

Dynamic Visual SLAM using a General 3D Prior

Xingguang Zhong¹ Liren Jin¹ Marija Popović² Jens Behley¹ Cyrill Stachniss^{1,3}

¹Center for Robotics, University of Bonn, Germany ²MAVLab, TU Delft, the Netherlands

³Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Abstract

Reliable incremental estimation of camera poses and 3D reconstruction is key to enable various applications including robotics, interactive visualization, and augmented reality. However, online SLAM is particularly challenging in dynamic natural environments, where scene dynamics can severely deteriorate camera pose estimation accuracy. In this work, we propose a novel monocular visual SLAM system that can robustly estimate camera poses in dynamic scenes. We leverage the complementary strengths of geometric patch-based online bundle adjustment and recent feed-forward reconstruction models. Specifically, we propose a feed-forward reconstruction model to precisely filter out dynamic regions, while also utilizing its depth prediction to enhance the robustness of the patch-based visual SLAM. By aligning depth prediction with estimated patches from bundle adjustment, we handle the inherent scale ambiguities of the batch-wise application of the feed-forward reconstruction model. Extensive experiments on multiple tasks show the superior performance of our proposed method compared to state-of-the-art approaches.

1. Introduction

Robust estimation of camera poses from a sequence of RGB images is a fundamental problem in computer vision and robotics. In particular, incremental pose estimation using only the available camera frames is essential for providing real-time pose information to downstream applications such as safe robot navigation, interactive visualization, and immersive augmented reality. Over the past decades, numerous visual odometry [14, 40, 42, 54] and simultaneous localization and mapping (SLAM) methods [4, 9, 10, 27, 38, 41] have been proposed, with recent approaches increasingly incorporating learning-based components and scene representations to enhance efficiency, robustness, and generalization [23, 39, 53].

Despite their effectiveness, most visual odometry and SLAM approaches assume a static environment, an assumption that rarely holds in real-world settings, particularly in

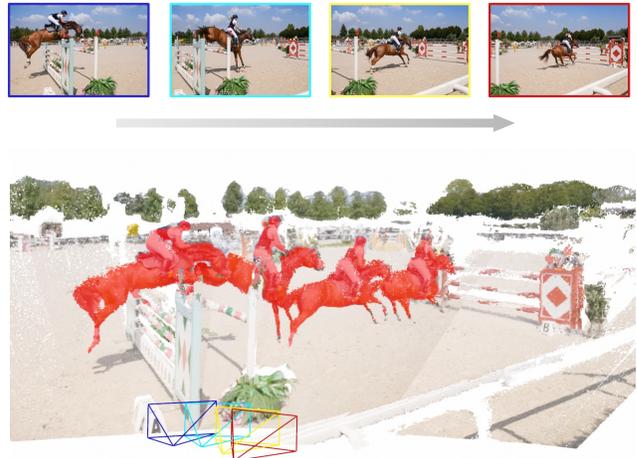


Figure 1. Given an input image sequence, our SLAM system tightly integrating a conventional geometric pipeline with a feed-forward reconstruction model can incrementally perform moving object segmentation (red masks), scale-consistent depth estimation, and camera pose estimation.

dynamic urban scenes. The presence of moving objects often leads to incorrect data associations, resulting in inaccurate camera pose estimates and degraded map quality. In addition, these methods typically perform online sparse reconstruction for computational efficiency, which limits their ability to capture fine-grained scene geometry. At the same time, the absence of geometric priors further constrains their robustness and stability under challenging conditions in a monocular setup.

Recently, feed-forward reconstruction models [44] have been proposed to directly regress dense scene structure and camera poses from sequences of RGB images, thereby eliminating the need for modular system design. Trained on large-scale multi-view datasets, these models learn rich geometric priors, achieving impressive performance in challenging scenarios where traditional visual odometry or SLAM systems often fail. Moreover, this learning-based paradigm offers extensibility, allowing the integration of additional prior knowledge such as semantics [59] or moving object segmentation [8, 55]. Motivated by this, we pose the following question: Can the priors encoded in feed-

forward reconstruction models be effectively exploited to overcome the limitations commonly faced by visual odometry and SLAM systems?

In this work, we propose a visual SLAM framework that tightly integrates a conventional geometric pipeline with feed-forward reconstruction models. We employ patch-based bundle adjustment for accurate camera pose estimation, while leveraging a feed-forward reconstruction model to detect and exclude moving objects via its learned geometric and semantic priors. This integration goes beyond simple combination and exploits the complementary strengths of both paradigms. First, a keyframe-based scale estimation, enabled by patch-based bundle adjustment, resolves the scale drift caused by incremental usage of feed-forward reconstruction models. Second, depth and confidence outputs from the feed-forward reconstruction model further robustify the patch-based bundle adjustment. Together, these components enable reliable handling of scene dynamics and accurate camera tracking, leading to strong performance across multiple downstream tasks, as demonstrated in Fig. 1 and our extensive experimental evaluation.

In summary, the main contributions of this paper are:

- We propose a novel feed-forward reconstruction model with integrated moving object segmentation, enabling our SLAM system to robustly handle dynamic scenes.
- We propose a unified framework that integrates depth prior from the feed-forward reconstruction model with geometric SLAM through an uncertainty-aware bundle adjustment mechanism, enabling superior pose estimation performance in dynamic environments.
- We address scale ambiguities inherent from incremental usage of the feed-forward reconstruction model through scale alignment with patch-based SLAM, achieving scale-consistent depth estimation over long sequences.

We released our source code ¹ to improve reproducibility.

2. Related Work

Visual SLAM in Dynamic Environments. Visual odometry [13, 14, 40, 42, 47, 54] and visual SLAM [4, 4, 9, 10, 12, 27, 38, 39, 41] are classical topics in computer vision and robotics with many advances over the last decades. However, most visual SLAM systems do not explicitly handle moving objects and instead treat them as outliers, which makes it difficult for these methods to remain stable in highly dynamic environments. To address this limitation, several works [1, 7, 20, 36] integrate object detectors or segmentation networks to distinguish common dynamic objects such as humans from the static background. However, such approaches may struggle to generalize to unseen categories of moving objects and they cannot differentiate motion states within the same object category. More re-

cent approaches [5, 6, 50] distinguish static and dynamic tracks by analyzing point trajectories over multiple frames. But they need to use the offline segmentation model like segment anything model (SAM) [17, 32] to obtain object masks, preventing them from providing an online static map for downstream tasks. WildGS-SLAM [57] builds on NeRF on-the-go [33] and WildGaussians [18], using DINOv2 [28] features and differentiable rendering to train an MLP online for dynamic object segmentation. It requires multiple observations to initialize the MLP reliably, which limits performance under fast sensor motion. A method closely related to ours, MegaSaM [21], trains a neural network on top of DROID-SLAM [39] to predict motion probabilities. Its segmentation is limited by network capacity, and as an offline method with high memory usage, it is difficult to apply in online robotic settings.

Feed-Forward Reconstruction Models. Given a set of input images, feed-forward reconstruction models can directly infer scene structure and camera poses in a single forward pass, opening new possibilities for 3D reconstruction. Starting from DUST3R [46] and MAST3R [19], which are limited to two-frame inputs, feed-forward reconstruction models have recently made significant progress. This has led to the development of offline multi-view models [44, 48, 52] and incremental models [25, 43, 45, 49, 58].

In an offline multi-view setup, VGGT [44] and π^3 [48] leverage 2D foundation model DINOv2 [28] and self-attention mechanisms to demonstrate state-of-the-art 3D reconstruction performance, even in challenging scenarios with low texture and limited overlap where traditional SLAM or structure-from-motion methods struggle. However, a key limitation is that both memory consumption and inference latency grow rapidly with increasing number of input views, which restricts their direct application in online settings. Although some methods [11, 26] adopt mechanisms to perform local map prediction through multi-view feed-forward inference followed by submap fusion, they remain susceptible to accumulated drift, particularly in the presence of moving objects. Incremental methods, such as CUT3R [45], encode local scene information into spatial memory, enabling per-frame predictions conditioned on the evolving memory state. This design allows constant time and memory consumption during inference. However, on long sequences, these methods still inevitably exhibit significant pose estimation errors and trajectory drift, falling short of the stability achieved by classical visual SLAM.

Another encouraging advantage of feed-forward models is their robustness in dynamic environments. Through training on large-scale dynamic scene datasets, models such as MonST3R [55], CUT3R [45], and π^3 [48] can stably predict camera poses and depth even in highly dynamic scenarios. This observation motivates us to leverage the prior information extracted by these models to address dynamic visual

¹Code: <https://github.com/PRBonn/Pi3MOS-SLAM>

SLAM challenges.

3. Our Approach

Given an image sequence $\mathcal{I} = \{\mathbf{I}_i \in \mathbb{R}^{H \times W \times 3}\}_{i=1}^N$ captured in a dynamic environment, the goal of this work is to robustly estimate camera poses $\mathbf{T}_i \in SE(3)$ and scale-consistent dense depth maps $\mathbf{D}_i \in \mathbb{R}^{H \times W}$ online. We build our SLAM system upon a monocular SLAM system DPV-SLAM [22, 40] and our proposed feed-forward reconstruction model π_{mos}^3 . Specifically, π_{mos}^3 can accurately segment moving objects and predict a high-quality depth map at the same time. We integrate these priors within an optical-flow based monocular SLAM framework to achieve more accurate and stable camera pose estimation over long sequences in dynamic scenes.

In the following sections, we first briefly summarize the core architecture of DPV-SLAM, then introduce π_{mos}^3 and how it achieves moving object segmentation. Finally, we present our fusion strategy and key improvements that combine the strengths of both systems.

3.1. Patch-Based Visual SLAM formulation

Given two images $\{\mathbf{I}_i, \mathbf{I}_j\}$, DPV-SLAM samples K square image patches $\mathbf{P}_{1 \dots K}^i$ from \mathbf{I}_i , and then uses a lightweight neural network to track their center pixels $\mathbf{p}_{1 \dots K}^{ij} \in \mathbb{R}^2$ in image \mathbf{I}_j . After establishing these correspondences, we perform sliding-window bundle adjustment (BA), where the optimization objective is the reprojection error, defined as:

$$\mathcal{L}_{\text{BA}} = \sum_{(i,j) \in \mathcal{E}} \sum_{k=1}^K \left\| \mathbf{p}_{1 \dots K}^{ij} - \pi(\mathbf{T}_j \mathbf{T}_i^{-1} \pi^{-1}(\mathbf{p}_k^i, d_k^i)) \right\|^2, \quad (1)$$

where \mathcal{E} denotes the set of frame pairs within the window, $\mathbf{T}_i \in SE(3)$ is the camera pose of frame i , \mathbf{p}_k^i and d_k^i are the center pixel and the inverse depth of patch \mathbf{P}_k^i in frame i , respectively. $\pi(\cdot)$ and $\pi^{-1}(\cdot)$ represent the camera projection and back-projection functions.

We further optimize \mathcal{L}_{BA} using the Levenberg-Marquardt algorithm to jointly estimate the camera pose of each frame and the depth of each patch within the sliding window. After linearizing Eq. (1), the corresponding normal equations can be formulated as follows:

$$\begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^\top & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\xi} \\ \Delta \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \quad (2)$$

where \mathbf{B} and \mathbf{C} are the Hessian matrices for frame poses and patch inverse depths, \mathbf{E} is the cross-coupling matrix, $\Delta \boldsymbol{\xi}$ and $\Delta \mathbf{d}$ are the pose and depth increments, and \mathbf{v} , \mathbf{w} are the corresponding gradients. This linear equation can be efficiently solved using the Schur complement:

$$\Delta \boldsymbol{\xi} = [\mathbf{B} - \mathbf{E} \mathbf{C}^{-1} \mathbf{E}^\top]^{-1} (\mathbf{v} - \mathbf{E} \mathbf{C}^{-1} \mathbf{w}) \quad (3)$$

$$\Delta \mathbf{d} = \mathbf{C}^{-1} (\mathbf{w} - \mathbf{E}^\top \Delta \boldsymbol{\xi}). \quad (4)$$

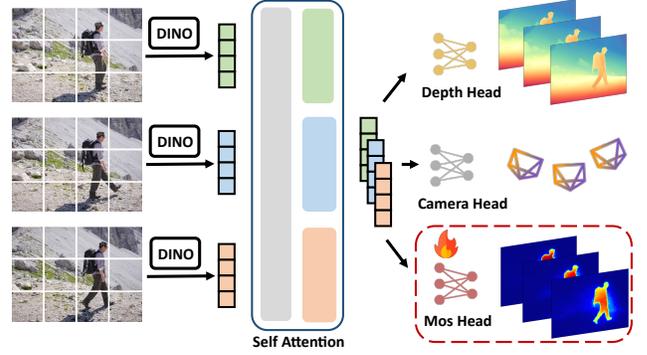


Figure 2. Network architecture of π_{mos}^3 . After patchifying input images into tokens using DINOv2, we apply alternating frame-wise and global self-attention. The fused tokens are fed into three prediction heads: a depth head for estimating depth, a camera head for predicting relative poses, and a moving object segmentation (MOS) head for producing motion probabilities. We highlight the MOS head as the key addition compared to π^3 .

3.2. Moving Object Segmentation by π_{mos}^3

Our π_{mos}^3 extends original π^3 [48] while preserving its core architecture. As shown in Fig. 2, given a set of images, π_{mos}^3 first extracts semantic features from each image using DINOv2 [28] and then employs multiple layers of alternating frame-wise and global self-attention to fuse information across images. Finally, three lightweight decoders are used to regress the relative camera pose, depth map and pixel-wise motion probability for each image.

More specifically, given an input batch of RGB images $\mathcal{I} = \{\mathbf{I}_i \in \mathbb{R}^{H \times W \times 3}\}_{i=1}^N$, the predictions of π_{mos}^3 are:

$$\pi_{\text{mos}}^3(\mathcal{I}) = (\mathbf{T}_i, \mathbf{D}_i, \mathbf{C}_i, \mathbf{M}_i)_{i=1}^N, \quad (5)$$

where $\mathbf{T}_i \in SE(3)$ denotes the camera pose, $\mathbf{D}_i \in \mathbb{R}^{H \times W}$ is the dense depth image derived from the predicted point cloud with corresponding confidence scores $\mathbf{C}_i \in \mathbb{R}^{H \times W}$, and $\mathbf{M}_i \in [0, 1]^{H \times W}$ represents the per-pixel motion probability. Note that \mathbf{D}_i is defined only up to an unknown scale, necessitating an additional scale estimation step, described in Sec. 3.4, to ensure consistent scaling.

3.3. System Overview

As illustrated in Fig. 3, once the current frame i arrives, it is paired with a set of selected historical keyframes and jointly processed by π_{mos}^3 to predict the per-pixel motion probability $\mathbf{M}_i \in \mathbb{R}^{H \times W}$ and depth map $\mathbf{D}_i \in \mathbb{R}^{H \times W}$. We then apply a threshold s_d to identify pixels with $\mathbf{M}_i < s_d$ as the static region, from which we randomly sample K patches to compute optical flow with respect to historical frames, thereby establishing inter-frame geometric constraints. This ensures that subsequent pose optimization is unaffected by moving objects. We further align the predicted depth \mathbf{D}_i

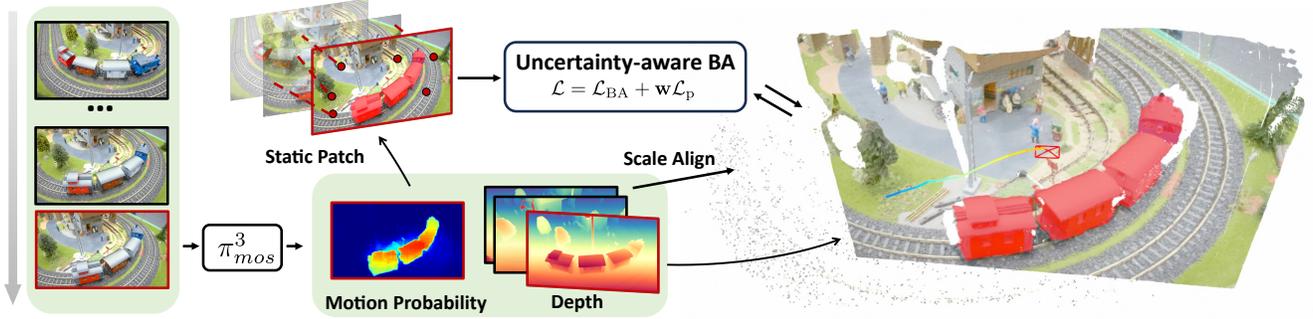


Figure 3. Overview of our SLAM system. We highlight the current frame with a red outline. It is processed together with a set of selected historical frames by our π_{mos}^3 model, which outputs per-frame motion probabilities and depth predictions. We use the current frame’s motion probabilities to initialize new patches only on the static background. The depth maps of the historical frames are used to align the scale of the current prediction via patch registration. After scale alignment, the refined depth and the optical flow constraints are fed into our BA module to jointly optimize camera poses and patch depths.

with the sparse point cloud formed by previously optimized patches to resolve the scale ambiguity of π_{mos}^3 predictions and ensure consistency with the system scale. Finally, the aligned depth and the established flow constraints are added into the sliding window, where an uncertainty-aware bundle adjustment jointly optimizes the camera poses and patch depths.

3.4. Depth Scale Estimation

We observe that the predicted depth scales remain consistent across frames within the same input batch, but this consistency does not hold across different batches. For instance, when the same image is paired with different images to form separate input batches for π_{mos}^3 inference, its predicted depth scale may vary. This necessitates aligning the depth predictions from π_{mos}^3 with a constant scale to enable incremental 3D reconstruction.

To address this scale ambiguity, we select $N - 1$ historical keyframes and feed them, together with the current frame, into π_{mos}^3 to obtain the corresponding depth maps $\{\mathbf{D}_i\}_{i=1}^N$ and confidence maps $\{\mathbf{C}_i\}_{i=1}^N$ as described in Eq. (5), where \mathbf{D}_N corresponds to the depth map of the current frame. Since we do not have a reference scale for the current frame, we refer to historical keyframes, whose scales have been determined and consistent with each other. For each historical frame processed by π_{mos}^3 , we sample \mathbf{D}_i and \mathbf{C}_i at the center location of \mathbf{P}_k^i to obtain the corresponding inverse depths $\hat{d}_k^i = 1/\hat{z}_k^i$ and confidence scores c_k^i . We want to estimate a single scale factor s in the inverse-depth domain that aligns the newly predicted depth from all selected historical frames $\{\mathbf{D}_i\}_{i=1}^{N-1}$ with their previously estimated inverse depths of patches. To this end, we optimize the scale s using:

$$s^* = \arg \min_s \sum_{i=1}^{N-1} \sum_{k=1}^K c_k^i \rho \left(d_k^i - s \hat{d}_k^i \right), \quad (6)$$

where $\rho(\cdot)$ denotes the Huber loss function, and \hat{d}_k^i represents the already scaled depth of patch \mathbf{P}_k^i in a historical frame. We initialize s using the weighted median of the ratio d_k/d_k^p , where the weights are derived from the confidence scores provided by π_{mos}^3 . This initialization yields a robust starting point that is less sensitive to outliers than a simple mean. Finally, we scale the current depth map \mathbf{D}_N with the optimized scale value s^* , thereby achieving consistent scaling during online mapping. Since the patches are selected exclusively from static regions, the scale estimation remains unaffected by moving objects.

3.5. Uncertainty-aware BA

In this section, we describe how the bundle adjustment module integrates optical flow constraints with the predicted depths from the feed-forward reconstruction model. We observe that when optical flow tracking is reliable and the camera exhibits sufficient translational motion, solving bundle adjustment can effectively estimate camera poses and inverse depth of patches. However, when the camera’s translational motion is insufficient, relying solely on optical flow constraints to compute patch depths introduces significant uncertainty, making it difficult to accurately recover the geometric structure of the environment. Introducing priors from a 3D feed-forward reconstruction model at this stage can improve the robustness of state estimation.

To incorporate depth predictions from π_{mos}^3 as priors, we introduce a depth prior loss:

$$\mathcal{L}_p = \sum_{f=1}^F \sum_{k=1}^K \left\| d_k^f - s_f \hat{d}_k^f \right\|^2, \quad (7)$$

where F is the number of frames in the optimization window, K is the number of patches per frame, \hat{d}_k^f is the inverse depth of patch \mathbf{P}_k^f predicted by π_{mos}^3 , and s_f is the scale factor for frame f that ensures scale consistency of the depth

predictions. The final loss function \mathcal{L} to be optimized combines the bundle adjustment loss \mathcal{L}_{BA} with the depth prior loss \mathcal{L}_p as follows:

$$\mathcal{L} = \mathcal{L}_{\text{BA}} + \mathbf{w}\mathcal{L}_p \quad (8)$$

$$= \mathcal{L}_{\text{BA}} + \sum_{f=1}^F w_f \sum_{k=1}^K \left\| d_k - s_f \hat{d}_k^f \right\|^2, \quad (9)$$

where $\mathbf{w} = [w_1, w_2, \dots, w_F]^T$ is a weight vector. To enhance camera tracking stability using depth priors while preventing accuracy degradation caused by over-reliance on them, we introduce an adaptive weighting strategy based on the uncertainty of state estimation, inspired by MegaSaM [21]. Specifically, when the uncertainty is high, we increase the weight w_f to stabilize the optimization. When the uncertainty is low, we reduce its influence and rely more on the original BA optimization. We quantify uncertainty by computing the variance of patch depth estimates. Following NeRF-SLAM [34, 35], the marginal covariance of the inverse depths $\Sigma_{\mathbf{d}}$ can be derived from Eq. (2) as

$$\Sigma_{\mathbf{d}} = \mathbf{C}^{-1} + \mathbf{C}^{-1} \mathbf{E}^T \Sigma_T \mathbf{E} \mathbf{C}^{-1}, \quad (10)$$

where

$$\Sigma_T = (\mathbf{B} - \mathbf{E} \mathbf{C}^{-1} \mathbf{E}^T)^{-1} \quad (11)$$

denotes the covariance of the camera poses, i.e., the inverse of the Schur complement of the full Hessian in Eq. (2) with respect to \mathbf{C} . This matrix inversion can be efficiently computed via Cholesky decomposition. The variance $\sigma_{d_j}^2$ of an individual inverse depth d_j is

$$\sigma_{d_j}^2 = [\text{diag}(\Sigma_{\mathbf{d}})]_j, \quad (12)$$

where j indexes the corresponding patch in the optimization window. Using standard nonlinear uncertainty propagation, the standard deviation of the actual depth $z_j = 1/d_j$ is $\sigma_{z_j} = \sigma_{d_j}/d_j^2$. To achieve scale-invariant uncertainty, we define the relative standard deviation as

$$\sigma_{z_j}^{\text{rel}} = \frac{\sigma_{z_j}}{z_j} = \frac{\sigma_{d_j}}{d_j}. \quad (13)$$

Finally, the frame-wise uncertainty is summarized by the median relative standard deviation $\sigma_{z,\text{med}}^{\text{rel}}$, which is mapped to a frame weight via a sigmoid function:

$$w_f = 1/(1 + \exp(-\alpha(\sigma_{z,\text{med}}^{\text{rel}} - \beta))), \quad (14)$$

where α and β are hyperparameters that control the steepness and the offset of the sigmoid curve.

In addition, during scale estimation in Sec. 3.4, we remove points whose relative standard deviation $\sigma_{z_j}^{\text{rel}}$ exceeds a threshold t_σ to improve the stability of scale estimation.

3.6. Implementation Details

We trained our π_{mos}^3 model using four datasets: three synthetic datasets, Kubric [15], Dynamic Replica [16], and Virtual KITTI 2 [3], and one real-world indoor dataset, HOI4D [24]. These datasets either provide ground-truth moving object masks directly or contain scene flow annotations from which such masks can be derived. During training, we use the weights from the original π^3 as the start-point and only train the newly introduced MOS head. The motion probability prediction is supervised using a simple binary cross-entropy loss. In total, we sampled approximately 250k images from HOI4D, 120k from Kubric, 14.5k from Dynamic Replica, and 21k from Virtual KITTI 2. Training was performed on 8 NVIDIA A40 GPUs for 48 hours in total. We provide the hyperparameter settings of our SLAM system in the supplementary.

4. Experiments

In this section, we evaluate our method against state-of-the-art approaches on multiple tasks to demonstrate its effectiveness and validate our contributions. For all the tables in this section, we highlight the best results in **bold** and the second best results are underscored.

4.1. Moving Object Segmentation

Datasets. We first discuss the moving object segmentation module in our pipeline, which serves as a crucial component of the SLAM system. Following the experimental setup in Easi3R, we conduct experiments on the moving object segmentation benchmarks DAVIS-16 [30] and the more challenging DAVIS-17 [31]. Since our goal is online robotic applications, we directly evaluate the performance of moving object segmentation from each model’s raw output, without applying any time-consuming post-processing.

Metric and Baselines. We report performance using the IoU mean (JM) and IoU recall (JR) metrics. JM measures the overall segmentation quality, while JR reflects segmentation robustness, representing the percentage of frames with IoU greater than 50 across the entire sequence. We compare our π_{mos}^3 with several state-of-the-art pose-free 4D reconstruction methods, including DUST3R [46], MonST3R [55], DAS3R [51], and Easi3R [8] using DUST3R and MonST3R as the backbone. All these methods are built upon the DUST3R framework and extend it to handle dynamic video inputs.

Results. We present the qualitative results in Tab. 3. DUST3R and MonST3R rely on optical flow estimation to distinguish moving objects, which leads to degraded performance on textureless scenes. DAS3R extends DUST3R model by fine-tuning on dynamic datasets, but yields only minor gains on DAVIS-17 and even worse performance on DAVIS-16 benchmark. Easi3R demonstrates promis-

Table 1. Camera tracking performance on the Bonn RGB-D Dynamic Dataset [29] (ATE RMSE ↓ [cm]).

Method	Balloon	Balloon2	Crowd	Crowd2	Person	Person2	Moving	Moving2	Avg.
<i>RGB-D Methods</i>									
Refusion [29]	17.5	25.4	20.4	15.5	28.9	46.3	7.1	17.9	22.38
ORB-SLAM2 [27]	6.5	23.0	4.9	9.8	6.9	7.9	3.2	3.9	6.36
DynaSLAM [1]	3.0	2.9	1.6	3.1	6.1	7.8	23.2	3.9	6.45
DDN-SLAM [20]	1.8	4.1	1.8	2.3	4.3	3.8	2.0	3.2	2.91
<i>Monocular Methods</i>									
DROID-SLAM [39]	7.5	4.1	5.2	6.5	4.3	5.4	2.3	4.0	4.91
MonST3R [55]	5.4	7.2	5.4	6.9	11.9	11.1	3.3	7.4	7.3
MegaSaM [21]	3.7	2.6	1.6	7.2	4.1	4.0	1.4	3.4	3.51
LEAP-VO [5]	24.4	5.7	6.4	8.1	15.6	15.0	8.1	24.2	13.4
WildGS-SLAM [57]	<u>2.8</u>	<u>2.4</u>	<u>1.5</u>	<u>2.3</u>	3.1	2.7	1.6	<u>2.2</u>	<u>2.36</u>
Ours	2.6	2.2	1.3	2.2	<u>3.2</u>	<u>3.0</u>	1.2	1.9	2.20

Table 2. Camera tracking performance on the Wild-SLAM MoCap Dataset [57] (ATE RMSE ↓ [cm]).

Method	Person	Racket	Table1	Table2	Umbre.	Avg.
<i>RGB-D Methods</i>						
Refusion	5.0	10.4	99.1	101.0	10.7	45.24
DynaSLAM	0.5	0.8	1.2	34.8	34.7	14.40
<i>Monocular Methods</i>						
DROID-SLAM	0.6	1.5	48.0	95.6	3.8	29.90
DynaSLAM	<u>0.4</u>	<u>0.6</u>	1.8	42.1	1.2	9.22
MonST3R	7.2	13.2	4.8	33.7	5.5	12.88
MegaSaM	3.2	1.6	1.0	9.4	<u>0.6</u>	3.16
LEAP-VO	16.0	11.9	15.7	70.8	2.7	23.4
WildGS-SLAM	0.8	0.4	<u>0.6</u>	<u>1.3</u>	0.2	<u>0.66</u>
Ours	0.2	0.4	0.2	1.1	0.2	0.42

Table 3. Moving object segmentation performance on DAVIS-16 and DAVIS-17 benchmarks.

Method	DAVIS-16		DAVIS-17	
	JM↑	JR↑	JM↑	JR↑
DUST3R [46]	42.1	45.7	35.2	35.3
MonST3R [55]	40.9	42.2	38.6	38.2
DAS3R [51]	41.6	39.0	43.5	42.1
Easi3R _{DUST3R} [8]	53.1	60.4	49.0	56.4
Easi3R _{MonST3R} [8]	<u>57.7</u>	<u>71.6</u>	<u>56.5</u>	<u>68.6</u>
ours	68.1	78.3	70.6	81.3

ing results by utilizing attention maps from DUST3R or MonST3R model without any fine-tuning on dynamic data. However, all these DUST3R-based models are limited by the lack of comprehensive spatial context, as DUST3R is originally designed for image-pair inputs. In contrast, our feed-forward reconstruction model π_{mos}^3 is trained for multi-view input and also enhanced with general semantic features provided by DINOv2 [28]. Across both benchmarks, our approach achieves the best performance in terms of both segmentation quality and robustness, with a substantial margin over all baseline approaches.

We further illustrate the quantitative results in Fig. 4. Our approach produces precise moving object masks, even in challenging scenarios with motion blur or thin structures in the input frames. These results highlight the model’s capabilities to preserve fine spatial details and maintain segmentation stability under real-world dynamic conditions.

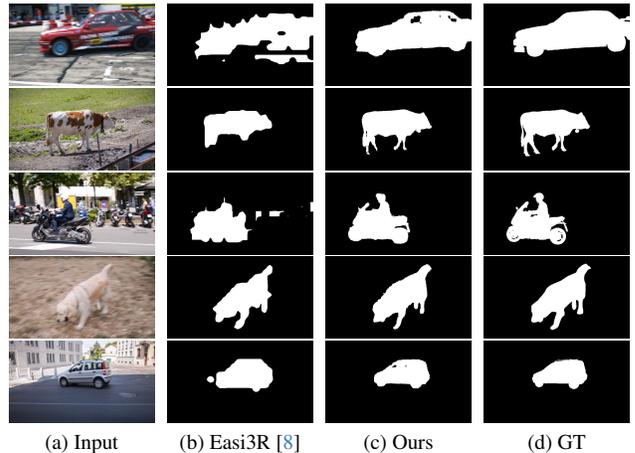


Figure 4. Qualitative comparison of moving object segmentation on the DAVIS-16 dataset. We show the moving object masks predicted from Easi3R_{MonST3R} and our π_{mos}^3 model. Our method produces more accurate segmentation results, particularly in challenging scenarios with motion blur and thin structures.

4.2. Camera Tracking

Datasets. We evaluate the camera tracking performance of our SLAM system on three dynamic scene datasets. Among them, two are real-world datasets, the Bonn RGB-D Dynamic Dataset [29] and the Wild-SLAM MoCap Dataset [29]. Both of these two datasets are recorded in single-room indoor environments and provide accurate ground-truth camera poses obtained using a motion capture system. They contain various types of moving objects beyond humans, such as tables, laptops, and umbrellas, which enable a comprehensive evaluation of the robustness of dynamic SLAM algorithms in different scenarios. The Bonn RGB-D sequences range from 400–1000 frames, and those in the Wild-SLAM dataset mostly contain 1000–2000 frames. In addition, we further use the MPI Sintel dataset [2] to evaluate our system under more challenging conditions. Sintel contains sequences rendered from 3D animated movies, covering large-scale outdoor scenes with significant depth variation, complex object motion, and

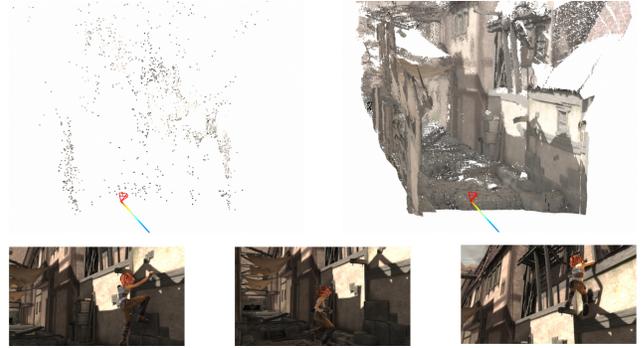
Table 4. Camera tracking performance on the Sintel [2] Dataset.

Method	ATE (cm)↓	RTE (cm)↓	RRE (deg)↓
DROID-SLAM [39]	17.5	8.4	1.91
TartanVO [47]	23.8	9.3	1.31
DytanVO [37]	13.1	9.7	1.54
MonST3R [55]	7.8	3.8	0.49
DPVO [40]	11.5	7.2	1.98
LEAP-VO [5]	8.9	6.6	1.25
BA-Track [6]	3.4	2.3	0.12
WildGS-SLAM [57]	18.2	9.4	1.57
Ours	1.9	1.0	0.11

fast camera movement with motion blur. We select 14 sequences from this dataset, each containing 20–50 frames.

Metric and Baselines. For the Bonn [29] and Wild-SLAM [57] datasets, we report the absolute trajectory error (ATE) for each sequence, which is computed as the root mean square error (RMSE) of all poses after alignment. For the Sintel dataset, following the evaluation setup in [5, 6, 56], we report the average ATE, relative translation error (RTE), and relative rotation error (RRE) of all selected sequences. The RTE and RRE are computed as the mean translation and rotation differences between consecutive estimated and ground-truth poses over all frame pairs. The baselines include approaches that do not explicitly handle moving objects, ORB-SLAM2 [27], DROID-SLAM [39], and TartanVO [47]; methods that rely on closed-world deep learning-based object detection, DynaSLAM [1] (which has both RGB-D and RGB versions) and DDN-SLAM [20]; a geometry-based approach Refusion [29]; feed-forward reconstruction model MonST3R [55]; methods that distinguish static and dynamic tracks via long-term point tracking, including LEAP-VO [5] and BA-Track [6]; WildGS-SLAM [57] and offline MegaSaM [21] approach. For MegaSaM, we report its performance from segment-aligned version implemented in WildGS-SLAM [57].

Results. We present quantitative results in Tab. 1, Tab. 2 and Tab. 4. The results show that methods without explicit handling of moving objects [27, 39, 47] and purely geometry-based systems [29] struggle to run accurately in highly dynamic environments, yielding the lowest performance. DynaSLAM [1] performs poorly when the moving objects exceed the detector’s recognition capability, as observed in the Table2 sequence of Tab. 2. MonST3R [55] segments static backgrounds via optical flow and performs sliding-window registration, but due to prediction errors from both the feed-forward reconstruction and flow models, it exhibits significant drift over long sequences. Similarly, the offline reconstruction method MegaSaM [21] cannot process an entire sequence within the limited GPU memory (16 GB in our experimental setup), the necessary segmented alignment strategy results in unstable performance on longer sequences. Leap-VO [5] and BA-Track [6]



(a) Qualitative results on the Sintel dataset [2].



(b) Qualitative results on the Wild-SLAM MoCap Dataset [29].

Figure 5. Qualitative results on different datasets. We show the sparse patch point clouds, the static dense point clouds accumulated from aligned depth maps predicted by π_{mos}^3 , and the input frames containing various dynamic objects.

rely on long-term point tracking to detect dynamic points sparsely, which becomes unreliable when dynamic objects dominate the scene or exhibit complex non-rigid motion. WildGS-SLAM [57] performs comparably to ours in small-scale indoor scenes with limited camera motion range (as shown in Tab. 1 and Tab. 2). But it relies on many overlapping views to identify distractors. Consequently, its performance degrades severely when the camera moves quickly or with little view overlap, as seen in the Sintel dataset (Tab. 4). In contrast, our method achieves the best results across all three datasets, demonstrating superior robustness and accuracy under diverse dynamic conditions. We show the system trajectories and reconstruction results on different datasets in Fig. 5. It can be seen that our method robustly filters out dynamic objects and reconstructs clean maps.

4.3. Video Depth Estimation

Datasets. To evaluate the ability of our method for generating accurate and scale-consistent dense depth maps, we conducted video depth estimation experiments. Following the setup from CUT3R [45] and π^3 [48], we used the Bonn RGB-D Dynamic Dataset [29] and the MPI Sintel Dataset [2] introduced in Sec. 4.2. For the Bonn dataset, we selected five highly dynamic sequences, each containing 400–900 frames, while for Sintel we used the same 14

Table 5. Video depth estimation results on the Sintel [2] and Bonn [29] Datasets.

Method	Sintel		Bonn	
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
<i>Offline Methods</i>				
Fast3R [52]	0.638	0.422	0.194	0.772
VGGT [44]	<u>0.299</u>	<u>0.638</u>	<u>0.057</u>	<u>0.966</u>
π^3 [48]	0.233	0.664	0.049	0.975
<i>Online Methods</i>				
DUST3R [46]	0.662	0.434	0.151	0.839
MASt3R [19]	0.558	0.487	0.188	0.765
MonST3R [55]	0.399	0.519	0.072	0.957
Spann3R [43]	0.622	0.426	0.144	0.813
CUT3R [45]	0.417	0.507	0.078	0.937
Point3R [49]	0.452	0.489	0.060	0.960
StreamVGGT [58]	<u>0.323</u>	0.657	<u>0.059</u>	<u>0.972</u>
Ours	0.287	<u>0.625</u>	0.054	0.985

sequences as in Sec. 4.2 consisting of 20–50 frames each.

Metric and Baselines. We use absolute relative error (Abs Rel) and percentage of predicted depths within a 1.25-factor of true depth ($\delta < 1.25$) as metrics. These metrics are evaluated under per-sequence scale alignment. We mainly selected a diverse set of feed-forward reconstruction models as baselines. These include the offline models Fast3R [52], VGGT [44], and π^3 [48], which process entire sequences as a single batch. While this helps maintain better scale consistency, it comes at the cost of substantial GPU memory consumption. In addition, we included several memory-efficient online models, such as pair-wise models DUST3R [46], MASt3R [19], and MonST3R [55], which fine-tuned on dynamic scenes. We also evaluated incremental methods including Spann3R [43], CUT3R [45], Point3R [49], and StreamVGGT [58], where Spann3R, CUT3R, and Point3R are extended from DUST3R, and StreamVGGT is extended from VGGT.

Results. As shown in Tab. 5, our method achieves the best performance among all online approaches on the Bonn dataset, with an Abs Rel error close to that of the best offline model, π^3 . On the Sintel dataset, our performance is slightly lower than π^3 , yet our method still attains the lowest Abs Rel among all online baselines and ranks second in $\delta < 1.25$. These results demonstrate that our approach can produce accurate, scale-consistent depth estimations and achieve performance comparable to state-of-the-art offline models.

4.4. Ablation Study

We conduct ablation experiments of camera tracking performance on the Bonn RGB-D Dynamic Dataset [29] and the Wild-SLAM MoCap Dataset [29] to evaluate the impact of each component in our system. Specifically, we analyze three key modules introduced in Sec. 3: (1) Whether to use

Table 6. Ablation study of camera tracking performance on Bonn and Wild-SLAM datasets (ATE RMSE ↓ [cm]).

	Moving mask	Depth prior	Uncertainty BA	Bonn	Wild
(a)	✗	✗	✗	4.82	1.23
(b)	✗	✓	✓	3.91	3.78
(c)	✓	✗	✗	2.67	0.98
(d)	✓	✓	✗	<u>2.52</u>	<u>0.82</u>
(e)	✓	✓	✓	2.20	0.42

the moving object mask from π_{mos}^3 to exclude dynamic regions when selecting and tracking patches; (2) Whether to incorporate the scale-aligned depth map predicted by π_{mos}^3 , i.e., whether to use the depth prior loss \mathcal{L}_p in Sec. 3.5; (3) Whether to adaptively weight the depth prior loss \mathcal{L}_p based on frame-wise uncertainty during bundle adjustment. When the Depth prior is enabled but Uncertainty-aware BA is disabled, we use a fixed loss weight of 1.0 for all frames.

It is worth noting that the patch-based SLAM framework underlying our system inherently filters out points with large reprojection errors to mitigate the influence of outliers. Even so, as shown in Tab. 6, disabling the moving object mask, as in settings (a) and (b), leads to a significant degradation in camera tracking accuracy. Moreover, incorporating the depth prior without masking may even introduce additional errors, since uncertain dynamic patches can remain in the optimization window, as reflected in the poor performance of (b) on the Wild dataset. In contrast, using only the moving object mask, as in (c), already yields strong performance, highlighting the importance of the mask from π_{mos}^3 for robust operation in dynamic environments. Adding a fixed-weight depth prior further improves accuracy slightly, while incorporating Uncertainty-aware BA achieves the best overall tracking performance.

5. Conclusion

In this paper, we present a monocular visual SLAM system designed to operate robustly in dynamic environments. Our approach integrates the proposed feed-forward reconstruction model, π_{mos}^3 , into a classical patch-based SLAM framework to jointly estimate accurate camera poses, predict scale-consistent depth maps, and generate precise moving object masks. A key contribution lies in explicitly activating the latent capability of large-scale trained feed-forward reconstruction models to disentangle dynamic objects with little additional training, enabling reliable moving object segmentation. This suggests a broader insight: when large-scale learning addresses a more fundamental task such as scene reconstruction, traditionally difficult sub-problems like moving object segmentation can emerge naturally as by-products. Overall, this work demonstrates a practical pathway toward integrating scalable feed-forward reconstruction models with SLAM systems for robust real-world deployment in dynamic environments.

Acknowledgments

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy, EXC-2070 – 390732324 – PhenoRob, and by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG). The authors thank for the access to the Marvin cluster of the University of Bonn.

References

- [1] Berta Bescos, José M. Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robotics and Automation Letters (RA-L)*, 3(4): 4076–4083, 2018. [2](#), [6](#), [7](#)
- [2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2012. [6](#), [7](#), [8](#)
- [3] Johann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint*, arXiv:2001.10773, 2020. [5](#)
- [4] 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 multi-map slam. *IEEE Trans. on Robotics (TRO)*, 37(6):1874–1890, 2021. [1](#), [2](#)
- [5] Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. LEAP-VO: Long-term Effective Any Point Tracking for Visual Odometry. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [6](#), [7](#)
- [6] 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 *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025. [2](#), [7](#)
- [7] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, and Cyrill Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019. [2](#)
- [8] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3R: Estimating Disentangled Motion from DUST3R Without Training. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025. [1](#), [5](#), [6](#)
- [9] Titus Cieslewski, Siddharth Choudhary, and Davide Scaramuzza. Data-Efficient Decentralized Visual SLAM. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. [1](#), [2](#)
- [10] Alejo Concha, Giuseppe Loianno, Vijay Kumar, and Javier Civera. Visual-Inertial Direct SLAM. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016. [1](#), [2](#)
- [11] 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](#)
- [12] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014. [2](#)
- [13] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(3):611–625, 2018. [2](#)
- [14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014. [1](#), [2](#)
- [15] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. [5](#)
- [16] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. DynamicStereo: Consistent Dynamic Depth from Stereo Videos. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. [5](#)
- [17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023. [2](#)
- [18] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. WildGaussians: 3D Gaussian Splatting in the Wild. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2024. [2](#)
- [19] Vincent Leroy, Johann Cabon, and Jerome Revaud. Grounding Image Matching in 3D with MAST3R. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2024. [2](#), [8](#)
- [20] 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 (RA-L)*, 10(5):4300–4307, 2025. [2](#), [6](#), [7](#)
- [21] 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. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. [2](#), [5](#), [6](#), [7](#)
- [22] Lahav Lipson, Zachary Teed, and Jia Deng. Deep Patch Visual SLAM. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2024. [3](#)
- [23] Lahav Lipson, Zachary Teed, and Jia Deng. Deep Patch Visual SLAM. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2025. [1](#)
- [24] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and

- Li Yi. HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5
- [25] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. SLAM3R: Real-Time Dense Scene Reconstruction from Monocular RGB Videos. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2
- [26] Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM Optimized on the SL (4) Manifold. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2025. 2
- [27] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. on Robotics (TRO)*, 33(5): 1255–1262, 2017. 1, 2, 6, 7
- [28] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafranec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shangwen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Trans. on Machine Learning Research (TMLR)*, pages 1–31, 2024. 2, 3, 6
- [29] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019. 6, 7, 8
- [30] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5
- [31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv preprint*, arXiv:1704.00675, 2017. 5
- [32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2025. 2
- [33] Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. NeRF on-the-go: Exploiting Uncertainty for Distractor-free NeRFs in the Wild. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [34] Antoni Rosinol, John J Leonard, and Luca Carlone. NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields. *arXiv preprint*, arXiv:2210.13641, 2022. 5
- [35] Antoni Rosinol, John J. Leonard, and Luca Carlone. Probabilistic Volumetric Fusion for Dense Monocular SLAM. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2023. 5
- [36] Martin Rünz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *Proc. of the Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2018. 2
- [37] Shihao Shen, Yilin Cai, Wenshan Wang, and Sebastian Scherer. Dytanvo: Joint refinement of visual odometry and motion segmentation in dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023. 7
- [38] Young-Sik Shin, Yeong Sang Park, and Ayoung Kim. Direct visual slam using sparse depth for camera-lidar system. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. 1, 2
- [39] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 6, 7
- [40] Zachary Teed, Lahav Lipson, and Jia Deng. Deep Patch Visual Odometry. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 3, 7
- [41] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios. *IEEE Robotics and Automation Letters (RA-L)*, 3(2):994–1001, 2018. 1, 2
- [42] Lukas von Stumberg, Vladyslav Usenko, and Daniel Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. 1, 2
- [43] Hengyi Wang and Lourdes Agapito. 3D Reconstruction with Spatial Memory. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2025. 2, 8
- [44] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2, 8
- [45] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 7, 8
- [46] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUst3R: Geometric 3D Vision Made Easy. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5, 6, 8
- [47] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. TartanVO: A Generalizable Learning-based VO. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2020. 2, 7
- [48] 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](#), [3](#), [7](#), [8](#)
- [49] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D Reconstruction with Explicit Spatial Pointer Memory. *arXiv preprint*, arXiv:2507.02863, 2025. [2](#), [8](#)
- [50] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: 3D Point Tracking Made Easy. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025. [2](#)
- [51] Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao. DAS3R: Dynamics-Aware Gaussian Splatting for Static Scene Reconstruction. *arXiv preprint*, arXiv:2412.19584, 2024. [5](#), [6](#)
- [52] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D Reconstruction of 1000+ Images in One Forward Pass. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. [2](#), [8](#)
- [53] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018. [1](#)
- [54] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#), [2](#)
- [55] 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. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2025. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [56] Zhuotong Zhang, Forrester Cole, Zhengqi Li, Noah Snavely, Michael Rubinstein, and William T. Freeman. Structure and Motion from Casual Videos. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022. [7](#)
- [57] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Armeni Iro. WildGS-SLAM: Monocular Gaussian Splatting SLAM in Dynamic Environments. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. [2](#), [6](#), [7](#)
- [58] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4D Visual Geometry Transformer. *arXiv preprint*, arXiv:2507.11539, 2025. [2](#), [8](#)
- [59] Lojze Zust, Yohann Cabon, Juliette Marrie, Leonid Antsfeld, Boris Chidlovskii, Jerome Revaud, and Gabriela Csurka. PanSt3R: Multi-view Consistent Panoptic Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025. [1](#)