Chapter 18
Neural Radiance Field

Lin ZHANG
School of Computer Science and Technology
Tongji University

Outline

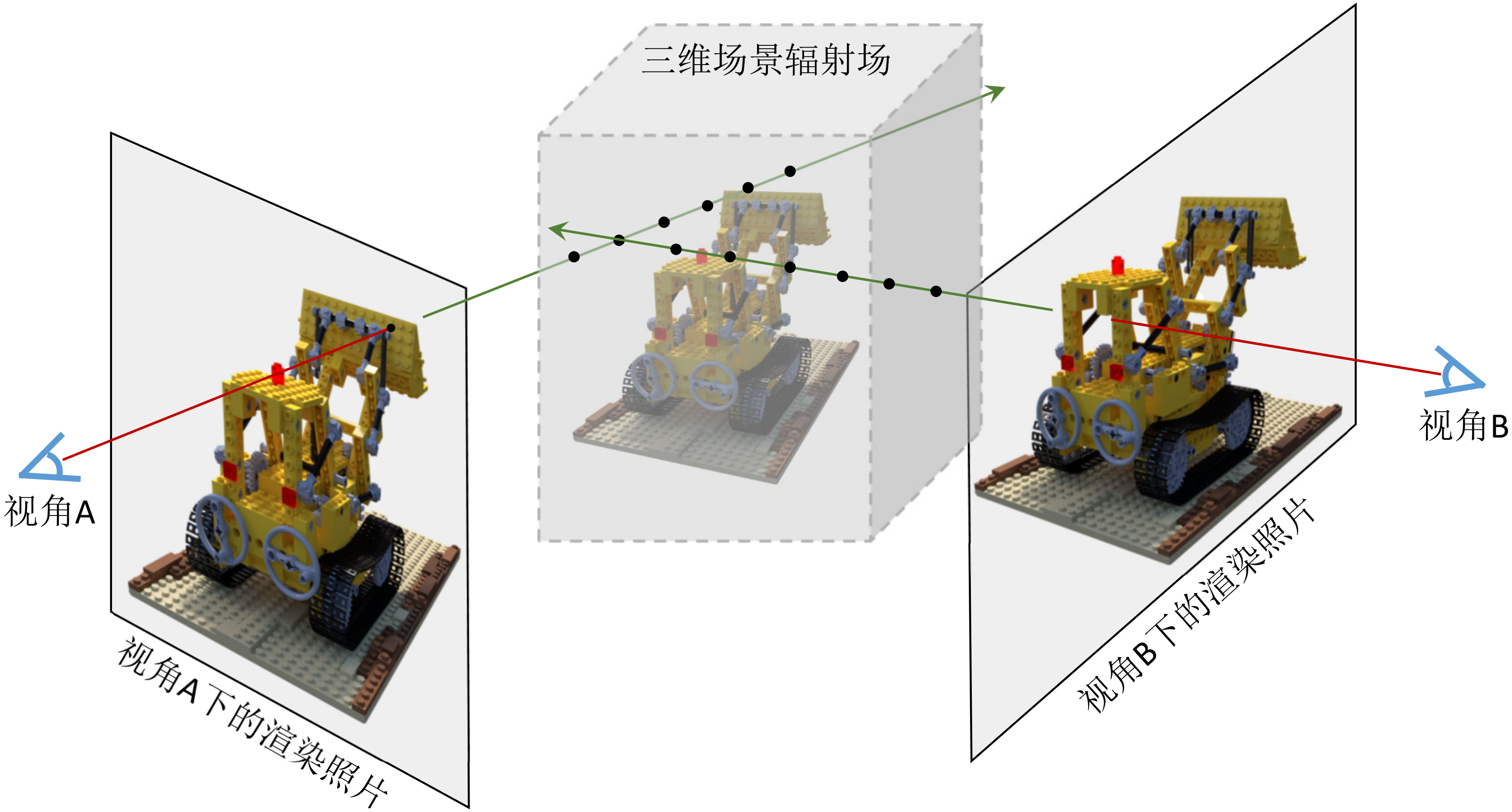
- Rendering based on Radiance field
  - Continuous form
  - Discrete form
- Neural Radiance Field (NeRF) and Its Training

Rendering based on radiance field—Continuous form

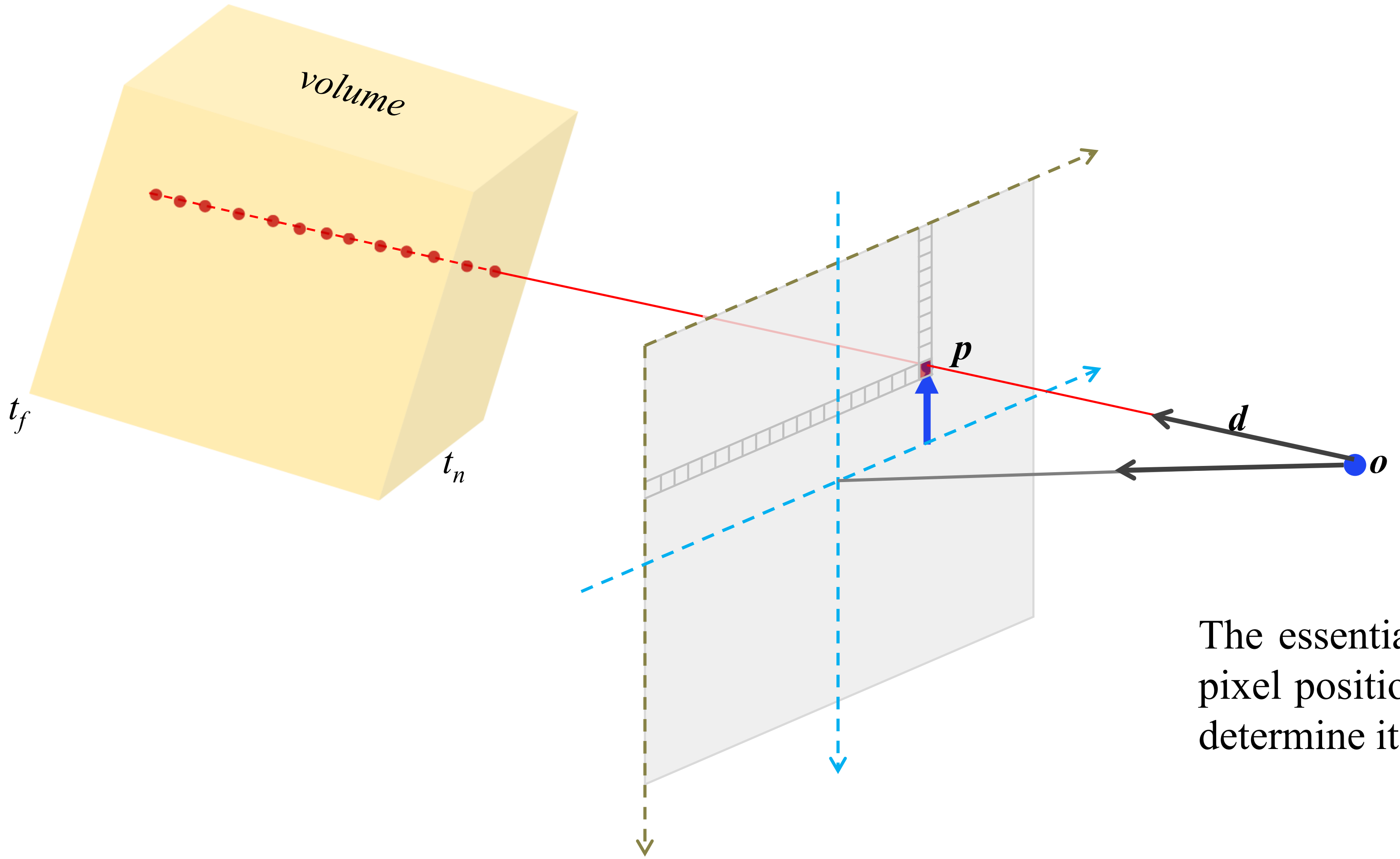
Radiance field based rendering

- This rendering method treats the physical space containing the scene as a radiance field: each point in the space possesses two attributes, **opacity and view-dependent color**
- The goal of the rendering process is to generate an image of the scene from the perspective of a virtual camera
- The viewpoint of the virtual camera is defined by its pose in the world coordinate system. The essence of the rendering operation is to determine the color of each point on the imaging plane of the virtual camera.

Rendering based on radiance field—Continuous form



Rendering based on radiance field—Continuous form



The essential question is: for a given pixel position $\boldsymbol{p}$ on the image, how to determine its color?

Rendering based on radiance field—Continuous form

$\boldsymbol{p}$'s color can be calculated as,

$$\hat{\boldsymbol{u}}(\boldsymbol{p}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\mathbf{c}(\boldsymbol{r}(t), \boldsymbol{d})dt \quad (1)$$

$\sigma(\boldsymbol{x})$ represents the opacity at the spatial location $\boldsymbol{x}$

$\boldsymbol{r}(t) = \boldsymbol{o}+t\boldsymbol{d}$ ($t \in [t_n, t_f]$) is one point along the ray $\overrightarrow{\boldsymbol{op}}$, parameterized by t; $\boldsymbol{o}$ is the camera center; $\boldsymbol{d}$ is the unit vector of the ray $\overrightarrow{\boldsymbol{op}}$

$\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})$ is the color of the spatial position $\boldsymbol{r}(t)$ when viewed along the direction $\boldsymbol{d}$

$T(t)$ is the accumulated opacity along the ray $\overrightarrow{\boldsymbol{op}}$ from the point $\boldsymbol{r}(t_n)$ to $\boldsymbol{r}(t)$,

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

Rendering based on radiance field—Discrete form

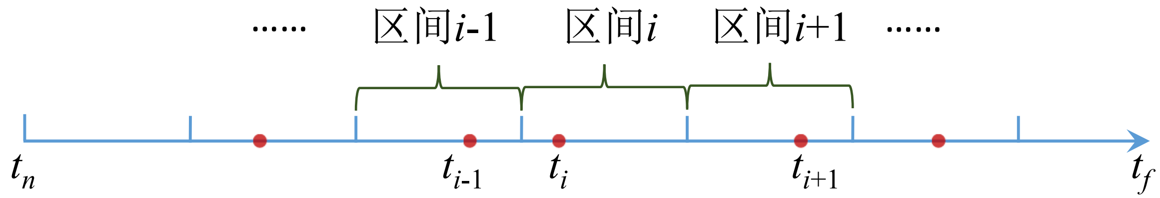
With Eq. (1), to compute the color $\boldsymbol{u}(\boldsymbol{p})$, we need to compute an integral along the ray $\overrightarrow{\boldsymbol{op}}$, which is not easy to be implemented

Need to find a way to convert
the integral to summation

The discrete form of the radiance field based rendering

Rendering based on radiance field—Discrete form

Divide the range [t_n, t_f] to N sub-ranges,



The ith sub-range is $\left[t_n+\frac{i-1}{N}\left(t_f-t_n\right),t_n+\frac{i}{N}\left(t_f-t_n\right)\right],i=1,\dots,N$

In sub-range i, randomly sample one point t_i whose value conforms to a uniform distribution,

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

Then, Eq. (1) can be approximated as,

$$\hat{\boldsymbol{u}}(\boldsymbol{p}) \approx \sum_{i=1}^{N-1}\hat{\boldsymbol{u}}_i(\boldsymbol{p}) = \sum_{i=1}^{N-1}\int_{t_i}^{t_{i+1}} T(t)\sigma\left(\boldsymbol{r}(t)\right)\boldsymbol{c}\left(\boldsymbol{r}(t),\boldsymbol{d}\right)dt \tag{2}$$

Rendering based on radiance field—Discrete form

By considering the opacity and the color in the sub-range $[t_i, t_{i+1}]$ as constants,

$$\sigma_i = \sigma\left(\boldsymbol{r}\left(t_i\right)\right) \qquad \boldsymbol{c}_i = \boldsymbol{c}\left(\boldsymbol{r}\left(t_i\right), \boldsymbol{d}\right)$$

We can have,

$$\hat{\boldsymbol{u}}_i\left(\boldsymbol{p}\right) = \int_{t_i}^{t_{i+1}} T\left(t\right)\sigma\left(\boldsymbol{r}\left(t\right)\right)\boldsymbol{c}\left(\boldsymbol{r}\left(t\right), \boldsymbol{d}\right)dt \approx \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_n}^{t} \sigma\left(s\right)ds\right)\sigma_i \boldsymbol{c}_i dt \qquad (3)$$

$$\begin{aligned}\hat{\boldsymbol{u}}_i\left(\boldsymbol{p}\right) &\approx \sigma_i \boldsymbol{c}_i \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_n}^{t} \sigma\left(s\right)ds\right)dt = \sigma_i \boldsymbol{c}_i \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_n}^{t_i} \sigma\left(s\right)ds\right)\exp\left(-\int_{t_i}^{t} \sigma\left(s\right)ds\right)dt \qquad (4)\\ &= \sigma_i \boldsymbol{c}_i T_i \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_i}^{t} \sigma\left(s\right)ds\right)dt\end{aligned}$$

$$\int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_i}^{t} \sigma\left(s\right)ds\right)dt \approx \int_{t_i}^{t_{i+1}} \exp\left(-\sigma_i\left(t - t_i\right)\right)dt = \left.\frac{\exp\left(-\sigma_i\left(t - t_i\right)\right)}{-\sigma_i}\right|_{t_i}^{t_{i+1}} = \frac{1}{\sigma_i}\left[1 - \exp\left(-\sigma_i\left(t_{i+1} - t_i\right)\right)\right]$$

$$\hat{\boldsymbol{u}}_i\left(\boldsymbol{p}\right) \approx \sigma_i \boldsymbol{c}_i T_i \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_i}^{t} \sigma\left(s\right)ds\right)dt \approx \sigma_i \boldsymbol{c}_i T_i \frac{1}{\sigma_i}\left[1 - \exp\left(-\sigma_i\left(t_{i+1} - t_i\right)\right)\right] = \boldsymbol{c}_i T_i\left[1 - \exp\left(-\sigma_i \underbrace{\left(t_{i+1} - t_i\right)}_{\delta_i}\right)\right]$$

Rendering based on radiance field—Discrete form

We can have,

$$\hat{\boldsymbol{u}}(\boldsymbol{p}) \approx \sum_{i=1}^{N-1} \hat{\boldsymbol{u}}_i(\boldsymbol{p}) = \sum_{i=1}^{N-1} \boldsymbol{c}_i T_i \left[1-\exp(-\sigma_i \delta_i)\right] \tag{5}$$

Discrete the computation of T_i,

$$T_i = \exp\left(-\int_{t_n}^{t_i} \sigma(s)\,ds\right) \approx \exp\left(-\left[\sigma_1(t_2-t_1)+\sigma_2(t_3-t_2)+\ldots+\sigma_{i-1}(t_i-t_{i-1})\right]\right)$$

$$= \exp\left(-\sum_{j=1}^{i-1} \sigma_j (t_{j+1}-t_j)\right) = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \tag{6}$$

Eq. (5) and Eq. (6) together give the discrete form of the continuous rendering model described by Eq. (1)

Outline

- Rendering based on Radiance field
  - Continuous form
  - Discrete form
- Neural Radiance Field (NeRF) and Its Training

NeRF and its training

Radiance field can be naturally represented as a mapping function,

$$F_{\Theta}\left(\boldsymbol{x},\boldsymbol{d}\right)\rightarrow\left(\boldsymbol{c},\sigma\right)$$

where F_{Θ} is the mapping function, Θ is the parameter set of F; $\boldsymbol{x}$ is a 3D point in the radiance field while $\boldsymbol{d}$ is the viewing direction; $\boldsymbol{\sigma}$ is the opacity of $\boldsymbol{x}$ in the radiance field, and $\boldsymbol{c}=(r,g,b)$ is the color of $\boldsymbol{x}$ when viewed along the direction of $\boldsymbol{d}$

For a scene to be rendered, if its associated mapping model F_{Θ} is fixed, its radiance field is fixed or in other words, its radiance field is implicitly represented by F_{Θ}

F_{Θ} can be naturally represented as a neural network

NeRF and its training

Theoretically, F_Θ maps a 6*d* vector (***x***, ***d***) to a 4*d* vector (***c***, σ). However, such a straightforward idea cannot characterize high-frequency information well

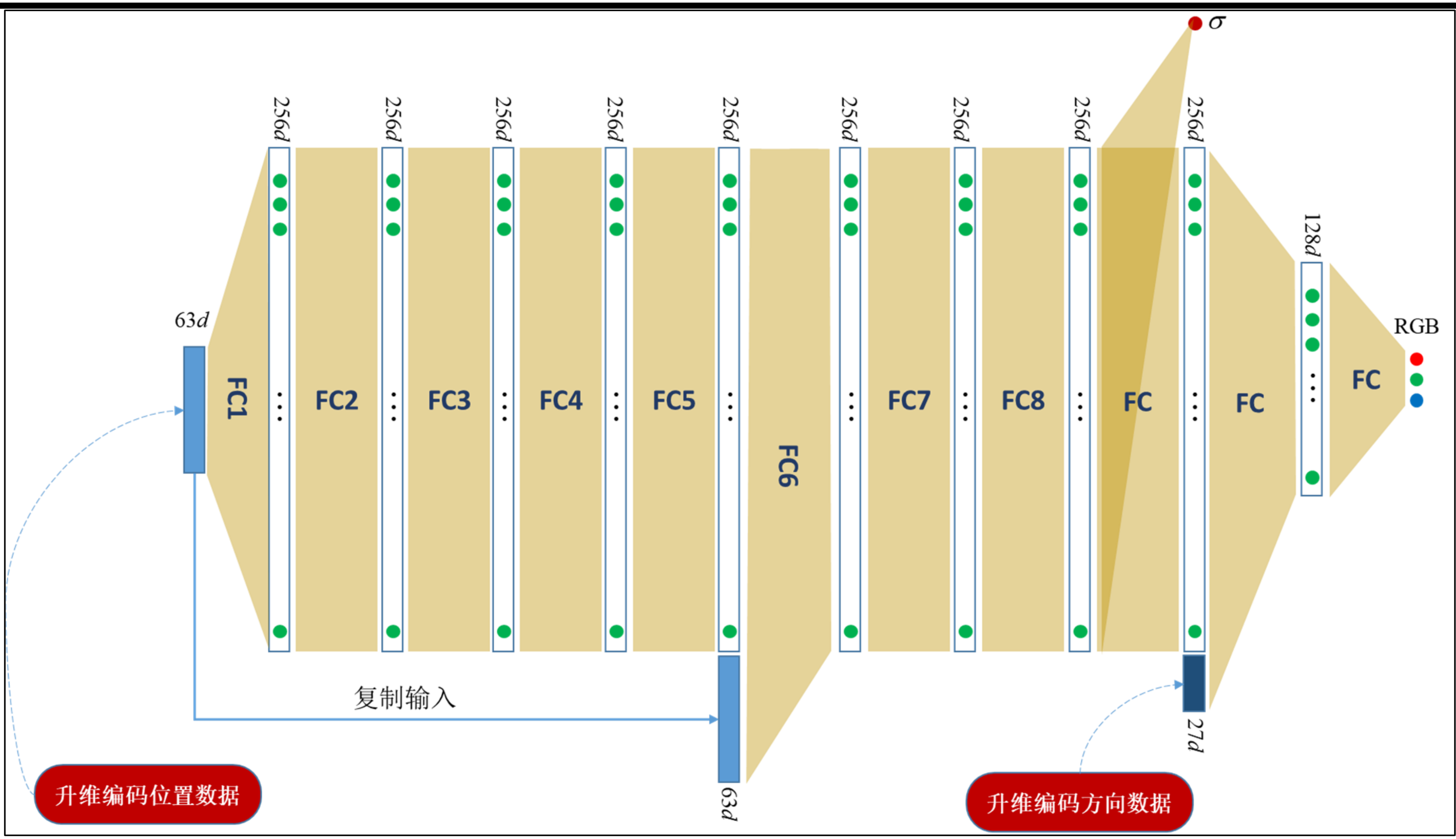
Dimensionality-increasing encoding for the position ***x*** and the direction ***d***

Such a process can be represented as $\gamma(p): \mathbb{R} \rightarrow \mathbb{R}^{2L}$:

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \ldots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right)\right)$$

In the reference implementation, for the position encoding, L=10. Thus, after such a dimensionality-increasing encoding process, the data dimension is 2×10×3+3=63. For the direction encoding, L=4. Thus, after such a dimensionality-increasing encoding process, the data dimension related to dimension is 2×4×3+3=27

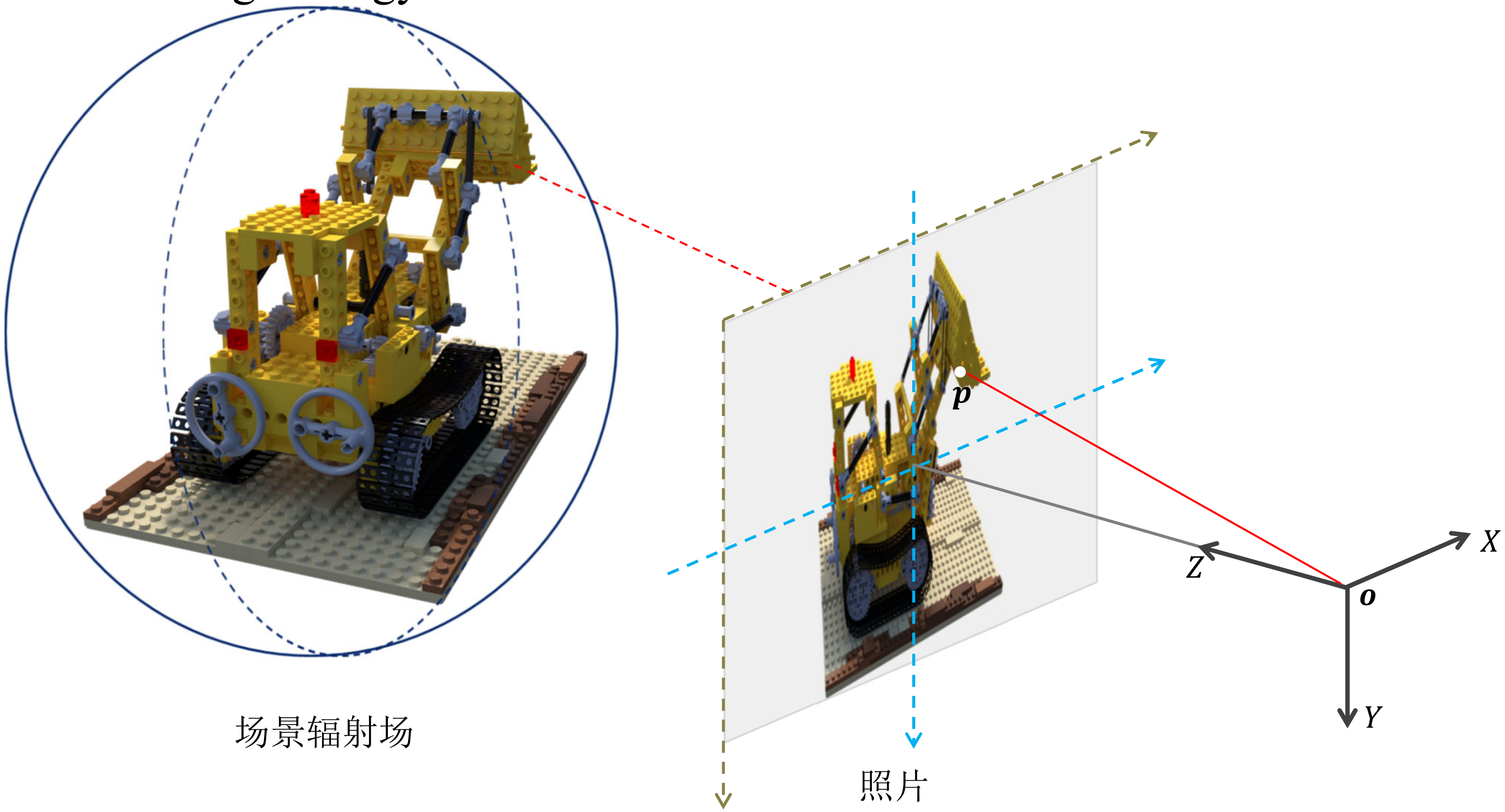
NeRF and its training



Neural network for implicitly representing the radiance field

NeRF and its training

- The training strategy of NeRF



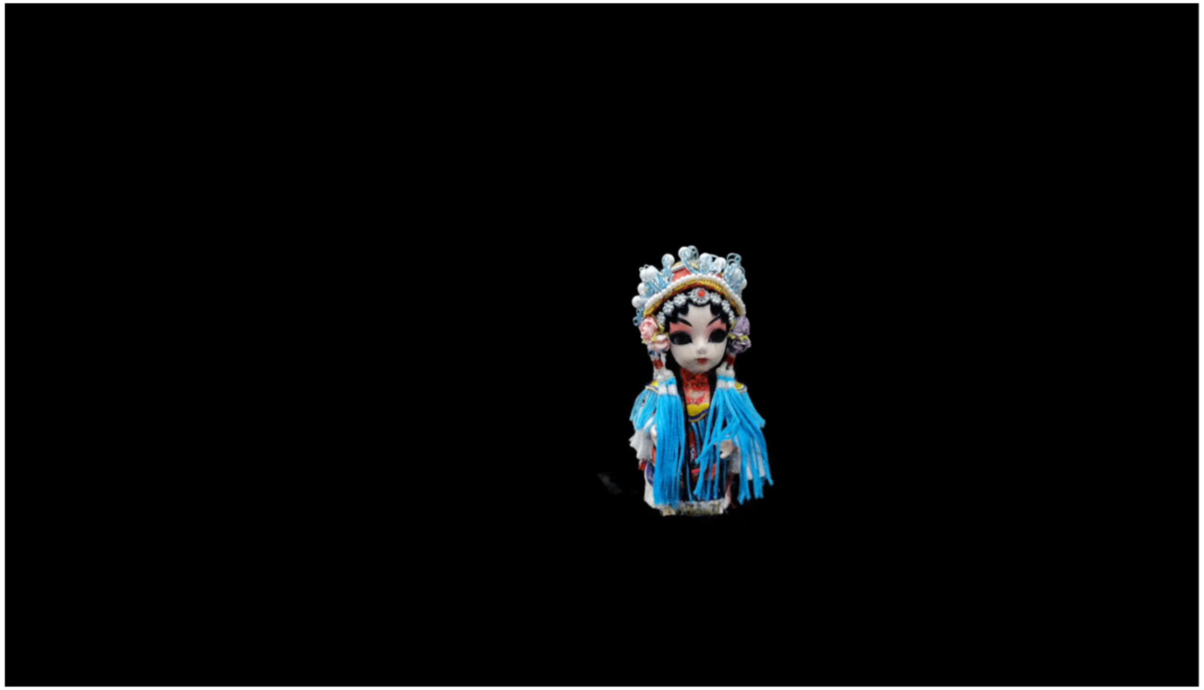
NeRF and its training

- The training strategy of NeRF
  - Collect a set of images $\mathcal{P}$ for the scene to be rendered; for each image in $\mathcal{P}$, the associated intrinsics and extrinsics of the camera capturing it are known
  - The training loss for F_{Θ} is designed as,

$$l(\Theta) = \sum_{\boldsymbol{p} \in \Omega} \left\| \hat{\boldsymbol{u}}(\boldsymbol{p}) - \boldsymbol{u}(\boldsymbol{p}) \right\|_2^2$$

    where Ω is the pixel set from the image set $\mathcal{P}$; $\boldsymbol{u}(\boldsymbol{p})$ is the ground-truth color of pixel $\boldsymbol{p}$; $\hat{\boldsymbol{u}}(\boldsymbol{p})$ is the estimated color for the pixel $\boldsymbol{p}$ with the radiance field F_{Θ}

NeRF rendering results




Q+A