Towards Practical Applications of NeRF

Songyou Peng

ETH zürich

MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

Adobe Research
Sep 06, 2022

Who Am I?

- PhD Student
  - Marc Pollefeys
  - Andreas Geiger

ETH zürich

MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

- Internships during PhD
  - 2021: Michael Zollhoefer
  - Now: Tom Funkhouser

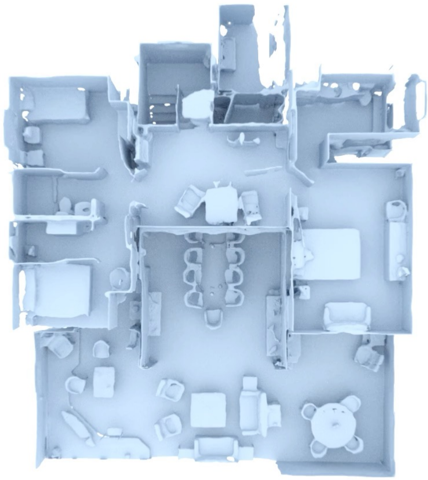
Meta

Google Research

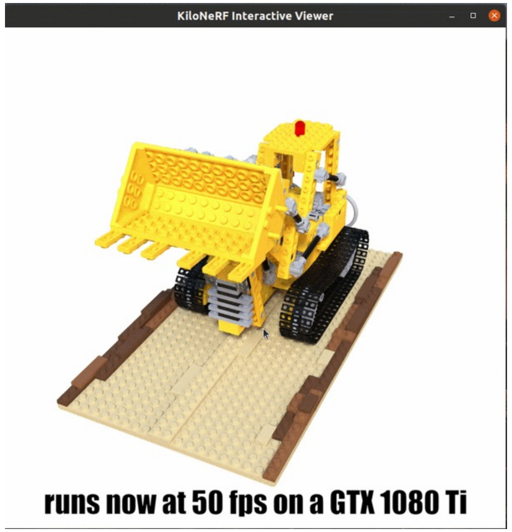
- Graduate next summer

pengsongyou.github.io

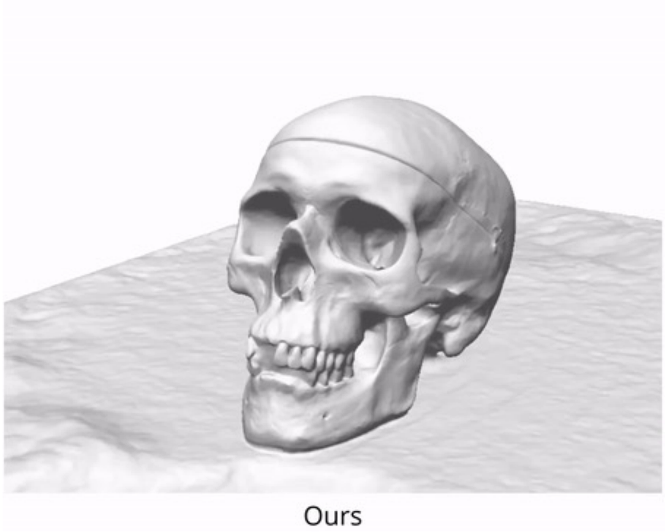
My PhD Topics: Neural Scene Representations
for 3D reconstruction, novel view synthesis, and SLAM



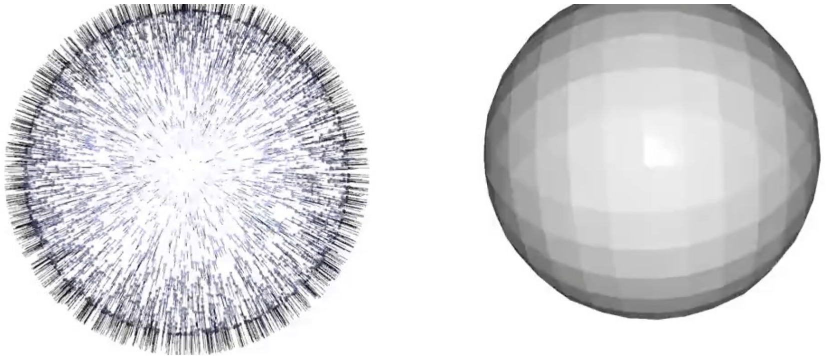
Convolutional Occupancy Networks
ECCV 2020 (Spotlight)



KiloNeRF
ICCV 2021

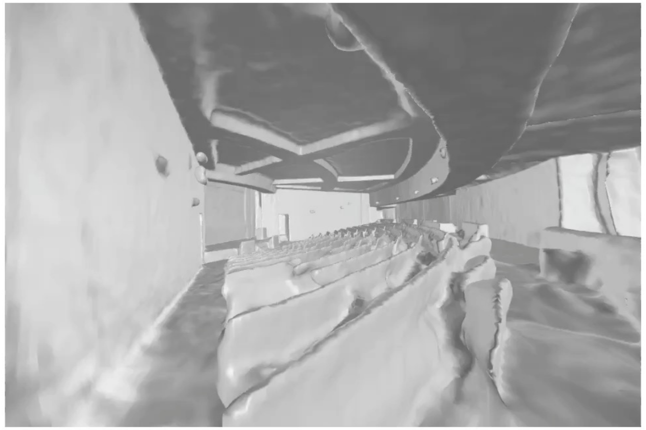


UNISURF
ICCV 2021 (Oral)



Shape As Points
NeurIPS 2021 (Oral)

NICE-SLAM
CVPR 2022



Ours
MonoSDF
arXiv 2022

NeRF is awesome!

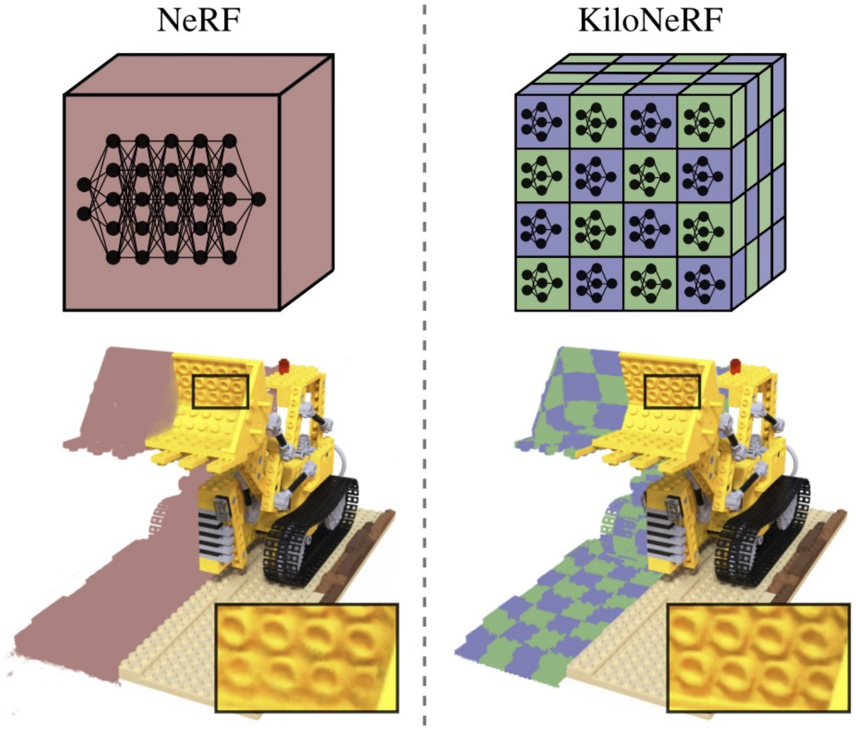
Some problems still exist…

😢 Slow rendering speed	😊 KiloNeRF
😢 Poor underlying geometry	😊 UNISURF + MonoSDF
😢 Camera poses needed	😊 NICE-SLAM

[1] Mildenhall*, Srinivasan*, Tancik* et al: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV 2020

KiloNeRF

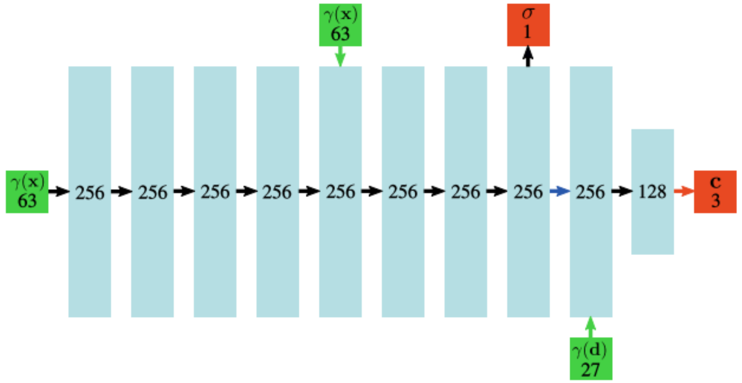
Speeding up NeRF with Thousands of Tiny MLPs



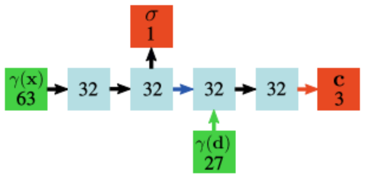
Key Idea

- Partition a scene into a 16^3 uniform grid
- Each grid cell is represented by a tiny MLP

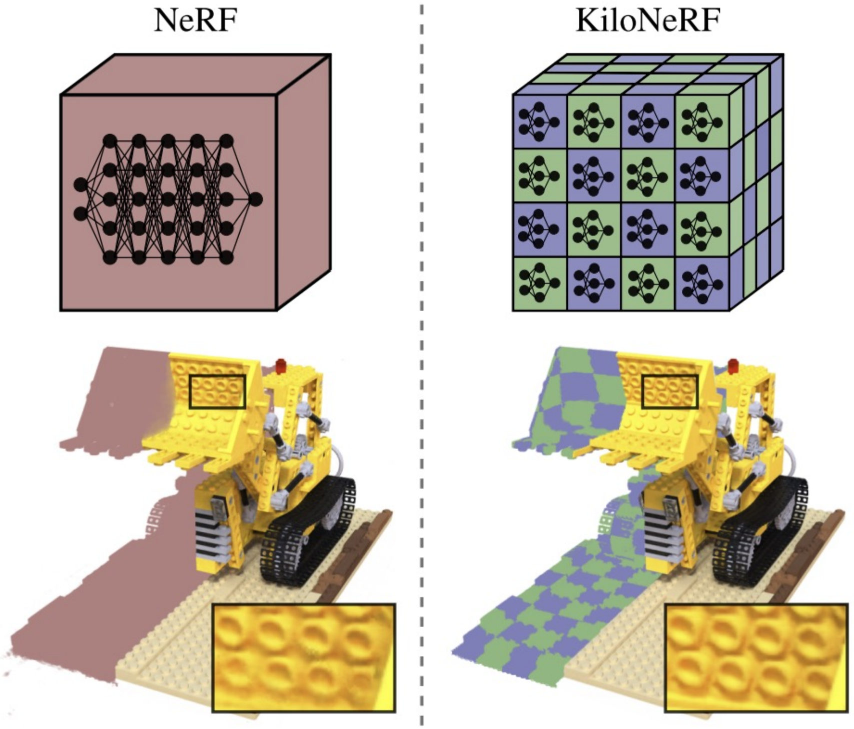
NeRF: ~1056 kFLOPs



KiloNeRF: ~12 kFLOPs



87x reduction in FLOPs!



* FLOP: floating points operations

KiloNeRF

Training:

1. Distill a trained NeRF model into our KiloNeRF model
   - Randomly sampled points, their predicted alpha & color values should match!
2. Finetune the thousand MLPs on training images

(a) Without Distillation (b) With Distillation

KiloNeRF

Training:

1. Distill a trained NeRF model into our KiloNeRF model
   - Randomly sampled points, their predicted alpha & color values should match!
2. Finetune the thousand MLPs on training images

Inference:

1. Empty Space Skipping (ESS) with a pre-computed 256^3 occupancy grid
2. Early Ray Termination (ERT): when transmittance < ε, stop!
3. Evaluate tiny MLPs in parallel

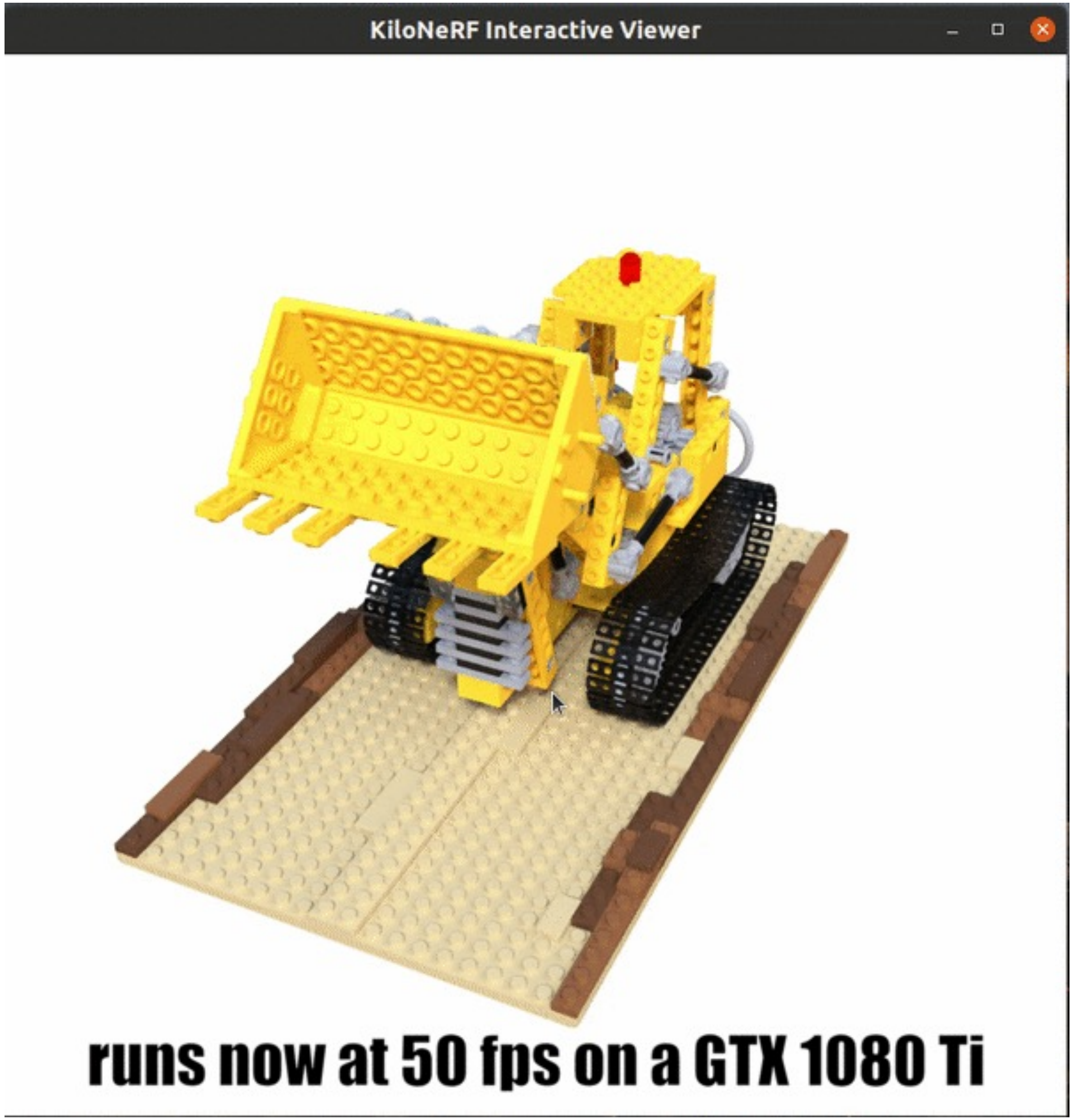
Method	Render time ↓	Speedup ↑
NeRF	56185 ms	–
NeRF + ESS + ERT	788 ms	71
KiloNeRF	**22** ms	**2554**

* Tested with NVIDIA GTX 1080 Ti

Results

56 s

0.02 s (50 fps)

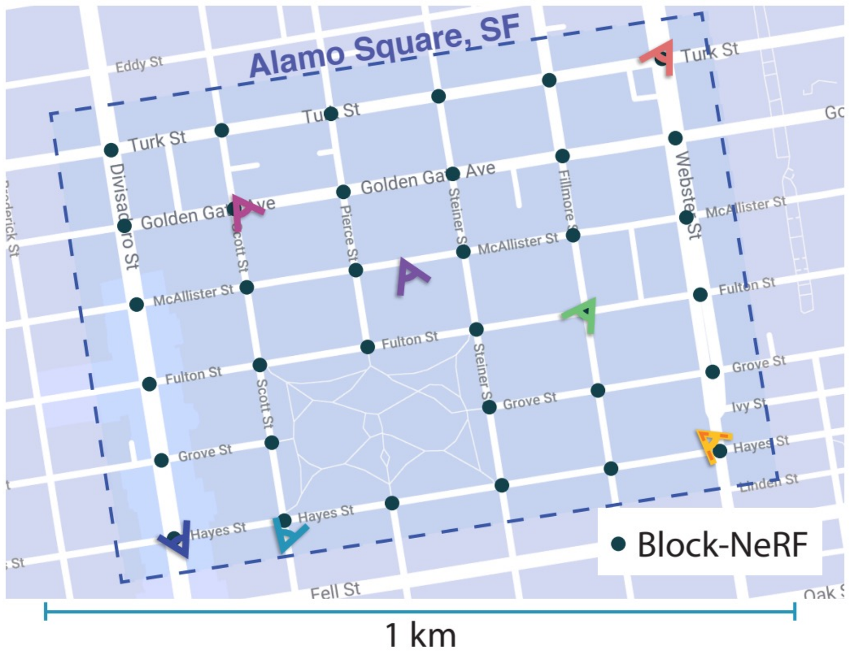


https://github.com/creiser/kilonerf

Comparison to Concurrent Works

Type	Neural	Tabulation-based		
Method	**KiloNeRF**	PlenOctree	SNeRG	FastNeRF
GPU Memory	**< 100 MB**	1930 MB	3442 MB	7830 MB

⇒ KiloNeRF has a larger potential for large-scale NVS!



BlockNeRF applied our idea for city-level NVS ☺

Take-home Message

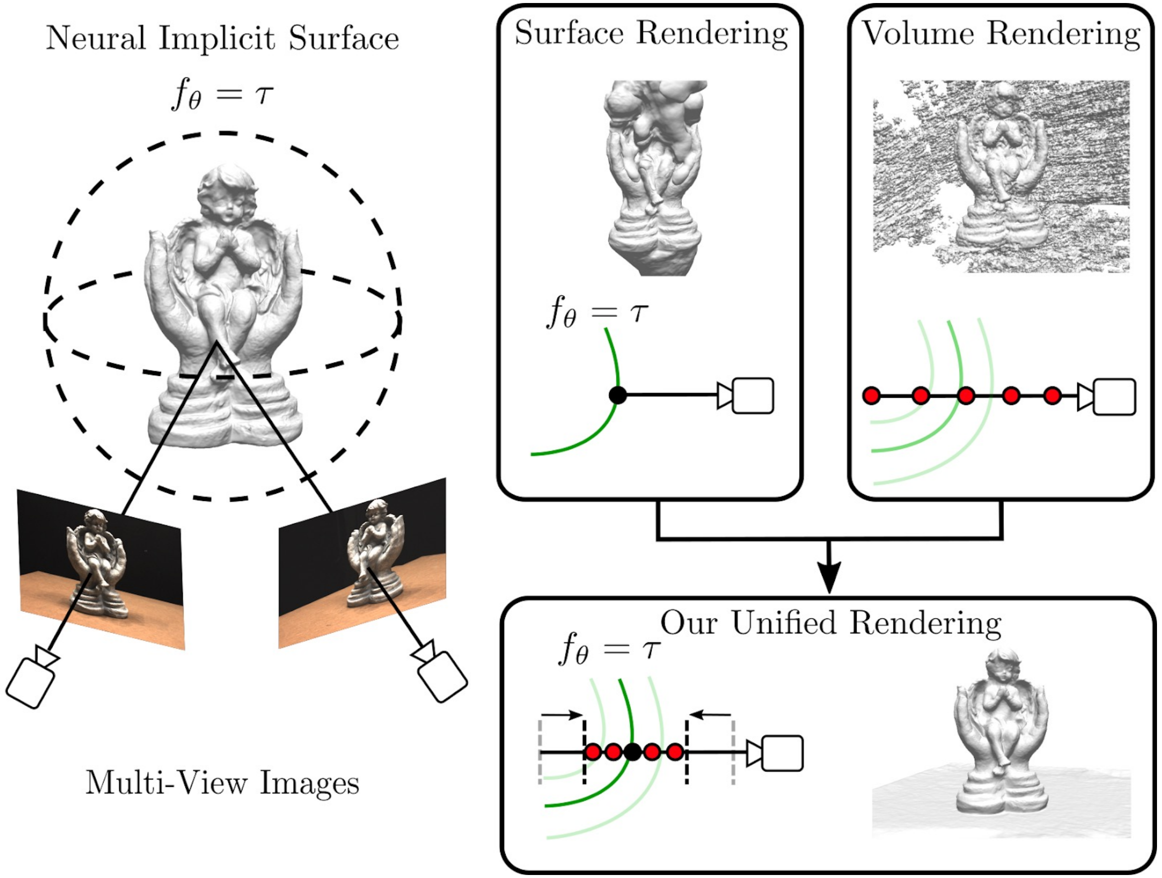
- Speed up NeRF significantly (~2000x) without loss of quality
- A memory more friendly representation!

Limitations

- Only work on bounded scenes
- Expensive training time

UNISURF

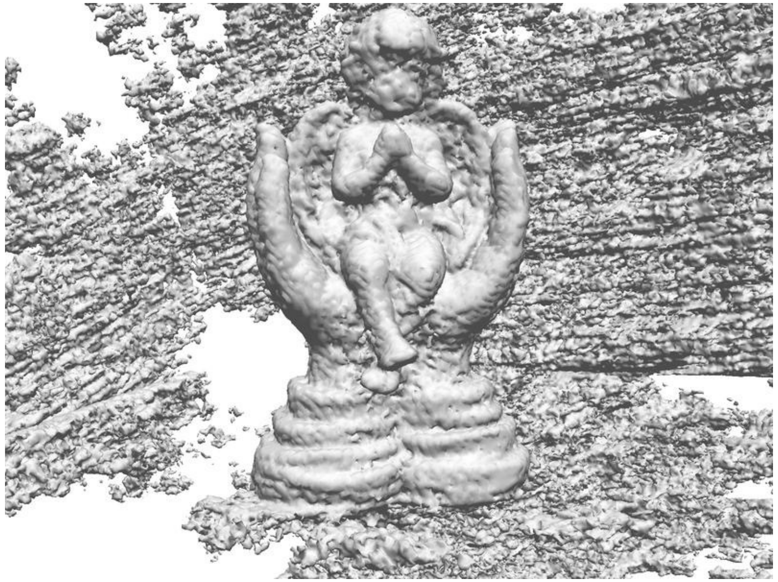
Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction



Motivation

The underlying geometry of NeRF (volume rendering) is poor

NeRF Rendering



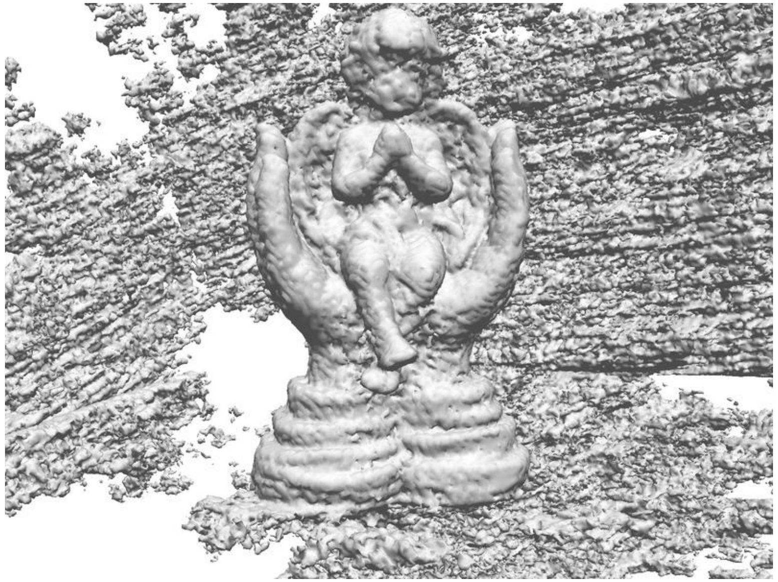
NeRF Geometry

[1] Yariv et al.: Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. NeurIPS 2020

Motivation

Surface rendering methods produce great geometry, but require object masks

NeRF Rendering

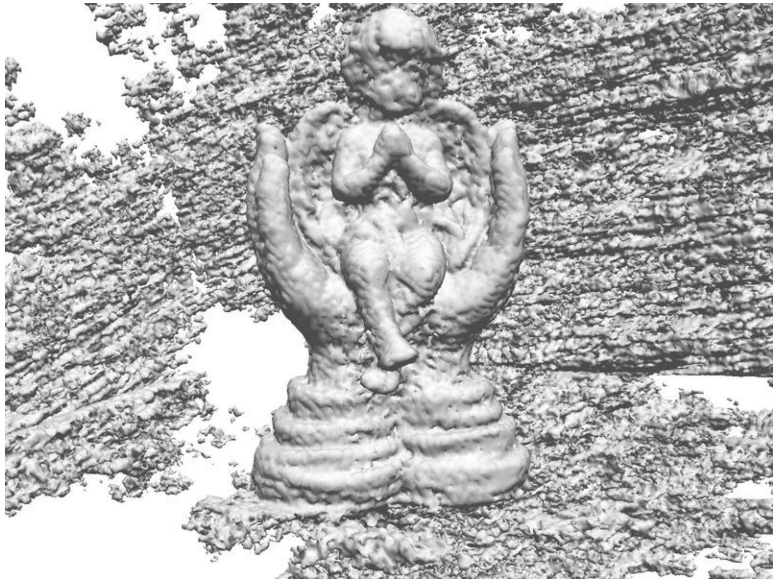


NeRF Geometry

IDR [1] Geometry

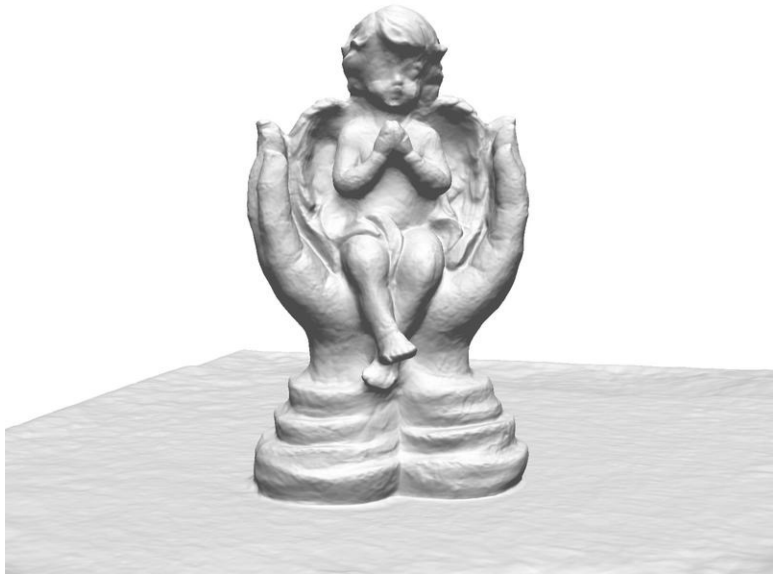
[1] Yariv et al.: Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. NeurIPS 2020

Can we obtain accurate geometry without masks?



NeRF

IDR

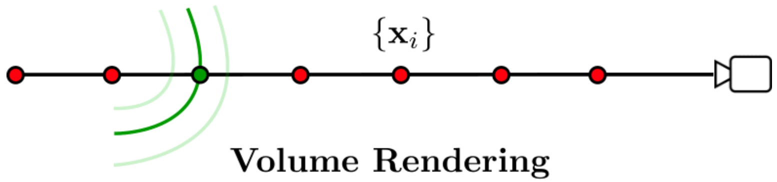


UNISURF

UNISURF

Unify radiance fields and implicit surface model ☺

UNISURF



Early Stage: Volume rendering, but reformulate density to occupancy

NeRF rendering: $$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i(\mathbf{x}_i) \prod_{j<i} (1 - \alpha_j(\mathbf{x}_j))\, c(\mathbf{x}_i, \mathbf{d}) \qquad \alpha_i(\mathbf{x}) = 1 - \exp\left(-\sigma(\mathbf{x})\, \delta_i\right)$$

Assuming a solid object, the alpha is the continuous occupancy field

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \boxed{o(\mathbf{x}_i) \prod_{j<i} (1 - o(\mathbf{x}_j))}\, c(\mathbf{x}_i, \mathbf{d})$$

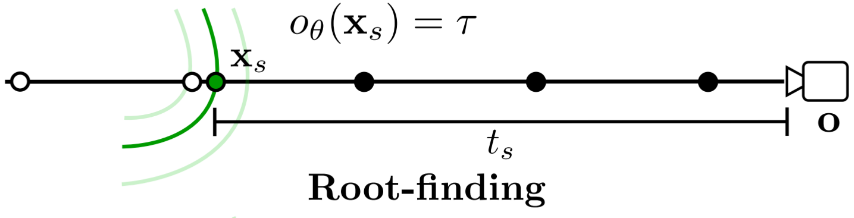
1 for the first occupied sample
0 for all other samples

Points near to the surface have larger influence to the predicted color

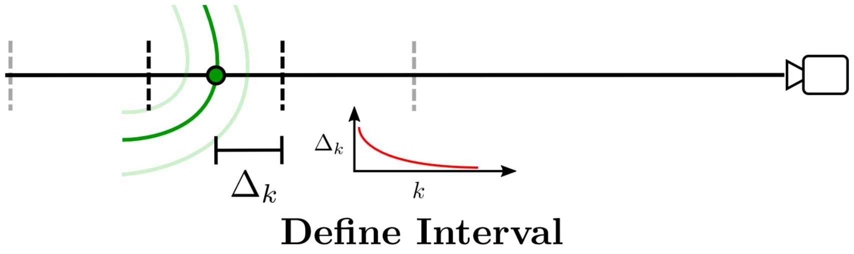
UNISURF

Later Stage: Find surface points, decrease the range of volume rendering

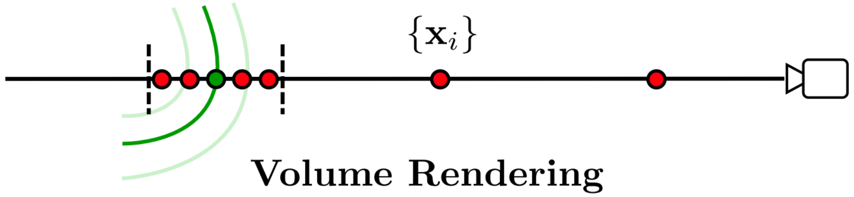
a) Find the surface point



b) Define the new interval



c) Volume rendering



Loss Functions

a) Image reconstruction loss

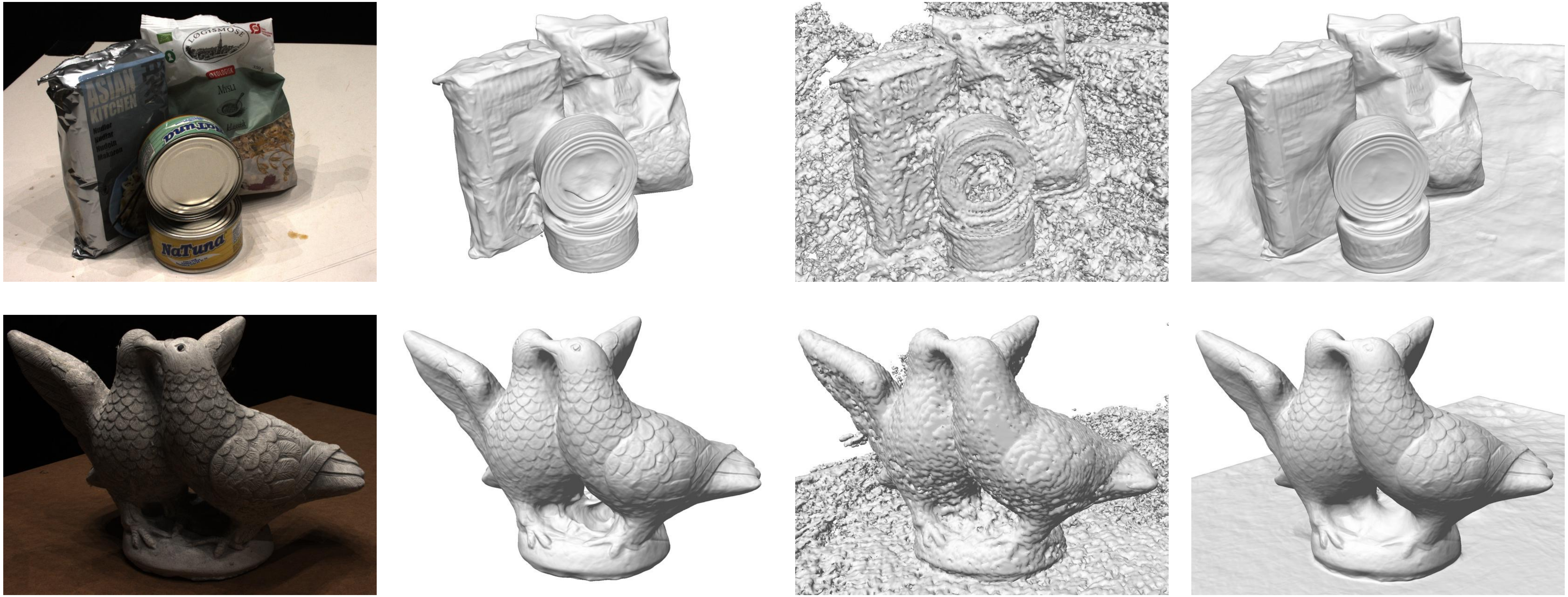
$$\mathcal{L}_{rec} = \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{C}_v(\mathbf{r}) - C(\mathbf{r})\|_1$$

b) Surface smoothness regularization

$$\mathcal{L}_{reg} = \sum_{\mathbf{x}_s \in \mathcal{S}} \|\mathbf{n}(\mathbf{x}_s) - \mathbf{n}(\mathbf{x}_s + \boldsymbol{\epsilon})\|_2$$

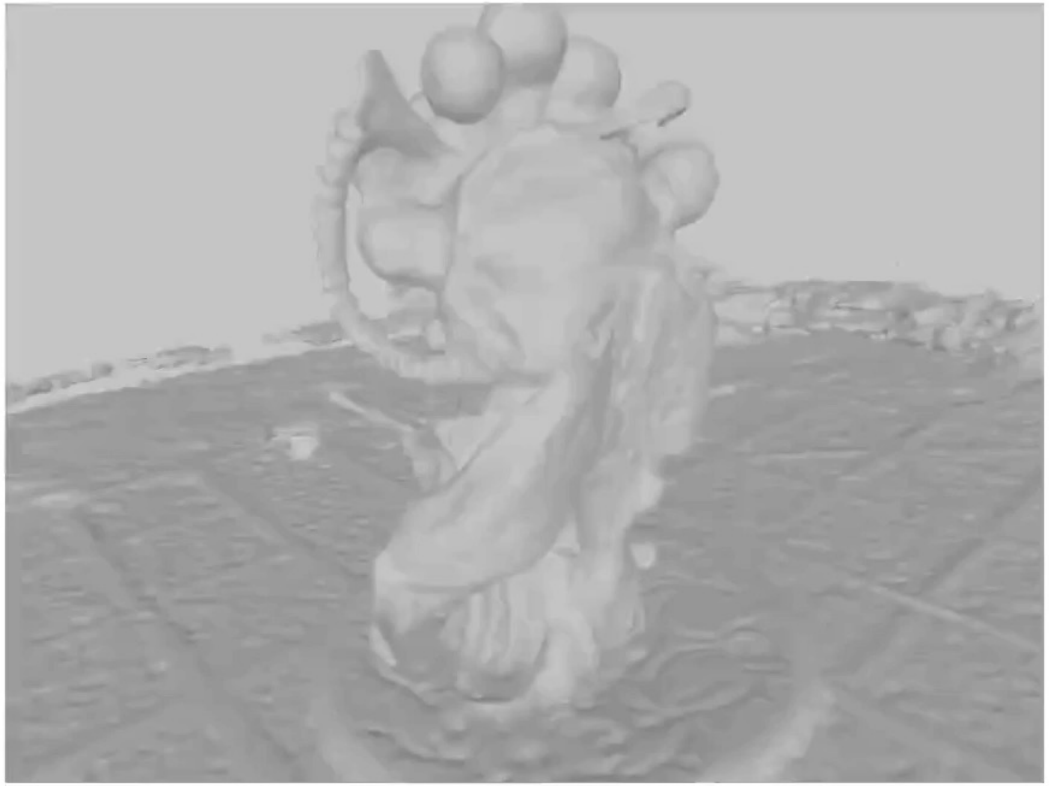
Results

DTU



GT View | IDR | NeRF | **UNISURF**

BlendedMVS



Take-home Message

- Volume rendering and implicit surfaces can be unified!
- Accurate reconstruction without the need of masks
- Many cocurrent & follow-up works: NeuS, VolSDF, NeuralWarp, GeoNeuS…

Limitations

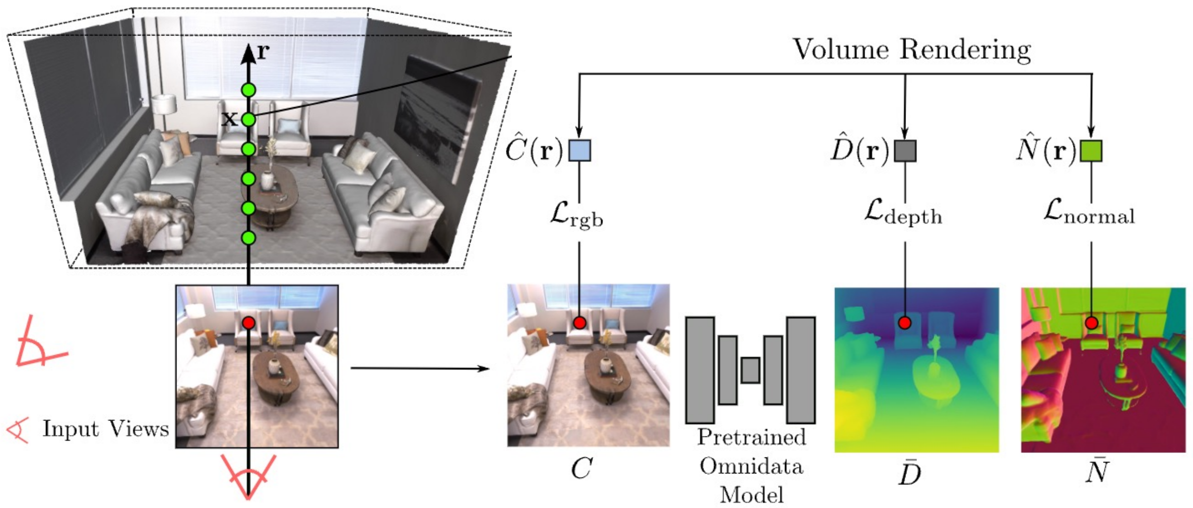
- Hard to reconstruct texture-less regions
- Still limited to small object-centric scenes
- Won’t work given only sparse views

MonoSDF

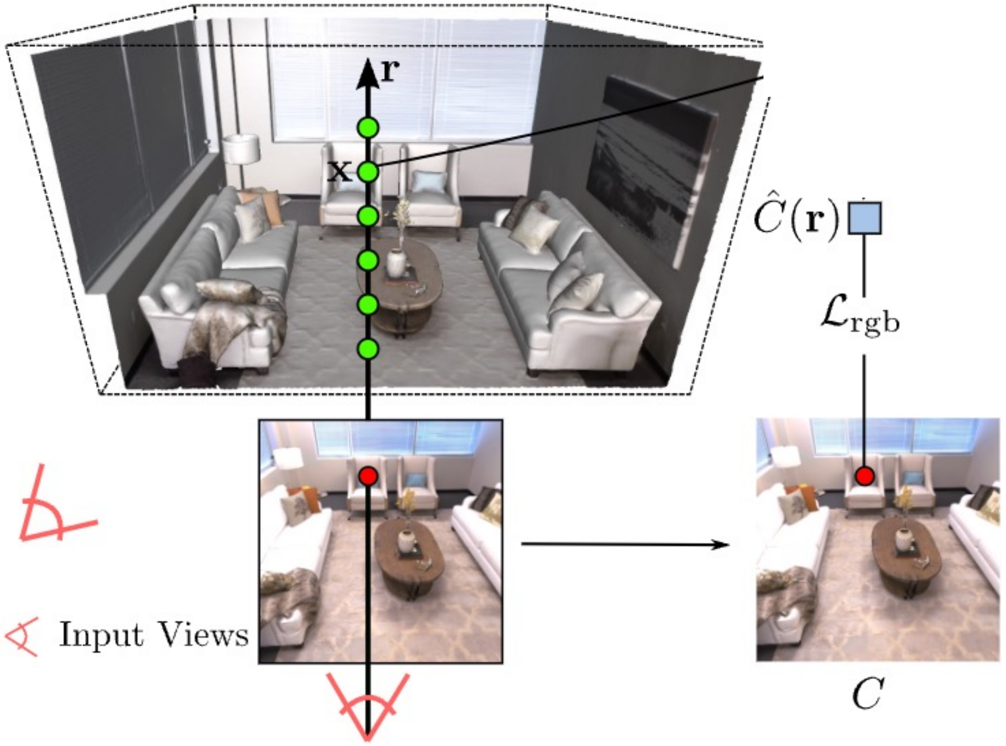
Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction

Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, Andreas Geiger

arXiv 2022



VolSDF / NeuS / UNISURF

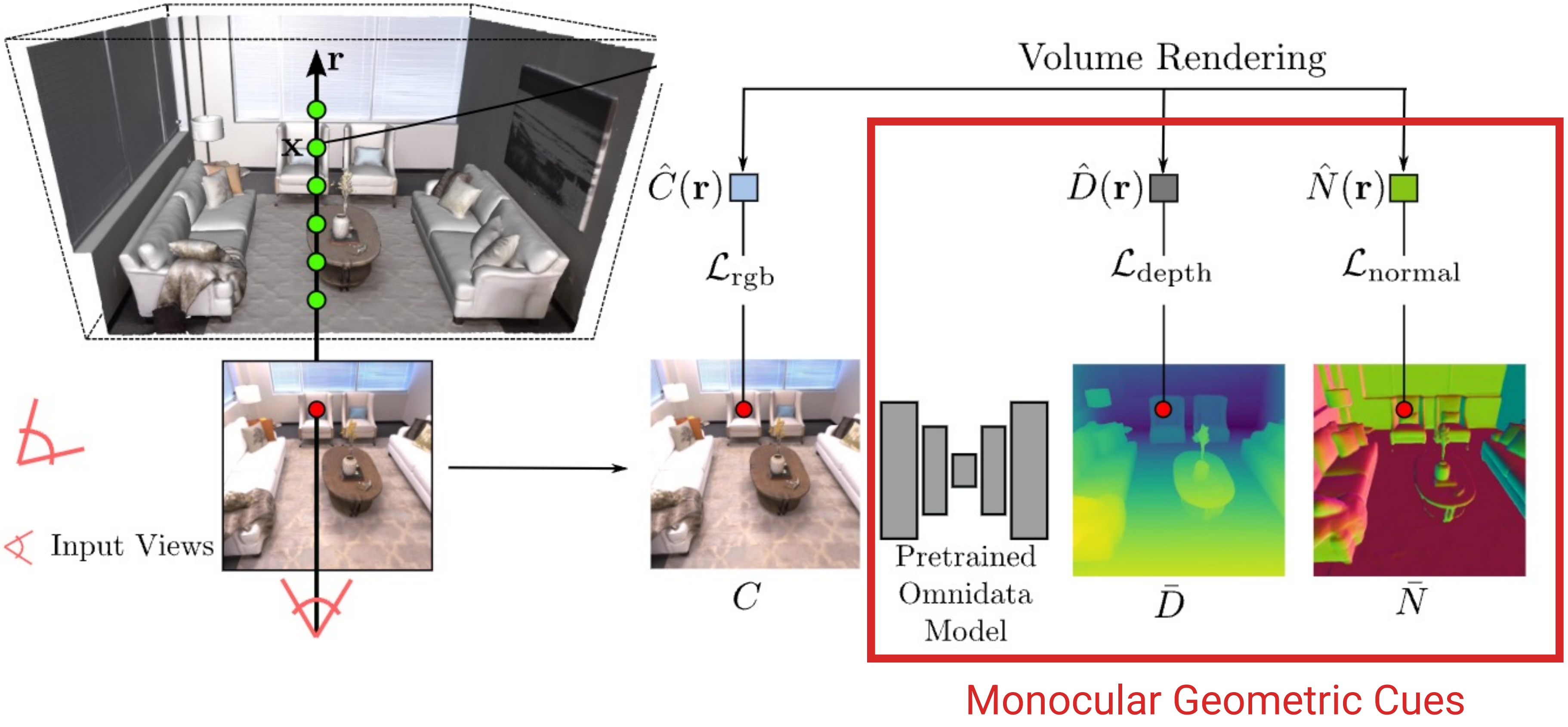


Only supervision: multi-view RGB images

MonoSDF

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1 + \|1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r})\|_1$$

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r}\in\mathcal{R}} \|(w\hat{D}(\mathbf{r}) + q) - \bar{D}(\mathbf{r})\|^2$$



Results

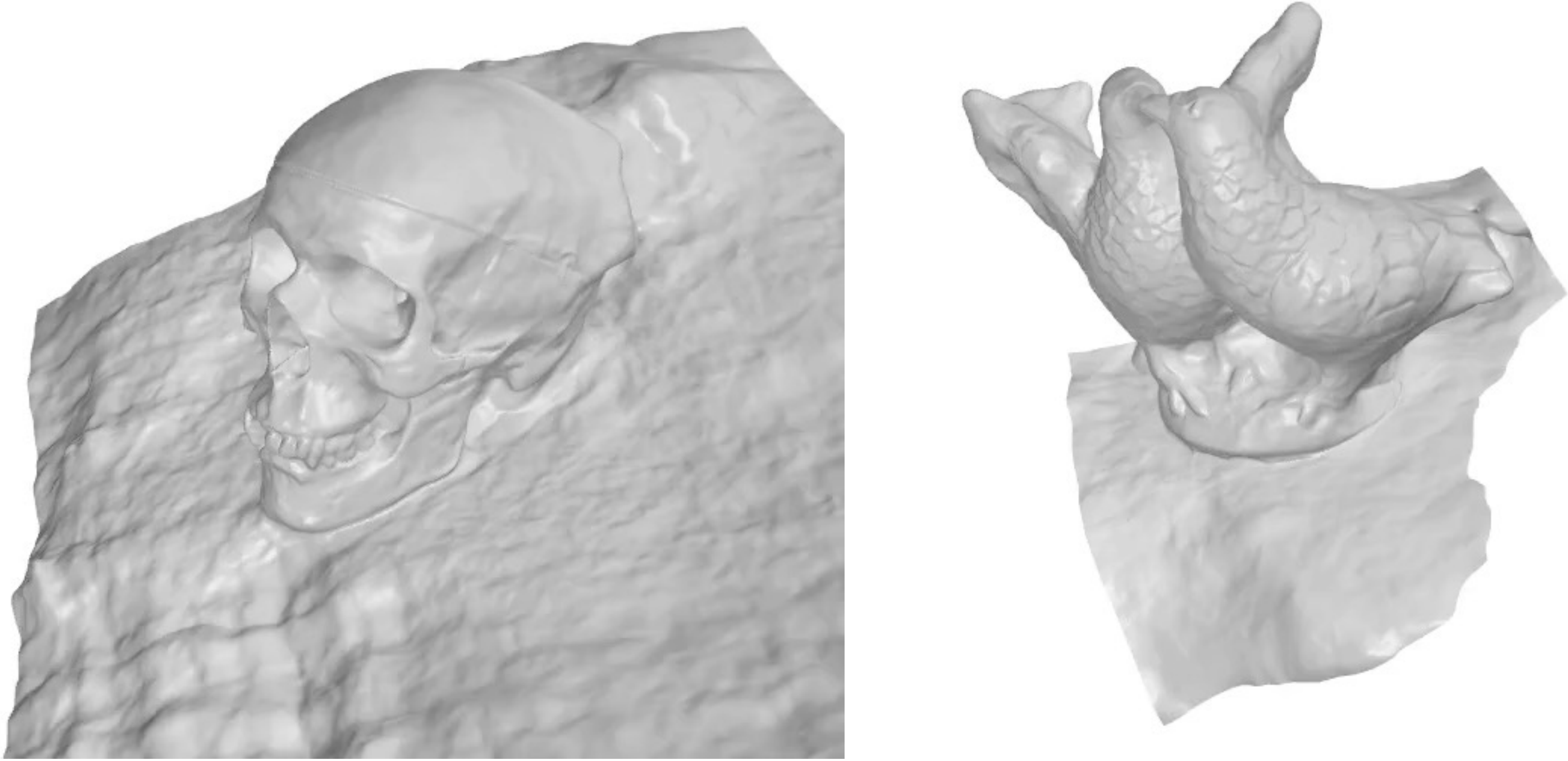
Large-scale Indoor Scenes

GT Mesh … [TOG'17]


Our Tanks & Temple Result

Results

DTU with 3 Input Views



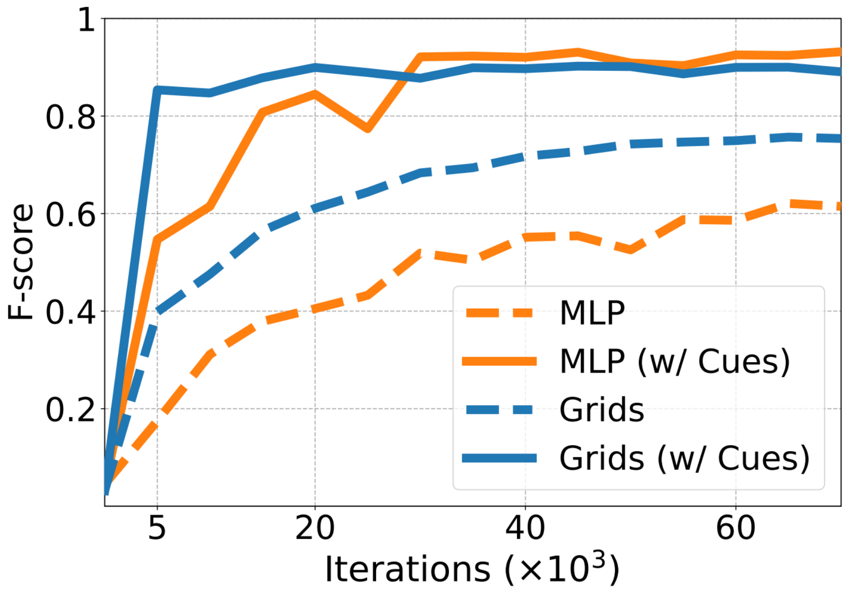
TSDF Fusion of Estimated Depths

COLMAP [ECCV'16]

VolSDF [NeurIPS'21]

Take-home Message

- Easy-to-obtain monocular cues are important!
- Also help converge faster and better!



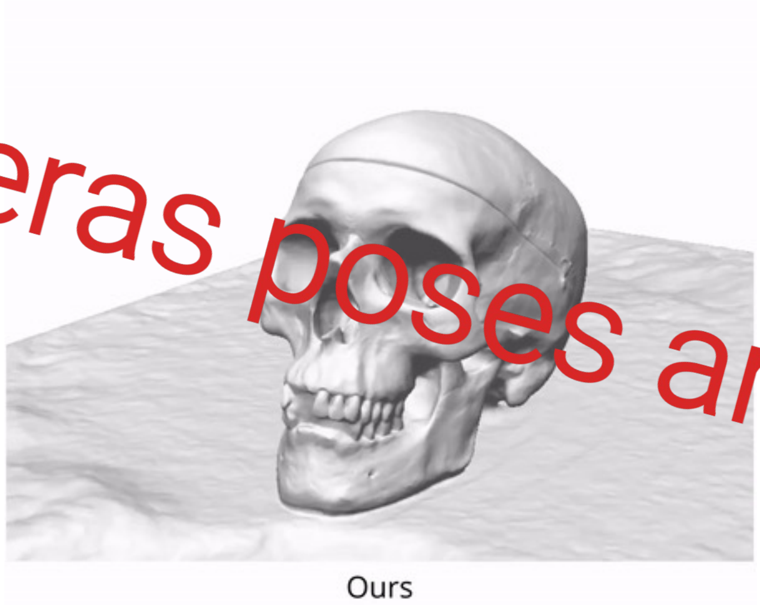
Limitations

- Depends on the quality of the monocular cues

What is missing?

KiloNeRF

UNISURF



MonoSDF

Cameras poses are given!

NICE-SLAM

Neural Implicit Scalable Encoding for SLAM

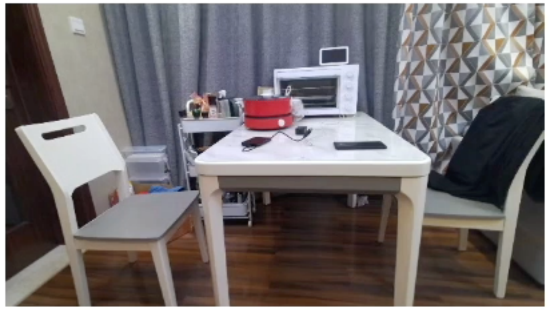
CVPR 2022

Zihan Zhu* Songyou Peng* Viktor Larsson Weiwei Xu Hujun Bao
Zhaopeng Cui Martin R. Oswald Marc Pollefeys

* Equal Contributions

ETH zürich

RGB-D Sequences



40x Speed

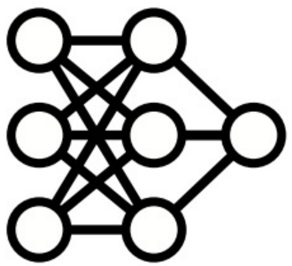
iMAP

[Sucar et al., ICCV'21]

First neural implicit-based online SLAM system

iMAP

[Sucar et al., ICCV’21]



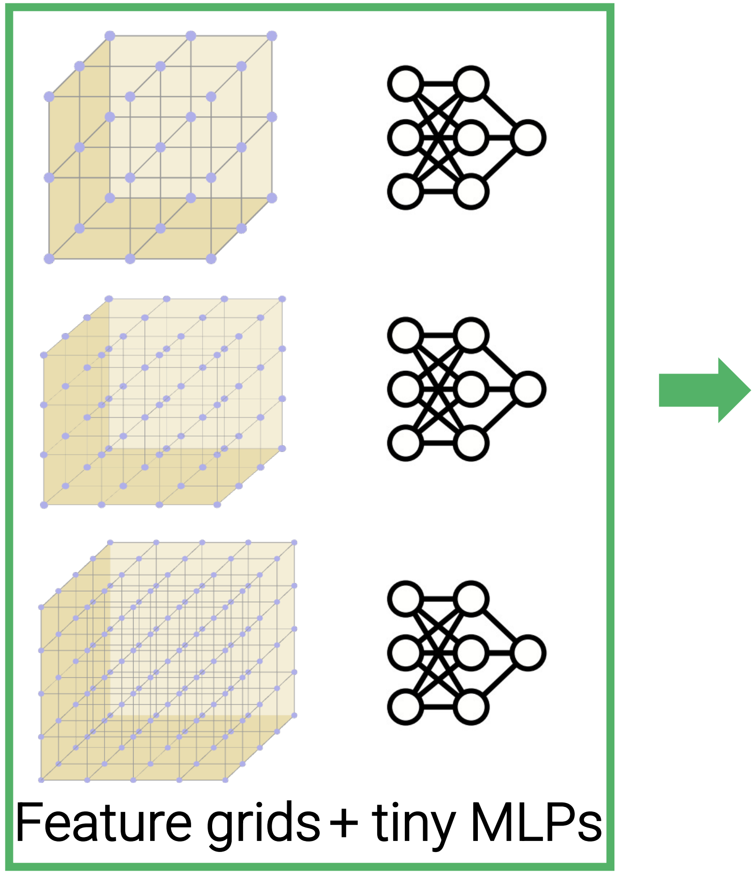
A single MLP

- Fail when scaling up to larger scenes
- Global update → Catastrophic forgetting
- Slow convergence

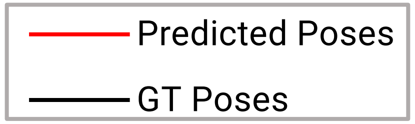
Predicted Poses

GT Poses

NICE-SLAM

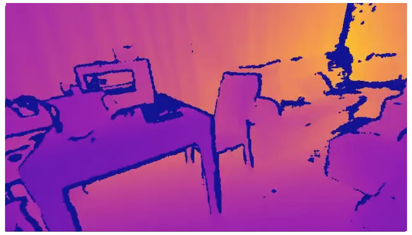


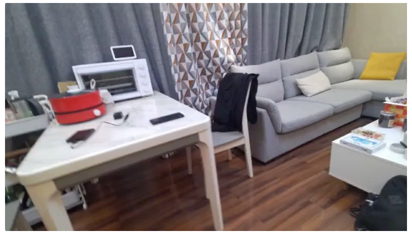
- Applicable to **large-scale scenes**
- Local update → **No forgetting problem**
- **Fast** convergence



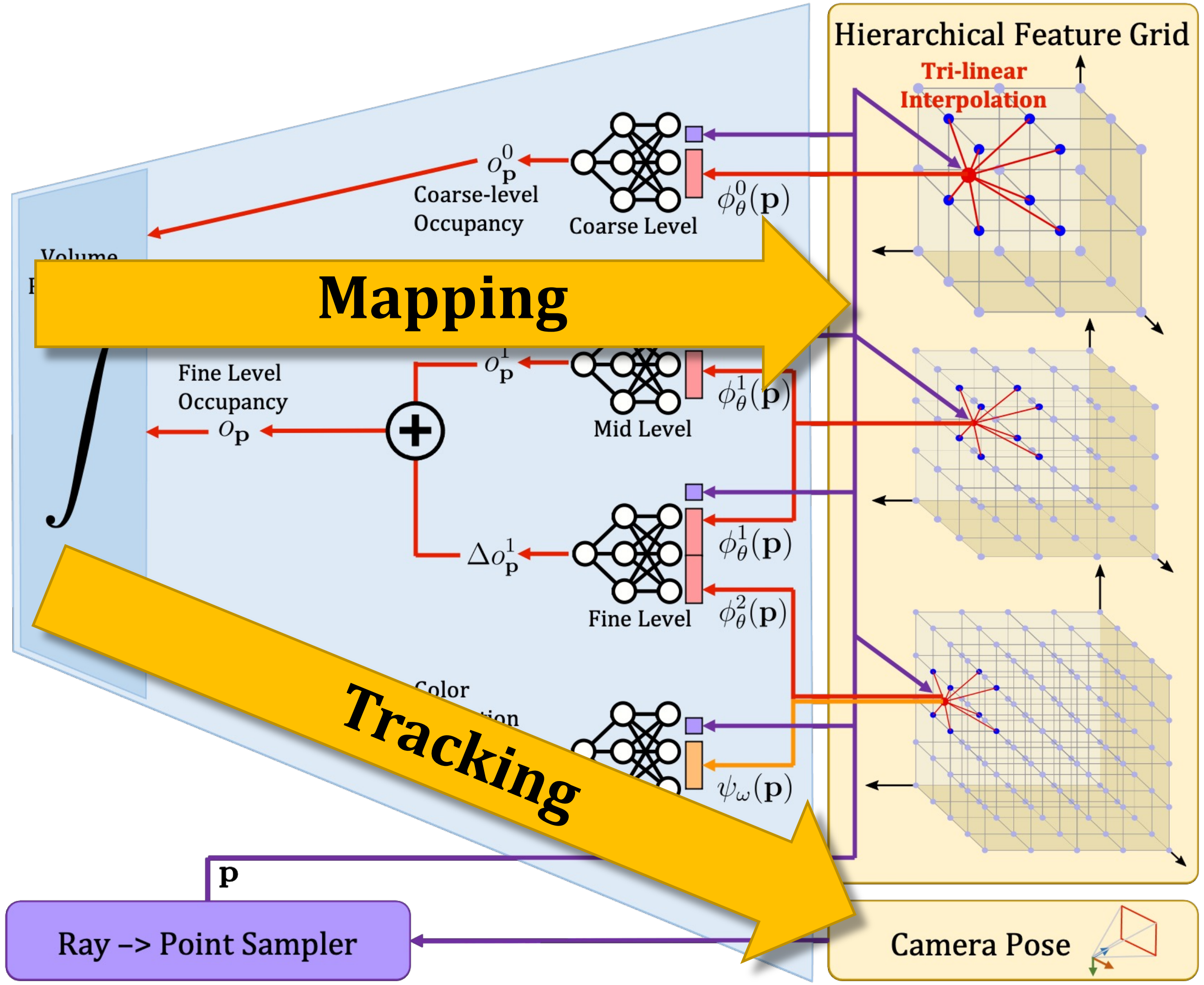
Pipeline

Input Depth





Input RGB



Results

iMAP*

(our re-implementation of iMAP)

NICE-SLAM

4x Speed

Predicted Poses

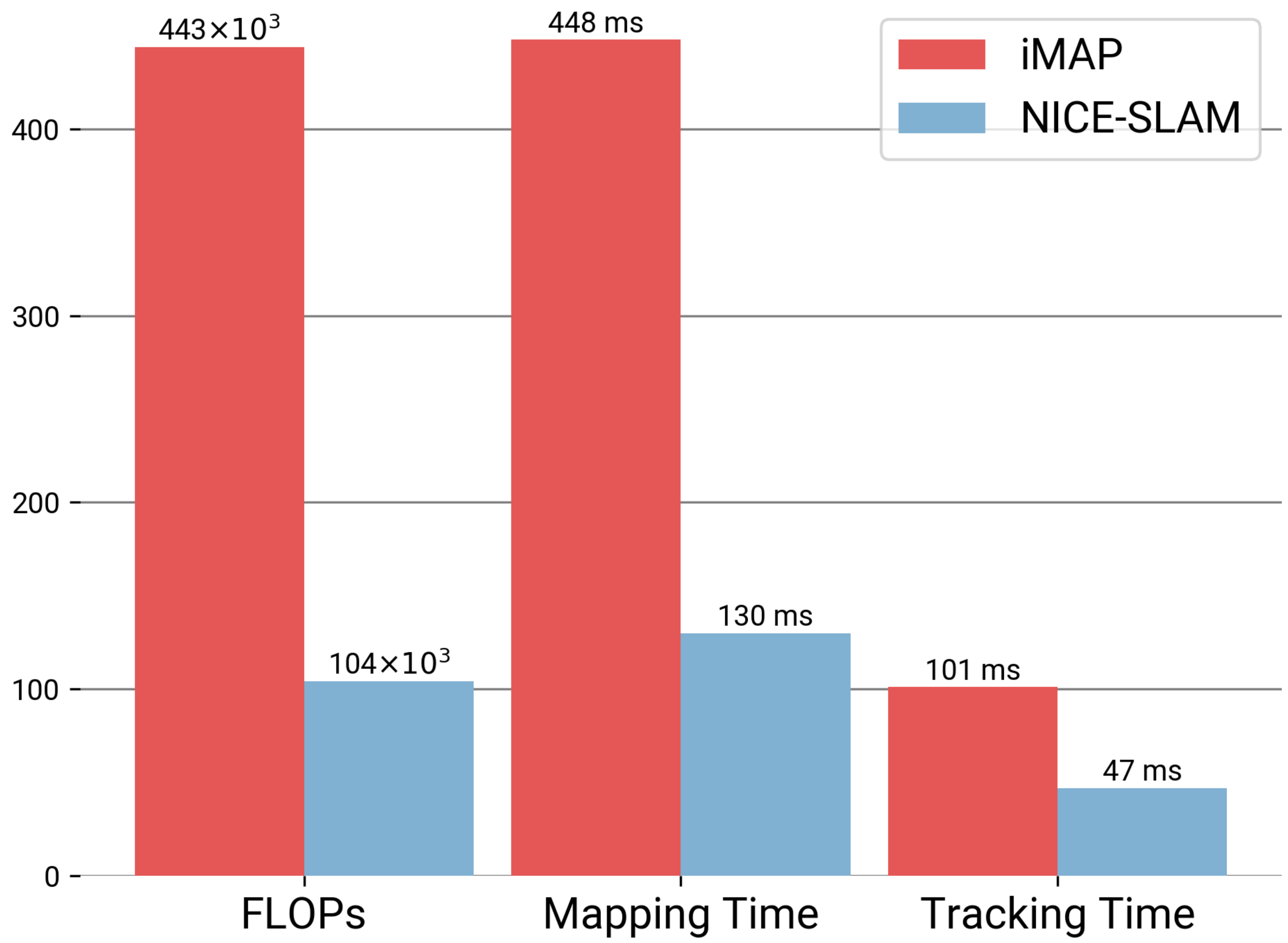
GT Poses

iMAP*

(our re-implementation of iMAP)

NICE-SLAM

10x Speed



443×10^3
448 ms
iMAP
NICE-SLAM
400
300
200
130 ms
104×10^3
101 ms
100
47 ms
0
FLOPs
Mapping Time
Tracking Time

Take-home Message

- A NICE online implicit SLAM system for indoor scenes
- Hierarchical feature grids + a tiny MLP seems to be a trend!
  - Instant-NGP [TOG]

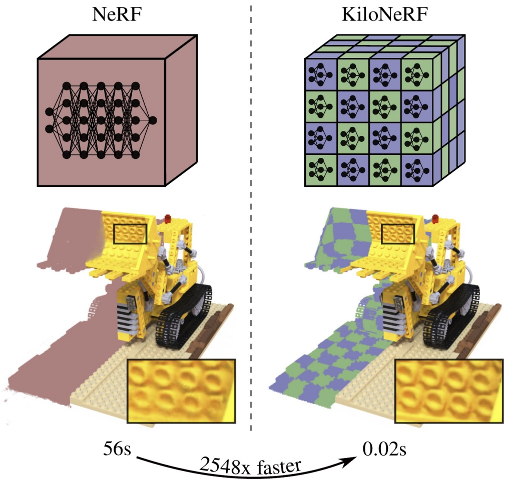
Limitations

- Requires depths as input
- Only bounded scenes
- Still not real-time

Final Remarks

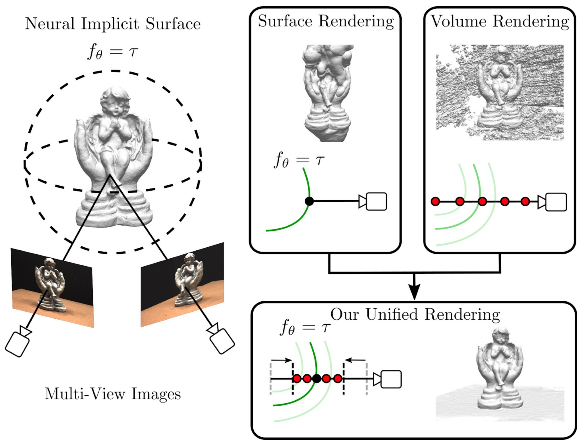
- NeRF has been sped up significantly for both rendering and optimization
- NeRF-based multi-view surface reconstruction still has rooms to improve
- A completely COLMAP-free NeRF pipeline?
- What is THE representation?

Thanks!



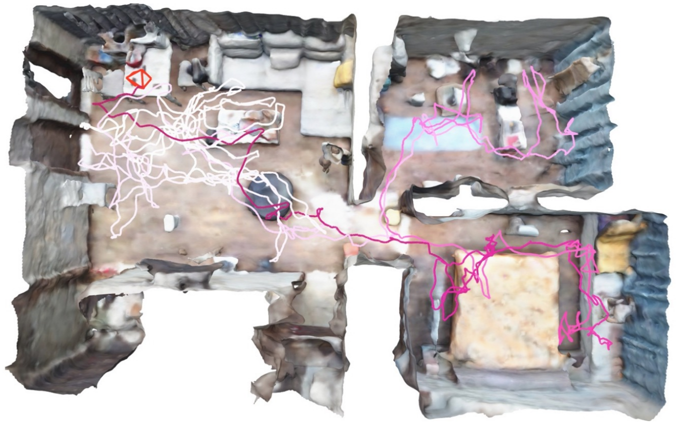
KiloNeRF

github.com/creiser/kilonerf



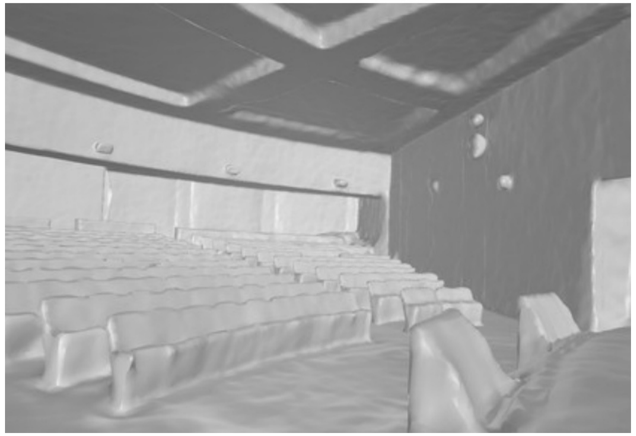
UNISURF

github.com/autonomousvision/unisurf



NICE-SLAM

github.com/cvg/nice-slam



MonoSDF

github.com/autonomousvision/monosdf