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

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

Abstract. We address the task of estimating camera parameters from a set of images depicting a scene. Popular feature-based structure-from-motion (SfM) tools solve this task by incremental reconstruction: they repeat triangulation of sparse 3D points and registration of more camera views to the sparse point cloud. We re-interpret incremental structure-from-motion as an iterated application and refinement of a visual relocalizer, that is, of a method that registers new views to the current state of the reconstruction. This perspective allows us to investigate alternative visual relocalizers that are not rooted in local feature matching. We show that *scene coordinate regression*, a learning-based relocalization approach, allows us to build implicit, neural scene representations from unposed images. Different from other learning-based reconstruction methods, we do not require pose priors nor sequential inputs, and we optimize efficiently over thousands of images. In many cases, our method, ACE0, estimates camera poses with an accuracy close to feature-based SfM, as demonstrated by novel view synthesis.
Project page: https://nianticlabs.github.io/acezero/

1 Introduction

In the beginning there was structure-from-motion.

The genesis of numerous computer vision tasks lies in the estimation of camera poses and scene geometry from a set of images. It is the first fundamental step that lets us leave the image plane and venture into 3D. Since structure-from-motion (SfM) is such a central capability, we have researched it for decades. By now, refined open-source tools, such as COLMAP [80], and efficient commercial packages, such as RealityCapture [68], are available to us. Feature matching-based SfM is the gold standard for estimating poses from images, with a precision that makes its estimates occasionally considered "ground truth" [3, 5, 13, 46, 69].

The success of Neural Radiance Fields (NeRF) [61] has renewed interest in the question of whether SfM can be solved differently, based on neural, implicit scene representations rather than 3D point clouds. There has been some progress in recent years but, thus far, learning-based approaches to camera pose recovery

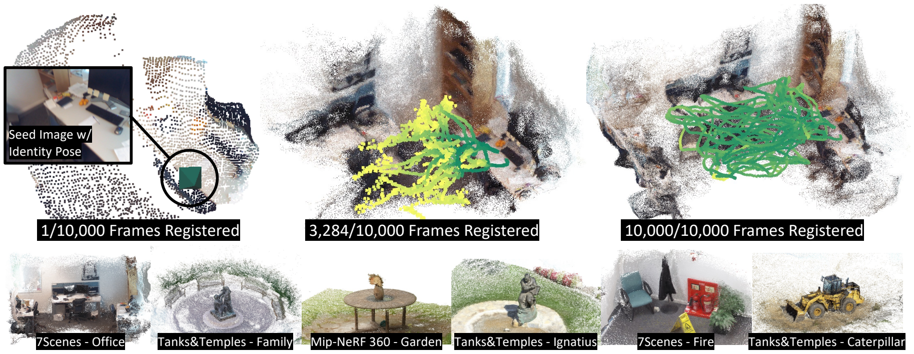

Fig. 1: Reconstructing 10,000 Images. Top: Starting from a single image and the identity pose, we train a learning-based visual relocalizer. The relocalizer allows us to estimate the poses of more views, and the additional views allow us to refine the relocalizer. We show three out of six iterations for this scene (7Scenes Office [81]). All 10k images have been posed in roughly 1 hour on a single GPU. In comparison NoPe-NeRF [10] needs two days to pose 200 images. The point cloud is a visualization of the implicit scene representation of the relocalizer. Camera positions are color coded by relocalization confidence from yellow (low) to green (high). **Bottom:** Point clouds from Nerfacto [89] trained on top of our poses for a few scenes from our experiments.

still have significant limitations. They either require coarse initial poses [45, 54, 98, 101], prior knowledge of the pose distribution [60] or sequential inputs [10, 11, 90]. In terms of the number of images that learning-based approaches can handle, they are either explicitly targeted at few-frame problems [53, 82, 95, 97, 99, 103] or they are computationally so demanding that they can realistically only be applied to a few hundred images at most [10, 54, 98]. We show that none of these limitations are an inherent consequence of using learning-based scene representations.

Our approach is inspired by incremental SfM and its relationship to another computer vision task: visual relocalization. Visual relocalization describes the problem of estimating the camera pose of a query image w.r.t. to an existing scene map. Incremental SfM can be re-interpreted as a loop of 1) do visual relocalization to register new views to the reconstruction, and 2) refine/extend the reconstruction using the newly registered views. Local feature matching is a traditional approach to visual relocalization [70, 74–76]. In recent years, multiple learning-based relocalizers have been proposed that encode the scene implicitly in the weights of a neural network [3, 12, 14, 15, 17, 22, 47]. Not all of them are suitable for building a SfM pipeline. We need a relocalizer with high accuracy and good generalization. Training of the relocalizer has to be swift. We need to be able to bootstrap relocalization without ground truth poses. And we need to be able to tell whether registration of a new image was successful.

We show that "scene coordinate regression", an approach to visual relocalization proposed a decade ago [81], has the desirable properties and can serve as

the core of a new approach to learning-based SfM: Scene coordinate *reconstruction.* Rather than optimizing over image-to-image matches, like feature-based SfM, scene coordinate reconstruction regresses image-to-*scene* correspondences directly. Rather than representing the scene as a 3D point cloud with high dimensional descriptors, we encode the scene into a lightweight neural network. Our approach works on unsorted images without pose priors and efficiently optimises over thousands of images, see Figure 1.
We summarize our **contributions**:

- *Scene Coordinate Reconstruction*, a new approach to SfM based on incremental learning of scene coordinate regression, a visual relocalization principle.
- We turn the fast-learning visual relocalizer ACE [12] into a SfM framework that is able to predict the camera poses of a set of unposed RGB images. We refer to this new SfM pipeline as *ACE0* (ACE Zero).
- Compared to ACE [12], we add the capability to train in a self-supervised fashion. We start from a single image, and iterate between learning the map and registering new views. We expedite reconstruction times by early stopping, and increase reconstruction quality by pose refinement.

2 Related Work

Reconstruction. SfM pipelines either ingest a collection of *unordered images* [20,78,84] or *an ordered image sequence* [6,30,65,66,88] from a video to recover 3D structure of a scene and camera poses ("motion") of the images.

SIFT [58] and other robust descriptors allow matching image features across wide baselines enabling systems to reconstruct large-scale scenes using Internet images [80,84,85,100]. Image-to-image matches can also be regressed directly in a detector-free setup [41,87]. Feature tracks across multiple images are built from image-to-image matches. Feature tracks and estimated relative poses are used to solve for the 3D feature coordinates, camera poses and calibrations (intrinsic matrices). This geometric optimization problem is mainly solved using bundle adjustment which was explored in photogrammetry and geodesy [19, 50] and became standard in the computer vision community [40,92]. Bundle adjustment relies on the initialization being close to the solution (*i.e.*, camera poses and 3D points are already mostly accurate).

There are two main approaches to this problem. *Incremental SfM* [6, 66] starts the reconstruction from very few images to create a high-quality seed reconstruction that progressively grows by registering more images and refining the reconstruction until convergence. *Global SfM* approaches solve for "global" poses of all images using estimates of relative poses, *i.e.*, motion averaging [37, 38], rotation averaging [39,59] and pose-graph optimization [21]. Various techniques were proposed to improve SfM runtime for very large sets of images [1,2,8,9,28, 36,42,86,91]. Our work is similar to Incremental SfM, as we also progressively register images to the reconstructed scene starting from a seed reconstruction. However, we do not explicitly compute image matches, nor feature tracks across images, which can be computationally expensive.

Visual Relocalization. A reconstructed (or mapped) scene is a database of images with known camera poses. This database can be used by a visual relocalizer to estimate poses for new query images to "relocalize" a camera in the scene. Feature-based approaches extract 2D local features [31,33,56,58,71,72,87] from a query image and match them to 3D points to solve for the query pose using perspective-n-point (PnP) solvers [35], *e.g.*, [73–75]; or match query features to 2D local features of mapped images to triangulate the query image, *e.g.*, [104, 105]. For large scenes, matching features on a subset of database images relevant to the query can improve the speed and accuracy, *e.g.*, [43,67,70,76].

Some learning-based approaches encode the map of a scene in the weights of a neural network. PoseNet [47] directly regresses the absolute camera pose given an input image using a CNN that was trained on image-pose pairs of the reconstructed scene. Sattler *et al.* show that extrapolating outside the mapped poses can be challenging for absolute pose regression approaches [77]. Relative pose regression networks [4, 32, 51, 93] estimate the relative pose for a pair of images, allowing triangulation of the query image from multiple map images, or estimate a scale-metric relative pose w.r.t. a single map image [3]. Scene coordinate regression [12,14,15,17,22,23,52,81] directly predicts 3D coordinates of a point given a patch. This approach generalizes well as the camera pose is solved using PnP [35] within a RANSAC loop [34]. ACE [12] shows that training of scene coordinate regression can be greatly accelerated. In our work we train the ACE localizer, but we start from images without poses.

Image-Based Rendering. In recent years, neural radiance fields (NeRFs) [61] have seen a lot of attention from the community. NeRFs allow photorealistic novel view synthesis when trained from image-pose pairs. Typically, estimation of the image poses is done in advance by using an SfM pipeline such as COLMAP, *e.g.*, [5]. Nonetheless, research exists that estimates camera poses with NeRFs, facilitates camera localization for novel views after NeRF training [24–26, 62, 102], or simultaneously estimates camera poses during NeRF training from images alone [10, 27, 54, 82, 98]. However, these approaches either assume that the scene is captured from the front [45, 98, 101], that coarse poses are available for initialization [54], or that images are captured sequentially [10]. Techniques that rely on a multi-layer perceptron (MLP) representation of the scene, *e.g.* [10, 27], are slow to train, taking days to converge. While radiance fields can be trained faster using multi-layer hash grids [63] or Gaussian splats [48], their efficacy in pose estimation without approximate prior pose initialization [55, 62] or sequential image inputs [48] remains unproven. Concurrently, several learning-based camera pose estimation methods have been proposed [53, 97, 103]. Due to GPU memory constraints, these methods estimate poses for sparse sets of images.

3 Method

Preliminaries. The input to our system is a set of RGB images, denoted by $\mathcal{I} = \{I_i\}$, where i refers to the image index. Our system estimates the corresponding

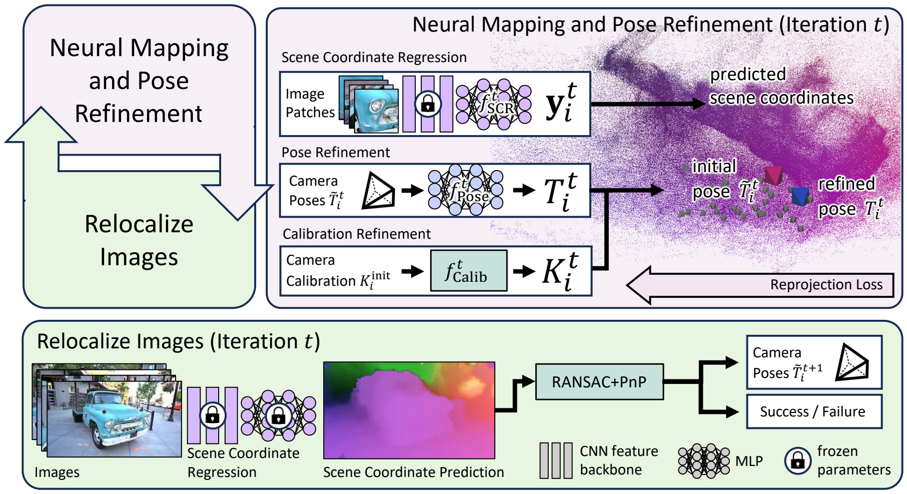
Neural Mapping and Pose Refinement
Relocalize Images
Neural Mapping and Pose Refinement (Iteration t)
Scene Coordinate Regression
Image Patches
f^t_{SCR}
$\mathbf{y}^t_i$
predicted scene coordinates
Pose Refinement
Camera Poses $\tilde{T}^t_i$
f^t_{Pose}
T^t_i
initial pose $\tilde{T}^t_i$
refined pose T^t_i
Calibration Refinement
Camera Calibration K^{init}_i
f^t_{Calib}
K^t_i
Reprojection Loss
Relocalize Images (Iteration t)
Images
Scene Coordinate Regression
Scene Coordinate Prediction
RANSAC+PnP
Camera Poses $\tilde{T}^{t+1}_i$
Success / Failure
CNN feature backbone
MLP
frozen parameters

Fig. 2: ACE0 Framework. Top left: We loop between learning a reconstruction from the current set of images and poses ("neural mapping"), and estimating poses of more images ("relocalization"). **Top right:** During the mapping stage, we train a scene coordinate regression network as our scene representation. Camera poses of the last relocalization round and camera calibration parameters are refined during this process. We visualize scene coordinates by mapping XYZ to the RGB cube. **Bottom:** In the relocalization stage, we re-estimate poses of images using the scene coordinate regression network, including images that were previously not registered to the reconstruction. If the registration of an image succeeds, it will be used in the next iteration of the mapping stage; otherwise it will not.

set of camera parameters, both intrinsics and extrinsics: $\mathcal{H} = \{(K_i, T_i)\}$. Each T_i refers to a 3×4 matrix containing a rotation and translation, while K_i refers to a 3×3 matrix with the calibration parameters. We assume no particular image order or any prior knowledge about the pose distribution.

We also want to recover the 3D structure of the scene: Each pixel j in image i with 2D pixel position $\mathbf{p}_{ij}$ has a corresponding coordinate in 3D, denoted as $\mathbf{y}_{ij}$. The 2D pixel positions and 3D scene coordinates are related by the camera pose and the projection function $\boldsymbol{\pi}$:

$$\mathbf{p}_{ij} = \boldsymbol{\pi}(K_i, T_i, \mathbf{y}_{ij}), \tag{1}$$

where T_i maps camera coordinates to scene coordinates, and K_i projects camera coordinates to the image plane.

As our scene representation, we utilize a scene coordinate regression model [81], *i.e.*, a learnable function f_{SCR} that maps an image patch of image I_i, centered around pixel position $\mathbf{p}_{ij}$ to a scene coordinate: $\mathbf{y}_{ij} = f_{\mathrm{SCR}}(\mathbf{p}_{ij}, I_i)$.

Given a set of 2D-3D correspondences predicted by f_{SCR} for any image I_i, we can recover this image's camera pose T_i using a pose solver g:

$$T_i = g\left(K_i, \{(\mathbf{p}_{ij}, \mathbf{y}_{ij})\}\right). \tag{2}$$

Since 2D-3D correspondences can be inaccurate, and contain incorrect predictions, g combines a PnP solver [35] with a RANSAC loop [34].

Normally, scene coordinate regression models are trained in a supervised fashion for the task of visual relocalization [12, 14, 15, 17, 22, 23, 52, 81]. That is, f_{SCR} is trained using images with known ground truth camera parameters $\{(I_i, T_i^{\text{GT}}, K_i^{\text{GT}})\}$, and used for estimating the poses of unseen query images. Instead, we show how these models can be trained self-supervised, without ground truth poses, to estimate the poses of the mapping images themselves. Thus, we turn scene coordinate regression into scene coordinate reconstruction, a learning-based SfM tool.

3.1 Neural Mapping

We train the scene coordinate regression model iteratively where we denote the current time step as t and the corresponding scene model as f^t_{SCR}. We iterate between training the scene model, and registering new views, see Figure 2.

At iteration t we assume that a subset of images has already been registered to the scene, $\mathcal{I}^t_{\text{Reg}} \subset \mathcal{I}$, and where corresponding camera parameters, T^t_i and K^t_i, have already been estimated. Using these as pseudo ground truth, we train the scene model by minimizing the pixel-wise reprojection error:

$$\sum_{I_i \in \mathcal{I}^t_{\text{Reg}}} \sum_{j \in I_i} \left\| \mathbf{p}_{ij} - \boldsymbol{\pi}(K^t_i, T^t_i, \mathbf{y}^t_{ij}) \right\|, \tag{3}$$

where the scene model f^t_{SCR} predicts coordinates $\mathbf{y}^t_{ij}$.

Mapping Framework. We optimize Eq. 3 using stochastic gradient descent, using the fast-learning scene coordinate regressor ACE [12] (Accelerated Coordinate Encoding). ACE is trained in minutes, even for thousands of views. The training speed is important since we have to train the model in multiple iterations. ACE approximates Eq. 3 by sampling random patches from the training set, $\mathcal{I}^t_{\text{Reg}}$. To do that efficiently, ACE employs a pre-trained encoder to precompute high dimensional features for a large number of training patches. The encoder stays frozen during mapping. The actual scene model, which is trained, is a multi-layer perceptron (MLP) that maps encoder features to scene coordinates; see Fig. 2 (top right) for a visual representation.

Pose Refinement. Differently from the ACE [12] protocol, the ground truth poses T_i^{GT} are unknown during training. Instead, we have T^t_i, estimates based on earlier iterations of the reconstruction. Since these estimates can be inaccurate, we add the ability to refine poses during mapping. We implement refinement using an MLP:

$$T^t_i = f^t_{\text{Pose}}(\tilde{T}^t_i), \tag{4}$$

where $\tilde{T}^t_i$ denotes the initial pose estimate at the start of a mapping iteration. Inspired by [106], the refinement MLP ingests $\tilde{T}^t_i$ as $3 \times 4 = 12$ values and predicts 12 additive offsets. We orthonormalize rotations using Gram-Schmidt [18]. We jointly optimize f^t_{Pose} and f^t_{SCR} to minimize the reprojection error of Eq. 3. We

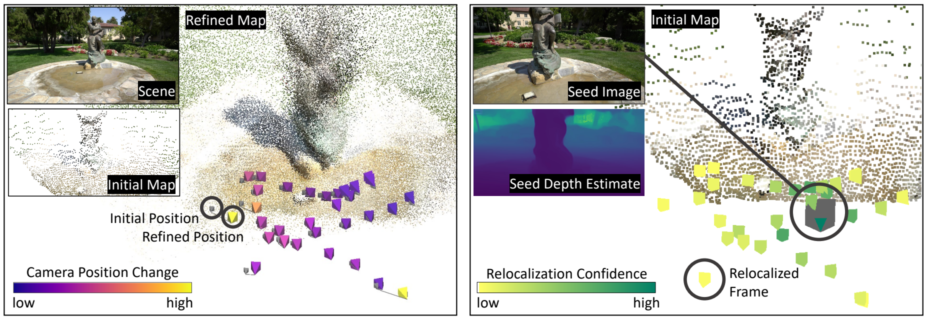

Fig. 3: Left: Pose Refinement. Since we register images based on a coarse and incomplete state of the reconstruction, we add the ability to refine poses during neural mapping. An MLP predicts pose updates relative to the initial poses, supervised by the reprojection error of scene coordinates. **Right: Initialization.** To start the reconstruction, we train the network using one image, the identity pose and a depth estimate, here ZoeDepth [7]. In this example, we register 33 views to the initial reconstruction. Depth estimates are only used for this step.

show the impact of pose refinement for one iteration in Fig. 3 (left). We discard the refinement MLP after each mapping iteration. Its purpose is to enable the scene model, f^t_{SCR}, to converge to a consistent scene representation.

With neither poses nor 2D-3D correspondences fixed in Eq. 3, the scene coordinate regressor could drift or degenerate. As regularization, we optimize the pose refiner f^t_{Pose} using AdamW [57] with weight decay, biasing the MLP to predict small updates relative to the initial estimates $\tilde{T}^t_i$. This relies on the assumption that mapping images in $\mathcal{I}^t_{\text{Reg}}$ have been registered close to their true position. If that assumption holds, the smoothness prior of the networks [94] encourages a multi-view consistent solution as shown in previous work [12,15,17].

As an alternative to an MLP refiner, we could back-propagate directly to the input poses [98, 102]. However, when optimizing over thousands of views, the signal for each single pose becomes sparse. Cameras are correlated via the scene representation. If the optimization removes drift in the scene, multiple cameras need to move. The MLP refiner models the correlation of cameras.

Calibration Refinement. We do not assume information about precise calibration parameters, although often reported by devices. We do assume that the principal point is in the center, that pixels are unskewed and square. We do not model image distortion. While these are reasonable assumptions for many data regimes, we cannot rely on the focal length to be given. Thus, we refine the focal length starting from a heuristic: $K^t_i = f^t_{\text{Calib}}(K^{\text{init}}_i)$. The refinement function f^t_{Calib} entails a single learnable parameter α^t such that the focal length $f^t_i = f^{\text{init}} \cdot (1 + \alpha^t)$. As before, superscript t denotes the time step. We optimize α^t using AdamW [57] with weight decay, biasing it towards a small relative scale factor. Estimates of α are carried over across iterations. We set f^{init} to 70% of the image diagonal and it is shared by all cameras in our experiments.

3.2 Relocalization

Given the scene model of iteration t, we attempt to register more images to determine the training set for the next mapping iteration, $\mathcal{I}^{t+1}_{\text{Reg}}$. We pass all images in $\mathcal{I}$ to the scene coordinate regressor f^t_{SCR} to gather 2D-3D correspondences, and solve for their poses using RANSAC and PnP:

$$\tilde{T}^{t+1}_i, s^{t+1}_i = g\left(K^t_i, \left\{\left(\mathbf{p}_{ij}, \mathbf{y}^t_{ij}\right)\right\}\right). \tag{5}$$

Here, we assume that the pose solver returns a confidence score s^{t+1}_i alongside the pose itself that lets us decide whether the pose of the image has been estimated successfully. We simply utilize the inlier count as score s^{t+1}_i and apply a threshold to form the training set of the next mapping iteration: $\mathcal{I}^{t+1}_{\text{Reg}} = \left\{I_i | s^{t+1}_i > \tau_s\right\}$. The relocalization process is depicted in Figure 2, bottom.

3.3 Initialization

We start the reconstruction with one image: $\mathcal{I}^0_{\text{Reg}} = \{I_{\text{seed}}\}$. We set the seed pose T^0_{seed} to identity, and we initialize the calibration $K^{\text{init}}_{\text{seed}}$ as explained above.

We cannot train a scene coordinate regression network using Eq. 3 with a single image. The reprojection error is ill-conditioned without multiple views constraining the depth. Therefore, we optimize a different objective in the seed iteration, inspired by Map-free Relocalization [3]. Arnold *et al.* argue that a single image and a depth estimate allow to relocalize query images, albeit with limited accuracy. Our experiments show that such coarse relocalizations of a few images suffice as initialization for optimizing the reprojection error of Eq. 3.

Let d_{ij} be a depth value predicted for pixel j of image i. We derive a target scene coordinate by back-projection as $\hat{\mathbf{y}}_{ij} = d_{ij}(K^{\text{init}}_{\text{seed}})^{-1}\mathbf{p}_{ij}$. We train an initial scene coordinate regression network f^0_{SCR} by optimizing $\sum_{j \in I^0} \|\hat{\mathbf{y}}_{ij} - \mathbf{y}_{ij}\|$. Fig. 3 (right) shows a scene coordinate point cloud learned from a depth estimate, and successful relocalization against it.

We found our pipeline to be robust w.r.t. selecting the seed image. However, when a randomly selected image has little to no visual overlap with the remaining images, the whole reconstruction would fail. Selecting an unfortunate seed image, *e.g.*, at the very end of a long camera trajectory, can increase reconstruction times. To decrease the probability of such incidents, we try 5 random seed images and choose the one with the highest relocalization rate across 1000 other mapping images. Since mapping seed images is fast, *ca.* 1 min on average, this poses no significant computational burden.

3.4 Implementation

We base our pipeline on the public code of the ACE relocalizer [12]. ACE uses a convolutional feature backbone, pre-trained on ScanNet [29], that ingests images scaled to 480px height. On top of the backbone is the mapping network, a 9-layer MLP with 512 channels, consuming 4MB of weights in 16-bit floating point precision. This is our scene representation.

The ACE training process is reasonably fast, taking 5 minutes to train the scene network. Since we repeat training the scene network in multiple iterations, we expedite the process further to decrease our total reconstruction time.

Adaptive Sampling of Training Patches. ACE trains the scene representation based on 8M patches sampled from the mapping images, a process that takes 1 minute. This is excessive when having very few mapping images in the beginning of the reconstruction, thus we loop over the mapping images at most 10 times when sampling patches, or until 8M patches have been sampled. Using this adaptive strategy, sampling patches for the seed reconstruction where we have 1 image only takes 2 seconds instead of 1 minute.

Adaptive Stopping. ACE trains the scene model using a fixed one-cycle learning rate schedule [83] to a total of 25k parameter updates. We make the training schedule adaptive since the network is likely to converge fast when trained on few images only. We monitor the reprojection error of scene coordinates within a mini-batch. If for 100 consecutive batches 70% of reprojection errors are below an inlier threshold of 10px, we stop training early. We approximate the one-cycle learning rate schedule in a linear fashion: we increase the learning rate in the first 1k iteration from 5×10^{-4} to 3×10^{-3} and when the early stopping criterion has been met, we decrease the learning rate to 5×10^{-4} within 5k iterations.

We report more implementation details and hyper-parameters in the supplement. Our code is also publicly available to ensure reproducibility.

4 Experiments

We refer to our SfM pipeline as ACE0 (ACE Zero) since it builds on top of the ACE relocalizer [12] but adds the ability to train from scratch, without poses. We demonstrate the effectiveness of our approach on three datasets and 31 scenes in total. For all experiments, we rely on ZoeDepth [7] to initialize our reconstructions. All timings reported for ACE0 are based on a single V100 GPU.

Baselines. We consider the pose estimates of COLMAP with default parameters as our pseudo ground truth, obtained by extensive processing. We also run COLMAP with parameter settings recommended for large image collections of 1k images and more [79]. This variation achieves much faster processing (denoted *COLMAP fast*). We use a V100 GPU for COLMAP feature extraction and matching, and we specify the exact parameters of COLMAP in the supplement. Furthermore, we show some results of RealityCapture [68], an efficient commercial feature-based SfM pipeline.

We compare to learning-based SfM approaches that are not restricted to few-frame scenarios, namely to NeRF-based BARF [54] and NoPe-NeRF [10]. We also compare to DUSt3R [97], a non-NeRF learning-based SfM method. While we focus on reconstructing unsorted image collections, the datasets we consider allow for sequential processing. Hence, for context, we also show results of DROID-SLAM [90], a neural SLAM approach. Unless specified otherwise, we report timings based on a single V100 GPU.

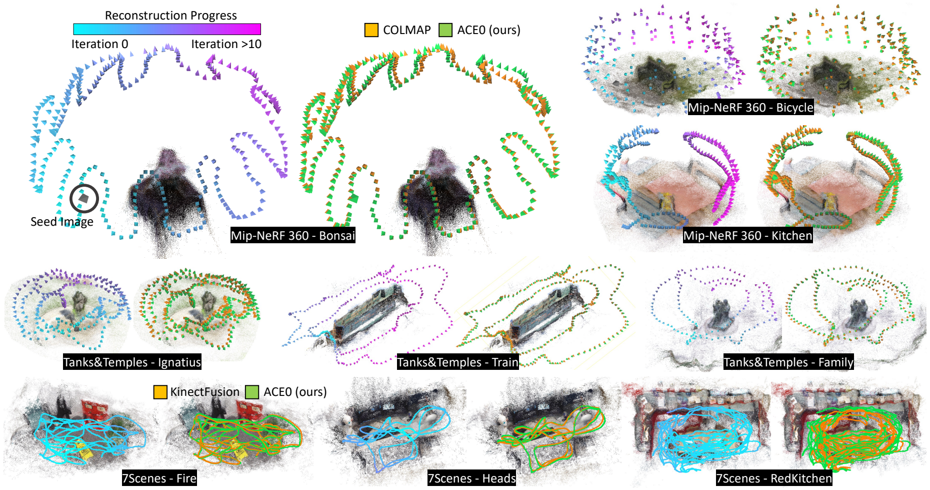

Fig. 4: Reconstructed Poses. We show poses estimated by ACE0 for a selection of scenes. We color code the reconstruction iteration in which a particular view has been registered. We show the ACE0 point cloud as a representation of the scene. The seed image is shown as a gray frustum. We also compare our poses to poses estimated by COLMAP (Mip-NeRF 360, Tanks and Temples) and KinectFusion (7-Scenes).

Benchmark. We show results on 7-Scenes [81], a relocalization dataset, on Mip-NeRF 360 [5], a view synthesis dataset and on Tanks and Temples [49], a reconstruction dataset. Comparing poses on these datasets is problematic, as our pseudo ground truth is estimated rather than measured. For example, an approach might be more accurate than COLMAP on individual scenes. Computing pose errors w.r.t. COLMAP would result in incorrect conclusions. Therefore, we gauge the pose quality in a self-supervised way, using novel view synthesis [96].

We let each method estimate the poses of all images of a scene. For evaluation, we split the images into training and test sets. We train a Nerfacto [89] model on the training set, and synthesize views for the test poses. We compare the synthesized images to the test images and report the difference as peak signal-to-noise ratio (PSNR). To ensure a fair comparison to NeRF-based competitors, we use these methods in the same way as the other SfM methods: we run them to pose all images, and train a Nerfacto model on top of their poses. This is to ensure that we solely compare the pose quality across methods, and not different capabilities in view synthesis. The quality of poses affects the PSNR numbers in two ways: Good training poses let the NeRF model fit a consistent scene representation. Good testing poses make sure that the synthesized images are aligned with the original image. We explain further details in the supplement which also includes additional, perceptual metrics. In spirit, our evaluation is similar to the Tanks and Temples benchmark which evaluates a derived scene mesh rather than camera poses. However, our evaluation can be applied to arbitrary datasets as it does not need ground truth geometry.

	Frames	Pseudo Ground Truth		All Frames				200 Frames			50 Frames	
		Kinect Fusion	COLMAP (default)	COLMAP (fast)	DROID-SLAM† [90]	ACE0 (ours)	KF+ACE0 (ours)	BARF [54]	NoPE-NeRF† [10]	ACE0 (ours)	DUSt3R [97]	ACE0 (Ours)
Chess	6k	19.6	23.6	23.5	19.3	23.3	23.0	12.8	12.6	**22.7**	18.9	**19.2**
Fire	4k	19.2	22.6	22.6	13.0	22.3	22.3	12.7	11.8	**22.1**	18.8	**19.5**
Heads	2k	17.0	18.8	18.9	17.6	18.8	19.1	10.7	11.8	**19.9**	18.4	**21.3**
Office	10k	18.9	21.4	21.6	failed	21.1	21.5	11.9	10.9	**19.8**	12.5	**13.7**
Pumpkin	6k	19.9	24.1	23.8	18.3	24.1	23.8	19.6	14.2	**24.7**	21.7	**22.3**
RedKitchen	12k	17.6	21.4	21.4	10.9	20.8	20.9	11.6	11.2	**18.9**	**13.8**	13.7
Stairs	3k	19.0	16.7	21.0	13.0	17.7	19.9	15.8	15.9	**18.8**	15.3	**15.4**
Average		18.7	21.2	21.8	N/A	21.2	21.5	13.6	12.6	**21.0**	17.1	**17.9**
Avg. Time		realtime	38h	13h	18min	1h	7min	8.5h	47h	**27min**	**4min***	16min

Table 1: 7-Scenes. We show the pose accuracy via view synthesis with Nerfacto [89] as PSNR in dB, and the reconstruction time. Results for *All Frames* are color coded w.r.t. similarity to the COLMAP pseudo ground truth: $>$ 0.5 dB better within $\pm$0.5 dB $>$ 0.5 dB worse $>$1 dB 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.

4.1 7-Scenes

The 7-Scenes dataset [81] consists of seven indoor scenes, scanned with a Kinect v1 camera. Multiple, disconnected scans are provided for each scene to a total of 2k-12k images. For each method, we assume a shared focal length across scans and initialize with the default calibration of a Kinect v1. The dataset comes with pseudo ground truth camera poses estimated by KinectFusion [44, 64], a depth-based SLAM system. Individual scans were registered but not bundle-adjusted [13]. Inspired by [13], we recompute alternative, bundle-adjusted pseudo ground truth by running COLMAP with default parameters.

Discussion. We show results in Table 1. Of both pseudo ground truth versions, KinectFusion achieves lower PSNR numbers than COLMAP, presumably due to the lack of global optimization. COLMAP with *fast* parameters shows PSNR numbers similar to COLMAP with *default* parameters, on average. Both versions of running COLMAP take considerable time to reconstruct each scene. We note that COLMAP has been optimised for quality, rather than speed. Not all acceleration strategies from the feature-based SfM literature have been implemented in COLMAP, so presumably comparable quality can be obtained faster. DROID-SLAM [90] does not perform well on 7-Scenes and partially fails altogether, presumably due to the jumps between individual scans of each scene.

Our approach, ACE0, achieves a pose quality comparable to the COLMAP pseudo ground truth while reconstructing each scene in ∼1 hour despite the large number of images. We show qualitative examples in Figure 4 and in the supplement. We also demonstrate that ACE0 can swiftly optimize an initial set of approximate poses. When starting from KinectFusion poses, ACE0 increases PSNR significantly in less than 10 minutes per scene, see "KF+ACE0" in Table 1. In the supplement, we include a parameter study on 7-Scenes to show that ACE0 is robust to the choice of depth estimator. We also show the positive impact of pose refinement on the reconstruction quality as well as the reconstruction speedup due to our early stopping schedule.

	ACE [12]	ACE [12]	ACE0 (ours)
Supervision	KinectFusion	COLMAP	*self-supervised*
Chess	96.0 %	**100.0 %**	**100.0 %**
Fire	98.4 %	**99.5 %**	98.8 %
Heads	**100.0 %**	**100.0 %**	**100.0 %**
Office	36.9 %	**100.0 %**	99.1 %
Pumpkin	47.3 %	**100.0 %**	99.9 %
Redkitchen	47.8 %	**98.9 %**	98.1 %
Stairs	74.1 %	**85.0 %**	61.0 %
Average	71.5 %	**97.6 %**	93.8 %

Table 2 (a): Relocalization on 7-Scenes. % poses below 5cm, 5° error, computed w.r.t. COLMAP pseudo GT.

	Pseudo GT (COLMAP)	DROID-SLAM† [90]	BARF [54]	NoPe-NeRF† [10]	ACE0 (ours)
Bicycle	21.5	10.9	11.9	12.2	**18.7**
Bonsai	27.6	10.9	12.5	14.8	**25.8**
Counter	25.5	12.9	11.9	11.6	**24.5**
Garden	26.3	16.7	13.3	13.8	**25.0**
Kitchen	27.4	13.9	13.3	14.4	**26.1**
Room	28.0	11.3	11.9	14.3	**19.8**
Stump	16.8	13.9	15.0	13.7	**20.5**
Average	24.7	12.9	12.8	13.5	**22.9**

Table 2 (b): Mip-NeRF 360. Pose quality in PSNR, higher is better. Best in **bold**. †Method needs sequential inputs.

For our learning-based competitors, we sub-sampled images due to their computational constraints, see right side of Table 1. Even using only 200 images, NoPe-NeRF [10] takes 2 days to fit a model and estimate poses. Despite these long processing times, we observe poor pose quality of BARF and NoPe-NeRF in terms of PSNR. BARF [54] requires pose initialisation. We provide identity poses since the scenes of 7Scenes are roughly forward-facing. Still, the camera motions are too complex for BARF to handle. NoPe-NeRF does not require pose initialisation but needs roughly sequential images, which we did provide. NoPe-NeRF relies upon successive images having similar poses to perform well, and struggles with the large jumps between the subsampled images.

For the comparison with DUSt3R [97], we had to subsample the sequences further, as we were only able to run it with 50 images at most, even when using an A100 GPU with 40GB memory. DUSt3R achieves reasonable PSNR numbers but consistently lower than ACE0.

Relocalization. ACE0 is a learning-based SfM tool but it is also a self-supervised visual relocaliser. In Table 2 (a), we compare it to the supervised relocalizer ACE [12]. Using the scale-metric pseudo ground truth of [13], we train ACE with COLMAP mapping poses, and evaluate it against COLMAP query poses. Unsurprisingly, ACE achieves almost perfect relocalization under the usual 5cm, 5° error threshold. Interestingly, ACE0 achieves almost identical results when mapping the scene self-supervised, and evaluating the relocalized query poses against the COLMAP pseudo ground truth. For context, when training ACE with KinectFusion mapping poses and evaluating against COLMAP pseudo ground truth, results are far worse. This signifies that ACE0 mapping poses are very similar to the COLMAP mapping poses, and less similar to KinectFusion mapping poses. We give more details about this experiment in the supplement.

4.2 Mip-NeRF 360

The Mip-NeRF 360 dataset [5] consists of seven small-scale scenes, both indoor and outdoor. The dataset was reconstructed with COLMAP and comes with intrinsics (which we ignore), camera poses and undistorted (pinhole camera) images. For each method, we assume a shared focal length per scene.

Discussion. We present PSNR results in Table 2 (b). NoPe-NeRF does not perform well on this dataset, despite processing each scene for 2 days. The differences

		Frames	COLMAP (default)	Reality Capture	DROID-SLAM† [90]	ACE0 (ours)	Frames	COLMAP (fast)	Reality Capture	DROID-SLAM† [90]	ACE0 (ours)	Sparse COLMAP + Reloc + BA	Sparse COLMAP + ACE0 (ours)
Training	Barn	410	24.0	21.2	19.0	16.5	12.2k	24.4	16.9	13.5	17.7	26.3	25.1
	Caterpillar	383	17.1	15.9	16.6	16.9	11.4k	18.6	17.9	18.9	18.6	18.7	18.8
	Church	507	18.3	17.6	14.3	17.2	19.3k	12.1	-	11.5	16.5	18.5	17.3
	Ignatius	264	20.1	17.7	17.8	19.8	7.8k	20.8	18.6	19.1	20.7	20.9	20.7
	Meetingroom	371	18.6	18.1	15.6	18.0	11.1k	19.4	18.2	17.1	16.6	20.8	20.3
	Truck	251	21.1	19.0	18.3	20.1	7.5k	23.6	19.1	20.6	23.0	23.4	23.1
	Average	364	19.9	18.2	16.9	18.1	14.6k	19.8	18.2	16.8	18.9	21.4	20.9
	Avg Time		1h	3min	5min	1.1h		74h	14h	18min	2.2h	8h	1.8h
Intermediate	Family	152	19.5	18.8	17.6	19.0	4.4k	21.2	19.8	19.8	18.0	21.3	21.3
	Francis	302	21.6	20.7	20.7	20.1	7.8k	19.9	20.4	21.8	21.7	22.5	22.7
	Horse	151	19.2	19.0	16.3	19.5	6.0k	21.6	20.7	19.2	21.7	22.6	22.3
	Lighthouse	309	16.6	16.5	13.6	17.5	8.3k	19.0	16.6	18.9	18.6	19.5	20.5
	Playground	307	19.1	19.2	11.4	18.7	7.7k	17.9	16.5	11.3	20.4	21.2	21.0
	Train	301	16.8	15.4	13.8	16.2	12.6k	19.6	14.4	15.6	18.5	19.8	18.5
	Average	254	18.8	18.3	15.6	18.5	7.8k	19.9	18.1	17.8	19.8	21.1	21.0
	Avg Time		32min	2min	3min	1.3h		48h	11h	14min	2.2h	5h	1h
Advanced	Auditorium	302	19.6	12.2	16.7	18.7	13.6k	13.7	-	16.6	20.0	21.4	19.8
	Ballroom	324	16.3	18.3	13.1	17.9	10.8k	17.2	-	10.4	18.9	18.0	15.6
	Courtroom	301	18.2	17.2	12.3	17.1	12.6k	14.6	-	10.2	16.3	18.7	17.8
	Palace	509	14.2	11.7	10.8	10.7	21.9k	13.8	-	8.6	11.0	15.3	12.3
	Temple	302	18.1	15.7	11.8	9.7	17.5k	13.3	-	11.9	14.8	19.6	16.1
	Average	348	17.3	15.0	12.9	14.8	15.6k	14.5	-	11.5	16.2	18.6	16.3
	Avg Time		1h	2min	4min	1h		71h		27min	2.8h	10h	2.1h

Table 3: Tanks and Temples. We show the pose accuracy via view synthesis with Nerfacto [89] as PSNR in dB, and the reconstruction time. We color code results compared to COLMAP, *default* and *fast*, respectively: > 0.5 dB better within ±0.5 dB > 0.5 dB worse >1 dB worse. †Method needs sequential inputs.

between sequential images are too large. DROID-SLAM fails for the same reason. BARF performs poorly because the identity pose is not a good initialisation for most scenes. In contrast, ACE0 reconstructs the dataset successfully. While it achieves slightly lower PSNR than COLMAP, its pose estimates are similar, *cf*. Figure 4. The supp. shows that synthesized images based on ACE0 are visually close to those of COLMAP while our learning-based competitors are far off.

4.3 Tanks and Temples

The Tanks and Temples dataset [49] contains 21 diverse scenes, both indoors and outdoors, and with varying spatial extent. We remove two scenes (Panther, M60) with military associations. We also remove two scenes (Courthouse, Museum) where the COLMAP baseline did not finish the reconstruction after 5 days of processing or ran out of memory. For the latter two scenes, we provide ACE0 results in the supplement. The dataset provides 150-500 images per scene but also the original videos as a source for more frames. Thus, we consider each scene in two versions: Using 150-500 images and using 4k-22k frames. For all methods, we assume a pinhole camera model with shared focal length across images, and images to be unordered. We found none of the learning-based SfM competitors to be applicable to this dataset. NoPe-NeRF would run multiple days per scene, even when considering only a few hundred images. DUSt3R would run out of memory. BARF needs reasonable pose initializations which are not available.

Discussion. We show PSNR numbers of RealityCapture, DROID-SLAM and ACE0 in Table 3, color-coded by similarity to the COLMAP pseudo GT. ACE0 achieves reasonable results when reconstructing scenes from a few hundred images (Table 3, left). ACE0 generally predicts plausible poses (*cf*., Figure 4), even if PSNR numbers are sometimes lower than those of COLMAP. RealityCapture performs similar to ACE0 while DROID-SLAM struggles on the sparse images.

Next, we consider more than 1k images per scene (right side of Table 3). Here, we run COLMAP with parameters tuned for large images collections (*fast)* due to the extremely large image sets. ACE0 offers a reconstruction quality comparable to COLMAP on average while also being fast. We run RealityCapture on some of the scenes but it produces fractured reconstructions for these large images sets, leading to low PSNR numbers. DROID-SLAM still struggles on many scenes despite having access to sequential images that are temporally close.

In the two rightmost columns, we initialise with a sparse COLMAP reconstruction from 150-500 images, and extend and refine it using all available frames. Firstly, using a feature-based baseline, we register the full set of frames using the relocalization mode of COLMAP, followed by a final round of bundle adjustment. Secondly, we run ACE0 initialised with the poses of the sparse COLMAP reconstruction. Both variants are considerably faster than running COLMAP from scratch on the full set of frames. ACE0 is able to register and refine all additional frames in 1-2 hours, on average. Again, we find the pose quality of ACE0 comparable to the feature-based alternative.

5 Conclusion and Future Work

We have presented scene coordinate reconstruction, a new approach to learning-based SfM. We learn an implicit, neural scene representation from a set of unposed images. Our method, ACE0, is able to reconstruct a wide variety of scenes. In many cases, the accuracy of estimated poses is close to that of COLMAP and synthesized images visually similar. Unlike previous learning-based SfM methods, ACE0 can be applied to multiple thousand unsorted images, without pose priors, and reconstructs them within a few hours.

Limitations. We show some failure cases in the supplement. Scene coordinate regression struggles with repetitive structures since the network is not able to make multi-modal predictions for visually ambiguous inputs. Scene coordinate regression also struggles with representing large areas. The common solution is to use network ensembles based on pre-clustering of the scene [12, 16] which is difficult in a reconstruction setting. While scene coordinate regression generalises quite well, it has difficulties to bridge extreme view point or lighting changes, such as day versus night. In our experiments, we assumed a simple pinhole camera model with shared intrinsics across images. To the best of our knowledge, scene coordinate regression has not been coupled with image distortion, thus far.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a day. ACM TOG (2011)
2. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: ECCV (2010)
3. Arnold, E., Wynn, J., Vicente, S., Garcia-Hernando, G., Monszpart, A., Prisacariu, V.A., Turmukhambetov, D., Brachmann, E.: Map-free visual relocalization: Metric pose relative to a single image. In: ECCV (2022)
4. Balntas, V., Li, S., Prisacariu, V.A.: RelocNet: Continuous metric learning relocalisation using neural nets. In: ECCV (2018)
5. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
6. Beardsley, P.A., Zisserman, A., Murray, D.W.: Sequential updating of projective and affine structure from motion. IJCV (1997)
7. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: ZoeDepth: Zero-shot transfer by combining relative and metric depth. arXiv (2023)
8. Bhowmick, B., Patra, S., Chatterjee, A., Govindu, V.M., Banerjee, S.: Divide and conquer: Efficient large-scale structure from motion using graph partitioning. In: ACCV (2015)
9. Bhowmick, B., Patra, S., Chatterjee, A., Govindu, V.M., Banerjee, S.: Divide and conquer: A hierarchical approach to large-scale structure-from-motion. CVIU (2017)
10. 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)
11. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: CodeSLAM — Learning a compact, optimisable representation for dense visual SLAM. In: CVPR (2018)
12. Brachmann, E., Cavallari, T., Prisacariu, V.A.: Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In: CVPR (2023)
13. Brachmann, E., Humenberger, M., Rother, C., Sattler, T.: On the limits of pseudo ground truth in visual camera re-localisation. In: ICCV (2021)
14. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: DSAC-differentiable RANSAC for camera localization. In: CVPR (2017)
15. Brachmann, E., Rother, C.: Learning less is more-6D camera localization via 3D surface regression. In: CVPR (2018)
16. Brachmann, E., Rother, C.: Expert sample consensus applied to camera relocalization. In: ICCV (2019)
17. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE TPAMI (2021)
18. Brégier, R.: Deep regression on manifolds: a 3D rotation case study. In: 3DV (2021)
19. Brown, D.: The bundle adjustment-progress and prospect. In: Congr. of the Int. Soc. for Photogr. (1976)
20. Brown, M., Lowe, D.G.: Unsupervised 3D object recognition and reconstruction in unordered datasets. In: 3DIM (2005)
21. Carlone, L., Tron, R., Daniilidis, K., Dellaert, F.: Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In: ICRA (2015)

22. Cavallari, T., Bertinetto, L., Mukhoti, J., Torr, P.H., Golodetz, S.: Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In: 3DV (2019)
23. Cavallari, T., Golodetz, S., Lord, N.A., Valentin, J., Di Stefano, L., Torr, P.H.: On-the-fly adaptation of regression forests for online camera relocalisation. In: CVPR (2017)
24. Chen, S., Bhalgat, Y., Li, X., Bian, J., Li, K., Wang, Z., Prisacariu, V.A.: Neural refinement for absolute pose regression with feature synthesis. In: CVPR (2024)
25. Chen, S., Li, X., Wang, Z., Prisacariu, V.: DFNet: Enhance absolute pose regression with direct feature matching. In: ECCV (2022)
26. Chen, S., Wang, Z., Prisacariu, V.: Direct-PoseNet: Absolute pose regression with photometric consistency. In: 3DV (2021)
27. Cheng, Z., Esteves, C., Jampani, V., Kar, A., Maji, S., Makadia, A.: LU-NeRF: Scene and pose estimation by synchronizing local unposed NeRFs. In: ICCV (2023)
28. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: CVPR (2011)
29. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: CVPR (2017)
30. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: ICCV (2003)
31. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-supervised interest point detection and description. In: CVPRW (2018)
32. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: CamNet: Coarse-to-fine retrieval for camera re-localization. In: ICCV (2019)
33. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable CNN for joint description and detection of local features. In: CVPR (2019)
34. 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)
35. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE TPAMI (2003)
36. Gherardi, R., Farenzena, M., Fusiello, A.: Improving the efficiency of hierarchical structure-and-motion. In: CVPR (2010)
37. Govindu, V.M.: Combining two-view constraints for motion estimation. In: CVPR (2001)
38. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: CVPR (2004)
39. Hartley, R., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. IJCV (2013)
40. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
41. He, X., Sun, J., Wang, Y., Peng, S., Huang, Q., Bao, H., Zhou, X.: Detector-free structure from motion. In: CVPR (2024)
42. Heinly, J., Schönberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the World in six days. In: CVPR (2015)
43. Humenberger, M., Cabon, Y., Pion, N., Weinzaepfel, P., Lee, D., Guérin, N., Sattler, T., Csurka, G.: Investigating the role of image retrieval for visual localization: An exhaustive benchmark. IJCV (2022)

44. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A.J., Fitzgibbon, A.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: UIST (2011)
45. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: ICCV (2021)
46. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice. IJCV (2021)
47. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In: ICCV (2015)
48. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. ACM TOG (2023)
49. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and Temples: Benchmarking large-scale scene reconstruction. ACM TOG (2017)
50. Kraus, K.: Photogrammetry. No. v. 1 in Photogrammetry, Ferdinand Dummlers Verlag (1993)
51. Laskar, Z., Melekhov, I., Kalia, S., Kannala, J.: Camera relocalization by computing pairwise relative poses using convolutional neural network. In: ICCV Workshops (2017)
52. Li, X., Wang, S., Zhao, Y., Verbeek, J., Kannala, J.: Hierarchical scene coordinate classification and regression for visual localization. In: CVPR (2020)
53. Lin, A., Zhang, J.Y., Ramanan, D., Tulsiani, S.: Relpose++: Recovering 6d poses from sparse-view observations. In: 3DV (2024)
54. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: BARF: Bundle-adjusting neural radiance fields. In: ICCV (2021)
55. Lin, Y., Müller, T., Tremblay, J., Wen, B., Tyree, S., Evans, A., Vela, P.A., Birchfield, S.: Parallel inversion of neural radiance fields for robust pose estimation. In: ICRA (2023)
56. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local feature matching at light speed. In: ICCV (2023)
57. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
58. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
59. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR (2007)
60. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: GNeRF: GAN-based Neural Radiance Field without Posed Camera. In: ICCV (2021)
61. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
62. Moreau, A., Piasco, N., Bennehar, M., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: CROSSFIRE: Camera relocalization on self-supervised features from an implicit representation. ICCV (2023)
63. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022)
64. Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: ISMAR (2011)
65. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: CVPR (2004)
66. Pollefeys, M., Koch, R., Vergauwen, M., Van Gool, L.: Automated reconstruction of 3D scenes from sequences of images. J. of Photogr. and Rem. Sens. (2000)

67. Rau, A., Garcia-Hernando, G., Stoyanov, D., Brostow, G.J., Turmukhambetov, D.: Predicting visual overlap of images through interpretable non-metric box embeddings. In: ECCV (2020)
68. Reality, C.: Reality Capture. https://www.capturingreality.com/realitycapture (2016), [accessed 15-Nov-2023]
69. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In: ICCV (2021)
70. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: CVPR (2019)
71. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning feature matching with graph neural networks. In: CVPR (2020)
72. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T.: Back to the feature: Learning robust camera localization from pixels to pose. In: CVPR (2021)
73. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2D-to-3D matching. In: ICCV (2011)
74. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: ECCV (2012)
75. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE TPAMI (2016)
76. Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., Pajdla, T.: Are large-scale 3D models really necessary for accurate visual localization? In: CVPR (2017)
77. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: CVPR (2019)
78. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "How do i organize my holiday snaps?". In: ECCV (2002)
79. Schönberger, J.L.: Colmap Github Issues. https://github.com/colmap/colmap/issues/116#issuecomment-298926277 (2017), [accessed Nov/15/2023]
80. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
81. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: CVPR (2013)
82. Sinha, S., Zhang, J.Y., Tagliasacchi, A., Gilitschenski, I., Lindell, D.B.: Sparse-Pose: Sparse-view camera pose regression and refinement. In: CVPR (2023)
83. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Arti. Intell. and Mach. Learn. for Multi-Domain Operations Appli. (2019)
84. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. ACM TOG (2006)
85. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. IJCV (2008)
86. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR (2008)
87. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. In: CVPR (2021)
88. Szeliski, R., Kang, S.B.: Recovering 3D shape and motion from image streams using nonlinear least squares. J. of Vis. Com. and Image Repr. (1994)

89. 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)
90. Teed, Z., Deng, J.: DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In: NeurIPS (2021)
91. Toldo, R., Gherardi, R., Farenzena, M., Fusiello, A.: Hierarchical structure-and-motion recovery from uncalibrated images. CVIU (2015)
92. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: Int. Worksh. on Vis. Alg. (2000)
93. Türkoğlu, M.Ö., Brachmann, E., Schindler, K., Brostow, G., Monszpart, A.: Visual camera re-localization using graph neural networks and relative pose supervision. In: 3DV (2021)
94. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: CVPR (2018)
95. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: DeMoN: Depth and motion network for learning monocular stereo. In: CVPR (2017)
96. Waechter, M., Beljan, M., Fuhrmann, S., Moehrle, N., Kopf, J., Goesele, M.: Virtual rephotography: Novel view prediction error for 3d reconstruction. ACM TOG (2017)
97. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3D vision made easy. In: CVPR (2024)
98. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF--: Neural radiance fields without known camera parameters. arXiv (2021)
99. Wei, X., Zhang, Y., Li, Z., Fu, Y., Xue, X.: DeepSFM: Structure from motion via deep bundle adjustment. In: ECCV (2020)
100. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV (2013)
101. Xia, Y., Tang, H., Timofte, R., Van Gool, L.: SiNeRF: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. In: BMVC (2022)
102. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: iNeRF: Inverting neural radiance fields for pose estimation. In: IROS (2021)
103. Zhang, J.Y., Lin, A., Kumar, M., Yang, T.H., Ramanan, D., Tulsiani, S.: Cameras as rays: Pose estimation via ray diffusion. In: ICLR (2024)
104. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT (2006)
105. Zhou, Q., Sattler, T., Pollefeys, M., Leal-Taixe, L.: To learn or not to learn: Visual localization from essential matrices. In: ICRA (2020)
106. Zhou, Y., Barnes, C., Jingwan, L., Jimei, Y., Hao, L.: On the continuity of rotation representations in neural networks. In: CVPR (2019)