Neural Radiance Fields

Video from the original ECCV'20 paper

CS180/280A: Intro to Computer Vision and Comp. Photography

Angjoo Kanazawa, Alexei Efros, UC Berkeley Fall 2025

Lots of content from Noah Snavely and [ECCV 2022 Tutorial on Neural Volumetric Rendering for Computer Vision]

Where’s Angjoo?

Who am I?

Justin Kerr! 5th year PhD

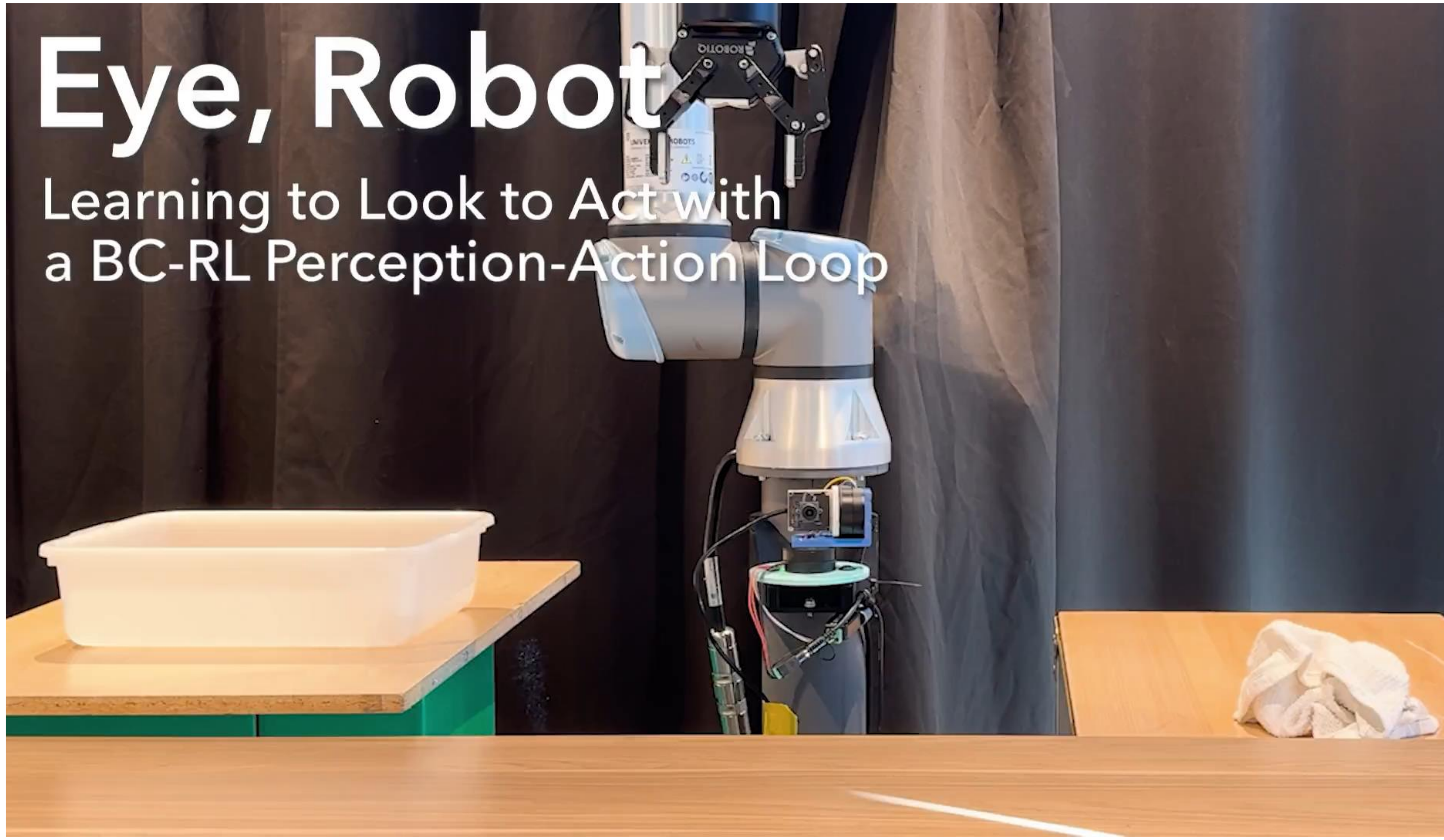
Angjoo Kanazawa Ken Goldberg

My advisors

Who am I?

My research: (roughly speaking) vision and robotics

NeRF + Language



Robot Eyeball

Neural Radiance Fields

Video from the original ECCV’20 paper

Today: A Gentle Birds Eye View Intro

Birds Eye View

- What is NeRF trying to solve?
- How is it different or similar to existing approaches?
- What is its historical context?
- How does NeRF represent 3D?

Problem Statement

Input:
A set of calibrated Images

Output:
A 3D scene representation that renders novel views

“Novel View Synthesis"

Input:
A set of calibrated Images

Output:
A 3D scene representation that renders novel views

A note on inputs

- Need to know the camera parameters: **extrinsic** (viewpoint) & **intrinsics** (focal length, distortion, etc)

How do we get this from images?

Structure from Motion! (last lecture)

Input: set of images.
Output: extrinsics, intrinsics, 3D points, pixel correspondences

Proc. R. Soc. Lond. B. **203**, 405–426 (1979)
Printed in Great Britain

The interpretation of structure from motion

BY S. ULLMAN

Artificial Intelligence Laboratory, Massachusetts Institute of Technology,
545 *Technology Square* (*Room* 808), *Cambridge, Massachusetts* 02139 *U.S.A.*

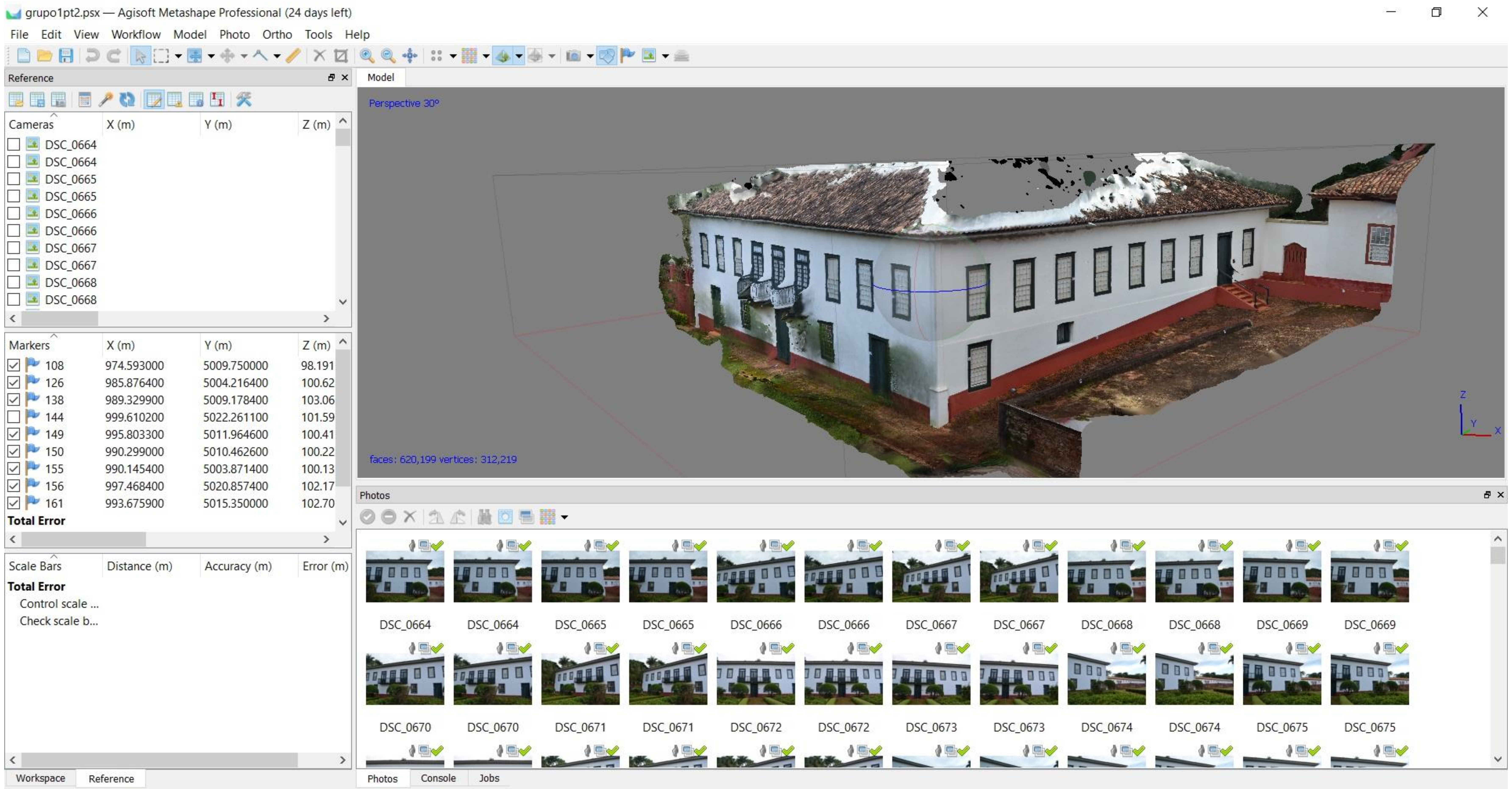
NeRF is AFTER Structure from Motion

To train NeRF you need to run SfM on images to estimate camera parameters

Colmap: Schönberger et al. 2016

What was before NeRF? "Photogrammetry"

- Problem: Given calibrated cameras, recover highly detailed 3D **surface** model
- Often the output is textured meshes



Photogrammetry

What you see for most existing 3D scanning systems, ie sketchfab, or what’s in your video game

Shortcomings of photogrammetry

Because they often model surfaces, struggles on Thin / Amorphus / Shiny objects

Enter NeRF!

nerfstudio

from Berkeley!

2020: Neural Radiance Field (NeRF) Paper

Mildenhall*, Srinivasan*, Tancik*, Barron, Ramamoorthi, Ng, ECCV 2020

Why care?

- Huge amount of impact: **14805+** citations in 5 years
- Somewhat “replaced” photogrammetry
- Lots of cool followups and applications!

What made it take off?

High quality reconstruction with view-dependent effects

Can represent non-opaque objects

What made it take off?

The “neural” part gives lots of flexibility in what it can represent

Nerf-W [Martin-Brualla et al. CVPR 2021]

What made it take off?

Huge compression efficiency… this whole scene is ~100 MB

Lots of cool things you can do with it!

City-Scale NeRFs

BlockNeRF
[Tancik et al.
CVPR 2022]

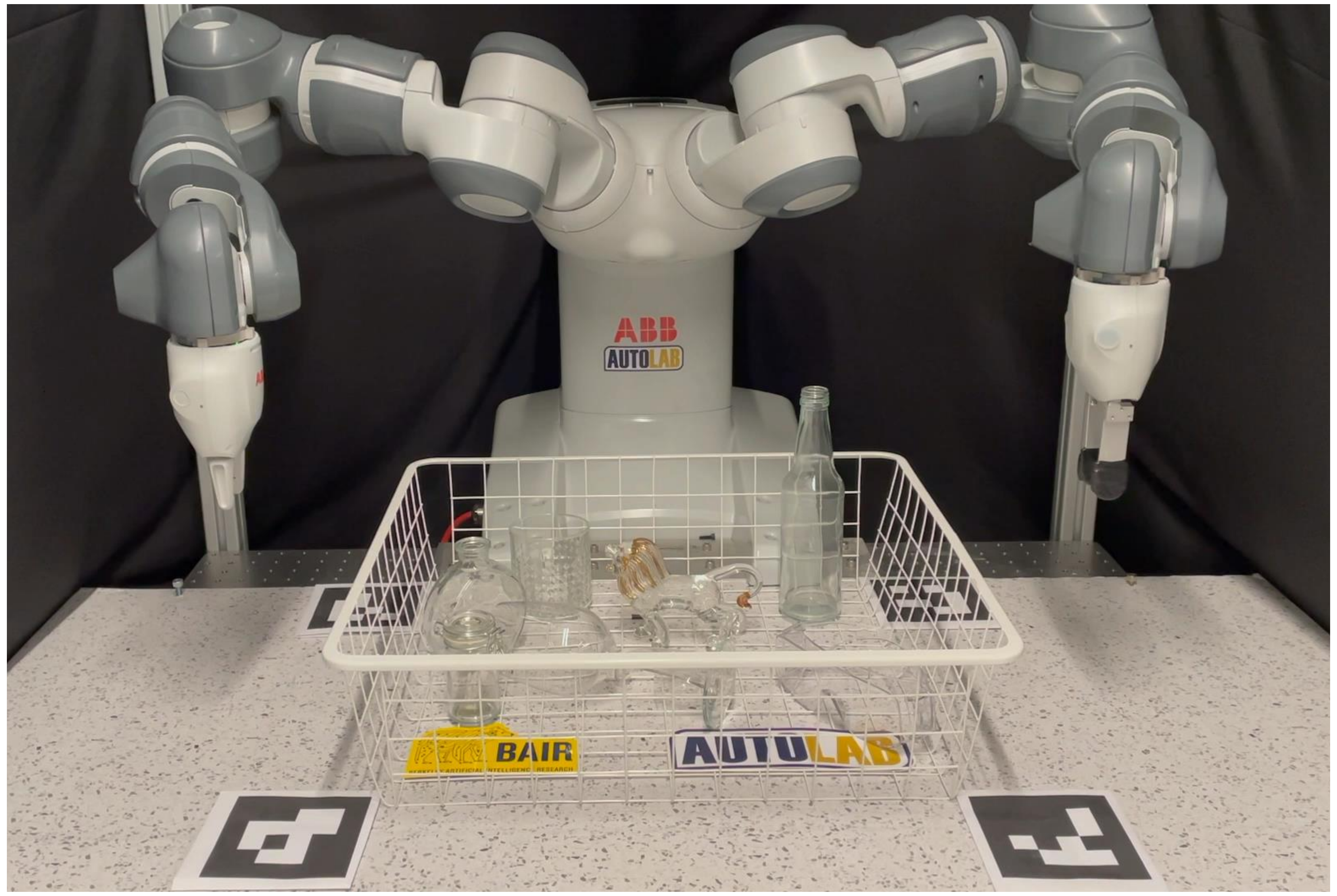
Generating 3D scenes with diffusion models

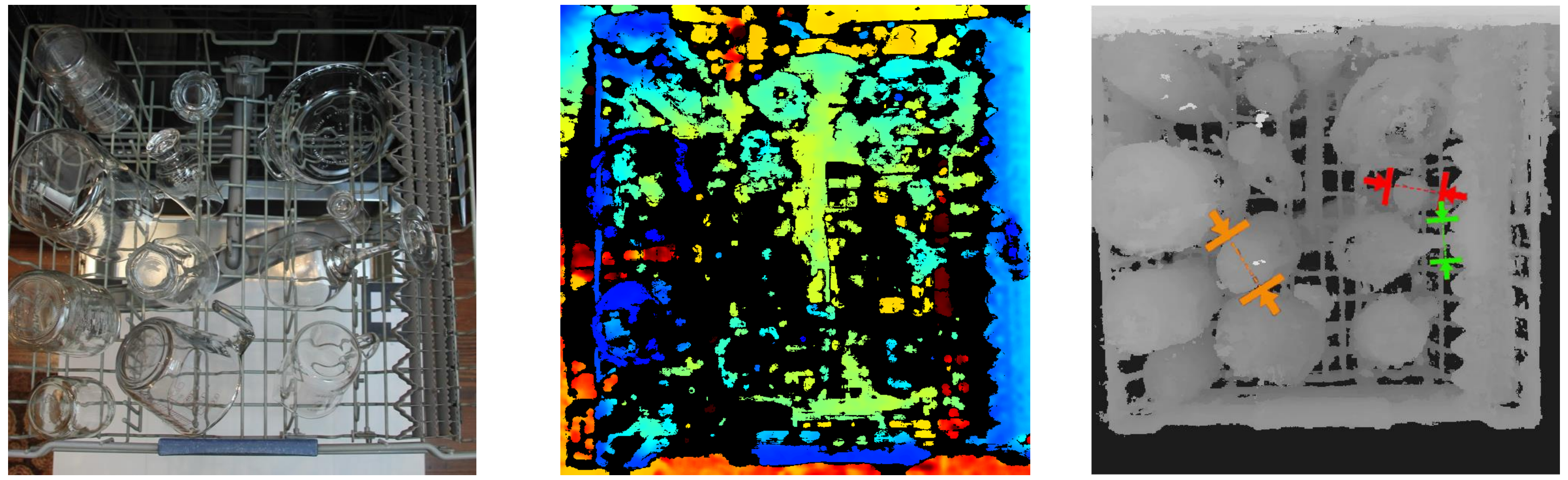
DreamFusion [Poole et al. ICLR 2023]

Generative 3D Faces

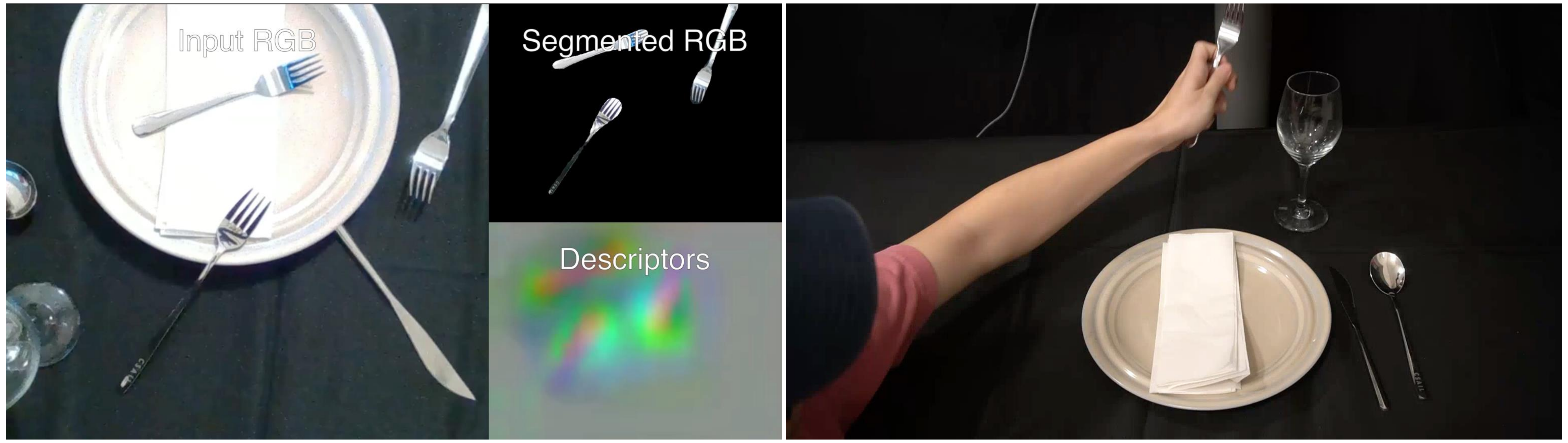
EG3D: Efficient Geometry-aware 3D Generative Adversarial Networks, Chan et al. CVPR 2022

Robotics

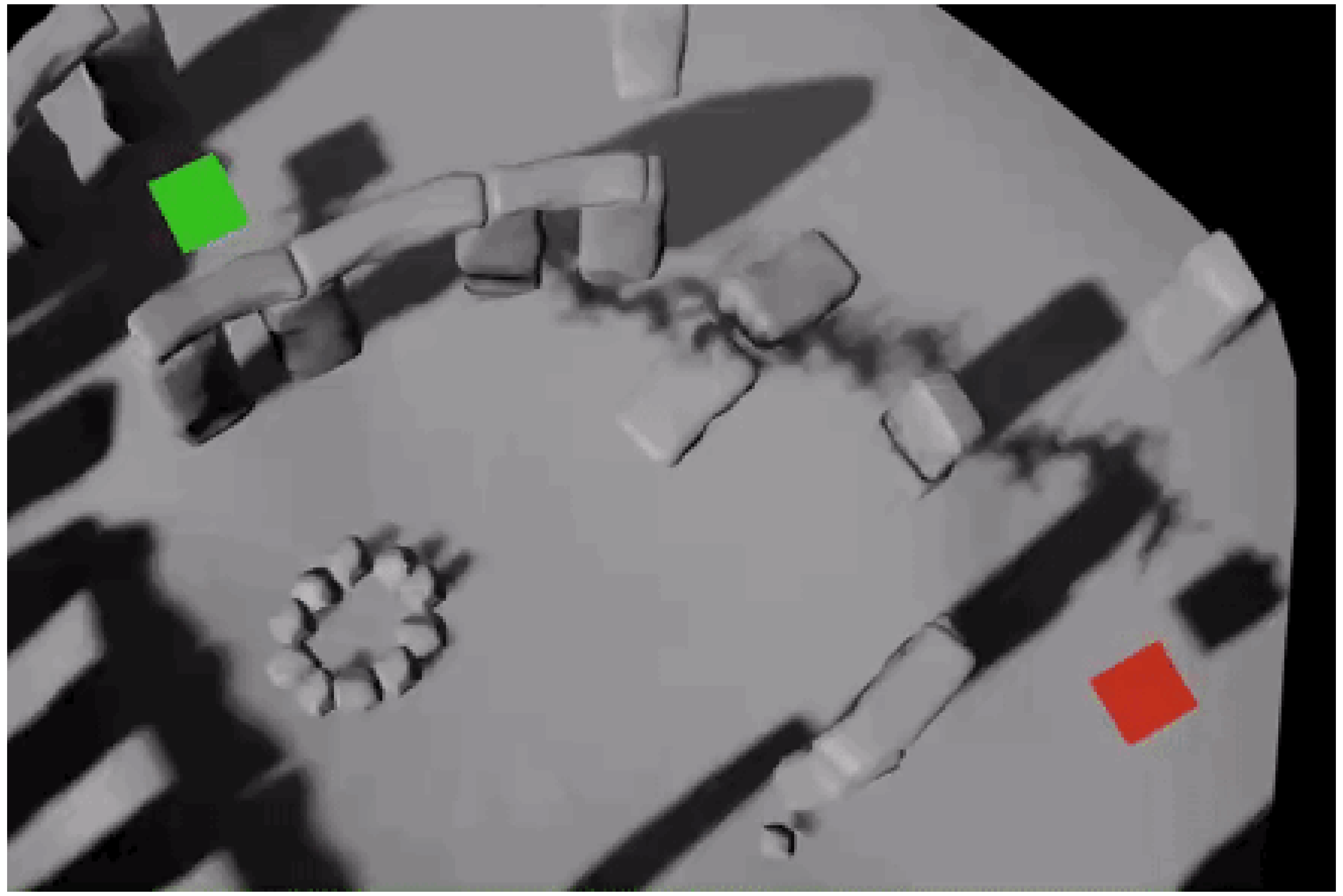




Dex-NeRF: Using a Neural Radiance field to Grasp Transparent Objects, [Ichnowski and Avigal et al. CoRL 2021]



NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields, [Yen-Chen et al. ICRA 2022]



Vision-Only Robot Navigation in a Neural Radiance World [Adamkiewicz and Chen et al. ICRA 2022]

Querying with Language

[Kerr* and Kim* et al. LERF, ICCV 2023]

…for robot grasping

[Rashid* and Sharma* et al. LERF-TOGO, CoRL 2023]

Editing with Instructions

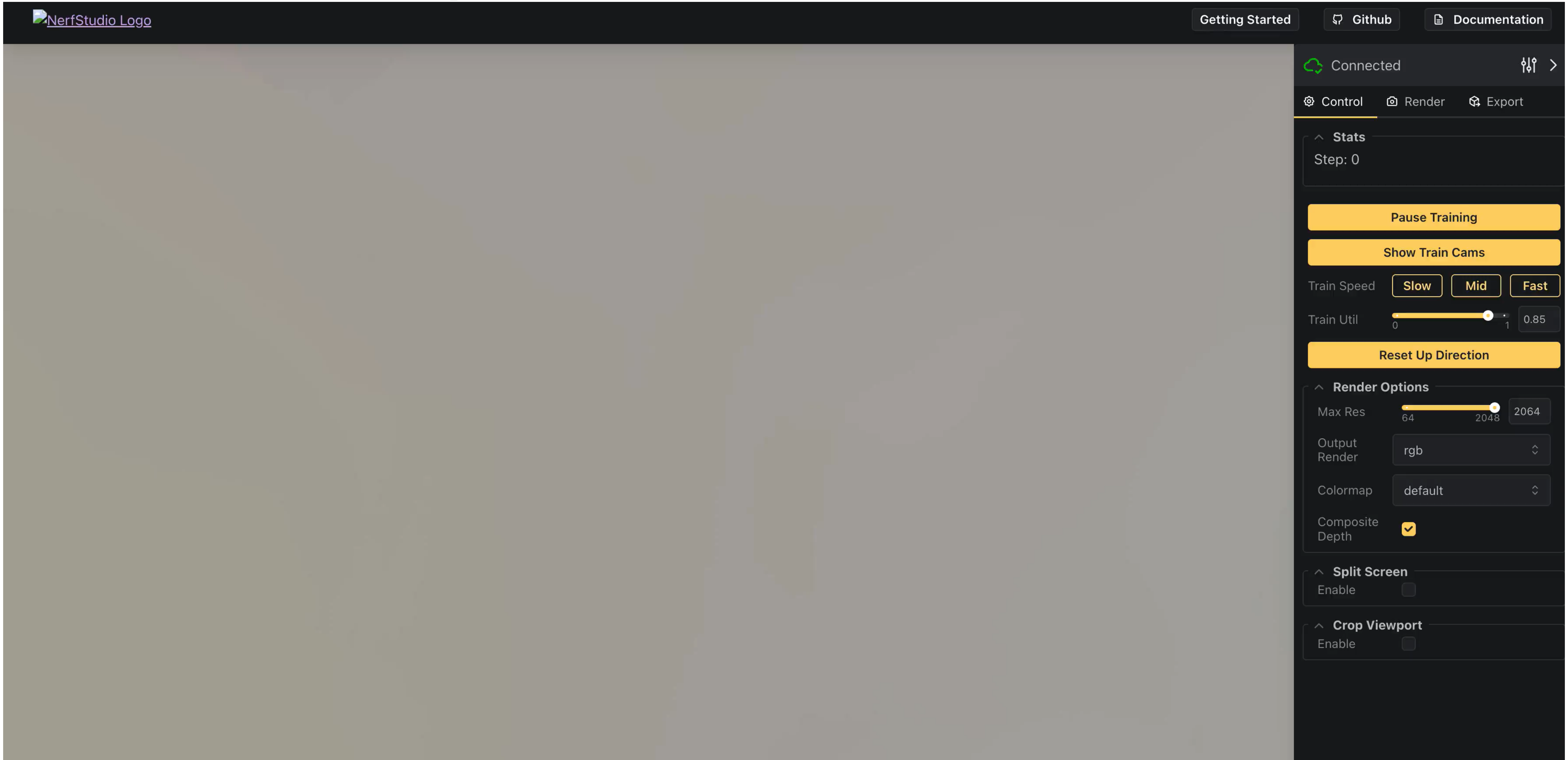
[Haque, Tancik, Efros, Holynski, Kanazawa, ICCV 2023]

Easy-to-use tools

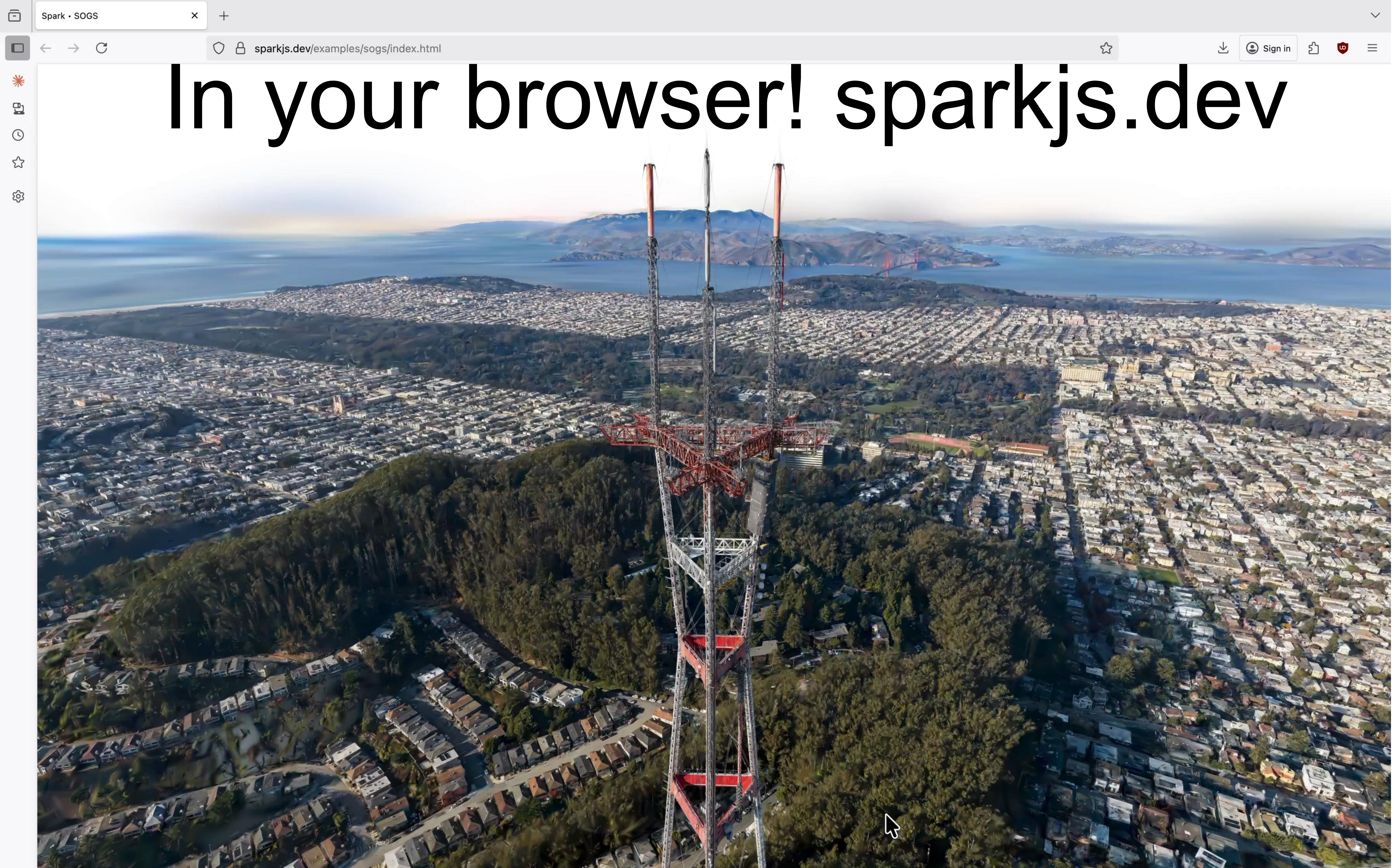

nerfstudio
Public
Fork 1.4k
Starred 10.3k
gsplat
Public
Fork 450
Star 3k

Speed: Gaussian Splats

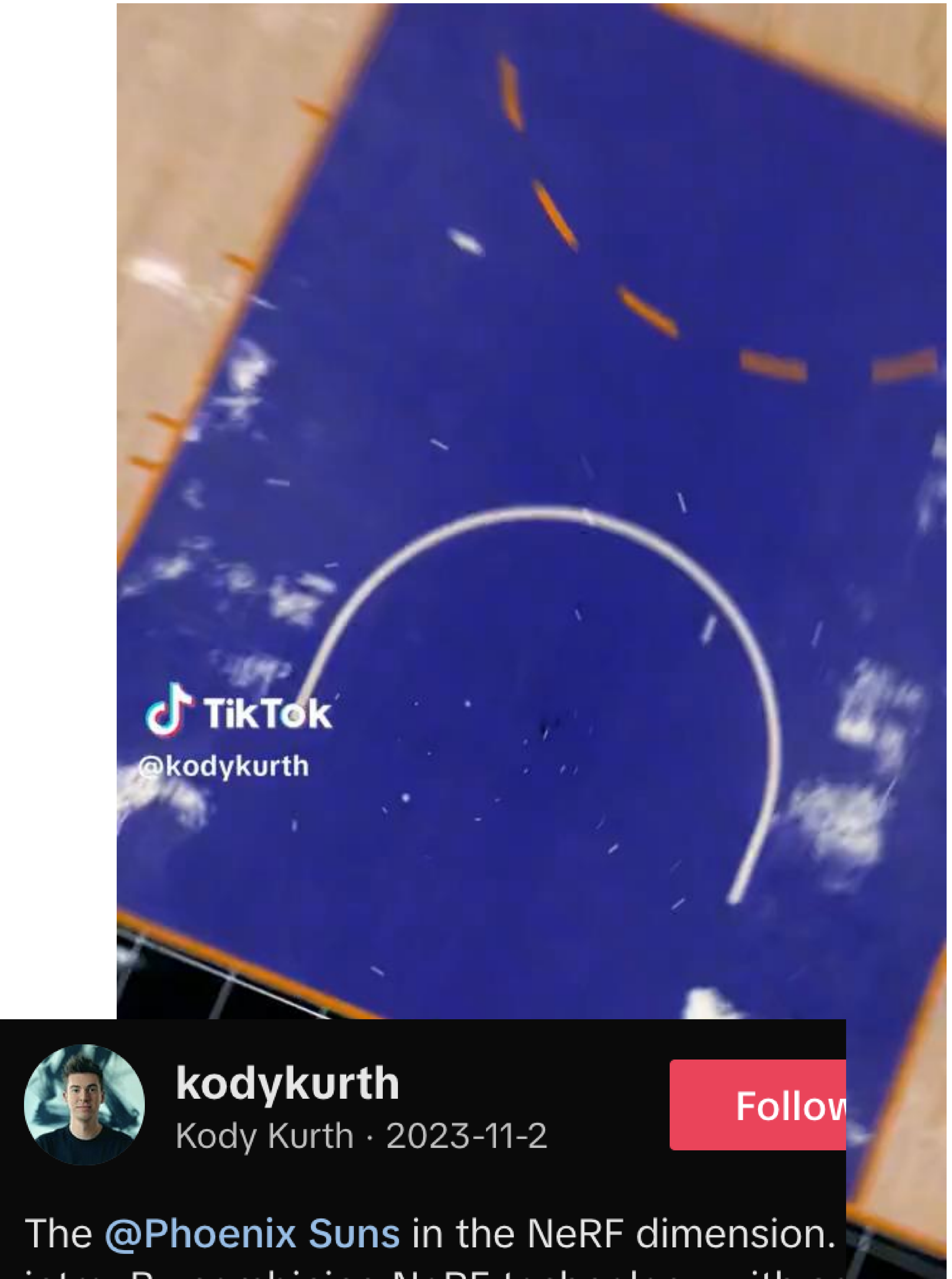
Gaussian Splats: NeRF without the Neural



In your browser! sparkjs.dev

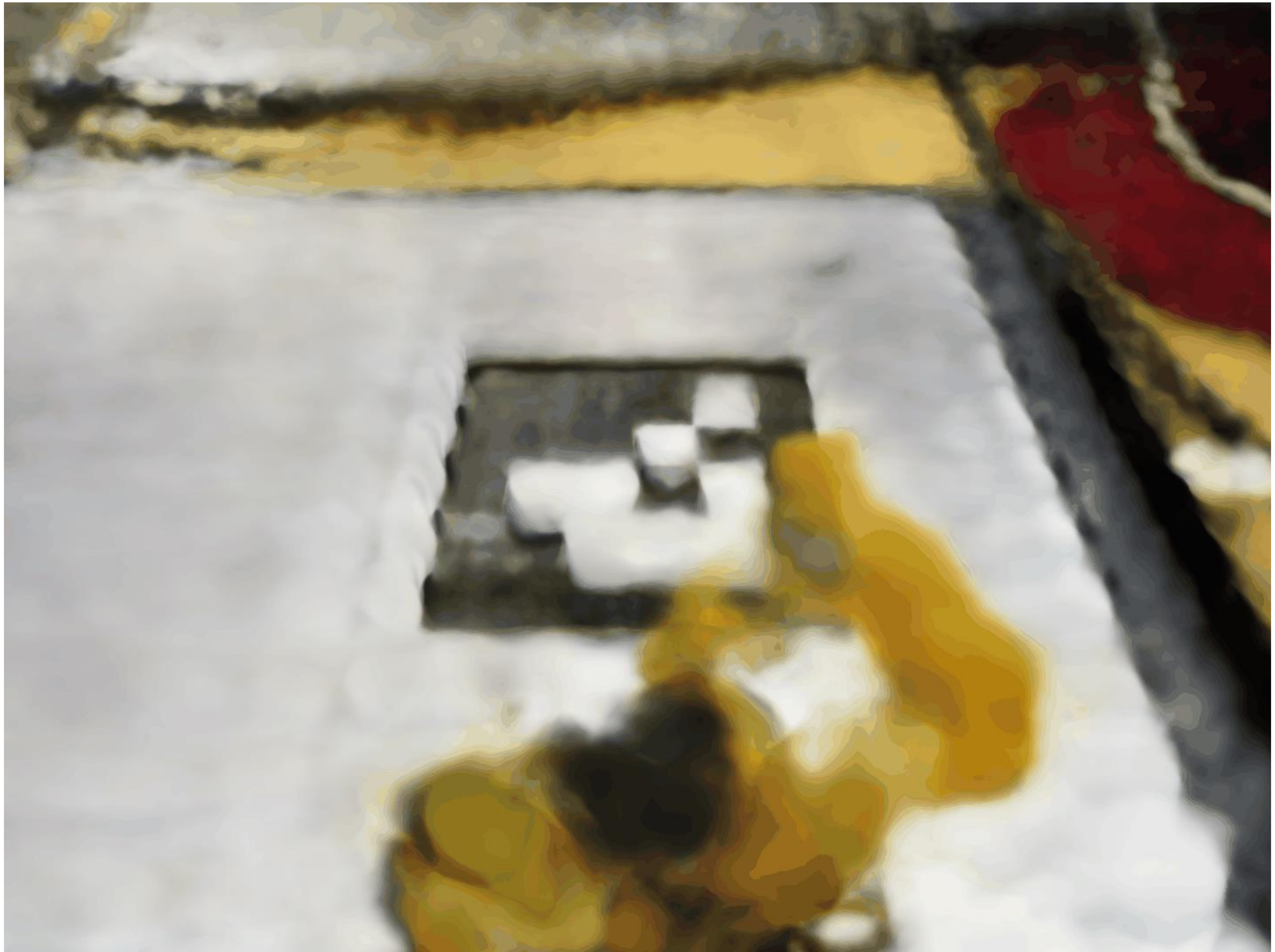


In industry! (VFX, maps, etc)

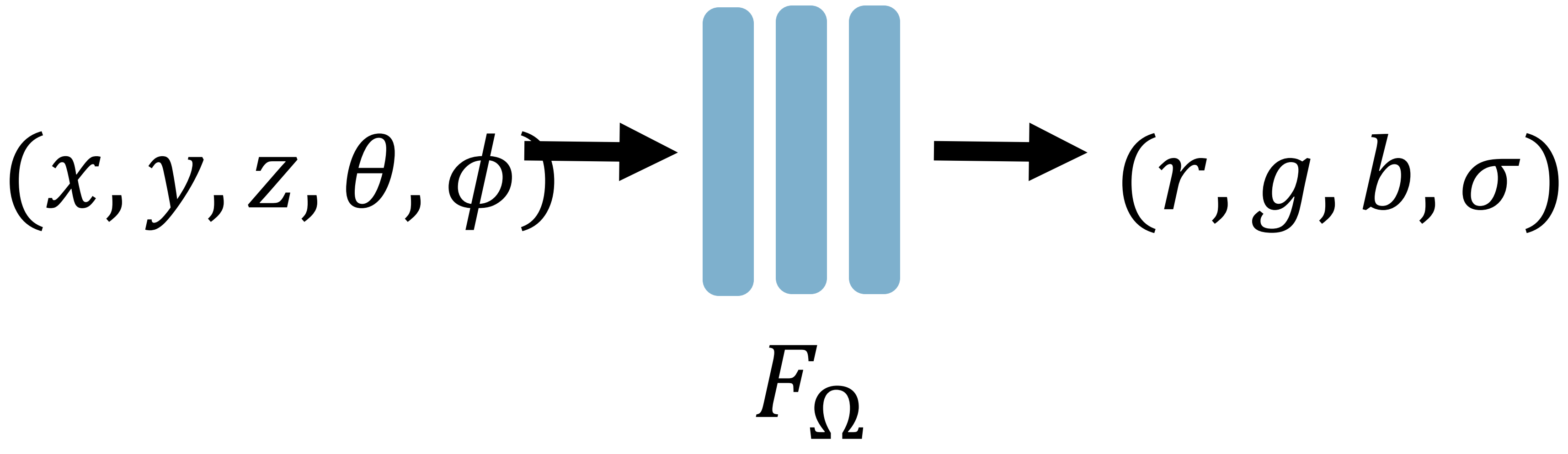


Google Maps

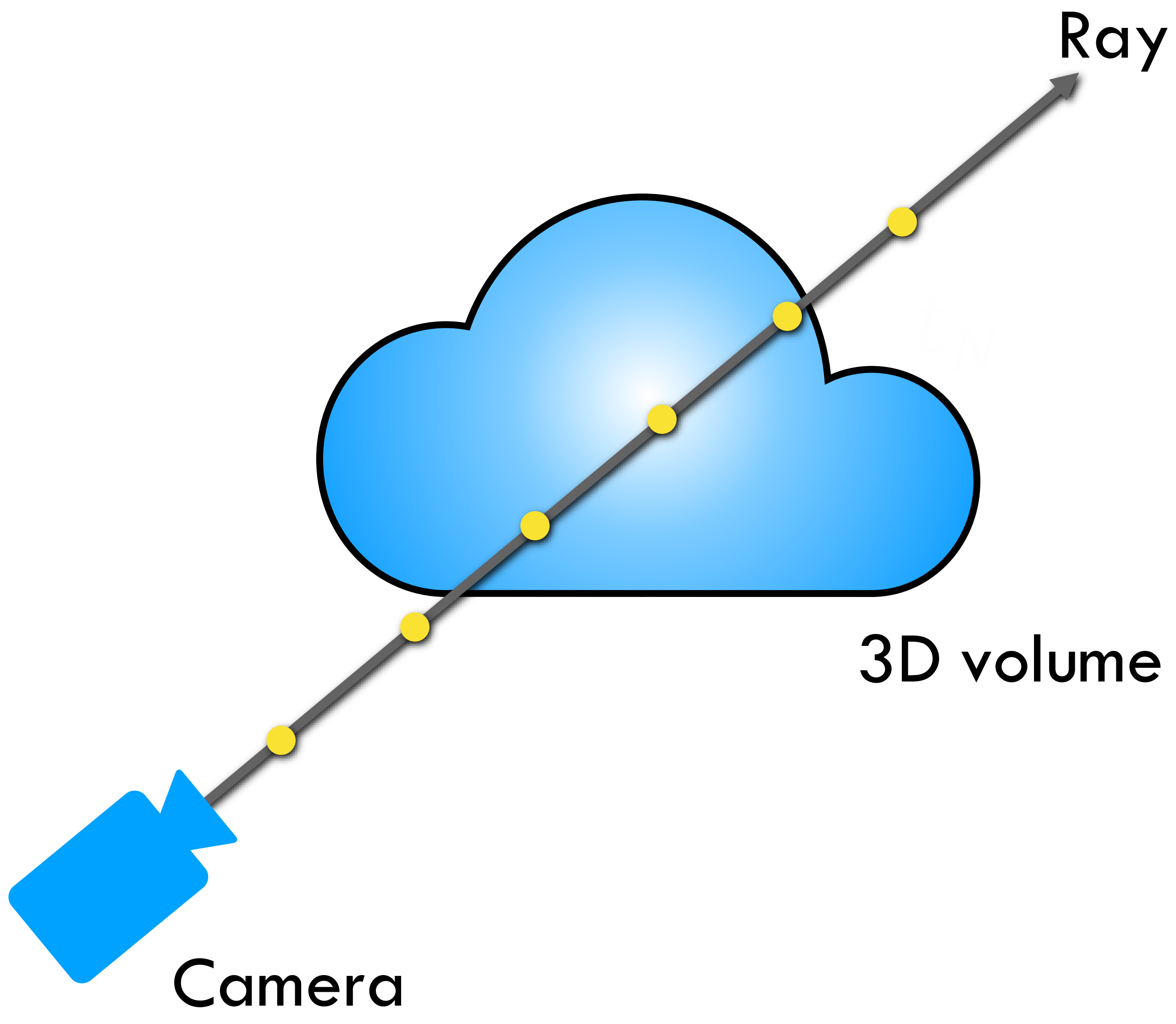
You will build a (simple) NeRF in proj4!



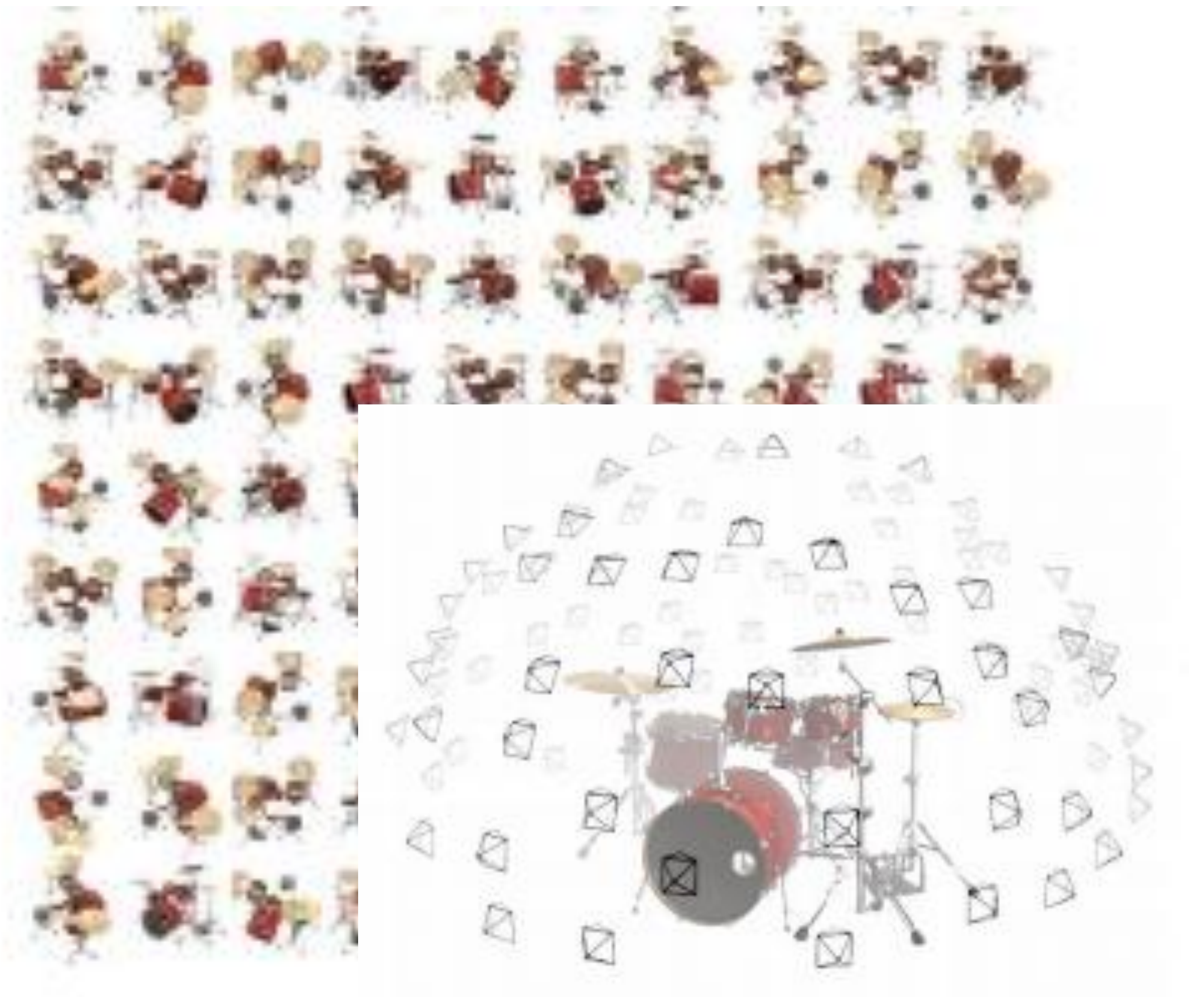
NeRF’s Three Key Components



Neural Volumetric 3D Scene Representation



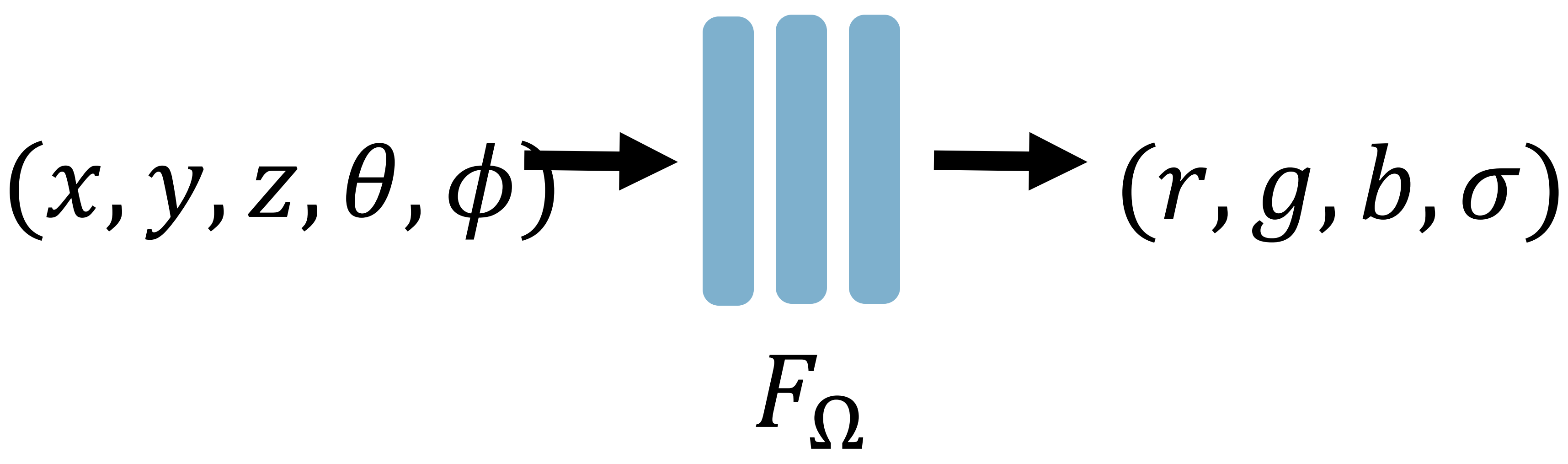
Differentiable Volumetric Rendering Function



Optimization via Analysis-by-Synthesis

NeRF's Three Key Components

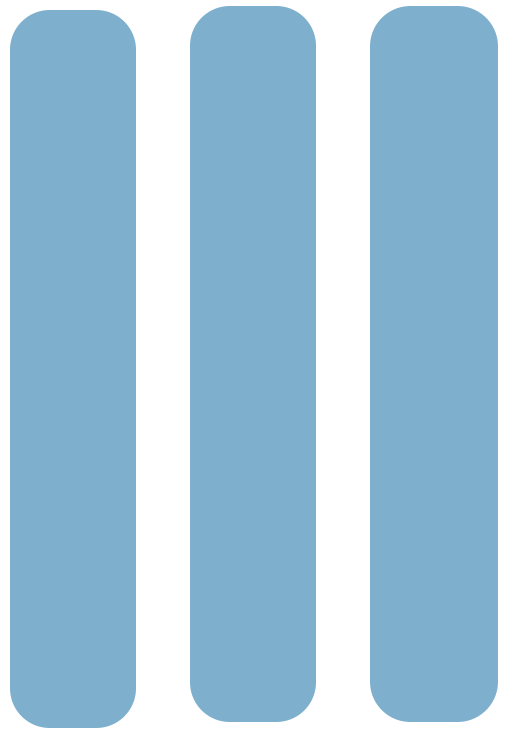
Today: focus on understanding this as an abstract function



Neural Volumetric 3D
Scene Representation

NeRF represents scenes as an *implicit* function

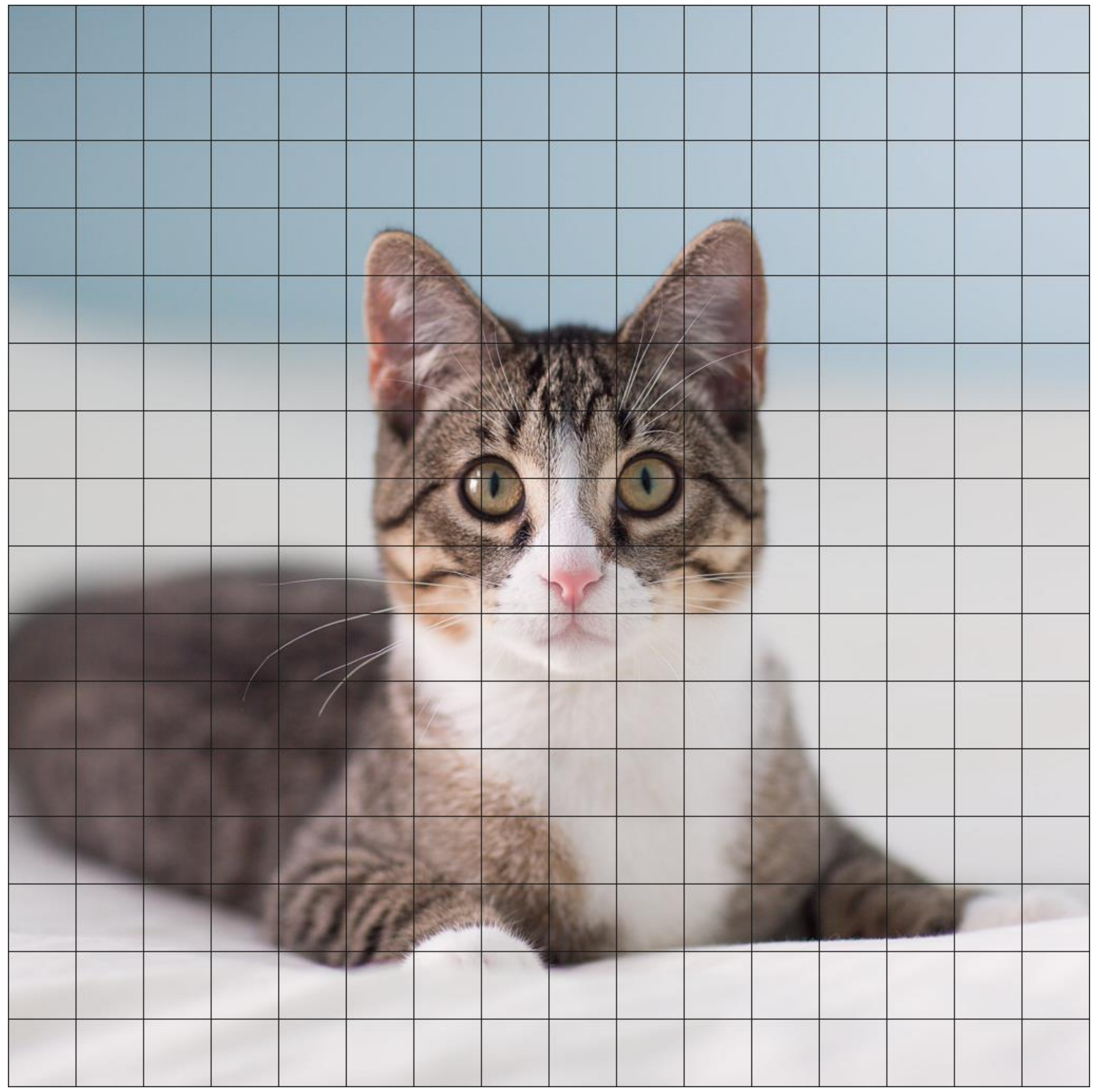
=



MLP

F_{Ω}

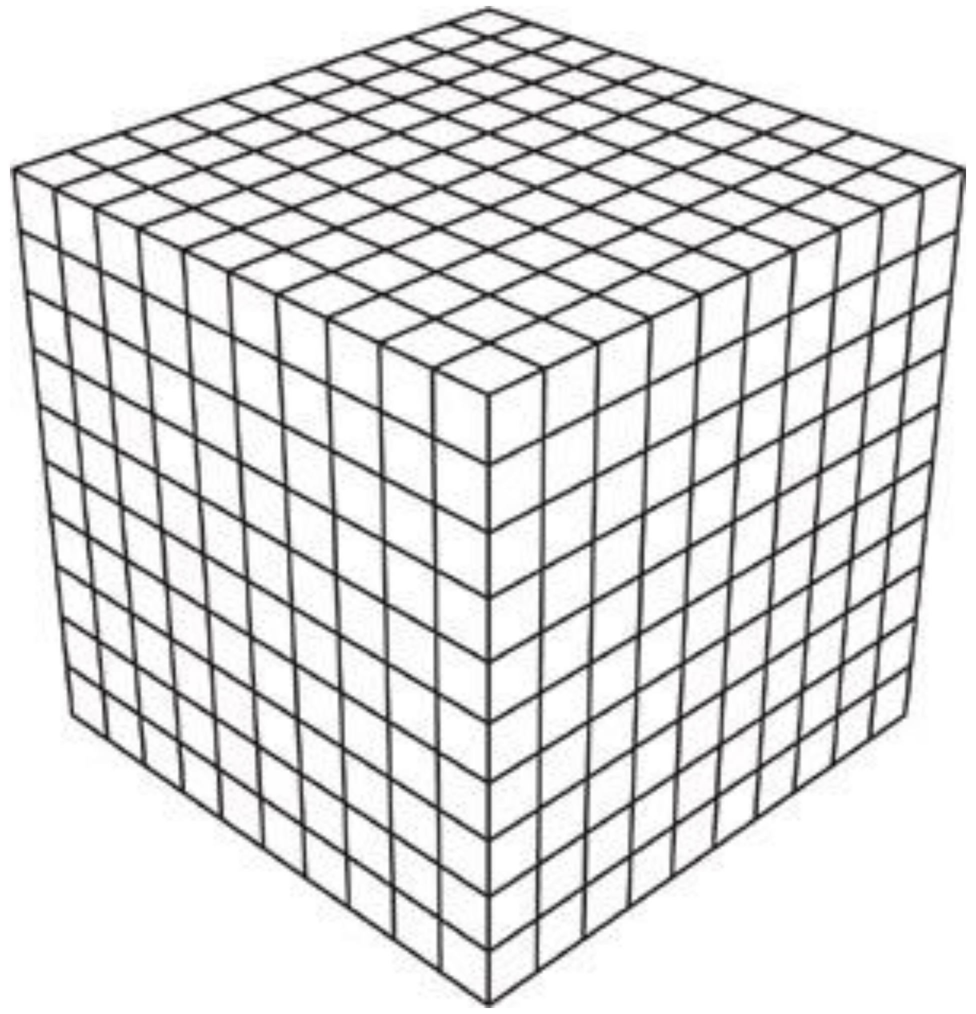
Life is nice in 2D-land…



Unified representation across applications, tools, and sensors

Slide Credit: Shubham Tulsiani

What's a 3D pixel? Voxels



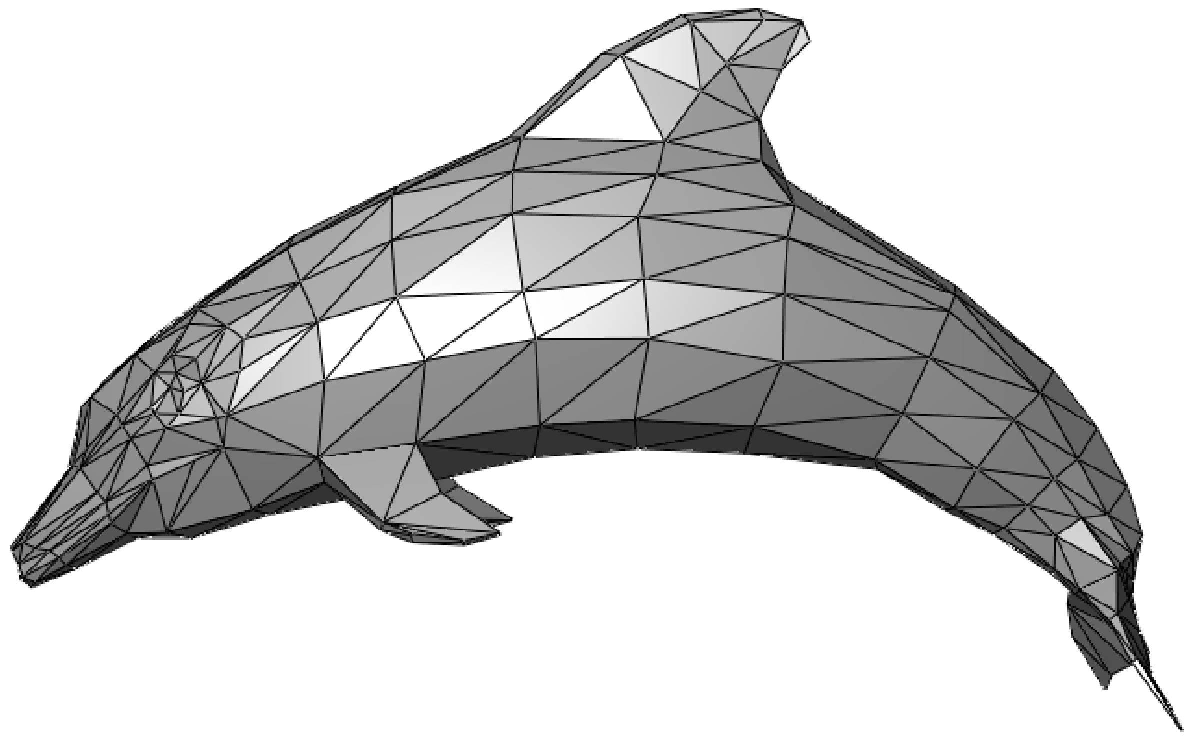
Discretize a 3-D space with some resolution: D x D x D

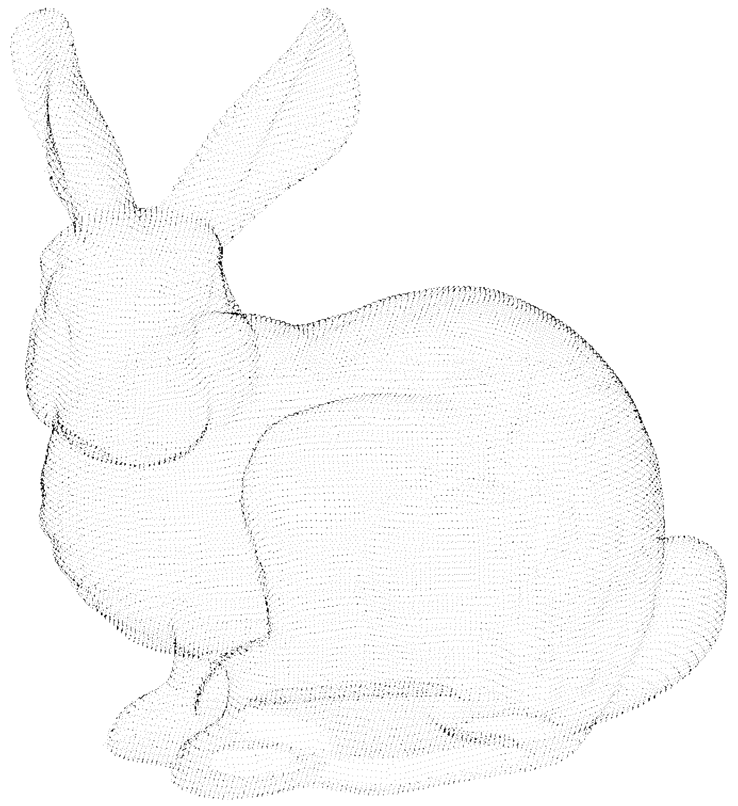
What is stored at each value can be:

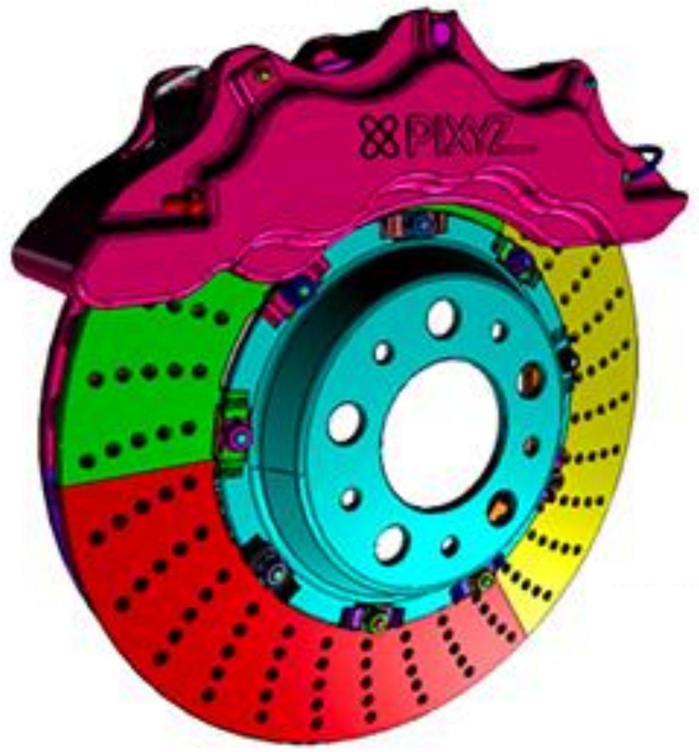
- 1 or 0 (occupancy)
- Signed distance
- Color/attributes

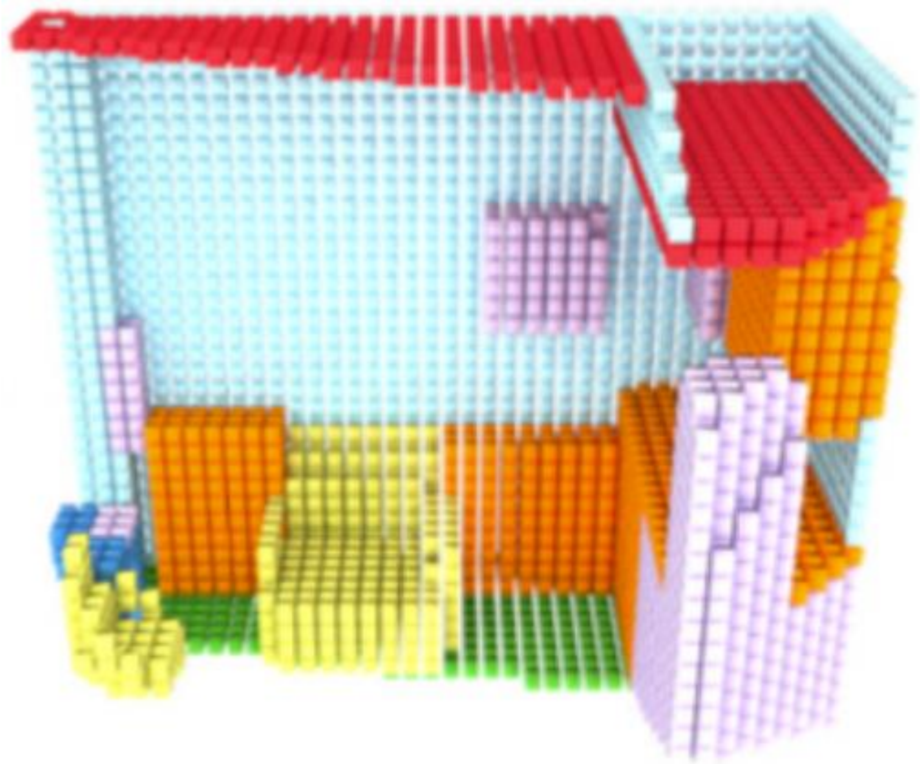
Extremely memory-intense!

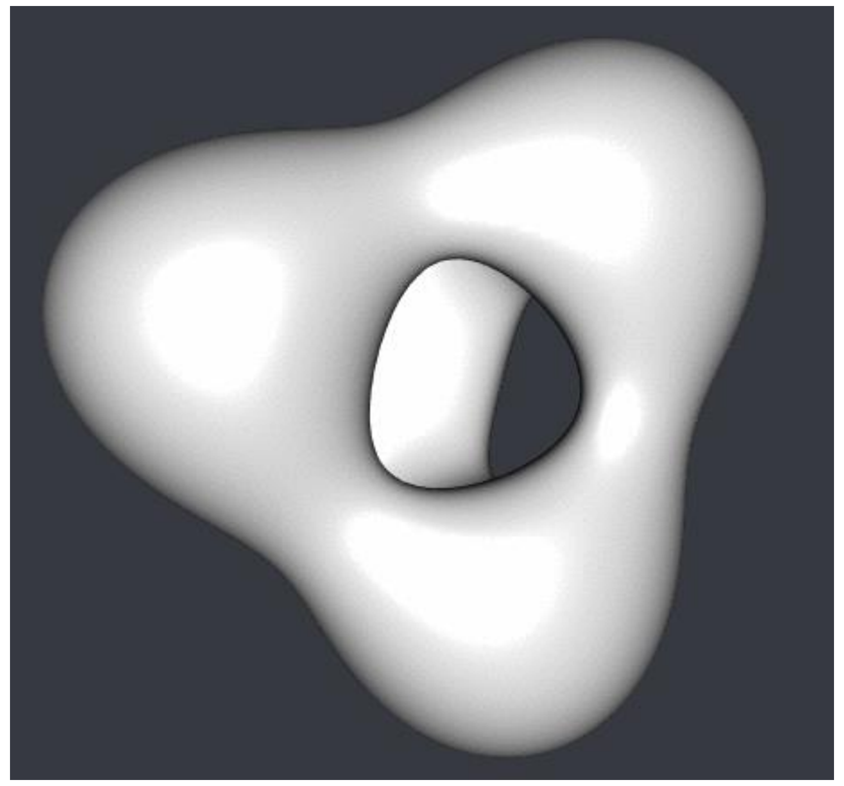
Representing 3D…



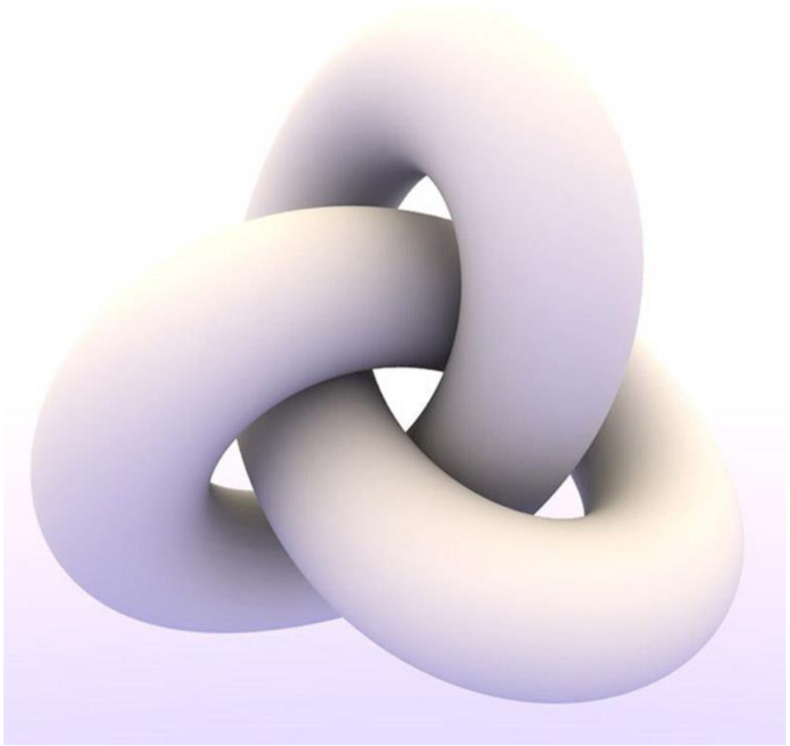




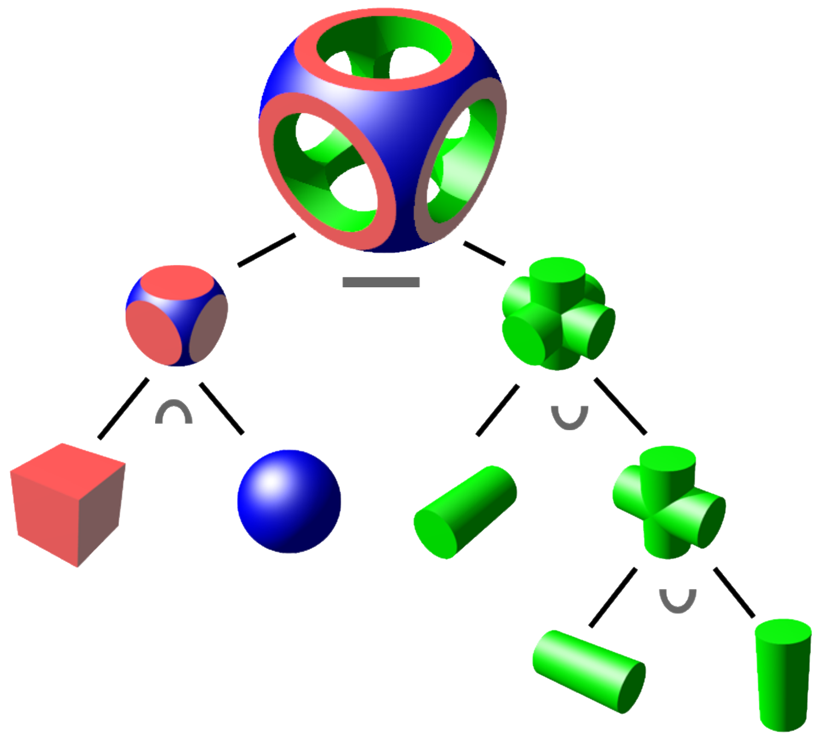




$\{\mathbf{p} \mid f(\mathbf{p}) = 0\}$



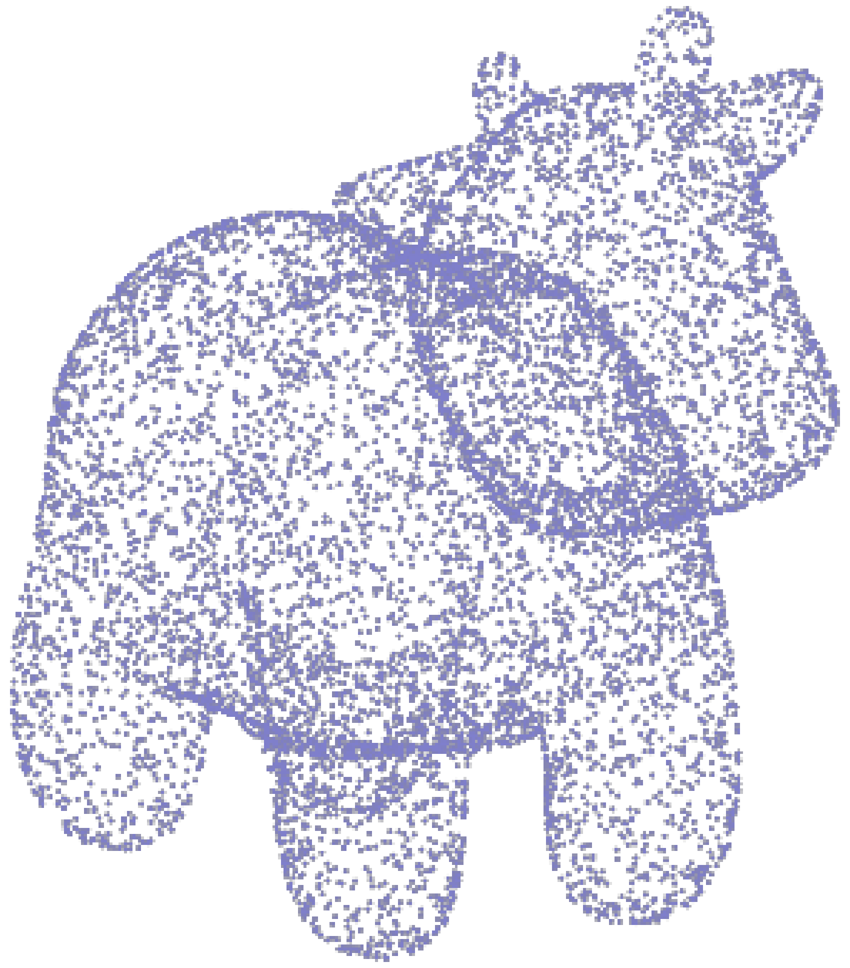
$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^3$

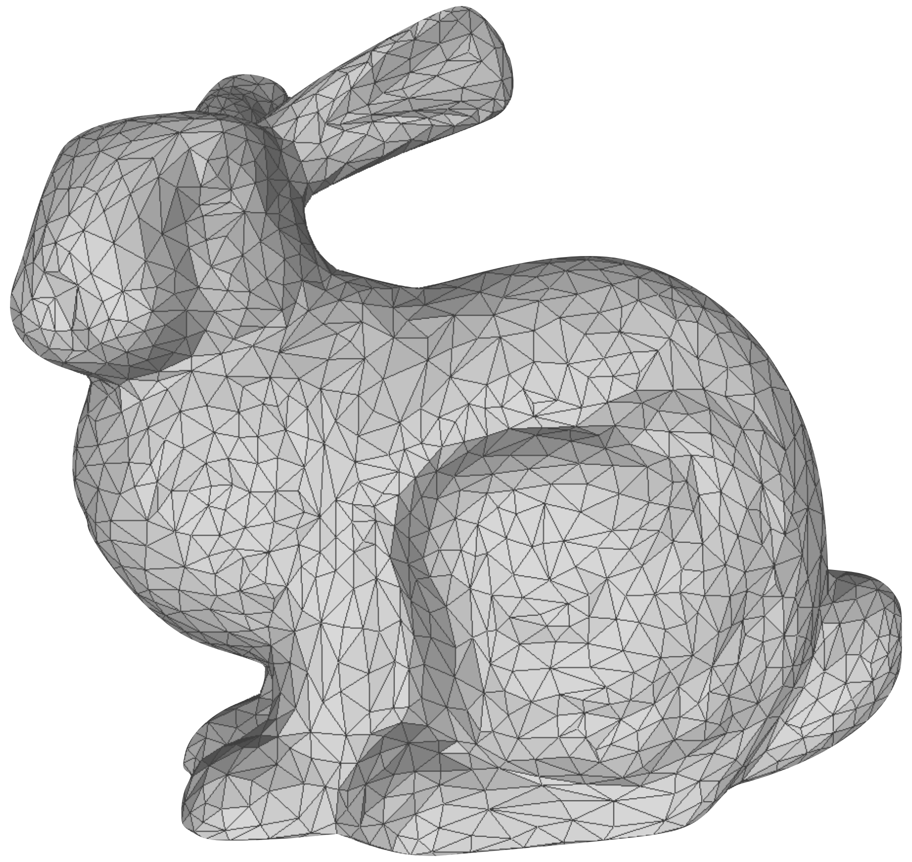


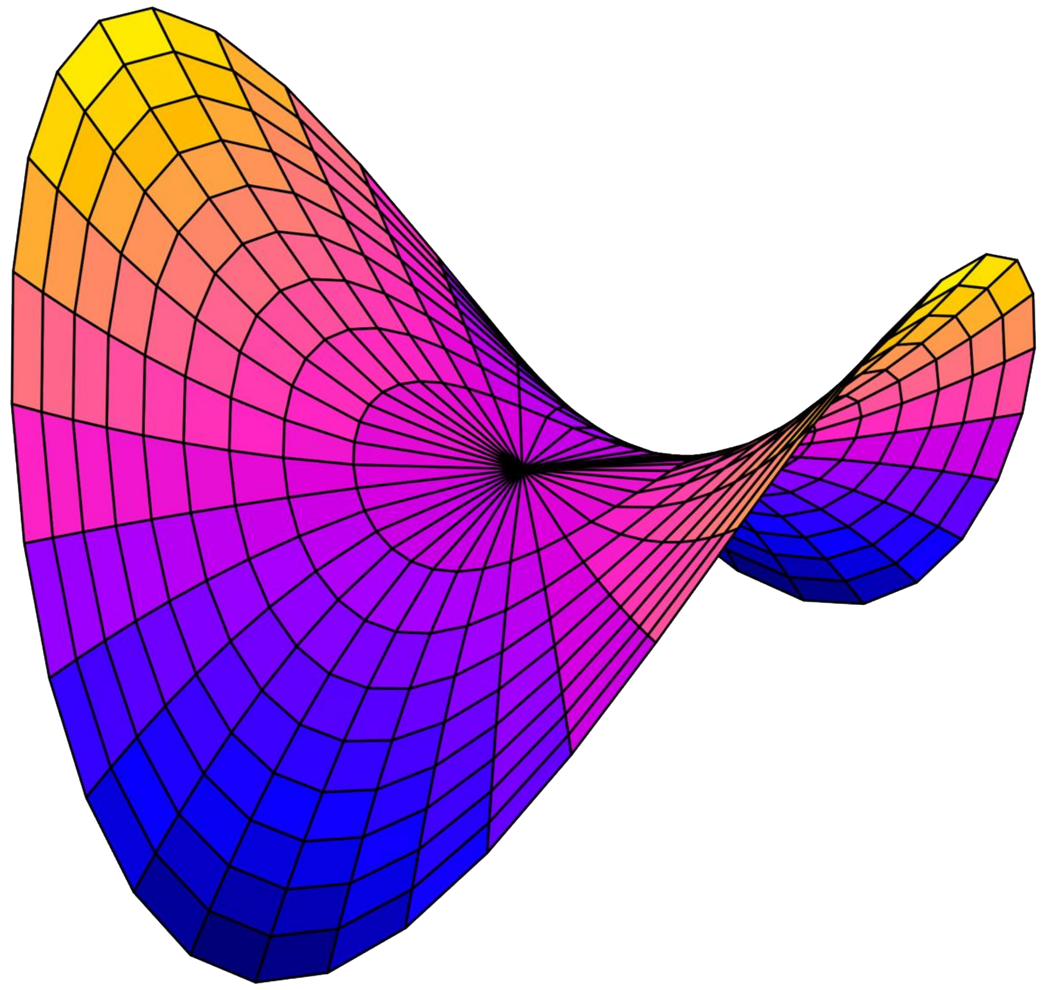
and many more …

Slide Credit: Shubham Tulsiani

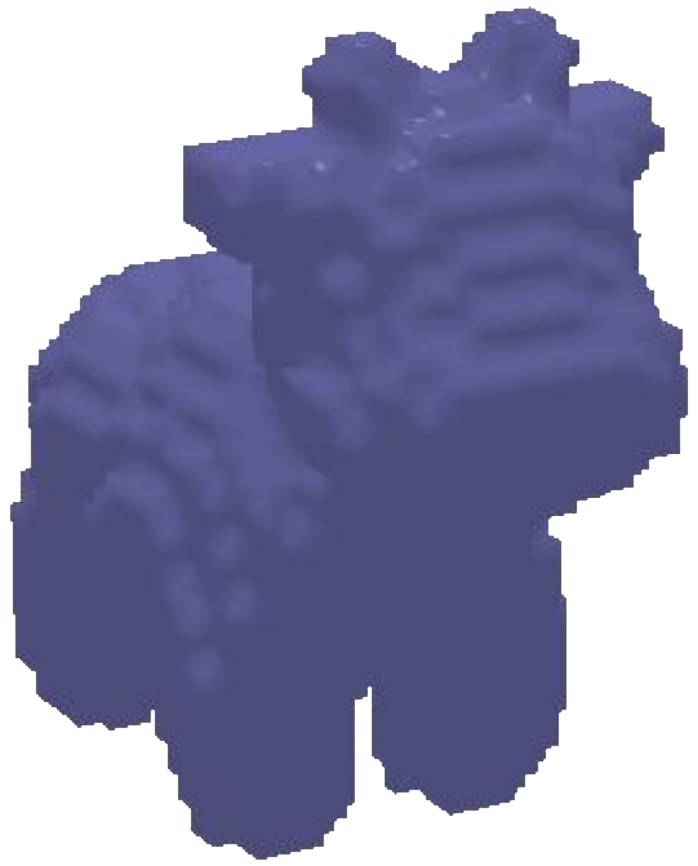
Two broad categories

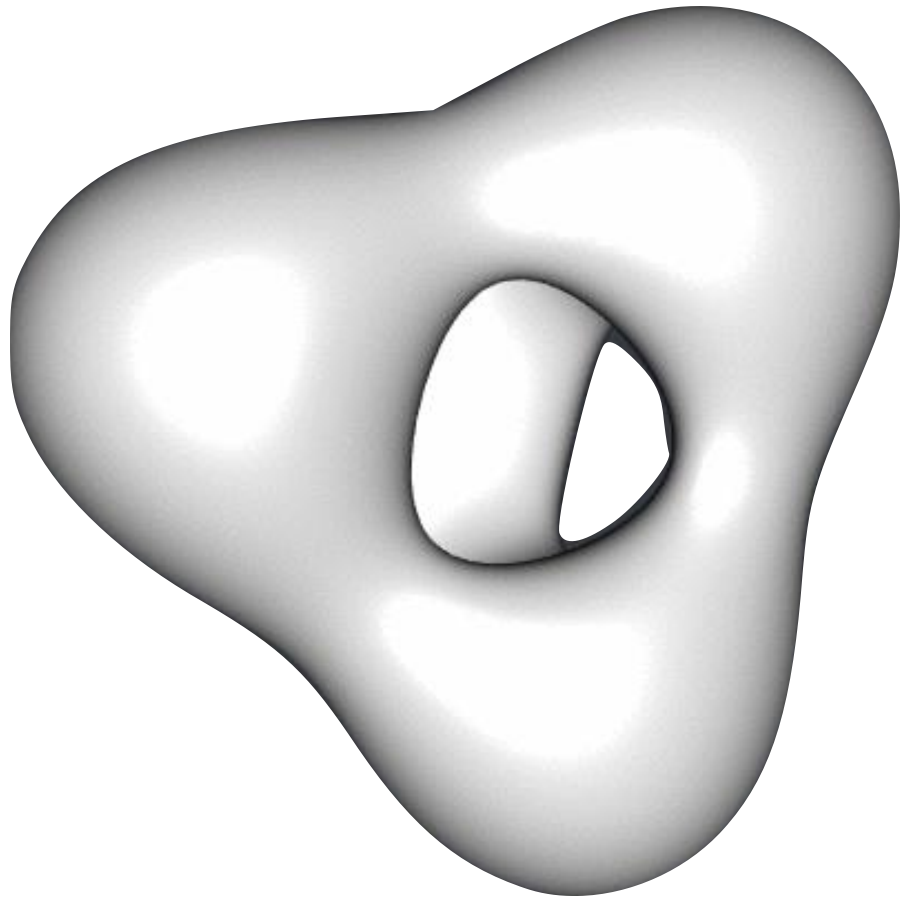






Surface Representations





Volume Representations

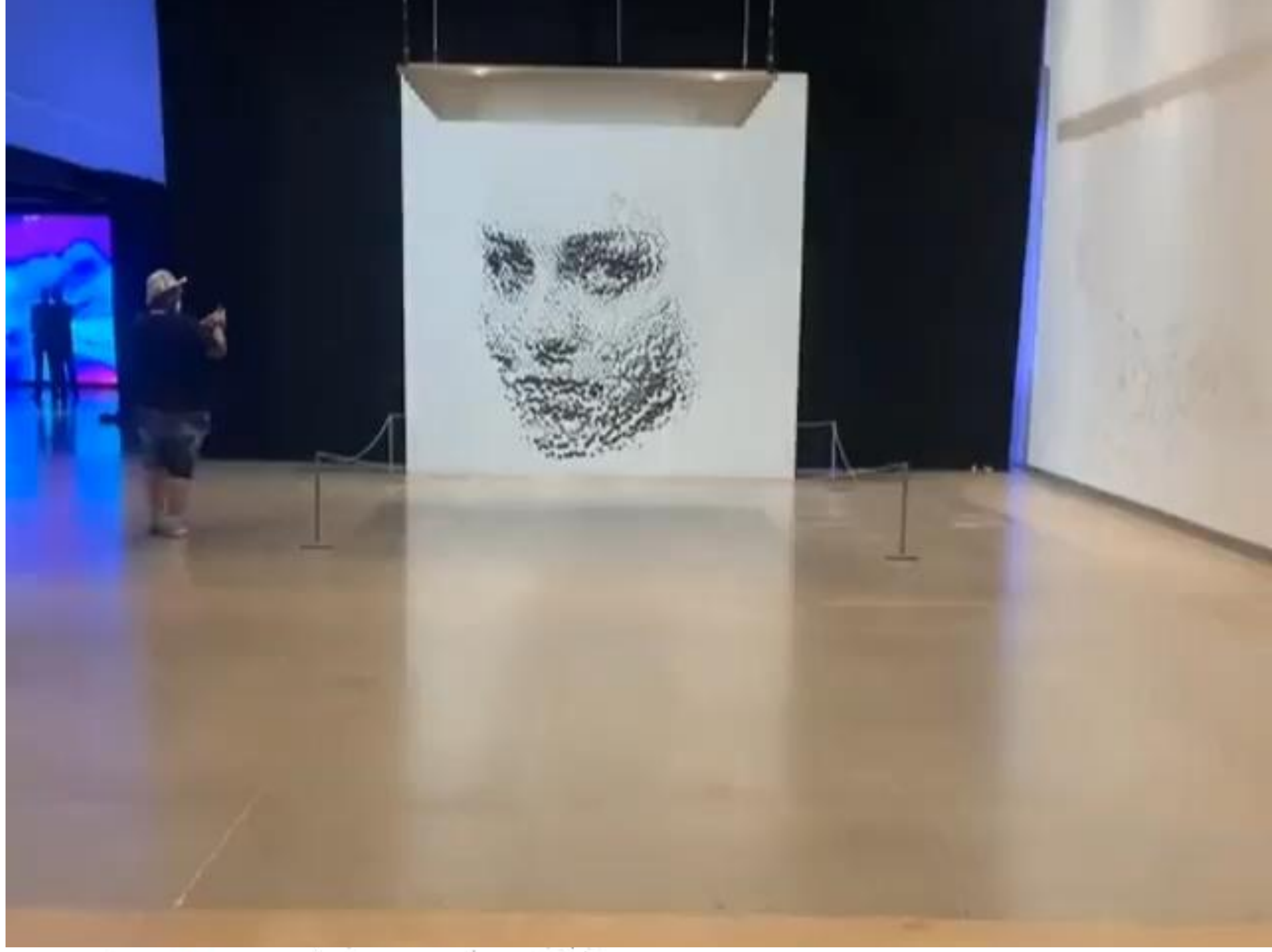
Slide Credit: Shubham Tulsian

Point clouds

A basic point cloud = {(x, y, z)_i}
With other attributes (color, normal)

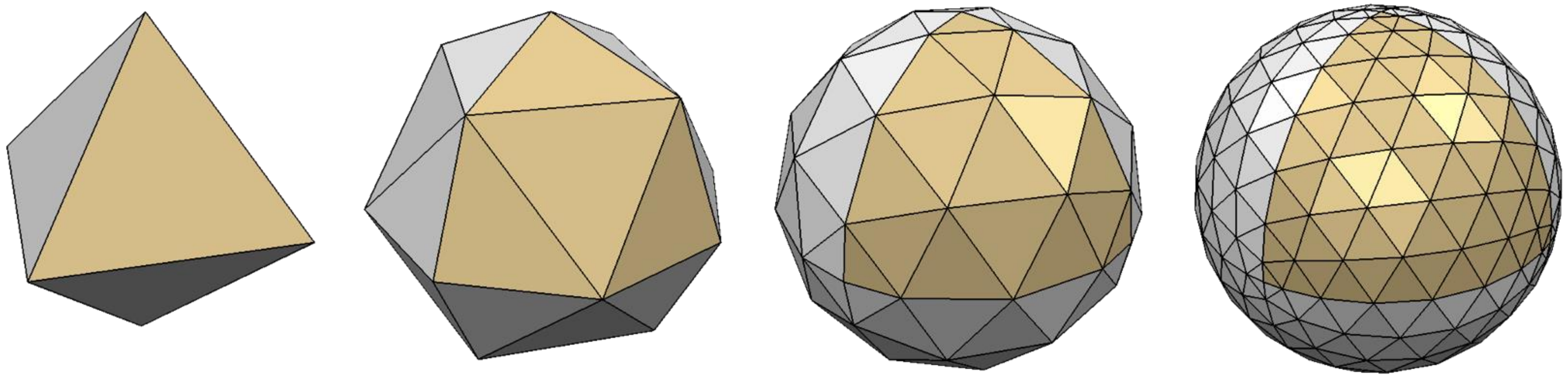
Obtained from

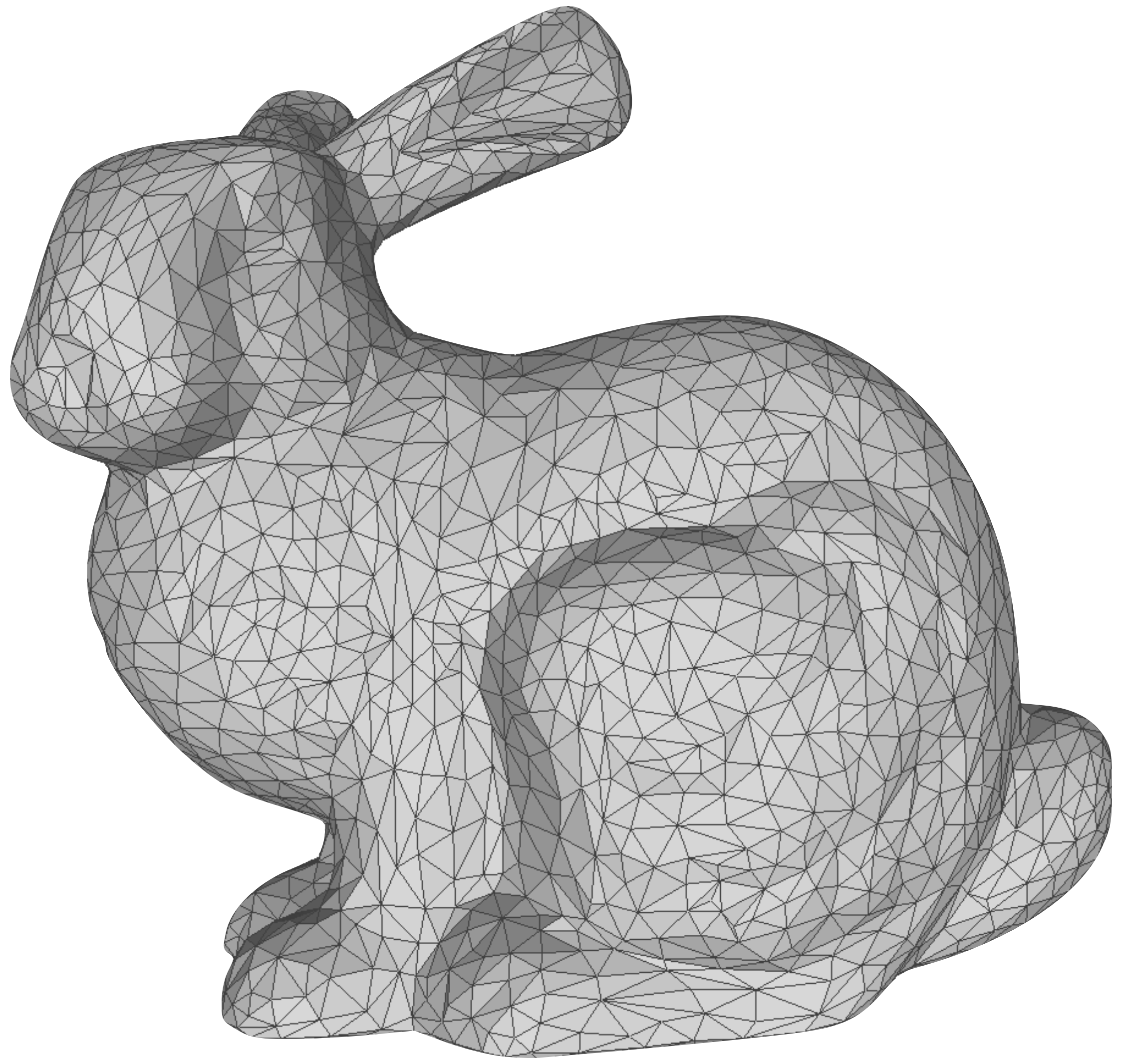
- **Depth images / Lidar**
- **SfM outputs**
- **Scanner outputs**



Polygonal Meshes

- A mesh is a set of vertices with faces that defines the topology
- Mesh = {Vertices, Faces}
  - Vertices: N x 3
  - Faces: F x {3, 4, …} specifying the edges of a polygon
  - Triangle faces most common but tetrahedrons (tets) are also.
- **Surface** is explicitly modeled by the faces
- Most common modeling representation

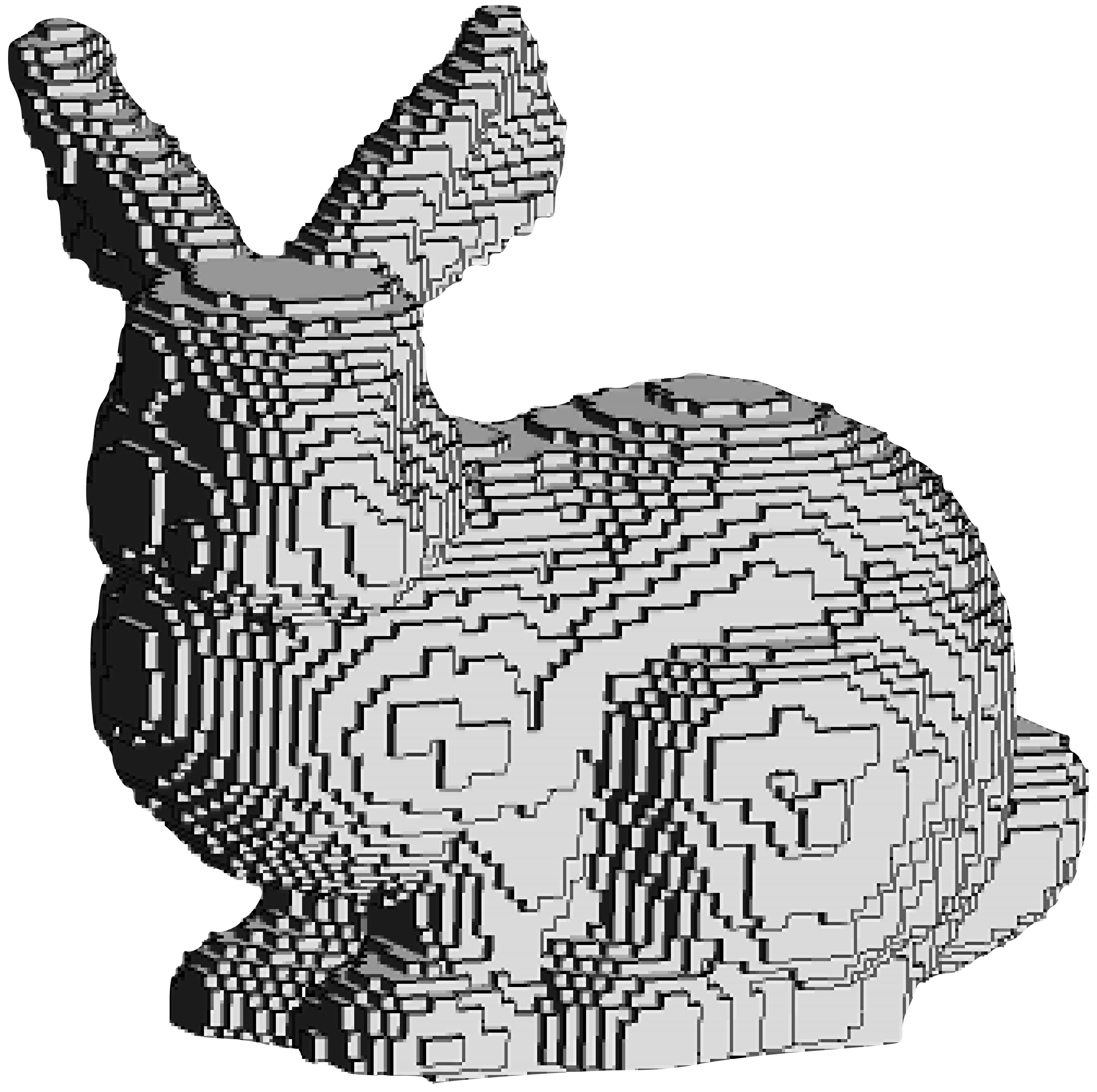




Volumetric representations

Model the *entire* space

Can be explicit (voxels) or implicit (NeRF)



NeRF: <u>implicit</u> volumetric representation

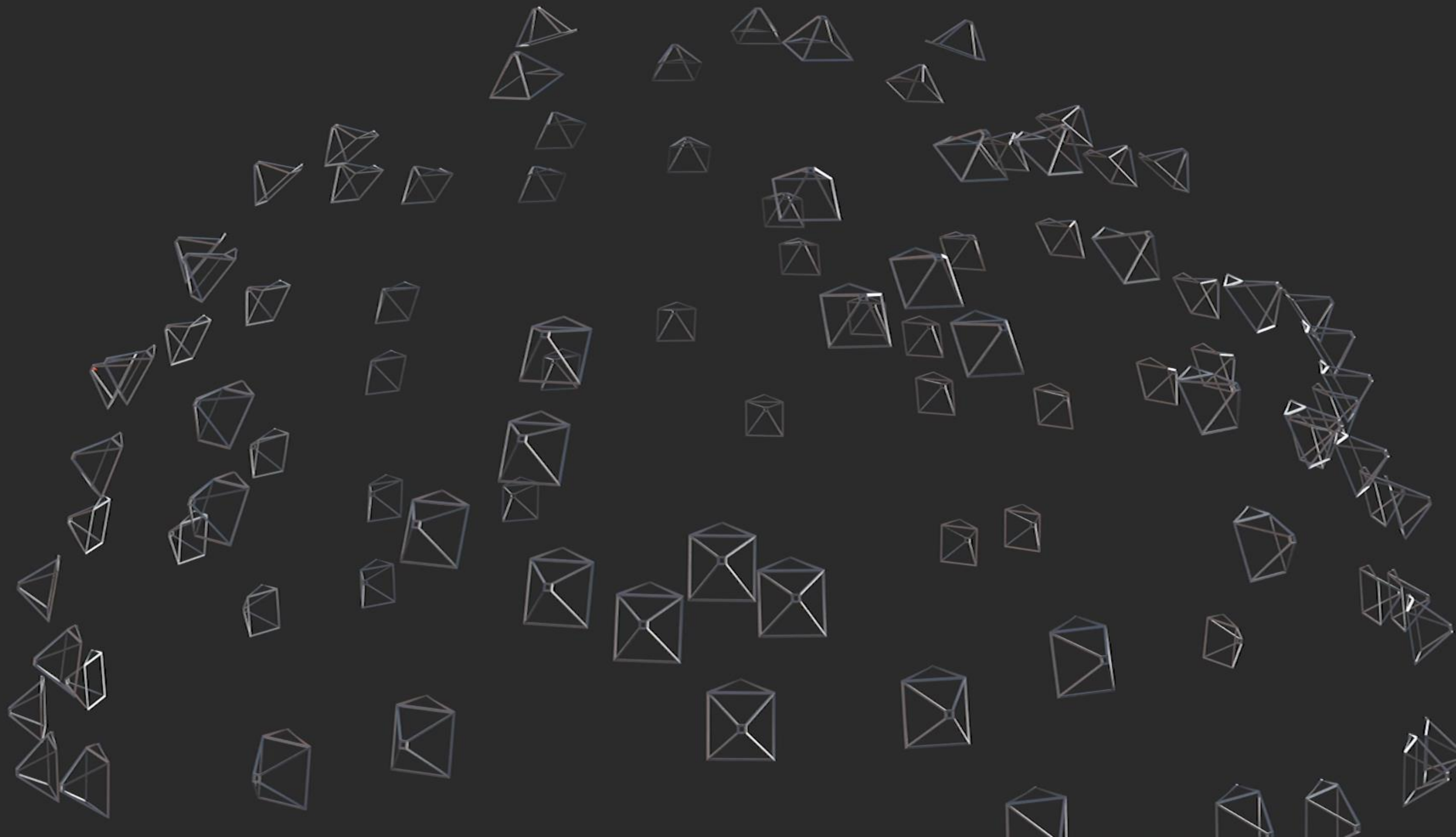
to find where “stuff” is, have to exhaustively query!

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

F_{Ω}

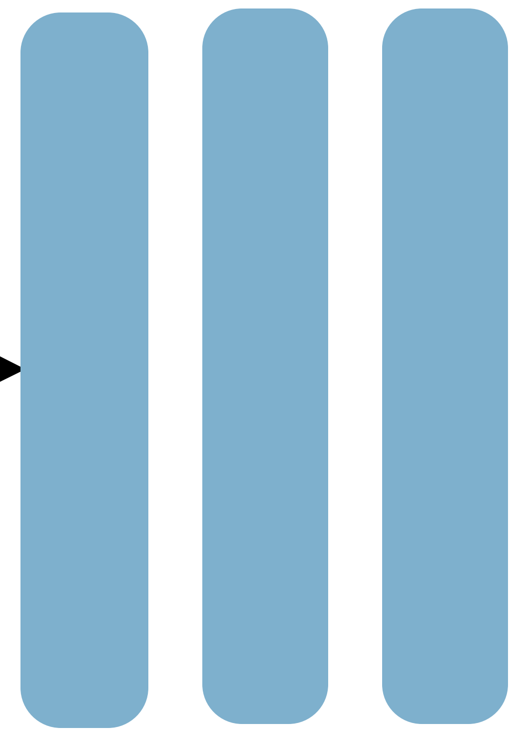
MLP

What does querying this look like?



What does this function mean???

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



It starts with The Plenoptic Function

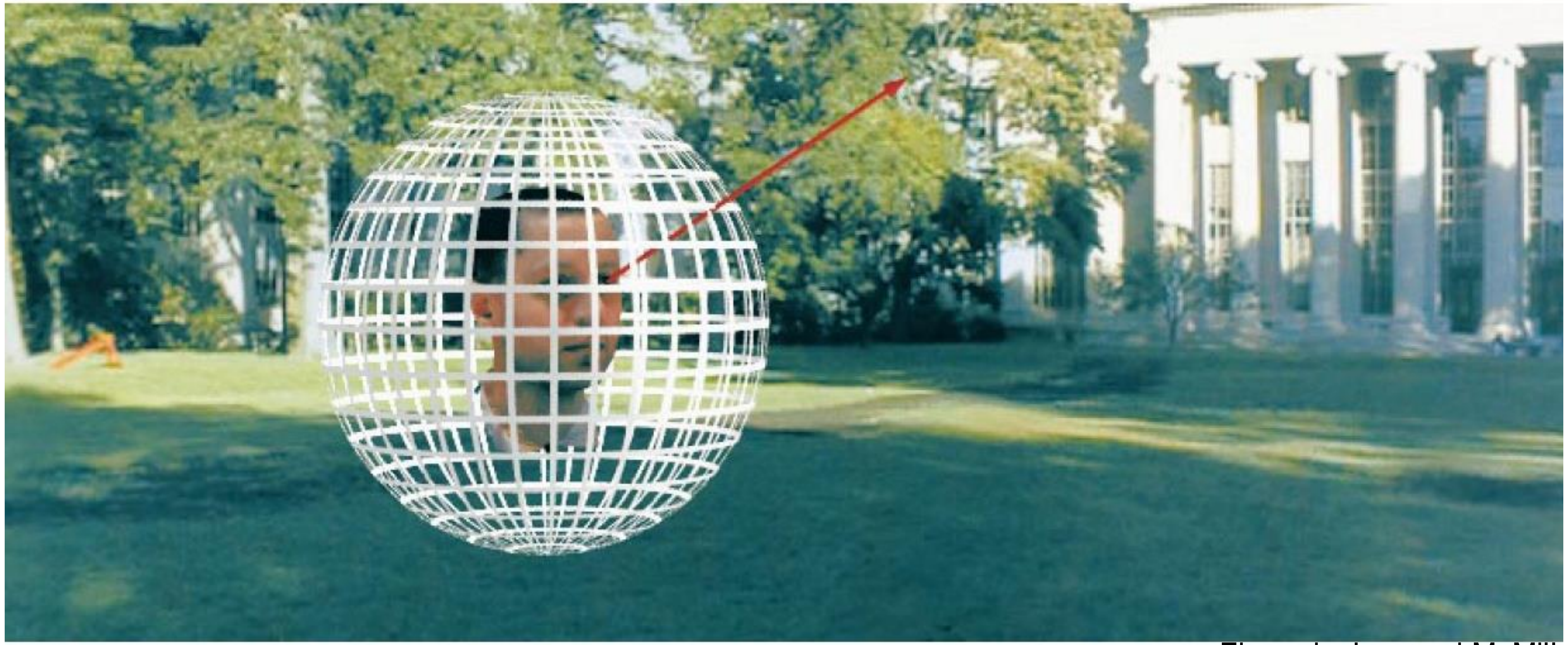


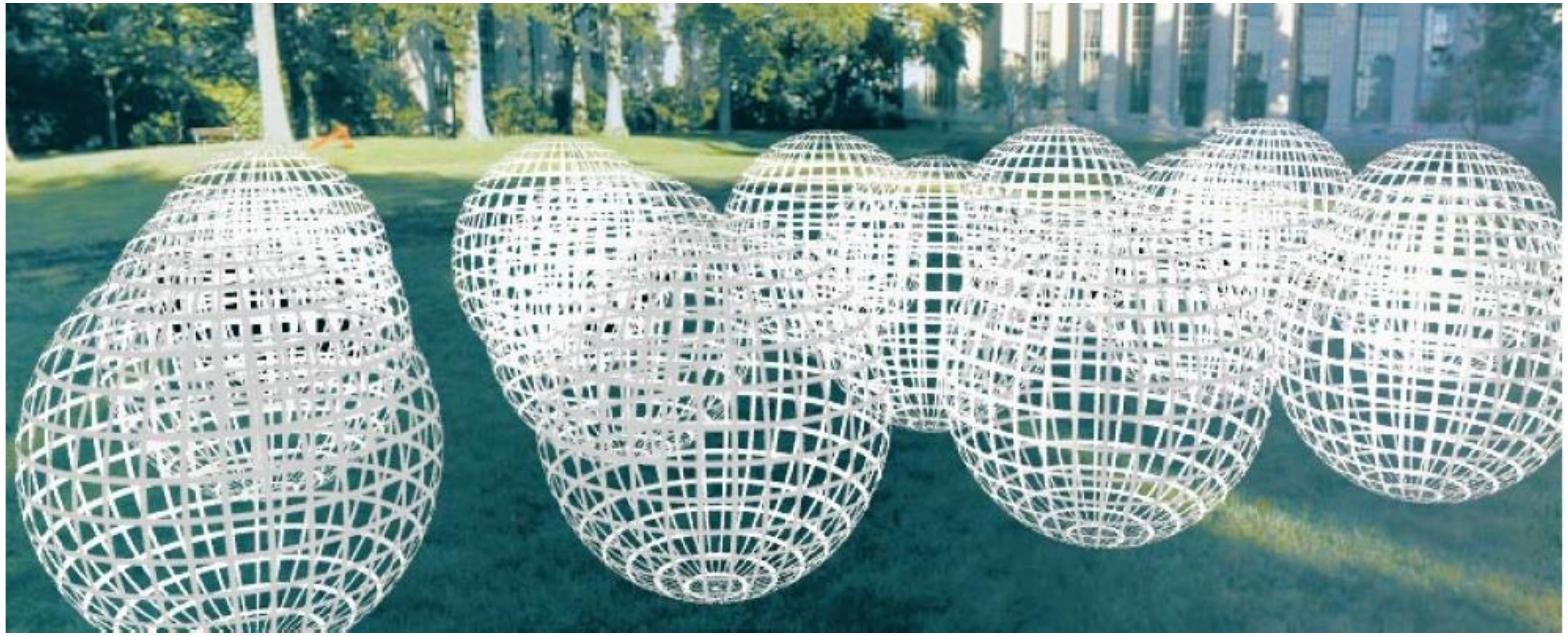
Figure by Leonard McMillan

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

A: The Plenoptic Function (Adelson & Bergen '91)

Slide credit:
Alyosha Efros

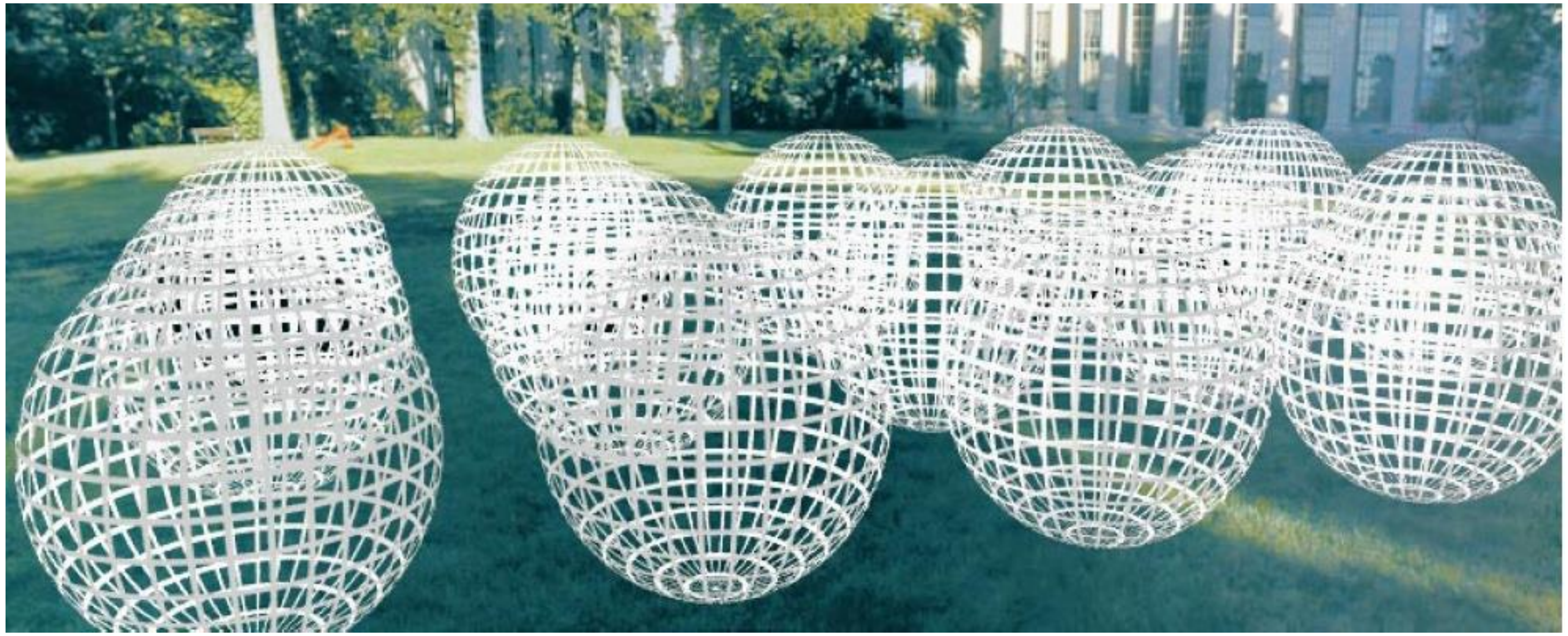
A holographic movie



$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

- is intensity of light
  - Seen from ANY position and direction
  - Over time
  - As a function of wavelength

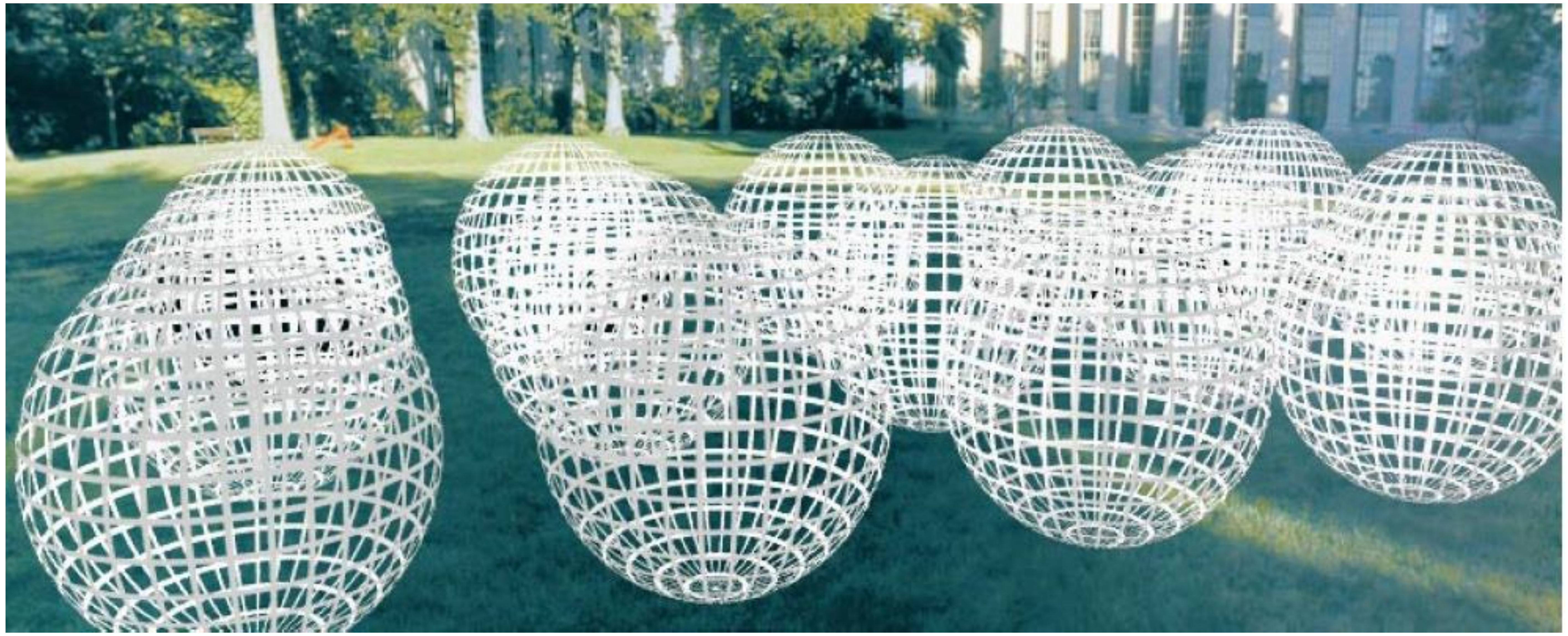
The plenoptic function



$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

7D function, that can reconstruct every position & direction,
at every moment, at every wavelength
= it recreates the entirety of our visual reality!

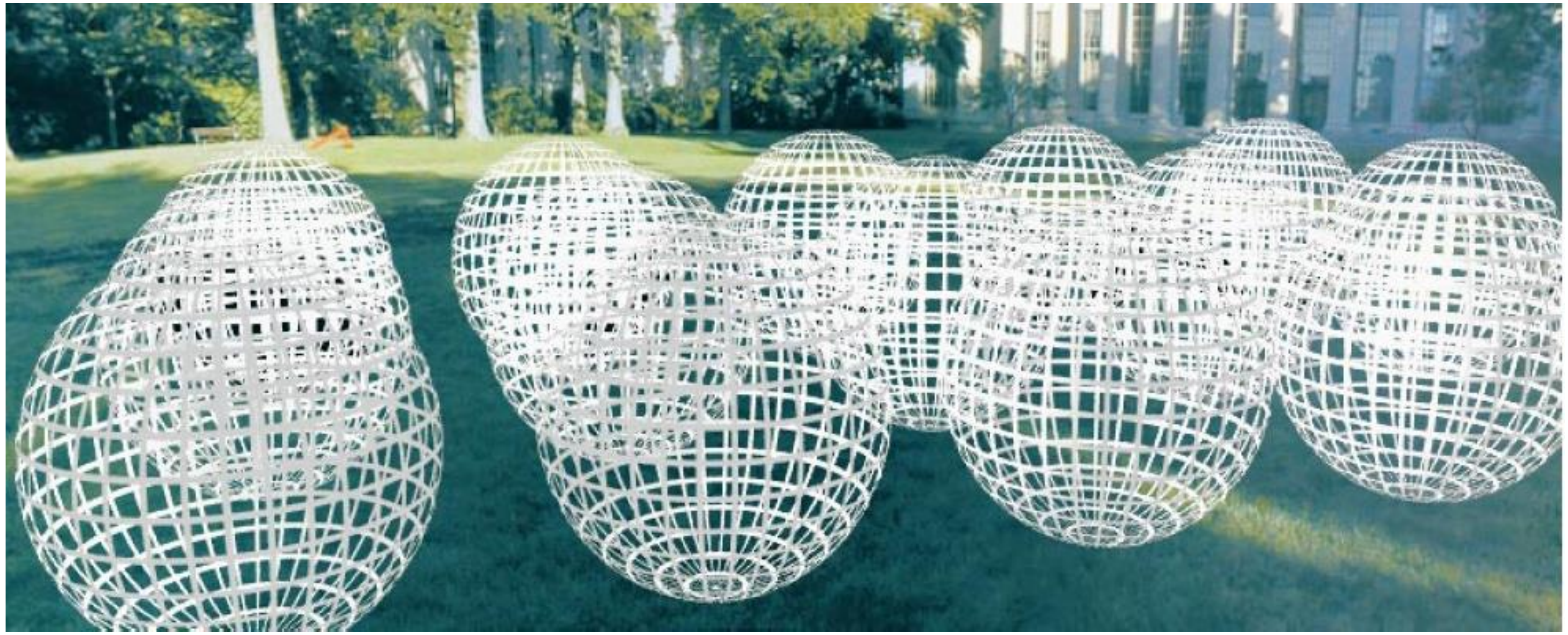
Goal: Plenoptic Function from a set of images



- Objective: Recreate the visual reality
- All about recovering photorealistic pixels, not about recording 3D point or surfaces
  —Image Based Rendering

aka **Novel View Synthesis**

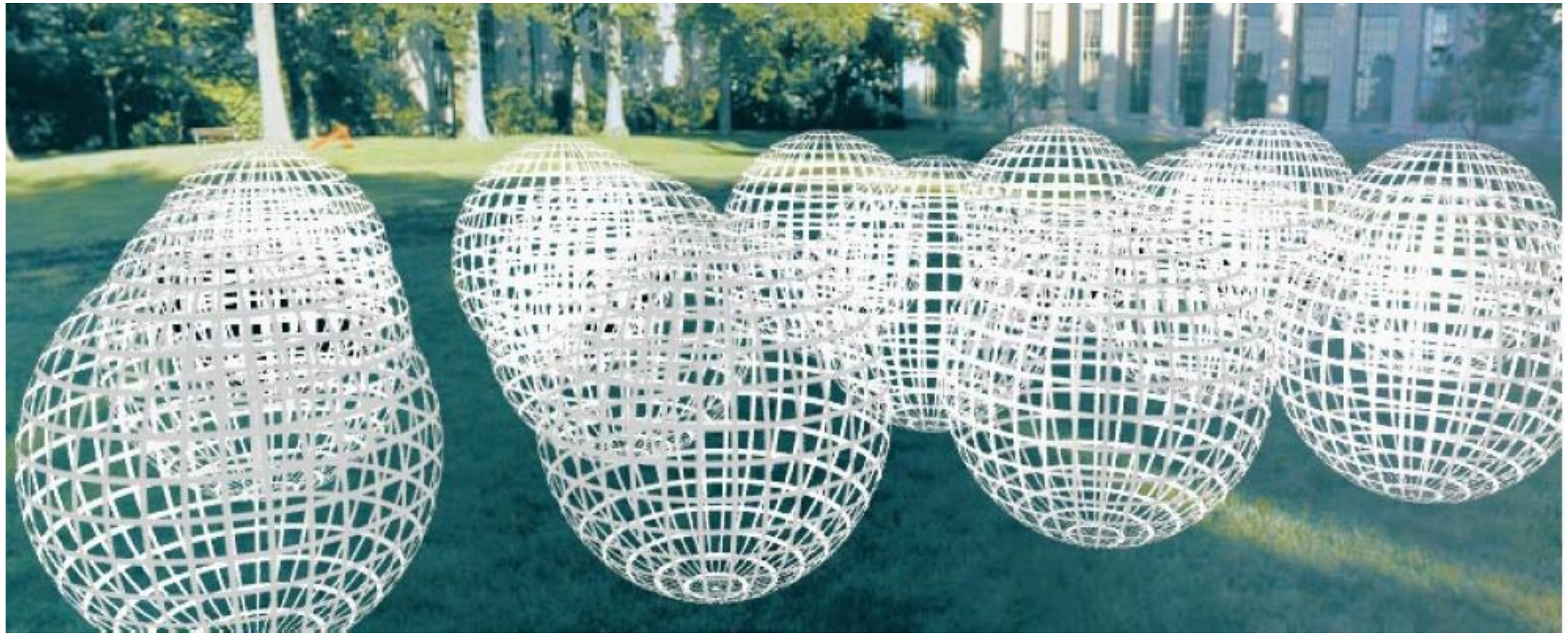
Goal: Plenoptic Function from a set of images



It is a conceptual device

Adelson & Bergen do not discuss how to solve this

Plenoptic Function



7D function:
2 – direction
1 – wavelength
1 – time
3 – location

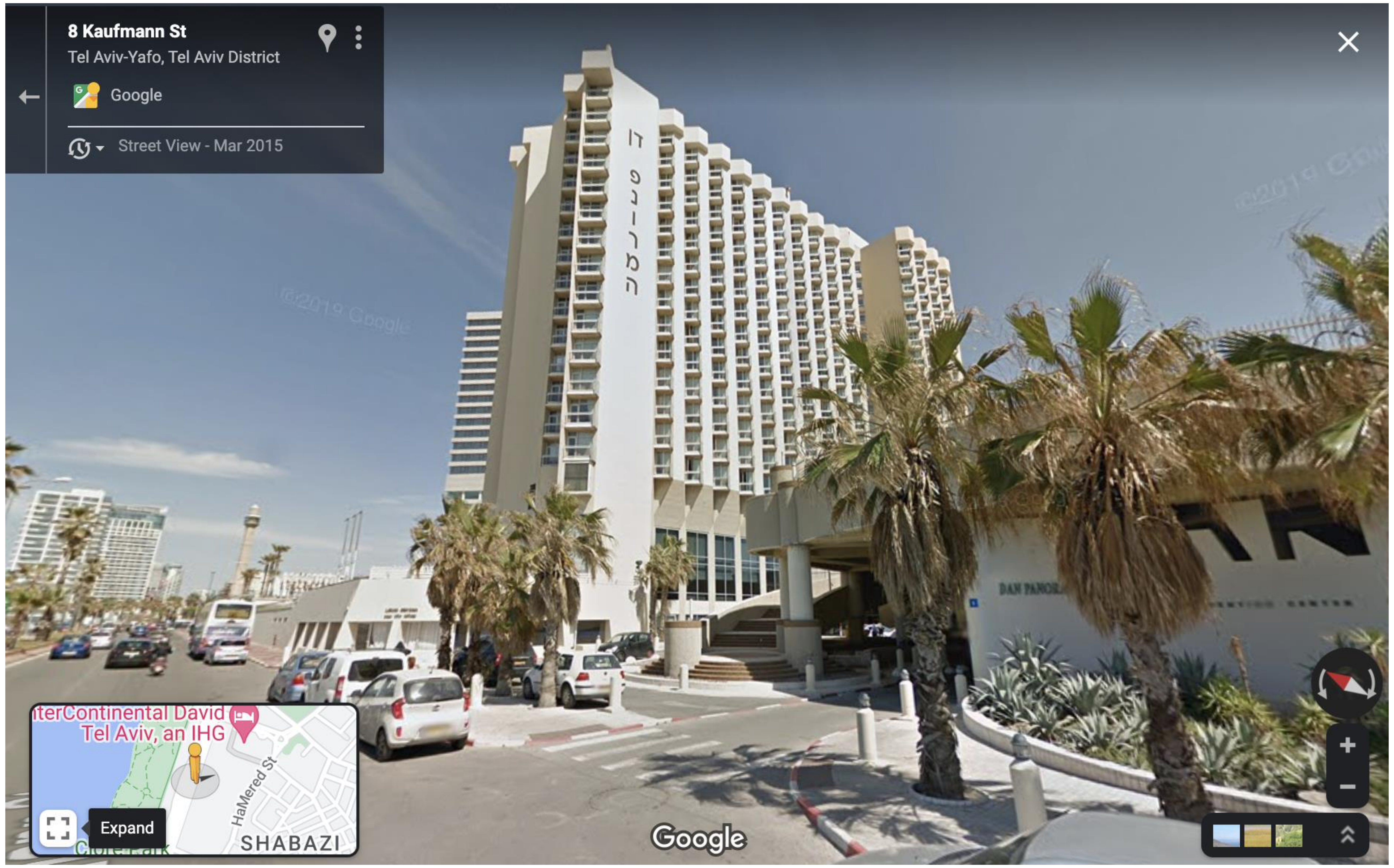
$$P(\theta,\phi,\cancel{\lambda},\cancel{t},V_X,V_Y,V_Z) \longrightarrow P(\theta,\phi,V_X,V_Y,V_Z)$$

Look familiar 🙂?

Let's simplify:

1. Remove the time
2. Remove the wavelength & let the function output RGB colors

An example of a sparse plenoptic function



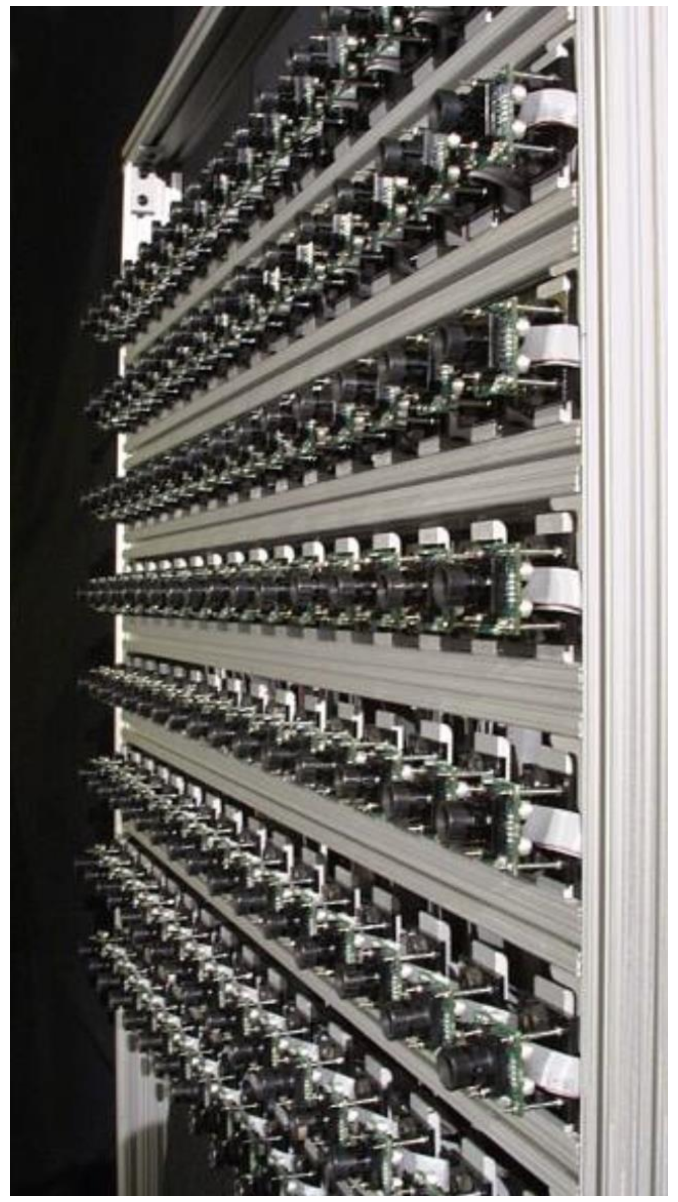
If street view was super dense
(360 view from any view point)
then it is the Plenoptic Function

Levoy and Hanrahan, SIGGRAPH 1996
Gortler et al. SIGGRAPH 1996

Lightfield / Lumigraph

- Previous approaches for modeling the Plenoptic Function
- Take a lot of pictures from many views
- Interpolate the rays to render a novel view

Stanford Gantry
128 cameras

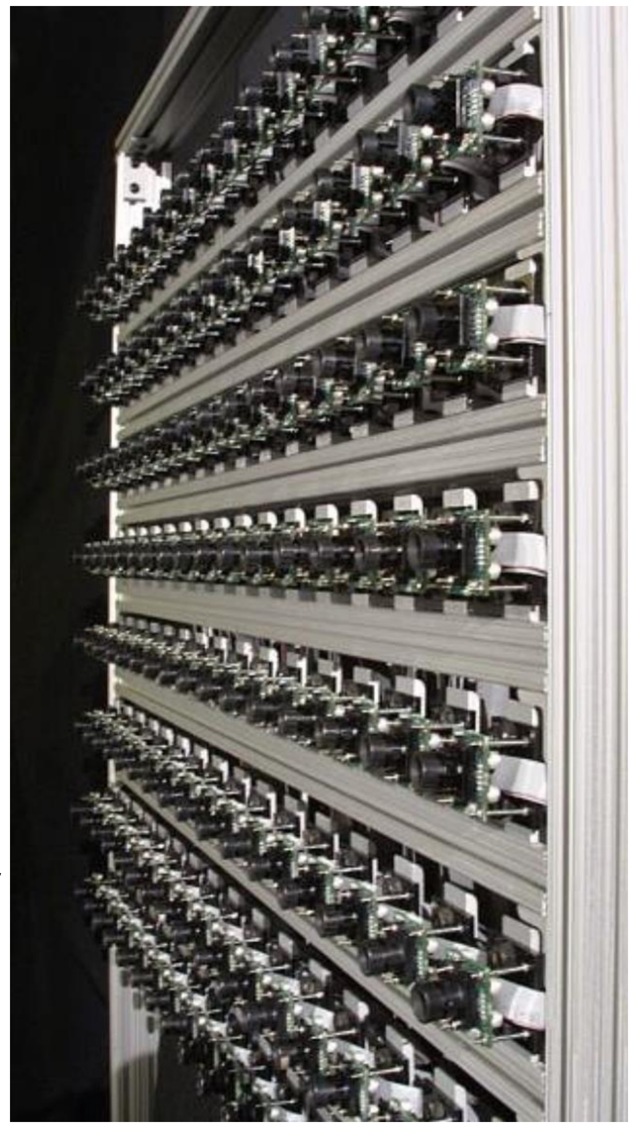


Lytro camera

Levoy and Hanrahan, SIGGRAPH 1996
Gortler et al. SIGGRAPH 1996

Lightfield / Lumigraph

- Previous approaches for modeling the Plenoptic Function
- Take a lot of pictures from many views
- Interpolate the rays to render a novel view



Stanford Gantry
128 cameras

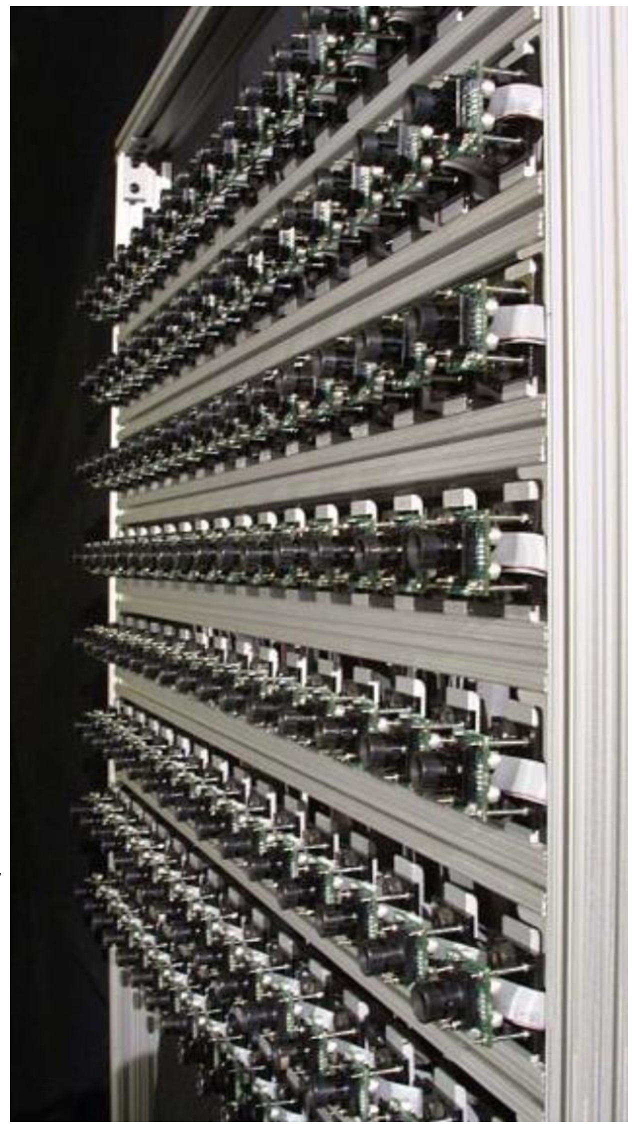
Lytro camera

Figure from Marc Levoy

Levoy and Hanrahan, SIGGRAPH 1996
Gortler et al. SIGGRAPH 1996

Lightfield / Lumigraph

- Previous approaches for modeling the Plenoptic Function
- Take a lot of pictures from many views
- Interpolate the rays to render a novel view



Stanford Gantry
128 cameras

Lytro camera

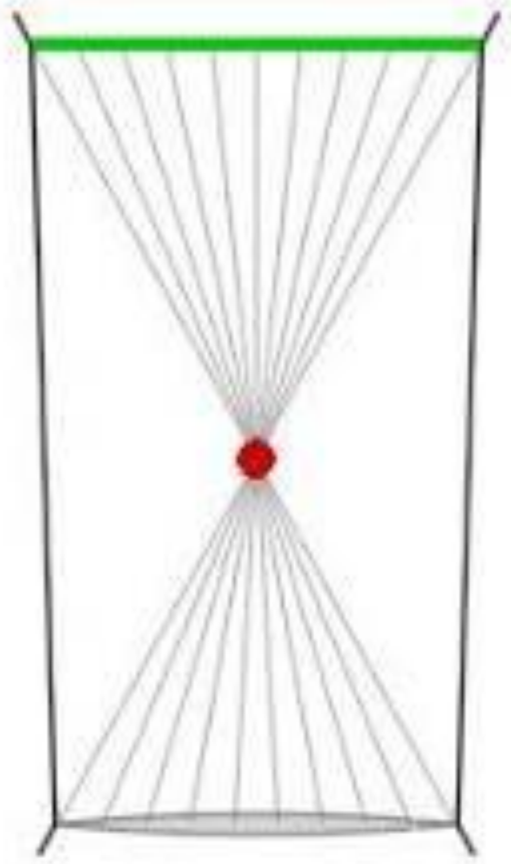
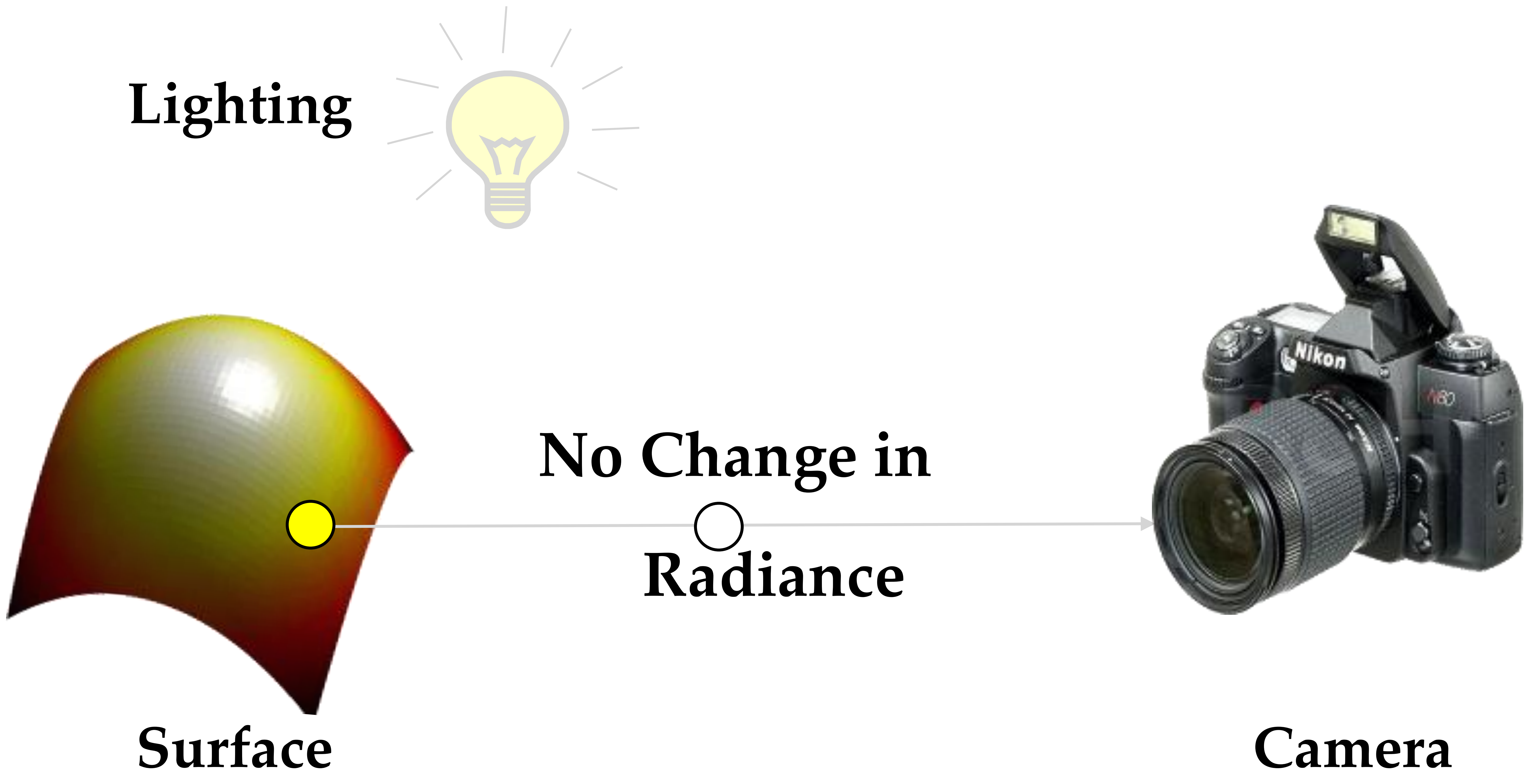


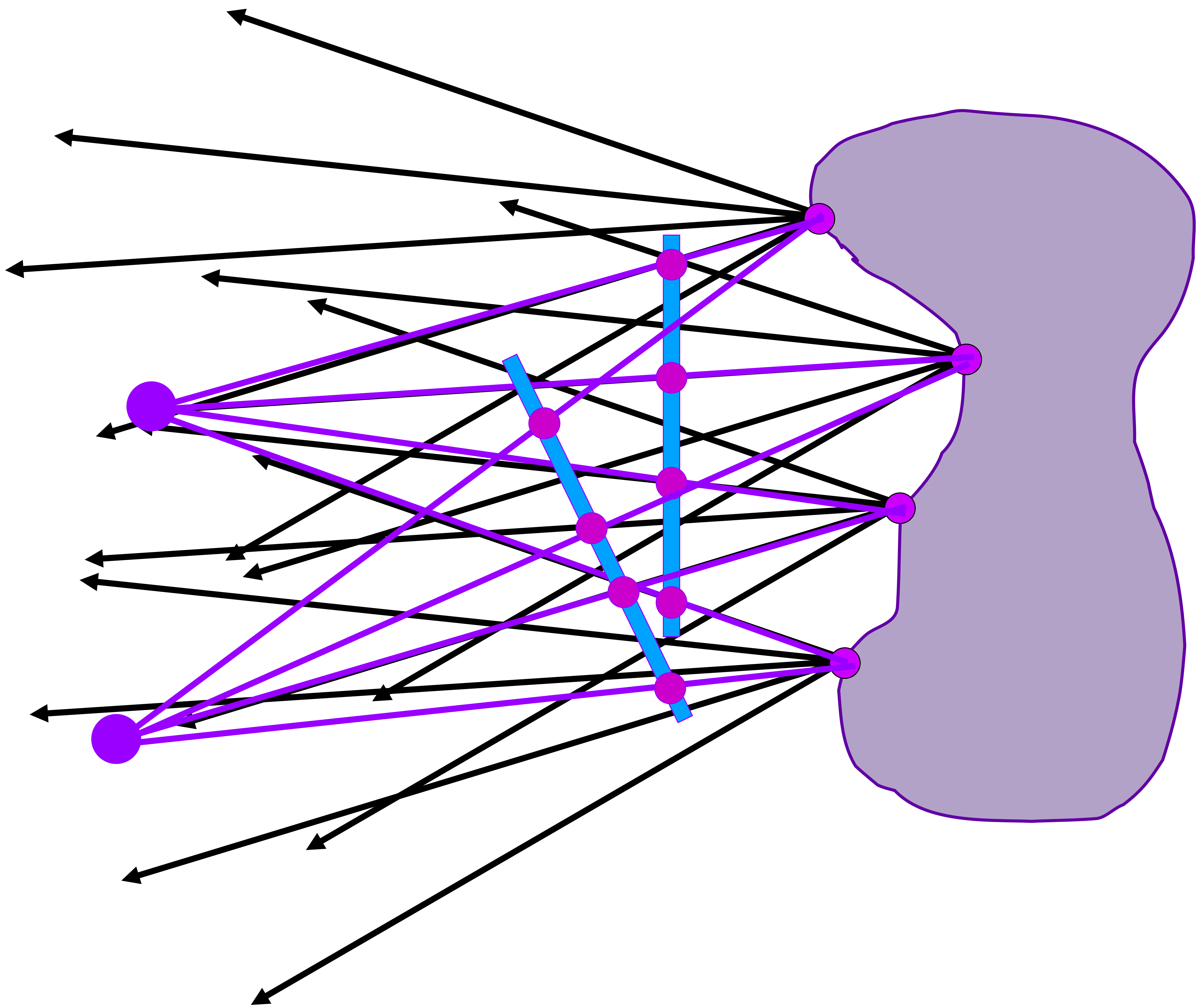
Figure from Marc Levoy

Big Assumption: a ray never changes color



True if there is no occlusion or fog

Synthesizing novel views with light field



Slide by Rick Szeliski and Michael Cohen

Image Based Rendering

- These methods are called Image Based Rendering, because they literally interpolate the ray colors to make a new image
- i.e. no 3D information is recovered (you have to know the camera)

Levoy and Hanrahan, SIGGRAPH 1996
Gortler et al. SIGGRAPH 1996

Ray Reuse Assumption

Because of this it only models the plenoptic surface:

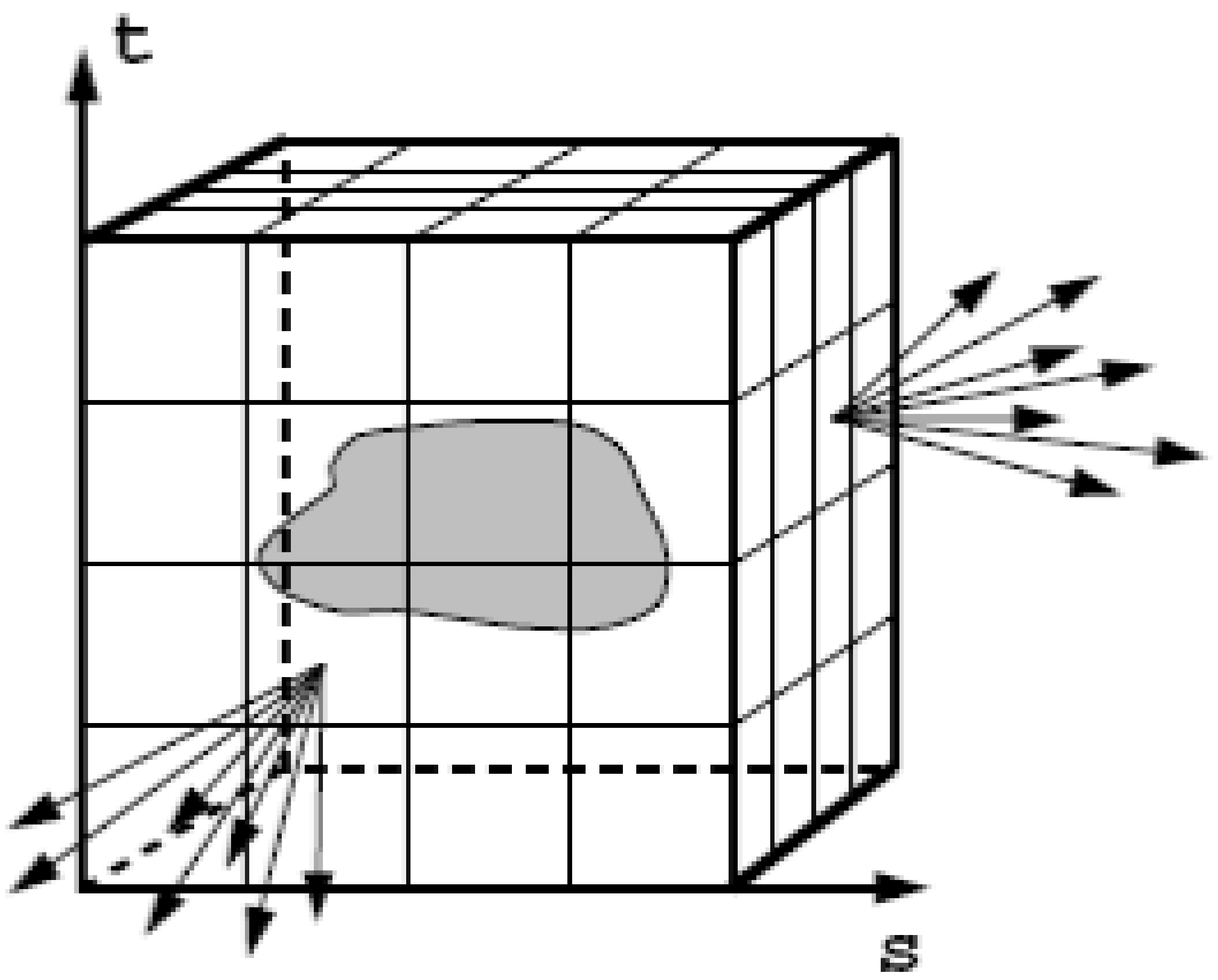


Figure 1: The surface of a cube holds all the radiance information due to the enclosed object.

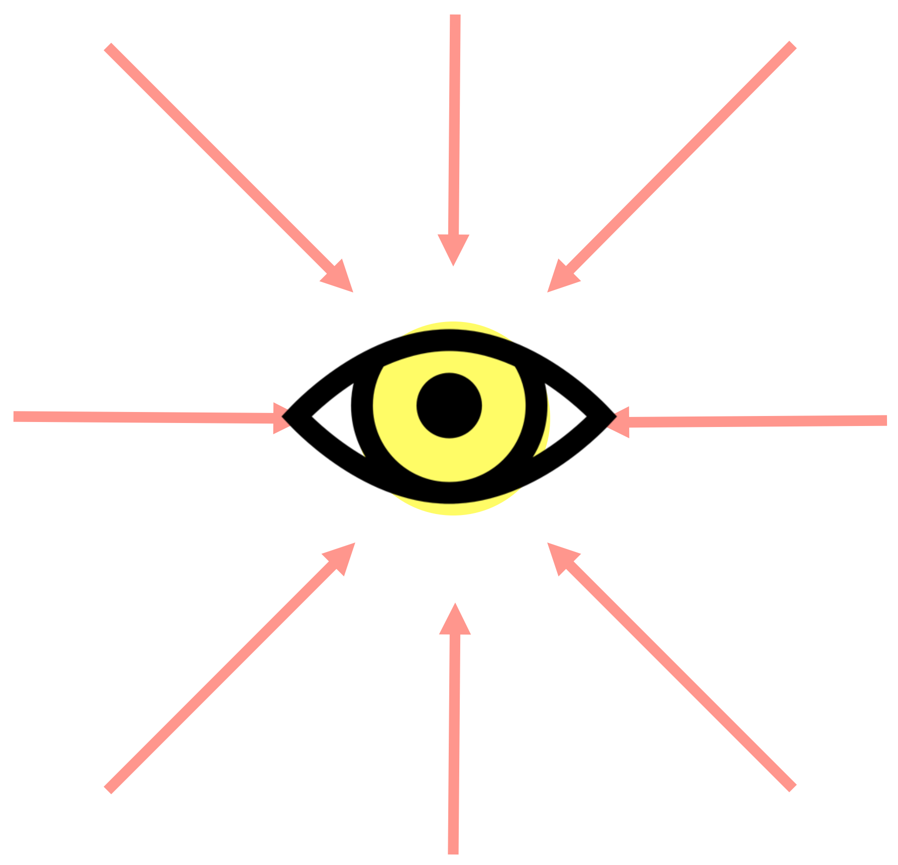
It's like

NeRF models a “flipped” Plenoptic Function

Plenoptic Function

$P(\underline{\theta, \phi}, V_X, V_Y, V_Z)$

Angle that we’re looking **at**

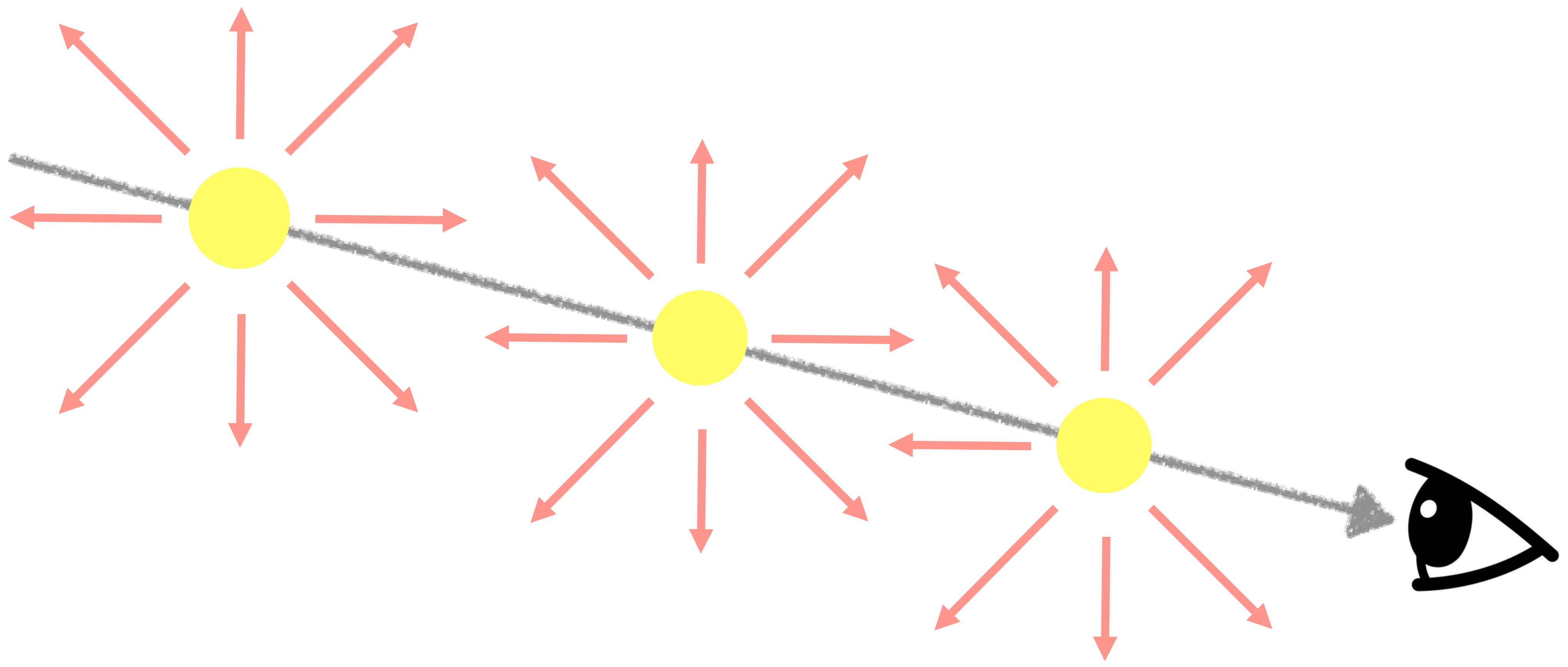


Plenoptic Function

NeRF

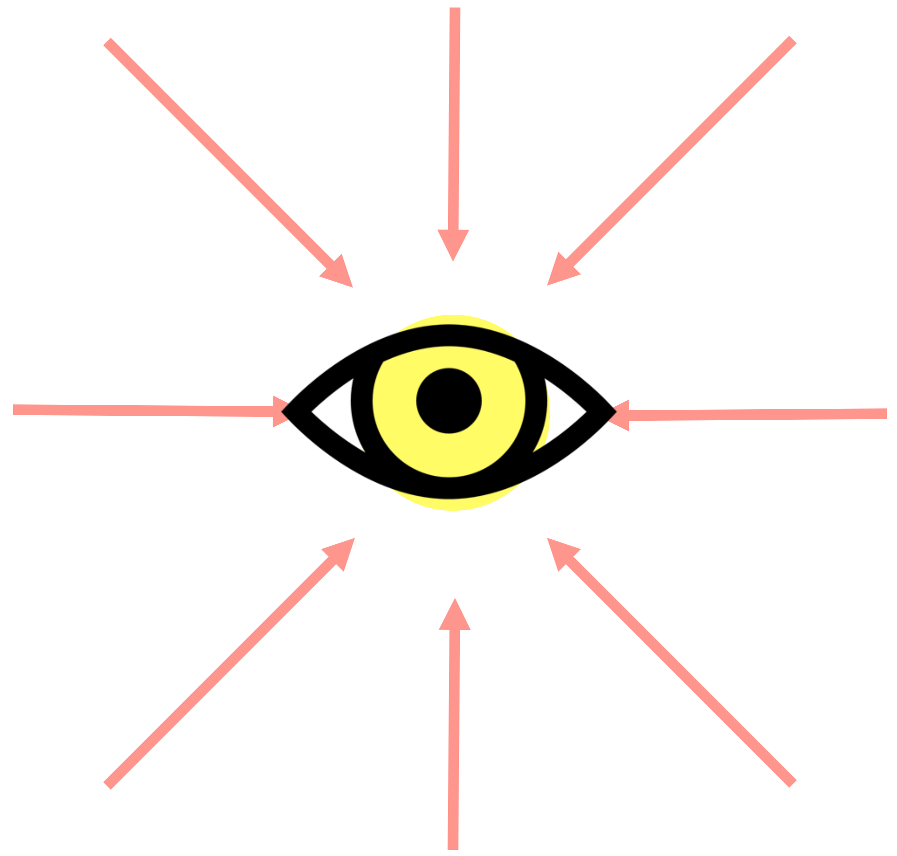
$P(\underline{\theta, \phi}, V_X, V_Y, V_Z)$

Angle that we’re looking **from**

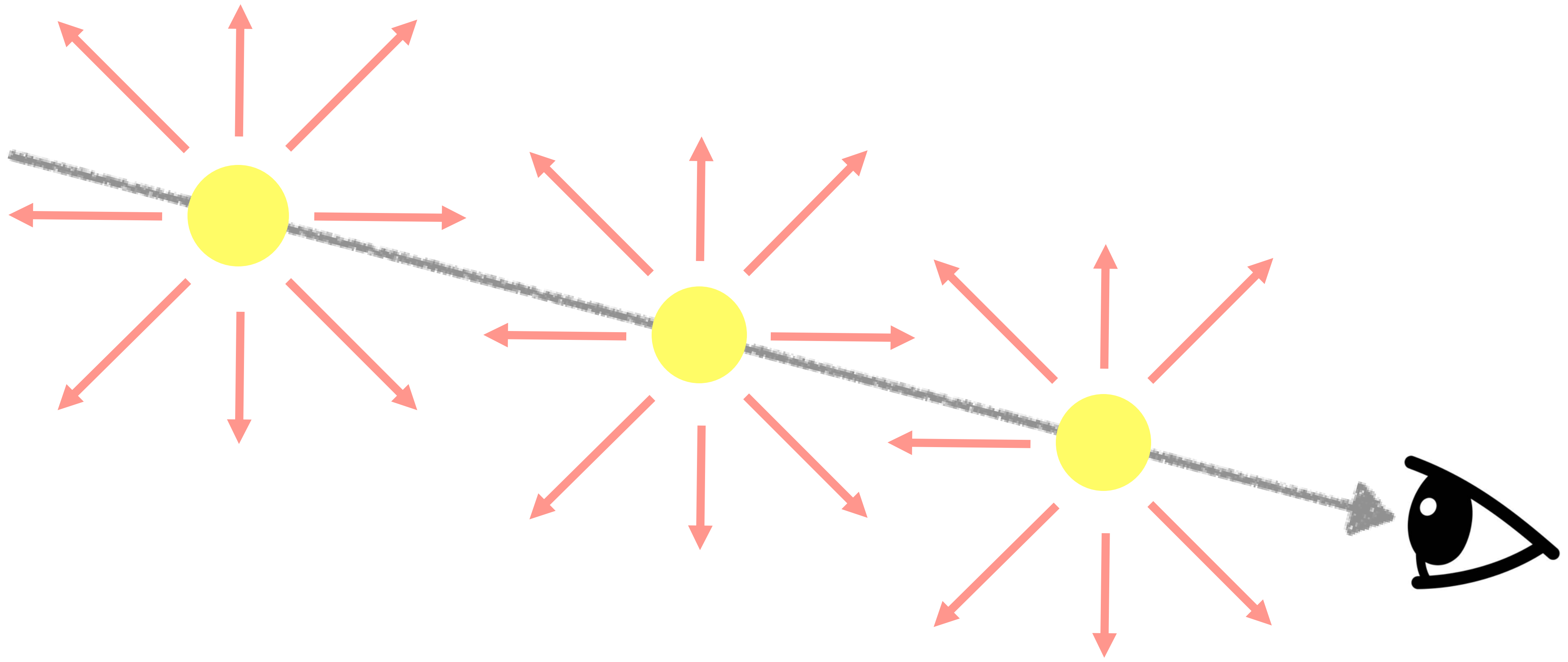


NeRF

An important difference



Plenoptic Function



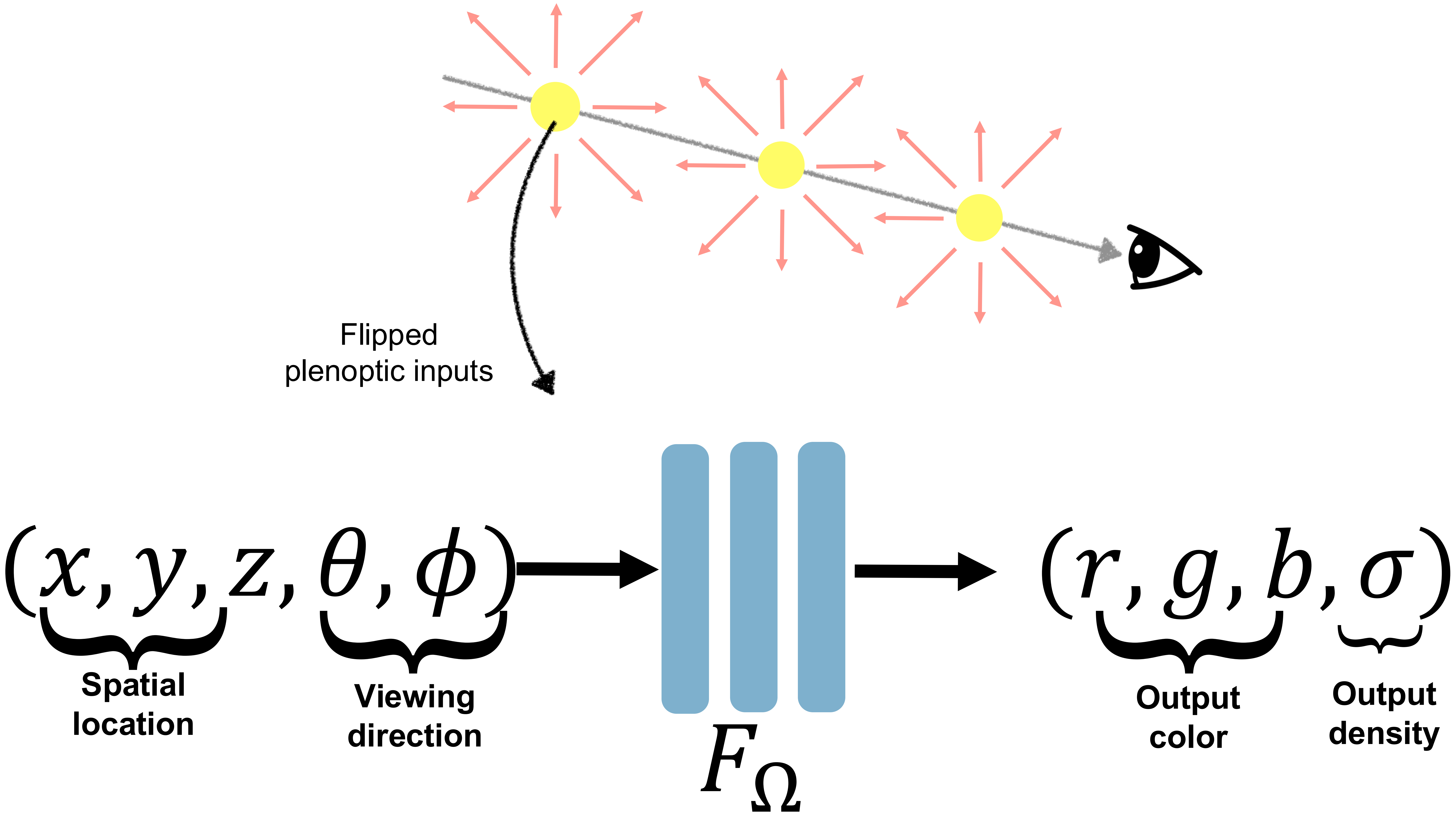
NeRF

NeRF requires *integration* along the viewing ray to compute the Plenoptic Function

Bottom line: it models a 5D plenoptic function!

Visualizing view direction

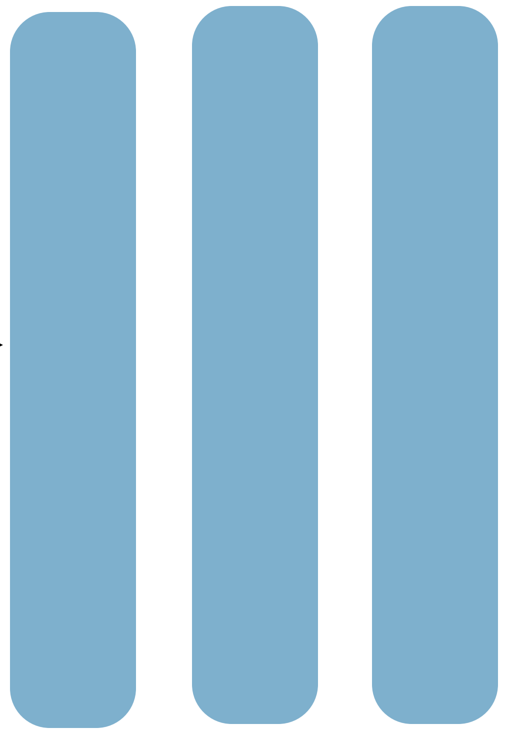
- NeRF can capture non-Lambertian (specular, shiny surfaces) because it models the color in a view-dependent manner
- This is hard to do with meshes unless you model the physical materials & lighting interactions


Flipped
plenoptic inputs
(x, y, z, θ, ϕ)
Spatial
location
Viewing
direction
F_Ω
(r, g, b, σ)
Output
color
Output
density

Density: Second key difference from lightfields, plenoptic function

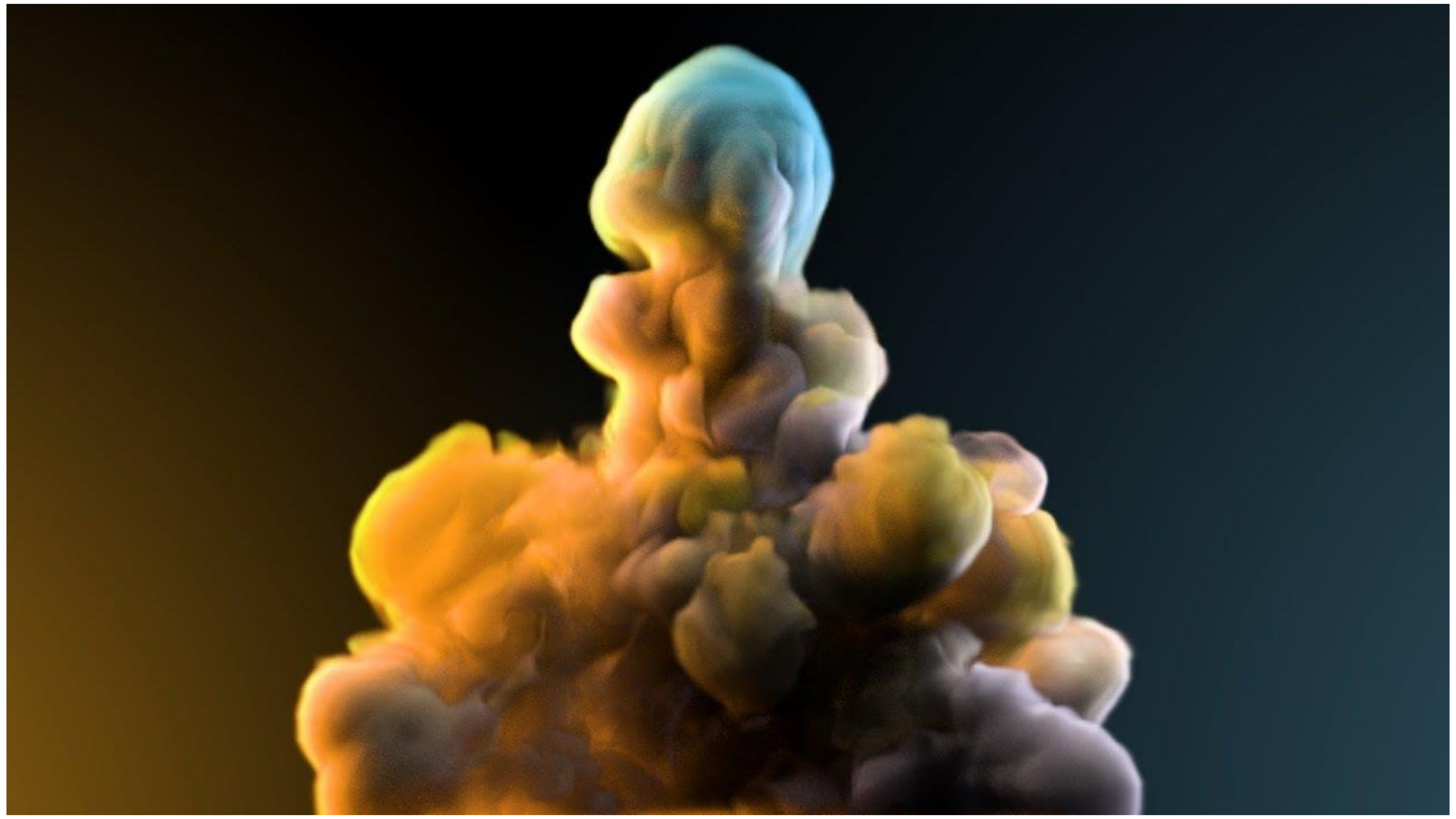
- Continuous probability density function (PDF) over “stuff”
- Connected to opacity: high density == very opaque, solid

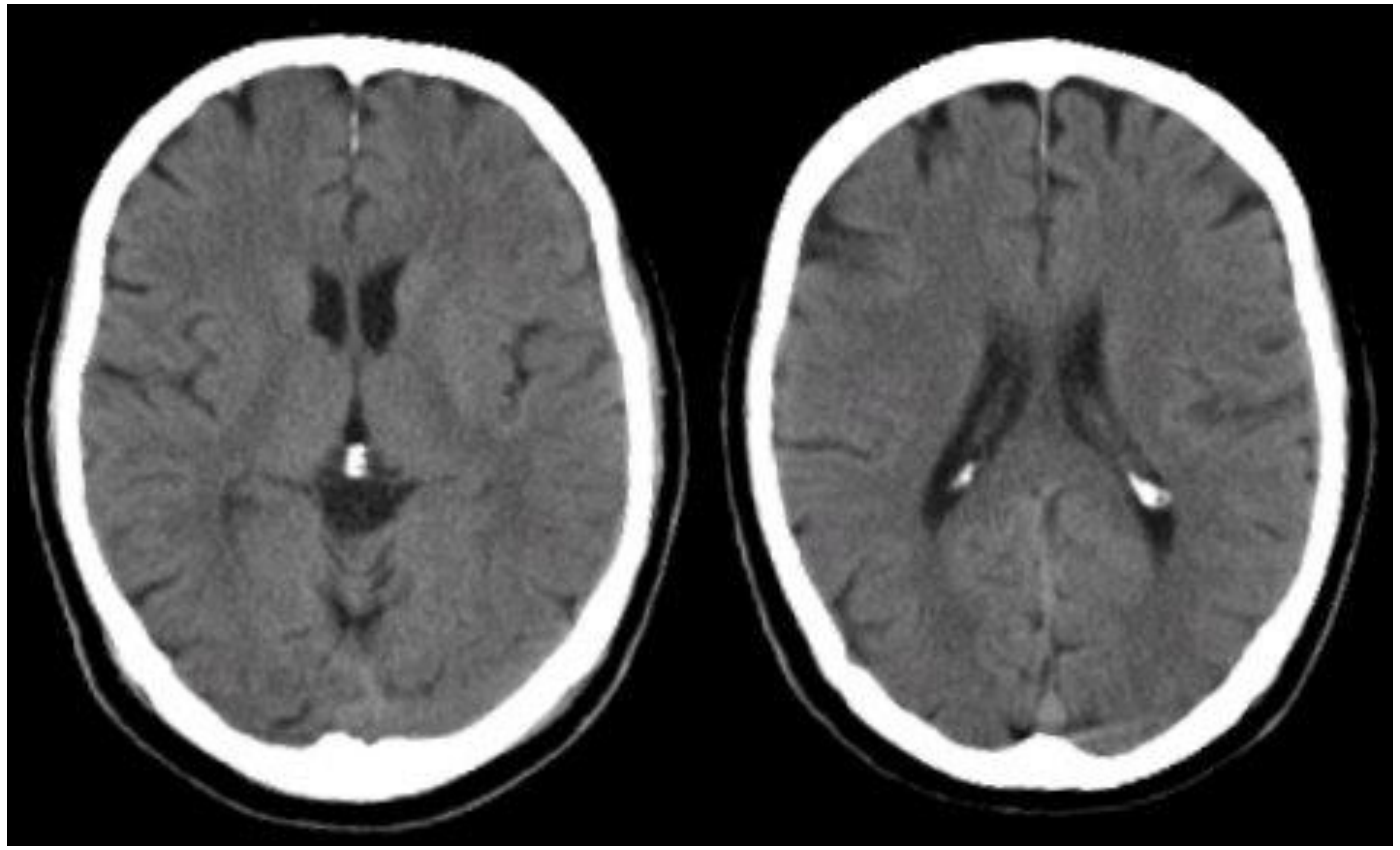
$$(\underbrace{x, y, z}_{\text{Spatial location}}, \underbrace{\theta, \phi}_{\text{Viewing direction}}) \rightarrow F_{\Omega} \rightarrow (\underbrace{r, g, b}_{\text{Output color}}, \underbrace{\sigma}_{\text{Output density}})$$



Examples of Density Fields?

Examples of Density Fields?





NeRF represents the whole world with a “fuzzy” model!

Where NeRF stands

- can do Image Based Rendering well, while also being a 3D representation
- Does not suffer from limitations of surface models
- Easy to optimize from images

Appearance Based Reconstruction (Image Based Rendering)

NeRFs

Physics based Reconstruction (3D Surface Modeling)

Lightfield/Lumigraph (No 3D representation)

Layered Depth Images (LDIs)

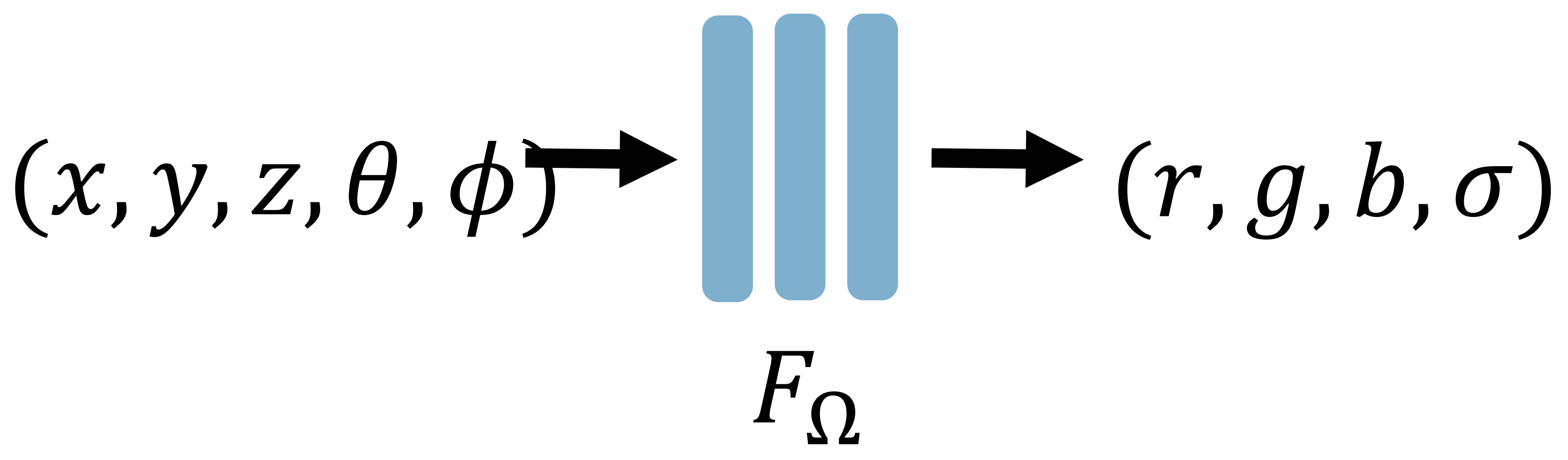
Multi-Plane Images (MPIs)

One 3D Surface, View-Dependent Texture Mapping

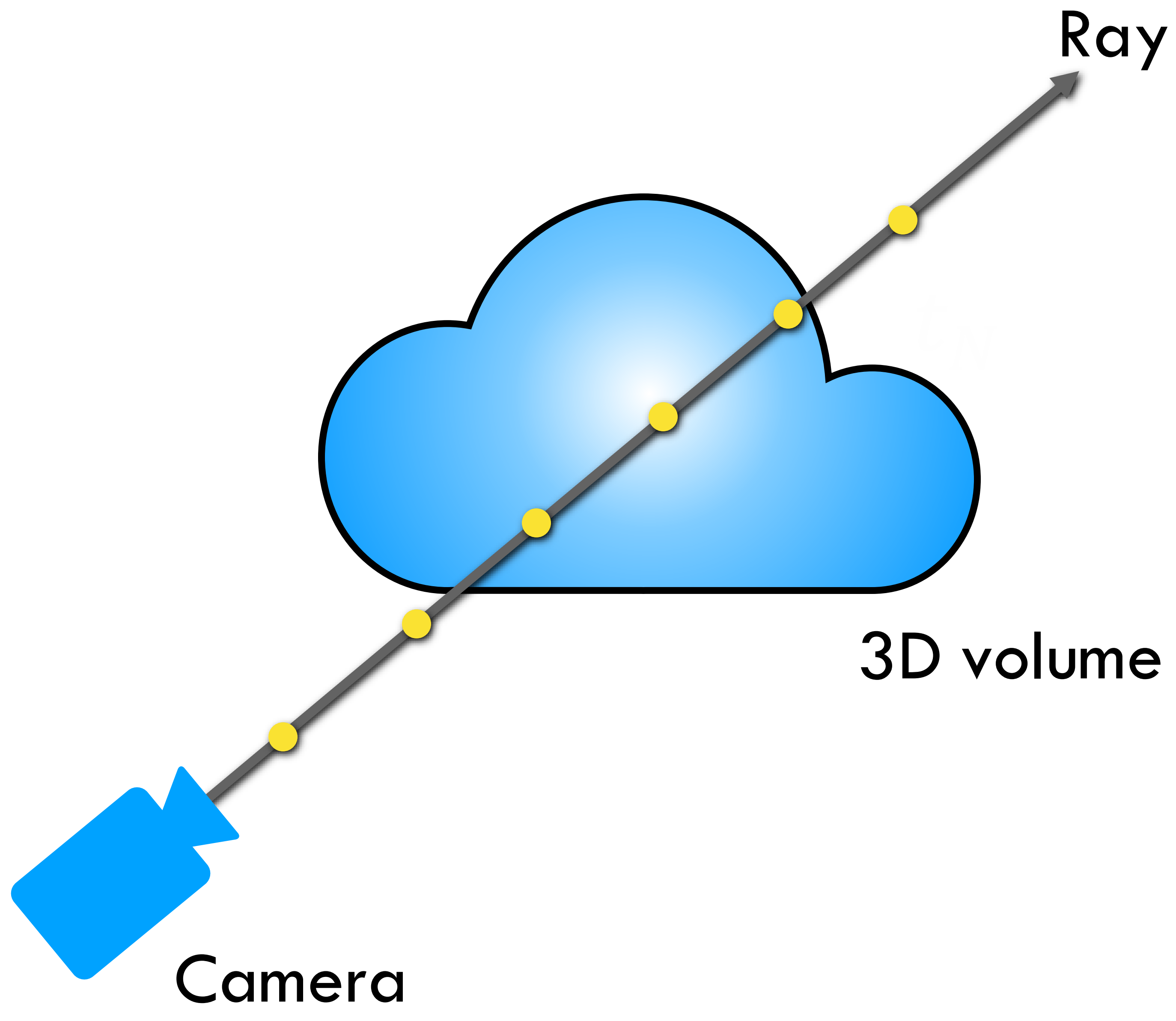
One 3D Surface, Single Albedo Texture

Conventional Graphics Pipeline

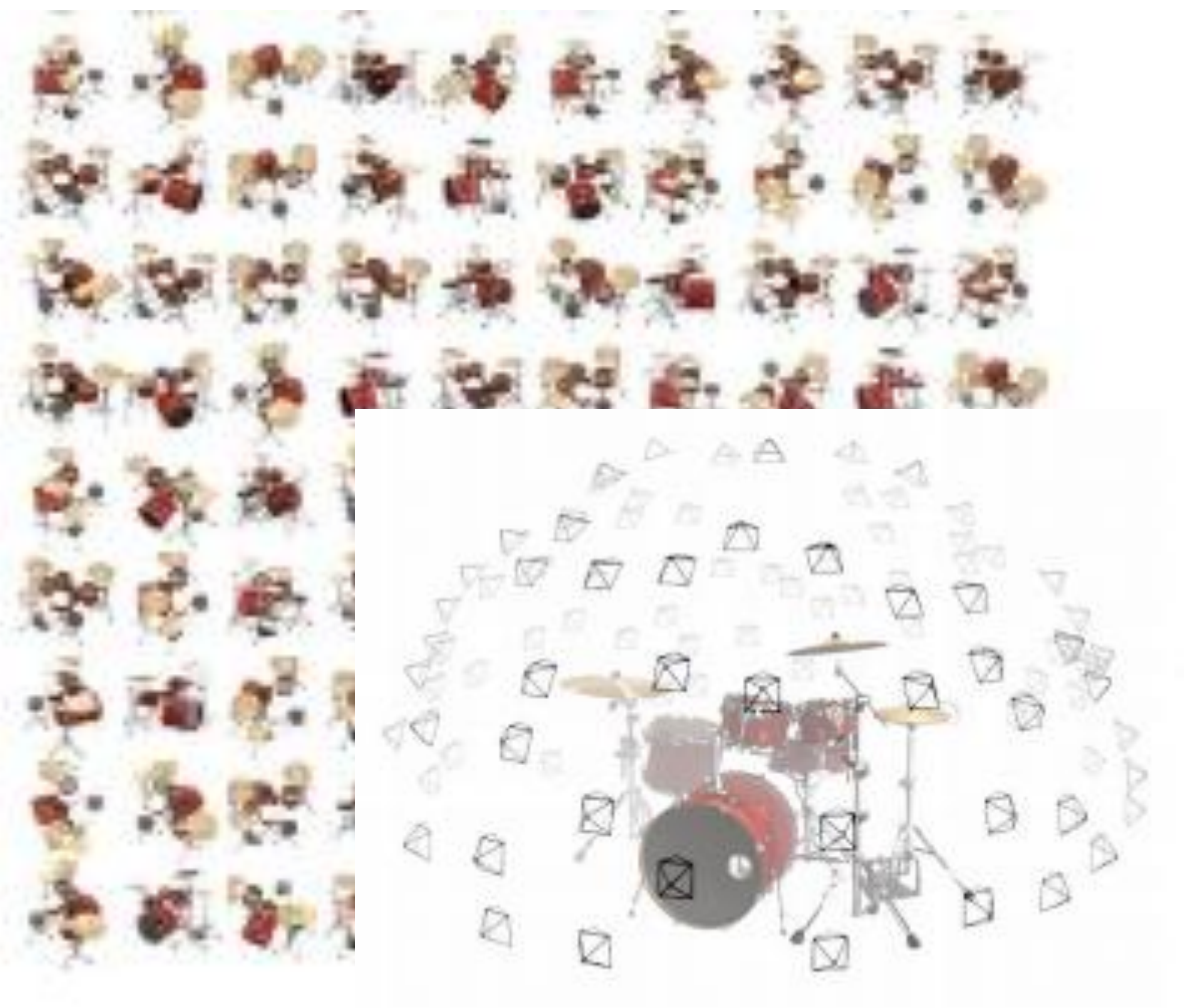
NeRF: 3 Key Components



Neural Volumetric 3D Scene Representation



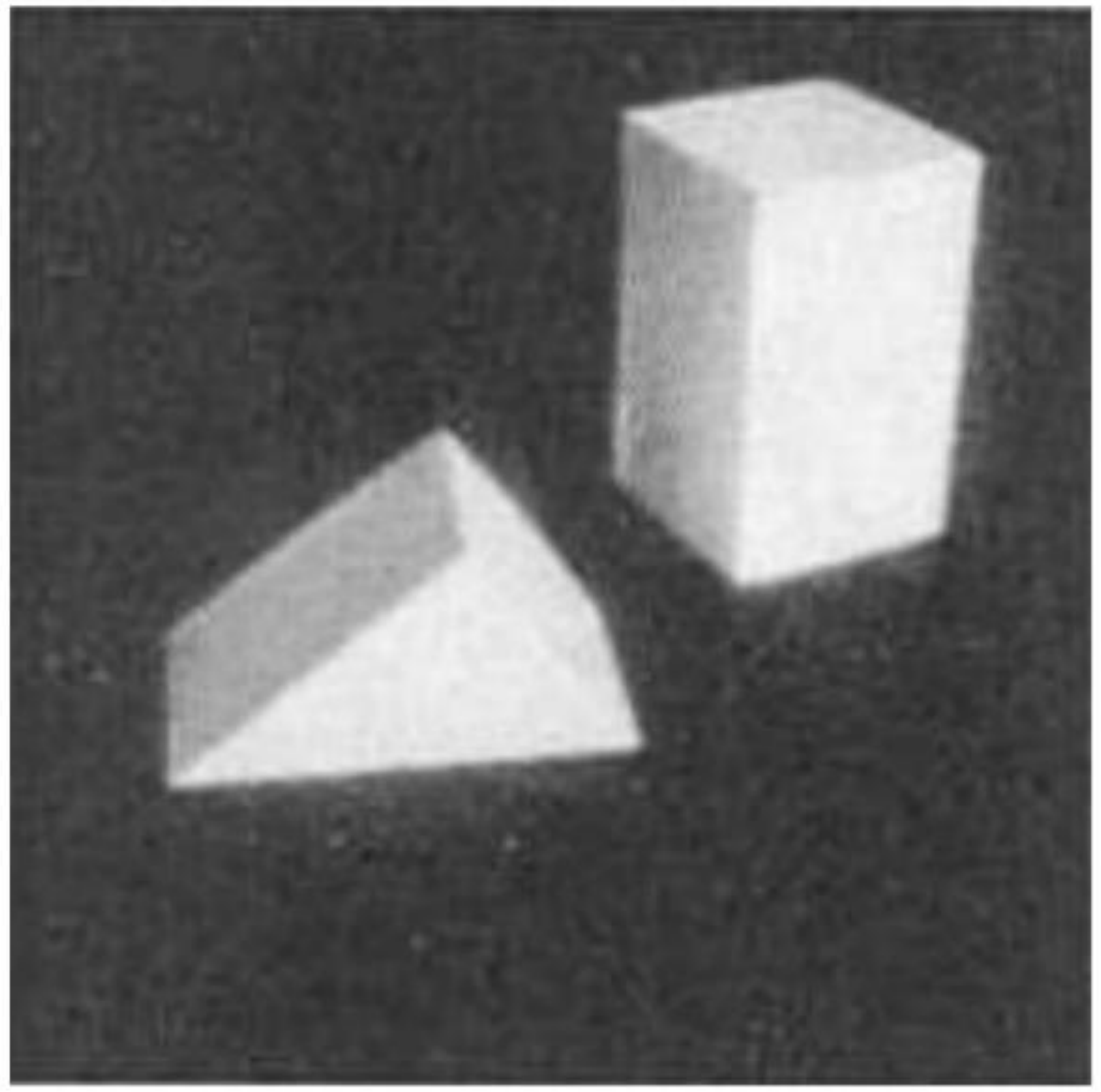
Differentiable Volumetric Rendering Function



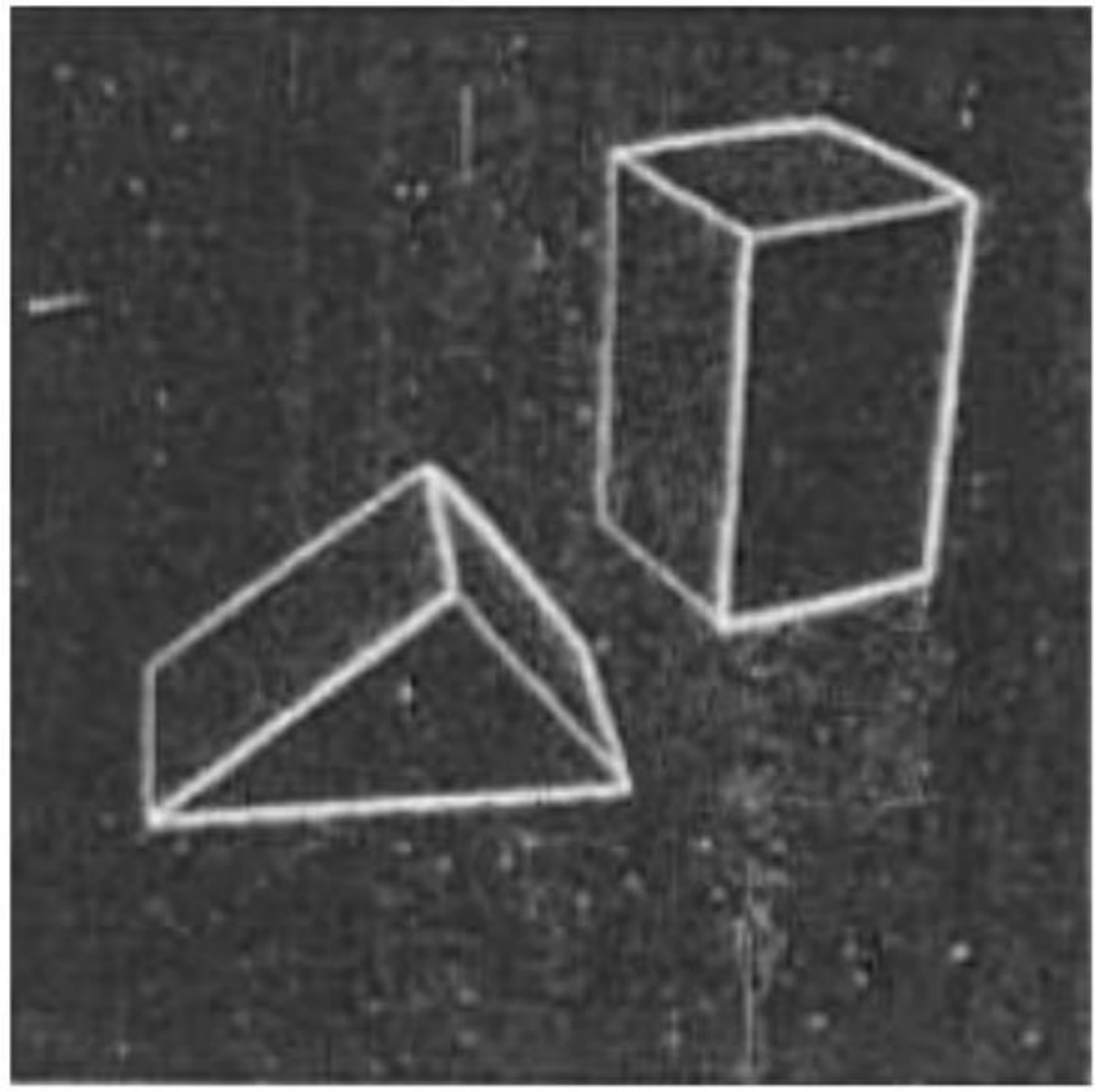
Optimization via **Analysis-by-Synthesis**

“Analysis-by-Synthesis"

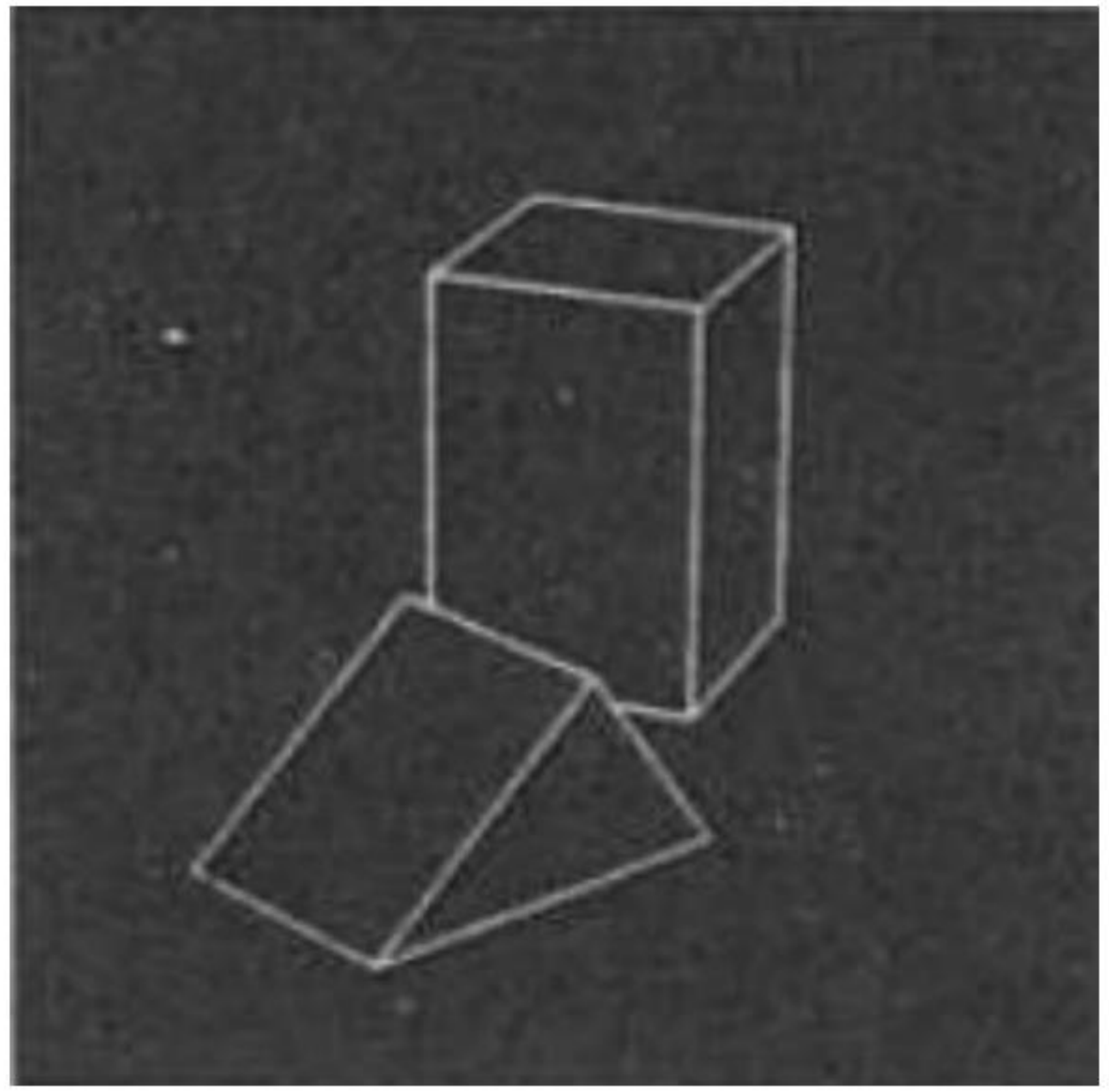
Larry Roberts
“Father of Computer Vision”



Input image



2x2 gradient operator



computed 3D model
rendered from new viewpoint

- History goes way back to the **first** Computer Vision paper!
  Roberts: Machine Perception of Three-Dimensional Solids, MIT, 1963

“Analysis-by-Synthesis"

- Search for a world state from which you can explain many observations through synthesis
- In English: “If you understand (analyze) something, you can create (synthesize) it”

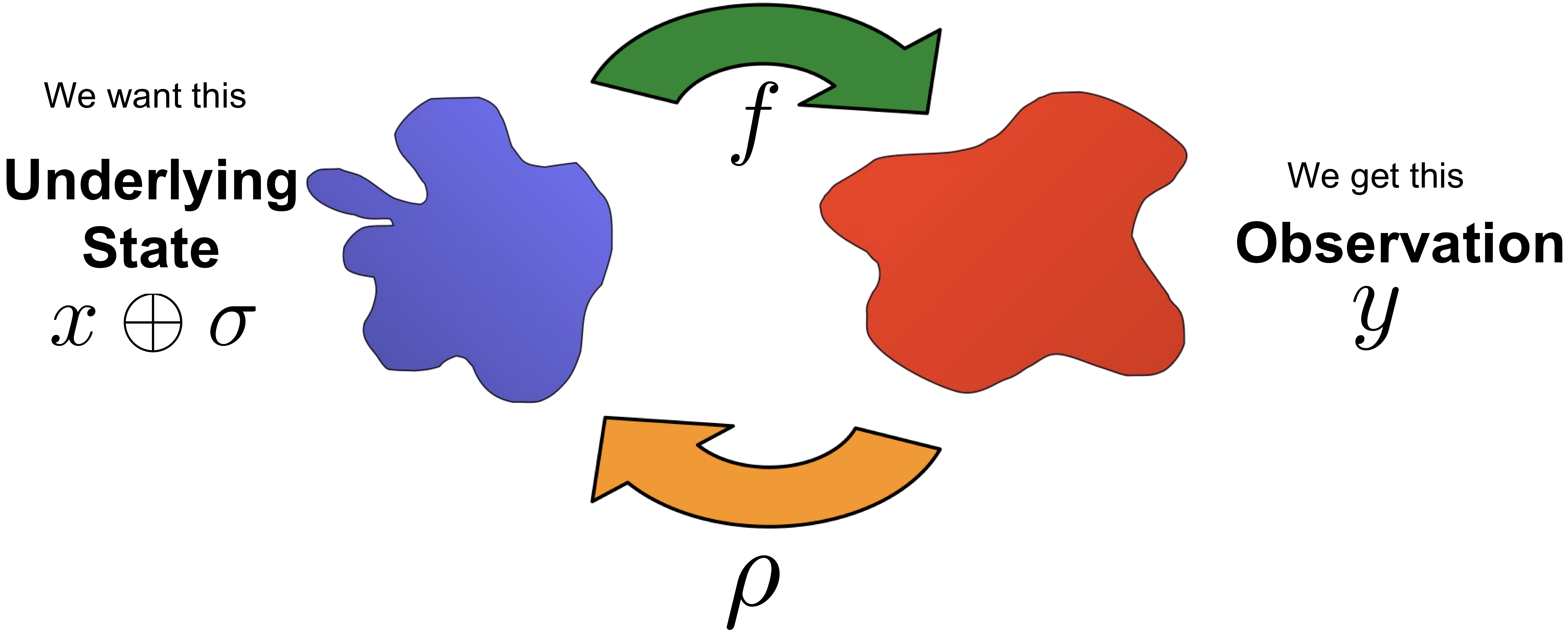
“Analysis-by-Synthesis"

- Search for a world state from which you can explain many observations through synthesis
- In English: “If you understand (analyze) something, you can create (synthesize) it”
- (For NeRF): “If you really know what a scene looks like, you can render it from any view”
- (For Chemistry): “If you know how a molecule is structured, you can synthesize it from other molecules”
- Commonly used paradigm across CV!

More specifically…

- In NeRF we iteratively update a 3D model (analyze) by synthesizing image observations and running gradient descent through our model.

Analysis-by-Synthesis vs Machine Learning



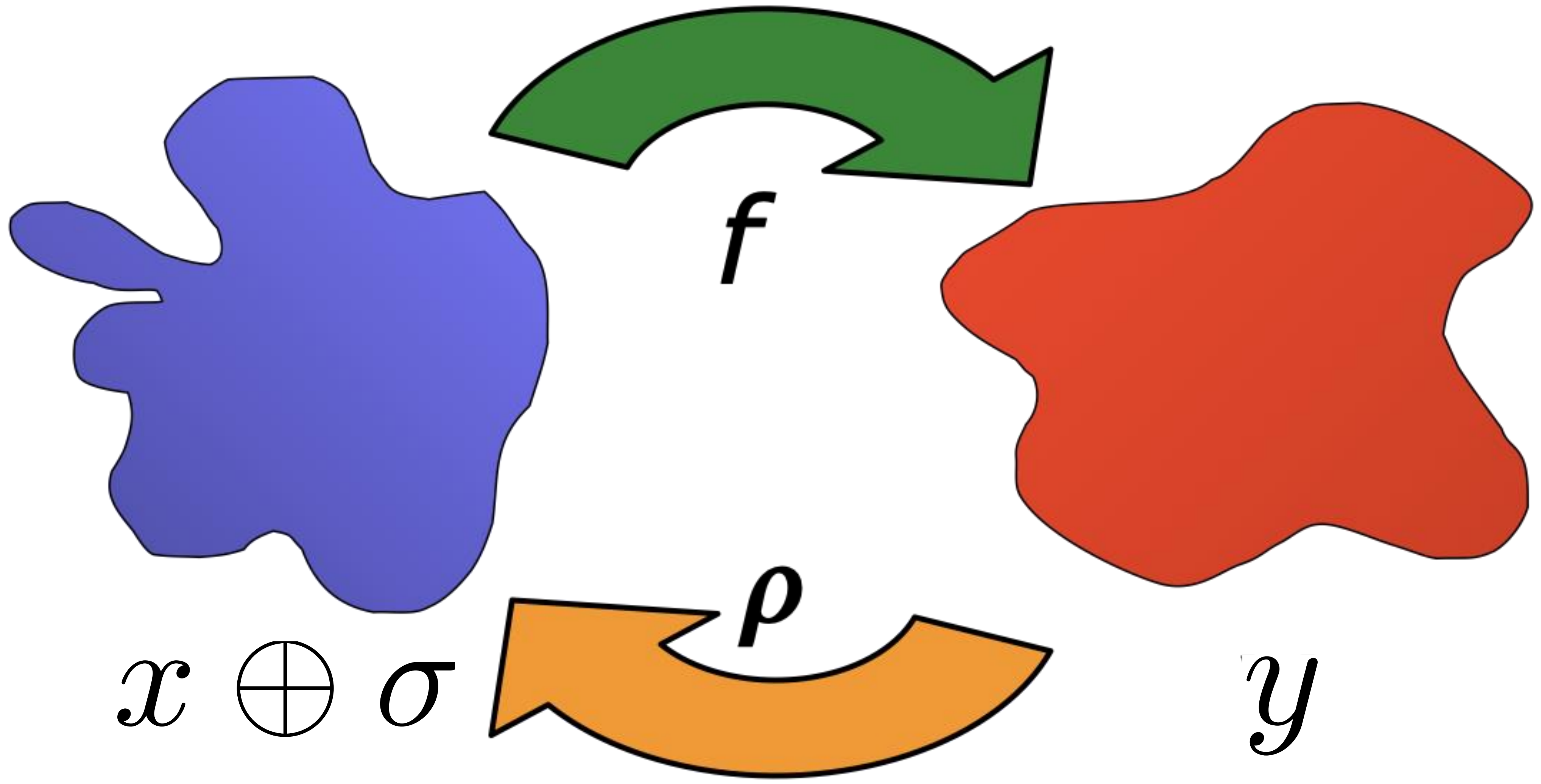
Analysis-by-Synthesis vs Machine Learning

AbS: Estimate x from y by reversing the forward model

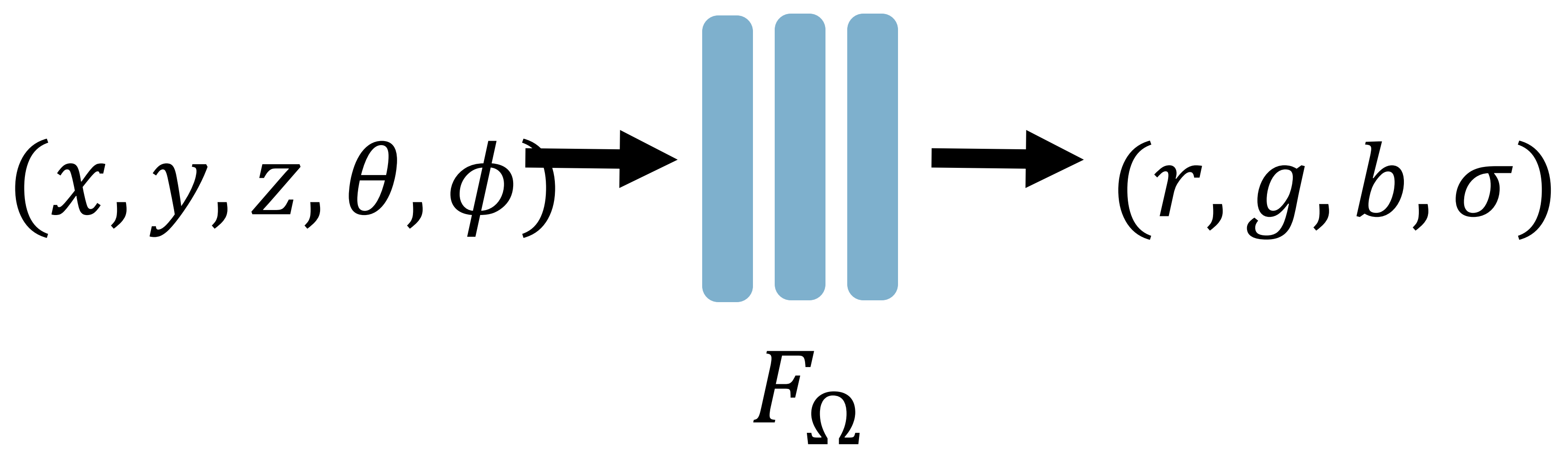
$$\rho(y) = \arg\min_{x\in X} \ell_y(f(x), y)$$

ML: Estimate *ρ* from examples

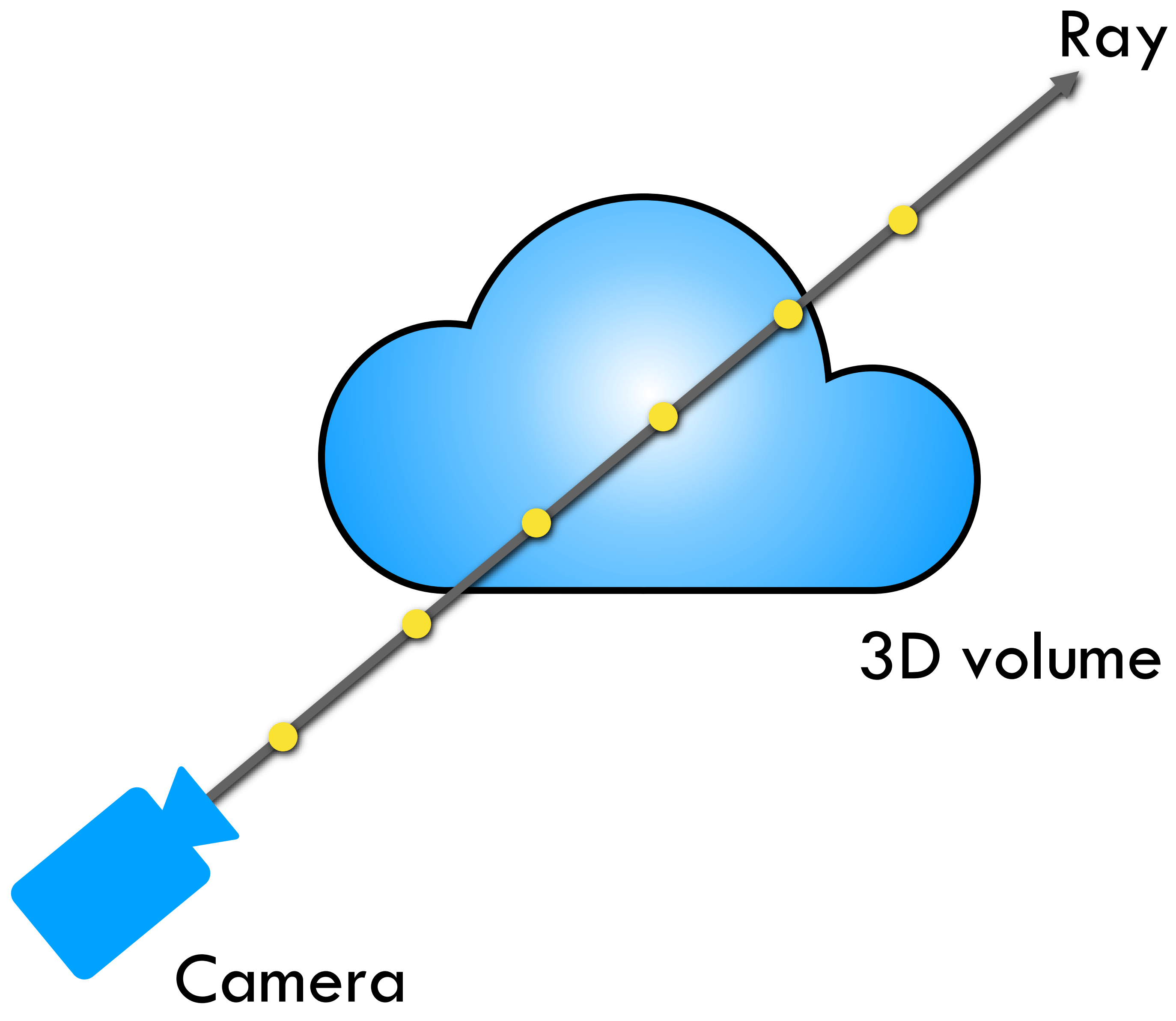
$$\rho(y) = \overset{\rho}{\arg\min_{\varphi\in\mathcal{F}}} \sum_{I=1}^{n} l_x(\rho(y_i), x_i)$$



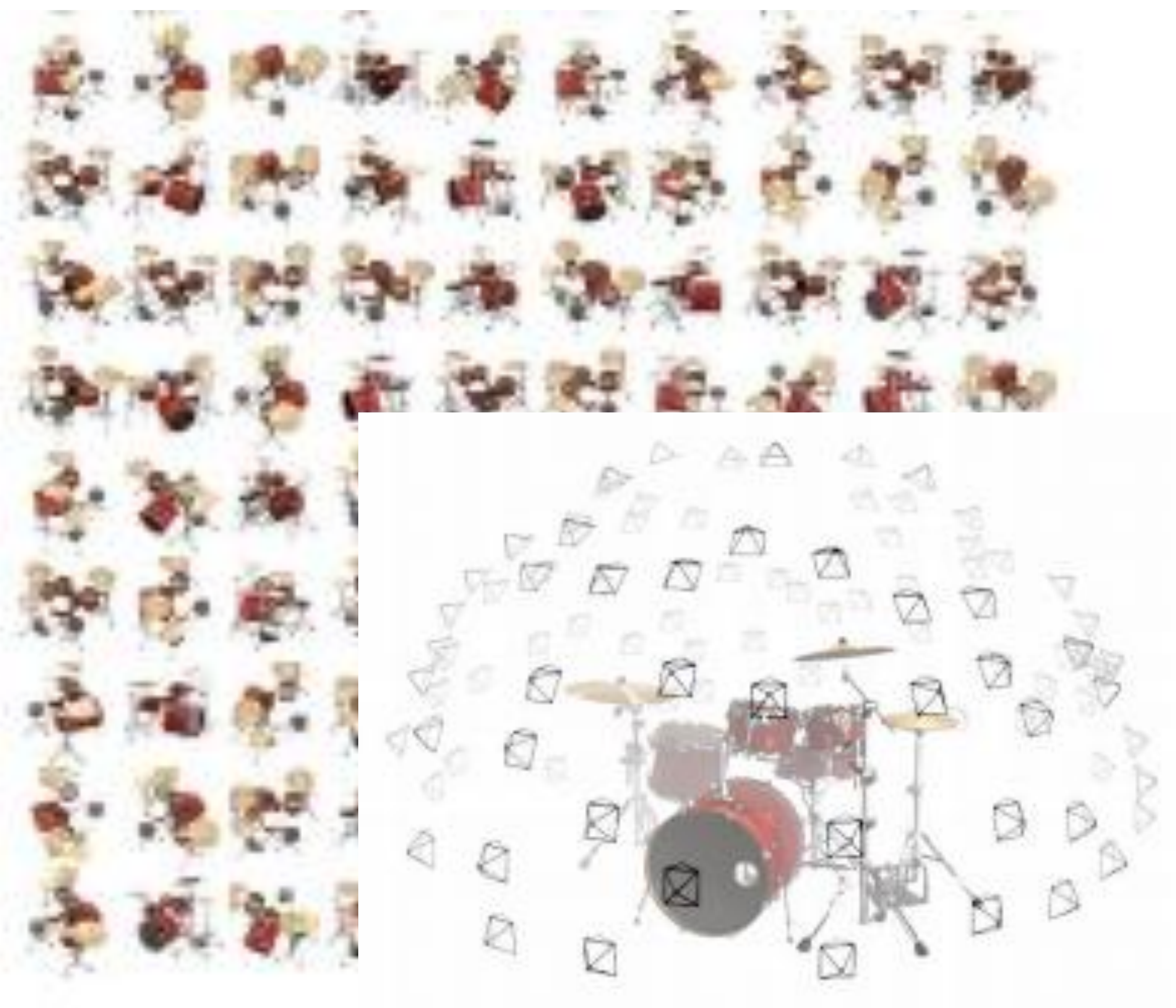
Next time: Details on Components



Neural Volumetric 3D Scene Representation



Differentiable Volumetric Rendering Function



Optimization via Analysis-by-Synthesis