PlanaReLoc: Camera Relocalization in 3D Planar Primitives via Region-Based Structure Matching

Supplementary Material

https://github.com/3dv-casia/PlanaReLoc

In this supplementary material, we provide the following:

- details on dataset preparation (Section A);
- auxiliary settings for baseline methods, including map truncation and the oracle coarse initialization (Section B);
- comprehensive implementation details for PlanaReLoc, including the network architecture, training scheme, and the pose estimation and post refining process (Section C);
- additional analysis and visualizations (Section D);
- discussion of limitations and future work (Section E).

A. Dataset Preparation

In this section, we provide details on the preparation of our experimental datasets. Both the data and the preparation code will be made publicly available.

3D Planar Maps for both the ScanNet and 12Scenes datasets are extracted from their official dense reconstructions using the sequential RANSAC [5] plane-fitting scripts provided by [13, 30]. Specifically, maps from the ScanNet are extracted under the guidance of semantic annotations: (1) mesh vertices are first grouped by their semantic instance labels, and plane fitting is performed within each instance that belongs to plane-supporting categories (*e.g.*, walls, floors, and tables); (2) vertices identified as inliers supporting a primitive are then projected onto their corresponding plane, while preserving their internal connectivity; (3) adjacent planar primitives within the same category are merged to produce a set of complete and semantically aware planes. Lastly, primitives with an area smaller than 0.01 m^2 are discarded. Non-planar vertices and edges connecting distinct primitives are also removed. Meanwhile, each primitive is associated with its planar parameters $\boldsymbol{\pi} := [\mathbf{n}^\top, d]^\top$, where the plane normal $\mathbf{n}$ is oriented consistently with the original surface normal.

In contrast, for the 12Scenes dataset, map primitives are extracted directly using sequential RANSAC without auxiliary semantic annotations and are not further merged, resulting in potentially fragmented and irregular configurations. To ensure a level of compactness comparable to that of ScanNet, each map primitive in 12Scenes is further optimized using the Isotropic Explicit Remeshing method implemented in MeshLab [3].

To obtain colored maps for baselines that require realistic map appearance in our main experiments (*see* Tab. 1), we preserve all plane-supporting vertices along with their original colors. Conversely, when visual appearance is not required, the map can be further compressed by retaining only a few key vertices per primitive to represent its spatial extent, using geometry simplification techniques such as the quadric-based edge collapse [6] or cascaded polygon union [7]. Figure 7 groups the constructed 3D planar maps from ScanNet by the total number of planar primitives. For each bin, it reports the proportion of maps in the group (left y-axis) and the average storage footprint of both the colored and the simplified map versions (right y-axis). The simplified maps occupy an average of 154.3 KiB, which is approximately 3.2% of the size of their colored counterparts.

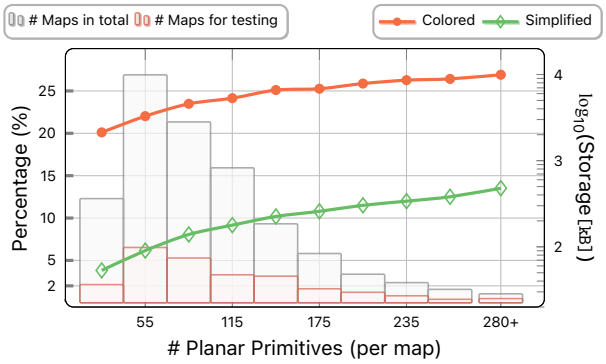
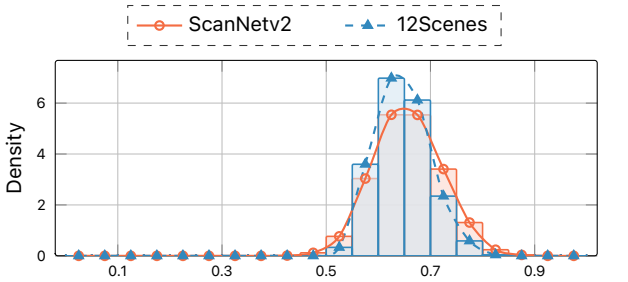


Figure 7. **Statistics of 1513 3D planar maps constructed from ScanNet.** Left y-axis: distribution of 3D planar maps grouped by their number of planar primitives. Right y-axis: average storage footprint of the colored and simplified map versions in each bin.

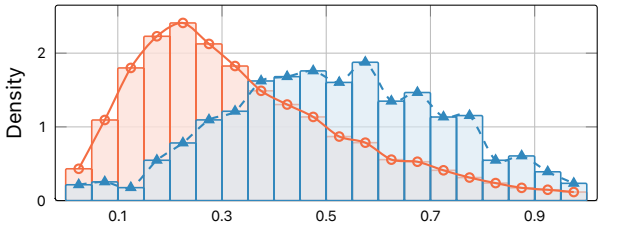
Query Images are sampled from the original RGB-D sequences at regular intervals: every 20 and 30 frames for the ScanNet train and test splits, respectively, and every 5 frames for the 12Scenes test split. Following the protocol of [24], each sampled frame is then verified using its ground-truth camera pose and depth. Frames that fail this consistency check are discarded to ensure precise alignment between the query and the corresponding map. Moreover, frames capturing fewer than three map primitives are excluded to guarantee adequate plane observations and geometric constraints for viable pose estimation.

B. Auxiliary Settings for Baseline Methods

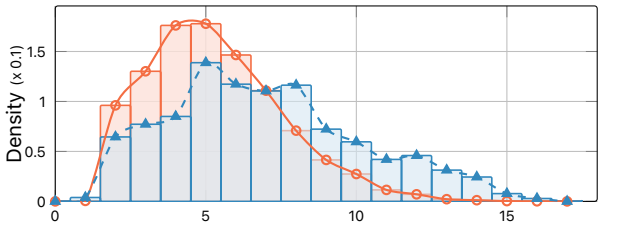
Map Truncation (Map Trunc.) is employed to crop structures far from the ground-truth pose for the image-to-point cloud registration baselines, *i.e.*, *GeoTransformer* [20] and *Free-FreeReg* [27], which may struggle to operate stably



(a) Visual overlap between the GT pose and the truncated map



(b) Visual overlap between the GT pose and the *Coarse Init.*



(c) Number of covisible planes between the GT pose and the *Coarse Init.*

Figure 8. **Visual overlap analysis for the map truncation in (a) and the oracle coarse initialization strategy in (b) and (c).**

on the full-scene maps (*see* Tab. 1). Specifically, given the ground-truth pose and depth map, we define a reference point as the 3D point on the principal ray located at the mean scene depth. Then, we retain only the structures within a 3 m-sized axis-aligned bounding box centered at the reference point. As illustrated in Fig. 8a, the strategy yields an average 2D–3D overlap exceeding 60 %, as measured by the protocol from [20].

Oracle Coarse Initialization (Coarse Init.) is used to provide a 6-DoF reference pose for the baselines that rely on map renderings (*see* Tab. 1). Specifically, we construct a sub-map containing 20 primitives by first including all visible ones and then supplementing with nearest ones to the reference point defined above. To obtain a reference rotation, we first compute the mean normal vector of these primitives. The camera's viewing direction is then aligned with the opposite of this average normal vector, towards the map, while its up vector is fixed to the global up direction $(0, 0, 1)^\top$. The reference translation is computed by shifting the center of each primitive by 2 meters along its normal direction and then taking the mean of the resulting shifted centers. Figures 8b and c show the distributions of visual overlap (as defined by [22]) and the number of covisible planes resulting from these heuristic rules, respectively. In summary, our oracle initialization heuristics achieve an average visual overlap of 34.7 % (∼5.3 covisible planes) on ScanNet and 51.6 % (∼7.0 covisible planes) on 12Scenes. This provides reasonable viewpoints for rendering synthetic images used in matching.

More Details. For MeshLoc methods, following the protocol of [17, 18], we render synthetic views using only the model's base vertex color, without computing any lighting effects, and use PoseLib [12] as the robust pose estimator for all point-matching approaches. For all methods, we apply a final clamping step to ensure that the estimated poses lie within the bounds of the scene maps.

C. Implementation Details for PlanaReLoc

2D Plane Embeddings. As shown in Fig. 9, instead of employing an independent image backbone, we augment MoGe-2 [28], the monocular geometry estimation model for plane recovery, with an additional head comprising lightweight convolutional layers to extract dense features for 2D plane embeddings. This design allows the model to leverage powerful visual representations learned on large-scale datasets, thereby improving accuracy while maintaining efficiency during both training and inference. Further ablation studies of this design choice can be found in Sec. D.

The query images are first resized to a resolution of 640 × 480 and fed into MoGe-2 to obtain an estimated metric depth map, which is subsequently downsampled to match the size of the feature map (80 × 60). The RANSAC module [5, 29] operates on the downsampled depth map directly, as higher resolution yields little performance improvement but incurs extra computational overhead. During plane extraction, a point is considered an inlier of a plane hypothesis if both (1) its distance residual to the plane is less than 10 cm and (2) their normal similarity, measured by the dot product, exceeds 0.9. The module iteratively extracts planes from the depth map, until either 16 primitives have been extracted or the number of inliers falls below 1 % of the total number of pixels in the depth map.

3D Plane Embeddings. Both the object and scene encoders are instantiated using PointNet [19], operating on point clouds sampled from the map primitives. For each map primitive, we uniformly sample L=1024 points, which are then centralized, batched, and fed into the object encoder. For the scene encoder, we first retain 16 points per primitive to ensure each primitive will be represented. Additional points are then randomly sampled from the entire map until the total number of points reaches 16 × 1024=16 384 (based on the assumption of a maximum of 1024 map primitives).

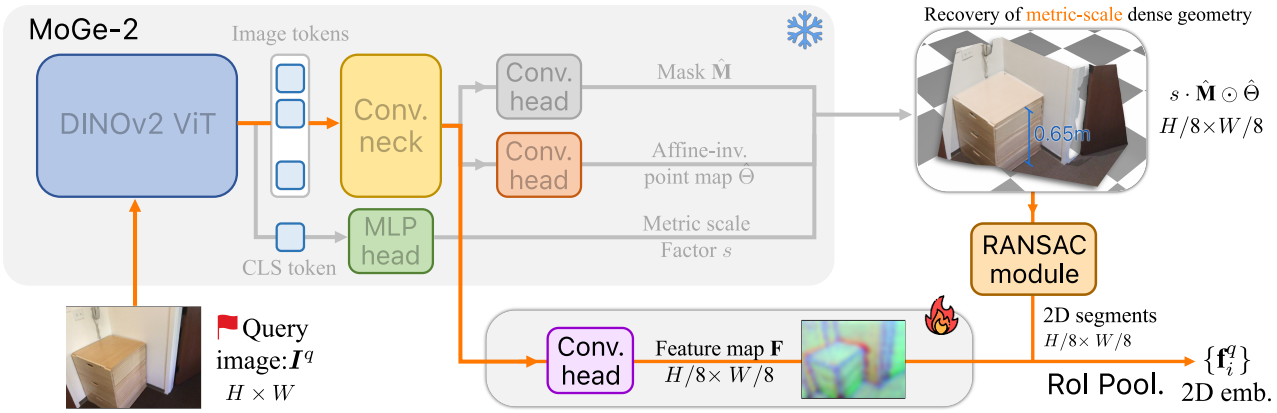


Figure 9. **Architecture of the front-end network on the query side.** We retrofit the monocular geometry estimation model MoGe-2 [28] with an additional head to encode dense features for 2D plane embedding.

Training Scheme. Our model, which consists of the front-end encoders (*see* Sec. 3.1) and the matching network (*see* Sec. 3.2), is trained on the ScanNet training split by minimizing the loss defined in Eq. (3). We train the model with a batch size of 16, distributed across 2 NVIDIA A800 GPUs, for 90 000 iterations, equivalent to approximately 31 epochs. The overall training scheme consists of two stages: (1) during the initial 45k iterations, we use the ground-truth primitives augmented with noise to replace the online monocular plane recovery for improved training efficiency; (2) in the remaining 45k iterations, we switch to the full pipeline, enabling online plane recovery. For both stages, we employ the AdamW optimizer [14] with an initial learning rate of 1×10^{-4}. The learning rate is decreased by a factor of 0.1 at 24k and 36k iterations using a multi-step scheduler. Throughout training, we apply data augmentation to both the query images (random resizing and cropping) and the maps (random rotation and scaling). The entire training process completes in about 12 h.

Estimating Camera Translation, as introduced in Eq. (7), involves the joint optimization of the metric scale factor s:

$$\mathbf{t}_0, s^* = \arg\min_{\mathbf{t},s} \sum_{(i,j)\in\widehat{\mathcal{M}}} \omega_i (\mathbf{t}^\top \mathbf{n}_j^m - d_j^m + s d_i^q)^2.$$

The initial translation $\mathbf{t}_0$ and the optimal s^* can be solved efficiently by rewriting the above equation as a standard linear least-squares problem: (1) for each correspondence $(i,j) \in \widehat{\mathcal{M}}$, we construct a linear equation by setting the row vector to $a := [(\mathbf{n}_j^m)^\top, d_i^q]$ and the target to $b := d_j^m$; (2) stacking all correspondences yields a linear system $\boldsymbol{A}\mathbf{x} \approx \boldsymbol{b}$, with the variable vector $\mathbf{x} := [\mathbf{t}_0^\top, s]^\top \in \mathbb{R}^4$, the data matrix $\boldsymbol{A} \in \mathbb{R}^{|\widehat{\mathcal{M}}|\times 4}$, and the target $\boldsymbol{b} \in \mathbb{R}^{|\widehat{\mathcal{M}}|}$; (3) we introduce the diagonal weight matrix $\boldsymbol{W} = \mathtt{diag}(\sqrt{\omega_i})$ and formulate the final weighted linear least-squares problem as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\boldsymbol{W}(\boldsymbol{A}\mathbf{x} - \boldsymbol{b})\|_2^2, \tag{1}$$

which can be efficiently solved in closed form using SciPy [26]. The degeneracy rate, *i.e.*, the percentage of cases where the number of correspondences with non-parallel normals is less than 3, is empirically observed to be less than 2 %.

Pose Refinement. We optimize for the relative transformation $\mathbf{T}_{\text{tr}}^*$ alongside the offset seeds $\{\delta_i^*\}$ by minimizing the depth alignment cost E_{depth} defined in Eq. (10). The optimization is performed using the Adam optimizer [11], following the practice in [15]. Specifically, the optimization variable $\mathbf{T}_{\text{tr}}$ is converted into a differentiable 6-dimensional vector via LieTorch [25] and then optimized for 200 iterations with a learning rate of 1×10^{-3}. The offset seeds $\{\delta_i\}$ are initialized to one and optimized with a learning rate of 1×10^{-4}. For efficiency, in each iteration we compute E_{depth} on a multinomial sample of 4096 pixels from the 2D plane segments, rather than capitalizing on every pixel.

D. Additional Experimental Results

Cumulative Accuracy Curves. Figure 10 presents the cumulative accuracy curves w.r.t. translation and rotation errors on both ScanNet and 12Scenes datasets. Our method achieves solid performance on both datasets, with particularly favorable results in camera rotation, which may benefit from the reliable normal priors predicted by the powerful monocular model. In contrast, estimating camera translation is largely affected by depth inaccuracy, a limitation that can be mitigated through the post-refinement procedure introduced in Sec. 3.4.

Ablating 2D Encoder Variants. We compare different designs of the query-side encoder. In addition to our default design depicted in Fig. 9, we also evaluate three alternative configurations: (1) training a dedicated ResNet-50 [8] from scratch; (2) employing the official pretrained DINOv2-Vit-L/14 [16] as a frozen backbone; (3) a variant of (2) aug-

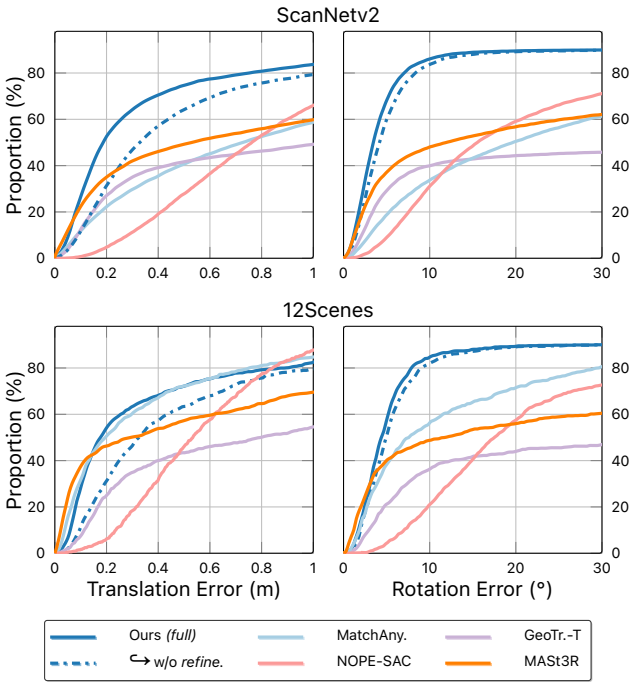


Figure 10. **Camera relocalization results on ScanNet and 12Scenes datasets.** We plot cumulative accuracy curves that show the proportion of correctly localized frames as a function of varying translation and rotation error thresholds.

Table 6. **Ablation study on 2D encoder variants.** Experiments are conducted on ScanNet with no post-refinement.

	Plane Matching (%) ↑			Pose	Time
	Prec.	Rec.	F_1	Recall ↑	(ms/iter)
(1) ResNet50 [8]	62.3	56.9	59.5	34.2	∼ 63.9
(2) DINOv2 [16]	65.5	59.6	62.4	35.5	∼ 95.0
↪ (3) *w/* Conv. head	*66.3*	*60.3*	*63.1*	*36.2*	∼ 97.1
Ours	**67.6**	**61.3**	**64.3**	**37.1**	∼ 59.9

mented with the same learnable convolutional head as in our default design to better adapt the features to our task. As reported in Tab. 6, the results suggest that the DINOv2 ViT encoder fine-tuned by MoGe-2 [28] possesses enhanced geometric and structural perception, contributing to the slight performance improvement over the official frozen DINOv2 features. Moreover, the reuse of visual representations from the pretrained backbone for 2D plane embedding provides good computational efficiency. Collectively, these results highlight the effectiveness of our architectural design.

Impact of the Query Primitive Size. We analyze how plane matching performance varies w.r.t. the size of primitives observed in query images. Specifically, all detected query primitives on ScanNet are collected and categorized into bins according to their pixel areas. We then compute the matching precision for each bin, defined as the proportion of correctly matched primitives. As shown in

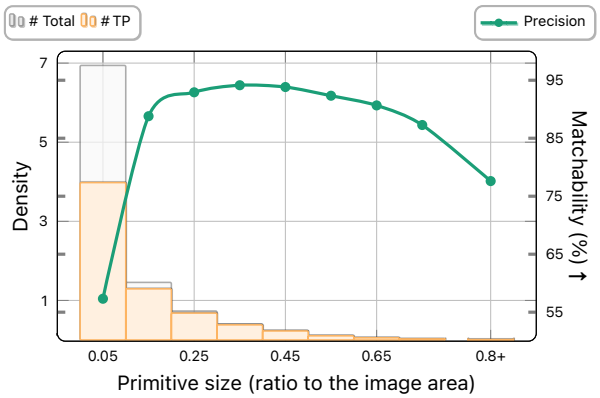


Figure 11. **Impact of the Query Primitive Size on Matching Performance.** Left y-axis: the size distribution of all detected query primitives, where size is defined as the percentage of the image area occupied. Right y-axis: the matching precision for each size bin.

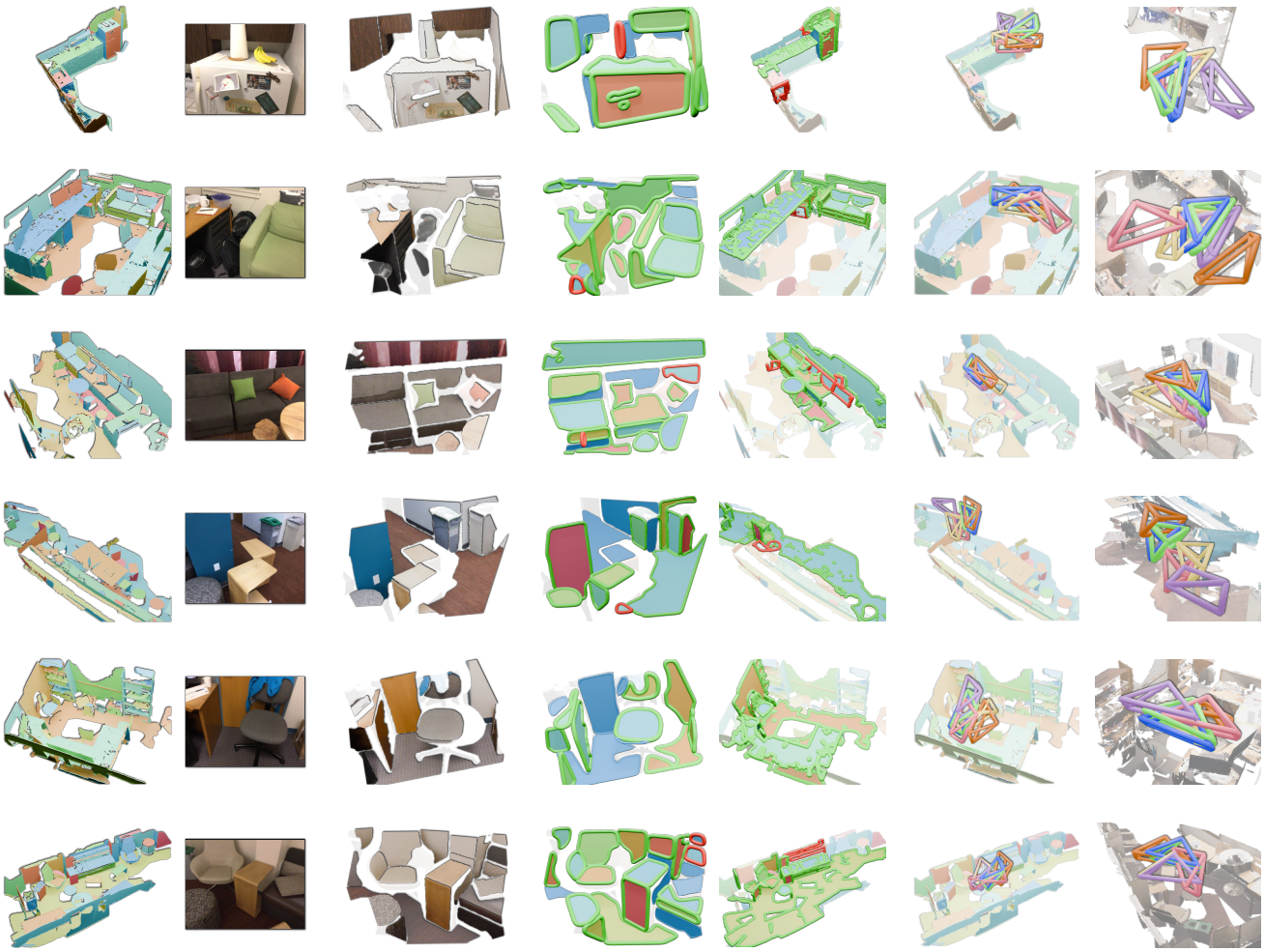
Fig. 11, our purely geometric plane recovery method tends to over-segment planar regions due to occlusions and noise. Consequently, a large number of small plane segments are produced, typically covering less than 10 % of the image and exhibiting low matching reliability. Moreover, the Mutual Nearest Neighbor (MNN) matching strategy, which imposes a one-to-one correspondence constraint, also leads to lower matching precision in these small planar primitives. Meanwhile, performance drops significantly for extremely large planes that occupy more than 80 % of the image area. As noted in our main paper, this likely stems from the reduced plane richness and the consequent lack of discriminative patterns. In contrast, the remaining medium-sized query primitives strike a good balance between salience and discriminativeness, achieving a remarkable matching precision of over 90 % on average.

Evaluation on 7Scenes. To better position the task and the PlanaReLoc method within the broader visual localization literature, we additionally include a cross-dataset evaluation on the standard 7Scenes dataset [23] to compare against more representative methods and assess the performance gap. Specifically, we construct planar maps using the depth scans from the 7Scenes train split, and evaluate PlanaReLoc on the test split. The results are reported in Tab. 7. While maintaining a compact map representation and exhibiting consistent cross-dataset performance, we acknowledge that PlanaReLoc still lags behind state-of-the-art visual localization methods that fully leverage thousands of reference images which provides rich visual cues and pose priors.

Additional Qualitative Results. More visualizations of intermediate outputs and relocalization results on 12Scenes and ScanNet are shown in Fig. 12 and Fig. 13, respectively.

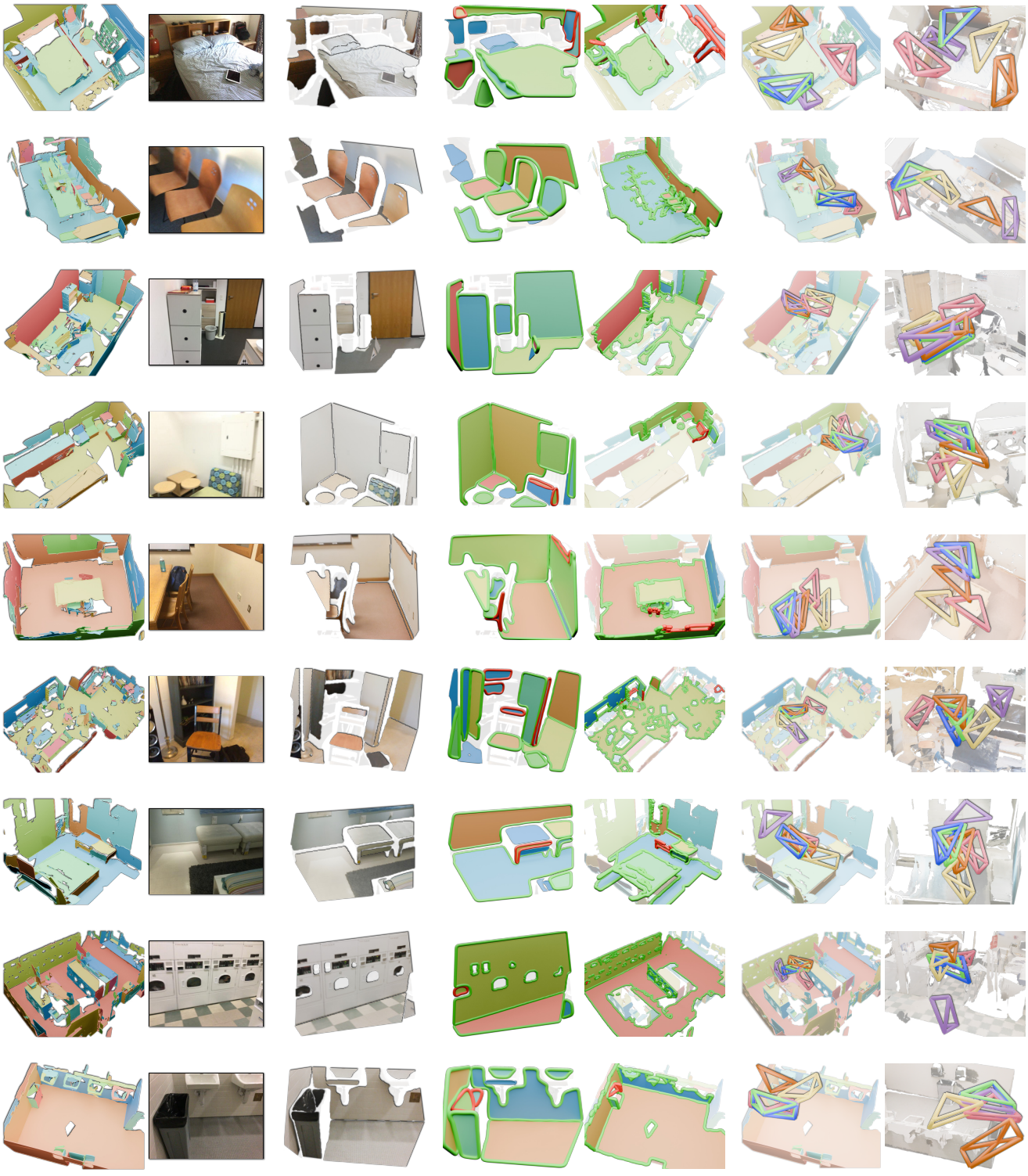
Table 7. **Camera relocalization results on the 7Scenes dataset.** We report median position and rotation errors in centimeters (cm) and degrees (°), respectively. We summarize the map type, map size, the time needed for mapping, and whether mapping and localization rely on visual appearance or pose priors (*e.g.*, image retrieval). Results of other methods are taken from the literature.

Methods	Map Type	Map Size ↓	Mapping Time ↓	Visual Cues	Pose Prior	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	**Avg.**↓ (cm/°)
PoseNet17 [10]	Network	50 MB	4 h–24 h	✓		13/4.5	27/11.3	17/13.0	19/5.6	26/4.8	23/5.4	35/12.4	23/8.1
HLoc(SP+SG) [21]	SfM points	∼2 GB	∼ 1.5 h	✓	✓	2/0.8	2/0.9	1/0.8	3/0.9	5/1.3	4/1.4	5/1.5	3/1.1
GoMatch(SP) [31]	SfM points	∼56 MB	∼ 1.5 h		✓	4/1.6	12/3.7	5/3.4	7/1.8	8/5.7	14/3.0	58/13.1	18/4.6
ACE [2]	Network	5 MB	5 min	✓		0.6/0.2	0.8/0.3	0.5/0.3	1/0.3	1/0.2	0.8/0.2	3/0.8	1/0.3
Reloc3r [4]	Ref. images	∼1.6 GB	7 s	✓	✓	3/0.9	3/0.8	1/1.0	4/0.9	6/1.1	4/1.3	7/1.3	4/1.0
STDLoc [9]	3DGS	∼0.8 GB	∼ 2 h	✓		0.5/0.2	0.6/0.2	0.4/0.3	1/0.2	1/0.2	0.6/0.2	1/0.4	0.8/0.2
Ours	Planes	0.4 MB	2 min			26/3.7	19/3.8	32/3.7	19/3.6	27/2.6	43/3.3	61/6.2	32/3.9



(a) Input: map & query (b) Monocular plane recovery (c) Plane correspondences (d) Poses (viewpoint 1 & 2)

Figure 12. **Qualitative examples on 12Scenes.** Correspondences in (c) are color-coded, with true positives outlined in green and false ones in red. Legend for different relocalizers in (d): the ground truth, PlanaReLoc(Ours), GeoTransformer-T, *Coarse Init.*, MASt3R, NOPE-SAC.



(a) Input: map & query (b) Monocular plane recovery (c) Plane correspondences (d) Poses (viewpoint 1 & 2)

Figure 13. **More qualitative examples on ScanNet.** Correspondences in (c) are color-coded, with true positives outlined in green and false ones in red. Legend for different relocalizers in (d): the ground truth, PlanaReLoc(Ours), GeoTransformer-T, *Coarse Init.*, MASt3R, NOPE-SAC.

E. Limitations and Future Work

Limitations A key bottleneck of our method lies in the monocular plane recovery module, given its critical role in providing 2D plane proposals and geometric priors for subsequent matching and pose estimation. Despite significant progress in this area, unreliable predictions from this module under challenging scenarios can still lead to catastrophic failures, even with our robust pose estimation and refinement pipeline designed to mitigate errors.

Another issue arises in environments with a limited level of detail or exhibiting highly repetitive structures, or when the query image captures only weak structural hints (*see* Fig. 14). This limitation is also indicated by Tab. 5 in the main paper: even when provided with ground-truth monocular plane recoveries, PlanaReLoc may still fail to establish enough correct matches in certain cases.

Furthermore, in large multi-room scenarios (*see* Fig. 15), PlanaReLoc's performance is constrained by the increasing structural ambiguities and the fixed point budget that the scene encoder consumes. Increasing the point budget or processing subdivided regions in parallel yield limited performance improvements, but at the cost of substantial memory and computational overhead.

Finally, PlanaReLoc is currently better suited to indoor settings and is not trained or validated in outdoor environments, where the plane distribution may differ significantly.

Future Work. Although PlanaReLoc demonstrates strong performance in cross-modal 2D–3D matching and enables a plane-centric paradigm for room-level 6-DoF camera relocalization, scaling it up to larger and more complex scenes demands improved scene understanding and structural disambiguation. This could be addressed by enhancing structural feature encoding, incorporating plane semantics, and adopting a coarse-to-fine strategy. Moreover, exploring an end-to-end approach that jointly tackles structural matching and pose estimation could further improve robustness and accuracy. Lastly, extending the method to sequential inputs offers another promising direction for practical use.

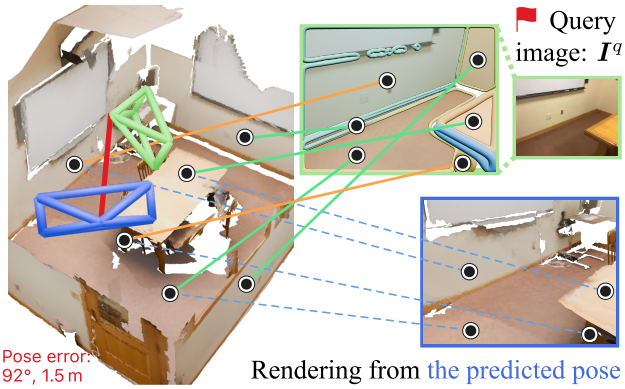


Figure 14. **A representative case where PlanaReLoc underperforms.** Despite four out of six primitives being correctly matched (colored in green), the pose estimation framework fails to reject matching outliers (colored in orange) due to the perfectly repeated pattern (compare the query image with the colored rendering from the predicted pose).

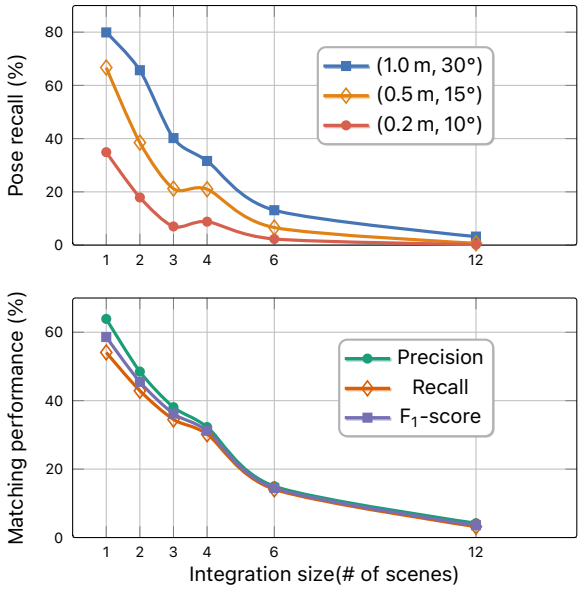


Figure 15. **Relocalization results on Integrated Rooms.** Following [1], we arrange scenes in 12Scenes inside a 2D grid with a cell size of 5 m and integrate varying numbers of adjacent scenes to form larger maps. As the integration size increases, PlanaReLoc's performance degrades due to increased ambiguities and the scene encoder's limited capacity for larger maps.

References

[1] Eric Brachmann and Carsten Rother. Expert Sample Consensus Applied to Camera Re-Localization. In *ICCV*. 2019. 7

[2] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In *CVPR*. 2023. 5

[3] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: An open-source mesh processing tool. In *Eurographics Italian Chapter Conference*. 2008. 1

[4] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-Scale Training of Relative Camera Pose Regression for Generalizable, Fast, and Accurate Visual Localization. In *CVPR*. 2025. 5

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of The Acm*, 24(6):381–395, 1981. 1, 2

[6] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*. 1997. 1

[7] GEOS contributors. GEOS computational geometry library. 2025. Available at https://libgeos.org/. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[9] Zhiwei Huang, Hailin Yu, Yichun Shentu, Jin Yuan, and Guofeng Zhang. From sparse to dense: Camera relocalization with scene-specific detector from Feature Gaussian Splatting. In *CVPR*, 2025. 5

[10] Alex Kendall and Roberto Cipolla. Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*. 2017. 5

[11] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*. 2015. 3

[12] Viktor Larsson and contributors. PoseLib - minimal solvers for camera pose estimation, 2020. Available at https://github.com/vlarsson/PoseLib. 2

[13] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3D plane detection and reconstruction from a single image. In *CVPR*. 2019. 1

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 3

[15] Kirill Mazur, Gwangbin Bae, and Andrew J. Davison. SuperPrimitive: Scene Reconstruction at a Primitive Level. In *CVPR*, 2024. 3

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, *arXiv preprint arXiv:2304.07193*, 2023. Available at http://arxiv.org/abs/2304.07193. 3, 4

[17] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-based visual localization. In *ECCV*, 2022. 2

[18] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Visual Localization using Imperfect 3D Models from the Internet. In *ICCV*. 2023. 2

[19] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*. 2017. 2

[20] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. GeoTransformer: Fast and Robust Point Cloud Registration with Geometric Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):9806–9821, 2023. 1, 2

[21] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*. 2019. 5

[22] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *ECCV*. 2022. 2

[23] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-d images. In *CVPR*, 2013. 4

[24] Bin Tan, Nan Xue, Tianfu Wu, and Gui-Song Xia. NOPE-SAC: Neural one-plane RANSAC for sparse-view planar 3D reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45 (12):15233–15248, 2023. 1

[25] Zachary Teed and Jia Deng. Tangent space backpropagation for 3D transformation groups. In *CVPR*, 2021. 3

[26] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. 3

[27] Haiping Wang, Yuan Liu, Bing Wang, Yujing Sun, Zhen Dong, Wenping Wang, and Bisheng Yang. FreeReg: Image-to-Point Cloud Registration Leveraging Pretrained Diffusion Models and Monocular Depth Estimators. In *ICLR*, 2024. 1

[28] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. In *NeurIPS*. 2025. 2, 3, 4

[29] Jamie Watson, Filippo Aleotti, Mohamed Sayed, Zawar Qureshi, Oisin Mac Aodha, Gabriel Brostow, Michael Firman, and Sara Vicente. AirPlanes: Accurate plane estimation via 3D-consistent embeddings. In *CVPR*, 2024. 2

[30] Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. PlanarRecon: Realtime 3D Plane Detection and Reconstruction from Posed Monocular Videos. In *CVPR*, 2022. 1

[31] Qun Jie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough For Matching In Visual localization? In *ECCV*. 2022. 5