Neural Refinement for Absolute Pose Regression with Feature Synthesis

Shuai Chen[1] Yash Bhalgat[2] Xinghui Li[1] Jia-Wang Bian[1]
Kejie Li[1] Zirui Wang[1] Victor Adrian Prisacariu[1]
[1]Active Vision Lab, University of Oxford
[2]Visual Geometry Group, University of Oxford

1. Supplementary

1.1. Implementation Details

1.1.1 Architecture details

The model is trained with the re-aligned and re-centred poses in SE(3), as in [14]. We use a coarse-to-fine sampling strategy with 64 sampled points per ray in both stages. The width of the MLP layers is 128 and we output $N_c = 3$ and $N_f = 128$ in the last layer of the fine stage MLP. For the exposure-adaptive ACT module, we compute the query image's histogram $\mathbf{y}_I$ in YUV color space and bin the luminance channel into $N_b = 10$ bins. We then feed the binned histogram to 4-layer MLPs with a width of 32. The exposure-adaptive ACT module outputs the exposure compensation matrix $\mathbf{K}$ and the bias $\mathbf{b}$, which directly transform the integrated colors $\hat{\mathbf{C}}_{NFS}(\mathbf{r})$ of the main networks, with negligible computational overhead. We run the APR refinement process for m iterations per image using the direct feature matching loss $\mathcal{L}_{feature}$ with a learning rate of 1×10^{-5}. Our default value for m is 50 unless specified, denoted as NeFeS$_{50}$. The NeFeS model renders features with a shorter side of 60 pixels and then upsample them using bicubic interpolation to 240 for feature matching.

1.1.2 Progressive training schedule

The training process for the NeFeS network starts with the photometric loss only for $T_1 = 600$ epochs by setting $\lambda_1 = \lambda_2 = 0$ in Eq. (4). The color and density components of the model are trained with a learning rate of 5×10^{-4} which is exponentially decayed to 8×10^{-5} over 600 epochs. We randomly sample 1536 rays per image and use a batch size of 4. After 600 epochs, we reset the learning rate to 5×10^{-4} and switch on the feature loss ($\mathcal{L}_f$ in Eq. (6)) for the next $T_2 = 200$ epochs with $\lambda_1 = 0.04, \lambda_2 = 0$. The fusion loss ($\mathcal{L}_{fusion}$ in Eq. (7)) is switched on for the last $T_3 = 400$ epochs with coefficients $\lambda_1 = 0.02, \lambda_2 = 0.02$. During the third training stage T_3, instead of randomly sampling image rays, we randomly sample $N_{crop} = 7$ image patches of size $S \times S$ where $S = 16$. To extract image features (i.e. $\mathbf{F_{img}}(I, \cdot)$) as pseudo-groundtruth, we use the finest-level features from DFNet's [7] feature extractor module. We resize the shorter sides of the feature labels to 60.

Dataset	PoseNet	+ Ours	MS-Trans.	+ Ours	DFNet	+ Ours
7-Scenes	pose error in m/°					
Chess	0.10/4.02	0.04/1.35	0.11/6.38	0.06/1.96	0.03/1.12	0.02/0.57
Fire	0.27/10.0	0.03/1.20	0.23/11.5	0.06/2.55	0.06/2.30	0.02/0.74
Heads	0.18/13.0	0.12/7.91	0.13/13.0	0.09/6.19	0.04/2.29	0.02/1.28
Office	0.17/5.97	0.02/0.72	0.18/8.14	0.05/1.69	0.06/1.54	0.02/0.56
Pumpkin	0.19/4.67	0.06/1.57	0.17/8.42	0.07/1.85	0.07/1.92	0.02/0.55
Kitchen	0.22/5.91	0.02/0.68	0.16/8.92	0.08/2.31	0.07/1.74	0.02/0.57
Stairs	0.35/10.5	0.27/6.35	0.29/10.3	0.34/7.64	0.12/2.63	0.05/1.28
Average	0.21/7.74	**0.08/2.83**	0.18/9.51	**0.11/3.46**	0.06/1.93	**0.02/0.79**
Cambridge	pose error in m/°					
Kings	1.66/4.86	0.38/0.56	0.83/1.47	0.43/0.59	0.73/2.37	0.37/0.54
Hospital	2.62/4.90	1.15/1.30	1.81/2.39	0.61/1.06	2.00/2.98	0.52/0.88
Shop	1.41/7.18	0.21/0.81	0.86/3.07	0.18/0.98	0.67/2.21	0.15/0.53
Church	2.45/7.96	0.42/1.52	1.62/3.99	0.48/1.53	1.37/4.03	0.37/1.14
Average	2.04/6.23	**0.54/1.05**	1.28/2.73	**0.43/1.04**	1.19/2.90	**0.35/0.77**

Table 1. **Pose refinement on different APR architectures.** Our refinement method can effectively improve pose estimation results for different APR methods. PoseNet is the classic pose regression architecture. MS-Transformer is denoted as MS-Trans., which combines EfficentNet CNN backbones with transformer blocks. DFNet is a multi-task network that predicts domain invariant features and poses.

1.2. Refinement for Different APRs Full

This is the supplementary full table for Section 4.3 of the main paper (Tab. 1).

1.3. Qualitative Comparisons

In Fig. 2, we qualitatively compare the refinement accuracy of different APR methods - namely PoseNet[9–11], MS-Transformer[27], DFNet [7] - with our method, i.e. DFNet+NeFeS$_{50}$. We can observe that our method produces the most accurate poses (compared to ground-truth) and has a significant improvement over DFNet in different scenes such as fire [1000-1500] and kitchen [1000-1500].

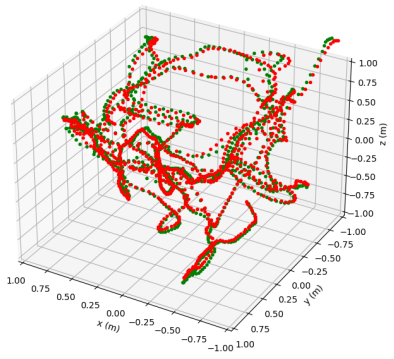


Figure 1. A visualization of camera trajectories of *7-Scene: Chess* scene. The original 'GT' poses are obtained using dSLAM [19] (green). In this paper, we use SfM GT poses provided by [4] (red) for better GT pose accuracy. Two GT trajectories have a median ATE error of 3.5cm/1.46°.

Methods	Average(cm/°)
PoseNet(PN)[11]	44/10.4
PN Learn σ^2[10]	24/7.87
geo. PN[10]	23/8.12
LSTM PN[30]	31/9.85
Hourglass PN[13]	23/9.53
BranchNet[33]	29/8.30
MapNet[5]	21/7.77
Direct-PN[6]	20/7.26
TransPoseNet[26]	18/7.78
MS-Transformer[27]	18/7.28
MS-Transformer+PAE [25]	15/7.28
CoordiNet[16]	22/9.7
CoordiNet+LENS[15]	8/3.0
DFNet [7]	12/3.71
DFNet + NeFeS$_{50}$ (dSLAM)	8/**2.80**
DFNet + NeFeS$_{50}^{--}$ (dSLAM)	**7**/2.87

Table 2. We compare the proposed refinement method using 7-Scene dSLAM GT pose [28] with prior single-frame APR methods, in average of median **translation error (cm)** and **rotation error** (°). Numbers in **bold** represent the best performance.

1.4. 7-Scenes Dataset Details

In Sec. 4.2 of the main paper, we mention the difference between the dSLAM-generated ground-truth pose and the SfM-generated ground-truth pose for the 7-Scenes dataset. We provide more details in this section.

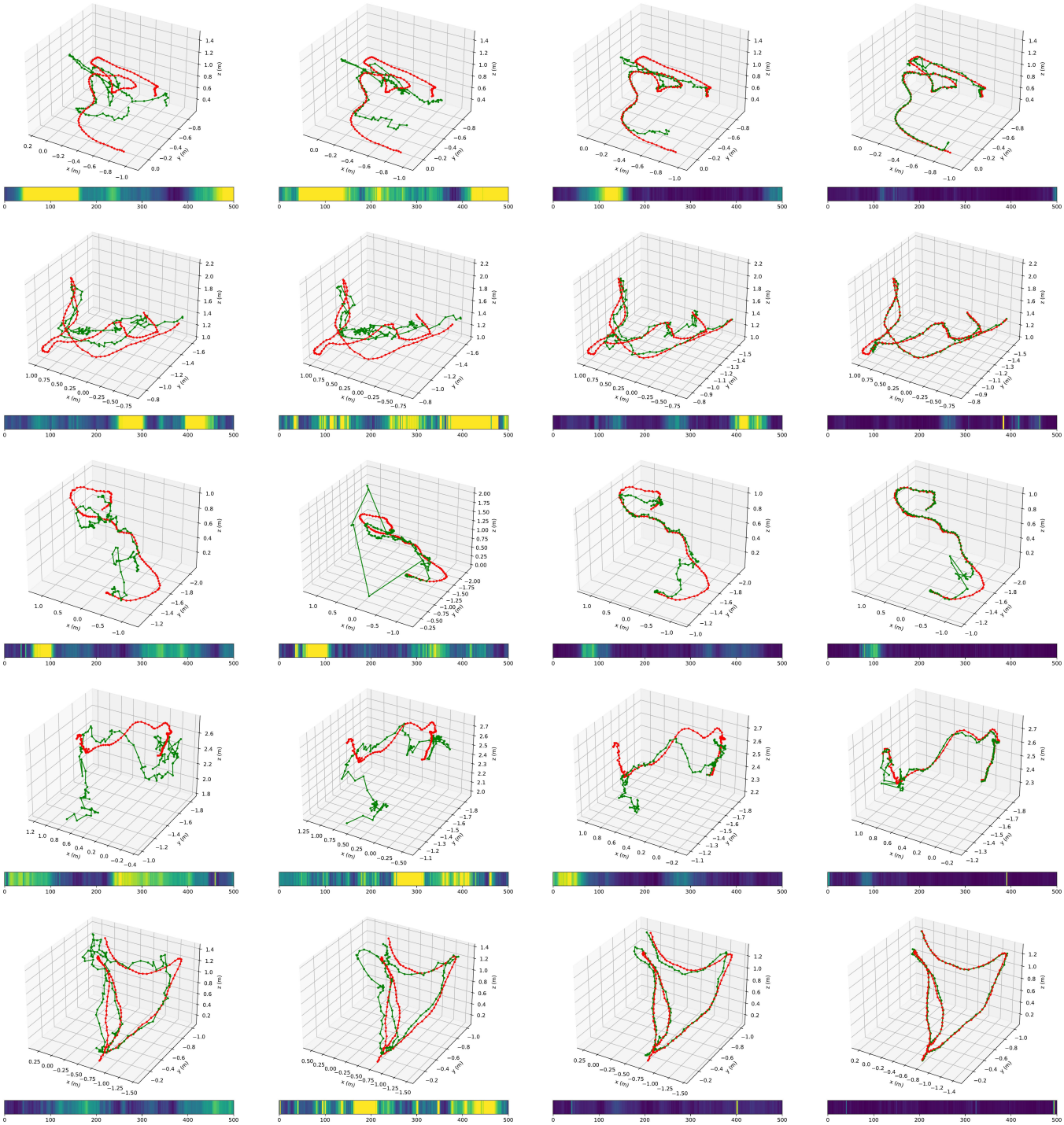
dSLAM vs. SfM GT pose Brachmann *et al.* [4] identified imperfections in the original 'ground-truth' (GT) poses generated by dSLAM in the 7-Scenes dataset. The erroneous GT poses originate from sensor asynchronization between the captured RGB images and depth maps. Therefore, Brachmann et al. employed SfM to regenerate a new set of 'ground-truth' poses, which subsequently aligned and scaled to match the dSLAM-derived poses. As described in Sec. 4.2 of the main paper, we notice that when trained with the SfM ground-truth poses, the rendering quality of NeRF is noticeably boosted compared with using the dSLAM GT poses. The comparison between the trajectories of two sets of ground-truth poses is visualized in Fig. 1. An interesting observation is made based on the results presented in Table 2 of our main paper. We notice DFNet achieves superior performance when trained with SfM-grounded GT data, surpassing its performance as originally reported [7]. This phenomenon may be attributed to utilizing the improved synthetic dataset generated by NeRF during DFNet's *Random View Synthesis* training.

APR Comparisons with dSLAM GT. To supplement Table 2 of the main paper, we compare previous methods and our method when trained and evaluated using dSLAM GT poses. The results can be found in Tab. 2. Note that the pose error is presented in cm/degree to emphasize the distinctions in translational accuracy. Despite NeFeS models being trained using suboptimal dSLAM GT poses in this experiment which reduces the quality of the feature rendering, our model is able to achieve SOTA performance on single-frame APR comparisons. Notably, Coordinet+LENS [15] is the only single-frame APR technique that achieves our method's proximate outcomes (on translational error). However, it's pertinent to note that LENS requires several days to train a high-quality NeRF model per scene. In stark contrast, the NeFeS model requires a much shorter training duration of approximately 5-20 hours, accompanied by an inference speed over 110 times faster and obviated the need for manual parameter tuning, making NeFeS a notably more cost-effective prospect.

Furthermore, we experiment to see if the current dSLAM pose results can be improved if a better quality NeFeS model is used. We performed joint optimization of NeFeS and ground truth camera poses during training using the method introduced in NeRF– – [32]. The outcomes reveal that while the NeFeS model attains an enhanced training PSNR from 23.33dB to 27.88dB and the median translation error improves by $1cm$, the rotation error worsens by $0.07°$ since jointly optimizing the dSLAM GT training poses also slightly shifts the world coordinate system of the radiance fields. This refined model's performance is denoted as DFNet + NeFeS$_{50}^{--}$, as indicated in Tab. 2.

1.5. Comparison with Other Camera Localization Approaches

Although our paper mainly focuses on test-time refinement on single-frame APR methods, it is only one family of approaches in camera relocalization (see our Related Work section). In Tab. 3, we compare geometry-based methods and sequential-based methods for camera localization, as



(a) PoseNet (b) MS-Transformer (c) DFNet (d) **DFNet+NeFeS**$_{50}$

Figure 2. Qualitative comparison on the 7-Scenes dataset. The 3D plots show the camera positions: green for ground truth and red for predictions. The bottom color bar represents rotational errors for each subplot, where yellow means large error and blue means small error for each test sequence. Sequence names from top to bottom are: fire [1000-1500], office [2500-3000], pumpkin [500-1000], kitchen [1000-1500], kitchen [1500-2000]. Since each scene has different numbers of frame, we select 500 frames from each of them and append the range after scene's name.

Family	Method	Cambridge	7-Scenes
Seq. 3D	KFNet[35]	13/0.3	3/0.88
3D	AS[23]	29/0.6	5/2.5
	AS[24]	11/0.3	4/1.2
	DSAC[3]	32/0.8	20/6.3
	DSAC*[2]	15/0.4	3/1.4
	DSAC*[4] (COLMAP)	-	1/0.34
	PixLoc[22] (COLMAP to build 3D model)	11/0.3	3/1.1
	HLoc [21] (COLMAP to build 3D model)	10/0.2	3/1.09
Seq. APR	MapNet+PGO[5]	-	18/6.55
	AtLoc+[31]	-	19/7.08
	TransAPR[20]	94/2.12	17/6.29
	VLocNet [29]	78/2.82	5/3.80
1-frame APR	PoseNet(PN)[11]	204/6.23	44/10.4
	PN Learn σ^2[10]	143/2.85	24/7.87
	geo. PN[10]	163/2.86	23/8.12
	LSTM PN[30]	130/5.51	31/9.85
	Hourglass PN[13]	-	23/9.53
	BranchNet[33]	-	29/8.30
	MapNet[5]	163/3.64	21/7.77
	Direct-PN[6]	-	20/7.26
	TransPoseNet[26]	91/3.50	18/7.78
	MS-Transformer[27]	128/2.73	18/7.28
	MS-Transformer+PAE [25]	96/2.73	15/7.28
	E-PoseNet [18]	94/2.12	17/7/32
	CoordiNet[16]	92/2.58	22/9.7
	CoordiNet+LENS[15]	39/1.15	8/3.0
	DFNet [7]	119/2.90	12/3.71
	DFNet [7] (COLMAP)	-	6/1.93
	DFNet + NeFeS$_{50}$	**35/0.77**	**8/2.80**
	DFNet + NeFeS$_{50}$ (COLMAP)	-	**2/0.79**

Table 3. This table compares different types of camera relocalization on Cambridge Landmarks and 7-Scenes dataset. We show representative methods for each school of approach: geometry-based methods (3D), sequential-based APR methods (Seq. APR), and single-frame APR methods (1-frame APR). We report the average of median **translation error (cm)** and **rotation error (°)**. Numbers in **bold** represent the performance of our methods.

well as adding several other single-frame APR methods, including some without code available publicly to support a more thorough comparison. The results on 7-Scenes dataset are evaluated using the original SLAM ground-truth pose, except methods marked by "(COLMAP)", which indicates the results evaluated using the COLMAP ground-truth pose for 7-Scenes. The methods that marked by "(COLMAP to build 3D model)" indicates COLMAP generated 3D models are used in training and evaluation.

We show that when compared with sequential-based APR methods, our method achieves very competitive results on Cambridge Landmark dataset and 7-Scenes dataset. In addition, for the first time, we show that a single-frame APR method can obtain accuracy of the same magnitude as 3D geometry-based approaches.

1.6. Featuremetric vs. Photometric Refinement

In this section, we study the differences between featuremetric refinement and photometric refinement. Prior literature such as iNeRF [34], NeRF−− [32], BARF [12], GARF [8], and NoPe-NeRF [1], have attempted to 'invert' a NeRF model with photometric loss for pose optimization.

Methods	Hospital
DFNet	2.00m/2.98°
DFNet + Sparse NeRF photometric$_{50}$	1.19m/1.52°
DFNet + Dense NeRF photometric$_{50}$	0.80m/1.12°
DFNet + NeFeS$_{50}$	**0.52m/0.88°**

Table 4. We compare our featuremetric refinement method using the proposed **NeFeS** network with photometric-based refinement baselines on *Cambridge Hospital*.

However, directly comparing our featuremetric method with these methods would not be appropriate due to the following reasons: **Firstly**, these methods [1, 8, 12, 32] optimize both camera and NeRF model parameters simultaneously but are unsuitable for complex scenes with large motion (e.g. 360°scenes) since each frame's camera pose is initialized from an identity matrix. **Secondly**, these methods do not effectively handle exposure variations, resulting in suboptimal rendering quality. **Thirdly**, even with a coarse camera pose initialization, photometric-based inversion methods cannot prevent drifting in refined camera poses, leading to misalignment with the ground truth poses of testing sequences.

Therefore, for a fair comparison with photometric methods, we define a photometric refinement model as the baseline model to compare with. Specifically, for the baseline model, the main architecture from the NeFeS model is maintained but without the feature outputs, and only the RGB colors $\hat{\mathbf{C}}(\mathbf{r})$ are used for photometric pose refinement. The performance of two cases with photometric refinement are demonstrated in Tab. 4: first is a sparse photometric refinement that randomly samples pixel-rays, similar to iNeRF [34] or BARF[12]-like methods; and the other uses dense photometric refinement, which renders entire RGB images. The results indicate that our featuremetric refinement is more robust than all the photometric refinement baselines, as it achieves lower pose errors after 50 iterations of optimization.

1.7. Benefit of splitting lr_R and lr_t

As described in Sec. 3.3 of the main paper, we find using different learning rates for translation and rotation components as beneficial for fast convergence when we directly refine the camera pose parameters. In this section, we use a toy experiment to illustrate how we determine to use this strategy. We select 20% of *Cambridge: Shop Facade*'s test images and perform direct pose refinement for 20 iterations using our NeFeS model. In Tab. 5, we compare our *different* learning rate setting with several cases of *same* learning rate settings. The learning rate $lr_R = lr_t = 0.003$ is used in [8, 12] and $lr_R = lr_t = 0.001$ is used in [32, 34]. We

	LR Settings	Shop-20% (+NeFeS$_{20}$)
Same lr	Initial Pose Error	0.58m/3.14°
	$lr_R = lr_t = 0.1$	0.91m/22.70°
	$lr_R = lr_t = 0.01$	0.49m/1.51°
	$lr_R = lr_t = 0.003$	0.54m/2.44°
	$lr_R = lr_t = 0.001$	0.57m/2.48°
Different lr	$lr_R = 0.01, lr_t = 0.1$	**0.27m/1.77°**

Table 5. We use a toy example to show the benefit of using *different* learning rates over *same* learning rates for translation and rotation components during direct pose refinement. We show four cases for *same* learning rate including two settings that are used in prior works. Our pose refinement results are evaluated by using 20% test data of *Cambridge: Shop Facade* and 20 iterations of optimization.

show that by utilizing a different learning rate strategy, the pose error converges much faster and is more stable for both camera position and orientation.

1.8. Runtime Analysis

Runtime cost. Due to better implementation flexibility, we used an unoptimized version of NeFeS in this study. The pytorch-based NeFeS currently runs at 6.9 fps per image including its backpropagation, which is 3x faster than DFNet's NeRF-Hist [7] and 110x faster than LENS's NeRF-W [15]. It is crucial to emphasize that further optimization can be pursued to attain commercial-level efficiency. For example, NeFeS can potentially be accelerated up to 66x using the C++/CUDA-based `tiny-cuda-nn` and `instant-ngp` [17] frameworks.

Training cost. Our NeFeS can be trained in parallel with the APR method such as DFNet and takes roughly the same time as the underlying APR method (*i.e.* 5-20 hrs depending on scene size). However, the NeFeS model only needs to be trained **once** and the same model can be applied to different APR methods.

1.9. Additional Insight

DFNet features vs. other type of features. We were curious about how NeFeS performs when trained with features other than the DFNet. Thus, we experimented with training the NeFeS model with PixLoc [22] features in our refinement pipeline. While we did find positive results, the refined performance didn't reach that of DFNet features. This is because DFNet is trained to close the domain gap between features extracted from natural query images and features rendered by NeRF.

References

[1] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. NoPe-NeRF: Optimising neural radiance field with no pose prior. In *CVPR*, 2023. 4

[2] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *IEEE TPAMI*, 2021. 4

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 4

[4] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, 2021. 2, 4

[5] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2018. 2, 4

[6] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *3DV*, 2021. 2, 4

[7] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. DFNet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022. 1, 2, 4, 5

[8] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. In *ECCV*, 2022. 4

[9] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 1

[10] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 2, 4

[11] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 2, 4

[12] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 4

[13] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *ICCVW*, 2017. 2, 4

[14] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[15] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by nerf synthesis. In *CoRL*, 2021. 2, 4, 5

[16] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization. In *WACV*, 2022. 2, 4

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 5

[18] Mohamed Adel Musallam, Vincent Gaudillie're, Miguel Ortiz del Castillo, Kassem Al Ismaeil, and Djamila Aouada. Leveraging equivariant features for absolute pose regression. In *CVPR*, 2022. 4

[19] R. Newcombe, A. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 2

[20] Chengyu Qiao, Zhiyu Xiang, Yuangang Fan, Tingming Bai, Xijun Zhao, and Jingyun Fu. TransAPR: Absolute camera pose regression with spatial and temporal attention. In *IEEE Robotics and Automation Letters*, 2023. 4

[21] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 4

[22] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021. 4, 5

[23] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012. 4

[24] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. In *IEEE TPAMI*, 2017. 4

[25] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *ECCV*, 2022. 2, 4

[26] Yoli Shavit, Ron Ferens, and Yosi Keller. Paying attention to activation maps in camera pose regression. In *arXiv preprint arXiv:2103.11477*, 2021. 2, 4

[27] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *ICCV*, 2021. 1, 2, 4

[28] 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. 2

[29] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 2018. 4

[30] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, 2017. 2, 4

[31] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, 2020. 4

[32] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2, 4

[33] J. Wu, L. Ma, and X. Hu. Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In *ICRA*, 2017. 2, 4

[34] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *arxiv arXiv:2012.05877*, 2020. 4

[35] Lei Zhou, Zixin Luo, Tianwei Shen, Jiahui Zhang, Mingmin Zhen, Yao Yao, Tian Fang, and Long Quan. Kfnet: Learning temporal camera relocalization using kalman filtering. In *CVPR*, 2020. 4