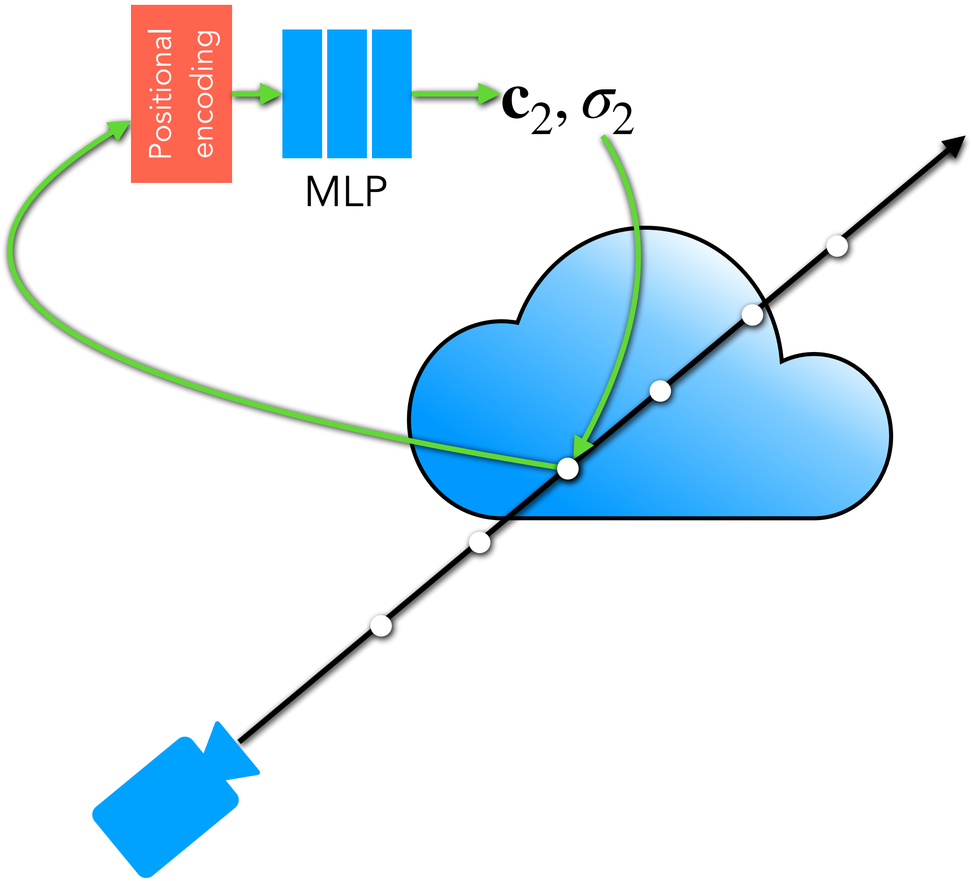
NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



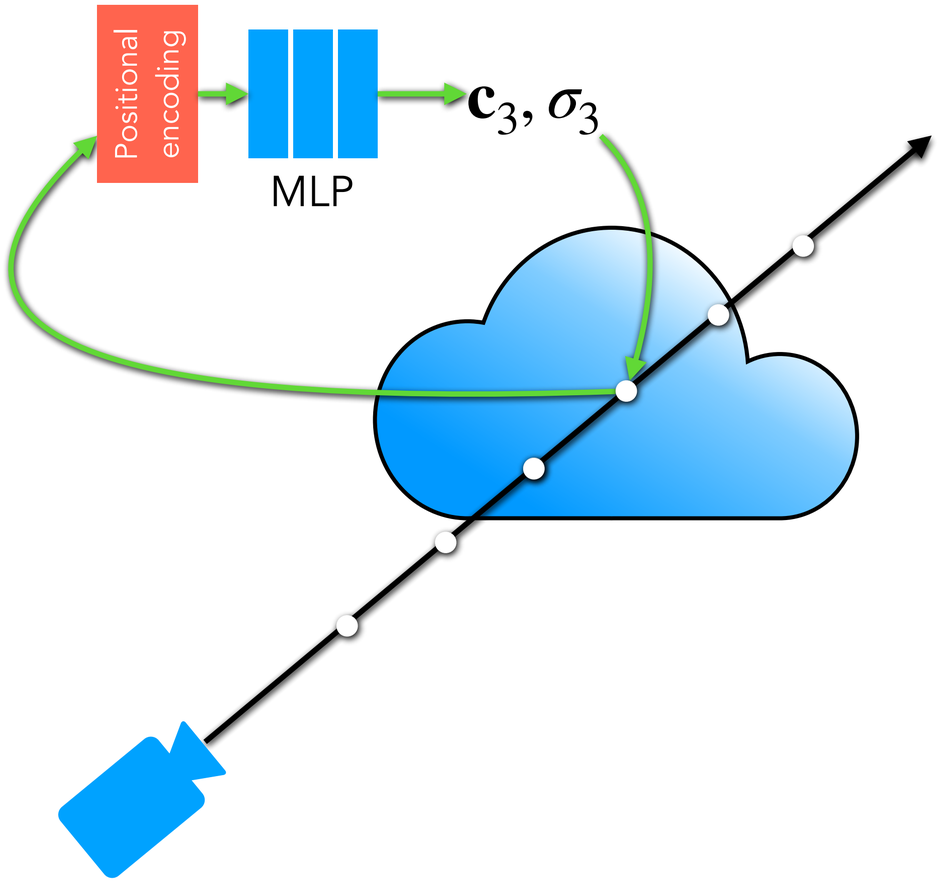
NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



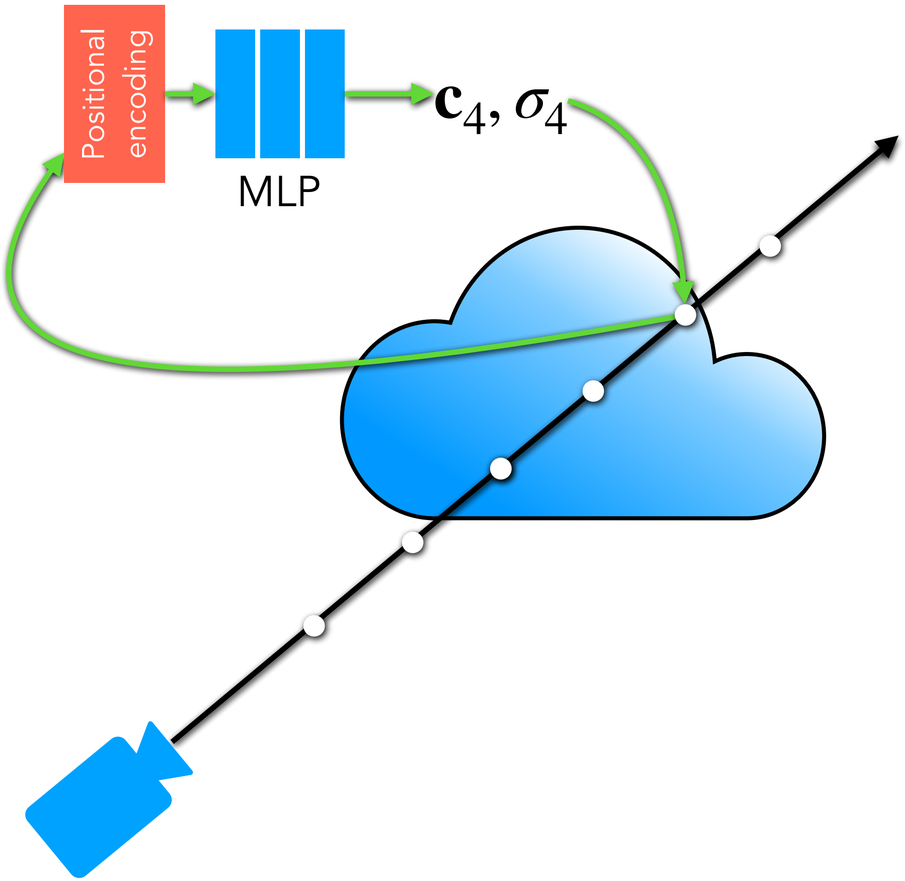
NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



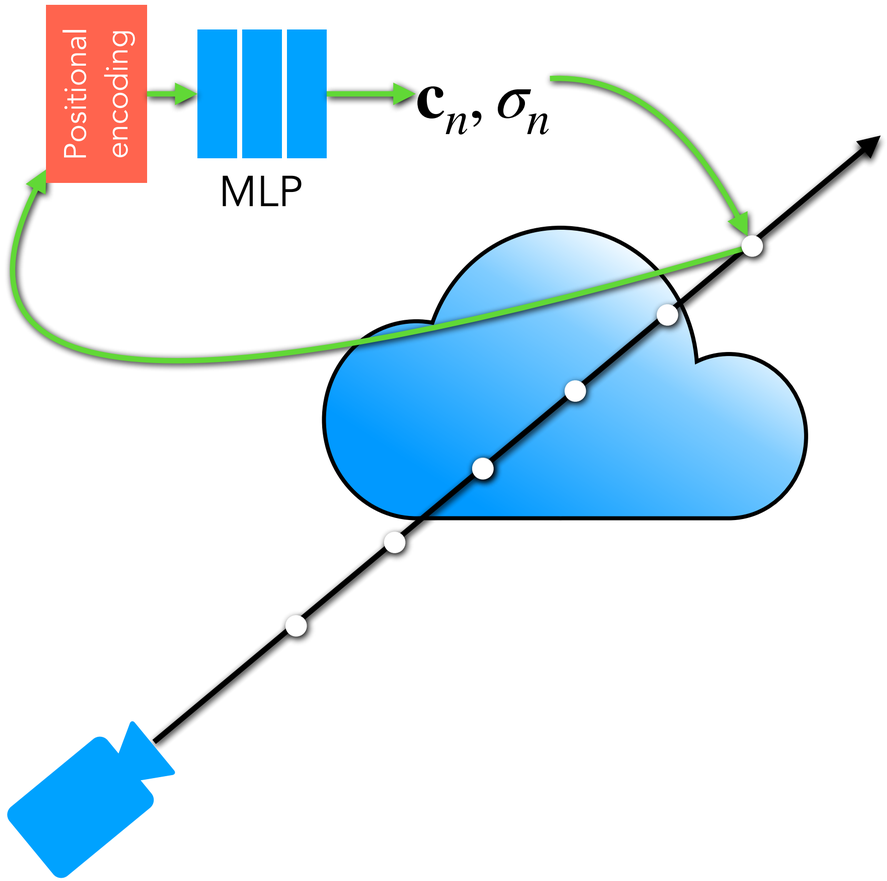
NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



NeRF stores the values of $\mathbf{c}, \sigma$ at each point in space



Volume rendering integral estimate

Rendering model for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

$$\mathbf{c} \approx \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

colors

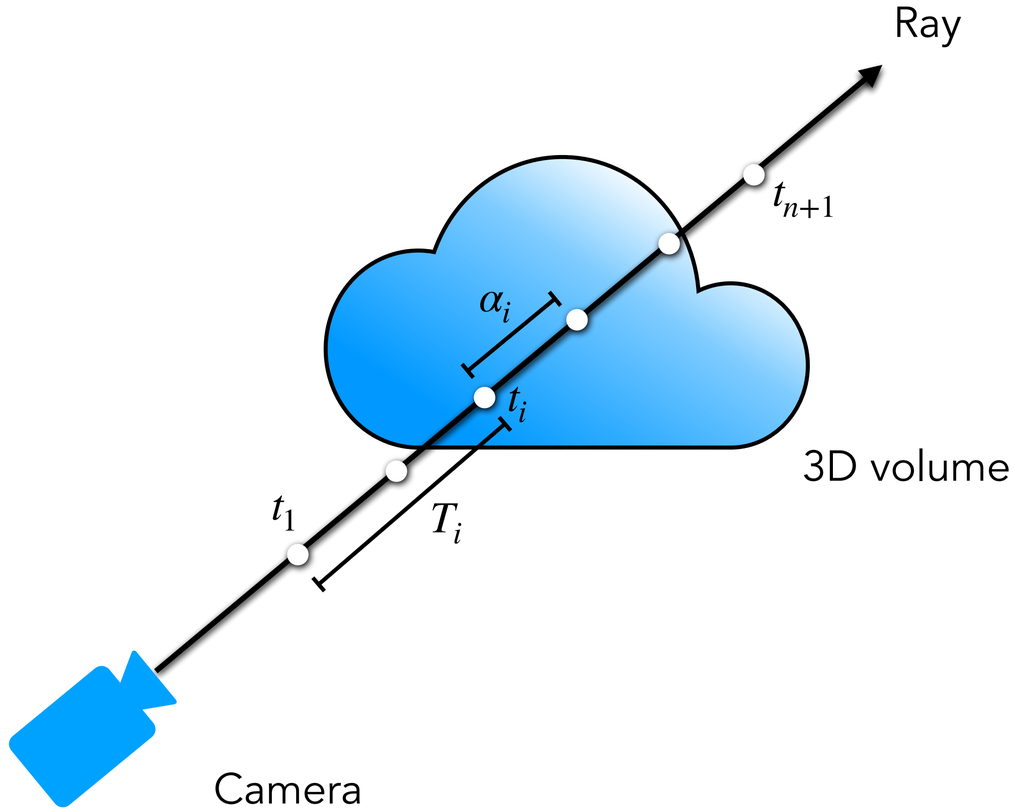
weights

How much light is blocked earlier along ray:

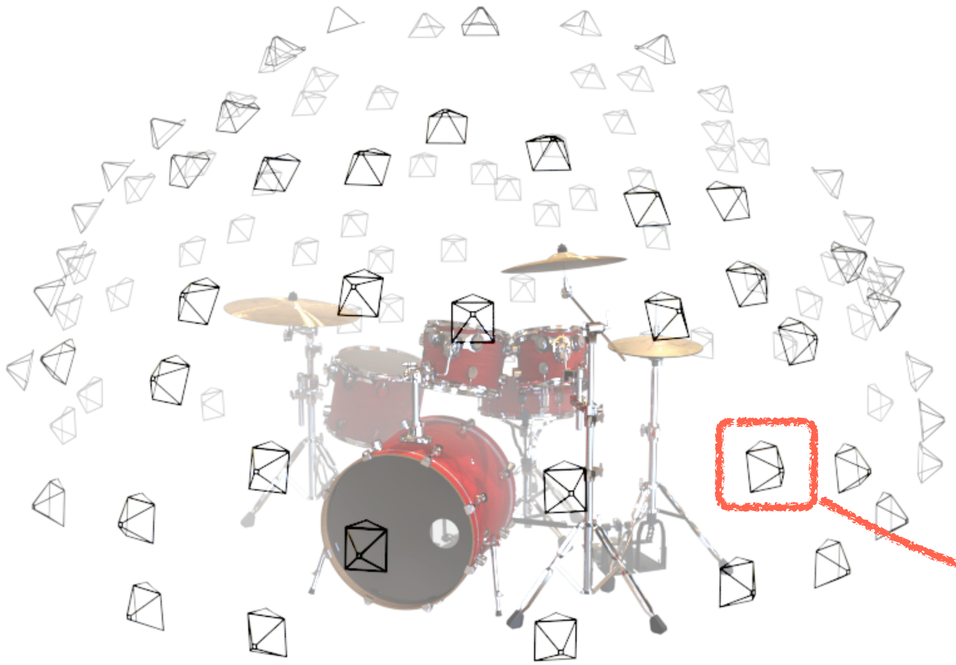
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

How much light is contributed by ray segment *i*:

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$



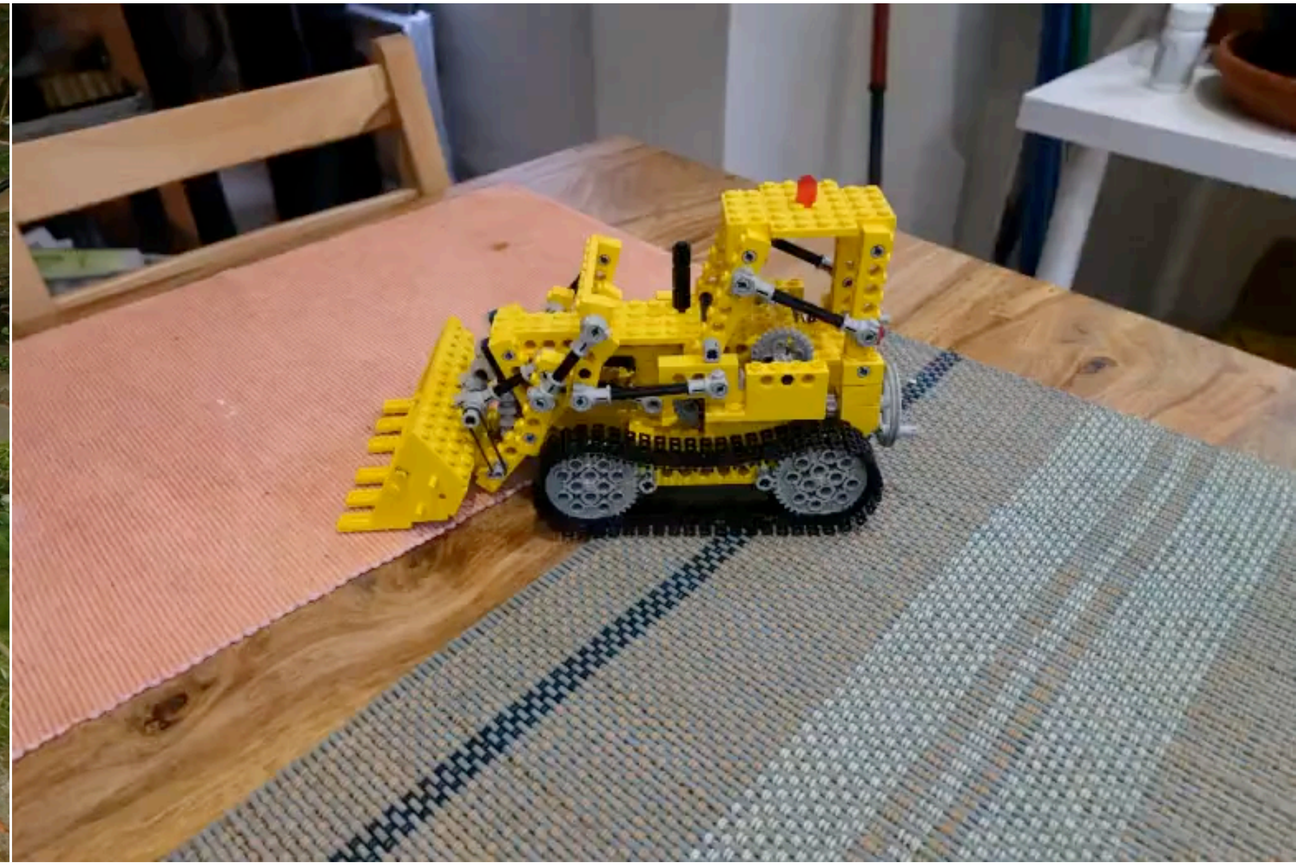
Train network using gradient descent to reproduce all input views of scene



Volume rendering of MLP colors/densities

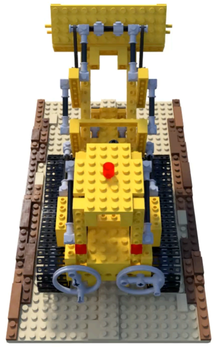
Ground truth image

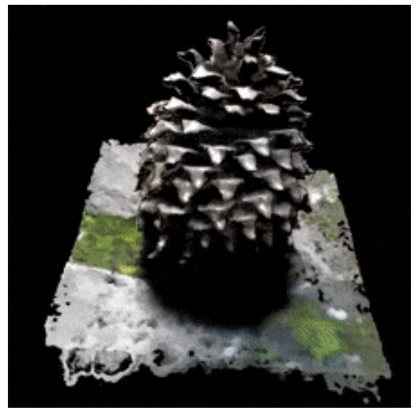
$$\nabla \left\| \quad - \quad \right\|^2$$



Barron et al 2021, *Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields*

What should the Neural Field store at any point?



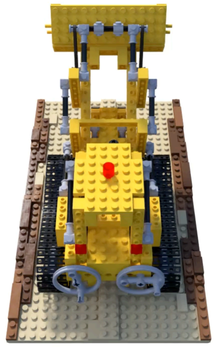


viewpoint

viewpoint,
lighting,
material

Vary viewpoint, lighting, materials

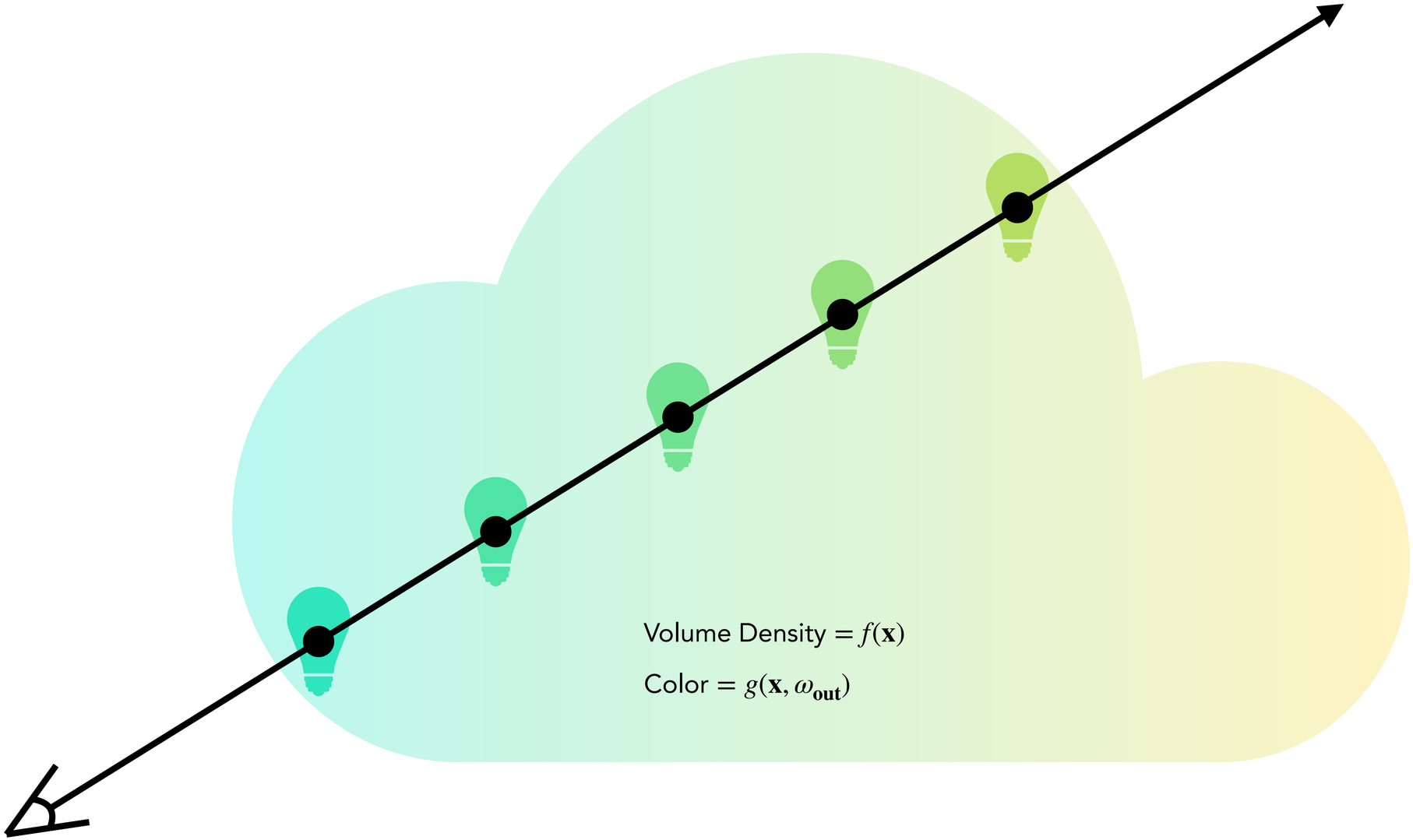
Neural Reflectance Field stores volume density, local geometry orientation (normal vectors), BRDF parameters



viewpoint

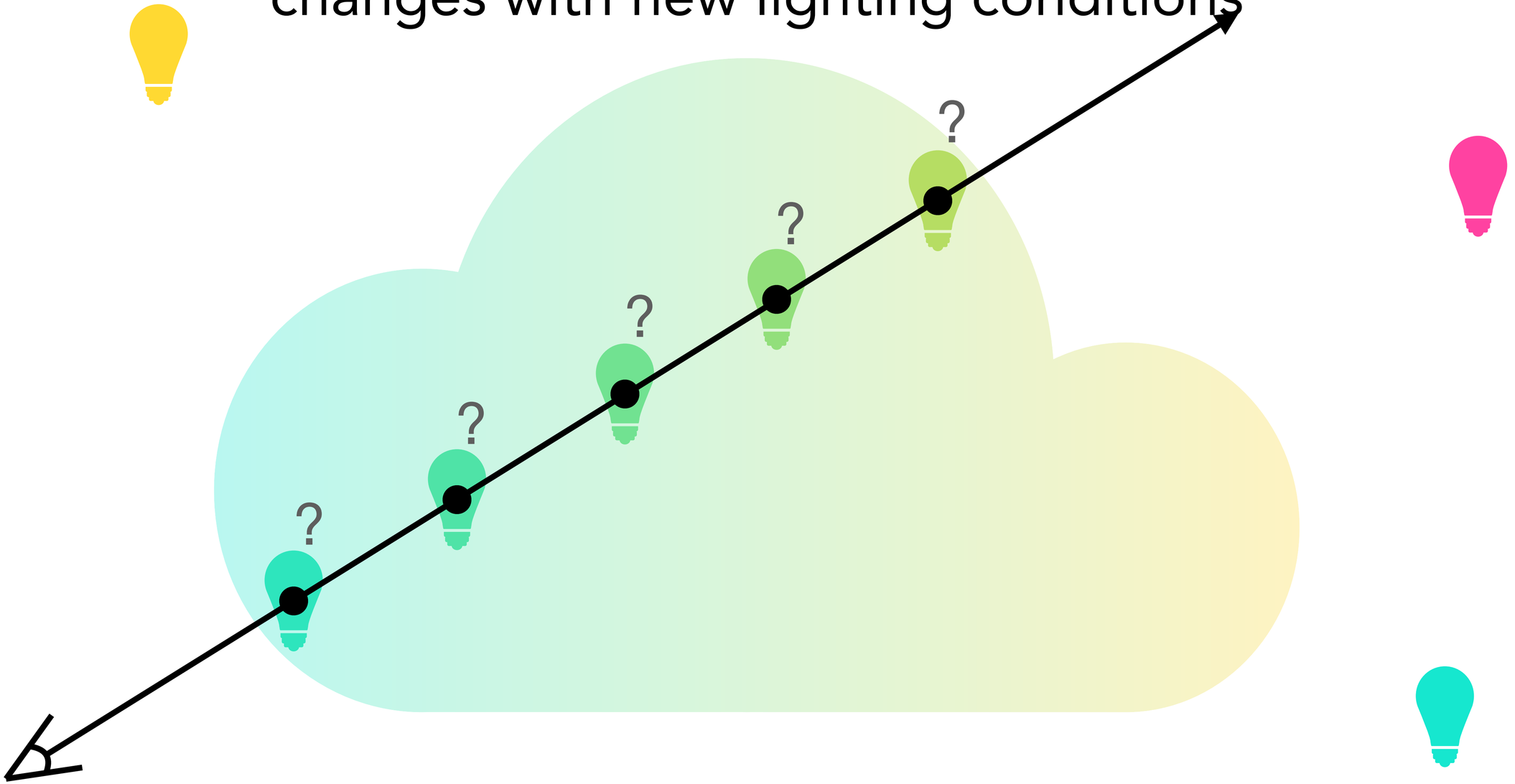
viewpoint,
lighting,
material

NeRF represents a volume of particles that emit light

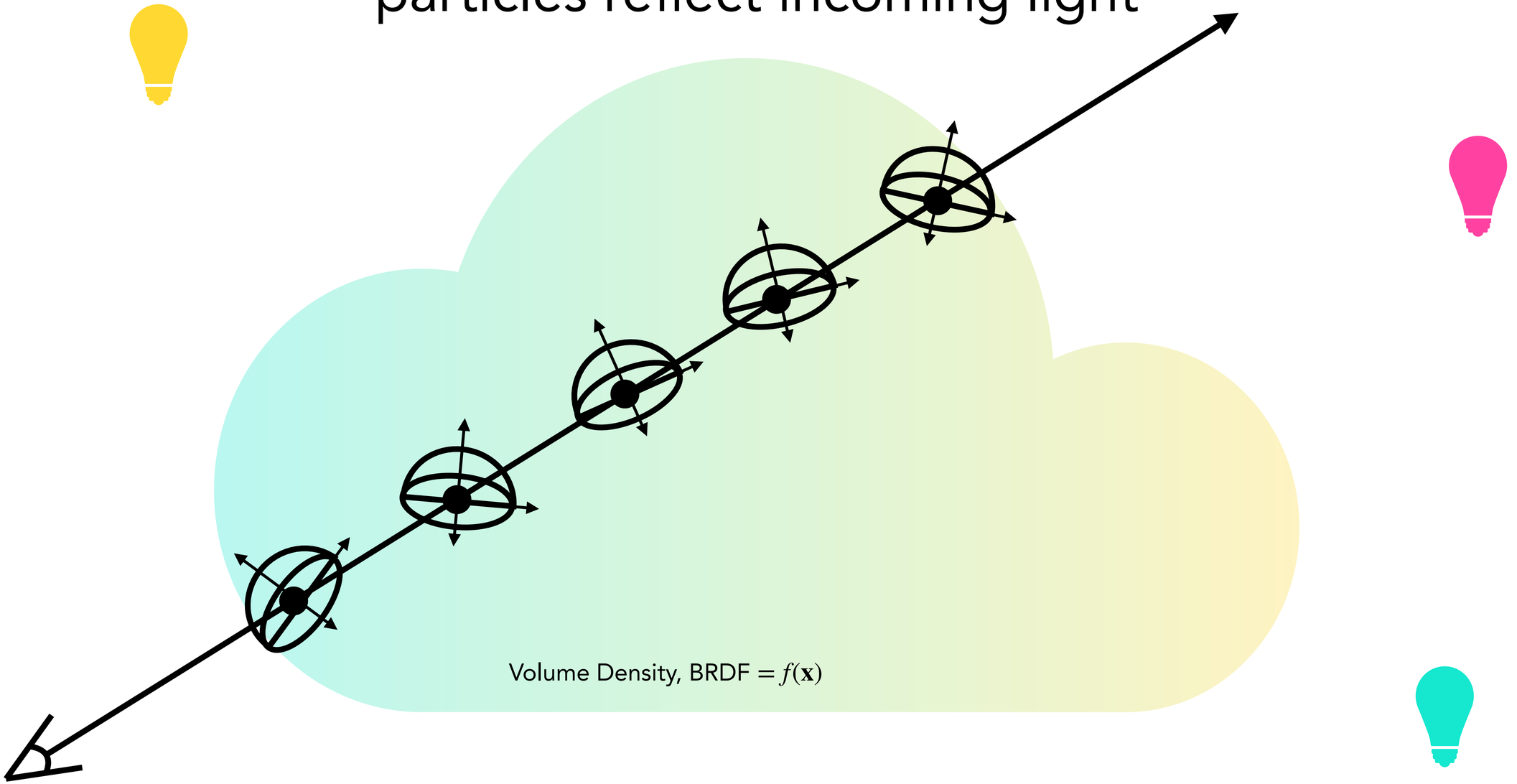


NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.
Ben Mildenhall*, Pratul Srinivasan*, Matt Tancik*, Jon Barron, Ravi Ramamoorthi, Ren Ng. ECCV 2020.

But doesn’t let us simulate how light leaving a point changes with new lighting conditions

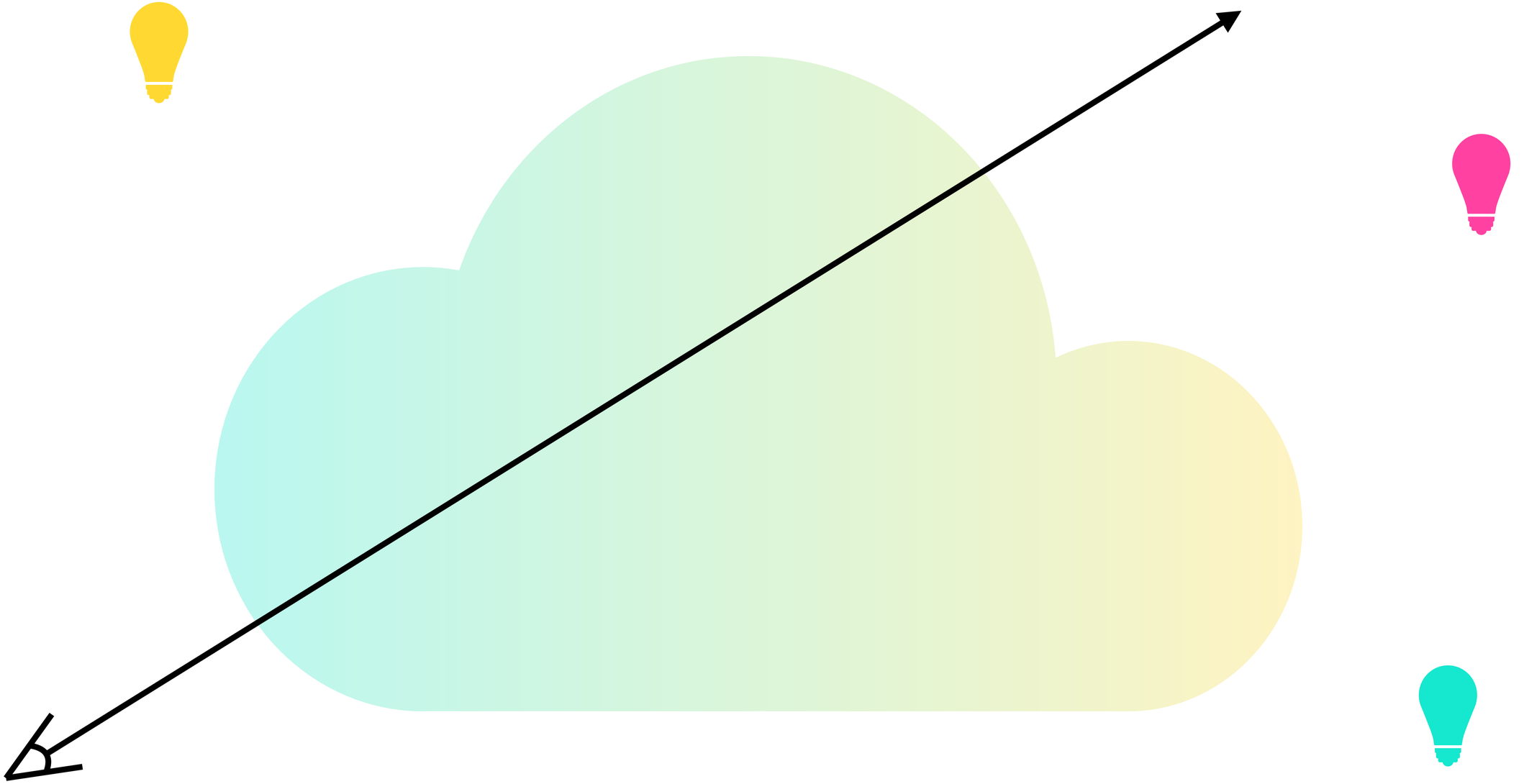


First step: replace emitted light with BRDFs that describe how particles reflect incoming light

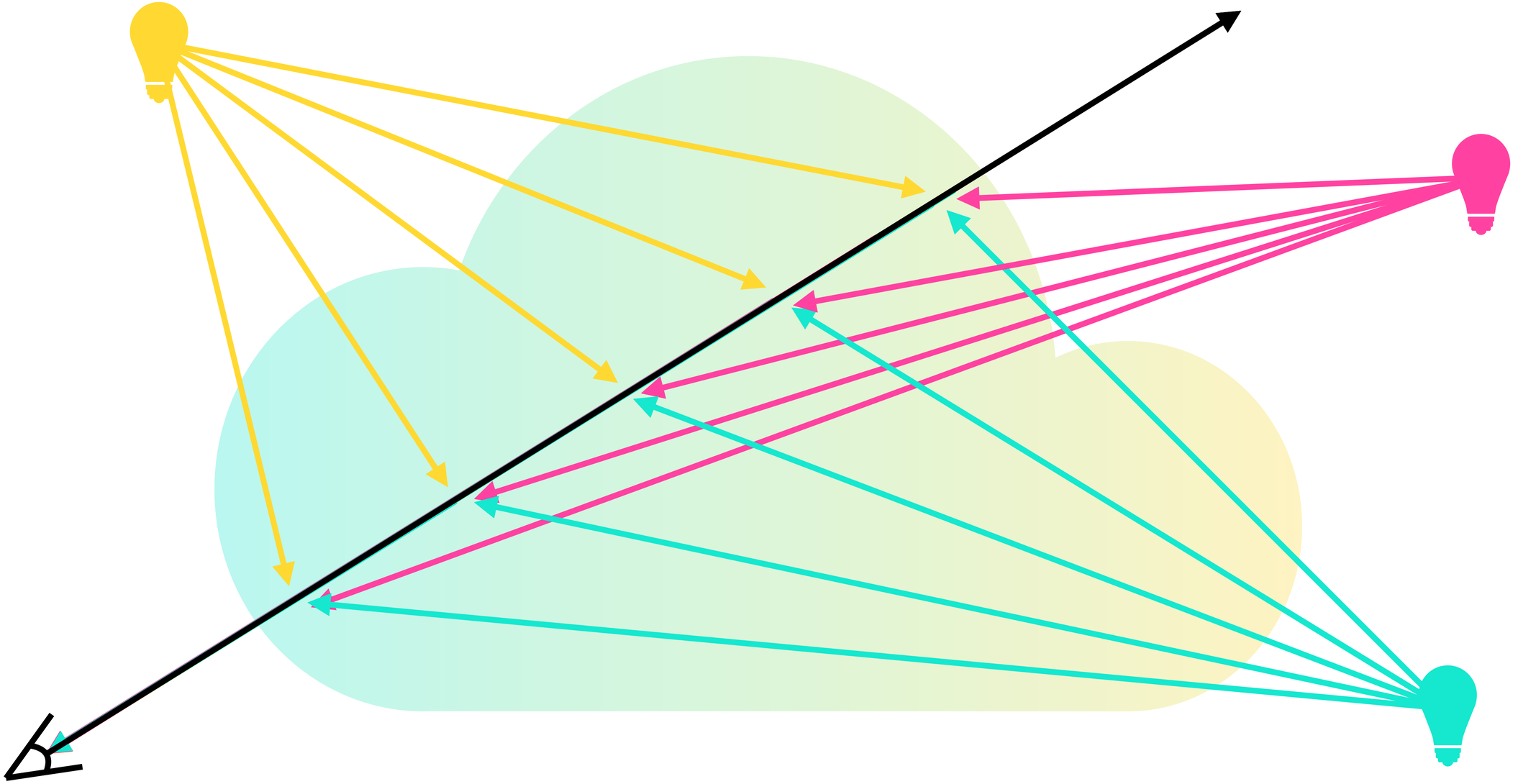


Neural Reflectance Fields for Appearance Acquisition.
Sai Bi*, Zexiang Xu*, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David Kriegman, Ravi Ramamoorthi. arXiv 2020.

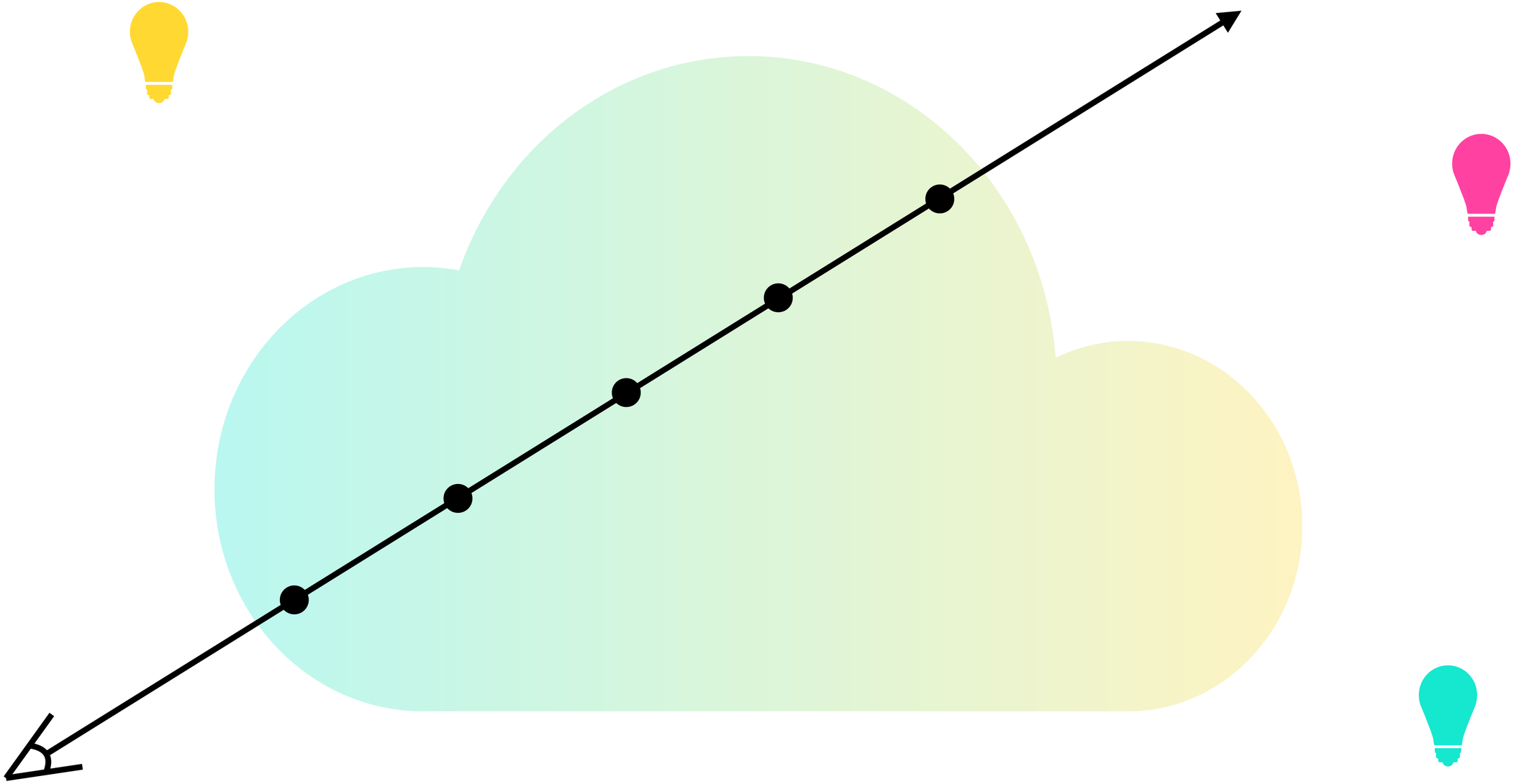
Rendering a neural volumetric field with direct lighting



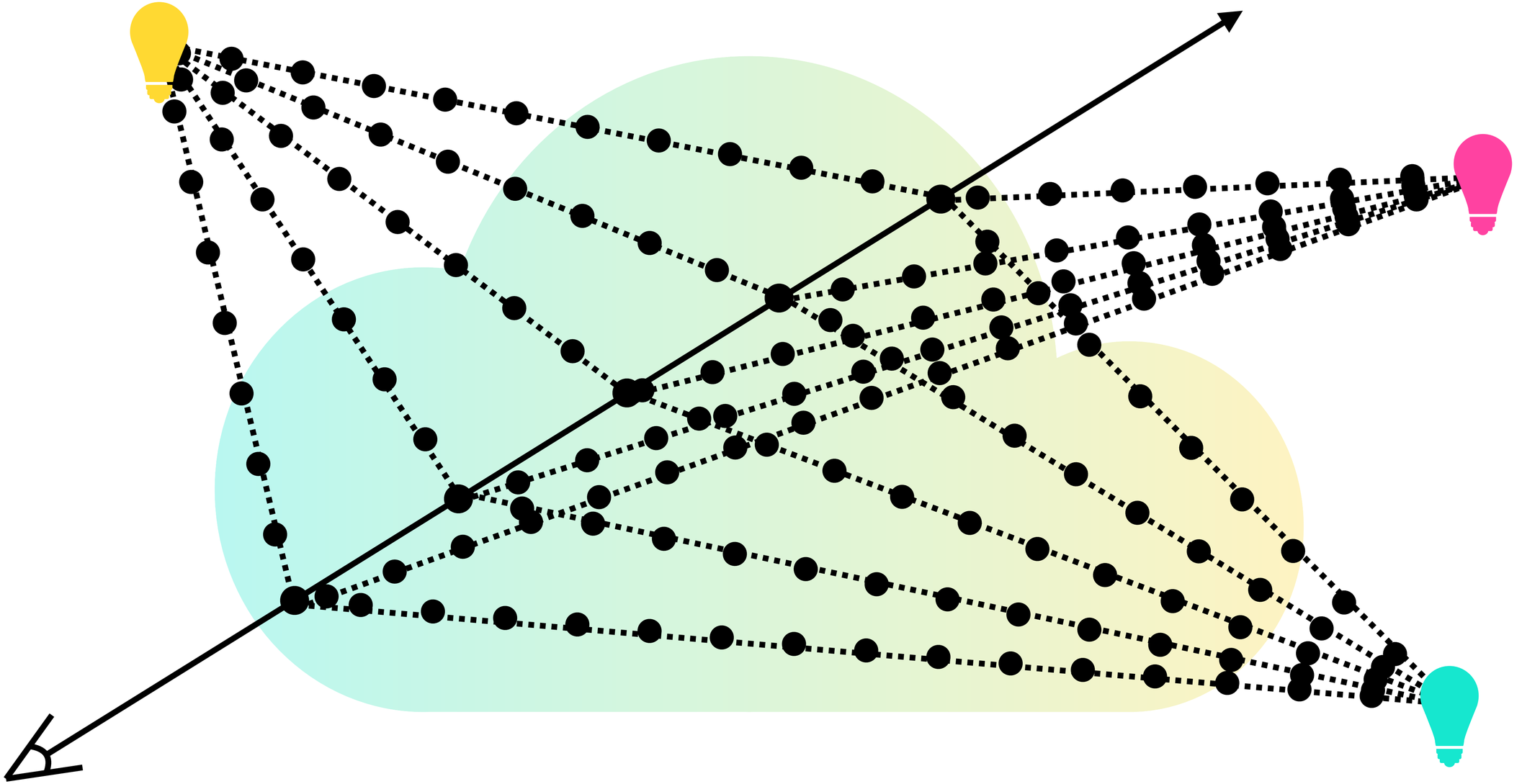
Rendering a neural volumetric field with direct lighting



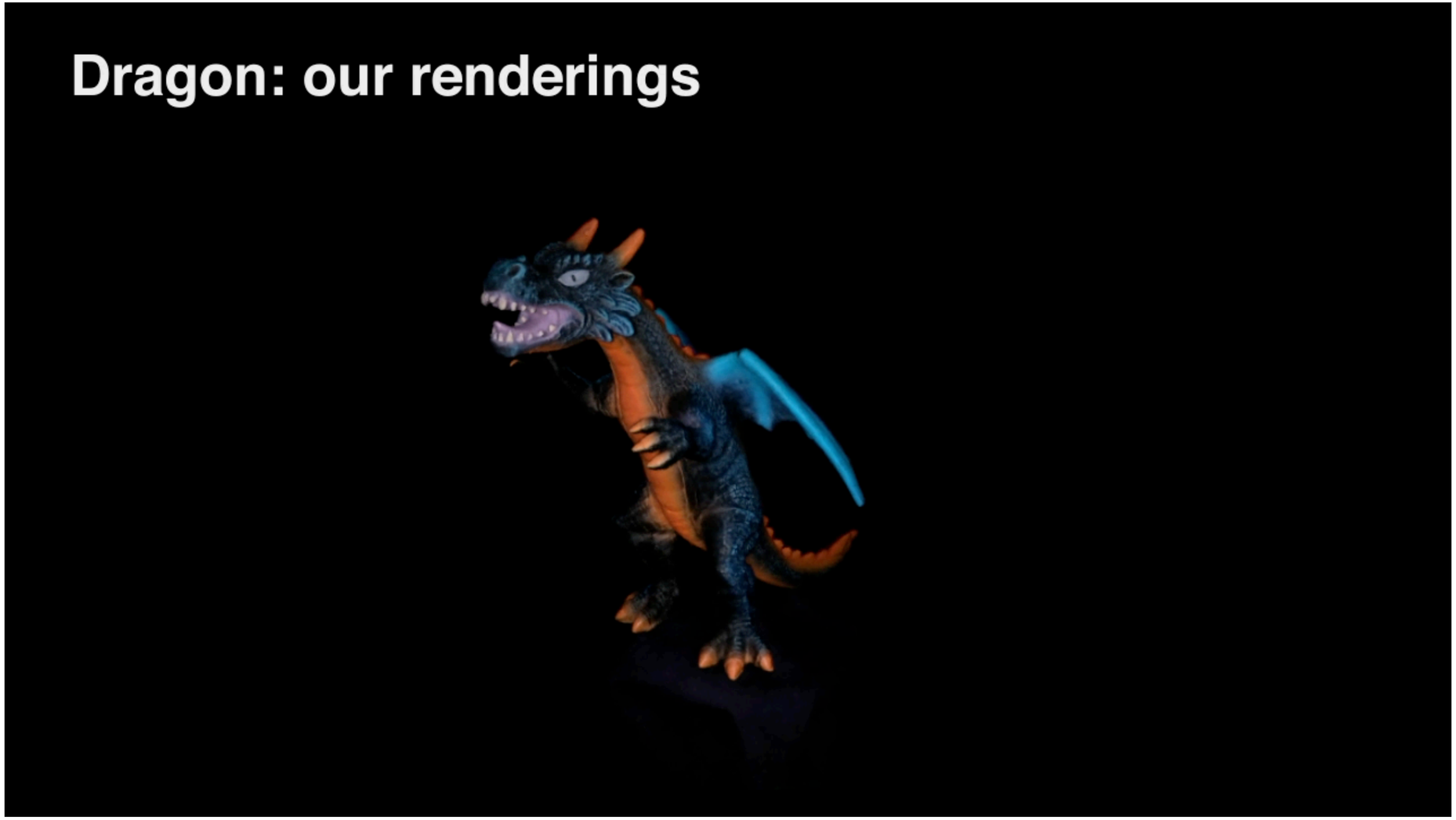
Rendering a neural volumetric field with direct lighting



Rendering a neural volumetric field with direct lighting



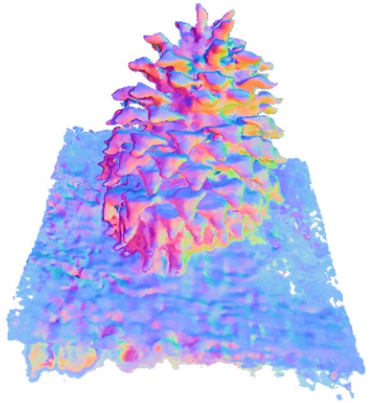
Neural Reflectance Fields enable relighting



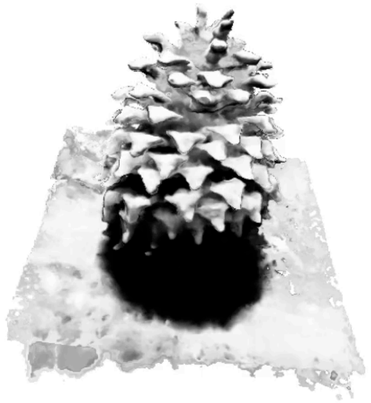
Neural Reflectance Fields for Appearance Acquisition.
Sai Bi*, Zexiang Xu*, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David Kriegman, Ravi Ramamoorthi. arXiv 2020.

NeRFactor makes relighting more efficient

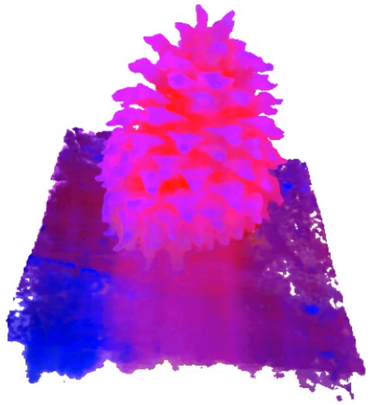
Normals



Visibility



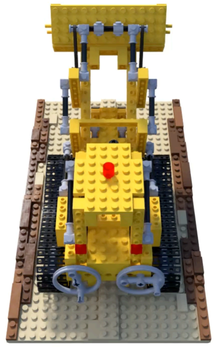
BRDF

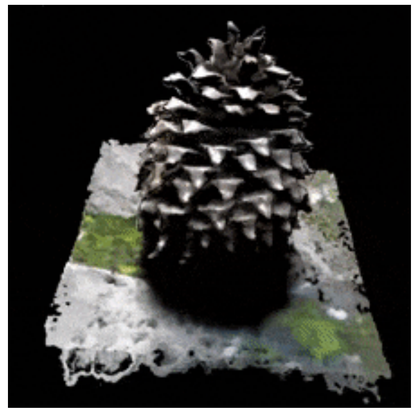


Albedo

NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination.
Xiuming Zhang, Pratul Srinivasan, Boyang Deng, Paul Debevec, William Freeman, Jon Barron. SIGGRAPH Asia 2021.

What should the Neural Field store at any point?



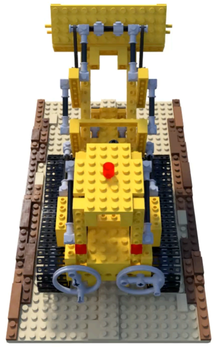


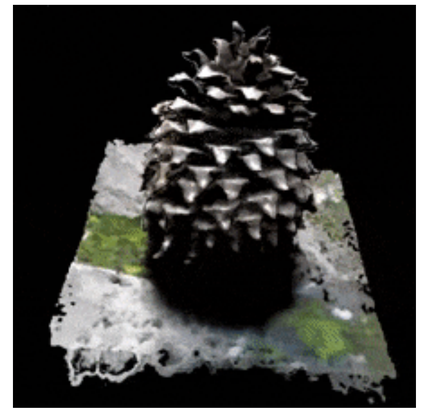
viewpoint

viewpoint,
lighting,
material

What should the Neural Field store at any point?

Spectrum of simulation vs. “caching”



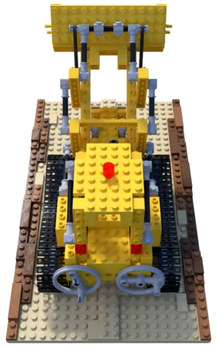


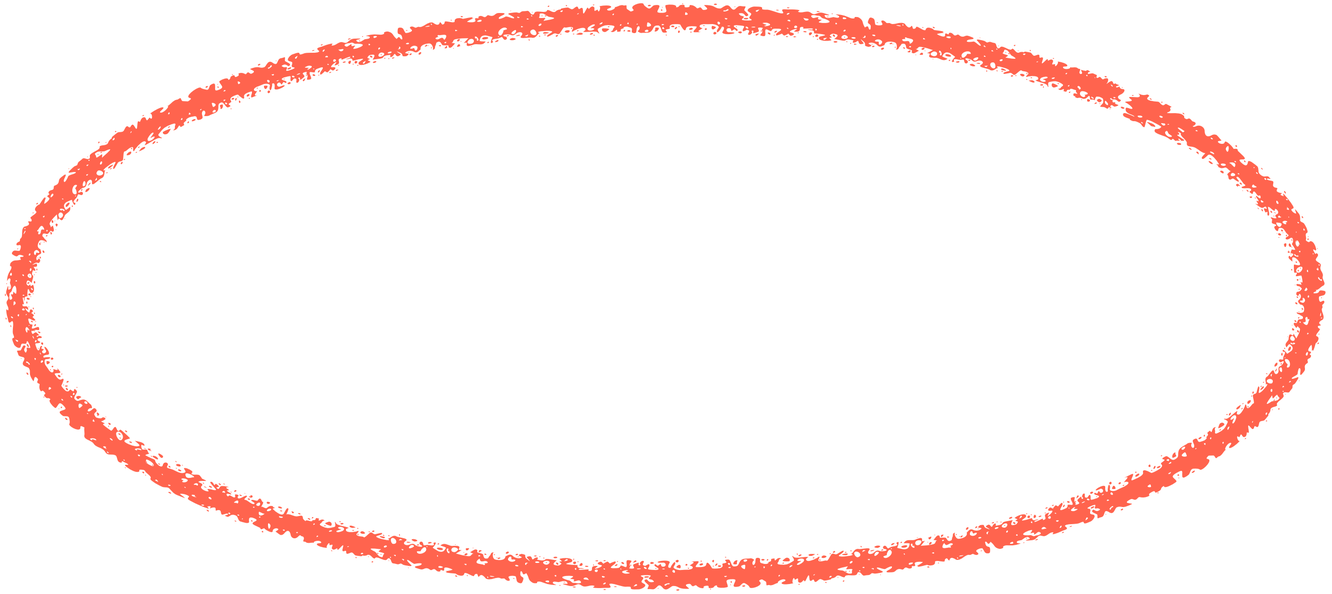
caching

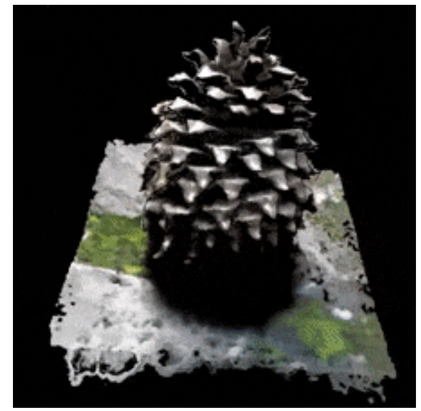
simulation

What should the Neural Field store at any point?

What lies in between these extremes?







viewpoint

viewpoint,
lighting,
material

Google Research

Harvard John A. Paulson School of Engineering and Applied Sciences

Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields


Dor Verbin


Peter Hedman


Ben Mildenhall


Todd Zickler


Jonathan T. Barron


Pratul P. Srinivasan

dorverbin.github.io/refnerf


Ours

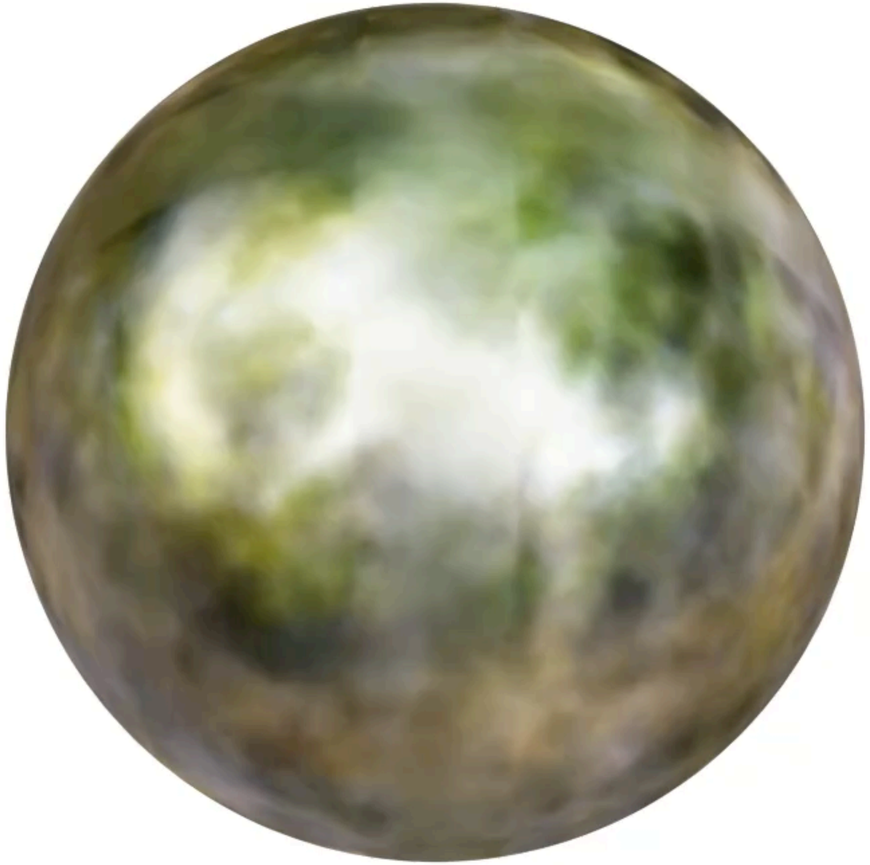
Why does NeRF fail to represent shiny scenes?

Input

Why does NeRF fail to represent shiny scenes?

Input

NeRF

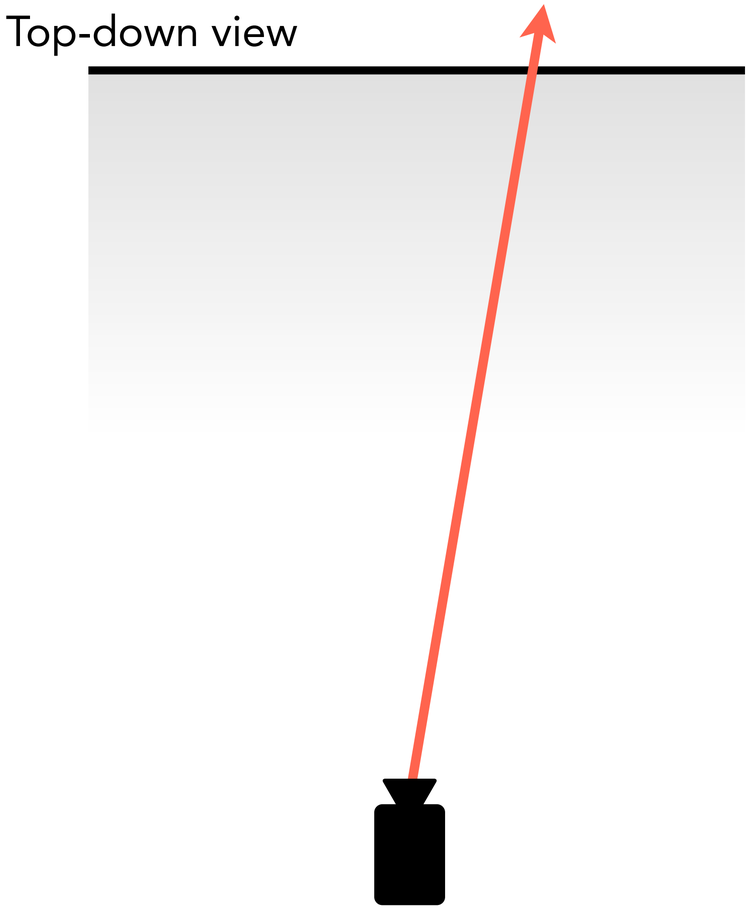


Why does NeRF fail to represent shiny scenes?

Input

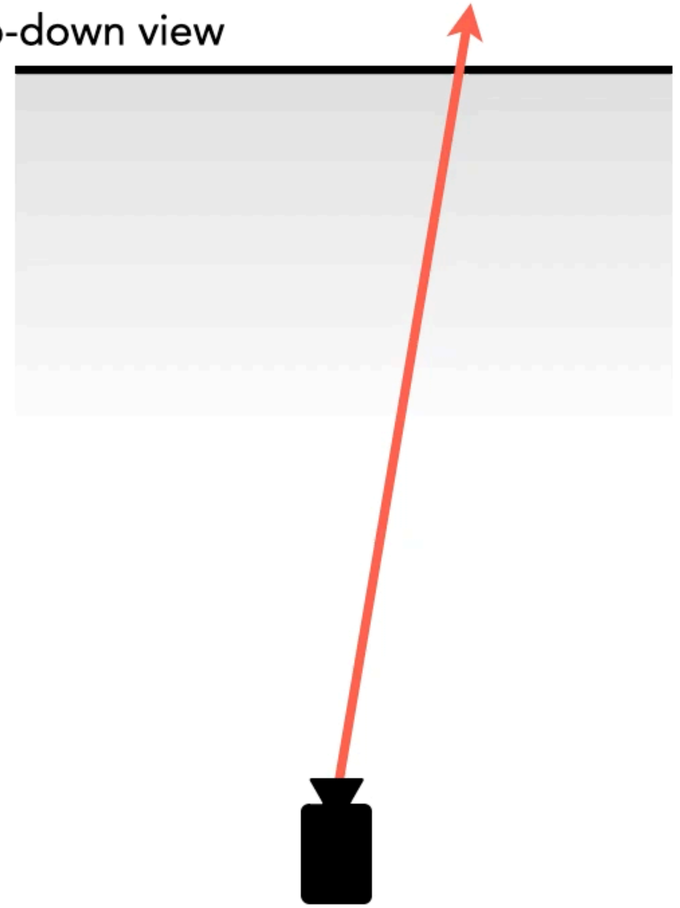
Our model

Specular reflections represented as glowing emitters in a “foggy” volume



Specular reflections represented as glowing emitters in a “foggy” volume

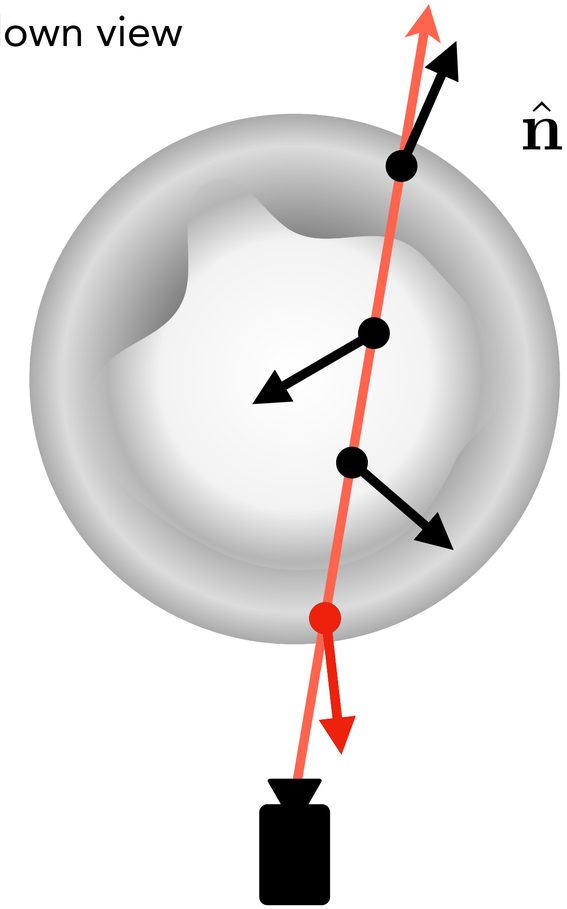
Top-down view



Back-to-front rendering

Visualizing “foggy” geometry with accumulated normal vectors

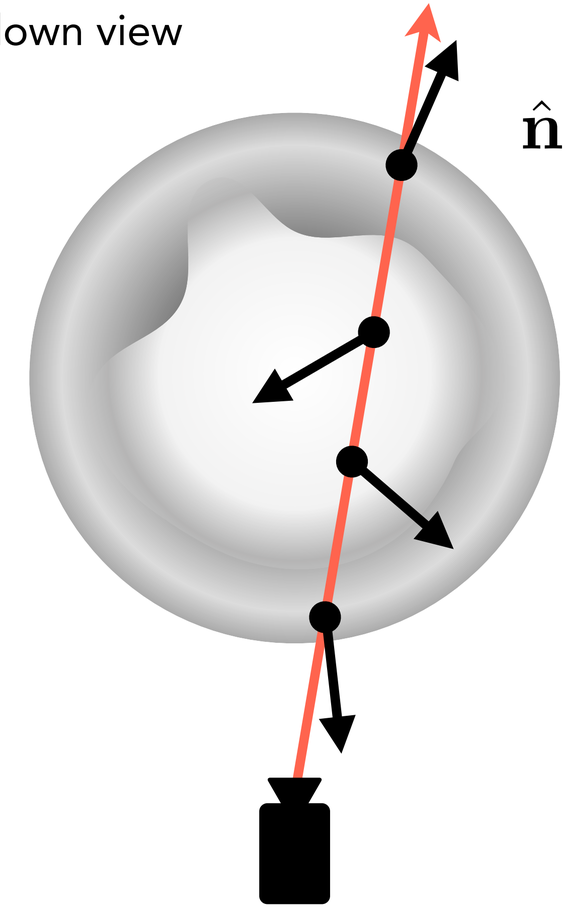
Top-down view



$$\hat{\mathbf{n}}(\mathbf{x}) = -\frac{\nabla\tau(\mathbf{x})}{\|\nabla\tau(\mathbf{x})\|}$$

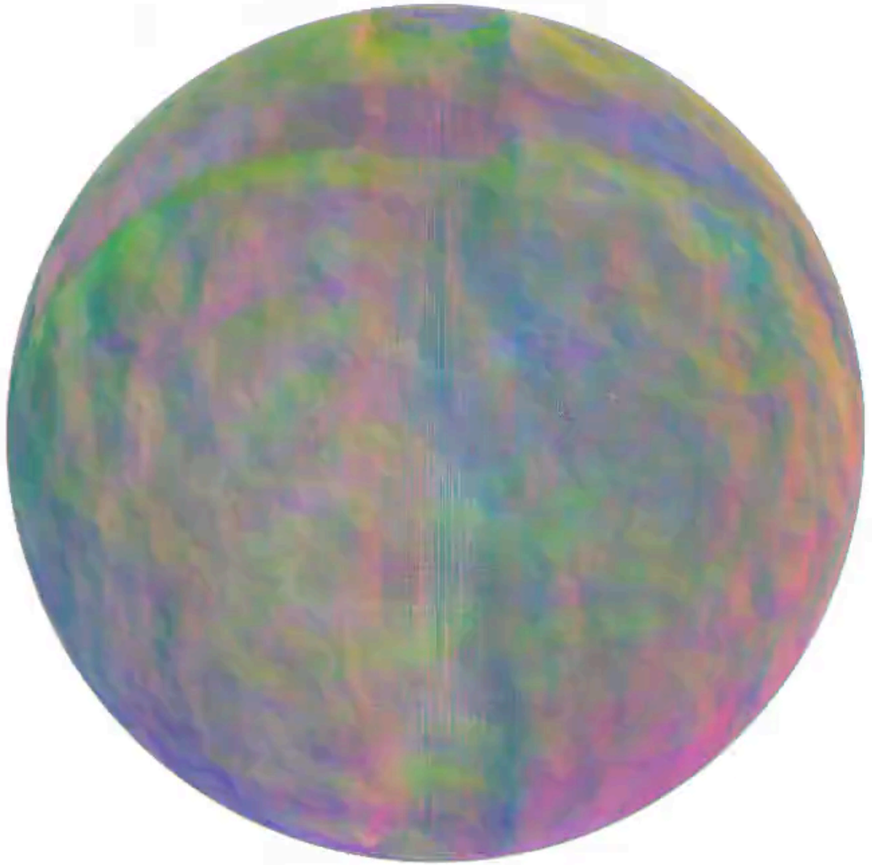
Visualizing “foggy” geometry with accumulated normal vectors

Top-down view



$$\hat{\mathbf{n}}(\mathbf{x}) = -\frac{\nabla\tau(\mathbf{x})}{\|\nabla\tau(\mathbf{x})\|}$$

NeRF



How to fix NeRF’s view-dependent appearance

1. Regularize NeRF to fix its foggy geometry and normals

2. Use fixed normals to improve view-dependent appearance

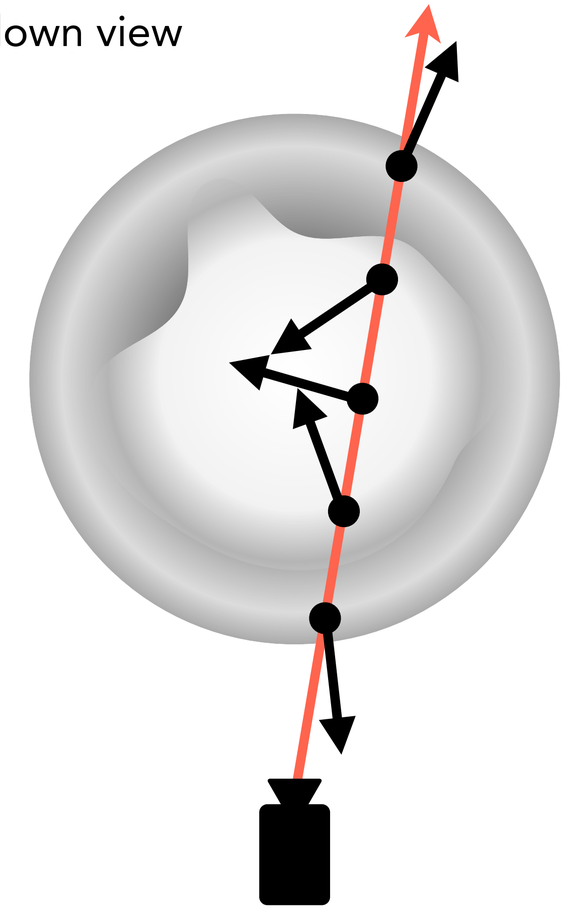
How to fix NeRF’s view-dependent appearance

1. Regularize NeRF to fix its foggy geometry and normals

2. Use fixed normals to improve view-dependent appearance

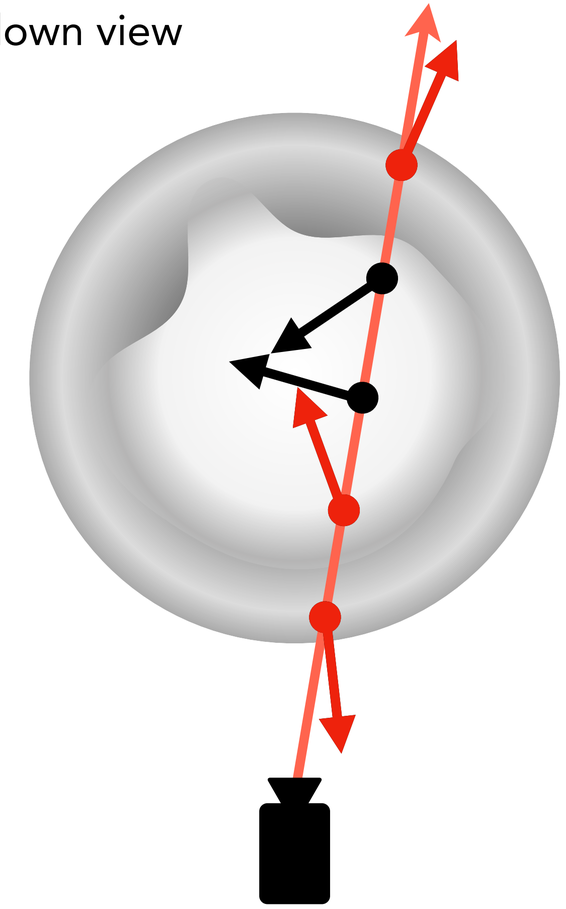
Foggy geometry causes alternating normal vectors

Top-down view



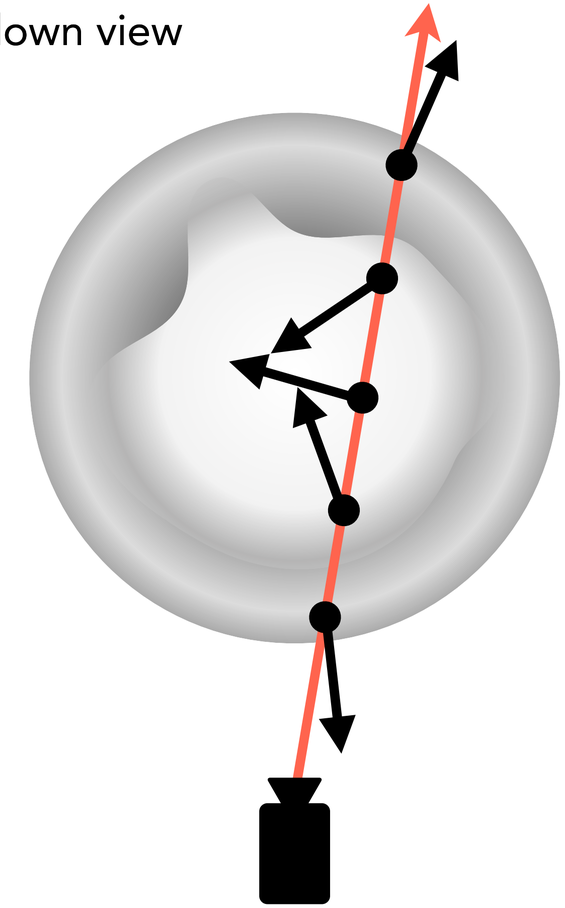
Penalize normals facing away from the camera

Top-down view

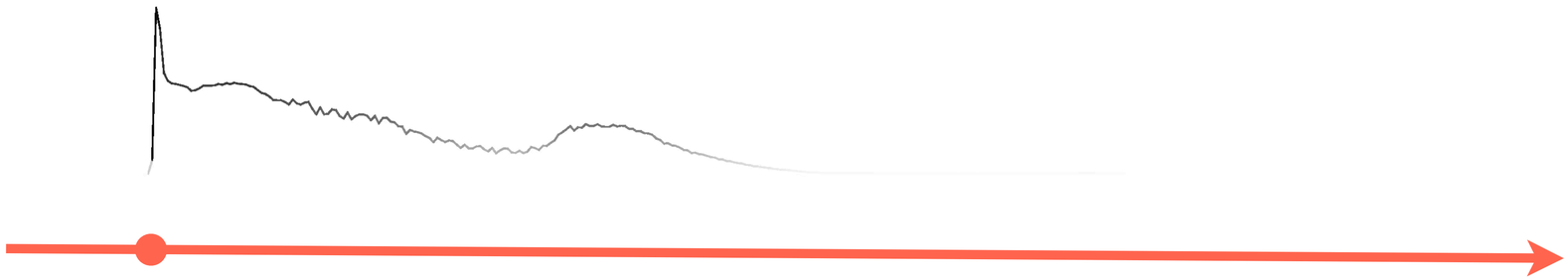


Penalize normals facing away from the camera

Top-down view



Weights



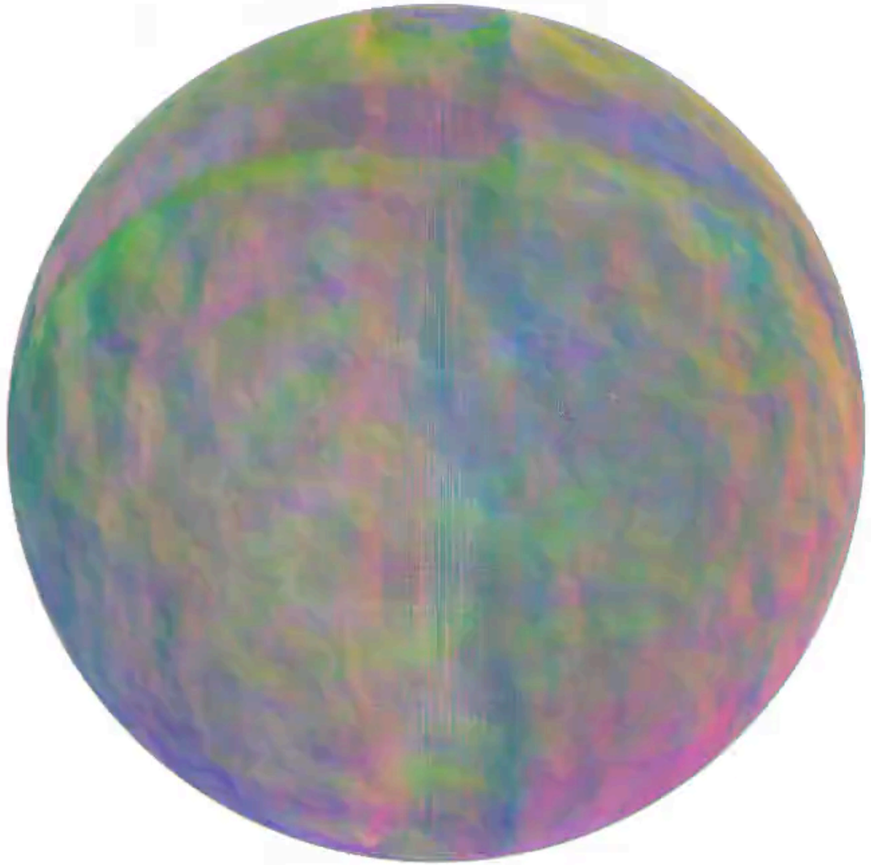
Simple regularizer on normals improves geometry

Top-down view

Back-to-front rendering

Simple regularizer on normals improves geometry

NeRF



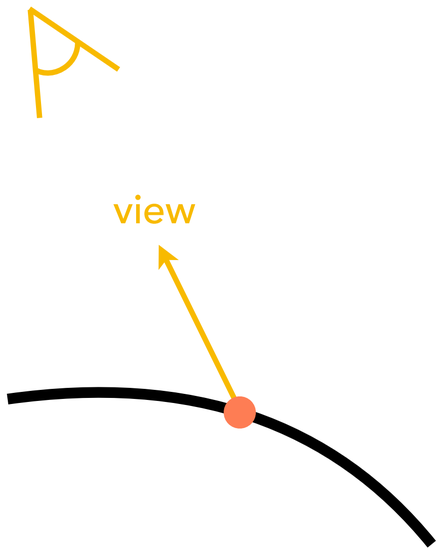
Ours

How to fix NeRF's view-dependent appearance

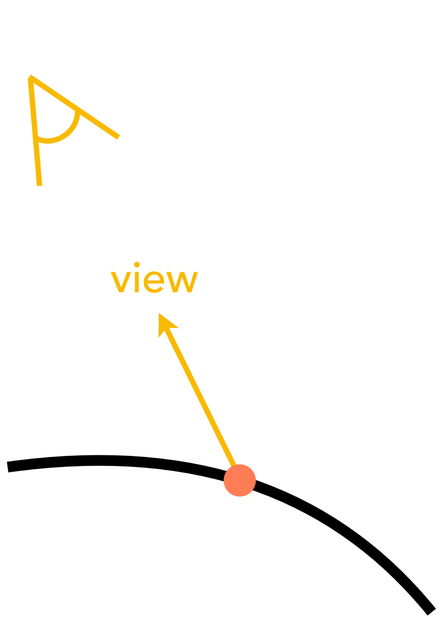
1. Regularize NeRF to fix its foggy geometry and normals

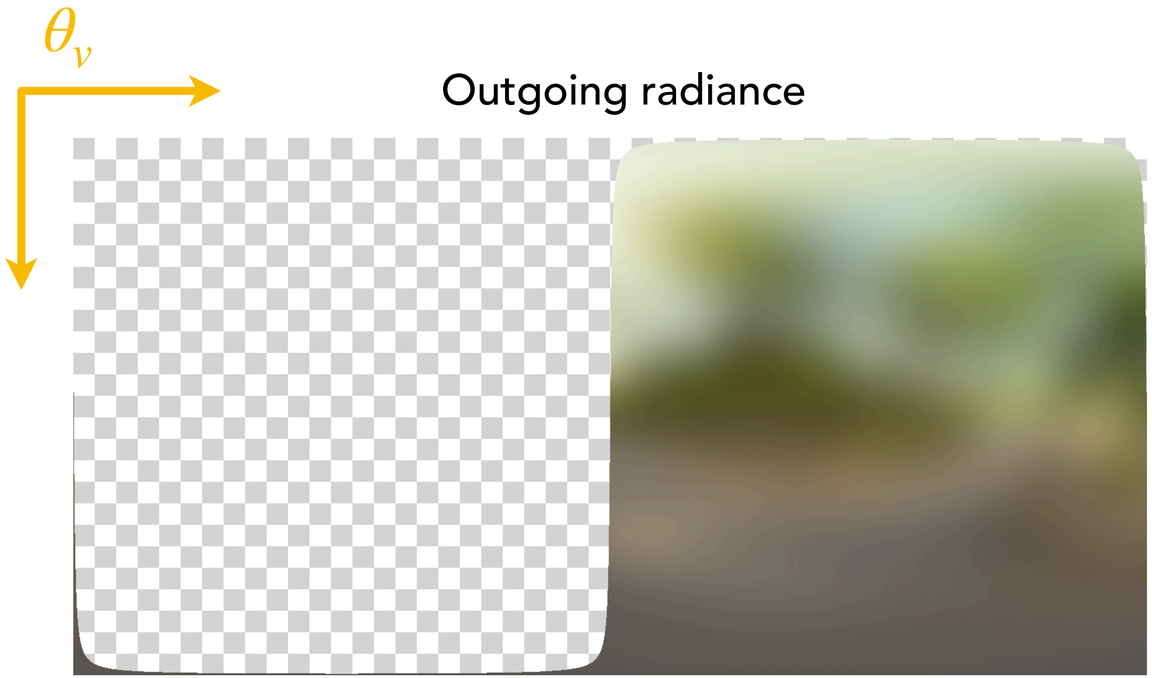
2. Use fixed normals to improve view-dependent appearance

Why is NeRF's parameterization problematic?

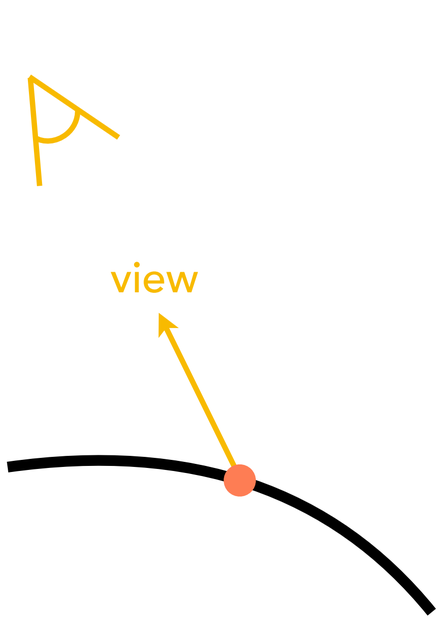


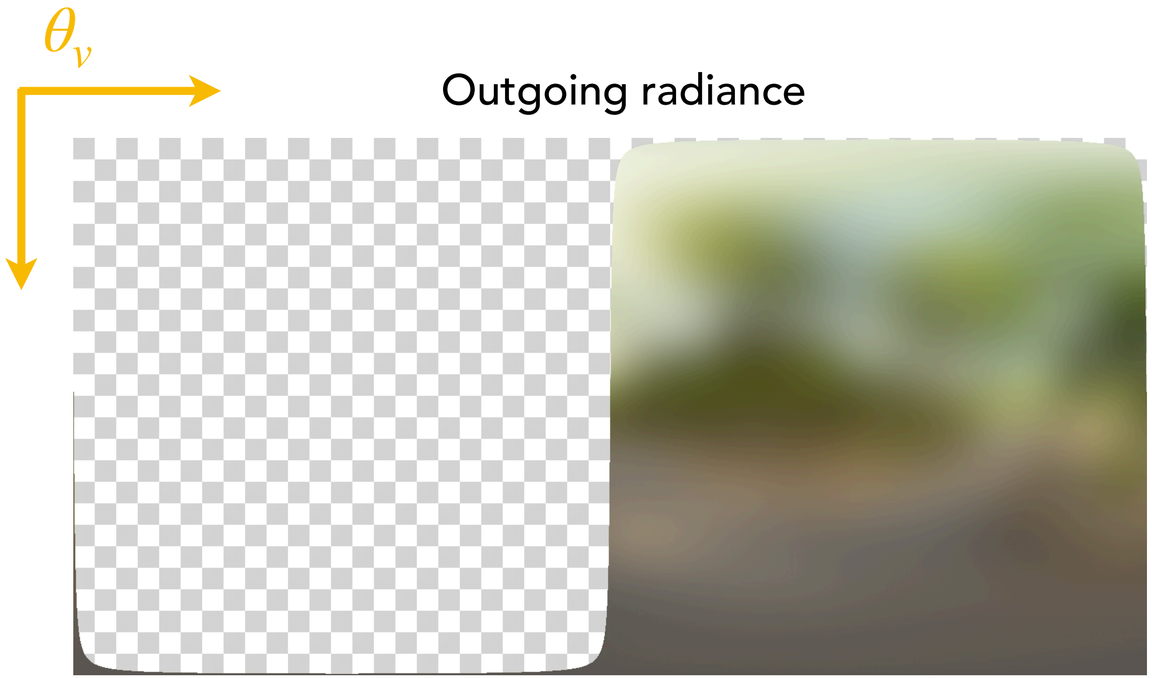
Why is NeRF's parameterization problematic?



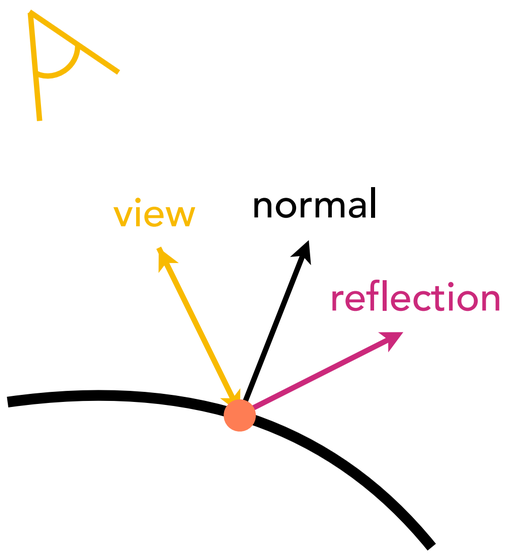


Why is NeRF's parameterization problematic?

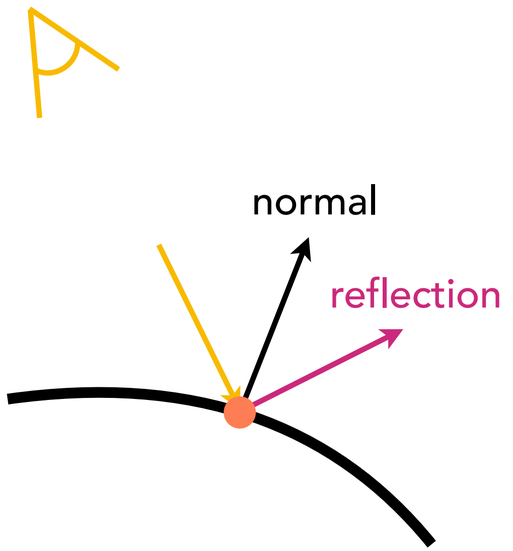




Reparameterizing view-dependent appearance by reflection direction



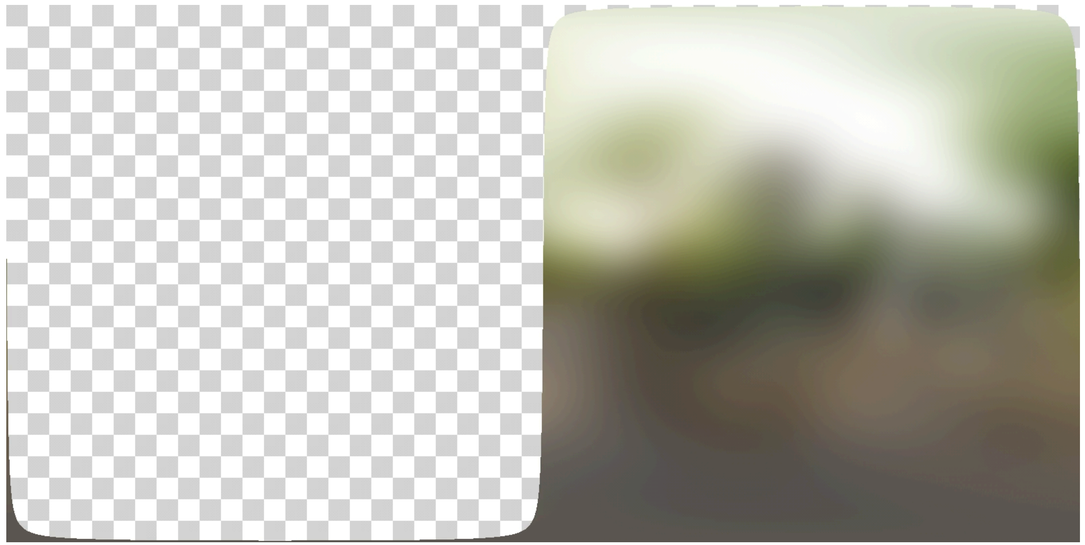
Reflection reparameterization makes function simpler



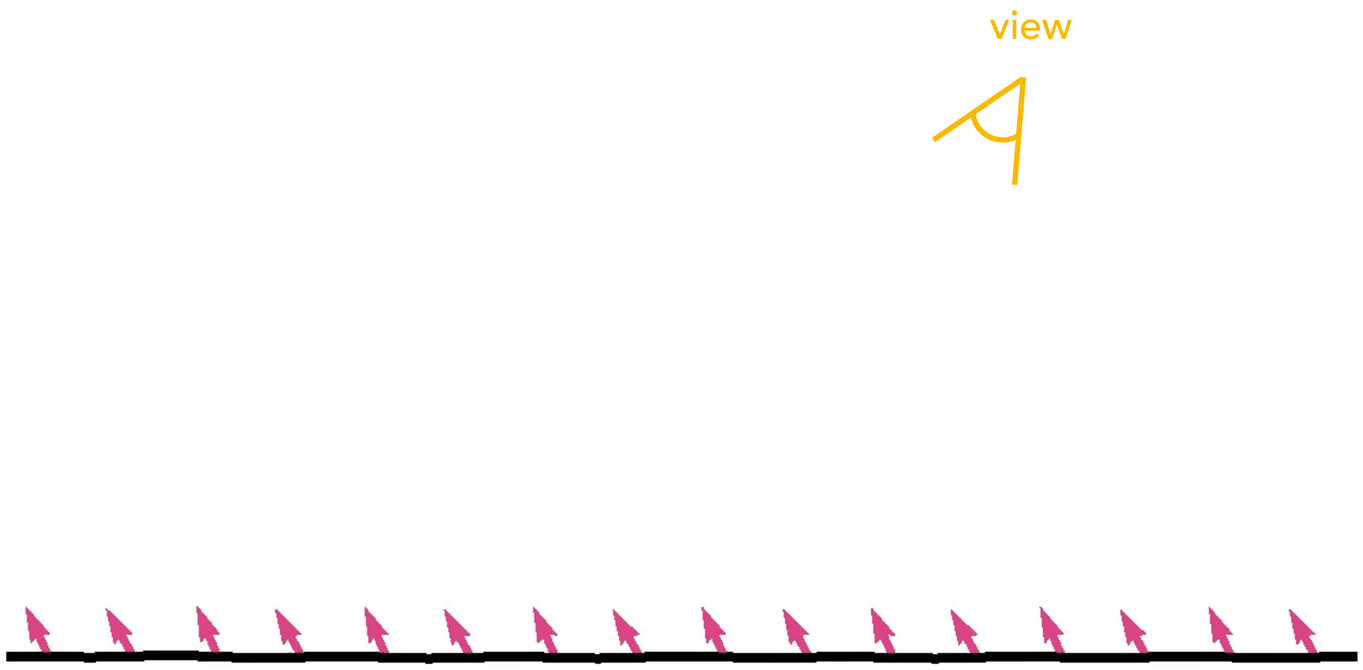
θ_r

ϕ_r

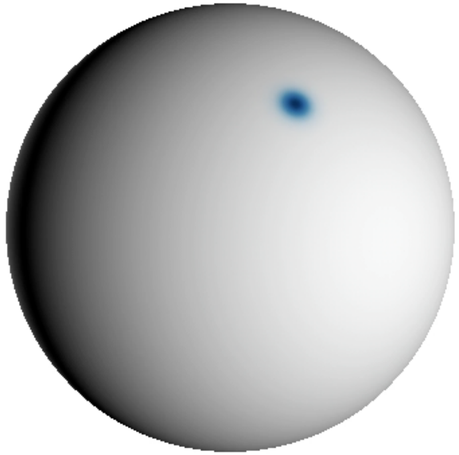
Outgoing radiance



Roughness determines distribution of reflection directions

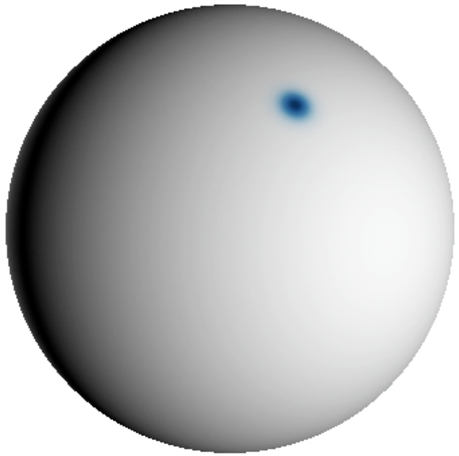


reflection direction distribution

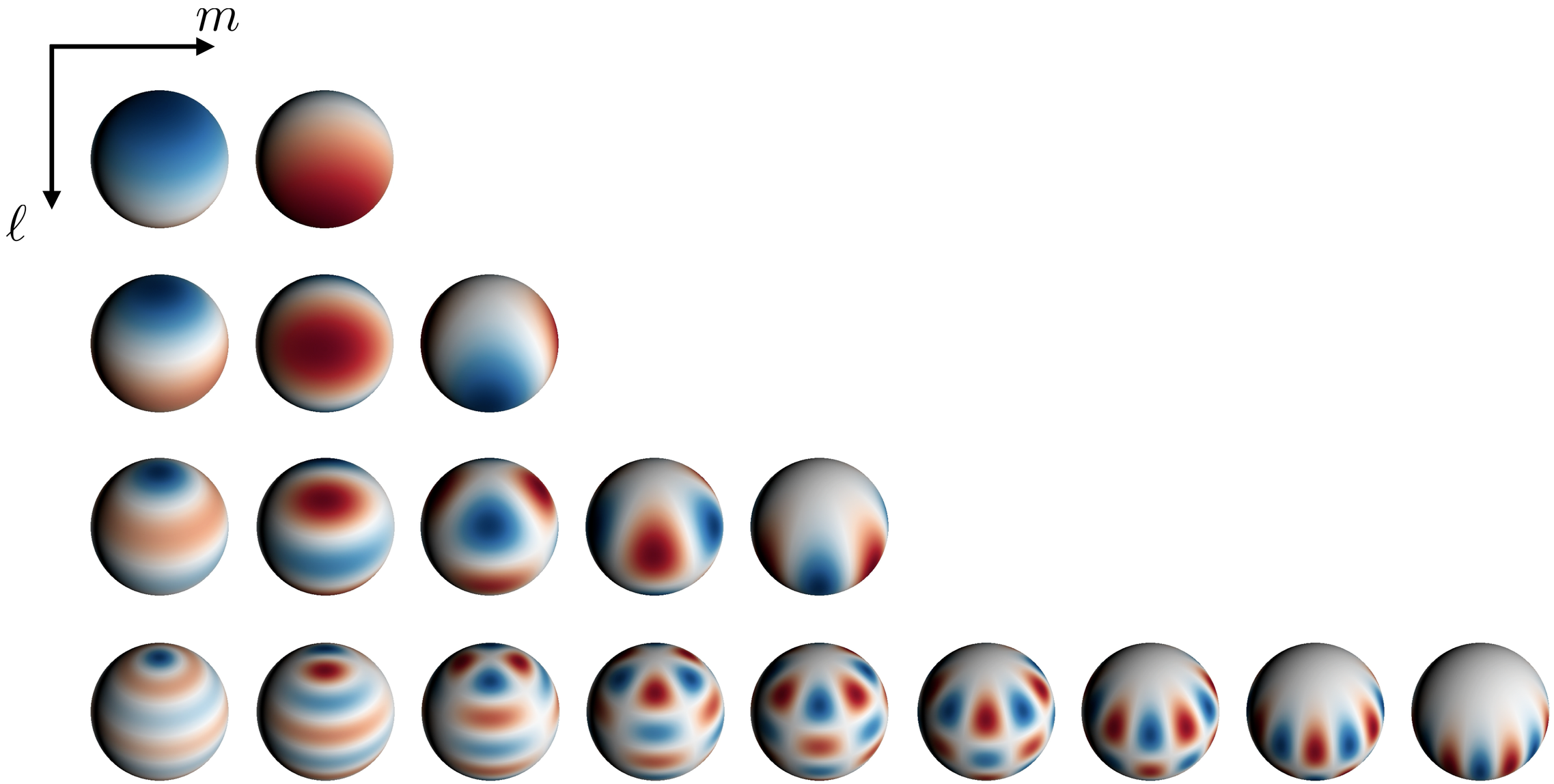


Encoding roughness as a distribution of directions

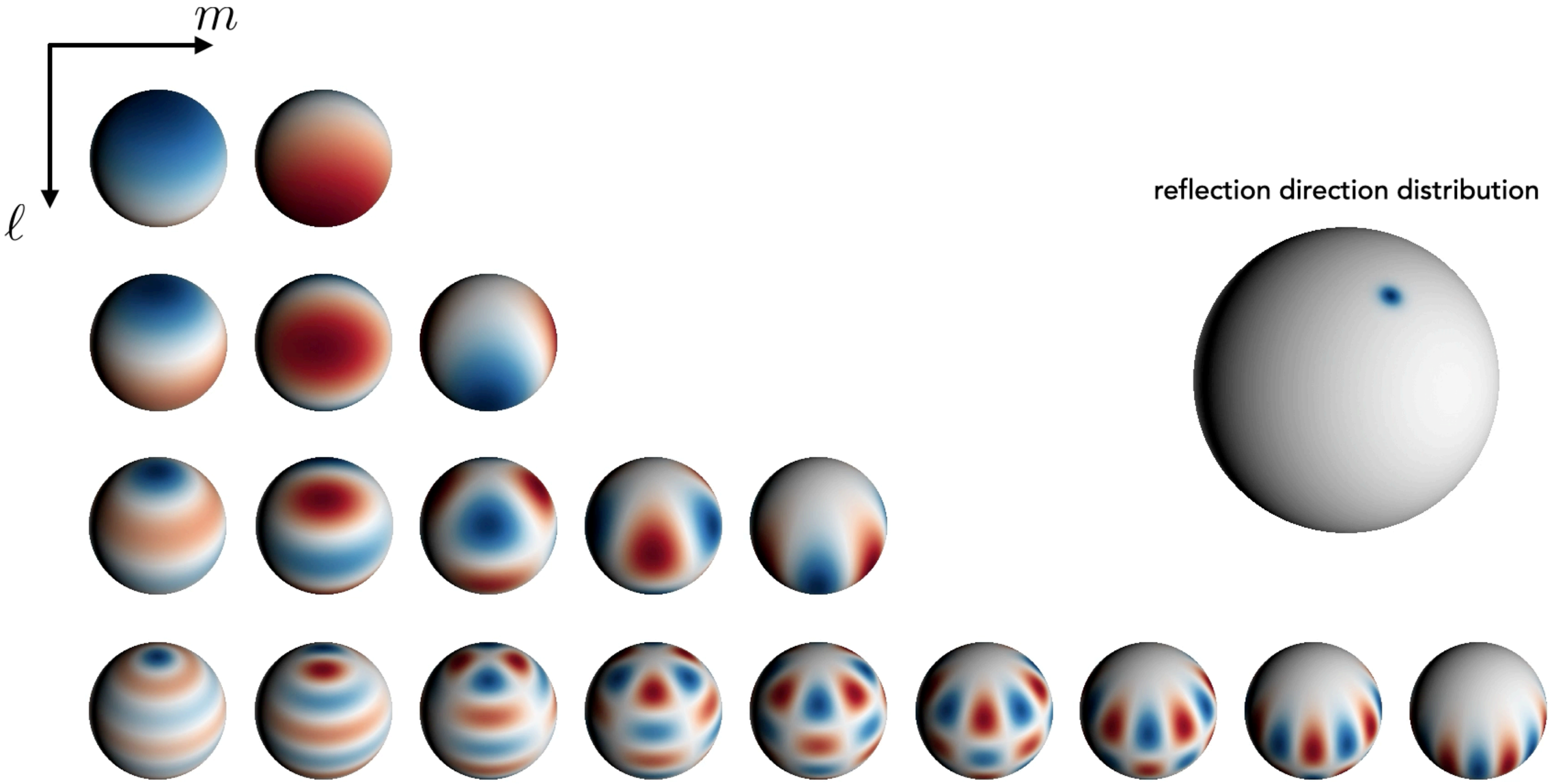
reflection direction distribution
with “width” r



Directional encoding using spherical harmonics



Integrated Directional Encoding



Improved geometry and specular appearance

Our normals

Our renderings

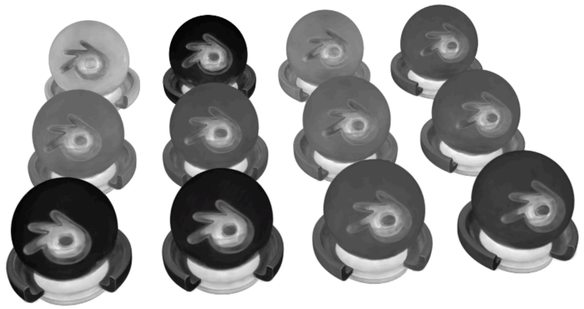
Results

Mip-NeRF


Our Renderings

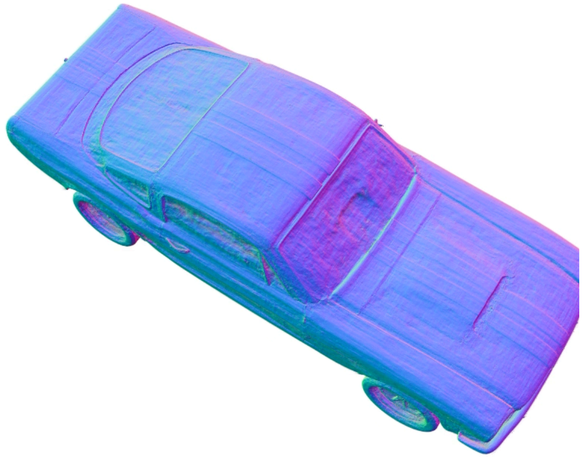
Ref-NeRF decomposes appearance into meaningful components

Normals

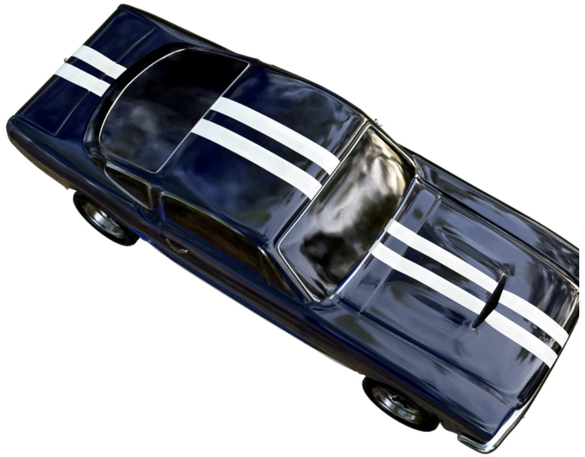


Roughness

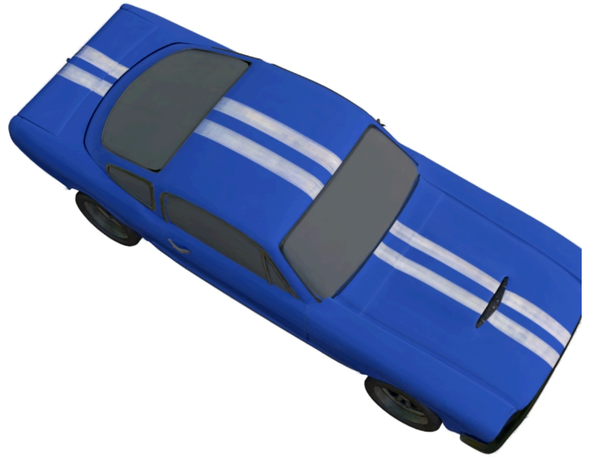
Tint



Normals

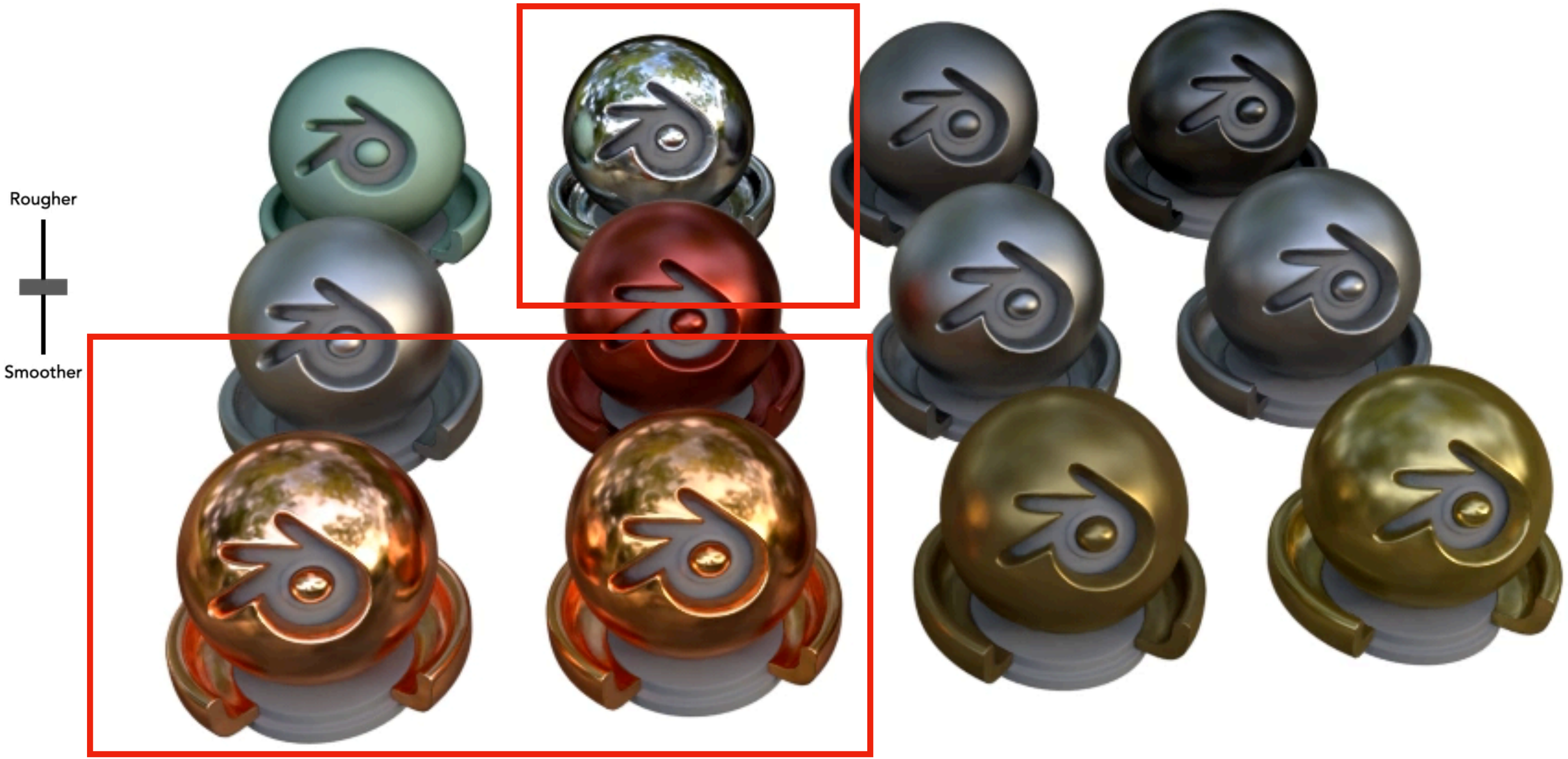


Specular color



Diffuse color

Roughness editing



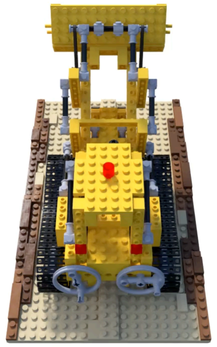
Roughness editing

Editing specular and diffuse colors

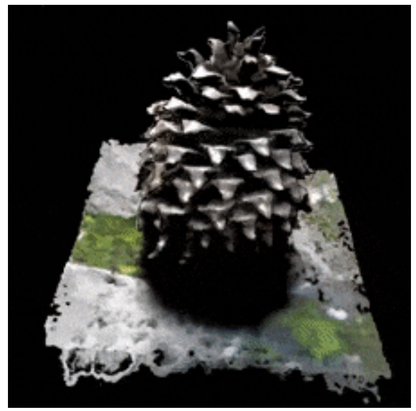
Editing specular and diffuse colors

What should the Neural Field store at any point?

What lies in between these extremes?



other representations
worth considering?



viewpoint

viewpoint,
lighting,
material