ImPosing: Implicit Pose Encoding for Efficient Visual Localization

Arthur Moreau
MINES ParisTech
Huawei Technologies

Thomas Gilles
MINES ParisTech
Huawei Technologies

Nathan Piasco
Huawei Technologies

Dzmitry Tsishkou
Huawei Technologies

Bogdan Stanciulescu
MINES ParisTech

Arnaud de La Fortelle
MINES ParisTech

Abstract

We propose a novel learning-based formulation for visual localization of vehicles that can operate in real-time in city-scale environments. Visual localization algorithms determine the position and orientation from which an image has been captured, using a set of geo-referenced images or a 3D scene representation. Our new localization paradigm, named Implicit Pose Encoding (ImPosing), embeds images and camera poses into a common latent representation with 2 separate neural networks, such that we can compute a similarity score for each image-pose pair. By evaluating candidates through the latent space in a hierarchical manner, the camera position and orientation are not directly regressed but incrementally refined. Very large environments force competitors to store gigabytes of map data, whereas our method is very compact independently of the reference database size. In this paper, we describe how to effectively optimize our learned modules, how to combine them to achieve real-time localization, and demonstrate results on diverse large scale scenarios that significantly outperform prior work in accuracy and computational efficiency.

1. Introduction

Positioning systems are a necessary component for automated vehicles, mobile robots and augmented reality applications. The precise ego-position inside of a known environment can be recovered in multiple ways using a wide range of sensors. Visual-based localization algorithms [29] predict the 6 degrees of freedom camera pose of a query image, given a set of reference images captured in the environment and labeled with corresponding poses.

We aim to develop relocalization algorithms able to operate efficiently in embedded devices of autonomous vehicles in a deployment scenario where the target area is wide and collected datasets are large. This problem is challenging due to kilometer-scale maps and dynamic outdoor environments. Most accurate visual localization methods [36, 57] first retrieve a coarse localization (i.e. which area is depicted in the image) before computing an accurate camera pose with geometric reasoning by connecting 2D image features to 3D points stored in memory with their corresponding descriptors. The resulting accuracy comes at the cost of a high memory footprint and low latency which increase with the environment size and the reference database. Direct learning-based methods [18, 26, 7] circumvent this limitation by learning the entire task with a single neural network that directly regresses the camera pose from the image. This solution is convenient for embedded deployment : high throughput, low memory footprint and ability to benefit from large amount of data during training without sacrificing test time efficiency. On the other hand, image features extraction and map memorization are entangled in the network's weights, resulting in a limited accuracy [39], slow scene specific training and poor ability to adapt to large environments [7]. Our proposal improves the accuracy and the scalability of direct learning-based methods while preserving the computational efficiency properties.

The common approach to represent scenes in computer vision is to use explicit representations such as point clouds, octrees, voxels or meshes. However, all of them store discrete information, while the underlying signal they represent is inherently continuous. As a consequence, these representations involve a trade-off between resolution and memory consumption. Recently, implicit neural representations [54], that connect scene coordinates to latent codes with a neural network, have shown great success for many computer vision tasks thanks to their ability to model continuous signals embedded into compact network's weights [25].

In this paper, we propose a new direct approach for visual localization in large scenes that perform better than pose regression methods by dissociating image and map encodings, while avoiding the computational cost and memory footprint of structure-based methods thanks to an implicit map representation. The core idea is to connect image and camera pose representations, which are learned separately by two distinct neural networks, in a common latent space. We use an implicit neural representation to encode a specific viewpoint in the scene (i.e. a 6-DoF camera pose) into a higher dimensional vector. With this formulation, the continuous representation of any camera pose in the scene (even a pose not observed in reference images) can be computed in a single network forward pass. We take advantage of this property to solve the localization task by searching the poses candidates which are the most similar to the learned image representation. To do so, we introduce a hierarchical sampling process able to retrieve the correct camera viewpoint using only a few batched queries on the pose encoder network. Our localization method, called Implicit Pose Encoding (ImPosing), provides real-time sub-metric localization performances that can be rapidly deployed on large areas.

We evaluate our system on a wide range of visual localization datasets, including several kilometers-scale road environments with challenging conditions (seasonal and appearance changes, limited training data). We observe that our method outperforms its regression-based competitors in terms of accuracy and training efficiency, especially in large-scale scenarios.

2. Related work

Image-based localization. Camera localization from RGB images for real-time application can be tackled by different classes of prior methods discussed below:

Absolute pose regression addresses the problem through end-to-end supervised regression between the input image and the camera pose using deep neural networks. PoseNet [18] is the pioneering work, and uses an encoder-decoder architecture where the encoder is a CNN pretrained on ImageNet and the decoder regresses the pose with fully connected layers. Since then, many architectural improvements have been proposed: notably, VidLoc [10] incorporates spatio-temporal constraints using consecutive video frames, AtLoc [52] uses an attention-based module before the decoding step, Xue et al. [55] model the problem with graph neural networks, TransPoseNet [42] with transformers, and CoordiNet [26] uses a fully-convolutional architecture with geometrical inductive biases in the decoder layers. The main advantages of this class of methods are the compatibility with real-time deployment thanks to fast inference, low memory requirements and uncertainty estimation [17, 26] which enables to filter out failure cases. The localization accuracy exhibited by absolute pose regression is limited compared to other methods [39], but has been observed to be highly dependent on the quantity and diversity of available training images, which can be improved with novel view synthesis [27]. ImPosing does not not explicitly regress the pose of the camera but learns a latent representation which connects the query image to an implicit map. In the following, we show through experiments that this formulation is better suited than absolute pose regression for localization in large urban area.

Scene coordinate regression learns the correspondence between the 2D image features and 3D scenes coordinates of observable image patches. It enables to retrieve the camera pose using projective geometry, by solving the Perspective-N-Points problem robustly with RANSAC [14]. Seminal work on scene coordinate regression rely on RGB-D images and use random forest to store the 3D coordinates [43]. Since then, the scene coordinate regression pipeline has been adapted to RGB images processed by fully convolutional networks [24, 6]. The RANSAC step has been replaced by its differentiable counterpart DSAC [5], and ESAC [7] uses mixtures of expert to improve scaling to large environments. This class of methods exhibit higher accuracy than absolute pose regression and the efficiency enables real-time computation, however these methods are limited to relatively small environments [8]. By considering global image description instead of local features extraction, ImPosing is able to scale up to larger scenes at the cost of minor loss in localization performances.

Image retrieval algorithms for localization solve a slightly different task: instead of computing a pose for the query image, these methods retrieve the closest georeferenced image from the query within a large database [1, 15, 32, 30]. The top ranked images are used to define a coarse localization of the query image. Poses averaging [50] or specific re-ranking based on GPS information [38] are used to improve the localization accuracy. Image retrieval methods use global image descriptors obtained by features maps pooling [1] or dense local features extraction [49] to represent the discriminating content of the image. Nearest neighbour search in the descriptor space associates the query to the most similar examples in the database. These methods naturally scale to very large environments [40] but their accuracy is bounded by the density and diversity of reference images in the scene. Such a large database is difficult to collect and enlarging it linearly increases the memory footprint and the nearest neighbour search computational cost. This property make image retrieval an appealing solution for visual place recognition but not convenient for camera pose estimation. Our method share similarities with image retrieval: a global image descriptor is matched against the map. In our case, the georeferenced image batabase is replaced by an implicit map

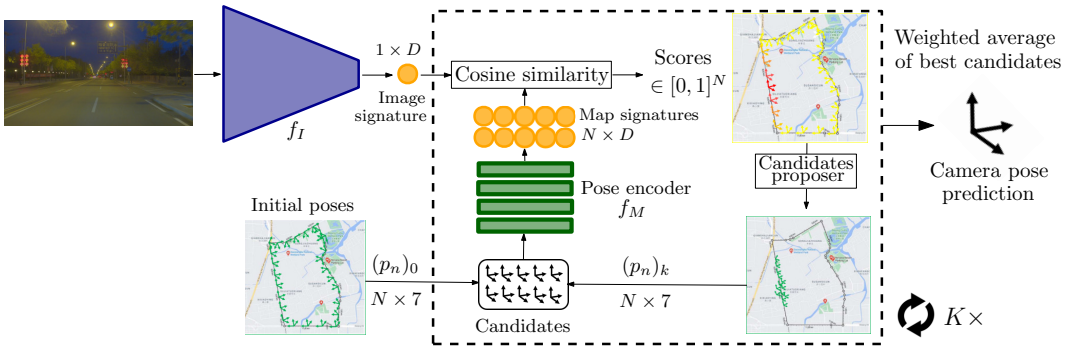

Figure 1: **Implicit pose encoding for hierarchical image localization.** A set of initial map signatures is compared to the image signature to determine the most probable localization of the camera. The similarity scores guides the selection of a new batch of pose candidates that are used to compute the new map signatures for the second refined localization step. This process is repeated multiple time to predict the final camera pose.

representation. As a result, we can compute the descriptor of any camera pose in the map instead of being limited a finite set of reference images. Morever, for a given scene, larger datasets improve the resolution of the map representation without increasing the memory footprint of our map, stored as network weights.

Structure-based methods compare local 2D image features to a 3D model to estimate the camera pose. 2D features are extracted from the query image using a CNN such as SuperPoint [11], and matched against the 3D model [35] to establish robust 2D-3D correspondences, that enable to compute the pose with PNP + RANSAC [34] or by Levenberg-Marquadt optimization [51, 36]. The 3D model, usually represented as a point cloud of descriptors, enables to use geometric reasoning to solve the task. However, in large dynamic environments, highly accurate 3D reconstructions are challenging to make and memory demanding. Relative pose regression [19, 3, 12] from nearest images can alternatively be used to predict the pose, but the storage requirement is even bigger. ImPosing does not rely on a 3D model of the scene and operate only with images and references poses.

Implicit representations. Neural networks performances highly depend on the representation used for a given space. Recent research has shown that using fully-connected neural networks to represent 3D data offers many benefits: the representation is continuous, memory-efficient and convenient to learn in any differentiable pipeline [54]. Successful examples of neural representations include 3D shapes [28, 2], sound [44], neural rendering of static [25, 45, 16] and dynamic scenes [23] or real-time RGB-D SLAM [47].

In this paper, we aim to learn an effective representation of the map for camera relocalization inside of a given scene. The map is given as a set of images with 6D camera poses: a 3D translation vector and a 3D rotation represented by quaternions, euler angles, axis-angle or rotation matrix. Zhou et al. [56] have demonstrated that none of these rotation representations are continuous, in the sense of continuously mapping coordinates to a latent space produced by a neural network, which is precisely our problem of interest. Zhu et al. [58] proposed a learned camera pose representation which is beneficial for view synthesis and pose regression. We propose to use a related camera pose representation optimized to be directly matched against the input image representation, enabling pose estimation by iterative sampling and evaluation of pose candidates.

3. Method

Our method, ImPosing, estimates the 6-DoF camera pose $(t, q) \in SE(3)$ of a query image I, where t is a translation vector and q is a unit quaternion. We train our solution using a reference dataset of posed images (I_k) collected in the target area and we do not make use of an additional 3D model of the scene.

The proposed algorithm, presented in figure 1, computes a vector that represents the image through the image encoder. Then, the camera pose is searched by evaluating initial pose candidates distributed across the map. Poses are processed by the pose encoder to produce a latent representation that can be matched against the image vector. Each pose candidate receives a score, based on distance to camera pose. High scores provide a coarse localization prior which is used to select new candidates. By repeating this process several times, our pool of candidates converges to the actual camera pose.

3.1. ImPosing localization process

This section describes the localization process step by step from the image to the final camera pose estimate, displayed in figure 1,

1. Image encoder: we compute a global image features vector $f_I(I) \in \mathbb{R}^d$ from the input query I using our image encoder. The encoder architecture consists in a pretrained CNN backbone followed by a Global Average Pooling, and a fully-connected layer with d output neurons. The feature vector is one order of magnitude smaller than global image descriptors commonly used in image retrieval (we use $d = 256$ whereas Revaud et al. [32] use $d = 2048$) in order to efficiently compare it to a large set of pose candidates at later steps.

2. Initial pose candidates: Our starting point is a set of N camera poses $(p_n)_0$, sampled from the set of reference poses (= training poses). Through this initial selection, we introduce a prior for the localization process, similar to the anchors poses in [33] or regression methods that compute relative instead of absolute pose [13]. We observed that the algorithm is robust to this choice: a 2D grid on the map yield similar results.

3. Pose encoder: Pose candidates are processed by a neural network which outputs latent vectors. This implicit representation learns the correspondence between camera viewpoints in a given scene and features vectors provided by the image encoder. First, following Tancik et al. [48], each component of the camera pose $(tx, ty, tz, qx, qy, qz, qw)$ is projected to higher dimension using Fourier features : $x \rightarrow (x, sin(2kx), cos(2kx))_{0 \leq k \leq 10}$, as it helps networks with low dimensional input to fit high frequency functions. Then, we use a MLP f_M with 4 layers of 256 neurons and ReLU activations on hidden layers. Each set of pose candidates is computed in a single batched forward pass.

4. Similarity scores: we obtain a similarity score s by computing the cosine similarity between $f_I(I)$ and $f_M(p)$ for each image-pose pair (I, p). We add a ReLU layer after the dot product, such that $s \in [0, 1]$. Intuitively, we aim to learn high scores for poses candidates close to the actual camera pose. With this formulation, we can evaluate hypotheses on the camera pose and search for pose candidates with high scores. Formally, our score is defined by:

$$s(I, p) = \frac{\langle f_I(I), f_M(p) \rangle}{\|f_I(I)\| \|f_M(p)\|} \mathbb{1}_{\langle f_I(I), f_M(p) \rangle > 0} \qquad (1)$$

5. Candidates proposer: new poses $(p_n)_k$ are selected for the k^{th} iteration based on scores obtained with poses $(p_n)_{k-1}$ at the previous iteration. First, we select the poses with top $B = 100$ higher scores $(h_i)_{0 \leq i < B} \subset (p_n)_{k-1}$. Then, new candidates are sampled from (h_i) in a Gaussian Mixture Model with density:

$$P(x) = \sum_{i=1}^{100} \pi_i \mathcal{N}(x | h_i, v/k) \quad \text{where} \quad \pi_i = \frac{s(I, h_i)}{\sum_{l=1}^{100} s(I, h_l)}. \qquad (2)$$

$v = [v_{tx}, v_{ty}, v_{tz}, v_{rx}, v_{ry}, v_{rz}]$ is the variance of the sampling process, a hyperparameter composed of a translation vector and Euler angles.

6. Iterative pose refinement: we repeat K times the evaluation of pose candidates described in steps 3-4-5. After each iteration, the noise vector v is divided by 2, such that new candidates are sampled closer to previous high scores. As a result, we can converge to a precise pose estimate in kilometers-scale maps while only evaluating a limited sparse set of poses. We evaluate each camera frame independently at each time step, however one could use localization priors from previous time steps to reduce the number of iterations in vehicles navigation scenarios. An example of selected poses at each iteration is shown in Fig. 2. By sampling N candidates for initial poses, we preserve a constant memory peak.

7. Pose averaging: our final camera pose estimate is a weighted average of the 256 pose candidates with higher scores, which exhibits better interpolation properties than selecting the best score pose. We use scores as weighting coefficients and 3D rotation averaging is implemented following [22].

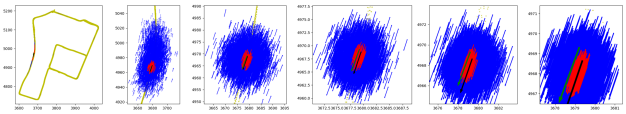

Figure 2: **Iterative candidates refinement.** At each k step of the localization process, top scored poses are selected to sample the new candidate poses at step $k + 1$. From left to right: top scored poses at $k = 0$ to $k = 5$, yellow points are positions of the training example, blue arrows are pose candidates and red arrows are the selected poses among the candidates.

The entire inference procedure requires 1 forward pass on the image encoder and K passes on the pose encoder.

3.2. Training procedure

We do not train the system by minimizing the error on the final camera pose estimate. Instead, we apply our loss function directly on the predicted scores. As a result, one training iteration provides supervision on the $K \times N$ image-pose pairs that contains more information than the single localization error. We observed that this property results in

superior training efficiency than regression approaches (see 4.4). We define target scores s_t based on translation and rotation distances between the camera pose $p_I = (t_I, q_I)$ and the candidate pose $p = (t, q)$:

$$s_t(I, p) = ReLU\left(1 - \lambda_t \|t_i - t\|_2 - \lambda_r G(q_i, q)\right) \quad (3)$$

where λ_t and λ_r are weighting parameters set to 5 and 0.1 and G is geodesic distance, defined as the minimal angle between 2 rotations:

$$G(q_1, q_2) = cos^{-1}\left(\frac{(tr(M_{q_1} M_{q_2}^{-1}) - 1)}{2}\right), \quad (4)$$

M_q being the 3D rotation matrix associated with rotation q.

We train f_I and f_M by computing scores between reference images and pose candidates sampled at K different resolutions as described in section 3.1. For training purpose, we add to initial poses an uniform noise sampled in $[-v, v]$ as we observed that it reduces overfitting. We also use poses associated with the top target scores in the candidates proposer, in addition with top predicted scores in order to guide training convergence in early iterations.

Finally, our optimization objective is:

$$L = \frac{1}{N} \sum_{k=0}^{K} \sum_{n=0}^{N-1} |s(I, p_{n,k}) - s_t(I, p_{n,k})| \quad (5)$$

An analogy can be made with content-based image retrieval [1, 32]: global descriptors are usually trained using image triplets composed of a query image, a positive and a negative example. Positive samples are data close to the query, in metric or semantic domain depending on the final application, and negative samples are images with unrelated content to the query. Global descriptors can be trained by minimizing a triplet margin loss [1]. In our case, positive examples are the poses with a non-zero score whereas negative examples are candidates farther from the camera pose than an arbitrary threshold. Instead of binary classification (positive or negative example), we rank the relative importance of the positive samples according to their distance to the ground truth label.

4. Experiments

We compare our approach against recent methods on several datasets covering a wide range of autonomous driving scenarios in large scale outdoor maps. This task is highly challenging due to the dynamic part of outdoor environments (moving objects, illumination, occlusions, etc.). We verify that our formulation enables accurate localization in 9 different large outdoor scenes. Then we show that our method can be naturally extended to multi-map scenarios and we report results using this setup. We also compare the computational efficiency of our method with competitors and finally present an ablation study on hyperparameters of ImPosing. Video displaying trajectories is included in the supplementary material.

Implementation details: ImPosing is implemented in PyTorch. Images are computed at a small resolution 135×240. The image encoder uses a ResNet34 backbone pretrained on ImageNet. $N = 4096$ pose candidates are evaluated at each of the $K = 6$ refinement steps. For candidates sampling, the noise vector is set to $v = [8.0m, 0.2m, 8.0m, 1°, 5°, 1°]$ where y is the altitude axis, and we use 100 GMM components. We train the image encoder and pose encoder for 250 epochs with Adam optimizer at a constant learning rate of $1e^{-4}$. We did not tune these parameters specifically for each scene, suggesting that they should work for any autonomous driving scene. More details are provided in supplementary materials, including datasets configuration.

Baselines: Our first aim is to compare ImPosing to its direct learning-based methods competitors. We use CoordiNet [26] that report state-of-the-art results for absolute pose regression on Oxford Dataset as a baseline. We report previously published results on this dataset, and our own implementation for other datasets. We replace the EfficientNet backbone by ResNet34 for a fair comparison with ImPosing. As outlined in section 2, we share similarities with image retrieval by matching a global descriptor against the map. To compare ImPosing to retrieval, we use NetVLAD [1] (VGG16 backbone) and Revaud et al. [32] (GeM pooling, Resnet101 backbone) publicly available implementations[1]. Full sized images are used to compute global image descriptors followed by cosine similarity for features comparison, then we perform pose averaging on poses of top 20 database images as in [39]. Scene coordinate regression [7, 8] can not scale to large environments thus is not considered for evaluation. We did not conduct experiments with structure-based methods [34, 36, 57]. These methods are more accurate than ours thanks to geometric reasoning with a 3D model, but also operate at a different computation scale than ours (see figure 3) making embedded deployment difficult. In scenarios where it can be afforded, ImPosing can be considered as a coarse localization step, followed by refinement with a 3D model, similar to HLoc[34] architecture.

4.1. Single scene localization

Oxford RobotCar [21] contains images recorded by a vehicle in Oxford over a year. We reproduce experiments

[1] https://github.com/Nanne/pytorch-NetVlad and https://github.com/naver/deep-image-retrieval

Dataset		Pose regression		Image retrieval		ImPosing
		CoordiNet [26]	AtLoc [52]	NetVLAD [1]	GeM [32]	
Oxford Full	Median	3.55m/1.1°	11.1m/5.3°	1.42m/1.4°	**1.36m/1.3**°	1.90m/1.3°
	Mean	14.96m/5.7°	29.6m/12.4°	4.47m/2.4°	**3.49m/2.3**°	4.25m/4.3°
Oxford Loop	Median	2.27m/0.9°	5.36m/2.1°	2.16m/1.1°	2.39m/1.0°	**1.93m/1.0**°
	Mean	4.15m/1.4°	8.73m/4.6°	4.16m/1.9°	6.92m/3.1°	**3.03m/1.8**°
Average	Median	2.91m/1.0°	8.23m/3.7°	**1.79m**/1.2°	1.88m/**1.1**°	1.92m/**1.1**°
	Mean	9.56m/3.4°	19.17m/8.5°	4.32m/**2.1**°	5.20m/2.7°	**3.64m**/3.0°
Daoxiang Lake	Median	6.82m/0.4°	–	8.92m/0.8°	27.13m/1.1°	**1.62m/0.3**°
	Mean	25.18m/1.0°	–	152.2m/15.5°	328.8m/19.5°	**8.40m/0.5**°

Table 1: Localization error on Oxford RobotCar and Daoxiang Lake datasets.

commonly reported for learning-based methods [26, 52, 55]: we evaluate on the *Loop* and *Full* scenes, using only 2 sequences for training. Results are reported in Table 1.

First we observe that image retrieval performs better than pose regression. Previous learning-based methods struggle due to the low-data regime [26, 27] and the decrease of the regression accuracy in large maps. Oxford city is an environment with rich features similar to visual place recognition training datasets, that make NetVLAD [1] and GeM [32] strong baselines in this scenario. ImPosing exhibits state-of-the art accuracy on Oxford Loop scene, as well as the best mean error in average. These results are obtained by reducing a lot the number of large failure cases that occur with prior methods.

We also observe that despite newly provided RTK ground truth provided by the authors [20], the reference poses are largely inaccurate in some areas. As a result, evaluation metrics are not significant at a centimeter level and models training might be impacted by this erroneous pose labels. For this reason, we conduct a benchmark on two recently released datasets with more reliable ground-truth.

Daoxiang Lake [57] has been collected in a 12km loop in Beijing during 4 months. 8 recordings are available, we use 7 for training and 1 for testing with images from the front camera only. This scene contains the largest map and training dataset of our experiments. Median and mean errors are shown in Table 1. Daoxiang Lake is a more challenging dataset than Oxford because of repetitive areas with few discriminative features and various environments (urban, peri-urban, highways, nature, etc.). As a result, image retrieval performs worse than pose regression. ImPosing is way more accurate and exhibits a median error 4 times smaller than competitors.

4 seasons [53] contains data recorded in Munich area in various scenes (city, residential neighborhoods, countrysides) with varying seasonal conditions. We selected 6 scenes where at least 3 different recordings are provided: we use 1 for testing and others as training images. This benchmark is highly challenging due to extreme appearance changes between sequences, small data regime for some scenes, featureless environments (see illustration in supplementary materials) and kilometers-scale maps. Results are reported in table 2.

First, absolute localization accuracy is very heterogeneous between different scenes. We note that scenes with few training images are the most challenging. In particular, *Countryside* include navigation around fields and *City Loop* is a 10km map where the training dataset is composed of a winter sequence with snow and a rainy sequence with blur on camera lens. In these extreme cases, both pose regression and image retrieval fail to estimate reliable poses, whereas ImPosing is able to provide a coarse localization. With sufficiently large training datasets, our method still exhibits the more precise pose estimation.

4.2. Multi-scene localization

Learning-based methods for relocalization require scene specific training, inducing heavy computation for potential deployment in several areas at a large scale. Recent work [4, 41] has extended absolute pose regression to multi-scene scenarios. The core idea is to train a system with images from several maps while sharing image encoder parameters that could learn to extract features in a generic way. As our method separate image and map representation, ImPosing naturally extends to multi scenes scenarios. To adapt ImPosing to a multi-map scenario, we perform the following modifications: the image encoder backbone is shared between all maps, whereas one specific pose encoder is learned for each scene. We also learn scene specific parameters for the final linear layer of the image encoder, to facilitate image features projection to the desired map representation. We train a multi-scene model on the 6 maps of 4 seasons [53]. Results are reported in Table 2. The model has been trained for 20 epochs only because of computational constraints, but still outperform all competitors except single scenes ImPosing models. While the convergence for a single scene is slower in the multimap formulation (but training a multiscene on n maps is faster than performing n different trainings on each map, see supple-

	Dataset details			Image retrieval		CoordiNet [26]	ImPosing	
	Road length	Runs	Images	NetVLAD [1]	GeM [32]		Single sc.	Multi sc.
Neighborhood	2000	6	16520	0.72m/0.9°	0.69m/0.9°	0.74m/**0.6°**	**0.53m**/0.7°	0.82m/1.0°
Office loop	2600	5	20915	6.85m/3.0°	6.39m/2.8°	6.25m/1.5°	**0.99m/1.1°**	1.58m/1.3°
Countryside	6200	3	19804	32.24m/1.2°	30.87m/1.3°	47.33m/2.9°	**2.61m/0.9°**	5.46m/1.1°
Bus. campus	1000	2	6132	1.19m/1.3°	1.96m/**1.2°**	22.57m/6.0°	**1.16m**/1.3°	1.70m/1.6°
City loop	10000	2	17427	61.60m/3.5°	317.4m/6.9°	584.4m/14.4°	**5.32m/2.4°**	10.53m/2.5°
Old Town	4500	3	13959	3.45m/**1.2°**	4.46m/1.6°	50.83m/3.8°	**2.59m/1.2°**	3.71m/1.3°
Average	-	-	-	17.67m/1,8°	60.30m/2,4°	118.7m/4.9°	**2.2m/1.3°**	3.97m/1.5°

Table 2: Median localization error on 4Seasons dataset.

mentary materials), it enables to localize in huge areas with minimal memory storage requirements (see section 4.3).

4.3. Efficiency comparison

Storage footprint. Our method only needs to store neural networks weights and initial pose candidates in device. It represents 23MB for the image encoder, less than 1MB for the pose encoder and 1MB for the initial poses candidates. We also report in figure 3 the scaling law of memory footprint w.r.t. reference database size for different classes of visual localization methods. This is an important aspect in autonomous driving scenarios where large amounts of data are available. For a given map, learning-based methods have a constant memory requirement because the map information is embedded in the networks weights. To estimate storage requirement of retrieval methods, we consider the size of the database image descriptor (2048 for GeM and 4096 for NetVLAD) along with the size of the image encoder. Storage requirement of retrieval methods exceed 1 GB for large scale scene with more than 100k reference images. To estimate the memory requirement of structure-based methods we consider the numbers given in [34]: a 3D model built from 4328 images is composed of 685k 3D points. If we consider one local descriptor of size 128 by 3D points, we can derive a linear rule to determine the 3D model size according to the number of reference images. This is a rough estimation but we can estimate that structure based method require at least 3 times more storage capacity than image retrieval methods. Compressing techniques exist to make these methods more tractable [37, 9], however compressed maps still represent gigabytes and are less accurate.

Algorithms	**In device storage**	**Scalability**	**Latency**	**Accuracy**
IR+2D-3D matching	3D model + IR DB + NN (5-100GBs)	High	Low	High
IR+Relative PR	IR DB with images + NN (5-100GBs)	High	Low	Medium
IR	IR DB + NN (2-50GBs)	High	Medium	Low
APR	NN (≈ 25MB)	Medium	High	Low
SCR	NN (≈ 25MB)	Low	High	High
ImPosing (ours)	NN (25MB)	High	High	Medium

Table 3: **Qualitative comparison between methods.** We compare the properties of visual localization class of methods w.r.t. storage requirement, capability to operate in large maps (scalability), latency and accuracy. *IR* stands for Image Retrieval, *PR* for Pose Regression, *SCR* for Scene Coordinate Regression, *DB* for database and *NN* for neural networks weights. Storage of IR databases are detailed in [46].

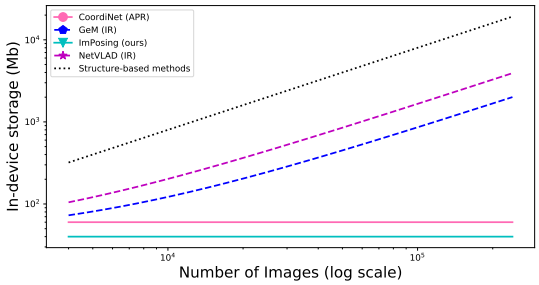

Figure 3: **In-device memory usage.** Structure-based methods (black) and image retrieval (blue and purple) use more memory when the reference dataset grows whereas pose regression methods and ImPosing (pink and cyan) storage requirement does not depend on dataset size.

Computational complexity. Our algorithm complexity depends on the image encoder backbone (3.6 billion FLOPs for ResNet34) and the hierarchical decoding process with the pose encoder. With the default hyperparameters, it involves 4.8 billion FLOPs. We measured a total inference time of 41ms for a single image using a NVIDIA RTX 2080 GPU. The complexity is linear w.r.t. the number of refinements K, the number of pose candidates N and the number of layers in the MLP. It is quadratic w.r.t. the latent dimension D. It should be noted that parallel computations reduce the impact of N and D on the inference time. Considering these properties and the ablations provided in 4.4, one can choose the corresponding hyperparameters that match its computational requirements.

Summary. ImPosing exhibits very compact storage requirements and fast inference time coupled with state-of-the-art accuracy. Notably, neither memory footprint and computational complexity depends on the number of images in the reference database, which is a great advantage over image retrieval methods [1, 32]. We also observe empirically that our method converges approximately 2 times

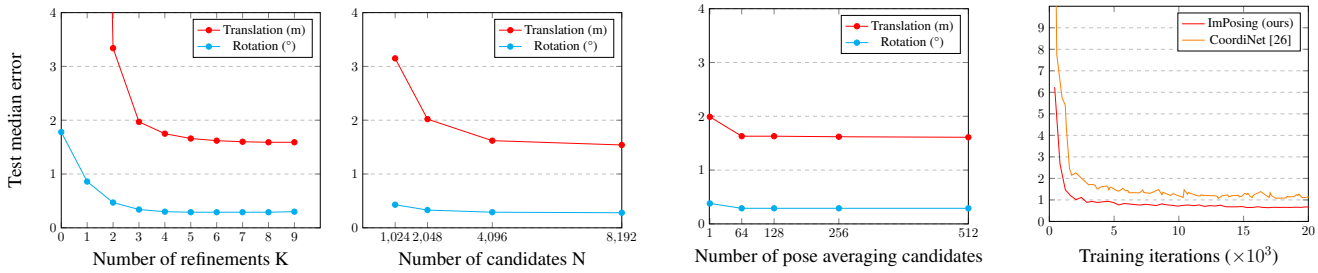

Figure 4: **From left to right:** median localization errors depending on number of refinements, pose candidates, and final number averaged poses. Training time comparison between pose regression [26] and ImPosing.

faster than pose regression competitors [26] w.r.t. the number of training iterations (see figure 4).

4.4. Ablation study

We report the influence of several hyperparameters on the localization accuracy of ImPosing in figure 4. We evaluate the number of refinement steps K, the number of pose candidates N and the number of best candidates used for pose averaging. We use the model trained on Daoxiang Lake and change the parameters at test time. Increasing the number of refinements and candidates improves localization accuracy, at the cost of a higher computational cost. We use a reasonable trade-off with $K = 6$ and $N = 4096$ as our default setup. We observe that pose averaging has a positive impact on accuracy, but the number of selected candidates is not critical. Additional ablation studies on number of layers in the pose encoder and computation of the similarity score are provided in supplementary materials.

5. Discussion

What does the pose encoder learn? In the pose regression approach, image and camera pose are connected by being the respective input and output of a single feedforward neural network. This formulation entangles features extraction, map memorization and camera pose prediction in a single model. While deep neural networks are known to perform well for the first, they have been observed to be inaccurate for pose prediction [39]. Our solution circumvents this problem by "inverting" the decoder layers with the pose encoder. We don't try to predict the pose from features but to connect a given pose to its respective latent features. We let the network learn the optimal latent space to connect images and camera poses, with a single constraint: pose candidates close to the actual camera pose must have a vector relatively similar to the image representation. This property enables to search the best pose candidates in a coarse to fine manner, and interpret the resulting scores has a multimodal distribution of positions across the map. We provide visualizations of these distributions and of the latent space structure in supplementary materials.

Benefits, limitations and future work. Our method keeps the main advantages of direct learning-based methods: we obtain the pose efficiently with neural networks inference, we do not use a 3D model of the scene or a retrieval database, resulting in a very compact memory footprint. We observe that the accuracy our method highly depends on the quantity of training data available. Similar to regression, our method does not extrapolate to camera positions far from trainings examples. However, recent approaches has shown that these limitations can be overcome with synthetic datasets [27]. Moreover, in the driving scenario, a coarse localization estimate can be sufficient because horizontal localization (road lane) can be recovered thanks to perception [31]. The new paradigm we propose could be improved in many ways. It includes exploring better architectures for the pose encoder, inspired from recent work on coordinate-based representations [58]. Another interesting direction is to extend the implicit map representation to local features instead of global image signatures, by finding a way to represent implicitly a 3D model.

6. Conclusion

We have proposed a new formulation for visual localization that perform state-of-the art accuracy for direct learning-based methods in large environments. By using an implicit representation of the map, we connect camera poses and image features in a latent high dimensional manifold well suited for localization. We have shown that with a simple pose candidates sampling procedure, we are able to estimate the absolute pose of an image. Our proposal can be directly applied in autonomous driving systems, by providing an efficient and accurate image-based localization algorithm that can operate at large scales in real-time. We believe that, beyond our work, implicit scene representations, by their ability to model complex continuous signals in a fixed size neural network, are a promising research direction for camera pose estimation.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[4] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–39, 2020.

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6684–6692, 2017.

[6] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019.

[8] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using dsac. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[9] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7645–7654, 2019.

[10] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6856–6864, 2017.

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

[12] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera relocalization. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2871–2880, 2019.

[13] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera relocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2871–2880, 2019.

[14] M. Fischler and R. 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.

[15] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017.

[16] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, page 4762–4769. IEEE Press, 2016.

[18] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[19] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[20] Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time kinematic ground truth for the oxford robotcar dataset. 2020.

[21] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. volume 36, pages 3–15, 2017.

[22] F Landis Markley, Yang Cheng, John L Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[23] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[24] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip H.S. Torr. Random forests versus neural networks — what's best for camera localization? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5118–5125, 2017.

[25] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[26] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2229–2238, 2022.

[27] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1347–1356. PMLR, 2022.

[28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109, 2018.

[30] Nathan Piasco, Désiré Sidibé, Valérie Gouet-Brunet, and Cédric Demonceaux. Learning scene geometry for visual localization in challenging conditions. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9094–9100, 2019.

[31] Tong Qin, Yuxin Zheng, Tongqing Chen, Yilun Chen, and Qing Su. A light-weight semantic map for visual localization towards autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11248–11254, 2021.

[32] J. Revaud, J. Almazan, R.S. Rezende, and C.R. de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019.

[33] Soham Saha, Girish Varma, and CV Jawahar. Improved visual relocalization by discovering anchor points. *arXiv preprint arXiv:1811.04370*, 2018.

[34] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

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

[36] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021.

[37] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2102–2110, 2015.

[38] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1582–1590, 2016.

[39] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-taixé. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *CVPR*, pages 3302–3312, 2019.

[40] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR07)*, Minneapolis, June 2007.

[41] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021.

[42] Yoli Shavit, Ron Ferens, and Yosi Keller. Paying attention to activation maps in camera pose regression. In *arxiv preprint, arxiv:2103.11477*, 2021.

[43] 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 *Proc. Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2013.

[44] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[45] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019.

[46] Yuxin Song, Ruolin Zhu, Min Yang, and Dongliang He. Dalg: Deep attentive local and global modeling for image retrieval. *arXiv preprint arXiv:2207.00287*, 2022.

[47] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[48] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[49] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015.

[50] Akihiko Torii, Josef Sivic, and Tomas Pajdla. Visual localization by linear combination of image descriptors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 102–109. IEEE, 2011.

[51] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. Gn-net: The gauss-newton loss for multi-weather relocalization. *IEEE Robotics and Automation Letters*, 5(2):890–897, 2020.

[52] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10393–10401, 2020.

[53] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2020.

[54] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond, 2021.

[55] F. Xue, X. Wu, S. Cai, and J. Wang. Learning multi-view camera relocalization with graph neural networks. In

2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11372–11381, 2020.

[56] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[57] Yao Zhou, Guowei Wan, Shenhua Hou, Li Yu, Gang Wang, Xiaofei Rui, and Shiyu Song. Da4ad: End-to-end deep attention-based visual localization for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[58] Y. Zhu, R. Gao, S. Huang, S. Zhu, and Y. Wu. Learning neural representation of camera pose with matrix representation of pose shift via view synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9954–9963, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.