Scene Coordinate Reconstruction: Posing of Image Collections via Incremental Learning of a Relocalizer – Supplementary Material –

Eric Brachmann[1], Jamie Wynn[1], Shuai Chen[2], Tommaso Cavallari[1], Áron Monszpart[1], Daniyar Turmukhambetov[1], and Victor Adrian Prisacariu[1,2]

[1] Niantic
[2] University of Oxford

1 Implementation

Reconstruction Loop. We start the reconstruction as explained in Sec. 3.3 of the main paper. We repeat neural mapping and registration of new views until all views have been registered, or less than 1% of additional views are registered compared to the last relocalization stage. In each relocalization iteration, we re-estimate the poses of all views, even if they have been registered before. We also tried a variant that skips estimating poses that have been registered before. But this version achieved slightly lower quality while not giving a significant speed advantage. Re-estimating all poses gives the pipeline the ability to reduce inaccuracies and to correct previous outlier estimates. We consider an image to be registered successfully, if it has at least 500 inlier correspondences with an inlier threshold of 10px. At the end of the reconstruction, we apply a more conservative threshold of 1000 inliers to make sure only high quality poses are kept. For each neural mapping iteration, we load the network weights of the scene coordinate regression network of the previous mapping iteration and refine further. We conclude the reconstruction loop with a final refit of the model, using the standard ACE training parameters. In this last iteration, we start with a randomly initialised network, rather than loading previous weights.

Neural Mapping. The ACE backbone predicts 512-dimensional features densely for the input image, but sub-sampled by a factor of 8. Training images are augmented using random scaling and rotation as well as brightness and contrast changes as in ACE [3]. When filling the training buffer, we randomly choose 1024 features per training image, and we sample features from each training image at most 10 times, *i.e.* for 10 different augmentations. During training, we use a batch size of 5120 features. Our pose refinement MLP has 6 layers with 128 channels and a skip connection between layer 1 and 3. We optimize the pose refinement MLP, and the relative scaling factor for the focal length, with separate AdamW [10] instances and a learning rate of 10^{-3}. In the seed iteration, we optimize neither the seed pose (which is identity) nor the initial calibration parameters.

ACE uses a soft-clamped version of the reprojection error in Eq. 3 (main paper) with a soft threshold that reduces from 50px to 1px throughout training. We use the same, soft-clamped loss but keep the threshold fixed at 50px as we saw no benefit when using the loss schedule but it complicates early stopping. During the seed iteration, we switch from optimizing the Euclidean distance to optimizing the reprojection error if the reprojection error of a scene coordinate is lower than 10px. This hybrid objective is inspired by DSAC* [5].

When starting neural mapping from a set of initial poses, *e.g.* from KinectFusion, a sparse COLMAP reconstruction or in the final model refit of the reconstruction loop, we keep the pose refinement MLP frozen for the first 5000 iterations. The refinement MLP cannot predict sensible updates for an uninitialised scene representation, and the reconstruction would become unstable. Note that this standby time is not required in all but the last training iteration, as we initialise the scene coordinate regressor using the weights of the last iteration, so the pose refiner can predict sensible updates right away.

Relocalization. For registering images to the scene reconstruction, we utilize the pose optimization of ACE [3]. It consists of a PnP solver within a RANSAC loop. ACE samples 64 pose hypotheses per image. If sampling a hypothesis fails, ACE tries again, up to 1M times. After sampling, ACE ranks hypotheses according to inlier counts using a reprojection error threshold of 10px. The best hypothesis is refined by iteratively resolving PnP on the growing inlier set.

Usually, this procedure is fast, taking approximately 30ms per image. However, when the predicted image-to-scene correspondences are bad, sampling hypothesis can take a long time due to the large number of re-tries. For ACE0, this will happen often, namely when we try to register images that are not yet covered by the reconstruction. Thus, we decrease the number of re-tries to from 1M to 16. When registering images throughout the reconstruction, we also decrease the number of hypotheses drawn from 64 to 32. This yields registration times of 20ms per image, irrespective of whether the registration succeeds or fails.

Runtime Complexity. The reconstruction time depends on the number of images but also on the spatial distribution of cameras. In the worst case, complexity is $O(n^2)$. *E.g.* a long camera trajectory without intersections or loops would require proportional to n ACE0 iterations, with n relocalizations in each step, giving $O(n^2)$. The best case complexity is $\Omega(n)$, *e.g.* for a forward facing scene where all images can be registered in 1-2 ACE0 iterations independent of the number of images. In practise, we observe attractive run times of ACE0.

2 Benchmark

We benchmark the quality of each set of poses by training a NeRF model on the train views and evaluating on the test views. We use Nerfstudio's Nerfacto model for this [14]. Where datasets already include a train/test split – as in 7-Scenes – we use that split. Where a dataset does not already include a train/test split,

we take every eighth view as a test view, which is a common choice for many NeRF datasets.

Many scenes in this work have large test sets of several thousand images or more. This can lead to long evaluation times due to the cost of rendering thousands of test views with NeRF – much longer than the time to fit the NeRF itself. For this reason we subsample very large test sets so that they contain no more than 2k views. To ensure that this reduced test set is representative, we do this by subsampling evenly throughout the full-size test set in a deterministic manner. The code to compute the resulting splits will be included in our code release.

We use Nerfacto's default normalization of poses to fit into a unit cube, and enable scene contraction so that distant elements of the scene can be modelled more accurately. We train each scene for 30k iterations. For NeRF training and evaluation we downscale the resolution so that the longest side of an image is not greater than 640 pixels, as NeRFs often perform poorly with high-resolution training images.

Where a pose for a test view is not provided by a given method, we use the identity pose for that view. This results in a very low PSNR for that image, penalising the method for its failure to provide a pose. This means that we evaluate all methods on the full set of test views, even where they do not provide poses for all such views. If a reconstruction results in multiple disconnected components, we use the largest component in our benchmark and consider the other frames as missing. This happens rarely for COLMAP, but more often for RealityCapture (*c.f.* Sec. 4).

ACE0 pose estimates are generally available for all images, but we consider an image registered if its confidence exceeds the threshold of 1000 inliers. However, when evaluating test images we always take the estimated pose instead of the identity pose as basis for the evaluation, even if the confidence is lower than 1000. Note that the question of whether a frame is considered "registered" or not is still important when fitting the NeRF, as such images are excluded from NeRF training. See Sec. 3 for further information about registration rates for different methods.

Relocalization. In the following, we give more information about the relocalization experiment of Sec. 4.1 and Table 2(a) of the main paper. We conducted the experiment on the 7-Scenes dataset adhering to its training / test split. Our experiments utilize the pseudo ground truth (pGT) poses of KinectFusion that come with the dataset, as well as the COLMAP pGT of [4]. The authors of [4] obtained the COLMAP pGT by reconstructing the training images of each scene with COLMAP from scratch. To make the pGT scale-metric, they aligned the COLMAP reconstruction with the KinectFusion poses. To obtain poses for the test images of each scene as well, they registered the test images to the training reconstruction. Lastly, they ran a final round of bundle adjustment to refine the test poses while keeping the training poses fixed.

In our experiment, we train the ACE relocalizer on the training images of each scene. Firstly, we train ACE using the KinectFusion poses that come with the

dataset. Secondly, we train ACE using the COLMAP pGT of [4]. We evaluate both version of the relocalizer using the COLMAP pGT. As expected, ACE trained with KinectFusion poses scores lower than ACE trained with COLMAP pGT, when evaluating against COLMAP pGT. This confirms the finding of [4] that KinectFusion and COLMAP produce different pose estimates.

Next, we run ACE0 on the training set of each scene to reconstruct it and estimate the training poses. One output of ACE0 is an ACE relocalizer, trained from scratch without poses, in a self-supervised fashion. We apply the ACE0 relocalizer to the test images to estimate their poses - without any further model refinement or joint optimization. The ACE0 reconstruction is scale-less, or only roughly scaled based on the ZoeDepth estimate of the seed image. To enable the scale-metric evaluation of Table 2(a) of the main paper, we align the ACE0 pose estimates to the test pseudo ground truth by fitting a similarity transform. Specifically, we run RANSAC [6] with a Kabsch [7] solver on the estimated and the ground truth camera positions with 1000 iterations. Due to the optimization of the similarity transform, we might underestimate the true relocalization error. For a fair comparison, we perform the exact same evaluation procedure for our ACE baselines and fit a similarity transform although they have already been trained with scale-metric poses. The effect of fitting the similarity transform is small. ACE, trained on COLMAP pGT, achieves 97.1% average relocalization accuracy when we fit the similarity transform, and 97.6% average accuracy without fitting the transform. We compute accuracy as the percentage of test images where the pose error is smaller than 5cm and 5° compared to ground truth. The ACE0 relocalizer achieves 93.8% average accuracy when evaluated against the COLMAP pGT. The accuracy differences are smaller than 1% for all scenes except Stairs. This experiment shows that ACE0 is a viable self-supervised relocalizer, and that ACE0 poses are very close to COLMAP poses up to a similarity transform - much closer than *e.g.* KinectFusion poses are to COLMAP poses.

3 Additional Results

More Qualitative Results. We show qualitative results in terms of estimated poses for more scenes in Fig. 1. We show failure cases in Fig. 2 corresponding to some of the limitations discussed in Section 5 of the main paper.

Additionally, we show synthesized views based on ACE0 poses. Corresponding to our quantitative results in the main paper (Table 1, Table 2 (b), and Table 3 of the main paper), we select one test image per scene that is representative of the view synthesis quality of ACE0 on that scene. More specifically, we select the test image where ACE0 achieves its median PSNR value. For comparison, we synthesise the same image using the estimated poses of our competitors. For synthesised images of 7-Scenes, see Fig. 3. For comparison with BARF and NoPe-NeRF, the main paper reports results on a 200 image subset per scene. We show the associated synthesized images in Fig. 4. For comparison with DUSt3R, the main paper reports results on a 50 image subset per scene. We show the associated synthesized images in Fig. 5. We show synthesized images for the

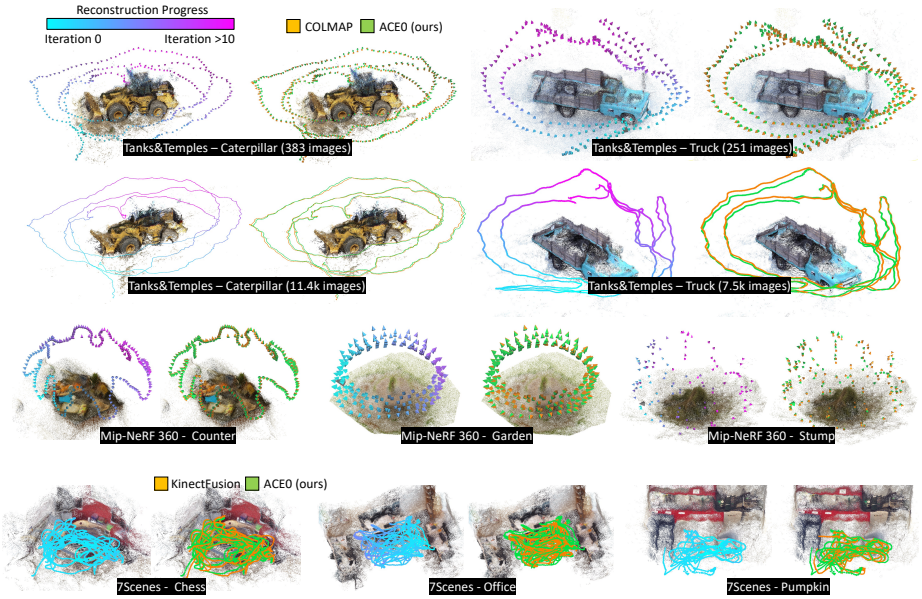

Fig. 1: More Reconstructed Poses. We show poses estimated by ACE0. We color code the reconstruction iteration in which a particular view has been registered. We show the ACE0 point cloud as a representation of the scene. We also compare our poses to poses estimated by COLMAP (Mip-NeRF 360, Tanks and Temples) and KinectFusion (7-Scenes).

MIP-NeRF 360 dataset in Fig. 6. We also show synthesized images for Tanks and Temples, in Fig. 7 for *Training* scenes, in Fig. 8 for *Intermediate* scenes, and in Fig. 9 for *Advanced* scenes.

More Quantitative Results. For completeness, we augment Table 1 in the main paper with SSIM [17] and LPIPS [18] scores for the 7-Scenes dataset, *c.f.*, Tables 1 and 2 respectively. Similarly, Tables 11 and 12 show SSIM [17] and LPIPS [18] scores for the Mip-NeRF 360 dataset, augmenting Table 2 (b) in the main paper. Furthermore, Tables 9 and 10 show SSIM [17] and LPIPS [18] scores for the Tanks and Temples dataset, augmenting Table 3 in the main paper. In all cases, SSIM and LPIPS behave very similar to PSNR in our experiments, and support the conclusions of the main paper.

Analysis. We show variations of ACE0 in Table 3 alongside PSNR numbers and reconstruction times for 7-Scenes.

ACE0 picks 5 random seed images to start the reconstruction. We show average statistics of ACE0 over five runs, starting off with different sets of seed images each time. The standard deviation is low with 0.2 dB in PSNR, and about 4 minutes in terms of reconstruction time.

	Frames	Pseudo Ground Truth		All Frames				200 Frames			50 Frames	
		Kinect Fusion	COLMAP (default)	COLMAP (fast)	DROID-SLAM† [15]	ACE0 (ours)	KF+ACE0 (ours)	BARF [8]	NoPE-NeRF† [2]	ACE0 (ours)	DUSt3R [16]	ACE0 (Ours)
Chess	6k	0.69	0.86	0.86	0.76	0.84	0.84	0.55	0.51	**0.82**	0.66	**0.70**
Fire	4k	0.55	0.73	0.73	0.44	0.69	0.70	0.45	0.41	**0.70**	0.55	**0.59**
Heads	2k	0.69	0.82	0.82	0.78	0.81	0.81	0.54	0.52	**0.80**	0.69	**0.79**
Office	10k	0.72	0.85	0.84	failed	0.83	0.83	0.59	0.55	**0.80**	0.57	**0.61**
Pumpkin	6k	0.71	0.84	0.84	0.69	0.83	0.82	0.75	0.65	**0.84**	0.76	**0.77**
RedKitchen	12k	0.61	0.80	0.80	0.51	0.77	0.78	0.52	0.50	**0.72**	0.54	**0.55**
Stairs	3k	0.63	0.59	0.77	0.47	0.57	0.70	0.65	0.66	**0.68**	0.54	**0.55**
Average		0.66	0.78	0.81	N/A	0.76	0.78	0.58	0.54	**0.76**	0.62	**0.65**
Avg. Time		realtime	38h	13h	18min	1h	7min	8.5h	47h	**27min**	**4min***	16min

Table 1: 7-Scenes - SSIM (↑). We show the pose accuracy via view synthesis with Nerfacto [14] as **SSIM** (↑) [17], and the reconstruction time. Results for *All Frames* are color coded w.r.t. similarity to the COLMAP pseudo ground truth: > 0.05 better, within ± 0.05, > 0.05 worse, > 0.1 worse. For some competitors, we had to sub-sample the images due to their computational complexity (right side). †Method needs sequential inputs. *Results on more powerful hardware.

	Frames	Pseudo Ground Truth		All Frames				200 Frames			50 Frames	
		Kinect Fusion	COLMAP (default)	COLMAP (fast)	DROID-SLAM† [15]	ACE0 (ours)	KF+ACE0 (ours)	BARF [8]	NoPE-NeRF† [2]	ACE0 (ours)	DUSt3R [16]	ACE0 (Ours)
Chess	6k	0.37	0.17	0.17	0.33	0.18	0.18	0.80	0.76	**0.20**	0.42	**0.36**
Fire	4k	0.44	0.27	0.27	0.70	0.29	0.29	0.79	0.80	**0.27**	0.40	**0.37**
Heads	2k	0.42	0.26	0.26	0.32	0.26	0.27	0.75	0.70	**0.28**	0.40	**0.28**
Office	10k	0.42	0.25	0.24	failed	0.27	0.26	0.77	0.76	**0.31**	0.64	**0.54**
Pumpkin	6k	0.40	0.23	0.24	0.50	0.24	0.25	0.45	0.74	**0.22**	0.30	**0.27**
RedKitchen	12k	0.51	0.25	0.25	0.82	0.27	0.27	0.84	0.84	**0.35**	**0.62**	0.70
Stairs	3k	0.39	0.50	0.30	0.83	0.51	0.34	0.68	0.70	**0.43**	0.74	**0.61**
Average		0.42	0.28	0.25	N/A	0.29	0.26	0.73	0.76	**0.29**	0.50	**0.45**
Avg. Time		realtime	38h	13h	18min	1h	7min	8.5h	47h	**27min**	**4min***	16min

Table 2: 7-Scenes - LPIPS (↓). We show the pose accuracy via view synthesis with Nerfacto [14] as **LPIPS** (↓) [18], and the reconstruction time. Results for *All Frames* are color coded w.r.t. similarity to the COLMAP pseudo ground truth: > 0.05 better (lower), within ± 0.05, > 0.05 worse (higher), > 0.1 worse (higher). For some competitors, we had to sub-sample the images due to their computational complexity (right side). †Method needs sequential inputs. *Results on more powerful hardware.

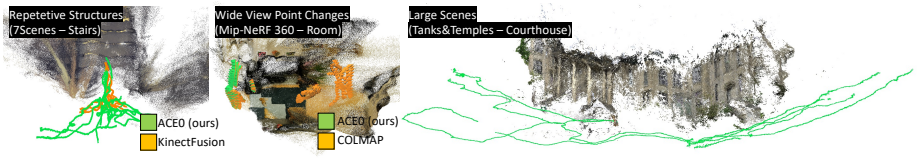

Fig. 2: Failure Cases. Stairs: Parts of the reconstruction collapse due to visual ambiguity. **Room:** ACE0 fails to register the views on the right due to low visual overlap. **Courthouse** is too large to be represented well by a single MLP.

We pair ACE0 with different depth estimators, namely ZoeDepth [1] and PlaneRCNN [9] but observe only small differences. The depth estimates are only used for the seed iteration, and their impact fades throughout the reconstruction process. Indeed, using ground truth depth, measured by a Kinect sensor, yields no noteworthy advantage either.

We run a version of ACE0 that ingests ZoeDepth estimates for all frames and uses them in all reconstruction iterations. Thus, rather than optimizing the reprojection error of Eq. 3 of the main paper, we optimize the Euclidean distance to pseudo ground truth scene coordinates computed from depth estimates. Accuracy is slightly lower, and reconstruction times longer. The depth estimates are not multi-view consistent, and the model has problems to converge.

Omitting pose refinement during neural mapping leads to a drop in PSNR of more than 2 dB. Refinement via an MLP that predicts updates has a small but noticeable advantage over direct optimization of poses via back-propagation.

Finally, PNSR numbers with early stopping are similar to using the static ACE training schedule, but reconstruction times are shorter.

More Scenes. As mentioned in the main paper, we removed two scenes from the Tanks and Temples dataset because the COLMAP baseline was not able to reconstruct them after days of processing or ran out of memory. This concerns the Courthouse and the Museum scenes in the variation with 4k frames and more. In Table 4 we show ACE0 results for these two scenes, COLMAP results when using only a few hundred images, as well as other baselines.

Registration Rates. Reconstruction algorithms do not always succeed in registering all input frames to the reconstruction. While high registration rates are in general desirable, it is also disadvantageous to register images incorrectly. Clearly, an algorithm that always returns the identity pose has a 100% registration rate but is not very useful. Therefore, in our main experimental results, we compare algorithms based on PSNR numbers rather than registration rates. The way we calculate PSNR numbers punishes an algorithm for any incorrect estimate whether its considered "registered" or not, see Sec. 2 for details. Nevertheless, we provide registration rates of ACE0 and COLMAP in Tables 5 and 6. Both algorithms achieve high registration rates for many scenes. Scenes with low registration rates also show low PSNR numbers in our main experimental results.

	PSNR (dB)	Time (min)
Stability		
Main Run	21.2	54
5 Randomized Runs	21.3±0.2	52±4
Depth Estimator		
ZoeDepth (default)	21.2	54
PlaneRCNN	21.0	**49**
Ground Truth (Kinect)	**21.3**	58
Depth Usage		
Seed only (default)	**21.2**	**54**
All Iterations	20.9	79
Pose Refinement		
MLP (default)	**21.2**	**54**
Direct Pose Updates	20.7	61
No Refinement	18.9	63
Early Stopping		
with (default)	21.2	**54**
without	**21.3**	64

Table 3: ACE0 Variations. We demonstrate the impact of various design and parameter choices as PSNR in dB and reconstruction time in minutes, on 7-Scenes.

	150-500 Images			4k-22k Frames		
	COLMAP (default)	Reality Capture	ACE0 (ours)	Reality Capture	ACE0 (ours)	Sparse COLMAP +ACE0 (ours)
Courthouse	18.2	15.5	14.4	13.2	12.5	17.2
Museum	17.0	16.2	13.0	-	14.5	17.2

Table 4: Additional ACE0 Results. We provide results for two additional scenes of Tanks and Temples where the COLMAP baseline did not finish the reconstruction (with 4k+ frames) after running for more than 5 days or ran out of memory.

7-Scenes	ACE0 (ours)	COLMAP ("fast")	Mip-NeRF 360	ACE0 (ours)	COLMAP (default)
Chess	100%	100%	Bicycle	97.9%	100%
Fire	100%	100%	Bonsai	100%	100%
Heads	100%	100%	Counter	100%	100%
Office	100%	100%	Garden	100%	100%
Pumpkin	100%	100%	Kitchen	100%	100%
R.Kitchen	98.4%	100%	Room	50.8%	100%
Stairs	100%	100%	Stump	96.0%	100%

Table 5: Registration Rates on 7-Scenes and Mip-NeRF 360. We show the percentage of registered images for ACE0 and COLMAP. Note that the way we calculate PSNR numbers already accounts for registration rates below 100%.

		150-500 Images		4k-22k Frames		
		ACE0 (ours)	COLMAP (default)	ACE0 (ours)	COLMAP ("fast")	Sparse COLMAP +ACE0 (ours)
Training	Barn	94.6%	100%	96.3%	100%	100%
	Caterpillar	100%	100%	100%	100%	100%
	Church	95.5%	100%	90.3%	99.2%	91.2%
	Ignatius	100%	100%	100%	100%	100%
	Meetingroom	100%	100%	94.8%	100%	99.2%
	Truck	100%	100%	100%	100%	100%
Intermediate	Family	100%	100%	100%	100%	100%
	Francis	99.7%	100%	100%	100%	100%
	Horse	100%	100%	100%	100%	100%
	Lighthouse	94.5%	100%	93.3%	100%	97.8%
	Playground	100%	100%	99.6%	100%	100%
	Train	96.3%	100%	96.4%	100%	97.9%
Advanced	Auditorium	98.0%	98.7%	98.7%	99%	98.3%
	Ballroom	100%	100%	100%	100%	97.8%
	Courtroom	96.3%	100%	93.1%	100%	97.8%
	Palace	25.7%	99%	56.0%	74.4%	77.2%
	Temple	7.3%	100%	74.4%	100%	92.4%

Table 6: Registration Rates on Tanks and Temples. We show the percentage of registered images for ACE0, COLMAP, and the combination of ACE0 and COLMAP. Note that the way we calculate PSNR numbers already accounts for registration rates below 100%.

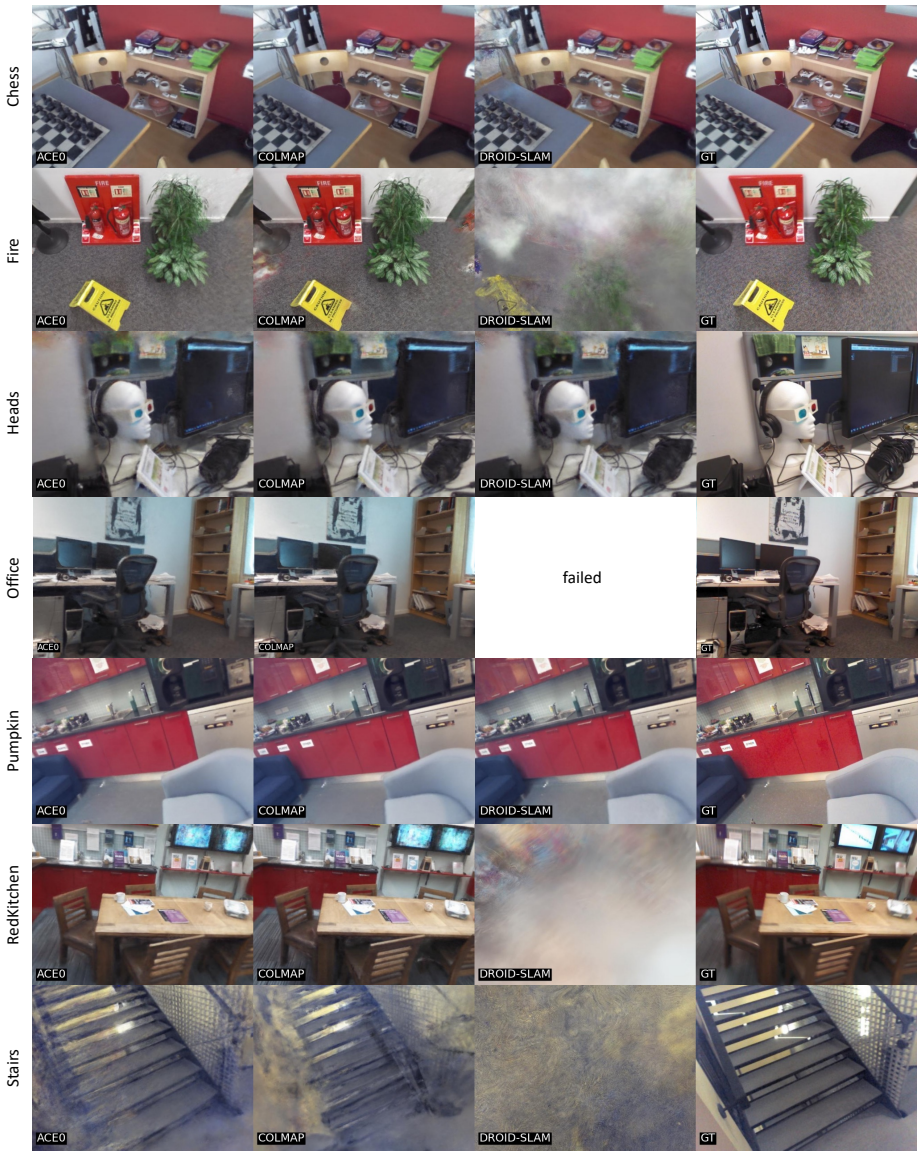

Fig. 3: View Synthesis Quality on 7-Scenes. For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). These results correspond to Table 1 (left) of the main paper. "COLMAP" refers to COLMAP with *default* parameters.

Fig. 4: View Synthesis Quality on 7-Scenes (200 Images Subset). For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). For all methods, we obtain these results on a subset of 200 images per scene to achieve a reasonable training time of NoPe-NeRF. These results correspond to Table 1 (right) of the main paper.

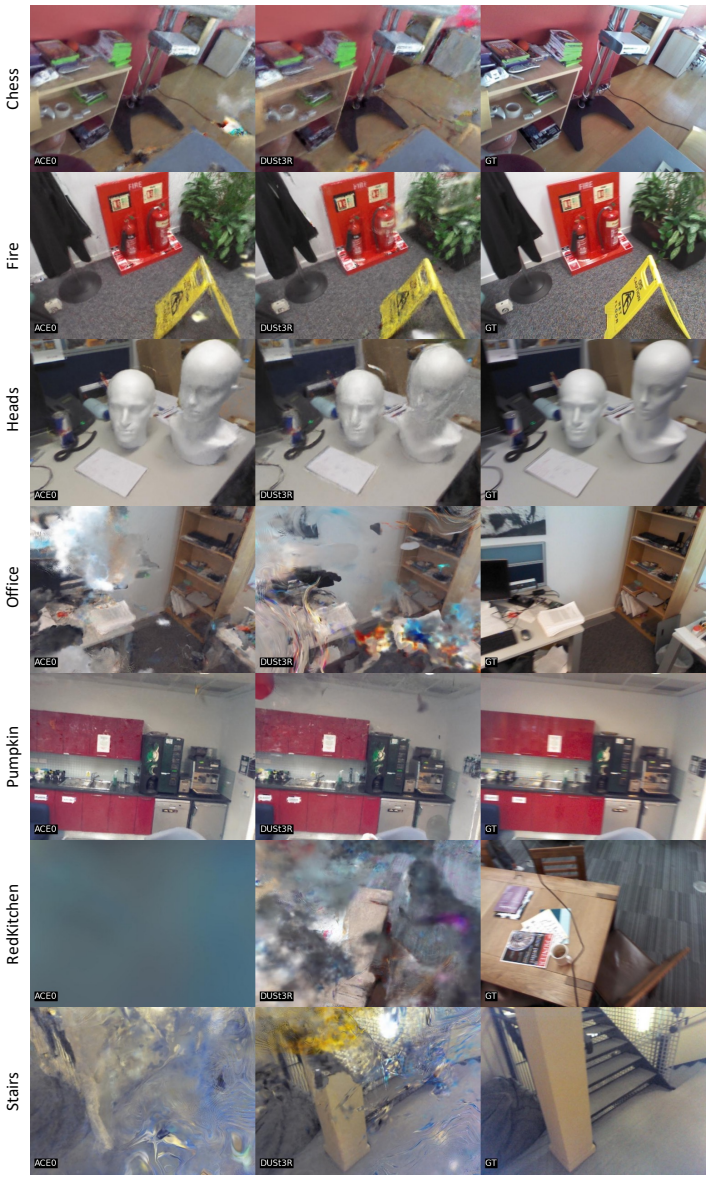

Fig. 5: View Synthesis Quality on 7-Scenes (50 Images Subset). For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). For all methods, we obtain these results on a subset of 50 images per scene to prevent DUSt3R from running out of GPU memory. These results correspond to Table 1 (right) of the main paper.

Fig. 6: View Synthesis Quality on the MIP-NeRF 360 dataset. For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). These results correspond to Table 2 (b) of the main paper.

4 Baselines and Competitors

COLMAP. We run COLMAP in three variations: *Default*, *fast*, and "Sparse COLMAP + Reloc + BA". *Fast* uses parameters recommended for large image collections beyond 1000 images [13]. "Sparse COLMAP + Reloc + BA" uses *default* parameters to obtain the initial reconstruction, and *fast* parameters to register the remaining frames. We provide the parameters of the `mapper` of the *default* variation in Table 13. For the *fast* version, we provide the parameters that have been changed compared to *default* in Table 14. We also include the parameters of the feature extractor and the matcher shared by both variations in Tables 15 and 16.

As additional comparison to further optimize for efficiency, for the Tanks and Temples dataset we also computed in Table 3 in the main paper and the second to last column of Tables 9 and 10 a COLMAP variant "Sparse COLMAP + Reloc + BA". As presented at the end of Section 4.3 in the main paper, this variant builds on the sparse reconstruction results, and performs the following steps:

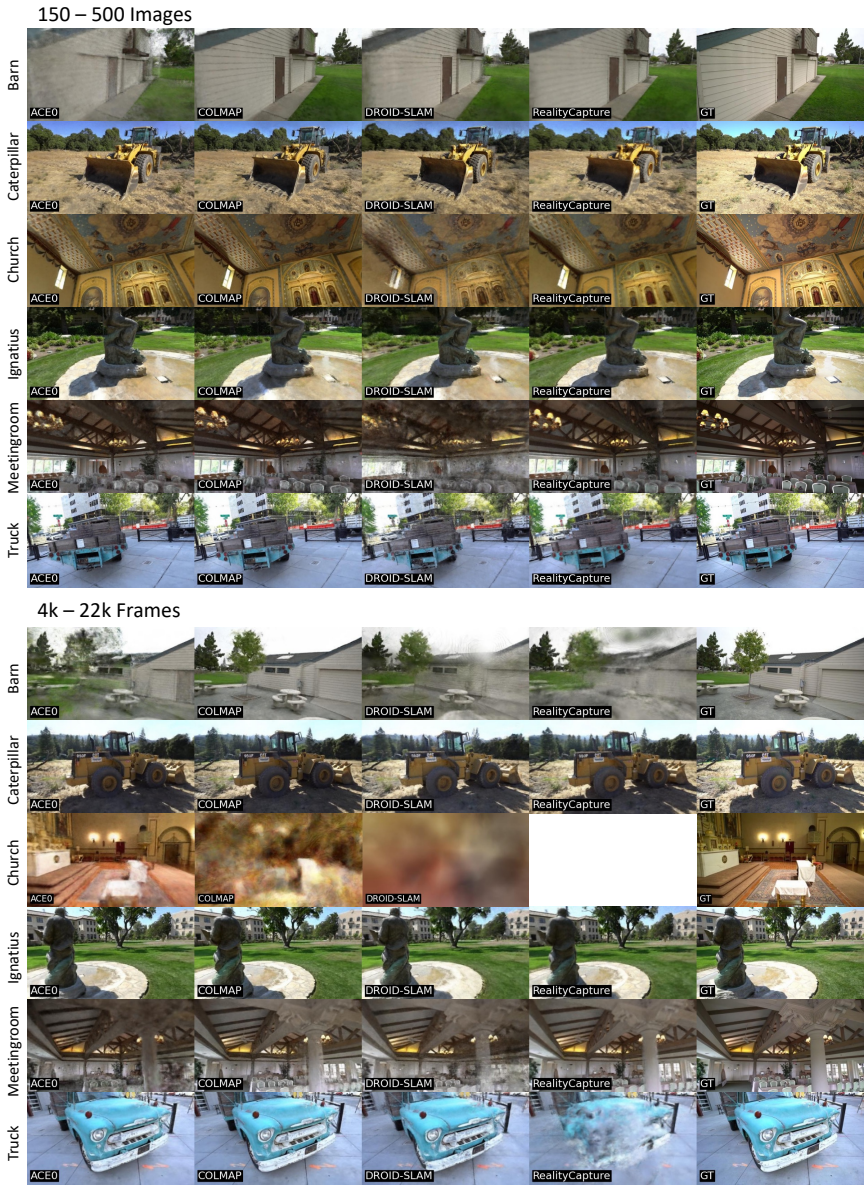

Fig. 7: View Synthesis Quality on Tanks and Temples (Training Scenes). For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). These results correspond to Table 3 of the main paper. **Top:** Results based on 150-500 images per scenes. **Bottom:** Results based on 4k-22k frames sampled from the full video.

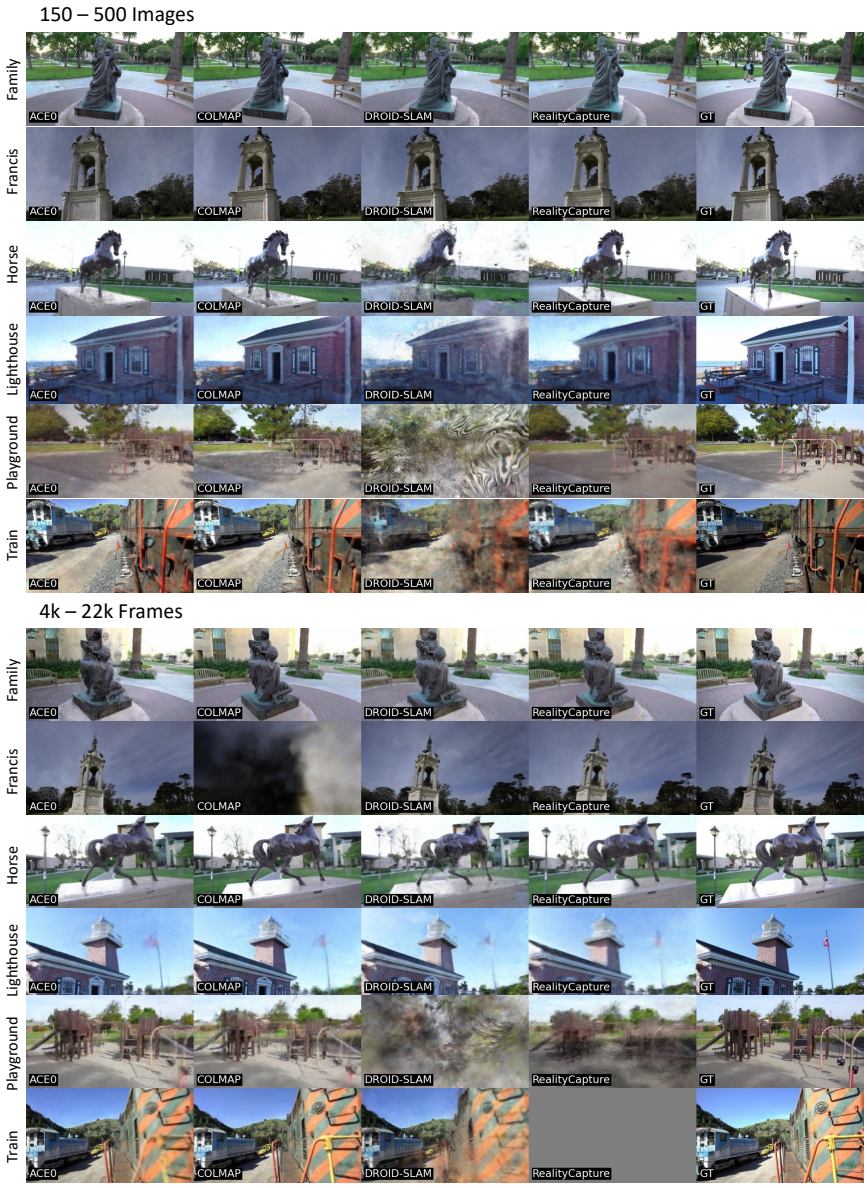

Fig. 8: View Synthesis Quality on Tanks and Temples (Intermediate Scenes). For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). These results correspond to Table 3 of the main paper. **Top:** Results based on 150-500 images per scenes. **Bottom:** Results based on 4k-22k frames sampled from the full video.

Fig. 9: View Synthesis Quality on Tanks and Temples (Advanced Scenes). For each scene, we show the test image where ACE0 achieves its median PSNR value. For comparison, we show the corresponding results of our baselines as well as the true test image (GT). These results correspond to Table 3 of the main paper. **Top:** Results based on 150-500 images per scenes. **Bottom:** Results based on 4k-22k frames sampled from the full video.

1. (Custom database surgery to prefix the old image names from the "Sparse COLMAP" model to avoid name clashes.)
2. Feature extraction (`colmap feature_extractor ...`) of the new images using the same parameters as COLMAP *default* and *fast*, *cf*., Table 15.
3. Vocabulary tree matching (`colmap vocab_tree_matcher`) of the new images to the old images and each other, using the same parameters as COLMAP *default* and *fast*, *cf*., Table 16.
4. Image registration (`colmap image_registrator ...`) of the new images to the old images and each other, given the matches computed in the previous step. The parameters for this step (*cf*., Table 17) were set to match the `mapper` parameters used in COLMAP *fast*, (*cf*., Table 14).
5. A final BA step (`colmap bundle_adjuster ...`) to refine the poses of the whole scene. Parameters were set similarly to COLMAP *fast*, *cf*., Table 18.

Note, such an algorithm heavily relies on the fact, that the sparse set of images has been selected so that it represents the entire scene, and would need further iterative optimization for datasets where the spatial distribution of images is not known beforehand (*e.g.*, not sequential data or data without priors from other sensors, such as GPS or WiFi).

NoPe-NeRF. We estimate poses with NoPe-NeRF by using the official code at https://github.com/ActiveVisionLab/nope-nerf. We modify the code to ingest both training and test views, since we aim at estimating the poses of all images jointly. For Mip-NeRF 360 scenes, we use all images per each scene reconstruction. As NoPe-NeRF assumes ordered image sequence, we sorted the images by name for training. For 7-Scenes, using all frames is prohibitively expensive. For example, the Chess scene has 6000 frames and the implementation just freezes when using them all. The majority of experiments in the NoPe-NeRF paper were done with a few hundred frames, so we subsampled the 7-Scenes scans to extract a total of 200 frames per each scene.

DUSt3R. We follow the official code at https://github.com/naver/dust3r to reconstruct scenes within the 7-Scenes dataset from uncalibrated and unposed cameras. When comparing to DUSt3R, we run ACE0 *without* the initialisation from a standard Kinect v1 focal length. That is, like DUSt3R, ACE0 estimates the focal length from scratch in the last column of Table 1 of the main paper. The DUSt3R framework cannot process all available images of a scene due to GPU memory constraints. By following the guidelines from the official repository, we configured the system to utilize a sliding window approach (w=3), enabling the accommodation of 50 frames within the 16GB memory capacity of a NVIDIA® V100 [11] GPU. As a second version, we attempted to reconstruct 50 frames using a NVIDIA® A100-40GB GPU in conjunction with the leaner 224x224 pre-trained model and an exhaustive matching configuration. We were not able to pair the full 512x512 model with exhaustive matching, even on a A100-40GB. Of the two configurations that fit into memory, the second one yields the most accurate poses under our experimental settings. Thus, we report the superior

results in Table 1 of our main paper. For further insight, we have compiled the experimental statistics in Table 7.

7-Scenes	DUSt3R$_{512}$	DUSt3R$_{224}$
Chess	16.2	**18.9**
Fire	13.4	**18.8**
Heads	13.4	**18.4**
Office	10.2	**12.5**
Pumpkin	21.0	**21.7**
R.Kitchen	11.5	**13.8**
Stairs	**16.3**	15.3
Config.	S.W.	E.M.
Hardware	V100	A100
Memory	16GB	40GB
Recon. Time	3min	4min

Table 7: DUSt3R on 7-Scenes S.W. denotes the sliding window configuration (window=3). E.M. denotes the exhaustive matching configuration. The performance is evaluated in PSNR (dB) using the same experiments described in Section 4.1 of the main paper. In the bottom part of the table, we show the configuration, GPU hardware, GPU memory, and reconstruction time for 50 frames on each scene.

DROID-SLAM. We estimated poses for the 7-Scenes (using only the RGB images), Tanks and Temples, and Mip-NeRF 360 datasets using the official code from: https://github.com/princeton-vl/DROID-SLAM. In our experiments we used the same parameters that the DROID-SLAM authors chose to evaluate the method on the ETH-3D dataset and resized the input images to a resolution having an area of 384×512 pixels, while preserving the aspect ratio. As the SLAM method requires sequential input, we sorted the images by name, although that does not remove all jumps from the datasets. For example, each scene in the 7-Scenes dataset is composed of a sequence of disjoint scans observing the same area, and the algorithm might have difficulties in tracking accurately around the discontinuities.

Reality Capture®. We also reconstructed scenes from the Tanks and Temples dataset using Reality Capture®, as discussed in Section 4.3 in the main paper. For completeness, extending the PSNR results in Table 3 in the main paper, we also computed SSIM [17] and LPIPS [18] scores in Tables 9 and 10 respectively. On the sparse images (150-500), Reality Capture performs one order of magnitude faster than COLMAP but with slightly worse pose quality. When we ran it on Tanks and Temples with thousands of frames, Reality Capture produced a high number of disconnected components. We use the largest component as basis for our evaluation and regard all other frames as missing, as described in Section 2.

Name	value
Max features per mpx	10 000
Max features per image	40 000
Image overlap	Medium
Image downscale factor	1
Max feature reprojection error	2.0
Force component rematch	**Yes**
Background feature detection	No
Background thread priority	Low
Preselector features	10 000
Detector sensitivity	Medium
Distortion model	Brown3

Table 8: Reality Capture alignment settings. We used the default settings for our experiments, except for settings in **bold**.

We used Reality Capture version `1.3.2.117357` with the default alignment parameters[3], *cf*., Table 8.

The runtimes in Tables 9 and 10 were measured for the iv) *alignment* step only, **excluding** the steps i) *starting* the software, ii) *checking license*, iii) *importing images* from local SSD before, and v) *exporting poses* as XMP, vi) *saving* the project to local SSD and vii) *closing* the software after the iv) *alignment* step. In order to find a good balance between CPU and GPU compute power, all our Reality Capture experiments were run on cloud machine with 48 logical CPU cores (Intel® Skylake, Xeon@2.00 Ghz), a single NVIDIA® T4 [12] GPU and 312GB RAM.

References

1. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: ZoeDepth: Zero-shot transfer by combining relative and metric depth. arXiv (2023)
2. Bian, W., Wang, Z., Li, K., Bian, J.W., Prisacariu, V.A.: NoPe-NeRF: Optimising neural radiance field with no pose prior. In: CVPR (2023)
3. Brachmann, E., Cavallari, T., Prisacariu, V.A.: Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In: CVPR (2023)
4. Brachmann, E., Humenberger, M., Rother, C., Sattler, T.: On the limits of pseudo ground truth in visual camera re-localisation. In: ICCV (2021)
5. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE TPAMI (2021)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM (1981)

[3] https://rchelp.capturingreality.com/en-US/tutorials/setkeyvaluetable.htm

		Frames	COLMAP (default)	Reality Capture	DROID-SLAM† [15]	ACE0 (ours)	Frames	COLMAP (fast)	Reality Capture	DROID-SLAM† [15]	ACE0 (ours)	Sparse COLMAP + Reloc + BA	Sparse COLMAP + ACE0 (ours)
Training	Barn	410	0.77	0.61	0.61	0.54	12.2k	0.77	0.56	0.49	0.58	0.84	0.79
	Caterpillar	383	0.63	0.49	0.51	0.57	11.4k	0.72	0.60	0.62	0.67	0.72	0.70
	Church	507	0.64	0.56	0.41	0.57	19.3k	0.29	-	0.34	0.54	0.66	0.59
	Ignatius	264	0.69	0.46	0.50	0.65	7.8k	0.77	0.58	0.61	0.75	0.78	0.75
	Meetingroom	371	0.67	0.62	0.48	0.63	11.1k	0.67	0.63	0.55	0.54	0.76	0.72
	Truck	251	0.76	0.60	0.59	0.72	7.5k	0.83	0.59	0.69	0.80	0.83	0.81
	Average	364	0.69	0.56	0.52	0.61	14.6k	0.68	0.59	0.55	0.65	0.76	0.73
	Avg Time		1h	3min	5min	1.1h		74h	14h	18min	2.2h	8h	1.8h
Intermediate	Family	152	0.77	0.62	0.58	0.68	4.4k	0.83	0.66	0.67	0.63	0.84	0.82
	Francis	302	0.84	0.74	0.77	0.74	7.8k	0.77	0.71	0.80	0.79	0.86	0.83
	Horse	151	0.77	0.70	0.58	0.74	6.0k	0.79	0.72	0.69	0.82	0.85	0.83
	Lighthouse	309	0.65	0.56	0.52	0.62	8.3k	0.70	0.59	0.68	0.68	0.72	0.73
	Playground	307	0.54	0.51	0.23	0.50	7.7k	0.53	0.45	0.32	0.62	0.65	0.63
	Train	301	0.62	0.45	0.39	0.54	12.6k	0.73	0.41	0.51	0.63	0.74	0.63
	Average	254	0.70	0.60	0.51	0.64	7.8k	0.73	0.59	0.61	0.70	0.78	0.75
	Avg Time		32min	2min	3min	1.3h		48h	11h	14min	2.2h	5h	1h
Advanced	Auditorium	302	0.74	0.39	0.58	0.66	13.6k	0.50	-	0.58	0.71	0.80	0.70
	Ballroom	324	0.46	0.47	0.30	0.51	10.8k	0.48	-	0.25	0.57	0.54	0.41
	Courtroom	301	0.61	0.50	0.32	0.52	12.6k	0.43	-	0.32	0.52	0.67	0.59
	Palace	509	0.54	0.43	0.41	0.34	21.9k	0.50	-	0.38	0.37	0.57	0.46
	Temple	302	0.68	0.52	0.43	0.22	17.5k	0.44	-	0.45	0.50	0.76	0.58
	Average	348	0.61	0.46	0.41	0.45	15.6k	0.47	-	0.39	0.53	0.67	0.55
	Avg Time		1h	2min	4min	1h		71h		27min	2.8h	10h	2.1h

Table 9: Tanks and Temples - SSIM (↑). We show the pose accuracy via view synthesis with Nerfacto [14] as **SSIM** (↑) [17], and the reconstruction time. We color code results compared to COLMAP, *default* and *fast*, respectively: > 0.05 better, within ± 0.05, > 0.05 worse, > 0.1 worse. †Method needs sequential inputs.

7. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography **32**(5), 922–923 (1976)
8. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: BARF: Bundle-adjusting neural radiance fields. In: ICCV (2021)
9. Liu, C., Kim, K., Gu, J., Furukawa, Y., Kautz, J.: PlaneRCNN: 3D plane detection and reconstruction from a single image. In: CVPR (2019)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
11. NVIDIA®: NVIDIA V100 TENSOR CORE GPU. https://www.nvidia.com/en-us/data-center/tesla-v100/ (2017), [acessed March/14/2024]
12. NVIDIA®: NVIDIA T4. https://www.nvidia.com/en-us/data-center/tesla-t4/ (2018), [acessed March/14/2024]
13. Schönberger, J.L.: Colmap Github Issues. https://github.com/colmap/colmap/issues/116#issuecomment-298926277 (2017), [accessed Nov/15/2023]
14. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM TOG (2023)
15. Teed, Z., Deng, J.: DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In: NeurIPS (2021)
16. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3D vision made easy. In: CVPR (2024)
17. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP (2004)

		Frames	COLMAP (default)	Reality Capture	DROID-SLAM† [15]	ACE0 (ours)	Frames	COLMAP (fast)	Reality Capture	DROID-SLAM† [15]	ACE0 (ours)	Sparse COLMAP + Reloc + BA	Sparse COLMAP + ACE0 (ours)
Training	Barn	410	0.21	0.36	0.42	0.51	12.2k	0.19	0.53	0.63	0.42	0.13	0.18
	Caterpillar	383	0.29	0.39	0.38	0.32	11.4k	0.18	0.29	0.26	0.23	0.18	0.19
	Church	507	0.31	0.38	0.64	0.38	19.3k	0.79	-	0.94	0.39	0.26	0.36
	Ignatius	264	0.21	0.34	0.35	0.25	7.8k	0.15	0.27	0.25	0.16	0.14	0.16
	Meetingroom	371	0.33	0.36	0.60	0.38	11.1k	0.31	0.37	0.50	0.45	0.21	0.27
	Truck	251	0.18	0.31	0.34	0.22	7.5k	0.11	0.37	0.24	0.14	0.11	0.13
	Average	364	0.26	0.36	0.46	0.34	14.6k	0.29	0.37	0.47	0.30	0.17	0.22
	Avg Time		1h	3min	5min	1.1h		74h	14h	18min	2.2h	8h	1.8h
Intermediate	Family	152	0.17	0.25	0.32	0.24	4.4k	0.12	0.21	0.23	0.27	0.11	0.13
	Francis	302	0.15	0.23	0.23	0.28	7.8k	0.24	0.23	0.17	0.18	0.11	0.13
	Horse	151	0.21	0.23	0.41	0.23	6.0k	0.17	0.21	0.27	0.14	0.11	0.13
	Lighthouse	309	0.36	0.43	0.57	0.37	8.3k	0.29	0.45	0.32	0.34	0.27	0.26
	Playground	307	0.45	0.41	0.72	0.48	7.7k	0.45	0.55	0.82	0.34	0.32	0.33
	Train	301	0.29	0.46	0.60	0.37	12.6k	0.20	0.64	0.45	0.29	0.18	0.30
	Average	254	0.27	0.34	0.47	0.33	7.8k	0.24	0.38	0.38	0.26	0.18	0.21
	Avg Time		32min	2min	3min	1.3h		48h	11h	14min	2.2h	5h	1h
Advanced	Auditorium	302	0.31	0.76	0.60	0.43	13.6k	0.75	-	0.64	0.37	0.21	0.37
	Ballroom	324	0.43	0.38	0.68	0.39	10.8k	0.44	-	0.92	0.33	0.34	0.46
	Courtroom	301	0.34	0.42	0.70	0.44	12.6k	0.61	-	0.79	0.43	0.24	0.34
	Palace	509	0.56	0.77	0.87	0.78	21.9k	0.60	-	0.84	0.72	0.51	0.67
	Temple	302	0.30	0.53	0.75	0.82	17.5k	0.65	-	0.70	0.53	0.21	0.44
	Average	348	0.39	0.57	0.72	0.57	15.6k	0.61	-	0.78	0.47	0.30	0.46
	Avg Time		1h	2min	4min	1h		71h		27min	2.8h	10h	2.1h

Table 10: Tanks and Temples - LPIPS (↓). We show the pose accuracy via view synthesis with Nerfacto [14] as **LPIPS** (↓) [18], and the reconstruction time. We color code results compared to COLMAP, *default* and *fast*, respectively: > 0.05 better (lower), within ± 0.05, > 0.05 worse (higher), > 0.1 worse (higher). †Method needs sequential inputs.

18. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)

	Pseudo GT (COLMAP)	DROID-SLAM† [15]	BARF [8]	NoPe-NeRF† [2]	ACE0 (ours)
Bicycle	0.68	0.20	0.22	0.24	**0.47**
Bonsai	0.89	0.18	0.31	0.36	**0.83**
Counter	0.82	0.25	0.26	0.27	**0.77**
Garden	0.86	0.38	0.27	0.26	**0.77**
Kitchen	0.87	0.28	0.34	0.33	**0.80**
Room	0.91	0.34	0.36	0.38	**0.61**
Stump	0.40	0.23	0.23	0.21	**0.40**
Average	0.78	0.26	0.28	0.30	**0.67**

Table 11: Mip-NeRF 360 - SSIM (↑). Pose quality in **SSIM** (↑) [17], higher is better. Best in **bold**. †Method needs sequential inputs.

	Pseudo GT (COLMAP)	DROID-SLAM† [15]	BARF [8]	NoPe-NeRF† [2]	ACE0 (ours)
Bicycle	0.20	0.80	0.82	0.85	**0.33**
Bonsai	0.06	0.74	0.86	0.66	**0.11**
Counter	0.10	0.64	0.83	0.76	**0.12**
Garden	0.08	0.38	0.84	0.84	**0.12**
Kitchen	0.06	0.71	0.84	0.74	**0.11**
Room	0.05	0.85	0.83	0.59	**0.43**
Stump	0.45	0.85	0.79	0.81	**0.36**
Average	0.14	0.71	0.83	0.75	**0.23**

Table 12: Mip-NeRF 360 - LPIPS (↓). Pose quality in **LPIPS** (↓) [18], lower is better. Best in **bold**. †Method needs sequential inputs.

random_seed	0
Mapper.min_num_matches	15
Mapper.ignore_watermarks	0
Mapper.multiple_models	1
Mapper.max_num_models	50
Mapper.max_model_overlap	20
Mapper.min_model_size	10
Mapper.init_image_id1	-1
Mapper.init_image_id2	-1
Mapper.init_num_trials	200
Mapper.extract_colors	1
Mapper.num_threads	-1
Mapper.min_focal_length_ratio	0.1
Mapper.max_focal_length_ratio	10
Mapper.max_extra_param	1
Mapper.ba_refine_focal_length	1
Mapper.ba_refine_principal_point	0
Mapper.ba_refine_extra_params	1
Mapper.ba_min_num_residuals_for_multi_threading	50000
Mapper.ba_local_num_images	6
Mapper.ba_local_function_tolerance	0
Mapper.ba_local_max_num_iterations	25
Mapper.ba_global_use_pba	0
Mapper.ba_global_pba_gpu_index	-1
Mapper.ba_global_images_ratio	1.1
Mapper.ba_global_points_ratio	1.1
Mapper.ba_global_images_freq	500
Mapper.ba_global_points_freq	250000
Mapper.ba_global_function_tolerance	0
Mapper.ba_global_max_num_iterations	50
Mapper.ba_global_max_refinements	5
Mapper.ba_global_max_refinement_change	0.0005
Mapper.ba_local_max_refinements	2
Mapper.ba_local_max_refinement_change	0.001
Mapper.snapshot_images_freq	0
Mapper.fix_existing_images	0
Mapper.init_min_num_inliers	100
Mapper.init_max_error	4
Mapper.init_max_forward_motion	0.95
Mapper.init_min_tri_angle	16
Mapper.init_max_reg_trials	2
Mapper.abs_pose_max_error	12
Mapper.abs_pose_min_num_inliers	30
Mapper.abs_pose_min_inlier_ratio	0.25
Mapper.filter_max_reproj_error	4
Mapper.filter_min_tri_angle	1.5
Mapper.max_reg_trials	3
Mapper.local_ba_min_tri_angle	6
Mapper.tri_max_transitivity	1
Mapper.tri_create_max_angle_error	2
Mapper.tri_continue_max_angle_error	2
Mapper.tri_merge_max_reproj_error	4
Mapper.tri_complete_max_reproj_error	4
Mapper.tri_complete_max_transitivity	5
Mapper.tri_re_max_angle_error	5
Mapper.tri_re_min_ratio	0.2
Mapper.tri_re_max_trials	1
Mapper.tri_min_angle	1.5
Mapper.tri_ignore_two_view_tracks	1

Table 13: COLMAP's *default* incremental `mapper` options. **Bold** denotes options later changed in the *fast* version, *cf.*, Table 14.

Mapper.ba_global_images_ratio	**1.2**
Mapper.ba_global_points_ratio	**1.2**
Mapper.ba_global_points_freq	**200000**
Mapper.ba_global_max_num_iterations	**20**
Mapper.ba_global_max_refinements	**3**

Table 14: The incremental `mapper` options we used when running COLMAP *fast*. Other parameters were not changed, and can be seen in Table 13. Parameters were recommended for large image collections of more than 1000 images by [13].

random_seed	0
descriptor_normalization	l1_root
ImageReader.mask_path	-
ImageReader.camera_model	**SIMPLE_PINHOLE**
ImageReader.single_camera	**1**
ImageReader.single_camera_per_folder	0
ImageReader.single_camera_per_image	0
ImageReader.existing_camera_id	-1
ImageReader.camera_params	**f, cx, cy** from dataset
ImageReader.default_focal_length_factor	1.2
ImageReader.camera_mask_path	-
SiftExtraction.num_threads	-1
SiftExtraction.use_gpu	1
SiftExtraction.gpu_index	-1
SiftExtraction.max_image_size	3200
SiftExtraction.max_num_features	8192
SiftExtraction.first_octave	-1
SiftExtraction.num_octaves	4
SiftExtraction.octave_resolution	3
SiftExtraction.peak_threshold	$0.006\dot{6}$
SiftExtraction.edge_threshold	10
SiftExtraction.estimate_affine_shape	0
SiftExtraction.max_num_orientations	2
SiftExtraction.upright	0
SiftExtraction.domain_size_pooling	0
SiftExtraction.dsp_min_scale	$0.16\dot{6}$
SiftExtraction.dsp_max_scale	3
SiftExtraction.dsp_num_scales	10

Table 15: COLMAP's `feature_extractor` options. **Bold** denotes options changed w.r.t. default. Ran on an NVIDIA® V100 [11] GPU.

random_seed	0
SiftMatching.num_threads	-1
SiftMatching.use_gpu	1
SiftMatching.gpu_index	-1
SiftMatching.max_ratio	0.8
SiftMatching.max_distance	0.7
SiftMatching.cross_check	1
SiftMatching.max_error	4
SiftMatching.max_num_matches	32768
SiftMatching.confidence	0.999
SiftMatching.max_num_trials	10000
SiftMatching.min_inlier_ratio	0.25
SiftMatching.min_num_inliers	15
SiftMatching.multiple_models	0
SiftMatching.guided_matching	0
SiftMatching.planar_scene	0
SiftMatching.compute_relative_pose	0
VocabTreeMatching.num_images	100
VocabTreeMatching.num_nearest_neighbors	5
VocabTreeMatching.num_checks	256
VocabTreeMatching.num_images_after_verification	0
VocabTreeMatching.max_num_features	-1
VocabTreeMatching.vocab_tree_path	https://demuc.de/colmap/vocab_tree_flickr100K_words256K.bin

Table 16: COLMAP's default `vocab_tree_matcher` options used. Ran on an NVIDIA® V100 [11] GPU.

Mapper.ba_global_images_ratio	**1.2**
Mapper.ba_global_points_ratio	**1.2**
Mapper.ba_global_max_num_iterations	**20**
Mapper.ba_global_max_refinements	**3**
Mapper.ba_global_points_freq	**200 000**

Table 17: The `image_registrator` options used for COLMAP in the "Sparse COLMAP + Reloc + BA" step. **Bold** denotes options changed w.r.t. default. Parameters set to match Table 14, as recommended for large image collections of more than 1000 images by [13].

BundleAdjustment.max_num_iterations	**20**

Table 18: The `bundle_adjuster` options used for COLMAP in the "Sparse COLMAP + Reloc + BA" step. **Bold** denotes options changed w.r.t. default. Parameters set to match Table 14, as recommended for large image collections of more than 1000 images by [13].