Hands-on AI based 3D Vision– Summer 25

Lecture 6_1 – Neural Fields

Prof. Dr.-Ing. Gerard Pons-Moll
University of Tübingen / MPI-Informatics

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

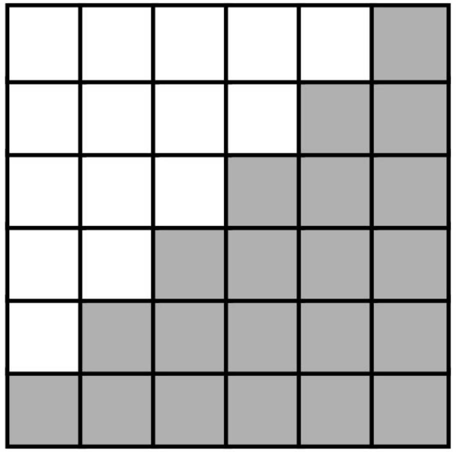
What is a good representation for 3D data?

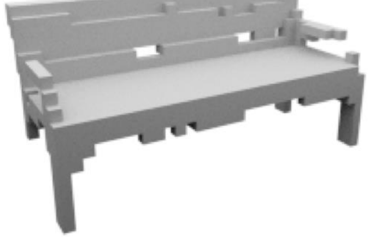
What is a good representation for 3D data?

- Compatible with neural networks.
- Flexible
- High fidelity

Voxels

- Discretization of 3D space into grid.
- Easy to process with neural networks.
- Cubic memory $\mathcal{O}(n^3) \longrightarrow$ limited resolution.



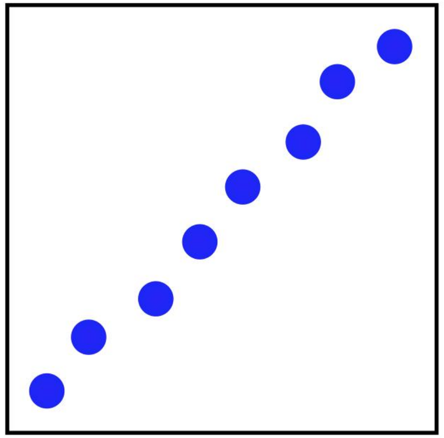


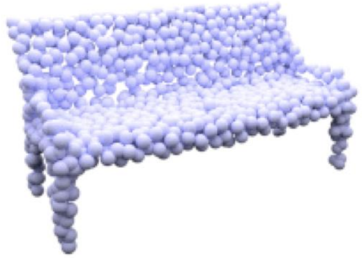
[Liao et al. CVPR'18]
[Chov et al. ECCV'16]

Image credit: Mescheder et al. CVPR'19

Pointclouds

- Discretization of 3D space into 3D points.
- Does not model connectivity/topology.
- Limited number of points.



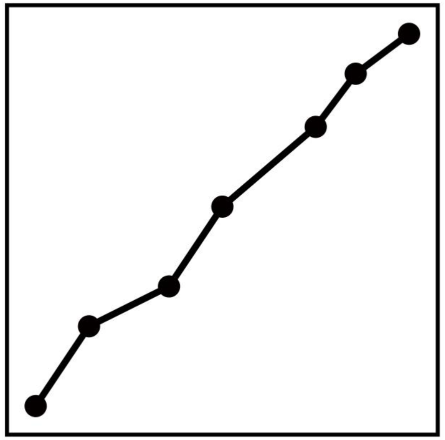


[Liao et al. CVPR'18]
[Chov et al. ECCV'16]

Image credit: Mescheder et al. CVPR'19

Meshes

- Discretization into vertices and faces.
- Limited number of vertices/granularity.
- Requires class specific template.
- Leads to self-intersections.

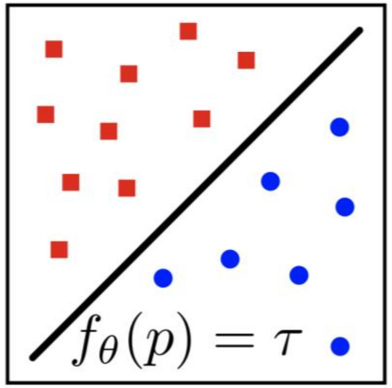


[Wang et al. ECCV'18]

Image credit: Mescheder et al. CVPR'19

Implicit representation

- Implicit representation $\rightarrow$ No discretization.
- Arbitrary topology and resolution.
- Low memory footprint.
- Not restricted to specific class.



[Mescheder et al. CVPR'19]
[Chen et al. CVPR'19]
[Park et al. CVPR'19

Image credit: Mescheder et al. CVPR'19

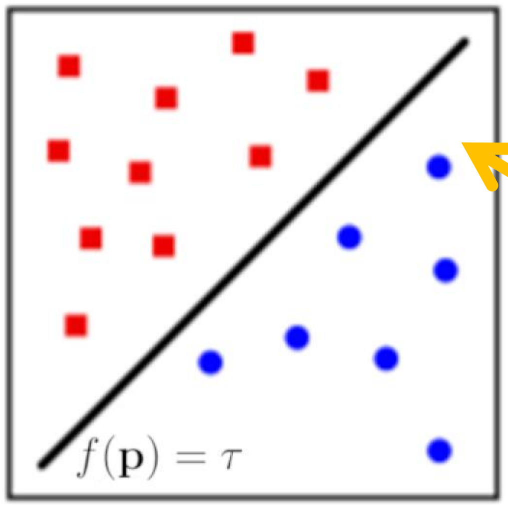
Surfaces as an Implicit Function

A function tells us whether a point is inside or outside an object

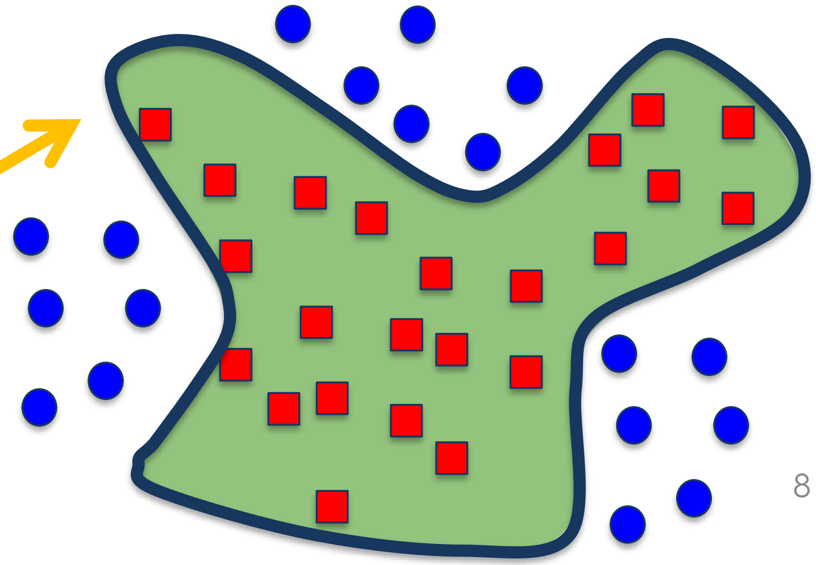
$$f(\mathbf{p}) = \begin{cases} 0, & \text{if} \quad \mathbf{p} \in \text{outside} \\ 1, & \text{if} \quad \mathbf{p} \in \text{inside} \end{cases}$$

$$\mathbf{p} = (x, y, z) \in \mathbb{R}^3$$

If the function is continuous, a levelset of it defines a surface $\mathcal{S}$



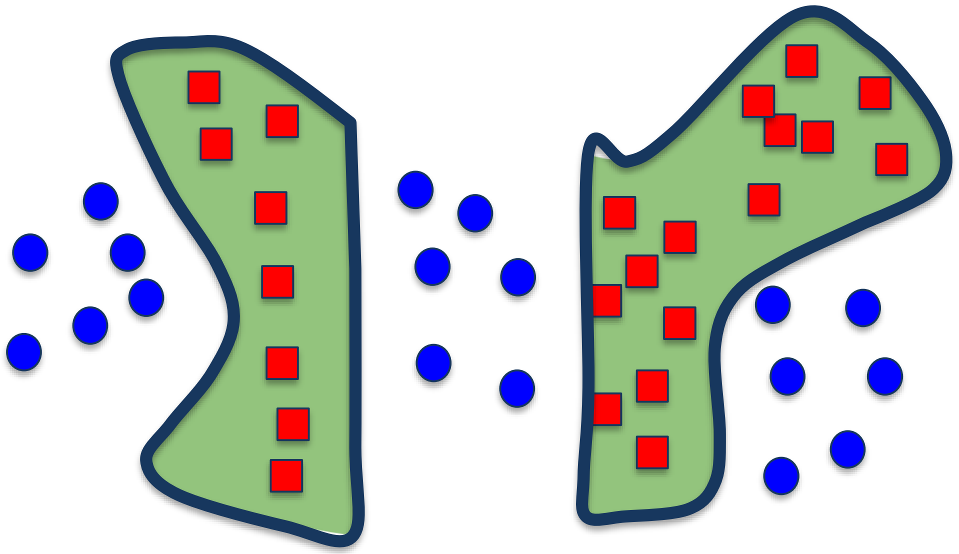
$$\mathcal{S} = \{\mathbf{p},\ f(\mathbf{p}) = \tau\}$$

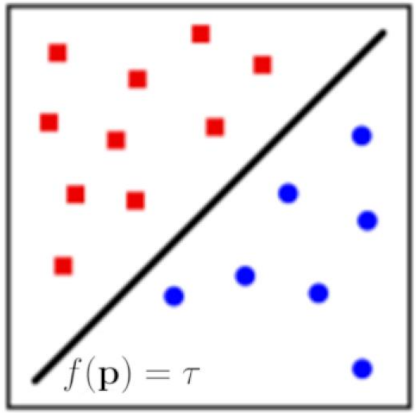


Surfaces as an Implicit Function

$$\mathbf{p} = (x, y, z) \in \mathbb{R}^3$$

$$f(\mathbf{p}) = \begin{cases} 0, & \text{if} \quad \mathbf{p} \in \text{outside} \\ 1, & \text{if} \quad \mathbf{p} \in \text{inside} \end{cases}$$





$$\mathcal{S} = \{\mathbf{p},\ f(\mathbf{p}) = \tau\}$$

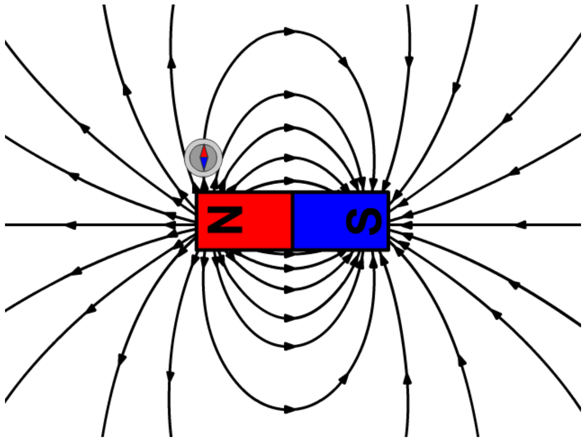
✓ With implicit functions, Topology changes only require changing $f(\mathbf{p})$

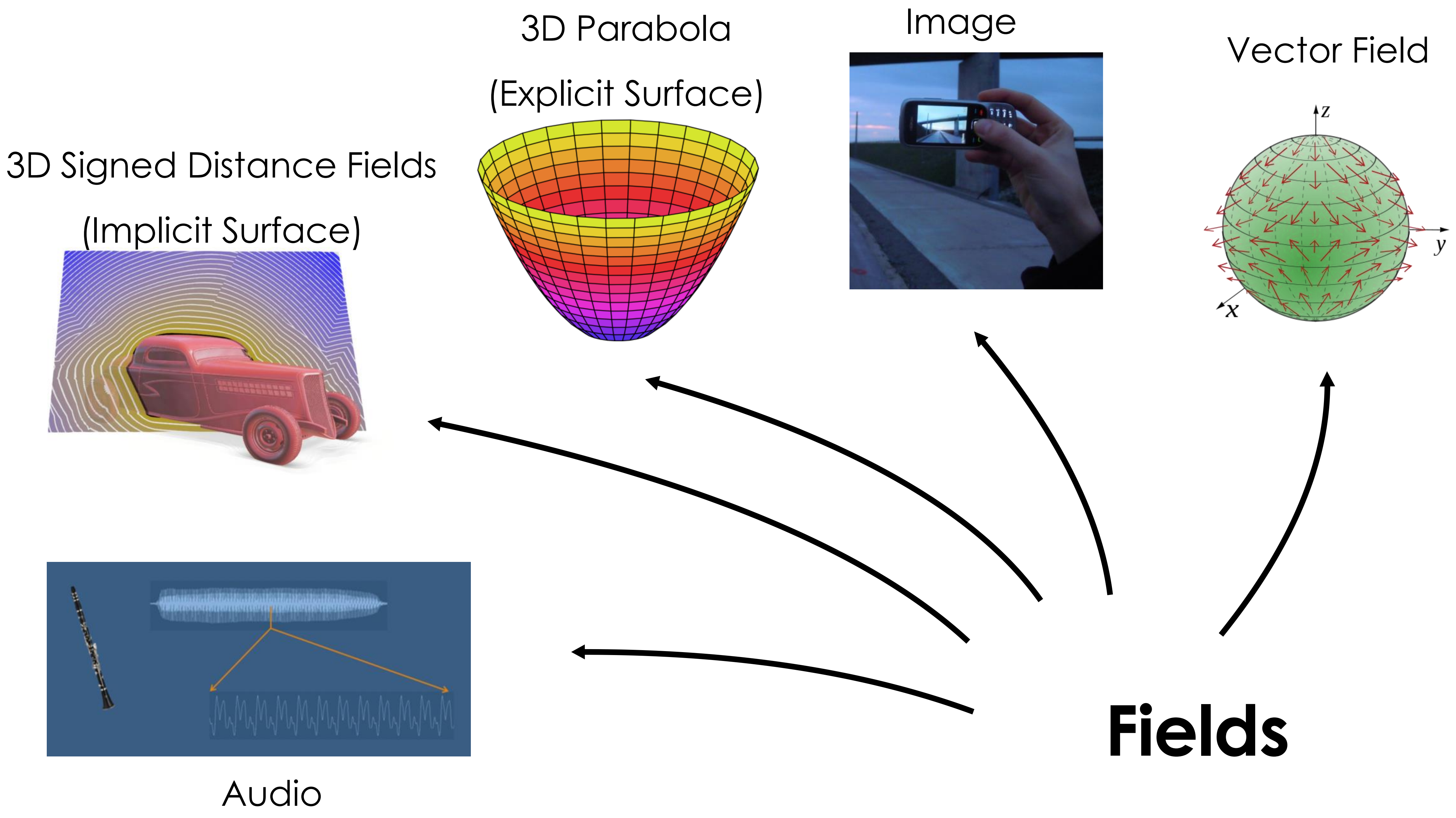
✗ Mesh based representations would struggle

Neural Fields

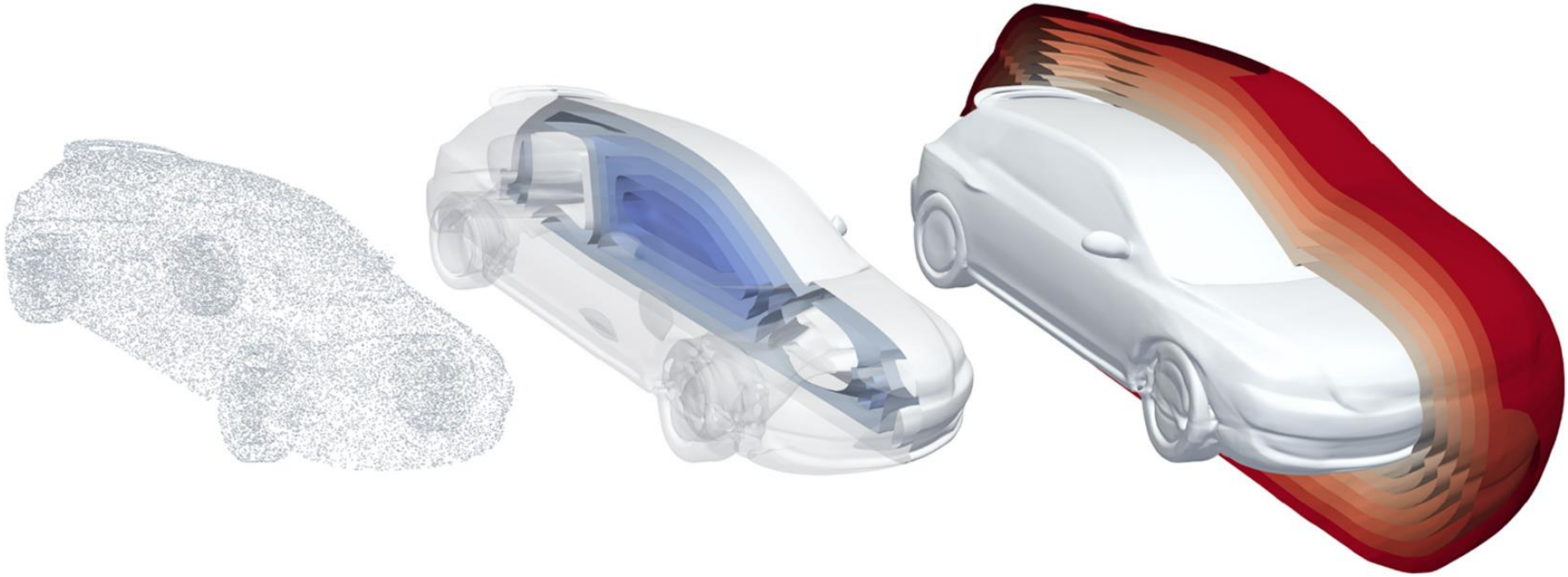
What is a Field? (Physics)

A **Field** is a mapping associating a scalar or vector quantity to each point in space




3D Parabola
(Explicit Surface)
Image
Vector Field
3D Signed Distance Fields
(Implicit Surface)
Audio
Fields

You have already seen fields!



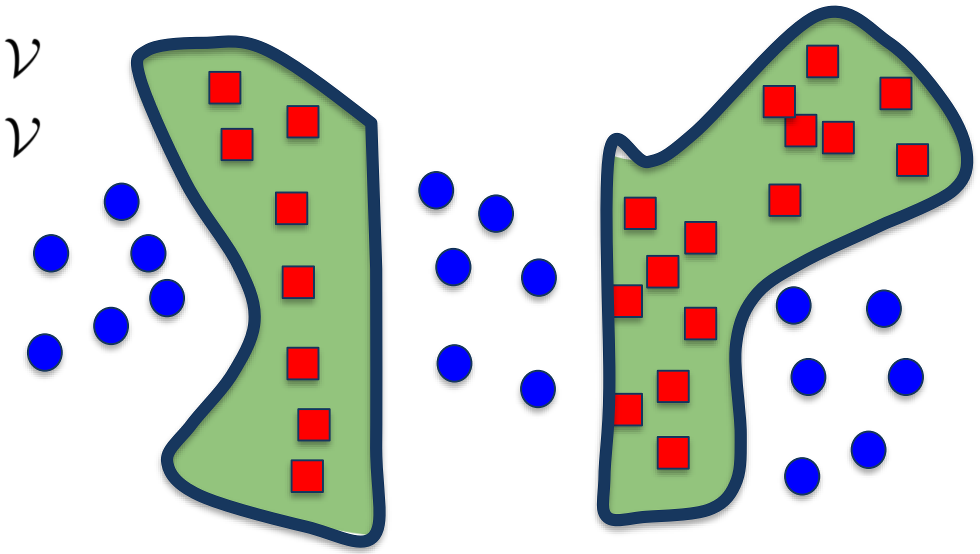
Implicit Geometric Regularization for Learning Shapes. Gropp et al.

You have already seen fields!

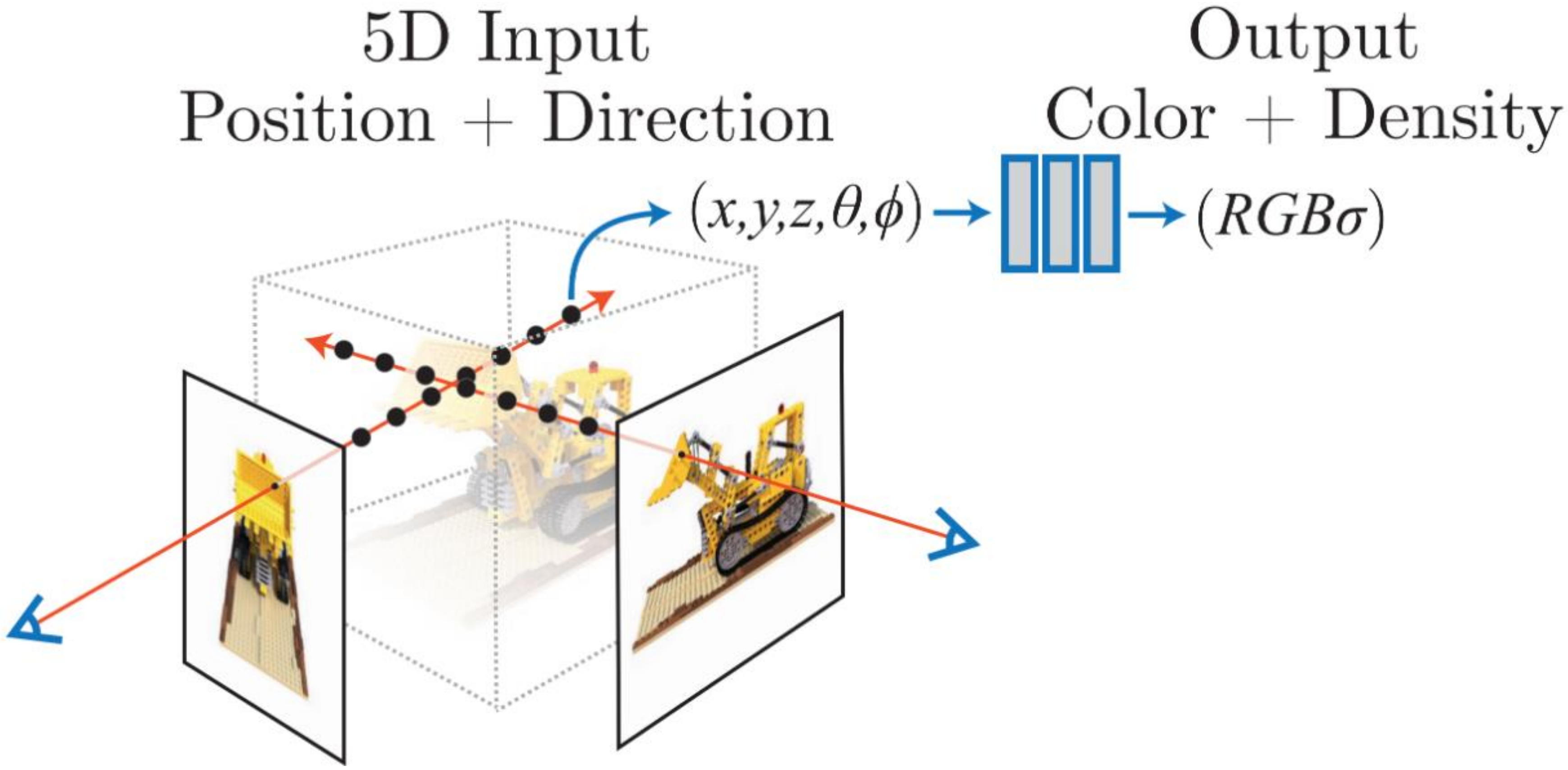
Occupancy Fields: $f(\mathbf{p}) = \begin{cases} 1, & \text{if } \mathbf{p} \in \mathcal{V} \\ 0, & \text{if } \mathbf{p} \notin \mathcal{V} \end{cases}$

Unsigned Distance Fields: $f(\mathbf{p}) = \mathrm{dist}(\mathbf{p}, \mathcal{S})$

Signed Distance Fields: $f(\mathbf{p}) = \begin{cases} \mathrm{dist}(\mathbf{p}, \mathcal{S}), & \text{if } \mathbf{p} \in \mathcal{V} \\ -\mathrm{dist}(\mathbf{p}, \mathcal{S}), & \text{if } \mathbf{p} \notin \mathcal{V} \end{cases}$

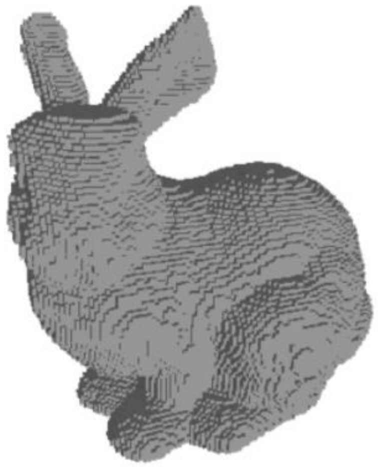


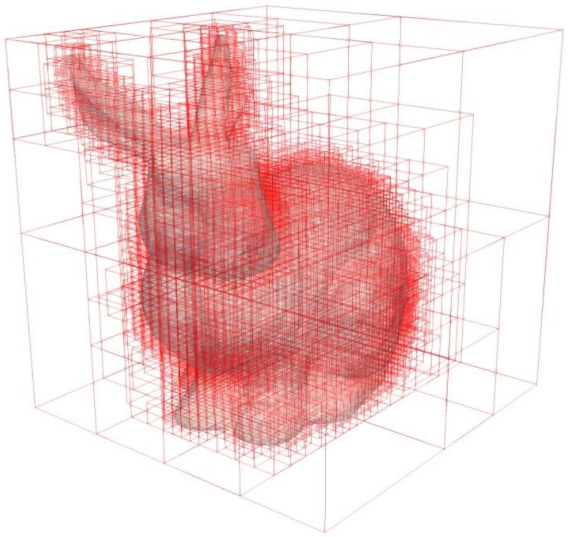
Fields can be anything



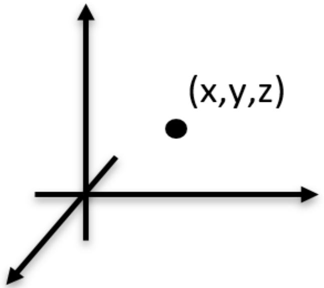
Nerf: Representing scenes as neural radiance fields for view synthesis.
Mildenhall et al.

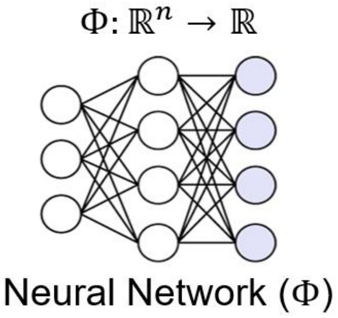
Parameterizing Fields



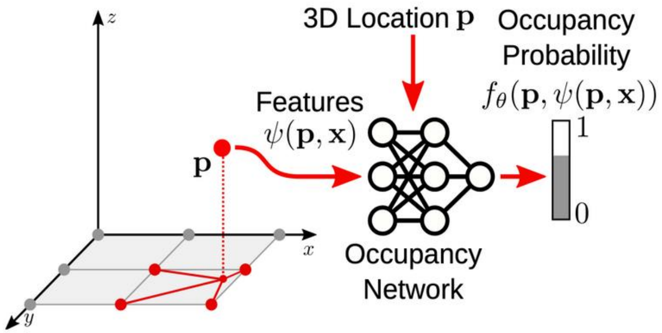


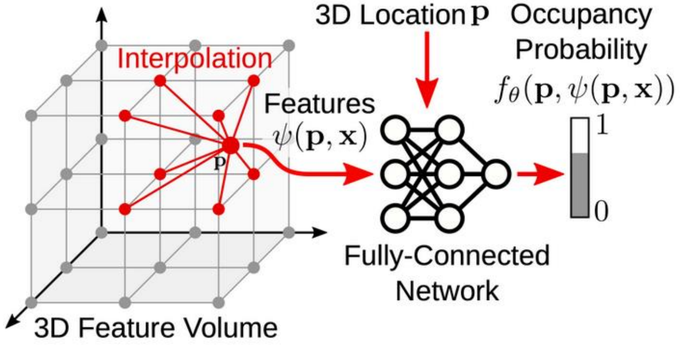
Discrete





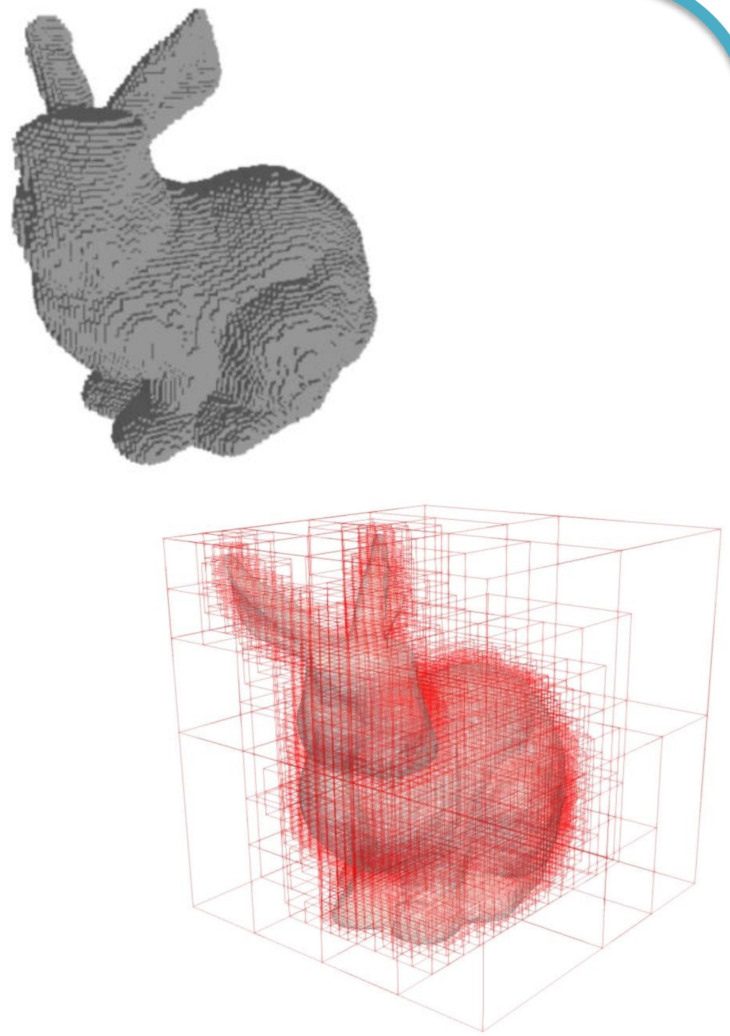
Neural



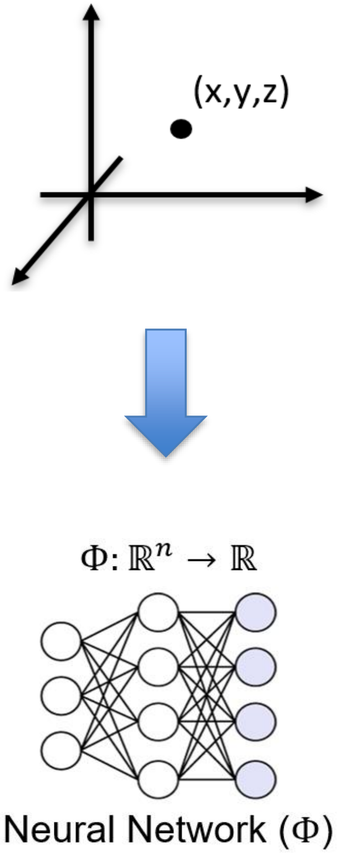


Hybrid

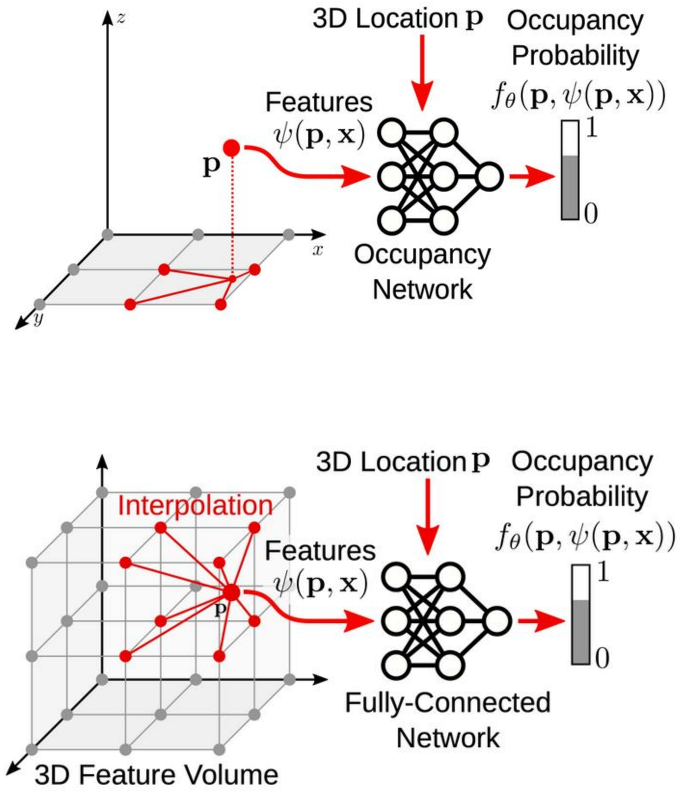
Parameterizing Fields



Discrete



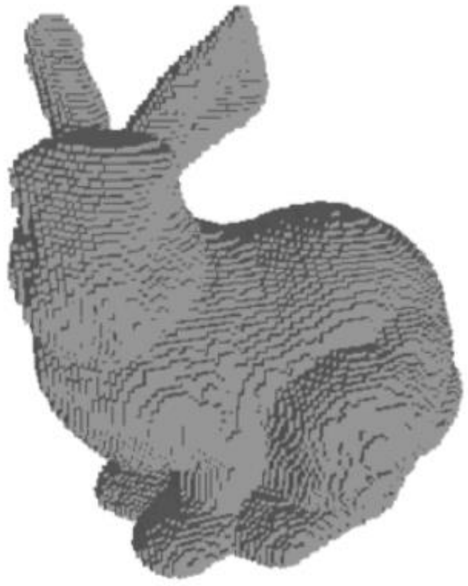
Neural

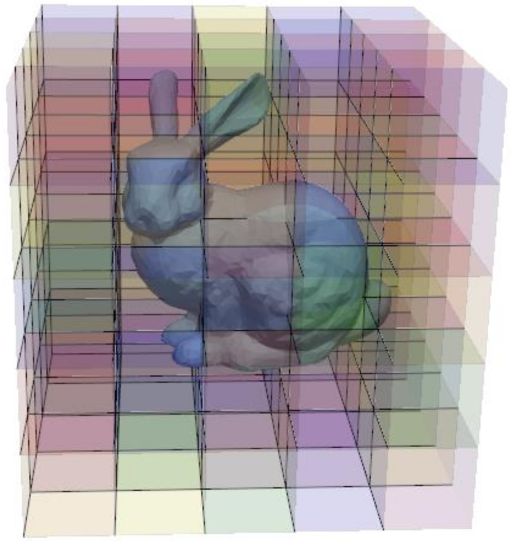


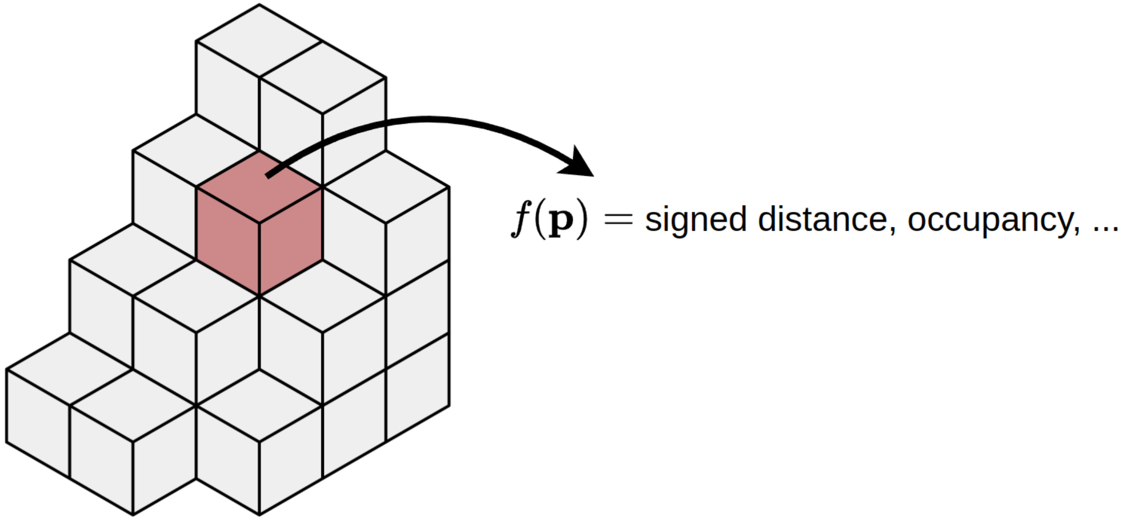
Hybrid

Voxel Grid

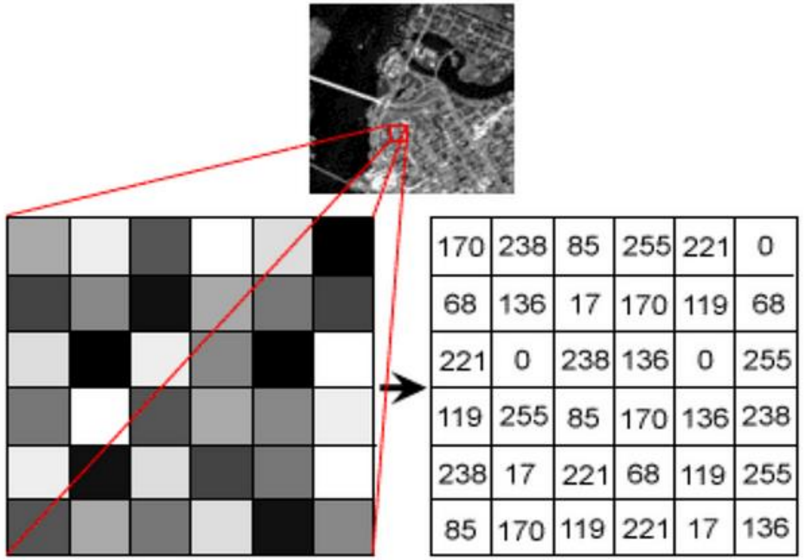
- Subdivide 3D space into voxel grid of equal sized bins
- For each voxel store field value

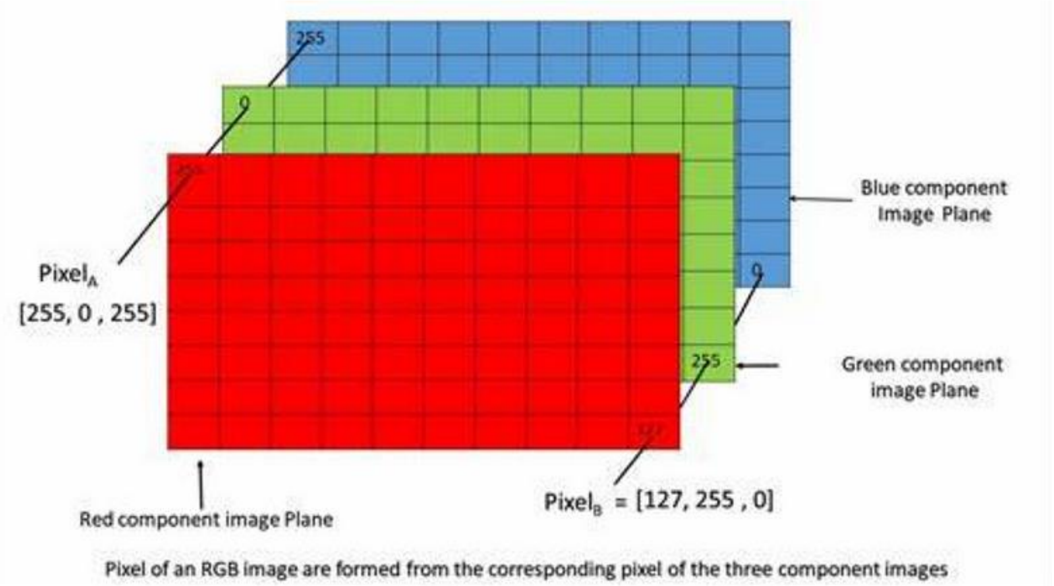






Images are 2D Voxel Grids





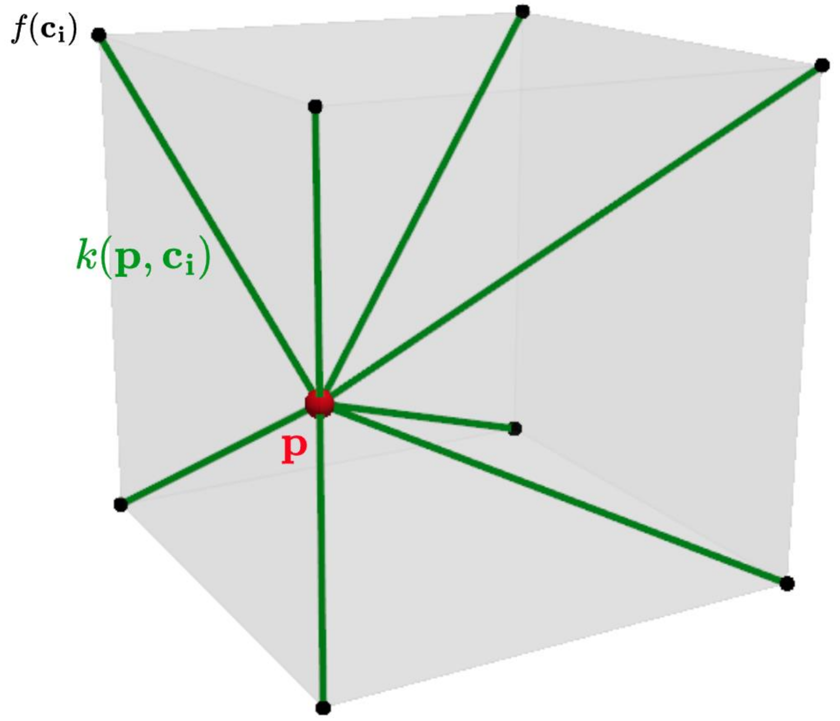
Continuous Fields with Voxels

- Values can also be stored at corners of the voxels
- For a query point $\mathbf{p} \in \mathbb{R}^3$ interpolate via

$$f(\mathbf{p}) = \sum_{c_i} f(\mathbf{c}_i) k(\mathbf{p}, \mathbf{c_i})$$

Stored value at corner *Kernel function*

E.g. $k(\mathbf{p}, \mathbf{c_i}) = \frac{d(\mathbf{p}, \mathbf{c}_i)}{\sum_{c_j} d(\mathbf{p}, \mathbf{c}_j)}$



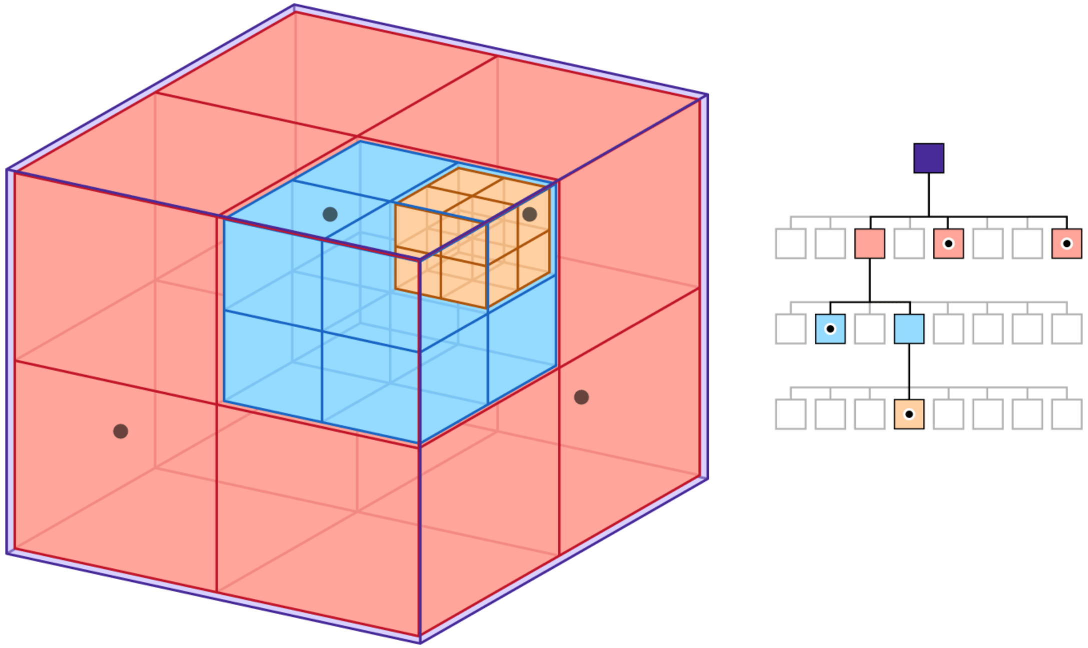
Properties of Voxel Grids

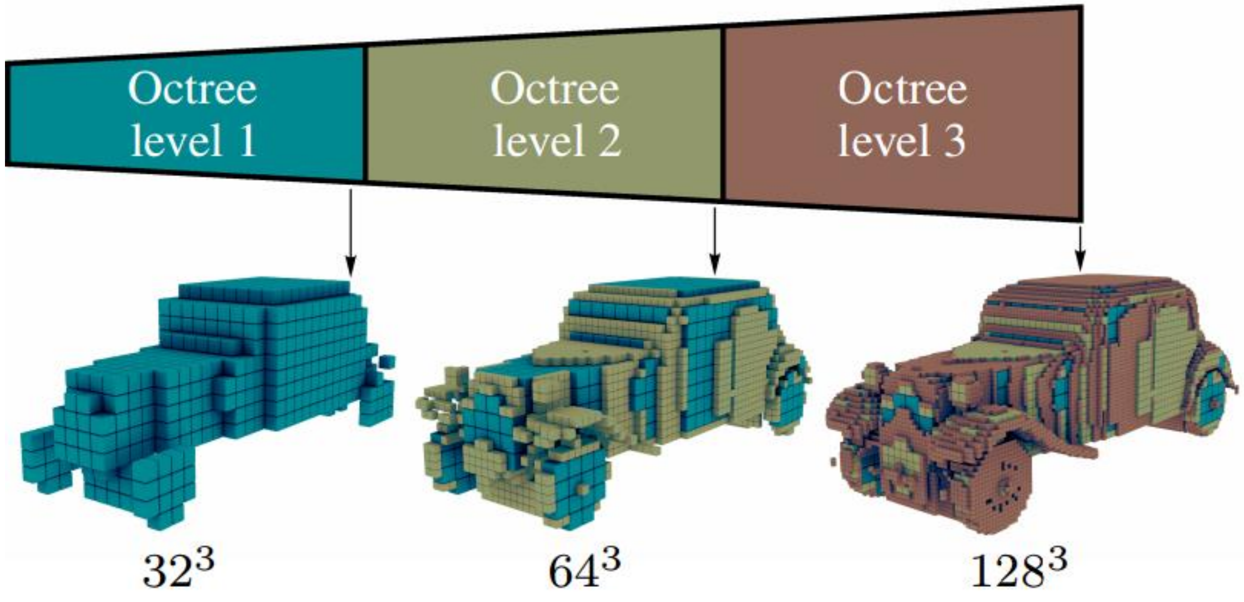
- For *d* input dimensions, and *n* bins per dimension the memory required is $\mathcal{O}(n^d)$
- Intractable in higher dimensions
- For 3D scenes: Lots of voxels store no useful information

- Retains locality information
- Easy to extend CNNs to 3D convolutions

Octrees

- Divide voxels into 8 subvoxels if it has more than N points in it
- Depth of octree depends on desired resolution



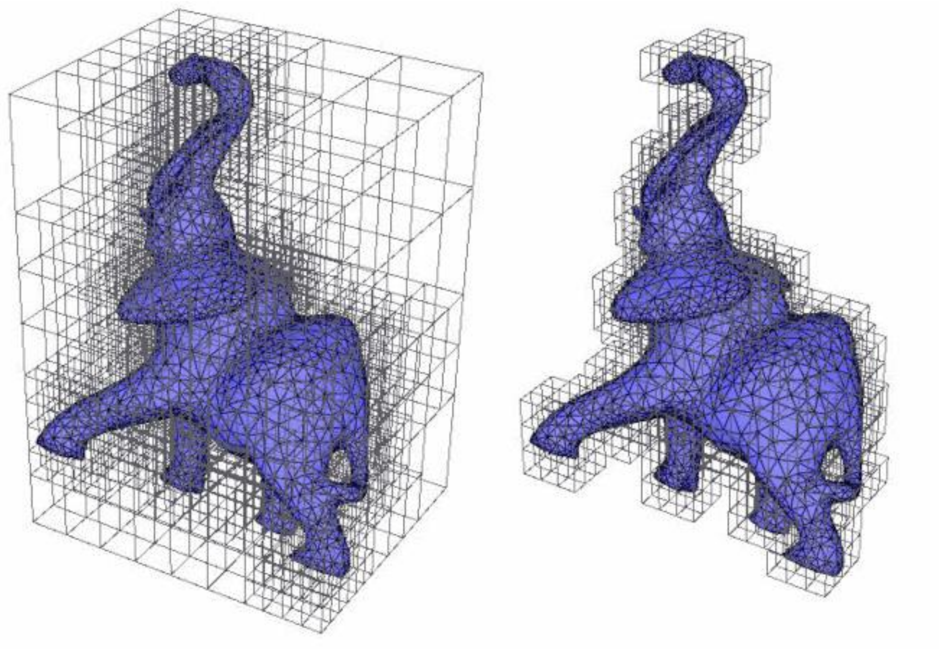


https://dulalsaurab.github.io/assets/img/octree.png

https://lmb.informatik.uni-freiburg.de/people/tatarchm/ogn/

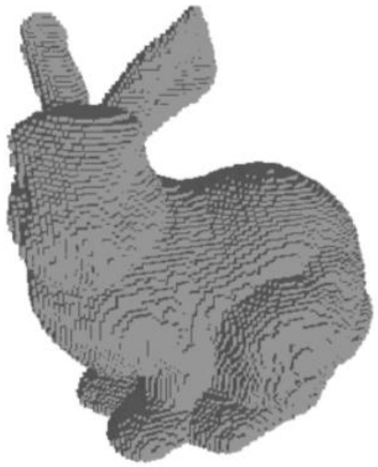
Discussion of Octrees

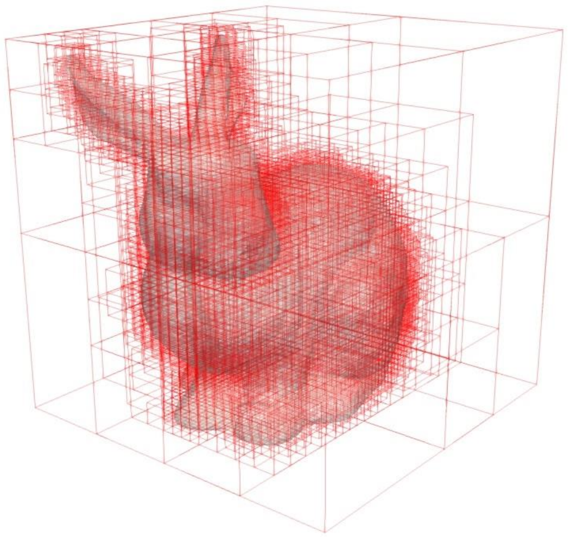
- Reduces memory wasted by empty space
- Still memory extensive (depending on desired resolution)
- More expensive to construct
- Sub-division is not differentiable
- To sample a point, have to traverse hierarchy



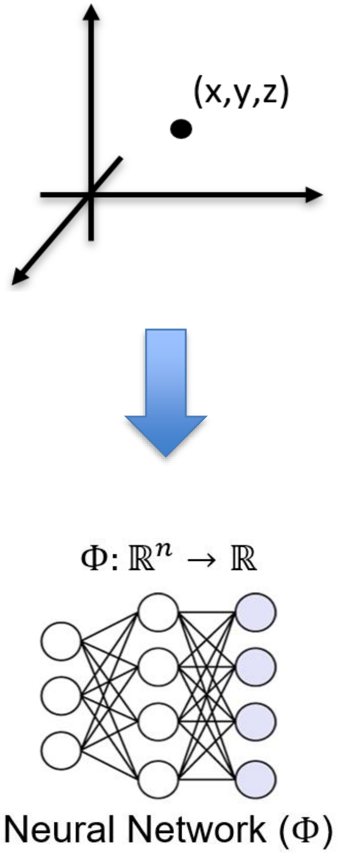
https://doc.cgal.org/latest/Orthtree/index.html

Parameterizing Fields

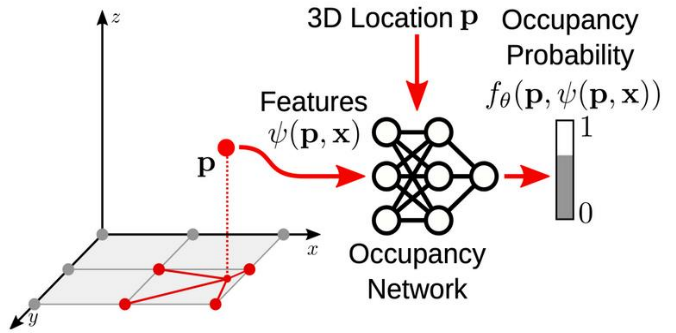


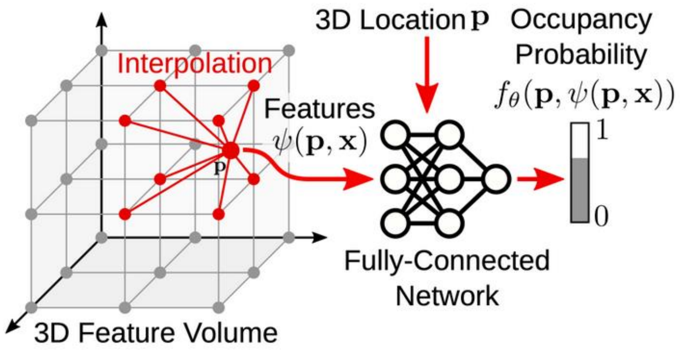


Discrete



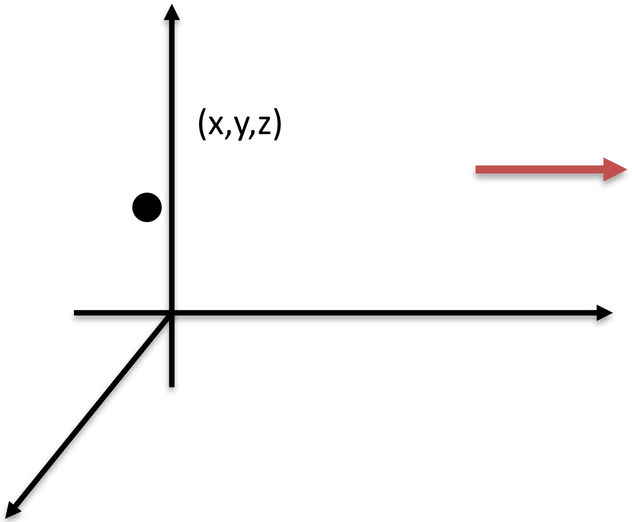
Neural



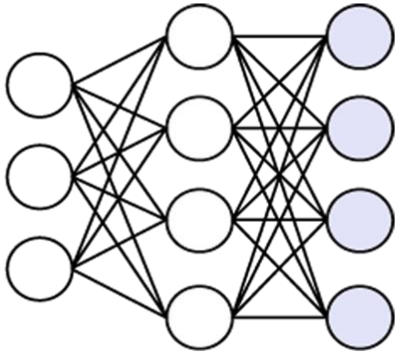


Hybrid

Neural Fields

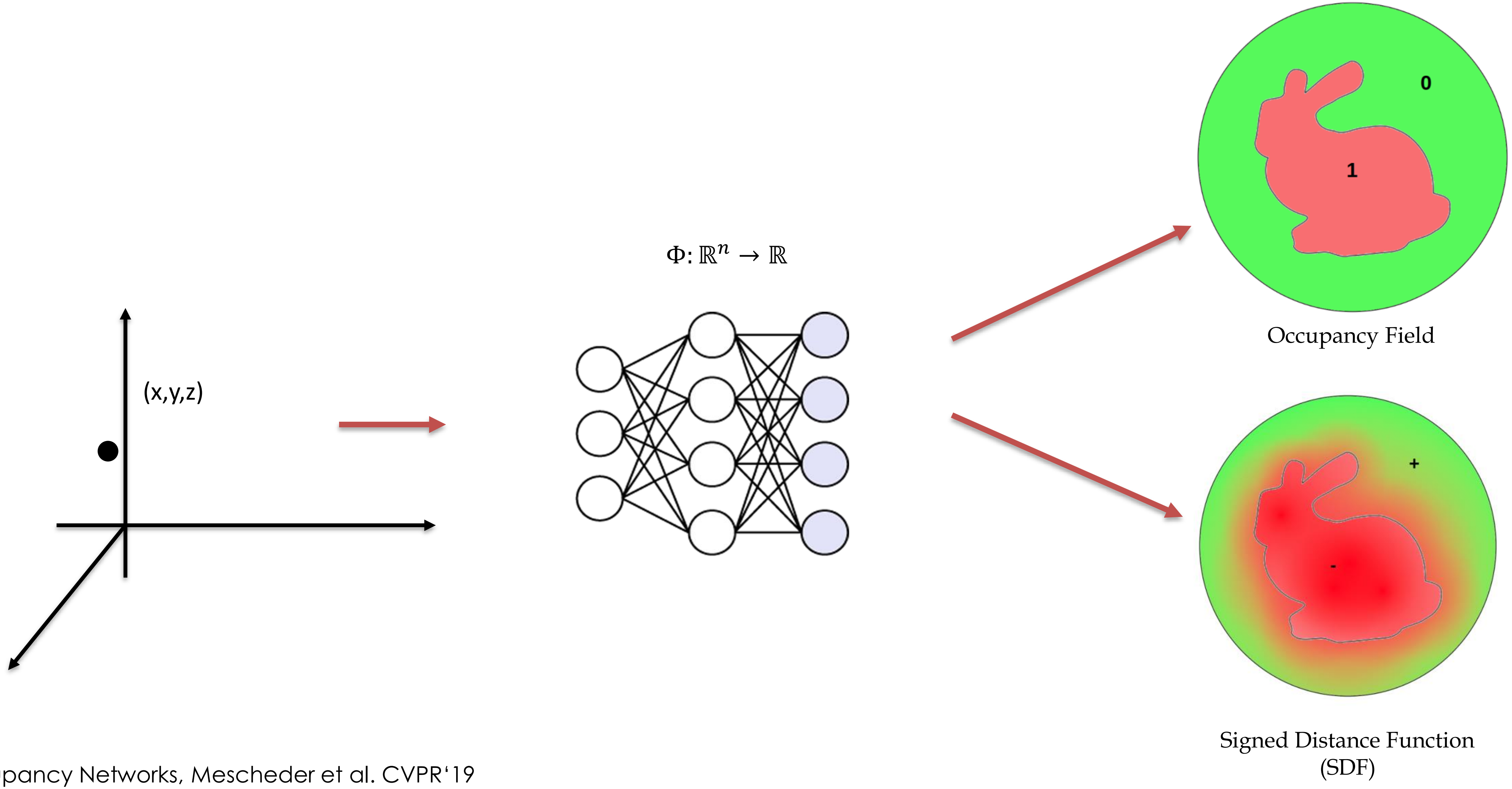


$\Phi: \mathbb{R}^n \rightarrow \mathbb{R}$



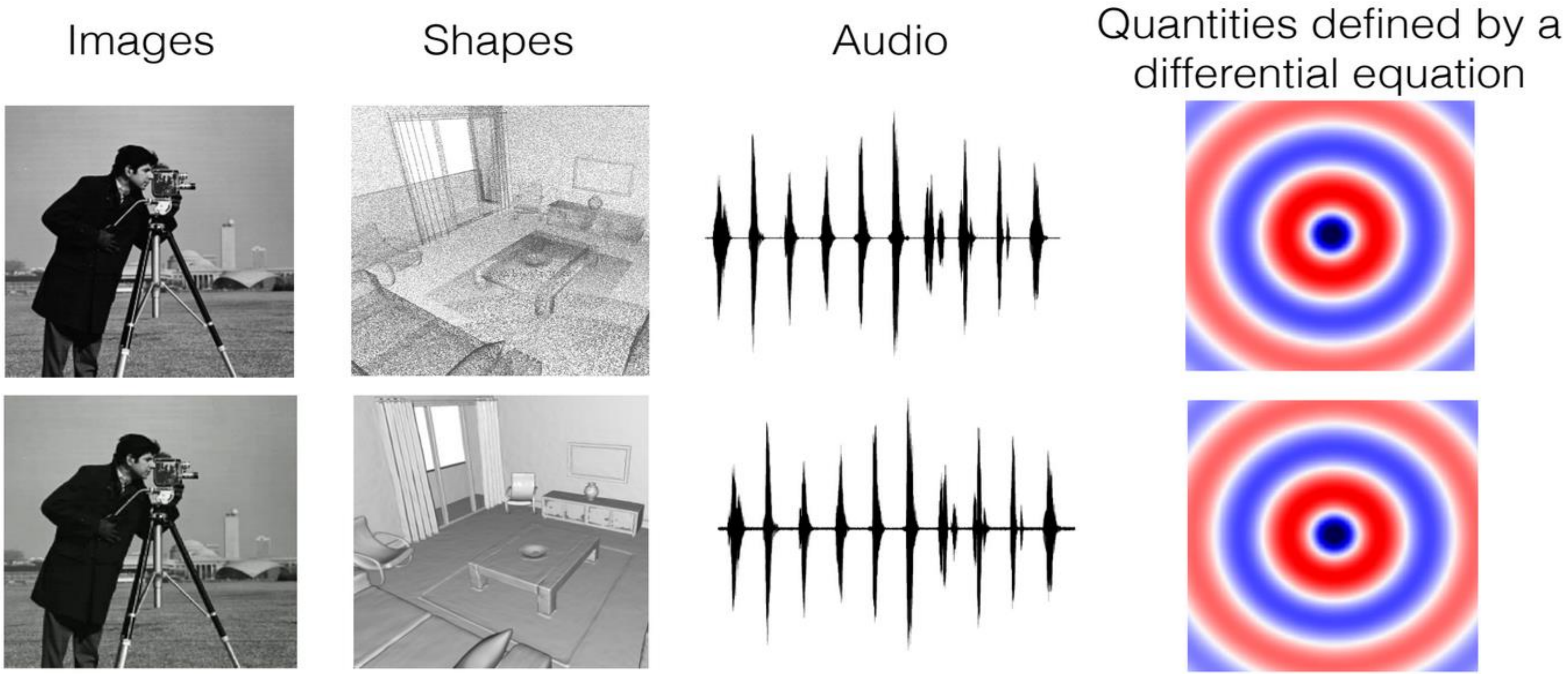
Signed Distance Function (SDF)

Neural Fields for 3D Scenes



Occupancy Networks, Mescheder et al. CVPR'19
IM-NET, Chen et al. CVPR'19
DeepSDF, Park et al. CVPR'19

Neural Fields For Other Data



SIREN, Sitzmann et al. NeurIPS '20

Memory

~1MB memory

Mesh: 110MB

SIREN, Sitzmann et al. NeurIPS '20

Discussion of Neural Fields

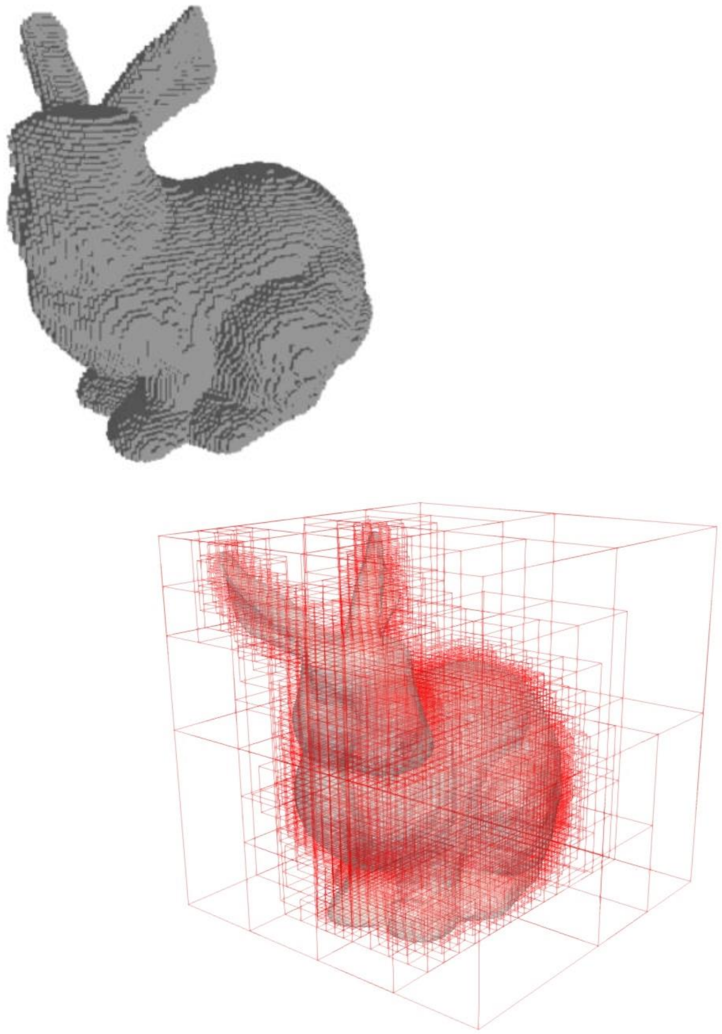
- Storage memory does not grow with spatial resolution or number of spatial dimensions
- Resolution chosen at test time
- Adaptive resolution: Assigns more compute to higher frequency areas
- Work for arbitary dimensions
- Slow training and sampling
- Does not expose locality
- Inconvenient processing: Cannot run convolutions
- Editing is hard

Comparison

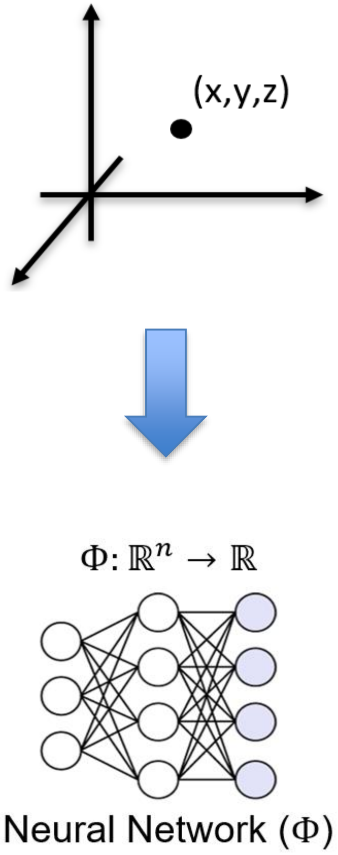
	Memory	Speed	Locality	Resolution
Discrete	✗	✓	✓	✗
Neural	✓	✗	✗	✓

Hybrid Parametrizations: Best of both worlds

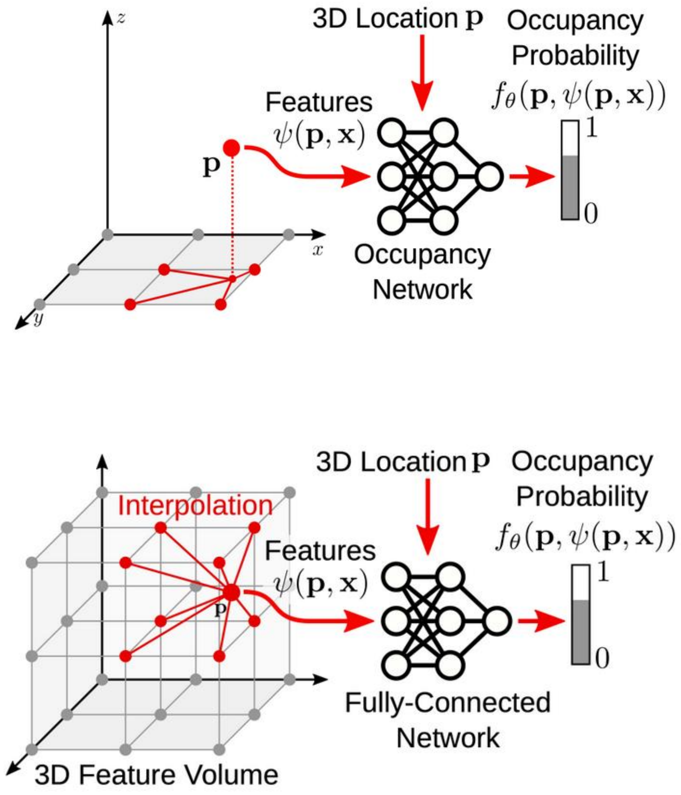
Parameterizing Fields



Discrete

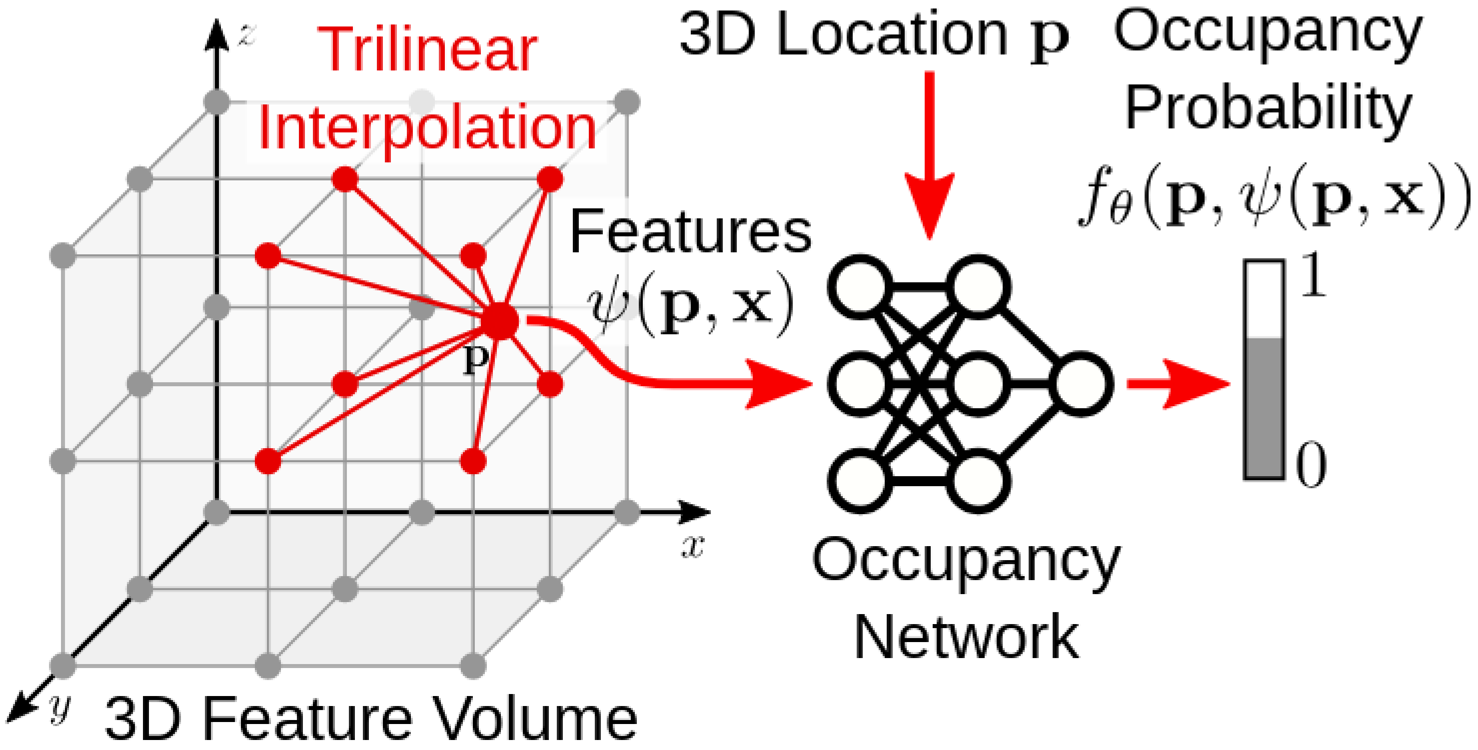


Neural



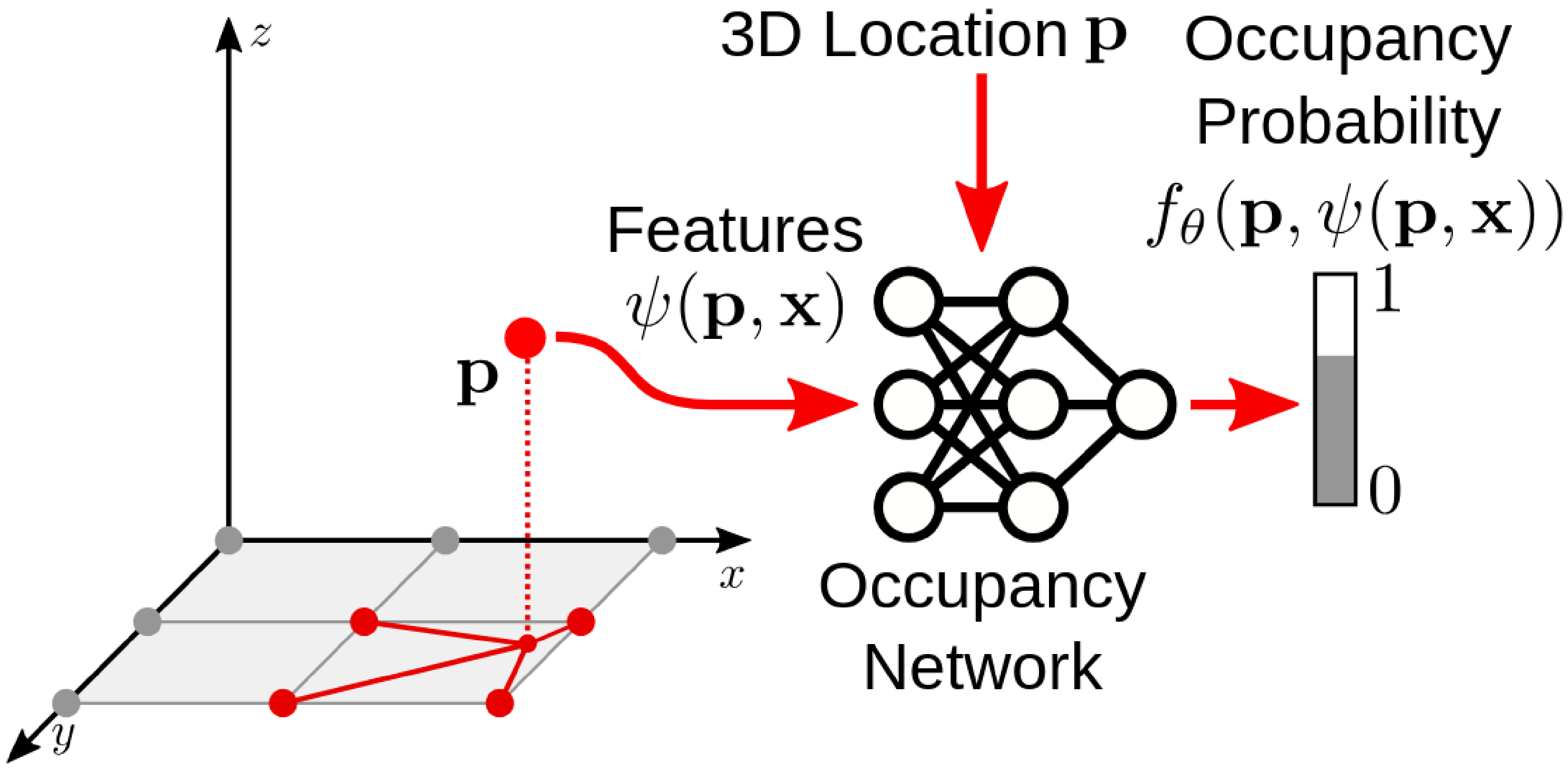
Hybrid

Voxel-Features + Neural Field



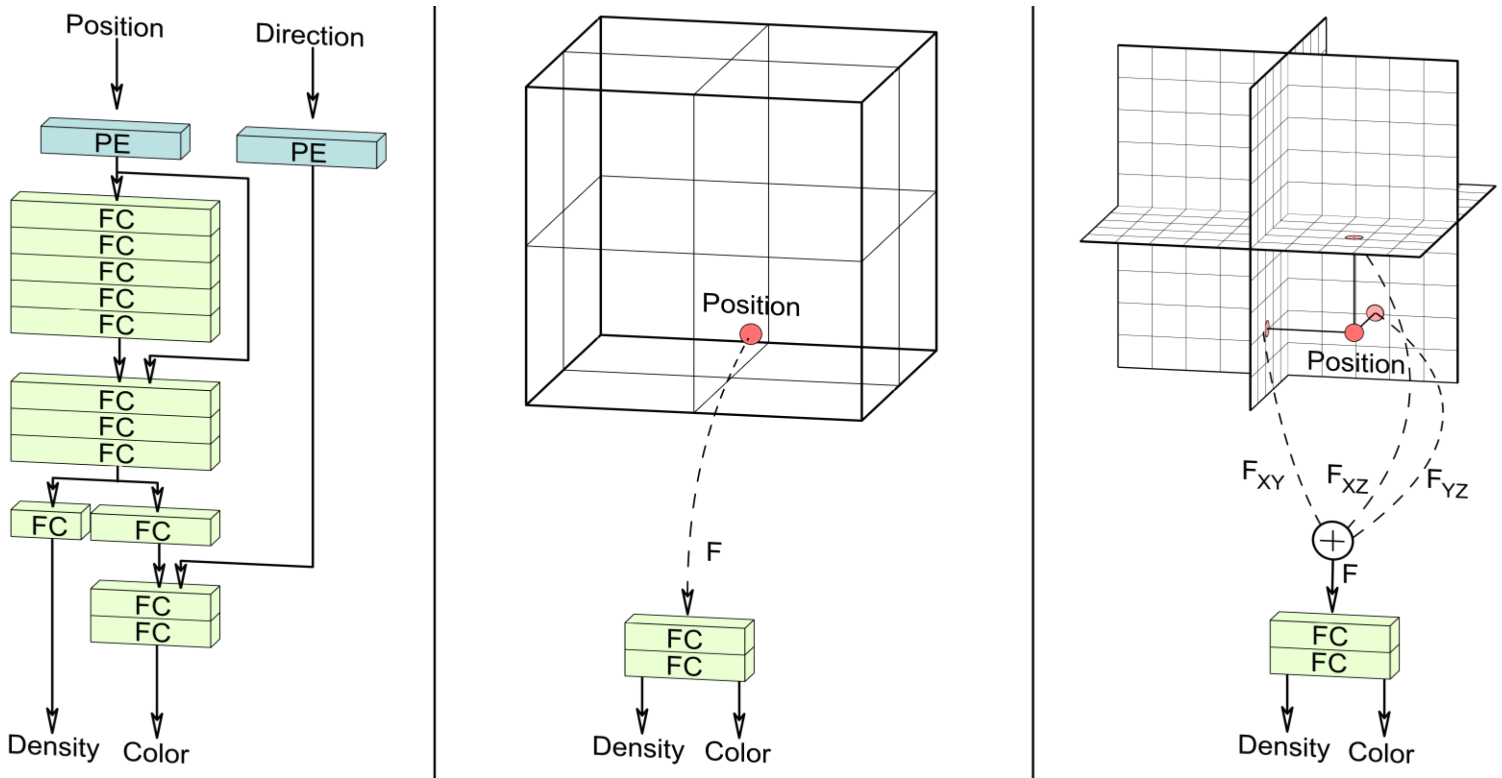
IF-Nets, Chibane et al. CVPR '20
Convolutional Occupancy Networks, Peng et al. ECCV '20
Neural Sparse Voxel Fields, Liu et al. NeurIPS '20

Orthographic Projection onto Feature Plane



Convolutional Occupancy Networks, Peng et al. ECCV '20

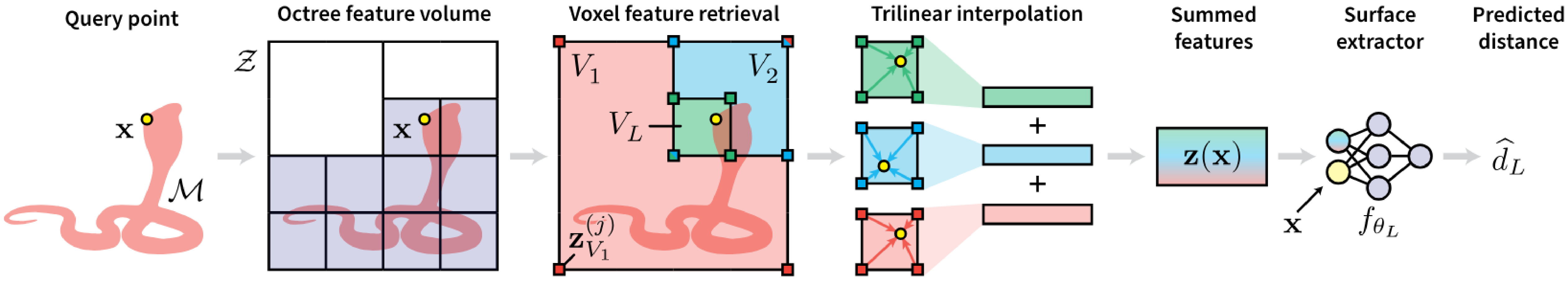
Triplane Representation



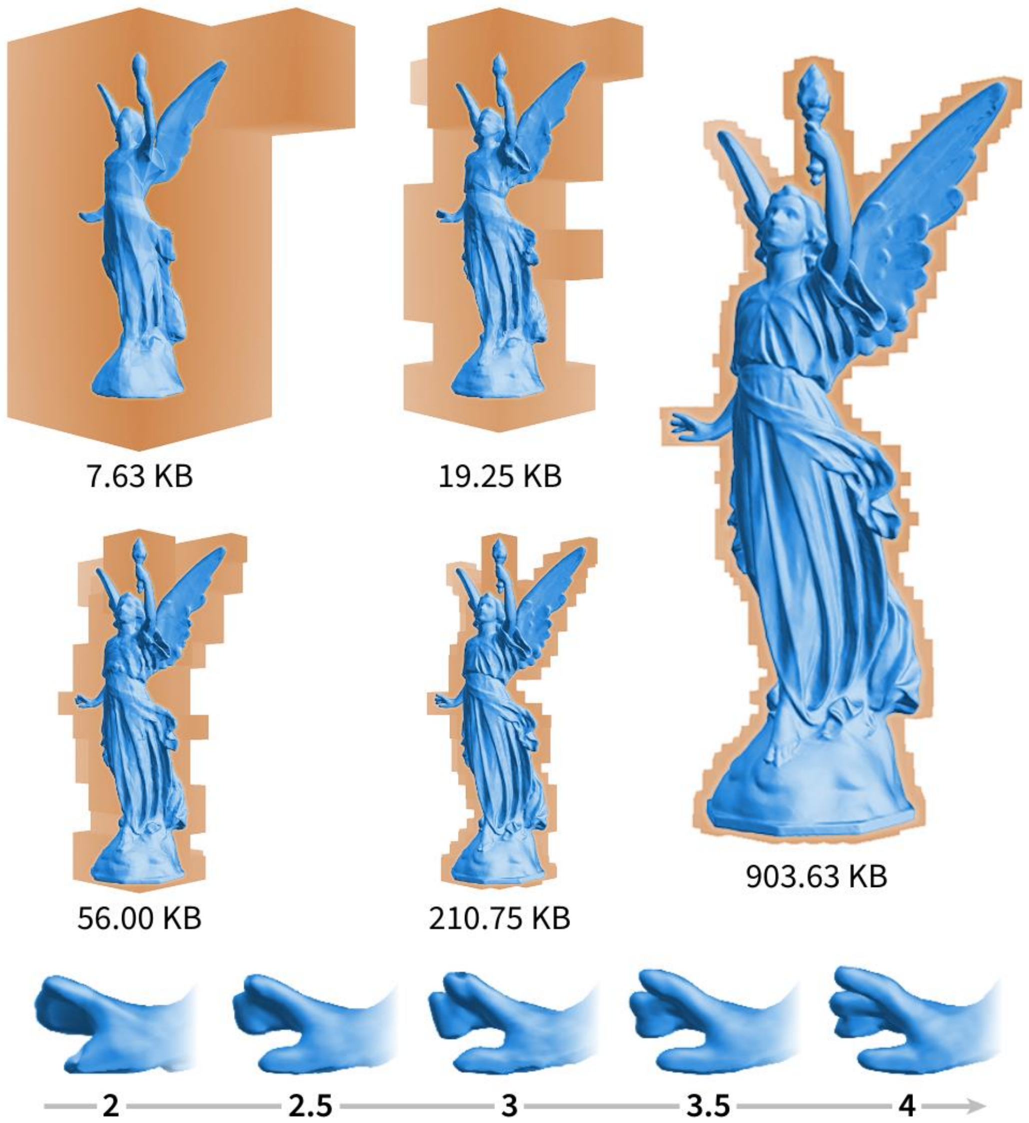
(a) NeRF (Implicit) (b) Voxels (Explicit or Hybrid) (c) Ours (Hybrid)

Convolutional Occupancy Networks, Peng et al. ECCV '20
EG3D, Chan et al. CVPR '22

Octree-Feature Volumes

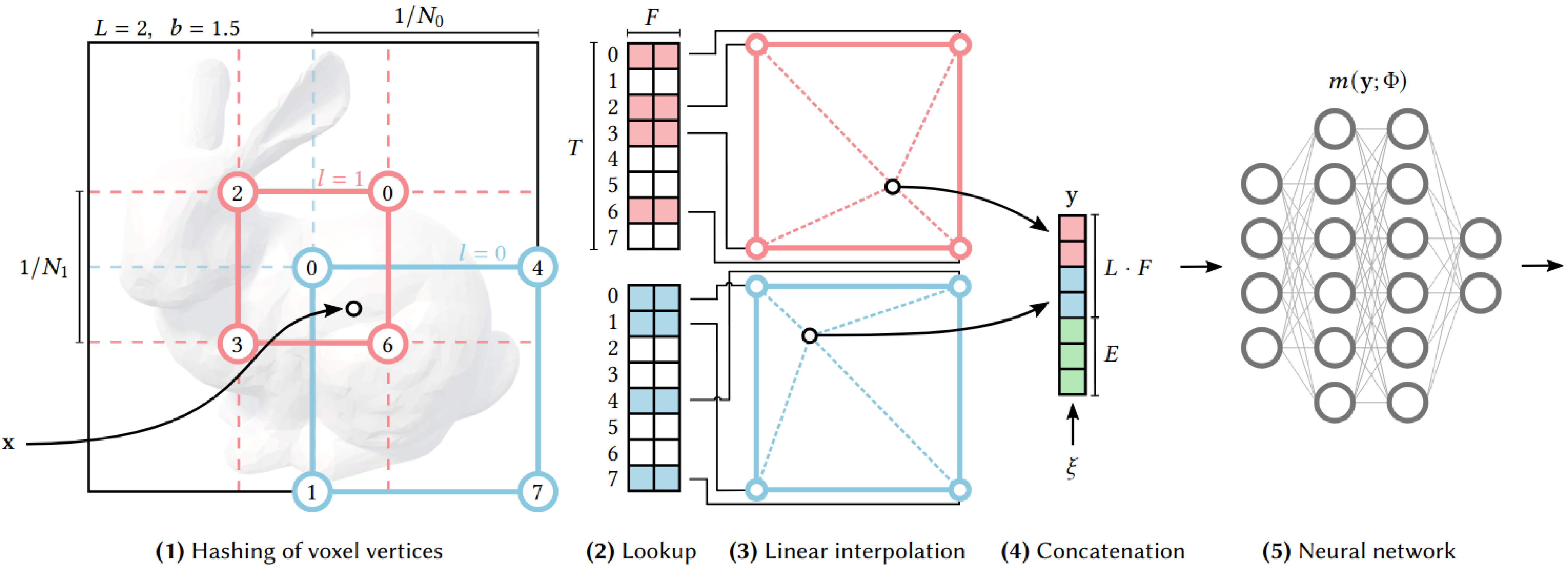


Octree-Feature Volumes



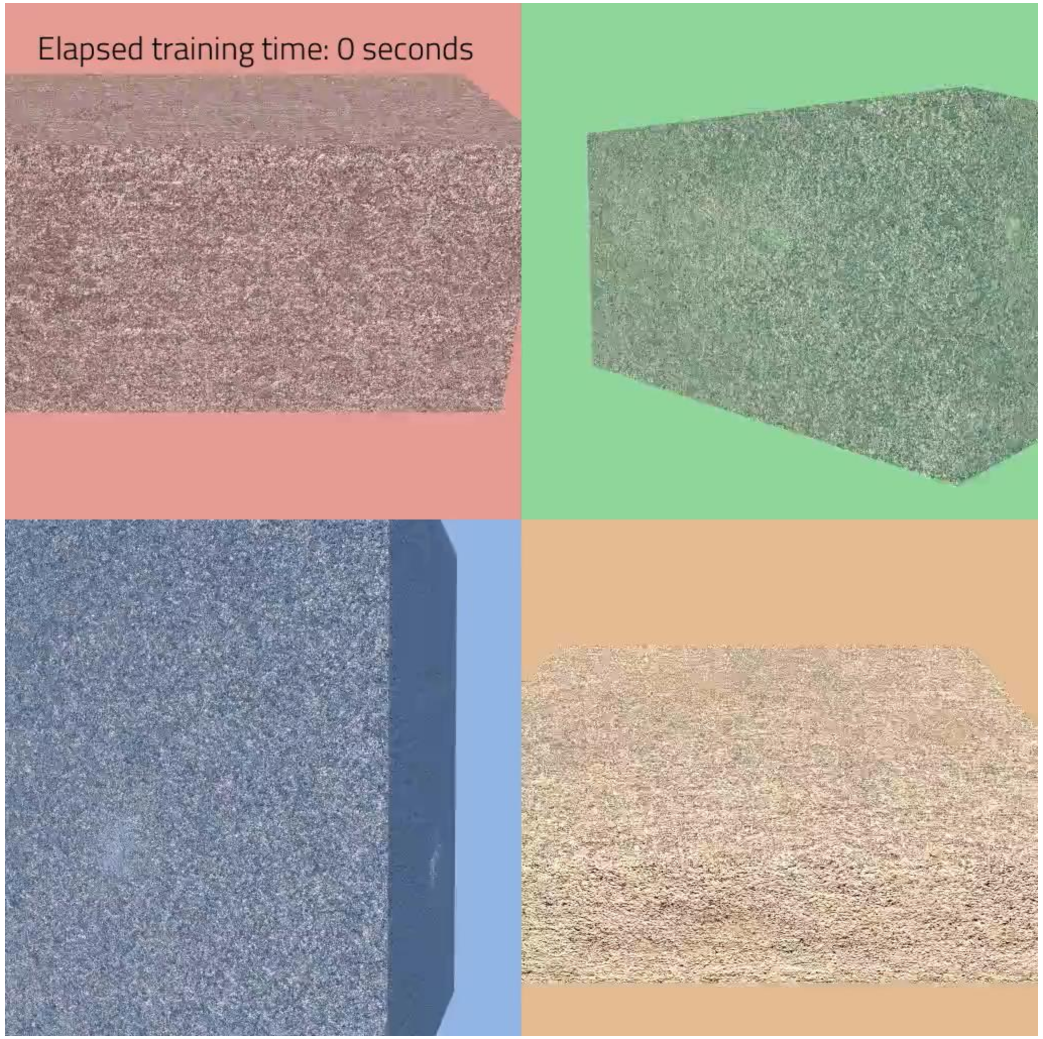
NGLOD, Takikawa et al. CVPR '21

Multi-Scale Voxel Grid + Hash Encoding



Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, Müller et al. SIGGRAPH '22

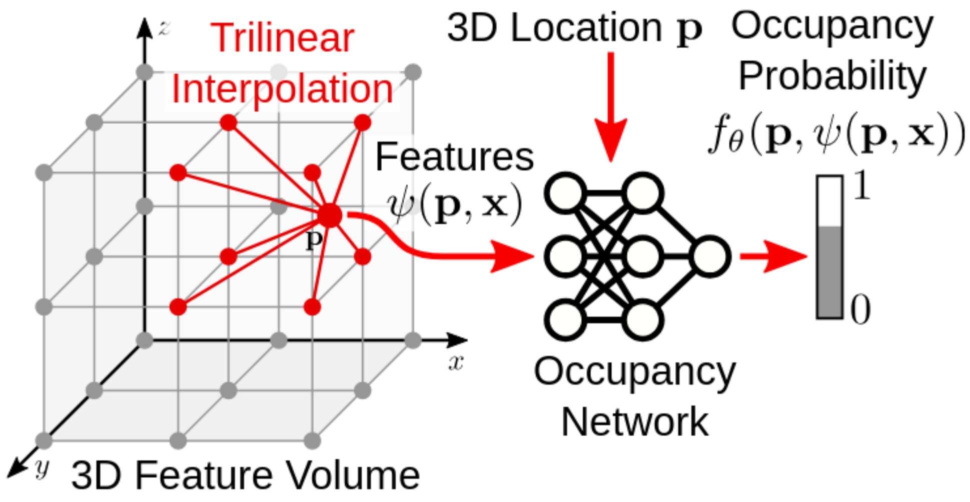
Mult-Scale Voxel Grid + Hash Encoding



Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, Müller et al. SIGGRAPH '22

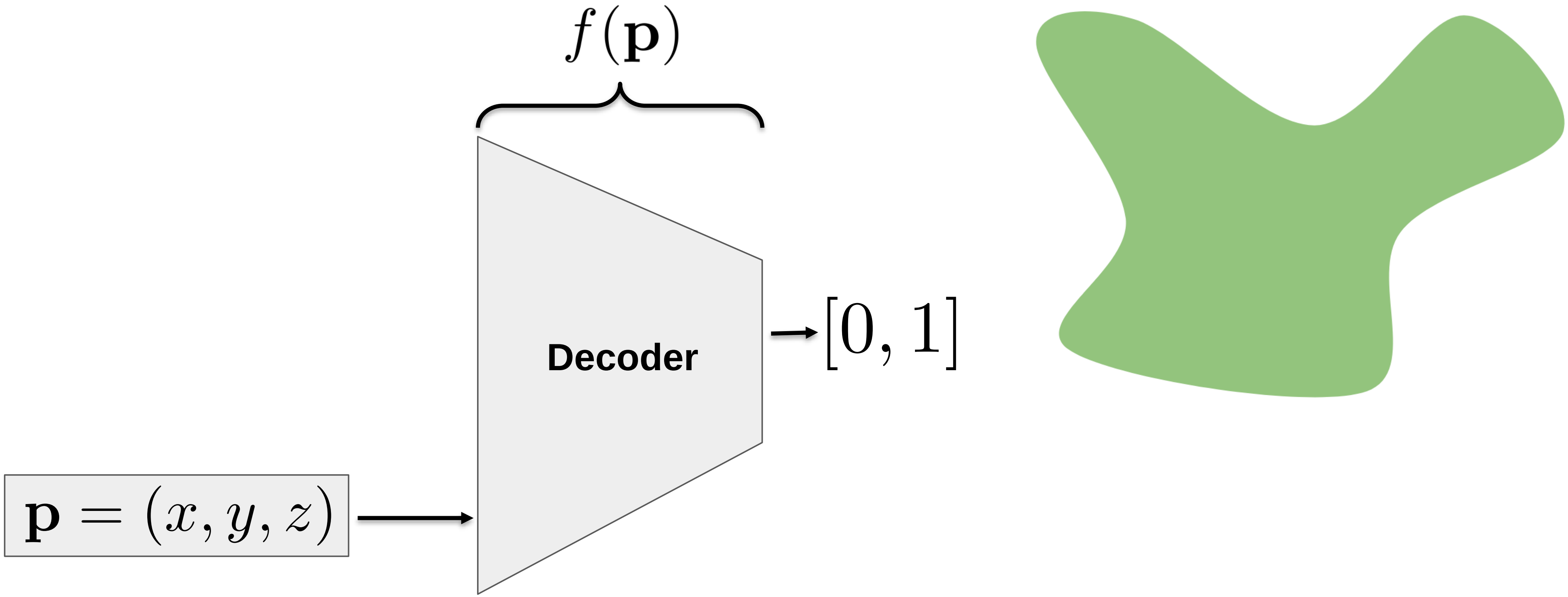
Hybrid Representations

- Trade-off between memory and speed
- Best of both worlds
- Neural network can locally "super-resolve" discrete representation

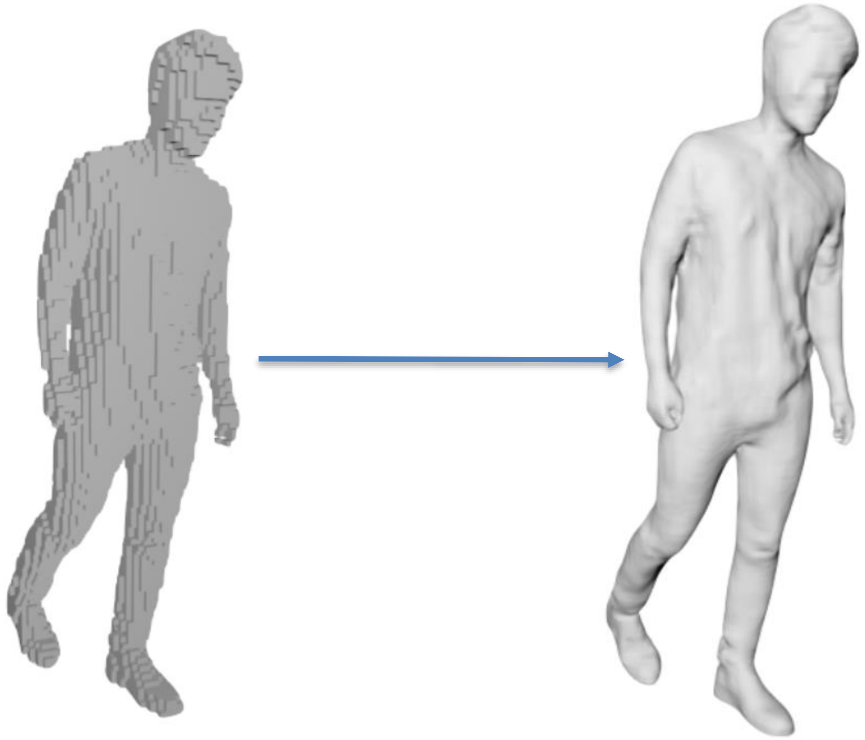


Neural Fields for 3D Reconstruction

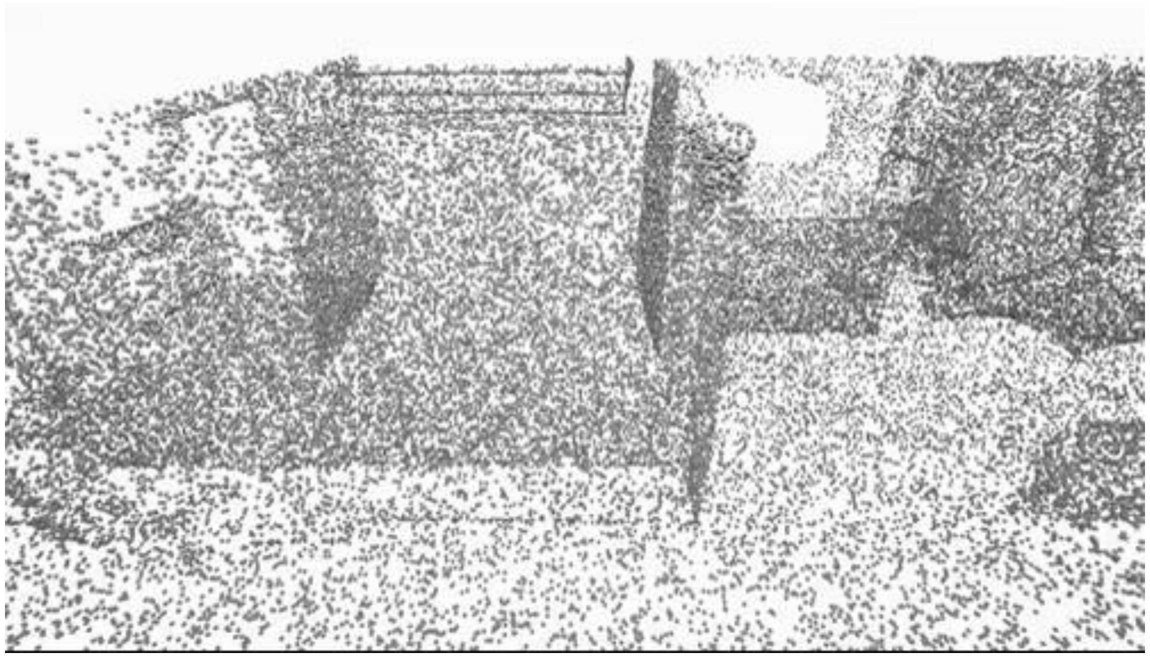
Neural Implicits for 3D Reconstruction



3D Reconstruction Tasks



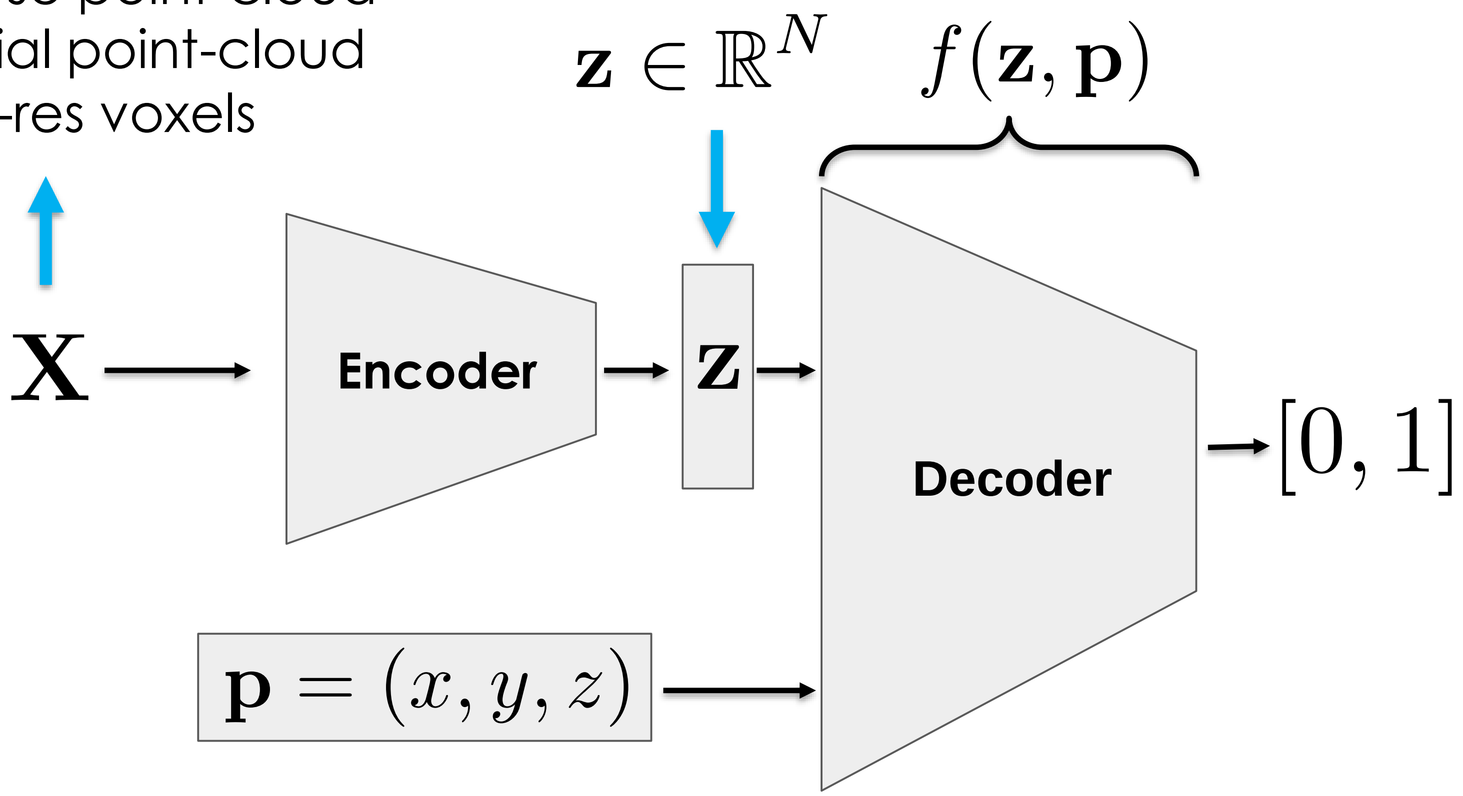
Voxel Super-Resolution



Point Cloud Completion

Previous Implicit Function Learning Architecture

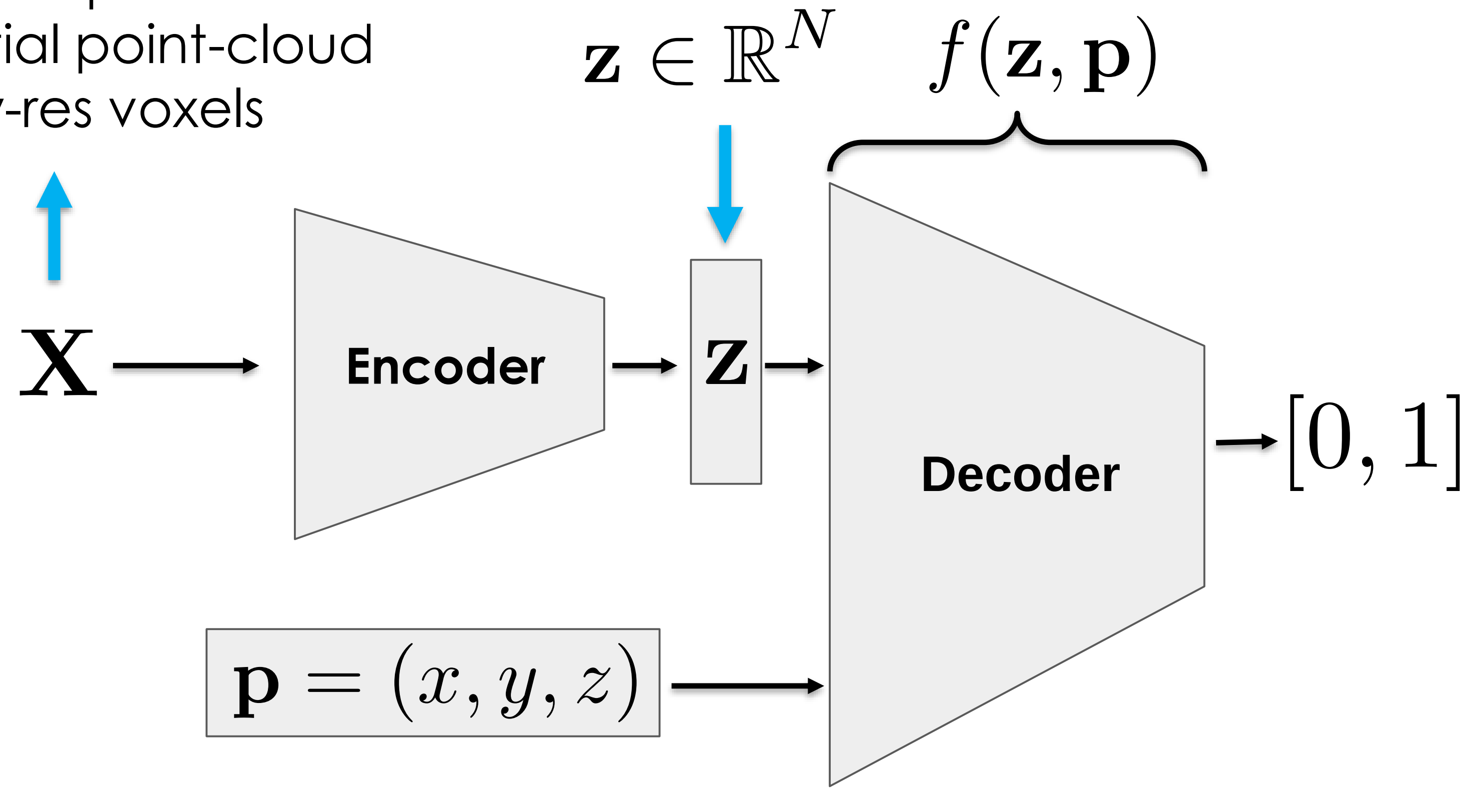
$\mathbf{X}$ is a shape observation:
- Sparse point-cloud
- Partial point-cloud
- Low-res voxels

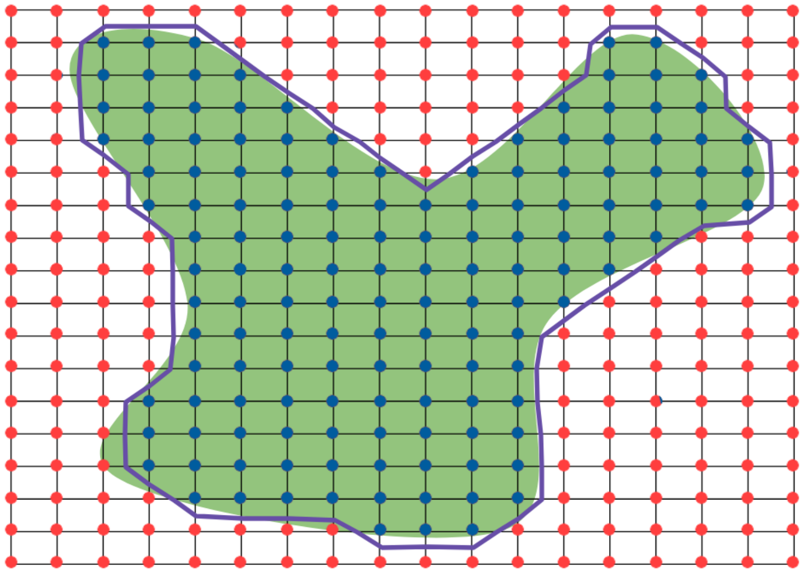


[Mescheder et al. CVPR'19
Chen et al. CVPR'19
Park et al. CVPR'19]

Previous Implicit Function Learning Architecture

$\mathbf{X}$ is a shape observation:
- Sparse point-cloud
- Partial point-cloud
- Low-res voxels





[Mescheder et al. CVPR'19
Chen et al. CVPR'19
Park et al. CVPR'19]

Neural Implicits for common objects

Work well for rigid objects: ✓ Continuous ✓ Multiple topologies

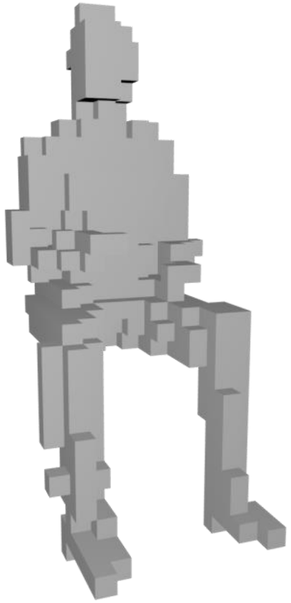
[Park et al. CVPR'19]

[Chen et al. CVPR'19]

[Mescheder et al. CVPR'19]

Problem with Previous Work

Reconstruct Articulations

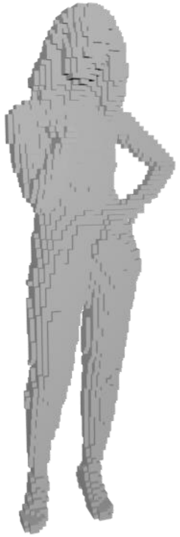


X

[Chen et al. CVPR'19]

[Mescheder et al. CVPR'19]

Retain Details



X

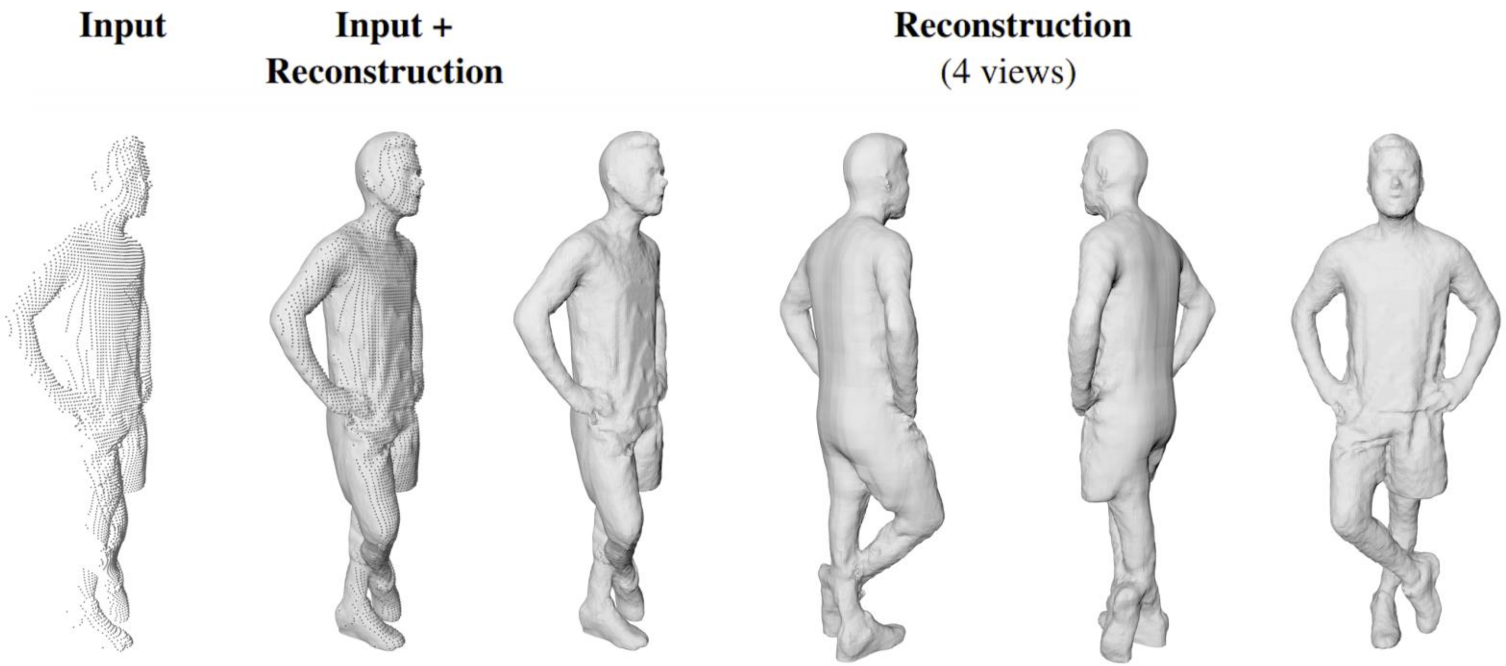
[Chen et al. CVPR'19]

[Mescheder et al. CVPR'19]

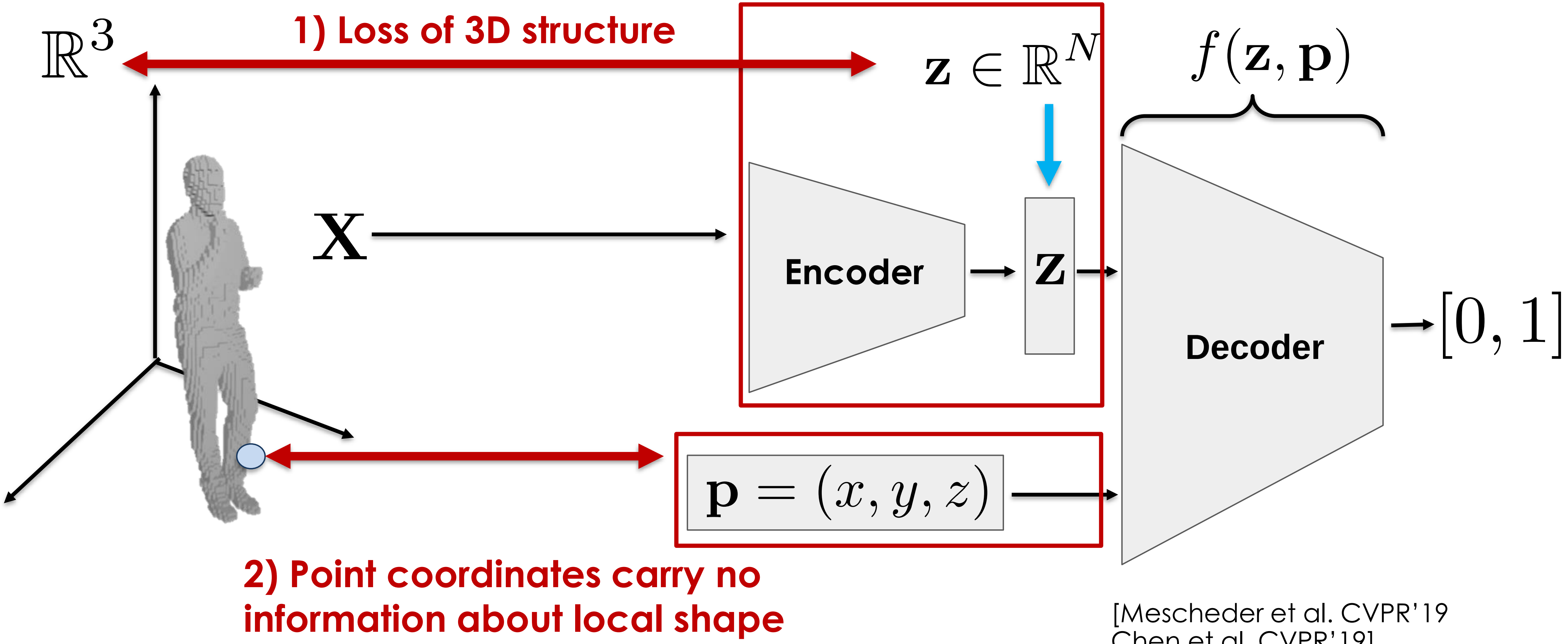
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion

Julian Chibane[1,2], Thiemo Alldieck[1,3], Gerard Pons-Moll[1]

CVPR 2020

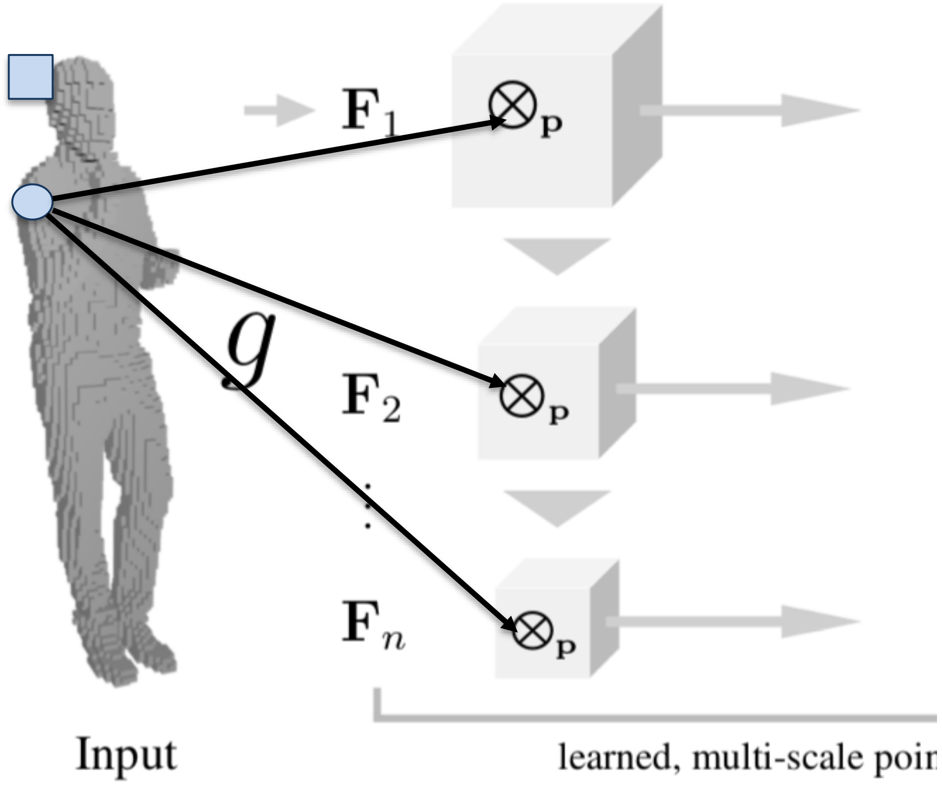


Problems with previous work



[Mescheder et al. CVPR'19
Chen et al. CVPR'19]

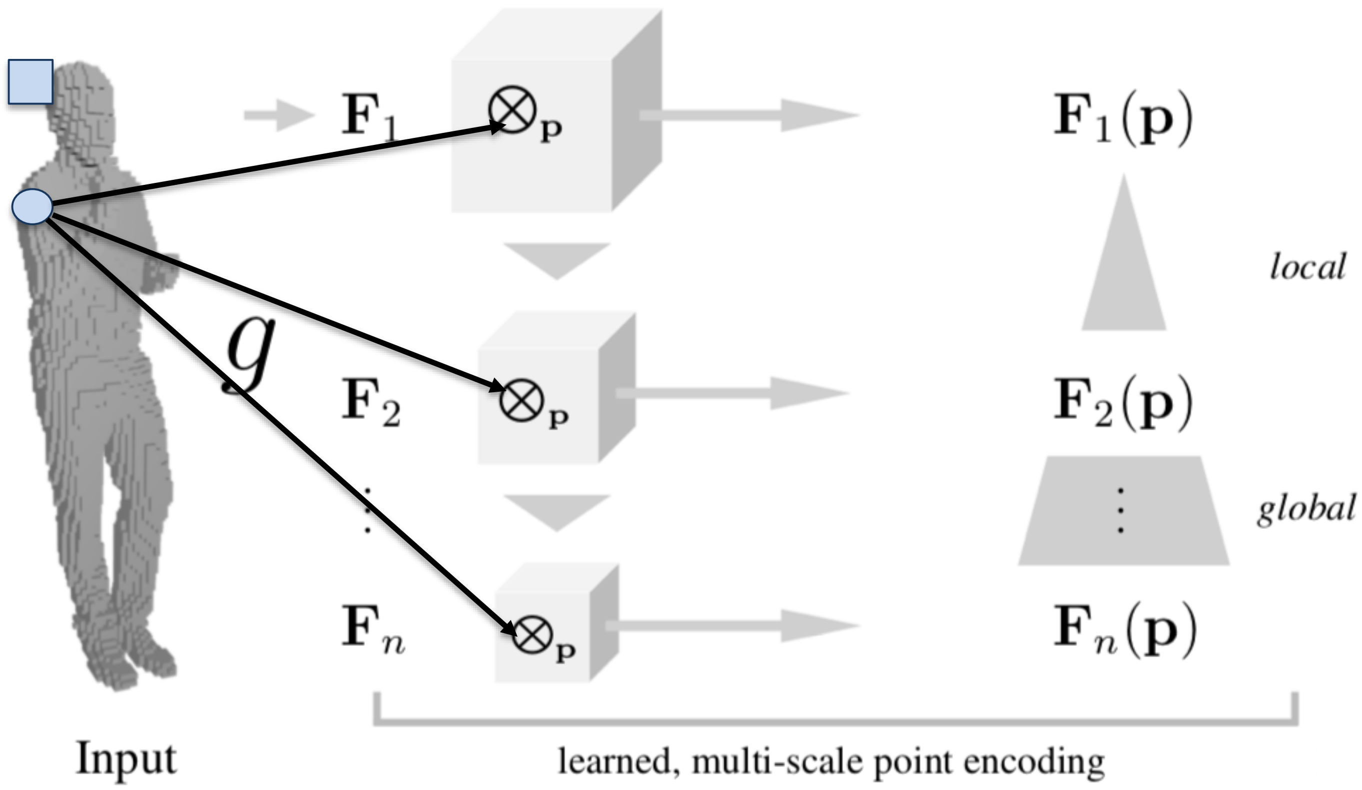
Implicit Feature Networks (IF-Nets)



3D Grid
$K \times K \times K$

Chibane et al. IF-Nets CVPR'20

Implicit Feature Networks (IF-Nets)



3D Grid
$K \times K \times K$

Chibane et al. IF-Nets CVPR'20

Representation of IF-Nets

Previous: $$f(\mathbf{z}, \mathbf{p}) \mapsto [0, 1]$$

$$\downarrow$$

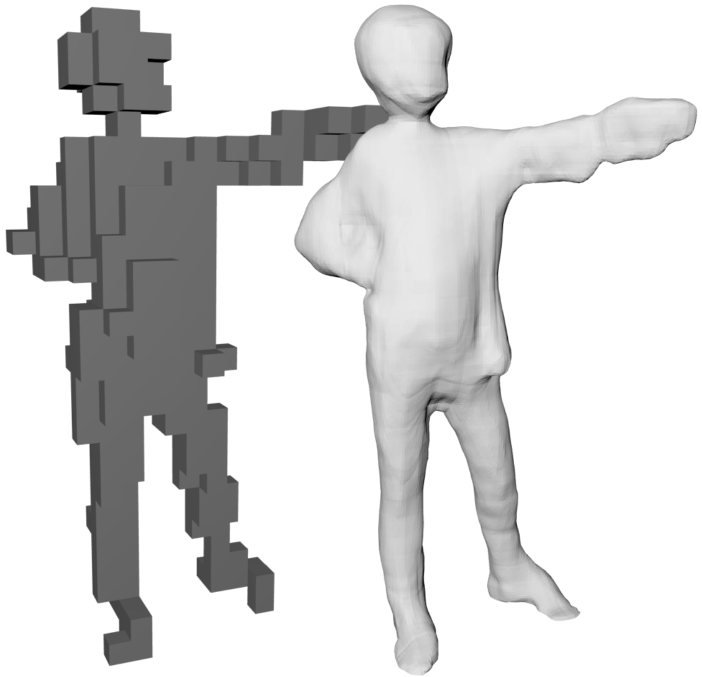
$$\mathbf{F}_1, \ldots, \mathbf{F}_n, \quad \mathbf{F} \in \mathcal{F}^{K \times K \times K}$$

Ours: $$f(\mathbf{F}_1(\mathbf{p}), \ldots, \mathbf{F}_n(\mathbf{p})) \mapsto [0, 1]$$

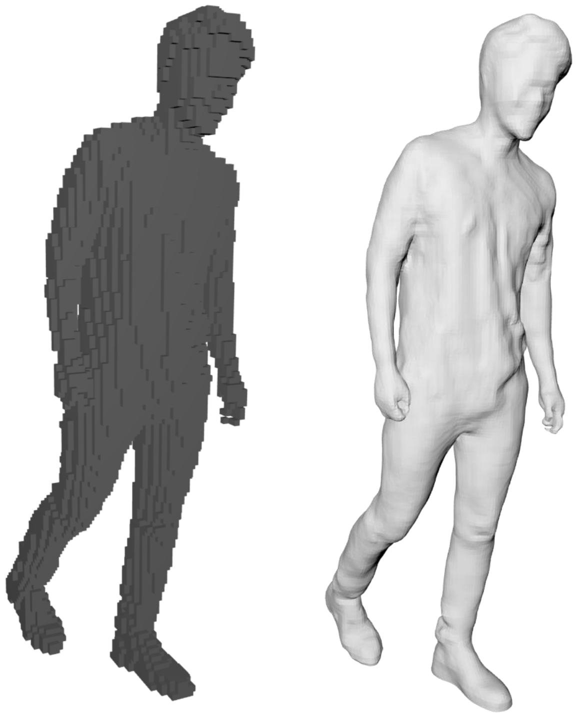
Chibane et al. IF-Nets CVPR'20

IF-Nets for 3D Shape Reconstruction and Completion

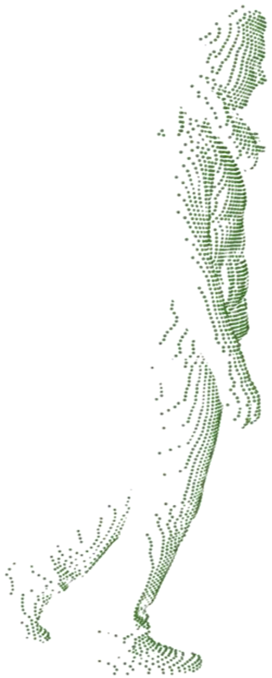
✓ Reconstruct Articulations



✓ Retain Details



✓ Complete Shape



IF-Nets for Texture completion

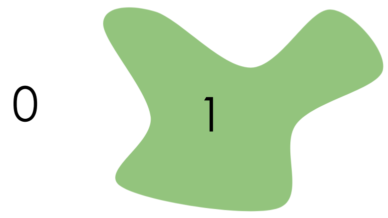
ECCV SHARP CHALLENGE

Input Prediction GT Input Prediction GT

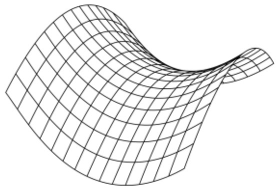
[Chibane and Pons-Moll, IF-Nets for texture. SHARP 2020
Chibane et al. IF-Nets CVPR'20]

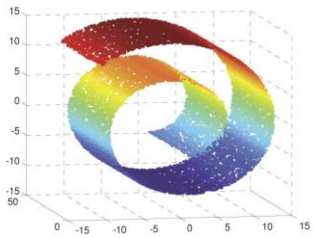
- Surfaces that do not divide the space in two regions can not be represented.

- We need a different **output representation.**

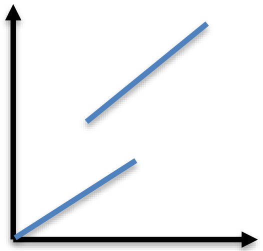


Only water-tight surfaces





Open surfaces and manifolds



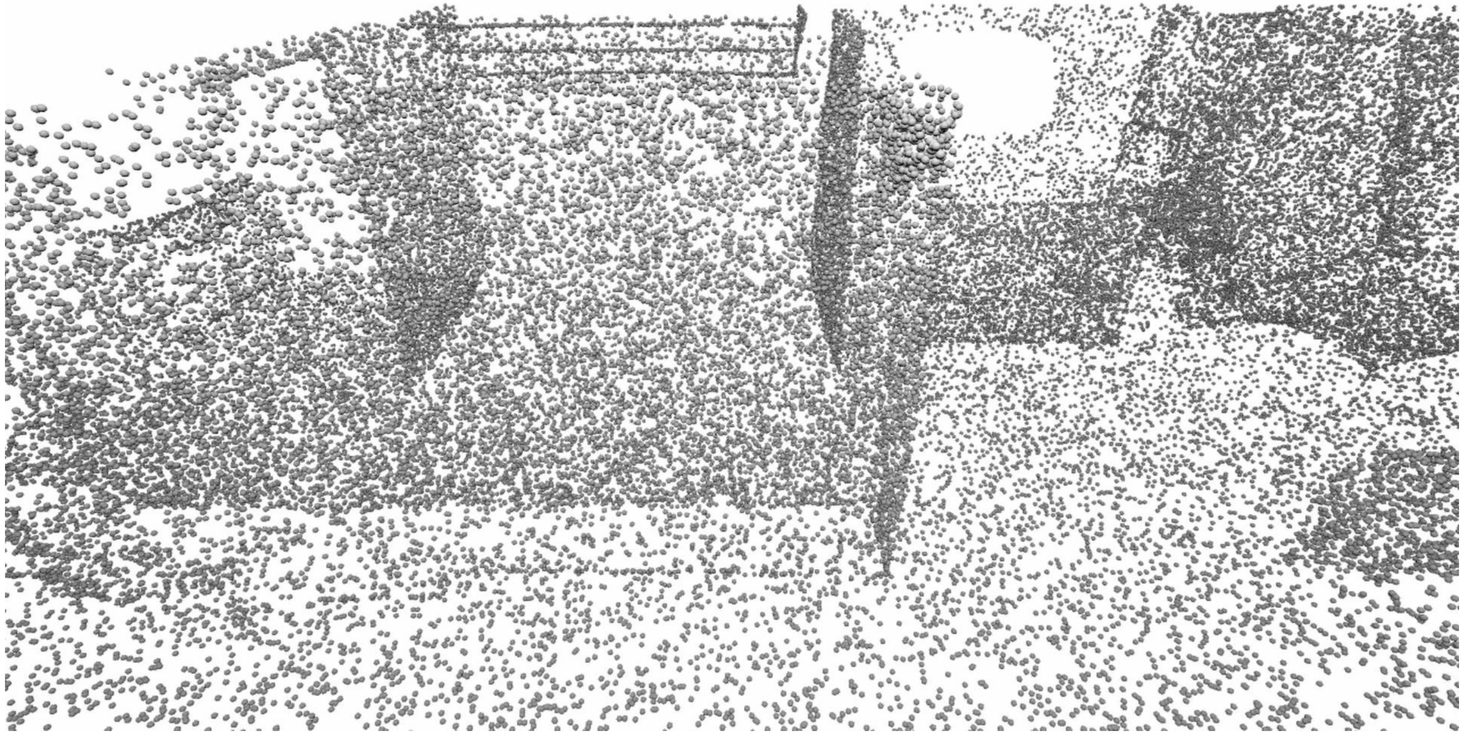
Functions

Complex shapes

Neural Unsigned Distance Fields for Implicit Function Learning

Julian Chibane, Aymen Mir, Gerard Pons-Moll

NeurIPS 2020



Chibane et al. **NDF**, NeurIPS 2020

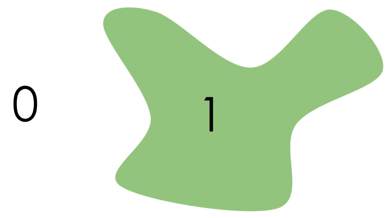
Our Solution

Change the output representation

$$f(\mathbf{F}_1(\mathbf{p}), \ldots, \mathbf{F}_n(\mathbf{p})) \mapsto [0, 1] \qquad f(\mathbf{F}_1(\mathbf{p}), \ldots, \mathbf{F}_n(\mathbf{p})) \mapsto \mathbb{R}^+$$

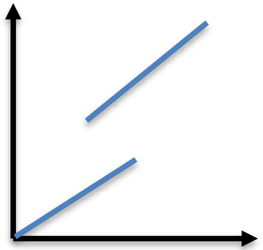
Unsigned distance:

$$\boxed{f(\mathbf{p}) = \min_{\mathbf{q} \in \mathcal{S}} ||\mathbf{p} - \mathbf{q}||}$$



Only water-tight surfaces

Open surfaces and manifolds



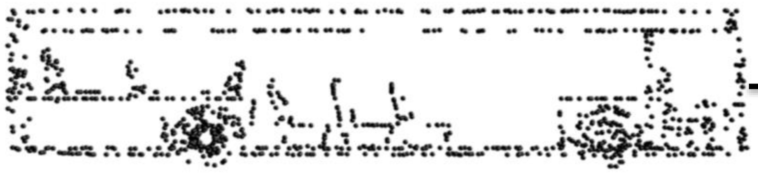
Functions

Complex shapes

Chibane et al. **NDF**, NeurIPS 2020

Neural Distance Fields

a)



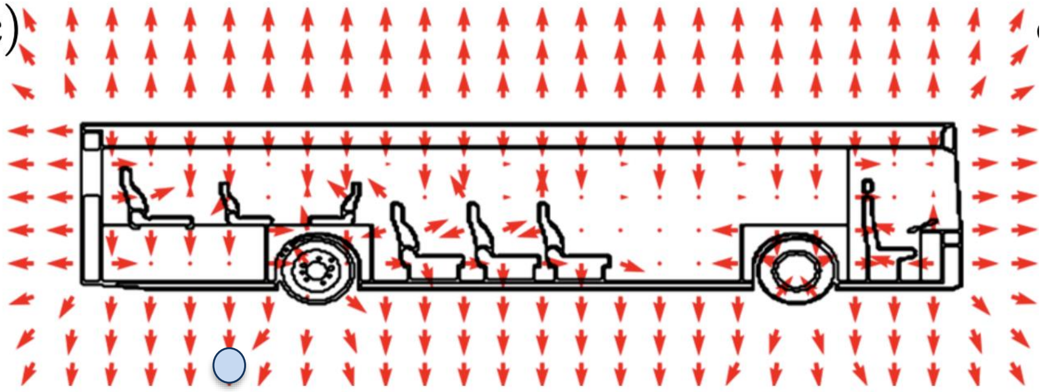
$\mathbf{X}$

$$f(\mathbf{p}) = \min_{\mathbf{q} \in \mathcal{S}} \|\mathbf{p} - \mathbf{q}\|$$

b)

$$\mathcal{S} = \{\mathbf{p} \in \mathbb{R}^d \mid f(\mathbf{p}) = 0\}$$

c)



$$\mathbf{q} = \mathbf{p} - f(\mathbf{p}) \nabla_{\mathbf{p}} f(\mathbf{p})$$

Chibane et al. **NDF**, NeurIPS 2020

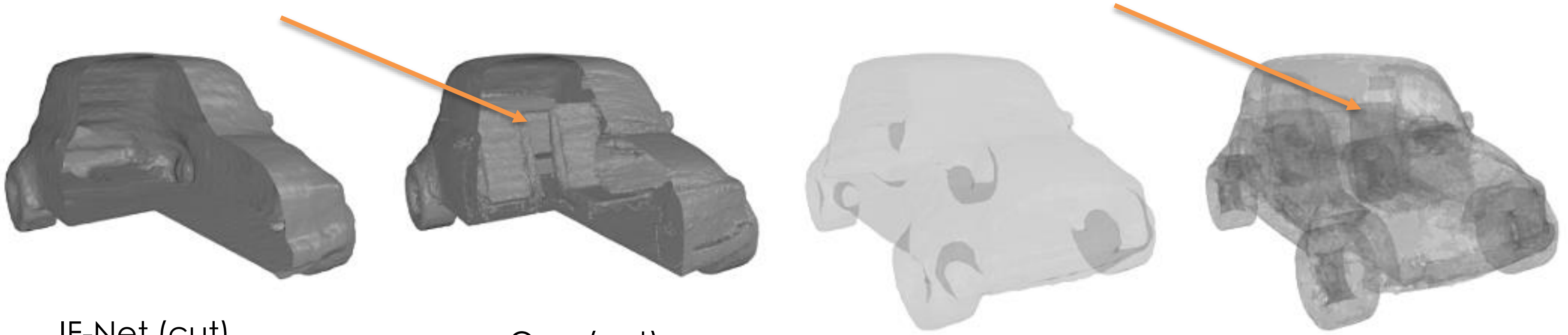
Neural processing of arbitrary surfaces

Next, we illustrate the capabilities of NDF to neurally process arbitrary surfaces, not representable by prior learned implicit work:

- **Mathematical Functions and Manifolds** – We train a single NDF on a dataset consisting of 1000 functions per type: linear function, parabola, sinusoids and spirals.
- **Garments** – Open Surfaces, without thickness. Training on ~300 garments of five types from [Bhatnagar et al. ICCV'19].
- **Scenes** – Open surfaces with holes and no thickness. Training on 34 real world scenes captured by RGBD Sensors from [Xia et al. CVPR'18].

Chibane et al. **NDF**, NeurIPS 2020

NDF results



IF-Net (cut) | Ours (cut) | IF-Net (Transparent) | Ours (Transparent)

1) Comparison with IF-Nets

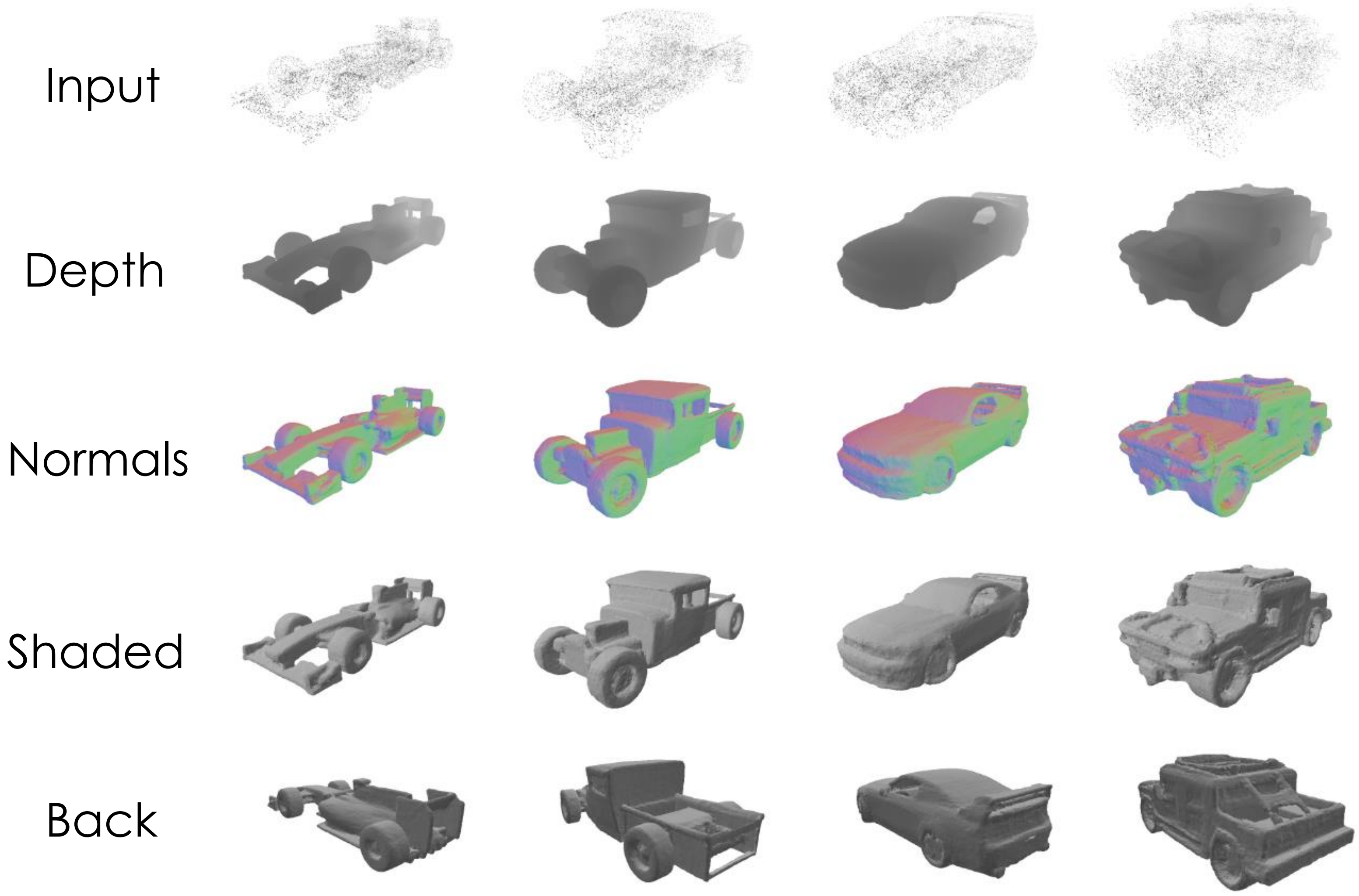
GT Mesh | Input | Output 1Mio. PC | Direct Rendering (Front View) | Direct Rendering (Side View)

Open surface

2) Garment reconstruction

Chibane et al. **NDF**, NeurIPS 2020

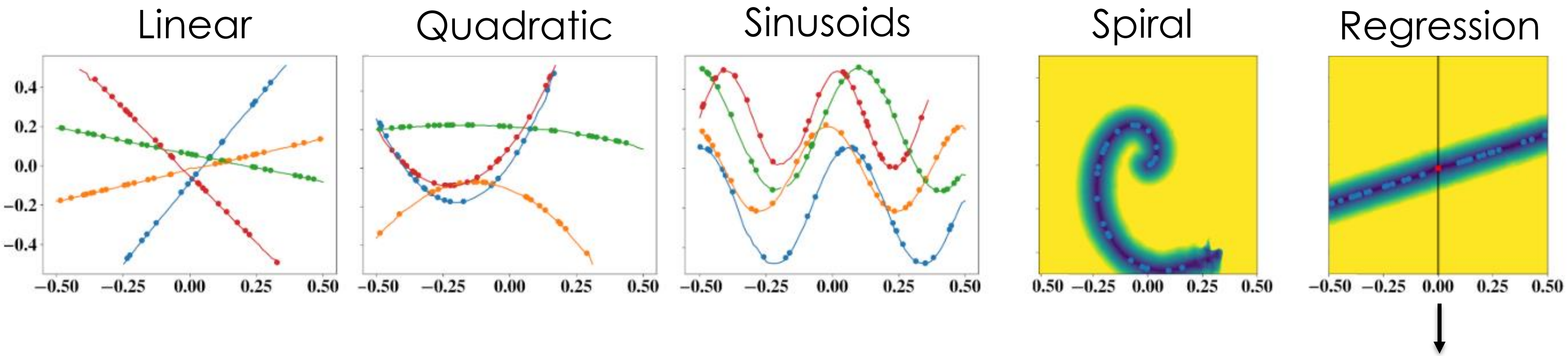
Direct Rendering of NDF



Sphere
marching

Chibane et al. **NDF**, NeurIPS 2020

Representation and Regression of Functions



$$y = h(x_1, \ldots, x_n) \longrightarrow f(x_1, \ldots, x_n, y) = 0$$

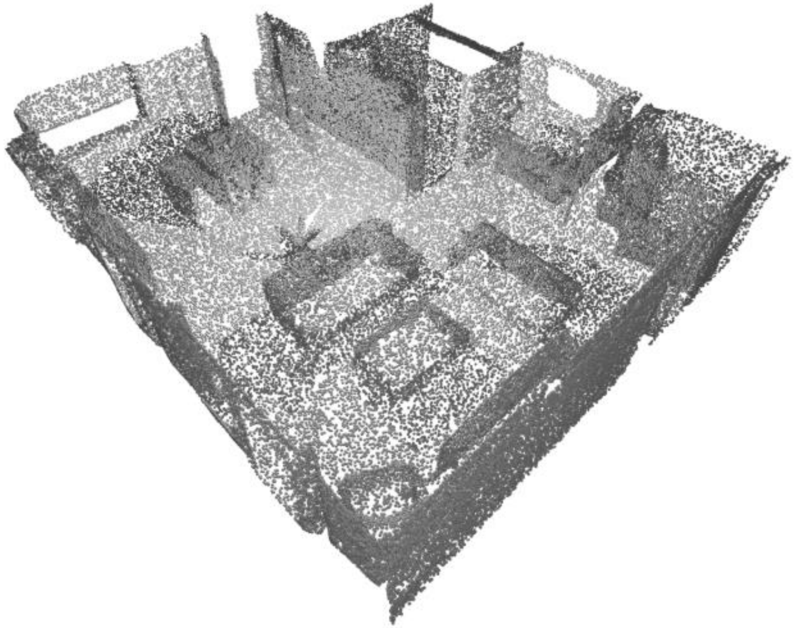
$$\mathbf{p}(\lambda) = (x_1, \ldots, x_n, 0) + \lambda(0, \ldots, 0, 1) \longrightarrow f(\mathbf{p}(\lambda)) = 0$$

Classical regression using NDFs and an adapted sphere tracing (ray tracing method)

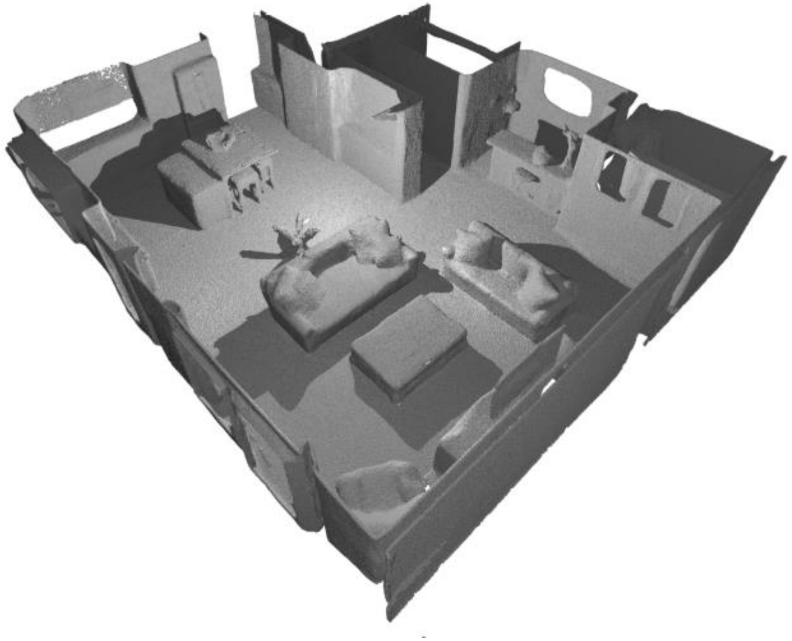
Chibane et al. **NDF**, NeurIPS 2020

Representation and Completion of Scenes

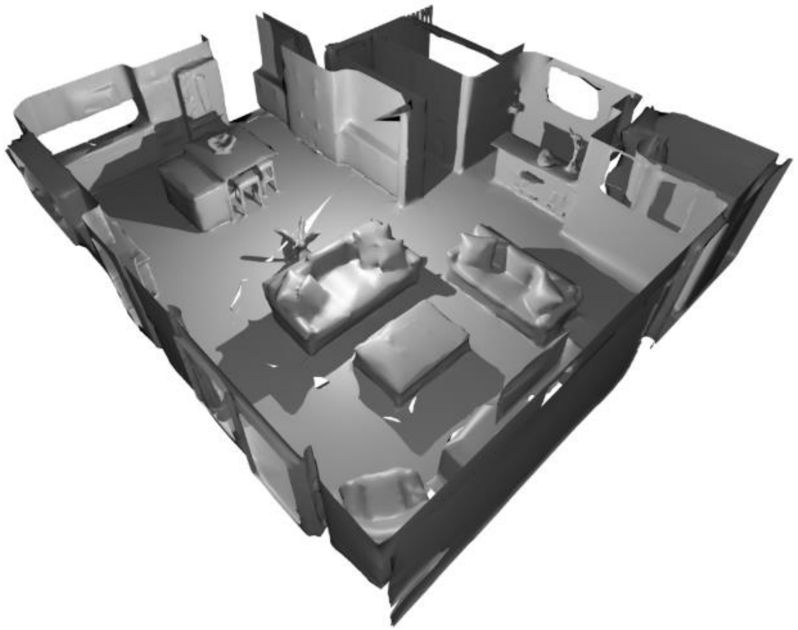
Input



Output



Ground Truth



Chibane et al. **NDF**, NeurIPS 2020

Meshes vs Implicits

1) Parametric Meshes

[Alldieck et al. CVPR'18
Bhatnagar et al. ICCV'19, ECCV'20
Tiwari et al. ECCV'20]

2) Implicit Functions

General objects and humans

[Chibane et al. CVPR'20
Chibane et al. NeurIPS'20]

	Control /Meaning	Topology	Details
1)	✓	✗	✗
2)	✗	✓	✓

2) Compatible with learning

Slides credit and resources

Thanks to
Julian Chibane, Enric Corona and Qianli Ma
for providing materials.

Parts of the slides were inspired by the course Inverse Graphics by Vincent Sitzmann

TUM AI Lecture Series - Neural Implicit Representations for 3D Vision
(talk by Prof. Pons-Moll)