The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement

Gabriele Trivigno[1] Carlo Masone[1] Barbara Caputo[1] Torsten Sattler[2]
[1] Politecnico di Torino
[2] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
{gabriele.trivigno,carlo.masone,barbara.caputo}@polito.it torsten.sattler@cvut.cz

Abstract

Pose refinement is an interesting and practically relevant research direction. Pose refinement can be used to (1) obtain a more accurate pose estimate from an initial prior (e.g., from retrieval), (2) as pre-processing, i.e., to provide a better starting point to a more expensive pose estimator, (3) as post-processing of a more accurate localizer. Existing approaches focus on learning features / scene representations for the pose refinement task. This involves training an implicit scene representation or learning features while optimizing a camera pose-based loss. A natural question is whether training specific features / representations is truly necessary or whether similar results can be already achieved with more generic features. In this work, we present a simple approach that combines pre-trained features with a particle filter and a renderable representation of the scene. Despite its simplicity, it achieves state-of-the-art results, demonstrating that one can easily build a pose refiner without the need for specific training. The code is at https://github.com/ga1i13o/mcloc_poseref

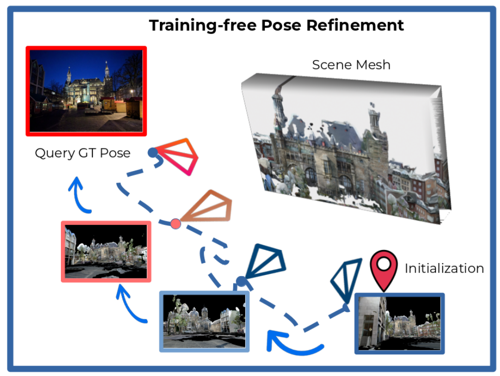

Figure 1. **MCLoc** localizes images with a *render&compare* strategy. Given a starting hypothesis, a particle filter is used to perturb it and sample new candidates, which are rendered, and compared to the query using generic pre-trained features.

1. Introduction

Visual localization estimates the position and the rotation of a camera in a given scene. It is essential in a wide range of applications such a Simultaneous Localization and Mapping (SLAM) [5, 30], Structure-from-Motion (SfM) [97, 98], autonomous navigation [25, 77], robotics [35, 36], and Augmented-Virtual Reality (AR/VR) [41, 86].

State-of-the-art methods follow a structure-based approach [89, 90] where a 3D map of the scene is available and a query image is localized against it by deriving 2D-3D matches. Such 2D-3D correspondences are obtained by matching local features [26, 31, 88] between the query image and the 3D points in the map, typically an SfM [97, 98] sparse point cloud. These matches are used to estimate the camera pose with minimal solvers [42, 82] integrated with robust optimization [24, 34]. Generating the point cloud via SfM involves local feature detection and description on a set of reference images, feature matching, and triangulation of image points that are co-visible in several images [42, 97]. The resulting 3D points are then associated with visual descriptors from the reference images.

While SfM-based point clouds enable robust and accurate localization [90, 91, 96, 105], they remain *unflexible* as they are tied to the specific features used for the reconstruction and their use is limited to the localization task [10, 78, 80]. A feature-agnostic alternative to the previous map representations are meshes [80, 81, 112], as they support different tasks in the ecosystem of pose estimation, such as SLAM [10, 78, 104, 120], tracking [58], path planning [45], and relocalization [4, 112] while providing the 3D information necessary for visual localization. Such models are easily obtained [15, 50, 74], and are rendered rather efficiently (*e.g.*, 1 ms or less) even for large, textured models [80], relying on mature graphics primitives.

A different approach to visual localization is to refine an initial pose estimate. This strategy can either be applied to refine a pose estimate obtained from 2D-3D matches, or to obtain a more accurate pose starting from an initial hypothesis provided by image retrieval [47, 89]. As such, these methods are to a large degree complementary to the matching-based methods described above. These approaches typically follow a *render&compare* framework [58, 64]: in each iteration, a rendering (either an image [58, 115, 117] or the projection of a sparse set of features [40, 92, 114]) of the scene obtained from the current pose estimate is compared to the actual image. Based on this comparison, an update is computed for the pose in order to better align the query with the scene representation. Existing approaches learn specific features for this task [20, 40, 73, 92], potentially optimized together with the scene representation [19, 67, 72].

We argue that in a *render&compare* framework, the main requirement is being able to evaluate the visual similarity of a synthetic view versus a real image. It has been shown repeatedly that generic deep features are a reliable estimator of this measure [39, 54, 116], and that this property of dense features makes them suitable to re-rank poses [106]. This is in contrast with the aforementioned refinement approaches, which rely on sparse features that need to be optimized for the task, and it leads us to the research question of whether it is truly essential to train specialized features for localization, or if analogous results can be attained exploiting the properties of dense features from generally available, off-the-shelf architectures.

Opting out of feature optimization also removes the need to articulate a differentiable feature-to-pose pipeline, required to compute gradients. Instead, to refine the pose, we adopt a simple particle filter-based optimizer [57, 109] that efficiently explores the hypothesis space [22]. Despite its simplicity, our **MCLoc** approach outperforms modern pose regressors [19, 72] and is comparable or better than refinement pipelines based on implicit fields [20, 40, 73], even though both these families of methods are optimized per-scene. Unlike them, our method also scales to large scenes. While matching-based methods still hold the state-of-the-art, our method brings complementary strengths, in that it can be used to improve the performance of matching-based approaches as a post- or pre-processing step. We demonstrate these strengths through extensive experiments, both indoor and outdoor, as well as large scale scenarios, providing evidence that it is possible to construct a pose refiner that generalizes across different domains and representations, without the need for specialized training.

Contributions:

- A simple yet powerful particle-filter based optimization which can be applied to different scene representations and scoring functions
- We provide an analysis on the effectiveness of general, pre-trained features at different layers of deep networks as a robust cost function
- We show a versatile pose-refinement approach which does not entail per-scene training or fine-tuning, that can be used either standalone, or to obtain a better pose prior, or to refine previous pose estimates
- The code, which allows to experiment with different backbones, scoring functions and scene representations, is available at https://github.com/ga1i13o/mcloc_poseref

2. Related works

Visual Localization. Visual Localization aims at estimating the camera pose of a given query within a known environment. A popular strategy is to rely on sparse 3D models, obtained from SfM [97], to represent the scene. These point clouds associate to each 3D location features triangulated from the available database. For inference, local features are used to find matches between a query and the 3D model [15, 63, 90, 91, 94, 95, 99, 106, 107]. Once 2D-3D matches are obtained, the query pose can be estimated via a PnP solver [59]. To avoid matching against the entire database, it is common to apply a hierarchical approach [46, 47, 89], where a network for Place Recognition [9, 84] selects database images with potentially covisible regions [3, 8, 111]. There also exist methods that replace the SfM model in favor of more versatile dense representation, either point clouds from Multi-View stereo or LIDAR [98, 101, 106] a mesh [15, 80, 117], or NeRFs [67, 71, 115]. In this work we show that by relying on a renderable representation of the scene, it is possible to align the query pose by comparing features pixelwise, without resorting to exhaustive feature matching. While matching-based methods retain state-of-the-art performances, our method has complementary strengths: it can be effortlessly applied either to refine initial poses or final estimates, improving matching results while adding little computational overhead.

Implicit representations for Visual Localization. Both sparse and dense models store explicit information about the geometry of a scene. On the other hand, a parallel line of research has focused on implicit representations, that embed information about the scene into the weights of a neural network. Traditionally this was done either by training models to regress the camera pose [27, 51, 72, 102], or with Scene Coordinate Regressors, which encode for each image patch the corresponding 3D points[11, 13, 17, 18]. More recently, neural radiance fields gained popularity [7, 71, 76]. These methods map each point of the scene to a view-dependent color and density values, through a MLP-based network. These representations can be exploited for localization by embedding features in the implicit representation [40, 67, 73] or by inverting the neural field [66, 115]. Implicit representations have also been used as data augmentation to generate samples to train pose regressors [19, 72].

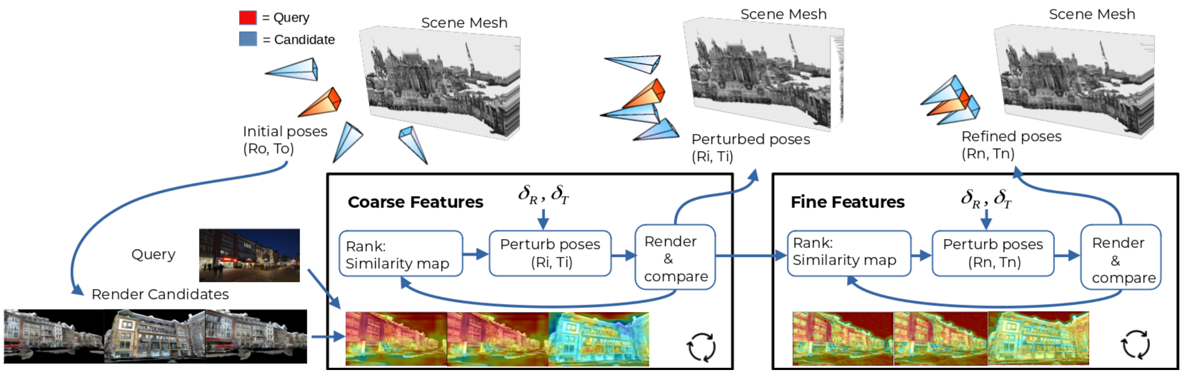

Figure 2. **Architecture of MCLoc**. It exemplifies our iterative pose refinement. Given an initial pose estimate, we perturb it and render new candidates. Candidates are ranked based on dense, pixelwise feature similarity. As optimization progresses, we exploit the hierarchical properties of deep features by switching to shallower features, which are better for fine-grained comparison.

Pose refinement and image alignment. Pose refinement is a relevant area of research, in which the main idea is to iteratively refine the pose estimate by minimizing an objective function. In this family, a longstanding approach is represented by Direct Alignment methods, which minimize differences in pixel intensities when projecting the scene into the current estimate [6, 33], using gradient-based optimizers such as Levenberg-Marquardt [61, 70] or Gauss-Newton methods. These approaches are popular in SLAM scenarios [1, 100], and typically rely on photometric error, hardly robust to appearance variations. They have been applied on learned features as well [92, 113, 114]. Indirect methods define geometric correspondences in order to minimize the reprojection error [83]. Among direct methods, a notable example is PixLoc [92], which trains features end-to-end from pixels to camera pose. Inference is performed via feature-metric alignment relying on the SfM model. Lately, pose refinement methods based on implicit representations have gained popularity. [115] learns a radiance field that is used to render candidates, for which a photometric errors is computed and backpropagated. Alternatively, implicit representations can be used to model a feature field, rather than appearance. Within these methods, FQN [40] relies on reprojection error, whereas [20, 73] match the rendered features and then invert the descriptor field by backpropagating errors. These approaches require to train features per-scene, and are only applicable for small scenes, given the limited scalability of implicit models. Our method relies on general, pre-trained features, which work on any dataset. Moreover, being agnostic to the scene representation, it can scale to arbitrarily large scenes where a mesh can be easily obtained [80]. Our findings also relate to [116], that showed how deep learning models are surprisingly good at evaluating image similarities, outperforming all "handcrafted" metrics. We extend this analysis beyond perceptual similarity and show how generic features can discern among fine-grained pose discrepancies.

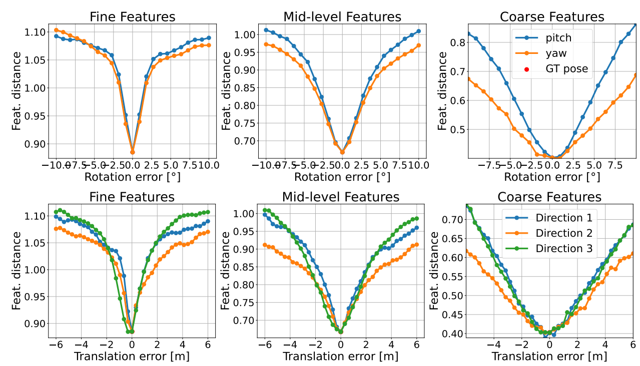

Figure 3. **Convergence Basin in Optimization Space at Multiple Scales**. We perturb rotation and translation for a query from Aachen and compute the dense, pixelwise feature distance at different depths. First row: rotating along yaw and pitch axis. Second row: moving away from the GT along 3 random directions.

Localization with Particle filters. Our work is not the first to employ a particle filter for localization [85]. A similar optimization technique to ours is adopted in [66, 69], although both these methods require a radiance field of the scene, thus sharing the limitations listed above, and rely on the less-than-robust pixel error.
Such approaches are popular also in mobile robotics, where they have been used for localization[37, 48, 49] and visual-tracking [23]. They have also been used in remote-sensing to localize against satellite images [44]. Theoretical properties of particle filters have been studied in [16, 22, 56, 57].

3. MCLoc

Overview. MCLoc localizes a query image within a *render-and-compare* framework, powered by Monte Carlo simulation. Localization is performed through iterative pose-refinement, as exemplified in Fig. 2. Given a query, and an initial hypothesis of its pose (which can be obtained in

several different ways), we perturb it with some noise and rely on a renderable representation of the scene to generate the corresponding views. A simple, generic feature extractor is used as a cost function to evaluate which candidates are more similar to the query. The pose estimates are modeled and perturbed with a particle filter [109], which serves as a stochastic optimizer.
Our method is agnostic to the scene representation used to render candidates, and we show that a general purpose feature extractor is suited to evaluate pose alignment, with no need for fine-tuning or per-scene training.

Motivation. This work aims at answering the following research question: *do we really need to train specialized descriptors or can generic features be used for localization?*. This question is rooted in the observation that activations of deep network are extremely reliable estimators of *perceptual similarity* [39, 116], being also robust to domain changes, blur and distortion [54]. Intuitively, perceptual similarity seems a promising metric for measuring pose similarity via a *render & compare* approach. We show that this property, coupled with the natural spatial structure of feature maps, yields a simple and effective tool to measure pose discrepancies using perceptual similarity as a way to measure pose similarity. To this end, we integrate a perceptual metric into a particle-filter-based optimizer [23, 109], that is used to generate new pose hypothesis to be then rendered & compared.

Problem setting. Our objective is, for a given query image I_q, to estimate its 6-DoF pose. Following [47, 52], we parametrize the pose as $T_q = (\mathbf{c}, \mathbf{q})$, where $\mathbf{c} \in \mathbb{R}^3$ represents the camera center and $q \in \mathbb{R}^4$ is a unit quaternion. Quaternion-based parametrizations provide a framework for manipulating rotations which is numerically stable, compact and avoids gimbal lock [60]. This formulation decouples translation and rotations updates, lying on the manifold of $\mathrm{SO}(3) \times \mathrm{T}(3)$ [16, 66].
We cast the problem as the following optimization:

$$\hat{T}_q = \underset{T \in \mathrm{SO}(3) \times \mathrm{T}(3)}{\text{argmin}} \mathcal{L}_{\mathcal{F}_\theta}(T | I_q, I_T) \tag{1}$$

where I_q, I_T are the query image and the rendered candidate with pose T, $\mathcal{F}_\theta$ is a feature extractor, and the loss function is the distance between query and rendered candidate in feature space, *i.e.*, $\mathcal{L}_{\mathcal{F}_\theta}(T|I_q, I_T) = ||\mathcal{F}_\theta(I_q) - \mathcal{F}_\theta(I_T)||_2$. We optimize this loss via a particle filter-based approach.

3.1. Pose alignment with Pre-trained features

To evaluate the loss in Eq. (1) associated with a candidate pose w.r.t. the query, we forward both through an off-the-shelf CNN. More details on the specific architecture will be discussed later on. We obtain a hierarchy of feature volumes, $F_l \in \mathbb{R}^{C_l \times H_l \times W_l}$, for each level $l \in \{1..L\}$. These feature pyramids have decreasing resolution, and encode increasingly richer semantic clues as the receptive field of each neuron grows. It has been demonstrated as an *emergent property* [38, 116] that such hierarchies of features can measure perceptual similarities at different conceptual levels [2]; to the best of our knowledge no previous works leverage this property of dense feature maps to assess *pose similarity* in a pose refinement algorithm.
We employ a simple scoring function that exploits this property; at a given step s of our optimization, we choose level $l(s)$ to compute the score of a candidate I_T against query I_q as follows:

$$\begin{aligned} S(h, w|l) &= \left\| \frac{F_l^{h,w}(I_q)}{||F_l^{h,w}(I_q)||_2} - \frac{F_l^{h,w}(I_T)}{||F_l^{h,w}(I_T)||_2} \right\|^2 \\ \mathcal{L}_{\mathcal{F}_\theta}(T|I_q, I_T, l) &= \frac{1}{h_l w_l} \sum_{h,w} S(h, w|l) \end{aligned} \tag{2}$$

where $F_l^{h,w} \in \mathbb{R}^{C_l}$. In practice, we compare pixelwise dense, normalized descriptors, obtaining a spatial similarity map $S \in \mathbb{R}^{H_l \times W_l}$, which is then averaged.
In the early stages of the optimization, we need to deal with large baselines as initial hypothesis might be far off from the ground truth. Thus, to increase the convergence basin, we adopt a hierarchical *Coarse-to-Fine* approach. Initially we rely on deeper features: their receptive fields are larger, hence even if the poses deviate significantly, there is a higher chance for two receptive fields of a pixel position to overlap, providing a meaningful signal. Moreover, since their features are semantically richer, they are more prone to ignore low-level details, transient objects and artifacts introduced by the scene representation. This allows to handle misalignment to a certain degree. As the optimization converges towards more accurate poses, the focus becomes discerning fine-grained details and small orientation displacements. At this stage, shallower features are more suited, as we can exploit their smaller receptive fields and higher spatial resolution. We discover that pre-trained features, across different architectures and training methods, are *unreasonably effective*, as deemed in [116], meaning that they present a nicely shaped convex basin around each pose. We experimentally demonstrate this finding in Fig. 3, which also illustrates how moving up the hierarchy we can control the width of the basin, and that shallower features are able to precisely discern among even the finer differences.

3.2. Particle filter optimization

Overview. Particle filters are a set of Monte Carlo methods that estimate the state of a system based on observations and dynamics of the system [57, 109]. Such algorithms can approximate a wide range of distributions, and are computationally efficient since they focus on regions of the state space with high likelihood [37]. The application of particle

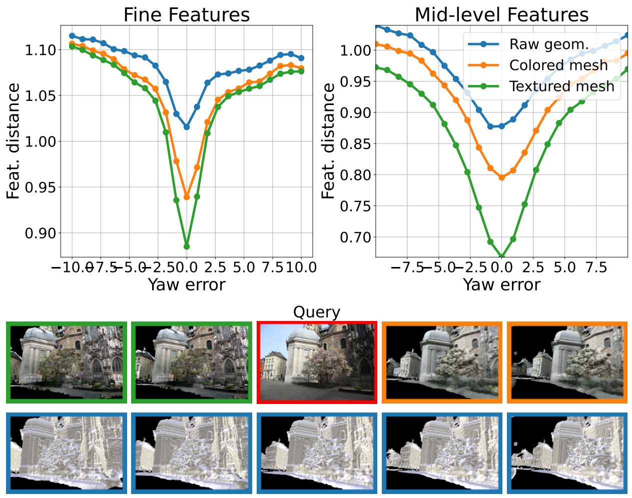

Figure 4. **Robustness of the Convergence Basin to the Rendering Domain**. We render images rotating along the yaw axis, using different meshes:Textured, Colored and Raw Geometry, and evaluate the feature distance at different depths. The domain shift affects absolute values but not the basin shape.

filters for visual localization is not novel, and their effectiveness has been demonstrated in [49, 66, 69, 85].
The basic idea is that, given a starting hypothesis, by perturbing the initial state we can obtain new state hypothesis, evaluate the cost function and refine the estimate iteratively. In our case, the state variable is the camera pose T and the scoring function to evaluate an hypothesis is the *perceptual similarity*. Specifically, at each step the particle filter models the posterior distribution $p(\mathbf{T}_q|\mathbf{Z}_i)$ of the query pose $\mathbf{T}_q$, with a set of particles $\mathbf{Z}_i = \{(T_i^1, \pi_i^1), ..(T_i^n, \pi_i^n)\}$. Particles have a weight π_i^n that represents their likelihood, estimated via Eq. (1). Since the particles states $T_i^1, \ldots, T_i^n$ are parameterized on the $\mathrm{SO}(3) \times \mathrm{T}(3)$ manifold, we can perturb them using their Lie algebra, as it was proven by [22, 23, 56] that particle filtering on Lie groups is coordinate-invariant, *i.e.*, same perturbation on different states (poses) results in the same motion.

Our approach. In general, the loss function in Eq. (1) is not convex over the 7D optimization space, and the convergence basin is highly affected by initialization. Thus the main challenges are: exploring efficiently the otherwise large hypothesis space, and increasing the convergence basin. To address the former, we rely on multi-hypothesis tracking [23] of multiple *beams*. With *beam* we denote a set of particles that evolves and is optimized independently from other *beams*. This is equivalent to having separate optimization threads. It allows to explore in breadth the state space [29], and if some threads get stuck in local minima, it does not affect the others. The beams are optimized in parallel, to augment the probability that some of them will move in the right direction. Additionally, to minimize the cost of these initial steps, candidates can be rendered at low resolution (256×340), as fine-grained details are not needed at this stage. Every N_0 iterations, a *resampling step* is performed. The best candidates are pooled among all the beams, and each of them is used to initialize a new beam that is optimized independently again for N_0 steps. In this way we avoid pursuing unpromising hypothesis as beams that do not converge to good poses are halted.
To enhance convergence probability, the *Coarse-to-Fine* strategy discussed in Sec. 3.1 is adopted. Following this reasoning, after N_1 *resampling steps*, we switch to shallower feature maps in our feature extractor. Additionally, over the iterations image resolution is gradually increased, while the number of beams and particles in each beam is decreased. This strategy enables to keep computational cost low, while balancing the need to explore in breadth the sample space in the beginning, and to have fine-grained comparisons as we refine further the pose. More details on these hyperparameters are given in Sec. 4.1, and the pseudo-code of the algorithm is provided in the Supplementary. The code will be publicly released upon acceptance.

3.3. Adapting to different domains

Given that our framework entails comparing query images against rendered candidates, it raises the question of whether we might need an adaptation technique to bridge the gap between domains. Recently, this issue was addressed in [80, 117], showing that matching performances are not hindered by the rendering domain. We extend their analysis since in our setup we compare dense feature maps, which is different than matching local descriptors.
We test different rendering domains (textured, colored, raw geometry) and find that these shifts indeed cause a discrepancy in the extracted features, meaning that the distance between a query and the rendered ground truth pose will not be 0. Nevertheless, we are not interested in absolute values, as the only requirement for our optimization to converge is that relative differences in pose are reflected by relative changes in similarity. Figure 4 exemplifies this effect.
We show that in practice this assumption holds, and different domains only affect absolute values, preserving relative differences, since the rendered images domain is uniform. This finding highlights an advantage of our formulation, being agnostic to the scene representation.

4. Experiments

Datasets. We evaluate our pose refinement approach on multiple datasets. Aachen Day-Night v1.1 [93, 96, 117] is a common benchmark for large-scale localization [90, 92]; it contains 6,697 day-time database images and 1,015 queries, collected by handheld devices. Beyond the large area that it covers, it contains night queries and strong view-

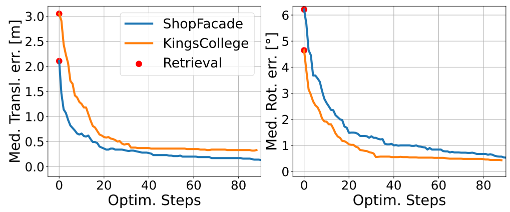

Figure 5. **Optimization trajectory**. Behavior of median errors over the iterations for 2 scenes from Cambridge Landmarks.

point changes between the database and query images. We also evaluate on smaller datasets widely used in the literature, namely Cambridge Landmarks [52] and 7scenes [103]. They contain respectively 5 outdoor and 7 indoor scenes. In both datasets query sequences are captured along different trajectories w.r.t. the available database. Following common practices, we report for Aachen the recall at thresholds $(25\text{cm}, 2°)$, $(50\text{cm}, 5°)$, and $(5\text{m}, 10°)$ [90, 96], while for the remaining datasets we evaluate median translation (m) and rotation (°) errors [67, 72, 73].

Coarse Features	Fine Features	ShopFacade	OldHospital
ResNet-18			
CosPlace [8]	ImageNet	12 / 0.45	39 / 0.73
ImageNet	ImageNet	12 / 0.55	46 / 0.80
SimCLR [21]	SimCLR [21]	18 / 0.62	50 / 0.83
ALIKED [118]	ALIKED [118]	17 / 0.64	49 / 0.84
AlexNet [55]	AlexNet [55]	15 / 0.74	53 / 0.88

Table 1. **Ablation on feature extractors**, shows that the property of dense features being robust estimators of *visual similarity* holds across architectures and training protocols. Errors in cm, °.

4.1. Implementation details

For our main experiments, we adopt a lightweight ResNet-18 [43] trained for Place Recognition in [8]. This network was fine-tuned from *conv3*, freezing earlier layers. Thus when in our optimization we switch to *conv2*, we are actually using ImageNet features. We find that this network slightly improves results over vanilla ImageNet (see Tab. 1). We hypothesize that being trained for Place Recognition, deeper layers have learnt to focus on buildings and landmarks, ignoring transient objects, which is useful also for localization. After N_1 steps, we switch to *conv2* features, pre-trained only on ImageNet; finally, the last refinement steps (after N_2 iterations), are performed with *conv1*. The choice of N_1, N_2 is not critical to achieve good results; what matters is that initial steps are carried out with coarser features, and the very last with finer ones, as *conv1* features present a narrower converge basin. In the Supplementary we report experiments to demonstrate robustness to these hyperparameters, and a convergence analysis. Their values also depend on the use-case: when using our algorithm as post-processing, there is no need to start from Coarse features; viceversa when acting as pre-processing we do not use the shallower features. For the standalone experiments starting from retrieval poses, we set N_1 to 30 for all datasets, and N_2 to $N_t - 10$, *i.e.* the last 10 steps are with *conv1*.
In Tab. 1, we ablate the choice of network, showing that our method works with any off-the-shelf architecture. For perturbing the camera center, we use Gaussian noise, with the only precaution that the standard deviation on the vertical axis is reduced to $1/10$ wrt the other directions. This comes from the prior knowledge that while we need to explore the scene in breadth, as initialization can be far off, height variations are typically limited to human size, thus we can avoid wasting resources exploring the vertical axis. We perturb the rotation around a random axis, with uniform noise. The magnitude of the noise is reduced linearly, and every $N_0 = 20$ steps it is reset. This *start-and-stop* scheduling is akin to the CosineAnnealing strategy [68].

Renderable scene representations. The only requirement for our method is to have a renderable model of the scene; *i.e.* that allows to generate a view given any pose $(R, t) \in \text{SE}(3)$. Recently, [80, 81] highlighted the advantages of 3D meshes and how they can be obtained. Specifically, these advantages are flexibility of supporting different tasks, and efficient rendering, thanks to rendering pipelines being tailored to meshes for decades. As an alternative to meshes, modern neural radiance fields [71, 108] offer photorealistic renderings, at the cost of being typically slower. Although several efforts greatly cut down on NeRF rendering times [75, 87], they remain at least 1 order of magnitude slower than mesh-based rendering. In light of this, we experiment with Gaussian Splatting [53], which offers high-quality images and matches the speed of a mesh. In our experiments we find that a low-detail, compressed mesh is sufficient to achieve good results. On the large scale scene of Aachen [93, 96, 117], we use the models provided by [80]. Thanks to the mesh being compressed, and the low resolutions that we adopt, a textured image can be rendered efficiently in $500\mu s$. For smaller scenes of Cambridge Landmarks [52] and 7scenes [103], starting from the point cloud of the dataset we optimize a set of 3D Gaussians following [53], which requires only $10min$, and can then be rendered in $600 - 900\mu s$, depending on the scene. Times were measured on a RTX 4090 GPU.

4.2. Experimental results

In this section, we perform an ablative study to support our choices and the motivations of the paper, together with visualizations to highlight salient aspects. We then validate our results on against state-of-the-art (sota) matching methods, pose regressors and implicit features-based refiners.

Ablation studies. Tab. 1 ablates different architectural

Method		Cambridge Landmarks			
		King's	Hospital	Shop	St. Mary's
Retrieval					
DenseVLAD [110]	-	2.8/5.7	4.0/7.1	1.1/7.6	2.3/8.0
CosPlace [8]	-	3.1/4.4	4.5/6.7	2.1/6.2	3.2/7.2
SOTA					
AS [95]†	-	0.13/0.22	0.20/0.36	0.04/0.21	0.08/0.25
hloc [90]	TL	0.12/0.20	0.15/0.30	0.04/0.20	0.07/0.21
DSAC* [12]	TS	0.15 / 0.3	0.21 / 0.4	0.05 / 0.3	0.13 / 0.4
HACNet [62]	TS	0.18 / 0.3	0.19 / 0.3	0.06 / 0.3	0.09 / 0.3
PixLoc [92]	TL	0.14/0.24	0.16/0.32	0.05/0.23	0.10/0.34
Pose Regressors					
MS-Transformer [102]	TS	0.83 / 1.47	1.81 / 2.39	0.86 / 3.07	1.62 / 3.99
DFNet [19]	TS	0.73 / 2.37	2 / 2.98	0.67 / 2.21	1.37 / 4.03
LENS [72]	TS	0.33 / 0.5	0.44 / 0.9	0.25 / 1.6	0.53 / 1.6
Pose Refiners					
FQN [40]	TS	**0.28** / **0.4**	0.54 / 0.8	0.13 / 0.6	0.58 / 2.0
CROSSFIRE [73]	TS	0.47 / 0.7	0.43 / **0.7**	0.2 / 1.2	0.39 / 1.4
NeFeS (DFNet) [20]	TS	0.37 / 0.62	0.55 / 0.9	0.14 / 0.47	0.32 / 0.99
MCLoc (ours)	-	0.31 / 0.42	**0.39** / 0.73	**0.12** / **0.45**	**0.26** / **0.88**

Table 2. **Results on the Cambridge Landmarks dataset.** We show that our simple approach outperforms methods that train per-scene descriptors. TM marks methods trained for feature matching, TL trained for localization, TS trained per scene.

Method	Aachen Day-Night v1.1	
	Day	Night
Retrieval		
NetVLAD [3]	0.0 / 0.2 / 18.9	0.0 / 0.0 / 14.3
CosPlace [8]	0.0 / 0.4 / 27.1	0.0 / 0.0 / 24.1
Pose Refiners		
Pixloc [92]	63.2 / 67.8 / 75.5	38.7 / 47.1 / 60.7
MCLoc (ours)	55.8 / 73.3 / 89.7	42.4 / 66.5 / 86.9
Matching based		
AS [95]	85.3 / 92.2 / 97.9	39.8 / 49.0 / 64.3
hloc [90]	87.4 / **95.0** / 98.1	71.7 / **88.5** / **97.9**
+ PixLoc refine	86.2 / 94.9 / 98.1	70.8 / **88.5** / **97.9**
+ **(ours)** refine	**87.9** / 94.9 / **98.9**	**73.8** / **88.5** / **97.9**

Table 3. **Large scale Visual localization on Aachen v1.1 dataset.** We show competitive results against PixLoc refinement, and how our method can be coupled with sota pipelines to improve results.

choices. For all architectures, we extract dense features and use them as in Eq. (2). While CosPlace+ImageNet achieves slightly superior results, our method works regardless of the architecture, training protocol and/or dataset. These results expand on the findings of the LPIPS paper [116], proving the *unreasonable effectiveness* of generic features not only for perceptual similarity, but for pose similarity as well. In particular, Tab. 1 shows that dense features, trained either on supervised or unsupervised objectives, for generic classification, place recognition or feature matching (ALIKED [118]), show the same property that is visualized in Fig. 3. That is, ability to estimate image alignment, with high precision in shallower layers, and with wider baselines in deeper features. These findings also align with the foundation behind transfer learning [28], a cornerstone of modern computer vision, which can be summarized in very simple words as "a good feature is a good feature anywhere" [116]. These networks, regardless of the architecture or the pre-training task, learned how to extract generic features, and we can exploit the natural spatial structure of the feature maps to discriminate pose variations. In the Supplementary we also provide an ablation on different scoring functions, showing that a simple, dense pixelwise comparison is *all you need*, against more elaborate formulations.

Baselines. To evaluate whether the generic features that we adopt are competitive, we benchmark against methods that train per-scene.

- **Pose Regressors**: These method train a network to directly predict the camera pose. We consider DFNet [19], LENS [72] and MS-Transformer [102]
- **Pose Refiners**: These are the closest to our method. Among them, FQN [40], NeFeS [20] and CROSSFIRE [73] optimize per-scene descriptors in an implicit field and as such are limited to small scenes. PixLoc [92] trains features specific for localization on MegaDepth [65]. Their localization pipeline minimizes a feature-metric objective with first order methods (FQN/NeFeS/Crossfire) or second order optimizers (PixLoc), whereas we rely on a simple MonteCarlo algorithm
- **Matching based**: These methods represent the state-of-the-art. We show how our method can be coupled with them to further improve performances

MeshLoc [80] pipeline	Top K Matched	Aachen Night v1.1
Textured Mesh		
LoFTR [105]	50	73.3 / 89.0 / 95.8
LoFTR [105]	20	71.2 / 89.0 / 94.8
LoFTR [105]	10	70.7 / 86.4 / 94.8
(ours) + LoFTR [105]	20	**74.3** / **91.1** / **99.5**
(ours) + LoFTR [105]	10	73.8 / **91.1** / 99.1
Raw Geometry		
P2P[119] + SG [91]	50	8.4 / 27.7 / 60.7
P2P[119] + SG [91]	10	6.8 / 20.4 / 52.4
(ours) + P2P[119] + SG [91]	1	16.8 / 37.7 / 66.0

Table 4. **Preprocessing on Aachen Night**: MCLoc can improve initial poses from retrieval, before a more expensive localizer.

Comparison with methods trained per-scene. In Tab. 2 we compare our method mainly against other refinement methods [20, 40, 73] and pose regressors [19, 72] on the Cambridge Landmarks benchmark [52]. The main rationale of this set of experiments is to compare against approaches that train scene-specific descriptors and/or representations for localization. While pose regressors surely achieve the faster inference time, they generally perform worse. Despite the absence of any kind of fine-tuning, we outperform all implicit feature-based refiners, except a small gap on King's College were FQN is slightly better. On these datasets, our optimization converges in 80 refinement steps, although satisfying results are achieved much earlier.

Fig. 5 shows the optimization trajectory on 2 distinct scenes. Overall, we conclude that to achieve satisfying results there is no need to fine-tune per-scene, or at all. Among methods that train per-scene, Scene Coordinate Regressors [12, 62] perform best.

Method	7 scenes: DSLAM ground truths median error in (cm/°) ↓						
	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
Retrieval							
DenseVLAD [110]	21/12.5	33/13.8	15/14.9	28/11.2	31/11.3	30/12.3	25/15.8
CosPlace [8]	31 / 11.4	45 / 14.6	23 / 13.7	43 / 11.2	52 / 11.4	48 / 11.1	46 / 14.8
SOTA							
AS [95]	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01
DSAC [13]	2/1.10	2/1.24	1/1.82	3/1.15	4/1.34	4/1.68	3/1.16
HACNet [62]	2/0.7	2/0.9	1/0.9	3/0.8	4/1.0	4/1.2	3/0.8
hloc [90]	2/0.85	2/0.94	1/0.75	3/0.92	5/1.30	4/1.40	5/1.47
Pose Regressors							
MS-Transf. [102]	11 / 4.7	24 / 9.6	14 / 12.2	17 / 5.66	18 / 4.4	17 / 6.0	17 / 5.9
DFNet[19]	5 / 1.9	17 / 6.5	6 / 3.6	8 / 2.5	10 / 2.8	22 / 5.5	16 / 2.4
LENS [72]	3 / 1.3	10 / 3.7	7 / 5.8	7 / 1.9	8 / 2.2	9 / 2.2	14 / 3.6
Pose Refiners							
FQN-PnP [40]	4 / 1.3	10 / 3.0	4 / 2.4	10 / 3.0	9 / 2.4	16 / 4.4	140 / 34.7
CROSSFIRE [73]	1 / 0.4	5 / 1.9	3 / 2.3	5 / 1.6	3 / 0.8	2 / 0.8	12 / 1.9
MCLoc (ours)	5 / 1.8	4 / 2.0	4 / 1.9	10 / 3.6	10 / 3.7	8 / 3.1	10 / 2.5
	SFM ground truths [14]						
MS-Transf. [102]	11 / 6.4	23 / 11.5	13 / 13.0	18 / 8.1	17 / 8.4	16 / 8.9	29 / 10.3
DFNet [19]	3 / 1.1	6 / 2.3	4 / 2.3	6 / 1.5	7 / 1.9	7 / 1.7	12 / 2.6
NeFeS [20]	2 / 0.8	2 / 0.8	2 / 1.4	2 / 0.6	2 / 0.6	2 / 0.6	5 / 1.3
MCLoc (ours)	2 / 0.8	3 / 1.4	3 / 1.3	4 / 1.3	5 / 1.6	6 / 1.6	6 / 2.0
(ours) w. DINOv2 [79]	3 / 0.9	4 / 1.8	3 / 1.5	6 / 1.4	7 / 2.1	8 / 1.8	9 / 2.2
(ours) w. RoMa [32]	2 / 0.7	3 / 1.2	2 / 1.0	3 / 1.1	4 / 1.0	5 / 1.4	6 / 1.5

Table 5. **Indoor localization**. Indoor scenarios are challenging for our algorithm. Despite this, we achieve competitive results.

Large Scale Localization. Tab. 3 reports results on the large scale benchmark of Aachen v1.1 [93, 96, 117]. The objective of these experiments is to demonstrate the applicability of our method in this scenario in which the previously considered competitors in Tab. 2, namely pose regressors and refiners based on implicit fields, fail to scale. In this setting we mainly compare against PixLoc [92], which is another refinement methods based on a similar idea to our *render&compare* framework, with a feature-metric error. We first report results starting from retrieval initialization, showing that our algorithm performs better (except on the finer threshold for Day queries), despite PixLoc trains end-to-end specialized features for localization. In the Supp. Mat. we further discuss trade-offs and similarities of our method with PixLoc, as well as computational cost.
Additionally, the table shows how our method can complement sota matching-based methods from hloc [90]. In this setup, we first run localization using the hloc pipeline. We use the estimated poses as an initialization for our method. Results show that we are able to obtain more accurate poses with just 5 refinement steps, adding little overhead.
Tab. 4 reports another use-case for our method. Recently, MeshLoc [80] has shown a localization pipeline for matching methods using rendered images and a mesh. Depth maps are used to lift the 2D-2D matches to 2D-3D, instead of relying on the SfM point cloud. We use our method to refine the initial estimate from retrieval; in this way we can provide our refined poses as initialization to a more accurate localizer. The table shows that, starting from our refined poses, it is possible to achieve a boost in performances while reducing the number of top-K candidates considered.

Indoor Localization. On 7scenes [103], as for Cambridge Landmarks, we compare against method that train on each scene. Indoor scenarios are more challenging for our method, since it is common to have repetitive, textureless surfaces (e.g. walls, floor), which don't provide a meaningful signal for *perceptual similarity*. Despite this, we achieve comparable performances, at the cost of increasing the number of iterations. Another factor that affects the evaluation on 7scenes is Ground Truth (GT) accuracy. [14] demonstrated that the original DSLAM labels are inaccurate, and relased an updated version, named SFM labels. On these more accurate GTs, our results are more competitive. We also test our approach with DINOv2 [79], which leads to comparable precision wrt ImageNet features. This is due to the fact that ViT-based models have a fixed patch size and thus coarser, less-localizable features. To this end, we test the approach from RoMa [32], which refines DINO features, and found that they surpass or match other specialized pose refiners. However, RoMA was trained for feature matching, an essential step for pose estimation. The improvement suggests that task-specific training can of course improve performance, opening up interesting directions for test-time optimization. More details on these methods are discussed in the Supp. Mat..

5. Conclusion

In this work, we investigated whether generic pre-trained features can be transferred to the localization task, thus removing the burden of training dedicated descriptors. Building on the notion that dense feature are robust estimators of *perceptual similarity*, we showed a connection between the latter and *pose similarity*. We demonstrated that this link can be exploited to construct a refinement algorithm within a render & compare framework, paired with MonteCarlo sampling. Experimental results exhibit that our **MCLoc** can be applied in both large and small scenes, either as a standalone refiner or paired with more accurate localizers, and that it can outperform several competitor approaches that optimize dedicated descriptors, especially in outdoor scenarios.

Acknowledgements This work was supported by CINI, the European Lighthouse on Secure and Safe AI – ELSA, HORIZON EU Grant ID: 101070617, Czech Science Foundation (GACR) EXPRO (grant no. 23-07973X). We thank our colleagues who provided helpful feedback and suggestions, in particular Assia Benbihi and Vojtech Panek.

References

[1] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based slam. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*, pages 324–341. Springer, 2017. 3

[2] Seyed Ali Amirshahi, Marius Pedersen, and Stella X. Yu. Image quality assessment by comparing cnn features between images. In *Image Quality and System Performance*, 2016. 4

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 2, 7

[4] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, pages 690–708. Springer, 2022. 1

[5] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006. 1

[6] Simon Baker, Ralph Gross, and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56, 2003. 3

[7] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[8] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *CVPR*, 2022. 2, 6, 7, 8

[9] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep visual geo-localization benchmark. In *CVPR*, 2022. 2

[10] Michael Bloesch, Tristan Laidlow, Ronald Clark, Stefan Leutenegger, and Andrew Davison. Learning meshes for dense visual slam. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5854–5863, 2019. 1

[11] Eric Brachmann and Carsten Rother. Learning Less is More-6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 2

[12] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *TPAMI*, 2021. 7, 8

[13] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 2, 8

[14] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6218–6228, 2021. 8

[15] Jan Brejcha, Michal Lukáč, Yannick Hold-Geoffroy, Oliver Wang, and Martin Čadík. Landscapear: Large scale outdoor augmented reality by matching photographs with terrain models using learned descriptors. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*, page 295–312, Berlin, Heidelberg, 2020. Springer-Verlag. 1, 2

[16] Benjamin Busam, Tolga Birdal, and Nassir Navab. Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2436–2445, 2017. 3, 4

[17] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In *3DV*, 2019. 2

[18] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (10):2465–2477, 2020. 2

[19] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022. 2, 7, 8

[20] Shuai Chen, Yash Bhalgat, Xinghui Li, Jiawang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Refinement for absolute pose regression with neural feature synthesis. *arXiv preprint arXiv:2303.10087*, 2023. 2, 3, 7, 8

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 6

[22] A. Chiuso and S. Soatto. Monte carlo filtering on lie groups. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, pages 304–309 vol.1, 2000. 2, 3, 5

[23] Changhyun Choi and Henrik Christensen. Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features. *Intl. Jour. of Robotics Research*, 33, 2012. 3, 4, 5

[24] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, pages 236–243. Springer, 2003. 1

[25] Mark J Cummins and Paul M Newman. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 3–10, 2010. 1

[26] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018. 1

[27] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera re-localization. In *ICCV*, 2019. 2

[28] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 7

[29] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005. 5

[30] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 1

[31] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019. 1

[32] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Revisiting robust losses for dense feature matching. *arXiv preprint arXiv:2305.15404*, 2023. 8

[33] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *TPAMI*, 40(3):611–625, 2017. 3

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

[35] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *Aaai/iaai*, 1999(343-349):2–2, 1999. 1

[36] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of artificial intelligence research*, 11:391–427, 1999. 1

[37] Dieter Fox, Sebastian Thrun, Wolfram Burgard, and Frank Dellaert. *Particle Filters for Mobile Robot Localization*, pages 401–428. Springer New York, New York, NY, 2001. 3, 4

[38] Fei Gao, Yi Wang, Panpeng Li, Min Tan, Jun Yu, and Yani Zhu. Deepsim: Deep similarity for image quality assessment. *Neurocomputing*, 257, 2017. 4

[39] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4

[40] Hugo Germain, Daniel DeTone, Geoffrey Pascoe, Tanner Schmidt, David Novotny, Richard Newcombe, Chris Sweeney, Richard Szeliski, and Vasileios Balntas. Feature query networks: Neural surface description for camera pose refinement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5067–5077, 2022. 2, 3, 7, 8

[41] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 1

[42] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[44] Sixing Hu and Gim Hee Lee. Image-based geo-localization using satellite imagery. *International Journal of Computer Vision*, 128(5):1205–1219, 2020. 3

[45] Nicolas Hudson, Fletcher Talbot, Mark Cox, Jason Williams, Thomas Hines, Alex Pitt, Brett Wood, Dennis Frousheger, Katrina Lo Surdo, Thomas Molnar, et al. Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge. *arXiv preprint arXiv:2104.09053*, 2021. 1

[46] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020. 2

[47] 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*, 130(7):1811–1836, 2022. 2, 4

[48] Wardah Inam. Particle filter based self-localization using visual landmarks and image database. In *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*, pages 246–251, 2009. 3

[49] Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *Conference on robot learning*, pages 169–178. PMLR, 2018. 3, 5

[50] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32:29:1–29:13, 2013. 1

[51] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 2

[52] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *ICCV*, 2015. 4, 6, 7

[53] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 6

[54] Jongyoo Kim and Sanghoon Lee. Deep learning of human visual sensitivity in image quality assessment framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 6

[56] Junghyun Kwon and Frank C. Park. Visual tracking via particle filtering on the affine group. In *2008 International Conference on Information and Automation*, pages 997–1002, 2008. 3, 5

[57] Junghyun Kwon, Minseok Choi, Frank Park, and Changmook Chun. Particle filtering on the euclidean group: Framework and applications. *Robotica*, 25:725–737, 2007. 2, 3, 4

[58] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022. 1, 2

[59] Viktor Larsson, Torsten Sattler, Zuzana Kukelova, and Marc Pollefeys. Revisiting Radial Distortion Absolute Pose. In *ICCV*, 2019. 2

[60] Vincent Lepetit and Pascal Fua. *Monocular Model-Based 3D Tracking of Rigid Objects: A Survey*. 2005. 4

[61] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 3

[62] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 7, 8

[63] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3D point clouds. In *ECCV*, 2012. 2

[64] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 2

[65] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 7

[66] Yunzhi Lin, Thomas Müller, Jonathan Tremblay, Bowen Wen, Stephen Tyree, Alex Evans, Patricio A Vela, and Stan Birchfield. Parallel inversion of neural radiance fields for robust pose estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9377–9384. IEEE, 2023. 2, 3, 4, 5

[67] Jianlin Liu, Qiang Nie, Yong Liu, and Chengjie Wang. Nerf-loc: Visual localization with conditional neural radiance field. *arXiv preprint arXiv:2304.07979*, 2023. 2, 6

[68] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[69] Dominic Maggio, Marcus Abate, Jingnan Shi, Courtney Mario, and Luca Carlone. Loc-nerf: Monte carlo localization using neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4018–4025. IEEE, 2023. 3, 5

[70] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 3

[71] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 2, 6

[72] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022. 2, 6, 7, 8

[73] Arthur Moreau, Nathan Piasco, Moussab Bennehar, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Crossfire: Camera relocalization on self-supervised features from an implicit representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 252–262, 2023. 2, 3, 6, 7, 8

[74] Markus S. Mueller, Thorsten Sattler, Marc Pollefeys, and Boris Jutzi. Image-to-image translation for enhanced feature matching, image retrieval and visual localization. *ISPRS annals*, IV-2/W7:111–119, 2019. 1

[75] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41 (4):102:1–102:15, 2022. 6

[76] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[77] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss. Robust visual localization across seasons. *IEEE Transactions on Robotics*, 34(2):289–302, 2018. 1

[78] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327, 2011. 1

[79] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 8

[80] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-Based Visual Localization. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 5, 6, 7, 8

[81] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Visual localization using imperfect 3d models from the internet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13175–13186, 2023. 1, 6

[82] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *Proceedings of the European conference on computer vision (ECCV)*, pages 318–332, 2018. 1

[83] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. Segloc: Learning segmentation-based representations for privacy-preserving visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15380–15391, 2023. 3

[84] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking image retrieval for visual localization. In *3DV*, 2020. 2

[85] Christian Poglitsch, Clemens Arth, Dieter Schmalstieg, and Jonathan Ventura. [poster] a particle filter approach to outdoor localization using image-based rendering. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, pages 132–135, 2015. 3, 5

[86] Gerard Pons-Moll, Vladimir Guzov, Julian Chibane, Riccardo Marin, Yannan He, and Torsten Sattler. Interaction replica: Tracking human-object interaction and scene changes from human motion. 2023. 1

[87] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *International Conference on Computer Vision (ICCV)*, 2021. 6

[88] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noé Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1

[89] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Conference on Robot Learning*, pages 456–465. PMLR, 2018. 1, 2

[90] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 2, 5, 6, 7, 8

[91] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 7

[92] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Victor 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 *CVPR*, 2021. 2, 3, 5, 7, 8

[93] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 5, 6, 8

[94] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, page 2102–2110, USA, 2015. IEEE Computer Society. 2

[95] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *TPAMI*, 39(9):1744–1756, 2016. 2, 7, 8

[96] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 1, 5, 6, 8

[97] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2

[98] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 2

[99] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, 2018. 2

[100] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019. 3

[101] Qi Shan, Changchang Wu, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M. Seitz. Accurate geo-registration by ground-to-aerial image matching. In *Proceedings of the 2014 2nd International Conference on 3D Vision - Volume 01*, page 525–532, USA, 2014. IEEE Computer Society. 2

[102] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. *arXiv preprint arXiv:2103.11468*, 2021. 2, 7, 8

[103] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 6, 8

[104] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 1

[105] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 1, 7

[106] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *TPAMI*, 2019. 2

[107] Hajime Taira, Ignacio Rocco, Jiri Sedlar, Masatoshi Okutomi, Josef Sivic, Tomas Pajdla, Torsten Sattler, and Akihiko Torii. Is this the right place? geometric-semantic pose verification for indoor visual localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4373–4383, 2019. 2

[108] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 6

[109] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2005. 2, 4

[110] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 7, 8

[111] Gabriele Trivigno, Gabriele Berton, Juan Aragon, Barbara Caputo, and Carlo Masone. Divide&classify: Fine-grained classification for city-wide visual geo-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11142–11152, 2023. 2

[112] Jonathan Ventura, Zuzana Kukelova, Torsten Sattler, and Dániel Baráth. P1ac: Revisiting absolute pose from a single affine correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19751–19761, 2023. 1

[113] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. GN-Net: The Gauss-Newton loss for multi-weather relocalization. *RA-L*, 5(2):890–897, 2020. 3

[114] Lukas Von Stumberg, Patrick Wenzel, Nan Yang, and Daniel Cremers. LM-Reloc: Levenberg-Marquardt based direct visual relocalization. In *3DV*, 2020. 2, 3

[115] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021. 2, 3

[116] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2, 3, 4, 7

[117] 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*, 129:821–844, 2021. 2, 5, 6, 8

[118] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 2023. 6, 7

[119] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4669–4678, 2021. 7

[120] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 1