Map-Relative Pose Regression for Visual Re-Localization

Supplementary Material

1. Supplementary Ablations

1.1. Impact of auxiliary losses

In Tab. 1 we present an analysis of the impact of incorporating auxiliary losses $\mathcal{L}_{\hat{P}_0}, \mathcal{L}_{\hat{P}_1}$ into our model training protocol, contrasted with the model devoid of such losses. As mentioned in Section 3.3 of the main paper, we found this implementation beneficial to the overall pose regression performance.

Architecture	Accuracy $10cm/5°$	Accuracy $0.5m/5°$
marepo w/ auxiliary losses (Ours)	**47.2%**	**63.4%**
marepo w/o auxiliary losses	45.7%	62.2%

Table 1. Performance of ***marepo*** trained with and without auxiliary losses as in Equation 7 of the main paper.

1.2. Impact of rotation representation: 9D SVD orthogonalization vs. 6D. Gram-Schmidt

Additionally, we investigated the effects of utilizing alternative rotation representations on our model's performance. For example, Levinson *et al.* [8] demonstrated that SVD orthogonalization facilitates a continuous mapping of a 9D representation onto SO(3), potentially improving pose prediction accuracy beyond that achievable with a 6D representation [12] used in our model. We replaced our 6D with the 9D representation and trained the full ***marepo*** models to assess the differences. The findings indicate that, within our model's framework, the prediction accuracy for 9D rotations marginally lags behind that of 6D rotations (Tab. 2), thereby verify our design choice in the paper.

Accuracy	***marepo*** (9D)	***marepo*** (6D)
$10cm/5°$	46.8%	**47.2%**

Table 2. Ablation on rotation representations using the ***marepo*** model. Accuracies are reported on Wayspots.

1.3. Impact of the SCR component

We use two methods to study the impact of the choice of Scene Coordinate Regression component on the pose estimation performance. First, we replaced the pretrained ACE backbone with a VGG network, then retrained the scene-specific SCRs. The SCRs' outputs were then passed to our pretrained pose regressor $\mathcal{M}$. As indicated in Tab. 3, the choice of SCR does indeed affect the pose regressor's accuracy. However, ***marepo*** also displays robustness to the quality of the input scene coordinates, as the overall performance degradation is not large, demonstrating the capability of our approach to predict accurate poses from scene coordinates generated by different means.

Furthermore, we performed quantitative experiments adding random noise to the scene coordinates passed to $\mathcal{M}$. Specifically, we applied randomly generated noise of different magnitudes (up to 10cm, and up to 50cm) to a variable proportion of the scene coordinates. We show that ***marepo*** is able to cope with large proportions of errors in the input coordinates, without significant drops in performance (up to 60% of the coordinates can be perturbed with 10cm noise, and up to 40% for 50cm noise) (see Tab. 4).

Accuracy	ACE backbone SCR + $\mathcal{M}$	VGG backbone SCR + $\mathcal{M}$
$10cm/5°$	47.2%	46.0%

Table 3. Effect of different scene coordinate regression backbones on the accuracy of the downstream regressor $\mathcal{M}$ on Wayspots.

SCR Noise	0%	20%	40%	60%	80%	100%
$10cm$	47.2%	46.9	46.0	44.5	38.2	26.9
$50cm$	47.2%	46.3	43.2	21.1	10.3	0.3

Table 4. Effect of increasing amounts of random noise applied to the SCR predictions. The top row indicates the proportion of the pixels in each scene coordinate map affected by uniform noise with maximum value indicated at the beginning of each row. We report the $10cm/5°$ accuracy on Wayspots.

2. Supplementary Video

To complement our quantitative analysis, we provide a supplementary video offering a qualitative perspective, primarily focusing on visually comparing the predicted camera trajectories. The trajectories are superimposed on point clouds rendered from the respective scenes, providing an intuitive understanding of each method's performance.

The first segment of the video showcases a comparative analysis of our approach against other open-sourced APR-based methods on the 7-Scenes dataset [5, 10], where we compare vs. PoseNet[6, 7] and DFNet(EB0)[4]; and the Wayspots dataset [1, 2], where we compare vs. PoseNet and MS-Transformer[9], note that both train their models in under one day. For PoseNet, we have utilized the PyTorch implementation provided by Chen *et al.*[3].

Moreover, in the second segment of the video, we show that ***marepo*** compares well qualitatively with the Accelerated Coordinate Encoding[2] structure-based method. This comparison demonstrates that our method achieves similar accuracy to ACE, with the added benefit of producing smoother trajectory estimations in certain scenarios. Notably, our approach provides a faster throughput during inference, underscoring its practical applicability in demanding scenarios.

3. Experiments on 12-Scenes dataset

We show experimental results on the 12-Scenes dataset [11] in Tab. 5. We compare ***marepo*** to the baseline APR methods PoseNet and MS-Transformer. Since the original PoseNet code was implemented on Caffe, we used the open-sourced code from [3]. The results show that ***marepo*** significantly outperforms the baseline APRs, which is consistent with the behavior shown in the main paper compared to the benchmark APR approaches.

Scene	PoseNet	MST	***marepo***
Apt.1 Kitchen	14.3%	3.4%	98.0%
Apt.1 Living	11.2%	9.7%	98.6%
Apt.2 Bed	18.1%	2.9%	96.0%
Apt.2 Kitchen	38.6%	13.8%	100%
Apt.2 Living	13.5%	4.6%	99.7%
Apt.2 Luke	9.1%	4.8%	89.4%
Office 1 Gates 362	34.5%	14.0%	97.2%
Office 1 Gates 381	8.1%	4.1%	84.6%
Office 1 Lounge	17.1%	14.1%	93.9%
Office 1 Manolis	13.7%	8.9%	94.8%
Office 2 Floor 5a	5.2%	1.4%	90.5%
Office 2 Floor 5b	5.2%	7.2%	83.5%
Average	13.5%	7.4%	**93.9%**
median error	9.4cm/3.9°	11.1cm/5.5°	**2.6cm/1.3°**

Table 5. Performance on the 12-Scenes [11] dataset. The accuracy is reported as percentage of query frames localized within $5cm/5°$.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022. 1

[2] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023. 1, 2

[3] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *3DV*, 2021. 1, 2

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

[5] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *ISMAR*, 2013. 1

[6] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 1

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

[8] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *NeurIPS*, 2020. 1

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

[10] 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

[11] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*. IEEE, 2016. 2

[12] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 1