

Neural Graph Map: Dense Mapping with Efficient Loop Closure Integration

Supplementary Material

6. Supplementary Video

In the attached video, we first showcase the result on the Kintinuous scene visualizing the field centers and their movement during loop closure. The second part shows a comparison with an ablated, single-field version of our method (supplementing Fig. 6 in the main paper). In the last part, we show the mapping on the large-scale apt0 scene which includes two smaller loop closures. More videos can be found on our project page https://kth-rpl.github.io/neural_graph_mapping/.

7. Limitations and Discussion

While our approach enables efficient integration of loop closure into the volumetric map, it is not without drawbacks. In particular, the multi-field representation is less memory-efficient compared to monolithic neural field representations, since those can use the available network capacity relatively unconstrained, whereas this adaptiveness is constrained to the local spheres in our approach. Sharing hash tables among multiple fields might be one direction to reduce the memory overhead.

Another downside of our method is that the neural scene representation is currently not used to improve the SLAM result and a tighter integration of dense mapping and sparse tracking could lead to improved robustness. A similar limitation applies to GO-SLAM [10], which also does not use the neural map for pose estimation. In [4] this property is referred to as “decoupled”. While “coupled” approaches that use the map for frame-to-map alignment appear elegant they are not without drawbacks. Current methods for frame-to-map alignment using neural representation have a small basin of convergence and hence only work for slow sequences with small baselines. They also typically rely on depth measurements to be available. In contrast, feature-based methods can potentially estimate poses even for large baselines and can directly benefit from progress in image matching. In the future we want to investigate how to combine sparse features and neural maps in a principled way.

8. Method Details

8.1. Sampling Strategy

As described in Sec. 3.3 and illustrated in Fig. 4 in the main paper, a three-stage sampling procedure is used. First, a subset of fields is sampled, then rays are sampled for each field, and finally points are sampled along each ray. In the

following we describe the details of each stage.

Fields Especially with larger scenes, it is important that recently added fields and those currently observed are optimized with a higher rate than out-of-view fields that have already been optimized before. To achieve this, the currently observed fields $\mathcal{F}_t^{\text{obs}} \subseteq \mathcal{F}_t$ are determined and sampled with a higher probability. Specifically, a total of N_f fields are sampled; half from the currently observed fields and the remaining ones from all fields \mathcal{F}_t discarding duplicates.

Rays To sample supervision targets, each sampled field i is approximated by a set of points $\mathbf{q}_j^i, j = 1, \dots, N_{\text{approx}}$ sampled uniformly on the field’s sphere of radius r . These points are projected into all keyframes. A field is considered visible in a keyframe, if at least one of the field’s points \mathbf{q}_j^i is inside the keyframe’s frustum and the projected depth of \mathbf{q}_j^i is smaller than the observed depth at the projected 2D point. This yields a set of keyframes $\mathcal{K}_t^i \subseteq \mathcal{K}_t$. N_r rays per field (i.e., target rays) are then sampled via the 2D bounding boxes of the projected points \mathbf{q}_j^i in the keyframes \mathcal{K}_t^i . For each target ray (\mathbf{o}, \mathbf{d}) the closest point to the field center is computed as $\mathbf{o} + l_c \mathbf{d}$ and only a ray segment $[l_c - r, l_c + r]$ covering the sphere will be considered for optimization.

Points Given a ray segment $[l_{\min}, l_{\max}]$, N_{up} points are uniformly sampled across the segment, and N_{dp} points are uniformly sampled in the truncation interval τ around the observed depth, that is, in the interval $[l_{\text{obs}} - \tau, l_{\text{obs}} + \tau]$; the full ray interval is used, if there is no depth measurement or $l_{\text{obs}} \notin [l_{\min}, l_{\max}]$. This yields a total of $N_p = N_{\text{up}} + N_{\text{dp}}$ query points per ray segment during optimization.

In total, each optimization iteration will contain a maximum of $N_f N_r N_p$ query points.

8.2. K-Nearest Neighbors Queries

We compute the color and signed distance at a query point as the weighted average of the k nearest fields. Specifically, let $\mathbf{x} \in \mathbb{R}^3$ denote the query point. Let $\mathbf{c}_i, s_i, d_i, i = 1, \dots, k$ denote the returned color, signed distance, and distance to the field center for the k nearest fields for the query point \mathbf{x} . We compute weights based on the softmax of the negative distances, that is,

$$u_i = \frac{e^{-\xi d_i}}{\sum_{j=1}^k e^{-\xi d_j}}, \quad (1)$$

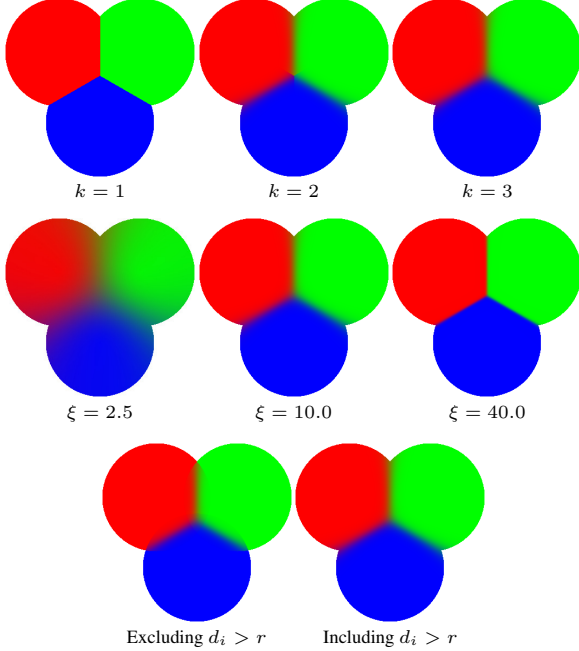


Figure 9. Visualization of k -nearest neighbor distance-based averaging. The top row shows the effect of varying k . The second row shows the effect of varying ξ . The last row shows the effect of excluding fields with distances d_i greater than the field radius r from the averaging.

where ξ determines the transition speed. The combined color and signed distance are then computed as a weighted sum, that is, $c = \sum_{i=1}^k u_i c_i$ and $s = \sum_{i=1}^k u_i s_i$.

We always use the k nearest fields even when only the closest field is within radius r . However, we set ξ sufficiently high such that the transition region becomes small and fields with $d_i \gg r$ will have no significant contribution to the final value. This is a feasible strategy, since fields are optimized for all ray segments intersecting them even when the segment is terminating outside the sphere. Hence, each field will in practice capture a region larger than a sphere with radius r .

Figure 9 illustrates the effect of this weighted averaging for different values of ξ , k , and with and without $u_i = 0$ for $d_i > r$ on a 2D toy example with three fields of fixed color. Note that the distance-weighted averaging leads to smooth transitions in the overlapping regions. When forcing $u_i = 0$ for $d_i > r$ transitions on the boundaries are unavoidable, hence we opt for the strategy described in the previous paragraph. For the experiments we use $k = 2$ and $\xi = 10$.

8.3. Parameters

In Tab. 3 we provide a full list of parameters, the used value to achieve the experimental results, and a brief description. Parameters were tuned manually and the same setting is used for all experiments (with the exception of τ ,

which is increased for the real-world datasets).

9. Experiment Details

Baseline Setup All baselines are evaluated using the parameters published as part of the published code. For additional datasets for which no parameters were provided, the most similar dataset’s parameters were adopted (i.e., for Replica-Big the provided setup for Replica is used; for Kintinuous the setup for ScanNet). Scene boundaries were manually adjusted to cover the observed area with extra margin to account for errors in positioning.

In our experiments, we noticed that Co-SLAM [9] uses the ground-truth pose of the first frame to initialize the SLAM system, which leads to axis-aligned planes. We found that planes (such as walls, floors, and ceilings) which are axis-aligned are significantly better completed using the one-blob encoding [5] than generally-oriented planes. Therefore, for a fair comparison, we modified Co-SLAM’s implementation to start from a random orientation instead. We note that this mainly reduces qualitative scene completion, however, on one of the Replica scenes it leads to tracking issues and hence poor reconstruction results. Figure 10 shows an example of the scene completion capability of Co-SLAM with and without ground-truth initialization (i.e., with and without axis-aligned planes).

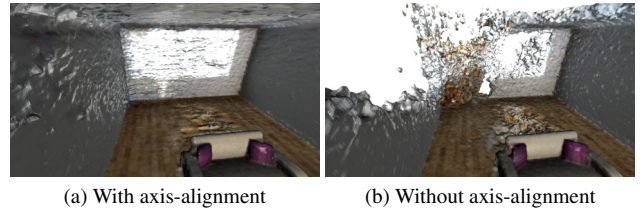


Figure 10. Co-SLAM result with and without ground-truth initialization. Ground-truth initialization leads to axis-aligned walls and floors, which in turn leads to significantly better scene completion.

9.1. Evaluation Protocol

Let ${}^w\mathcal{M}_{\text{gt}}$ and ${}^w\mathcal{M}_{\text{est}}$ denote the ground-truth mesh and estimated mesh, respectively. We assume that ${}^w\mathcal{M}_{\text{est}}$ has already been globally aligned with ${}^w\mathcal{M}_{\text{gt}}$, which is typically achieved by either aligning the first frame in the sequence or by aligning the trajectories using the Umeyama algorithm. Starting from ${}^w\mathcal{M}_{\text{gt}}$ and ${}^w\mathcal{M}_{\text{est}}$ further preprocessing steps are performed before the evaluation metrics are computed.

1. Unobserved parts of the ground-truth mesh ${}^w\mathcal{M}_{\text{gt}}$ are removed. In particular, we apply two removal steps. First, vertices falling more than 2 cm outside the scene bounding box are removed. The scene bounding box is computed as the intersection of a manually-set bounding box (used to exclude outliers in the depth map

Table 3. Overview of parameters.

Parameter	Value	Description
r	1 m	Field radius
N_f	2	Number of fields optimized in parallel in each iterations
N_r	512	Number of ray segments sampled per field during optimization
N_{up}	8	Number of uniformly sampled points distributed along each ray segment during optimization
N_{dp}	16	Number of depth-guided points distributed along each ray segment during optimization
τ	0.1 m or 0.2 m*	Truncation distance; used for scaling depth-guided sampling and for capping the supervision range (i.e., dividing samples into free-space samples and TSDF samples)
η	20.0	Determines how fast occupancy probability decays around surfaces
k	2	Number of nearest neighbors used during evaluation
ξ	10.0	Distance weighing determining transition speed between two fields
L	1	Number of MLP layers following the permutohedral encoding; excluding the final linear layer
T	2^{12}	Hash table size for permutohedral encoding
N_{levels}	16	Number of resolution levels for permutohedral encoding
N_{fpl}	16	Number of features per level for permutohedral encoding
r_{coarse}	0.1	Coarsest resolution for permutohedral encoding
r_{fine}	0.0001	Finest resolution for permutohedral encoding
λ_{color}	1.0	Weight of color loss
λ_{depth}	1.0	Weight of depth loss
λ_{fs}	40.0	Weight of free-space loss
λ_{tsdf}	50.0	Weight of TSDF loss
δ	5 cm	Huber loss threshold
γ	1×10^{-3}	Learning rate used for Adam optimizer [3]
λ	1×10^{-5}	Weight decay used for Adam optimizer [3]

* The truncation distance is increased for real-world datasets to account for the increase in depth noise.

present in some scenes) and an automatically computed bounding box (based on the ground-truth trajectory and depth maps). Second, vertices that are not in front or up to 3 cm behind any rendered depth map are removed. These depth maps are rendered from the ground-truth trajectory and from additional virtual views manually placed to improve the evaluation of scene completion (same as in Co-SLAM [9]). This yields a culled ground-truth mesh used for evaluation ${}^w\mathcal{M}_{gt}^*$.

2. To further equalize slight differences in alignment between different methods, we perform another alignment step using point-to-plane-based iterative closest point from ${}^w\mathcal{M}_{est}$'s vertices to ${}^w\mathcal{M}_{gt}$'s vertices yielding an aligned estimated mesh ${}^w\mathcal{M}_{est}$.
3. The aligned estimated mesh ${}^w\mathcal{M}_{est}$ follows the same removal process as the ground-truth mesh (see step 1 above) yielding the culled estimated mesh used for evaluation ${}^w\mathcal{M}_{est}^*$.

For the evaluation, $N_{samples} = 200\,000$ points are uniformly sampled on both meshes yielding the point sets $\mathcal{G} = \{\mathbf{x}_i \sim \mathcal{U}({}^w\mathcal{M}_{gt}^*) \mid i = 1, \dots, N_{samples}\}$ and $\mathcal{E} = \{\mathbf{y}_i \sim \mathcal{U}({}^w\mathcal{M}_{est}^*) \mid i = 1, \dots, N_{samples}\}$, where $\mathcal{U}(\cdot)$ denotes the uniform distribution. The point sets are used to compute accuracy, completion, accuracy ratio, and comple-

tion ratio as

$$\text{Acc}(\mathcal{G}, \mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{\mathbf{y} \in \mathcal{E}} \min_{\mathbf{x} \in \mathcal{G}} \|\mathbf{y} - \mathbf{x}\| \quad (2)$$

$$\text{Comp}(\mathcal{G}, \mathcal{E}) = \frac{1}{|\mathcal{G}|} \sum_{\mathbf{x} \in \mathcal{G}} \min_{\mathbf{y} \in \mathcal{E}} \|\mathbf{x} - \mathbf{y}\| \quad (3)$$

$$\text{AR}(\mathcal{G}, \mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{\mathbf{y} \in \mathcal{E}} \left[\min_{\mathbf{x} \in \mathcal{G}} \|\mathbf{y} - \mathbf{x}\| < \Delta \right] \quad (4)$$

$$\text{CR}(\mathcal{G}, \mathcal{E}) = \frac{1}{|\mathcal{G}|} \sum_{\mathbf{x} \in \mathcal{G}} \left[\min_{\mathbf{y} \in \mathcal{E}} \|\mathbf{x} - \mathbf{y}\| < \Delta \right], \quad (5)$$

where $[\cdot]$ denotes the Iverson bracket and $\Delta = 5$ cm in our experiments. Since accuracy ratio and completion ratio can be interpreted as precision and recall of the reconstruction, we further use the F1-score

$$F_1(\mathcal{G}, \mathcal{E}) = \frac{2}{\text{AR}(\mathcal{G}, \mathcal{E})^{-1} + \text{CR}(\mathcal{G}, \mathcal{E})^{-1}} \quad (6)$$

to summarize reconstruction performance in one metric.

10. Additional Results

Detailed Results on Replica and NRGBD Table 4 and Table 5 shows per-scene results on the Replica and NRGBD dataset, respectively. Overall, Co-SLAM achieves the best result, albeit with small margins; it fails on `room1` leading to worse average results. Loopy-SLAM achieves near perfect accuracy at the cost of worse completion. As noted in the paper it uses the ground-truth depth for rendering and

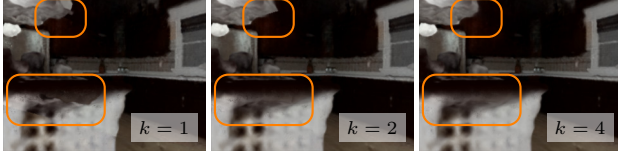


Figure 11. Comparison of rendered views with different number of neighbors k . By increasing the number of nearest neighbors taken into account, the number of visible field transitions decreases. These transitions are most visible in unobserved regions, such as the underside of the countertop shown here (see highlighted regions).

mesh extraction and is hence not fairly comparable to the other methods. Overall, this experiment highlights that our contributions aiming for improved loop closure integration do not significantly worsen performance on small scenes.

Additional Qualitative Results In Figs. 12 to 16 additional qualitative results on the Replica, NRGBD, and ScanNet datasets are shown. In most scenes our approach performs close to the best performing method Co-SLAM. Co-SLAM achieves slightly more detailed and smoother results, which might be due to their more effective hash table use and the additional smoothness loss. Loopy-SLAM sometimes produces meshes with wrong colors; MIPS-Fusion shows transition and streaking artifacts; and GO-SLAM produces more noisy results particularly in terms of appearance.

Our method performs most consistent compared to other methods that support loop closure.

K-Nearest Neighbor Averaging Figure 11 shows renderings of our model with varying values for k (see Sec. 8.2). For higher k visible transitions between fields are reduced and smoothed out. Note that due to our independent training scheme, all fields are trained in overlapping regions making averaging at the query point a viable strategy. While higher values for k lead to improved results, it also multiplies the number of queries required for rendering and mesh extraction (note that optimization time is unaffected by k).

References

- [1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *CVPR*, pages 6290–6301, 2022. 18, 19
- [2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 20
- [3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 12
- [4] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R Oswald. Loopy-SLAM: Dense neural SLAM with loop closures. In *CVPR*, pages 20363–20373, 2024. 10, 14, 15, 16, 17, 18, 19, 20
- [5] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM TOG*, 38(5):1–19, 2019. 11
- [6] Julian Straub et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 16, 17
- [7] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, 2021. 16, 17
- [8] Yijie Tang, Jiazhao Zhang, Zhinan Yu, He Wang, and Kai Xu. MIPS-FUSION: Multi-implicit-submaps for scalable and robust online neural RGB-D reconstruction. *ACM TOG*, 42(6):1–16, 2023. 14, 15, 16, 17, 18, 19, 20
- [9] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM. In *CVPR*, pages 13293–13302, 2023. 11, 12, 14, 15, 16, 17, 18, 19, 20
- [10] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. GO-SLAM: Global optimization for consistent 3D instant reconstruction. In *ICCV*, pages 3727–3737, 2023. 10, 14, 15, 16, 17, 18, 19, 20
- [11] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for SLAM. In *CVPR*, 2022. 14, 15, 16, 17, 18, 19, 20

Table 4. Comparison of mesh quality on Replica (**best** 🟡, second best 🟢, third best 🟠).

		offi0	offi1	offi2	offi3	offi4	room0	room1	room2	Avg.
NICE-SLAM [11]	Acc (cm)	1.90	1.61 🟢	3.13	2.92 🟠	2.60	2.47	2.21 🟢	2.17 🟢	2.38 🟠
	Acc.-Ratio (%)	94.87	95.30	89.78	90.17 🟠	93.21 🟡	93.38	94.92 🟢	93.75 🟡	93.17 🟡
	Comp. (cm)	2.37	2.15	2.89	3.42	3.91	2.93	2.31	2.77 🟠	2.84
	Comp. Ratio (%)	92.58	92.15	88.78	86.20	85.72	90.90	93.57 🟡	90.97 🟢	90.11
	F1-Score (%)	93.71	93.70 🟠	89.28	88.14	89.31	92.12	94.24 🟡	92.34 🟢	91.60
Co-SLAM [9]	Acc (cm)	1.55 🟡	1.33 🟡	2.76 🟠	2.61 🟡	2.22 🟡	1.99 🟡	19.90	1.92 🟡	4.28
	Acc.-Ratio (%)	96.15	96.75 🟢	90.92	92.04 🟡	92.70	95.37 🟡	38.68	93.51 🟢	87.02
	Comp. (cm)	1.54 🟡	1.68 🟡	2.39 🟡	2.73 🟡	2.47 🟡	2.37 🟡	17.47	2.08 🟡	4.09
	Comp. Ratio (%)	96.04 🟡	94.57 🟡	91.99 🟡	90.92 🟡	90.96 🟡	93.43 🟡	40.03	93.16 🟡	86.39
	F1-Score (%)	96.09 🟡	95.65 🟡	91.46 🟢	91.48 🟡	91.82 🟡	94.39 🟡	39.34	93.34 🟡	86.70
GO-SLAM [10]	Acc (cm)	1.88	1.72	3.51	3.97	3.56	3.45	2.15 🟡	2.90	2.89
	Acc.-Ratio (%)	96.31 🟡	97.55 🟡	84.49	79.64	86.34	84.59	96.04 🟡	89.50	89.31
	Comp. (cm)	3.12	4.17	6.74	7.43	8.39	6.83	4.26	8.46	6.17
	Comp. Ratio (%)	85.21	82.89	69.40	63.81	67.33	69.44	82.89	72.65	74.20
	F1-Score (%)	90.42	89.63	76.20	70.85	75.66	76.27	88.98	80.20	81.03
MIPS-Fusion [8]	Acc (cm)	1.65	1.61 🟠	3.16	3.43	2.62	2.30	2.64	2.52 🟠	2.49
	Acc.-Ratio (%)	96.16 🟠	95.41 🟠	91.21 🟠	88.96	89.99	94.95 🟢	93.76	90.60 🟠	92.63
	Comp. (cm)	1.78	1.86 🟢	2.99	3.22	2.67 🟢	2.78	2.13 🟡	2.69 🟢	2.52 🟢
	Comp. Ratio (%)	94.96	93.09 🟢	89.16	88.40	89.64 🟢	92.37 🟢	93.48 🟢	89.92 🟢	91.38 🟡
	F1-Score (%)	95.56 🟠	94.24 🟢	90.17	88.68	89.82	93.64 🟢	93.62 🟢	90.26 🟠	92.00 🟢
Loopy-SLAM [4]	Acc (cm)	1.05	0.84	1.33	1.53	1.49	1.45	1.13	1.20	1.25
	Acc.-Ratio (%)	100.00	100.00	100.00	99.98	99.99	99.99	100.00	99.99	99.99
	Comp. (cm)	1.65	2.16	3.64	3.03	3.80	3.45	2.95	2.65	2.92
	Comp. Ratio (%)	93.43	90.80	86.33	88.67	86.32	88.58	89.78	90.32	89.28
	F1-Score (%)	96.61	95.18	92.66	93.98	92.65	93.94	94.62	94.91	94.32
Ours-SF	Acc (cm)	1.55 🟢	1.79	2.49 🟢	3.09	2.38 🟠	2.12 🟢	2.52 🟠	2.53	2.31 🟡
	Acc.-Ratio (%)	96.25 🟢	94.20	92.51 🟢	89.94	92.83 🟢	94.51 🟠	94.59 🟠	90.08	93.11 🟢
	Comp. (cm)	1.59 🟢	2.01	2.66 🟢	2.94 🟠	2.89 🟠	2.46 🟢	2.23 🟠	3.16	2.49 🟡
	Comp. Ratio (%)	95.88 🟢	92.32 🟠	90.65 🟢	89.31 🟢	88.24 🟠	91.92 🟠	93.20	88.75	91.28 🟢
	F1-Score (%)	96.07 🟢	93.25	91.57 🟡	89.63 🟠	90.47 🟢	93.20 🟠	93.89 🟢	89.41	92.19 🟡
Ours	Acc (cm)	1.60 🟠	1.86	2.46 🟡	2.81 🟢	2.36 🟢	2.14 🟠	2.73	2.85	2.35 🟢
	Acc.-Ratio (%)	95.74	94.20	92.52 🟡	90.53 🟢	92.81 🟠	94.46	92.55	89.26	92.76 🟠
	Comp. (cm)	1.67 🟠	2.00 🟠	2.76 🟠	2.78 🟢	3.01	2.71 🟠	2.16 🟢	4.00	2.64 🟠
	Comp. Ratio (%)	95.27 🟠	91.74	89.65 🟢	89.01 🟠	88.02	91.88	93.41 🟠	87.96	90.87 🟠
	F1-Score (%)	95.50	92.95	91.06 🟠	89.76 🟢	90.35 🟠	93.15	92.98	88.61	91.80 🟠

Table 5. Comparison of mesh quality on NRGBD (best 🟡, second best ⚪, third best 🟠).

		br	ck	gr	gwr	ki	ma	sc	tg	wa	Avg.
NICE-SLAM [11]	Acc (cm)	2.46	10.76	2.33	2.71	9.18	1.70 ⚪	4.55	8.37	7.37	5.49
	Acc.-Ratio (%)	92.41	65.34	93.87	93.63	57.29	95.06 ⚪	71.69	56.22	77.38	78.10
	Comp. (cm)	4.82	14.21	3.91	3.19	12.82	3.36	10.69	8.02	5.39	7.38
	Comp. Ratio (%)	86.18	53.58	86.66	87.64	50.04	84.06 🟠	58.79	59.24	69.04	70.58
	F1-Score (%)	89.19	58.88	90.12	90.54	53.42	89.22 ⚪	64.60	57.69	72.97	74.07
Co-SLAM [9]	Acc (cm)	2.21 🟠	4.73	1.89 🟠	2.02 🟠	7.40	1.74	3.30 🟠	2.07 🟠	6.24	3.51
	Acc.-Ratio (%)	93.24 🟠	75.16	95.17 ⚪	94.84	77.72	93.98	78.01 🟠	92.05 🟠	84.92	87.23
	Comp. (cm)	2.06 🟡	8.76 ⚪	2.93 🟠	2.41 🟡	5.14 ⚪	2.75 🟡	4.29 ⚪	2.83 🟡	3.85 ⚪	3.89 🟡
	Comp. Ratio (%)	93.49 🟠	63.17 🟠	91.32 🟡	93.96 🟡	78.19 🟡	86.41 🟡	70.90 🟠	86.62 🟡	81.70 ⚪	82.86 🟠
	F1-Score (%)	93.37 🟠	68.65 🟠	93.21 🟡	94.40 🟡	77.96	90.03 🟡	74.29 🟠	89.26 🟡	83.28 🟠	84.94 🟠
GO-SLAM [10]	Acc (cm)	3.89	4.08	2.50	2.87	3.28 🟠	1.54 🟡	6.46	1.48 🟡	5.46	3.51
	Acc.-Ratio (%)	77.64	81.94	91.46	86.87	84.74 🟠	97.44 🟡	66.62	96.32 🟡	73.90	84.10
	Comp. (cm)	9.25	29.60	9.50	4.50	5.11 🟡	4.60	12.35	7.26	12.57	10.53
	Comp. Ratio (%)	64.29	54.56	71.31	75.12	72.58	75.53	54.24	70.78	57.33	66.19
	F1-Score (%)	70.34	65.51	80.14	80.57	78.19 🟠	85.10	59.80	81.59	64.57	73.98
MIPS-Fusion [8]	Acc (cm)	2.44	3.48 ⚪	1.94	2.10	9.24	1.72 🟠	3.49	1.61 ⚪	3.30 🟠	3.26 🟠
	Acc.-Ratio (%)	91.20	84.69 🟠	94.53	95.76 🟠	75.62	93.68	75.92	93.59 ⚪	89.61 🟠	88.29 🟠
	Comp. (cm)	5.05	35.03	8.33	3.07	8.91 🟠	3.15 🟠	8.63	3.61 🟠	11.52	9.70
	Comp. Ratio (%)	83.49	55.36	83.09	89.38	70.86	82.81	64.98	79.77	72.89	75.85
	F1-Score (%)	87.18	66.95	88.44	92.46	73.16	87.91	70.02	86.13 ⚪	80.39	81.40
Loopy-SLAM [4]	Acc (cm)	1.44	3.41	1.93	2.00	20.41	1.14	2.59	×	3.02	4.49
	Acc.-Ratio (%)	99.85	85.01	98.99	98.52	61.61	99.93	94.86	×	85.87	90.58
	Comp. (cm)	3.24	13.57	3.41	3.14	8.31	3.17	3.39	×	4.40	5.33
	Comp. Ratio (%)	90.29	71.09	88.58	89.86	68.23	83.98	85.03	×	76.88	81.74
	F1-Score (%)	94.83	77.43	93.50	93.99	64.75	91.26	89.67	×	81.13	90.58
Ours-SF	Acc (cm)	1.68 🟡	3.57 🟠	1.82 ⚪	1.95 ⚪	2.28 ⚪	1.79	2.33 🟡	2.55	3.10 ⚪	2.34 ⚪
	Acc.-Ratio (%)	94.57 🟡	86.96 ⚪	95.12 🟠	96.34 🟡	95.24 ⚪	93.90	95.66 🟡	89.75	90.47 ⚪	93.11 🟡
	Comp. (cm)	2.16 ⚪	7.49 🟡	2.86 🟡	2.64 ⚪	25.86	3.10 ⚪	3.84 🟡	3.79	3.73 🟡	6.16 ⚪
	Comp. Ratio (%)	94.87 🟡	78.60 🟡	90.95 ⚪	91.76 ⚪	74.19 ⚪	84.16 ⚪	88.77 🟡	82.27 🟠	83.45 🟡	85.45 🟡
	F1-Score (%)	94.72 🟡	82.57 🟡	92.99 🟠	93.99 ⚪	83.41 🟡	88.76 🟠	92.09 🟡	85.85 🟠	86.82 🟡	89.02 🟡
Ours	Acc (cm)	1.77 ⚪	3.19 🟡	1.78 🟡	1.94 🟡	2.18 🟡	1.78	2.42 ⚪	2.65	2.92 🟡	2.29 🟡
	Acc.-Ratio (%)	93.83 ⚪	88.67 🟡	95.50 🟡	96.30 ⚪	96.00 🟡	94.24 🟠	94.95 ⚪	87.40	90.90 🟡	93.09 ⚪
	Comp. (cm)	2.46 🟠	8.98 🟠	2.88 ⚪	2.67 🟠	29.26	3.25	4.33 🟠	3.33 ⚪	4.18 🟠	6.82 🟠
	Comp. Ratio (%)	93.84 ⚪	76.89 ⚪	90.83 🟠	91.31 🟠	73.69 🟠	82.80	87.90 ⚪	83.53 ⚪	81.51 🟠	84.70 🟠
	F1-Score (%)	93.83 ⚪	82.36 ⚪	93.11 ⚪	93.74 🟠	83.38 ⚪	88.15	91.29 ⚪	85.42	85.95 ⚪	88.58 ⚪



Figure 12. Qualitative comparison of final meshes extracted by all methods on the Replica dataset [6, 7] (part 1).



Figure 13. Qualitative comparison of final meshes extracted by all methods on the Replica dataset [6, 7] (part 2).

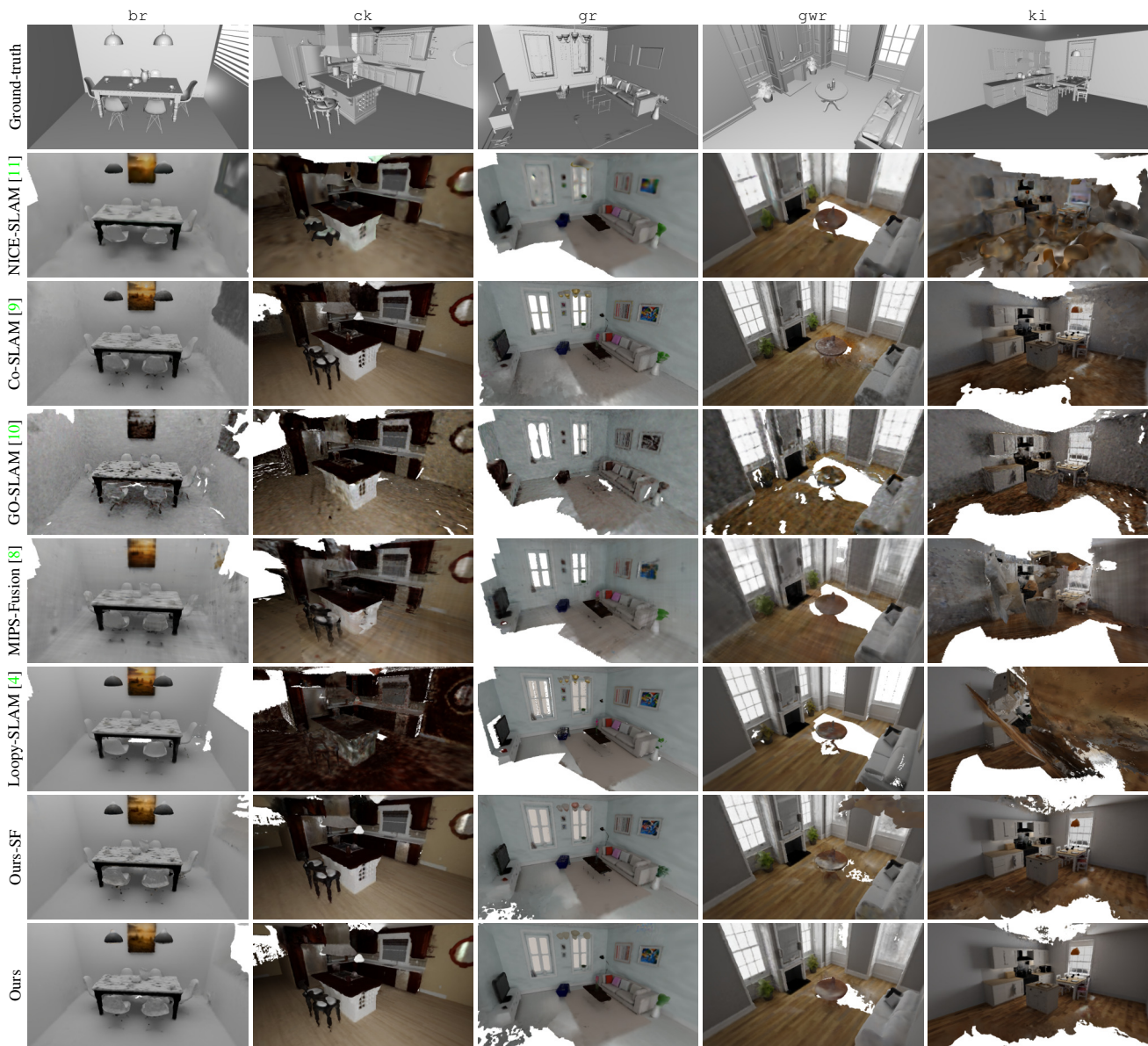


Figure 14. Qualitative comparison of final meshes extracted by all evaluated methods on the NRGBD dataset [1] (part 1).

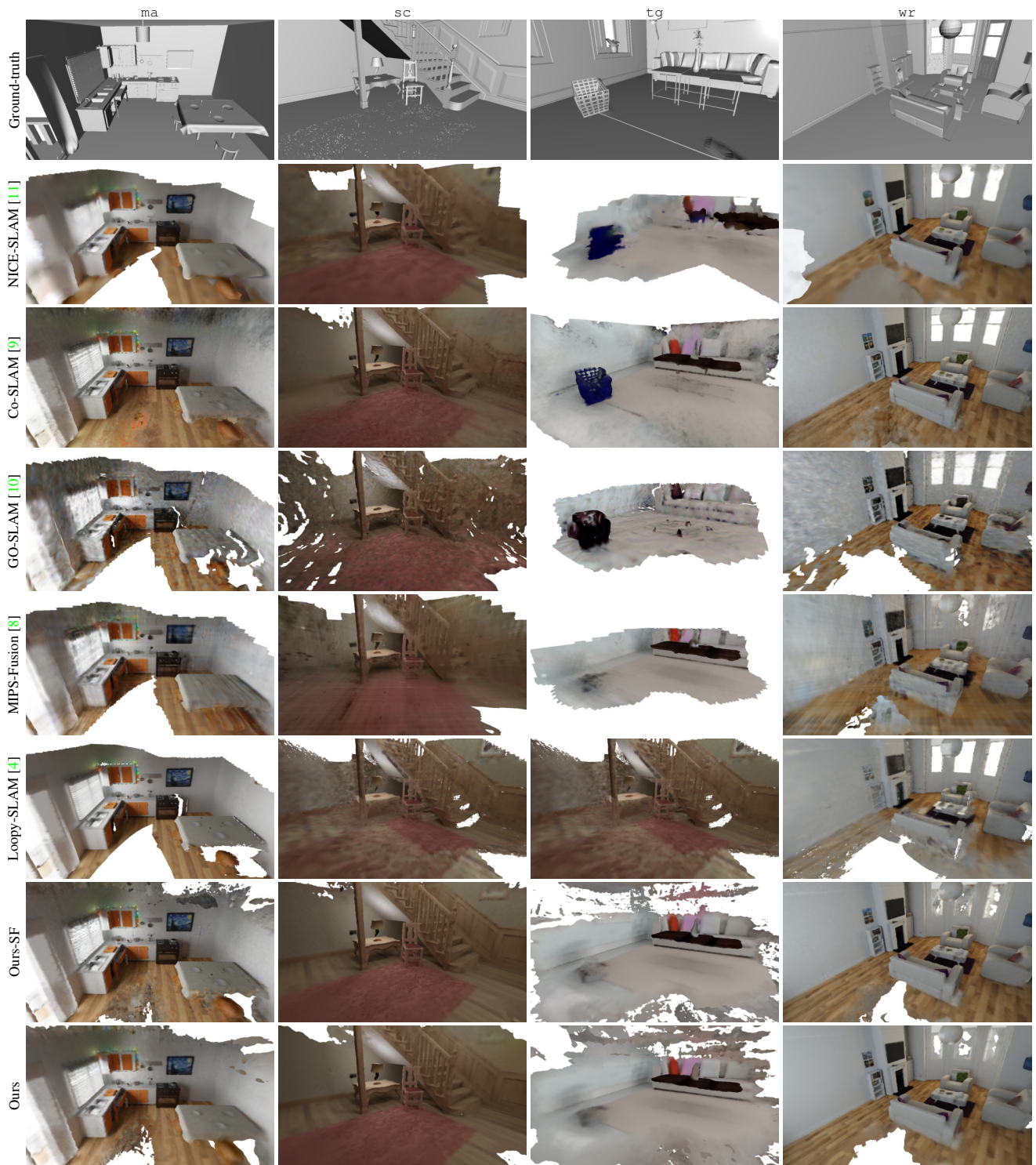


Figure 15. Qualitative comparison of final meshes extracted by all evaluated methods on the NRGBD dataset [1] (part 2).



Figure 16. Qualitative comparison of final meshes extracted by all methods on the ScanNet dataset [2]. Our method fails on scene0181_00 due to tracking issues in the underlying SLAM system in a feature-less region.