WildPose: A Unified Framework for Robust Pose Estimation in the Wild

Supplementary Material

Abstract

In the supplementary material, we provide additional details about the following:

1. *More information about the training dataset (Sec. 6).*
2. *Implementation details of WildPose, including more training details and model architecture (Sec. 7).*
3. *Additional results and discussion (Sec. 8).*
4. *Limitations and future work (Sec. 9).*

6. Training Dataset

We trained our model on four publicly available datasets supplemented with data that we generated using the Kubric simulator [15]. The training datasets encompass both static and dynamic environments. A comprehensive list of the datasets we used is provided in Table 7.

While the TartanAir V2 [44] and TartanGround [35] datasets primarily feature static scenes, we identify dynamic content in 8 of their 65 environments. Among these, ground truth motion masks can be reliably extracted via semantic segmentation for only four specific environments: `AmusementPark`, `SeasonalForestAutumn`, `WaterMillDay`, and `WaterMillNight`. The dynamic content in the remaining four scenes (`CarWelding`, `CyberPunk Downtown`, `Ocean`, and `SoulCity`) is complex and difficult to segment accurately. Hence, these four scenes are discarded from the training set. Furthermore, sequences containing significant photographic or geometric artifacts (e.g., camera trajectories passing directly through walls in sequence P004 of `OldBrickHouseDay`) are also discarded from the training set to maintain data quality.

The OmniWorld-Game dataset [61] provides foreground masks generated using Grounding DINO [28] to identify bounding boxes, which are then fed as prompts to SAM [23] to extract fine masks. However, these generated masks are often noisy, exhibiting incomplete segmentation of instances or entirely missing dynamic objects. Therefore, we do not use their annotated masks as ground truth for calculating the BCE loss in Eq. (4) when training the motion mask detector, but we incorporate them in the second stage of update operator finetuning to encourage robustness against noisy motion maps.

We utilized the Kubric simulator [15] to generate diverse motion patterns that are under-represented in the public datasets. We considered three specific camera trajectory types: (1) Linear Translation, defined by a linear movement between two randomly selected waypoints; (2) Pure Rotation, involving zero translation, with trajectories encompassing rotation along single or multiple axes; and (3) Target-Locked Motion, where the camera's translation varies but its rotation is constrained to maintain focus on a fixed point. The third type included four distinct translation patterns: lateral, vertical, orbital, and spiral movement. For both static and dynamic scene configurations, we generated a total of 5,500 sequences, each comprising 24 or 36 frames. This generated dataset will be publicly released.

Dataset Name	Dynamic?	Dynamic Mask?	Camera Motion Trajectory
TartanAir V2 [44]	No*	-	Drone-style
TartanGround [35]	No*	-	Grounded Robot
Dynamic Replica [20]	Yes	Yes	Handheld 6-DOF
OmniWorld-Game [61]	Yes	Noisy	Player-controlled
Kubric (Generated) [15]	Mixed	Yes	Specified motion patterns

Table 7. **Training Datasets.** We utilized four public data sources and generated data from Kubric [15] for training WildPose. The table details the scene type, the availability and quality of ground truth dynamic masks, and the type of camera motions. OmniWorld-Game [61] uses open-source segmentation models [23, 28] to generate dynamic mask, hence noisy. While most environments in TartanAir V2 [44] and TartanGround [35] are static, there are eight environments containing dynamic objects.

7. Implementation Details

7.1. Additional Training Details

Following [43], we sample 7 frames per batch from the training sequence. We constrain the average optical flow magnitude between neighboring pairs to fall within the range of 8 to 96 pixels. For all frames, we apply standard data augmentation, comprising photometric transformations (color jitter, grayscale, and Gaussian blurring) and spatial randomization via center resizing. The images are then cropped to the fixed input resolution of 384×512. Subsequently, a frame graph is constructed over the 7 frames with edges determined by flow distance. We execute 15 iterations of the update operator and differentiable BA.

The total training objective for the update operator combines three weighted loss components. The Pose Loss $\mathcal{L}_{cam}$ is the geodesic distance computed by mapping the relative pose error $\mathbf{G}_{\text{rel}}\mathbf{P}_{\text{rel}}^{-1}$ to the Lie Algebra $\mathfrak{se}(3)$ using the log map. Here, $\mathbf{G}_{\text{rel}}$ and $\mathbf{P}_{\text{rel}}$ denote the ground truth and predicted relative poses, respectively. $\mathcal{L}_{\text{cam}}$ is defined as the average of the L_2 norms of the translational τ and rotational ϕ components. We further utilize a Flow Loss $\mathcal{L}_{\text{flow}}$ defined by the L_2 norm of the difference between the ground truth optical flow and the flow induced by the estimated pose and disparity $\tilde{\mathbf{f}}$. Lastly, we use the Residual Loss $\mathcal{L}_{\text{res}}$, which is

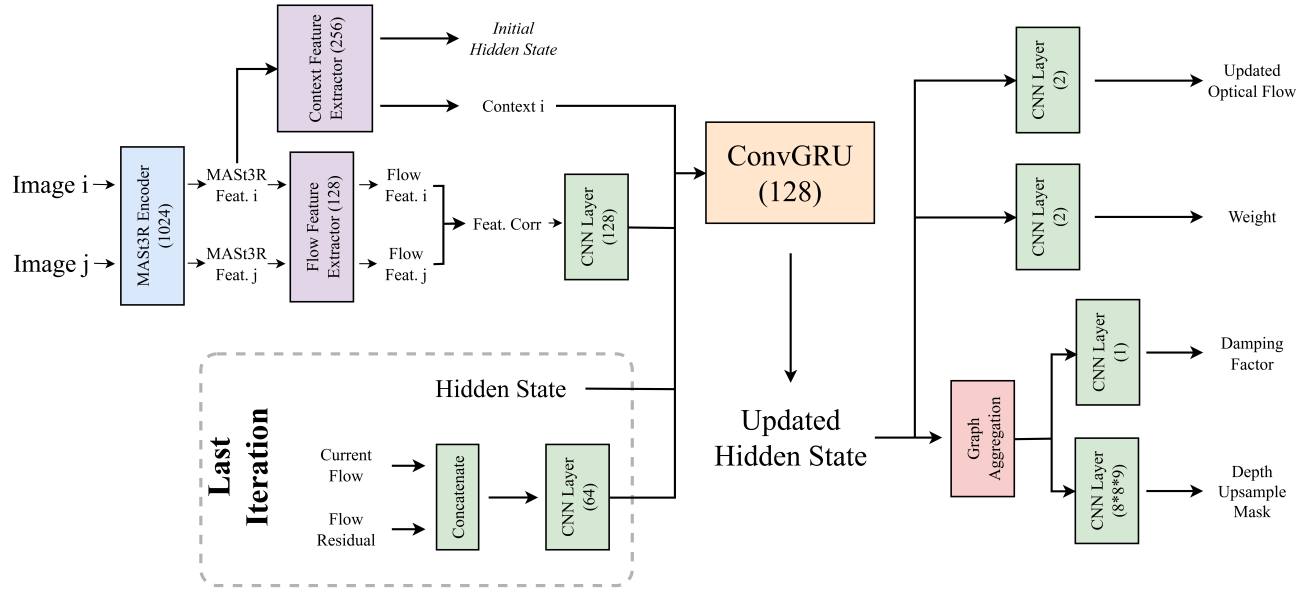

Figure 5. **Architecture of Update Operator.** The ConvGRU iteratively updates the hiddens state from the image feature correlation, context features, and the current optical flow. The updated hidden state is further decoded to variables that will be used to guide pose and disparity estimation in the differentiable BA process.

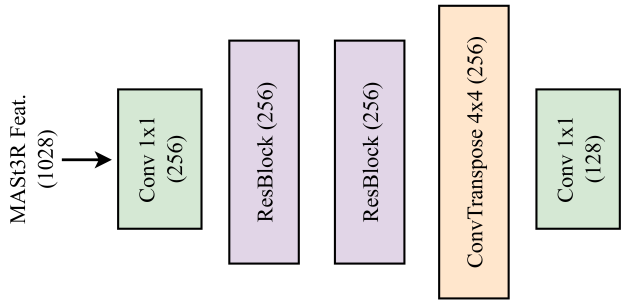

Figure 6. **Architecture of the flow feature and context encoders.** Both encoders take the MASt3R features as input and output features at 1/8 of the image resolution. For the context encoder, the dimension of the last convolution layer is 256.

calculated by the L_1 distance between $\tilde{\mathbf{f}}$ and the flow predicted by the update operator $\hat{\mathbf{f}}$. All three losses are applied across the 15 BA steps, utilizing an increasing temporal weighting scheme $w_k = \gamma^{(15-k)}$, where $\gamma = 0.9$ and k is the iteration step.

When training the motion mask detector, the parameters of the update operator are frozen. As mentioned in the main paper, we integrate the predicted motion mask as an additional weight during the 15 BA steps. We retain the Pose Loss $\mathcal{L}_{cam}$, exclude $\mathcal{L}_{\text{flow}}$ and $\mathcal{L}_{\text{res}}$, and add the BCE loss $\mathcal{L}_{\text{BCE}}$, which is only applied when reliable annotated ground truth motion masks are available.

For the initial static pretraining stage of the update operator, we utilize only the static portions of the TartanAir V2, TartanGround, and Kubric-Generated datasets. All remaining datasets, including dynamic sequences, are then incorporated for the second stage of update operator finetuning and subsequent motion mask detector training.

7.2. Model Architecture

Our update operator is extended from DROID-SLAM [43]. The full architecture is illustrated in Fig. 5. Given a pair of input images $(\mathbf{I}_i, \mathbf{I}_j)$, we first extract 3D-aware features from the pre-trained MASt3R Encoder. These high-dimensional features (1024 channels) are produced at $\frac{1}{16}$ resolution relative to the original image. We then utilize two separate encoders, the Context Feature Extractor and the Flow Feature Extractor, to process the MASt3R output. The architecture shared by these encoders is shown in Fig. 6. This stage lifts the feature resolution to the required $\frac{1}{8}$ scale. The Flow Feature Extractor output is subsequently used to compute the 4D correlation volume between the image features. The key component of the update operator is the ConvGRU layer. It takes the context feature, flow feature correlation, current hidden state, and the optical flow from the last iteration as input, and outputs an updated hidden state, which is further used to extract intermediate variables for differentiable BA, such as the updated optical flow and the confidence weight.

8. Additional Results

Full Tracking Results on the static Dataset. In the main paper, we summarize the average ATE for the TUM RGB-D (static) [42] and 7-Scenes [40] datasets. Here, we present

Method	360	desk	desk2	floor	plant	room	rpy	teddy	xyz
Keyframe Poses									
MASt3R-SLAM [33]	**0.049**	**0.016**	**0.024**	0.025	0.020	0.061	0.027	0.041	**0.009**
VGGT [46]	**0.049**	0.023	0.029	0.084	0.024	0.095	0.025	0.033	0.012
π^3 [50]	0.065	0.018	0.028	0.085	0.026	0.061	0.024	**0.028**	0.011
Full Trajectory									
DROID-SLAM [43]	0.111	0.018	0.042	0.021	**0.016**	0.049	0.026	0.048	0.012
WildGS-SLAM [60]	0.069	0.018	0.025	0.023	0.044	0.061	0.024	0.058	**0.009**
MegaSaM [27]	0.055	0.019	0.040	0.278	0.038	0.080	0.024	0.043	0.013
ViPE [17]	0.067	0.024	0.035	0.320	0.025	0.048	0.020	0.038	0.011
WildPose (Ours)	0.057	**0.016**	**0.024**	**0.019**	0.018	0.039	**0.018**	0.035	**0.009**

Table 8. **Tracking Performance on TUM RGB-D (static) Dataset [42]** (ATE RMSE ↓ [m]).

Method	chess	fire	heads	office	pumpkin	kitchen	stairs
Keyframe Poses							
MASt3R-SLAM [33]	0.053	0.025	**0.015**	0.097	**0.088**	0.041	**0.011**
VGGT [46]	0.039	0.027	0.017	0.098	0.128	0.059	0.036
π^3 [50]	**0.034**	0.034	0.017	0.080	0.118	**0.037**	0.047
Full Trajectory							
DROID-SLAM [43]	0.036	0.027	0.025	0.066	0.127	0.040	0.026
WildGS-SLAM [60]	0.037	0.032	0.027	0.054	0.154	**0.037**	0.019
MegaSaM [27]	0.040	0.040	0.024	0.069	0.147	0.045	0.025
ViPE [17]	0.035	**0.027**	0.023	0.070	0.131	0.039	0.024
WildPose (Ours)	0.038	0.029	0.024	**0.053**	0.142	**0.037**	0.023

Table 9. **Tracking Performance on 7-Scenes Dataset [40]** (ATE RMSE ↓ [m]).

Method	*0000*	*0059*	*0106*	*0169*	*0181*	*0207*	Avg.
DROID-SLAM [43]	**5.48**	9.00	6.76	**7.89**	7.41	7.97	7.42
WildGS-SLAM [60]	5.74	8.15	9.15	8.31	7.17	7.31	7.64
MegaSaM [27]	*OOM*	8.56	37.35	10.16	8.35	8.54	-
WildPose (Ours)	6.11	**8.07**	**6.75**	8.01	**6.56**	**7.36**	**7.14**

Table 10. **Tracking Performance on ScanNet Dataset [8]** (ATE RMSE ↓ [m]).

Method	ATE [cm] ↓	Avg. fps ↑	Peak GPU Use [GiB] ↓
DROID-SLAM [43]	16.17	**11.27**	**7.52**
WildGS-SLAM [60]	0.46	0.49	14.65
MegaSaM [27]	2.40	1.86	21.87
Vipe [17]	2.59	6.44	13.45
WildPose (Ours)	**0.39**	2.98	18.62

Table 11. **Run time and memory usage on Wild-SLAM [60].** We compute FPS by dividing the total number of frames by the total running time. The experiments are conducted on an RTX 4090 GPU.

the results of full sequences in Table 8 (TUM RGB-D) and Table 9 (7-Scenes). On the TUM RGB-D sequences, our method shows superior performance compared to the baselines, achieving the best result on 5 out of 9 sequences. Notably, compared to recent dynamic methods like MegaSaM (ATE: 0.278 m) and ViPE (ATE: 0.320 m), WildPose exhibits higher stability on the challenging `floor` sequence (ATE: 0.020 m). In this sequence, some temporally close frames contain large relative motion. This rapid change in viewpoint reduces feature overlap and induces large pixel displacements. Our resilience on this sequence stems from the broad variety of camera motion types included in our training data and the incorporation of loop closure. Furthermore, on the 7-Scenes dataset, where most methods exhibit only marginal performance differences, WildPose maintains competitive accuracy across all sequences. The consistently strong behavior across these benchmarks affirms the robustness of our method in static environments.

Addtional Evaluation Results on the ScanNet dataset [8]. To evaluate our method's robustness to motion blur and its generalization across diverse static indoor environments, we conduct additional experiments on the ScanNet dataset. The tracking results are reported in Table 10. DROID-SLAM results are sourced from GO-SLAM and Nicer-SLAM [56]. WildPose still outperforms other baselines in this static benchmark.

Runtime Analysis. We compare the average processing frame rate (FPS) and the peak GPU usage of our method against the baselines that estimate poses of the full trajectory, with detailed results presented in Table 11. For a fair comparison, MegaSaM's FPS includes depth preprocessing but excludes video depth optimization. While lightweight systems (DROID-SLAM, Vipe) are faster, they struggle with dynamic scenes. Compared to more complex dynamic frameworks like MegaSaM and WildGS-SLAM, WildPose offers supe-

Figure 7. **Limitations.** We visualize sampled images from Bonn RGB-D Dynamic Dataset [34] (`Person` sequence). The dataset has inconsistent exposure, which is challenging to our approach.

rior tracking accuracy (lowest ATE) with a higher FPS. Our peak memory usage stems from foundation models (Moge2, MASt3R) but is mitigatable via preprocessing, similar to MegaSaM, or using a distilled model.

9. Limitations

WildPose's learnable modules are trained exclusively on synthetic data. Although our curriculum is diverse, a domain gap to real-world scenarios inevitably exists. This gap is evident in sequences with unobserved phenomena, such as the significant photometric variations in the Bonn RGB-D Dynamic Dataset (Fig. 7). Our model, lacking explicit training on such exposure changes, achieves slightly worse accuracy on this dataset, particularly on the `Person` and `Person 2` sequences. Furthermore, our framework assumes known and fixed camera intrinsics, limiting its applicability to uncalibrated videos. Addressing these gaps through domain adaptation and online intrinsic refinement remains an important avenue for future work.

References

[1] Berta Bescos, José M. Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes. *RAL*, 2018. 3

[2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 6, 7

[3] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021. 2

[4] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22831–22840, 2025. 8

[5] Weirong Chen, Ganlin Zhang, Felix Wimbauer, Rui Wang, Nikita Araslanov, Andrea Vedaldi, and Daniel Cremers. Back on track: Bundle adjustment for dynamic scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4951–4960, 2025. 3

[6] Jiyu Cheng, Yuxiang Sun, and Max Q-H Meng. Improving monocular visual slam in dynamic environments: an optical-flow-based approach. *Advanced Robotics*, 2019. 3

[7] Shuhong Cheng, Changhe Sun, Shijun Zhang, and Dianfan Zhang. Sg-slam: A real-time rgb-d visual slam toward dynamic scenes with semantic and geometric information. *IEEE Transactions on Instrumentation and Measurement*, 2022. 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 6, 3

[9] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[10] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 2

[11] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE TPAMI*, 2017. 2

[12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2017. 1

[13] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014. 2

[14] Lily Goli, Sara Sabour, Mark Matthews, Marcus A Brubaker, Dmitry Lagun, Alec Jacobson, David J Fleet, Saurabh Saxena, and Andrea Tagliasacchi. Romo: Robust motion segmentation improves structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6155–6164, 2025. 3

[15] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 5, 1

[16] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. 8

[17] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. 2, 3, 5, 6, 7, 8

[18] Haochen Jiang, Yueming Xu, Kejie Li, Jianfeng Feng, and Li Zhang. Rodyn-slam: Robust dynamic dense rgb-d slam with neural radiance fields. *IEEE Robotics and Automation Letters*, 2024. 3

[19] Masaya Kaneko, Kazuya Iwami, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation. In *CVPR Workshops*, 2018. 3

[20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. 5, 1

[21] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *arXiv:2509.13414*, 2025. 2

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 2

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1

[24] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 2, 3

[25] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 1, 2, 3, 4, 7

[26] Mingrui Li, Zhetao Guo, Tianchen Deng, Yiming Zhou, Yuxiang Ren, and Hongyu Wang. Ddn-slam: Real time dense dynamic neural implicit slam. *IEEE Robotics and Automation Letters*, 2025. 3

[27] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *CVPR*, 2025. 2, 3, 6, 7, 8

[28] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 1

[29] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *NeurIPS*, 2025. 3

[30] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *CVPR*, 2024. 1, 2

[31] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 2017. 2

[32] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015. 1

[33] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *CVPR*, 2025. 2, 3, 6, 7

[34] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *iros*, 2019. 3, 6, 7, 8, 4

[35] Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. *arXiv preprint arXiv:2505.10696*, 2025. 5, 1

[36] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 4

[37] Erik Sandström, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Luc Van Gool, Martin R Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. *arXiv preprint arXiv:2405.16544*, 2024. 2, 5, 6

[38] Nicolas Schischka, Hannah Schieber, Mert Asim Karaoglu, Melih Gorgulu, Florian Grötzner, Alexander Ladikos, Nassir Navab, Daniel Roth, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic neural radiance fields. *IEEE Robotics and Automation Letters*, 2024. 3

[39] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. 2018. 3

[40] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 6, 7, 2, 3

[41] João Carlos Virgolino Soares, Marcelo Gattass, and Marco Antonio Meggiolaro. Crowd-slam: visual slam towards crowded environments using object detection. *Journal of Intelligent & Robotic Systems*, 2021. 3

[42] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 6, 7, 8, 2, 3

[43] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6, 7

[44] The AirLab. TartanAir-V2 Dataset. https://tartanair.org, 2022. Accessed: 2025-10-28. 5, 1

[45] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI*, 1991. 6

[46] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025. 1, 2, 4, 6, 7, 3

[47] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 2

[48] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 4, 5

[49] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2

[50] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 2, 4, 6, 7, 3

[51] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6112–6122, 2021. 3

[52] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *arXiv preprint arXiv:2510.07316*, 2025. 8

[53] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. Dg-slam: Robust dynamic gaussian splatting slam with hybrid pose optimization. *Advances in Neural Information Processing Systems*, 37:51577–51596, 2024. 3

[54] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. DG-SLAM: Robust Dynamic Gaussian Splatting SLAM with Hybrid Pose Optimization. In *NeurIPS*, 2024. 2

[55] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, pages 10371–10381, 2024. 8

[56] Ganlin Zhang, Erik Sandström, Youmin Zhang, Manthan Patel, Luc Van Gool, and Martin R Oswald. Glorie-slam: Globally optimized rgb-only implicit encoding point cloud slam. *arXiv preprint arXiv:2403.19549*, 2024. 2, 5, 6

[57] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024. 2

[58] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *ECCV*, pages 20–37. Springer, 2022. 3

[59] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*, pages 523–542. Springer, 2022. 3

[60] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *CVPR*, pages 11461–11471, 2025. 2, 3, 5, 6, 7, 8

[61] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025. 5, 1

[62] Liyuan Zhu, Yue Li, Erik Sandström, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. In *3DV*, 2025. 1

[63] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 2022. 1

[64] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. 2024. 2