Towards Practical Applications of NeRF

for Novel View Synthesis & 3D Reconstruction

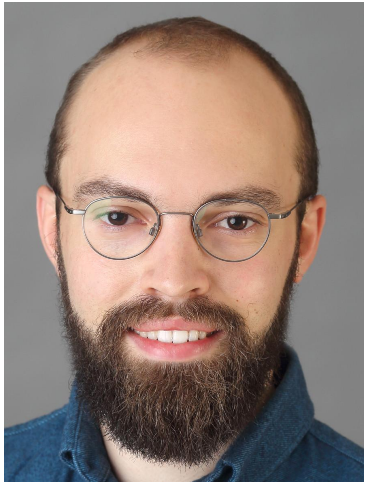
Songyou Peng

ETH zürich

MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

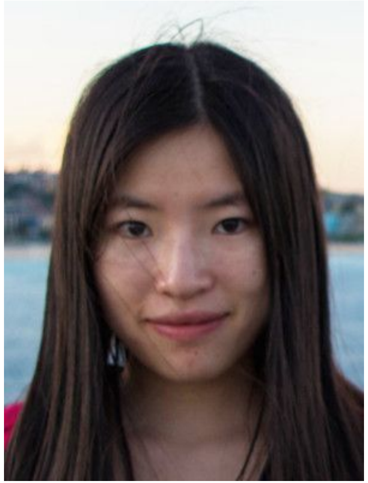
Graphics And Mixed Environment Seminar (GAMES)
21.06.2021

Collaborators



Christian
Reiser

Michael
Oechsle



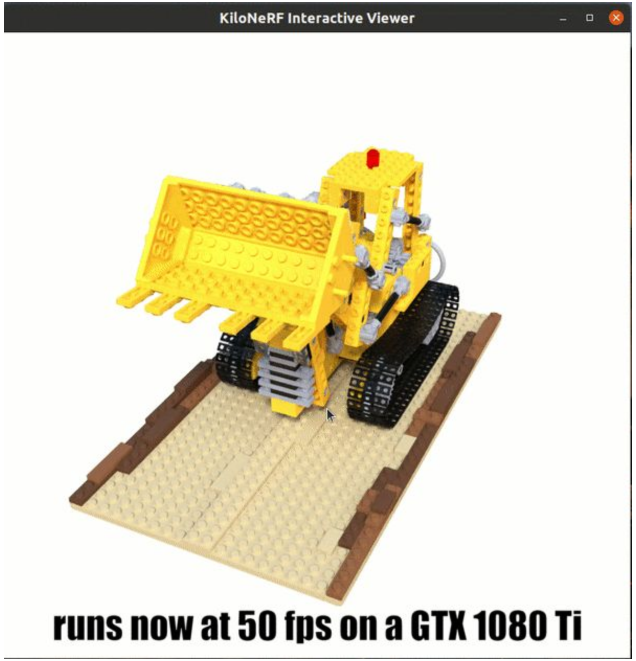
Yiyi
Liao

Andreas
Geiger

KiloNeRF: Speeding up NeRF with Thousands of Tiny MLPs

Christian Reiser, Songyou Peng, Yiyi Liao, Andreas Geiger

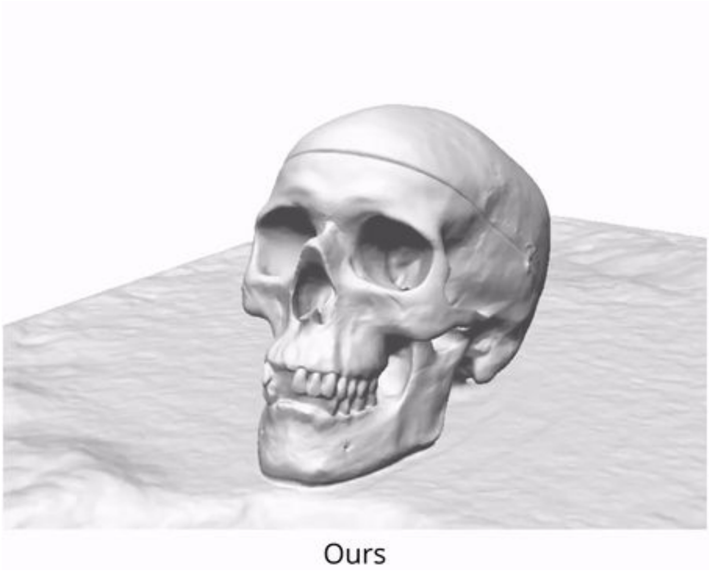
https://arxiv.org/abs/2103.13744



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

Michael Oechsle, Songyou Peng, Andreas Geiger

https://arxiv.org/abs/2104.10078



Motivation

NeRF is awesome!

But some problems still exist...

Motivation

Problem 1: NeRF's inference time is super long

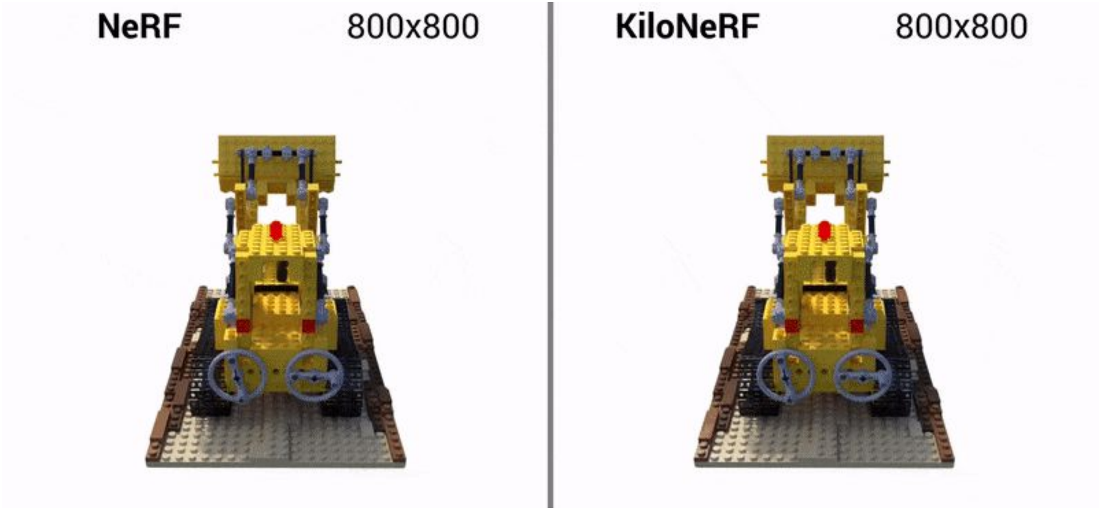
56 s

😢 Not suitable for real-world applications, e.g. VR/AR

* Test with NVIDIA GTX 1080 Ti

Motivation

Problem 1: NeRF's inference time is super long



56 s

0.02 s (50 fps)

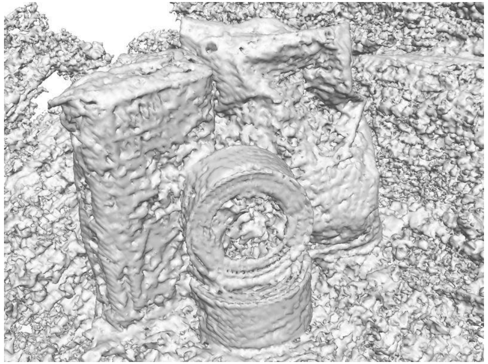
😊 **KiloNeRF** speeds up NeRF rendering by >2000x

* Tested with NVIDIA GTX 1080 Ti

Motivation

Problem 2: NeRF's underlying geometry is poor

Rendering

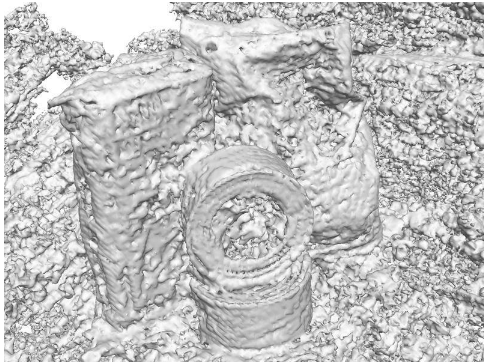


NeRF Geometry

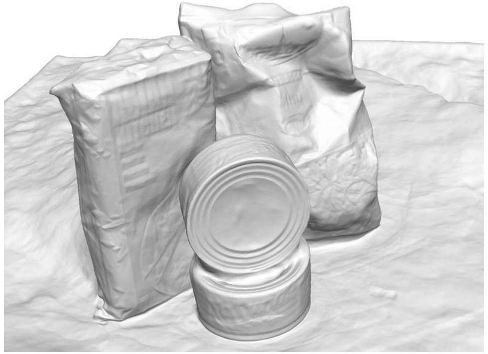
Motivation

Problem 2: NeRF’s underlying geometry is poor

Rendering



NeRF Geometry

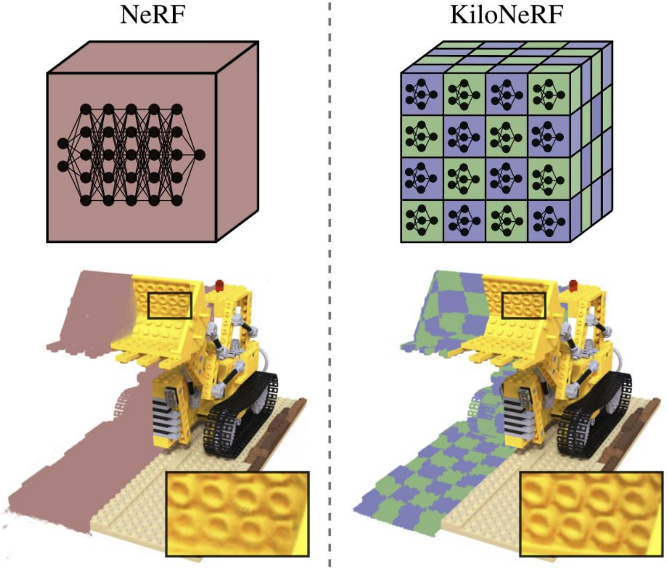


UNISURF Geometry

😊 **UNISURF** unifies NeRF & surface rendering for accurate reconstruction

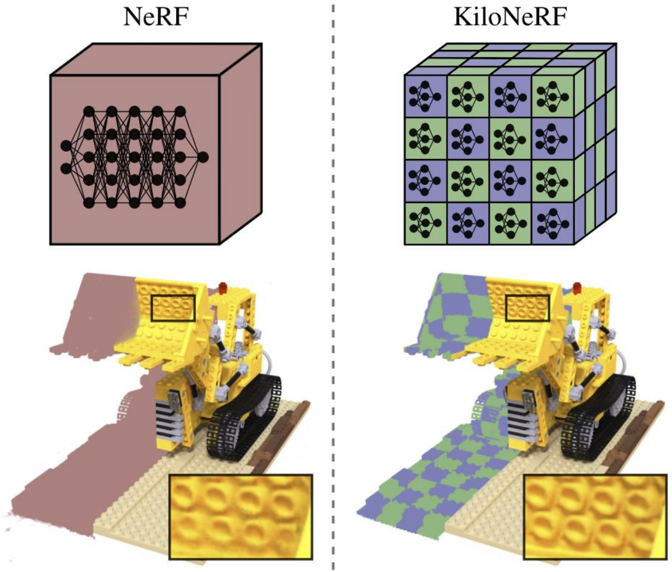
KiloNeRF

Speeding up NeRF with Thousands of Tiny MLPs



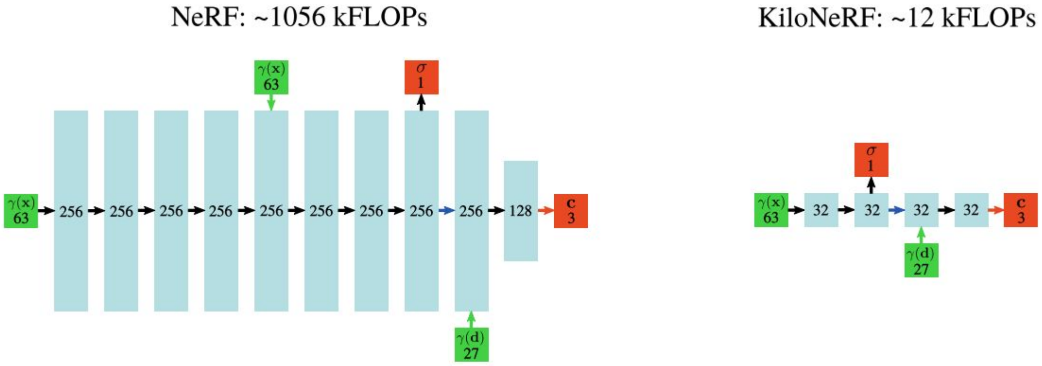
Key Idea

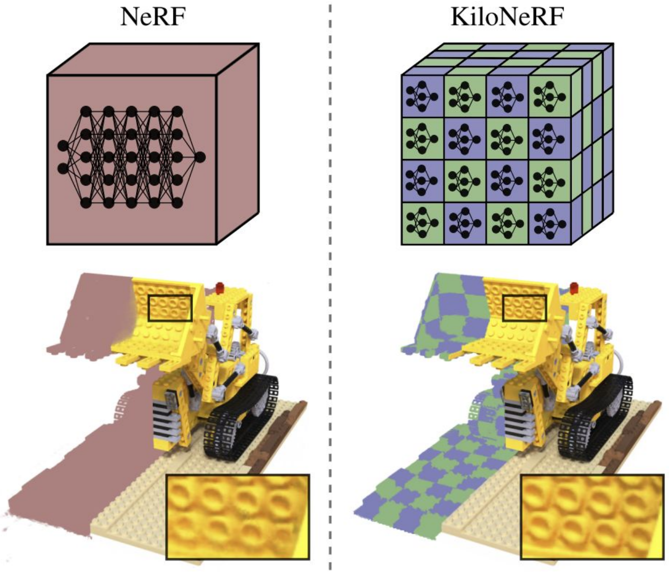
- Partition a scene into a 16^3 uniform grid
- Each grid cell is represented by a tiny MLP



Key Idea

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





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. Fine-tune the KiloNeRF model on training images

Distillation

Reason: A global NeRF implicitly enforces the multi-view consistency, while KiloNeRF is locally restricted to individual MLPs.

(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. Fine-tune the KiloNeRF model 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

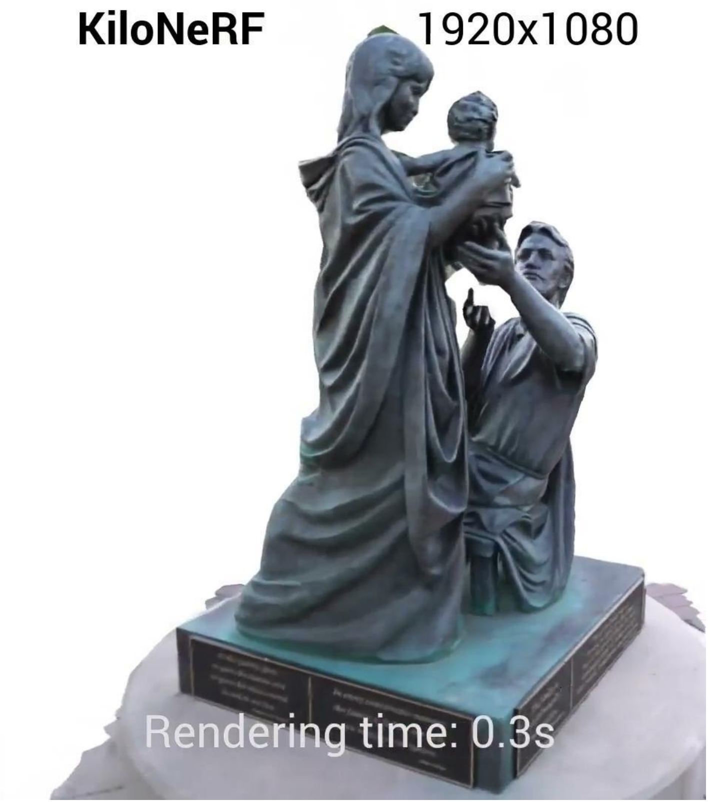
KiloNeRF

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

* Tested with NVIDIA GTX 1080 Ti

Results

Qualitative Results



Quantitative Results

Resolution		BlendedMVS 768 × 576	Synthetic-NeRF 800 × 800	Synthetic-NSVF 800 × 800	Tanks & Temples 1920 × 1080
LPIPS ↓	NeRF	0.07	0.08	0.04	0.11
	KiloNeRF	**0.06**	**0.03**	**0.02**	**0.09**
Render time (milliseconds) ↓	NeRF	37266	56185	56185	182671
	KiloNeRF	**30**	**26**	**26**	**91**
Speedup over NeRF ↑	KiloNeRF	1258	2165	2167	2002

Comparison to concurrent NeRF speed-up papers

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

⇒ KiloNeRF has a larger potential for large-scale NVS

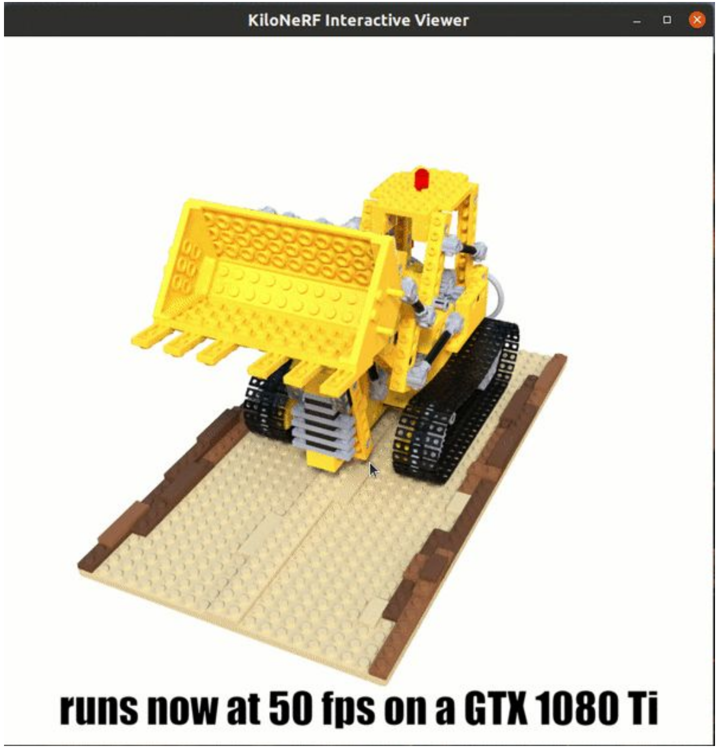
Stay tuned! We will release a blog post providing more thorough comparisons.

Conclusion

- Speed up NeRF significantly (~ 2000x) without loss of quality
- Compared to concurrent works, KiloNeRF requires much less GPU memory
- Can be plugged into almost all coordinate-based networks

Limitations

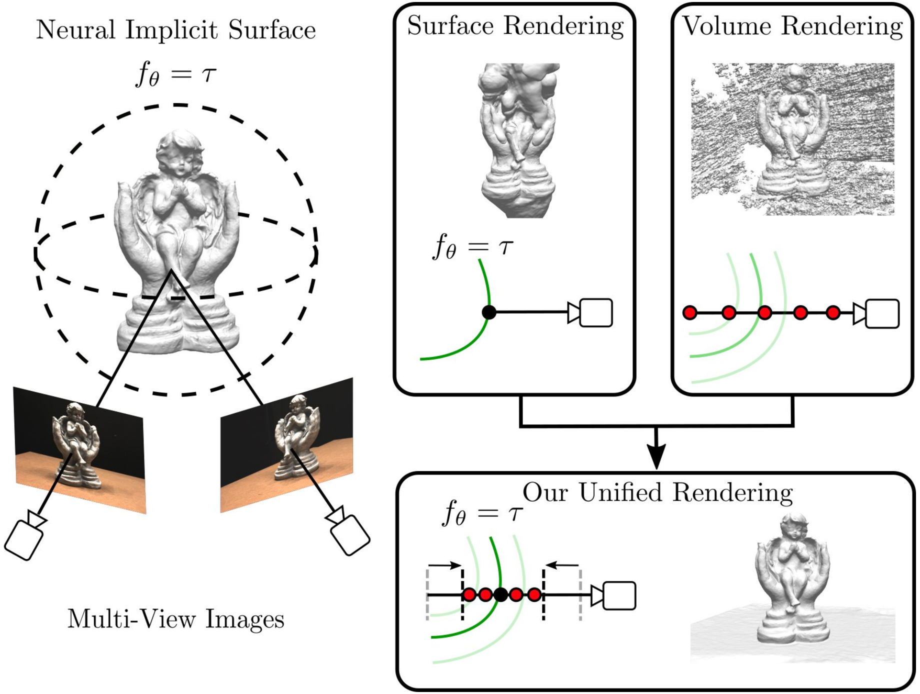
- KiloNeRF can only work on bounded scenes
  - Efficient data structures (e.g. Octree) could help to scale to larger scenes
- Expensive training time
  - Combine with PixelNeRF or MVSNeRF can help learning fast



https://github.com/creiser/kilonerf

UNISURF

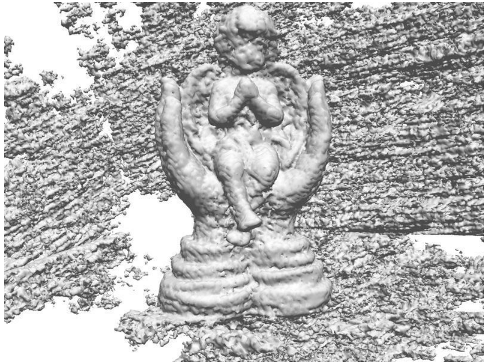
Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction



Motivation

The underlying geometry of NeRF (volume rendering) is poor [1, 2]

Rendering



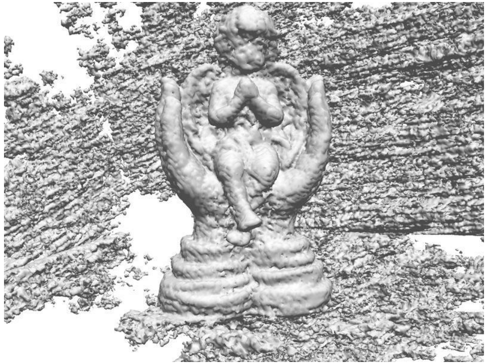
NeRF Geometry

[1] Kellnhofer et al.: Neural Lumigraph Rendering, CVPR 2021
[2] Azinovic et al.: Neural RGB-D Surface Reconstruction, 2021

Motivation

Surface rendering methods have great geometry, but require object masks

Rendering



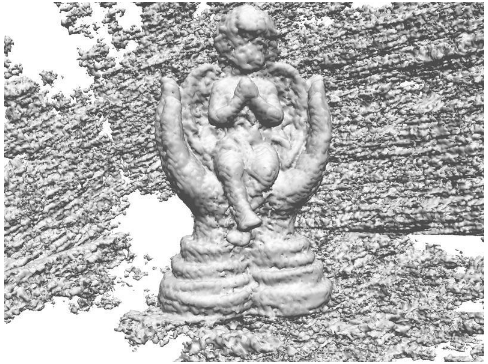
NeRF Geometry

IDR Geometry [1]

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

Can we obtain accurate geometry without the need of object masks?

Can we obtain accurate geometry without the need of object masks?



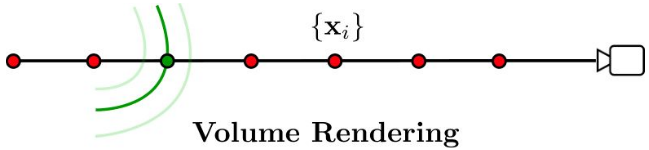
NeRF

IDR

UNISURF

We unify radiance fields and implicit surface models, enabling both **volume rendering** and **surface rendering**

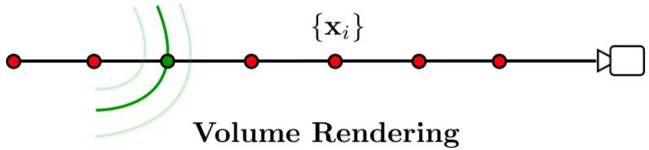
UNISURF



Early Stage: Volume rendering like in NeRF, but with occupancies

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

UNISURF



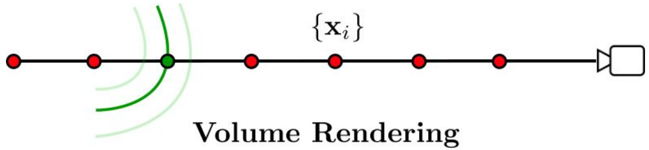
Early Stage: Volume rendering like in NeRF, but with occupancies

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

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

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

UNISURF



Early Stage: Volume rendering like in NeRF, but with occupancies

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

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

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

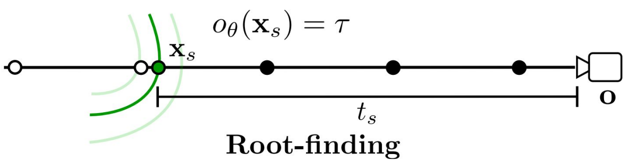
1 for the first occupied sample
0 for all other samples

➡ Sampled points near to the surface have larger influence to the predicted color

UNISURF

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

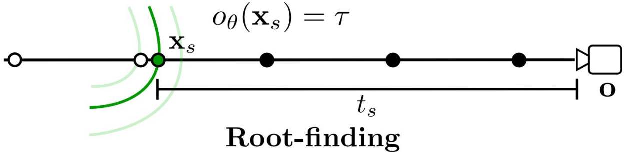
a) Find the surface point



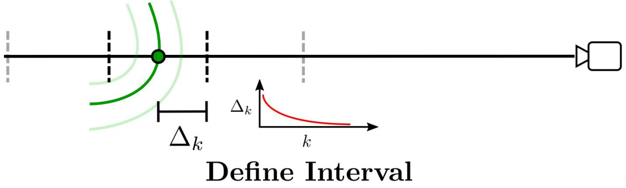
UNISURF

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

a) Find the surface point



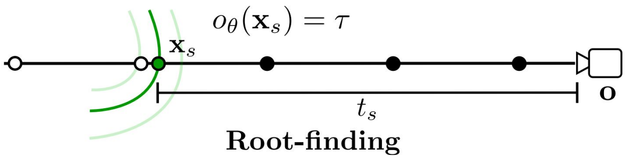
b) Define the interval



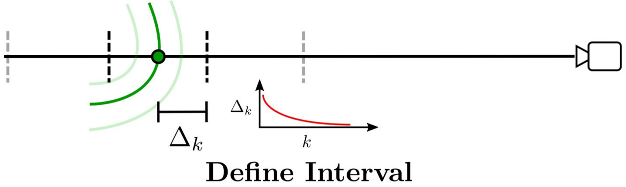
UNISURF

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

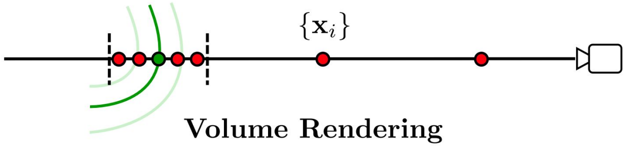
a) Find the surface point



b) Define the interval



c) Volume rendering



Loss Function

a) Image reconstruction loss

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

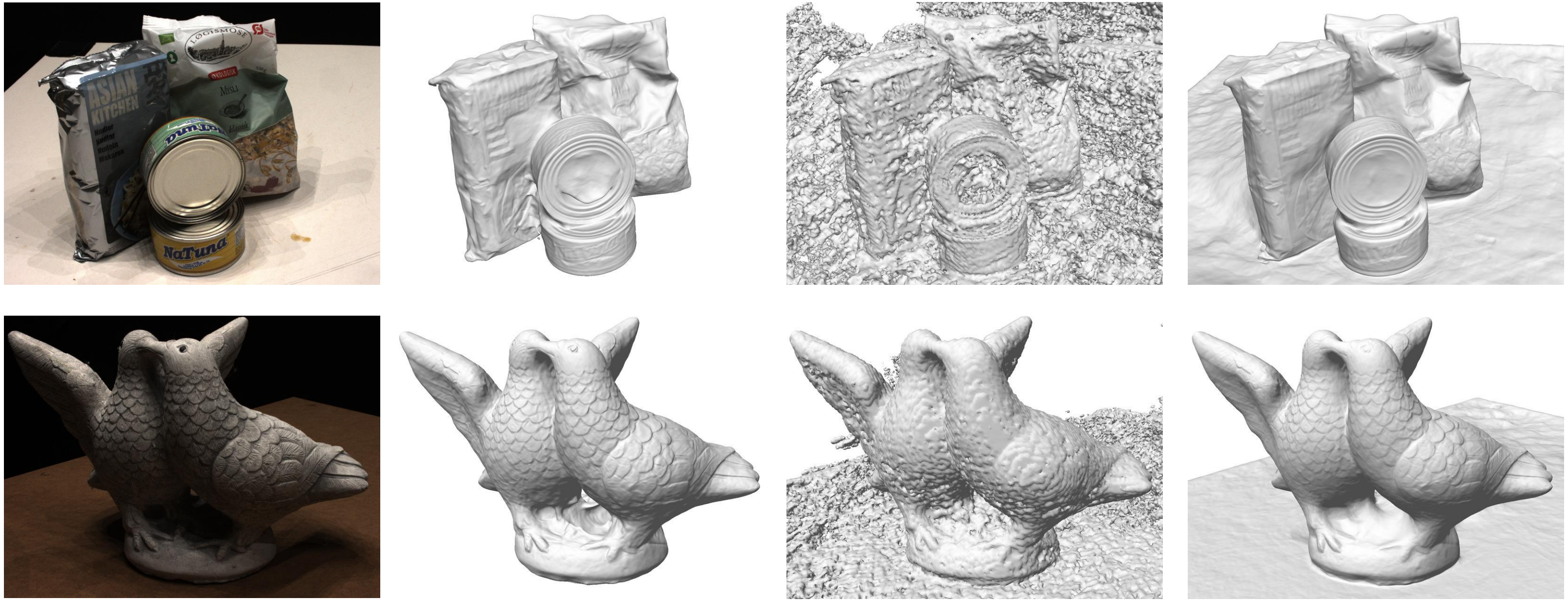
b) Surface smoothness regularization

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

Results

Results on DTU

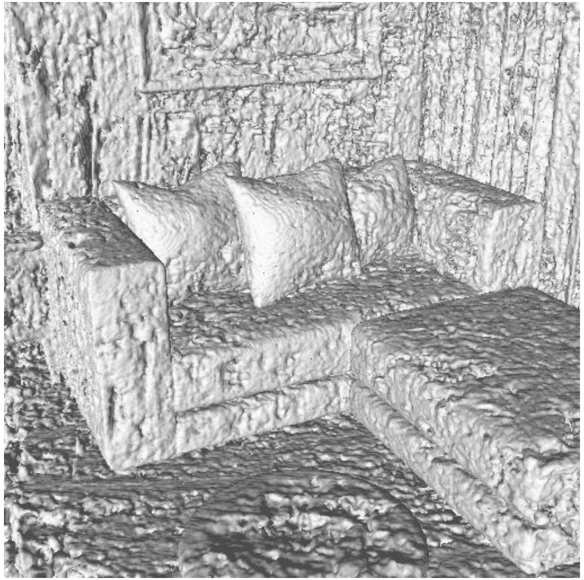
With Masks | Without Masks



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

Results on Indoor Scene

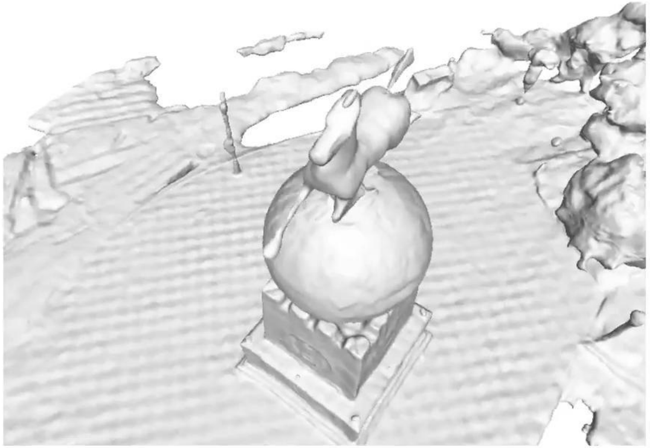
GT View



NeRF

UNISURF

Results on BlendedMVS



Conclusion

- Unify NeRF and implicit surfaces for 3D reconstruction from multi-view images
- Accurate reconstruction without the need of masks

Limitations

- Hard to reconstruct textureless regions
- Slow inference / meshing time
    - **Our latest work to tackle this point**
      Peng et al.: Shape As Points: A Differentiable Poisson Solver. https://arxiv.org/abs/2106.03452

More NeRF-related Works from Our Group

GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis

Katja Schwarz*, Yiyi Liao*, Michael Niemeyer and Andreas Geiger

NeurIPS 2020

https://github.com/autonomousvision/graf

GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields

Michael Niemeyer and Andreas Geiger

CVPR 2021 **(Best Paper Candidate)**

https://github.com/autonomousvision/giraffe

Thank you!