# NeRF 실습

Presenter: 형준하

# NeRF

## Representing Scenes as Neural Radiance Fields for View Synthesis

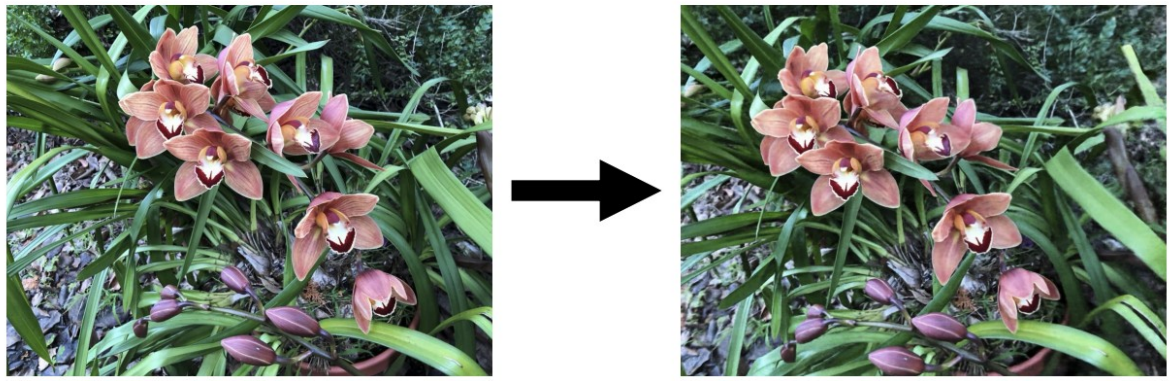
### ECCV 2020 Oral – Best Paper Honorable Mention

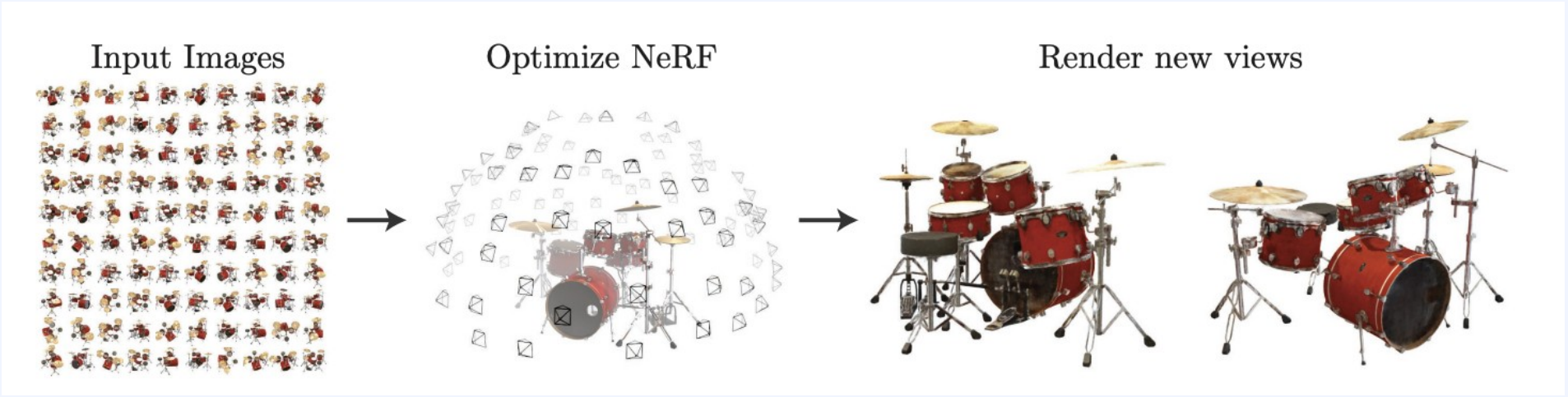
Ben Mildenhall* UC Berkeley
Pratul P. Srinivasan* UC Berkeley
Matthew Tancik* UC Berkeley
Jonathan T. Barron Google Research
Ravi Ramamoorthi UC San Diego
Ren Ng UC Berkeley

*Denotes Equal Contribution

# Introduction

Task : Novel View Synthesis(NVS) or View Interpolation



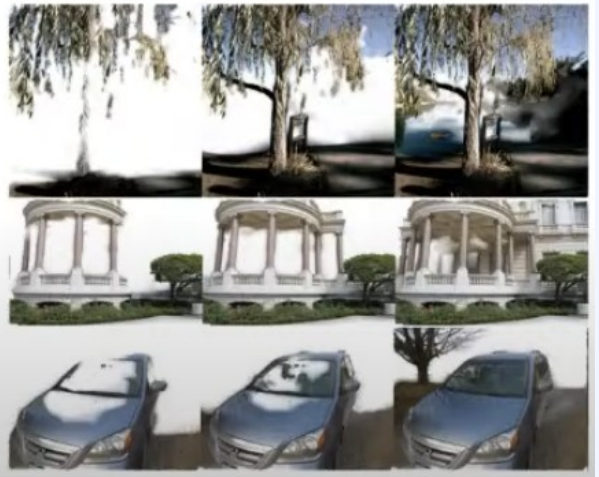




Mildenhall et al, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”, ECCV 2020

# Introduction

RGB-alpha volume rendering for view synthesis

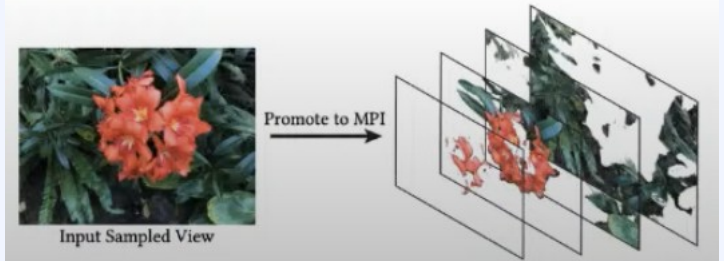
**Soft 3D**
(Penner & Zhang 2017)
Culmination of non-deep stereo matching techniques



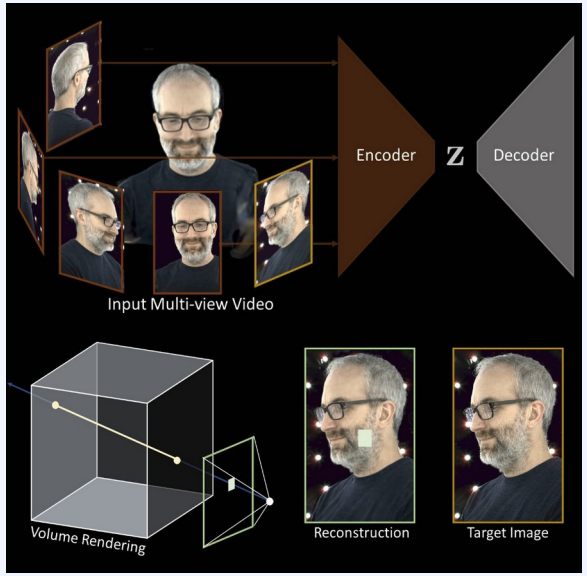
**Multiplane image methods**

Stereo Magnification (Zhou et al. 2018)
Pushing the Boundaries… (Srinivasan et al. 2019)
Local Light Field Fusion (Mildenhall et al. 2019)
DeepView (Flynn et al. 2019)
Single-View… (Tucker & Snavely 2020)

Typical deep learning pipelines - images go into a 3D CNN, big RGBA 3D volume comes out




**Neural Volumes**
(Lombardi et al. 2019)
Direct gradient descent to optimize an RGBA volume, regularized by a 3D CNN




1~10GB storage consumption ..!

Figures borrowed from John Barron's slides

# Introduction

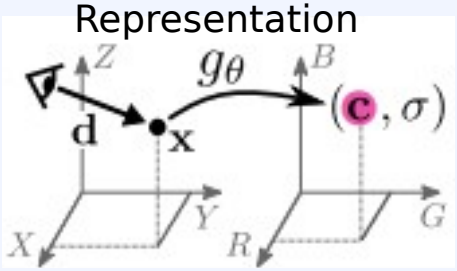
- Voxel representations
  - leads to discretization artifacts or degrades view-consistency
  - Large memory consumption!
- Implicit Representation has the advantages as follows:

  ***1) Lighter than voxel representation***

  ***2) Continuous***
- Neural Radiance Field encodes a continuous volume within the deep neural network, whose input is ***a single 5D coordinate*** and whose output is the ***volume density*** and ***view-dependent RGB color.***

Voxel Representation

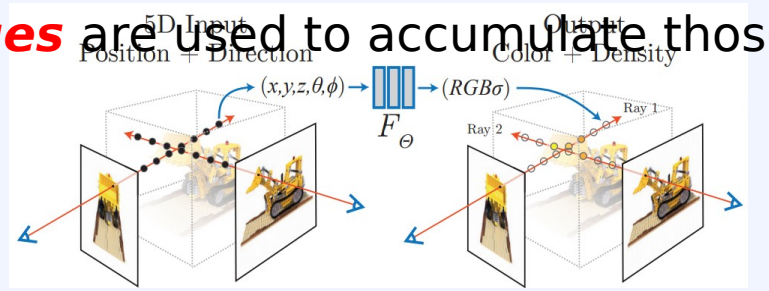
Implicit Representation

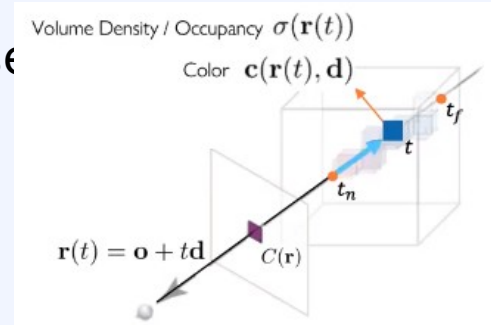



Neural Radiance Field

# NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

- The paper proposes a method that ***synthesizes novel view*** of complex scenes by optimizing an underlying continuous ***volumetric scene function*** using a sparse set of input views.
- This algorithm represents a scene using a fully-connected deep network, whose input is ***a single 5D coordinate*** and whose output is the ***volume density*** and ***view-dependent RGB color*** at that spatial location.
- Classical ***volume rendering techniques*** are used to accumulate those ties into a 2D image.

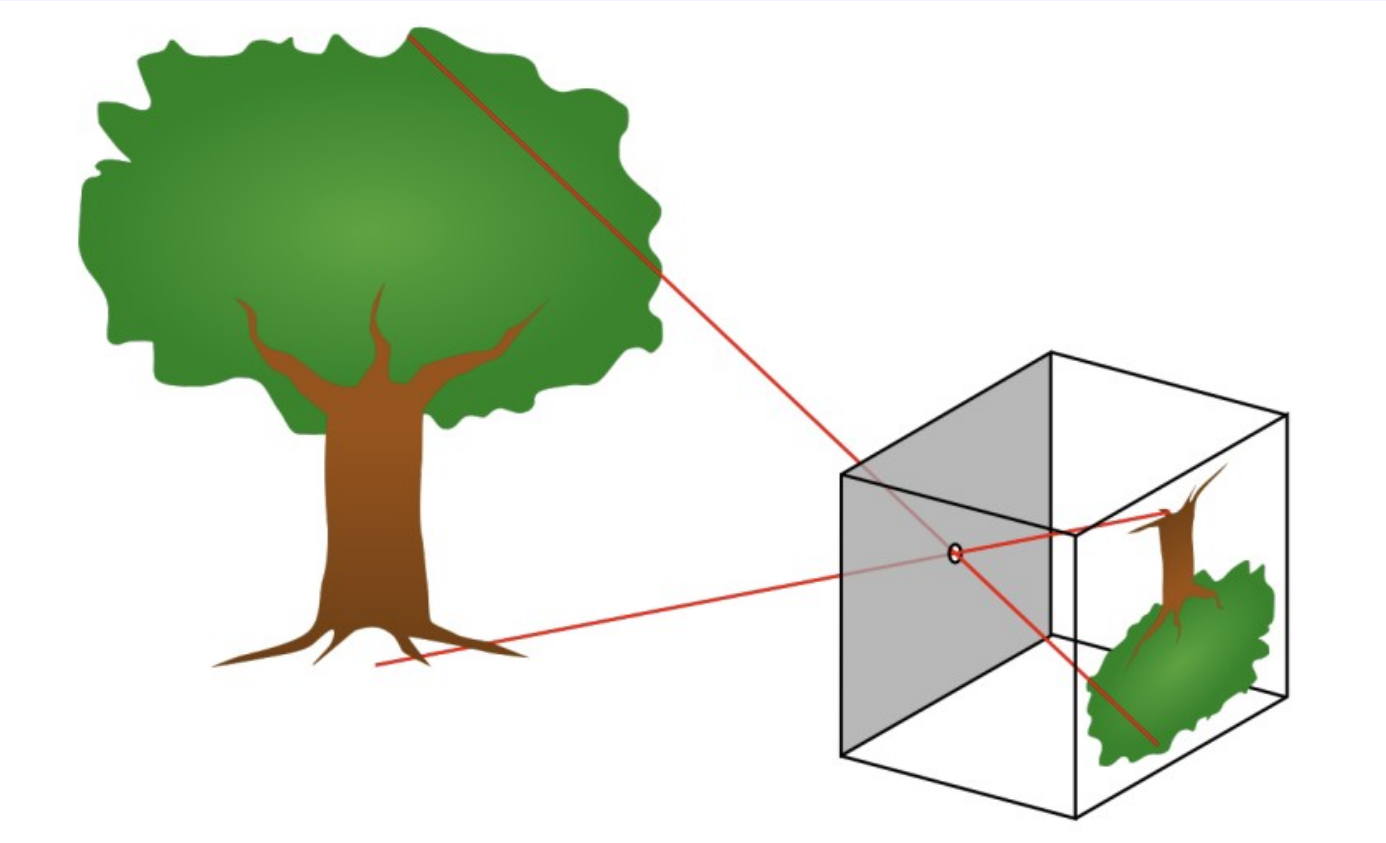


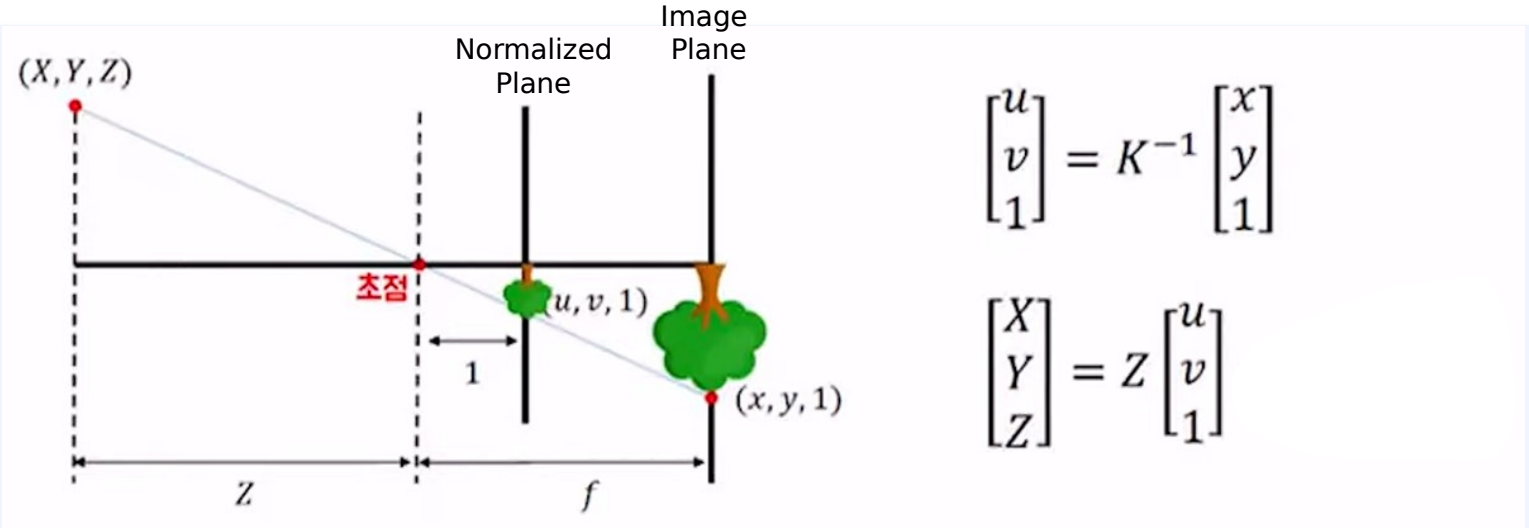

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

# Pinhole Camera Model

- Pinhole camera model

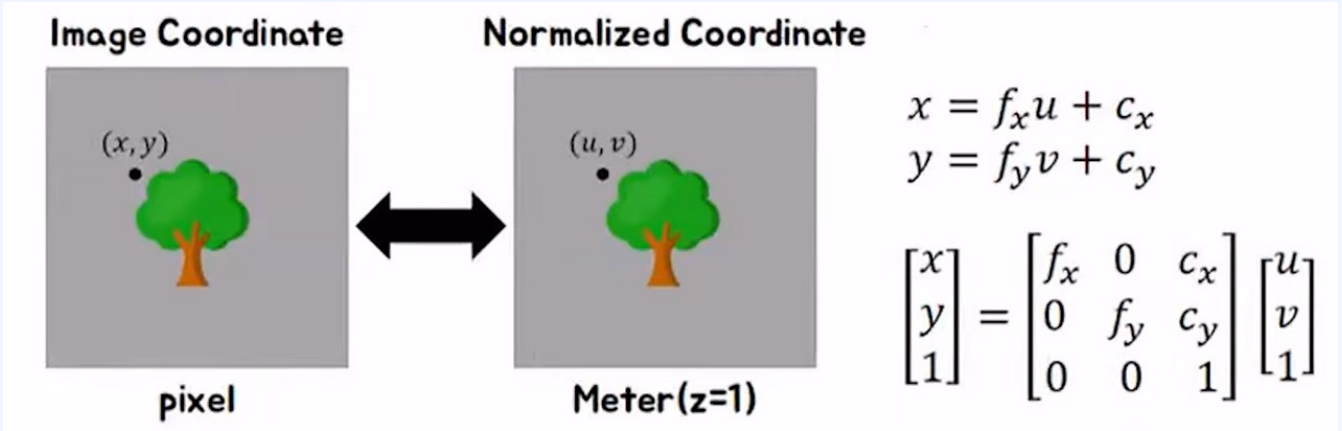


# Camera Intrinsic and Extrinsic Parameters




$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

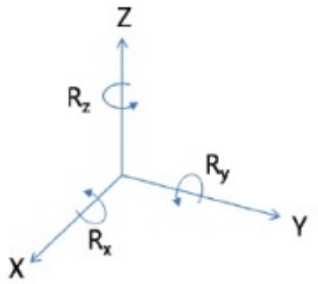



$$x = f_x u + c_x$$
$$y = f_y v + c_y$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

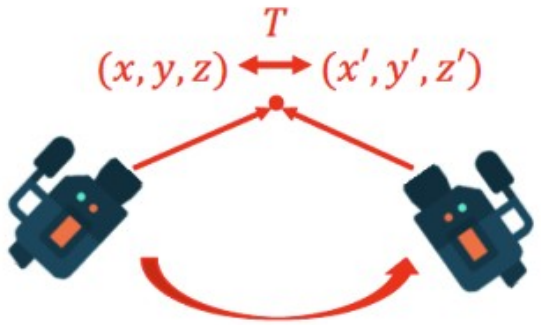
# Camera Intrinsic and Extrinsic Parameters

- Extrinsic Parameters




$$R_x(\theta)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad R_y(\theta)=\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_z(\theta)=\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R=R_z(\theta_3)R_y(\theta_2)R_x(\theta_1)$$

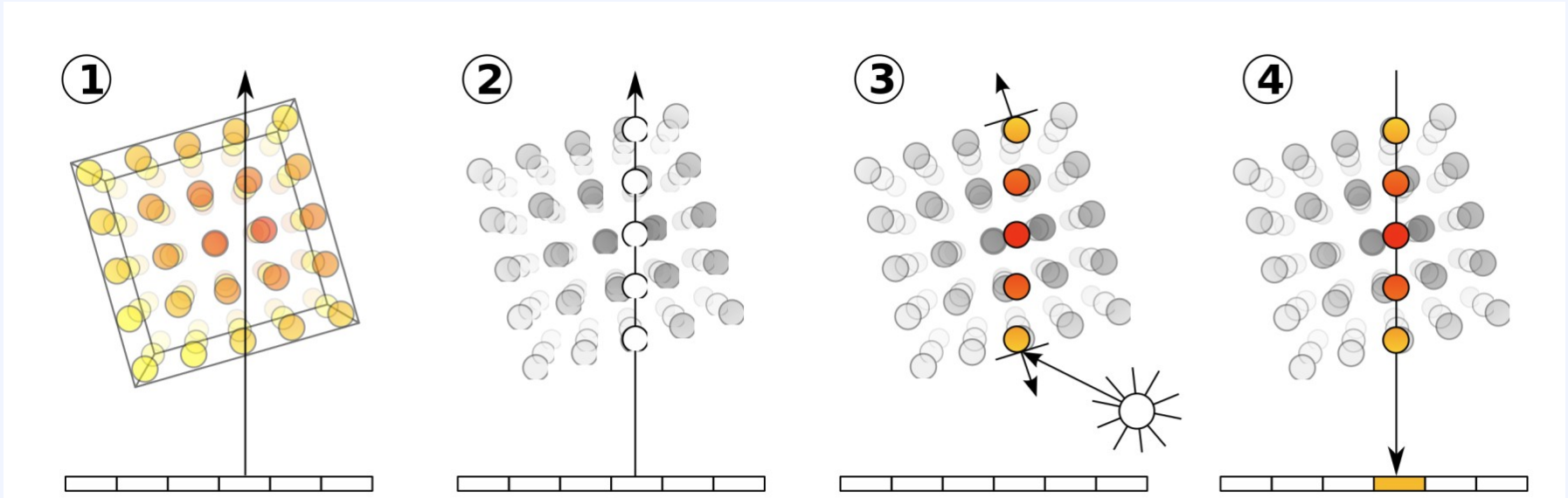



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{11} & r_{12} & r_{13} & t_x \\ r_{11} & r_{12} & r_{13} & t_x \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = [R|t]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# Method

- Neural Radiance Fields

$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$




Mildenhall et al, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”, ECCV 2020

# Method

- Volume rendering is trivially differentiable

Rendering model for ray r(t) = o + td:

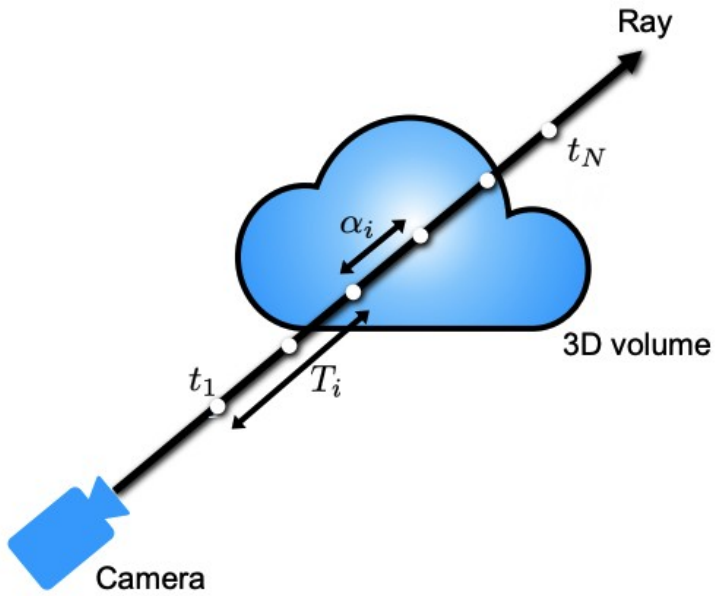
$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

weights

colors

How much light is blocked earlier along ray:

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

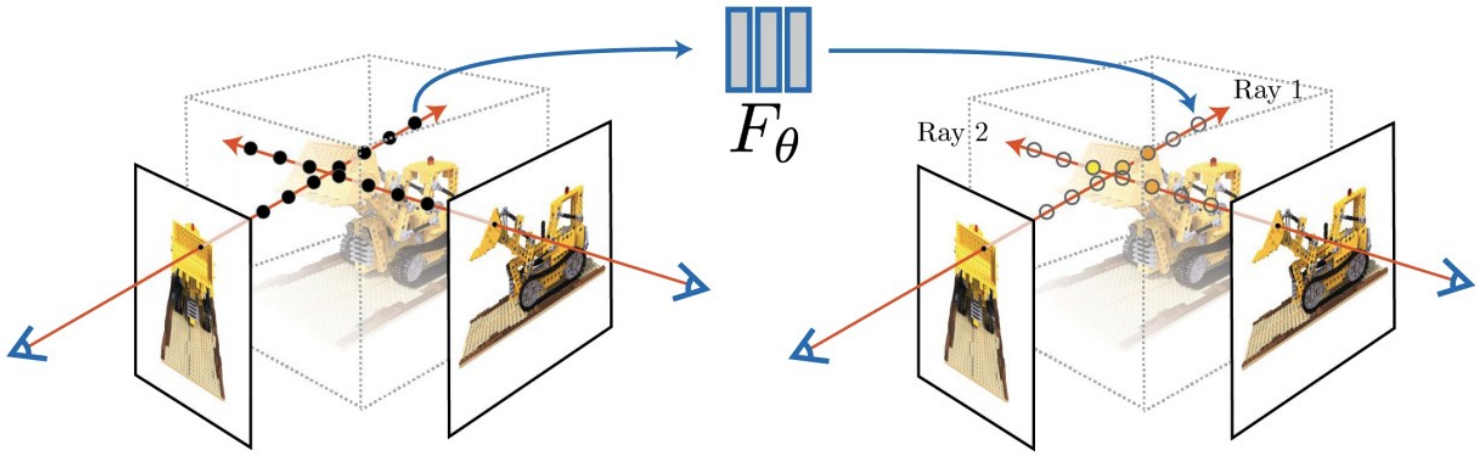



How much light is contributed by ray segment *i*:

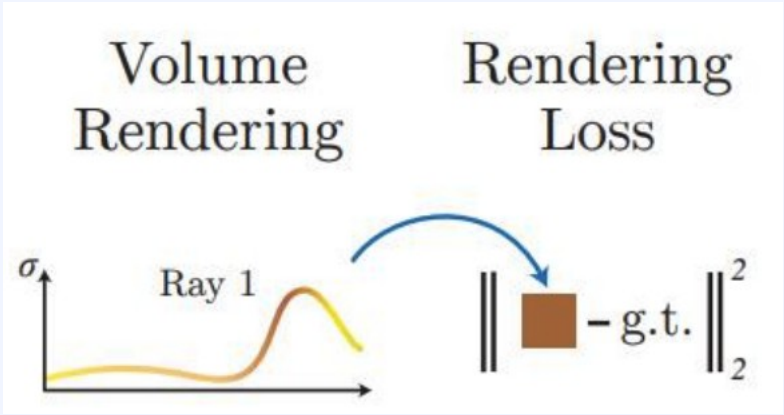
$$\alpha_i = 1 - e^{-\sigma_i \delta t_i}$$ ← Density * Distance Between Points

# Method

- Optimize with gradient descent on rendering loss

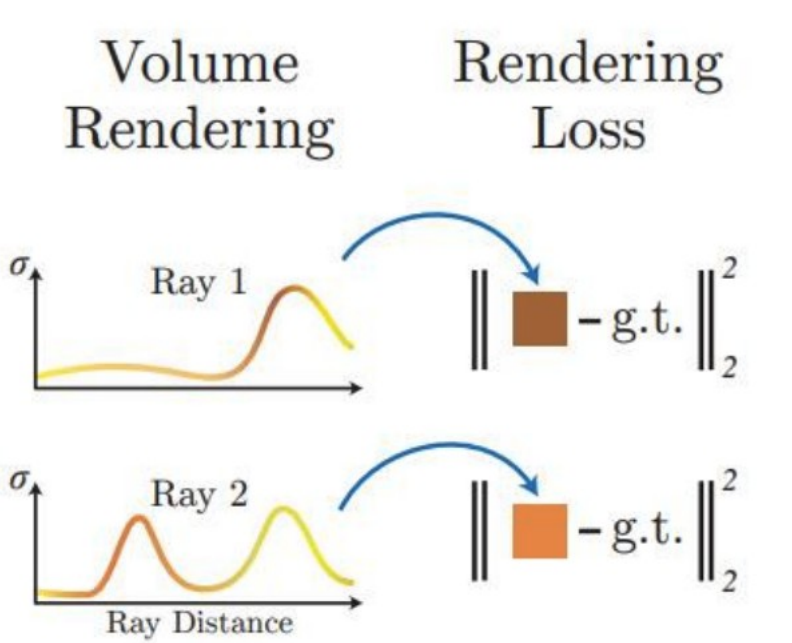



$$\min_\theta \sum_i || \text{render}_i(F_\theta) - I_i ||^2$$



# Method

- Training network to reproduce all input views of the scene



# Results – View direction dependencies

# Results – View direction dependencies

# Results – Depth map

# Results – Depth map

# Future Works

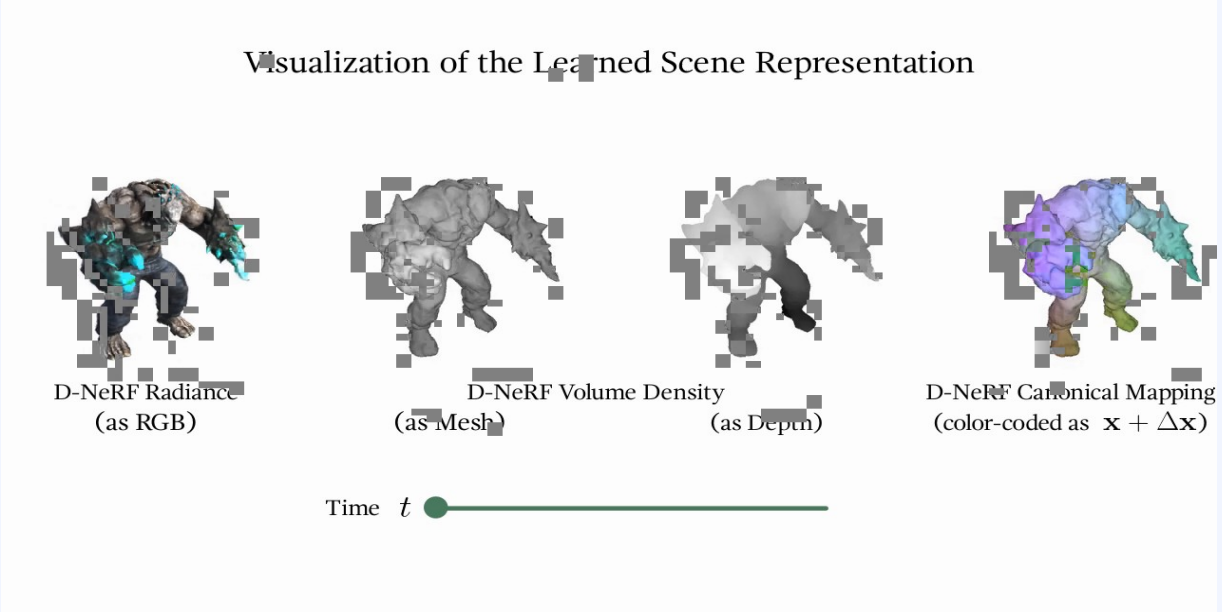
- In-the-wild setting
  - Appearance variations, occlusions, inaccurate camera poses

# Future Works

- NeRF + GAN
  - No need of Multiview images
  - Generate diverse images & scenes

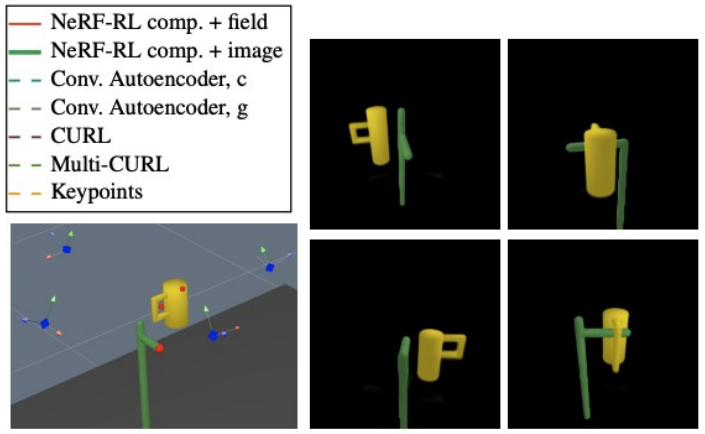
# Future Works

- Dynamic Scenes



# Future Works

- 3D understanding
  - Any task based on 2D images/videos that require 3D understanding(RL, )




(b) Left: Blue coordinate frames denote the four camera poses. Red points are the expert keypoints. Right: NeRF renderings (different scenes).