HAL
open science

Back to the Feature: Learning Robust Camera Localization from Pixels to Pose

Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Lars Hammarstrand, Vincent Lepetit, Fredrik Kahl, et al.

To cite this version:

Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, et al.. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. Conference on Computer Vision and Pattern Recognition, 2021, Online, United States. hal-03482290

HAL Id: hal-03482290

https://hal.science/hal-03482290v1

Submitted on 15 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:2103.09213v2 [cs.CV] 7 Apr 2021

Back to the Feature: Learning Robust Camera Localization from Pixels to Pose

Paul-Edouard Sarlin[1*] Ajaykumar Unagar[2*] Måns Larsson[3,4] Hugo Germain[5] Carl Toft[3]
Viktor Larsson[1] Marc Pollefeys[1,6] Vincent Lepetit[5] Lars Hammarstrand[3] Fredrik Kahl[3] Torsten Sattler[3,7]

[1] Department of Computer Science, ETH Zurich [2] ETH Zurich [3] Chalmers University of Technology
[4] Eigenvision [5] Ecole des Ponts [6] Microsoft [7] Czech Technical University in Prague

Abstract

Camera pose estimation in known scenes is a 3D geometry task recently tackled by multiple learning algorithms. Many regress precise geometric quantities, like poses or 3D points, from an input image. This either fails to generalize to new viewpoints or ties the model parameters to a specific scene. In this paper, we go Back to the Feature: we argue that deep networks should focus on learning robust and invariant visual features, while the geometric estimation should be left to principled algorithms. We introduce PixLoc, a scene-agnostic neural network that estimates an accurate 6-DoF pose from an image and a 3D model. Our approach is based on the direct alignment of multiscale deep features, casting camera localization as metric learning. PixLoc learns strong data priors by end-to-end training from pixels to pose and exhibits exceptional generalization to new scenes by separating model parameters and scene geometry. The system can localize in large environments given coarse pose priors but also improve the accuracy of sparse feature matching by jointly refining keypoints and poses with little overhead. The code will be publicly available at `github.com/cvg/pixloc`.

1. Introduction

Visual localization is the problem of estimating the camera position and orientation for a given image in a known scene. Solving this problem is a key step towards truly autonomous robots such as self-driving cars and is a prerequisite for Augmented and Virtual Reality systems.

State-of-the-art approaches to visual localization commonly rely on correspondences between 2D pixel positions and 3D points in the scene [13, 16, 29, 53, 72, 73, 75, 83, 84, 86]. Such a formulation estimates the camera pose using a Perspective-n-Point (PnP) solver [1, 14, 31, 40, 41] inside a RANSAC loop [6, 19, 28, 44]. These 2D-3D correspondences are traditionally computed by matching local image features. Recent localization systems can handle large scenes with complex geometry and appearance changes over time.

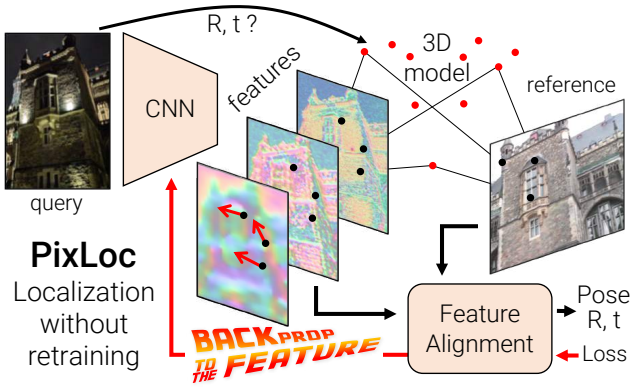

Figure 1. **Learning scene-agnostic localization.** Deep neural networks should not have to rediscover well-understood geometric principles. We only need to learn good features: PixLoc is trained end-to-end to estimate the pose of an image by aligning deep features with a reference 3D model via a differentiable optimization.

They leverage deep neural networks that learn to extract such features [8, 23, 25, 65, 69, 80, 96], to match them [65, 73], and to filter outlier correspondences [12, 42, 60, 73, 87].

Training a feature matching pipeline in an end-to-end manner is challenging and unstable as its complexity hinders gradients propagation [8]. An alternative is to train a convolutional neural network (CNN) to regress geometric quantities such as camera poses [5, 24, 35, 37, 43, 92, 99] or the 3D scene coordinate corresponding to each pixel [9–11, 13, 16, 17, 46, 82, 95]. While these approaches can be trained end-to-end, they come with their own drawbacks. Absolute pose and coordinate regression are scene-specific and require to be trained for or adapted to new scenes [16, 17]. Generalization to new viewing conditions, *e.g.*, localizing night-time images when training only on daytime photos, and handling larger, more complex scenes [80, 84] are open challenges for such approaches. Additionally, absolute or relative pose regression has limited accuracy and often fails to generalize to new viewpoints [78, 99]. While regressing poses relative to a set of reference images [5, 24, 43, 99] is in theory scene-agnostic, generalization to strongly differing scenes without a significant drop in pose accuracy [78, 99] has, to the best of our knowledge, not been shown so far.

*denotes equal contribution

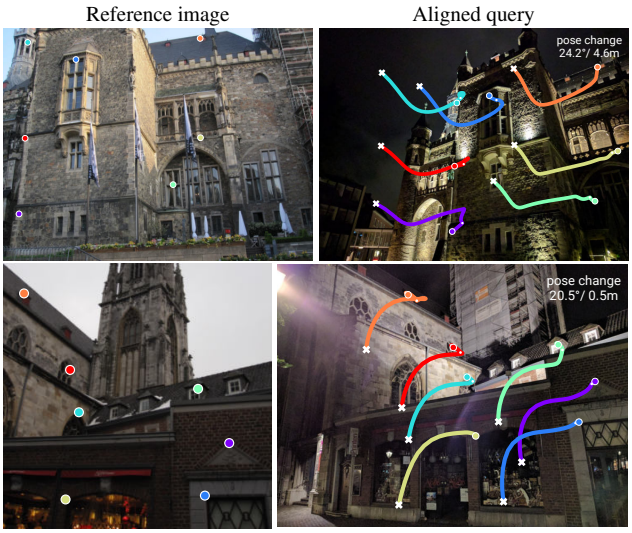

Figure 2. **Alignment for localization.** Although only based on local gradients, direct alignment works well thanks to deep features, despite the coarse initial pose estimate and strong appearance changes. Here points travel from crosses to colored dots.

What hinders the generalization of existing end-to-end regression methods is that they predict camera poses or 3D geometry solely from image information. In practice, such quantities are often readily available. Pose priors can be obtained via image retrieval or sensors such as GPS. At the same time, the 3D scene geometry is often provided as a by-product of the 3D reconstruction systems that generate the training poses, *e.g.* with Structure-from-Motion or SLAM.

Inspired by direct image alignment [22, 26, 27, 63, 90, 91] and learned image representations for outlier rejection [42], we argue that end-to-end visual localization algorithms should focus on representation learning. Rather than devoting model capacity and data to learn basic geometric relations or encode 3D maps, they should rely on well-understood geometric principles and instead learn robustness to appearance and structural changes.

In this paper, we introduce a trainable algorithm, PixLoc, that localizes an image by aligning it to an explicit 3D model of the scene based on dense features extracted by a CNN (Figure 1). By relying on classical geometric optimization, the network does not need to learn pose regression itself, but only to extract suitable features, making the algorithm accurate and scene-agnostic. We train PixLoc end-to-end, from pixels to pose, by unrolling the direct alignment and supervising only the pose. Given an initial pose obtained by image retrieval, our formulation results in a simple localization pipeline competitive with complex state-of-the-art approaches, even when the latter are trained specifically per scene. PixLoc can also refine poses estimated by any existing approach as a lightweight post-processing step. Through detailed experiments, we show that our method generalizes well to new scenes, *e.g.*, from outdoor to indoor scenes, and challenging viewing conditions. To the best of our knowledge, PixLoc is the first end-to-end visual localization approach to exhibit such exceptional generalization.

2. Related work

Accurate visual localization commonly relies on estimating correspondences between 2D pixel positions and 3D scene coordinates. Such approaches detect, describe [7, 50], and match [32, 47, 49, 75, 83, 98] local features, maintain an explicit sparse 3D representation of the environment, and sometimes leverage image retrieval [33, 88] to scale to large scenes [32, 59, 72, 77, 84, 89]. Recently, many of these components have been learned with great success [2, 23, 25, 60, 62, 67, 69, 73, 97], but often independently and not end-to-end due to the complexity of such systems. Here we introduce a simpler alternative to feature matching, finally enabling stable end-to-end training. Our solution can learn more powerful priors than individual blocks, yet remains highly flexible and interpretable.

End-to-end learning for localization has recently received much attention. Common approaches encode the scene into a deep network by regressing from an input image to an absolute pose [35, 37, 61, 68, 92] or 3D scene coordinates [9, 13, 16, 17, 82]. Pose regression lacks geometric constraints and thus does not generalize well to novel viewpoints or appearances [78, 80], while coordinate regression is more robust. Both do not scale well due to the limited network capacity [11, 84] and require for each new scene either costly retraining or adaptation [16, 17]. ESAC [11] improves the scalability by training an ensemble of regressors, each specialized in a scene subset, but is still significantly less accurate than feature-based methods in larger environments.

Differently, some approaches regress a camera pose relative to one or more training images [5, 24, 43, 99], often after an explicit retrieval step. They do no memorize the scene geometry and are thus scene-agnostic, but, similar to absolute regressors, are less accurate than feature-based methods [78, 99]. Closer to ours, SANet [95] takes the scene representation out of the network by regressing 3D coordinates from an input 3D point cloud. Critically, all top-performing learnable approaches are at least trained per-dataset, if not per-scene, and are limited to small environments [37, 82]. In this work we demonstrate the first end-to-end learnable network that generalizes across scenes, including from outdoor to indoor, and that delivers performance competitive with complex pipelines on large real-world datasets, thanks to a differentiable pose solver.

Learning camera pose optimization can be tackled by unrolling the optimizer for a fixed number of steps [21, 52, 54, 85, 93, 94], computing implicit derivatives [13, 15, 18, 34, 70], or crafting losses to mimic optimization steps [90, 91]. Multiple works have proposed to learn components of these optimizers [21, 52, 85], with added complexity and unclear

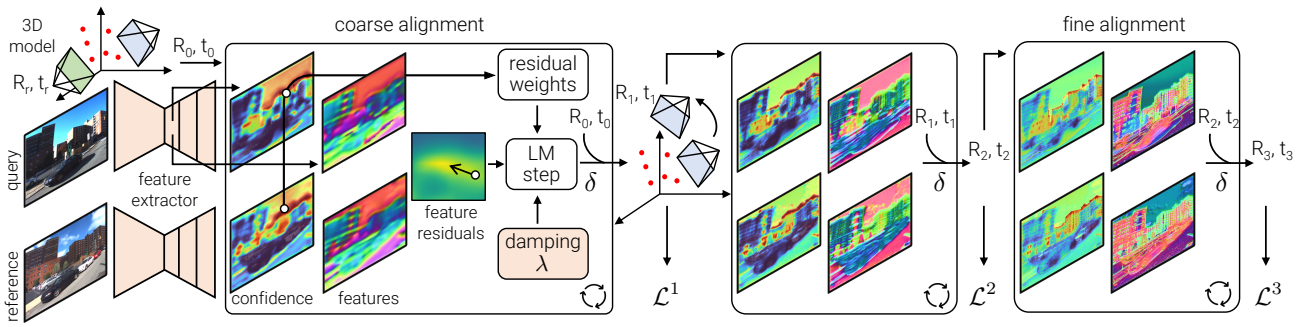

Figure 3. **Pose estimation with PixLoc.** Given a sparse 3D model and a coarse initial pose $(\mathbf{R}_0, \mathbf{t}_0)$, PixLoc extracts multilevel features with pixelwise confidences for query and reference images. The Levenberg-Marquardt optimization then aligns corresponding features according to the 3D points, guided by the confidence, from the coarse to the fine level. We only supervise the pose predicted at each level.

generalization. Some of these formulations optimize reprojection errors over sparse points, while others use direct objectives for (semi-)dense image alignment. The latter are attractive for their simplicity and accuracy, but usually do not scale well. Like their classical counterparts [26, 38], they also suffer from a small basin of convergence, limiting them to frame tracking. In contrast, PixLoc is explicitly trained for wide-baseline cross-condition camera pose estimation from sparse measurements (Figure 2). By focusing on learning good features, it shows good generalization yet learns sensible data priors that shape the optimization objective.

3. PixLoc: from pixels to pose

Overview: PixLoc localizes by aligning query and reference images according to the known 3D structure of the scene. The alignment consists of a few steps that minimize an error over deep features predicted from the input images by a CNN (Figure 3). The CNN and the optimization parameters are trained end-to-end from ground truth poses.

Motivation: In absolute pose and scene coordinate regression from a single image, a deep neural network learns to i) recognize the approximate location in a scene, ii) recognize robust visual features tailored to this scene, and iii) regress accurate geometric quantities like pose or coordinates. Since CNNs can learn features that generalize well across appearances and geometries, i) and ii) do not need to be tied to a specific scene, and i) is already solved by image retrieval. On the other hand, iii) is tackled by classical geometry using feature matching [19, 20, 28] or image alignment [4, 26, 27, 51] and a 3D representation. We should thus focus on learning robust and generic features, making the pose estimation scene-agnostic and tightly constrained by geometry. The challenge lies in how to define good features to localize. We solve this by making the geometric estimation differentiable and supervise only the final pose estimate. Differently from pose or coordinate regression, we assume that a 3D scene representation is available. This requirement is easily met in practice since the reference poses are usually obtained by sparse or dense 3D reconstruction.

Problem formulation: Our goal is to estimate the 6-DoF pose $(\mathbf{R}, \mathbf{t}) \in \mathbf{SE}(3)$ of a query image $\mathbf{I}_q$, where $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation vector in the camera frame. We are given a 3D representation of the environment, such as a sparse or dense 3D point cloud $\{\mathbf{P}_i\}$ and posed reference images $\{\mathbf{I}_k\}$, collectively called the reference data.

3.1. Localization as image alignment

Image Representation: The sparse alignment is performed over learned feature representations of the images. We leverage CNNs and their ability to extract a hierarchy of features at multiple levels. For each query image $\mathbf{I}_q$ and reference image $\mathbf{I}_k$, a CNN extracts a D_l-dimensional feature map $\mathbf{F}^l \in \mathbb{R}^{W_l \times H_l \times D_l}$ at each level $l \in \{L, ..., 1\}$. Those have decreasing resolution and progressively encode richer semantic information and a larger spatial context of the image. The features are L_2-normalized along the channels to improve their robustness and generalization across datasets.

This learned representation, inspired by past works on handcrafted and learned features for camera tracking [22, 52, 63, 85, 90, 93], is robust to large illumination or viewpoint changes and provides meaningful gradients for successful alignments despite poor initial pose estimates. In contrast, classical direct alignment [4, 26, 27, 51] operates on the original image intensity, which is not robust to long-term changes encountered in common localization scenarios, and resorts to Gaussian image pyramids, which still largely limits the convergence to frame-to-frame tracking.

Direct alignment: The goal of the geometric optimization is to find the pose $(\mathbf{R}, \mathbf{t})$ which minimizes the difference in appearance between the query image and each reference image. For a given feature level l and each 3D point i observed in each reference image k, we define a residual:

$$\mathbf{r}_k^i = \mathbf{F}_q^l \left[\mathbf{p}_q^i\right] - \mathbf{F}_k^l \left[\mathbf{p}_k^i\right] \in \mathbb{R}^D \quad , \tag{1}$$

where $\mathbf{p}_q^i = \Pi\left(\mathbf{R}\mathbf{P}_i + \mathbf{t}\right)$ is the projection of i in the query given its current pose estimate and $[\cdot]$ is a lookup with sub-

pixel interpolation. The total error over N observations is

$$E_l(\mathbf{R}, \mathbf{t}) = \sum_{i,k} w_k^i \, \rho \left(\left\| \mathbf{r}_k^i \right\|_2^2 \right) \quad , \tag{2}$$

where ρ is a robust cost function [30] with derivative ρ' and w_k^i is a per-residual weight. This nonlinear least-squares cost is iteratively minimized from an initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ using the Levenberg-Marquardt (LM) algorithm [45, 58].

To maximize the convergence basin, we optimize each feature level successively, starting with the coarsest level l=1, and initialize each with the result of the previous level. Low-resolution feature maps are thus responsible for the robustness of the pose prediction while finer features enhance its accuracy. Each pose update $\boldsymbol{\delta} \in \mathbb{R}^6$ is parametrized on the $\mathbf{SE}(3)$ manifold using its Lie algebra. We stack all residuals into $\mathbf{r} \in \mathbb{R}^{ND}$ and all weights into $\mathbf{W} = \mathrm{diag}_{i,k} \left(w_k^i \, \rho' \right)$ and write the Jacobian and Hessian matrices as

$$\mathbf{J}_{i,k} = \frac{\partial \mathbf{r}_k^i}{\partial \boldsymbol{\delta}} = \frac{\partial \mathbf{F}_q}{\partial \mathbf{p}_q^i} \frac{\partial \mathbf{p}_q^i}{\partial \boldsymbol{\delta}} \quad \text{and} \quad \mathbf{H} = \mathbf{J}^\top \mathbf{W} \mathbf{J} \quad . \tag{3}$$

The update is computed by damping the Hessian and solving the linear system:

$$\boldsymbol{\delta} = - \left(\mathbf{H} + \lambda \, \mathrm{diag} \left(\mathbf{H} \right) \right)^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{r} \quad , \tag{4}$$

where λ, the damping factor, interpolates between the Gauss-Newton (λ=0) and gradient descent ($\lambda \to \infty$) formulations and is usually adjusted at each iteration using diverse heuristics [45, 56, 58]. Finally, the new pose is computed by left-multiplication on the manifold as

$$\begin{bmatrix} \mathbf{R}^+ & \mathbf{t}^+ \end{bmatrix} = \exp \left(\boldsymbol{\delta}^\wedge \right)^\top \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad , \tag{5}$$

where $\cdot^\wedge$ is the skew operator. The optimization stops when the update $\boldsymbol{\delta}$ is small enough.

Infusing visual priors: The steps described above are identical to the classical photometric alignment [4, 26, 51]. The CNN is however capable of learning complex visual priors – we therefore would like to give it the ability to steer the optimization towards the correct pose. To this end, the CNN predicts an uncertainty map $\mathbf{U}_k^l \in \mathbb{R}_{>0}^{W_l \times H_l}$ along with each feature map. The pointwise uncertainties of the query and reference images are combined into a per-residual weight as

$$w_k^i = u_q^i \, u_k^i = \frac{1}{1 + \mathbf{U}_q^l \left[\mathbf{p}_q^i \right]} \frac{1}{1 + \mathbf{U}_k^l \left[\mathbf{p}_k^i \right]} \in [0, 1] \quad . \tag{6}$$

The weight is 1 if the 3D point projects into a location with low uncertainty in both the query and the reference images. It tends to 0 as either of the location is uncertain. Here w_k^i is not explicitly supervised, but rather learned as to maximize the pose accuracy. A similar formulation was applied to direct RGB-D frame tracking in a concurrent work [94].

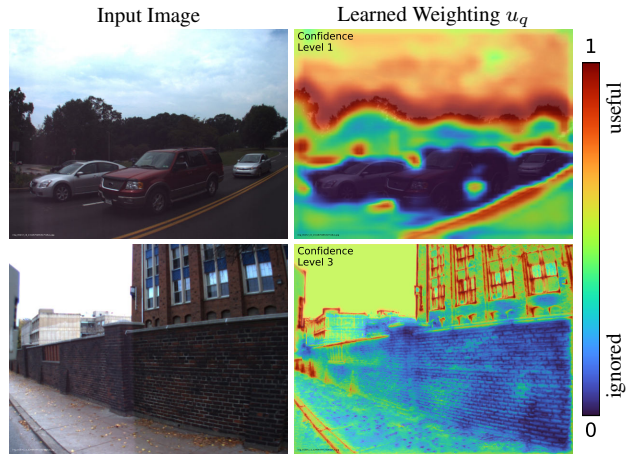

Figure 4. **Good features to localize.** PixLoc learns to ignore dynamic objects like cars (top) or fallen leaves (bottom) and repeated patterns like the brick wall. It focuses on road markings, silhouettes of trees, or prominent structures on buildings. See also Figure 6.

This weighting can capture multiple scenarios. First, the network can learn to be uncertain when it cannot predict invariant features, *e.g.*, because of domain shift, similarly to an aleatoric uncertainty [36]. The uncertainty can also be high for locations that can be well described by the CNN, but which consistently push the optimization away from the correct pose by introducing local minima in the cost landscape. This encompasses dynamic objects or repeated patterns and symmetries, as shown in Figures 4 and 6. The uncertainty is different for each level, as different cues might be useful at different stages of the optimization.

Fitting the optimizer to the data: Levenberg-Marquardt is a generic optimization algorithm that involves several heuristics, such as the choice of robust cost function ρ or of the damping factor λ. Past works on learned optimization employ deep networks to predict ρ' [52], λ [52, 85], or even the pose update $\boldsymbol{\delta}$ [21, 54], from the residuals and visual features. We argue that this can greatly impair the ability to generalize to new data distributions, as it ties the optimizer to the visual-semantic content of the training data. Instead, it is desirable to fit the optimizer to the distribution of poses or residuals but not to their semantic content. As such, we propose to make λ a fixed model parameter and learn it by gradient descent along with the CNN.

Importantly, we learn a different factor for each of the 6 pose parameters and for each feature level, replacing the scalar λ by $\boldsymbol{\lambda}_l \in \mathbb{R}^6$, parametrized by $\boldsymbol{\theta}_l$ as

$$\log_{10} \boldsymbol{\lambda}_l = \lambda_{\min} + \mathrm{sigmoid} \left(\boldsymbol{\theta}_l \right) \left(\lambda_{\max} - \lambda_{\min} \right) \quad . \tag{7}$$

This adjusts the curvature of the individual pose parameters during training, and directly learns motion priors from the data. For example, when the camera is mounted on a car or a robot that is mostly upright, we expect the damping for the in-plane rotation to be large. In contrast, common heuristics treat all pose parameters equally and do not permit

a per-parameter damping. We show in Appendix B that the learned damping parameters vary with the training data.

3.2. Learning from poses

As the CNN never sees 3D points, PixLoc can generalize to any 3D structure available. This includes sparse SfM point clouds, dense depth maps from stereo or RGBD sensors, meshes, Lidar scans, but also lines and other primitives.

Training: The optimization algorithm presented here is end-to-end differentiable and only involves operations commonly supported by deep learning frameworks. Gradients thus flow from the pose all the way to the pixels, through the feature and uncertainty maps and the CNN. Thanks to the uncertainties and robust cost, PixLoc is robust to incorrect 3D geometry and works well with noisy reference data like sparse SfM models. During training, an imperfect 3D representation is sufficient – our approach does not require accurate or dense 3D models.

Loss function: Our approach is trained by comparing the pose estimated at each level $(\mathbf{R}_l, \mathbf{t}_l)$ to its ground truth $(\bar{\mathbf{R}}, \bar{\mathbf{t}})$. We minimize the reprojection error of the 3D points:

$$\mathcal{L} = \frac{1}{L} \sum_l \sum_i \left\| \Pi\left(\mathbf{R}_l \mathbf{P}_i + \mathbf{t}_l\right) - \Pi\left(\bar{\mathbf{R}} \mathbf{P}_i + \bar{\mathbf{t}}\right) \right\|_\gamma , \quad (8)$$

where γ is the Huber cost. This loss weights the supervision of the rotation and translation adaptively for each training example [35] and is invariant to the scale of the scene, making it possible to train with data generated by SfM. To prevent hard examples from smoothing the fine features, we apply the loss at a given level only if the previous one succeeded in bringing the pose sufficiently close to the ground truth. Otherwise, the subsequent loss terms are ignored.

3.3. Comparisons to existing approaches

PixLoc vs. sparse matching: Pose estimation via local feature matching comprises multiple operations that are non-differentiable, such as keypoint and correspondence selection or RANSAC. Bhowmik *et al*. [8] proposed a formulation based on reinforcement learning, which suffers from high variance and thus requires a strong pretraining. In contrast, our approach is extremely simple and converges well from generic weights trained for image classification.

PixLoc vs. GN-Net: Von Stumberg *et al*. [90, 91] recently trained deep features for cross-season localization via direct alignment. Their works focus on small-baseline scenarios and require accurate pixelwise ground truth correspondences and substantial hyperparameter tuning. In contrast, we leverage the power of differentiable programming to match the test and training conditions and learn additional strong priors from noisy data. We compare with their loss in Section 5.4.

Figure 5. **Wide convergence.** For a red point in the reference image (left), we highlight in the query (right) the multilevel basin of attraction colored by the 2D gradient angle ${}^{\partial \mathbf{F}_q}/_{\partial \mathbf{P}_q^i}{}^\top \mathbf{r}_k^i$. Deep features ensure a wide convergence despite appearance changes.

4. Localization pipeline

PixLoc can be a competitive standalone localization module when coupled with image retrieval, but can also refine poses obtained by previous approaches. It only requires a 3D model and a coarse initial pose, which we now discuss.

Initialization: How accurate the initial pose should be depends on the basin of convergence of the alignment. Features from a deep CNN with a large receptive field ensure a large basin (Figure 5). To further increase it, we apply PixLoc to image pyramids, starting at the lowest resolution, yielding coarsest feature maps of size W=16. To keep the pipeline simple, we select the initial pose as the pose of the first reference image returned by image retrieval. This results in a good convergence in most scenarios. When retrieval is not sufficiently robust and returns an incorrect location, as in the most challenging conditions, one could improve the performance by reranking using covisiblity clustering [72, 75] or pose verification with sparse [74, 98] or dense matching [84].

3D structure: For simplicity and unless mentioned, for both training and evaluation, we use sparse SfM models triangulated from posed reference images using hloc [71, 72] and COLMAP [79, 81]. Given a subset of reference images, *e.g*. top-5 retrieved, we gather all the 3D points that they observe, extract multilevel features at their 2D observations, and average them based on their confidence.

5. Experiments

We first compare against existing learning-based localization approaches and show that PixLoc often performs better than those trained for each scene and generalizes well across environments. We then compare PixLoc with state-of-the-art feature matching pipelines on a large-scale benchmark and show that it delivers competitive accuracy, but can also enhance them when used as a post-processing. Finally, we provide insights into PixLoc through an ablation study.

Architecture: We employ a UNet feature extractor based on a VGG19 encoder pretrained on ImageNet, and extract L=3 feature maps with strides 1, 4, and 16, and dimensions D_l=32, 128, and 128, respectively. PixLoc is implemented in PyTorch [64], extracts features for an image in around

	Method	Cambridge Landmarks - outdoor					7Scenes - indoor							
		Court	King's	Hospital	Shop	St. Mary's	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Recall↑
IR	DenseVLAD [88]	-	280/5.7	401/7.1	111/7.6	231/8.0	21/12.5	33/13.8	15/14.9	28/11.2	31/11.3	30/12.3	25/15.8	-
IR	Oracle	207/7.0	137/7.2	323/8.3	133/7.8	204/8.1	16/12.3	26/13.6	12/14.7	20/11.5	19/14.0	18/15.0	17/18.1	0.17
FM	AS [75]†	24/0.13	13/0.22	20/0.36	4/0.21	8/0.25	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01	68.7
FM	InLoc [84]	-	-	-	-	-	3/1.05	3/1.07	2/1.16	3/1.05	5/1.55	4/1.31	9/2.47	66.3
FM	hloc [72]	16/0.11	12/0.20	15/0.30	4/0.20	7/0.21	2/0.85	2/0.94	1/0.75	3/0.92	5/1.30	4/1.40	5/1.47	73.1
end-to-end	DSAC* [13]	49/0.3	15/0.3	21/0.4	**5**/0.3	13/0.4	**2**/1.10	**2**/1.24	**1**/1.82	**3**/1.15	**4**/1.34	4/1.68	**3**/1.16	**85.2**
end-to-end	HACNet [46]	**28**/0.2	18/0.3	19/**0.3**	6/0.3	**9/0.3**	**2/0.7**	**2**/0.9	**1**/0.9	**3/0.8**	**4/1.0**	4/**1.2**	**3/0.8**	84.8
end-to-end	CAMNet [24]	-	-	-	-	-	4/1.73	3/1.74	5/1.98	4/1.62	**4**/1.64	4/1.63	4/1.51	-
end-to-end	SANet [95]	328/1.95	32/0.54	32/0.53	10/0.47	16/0.57	3/0.88	3/1.08	2/1.48	**3**/1.00	5/1.32	4/1.40	16/4.59	68.2
end-to-end	**PixLoc**	30/**0.14**	**14/0.24**	**16/0.32**	**5/0.23**	10/**0.34**	**2**/0.80	**2/0.73**	**1/0.82**	**3/0.82**	**4**/1.21	**3/1.20**	5/1.30	75.7
end-to-end	+ Oracle prior	21/0.12	13/0.24	16/0.31	5/0.22	9/0.28	2/0.80	2/0.70	1/0.78	3/0.80	4/1.13	3/1.14	4/1.08	81.7

Table 1. **Visual localization on the Cambridge Landmarks and 7Scenes datasets.** We report the median translation (cm) and rotation (°) errors and the average recall at (5cm, 5°). Despite its simplicity, PixLoc is competitive with complex feature matching (FM) pipelines and performs similarly to, and often better than, geometric regression models, including those specifically trained per scene (red). Our model, trained solely on outdoor data, generalizes well to unseen outdoor and indoor scenes, and can benefit from improved image retrieval (IR). The best results in the end-to-end category are in bold (oracle excluded). †The results for AS were kindly provided by the authors.

100ms, and optimizes the pose in 200ms to 1s depending on the number of points. More details are in the Appendix.

Training: We train two versions of PixLoc to demonstrate its ability to learn environment-specific priors. The benefits of such priors are analyzed in Appendix B. One version is trained on the MegaDepth dataset [48], composed of crowd-sourced images depicting popular landmarks around the world, and the other on the training set of the Extended CMU Seasons dataset [3, 76, 86], a collection of sequences captured by car-mounted cameras in urban and rural environments. The latter dataset exhibits large seasonal changes with often only natural structures like trees being visible in the images, which are challenging for feature matching. We sample covisible image pairs and simulate the localization of one image with respect to the other, given its observed 3D points. The optimization runs for 15 iterations at each level and is initialized with the pose of the reference image.

5.1. Comparison to learned approaches

We first evaluate on the Cambridge Landmarks [37] and 7Scenes [82] datasets, which are commonly used to compare learning-based approaches.

Evaluation: The two datasets contain 5 outdoor and 7 indoor scenes, respectively, each composed of posed reference images and query images captured along different trajectories and conditions. We report for each scene the median translation (cm) and rotation (°) errors [37], as well as the average localization recall at (5cm, 5°) for 7Scenes [82].

Baselines: We compare with multiple state-of-the-art learning-based approaches. Those trained per scene include 3D coordinate regression networks DSAC* RGB [13] and HACNet [46], and CAMNet [24], which regresses a relative pose following image retrieval. SANet [95] is scene-agnostic. All methods, including PixLoc, use 3D points from SfM and dense depth maps for Cambridge and 7Scenes, respectively.

We report image retrieval with DenseVLAD [88] but not PoseNet and its variants as they perform similarly [78]. We also compare with feature matching pipelines. Active Search (AS) [75] performs global matching with SIFT [50]. InLoc [84] and hloc [72] first perform image retrieval before matching features to the retrieved images. The former matches dense deep descriptors and relies on a dense reference 3D model, while hloc matches SuperPoint [23] features with SuperGlue [73] and builds a sparse 3D SfM reference point cloud. PixLoc, trained on MegaDepth, is initialized with image retrieval obtained with either DenseVLAD [88] or an oracle, which returns the reference image containing the largest number of inlier matches found by hloc. This oracle shows the benefits of better image retrieval using a more complex pipeline without ground truth information.

Results: The evaluation results are reported in Table 1. On outdoor data, PixLoc consistently outperforms the only end-to-end scene-agnostic method, SANet, and performs similarly to, or better than scene-specific approaches. It is competitive for indoor scenes, despite being trained on outdoor Internet data only. This confirms that deep features are all we need for accurate localization and that they generalize well despite end-to-end training. PixLoc performs comparably to the best feature matching localizer hloc – a complex pipeline that integrates learned feature detection, description, and matching. Localizing with the oracle prior only marginally improves the performance, confirming that image retrieval can be sufficiently accurate for the pose optimization to converge to the correct minimum.

5.2. Large-scale localization

We now evaluate on a large-scale, long-term localization benchmark [76] that exhibits considerably more diversity in geometry and appearance than Cambridge and 7Scenes.

Evaluation: The benchmark is composed of three datasets.

	Method	Aachen Day-Night		RobotCar Seasons		Extended CMU Seasons		
		Day	Night	Day	Night	Urban	Suburban	Park
IR	DenseVLAD [88]	0.0 / 0.1 / 22.8	0.0 / 1.0 / 19.4	7.6 / 31.2 / 91.2	1.0 / 4.4 / 22.7	14.7 / 36.3 / 83.9	5.3 / 18.7 / 73.9	5.2 / 19.1 / 62.0
IR	NetVLAD [2]	0.0 / 0.2 / 18.9	0.0 / 0.0 / 14.3	6.4 / 26.3 / 90.9	0.3 / 2.3 / 15.9	12.2 / 31.5 / 89.8	3.7 / 13.9 / 74.7	2.6 / 10.4 / 55.9
IR	Oracle	0.0 / 0.2 / 22.1	0.0 / 1.0 / 22.4	9.6 / 38.1 / 96.3	4.3 / 16.4 / 84.9	21.2 / 52.2 / 98.2	8.6 / 29.5 / 94.3	8.2 / 31.5 / 90.2
E2E	ESAC [11]	42.6 / 59.6 / 75.5	6.1 / 10.2 / 18.4	-	-	-	-	-
E2E	**Pixloc**	64.3 / 69.3 / 77.4	51.0 / 55.1 / 67.3	52.7 / 77.5 / 93.9	12.0 / 20.7 / 45.4	88.3 / 90.4 / 93.7	79.6 / 81.1 / 85.2	61.0 / 62.5 / 69.4
E2E	+ Oracle prior	68.0 / 74.6 / 80.8	57.1 / 69.4 / 76.5	55.8 / 80.8 / 96.4	23.6 / 40.3 / 77.8	92.8 / 95.1 / 98.5	91.9 / 93.4 / 95.8	84.0 / 85.8 / 90.9
FM	AS [75]	85.3 / 92.2 / 97.9	39.8 / 49.0 / 64.3	50.9 / 80.2 / 96.6	6.9 / 15.6 / 31.7	81.0 / 87.3 / 92.4	62.6 / 70.9 / 81.0	45.5 / 51.6 / 62.0
FM	D2-Net [25]	84.3 / 91.9 / 96.2	75.5 / 87.8 / 95.9	54.5 / 80.0 / 95.3	20.4 / 40.1 / 55.0	94.0 / 97.7 / 99.1	93.0 / **95.7** / **98.3**	**89.2** / **93.2** / **95.0**
FM	S2DNet [29]	84.5 / 90.3 / 95.3	74.5 / 82.7 / 94.9	53.9 / 80.6 / 95.8	14.5 / 40.2 / 69.7	-	-	-
FM	hloc [72]	**89.6** / **95.4** / **98.8**	**86.7** / **93.9** / **100.**	**56.9** / 81.7 / **98.1**	33.3 / 65.9 / 88.8	95.5 / 98.6 / **99.3**	90.9 / 94.2 / 97.1	85.7 / 89.0 / 91.6
FM	**+ PixLoc refine**	84.7 / 94.2 / **98.8**	81.6 / **93.9** / **100.**	**56.9** / **82.0** / **98.1**	**34.9** / **67.7** / **89.5**	**96.9** / **98.9** / **99.3**	**93.3** / 95.4 / 97.1	87.0 / 89.5 / 91.6

Table 2. **Large-scale localization on the Aachen, RobotCar, and CMU datasets.** PixLoc, when initialized from image retrieval (IR), can substantially improve IR accuracy. It consistently outperforms the only scalable end-to-end (E2E) method ESAC, and performs reasonably compared to complex feature matching (FM) pipelines. PixLoc can also improve their accuracy by refining their local features (+ refine).

The Aachen Day-Night [76, 77] dataset is captured by hand-held devices. The RobotCar [55, 76] and the Extended CMU [3, 86] seasons datasets are captured by car-mounted cameras across different seasons, weather, and times, in urban and rural areas. All datasets have posed reference images, SfM models, and query images. We report the localization recall at thresholds (25cm, 2°), (50cm, 5°), and (5m, 10°).

Baselines: Multiple past works [11, 78, 80, 84] report that end-to-end learning-based methods cannot be stably trained on such large-scale datasets. The only exception is ESAC [11], which reports results for Aachen only. We additionally compare against image retrieval with DenseVLAD [88] and NetVLAD [2] and feature matching pipelines based on Active Search [75], D2-Net [25], S2DNet [29], and hloc [72]. PixLoc is trained on MegaDepth (CMU) when evaluated on Aachen (RobotCar and CMU). It is initialized by the weighted average [66] of the top-3 poses retrieved by NetVLAD for Aachen and top-1 for RobotCar and CMU. The oracle prior is identical to Section 5.1.

Results: We report the results in Table 2. When the initial pose prior is provided by image retrieval, PixLoc is a simple localization system that is more accurate than ESAC, especially in the challenging condition of night. This improvement is not brought by the significantly less accurate image retrieval. PixLoc is however less robust than the feature matching pipelines, which is mostly due to the naive pose prior, as our algorithm cannot converge if the retrieval returns the incorrect location. Using the oracle prior partially bridges the gap, and makes PixLoc competitive on driving datasets like CMU and RobotCar. It however lags behind on Aachen, where the reference images are significantly sparser and the initial priors are therefore much coarser. Naturally, this is challenging for direct alignment, irrespective of the daytime or nighttime condition. PixLoc is nevertheless the only end-to-end trained method that can scale to this large extent without requiring retraining.

Method	AUC 25cm	AUC 1m
● Photometric optimization	1.2	3.3
● + deep features (GN loss)	13.2	21.4
● + unroll (fixed damping λ)	49.2	67.0
● + confidence $w_{i,k}$	53.3	72.5
● + learned λ **(PixLoc-full)**	**59.8**	**79.0**
● D=128→16 (PixLoc-light)	50.4	70.2

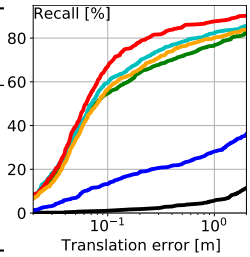

Table 3. **Ablation study.** Unrolling the optimizer and learning features, damping factor, and confidences all contribute to the performance of PixLoc over classical photometric alignment. Learning compact features as in past works [52, 90] results in a drop of performance compared to high-dimensional representations.

5.3. Pose post-processing with PixLoc

We showed that too large baselines between query and reference images can cause PixLoc to converge to an incorrect local minima. Naturally, PixLoc can also serve as a post-processing step for any other localization pipeline.

Refinement in challenging conditions: We apply PixLoc to refine the poses estimated by hloc in the previous localization experiment. We consider all 3D points that have at least one inlier match. The results are shown in the last row of Table 2. PixLoc brings consistent improvement on CMU, especially in the fine threshold, with up to +2.4% recall. It also increases the pose accuracy at all thresholds on RobotCar Night, which exhibits significant motion blur, a difficult condition for sparse keypoint detection. However, no improvement can be observed on RobotCar Day, while the refinement is detrimental on Aachen at 0.25m. This might be due to inaccurate ground truth poses or camera intrinsics, for which we provide evidence in Appendix D.

5.4. Additional insights

Ablation study: We justify our design decisions by comparing different variants of PixLoc. We have attempted to train our CNN with the Gauss-Newton loss [90], but it fails to

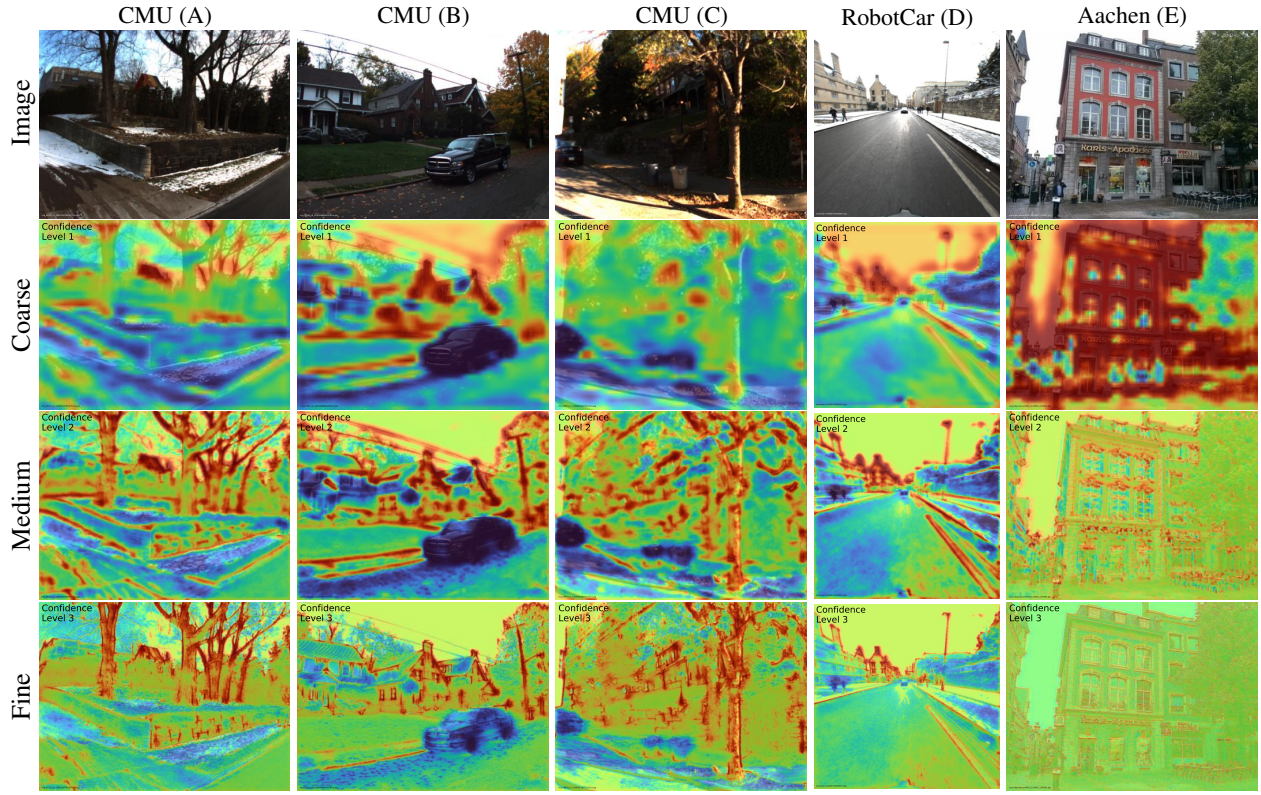

Figure 6. **Which features matter?** In driving scenarios (A-D), besides dynamic objects such as cars, PixLoc learns to ignore (in blue) more subtle short-term entities like snow (A), fallen leaves (B), trash bins (C), or shadows at all feature levels. Instead, it focuses (in red) on poles, tree trunks, road markings, power lines, or building silhouettes. Repetitive structures like windows or road cracks are often ignored at first but later on used for fine alignment. Differently, when trained on urban scenes (E), it ignores trees as buildings are more stable structures.

converge on our challenging training data despite extensive hyperparameter tuning. We select difficult query-reference pairs in the CMU validation set and report the recall curve and its area (AUC) in Table 3. As can be seen, all components significantly contribute to PixLoc's performance.

Interpretability: Visualizing the weight maps u_q learned by PixLoc helps us discover what cues are useful or detrimental for localizing in which environments. We show visualizations in Figure 6 and in Appendix E.

Limitations: PixLoc relies on gradients of CNN features, which can only encode a limited context. It is thus a local method and can fall into incorrect minima for excessively large initial reprojection errors arising from large viewpoint changes. We analyze the convergence in Appendix A. PixLoc can also fail for large outliers ratios due prominent occluders and is more sensitive to camera miscalibration.

Acknowledgements: The authors thank Mihai Dusmanu, Rémi Pautrat, and Xingxing Zuo for their thoughtful comments. This work received funding through the EU Horizon 2020 project RICAIP (grant agreeement No 857306), the European Regional Development Fund under project IMPACT No. CZ.02.1.01/0.0/0.0/15 003/0000468, the Chalmers AI Research Centre (CHAIR) (VisLocLearn), and the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots). Paul-Edouard Sarlin was supported by gift funding from Huawei, and Viktor Larsson by an ETH Zurich Postdoctoral Fellowship.

6. Conclusion

In this paper, we have introduced a simple solution to end-to-end learning of camera pose estimation. In contrast to previous approaches that regress geometric quantities, we do not try to teach a deep network basic geometric principles or 3D map encoding. Instead, we go Back to the Feature: we show that learning robust and generic features is sufficient for accurate localization by leveraging classical image alignment with existing 3D maps. To the best of our knowledge, the resulting system, PixLoc, is the first end-to-end trainable approach capable of being deployed into new scenes widely differing from its training data without retraining or fine-tuning. PixLoc achieves a pose accuracy competitive with significantly more complex state-of-the-art pipelines. End-to-end training combined with uncertainty modeling enables PixLoc to learn complex yet interpretable priors.

PixLoc learns which features and objects matter for robust, long-term localization. Yet, it requires a good initialization to successfully localize. We thus see PixLoc as a first step towards deep networks that learn and reason about long-term, extreme changes of appearance and 3D structure. We believe that taking steps towards human-level spatiotemporal understanding will ultimately lead to robust, reliable, and accurate localization systems.

Appendix

A. Convergence and initial pose

Convergence: The pose optimization in PixLoc tends to converge to spurious local minima if the initial pose is too coarse, such as on the Aachen dataset, in which reference images are sparse. Since the receptive field of the CNN is limited, the convergence mostly depends on the initial 2D reprojection error, which accounts for the rotation and translation errors and for the distance to the 3D structure. The exact density of reference images required for high success thus depends on the distance to the scene.

We report in Figure 7 the success rate for different initial reprojection errors and their distribution for the oracle retrieval, with hloc as pseudo ground truth. Convergence within 1 meter is observed for 80% of the cases only when the initial error is smaller than 200 pixels and is significantly reduced for larger errors.

Initial pose: The 7Scenes and Cambridge datasets have reference poses with a high density. In driving scenarios like in the RobotCar and CMU datasets, there are no rotation changes between reference and query poses. In all these scenarios, initializing PixLoc with the pose of the first retrieved image is therefore sufficient.

To improve the performance on the Aachen dataset, the results in Table 2 rely on additional filtering steps. We first cluster the top-3 retrieved reference images based on their covisibility [72, 75] and only retain the images that belong to the largest cluster. We then perform a weighted average of the reference poses [57], where the weights are computed from the similarity of the global descriptors [66]. We compare in Table 4 the results obtained with this pose averaging and with the top-1 retrieval. To further improve the convergence, one could also rerank based on featuremetric error or initialize with poses randomly sampled around top-retrieved poses.

B. Benefits of training on different datasets

The training datasets CMU and MegaDepth reflect different scenarios, autonomous driving and tourism landmark photography, respectively. Training on each one separately allows to learn task-specific priors and demonstrates the ability of PixLoc to adapt to the environment.

Each dataset depicts scenes with different semantic elements (street-level landscapes and urban landmarks, respectively) and different changes of conditions (weather and season for CMU, cameras, occluders, and viewpoints for MegaDepth). Figure 6 mentions that the models learn to ignore different unreliable elements depending on the training dataset. For example, tree silhouettes are reliable on CMU due to the small viewpoint changes, but are ignored by the model trained on MegaDepth.

Cameras also exhibit different motions, as they are either

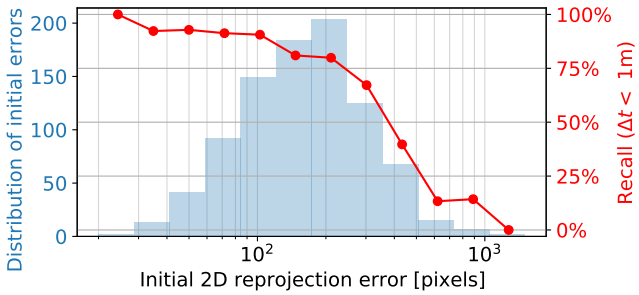

Figure 7. **Impact of the initial pose on the Aachen dataset.** The success of the pose optimization decreases with larger initial reprojection errors, which vary significantly across the 922 queries.

Initial pose	Aachen Day-Night		CMU Seasons
	Day	Night	Park
top-1	61.7 / 67.6 / 74.8	46.9 / 53.1 / 64.3	61.0 / 62.5 / 69.4
top-3 averaging	64.3 / 69.3 / 77.4	51.0 / 55.1 / 67.3	64.9 / 66.8 / 71.7
oracle prior	68.0 / 74.6 / 80.8	57.1 / 69.4 / 76.5	84.0 / 85.8 / 90.9

Table 4. **Selection of the initial pose.** Averaging the poses of the top retrieved images improves the convergence of PixLoc compared to simply selecting the pose of the first image.

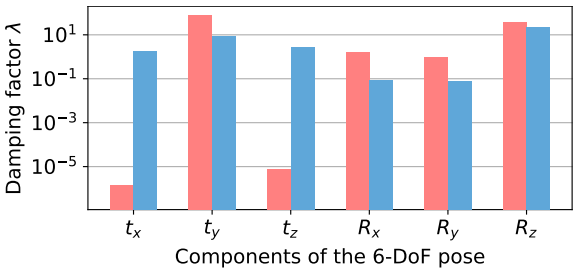

Figure 8. **Learned motion prior.** Training on data recorded with 3-DoF car-mounted cameras (CMU, in red) or with 6-DoF hand-held devices (MegaDepth, in blue) results in different motion priors learned by the damping factor $\boldsymbol{\lambda}$. Larger relative values indicate smaller expected motion in the corresponding direction.

car-mounted or hand-held. Such priors are learned by the model through the damping factors, which we visualize in Figure 8. On CMU, the motion across query and reference images is mostly a translation along the x and z axis of the camera, and never along the y axis (fixed height above the ground plane) or a rotation around the z axis (fixed roll). Differently, the motion on MegaDepth is more uniformly distributed among the 6 DoF, resulting in similar factors. The relative scale between the two sets of factors is irrelevant.

These learned priors have a noticeable impact on the performance, as shown in Table 5. Training on CMU performs better than training on MegaDepth when evaluating on a driving dataset like RobotCar. When evaluating on a totally different environment like Aachen, it however still performs better than a scene-specific approach like ESAC (shown in Table 2). PixLoc thus generalizes well across scenarios but can also learn and exploit their specificities.

Training dataset	Aachen (urban scenes like MD)		CMU (natural scenes)	
	Day	Night	Urban	Park
MD	**68.0 / 74.6 / 80.8**	**57.1 / 69.4 / 76.5**	78.3 / 81.8 / 94.6	72.5 / 75.5 / 90.3
CMU	54.4 / 62.6 / 74.3	46.9 / 54.1 / 68.4	**91.9 / 93.4 / 95.8**	**84.0 / 85.8 / 90.9**

Table 5. **Cross-dataset evaluation with oracle prior.** Training and testing in different environments does not perform as well as training for the target distribution. Task-specific priors learned by PixLoc, like semantics and motion, are thus largely beneficial.

3D from	median error in translation/rotation (cm/°) ↓							R↑
	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	
SfM	3/0.90	**2**/0.87	**1/0.79**	**3**/0.96	5/1.42	4/1.44	6/1.38	69.5
RGB-D	**2/0.80**	**2/0.73**	**1**/0.82	**3/0.82**	**4/1.21**	**3/1.20**	**5/1.30**	**75.7**

Table 6. **Depth sensor fusion vs. SfM point cloud.** For the 7Scenes indoor environment, localizing with 3D points obtained from depth maps fused across multiple view (RGB-D SLAM) is more accurate than with point clouds triangulated via SfM.

C. Accuracy of the 3D model

When localizing on the Cambridge Landmarks dataset, PixLoc relies on SfM models triangulated by hloc [72, 73]. For indoor scenes of the 7Scenes dataset, we found that the 3D SfM points are less accurate than the dense depth provided with the dataset. The results in the main paper (Table 1) are thus based on this dense depth.

More specifically, we rely on the depth maps rendered by Brachmann *et al*. [13], which are aligned to the color images and are less noisy than the original depth maps. We simply replace each 3D SfM point by back-projecting one of its 2D observations using the interpolated depth and the image pose. This 3D model has the same sparsity as the SfM point cloud but is more accurate. This process is fair as it relies on the same data as all other learning-based approaches, which use the full dense 3D model for training.

We show in Table 6 the impact on the performance of PixLoc. Using this corrected 3D model results in more accurate localization than the triangulated SfM model.

D. Inaccuracy of the ground truth poses

The RobotCar v2 dataset has publicly available ground truth poses for a subset of the queries. We project 3D SfM points into the query images using ground truth poses and those estimated by hloc. We observe in many instances a large reprojection error, where hloc poses look qualitatively more accurate. Some examples are shown in Figure 9. This might explain why no method localizes more than 58% of the daytime images at the finest threshold according to the public leaderboard [1]. This might also explain why refining poses with PixLoc does not show improvements for the day-time queries of RobotCar, as observed in Section 5.3.

Similar artifacts were found in training sequences of the Extended CMU Seasons dataset, making the training supervision noisier. We however do not know if this also applies to the poses of the test sequences because such poses are not publicly available.

[1] https://www.visuallocalization.net/benchmark/

E. Qualitative examples

We show examples of successful localization on the Extended CMU Seasons dataset in Figure 10. We show failure cases in Figure 11. Similarly, we show successful and failed examples on the Aachen Day-Night dataset in Figures 12 and 13, respectively. Videos and animations of the uncertainties and the optimization are available along with the code and trained weights at `github.com/cvg/pixloc`.

F. Attraction basin

Computation: We compute the basin of attraction of a given point by backtracking feature gradients throughout the levels and scales. For each pixel, we consider the 2 neighbors, in an 8-connected neighborhood, that are in the direction opposite to the feature gradient $\partial \mathbf{F}_q / \partial \mathbf{p}_q^i{}^\top \mathbf{r}_k^i$. A pixel is in the basin of attraction if any of those two are themselves in the basin. The voting is performed in a soft manner using the gradient angle, resulting in a basin score for each pixel. We first label the point of interest as in the basin and then iteratively run the algorithm at each level, from the finest to the coarsest level, moving to the next one when the scores stop changing. Note that the total convergence basin of the pose, which corresponds to the aggregation of all the points, might be smaller or larger.

Visualization: We show one example in Figure 5 in the main paper, where we color pixels that belong to the basin by changing their hue according to the angle of the total gradient. We show additional examples in Figure 14, but showing the gradient field as arrows only.

G. Experimental details

We now provide more details about the implementation of PixLoc and the experiments.

Implementation: The CNN and the optimizer are implemented in PyTorch [64]. The linear system of the Levenberg-Marquardt step (Equation 4) is solved using the Cholesky decomposition. The lookup of features and uncertainty is computed via bilinear interpolation. We use the Cauchy robust cost function with scale 0.1. When computing the residuals or the Jacobian matrix, we ignore points that project outside the image or within 2 pixels of the image borders. We set $\lambda_{\min}{=}{-6}$ and $\lambda_{\max}{=}5$.

Training: We train PixLoc with image pairs composed of a query image and a single reference image. For each pair, we sample 512 3D points visible in the reference image

according to the SfM covisibility information. We apply gradient checkpointing to each block of the encoder and of the decoder to minimize the GPU memory consumption. The network is trained for 50k iteration with a constant learning rate of 10^{-5} and the Adam optimizer [39]. To stabilize the training, the average loss per pair is clamped to 50 pixels and the per-parameter gradients are clipped to $[-1, 1]$.

When training on the CMU dataset, we use slices 8-12 and 22-25 for training and slices 6, 13, 21 for validation. We train with batches of 3 image pairs. The images are resized such that their smallest dimension is 720 pixels and we sample square crops of 720 pixels The query pose is initialized with the pose of the reference image.

When training on the MegaDepth dataset, we use the same split of scenes as Dusmanu *et al.* [25] and sample image pairs with an overlap score in $[0.3, 1]$. In addition, we rotate images that are not upright using the gravity direction of each scene. All images are resized such that their smallest dimension is 512 pixels, and we sample square crops of 512 pixels. PixLoc is then trained with batches of 6 image pairs. The initial pose is sampled in the range $t \in [0.75, 1]$ of the linear interpolation between the reference pose (t=0) and the ground truth query pose (t=1). Sampling initial poses that are too difficult can result in coarse features that are too smooth and uninformative at the lower-resolution scale.

Inference: In order to keep the runtime reasonable, we use 5 or 3 reference images when initializing from hloc or retrieved reference poses, respectively. The optimization runs at each level for at most 100 iterations, but stops when either the gradient or the step are small enough. When refining hloc poses, we only optimize over the medium and fine levels as the initial estimate is always sufficiently good. All images are resized such that their longest dimension is equal to 1024 pixels. For the multiscale inference, the resized images are successively aligned at scale 1/4 and 1.

Ablation study: We sample 2000 query images from slices 6, 7, 13, and 21 of the CMU dataset. To generate challenging pairs, we select the closest reference image that is at least 4 meters away. For the baseline based on a fixed damping factor λ, we use λ=10^{-2}. As GN-Net [90] has no publicly-available implementation, we reimplemented it and trained it with our settings on the CMU dataset. The GN-Net loss has several hyperparameters: the Gauss-Newton sampling vicinity, the weight of the contrastive loss, and the margin of its negative term. We performed an extensive hyperparameter search and report the best results obtained. Our training data is significantly more difficult than the one used in the original paper [90], with significantly larger baselines and appearance changes. This explains the large performance gap observed in Table 3 compared to the results originally reported.

Figure 9. **Inaccurate RobotCar ground truth poses.** We plot the projection of 3D SfM points in the query images according to the ground truth (in blue) and hloc (in red) poses. We project the same points in the reference images using the reference poses (in blue). Query points using hloc are better aligned to the reference points, indicating that the ground truth query poses are inaccurate.

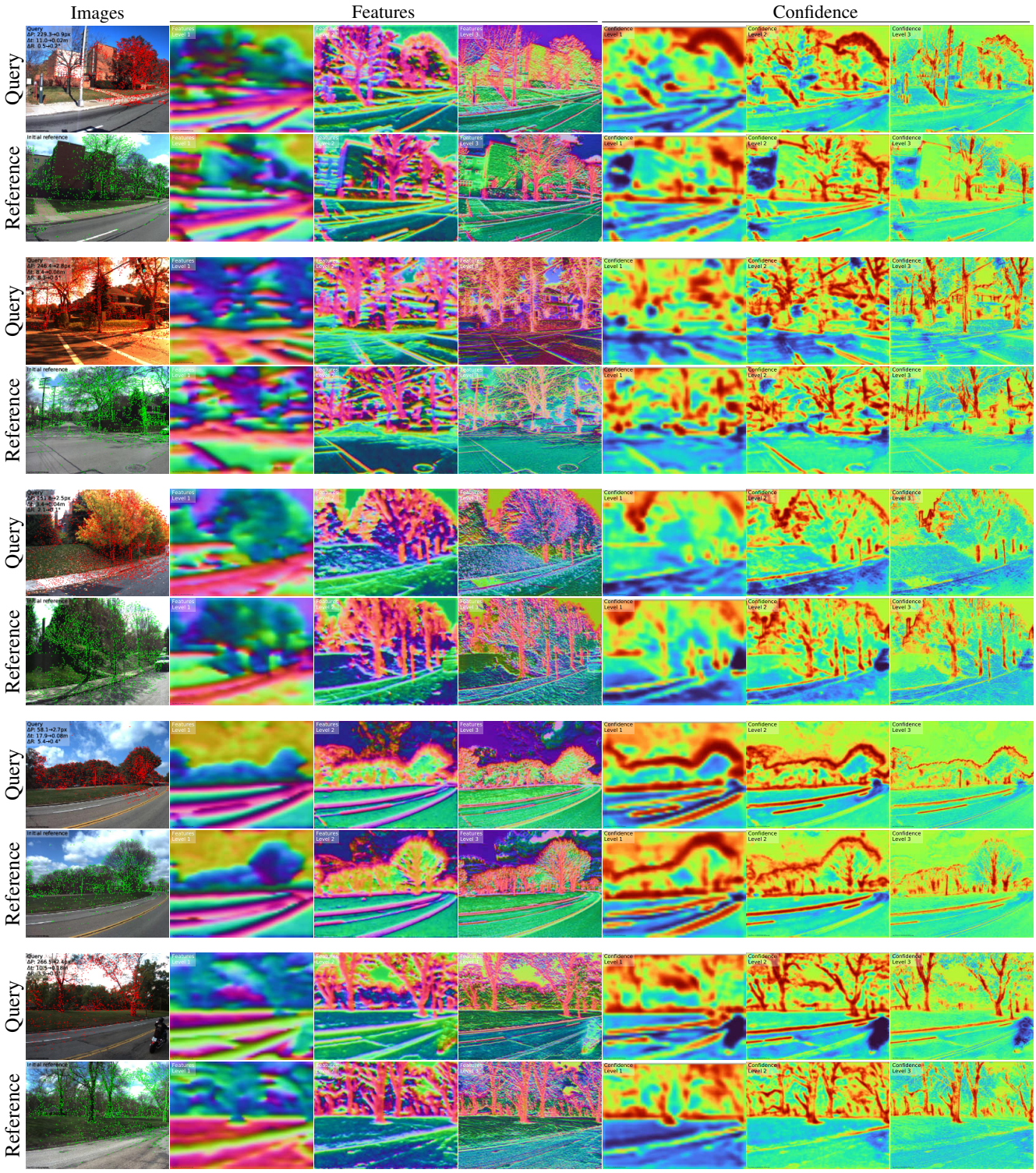

Figure 10. **Successful localization on the CMU dataset.** We show 5 challenging queries with large initial errors and large cross-season appearance changes that are successfully localized by PixLoc. We project 3D SfM points into the initial reference image (in green) and into the query image using the estimated pose (in red). We show the features at the 3 different levels, mapping them to RGB using PCA. We also show the confidence maps, where blue pixels are ignored while red ones are more important for the optimization. Features useful for localization are invariant across seasons and thus appear in similar colors.

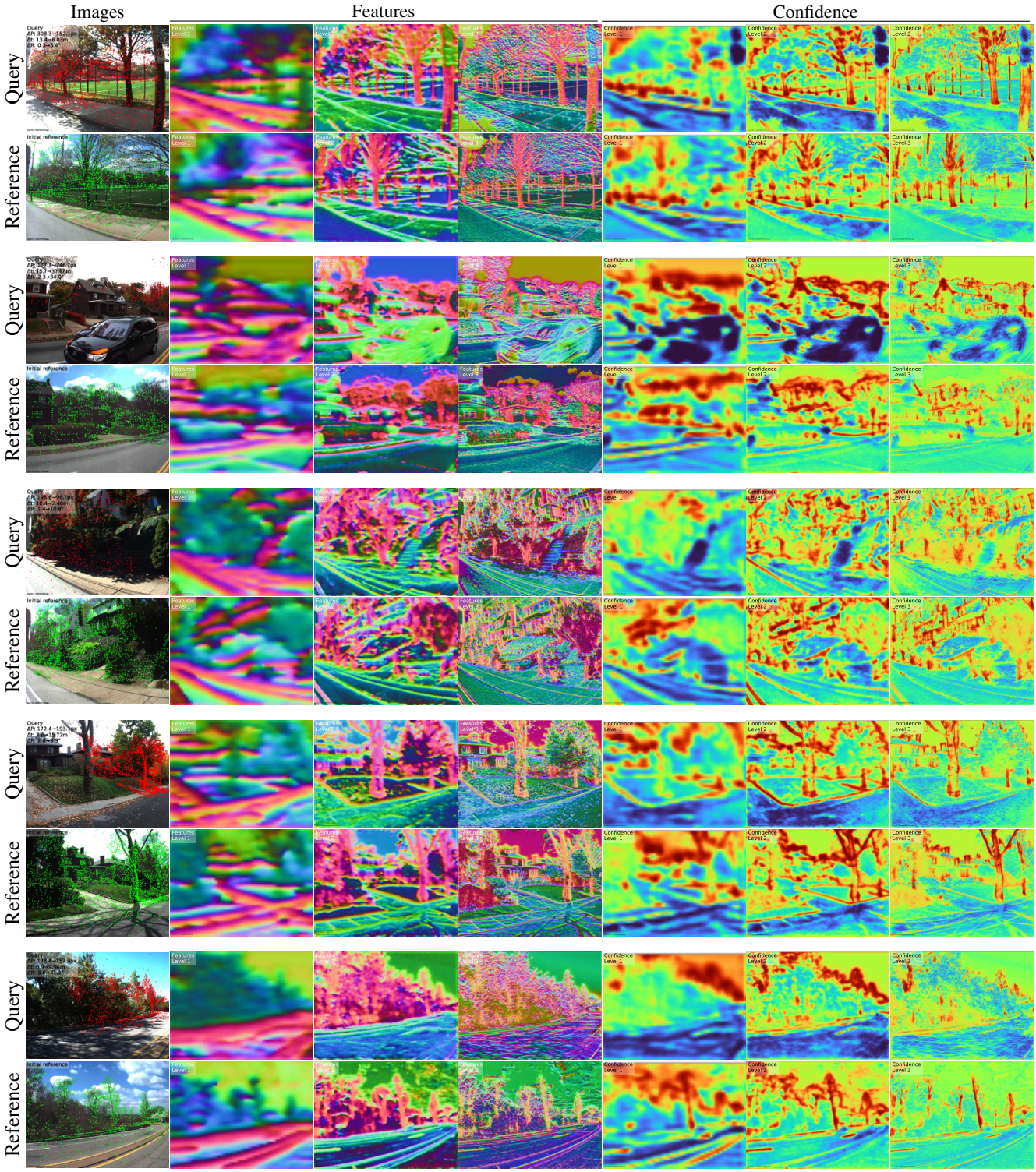

Figure 11. **Failure cases on the CMU dataset.** We show examples for which the optimization results in a large final error. This is often due to repeated elements or to a lack of spatial context of the coarse features or a lack of distinctive elements. Natural scenes are be particularly challenging when tree trunks and vegetation cannot be easily distinguished.

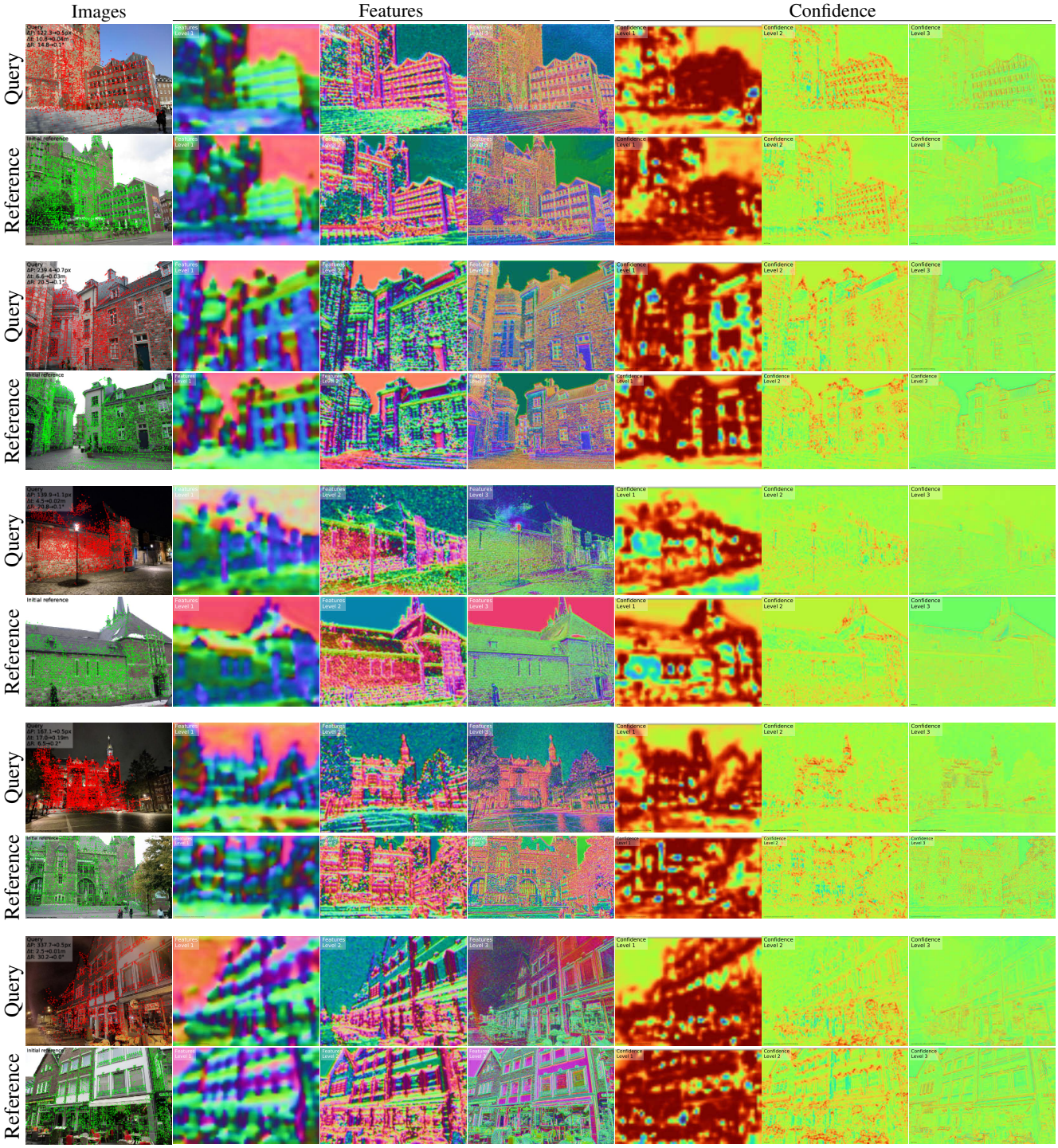

Figure 12. **Successful localization on the Aachen dataset.** We show 5 challenging queries with large initial errors and large day-night appearance changes that are successfully localized by PixLoc. The reprojection and pose errors are computed with respect to the pose estimated by hloc.

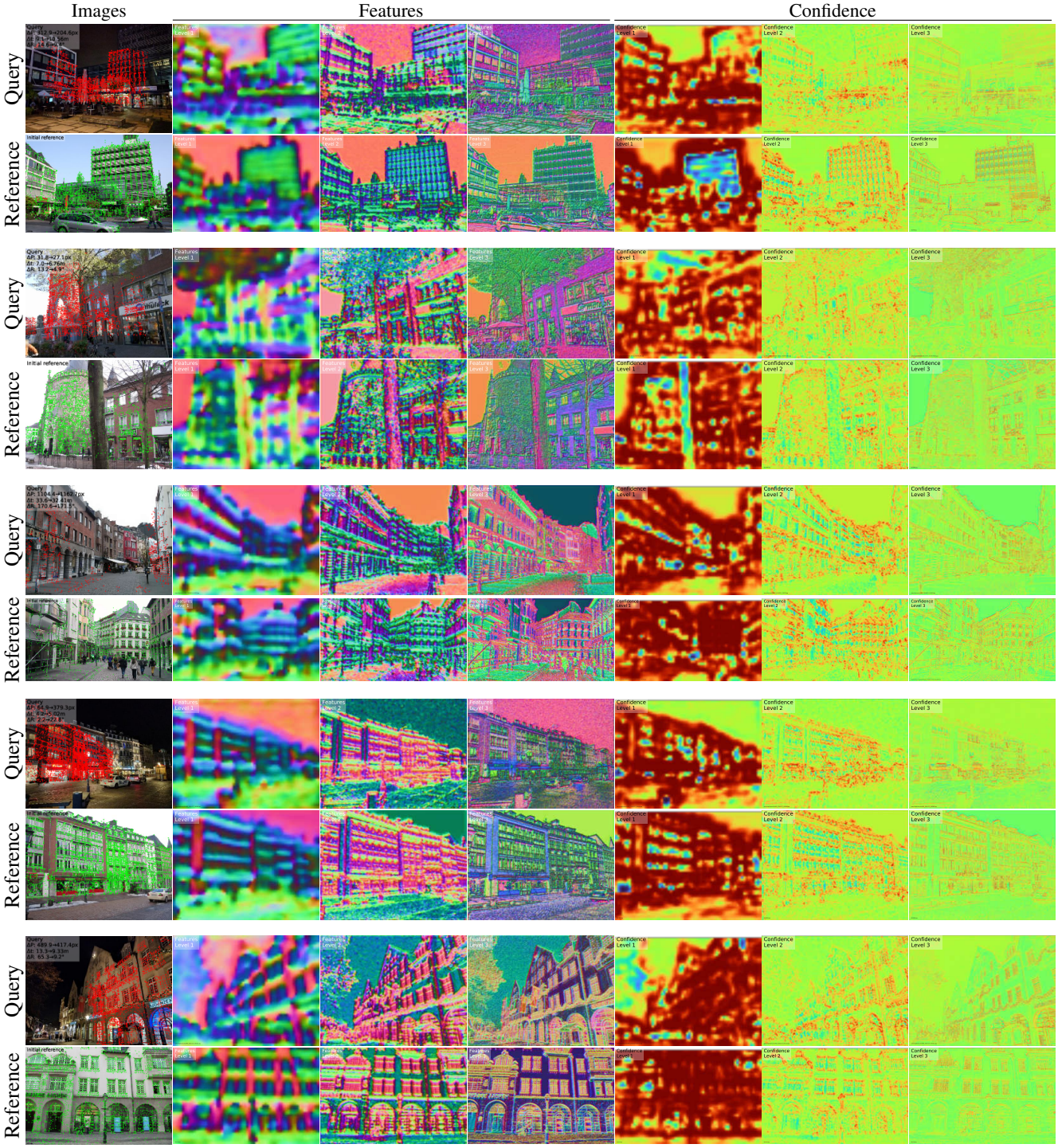

Figure 13. **Failure cases on the Aachen dataset.** Convergence to a local and incorrect minima can be due to large appearance changes (row 1), occlusion (row 2), large viewpoint change (row 3) or repeated structures on facades (rows 4 and 5).

Figure 14. **Convergence basin.** We show the convergence basins of individual selected points given cross-season query and reference images from the CMU dataset. The last row shows smaller basins due to repeated patterns like poles or tree silhouettes.

References

[1] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. Rolling Shutter Absolute Pose Problem With Known Vertical Direction. In *CVPR*, 2016. 1

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

[3] Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium*, pages 794–799. IEEE, 2011. 6, 7

[4] Simon Baker, Ralph Gross, and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56, 2003. 3, 4

[5] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous metric learning relocalisation using neural nets. In *ECCV*, 2018. 1, 2

[6] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *CVPR*, 2019. 1

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 2

[8] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *CVPR*, 2020. 1, 5

[9] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 1, 2

[10] Eric Brachmann and Carsten Rother. Learning Less is More-6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 1

[11] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 1, 2, 7

[12] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 1

[13] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *TPAMI*, 2021. 1, 2, 6, 10

[14] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *CVPR*, 2008. 1

[15] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *ECCV*, 2020. 2

[16] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In *3DV*, 2019. 1, 2

[17] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *CVPR*, 2017. 1, 2

[18] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating PnP optimization. In *CVPR*, 2020. 2

[19] Ondřej Chum and Jiří Matas. Optimal Randomized RANSAC. *TPAMI*, 30(8):1472 –1482, 2008. 1, 3

[20] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, pages 236–243. Springer, 2003. 3

[21] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. LS-Net: Learning to solve nonlinear least squares for monocular stereo. In *ECCV*, 2018. 2, 4

[22] Jan Czarnowski, Stefan Leutenegger, and Andrew J. Davison. Semantic texture for robust dense tracking. In *ICCV Workshops*, 2017. 2, 3

[23] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018. 1, 2, 6

[24] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera relocalization. In *ICCV*, 2019. 1, 2, 6

[25] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019. 1, 2, 7, 11

[26] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *TPAMI*, 40(3):611–625, 2017. 2, 3, 4

[27] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, 2014. 2, 3

[28] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 3

[29] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. In *ECCV*, 2020. 1, 7

[30] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*. Wiley, 1986. 4

[31] R.M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 13(3):331–356, 1994. 1

[32] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 2

[33] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 2

[34] Eskil Jörgensen, Christopher Zach, and Fredrik Kahl. Monocular 3D object detection and box fitting trained end-to-end using intersection-over-union loss. *arXiv:1906.08070*, 2019. 2

[35] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 1, 2, 5

[36] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017. 4

[37] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *ICCV*, 2015. 1, 2, 6

[38] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for RGB-D cameras. In *IROS*, 2013. 3

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 11

[40] L Kneip, D Scaramuzza, and R Siegwart. A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. In *CVPR*, 2011. 1

[41] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In *ICCV*, 2013. 1

[42] Måns Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization. In *ICCV*, 2019. 1, 2

[43] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *ICCV Workshops*, 2017. 1, 2

[44] Karel Lebeda, Juan E. Sala Matas, and Ondřej Chum. Fixing the Locally Optimized RANSAC. In *BMVC*, 2012. 1

[45] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 4

[46] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 1, 6

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

[48] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6

[49] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2D-3D matching for camera localization in a large-scale 3D map. In *ICCV*, 2017. 2

[50] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2, 6

[51] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 3, 4

[52] Zhaoyang Lv, Frank Dellaert, James M Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *CVPR*, 2019. 2, 3, 4, 7

[53] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *IJRR*, 39(9):1061–1084, 2020. 1

[54] Wei-Chiu Ma, Shenlong Wang, Jiayuan Gu, Sivabalan Manivasagam, Antonio Torralba, and Raquel Urtasun. Deep feedback inverse problem solver. In *ECCV*, 2020. 2, 4

[55] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *IJRR*, 36(1):3–15, 2017. 7

[56] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for non-linear least squares problems. 2004. 4

[57] F Landis Markley, Yang Cheng, John L Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007. 9

[58] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 4

[59] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-DOF localization on mobile devices. In *ECCV*, 2014. 2

[60] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 1, 2

[61] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-DoF global localization in outdoor environments. In *IROS*, 2017. 2

[62] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *CVPR*, 2017. 2

[63] Seonwook Park, Thomas Schöps, and Marc Pollefeys. Illumination Change Robustness in Direct Visual SLAM. In *ICRA*, 2017. 2, 3

[64] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5, 10

[65] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *ECCV*, 2020. 1

[66] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking image retrieval for visual localization. In *3DV*, 2020. 7, 9

[67] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *TPAMI*, 2018. 2

[68] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *RA-L*, 3(4):4407–4414, 2018. 2

[69] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noé Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1, 2

[70] Chris Russell, Matteo Toso, and Neill Campbell. Fixing implicit derivatives: Trust-region based learning of continuous energy functions. In *NeurIPS*, 2019. 2

[71] Paul-Edouard Sarlin. Visual localization made easy with hloc. https://github.com/cvg/Hierarchical-Localization/. 5

[72] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 2, 5, 6, 7, 9, 10

[73] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 6, 10

[74] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *CVPR*, 2016. 5

[75] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *TPAMI*, 39(9):1744–1756, 2016. 1, 2, 5, 6, 7, 9

[76] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 6, 7

[77] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 2, 7

[78] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *CVPR*, 2019. 1, 2, 6, 7

[79] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5

[80] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, 2018. 1, 2, 7

[81] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 5

[82] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 1, 2, 6

[83] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *TPAMI*, 2017. 1, 2

[84] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *TPAMI*, 2019. 1, 2, 5, 6, 7

[85] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. In *ICLR*, 2019. 2, 3, 4

[86] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. Long-Term Visual Localization Revisited. *TPAMI*, pages 1–1, 2020. 1, 6, 7

[87] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic Match Consistency for Long-Term Visual Localization. In *ECCV*, 2018. 1

[88] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 2, 6, 7

[89] Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, and Torsten Sattler. Are large-scale 3D models really necessary for accurate visual localization? *TPAMI*, 2019. 2

[90] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. GN-Net: The Gauss-Newton loss for multi-weather relocalization. *RA-L*, 5(2):890–897, 2020. 2, 3, 5, 7, 11

[91] Lukas Von Stumberg, Patrick Wenzel, Nan Yang, and Daniel Cremers. LM-Reloc: Levenberg-Marquardt based direct visual relocalization. In *3DV*, 2020. 2, 5

[92] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *ICCV*, 2017. 1, 2

[93] Chaoyang Wang, Hamed Kiani Galoogahi, Chen-Hsuan Lin, and Simon Lucey. Deep-LK for efficient adaptive object tracking. In *ICRA*, 2018. 2, 3

[94] Binbin Xu, Andrew J Davison, and Stefan Leutenegger. Deep probabilistic feature-metric tracking. *RA-L*, 6(1):223–230, 2020. 2, 4

[95] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *ICCV*, 2019. 1, 2, 6

[96] Tsun-Yi Yang, Duy-Kien Nguyen, Huub Heijnen, and Vassileios Balntas. UR2KiD: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. *arXiv:2001.07252*, 2020. 1

[97] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016. 2

[98] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *ICCV*, 2015. 2, 5

[99] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To Learn or Not to Learn: Visual Localization from Essential Matrices. In *ICRA*, 2020. 1, 2