Supplementary Materials for ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields

Mohammad Mahdi Johari
Idiap Research Institute, EPFL
mohammad.johari@idiap.ch

Camilla Carta
ams OSRAM
camilla.carta@ams-osram.com

François Fleuret
University of Geneva, EPFL
francois.fleuret@unige.ch

1. Further Implementation Details

This section provides additional implementation details of our method. For the sake of completeness, we also reiterate the points mentioned in the main article.

The truncation distance T is set to 6 cm in our method. We employ coarse feature planes with a resolution of 24 cm for both geometry and appearance. For fine feature planes, we use a resolution of 6 cm for geometry and 3 cm for appearance. All feature planes have 32 channels, resulting in a 64-channel concatenated feature input for the decoders. The decoders are two-layer MLPs with 32 channels in the hidden layer. ReLU activation function is used for the hidden layer, and Tanh and Sigmoid are respectively used for the output layers of TSDF and raw colors.

We use different set of loss coefficients for mapping and tracking. During mapping we set $\lambda_{fs} = 5$, $\lambda_{T\text{-}m} = 200$, $\lambda_{T\text{-}t} = 10$, $\lambda_d = 0.1$, and $\lambda_c = 5$. And during tracking, we set $\lambda_{fs} = 10$, $\lambda_{T\text{-}m} = 200$, $\lambda_{T\text{-}t} = 50$, $\lambda_d = 1$, and $\lambda_c = 5$. These coefficients are obtained by performing grid search in our experiments.

For the scenes from Replica [5], we sample $N_{strat} = 32$ points for stratified sampling and $N_{imp} = 8$ points for importance sampling on each ray. We perform 15 optimization iterations for mapping and randomly select 4000 rays for each iteration. For camera tracking, 2000 rays are chosen at random and 8 optimization iterations are performed. Since ScanNet's [1] scenes are at a larger scale and more challenging, we set $N_{strat} = 48$ and $N_{imp} = 8$. Also, we perform 30 optimization iterations for both mapping and tracking in ScanNet's [1] scenes. For the scenes in TUM RGB-D dataset [6], we similarly set $N_{strat} = 48$ and $N_{imp} = 8$. For this dataset, We perform 60 optimization iterations for mapping and 200 optimization iterations for tracking, and we randomly sample 5000 rays for each iteration.

We initiate the mapping process every 4 input frames and use a window of $W = 20$ keyframes for jointly optimizing the feature planes, MLP decoders, and camera poses of the selected keyframes. We use Adam [2] for optimizing all learnable parameters of our method and set the learning rates according to a simple grid search in our experiments. We use a learning rate of 0.001 for the MLP decoders and a learning rate of 0.005 for the feature planes. We always use a learning rate of 0.001 for the camera poses, *i.e.* rotation and translation $\{R, t\}$, of the selected keyframes during the joint optimization of the mapping step. During the tracking step in the Replica's [5] scenes, we use a learning rate of 0.001 for camera rotation and translation. For camera tracking in the scenes of ScanNet [1], we use a learning rate of 0.0005 for camera translation and a learning rate of 0.0025 for camera rotation. Lastly, For camera tracking in the scenes of TUM RGB-D [6], we use a learning rate of 0.01 for camera translation and a learning rate of 0.002 for camera rotation. We model the camera rotation parameters with quaternions [4].

Once all input frames are processed, and for evaluation purposes, we build a TSDF volume for each scene and use the marching cubes algorithm [3] to obtain 3D meshes. We do not employ any post-processing for our representation or extracted meshes except that for quantitative evaluation, we cull faces from a mesh that are not inside any camera frustum or are occluded in all RGB-D frames. To ensure fairness, we do the same mesh culling before evaluating the previous approaches.

2. Ablation Study

In this section, we conduct various experiments to show the robustness of our method in different experimental settings and to validate our architecture design choices.

Robustness to Depth Quality. In this experiment, we evaluate how robust the methods are to the quality of input depths. Accordingly, we downsample input depths of `room0` of the replica dataset [5] to $\frac{1}{8}$ of the original resolution. The results in Tab. 1 reveal that our method's reconstruction and localization are less sensitive to the resolution of input depth maps.

	Method	ATE↓	Acc.↓	Comp.↓
$\frac{1}{1}$ D	NICE-SLAM [8]	1.69	1.71	1.69
	ESLAM (ours)	**0.71**	**1.07**	**1.12**
$\frac{1}{8}$ D	NICE-SLAM [8]	2.01	2.18	1.98
	ESLAM (ours)	**0.72**	**1.16**	**1.23**

Table 1. Robustness to depth resolution comparison of our method with NICE-SLAM [8] in terms of ATE RMSE (cm), reconstruction accuracy (cm), and reconstruction completion (cm) on `room0` of the Replica [5] dataset. Our method's accuracy is less affected when input depth is downsampled by a factor of $\frac{1}{8}$.

Method	ATE↓	Acc.↓	Comp.↓
NICE-SLAM [8]	1.69	1.71	1.69
NICE-SLAM w/ Our Key. Policy	1.65	1.68	1.66

Table 2. Analysis of the impact of our keyframe updating policy on NICE-SLAM [8] (Sec. 3.4 in the main paper). The experiment is conducted on `room0` of Replica [5], and the metrics are ATE RMSE (cm), reconstruction accuracy (cm), and reconstruction completion (cm). NICE-SLAM [8] only slightly benefits from our updating policy.

Keyframe Policy. Whenever we perform a mapping step for an input frame, we always include that frame in our global keyframe list (see Sec. 3.4 in the main paper). NICE-SLAM [8], on the other hand, only updates its keyframe list once per 10 mapping steps. To make sure that our evaluations are fair, we also run NICE-SLAM [8] with our own keyframe updating policy on `room0` of the Replica [5] dataset. The results in Tab. 2 show that NICE-SLAM [8] only slightly benefits from this updating policy.

Our Design Choices. We conduct multiple experiments in Tab. 3 to defend our design choices in ESLAM. These experiments are conducted on the scenes in the Replica [5] and ScanNet [1] datasets, and the details of the experimental settings are as follows. (a) We use shared feature planes for geometry and appearance (see Sec. 3.1 and Fig. 2 in the main paper). (b) We only employ coarse feature planes (see Sec. 3.1 and Fig. 2 in the main paper). (c) We only employ fine feature planes (see Sec. 3.1 and Fig. 2 in the main paper). (d) We add the interpolated coarse $f^c_*(p_n)$ and fine $f^f_*(p_n)$ features instead of concatenating them (see Sec. 3.1 and Fig. 2 in the main paper). (e) We discard importance sampling and use stratified sampling for all N points on a ray (see Sec. 3.2 in the main paper). (f) We only exploit depth inputs and discard color rendering and RGB inputs (see Sec. 3.2 in the main paper). (g) We do not consider separate loss functions for the points that are at the tail of the truncation region $P_r^{T\text{-}t}$ and for the points that are in the middle $P_r^{T\text{-}m}$ (see Sec. 3.3 in the main paper). (h) We do not jointly optimize camera poses during the mapping step (see Sec. 3.4 in the main paper). (i) We evaluate our full model. Note that due to the incompleteness of ScanNet's [1] ground truth meshes, we only evaluate localization accuracy on this dataset.

3. Additional Qualitative Analysis

This section provides additional qualitative analysis to contrast the capability of our method to preserve scene details in comparison to previous NeRF-based dense visual SLAM methods, iMAP* [7] and NICE-SLAM [8]. We provide this analysis on the Replica dataset [5] in Fig. 1 with both textured and untextured meshes. The results demonstrate that our method produces more accurate meshes with fewer artifacts.

4. Per-Scene Breakdown of the Results

In this section, we breakdown the quantitative analysis of Tab. 1 in the main paper into a per-scene analysis. Tab. 4 shows the per-scene quantitative evaluation of our method in comparison with iMAP* [7] and NICE-SLAM [8] on the Replica dataset [5]. As it is shown in Tab. 4, our method outperforms previous approaches in all scenes of Replica [5]. Also, lower variances in our experiments are an indication that our method is more stable from run to run.

5. Effect of Frame Processing Time

In this section, we investigate the trade-off between frame processing time and our method's reconstruction and localization accuracy. In this study, we increase the number of optimization iterations during the mapping and tracking. By default, our ESLAM method performs $Iter_m = 15$ optimization iterations during mapping and $Iter_t = 8$ optimization iterations during tracking for the scenes of the Replica dataset [5]. We define ESLAM x2 as our method when we double the number of optimization iterations, *i.e.* $Iter_m = 30$ and $Iter_t = 16$. And similarly, we define ESLAM x10 as our method with $Iter_m = 150$ and $Iter_t = 80$.

Tab. 5 provides a quantitative analysis of ESLAM x2 and ESLAM x10, as well as a comparison with our default ESLAM method and existing approaches. The results show that at the cost of increased frame processing time, our method yields more accurate scene reconstruction and camera trajectory. It should be noted that even ESLAM x10 runs faster than the existing state-of-the-art method, NICE-SLAM [8].

Fig. 2 provides a qualitative analysis of ESLAM x10 compared to our default ESLAM method. In this analysis, we render the scenes with untextured meshes to contrast the quality of geometry reconstruction. While the quality difference is subtle, Fig. 2 indicates that increasing the number of optimization iterations results in more accurate geometry reconstruction and smoother surfaces.

Experiment	ScanNet [1]	Replica [5]		
	ATE↓	ATE↓	Accuracy↓	Compeletion↓
a. Using shared feature planes for geometry and appearance.	7.49	0.65	0.99	1.08
b. Using only the coarse planes and discarding the fine ones.	7.53	0.97	1.12	1.29
c. Using only the fine planes and discarding the coarse ones.	8.27	0.72	1.00	1.09
d. Replacing the concatenation with a summation.	7.55	0.64	0.98	1.07
e. No importance sampling.	7.44	0.67	1.08	1.14
f. No color rendering.	8.31	0.68	1.03	1.08
g. One loss function for the whole truncation region.	8.28	0.71	1.01	1.10
h. No camera pose optimization during mapping.	11.27	4.85	2.23	2.21
i. Full ESLAM method.	**7.38**	**0.63**	**0.97**	**1.05**

Table 3. Ablation study of our design choices on the ScanNet [1] and Replica [5] datasets. The metrics are ATE RMSE (cm), reconstruction accuracy (cm), and reconstruction completion (cm). For the details of this study, see Sec. 2.

	Methods	Reconstruction (cm)				Localization (cm)	
		Depth L1↓	Acc.↓	Comp.↓	Comp. Ratio (%)↑	ATE Mean↓	ATE RMSE↓
room0	iMAP* [7]	6.56 ± 0.39	5.89 ± 0.19	6.07 ± 0.22	66.55 ± 1.58	3.12 ± 0.84	5.23 ± 1.41
	NICE-SLAM [8]	2.77 ± 0.13	1.71 ± 0.03	1.69 ± 0.03	97.61 ± 0.09	1.43 ± 0.09	1.69 ± 0.17
	ESLAM (Ours)	**0.97** ± **0.04**	**1.07** ± **0.01**	**1.12** ± **0.01**	**99.06** ± **0.05**	**0.61** ± **0.06**	**0.71** ± **0.13**
room1	iMAP* [7]	5.97 ± 1.14	5.71 ± 0.31	5.57 ± 0.40	66.04 ± 3.45	2.54 ± 0.37	3.09 ± 0.48
	NICE-SLAM [8]	2.52 ± 0.11	1.36 ± 0.03	1.34 ± 0.04	98.60 ± 0.14	1.70 ± 0.29	2.13 ± 0.24
	ESLAM (Ours)	**1.07** ± **0.07**	**0.85** ± **0.01**	**0.88** ± **0.01**	**99.64** ± **0.06**	**0.56** ± **0.02**	**0.70** ± **0.02**
room2	iMAP* [7]	7.82 ± 0.94	6.34 ± 0.32	5.47 ± 0.27	69.87 ± 4.15	2.31 ± 0.20	2.58 ± 0.19
	NICE-SLAM [8]	3.54 ± 0.35	1.75 ± 0.06	1.71 ± 0.03	96.52 ± 0.26	1.41 ± 0.24	1.87 ± 0.39
	ESLAM (Ours)	**1.28** ± **0.07**	**0.93** ± **0.01**	**1.05** ± **0.01**	**98.84** ± **0.06**	**0.43** ± **0.01**	**0.52** ± **0.01**
office0	iMAP* [7]	7.57 ± 0.70	7.44 ± 0.26	5.13 ± 0.37	70.97 ± 3.52	1.69 ± 1.06	2.40 ± 1.05
	NICE-SLAM [8]	2.17 ± 0.14	1.43 ± 0.06	1.56 ± 0.05	96.30 ± 0.33	1.12 ± 0.22	1.26 ± 0.24
	ESLAM (Ours)	**0.86** ± **0.02**	**0.85** ± **0.01**	**0.96** ± **0.01**	**98.34** ± **0.05**	**0.42** ± **0.03**	**0.57** ± **0.04**
office1	iMAP* [7]	8.91 ± 0.65	10.34 ± 0.15	5.58 ± 0.24	72.08 ± 3.21	1.03 ± 0.17	1.17 ± 0.25
	NICE-SLAM [8]	2.41 ± 0.11	1.16 ± 0.07	1.15 ± 0.03	98.04 ± 0.19	0.74 ± 0.19	0.84 ± 0.17
	ESLAM (Ours)	**1.26** ± **0.02**	**0.83** ± **0.06**	**0.81** ± **0.01**	**98.85** ± **0.08**	**0.46** ± **0.05**	**0.55** ± **0.04**
office2	iMAP* [7]	11.04 ± 0.69	9.15 ± 0.39	6.27 ± 0.37	62.24 ± 2.62	3.99 ± 0.98	5.67 ± 1.82
	NICE-SLAM [8]	4.96 ± 0.58	1.83 ± 0.07	1.72 ± 0.03	96.96 ± 0.25	1.42 ± 0.10	1.71 ± 0.14
	ESLAM (Ours)	**1.71** ± **0.07**	**1.02** ± **0.01**	**1.09** ± **0.01**	**98.60** ± **0.12**	**0.47** ± **0.03**	**0.58** ± **0.09**
office3	iMAP* [7]	10.12 ± 1.31	7.14 ± 0.27	6.02 ± 0.20	66.07 ± 1.65	4.05 ± 0.93	5.08 ± 1.37
	NICE-SLAM [8]	4.91 ± 0.70	2.24 ± 0.17	2.17 ± 0.05	93.08 ± 0.40	2.31 ± 0.51	3.98 ± 1.79
	ESLAM (Ours)	**1.43** ± **0.05**	**1.21** ± **0.01**	**1.42** ± **0.01**	**96.80** ± **0.03**	**0.61** ± **0.03**	**0.72** ± **0.02**
office4	iMAP* [7]	7.85 ± 1.32	5.32 ± 0.18	6.51 ± 0.20	63.63 ± 1.39	1.93 ± 0.21	2.23 ± 0.35
	NICE-SLAM [8]	3.81 ± 0.74	2.09 ± 0.16	2.03 ± 0.17	95.00 ± 1.31	2.22 ± 0.68	2.82 ± 0.71
	ESLAM (Ours)	**1.06** ± **0.08**	**1.15** ± **0.02**	**1.27** ± **0.01**	**97.65** ± **0.14**	**0.52** ± **0.02**	**0.63** ± **0.03**
Average	iMAP* [7]	8.23 ± 0.88	7.16 ± 0.26	5.83 ± 0.27	67.17 ± 2.70	2.59 ± 0.58	3.42 ± 0.87
	NICE-SLAM [8]	3.29 ± 0.33	1.66 ± 0.07	1.63 ± 0.05	96.74 ± 0.36	1.56 ± 0.29	2.05 ± 0.45
	ESLAM (Ours)	**1.18** ± **0.05**	**0.97** ± **0.02**	**1.05** ± **0.01**	**98.60** ± **0.07**	**0.52** ± **0.03**	**0.63** ± **0.05**

Table 4. Per-scene quantitative comparison of our proposed ESLAM with existing NeRF-based dense visual SLAM models on the Replica dataset [5] for both reconstruction and localization accuracy. The results are the average and standard deviation of five independent runs on each scene of the Replica dataset [5]. Our method outperforms previous works by a high margin and has lower variances, indicating it is also more stable from run to run. The evaluation metrics for reconstruction are L1 loss (cm) between rendered and ground truth depth maps of 1000 random camera poses, reconstruction accuracy (cm), reconstruction completion (cm), and completion ratio (%). The evaluation metrics for localization are mean and RMSE of ATE (cm) [6]. It should also be noted that our method runs up to ×10 faster on this dataset (see Sec. 4.2 in the main paper for runtime analysis).

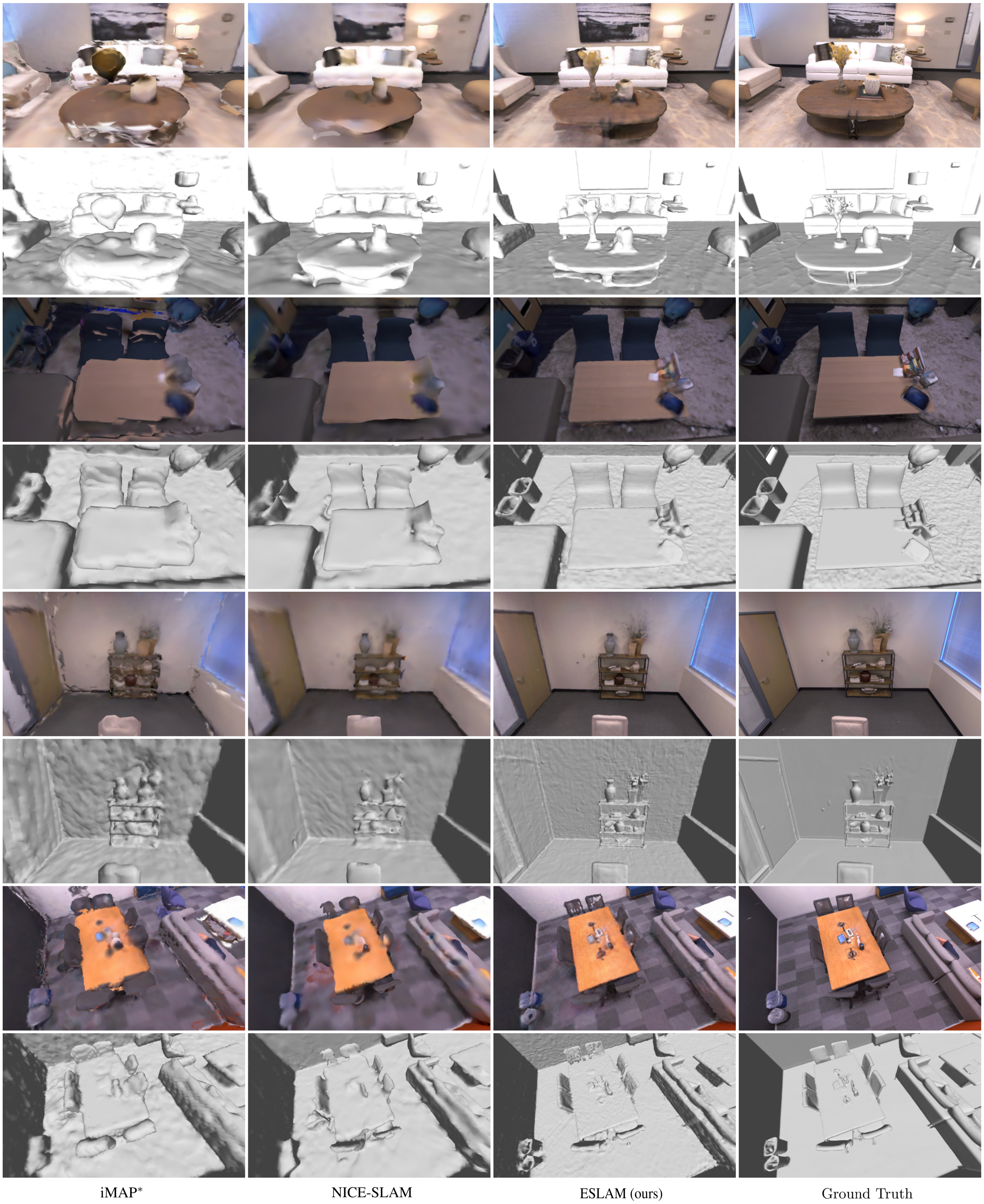

Figure 1. Qualitative comparison of our method's scene reconstruction with iMAP* [7] and NICE-SLAM [8] on Replica [5]. Our method produces more accurate detailed geometry as well as higher-quality textures. The scenes are rendered with both textured and untextured meshes and the ground truth textured images are rendered with the ReplicaViewer software [5]. It should also be noted that our method runs up to ×10 faster on this dataset (see Sec. 4.2 in the main paper for runtime analysis).

Method	Optimization Iterations	Acc. (cm)↓	Comp. (cm)↓	ATE (cm)↓	FPT (s)↓
iMAP* [7]	-	7.16	5.83	3.42	5.20
NICE-SLAM [8]	-	1.66	1.63	2.05	2.10
ESLAM (ours)	$Iter_m = 15, \quad Iter_t = 8$	0.97	1.05	0.63	**0.18**
ESLAM x2 (ours)	$Iter_m = 30, \quad Iter_t = 16$	0.95	1.03	0.42	0.35
ESLAM x10 (ours)	$Iter_m = 150, \quad Iter_t = 80$	**0.92**	**1.01**	**0.31**	1.72

Table 5. Quantitative analysis of the effect of the number of optimization iterations during mapping and tracking on our method's reconstruction and localization accuracy. $Iter_m$ stands for the number of optimization iterations during mapping, and $Iter_t$ denotes the number of optimization iterations during tracking. The evaluation metrics are reconstruction accuracy (cm), reconstruction completion (cm), and ATE RMSE (cm) [6]. Average Frame Processing Time (FTP) is also shown to highlight the trade-off between the accuracy and throughput of our method. For reference, we reiterate the performance of the existing approaches, iMAP* [7] and NICE-SLAM [8]. It should be noted that even ESLAM x10 runs faster than the existing state-of-the-art method, NICE-SLAM [8]. Refer to Sec. 5 for the details of this experiment, and see Fig. 2 for the qualitative analysis.

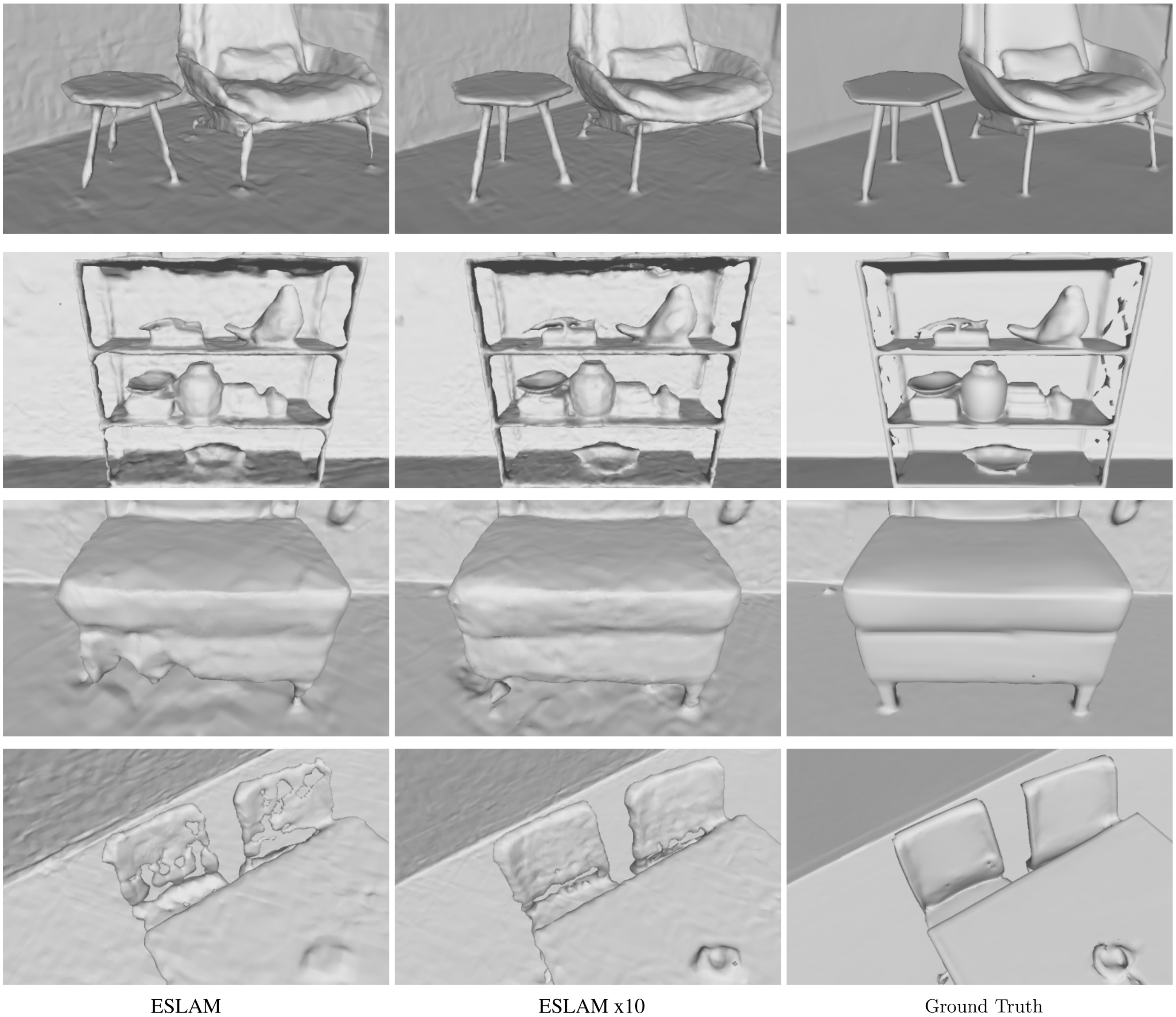

Figure 2. Qualitative analysis of the effect of the number of optimization iterations during mapping and tracking on our method's reconstruction quality. ESLAM x10 is our method when we multiply the number of optimization iterations by 10. Refer to Sec. 5 for the details of this experiment, and see Tab. 5 for the quantitative analysis.

References

[1] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 3

[2] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 1

[3] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 1

[4] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985. 1

[5] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 2, 3, 4

[6] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 1, 3, 5

[7] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 2, 3, 4, 5

[8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 2, 3, 4, 5