Learning Neural Volumetric Pose Features for Camera Localization

Jingyu Lin[1], Jiaqi Gu[2], Bojian Wu[3], Lubin Fan[2(✉)], Renjie Chen[1(✉)], Ligang Liu[1], and Jieping Ye[2]

[1] University of Science and Technology of China, Hefei, China
sa1022@mail.ustc.edu.cn, {renjiec,lgliu}@ustc.edu.cn
[2] Alibaba Cloud, Hangzhou, China
{gujiaqi.gjq,lubin.flb,yejieping.ye}@alibaba-inc.com
[3] Zhejiang University, Hangzhou, China
ustcbjwu@gmail.com
https://gujiaqivadin.github.io/posemap/

Abstract. We introduce a novel neural volumetric pose feature, termed PoseMap, designed to enhance camera localization by encapsulating the information between images and the associated camera poses. Our framework leverages an Absolute Pose Regression (APR) architecture, together with an augmented NeRF module. This integration not only facilitates the generation of novel views to enrich the training dataset but also enables the learning of effective pose features. Additionally, we extend our architecture for self-supervised online alignment, allowing our method to be used and fine-tuned for unlabelled images within a unified framework. Experiments demonstrate that our method achieves 14.28% and 20.51% performance gain on average in indoor and outdoor benchmark scenes, outperforming existing APR methods with state-of-the-art accuracy.

Keywords: Absolute pose regression · Pose feature · NeRF

1 Introduction

Image-based camera localization is a key task in both academic and industrial communities, which is a core component for various applications, such as reconstruction [14], navigation [2], and so on. With a reference world coordinate system, the goal of camera localization is to compute the absolute camera pose, such as position and orientation, from the captured image.

Recently, Absolute Pose Regression (APR) methods have attracted more attention. Compared with traditional structure-based methods, such as Structure-from-Motion (SfM), its efficiency and better understanding of ambiguous images with textureless or repetitive patterns are more intriguing. The problem is formulated as training a regressor in a supervised manner, by taking a

This work was done when Jingyu Lin was an intern at Alibaba Cloud. J. Lin and J. Gu—Equal contributions.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-72995-9_12.

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15103, pp. 198–214, 2025.
https://doi.org/10.1007/978-3-031-72995-9_12

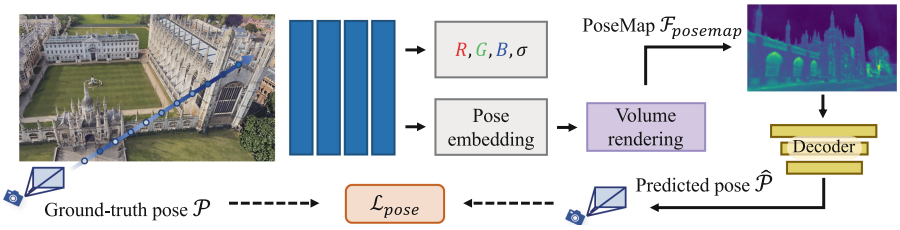


Fig. 1. The generation of PoseMap. To capture the implicit pose characteristics, we enhance the original NeRF by introducing a unique pose embedding. Subsequently, we generate a PoseMap through volumetric rendering. We believe that the learned volumetric features integrate the implicit information of camera pose and can be used to improve the accuracy of camera localization tasks.

set of images and the associated poses estimated usually by SfM [32], as inputs for network training. Subsequently, the trained networks can directly regress the camera pose for the query image [7]. Due to its friendly storage and computational efficiency, the APR methods are promising research directions for camera localization.

As discussed in [31], APR resembles pose approximation through image retrieval, implying that the performance is proportional to the scale of the observed scene images. With the advent of neural rendering techniques which can synthesize images from any novel viewpoints efficiently, Neural Radiance Fields (NeRF) [23] has been incorporated into camera localization frameworks to augment both offline and online data. For instance, DFNet [7] and LENS [25] make efforts to synthesize additional images that share the feature distribution of the training images, to narrow the domain gap between real and synthetic images. It leads to significant improvement in the localization problem.

Our key observations are two-fold. First, the existing methods predominantly utilize the forward pass of NeRF for rendering, overlooking the potential to explore the richly encoded features within the learned volumetric field, which is an aggregation of information from images and poses, suggesting that the volumetric field of NeRF should inherently encapsulate implicit information regarding to each camera pose. Second, it is widely acknowledged that occlusions, edges, and shadows serve as explicit clues for camera localization, typically encoded as deep features within neural networks. This intuitively implies that pose estimation should be related with these feature maps.

We propose a scene-specific neural volumetric pose feature, termed PoseMap, for camera localization. As mentioned above, in order to better explore the features from the neural radiance field, by extending the original NeRF with a pose branch (NeRF-P), see Fig. 1, we augment the pose estimation with an additional neural feature embedding. Experimental results demonstrate that PoseMap can learn the discriminative visual and geometric features, which could distinguish query images from different viewpoints. Quantitatively, the PoseMap for APR training improves the localization accuracy by 14.28% and 20.51% on average over all test sets than the SOTAs. The contributions are summarized as follows.

- We are the first to propose a neural volumetric pose feature, called PoseMap, for camera localization. It explores the implicit information of the image and the corresponding camera pose encoded in the neural volumetric field.
- We devise an APR architecture with NeRF-P, integrated with PoseMap. Both the original APR losses and pose feature loss are combined for training. We also extend our architecture for self-supervised online fine-tuning, leading to a unified framework that is also suitable for unlabelled images.
- Our method achieves 14.28% and 20.51% performance gain on average in both indoor and outdoor benchmark datasets, which outperforms the existing APR methods and sets a state-of-the-art accuracy.

2 Related Work

Camera localization typically consists of two distinct stages: in-sample data acquisition and out-of-sample testing as in Table 1. The initial stage involves capturing a set of in-sample images from an unknown scene and determining the corresponding camera poses. To achieve this, the structure-based methods [26,32] utilize 2D keypoints matching to obtain camera poses while concurrently constructing a 3D structure. Alternatively, NeRF or 3D Gaussian Splatting based approaches [13,19] encapsulate and extract the pose information within their implicit models. During the out-of-sample testing phase, assuming that the poses of in-sample images have been precisely established, it is crucial to apply pose regression methods to unseen out-of-sample images, which are generally categorized into two main types. The Relative Pose Regression (RPR)-based methods [21,30,37–39] usually deduce the pose of an out-of-sample image in relation to the known poses of in-sample images, and the Absolute Pose Regression (APR)-based methods [7,16,17,40,41] seek to directly estimate the absolute pose within the global coordinate framework. After yielding the reliable pose prior, further refinement processes [6,24] can be invoked to iteratively enhance the camera pose estimation by leveraging pixel-wise features.

2.1 Absolute Pose Regression (APR)

The Absolute Pose Regression (APR) methods, can directly predict camera pose using convolutional neural networks. These end-to-end methods exhibit notable efficiency at query time and demonstrate promising accuracy. Kendall *et al.* [17] proposed PoseNet, pioneering Absolute Pose Regression using a pre-trained GoogLeNet and introduced Cambridge Landmarks dataset for the task. Other variants of PoseNet, such as incorporating Bayesian methods [15], LSTM [41] and projection loss [16], have significantly enhanced the original framework's performance, since these modifications address uncertainties, capture temporal dependencies and refine pose estimation accuracy.

Confronted with limited training data, researchers have pursued two main paradigms for the improvements of APR. Sequential-based methods [28,40]

Table 1. Principle categories of techniques for the camera localization task.

Stages	Category	Technology	Methods
1st: In-sample Building	3D model-based	SfM, SLAM	COLMAP [32], ORB-SLAM [26], ...
	NeRF-based	Pose-free NeRF	BARF [19], COLMAP-Free 3DGS [13], ...
2nd: Out-of-sample Testing	Relative Regression	Image Retrieval, 2D-3D Matching	HLoc [29], LazyVL [12], ...
	Absolute Regression	Absolute Pose Regression(APR)	PoseNet [17], DFNet [7], ...
	Iterative Refinement	Iterative Pose Optimization	NeFeS [6], CROSSFIRE [24], ...

leverage multi-frame images to learn both absolute and relative poses by constraining the input data with temporal hints. On the other hand, rendering-based methods propose robust data augmentation to generate more training examples. For example, VGGRegNet [27] used synthetic depth as a reference to render and create additional viewpoints with varied pitch, zoom, and yaw settings.

Several methods have also taken into account how the representation of the pose can affect regression outcomes. For example, MapNet [4] made efforts to convert rotation matrices into the logarithm of unit quaternions, aiming to ease the challenge associated with camera pose regression. RelocNet [1] introduced the concept of camera frustum overlaps to enhance the learning of pose features. Meanwhile, CamNet [10] suggested a bilateral frustum loss for challenging case sampling. Another approach, PAE [35] brought in camera pose auto-encoders as the pose representation, employing a teacher-student training strategy.

2.2 NeRF for APR

Neural Radiance Fields (NeRF) [23] apply differentiable volume rendering techniques to predict color and opacity from 3D position and 2D viewing direction. It has proven to be a valuable tool in a range of camera localization tasks with three main goals: offline data augmentation, online viewpoint synthesis, and feature extraction. Such as LENS [25] utilized a pretrained NeRF to densely sample virtual camera positions across the scene and built a synthetic dataset for training. DFNet [7] adopted an online rendering strategy to render the estimated pose to a novel image and then compare the similarity of features between the real and rendered images. NeRF-loc [38] integrated implicit 3D descriptors from a generalizable NeRF, providing comprehensive 3D scene information. CROSSFIRE [24] introduced an implicit local descriptor representation for 2D-3D iterative feature matching. NeFeS [6] employed a post-processing pose refinement strategy with a reliable pose prior.

Recently, certain approaches [9,11,18] have been delving deeper into extracting knowledge from neural volumes, encompassing both geometrical and semantic information. In the domain of camera localization, NeRF possesses an appealing characteristic of aggregating information across multiple perspectives, which retains the associations between images and poses. Theoretically, the neural volumes of NeRF inherently encapsulate implicit information about each camera pose. Drawing inspiration from these insights, we directly extract pose features from neural volumes and generate a PoseMap to represent the pose information, demonstrating superior performance.

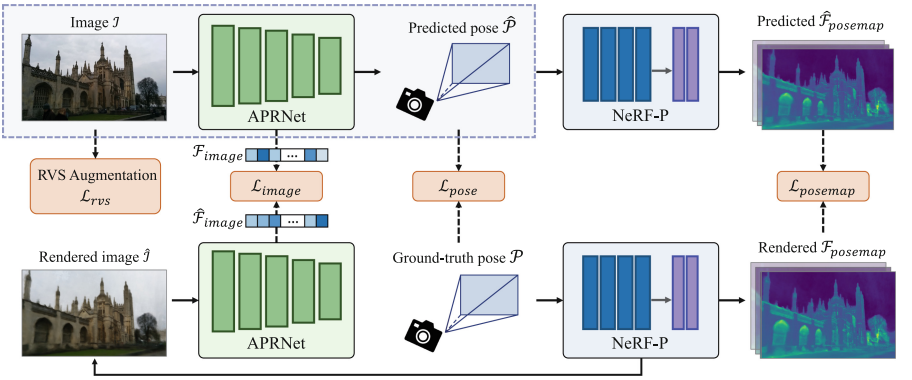


Fig. 2. Overview of the camera localization pipeline with PoseMap. The training stage of our pipeline, including two main modules: APRNet for camera pose regression and extracting image features and NeRF-P for view synthesis and extracting pose features. The inference stage of our pipeline with a simple APRNet for fast inference is highlighted in blue dotted box.

2.3 Positioning of Our Work

The discussions above have been exclusively focused on APR, as our method falls under this category as in Table 1. In terms of alternative techniques related to camera localization, we have classified them into several principal categories.

We emphasize that APR methods can deduce camera poses from unseen images using a single inference pass of a network, obviating the need for 2D-2D/3D matching with pre-built models or pose priors.

3 Method

Overview. Given a set of images and the associated camera poses $\{(I, p)\}$, our goal is to train a neural network that takes a query image I^* as input and directly outputs its corresponding camera pose p^*. Here the camera pose is represented as a 3×4 matrix which is a combination of translation and rotation with regard to a reference coordinate system. The whole pipeline is demonstrated in Fig. 2. It contains two main entangled modules: APRNet and NeRF-P. Concretely, with an input image I, APRNet leverages separate branches to extract image features $\mathcal{F}_{image}(I)$ and estimates the camera pose $\hat{p}$. With the given ground-truth pose, NeRF-P subsequently renders a synthetic image $\hat{I}$, which will be forwarded to the feature extraction branch of APRNet and obtain $\mathcal{F}_{image}(\hat{I})$. On the other hand, we propose a novel implicit pose features $\mathcal{F}_{pose}(\hat{p})$ called *PoseMap*, by enhancing the volumetric rendering module with an extra pose feature branch on original NeRF architecture, which will be further used for pose prediction. The key idea of this design choice is that NeRF itself is an abstraction of visual and geometric information. We propose an autoencoder-style pose branch to leverage NeRF to

aggregate global attributes from samples of light rays on PoseMap and a pose decoder of a series of MLP decoders for the distillation of implicit pose features. This novel combination allows for a more precise and detailed representation of the camera pose. In all, APRNet is optimized, with the supervision of the ground-truth PoseMap $\mathcal{F}_{pose}(p)$, and by minimizing the discrepancies of image features. This section is organized as follows: the camera localization pipeline and its core modules are discussed in Sect. 3.1, the neural volumetric pose feature PoseMap is detailed in Section 3.2, followed by a self-supervised scheme for online feature alignment using unlabelled images with the guidance of PoseMap in Sect. 3.3.

3.1 Camera Localization with PoseMap

Architecture. Our proposed camera localization architecture (Fig. 2) consists of two main modules: *APRNet* and *NeRF-P*, as introduced in the overview.

APRNet comprises two sub-networks: a pose estimator and an image feature extractor. The pose estimator predicts the camera pose $\hat{p}$ for an input image I. Meanwhile, the feature extractor generates deep features $\mathcal{F}_{image}(I)$ of I. The architecture of the APRNet is shown in Fig. 2. Similar to [7], the final image features are fused by feature maps from each convolutional block in the pose estimator. During the training stage, the pose estimator network is refined by aligning image features from the feature extractors. In the inference stage, the pose estimator is applied directly.

NeRF-P includes two distinct branches: the rendering branch and the volumetric pose extraction branch for generating PoseMap, which is the key component of our design. Unlike CROSSFIRE [24] and NeFeS [6] that extend NeRF to synthesize CNN image features, we focuses on establishing a direct relationship between implicit 3D information and camera poses. The details will be described in Sect. 3.2.

Pipeline. Given a set of input images paired with the camera poses $\{(I, p)\}$, NeRF-P undergoes a two-step training process. Initially, to maintain the quality of the original NeRF, the rendering branch is trained without considering pose features. Subsequently, the learned weights of the network are frozen, and the pose branch is trained independently. Utilizing the input image I, APRNet estimates a camera pose $\hat{p}$, and NeRF-P then generates the PoseMap $\mathcal{F}_{pose}(\hat{p})$ from $\hat{p}$. Conversely, with the ground-truth camera pose of I, NeRF-P also generates a PoseMap $\mathcal{F}_{pose}(p)$ and produces a synthetic image $\hat{I}$. The discrepancies between the image features from the real image I and the synthetic image $\hat{I}$, as well as the differences between pose features $\mathcal{F}_{pose}(\hat{p})$ and $\mathcal{F}_{pose}(p)$, are computed. The pose estimator of APRNet is optimized by minimizing domain gaps in both image features and pose features. To quantify the differences in image features, we apply the triplet loss from [7] as the image feature loss $\mathcal{L}_{image}$. In practice, we employ L_2 loss to compute $\mathcal{L}_{pose}$ and adopt cosine similarity loss for $\mathcal{L}_{posemap}$.

Considering the scalability of performance related to the number of observed images, we apply the NeRF-P module to synthesize more unseen views from

randomly perturbed poses, named random view synthesis (RVS). Given a camera pose p, a perturbed pose p_{rvs} is generated with a random translation and rotation. Given a synthetic image I_{rvs} with the perturbed pose p_{rvs}, the APRNet estimates its camera pose $\hat{p}_{rvs}$, followed by the NeRF-P to extract its pose feature map $\mathcal{F}_{rvs}$. The Same supervising strategies then are applied with the loss function defined as

$$\mathcal{L}_{rvs} = \|\hat{p}_{rvs} - p_{rvs}\|_2 + \|\hat{\mathcal{F}}_{rvs} - \mathcal{F}_{rvs}\|_2. \tag{1}$$

Loss Function. To summarize, the total loss for optimizing the pose estimator in the APRNet is:

$$\mathcal{L}_{total} = \mathcal{L}_{pose} + \mathcal{L}_{image} + \mathcal{L}_{posemap} + \mathcal{L}_{rvs}. \tag{2}$$

3.2 Neural Volumetric Features of PoseMap

Here, we introduce the detailed design of PoseMap, which is generated from the pose branch of the augmented NeRF, aka NeRF-P.

Given an image I, its appearance depends on the geometric and visual features of the scene and the camera settings, especially the camera pose. Intuitively, the visual clues, such as edges, shadows, and semantics, are proven to be crucial to camera localization. Thus, the existing methods leverage convolutional neural networks to extract deep features and subsequently predict the camera poses. However, most of the current methods are data-driven, which often struggle to deal with the data with sparse distribution because of a lack of global observation and any prior information on the whole scene.

Inspired by NeRF, which is a natural fusion of both images and poses, the volumetric representation of the scene can provide strong priors for the downstream applications, and avoid the aforementioned problems. Along with the neural rendering pipeline, we seek to explore the neural radiance field and separately abstract pose features from the volumetric feature space. Concretely, we learn a k-dimensional pose features vector for each sampled point on the camera ray and also apply a neural rendering approach to calculate the integrated features for each pixel, which is defined as PoseMap in our case. We sample N points along the ray and approximate the pose feature vector $\hat{F}$ by

$$\hat{F} = \sum_{i=1}^{N} w_i f_i, \tag{3}$$

$$\text{with } w_i = T_i \left(1 - \exp(-\sigma_i \delta_i)\right), T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{4}$$

where w_i is the weight of rendering, σ_i and f_i denote the density and pose feature vector at each sampled point i on the ray, and δ_j is the distance between adjacent samples. Figure 1 illustrates an example of PoseMap.

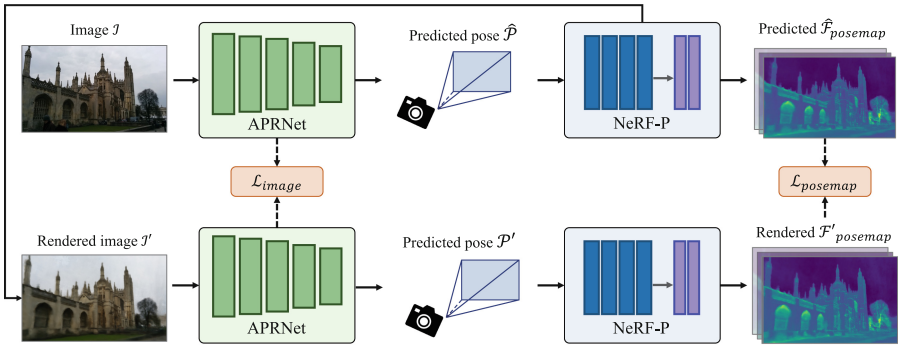


Fig. 3. Self-supervised online feature alignment scheme. We keep the $\mathcal{L}_{image}$ and $\mathcal{L}_{posemap}$ in the self-supervised pipeline. This scheme is suitable for any unlabelled images or images from the internet without matching with the 3D SfM model.

3.3 Self-supervised Feature Alignment with PoseMap

As previously noted, the accuracy of the APR-based method tends to correlate with the size of the observed scene images. We not only randomly synthesize labelled images but also incorporate unlabelled images with unknown camera poses to further enhance the training dataset, we propose a self-supervised scheme aimed at refining the pre-trained APRNet with PoseMap(refer to Fig. 3). When presented with an unlabelled image I, we utilize the pre-trained APRNet to estimate its camera pose, denoted as $\hat{p}$. Subsequently, the PoseMap $\mathcal{F}_{pose}(\hat{p})$ is generated from NeRF-P using $\hat{p}$, and a new unlabelled image I' is rendered. We follow the same procedure for image I' as we do for I, with the exception of omitting random view synthesis. The refinement of APRNet involves minimizing gaps in both image features and pose features. The loss for APRNet online feature alignment is

$$\mathcal{L}_{align} = \mathcal{L}_{image} + \mathcal{L}_{posemap}. \tag{5}$$

4 Experiments

In this section, we evaluate our camera localization method combined with pose features on standard benchmarks and perform comprehensive studies to demonstrate the effectiveness of our design.

4.1 Datasets and Implementation Details

Datasets. We evaluate our method on two visual localization datasets covering both indoor and outdoor scenes:

– 7-scenes [36]: it consists of 7 small-scale (about $1 \sim 18m^2$) static indoor scenes and up-to 7000 training images that are captured by a Kinect RGB-D sensor.

The test sequences have completely different paths from train sequences which make it to be a difficult dataset for camera estimation tasks.
- Cambridge Landmarks [17]: it contains 4 large-scale (about $900 \sim 5500m^2$) dynamic outdoor scenes with images and large SfM reconstructions, including all the recovered camera poses. The training sequences contain $200 \sim 1500$ samples, and testing sets are captured from different sequences.

For all studies, we use the ground-truth image-pose pairs for training. For each scene, camera poses are obtained from SfM or SLAM and later re-aligned and re-centered in $SE(3)$ to relieve the scale and distribution bias. In the following experiments, we report the median translation (m) and rotation (°) errors for evaluations.

NeRF-P Settings. We implement NeRF-P based on the source code from [7] and use the default parameters for fore-/background separation. An MLP encoder after the static branch is implemented to recover a pose embedding. Then the PoseMap is rendered as the size of $H \times W \times C$, where C is 256 according to experiments. The PoseMap is downsampled by a factor of 16 with the $H \times W$ dimensions flattened, followed by 4 layers of MLPs $(1536, 256, 128, 12)$ to yield the predicted pose. Practically, we first train the NeRF-P without the pose branch, then freeze the main weights and only train the pose branch and decoder.

Pipeline Settings. We train the network on each dataset from scratch. The APRNet adapts a pre-trained VGG-16 on ImageNet as the backbone and applies the Adam optimizer with a learning rate of 1×10^{-4} for training. The image feature embedding is extracted following the procedure as [7]. To synthesize random views, we add random perturbations (i.e., $\delta_t = [0.2, 0.2, 0.2]m, \delta_r = [10, 10, 10]^\circ$ for indoor scenes and $\delta_t = [3.0, 3.0, 3.0]m, \delta_r = [7.5, 7.5, 7.5]^\circ$ for outdoor scenes) on the training poses. For fair comparison to other methods, we use 20%/50% of the original data for indoor/outdoor scenes training, respectively.

Self-supervised Settings. At the online feature alignment stage, we load the pre-trained weights of APRNet and refine it with images without ground-truth camera poses. Following the setting in [4,7,8], we use 10% (indoor) and 50% of images (outdoor) without ground-truth poses from the validation dataset, respectively. The batch size is 1 and the learning rate is 1×10^{-5}. We name the pipeline before and after self-supervised alignment with unlabelled data as **PMNet** and **PMNet**$_{ud}$, respectively.

Other Details. We utilize a multi-stage training schedule, starting from NeRF-P, followed by APRNet using labeled images, and subsequently incorporating unlabeled images. Additionally, we generate novel views using an online RVS strategy from the synthesized poses within the batch samples, along with real images. The inference time for each image (480×854) is 6.27ms on average. All experiments are conducted on NVIDIA RTX 4090 GPU with PyTorch.

Table 2. Single-frame APR results on 7-scenes dataset. We compared our pipeline with prior single-frame APR methods in median translation (m) and rotation error (°). For better visualization, the best results are highlighted in **bold**.

Category	Methods	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
Single-frame APR	PoseNet(PN) [17]	0.32/8.12	0.47/14.4	0.29/12.0	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.8	0.44/10.4
	BayesianPN [15]	0.37/7.24	0.43/13.7	0.31/12.0	0.48/8.04	0.61/7.08	0.58/7.54	0.48/13.1	0.47/9.81
	PN Learn σ^2 [16]	0.14/4.50	0.27/11.8	0.18/12.1	0.20/5.77	0.25/4.82	0.24/5.52	0.37/10.6	0.24/7.87
	geo. PN [16]	0.13/4.48	0.27/11.3	0.17/13.0	0.19/5.55	0.26/4.75	0.23/5.35	0.35/12.4	0.23/8.12
	LSTM PN [41]	0.24/5.77	0.34/11.9	0.21/13.7	0.30/8.08	0.33/7.00	0.37/8.83	0.40/13.7	0.31/9.85
	Hourglass PN [22]	0.15/6.17	0.27/10.8	0.19/11.6	0.21/8.48	0.25/7.00	0.27/10.2	0.29/12.5	0.23/9.53
	BranchNet [42]	0.18/5.17	0.34/8.99	0.20/14.2	0.30/7.05	0.27/5.10	0.33/7.40	0.38/10.3	0.28/8.30
	GPoseNet [5]	0.20/7.11	0.38/12.3	0.21/13.8	0.28/8.83	0.37/6.94	0.35/8.15	0.37/12.5	0.31/9.95
	MapNet [4]	0.08/3.25	0.27/11.7	0.18/13.3	0.17/5.15	0.22/4.02	0.23/4.93	0.30/12.1	0.21/7.77
	Direct-PN [8]	0.10/3.52	0.27/8.66	0.17/13.1	0.16/5.96	0.19/3.85	0.22/5.13	0.32/10.6	0.20/7.26
	TransPoseNet [34]	0.07/5.68	0.24/10.6	0.13/12.7	0.17/6.34	0.17/5.60	0.19/6.75	0.30/7.02	0.18/7.78
	MSPN [3]	0.09/4.76	0.29/10.5	0.16/13.1	0.16/6.80	0.19/5.50	0.21/6.61	0.31/11.6	0.20/8.41
	MS-Transformer [33]	0.11/4.66	0.24/9.60	0.14/12.2	0.17/5.66	0.18/4.44	0.17/5.94	0.17/5.94	0.18/7.28
	PAE [35]	0.12/4.95	0.24/9.31	0.14/12.5	0.19/5.79	0.18/4.89	0.18/6.19	0.25/8.74	0.19/7.48
	DFNet [7]	0.05/1.88	0.17/6.45	**0.06/3.63**	0.08/2.48	**0.10/2.78**	0.22/5.45	**0.16/3.29**	0.12/3.71
	PMNet(Ours)	**0.04/1.70**	**0.10/4.51**	0.07/4.23	**0.07/1.96**	0.14/3.33	**0.14/3.36**	**0.16**/3.62	**0.10/3.24**
APR with Unlabelled Data.	$\text{MapNet}_{+(seq.)}$ [4]	0.10/3.17	0.20/9.04	0.13/11.1	0.18/5.38	0.19/3.92	0.20/5.01	0.30/13.4	0.19/7.29
	$\text{MapNet}_{+PGO(seq.)}$ [4]	0.09/3.24	0.20/9.29	0.12/8.45	0.19/5.42	0.19/3.96	0.20/4.94	0.27/10.6	0.18/6.55
	Direct-PN+U [8]	0.09/2.77	0.16/4.87	0.10/6.64	0.17/5.04	0.19/3.59	0.19/4.79	0.24/8.52	0.24/8.52
	$\text{LENS}_{+(seq.)}$ [25]	**0.03**/1.30	0.10/3.70	0.07/5.80	0.08/1.90	0.08/2.20	0.09/**2.20**	0.14/3.60	0.08/2.96
	DFNet_{dm} [7]	0.04/1.48	**0.04**/2.16	0.03/1.82	0.07/2.01	0.09/2.26	0.09/2.42	0.14/3.31	0.07/2.21
	PMNet_{ud}(Ours)	**0.03/1.26**	**0.04/1.76**	**0.02/1.68**	**0.06/1.69**	**0.07/1.96**	**0.08**/2.23	**0.11/2.97**	**0.06/1.93**

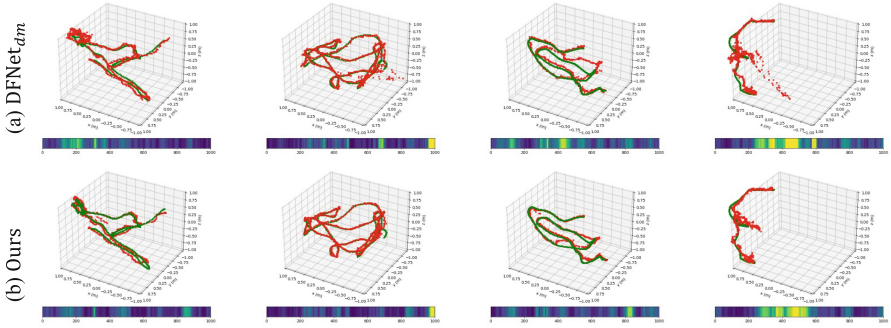


Fig. 4. Visual comparison of camera localization between DFNet$_{dm}$ (top) and our method (bottom) on 7-scenes dataset. For each plot, we show the ground truth camera trajectory in **green** and the estimated trajectory in **red**. The color bar under each plot shows rotation errors. Yellow represents high rotation error, and blue represents low rotation error. Sequence names from left to right are: office-seq7, chess-seq3, fire-seq3 and kitchen-seq4. (Color figure online)

4.2 Evaluation on Datasets

Evaluation on 7-scenes Dataset. First, we compare our methods with one-frame APR methods on 7-scenes dataset. Table 2 shows that our method achieves the best results on all indoor scenes. To be specific, our main pipeline, denoted as PMNet, achieves state-of-the-art performance. Moreover, PMNet$_{ud}$ finetuned in the self-supervising stage with unlabelled data further enhances the pose regression accuracy. The translation and rotation errors are reduced by 14.28% (0.07→0.06) and 12.67%(2.21→1.93) on average, respectively. For the Heads

Table 3. Single-frame APR results on Cambridge Landmarks dataset. We compare with APR-based methods without iterative refinement and report the median position and orientation error in m/°. The best results are highlighted in **bold**.

Category	Methods	Kings	Hospital	Shop	Church	Average
Single-frame APR	PoseNet(PN) [17]	1.66/4.86	2.62/4.90	1.41/7.18	2.45/7.95	2.04/6.23
	BayesianPN [15]	1.74/4.06	2.57/5.14	1.25/7.54	2.11/8.38	1.92/6.28
	PN Learn σ^2 [16]	0.99/1.06	2.17/2.94	1.05/3.97	1.49/3.43	1.43/2.85
	geo. PN [16]	0.88/**1.04**	3.20/3.29	0.88/3.78	1.57/**3.32**	1.63/2.86
	LSTM PN [41]	0.99/3.65	1.51/4.29	1.18/7.44	1.52/6.68	1.30/5.51
	GPoseNet [5]	1.61/2.29	2.62/3.89	1.14/5.73	2.93/6.46	2.08/4.59
	MapNet [4]	1.07/1.89	1.94/3.91	1.49/4.22	2.00/4.53	1.63/3.64
	MSPN [3]	1.73/3.65	2.55/4.05	2.02/7.49	2.67/6.18	2.47/5.34
	MS-Transformer [33]	0.83/1.47	1.81/2.39	0.86/3.07	1.62/3.99	1.28/2.73
	PAE [35]	0.90/1.49	2.07/2.58	0.99/3.88	1.64/4.16	1.40/3.03
	DFNet [7]	0.73/2.37	2.00/2.98	0.67/2.21	1.37/4.03	1.19/2.90
	PMNet(Ours)	**0.68**/1.97	**1.03/1.31**	**0.58/2.10**	**1.33**/3.73	**0.90/2.27**
APR with Unlabelled Data.	$\text{LENS}_{+(seq.)}$ [25]	0.33/**0.50**	**0.44**/0.90	0.27/1.60	0.53/1.60	0.39/1.15
	DFNet_{dm} [7]	0.43/0.87	0.46/0.87	**0.16/0.59**	0.50/1.49	0.39/0.96
	PMNet_{ud}(Ours)	**0.31**/0.55	**0.44/0.79**	0.17/0.86	**0.31/0.96**	**0.31/0.79**
APR + Iterative Refinement.	DFNet [7]+NeFeS_{30} [6]	0.37/0.64	0.98/1.61	0.17/0.60	0.42/1.38	0.48/1.06
	DFNet [7]+NeFeS_{50} [6]	0.37/0.62	0.55/0.90	0.14/0.47	0.32/0.99	0.35/0.75
	CROSSFIRE [24]	0.47/0.70	0.43/0.70	0.20/1.20	0.39/1.40	0.37/1.00
	HR-APR [20]	0.36/0.58	0.53/0.89	0.13/0.51	0.38/1.16	0.35/0.78

and Pumpkin scenes, our results are comparable to those of DFNet, though not superior. Upon reviewing the dataset, we found that these scenes are relatively small, with limited visible range and camera pose variations. Consequently, the benefits of our PoseMap may not be particularly significant.

We visualize the camera localization sequences on 7-scenes dataset in Fig. 4. It shows that our trajectories mostly coincide with the ground truth that means our method could estimate camera positions with high accuracy. Comparing with the SOTA method (i.e., DFNet), our method achieves better estimation results since our trajectories are much closer to the ground truth, and the number of high rotation error zones is less.

Evaluation on Cambridge landmarks dataset. Then we evaluate our methods on the more challenging dataset, i.e., Cambridge Landmarks. As shown in Table 3, our PMNet leads to the dominant performance advantage over all scenes in single-frame APR methods. Compared to DFNet_{dm} [7] which also uses unlabelled data for online training, our method gets performance gain by 20.51% (0.39→0.31) and 17.71% (0.96→0.79) for translations and rotations, respectively.

Besides the overall performance, it's also crucial to achieve high accuracy for frames with sparse distribution, such as images taken far from training views. To evaluate the robustness, we perform spatial clustering on the distribution of camera positions of the training set and calculate the Hausdorff distance from each camera pose of the testing set to the nearest cluster center, which we refer to as the outlier distance. Camera poses of these testing images have the largest outlier distance from the training set. We experimented with 30 images selected from

Table 4. Camera localization results on TOP 30 outliers of camera poses in the testing split on Cambridge Landmarks dataset. We report the mean translation and rotation error in m/°. The better results are highlighted in **bold**.

Methods	Kings	Hospital	Shop	Church	Average
DFNet$_{dm}$ [7]	5.17/3.25	**0.55**/1.11	**0.34**/1.53	1.72/3.42	1.94/2.32
PMNet$_{ud}$(Ours)	**1.61**/**1.65**	0.91/**1.01**	0.37/**1.48**	**0.54**/**1.67**	**0.85**/**1.45**

Table 5. Ablation results on different loss settings in Kings of Cambridge Landmarks. Table 5a shows the exploration of different pose features of PMNet. In Table 5b, we ablate different settings of both PMNet and PMNet$_{ud}$ in median translation (m) and rotation error (°). Note that Exp. 5 and 6 of PMNet$_{ud}$ are finetuned on the pretrained model of Exp. 4 of PMNet.

(a) The ablation of pose features.

Pose Feature	Performance
None	1.03 / 2.94
with $\mathcal{L}_{nerfmap}$	0.95 / 5.76
with $\mathcal{L}_{posemap}$	**0.68 / 1.97**

The ablation study of different loss settings.

Method	Exp. Index	Loss Settings				Performance
		$\mathcal{L}_{pose}$	$\mathcal{L}_{image}$	$\mathcal{L}_{rvs}$	$\mathcal{L}_{posemap}$	
PMNet	1	✓	✓	✓		1.03 / 2.94
	2	✓		✓	✓	0.75 / 3.04
	3	✓	✓		✓	0.94 / 2.34
PMNet$_{ud}$	4	✓	✓	✓	✓	**0.68 / 1.97**
	5		✓			0.39 / 0.63
	6		✓		✓	**0.31 / 0.55**

the testing set in Kings of Cambridge Landmarks. Table 4 shows that PMNet$_{ud}$ obtains lower average errors than DFNet with both position (0.85 vs. 1.94) and orientation (1.45 vs. 2.32). It demonstrates that PoseMap could reinforce the network to learn and explore the less observed views.

4.3 Ablation Studies

Design Choices of PoseMap. Thanks to the neural volumetric pose feature, aka PoseMap, our method achieves more accurate estimation results than previous methods. The underlying strength of PoseMap lies in its ability to establish stable geometric correspondences between the image and the camera pose across diverse scenes. To demonstrate the impact of PoseMap, we break down the pose features into three configurations in Table 5a. The baseline corresponds to the default settings of DFNet [7] without any pose features supervision. $L_{nerfmap}$ refers to using the feature embedding extracted from the last MLP layer of pretrained NeRF-H by directly rendering a feature map (termed as NeRFMap) for APRNet supervision. The comparatively weak results indicate that the original NeRF features do not inherently encapsulate camera pose information. By incorporating a pose branch and a decoder, without modifying the original NeRF training protocol, PoseMap is able to learn the implicit features from the ground truth pose. With the assistance of PoseMap and the $L_{posemap}$ loss.

Effectiveness of the Loss of PoseMap. We conduct ablation studies on each loss component in Exp. 1, 2 and 3 to evaluate the individual contributions.

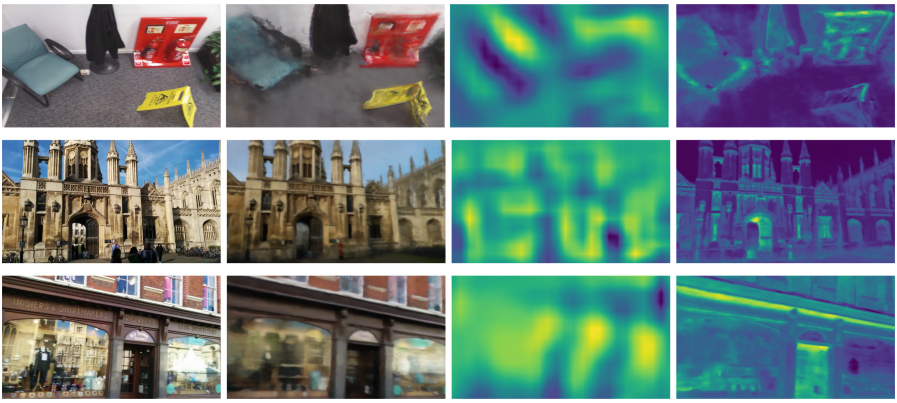


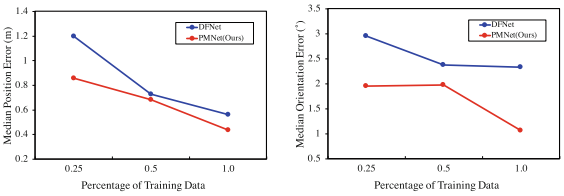
Fig. 5. Visualization of localization results. From left to right, we show the input real image (left), the rendered image of the pose estimated by PMNet$_{ud}$ (2nd column), APR feature map (3rd column), and our PoseMap (right). The dimensionality reduction via PCA is utilized to visualize the PoseMap with pseudo color.

L_{image} acts as an effective feature adapter, bridging the domain gap between real and synthetic image features from our APRNet. It can yield a performance gain of 9.33% in Exp.2 and 4. The RVS loss L_{rvs}, designed for novel view augmentation using a pretrained NeRF, shows an improvement of around 27.66% in Exp.3 and 4. $L_{posemap}$ directs APRNet to learn implicit pose features, enhancing performance by 33.98% in Exp.1 and 4 of PMNet and 20.51% in Exp. 5 and 6. of PMNet$_{ud}$. The largest performance drop without $L_{posemap}$ also indicates the effectiveness of PoseMap. Additionally, for PMNet$_{ud}$, Exp.6 and 7 in Table 5b also highlight the significance of $L_{posemap}$ for feature alignment during the online refinement stage without labels.

Visualization of PoseMap. In Fig. 5, we visualize APR image features and PoseMap features. The APR image features are extracted from the output before the fully-connected layer of VGG16 network. Compared with APR image features, the proposed PoseMap features capture implicit features of the camera pose by aggregating global attributes from samples of light rays, resulting in local features with clearer geometric information than those from 2D-CNN backbones, which is useful on camera localization tasks as proven structure-based methods.

4.4 Discussions

Robustness of Training Size. Since the effectiveness of camera localization correlates with the scale of the observed images [25,31], we perform an ablation experiment on the quantity of training data. We train the whole pipeline with 100%, 50%, and 25% of the training set in Kings of Cambridge Landmarks dataset, the evaluations are shown in Fig. 6. As the data size decreases, the original DFNet [7] exhibits a rapid decrease in accuracy, indicating its high



(a) The translation errors. **(b)** The rotation errors.

Fig. 6. The quantitative results of training data size.

sensitivity to the quantity of input data. By comparison, our results are more robust to the size of training data as well as better performance.

Discussion with Related Methods. As we state in Section 2.1, some post-processing refinement methods, *e.g.*, CROSSFIRE [24] or NeFeS [6], employ iterative pose refinement on reliable camera pose priors with implicit features matching or RANSAC-based PnP. Although it's unfair to directly compare our method with them, we are surprised to see that our method shows competitive performance in Table 3 without any post-processing procedures, which also indicates the superiority of our PoseMap design.

Limitations. Similar to other learning-based methods, PMNet also shares limitations with NeRF and APRNet. Firstly, the accuracy of pose estimation is influenced by the quality of synthetic images, emphasizing the need for robust NeRF models to enhance results. Secondly, current APR-based approaches have yet to fully leverage the geometric structures present in images. Integrating more explicit structure cues, such as 2D lines and 3D depths, could be beneficial. Finally, at the inference stage, the adoption of hierarchical optimization techniques could refine the final results.

5 Conclusion

In this paper, we introduce PoseMap, a novel neural volumetric pose feature designed to enhance camera localization. This feature captures the implicit information of an image and its corresponding camera pose within a neural volumetric field and can be rendered by augmenting the original NeRF with a dedicated pose branch. We have developed a new APR architecture that incorporates a pose features extraction module for the task of camera localization. This architecture allows for online refinement with unlabelled data through self-supervision. Experiments show that our method achieves an average performance improvement of 14.28% and 20.51%, surpassing existing APR methods and establishing a new benchmark of accuracy. However, like other learning-based camera localization techniques, our method still falls short of the accuracy provided by structure-based methods. A potential avenue for future research is to integrate more structure-based features into our pipeline, as the demonstrated benefits of pose features are quite promising.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (62072422).

References

1. Balntas, V., Li, S., Prisacariu, V.: RelocNet: continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 751–767 (2018)
2. Biswas, J., Veloso, M.: Depth camera based indoor mobile robot localization and navigation. In: 2012 IEEE International Conference on Robotics and Automation, pp. 1697–1702. IEEE (2012)
3. Blanton, H., Greenwell, C., Workman, S., Jacobs, N.: Extending absolute pose regression to multiple scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 38–39 (2020)
4. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2616–2625 (2018)
5. Cai, M., Shen, C., Reid, I.: A hybrid probabilistic model for camera relocalization (2019)
6. Chen, S., Bhalgat, Y., Li, X., Bian, J., Li, K., Wang, Z., Prisacariu, V.A.: Refinement for absolute pose regression with neural feature synthesis. ArXiv **abs/2303.10087** (2023). https://api.semanticscholar.org/CorpusID:257622702
7. Chen, Shuai, Li, Xinghui, Wang, Zirui, Prisacariu, Victor A..: DFNet: enhance absolute pose regression with direct feature matching. In: Avidan, Shai, Brostow, Gabriel, Cissé, Moustapha, Farinella, Giovanni Maria, Hassner, Tal (eds.) Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X, pp. 1–17. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20080-9_1
8. Chen, S., Wang, Z., Prisacariu, V.: Direct-PoseNet: absolute pose regression with photometric consistency. In: 2021 International Conference on 3D Vision (3DV), pp. 1175–1185. IEEE (2021)
9. Chitta, K., Prakash, A., Geiger, A.: NEAT: neural attention fields for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15793–15803 (2021)
10. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: CamNet: coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2871–2880 (2019)
11. Ding, Y., et al.: PDF: Point diffusion implicit function for large-scale scene neural representation. In: Thirty-seventh Conference on Neural Information Processing Systems (2023). https://openreview.net/forum?id=k8U8ZijXHh
12. Dong, S., Liu, S., Guo, H., Chen, B.X., Pollefeys, M.: Lazy visual localization via motion averaging. ArXiv **abs/2307.09981** (2023). https://api.semanticscholar.org/CorpusID:259982500
13. Fu, Y., Liu, S., Kulkarni, A., Kautz, J., Efros, A.A., Wang, X.: COLMAP-Free 3D Gaussian splatting. ArXiv **abs/2312.07504** (2023). https://api.semanticscholar.org/CorpusID:266174793
14. Furukawa, Y., Hernández, C., et al.: Multi-view Stereo: a tutorial. Found. Trends® Comput. Graph. Vis. **9**(1-2), 1–148 (2015)

15. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 4762–4769. IEEE (2016)
16. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5974–5983 (2017)
17. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
18. Kundu, A., et al.: Panoptic neural fields: a semantic object-aware neural scene representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12871–12881 (2022)
19. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: BARF: bundle-adjusting neural radiance fields. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 5721–5731 (2021). https://api.semanticscholar.org/CorpusID:233219786
20. Liu, C., Chen, S., Zhao, Y., Huang, H., Prisacariu, V., Braud, T.: HR-APR: APR-agnostic framework with uncertainty estimation and hierarchical refinement for camera relocalisation (2024). https://api.semanticscholar.org/CorpusID:267782842
21. Liu, L., Li, H., Dai, Y.: Efficient global 2D-3D matching for camera localization in a large-scale 3D map. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2372–2381 (2017)
22. Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Image-based localization using hourglass networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 879–886 (2017)
23. Mildenhall, Ben, Srinivasan, Pratul P.., Tancik, Matthew, Barron, Jonathan T.., Ramamoorthi, Ravi, Ng, Ren: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, Andrea, Bischof, Horst, Brox, Thomas, Frahm, Jan-Michael. (eds.) Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I, pp. 405–421. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_24
24. Moreau, A., Piasco, N., Bennehar, M., Tsishkou, D.V., Stanciulescu, B., de La Fortelle, A.: Crossfire: camera relocalization on self-supervised features from an implicit representation. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 252–262 (2023). https://api.semanticscholar.org/CorpusID:257427144
25. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: LENS: localization enhanced by nerf synthesis. In: Conference on Robot Learning, pp. 1347–1356. PMLR (2022)
26. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015). https://doi.org/10.1109/tro.2015.2463671
27. Naseer, T., Burgard, W.: Deep regression for monocular camera-based 6-DoF global localization in outdoor environments. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1525–1530. IEEE (2017)
28. Radwan, N., Valada, A., Burgard, W.: VLocNet++: deep multitask learning for semantic visual localization and odometry. IEEE Robot. Autom. Lett. **3**(4), 4407–4414 (2018)

29. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: robust hierarchical localization at large scale. In: CVPR (2019)
30. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE Trans. Pattern Anal. Mach. Intell. **39**(9), 1744–1756 (2016)
31. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019). https://doi.org/10.1109/cvpr.2019.00342
32. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
33. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2733–2742 (2021)
34. Shavit, Y., Ferens, R., Keller, Y.: Paying attention to activation maps in camera pose regression. arXiv preprint arXiv:2103.11477 (2021)
35. Shavit, Yoli, Keller, Yosi: Camera pose auto-encoders for improving pose regression. In: Avidan, Shai, Brostow, Gabriel, Cissé, Moustapha, Farinella, Giovanni Maria, Hassner, Tal (eds.) Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X, pp. 140–157. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20080-9_9
36. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2930–2937 (2013)
37. Svärm, L., Enqvist, O., Kahl, F., Oskarsson, M.: City-scale localization for cameras with known vertical direction. IEEE Trans. Pattern Anal. Mach. Intell. **39**(7), 1455–1461 (2016)
38. Taira, H., et al.: InLoc: indoor visual localization with dense matching and view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7199–7209 (2018)
39. Toft, C., et al.: Semantic match consistency for long-term visual localization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 383–399 (2018)
40. Valada, A., Radwan, N., Burgard, W.: Deep auxiliary learning for visual localization and odometry. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 6939–6946. IEEE (2018)
41. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using LSTMs for structured feature correlation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 627–637 (2017)
42. Wu, J., Ma, L., Hu, X.: Delving deeper into convolutional neural networks for camera relocalization. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5644–5651. IEEE (2017)