Sparse–View Localization via Online Neural 3D Regression

Ludvig Dillén Magnus Oskarsson Viktor Larsson
Centre for Mathematical Sciences, Lund University
{ludvig.dillen, magnus.oskarsson, viktor.larsson}@math.lth.se

Abstract

We present ON3R, an online-trained neural regressor addressing sparse-view structureless localization, where database images have limited visual overlap and no pre-built 3D map. Given any sparse matches between a query and a K-tuple of posed database views, ON3R predicts 3D coordinates for matched query keypoints, supervised by database reprojection residuals and a monocular depth prior. Afterwards, the absolute pose of the query is estimated via P3P-RANSAC and refined with lightweight bundle adjustment. Across MegaDepth, Cambridge Landmarks, and a sparsified version of Aachen Day-Night, ON3R outperforms existing methods. ON3R is particularly effective when the data is extremely sparse – we focus on $K \leq 10$ database images. The code is available at https://github.com/ludvigdillen/ON3R.

1. Introduction

Finding the camera pose of a query image has been addressed by multiple types of methods over the years [37, 43, 52, 54]. All such methods require some representation of the world and/or posed database images defining the coordinate system. If the world is represented by estimated 3D points, we call the method *structure-based*; if not, it is *structureless*. Most research today focuses on structure-based localization methods known for their high accuracy, whereas structureless approaches remain underexplored [33]. However, in sparse-view settings, low visual overlap makes triangulation unreliable and consequently structure-based methods brittle. Beyond this, structureless localization methods often offer advantages over their structure-based counterparts in terms of reduced precomputation, easier database maintenance, and lower storage requirements [13]. This work addresses sparse-view structureless localization and presents a new method which we call ON3R (Online Neural 3D Regression).

Under curated data acquisition, structure-based methods typically dominate due to their accuracy in settings with rich covisibility [37, 43]. However, in surveillance or stationary camera setups, cameras are commonly placed with limited overlap to minimize cost, resulting in sparse coverage (camera poses are typically obtained through offline manual calibration due to the limited image overlap). In such scenarios, localizing against live imagery is often required due to frequent scene changes (*e.g.*, vehicles in parking lots or packages on factory floors). Sparse-view settings also occur when storing all captured images is infeasible, *e.g.*, when autonomous platforms rapidly acquire large volumes of data but, due to storage or transmission constraints, retain only a sparse subset. In this setting, the data often comes from multi-camera systems with naturally low overlap, and poses are obtained from on-device odometry. We target this

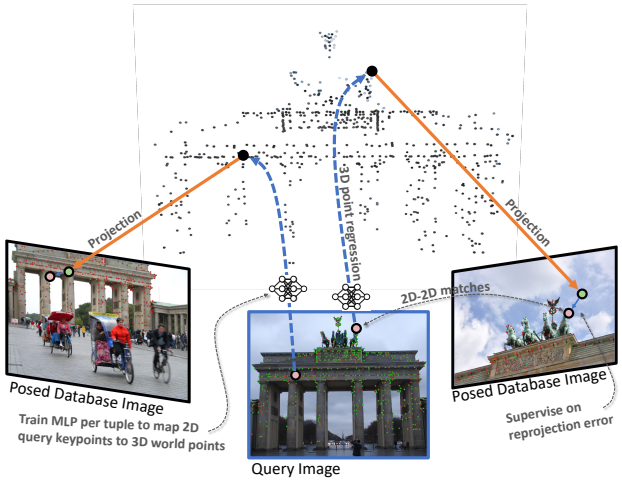

Figure 1. **Regressing 2D-3D correspondences with ON3R.** ON3R estimates the absolute pose of a query image by comparing it to K posed database images and remains robust even when the database images have little or no mutual overlap (*i.e.*, *star-topology* data). As input, ON3R takes sparse matches (no visual features) between the query and each database image. For every matched query keypoint, ON3R regresses a 3D point by training a compact MLP on-the-fly, supervised by reprojection errors and a monocular depth prior. The resulting 2D–3D matches are then used to estimate the absolute pose, which, together with the regressed 3D points, is refined with lightweight bundle adjustment.

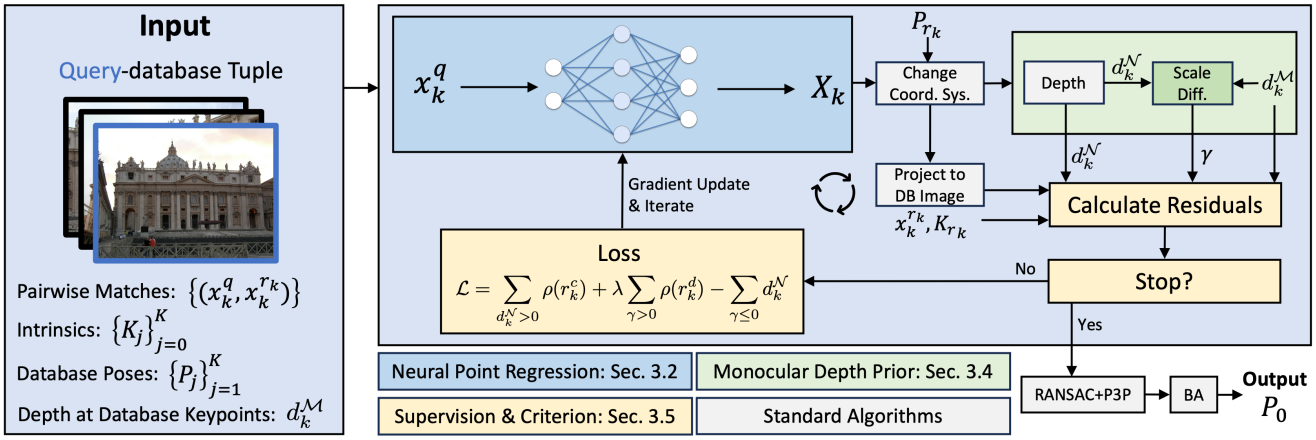

Figure 2. **Flowchart** of ON3R illustrating how our lightweight MLP that is trained per query-database tuple, regresses 3D points using reprojection and depth (estimated using MoGe-2 [53]) residuals. Once convergence is reached (low residuals, no improvement, or maximum epochs), the absolute pose is estimated from the resulting 2D-3D correspondences and subsequently refined through bundle adjustment.

regime and propose ON3R: it online learns 2D-3D correspondences supervised by database reprojection errors and a monocular depth prior, then estimates the pose via robust estimation followed by lightweight bundle adjustment – integrating as a drop-in pose estimator after standard image retrieval and matching.

In Fig. 1 and Fig. 3, typical sparse-view settings are shown. We call such a data structure *star-topology*, *i.e.*, the query image overlaps with each retrieved database image, while the database images have *little* or *no overlap* with each other. For star-topology data, structure-based methods can triangulate only a limited – or even no – 3D model. Current structureless methods, however, can be used to get the absolute pose of a query image but lack performance. In these challenging cases where existing methods fail, ON3R maintains competitive performance.

The core idea of ON3R is that when reasoning about the query-database geometry, it can be beneficial to do so jointly over database images, even if their overlap is limited. Namely, ON3R regresses 3D correspondences for matched query keypoints and uses the smoothness of neural networks to reason about 3D space. The query keypoints are lifted into 3D space with an MLP, which is trained per query supervised by reprojection residuals in database images. To be even more robust to star-topology data where depth is ambiguous, we also use a monocular depth prior. Finally, the absolute pose is obtained via P3P-RANSAC [12, 17] on the 2D-3D correspondences, followed by pose refinement with bundle adjustment. See Fig. 2 for an overview of ON3R.

Our main contributions are

- We propose a novel structureless localization framework designed for *star-topology* data, where database images have little or no mutual overlap.
- We introduce a new perspective on camera localization, where a neural network is learned from *online-guided explicit geometric supervision* instead of relying on prebuilt 3D maps or fragile pose regression techniques.
- ON3R achieves state-of-the-art performance on challenging sparse-view localization benchmarks, including settings where structure-based methods fail.

We hope this work will encourage further research on localization for sparse-view scenarios with explicit geometry.

2. Related Work

Beyond the structure-based and structureless categorization, it is also possible to categorize methods by their network's scene dependency. Methods like absolute pose regression and scene coordinate regression learn different network weights for each scene, making them *scene-specific*. In contrast, methods such as PixLoc [39] and HLOC [37] require a 3D map to localize but have no scene-specific weights, making them *scene-agnostic*. Conversely, ON3R learns a query-specific network.

Important localization aspects – including privacy, scalability, maintainability, flexibility, speed, storage efficiency, and accuracy – largely depend on the chosen method category. More concretely, for structure-based or scene-specific methods, removing an image from the database requires updating the corresponding 3D descriptors or network weights. In other cases, such as rapidly changing scenes, continuously reconstructing SfM maps or retraining large networks becomes computationally expensive.

2.1. Structure-based Localization

Traditional image-based localization methods rely on incrementally [43] or globally [32] constructed 3D maps. State-of-the-art pipelines such as HLOC [37] use these maps to determine the absolute pose of a query image.

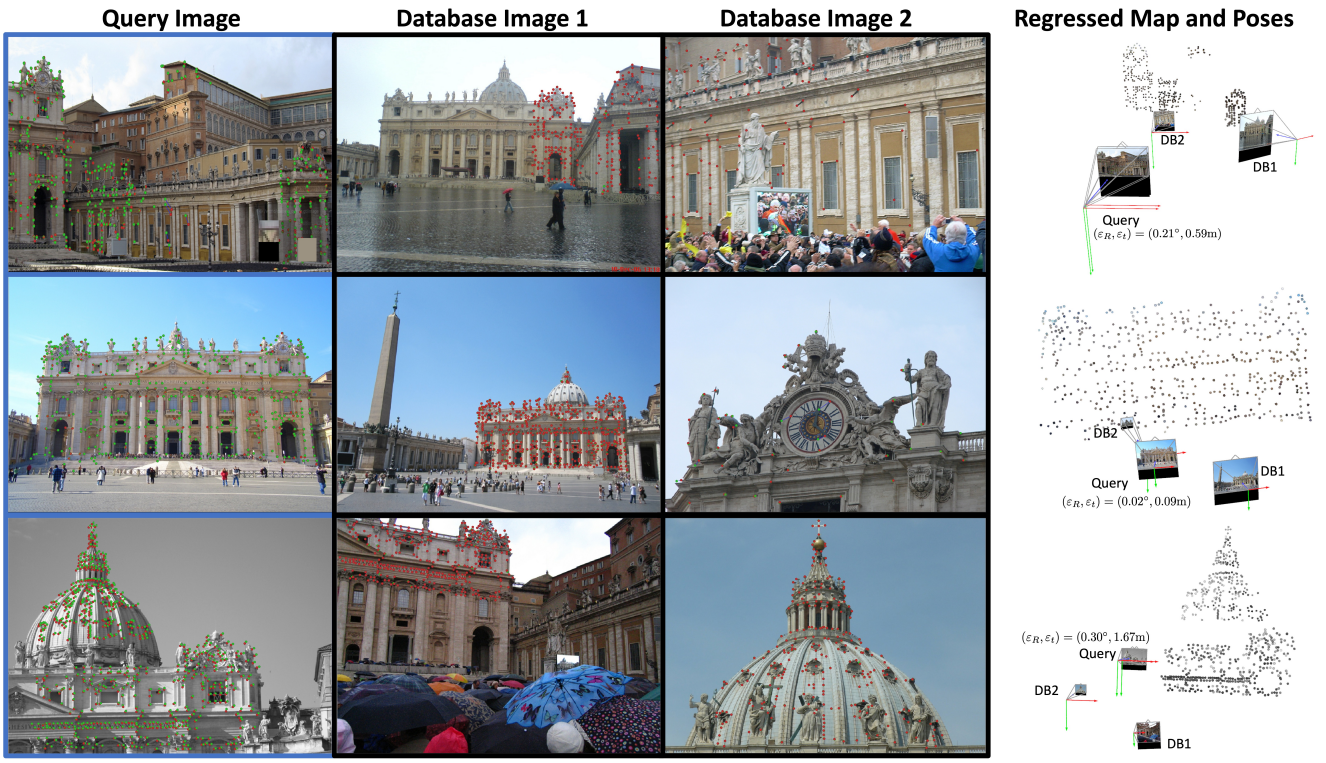

Figure 3. **Example tuples**. Each row presents a test query and a 2-tuple of database images from the MegaDepth dataset and the model output. Keypoints with matches are shown in red, projected 3D points in green, and residuals in blue (barely visible due to their small magnitude). For the query images, keypoints are projected into the ground truth pose for visualization. The 3D model is optimized during training, first with a coarse network prediction, followed by refinement via bundle adjustment. In the scene visualizations, we show the true poses of the query and database images, as well as the estimated query pose (not visible for the second row due to the viewing angle).

Given a query, HLOC [37] first retrieves the K most similar database images. A sparse matcher – such as SuperGlue [38] or LightGlue [30] – on top of descriptors like SuperPoint [10] or DISK [49] is then used for 2D-2D image matching. Each matched database keypoint with an associated 3D point provides a 2D-3D correspondence. All such correspondences are subsequently passed to RANSAC [17, 25] with a P3P solver [12, 35] to find the absolute pose of the query image. These methods remain state-of-the-art for large-scale localization but are inflexible, computation-heavy, and require significant engineering and storage overhead.

2.2. Scene Coordinate Regression

We call scene coordinate regression (SCR) methods *implicit structure-based* as they learn a neural representation of a scene using posed database images without storing explicit scene points [2–4, 28, 46, 50]. A scene-specific network is trained on top of a scene-agnostic feature extractor by predicting 3D coordinates from image patches, supervised using reprojection errors. The depth of 3D points should be ambiguous as only single-view constraints are used, but it can be inferred due to neural network smoothness. This is similar to ON3R, with the key difference that in SCR, smoothness is enforced over visually similar patches, whereas ON3R is smooth in the spatial domain. SCR methods are faster than structure-based counterparts for query pose estimation, thanks to a scene-agnostic feature backbone [3], the single-view constraints, and only needing one inference pass to get 2D-3D matches. Nevertheless, they typically require hundreds of database images and several minutes [3] to learn a scene-specific network (in contrast to our query-specific network). Thus, they are inherently scene-specific and impractical in sparse-view or dynamic settings, as we also show in our experiments.

2.3. Structureless Localization

Structureless localization methods operate without a prebuilt 3D map, placing ON3R in this category.

Pose regression can be divided into two categories: absolute pose regression (APR) [8, 22, 23] and relative pose regression (RPR) [11, 14, 26]. RPR methods take an image pair as input and predict the relative transformation between their camera poses. These approaches are typically fast since they solve the problem end-to-end with a neural network, though it often comes at the cost of interpretability.

Laskar et al. [26] retrieve the most similar database images and then use a CNN to regress the relative pose between the query and retrieved images. The more recent method Reloc3r [14] uses a vision transformer backbone encoder and decoder [15, 54–56] to encode the image pair, to then obtain the relative pose with a pose regression head. RPR methods are suitable as baselines in our setup, as relative poses between a query and database images can be aggregated via motion averaging to an absolute query pose estimate.

APR methods, in contrast, regress the absolute pose of a single image. PoseNet [23] pioneered this line of work by solving the problem fully end-to-end with a neural network. More recently, Chen et al. [8] observed that APR networks often struggle when trained with limited data per scene. To address this, they proposed combining a scene-specific representation based on scene coordinate regression with a scene-agnostic absolute pose regression network, thereby improving performance in data-scarce settings. However, APR methods still require substantially more than just a few images, making them unsuitable for our sparse-view setup.

Foundation models have in the past few years profoundly impacted the computer vision community. With the introduction of DUSt3R [54], a wide range of 3D tasks became directly solvable using an end-to-end neural network and well-known algorithms. Following its release, numerous extensions and derivatives have been proposed [5, 19, 21, 27, 48, 51, 52, 57] building upon DUSt3R's model and dense correspondence fields. VGGT [52] extends DUSt3R to multi-view settings, taking an arbitrary number of input images and jointly predicting camera parameters , depth maps, point maps, and point tracks.

SfM models on the fly can be created using transitive matching, meaning that whenever a query keypoint matches at least two database images, an implicit 2D-3D correspondence is obtained as the 3D point can be triangulated from the posed database images [18, 33, 36, 41]. The absolute pose of the query is then estimated with P3P inside RANSAC followed by potential bundle adjustment. After that, a second pass of the method can be run with all query keypoints added to the tracks, creating additional tracks and more constraints for already triangulated 3D points. This second pass is crucial when the number of tracks from the first pass is limited. Once again, after 2D-3D matches have been established and the refined query pose estimated, bundle adjustment can be run.

Motion averaging can be used when relative poses between the query and the retrieved images have been estimated [13, 26, 33, 58]. Typically, the global rotations are first estimated using rotation averaging [6, 7, 31] after which the global translations are recovered by triangulating camera centers in a common coordinate frame [6, 13].

3. Method

We will now describe ON3R, which takes a K-tuple of posed database images and estimates the camera pose of a query image. The core component is the regression of 3D coordinates, for all query keypoints. This is done with a compact neural network ($\mathcal{N}$), trained online for each query. Given sparse matches and a monocular depth prior, the absolute pose of the query camera is estimated using the 2D-3D correspondences. Afterwards, the solution is refined with bundle adjustment. An overview of ON3R is given in Fig. 2. In the following sub-sections, we present our approach in detail.

3.1. Problem Formulation

Our goal is to estimate the 6-DoF world-to-camera pose $P_0 = [R_0 \,|\, t_0]$, where $R_0 \in \mathrm{SO}(3)$, $t_0 \in \mathbb{R}^3$, of a query image with known intrinsics K_0 by comparing it to a K-tuple of database images with known intrinsics $\{K_j\}_{j=1}^{K}$ (note that K with subscript denotes intrinsics and without subscript denotes tuple length) and extrinsics $\{(R_j, t_j)\}_{j=1}^{K}$. The main input to ON3R is a set of matches between query keypoints $\{x_{i0}\}_{i=1}^{N}$ and database keypoints $\{x_{ij}\}_{i=1,j=1}^{N,K}$, with a mask $\{m_{ij}\}_{i=1,j=1}^{N,K}$ indicating match presence ($m_{ij} = 1$) or absence ($m_{ij} = 0$). This gives $M_j = \sum_{i=1}^{N} m_{ij}$ active keypoints per database image and $M = \sum_{j=1}^{K} M_j$ overall.

3.2. Neural Point Regression

The network $\mathcal{N}$ predicts a 3D point, X_i, in the world coordinate system given a 2D query keypoint, x_{i0}, yielding 2D-3D correspondences $\{(x_{i0}, X_i)\}_{i=1}^{N}$ used to estimate the absolute pose. At every network iteration, the 3D points are projected to each database images by,

$$x_{ij}^{\mathcal{N}} = \frac{X_{ij}}{d_{ij}^{\mathcal{N}}}, \quad \text{with} \quad X_{ij} = R_j X_i + t_j, \tag{1}$$

where $d_{ij}^{\mathcal{N}}$ is the depth (third element of X_{ij}). Then, we measure consistency with the matched keypoints,

$$r_{ij}^{c} = \left\| x_{ij}^{\mathcal{N}} - K_j^{-1}(x_{ij}, 1)^T \right\|_2 . \tag{2}$$

Residuals are retained only where $m_{ij} = 1$ (*i.e.*, a match exists), corresponding to applying the match mask m_{ij} to r_{ij}^{c}, yielding r_k^{c}, $k = 1, 2, \ldots, M$ ($M = \sum_{i=1}^{N} \sum_{j=1}^{K} m_{ij}$). Note that, similar to SCR, this does not require the keypoint to be seen in multiple map images. Importantly, the network is trained from scratch for each new query-database tuple.

When the query keypoints are input to ON3R, they are first normalized using the query image width and height to be in the range [0, 1]. Then, a fixed positional encoding $[\sin(2^f x), \cos(2^f x)]_{f=0}^{4}$ is applied. It is then concatenated with the normalized coordinates, x, and input to a 7-layer MLP that outputs a 3D world coordinate for each query

keypoint. Note that no visual features are used and that each keypoint is processed separately. See Appendix A for details on how we initialize the bias of the final layer and apply a similarity transform to the database poses, placing the initial predictions in a plausible region of the scene.

3.3. Monocular Depth as Prior

In cases where the database images have no overlap at all, the network cannot infer the point depth by simply minimizing reprojection residuals. To solve this issue for most cases, we regularize the depth in the reference views using the monocular depth estimator MoGe-2 [53]. MoGe-2 predicts metric scale depth for single images and sets a clear state-of-the-art on several benchmarks. Since our scenes are not necessarily in metric scale, we convert our predicted depths, $d_k^{\mathcal{N}}$, to metric scale by estimating a global scale factor γ. Let $d_k^{\mathcal{N}}, d_k^{\mathcal{M}}, k = 1, \dots, M$ denote depths from our network and MoGe-2, respectively. The scale factor γ minimizes

$$\sum_{k=1}^{M} w_k^2 \left(\gamma d_k^{\mathcal{N}} - d_k^{\mathcal{M}}\right)^2, w_k = \frac{s^2}{s^2 + \min(r_k^c, s_{\text{init}}^d)^2}, \quad (3)$$

aligning our depths with MoGe-2's metric scale. The weights w_k downweight outliers based on the reprojection residuals r_k^c, and depend on the robust scaling factor s used in our loss (see Eq. (6) and Appendix C). The starting scale factor for the depth loss is $s_{\text{init}}^d = \frac{500}{f}$, where f is the focal length. Setting the derivative of Eq. (3) to zero yields

$$\gamma = \sum_{k=1}^{M} w_k^2 d_k^{\mathcal{N}} d_k^{\mathcal{M}} \Big/ \sum_{k=1}^{M} w_k^2 d_k^{\mathcal{N}} d_k^{\mathcal{N}}. \quad (4)$$

We can then compute depth residuals in the same scale as image coordinates by

$$r_k^d = \left(\gamma d_k^{\mathcal{N}} - d_k^{\mathcal{M}}\right) / d_k^{\mathcal{M}}. \quad (5)$$

If $\gamma < 0$, residuals may be small despite opposite depth signs. For such cases, we omit supervising on r_k^d.

3.4. Supervision

The residuals r_k^c (Eq. (2)) and r_k^d (Eq. (5)) are used in a modified version of the robust Cauchy loss where

$$\rho(r) = s \ln\left(1 + \frac{r^2}{s^2}\right). \quad (6)$$

The difference between this and the standard Cauchy loss [1] is that we multiply the logarithmic term by s rather than by s^2. Why we do this, how we robustly scale s during training, and how the value of s determines the network stopping criterion is specified in Appendix C.

Finally, we also add a depth loss that is activated whenever $\gamma \leq 0$, penalizing 3D points appearing behind their corresponding database camera, forcing the 3D scene to be in front of the reference views. Added together, the full network loss can be formulated as

$$\mathcal{L} = \sum_{d_k^{\mathcal{N}} > 0} \rho(r_k^c) + \begin{cases} \lambda \sum_k \rho(r_k^d) & \text{if } \gamma > 0, \\ -\sum_k d_k^{\mathcal{N}} & \text{if } \gamma \leq 0, \end{cases} \quad (7)$$

with $\lambda = 0.1$. Before adding the reprojection term to the loss, we let the network train a few dozen epochs to obtain a crude depth estimate.

3.5. Robust Estimation and Bundle Adjustment

Once the network has converged, all 2D-3D correspondences regressed by the network are inserted into P3P-RANSAC [12, 17, 25], giving an initial query pose estimate. Then, the query pose and the 3D points are refined with bundle adjustment, where the database poses are fixed. Inside the bundle adjustment, the residuals are input to the robust Cauchy loss, and the objective is minimized with the Sparse Schur Complement trick implemented in Ceres [1].

4. Experiments

We first present the metrics used for the experiments. After that, we carefully go through all the baselines, as this type of extremely sparse setting has no standard baselines. We evaluate our method by subsampling existing localization datasets with dense map coverage. This choice is driven by the difficulty of obtaining accurate ground truth poses in genuinely sparse real-world scenes, where reconstruction pipelines often fail or require substantial manual effort. This follows established practice in prior work (see *e.g.* [20]), where images from dense view-graphs are selectively removed to simulate sparse coverage. We benchmark our method for tuple lengths 2, 3, and 4 on the datasets MegaDepth [29] and Cambridge Landmarks [23]. For both datasets, we test on 300 tuples for each tuple length. Next, we compare the localization performance on sparsified versions of the Cambridge Landmarks and Aachen Day-Night datasets between the state-of-the-art HLOC and ON3R. Simultaneously, we show that our network settings are insensitive to the dataset and the tuple length, as we use the same hyperparameter configuration for every experiment. If not mentioned otherwise, we use LightGlue [30] as a matching method and SuperPoint [10] as keypoint detector and descriptor. In Appendix B, we show that ON3R can also be applied with the dense matcher RoMa [16] on MegaDepth and ScanNet [9]. All experiments were run on an NVIDIA GeForce RTX 4090. More details on the network architecture and hyperparameters are given in Appendix D. To get a better sense of the type of sparse-view data we evaluate on and the 3D reconstruction that ON3R outputs, see Fig. 3.

	$K=2$		$K=3$		$K=4$	
	$(\varepsilon_R, \varepsilon_t)\downarrow$	$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow$	$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow$	$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)\uparrow$
Transitive Matching	1.34, 1.89	4.7, 18.0, 41.3, 62.7	0.28, 0.38	**10.0**, 37.3, 74.7, 89.7	0.18, 0.28	10.7, 45.3, 88.0, 96.7
Motion Averaging	1.40, 3.06	0.7, 3.7, 21.7, 63.0	0.66, 2.50	0.0, 4.7, 28.7, 68.0	0.74, 4.06	0.0, 3.7, 20.3, 55.3
VGGT	1.11, 3.67	0.0, 2.3, 14.3, 58.0	0.80, 2.95	0.0, 0.7, 19.7, 63.7	0.78, 2.65	0.0, 2.3, 23.3, 65.7
Reloc3r	0.90, 2.83	0.0, 2.0, 16.3, 64.7	0.61, 2.14	0.0, 1.0, 26.0, 71.7	0.61, 2.78	0.7, 1.3, 25.0, 68.0
ACE	27.62, 4.35	0.0, 0.3, 1.3, 11.3	28.91, 3.65	0.0, 0.0, 1.3, 8.0	28.03, 3.94	0.0, 0.0, 2.7, 11.7
ON3R	**0.37**, **0.45**	**7.7**, **27.3**, **71.0**, **85.0**	**0.21**, **0.30**	9.3, **42.3**, **85.3**, **94.0**	**0.15**, **0.21**	**13.3**, **55.3**, **94.0**, **97.7**

Table 1. **Results on MegaDepth for** $K = 2, 3, 4$. Metrics are median rotation/translation errors $(\varepsilon_R, \varepsilon_t)$ and recall $(\varepsilon_1$–$\varepsilon_4)$.

4.1. Metrics

For all experiments, we report a subset of the following metrics: the median rotation error in degrees, the median translation error in meters, and recall measured at thresholds $\varepsilon_1 = (1°, 0.1\text{m}), \varepsilon_2 = (2°, 0.25\text{m}), \tilde{\varepsilon}_3 = (5°, 0.5\text{m}), \varepsilon_3 = (5°, 1\text{m}), \varepsilon_4 = (10°, 5\text{m})$. Recall is defined as the percent of test queries for which both the rotation and translation errors fall below the specified thresholds. The rotation error and translation errors are calculated as

$$(\varepsilon_R, \varepsilon_t) = \left(\arccos \frac{\text{Tr}(\hat{R}^T R)-1}{2}, \|R^T t - \hat{R}^T \hat{t}\|_2\right). \quad (8)$$

For simplicity, the median rotation and translation error are denoted as ε_R and ε_t in the tables. We use **bold** and underline to highlight the best and the second best method.

4.2. Baselines

We have five different baselines for our main experiments. (1): *Transitive matching* (as explained in Sec. 2.3) with a second pass and bundle adjustment after both passes. (2): *Matching+motion averaging* first estimates a relative pose $P_{j\to 0}$ between each query-database pair. Then, Reloc3r's [14] motion averaging module is used to convert the K relative pose estimates to one absolute pose estimate. For the rotation part, we obtain K absolute query rotation estimates $R_0^{(j)} = R_{j\to 0} R_j, j = 1, \ldots, K$ in world-to-camera form. After that, the absolute rotation is chosen as the median rotation in the quaternion representation. For the translation part, the sum of squared distances from the estimated camera center to the translation rays, coming from the relative pose estimates, is minimized using a least squares solver. For both *transitive matching* and *motion averaging*, the same input matches as for ON3R are used.

(3): *VGGT* [52] is run by estimating the poses of all database images in the query's coordinate frame. After that, all query-database relative pose estimates are converted to the world coordinate frame using the motion averaging module in Reloc3r. (4): *Reloc3r* [14] relative pose estimates are converted to a query absolute pose as above.

(5): the SCR method *ACE* [3] is the last baseline. For smaller scenes like ours with only a few images, feature extraction is nearly instantaneous, but head training still requires several seconds to tens of seconds. Across all tested configurations, the results were both slow and inaccurate, so we report only one of the faster variants to enable testing across multiple datasets. Notably, extending head training yielded no significant accuracy improvements.

4.3. MegaDepth

MegaDepth [29] is a large outdoor dataset comprising many scenes. As ON3R is trained online, only the two test scenes are used. Star-topology data can be sampled using the covisibility data coming with the dataset. The star-topology sampling algorithm first searches for difficult query–database tuples. If too few are found, the covisibility threshold is relaxed and sampling is repeated (see code for details). Because MegaDepth scenes are not in metric scale, we have manually estimated the metric scale factor using Google Maps to get more realistic translation errors in the evaluation.

As seen in Tab. 1, our method surpasses the baseline methods with a large margin, with the difference being most prominent for only two database images ($K = 2$). Transitive matching comes closest, though it still clearly lags behind. The accuracy of pose regression methods is simply too low, and ACE has a hard time inferring depth with only single-view constraints when training on so few images.

4.4. Cambridge Landmarks

Cambridge Landmarks [23] is a dataset covering five outdoor sites around the Cambridge campus. As we want our test set to consist of star-topology data, we need to use covisibility information. We therefore construct it from COLMAP [37, 43–45] SfM data and define covisibility as $\frac{|S_1 \cap S_2|}{2}\left(\frac{1}{|S_1|} + \frac{1}{|S_2|}\right)$ where S_1 and S_2 are the sets of 3D points for two images.

In Tab. 2, we report the experimental results. As on MegaDepth, we achieve state-of-the-art performance. Interestingly, our method and transitive matching estimate translation considerably more accurately than other methods, whereas VGGT is competitive in terms of estimating the orientation. Across all tuple lengths, ON3R achieves the highest or tied highest overall recall, confirming its ro-

	$K=2$		$K=3$		$K=4$	
	$(\varepsilon_R, \varepsilon_t)\downarrow$	$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow$	$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow$	$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)\uparrow$
Transitive Matching	2.49, 1.02	5.7, 23.7, 49.7, 61.0	0.71, **0.31**	**12.3**, **41.0**, 78.0, 90.0	**0.39**, **0.18**	**25.3**, 58.0, **91.7**, **99.0**
Motion Averaging	1.17, 1.69	1.7, 6.0, 38.0, 74.3	0.98, 1.02	2.3, 10.7, 49.3, 83.3	0.85, 0.94	4.3, 17.7, 51.7, 87.0
VGGT	**0.63**, 1.52	1.3, 5.7, 35.3, 77.3	**0.61**, 0.90	2.0, 10.0, 53.7, 90.0	0.45, 0.66	4.0, 20.7, 61.7, 93.3
Reloc3r	0.88, 1.72	1.0, 6.3, 30.3, 74.0	0.83, 1.40	0.0, 6.3, 41.3, 83.0	0.64, 0.91	2.0, 13.7, 53.3, 88.3
ACE	18.36, 8.46	0.0, 0.0, 2.0, 13.0	12.12, 6.19	0.0, 0.0, 5.0, 29.7	8.39, 3.67	0.0, 1.3, 12.3, 48.3
ON3R	0.91, **0.51**	**8.7**, **28.3**, **68.7**, **87.3**	0.66, 0.32	12.0, **41.0**, **79.0**, **95.3**	0.40, **0.18**	23.0, **60.0**, 90.0, **99.0**

Table 2. **Results on Cambridge Landmarks for** $K = 2, 3, 4$. Metrics are median rotation/translation errors $(\varepsilon_R, \varepsilon_t)$ and recall $(\varepsilon_1$–$\varepsilon_4)$.

	GreatCourt	**KingsCollege**	**OldHospital**	**ShopFacade**	**StMarysChurch**
	$(\varepsilon_R, \varepsilon_t)\downarrow \mid (\varepsilon_2, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow \mid (\varepsilon_2, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow \mid (\varepsilon_2, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow \mid (\varepsilon_2, \varepsilon_4)\uparrow$	$(\varepsilon_R, \varepsilon_t)\downarrow \mid (\varepsilon_2, \varepsilon_4)\uparrow$
HLOC, $K=5$	30.80, 27.91 | 0.4, **41.3**	0.66, 0.43 | 22.5, **86.3**	0.62, 0.40 | 23.6, **94.5**	64.98, 16.66 | 23.3, 42.7	53.50, 24.03 | 0.0, 0.0
ON3R, $K=5$	**23.73**, **5.23** | **4.1**, 32.3	**0.57**, **0.37** | **28.0**, 85.1	**0.61**, **0.34** | **33.5**, 93.4	**5.75**, **2.47** | **25.2**, **51.5**	**4.88**, **13.35** | **11.1**, **35.5**
HLOC, $K=10$	2.69, 3.92 | 3.4, 54.7	0.48, 0.34 | 33.5, 91.3	**0.44**, **0.29** | 37.9, 98.4	2.01, 0.43 | 38.8, 61.2	6.93, 16.83 | 13.0, 39.3
ON3R, $K=10$	**0.85**, **1.41** | **10.4**, **64.1**	**0.35**, **0.28** | **43.7**, **99.7**	0.53, 0.30 | **44.5**, **98.9**	**1.54**, **0.34** | **40.8**, **69.9**	**0.91**, **0.35** | **38.3**, **70.6**
HLOC full	**0.11**, **0.18** | **62.4**, **99.9**	**0.21**, **0.11** | 74.6, **100.0**	**0.30**, **0.14** | **67.6**, **100.0**	**0.20**, **0.04** | 95.2, **100.0**	**0.22**, 0.08 | **96.4**, **99.8**
ON3R full	0.12, 0.22 | 55.9, 99.1	0.22, 0.12 | **77.0**, **100.0**	0.35, 0.16 | 62.1, 98.9	**0.20**, 0.05 | **96.1**, **100.0**	0.24, **0.07** | 96.0, **99.8**

Table 3. **Full retrieval on subsampled Cambridge Landmarks scenes**. Each scene is subsampled so that only K database images remain, from which both HLOC and ON3R retrieve all. For reference, we also show the performance of HLOC and ON3R when retrieving $K=10$ images (the HLOC default) from the full database. In this non-sparse setting, HLOC performs slightly better.

bustness under extremely sparse conditions.

4.5. Comparison to Structure-based Localization

To compare ON3R against state-of-the-art structure-based approaches, we compare with HLOC [37] on Cambridge Landmarks [23] and Aachen Day-Night v1.1 [40, 42, 59].

4.5.1. Cambridge Landmarks

We test on two versions of the Cambridge Landmarks dataset: one with 5 and one with 10 images retained per scene. Both methods retrieve all available database images during localization. The results in Tab. 3 show that our approach outperforms HLOC, being worse on only 5 metrics, and better on the remaining 35 metrics. HLOC performs competitively only on OldHospital, while our method achieves superior results on all other scenes.

4.5.2. Aachen Day-Night

For Aachen, we create a 99% sparsified version of the outdoor dataset where only 67 of 6697 images remain. This makes absolute pose estimation difficult: database images may not be covisible, rendering triangulation infeasible and depths inestimable from query-database correspondences. Another challenge is maintaining robustness to outliers among the K retrieved images. HLOC handles this well, as false image pairs typically yield few correspondences that RANSAC [17] can reject.

In our pipeline, it is crucial that the regressed 3D scene remains within the query camera frustum. As the network is a relatively smooth function over function input (query image coordinates), residuals from outliers must be down-weighted to avoid erroneous reconstructions. Our problem addresses this with the robust Cauchy loss, though the setup continues to be inherently difficult. Initializing the scene origin is also challenging, as database cameras may lie in widely separated parts of the map.

	Day	**Night**
	$(\varepsilon_2, \tilde{\varepsilon}_3, \varepsilon_4)\uparrow$	$(\varepsilon_2, \tilde{\varepsilon}_3, \varepsilon_4)\uparrow$
HLOC sparse	**3.8**, 6.4, 12.5	2.6, **5.8**, 9.9
ON3R sparse	3.5, **7.5**, **22.7**	**3.1**, 5.2, **15.7**
HLOC full	**89.1**, **96.1**, **99.3**	**74.3**, **89.0**, **99.0**
ON3R full	67.1, 86.2, 95.1	58.6, 79.1, 90.1

Table 4. **Sparsified Aachen**. Comparison between HLOC and ON3R on Aachen Day-Night v1.1 under 99% database sparsity, using $K=3$ retrieval. For reference, we also report results for HLOC and ON3R on the full database with $K=50$. Note that ON3R is not designed for such a large retrieval setting, although it still functions correctly. ON3R is inherently more sensitive to faulty retrieval, as all residuals and database poses influence both the learning process and the initialization of the 3D point cloud.

Results from this experiment, evaluated on *visuallocalization.net*, are reported in Tab. 4. At lower thresholds, performance is largely tied between HLOC and ON3R. However, at the upper threshold $\varepsilon_4 = (10°, 5\text{m})$, ON3R has 81.6% and 58.6% higher recall at day and night, respectively. One reason for this could be that ON3R, unlike HLOC, does not rely on covisibility between database images, enabling localization for star-topology structured data.

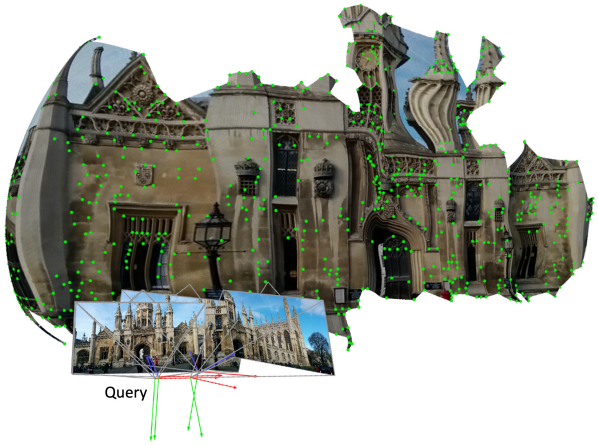

Figure 4. **Network smoothness on KingsCollege**. A dense reconstruction of KingsCollege from Cambridge Landmarks learned by our network. The green points are the only ones used during training, whereas the rest of the pixels (the concave hull of the query keypoints) lifted to 3D space are *given for free* following the network smoothness. The 3D points here have not been refined with bundle adjustment, so in practice, the locations of the query keypoints are even more precise.

4.6. Ablation Study

In Tab. 5, we modify individual components of our method to analyze their impact. Evidently, bundle adjustment is highly important for accuracy while adding merely 20 milliseconds of computation time. The monocular depth prior is also a crucial component, even though it accounts for 40% of the total processing time. For this particular dataset, the maximum number of network epochs can be reduced by a factor of 10 without a substantial drop in performance. Interestingly, removing MoGe-2 and limiting training to a maximum of 50 epochs provides a fast alternative when speed is prioritized. Lastly, we tested the DISK [49] descriptor and observed that it yields higher accuracy than SuperPoint [10] but at a higher computational cost.

	time [s]↓	$\varepsilon_R, \varepsilon_t \downarrow$	$\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \uparrow$
ON3R	0.55	0.37, 0.45	7.7, 27.3, 71.0, 85.0
w/o BA	0.53	4.11, 9.31	0.0, 0.3, 7.0, 33.0
w/o MoGe-2	0.33	0.55, 0.84	6.7, 21.0, 54.3, 65.3
50 epochs	0.41	0.39, 0.47	7.7, 25.3, 66.0, 79.3
50 ep. w/o MoGe-2	**0.26**	0.58, 0.86	6.0, 19.3, 52.7, 64.3
DISK [49]	0.62	**0.31, 0.41**	**8.0, 31.7, 71.3, 87.7**

Table 5. **Ablation study** for the $K = 2$ case on MegaDepth. By default, ON3R uses 500 maximum epochs and SuperPoint [10].

4.7. Additional Results

Timings for all methods on Megadepth are presented in Tab. 6. The timings of ON3R include SuperPoint keypoint extraction and description, LightGlue matching, and all parts of our module (*e.g.*, monodepth estimation, training from scratch, pose estimation, and bundle adjustment). As seen, our method is the second slowest, mainly due to the online training. However, the overall runtimes remain of the same order of magnitude (disregarding ACE). We believe that ON3R could be made significantly faster, at least for some datasets, without noticeable performance drops (see Tab. 5). Nevertheless, if speed is the primary concern, other methods may be more suitable.

Finally, we demonstrate in Fig. 4 that our network is smooth over spatial query coordinates. As seen, even if the network only regresses a sparse set of 3D keypoints, it can learn good 3D locations for the other pixels as well. As expected, the network is best in regions with high keypoint density, whereas on the pillars where almost no keypoints exist, the network is less accurate. For more examples like this, see Appendix E.1. For a visualization of how the 3D scene output by ON3R gradually improves over training epochs, see Appendix E.2.

	$K = 2$ [s] ↓	$K = 3$ [s] ↓	$K = 4$ [s] ↓
Transitive Matching	0.13	0.21	0.27
Motion Averaging	**0.08**	**0.11**	**0.15**
VGGT	0.17	0.23	0.26
Reloc3r	0.23	0.33	0.43
ACE	∼3	∼3	∼3
ON3R	0.55	0.63	0.73

Table 6. **Timings** on MegaDepth for different tuple sizes K.

5. Conclusion

This work introduces an intuitive route to absolute pose estimation when facing sparse, low-overlap database images. Namely, regressing 2D-3D correspondences supervised by database reprojection residuals and a monocular-depth prior, followed by P3P-RANSAC and lightweight bundle adjustment. The approach works without a prebuilt 3D map and delivers improvements over existing methods when only a few low-overlapping database images are available. ON3R remains sensitive to retrieval outliers and bias initialization errors, and is limited by the computational cost of online training. Promising directions include initializing from a pretrained encoder to reduce training time, using extra samples from the query image to regularize the network, analyzing and shaping the loss landscape to mitigate local minima, and tightening robustness under poor retrievals.

Acknowledgments

This work was supported by the strategic research environment ELLIIT funded by the Swedish government, and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 2023. 5

[2] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2021. 3

[3] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 6

[4] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *European Conference on Computer Vision (ECCV)*, 2024. 3

[5] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 4

[6] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *International Conference on Computer Vision (ICCV)*, 2013. 4

[7] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 40(4):958–972, 2017. 4

[8] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 3, 4

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 2

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 3, 5, 8

[11] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera relocalization. In *International Conference on Computer Vision (ICCV)*, 2019. 3

[12] Yaqing Ding, Jian Yang, Viktor Larsson, Carl Olsson, and Kalle Åström. Revisiting the p3p problem. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 5

[13] Siyan Dong, Shaohui Liu, Hengkai Guo, Baoquan Chen, and Marc Pollefeys. Lazy visual localization via motion averaging. *arXiv preprint arXiv:2307.09981*, 2023. 1, 4

[14] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 3, 4, 6

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 4

[16] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *Computer Vision and Pattern Recognition (CVPR)*, 2024. 5, 1, 2

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 3, 5, 7

[18] Martin Humenberger, Yohann Cabon, Noé Pion, Philippe Weinzaepfel, Donghwan Lee, Nicolas Guérin, Torsten Sattler, and Gabriela Csurka. Investigating the role of image retrieval for visual localization: An exhaustive benchmark. *International Journal of Computer Vision (IJCV)*, 130(7): 1811–1836, 2022. 4

[19] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 4

[20] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision (IJCV)*, 129(2):517–547, 2021. 5

[21] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feedforward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026. 4

[22] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[23] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision (ICCV)*, 2015. 3, 4, 5, 6, 7

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 3

[25] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 3, 5

[26] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *International Conference on Computer Vision Workshops (ICCVW)*, 2017. 3, 4

[27] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision (ECCV)*, 2024. 4

[28] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[29] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 6

[30] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 5

[31] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *International Conference on Computer Vision (ICCV)*, 2013. 4

[32] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[33] Vojtech Panek, Qunjie Zhou, Yaqing Ding, Sérgio Agostinho, Zuzana Kukelova, Torsten Sattler, and Laura Leal-Taixé. A guide to structureless visual localization. *arXiv preprint arXiv:2504.17636*, 2025. 1, 4

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Neural Information Processing Systems (NeurIPS)*, 2019. 4

[35] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[36] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking image retrieval for visual localization. In *International Conference on 3D Vision (3DV)*, 2020. 4

[37] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 6, 7

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[39] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[40] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *British Machine Vision Conference (BMVC)*, 2012. 7

[41] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[42] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[43] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 6

[44] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[45] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 6

[46] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 3

[47] Danny Stoll, Neeratyoy Mallik, Simon Schrodi, Eddie Bergmann, Maciej Janowski, Samir Garibov, Tarek Abou Chakra, Daniel Rogalla, Eddie Bergman, Carl Hvarfner, Ru Binxin, and Frank Hutter. Neural Pipeline Search (NePS), 2024. 4

[48] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 4

[49] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Neural Information Processing Systems (NeurIPS)*, 2020. 3, 8

[50] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[51] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *International Conference on 3D Vision (3DV)*, 2025. 4

[52] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt:

Visual geometry grounded transformer. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 4, 6

[53] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. *arXiv preprint arXiv:2507.02546*, 2025. 2, 5

[54] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 4

[55] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Neural Information Processing Systems (NeurIPS)*, 2022.

[56] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *International Conference on Computer Vision (ICCV)*, 2023. 4

[57] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 4

[58] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *Third international symposium on 3D data processing, visualization, and transmission (3DPVT)*, 2006. 4

[59] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision (IJCV)*, 129(4):821–844, 2021. 7