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

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 sceneagnostic 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.

Keywords: 3D Geometry, Robust Visual Features, Geometric Estimation.

1 Introduction

In "Back to the Feature: Learning Robust Camera Localization from Pixels to Pose," the authors introduce PixLoc, a novel approach leveraging neural networks for accurate camera pose estimation. This method is distinct in its ability to function independently of specific scenes, focusing on learning durable and invariant visual features. PixLoc uniquely combines the strengths of deep learning with principled geometric estimation algorithms, showcasing exceptional adaptability to new environments. This paper marks a significant stride in 3D geometry and camera localization, pushing the boundaries of how visual data and geometric principles can coalesce for advanced spatial understanding.

2 Related works

2.1 Accurate visual localization

commonly relies on estimating correspondences between 2D pixel positions and 3D scene coordinates. Such approaches detect, describe, and match [1] local features, maintain an explicit sparse 3D

representation of the environment, and sometimes leverage image retrieval [8] to scale to large scenes [2]. Recently, many of 1these components have been learned with great success [4], 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.

2.2 Learning-based Approaches

The paper discusses various deep learning methods for feature extraction and matching, highlighting the challenges in training feature matching pipelines end-to-end due to their complexity.

2.3 End-to-end learning for localization

This method recently received much attention. Common approaches encode the scene into a deep network by regressing from an input image to an absolute pose [3]or 3D scene coordinates [10]. Pose regression lacks geometric constraints and thus does not generalize well to novel viewpoints or appearances [7], while coordinate regression is more robust. Both do not scale well due to the limited network capacity [6] and require for each new scene either costly retraining or adaptation. ESAC 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 image, 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. Closer to ours, SANet 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. 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 model.

2.4 Learning camera pose optimization

This one can be tackled by unrolling the optimizer for a fixed number of steps [9], computing implicit derivatives [5], or crafting losses to mimic optimization steps [11]. Multiple works have proposed to learn components of these optimizers. Some of these formulations optimize reprojection errors over sparse points, while others use direct objectives for (semi-)dense image alignment.

3 Method

3.1 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 1). The CNN and the optimization parameters are trained end-to-end from ground truth poses.

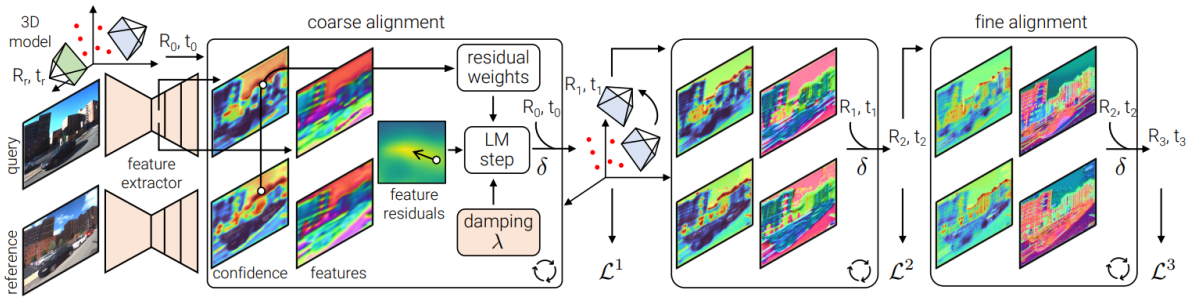

Figure 1. Overview of the method

3.2 Localization as image alignment

The approach uses a convolutional neural network (CNN) to regress geometric quantities like camera poses or 3D scene coordinates corresponding to each pixel. While these methods can be trained end-to-end, they have drawbacks such as being scene-specific and requiring training or adaptation for new scenes. Generalizing to new viewing conditions, such as localizing night-time images with training only on daytime photos, and handling larger, more complex scenes, remain open challenges for such approaches. Absolute or relative pose regression has limited accuracy and often fails to generalize to new viewpoints. Regressing poses relative to a set of reference images is theoretically scene-agnostic, but generalization to significantly different scenes without a significant drop in pose accuracy has not been adequately demonstrated.

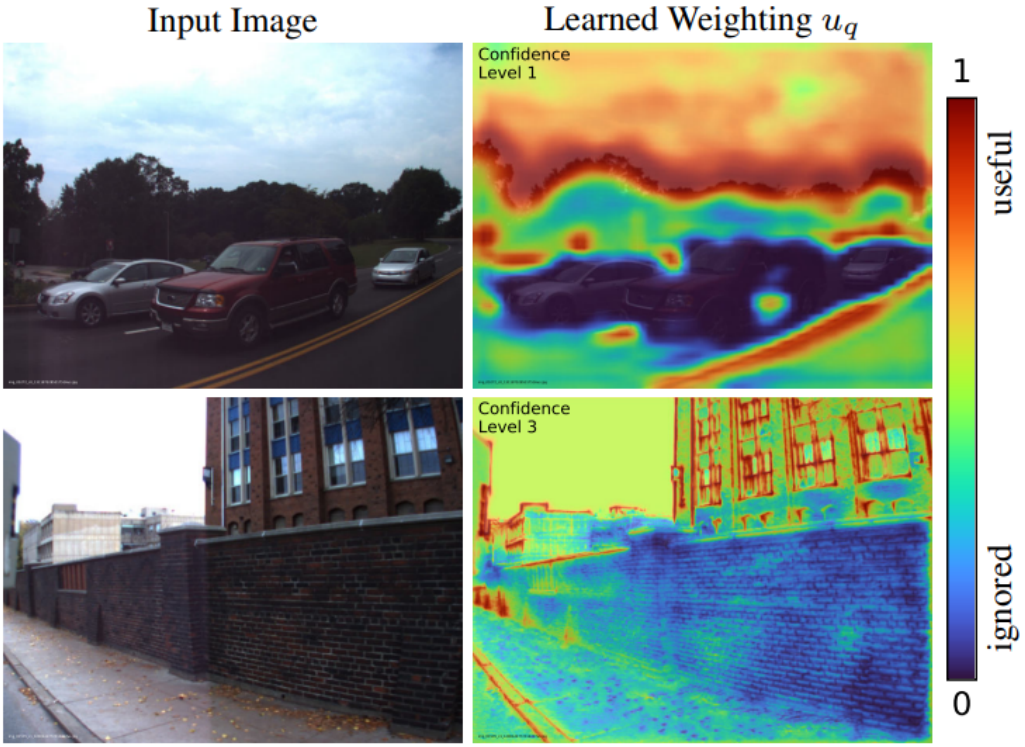

Figure 2. Good features to localize

3.3 Learning from poses

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

$$\frac{1}{L}\sum_{l}\sum_{i}\left|\left|\prod(R_l P_i + tl) - \prod(\bar{R}P_i + \bar{t})\right|\right|^{\gamma}$$

4 Implementation details

4.1 Comparing with the released source codes

The related source codes have been provided by the authors of the paper, and they can be downloaded from the following link https://github.com/cvg/pixloc. In the process of reproducing the code, I utilized the Cambridge ShopFacade dataset provided by the author. However, during reproduction, it became apparent that the code lacked a robust visualization process for the 3D reconstruction model. It solely relied on pre-existing 3D models as input parameters, making it difficult to ascertain their accuracy and reliability. To address this issue, I introduced a new function in my reproduction work to enable comprehensive visualization of the 3D model. This function allows us to choose between outputting either sparse or dense reconstruction models and export them as corresponding ply files for storage.

4.2 Experimental environment setup

In order to achieve the visualization function of 3D models, we need to correctly configure colmap software in the experimental environment. The colmap can be downloaded from the following link http://colmap.github.io.

4.3 Interface design

Get Mesh

```
[13]: import pixloc.getMesher as mesh

# 图像路径
image_path=Path('../datasets/' + dataset + '/ShopFacade/')
# 输出路径
output_path=Path('../outputs/hloc/' + dataset + '/ShopFacade/')

# 分别导出稀疏点云的txt文件和ply文件
mesh.bin2_txt(output_path)
mesh.bin2_ply(output_path)

# 图片最大尺寸
max_image_size=8000
# 立体匹配的窗口的大小
window_radius=5
# 最低NCC阈值
filter_min_ncc=0.1

# 图像畸变矫正
mesh.undistorter(image_path,output_path,max_image_size)
# 立体块匹配
mesh.patch_match_stereo(output_path,max_image_size,window_radius,filter_min_ncc)
# 点云融合
mesh.stereo_fusion(output_path)
# 分别用泊松重建和Delaunay三角化来进行表面重建并导出ply文件
mesh.poisson_mesher(output_path)
mesh.dulaunay_mesher(output_path)
```

Figure 3. Interface design

4.4 Main contributions

A new function has been added to the code: 3D model visualization. This function enables the export of 3D reconstruction model ply files for experiments, so as to conduct visualization display in relevant software (such as cloudcompare). The following visualization video can be found in the appendix.

稀疏重建模型model.ply在CloudCompare上的预览效果

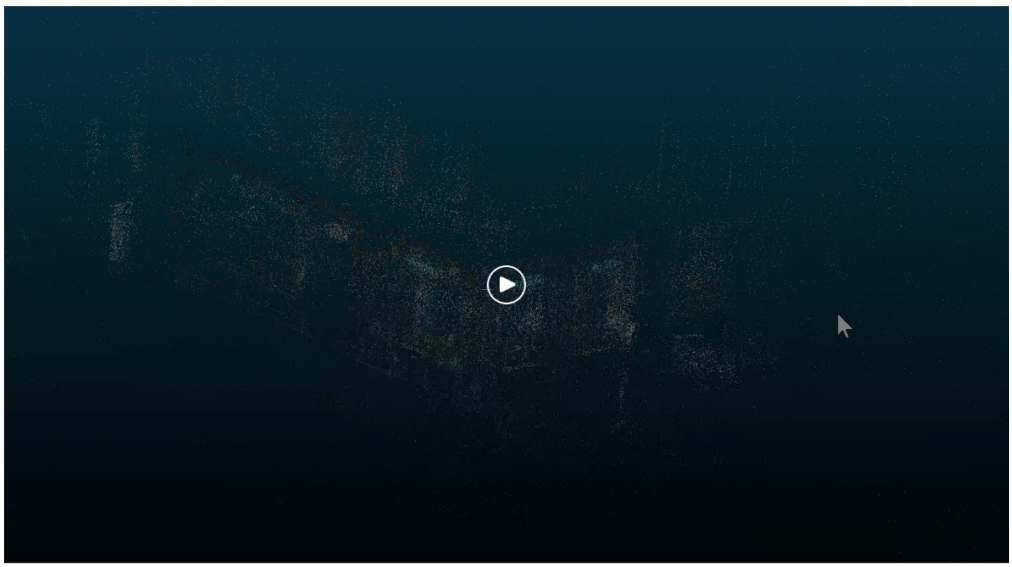

Figure 4. sprase model

稠密重建模型dense.ply在CloudCompare上的预览效果

Figure 5. dense model

泊松重建模型poisson_meshed.ply在CloudCompare上的预览效果

Figure 6. poisson model

Delaunay三角化模型delaunay_meshed.ply在CloudCompare上的预览效果

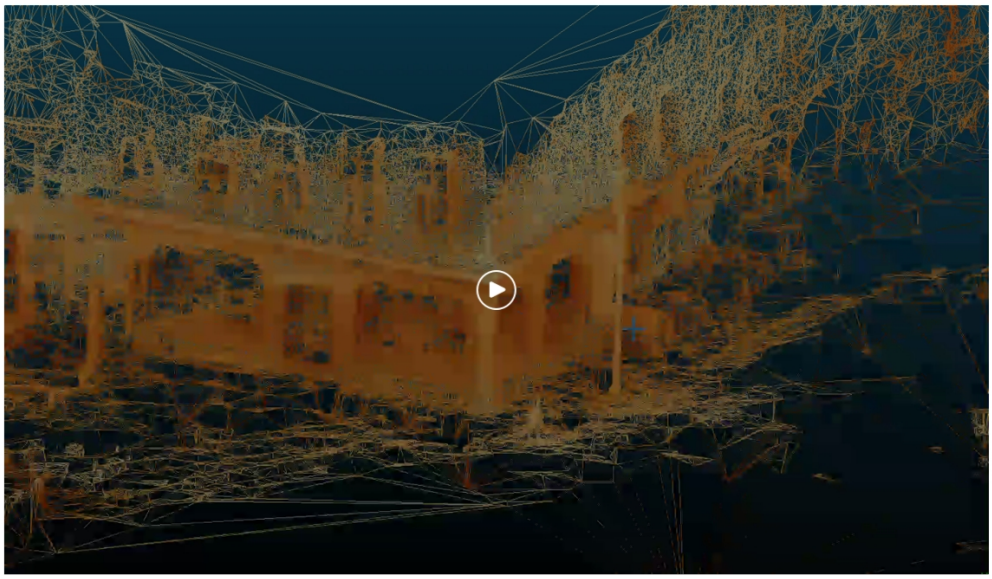

Figure 7. delaunay model

5 Results and analysis

The following result video can be found in the appendix

Generate a video

```
[7]: T_w2q_all = torch.stack(tracker.T)
     p2d_q_all, mask_q_all = cam_q.world2image(T_w2q_all * p3d)
     keep = subsample_steps(T_w2q_all, p2d_q_all, mask_q_all, cam_q.size.numpy())
     print(f'Keep {len(keep)}/{len(p2d_q_all)} optimization steps for visualization.')

     /tmp/ipykernel_40655/282232336.py:1: UserWarning: Defining your `__torch_function__` as a plain method is deprecated and will be an error in future, ple
     ase define it as a classmethod. (Triggered internally at /home/conda/feedstock_root/build_artifacts/pytorch-recipe_1675740251786/work/torch/csrc/utils/p
     ython_arg_parser.cpp:350.)
       T_w2q_all = torch.stack(tracker.T)
     Keep 37/82 optimization steps for visualization.

[8]: video = 'pixloc.mp4'
     writer = VideoWriter('./tmp')
     for i in tqdm(keep):
         plot_images([image_r, image_q], dpi=100, autoscale=False)
         plot_keypoints([p2d_r[mask_r], p2d_q_all[-1][mask_q_all[-1]]], 'lime')
         plot_keypoints([None, p2d_q_all[i][mask_q_all[i]]], 'red')
         add_text(0, 'reference')
         add_text(1, 'query')
         writer.add_frame()
     writer.to_video(video, duration=4, crf=23)  # in seconds

     display_video(video)

     100%|██████████| 37/37 [00:46<00:00,  1.26s/it]
     [12/08/2023 14:59:05 pixloc.visualization.animation INFO] Running ffmpeg.
```

Figure 8. generate a video

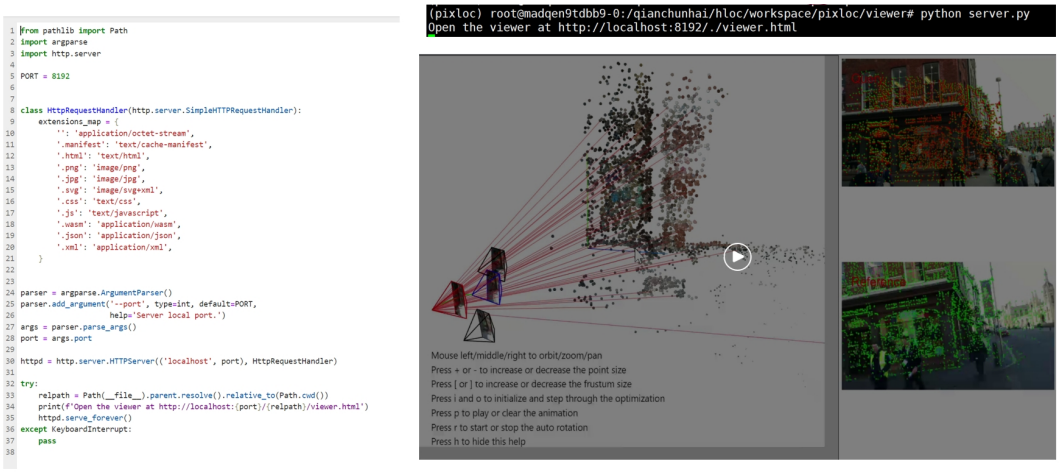

Figure 9. 3D result

We report the results in Figure 10 and Figure 11. 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

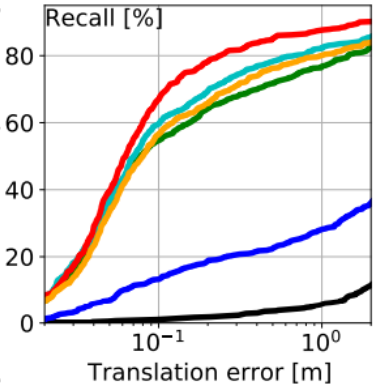

Figure 10. Ablation study. Unrolling the optimizer and learning features, damping factor, and confidences all contribute to the performance of PixLoc over classical photometric alignment. Learning compact features as in past works results in a drop of performance compared to high-dimensional representations.

	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	-
	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
	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
	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**
	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
	CAMNet [24]	-	-	-	-	-	4/1.73	3/1.74	5/1.98	4/1.62	**4**/1.64	4/1.63	4/1.51	-
	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
	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
	+ 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

Figure 11. Large-scale localization on the Aachen, RobotCar, and CMU datasets.

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. PixLoc can also fail for large outliers ratios due prominent occluders and is more sensitive to camera miscalibration.

6 Conclusion and future work

In this paper, authors have introduced a simple solution to end-to-end learning of camera pose estimation. In contrast to previous approaches that regress geometric quantities, they do not try to teach a deep network basic geometric principles or 3D map encoding. Instead, they go Back to the Feature: they show that learning robust and generic features is sufficient for accurate localization by leveraging classical image alignment with existing 3D maps. 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 finetuning. Yet, it lacks of some visualization function for 3D models and solely rely on pre-existing 3D models as input parameters, making it difficult to ascertain their accuracy and reliability. In order to address this issue, I introduced a new function which can comprehensive visualization of the 3D model. However, the process of visualization of this function is relatively long, because the 3D reconstruction using colmap requires a lot of time, so the future work is to minimize the time spent on visualization.

References

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

[2] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. TPAMI, 2021.

[3] Sebastian Nowozin Jamie Shotton Frank Michel Stefan Gumhold Eric Brachmann, Alexander Krull and Carsten Rother. Dsac-differentiable ransac for camera localization. CVPR, 2017.

[4] Cordelia Schmid Herve Jegou, Matthijs Douze and Patrick Perez. Aggregating local descriptors into a compact image ´ representation. CVPR, 2010.

[5] Cordelia Schmid Herve Jegou, Matthijs Douze and Patrick Perez. Aggregating local descriptors into a compact image ´ representation. CVPR, 2010.

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

[7] Qadeer Khan Lukas Von Stumberg, Patrick Wenzel and Daniel Cremers. Gn-net: The gauss-newton loss for multiweather relocalization. RA-L, 2020.

[8] Akihiko Torii Tomas Pajdla Relja Arandjelovic, Petr Gronat and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. CVPR, 2016.

[9] Jan Czarnowski Stefan Leutenegger Ronald Clark, Michael Bloesch and Andrew J Davison. Ls-net: Learning to solve nonlinear least squares for monocular stereo. ECCV, 2018.

[10] Ole Untzelmann Sven Middelberg, Torsten Sattler and Leif Kobbelt. Scalable 6-dof localization on mobile devices. ECCV, 2014.

[11] Marc Pollefeys Torsten Sattler, Qunjie Zhou and Laura LealTaixe. Understanding the limitations of cnn-based absolute camera pose regression. CVPR, 2019.