THE
PROBLEM

"Tired of complex database queries and endless coding just to get insights?"

"What if there was a simpler way?"

Computer Vision

Inverse Graphics & Neural Radiance Fields (NeRFs)

NeRF Slides adapted from material courtesy of Pratul Srinivasa

Rendering in computer graphics

3D
Scene
Representation

Rendering

Adapted from material from Pratul Srinivas

Computer vision as inverse rendering

Inverse Rendering

3D
Scene
Representation

Rendering

Adapted from material from Pratul Srinivas

Neural Radiance Fields (NeRF) as an approach to inverse rendering

Inverse Rendering

Neural Radiance Field

Rendering

Adapted from material from Pratul Srinivas

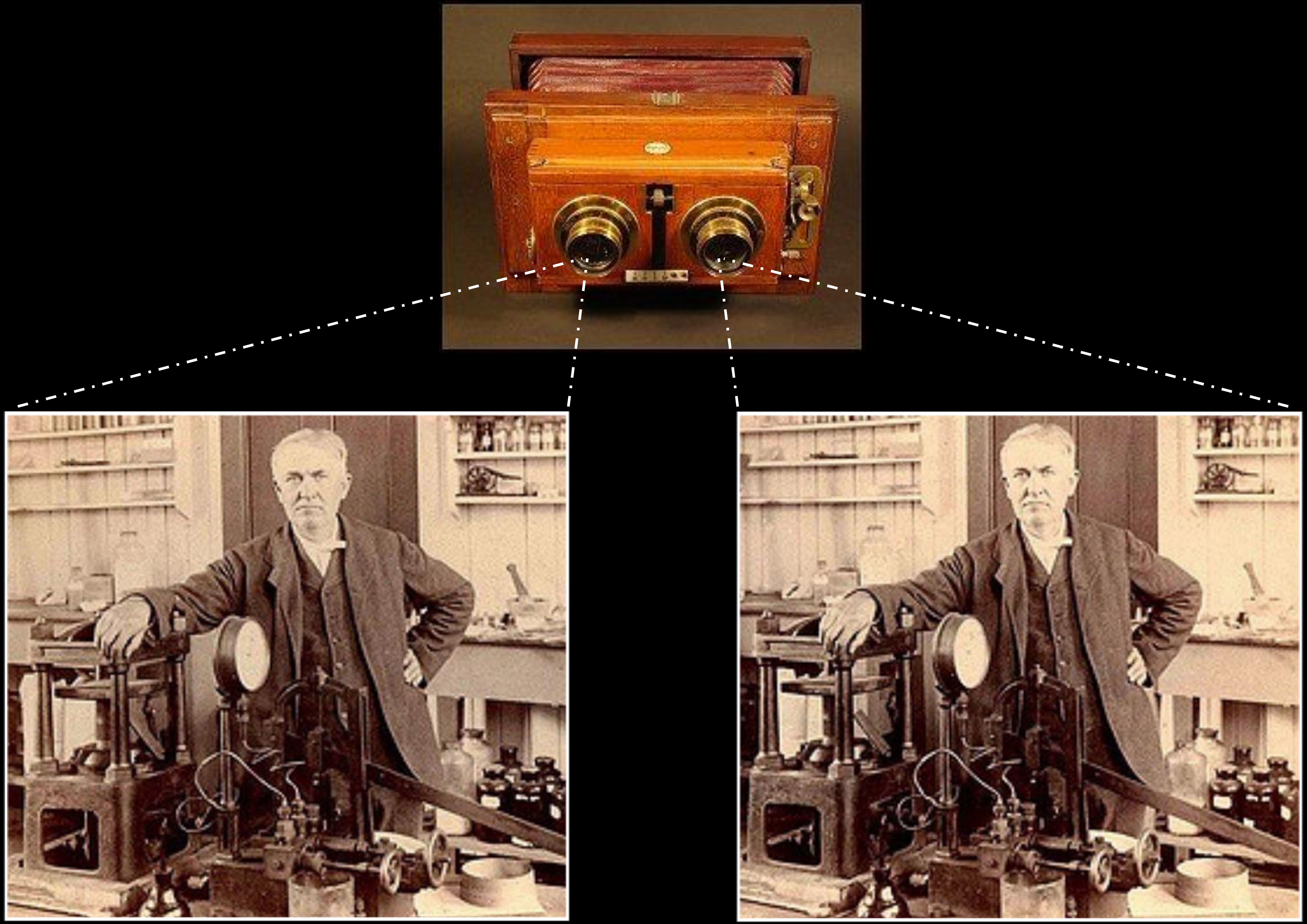
Deep learning for 3D reconstruction

- Previously: we reconstruct geometry by running stereo or multi-view stereo on a set of images
  - "Classical" approach
- How can we leverage powerful tools of deep learning?
  - Deep neural networks
  - GPU-accelerated stochastic gradient descent

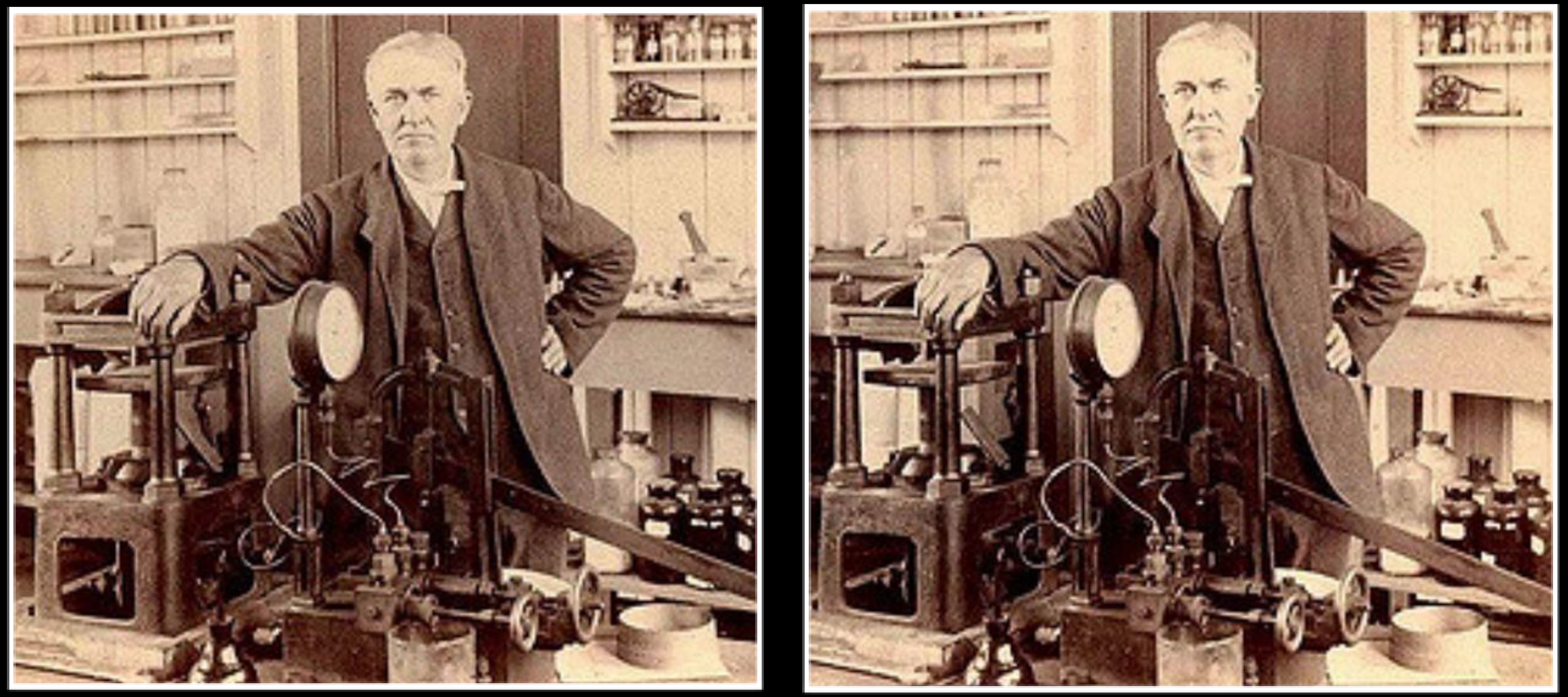
NeRF and related methods – Key ideas

- We need to create a loss function and a scene representation that we can optimize using gradient descent to reconstruct the scene
- *Differentiable rendering*

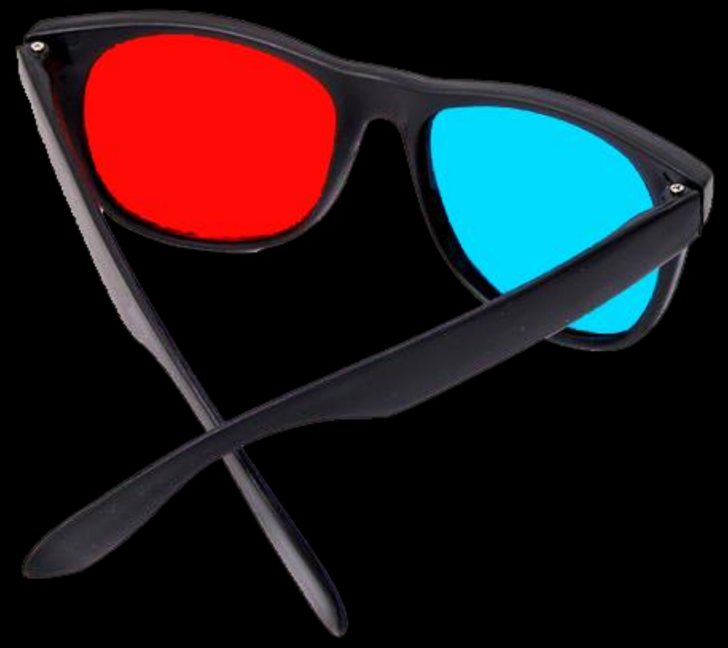
Side Topic: Stereo Photography



Stereo Photography



Viewing Devices



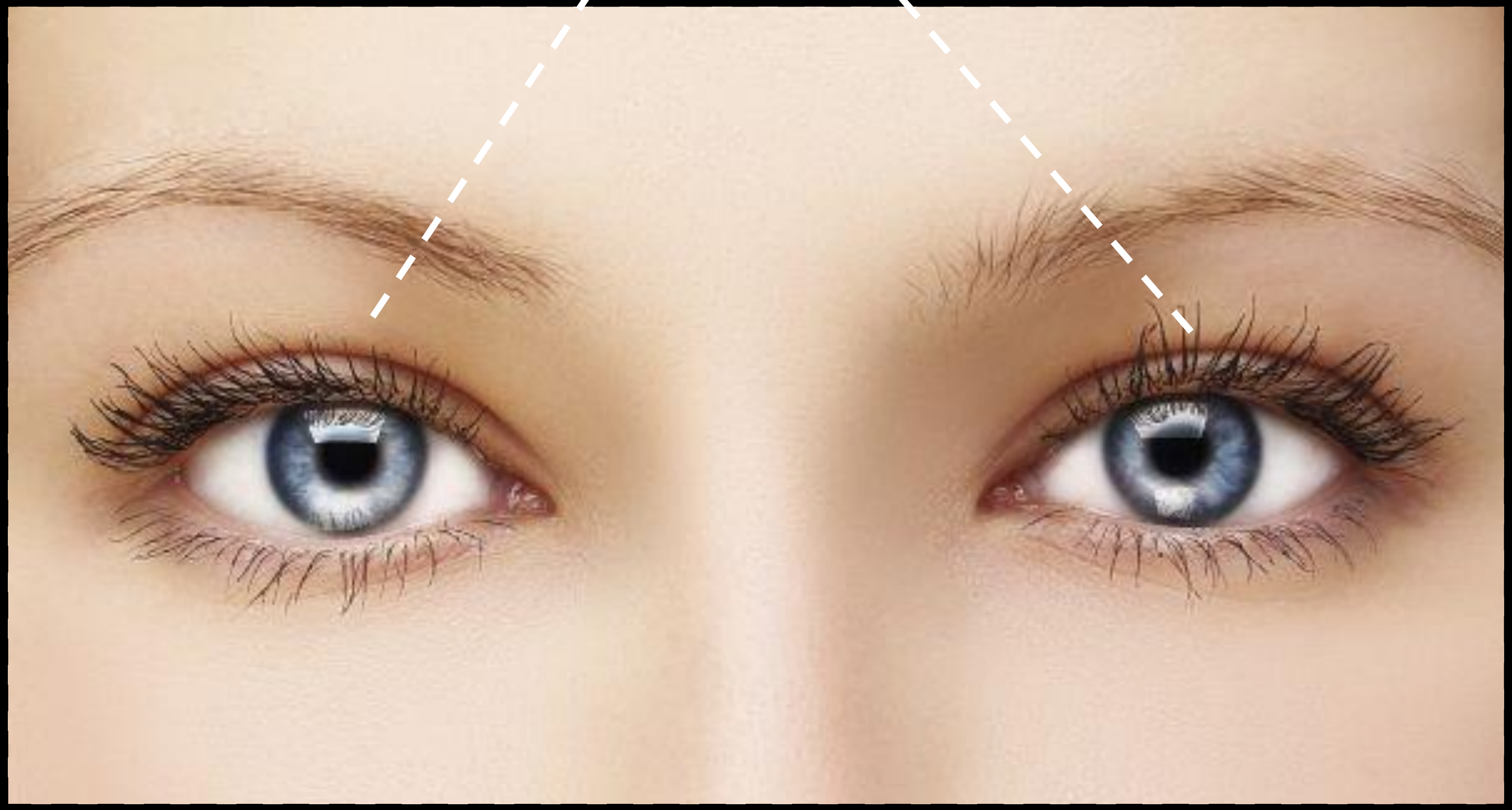
Stereo Photography

Queen Victoria at World Fair, 1851

Stereo Photography

Issue: Narrow Baseline

~6.5 cm



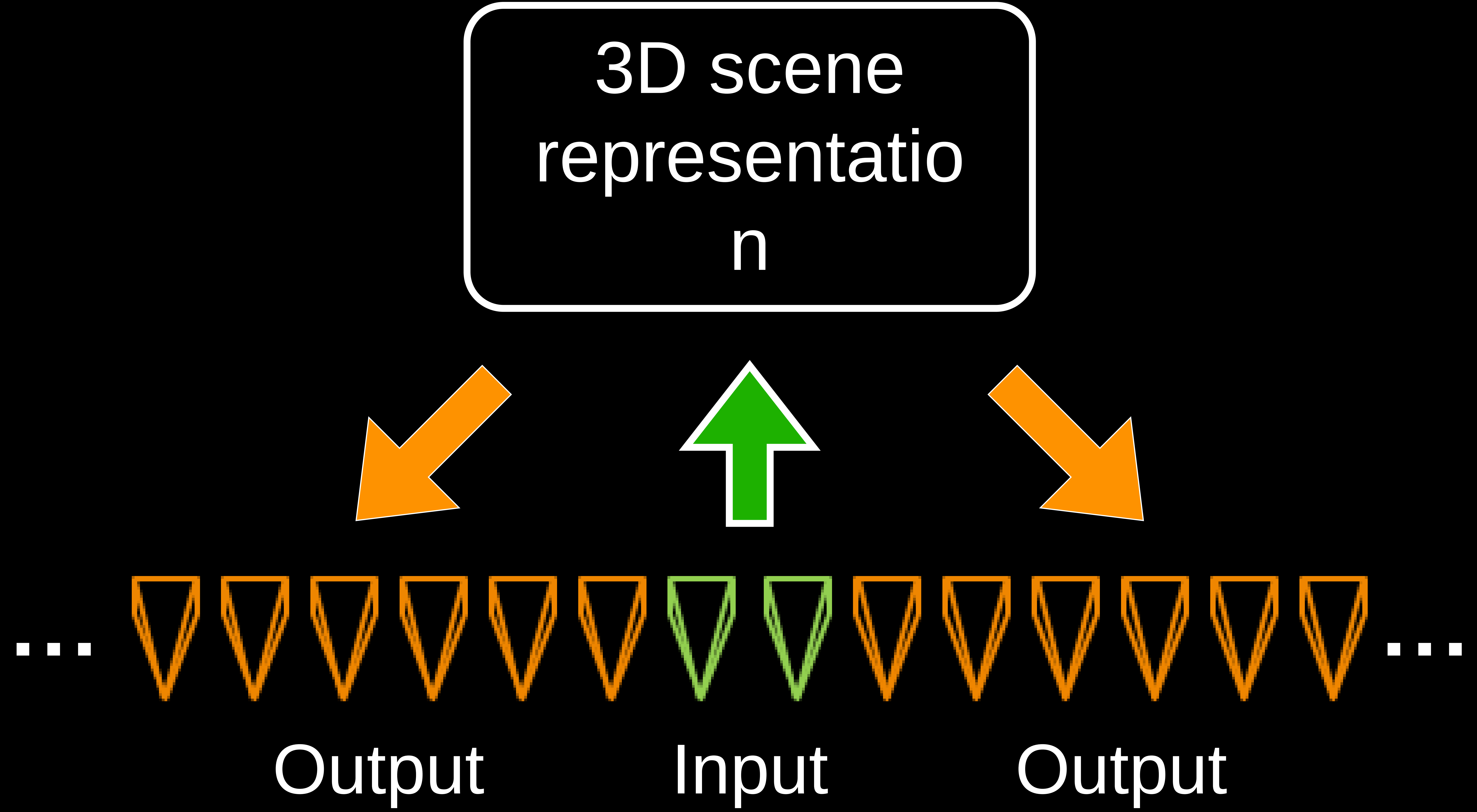
~1.5 cm

Left


Right
MAZZER

Output

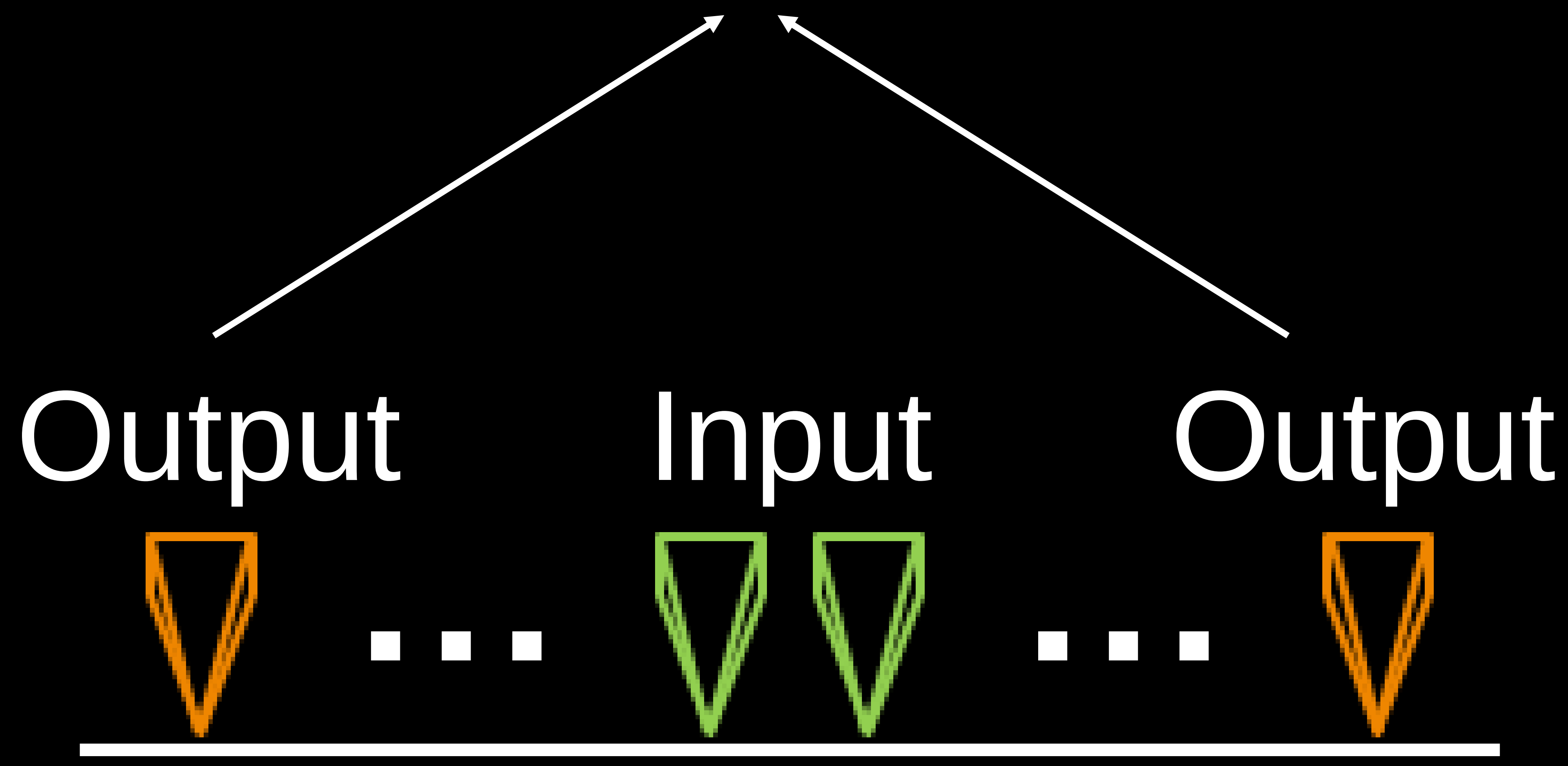
Problem Statement



Challenges

Extrapolation

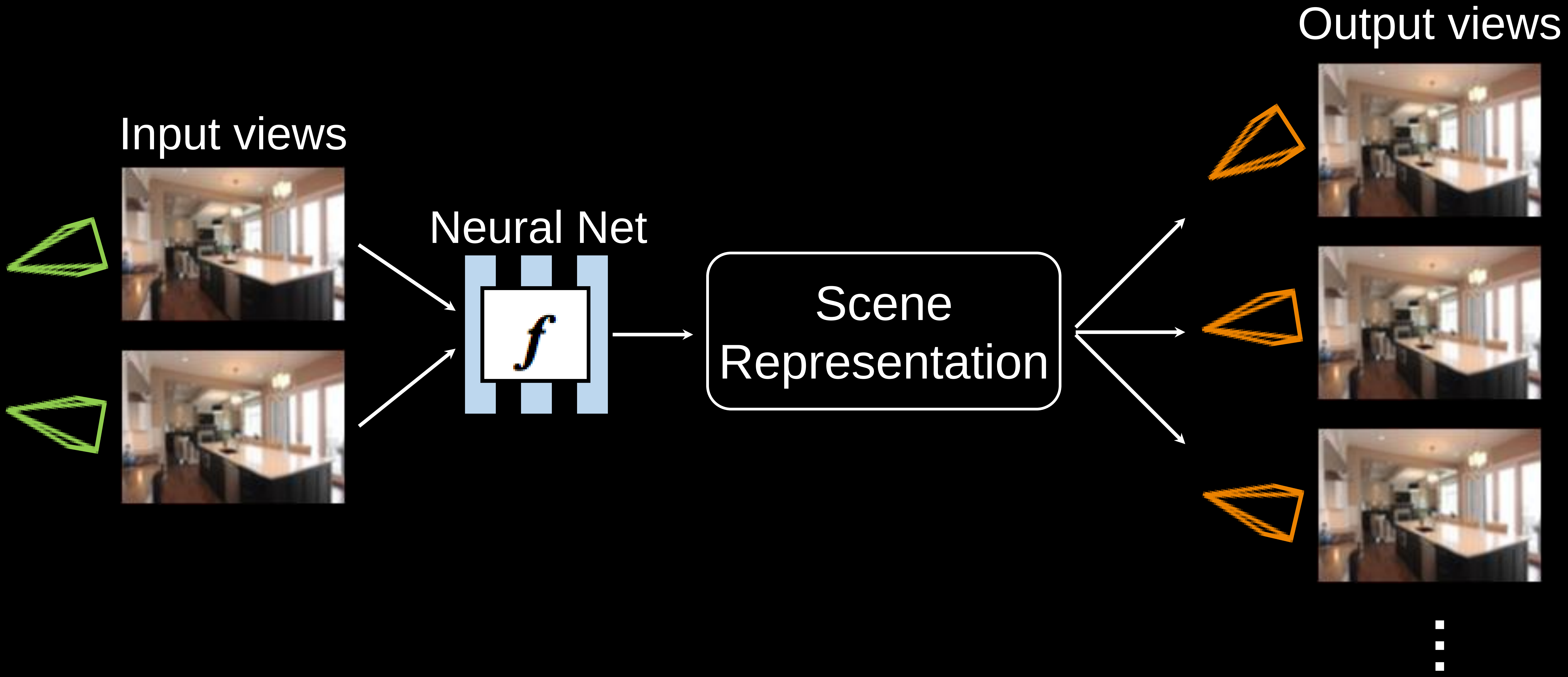
Large disocclusion



Non-Lambertian Effects

Reflections, transparencies, etc.

Neural prediction of scene representations



Stereo Magnification: Learning View Synthesis using Multiplane Images

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, Noah Snavely

SIGGRAPH 2018

Multiplane Camera (1937)

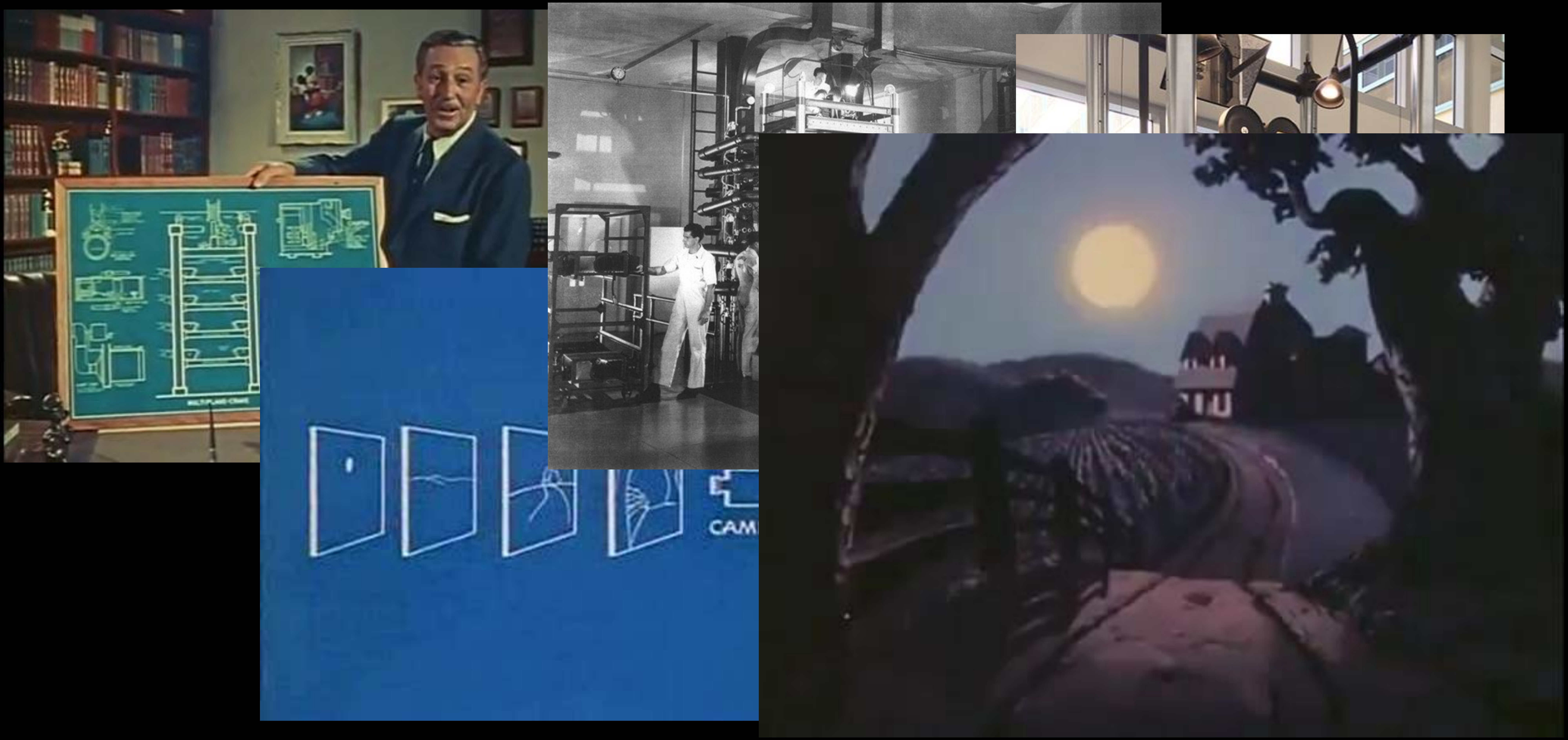
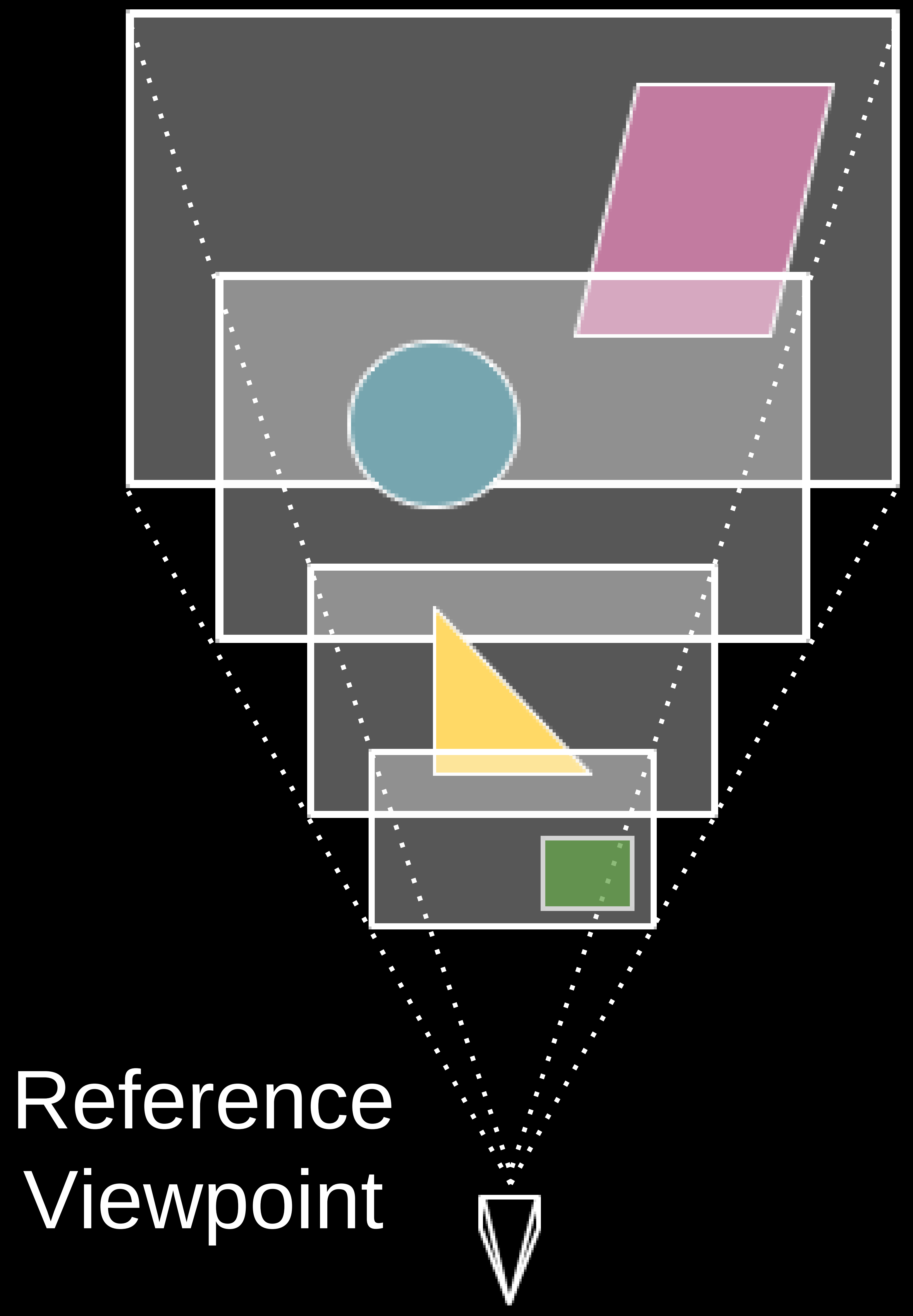


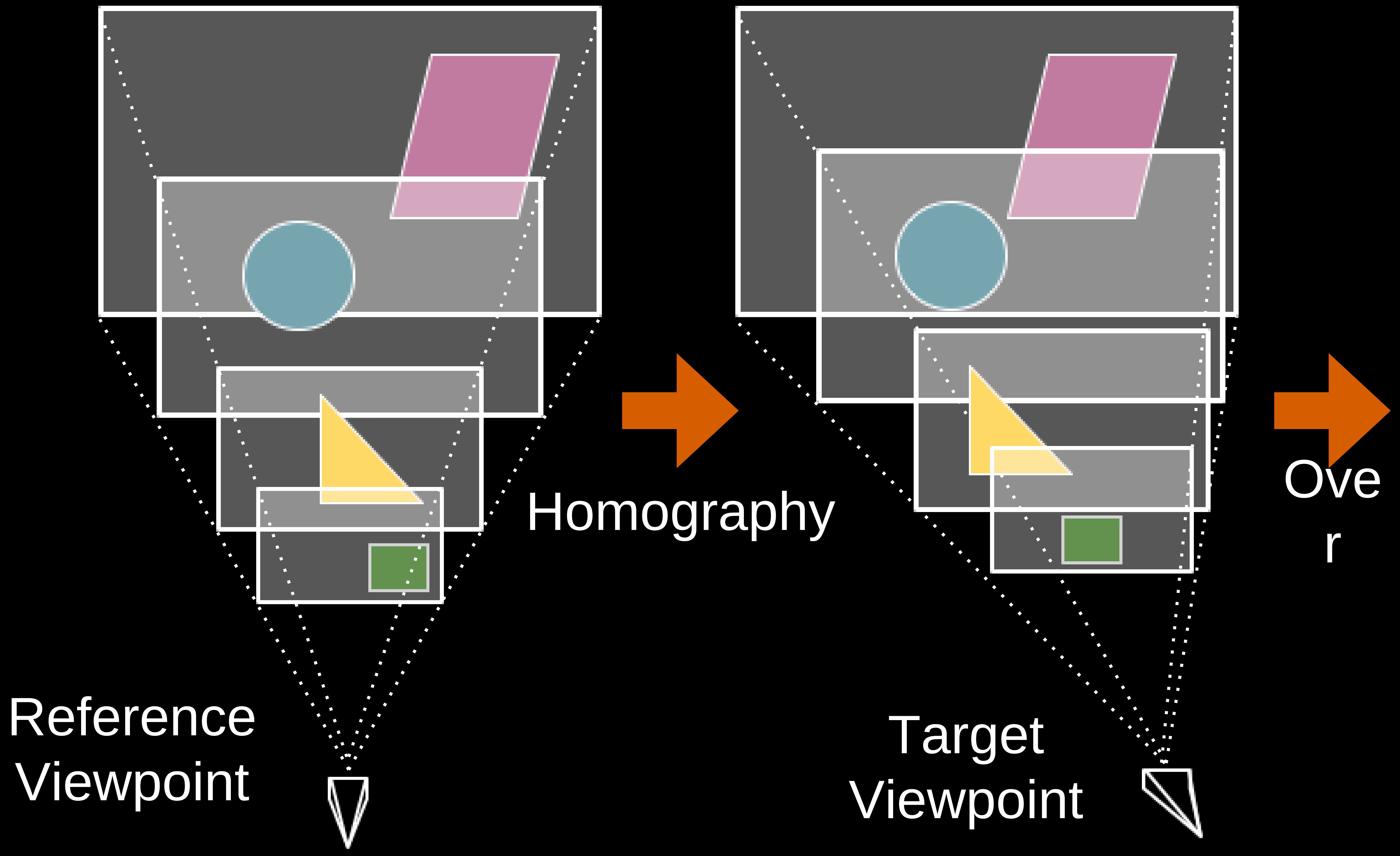
Image credits: Disney https://www.youtube.com/watch?v=kN-eCBAOw60 (from 1957)

Multiplane Images (MPIs)

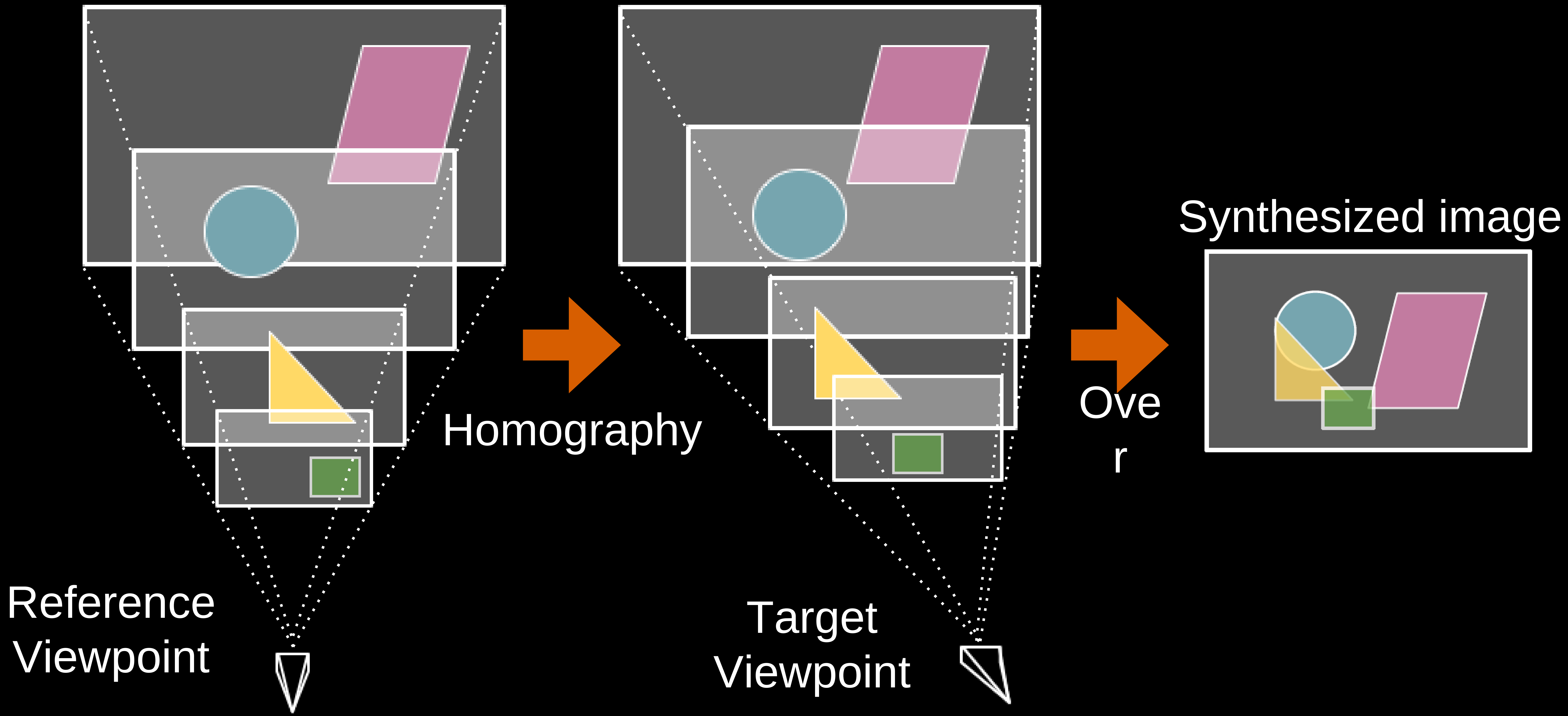


Each plane is at a fixed depth and encoded by an RGBA image

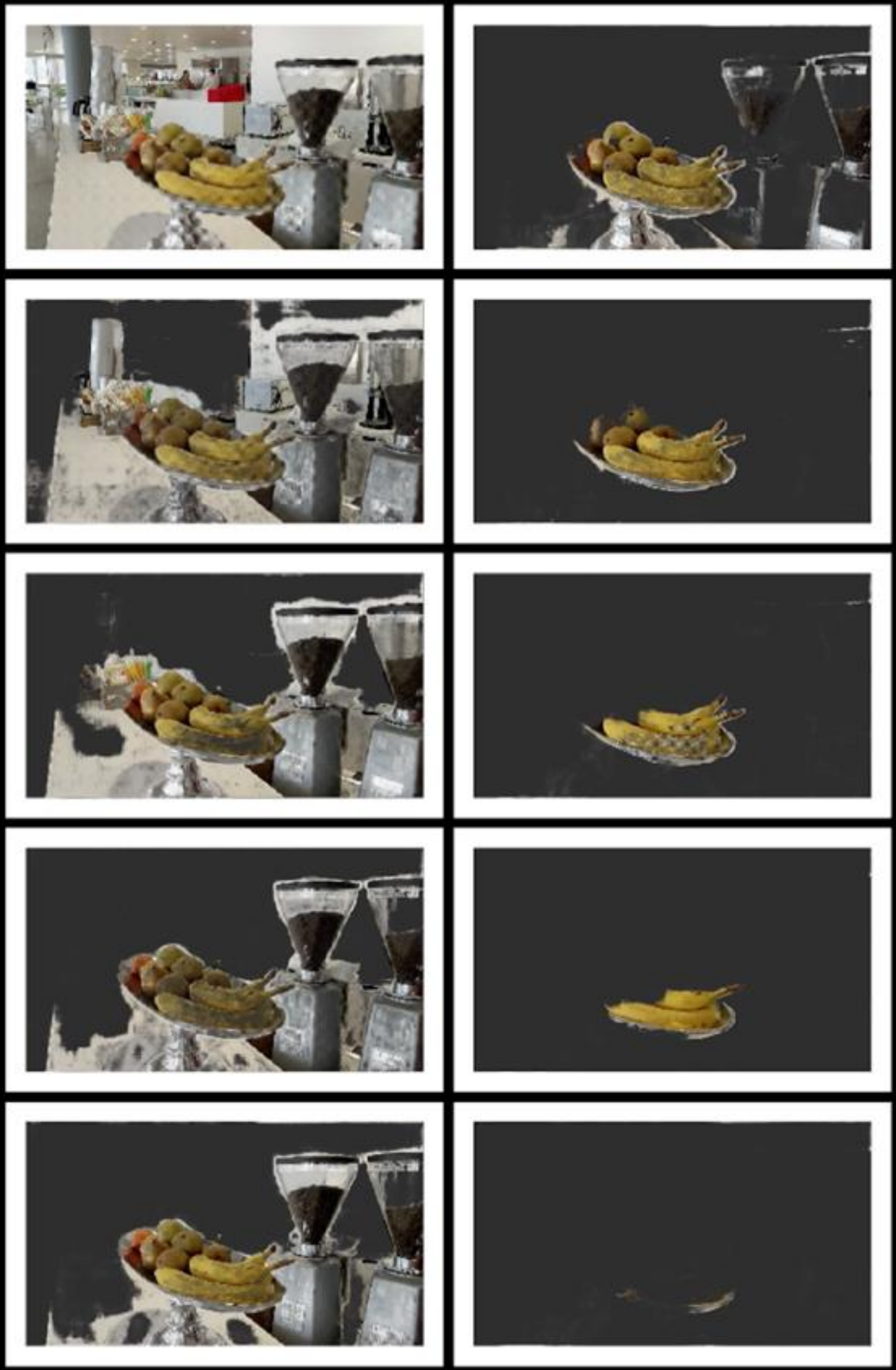
View Synthesis using Multiplane Images



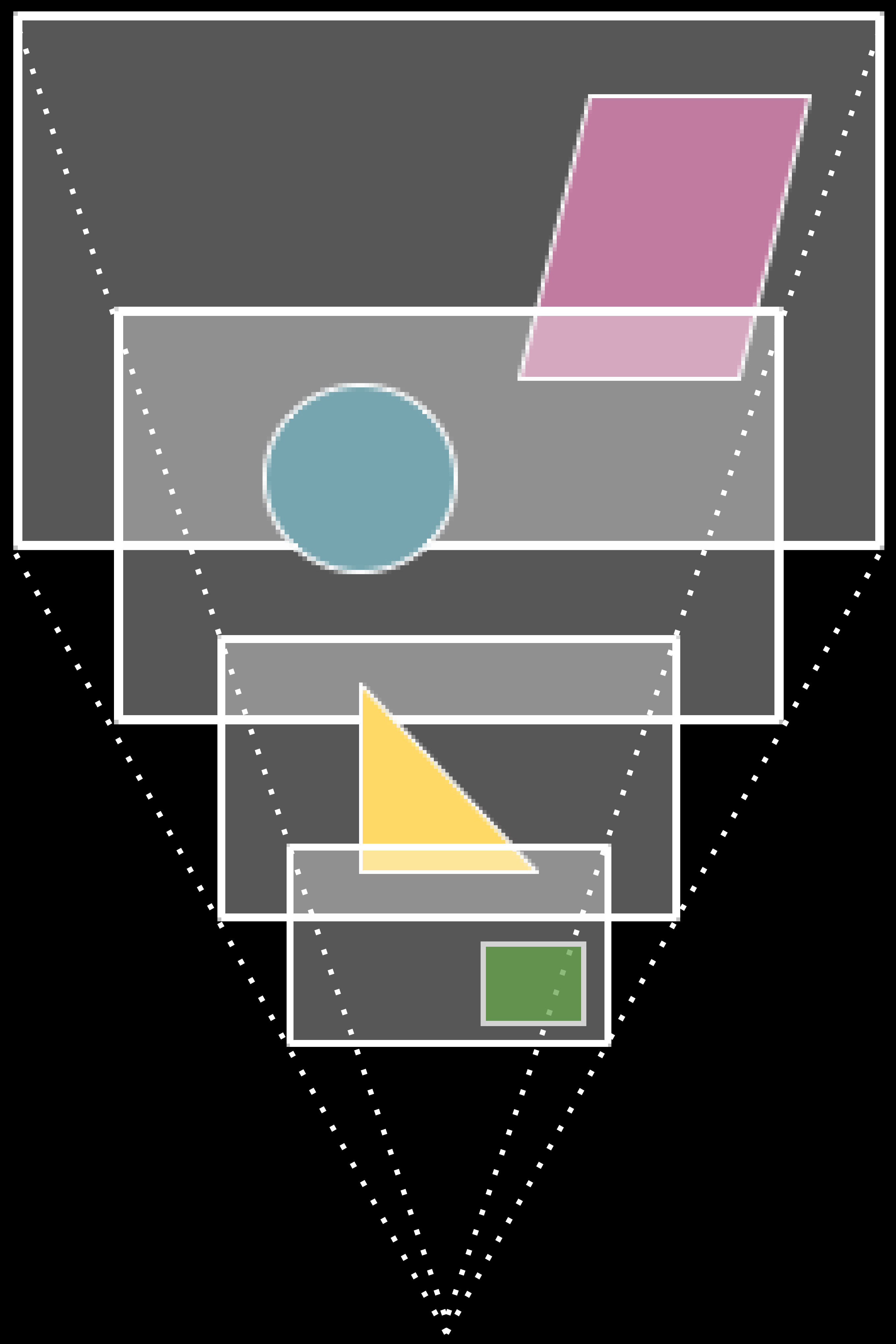
View Synthesis using Multiplane Images




MAZZER

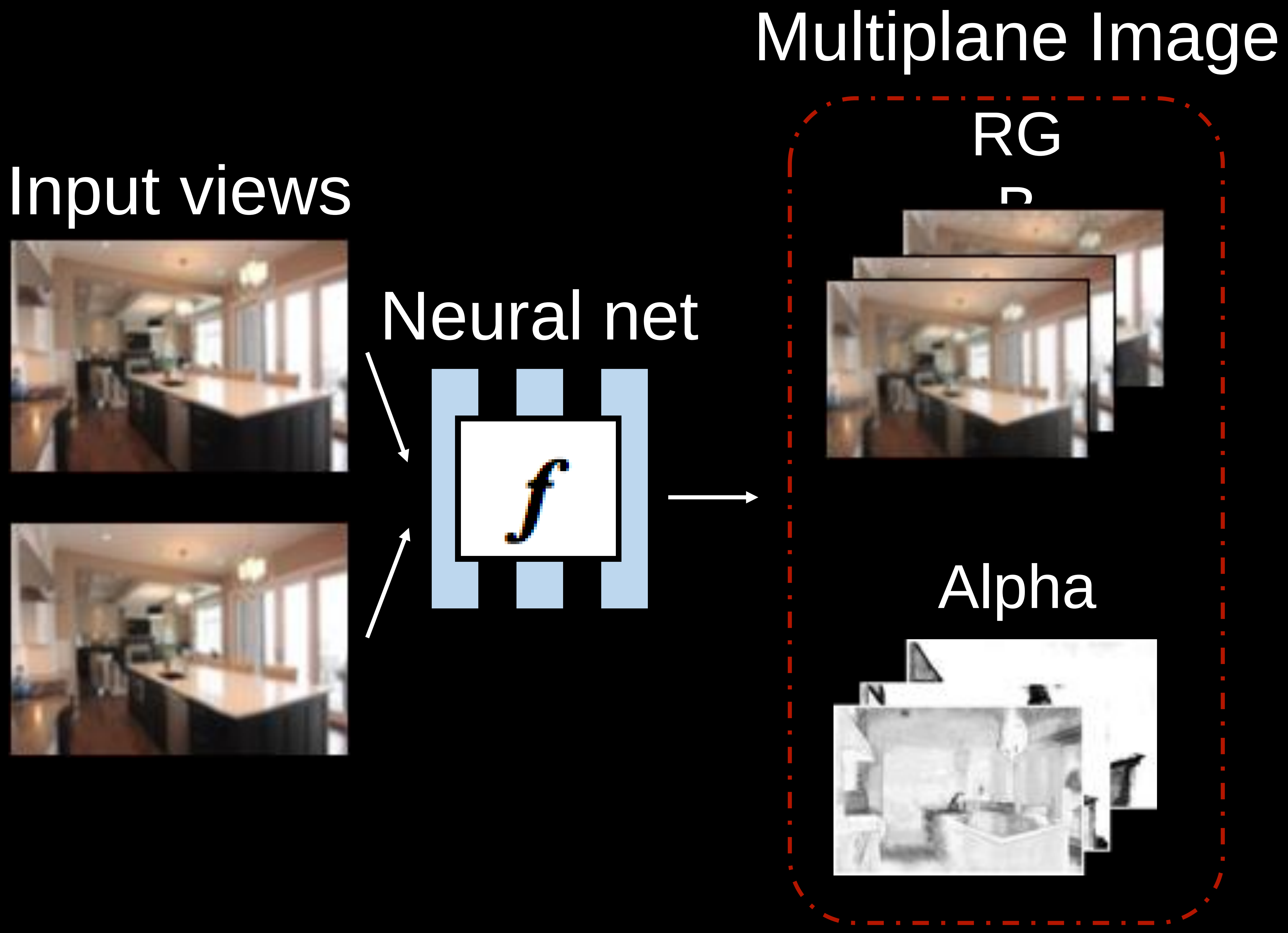


Properties of Multiplane Images

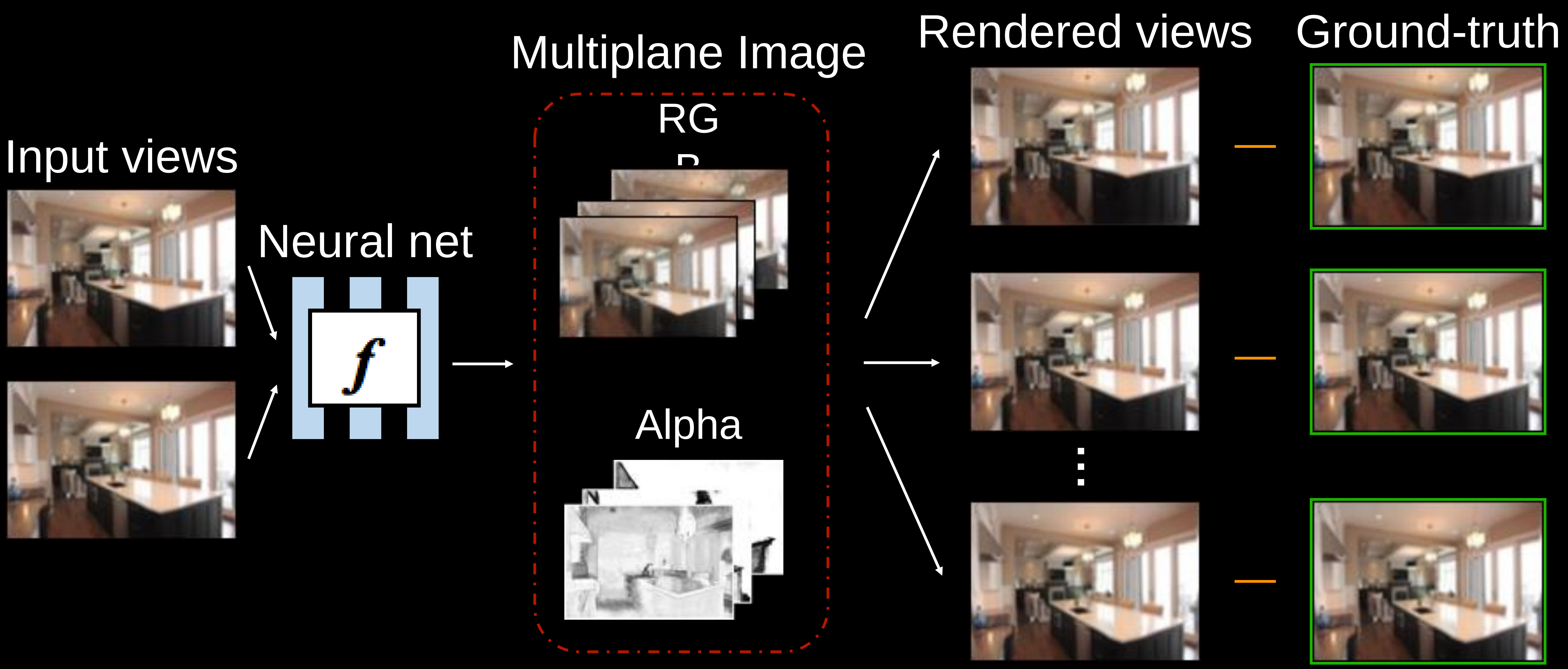


- Models disocclusion
- Models soft edges and non-Lambertian effects
- Efficient for view synthesis
- Differentiable rendering

Learning Multiplane Images



Learning Multiplane Images



Mapping image-shaped inputs to image-shaped outputs with the UNet architecture

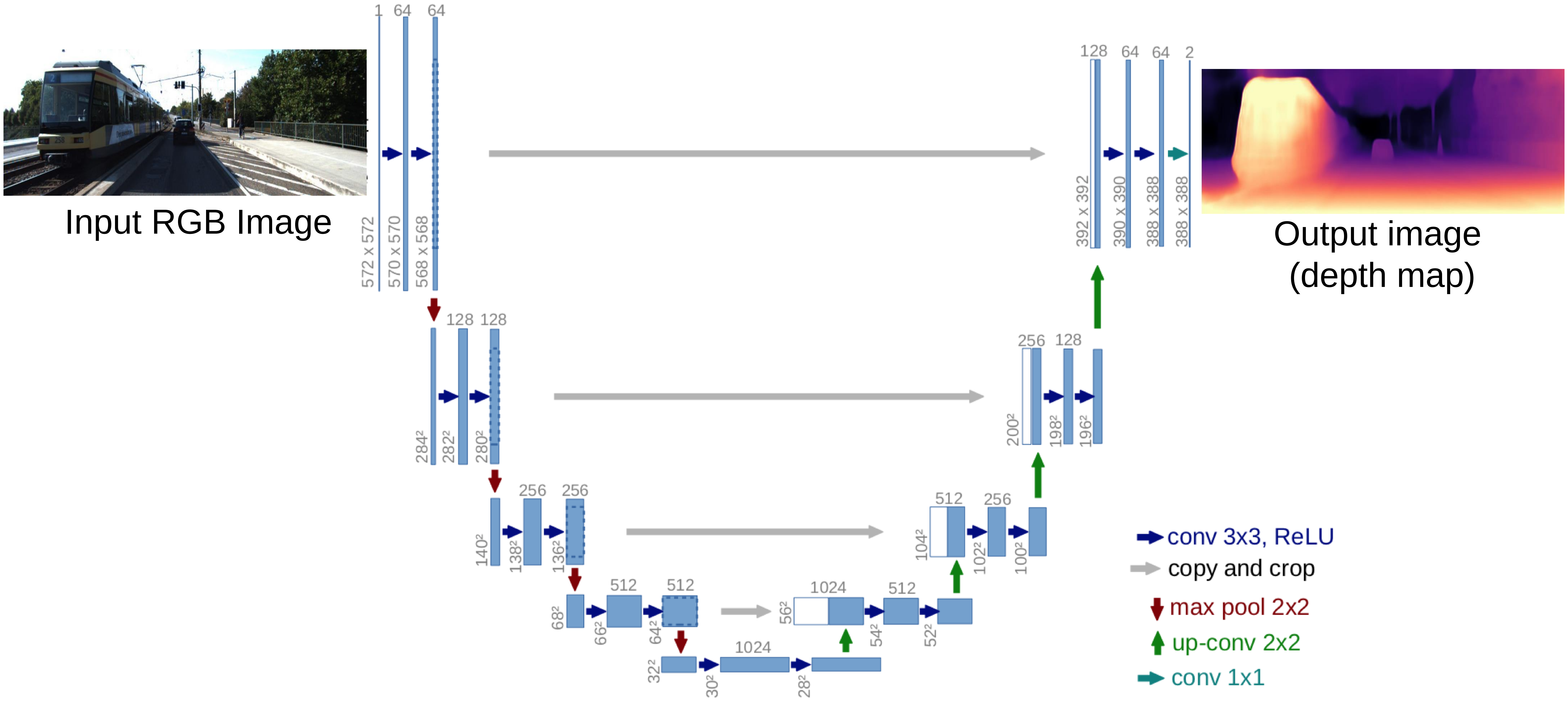
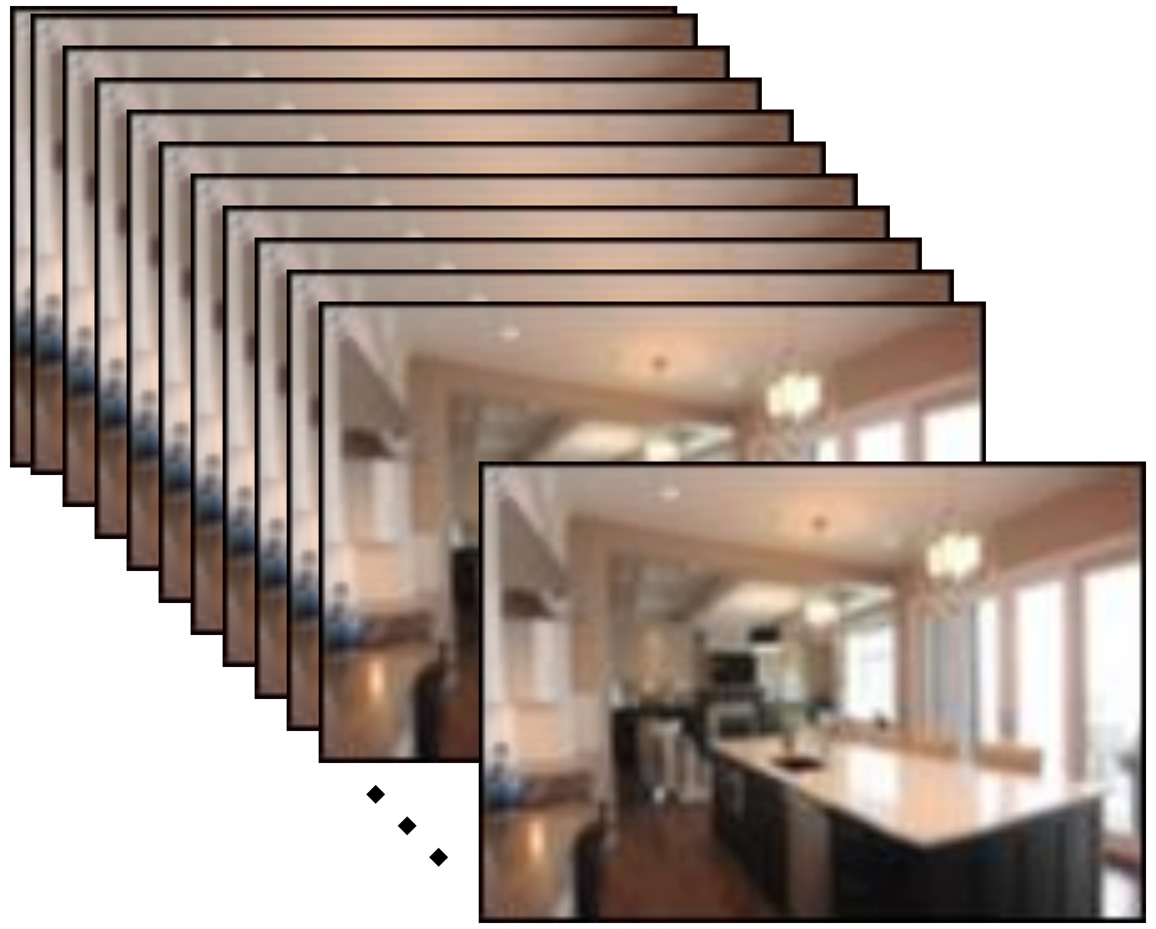


Image Pair → Multiplane Image

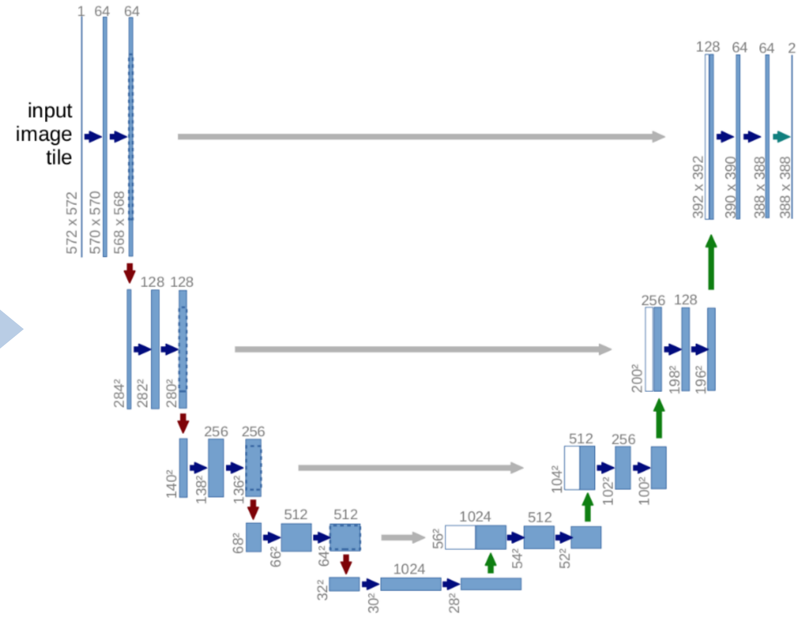
Suppose we want to map a pair of images to a 32-plane MPI

Input pair

Plane sweep volume



(32 RGB planes)



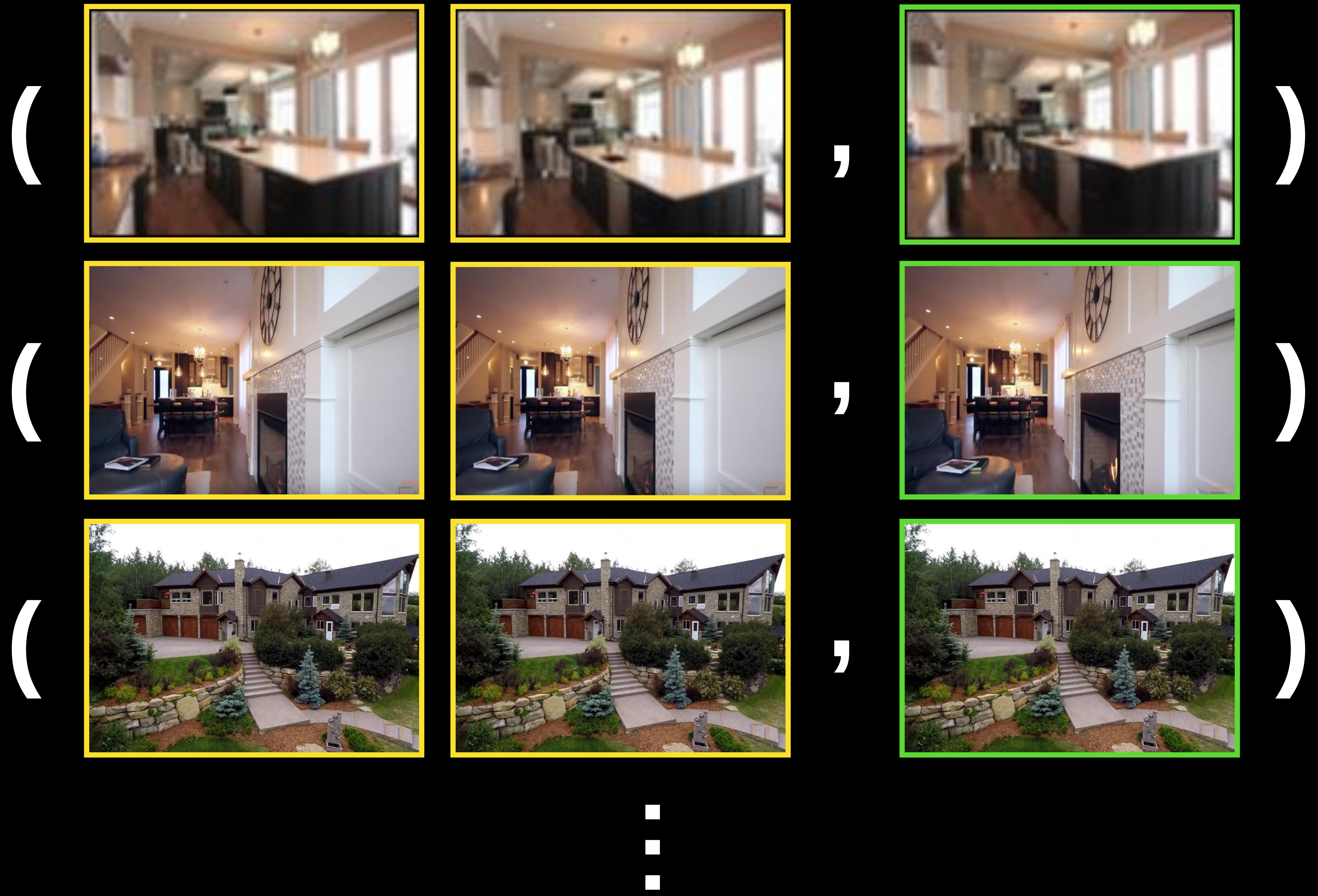
U-Net CNN

MPI Planes

(32 RGBA images)

Training Data

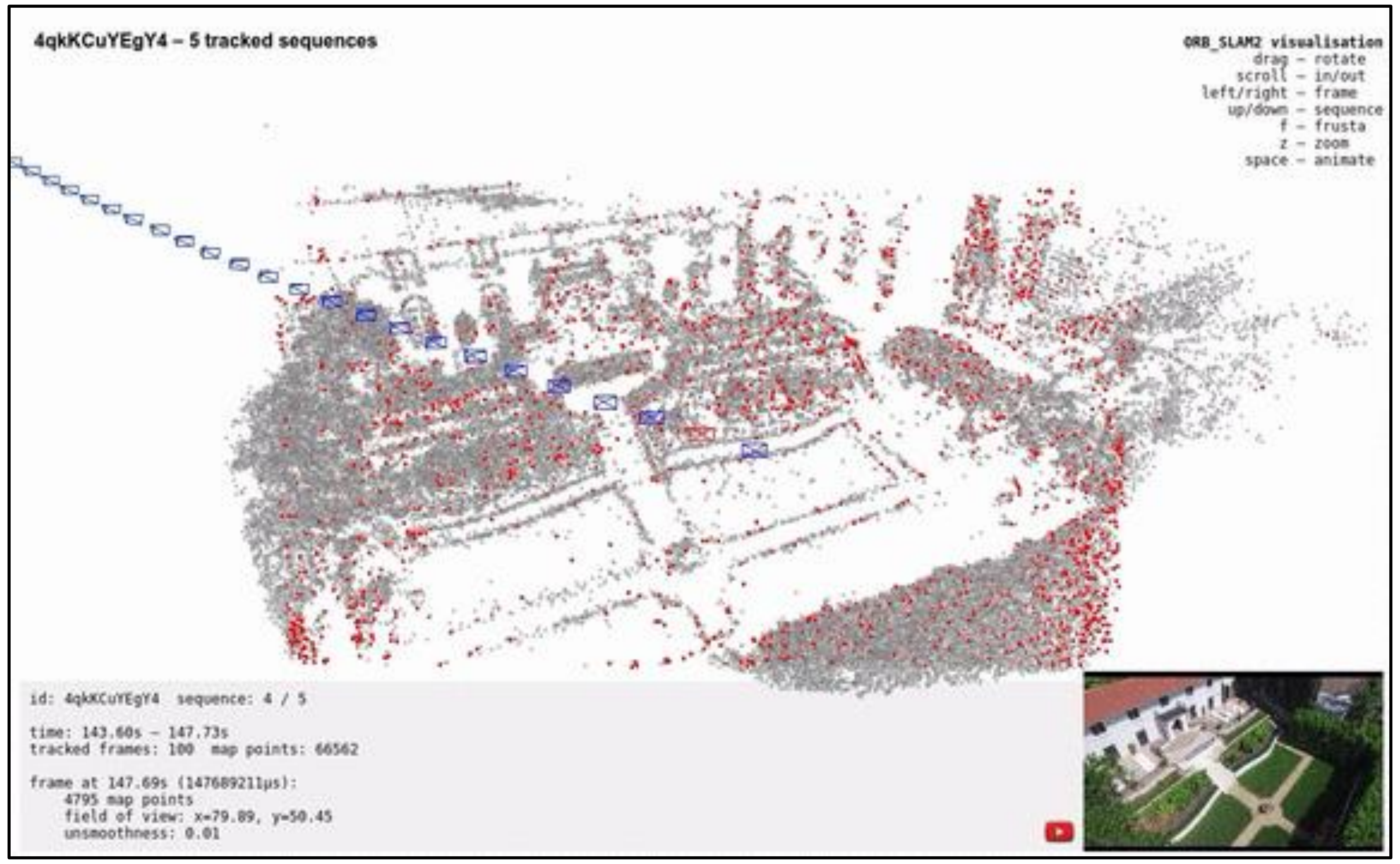
Input views   Target view



Need massive set of triplets with known camera poses

RealEstate10K

SLAM



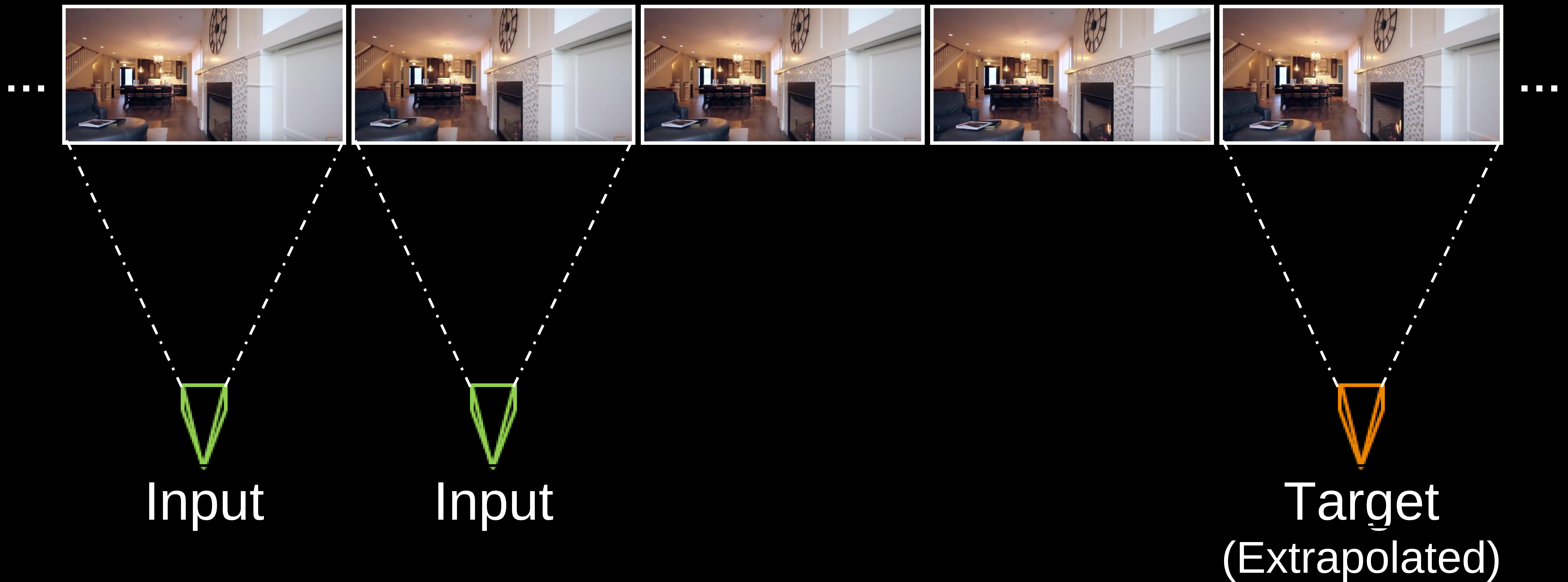
Running SLAM / SfM on YouTube videos at scale

RealEstate10K dataset

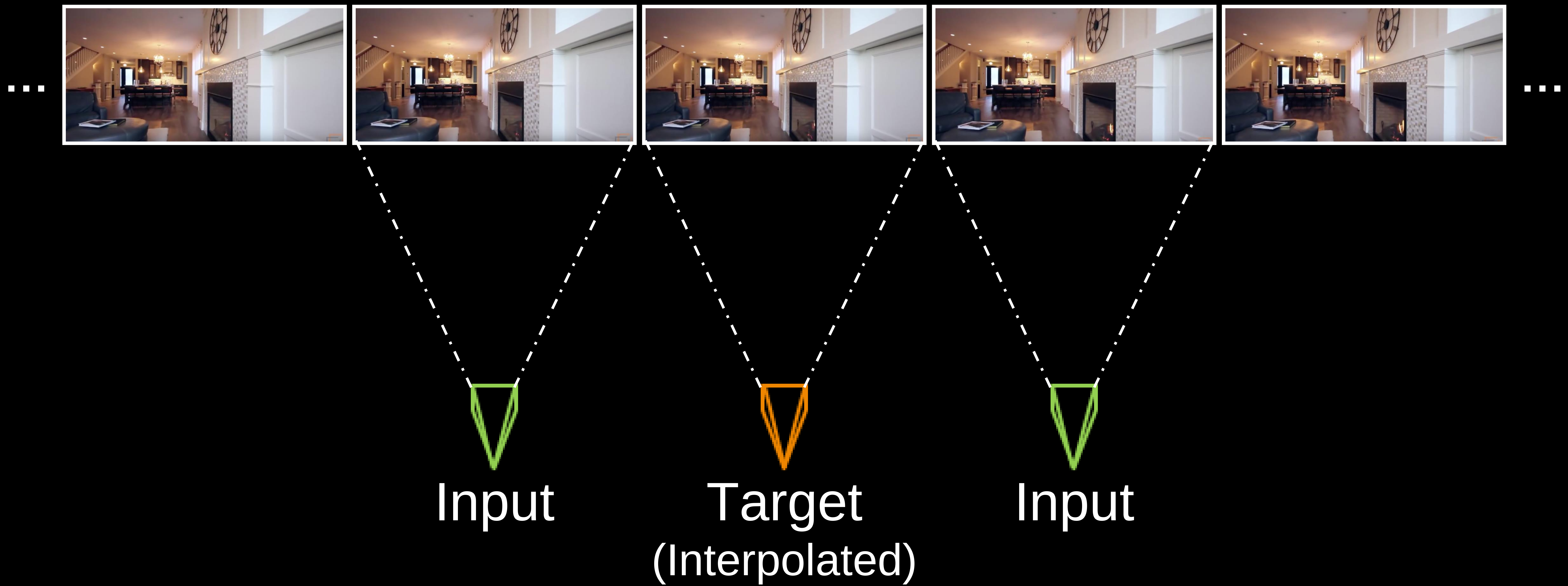
10 million frames from **80,000 video clips** from **10,000 videos**

https://google.github.io/realestate10k/

Sampling Training Examples



Sampling Training Examples



Results

Left

Right

Output

Image 1

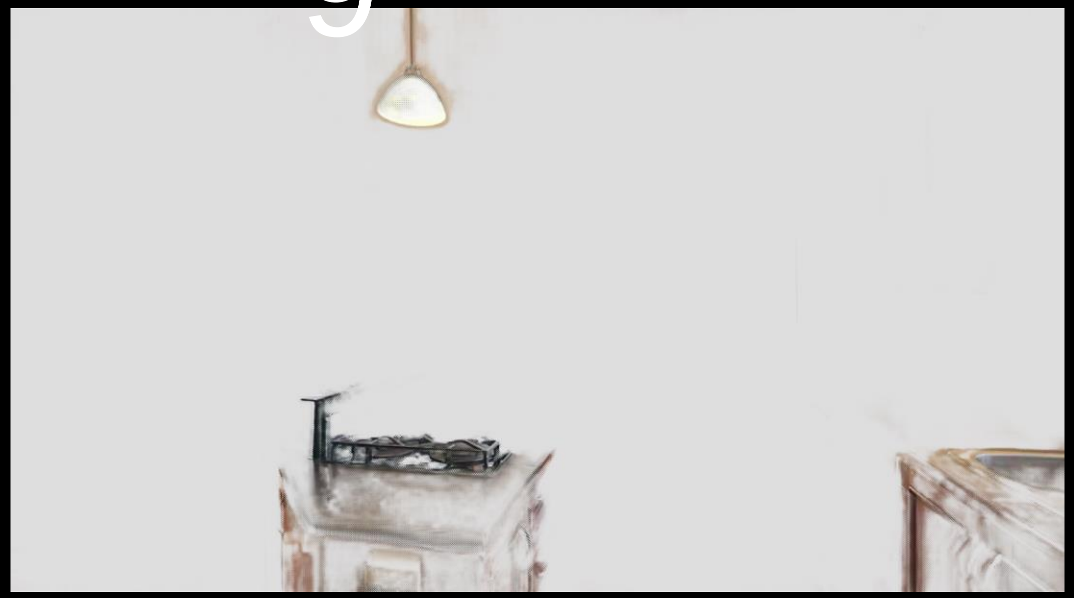
Image 2

Multi-plane Image (MPI)

Reference input view

Plane 0

Plane 9



Plane 13

Plane 16

Plane 24

Plane 26

Computer vision as inverse rendering

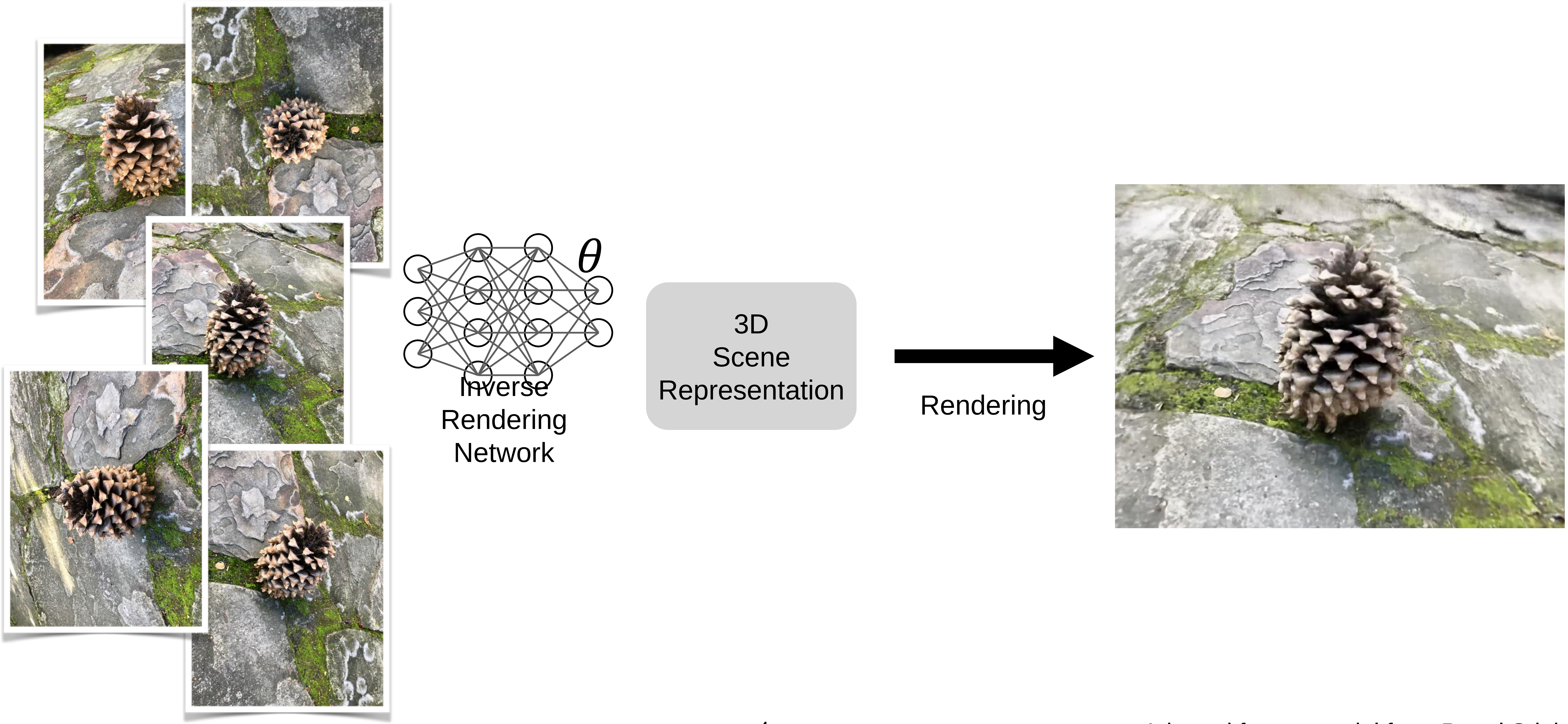
Inverse Rendering →

3D Scene Representation

→ Rendering

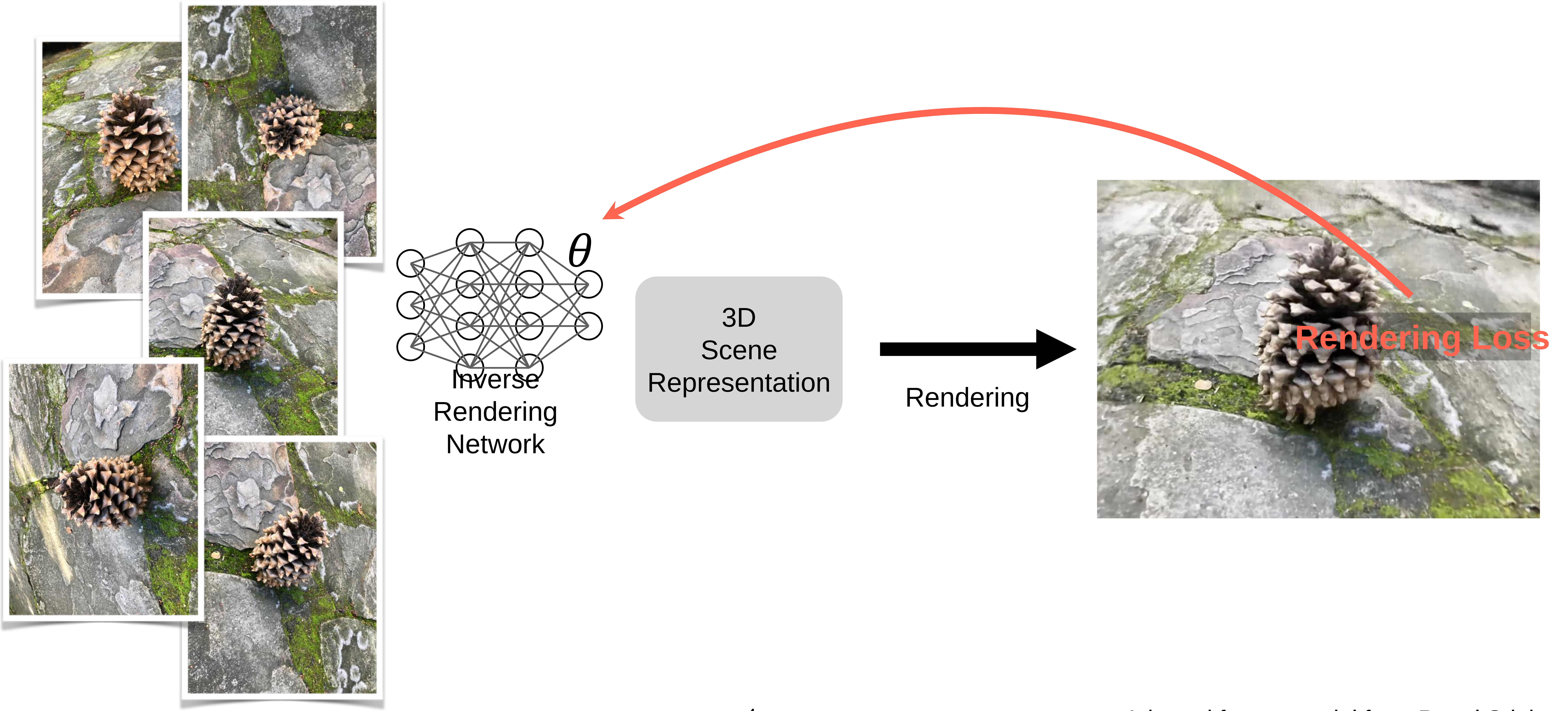
Adapted from material from Pratul Srinivas

Paradigm 1: “Feedforward” inverse rendering



Adapted from material from Pratul Srinivas

Paradigm 1: “Feedforward” inverse rendering



Adapted from material from Pratul Srinivas

Paradigm 2: “Render-and-compare”

Inverse Rendering

3D Scene Representation θ

Rendering

Adapted from material from Pratul Srinivas

Paradigm 2: “Render-and-compare”

Inverse Rendering

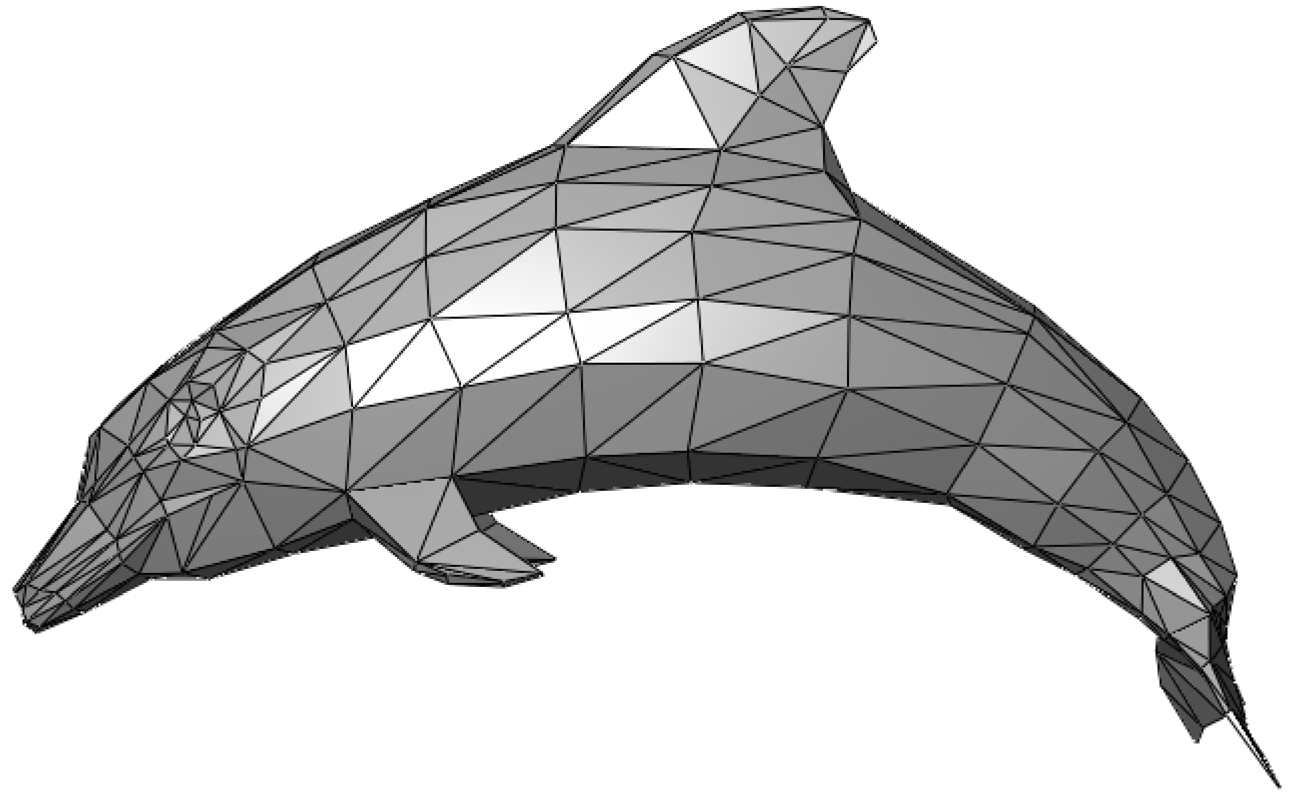
3D Scene Representation θ

Rendering

Adapted from material from Pratul Srinivas

What representation to use?

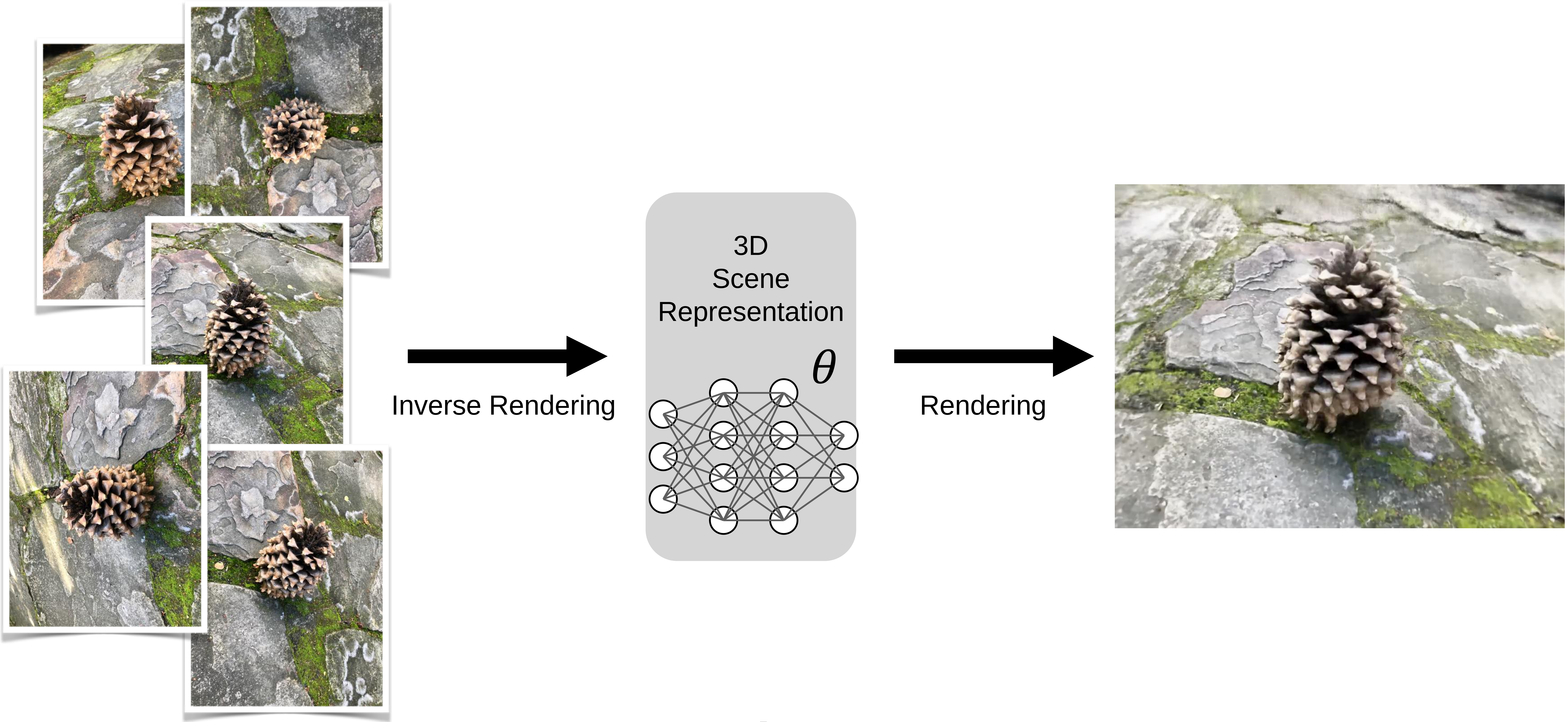
- Could use triangle meshes, but hard to differentiate during rendering



- Multiplane images (MPIs) are easy to differentiate, but only allow for rendering a small range of views

NeRF == Differentiable Rendering with a **Neural Volumetric Representation**

Paradigm 2: “Render-and-compare”



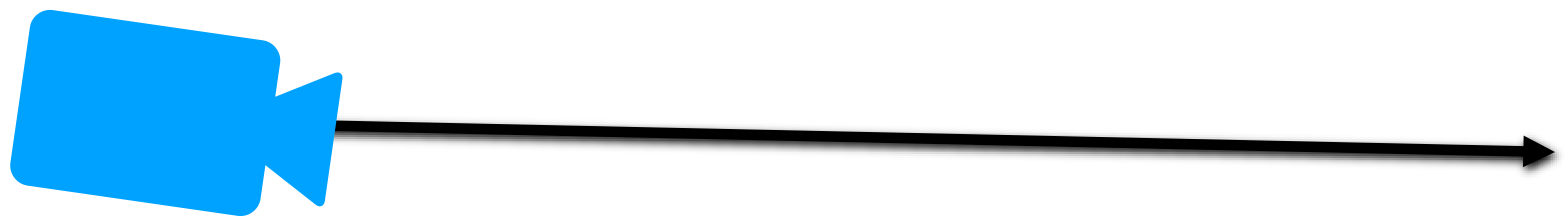
Adapted from material from Pratul Srinivas

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

Neural Volumetric Rendering

Neural Volumetric Rendering

querying the radiance value
along rays through 3D space



What color?

Adapted from material from Pratul Srinivas

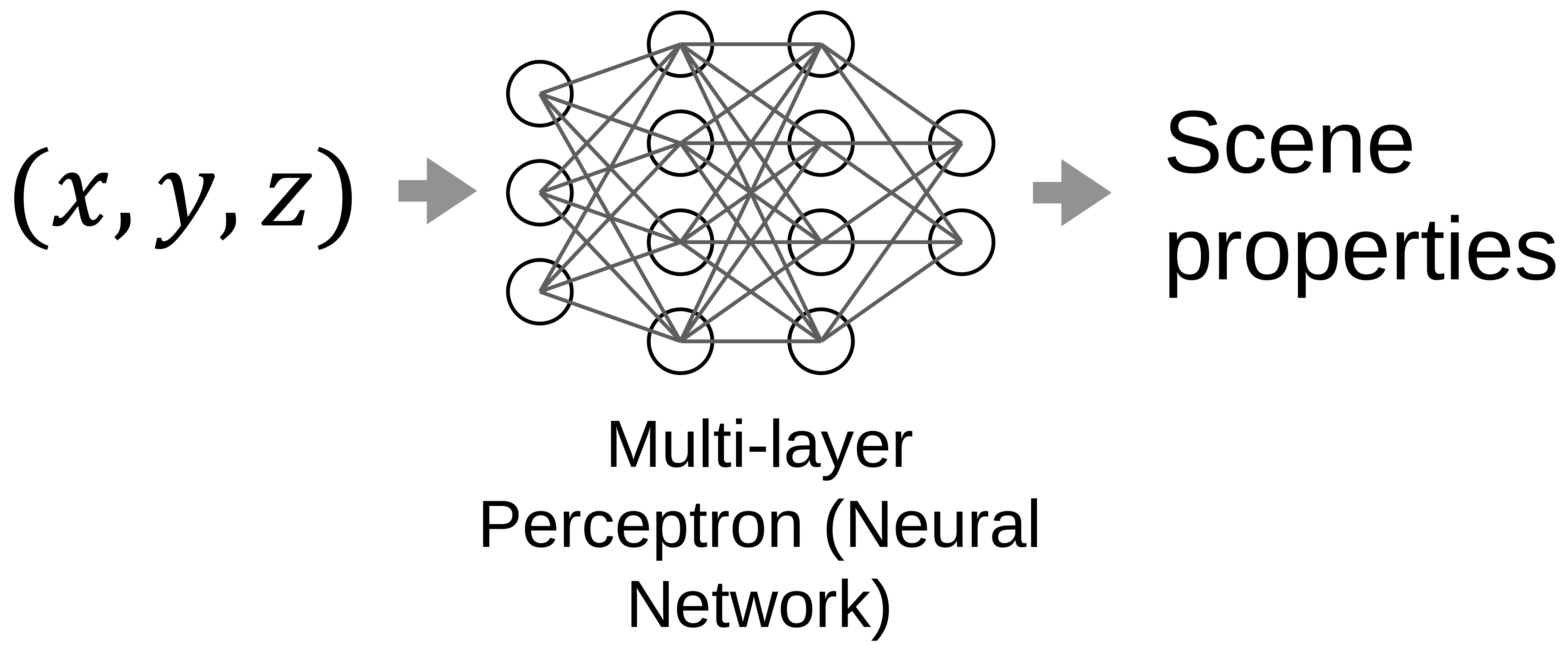
Neural Volumetric Rendering

continuous, differentiable
rendering model without concrete
ray/surface intersections

Adapted from material from Pratul Srinivas

Neural Volumetric Rendering

using a neural network as a scene representation, rather than a voxel grid of data



Adapted from material from Pratul Srinivas

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
ECCV 2020

Ben Mildenhall*

UC Berkeley

Pratul Srinivasan*

UC Berkeley

Matt Tancik*

UC Berkeley

Jon Barron

Google Research

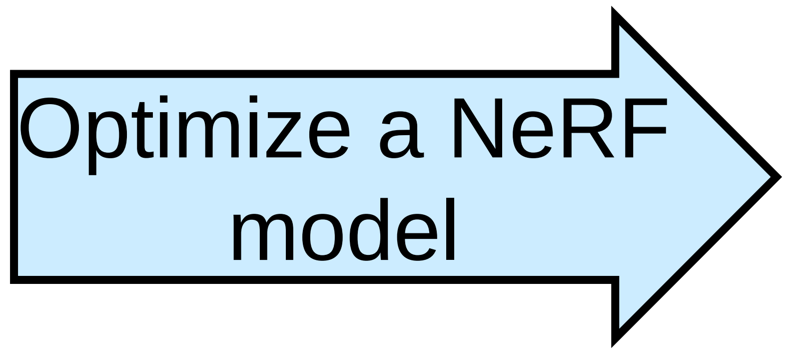
Ravi Ramamoorthi

UC San Diego

Ren Ng

UC Berkeley

Given a set of sparse views of an object with known camera poses



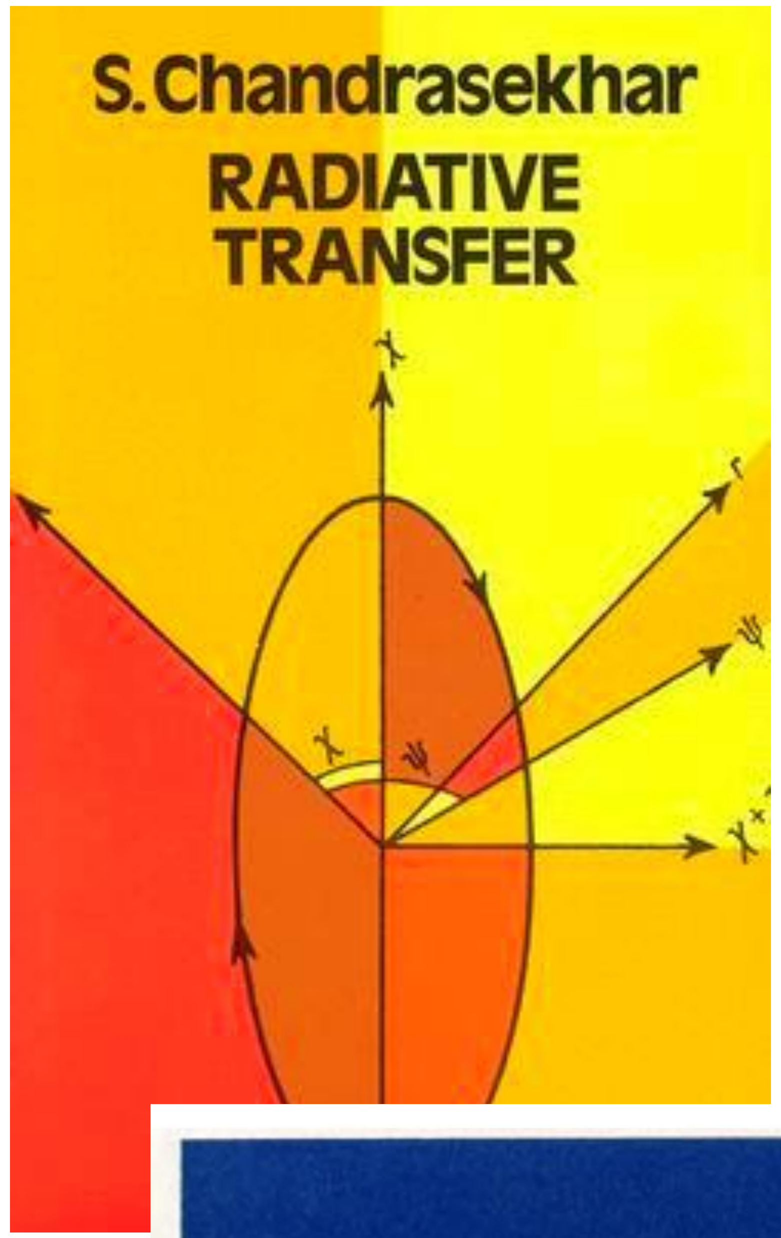
3D reconstruction viewable from any angle

NeRF Overview

- Volumetric rendering
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)

NeRF Overview

- Volumetric rendering
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)



Traditional volumetric rendering

Theory of volume rendering co-opted from physics in the 1980s: absorption, emission, out-scattering/in-scattering

Ray tracing simulated cumulus cloud [Kajiya]

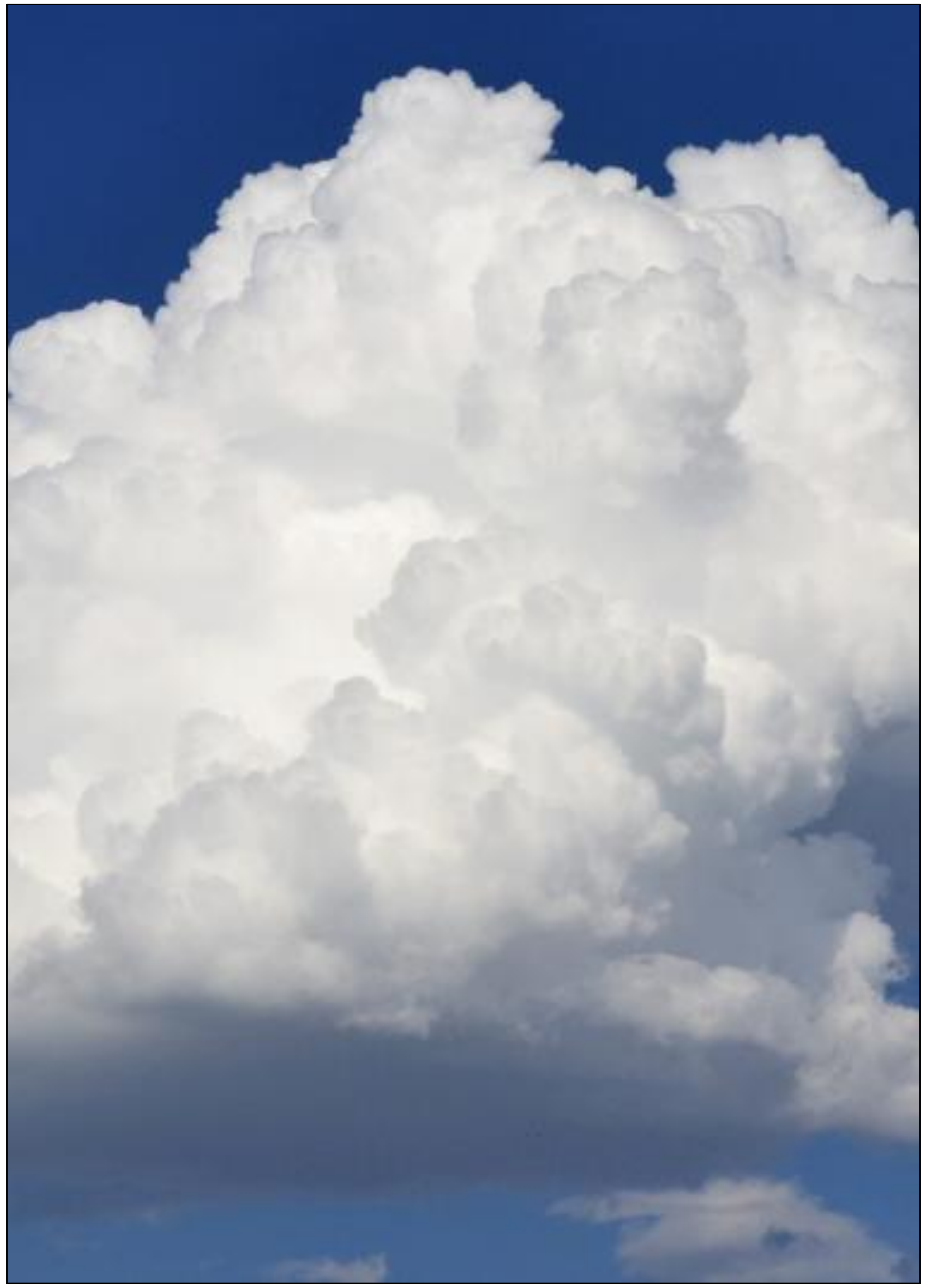
Chandrasekhar 1950, Radiative Transfer

Kajiya 1984, Ray Tracing Volume Densities

Full volumetric rendering formulation

Absorption **Scattering** **Emission**

http://commons.wikimedia.org



http://coclouds.com

http://wikipedia.org

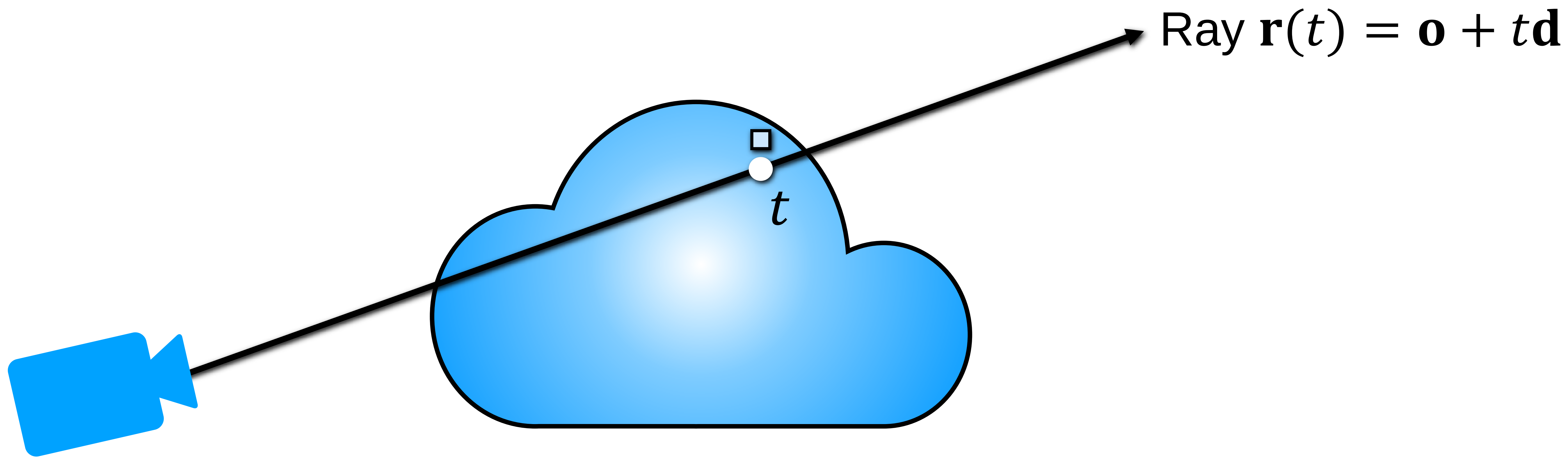
Slide credit: Novak et al 2018, Monte Carlo methods for physically based volume rendering

Volumetric formulation for NeRF

Scene is a cloud of colored fog

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

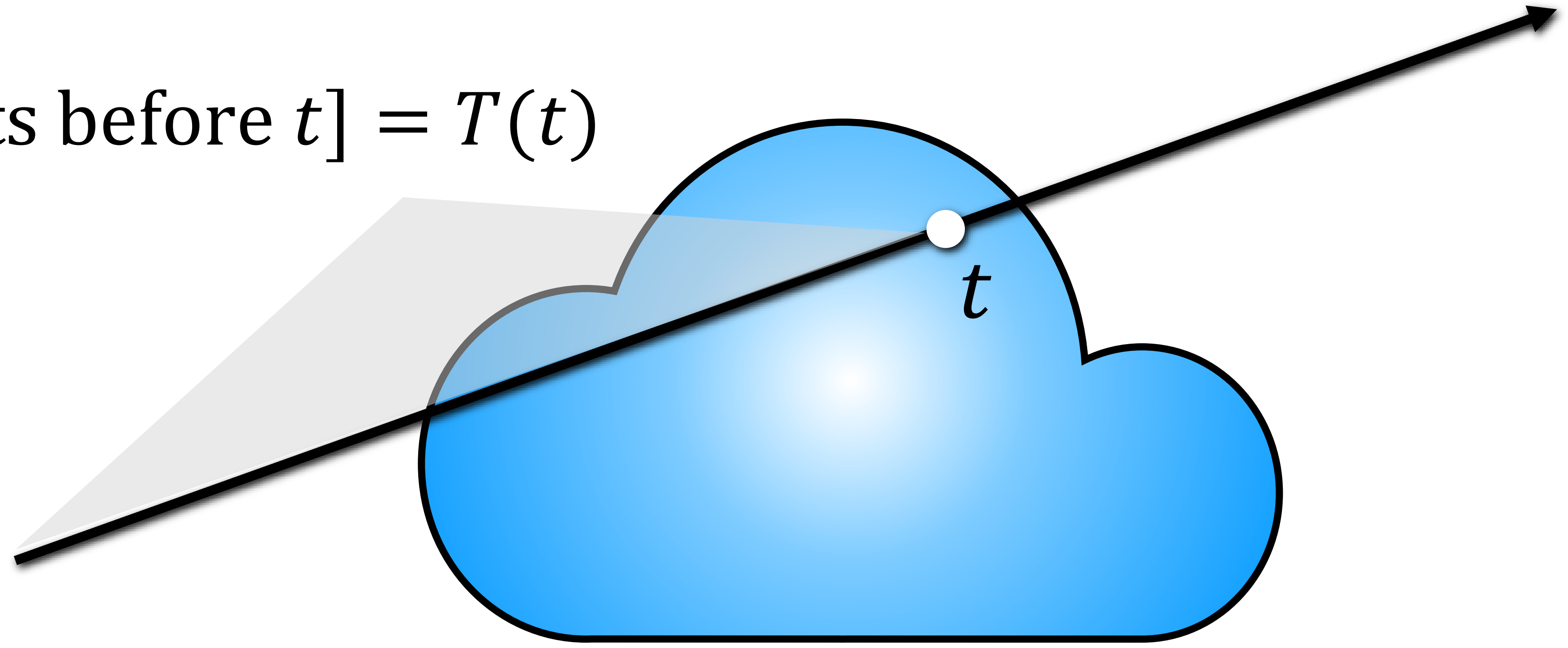
Volumetric formulation for NeRF



Consider a ray traveling through the scene, and a point at distance t along this ray. We look up its color $\mathbf{c}(t)$, and its opacity (alpha value) $\alpha(t)$ from a neural network

Volumetric formulation for NeRF

$P[\text{no hits before } t] = T(t)$



But t may also be blocked by earlier points along the ray. $T(t)$: probability that the ray didn't hit any particles earlier.

$T(t)$ is called "transmittance"

Adapted from material from Pratul Srinivas

Volume rendering estimation: integrating color along a ray

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

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

final rendered color along ray

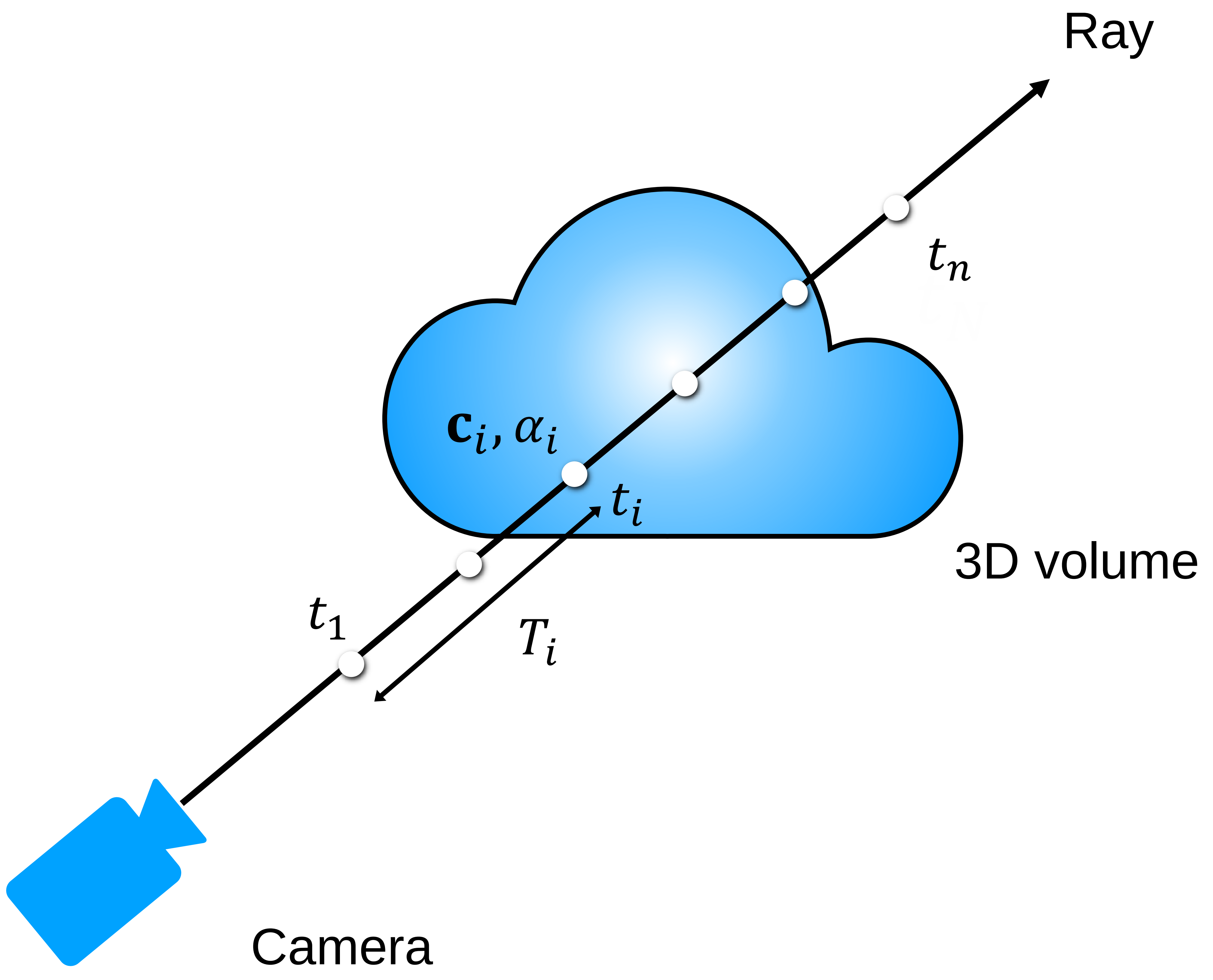
weights

colors

How much light is blocked earlier along ray:

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

Computing the color for a set of rays through the pixels of an image yields a rendered image



Adapted from material from Pratul Srinivas

Volume rendering estimation: integrating color along a ray

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

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

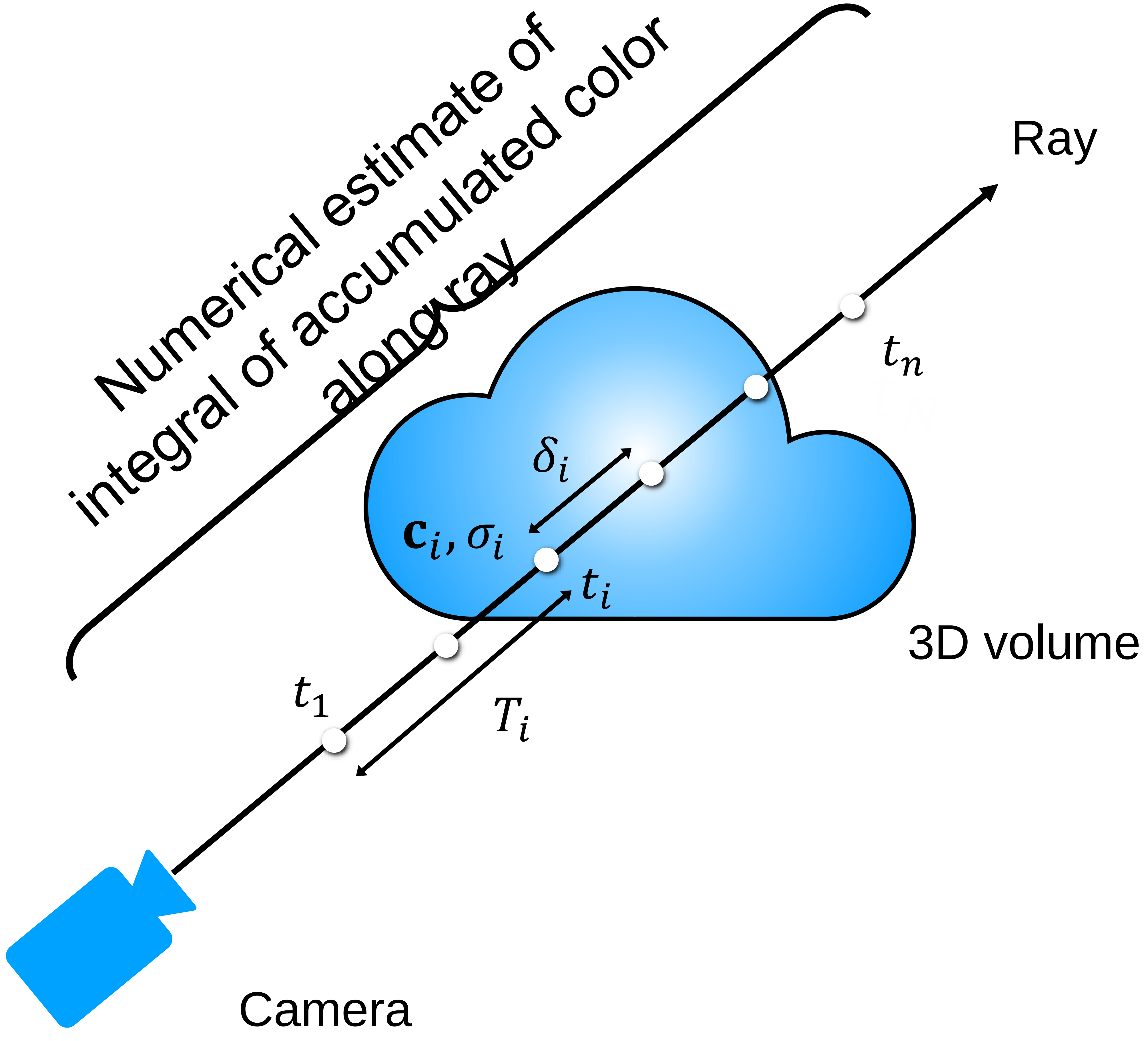
final rendered color along ray

weights

colors

How much light is blocked earlier along ray:

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



Slight modification: α is not directly stored in the volume, but instead is derived from a stored volume density sigma (σ) that is multiplied by the distance between samples delta (δ):

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

Adapted from material from Pratul Srinivas

Volume rendering estimation: integrating color along a ray

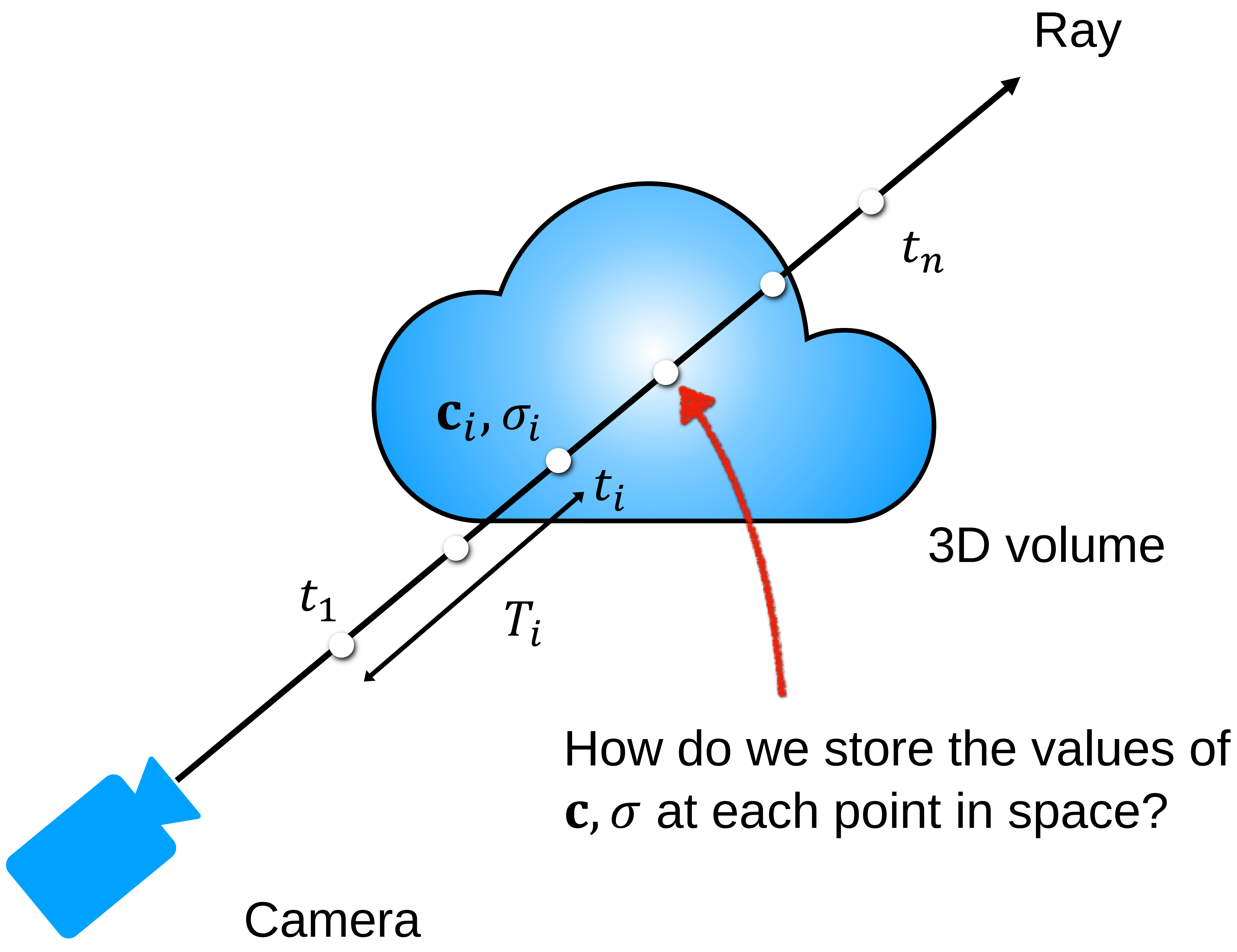
Rendering model for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

$$\underbrace{\mathbf{c}}_{\text{final rendered color along ray}} \approx \sum_{i=1}^{n} \underbrace{T_i \alpha_i}_{\text{weights}} \underbrace{\mathbf{c}_i}_{\text{colors}}$$

How much light is blocked earlier along ray:

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

Computing the color for a set of rays through the pixels of an image yields a rendered image



How do we store the values of $\mathbf{c}, \sigma$ at each point in space?

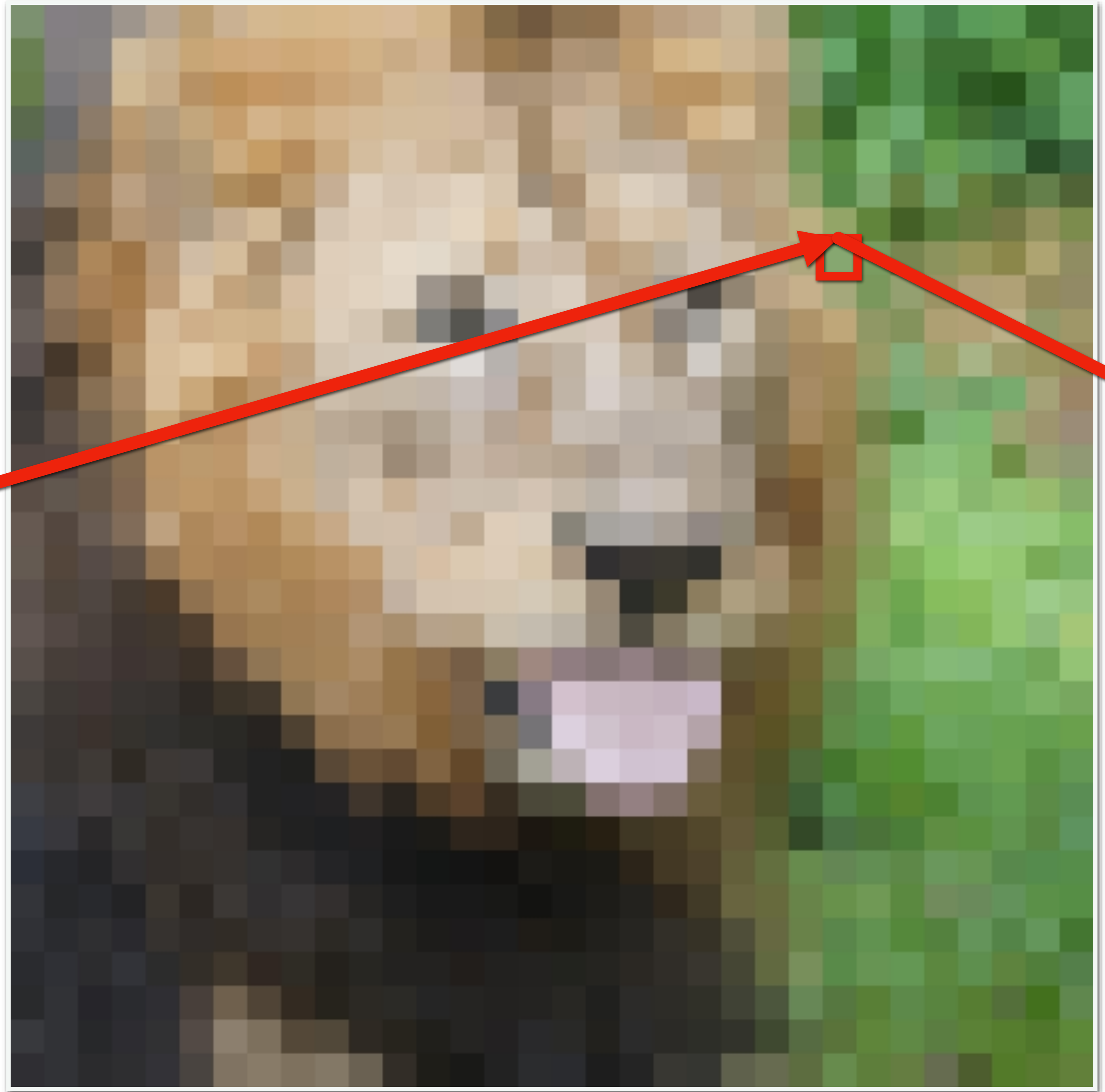
Adapted from material from Pratul Srinivas

NeRF Overview

- Volumetric rendering
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)

Toy problem: storing 2D image data

$$(x, y)$$

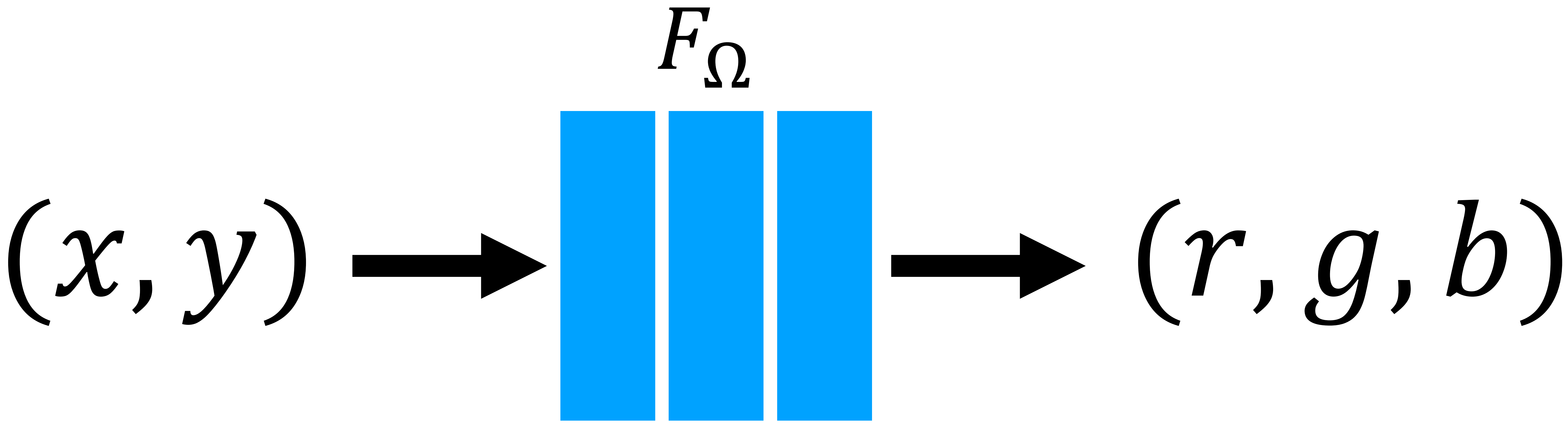


$$(r, g, b)$$

Usually we store an image as a
2D grid of RGB color values

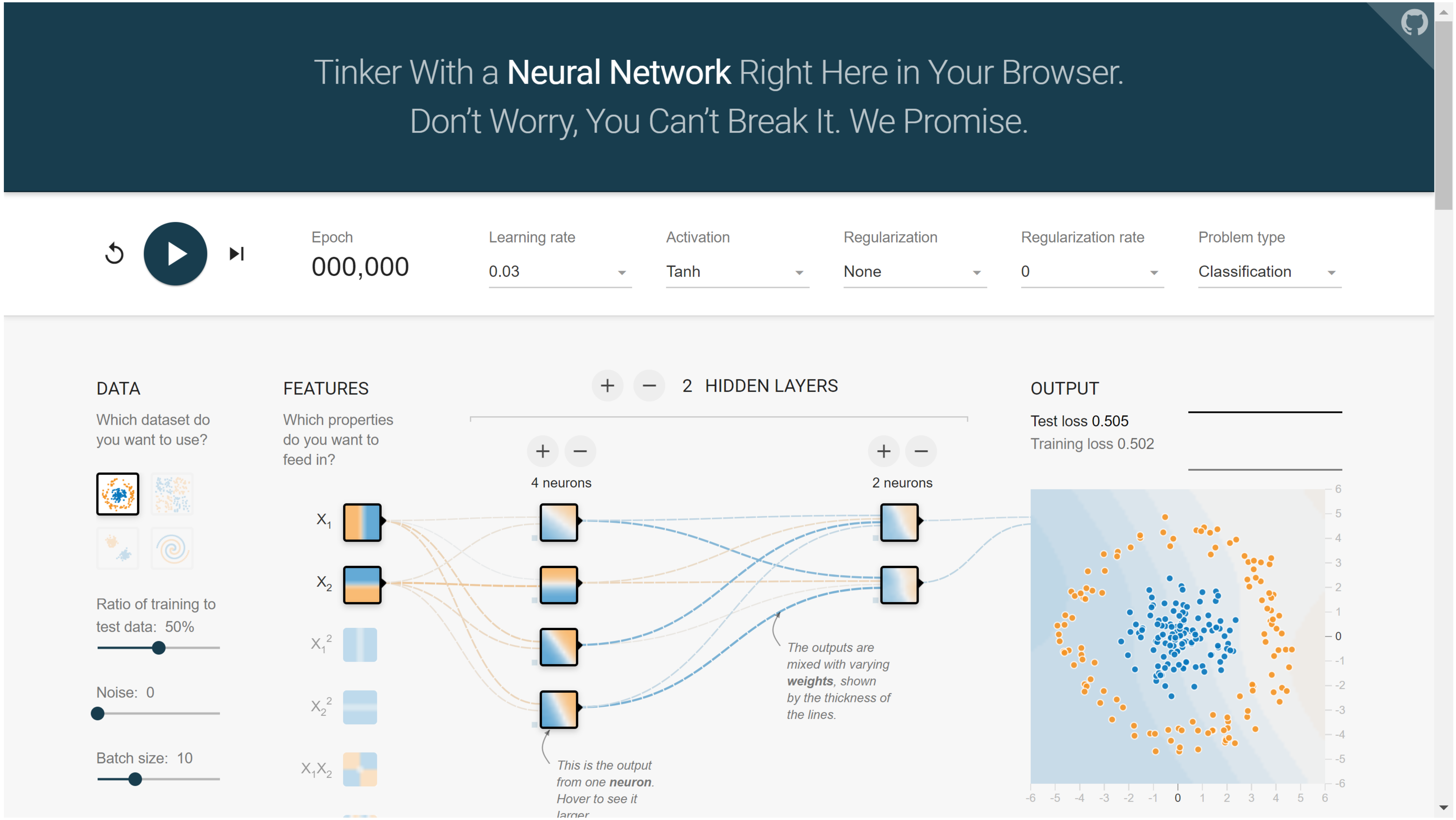
Adapted from material from Pratul Srinivas

Toy problem: storing 2D image data



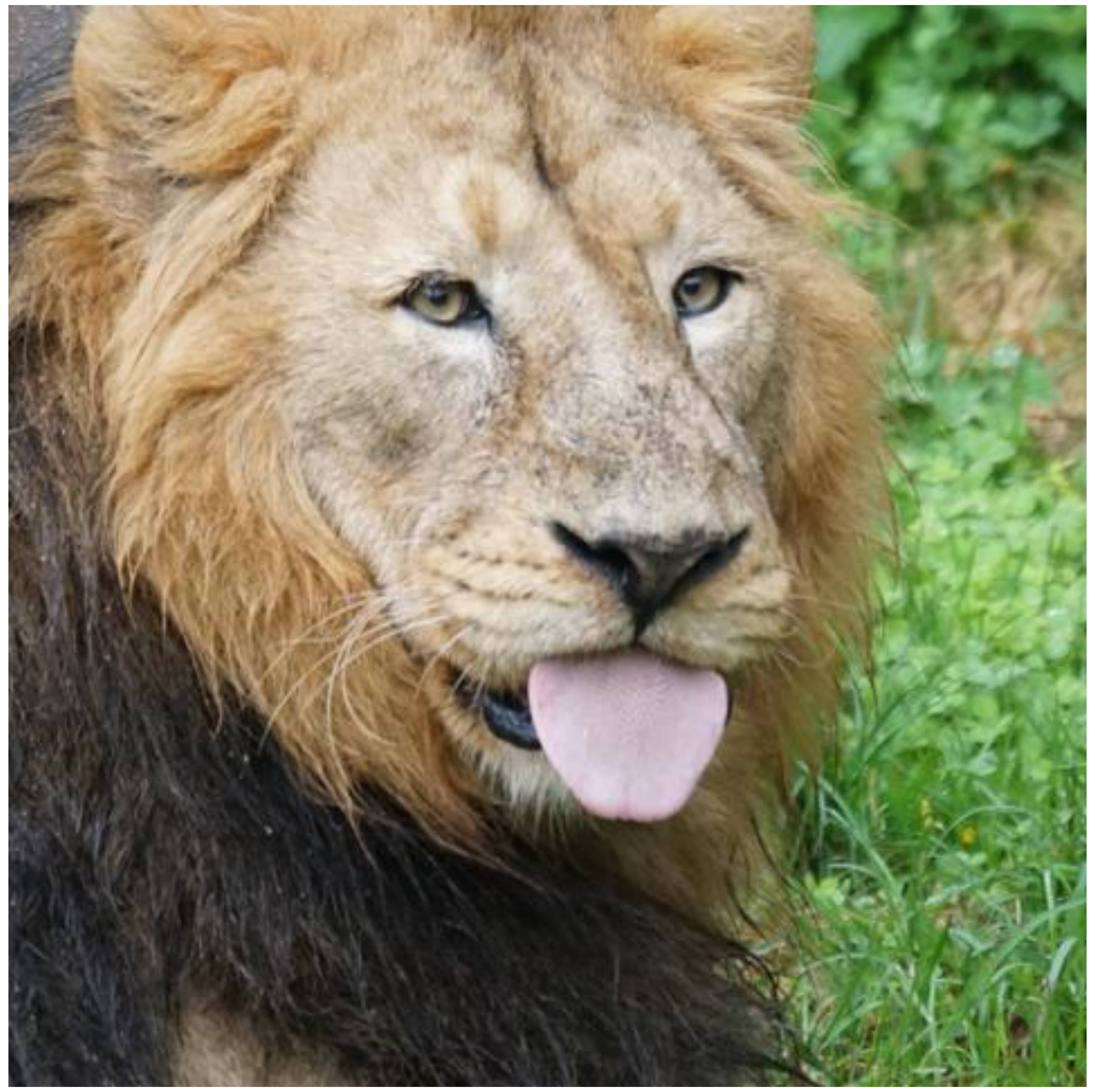
What if we train a simple fully-connected network (MLP) to do this instead?

Recall the TensorFlow playground



Same concept as before, except we are computing an image, instead of a classifier!

Naive approach fails!



Ground truth image

Neural network output fit
with gradient descent

Adapted from material from Pratul Srinivas

Problem:

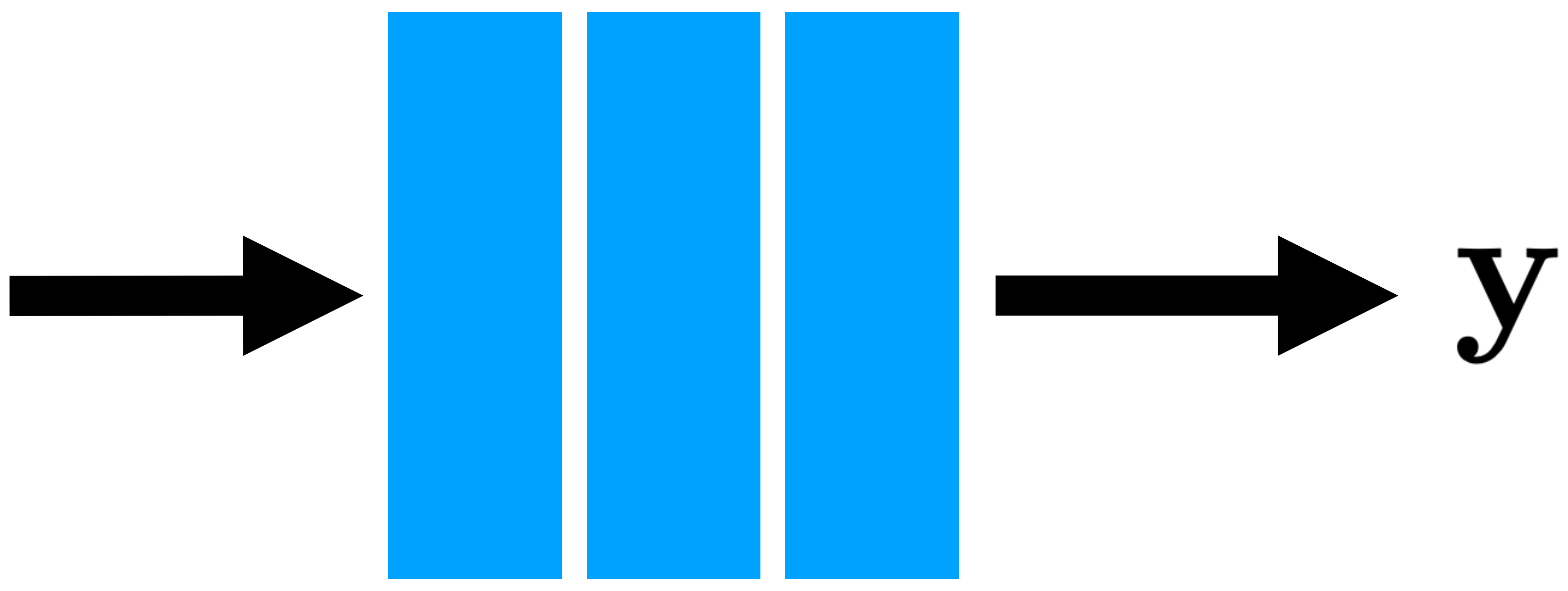
“Standard” coordinate-based MLPs cannot represent high frequency functions

Solution:

Pass input coordinates through a high frequency mapping first

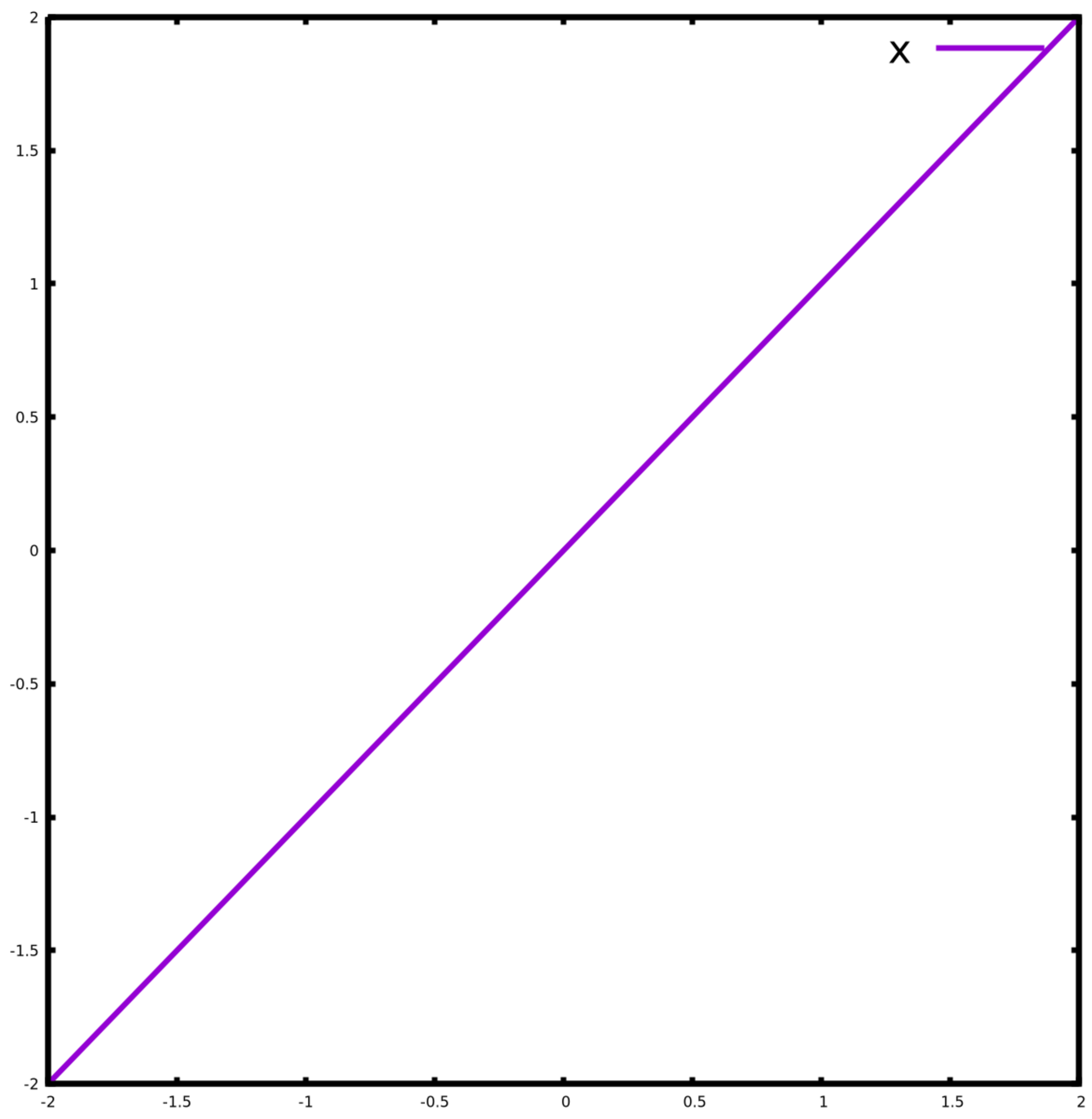
Example mapping: "positional encoding"

$$\begin{pmatrix} \sin(\mathbf{v}), \cos(\mathbf{v}) \\ \sin(2\mathbf{v}), \cos(2\mathbf{v}) \\ \sin(4\mathbf{v}), \cos(4\mathbf{v}) \\ \dots \\ \sin(2^{L-1}\mathbf{v}), \cos(2^{L-1}\mathbf{v}) \end{pmatrix}$$

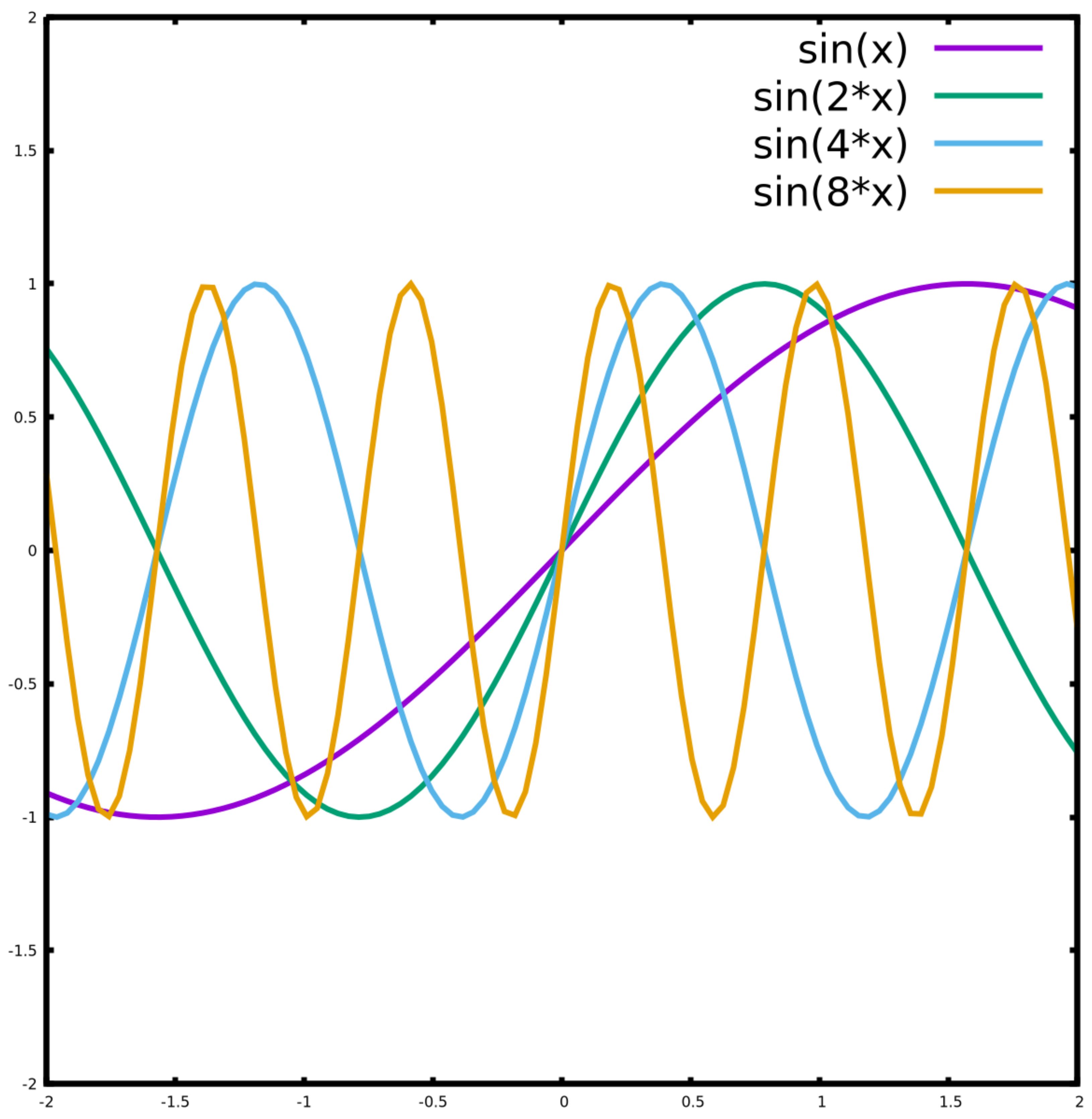


Adapted from material from Pratul Srinivas

Positional encoding



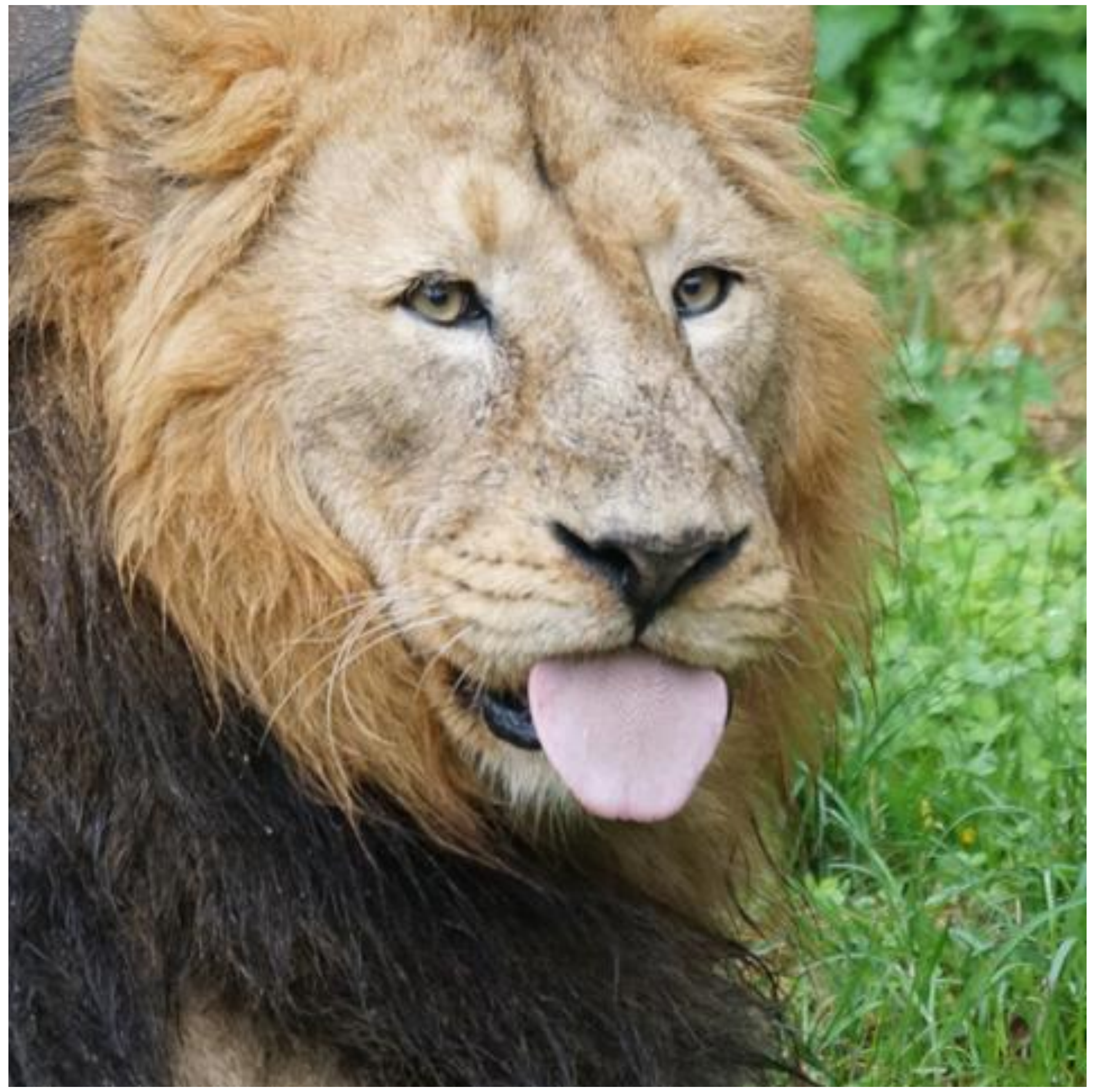
Raw encoding of a number ***x***



"Positional encoding" of a number ***x***

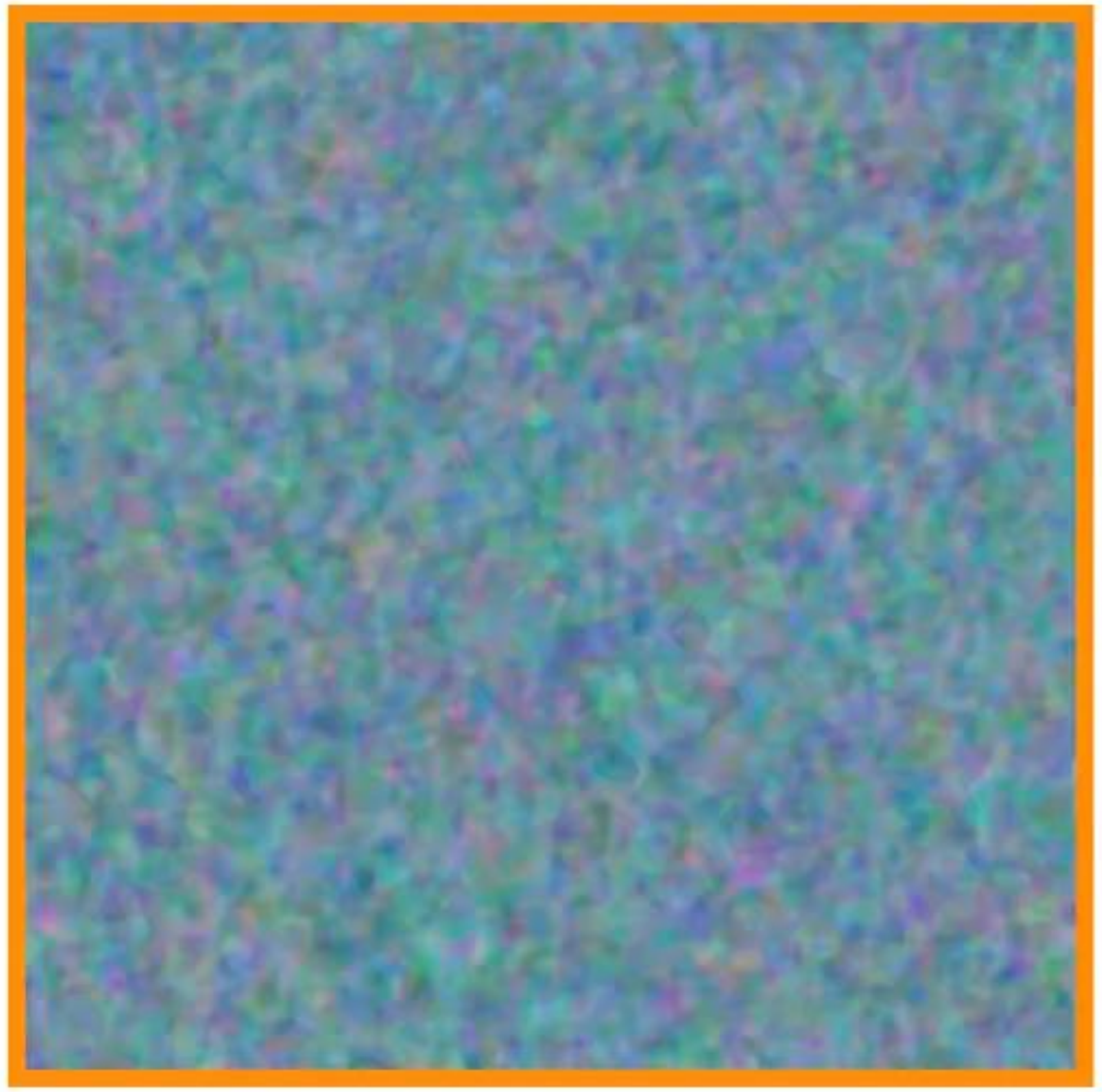
Adapted from material from Pratul Srinivas

Problem solved!



Ground truth image

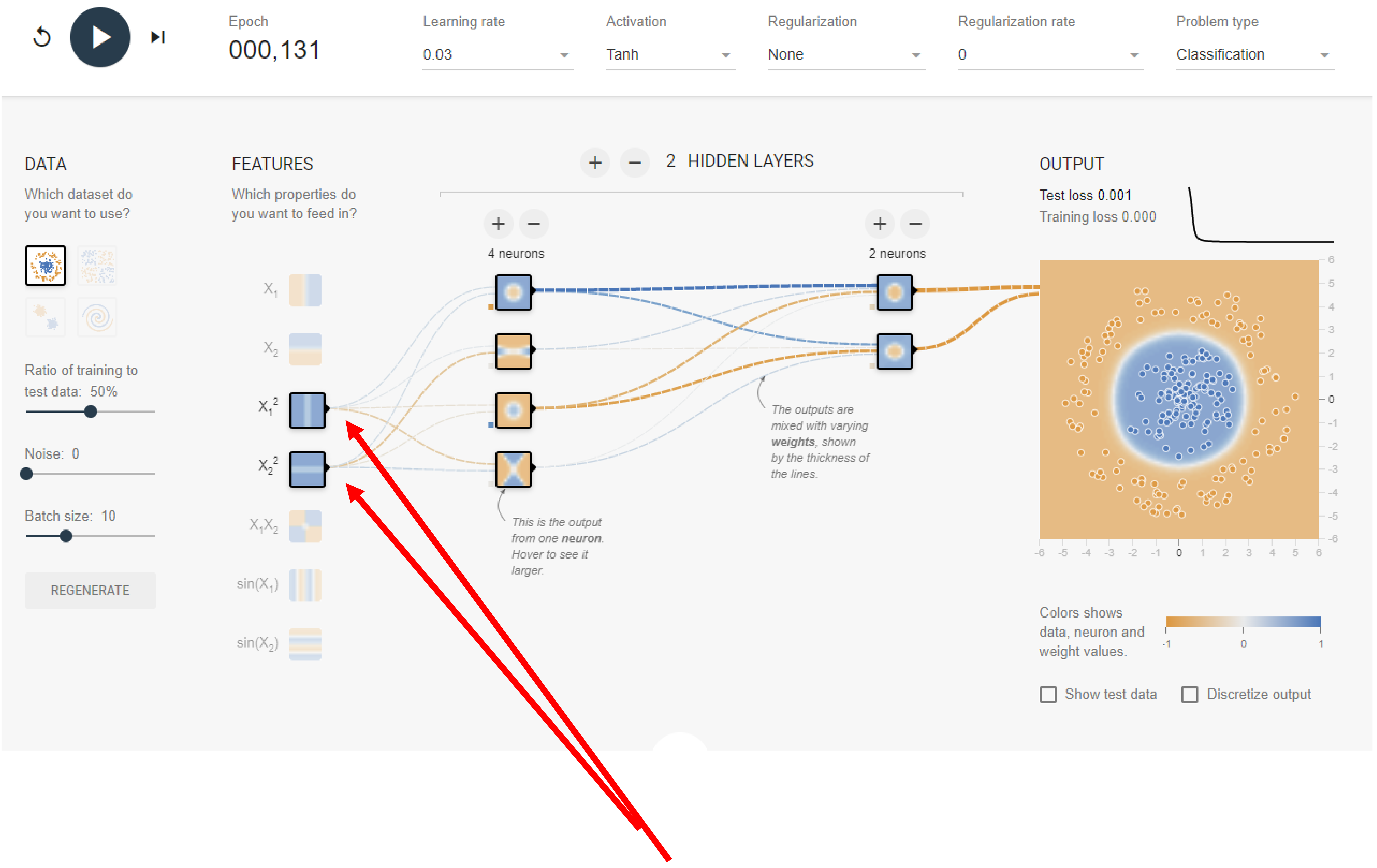
Neural network output without high frequency mapping



Neural network output with high frequency mapping

Adapted from material from Pratul Srinivas

Sometimes a better input encoding is all you need



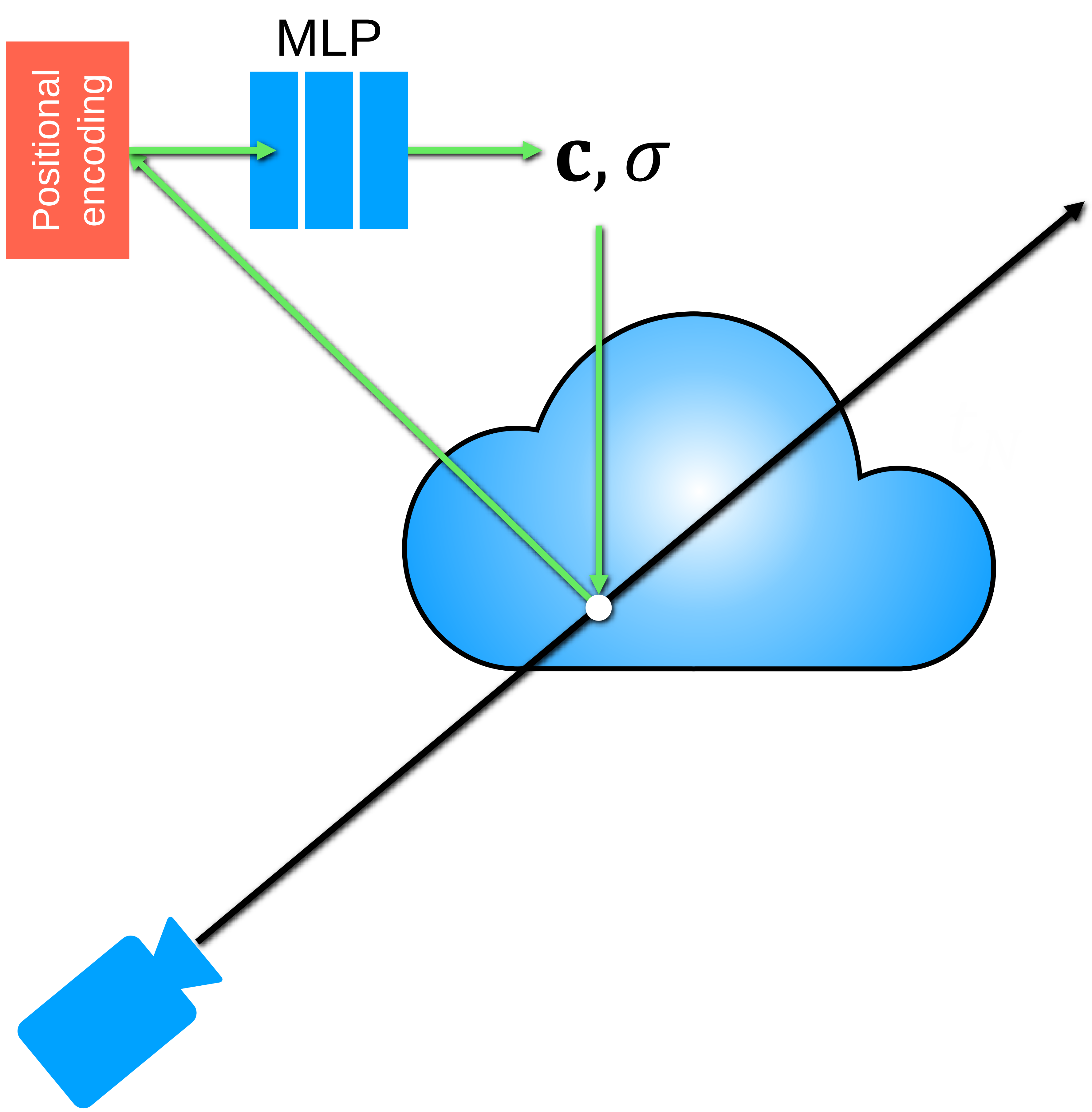
Recall “squared” encoding in TensorFlow Playground

NeRF Overview

- Volumetric rendering
- Neural networks as representations for spatial data
- Neural Radiance Fields (NeRF)

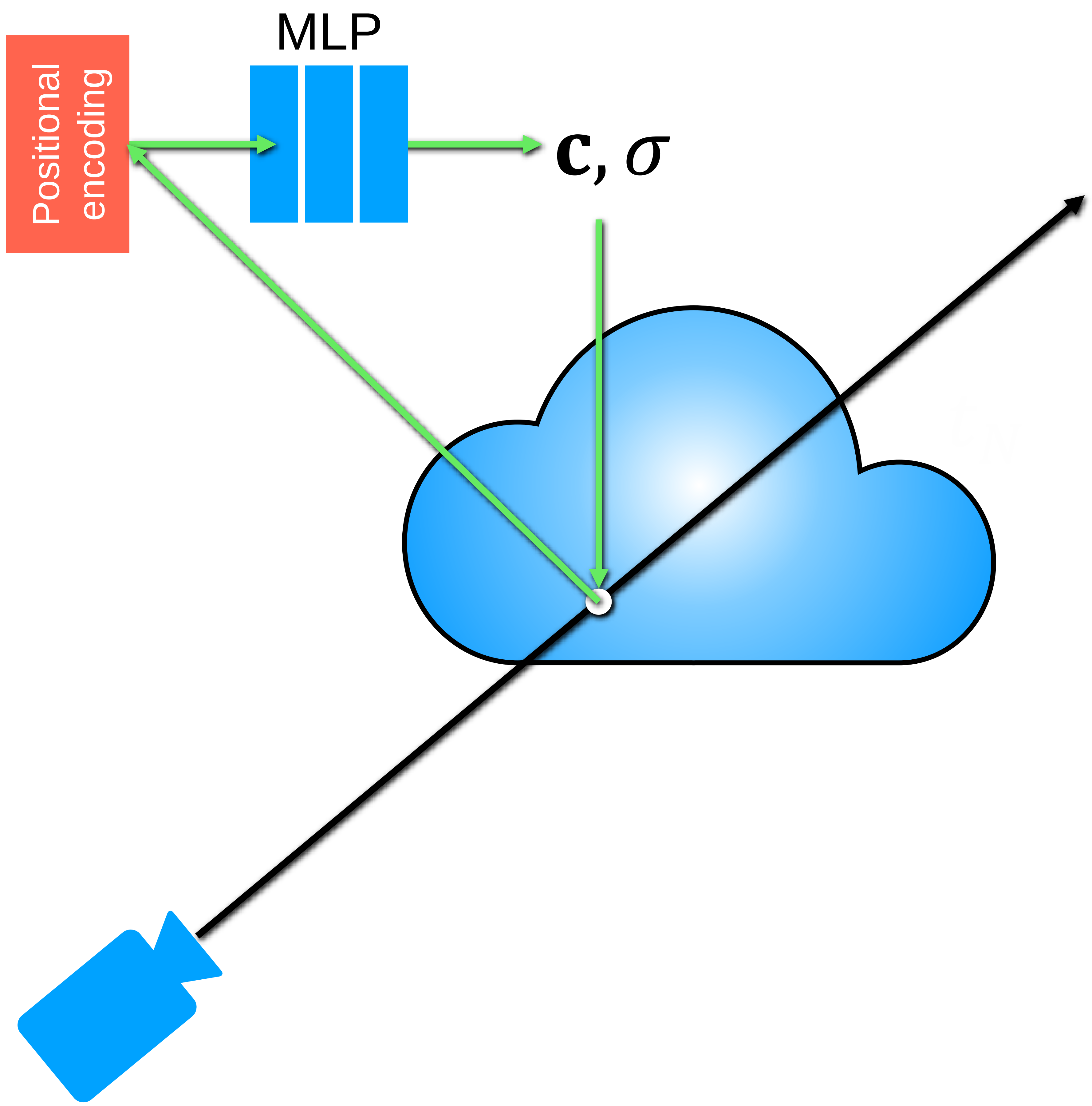
NeRF = volume rendering + coordinate-based network

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



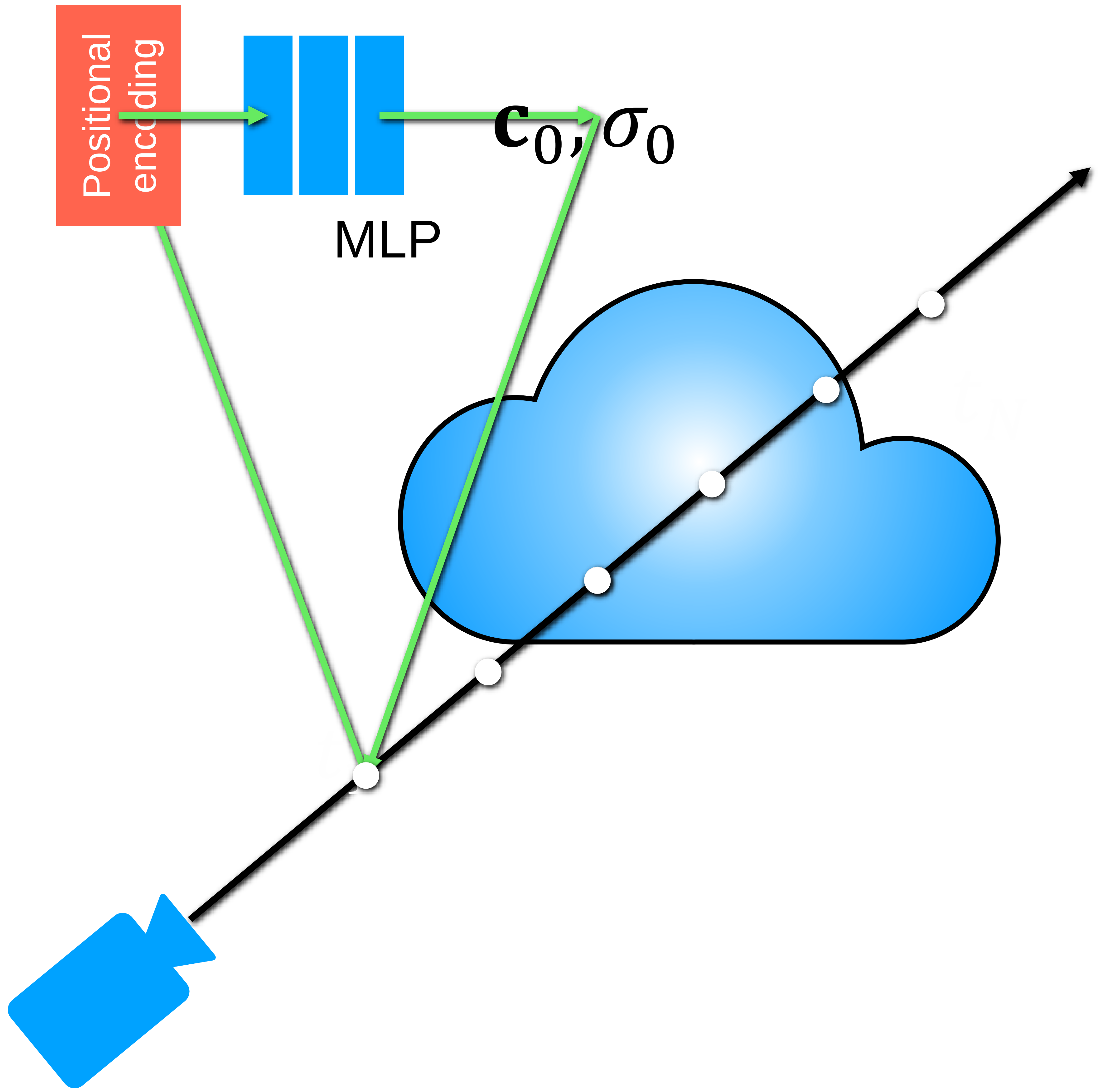
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



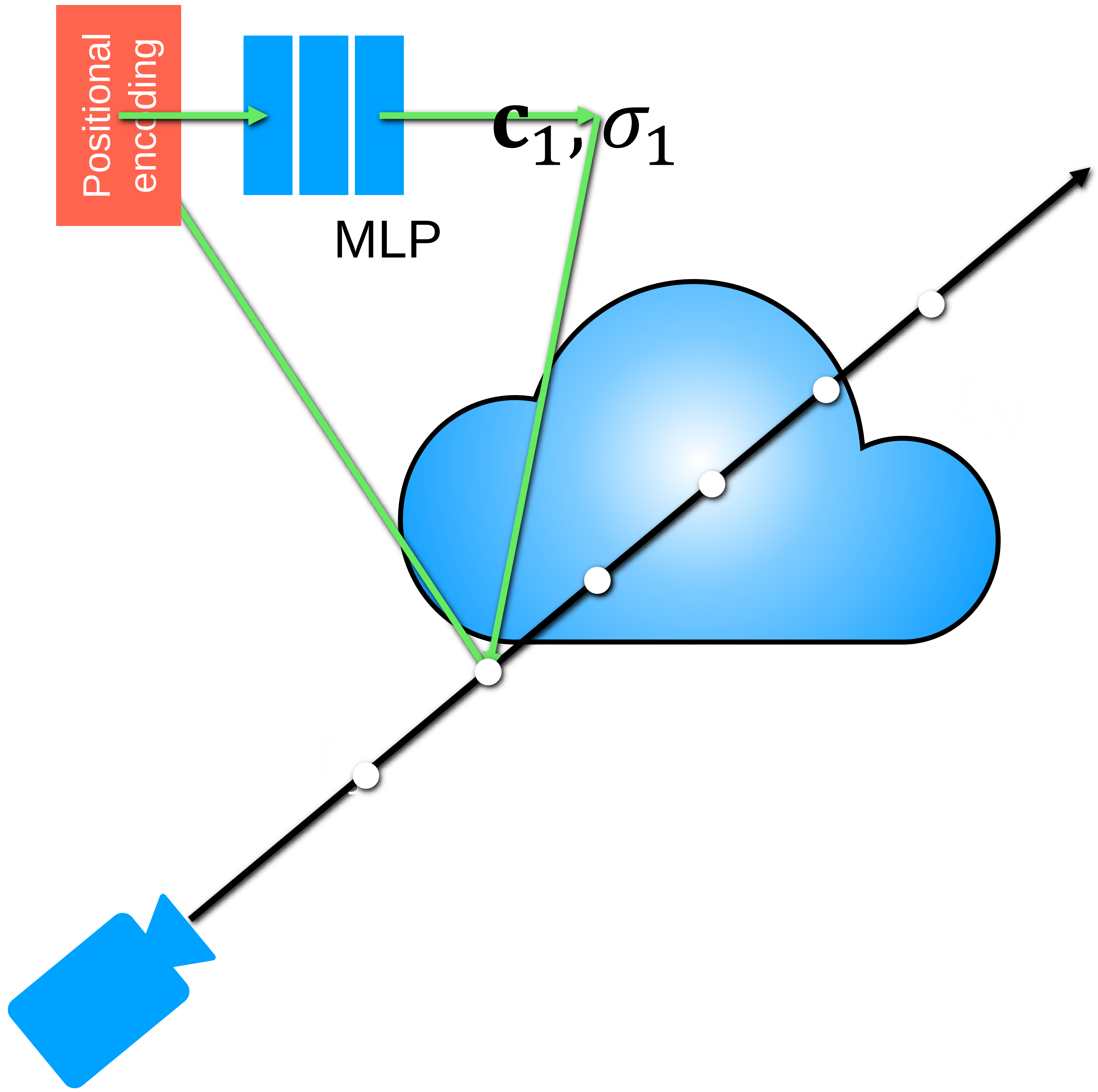
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



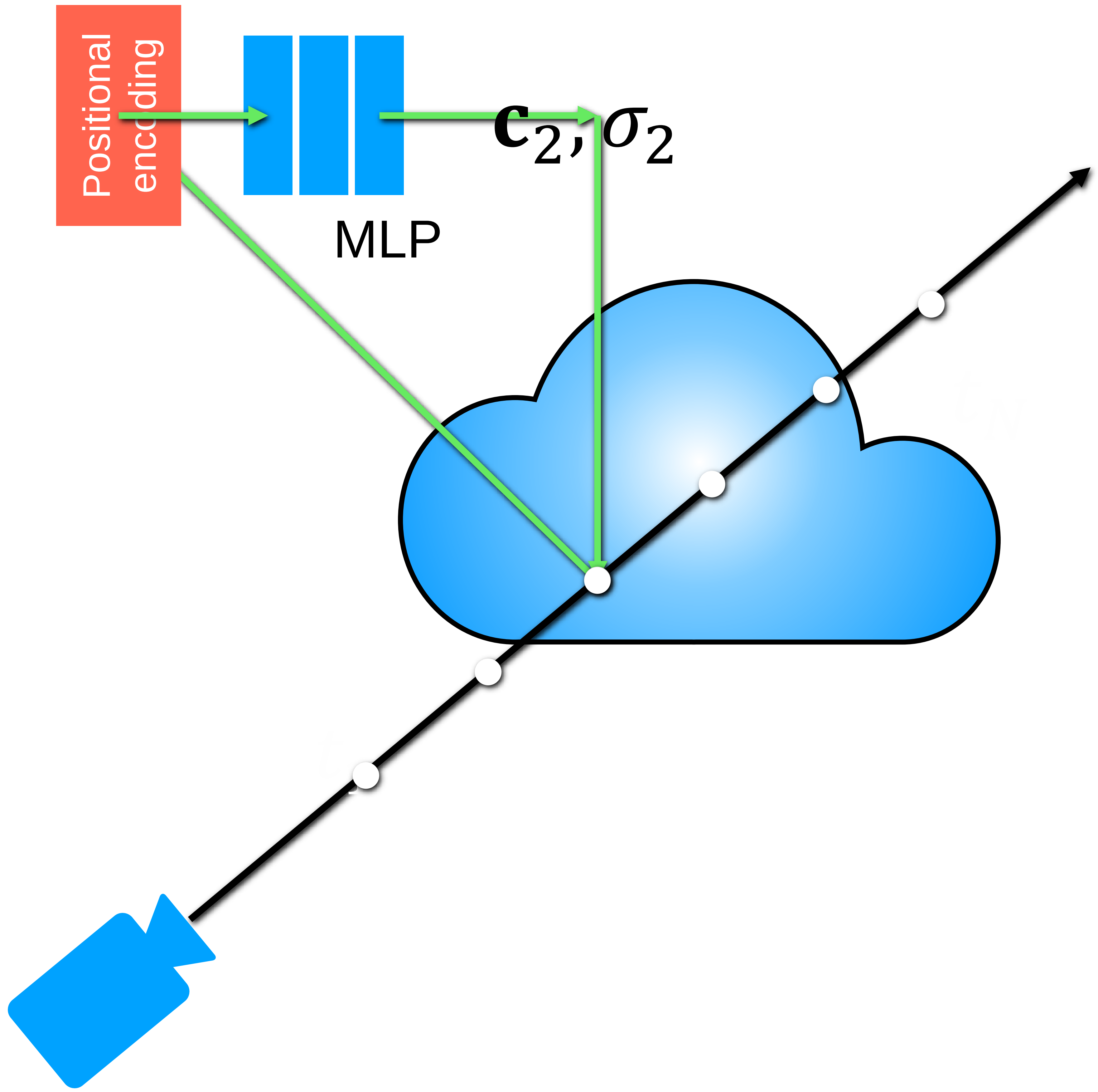
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



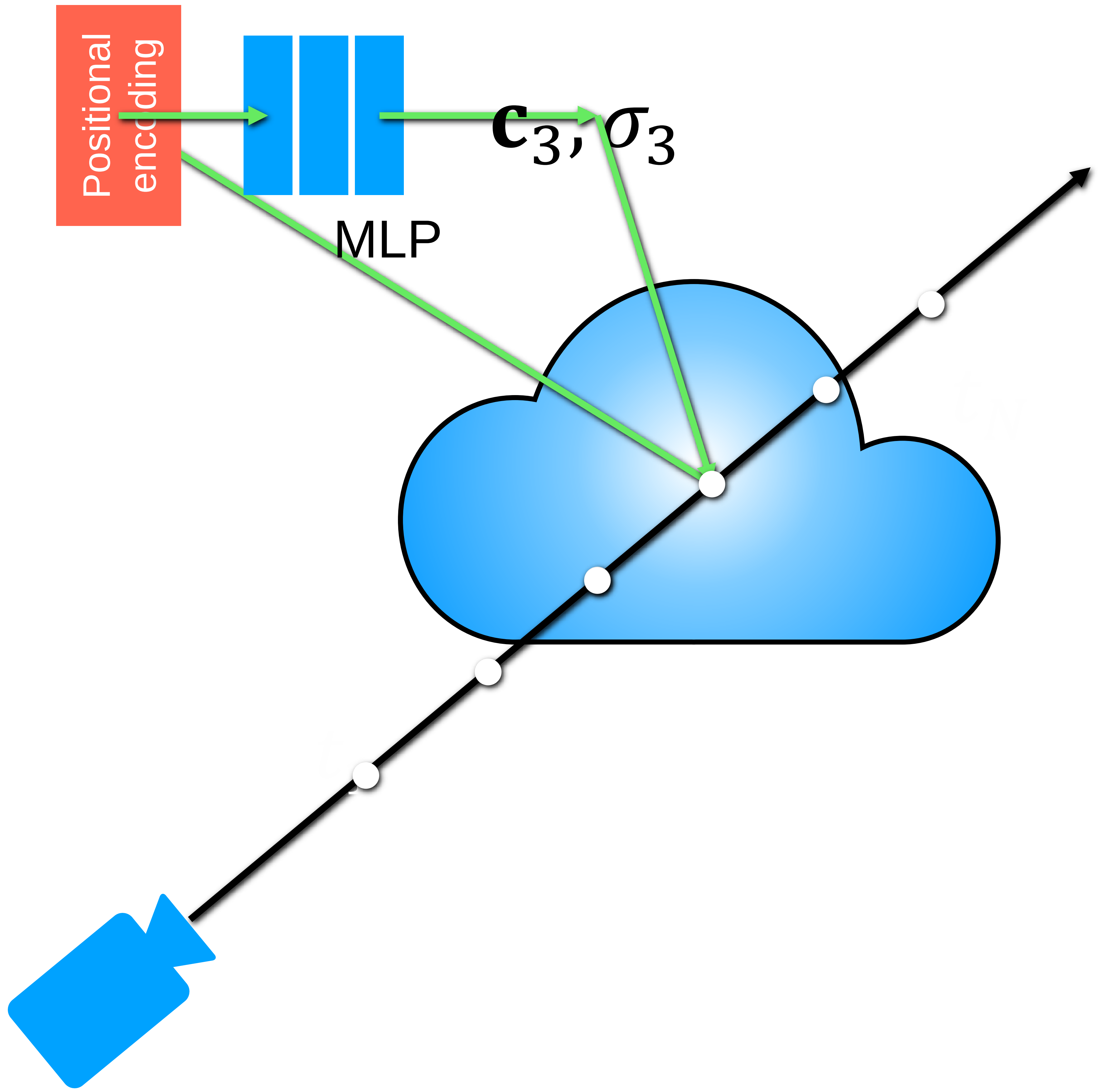
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



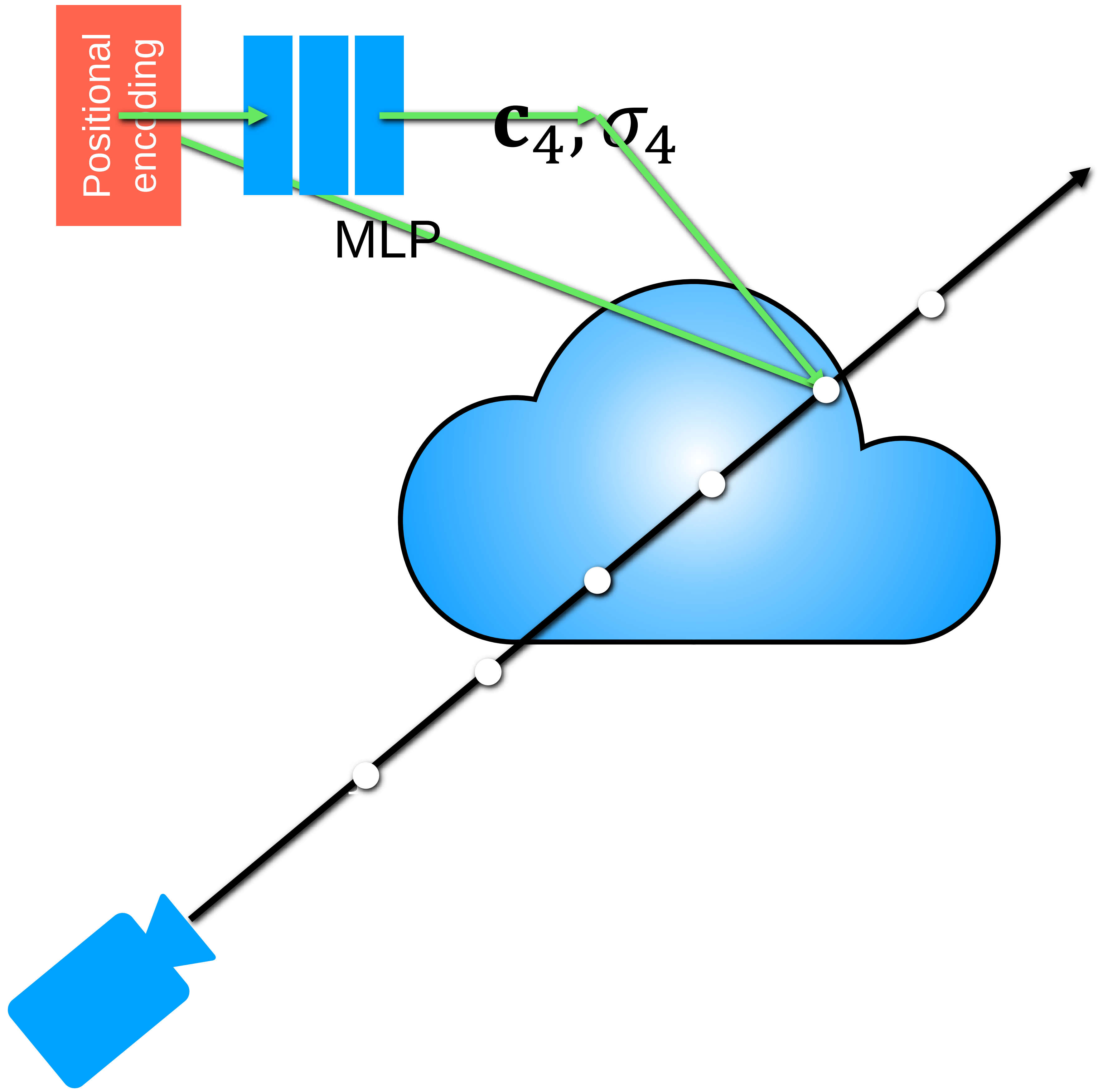
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



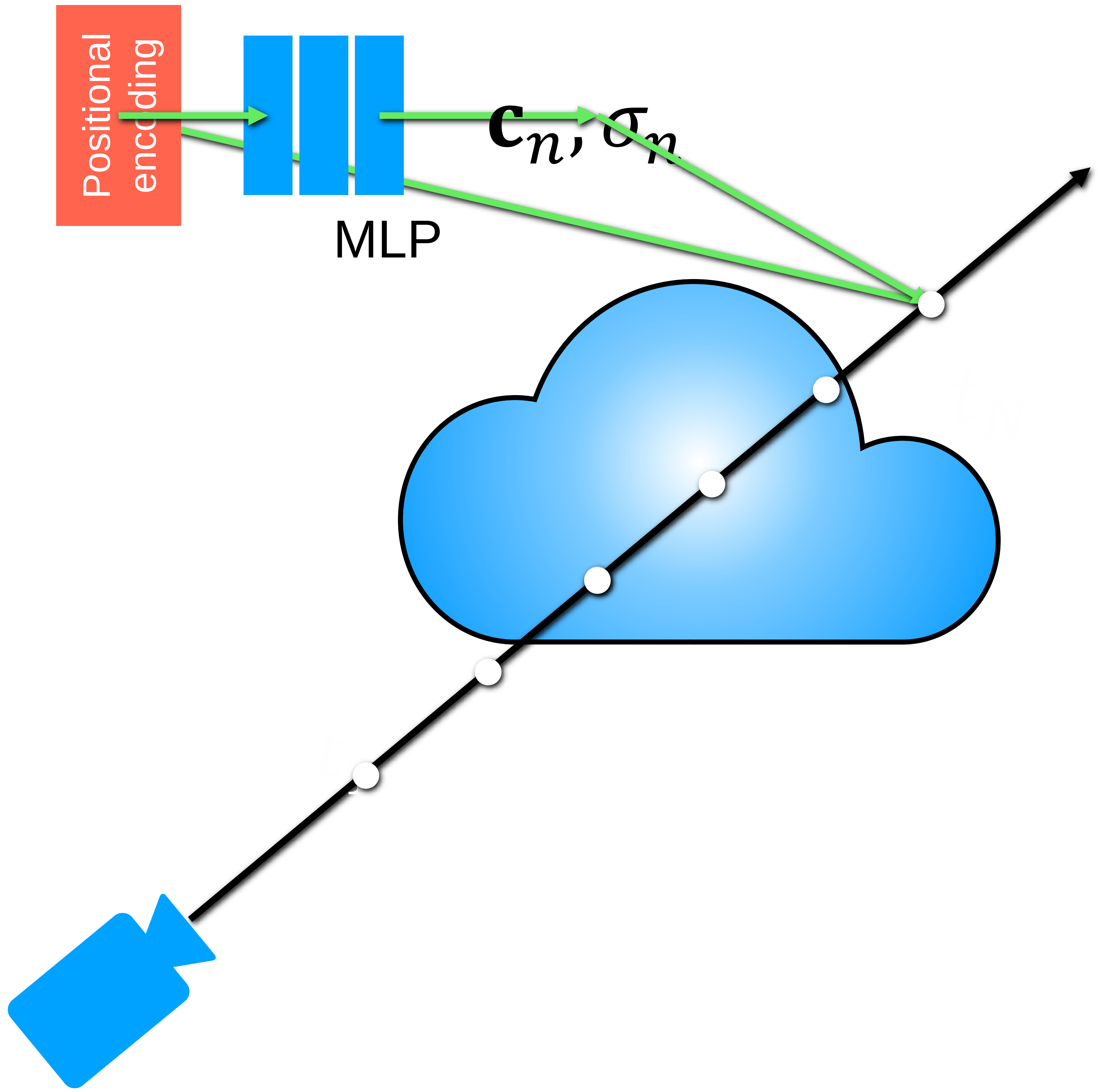
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



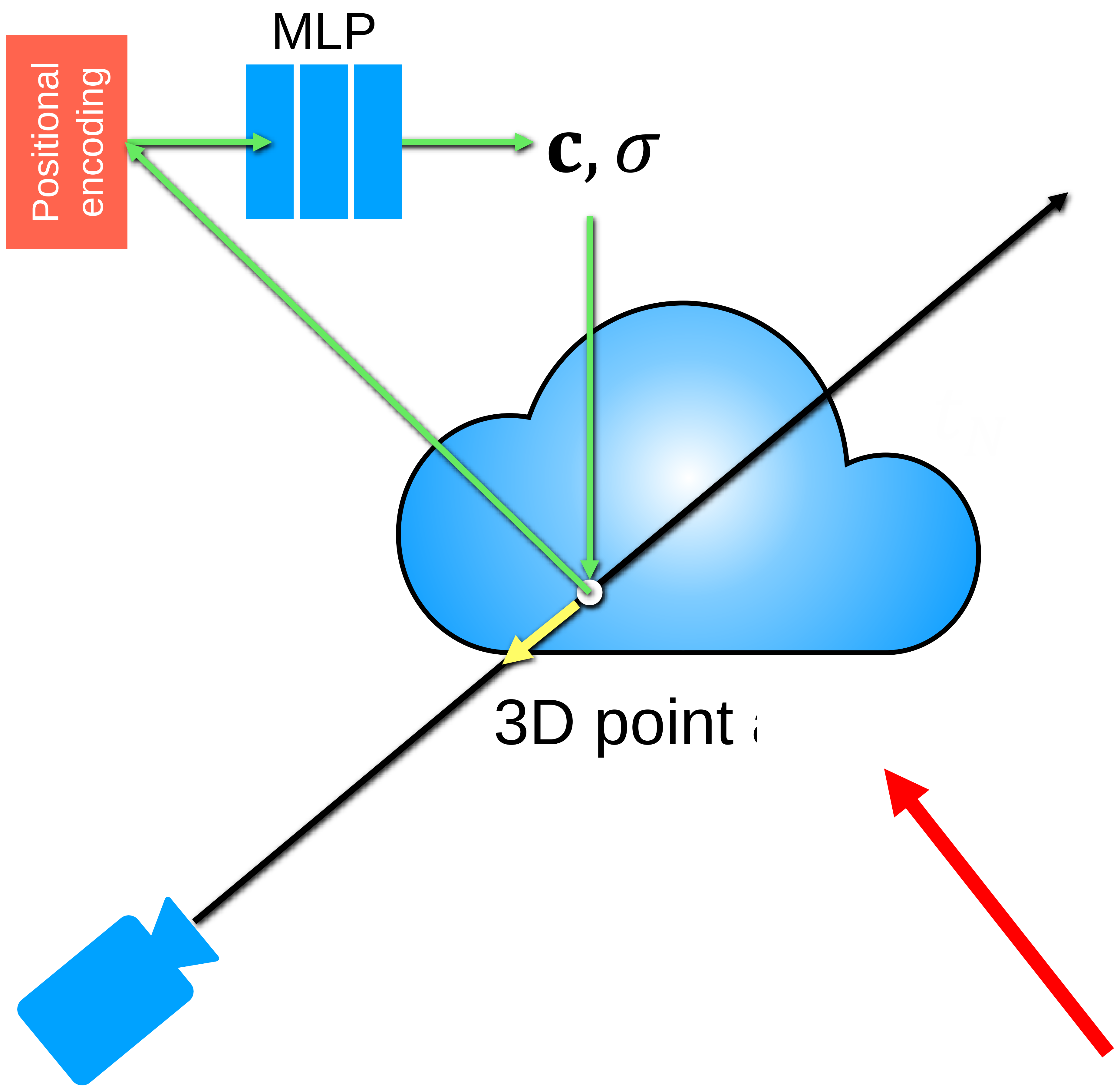
Adapted from material from Pratul Srinivas

How do we store the values of $\mathbf{c}, \sigma$ at each point in space?



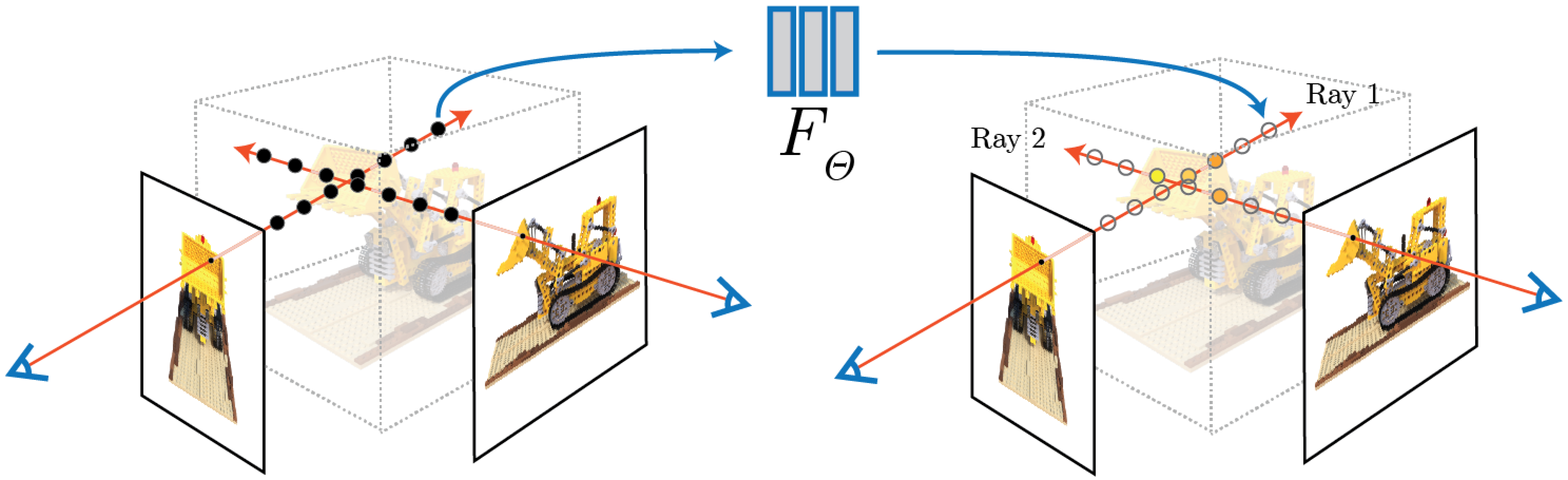
Adapted from material from Pratul Srinivas

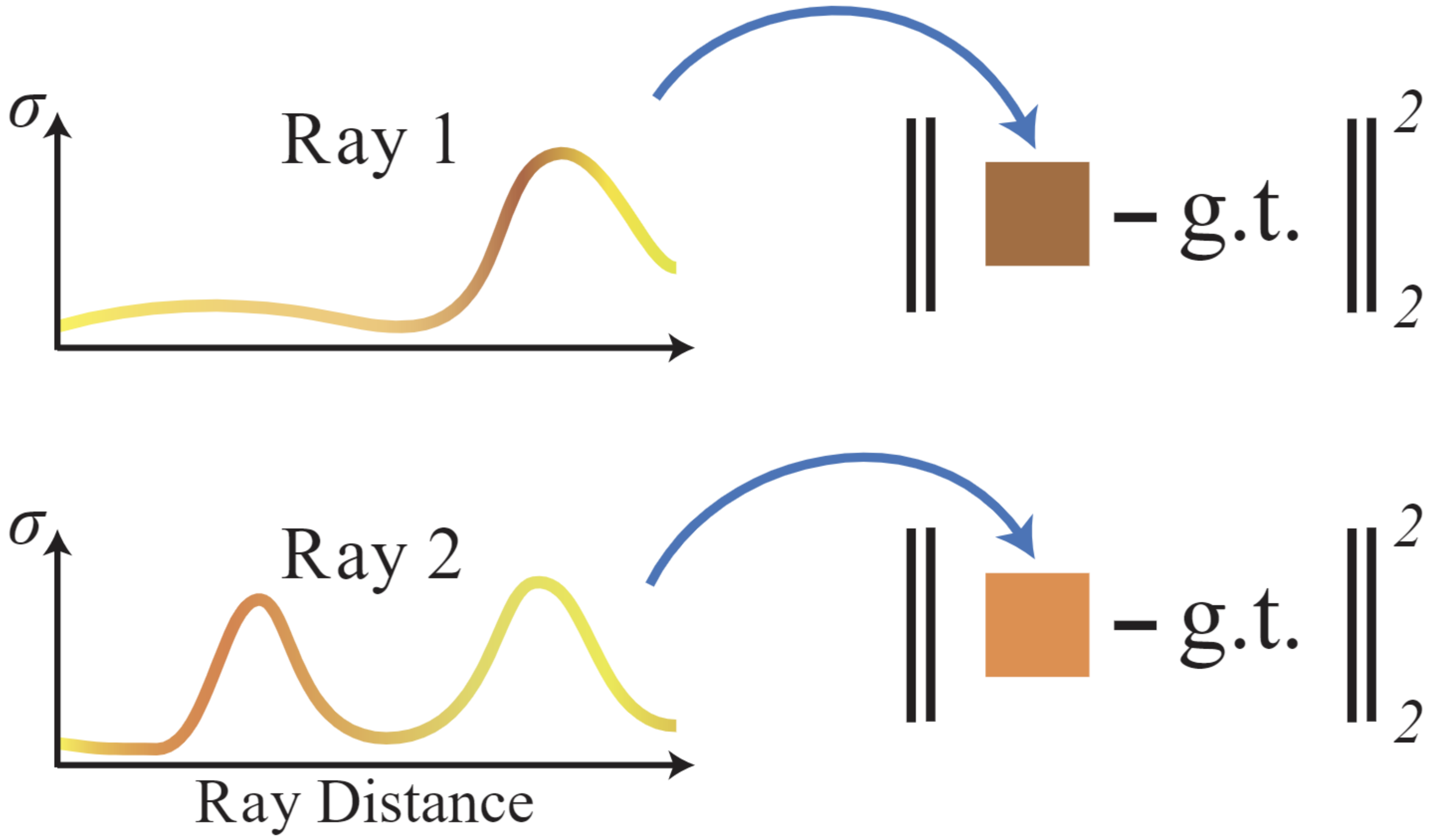
Extension: view-dependent field



Include the ray direction in the input to the MLP → allows for capturing and rendering view-dependent effects (e.g., shiny surfaces)

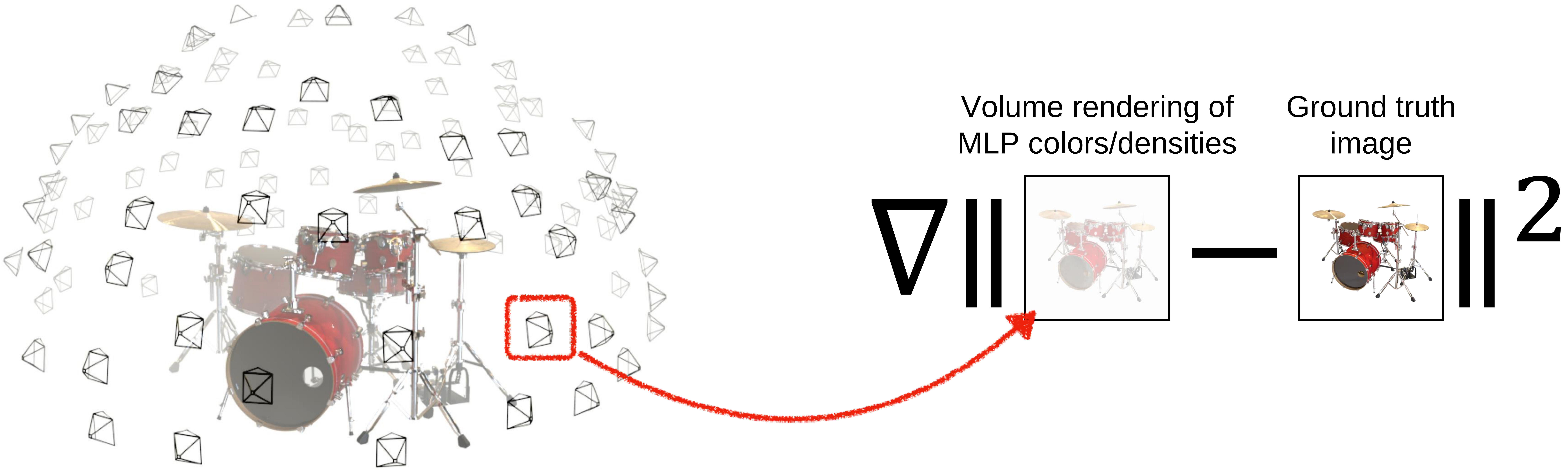
Putting it all together





Adapted from material from Pratul Srinivas

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



Adapted from material from Pratul Srinivas

Results

NeRF encodes convincing view-dependent effects using directional dependence

Adapted from material from Pratul Srinivas

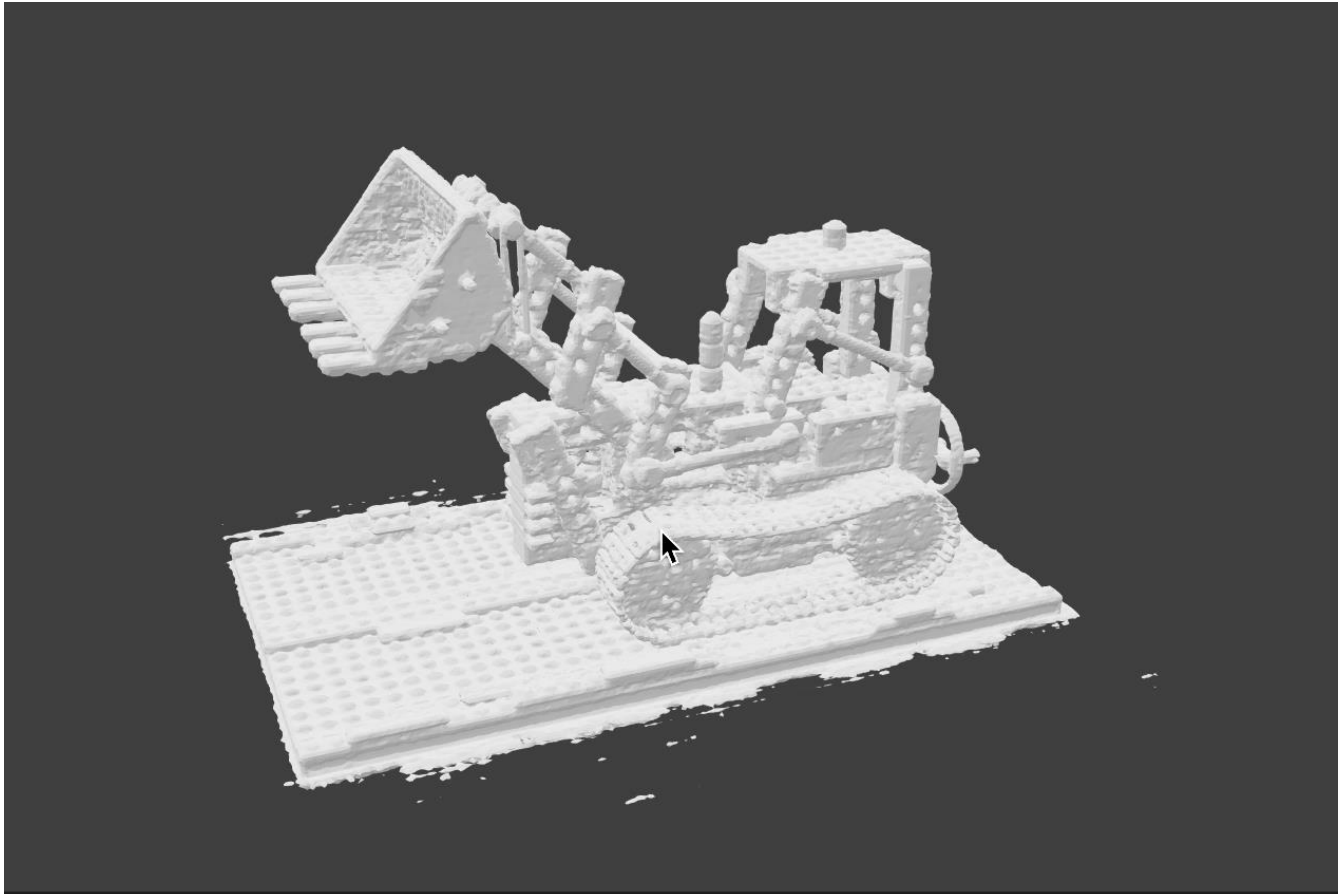
NeRF encodes convincing view-dependent effects using directional dependence

Adapted from material from Pratul Srinivas

NeRF encodes detailed scene geometry with occlusion effects

Adapted from material from Pratul Srinivas

NeRF encodes detailed scene geometry



Adapted from material from Pratul Srinivas

Summary

- Represent the scene as volumetric colored “fog”
- Store the fog color and density at each point as an MLP mapping 3D position (x, y, z) to color c and density σ
- Render image by shooting a ray through the fog for each pixel
- Optimize MLP parameters by rendering to a set of known viewpoints and comparing to ground truth images

Extension: NeRF in the Wild (NeRF-W)

Brandenburg Gate

Sacre Coeur

Trevi Fountain

Martin-Brualla*, Radwan*, Sajjadi*, Barron, Dosovitskiy, Duckworth. *NeRF in the Wild*. CVPR 2021.
https://www.youtube.com/watch?v=mRAKVQj5LRA

Inverse graphics beyond shape and color

Input images of
an object

Reconstructed shape,
albedo, and materials

Reconstructed models inserted into scene with new lighting

Zhang, Luan, Li, Snavely. CVPR

Questions?

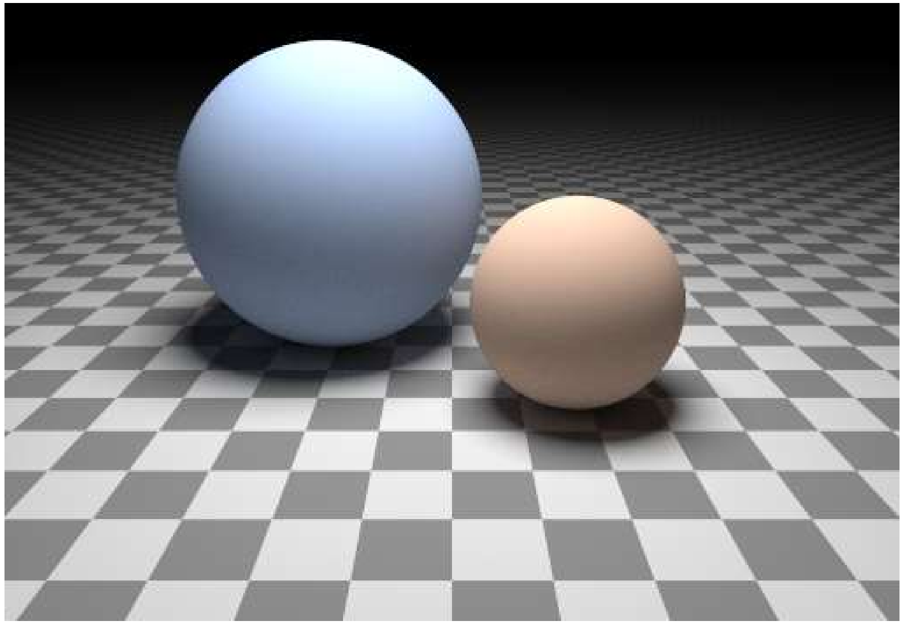
Computer Vision

Lecture 3 Light & Perception

Reading

- Szeliski 2nd Edition, Chapter 2.2

Can we determine shape from *lighting*?



- Are these spheres?
  - Or just flat discs painted with varying color (albedo)?
  - There is ambiguity between *shading* and *reflectance*
  - But still, as humans we can understand the shapes of these objects

What we know: Stereo

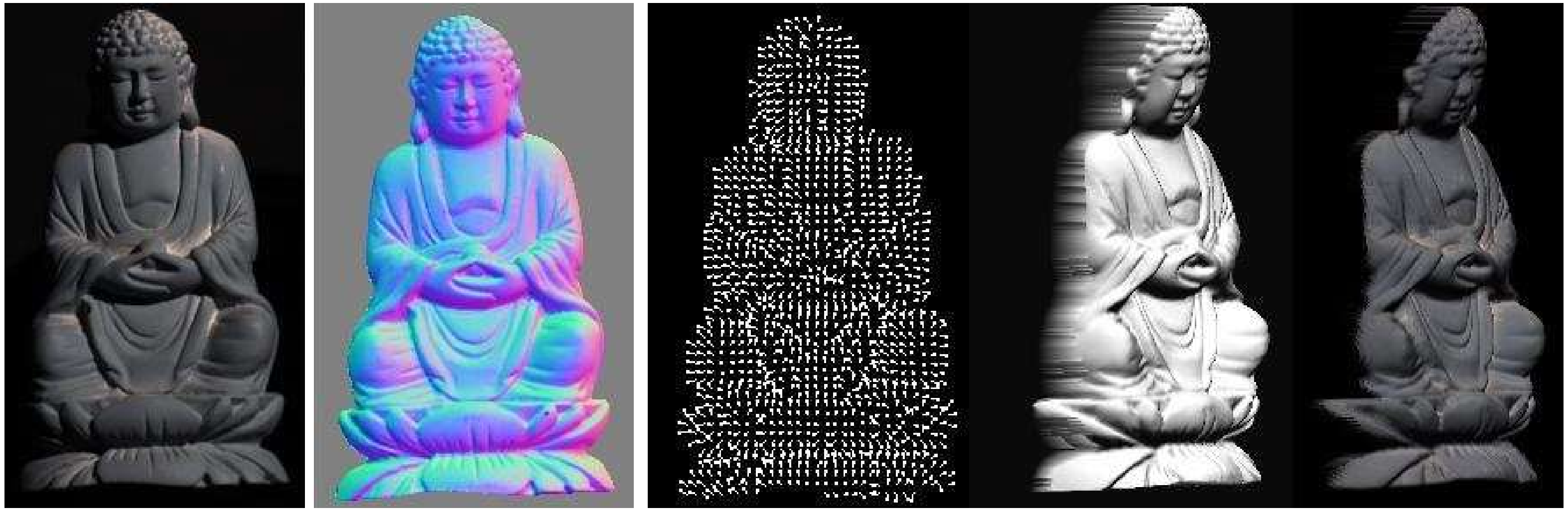
Key Idea: use camera motion to compute shape

Next: Photometric Stereo

Key Idea: use pixel brightness to understand shape

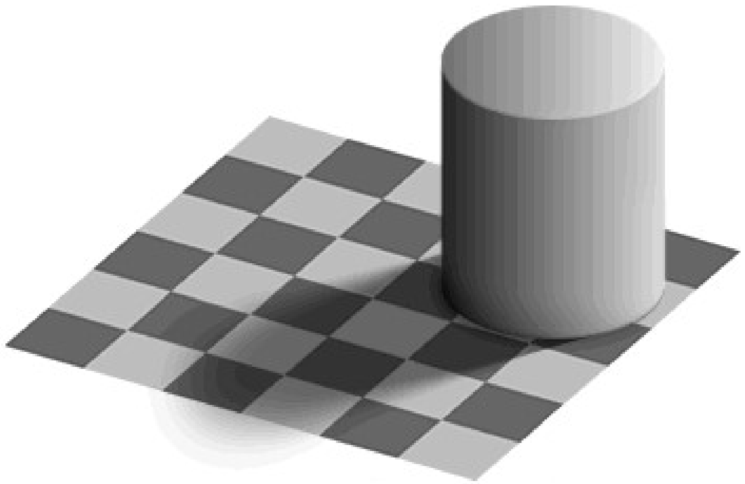
Photometric Stereo

What results can you get?



Input (1 of 12) | Normals (RGB colormap) | Normals (vectors) | Shaded 3D rendering | Textured 3D rendering

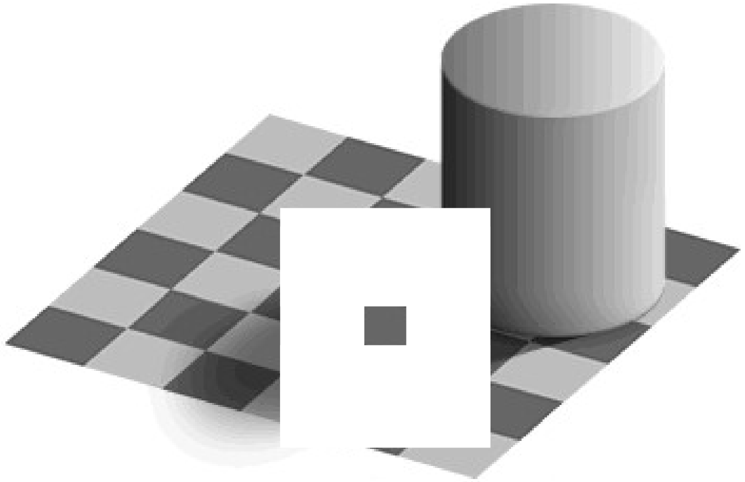
Light



by Ted Adelson

- Readings
  - Szeliski, 2.2, 2.3

Light



by Ted Adelson

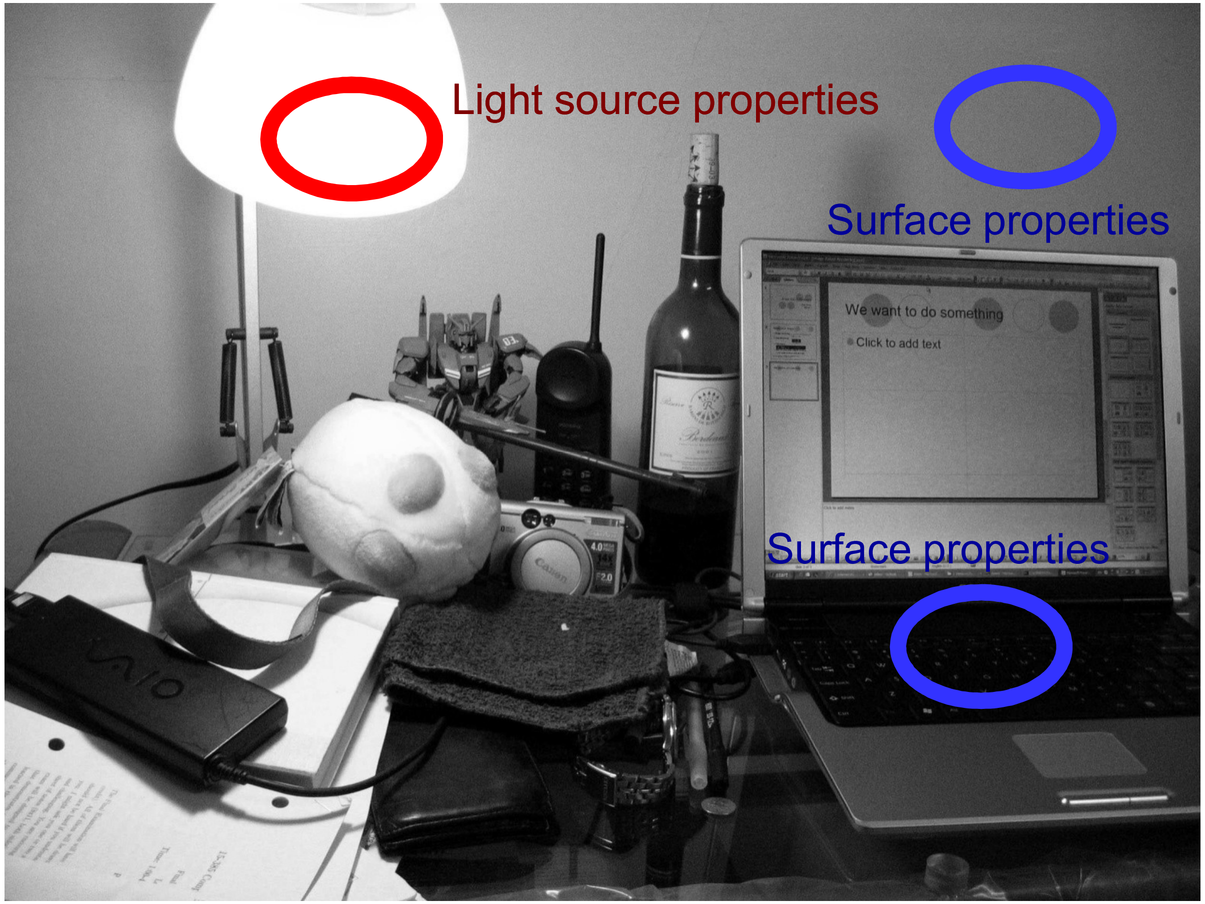
- Readings
  - Szeliski, 2.2, 2.3

Properties of light

- Today
  - What is light?
  - How do we measure it?
  - How does light propagate?
  - How does light interact with matter?

Radiometry

- What determines the brightness of a pixel?



Radiometry

- What determines the brightness of a pixel?

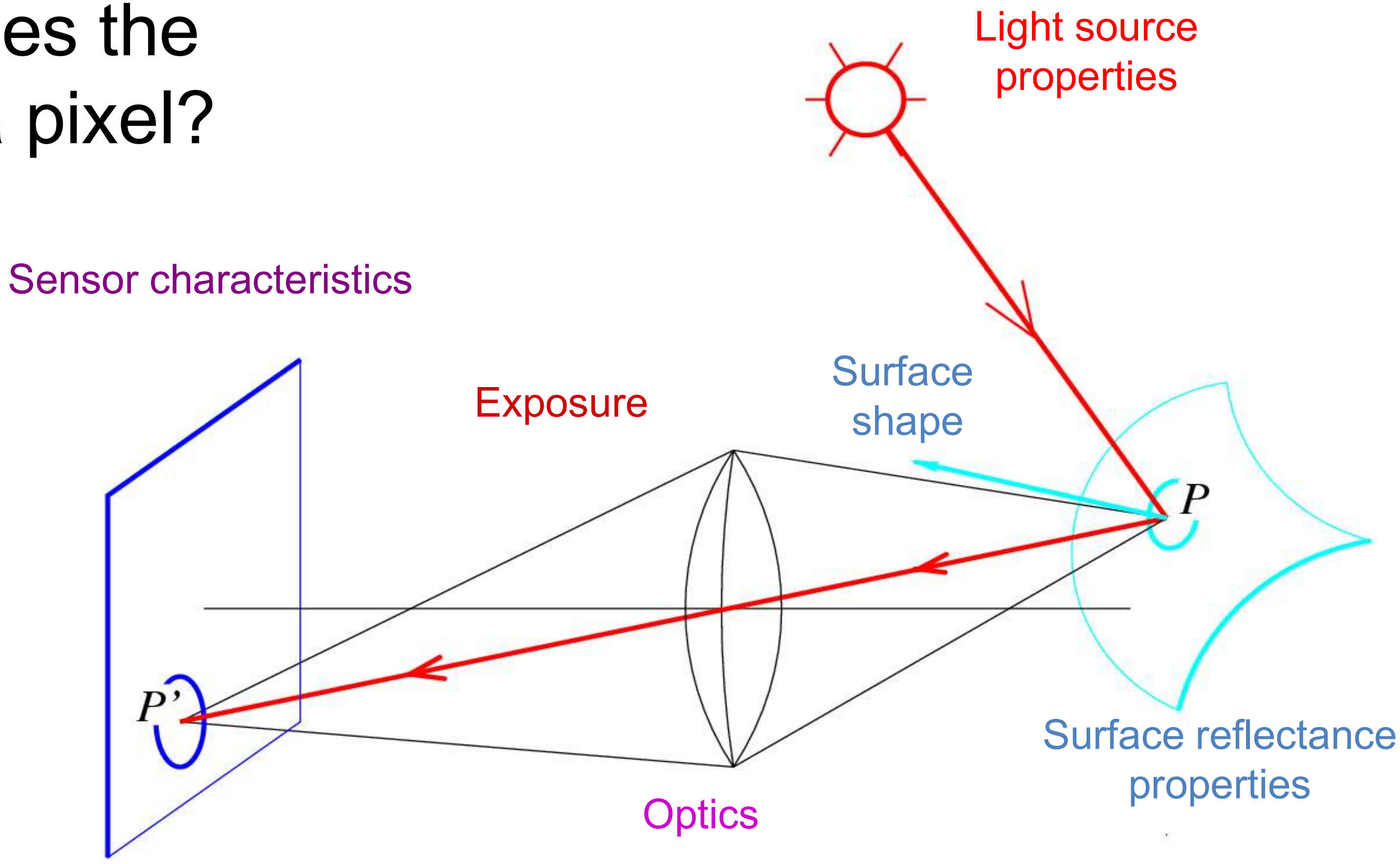
@robertwestonbreshears
https://www.instagram.com/p/BtgX55ZBhU-/

Radiometry

- What determines the brightness of a pixel?

Radiometry

- What determines the brightness of a pixel?

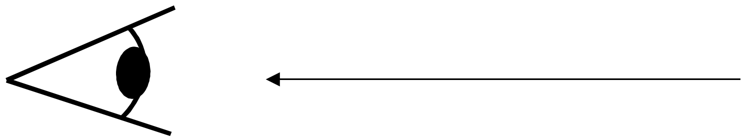


Slide by L. Fei-Fei

Color perception

Electromagnetic radiation (EMR) moving along rays in space

- R(λ) is EMR, measured in units of power (watts)
  - λ is wavelength

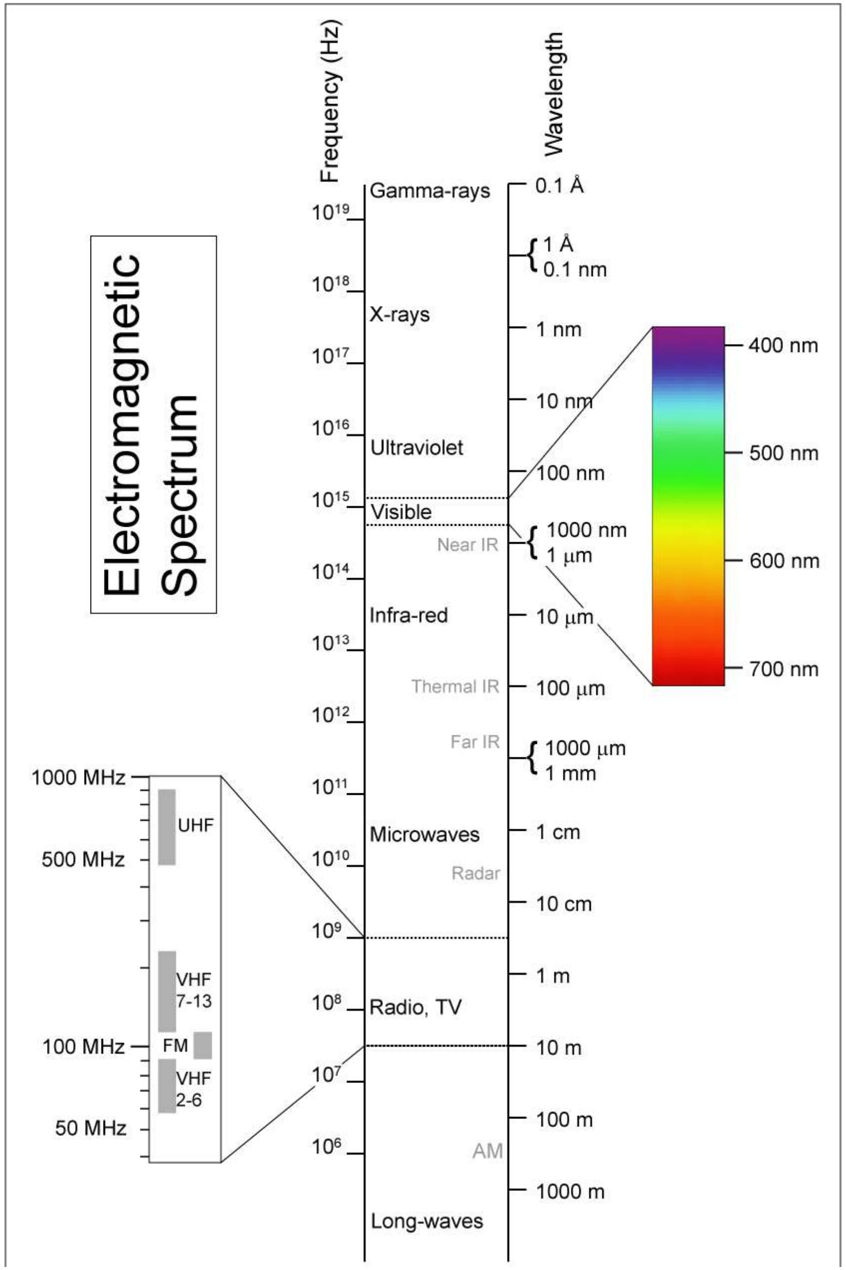


Perceiving light

- How do we convert radiation into “color”?
- What part of the spectrum do we see?

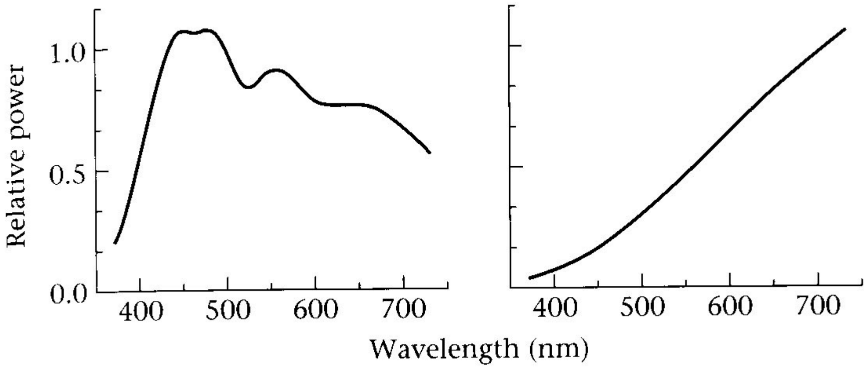
Visible light

We “see” electromagnetic radiation in a range of wavelengths

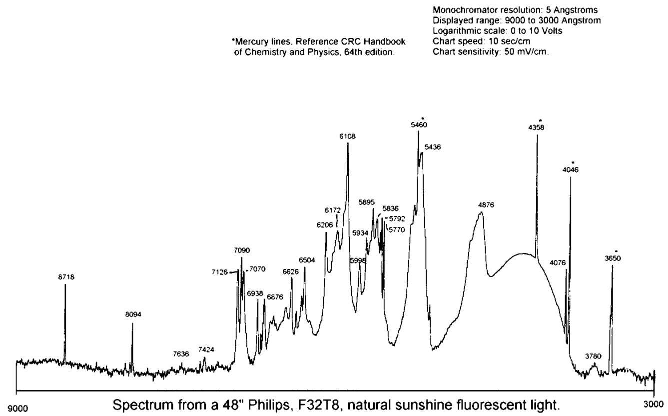


Light spectrum

- The appearance of light depends on its power **spectrum**
  - How much power (or energy) at each wavelength



daylight tungsten bulb



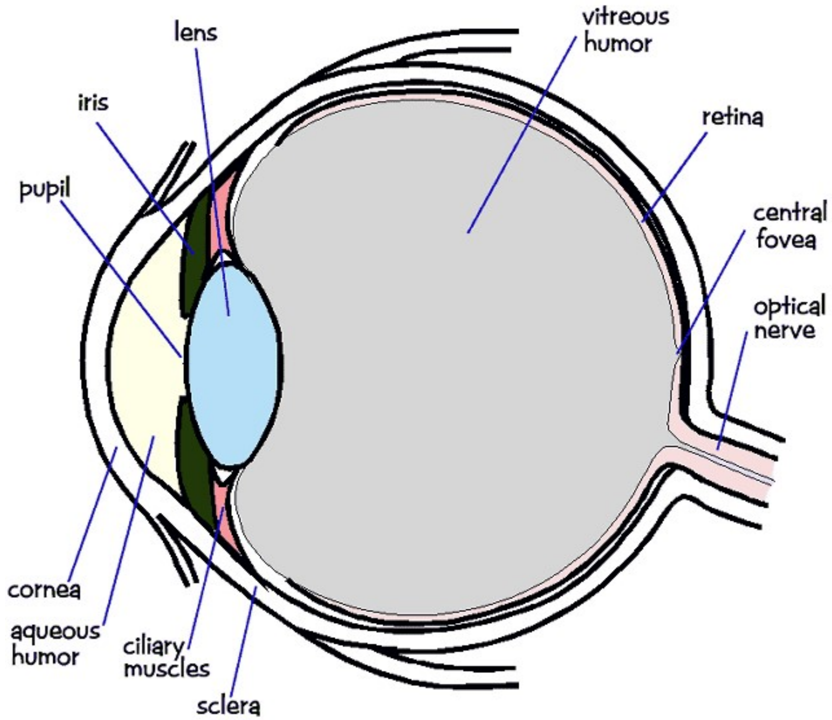
fluorescent bulb

Our visual system converts a light spectrum into “color”

- This is a rather complex transformation

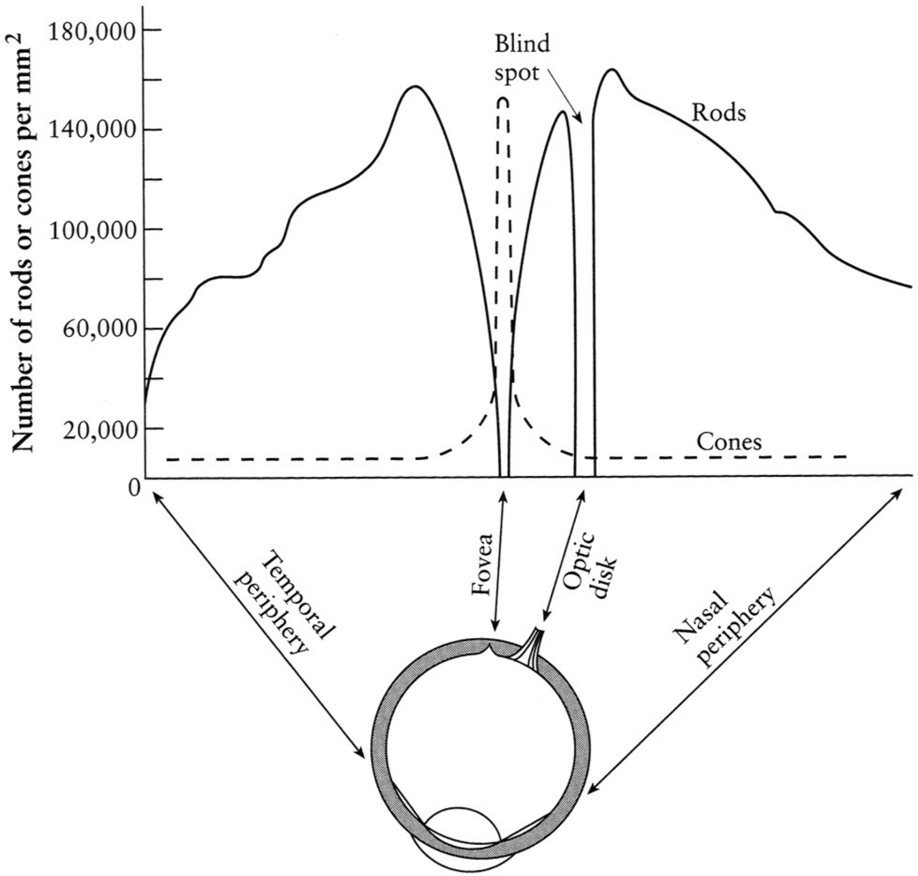
The human visual system

- Color perception
  - Light hits the retina, which contains photosensitive cells
    - rods and cones
  - These cells convert the spectrum into a few discrete values

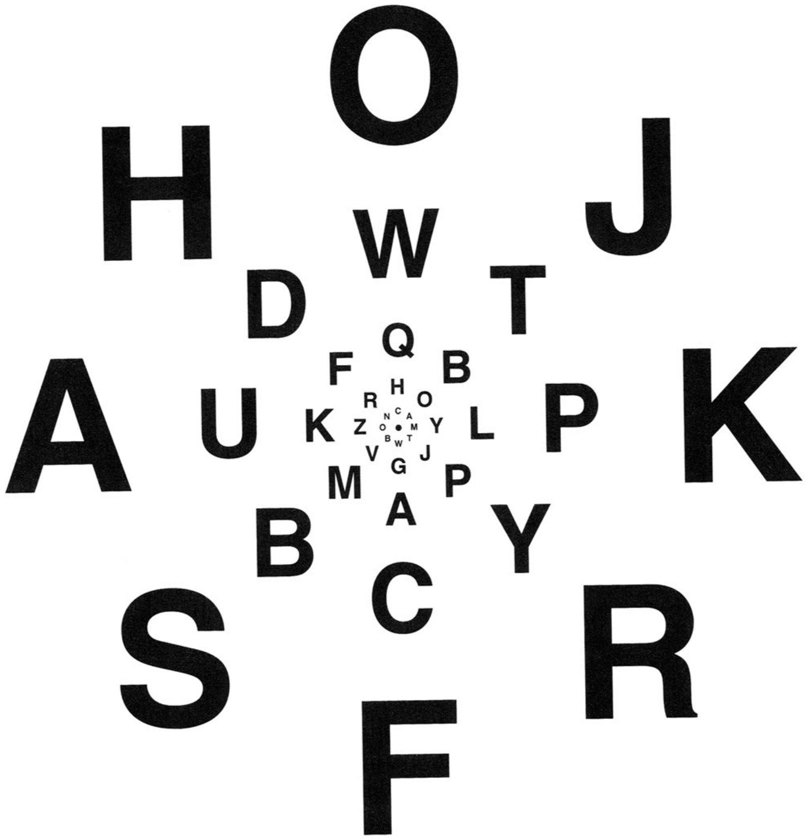


Density of rods and cones

- Rods and cones are *non-uniformly* distributed on the retina
  - Rods responsible for intensity, cones responsible for color
  - **Fovea**: Small region (1 or 2°) at the center of the visual field containing the highest density of cones (and no rods).
  - Less visual acuity in the periphery—many rods wired to the same neuron



Demonstrations of visual acuity



With one eye shut, at the right distance, all of these letters should appear equally legible (Glassner, 1.7).

Demonstrations of visual acuity

With left eye shut, look at the cross on the left. At the right distance, the circle on the right should disappear (Glassner, 1.8).

Brightness contrast and constancy

- The apparent brightness depends on the surrounding region
  - **brightness contrast**: a constant colored region seems lighter or darker depending on the surrounding intensity

  - **brightness constancy**: a surface looks the same under widely varying lighting conditions.

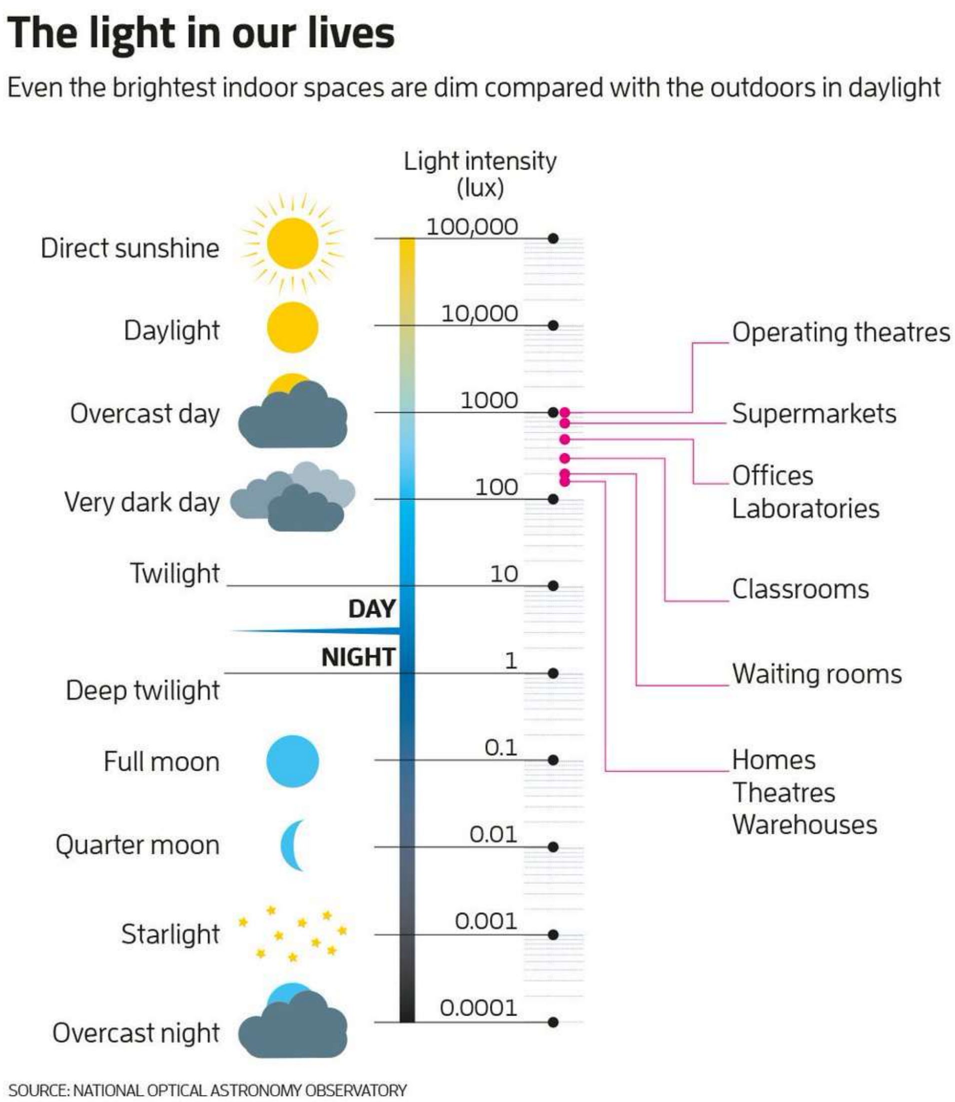
Light response is nonlinear

- Our **visual system** has a **large *dynamic range***
  - We can resolve **both light and dark things at the same time**
  - One mechanism for achieving this is that we <u>**sense light intensity**</u> on a ***logarithmic scale***
    - an exponential intensity ramp will be seen as a linear ramp
  - Another mechanism is <u>***adaptation***</u>
    - rods and cones adapt to be **more sensitive in low light, less sensitive in bright light.**

Visual dynamic range

A piece of white paper can be 1,000,000,000 times brighter in outdoor sunlight than in a moonless night.

BUT in a given lighting condition, light perception ranges over only about two orders of magnitude.



https://threader.app/thread/1134003178515701762

Learning to See in the Dark

Chen Chen, Qifeng Chen, Jia Xu and Vladlen Koltun

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

(a) Camera output with ISO 8,000 (b) Camera output with ISO 409,600 (c) Our result from the raw data of (a)

Figure. Extreme low-light imaging by a Sony a7S II camera using ISO 8000, f/5.6, 1/30 second. Dark indoor environment. The illuminance at the camera is <0.1 lux.

http://cchen156.web.engr.illinois.edu/SID.html

Dancing under the stars: video denoising in starlight

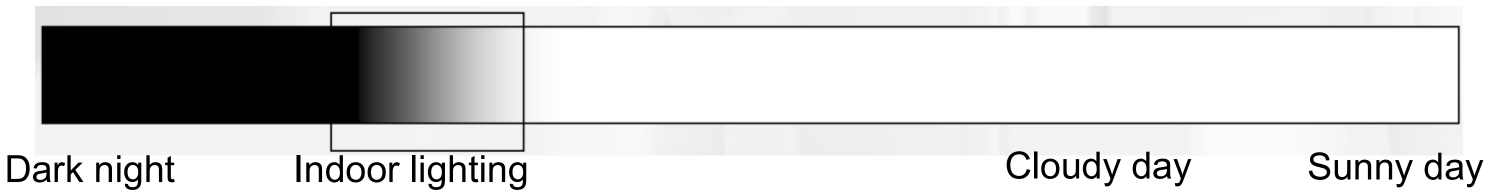
CVPR 2022

Kristina Monakhova — UC Berkeley
Stephan Richter — Intel Labs
Laura Waller — UC Berkeley
Vladlen Koltun — Intel Labs

Visual dynamic range

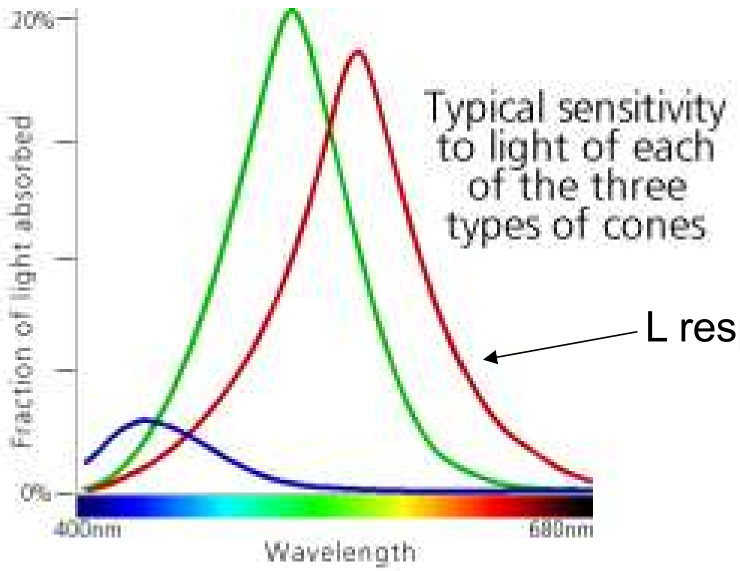
If we were sensitive to this whole range all the time, we wouldn't be able to discriminate lightness levels in a typical scene.

The visual system solves this problem by restricting the 'dynamic range' of its response to match the current overall or 'ambient' light level.



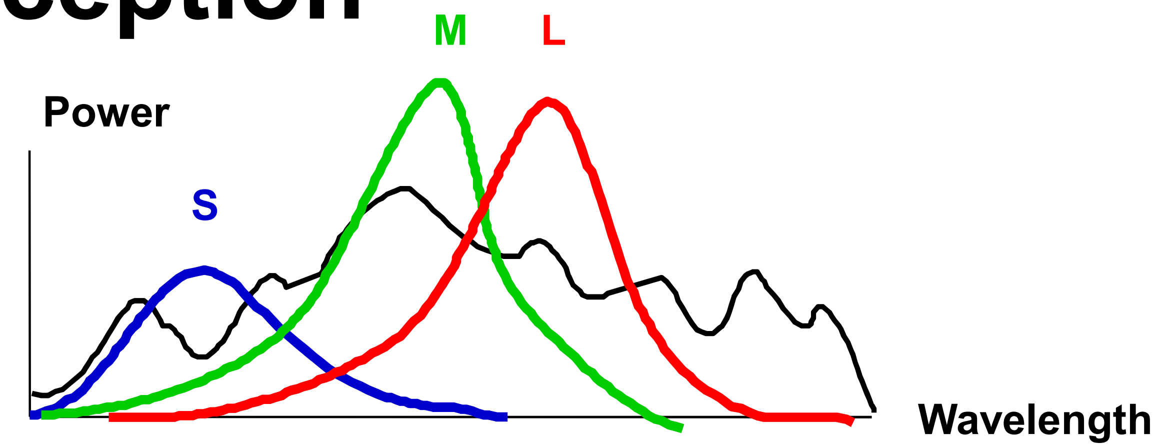
Color perception

- Three types of cones
  - Each is sensitive in a different region of the spectrum
    - but regions overlap
    - **Short (S) corresponds to blue**
    - **Medium (M) corresponds to green**
    - **Long (L) corresponds to red**
  - Different sensitivities: we are more sensitive to green than red
    - varies from person to person (and with age)
  - Colorblindness—deficiency in at least one type of cone



L response curve

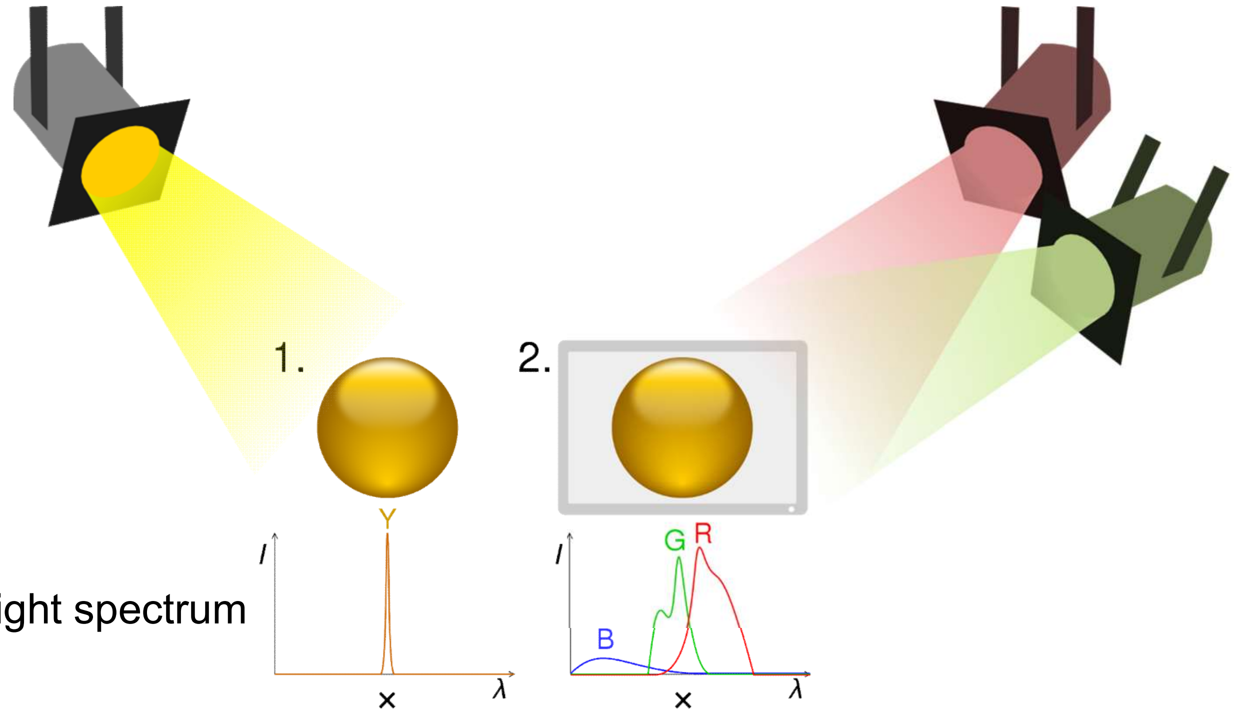
Color perception



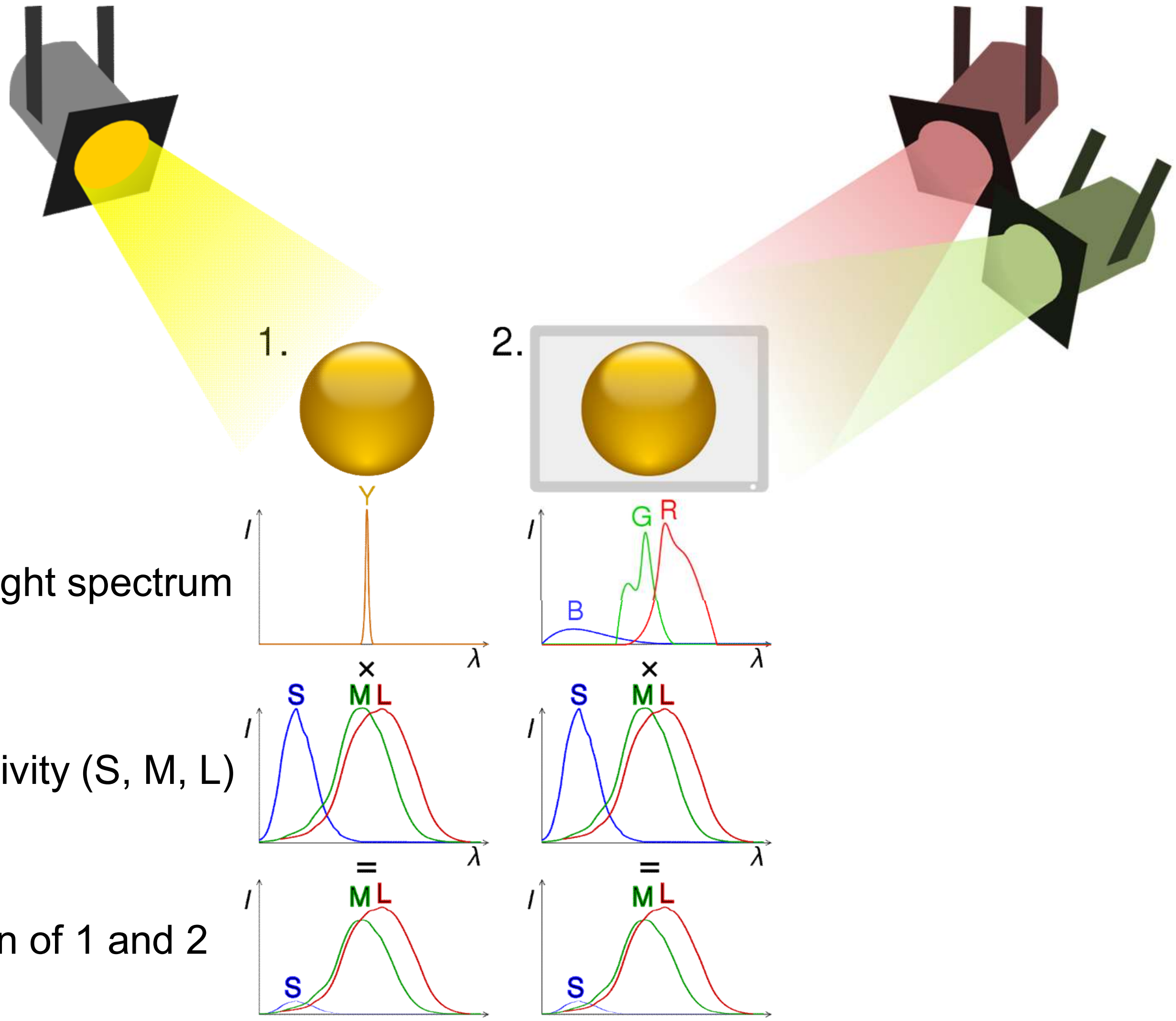
- Rods and cones act as filters on the spectrum
  - To get the output of a filter, multiply its response curve by the spectrum, integrate over all wavelengths
    - Each cone yields one number
  - Q: How can we represent an entire spectrum with 3 numbers?
  - A: We can't! Most of the information is lost
    - As a result, two different spectra may appear indistinguishable
      - such spectra are known as **metamers**

Metamers

1. Combined light spectrum



Metamers

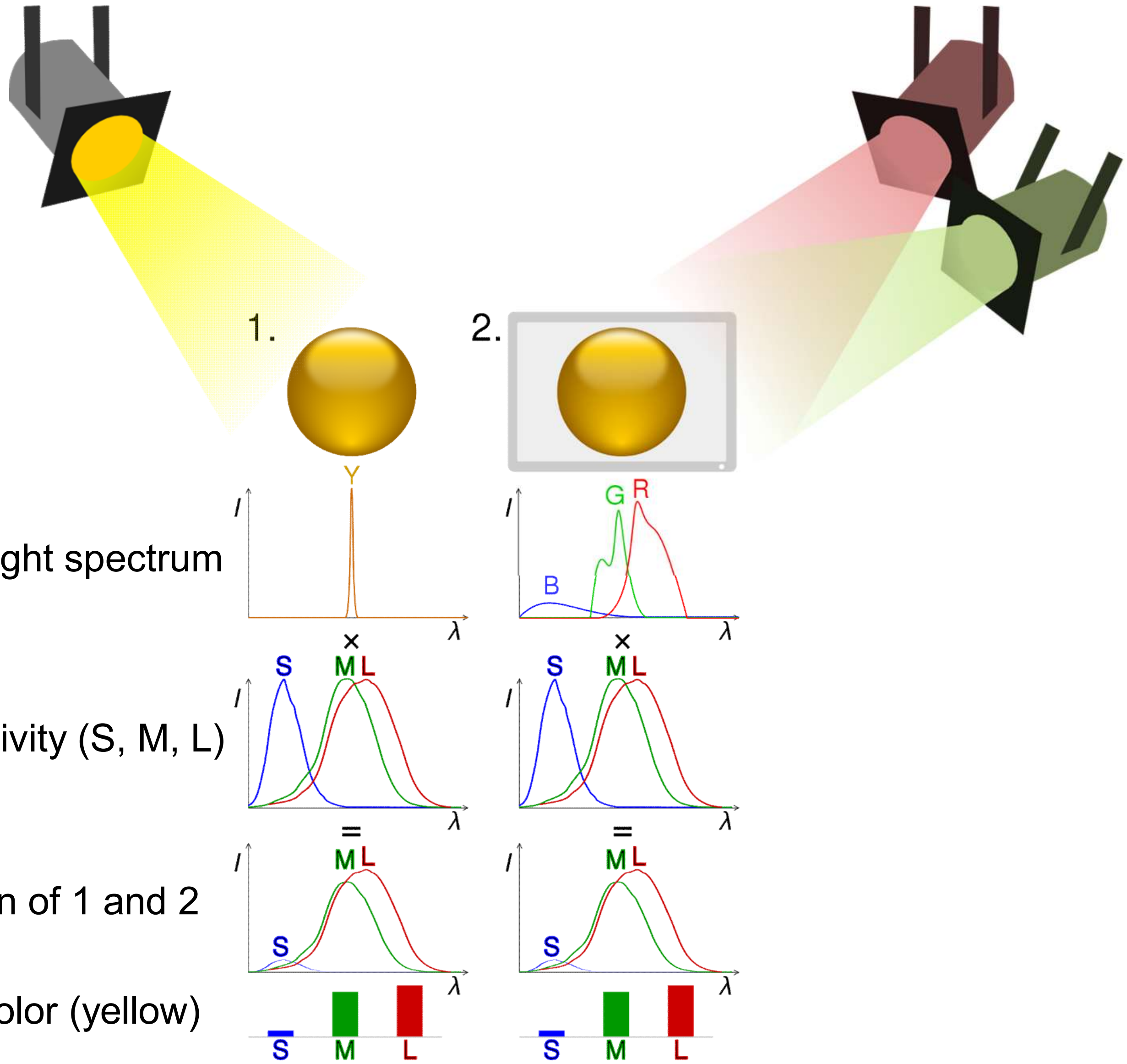


1. Combined light spectrum

2. Cone sensitivity (S, M, L)

3. Multiplication of 1 and 2

Metamers



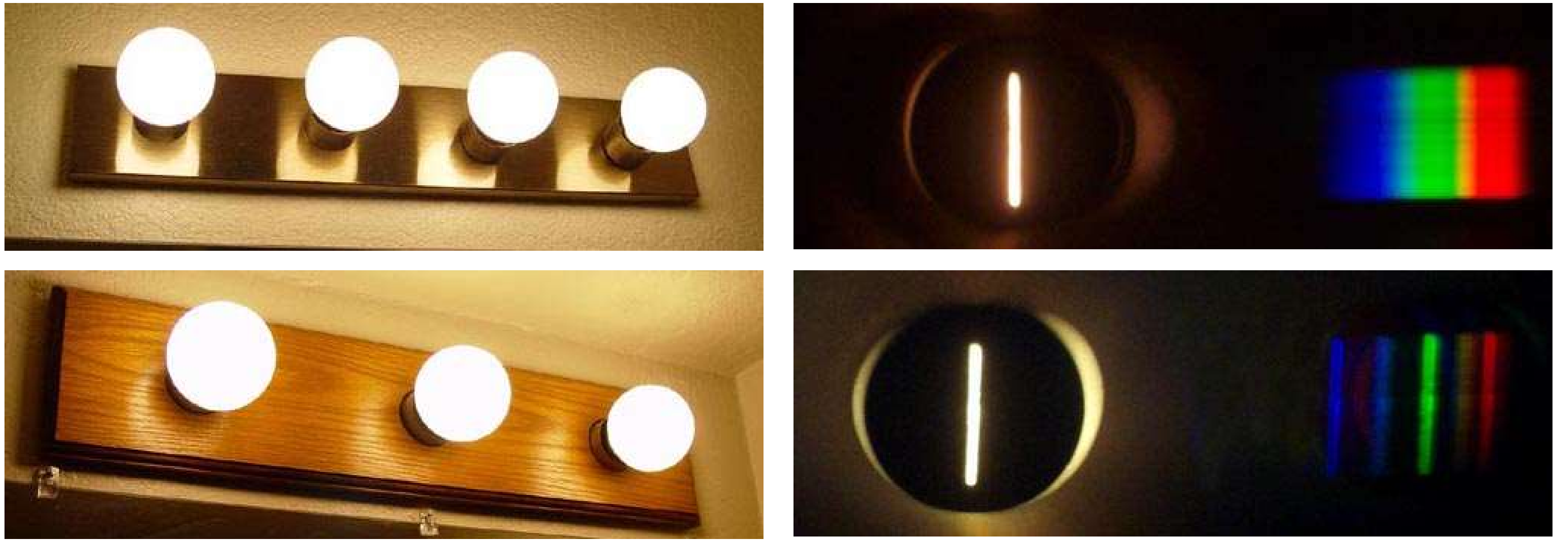
1. Combined light spectrum

2. Cone sensitivity (S, M, L)

3. Multiplication of 1 and 2

4. Observed color (yellow)

What kind of bulb is it?



http://www.chemistryland.com/CHM107Lab/Exp7/Spectroscope/Spectroscope.html

What color is the dress?

- White and gold?
- Black and blue?

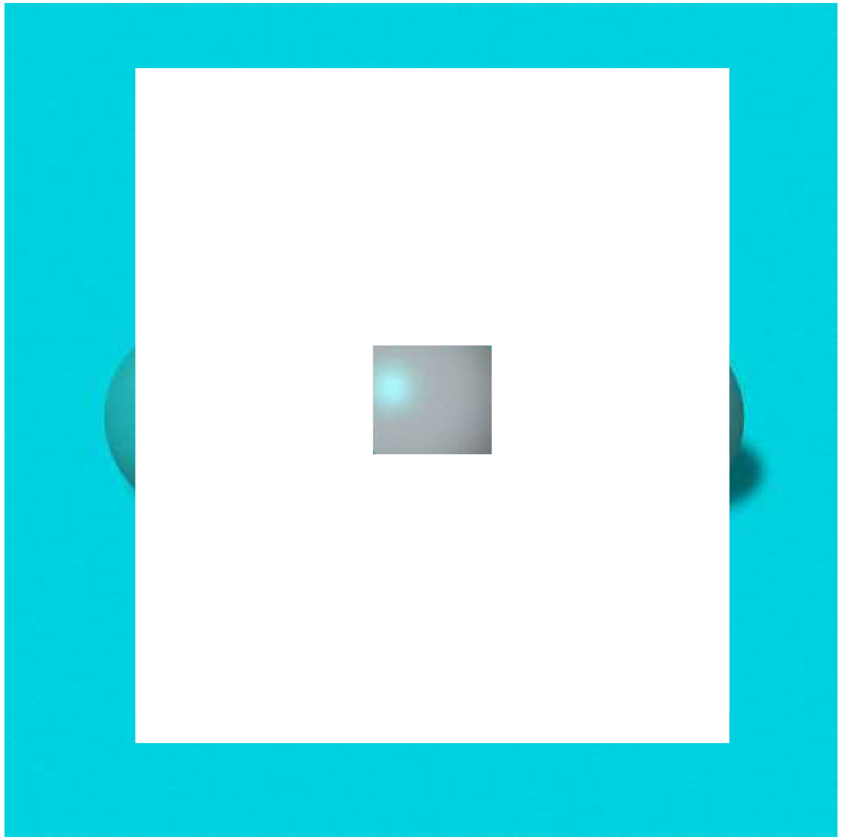
slido

What color is the dress?

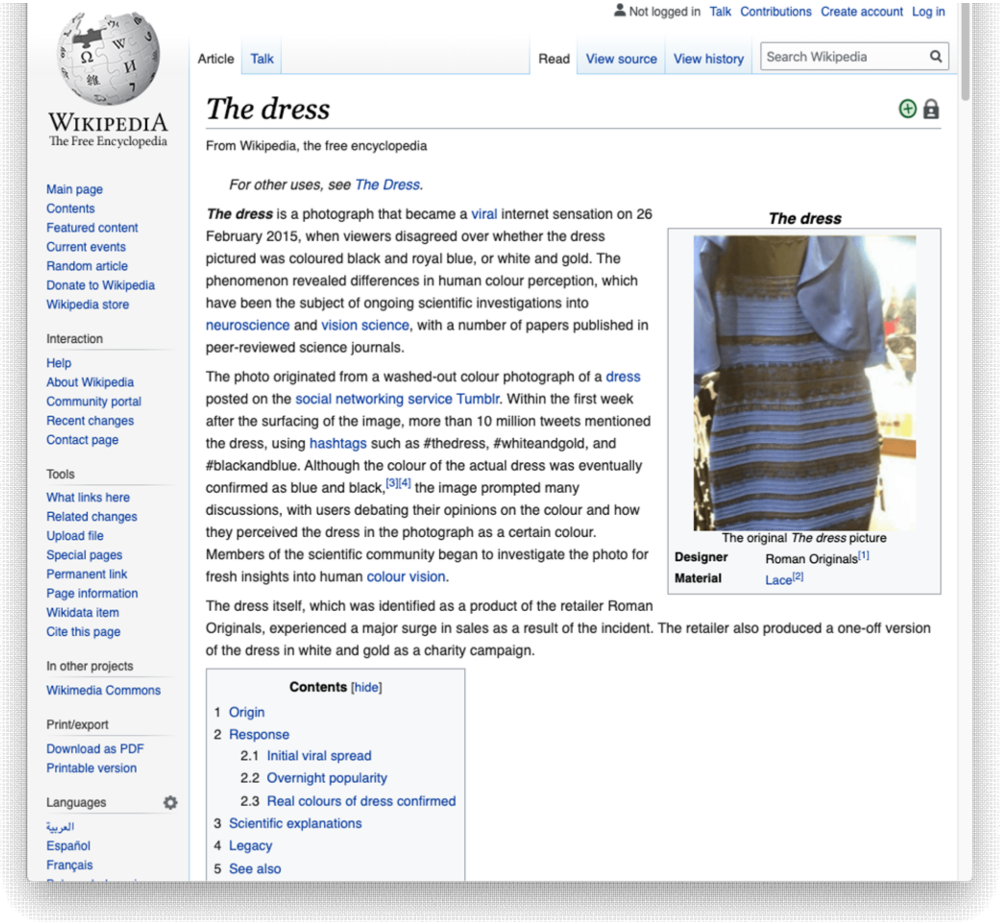
ⓘ Start presenting to display the poll results on this slide.

What color is the center ball?

What color is the center ball?



Reflectance and Illumination In Popular Culture…



Not logged in Talk Contributions Create account Log in

Article Talk Read View source View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

In other projects
Wikimedia Commons

Print/export
Download as PDF
Printable version

Languages
العربية
Español
Français

The dress

From Wikipedia, the free encyclopedia

For other uses, see The Dress.

The dress is a photograph that became a viral internet sensation on 26 February 2015, when viewers disagreed over whether the dress pictured was coloured black and royal blue, or white and gold. The phenomenon revealed differences in human colour perception, which have been the subject of ongoing scientific investigations into neuroscience and vision science, with a number of papers published in peer-reviewed science journals.

The photo originated from a washed-out colour photograph of a dress posted on the social networking service Tumblr. Within the first week after the surfacing of the image, more than 10 million tweets mentioned the dress, using hashtags such as #thedress, #whiteandgold, and #blackandblue. Although the colour of the actual dress was eventually confirmed as blue and black,[3][4] the image prompted many discussions, with users debating their opinions on the colour and how they perceived the dress in the photograph as a certain colour. Members of the scientific community began to investigate the photo for fresh insights into human colour vision.

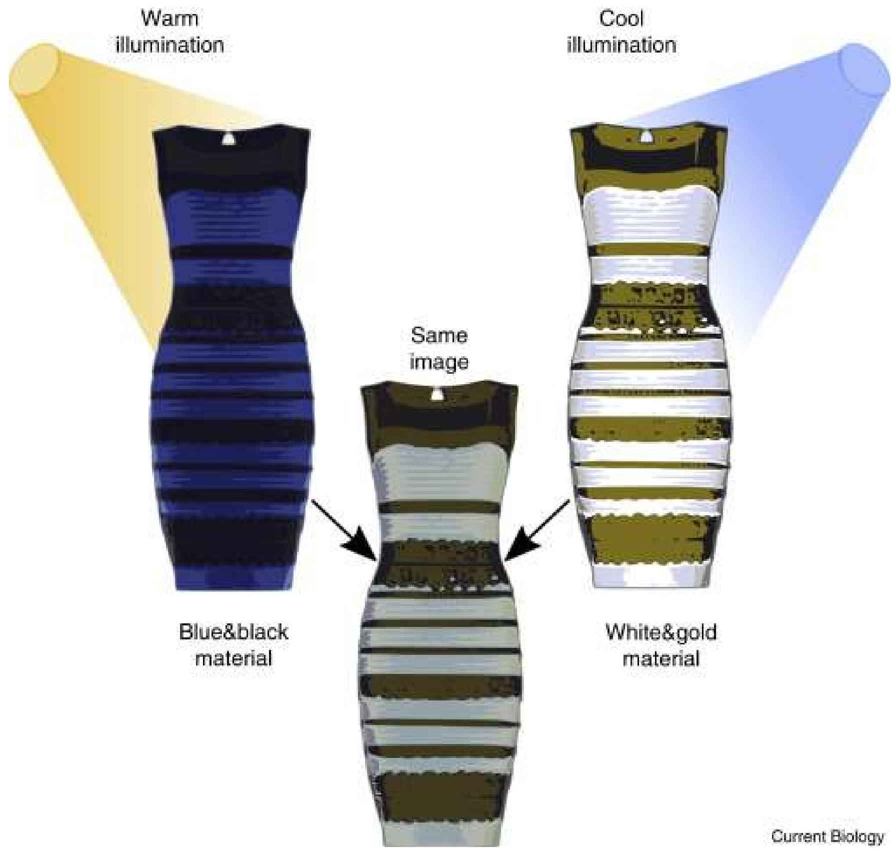
The dress

The original *The dress* picture

Designer	Roman Originals[1]
Material	Lace[2]

The dress itself, which was identified as a product of the retailer Roman Originals, experienced a major surge in sales as a result of the incident. The retailer also produced a one-off version of the dress in white and gold as a charity campaign.

Contents [hide]



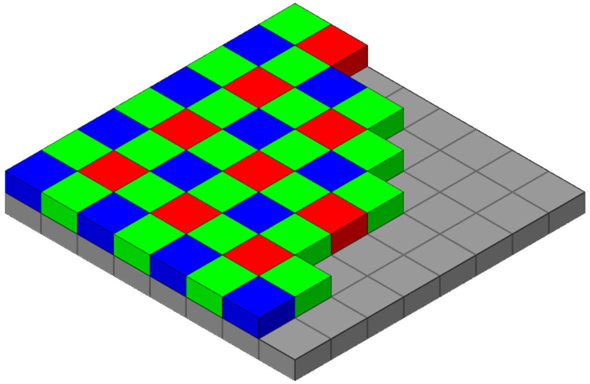
Perception summary

- The mapping from radiance to perceived color is quite complex!
  - We throw away most of the data
  - We apply a logarithm
  - Brightness affected by pupil size and adaptation of rods/cones
  - Brightness contrast and constancy effects

- The same is true for cameras
  - But we have tools to correct for these effects
    - (Computational Photography)

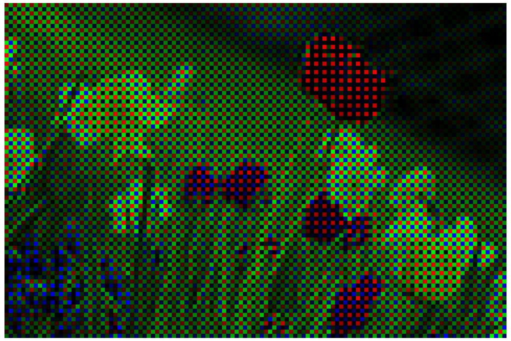
Cameras also see color

- Common technique is to place a mosaic of color filters (a *Bayer filter*) in front of the sensor



Bayer filter pattern in front of sensor

- Colors are interpolated to create a full-resolution "*demosaicked*" color image



What the camera sees
("raw" image)

Demosaicked image

https://en.wikipedia.org/wiki/Bayer_filter

Early color photography

- Prior to the invention of color film, Sergey Prokudin-Gorsky took three separate exposures with three different color filters

Blue, Green,
Red exposures

Combined color image (1911)

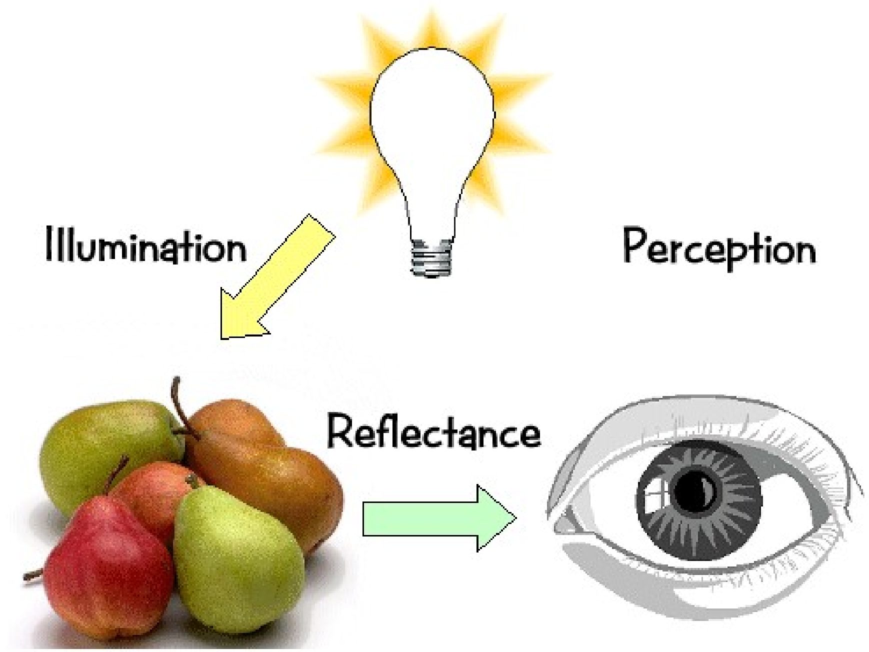
Film has its own sensitivity

- "… the film of Lincoln's era was sensitive only to blue and UV light, causing cheeks to appear dark, and overly emphasizing wrinkles by filtering out skin subsurface scatter which occurs mostly in the red channel. Hence, the deep lines and sharp creases that we associate with Lincoln's face are likely exaggerated by the photographic process of the time."

Time-Travel Rephotography
Xuan Luo, Xuaner Zhang, Paul Yoo, Ricardo Martin-Brualla, Jason Lawrence, Steven M. Seitz
SIGGRAPH Asia 2021

Questions?

Light transport



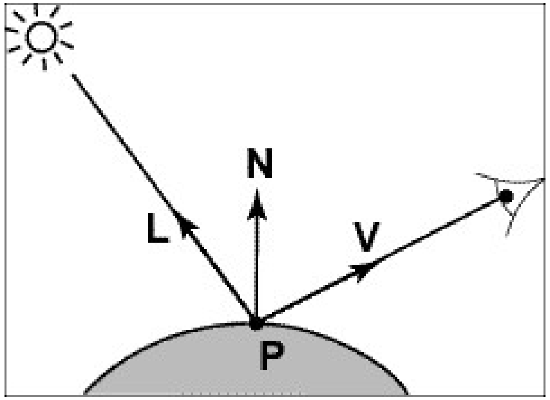
Light sources

- Basic types
  - point source
  - directional source
    - a point source that is infinitely far away
  - area source
    - a union of point sources
- More generally
  - a light field can describe *any* distribution of light sources
- What happens when light hits an object?

Modeling Image Formation

We need to reason about:

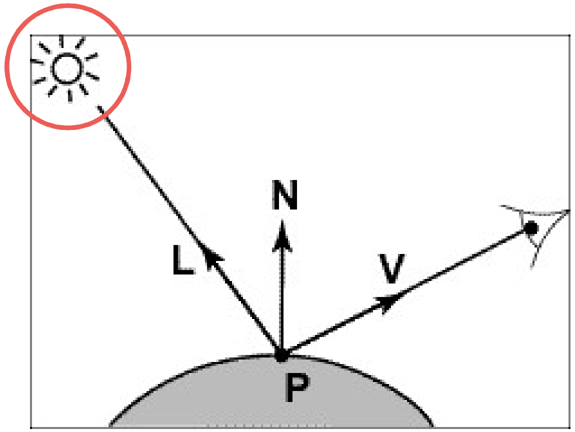
- How light interacts with the scene
- How a pixel value is related to light energy in the world



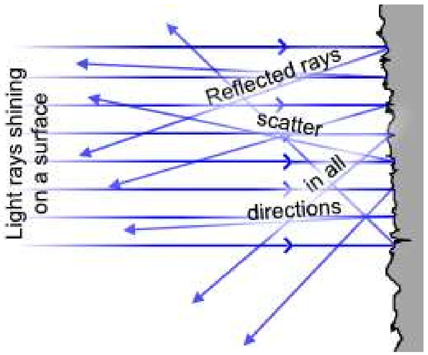
Track a “ray” of light all the way from light source to the sensor

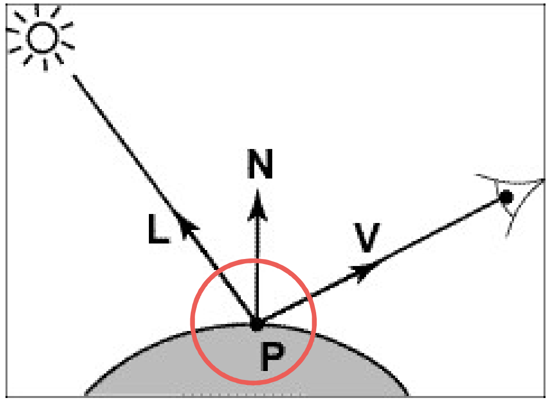
Directional Lighting

- Key property: all rays are parallel
- Equivalent to an infinitely distant point source



Lambertian Reflectance



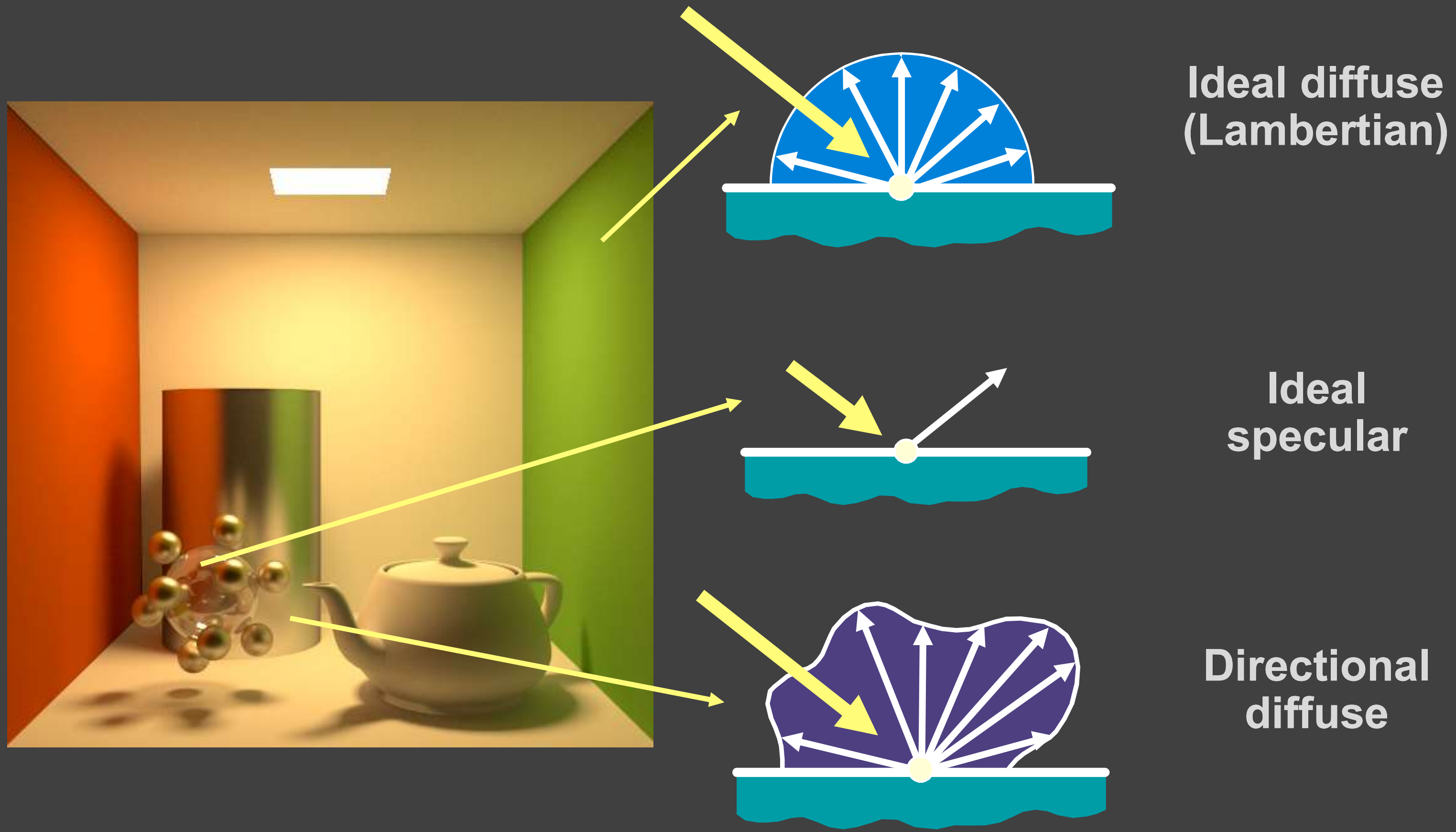


$$I = N \cdot L$$

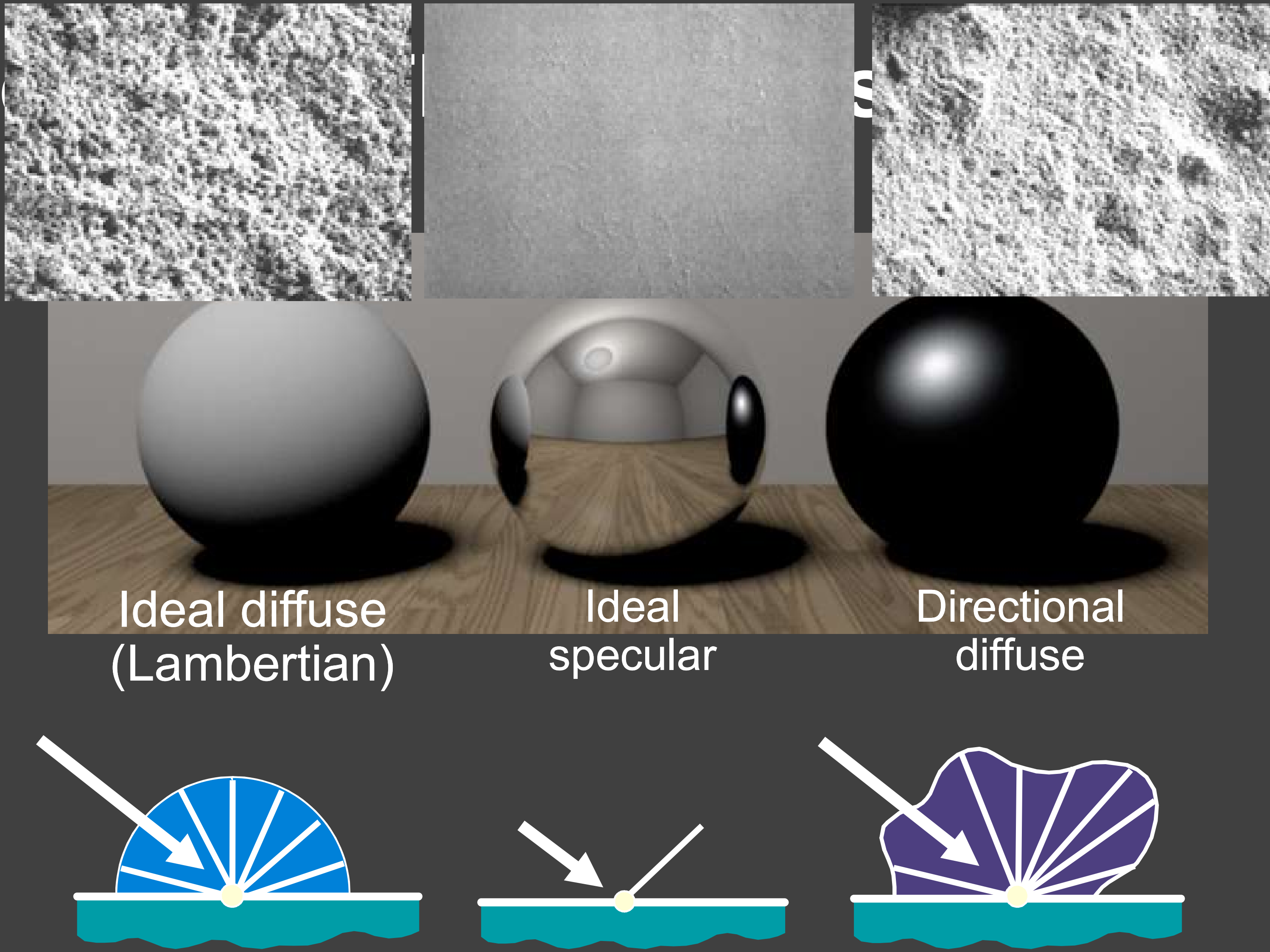
Image intensity	$=$	Surface normal	$\cdot$	Light direction

Image intensity $\propto$ cos(angle between N and L)

Materials - Three Forms

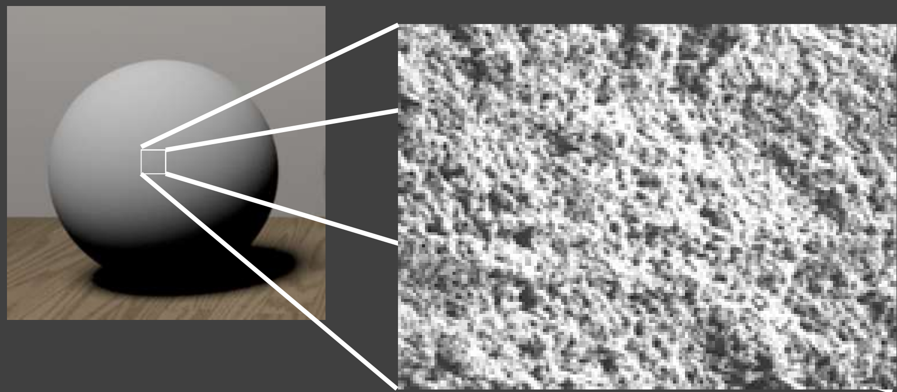


Refle

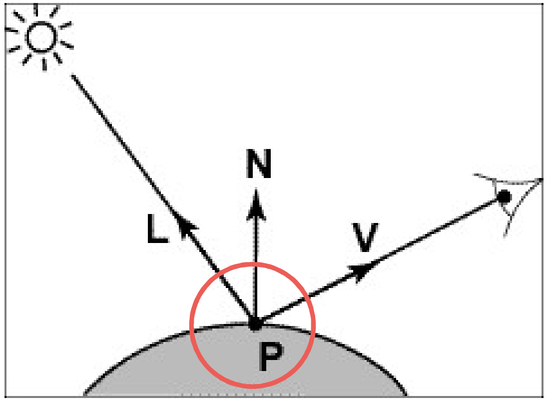

Ideal diffuse
(Lambertian)
Ideal
specular
Directional
diffuse

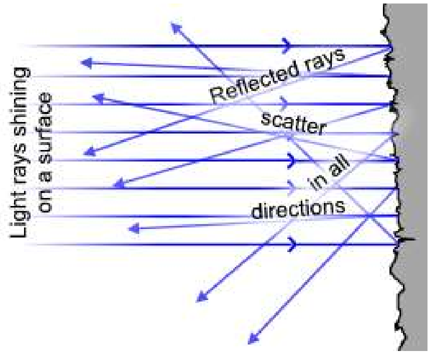
Ideal Diffuse Reflection

- Characteristic of multiple scattering materials
- An idealization but reasonable for matte surfaces



Lambertian Reflectance



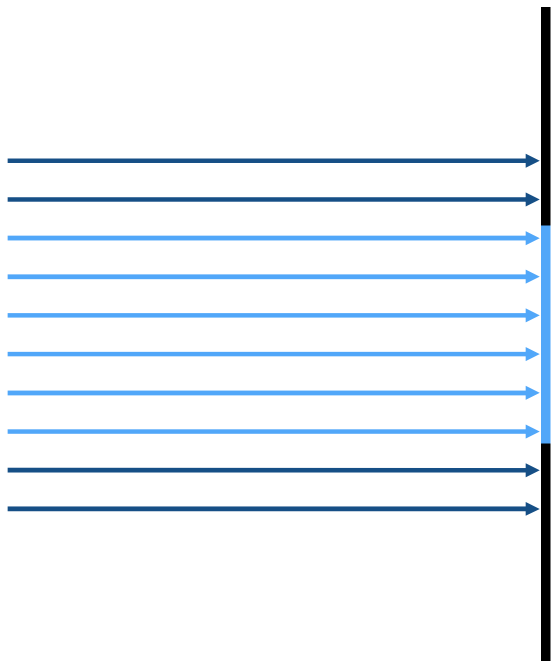


1. Reflected energy is proportional to cosine of angle between L and N **(incoming)**

2. Measured intensity is viewpoint-independent **(outgoing)**

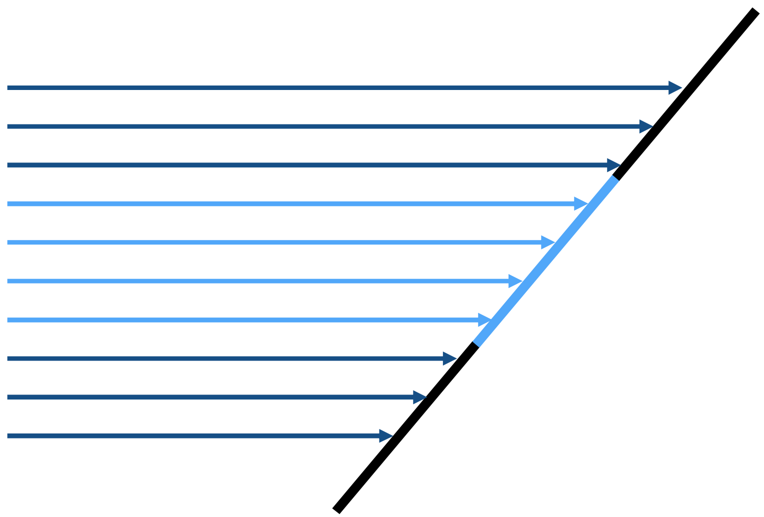
Lambertian Reflectance: Incoming

- Reflected energy is proportional to cosine of angle between L and N



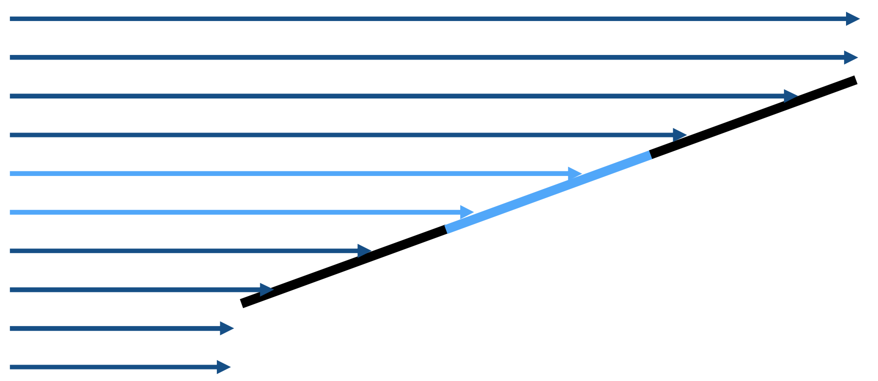
Lambertian Reflectance: Incoming

- Reflected energy is proportional to cosine of angle between L and N



Lambertian Reflectance: Incoming

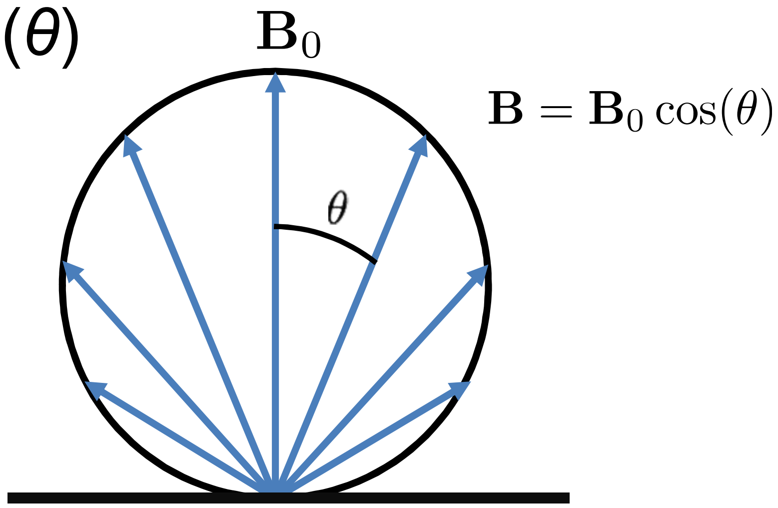
- Reflected energy is proportional to cosine of angle between L and N



Light hitting surface is proportional to the **cosine**

Lambertian appearance is view-independent

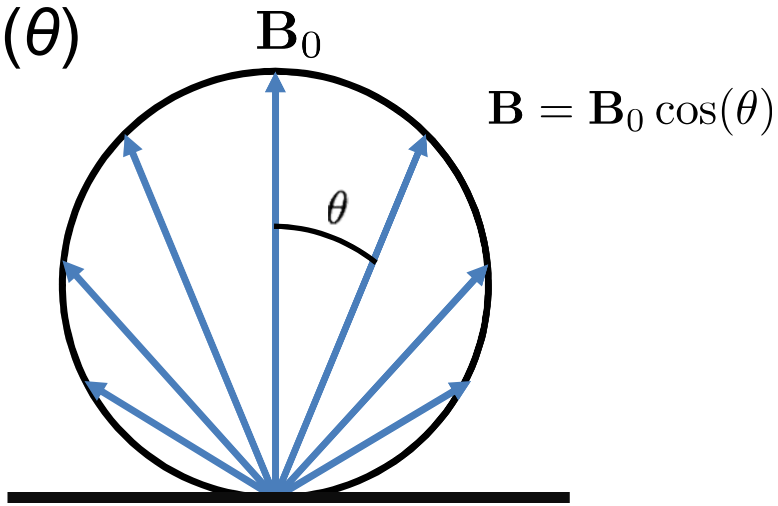
- Number of photons reflected to a given angle θ is proportional to cos(θ)



Lambert's cosine law: $\mathbf{B} = \mathbf{B}_0 \cos(\theta)$

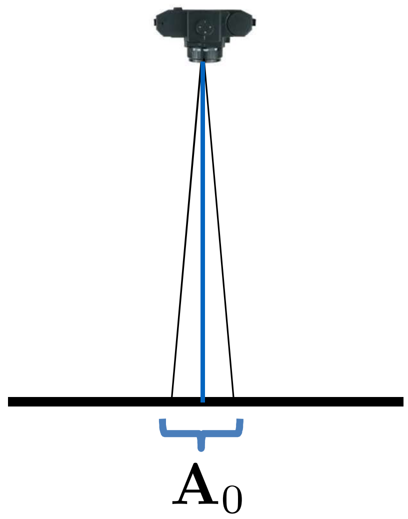
Lambertian appearance is view-independent

- Number of photons reflected to a given angle θ is proportional to $\cos(\theta)$



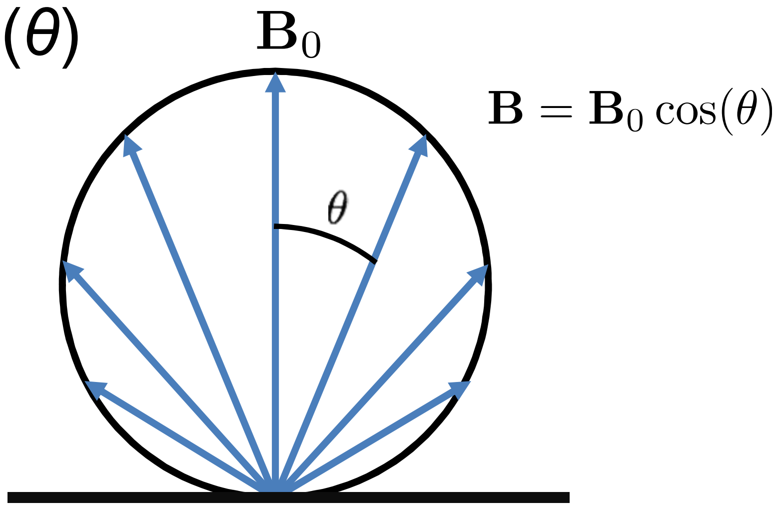
Lambert's cosine law: $\mathbf{B} = \mathbf{B}_0 \cos(\theta)$

- But appearance is the same from every angle due to larger pixel footprint at larger angles



Lambertian appearance is view-independent

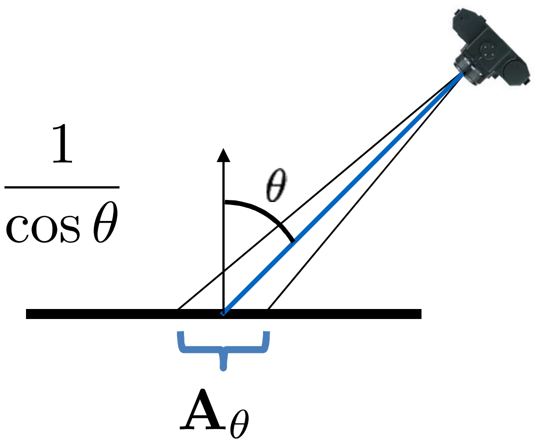
- Number of photons reflected to a given angle θ is proportional to $\cos(\theta)$



Lambert's cosine law: $\mathbf{B} = \mathbf{B}_0 \cos(\theta)$

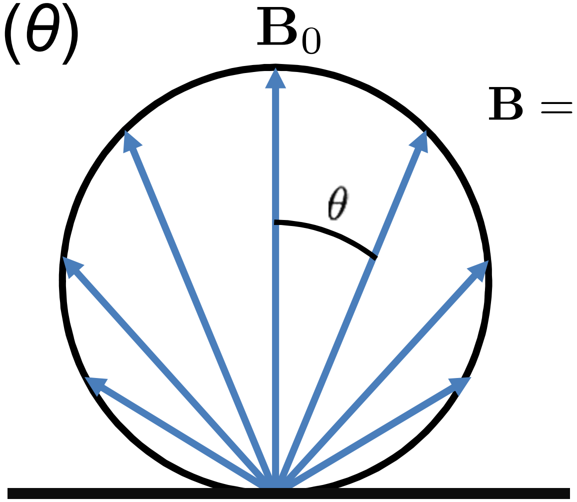
- But appearance is the same from every angle due to larger pixel footprint at larger angles

$$\mathbf{A}_\theta \propto \mathbf{A}_0 \frac{1}{\cos\theta}$$



Lambertian appearance is view-independent

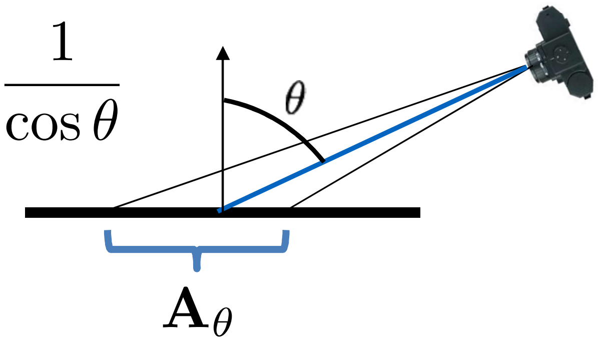
- Number of photons reflected to a given angle θ is proportional to $\cos(\theta)$



Lambert's cosine law: $\mathbf{B} = \mathbf{B}_0 \cos(\theta)$

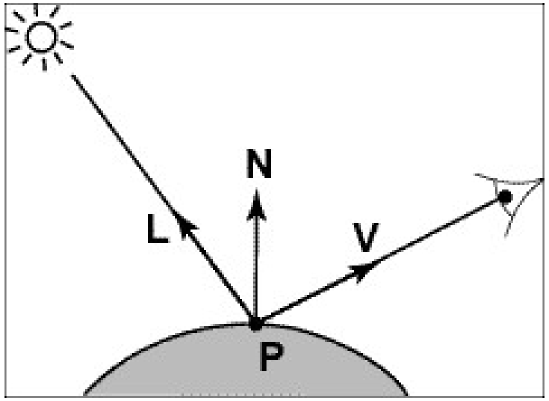
- But appearance is the same from every angle due to larger pixel footprint at larger angles

$$\mathbf{A}_\theta \propto \mathbf{A}_0 \frac{1}{\cos\theta}$$



Radiance (what eye sees) $\propto \mathbf{B}_0 \mathbf{A}_0 \cancel{\cos(\theta)} \frac{1}{\cancel{\cos(\theta)}}$

Final Lambertian image formation model



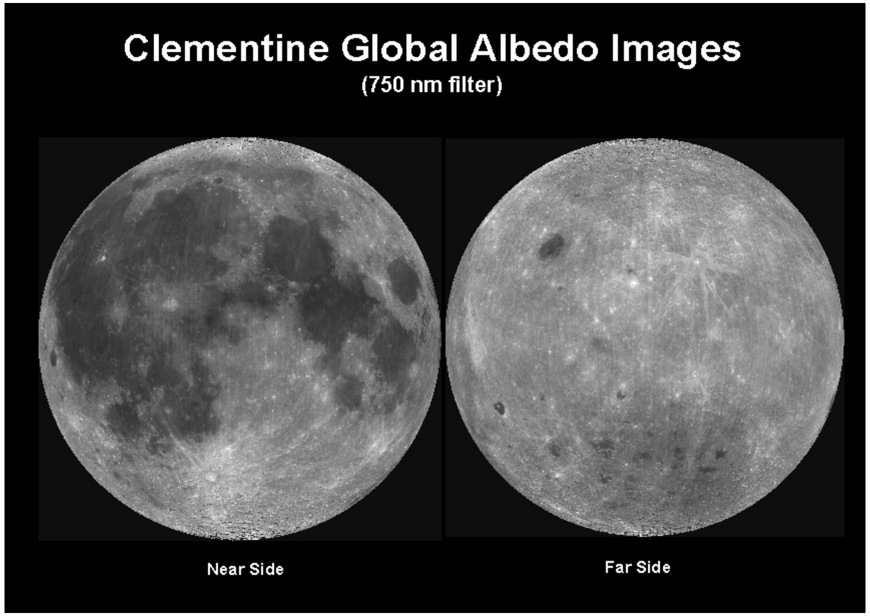
$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

1. Diffuse **albedo**: what fraction of incoming light is reflected?
   - Introduce scale factor k_d
2. Light intensity: how much light is arriving?
   - Compensate with camera exposure (global scale factor)
3. Camera response function
   - Assume pixel value is linearly proportional to incoming energy (perform radiometric calibration if not)

Albedo

Sample albedos

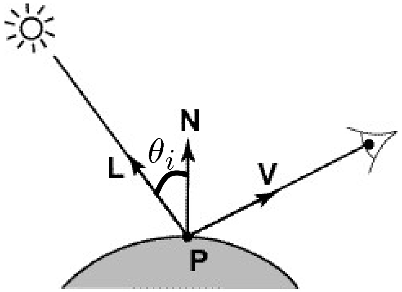
Surface	Typical albedo
Fresh asphalt	0.04[4]
Open ocean	0.06[5]
Worn asphalt	0.12[4]
Conifer forest (Summer)	0.08,[6] 0.09 to 0.15[7]
Deciduous trees	0.15 to 0.18[7]
Bare soil	0.17[8]
Green grass	0.25[8]
Desert sand	0.40[9]
New concrete	0.55[8]
Ocean ice	0.5–0.7[8]
Fresh snow	0.80–0.90[8]



Objects can have varying albedo and albedo varies with wavelength

Source: https://en.wikipedia.org/wiki/Albedo

A Single Image: Shape from shading



Suppose (for now) $k_d = 1$

$$
\begin{aligned}
I &= k_d \mathbf{N} \cdot \mathbf{L} \\
&= \mathbf{N} \cdot \mathbf{L} \\
&= cos\ \theta_i
\end{aligned}
$$

You can directly measure angle between normal and light source

- Not quite enough information to compute surface shape
- But can be if you add some additional info, for example
  - assume a few of the normals are known (e.g., along silhouette)
  - constraints on neighboring normals—"integrability"
  - smoothness
- Hard to get it to work well in practice
  - plus, how many real objects have constant albedo?
  - But, deep learning can help

https://www.good.is/optical-illusion-plates-and-bowls-upside-down-or-not

Application: Detecting composite photos

Fake photo

Real photo

Questions?

FEATURES & BENIFITS

- No coding required.
- No queries to learn.
- Actionable insights in seconds.

"Save time, reduce costs, and focus on decisions—not processes."