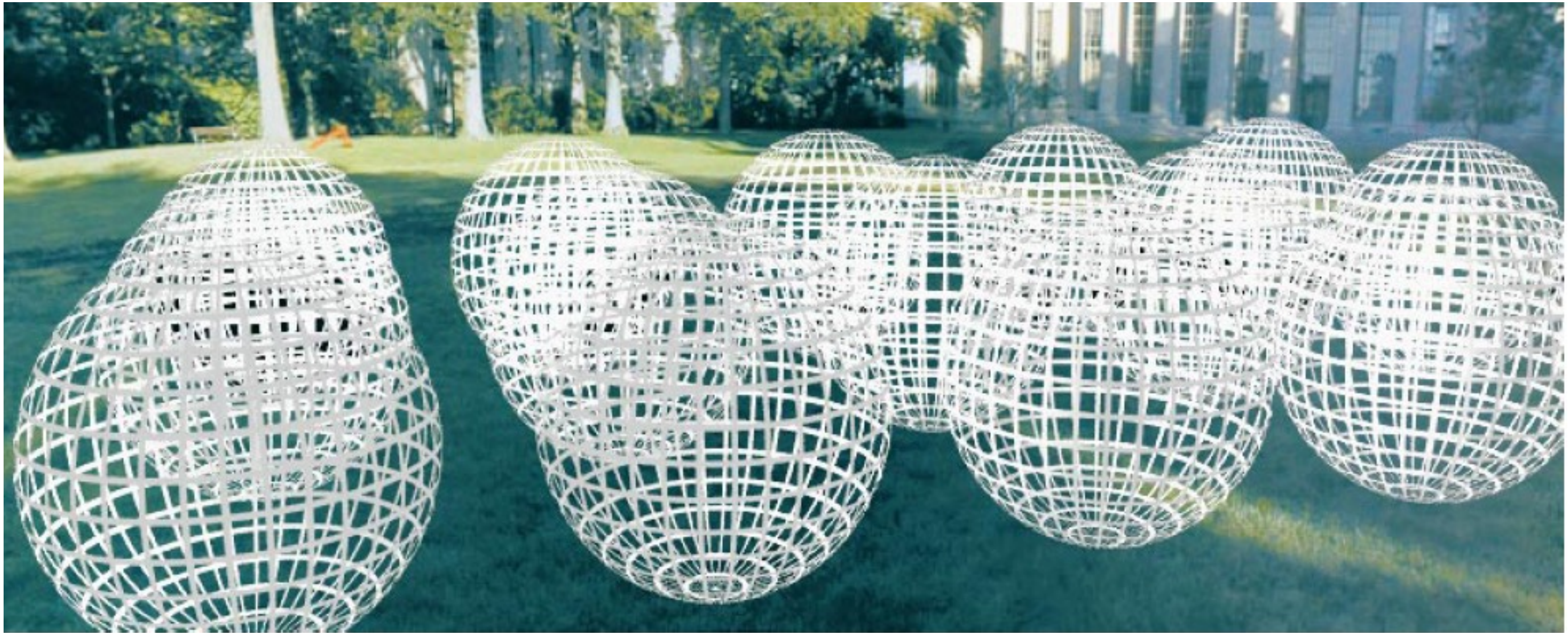
Modeling the plenoptic function



Adopted from: CS194: Intro to Comp. Vision, and Comp. Photo
Alexei Efros & Angjoo Kanazawa, UC Berkeley, Fall 2021 and Lana Lazebnik

Outline

- The plenoptic function
- Two-plane light fields
- Neural radiance fields (NeRFs)

Goal: Novel view rendering

- Given several images of the same object or scene from known viewpoints, how can we generate a rendering of the same scene from a novel viewpoint?
- Multiview stereo answer: create a textured 3D model from the images, use traditional graphics to render

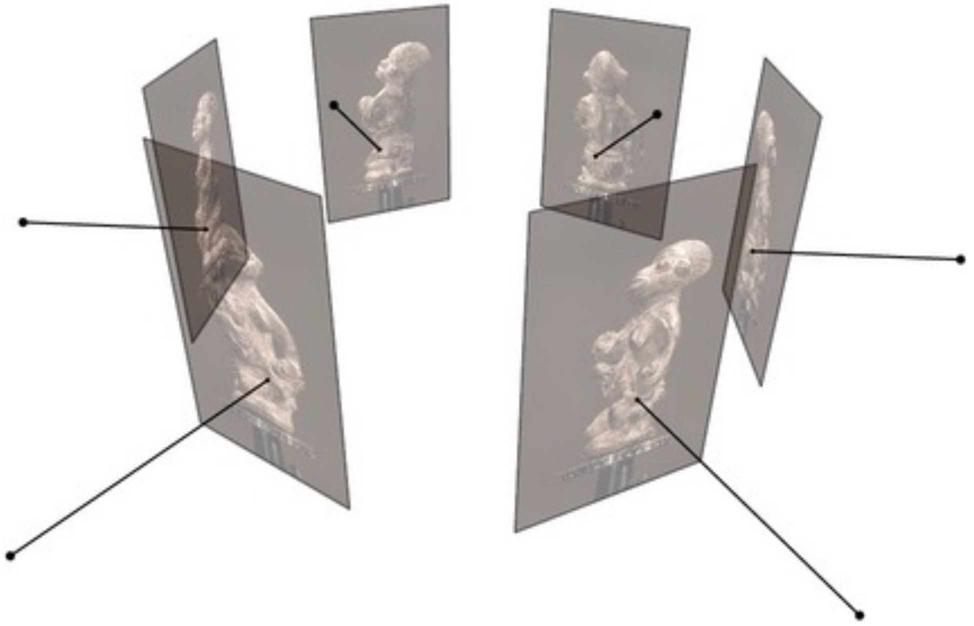
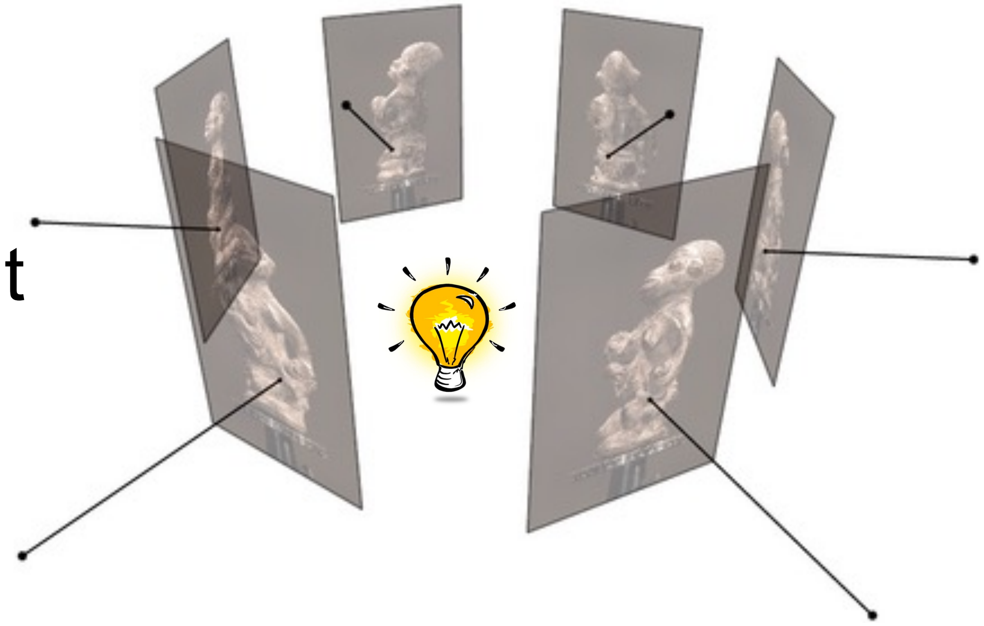


Figure source: C. Hernandez, N. Snavely

Goal: Novel view rendering

- Given several images of the same object or scene from known viewpoints, how can we generate a rendering of the same scene from a novel viewpoint?
- Multiview stereo answer: create a textured 3D model from the images, use traditional graphics to render
- Alternate answer: model the *light field* of the scene, sample new views from it



The light field, or plenoptic function

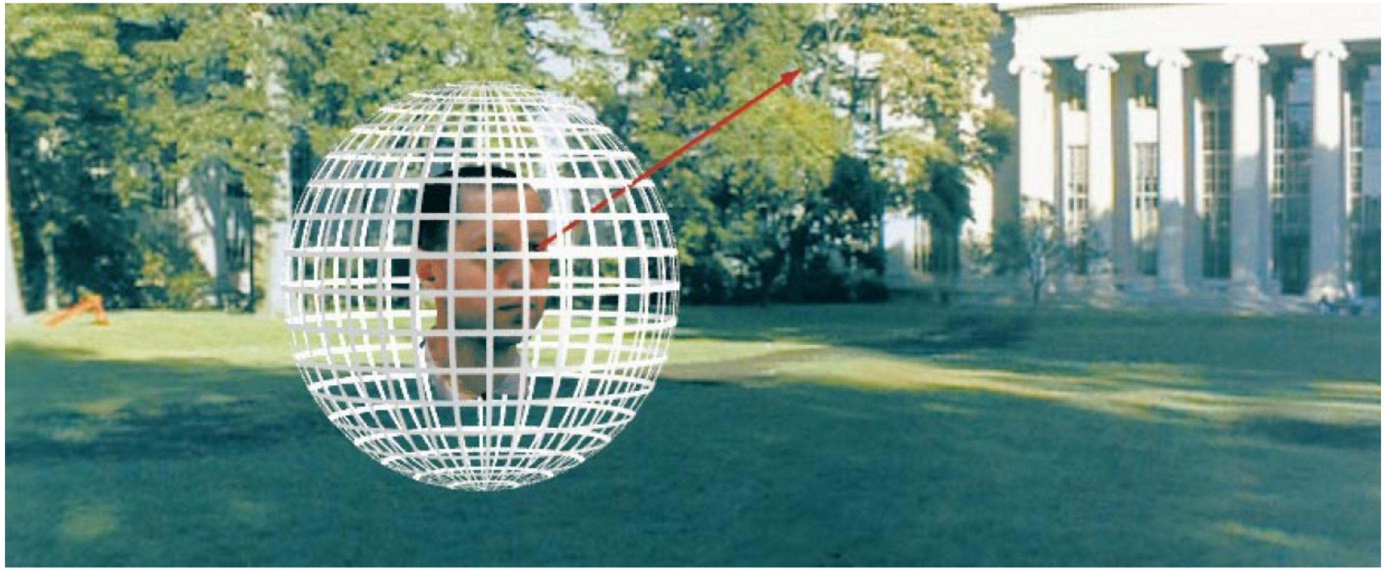


Figure by Leonard McMillan

Q: What is the set of all things that we can ever see?

A: The *plenoptic function*

E. Adelson and J. Bergen. The plenoptic function and the elements of early vision.
Computational models of visual processing, MIT Press, 1991

The light field, or plenoptic function

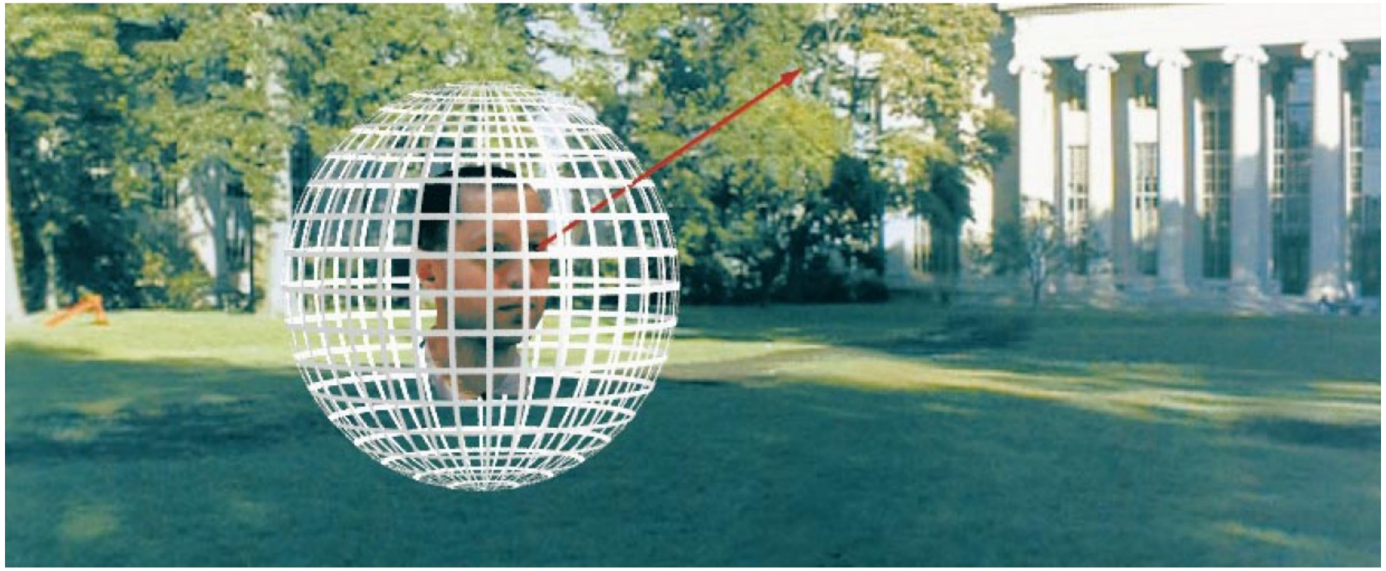


Figure by Leonard McMillan

Q: What is the set of all things that we can ever see?

A: The *plenoptic function*

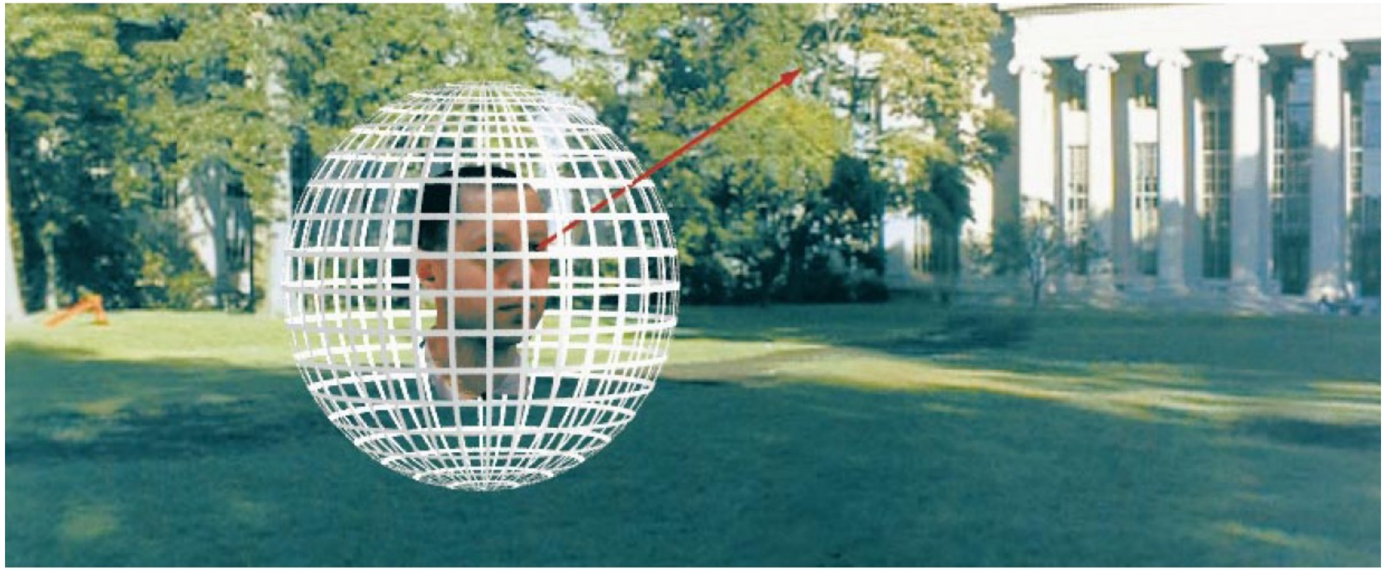
Let's start with a stationary person and try to parameterize everything that they can see…

Grayscale snapshot

$$L(\theta, \phi)$$

- Intensity of light
  - Seen from a single view point
  - At a single time
  - Averaged over the wavelengths of the visible spectrum

Color snapshot

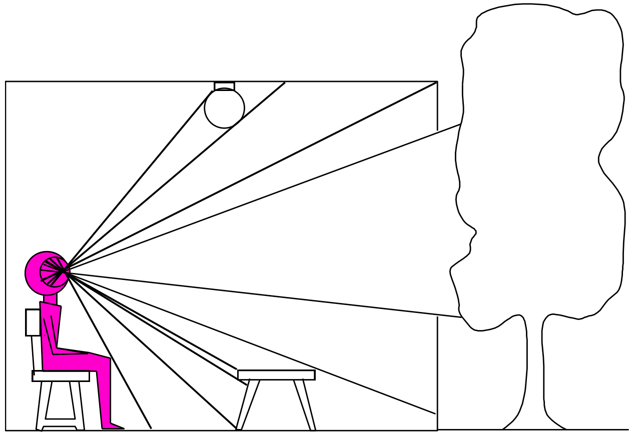


$$L(\theta, \phi, \lambda)$$

- Intensity of light
  - Seen from a single view point
  - At a single time
  - As a function of wavelength

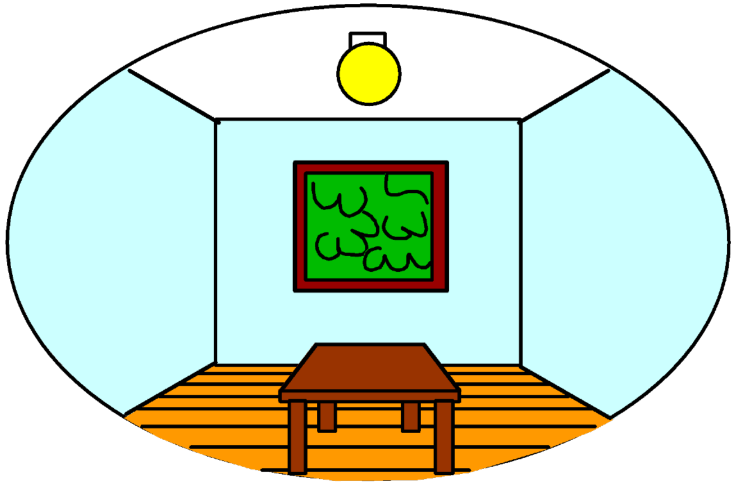
Modeling the light field

3D world



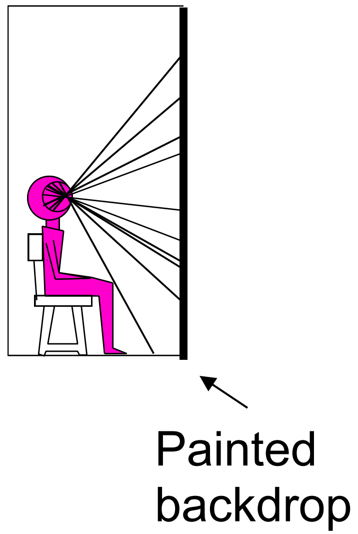
Point of observation

2D image

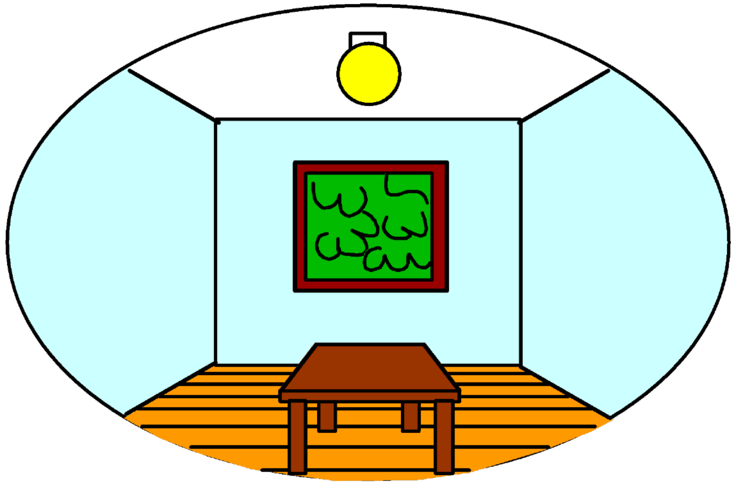


Modeling the light field

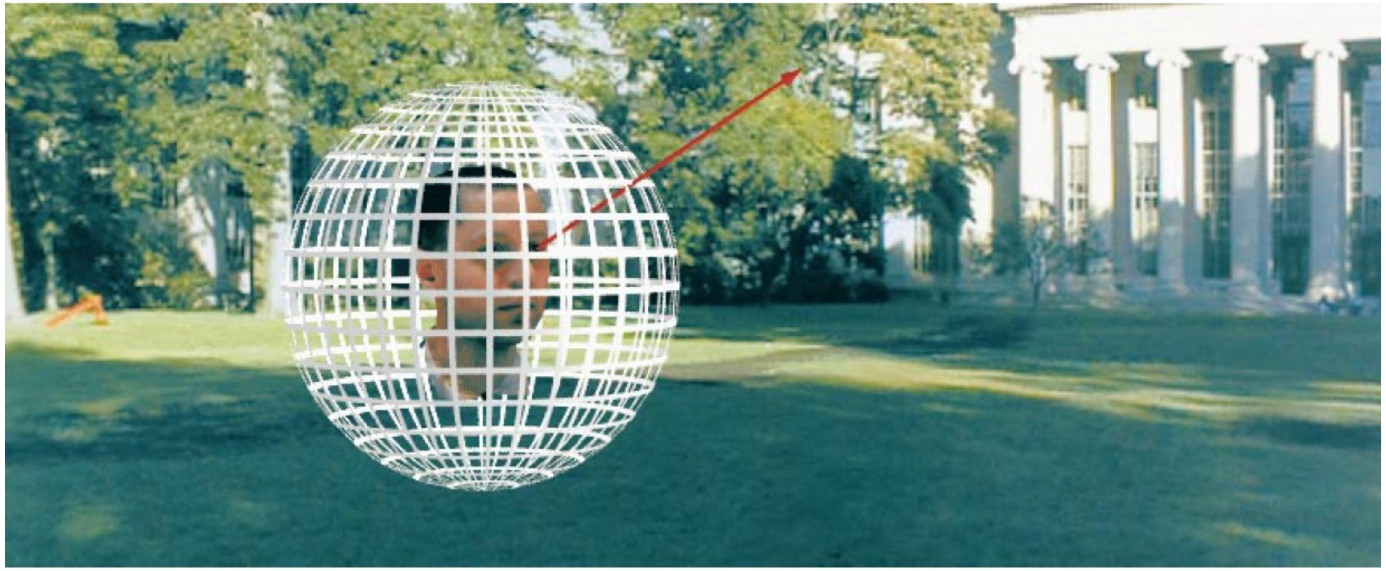
3D world



2D image



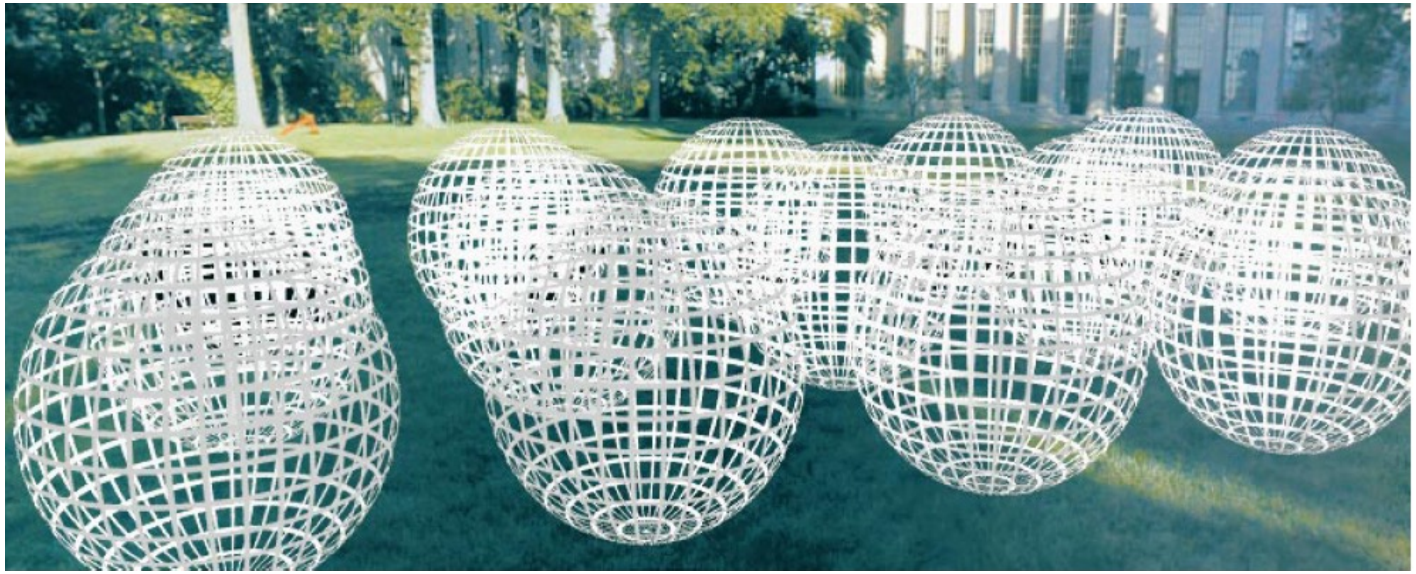
A movie



$$L(\theta, \phi, \lambda, t)$$

- Intensity of light
  - Seen from a single view point
  - Over time
  - As a function of wavelength

Holographic movie



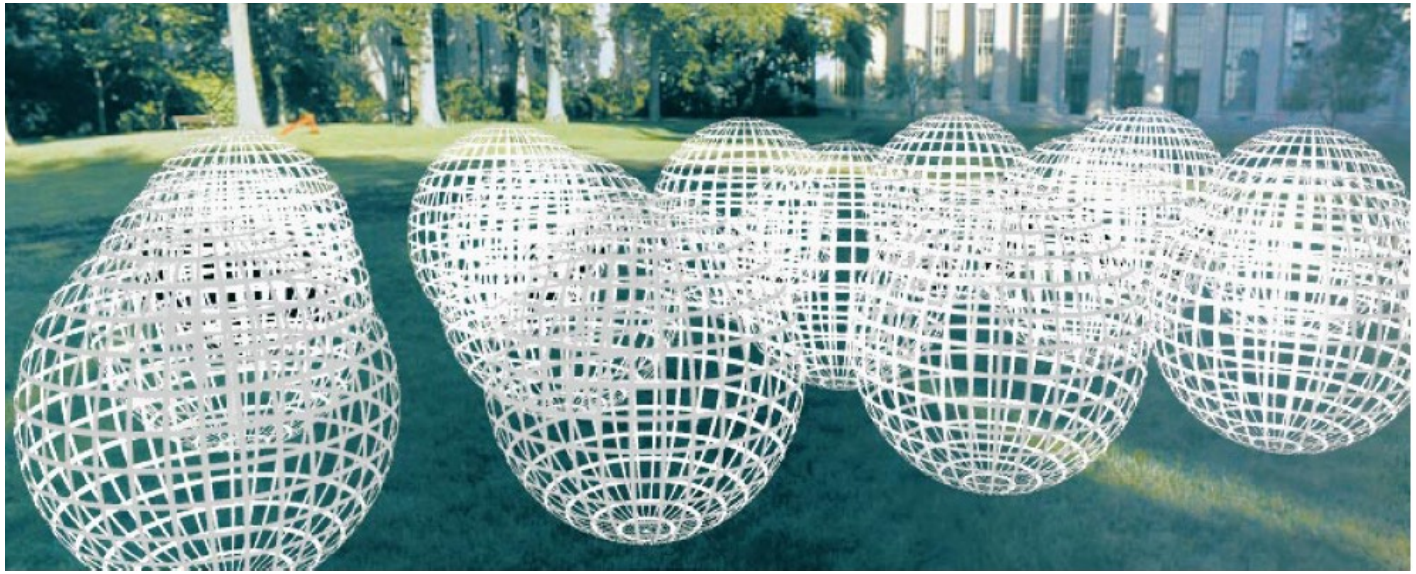
$$L(\theta, \phi, \lambda, t, x, y, z)$$

- Intensity of light
  - Seen from ANY viewpoint
  - Over time
  - As a function of wavelength

Light field modeling: Outline

- The plenoptic function
- Two-plane light fields
- Neural radiance fields (NeRFs)

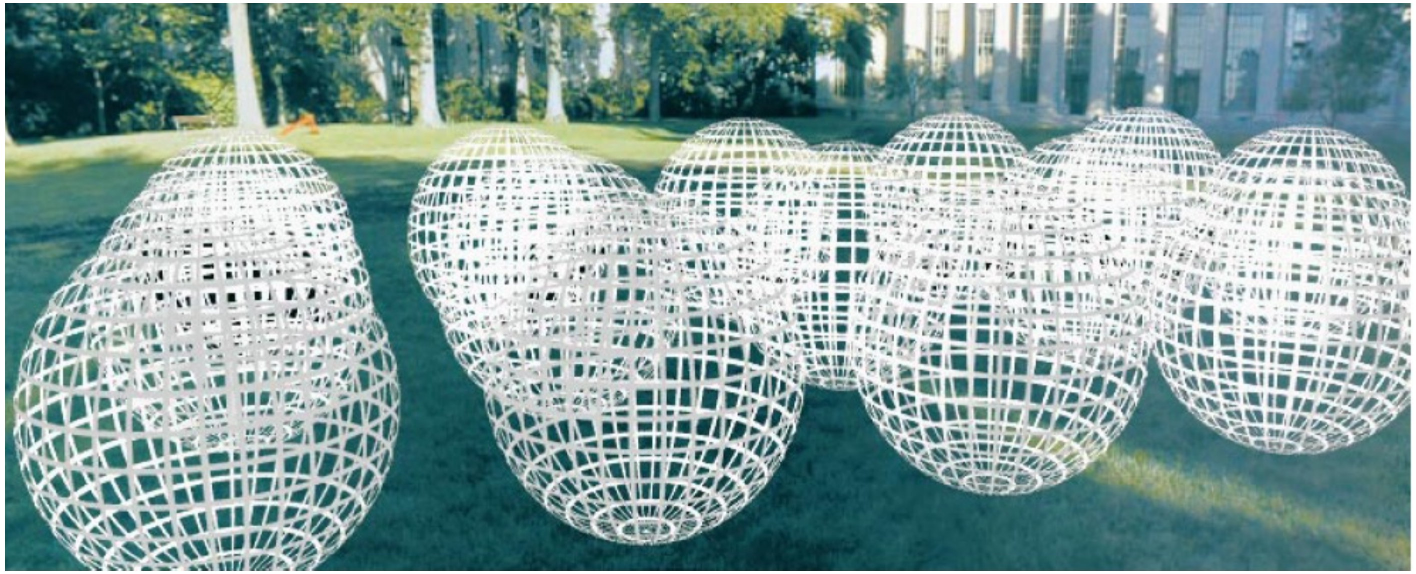
The plenoptic function



$$L(\theta, \phi, \lambda, t, x, y, z)$$

- Can reconstruct every possible view, at every moment, from every position, at every wavelength
- Contains every photograph, every movie, everything that anyone has ever seen! it completely captures our visual reality!
- Not bad for a function…

The plenoptic function: More practical version



$$L(\theta, \phi, x, y, z) = (r, g, b)$$

- Other simplifications/variants are possible, as we will see

Modeling the plenoptic function

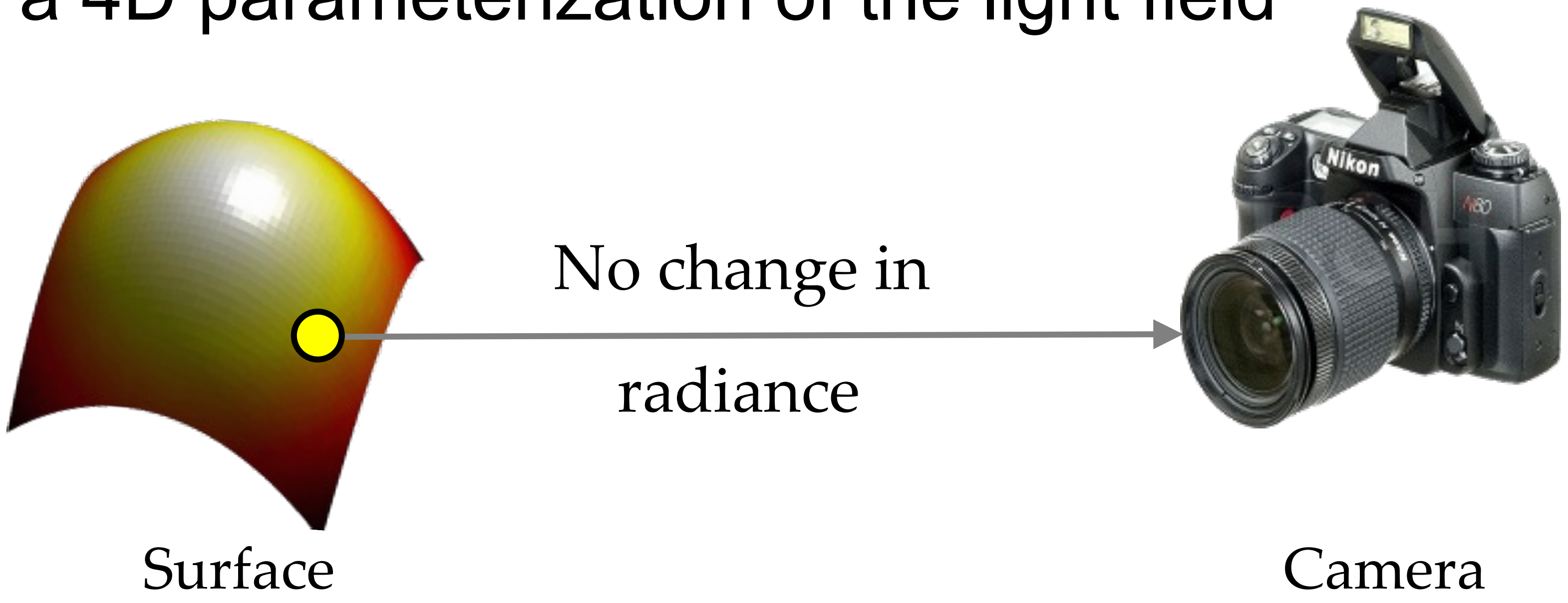
- Capture
  - Create a special camera setup to capture a slice of the plenoptic function
  - Combine captured rays for novel view synthesis, defocus, and other effects
- Optimization
  - Given a set of multi-view calibrated images, optimize a parametric representation of the plenoptic function of the scene

Outline

- The plenoptic function
- Two-plane light fields

Two-plane light fields

- Key idea: assuming light is constant along rays, we can create a 4D parameterization of the light field



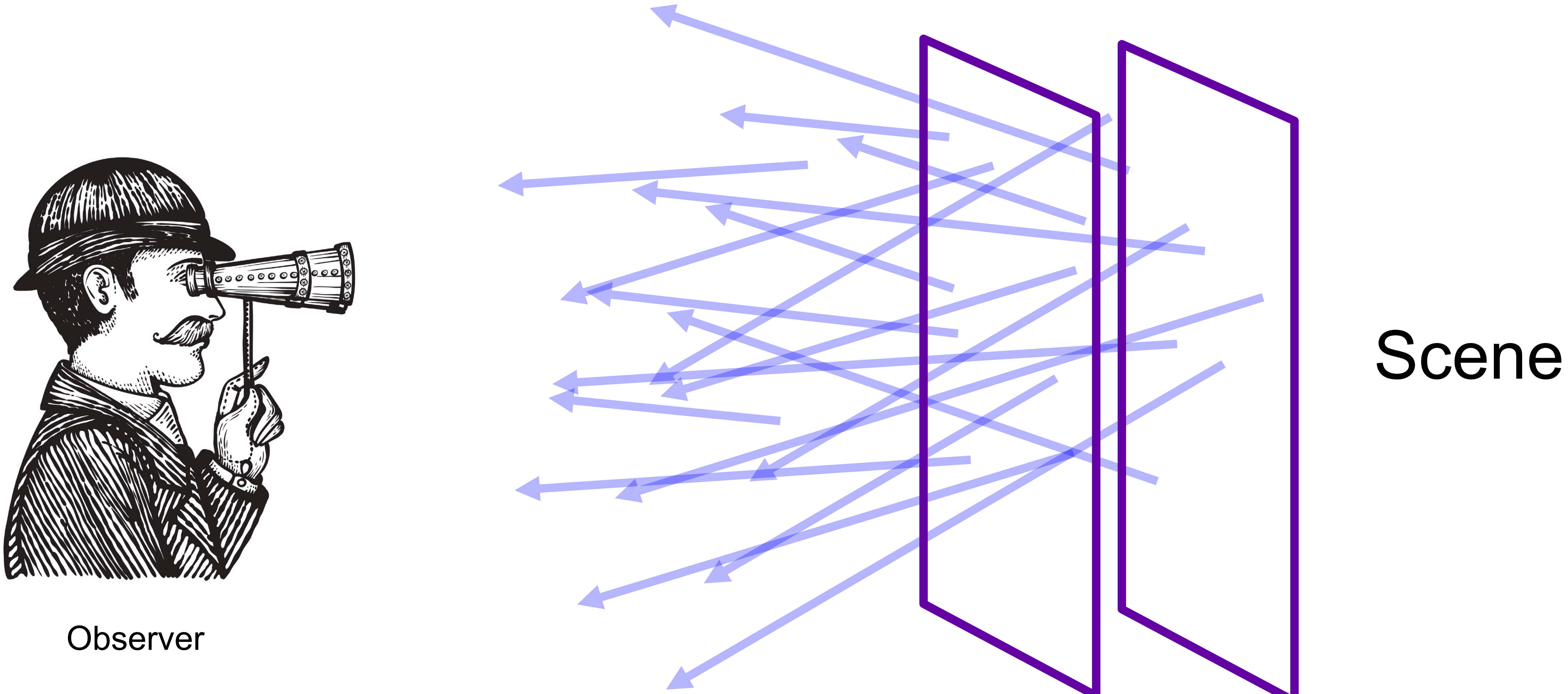
If there is no occlusion or fog

S. Gortler, R. Grzeszczuk, S. Szeliski, M. Cohen. The Lumigraph. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, 1996

M. Levoy and P. Hanrahan. Light field rendering. SIGGRAPH 1996

Two-plane light fields

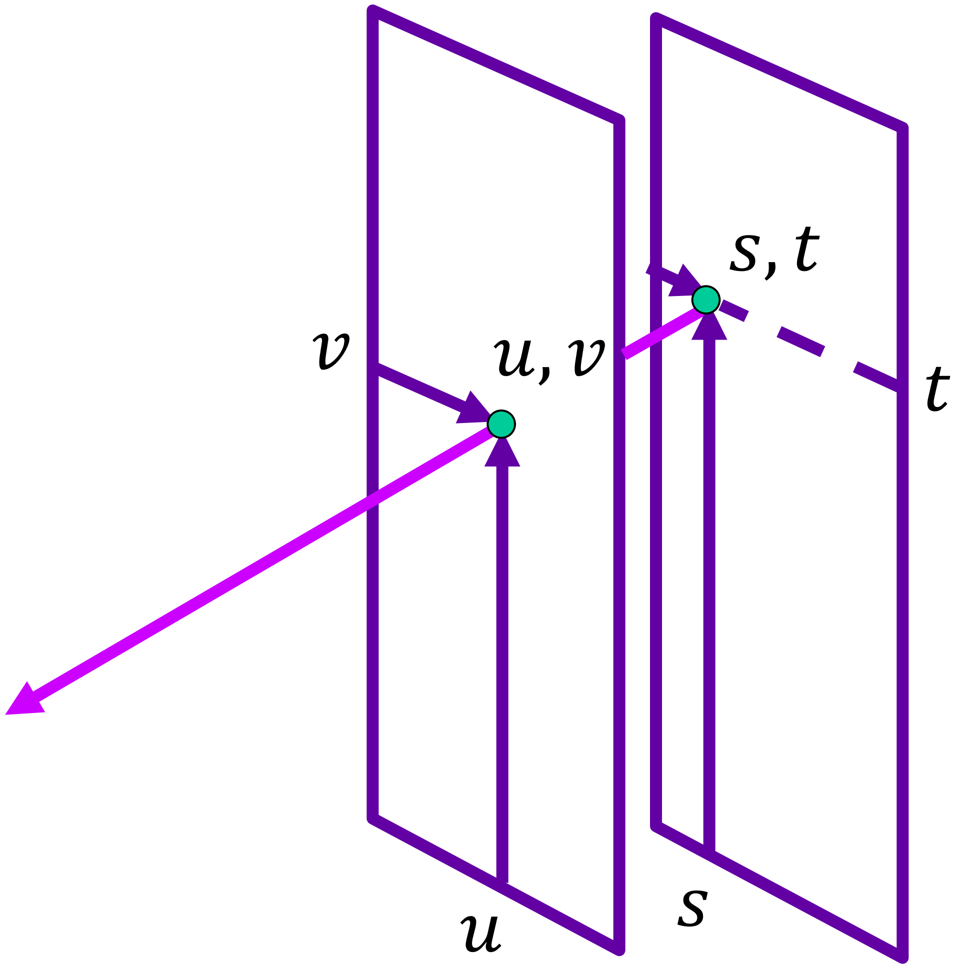
- Two-plane parameterization:



Two-plane light fields

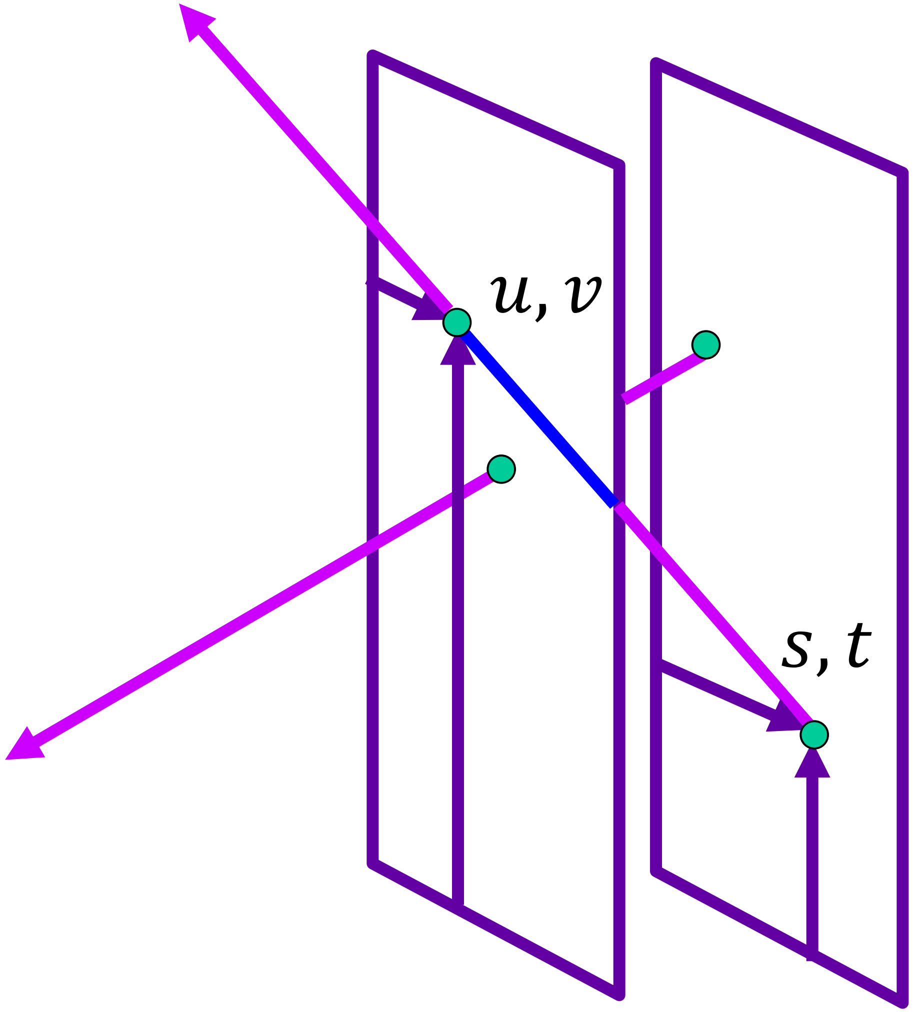
- Two-plane parameterization:

$$L(s,t,u,v) = (r,g,b)$$



Two-plane light fields

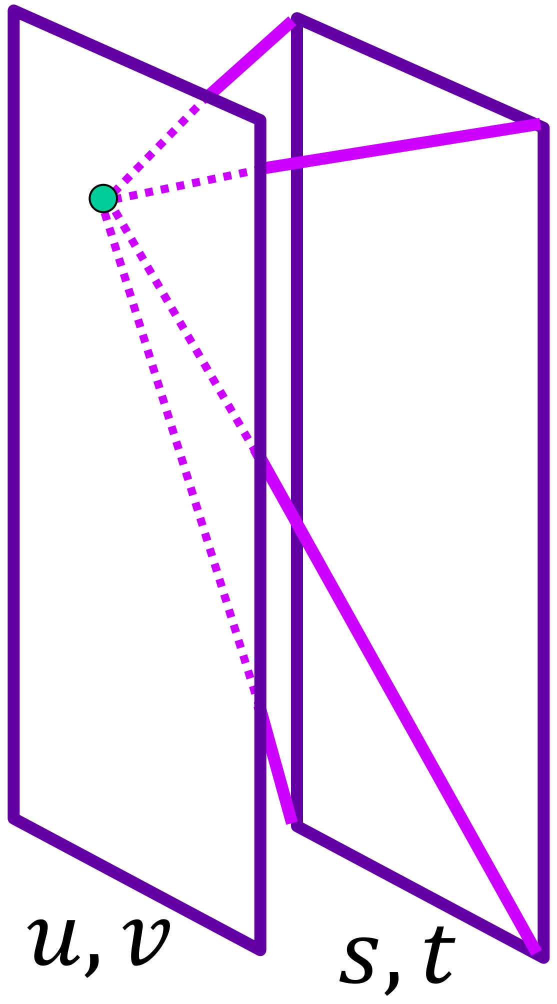
- Two-plane parameterization:



$$L(s, t, u, v) = (r, g, b)$$

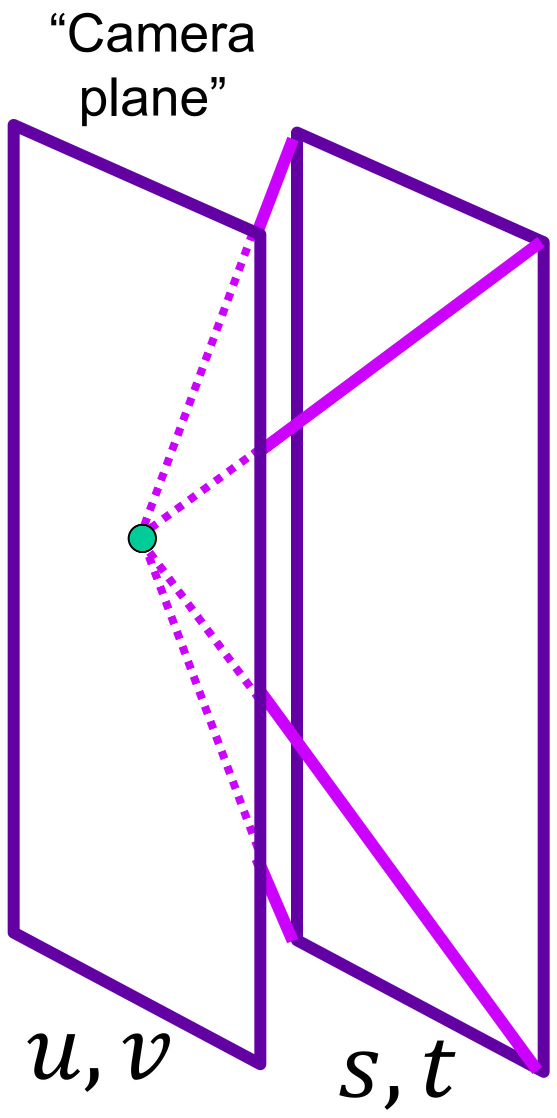
Two-plane light fields

- What do we get if we hold u, v constant and let s, t vary?
- An image!



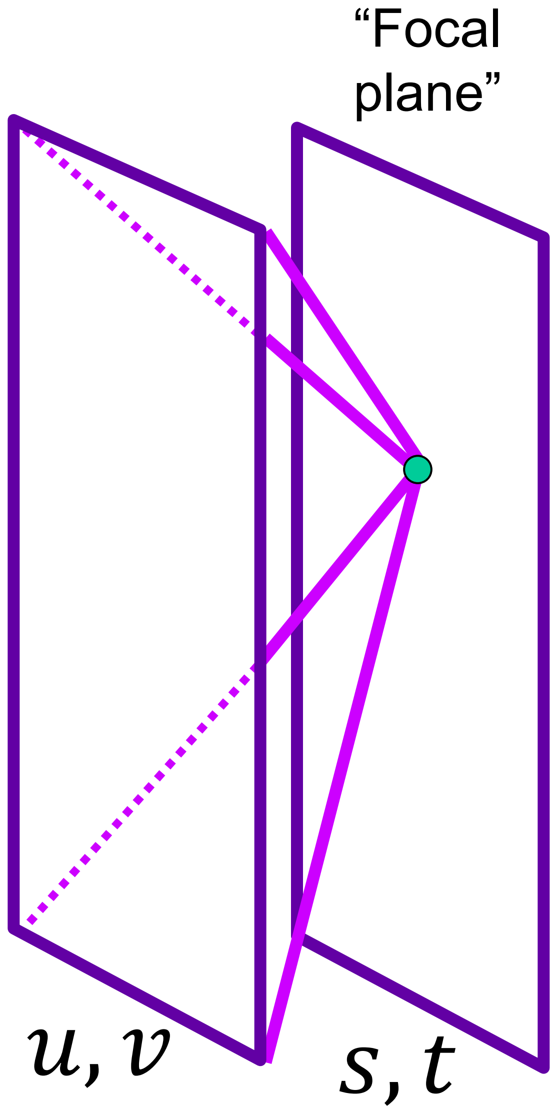
Two-plane light fields

- What do we get if we hold u, v constant and let s, t vary?
- An image!



Two-plane light fields

- What do we get if we hold s, t constant and let u, v vary?
- A set of rays leaving a point in the scene in a bundle of directions towards the image plane



Light field visualization

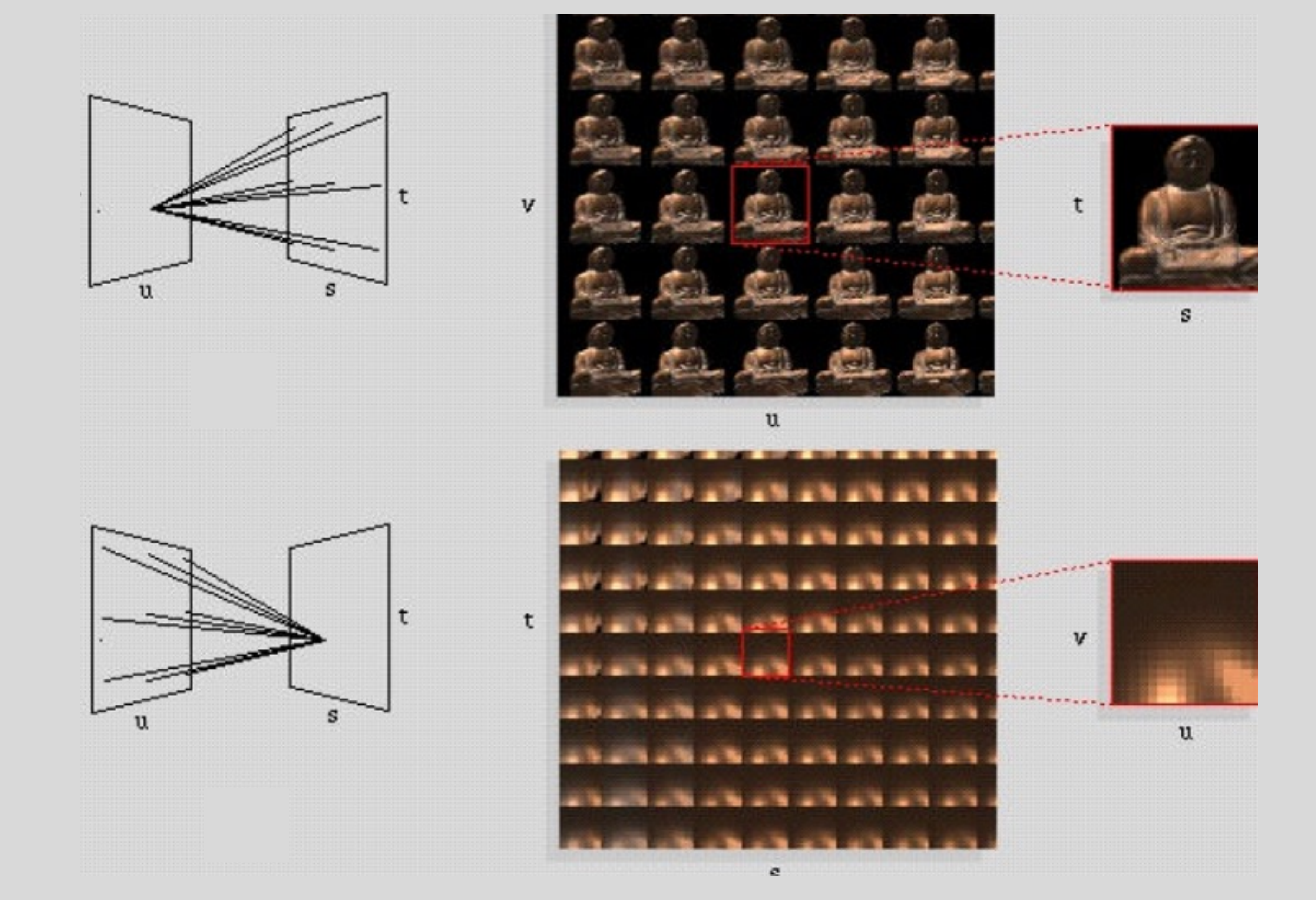
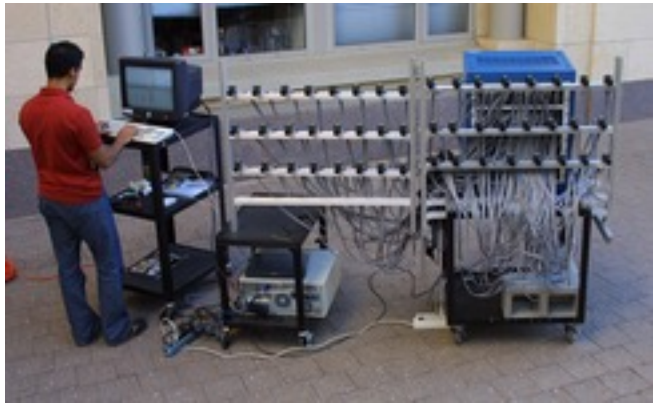
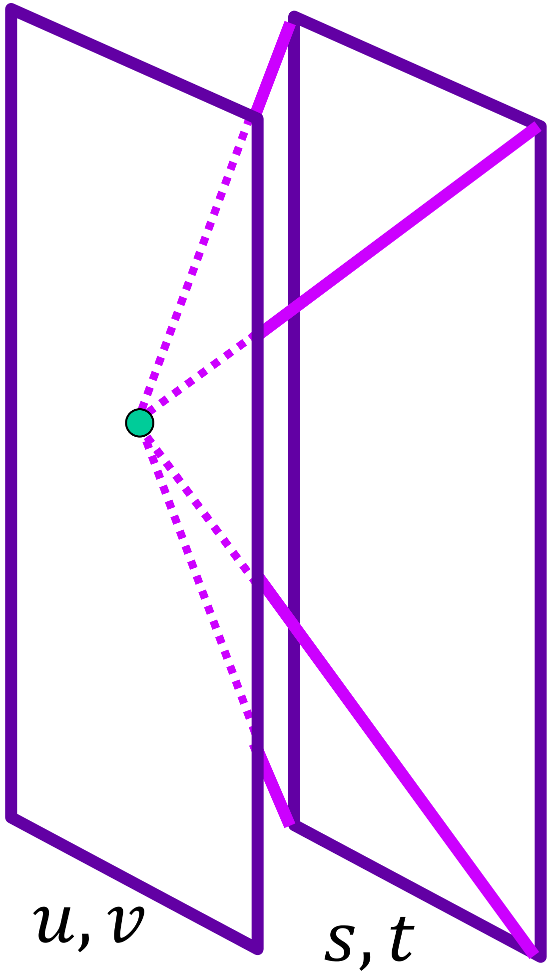


Figure source: M. Levoy and P. Hanrahan

Light field capture

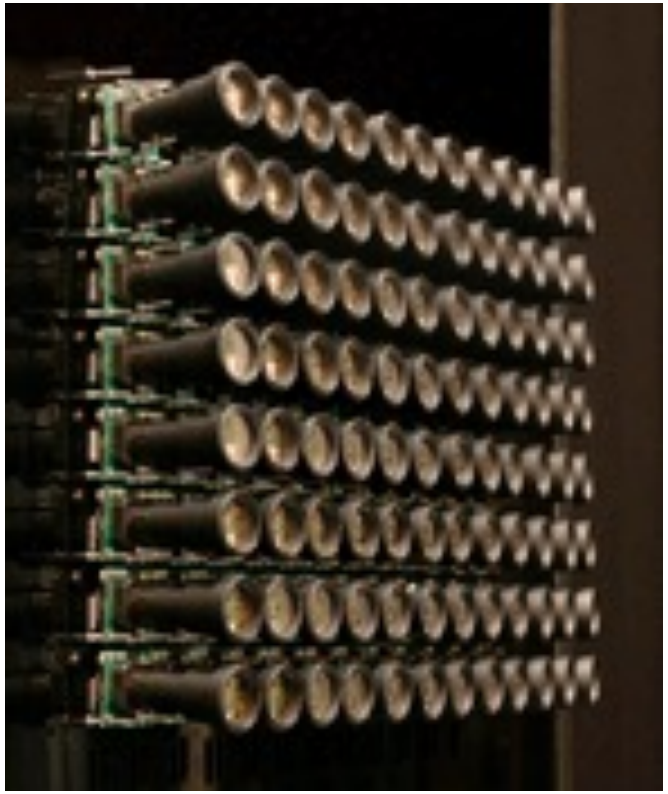
- Idea 1: move camera carefully over u, v plane

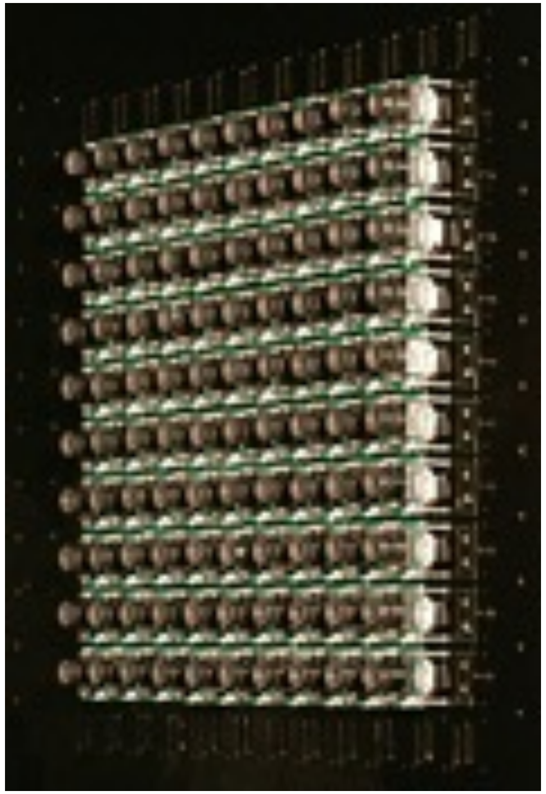




Stanford multi-camera array

- 640 × 480 pixels × 30 fps × 128 cameras
- Synchronized timing
- Continuous streaming
- Flexible arrangement

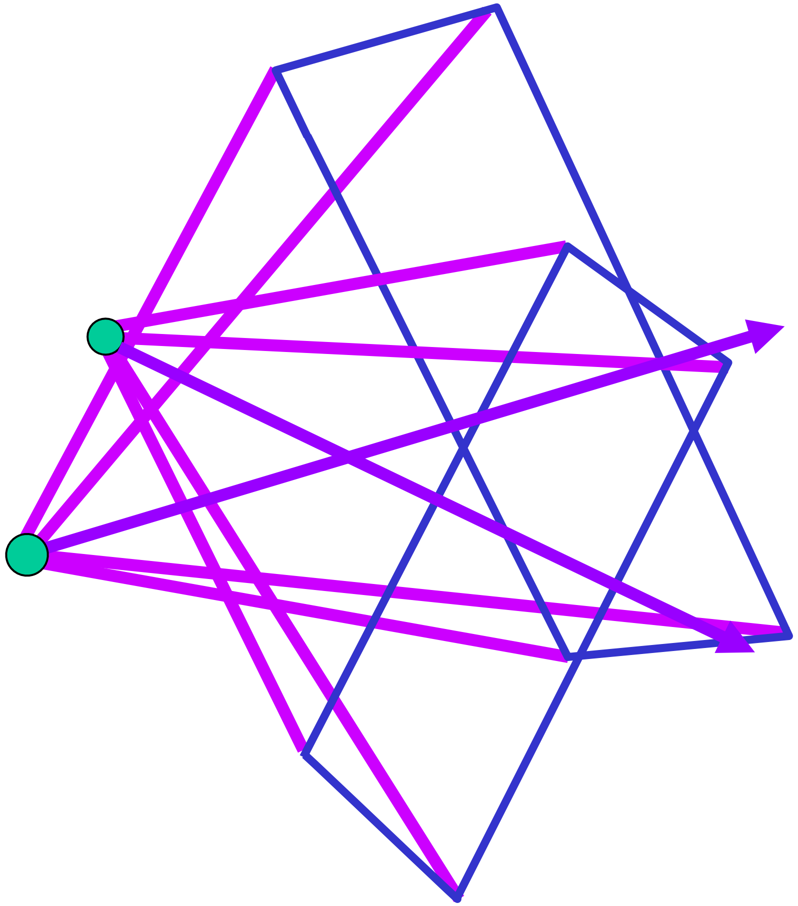




http://graphics.stanford.edu/projects/array/

Light field capture

- Idea 2: move camera anywhere, use rebinning or resampling



Light field capture

- Idea 2: move camera anywhere, use rebinning or resampling

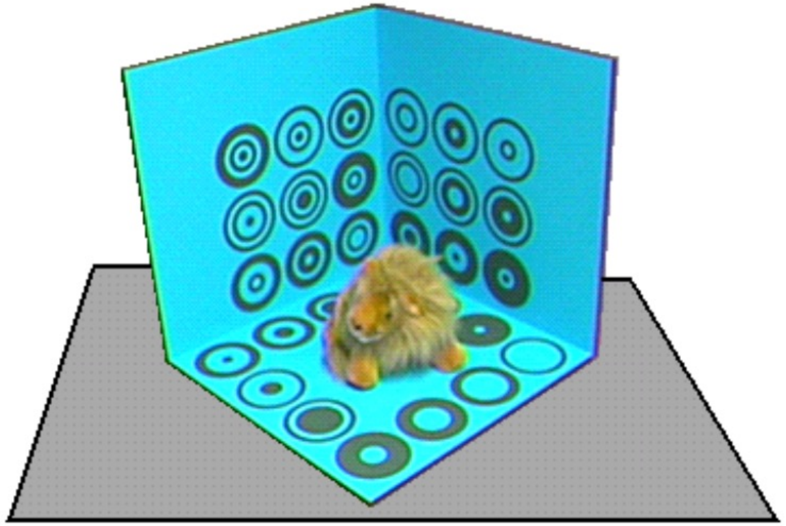
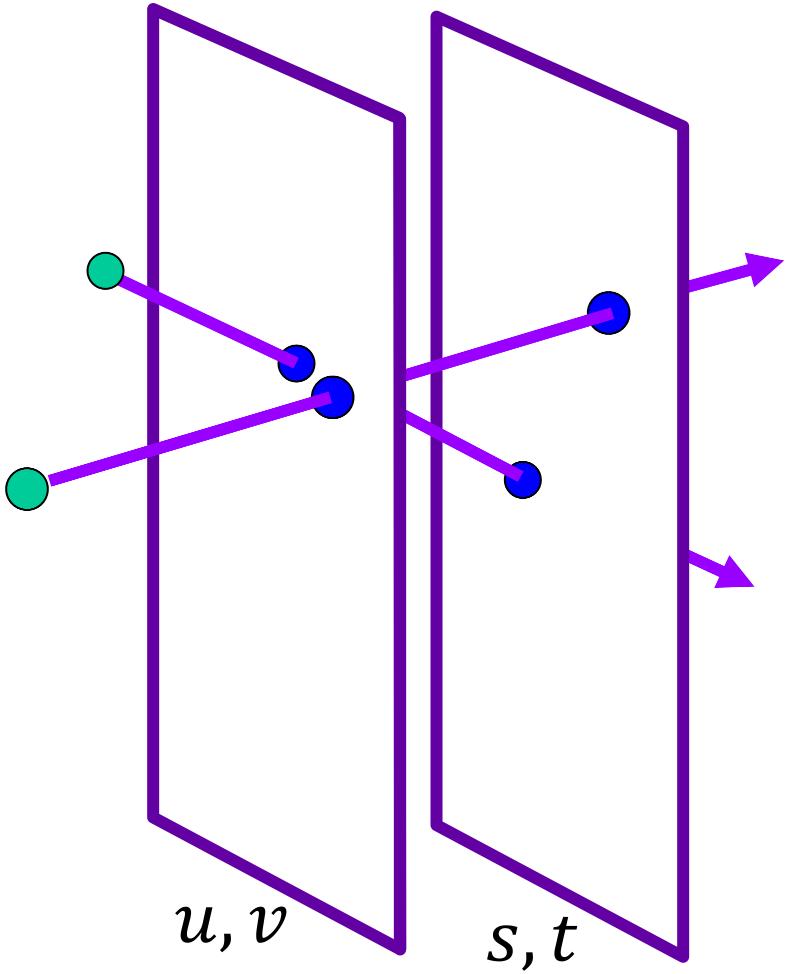


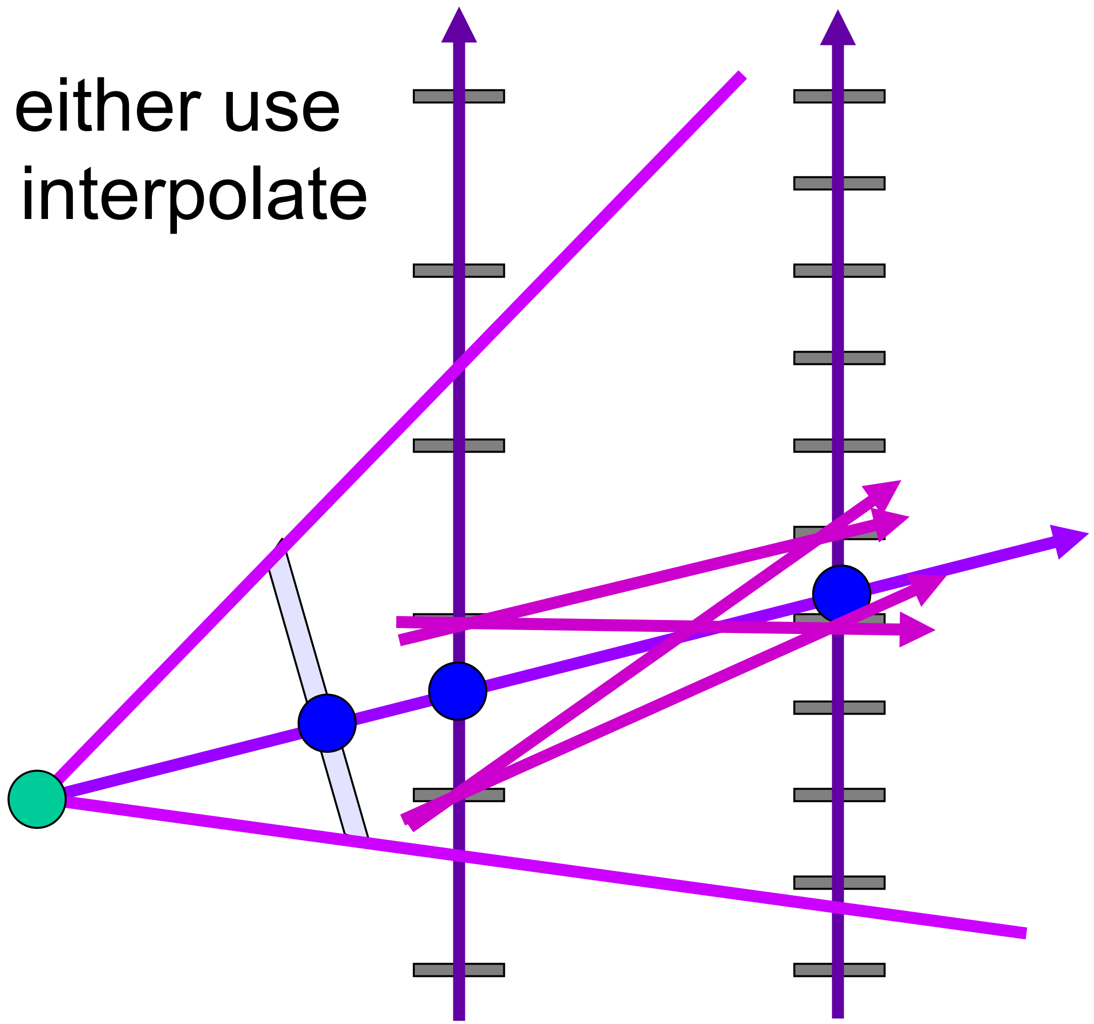
Figure 10: The capture stage

Figure source: S. Gortler et al.



Novel view synthesis

- For each output pixel, determine s, t, u, v, then either use closest discrete RGB or interpolate several nearby values



Slide by Rick Szeliski and Michael Cohen

Outline

- The plenoptic function
- Two-plane light fields

Outline

- The plenoptic function
- Two-plane light fields
- Neural radiance fields (NeRFs)

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

ECCV 2020 (best paper honorable mention)

Ben Mildenhall*

UC Berkeley

Pratul Srinivasan*

UC Berkeley

Matt Tancik*

UC Berkeley

Jon Barron

Google Research

Ravi Ramamoorthi

UC San Diego

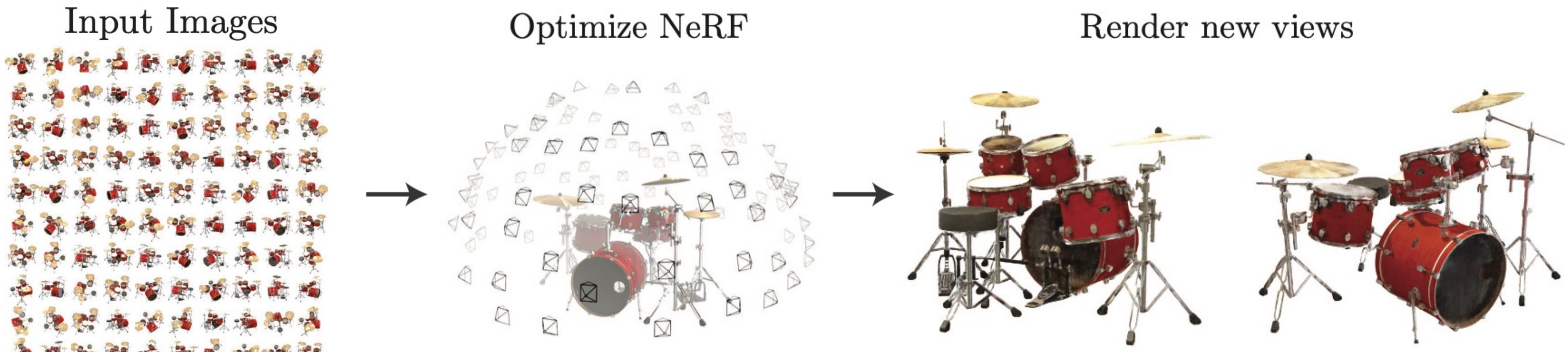
Ren Ng

UC Berkeley

https://www.matthewtancik.com/nerf

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

ECCV 2020 (best paper honorable mention)



Train a neural network to represent the plenoptic function

Inputs: sparsely sampled images of scene

Outputs: *new* views of same scene

tancik.com/nerf

Slide credit: Jon Barron

Neural radiance field

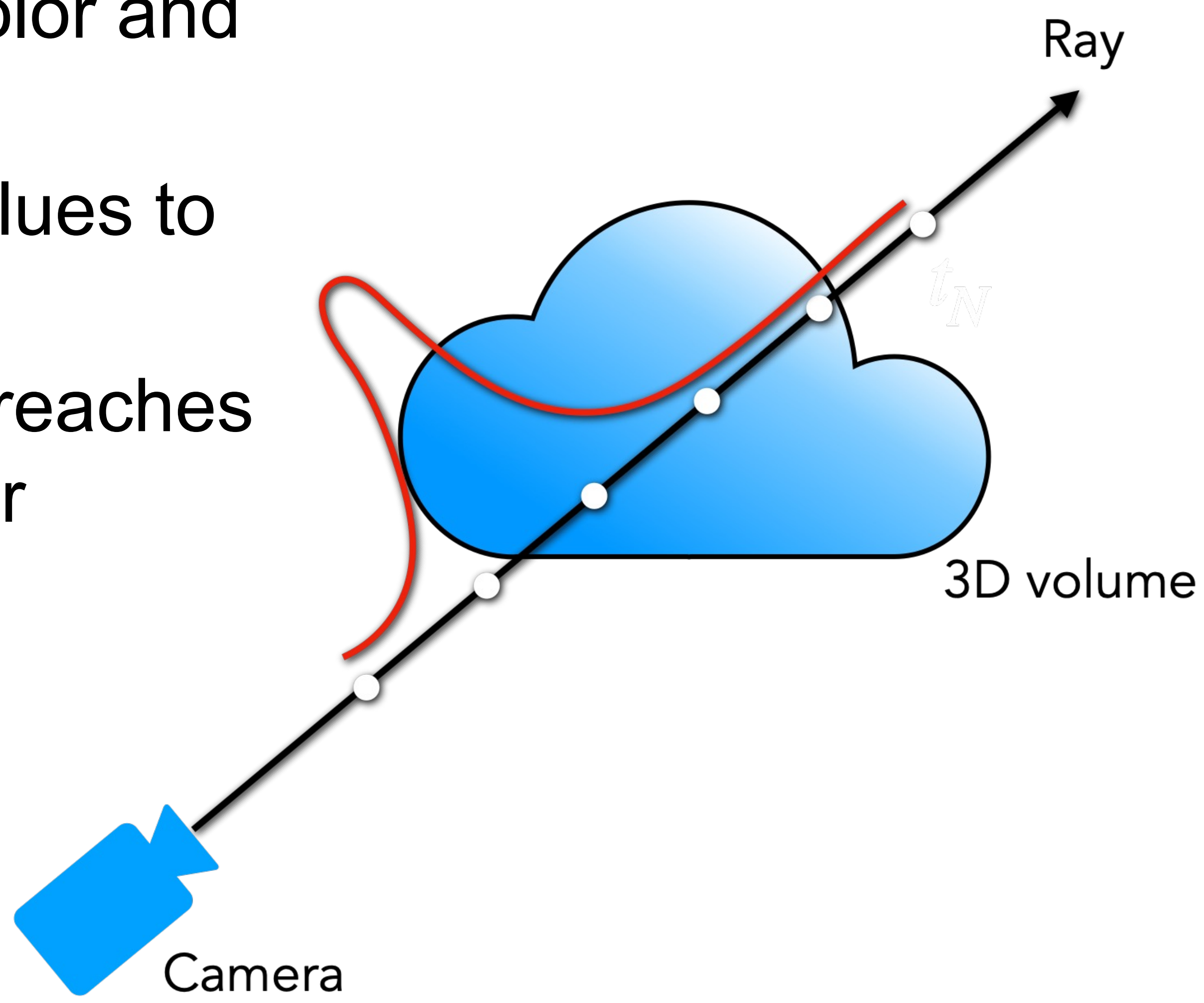
$$(\underbrace{x, y, z}_{\text{Spatial location}}, \underbrace{\theta, \phi}_{\text{Viewing direction}}) \rightarrow F_{\Omega} \rightarrow (\underbrace{r, g, b}_{\text{Output color}}, \underbrace{\sigma}_{\text{Output density}})$$

MLP
9 layers,
256 channels

Volumetric “fog” model: Every 5D input gets mapped to **Color** and **Density**

NeRF rendering

- At every point you know color and density: (c_i, σ_i)
- Need to integrate these values to render a pixel
- Idea: sum how much light reaches each point * visibility * color



How to render a pixel: Volume rendering

Given: a ray $\mathbf{r}(i) = \vec{o} + i\vec{d}$

At every point you know: (c_i, σ_i)

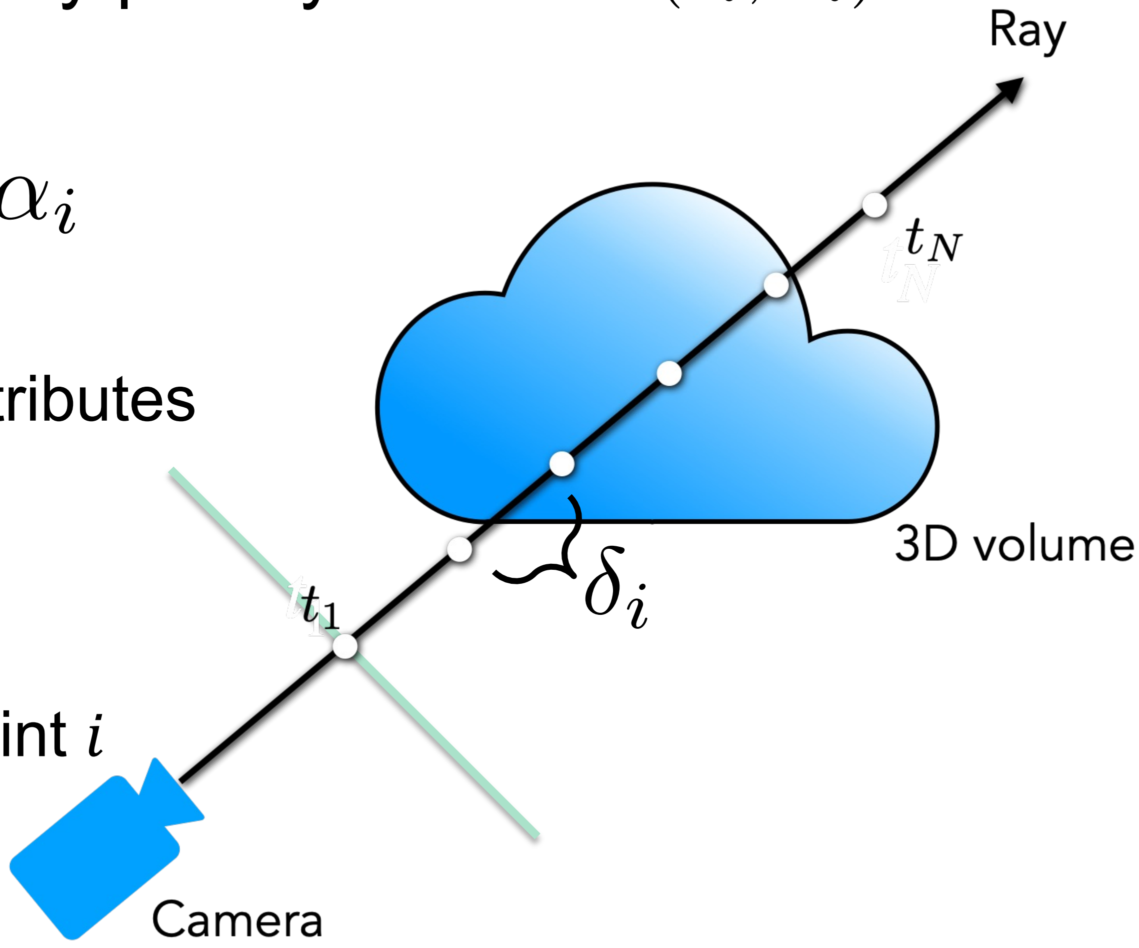
$$C(\mathbf{r}) \approx \sum_i^N w_i c_i \qquad w_i = T_i \alpha_i$$

Alpha: How much light a ray segment contributes

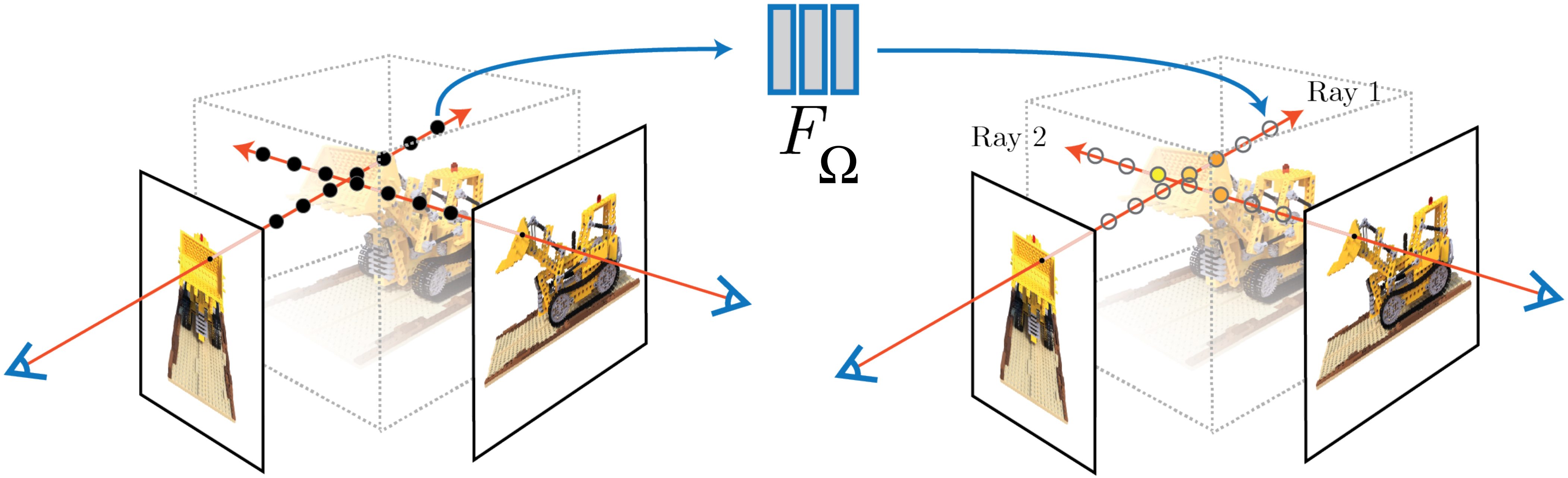
$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$

Transmittance: how much light reaches point i

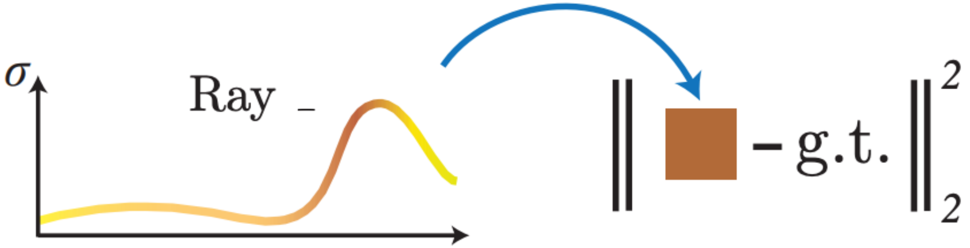
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$



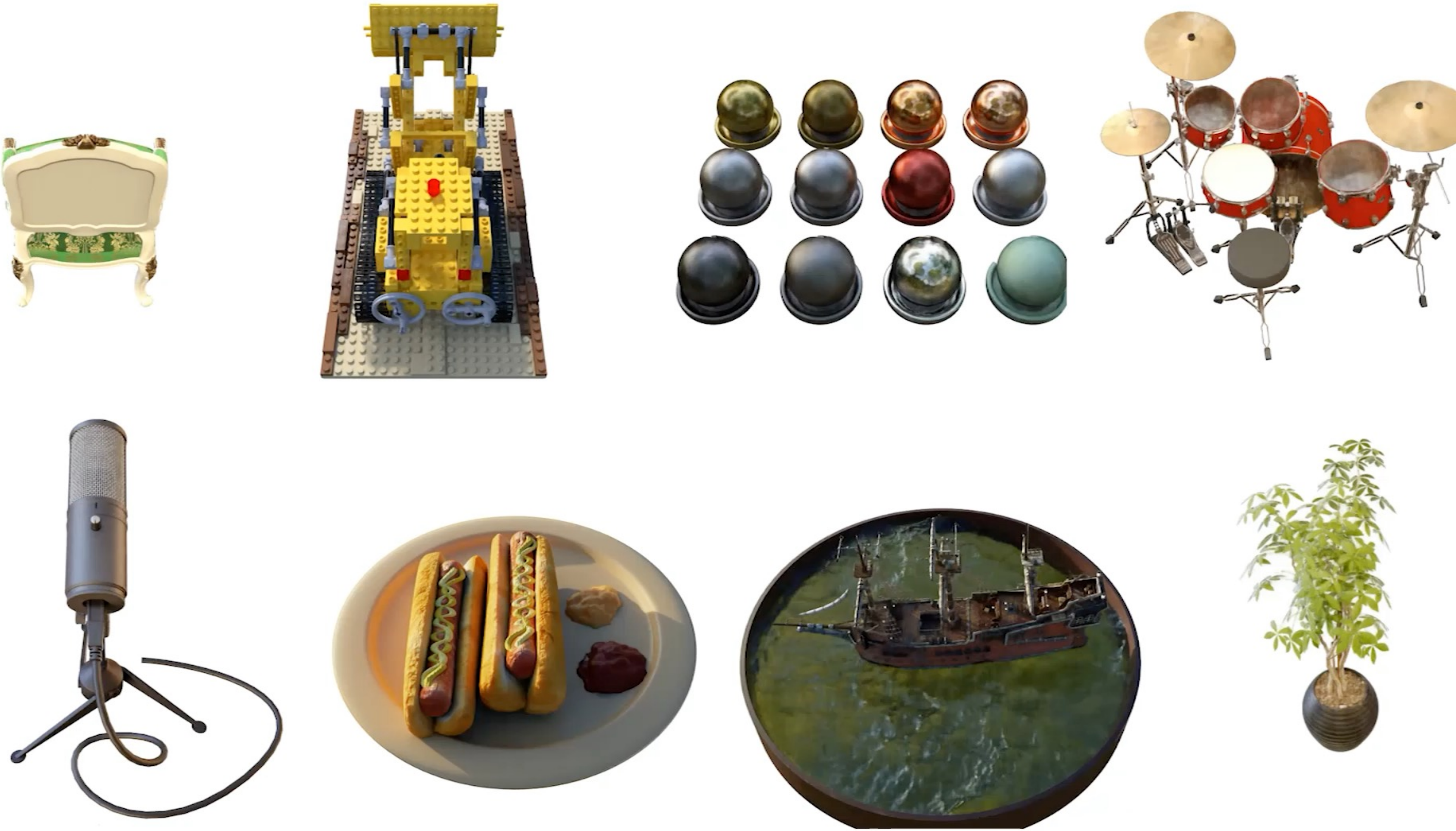
Training: Optimization with reconstruction loss



$$\min_{\Omega} \sum_i \|\text{render}^{(i)}(F_\Omega) - I_{\text{gt}}^{(i)}\|^2$$

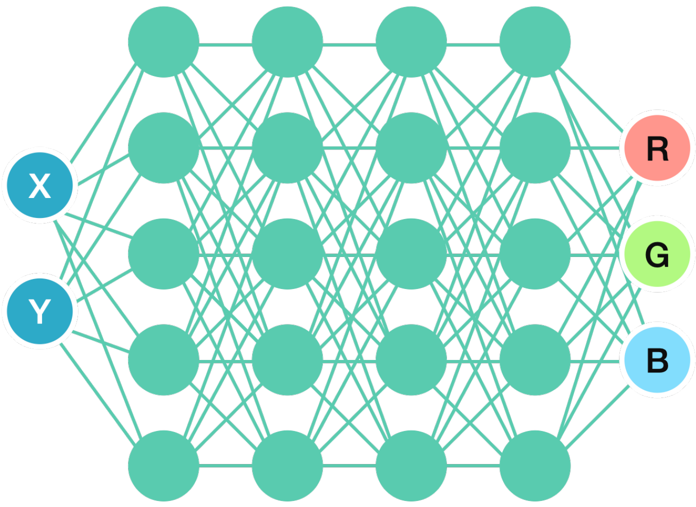


Example results



Positional Encoding

- As is, MLPs don't like to represent high-frequency functions



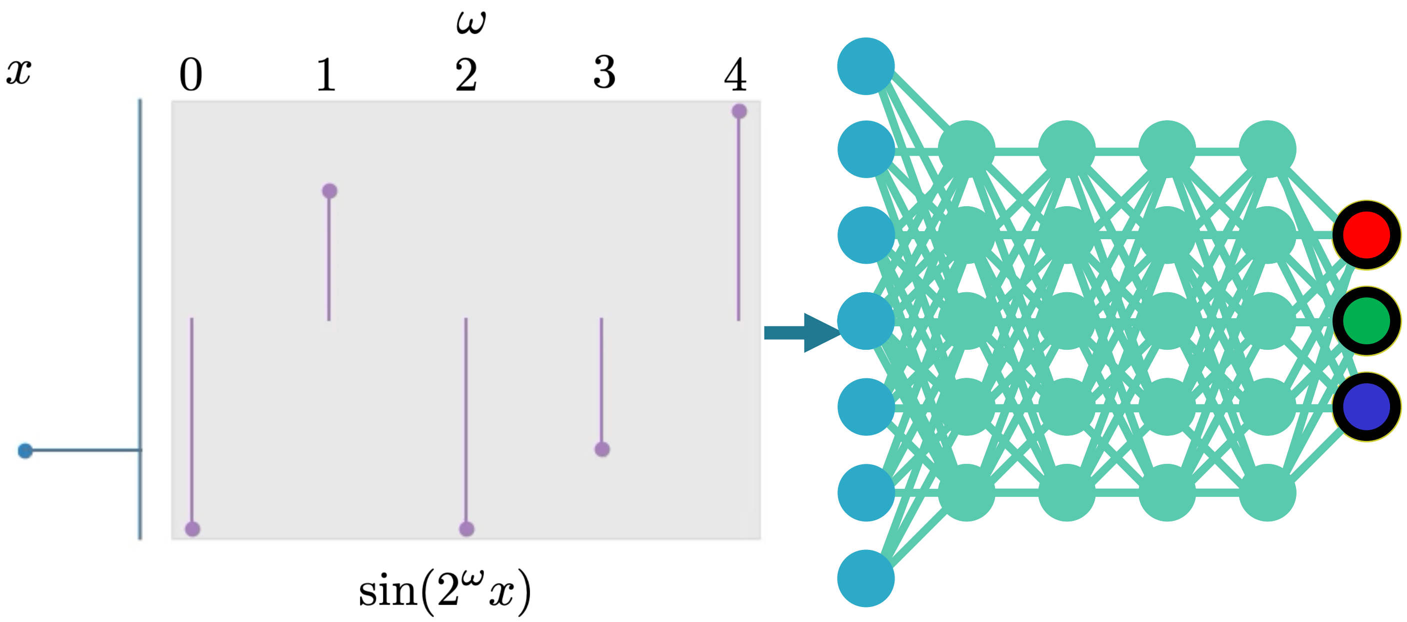
MLP output

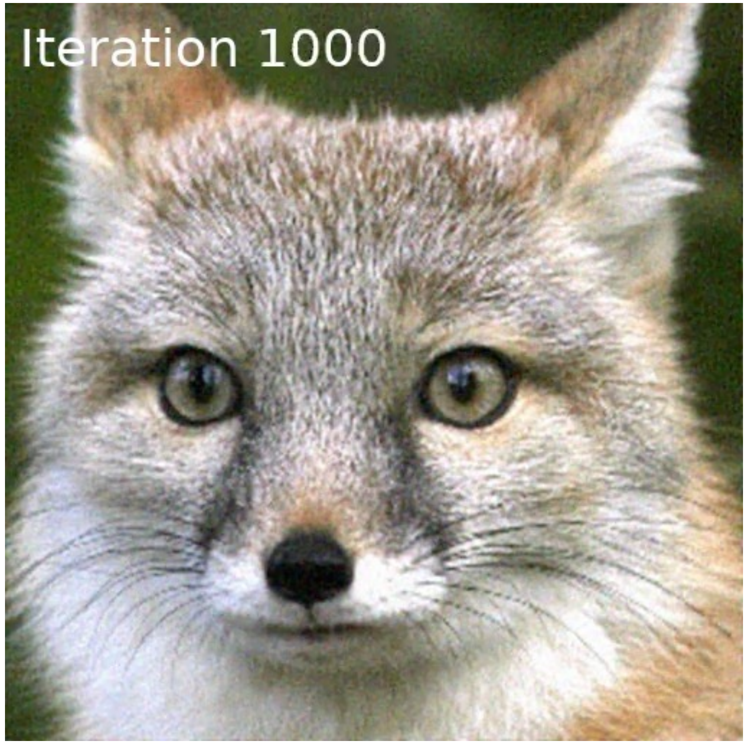
Supervision image

Rahaman et al. 2019, Basri et al. 2020

Positional Encoding

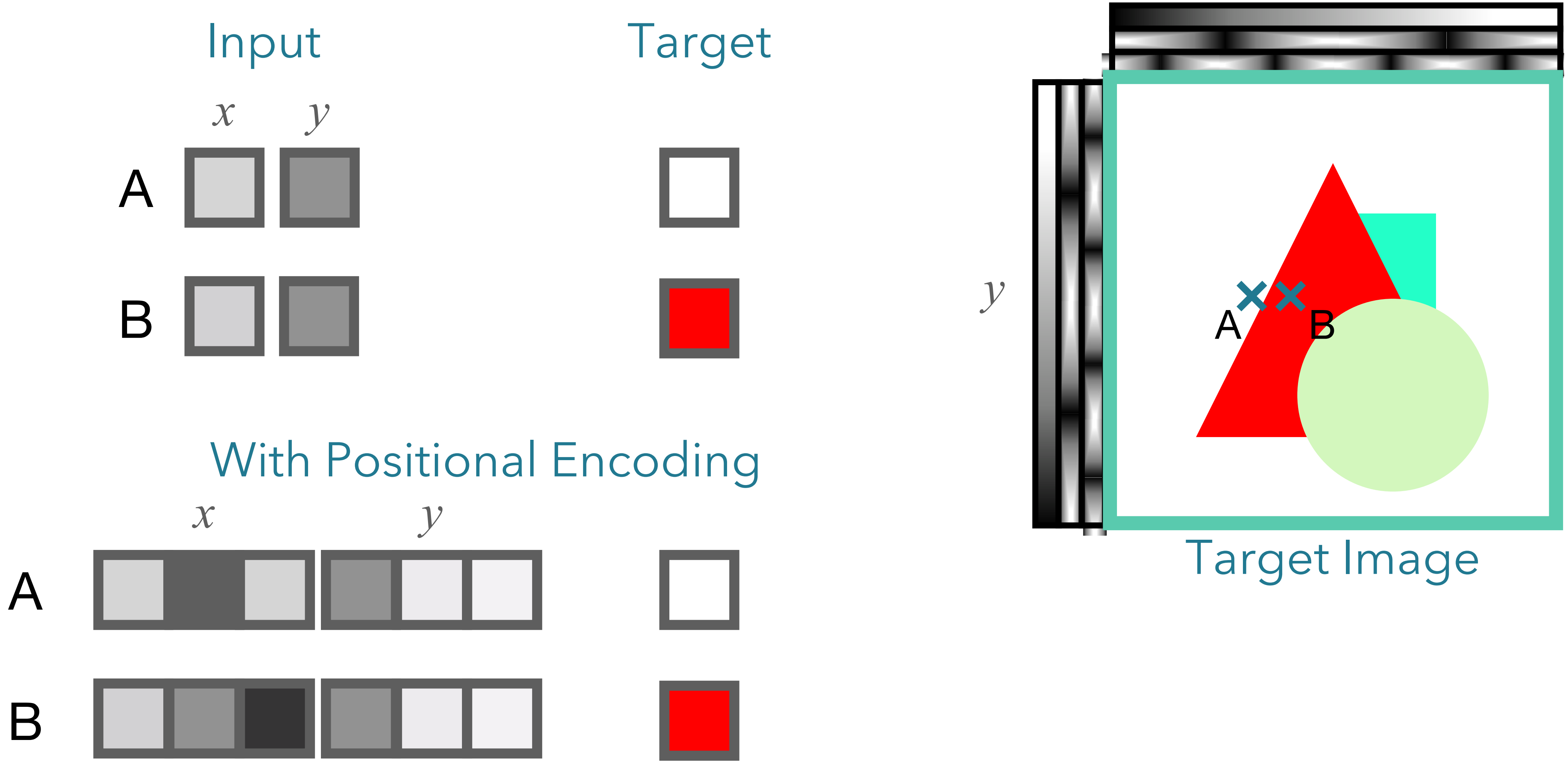
- Help the MLP by providing it with high-frequency transformations of input coordinates
- $\gamma(p) = \left(\sin(2^0\pi p), \cos(2^0\pi p), \cdots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)\right)$



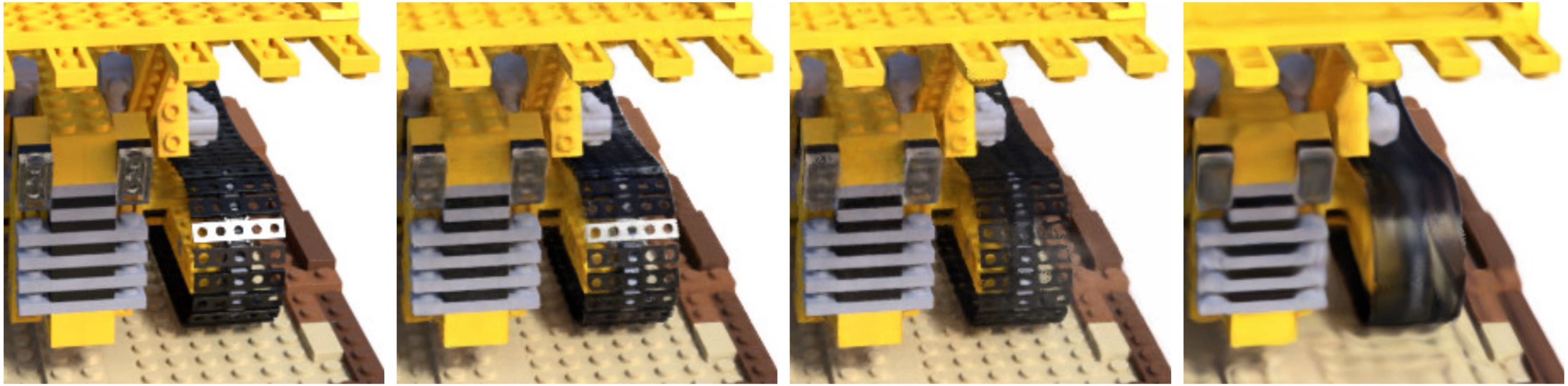


Rahaman et al. 2019, Basri et al. 2020

Why does positional encoding help?

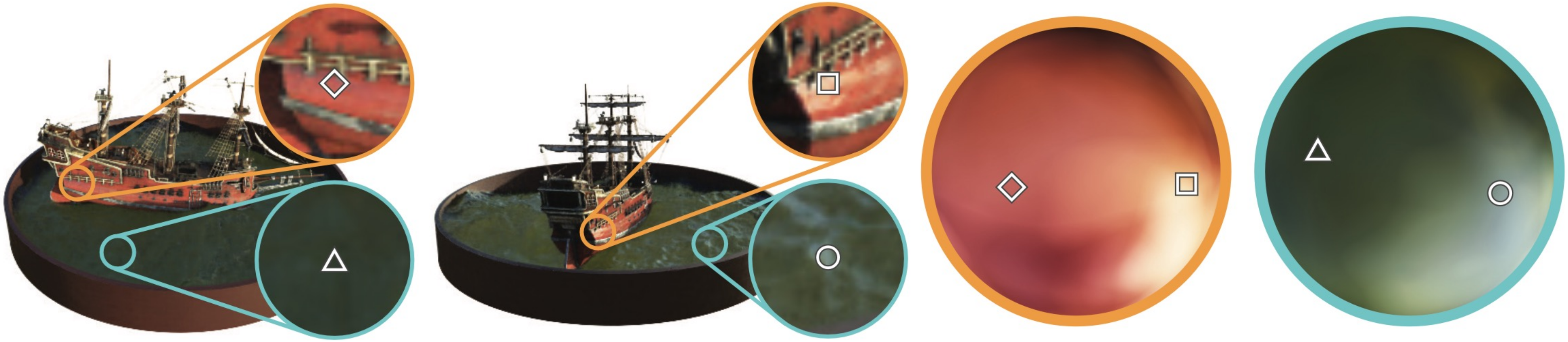


Positional Encoding (Results)



Ground Truth | Complete Model | No View Dependence | No Positional Encoding

View Dependent Emitted Radiance



(a) View 1 (b) View 2 (c) Radiance Distributions

Viewpoint-dependent effects

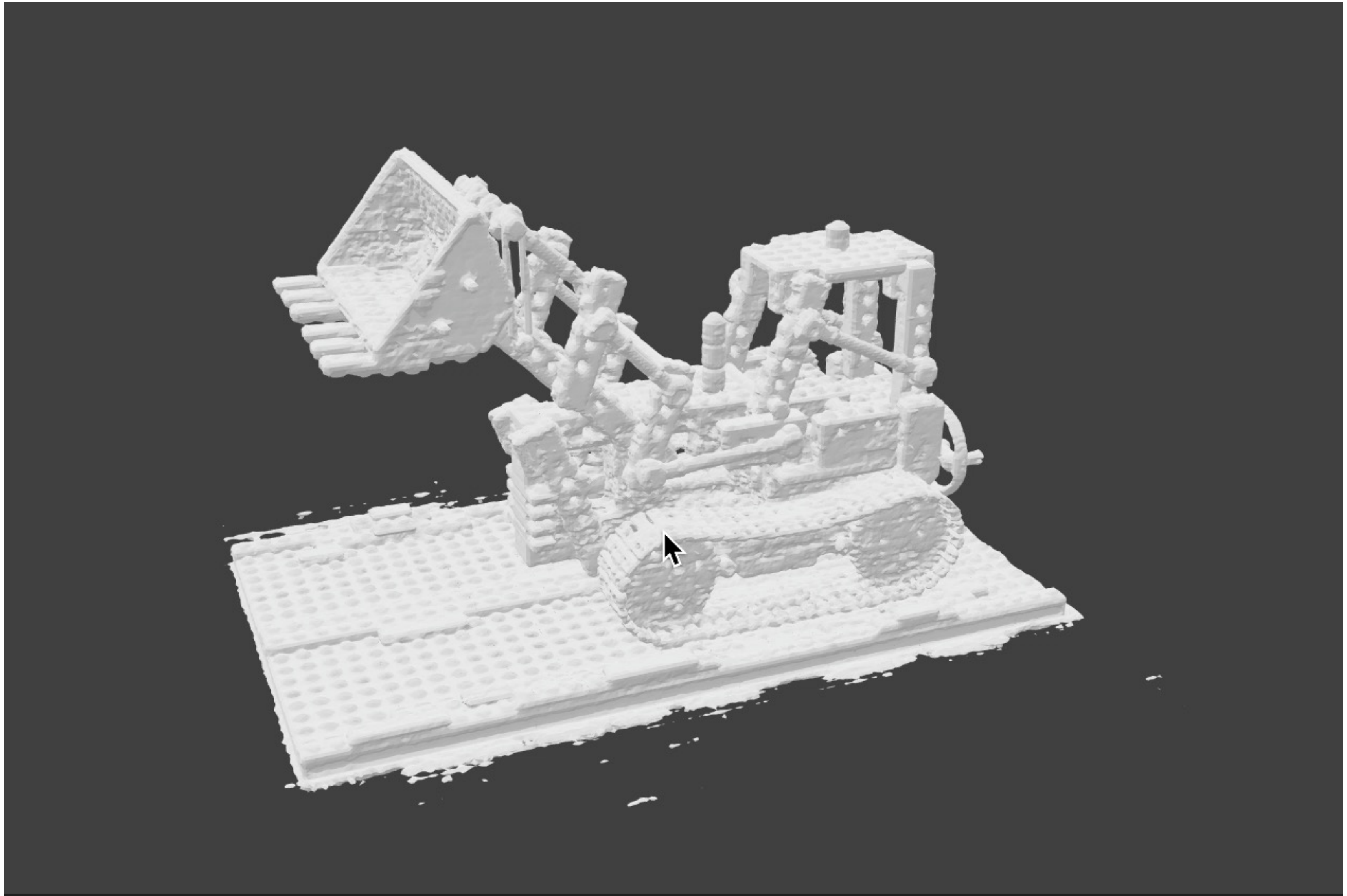
Viewpoint-dependent effects


Rendering expected depth

$$d(\mathbf{r}) \approx \sum_{i}^{N} T_i \alpha_i z_i$$


Because it models the entire plenoptic function you can insert objects with proper occlusion effects (in contrast to lightfields)

Extract surface on high density regions



NeRF limitations

- Expensive / slow to train and render
- Sensitive to sampling strategy
- Does not generalize between scenes
- Sensitive to pose accuracy
- Assumes static scene
- Assumes static lighting and camera focus
- Not a mesh

NeRF explosion

- ▶ Faster Inference
- ▶ Faster Training
- ▶ Unconstrained Images
- ▶ Deformable
- ▶ Video
- ▶ Generalization
- ▶ Pose Estimation
- ▶ Lighting
- ▶ Compositionality
- ▶ Scene Labelling and Understanding
- ▶ Editing
- ▶ Object Category Modeling
- ▶ Multi-scale
- ▶ Model Reconstruction
- ▶ Depth Estimation
- ▶ Robotics
- ▶ Large-scale scene

▼ Faster Training

- Depth-supervised NeRF: Fewer Views and Faster Training for Free, Deng et al., Arxiv 2021 | github | bibtex
- Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction, Sun et al., CVPR 2022 | github | bibtex
- Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, Müller et al., SIGGRAPH 2022 | github | bibtex
- Plenoxels Radiance Fields without Neural Networks, Yu et al., CVPR 2022 | github | bibtex
- TensoRF: Tensorial Radiance Fields, Chen et al., ECCV 2022 | github | bibtex
- BakedSDF: Meshing Neural SDFs for Real-Time View Synthesis, Yariv et al., Arxiv 2023 | bibtex

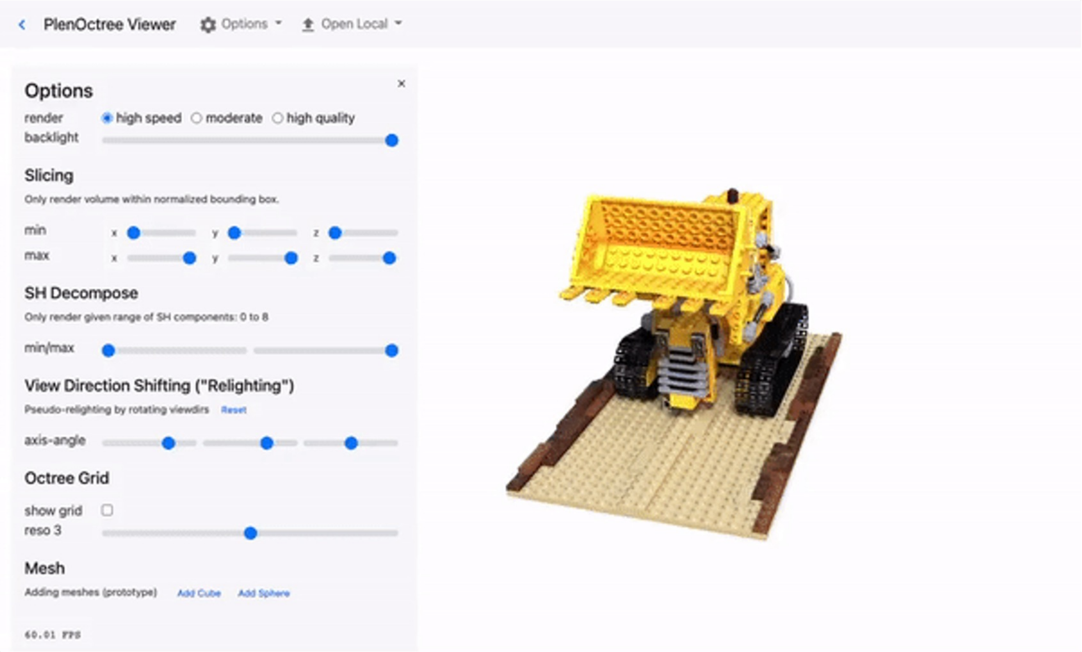
▼ Deformable

- Deformable Neural Radiance Fields, Park et al., Arxiv 2020 | github | bibtex
- D-NeRF: Neural Radiance Fields for Dynamic Scenes, Pumarola et al., CVPR 2021 | github | bibtex
- Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction, Gafni et al., CVPR 2021 | github | bibtex
- Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Deforming Scene from Monocular Video, Tretschk et al., Arxiv 2020 | github | bibtex
- PVA: Pixel-aligned Volumetric Avatars, Raj et al., CVPR 2021 | bibtex
- Neural Articulated Radiance Field, Noguchi et al., Arxiv 2021 | github | bibtex
- CLA-NeRF: Category-Level Articulated Neural Radiance Field, Tseng et al., ICRA 2022 | bibtex

https://github.com/yenchenlin/awesome-NeRF

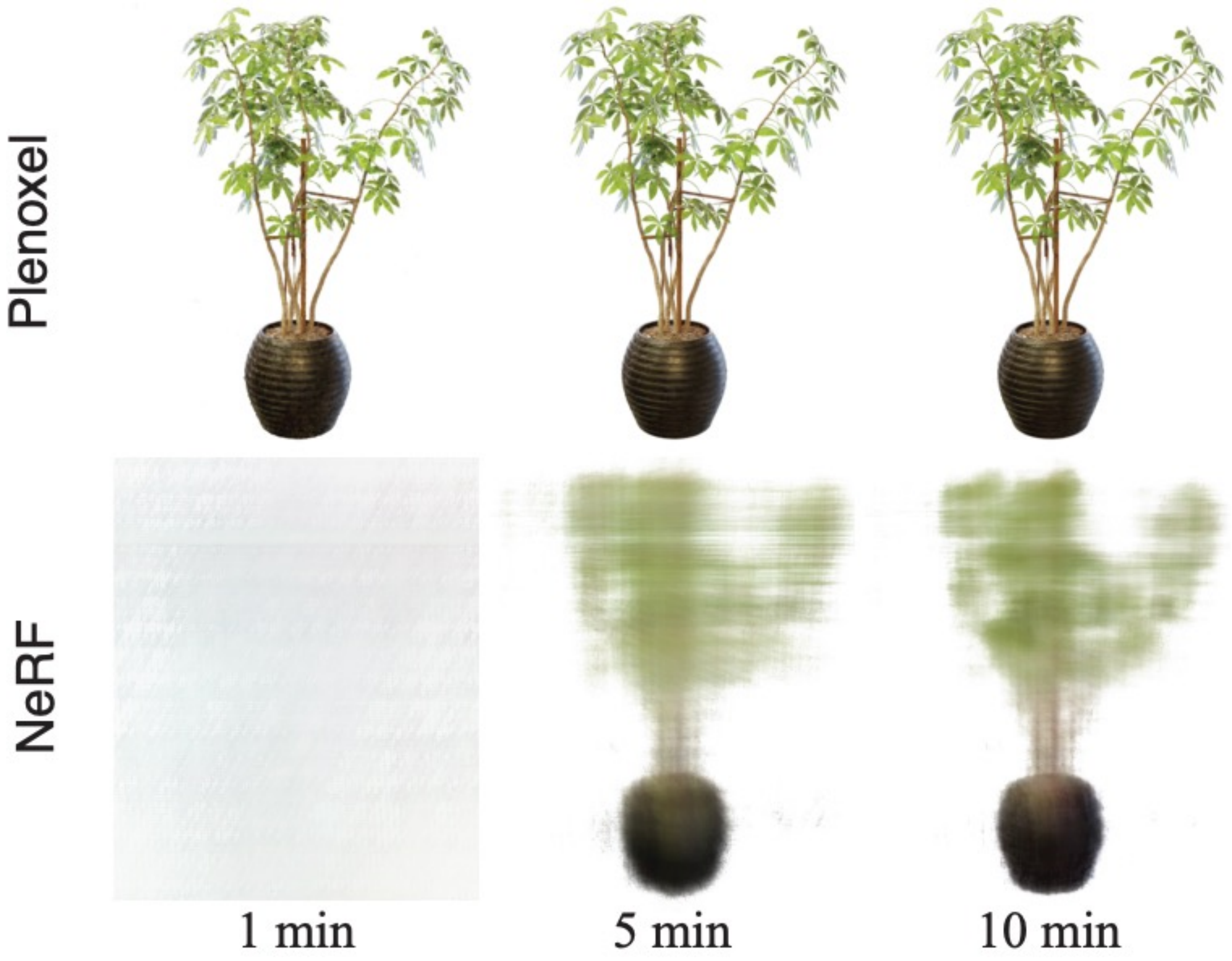
Fast Inference

- PlenOctrees [Yu et al. ICCV'21]
- SNeRG [Hedman et al. ICCV'21]
- FastNeRF [Garbin et al. ICCV'21]
- KiloNeRF [Reiser et al. ICCV'21]
- AutoInt [Lindell et al. CVPR'21]
- …



PlenOctrees [Yu et al. ICCV'21]

Plenoxels



S. Fridovich-Kiel et al. Plenoxels: Radiance Fields without Neural Networks. CVPR 2022

Plenoxels

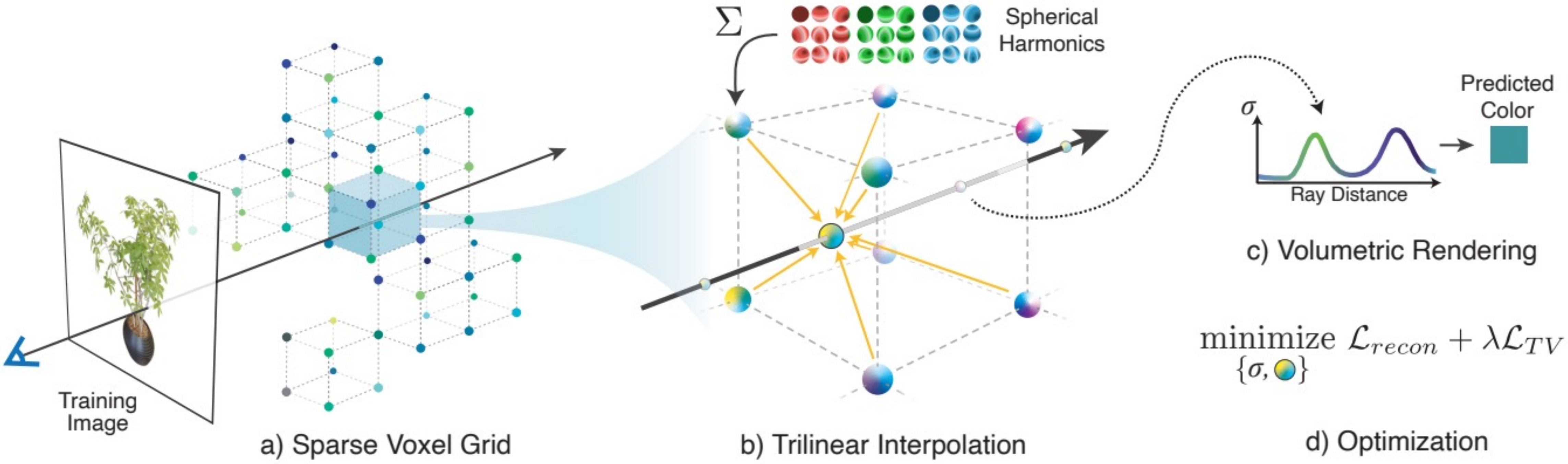
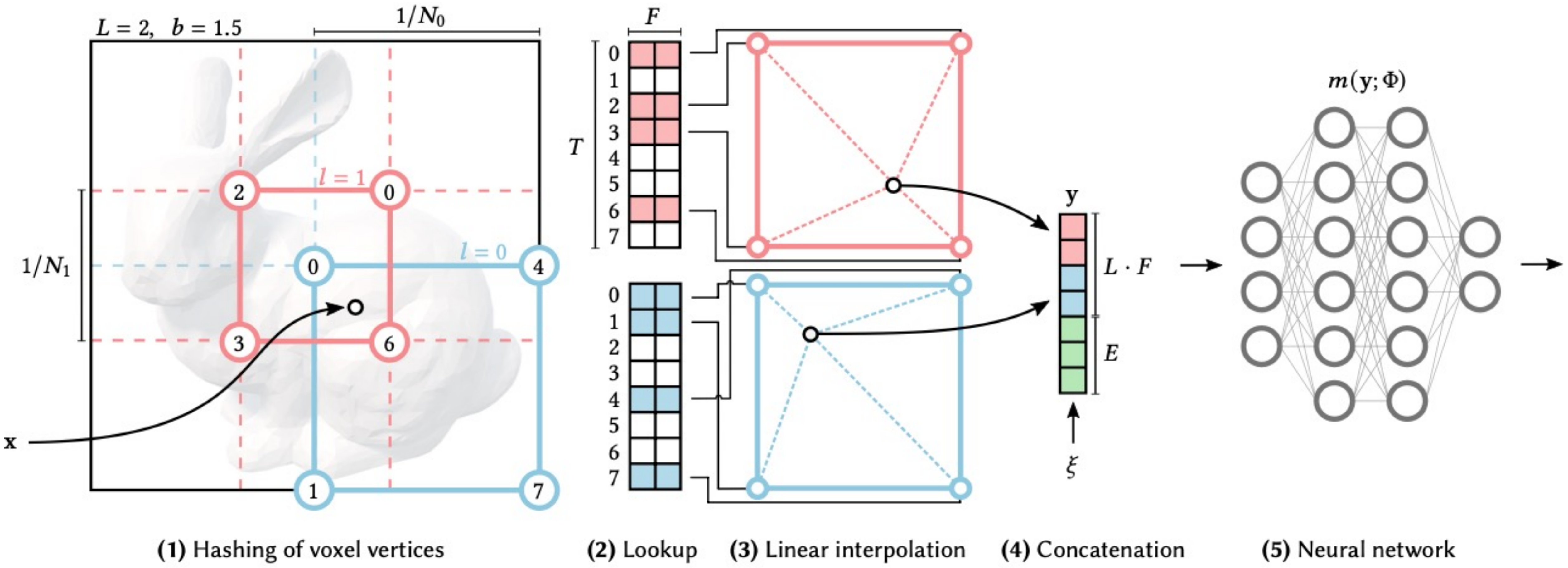


Figure 2. **Overview of our sparse Plenoxel model.** Given a set of images of an object or scene, we reconstruct a (a) sparse voxel ("Plenoxel") grid with density and spherical harmonic coefficients at each voxel. To render a ray, we (b) compute the color and opacity of each sample point via trilinear interpolation of the neighboring voxel coefficients. We integrate the color and opacity of these samples using (c) differentiable volume rendering, following the recent success of NeRF [26]. The voxel coefficients can then be (d) optimized using the standard MSE reconstruction loss relative to the training images, along with a total variation regularizer.

S. Fridovich-Kiel et al. Plenoxels: Radiance Fields without Neural Networks. CVPR 2022

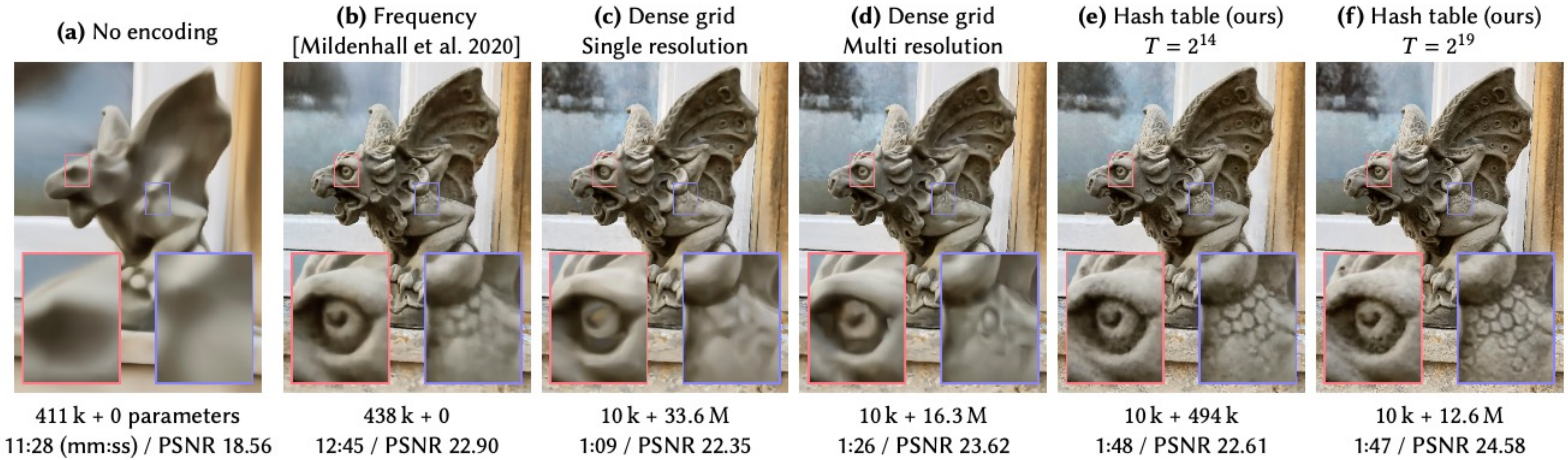
Instant NGP

- Multi-resolution hash encoding to use sparse feature maps
- Let NN deal with collisions at higher resolutions



Müller et al. **[Instant neural graphics primitives with a multiresolution hash encoding](#)**. SIGGRAPH 2022

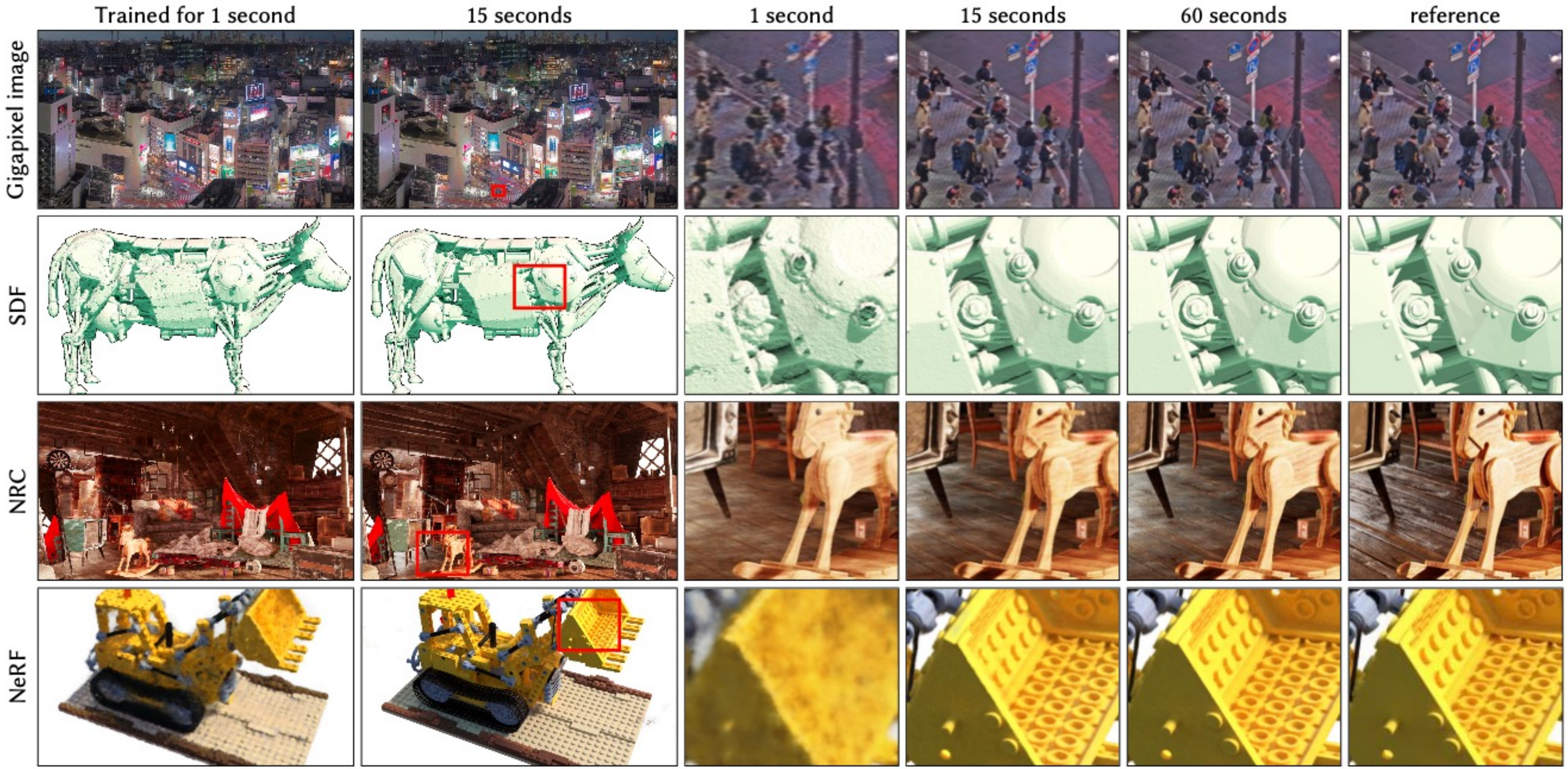
Instant NGP



Müller et al. **Instant neural graphics primitives with a multiresolution hash encoding**. SIGGRAPH 2022

Instant NGP

- Applicable to many other implicit functions



Müller et al. **[Instant neural graphics primitives with a multiresolution hash encoding](#)**. SIGGRAPH 2022