Imperial College London

dyson Robotics Lab

TUE-AM-090

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

vMAP: Vectorised Object Mapping for Neural Field SLAM




Xin Kong   Shikun Liu   Marwan Taher   Andrew Davison

https://kxhit.github.io/vMAP

Overview

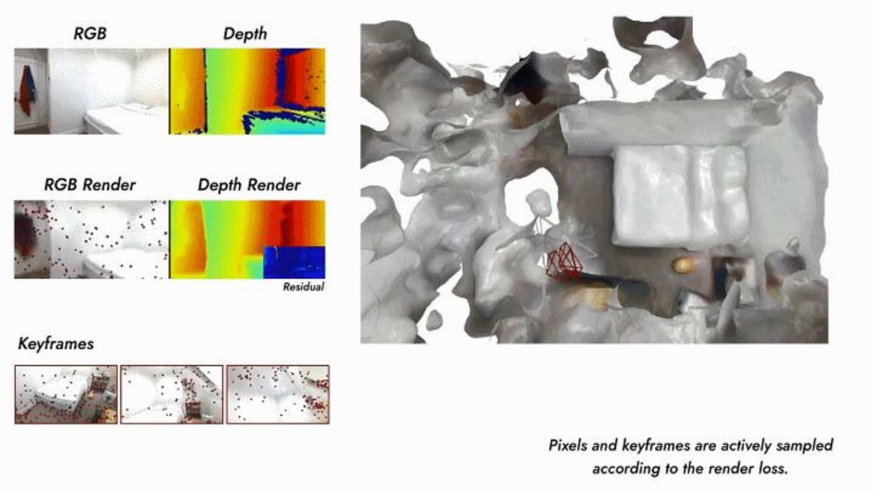
vMAP: Vectorised Object Mapping for Neural Field SLAM

CVPR 2023

Neural Field SLAM

- ☐ Not decomposable
- ☐ Unknown correspondences

iMAP [1]



- ✓ First real-time neural field SLAM
- ✓ Compact single MLP for entire 3D scene

NICE-SLAM [2]

- ✓ Hierarchical feature grids + MLP decoder
- ✓ Able to reconstruct large scenes

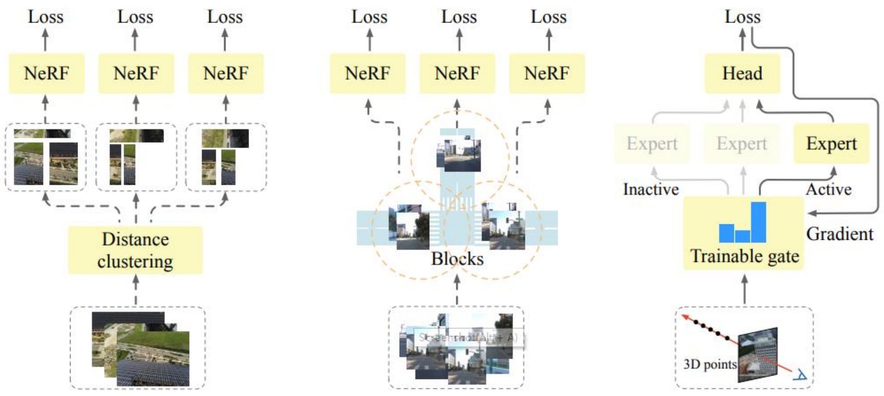
✓ Continuous surface with plausible hole-filling

[1] iMAP: Implicit mapping and positioning in real-time, ICCV'2021
[2] Nice-slam: Neural implicit scalable encoding for slam, CVPR'2022

Decomposed NeRF

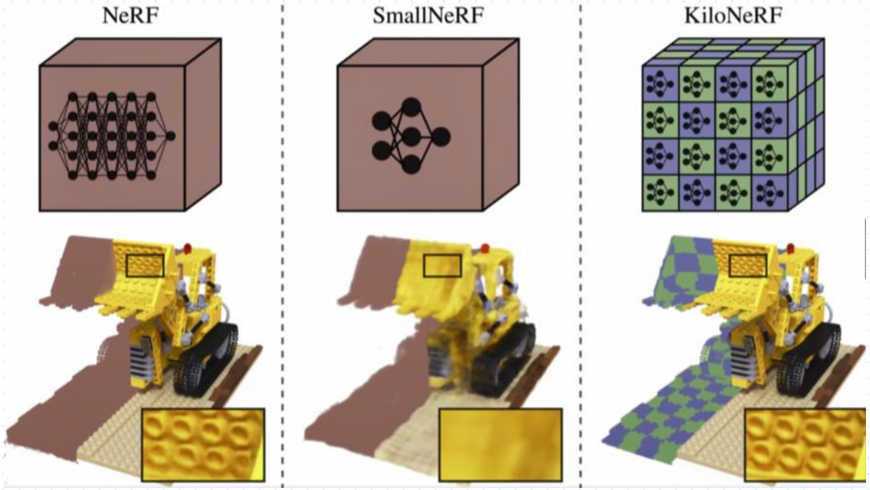
☐ No Semantics

Switch-NeRF [1]



(a) Learning after distance-based decomposition (*e.g.* Mega-NeRF)
(b) Learning after physical-distribution-based decomposition (*e.g.* Block-NeRF)
(c) Learning with scene decomposition (Ours)

Kilo-NeRF [2]



✓ 3D space-based decomposition

✓ Split 3D scene into thousand grids
✓ Many tiny MLPs speed up NeRF training

✓ Each part can be independently represented

[1] Switch-NeRF: Learning Scene Decomposition with Mixture of Experts for Large-scale Neural Radiance Fields. ICLR'2023
[2] Kilonerf: Speeding Up Neural Radiance Fields with Thousands of Tiny MLPs, ICCV'2021

Semantic NeRF

- ☐ Not real-time
- ☐ Not incremental

Feature Distillation NeRF [1]

Object NeRF [2]

Learning Object-Compositional Neural Radiance Field for Editable Scene Rendering

Bangbang Yang[1] Yinda Zhang[2] Yinghao Xu[3] Yijin Li[1]
Han Zhou[1] Hujun Bao[1] Guofeng Zhang[1] Zhaopeng Cui[1*]

[1]State Key Lab of CAD&CG, Zhejiang University
[2]Google [3]The Chinese University of Hong Kong

✓ Semantic feature field

✓ Object-level Code NeRF

✓ Semantic decomposition NeRF

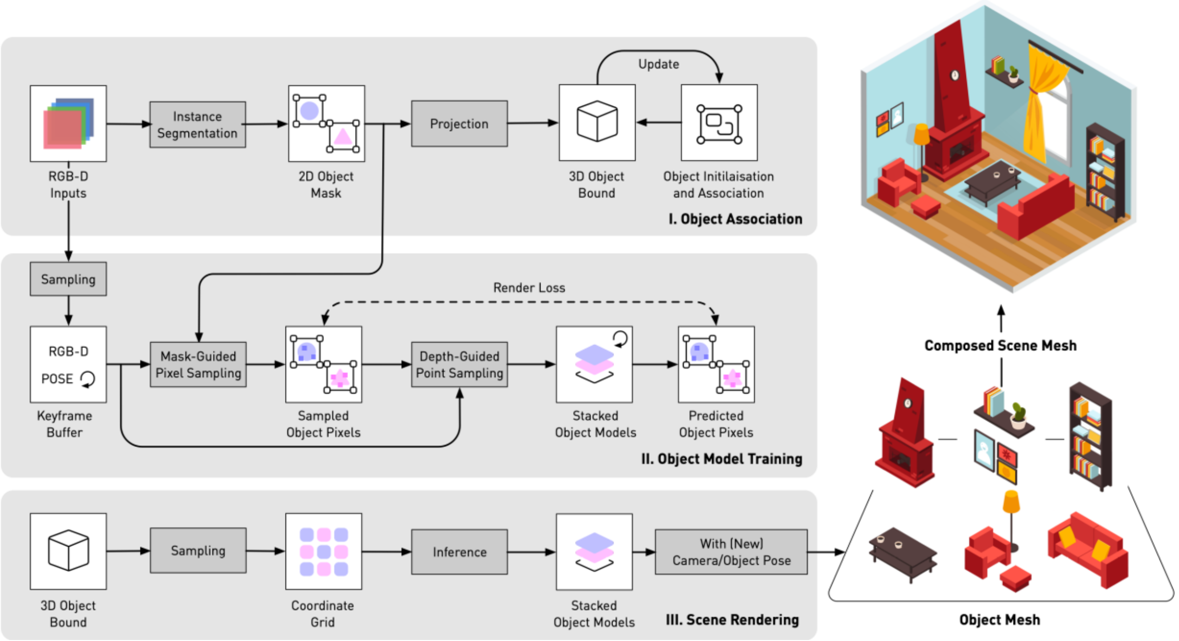
[1] Decomposing NeRF for Editing via Feature Field Distillation, NeurIPS'2022

[2] Learning Object-Compositional Neural Radiance Field for Editable Scene Rendering, ICCV'2021

System Overview

- ✓ **Object-level** representation: no 3D prior, only RGB-D video
- ✓ **Composable** map: Independently editable, trackable
- ✓ **Implicit** 3D representation: higher quality, lower memory, watertight
- ✓ **Efficient** realtime dense mapping: manage objects parallelly, easily stop and resume

Goal: A real-time object-level dense neural mapping system



Method

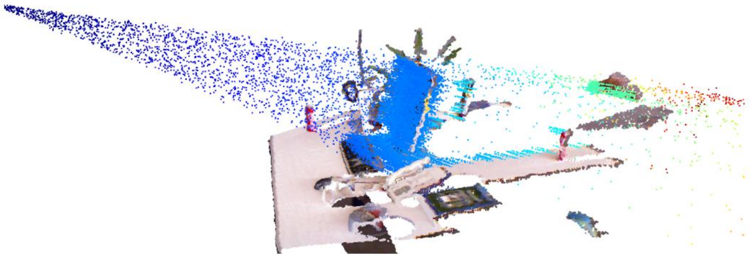


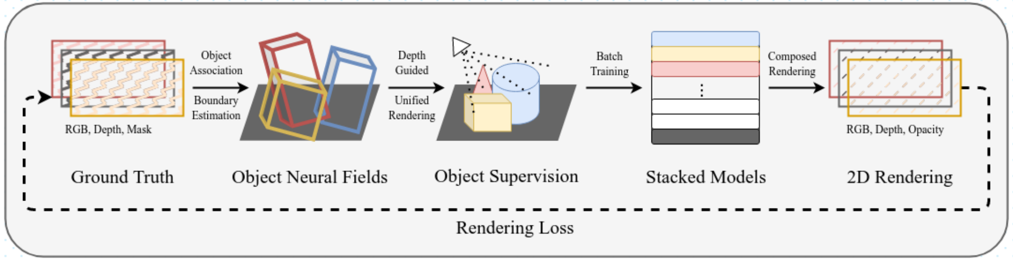
Figure A. Visualisation of depth guided sampling.

✓ Depth Guided Training

Sampling points along ray based on depth measurement
Training depth loss along with RGB
Adopt surface volume rendering to improve **geometry quality**

✓ Efficient Object Mapping

Object association by semantic-spatial consistency
Occlusion aware reconstruction
Vectorised training multiple models in one go



Plausible object completion without priors

Method

✓ Vectorised Training Demo Code

Python/PyTorch level implementation[1], no customised CUDA

1. Init batch of models with exact same structure.

```
fmodel, params, buffers = combine_state_for_ensemble(models)
[p.requires_grad_() for p in params]
optimiser.add_param_group({"params": params})
```

2. Get batch of prediction: batch_input is a stack of batch inputs in the first dim.

```
batch_pred = vmap(fmodel)(params, buffers, batch_input)
```

3. Backprop:

```
batch_loss = loss(batch_pred, batch_gt)
batch_loss.backward()
optimiser.step()
optimiser.zero_grad(set_to_none=True)
```

4. Update original models network params.

```
with torch.no_grad():
    for idx, model in enumerate(models):
        for i, param in enumerate(model.parameters()):
            param.set_(params[i][idx])
```

[1] functorch: JAX-like composable function transforms for PyTorch

Results

Vectorised v.s. Sequential

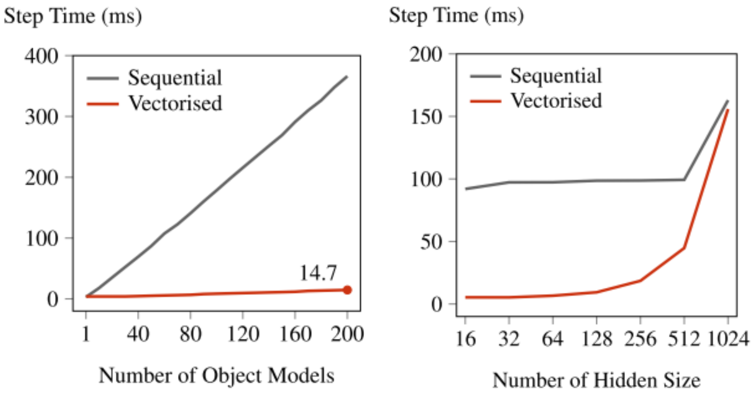


Figure 8. Vectorised operation allows extremely fast training speed compared to standard sequential operations using for loops.

Quality v.s. Model Size

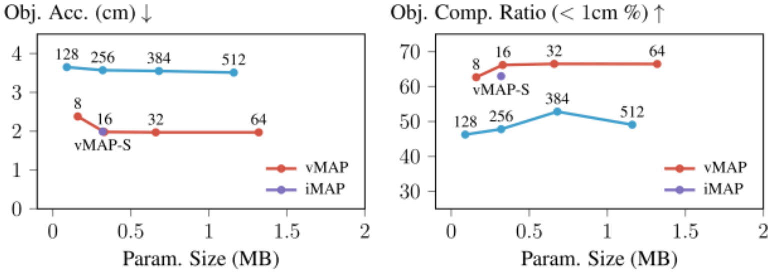


Figure 9. Object-level Reconstruction v.s. Model Param. (denoted by network hidden size). vMAP is more compact than iMAP, with the performance starting to saturate from hidden size 16.

- ✓ Object-level representation is highly compressible

Results

	Object Masks	Depth Quality	Pose Estimation
Replica	Perfect GT	Perfect GT	Perfect GT
ScanNet	Noisy	Noisy	Perfect GT
TUM RGB-D	Detic	Noisy	ORB-SLAM3
Our Recording	Detic	Noisy	ORB-SLAM3

Table 1. An overview of datasets we evaluated.

TUM RGB-D Evaluation

Camera Tracking

ATE RMSE [cm]↓	iMAP	NICE-SLAM	vMAP	ORB-SLAM2
fr1/desk	4.9	2.7	2.6	**1.6**
fr2/xyz	2.0	1.8	1.6	**0.4**
fr3/office	5.8	3.0	3.0	**1.0**

Table 3. Camera tracking results on TUM RGB-D.

- ✓ Camera tracking & Mapping reconstruction is highly interdependent
- ✓ Adopt ORB-SLAM for cleaner implementation, faster training speed, higher tracking quality

Real-time Mapping

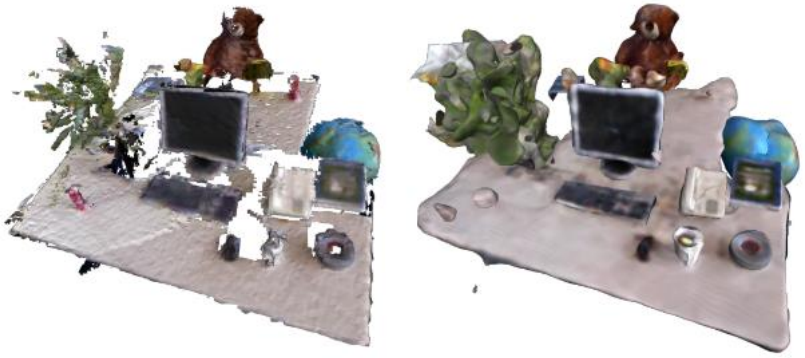


Figure 6. Visualisation of scene reconstruction from TSDF-Fusion (left) and vMAP (right) in a selected TUM RGB-D sequence, trained in real time for 99 seconds.

Results

2D Novel View Rendering

Protocol:
For each scene **randomly generate a new trajectory,** get GT 2D novel view, compared with rendered 2D results

		room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
NICE-SLAM	**Depth L1.** [cm] ↓	1.99	1.57	2.72	12.50	7.37	3.03	2.39	2.18	4.22
	PSNR. ↑	24.11	23.43	23.48	23.91	22.69	23.78	23.78	26.00	23.90
	SSIM ↑	0.73	0.74	0.82	0.83	0.82	0.83	0.84	0.85	0.81
	LPIPS ↓	0.11	0.09	0.09	0.15	0.28	0.11	0.10	0.09	0.13
NICE-SLAM*	**Depth L1.** [cm] ↓	1.87	1.63	2.94	13.43	7.63	2.83	2.62	1.97	4.36
	PSNR. ↑	24.03	23.61	23.54	23.59	23.19	22.22	23.32	26.20	23.71
	SSIM ↑	0.73	0.75	0.82	0.83	0.84	0.85	0.84	0.86	0.82
	LPIPS ↓	0.11	0.09	0.09	0.16	0.26	0.10	0.10	0.09	0.13
iMAP*	**Depth L1.** [cm] ↓	1.23	2.16	2.53	13.29	5.14	2.31	1.77	1.44	3.73
	PSNR. ↑	25.83	25.51	25.22	24.17	23.94	24.02	25.45	29.13	**25.41**
	SSIM ↑	0.77	0.79	0.86	0.83	0.87	0.88	0.89	0.90	**0.85**
	LPIPS ↓	0.09	0.07	0.07	0.17	0.22	0.08	0.07	0.07	**0.11**
vMAP	**Depth L1.** [cm] ↓	1.68	1.57	2.37	7.73	6.60	2.50	2.30	1.85	**3.33**
	PSNR. ↑	25.23	25.27	24.31	23.78	23.59	23.10	23.83	27.91	24.63
	SSIM ↑	0.77	0.78	0.85	0.84	0.88	0.88	0.88	0.89	**0.85**
	LPIPS ↓	0.09	0.07	0.08	0.16	0.23	0.07	0.08	0.07	**0.11**

Table 3. 2D novel view synthesis rendering results on the Replica dataset.

✓ Much clearer rendering, especially for foreground objects

Results

3D Scene-level & Object-level Geometry

	TSDF-Fusion*	iMAP	iMAP*	NICE-SLAM	NICE-SLAM*	vMAP
Scene Acc. [cm] ↓	**1.28**	4.43	2.15	2.94	3.04	3.20
Scene Comp. [cm] ↓	5.61	5.56	2.88	4.02	3.84	**2.39**
Scene Comp. Ratio [<5cm %] ↑	82.67	79.06	90.85	86.73	86.52	**92.99**
Object Acc. [cm] ↓	**0.45**	-	3.57	-	3.91	2.23
Object Comp. [cm] ↓	3.69	-	2.38	-	3.27	**1.44**
Object Comp. Ratio [<5cm %] ↑	82.98	-	90.19	-	83.97	**94.55**
Object Comp. Ratio [<1cm %] ↑	61.70	-	47.79	-	37.79	**69.23**

Table 2. Averaged reconstruction results for 8 indoor Replica scenes. $*$ represents the baselines we re-trained with ground-truth pose.

Scene-level metrics are dominated by the **background** meshes (Small objects have limited influence)

Object-level metrics are averaged across objects, each object contribute equally

Objects are more important than background in Robotics

Results

3D Object-level Geometry on ScanNet

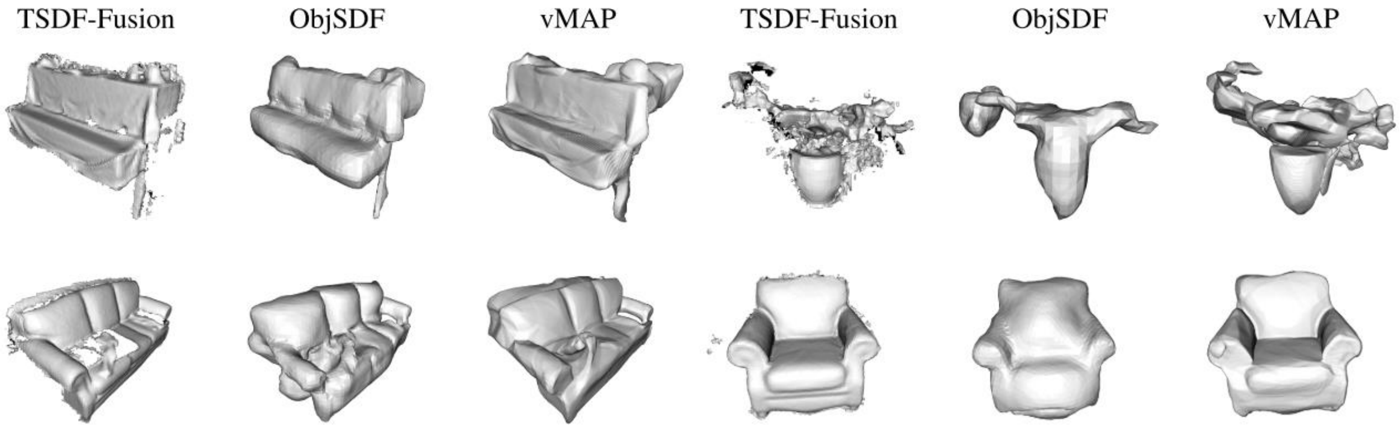


Figure 4. Visualisation of object reconstructions with vMAP compared to TSDF-Fusion and ObjSDF. Note that all object reconstructions from ObjSDF require much longer off-line training. All object meshes from ObjSDF are provided by the original authors.

[1] Object-Compositional Neural Implicit Surfaces, ECCV'2022

Results

	Object Masks	Depth Quality	Pose Estimation
Replica	Perfect GT	Perfect GT	Perfect GT
ScanNet	Noisy	Noisy	Perfect GT
TUM RGB-D	Detic	Noisy	ORB-SLAM3
Our Recording	Detic	Noisy	ORB-SLAM3

Table 1. An overview of datasets we evaluated.

https://kxhit.github.io/vMAP

ScanNet

vMAP: Vectorised Object Mapping for Neural Field SLAM

CVPR 2023

Our Recording with Kinect

vMAP: Vectorised Object Mapping for Neural Field SLAM

CVPR 2023