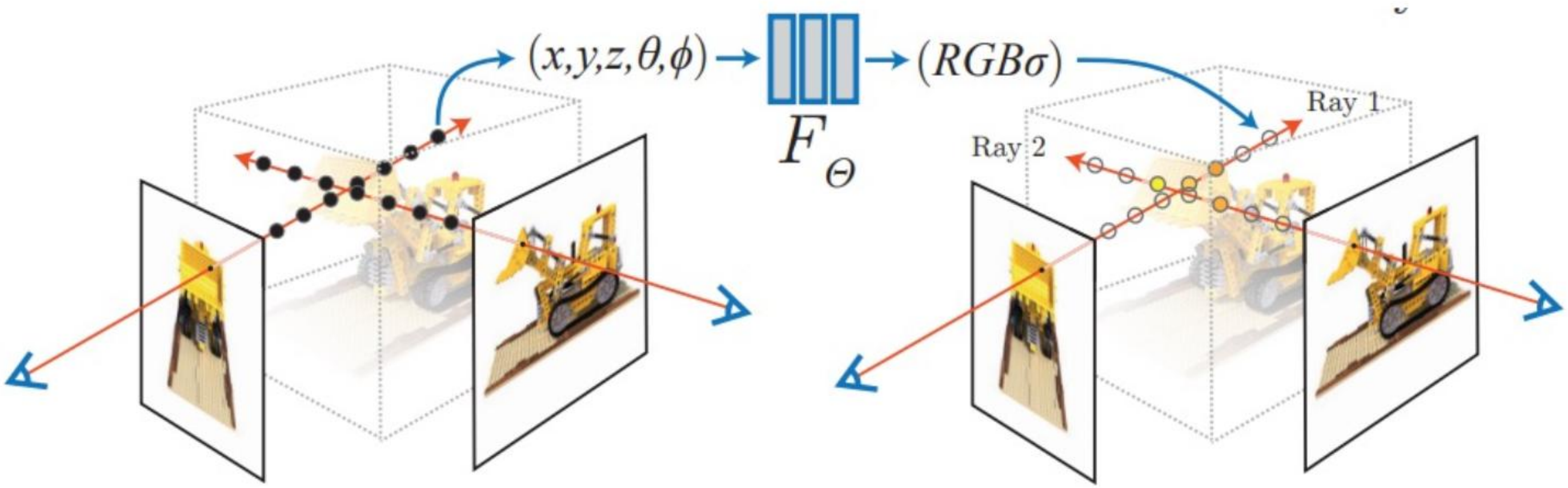
11. 3D Neural Rendering and Reconstruction



Outline

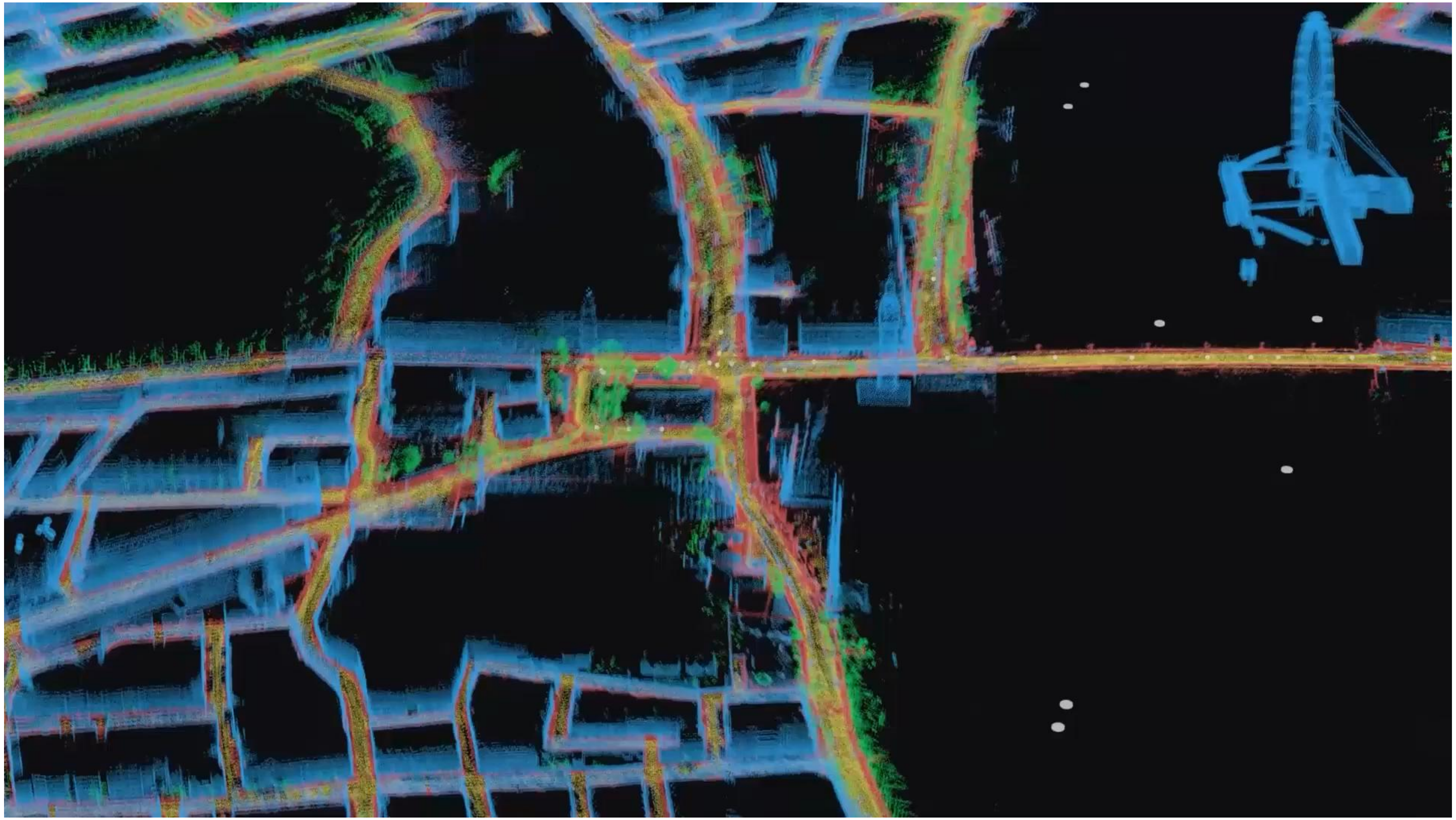
- Neural Radiance Fields (NeRF)
- 3D Gaussian Splatting (3DGS)

Task: Image-based Rendering

Input images　　3D model　　Novel views

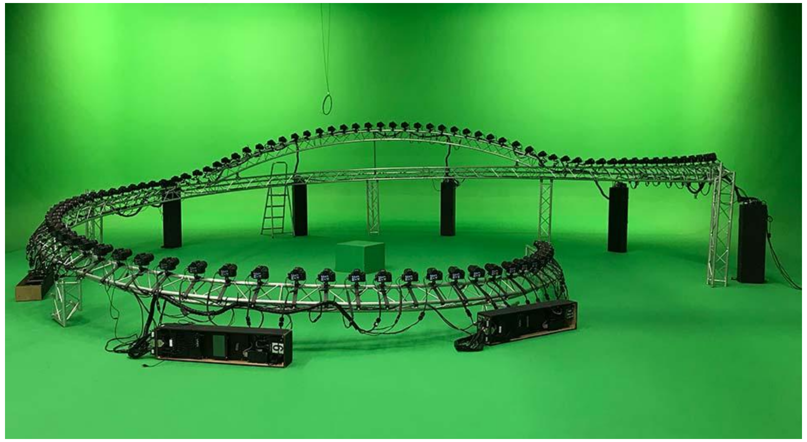
NeRF in the wild: Neural radiance fields for unconstrained photo collections. CVPR. 2021.

Applications: (1) VR City Tour



Google Immersive View

Applications: (2) Film special effects



Film "Matrix"

Applications: (3) Immersive remote viewing

https://www.youtube.com/clip/Ugkx200l8Vsn3ciOS4l38PSXl_gOqevhiN35

Applications: (4) Immersive memory

https://www.youtube.com/watch?v=TX9qSaGXFyg

Applications: (5) Simulator

https://www.youtube.com/watch?v=RVFIDEuNtt0

Image-based Rendering

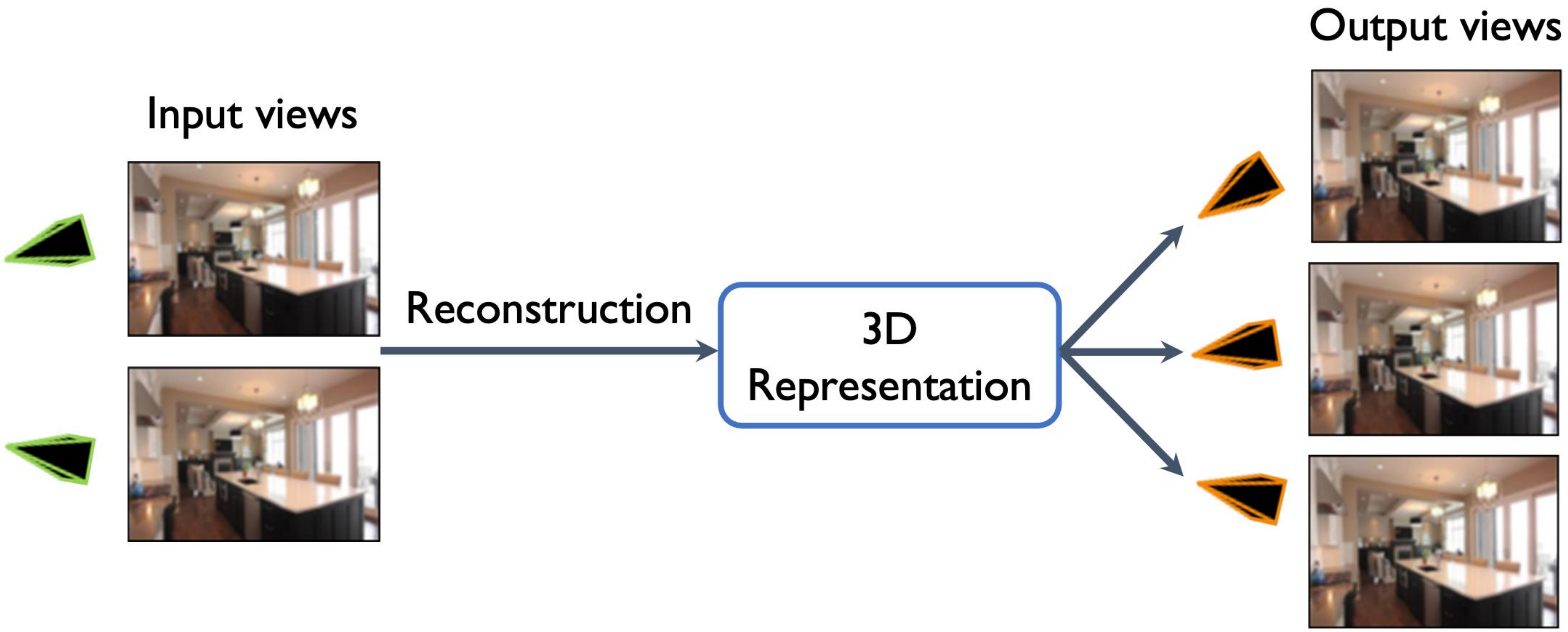
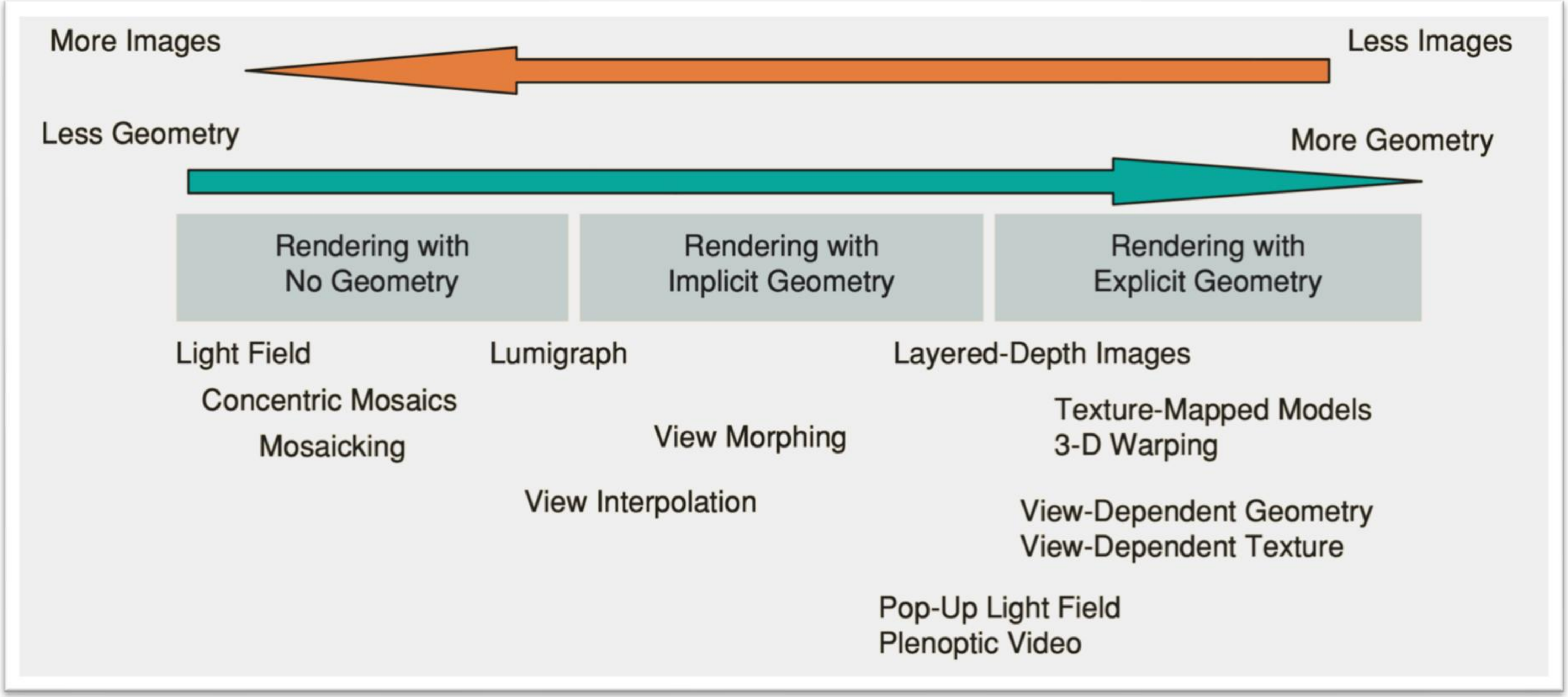


Image credit: Noah Snavely

Traditional Methods



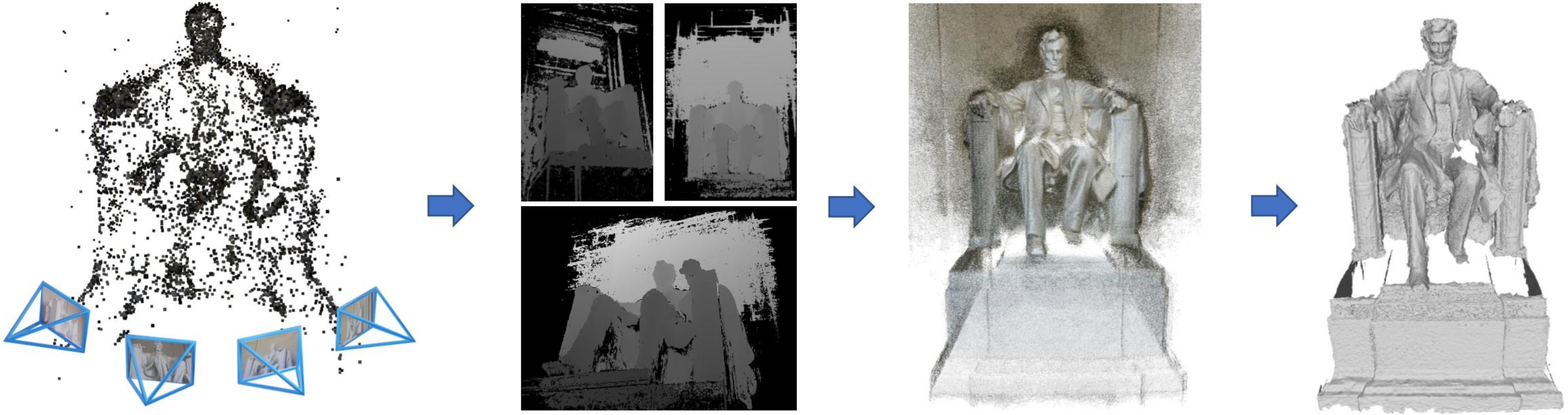
A Review of Image-based Rendering Techniques, Visual Communications and Image Processing 2000.

Common 3D Representation 1: Surface-based Representations

- Textured 3D Mesh

Common 3D Representation 1: Surface-based Representations

- Reconstruction process：



SfM   MVS   Fusion   Mesh extraction

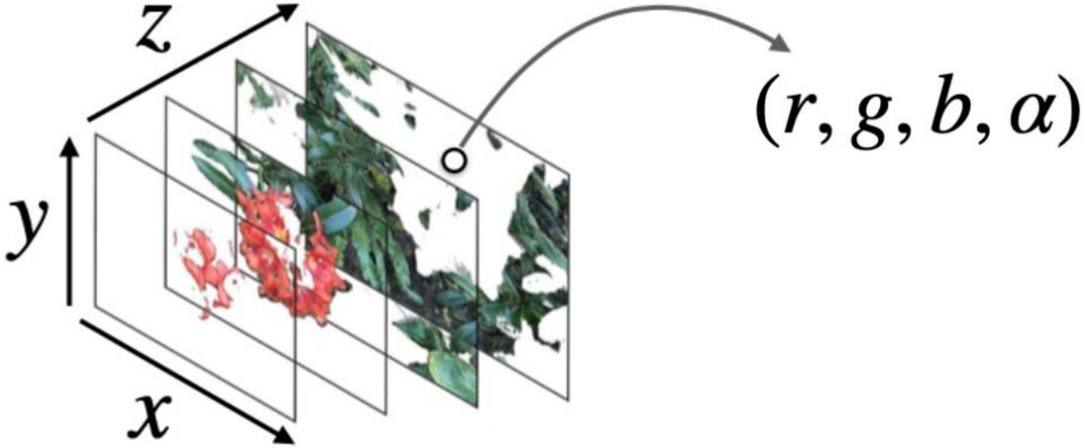
Challenges of Surface-based Representations

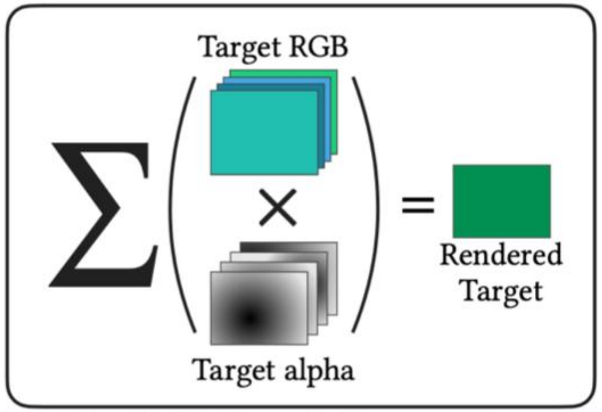
Two main challenges：

- Difficult to reconstruct high-quality surface geometry.
- Difficult to represent highly complex three-dimensional scenes.

Common 3D Representation 2: Volume-based Representations

- Multi-Plane image (MPI)





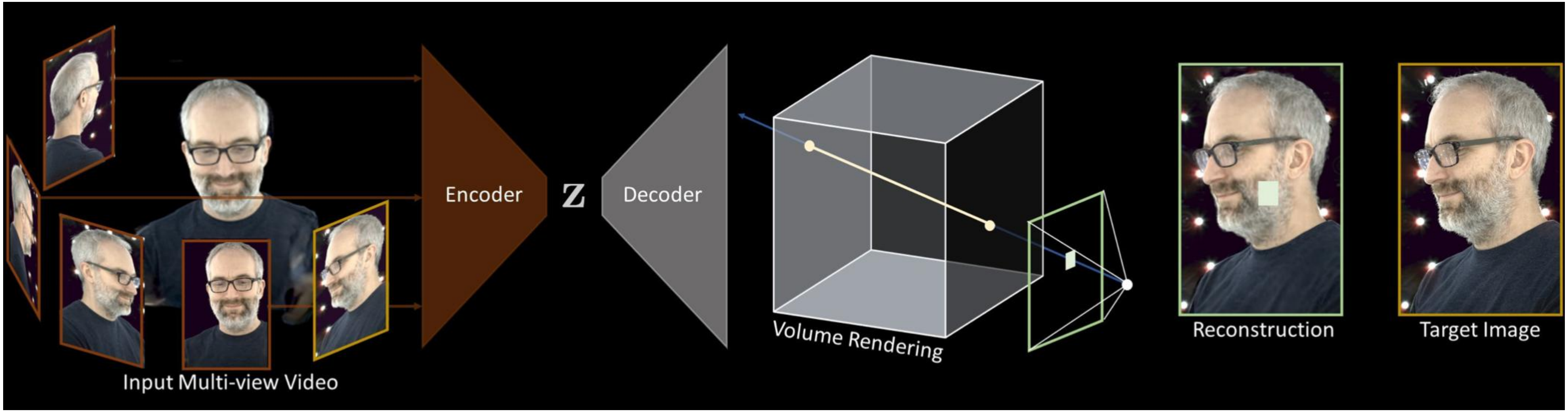
Blend RGBA renderings together
to render final output image

The Demo of Multi-Plane Image

https://augmentedperception.github.io/deepview/

Common 3D Representation 2: Volume-based Representations

- RGB-alpha volume



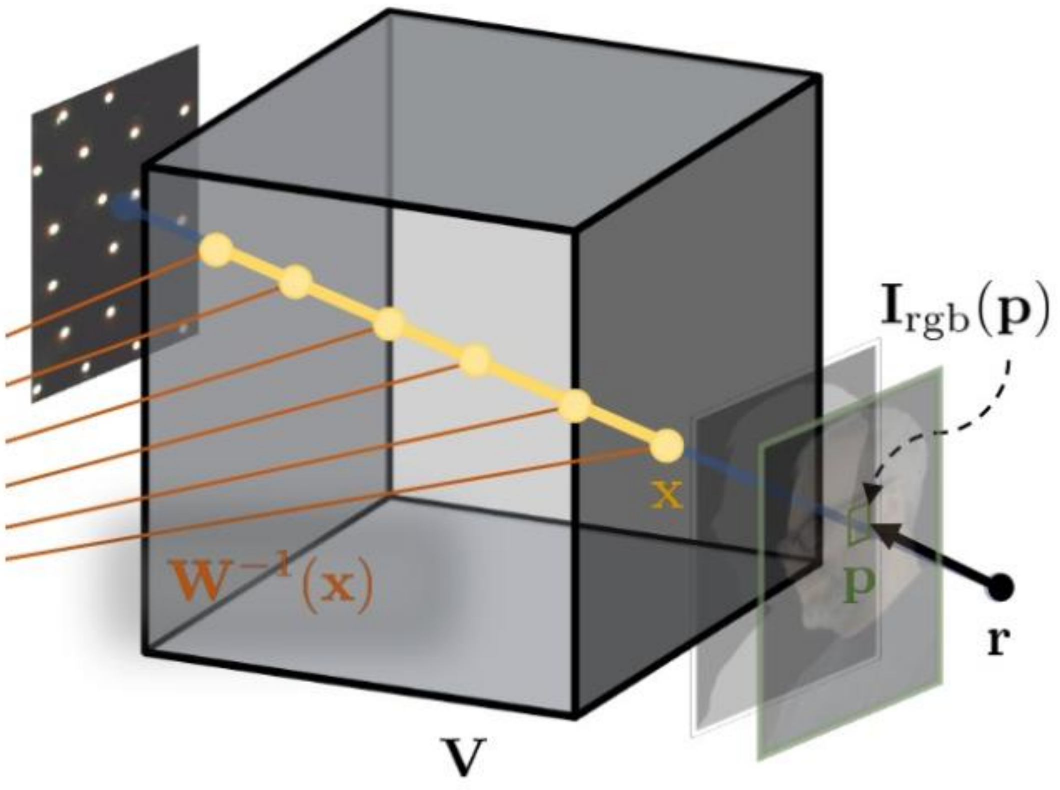
Learning Dynamic Renderable Volumes from Images. SIGGRAPH 2019.

The Demo of RGB-alpha Volume

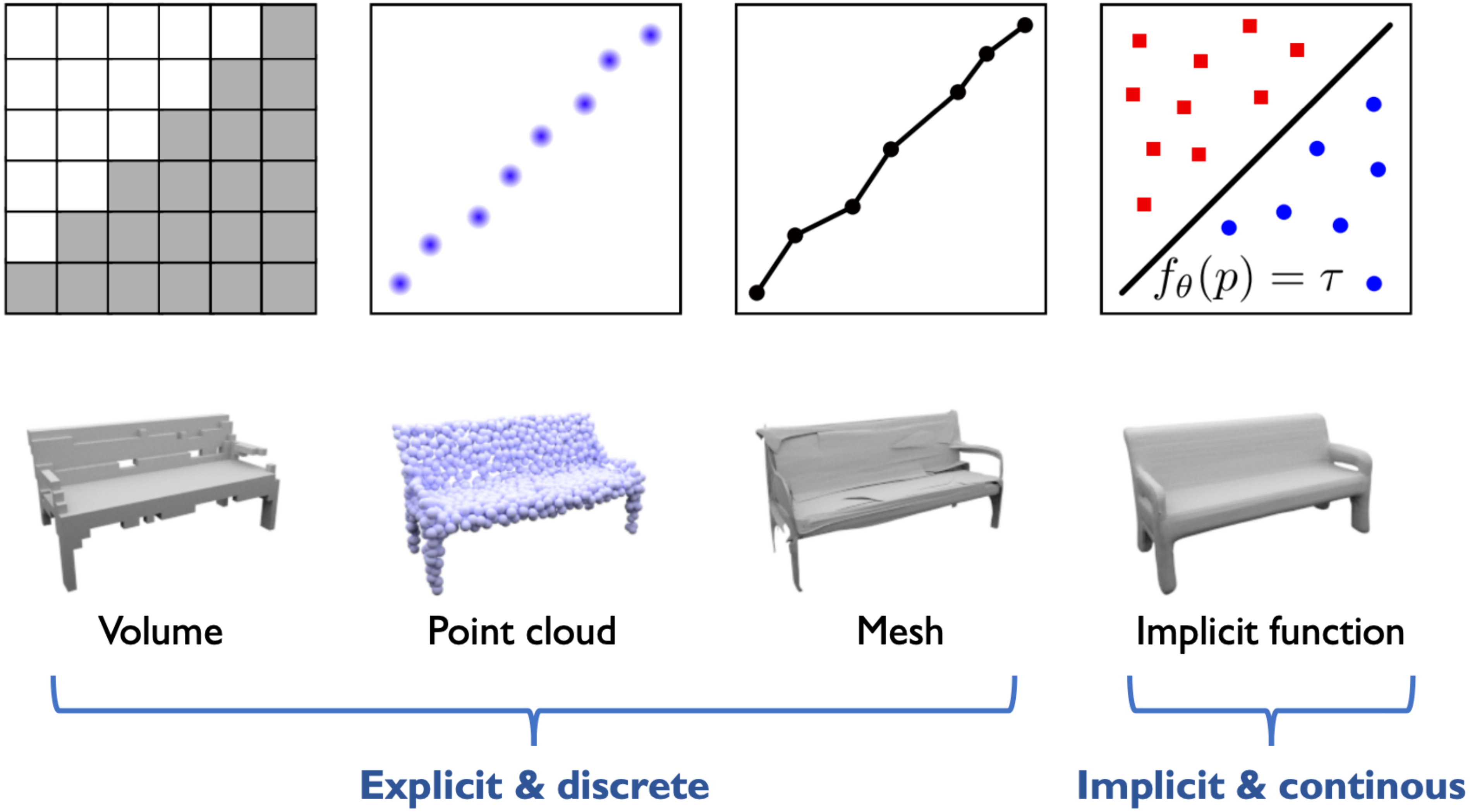
Learning Dynamic Renderable Volumes from Images. SIGGRAPH 2019.

Common 3D Representation 2: Volume-based Representations

- Advantages
  - Capable of representing highly complex scenes
  - Fast rendering speed
- Disadvantages
  - Prone to occupying large amounts of video memory
  - Unable to represent high-resolution scenes



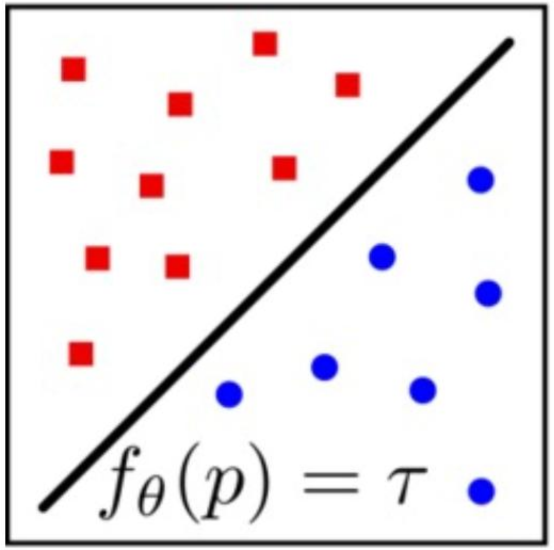
Common 3D Representation 3: Implicit Representations



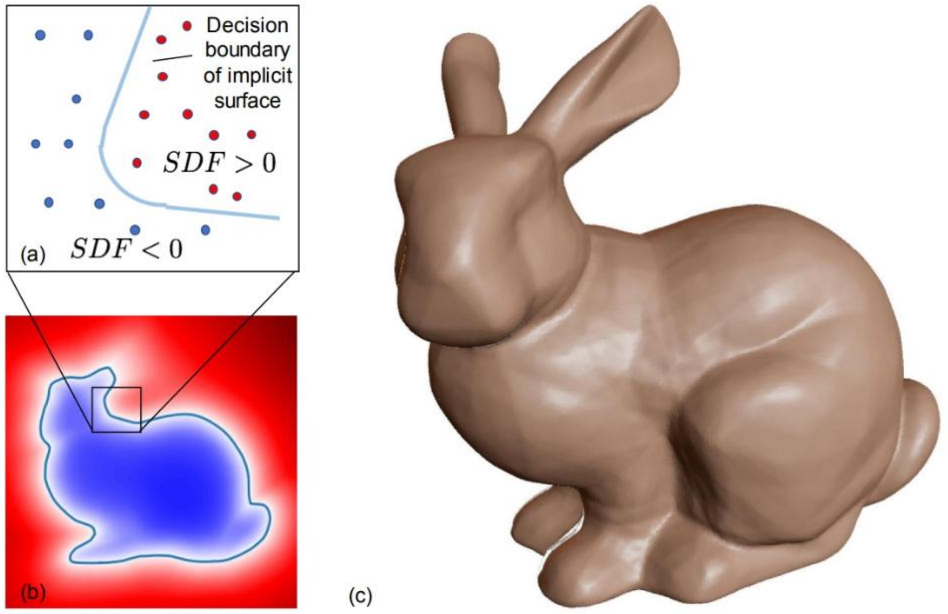
Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019.

Common 3D Representation 3：Implicit Representations

Occpupacy



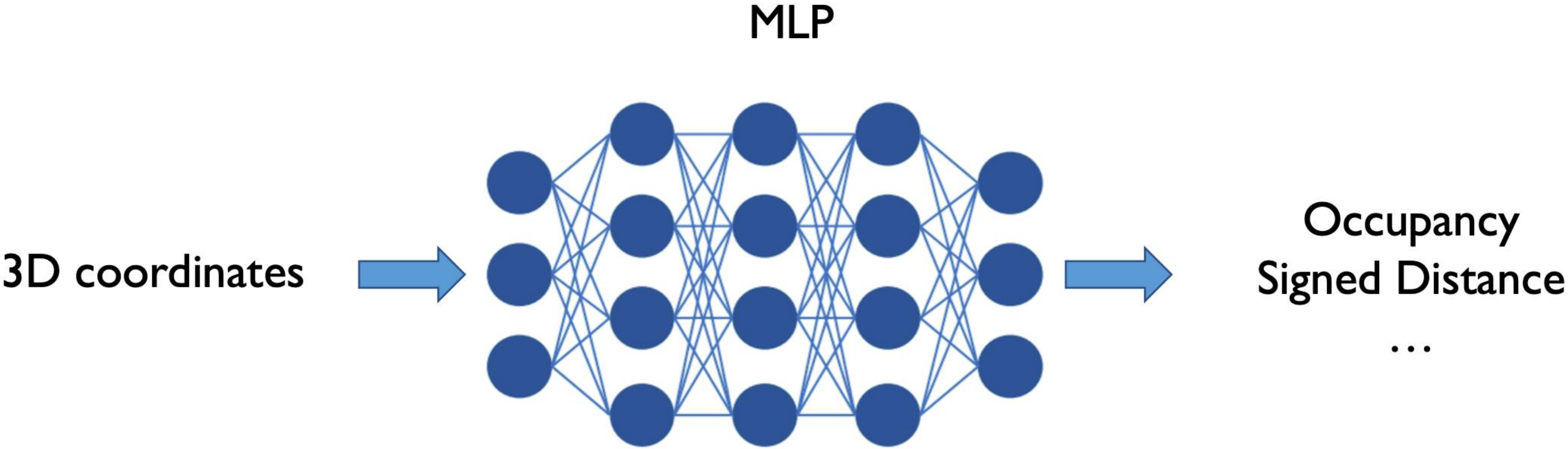
Signed distance function (SDF)



Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019.

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 2019.

Implicit Neural Representations

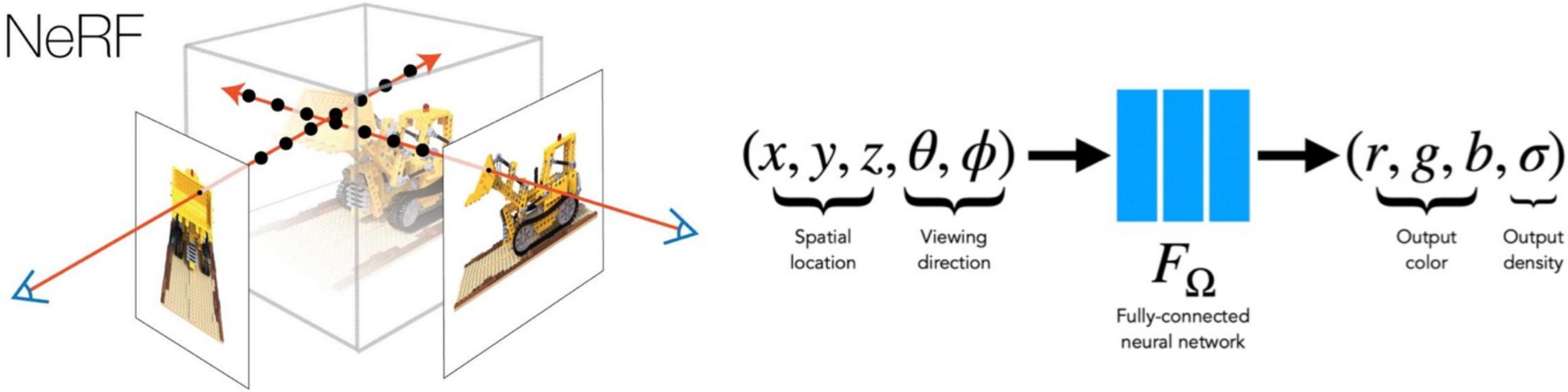


Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019.

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 2019.

Neural Radiance Fields (NeRF)

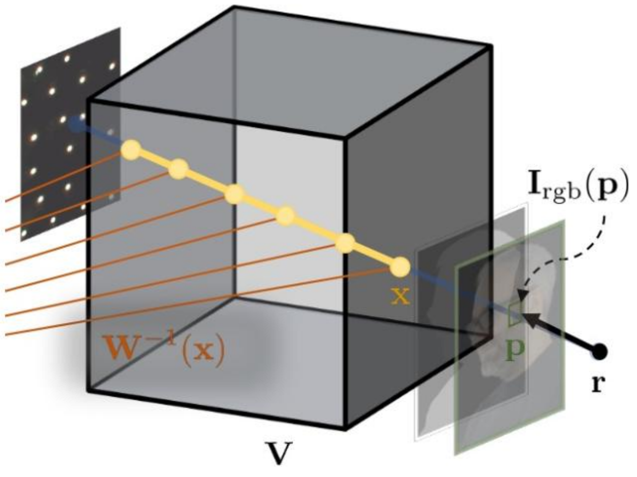
- Represent the scene as a continuous voxel density field and color field.



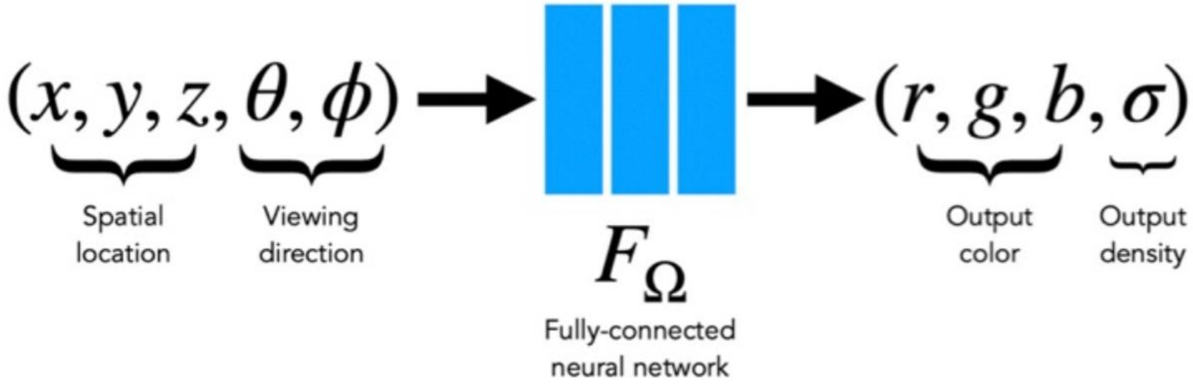
NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

Neural Radiance Fields (NeRF)

- Comparison between discrete representation and continuous representation.



Discrete RGB-α volume



Continuous RGB-α field

Neural Volumes: Learning Dynamic Renderable Volumes from Images, SIGGRAPH 2019.

NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

The Rendering Process of NeRF：Volume Rendering

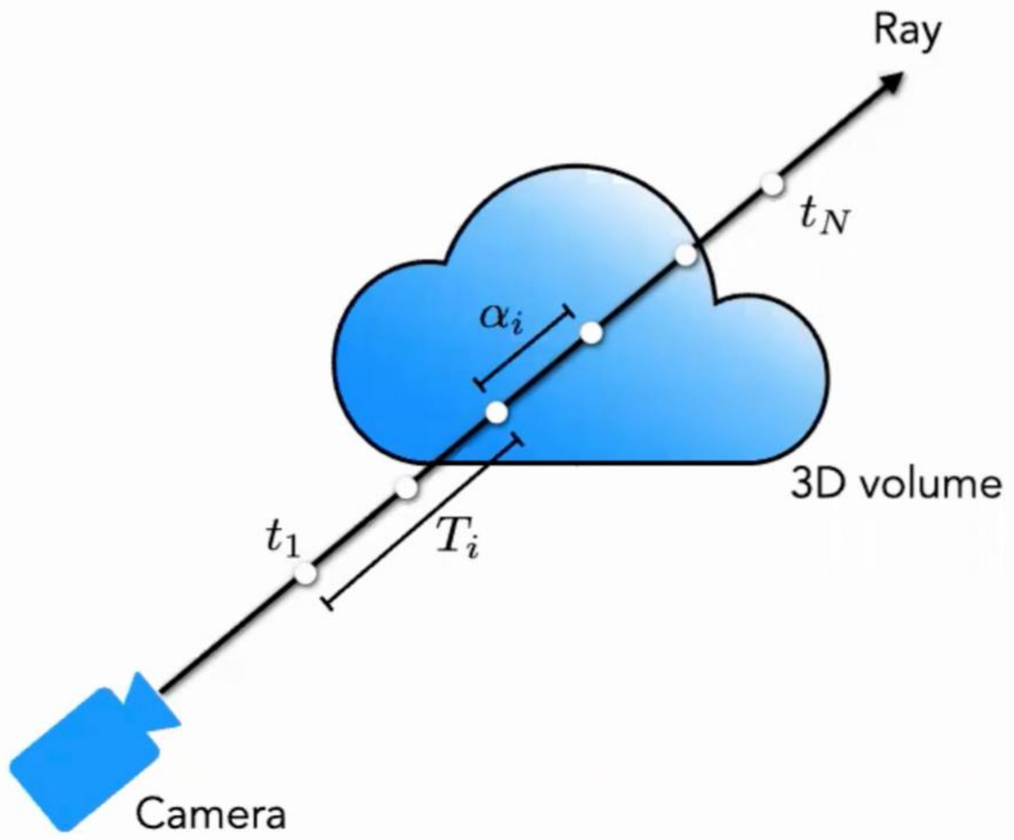
Rendering model for ray r(t) = o + td:

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

weights colors

How much light is blocked earlier along ray:

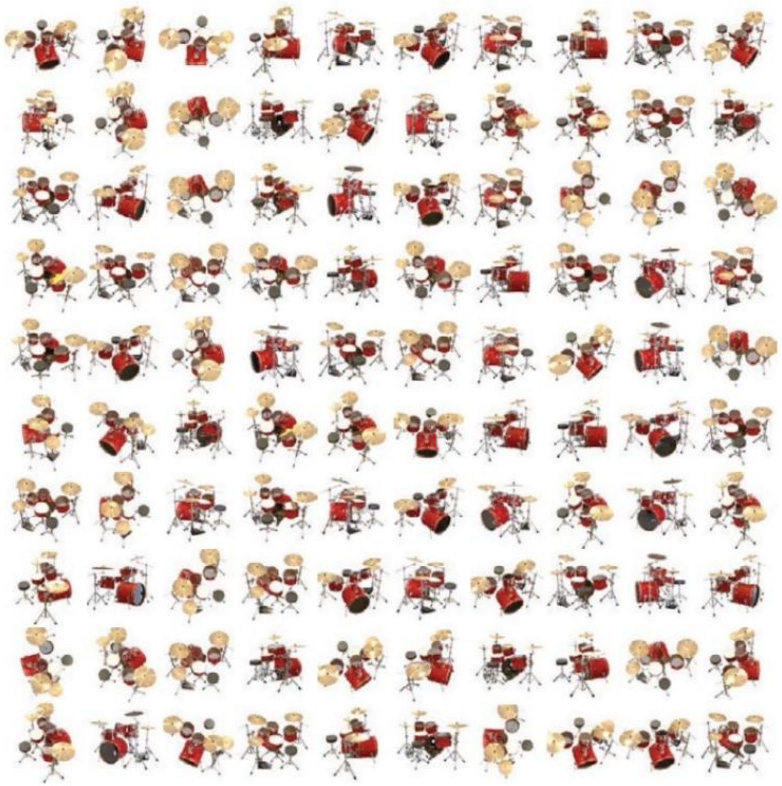
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$



NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

The Training Process of NeRF

- Optimizing NeRF Based on Multi-View Images.

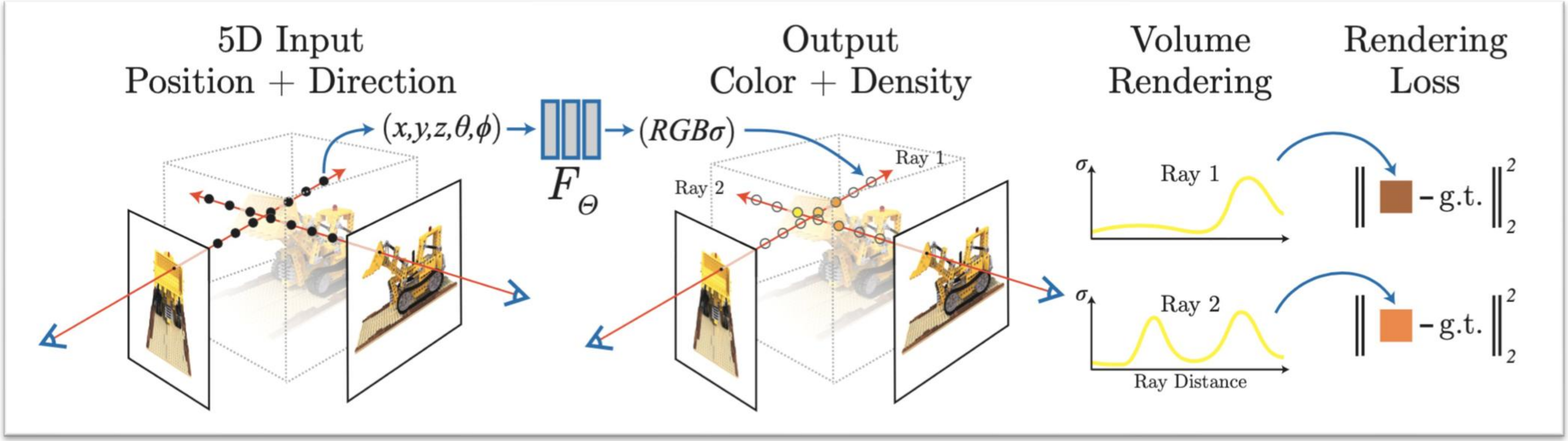


Input multi-view images

Optimizing NeRF

NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

The Training Process of NeRF

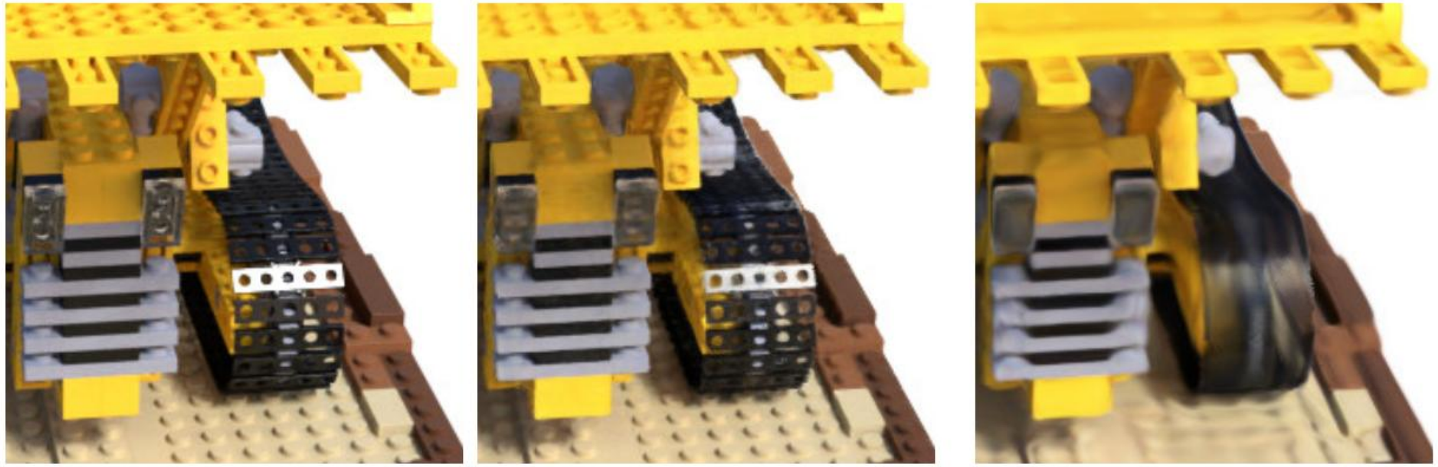


NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

The Key Innovation of NeRF: Positional Encoding

- Map input coordinates to high-dimensional vectors

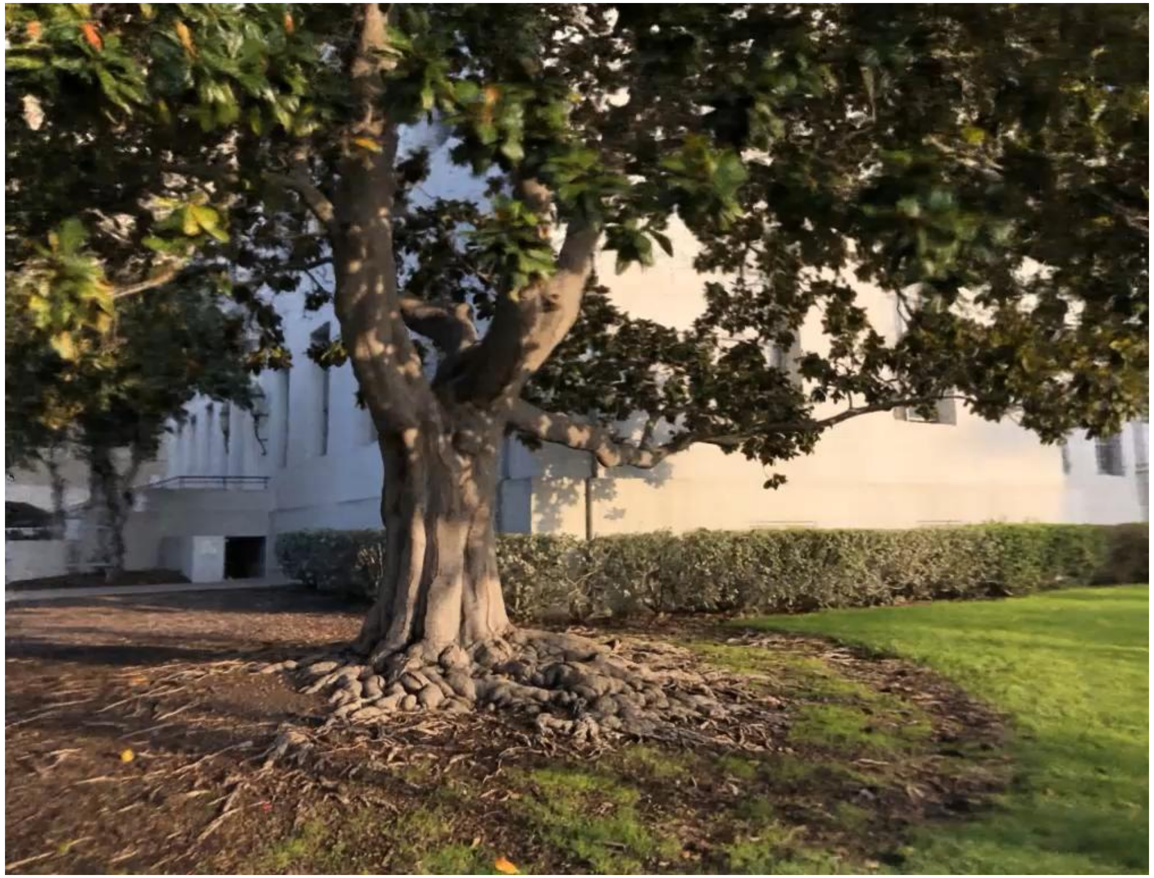
$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \cdots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right) \right)$$



Ground Truth　　Complete Model　　No Positional Encoding

NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

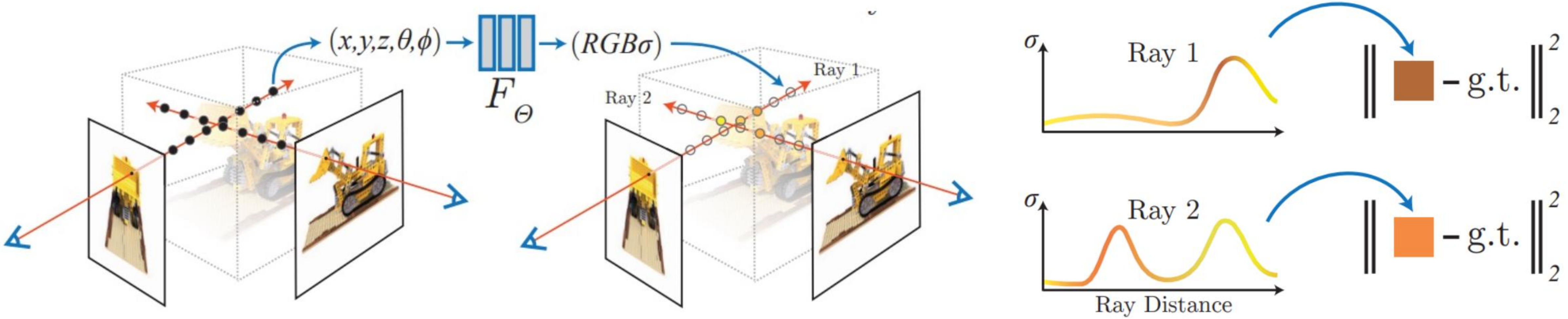
The Demo of NeRF



NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

Why NeRF is better?

- MLP networks can represent continuous high-resolution scenes

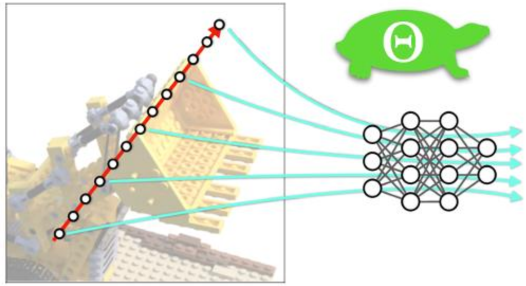


NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

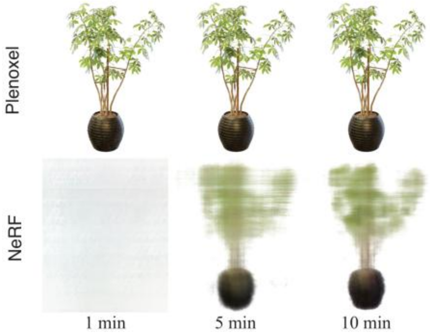
Problems faced by NeRF

The challenges faced by NeRF in static scene modeling

Slow rendering speed

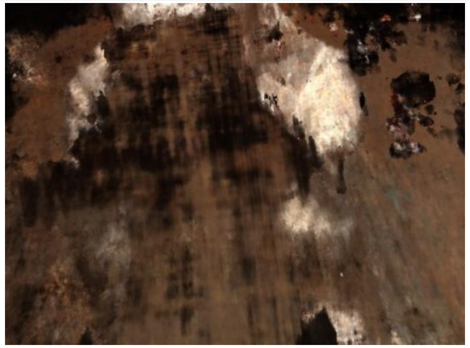


Slow training speed



Weak modeling capability

Poor model robustness



The challenges faced by NeRF in other types of scene modeling

Low geometric reconstruction quality

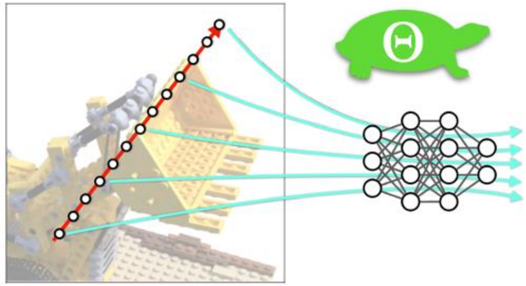
Unable to model multimodal signals

Unable to model dynamic scenes

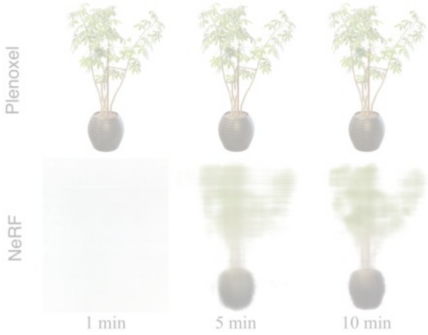
Problem 1 of NeRF: Slow rendering speed

The challenges faced by NeRF in static scene modeling

Slow rendering speed

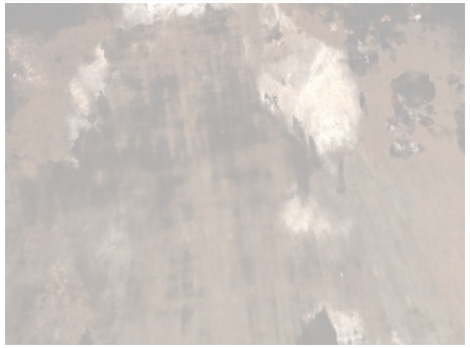


Slow training speed



Weak modeling capability

Poor model robustness



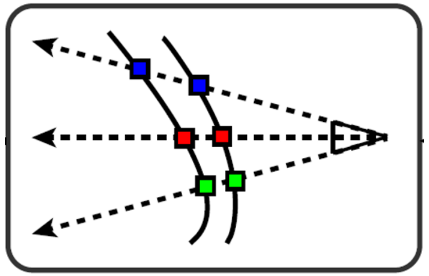
The challenges faced by NeRF in other types of scene modeling

Low geometric reconstruction quality

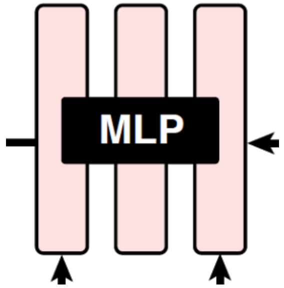
Unable to model multimodal signals

Unable to model dynamic scenes

Analysis of NeRF Rendering Costs



*

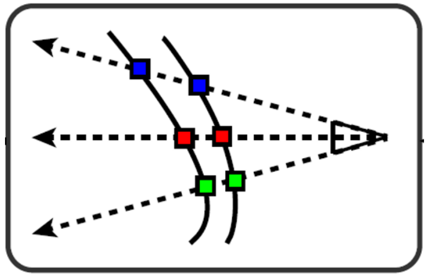


=

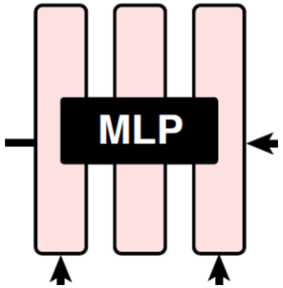
The number of sampled points * Point-level cost calculation = Rendering cost

Ideas for Reducing NeRF Rendering Costs

1. Reducing the Number of Sampling Points：NSVF、AdaNeRF、ENeRF
2. Reducing the Per-Point Computation Cost：SNeRG、PlenOctree、KiloNeRF、ObjectNeRF



*



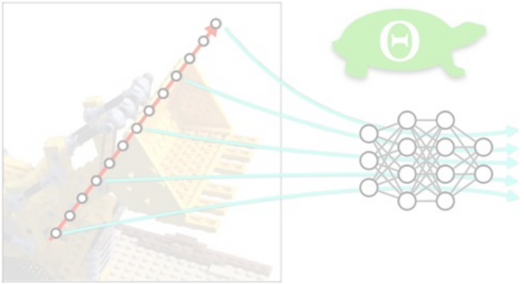
=

The number of sampled points * Point-level cost calculation = Rendering cost

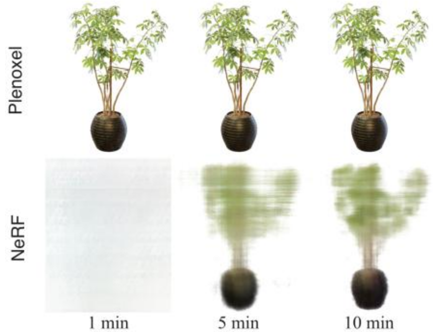
Problem 2 of NeRF: Slow training speed

The challenges faced by NeRF in static scene modeling

Slow rendering speed

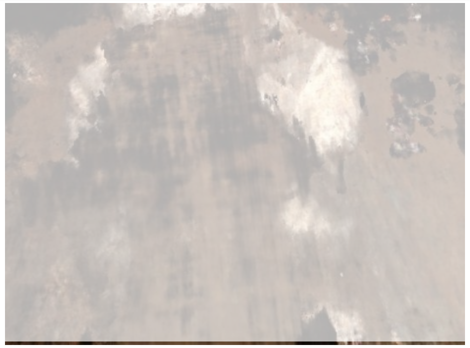


Slow training speed



Weak modeling capability

Poor model robustness



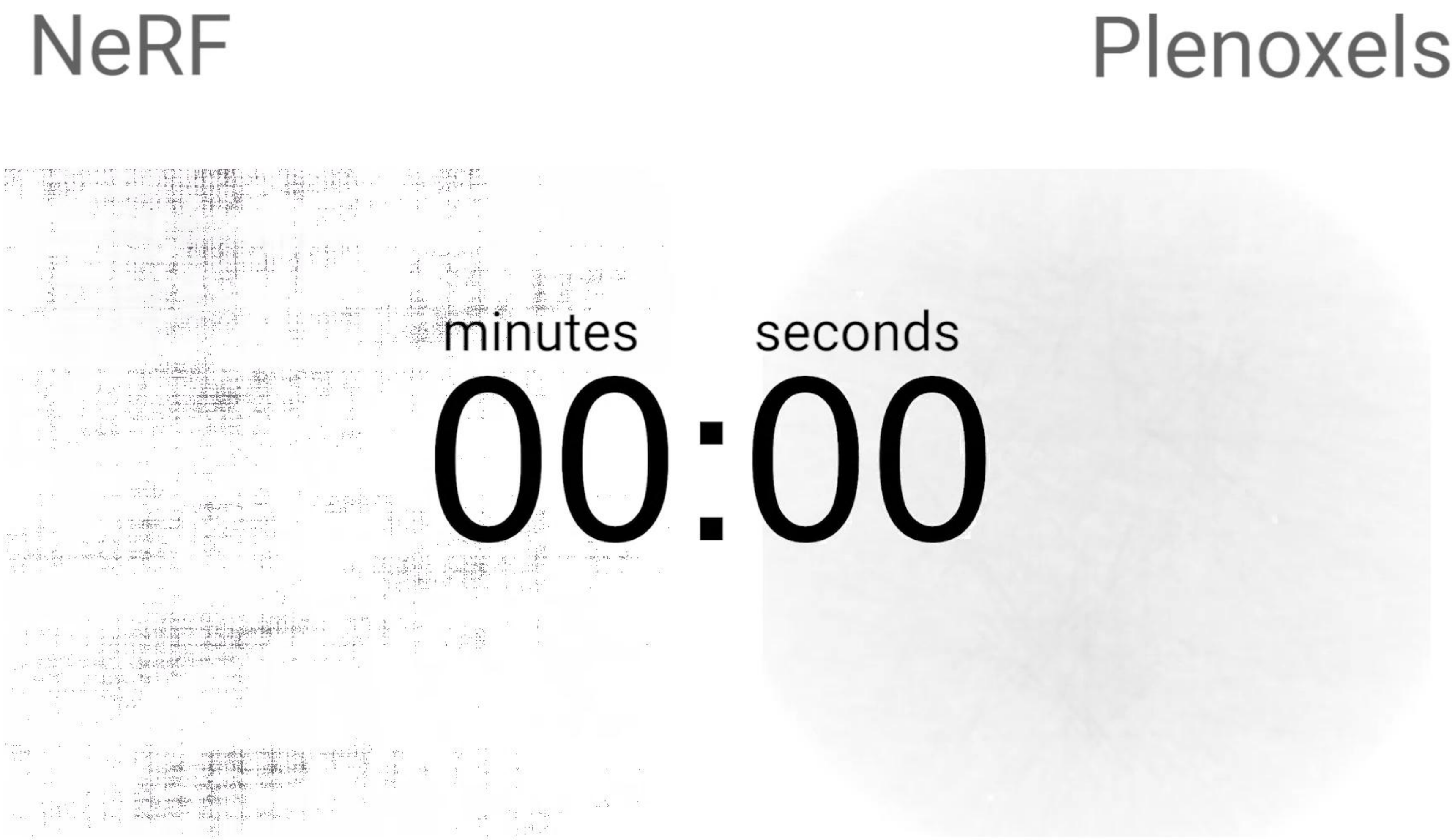
The challenges faced by NeRF in other types of scene modeling

Low geometric reconstruction quality

Unable to model multimodal signals

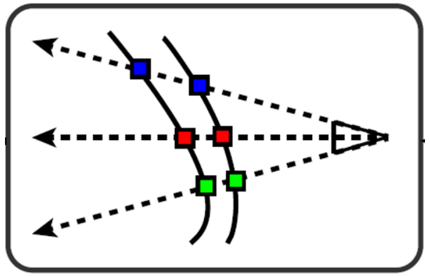
Unable to model dynamic scenes

NeRF's slow training speed

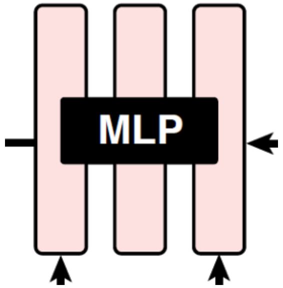


Plenoxels: Radiance Fields without Neural Networks. CVPR 2022.

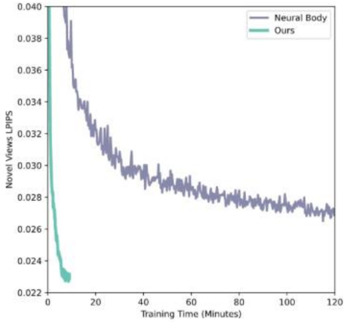
Analysis of NeRF's Training Cost



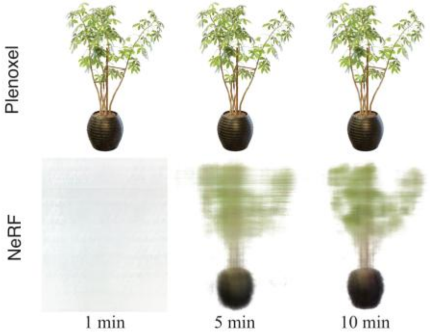
*



*



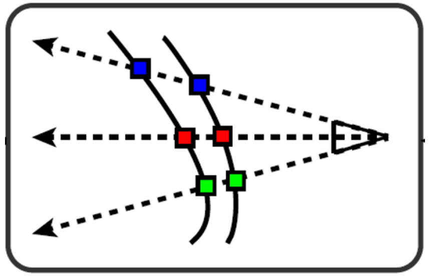
=



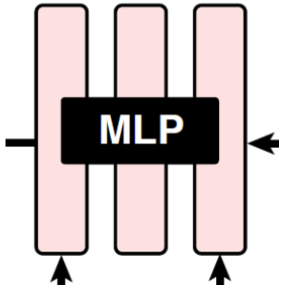
Number of 3D Points * Per-point Training Cost * Number of Iterations = Training Cost

Ideas for Reducing NeRF's Training Cost

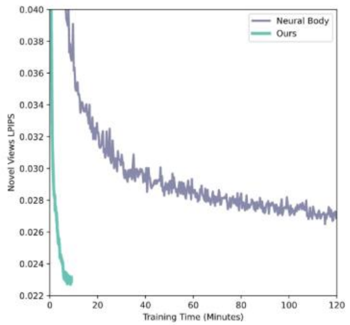
1. Reducing the number of 3D points：Plenoxel
2. Reducing per-point training cost：Instant NGP、TensoRF
3. Reducing the number of iterations：IBRNet



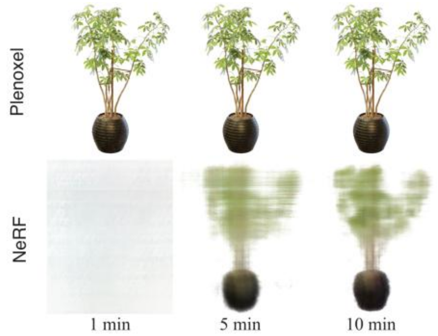
*



*



=



Number of 3D Points * Per-point Training Cost * Number of Iterations = Training Cost

Problem 3 of NeRF: Weak modeling capability

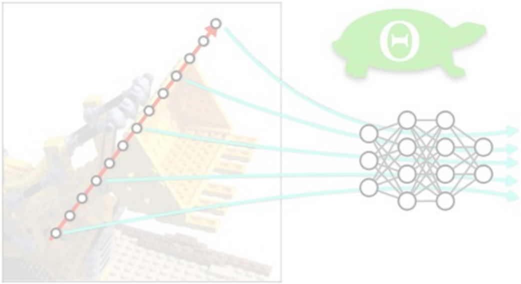
The challenges faced by NeRF in static scene modeling

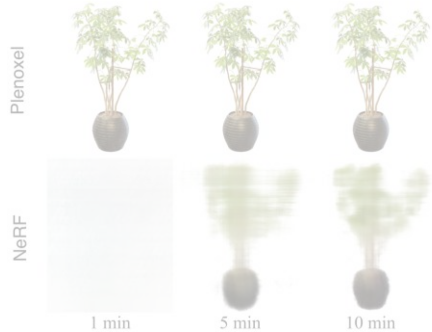
Slow rendering speed

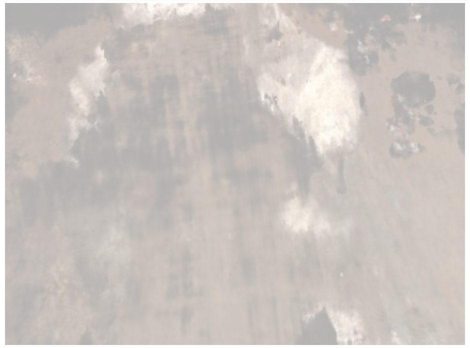
Slow training speed

Weak modeling capability

Poor model robustness







The challenges faced by NeRF in other types of scene modeling

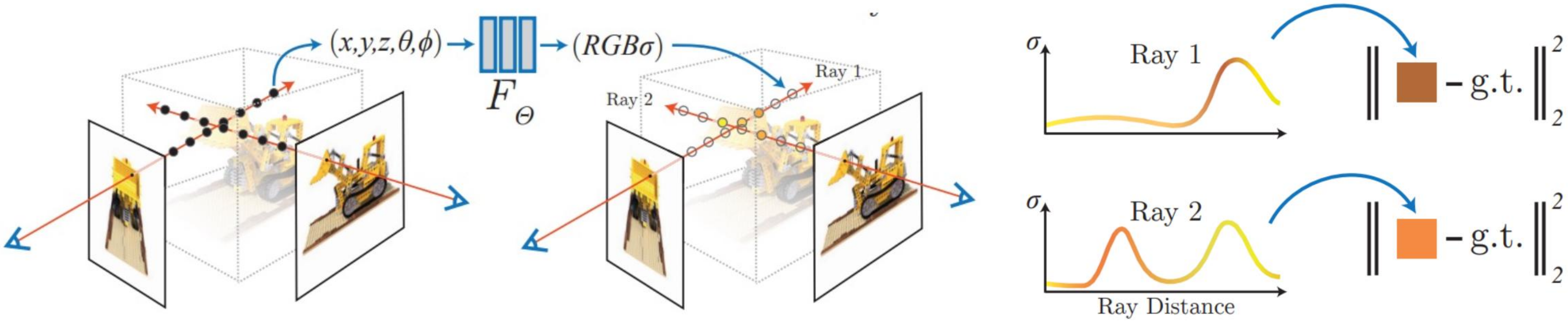
Low geometric reconstruction quality

Unable to model multimodal signals

Unable to model dynamic scenes

Reasons Why NeRF Fails to Model Large Scenes

- The capability of NeRF's MLP network is limited.
- NeRF samples 3D points along camera rays, thus failing to handle unbounded scenes.

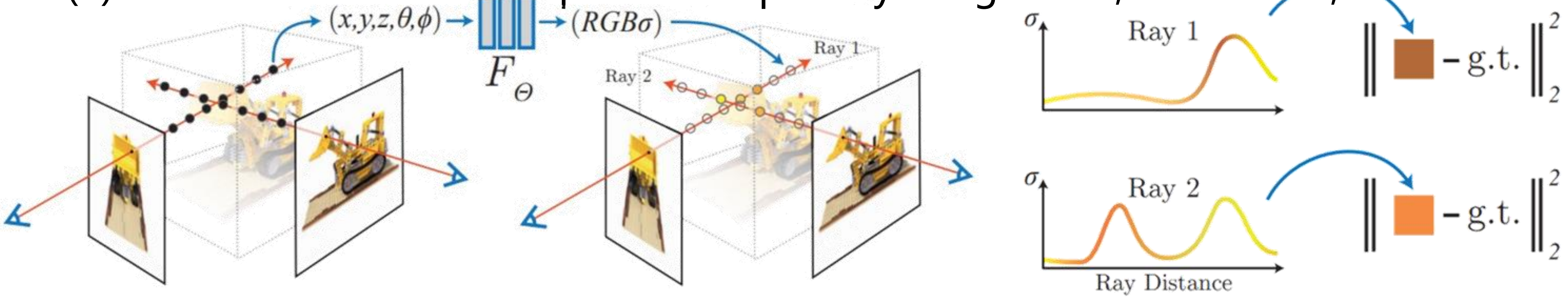


NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

Ideas for Enhancing NeRF's Large-Scene Modeling Capability

- The capability of NeRF's MLP network is limited.
- NeRF samples 3D points along camera rays, thus failing to handle unbounded scenes.

- Solutions:
- (1) Design a new scene representation: NeRF++;
- (2) Enhance the model's expressive capability: MegaNeRF, BlockNeRF, Grid-NeRF.

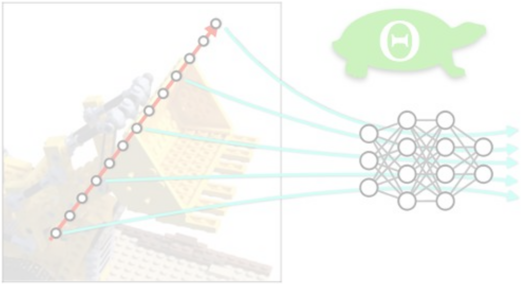


NeRF: Representing scenes as neural radiance fields for view synthesis, ECCV 2020.

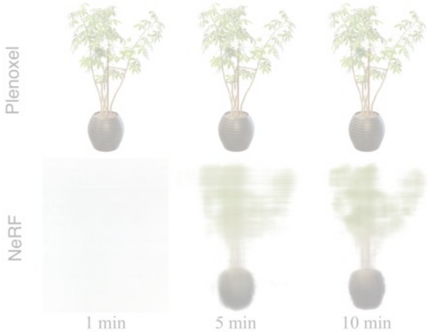
Problem 4 of NeRF: Poor model robustness

The challenges faced by NeRF in static scene modeling

Slow rendering speed

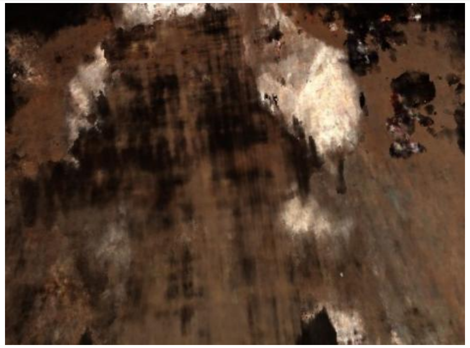


Slow training speed



Weak modeling capability

Poor model robustness



The challenges faced by NeRF in other types of scene modeling

Low geometric reconstruction quality

Unable to model multimodal signals

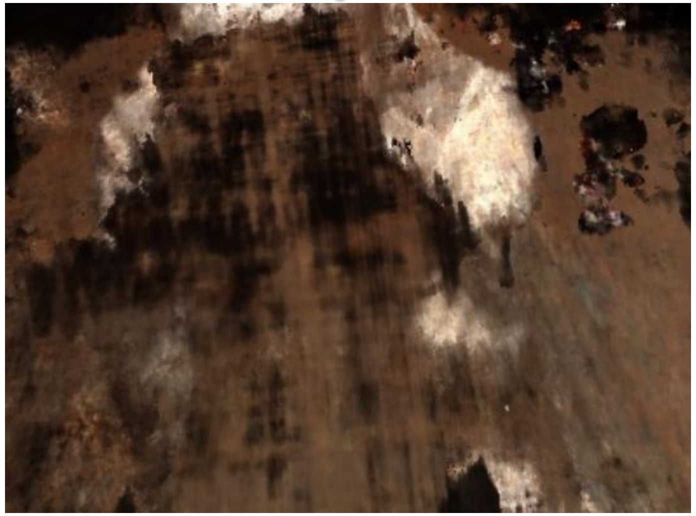
Unable to model dynamic scenes

Manifestations of NeRF's Model Non-Robustness

Inaccurate Camera Poses

BARF

Sparse Input Viewpoints



Reg-NeRF, SinNeRF

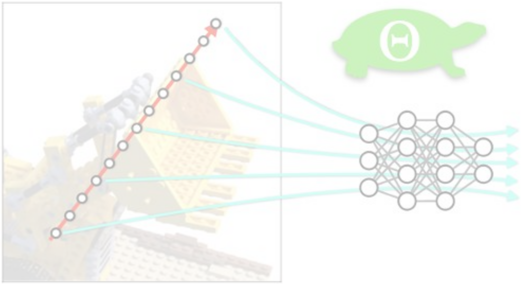
Illumination Changes

NeRF in the Wild

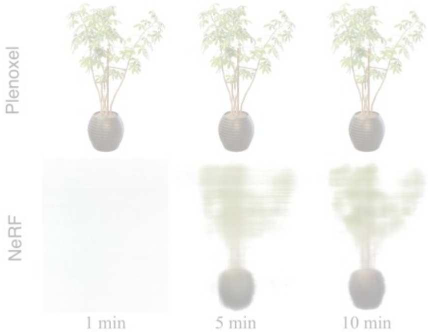
Problem 5 of NeRF: Low geometric reconstruction quality

The challenges faced by NeRF in static scene modeling

Slow rendering speed

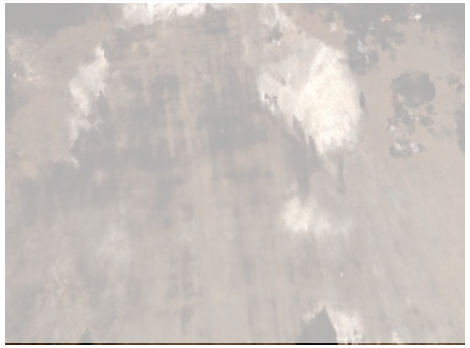


Slow training speed



Weak modeling capability

Poor model robustness



The challenges faced by NeRF in other types of scene modeling

Low geometric reconstruction quality

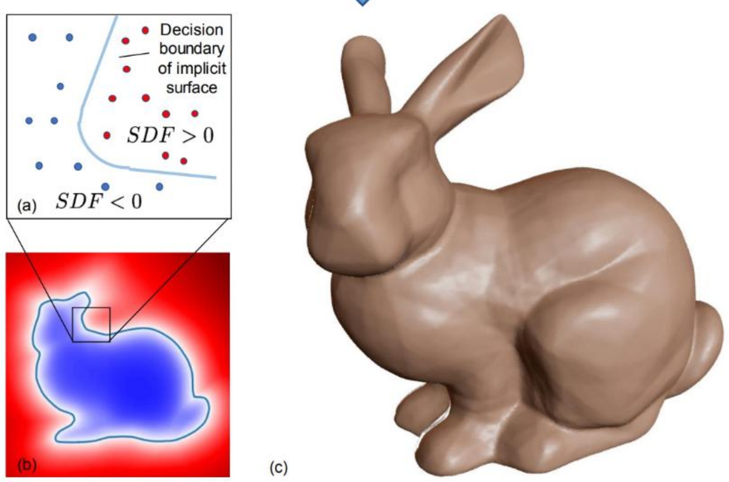
Unable to model multimodal signals

Unable to model dynamic scenes

Reasons for the poor quality of NeRF's geometric reconstruction

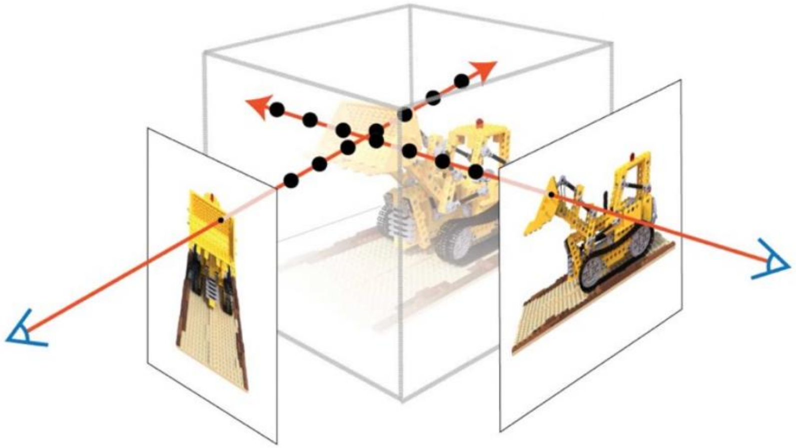
- NeRF lacks the definition of surface geometry.

(x, y, z)



Signed distance

(x, y, z)



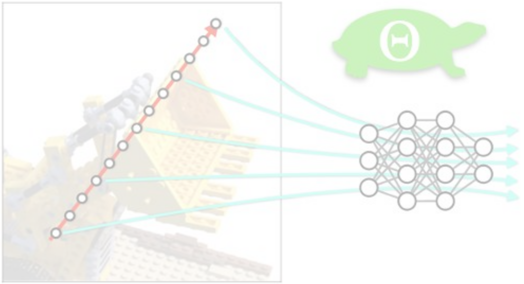
RGB + density

NeuS

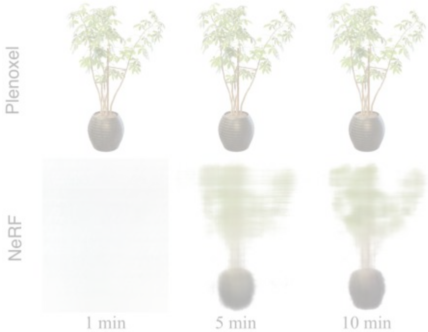
Problem 6 of NeRF: Unable to model multimodal signals

The challenges faced by NeRF in static scene modeling

Slow rendering speed

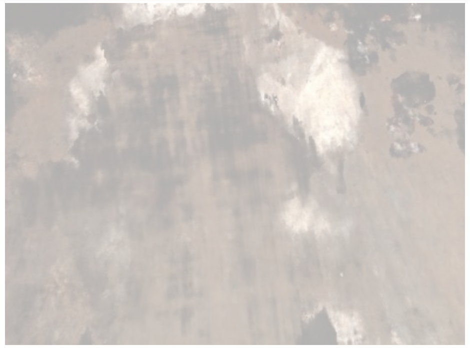


Slow training speed



Weak modeling capability

Poor model robustness



The challenges faced by NeRF in other types of scene modeling

Low geometric reconstruction quality

Unable to model multimodal signals

Unable to model dynamic scenes

NeRF lacks multimodal signal modeling

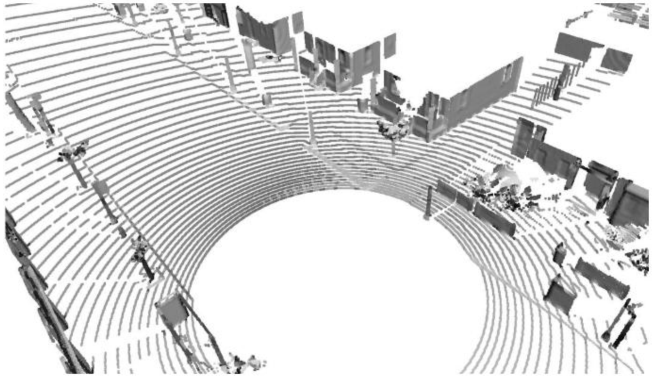
Semantic Field

Semantic NeRF

Object Material

InvRender

LiDAR Point Cloud

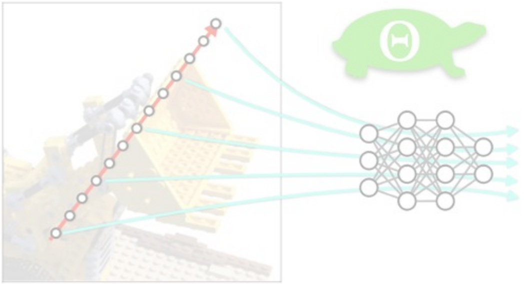


LiDAR-NeRF

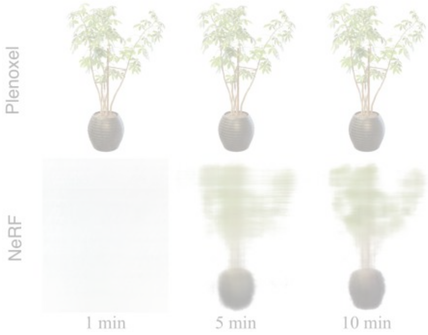
Problem 7 of NeRF: Unable to model dynamic scenes

The challenges faced by NeRF in static scene modeling

Slow rendering speed

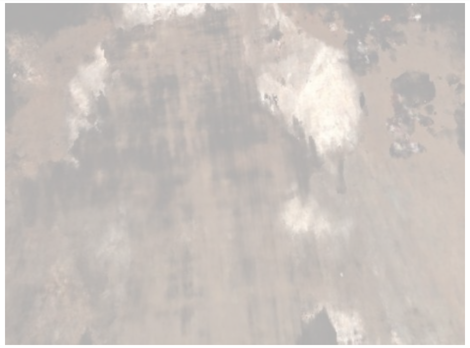


Slow training speed



Weak modeling capability

Poor model robustness



The challenges faced by NeRF in other types of scene modeling

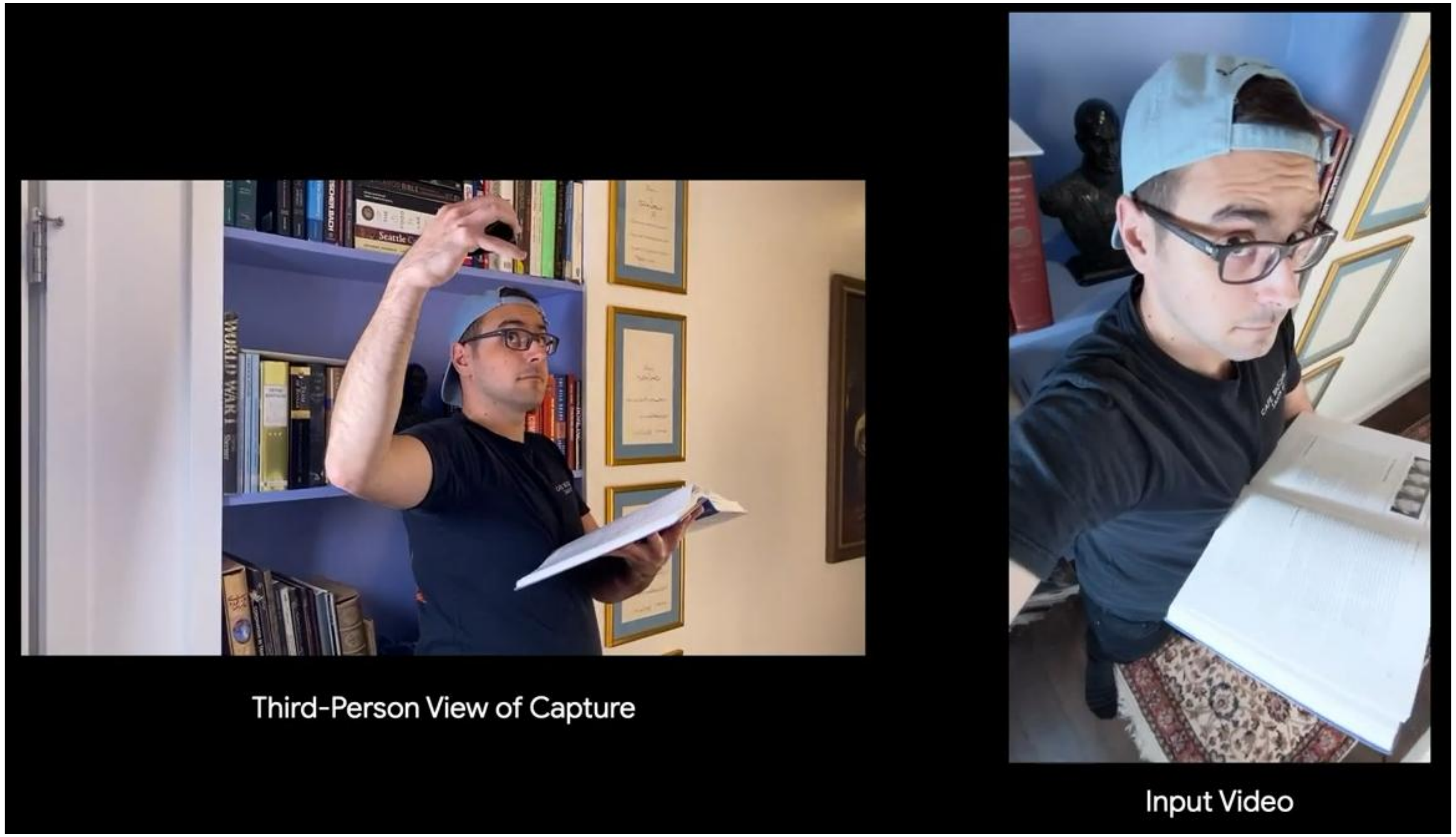
Low geometric reconstruction quality

Unable to model multimodal signals

Unable to model dynamic scenes

NeRF is unable to model dynamic scenes

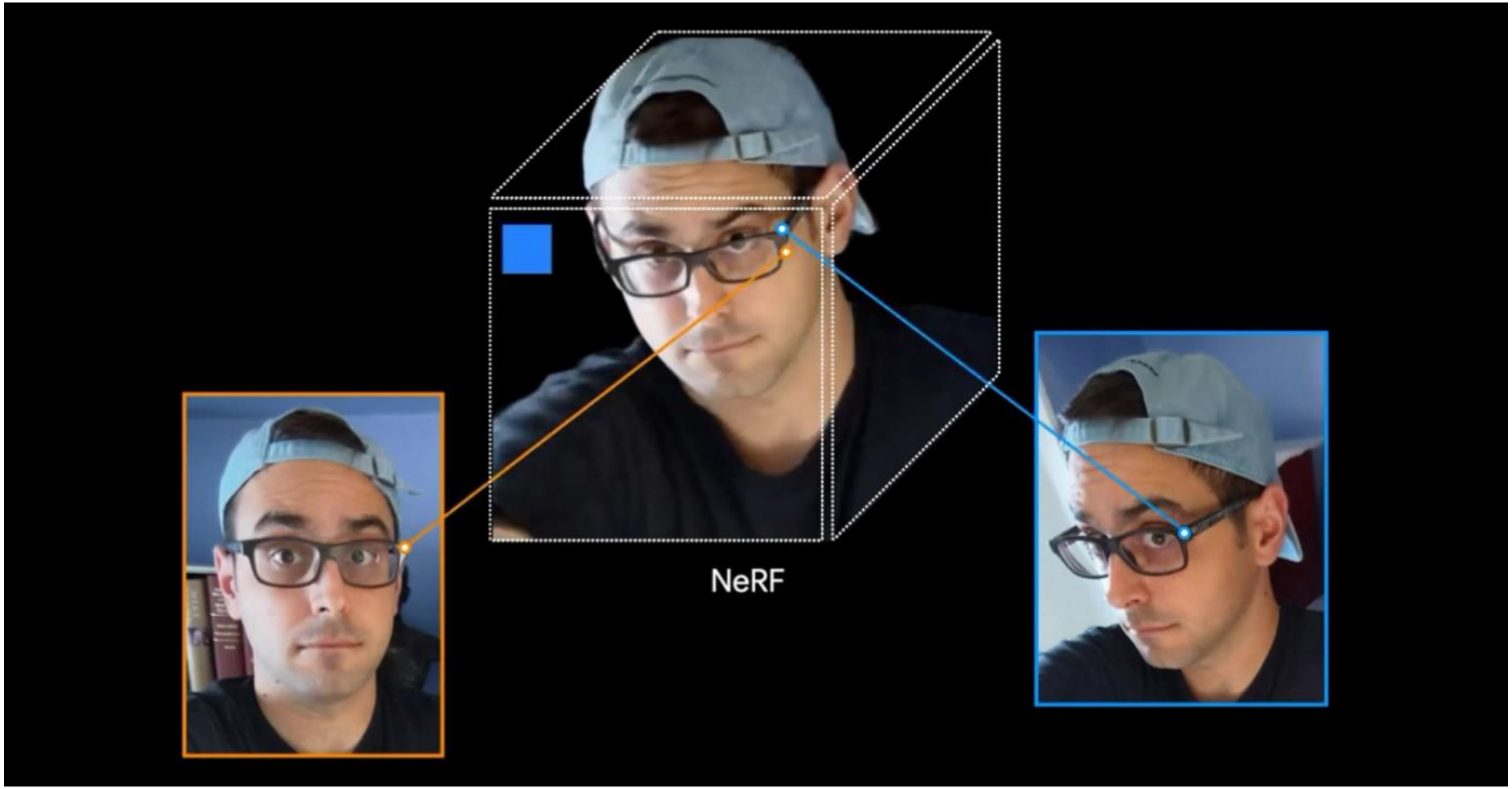
- NeRF cannot obtain reasonable reconstruction results from dynamic scenes.



Nerfies: Deformable neural radiance fields. ICCV 2021.

NeRF is unable to model dynamic scenes

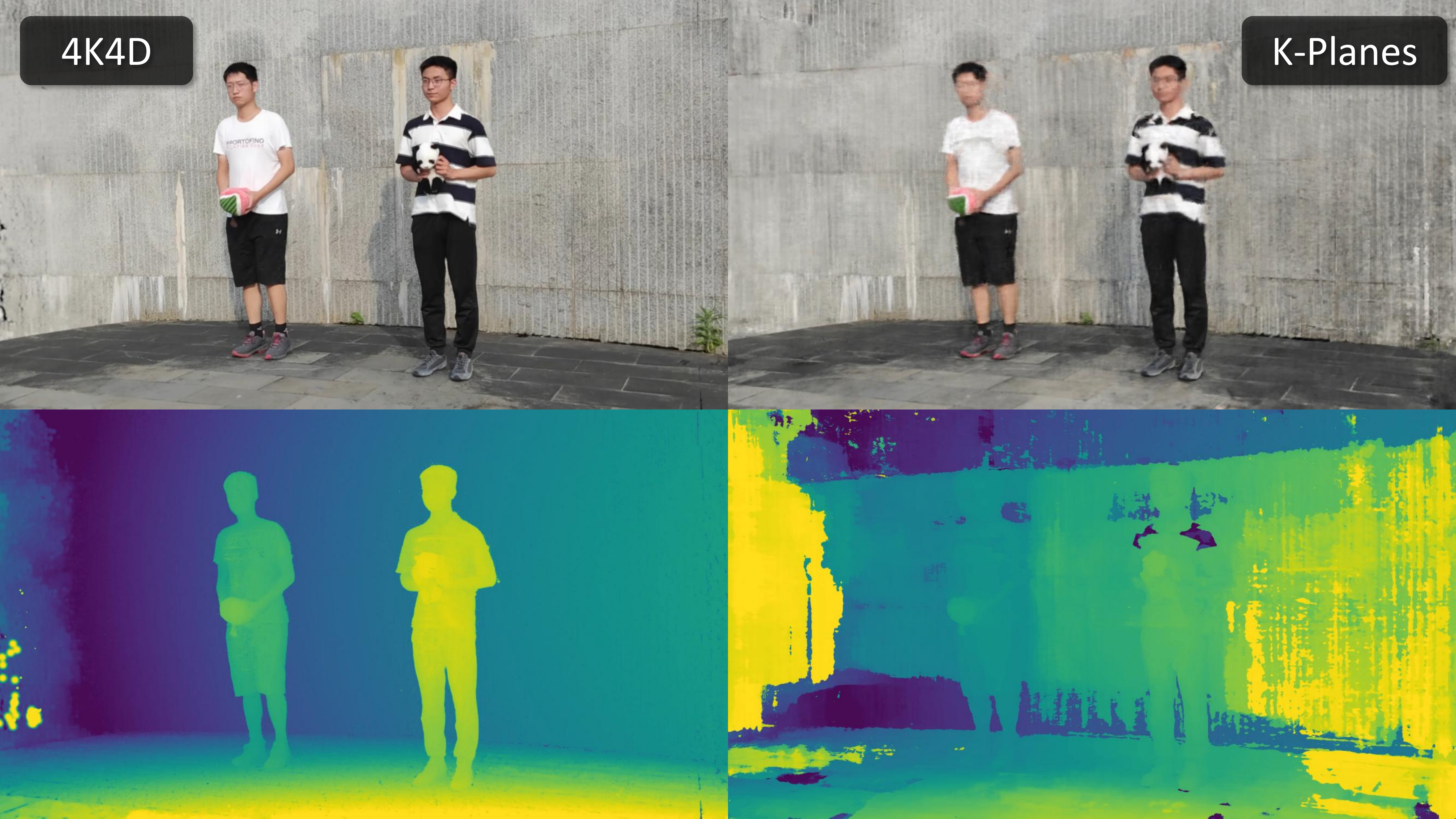
- Object movement in dynamic scenes prevents multi-view matching.



Nerfies: Deformable neural radiance fields. ICCV 2021.

Performance of Deformable NeRF

Nerfies: Deformable neural radiance fields. ICCV 2021.

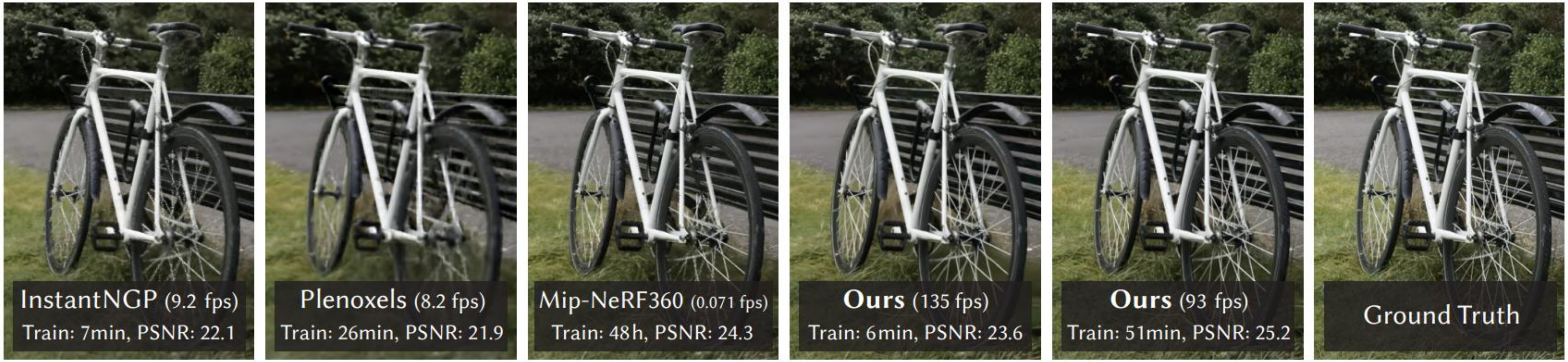

4K4D
K-Planes

Outline

- Neural Radiance Fields (NeRF)
- 3D Gaussian Splatting (3DGS)

3DGS

3DGS Modeling: Explicitly modeling the scene as attribute-differentiable 3D Gaussian ellipsoids



- Ideal NVS method:
  - High quality (material & geometry)
  - Efficient (training & rendering & memory)
  - User-friendly (easy to reproduce & good compatibility)

- 3DGS:
  - High quality (compared with Mip-NeRF360)
  - Efficient training (compared with Instant-NGP)
  - 100+ FPS real-time rendering (A6000 GPU)
  - Good compatibility with traditional rasterization pipelines

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Modeling

Ecstatica(1994)

- **Ellipsoid Modeling**: Explicitly model 3D scenes using 3D ellipsoids.
- **Differentiability**: 3D ellipsoids are represented by 3D Gaussians, which possess properties for expressing geometry/materials. These properties can be adjusted through differentiable rendering.
- **Rasterization**: 3D Gaussian ellipsoids enable convenient and efficient rasterization.

3DGS Modeling：Properties

point_cloud.ply

E: > 3DGS > Projects > gaussian-splatting > data > 360 > bonsai >

```
ply
format binary_little_endian 1.0
element vertex 1164227
property float x
property float y
property float z
property float nx
property float ny
property float nz
property float f_dc_0
property float f_dc_1
property float f_dc_2
property float f_rest_0
property float f_rest_1
property float f_rest_2
```

...

```
property float f_rest_42
property float f_rest_43
property float f_rest_44
property float opacity
property float scale_0
property float scale_1
property float scale_2
property float rot_0
property float rot_1
property float rot_2
property float rot_3
end_header
```

3DGS Modeling Result Storage

- Gaussian Properties (59 dimensions total)：
  - xyz： 3D geometric center of the 3D Gaussian ellipsoid, 3 dimensions in total (nxyz is unassigned and invalid in the storage file)
  - dc&rest： Spherical Harmonics (SH) for representing anisotropic colors, 48 dimensions in total (i.e., 3rd-order SH, $3 \times (3+1)^2$
  - opacity： Opacity of the Gaussian ellipsoid, 1 dimension
  - scale： Scale vector of the Gaussian ellipsoid, 3 dimensions
  - rot： Rotation quaternion of the Gaussian ellipsoid, 4 dimensions

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Modeling：Properties

Scale

Rotate

Opacity

Unit Gaussian sphere

Gaussian ellipsoid

Transparent Gaussian ellipsoid

- A 3D ellipsoid is represented based on a 3D Gaussian distribution：

$$G(x) = e^{-\frac{1}{2}(x)^T\Sigma^{-1}(x)}$$

- The volume and pose of a 3D ellipsoid are determined by its covariance matrix, which consists of two components: scaling and rotation：

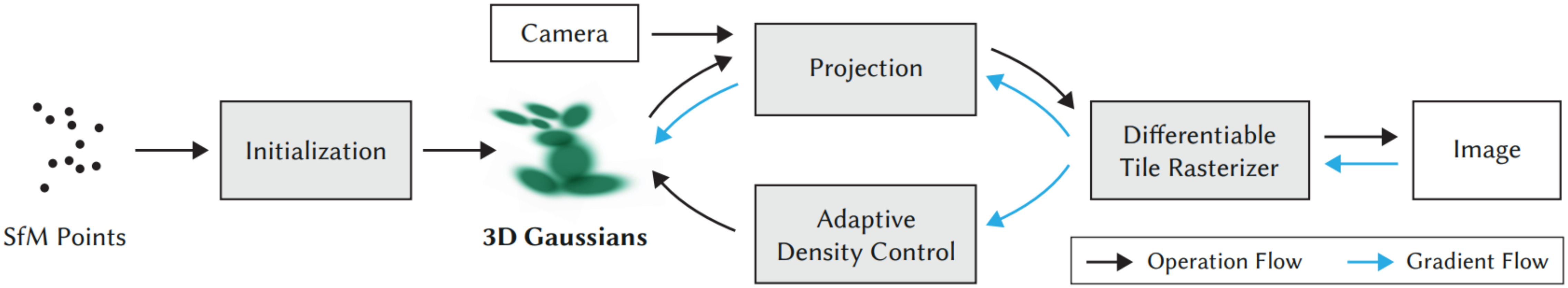
$$\Sigma = RSS^TR^T$$

- Apply the Gaussian distribution to splatting computation to obtain the splat opacity α, followed by alpha blending：

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j),$$

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

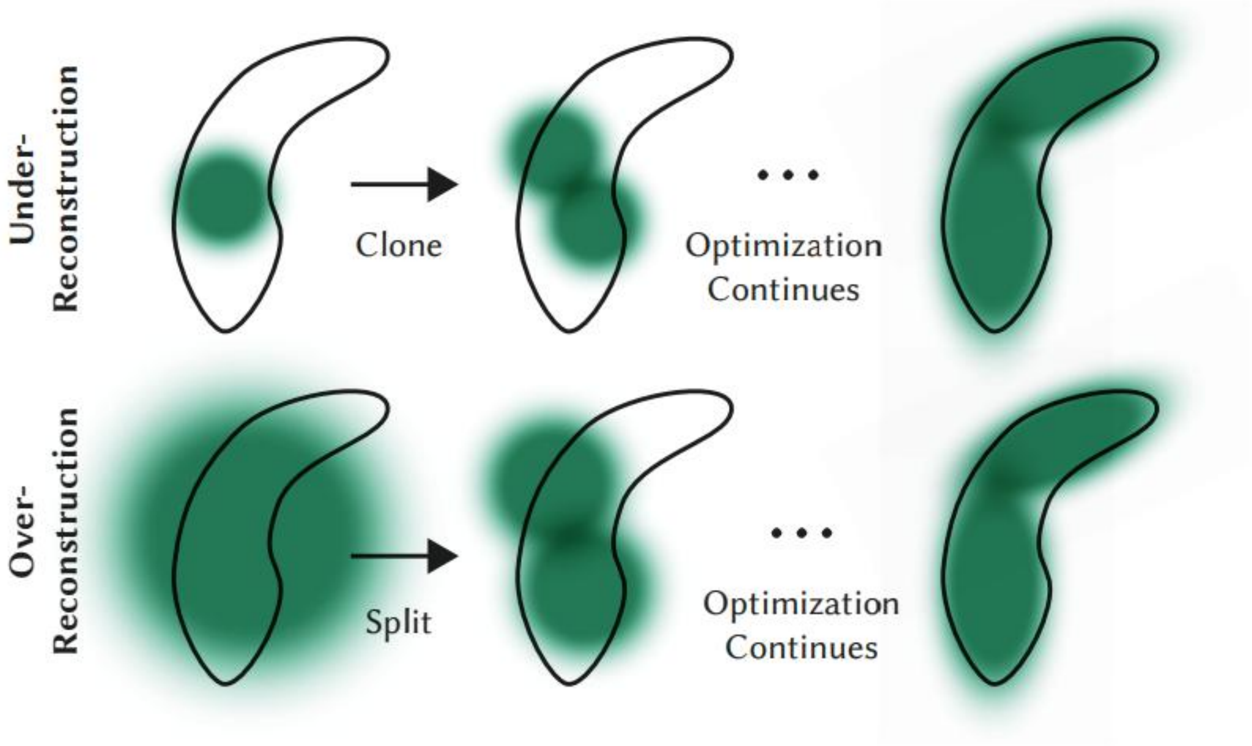
3DGS Modeling：Pipeline



- Gaussian Ellipsoid Modeling Pipeline (Input: multi-view images; Output: 3D Gaussian ellipsoid modeling results):
  - SFM Initialization: Compute a sparse point cloud from multi-view images using SFM to initialize 3D Gaussians.
  - Gaussian Rendering and Optimization: Project 3D Gaussians into screen space for rasterization, compute the loss, and perform backpropagation to optimize Gaussian properties.
  - Adaptation and Adjustment: Duplicate/split Gaussians based on gradients and scaling to achieve better fitting of scene objects.

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Modeling：Densification

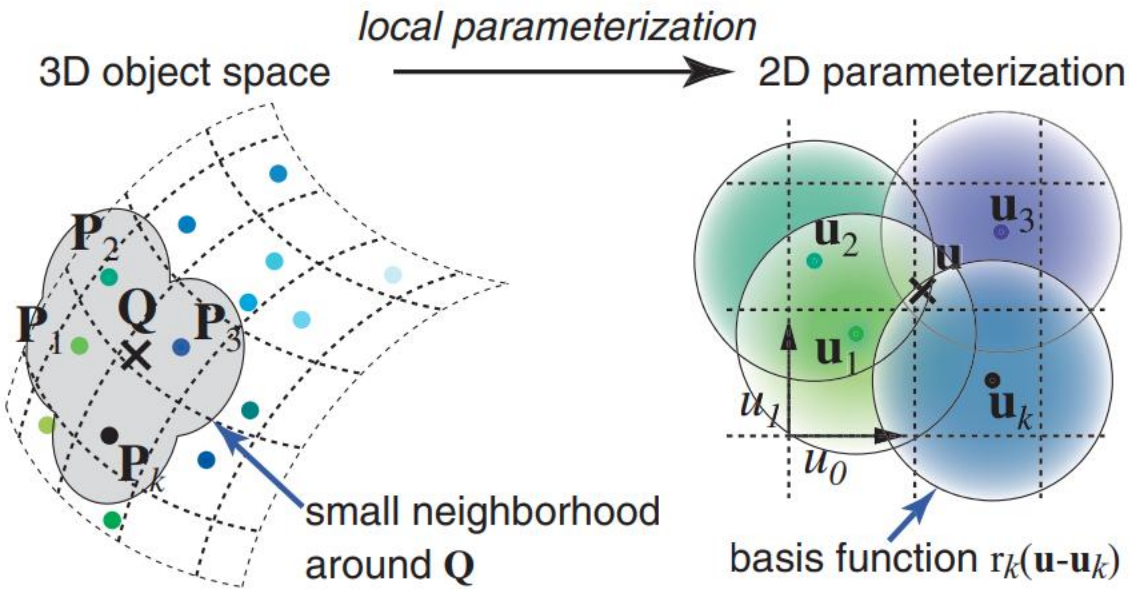


- Adaptive Density Control：
  1. Identify Gaussians with large gradients in their central position properties (i.e., Gaussians whose current position properties are insufficiently accurate) and perform adaptive densification based on Gaussian size：
     - Duplicate small Gaussians
     - Split large Gaussians
  2. After Gaussian duplication & splitting, prune Gaussians with excessively large radii or excessively small opacities.

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Rendering: Point Based Rendering Formula

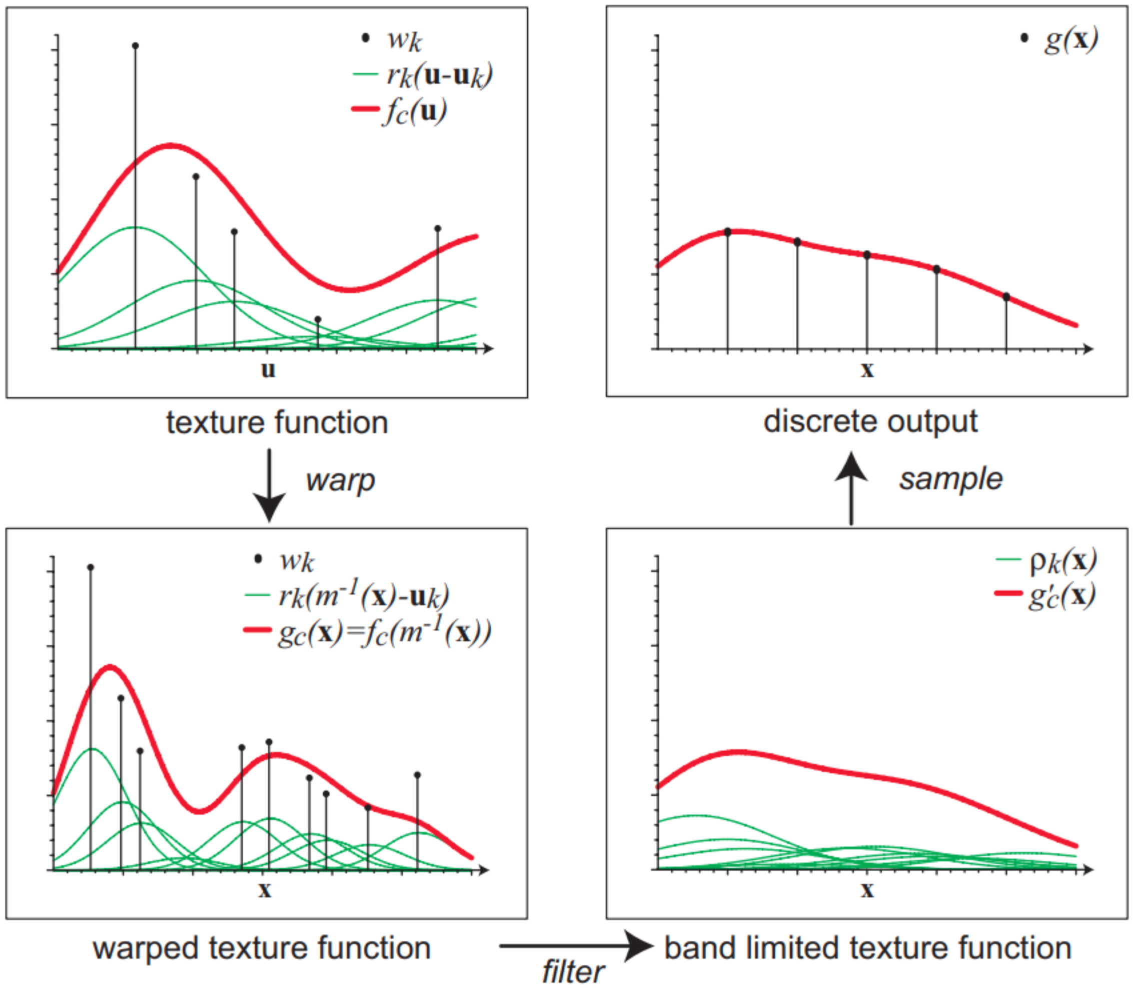
- Point-Based Rendering formula: $C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j),$
- Point-Based Rendering in 3DGS: $f_c(\mathbf{u}) = \sum_{k \in \mathbb{N}} w_k r_k(\mathbf{u} - \mathbf{u}_k).$



- w_k: The Gaussian color under the projection view, computed by Spherical Harmonics (SH)
- $r_k(u - u_k)$: Color blending weight coefficient. Used in **splatting rendering** (i.e., splat opacity), which decays with the distance between the rendering point and the Gaussian center.
- In implementation, the point-based rendering with opacity decay from front to back is still adopted.

Surface Splatting, SIGGRAPH 2001

3DGS Rendering：Splatting Rendering Formula



Surface Splatting, SIGGRAPH 2001

EWA Splatting, TVCG 2002

- The purpose of splatting rendering is to make the contribution of an object to the rendering point decrease with the distance between them.
- Splatting Rendering Pipeline：
  - Warp: Projection of 3D objects onto 2D space
  - Filter: Splat Coefficient Filtering (weight decays with distance)
  - Sample: Sample pixel
- EWA Splat Coefficient:

$$r'_k(\mathbf{x}) = \mathcal{G}_{\mathbf{V}_k}(\mathbf{m}^{-1}(\mathbf{x}) - \mathbf{u}_k) \quad \text{(Sample pixel)}$$

$$= \frac{1}{|\mathbf{W}^{-1}\mathbf{J}_k^{-1}|}\mathcal{G}_{\mathbf{V}'_k}(\mathbf{x} - \mathbf{m}(\mathbf{u}_k)),$$

$$\mathcal{G}_{\mathbf{V}}(\mathbf{x}) = \frac{1}{2\pi|\mathbf{V}|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{V}^{-1}\mathbf{x}}, \quad \text{(Splat Coefficient Filtering)}$$

$$\mathbf{V}'_k = \mathbf{J}_k\mathbf{W}\mathbf{V}_k\mathbf{W}^T\mathbf{J}_k^T. \quad \text{(Projection)}$$

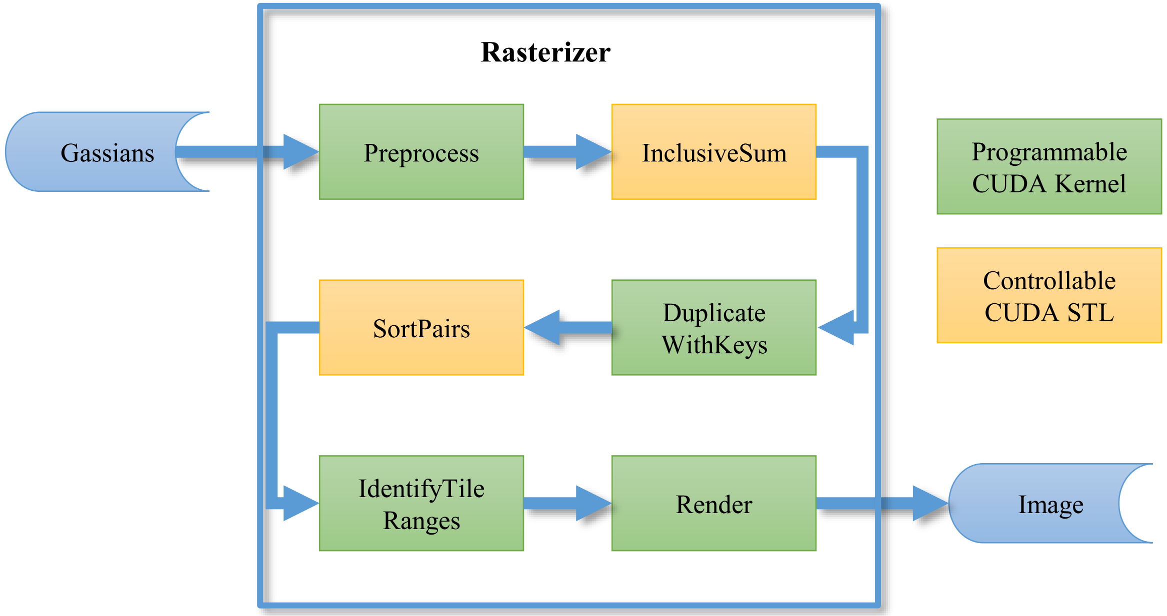
3DGS Rendering：Tile based rendering

Tile Visualization

- Tile-Based Rendering Idea：
  - Divide the image region into several pixel matrices (Tile)
  - Pixels within the same Tile render the same Gaussian set.
  - Pixels between different Tiles render different Gaussian sets.
- Advantages of Tile-Based Rendering：
  - Shared Memory: Pixels within the same tile share Gaussian information efficiently, reducing video memory usage.
  - Efficient Rendering: Pixel threads only render the Gaussians on the corresponding tile.

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Rendering：Rasterization



3DGS Rasterization Pipeline

- Rasterization Stage：
  - Preprocess: Preprocess Gaussians to obtain projection attributes such as projected position, covariance, and color.
  - InclusiveSum&DuplicateWithKeys: Obtain the Tiles covered by Gaussians and create indices for the Gaussian-Tile pairs to be rendered.
  - SortPairs&IdentifyTileRanges: Acquire the index range of Gaussians to be rendered for each Tile.
  - Render: Perform Splatting and AlphaBlending on the Gaussians on the Tile to which the pixel belongs.

3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Effects

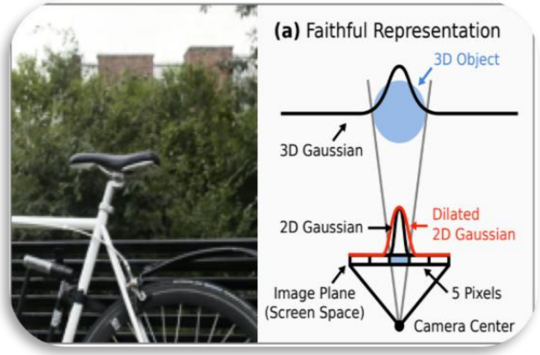
Dataset	Tanks&Temples					
Method|Metric	*SSIM*$^\uparrow$	*PSNR*$^\uparrow$	*LPIPS*$^\downarrow$	Train	FPS	Mem
Plenoxels	0.719	21.08	0.379	25m5s	13.0	2.3GB
INGP-Base	0.723	21.72	0.330	5m26s	17.1	13MB
INGP-Big	0.745	21.92	0.305	6m59s	14.4	48MB
M-NeRF360	0.759	22.22	0.257	48h	0.14	8.6MB
Ours-7K	0.767	21.20	0.280	6m55s	197	270MB
Ours-30K	0.841	23.14	0.183	26m54s	154	411MB

Dataset	Mip-NeRF360					
Method|Metric	*SSIM*$^\uparrow$	*PSNR*$^\uparrow$	*LPIPS*$^\downarrow$	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB
M-NeRF360	0.792†	27.69†	0.237†	48h	0.06	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB

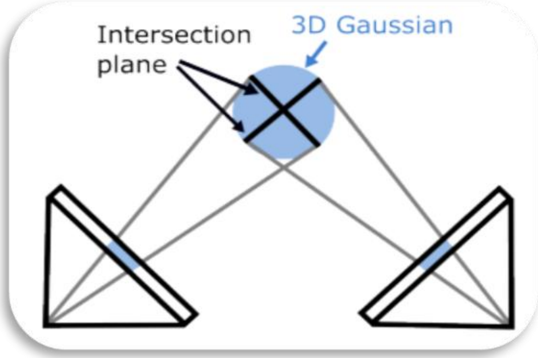
3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023

3DGS Extensions

Rendering Quality

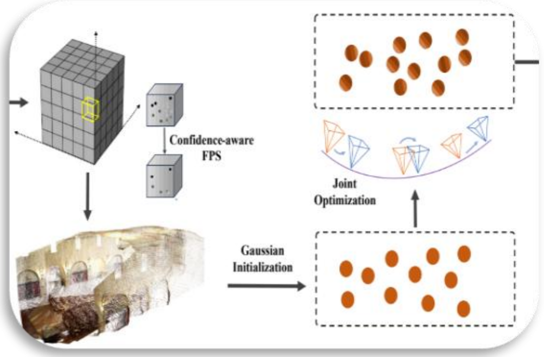


Geometric Quality



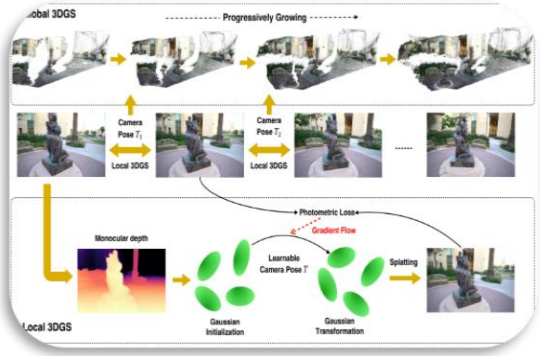
Model Size

Training Speed

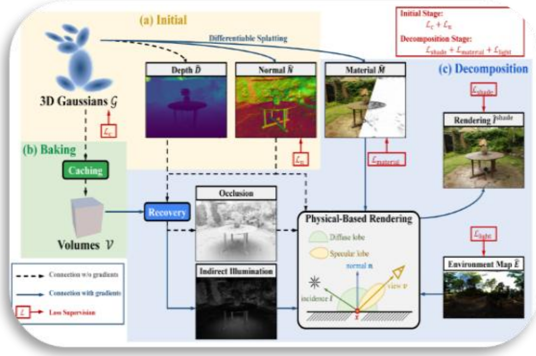


Large-Scale Scene Modeling

Model Robustness



Multimodal Signals



Questions?