Supplementary Materials for "ParticleSfM: Exploiting Dense Point Trajectories for Localizing Moving Cameras in the Wild"

Wang Zhao[1,3], Shaohui Liu[2,3], Hengkai Guo[3], Wenping Wang[4], and Yong-Jin Liu[1]

[1]Tsinghua University [2]ETH Zurich [3]ByteDance Inc. [4]Texas A&M University

A More Implementation Details

Point Trajectory. We use the pretrained *raft-things* model for RAFT [16] in all our experiments. It is trained on FlyingChairs and FlyingThings3D [10]. For optical flow forward-backward consistency check, we use the threshold of 1px for MPI Sintel dataset [2], and 3px for ScanNet dataset [4].

Motion Segmentation. For training trajectory motion segmentation network, we first prepare the trajectory data of FlyingThings3D dataset [10]. The dataset consists of over 2000 training scenes, and each scene contains 10 video frames, together with groundtruth optical flow, camera parameters and depth maps. We run our dense point trajectory generation algorithm to track trajectories from groundtruth optical flows, and then calculate trajectory groundtruth motion labels by comparing optical flow with rigid flow from depths and camera poses. We infer the relative depth information by pretrained MiDaS [13] model. For all experiments, we use the pretrained *midas-v21* model. During training, we directly take all the point trajectories from 10 video frames and output the per-trajectory motion label. Weighted binary cross-entropy loss is then applied.

Global Bundle Adjustment (BA). The implementation of our pipeline is mainly based on the Theia SfM system [15]. With the dense correspondences sampled from the point trajectories, we first compute two-view geometry [5] between valid image pairs and decompose the relative poses. In particular, the view pairs with very few or extremely noisy correspondences are detected with geometric verification [14] and ignored in the subsequent stages. Then, L1-IRLS rotation averaging [3] is applied to estimate the global orientations from the relative rotations among those valid pairs. After filtering outlier pairs with large errors on the relative rotations, we solve for the relative translations with the global rotations and apply LUD translation averaging [12] to get global translations. With these initial global poses, we incrementally triangulate 2D observations as in [14] and perform bundle adjustment on all the poses and 3D points to get the final output camera poses. Note that since we triangulate over the correspondences directly sampled from the dense point trajectories, each constraint in the bundle adjustment exactly corresponds to a part of the original point trajectory with geometric filtering, enabling effective global bundle adjustment over the input trajectory observations.

B Comparisons with SLAM methods

We provide the quantitative comparison results on Sintel dataset with representative monocular SLAM methods ORB-SLAM [11] and DynaSLAM [1]. ORB-SLAM [11] is the most popular feature-based monocular SLAM method which utilizes robust front-end tracking and local bundle-adjustment to achieve accurate camera localization. Built on top of ORB-SLAM, DynaSLAM [1] further introduces semantics to remove potentially dynamic objects to improve the robustness. Since ORB-SLAM and DynaSLAM only provide key-frame poses, we compared the localization accuracy solely on their key-frames. For Sintel dataset, both ORB-SLAM and DynaSLAM **consistently fail in 8 of 14 sequences**, and we summarize the results from other 6 successful sequences in Table 1. Our method surpasses both of them by a large margin even on their successful subset. Furthermore, ORB-SLAM and DynaSLAM **fail on all 17 ScanNet sequences**, probably due to large motion blur and poorly textured regions, while our method consistently provides reasonable camera poses.

Table 1. Evaluation on the successful subset (6 out of 14 sequences) of ORB-SLAM / DynaSLAM on MPI Sintel dataset.

Methods	ATE (m)	RPE trans (m)	RPE rot (deg)
ORB-SLAM	0.042	0.022	0.402
Ours	**0.009**	**0.006**	**0.101**
DynaSLAM	0.020	0.019	0.359
Ours	**0.007**	**0.005**	**0.090**

C Per-scene Results on MPI Sintel and ScanNet

We show the per-scene comparison results of MPI Sintel [2] and ScanNet [4] dataset in Table 2 and Table 3. For MPI Sintel, our method achieves the best performance on most sequences, demonstrating the advantage of the proposed system in dynamic scenarios. For fully static indoor dataset ScanNet, our method retain comparable performance with COLMAP [14], slightly behind on ATE and better on RPEs. COLMAP globally matches the feature points between image pairs, thus naturally has the ability of loop closure, while our method lacks as point trajectories are accumulated sequentially. Although our method could achieve good relative pose estimations, the trajectory error will be accumulated without loop closure. This is possibly the main reason why our method is worse than COLMAP in ATE but better in RPEs. In the future, we aim to implement the loop closure inside our system by matching trajectories across frames.

Table 2. Per-scene results on MPI Sintel dataset [2].

	Metrics	COLMAP [14]	MAT [21] + [14]	Mask-RCNN [6] + [14]	R-CVD [8]	Tartan-VO [18]	DROID-SLAM [17]	Ours
alley_2	ATE (m)	0.072	**0.002**	0.072	0.026	0.062	0.057	**0.002**
	RPE trans (m)	0.039	0.0009	0.038	0.056	0.049	0.035	**0.0007**
	RPE rot (deg)	0.678	0.014	0.679	0.821	0.856	1.047	**0.009**
ambush_4	ATE (m)	0.030	0.174	**0.029**	0.171	0.100	0.104	0.068
	RPE trans (m)	0.032	0.046	0.045	0.048	0.038	0.035	**0.027**
	RPE rot (deg)	0.377	3.425	0.541	3.025	1.320	1.385	**0.473**
ambush_5	ATE (m)	0.028	0.090	0.004	0.230	0.098	0.112	**0.002**
	RPE trans (m)	0.015	0.036	0.005	0.046	0.037	0.029	**0.001**
	RPE rot (deg)	0.607	0.817	0.204	4.105	1.107	1.580	**0.055**
ambush_6	ATE (m)	X	X	X	**0.199**	0.205	0.289	0.269
	RPE trans (m)	X	X	X	0.112	0.107	**0.078**	0.081
	RPE rot (deg)	X	X	X	4.147	4.293	4.596	**0.951**
cave_2	ATE (m)	X	X	X	**0.596**	1.167	0.351	0.961
	RPE trans (m)	X	X	X	0.171	**0.131**	0.172	0.142
	RPE rot (deg)	X	X	X	7.508	4.112	5.489	**3.678**
cave_4	ATE (m)	0.051	0.049	**0.044**	0.179	0.120	0.155	0.068
	RPE trans (m)	0.013	0.028	0.040	0.087	0.039	0.035	**0.012**
	RPE rot (deg)	0.451	0.700	0.600	2.040	1.327	2.710	**0.409**
market_2	ATE (m)	X	X	X	0.032	0.068	0.011	**0.003**
	RPE trans (m)	X	X	X	0.018	0.007	**0.006**	0.010
	RPE rot (deg)	X	X	X	0.141	0.090	**0.036**	0.041
market_5	ATE (m)	1.105	0.284	0.816	1.213	1.158	0.912	**0.012**
	RPE trans (m)	0.210	0.095	0.212	0.762	0.294	0.293	**0.006**
	RPE rot (deg)	2.232	0.055	2.380	1.863	1.100	3.334	**0.034**
market_6	ATE (m)	X	X	X	0.248	0.260	0.057	**0.018**
	RPE trans (m)	X	X	X	0.214	0.110	0.037	**0.007**
	RPE rot (deg)	X	X	X	0.817	1.287	1.296	**0.120**
shaman_3	ATE (m)	0.012	0.006	0.009	0.054	0.008	0.001	**0.0005**
	RPE trans (m)	0.003	0.005	0.007	0.023	0.006	0.002	**0.0004**
	RPE rot (deg)	0.537	0.978	0.977	0.718	0.185	0.199	**0.072**
sleeping_1	ATE (m)	**0.008**	0.013	**0.008**	0.029	0.017	0.011	**0.008**
	RPE trans (m)	**0.001**	0.006	**0.001**	0.019	0.011	0.006	**0.001**
	RPE rot (deg)	0.053	0.530	0.046	0.668	0.344	0.479	**0.042**
sleeping_2	ATE (m)	**0.0002**	**0.0002**	**0.0002**	0.043	0.013	0.005	0.0007
	RPE trans (m)	**0.0002**	**0.0002**	**0.0002**	0.049	0.022	0.0177	**0.0002**
	RPE rot (deg)	0.008	**0.006**	0.007	0.446	0.267	0.139	**0.006**
temple_2	ATE (m)	**0.004**	0.006	0.006	1.245	0.447	0.073	0.011
	RPE trans (m)	0.003	**0.002**	**0.002**	0.394	0.324	0.348	**0.002**
	RPE rot (deg)	0.012	**0.007**	0.013	1.318	0.789	1.298	0.019
temple_3	ATE (m)	X	X	X	0.769	0.331	**0.310**	0.381
	RPE trans (m)	X	X	X	0.161	0.105	**0.093**	0.149
	RPE rot (deg)	X	X	X	20.592	**1.166**	3.230	1.583

D Additional Visualization

We show more visualizations about trajectory motion segmentation and camera localization in Figure 1. Sequences are from 3DPW [9], Youtube-VOS [19], GOT-10K [7] and BANMO [20] dataset. See the attached video for better experience.

Table 3. Per-scene results on ScanNet dataset [4].

	Metrics	COLMAP [14]	R-CVD [8]	Tartan-VO [18]	DROID-SLAM [17]	Ours
	ATE (m)	**0.147**	0.442	0.418	0.978	0.199
scene0707_00	RPE trans (m)	0.059	0.092	0.065	0.043	**0.020**
	RPE rot (deg)	0.803	7.301	2.914	3.530	**0.574**
	ATE (m)	**0.143**	0.437	0.202	0.872	0.220
scene0709_00	RPE trans (m)	0.067	0.088	0.063	0.052	**0.015**
	RPE rot (deg)	0.780	6.852	2.698	3.187	**0.523**
	ATE (m)	**0.073**	0.429	0.306	0.631	0.247
scene0710_00	RPE trans (m)	0.016	0.039	0.027	0.030	**0.012**
	RPE rot (deg)	**0.371**	4.412	1.961	2.153	0.424
	ATE (m)	**0.051**	0.183	0.514	0.639	0.232
scene0712_00	RPE trans (m)	**0.016**	0.021	0.025	0.017	**0.016**
	RPE rot (deg)	**0.383**	3.807	1.943	2.221	0.619
	ATE (m)	**0.204**	0.472	0.515	0.616	0.309
scene0713_00	RPE trans (m)	0.124	0.090	0.047	0.047	**0.024**
	RPE rot (deg)	9.536	18.216	3.437	4.127	**1.287**
	ATE (m)	0.891	0.644	0.389	0.916	**0.372**
scene0714_00	RPE trans (m)	0.215	0.075	0.064	0.045	**0.019**
	RPE rot (deg)	9.190	7.498	2.630	3.485	**0.418**
	ATE (m)	0.267	**0.230**	0.239	0.511	0.341
scene0715_00	RPE trans (m)	0.156	0.039	0.049	0.039	**0.026**
	RPE rot (deg)	15.059	8.835	2.930	3.524	**0.611**
	ATE (m)	**0.091**	0.324	0.508	0.782	0.252
scene0717_00	RPE trans (m)	0.040	0.050	0.058	0.040	**0.022**
	RPE rot (deg)	0.586	7.267	3.006	3.453	**0.555**
	ATE (m)	X	0.350	**0.111**	0.385	0.295
scene0718_00	RPE trans (m)	X	0.080	0.065	0.066	**0.039**
	RPE rot (deg)	X	12.460	3.837	5.189	**0.844**
	ATE (m)	**0.051**	0.373	0.171	0.657	0.268
scene0719_00	RPE trans (m)	0.019	0.041	0.044	0.031	**0.012**
	RPE rot (deg)	**0.330**	6.919	2.423	3.380	0.401
	ATE (m)	**0.133**	0.390	0.331	0.389	0.815
scene0720_00	RPE trans (m)	**0.040**	0.034	0.030	0.027	0.021
	RPE rot (deg)	0.900	**0.668**	2.002	1.915	0.875
	ATE (m)	X	0.521	**0.259**	1.345	0.625
scene0721_00	RPE trans (m)	X	0.054	**0.042**	0.070	0.110
	RPE rot (deg)	X	6.439	**2.022**	2.260	5.304
	ATE (m)	**0.050**	0.427	0.319	0.486	0.467
scene0722_00	RPE trans (m)	0.027	0.041	0.042	0.031	**0.019**
	RPE rot (deg)	**0.444**	8.193	2.943	3.523	0.489
	ATE (m)	**0.139**	0.766	0.483	0.521	0.220
scene0723_00	RPE trans (m)	0.031	0.079	0.036	0.028	**0.015**
	RPE rot (deg)	0.796	5.675	2.201	2.304	**0.603**
	ATE (m)	**0.062**	0.647	0.429	0.702	0.332
scene0724_00	RPE trans (m)	0.040	0.090	0.036	0.027	**0.013**
	RPE rot (deg)	**0.854**	5.857	2.796	3.218	1.022
	ATE (m)	X	0.714	0.550	0.882	**0.548**
scene0725_00	RPE trans (m)	X	0.075	0.036	0.027	**0.013**
	RPE rot (deg)	X	9.665	2.272	2.612	**0.708**
	ATE (m)	**0.100**	0.474	0.258	0.380	0.193
scene0726_00	RPE trans (m)	0.051	0.055	0.043	0.035	**0.011**
	RPE rot (deg)	0.563	6.567	2.524	2.904	**0.458**

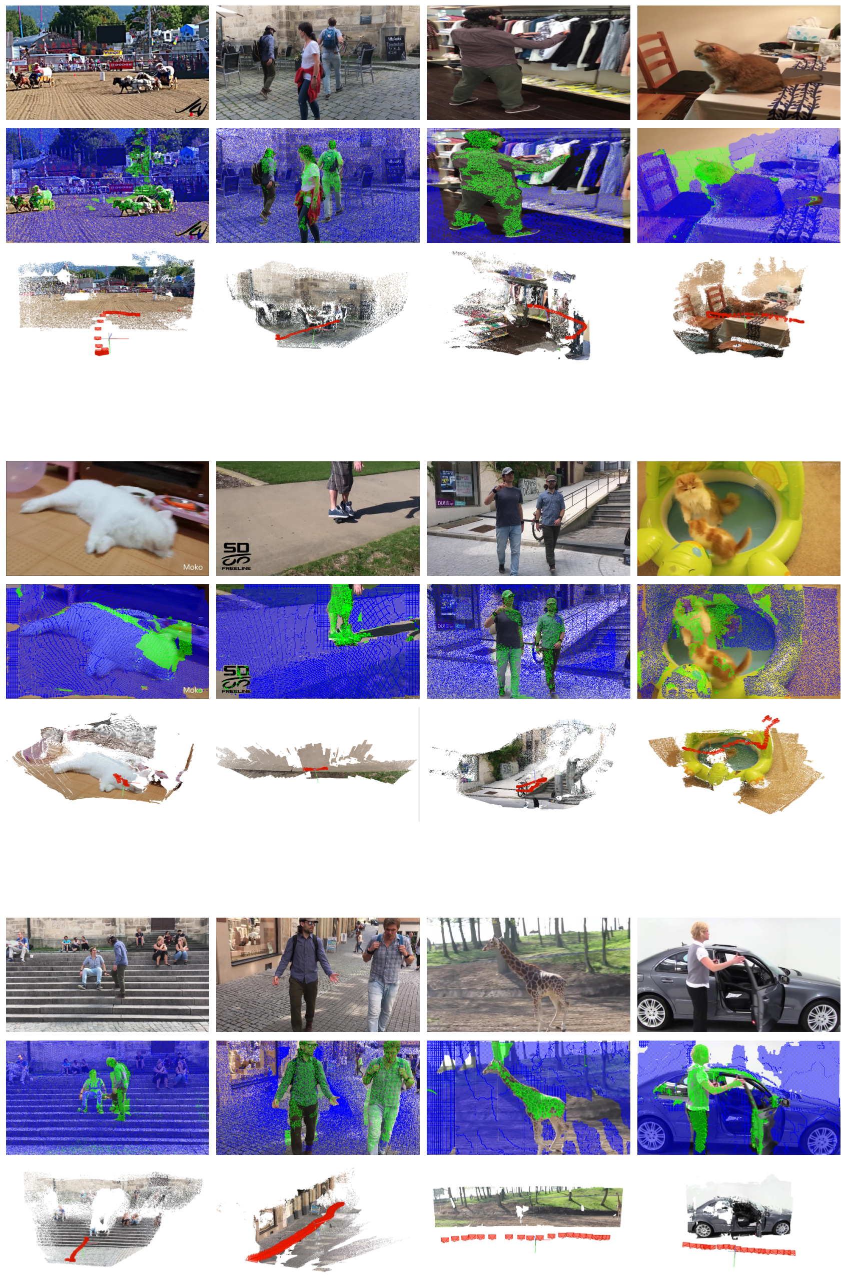

Fig. 1. Visualization of trajectory motion segmentation and camera localization of in-the-wild videos. **Moving pixels from point trajectories are colored in green and static background pixels are in blue. Free space with no colored pixels indicates that there are no trajectory points due to occlusion or large flow forward-backward consistency error.**

References

1. Bescos, B., Fácil, J.M., Civera, J., Neira, J.: Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. IEEE Robotics and Automation Letters **3**(4), 4076–4083 (2018)
2. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European conference on computer vision. pp. 611–625. Springer (2012)
3. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 521–528 (2013)
4. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
5. Hartley, R.I.: In defense of the eight-point algorithm. IEEE Transactions on pattern analysis and machine intelligence **19**(6), 580–593 (1997)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
7. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(5), 1562–1577 (2019)
8. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1611–1621 (2021)
9. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 601–617 (2018)
10. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
11. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
12. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2674–2683 (2015)
13. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv preprint arXiv:1907.01341 (2019)
14. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
15. Sweeney, C.: Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org
16. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
17. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in Neural Information Processing Systems **34** (2021)

18. Wang, W., Hu, Y., Scherer, S.: Tartanvo: A generalizable learning-based vo. arXiv preprint arXiv:2011.00359 (2020)
19. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 585–601 (2018)
20. Yang, G., Vo, M., Natalia, N., Ramanan, D., Andrea, V., Hanbyul, J.: Banmo: Building animatable 3d neural models from many casual videos. arXiv preprint arXiv:2112.12761 (2021)
21. Zhou, T., Wang, S., Zhou, Y., Yao, Y., Li, J., Shao, L.: Motion-attentive transition for zero-shot video object segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13066–13073 (2020)