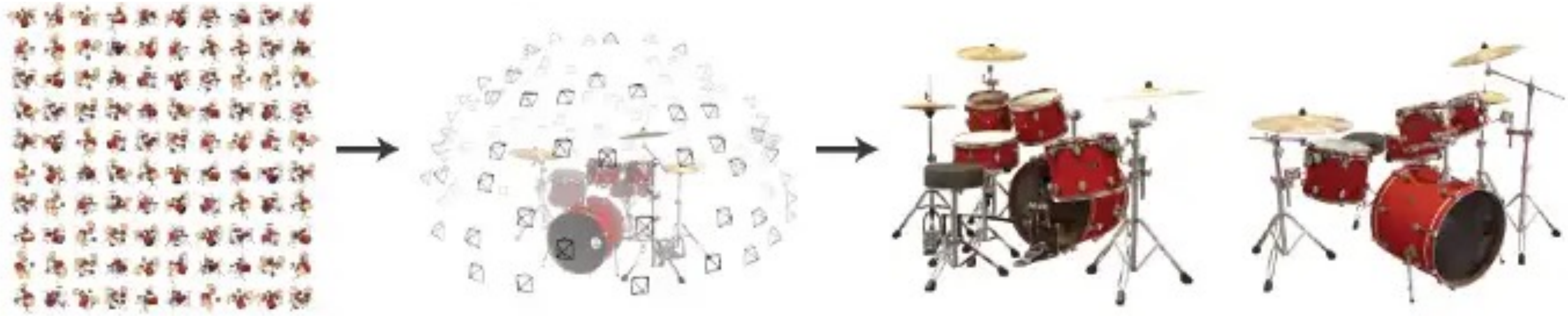
Virtual Humans – Winter 23/24

Lecture 10_1 – Humans and NeRF

Prof. Dr.-Ing. Gerard Pons-Moll
University of Tübingen / MPI-Informatics

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Goal: Novel View Synthesis



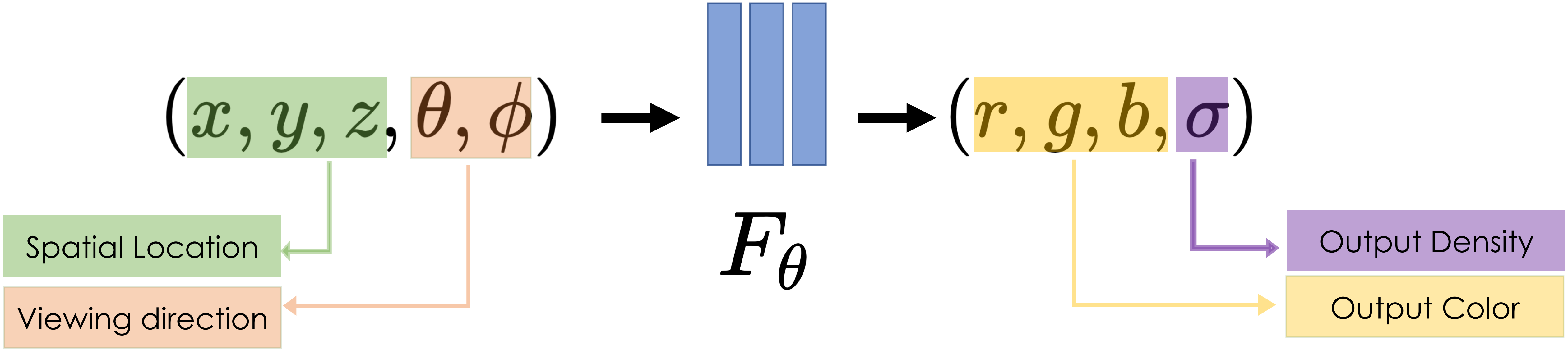
Input:
Sparsley sampled
images of the scene

Learn scene
representation

Novel view synthesis
of the scene

Novel View Synthesis

Learning radiance field representation of scene:



Volume Rendering

Given color and density (r, g, b, σ), we calculate the color of every camera ray using:

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

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

Near and far bounds

Color (r,g,b) at r(t)

Volume density: Probability of a ray terminating at an infinitesimal particle at location r(t)

Accumulated transmittance along ray :

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

Volume Rendering in NeRF

Given color and density (r, g, b, σ) , we calculate the color of every camera ray using:

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

Color (r,g,b) at t_i

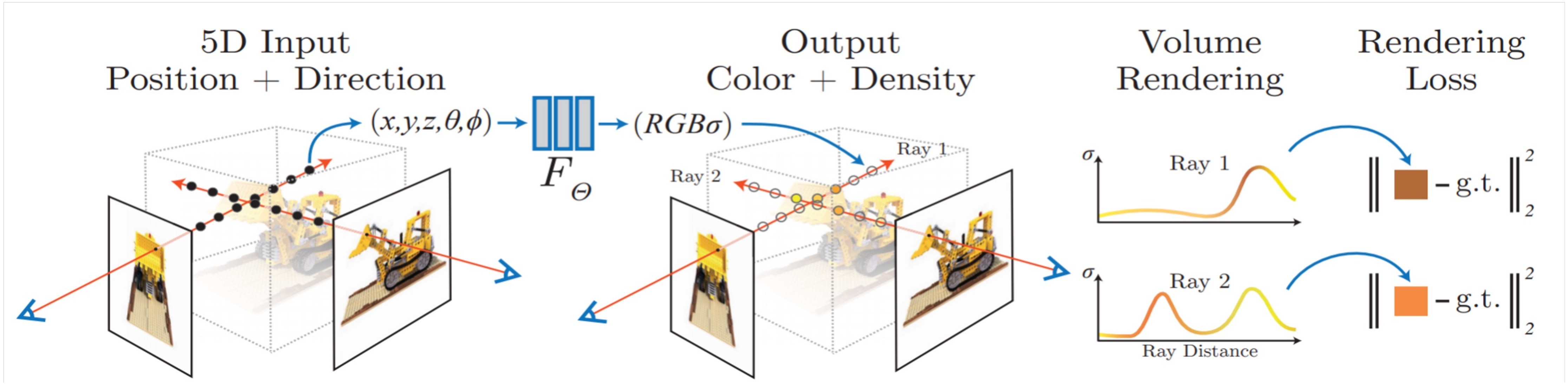
Uniform N samples
$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right]$

Distance between adjacent samples
$\delta_i = t_{i+1} - t_i$

Accumulated transmittance along ray
$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$

Alpha(in traditional alpha composting)
$\alpha_i = 1 - \exp(-\sigma_i \delta_i))$

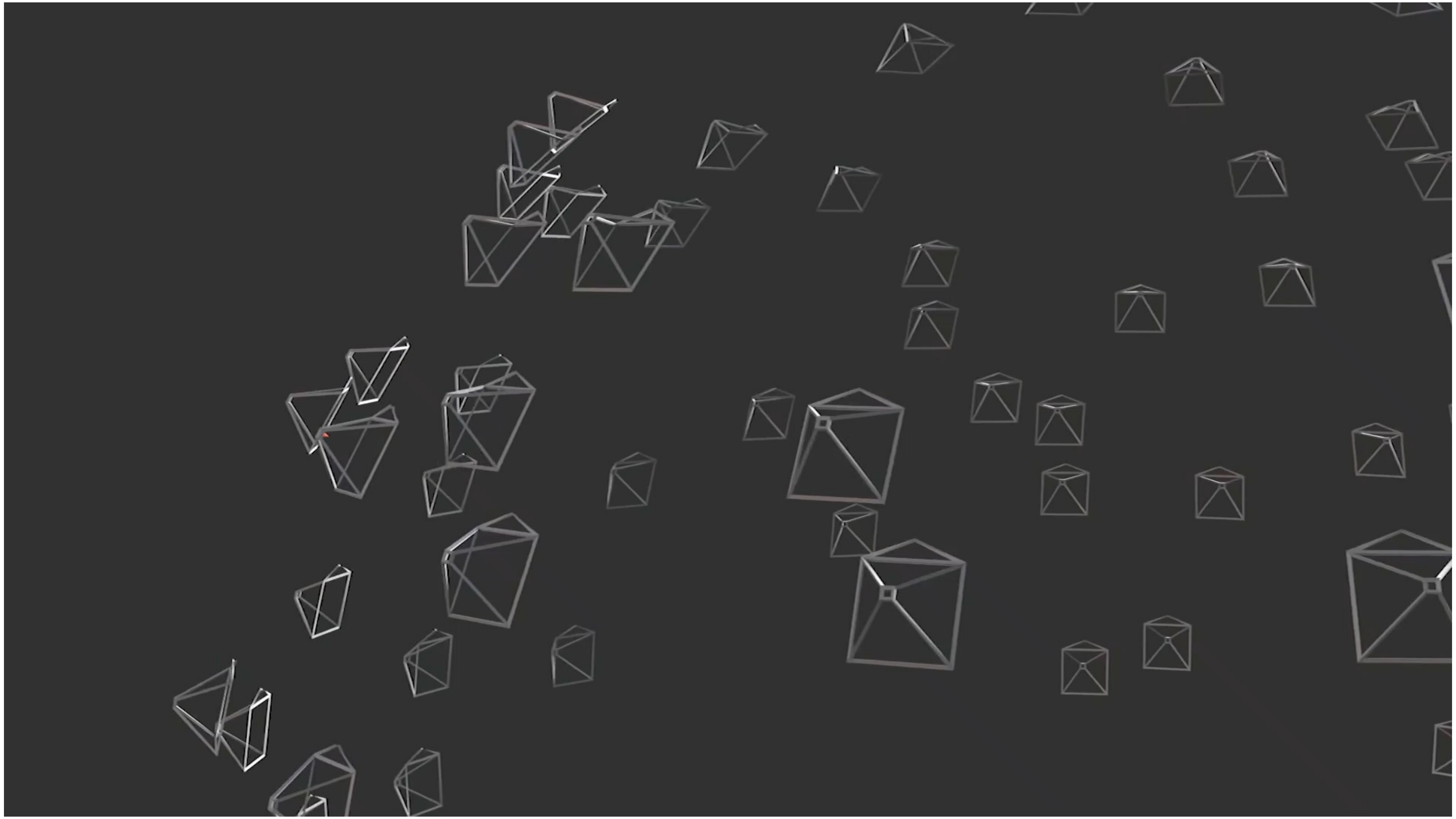
Neural Radiance Fields



<u>**Training NeRF**</u>:

1) **March camera rays** through the scene to generate a **sampled set of 3D points**
2) Use those **points and** their corresponding 2D **viewing directions** as input to the neural network to produce an output **set of colors and densities**
3) Use classical volume rendering techniques to **accumulate those colors and densities** into a 2D image
4) **Minimize** error between **rendered color and GT color**

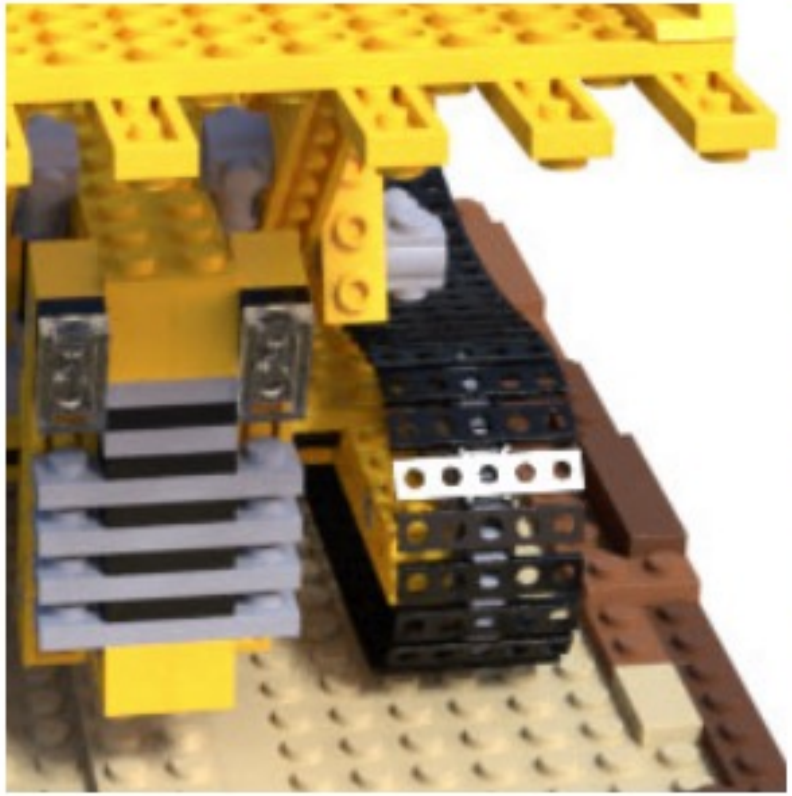
Neural Radiance Fields



Tannick et al. ECCV 20

Novel view synthesis using NeRF

Generated results are blurry

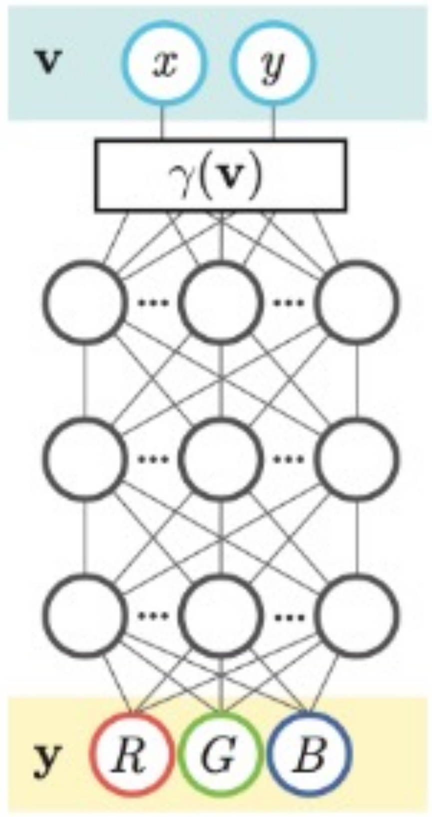


Ground Truth

No Positional Encoding

Why blurry results?

Can coordinate based MLP learn high-frequency details?

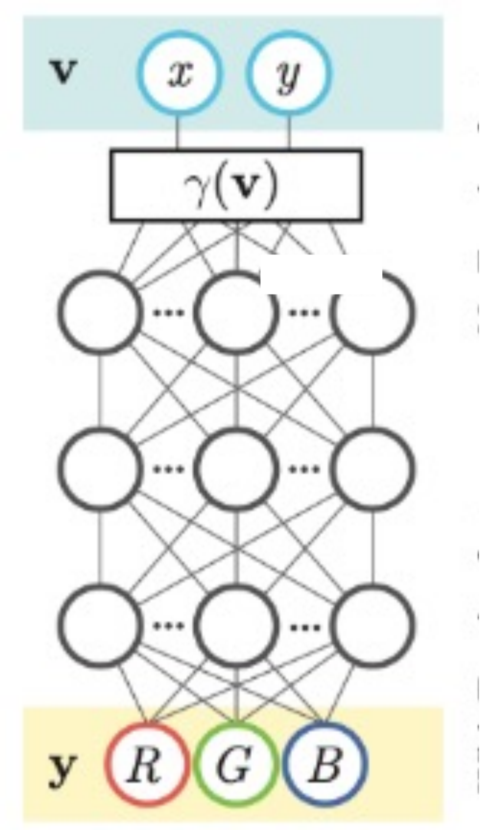


(a) Coordinate-based MLP

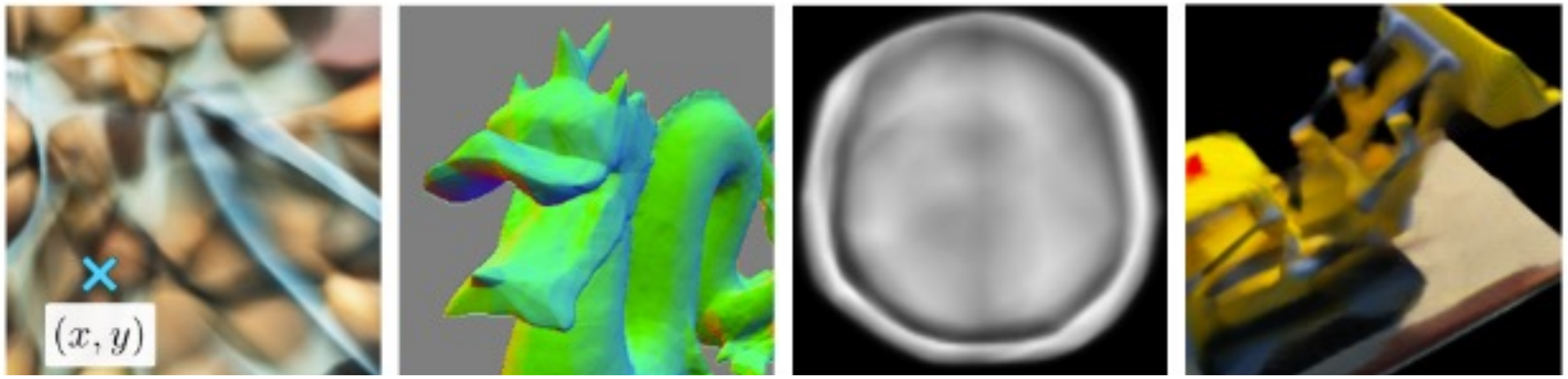
Tannick et al. NeurIPS 20

Why blurry results?

Coordinate based neural network fail to learn high frequency details for all kind of data including RGB image, 3D shape, density , etc



(a) Coordinate-based MLP



RGB | 3D Shape | Density | Radiance field (density, color)

Tannick et al. NeurIPS 20

Solution:

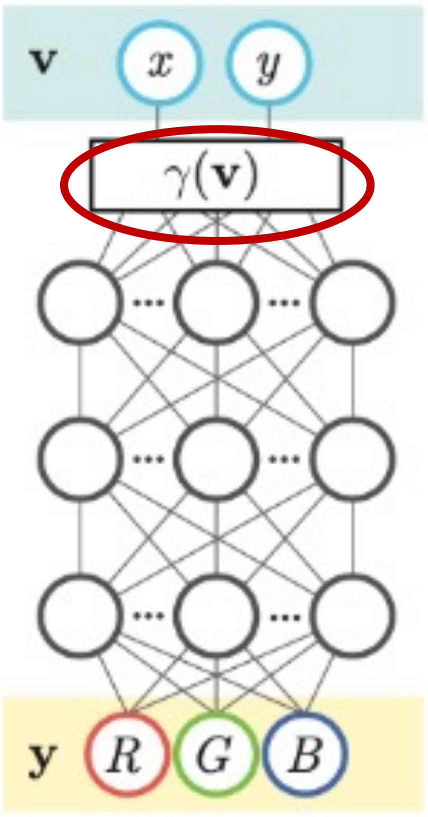
- In naive setting, the bandwidth of the Neural Tangent Kernel limits the spectrum of the recovered/learned function.
- Using a Fourier feature mapping transforms the neural kernel into a stationary kernel in our low-dimensional problem domains and increase the spectrum.

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

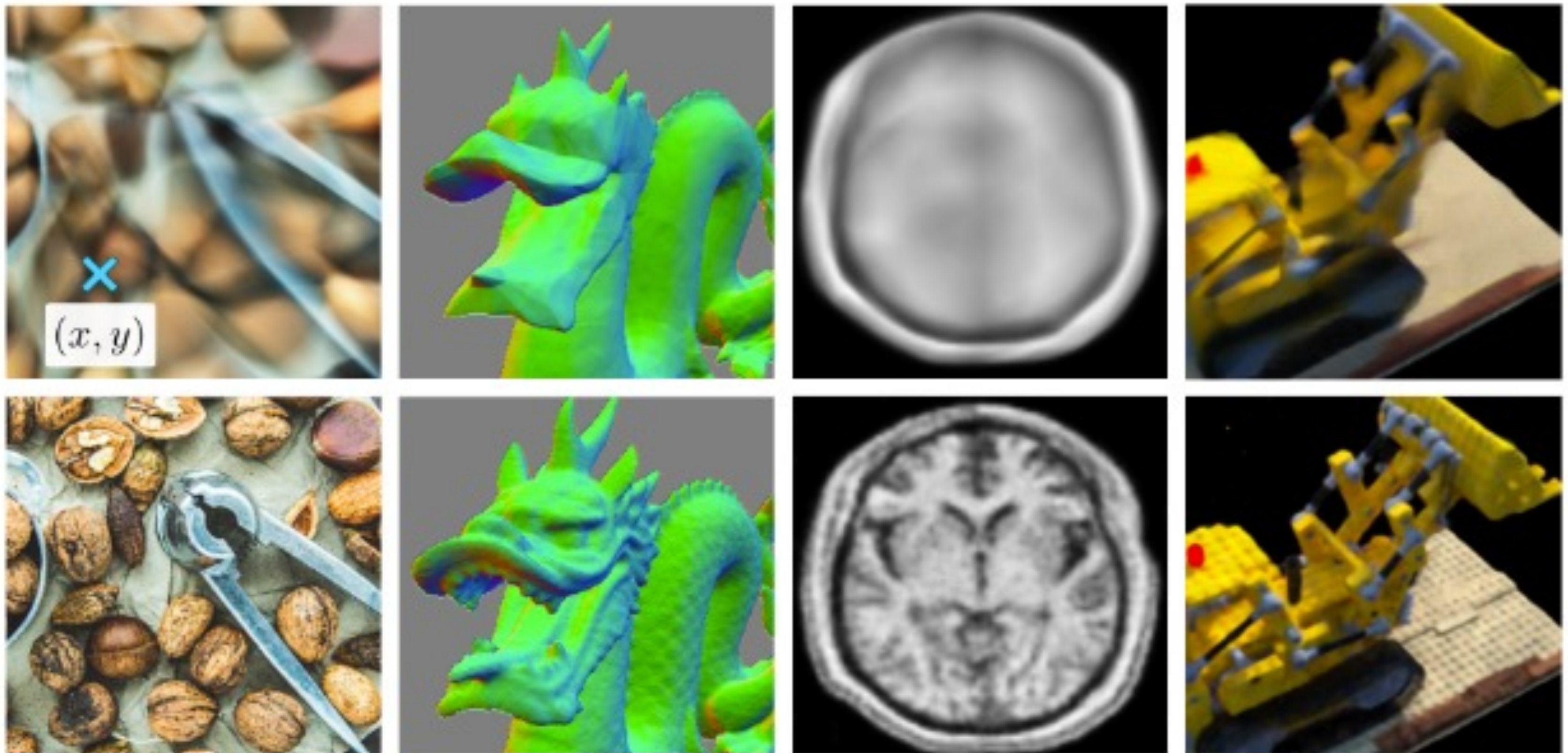
Coordinate input e.g. pixel location for images, 3D point for NeRF

Tannick et al. NeurIPS 20

Fourier Features in Coordinate MLPs



(a) Coordinate-based MLP



RGB

3D Shape

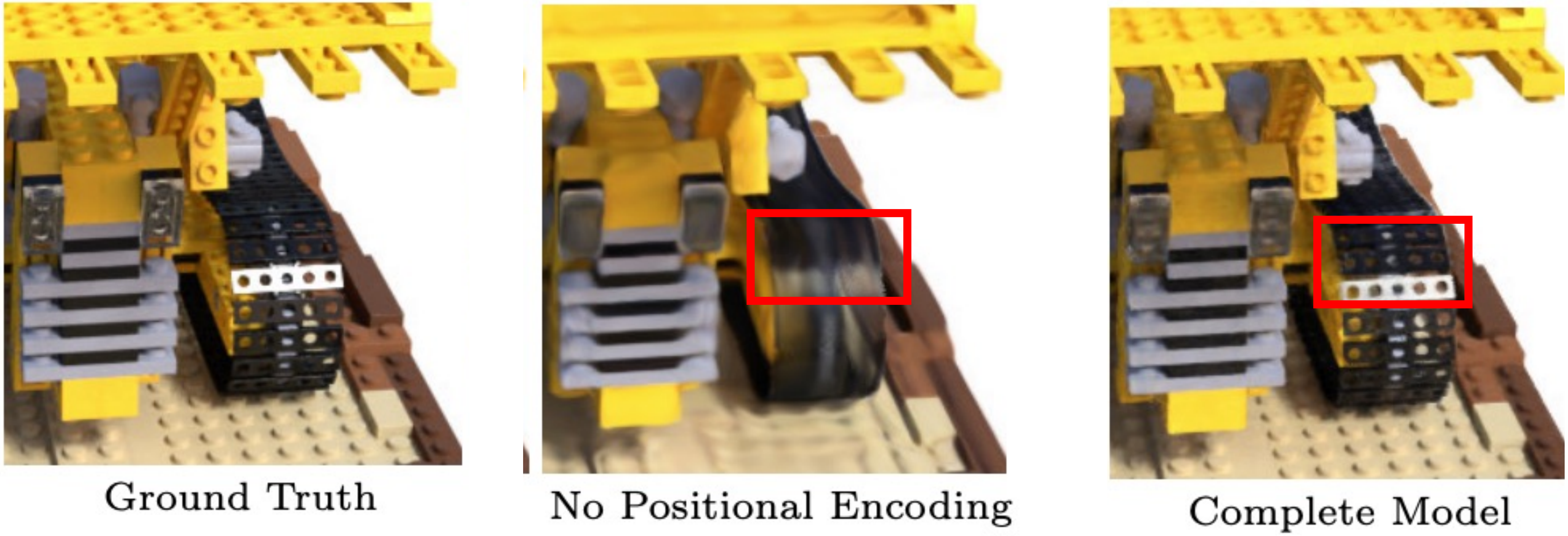
Density

Radiance field (density, color)

Tannick et al. NeurIPS 20

Positional Encoding in NeRF

With fourier features/positional encoding, NeRF learns high frequency details.



Tannick et al. NeurIPS 20

Positional Encoding in NeRF

With fourier features/positional encoding, NeRF learns high frequency details.

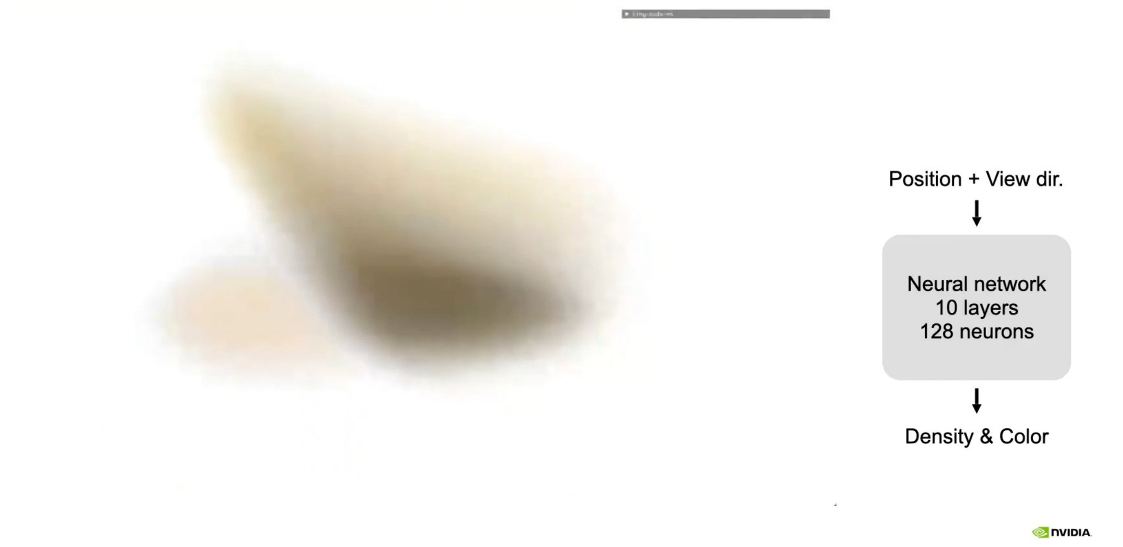
Without positional encoding

Mueller et al. SIGGRAPH 22

Positional Encoding in NeRF

With fourier features/positional encoding, NeRF learns high frequency details.

Without positional encoding

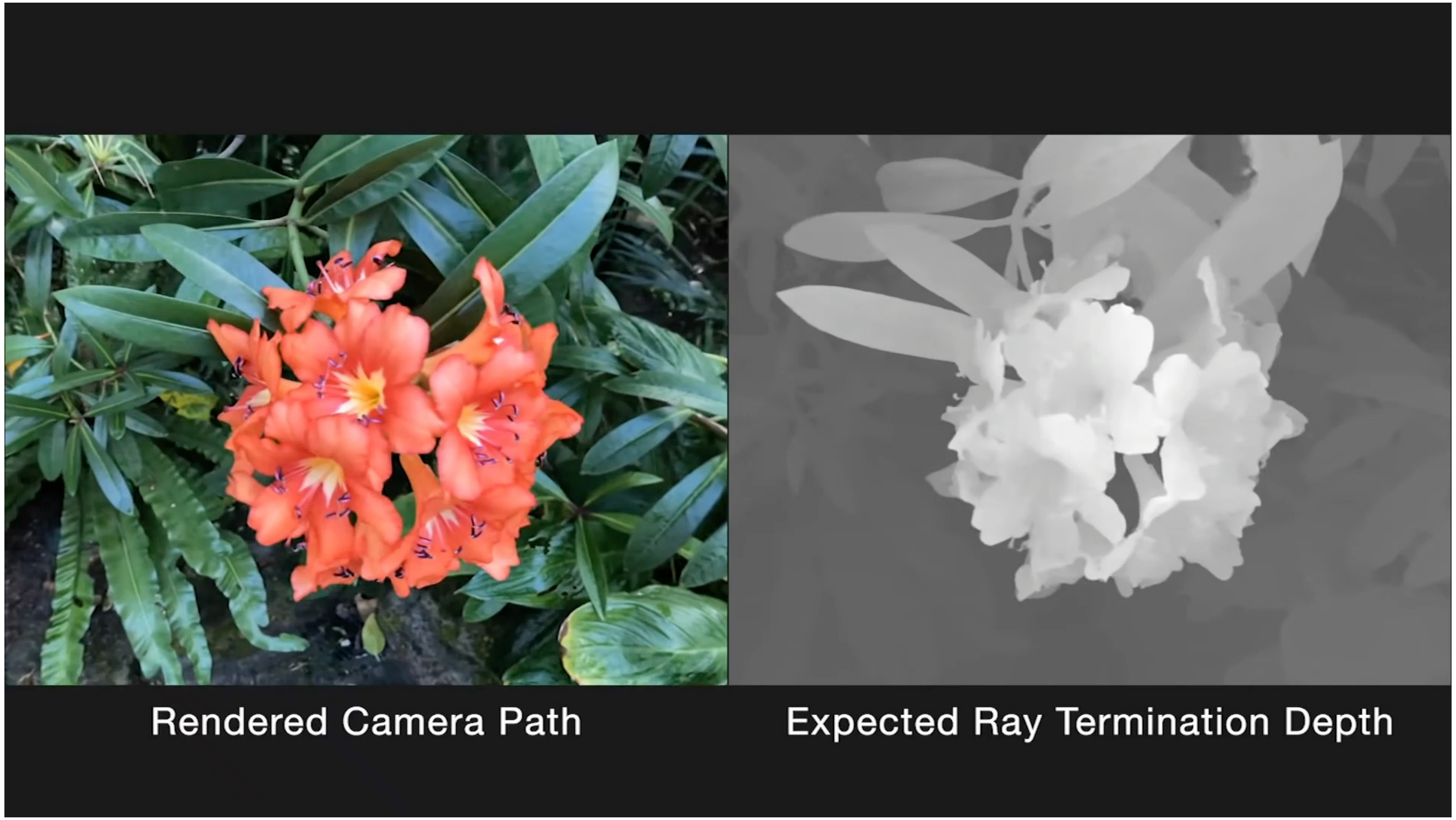


With positional encoding

Mueller et al. SIGGRAPH 22

Geometry in NeRF

Scene geometry can be approximated using threshold



Tannick et al. ECCV 20

Limitations of NeRF

1. Scene specific and only static scene can be modeled.

GT image of a dynamic scene

Image generated with NeRF

Limitations of NeRF

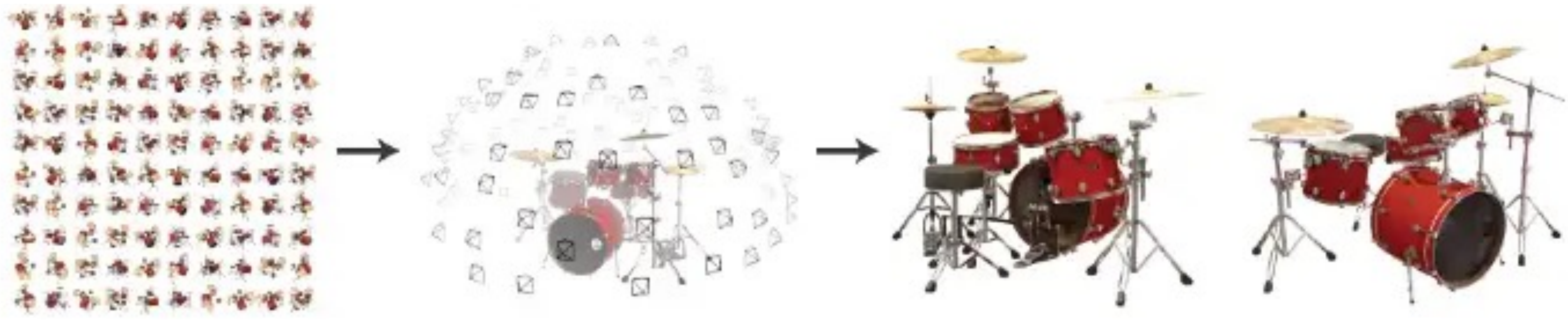
2. No editing and control

- Learned scene cannot be modified.
- Scene is memorized within the network

Limitations of NeRF

3. Generalization

- Scene specific models.
- Large number of images are needed



Limitations of NeRF

4. Expensive training:

- Training is slow(10 hours-up to few days)
- Inference is also not real time

Limitations of NeRF

5. Surface extracted is not accurate and depends on threshold.

GT image NeRF (density threshold $\sigma = 50$)

$\sigma = 1$ $\sigma = 10$ $\sigma = 50$ $\sigma = 100$ $\sigma = 500$

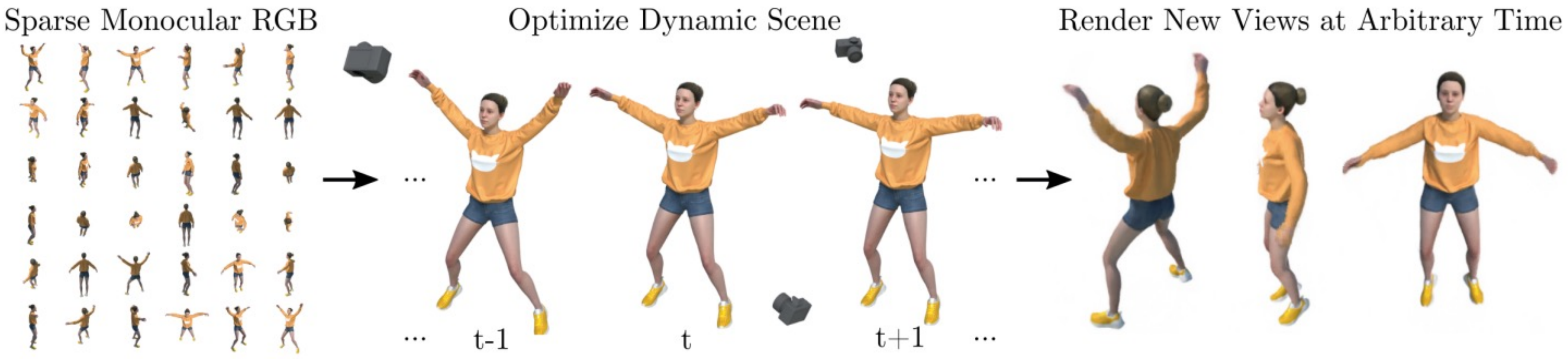
Limitations of NeRF

1. Scene specific and only static scene can be modeled.

GT image of a dynamic scene

Image generated with NeRF

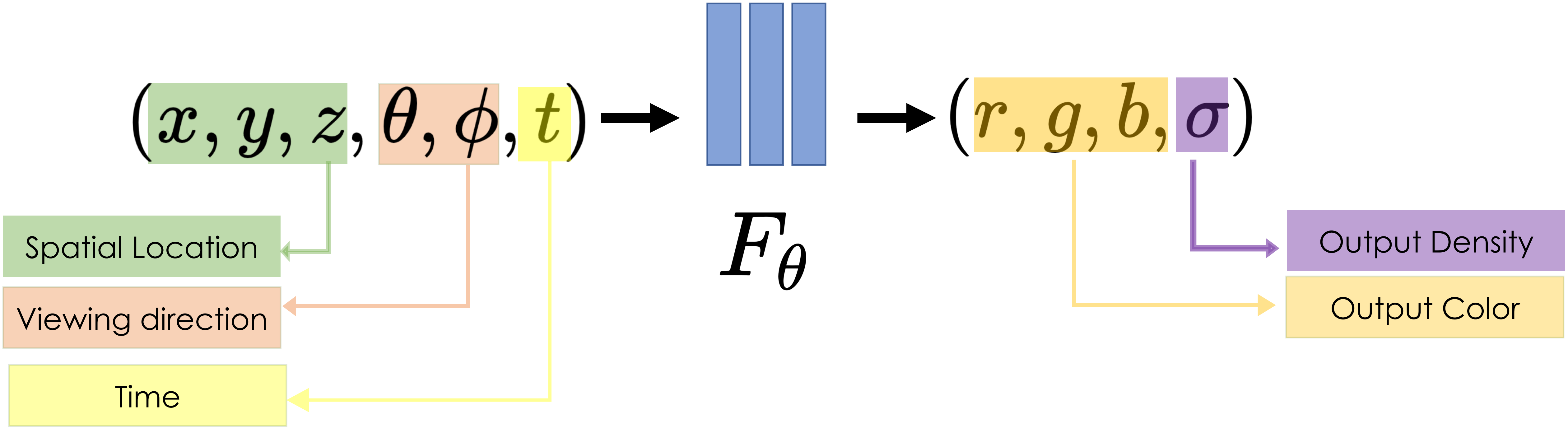
What about dynamic scenes?



Pumarola et al. CVPR'21

What about dynamic scenes?

Learn radiance field given 3d point, viewing direction and time



Can we learn this mapping directly using NeRF?

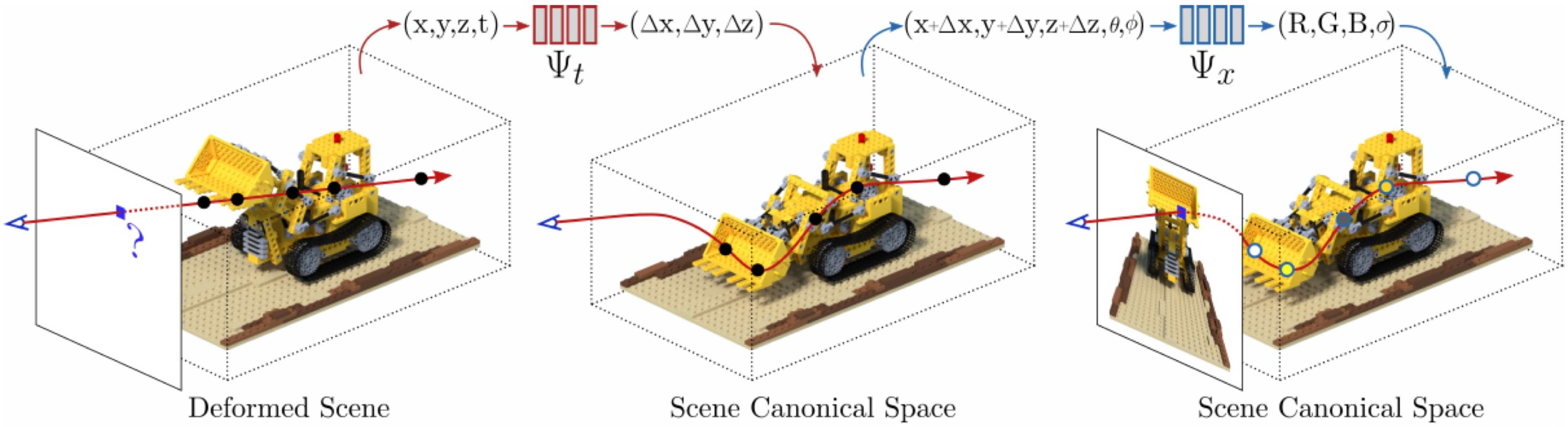
Learn radiance field given 3d point, viewing direction and time

Ground truth NeRF + time

Pumarola et al. CVPR'21

Proposed solution: D-NeRF

Learn canonical shape and radiance field in the canonical shape



Pumarola et al. CVPR'21

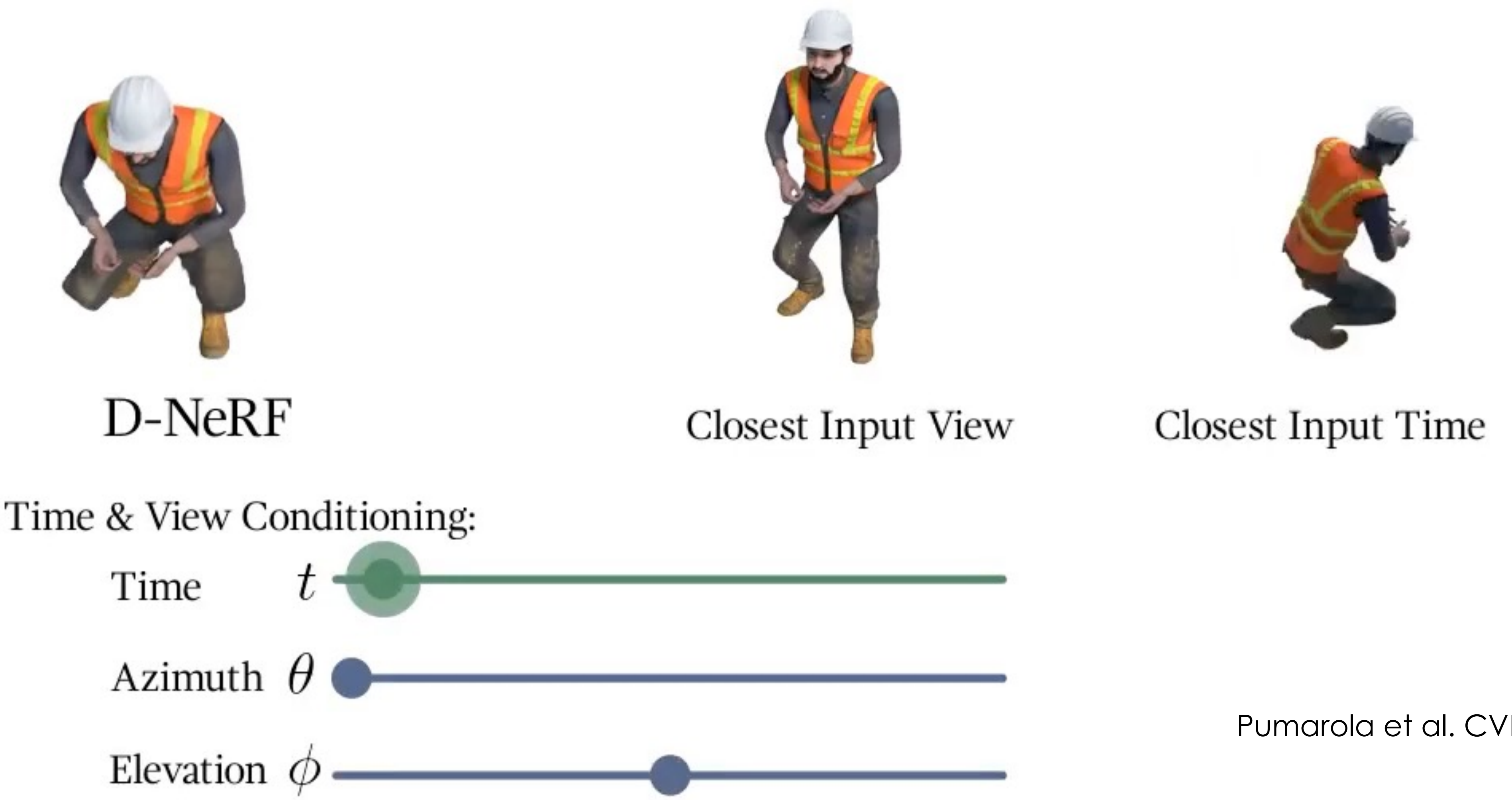
Proposed solution: D-NeRF

Learn canonical shape and radiance field in the canonical shape

Ground truth D-NeRF NeRF + time

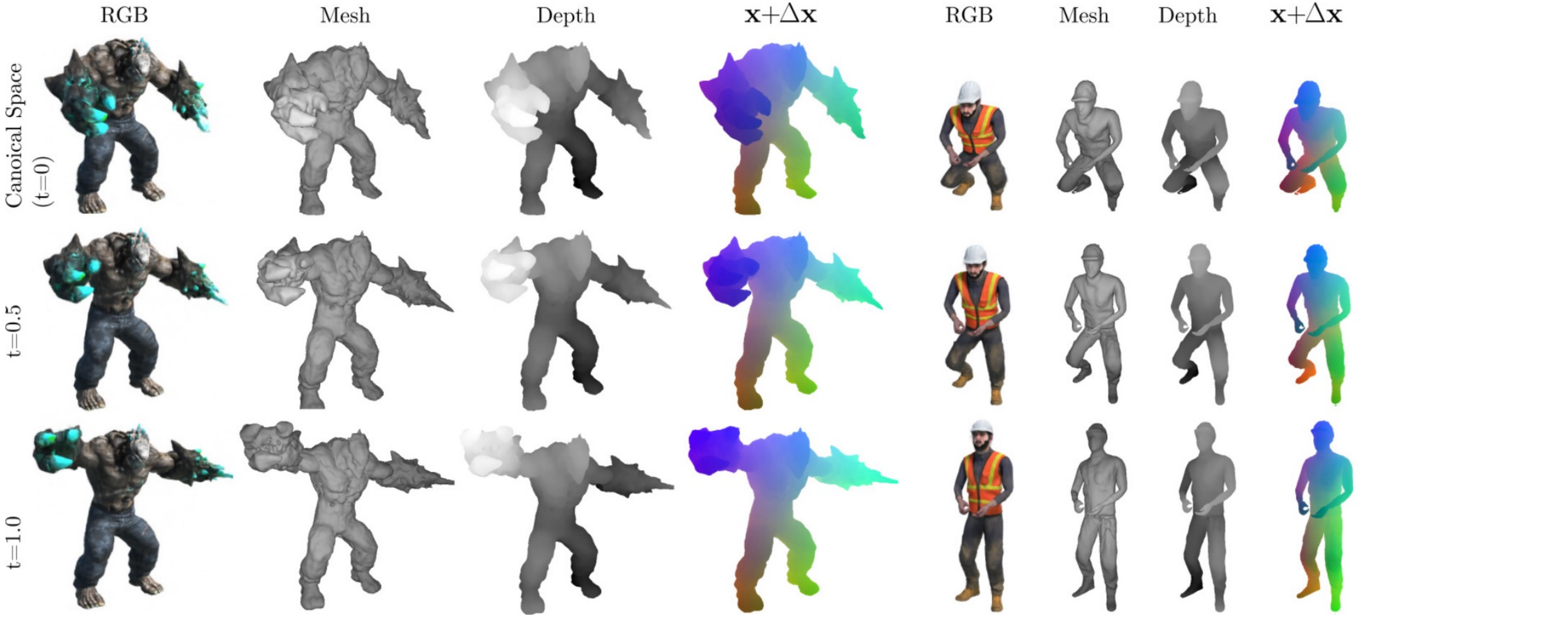
Pumarola et al. CVPR‘21

Synthesis Results



Pumarola et al. CVPR‘21

D-NeRF: Visualization of learned scene representation



Pumarola et al. CVPR'21

Conclusion:Dynamic Scenes with D-NeRF

- Disentangle time dependent deformation from neural rendering network.

Conclusion:Dynamic Scenes with D-NeRF

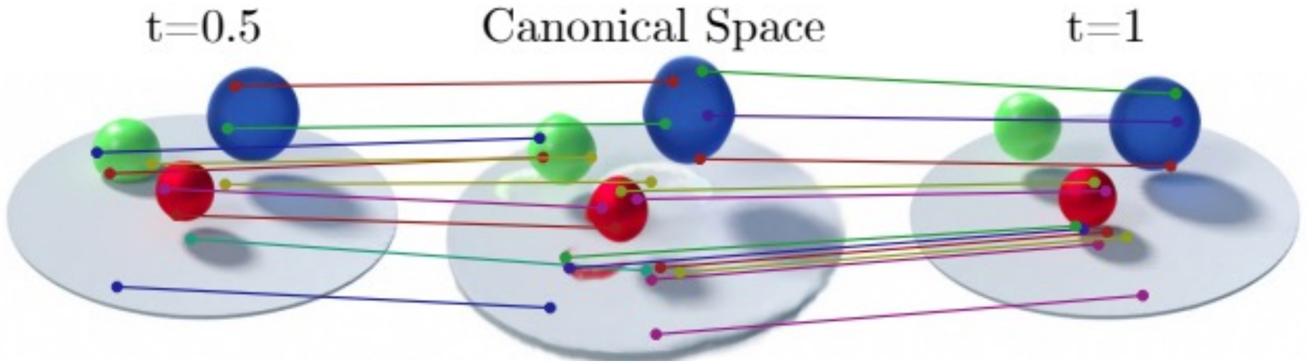
- Disentangle time dependent deformation from neural rendering network.
- Correspondence between canonical shape and deformed shape is defined by

D-NeRF Canonical Mapping
(color-coded as $\mathbf{x} + \Delta\mathbf{x}$)

Conclusion:Dynamic Scenes with D-NeRF

- Disentangle time dependent deformation from neural rendering network.
- Correspondence between canonical shape and deformed shape is defined by
- Time varying shading effects are modeled.

D-NeRF Canonical Mapping
(color-coded as $\mathbf{x} + \Delta\mathbf{x}$)



Limitations of NeRF:

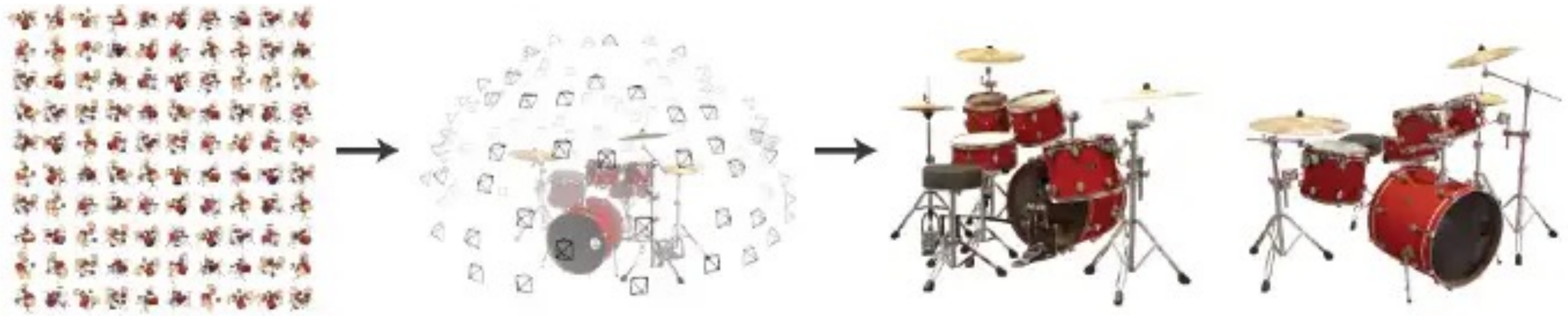
2. No editing and control

- Learned scene cannot be modified.
- Scene is memorized within the network

Limitations of NeRF

3. Generalization

- Scene specific models.
- Large number of images are needed



Control-NeRF

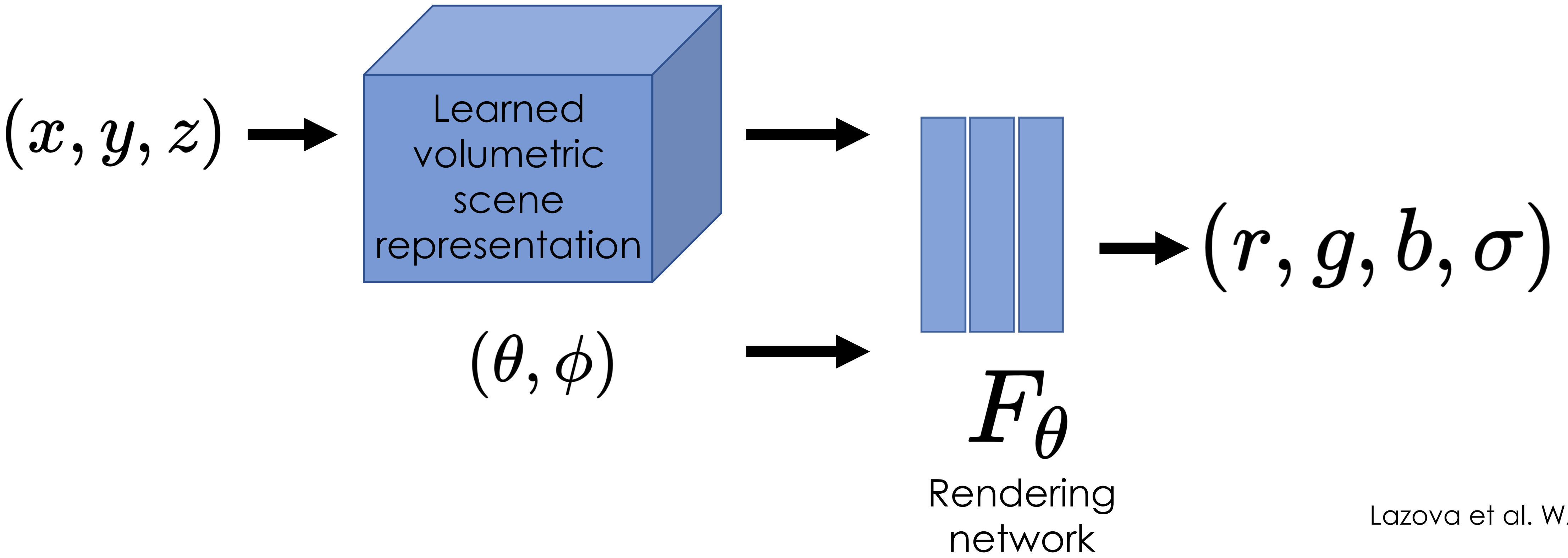
Control NeRF for scene manipulation and rendering

a) Original scenes b) Inserting objects from one scene into another c) Copying and moving objects within the scene

Lazova et al. WACV‘23

Control-NeRF

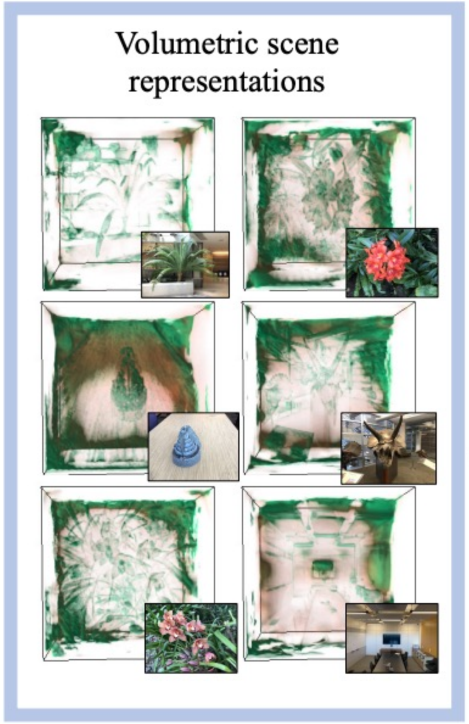
- **Prior Work**: Scene is memorized within the neural network, which makes compositing of scenes and editing hard.
- **Key Idea**: Decouple scene representation from neural rendering network.



Lazova et al. WACV‘23

Control-NeRF

1. Scene representation:



- Given a set of input images $\mathcal{I} = \{I_s^i\}_{i=1}^{N_s}$ from M training scenes
- Where $s \in \mathcal{S}$ is set of training scenes and $M = |\mathcal{S}|$
- Scene representation network learns a volumetric feature
- where $W{\times}H{\times}D$ is the spatial resolution of grid which feature vector of length F.

$$V_s \in \mathbf{R}^{WHDF}$$

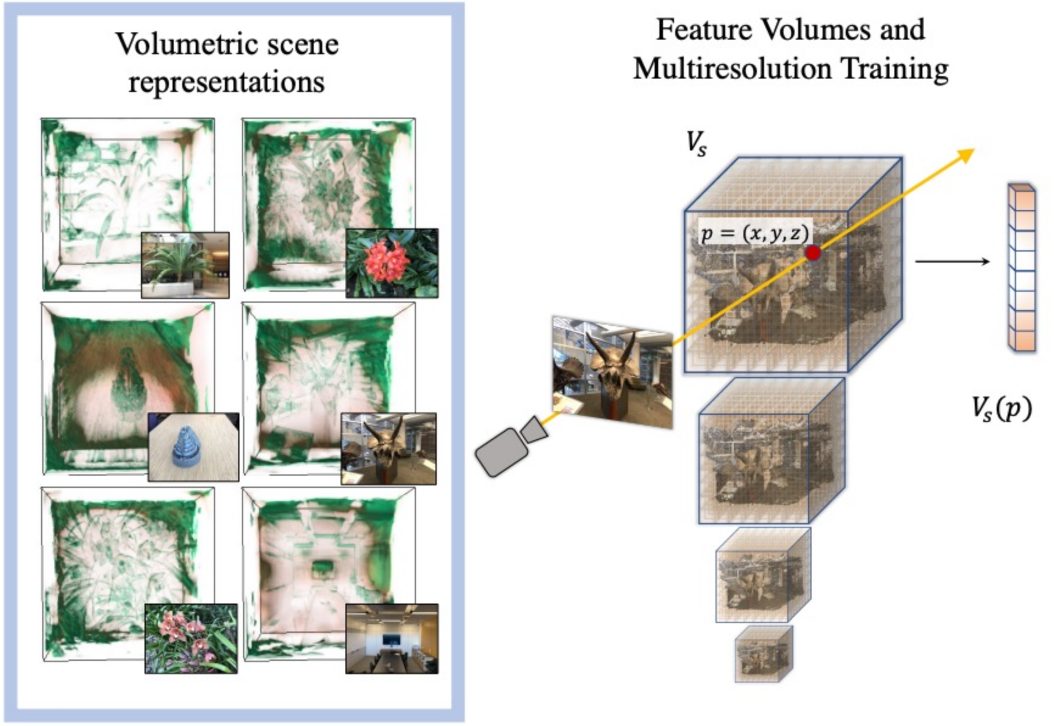
Learned volumetric feature for scene s

$W{\times}H{\times}D$ is the spatial resolution
F is length of feature vector

Lazova et al. WACV'23

Control-NeRF

2. Neural rendering network with feature volumes

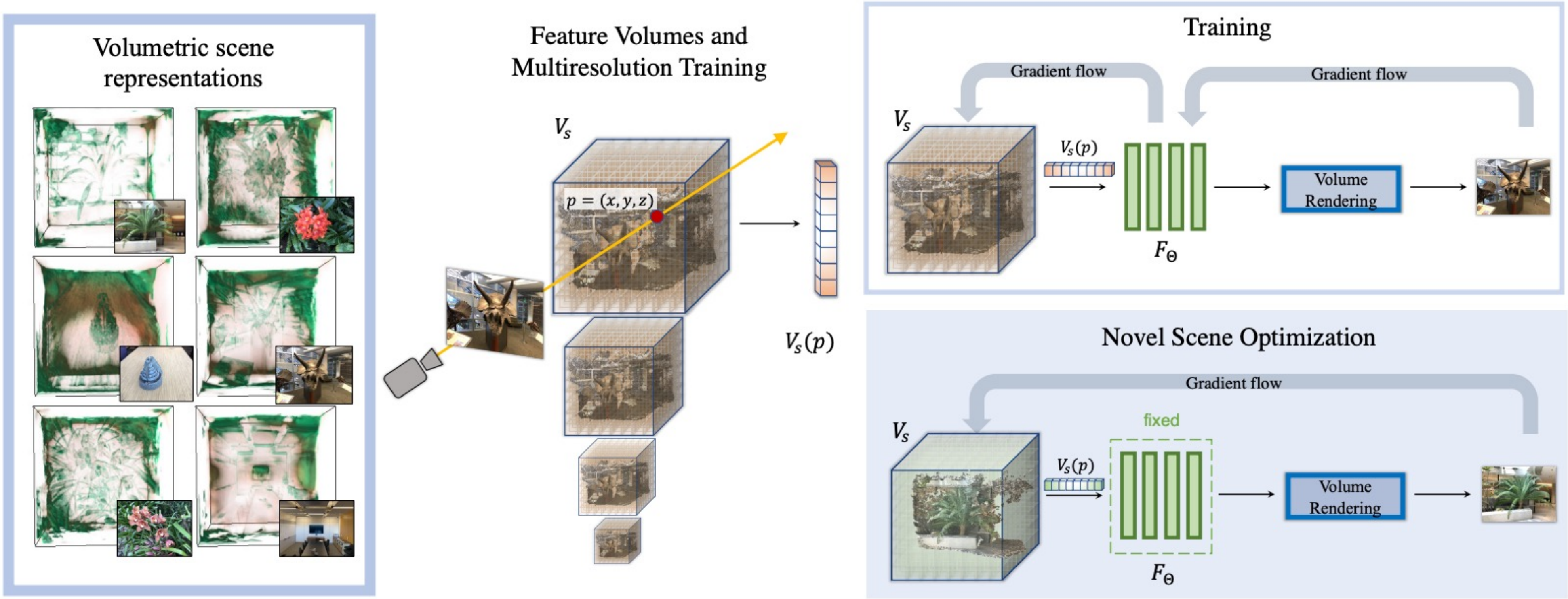


$V_s(p)$:Volumetric feature at query point p

Lazova et al. WACV'23

Control-NeRF

Training and Inference



Lazova et al. WACV'23

Control-NeRF: Training

1. **Multi-resolution Volume training**
   - Hierarchical training process is used to compute the volumes in a coarse-to-fine manner.
   - Train low resolution(16^3) volume till convergence.
   - Upsample the learnt feature volume and train till convergence

- Improved training time.
- High-quality image synthesis and manipulation.

Lazova et al. WACV'23

Control-NeRF: Training

2. **Multi Scene training**
   - Efficient training strategy: Sample one training scene and train for N iterations, before saving the volume grid

Lazova et al. WACV'23

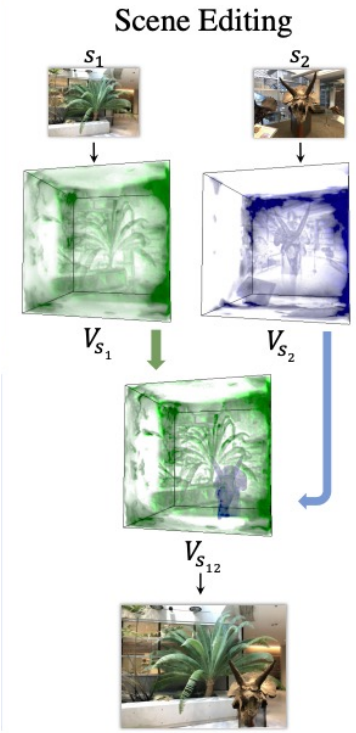
Control-NeRF: Training

3. **Generalization to Novel Scenes**
   - Fix neural rendering network and learn feature volume for novel scene.
   - Given sufficient training scenes, the learnt radiance function can be applied to optimize for novel scenes more efficiently.

Lazova et al. WACV'23

Control NeRF: Scene editing and manipulation

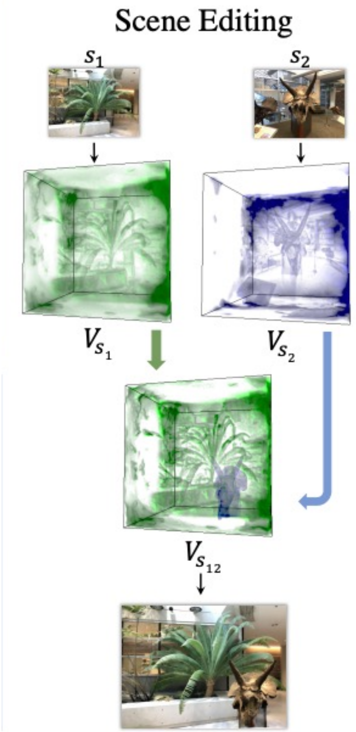
- Scene editing and composting with Control-NeRF:

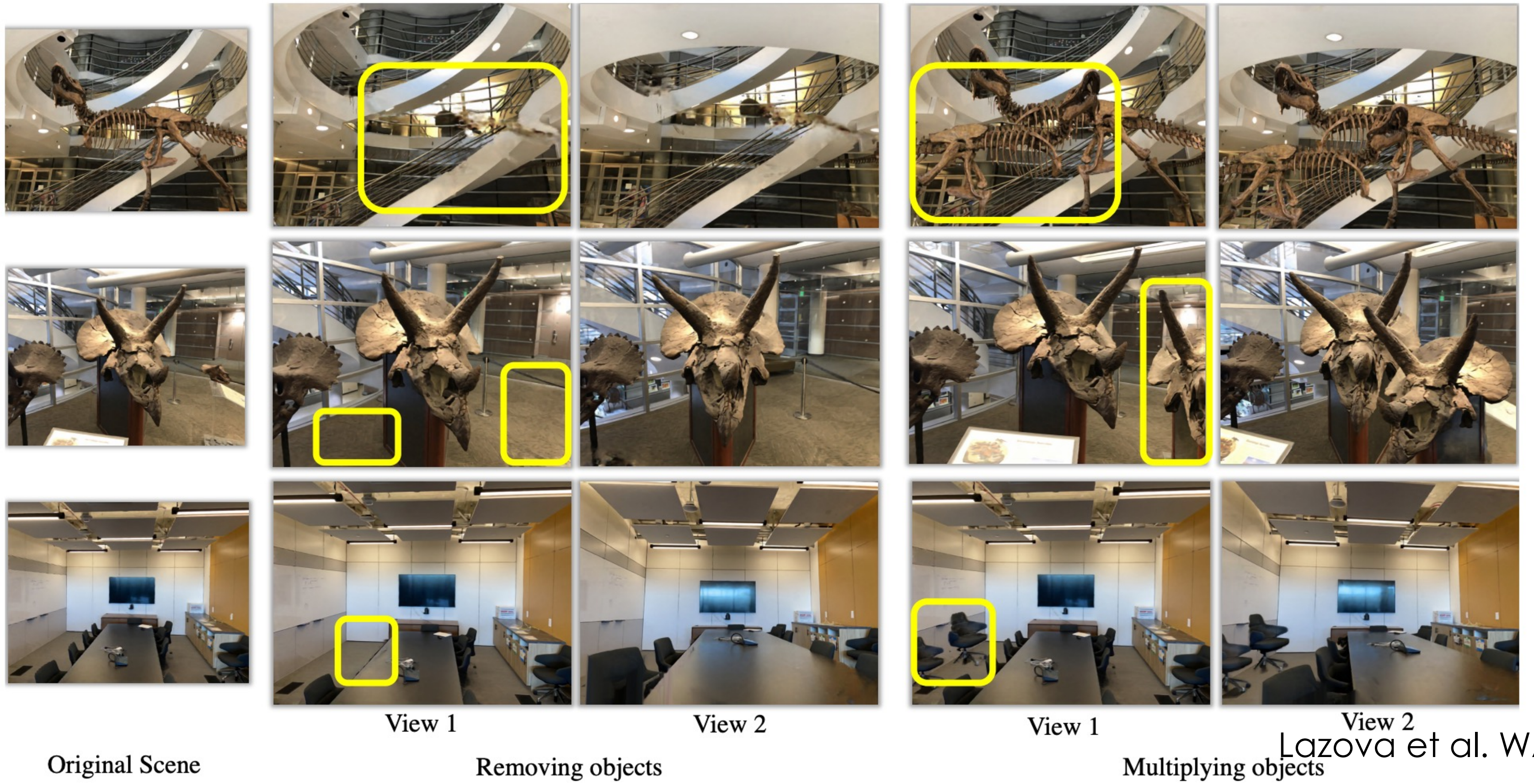


Lazova et al. WACV'23

Control NeRF: Scene editing and manipulation

- Scene editing and composting with Control-NeRF:





Lazova et al. WACV'23

Goal: Scene Editing

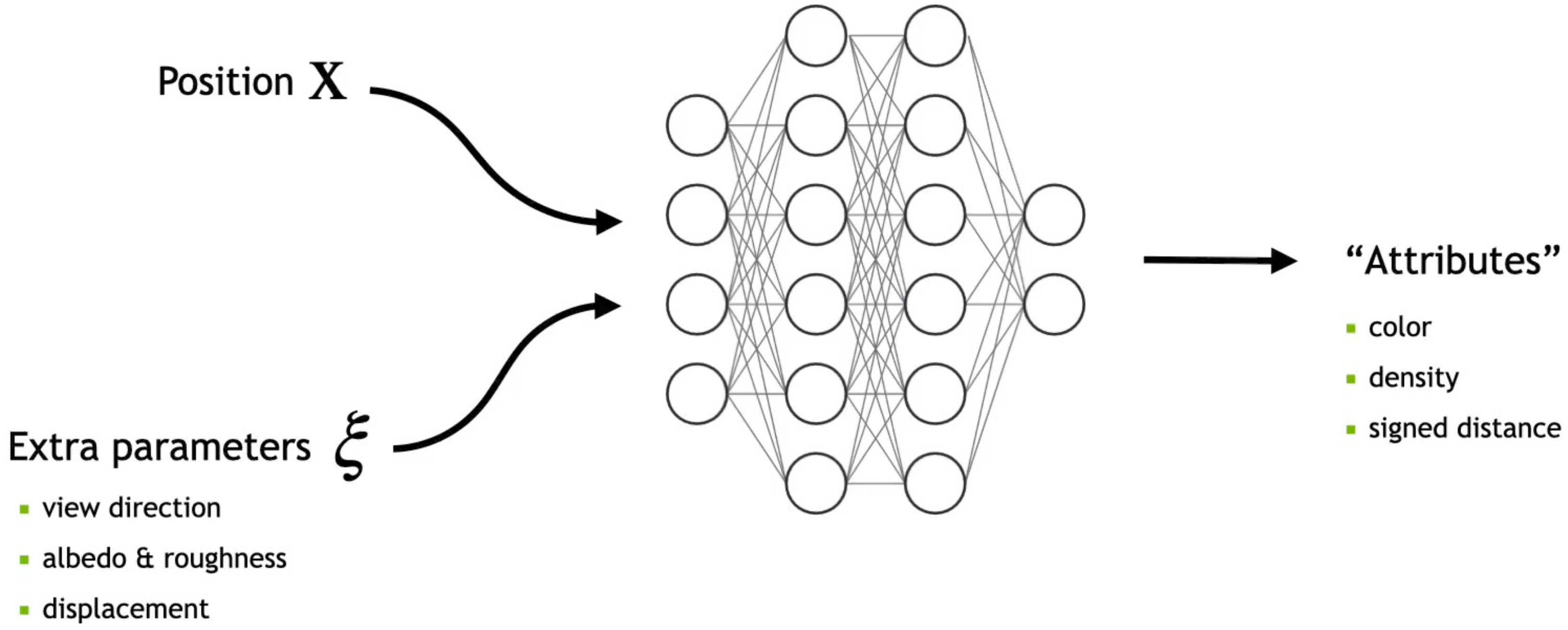
Limitations of NeRF:

4. Expensive training:

- Training is slow(10 hours-upto few days)
- Inference is also not real time

Neural Graphics Primitive

- An object (shape and appearance) represented by queries to a neural network.
  e.g. images, SDF, NeRF



Mueller et al. SIGGRAPH'22

Making NeRF(NGPs) faster

1. **Smaller neural network**

- Standard MLP with L layer, M neurons each, ReLU activations and no biases .For a constant batch size, the cost is:
    - Compute: $\mathcal{O}(M^2)$
    - Memory: $\mathcal{O}(M)$

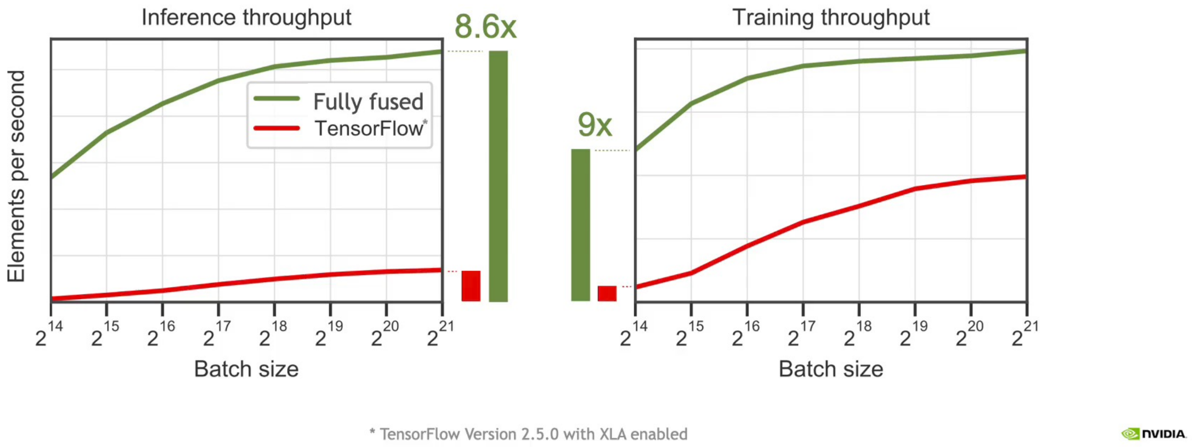
Mueller et al. SIGGRAPH'22

Making NeRF(NGPs) faster

1. **Smaller neural network**

- Standard MLP with L layer, M neurons each, ReLU activations and no biases .For a constant batch size, the cost is:
  - Compute: $\mathcal{O}(M^2)$
  - Memory: $\mathcal{O}(M)$

Entire neural network implemented as single (CUDA) kernel

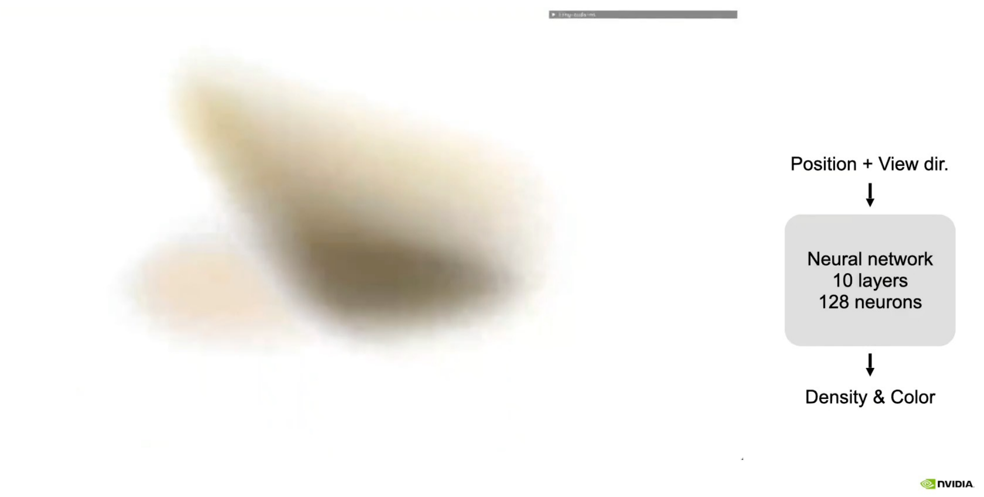


Mueller et al. SIGGRAPH'22

Positional Encoding in NeRF

2. Input encoding

Without positional encoding



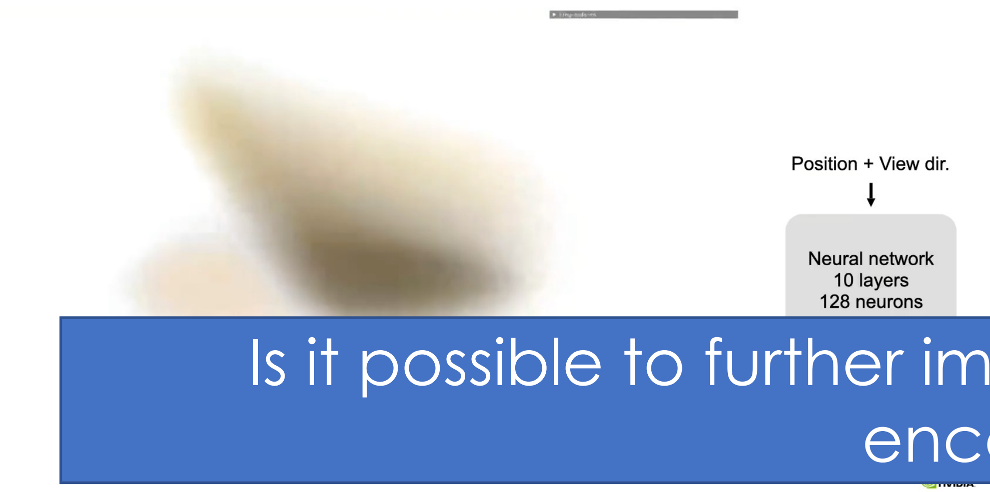
With positional encoding

Mueller et al. SIGGRAPH'22

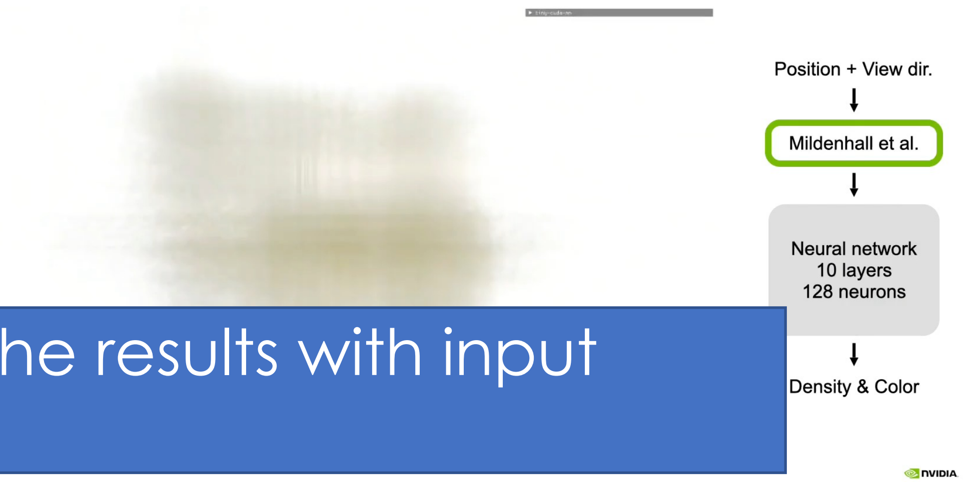
Positional Encoding in NeRF

2. Input encoding

Without positional encoding



With positional encoding



Is it possible to further improve the results with input encoding?

Mueller et al. SIGGRAPH'22

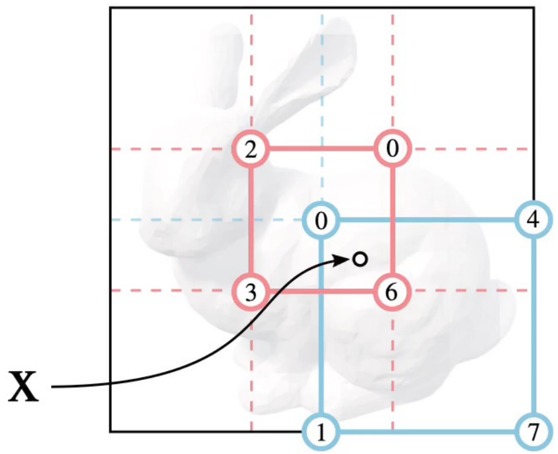
Making NeRF(NGPs) faster

2. Input encoding: Multiresolution Hash Encoding

Mueller et al. SIGGRAPH'22

Making NeRF(NGPs) faster

2. Input encoding: Multiresolution Hash Encoding



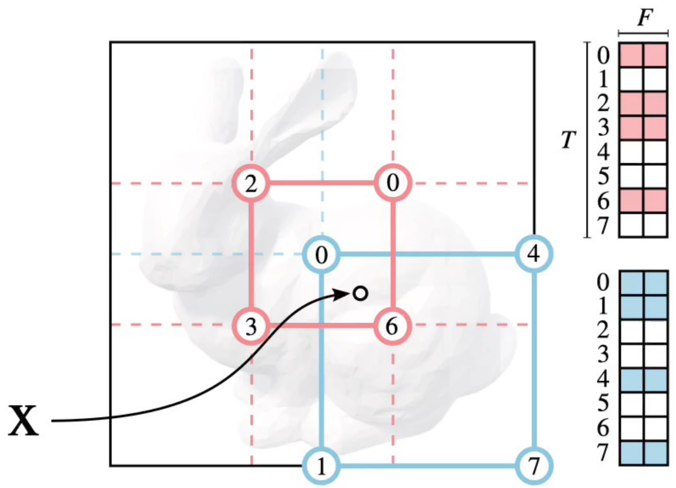
Multiresolution grids:

- Automatic level of details

Mueller et al. SIGGRAPH'22

Making NeRF(NGPs) faster

2. Input encoding: Multiresolution Hash Encoding

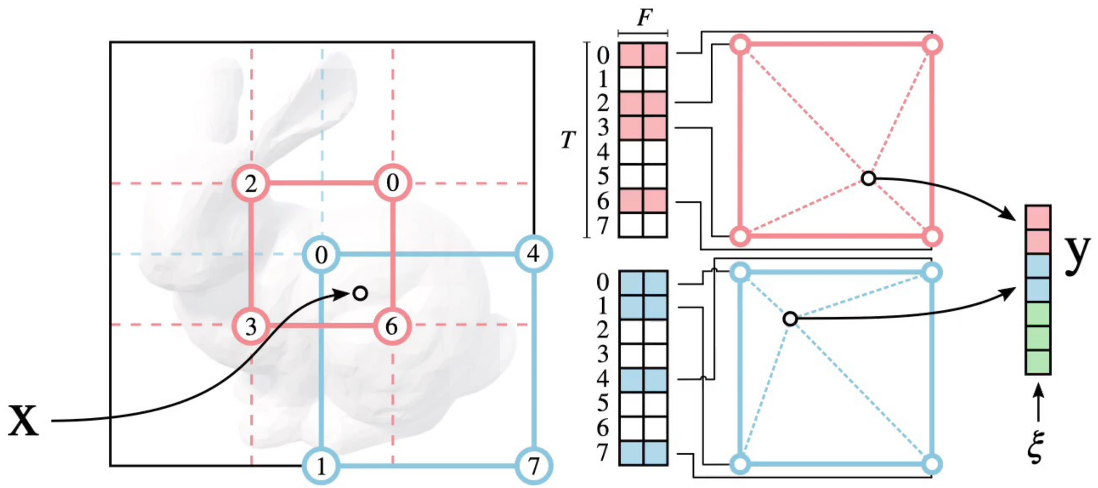


- Task agnostic
- Fast query and computation
- Table size T control quality vs. memory

Mueller et al. SIGGRAPH'22

Making NeRF(NGPs) faster

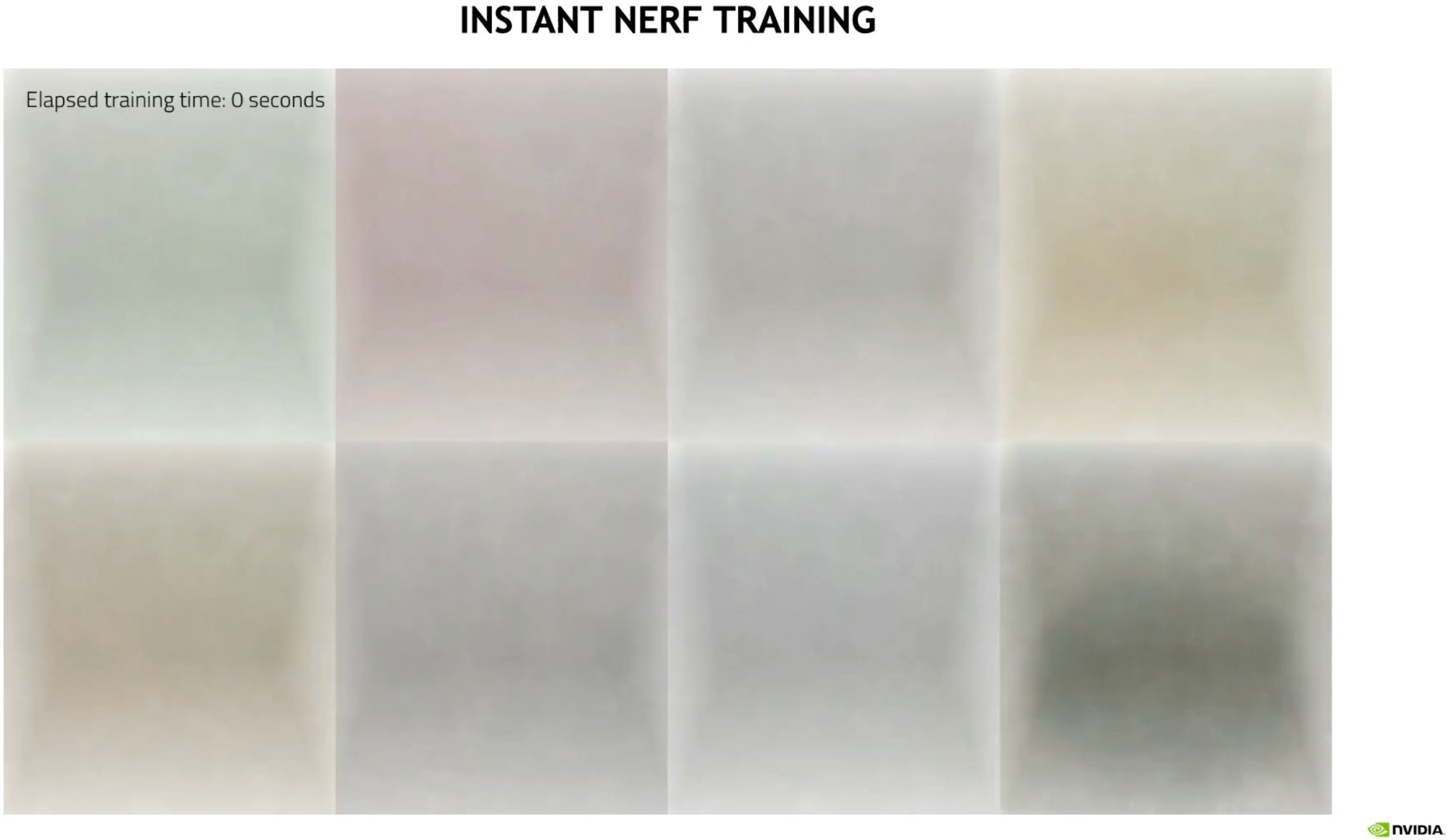
2. Input encoding: Multiresolution Hash Encoding



- Linear interpolation for continuous query.

Mueller et al. SIGGRAPH'22

Instant NeRF training



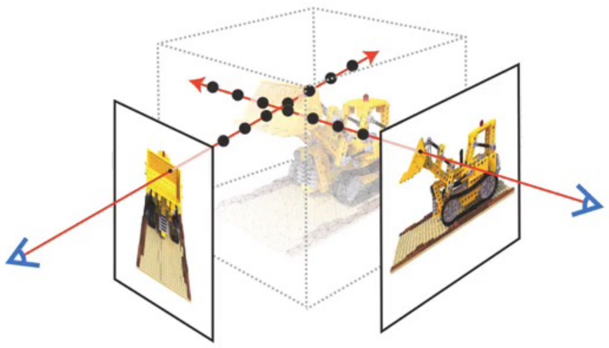
Mueller et al. SIGGRAPH‘22

Instant SDF training

Mueller et al. SIGGRAPH'22

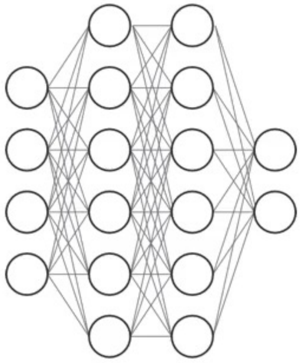
Conclusion: Instant-NGP

Rendering/training algorithm



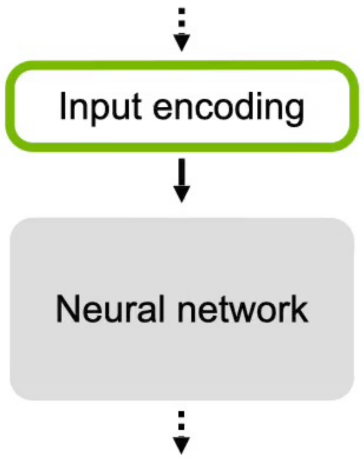
- Task-specific GPU implementation
- **10-100x faster than naïve tensor-based approach**

Small neural network



- Fully fused implementation
- **5-10x faster than TensorFlow**

"Good" input encoding

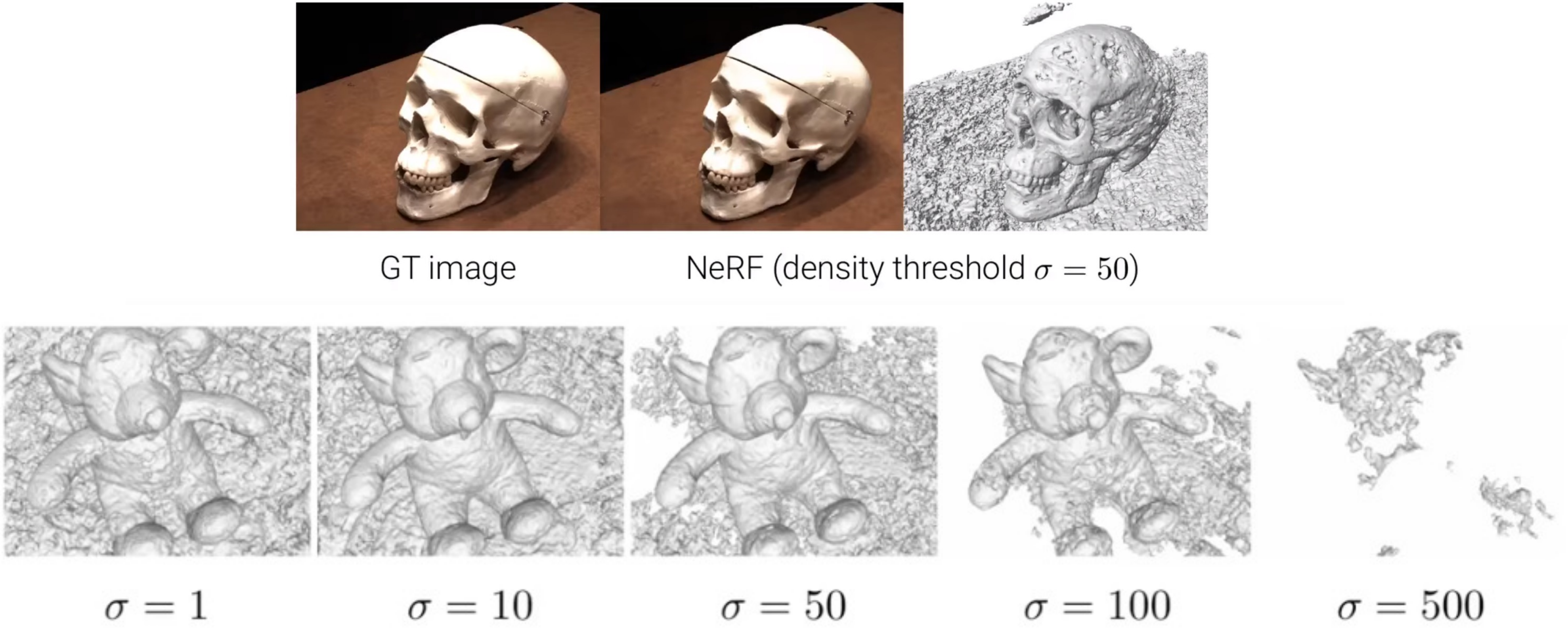


- Multiresolution hash encoding
- **Better speed-vs-quality tradeoff than prior work**
- **Task agnostic**

Mueller et al. SIGGRAPH'22

Limitations of NeRF

5. Surface extracted is not accurate and depends on threshold.

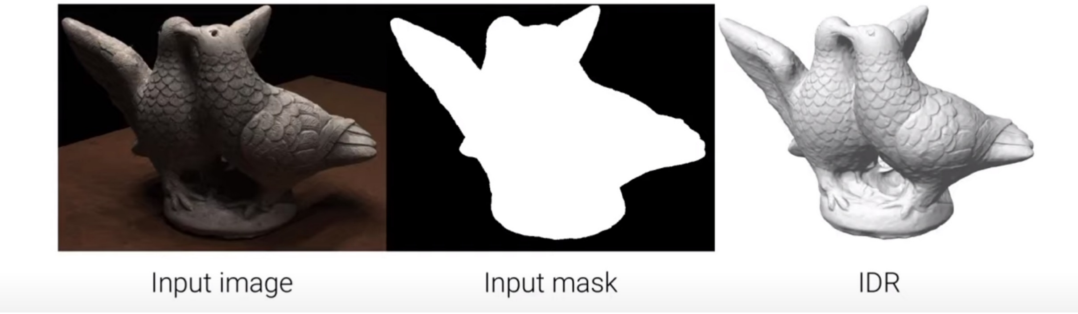


Surface Rendering v/s Volume Rendering

Surface Rendering
(Implicit Surfaces)

\+ High quality geometry
\+ Clear surface definition

— Mask supervision required
— Texture mapping is blurry



Surface Rendering v/s Volume Rendering

Surface Rendering (Implicit Surfaces)

+ High quality geometry
+ Clear surface definition

− Mask supervision required
− Texture mapping is blurry

Input image Input mask IDR

Volume Rendering (Radiance fields)

− Surface is approximated

+ Without mask supervision
+ High quality novel views and sharp textures/colors

Best of both worlds

Surface Rendering
(Implicit Surfaces)

+ High quality geometry
+ Clear surface definition

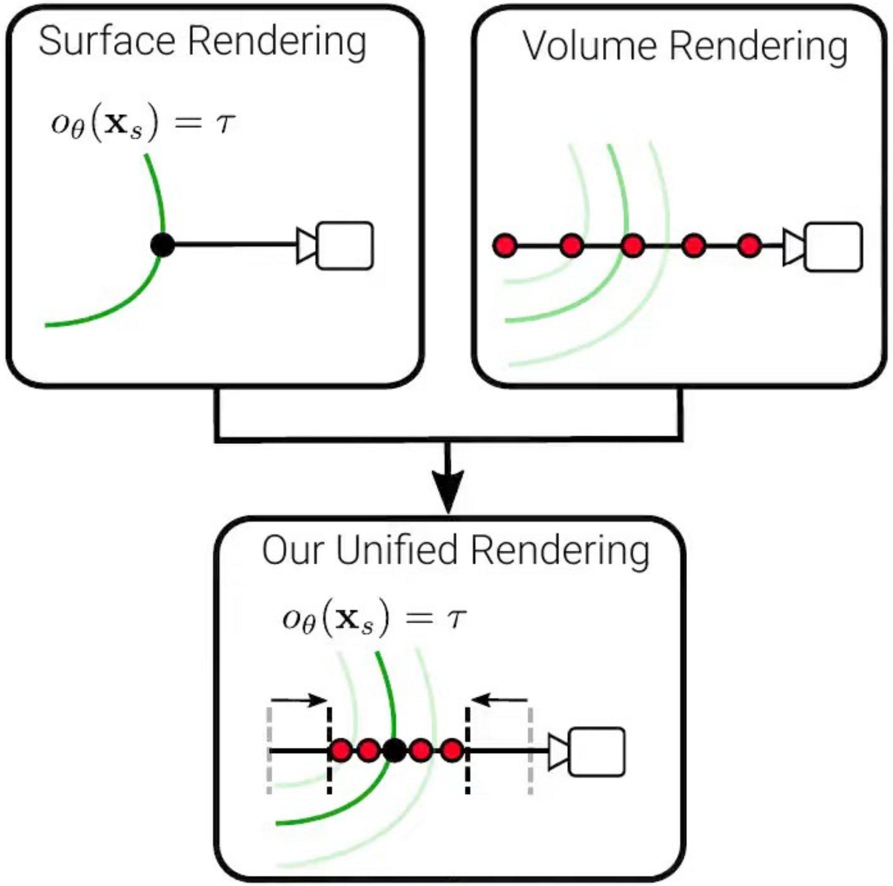
~~Mask supervision required~~
~~Texture mapping is blurry~~

Volume Rendering
(Radiance fields)

~~Surface is approximated~~

+ Without mask supervision
+ High quality novel views and sharp textures/colors

Unifying Implicit Surfaces and Radiance Fields



Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

NeRF Volume rendering:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i$$

NeRF Volume rendering with density

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \Pi_{j=1}^{i} (1 - \alpha_j)\mathbf{c}_i$$

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i))$$

Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

NeRF Volume rendering with density

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \Pi_{j=1}^{i}(1 - \alpha_j)\mathbf{c}_i$$

Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

NeRF Volume rendering with density

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \Pi_{j=1}^{i} (1 - \alpha_j) \mathbf{c}_i$$

Key Idea: For solid objects, $\alpha_i = 1 - \exp(-\sigma_i \delta_i))$, corresponds to an occupancy field $\boldsymbol{o_i}$ at ***i*** *th* sample.

$$\hat{C}_v(\mathbf{r}) = \sum_{i=1}^{N} o_i \Pi_{j=1}^{i} (1 - o_j) \mathbf{c}_i$$

Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

NeRF Volume rendering with density

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \Pi_{j=1}^{i} (1 - \alpha_j) \mathbf{c}_i$$

Key Idea: For solid objects, $\alpha_i = 1 - \exp(-\sigma_i \delta_i))$, corresponds to an occupancy field $\boldsymbol{o_i}$ at ***i*** *th* sample.

$$\hat{C}_v(\mathbf{r}) = \sum_{i=1}^{N} o_i \Pi_{j=1}^{i} (1 - o_j) \mathbf{c}_i$$

Given occupancy of surface, we can now render the same scene with surface rendering.

Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

Key Idea:

- Volume rendering in early stage:
  - Optimization without mask

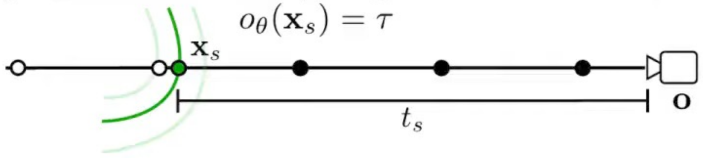
- Surface rendering in later stage:
  - Level-set surfaces

Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

Rendering Procedure

- Find surface along a ray: uniform sampling + iterative secant method

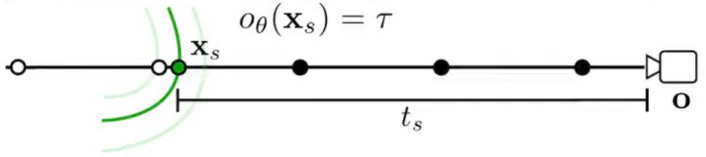


Oechsle et al. ICCV'21

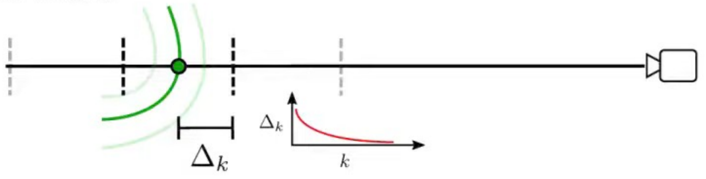
Unifying Implicit Surfaces and Radiance Fields

Rendering Procedure

- Find surface along a ray: uniform sampling + iterative secant method



- Define interval at the surface

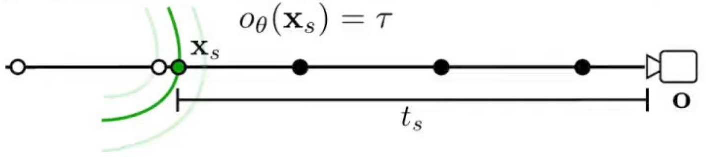


Oechsle et al. ICCV'21

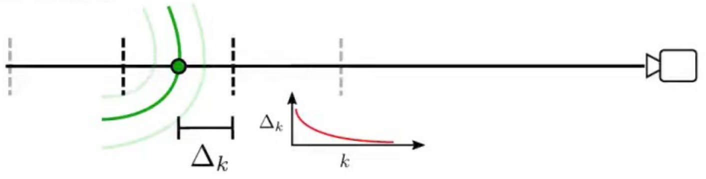
Unifying Implicit Surfaces and Radiance Fields

Rendering Procedure

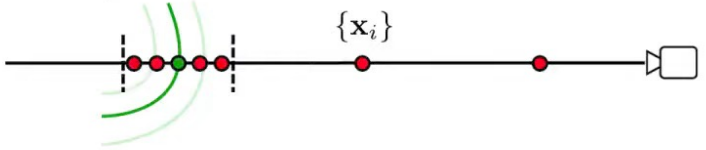
- Find surface along a ray: uniform sampling + iterative secant method



- Define interval at the surface



- Volume rendering with occupancies



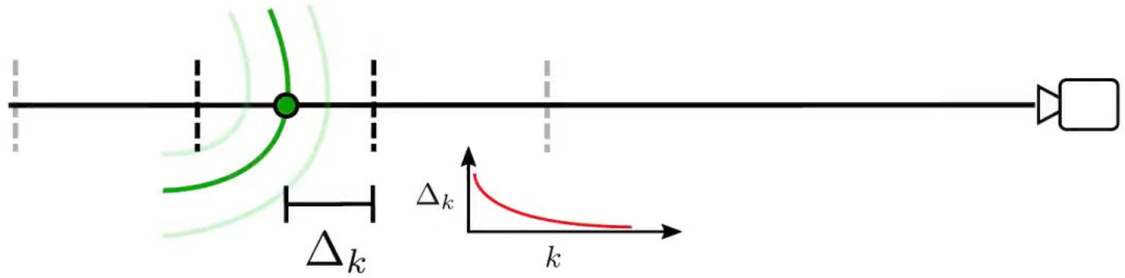
Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

Rendering Procedure

Transition from Volume rendering to Surface rendering

- Exponential decay of interval Δ_k wrt. iterations k



Volume Rendering
Rendering

Δk

decreases

Surface

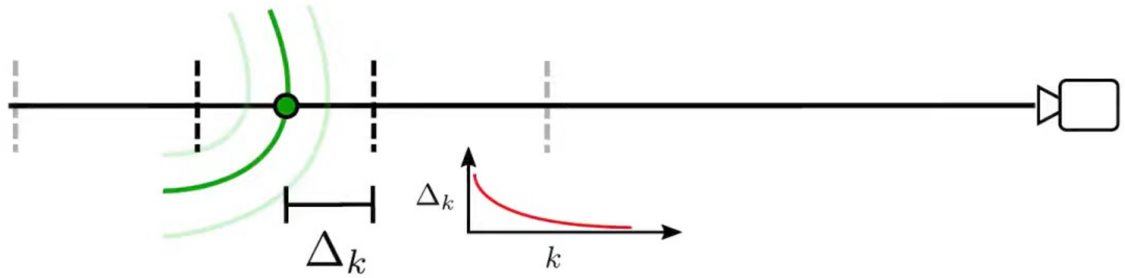
Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

Rendering Procedure

Transition from Volume rendering to Surface rendering

- Exponential decay of interval Δ_k wrt. iterations k



Theorem

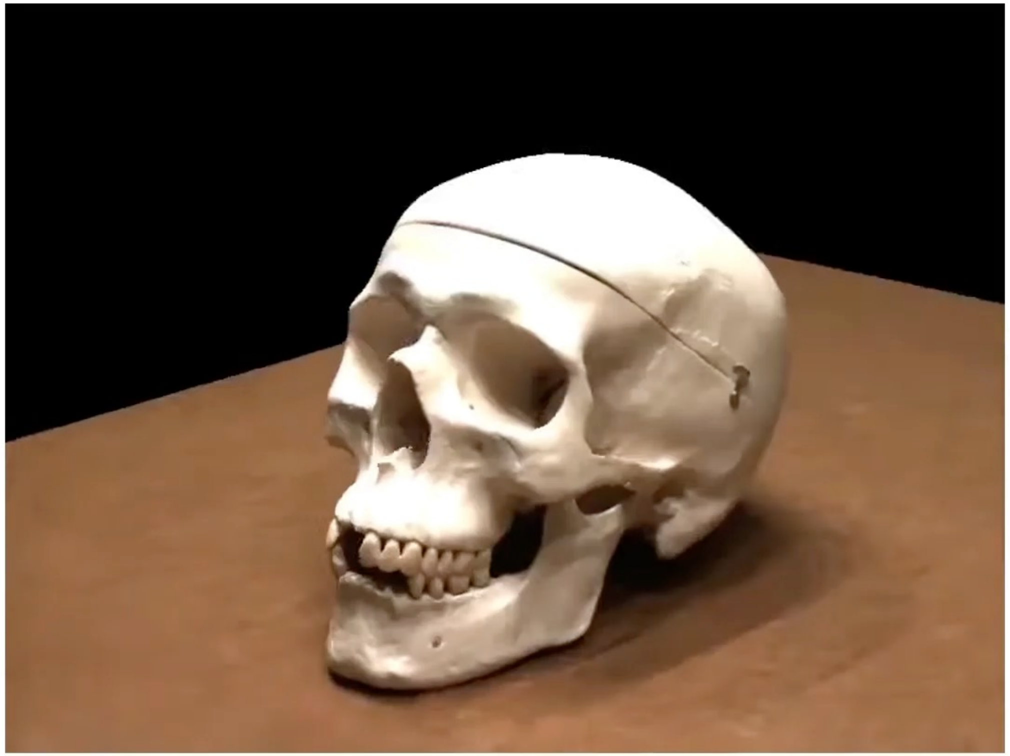
Volume and surface rendering become equivalent when reducing the interval and increasing the number of samples.

$$\lim_{\substack{\Delta \to 0 \\ N \to \infty}} \hat{C}_v(\mathbf{r}) = \hat{C}_s(\mathbf{r})$$

Oechsle et al. ICCV'21

Unifying Implicit Surfaces and Radiance Fields

Comparison on the DTU MVS dataset



Ours

Oechsle et al. ICCV'21

More on NeRF

Dynamic NeRFs:

- TöRF Attal et al., NeurIPS 2021
- NSFF, Li et al., CVPR 2021
- ....

NeRF from few images:

- PixelNeRF, Yu et al., CVPR 2021
- ....

NeRF + implicit surfaces:

- NeUS, Wang et al., NeurIPS 2021
- VolSDF, Yariv et al., NeurIPS 2021
- ....

- .... Many more

Checkout NeuralFields: https://neuralfields.cs.brown.edu/