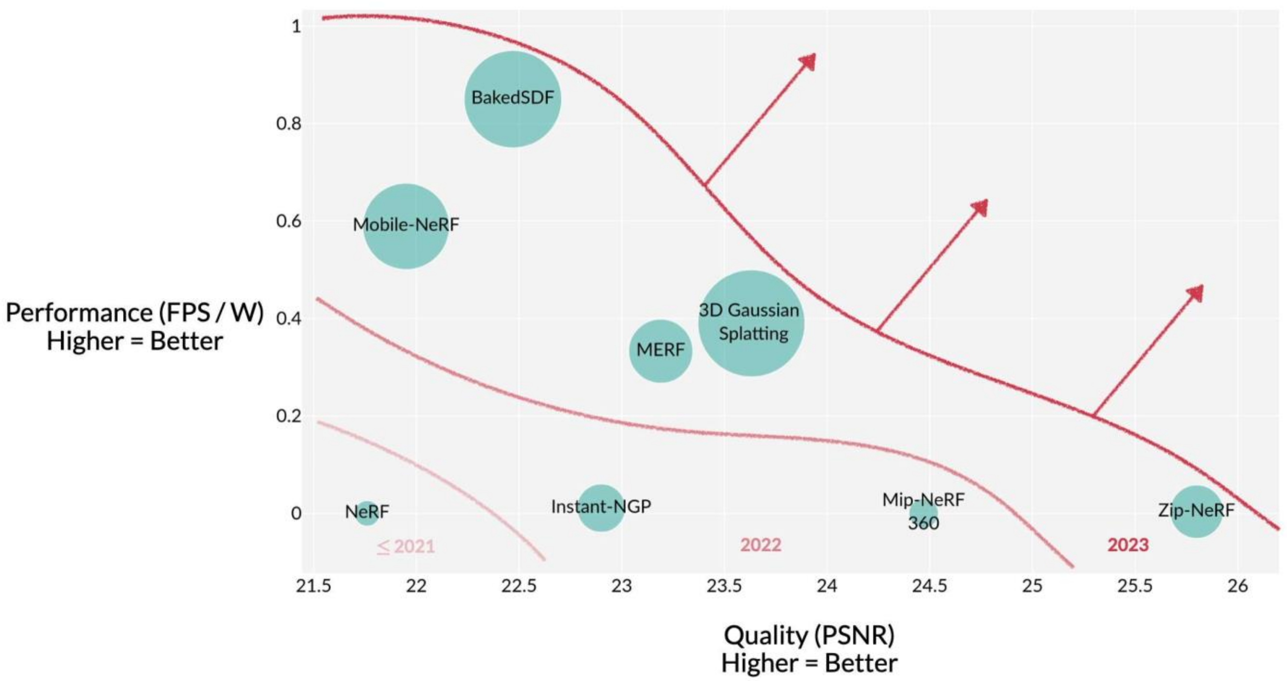
3D Gaussian Splatting for Real-Time Radiance Field Rendering

SIGGRAPH 2023 (Best Paper Award)

박지호
나의 야이아카데미아

Contents

Accuracy vs Efficiency



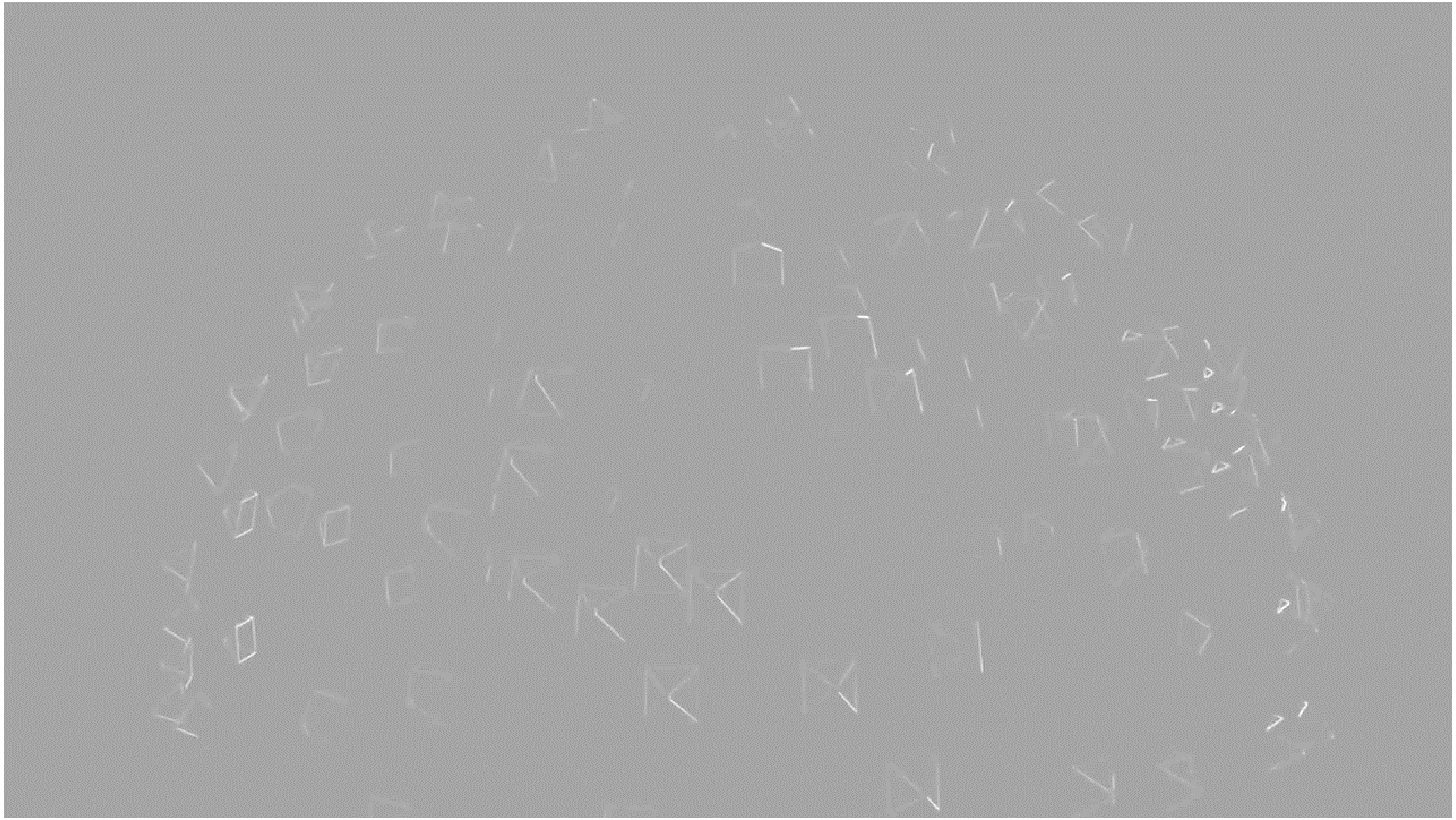
NeRF

Task: 3D Reconstruction from 2D Images

Method: Optimize 3D(MLP) with Ground Truth 2D Images

3D Representation(MLP) → 2D Render → Loss(rendered_img, GT) → optimize MLP

NeRF



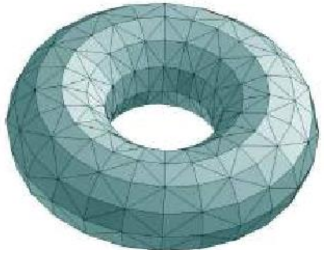
1. 3D Representations

Donut

1. 3D Representations

Donut

Explicit



Mesh:

Points of triangles

Implicit

$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$

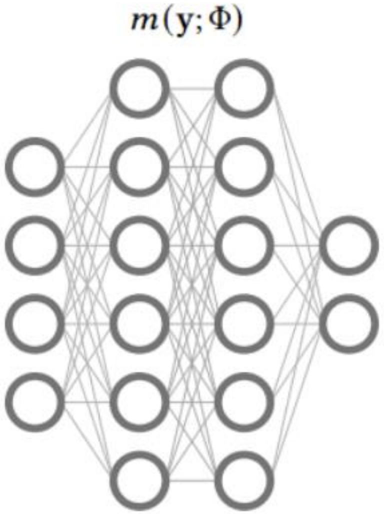
1. 3D Representations

Render 2D

Loss

Ground truth

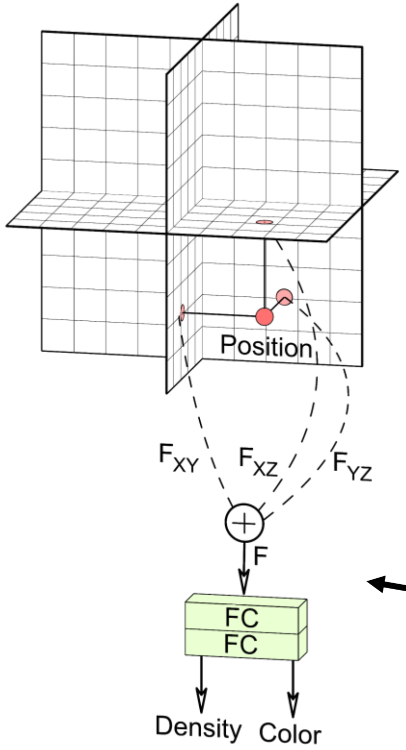
Implicit



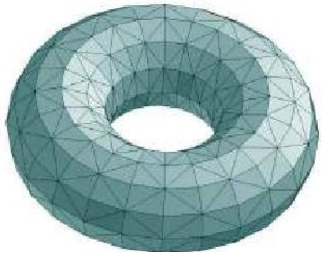
$$(R - \sqrt{x^2 + \cancel{y^2}})^2 + z^2 = r^2$$

1. 3D Representations

Grid Based Hybrid Representation → Sota



Explicit



Vertex:

Points of triangles

Implicit

$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$

1. 3D Representations

b) Hybrid: Grid-based

Implicit

- NeRF(2020): MLP w/ **hierarchical sampling**, positional encoding

    Train time: 12 hours ~ 1 day

    Rendering Speed < 0.1fps

Hybrid

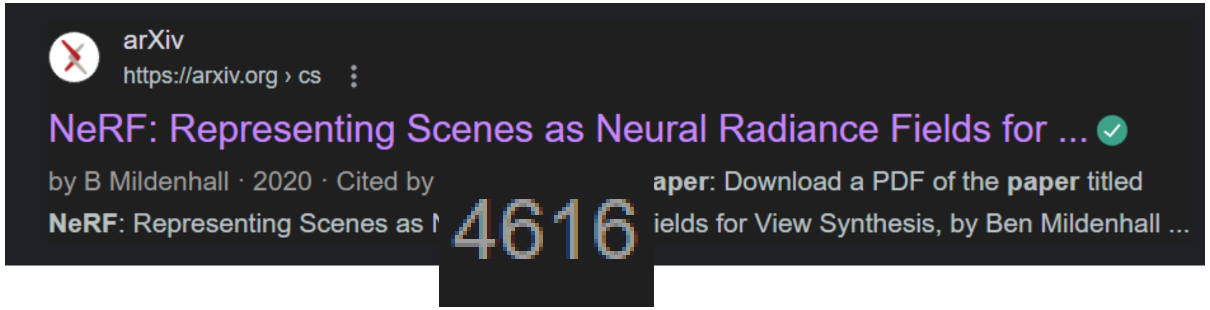
- Utilizing Explicit Representations well!

    Train time: < 30 mins

    Rendering Speed: 5~10 fps

1. 3D Representations

b) Grid-based



Implicit

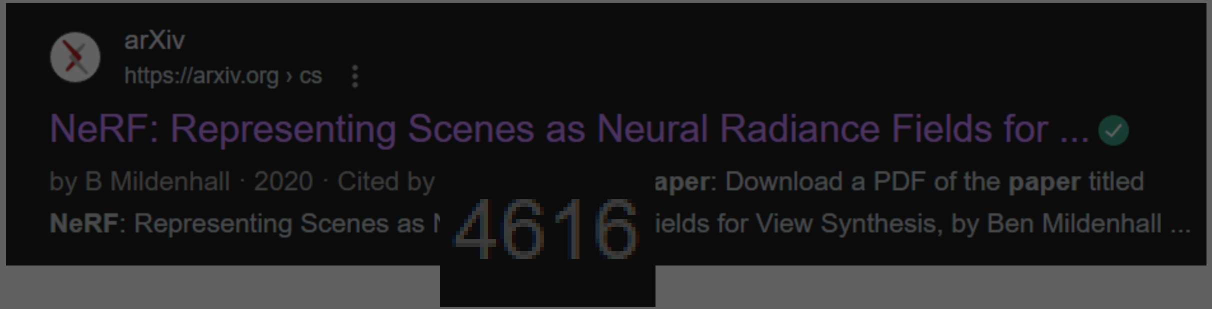
- NeRF(2020) → MLP w/ hierarchical sampling, positional encoding

Grid-Based

- PlenOctree(2021) → Octree Voxel
- Plenoxel(2021) → Sparse Voxel
- InstantNGP(2022) → Multi-resolution Grid
- TensoRF(2022) → Matrix Decomposed Voxel Grid

1. 3D Representations

b) Grid-based



Implicit

- NeRF(2020) → MLP w/ hierarchical sampling, positional encoding

What's in 2023?

Grid-Based

- PlenOctree(2021) → Octree Voxel
- Plenoxel(2021) → Sparse Voxel
- InstantNGP(2022) → Multi-resolution Grid
- TensoRF(2022) → Matrix Decomposed Voxel Grid

1. 3D Representations

c) Point-based

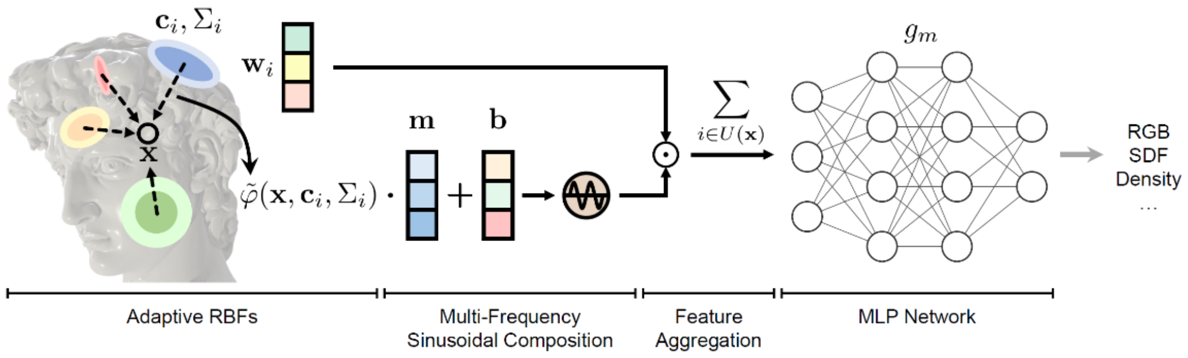
ICCV 2023 Oral

NeuRBF: A Neural Fields Representation with Adaptive Radial Basis Functions

Zhang Chen[1] Zhong Li[1] Liangchen Song[2] Lele Chen[1] Jingyi Yu[3] Junsong Yuan[2] Yi Xu[1]
[1]OPPO US Research Center [2]University at Buffalo [3]ShanghaiTech University

ICCV 2023 (Oral Presentation)

Paper arXiv Video Code



SIGGRAPH 2023 Best Paper Awards

3D Gaussian Splatting for Real-Time Radiance Field Rendering

SIGGRAPH 2023
(ACM Transactions on Graphics)

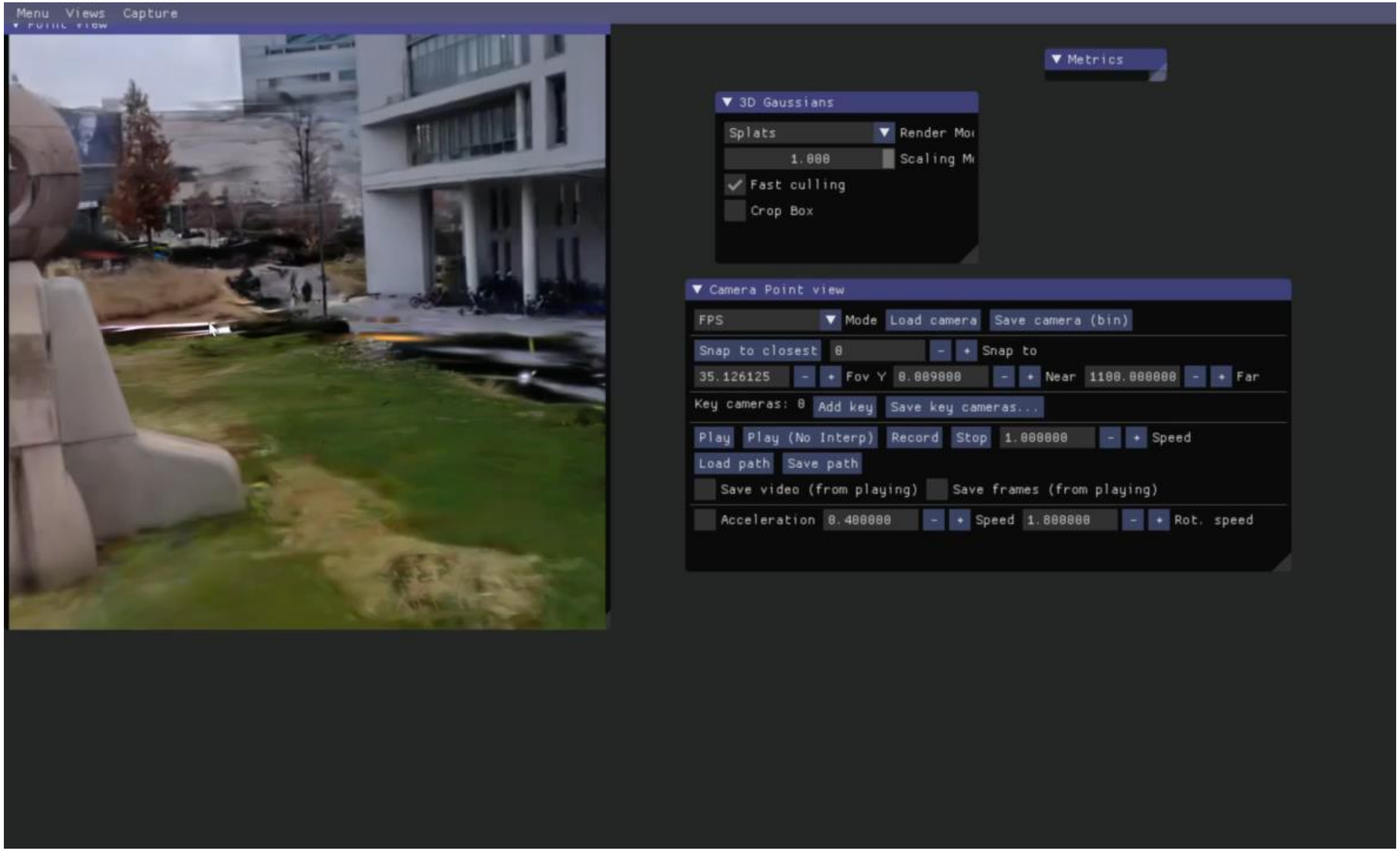
Bernhard Kerbl* [1,2] Georgios Kopanas* [1,2] Thomas Leimkühler[3] George Drettakis[1,2]
* Denotes equal contribution
[1]Inria [2]Université Côte d'Azur [3]MPI Informatik

2. 3D Gaussian Splatting

Train Data

1) 367 images extracted
2) COLMAP(SfM)
3) Trained for 17min (RTX 3090)

2. 3D Gaussian Splatting

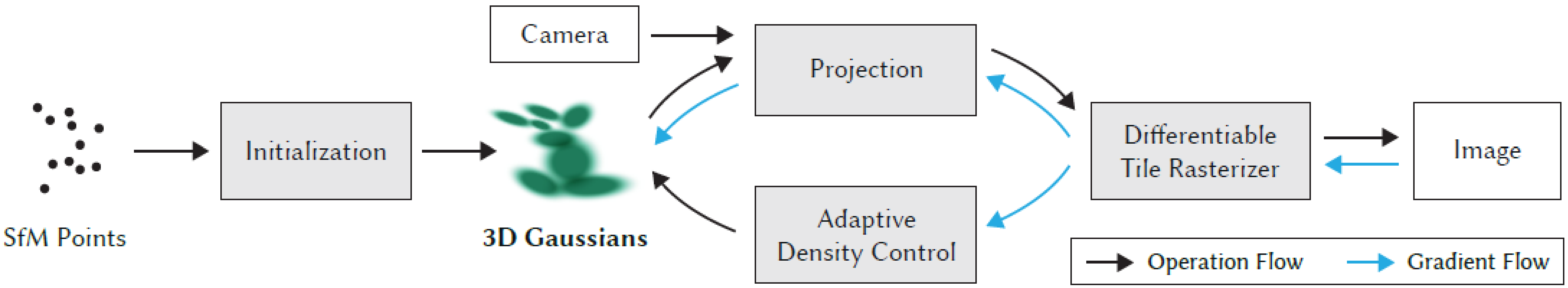


2. 3D Gaussian Splatting

1) Representation: Gaussian Point Cloud (no neural net!)

2) Rendering: Rasterization

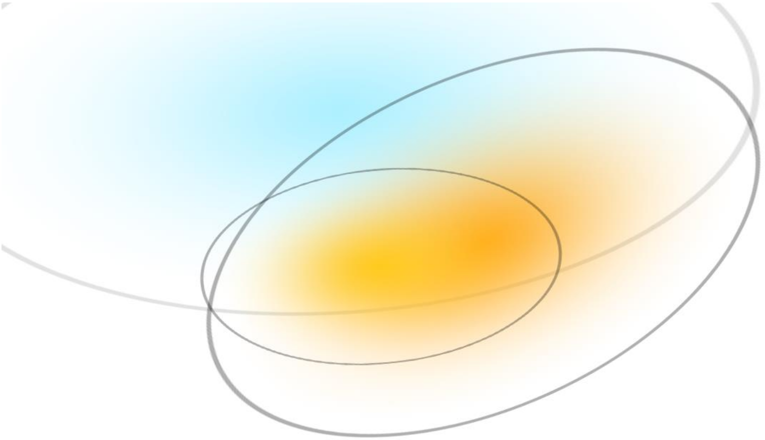
3) Results & 4) Discussion



2. 3D Gaussian Splatting

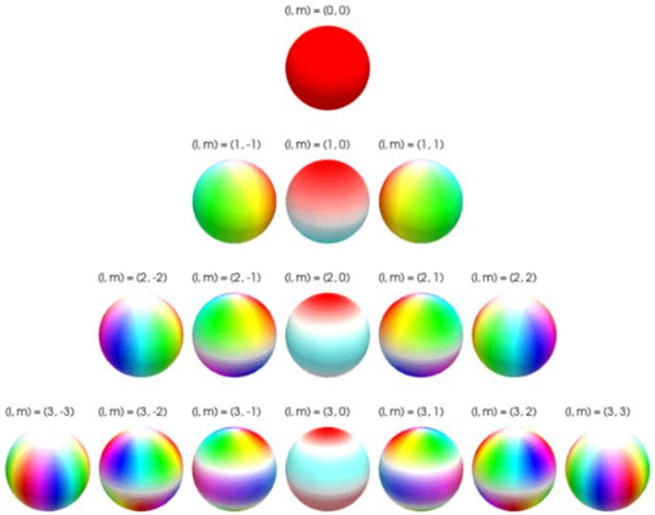
1) Gaussian Point Cloud

- coordinate: x,y,z
- covariance
    - rotation: r
    - scale: s



- density(opacity)
- color:

  Spherical Harmonics coefficient
  (view dependent)

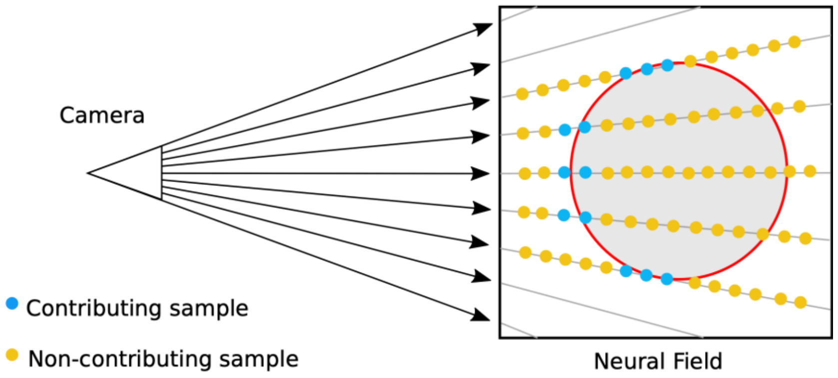


2. 3D Gaussian Splatting

2) **Rendering:** Ray vs Rasterization

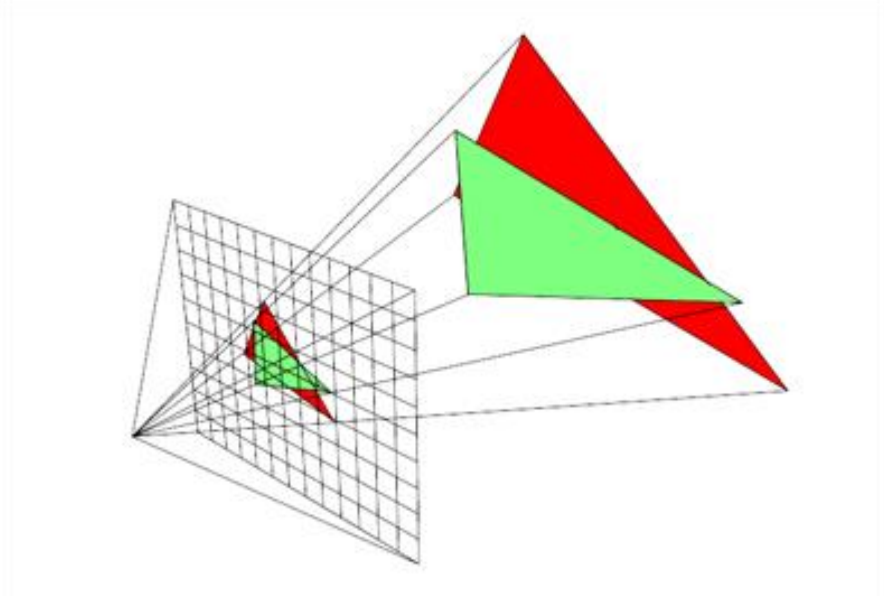
Ray Rendering

- Ray Centric: for NeRF, Implicit



Rasterization

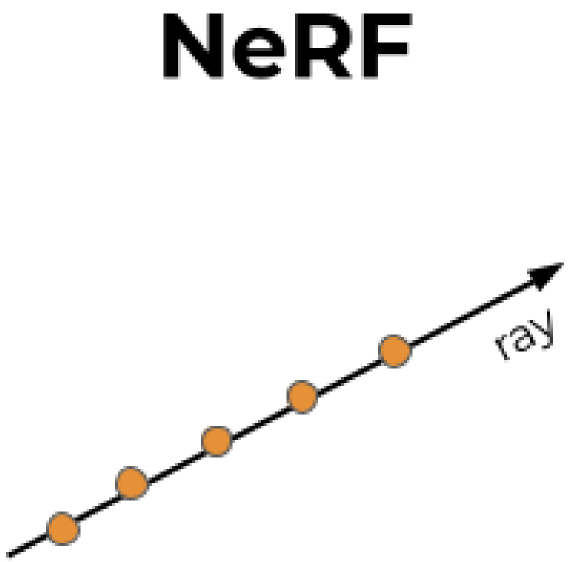
- Object Centric: for explicit(e.g. mesh)



2. 3D Gaussian Splatting

2) Rendering: Ray vs Rasterization

Ray Rendering



Rasterization

2. 3D Gaussian Splatting

2) Rendering: Rasterization

1. Tile-based Frustrum Culling

2. Project & Sorting

Projected Cov: $\Sigma' = JW\,\Sigma\,W^T J^T$

W: world → cam
J: 3D cov → 2D cov jacobian

$$\frac{1}{(2\pi)^{d/2}}|\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})\Sigma^{-1}(\underline{x}-\underline{\mu})^T\right\}$$

3. Point-based Rendering (= Volumetric Rendering formula)

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

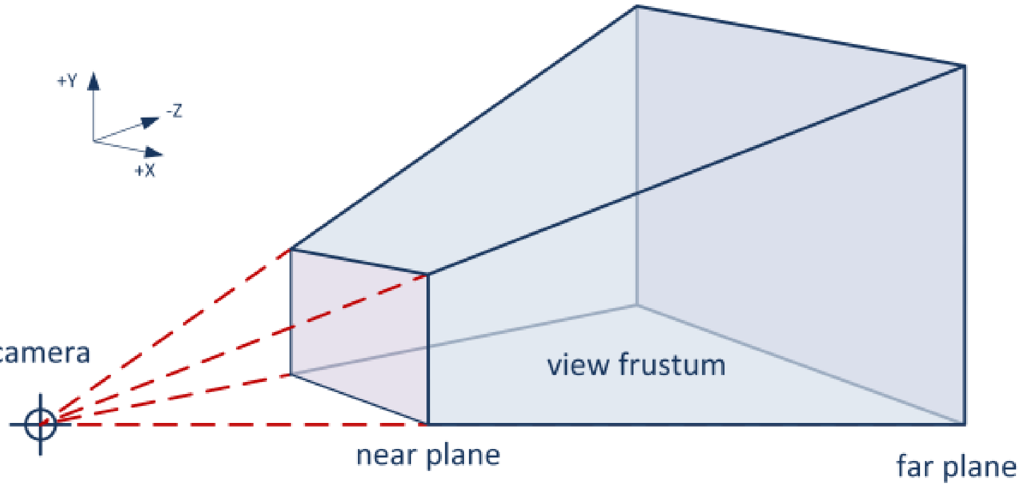
2. 3D Gaussian Splatting

2) Rendering: Rasterization

1. **Tile-based** Frustrum Culling

16x16 tile-based frustum culling *(Pulsar [Lassner and Zollhofer 2021])*

→ Just calculate the gaussian in each tile



$$\frac{1}{(2\pi)^{d/2}}|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})\Sigma^{-1}(\underline{x}-\underline{\mu})^T\right\}$$

2. 3D Gaussian Splatting

3) Result

Advantage

- Fast Rendering
- Fast Training
- High Quality (not best)

Limitation

- Large Memory
- Splotchy Artifacts

Dataset	Mip-NeRF360						Tanks&Temples					
Method|Metric	*SSIM*↑	*PSNR*↑	*LPIPS*↓	Train	FPS	Mem	*SSIM*↑	*PSNR*↑	*LPIPS*↓	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB	0.719	21.08	0.379	25m5s	13.0	2.3GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB	0.723	21.72	0.330	5m26s	17.1	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB	0.745	21.92	0.305	6m59s	14.4	48MB
M-NeRF360	0.792†	27.69†	0.237†	48h	0.06	8.6MB	0.759	22.22	0.257	48h	0.14	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB	0.767	21.20	0.280	6m55s	197	270MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB	0.841	23.14	0.183	26m54s	154	411MB

2. 3D Gaussian Splatting

4) Discussion

View-Dependent Color Recon

- Rasterization 으로도 High Quality Recon이 가능하다!
  (Ray 방식이 아니더라도)

Reflection

2. 3D Gaussian Splatting

4) **Discussion:** Advantages of Explicit

High Compatibility(호환성)

→ Easy Unity, Unreal plugin

Easy Physics Implementation

- without mesh

3. Upcoming Works

Previous NeRF Tasks

Rendering Quality(Anti-aliasing)

- Mip-splatting (Zehao et al)

Few-shot

- Depth-Regularized Gaussian (Jaeyoung et al)

Mesh Extraction

- SuGaR (Antoine et al)

3. Upcoming Works

Dynamic Scene Recon:

- 4D Gaussian Splatting(Gaunjun et al)
- Deformable 3D Gaussian(Ziyi et al)
- Dynamic 3D Gaussians (Jonathon et al)

Text-to-3D:

- DreamGaussian (Jiaxiang et al)
- HumanGaussian (Xian et al)