Implicit Mapping at Large Scale

Sven Steckler

München, 2. Dezember 2024

Implicit Mapping at Large Scale

Motivation

1. Problem Statement

- Large-scale implicit mapping is critical in fields like virtual/augmented reality (VR/AR), robotics, and autonomous navigation
- Traditional methods struggle with scalability, computational efficiency, and detail preservation

Related Work

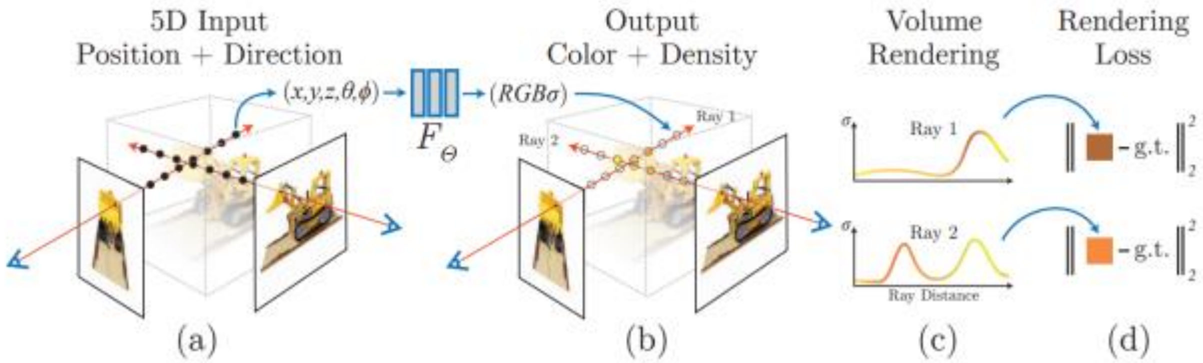
1. **Mega-NeRF:**
   - Uses Neural Radiance Fields (NeRF) as basis
   - Goal: Scaling NeRF up to large scenes and reducing training time to a minimum
   - Possible Usage: Search and Rescue

2. **SHINE-Mapping:**
   - Usage of an Octree-based structure with a shared MLP
   - Goal: Cover large areas based on 3D LiDAR for localization and navigation
   - Possible Use Cases: Mobile Robots (e.g. self-driving cars)

Method description: Mega-NeRF

1. **NeRF**:

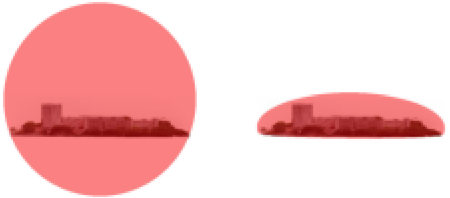
- Scene representation with continuous volumetric radiance field
- Encodes the scene in the weights of an MLP
- Through Volume Rendering Occlusion
- Positional Encoding for finer structures



Method description: Mega-NeRF

2. Architecture:

- Scenes decomposed into cells
- Each cell has its own NeRF
- Additional Usage of appearance embedding
- Centroid of each cell through tessellating
- Decomposition of the scene in foreground and background, both modelled by different Mega-NeRFs
- Usage of NeRF++ volume parametrization and raycasting formulation
  Adjustments: Ellipsoid sphere and camera pose to avoid unnecessary querying



NeRF++ <-> Mega-NeRF

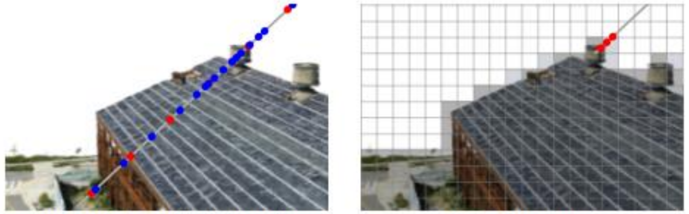
Method description: Mega-NeRF

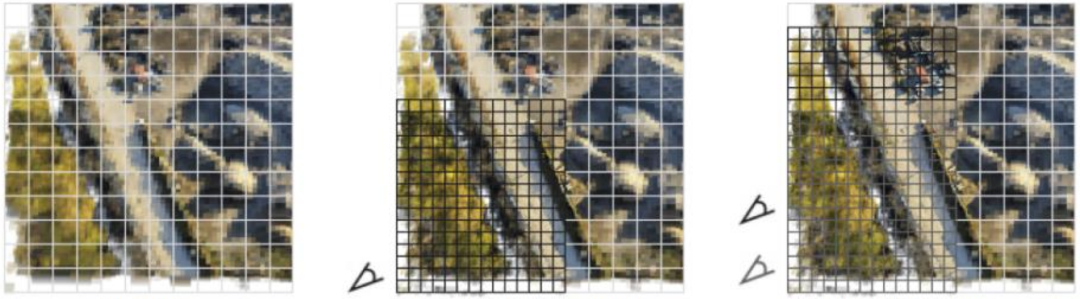
3. Training:

- Each Mega-NeRF submodule trained parallel
- Limit trainsets to relevant pixels → 10x reduction of training time
- Overlap factor 15% → avoid visual artifacts at boundaries
- Data Pruning after NeRFs get basic understanding of geometry data

4. Interactive Rendering:

- Caching of whole scene no option for this scene scales
- Precomputing a cache of opacity and color → renderer only needs to do fine adjustments
- Refined Octree gives an estimated scene geometry → ray sampling near surfaces of interest


Standard Hierachical Sampling   Guided Sampling


(a) Fixed Octree   (b) Dynamically Expanded Octree   (c) Reused Octree (next frame)

Method description: SHINE

1. Architecture:

- Octree-based map inspired by NGLOD
- Storing LiDAR Data
- Using multiple Resolution Levels to capture finer features
- Difference to NGLOD: Using Hash Tables for feature storage
- Using Morton Code for fast accessing
- SDF values inferred through a neural network (Using all resolution levels)
- Pretrained fixed MLP if mapping incrementally, for batch mode not necessary

Method description: SHINE

2. Training:

- Backpropagation possible because whole process is differentiable
- Can directly use range output of the LiDAR data as supervision
- Using sigmoid function before loss function
- Base Loss-Func.: Binary-Cross-Entropy
- Eikonal Loss for more Accuracy
- Regularization Loss against catastrophic forgetting

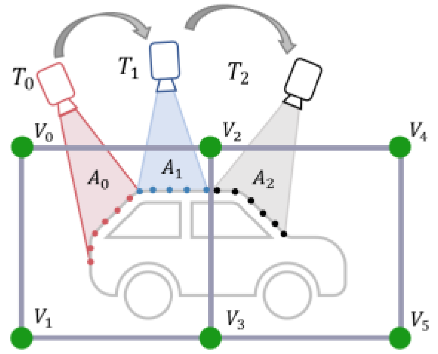
$$L_{\text{incr}} = L_{\text{bce}} + \lambda_e L_{\text{eikonal}} + \lambda_r L_r$$

$$L_{\text{bce}} = l_i \cdot \log(o_i) + (1 - l_i) \cdot \log(1 - o_i)$$

$$\underbrace{\lambda_e \left(\left\| \frac{\partial f_\theta (\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} \right\| - 1 \right)^2}_{\text{Eikonal loss}}$$

$$L_r = \sum_{i \in A} \Omega_i (\theta_i^t - \theta_i^*)^2$$

$$\Omega_i = \min \left(\Omega_i^* + \sum_{k=1}^{N} \left\| \frac{\partial L_{\text{bce}}(\boldsymbol{x}_k, l_k)}{\partial \theta_i} \right\|, \Omega_m \right)$$



Experiments and results: Mega-NeRF

Datasets:

- Mill 19 (Rubble, Building): Scenes of a former industrial complex
- Quad6k: SFM collected from Cornell University Arts Quad
- UrbanScene 3D: Scenery of a large urban Environment

Metrics:

- Peak Signal-to-Noise Ratio
- Structural Similarity Index
- Learned Perceptual Image Patch Similarity

Experiments and results: Mega-NeRF

	Mill 19 - Building				Mill 19 - Rubble				Quad 6k			
	↑PSNR	↑SSIM	↓LPIPS	↓Time (h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)
NeRF	19.54	0.525	0.512	59:51	21.14	0.522	0.546	60:21	16.75	0.559	0.616	62:48
NeRF++	19.48	0.520	0.514	89:02	20.90	0.519	0.548	90:42	16.73	0.560	0.611	90:34
SVS	12.59	0.299	0.778	38:17	13.97	0.323	0.788	37:33	11.45	0.504	0.637	29:48
DeepView	13.28	0.295	0.751	31:20	14.47	0.310	0.734	32:11	11.34	0.471	0.708	19:51
MVS	16.45	0.451	0.545	32:29	18.59	0.478	0.532	31:42	11.81	0.425	**0.594**	**18:55**
Mega-NeRF	**20.93**	**0.547**	**0.504**	**29:49**	**24.06**	**0.553**	**0.516**	**30:48**	**18.13**	**0.568**	0.602	39:43

	UrbanScene3D - Residence				UrbanScene3D - Sci-Art				UrbanScene3D - Campus			
	↑PSNR	↑SSIM	↓LPIPS	↓Time (h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)
NeRF	19.01	0.593	0.488	62:40	20.70	0.727	0.418	60:15	21.83	0.521	0.630	61:56
NeRF++	18.99	0.586	0.493	90:48	20.83	0.755	0.393	95:00	21.81	0.520	0.630	93:50
SVS	16.55	0.388	0.704	77:15	15.05	0.493	0.716	59:58	13.45	0.356	0.773	105:01
DeepView	13.07	0.313	0.767	30:30	12.22	0.454	0.831	31:29	13.77	0.351	0.764	33:08
MVS	17.18	0.532	**0.429**	69:07	14.38	0.499	0.672	73:24	16.51	0.382	**0.581**	96:01
Mega-NeRF	**22.08**	**0.628**	0.489	**27:20**	**25.60**	**0.770**	**0.390**	**27:39**	**23.42**	**0.537**	0.618	**29:03**

→Acceleration in Training time

→Also is outperforming the other Methods

Experiments and results: Mega-NeRF

best <u>second-best</u>	Mill 19					Quad 6k					UrbanScene3D				
	↑PSNR	↑SSIM	↓LPIPS	Preprocess Time (h)	Render Time (s)	↑PSNR	↑SSIM	↓LPIPS	Preprocess Time (h)	Render Time (s)	↑PSNR	↑SSIM	↓LPIPS	Preprocess Time (h)	Render Time (s)
Mega-NeRF-Plenoctree	16.27	0.430	0.621	<u>1:26</u>	**0.031**	13.88	0.589	0.427	<u>1:33</u>	**0.010**	16.41	0.498	0.530	**1:07**	**0.025**
Mega-NeRF-KiloNeRF	21.85	0.521	0.512	30:03	0.784	20.61	0.652	0.356	27:33	1.021	21.11	0.542	0.453	34:00	0.824
Mega-NeRF-Full	**22.96**	**0.588**	**0.452**	-	101	**21.52**	**0.676**	<u>0.355</u>	-	174	**24.92**	**0.710**	**0.393**	-	122
Plenoxels	19.32	0.476	0.592	-	0.482	18.61	0.645	0.411	-	<u>0.194</u>	20.06	0.608	0.503	-	0.531
Mega-NeRF-Initial	17.41	0.447	0.570	**1:08**	<u>0.235</u>	14.30	0.585	0.386	**1:31**	0.214	17.22	0.527	0.506	<u>1:10</u>	<u>0.221</u>
Mega-NeRF-Dynamic	<u>22.34</u>	<u>0.573</u>	<u>0.464</u>	**1:08**	3.96	<u>20.84</u>	<u>0.658</u>	**0.342**	**1:31**	2.91	<u>23.99</u>	<u>0.691</u>	<u>0.408</u>	<u>1:10</u>	3.219

→Best in Preprocessing for Mill 19 and Quad 6k

→Provides the best balance between quality and rendering time

	Mill 19			Quad 6k			UrbanScene3D		
	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS
Mega-NeRF-no-embed	20.42	0.500	0.561	16.16	0.544	0.643	19.45	0.587	0.545
Mega-NeRF-embed-only	21.48	0.494	0.566	17.91	0.559	0.638	22.79	0.611	0.537
Mega-NeRF-no-bounds	22.14	0.534	0.522	18.02	0.565	0.616	23.42	0.636	0.511
Mega-NeRF-dense	21.63	0.504	0.551	17.94	0.562	0.627	22.44	0.605	0.558
Mega-NeRF-joint	21.10	0.490	0.574	17.43	0.560	0.616	21.45	0.595	0.567
Mega-NeRF	**22.34**	**0.540**	**0.518**	**18.08**	**0.566**	**0.602**	**23.60**	**0.641**	**0.504**

→Each addition (e.g. embeddings, unit sphere) has a positive impact on the performance

Experiments and results: SHINE

Datasets:

- MaiCity: Sequence of 64 beam noise-free simulated LiDAR scans of urban scenario
- Newer College dataset: LiDAR measurements from Oxford University
- UrbanScene 3D: Scenery of a large urban Environment

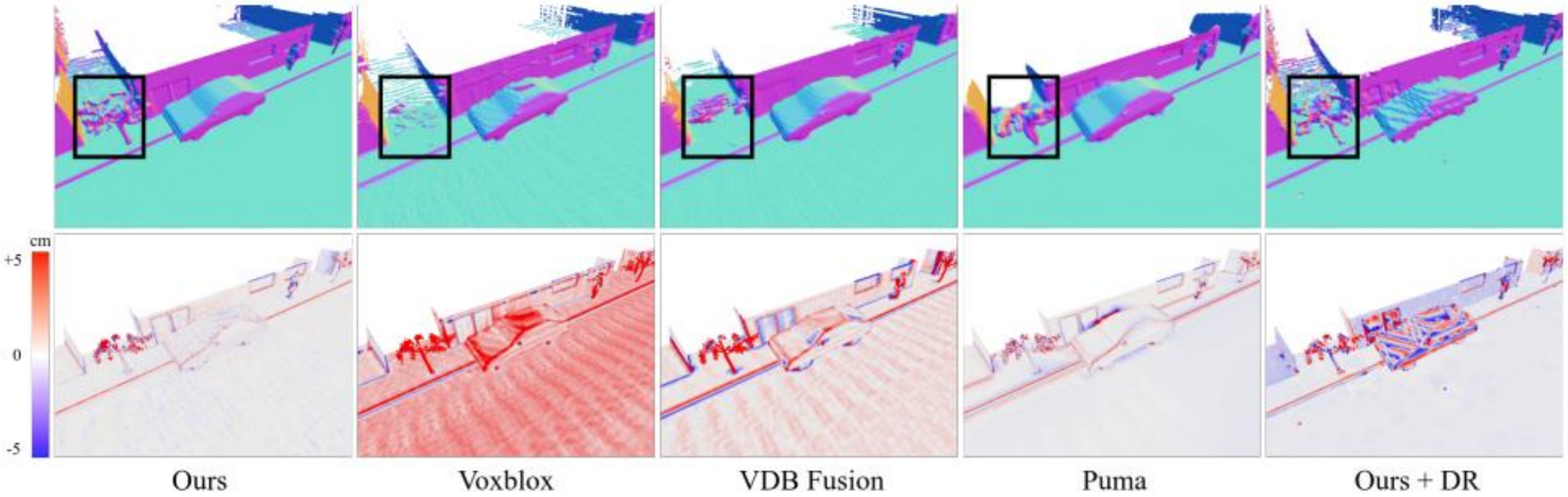
Metrics:

- Accuracy
- Completeness
- Memory efficiency

Experiments and results: SHINE

Method	Comp. ↓	Acc. ↓	C-l1 ↓	Comp.Ratio ↑	F-score ↑
Voxblox	7.1	1.8	4.8	84.0	90.9
VDB Fusion	6.9	1.3	4.5	90.2	94.1
Puma	32.0	1.2	16.9	78.8	87.3
Ours + DR	3.3	1.5	3.7	94.0	90.7
Ours	**3.2**	**1.1**	**2.9**	**95.2**	**95.9**

Method	Comp. ↓	Acc. ↓	C-l1 ↓	Comp.Ratio ↑	F-score ↑
Voxblox	14.9	9.3	12.1	87.8	87.9
VDB Fusion	12.0	6.9	9.4	91.3	92.6
Puma	15.4	7.7	11.5	89.9	91.9
Ours + DR	11.4	11.1	11.2	92.5	86.1
Ours	**10.0**	**6.7**	**8.4**	**93.6**	**93.7**



→Outperforming State-of-the-Art Methods

Personal comments

Strengths:

- SHINE-Mapping:
  - Excellent trade-off between scalability and accuracy
  - Effective use of hierarchical representation
- Mega-NeRF:
  - Pioneering parallelism for NeRFs
  - Highly practical for large-scale applications like city modeling

Personal comments

Weaknesses:

- SHINE-Mapping:
  - Potential challenges in fine-tuning hierarchical levels for diverse datasets
- Mega-NeRF:
  - Overlapping regions might introduce artifacts or redundancies

Future Work

Mega-NeRF:

- Explore applications in AR/VR where real-time rendering is critical
- Dynamic Scenes

Future Work

SHINE-Mapping:

- Explore integration with real-time mapping systems for robotics
- Improve adaptability for dynamic scenes (e.g., moving objects)
- Extend hierarchical representations to handle semantic information

Summary

- Both SHINE-Mapping and Mega-NeRF address critical bottlenecks in scaling 3D scene representation methods
- **SHINE-Mapping** introduces sparse hierarchical representations for efficient mapping
- **Mega-NeRF** demonstrates effective use of parallel processing for large-scale NeRFs
- These methods pave the way for broader applications in AR/VR, autonomous vehicles, and large-scale simulations

References

- “NeRF: Neural Radiance Field in 3D Vision, Introduction and Review” Kyle (Yilin) Gao, Graduate Student Member, IEEE, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, Member, IEEE, Jonathan Li, Fellow, IEEE
- Zhong, Xingguang, et al. "SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations." arXiv preprint arXiv:2210.02299 (2022).
- Turki, Haithem, Deva Ramanan, and Mahadev Satyanarayanan. "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
- https://www.youtube.com/watch?v=CRlN-cYFxTk
- “NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections” Ricardo Martin-Brualla*, Noha Radwan*, Mehdi S. M. Sajjadi*, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth
- “Soft Truncation: A Universal Training Technique of Score-based Diffusion Model for High Precision Score Estimation” Dongjun Kim1 Seungjae Shin 1 Kyungwoo Song 2 Wanmo Kang 1 Il-Chul Moon 1 3
- https://www.youtube.com/watch?v=0cJZn_hV2Ms