Back on Track: Bundle Adjustment for Dynamic Scene Reconstruction

Weirong Chen[1,2] Ganlin Zhang[1,2] Felix Wimbauer[1,2] Rui Wang[4] Nikita Araslanov[1,2]
Andrea Vedaldi[3] Daniel Cremers[1,2]

[1]TU Munich [2]Munich Center for Machine Learning [3]University of Oxford [4]Microsoft

Abstract

Traditional SLAM systems, which rely on bundle adjustment, struggle with the highly dynamic scenes commonly found in casual videos. Such videos entangle the motion of dynamic elements, undermining the assumption of static environments required by traditional systems. Existing techniques either filter out dynamic elements or model their motion independently. However, the former often results in incomplete reconstructions, while the latter can lead to inconsistent motion estimates. Taking a novel approach, this work leverages a 3D point tracker to separate camera-induced motion from the observed motion of dynamic objects. By considering only the camera-induced component, bundle adjustment can operate reliably on all scene elements. We further ensure depth consistency across video frames with lightweight post-processing based on scale maps. Our framework combines the core of traditional SLAM—bundle adjustment—with a robust learning-based 3D tracker. Integrating motion decomposition, bundle adjustment, and depth refinement, our unified framework, BA-Track, accurately tracks camera motion and produces temporally coherent and scale-consistent dense reconstructions, accommodating both static and dynamic elements. Our experiments on challenging datasets reveal significant improvements in camera pose estimation and 3D reconstruction accuracy.

Figure 1. **Framework preview.** Given a casual input video, BA-Track uses a 3D tracker to separate camera-induced motion from the total observed motion, enabling bundle adjustment to process both static and dynamic points. Using the aligned sparse point tracks from bundle adjustment, we refine the dense depth maps, producing a globally consistent dynamic scene reconstruction.

1. Introduction

With the increasing prevalence of casual videos, reconstructing scenes containing moving objects has become essential for applications like augmented reality and robotics. Traditional methods for structure-from-motion (SfM) [1] and simultaneous localization and mapping (SLAM) [11] excel at recovering camera poses and scene geometry. However, they leverage the epipolar constraint and thus apply only to static environments. Dynamic scenes with moving objects introduce ambiguities that violate the epipolar constraint. Resolving these ambiguities by analytical constraints or statistical outlier filtering alone is infeasible due to the complexity and diversity of real-world object motion. This challenge motivates a hybrid approach that combines traditional optimization with deep priors. Existing hybrid methods often address dynamic elements by detecting and removing them from optimization, leading to incomplete reconstructions or missing details associated with moving objects [2, 55]. Another approach is to rely on geometric cues, such as monocular depth models for per-frame reconstruction, but this requires additional optimization to achieve a globally aligned 3D reconstruction [23]. This proves challenging in practice due to the inconsistency between estimated camera pose and depth priors across different frames, which can result in misaligned

Project page: wrchen530.github.io/projects/batrack

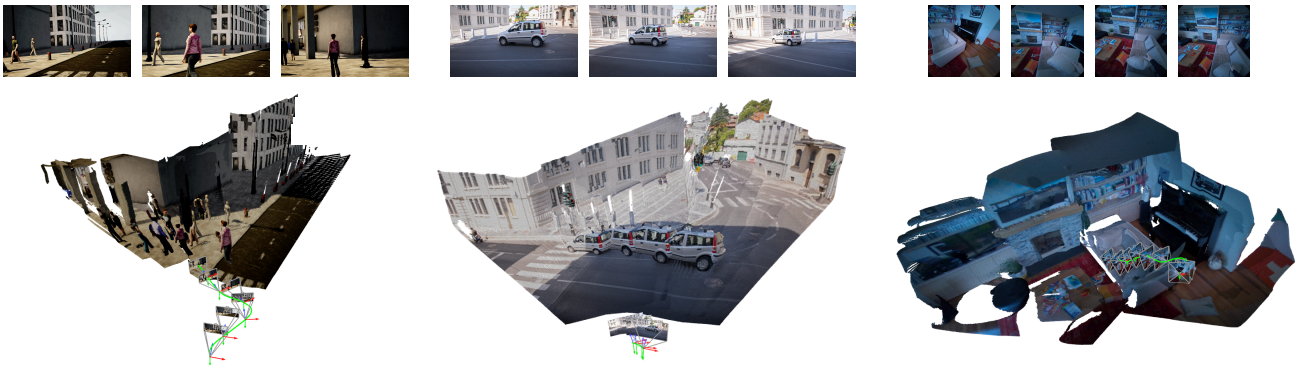

Figure 2. **Dynamic scene reconstruction results on Shibuya [34], DAVIS [21], and Aria Everyday Activities [29] datasets.** By leveraging sparse dynamic SLAM and global refinement, BA-Track achieves consistent dense 3D reconstructions across diverse dynamic scenes.

depth maps when integrating them into a coherent 3D scene.

We propose BA-Track, a framework that jointly reconstructs static and dynamic scene geometry from casual video sequences (see Fig. 1). Following traditional SfM and SLAM approaches, BA-Track extends the time-honored bundle adjustment (BA) [45] to dynamic scene reconstruction. The novel insight is to integrate a learning-based 3D tracker to decouple the motion of dynamic elements from the observed motion. By disentangling the camera-induced motion component of dynamic objects, BA-Track reconstructs 3D trajectories of dynamic points as if they were stationary relative to their local reference frame. As a result, the epipolar constraint becomes applicable to *all points*, dynamic or static, enabling robust operation of bundle adjustment.

To effectively decouple motion, we incorporate monocular depth priors with a 3D tracker. The tracker learns valuable 3D priors that help distinguish between camera- and object-based motion. Although such learning-based priors prove extremely helpful for reconstructing dynamic scenes, the scale of the depth maps exhibits temporal and spatial inconsistency. To address this issue, we develop a global refinement module, which leverages the accurate but sparse geometry from BA to refine the dense depth maps. As illustrated in Fig. 2, BA-Track achieves compelling reconstruction quality of both static and dynamic scene elements.

In summary, our framework consists of three components: *(1)* a robust 3D tracking *front-end*, which decouples camera-induced motion from observed (total) motion; *(2)* a bundle adjustment *back-end* for accurate pose and depth estimation, which leverages camera-induced motion for all points; and *(3) global refinement*, to achieve dense, scale-consistent depth. Our learnable pipeline extends epipolar geometry to dynamic scene reconstruction, leveraging the power of established optimization techniques. Our experiments demonstrate significant improvements in camera motion estimation and 3D scene reconstruction, even in challenging scenes.

2. Related Work

Point Tracking. Point tracking techniques estimate the 2D positions of one or more query points across a sequence of images. Revisiting the idea of video representation as moving particles [36], PIPs [16] takes a learning-based approach with a feed-forward network that iteratively refines point trajectories using multi-scale local features. TAP-Net [9] takes a different direction by leveraging global correlations to directly compute trajectories, thereby eliminating the need for iterative refinement. Subsequent research has explored various techniques to improve accuracy and efficiency, including inter-track relationships [18], temporal refinements [10], nearest-neighbor interpolation [24], and bidirectional 4D correlation [7]. Moving beyond 2D tracking, recent methods such as SpatialTracker [51] and SceneTracker [2] elevate the task into 3D space by incorporating depth priors.

Our approach extends 3D tracking by introducing a novel motion decoupling strategy that separates camera motion from object motion, enabling more effective integration with bundle adjustment.

Monocular Visual Odometry. Monocular visual odometry (VO) estimates camera motion using RGB image sequences. Traditional methods can be divided into feature-based and direct approaches. Feature-based methods detect and match keypoints across frames, using epipolar geometry and bundle adjustment for pose estimation [8, 30]. Direct methods bypass keypoints, instead optimizing photometric consistency across frames using full-image information [11, 12, 15]. Recent advances have shifted towards learning-based approaches that employ neural networks to enhance feature extraction and pose estimation [4, 41–44, 49, 50]. For example, TartanVO [49] regresses motion from optical flow using deep networks, while DPVO [44] integrates differentiable bundle adjustment with iterative correspondence updates. Dynamic environments introduce additional challenges, as moving objects can interfere with motion estimation. To improve robustness, some methods filter dynamic regions using masks [2, 39], while others apply trajectory filtering

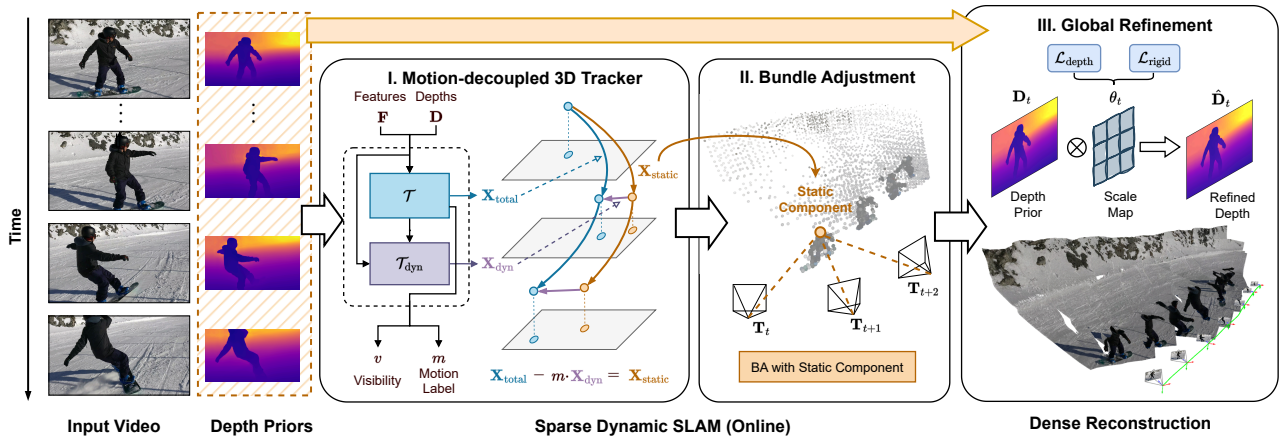

Figure 3. **Overview of the BA-Track framework.** Given a temporal window, we compute image features $\mathbf{F}$ and depth features $\mathbf{D}$. Our 3D tracker estimates local 3D tracks, visibility, dynamic labels, and decouples the static (camera-induced) motion of each query point. Operating on the static motion components, bundle adjustment (BA) recovers the camera poses and global tracks. The final refinement stage aligns the monocular depth priors with sparse BA estimates to ensure a temporally consistent and dense reconstruction.

to discard motion outliers [6, 55].

Dynamic Scene Reconstruction. Dynamic scene reconstruction aims to jointly recover geometry and motion in environments with moving objects. Traditional methods such as SfM [37] and MVS [38] assume static scenes, limiting their applicability in dynamic settings. Regression-based approaches leverage large-scale training to estimate depth from single RGB images [3, 20, 32, 35, 52], while video-based extensions further improve temporal coherence by incorporating multi-frame cues, depth consistency, or diffusion priors [14, 17, 25, 28]. More recent approaches extend beyond depth to regress scene coordinates, modeling dynamic scenes through per-frame point maps [13, 40, 48, 53]. To enforce geometric consistency, several methods combine learning with optimization, jointly refining scene structure and camera motion [23, 26, 53, 54]. For example, MonST3R [53] uses a point map representation and refines global poses by filtering dynamic regions with an optical flow-based motion mask. TracksTo4D [19], closely related to our work, reconstructs dynamic scenes from 2D point trajectories by directly regressing point cloud bases and camera poses. In contrast, our method decouples static and dynamic point motions and jointly optimizes them via bundle adjustment, achieving better robustness.

3. BA-Track

BA-Track consists of three stages (see Fig. 3): *(i)* A learning-based front-end decouples camera-induced motion from observed motion via a 3D tracker. *(ii)* A bundle adjustment back-end recovers camera pose and sparse 3D geometry for both static and dynamic points using camera-induced motion. *(iii)* A lightweight refinement stage leverages the sparse tracks to produce dense and temporally consistent depth.

3.1. Stage I: Motion-Decoupled 3D Tracker

Classical correspondence-based methods struggle to recover the 3D structure of dynamic objects from monocular videos due to violations of the epipolar constraint. As a result, previous work requires detecting and filtering out dynamic regions to ensure that only static points are used for camera tracking and reconstruction [2, 6, 55]. Here, we explore an alternative approach. Instead of discarding dynamic points, we infer the camera-induced component of their motion. As shown in Fig. 4, the observed 2D motion comprises camera-induced (static) and object-induced (dynamic) components. By accurately estimating the static component, dynamic points become pseudo-static in their local reference frame. Leveraging the static component, classical optimization, such as bundle adjustment, can operate seamlessly and requires no special treatment of dynamic points.

However, accurately estimating the camera-induced motion is non-trivial and requires strong motion priors. To effectively learn motion priors, we integrate off-the-shelf depth cues and develop a 3D tracking front-end. The tracker exploits the temporal context, while the additional depth cues enhance the tracker's 3D reasoning and improve its capacity to estimate the camera-induced motion. Furthermore, we empirically observe that training a single network to estimate the camera-induced component is suboptimal. Instead, we employ two networks: the first predicts the observed, "total" motion, while the other predicts the dynamic component corresponding to the object-induced motion. We further discuss and empirically verify this design choice in Sec. 4.4.

Formulation. Considering a temporal window of RGB frames $\mathbf{I} = (I_1, \ldots, I_S)$ with resolution $H \times W$ and known camera intrinsics, we use a monocular depth estimation network [3] to obtain their depth maps, $\mathbf{D} = (D_1, \ldots, D_S)$.

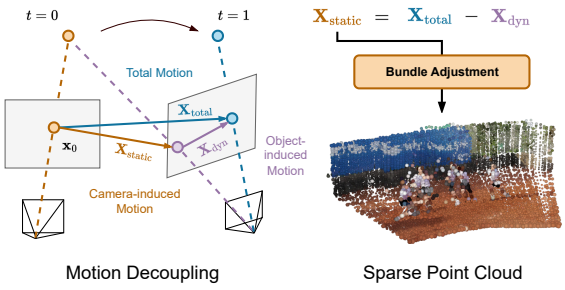

Figure 4. **Illustration of motion decoupling.** We decompose the total observed point motion into a static component (induced by the camera motion) and a dynamic component (induced by object motion). The static component is then used by bundle adjustment to provide camera poses and sparse reconstruction.

Given a 2D query point $\mathbf{x}^t = (u^t, v^t)$ in frame t, we sample its depth value $d^t = D_t[\mathbf{x}^t]$[1] from the initial depth map, obtaining a 3D query point $X^t = (\mathbf{x}^t, D_t[\mathbf{x}^t])$. Since the intrinsics are known, we represent the 3D point using 2D coordinates along with the depth value. The 3D tracker aims to estimate the corresponding 3D trajectory $\{X^t(s) \mid s \in \{1, \dots, S\}\}$ for any timestep in the window. For the remainder of this section, we omit the superscript t indicating the query's frame of reference.

To extract relevant features for tracking, we pass each RGB image and its corresponding depth map through a CNN-based feature encoder [51], and obtain C-dimensional feature maps, $\mathbf{F} = \{F_1, \dots, F_S\}$, where $F_s \in \mathbb{R}^{C \times H/4 \times W/4}$. We then define a point feature vector $f(s)$ associated with a 2D point $\mathbf{x}$, which encapsulates appearance, spatial, and depth information from the RGB-D input at each timestep s.

Motion Decoupling. Our front-end consists of two transformer networks sharing the architecture with CoTracker [18]. The tracker $\mathcal{T}$ predicts the observed motion X_{total}, visualized in blue in Fig. 3. The dynamic tracker $\mathcal{T}_{\text{dyn}}$ is two times shallower than $\mathcal{T}$ for efficiency, and predicts the dynamic component X_{dyn} of X_{total}. The tracker $\mathcal{T}$ also learns point visibility $v \in [0,1]^S$ and a static-dynamic label $m \in [0,1]$. Point visibility reflects occlusions, while the static-dynamic label indicates if the point belongs to a dynamic object.

After computing the total motion X_{total} from the tracker $\mathcal{T}$, we use the dynamic tracker $\mathcal{T}_{\text{dyn}}$ to predict the dynamic motion component. For motion decoupling, we decompose X_{total} into the camera motion X_{static} (visualized in orange in Fig. 3) and the object motion X_{dyn}, weighted by the dynamic label m:

$$X_{\text{static}} = X_{\text{total}} - m \cdot X_{\text{dyn}}, \tag{1}$$

where m acts as a gating factor. Observe that the motion of static points ($m = 0$) coincides with the observed motion. Conversely, the camera-induced motion of dynamic points ($m = 1$) results from subtracting their motion from the observed flow. By decoupling the motion into static and dynamic parts, we can effectively track dynamic points as if they were static in their local frame of reference. This novel mechanism allows the tracker to recover the 3D structure of dynamic objects with bundle adjustment in a seamless fashion, as we elaborate in Sec. 3.2.

Training. Under the hood, each transformer takes an initial 3D query location, extracts its point feature, and iteratively updates the point location and the corresponding point feature. Each iteration leverages the local context of the query point, aggregated from the multi-scale feature correlations [18]. We further incorporate depth map features into the context aggregation to embed 3D context. Let us denote the point location and its feature vector after the k-th iteration as $(X^{(k)},\ f^{(k)})$ for tracker $\mathcal{T}$, and $(X^{(k)}_{\text{dyn}}, f^{(k)}_{\text{dyn}})$ for the dynamic tracker $\mathcal{T}_{\text{dyn}}$. Correspondingly to Eq. (1), we obtain the static component for each iteration:

$$X^{(k)}_{\text{static}} = X^{(k)}_{\text{total}} - m \cdot X^{(k)}_{\text{dyn}}. \tag{2}$$

We supervise $\mathcal{T}$ and $\mathcal{T}_{\text{dyn}}$ with ground-truth total 3D point trajectories and the static point trajectories from synthetic data. To improve training convergence, we provide supervision for every iteration of the tracker,

$$\mathcal{L}_{3D}(X^{(k)}) = \gamma^{K-k} \left\| X^{(k)} - X^{\text{GT}} \right\|_1, \tag{3}$$

$$\mathcal{L}_{3D}(X^{(k)}_{\text{static}}) = \gamma^{K-k} \left\| X^{(k)}_{\text{static}} - X^{\text{GT}}_{\text{static}} \right\|_1, \tag{4}$$

where K is the number of iterations and γ is a hyperparameter (set empirically to 0.8).

To learn the visibility and the static-dynamic label, we use binary cross-entropy, defined as

$$\begin{aligned} \mathcal{L}_{vis} &= (1 - v^*) \log (1 - v) + v^* \log v, \\ \mathcal{L}_{dyn} &= (1 - m^*) \log (1 - m) + m^* \log m, \end{aligned} \tag{5}$$

where $v^* \in \{0,1\}^S$ and $m^* \in \{0,1\}$ are the ground-truth labels. Overall, the total loss for our 3D tracker is

$$\mathcal{L}_{total} = \mathcal{L}_{3D} + w_1 \mathcal{L}_{vis} + w_2 \mathcal{L}_{dyn}, \tag{6}$$

where w_1 and w_2 are scalar hyperparameters.

3.2. Stage II: Bundle Adjustment

We now employ our 3D tracker above to recover the camera poses and a sparse 3D structure of dynamic scenes. Note that Sec. 3.1 exemplified tracking of a single query point, but our window-based 3D tracker can handle *multiple* query points within a temporal window. Here, we assume to have a set of 3D trajectories represented by their static 3D components X_{static}, as well as their per-frame visibility v and the static-dynamic label m. To simplify the notation, we use X to denote the static component X_{static} in the following.

[1] $F[\mathbf{x}]$ denotes bilinear sampling from F at location $\mathbf{x}$.

Given a video sequence of length $L > S$, we extract a set of N query points $\mathbf{X}^t = (X_1^t, \ldots, X_N^t)$ from each frame $t \in \{1, \ldots, L\}$. Each query point $X_n^t = (u_n^t, v_n^t, d_n^t)$ represents the 2D pixel coordinates $\mathbf{x}_n^t = (u_n^t, v_n^t)$ of the point and its initial depth $d_n^t = D_t[\mathbf{x}_n^t]$. Using our 3D tracker, we estimate a 3D trajectory of each query point X_n^t within a local window of $S = 2S' + 1$ frames. Specifically, we obtain the trajectory as $\mathbf{X}_n^t = [X_n^t(1), \ldots, X_n^t(S)]$ defined over the window $I_{t-S':t+S'}$. Using a sliding window approach, we collect the 3D trajectories of all query points over the full sequence. The result is a local 3D trajectory tensor $\mathbf{X} \in \mathbb{R}^{L \times N \times S \times 3}$.

Since our tracker can estimate the static component of each point trajectory, we can leverage the classical BA optimization framework [45] for camera pose estimation and depth refinement. This approach enhances the robustness of camera pose estimation, since it isolates the object motion and reduces the impact of dynamic points on pose accuracy. In our BA framework, the optimization variables include the per-frame camera poses $\{\mathbf{T}_t \in SE(3)\}$ and the refined depths of the query points $\mathbf{Y} \in \mathbb{R}^{L \times N \times 1}$. Each 3D trajectory $\mathbf{X}$ connects a query point $\mathbf{x}_n^i$ extracted from its source frame i to its corresponding target locations in all other frames j.

The corresponding reprojection is defined as:

$$\mathcal{P}_j(\mathbf{x}_n^i, y_n^i) = \Pi(\mathbf{T}_j \mathbf{T}_i^{-1} \Pi^{-1}(\mathbf{x}_n^i, y_n^i)), \tag{7}$$

where Π is the pinhole projection function and Π^{-1} is the inverse projection. The RGB-D bundle adjustment is formulated as:

$$\begin{aligned} \underset{\{\mathbf{T}_t\},\{\mathbf{Y}\}}{\arg\min} \sum_{|i-j| \le S} \sum_n W_n^i(j) \left\| \mathcal{P}_j(\mathbf{x}_n^i, y_n^i) - X_n^t(j) \right\|_\rho \\ + \alpha \left\| y_n^i - d(\mathbf{X}_n^i) \right\|^2, \end{aligned} \tag{8}$$

where $\|\cdot\|_\rho$ is the Huber loss and α balances the reprojection loss and depth consistency. The confidence weight $W_{j,n}^i$ encapsulates the visibility of point $\mathbf{x}_n^i$ in frame j and its dynamic label by $W_n^i(j) = v_n^i(j) \cdot (1 - m_n^i)$. We solve Eq. (8) efficiently using the Gauss-Newton method with Schur decomposition [45].

3.3. Stage III: Global Refinement

Bundle adjustment recovers the camera poses and adjusts the depths of the query points in their source frames of reference. However, BA processes only a sparse set of query points, so the 3D positions of the other points in the depth maps remain unaffected. To ensure consistency of the depth maps with the sparse set of accurate 3D tracks estimated by BA, we need global refinement. Towards this goal, we introduce a function $\mathcal{H}_\theta : D \to \hat{D}$, where $\hat{D}$ is the refined depth map and θ is the parameter set. Global refinement jointly optimizes the function parameters θ and sparse 3D trajectories to achieve consistent geometry.

Our approach enforces two types of consistency: depth consistency and scene rigidity. The depth consistency loss encourages the refined depth map to match the sparse 3D trajectories within each frame, ensuring coherent alignment of sparse and dense depths. The scene rigidity loss maintains relative 3D distances between static trajectories across frames, preserving the rigidity of static structures.

Inspired by previous work [23], we define a 2D scale grid of resolution $H_g \times W_g$ for each frame, $\theta_t \in \mathbb{R}^{H_g \times W_g}$. The grid has a coarser resolution compared to the original image resolution $H \times W$. Its goal is to scale the depth estimates D_t: For each 2D query point $\mathbf{x}$ in frame t, we determine the associated depth scale using bilinear sampling on the grid θ_t. The refined depth is then

$$\hat{D}_t[\mathbf{x}] = \theta_t[\mathbf{x}] \cdot D_t[\mathbf{x}], \tag{9}$$

where $\theta_t[\mathbf{x}]$ serves as a scaling factor to adjust the initial depth $D_t[\mathbf{x}]$. We further introduce a local scale variable σ_n^t for each 3D point X_n^t from the local 3D trajectory tensor $\mathbf{X}$, allowing us to refine sparse trajectories alongside the adjustments in the dense depth map. This combination of global and local refinements accounts for both large-scale depth variations and fine-grained details. The depth consistency loss optimizes θ and σ via:

$$\mathcal{L}_{\text{depth}} = \sum_{t,n} \left\| \theta_t[\mathbf{x}_n^t] \cdot D_t[\mathbf{x}_n^t] - \sigma_n^t d_n^t \right\|, \tag{10}$$

while the scene rigidity loss optimizes θ as:

$$\begin{aligned} \mathcal{L}_{\text{rigid}} = \sum_{|i-j|<S} \sum_{(a,b) \in N} W_{\text{static}} \big(\left\| P_a^i(j) - P_b^i(j) \right\| - \\ \left\| P_a^i - P_b^i \right\| \big). \end{aligned} \tag{11}$$

Here,

$$P_k^i(j) = \Pi^{-1}(\mathbf{x}_k^i, \hat{D}_i[\mathbf{x}_k^i]), \quad k \in \{a, b\}, \tag{12}$$

is the 3D position derived from back-projecting 2D points using the refined dense depth. The weight W_{static} filters out the dynamic points as $W_{\text{static}} = (1 - m_a^i) \cdot (1 - m_b^i)$. Intuitively, $\mathcal{L}_{\text{rigid}}$ enforces constant distances between arbitrary static points a and b in the target frame j and the source frame i. We optimize the variables using the Adam optimizer [22].

4. Experiments

4.1. Implementation Details

We train our 3D tracker on the TAP-Vid-Kubric training set [9], which consists of 11,000 sequences, each with 24 frames. The training process uses the AdamW optimizer [27]

Category	Method	MPI Sintel [5]			AirDOS Shibuya [34]			Epic Fields [46]		
		ATE	RTE	RRE	ATE	RTE	RRE	ATE	RTE	RRE
w/ intrinsics	DROID-SLAM [43]	0.175	0.084	1.912	0.256	0.123	0.628	1.424	0.130	2.180
	TartanVO [49]	0.238	0.093	1.305	0.135	0.030	0.249	1.490	0.121	1.548
	DytanVO [39]	0.131	0.097	1.538	0.088	0.026	0.251	1.608	0.119	1.556
	DPVO [44]	0.115	0.072	1.975	0.146	0.091	0.387	0.394	0.078	**0.734**
	LEAP-VO [6]	0.089	0.066	1.250	0.031	0.111	**0.124**	0.486	0.071	1.018
w/o intrinsics	Robust-CVD [23]	0.360	0.154	3.443	—	—	—	—	—	—
	ParticleSfM [55]	0.129	0.031	0.535	0.275	0.155	0.750	—	—	—
	CasualSAM [54]	0.141	0.035	0.615	0.209	0.202	0.620	—	—	—
	MonST3R [32]	0.108	0.042	0.732	(0.512)	(0.075)	(0.566)	—	—	—
w/ intrinsics	BA-Track (Ours)	**0.034**	**0.023**	**0.115**	**0.028**	**0.009**	0.150	**0.385**	**0.066**	1.029

Table 1. **Camera pose evaluation results on Sintel [5], Shibuya [34], and Epic Fields [46] datasets.** $(\cdot)$ denotes evaluation on subsequence due to memory constraints. Our method shows better results on ATE compared to other competitive baselines.

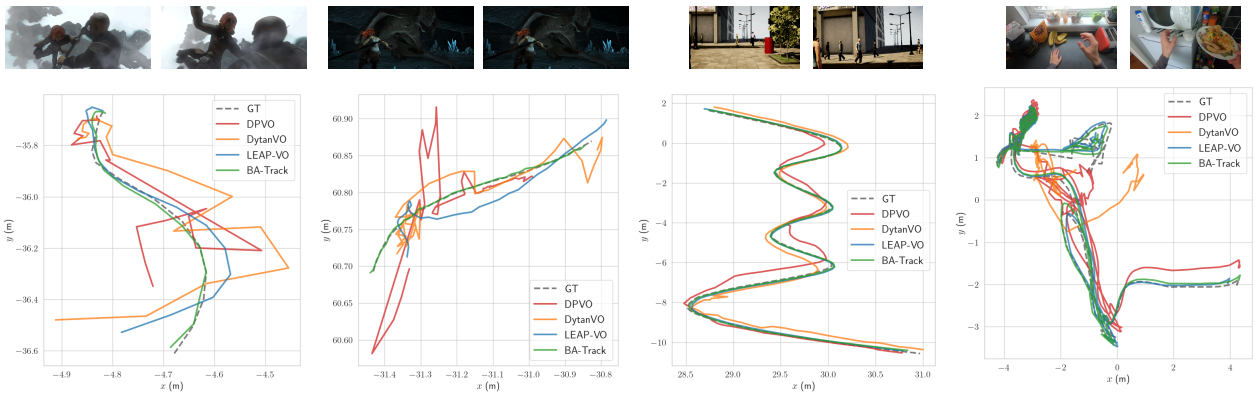

Figure 5. **Qualitative camera pose estimation results on Sintel [5], Shibuya [34], and Epic Fields [46].** Visualizations demonstrate that our method achieves more robust and accurate camera trajectories in challenging dynamic scenes.

with a learning rate of 3×10^{-4} for a total of 100,000 steps, performed on 4 NVIDIA A100 GPUs. During training, we set the weights for visibility and dynamic labels to $w_1 = 5$ and $w_2 = 5$, employ $K = 4$ iterative updates, a window size of $S = 12$, and a total of $N = 512$ query points. For BA, we use a window of 15 frames and apply 4 Gauss-Newton updates per window. We set $\alpha = 0.05$ to balance reprojection and depth loss in Eq. (8). For BA weight filtering, we set $\delta_v = 0.9$ and $\delta_m = 0.9$. For further implementation details, please refer to our supplementary material.

4.2. Camera Pose Evaluation

Datasets. We use three datasets for camera pose evaluation. MPI Sintel [5] provides sequences from 3D animated films featuring complex, fast object motion and exposure effects like motion blur. We use MPI Sintel to evaluate dynamic VO performance, with sequences spanning 20 to 50 frames. AirDOS Shibuya [34] includes sequences of 100 frames tracking over 30 individuals in two scene types. "Road Crossing" features diverse object motion, while "Standing Humans" contains dominant camera motion. EPIC Fields [46] includes egocentric videos with dynamic interactions, such as moving hands with kitchen tools, and is therefore ideal for studying long dynamic scenarios. We randomly select 8 sequences with reliable ground-truth trajectories, sampling every third frame of the first 3000 frames, yielding 1000 frames per sequence.

Metrics. We adopt three metrics. Absolute Translation Error (ATE) calculates the root mean square error between the estimated and ground-truth trajectories. Relative Translation Error (RTE) and Relative Rotation Error (RRE) measure translation (in meters) and rotation (in degrees) errors over a set distance, with both metrics averaged across all poses. We perform all evaluations after $\mathrm{Sim}(3)$ Umeyama alignment [47] with the ground-truth trajectory.

Results. We compare BA-Track against learning-based SLAM and VO approaches, such as DROID-SLAM [43], ParticleSfM [55], DytanVO [39], and LEAP-VO [6]. Additionally, we compare BA-Track to test-time optimization techniques, such as Robust-CVD [23] and CasualSAM [54],

Method	MPI Sintel [5]		AirDOS Shibuya [34]		Bonn [31]	
	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
Robust-CVD [23]	0.703	47.8	—	—	—	—
CasualSAM† [54]	0.387	54.7	0.261	79.9	0.169	73.7
MonST3R* [20]	0.335	58.5	(0.208)	(71.2)	0.063	96.4
ZoeDepth [3]	0.467	47.3	0.571	43.8	0.087	94.8
BA-Track with ZoeDepth (Ours)	0.408	54.1	0.299	55.1	0.084	95.0

† test-time network finetuning; * depth from image pairs; the numbers in $(\cdot)$ denote results on a sub-sequence due to memory constraints.

Table 2. **Depth evaluation results on Sintel [5], Shibuya [34], and Bonn [31] datasets.** Our method achieves improved depth accuracy compared to the depth priors from ZoeDepth, demonstrating the effectiveness of our depth refinement.

which typically require several hours of processing time.

Table 1 shows a quantitative comparison of our method against state-of-the-art approaches across three benchmarks. Our approach consistently delivers superior performance in ATE and remains competitive in RTE and RRE. On the MPI Sintel dataset, our method significantly outperforms all baselines. As Fig. 5 illustrates, BA-Track produces notably more accurate trajectory estimates, particularly in dynamic scenes where other methods struggle. On the challenging Epic Fields dataset, characterized by rapid dynamic content and complex camera motion, BA-Track maintains compelling VO accuracy. Overall, BA-Track demonstrates remarkable accuracy across all benchmarks with high memory efficiency. By contrast, prior work like MonST3R can only process up to 90 frames at once on a 48GB GPU.

4.3. Depth Evaluation

Datasets. MPI Sintel [5] and AirDOS Shibuya [34] are selected as both provide dense ground-truth depth maps. We add a real-world dataset, Bonn RGB-D, which captures indoor activities such as object manipulation across 24 dynamic sequences. Following Zhang et al. [53], we select 5 dynamic sequences, each consisting of 110 frames.

Metrics. We adopt the absolute relative error (Abs Rel) and the threshold accuracy (TA) $\delta < 1.25$. Let d_i and d_i^* denote the predicted and the ground-truth depths for pixel i. Abs Rel is $\frac{1}{N}\sum_{i=1}^{N}\frac{|d_i^*-d_i|}{d_i^*}$ with N the total number of pixels. TA is computed as the percentage of pixels satisfying $\max\left(\frac{d_i^*}{d_i}, \frac{d_i}{d_i^*}\right) < 1.25$. We apply shift-scale alignment by computing a single shift and scale factor over the entire video to align the predicted depths with the ground truth [53].

Results. We compare BA-Track to joint pose-depth optimization methods in Tab. 2. Using ZoeDepth [3] as input, our method consistently enhances depth estimates and achieves competitive accuracy compared to other joint refinement methods. As we show later in Sec. 4.5, our global refinement plays a crucial role here, allowing BA-Track to handle challenging scenarios with dynamic content. On MPI Sintel, we achieve an Abs Rel of 0.408 and an accuracy of 54.1%, demonstrating comparable depth quality w.r.t. leading methods trained on much larger datasets, such as MonST3R [53]. In more complex scenarios, such as the outdoor urban scenes in AirDOS Shibuya, our method maintains robust results with an Abs Rel of 0.299 and 55.1% accuracy, outperforming competitors such as RobustCVD [23]. Our scale-map-based refinement offers efficient optimization with few parameters while remaining relatively simple. Investigating more advanced deformation models, such as neural networks, could be an interesting direction for future work.

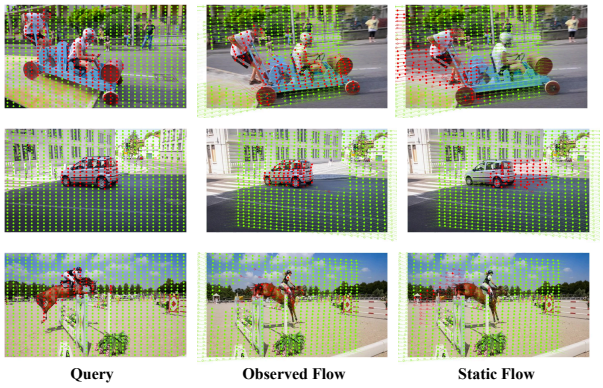

Figure 6. **Motion decoupling on the DAVIS dataset [21].** From left to right, the three columns show the reference frame for queries, total observed motion, and the static component. Green indicates static trajectories, and red indicates dynamic ones. The static component retains points on dynamic objects in their original (static) positions within the corresponding reference frame.

4.4. Motion Decoupling

In our 3D tracker, motion decoupling isolates the static (camera-induced) component from the observed point motion, allowing us to treat dynamic points as if they were stationary. This approach reduces the impact of dynamic regions on camera pose estimation and dramatically enhances the robustness of BA. It also aids the subsequent depth refinement, providing more coherent reconstructions of both static and dynamic parts. Fig. 6 illustrates the observed and static point motion on the DAVIS [21] dataset. Red points

Setting	Dynamic Handling		MPI Sintel		
	Trajectory	Camera Mask	ATE	RTE	RRE
(a)	Total	—	0.137	0.044	0.385
(b)	Total	✓	0.047	0.025	0.137
(c)	Static*	—	0.091	0.038	0.380
(d)	Static*	✓	0.072	0.032	0.233
(e)	Total-Dynamic	—	0.065	0.025	0.197
(f)	Total-Dynamic	✓	**0.034**	**0.023**	**0.115**

Table 3. **Ablation study of dynamic handling methods on Sintel [5].** Combining motion decoupling with camera masking achieves the best results. $*$ denotes a setting trained for ablation.

Method	Depth Refinement		Bonn crowd2 [31]	
	$\mathcal{L}_{\text{depth}}$	$\mathcal{L}_{\text{rigid}}$	Abs Rel ↓	$\delta < 1.25$ ↑
BA-Track	—	—	0.121	89.6
	✓	—	0.103	94.8
	—	✓	0.117	88.4
	✓	✓	**0.089**	**95.0**

Table 4. **Ablation study on depth refinement on Bonn crowd2 sequence [31].** Applying both losses has a complementary effect.

indicate dynamic trajectories, while green points represent static ones, highlighting the 3D tracker's ability to decouple the dynamic component from the observed motion.

To further demonstrate the benefits of motion decoupling, we analyze alternative motion representations for BA optimization in Tab. 3. By applying motion decoupling with the dynamic tracker $\mathcal{T}_{\text{dyn}}$, we achieve a dramatic performance boost: The ATE drops by nearly half, from 0.137 *(a)* to 0.065 *(e)*. We observe a further moderate boost in VO accuracy by setting the weights of dynamic points in BA optimization to zero *(f)*. Note that this surpasses the VO accuracy of the setting with a single tracker predicting the observed motion *(b)*. As an alternative tracker architecture, we investigate an approach with a static tracker (*c*-*d*). In this case, we train the tracker to directly regress the static component. However, this approach leads to inferior VO accuracy compared to our decoupling approach (*e*-*f*). Presumably, a single network struggles to learn two aspects simultaneously: visual tracking and motion patterns. In contrast, our dual-network design addresses these tasks independently.

4.5. Dynamic Reconstruction

We present qualitative results for camera tracking and 3D reconstruction on dynamic scenes in Fig. 2 using an alternative depth model from UniDepth [32], illustrating that our model is agnostic to the choice of depth priors. Using estimated camera poses, we back-project points into 3D via refined depth maps, yielding accurate camera motion estimation and consistent reconstructions. Our method remains robust in challenging scenarios with multiple moving people, rapid egocentric camera motion, and dynamic object interactions.

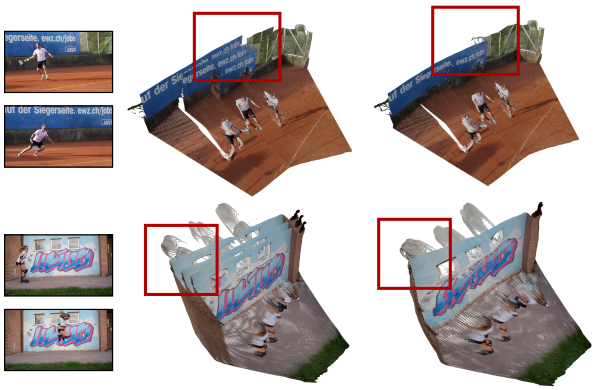

Figure 7. **Visualization of global refinement on the DAVIS [21].** We compare the 3D consistency of fused point clouds obtained from raw depth priors and our refined depths. Our refined depths yield a significantly more consistent 3D structure over time.

The superior reconstruction quality primarily stems from the proposed global refinement. To demonstrate this, we conduct an ablation study as shown in Tab. 4. We experiment with switching off $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{rigid}}$—defined in Eq. (10) and Eq. (11)—together, as well as alternately. Omitting both losses leads to suboptimal accuracy. Including $\mathcal{L}_{\text{depth}}$ alone reduces the Abs Rel error from 0.121 to 0.103 and boosts the inlier percentage from 89.6% to 95.0%. The rigid consistency alone offers modest gains, but the best result is achieved by combining both losses, illustrating their mutual complementarity. Fig. 7 shows two intuitive comparisons between the reconstructions with and without our depth refinement. Directly fusing the monocular depth map with the estimated camera pose leads to inconsistent reconstructions with duplicated 3D structures, whereas our global refinement significantly improves 3D coherence.

5. Conclusion

In this work, we addressed the challenge of reconstructing dynamic scenes from casual video sequences using a traditional SLAM method. We equipped a 3D point tracker with a motion decoupling mechanism to separate camera-induced motion from object motion. Operating on the isolated static motion components, bundle adjustment leads to substantial improvements in camera pose accuracy and reconstruction quality. Additionally, global refinement leverages the sparse depth estimates from BA to ensure scale consistency and alignment of depth maps across video frames. Overall, BA-Track enables precise 3D trajectory estimation and dense reconstructions in highly dynamic environments. More broadly, our work demonstrates that traditional optimization enhanced by deep priors can provide an accurate and robust solution to challenging real-world scenarios.

Acknowledgements. This work was supported by the ERC Advanced Grant SIMULACRON, by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV) through the AuSeSol-AI project (grant 67KI21007A), and by the TUM Georg Nemetschek Institute Artificial Intelligence for the Built World (GNI) through the AICC project.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. *Commun. ACM*, 54(10):105–112, 2011. 1

[2] Berta Bescos, José M Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *Robotics and Automation Letters*, 3(4):4076–4083, 2018. 1, 2, 3

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv:2302.12288*, 2023. 3, 7

[4] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. CodeSLAM—Learning a compact, optimisable representation for dense visual SLAM. In *CVPR*, pages 2560–2568. IEEE, 2018. 2

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625. Springer, 2012. 6, 7, 8

[6] Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. LEAP-VO: Long-term effective any point tracking for visual odometry. In *CVPR*, pages 19844–19853. IEEE, 2024. 3, 6

[7] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *ECCV*, pages 306–325. Springer, 2025. 2

[8] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE TPAMI*, 29(6):1052–1067, 2007. 2

[9] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video. *NeurIPS*, 35:13610–13626, 2022. 2, 5, i, ii

[10] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, pages 10061–10072. IEEE, 2023. 2

[11] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: large-scale direct monocular SLAM. In *ECCV*, pages 834–849. Springer, 2014. 1, 2

[12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE TPAMI*, 40(3):611–625, 2017. 2

[13] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. St4RTrack: Simultaneous 4D reconstruction and tracking in the world. In *ICCV*, 2025. 3

[14] Ziyue Feng, Liang Yang, Longlong Jing, Haiyan Wang, YingLi Tian, and Bing Li. Disentangling object motion and occlusion for unsupervised multi-frame monocular depth. In *ECCV*, pages 228–244. Springer, 2022. 3

[15] Xiang Gao, Rui Wang, Nikolaus Demmel, and Daniel Cremers. LDSO: Direct sparse odometry with loop closure. In *IROS*, pages 2198–2204. IEEE, 2018. 2, i

[16] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, pages 59–75. Springer, 2022. 2

[17] 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. In *CVPR*, pages 2005–2015, 2025. 3

[18] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *ECCV*, pages 18–35. Springer, 2024. 2, 4, i

[19] Yoni Kasten, Wuyue Lu, and Haggai Maron. Fast encoder-based 3d from casual videos via point track processing. In *NeurIPS*, pages 96150–96180, 2024. 3

[20] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, pages 9492–9502. IEEE, 2024. 3, 7

[21] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, pages 123–141. Springer, 2018. 2, 7, 8, ii, iii

[22] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *CVPR*, pages 1611–1621. IEEE, 2021. 1, 3, 5, 6, 7

[24] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense Optical Tracking: connecting the dots. In *CVPR*, pages 19187–19197. IEEE, 2024. 2

[25] Siyuan Li, Yue Luo, Ye Zhu, Xun Zhao, Yu Li, and Ying Shan. Enforcing temporal consistency in video depth estimation. In *ICCV*, pages 1145–1154. IEEE, 2021. 3

[26] 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 *CVPR*, pages 10486–10496, 2025. 3

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[28] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM TOG*, 39(4):71–1, 2020. 3

[29] Zhaoyang Lv, Nicholas Charron, Pierre Moulon, Alexander Gamino, Cheng Peng, Chris Sweeney, Edward Miller, Huixuan Tang, Jeff Meissner, Jing Dong, et al. Aria everyday activities dataset. *arXiv preprint arXiv:2402.13349*, 2024. 2

[30] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular

SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 2

[31] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguère, and Cyrill Stachniss. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *IROS*, pages 7855–7862. IEEE, 2019. 7, 8

[32] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *CVPR*, pages 10106–10116. IEEE, 2024. 3, 6, 8, ii

[33] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. ii

[34] Yuheng Qiu, Chen Wang, Wenshan Wang, Mina Henein, and Sebastian Scherer. AirDOS: Dynamic SLAM benefits from articulated objects. In *ICRA*, pages 8047–8053. IEEE, 2022. 2, 6, 7

[35] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021. 3

[36] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80:72–91, 2008. 2

[37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113. IEEE, 2016. 3, ii

[38] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518. Springer, 2016. 3

[39] Shihao Shen, Yilin Cai, Wenshan Wang, and Sebastian Scherer. DytanVO: Joint refinement of visual odometry and motion segmentation in dynamic environments. In *ICRA*, pages 4048–4055. IEEE, 2023. 2, 6

[40] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic Point Maps: A versatile representation for dynamic 3d reconstruction. In *ICCV*, 2025. 3

[41] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. In *ICLR*, 2019. 2

[42] Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. In *ICLR*, 2020.

[43] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. *NeurIPS*, 34:16558–16569, 2021. 6

[44] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *NeurIPS*, 36, 2024. 2, 6, i

[45] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment – A modern synthesis. In *ECCV*, pages 298–372. Springer, 1999. 2, 5

[46] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Laina, Diane Larlus, Dima Damen, and Andrea Vedaldi. EPIC Fields: Marrying 3D geometry and video understanding. In *NeurIPS*, 2023. 6

[47] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI*, 13(04): 376–380, 1991. 6

[48] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, pages 10510–10522, 2025. 3

[49] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. TartanVO: A generalizable learning-based VO. In *CoRL*, pages 1761–1772. PMLR, 2021. 2, 6

[50] Felix Wimbauer, Weirong Chen, Dominik Muhle, Christian Rupprecht, and Daniel Cremers. AnyCam: Learning to recover camera poses and intrinsics from casual videos. In *CVPR*, pages 16717–16727, 2025. 2

[51] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. SpatialTracker: Tracking any 2D pixels in 3D space. In *CVPR*, pages 20406–20417. IEEE, 2024. 2, 4, i

[52] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *NeurIPS*, pages 21875–21911, 2024. 3

[53] 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 *ICLR*, 2025. 3, 7

[54] 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, 6, 7, ii

[55] 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. 1, 3, 6

Back on Track: Bundle Adjustment for Dynamic Scene Reconstruction

Supplementary Material

A. Implementation Details

Model Architecture. We develop our 3D tracker based on CoTracker [18], incorporating additional modifications. The feature extractor is a CNN-based architecture, consisting of a 7×7 convolutional layer followed by several 3×3 residual blocks, which generate multi-scale feature maps. These feature maps are aggregated into a single feature map with additional convolutional layers, producing a resolution of $\frac{1}{4}$ of the input image. We adopt the depth map encoding strategy from SpatialTracker [51], omitting the tri-plane correlation for improved efficiency.

The 3D tracker $\mathcal{T}$ is a transformer-based architecture that alternates between spatial and temporal attention blocks, consisting of 6 layers. For motion decoupling, we employ a smaller 3-layer transformer $\mathcal{T}_{\text{dyn}}$ designed specifically to predict the static component. Each layer in both transformers includes a temporal and a spatial attention block, with each block comprising an attention layer followed by an MLP.

Refinement Formulation. The 3D tracker uses an iterative transformer-based refiner module $\mathcal{T}$ [18]. To provide the initial state as input to $\mathcal{T}$, we copy the 2D location of the query point across all frames $s \in (1, \dots, S)$ and sample the point features as

$$X^{(0)}(s) = X, \qquad f^{(0)}(s) = F_t[\mathbf{x}], \tag{13}$$

where the superscript $^{(\cdot)}$ denotes the iteration index. Aggregating context over all timesteps in the window, the 3D tracker $\mathcal{T}$ iteratively updates the point features $f(s)$ and the 3D trajectories $X(s)$. Dropping timestep s to avoid clutter, the update in the k-th iteration is

$$(X^{(k+1)},\ f^{(k+1)}) = \mathcal{T}(f^{(k)}, \text{PE}(X^{(k)}), C^{(k)}), \tag{14}$$

where $\text{PE}(\cdot)$ represents the positional embedding of the point track, encoding its 3D location and timestep with periodic bases.

For the dynamic tracker, $\mathcal{T}_{\text{dyn}}$ predicts the object-induced motion X_{dyn} and dynamic point feature f_{dyn} as

$$(X_{\text{dyn}}^{(k+1)}, f_{\text{dyn}}^{(k+1)}) = \mathcal{T}_{\text{dyn}}(f_{\text{dyn}}^{(k)}, \text{PE}(X_{\text{static}}^{(k)}), C^{(k)}), \tag{15}$$

where $X_{\text{dyn}}^{(0)}$ is set to 0 and $f_{\text{dyn}}^{(0)}$ is set to $F_t[\mathbf{x}]$.

Local Context. The local correlation map $C^{(k)}$ is a function of the point track $X^{(k)}$, *i.e.*

$$C^{(k)}(s) = C[X^{(k)}(s)], \tag{16}$$

and serves to enhance the spatial context of each tracked point. To compute C, we follow [51] and first apply Fourier embedding to the depth map $\mathbf{D}$ to obtain depth features $\mathbf{D}^{\text{Fourier}}$. These are concatenated with the image features $\mathbf{F}$ to produce fused features $\mathbf{F}^{\text{hyb}}$. We then apply bilinear interpolation to generate multi-scale hybrid feature maps $\tilde{\mathbf{F}}^{\text{hyb}}$. For each point $X^{(k)}(s)$, we extract a local region $\tilde{\mathbf{F}}_s^{\text{hyb}}$ centered at $X^{(k)}(s)$, and compute the correlation as the inner product between the point feature $f^{(k)}(s)$ and the surrounding hybrid features [18]. The resulting correlation map $C^{(k)}$ captures both appearance and geometric cues in a unified representation.

Training. We train our model on the TAP-Vid-Kubric training set [9], which includes 11,000 sequences. Each sequence consists of 24 frames derived from the MOVi-F dataset. Following the data preprocessing steps from CoTracker [18], we additionally extract dynamic labels, 3D total trajectory, and static trajectory ground truth. The static trajectory ground truth is generated by back-projecting queries from their 3D positions in the source frame into the target frames using ground-truth camera poses. The original image resolution of each sequence is 512×512, and we crop it to 384×512 during training. For image augmentation, we apply random resizing, flipping, cropping, Gaussian blurring, and color jitter. For depth augmentation, we adopt the scale-shift augmentation and Gaussian blurring techniques described in [15]. The augmented ground-truth depth is consistently used during the training process.

Bundle Adjustment. We build our bundle adjustment framework based on DPVO [44] and extend it to support RGB-D bundle adjustment. To enhance the robustness of pose estimation, we incorporate weight filtering during the pose update computation. We set the visibility threshold $\delta_v = 0.9$ and the dynamic label threshold $\delta_m = 0.9$ to ensure that camera pose updates rely exclusively on reliable point trajectories. This approach is adopted because recovering the camera pose primarily depends on a few accurate correspondences. In contrast, for depth updates, we consider all point trajectories to fully leverage the static components estimated through motion decoupling.

B. Additional Ablation Experiments

Comparison of single and dual network architectures. In Fig. 8, we further compare *(1)* a single-network tracker predicting only the static motion (Static*) and *(2)* our default tracker predicting the total and dynamic components. Static* struggles to capture the camera motion for dynamic points and produces motion label outliers, degrading pose estimates. This is likely because Static* predicts similar camera-induced motion for all points, making dynamic tracks hard

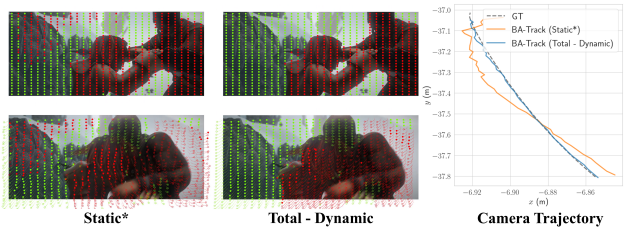

Figure 8. **Comparison of different trackers.** Red: estimated dynamic point tracks. Green: estimated static point tracks. The resulting camera trajectories from different trackers are shown on the right. Our dual networks with motion decoupling yield more accurate camera poses.

Figure 9. **Qualitative comparison across different depth priors.** Our method is robust to different types of depth priors.

to distinguish.

Robustness with different depth priors. To evaluate the robustness of our method to different depth prior models, we use two additional depth backbones: UniDepth-V2 [32] and Depth-Anything-V2 (DA-V2) [33], with per-video metric alignment based on the UniDepth-V2 scale. We compare the dynamic reconstruction results of our method using ZoeDepth, UniDepth-V2, and DA-V2. As shown in Fig. 9, our method remains robust across different depth priors for both camera pose estimation and reconstruction.

Failure modes. Our motion-decoupled tracker is robust to short-term occlusions and moderate motion but may fail under prolonged occlusions or complex, unseen motions. While BA-Track generalizes well to diverse rigid motion and performs strongly on most crowded scenes in Shibuya and DAVIS, its accuracy can degrade under severe occlusion or when tracking many small objects. Additionally, the method struggles with highly deformable or non-rigid objects, which are underrepresented in the TAP-Vid-Kubric [9] training set. Expanding training to larger and more diverse dynamic datasets could address these limitations.

C. Additional Qualitative Results

Motion Decoupling. We present additional results for motion decoupling on various video samples from the DAVIS [21] dataset, as shown in Fig. 10. The examples cover different motion scenarios, including single-object motion, multi-object motion, and occlusions. Our motion decoupling point tracker effectively distinguishes dynamic trajectories from static trajectories, providing robust estimates of the static components (illustrated in the last column). Treating the dynamic parts as static enables us to incorporate them into the geometry recovery process through bundle adjustment.

D. Limitations and Discussion

Joint Refinement with Camera Parameters. By design, BA-Track assumes the camera intrinsic parameters are provided either as ground truth or as estimates from off-the-shelf methods. These parameters are essential for bundle adjustment, which recovers the camera pose and the sparse point depths. However, calibration errors or estimation noise can introduce inaccuracies in the intrinsic parameters and lead to unreliable reprojection loss. One way to address this limitation is to extend our framework to jointly optimize camera intrinsics along with pose and point depths. Starting from an initial estimate, the intrinsics can be iteratively refined during optimization [37]. Future work could integrate this refinement with better initialization to improve convergence and reduce computational cost.

Depth Refinement. To enable track-guided depth refinement, we introduce a deformable scale map applied to the original dense depth map. This scale map facilitates refinement by aggregating information from sparse point tracks in a smooth and coherent manner, bridging the gap between sparse tracks and dense depth mapping. While effective, the scale map has limitations in handling complex error patterns present in monocular depth estimates. Future work could explore refinement models based on dense vector fields, which may improve continuity and smoothness. Additionally, representing the depth map with neural networks and refining it by backpropagating gradients to update the network weights presents another promising avenue for exploration [54].

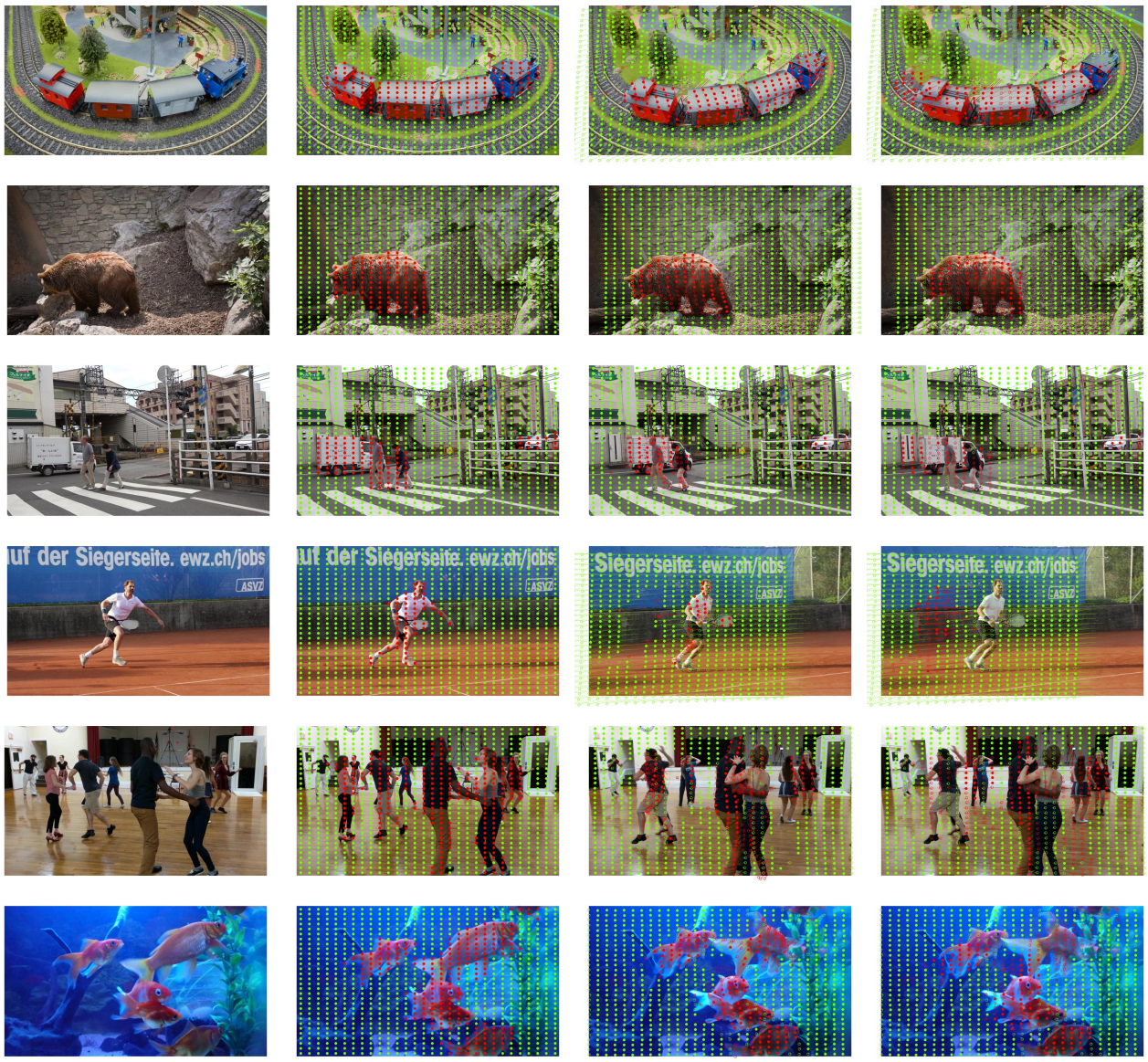

Figure 10. **Visualization of motion decoupling on the DAVIS dataset [21].** The reference frame corresponds to the first frame of the video, from which the queries are extracted. The total flow represents the combined motion of points, while the static flow refers to the motion attributed solely to its static components. The estimated static and dynamic trajectories are depicted in green and red, respectively. We observe that the red points in the static flow largely remain in their original reference frame, signifying successful motion decoupling.