Simulation of dynamic obstacles on the path of an automated car

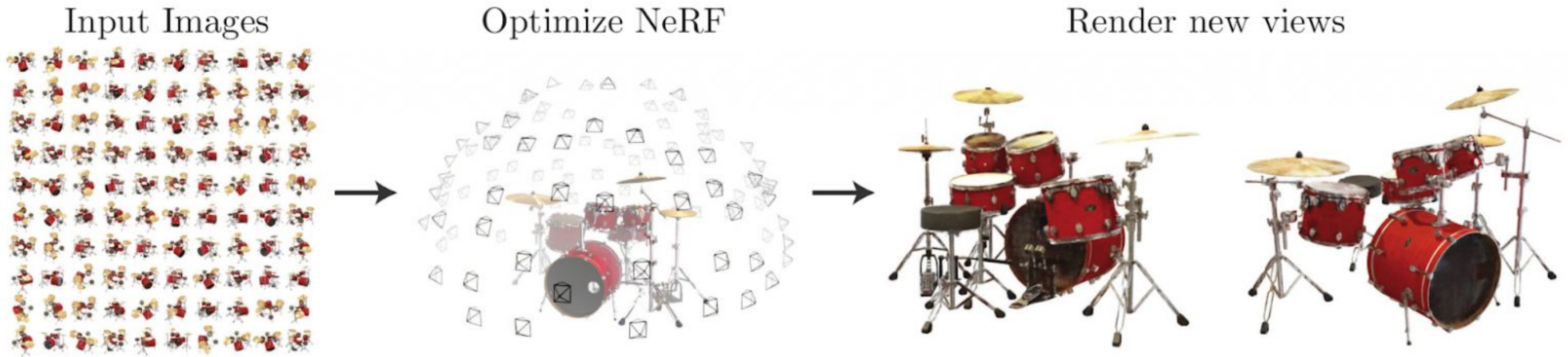
Anastasiy Belyaev, Alexander Kotov

girafe ai

Outline

- Problem statement
- Metrics
- Datasets
- Models
    - NeRF original
    - NeRF in the wild
    - Instant NeRF
    - NeRF++
    - NeRFies
    - Block-NeRF

Problem statement



Synthesize a target image with an arbitrary target camera pose from given source images and their camera poses.

Metrics

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|f(i,j) - g(i,j)\|^2$$

- f - original image mxn
- g - degraded image mxn
- MAX_f - maximum signal value that exists in our original “known to be good” image

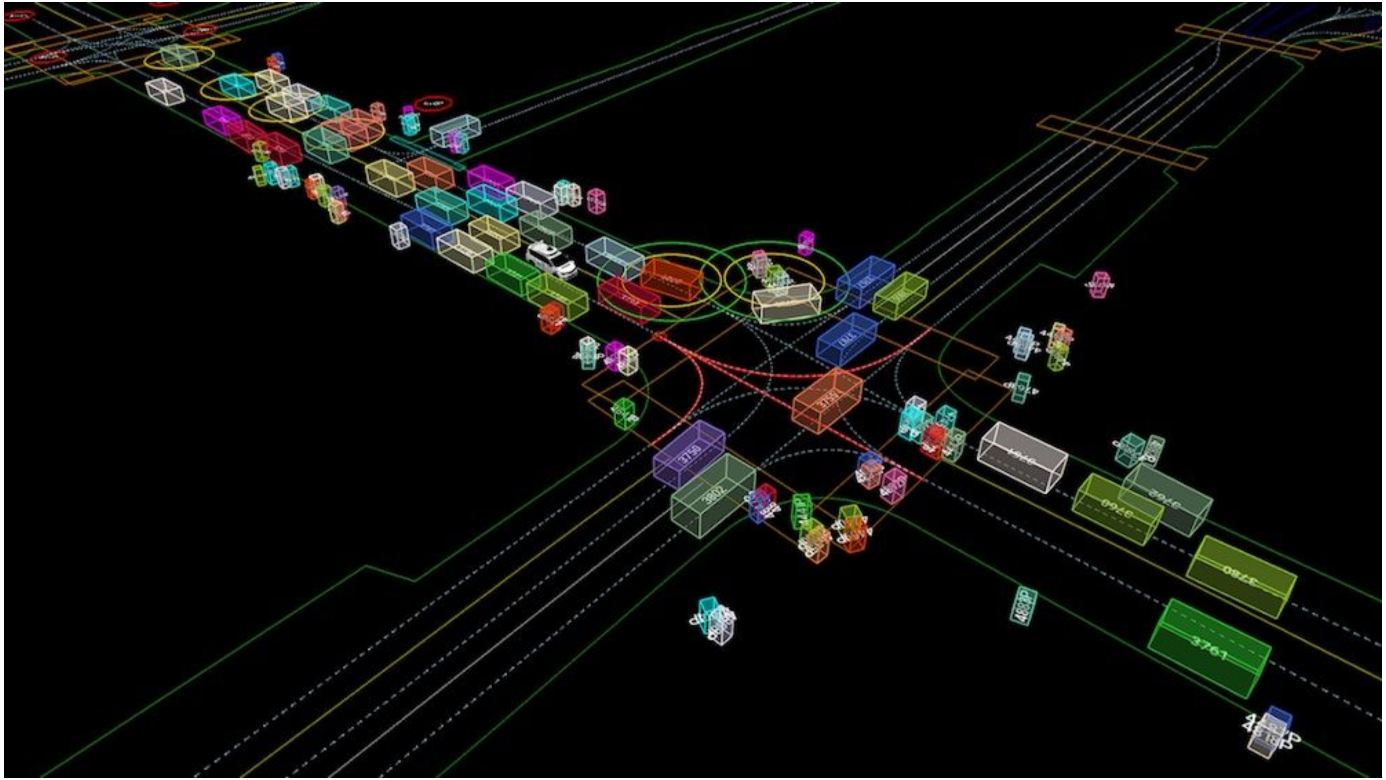
Datasets



- Original Nerf Dataset
- Waymo Dataset



Datasets



- Original Nerf Dataset
- Waymo Dataset

Nerf original

Representing scenes as neural radiance fields for view synthesis

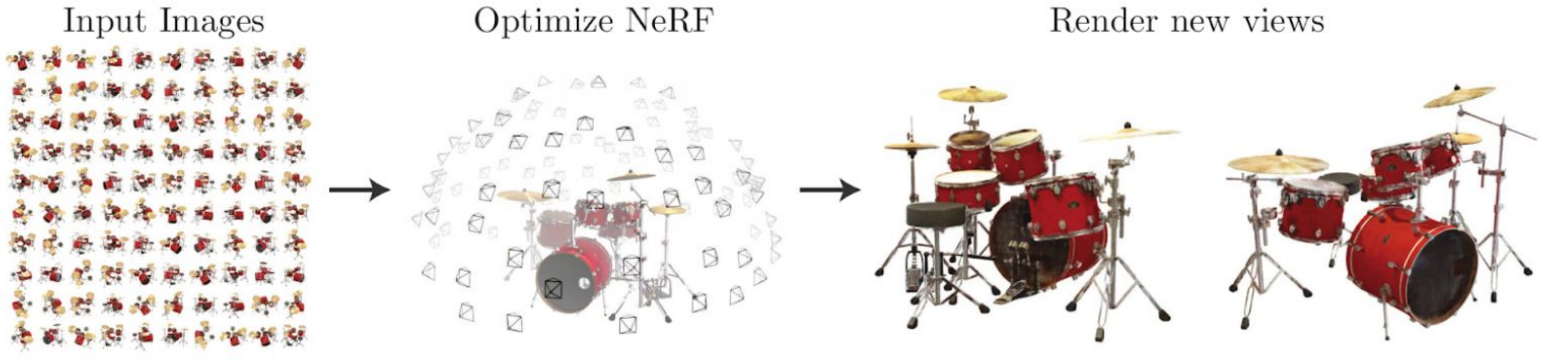
girafe
ai

01

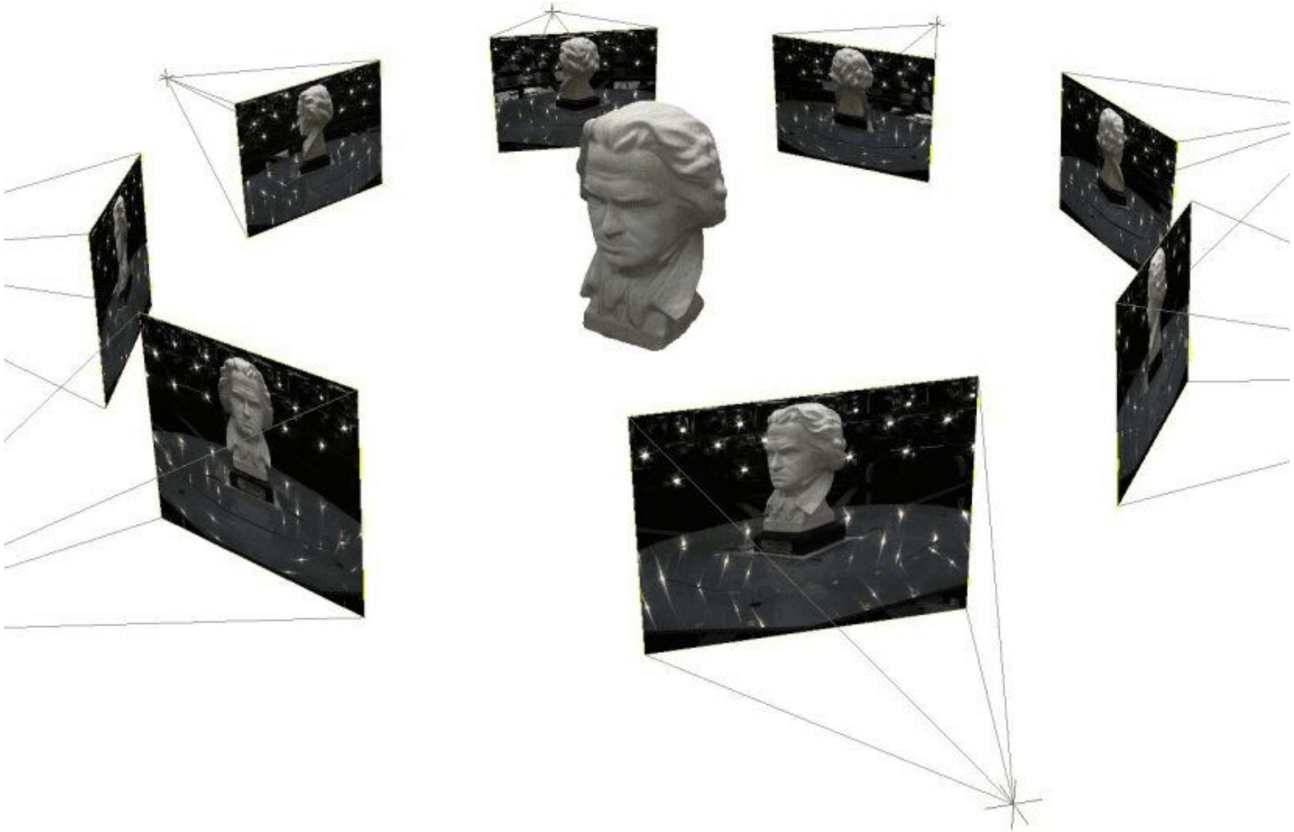
NeRF: Representing scenes as neural radiance fields for view synthesis

- year: 2020
- Conferece: CVPR
- Authros: B Mildenhall, PP Srinivasan, M Tancik
- Citations: 2092
- Link: .pdf
- Code: github

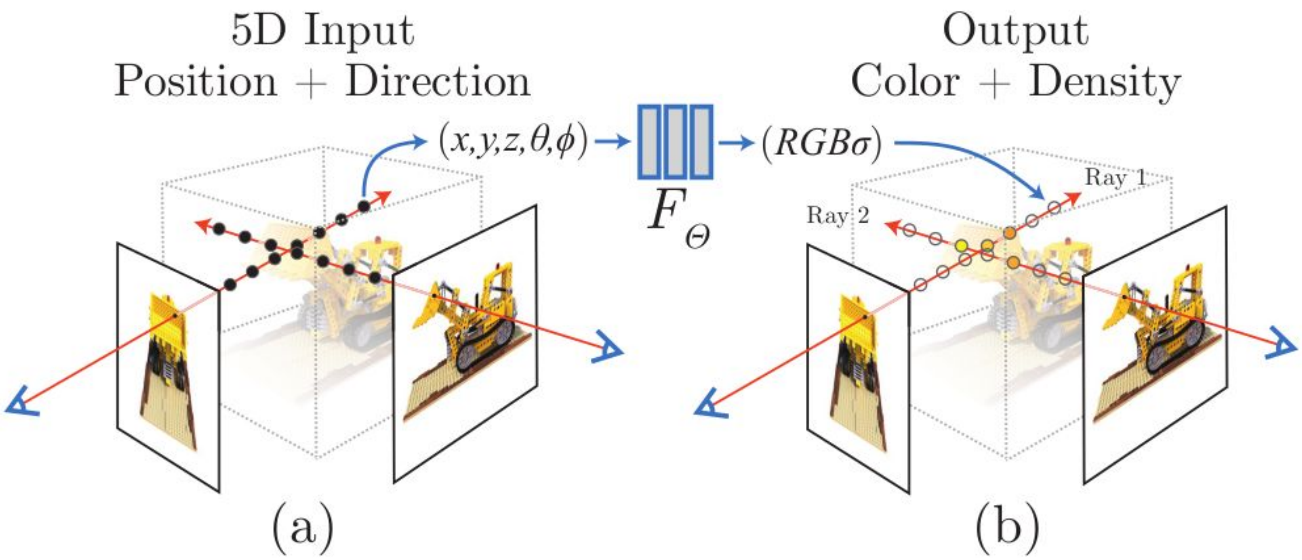
Problem



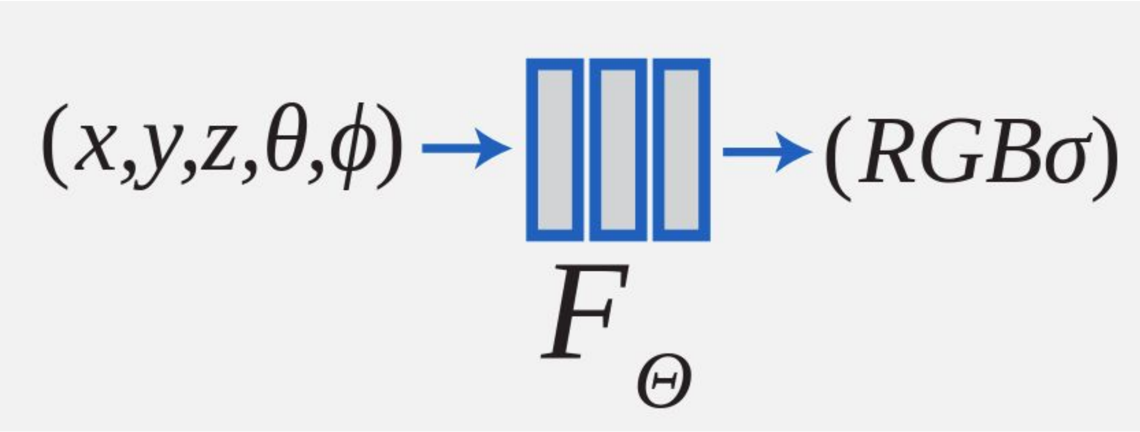
Given a large set of images, NeRF learns to implicitly represent the 3D shape, such that new views can later on be synthesised.

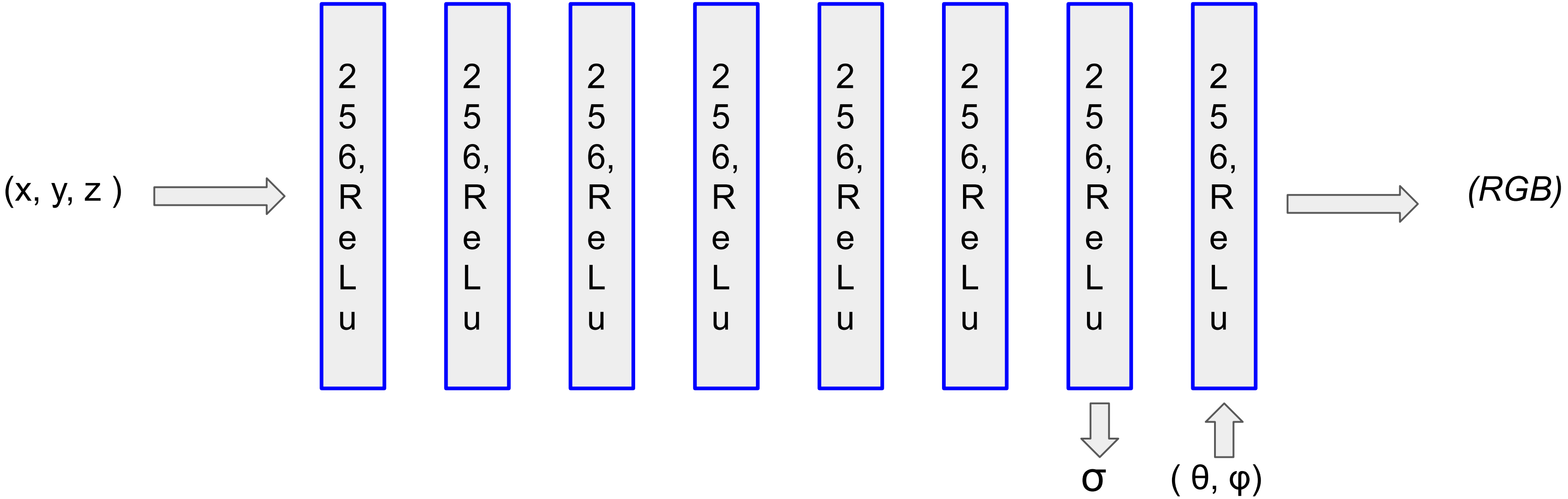


Algorithm



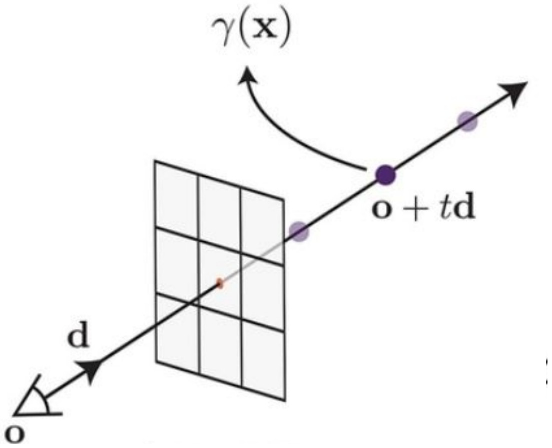
- Generate a sampled set of 3D points
- Produce an output set of densities and colors (using MLP)
- Accumulate densities and color into 2D image


(x,y,z,θ,ϕ)
(RGBσ)
F Θ


(x, y, z)
256,ReLu
256,ReLu
256,ReLu
256,ReLu
256,ReLu
256,ReLu
256,ReLu
256,ReLu
(RGB)
σ
(θ, φ)

(x, y, z, θ, ϕ)

1) march camera rays through the scene to generate a sampled set of 3D points ;
2) use those points and their correspondings 2D viewing directions as input to the neural network to produce an output set of colors and densities ;
3) use classic volume rendering techniques to accumulate those colors and densities into 2D image ;



Volume Rendering

σ(**x**) - differential probability of a ray terminating at an infinitesimal particle at location x ;

C(**r**) - expected color of camera ray **r**(t) = **o + d**t, t ∈ [t_n, t_f] ;

T(t) - probability that the ray travels from t_n to t without hitting any other particle;

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

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

Optimizing a NeRF

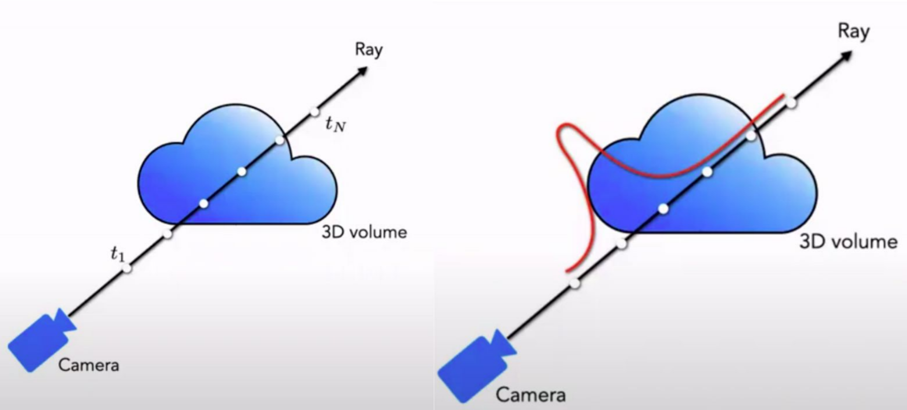
Positional encoding

$$F_\Theta = F'_\Theta \circ \gamma$$

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

Hierarchical volume sampling

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i\,, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i))\,.$$



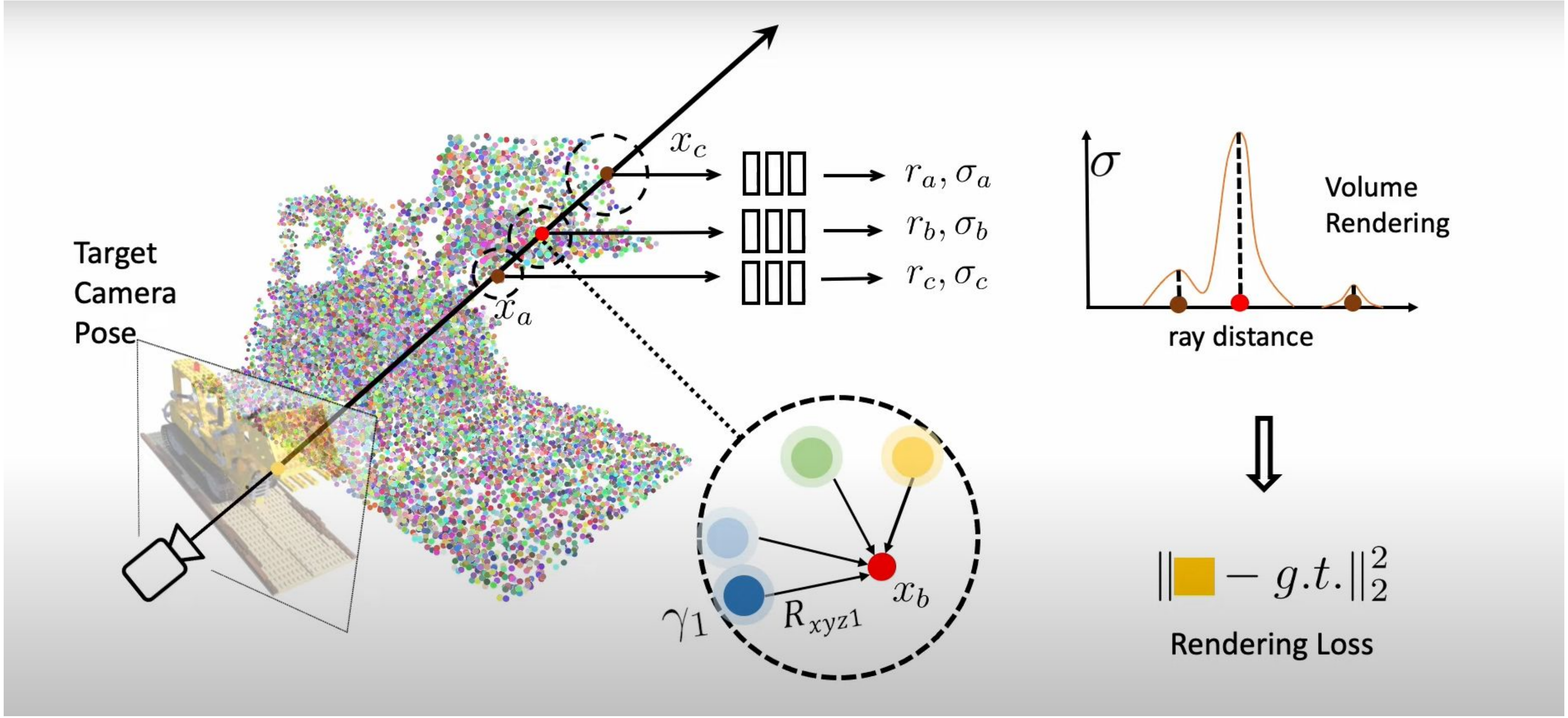
$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$

Quadrature

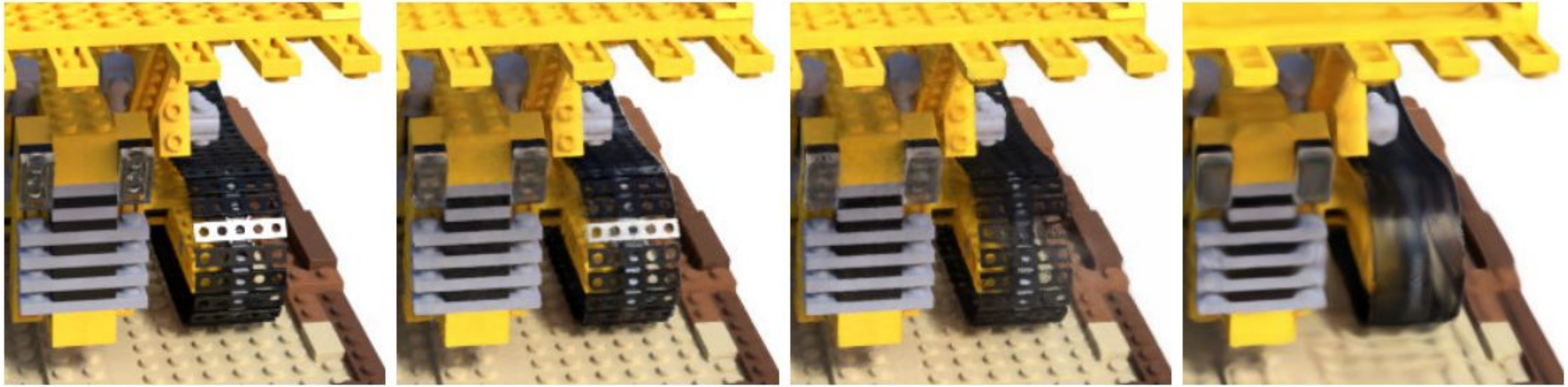
$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right].$$

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

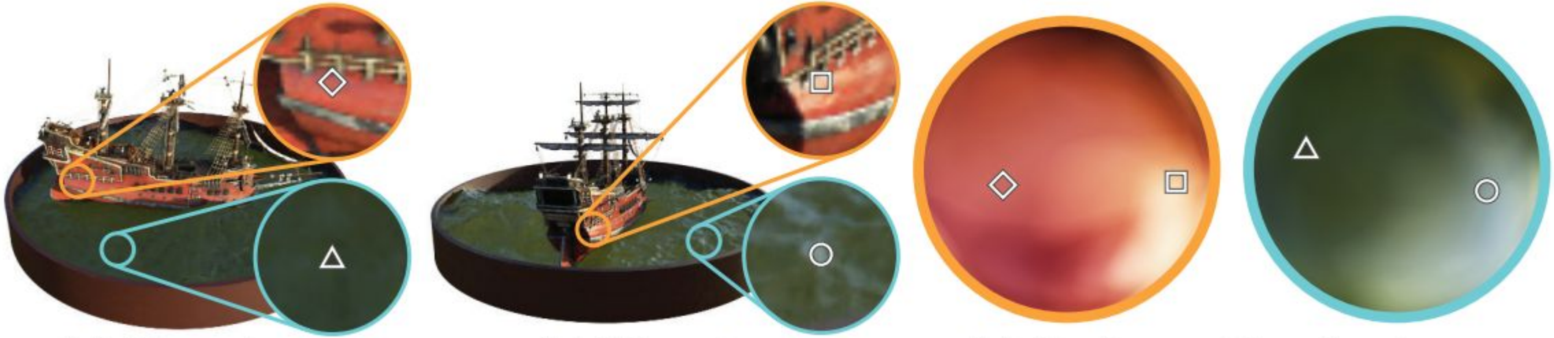
$$\delta_i = t_{i+1} - t_i$$



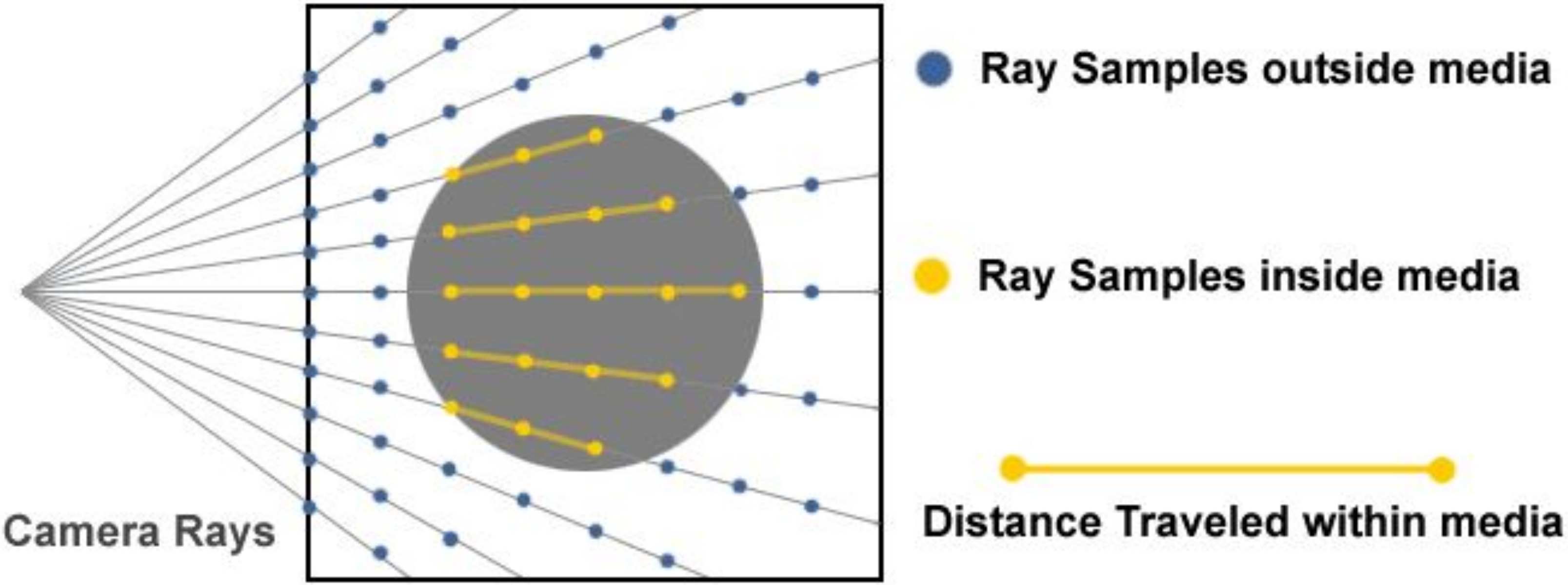
Target
Camera
Pose
x_c
x_a
r_a, σ_a
r_b, σ_b
r_c, σ_c
γ_1
R_{xyz1}
x_b
σ
Volume
Rendering
ray distance
$\|\blacksquare - g.t.\|_2^2$
Rendering Loss



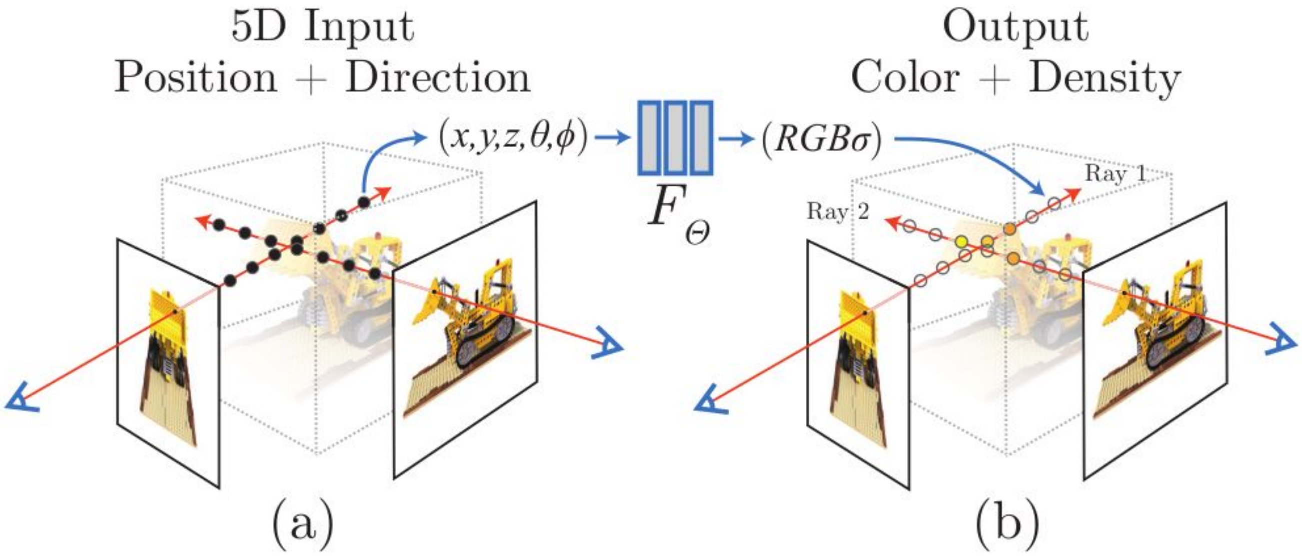
Ground Truth | Complete Model | No View Dependence | No Positional Encoding



(a) View 1 (b) View 2 (c) Radiance Distributions

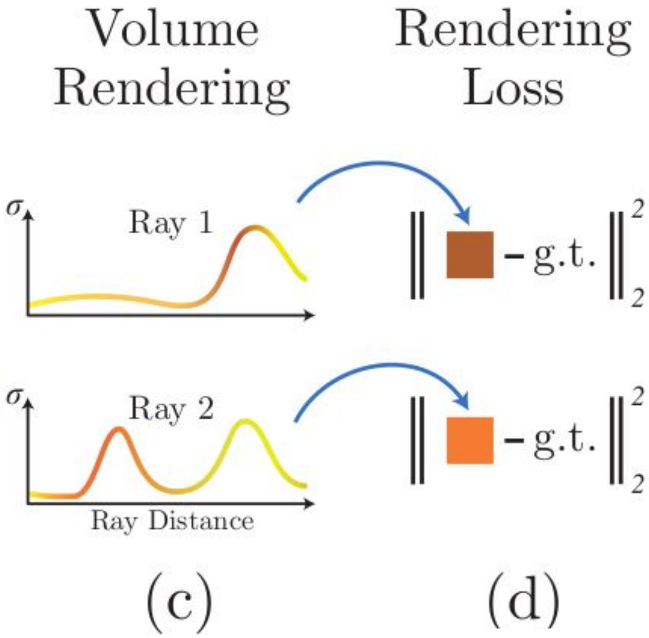

Ray Samples outside media
Ray Samples inside media
Distance Traveled within media
Camera Rays

Architecture



- Architecture is MLP
- Weights of network represent 3D image
- Ray from each pixel of each image

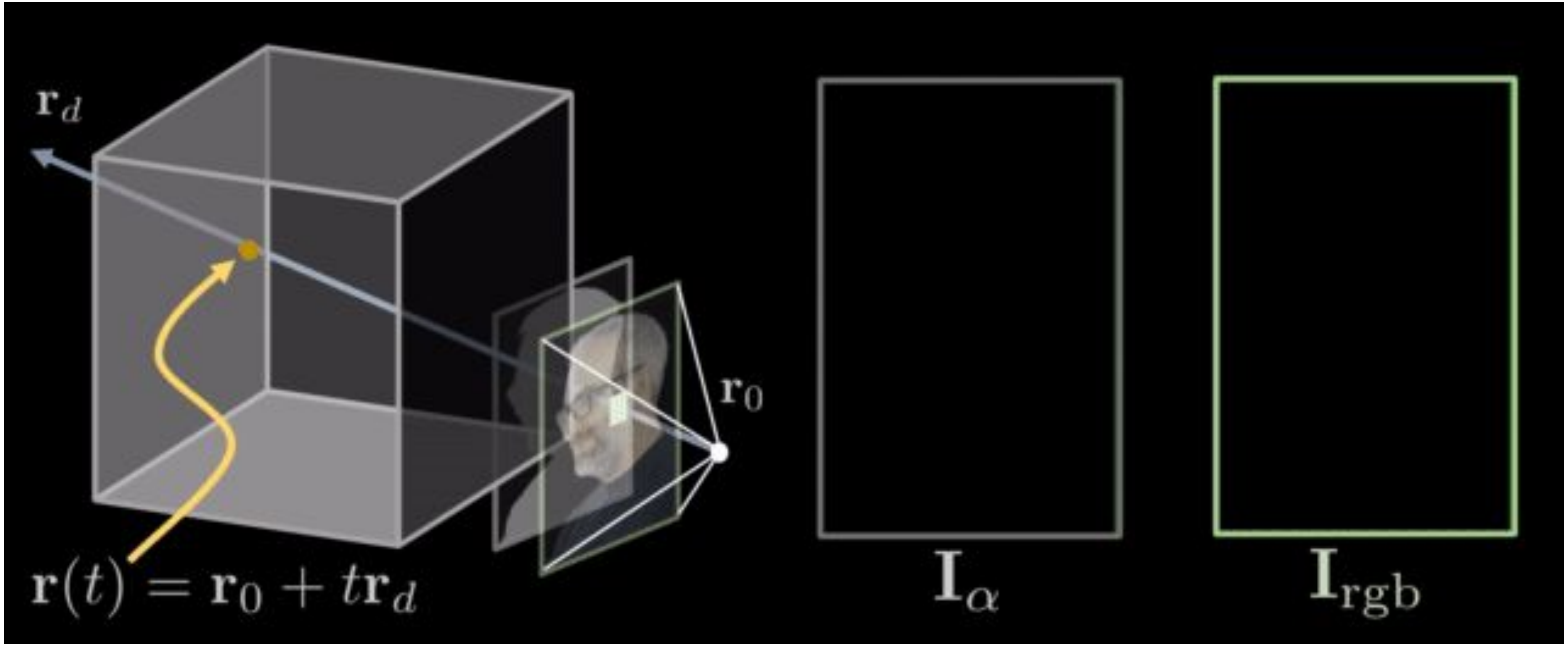
Loss



- Architecture is MLP
- Weights of network represent 3D image
- Ray from each pixel of each

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$

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



$\mathbf{r}_d$
$\mathbf{r}_0$
$\mathbf{r}(t) = \mathbf{r}_0 + t\mathbf{r}_d$
$\mathbf{I}_\alpha$
$\mathbf{I}_{\mathrm{rgb}}$

Nerf in the wild

Neural Radiance Fields for Unconstrained Photo Collections

girafe ai

02

NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections

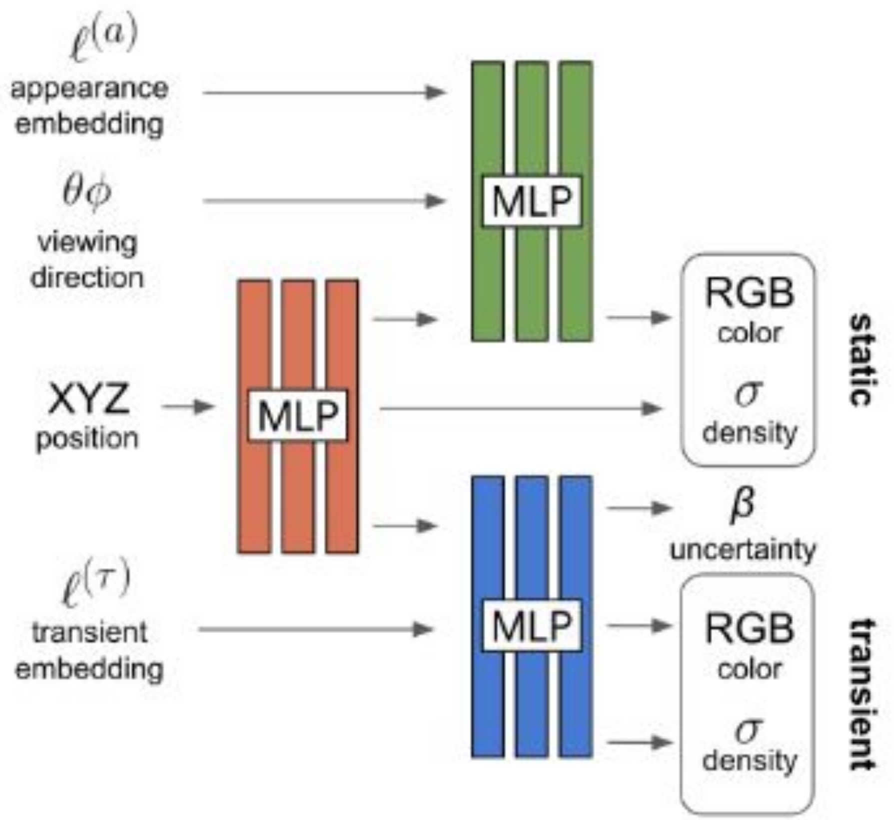
- year: 2021
- Conferece: CVPR
- Authros: R Martin-Brualla, N Radwan
- Citations: 445
- Link: .pdf
- Code: github

Problem

- Occluders
- Uncontrollable external conditions
- Not much data

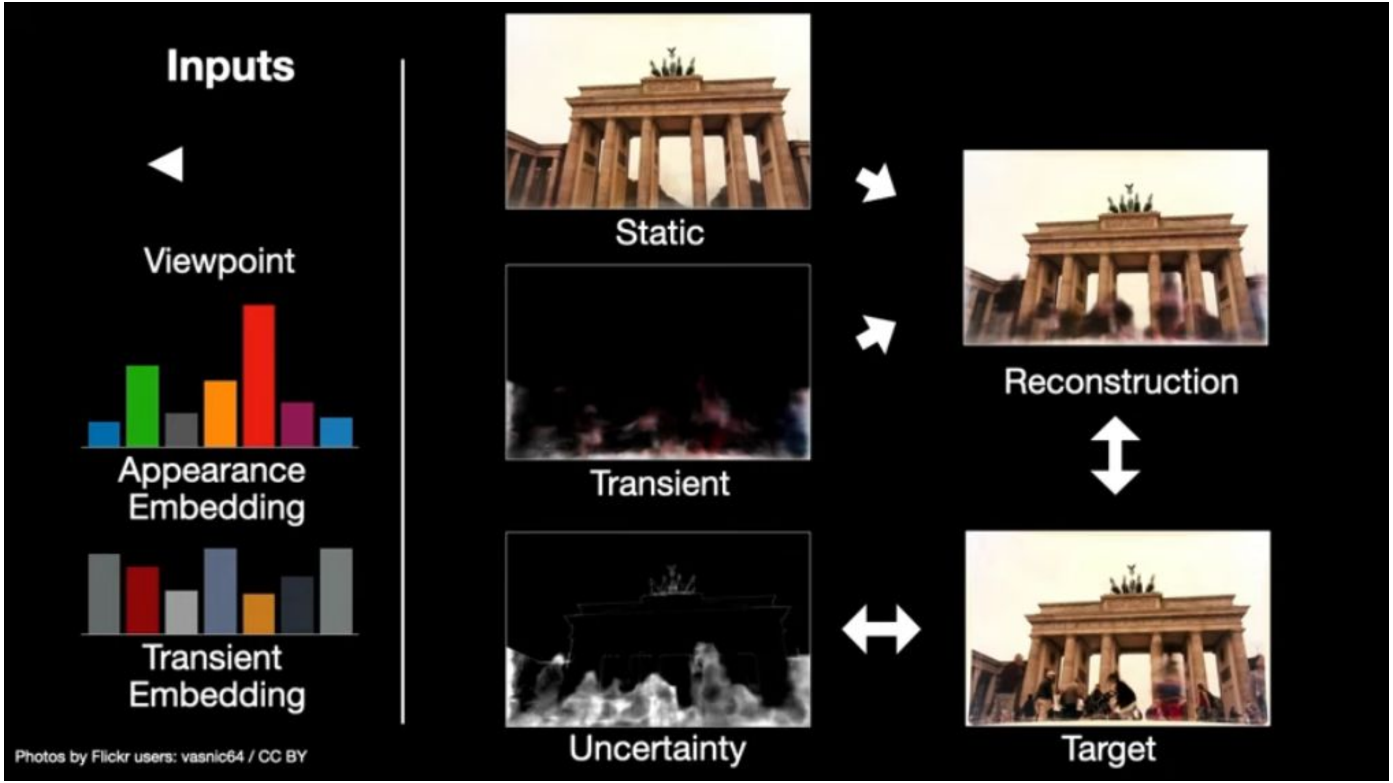
(a) Photos (b) Renderings

Architecture



- Two heads
- Appearance and transient embeddings
- Ray from each pixel of each

Architecture



Instant Nerf

Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

girafe
ai

03

Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

- year: 2022
- Conferece:
- Authros: T Müller, A Evans, C Schied, A Keller
- Citations: 346
- Link: .pdf
- Code: github

Tasks

- Neural gigapixel images
- Neural SDF
- NERF
- Neural volume
- High speed of training

Nerf ++

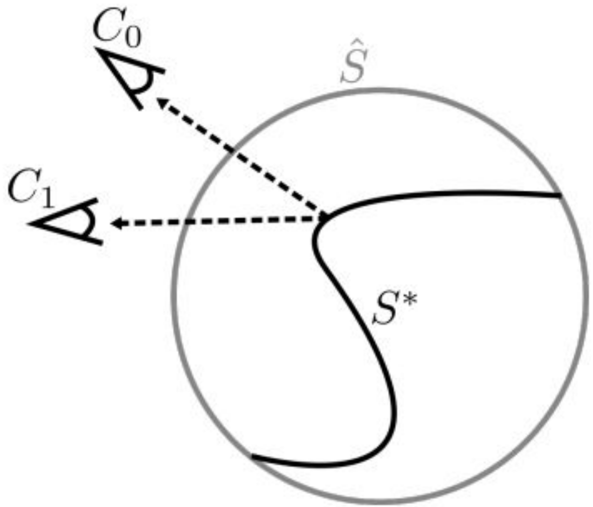
Analyzing and improving neural radiance fields.

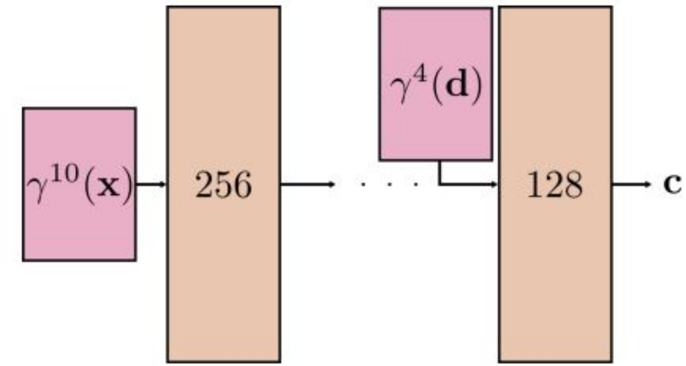
04

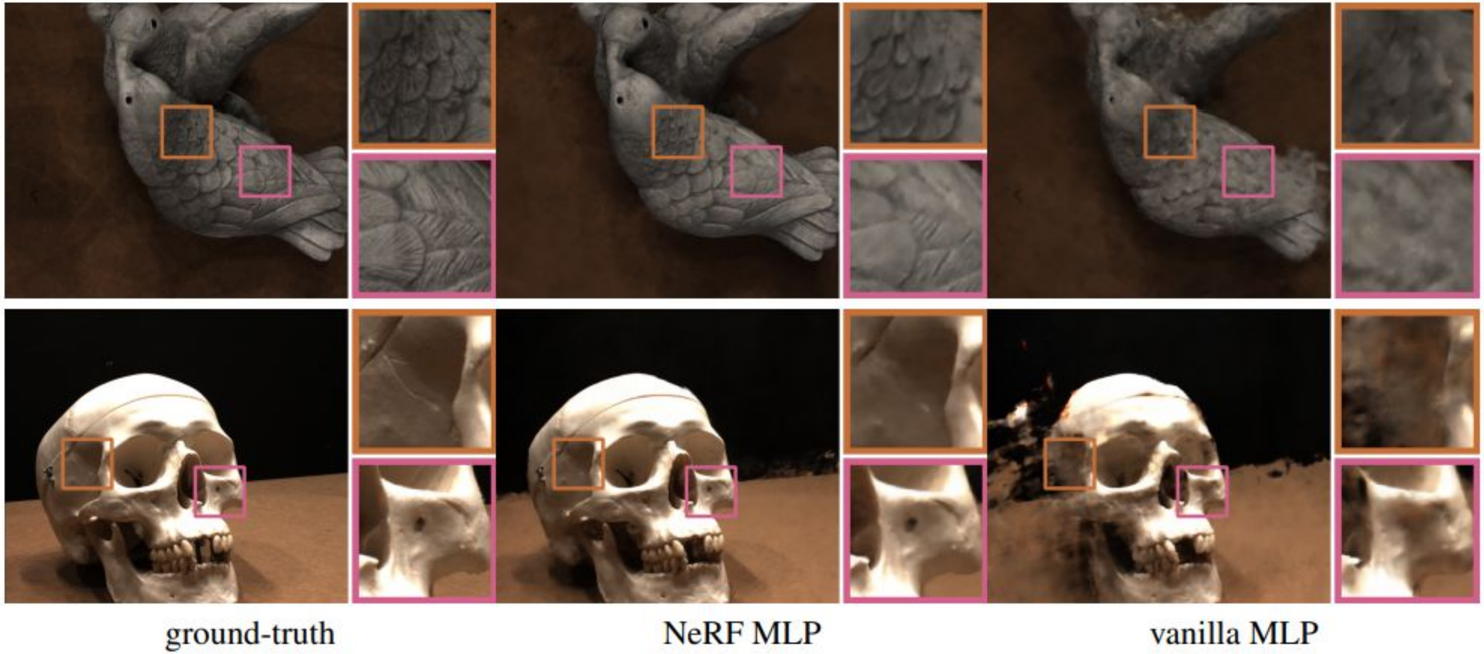
Nerf++: Analyzing and improving neural radiance fields.

- year: 2020
- Conferece:
- Authros: K Zhang, G Riegler, N Snavely, V Koltun
- Citations: 339
- Link: .pdf
- Code: github

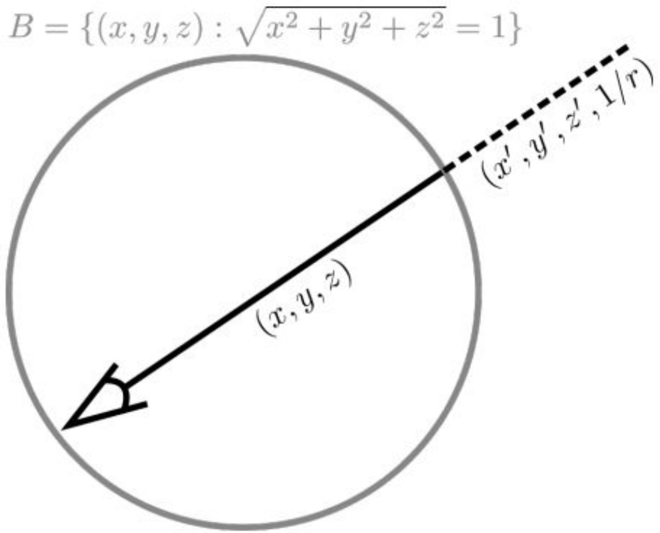
Shape-radiance ambiguity.

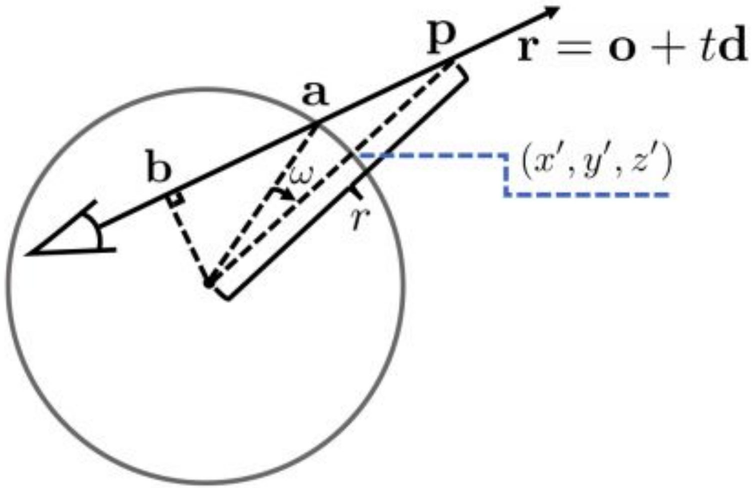





ground-truth
NeRF MLP
vanilla MLP

INVERTED SPHERE PARAMETRIZATION





(a) bounding volume for the truck only (b) bounding volume for the entire scene

(a) NeRF++ prediction (b) predicted foreground (c) predicted background

NeRF vs NeRF++

NeRF NeRF++

NeRFies

Deformable neural radiance fields.

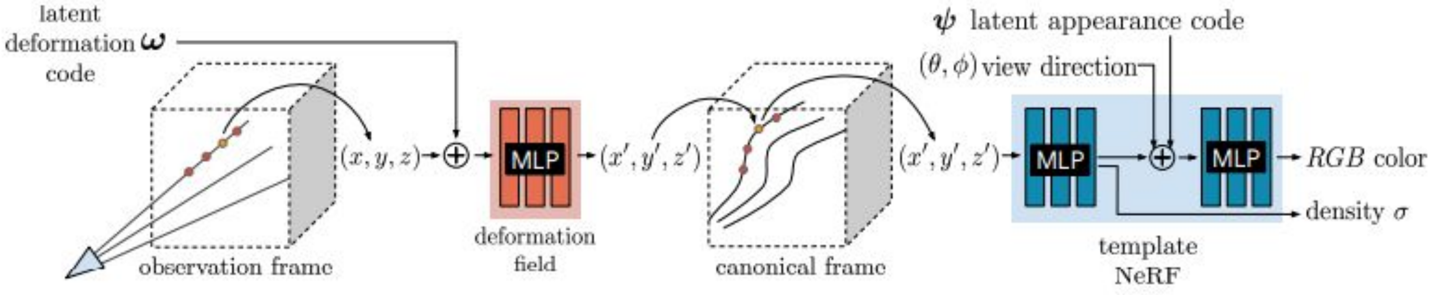
girafe
ai

05

Nerfies: Deformable neural radiance fields.

- year: 2021
- Conferece: ICCV2021
- Authros: K Park, U Sinha, JT Barron
- Citations: 467
- Link: .pdf
- Code: github

Nerfies



The method is able to turn arbitrarily taken selfie photos or videos into deformable Nerf models, which allow you to recreate photorealistic images of an object at any point.

Neural Deformation Field

$$G(\mathbf{x}, \mathbf{d}, \boldsymbol{\psi}_i, \boldsymbol{\omega}_i) = F\left(T(\mathbf{x}, \boldsymbol{\omega}_i), \mathbf{d}, \boldsymbol{\psi}_i\right) .$$

Elastic Regularization

$$L_{\text{elastic}}(\mathbf{x}) = \|\log \boldsymbol{\Sigma} - \log \mathbf{I}\|_F^2 = \|\log \boldsymbol{\Sigma}\|_F^2 ,$$

Background Regularization

$$L_{\text{bg}} = \frac{1}{K}\sum_{k=1}^{K}\left\|T(\mathbf{x}_k) - \mathbf{x}_k\right\|_2 .$$

Coarse-to-Fine Deformation Regularization

$$w_j(\alpha) = \frac{(1 - \cos(\pi \operatorname{clamp}(\alpha - j, 0, 1))}{2} ,$$

Block-nerf

Scalable Large Scene Neural View Synthesis

girafe ai

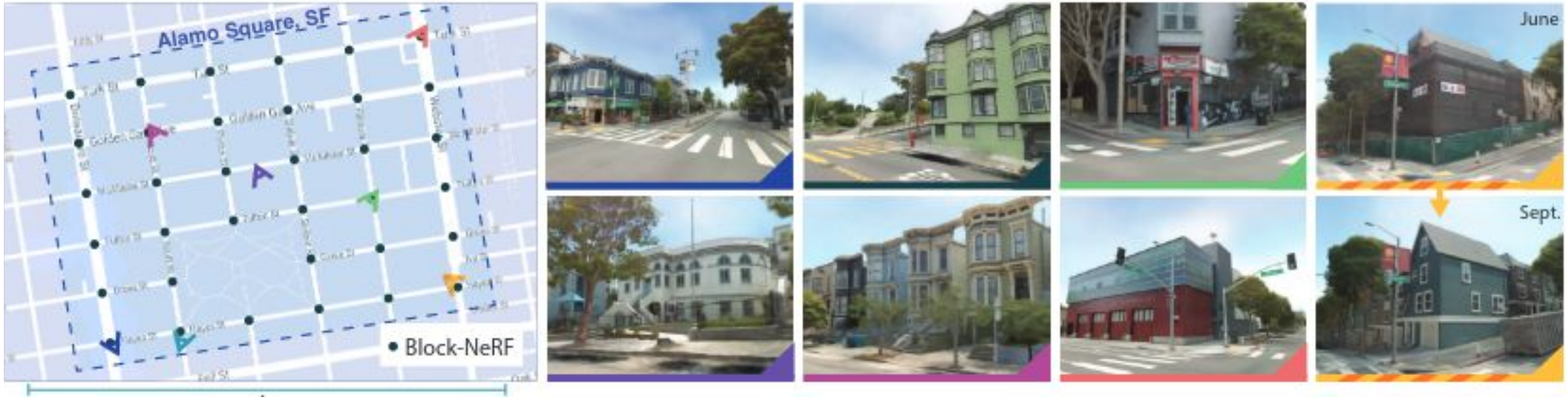
06

Block-NeRF: Scalable Large Scene Neural View Synthesis

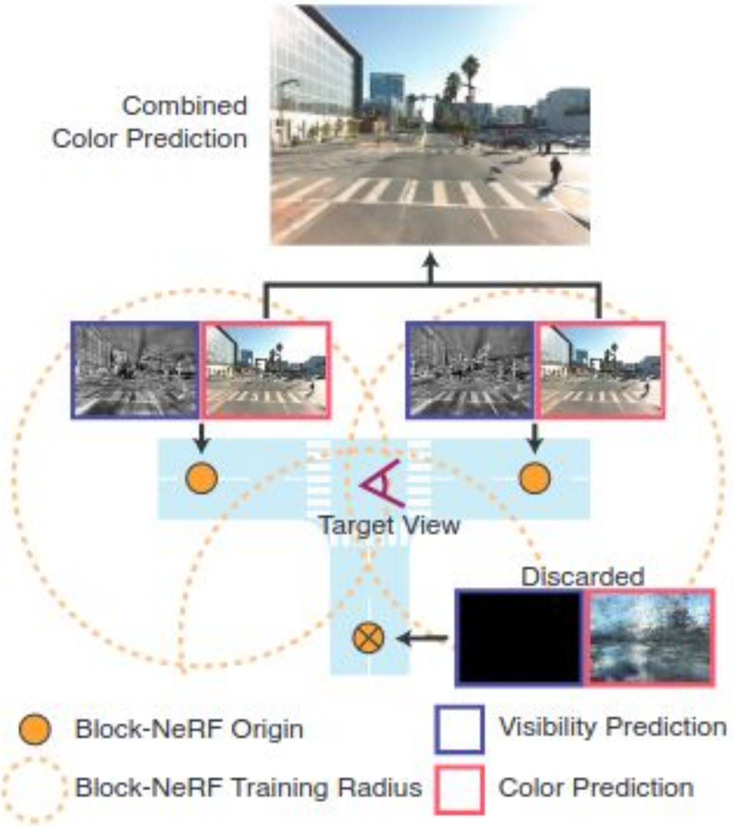
- year: 2022
- Conferece: CVPR
- Authros: M Tancik, V Casser, X Yan, S Pradhan
- Citations: 86
- Link: .pdf
- Code: github

Problem

- Want to represent large scenes
- Want to expand pretrained Nerf with new part
- Want good speed of training



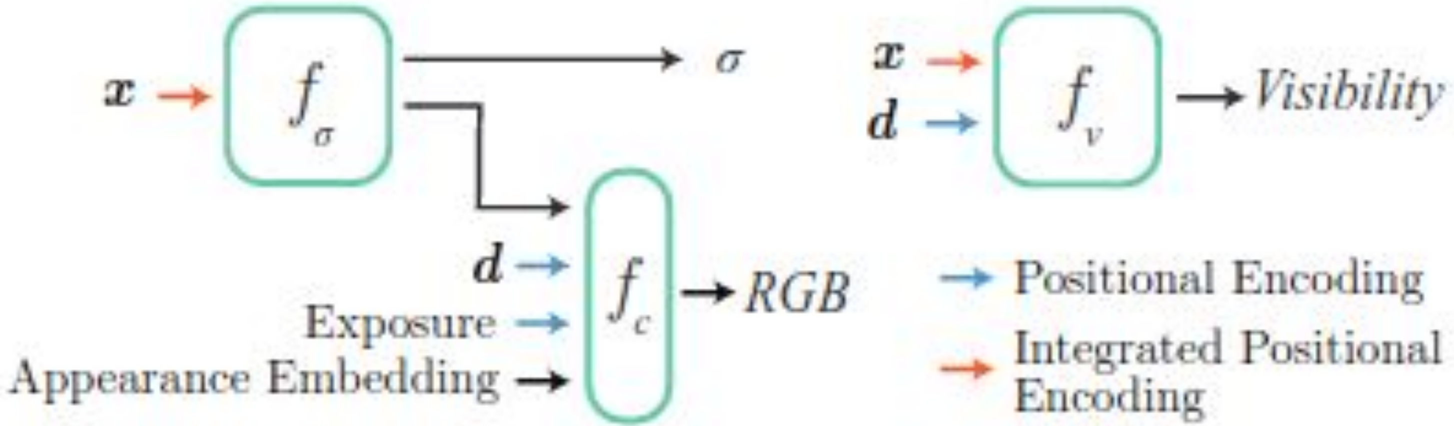
Algorithm



- Multiple nerfs with own sectors
- Target view generated by combining nerfs with good visibility
- Merging renderings based on block origin's distance

Architecture

- Appearance embeddings from nerf in the wild
- Separate MLP for visibility prediction



Thanks for attention!

Questions?

girafe ai