3D Geometry Representation and Processing for Deep Learning

Radiance Fields (neural and not)

Paolo Cignoni, Massimiliano Corsini, Daniela Giorgi, Luigi Malomo

Visual Computing Lab, ISTI-CNR, Pisa

National PhD in Artificial Intelligence & PhD in Computer Science (Univ. of Pisa)

AY 2023 – 2024



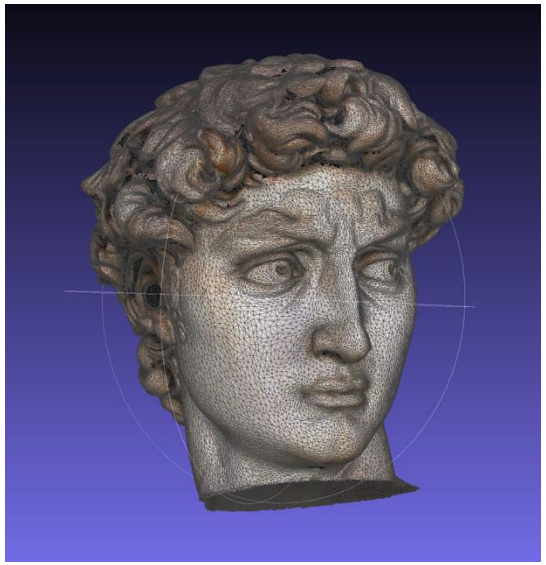
Radiance Field from a CG perspective

- Starting from the past
- Pick from Classic Nerf
- An attempt of categorization from *neuralfields.cs.brown.edu*
- Some perspective

Why this is a big thing

- 3D Scanning
  - Building a digital replica of something real
    - Old stuff (start in the 90s) that works well for many classes of objects
  - Traditional pipeline targets traditional 3D representation
    - 3D meshes and point cloud



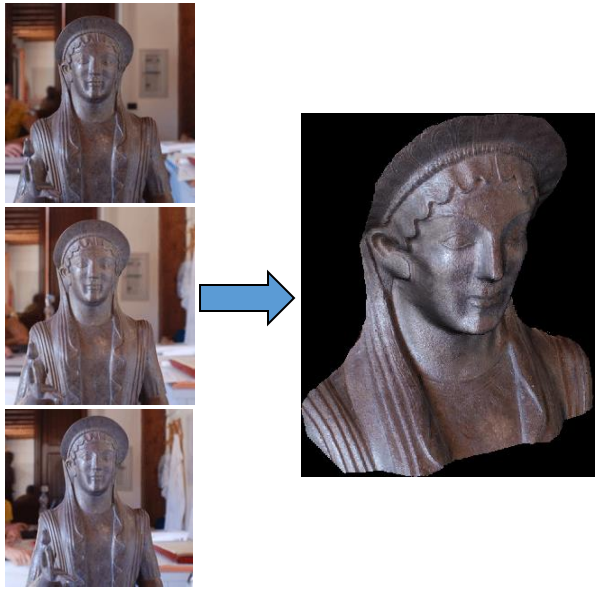


Reasonable when the output is something that can be represented as a surface

3D Scanning HowTo

- Many different techs
  - Active/Passive
  - Laser/structured lights/lidar/etc.

- Close Photogrammetry (aka 3D from photo) is a rising star
  - Cheap
  - Good appearance
  - Getting better and better

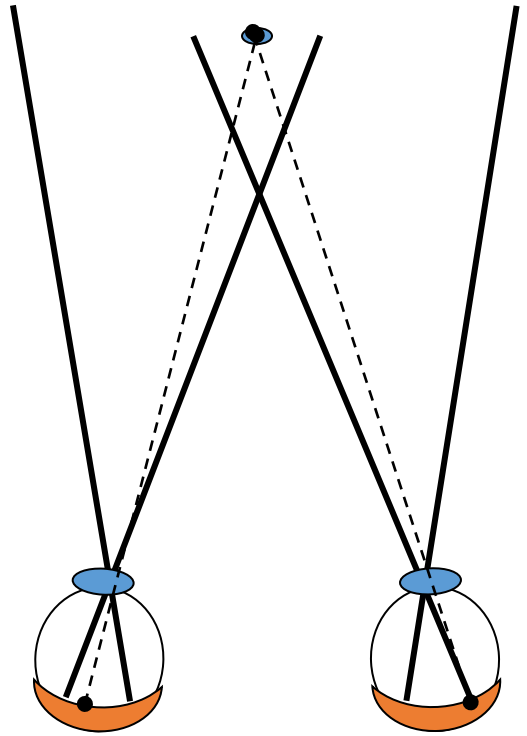


3D from photo: SW options

- Major players
- ***Agisoft MetaShape***
  - Powerful/versatile/costly
- ***RealityCapture***
  - Very fast / flexible licensing
- Many cell phone app arising
  - RealityScan
- OpenSource
  - **Colmap**
  - **Micmac**
  - ***Alicevision***

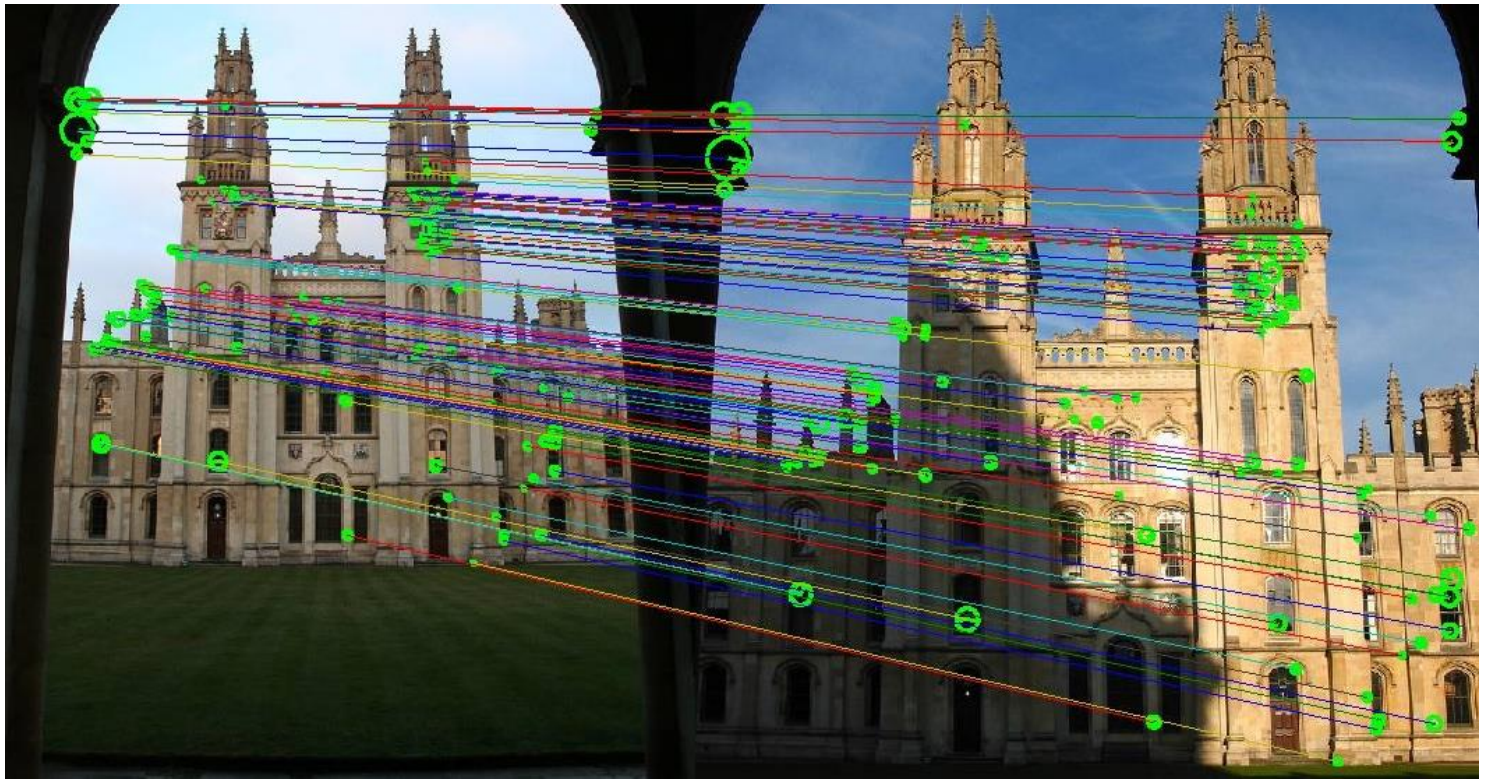
Close photogrammetry: How does it work?

- We do perceive the environment three-dimensionality thanks to our **stereo vision** AND the perspective projection that occurs in our eyes..
- Using the geometric laws of this two phenomena we can reconstruct the 3D geometry of the scene



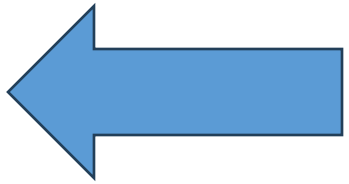
Close photogrammetry: Yes, but how does it ***really*** work?

- The entire process is based on finding matches between images.
  - *Matches are couple of visually similar points on the image that should hopefully be the same point in the world.*



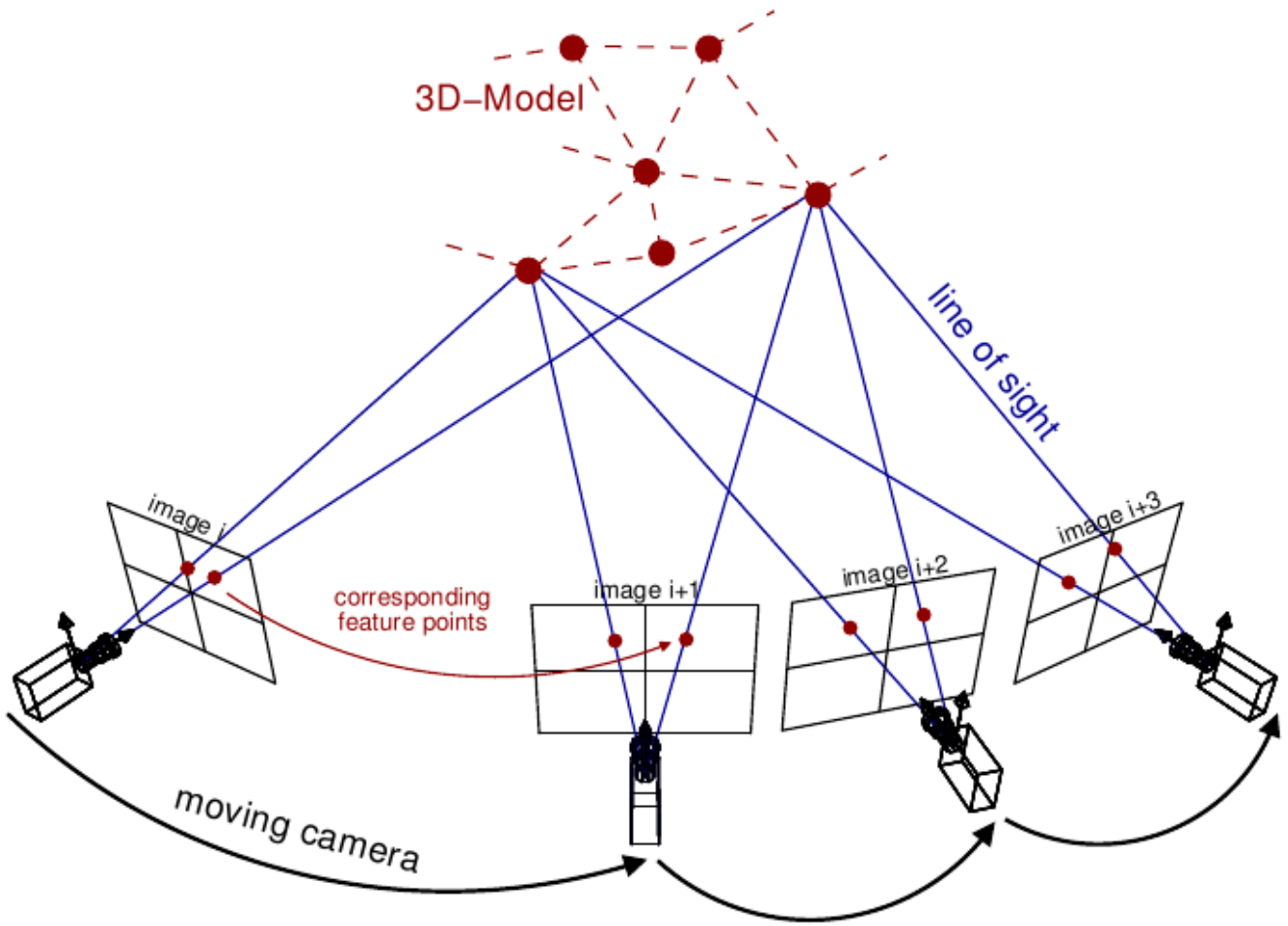
Close photogrammetry: How does it work?

- Multiple Steps
  - Finding Matches
    - Finding feature points and actually matching them
  - Camera Calibration
    - Position and calibration of Cameras
    - 3D Position of matched feature points
  - Dense Matching
  - Merging, meshing and texturing



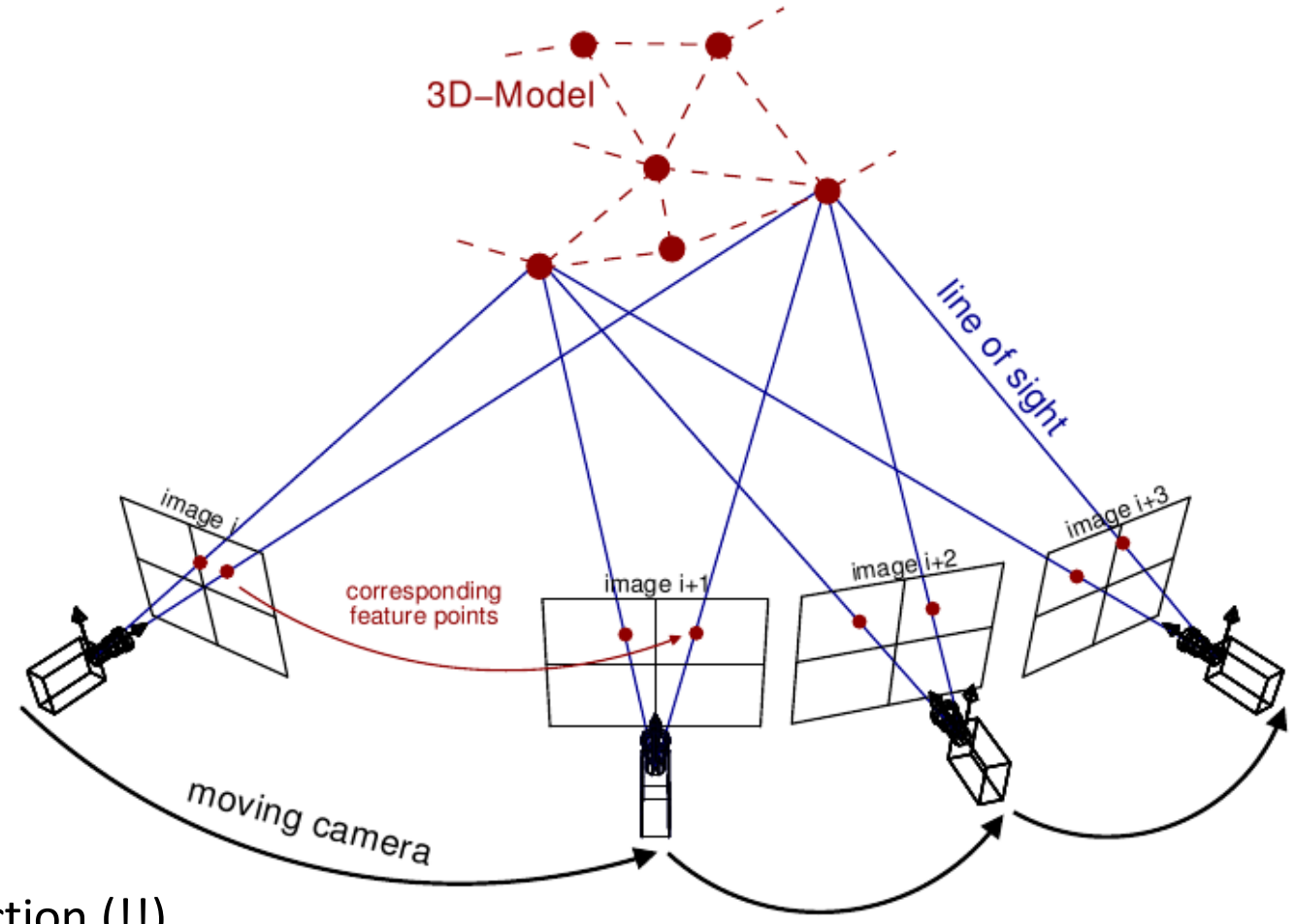
How does it work?

- Multiple Steps
  - Finding Matches
    - Finding feature points
    - Actually matching them
  - Recovering Cameras
  - Position and calibrations



Camera Calibration

- No prior knowledge about camera calibration is available, so all information must be recovered from the images
- It is important that **consistent features** are in the images!
  - *No white walls,*
  - *No changing appearance*
  - *No moving stuff*
  - *No blur*
- Important factors:
  - Position of the camera
  - Enough overlap
    (only points that are visible
    in at least 3 images are useful)



The scene to be reconstructed
and how you shot photos
have HUGE influence on the final reconstruction (!!)

Dense Matching

- After recovery of the camera calibration, multiple set of images are considered and for each image a (incomplete, noisy) dense depth map is computed

- These contain the depth of every pixel and a quality measure (how confident we are of each particular pixel)
- ***Output is a point cloud***

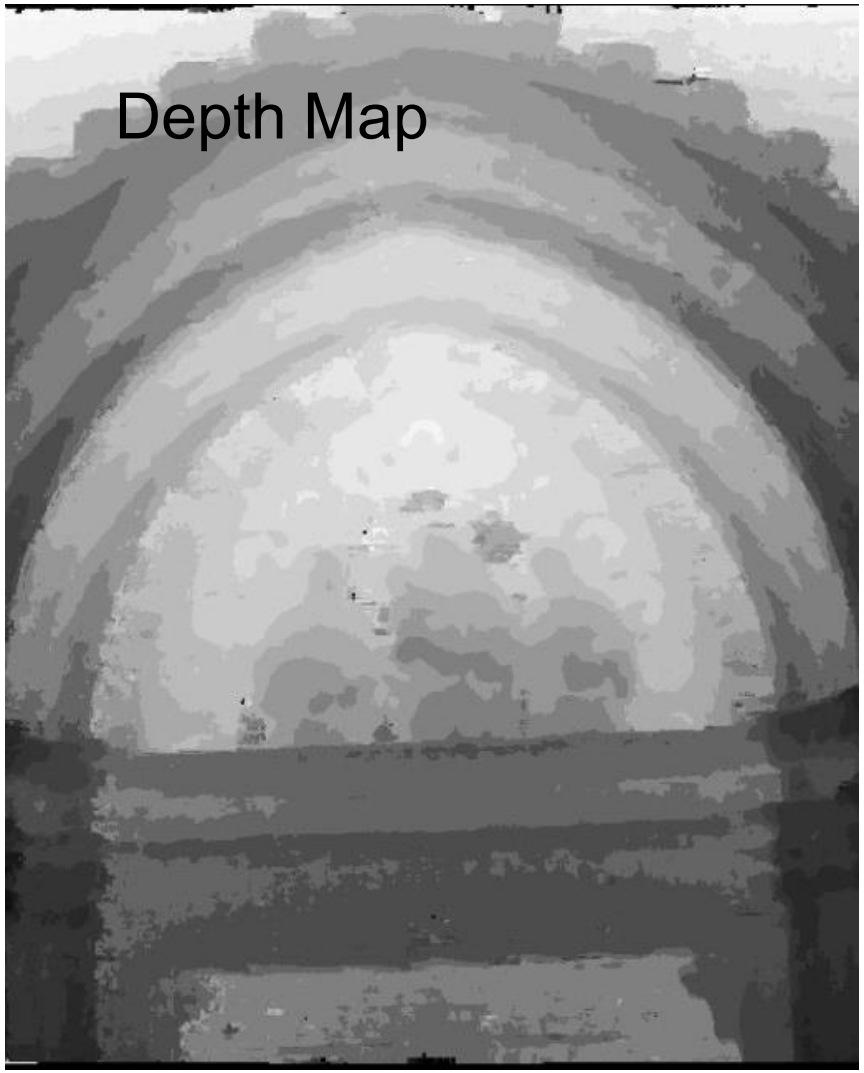
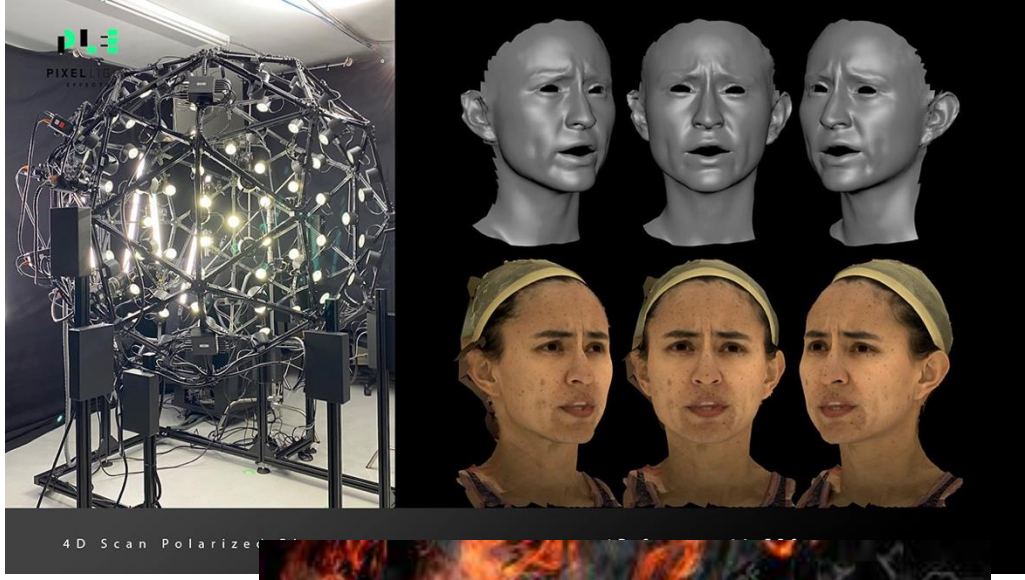


Image-based 3D Reconstruction

- ***Advantages:***
  - Cheap (no scanning device needed), fast tech evolution
  - Good flexibility (both small and huge model can be acquired)
  - Cameras are more easy to use than a scanner (lighter, no tripod, no power, multiple lenses …)
  - Non-expert users can create 3D models

- ***Disadvantages:***
  - Accuracy (not so accurate, problems with regions with insufficient detail)
  - Could be slower than active techniques (many images to process and merge)
  - Not all the objects can be acquired

Photogrammetry Limits



Traditional techniques fails for two main reasons:

- Not everything is a surface
  (e.g. can be easily represented by mesh or points)
- Not every kind surface is willing to be scanned









How to workaround the problem

Change the objective:

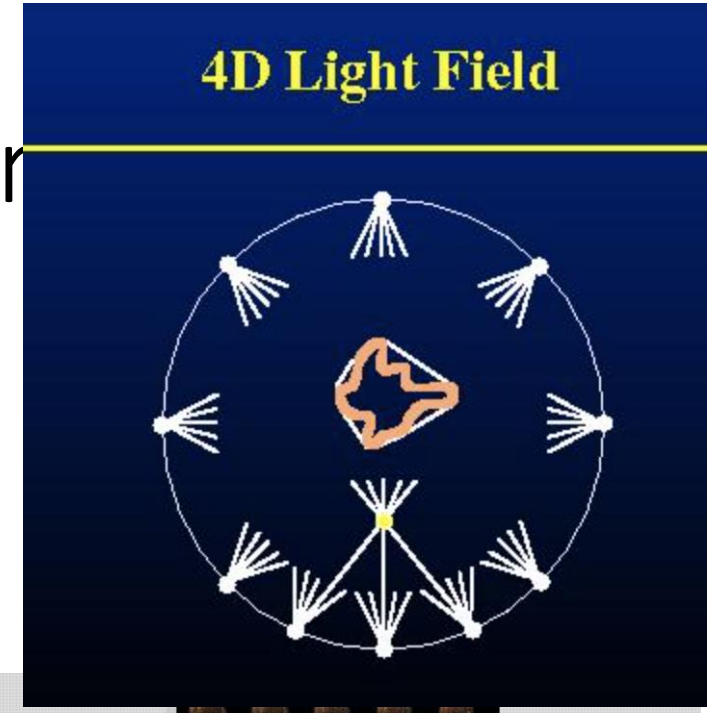
Building a 3D representation of shape and appearance

Versus

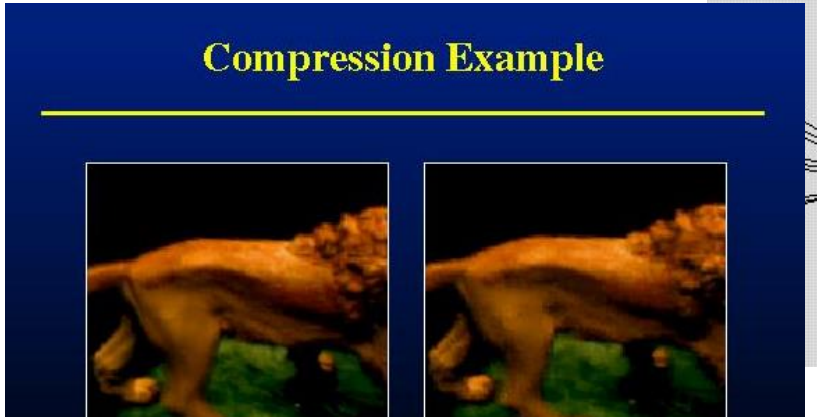
Targeting the reproduction of light going to a virtual observer

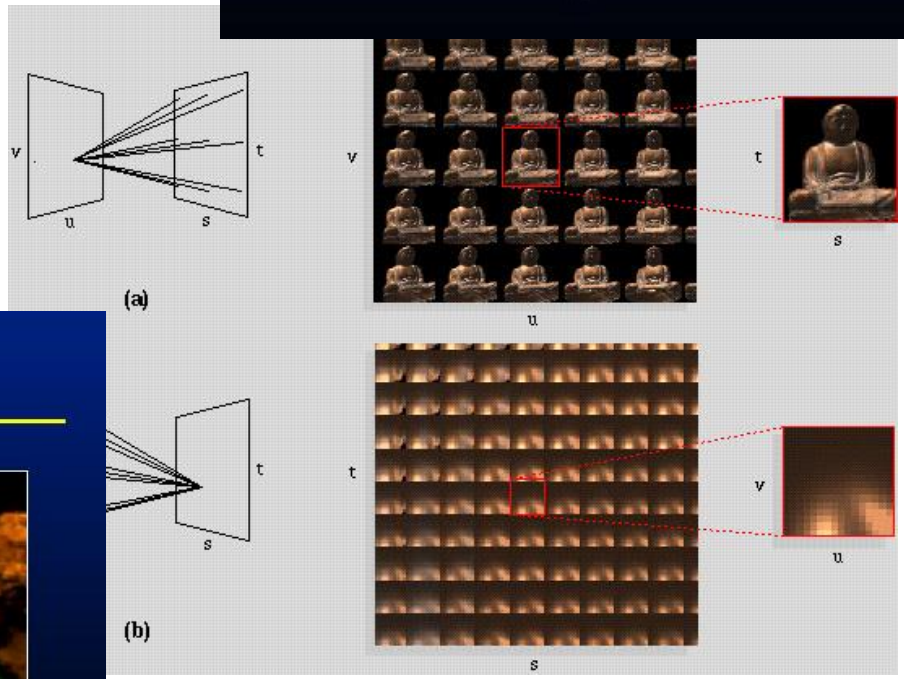
Not a new idea, many (partial) failur

- Light Field Rendering (Levoy Hanaran 1996)
- “They are based on the notion of generating (i.e. rendering) new images of a virtual scene from existing images rather than from 3D models”
- Lot of issues
- Huge number of images required
  - Lot of memory
  - Strong camera limitations









Not a new idea, many (partial) failures

- Lumigraph (Gortler et al. 1996)
- Idea restricting the plenoptic function (5d) to a surface surrounding an object (4d) and try to compress and resample this (denslely) sampled info
-

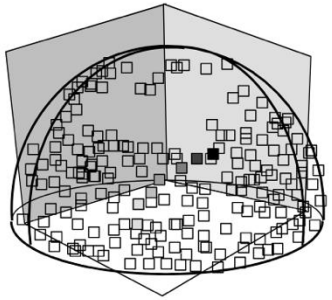


Figure 11: The user interface for the image capture stage displays the current and previous camera positions on a viewing sphere. The goal of the user is to "paint" the sphere.

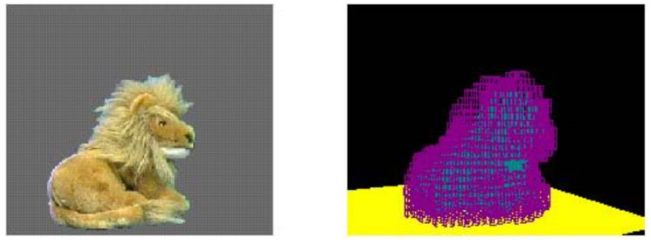


Figure 12: Segmented image plus volume construction

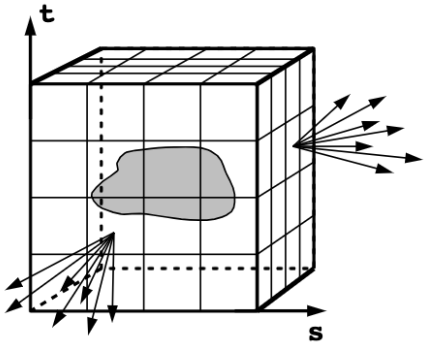
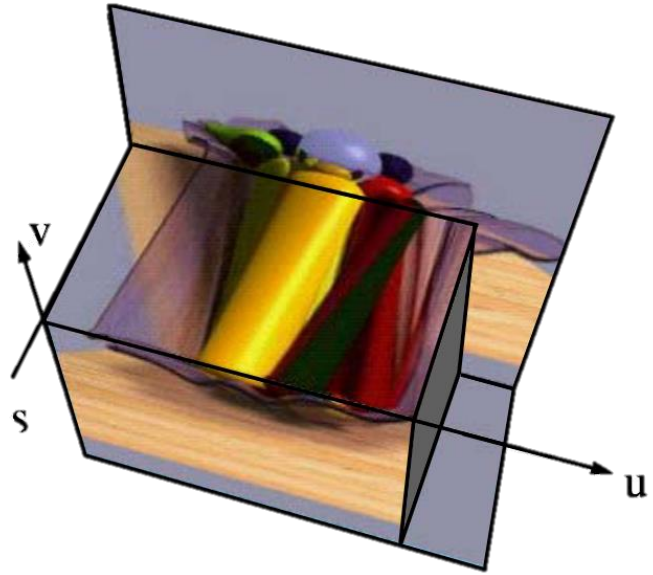


Figure 1: The surface of a cube holds all the radiance information due to the enclosed object.





Light Field Rendering. Not a new idea

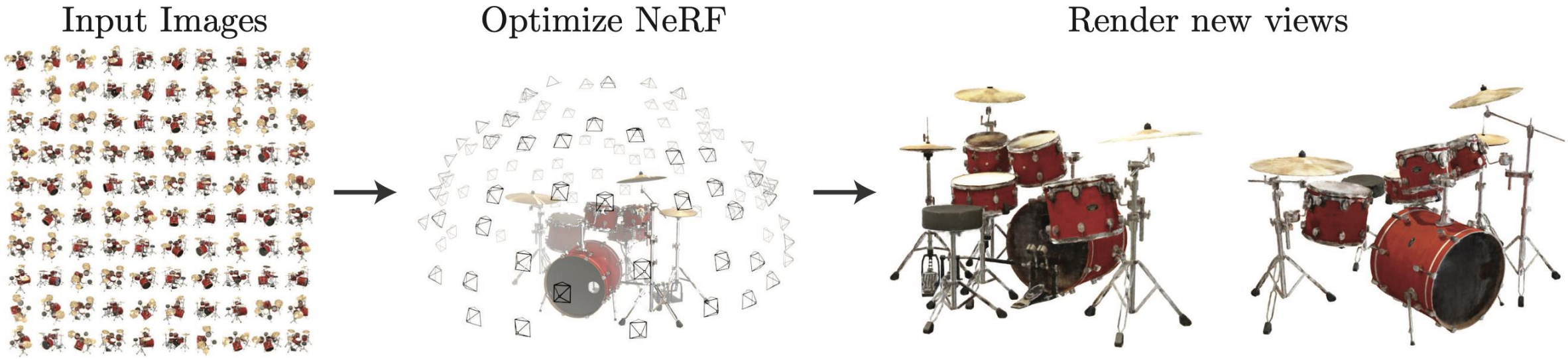
- Since 1996 Lot of efforts
  - Dynamically reparameterized light fields(Isaksne 2000)
  - Fourier Slice Photography (Ren Ng 2005)
  - Unstructured Light Fields(Abe Davis 2012)
  - The light field stereoscope (Huang 2015)



- Compression
  - these models where huge, GB size, in 1996 the avg pc had 8..16MB
- Sampling
  - They required very dense sampling (eg huge number of photos and their absence generated blurriness…)
- Constrained viewing
- Not very far from being good at the end…

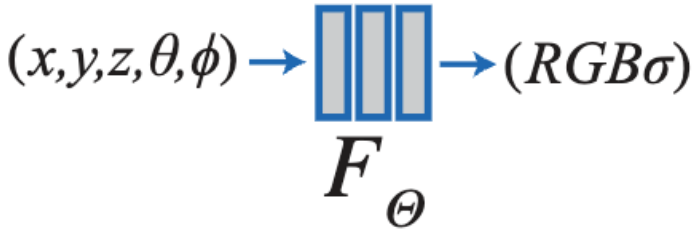
Neural Radiance field

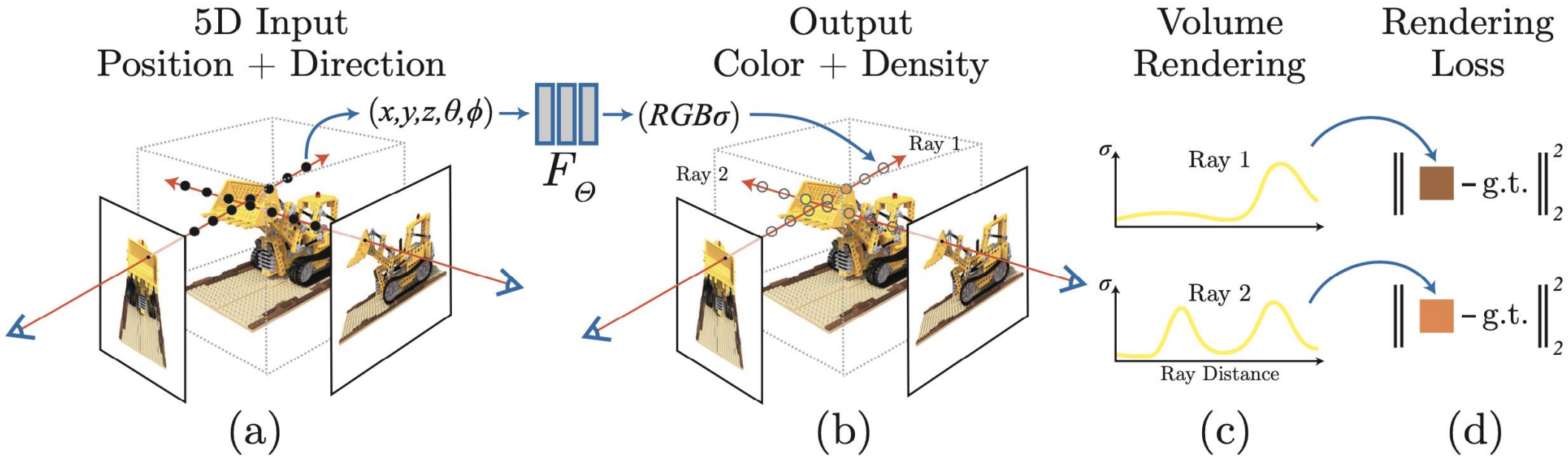
- Do the same but with a neural network
  - NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis (Mildenhall etal 2020)



Neural Radiance Field

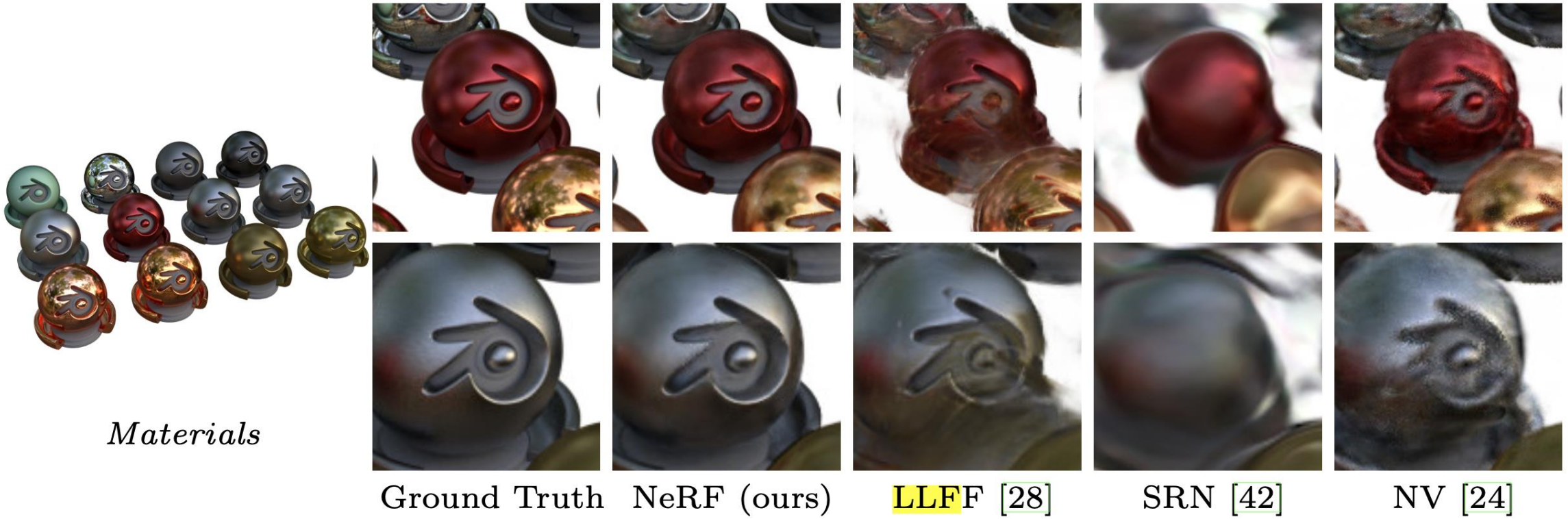
Main idea: to encode in the neural field a density + radiance spatial function e.g a function that for each point (x, y, z) and direction (θ, ϕ) of space it gives me back the density σ in this point and the radiance RGB along that direction



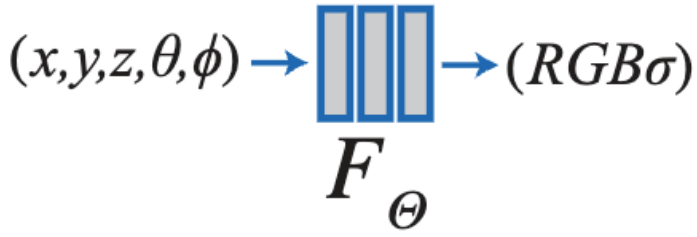


Not the first to explore this (neural) path too

- Neural Volumes (Lombardi 2019)
- Scene Representation Networks (Sitzmann 2019)
- Local Light Field Fusion (Mildenhall 2019)



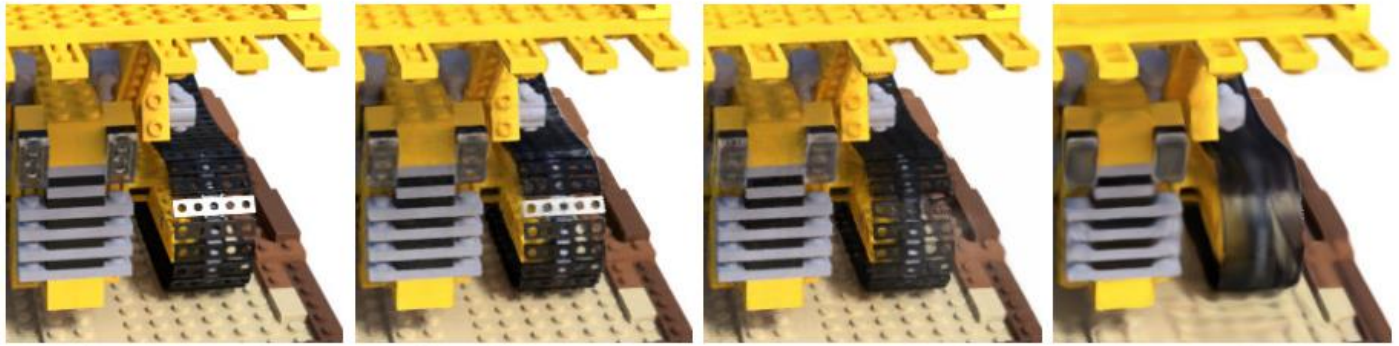
Main improvements



- Positional Encoding
  - Eg NN as general function approximation have a low freq bias
  - Exploits Transformers Positional Encoding trick
    - Map each coord R-> R^{2L}

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

$$F_\Theta = F'_\Theta \circ \gamma;$$



What happens after?

The “Cambrian Explosion” of Neural Fields

InstantNGP

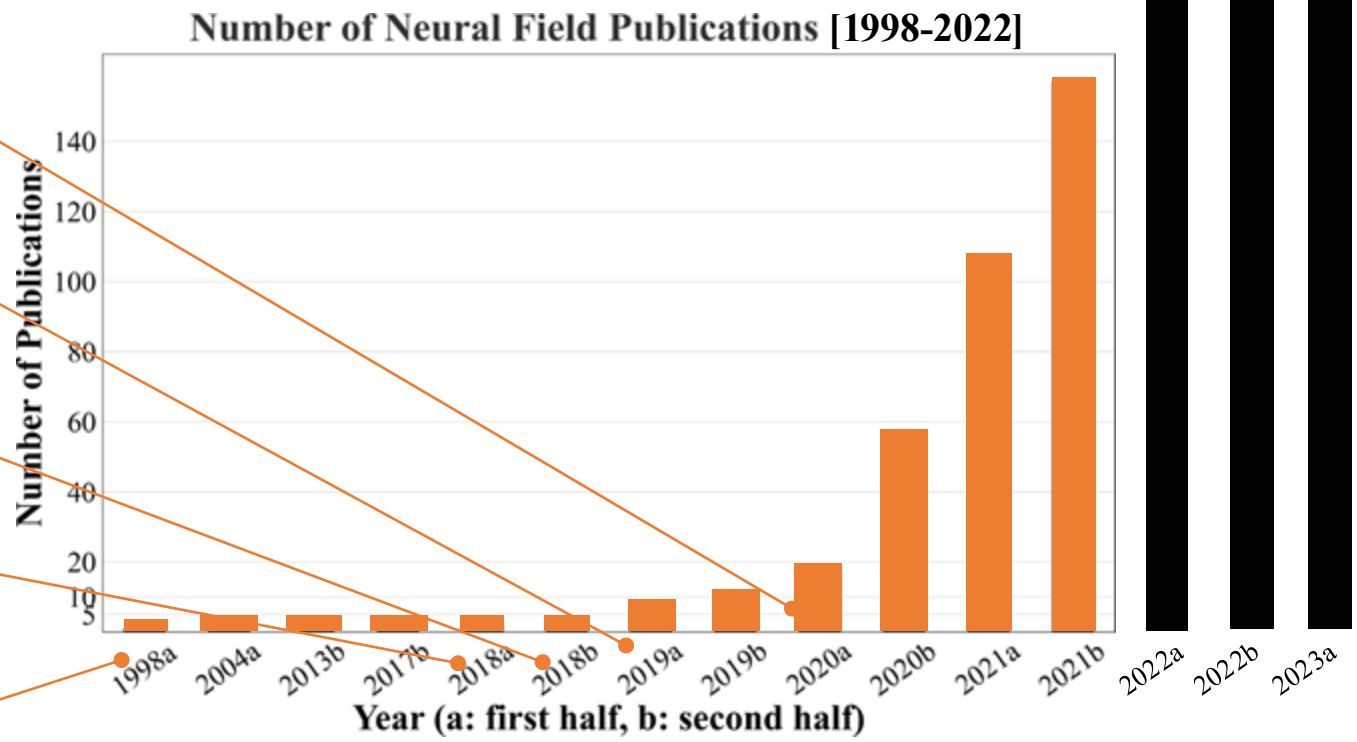
NeRF

DeepSDF, PIFu

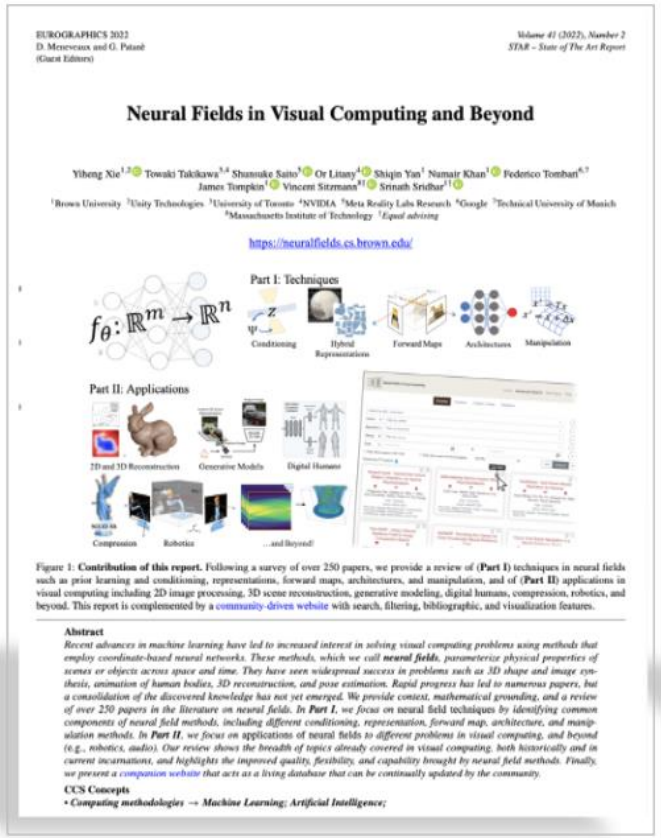
Occupancy Networks, IM-NET

AtlasNet

Approximating Reflectance Functions using Neural Networks



Great Reference
(and inspiration for most of the next slides)



EUROGRAPHICS 2022
D. Mesveaux and G. Patané
(Guest Editors)

Volume 41 (2022), Number 2
STAR – State of The Art Report

Neural Fields in Visual Computing and Beyond

Yiheng Xie[1,2] Towaki Takikawa[3,4] Shunsuke Saito[5] Or Litany[4] Shiqin Yan[1] Numair Khan[1] Federico Tombari[6,7]
James Tompkin[1] Vincent Sitzmann[8] Srinath Sridhar[1]
[1]Brown University [2]Unity Technologies [3]University of Toronto [4]NVIDIA [5]Meta Reality Labs Research [6]Google [7]Technical University of Munich
[8]Massachusetts Institute of Technology †Equal advising

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

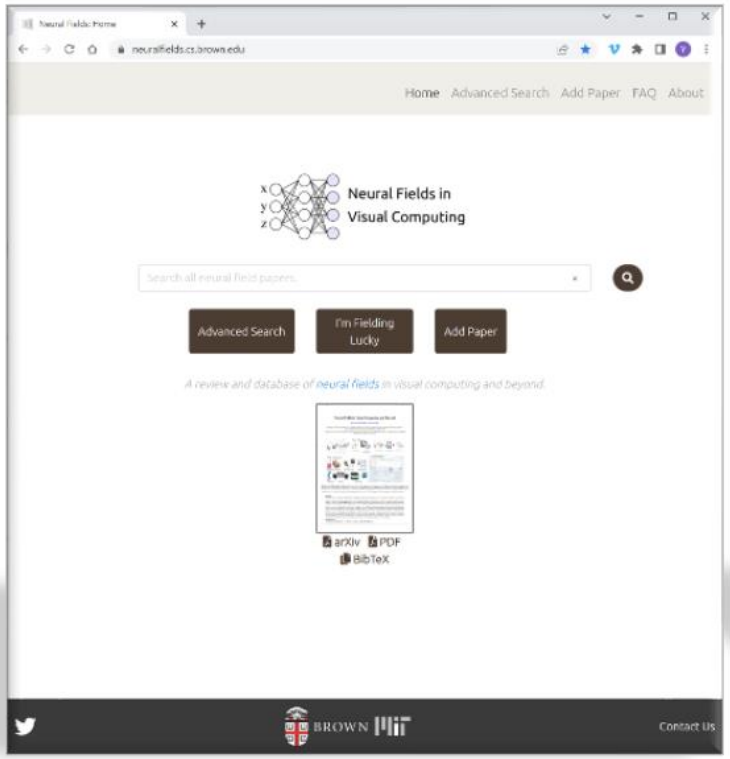
Figure 1: **Contribution of this report.** Following a survey of over 250 papers, we provide a review of (**Part I**) techniques in neural fields such as prior learning and conditioning, representations, forward maps, architectures, and manipulation, and of (**Part II**) applications in visual computing including 2D image processing, 3D scene reconstruction, generative modeling, digital humans, compression, robotics, and beyond. This report is complemented by a community-driven website with search, filtering, bibliographic, and visualization features.

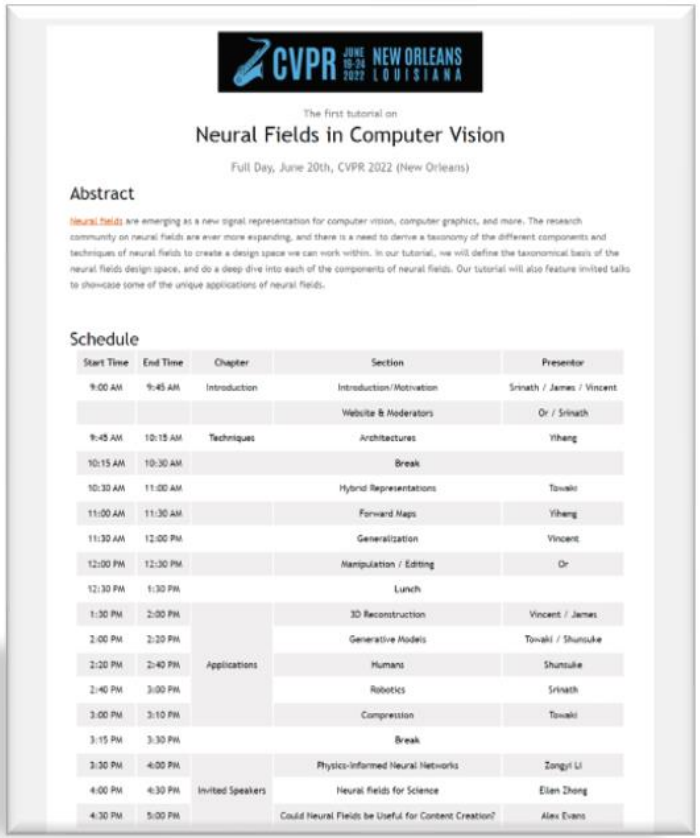
Abstract

Recent advances in machine learning have led to increased interest in solving visual computing problems using methods that employ coordinate-based neural networks. These methods, which we call ***neural fields****, parameterize physical properties of scenes or objects across space and time. They have seen widespread success in problems such as 3D shape and image synthesis, animation of human bodies, 3D reconstruction, and pose estimation. Rapid progress has led to numerous papers, but a consolidation of the discovered knowledge has not yet emerged. We provide context, mathematical grounding, and a review of over 250 papers in the literature on neural fields. In* ***Part I****, we focus on neural field techniques by identifying common components of neural field methods, including different conditioning, representation, forward map, architecture, and manipulation methods. In* ***Part II****, we focus on applications of neural fields to different problems in visual computing, and beyond (e.g., robotics, audio). Our review shows the breadth of topics already covered in visual computing, both historically and in current incarnations, and highlights the improved quality, flexibility, and capability brought by neural field methods. Finally, we present a companion website that acts as a living database that can be continually updated by the community.*

CCS Concepts
• ***Computing methodologies*** → ***Machine Learning***; ***Artificial Intelligence***;

neuralfields.cs.brown.edu





CVPR JUNE 19-24 2022 NEW ORLEANS LOUISIANA

The first tutorial on

Neural Fields in Computer Vision

Full Day, June 20th, CVPR 2022 (New Orleans)

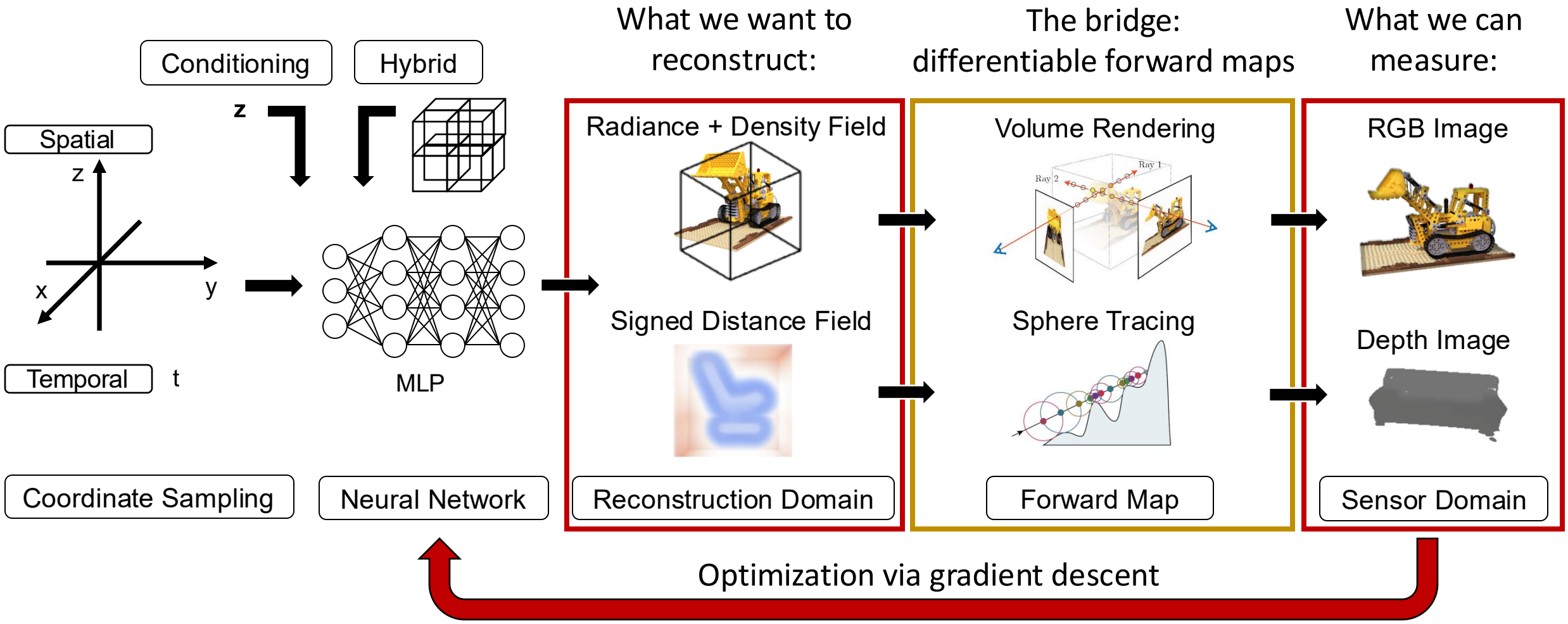
Abstract

Neural fields are emerging as a new signal representation for computer vision, computer graphics, and more. The research community on neural fields are ever more expanding, and there is a need to derive a taxonomy of the different components and techniques of neural fields to create a design space we can work within. In our tutorial, we will define the taxonomical basis of the neural fields design space, and do a deep dive into each of the components of neural fields. Our tutorial will also feature invited talks to showcase some of the unique applications of neural fields.

Schedule

Start Time	End Time	Chapter	Section	Presentor
9:00 AM	9:45 AM	Introduction	Introduction/Motivation	Srinath / James / Vincent
			Website & Moderators	Or / Srinath
9:45 AM	10:15 AM	Techniques	Architectures	Yiheng
10:15 AM	10:30 AM		Break	
10:30 AM	11:00 AM		Hybrid Representations	Towaki
11:00 AM	11:30 AM		Forward Maps	Yiheng
11:30 AM	12:00 PM		Generalization	Vincent
12:00 PM	12:30 PM		Manipulation / Editing	Or
12:30 PM	1:30 PM		Lunch	
1:30 PM	2:00 PM	Applications	3D Reconstruction	Vincent / James
2:00 PM	2:20 PM		Generative Models	Towaki / Shunsuke
2:20 PM	2:40 PM		Humans	Shunsuke
2:40 PM	3:00 PM		Robotics	Srinath
3:00 PM	3:10 PM		Compression	Towaki
3:15 PM	3:30 PM		Break	
3:30 PM	4:00 PM	Invited Speakers	Physics-Informed Neural Networks	Zongyi Li
4:00 PM	4:30 PM		Neural fields for Science	Ellen Zhong
4:30 PM	5:00 PM		Could Neural Fields be Useful for Content Creation?	Alex Evans

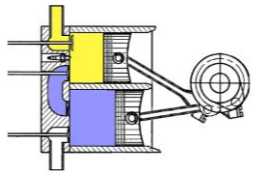
Neural Fields General Framework



Strengths and Weaknesses of Neural Fields

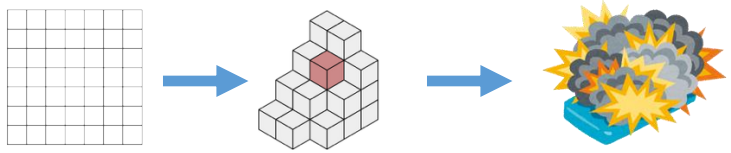
Strengths:

Compactness



$$\text{argmin}_x \|y - F(x)\| + \lambda P(x).$$

Self-regularizing



Domain agnostic

Weaknesses:

Computationally expensive

Not easily editable

Hard to model semantics and discrete data

Lack of theoretical understanding

Neural FieldTaxonomy

Field Parameterization

Geometry & Appearance Model

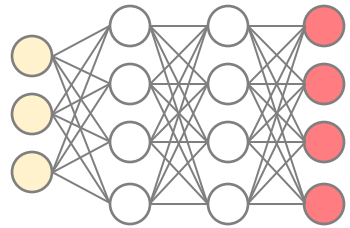
Forward Map

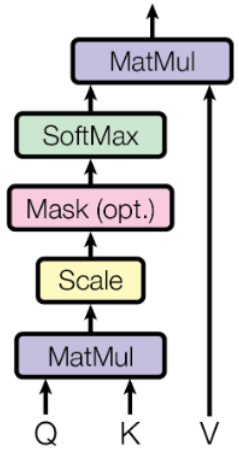
Inference Model

Recap of 3D parameterizations

MLP neural field

Monolithic MLP that maps coordinates to features



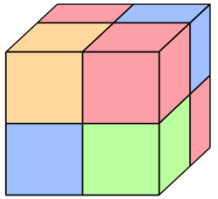


Transformer neural field

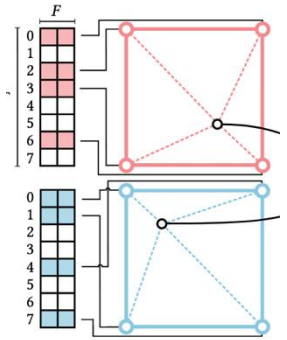
Neural field queried via cross-attention of coordinates over tokens

Hybrid discrete-continuous neural fields

Locally store MLP parameters / features



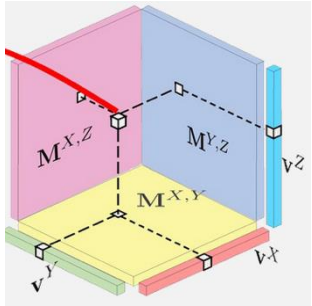


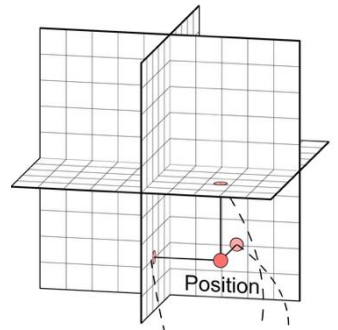


Voxels Hash Maps

Meshes Point clouds

Factorizations



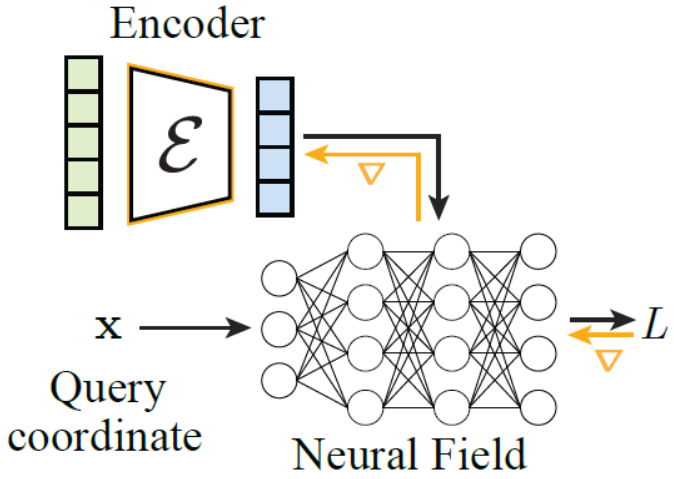


Recap of inference models

Per-scene Optimization

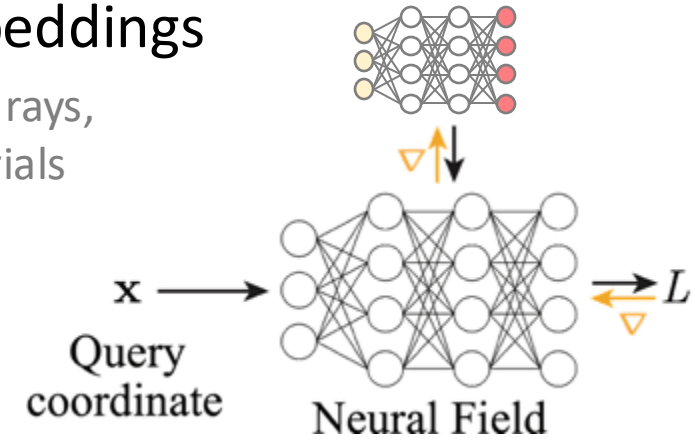


Feature encoding

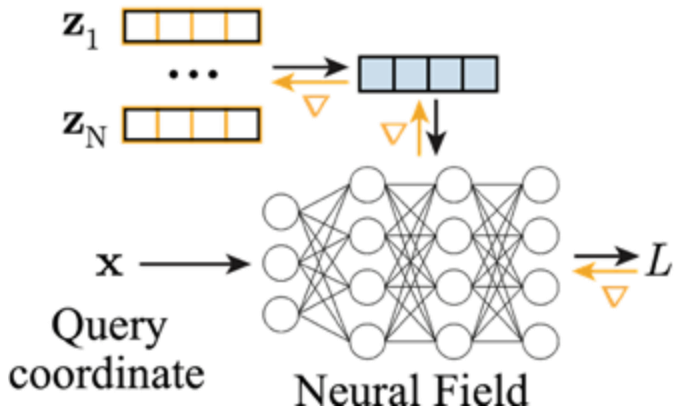


Pre-trained or optimized embeddings

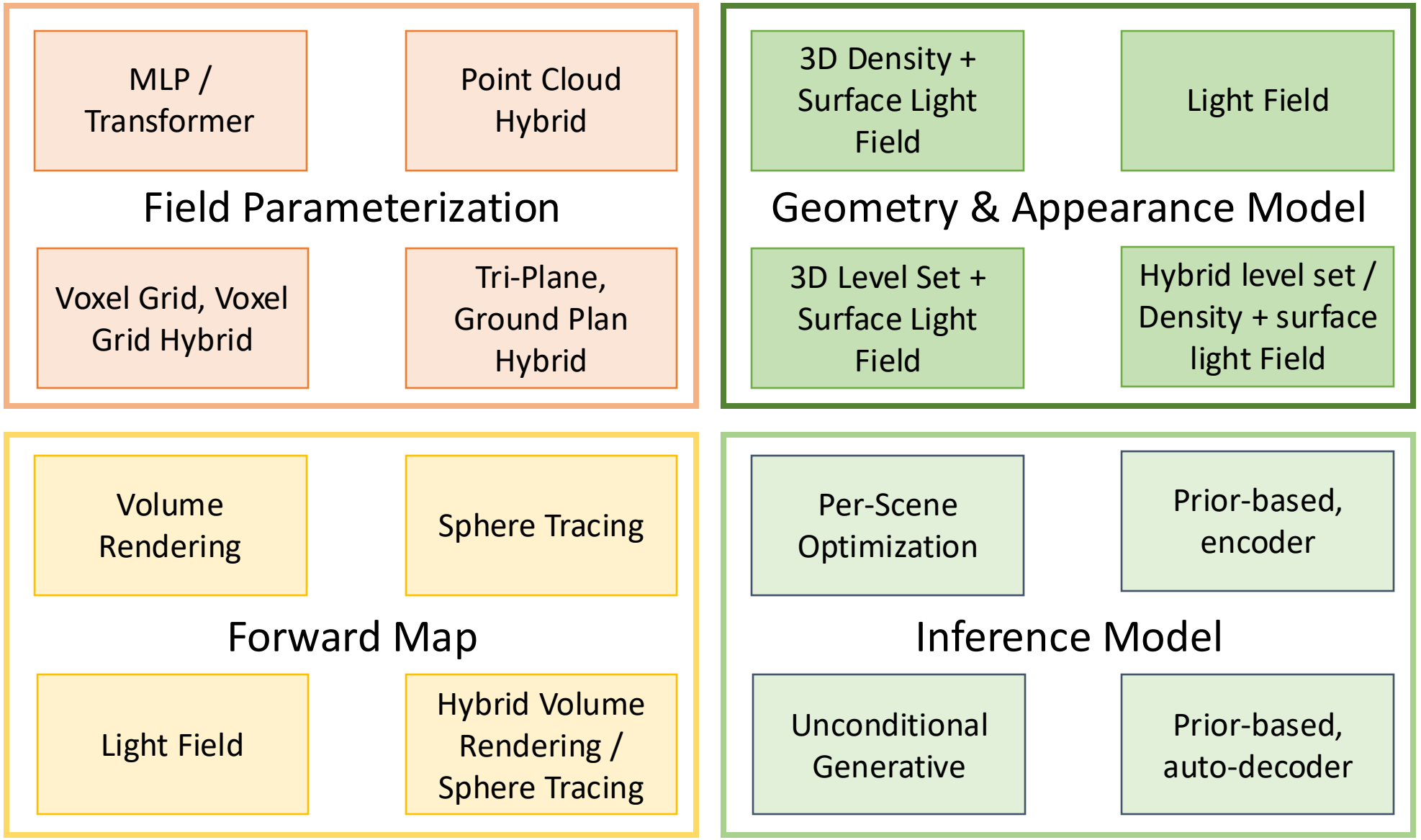
Time, rays, materials



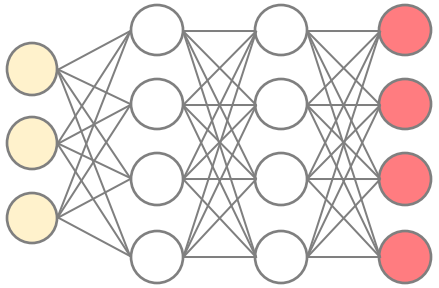
Auto-decoding



3D Inverse Graphics Taxonomy



Example: NeRF



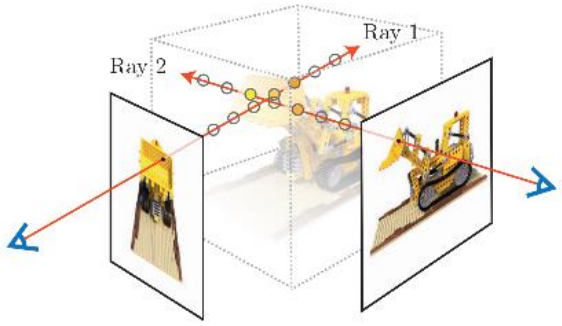
Field Parameterization

(σ, C)

Density Radiance

Geometry & Appearance Model

Volume Rendering



Forward Map

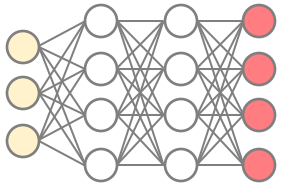
Per-Scene Optimization



Inference Model

Example: NeRF

Field Parameterization



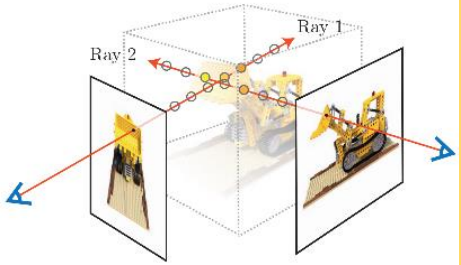
Geometry & Appearance Model

(σ, C)

Density Radiance



Forward Map



Volume Rendering

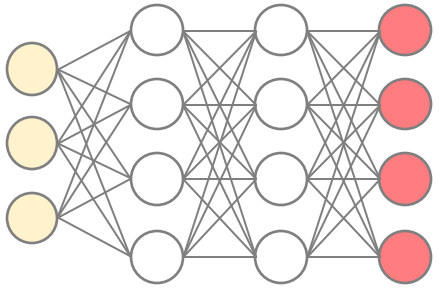
Inference Model



Per-Scene Optimization

- High quality
- Slow rendering
- High memory in training forward pass
- Low storage cost (weights)
- Noisy or cloudy geometry
- Requires many images

Example: NeRF



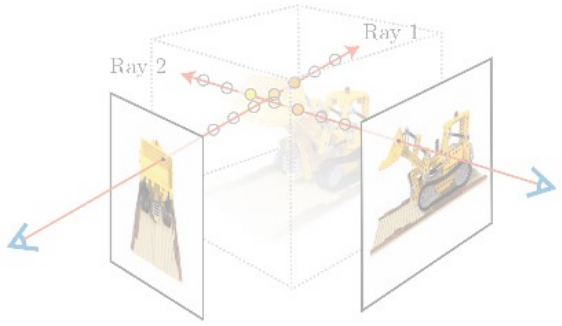
Field Parameterization

(σ, C)

Density Radiance

Geometry & Appearance Model

Volume Rendering



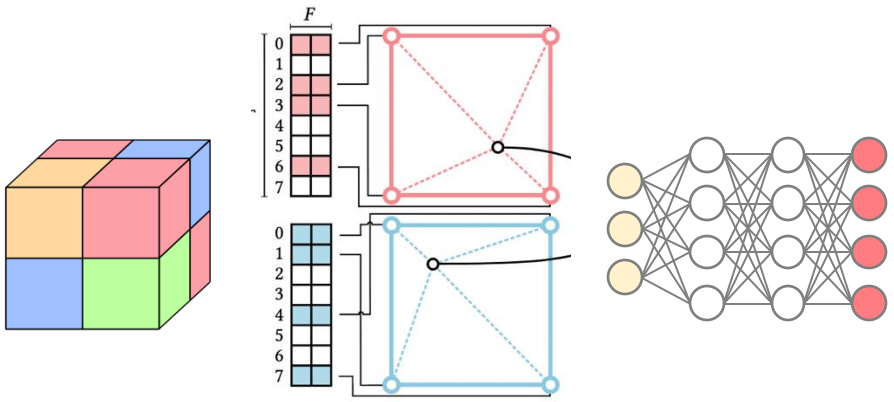
Forward Map

Per-Scene Optimization



Inference Model

Faster Radiance Fields w/ Hybrid Data Structures



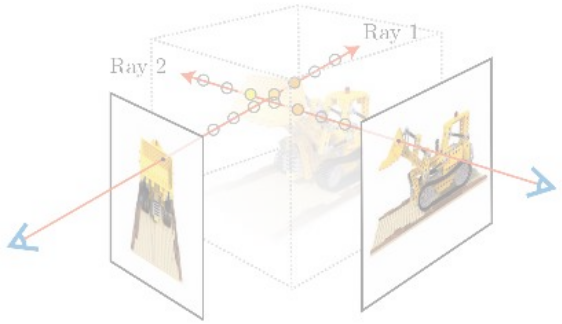
Field Parameterization

(σ, C)

Density Radiance

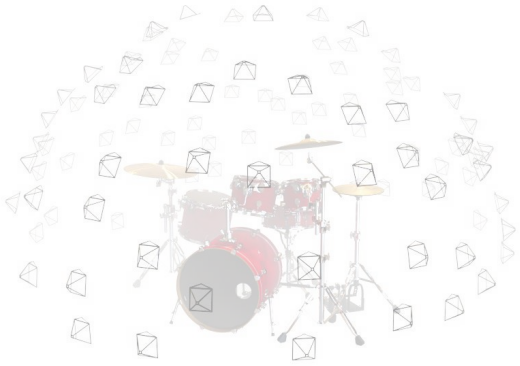
Geometry & Appearance Model

Volume Rendering

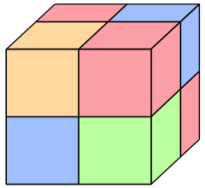


Forward Map

Per-Scene Optimization



Inference Model



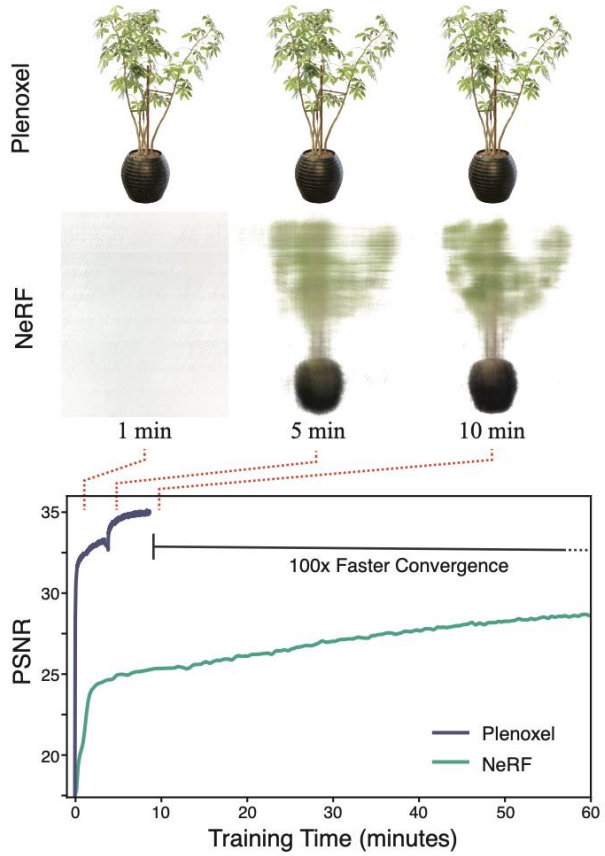
Voxel Grid

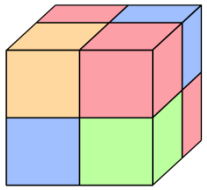


Plenoxels
[Yu et al. 2022]
Direct Voxel Grid Optimization
[Sun et al. 2021]

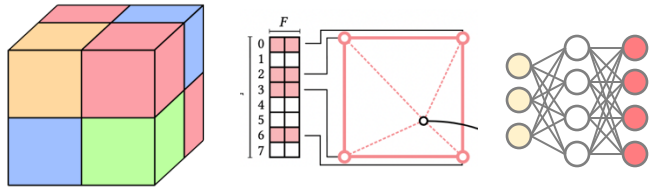
Plenoxels represent a scene as a sparse 3D grid with spherical harmonics (instead of a pure neural sampling function)

"*Our experiments suggest the key element of Neural Radiance Fields is not the neural network but the differentiable volumetric renderer*"





Voxel Grid



Voxel Grid + Hashmap + MLP



Plenoxels

[Yu et al. 2022]

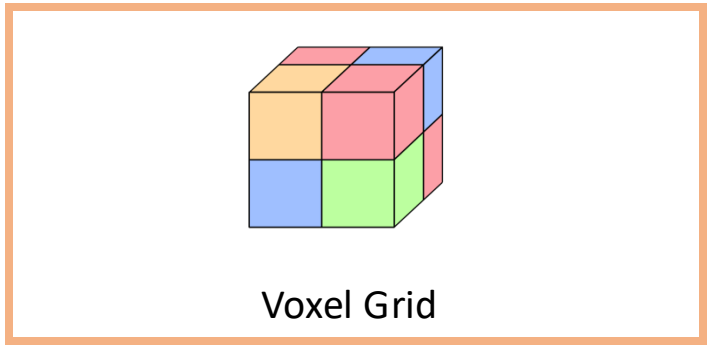
Direct Voxel Grid Optimization

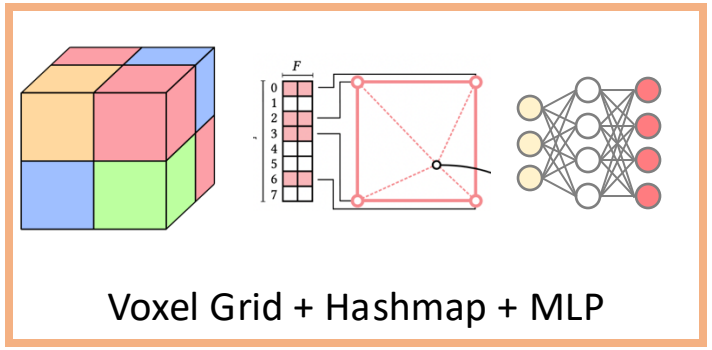
[Sun et al. 2021]

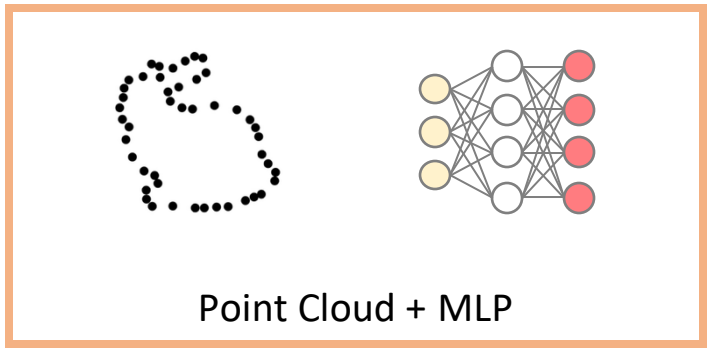


InstantNGP

[Müller et al. 2022]









Plenoxels

[Yu et al. 2022]

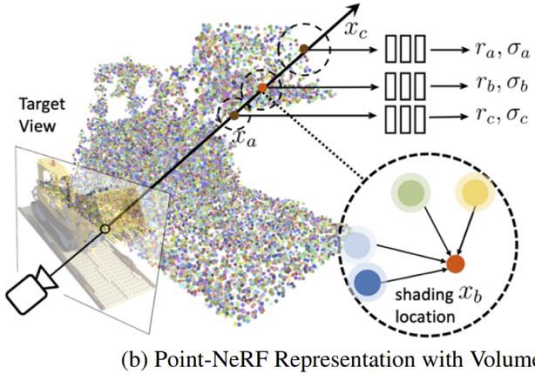
Direct Voxel Grid Optimization

[Sun et al. 2021]



InstantNGP

[Müller et al. 2022]

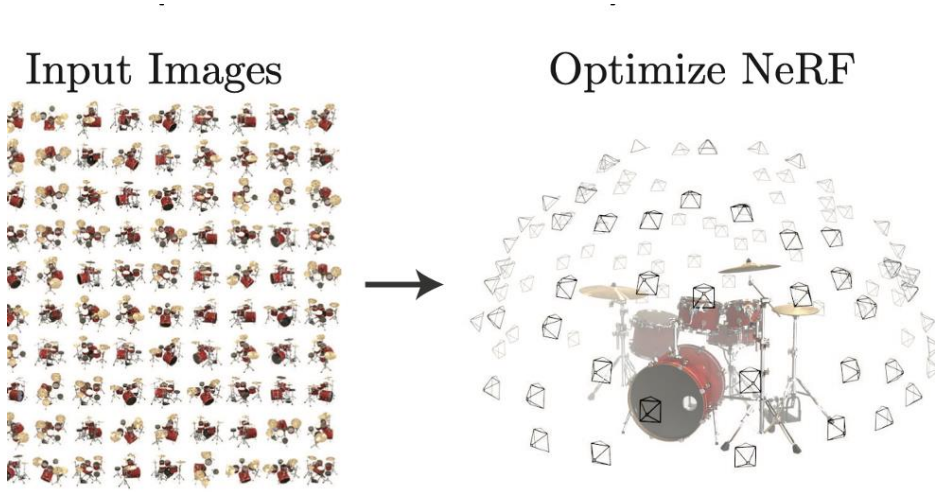


PointNeRF

[Xu et al. 2022]

Back to the origin

- Remember where we started?
- The vast majority of all the above still strongly rely on the first steps of close range photogrammetry to get correct camera calibration
- Scenes where these condition fails never represented in literature
  - all examples are: “problematic object in nice background” , why?



Radiance Fields without Neurons:

GAUSSIAN SPLATTING

Kerbl, Kopanas et al., 3D Gaussian Splatting for Real-Time Radiance Field Rendering, SIGGRAPH 2023 *Best Paper*

Rendering of real-world place



Rendering of real-world place

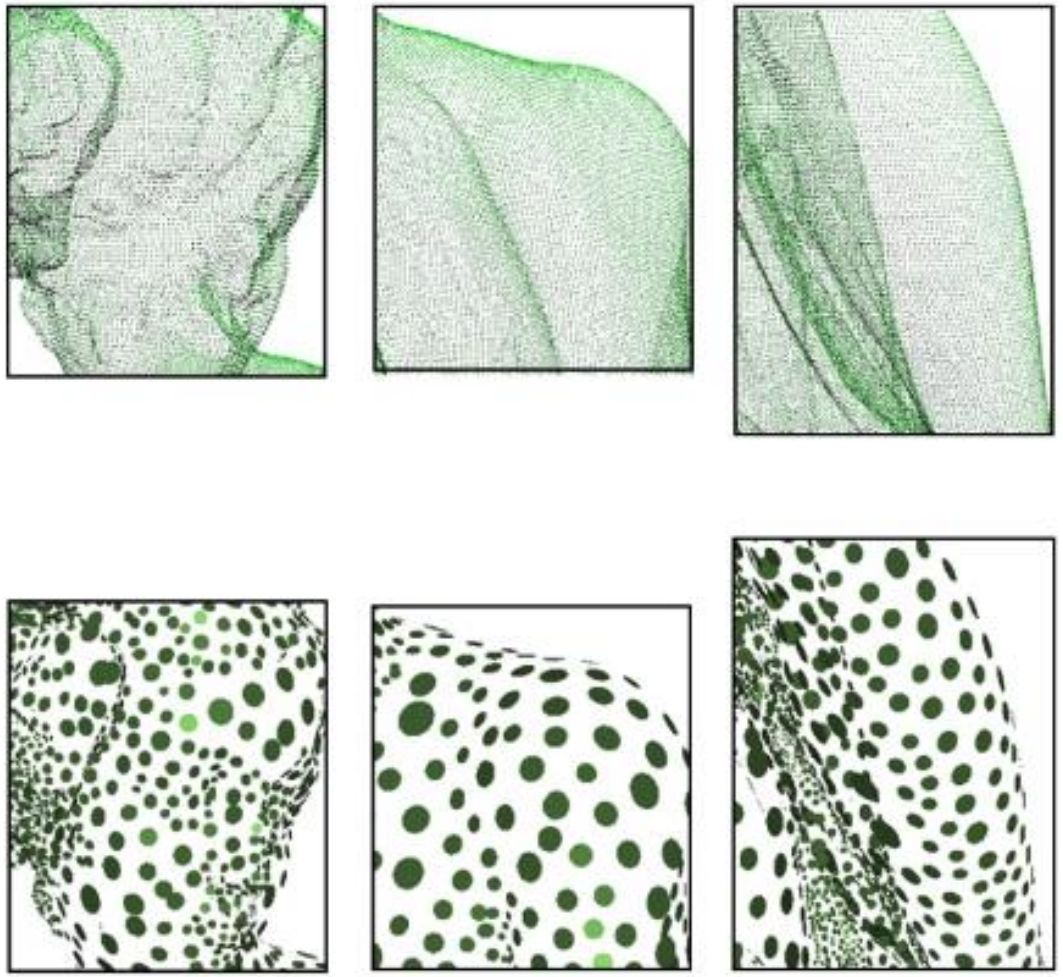


100fps 1080p rendering!
30 min. optimization time.

Splat Rendering

- Inspired by point-based rendering.
- Treat each point as an ellipsoidal volume.

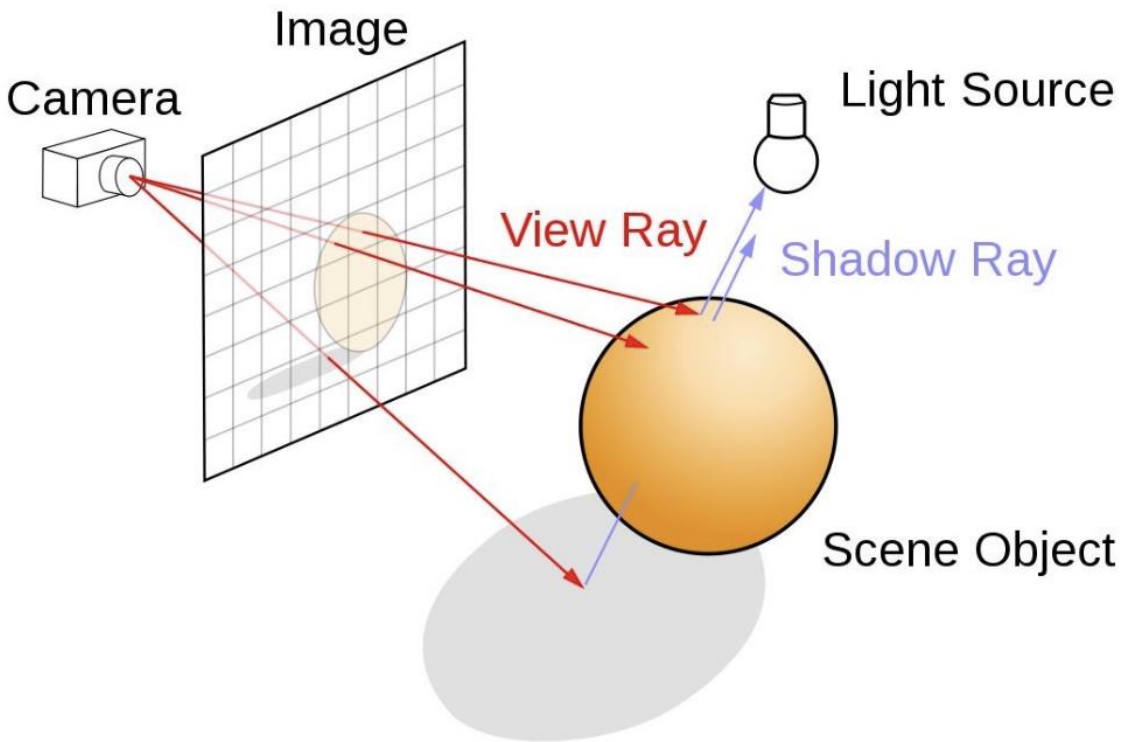




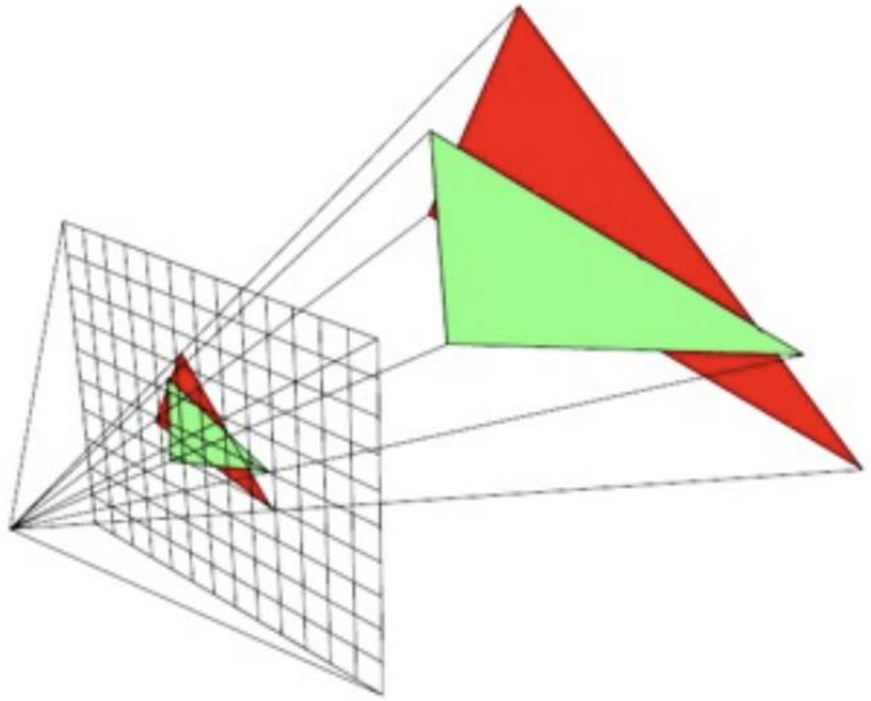
Advantages

- Accurate
  - Comparable PSNR with the state-of-the-art NeRF method.
- Fast
  - 100+ frames/second for rendering. ~1 hour for training.
- Memory efficient
  - Performs on mobile devices (< 6GM VRAM).
- Easy to integrate in existing graphics frameworks
  - Integrated to Unity, Unreal engine, etc.

- **Ray Casting**:
- Shoots rays from the camera into the scene

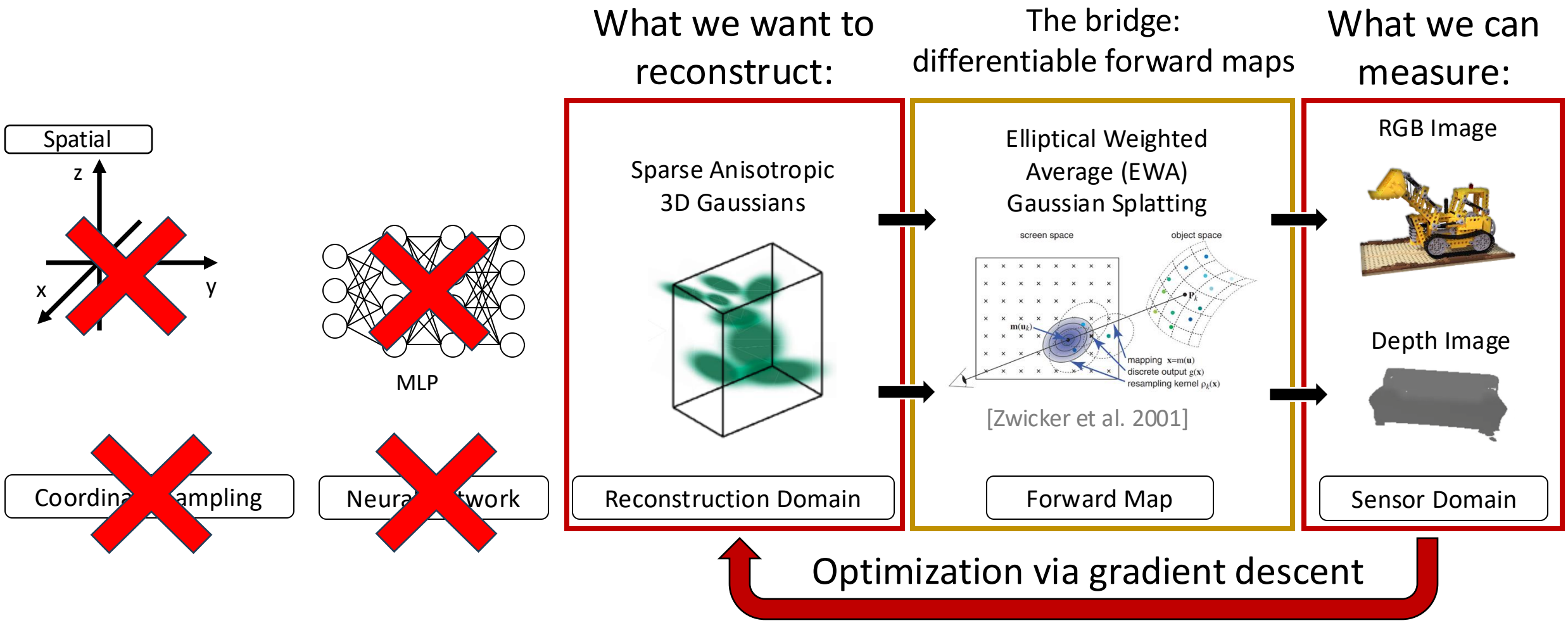


- **Rasterization**
- Projects scene onto the camera's image plane



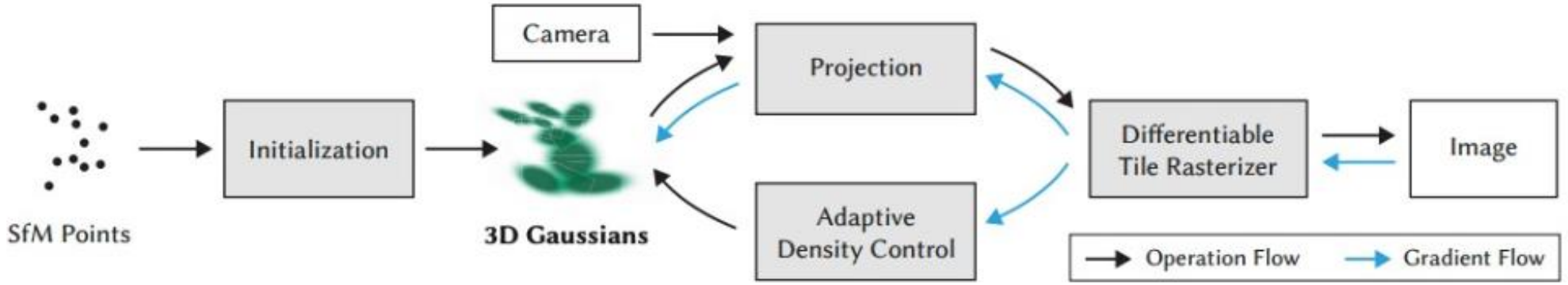
Radiance Fields without Neurons

3D Gaussian Splatting for Real-Time Radiance Field Rendering, (Kerbl, et al., 2023)



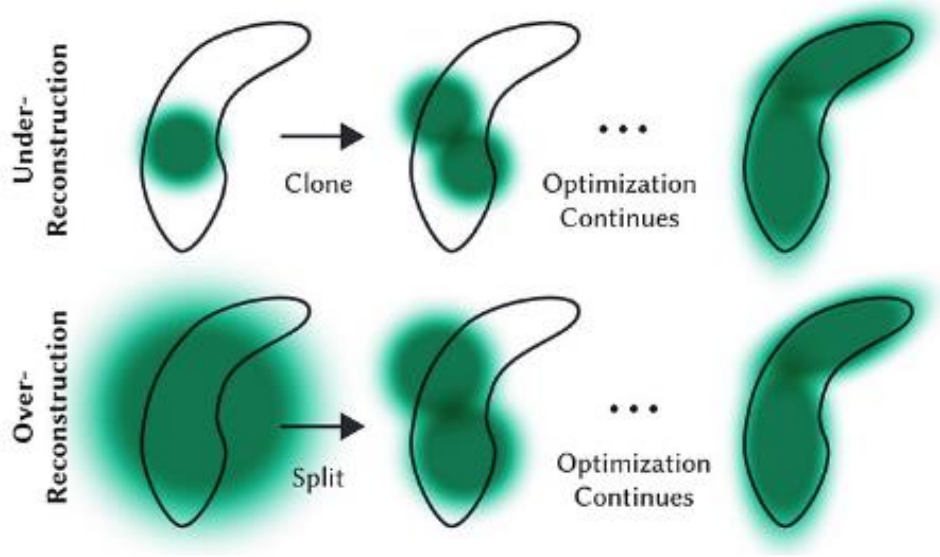
Gaussian Splatting – Training

- 1. Initialize splats with the sparse points from SfM. Set the covariance matrix to be the identity matrix.
- 2. Update the splats parameters with L1+SSIM loss and gradient descents.
- 3. Clone splats in the under-reconstructed regions, and split splats in the over-reconstructed regions.



Gaussian Splatting – Training

- 1. Initialize splats with the sparse points from SfM. Set the covariance matrix to be the identity matrix.
- 2. Update the splats parameters with L1+SSIM loss and gradient descents.
- 3. Clone splats in the under-reconstructed regions, and split splats in the over-reconstructed regions.



Conclusions and directions

- Things still hard
  - Difficult scenes
  - Better integrations with traditional 3D pipelines
    - editing, repurposing, animating, etc.
  - Integration in Learning Pipelines
    - They are a richer representation than images so you could eventually extract more…
    - “Higher-level semantic understanding within neural fields.
  - Priors
    - Most of the approaches exploit a zero knowledge approach optimizing only on the provided dataset
    - World is boring…
  - Dynamic Scenes

Learning Cameras

- One common (weak) point of both nerf and gaussian splatting approach:

Requiring Camera positions

  - All these approaches rely on standard photogrammetric approaches to recover the cameras

Recovering Cameras

- DUSt3R (**Dense and Unconstrained Stereo 3D Reconstruction**)**2024** introduces a novel paradigm in multi-view 3D reconstruction, **eliminating** the need for predefined **camera poses and intrinsics (works for pair of images)**

- **MUSt3R: Multi-view Network for Stereo 3D Reconstruction 2025**

