On the Limits of Pseudo Ground Truth in Visual Camera Re-Localization —Supplementary Material—

Eric Brachmann[1] Martin Humenberger[2] Carsten Rother[3] Torsten Sattler[4]
[1]Niantic [2]NAVER LABS Europe [3]Visual Learning Lab, HCI/IWR, Heidelberg University
[4]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague

Sec. 1 presents implementation details for our baseline methods (*c.f*. Sec. 5 - "Baselines" in the main paper). Sec. 2 provides visualisations comparing the original and the Structure-from-Motion (SfM) pseudo ground truth (pGT), complementing Figures 1, 3, 5, and 6 in the main paper. Sec. 3 extends Figure 4 in the main paper by providing 3D alignment statistics for all scenes in the 7Scenes [16] and 12Scenes [18] datasets. Finally, Sec. 4 provides detailed plots for the localization results presented in Sec. 5 of the main paper.

1. Implementation Details

In the following, we detail how we adjusted the source code of Active Search [14], hLoc [11, 12], R2D2 [6], and DSAC* [4] and provide training details for the latter.

1.1. Active Search (AS)

We use the source code of [14], but replace the original RANSAC method with the LO-RANSAC [8] implementation from [13]. Local optimisation is implemented by minimising the sum of squared reprojection errors over a subset of the inliers of the best pose found so far. In addition, we perform non-linear optimisation of the pose by minimising the sum of squared reprojection errors over all inliers after LO-RANSAC. In both cases, Ceres [1] is used to implement the optimisation. Based on preliminary experiments, both modifications significantly improve performance.

We set the inlier threshold for LO-RANSAC to 1% of the image diagonal[1] and use 10k visual words trained on an unrelated outdoor dataset for prioritization. For the SfM pGT, which provides an estimate of the radial distortion of the test images, we undistort the SIFT [9] feature positions in the test images before RANSAC-based pose estimation.

AS requires a SfM model of the scene for 2D-3D matching. We use COLMAP to build these models by triangulating the 3D structure of the scene from the known pGT poses of the training images. To establish the matches required for triangulation, we use COLMAP's image retrieval pipeline [15] to match each training image against the top-100 retrieved other training images. In addition, we match each training image against each other training image that has a pGT pose difference below 2m and 45°. For the original pGT of 7Scenes, we obtained better results by relaxing the thresholds COLMAP uses for triangulation. As in the main paper, we account for the transformation between the depth and the RGB cameras when building the SfM models for the original 7Scenes pGT.[2]

1.2. Hierarchical Localization (hLoc)

Similar to Active Search, hLoc [11, 12] is based on local features. Whereas Active Search relies on SIFT [9], hLoc employs SuperPoint features [5], a modern learned alternative. Active Search directly matches features extracted from the test image against descriptors associated with the 3D points. In contrast, hLoc first employs an image retrieval stage to identify a set of training images that potentially show the same part of the scene as the test image. The features found in the test image are then only matched against the 3D points visible in the top-k retrieved training images. For matching, the SuperGlue [12] approach is used to improve matching quality. The resulting 2D-3D matches are then used to estimate the camera pose by applying a P3P solver inside a RANSAC loop.

We use the source made publicly available by the authors and use their default settings. While the original publication describing the hierarchical localization pipeline [11] uses NetVLAD [2] descriptors, we use DenseVLAD [17] descriptors instead. DenseVLAD is a non-learned alternative to NetVLAD, where densely extracted RootSIFT [3] features are pooled into a VLAD [7] descriptor. We chose DenseVLAD as it, in our experience, performs better for the 7Scenes and 12Scenes datasets than NetVLAD and use the top-20 retrieved images.

[1]While we observed better results when tuning the threshold per scene, we want to avoid overfitting to the test set and thus use the same setting for all scenes.

[2]Note that the SfM pGT directly provides poses for the RGB images and it is not necessary to account for the transformation.

1.3. DenseVLAD+R2D2

DenseVLAD+R2D2 [6] follows the workflow of image retrieval as well as structure-based methods where, first, the most similar training images are retrieved using global image representations, and second, these image pairs are used for local feature matching. Same as for our hLoc experiments (we use exactly the same retrieval results), for image retrieval during localisation, we use DenseVLAD [17] features and for mapping, we use a list for matching training images that was obtained as a result of finding co-observations of reconstructed 3D points (using the AS map as basis). For local feature matching, and in addition to Active Search and hLoc (which use SIFT resp. SuperPoint), here we use R2D2 [10] features. DenseVLAD+R2D2 uses COLMAP for both, 3D point triangulation of the map and image registration using 2D-3D correspondences. The matches are obtained using the nearest neighbors in descriptor space (L2-norm), cross-validation, and geometric verification.

Instead of triangulating keypoint matches using the camera poses, for **DenseVLAD+R2D2 (+D)**, we construct the 3D map by projecting the keypoints to 3D space using the provided and registered [20] depth maps. For localization, we follow the same method as described above.

1.4. DSAC*

We use the public code of DSAC* [4] with default parameters. DSAC* supports different training modes utilising varying degrees of supervision. To achieve best results, we follow Brachmann and Rother [4] and initialize the DSAC* network using scene coordinate ground truth. Brachmann and Rother render ground truth scene coordinates using 3D models of each scene provided in the 7Scenes and 12Scenes datasets, respectively. Next to the pseudo ground truth camera poses of these datasets, the 3D models are an additional output of (RGB-)D SLAM. Hence, these 3D models would add an additional, non-trivial dependency of DSAC* training to the underlying dataset reference algorithm. To restrict the influence of the reference algorithm to pGT poses alone, we train DSAC* using ground truth scene coordinates that we obtain from the measured depth map of each image. We backproject the depth map to 3D using the camera calibration parameters, and transform them to scene space using the pGT pose. For 7Scenes, we manually register depth maps to RGB images using the calibration parameters provided by [20].

Since the DSAC* code does not support a camera model with radial distortion, we instead undistort RGB images using COLMAP before passing it to the DSAC* pipeline. We only do this for experiments with the SfM pGT since the (RGB-)D SLAM pGT assumes zero radial distortion.

We follow Brachmann and Rother [4] and train DSAC* for 1.1M iterations (initialization + end-to-end). This took approximately 16 hours per scene on a GeForce RTX 2080 Ti, which is considerably faster than the 60 hours reported in [4], presumably due to more recent hardware. Compared to the results published in [4], we observe slightly reduced accuracy, *e.g.* 82.9% versus 85.2% for DSAC (RGB) on 7Scenes (averaged over all scenes). We attribute this slight difference to our use of measured depth maps in the initialization training stage, which are more noisy and contain holes as well as large areas of invalid depth compared to the rendered ground truth scene coordinates used in [4].

2. Visual Comparisons of pGT

Visualisation of pGT reference poses. We plot depth-based SLAM pGT versus RGB-based SfM pGT for the Pumpkin scene of 7Scenes in Fig. 1. For this scene, we observe the largest visual drift between both versions of the pGT. We also show estimated camera trajectories for Active Search, DSAC* and DSAC* (+D), the top-performing methods depending on the pGT version, for both versions of the pGT. While the depth-based SLAM pGT on this scene seems to have defects that make it hard for all re-localization methods to follow the ground truth trajectory, results look smoother for the SfM pGT. Still, both DSAC* re-localizers fail to follow the SfM pGT exactly, exhibiting small, consistent offsets w.r.t. the pseudo ground truth trajectory. We observe similar, yet less pronounced, patterns for other scenes of 7Scenes, *c.f.* Fig. 2. Visual differences between the pGT versions are smaller for the 12Scenes dataset, see Fig. 3, but still noticeable.

Visualisation of SfM point clouds. In addition to the SfM point clouds for the Red Kitchen and apt2/kitchen scenes shown in Fig. 3 of the main paper, we show point cloud visualisations for all scenes. In contrast to the main paper, we also show the pGT poses as red dots.

Fig. 4 and Fig. 5 show the 7 scenes of the 7Scenes dataset. Fig. 6 to Fig. 8 show the 12 scenes of the 12Scenes dataset. As already seen in the main paper, the SfM pseudo ground truth results in less noisy point clouds.

3. Quantitative Comparisons of pGT

Fig. 4 in the main paper shows cumulative distributions over 3D alignment statistics for the 7Scenes and 12Scenes datasets (*c.f.* Sec. 4, "Evaluation based on 3D alignment metrics" in the main paper for details). While Fig. 4 in the main papers shows average statistics over all scenes in each dataset and one selected scene per dataset. In this document, we show the distributions for all scenes of the two datasets.

Fig. 9 shows the cumulative distributions for all scenes of the 7Scenes dataset. As can be seen, the original (RGB-)D SLAM pGT results in a more accurate alignment for most scenes compared to the SfM pGT. For the Red Kitchen and

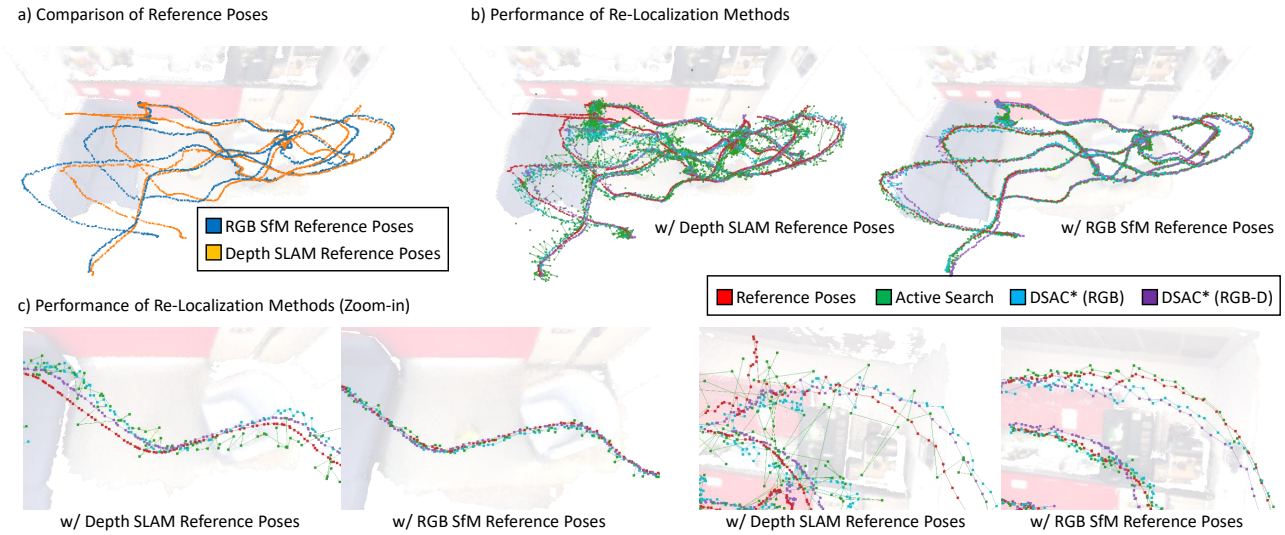

Figure 1. **Comparison of pGT on 7Scenes Pumpkin. a)** SfM-based reference poses (blue) versus SLAM-based reference poses (orange). **b)** We show estimated camera positions for Active Search (green), DSAC* w/ RGB inputs (cyan) and DSAC* w/ RGB-D inputs (purple). The respective reference poses, SLAM pGT on the left and SfM pGT on the right, are shown in red. We show close-up views in **c)**.

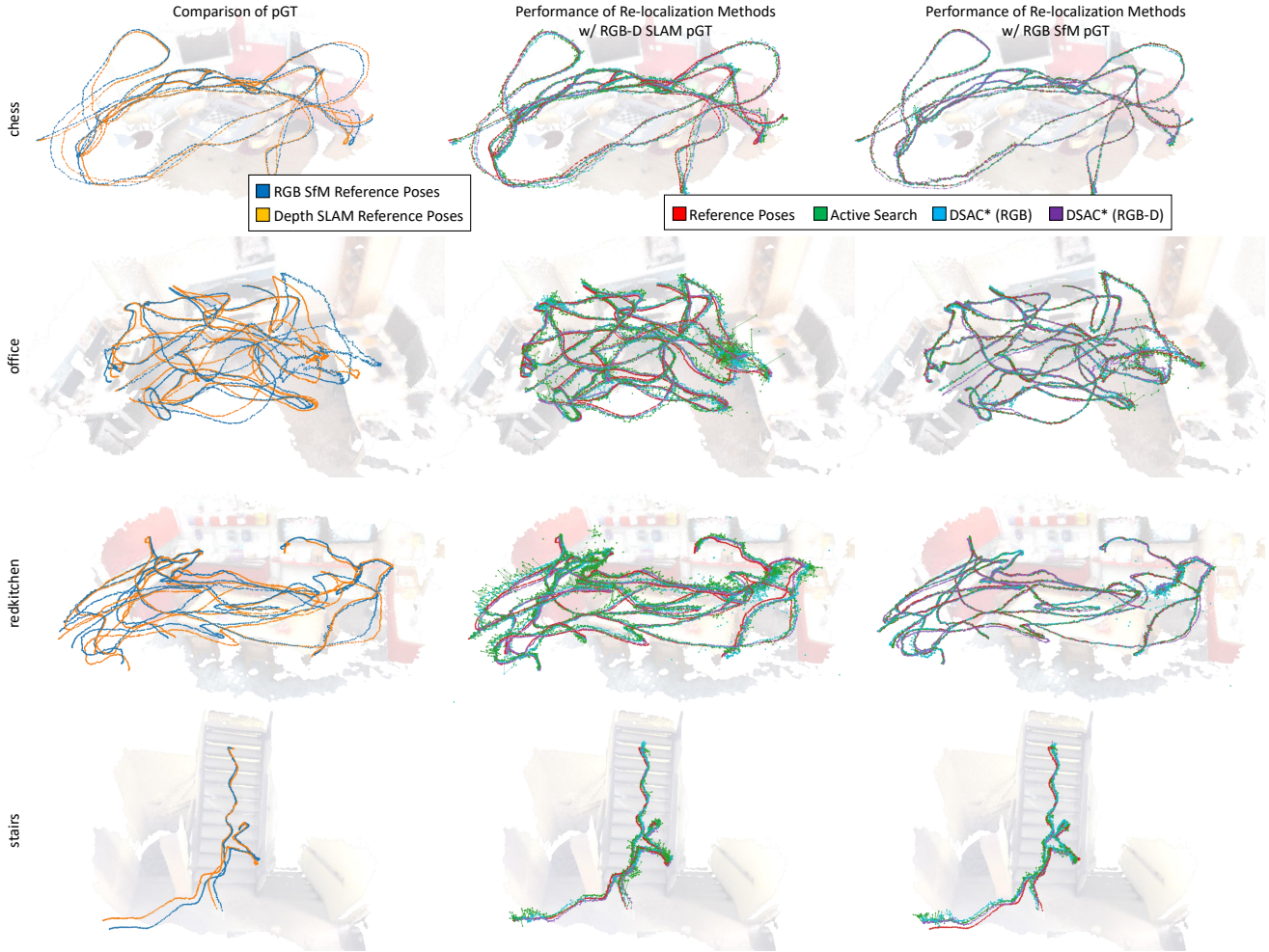

Figure 2. **Comparison of pGT on 7Scenes. Left:** Comparison of the pGT reference poses. **Middle:** Re-localization results based on the SLAM pGT. **Right:** Re-localization results based on the SfM pGT.

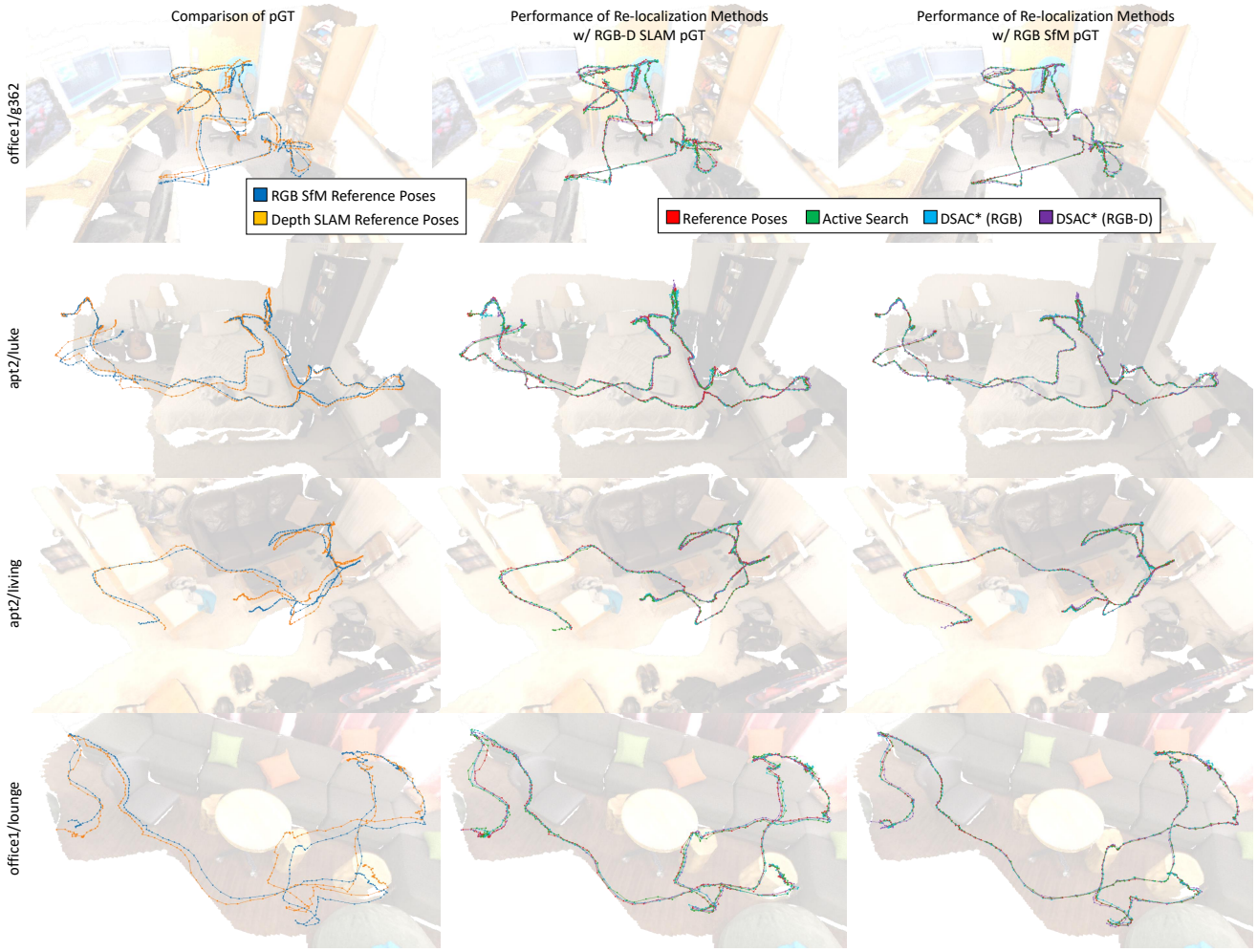

Figure 3. **Comparison of pGT on 7Scenes. Left:** Comparison of the pGT reference poses. **Middle:** Re-localization results based on the SLAM pGT. **Right:** Re-localization results based on the SfM pGT.

Stairs scenes, there is little difference between the two versions of the pGT and the SfM pGT produces a (slightly) more accurate alignment for the test/train pairs.

Similarly, Fig. 10 shows the cumulative distributions for all scenes of the 12Scenes dataset. Again, we observe that the original (RGB-)D SLAM pGT results in more accurate 3D alignments compared to the SfM pGT. However, for most scenes, the difference between both versions of the pGT is smaller than for the 7Scenes dataset.

4. Visual Re-Localization Evaluation

As an extension to Fig. 7(a) of the main paper, we show cumulative pose error plots and cumulative DCRE [19] error plots for all scenes of 7Scenes and 12Scenes, separately. See Fig. 11, Fig. 12 and Fig. 13 for pose error plots, max. DCRE error plots and mean DCRE error plots, respectively, for 7Scenes. We show the corresponding plots for 12Scenes in Fig. 14, Fig. 15 and Fig. 16.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org. 1

[2] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 1

[3] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1

[4] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *TPAMI*, 2021. 1, 2

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 1

[6] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust

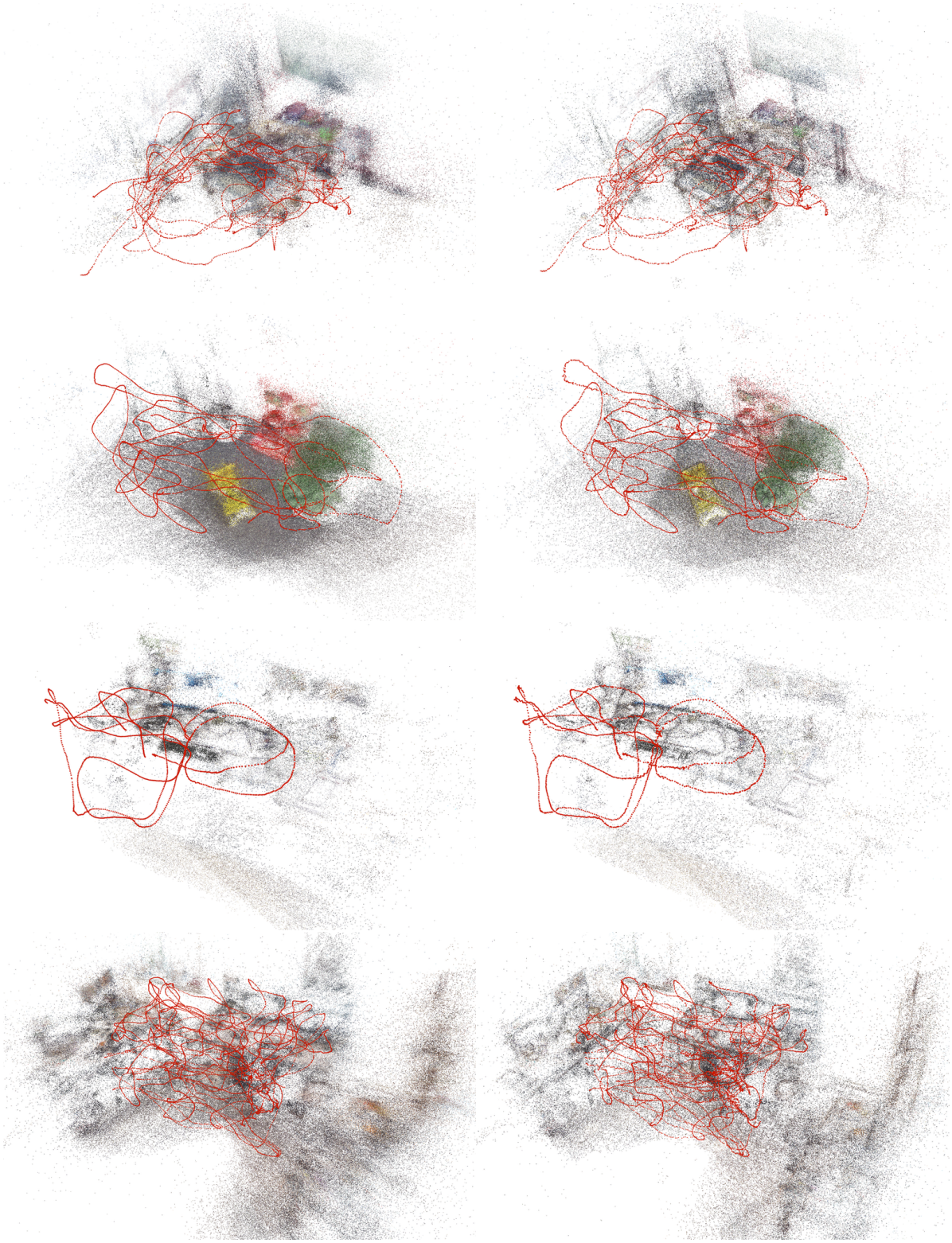

Figure 4. **Visualisation of SfM point clouds for the 7Scenes [16] dataset.** The left column shows the Structure-from-Motion (SfM) point clouds obtained by triangulating the 3D scene structure using the original pGT. The right column shows the SfM point clouds of the SfM pGT. The red dots denote the pGT camera positions of the test and training images. From top to bottom: Chess, Fire, Heads, Office.

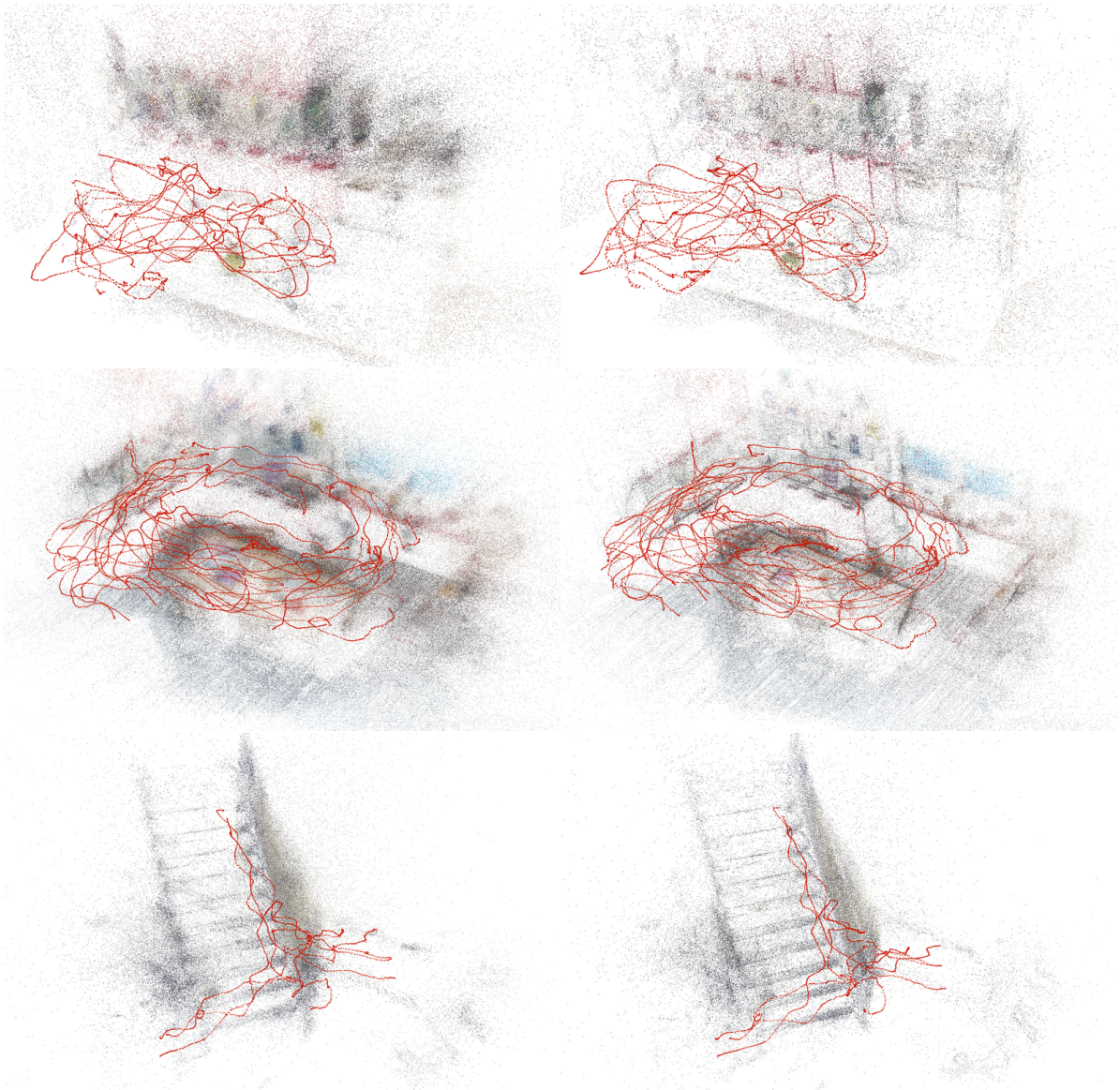

Figure 5. **Visualisation of SfM point clouds for the 7Scenes [16] dataset.** The left column shows the Structure-from-Motion (SfM) point clouds obtained by triangulating the 3D scene structure using the original pGT. The right column shows the SfM point clouds of the SfM pGT. The red dots denote the pGT camera positions of the test and training images. From top to bottom: Pumpkin, Red Kitchen, Stairs.

Image Retrieval-based Visual Localization using Kapture. arXiv:2007.13867, 2020. 1, 2

[7] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010. 1

[8] Karel Lebeda, Juan E. Sala Matas, and Ondřej Chum. Fixing the Locally Optimized RANSAC. In *British Machine Vision Conference (BMVC)*, 2012. 1

[9] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 1

[10] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 2

[11] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 1

[12] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020. 1

[13] Torsten Sattler et al. RansacLib - A Template-based *SAC Implementation, 2019. 1

[14] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 39(9):1744–1756, 2017. 1

[15] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval. In *ACCV*, 2016. 1

Figure 6. **Visualisation of SfM point clouds for the 12Scenes [18] dataset.** The left column shows the Structure-from-Motion (SfM) point clouds obtained by triangulating the 3D scene structure using the original pGT. The right column shows the SfM point clouds of the SfM pGT. The red dots denote the pGT camera positions of the test and training images. From top to bottom: apt1/kitchen, apt1/living, apt2/bed, apt2/kitchen.

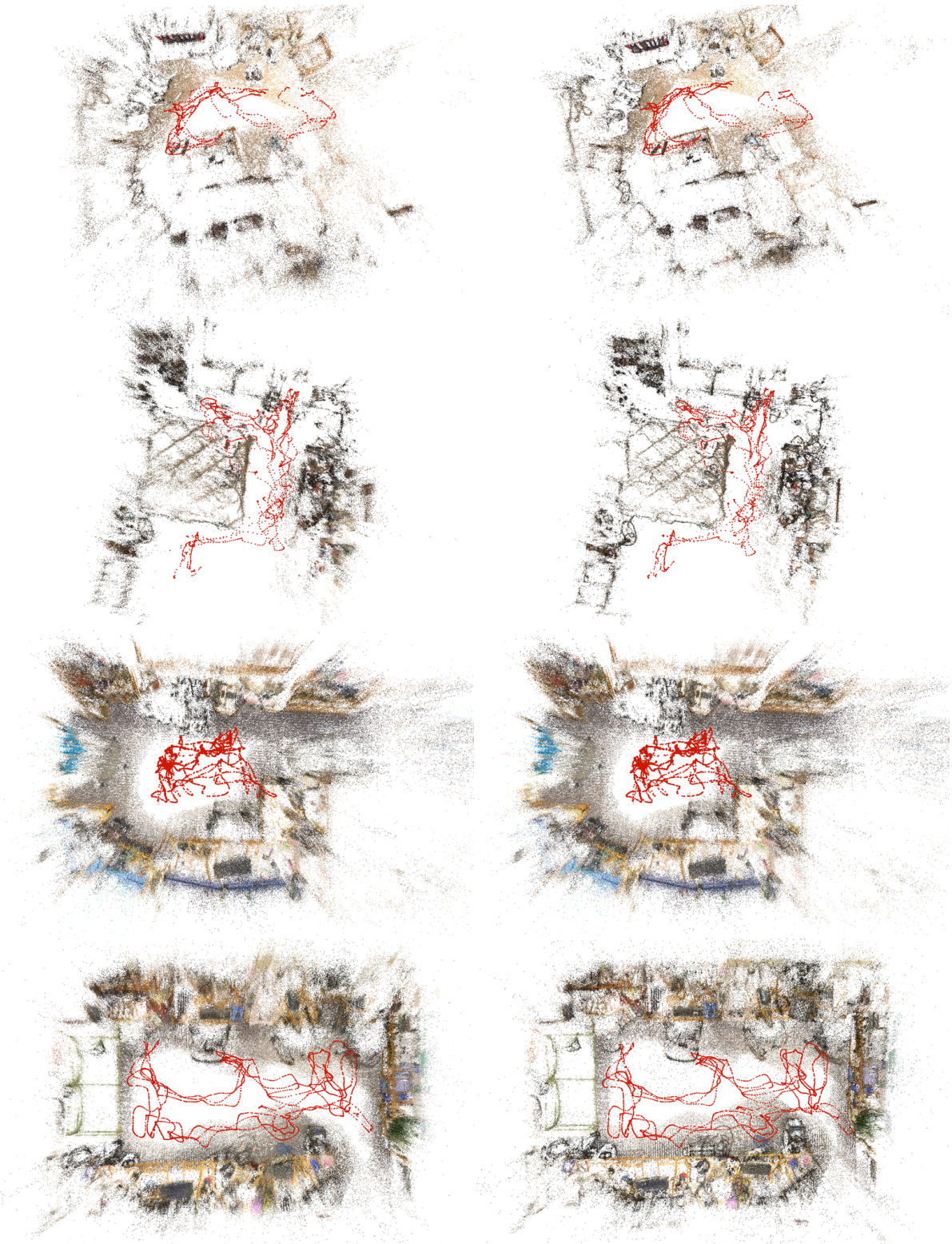

Figure 7. **Visualisation of SfM point clouds for the 12Scenes [18] dataset.** The left column shows the Structure-from-Motion (SfM) point clouds obtained by triangulating the 3D scene structure using the original pGT. The right column shows the SfM point clouds of the SfM pGT. The red dots denote the pGT camera positions of the test and training images. From top to bottom: apt2/living, apt2/luke, office1/gates362, office1/gates382.

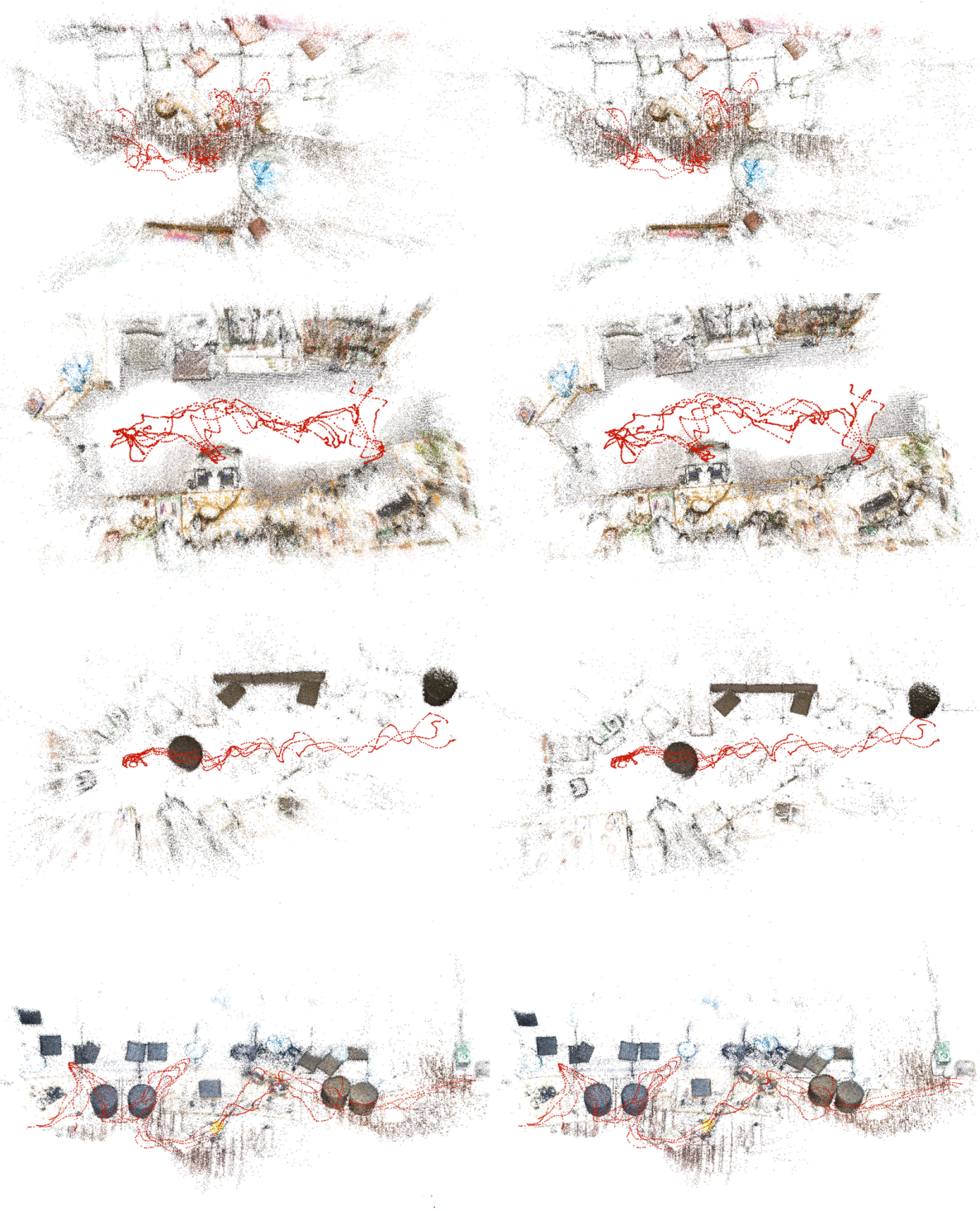

Figure 8. **Visualisation of SfM point clouds for the 12Scenes [18] dataset.** The left column shows the Structure-from-Motion (SfM) point clouds obtained by triangulating the 3D scene structure using the original pGT. The right column shows the SfM point clouds of the SfM pGT. The red dots denote the pGT camera positions of the test and training images. From top to bottom: office1/lounge, office1/manolis, office2/5a, office2/5b.

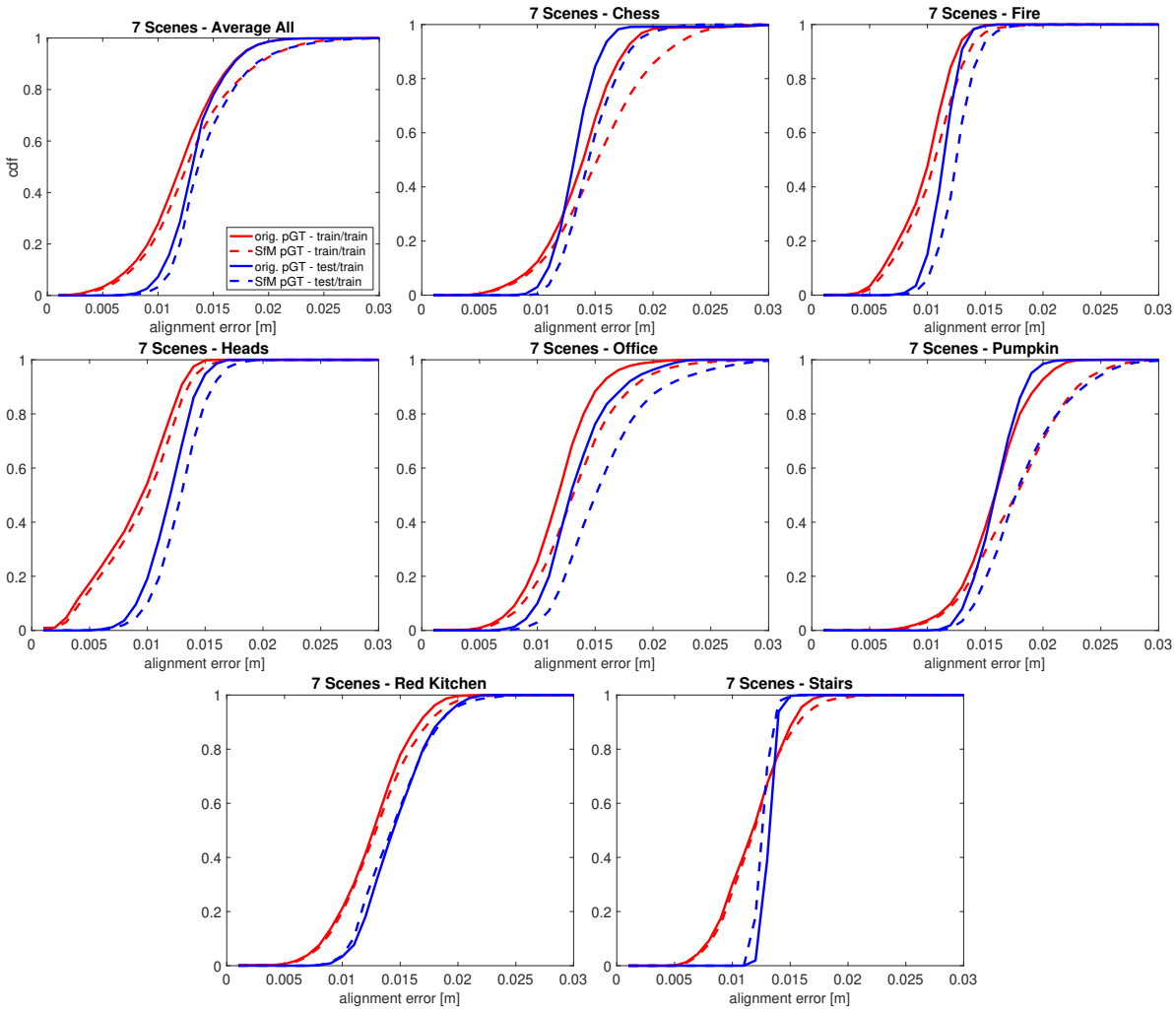

Figure 9. **3D alignment statistics for the 7Scenes [16] dataset.** We show cumulative distributions (cdfs) of the 3D alignment errors between the depth maps of train/train and test/train image pairs with a visual overlap of at least 30% for the original (RGB-)D SLAM and the SfM pseudo GT.

[16] 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. 1, 5, 6, 10, 12, 13

[17] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 1, 2

[18] Julien Valentin, Angela Dai, Matthias Niessner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to Navigate the Energy Landscape. In *3DV*, 2016. 1, 7, 8, 9, 11, 14, 15, 16

[19] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond Controlled Environments: 3D Camera Re-Localization in Changing Indoor Scenes. In *ECCV*, 2020. 4, 12, 13, 15, 16

[20] Christian Wolf, Eric Lombardi, Julien Mille, Oya Celiktutan, Mingyuan Jiu, Emre Dogan, Gonen Eren, Moez Baccouche, Emmanuel Dellandréa, Charles-Edmond Bichot, Christophe Garcia, and Bülent Sankur. Evaluation of video activity localizations integrating quality and quantity measurements. *Comput. Vis. Image Underst.*, 127:14–30, 2014. 2

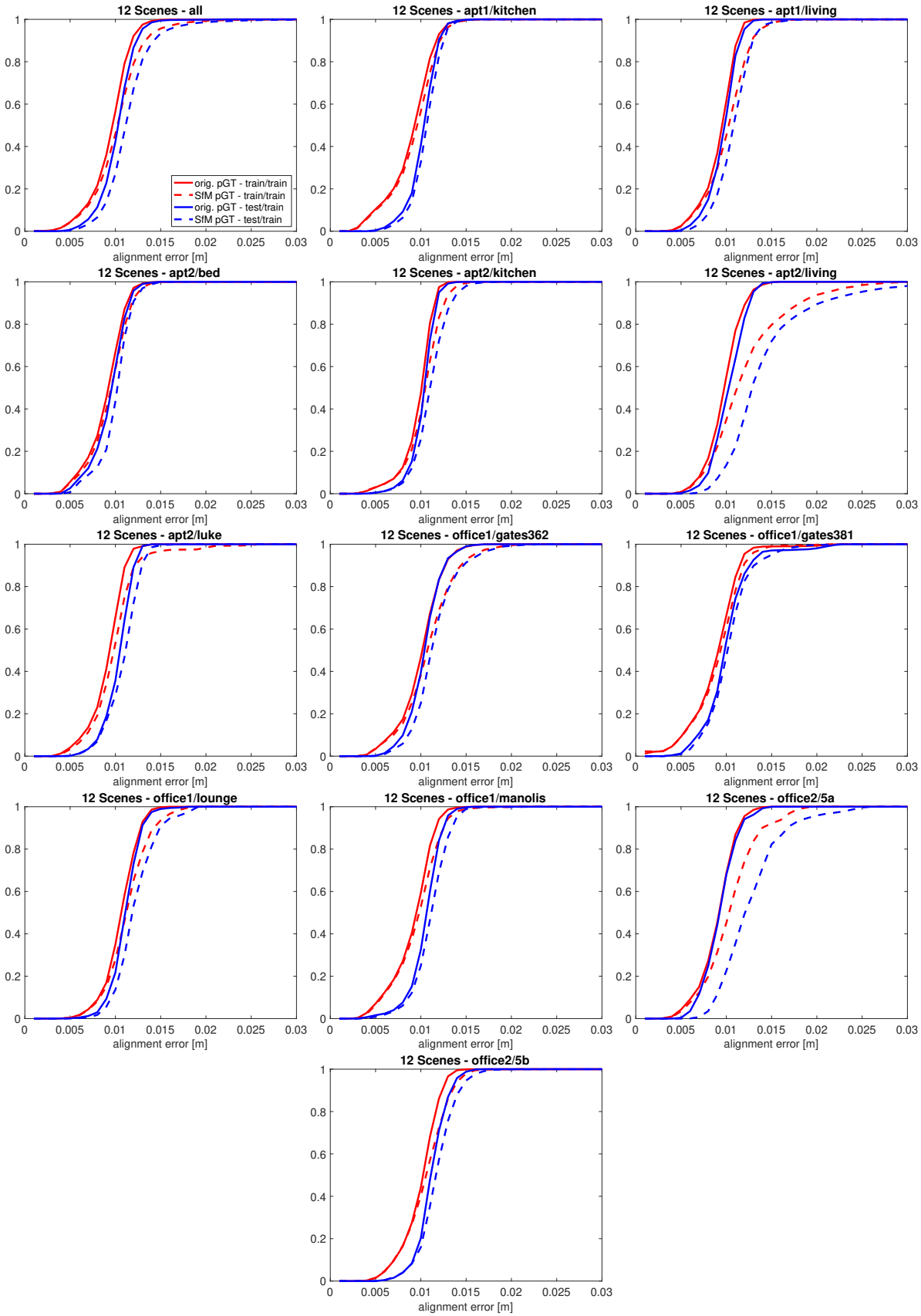

Figure 10. **3D alignment statistics for the 12Scenes [18] dataset.** We show cumulative distributions (cdfs) of the 3D alignment errors between the depth maps of train/train and test/train image pairs with a visual overlap of at least 30% for the original (RGB-)D SLAM and the SfM pseudo GT.

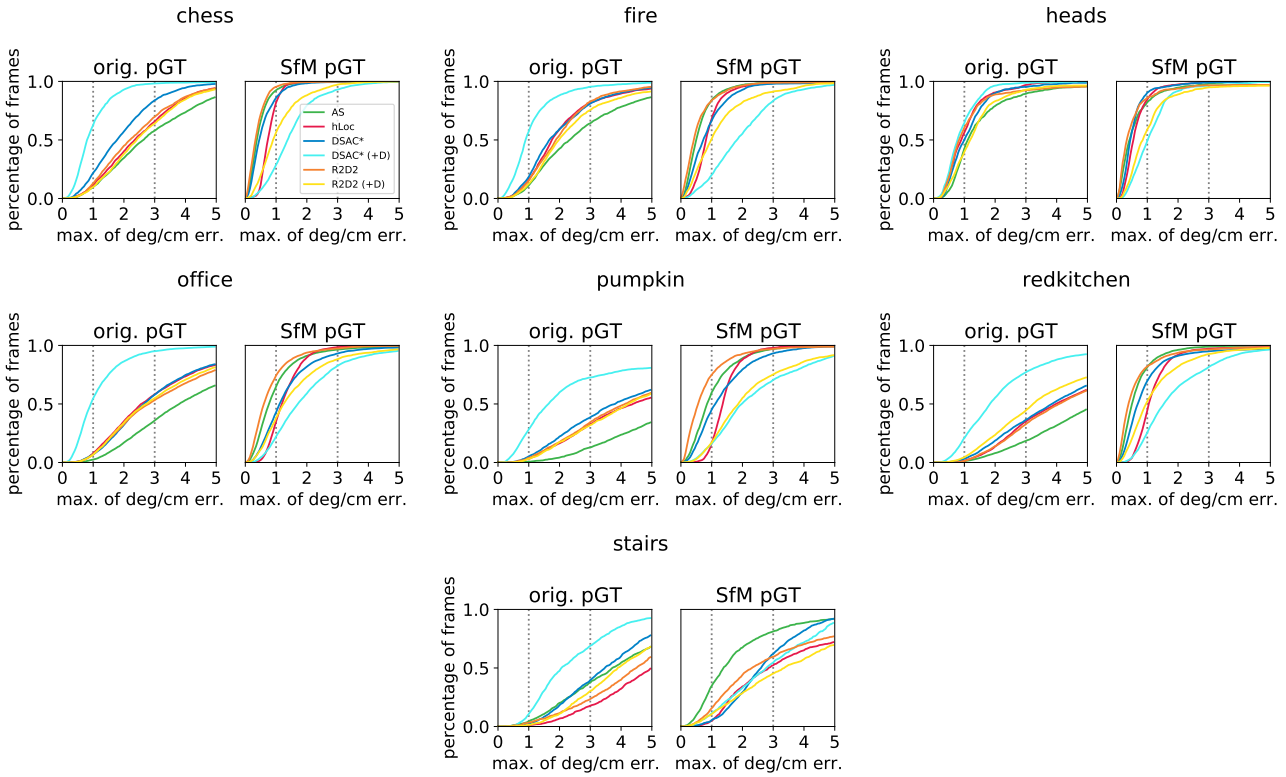

Figure 11. **Pose error for 7Scenes.** We show cum. distributions of the pose error (max. of rotation and translation error) for all scenes of 7Scenes [16]. Dotted vertical lines correspond to 1cm,1° and 3cm,3° thresholds for reference.

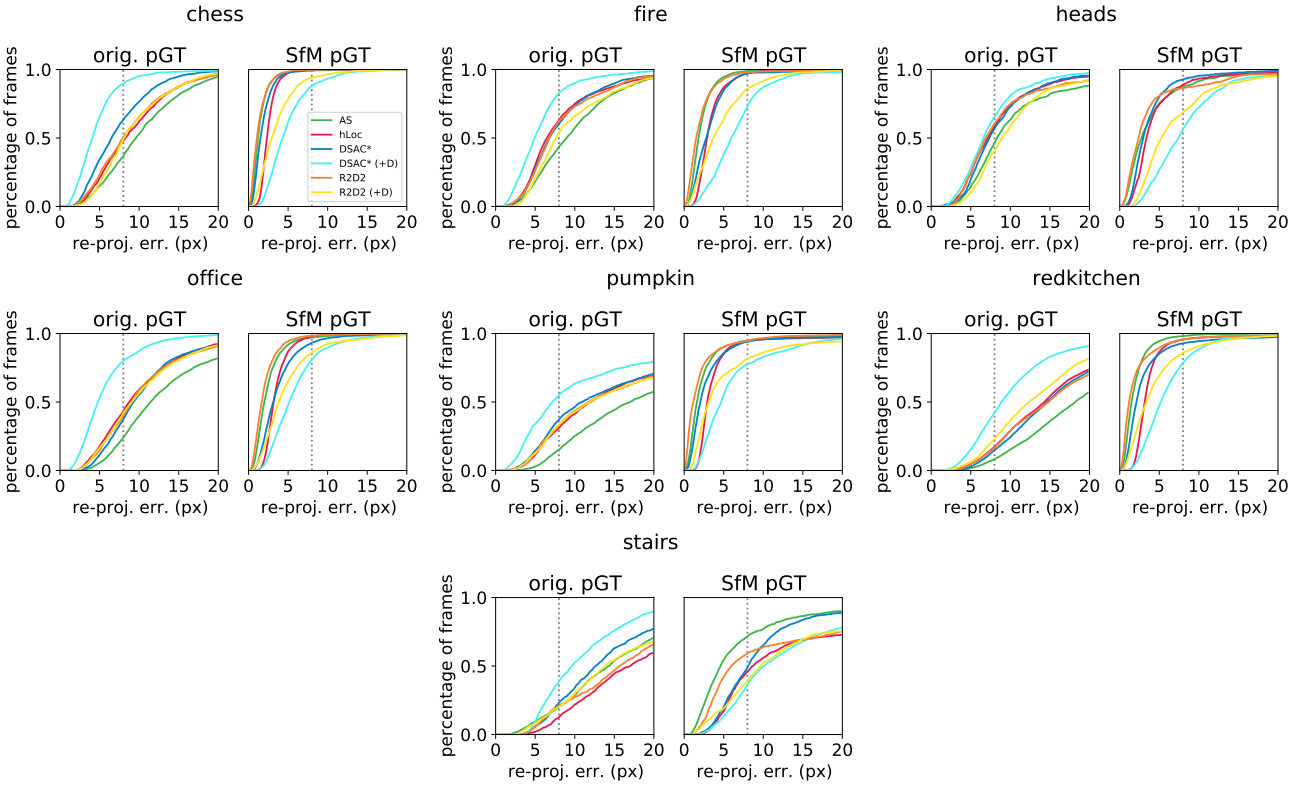

Figure 12. **Max. DCRE for 7Scenes.** We show cum. distributions of the DCRE (Dense Correspondence Re-Projection Error [19]) for all scenes of 7Scenes [16], taking the max. re-projection error per test image. The dotted line corresponds to 1% of the image diagonal.

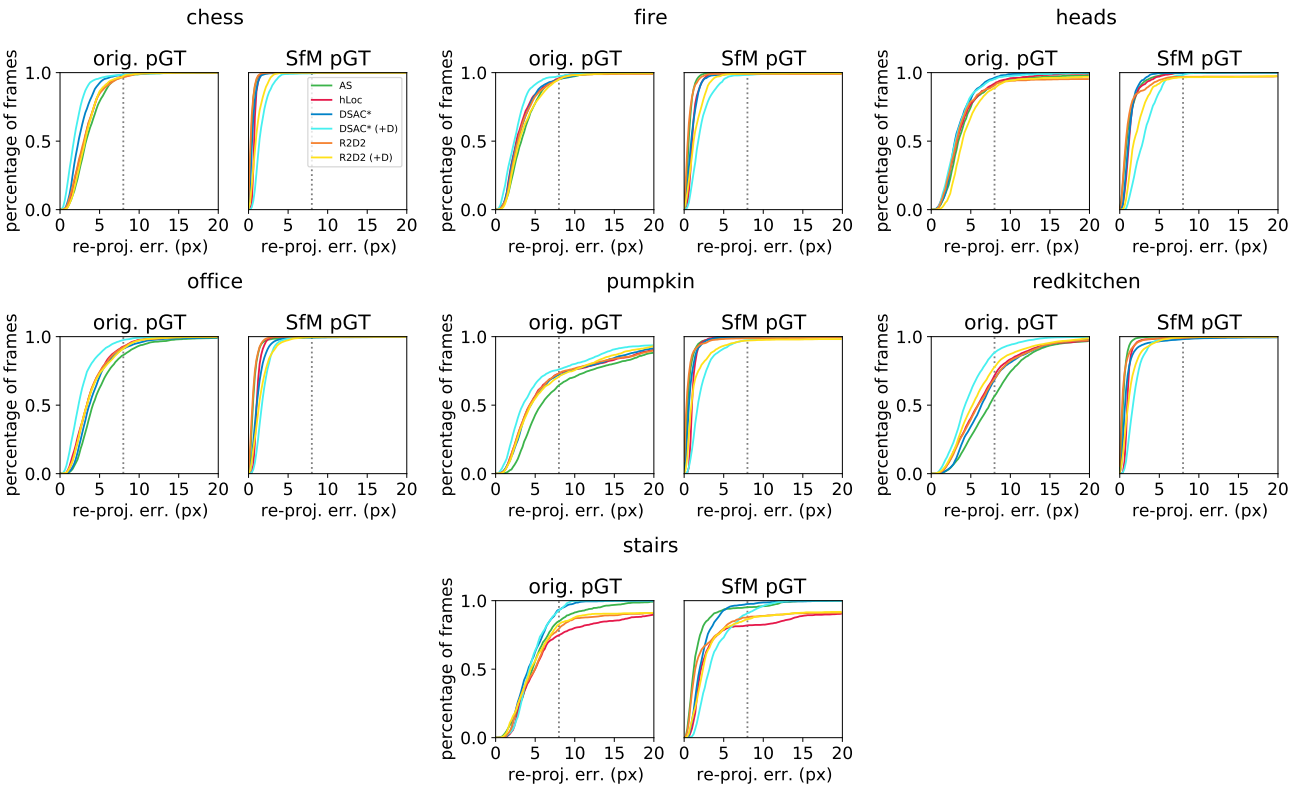

Figure 13. **Mean DCRE for 7Scenes.** We show cum. distributions of the DCRE (Dense Correspondence Re-Projection Error [19]) for all scenes of 7Scenes [16], taking the mean re-projection error per test image. The dotted line corresponds to 1% of the image diagonal.

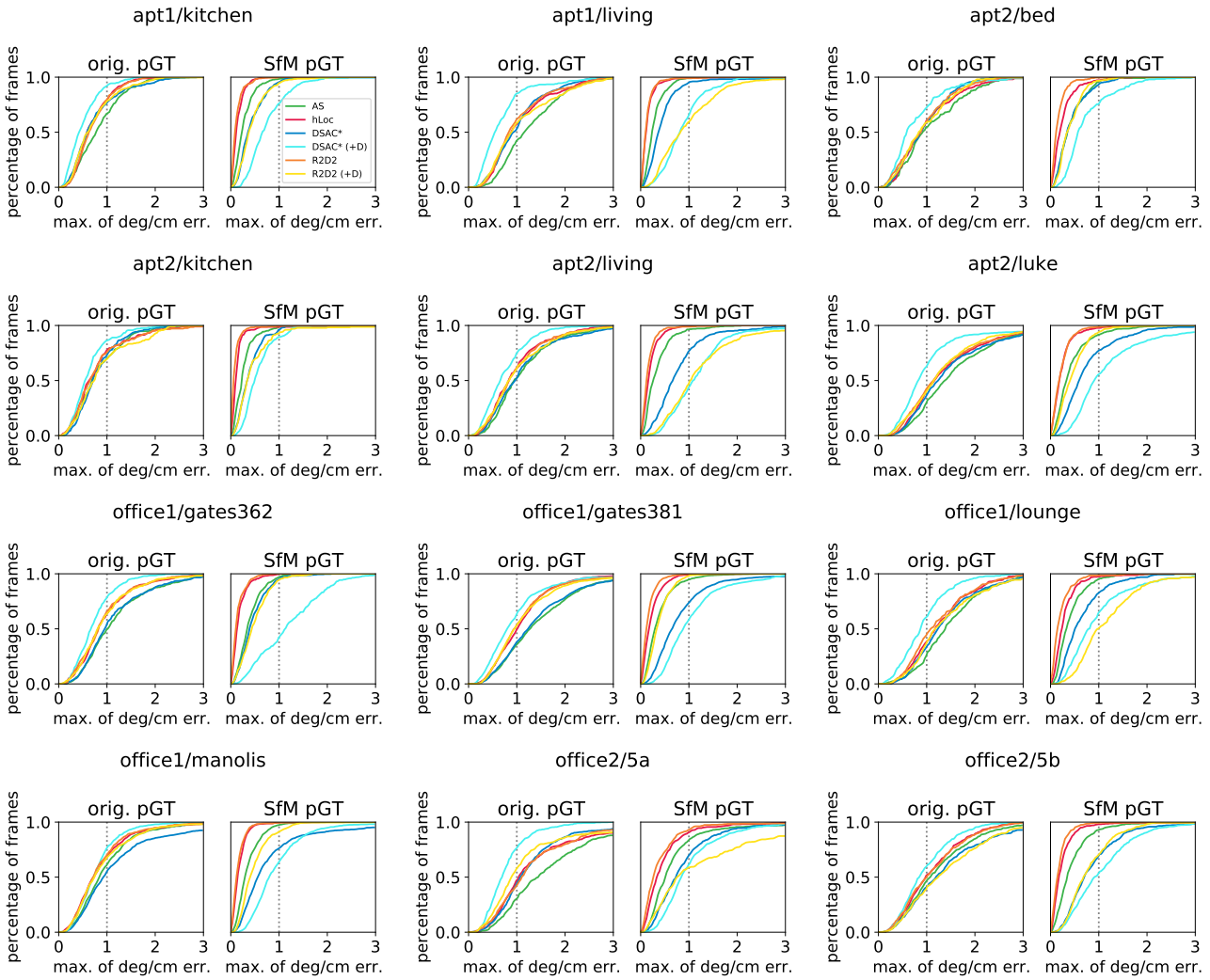

Figure 14. **Pose error for 12Scenes.** We show cum. distributions of the pose error (max. of rotation and translation error) for all scenes of 12Scenes [18]. Dotted vertical lines correspond to a 1cm,1° threshold for reference.

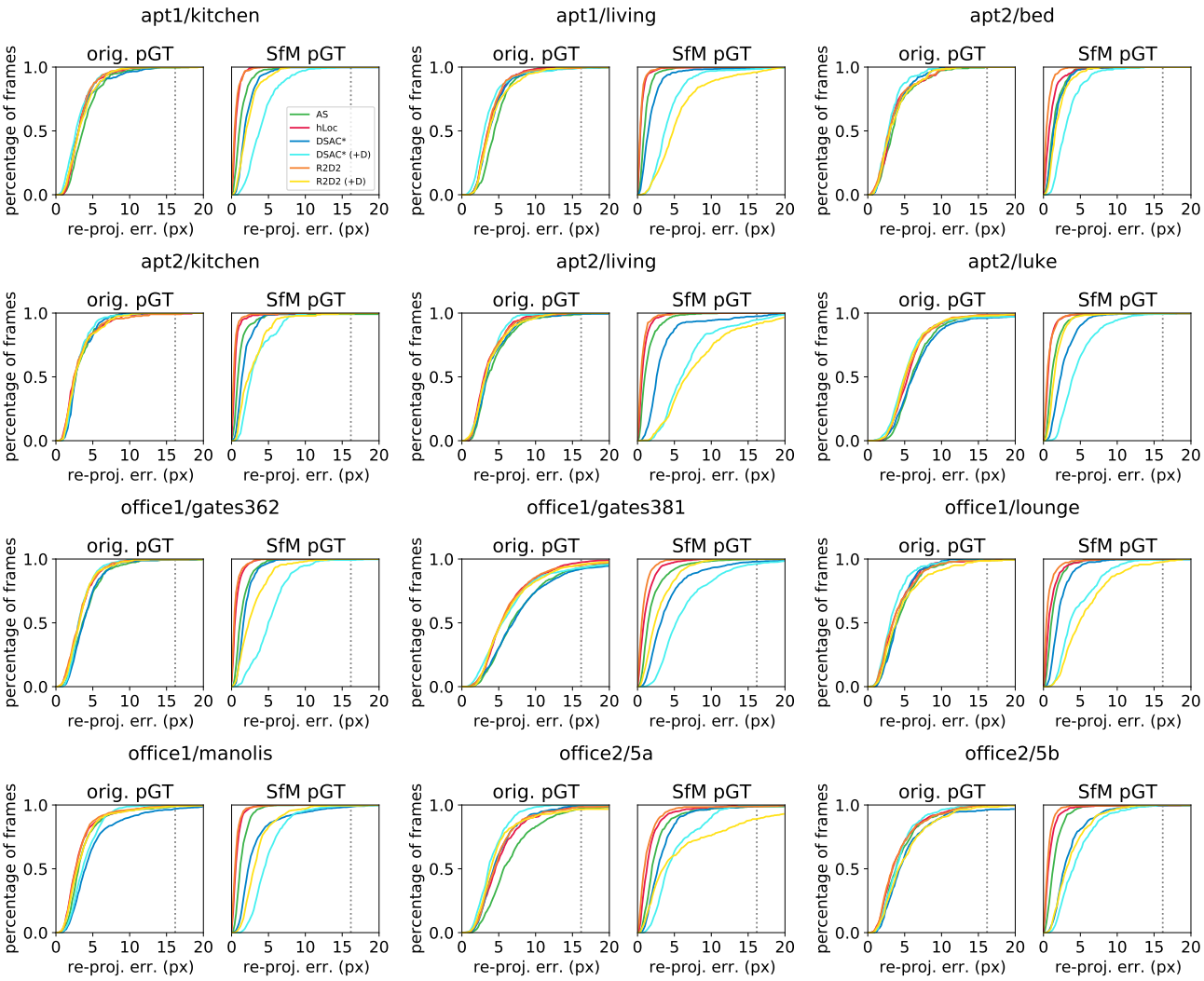

Figure 15. **Max. DCRE for 12Scenes.** We show cum. distributions of the DCRE (Dense Correspondence Re-Projection Error [19]) for all scenes of 12Scenes [18], taking the max. re-projection error per test image. The dotted vertical line corresponds to 1% of the image diagonal.

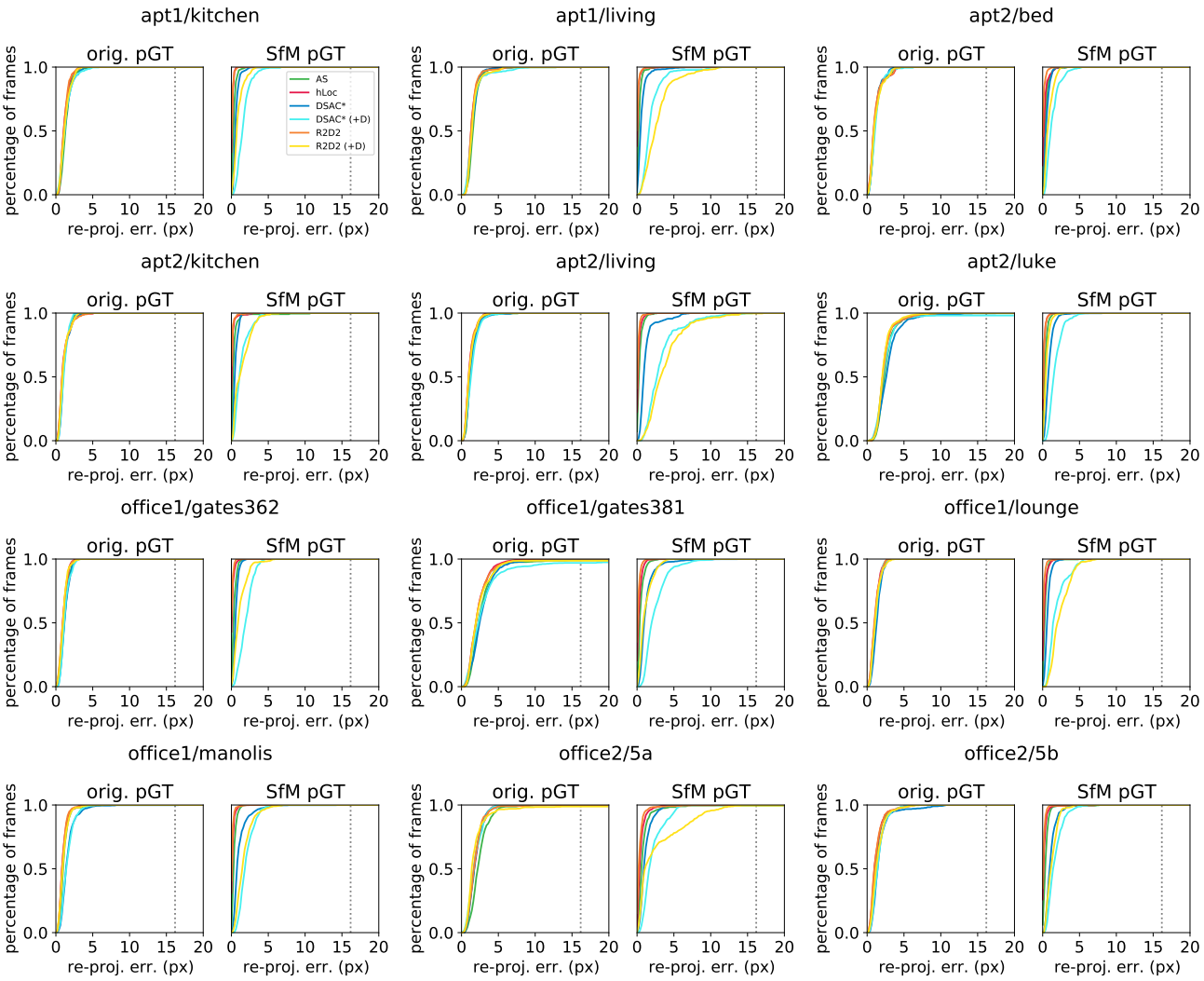

Figure 16. **Mean DCRE for 12Scenes.** We show cum. distributions of the DCRE (Dense Correspondence Re-Projection Error [19]) for all scenes of 12Scenes [18], taking the mean re-projection error per test image. The dotted vertical line corresponds to 1% of the image diagonal.