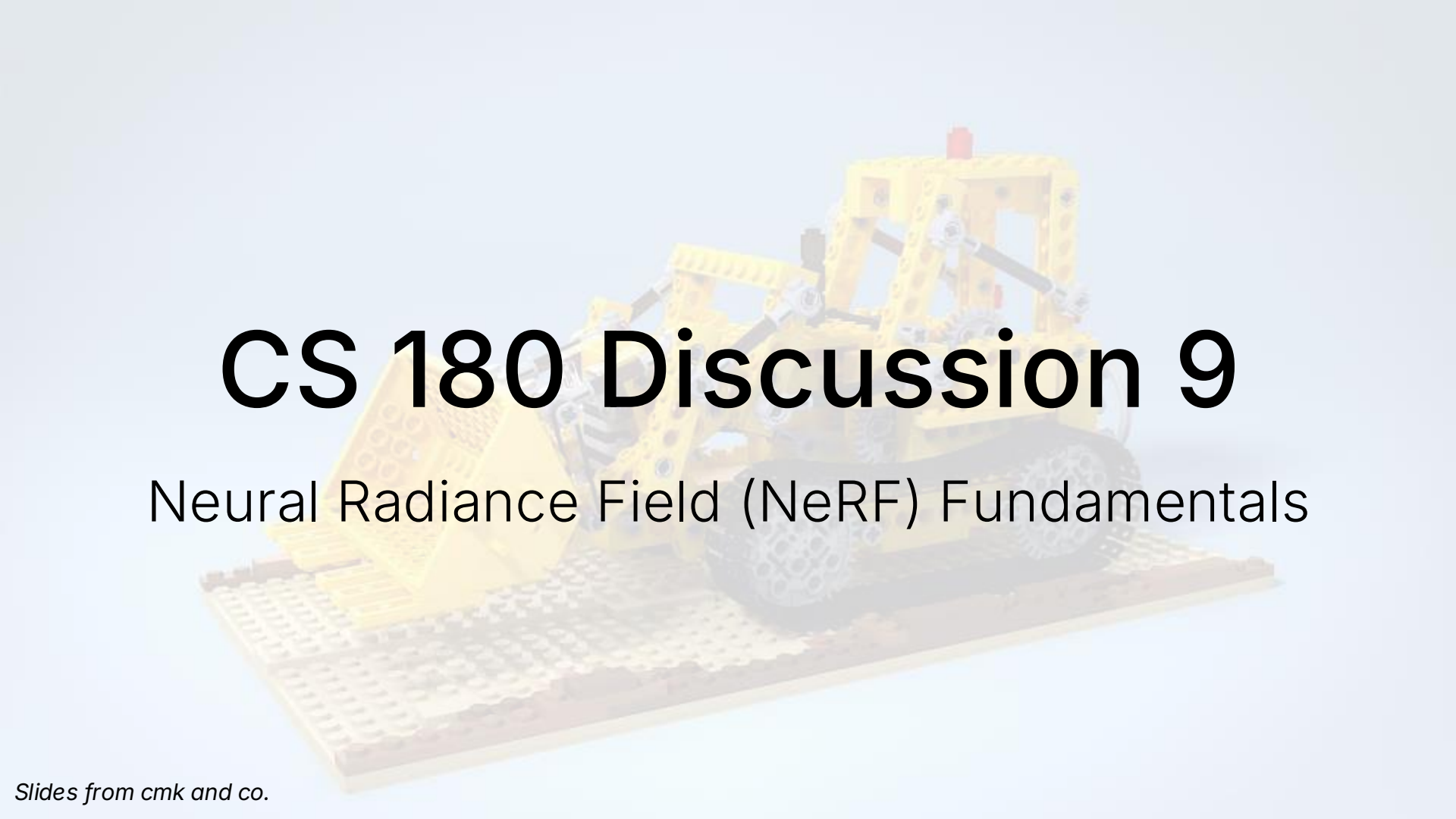


CS 180 Discussion 9

Neural Radiance Field (NeRF) Fundamentals

Slides from cmk and co.

Agenda

1. What is a NeRF “neural radiance fields”?
2. How do we train a NeRF?
    a. Positional encoding
    b. Volumetric rendering

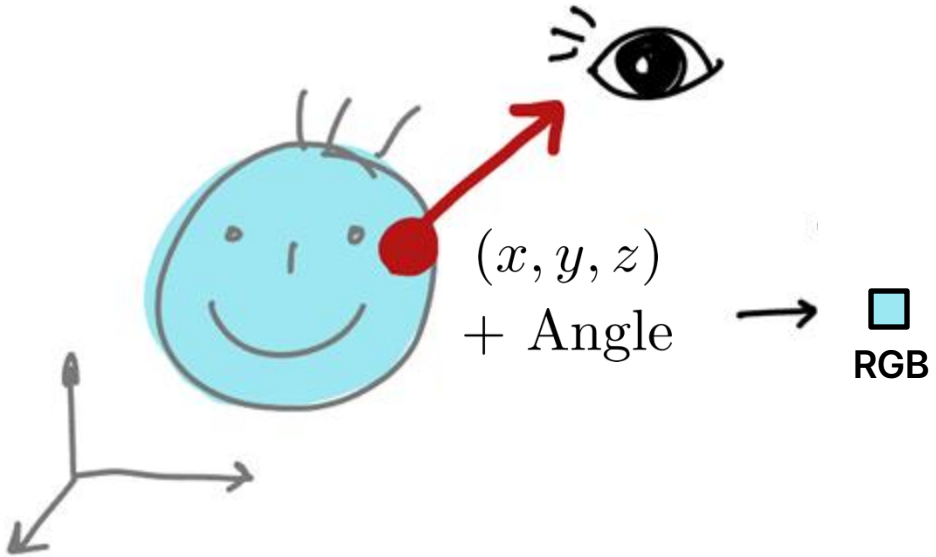
What is a NeRF?

“Neural Radiance Fields”

What is a NeRF?

"Neural Radiance Fields"

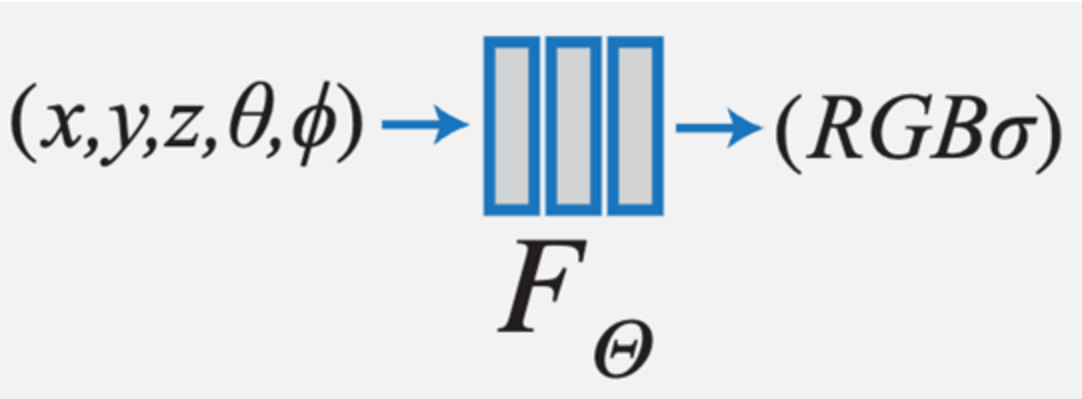
- Radiance Fields: point (+ viewing angle) -> color



What is a NeRF?

"Neural Radiance Fields"

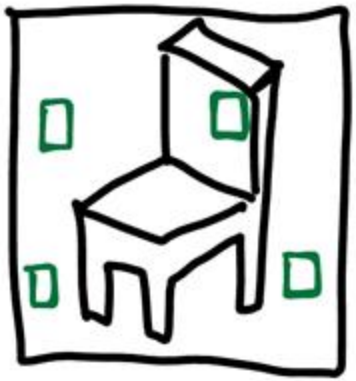
- Radiance Fields: point (+ viewing angle) -> color
- Neural: use neural networks to learn this mapping
  - Supervised learning: (input) -> (output: predicted value) <-> (actual value)



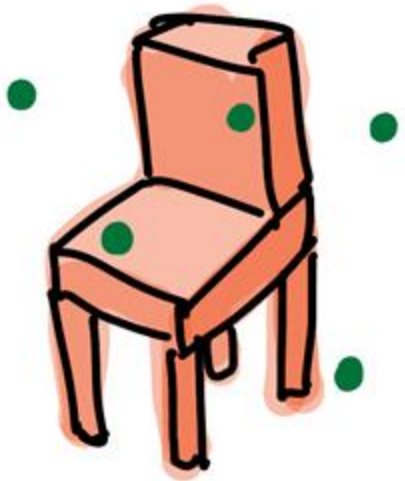
How do we train a NeRF?

- We can learn neural images (2D) or neural radiance fields (3D)

Supervise 2D pixels



Supervise 3D points



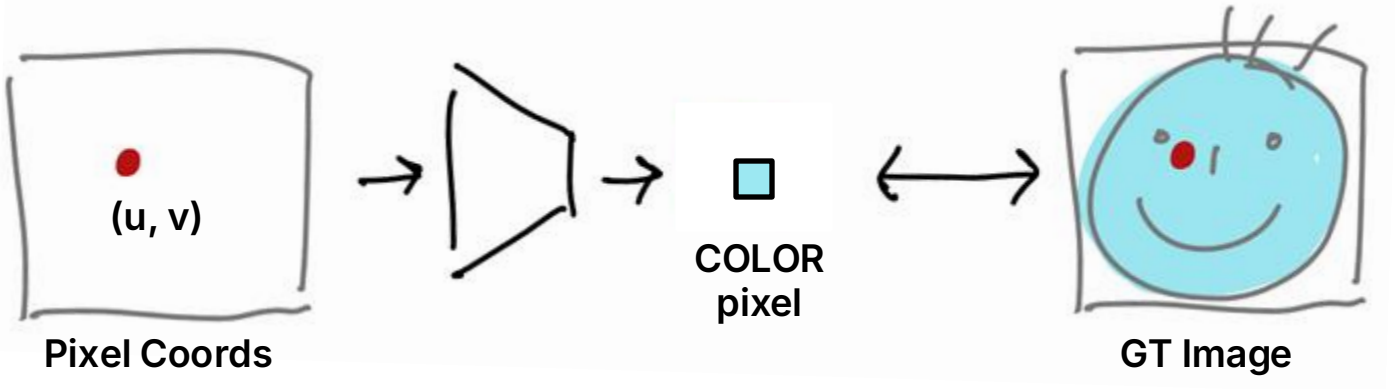
Training a neural image

Input: pixel coordinates (u, v)

Output: color (RGB)

Ray supervision :-)

$$(u, v) \rightarrow \text{predicted} \leftrightarrow \text{actual}$$



Training a neural image

Input: pixel coordinates (u, v)

Output: color (RGB)

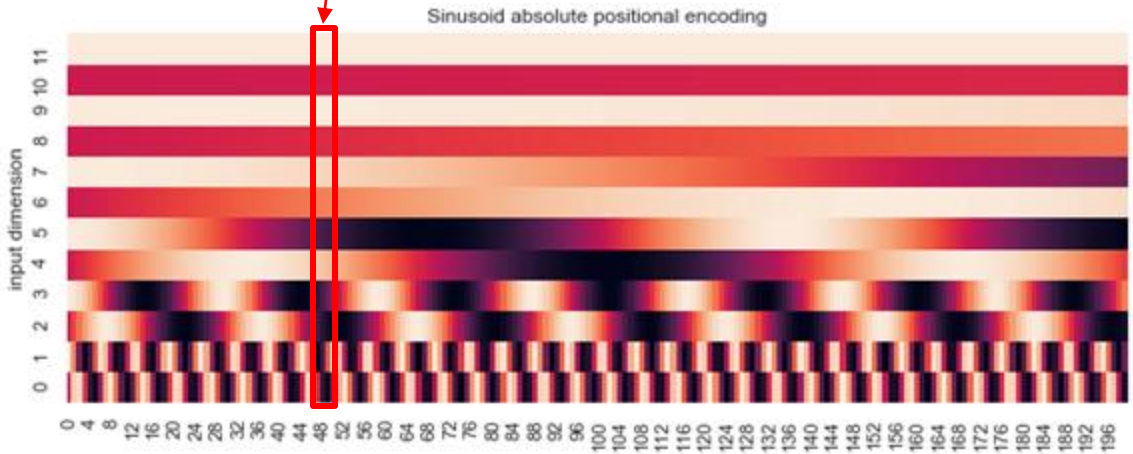
Does it work well?

Positional Encoding

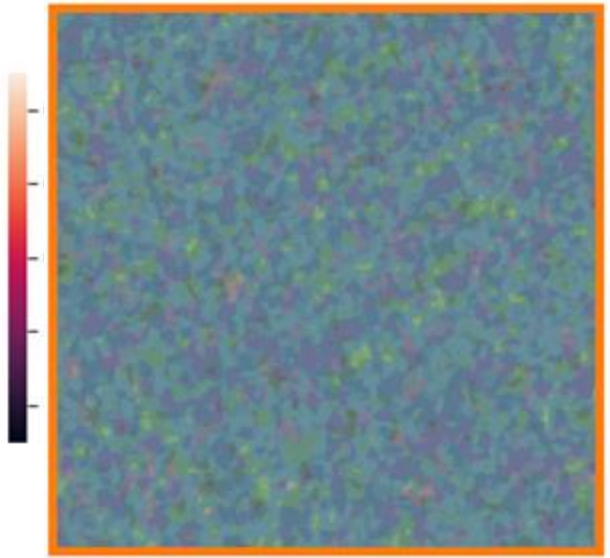
from 16 to 20

$$(u, v) \rightarrow [PE(u), PE(v)] \rightarrow \text{NN}$$

Helps model sharp changes (why?)

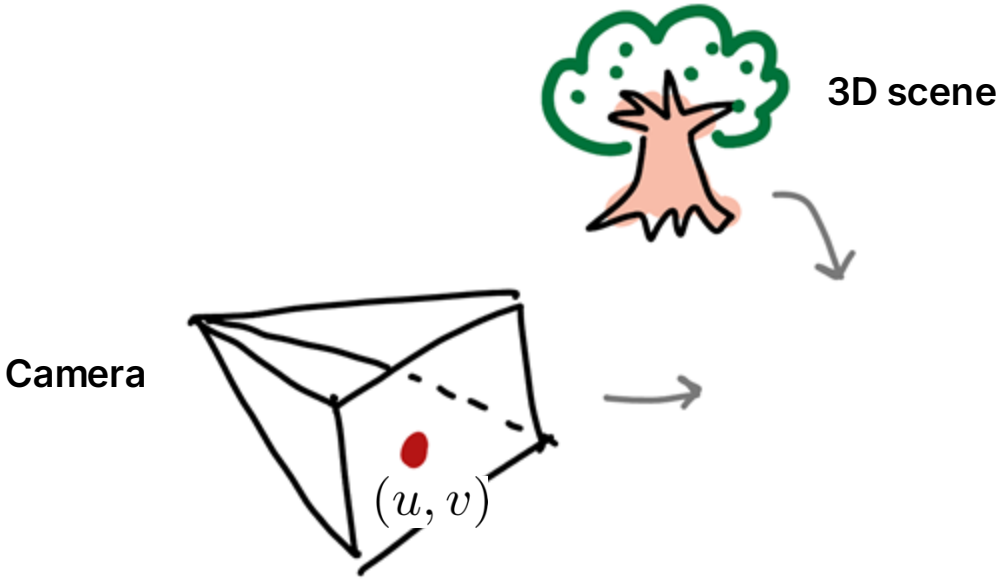


With PE

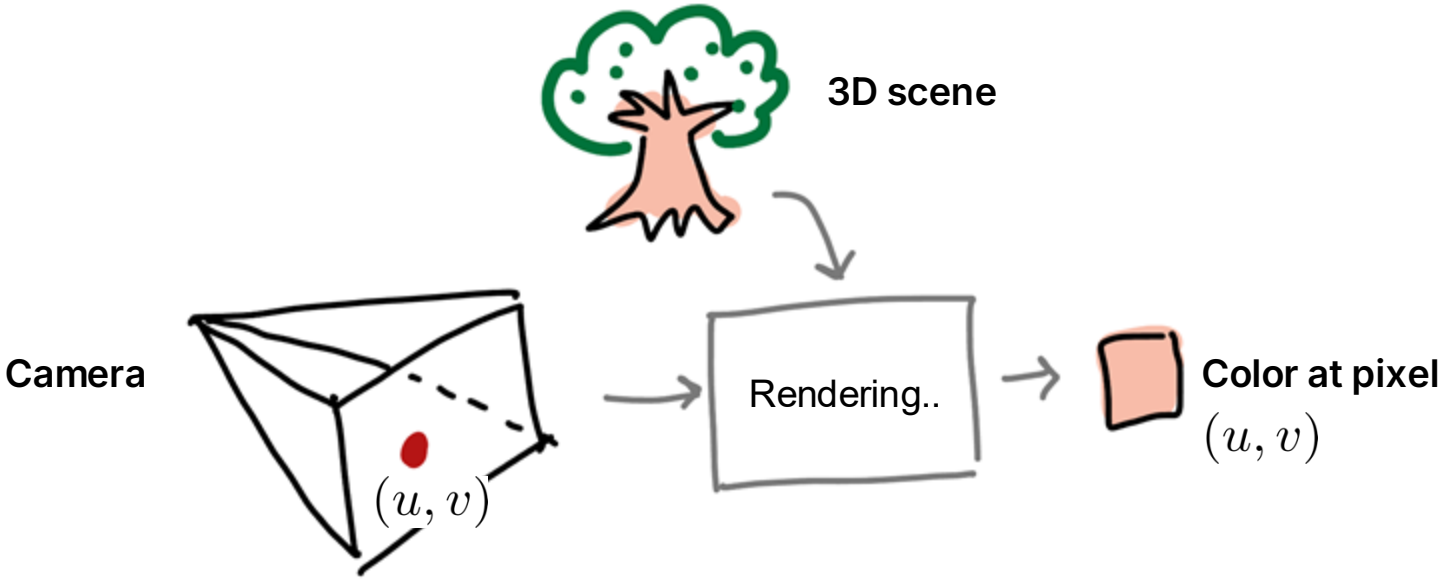


Training NeRF

Volumetric Rendering (Philosophy-level)



Volumetric Rendering (Philosophy-level)



Training a NeRF

Input: (camera) + (u,v)

Output: color (RGB)

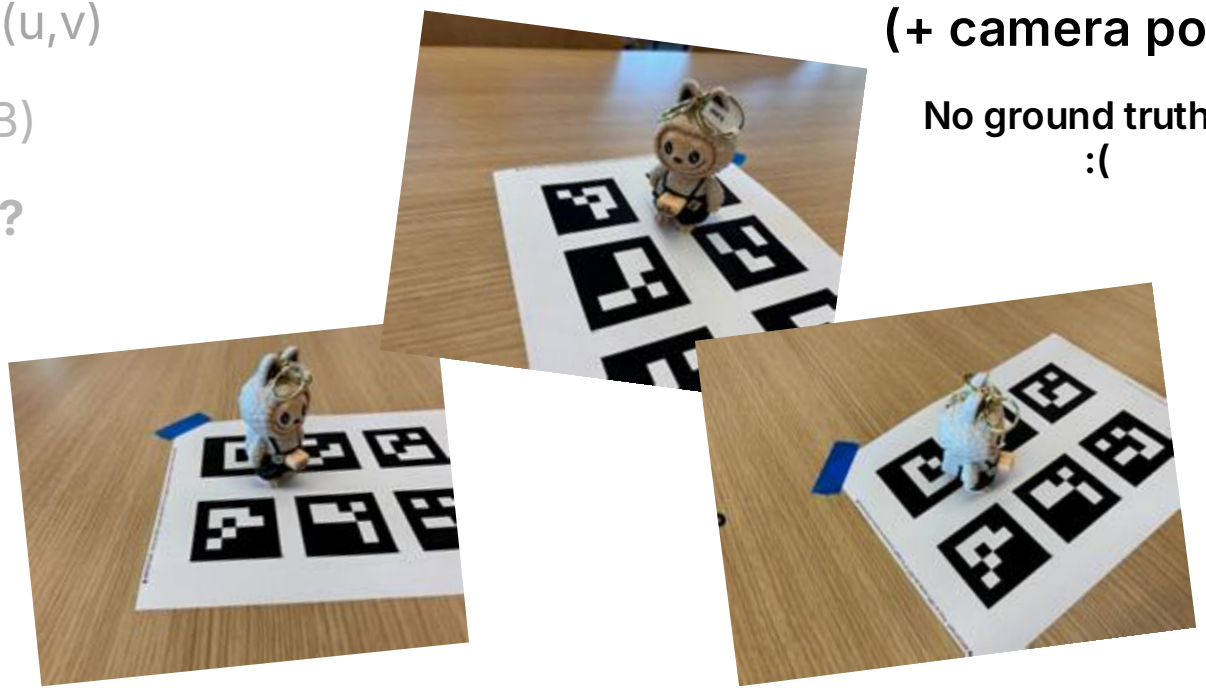
What do we have?

Training a NeRF

Input: (camera) + (u,v)

Output: color (RGB)

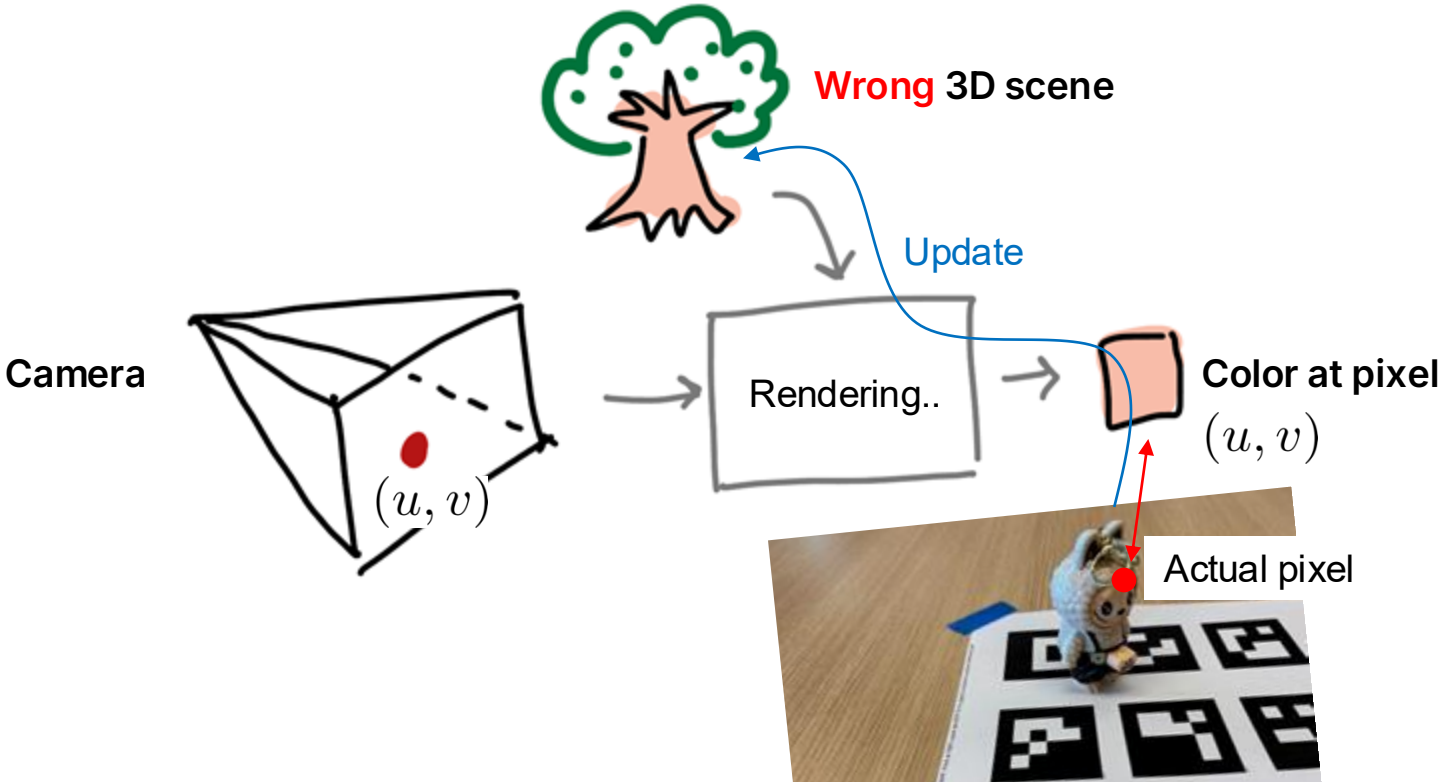
What do we have?



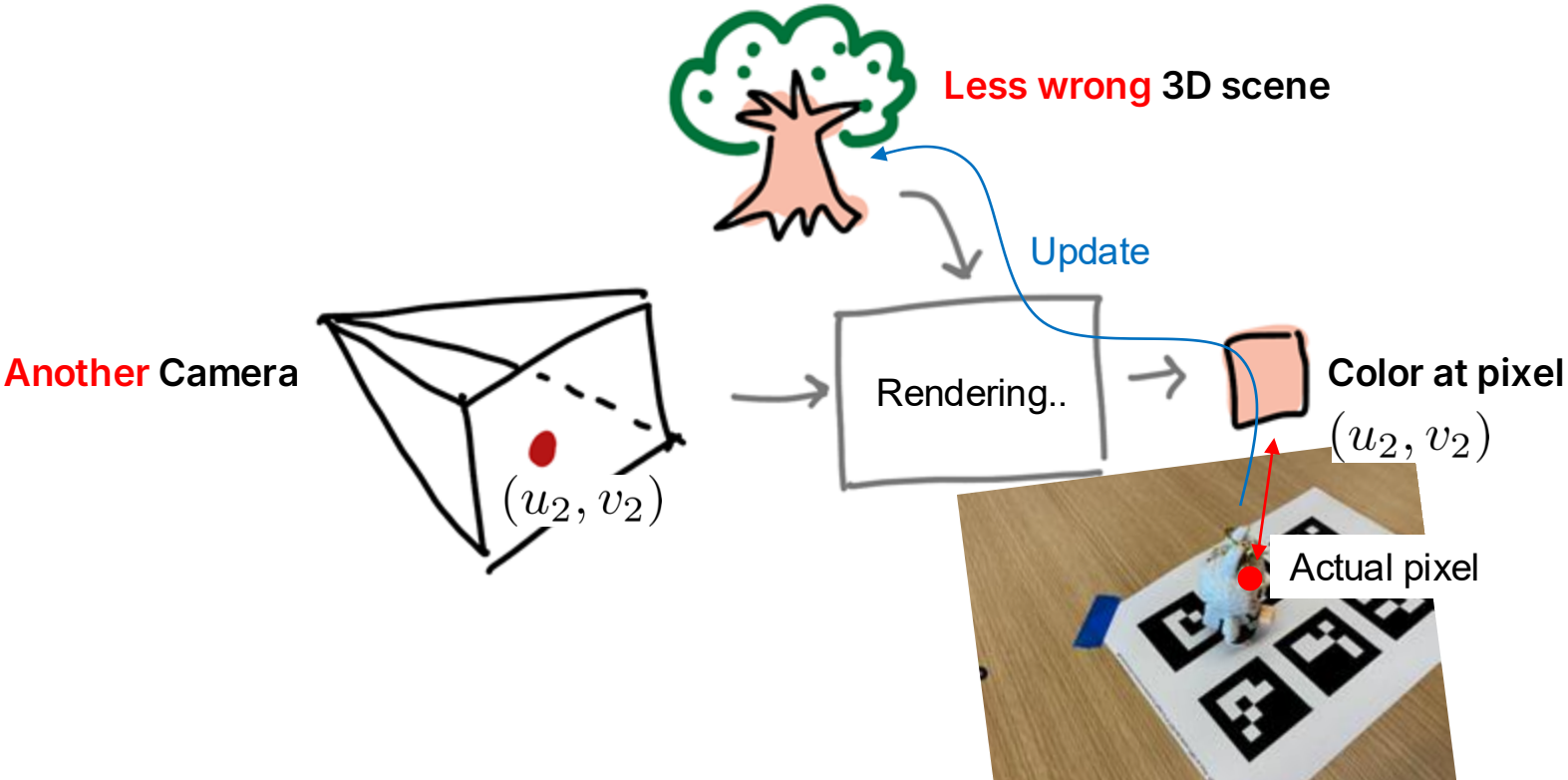
Images
(+ camera poses)

No ground truth 3D
:(

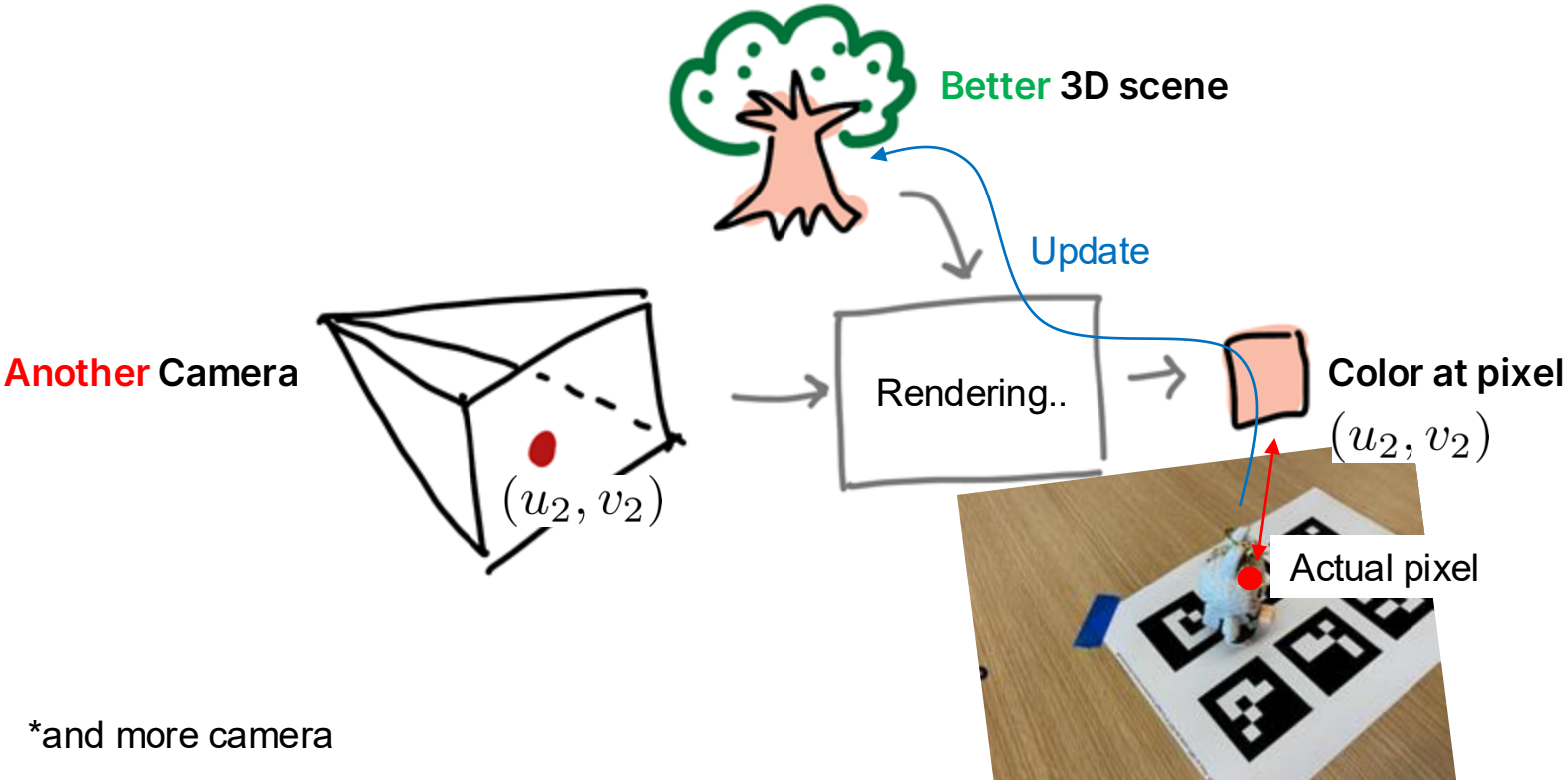
Training a NeRF (Philosophy-level)



Training a NeRF (Philosophy-level)



Training a NeRF (Philosophy-level)



*and more camera

How NeRF renders?

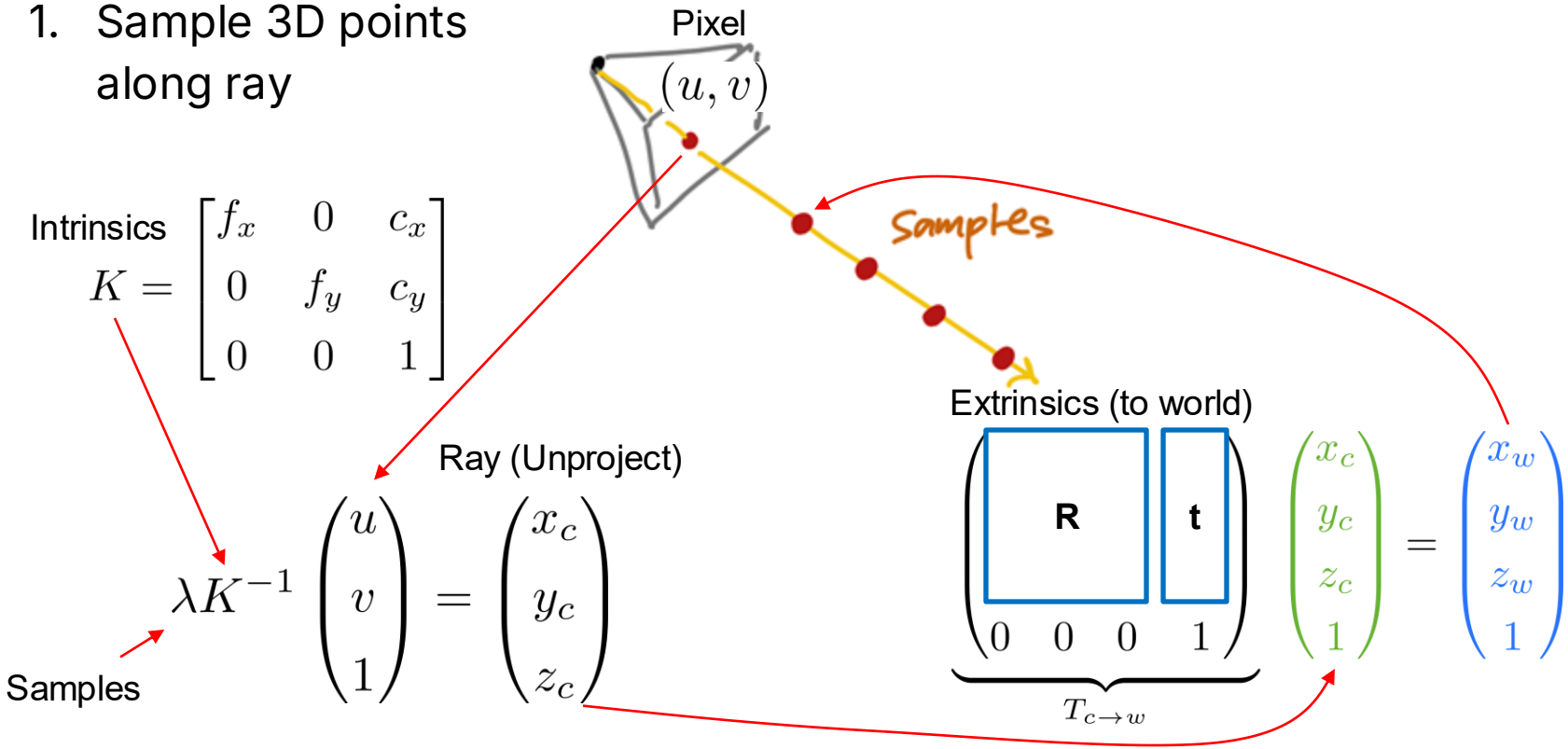
1. Sample points along the ray

2. Predict the color & density of each point (with Neural Networks)

3. Combine the colors with volumetric rendering

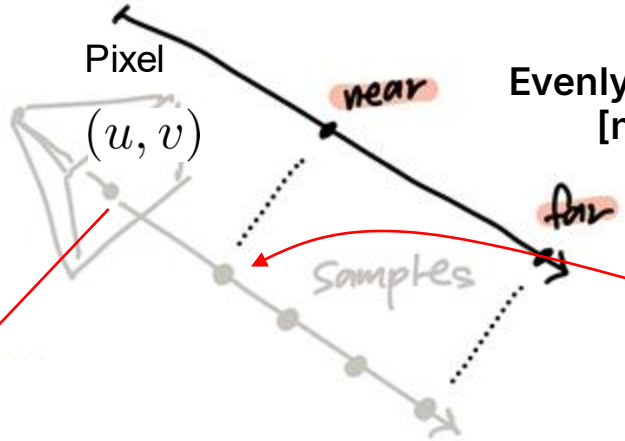
How NeRF renders? (3 steps)

1. Sample 3D points along ray



How NeRF renders?

1. Sample 3D points along ray



Evenly sample from [near, far]

Intrinsics

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Ray (Unproject)

$$\lambda K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}$$

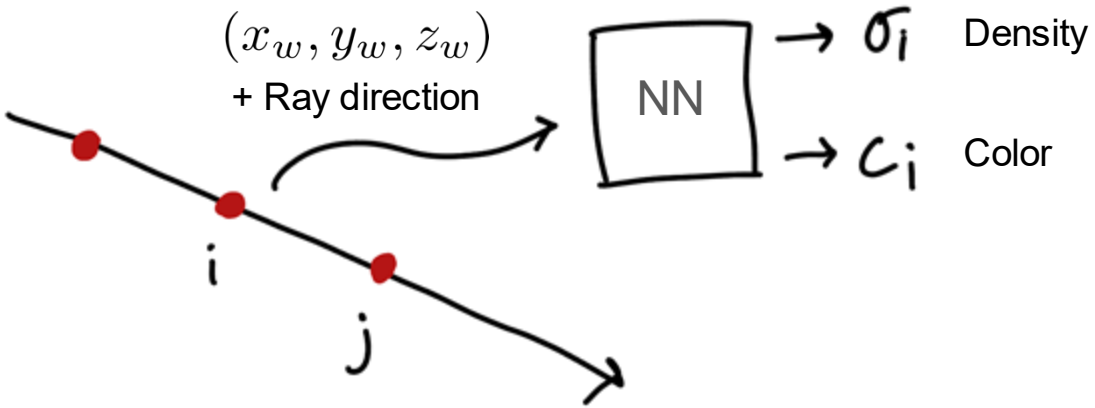
Samples

Extrinsics (to world)

$$\underbrace{\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0\ 0\ 0 & 1 \end{pmatrix}}_{T_{c \to w}} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$

How NeRF renders?

1. Sample 3D points along ray
2. Predict (color, density) at each 3D point
3. Render color at 2D pixel (using volumetric rendering)

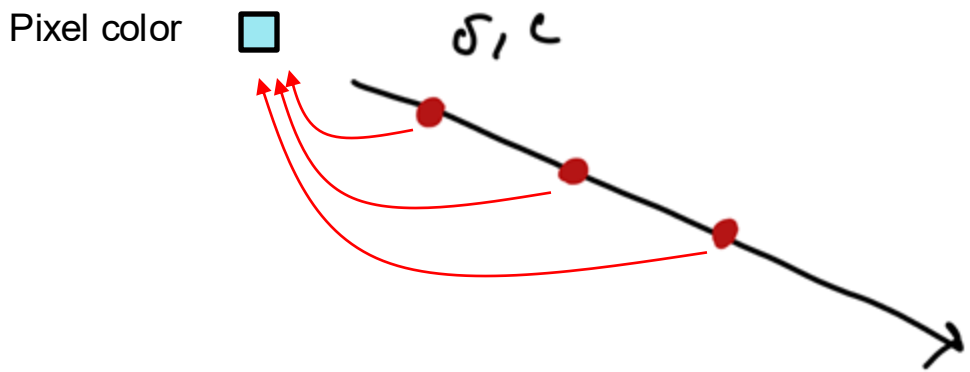


*Ray direction = View direction

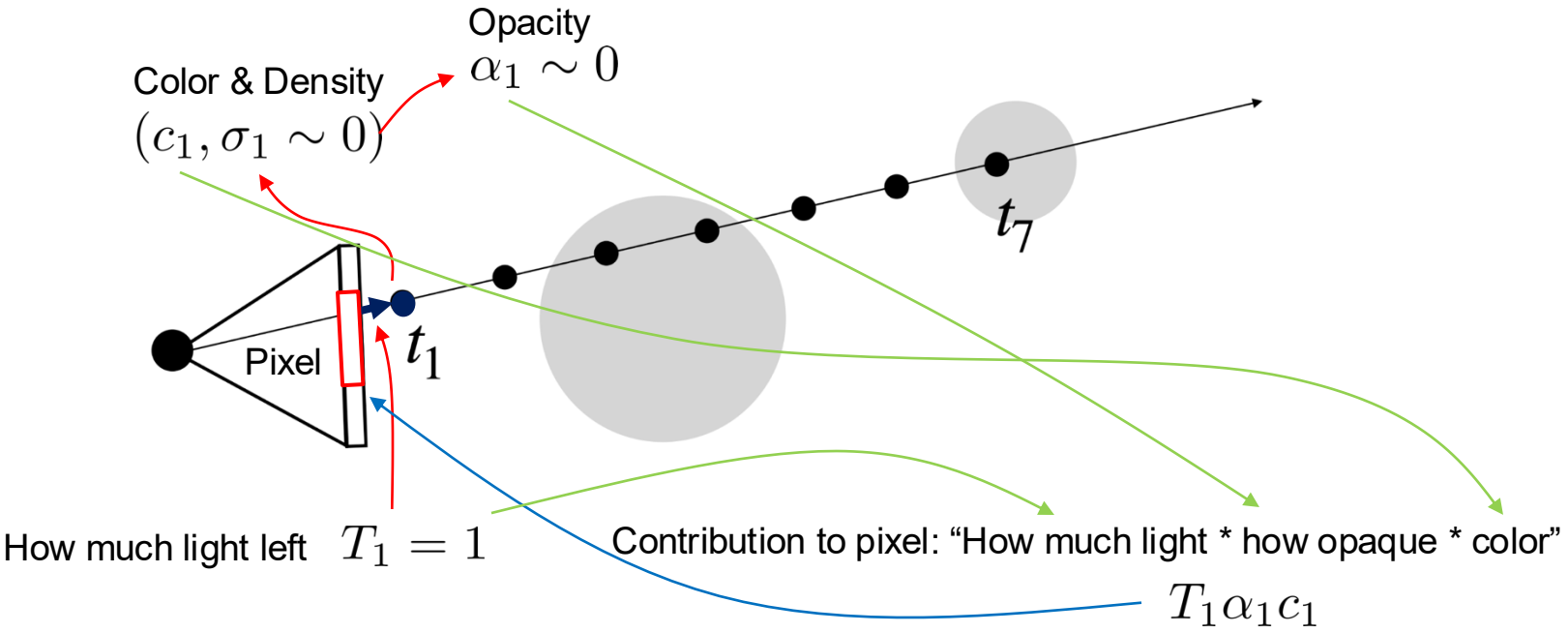
*NN is learnable!

How NeRF renders?

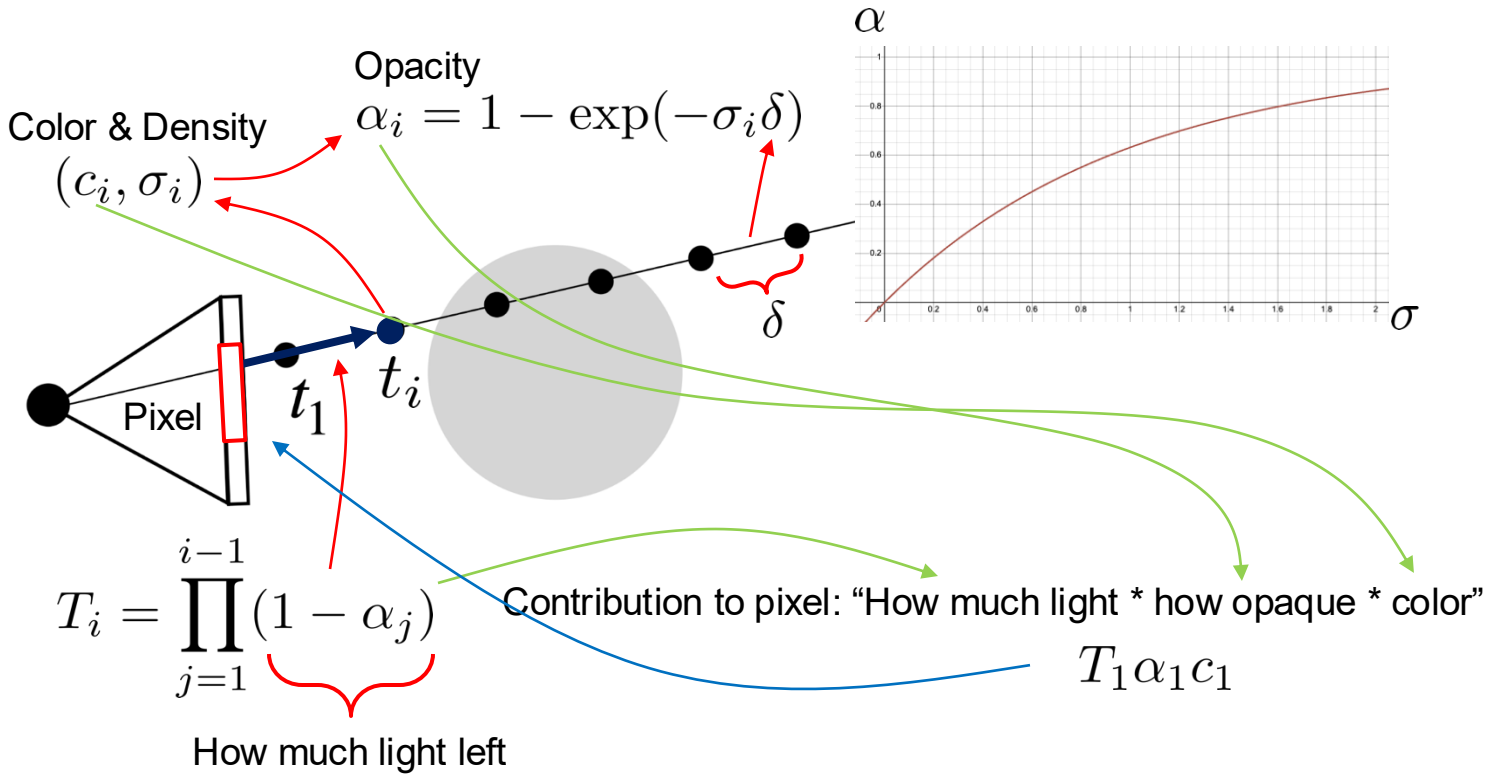
1. Sample 3D points along ray
2. Predict (color, density) at each 3D point
3. Render color at 2D pixel (combine using volumetric rendering)



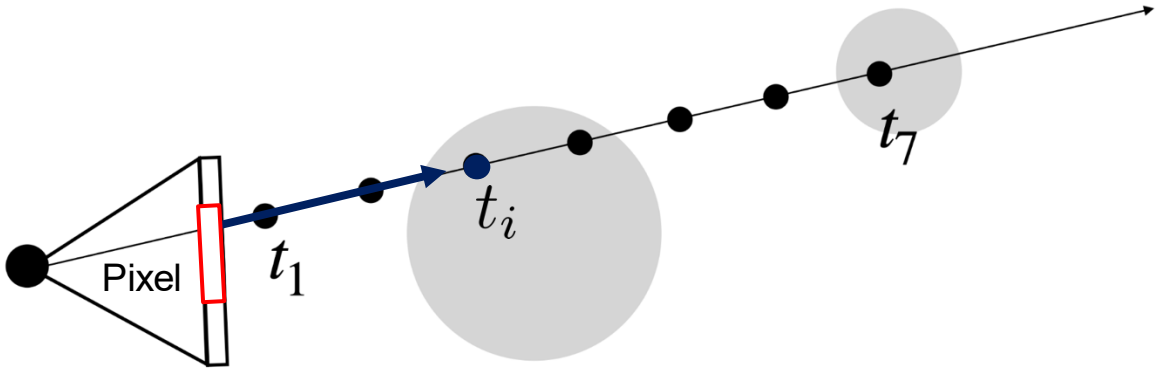
Accumulation of colors



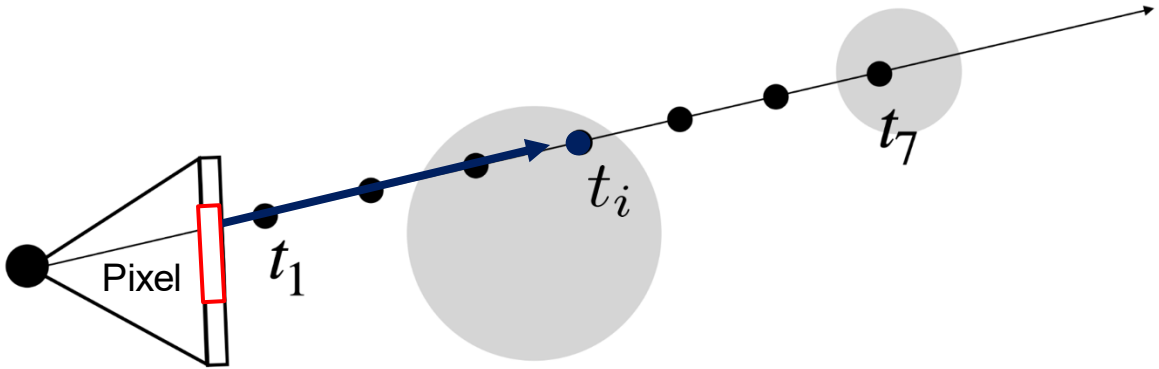
Accumulation of colors



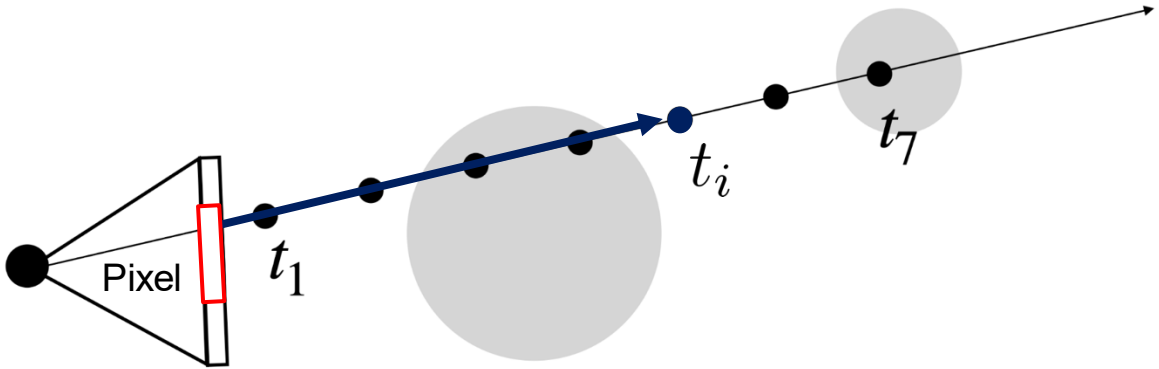
Accumulation of colors



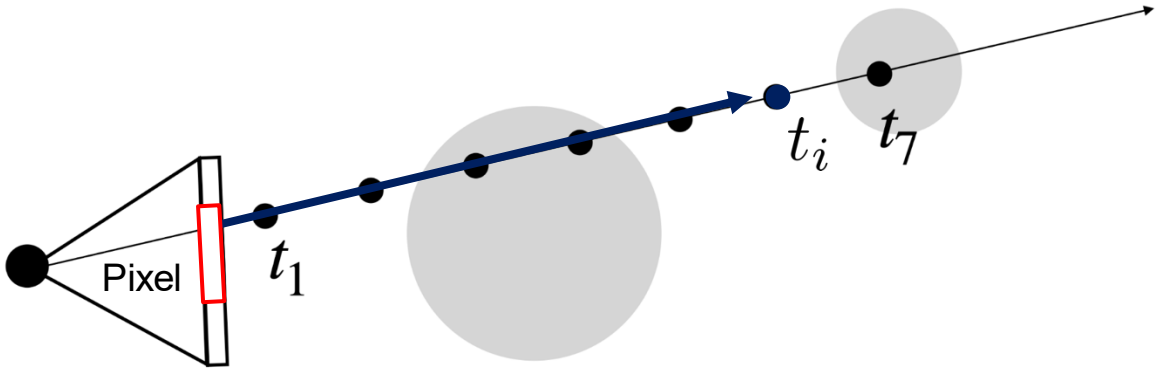
Accumulation of colors



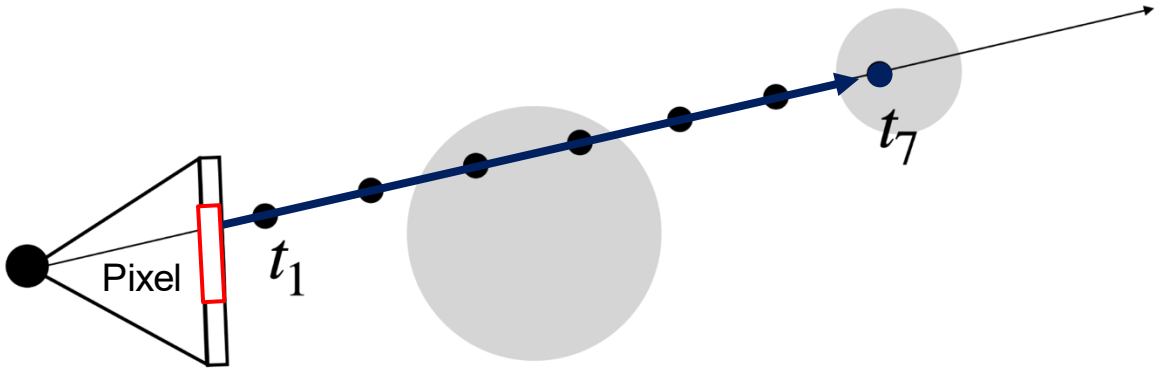
Accumulation of colors



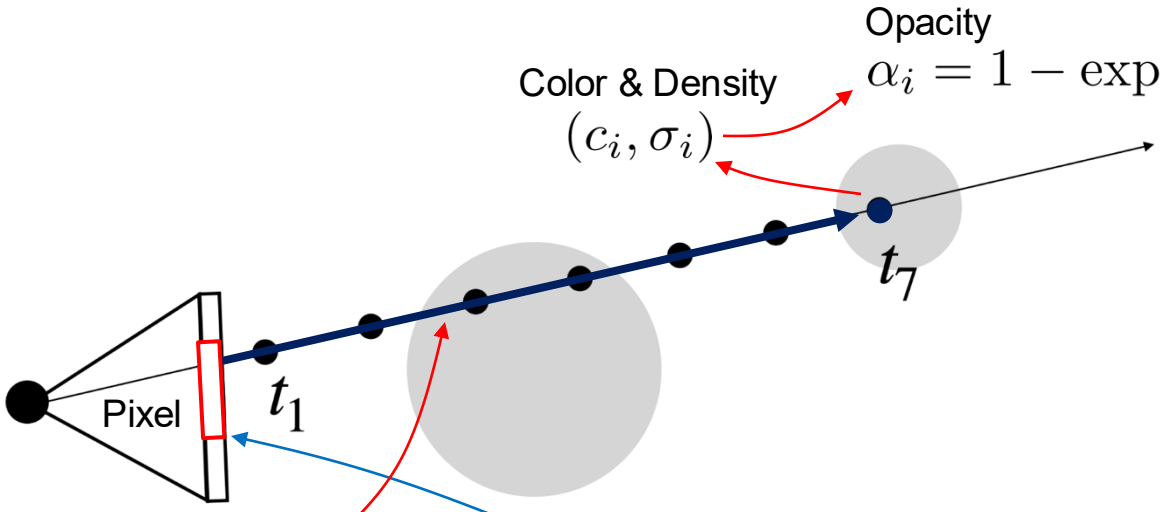
Accumulation of colors



Accumulation of colors



Accumulation of colors



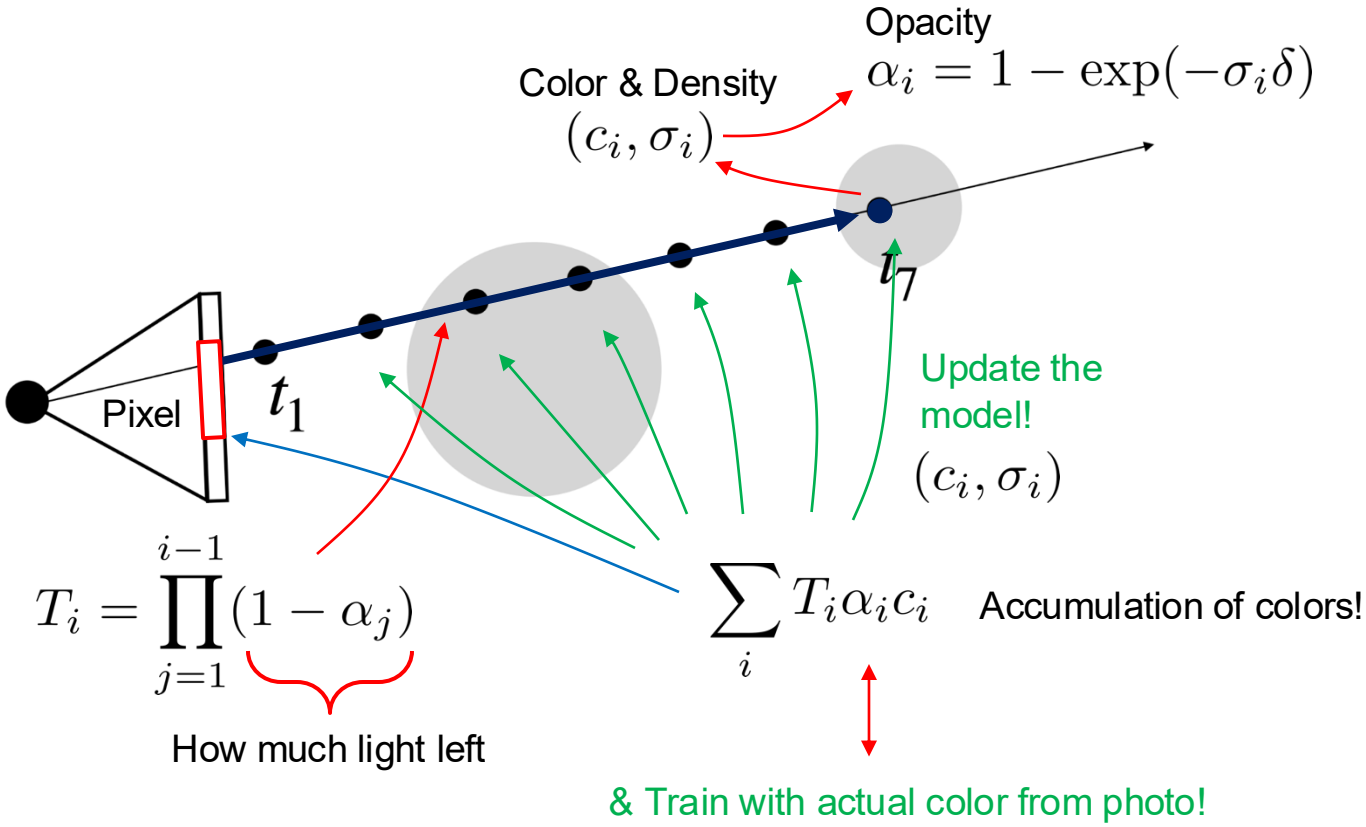
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

How much light left

$$\sum_i T_i \alpha_i c_i$$

Accumulation of colors!

Accumulation of colors



Worksheet :-)