[A4-004] 딥러닝 코딩 실습

Lecture 04: Neural Radiance Fields (NeRF)

Hak Gu Kim

hakgukim@cau.ac.kr

Immersive Reality & Intelligent Systems Lab (IRIS LAB)

Graduate School of Advanced Imaging Science, Multimedia & Film (GSAIM)

Chung-Ang University (CAU)

26 Jan. 2023

IRIS LAB
Immersive Reality &
Intelligent Systems Lab.

Topic

- Neural Rendering

— Neural Radiance Fields

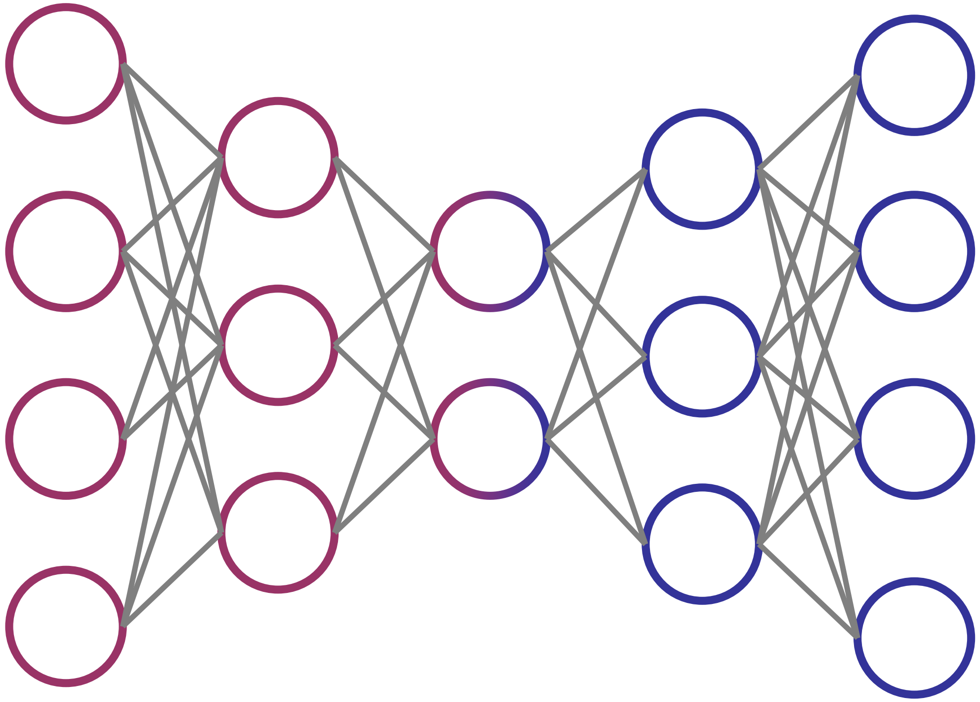
Background: **Recognize 3D from a 2D Image**

- Human can recognize 3D from a single image

Background: **Recognize 3D from a 2D Image**

- Can AI learn to infer 3D from a 2D image?

Input Images



Neural Networks

3D Reconstruction

Limitations of Existing Works

- Recently, learning-based 3D reconstruction methods have achieved impressive results

— Most learning-based methods are restricted to synthetic data, mainly because they require accurate 3D ground truth models as supervision

- To overcome this barrier, a novel approaches have been investigated that require only 2D supervision in the form of depth maps or multi-view images have been proposed

— They suffer from discretization artifacts and the computational cost limits them to small resolutions or deforming a fixed template mesh

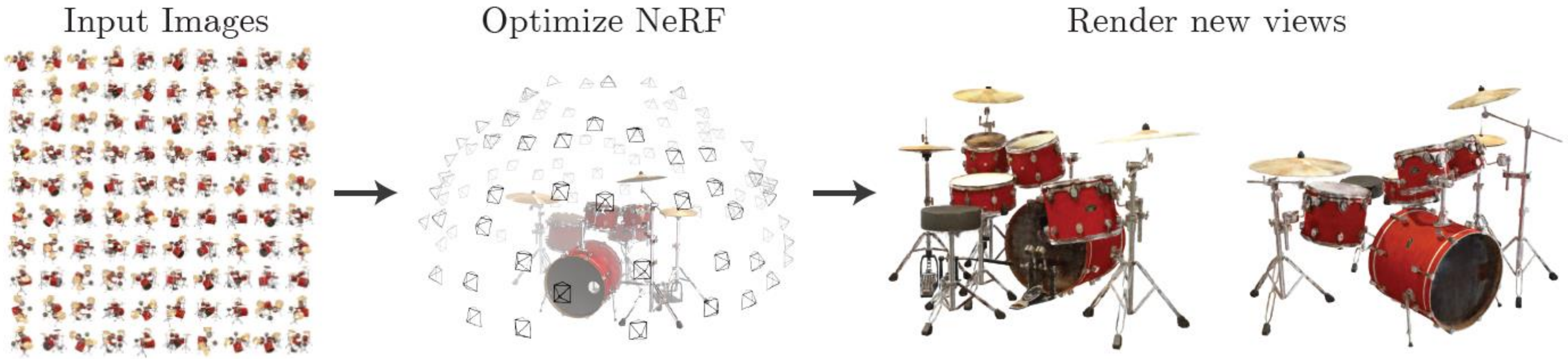
Limitations of Existing Works

- Most recently, implicit representations for shape and texture have been proposed which do not require discretization during training and have a constant memory footprint

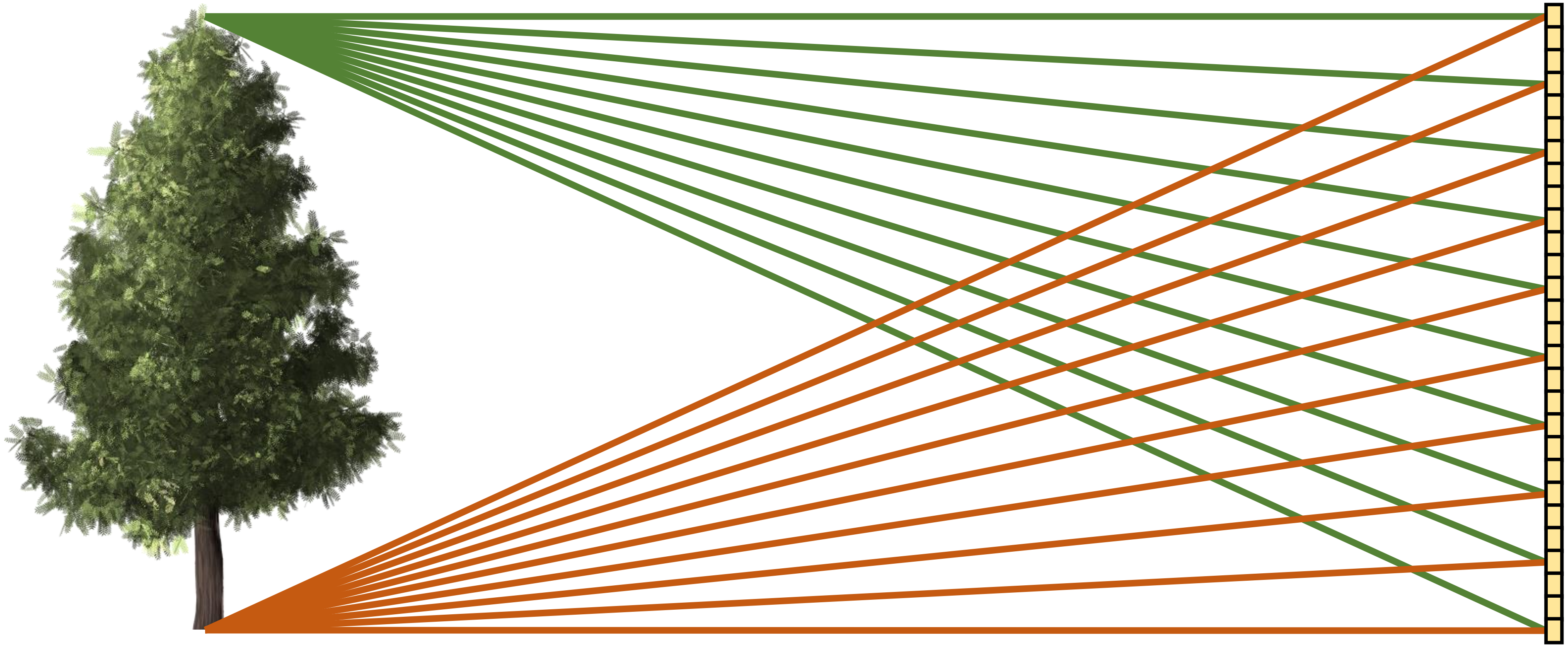
— However, the implicit representations-based approaches require 3D ground truth for training and it remains unclear how to learn implicit neural representations from image data alone

Background: **Scenario of Novel View Synthesis**

- Synthesis a 2D image at a novel viewpoint from *N* 2D images at various viewpoints



Background: **Pinhole Camera Model**

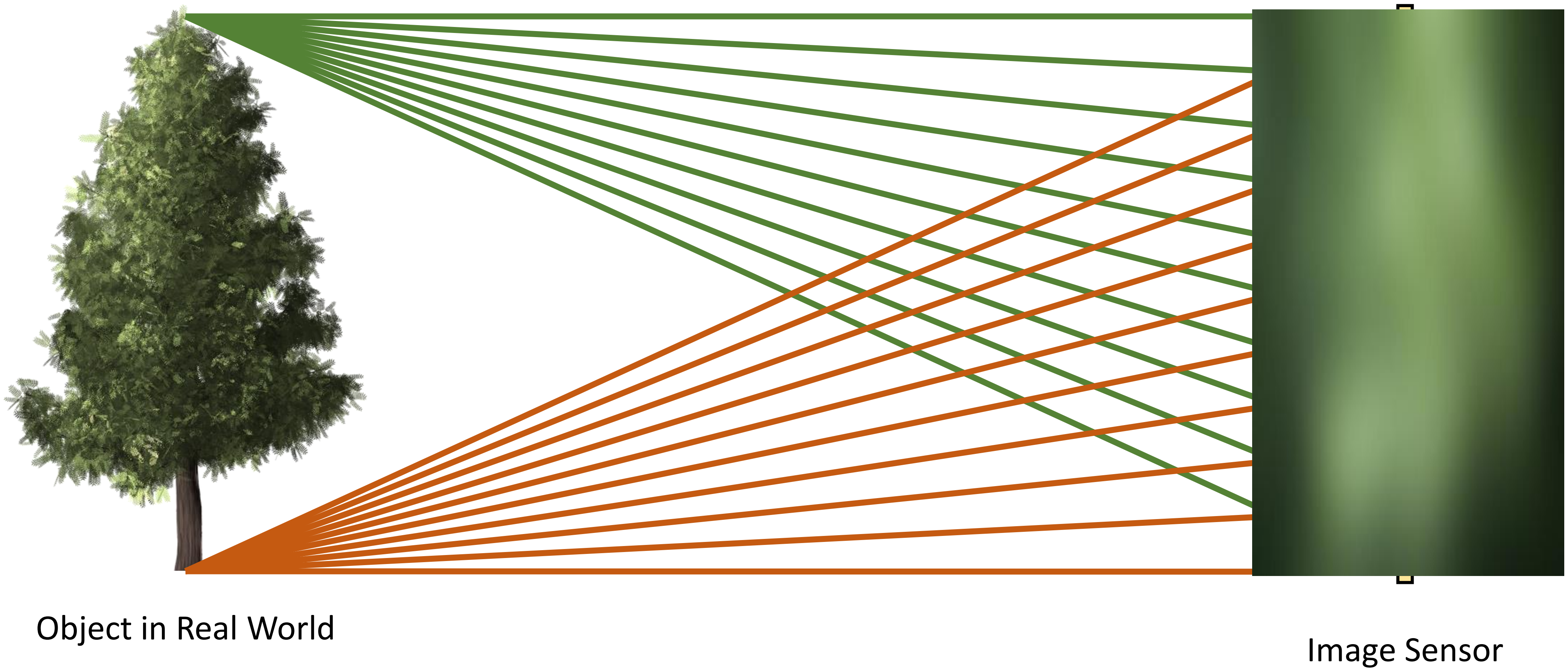


Object in Real World

Image Sensor

***Note**: Many of these lecture note slides were adapted from F. Durand (MIT), G. Wetzstein (Stanford), K. Kitani (CMU), I. Gkioulekas (CMU), and S. Süsstrunk (EPFL).

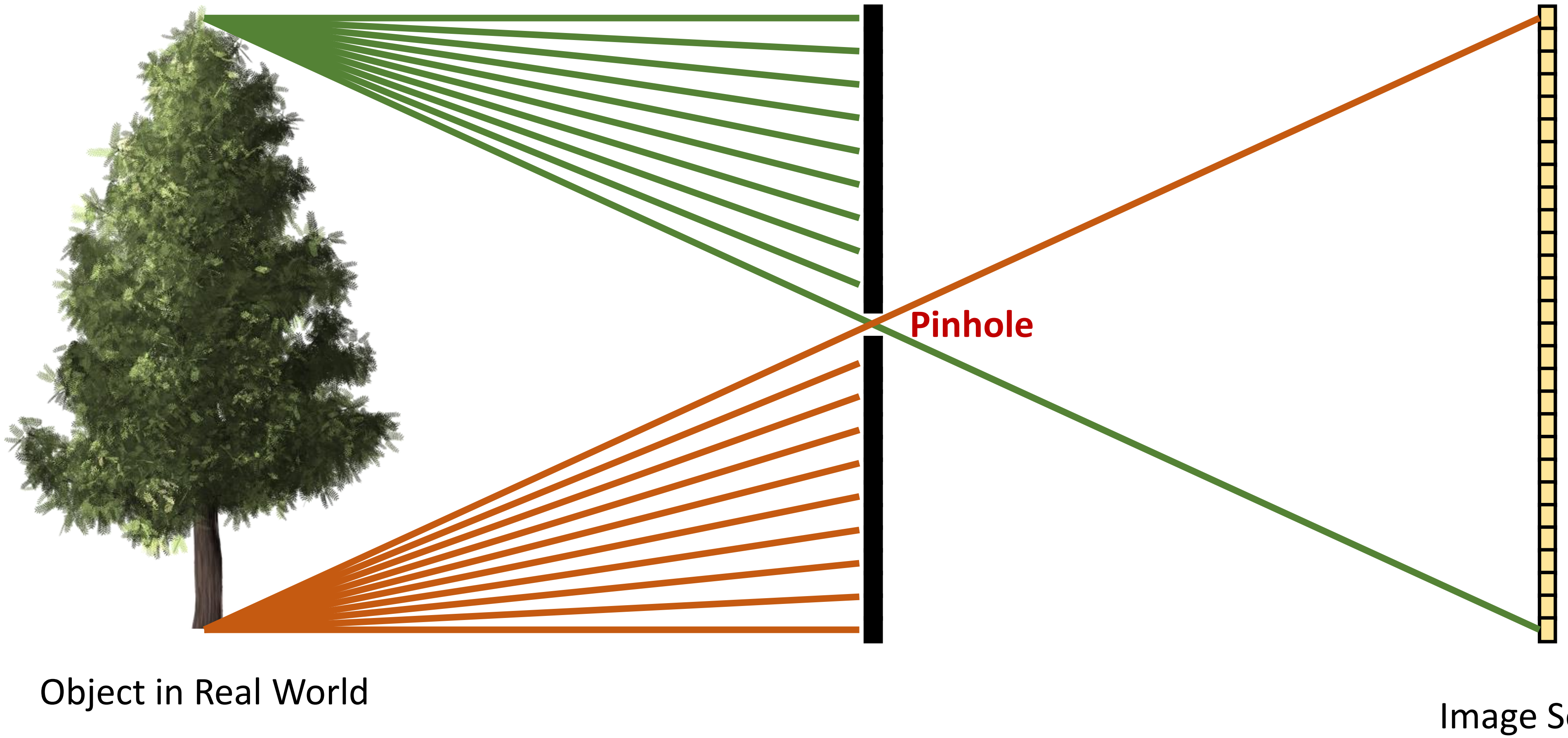
Background: **Pinhole Camera Model**



*Note: Many of these lecture note slides were adapted from F. Durand (MIT), G. Wetzstein (Stanford), K. Kitani (CMU), I. Gkioulekas (CMU), and S. Süsstrunk (EPFL).

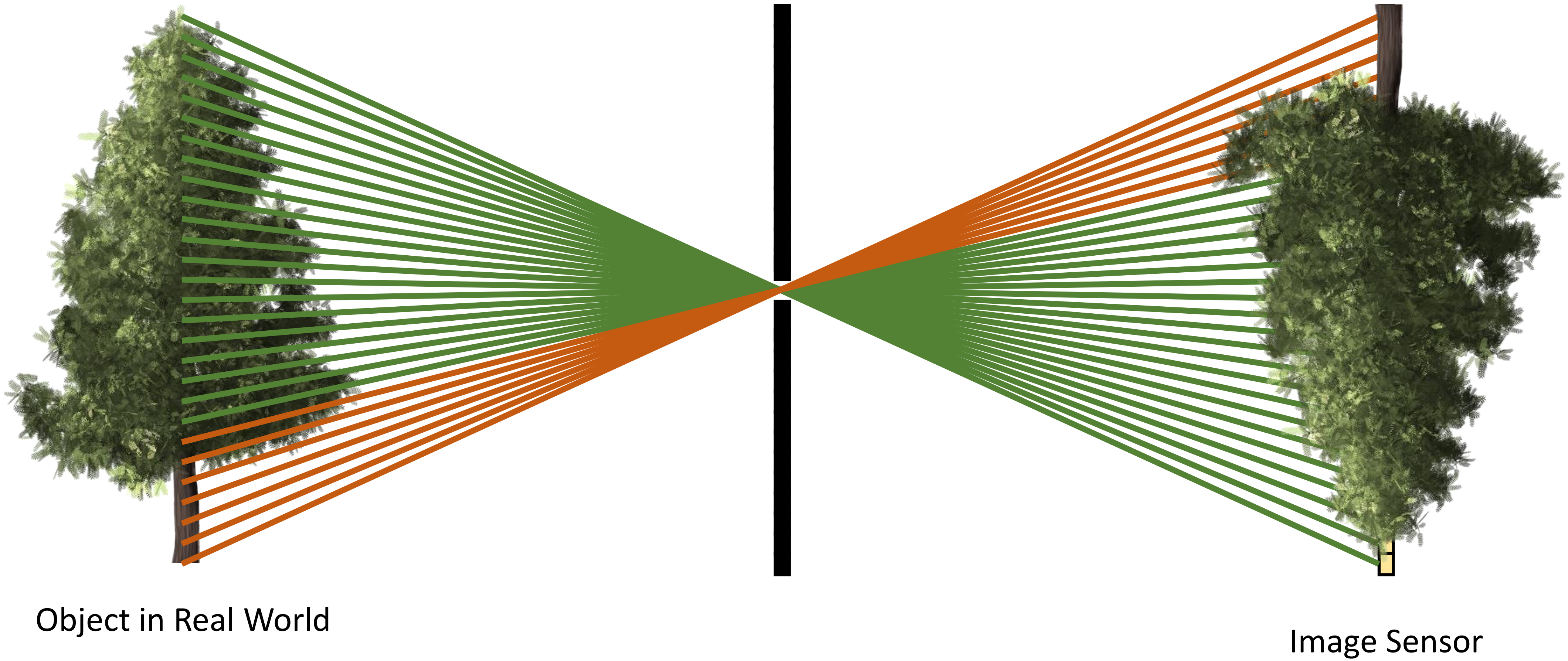
Background: **Pinhole Camera Model**

- Principal of Pinhole Camera

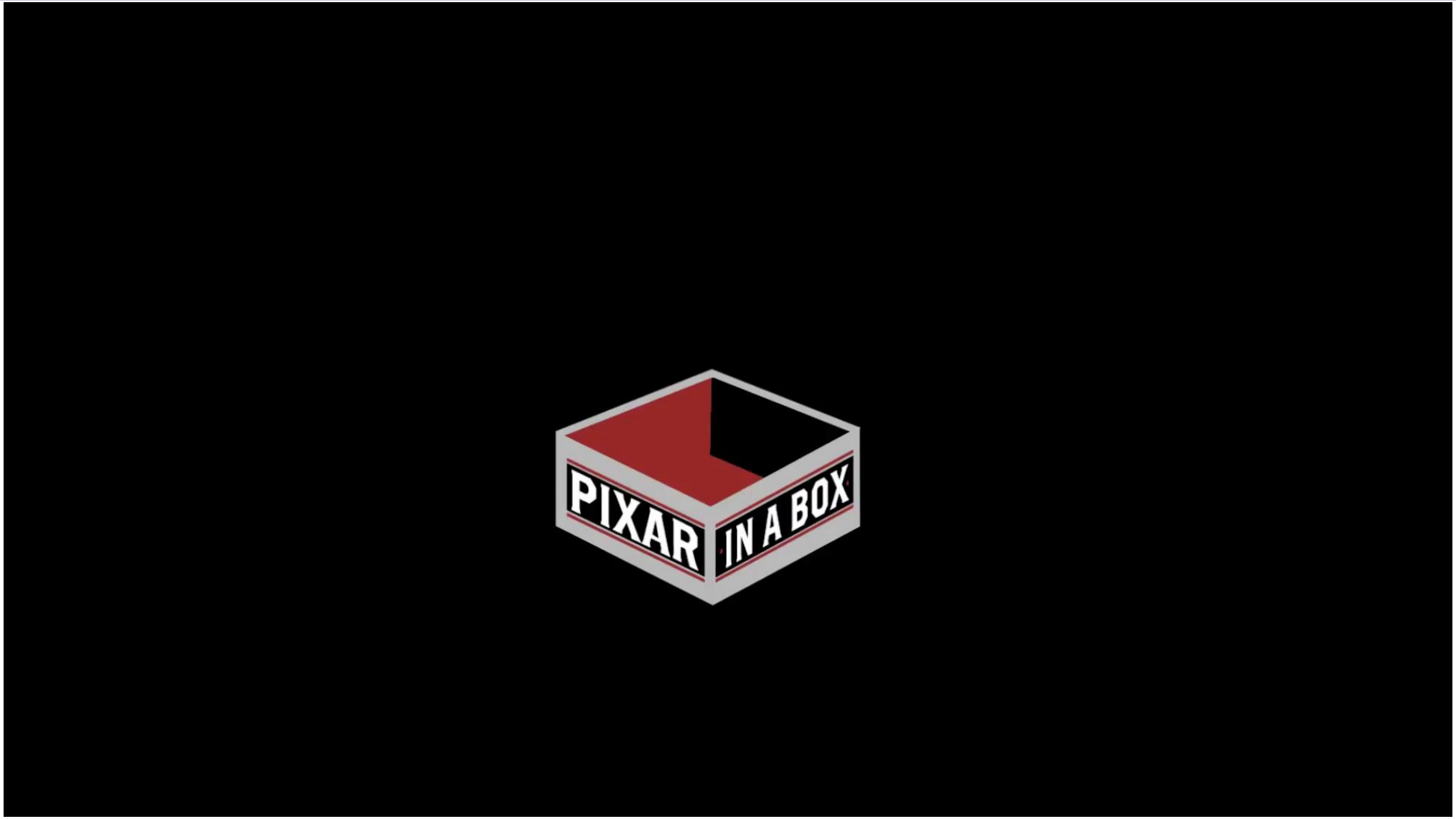


Background: **Pinhole Camera Model**

- Principal of Pinhole Camera

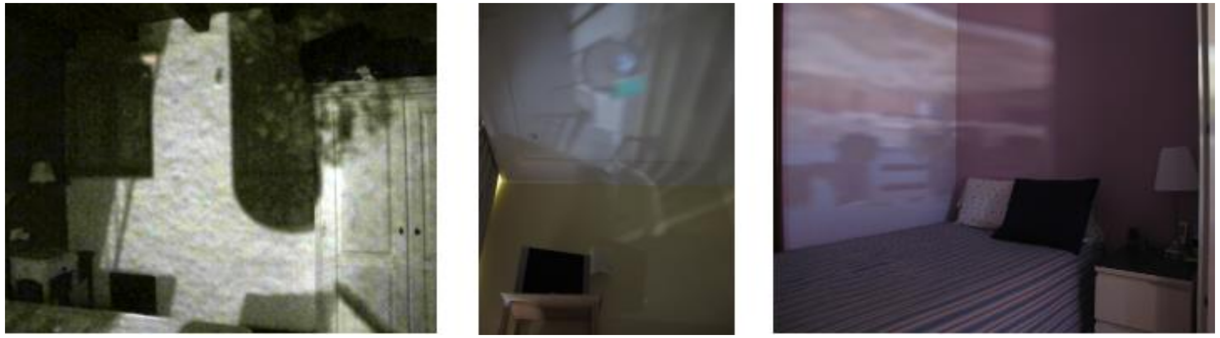


Background: **What Is A Pinhole Camera?**

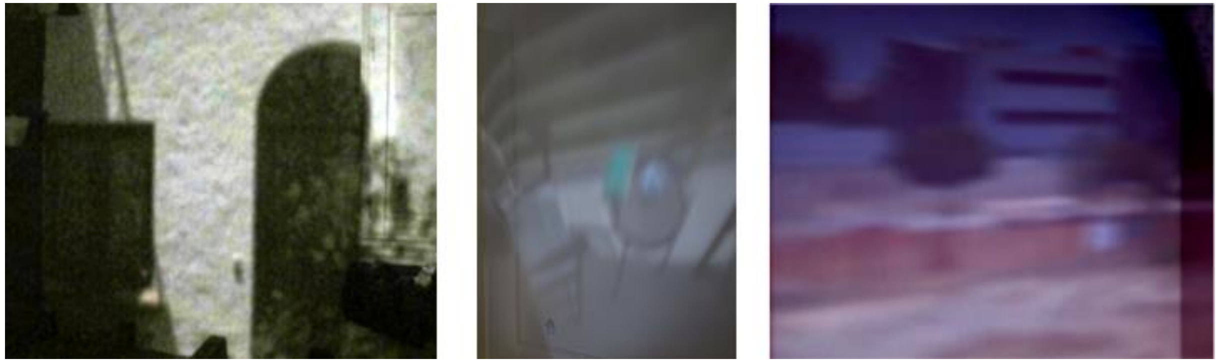


Background: **Accidental Pinhole Camera Image**

(a)



(b)



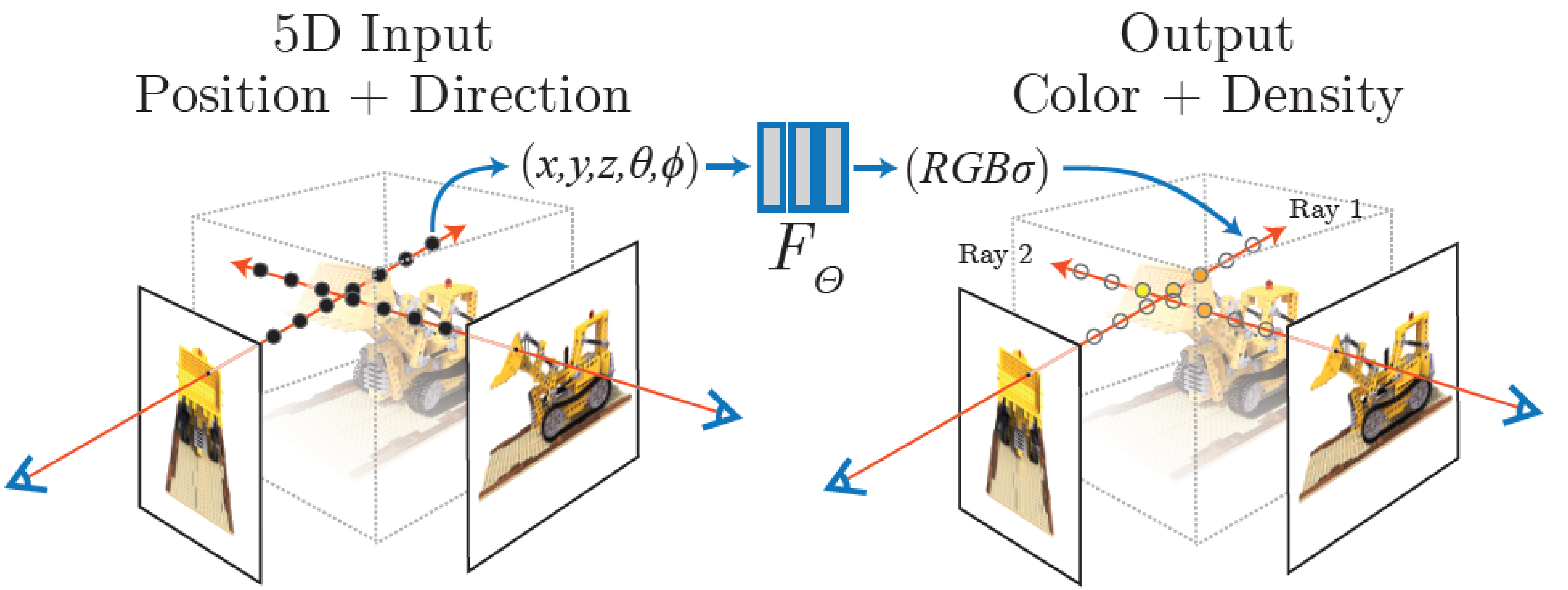
(c)

(d)

(a) Three different rooms illuminated by exterior light, creating shading patterns within the room
(b) The effect of closing the windows, leaving only a small aperture, turning the room in a camera obscura
(c) Upside-down images of (b)
(d) The true view from the window to the outside

A. Torralba and W. T. Freeman, Accidental pinhole and pinspeck cameras: revealing the scene outside the picture, **CVPR,** 2012

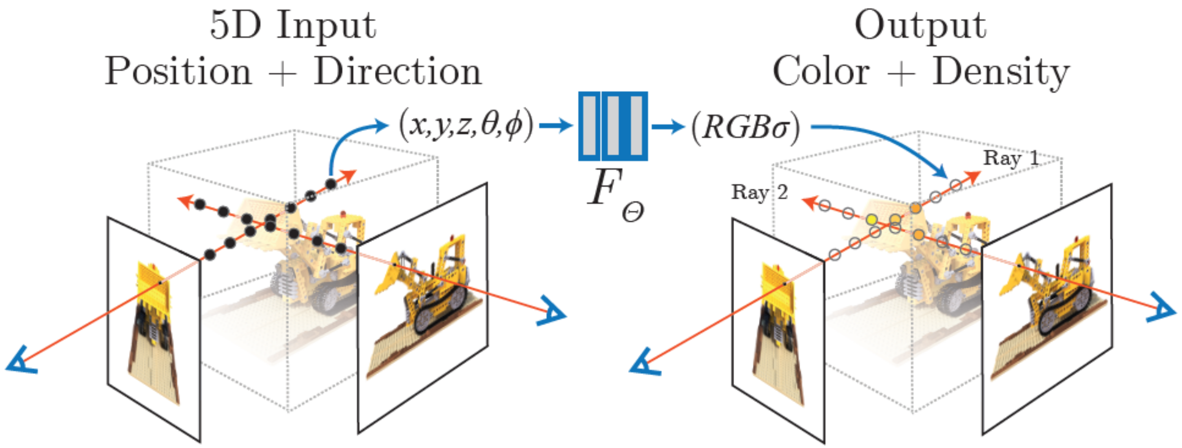
Neural Radiance Field (NeRF)



- Input: 5D coordinates (3D location & 2D viewing direction) along camera rays
- Output: Color and volume density produced by MLP from 5D coordinates

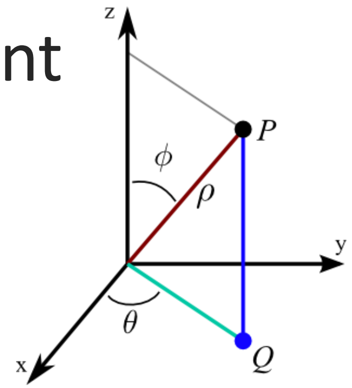
NeRF: **Input & Output**

- NeRF is a mapping function from 5D coordinates to colors and density

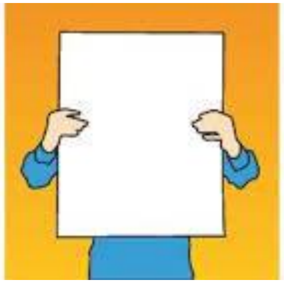


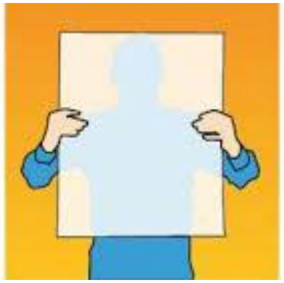
$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$

- $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$: 3D coordinates for each point
- $\mathbf{d} = (\theta, \phi) \in \mathbb{R}^2$: Viewing direction in 3D
- $\mathbf{c} = (R, G, B) \in \mathbb{R}^3$: RGB color channels
- $\sigma \in \mathbb{R}$: Volume density, [0, 1]



Viewing direction


High density


Low density

NeRF: **Projection**

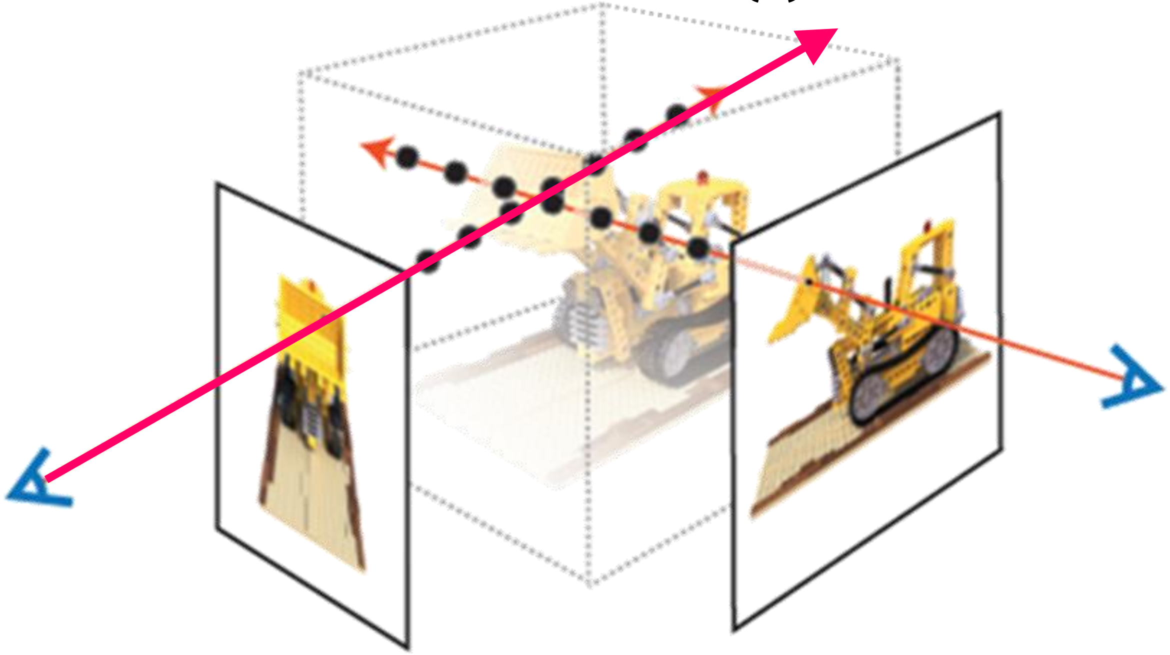
- Color Prediction: $\hat{C}(\mathbf{r})$

— The larger the density, the larger weight, $\sigma(\mathbf{r}(t))$

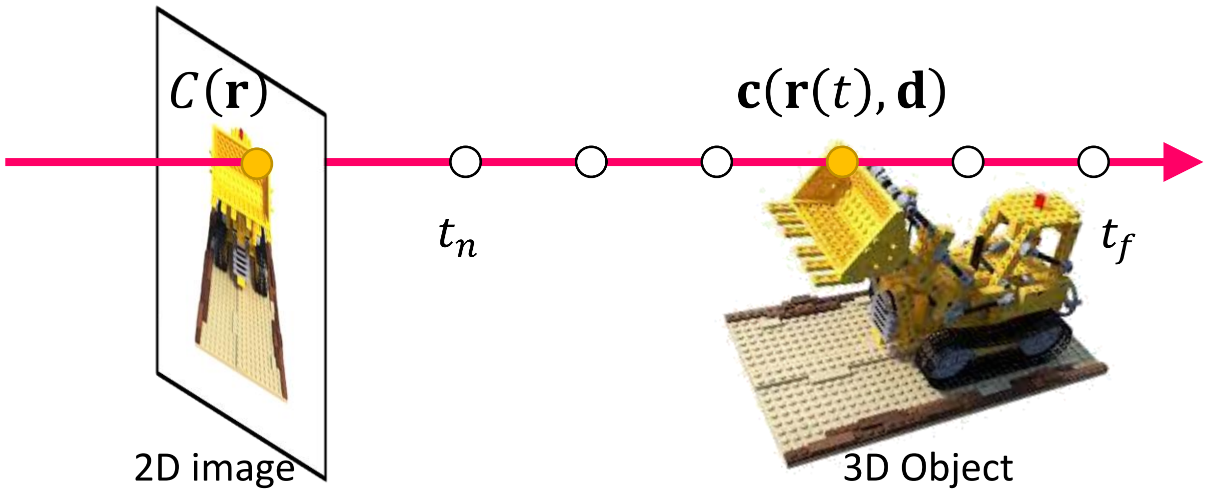
— The smaller the accumulated density, the larger weight, $T(t)$

$\mathbf{r}(t)$	: Camera ray, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$
$\mathbf{d}$	: 2D viewing direction, θ and ϕ
σ	: Volume density
T	: Probability that the ray travels without hitting other particles

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$



$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\boxed{\sigma(\mathbf{r}(t))}\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt$$



NeRF: **Projection**

- Color Prediction: $\hat{C}(\mathbf{r})$

— The larger the density, the larger weight, $\sigma\big(\mathbf{r}(t)\big)$

— The smaller the accumulated density, the larger weight, $T(t)$

$\mathbf{r}(t)$: Camera ray, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$

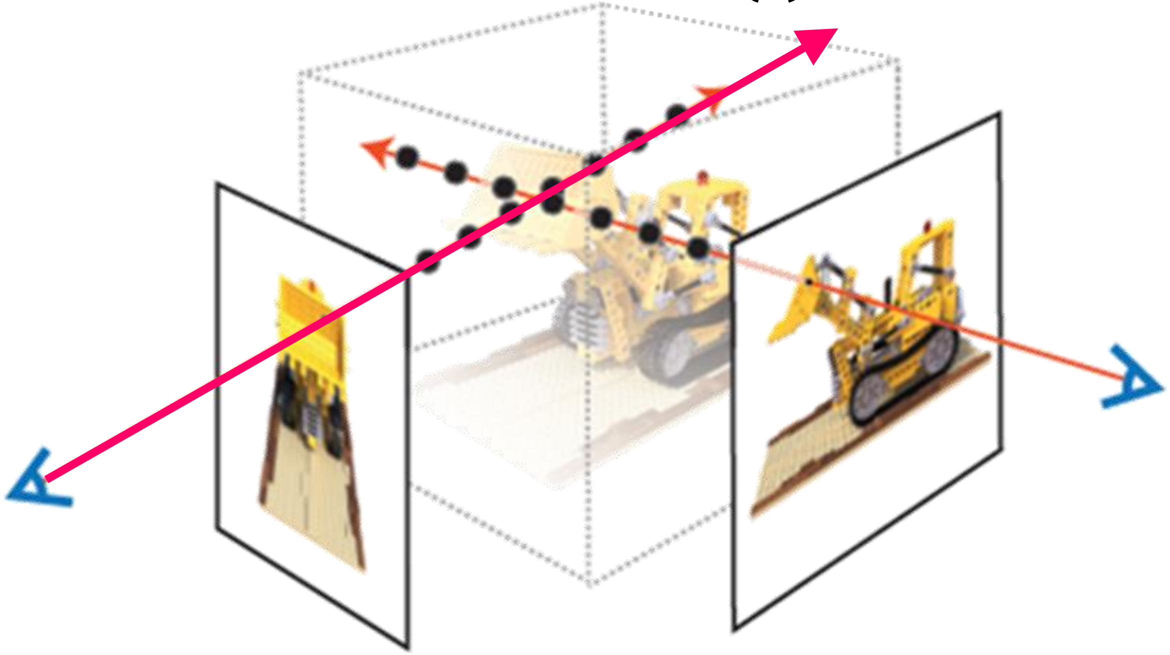
$\mathbf{d}$: 2D viewing direction, θ and ϕ

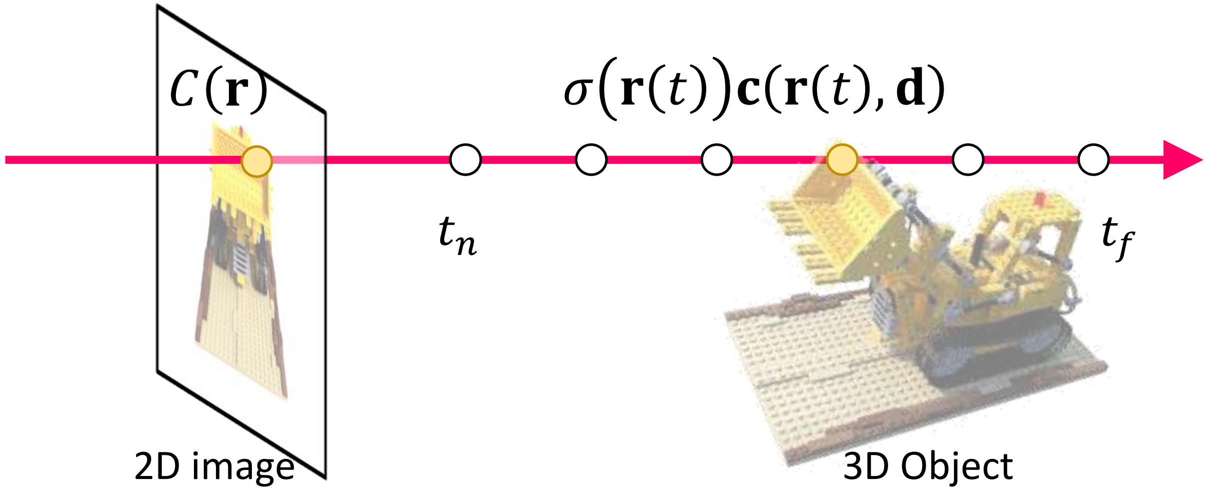
σ : Volume density

T : Probability that the ray travels without hitting other particles

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\boxed{\sigma\big(\mathbf{r}(t)\big)}\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt$$





NeRF: **Projection**

- Color Prediction: $\hat{C}(\mathbf{r})$

— The larger the density, the larger weight, $\sigma\big(\mathbf{r}(t)\big)$

— The smaller the accumulated density, the larger weight, $T(t)$

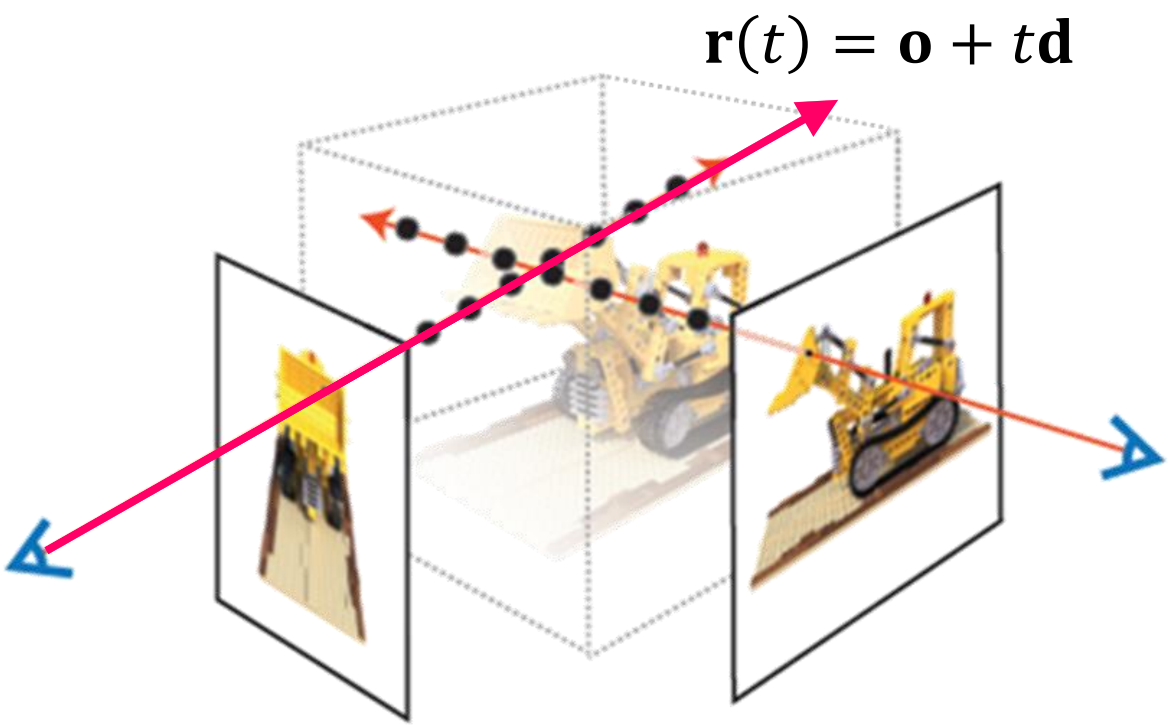
$\mathbf{r}(t)$: Camera ray, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$

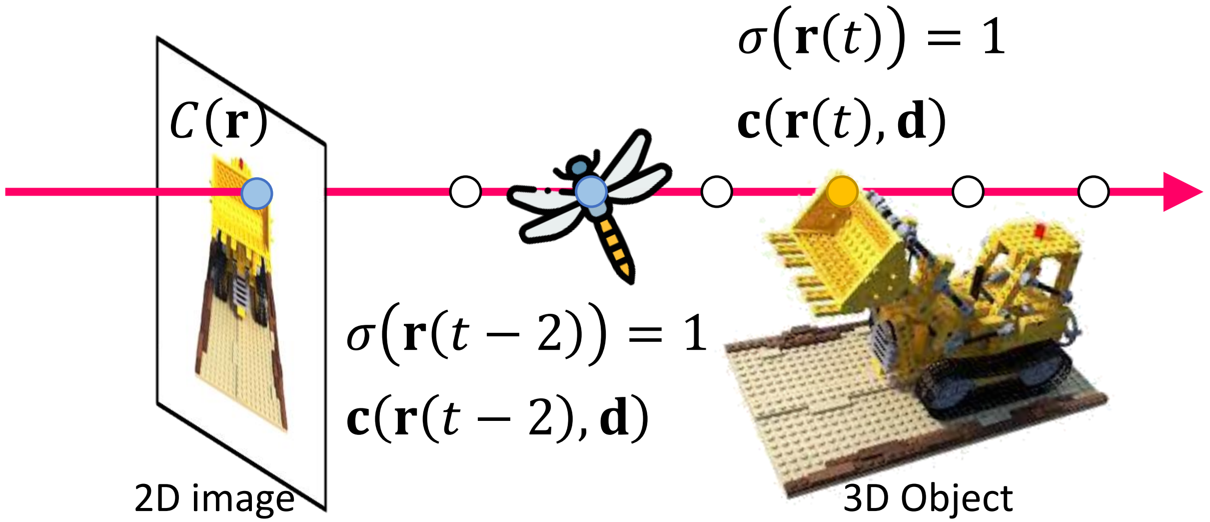
$\mathbf{d}$: 2D viewing direction, θ and ϕ

σ : Volume density

T : Probability that the ray travels without hitting other particles

$$C(\mathbf{r}) = \int_{t_n}^{t_f} \boxed{T(t)}\,\sigma\big(\mathbf{r}(t)\big)\mathbf{c}\big(\mathbf{r}(t), \mathbf{d}\big)\, dt$$





NeRF: **Projection**

$\mathbf{r}(t)$: Camera ray, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$

$\mathbf{d}$: 2D viewing direction, θ and ϕ

σ : Volume density

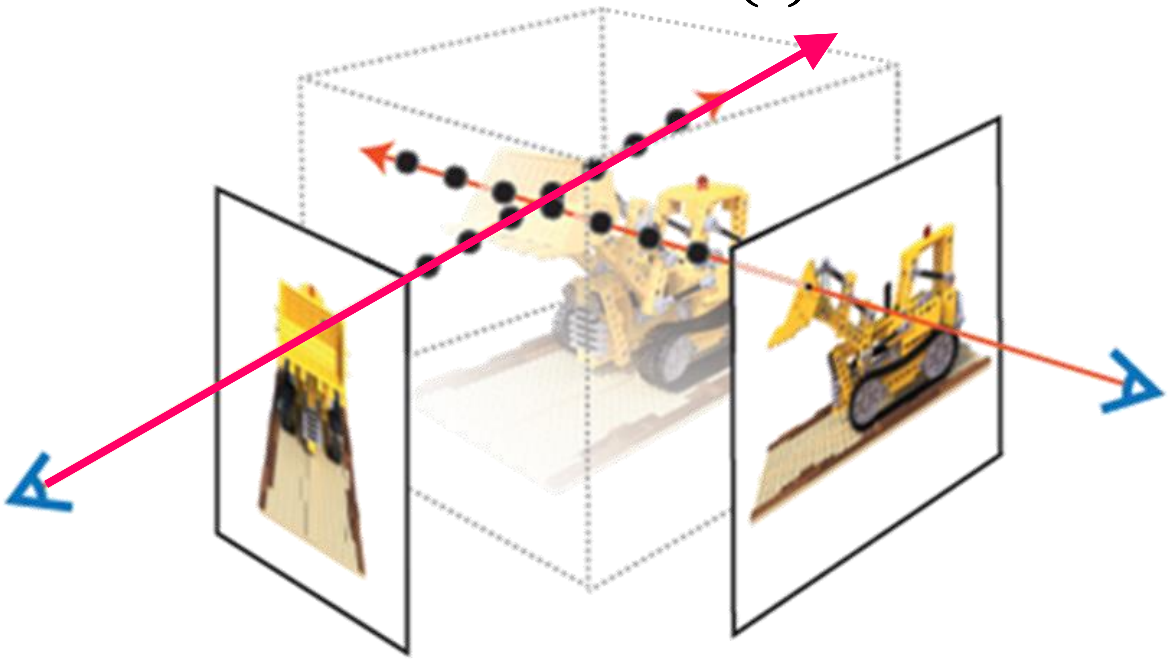
T : Probability that the ray travels without hitting other particles

- Color Prediction: $\hat{C}(\mathbf{r})$

— The larger the density, the larger weight, $\sigma(\mathbf{r}(t))$

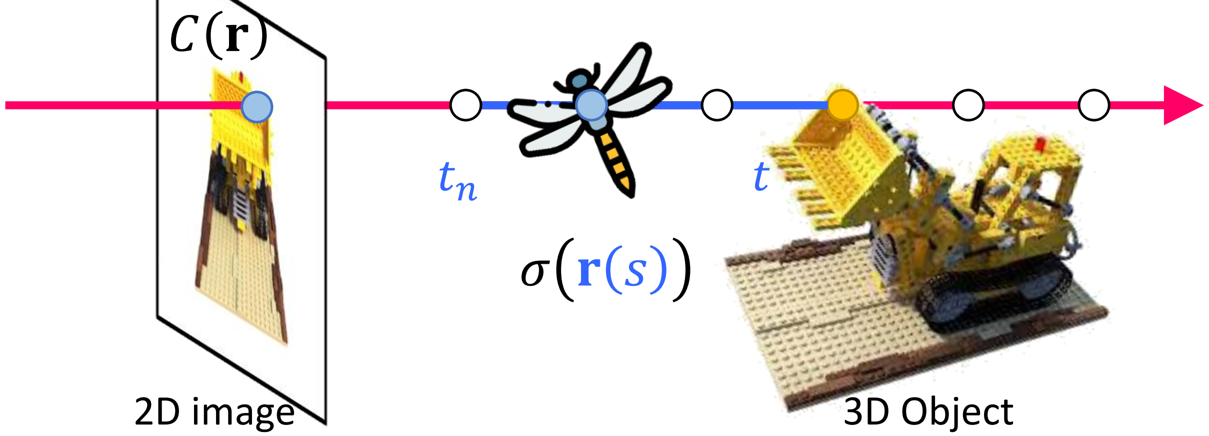
— The smaller the accumulated density, the larger weight, $T(t)$

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$



$$C(\mathbf{r}) = \int_{t_n}^{t_f} \boxed{T(t)}\,\sigma(\mathbf{r}(t))\,\mathbf{c}(\mathbf{r}(t), \mathbf{d})\,dt$$

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,ds\right)$$



NeRF: **Loss Function**

- MSE loss between the ground-truth color and the predicted color

— Ground-Truth Color

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma\big(\mathbf{r}(t)\big)\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt$$

— Predicted Color

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i\big(1 - \exp(-\sigma_i\delta_i)\big)\mathbf{c}_i \qquad \text{where} \quad T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right]$$

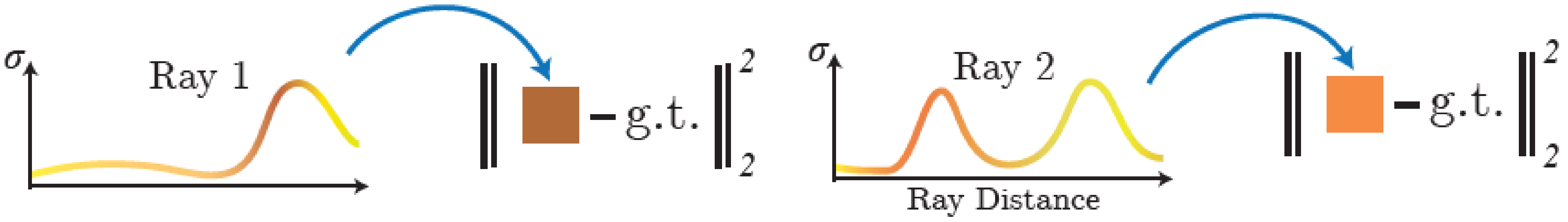
NeRF: **Hierarchical Volume Sampling**

- Train Coarse Network with Uniform Sampling (N_c)

— Obtain the distribution of initial $T(t)$ for each color point $\mathbf{c}(\mathbf{r}(t))$

- Train Fine Network with Uniform Sampling and Adaptive Sampling ($N_c + N_f$)

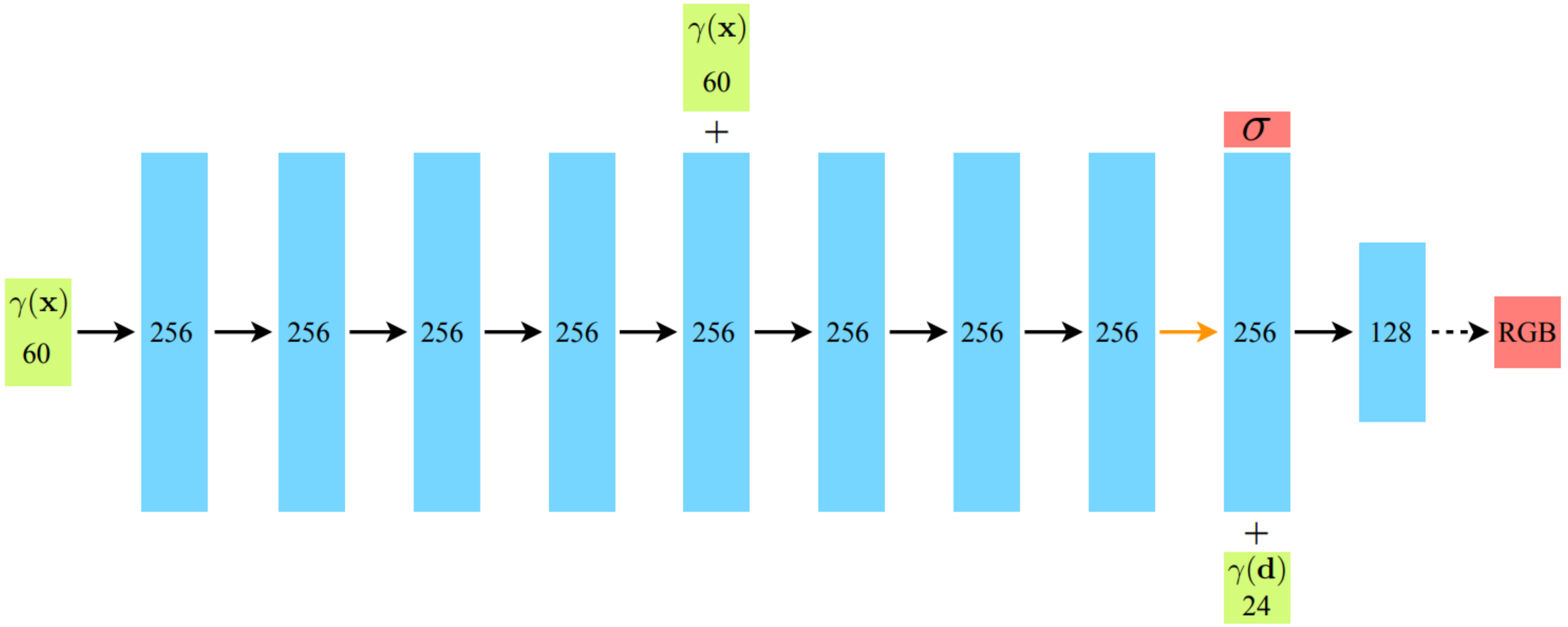
— Obtain the distribution of refined $T(t)$ for each color point $\mathbf{c}(\mathbf{r}(t))$



NeRF: **Positional Encoding**

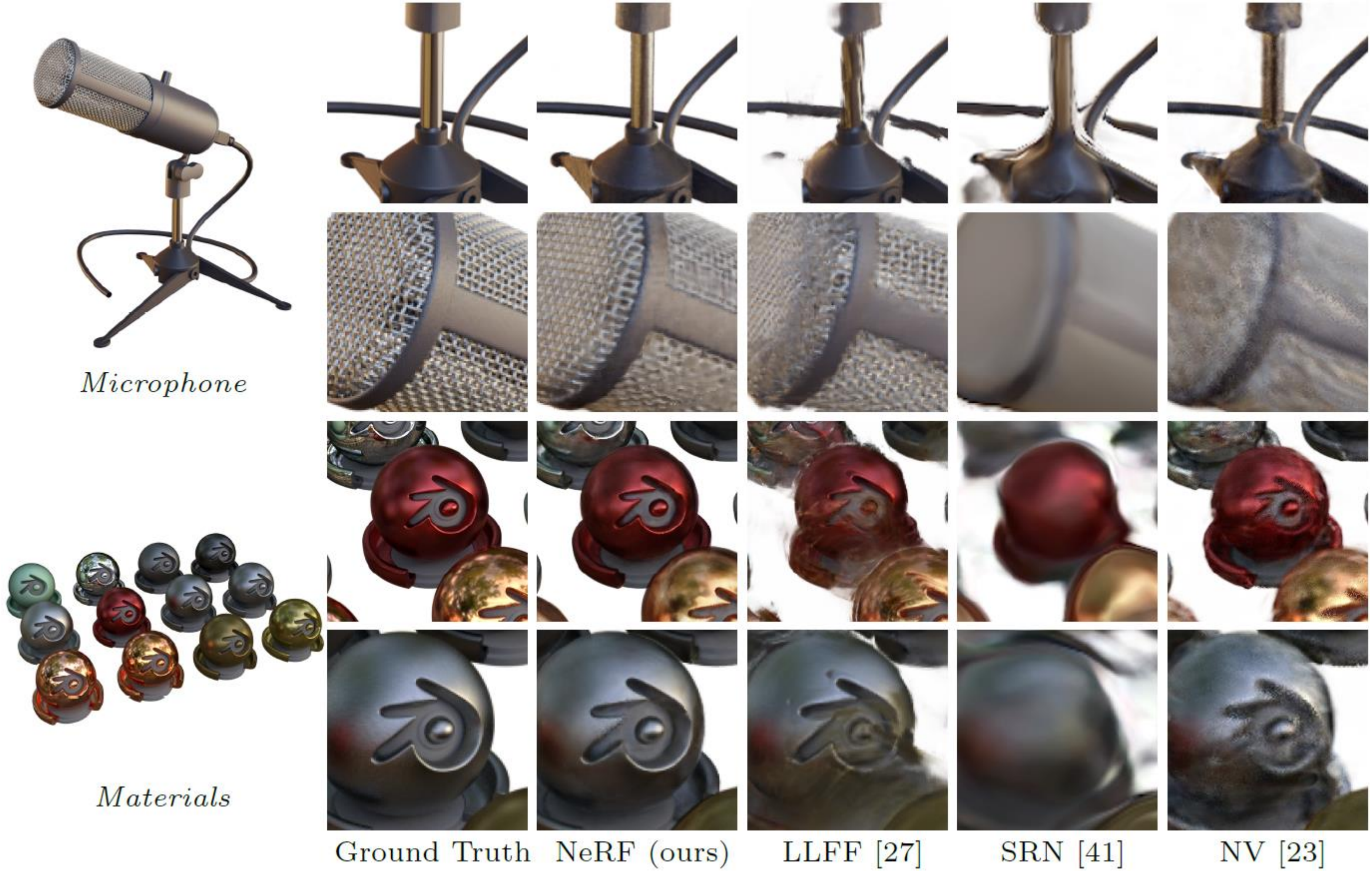
- In NeRF, the periodic function is used to map low dimensional continuous input coordinates into a higher dimensional space to enable the MLP to more easily approximate a higher frequency function

$$\gamma(p) = [\sin(2^0\pi p), \cos(2^0\pi p), \cdots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p),]$$



Experiments: **DEMO**

Experiments: **Qualitative Evaluation**



Experiments: **Ablation Study**

	Input	#Im.	L	(N_c, N_f)	PSNR↑	SSIM↑	LPIPS↓
1) No PE, VD, H	xyz	100	-	(256, -)	26.67	0.906	0.136
2) No Pos. Encoding	$xyz\theta\phi$	100	-	(64, 128)	28.77	0.924	0.108
3) No View Dependence	xyz	100	10	(64, 128)	27.66	0.925	0.117
4) No Hierarchical	$xyz\theta\phi$	100	10	(256, -)	30.06	0.938	0.109
5) Far Fewer Images	$xyz\theta\phi$	25	10	(64, 128)	27.78	0.925	0.107
6) Fewer Images	$xyz\theta\phi$	50	10	(64, 128)	29.79	0.940	0.096
7) Fewer Frequencies	$xyz\theta\phi$	100	5	(64, 128)	30.59	0.944	0.088
8) More Frequencies	$xyz\theta\phi$	100	15	(64, 128)	30.81	0.946	0.096
9) Complete Model	$xyz\theta\phi$	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

Table 2: An ablation study of our model. Metrics are averaged over the 8 scenes from our realistic synthetic dataset. See Sec. 6.4 for detailed descriptions.

Summary: NeRF

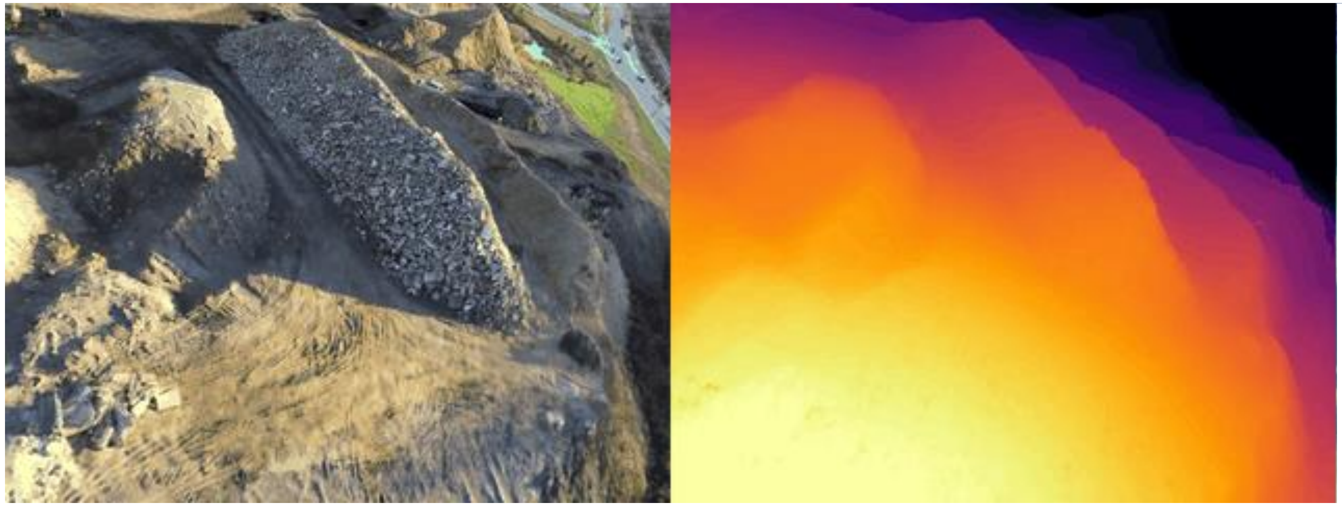
- A NeRF model stores a volumetric scene representation as the weights of an MLP, trained on many images with known pose
- One of the reasons NeRF is able to render with great detail is because it encodes a 3D point and associated view direction on a ray using periodic activation functions, i.e., *Fourier Features*
- Vanilla NeRF left many opportunities to improve upon:

— It is slow both for training and rendering; It can only represent static scenes; It bakes in lighting; A trained NERF does not generalize to other scenes

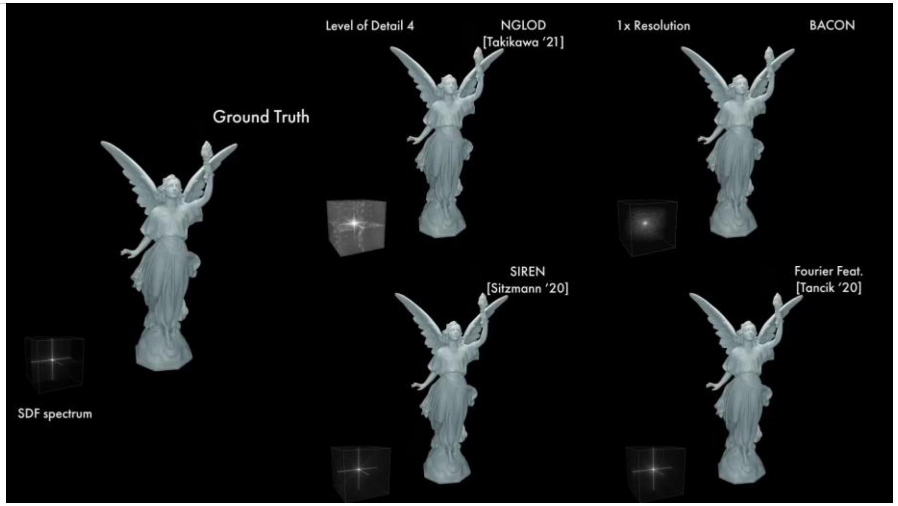
Topics: **Multi-Scale Representations**

Mip-NeRF [ICCV`21]

Mip-NeRF 360 [CVPR`22]

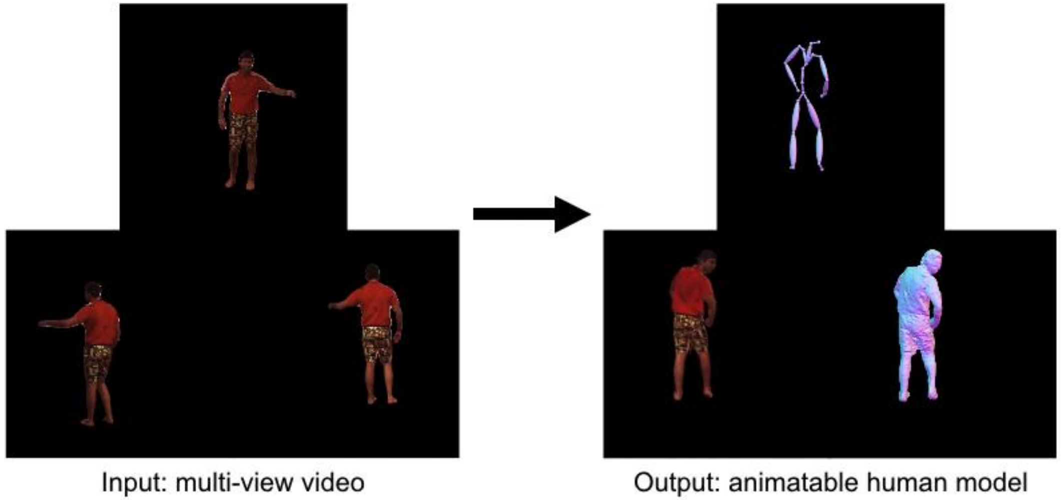


Mega-NeRF [CVPR`22]

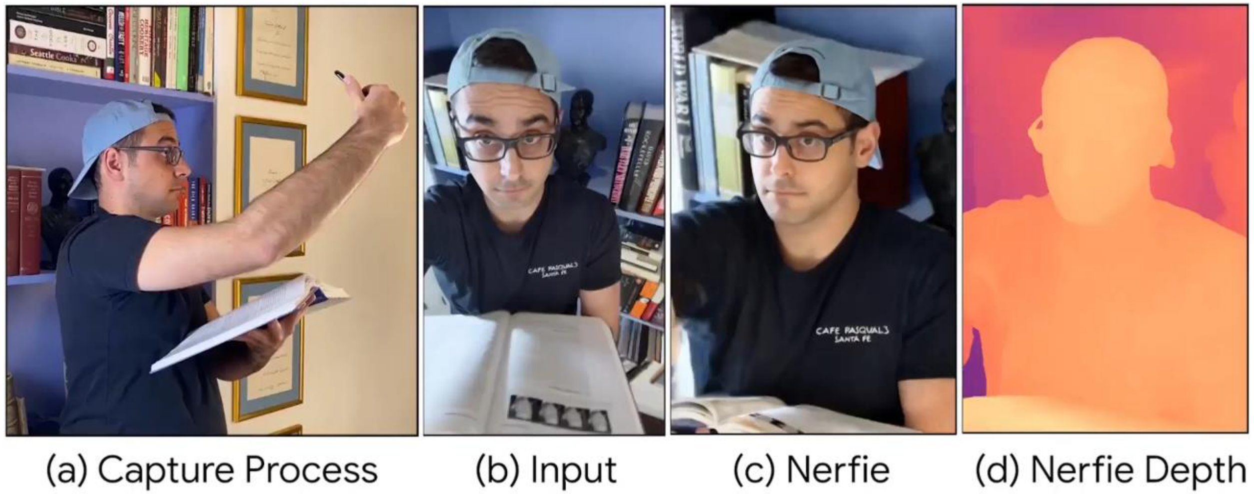


BACON [CVPR`22]

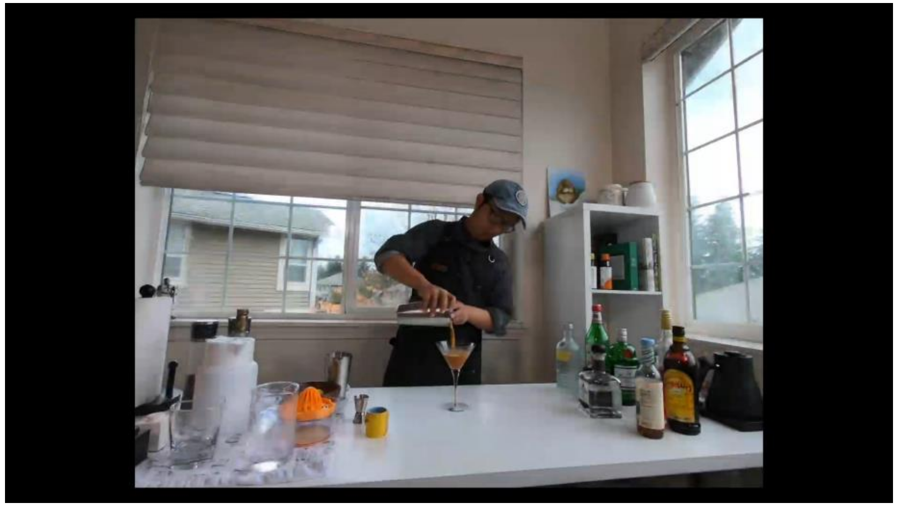
Topics: **Deformable & Video**



Animatable NeRF [ICCV`21]

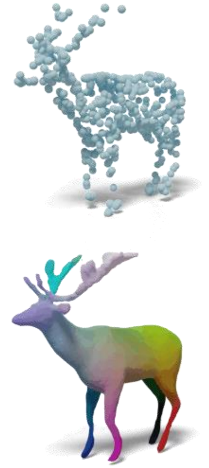


Nerfies [ICCV`21]



Neural 3D Video Synthesis [CVPR`22]

BANMo [CVPR`22]

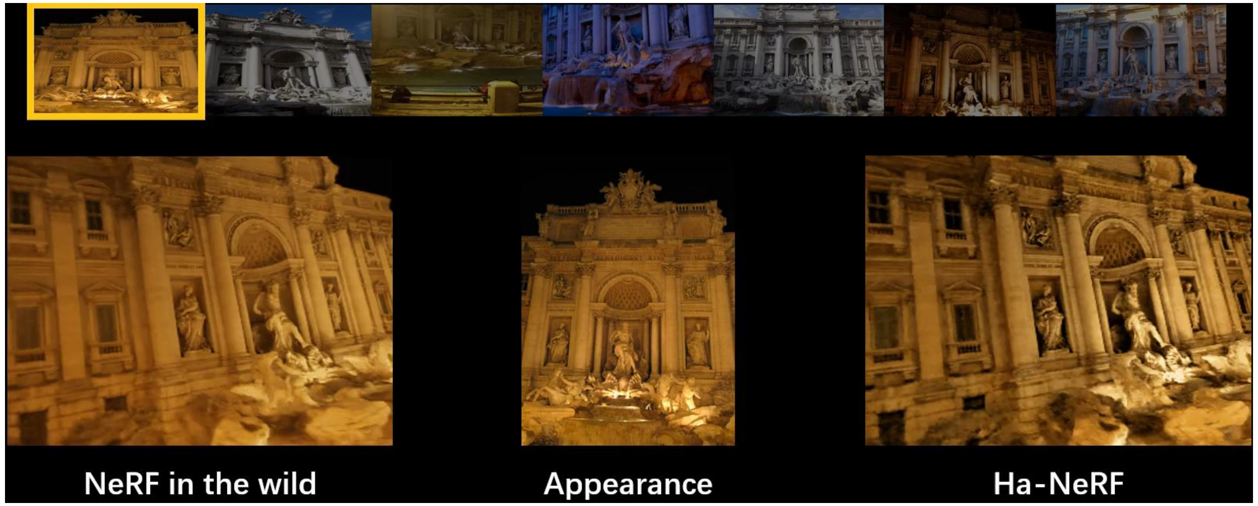


CaDeX [CVPR`22]

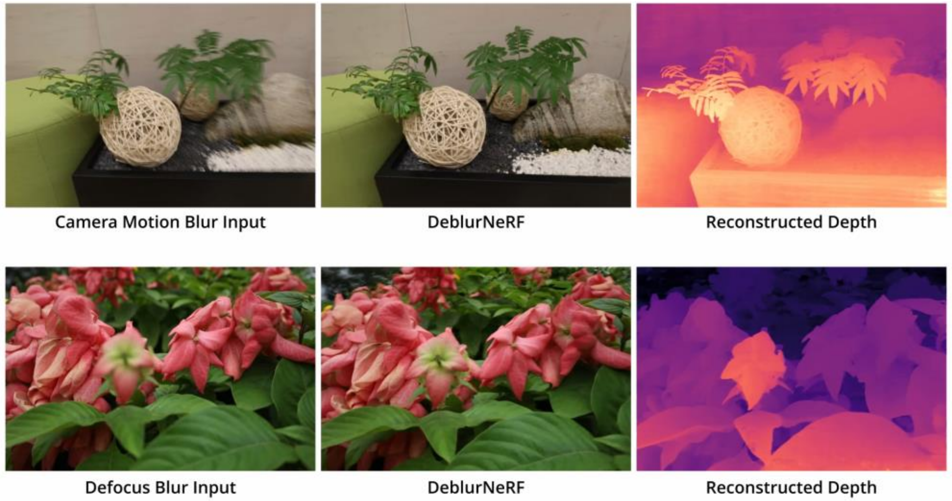
Topics: 3D Rendering **In The Wild**

NeRF in the Wild [CVPR'21]

Occlusion-aware NeuRay [CVPR'22]

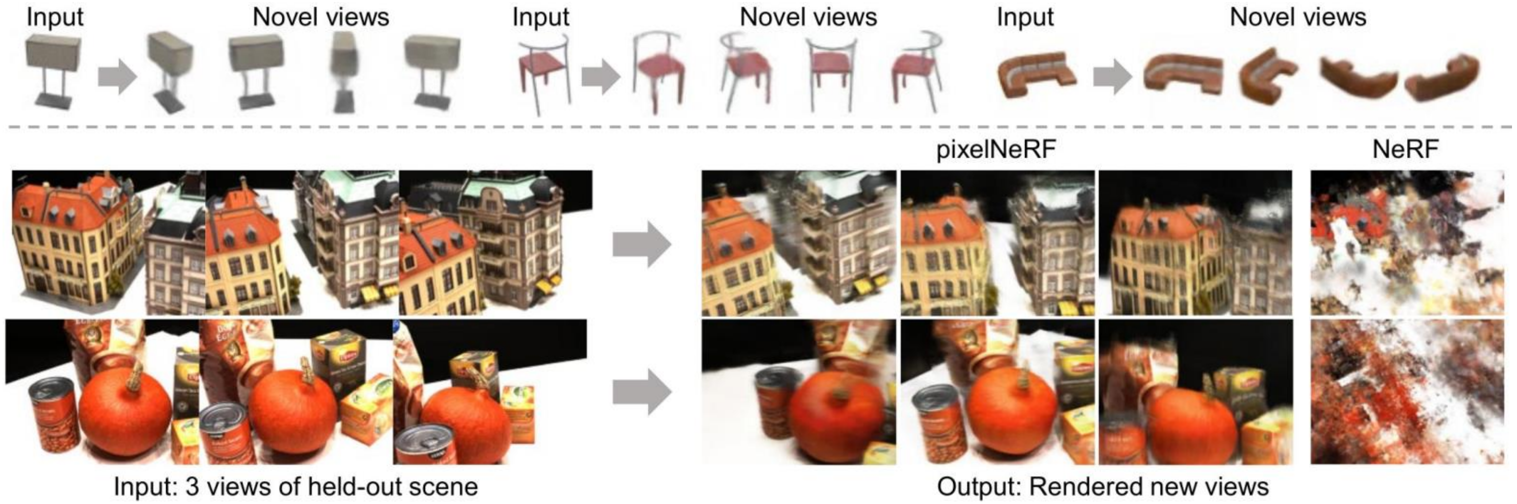


Hallucinated NeRF in the Wild [CVPR'22]



Deblur-NeRF [CVPR'22]

Topics: 3D Rendering **From Sparse Images**



pixelNeRF [CVPR'21]

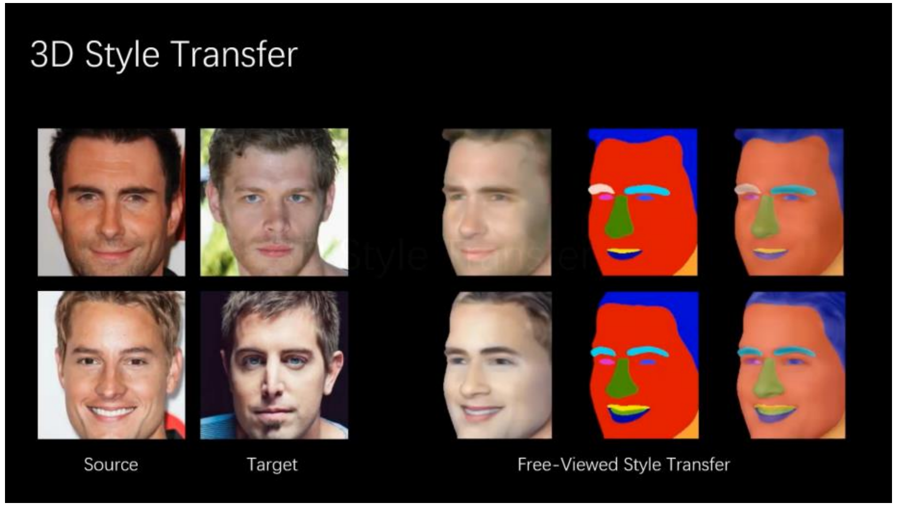
LOLNeRF [CVPR'22]

NeRS [NeurIPS'21]

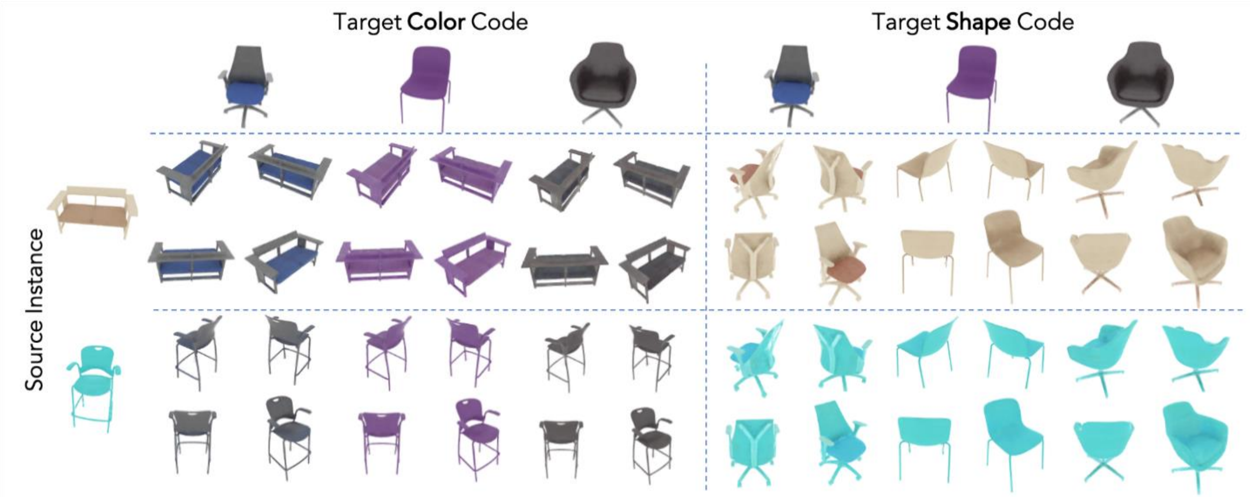
RegNeRF [CVPR'22]

Topics: **Stylized & Editable**

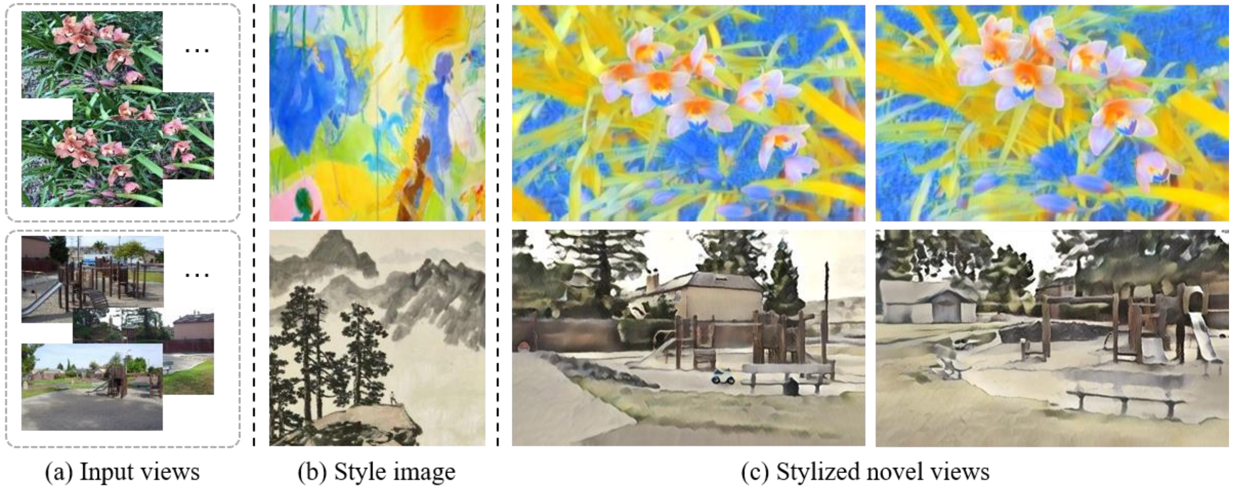
CLIP-NeRF [CVPR’22]



Face Editing in NeRF [CVPR’22]



Editing Conditional Radiance Fields [ICCV’21]



Stylized NeRF [CVPR’22]

Practice: Neural Rendering with NeRF

- Practice

— Neural Radiance Fields (NeRF) for Representing 3D Scenes