A Hand-on Tutorial on 3DGS for XR Applications

Shohei Mori[1], [2] Ke Li , [3] **Mana Masuda**
(mana.smile@keio.jp)
[1] Visualization Research Center (VISUS),
University of Stuttgart, Germany
[2] Human-Computer Interaction Group,
Hamburg University, Germany
[3] Keio University, Japan

T2: A PRACTICAL GUIDE TO RADIANCE FIELDS FOR XR RESEARCH AND APPLICATIONS

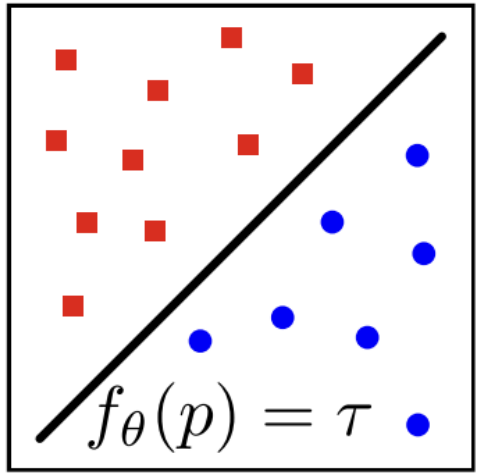
IEEE | IEEE COMPUTER SOCIETY | IEEE COMPUTER SOCIETY VGTC | Inria

Agenda

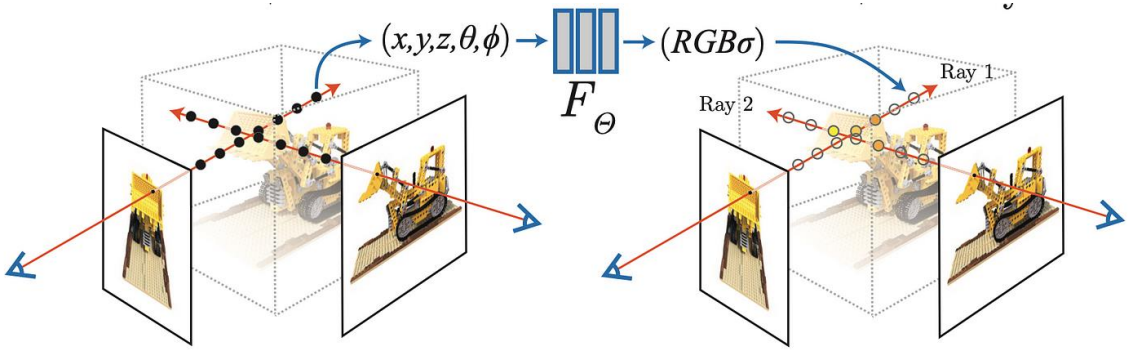
- Brief introduction to **3D Gaussian Splatting (3DGS)**
- Hands on of **UnityGaussianSplatting**
- Live demo on UnityGaussianSplatting toolkit
  - Scene rendering in VR
  - Interaction & editing

3D World World Representation

Implicit Functions (Implicit & Continuous)



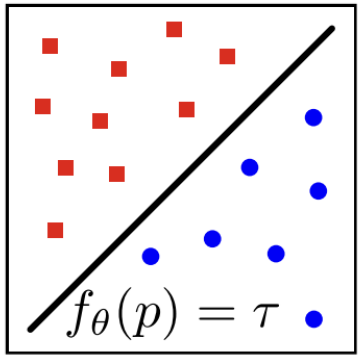
- Represent 3D world with neural networks
- Compress the 3D scene in to a function
- NeRF achieved the photorealistic representation



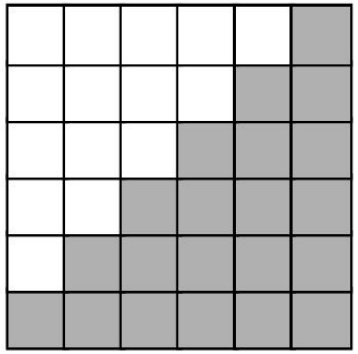
Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

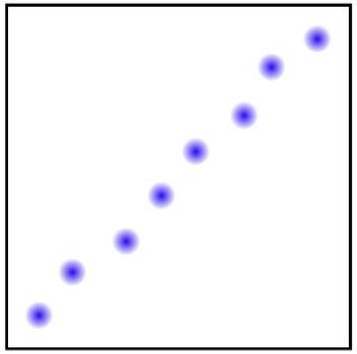
3D World World Representation

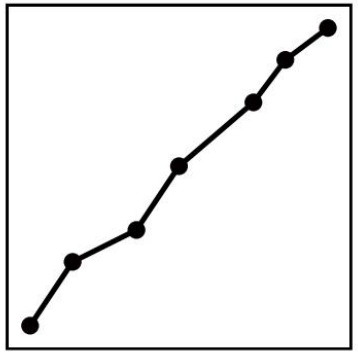
Implicit



Explicit

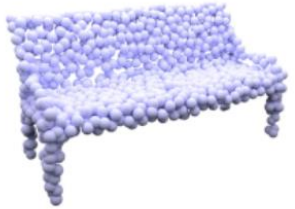






Scene Function

Voxel



Point Cloud

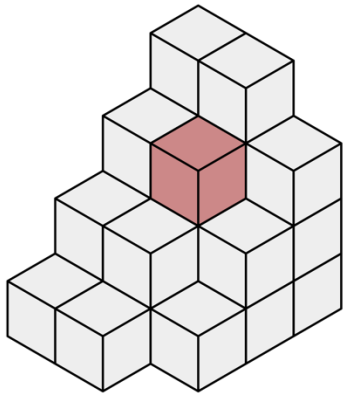
Mesh

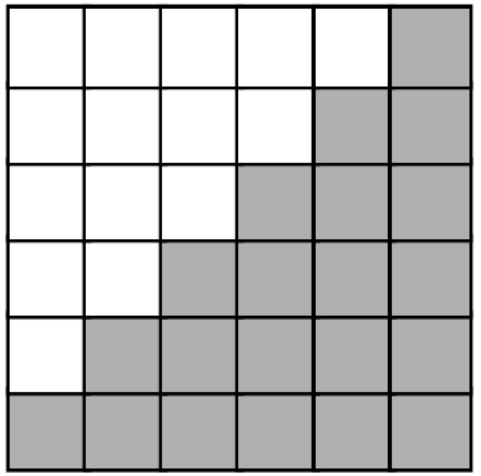
Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

3D World World Representation

Voxels (Explicit & Discrete)

- Represent 3D space with grids
- Easy to process
- Cubic memory O(n3) → limited resolution

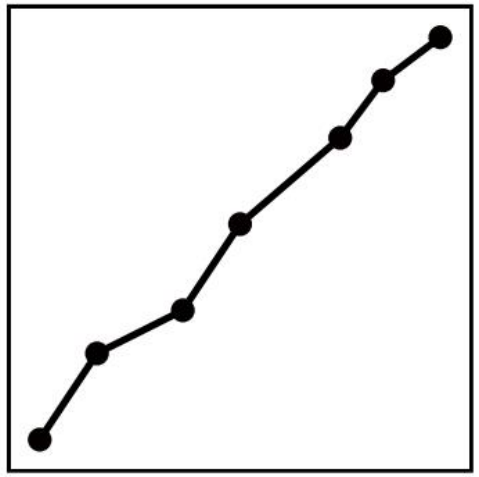




Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

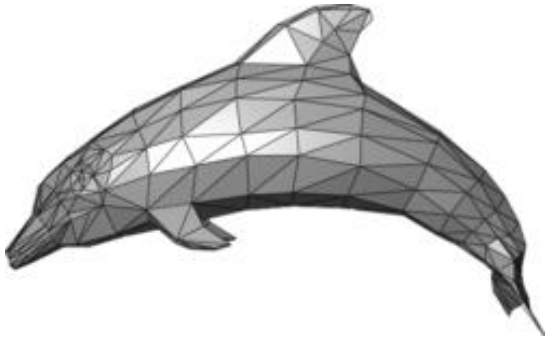
3D World World Representation

Mesh (Explicit & Discrete)



- Represent objects surface with vertices and faces
- Limited number of vertices

  → reconstruction accuracy is limited

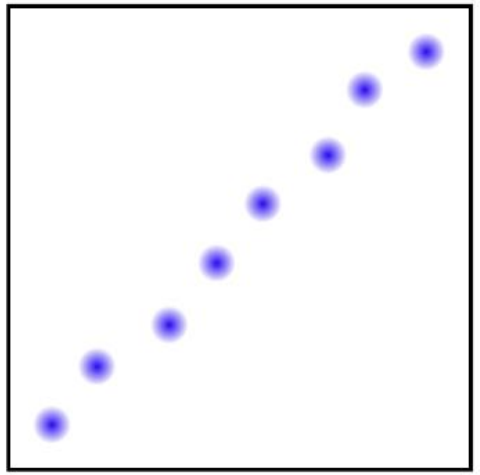


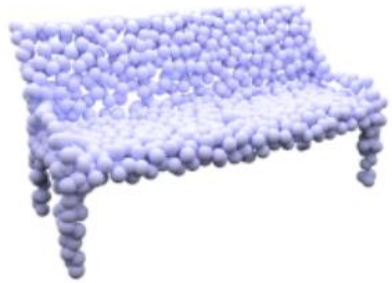
Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

3D World World Representation

Point Clouds (Explicit & Discrete)

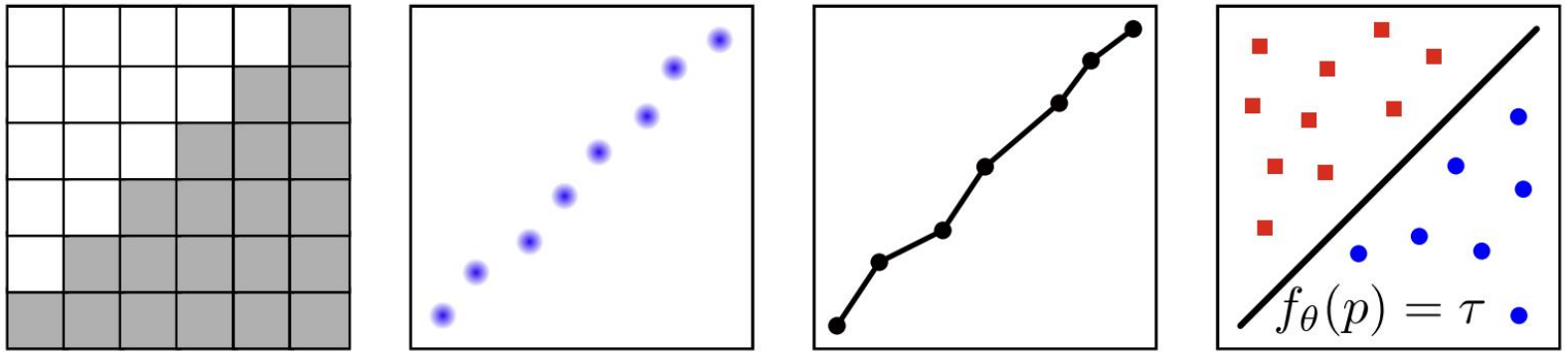
- Represent surface with 3D points
- Does not have connected surface
- Limited number of points

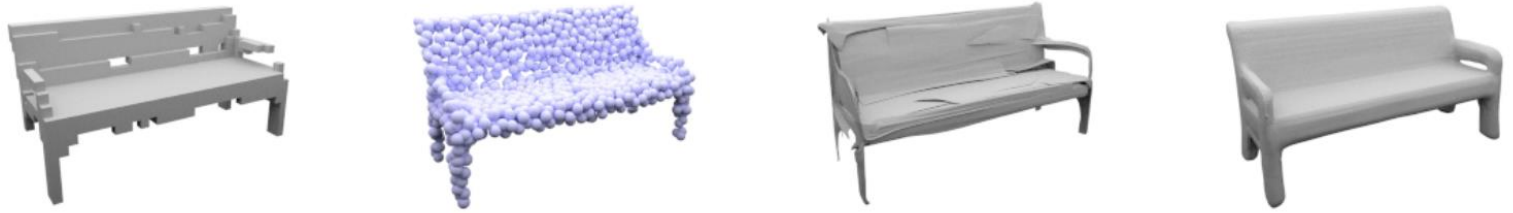




Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

How to Represent the 3D World?





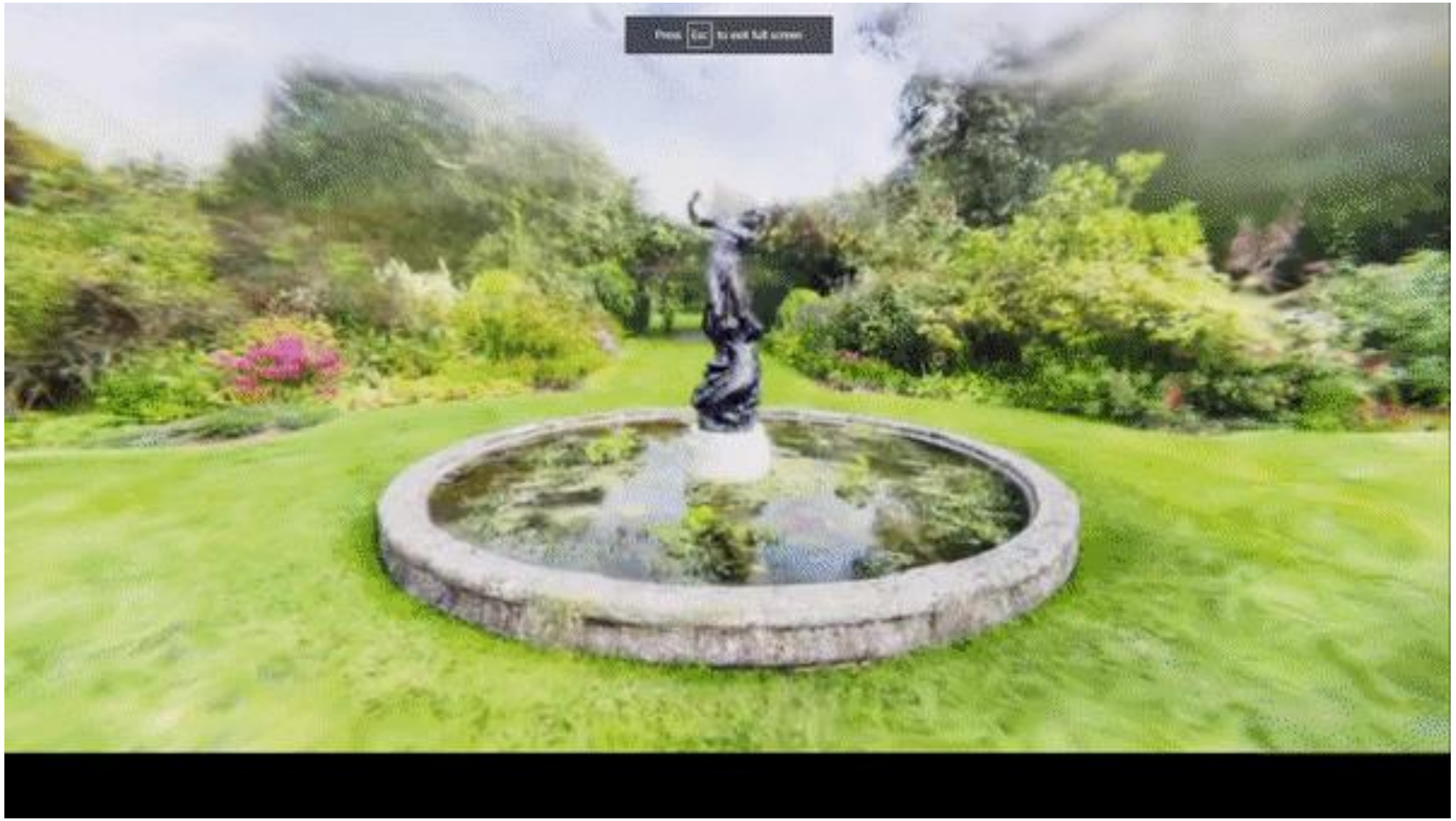
Voxel | Point Cloud | Mesh | Implicit Function

Which is the best representation of 3D world for **photo realistic** and **fast rendering speed**?

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019

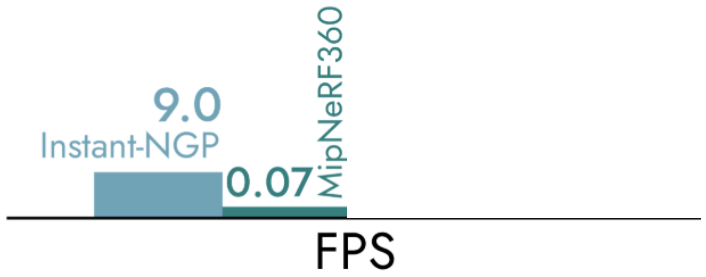
3D Gaussian Splatting

Represents radiance fields with points!

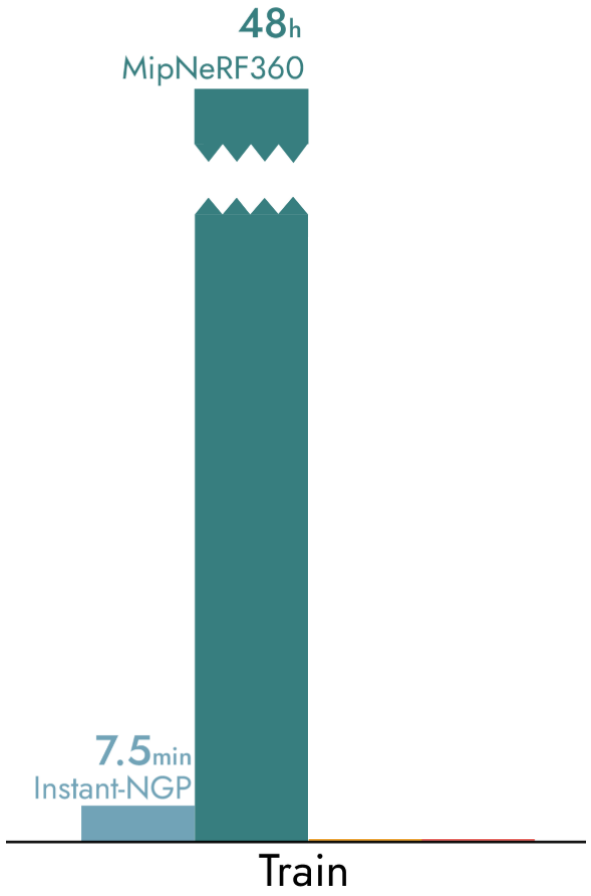


Rendering/Training Speed

Rendering Speed



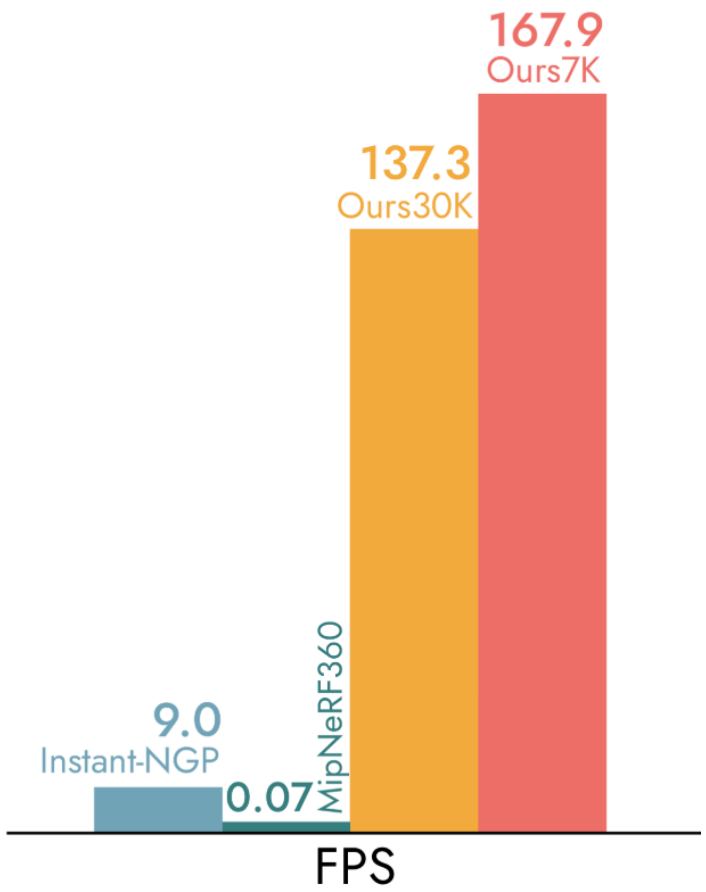
Training Speed



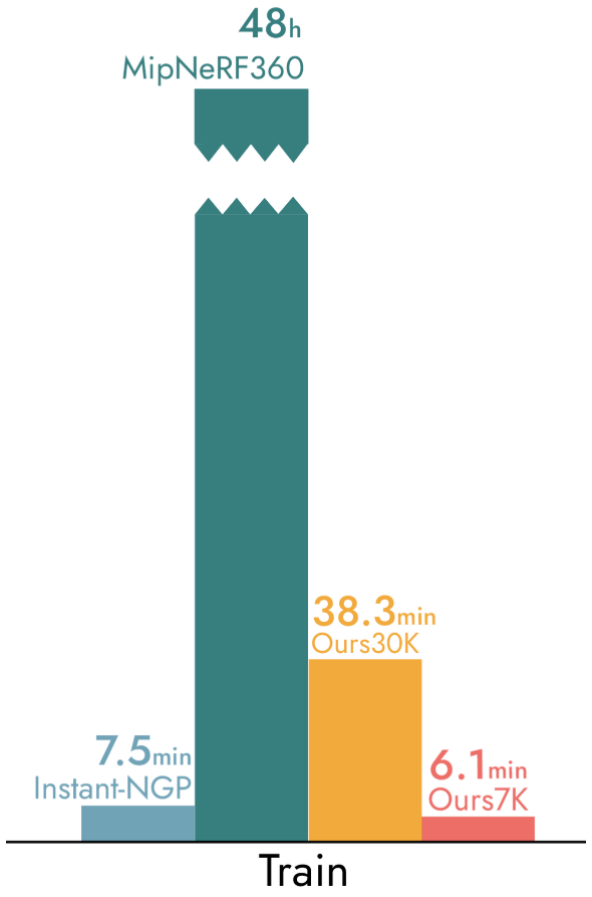
https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Rendering/Training Speed

Rendering Speed



Training Speed



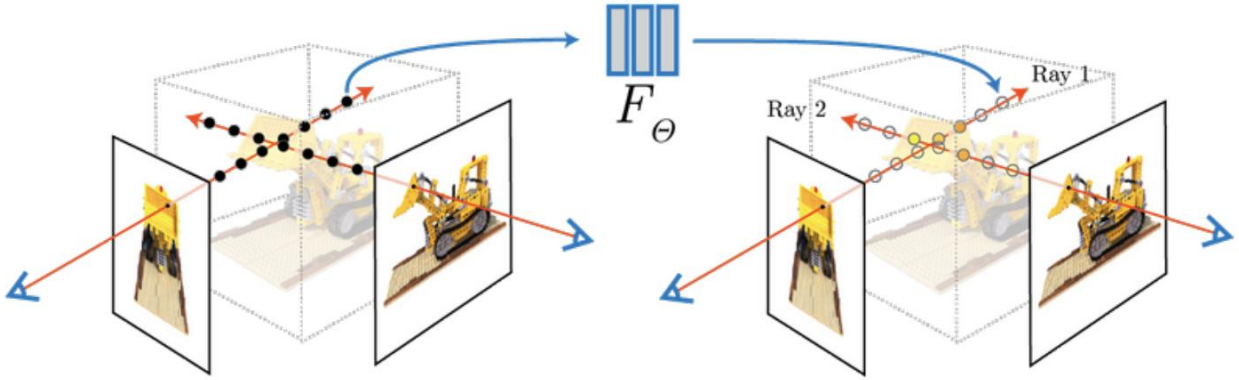
https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

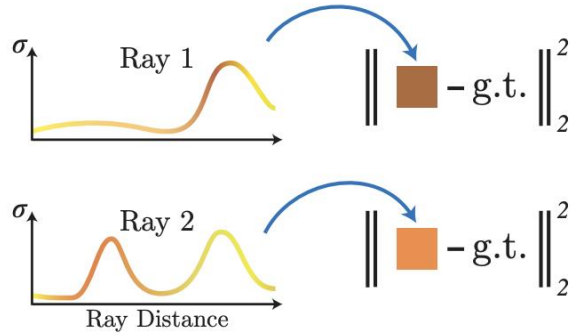
Neural Radiance Field

Parameterize Radiance Field **densely**, at **every point** in space

$$(x,y,z,\theta,\phi) \rightarrow \boxed{\quad} \rightarrow (RGB\sigma)$$

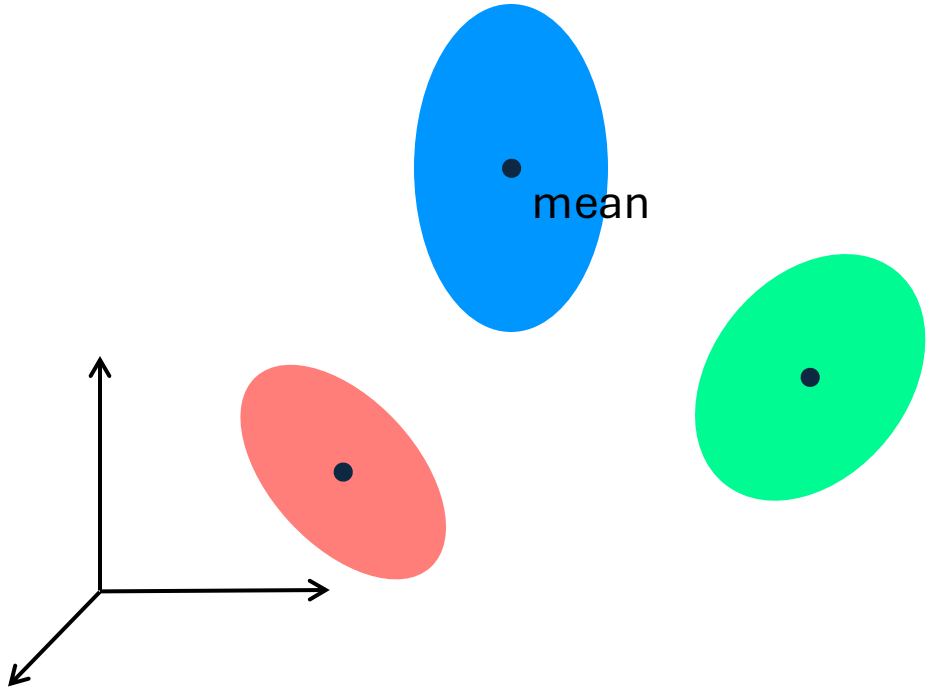
$$F_\Theta$$





3D Gaussians

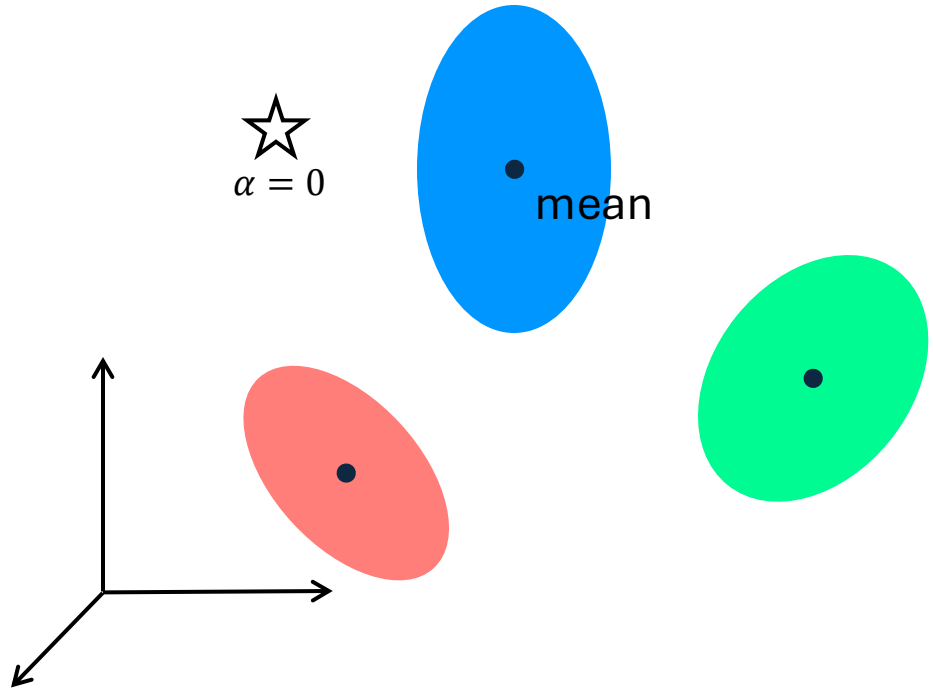
Key Idea: Parameterize Radiance Field **sparsely**, only where **density ≠ 0**



Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians

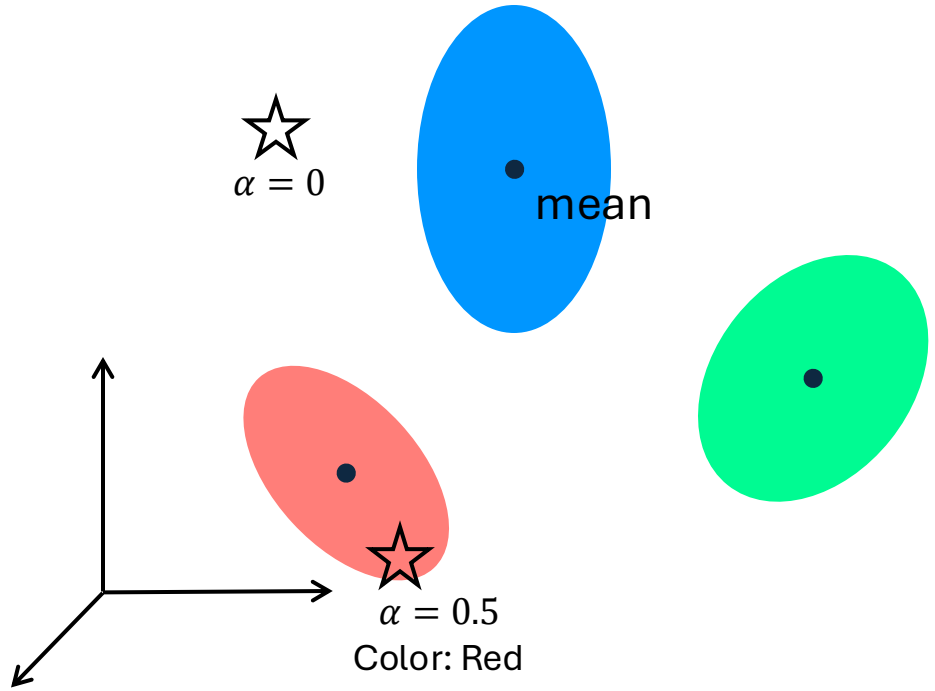
Key Idea: Parameterize Radiance Field **sparsely**, only where **density ≠ 0**



Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians

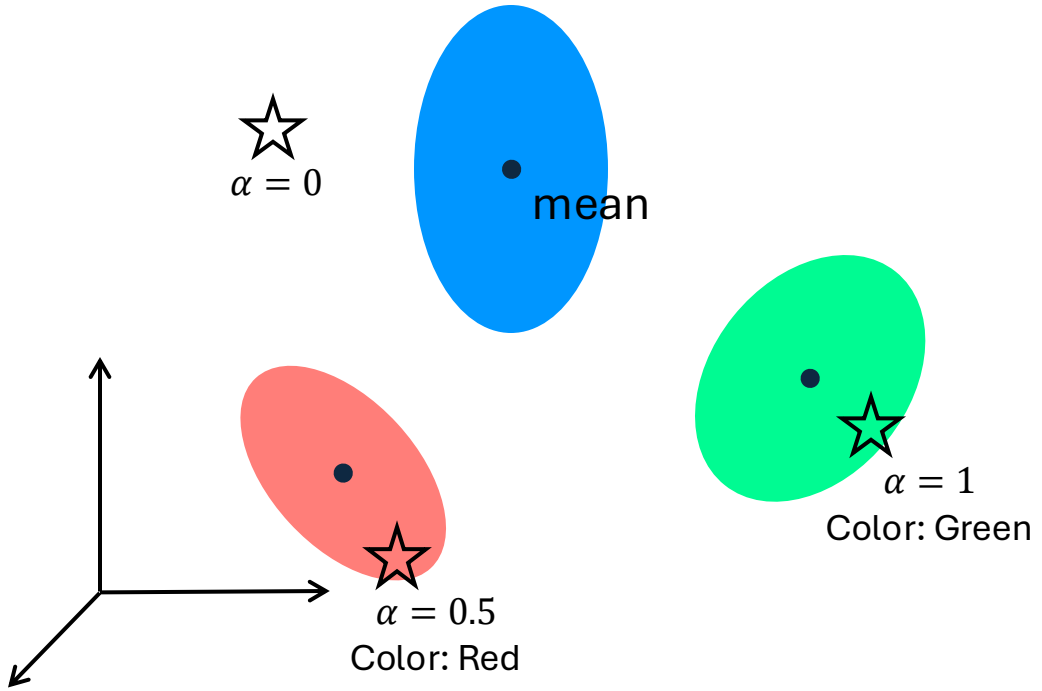
Key Idea: Parameterize Radiance Field **sparsely**, only where **density ≠ 0**



Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians

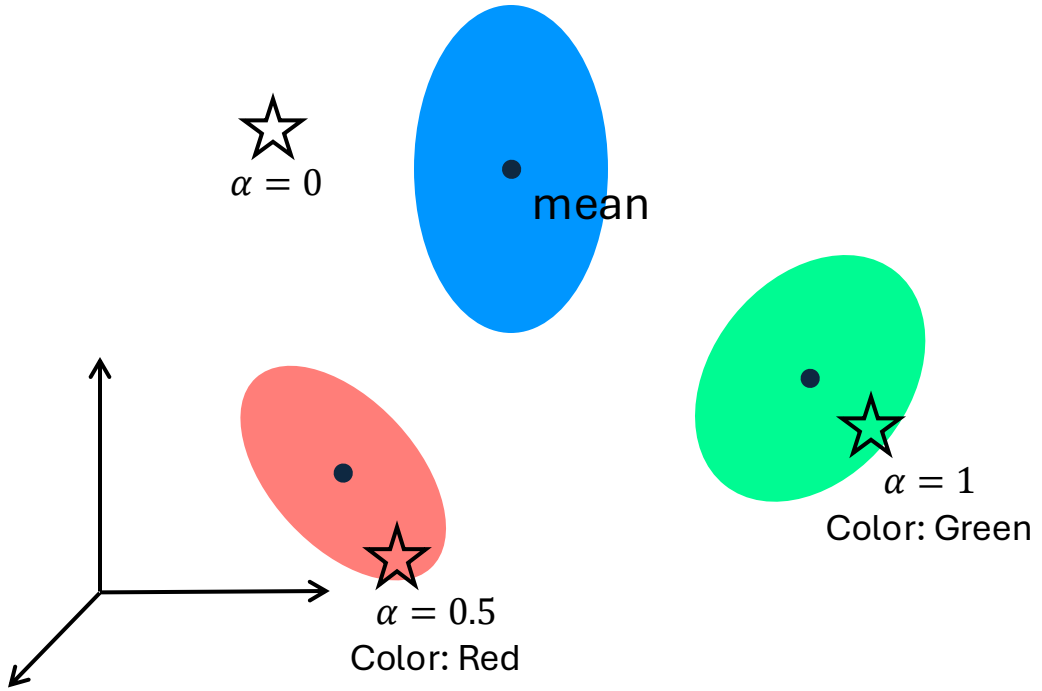
Key Idea: Parameterize Radiance Field **sparsely**, only where **density ≠ 0**



Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians

Key Idea: Parameterize Radiance Field **sparsely**, only where **density ≠ 0**

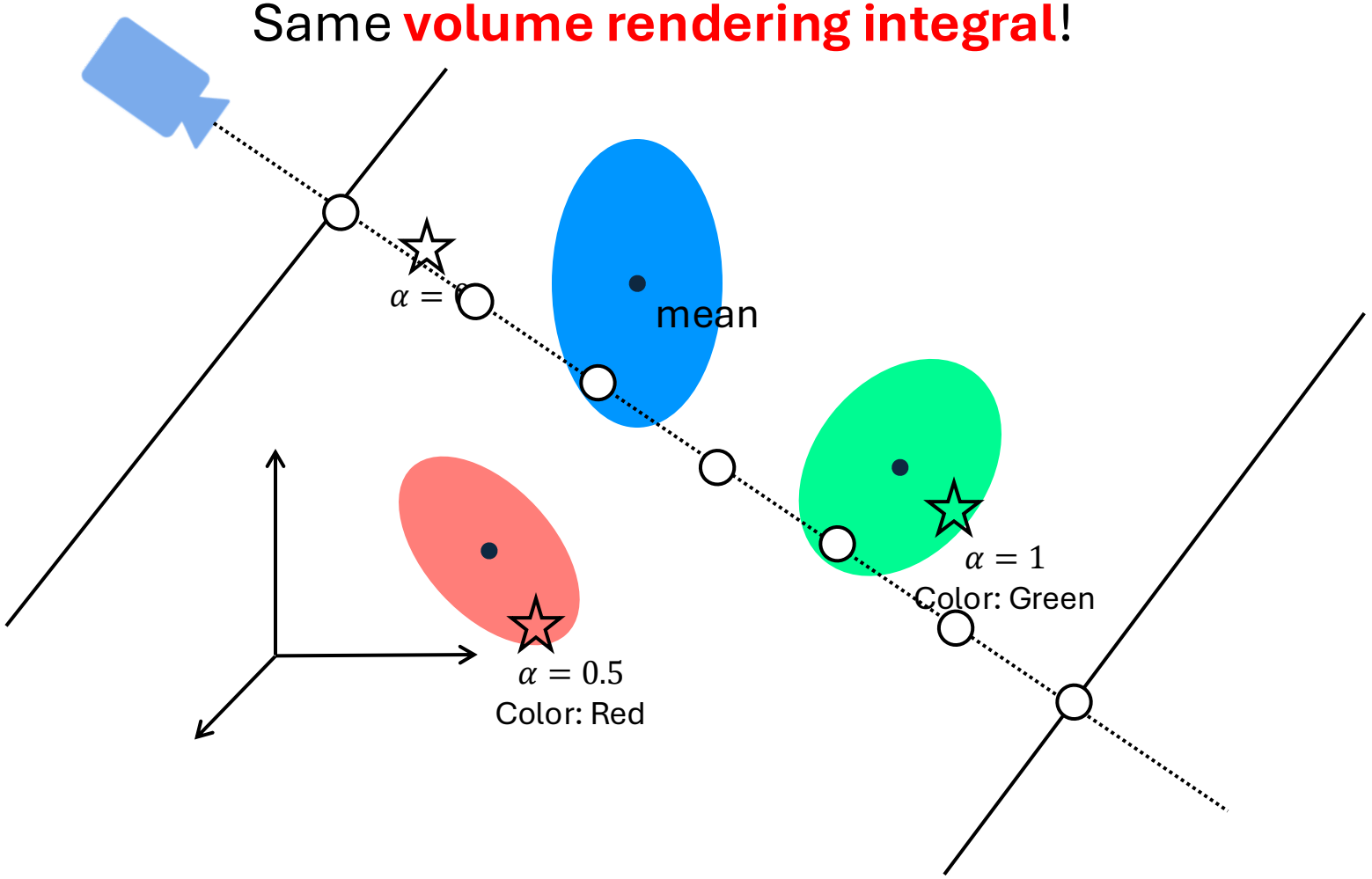


...How to render?

Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians -- Rendering

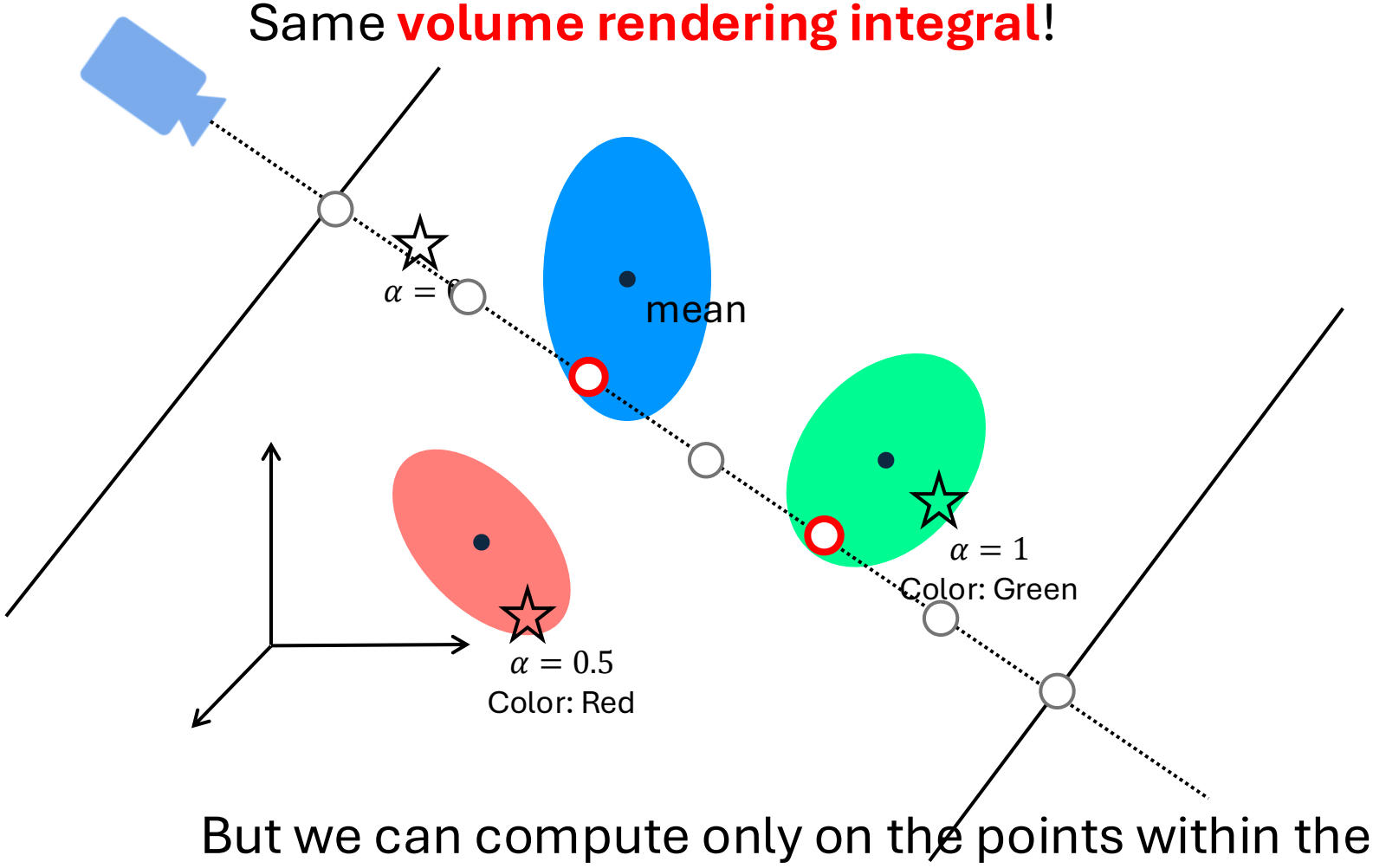
Same **volume rendering integral**!



Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians -- Rendering

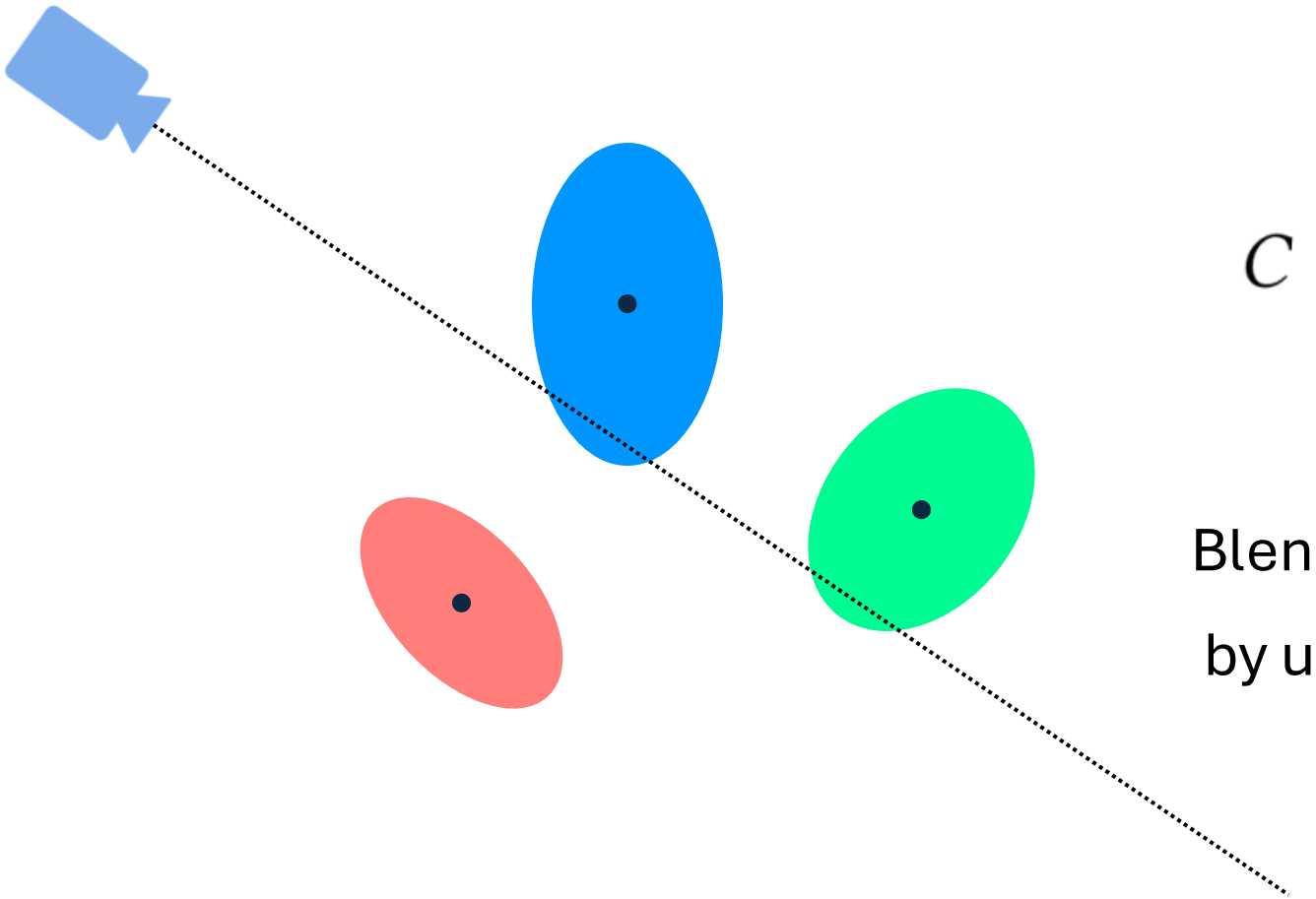
Same **volume rendering integral**!



But we can compute only on the points within the Gaussians!

Reference: 6.S980 – ML for Inverse Graphics – Vincent Sitzmann

3D Gaussians



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

Blend together the 3D Gaussian points

by using the density of each points

Rendering Functions – NeRF vs 3DGS

- Discreet volumetric rendering equation for NeRF

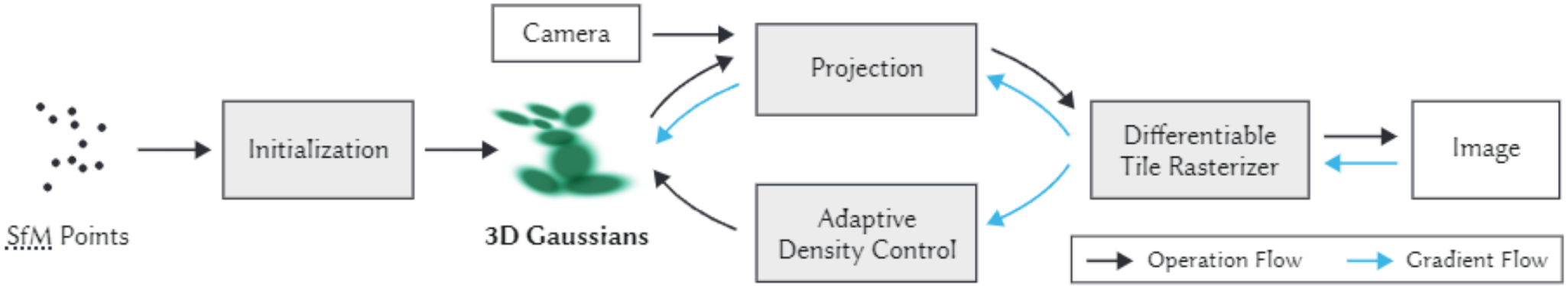
$$C = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \quad \text{with} \quad T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$$

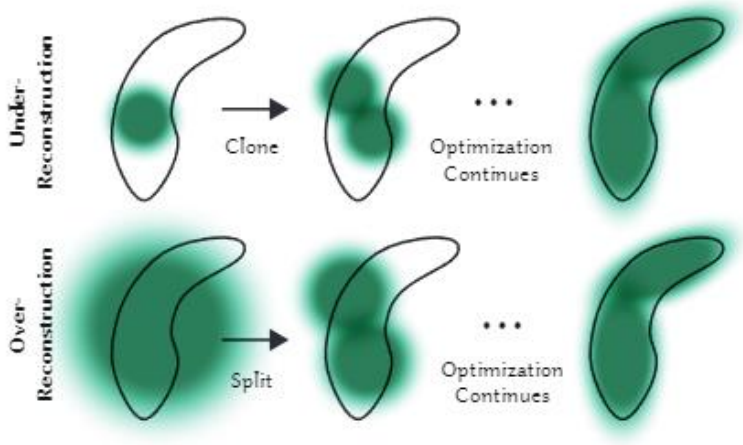
$$C = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \qquad \alpha_i = (1 - \exp(-\sigma_i \delta_i)) \text{ and } T_i = \prod_{j=1}^{i-1} (1 - \alpha_i).$$

- Neural point-based rendering for blending N order points overlapping the pixels

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

3DGS Training / Optimization





Split / Combine Gaussians until the Gaussian fits to the scene details!

Summary: NeRF or 3DGS

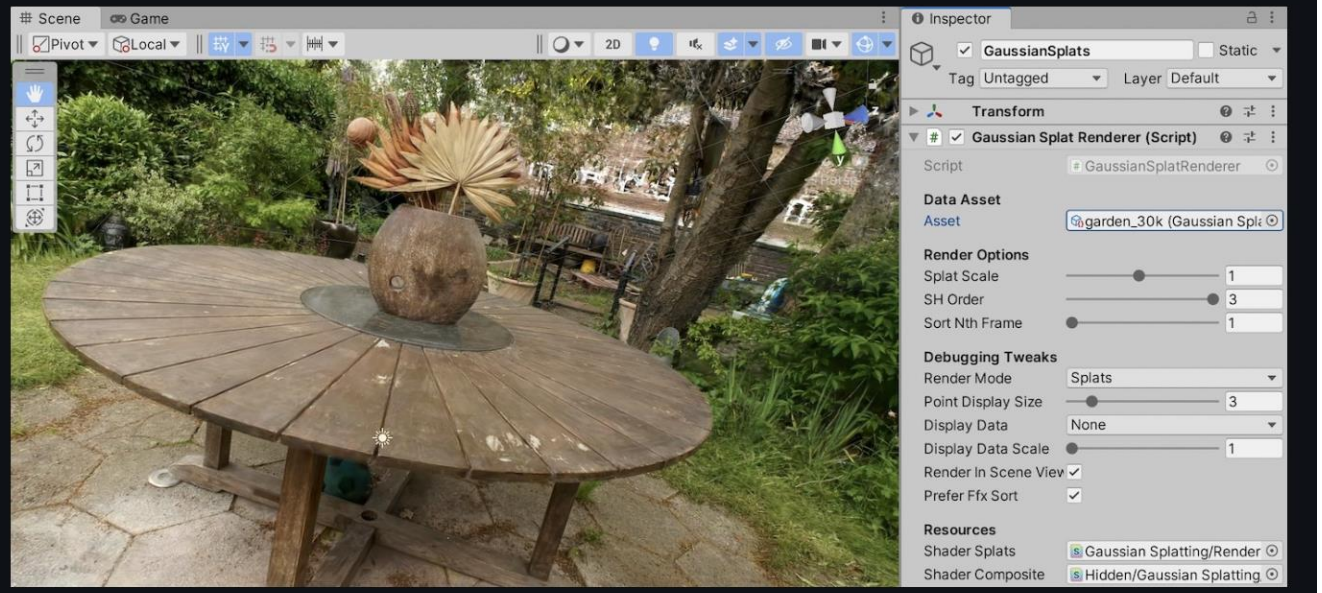
	NeRF	3D Gaussian Splatting
Representation	Implicit Function	3D Gaussian Points
Computational Cost	Heavy volumetric rendering	**Fast rasterization rendering**
Compatibility with 3D tools	Limited	**Relatively Easy**
Interactbility	Complicated geometry-based editing	**Relatively easy to add objects**
When insufficient input views?	Hallucinate details	Could not render details - Missing Gaussians
Pose Sensitivity	Require precise camera calibration	Require precise camera calibration

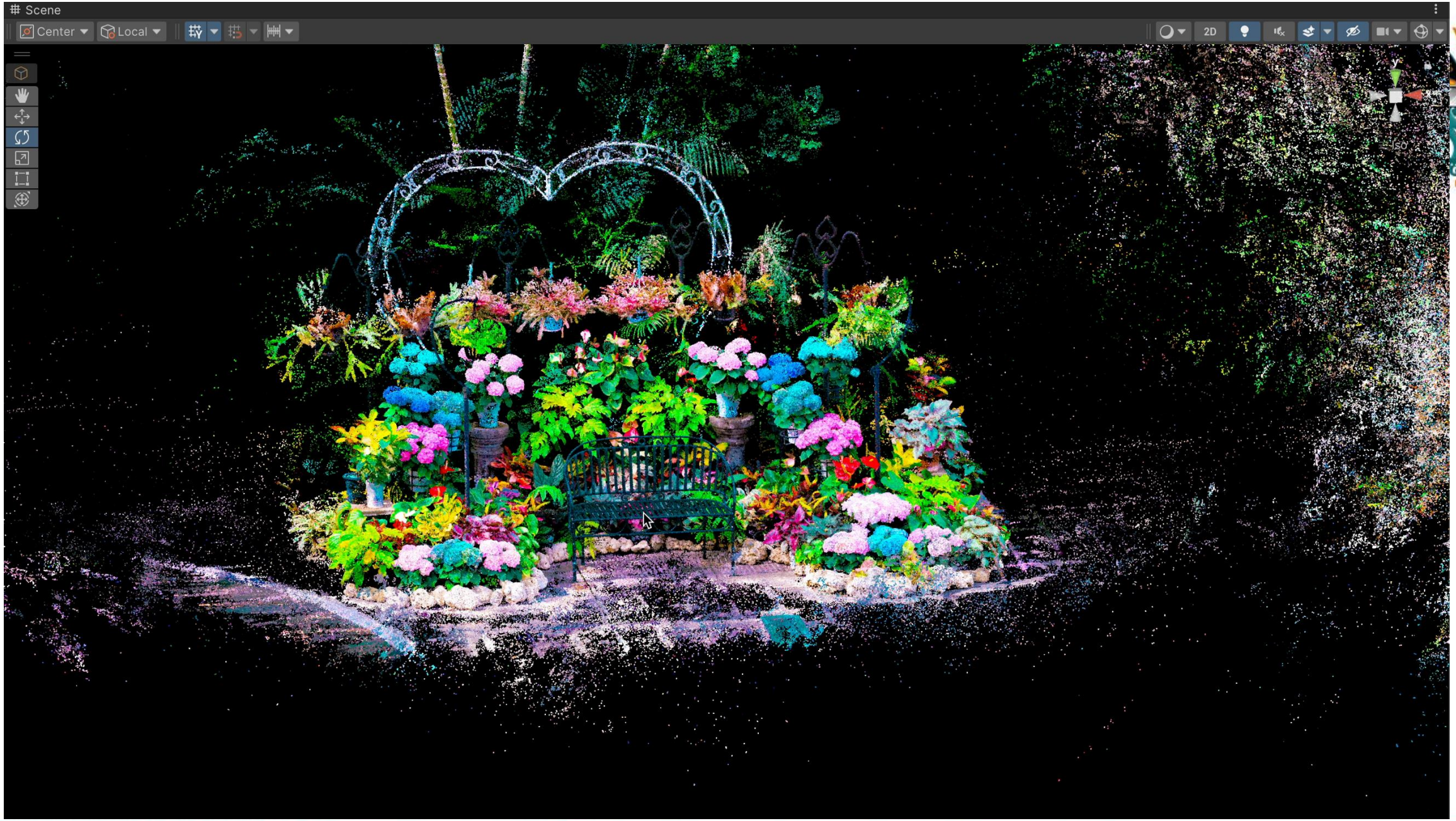
UnityGaussianSplatting

The open source tool to play 3D Gaussian splatting data in Unity!

Gaussian Splatting playground in Unity

SIGGRAPH 2023 had a paper "3D Gaussian Splatting for Real-Time Radiance Field Rendering" by Kerbl, Kopanas, Leimkühler, Drettakis that is really cool! Check out their website, source code repository, data sets and so on. I've decided to try to implement the realtime visualization part (i.e. the one that takes already-produced gaussian splat "model" file) in Unity.

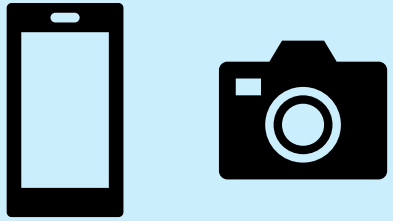



Scene
Center
Local
2D

Just 3 steps to play in Unity!

Step1

Recording



Step2

Train 3DGS

https://aras-p.info/blog/2023/09/27/Making-Gaussian-Splats-more-smaller/

Step3

Import to Unity

Unity

+

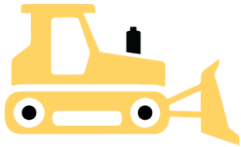
UnityGaussianSplatting

GitHub

Create 3D Gaussians?

There are lot of ways to create 3DGS scenes!

SCANIVERSE



nerfstudio

LUMA AI

... and also original 3DGS code!

How to use?

IEEE VR 2023 SAINT-MALO, FRANCE

Live Demo!