SCE-SLAM: Scale-Consistent Monocular SLAM via Scene Coordinate Embeddings

Supplementary Material

A. Bundle Adjustment Detail

For patch k with pixel coordinates $\mathbf{u}_k = [u_k, v_k]^T$ and depth d_k, we transform from image space to world coordinates. Define the normalized ray direction and coordinate transformations:

$$\mathbf{n}_k = \begin{bmatrix} (u_k - c_x)/f_x \\ (v_k - c_y)/f_y \\ 1 \end{bmatrix} \tag{1}$$

$$\mathbf{X}_k^c = \pi^{-1}(\mathbf{u}_k, d_k) = d_k \mathbf{n}_k \tag{2}$$

$$\mathbf{X}_k^w = \mathbf{R}_{t(k)} \mathbf{X}_k^c + \mathbf{t}_{t(k)} \tag{3}$$

where $\mathbf{X}_k^c$ and $\mathbf{X}_k^w$ are the 3D point in camera and world frames respectively, $t(k)$ denotes the frame index of patch k, and $\mathbf{T}_{t(k)} = [\mathbf{R}_{t(k)} \mid \mathbf{t}_{t(k)}]$ is the camera-to-world transformation. The scene coordinate residual is:

$$\mathbf{r}_k^{\text{xyz}} = w_k^{\text{xyz}} (\mathbf{X}_k^{\text{prior}} - \mathbf{X}_k^w) \in \mathbb{R}^3 \tag{4}$$

where $\mathbf{X}_k^{\text{prior}}$ is the scale-anchored coordinate prediction from the embedding $\mathbf{h}_k^{\text{xyz}}$, and w_k^{xyz} is the confidence weight.

A.1. Jacobian with Respect to Camera Pose

For a Lie algebra perturbation $\boldsymbol{\tau} \in \mathfrak{se}(3)$ with translation $\boldsymbol{\rho}$ and rotation $\boldsymbol{\phi}$, the left Jacobian of the pose action is:

$$\mathbf{J}_{t(k)}^{\text{xyz}} = \frac{\partial \mathbf{r}_k^{\text{xyz}}}{\partial \boldsymbol{\tau}_{t(k)}} = -w_k^{\text{xyz}} \begin{bmatrix} \mathbf{I}_{3\times 3} & -[\mathbf{X}_k^w]_\times \end{bmatrix} \in \mathbb{R}^{3\times 6} \tag{5}$$

where $[\mathbf{X}_k^w]_\times$ is the skew-symmetric matrix:

$$[\mathbf{X}_k^w]_\times = \begin{bmatrix} 0 & -Z & Y \\ Z & 0 & -X \\ -Y & X & 0 \end{bmatrix} \tag{6}$$

A.2. Jacobian with Respect to Depth

From the backprojection definition $\mathbf{X}_k^c = d_k \mathbf{n}_k$, we have:

$$\frac{\partial \mathbf{X}_k^c}{\partial d_k} = \mathbf{n}_k \tag{7}$$

Since $\mathbf{X}_k^w = \mathbf{R}_{t(k)} \mathbf{X}_k^c + \mathbf{t}_{t(k)}$ and the translation does not depend on depth:

$$\frac{\partial \mathbf{X}_k^w}{\partial d_k} = \mathbf{R}_{t(k)} \frac{\partial \mathbf{X}_k^c}{\partial d_k} = \mathbf{R}_{t(k)} \mathbf{n}_k \tag{8}$$

Therefore:

$$\mathbf{J}_{d_k}^{\text{xyz}} = \frac{\partial \mathbf{r}_k^{\text{xyz}}}{\partial d_k} = -w_k^{\text{xyz}} \mathbf{R}_{t(k)} \mathbf{n}_k \in \mathbb{R}^{3\times 1} \tag{9}$$

A.3. Hessian Matrix Construction

In the Gauss-Newton framework, the Hessian matrix is approximated by the first-order term:

$$\mathbf{H} = \mathbf{J}^T \mathbf{W} \mathbf{J} \tag{10}$$

where $\mathbf{W}$ is a diagonal weight matrix. We construct the weighted Jacobian matrices and residuals, and assemble them into a sparse block-structured linear system. Following the common practice in modern visual SLAM systems [7, 8], we solve this system using the Schur complement to efficiently marginalize the depth variables, yielding updates for the camera poses $\Delta\boldsymbol{\tau}$ and depth updates $\Delta\mathbf{d}$.

B. Runtime Analysis

We analyze the computational efficiency of our method by profiling the runtime of each component. As shown in Figure 7, the processing time is 213 seconds for the KITTI00 [4] sequence. The **backbone network** consumes the majority of the computation time (49.9%), which is expected as it processes high-resolution images for feature extraction. The **flow branch** and **scene coordinate branch** account for 25.3% and 11.8% respectively.

Notably, our bundle adjustment with two complementary optimization objectives consumes only 2.7% of the total time, demonstrating computational efficiency. Our SCE achieves superior accuracy with standard settings compared to heavier alternatives that show diminishing returns (Table 7). Following DPVO [8], we benchmark on EuRoC [1] with 4090 GPU, achieving 36 FPS. The main bottleneck lies in the deep learning components, while our optimization adds minimal overhead.

C. Additional Results

Virtual KITTI. We evaluate robustness under diverse environmental conditions on the Virtual KITTI dataset [3], which includes various weather and lighting conditions such as clone, fog, morning, overcast, rain, and sunset (Table 8). Our method achieves the best overall performance across all scenes and conditions, significantly outperforming existing methods including DPVSLAM++ [5].

4 Seasons. On the Parking Garage sequence [10] featuring low-light and repetitive structures (Figure 8), DPVSLAM++ produces fragmented, layered point clouds due to pose drift, while our method generates coherent reconstruction closely matching the ground truth, demonstrating superior accuracy in challenging indoor scenarios.

Method	w/o LC ATE(m)		w/ LC ATE(m)	
DPVO (baseline)	53.61 ($P=96, W=10$)		25.76 ($P=96, W=10$)	
DPVO (larger windows)	48.59 ($P=96, W=18$)	47.88 ($P=96, W=26$)	21.80 ($P=96, W=18$)	24.76 ($P=96, W=26$)
DPVO (more patches)	52.83 ($P=128, W=10$)	52.95 ($P=160, W=10$)	25.42 ($P=128, W=10$)	25.69 ($P=160, W=10$)
DPVO + Marginalization	46.62 ($P=96, W=10$)	45.62 ($P=128, W=18$)	22.85 ($P=96, W=10$)	24.17 ($P=128, W=18$)
Ours	**25.79** ($P=96, W=10$)		**14.07** ($P=96, W=10$)	

Table 7. **SCE effectiveness on KITTI-11 (ATE RMSE ↓).** Our SCE ($P=96, W=10$) outperforms heavier alternatives (larger windows, more patches, marginalization) that show diminishing returns, validating its scale-aware design.

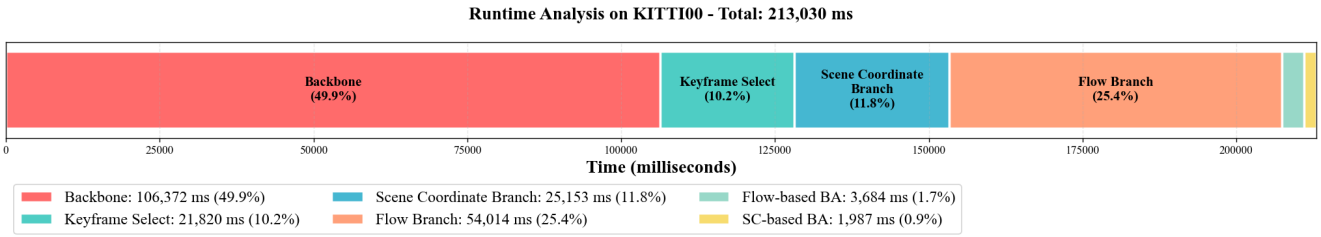

Figure 7. **Component-wise Runtime analysis of the proposed method.**

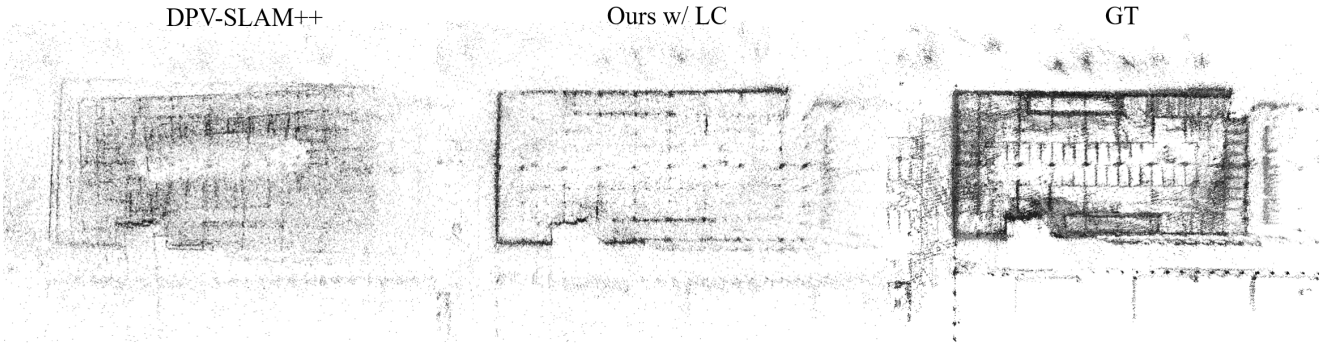

Figure 8. **Comparison of reconstruction on the Parking Garage sequence from the 4 Seasons dataset.**

	Scene 01						Scene 02					
Condition	*Clone*	*Fog*	*Morning*	*Overcast*	*Rain*	*Sunset*	*Clone*	*Fog*	*Morning*	*Overcast*	*Rain*	*Sunset*
DROID-SLAM [7]	1.027	1.868	0.989	1.015	0.776	1.145	0.098	0.040	0.049	0.047	0.036	0.112
MASt3R-SLAM [6]	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL
CUT3R [9]	43.304	62.191	50.608	38.735	51.548	43.785	23.771	9.948	28.415	24.644	7.963	25.973
VGGT-Long [2]	0.763	0.874	0.928	0.670	1.799	1.259	0.723	0.709	0.721	0.681	0.693	0.689
DPVSLAM++ [5]	0.392	**0.135**	0.492	**0.254**	0.446	**0.276**	**0.002**	0.028	**0.015**	0.014	0.021	0.017
Ours	**0.269**	0.137	**0.487**	0.265	**0.346**	0.285	0.014	**0.025**	0.022	**0.012**	**0.017**	**0.016**
	Scene 06						**Scene 18**					
Condition	*Clone*	*Fog*	*Morning*	*Overcast*	*Rain*	*Sunset*	*Clone*	*Fog*	*Morning*	*Overcast*	*Rain*	*Sunset*
DROID-SLAM [7]	0.063	0.024	0.030	0.051	TL	0.020	2.478	2.032	1.893	2.332	2.550	1.943
MASt3R-SLAM [6]	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL
CUT3R [9]	0.836	0.408	0.599	0.720	1.059	1.013	19.440	8.628	6.720	20.212	16.777	31.119
VGGT-Long [2]	0.365	0.543	0.376	0.402	0.559	0.382	1.651	0.797	1.288	1.256	1.648	1.740
DPVSLAM++ [5]	0.055	0.054	0.050	**0.069**	**0.055**	0.077	0.449	**0.016**	0.179	0.217	**0.151**	0.207
Ours	**0.048**	**0.050**	**0.047**	0.079	0.063	**0.075**	**0.382**	0.024	**0.166**	**0.185**	0.177	**0.184**
	Scene 20						**AVG**					
Condition	*Clone*	*Fog*	*Morning*	*Overcast*	*Rain*	*Sunset*	*01Avg.*	*02Avg.*	*06Avg.*	*18Avg.*	*20Avg.*	*All Avg.*
DROID-SLAM [7]	3.592	5.079	3.733	3.852	3.780	4.907	1.137	0.064	0.038	2.205	4.157	1.520
MASt3R-SLAM [6]	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL	TL
CUT3R [9]	129.498	76.962	117.948	114.512	66.700	116.529	48.362	20.119	0.772	17.149	103.692	38.019
VGGT-Long [2]	9.655	8.185	6.345	4.564	6.499	4.85	1.049	0.703	0.438	1.397	6.683	2.054
DPVSLAM++ [5]	0.924	2.257	0.648	0.648	1.242	0.9176	0.332	**0.016**	**0.060**	0.203	1.106	0.344
Ours	**0.743**	**1.544**	**0.479**	**0.593**	**0.876**	**0.794**	**0.298**	0.018	**0.060**	**0.186**	**0.838**	**0.280**

Table 8. **Comparison on the Virtual KITTI Dataset of ATE RMSE ↓ (m).** The **best** and second-best results are marked.

References

[1] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35 (10):1157–1163, 2016. 1

[2] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443*, 2025. 2

[3] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 1

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[5] Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *European Conference on Computer Vision*, pages 424–440. Springer, 2024. 1, 2

[6] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 2

[7] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 1, 2

[8] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36:39033–39051, 2023. 1

[9] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 2

[10] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2020. 1