Online 3D Scene Reconstruction Using Neural Object Priors

Supplementary Material

The supplementary material is organized as follows: we discuss limitations of our method and possible directions to explore in Sec. 1, provide additional comparisons with state-of-the-art methods both quantitatively and qualitatively in Sec. 2 and introduce more implementation details in Sec. 3.

1. Limitations

We have identified three main limitations of our method. First, our models are highly sensitive to the quality of input masks, which define object bounding boxes extension and may severely worsen reconstructions. Correcting masks online from multiple views and the fitted models should robustify the whole model. Second, our retrieval and registration often struggle when seeing objects from too different viewpoints from those used for the database models. Further research on this part, *e.g*. using stronger image feature models for efficient retrieval [14, 23] and 6D object pose estimation for registration [15, 21], should allow for more systematic object reuse and reduce computations. Third, our method focuses on mapping objects given camera poses provided by an external SLAM system without correcting them. Extending it to a complete object-level SLAM is an interesting future direction.

2. Additional comparisons with the state of the art

Comparison to RO-MAP. We evaluate here our method in the setting of RO-MAP [7]. While the RO-MAP [7] paper does not provide all the details about their evaluations and no code has been released to reproduce results on the Replica dataset, we reproduce the closest setting to their work for fair comparison, following discussion with RO-MAP authors. In this way, we extract meshes for each object using marching cubes [11] on a grid of size 64^3 as detailed in their paper. We evaluate two scenes with the same objects, *i.e*., 23 objects for the scene *room-0* and 14 for *office-1* as shared by RO-MAP authors. We present our quantitative results in Table 1 and qualitative examples of a scene and close-ups on few objects in Fig. 1 and Fig. 2. The meshes presented here for RO-MAP were shared by the authors. Our evaluations show that our models computed without any object prior are more accurate and have a better completion ratio at 1cm than RO-MAP, though the metric of completion distance is not so good. However, as shown in Fig. 1, RO-MAP objects possess numerous artefacts due to their uniform sampling of points along rays and different losses used, which explains their high accuracy error, very low completion distance and high completion ratio at 5cm, though large parts of objects are unseen in the input videos. In contrast, our object models are sampled only close to the surface during fitting, leading to much fewer outliers. Our method also extracts more faithful geometry on contours of objects as shown on Fig. 2. Leveraging object priors significantly improves the completion of objects in our method. While RO-MAP accumulates numerous views before computing a bounding box and restarts optimization from scratch each time these boxes change, our interpolation strategy allows us to reconstruct objects as soon as we observe them, even very partially, and then update their models at each frame without any reinitialization.

	Object prior	Scene	Whole objects			
			Acc. ↓	Comp. ↓	CR 1cm ↑	CR 5cm ↑
RO-MAP† [7]	—	room-0	3.65	0.93	69.3	98.5
		office-1	3.74	1.15	67.9	97.7
Ours	—	room-0	2.04	2.02	73.8	90.0
		office-1	2.34	1.32	74.6	95.3
	3D meshes	room-0	1.31	0.58	86.2	99.9
		office-1	1.27	0.57	86.0	100.0
	Prior video	room-0	2.09	1.75	74.9	91.2
		office-1	2.43	1.11	76.4	96.3

Table 1. Object-level reconstruction performance compared to RO-MAP [7], using the scenes and settings of RO-MAP's evaluations. Our results with object priors rely on full 3D meshes and shapes from previously viewed videos. Retrieval and registration in these cases are ground truth. CR stands for Completion Ratio. Results for RO-MAP are taken from the original paper.

	Object prior	Whole objects			
		Acc. ↓	Comp. ↓	CR 1cm ↑	CR 5cm ↑
vMAP† [10]	—	2.23	1.44	69.2	94.6
vMAP* [10]	—	1.84	2.32	63.6	91.5
Ours	—	1.52	2.58	73.9	91.0
	3D meshes	1.00	0.61	86.4	99.7
	Prior video	1.54	1.62	77.1	93.3

Table 2. Object-level reconstruction performance compared to vMAP [10], reproducing as closely as possible the evaluation setting of vMAP. Our results with object priors rely on full 3D meshes and shapes from previously viewed videos. Retrieval and registration in these cases are ground truth. CR stands for Completion Ratio. Results with † are taken from the original paper and with * are reproduced. The validity of the difference between the published and reproduced results for vMAP was confirmed by the authors of vMAP.

Comparison to vMAP. We further compare our method with vMAP on the same objects, metrics and evaluation pro-

	Objects	Object prior	Seen parts Acc. ↓	Seen parts Comp. ↓	Whole mesh Acc. ↓	Whole mesh Comp. ↓	Whole mesh CR 5cm ↑	Whole mesh CR 1cm ↑
TSDF* [3]	✗	—	0.55	0.41	2.66	4.31	87.0	81.8
iMAP* [20]	✗	—	0.92	0.91	1.89	2.42	90.1	74.7
ESLAM* [9]	✗	—	0.69	0.59	0.71	4.33	86.1	76.3
Point-SLAM* [18]	✗	—	0.67	0.59	0.67	4.76	85.3	79.5
vMAP* [10]	✓	—	1.03	0.91	2.85	2.49	91.2	75.0
Ours	✓	—	0.81	0.72	2.96	2.39	91.5	78.3
	✓	3D meshes	0.82	0.70	2.08	1.71	95.3	86.3
	✓	Prior video	0.80	0.70	2.69	2.25	92.3	79.3

Table 3. Averaged scene-level reconstruction metrics on the Replica dataset [19], focusing on the ground truth mesh parts observable in the input sequence (left) and whole scene (right). Our results with object priors rely on full 3D meshes and shapes from previously viewed videos. CR stands for Completion Ratio. Methods with * are reproduced results from the official code bases.

cedure as used by vMAP [10], to provide fair comparison to their published results and confirm that our updated setup proposed in the main paper does not artificially improve our performances over other baselines. Differing from the evaluations of the main paper, we consider here all objects in each scene, regardless of their size or presence of noise. We still extract object meshes at a 5mm resolution but set a maximum of 256 points per size and subsample 10k points in the GT and reconstructed meshes to compute metrics instead of considering full meshes. Results are presented in Tab. 2. For vMAP, we provide results taken from the original paper as well as reproduced results from the released code. Though these reproduced results differ from the published ones, we confirmed the validity of this difference with the main authors of vMAP. Reconstruction results with our models also confirm the trend observed in the main paper with our other evaluation setup: our models optimized from scratch are again more accurate than vMAP and have a higher completion ratio at 1cm. Their performances are similarly further increased by leveraging object shape priors. We finally present in Fig. 3 few visualizations of object meshes extracted after only 50 frames processed in the input sequence. In that setting, vMAP [10] tends to produce oversmoothed meshes which lack geometric and appearance details, *e.g.*, the texture of wood on the first row or the cushion's colored leaves on the last row. Conversely, our model from scratch recovers more faithful geometry and early texture for these objects. Adding object shape and texture prior further boosts the representations, leading to more complete and better textured models, in particular for our models obtained from a previous video.

Scene-level comparisons. For scene-level comparisons, our baselines are TSDF [3] (or rather its reimplementation [22]) with a grid resolution of 1cm, vMAP [10] and its reimplementation of iMAP [20], ESLAM [9] and Point-SLAM [18]. Again, we run all the compared methods with their released codes and the provided ground-truth camera poses for fair comparison. We extract scene meshes at a resolution of 1cm for all methods using marching cubes [11], and disable any mesh post-processing. Each method is run with 5 different initializations on each environment, and we present results averaged over all runs.

Results on the Replica dataset are presented in Table 3. As we reuse vMAP's background model, our method outperforms vMAP on the scene-level as well, aligning with the observed trend in object-level evaluation. Despite the improvement we achieve in object-level methods, our best object-level method with separate models per object still lags behind the best scene-level methods that consider scenes as a single entity when considering metrics on the seen parts or accuracy on whole meshes. Since the scene metrics depend heavily on the background quality, our background model which oversmoothes surfaces does not capture well details and explains this gap. Using a more accurate background model would benefit our method. However, our models present better completion distance and ratios on whole meshes than scene-level baselines, showing the interest of decomposing scenes in objects and reusing priors. In addition, it is worth noting that all the neural implicit model-based approaches perform worse on seen parts than the traditional TSDF [3], although they showcase advantages on the whole mesh.

We show meshes reconstructed by each of these methods on a Replica scene in Figure 4. The ground truth mesh is displayed here only for the parts that are seen in the input sequence. Methods relying on TSDF representations, *i.e.*, TSDF [3, 22] and Point-SLAM for its mesh extraction [18], are highly accurate for both objects and the background, though they do not reuse any prior information about the scene and are therefore unable to fill unseen parts. iMAP [20] and vMAP [10] extract oversmooth surfaces with some plausible completion for all objects and for the background, but they miss details of all reconstructed objects. ESLAM, which relies on a single tri-plane representation for the whole scene, proposes some completion for unseen parts of objects and for the background, but it misses important details about thin structures like pouf feet or the basket on the ground (left of the image). While our background model has limited ability to capture scene details, we obtain the highest level of details for all objects in the scene while being able to leverage prior knowledge to complete some parts, see for instance the back of poufs in the last image. We believe that further improvements in the background representation should make our method a strong competitor for online scene-level reconstructions.

Additional visualizations on Replica. As textures may prevent the reader from observing the geometry details of a mesh, we provide a textureless version of **??** from the main paper in Fig. 5. These textureless images emphasize on the higher level of details recovered by our method compared

to vMAP [10]. In particular, thin structures, *e.g.*, table and chair feet or handles, are more challenging for vMAP but are correcly recovered by our approach.

Object meshes on ScanNet. We show some reconstructed objects on the ScanNet dataset in Fig. 6, using our method as well as vMAP [10] and a TSDF [3, 22] implemented with object grids of 5mm resolution. All meshes for these visualizations are extracted at a 1cm resolution using marching cubes [11]. Unlike other methods, we provide TSDF with knowledge of each object extent before starting the reconstruction since it is a static representation. Our method, run with no object prior, is able to reconstruct object geometries that are more accurate than vMAP [10] on these real world sequences. However, these sequences have very noisy depth and segmentation masks, resulting in artefacts in TSDF's reconstructions for ScanNet and slightly noisier ones for our method. The update time for TSDF representations also grows significantly with the resolution of the grid and number of objects, making the access to finer reconstructions much more costly than coarse resolutions, unlike our representations which keep a constant computation speed for any object resolution.

Additional results on sequences captured in our laboratory. We show in Figure 7 additional reconstructions on a scene with 3 objects, with views from the front and the back of the objects. As in the main paper, TSDF reconstructs objects with some floating artefacts around their borders. This objects are incomplete for parts unseen in the current video. vMAP outputs more complete but inaccurate object geometry with oversmoothed texture. Conversely, our approach produces more faithful shapes and textures for both seen and unseen parts of objects, in particular thanks to object initializations from a prior video.

Computation time. On Replica's room-0, our average times per frame are 740ms for objects reconstructed from scratch and 1.4s for models reused from the library. On the same hardware, vMAP and iMAP take 420ms per frame, ESLAM takes 445ms, Point-SLAM needs 32s and the scene-level TSDF runs at 18ms per frame. Our implementation is however not yet parallelized, unlike vMAP, which would yield important time savings. Important time savings can be obtained in several ways. First, improving the implementation to parallelize the fitting of all object models instead of optimizing them sequentially should lead to significant speed gains. Second, our retrieval and registration (R&R) currently operate in the same thread as model optimization, which stops at each R&R attempt, and require around 200ms per retrieval and registration attempt. Performing this stage in a separate thread should allow model optimization to run faster. Third, as object models are fitted separately, their optimization can be paused after converging on the currently stored keyframes. This is particularly useful for objects leaving the field of view for a long time, with no new stored keyframe. In this case, fewer objects are optimized, which benefits both computation time and GPU memory. Pausing the optimization is not feasible when considering the scene as a single entity. Note that these times do not include segmentation and mask tracking times, which are assumed to be run separately.

3. Implementation details

3.1. Object-centric model and optimization

Sampling points. At each frame and for each object, we perform 3 successive optimization steps, sampling 9600 rays among 6 keyframes for each step and 14 points per ray, 13 close to the surface and 1 closer to the camera. As explained in Section 3.1 of the main paper, the surface sampling consists in drawing points close to the surface on rays u following a normal distribution centered at the depth measurement $\mathcal{N}(D(u), \sigma)$. We take $3\sigma = 5cm$ for Replica scenes and a larger $\sigma = 10cm$ on ScanNet. The latter was observed to give better reconstructions due to ScanNet's noisy depth measurements and outliers that grow excessively object bounding boxes, representing large empty spaces.

Bounding boxes. We compute our bounding boxes using axis-aligned boxes in the world frame defined for each scene. Such bounding boxes are more efficient to compute than randomly aligned ones, though they may encompass larger empty space. When updating a bounding box, we add a 10% margin to the box extent in order to avoid doing this update too often as parts of objects are discovered at each frame. For each object, we only consider frames for which the object mask contains at least 100 pixels to avoid updating models based on too few observations.

Keyframe criterion. We reuse the same keyframe criterion as vMAP [10], which consists in considering every 25-th frame as a keyframe for objects and every 50-th for the background. We store keyframes in a buffer of up to 20 keyframes for both objects and background. Storing keyframes represents the largest memory usage of our approach, our object feature grids consisting in only around 65k parameters for shape and appearance respectively.

Volume rendering details We provide here more details about the rendering formulas and losses used for the online reconstruction. For each ray u, we compute ray termination

weights $w_{k,i}$ at each point as:

$$w_{k,i} = \hat{o}_{k,i} \prod_{j=1}^{i-1} (1 - \hat{o}_{k,j}), \tag{1}$$

We then render the pixel color $\hat{C}_k$, depth $\hat{D}_k$, mask $\hat{M}_k$ and depth variance $\hat{V}_k$ of object O_k as:

$$\hat{C}_k(u) = \sum_{i=1}^{N} w_{k,i} \hat{c}_{k,i}, \quad \hat{D}_k(u) = \sum_{i=1}^{N} w_{k,i} d_i, \tag{2}$$

$$\hat{M}_k(u) = \sum_{i=1}^{N} w_{k,i}, \quad \hat{V}_k(u) = \sum_{i=1}^{N} w_{k,i} (d_i - \hat{D}_k(u))^2. \tag{3}$$

For each object O_k, the losses used during fitting penalize the difference between the inputs and the renderings:

$$\mathcal{L}_{col}(k,u) = M_k(u) \| C(u) - \hat{C}_k(u) \|_1, \tag{4}$$

$$\mathcal{L}_{depth}(k,u) = M_k(u) \frac{\| D(u) - \hat{D}_k(u) \|_1}{\sqrt{\hat{V}_k(u)}}, \tag{5}$$

$$\mathcal{L}_{mask}(k,u) = \| M_k(u) - \hat{M}_k(u) \|_1, \tag{6}$$

where M_k is the binary mask of O_k.

Optimization details. We implement our feature grids and MLP using the tiny-cuda-nn library [13]. Our object models are optimized with AdamW [12] with learning rates 5×10^{-3} and 3.5×10^{-4} respectively for the feature grids and MLPs, and weight decay 0.1 for both. For the background model, we reuse the same parameters as the vMAP paper [10].

3.2. Integrating object shape priors

Constructing the object library. As explained in **??** of the main paper, we build object models offline from either full 3D meshes or video sequences. We detail here the first case. For each object 3D mesh, we render 40 images from random viewpoints around the object using the BlenderProc [5] renderer, each image being of size 1024×1024. We show examples of these renders in Fig. 8. 3D meshes for the Replica dataset are ground-truth object meshes and have been obtained by extracting a closed surface from a single volumetric representation. The latter extraction of object meshes performed by vMAP [10] consists in splitting this closed surface in objects according to vertex instance Ids, resulting in all objects being open surfaces. Thus, if camera poses are randomly sampled all around an object for our renders, the same part of a surface may be observed from two opposite viewpoints, which in turn causes problems during reconstruction. To avoid that issue, we use normal information to only retain one side of the surface. From these rendered images, we fit an object model with the method explained in **??** of the main paper, with the only difference that all inputs are known at the beginning of the fitting. Hence, we do not need to perform the online optimization but instead sample all frames at each optimization step. We perform 500 optimization steps per object and store the model at the last step for our database.

Retrieval and registration. For retrieval, we use the CLIP [16] version *ViT-bigG-14* from OpenClip [2, 8] and filter out retrieved objects for which the cosine similarity score is larger than 0.7. For the registration part, the FPFH [17] features, Ransac [6] and point-to-plane ICP [1] algorithms are reused from Open3D's implementation [24]. We filter the fitness with a threshold of 0.8 and keep registered poses for which at least 90% of reprojected points in the camera frame belong to the input object mask and have a depth larger than the depth measurement, with a tolerance of $2cm$.

Synthesizing keyframes on the fly. Once an object model has been initialized from the object library, we use the retrieved model to render additional views to fit the current object model. In this way, we sample half of the camera poses at each optimization step among the poses stored in the object library. We then render color, depth and mask using 24 points per ray, which we sample uniformly in the whole bounding box. This sampling contains more points than the one from current views (*i.e.*, 14 points) to cope with the absence of depth information that would otherwise guide the sampling around the actual object surface.

3.3. Evaluation datasets

Replica. Objects considered for the evaluations of the main paper slightly differ from those used in vMAP [10]. We clean few noisy meshes to remove vertices that are outliers and discard a few other tiny objects, *i.e.*, with fewer than 50 vertices. Examples of such objects are shown on Figure 9. Following this cleaning, Replica scenes contain on average 50 objects each.

ScanNet. For the ScanNet dataset, we additionally preprocess each depth image to remove outliers. More specifically, at each frame and for each object O_k, we compute the mean m_k and standard deviation s_k of depth values falling in the object mask and discard points whose depth is outside the range $[m_k - \alpha s_k, m_k + \alpha s_k]$, where we choose $\alpha = 1.5$, making this interval close to the 90% confidence interval of normal distributions. We also compute a histogram of depth points belonging to mask k with 15 values in the camera depth range ($[0m, 6m]$) and keep only bins that contain at

least 5% of points. This removes a large number of outliers, though some remain that may have a strong impact on our object bounding boxes. Further joint pre-processing of depth maps and segmentation masks would benefit our reconstructions on real world images. For fair comparison in the real world reconstructions, we also apply this pre-processing to TSDF [3, 22] and vMAP [10].

Real-world sequences. For our videos, we apply the same depth image processing as for ScanNet sequences and additionally erode object masks by few pixels to remove outliers and obtain better geometry.

3.4. Evaluation metrics

For evaluation on seen parts of meshes, we first cull vertices that are not seen in any input frame. We reuse ESLAM's culling script [9] and remove points at a depth $D_{rec} > D_{input} + \tau$, where D_{rec} is the depth of reprojected mesh points in camera frames, D_{input} is the measured depth for that frame and τ is a tolerance. We choose $\tau = 3cm$ for the scene meshes and $2cm$ for object meshes which we found to be a good trade-off between keeping all seen reconstructed vertices that may be inaccurately positioned and discarding all unseen ones. When evaluating reconstructions on the whole objects, including parts that are not seen in the input video sequence, we consider the full ground truth meshes, without applying this culling operation.

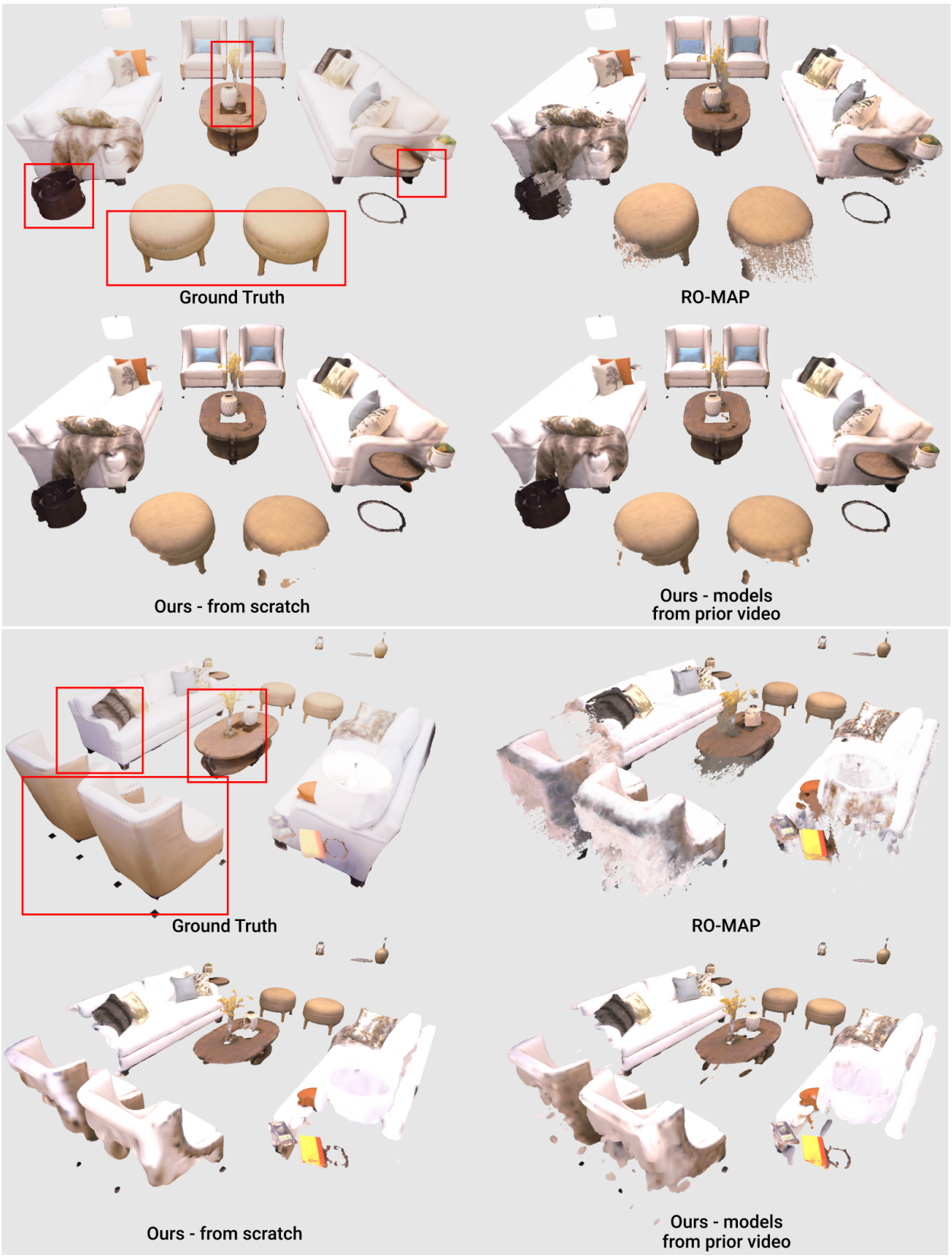

Figure 1. Comparison of object meshes from Replica [19] obtained by RO-MAP [7] and our method, using two different viewpoints. Meshes for RO-MAP were provided by the authors. Our meshes are extracted using marching cubes [11] on a grid of size 64^3 following RO-MAP's methodology and are restricted to objects reconstructed by RO-MAP. Regions of interest are highlighted in red.

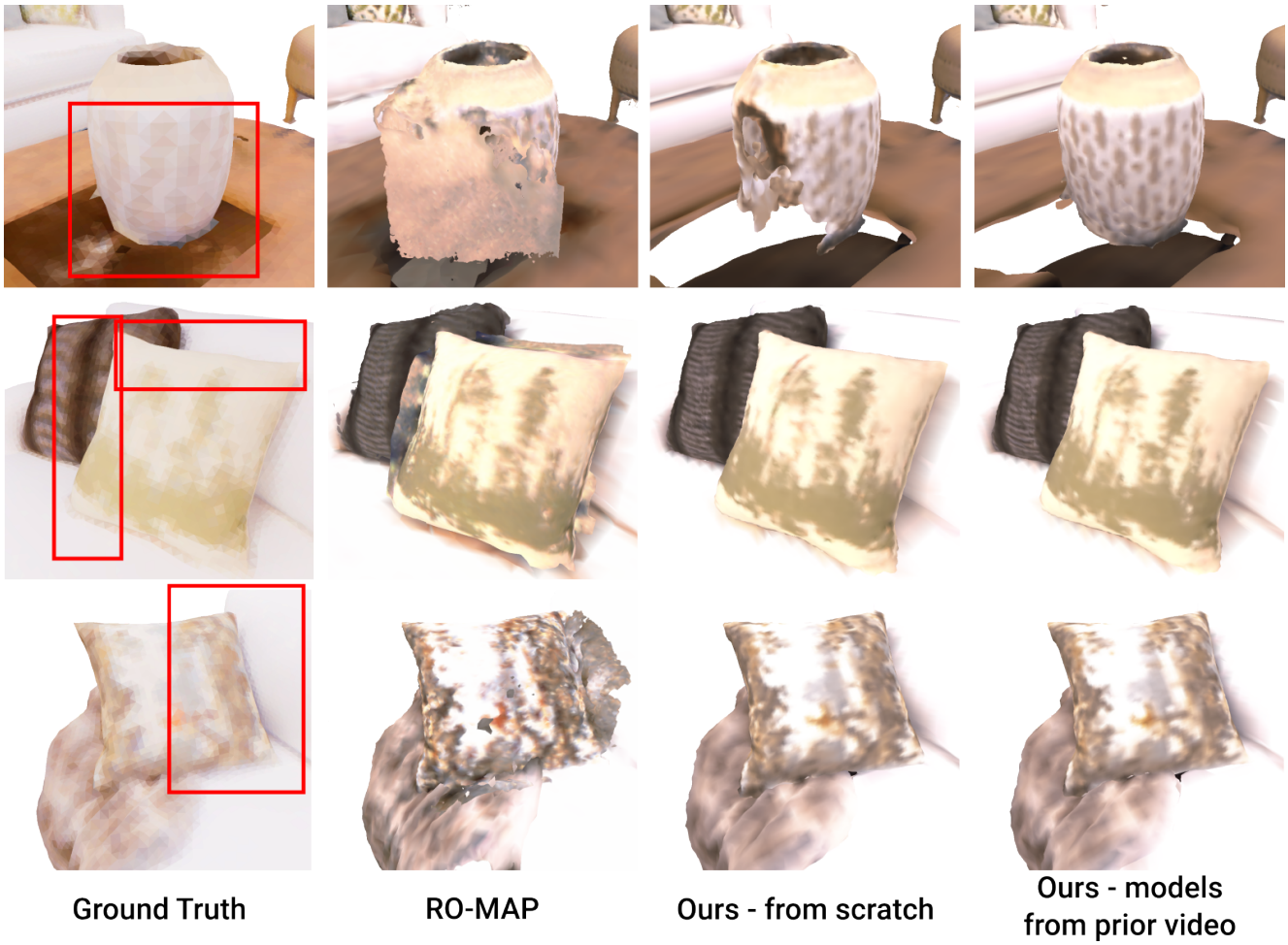

Figure 2. Comparison of object meshes obtained by RO-MAP [7] and our method. Meshes for RO-MAP were provided by the authors. Our meshes are extracted using marching cubes [11] on a grid of size 64^3 following RO-MAP's methodology. Regions of interest are highlighted in red. Note that the back of the vase on the first row is never seen in the first input sequence and only mapped in the second video which we use for our model shown in the last column.

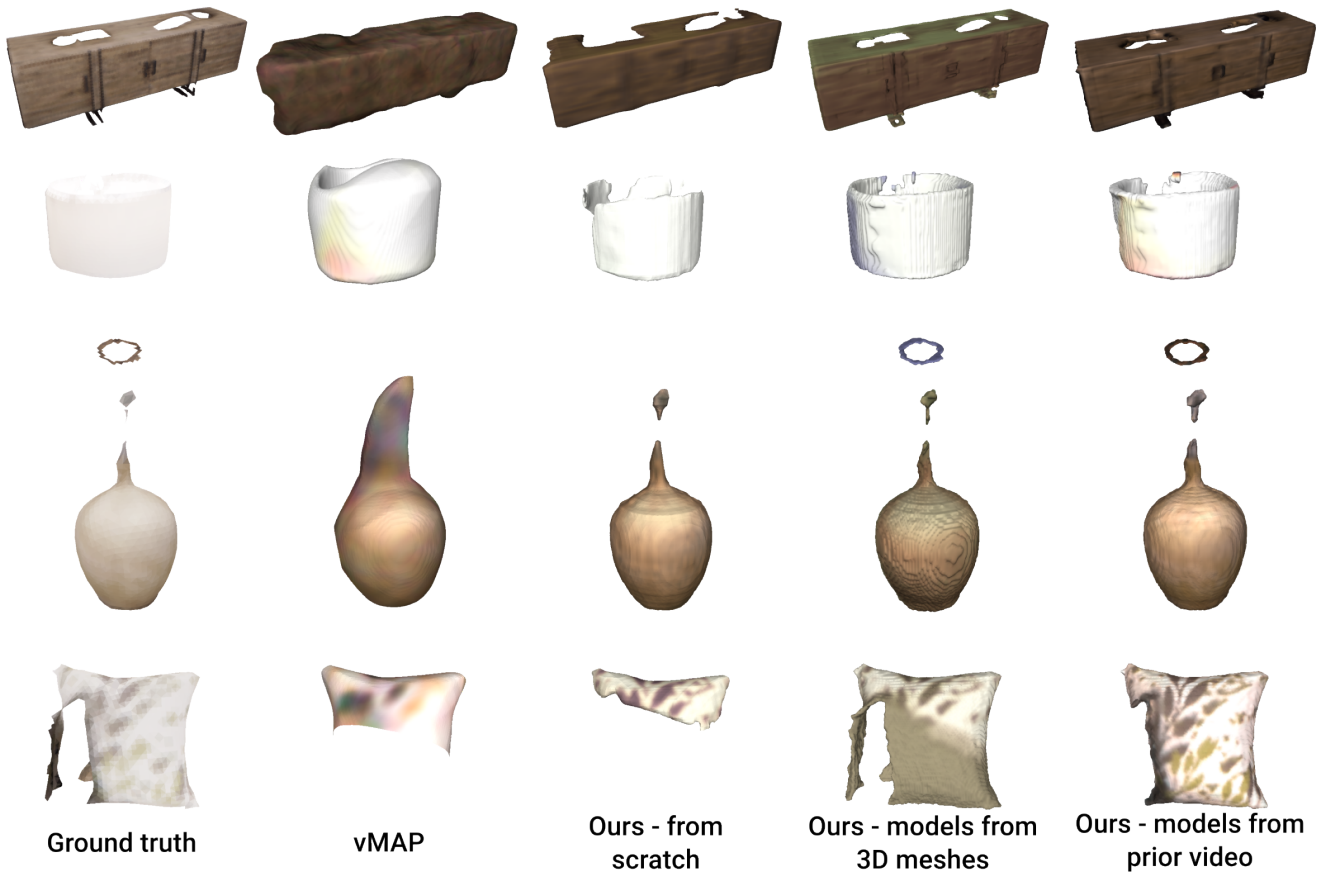

Figure 3. Comparison of meshes between vMAP [10] and the variants of our method on objects from the Replica dataset [19] after only 50 frames of optimization. Meshes are extracted with a resolution of 5mm and our models using object priors rely on ground truth retrieval and registration. While our reconstructions from scratch are more accurate than vMAP, adding object priors helps recovering faster the geometry and, optionally, texture of an object.

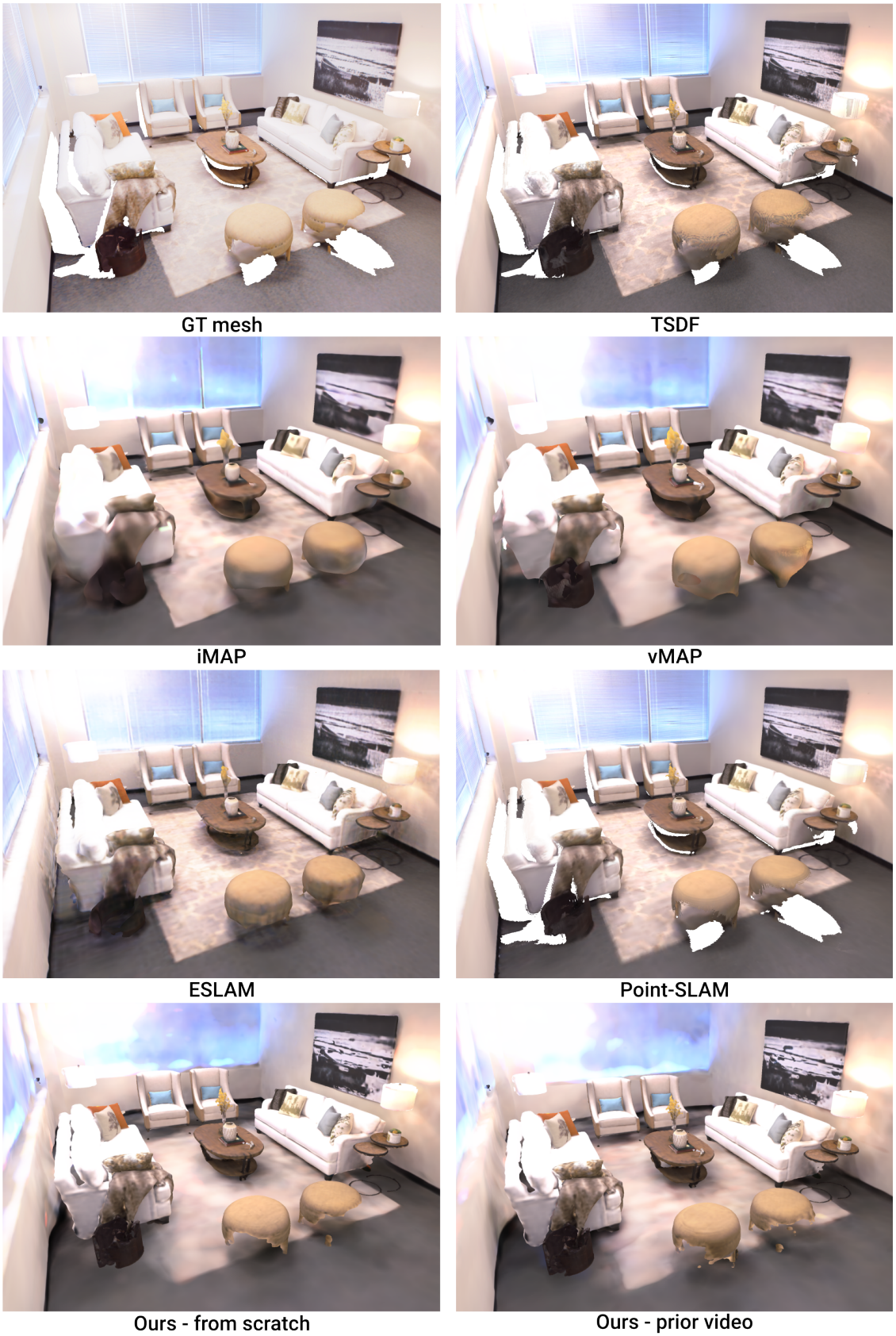

Figure 4. Visualization of reconstructed meshes at the scene level using TSDF-Fusion [3], iMAP [20] (through its reimplementation in [10]), vMAP [10], ESLAM [9], Point-SLAM [18] and our method, with or without object prior, on scene *room-0* of the Replica dataset [19].

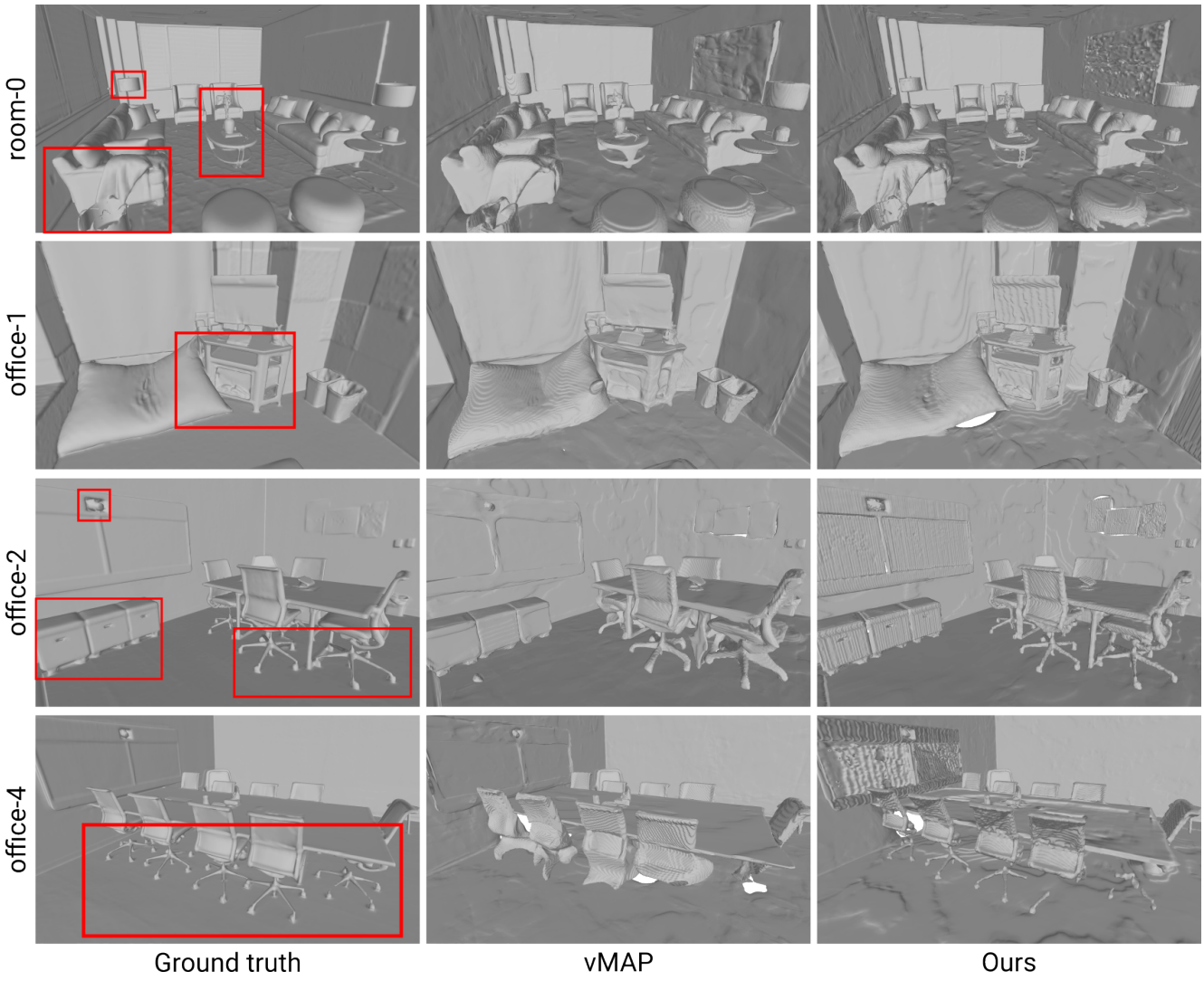

Figure 5. Textureless meshes reconstructed with vMAP [10] and our method on scenes of the Replica dataset [19]. The textured version is presented in **??** of the main paper.

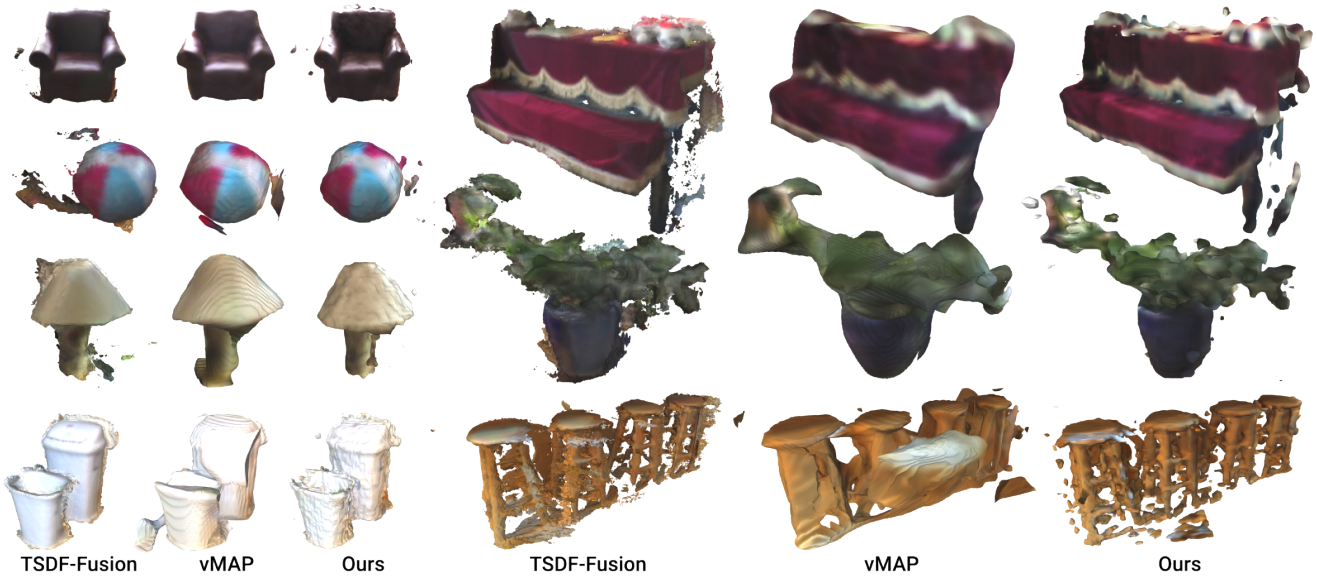

Figure 6. Comparison of meshes between TSDF [3, 22], vMAP [10] and our method on objects from the ScanNet dataset [4]. Meshes are extracted with a resolution of 1cm. Our reconstructions are more accurate than vMAP while being more sensitive about depth and segmentation errors than TSDF.

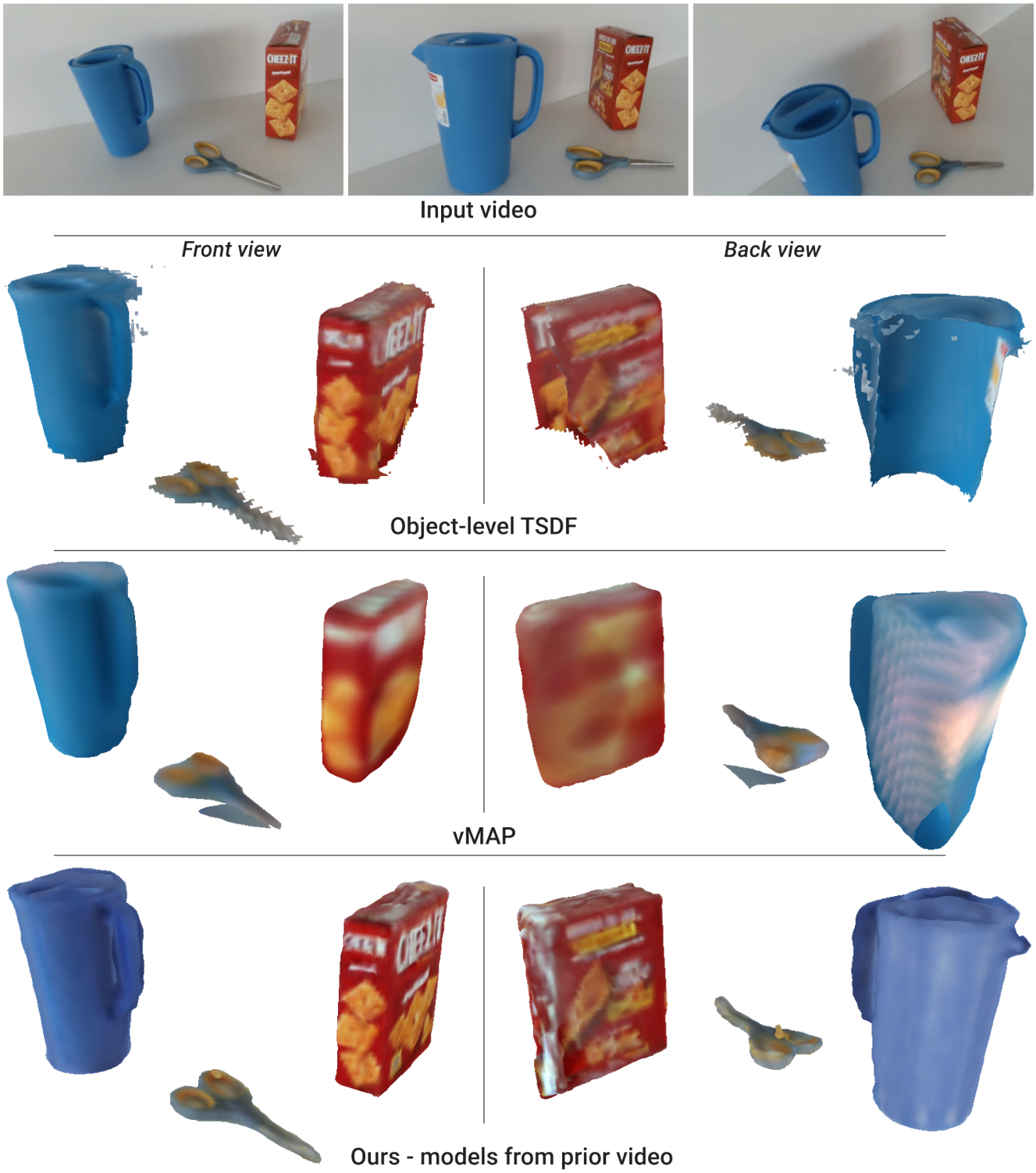

Figure 7. Additional reconstruction of a self-captured sequence with object-level TSDF, vMAP and our method using models from ground truth meshes, viewed from the front and the back.

Figure 8. Examples of rendered images of ground truth meshes using the Blenderproc [5] renderer. Our object models that leverage prior knowledge of GT meshes are fitted on these images.

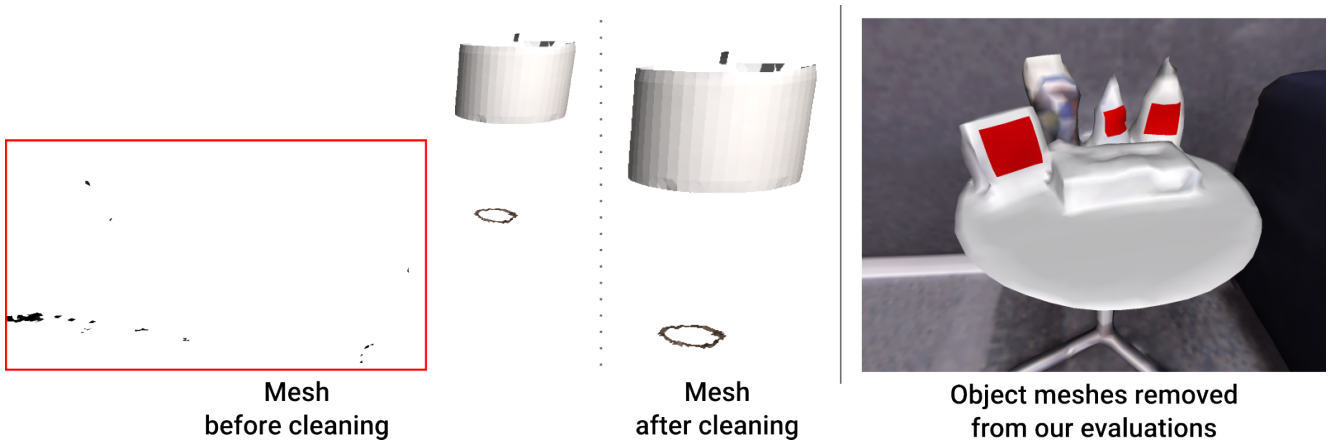

Figure 9. Examples of meshes cleaned before running our evaluations in the main paper. *(Left)*: few objects are cleaned to remove outliers, highlighted in red here. *(Right)*: other small and flat objects like product tags, colored in red, are also discarded.

References

[1] Yang Chen and Gérard G. Medioni. Object modelling by registration of multiple range images. *Image Vis. Comput.*, 10(3):145–155, 1992. 4

[2] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022. 4

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312. ACM, 1996. 2, 3, 5, 9, 11

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 2432–2443. IEEE Computer Society, 2017. 11

[5] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 4, 13

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 4

[7] Xiao Han, Houxuan Liu, Yunchao Ding, and Lu Yang. Ro-map: Real-time multi-object mapping with neural radiance fields. *IEEE Robotics and Automation Letters*, 8(9): 5950–5957, 2023. 1, 6, 7

[8] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 4

[9] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. ESLAM: efficient dense SLAM system based on hybrid representation of signed distance fields. In *CVPR*, pages 17408–17419. IEEE, 2023. 2, 5, 9

[10] Xin Kong, Shikun Liu, Marwan Taher, and Andrew J. Davison. vmap: Vectorised object mapping for neural field SLAM. *CVPR*, 2023. 1, 2, 3, 4, 5, 8, 9, 10, 11

[11] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, pages 163–169. ACM, 1987. 1, 2, 3, 6, 7

[12] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 4

[13] Thomas Müller. tiny-cuda-nn, 2021. 4

[14] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. 1

[15] Evin Pinar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. *CoRR*, abs/2311.18809, 2023. 1

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 4

[17] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3d registration. In *ICRA*, pages 3212–3217. IEEE, 2009. 4

[18] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based SLAM. *CoRR*, abs/2304.04278, 2023. 2, 9

[19] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng 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 A. Newcombe. The replica dataset: A digital replica of indoor spaces. *CoRR*, abs/1906.05797, 2019. 2, 6, 8, 9, 10

[20] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6209–6218. IEEE, 2021. 2, 9

[21] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, pages 17868–17879. IEEE, 2024. 1

[22] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas A. Funkhouser. 3dmatch: Learning local geometric descriptors from RGB-D reconstructions. In *CVPR*, pages 199–208. IEEE Computer Society, 2017. 2, 3, 5, 11

[23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11941–11952. IEEE, 2023. 1

[24] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *CoRR*, abs/1801.09847, 2018. 4