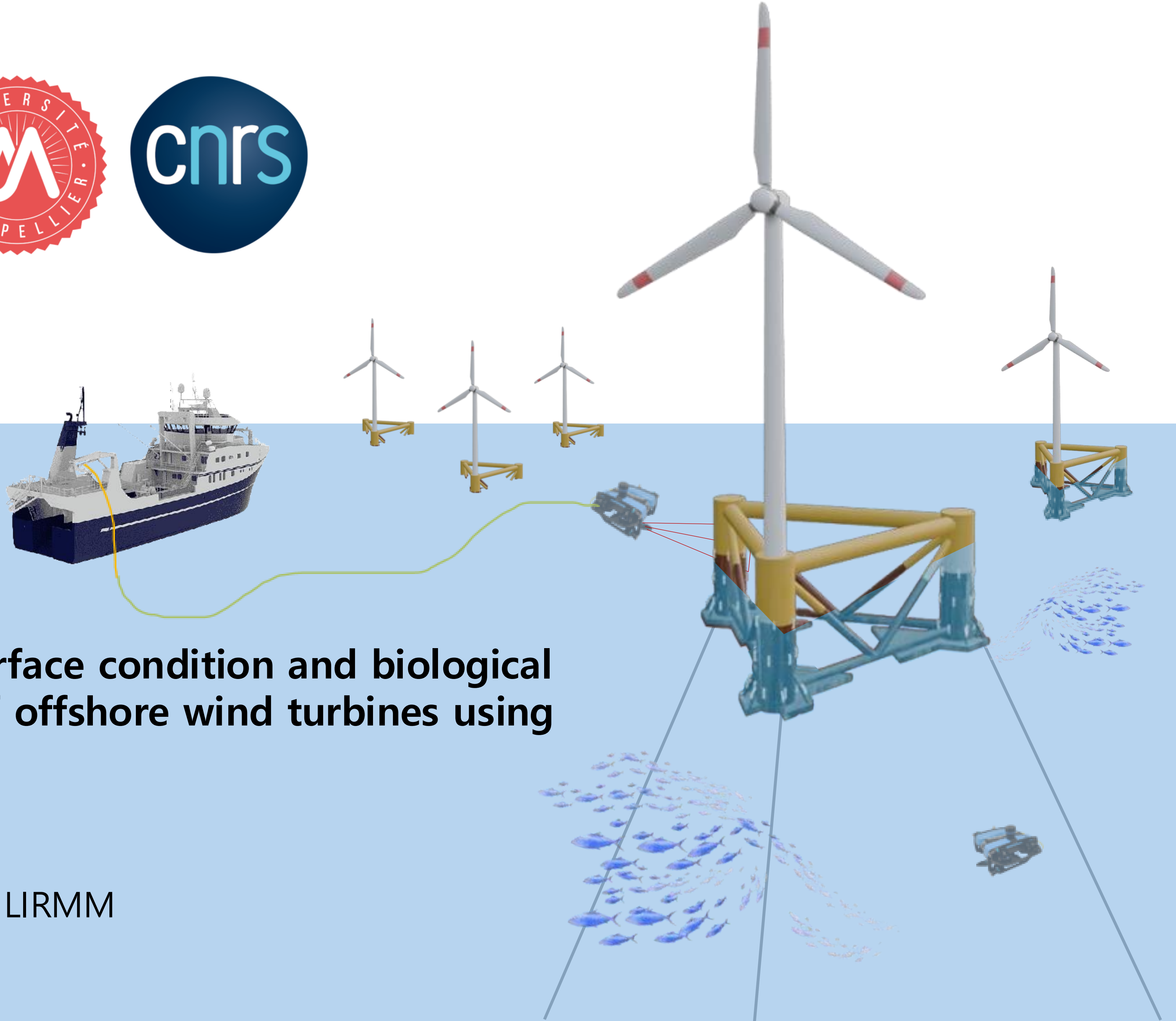
20/01/2026

PhD Updates

AcoustiBioMap:
Mapping and monitoring of the surface condition and biological colonization of submerged parts of offshore wind turbines using acoustic imaging



Alessandro Puglisi

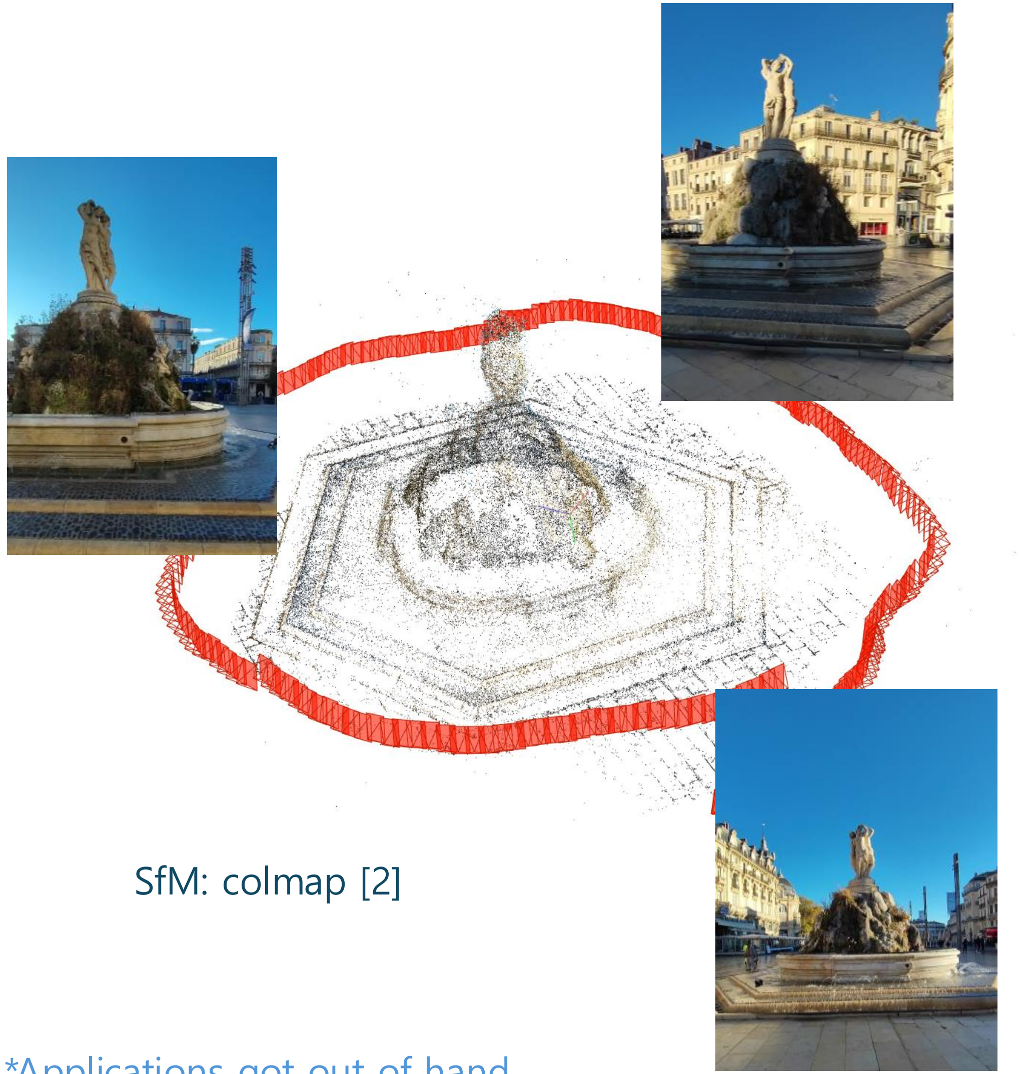
PhD Candidate in Marine Robotics, RSM, LIRMM

alessandro.puglisi@lirmm.fr

The new 3D trend

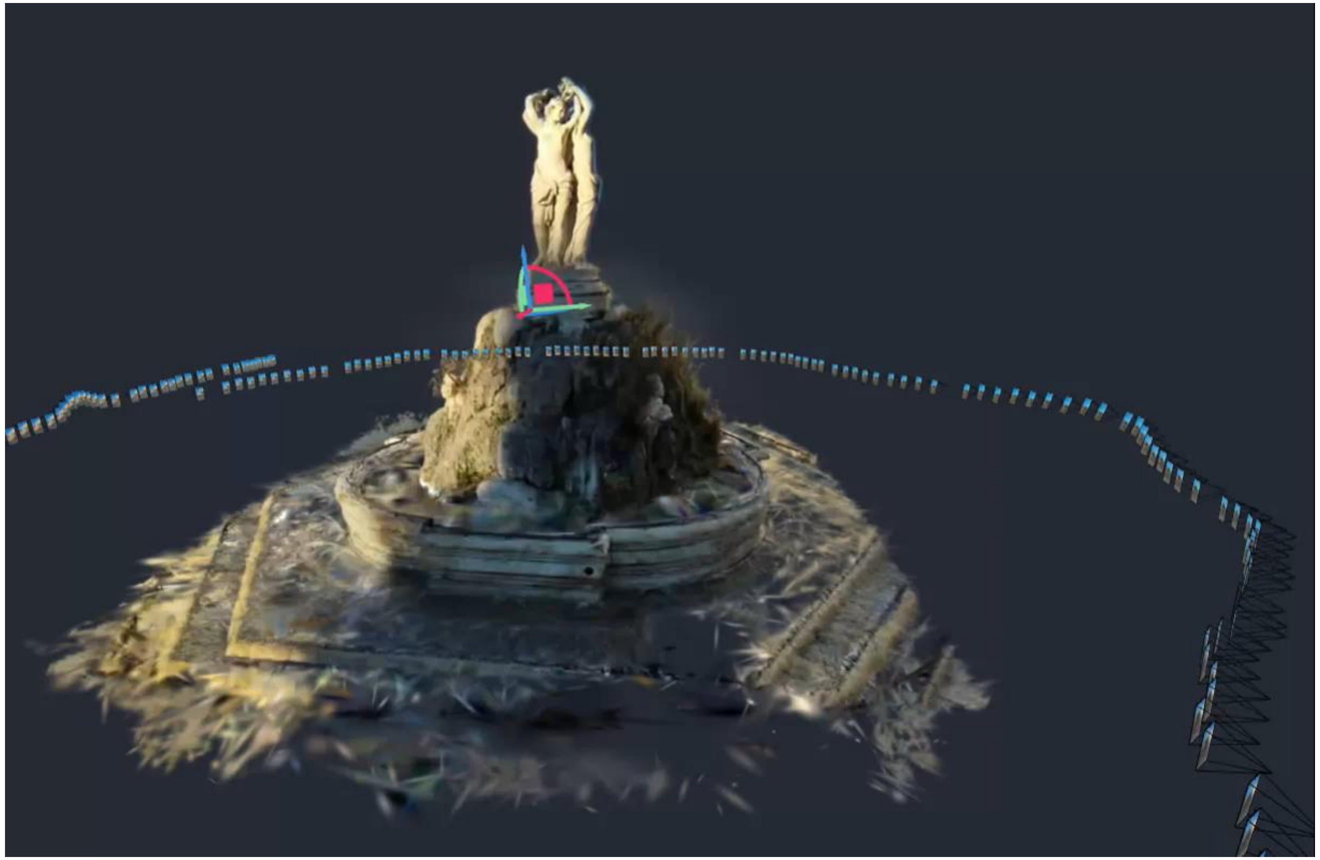
Radiance Fields Techniques

The (Initial*) Objective: Novel View Synthesis



SfM: colmap [2]

Geometry + Appearance



nerfstudio: gsplat [3]

*Applications got out of hand...

The Building Blocks: Radiance Field

Radiance

radiance is the radiant flux emitted, reflected, transmitted or received by a given surface, per unit solid angle per unit projected area

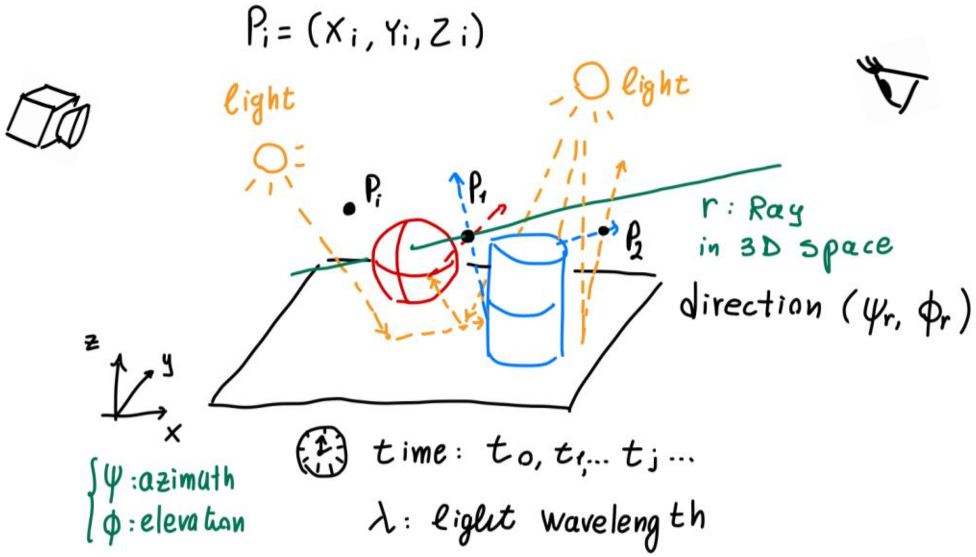
parametrizing it...

Plenoptic Function

$L(X, Y, Z, \psi, \phi, \lambda, t)$

Complete light intensity model

Independent from the observer



How much radiant energy passes through a specific area in a specific direction

Different terminology in radiometry and computer graphics...

Radiance Field (computer graphics perspective)

$$L : X, Y, Z, \psi, \phi \rightarrow r, g, b, \sigma$$

Spectral(RGB)-volumetric vector field

Model internal entities properties (density)

Light Field

$$L(X, Y, Z, \psi, \phi, \lambda, t) \quad \forall P = (X, Y, Z), d = (\psi, \phi), \lambda, t$$

Spectral-temporal vector field

No internal entities properties (density)

The Building Blocks: Rendering

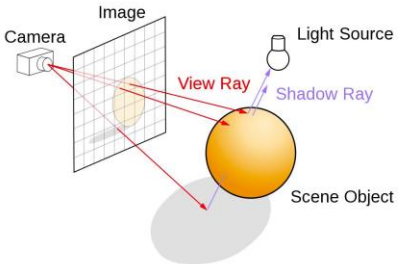
Rendering

3D Scene Model

2D View

"All rendering methods attempt to model the same physical phenomenon, that of light scattering off various types of surfaces." (J. T. Kajiya, 1986)

Ray Tracing

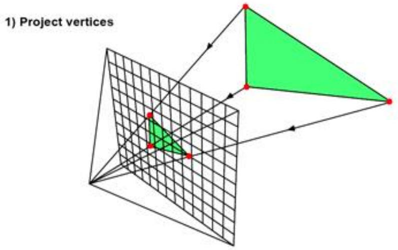


Realistically simulate the lighting of a scene and its objects.
Tracing the path of light from the view camera

Photorealistic

Computationally expensive

Rasterization



convert the triangles of the 3D models into pixels on a 2D screen.

Fast computation

Approximated render

Ray "Shooters"

Shoot ray from camera

Ray Casting (analytical)

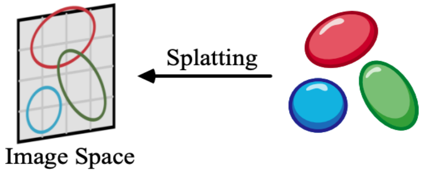
Compute intersections
Compute distances

Ray Marching (iterative)

March along ray, based on distance fields (sdf)
Look for intersections

Splatting

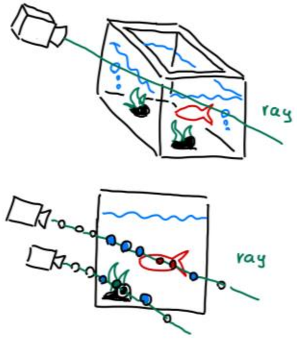
Project volume samples to the image plane
Weight distribution on the neighborhood pixels



Volume Rendering

"The display of data sampled in three dimensions." (L.Westover, 1989)

Domain of rendering algorithms for volumetric scene representations (differently from surface representation such as meshes)



How does Ray Tracing Work in Video Games and Movies? (YouTube)
Ray Tracing – NVIDIA Developer

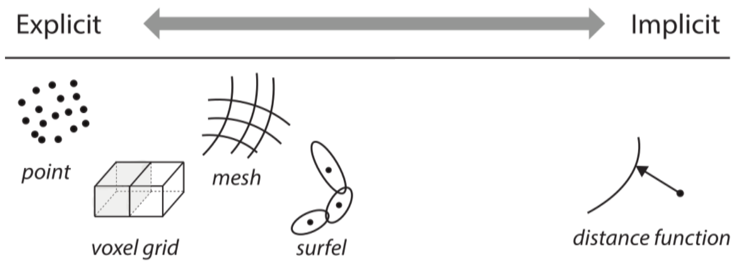
The Rendering Equation, J. T. Kajiya 1986
Interactive Volume Rendering, L. Westover, 1989

The Building Blocks: Representing and Reconstructing Scenes

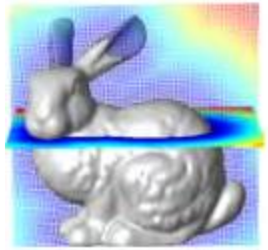
Explicit Representations

Implicit Representations

Point Cloud, Voxels, Meshes



Signed Distance Functions (SDF)



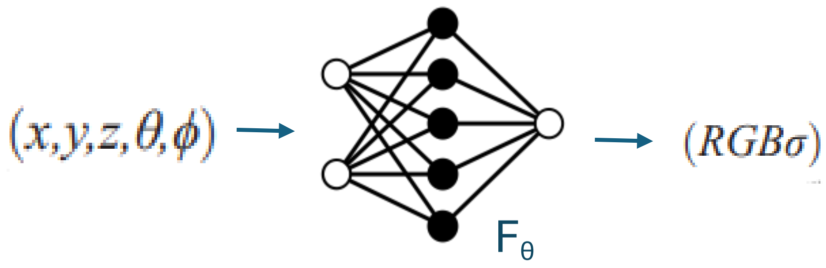
Optimization for 3D reconstruction:

Geometrical constraints

Photometric similarity

3D Gaussians

Neural Radiance Fields



The Metrics: Image Similarity

Pixel-wise similarity

Pixel intensity difference

Exact reconstruction, Noise level

Visual perception, Structural coherence

PSNR

Peak Signal to Noise Ratio

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

Structural similarity

Local image structure

Edges, Texture, Structural Integrity

High level semantics, Visual realism

SSIM

Structural SIMilarity

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha} \cdot [c(\mathbf{x},\mathbf{y})]^{\beta} \cdot [s(\mathbf{x},\mathbf{y})]^{\gamma}$$

$$l(\mathbf{x},\mathbf{y}) = \frac{2(1+R)}{1+(1+R)^2+C_1/\mu_x^2} \cdot$$ luminance

$$c(\mathbf{x},\mathbf{y}) = \frac{2\,\sigma_x\,\sigma_y + C_2}{\sigma_x^2+\sigma_y^2+C_2}$$ contrast

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy}+C_3}{\sigma_x\,\sigma_y+C_3}.$$ structure

C_i : avoid instability R : luminance change

Perceptual similarity

High-level visual features

Visual realism, perceptual similarity

Pixel accuracy (intentional)

LPIPS

Learned Perceptual Image Patch Similarity

Compare images at weighted features level (extracted by pretrained CNNs). Weights are tuned from human judgments

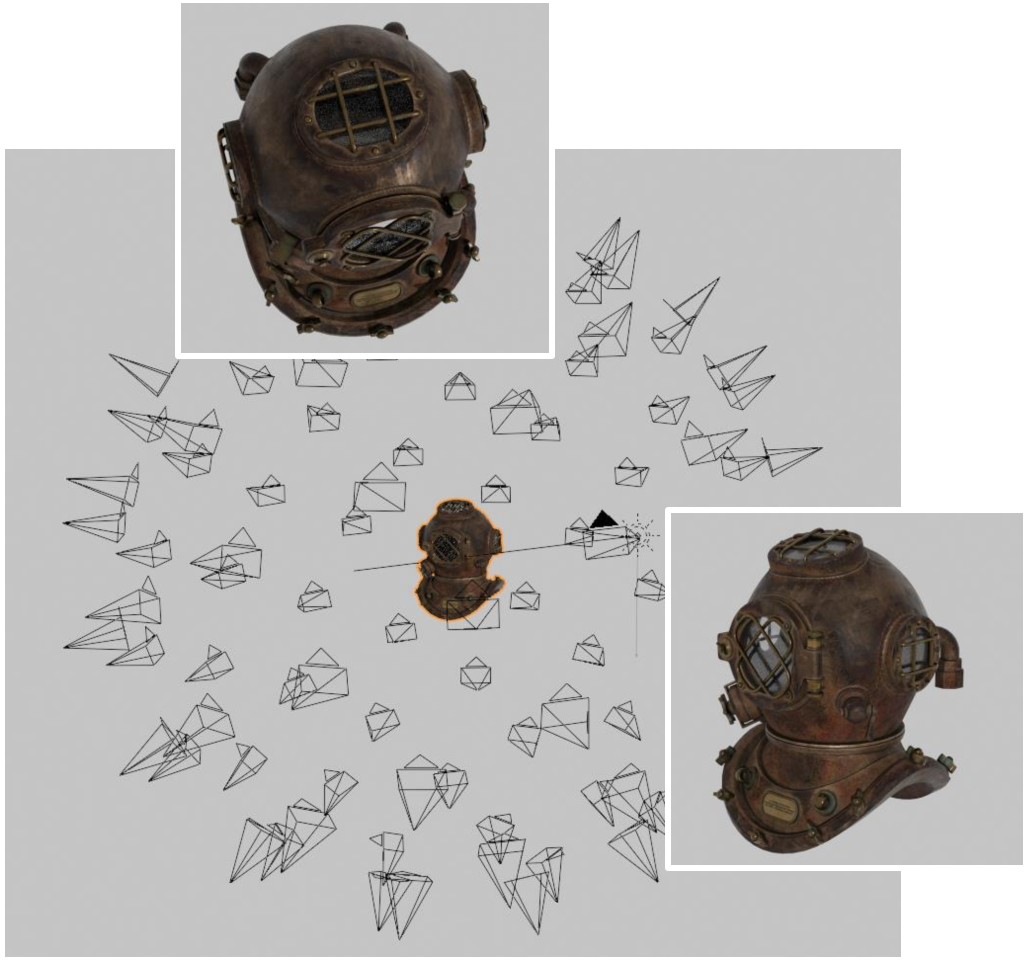
NeRF and 3DGS

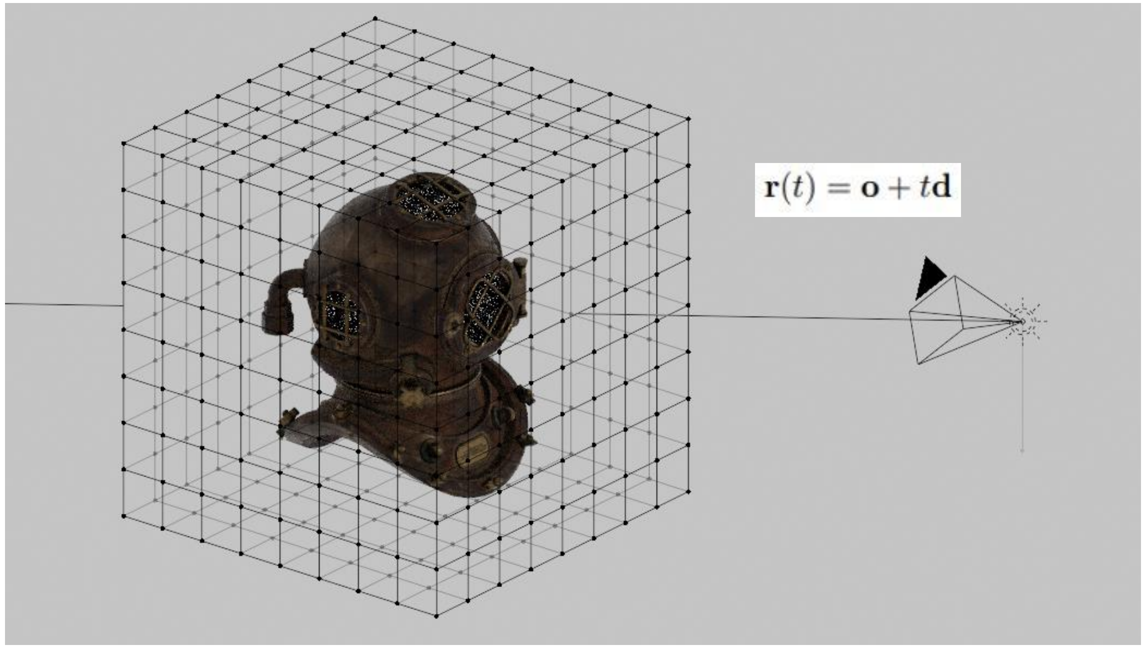
The Vanilla Methods

*The term **vanilla** is derived from the plain, unadorned flavor of vanilla ice cream, a connotation that dates back to its popularity as a universal base in desserts. Within computing, the term emerged as early as the 1980s, popularized in systems and user interfaces to describe default or base states. [wikipedia]*

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis [4]

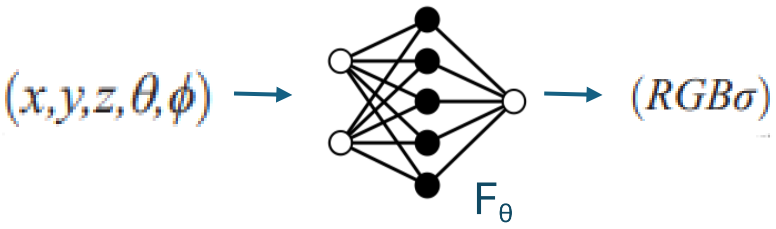
Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng





$(x, y, z, \theta, \phi) \rightarrow F_\theta \rightarrow (RGB\sigma)$

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis



Volumetric Rendering

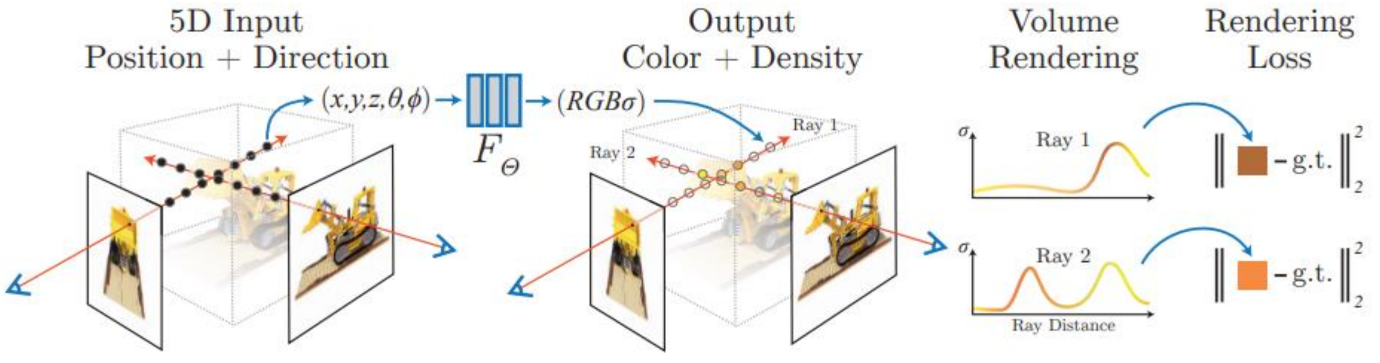
Accumulated transmittance

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \qquad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

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

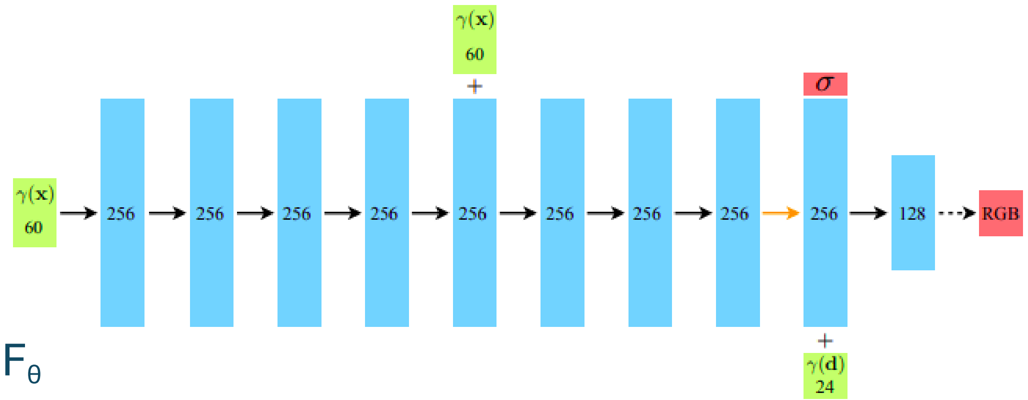
Discretization with quadrature rule

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



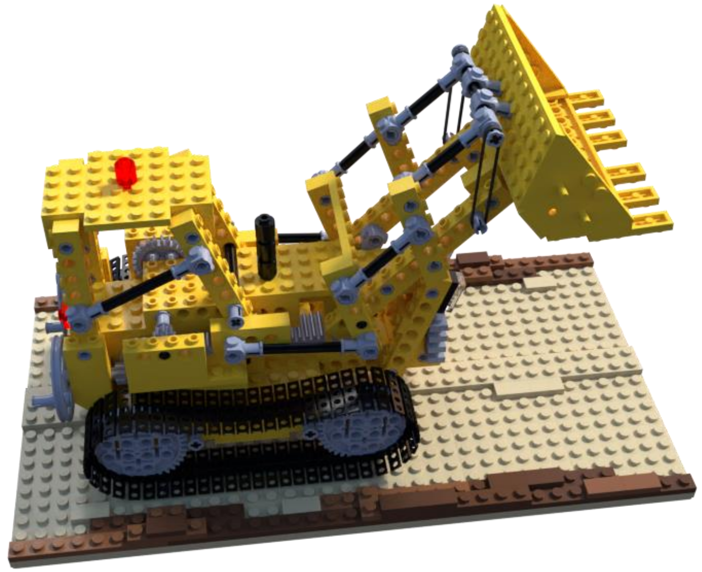
2 Stages MLP:
- **Density**: from position
- **Color**: from view direction + 1st stage features

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



Fθ

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis



Synthetic model

Ground Truth

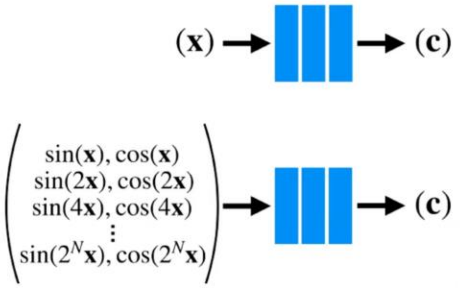
NeRF Rendering

Something missing...

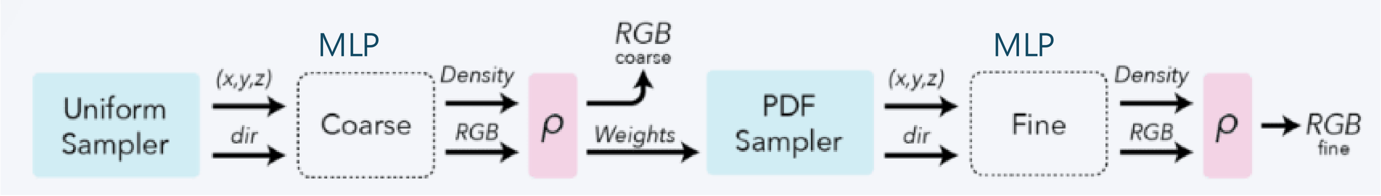
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Positional Encoding

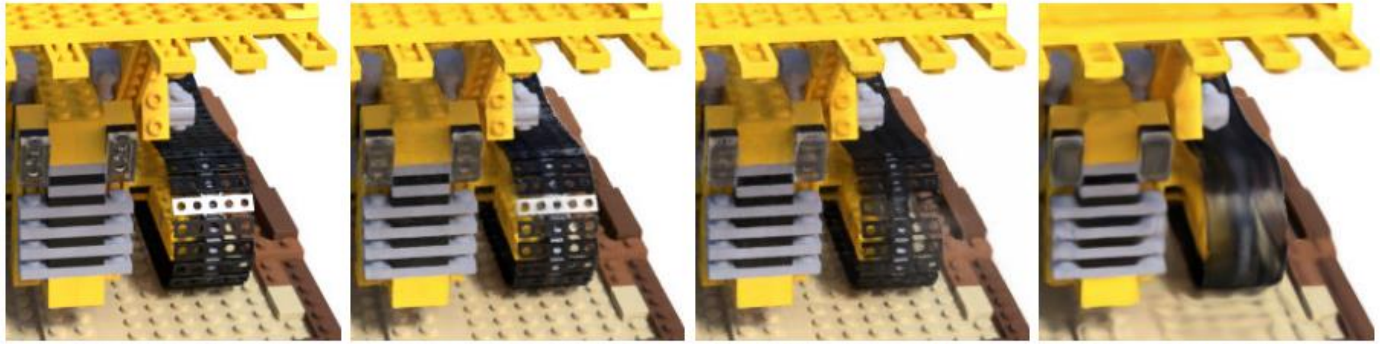
$$\gamma(p) = \left(\sin(2^0 \pi p), \cos(2^0 \pi p), \cdots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p) \right)$$



Hierarchical Sampling

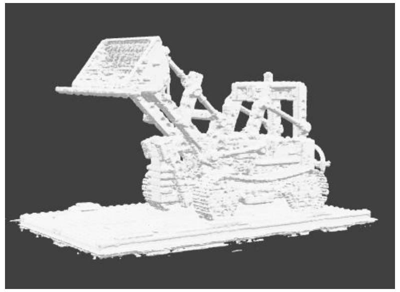


$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$



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

- Overfitted Neural Network
- Implicit scene representation
- View-dependent colors
- Geometric reconstruction



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

PSNR↑

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
SRN [42]	26.96	17.18	20.73	26.81	20.85	18.09	26.85	20.60
NV [24]	28.33	22.58	24.79	30.71	26.08	24.22	27.78	23.93
LLFF [28]	28.72	21.13	21.79	31.41	24.54	20.72	27.48	23.22
Ours	**33.00**	**25.01**	**30.13**	**36.18**	**32.54**	**29.62**	**32.91**	**28.65**

SSIM↑

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
SRN [42]	0.910	0.766	0.849	0.923	0.809	0.808	0.947	0.757
NV [24]	0.916	0.873	0.910	0.944	0.880	0.888	0.946	0.784
LLFF [28]	0.948	0.890	0.896	0.965	0.911	0.890	0.964	0.823
Ours	**0.967**	**0.925**	**0.964**	**0.974**	**0.961**	**0.949**	**0.980**	**0.856**

LPIPS↓

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
SRN [42]	0.106	0.267	0.149	0.100	0.200	0.174	0.063	0.299
NV [24]	0.109	0.214	0.162	0.109	0.175	0.130	0.107	0.276
LLFF [28]	0.064	0.126	0.130	**0.061**	0.110	0.117	0.084	0.218
Ours	**0.046**	**0.091**	**0.044**	0.121	**0.050**	**0.063**	**0.028**	**0.206**

PSNR↑

	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns
SRN [42]	27.29	21.37	18.24	26.63	17.37	24.63	22.87	24.33
LLFF [28]	28.42	22.85	19.52	29.40	18.52	25.46	24.15	24.70
Ours	**32.70**	**25.17**	**20.92**	**31.16**	**20.36**	**27.40**	**26.80**	**27.45**

SSIM↑

	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns
SRN [42]	0.883	0.611	0.520	0.641	0.449	0.738	0.761	0.742
LLFF [28]	0.932	0.753	**0.697**	0.872	0.588	**0.844**	0.857	**0.840**
Ours	**0.948**	**0.792**	0.690	**0.881**	**0.641**	0.827	**0.880**	0.828

LPIPS↓

	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns
SRN [42]	0.240	0.459	0.440	0.453	0.467	0.288	0.298	0.376
LLFF [28]	**0.155**	**0.247**	**0.216**	0.173	**0.313**	**0.174**	**0.222**	**0.193**
Ours	0.178	0.280	0.316	**0.171**	0.321	0.219	0.249	0.268

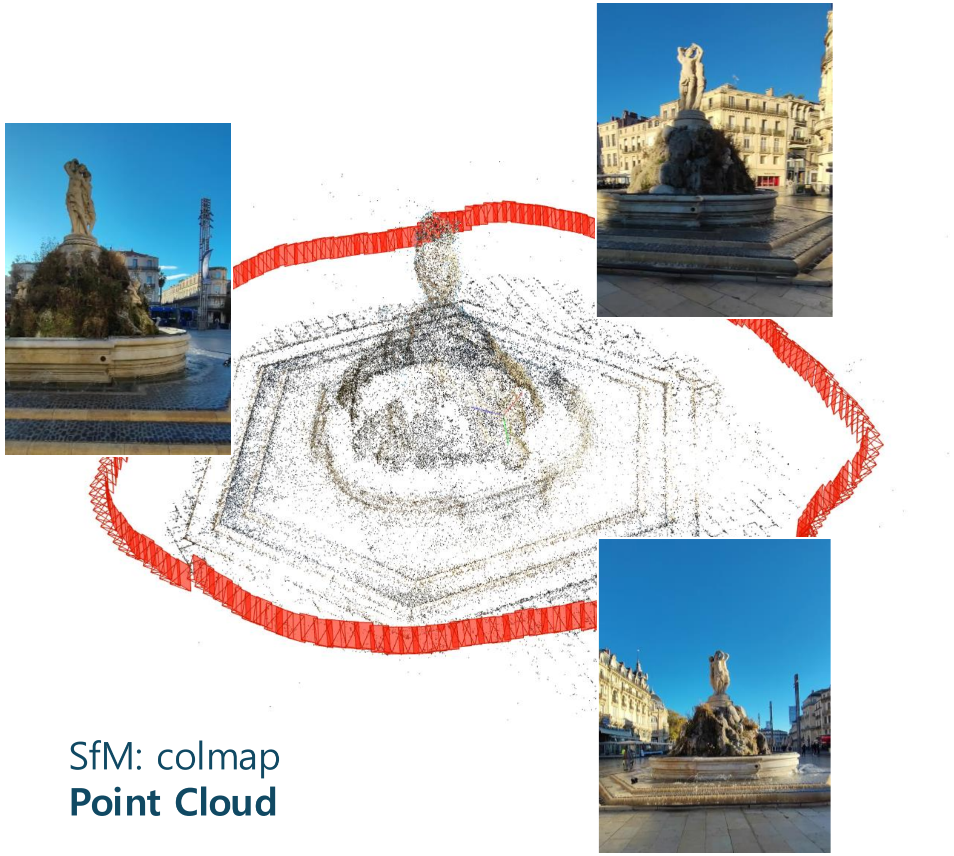
- Model lighter than images
- SOTA view synthesis
- Training time 24-48 hrs
- Novel view at 0.03 fps
- Forward facing
- Synthetic unbounded
- Limited interpretability

3D Gaussian Splatting for Real-Time Radiance Field Rendering [5]

Kerbl Bernhard, Kopanas Georgios, Leimkühler Thomas, Drettakis George

- Real-time rendering?
- Less training time?

- Explicit representation
- New Rasterization algorithm

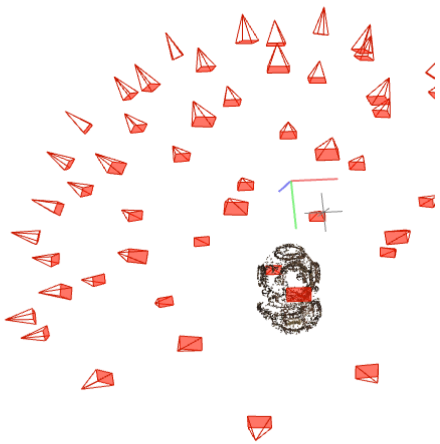


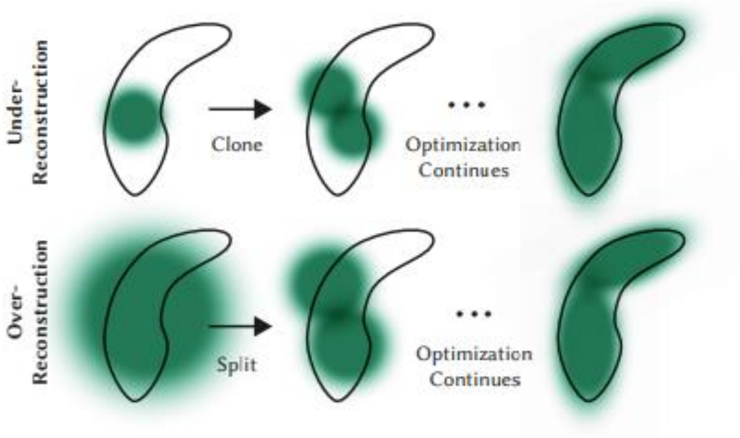
SfM: colmap
Point Cloud

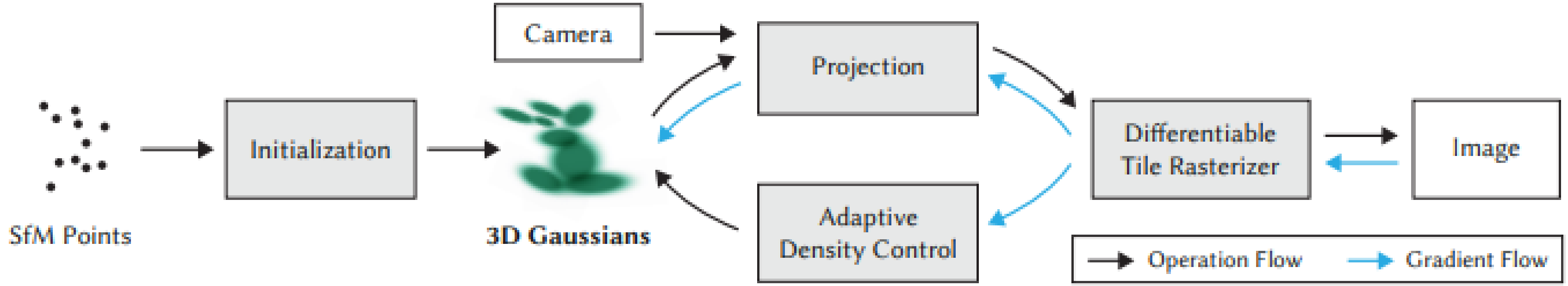
3D Gaussian Splatting for Real-Time Radiance Field Rendering

- Gaussians as primitives
- Encode view dependency in Spherical Harmonics
- No neural component
- GPU accelerated rasterizer
- Gaussian density control

Prune transparent gaussians
Densify in high positional gradient (clone/split)







$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$$

Parameters optimization via SGD

3D Gaussian Splatting for Real-Time Radiance Field Rendering

Ablation Analysis

	Truck-5K	Garden-5K	Bicycle-5K	Truck-30K	Garden-30K	Bicycle-30K	Average-5K	Average-30K
Limited-BW	14.66	22.07	20.77	13.84	22.88	20.87	19.16	19.19
Random Init	16.75	20.90	19.86	18.02	22.19	21.05	19.17	20.42
No-Split	18.31	23.98	22.21	20.59	26.11	25.02	21.50	23.90
No-SH	22.36	25.22	22.88	24.39	26.59	25.08	23.48	25.35
No-Clone	22.29	25.61	22.15	24.82	27.47	25.46	23.35	25.91
Isotropic	22.40	25.49	22.81	23.89	27.00	24.81	23.56	25.23
Full	22.71	25.82	23.18	24.81	27.70	25.65	23.90	26.05

SOTA NeRFs comparison

Dataset	Mip-NeRF360						Tanks&Temples					
Method|Metric	*SSIM*$^\uparrow$	*PSNR*$^\uparrow$	*LPIPS*$^\downarrow$	Train	FPS	Mem	*SSIM*$^\uparrow$	*PSNR*$^\uparrow$	*LPIPS*$^\downarrow$	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

- Real-time rendering 100fps
- Interpretability of the model

- GPU computations (20 GB VRAM)
- Speed up possible with custom CUDA kernels
- No regularization

NeRF (Vanilla) vs 3DGS

Query for every image pixel (pinhole camera)

- Ray Marching
- Implicit representation
- Discard SfM point cloud
- Exhaustive sampling

- View synthesis: 0.03 fps
- Training: 24-48 hrs

- Light memory model (few MBs)

- Expensive GPU usage

Gaussians mean, covariance, alpha and SH

- Rasterize Gaussians
- Explicit Representation
- Use all SfM output
- Efficient sampling

- View synthesis: 120 fps
- "Training": 10-50 mins

- "Heavier" model (few hundred MBs)

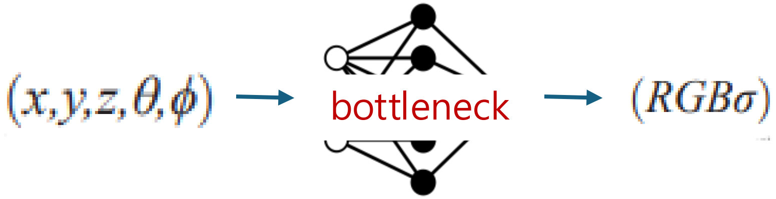
- Extremely Expensive GPU usage

What came next?

Beyond Vanilla's

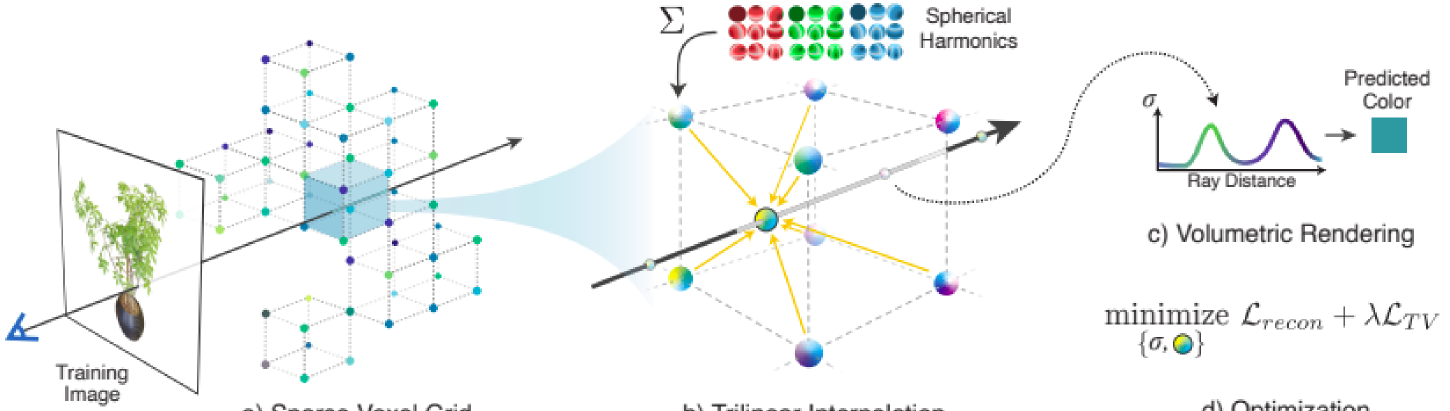
Plenoxels: Radiance Fields without Neural Networks [6]

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, Angjoo Kanazawa



Keep differentiable volume rendering

Different scene representation...



NeRF

Plenoxels

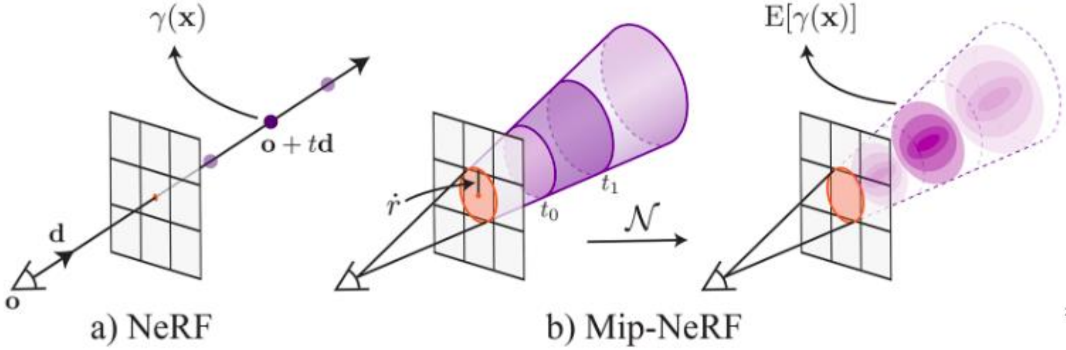
	PSNR ↑	SSIM ↑	LPIPS ↓	Train Time
Ours	31.71	**0.958**	**0.049**	**11 mins**
NV [20]	26.05	0.893	0.160	>1 day
JAXNeRF [7, 26]	**31.85**	0.954	0.072	1.45 days
Ours	26.29	**0.839**	**0.210**	**24 mins**
LLFF [25]	24.13	0.798	0.212	—*
JAXNeRF [7, 26]	**26.71**	0.820	0.235	1.62 days
Ours	20.40	**0.696**	**0.420**	**27 mins**
NeRF++ [57]	**20.49**	0.648	0.478	~4 days

Average on synthetic dataset

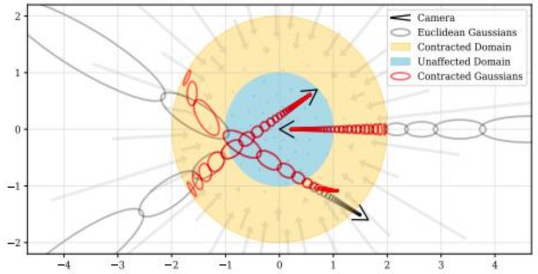
- Optimized voxel distribution:
  prune empty space and densify relevant space
- 15 fps rendering
- Regularization weights are scene dependent
- (No information on the occupied memory)

Mip-NeRF360: Unbounded Anti-Aliased Neural Radiance Fields [7]

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman



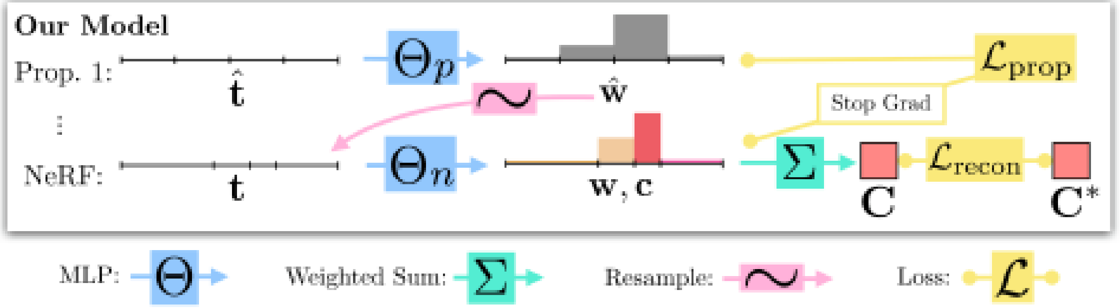
Mip-NeRF addresses resolution limitations of NeRF (anti-aliasing conical frustum to reduce blurriness)



Mip-NeRF360: Extension for unbounded (360) scenes

- Map input space with a warping function
- Small proposal MLP for sampling
- Novel regularizer

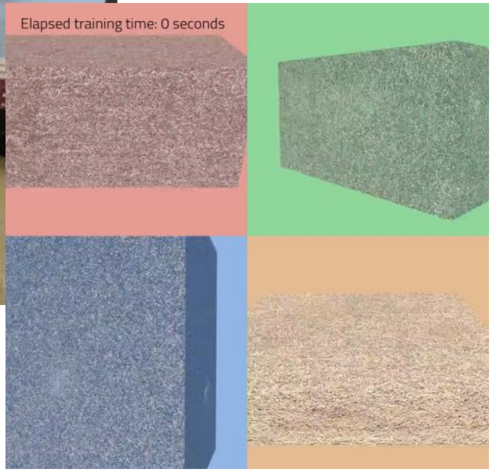
Training requires hours



Instant Neural Graphics Primitives with a Multiresolution Hash Encoding [8]

Thomas Müller, Alex Evans, Christoph Schied, Alexander Keller

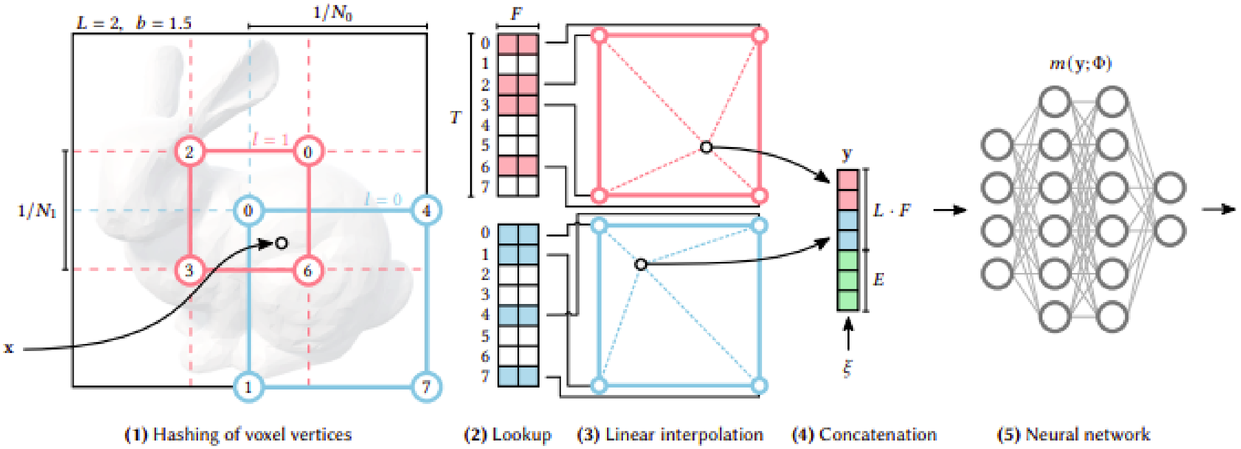
Neural gigapixel images Neural SDF NeRF



NGP = object with shape and appereance retrieved from a Neural Network

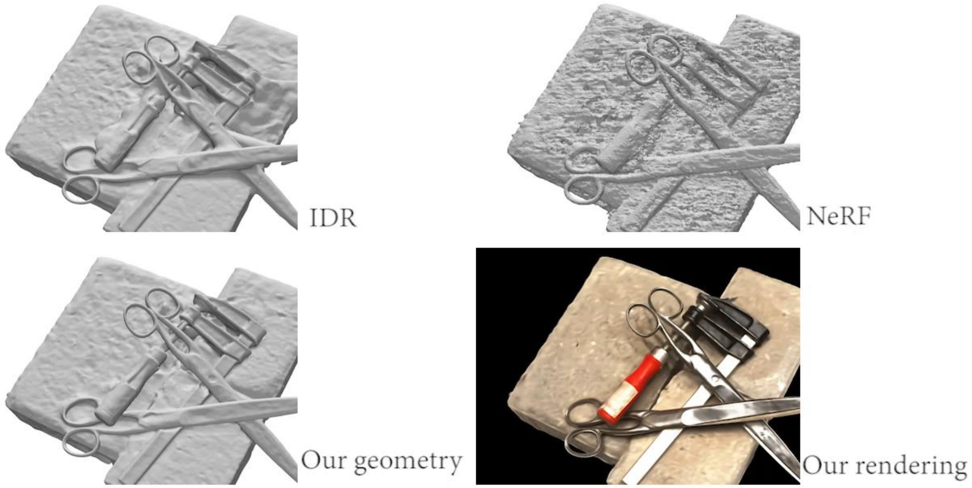
- Dedicated rendering algorithm: smarter space sampling
- Fully-fused NN: smaller MLP
- Encoding strategy: features encoding parameters learned with the network + Multi-resolution hash encoding

Minor artifacts due to hash collisions
Less accurate view-dependency from Mip-NeRF360



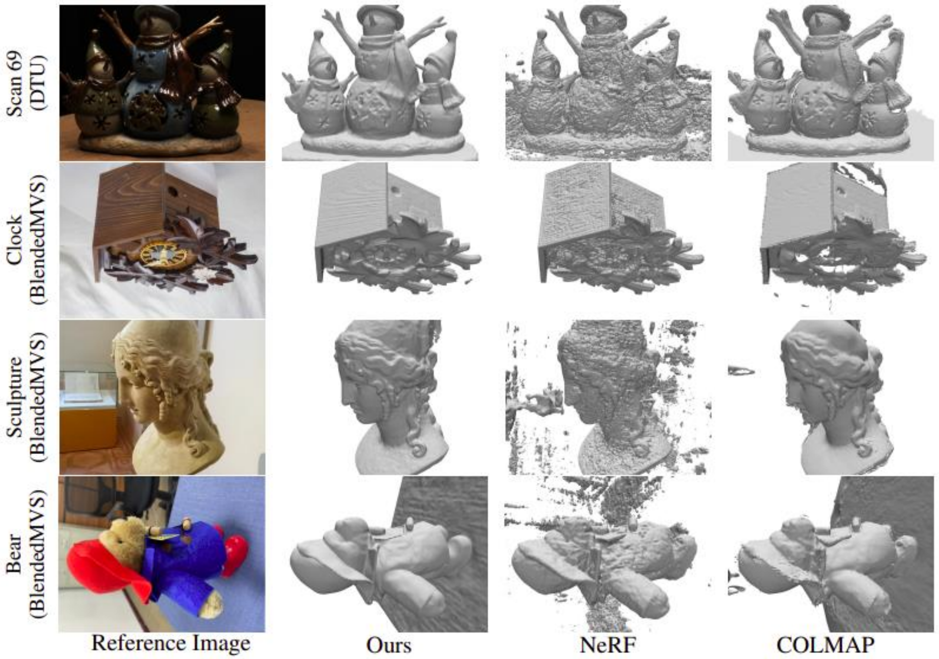
NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction [9]

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, Wenping Wang



$$\text{Chamfer Distance}(P,Q) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|_2^2 + \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|_2^2$$

mutual nearest neighbor distance for point cloud allignment



Interest in accurate surface reconstruction

- Constraint scene surface
- Rendering directly from implicit SDF
- Space as zero-level set of a MLP encoded SDF
- Hierarchical sampling with an s-density function

Long training time: 10-20 hrs

Navigation

NeRF-VINS: A Real-time Neural Radiance Field Map-based Visual-Inertial Navigation System

Saimouli Katragadda[1], Woosik Lee[1], Yuxiang Peng[1], Patrick Geneva[1], Chuchu Chen[1], Chao Guo[2], Mingyang Li[2], and Guoquan Huang[1]

Cultural Heritage

The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W8-2024
8th International ISPRS Workshop LowCost 3D - Sensors, Algorithms, Applications, 12–13 December 2024, Brescia, Italy

3D representation of Architectural Heritage: a comparative analysis of NeRF, Gaussian Splatting, and SfM-MVS reconstructions using low-cost sensors

Paolo Clini [1], Romina Nespeca [1], Renato Angeloni [1], Laura Coppetta [1]

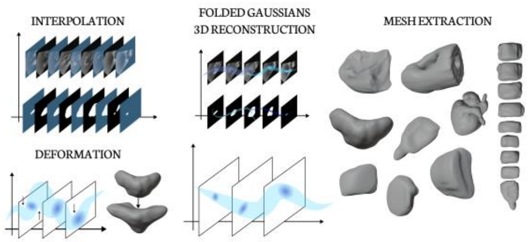
[1] Dept. of Civil and Building Engineering and Architecture, Università Politecnica della Marche, Ancona, Italy – (p.clini, r.nespeca, r.angeloni)@univpm.it, l.coppetta@pm.univpm.it

Keywords: Architectural Heritage, 3D reconstruction, NeRF, Gaussian Splatting, SfM-MVS, Low-cost sensors.

Medicine

MedGS: Gaussian Splatting for Multi-Modal 3D Medical Imaging

Kacper Marzol, Jagiellonian University
Ignacy Kolton, Jagiellonian University
Weronika Smolak-Dyżewska, Jagiellonian University
Joanna Kaleta, Warsaw University of Technology, Sano Centre for Computational Medicine
Marcin Mazur, Jagiellonian University
Przemysław Spurek, Jagiellonian University, IDEAS Research Institute
przemyslaw.spurek@uj.edu.pl



4D Representation (time)

4D Gaussian Splatting for Real-Time Dynamic Scene Rendering

Guanjun Wu[1]*, Taoran Yi[2]*, Jiemin Fang[3]†, Lingxi Xie[3], Xiaopeng Zhang[3], Wei Wei[1], Wenyu Liu[2], Qi Tian[3], Xinggang Wang[2]†‡
[1]School of CS, Huazhong University of Science and Technology
[2]School of EIC, Huazhong University of Science and Technology [3]Huawei Inc.
{guajuwu, taoranyi, weiw, liuwy, xgwang}@hust.edu.cn
{jaminfong, 198808xc, zxphistory}@gmail.com tian.qi1@huawei.com

What is going on recently?

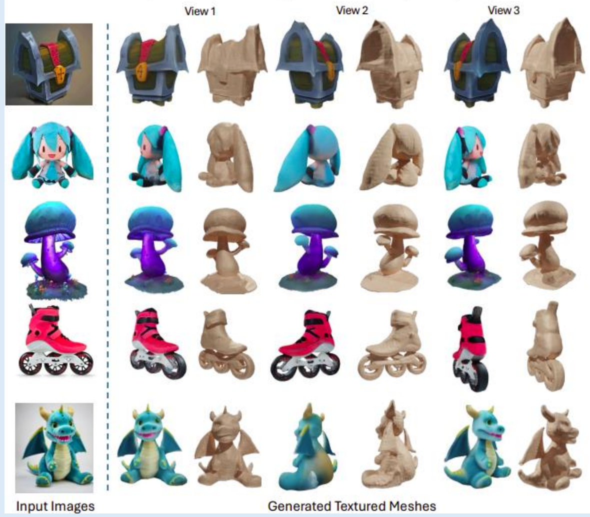
CVPR 2024
129 papers focus on this...
71 on NeRF + 58 on 3DGS

The Explosion of Radiance Fields

3D Generation

GECO: Generative Image-to-3D within a SECOnd

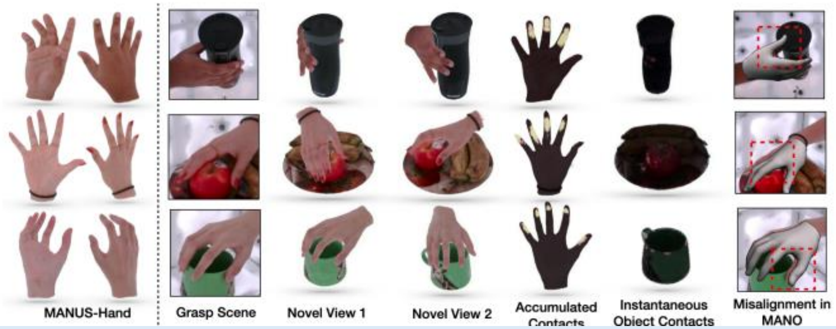
Chen Wang[1] Jiatao Gu[2] Xiaoxiao Long[3] Yuan Liu[3] Lingjie Liu[1]
[1]University of Pennsylvania [2]Apple [3]The University of Hong Kong



Manipulation

MANUS: Markerless Grasp Capture using Articulated 3D Gaussians

Chandradeep Pokhariya[1]* Ishaan Nikhil Shah[1]** Angela Xing[2]** Zekun Li[2]
Kefan Chen[2] Avinash Sharma[1] Srinath Sridhar[2]
[1]IIIT Hyderabad [2]Brown University
ivl.cs.brown.edu/research/manus



3D Segmentation

Segment Anything in 3D with Radiance Fields

Jiazhong Cen, Jiemin Fang, Zanwei Zhou, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian, *Fellow, IEEE*

Semantics

A Neural Representation Framework with LLM-Driven Spatial Reasoning for Open-Vocabulary 3D Visual Grounding

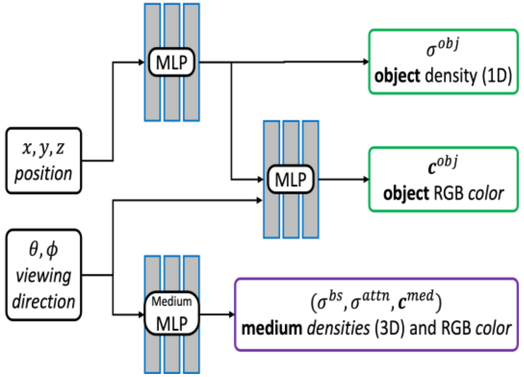
Zhenyang Liu, lzyzjhz@163.com, Fudan University, Shanghai Innovation Institute, Shanghai, China
Sixiao Zheng, sxzheng18@fudan.edu.cn, Fudan University, Shanghai Innovation Institute, Shanghai, China
Siyu Chen, siyu_chen279@163.com, Zhejiang University, Hangzhou, Zhejiang, China
Cairong Zhao, zhaocairong@tongji.edu.cn, Tongji University, Shanghai, China
Longfei Liang, longfei.liang@neuhelium.com, NeuHelium Co., Ltd, Shanghai, China
Xiangyang Xue*, xyxue@fudan.edu.cn, Fudan University, Shanghai, China
Yanwei Fu*, yanweifu@fudan.edu.cn, Fudan University, Shanghai Innovation Institute, Shanghai, China

What is Interesting for us?

Underwater Vision and Robotics Applications

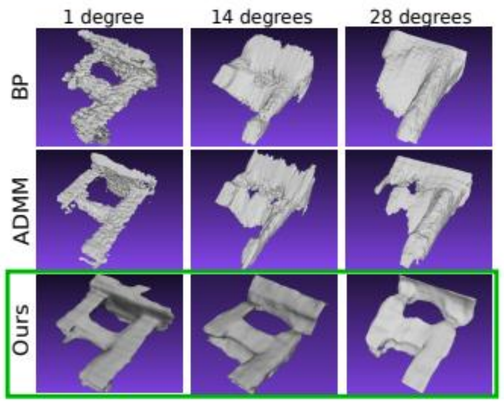
Underwater NeRF

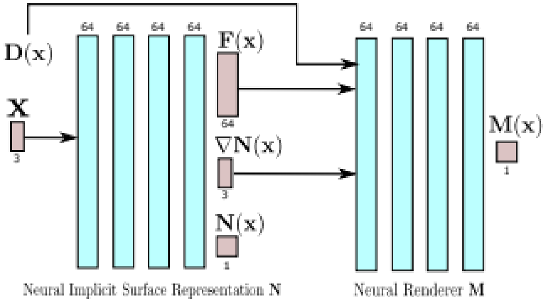
SeatThru-NeRF [10]



Include medium in the rendering
Extreme GPU memory allocation
Long training (10 hrs on A100)

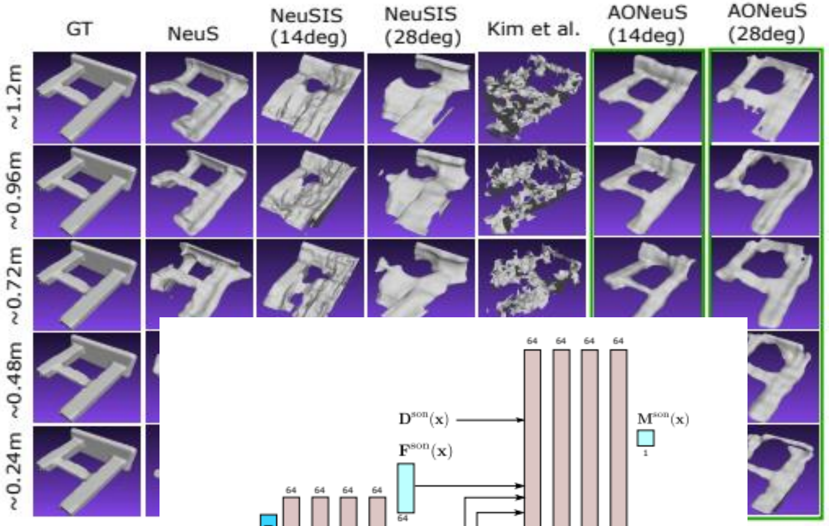
Neusis [11]

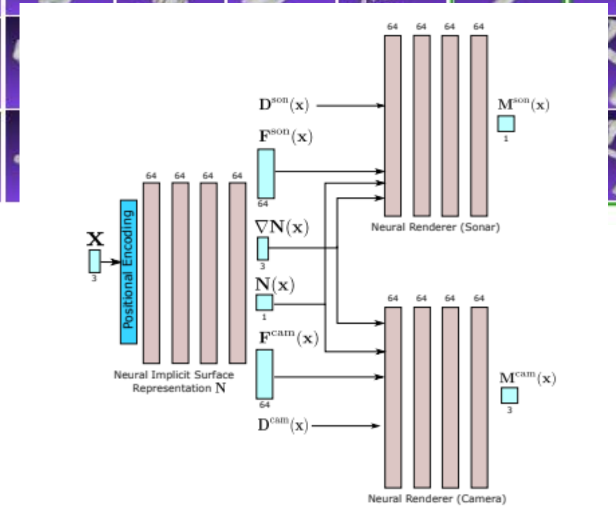




Acoustic differentiable volume renderer
Surfaces as zero-level sets of SDF
Learn two MLP
6 hrs on RTX 3090
Single object reconstruction

AONeuS [12]





Neusis + optical camera
For limited sensor baseline
Learn two MLP
No backscatter model
30 mins reconstruction

Underwater NeRF

NeuRSS: Enhancing AUV Localization and Bathymetric Mapping With Neural Rendering for Sidescan SLAM

Yiping Xie, Jun Zhang, Nils Bore, and John Folkesson, *Senior Member, IEEE*

WaterHE-NeRF: Water-ray Tracing Neural Radiance Fields for Underwater Scene Reconstruction

Jingchun Zhou[1], Tianyu Liang[1], Dehuan Zhang[1], Zongxin He[1]
[1] Dalian Maritime University

AquaNeRF: Neural Radiance Fields in Underwater Media with Distractor Removal

Luca Gough, Adrian Azzarelli, Fan Zhang, Nantheera Anantrasirichai
Visual Information Laboratory, University of Bristol, Bristol, UK

Bathymetric Surveying With Imaging Sonar Using Neural Volume Rendering

Yiping Xie, Giancarlo Troni, *Member, IEEE*, Nils Bore, and John Folkesson, *Senior Member, IEEE*

NFFLS: Rapid and Accurate Underwater 3-D Reconstruction With Neural Fields for Forward-Looking Sonar

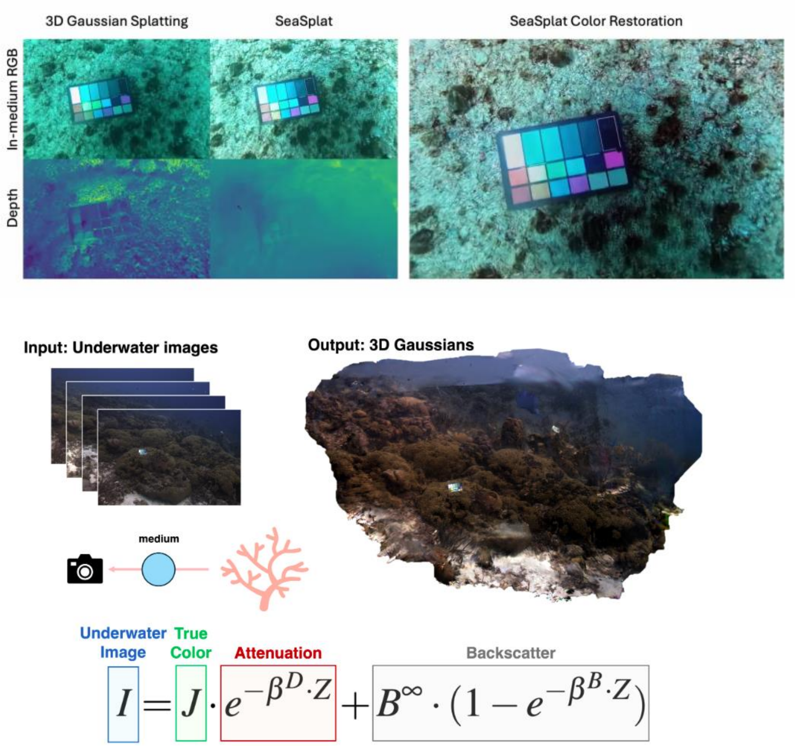
Cao Huang, Hongyu Yang, Jinchang Ren, *Senior Member, IEEE*, and Yulong Ji

UW-SDF: Exploiting Hybrid Geometric Priors for Neural SDF Reconstruction from Underwater Multi-view Monocular Images

Zeyu Chen[1], Jingyi Tang[1,2], Gu Wang[3], Shengquan Li[2], Xinghui Li[1], Xiangyang Ji[4], and Xiu Li[1,2,*]

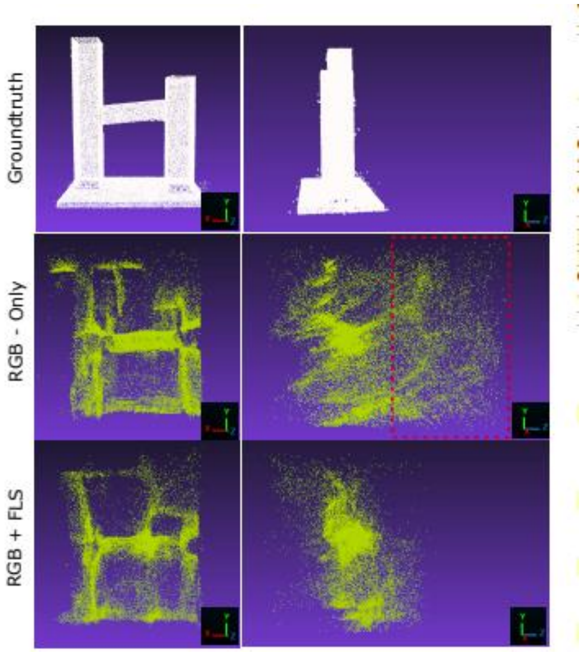
Underwater 3DGS

SeaSplat [13]



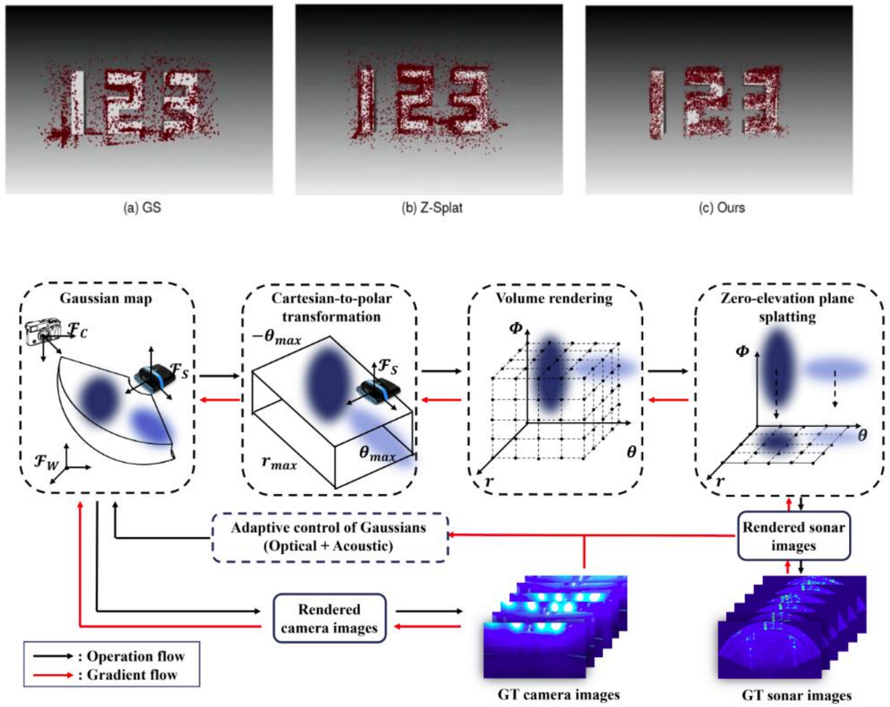
3DGS with underwater model
Additional loss constraint
No view dependency
No lights or dynamic scenes

Z-Splat [14]



Opti-Acoustic, sonar for missing cone problem in limited baseline
Sonar splatted on Z-axis
Approximated sonar model

Aqua-Splat [15]



Physically consistent model and reconstruction
Initialization rely on SfM (robust?)
Opti-Acoustic opacity relation unclear

Underwater 3DGS

UW-GS: Distractor-Aware 3D Gaussian Splatting for Enhanced Underwater Scene Reconstruction

Haoran Wang, Nantheera Anantrasirichai, Fan Zhang, and David Bull
School of Computer Science, University of Bristol, Bristol, UK
{yp22378,n.anantrasirichai,fan.zhang,dave.bull}@bristol.ac.uk

WaterSplatting: Fast Underwater 3D Scene Reconstruction Using Gaussian Splatting

Huapeng Li[1], Wenxuan Song[2], Tianao Xu[2], Alexandre Elsig[2], Jonas Kulhanek[3,2]
[1] University of Zurich; [2] ETH Zurich, [3] CTU in Prague

RecGS: Removing Water Caustic with Recurrent Gaussian Splatting

Tianyi Zhang[1], Weiming Zhi[1], Kaining Huang[1],
Joshua Mangelson[3], Corina Barbalata[2], Matthew Johnson-Roberson[1]

3D-UIR: 3D Gaussian for Underwater 3D Scene Reconstruction via Physics-Based Appearance-Medium Decoupling

Jieyu Yuan, Yujun Li, Yuanlin Zhang, Chunle Guo, Xiongxin Tang,
Ruixing Wang, and Chongyi Li, *Senior Member, IEEE*

SonarSplat: Novel View Synthesis of Imaging Sonar via Gaussian Splatting

Advaith V. Sethuraman, Max Rucker, Onur Bagoren, Pou-Chun Kung, Nibarkavi N.B. Amutha,
and Katherine A. Skinner

The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W10-2025
3D Underwater Mapping from Above and Below – 3rd International Workshop, 8–11 July 2025, TU Wien, Vienna, Austria

3D Reconstruction of Underwater Shipwrecks: 3D Gaussian Splatting and Structure from Motion for the Melania shipwreck

Dario Billi [1], Valeria Croce [2], Andrea Piemonte [1], Gabriella Caroti [1]

[1] Department of Civil and Industrial Engineering, ASTRO Laboratory, University of Pisa, L.go Lucio Lazzarino, 56122 Pisa, Italy;
dario.billi@phd.unipi.it; andrea.piemonte@ing.unipi.it, gabriella.caroti@ing.unipi.it
[2] LISPEN EA7515, Arts et Métiers Institute of Technology, 2 Cours des Arts et Métiers, 13617 Aix-en-Provence, France;
valeria.croce@ensam.eu

Keywords: Underwater, Photogrammetry, 3D-Gaussian Splatting, Melania shipwreck, Artificial Intelligence, Agisoft Metashape

Spatiotemporal Degradation-Aware 3D Gaussian Splatting for Realistic Underwater Scene Reconstruction

Shaohua Liu*
School of Astronautics
Beihang University
Beijing, China
liushaohua@buaa.edu.cn

Ning Gao
School of Astronautics
Beihang University
Beijing, China
gaoning_ai@buaa.edu.cn

Zuoya Gu
School of Astronautics
Beihang University
Beijing, China
zuoyagu@buaa.edu.cn

Hongkun Dou
School of Astronautics
Beihang University
Beijing, China
douhk@buaa.edu.cn

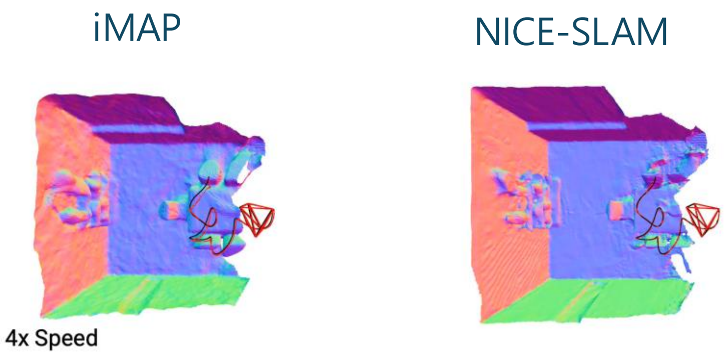
Yue Deng
School of Artificial Intelligence
Beihang University
Beijing, China
Zhongguancun Academy
Beijing, China
ydeng@buaa.edu.cn

Hongjue Li†‡§
Beihang University
School of Astronautics
Beijing, China
lihongjue@buaa.edu.cn

Robotics applications: (Semantic) SLAM

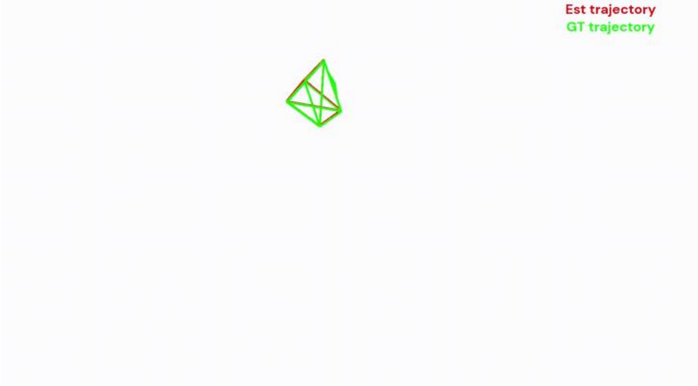
NeRF

Nice-SLAM [16]



Monocular RGB-D
Tiny MLPs at different res
No loop closures

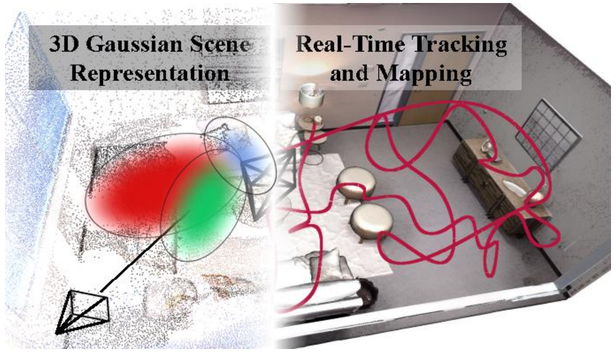
SNI-SLAM [17]



Monocular RGB-D
Attention based feature fusion
Pretrained DINOv2
Extreme GPU usage

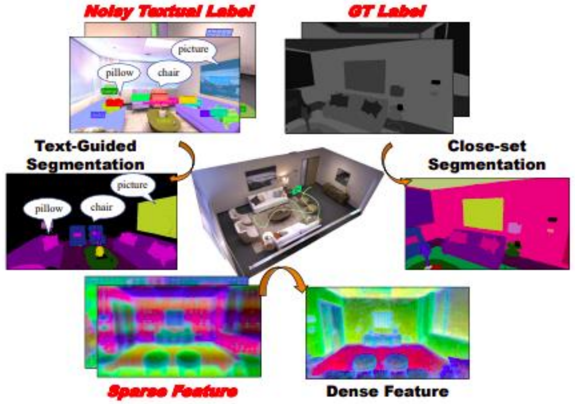
3DGS

GS-SLAM [18]



Smart keyframing
Gaussian shape regularization
No loop closure

GSFF-SLAM [19]



Semantic gaussians
Decoupled scene and semantic optimization
Offline semantic

Final Comments

- Advances comes from:
  - Alternative scene representations (smaller MLP, voxels, 3D Gaussians)
  - **Better usage of GPU** with tailored algorithms (custom CUDA kernels)
  - Optimal rendering that sample the space in a smarter way
  - **Different network inputs** feature encoding (e.g. multi resolution hash encoding)
  - Better **loss regularization**
- Multimodal approaches for better performances
- Online applications are possible
- Improvements with physical and classical multi-view geometry constraints

- GPU Requirements (High VRAM or multiple-GPU parallelization)

Further Resources

Books' Chapters

- Foundations of Computer Vision, chapter 45: Radiance Fields (MIT Press, 2024)
- SLAM Handbook, chapter 14: Map Representations with Differentiable Volume Rendering (Cambridge University Press, 2025)

Blog Posts

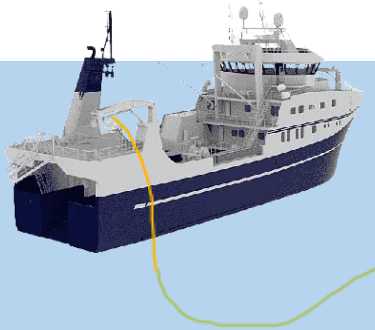
- 3D Gaussian Splatting Introduction – Paper Explanation & Training on Custom Datasets with NeRF Studio Gsplats
- The Annotated NeRF – Training on Custom Dataset from Scratch in Pytorch

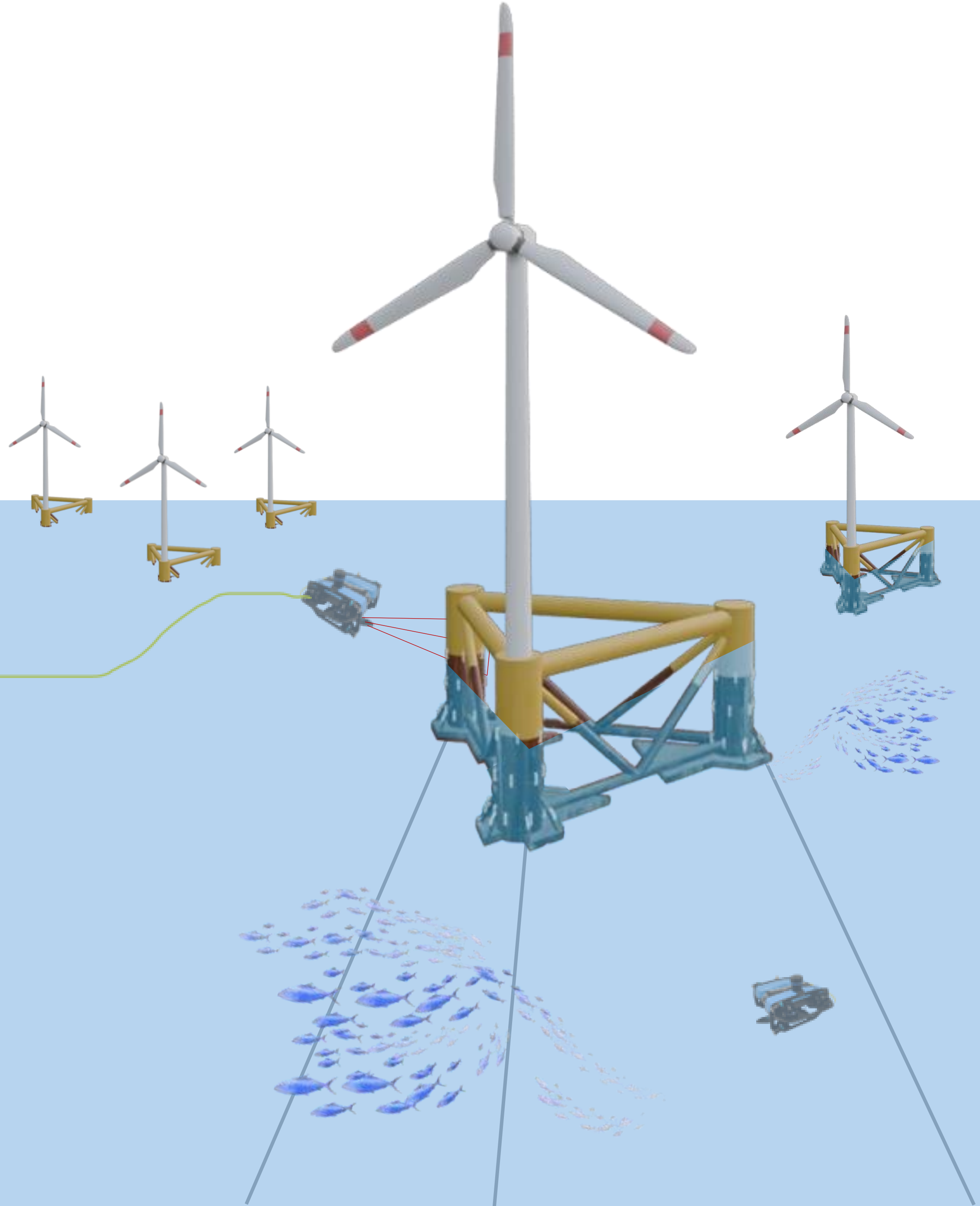
YouTube Videos

- NeRF and Gaussian Splatting – easily explained
- Short course on Radiance Fields in Visual Computing and Artificial Intelligence

"The sea, once it casts its spell, holds one in its net of wonder forever."

Jacques Yves Cousteau





20/01/2026

PhD Updates

Thank you for your attention!
Discussion and Questions

Alessandro Puglisi

PhD Candidate in Marine Robotics, RSM, LIRMM

alessandro.puglisi@lirmm.fr

References

[1] Grimaldi M., Cieslak P., Ochoa E., Bharti V., Rajani H., Carlucho I., Koskinopoulou M., Petillot Y., Gracias N., **Stonefish: Supporting Machine Learning Research in Marine Robotics**, 2025

[2] Johannes L. Schonberger, Jan-Michael Frahm, **Structure-from-Motion Revisited**, CVPR 2016

[3] **Nerfstudio: A Modular Framework for Neural Radiance Field Development**, ACM SIGGRAPH 2023 Conference Proceedings, 2023

[4] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, **NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis**, 2020

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, **3D Gaussian Splatting for Real-Time Radiance Field Rendering**, SIGGRAPH 2023

[6] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, Angjoo Kanazawa, **Plenoxels: Radiance Fields without Neural Networks**, CVPR 2022

[7] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, **Mip-NeRF360: Unbounded Anti-Aliased Neural Radiance Fields,** CVPR 2022

[8] Thomas Müller, Alex Evans, Christoph Schied, Alexander Keller, **Instant Neural Graphics Primitives with a Multiresolution Hash Encoding**, SIGGRAPH 2022

[9] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, Wenping Wang, **NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction**, NeurIPS 2021

[10] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, Tali Treibitz, **SeaThru-NeRF: Neural Radiance Fields in Scattering Media, CVPR 2023**

[11] Mohamad Qadri, Michael Kaess, Ioannis Gkioulekas, **Neural Implicit Surface Reconstruction using Imaging Sonar,** 2022

[12] Mohamad Qadri, Kevin Zhang, Akshay Hinduja, Michael Kaess, Adithya Pediredla, Christopher Metzler, **AONeuS: A Neural Rendering Framework for Acoustic-Optical Sensor Fusion,** SIGGRAPH 2024

[13] Daniel Yang, John J. Leonard, Yogesh Girdhar, **SeaSplat: Representing Underwater Scenes with 3D Gaussian Splatting and a Physically Grounded Image Formation Model,** ICRA 2025

[14] Ziyuan Qu, Omkar Vengurlekar, Mohamad Qadri, Kevin Zhang, Michael Kaess, Christopher Metzler, **Z-Splat: Z-Axis Gaussian Splatting for Camera-Sonar Fusion,** *IEEE Transactions on Pattern Analysis and Machine Intelligence 2025*

[15] Zijie Ling, Yunxuan Feng, Ao Meng, Renxiang Xiao, Shu Pan, Wenjie Lu, **Aqua-Splat: Physically-Informed Sonar-Camera Gaussian Splatting for Underwater 3D Reconstruction,** *IEEE Robotics and Automation Letters, 2025*

[16] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, Marc Pollefeys, **Neural Implicit Scalable Encoding for SLAM**, CVPR 2022

[17] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, Hesheng Wang, **SNI-SLAM: Semantic Neural Implicit SLAM**, 2024

[18] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, Andrew J. Davison, **Gaussian Splatting SLAM**, CVPR 2024

[19] Zuxing Lu, Xin Yuan, Shaowen Yang, Jingyu Liu, Changyin Sun, **GSFF-SLAM: 3D Semantic Gaussian Splatting SLAM via Feature Field,** Expert Systems with Applications 2026

[20] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, **ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM**, in *IEEE Transactions on Robotics*, 2021

The new 3D trend

Radiance Fields Techniques

Slides Notes

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng

Introduction

- Build on top of Volume rendering techniques
- Represent the scene as a Radiance Field, stored in a MLP Network
- Simple pixel color loss over model view rendering. Squared error
- Smart network architecture. Encourage multi-view consistency restricting the density to be learned just from the location x, while the RGB color from both location and view direction
- Space sampling in the radiance field based on stratified sampling discretized by quadrature rule (Ray integration)
- Tricks that make this works:
  - Positional Encoding: higher dimensional input space allows to capture higher frequency representations
  - Hierarchical Sampling: more efficient ray marching approach around high density space
    (This allocate MLP capacity towards space with visible scene content)
- The Hierarchical sampling is achieved through a parallel optimization of a coarse and a fine MLP, both considered in the Loss function. Training the coarse make the sampling of the fine better, even if the fince MLP output will be the one used as model output for view syntesis
- Requires a set of images with known camera poses to learn the scene (subset of the colmap output e.g)

- MLP requires less memory than the images itself to represent the scene
- Overfitted MLP on a single scene
- Extremely long time to learn the scene (2 days)
- Not real-time, each new view requires millions of network queries (reduced by hierarchical sampling), 0.03 fps at high res
- No Interpretability of the MLP stored radiance field
- Limited performance on unbounded scenes (works well in forward-facing views or synthetic no background scenes)

- Get a better look on the comparison metrics.. No idea of the range of it, how good they are? Min-max of that ?

3D Gaussian Splatting for Real-Time Radiance Field Rendering

Kerbl Bernhard, Kopanas Georgios, Leimkühler Thomas, Drettakis George

Introduction

- Solves the real-time display problem
- SOTA visual quality and limited training time, with 100 fps novel-view synthesis at 1080p resolution
- Three main contribution allow this result:
  - Usage of sparse point cloud as initialization fo 3D Gaussians position (No empty space computation)
  - Optimization and Density control over the 3D Gaussians to achieve accurate scene representation
  - Fast rendering algorithm with anisotropic gaussian splatting accelerating training and view synthesis
- 3D Gaussians preserves desirable properties of continuous volumetric radiance fields
- Results quality comparable to Mip-NeRF360 (SOTA quality with 48hrs of training required), with less than 1hr of training
- Real-time rendering is possible thanks to the fast GPU sorting algorithms for tile-based rasterization, with anisotropic splatting which respects visibility ordering
- The scene is represented in a set of 3D Gaussians that stores position, covariance, opacity alpha and Spherical harmonics SH, enabling complex geometric shape representation thanks to gaussians anisotropy and view-dependent colors by SH
- Gaussian's parameters gradients is computed explicitly to reduce the automatic differentiation computation during training
- The learning process is based on SGD parameters optimization interleaved with Gaussians density control. Custom CUDA kernels are defined for tailored optimization. The fast rasterizer is a key component of the optimization speed up, since novel views to compute the loss are rendered at higher rates than classical ray marching rendering techniques.
- Adaptive Control of Gaussians regulate their density over unit volume allocating most of the gaussians where detailed geometry needs to be captured, pruning or cloning them. Pruning based on the negligible transparency (alpha) gaussians. Densification instead under-reconstruction or over-reconstruction occurs, which happens where large positional gradient is computed (where the optimization try to move significantly the gaussians because of bad reconstruction)

3D Gaussian Splatting for Real-Time Radiance Field Rendering

Kerbl Bernhard, Kopanas Georgios, Leimkühler Thomas, Drettakis George

Introduction

- For under reconstruction areas, the Gaussians are cloned and translated based on the posiitonal gradient
- In over reconstructed areas the Gaussian is splitte and reduced in size, initializing the new position based on the PDF of the original big gaussian
- The ratserizer is not scene-specific tuned, it is general by an efficient tile-based approach and approximated alpha blending.

- Artifact presents in view-dependent appearance regions
- No regularization. It can help to reduce artifacts
- Very intense GPU usage (20 GB of VRAM at peak usage). Since most of the code is based in PyTorch … Speedup could be obtained by porting the optimization on CUDA kernels

Plenoxels: Radiance Fields without Neural Networks

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, Angjoo Kanazawa

Introduction

- Optimized for fast leraning (100x speedup)
- Highlight that the key of NeRF is not the NN but the diferentiable volume renderer
- Uses an explicit representation of the scene by Plenoxels (voxels with plenoptic informations)
- Plenoxels stores scalar density + vector of spherical harmonic (SH) coefficients of order 2
- Uses the same volume rendering equation presented for NeRF (here with constant step sampling)
- The discretization is made continuous by trilinear interpolation when casting the ray
- Optimization is performed directly on the voxel grid, using RMSProp optimizer (anotehr variant of gradient descent)
- The loss is a combination of the chromatic MSE and a Total Variation Regularization. The Regularization mainly help when few views are available in the dataset.
- Furthermore, optimal results are reached with a coarse to sparse approach, that optimizes voxel distribution by pruning empty space and densifying relevant space (subdividing voxels). Proper pruning is obtained by morphological opration to avoid the loss of relevant voxels for trilinear interpolation
- Techniques of other volume rendering research are introduced to reconstruct unbounded scenes and 360 views
- PSNR, SSIM, LPIPS indices show similar or even better performances than NeRF and other volume rendering methods.

- This is possible thanks to computational capacity to solve complex high dimensional non-linear optimization problems
- Some artifacts are still present and it is not yet real-time (approx 15 fps).
- Regularization terms are added for specific scene (forward looking, 360), and the regularization weight is also scene dependent. Fine tuning is required to converge to good result.
- A better rendering time could be obtained by different data structures (such as octrees)

Mip-NeRF360: Unbounded Anti-Aliased Neural Radiance Fields

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman

Introduction

- Extension of Mip-NeRF to account for unbounded scene and solve other limitations
- Address the following problems:
  - Unbounded representations : solved by mapping the input space through a warping function
  - Efficiency: big MLP to encode a big scene? Uses a small proposal MLP
  - Ambiguity: Artifacts on 360 views, solved with a novel regularizer
- Mip-NeRF extended on top of NeRF using an anti-aliasing conical frustum instead of rays, reducing bluryness.
- To improve efficiency and speed up training, it doesn't waste the computation on the coarse reconstruction. It trains a roposal MLP and a NeRF MLP. where the proposal Network produce just proposal weight for the final NeRF MLP.

- It impressively increases performances
- The training time is still in the order of hours

Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

Thomas Müller, Alex Evans, Christoph Schied, Alexander Keller

Introduction

- Neural Graphic Primitives: Object whose shape and appearance can be retrieved by a NN query (2D Image, SDF, NeRF)
- Built on top of 3 pillars to improve current SOTA NeRF models, multiplicatively speeding up by 1000x:
  - Dedicated Rendering algorithms: The rednering pipeline can be improved to be much faster (10-100x) with a smarter space sampling
  - Smaller NN: Fully-Fused NN are smaller MLP that allow faster computations (10x)
  - Good Encodings: Instead of using a fixed positional encoding, the parameters of features encoding are learned with the network weighs. Furthermore a Multi-resolution has encodings allow optimal GPU usage
- By rethinking at the low-level, incredible improvements have been accomplished.
- Generalizable results valid for any NGP without fine tuning are achieved.
- Hash-Encoding: maps multi-dimensional coordinates to a fixed size array, using an hash function
  Different coordinates can map to the same index, causinh Hash-collisions
- Accelerated ray marching by allocating samples using an occupancy grid that marks empy-non empty coarsely
- Achives incredible graphic primitives learning in seconds to minutes

- Less detailed view-dependent reflections with respect to Mip-NeRF (with several order of magnitude of speed up)
- Possible Hash-collisions create some minor artifacts on some reconstruction tasks such as SDF

NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, Wenping Wang

Introduction

- Classical Neural Scene representation techniques don't constrain surface sufficiently
- New volume rendering method to train a neural SDF representation. Render images directly from the implicit SDF representation
- Surface represented as the zero-level set of a MP encoded implicit SDF
- Reformulate the scene representation with two functions: the 3D SDF, and the view-dependent color function. Furthermore, a probability density function over the space (S-density) is defined as the derivative of the sigmoid
- The volume rendering is update with a weight function rigorously defined to be unbiased and occlusion-aware
- The loss uses an L1 Loss (MAE) and regularization factors, again the L1 loss is based on the color simlarity from the rendering
- The hierarchical sampling strategy is similar to NeRF, but only one network is mantained, using the computed s-density on the learned s to sample at fine level

- Define what is the chamfer distance (measure of point cloud distance). As a mutual nearest neighbor distance that properly capture precision and recall of your point cloud from the ground truth. This is a differentiable term that may be optimized

- Long training time (in the order of 10-20 hrs)

Underwater NeRF

SeatThru-NeRF

- Strong influence from the medium in the underwater environment
- NeRFs in scattering media
- Learn both scene informations and medium paramters (medium MLP + scene MLP)
- Uses the SeaThru model for attenuation and backscattering
- Assign separate color and density parameters to objects and medium
- Include backscatter density, attenuation density and medium color in the rendering equation
- Constrain medium parameters constant for a viewing direction, object density function of position only, and color also of viewing direction

- Doesn't account for artificial light
- High VRAM usage on GPU
- 10 hrs training on A100 GPU

Neusis

- dense 3D reconstruction of objects using an imaging sonar
- Implement a differentiable volumetric renderer that models the propagation of acoustic spherical wavefronts
- modeling pixel formation in imaging sonar explicitly
- 3D surfaces as zero-level sets of neural implicit functions
- Proper regularized rendering loss definition
- Two MLP network are learned. One extract neural implicit representation from pose, the other extract the overall intensity using ray direction, feature vector and signed distance gradient

- 6 hrs to converge on RTX 3090
- Focus on single-object reconstruction, not on large scene

AONeuS

- Direct extension of Neusis, integrating optical data
- Physics-based multimodal acoustic-optical neural surface reconstruction
- Achieve SOTA 3D reconstruction with limited baseline (sensor motion)
- Similar to Neusis, it estimate surfaces as zero-level sets of neural SDF trained here from camera and sonar data. Furthermore, predicted camera and sonar radiance are output of the MLP structures (on separate networks)
- During network training, first the reconstruction is biased towards depth information weighting 0 the camera loss, after a number of steps, also the camera is condiered in the reconstruction, solving elevation ambiguity (better wieghting can improve performances)
- No model of backscatter and absorption
- Around 30 mins for reconstruction
- Single-object and clear water

Underwater 3DGS

SeaSplat

- Constrained 3DGS with physically grounded underwater image formation model
- SeaSplat reconstruct the underwater scene with the same representation of 3DGS, recovering also true colors
- This allow to recover sota depth maps and medium parameters. The color restored (without medium) scene is also derived.
- The same image formation model of SeaThru-NeRF is used
- The loss is constrained with additional terms to ensure the inclusion of the medium in the optimization and wirghting the reconstruction loss with the depth
- Between 1hr – 2hr for training
- Real time rendering
- No view dependency (zero order SH)
- No light interactions or dynamic scenes

Z-Splat

- Opti-Acoustic fusion Gaussian SPlatting, exploiting sonar images to sample the missing cone region and obtain SOTA reconstruction in limited baseline data
- Sonar data is splatted along z-axis and combined with camera images
- In limited baseline, only camera measurements make the reconstruction problem ill posed
- Sensor fusion obtained with aweighted icombination of sonar and camera in the loss function
- Fls splat on the yz plane constraining covariance and means of gaussians on the z axis, while rgb camera constrain on xy plane, these are combined in the loss function
- 5-8 minutes for training
- Doesn't include scattering model
- Unprecise sonar image rendering model

Aqua-Splat

- Ensures physically consistent reconstruction incorporating sonar wave propagation modeling in the image formation process
- Volume rendering technique for sonar. Uses the real FLS sensor model to define 3D Gaussians along the arc
- Sonar-guided densification based on sonar positional gradient in polar coordinates
- Sonar volume rendering uses a tile-based space discretization

- The fusion is again obtained with a combined loss function of camera and sonar reconstruction loss
- Around 40-50 minutes for training, with 120 fps view-synthesis both in sonar and camera images
- Initialization from SfM on camera images (NOT ROBUST ?)
- Uses MoCAP systems to know the sensor extrinsics
- NOt clear undrstanding of opacity relationships between the two modalities

Robotics applications: (Semantic) SLAM

NeRF

Nice-SLAM

- Real-time Accurate dense geometry and camera tracking
- Take a monocular RGB-D camera input
- Neural Implicit representation (MLP representation)
- Extension of iMAP, to adapt for larger scenes
- NeRF like volume rendering
- Optimize on tiny MLP at different spatial resolution
- No loop closures,

SNI-SLAM [n]

- RGB-D Semantic SLAM
- Feature fusion method with attention architecture
- Joint optimization of map and semantics
- Pretrained segmentation network Dinov2 on RGB image to extract features
- High GPU computational power

3DGS

GS-SLAM

- Monocular SLAM using only Gaussians
- 3 fps
- Explicit Jacobian for direct camera pose estimation
- Smart keyframing
- Gaussian shape regularization and geometric verification

- No Loop closure or explicit surface extraction

GSFF-SLAM [n]

- Joint rendering of appearance, geometry and N-dimensional semantic features via Gaussians and feature fields
- Each gaussian embed semantic feature
- Decoupled semantic and scene reconstruction
- Rely on pre-trained foundation models:
  - Grounding-DINO: detect objects
  - SAM: Segment the scene
  - CLIP: encode feature class
- Not Robust in Dynamic Scenes
- Online SLAM, Offline semantic reconstruction