Supplementary Material for: Enhancing Scene Coordinate Regression with Efficient Keypoint Detection and Sequential Information

1. KDH Evaluation

To demonstrate the effectiveness of the proposed keypoint detection head (KDH), We compared the relocalization accuracy of our system and our variant that replaces the KDH with SuperPoint (SP) [1]. The comparison is conducted on the 7-Scenes, 12-Scenes, and Cambridge Landmarks datasets. The result is presented in Table I.

TABLE I: Evaluation of our keypoint detection head.

Methods	7-Scenes (%, ↑)	12-Scenes (%, ↑)	Cambridge (cm / °, ↓)	FPS on Jetson
ACE [2]	55.2	77.1	25 / 0.4	8.8
Ours (SP)	**67.5**	**86.1**	23 / **0.4**	6.6
Ours (KDH)	66.2	85.7	**22** / **0.4**	**8.9**

[1] We report the percentage of test frames below a (1cm, 1°) pose error on 7-Scenes and 12-Scenes, as well as the median pose errors on Cambridge Landmarks.

As shown in Table I, using SuperPoint slightly improves the relocalization accuracy of our system compared to KDH, but the improvement is not significant. Our KDH delivers satisfactory performance while substantially improving efficiency. Hence, we consider KDH to be a well-balanced and effective choice for achieving both accuracy and efficiency.

2. Keypoint Number

In our main paper, we empirically select the top 1000 keypoints for localization. To investigate the impact of the keypoint number on our system and whether this number is adaptable or scene-specific, we use different scenes from the 7-Scenes dataset to evaluate the recall under varying numbers of keypoints. The results are presented in Table II.

TABLE II: Runtime and recall (%) at different keypoint numbers.

	300	500	1000	1200	1500	2000
Chess	94.8	95.9	96.5	96.6	96.6	**96.7**
Fire	63.6	67.2	68.8	69.1	69.5	**69.8**
Heads	91.3	**91.9**	91.7	91.7	91.6	91.6
Office	57.2	57.6	59.8	60.2	60.5	**60.7**
Pumpkin	54.2	55.9	57.8	**58.1**	57.3	56.1
Kitchen	70.3	71.5	71.7	72.4	72.8	**73.1**
Stairs	15.6	**18.8**	17.4	17.3	15.7	14.7
FPS	**69.9**	65.8	59.0	52.1	43.9	36.4

It can be observed that the impact of the number of keypoints on recall varies across different scenes. For easy scenes such as *Chess* and *Heads*, changes in the number of keypoints have little effect on recall. This may be because the keypoints in these scenes are generally of high quality and produce consistent results for pose estimation, making it possible to achieve accurate localization with only a small number of keypoints. In contrast, for more challenging scenes like *Stairs* and *Pumpkin*, performance starts to degrade once the number of keypoints exceeds an optimal threshold. This may be because these two sequences contain a large number of repetitive textures or reflective surfaces, resulting in a limited number of distinctive keypoints. In such cases, adding more low-quality keypoints may actually reduce the system's accuracy. For the remaining regular scenes, increasing the number of keypoints tends to improve accuracy, but also significantly reduces efficiency.

3. Comparison with SLAM Methods

In this section, we compare our SCR system with several SLAM systems. Specifically, we select ORB-SLAM3 [3], DROID-SLAM [4], and MASt3R-SLAM [5] as baselines, and evaluate their median pose errors on the 7-Scenes dataset. The results are presented in Table III. Note that we only report translation errors, excluding rotation errors. This is because the trajectories estimated by SLAM systems are typically in coordinate frames that are inconsistent with the ground truth. We use evo [6] to align the estimated and ground-truth trajectories. However, we observed that while evo can effectively align the translations, it fails to align the rotations properly, as discussed in this issue (https://github.com/MichaelGrupp/evo/issues/551).

TABLE III: Comparison of median pose errors (cm) with SLAM systems on the 7-Scenes dataset.

	ORR-SLAM3 [3]	DROID-SLAM [4]	MASt3R-SLAM [5]	Ours
Chess	**0.42**	0.52	4.74	0.45
Fire	**0.57**	1.11	2.22	0.74
Heads	0.84	1.00	1.18	**0.40**
Office	2.23	1.43	6.67	**0.87**
Pumpkin	1.22	1.27	6.29	**0.92**
Kitchen	2.07	1.07	3.42	**0.68**
Stairs	2.58	6.85	**0.82**	2.37
Average	1.42	1.89	3.62	**0.92**
FPS	43	19	15	**59**

It can be seen that our system outperforms current state-of-the-art SLAM methods on the 7-Scenes dataset in terms of both speed and accuracy. This is because our system is a relocalization approach that leverages a prior map to perform drift-free localization. In contrast, SLAM methods estimate poses based on accumulated relative transformations, which

TABLE IV: Comparison on the 7-Scenes dataset. The relocalization recall is the proportion of test frames below a (2cm, 2°) pose error.

		Mapping Time	Map Size	FPS*	Relocalization Recall (%)							
					Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Avg.
FM	GoMatch [7]	1.5h	∼300M	1.23	18.5	9.4	16.8	6.7	3.0	3.9	4.1	8.9
	PixLoc [8]	1.5h	∼1.7G	1.23	94.8	82.4	87.0	83.7	76.5	81.6	27.5	76.2
	HLoc (SP+SG) [9]	1.5h	∼3.5G	0.46	99.4	92.4	95.0	93.2	95.9	91.3	39.2	86.6
SCR	DSAC* [10]	9h	26.3M	55	98.5	91.6	93.7	81.8	90.4	90.2	25.4	81.6
	ACE [2]	5 min	4.1M	56	99.5	92.3	98.3	87.8	88.2	89.7	23.7	82.8
	FocusTune [11]	6 min	4.1M	61	99.3	91.9	98.2	90.9	92.3	91.4	28.9	84.7
	GLACE [12]	18 min	9M	31	99.6	90.6	**98.5**	90.9	91.0	**94.0**	27.4	84.6
	Ours-Single	7 min	4.1M	90	**100**	91.5	96.6	92.3	94.2	90.1	37.6	86.0
	Ours-Sequence	7 min	4.1M	59	**100**	**94.6**	97.8	**94.1**	**97.5**	92.5	**44.8**	**88.8**

* Running time of relocalization measured in frame per second (FPS).

inevitably leads to drift over time. Although loop closure modules can help reduce these accumulated errors to some extent, their effectiveness depends on whether sufficient loops are formed during the trajectory.

4. Comparison with BA

To further demonstrate the efficiency of our sequence-based methods, we compare the runtime of our sequence-based method with that of the bundle adjustment (BA) method, which is a widely used approach in visual localization that leverages sequential information. We select the BA modules in ORB-SLAM3 [3] and AirSLAM [13] as baselines. For ORB-SLAM3, we use the monocular mode with its default configuration. For AirSLAM, we disable the line feature module and retain only point features. We measure the runtime of AirSLAM's BA module under different keyframe (KF) settings, specifically when the number of keyframes involved is 2, 3, 4, and 5, respectively. The results are summarized in Table V, where all methods are tested on a laptop with CPU-only execution. Note that the runtime of our method includes the total time for optical flow tracking, pose estimation, and scene point updating, whereas the runtime of BA does not account for additional processes such as feature tracking and triangulation.

TABLE V: Runtime comparison between our sequential method and the bundle adjustment.

	Method	Runtime (ms)
	Ours	5.95
	ORB-SLAM3 [3]	92.26
AirSLAM	2 KFs	15.94
	3 KFs	31.97
	4 KFs	50.48
	5 KFs (default)	71.08

The results clearly indicate that our system exhibits substantially higher efficiency compared to BA. Even when BA is applied to as few as 2 frames, its runtime surpasses the entire processing time of our SCR system. Therefore, integrating BA into an SCR system may significantly reduce the efficiency. In contrast, our method is able to significantly improve the accuracy of the SCR system while maintaining its high efficiency. Hence, our approach is more aligned with the goal of maintaining the high efficiency of SCR systems.

5. Additional Metrics

In this section, we evaluate the system using additional metrics. Specifically, Table IV presents the relocalization recall of each system under a 2cm/2° threshold, as a supplement to Table I of the main text. Table VI reports the median relocalization errors of our system on the 7-Scenes and 12-Scenes datasets. For reference, we also include the median errors of our baseline, ACE [2].

TABLE VI: Comparison of relocalization median error (cm / °) on the 7-Scenes and 12-Scenes datasets.

	Sequence	ACE[2]	Ours-Single	Ours-Sequence
7-Scenes	Chess	0.6 / 0.2	**0.5 / 0.1**	**0.5 / 0.1**
	Fire	0.8 / 0.3	0.8 / 0.3	**0.7 / 0.3**
	Heads	0.6 / 0.3	0.4 / 0.2	**0.4 / 0.2**
	Office	1.1 / 0.3	**0.9 / 0.2**	**0.9 / 0.2**
	Pumpkin	1.2 / 0.2	**1.0 / 0.2**	**1.0 / 0.2**
	Kitchen	0.8 / 0.2	0.8 / 0.2	**0.7 / 0.2**
	Stairs	2.8 / 0.8	2.7 / 0.8	**2.4 / 0.7**
12-Scenes	Apt1_kitchen	0.5 / 0.3	**0.4 / 0.2**	**0.4 / 0.2**
	Apt1_living	0.6 / 0.2	**0.5 / 0.2**	**0.5 / 0.2**
	Apt2_bed	0.5 / 0.2	**0.4 / 0.2**	**0.4 / 0.2**
	Apt2_kitchen	0.6 / 0.3	**0.5 / 0.2**	**0.5 / 0.2**
	Apt2_living	0.6 / 0.2	**0.4 / 0.2**	**0.4 / 0.2**
	Apt2_luke	0.7 / 0.3	0.6 / 0.2	**0.5 / 0.2**
	Office1_gates362	0.6 / 0.2	0.6 / 0.2	**0.5 / 0.2**
	Office1_gates381	0.8 / 0.3	0.7 / 0.3	**0.6 / 0.3**
	Office1_lounge	0.8 / 0.2	0.7 / 0.2	**0.6 / 0.2**
	Office1_manolis	0.7 / 0.3	0.7 / 0.3	**0.6 / 0.2**
	Office2_5a	0.8 / 0.3	0.8 / 0.3	**0.7 / 0.3**
	Office2_5b	0.7 / 0.3	0.6 / 0.2	**0.5 / 0.2**
	Average	0.83 / 0.28	0.74 / 0.25	**0.67 / 0.24**
	FPS	51	**88**	60

As shown in Table IV, under a more relaxed threshold, Ours-Sequence still achieves the highest recall rate, outperforming our baseline ACE by 7.2%. Ours-Single not only significantly improving the relocalization speed, but also boosts the recall rate by 4.4%. Table VI further demonstrates that, compared to the baseline, our method substantially reduces the median relocalization errors. These results validate the effectiveness and robustness of our methods.

6. Failure Case Analysis

In this section, we conduct a failure case analysis. Figure 1 presents scenes where our system exhibits relatively large errors. The bottom row shows the scene images, while the top row displays the reprojection error maps generated by

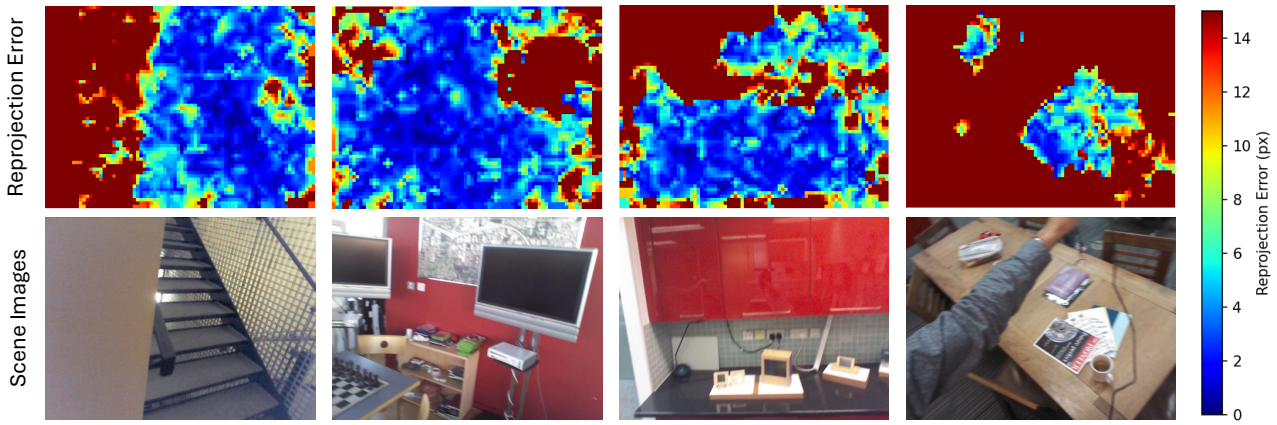

Fig. 1: Failure case analysis. The bottom row shows scene images where our system produces relatively large localization errors. The corresponding top row presents the reprojection error maps, obtained by projecting the 3D points predicted by our SCR network back onto the original images. It can be observed that in low-texture regions or areas containing dynamic objects, the predicted 3D points are unreliable, resulting in significant localization errors.

projecting the 3D points predicted by our SCR network back onto the original images. It can be observed that the reprojection errors are higher in low-texture regions or areas containing dynamic objects. This indicates that the network's implicit triangulation performs poorly in such regions, making the predicted 3D points unreliable. As a result, if the scene contains a large number of low-texture areas, the localization accuracy of our system will be degraded.

References

[1] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[2] E. Brachmann, T. Cavallari, and V. A. Prisacariu, "Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5044–5053.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[4] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[5] R. Murai, E. Dexheimer, and A. J. Davison, "Mast3r-slam: Real-time dense slam with 3d reconstruction priors," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 695–16 705.

[6] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[7] Q. Zhou, S. Agostinho, A. Ošep, and L. Leal-Taixé, "Is geometry enough for matching in visual localization?" in *European Conference on Computer Vision*. Springer, 2022, pp. 407–425.

[8] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3247–3257.

[9] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[10] E. Brachmann and C. Rother, "Visual camera re-localization from rgb and rgb-d images using dsac," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5847–5865, 2021.

[11] S. T. Nguyen, A. Fontan, M. Milford, and T. Fischer, "Focustune: Tuning visual localization through focus-guided sampling," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 3606–3615.

[12] F. Wang, X. Jiang, S. Galliani, C. Vogel, and M. Pollefeys, "Glace: Global local accelerated coordinate encoding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 562–21 571.

[13] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airslam: An efficient and illumination-robust point-line visual slam system," *IEEE Transactions on Robotics*, 2025.