

---

# TrackingWorld: World-centric Monocular 3D Tracking of Almost All Pixels

---

## A Appendix / supplemental material

### A.1 More implementation details

For hyperparameters, the stride  $s$  in Sec. 3.2 follows the same setting as in DELTA and is set to 4. The temperature  $\tau$  is set to 0.1. Each video clip contains 5 frames. The perturbation parameter  $\epsilon$  is also set to 0.1.

### A.2 Extended explanation of optimization losses

In dynamic background refinement (Stage 2), the depth consistency loss  $\mathcal{L}_{dc}$  is defined as

$$\mathcal{L}_{dc} = \sum_{i=1}^{N_{\text{static}}} \sum_{t=1}^T \|d(\mathbf{T}'_{\text{static}}(i, t), \pi_t) - \mathbf{D}_{\text{static}}(i, t)\|_2^2, \quad (1)$$

where  $d(\cdot)$  denotes the depth function, which computes the depth of the static point  $\mathbf{T}'_{\text{static}}(i, t)$  after it has been transformed into the camera coordinate system at timestep  $t$ ,  $\mathbf{D}_{\text{static}}(i, t)$  means the depth value for the  $i$ -th static point on time step  $t$ . The total loss is defined as,

$$\mathcal{L}_{\text{static}} = \lambda_{\text{ba}}\mathcal{L}_{\text{ba}} + \lambda_{\text{dc}}\mathcal{L}_{\text{dc}} + \lambda_{\text{asap}}\mathcal{L}_{\text{asap}}, \quad (2)$$

where the weights are set as follows for all datasets:  $\lambda_{\text{ba}} = 1$ ,  $\lambda_{\text{dc}} = 1$ , and  $\lambda_{\text{asap}} = 5$ .

In dynamic object tracking (Stage 3), the as-rigid-as-possible loss  $\mathcal{L}_{\text{arap}}$  [1, 2] is used to constrain geometric deformation and prevent extreme shape changes. Specifically, for each dynamic control point  $k$  in  $\mathbf{T}_{\text{dynamic}}$ , we apply KNN to find its nearest neighbors among other tracks and enforce that the relative distances between these neighboring points remain consistent over time. The loss is formulated as:

$$\mathcal{L}_{\text{arap}} = \sum_{t=1}^T \sum_k \sum_{j \in \mathcal{N}(k)} \|(\mathbf{T}_{\text{dynamic}}(k, t) - \mathbf{T}_{\text{dynamic}}(j, t)) - (\mathbf{T}_{\text{dynamic}}(k, t-1) - \mathbf{T}_{\text{dynamic}}(j, t-1))\|_2^2, \quad (3)$$

where  $\mathcal{N}(k)$  denotes the set of nearest neighbors of control point  $k$ , and  $\mathbf{T}_{\text{dynamic}}(k, t)$  is the position of point  $k$  at timestep  $t$ . In addition to the geometric constraint, we also introduce a temporal constraint—temporal smoothness loss  $\mathcal{L}_{\text{ts}}$  [2]—to penalize abrupt changes across frames and ensure temporal coherence. It is defined as:

$$\mathcal{L}_{\text{ts}} = \sum_{t=1}^T \sum_k \|\mathbf{T}_{\text{dynamic}}(k, t) - \mathbf{T}_{\text{dynamic}}(k, t-1)\|_2^2, \quad (4)$$

which enforces first-order smoothness in the trajectories of dynamic control points. The total loss is defined as:

$$\mathcal{L}_{\text{dyn}} = \lambda_{\text{ba}}\mathcal{L}_{\text{ba}} + \lambda_{\text{dc}}\mathcal{L}_{\text{dc}} + \lambda_{\text{arap}}\mathcal{L}_{\text{arap}} + \lambda_{\text{ts}}\mathcal{L}_{\text{ts}}, \quad (5)$$

where the weights are set as follows for all datasets:  $\lambda_{\text{ba}} = 1$ ,  $\lambda_{\text{dc}} = 1$ ,  $\lambda_{\text{arap}} = 100$ ,  $\lambda_{\text{ts}} = 10$ .

### A.3 Speeding-up the optimization

Directly using all tracks in camera pose optimization can be computationally prohibitive due to their large quantity. To mitigate this issue, we propose a simple strategy that reduces optimization

overhead while preserving the final trajectory density. Specifically, we apply downsampling to the static tracking points while keeping the dynamic points intact. The rationale is twofold:

(1) Optimizing static points involves jointly estimating both camera poses and static trajectories, which is computationally more expensive. In contrast, optimizing dynamic points only requires recovering dynamic trajectories, incurring a much lower cost.

(2) Dynamic motion is of primary importance for understanding the scene. Downsampling dynamic points may cause the loss of fine-grained motion details, which we aim to preserve.

Specifically, we first downsample the static tracking points on the image plane by a factor of  $\frac{1}{\varpi^2}$ . Let  $t$  denote the timestep, and let its coordinate in that frame be  $\mathbf{P}_{\text{static}}(i, t)$ . We perform downsampling by dividing both the  $x$  and  $y$  components of  $\mathbf{P}_{\text{static}}(i, t)$  by  $\varpi$  and rounding them to the nearest integers:

$$\mathbf{P}_{\text{static}}^{\text{down}}(i, t) = \text{round} \left( \frac{\mathbf{P}_{\text{static}}^x(i, t)}{\varpi}, \frac{\mathbf{P}_{\text{static}}^y(i, t)}{\varpi} \right), \quad (6)$$

where  $\text{round}(\cdot)$  denotes rounding to the nearest integer. We then concatenate the initial frame index  $t$  with the downsampled coordinate to form a spatiotemporal key  $(t, \mathbf{P}_{\text{static}}^{\text{down}}(i, t))$ . A unique operation is applied to this set of keys to eliminate duplicates and retain only unique spatiotemporal locations. As a result, we obtain a reduced yet representative set of tracking points  $\mathbf{P}_{\text{static}}^{\text{down}}$  for subsequent optimization. Next, we optimize the downsampled tracking points with the procedures described in the main paper. Upon completion of this optimization, we obtain the world-centric tracking points  $\mathbf{T}_{\text{static}}^{\text{down}}$ , and then perform an upsampling process to densify the tracking points and reconstruct the full-resolution trajectories. Specifically, for each tracking point  $i$ , we consider its position on the image plane in the initial frame, denoted as  $\mathbf{P}_{\text{static}}(i, t)$ . Using the associated depth value  $\mathbf{D}_{\text{static}}(i, t)$  and the estimated camera intrinsics, we back-project this point into the camera-centric 3D space to obtain  $\mathbf{P}_{\text{static}}^{3d}(i, t)$ . Similarly, the downsampled point  $\mathbf{P}_{\text{static}}^{\text{down}}$  can be back-projected into 3D space to yield  $\mathbf{P}_{\text{static}}^{\text{down}, 3d}$ .

To facilitate the upsampling, we search for the  $r$  nearest neighbors of  $\mathbf{P}_{\text{static}}^{3d}(i, t)$  among the downsampled 3D points  $\mathbf{P}_{\text{static}}^{\text{down}, 3d}$ . Importantly, we restrict the search to those downsampled points that are visible in the current frame. Once the nearest neighbors are identified, we perform inverse-distance weighted interpolation to reconstruct the high-resolution trajectories. Concretely, we first compute the indices and distances of the neighbors using the knn module:

$$\text{idx}, \text{dists} = \text{knn}(\mathbf{P}_{\text{static}}^{3d}(i, t), \mathbf{P}_{\text{static}}^{\text{down}, 3d}, K = r + 1), \quad (7)$$

where  $K = r + 1$  to include the query point itself. We then compute the interpolation weights as follows:

$$w_x = \frac{1}{\text{dists}_x + \epsilon}, \quad (8)$$

$$\hat{w}_x = \frac{w_x}{\sum_y w_y}, \quad (9)$$

where  $\epsilon$  is a small constant added for numerical stability. Finally, we recover the full-resolution world-centric trajectories by computing a weighted aggregation over the downsampled points:

$$\mathbf{T}_{\text{static}}^{\text{full}} = \mathcal{F}(\hat{w}_x, \mathbf{T}_{\text{static}}^{\text{down}}), \quad (10)$$

where  $\mathcal{F}$  denotes the weighted aggregation function. The visualization of the proposed speeding-up strategy is shown in Fig. 1, and its quantitative effectiveness on the Sintel dataset is reported in Table 1.

speeding-up strategy	Sintel					
	ATE ↓	RTE ↓	RRE ↓	Abs Rel ↓	$\delta < 1.25 \uparrow$	T (min)
$\times$	0.089	<b>0.034</b>	0.414	<b>0.207</b>	<b>73.8</b>	60
$\checkmark$	<b>0.088</b>	0.035	<b>0.410</b>	0.218	73.3	<b>8</b>

Table 1: **Quantitative evaluation of the speeding-up strategy on the Sintel dataset.** The proposed method achieves similar accuracy with reduced optimization time.

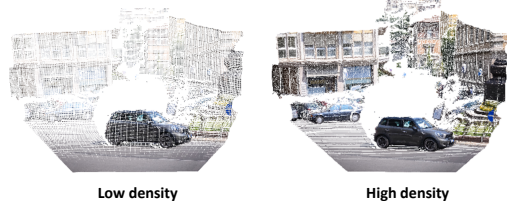


Figure 1: Effectiveness of the speeding-up strategy.

#### A.4 Consistent Video Depth Generation.

Our method is capable of producing temporally consistent video depth, as it performs dense 3D tracking for nearly all pixels. This enables the generation of consistent depth sequences through a straightforward interpolation-based propagation step.

Assume we have obtained  $M$  dense 3D tracking points across a video sequence. For each frame  $t \in \{1, 2, \dots, T\}$ , let  $\mathcal{M}_t$  denote the set of visible 3D tracking points and  $D_t$  their corresponding camera-centric depths. We also denote the raw monocular depth prediction for frame  $t$  as  $D_t^{\text{raw}}$ .

To align the raw monocular depth with our optimized tracking-based depth, we compute a per-point local scale ratio rather than a global frame-wise scale. For each tracking point  $\mathbf{M}_t^i \in \mathcal{M}_t$  with image-plane projection  $\mathbf{p}_t^i$ , the scale ratio is defined as:

$$r_t^i = \frac{D_t(\mathbf{p}_t^i)}{D_t^{\text{raw}}(\mathbf{p}_t^i)}. \quad (11)$$

For an arbitrary pixel  $\mathbf{p}_t$  in frame  $t$ , its final aligned depth  $\hat{D}_t(\mathbf{p}_t)$  is obtained by propagating the local scale information from nearby 3D tracking points. Let  $\mathcal{N}_t(\mathbf{p}_t) \subset \mathcal{M}_t$  denote its  $k$  nearest neighbors in the image plane. For each neighbor  $\mathbf{M}_t^j \in \mathcal{N}_t(\mathbf{p}_t)$ , we compute the 3D distance based on their raw-depth-lifted coordinates:

$$\mathbf{P}_t = D_t^{\text{raw}}(\mathbf{p}_t) \cdot K^{-1} \tilde{\mathbf{p}}_t, \quad \mathbf{Q}_t^j = D_t^{\text{raw}}(\mathbf{p}_t^j) \cdot K^{-1} \tilde{\mathbf{p}}_t^j, \quad (12)$$

$$d_t^j = \|\mathbf{P}_t - \mathbf{Q}_t^j\|_2, \quad (13)$$

where  $K$  is the camera intrinsic matrix and  $\tilde{\mathbf{p}}$  denotes the homogeneous coordinate of pixel  $\mathbf{p}$ .

We assign each neighbor an inverse-distance weight:

$$w_t^j = \frac{1}{d_t^j + \epsilon}, \quad \tilde{w}_t^j = \frac{w_t^j}{\sum_{j=1}^k w_t^j}, \quad (14)$$

where  $\epsilon$  is a small constant for numerical stability. The interpolated local scale ratio at pixel  $\mathbf{p}_t$  is then computed as:

$$r_{\mathbf{p}_t} = \sum_{j=1}^k \tilde{w}_t^j \cdot r_t^j, \quad (15)$$

and the final aligned depth is given by:

$$\hat{D}_t(\mathbf{p}_t) = r_{\mathbf{p}_t} \cdot D_t^{\text{raw}}(\mathbf{p}_t). \quad (16)$$

Through this weighted propagation, accurate scale information from the dense 3D tracks is effectively diffused across the entire image, producing a temporally consistent and spatially coherent video depth sequence. As shown in Tab. 2,

#### A.5 Runtime comparison with baseline

To further evaluate the efficiency and effectiveness of our optimization strategy, we compare our method with a strong baseline that combines camera poses and consistent depths from Uni-4D [?] ]

Category	Method	Sintel		Bonn		TUM D	
		Abs Rel $\downarrow$	$\delta < 1.25 \uparrow$	Abs Rel $\downarrow$	$\delta < 1.25 \uparrow$	Abs Rel $\downarrow$	$\delta < 1.25 \uparrow$
Single-frame depth	Depth Anything V2 [3]	0.348	59.2	0.106	92.1	0.211	78.0
	Depth Pro [4]	0.418	55.9	0.068	<b>97.4</b>	0.126	89.3
	ZoeDepth [5]	0.467	47.3	0.087	94.8	0.176	74.5
	Unidepth [6]	0.473	63.0	0.057	97.4	0.113	91.9
Video depth	ChronoDepth [7]	0.687	48.6	0.100	91.1	/	/
	DepthCrafter [8]	0.292	69.7	0.075	97.1	/	/
Joint video depth & pose	DUST3R [9]	0.422	54.2	0.144	84.5	0.239	71.1
	MonST3R [10]	0.335	58.6	0.063	96.4	0.301	55.8
	Align3R (Depth Pro) [11]	0.263	64.1	0.058	97.1	0.111	88.9
	Ours (DELTA [12])	<b>0.222</b>	<b>72.6</b>	0.058	97.3	<b>0.086</b>	<b>92.3</b>
	Ours (CoTrackerV3 [13])	0.232	71.4	<b>0.054</b>	97.3	0.090	91.7

Table 2: **Video depth estimation results.** We evaluate our model on three datasets: Sintel, Bonn and TUM D. **Best** results are highlighted.

with dense camera-centric 3D tracking from DELTA. Since both methods share the same 3D tracking backbone, the comparison focuses on optimization runtime and accuracy.

Uni4D estimates camera poses in a streaming manner, where the pose between consecutive frames is computed step by step. This results in a total runtime that scales linearly with the video length. In contrast, our approach performs pose estimation in a clip-to-global parallel manner, significantly improving computational efficiency.

In terms of 3D tracking accuracy, our method achieves higher precision by effectively distinguishing static and dynamic regions using dynamic masks, and by jointly optimizing both camera poses and 3D trajectories. This joint optimization leads to improved geometric consistency and reconstruction quality.

We report detailed quantitative results on camera pose estimation (Sintel, frames 30–50) and world-coordinate 3D tracking (ADT, first 64 frames) in Tab. 3. All experiments are conducted on the same hardware configuration for a fair comparison. The results demonstrate that our method achieves both higher accuracy and lower runtime compared to the Uni4D baseline. The improvement mainly stems from our clip-to-global parallel optimization strategy and the integration of dynamic mask filtering, which together enhance efficiency and reconstruction quality.

Setting	Sintel				ADT	
	ATE $\downarrow$	RTE $\downarrow$	RPE $\downarrow$	Avg. Time (min) $\downarrow$	APD <sub>3D</sub> $\uparrow$	Avg. Time (min) $\downarrow$
Uni4D + DELTA	0.118	0.048	0.610	19	68.95	28
<b>Ours (DELTA)</b>	<b>0.087</b>	<b>0.036</b>	<b>0.406</b>	<b>15</b>	<b>75.18</b>	<b>20</b>

Table 3: Runtime and accuracy comparison on Sintel (30–50 frames) and ADT (first 64 frames).

## A.6 More ablation study

**Ablation on the filtering mechanism.** In this section, we introduce the details of the filtering process. We first compute the complement set by discarding pixels that lie in any previously visible 2D track trajectory. However, this operation may introduce isolated pixels, which are typically not meaningful for downstream scene understanding. To mitigate this, we construct connected components from the newly selected 2D tracking points and apply a size threshold  $\tau$  to remove small components. Only connected regions with more than  $\tau$  pixels are retained. This ensures that the preserved 2D tracks are geometrically meaningful and more likely to correspond to coherent object parts. We found that this filtering procedure consistently improves accuracy, as most filtered points are either outliers or redundantly close to already tracked regions. Moreover, the threshold  $\tau$  is robust across different scenes: we use a fixed value of  $\tau = 50$  in all experiments without additional tuning.



Setting	Sintel			Bonn		
	ATE ↓	RTE ↓	RPE ↓	ATE ↓	RTE ↓	RPE ↓
w/o Filtering	0.105	0.038	0.442	0.018	0.007	0.601
w/ Filtering	<b>0.088</b>	<b>0.035</b>	<b>0.410</b>	<b>0.016</b>	<b>0.005</b>	<b>0.564</b>

Table 4: Ablation study on the filtering mechanism on Sintel and Bonn datasets. Filtering improves both camera pose estimation and tracking stability.

## A.7 More visualization

### A.7.1 Comparison on camera pose estimation

Fig. 2 illustrates qualitative comparisons of camera pose estimation on the Sintel [14], Bonn [15], and TUM-D [16] datasets. We evaluate our method against two joint depth and pose estimation baselines: DUST3R [9] and MonST3R [10]. As shown, our method produces camera trajectories that are more stable and better aligned with the ground truth, reflecting enhanced robustness and accuracy.

### A.7.2 World-centric dense tracking results

We present additional visualizations of the world-centric tracking results in Fig. 3. To enhance clarity, we only visualize the point clouds on temporally spaced keyframes. However, the displayed trajectories are computed by connecting 3D tracks across all frames, capturing the complete motion over time. For more vivid and dynamic results, please refer to the accompanying video in the supplementary material.

## A.8 Limitation and future work

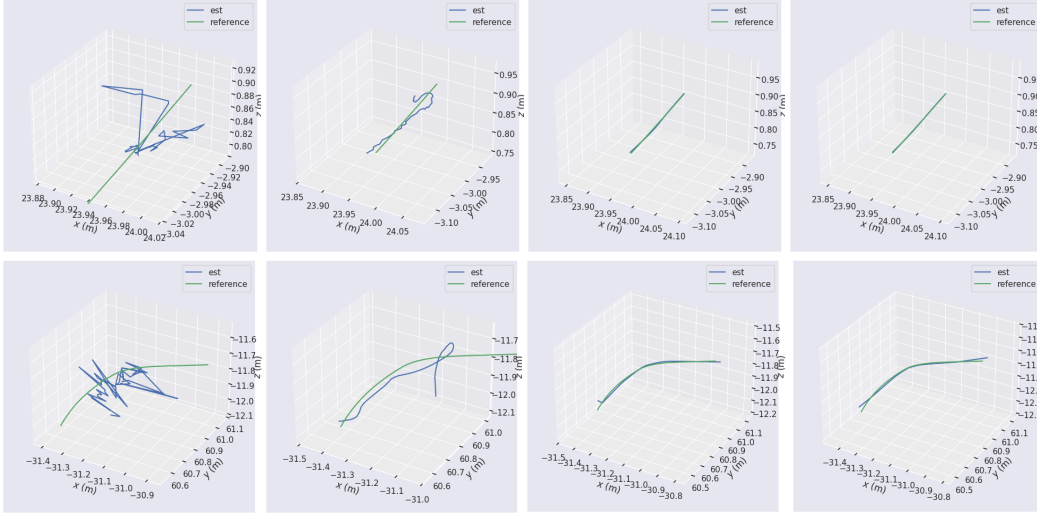
Our method currently relies on several auxiliary models to obtain 2D tracks, monocular depths, and dynamic masks. This dependence introduces additional computational overhead and imposes stringent quality requirements on these components. In the future, feed-forward solutions may offer a more suitable and efficient direction. For instance, St4RTrack [17] adopts a feed-forward design, but its pair-wise matching strategy inherently suffers from drift accumulation, which requires global optimization for correction. Inspired by VGGT [18], a promising direction may involve jointly processing all frames to directly predict the state of each frame across time. This could potentially enable a more consistent and globally coherent trajectory estimation.

## A.9 Assets availability

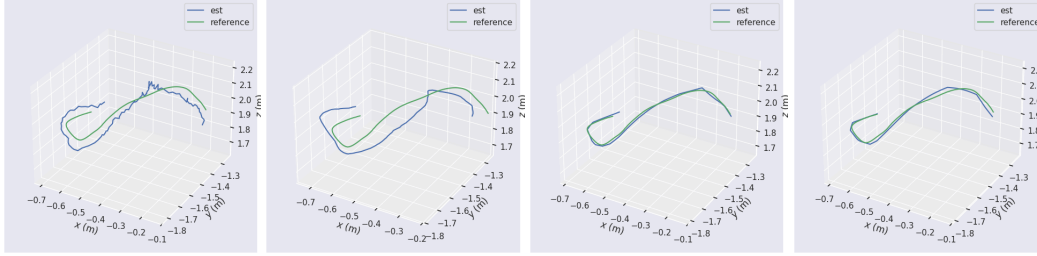
The datasets used in this study and their respective licenses are listed below:

- **Sintel** [14]: Available at <http://sintel.is.tue.mpg.de/>. This dataset is intended for optical flow evaluation. Please refer to the official website for specific license information.
- **Bonn RGB-D Dynamic Dataset** [15]: Available at <https://www.ipb.uni-bonn.de/data/rgbd-dynamic-dataset/>, licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
- **TUM RGB-D Dataset** [16]: Available at <https://cvg.cit.tum.de/data/datasets/rgbd-dataset/>, licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).
- **DAVIS** [19]: Available at <https://davischallenge.org/>. According to the DAVIS 2017 Challenge, the dataset is licensed under the Creative Commons Attribution 4.0 License.
- **Aria Digital Twin (ADT)** [20]: Available at <https://www.projectaria.com/datasets/adt/>. Provided by Meta Reality Labs Research; please consult the official website for license details.
- **Panoptic Studio Dataset** [21]: Available at <https://www.cs.cmu.edu/~hanbyulj/panoptic-studio/>. Provided by Carnegie Mellon University; please refer to the project website for license information.

## Sintel



## Bonn



## TUM-D

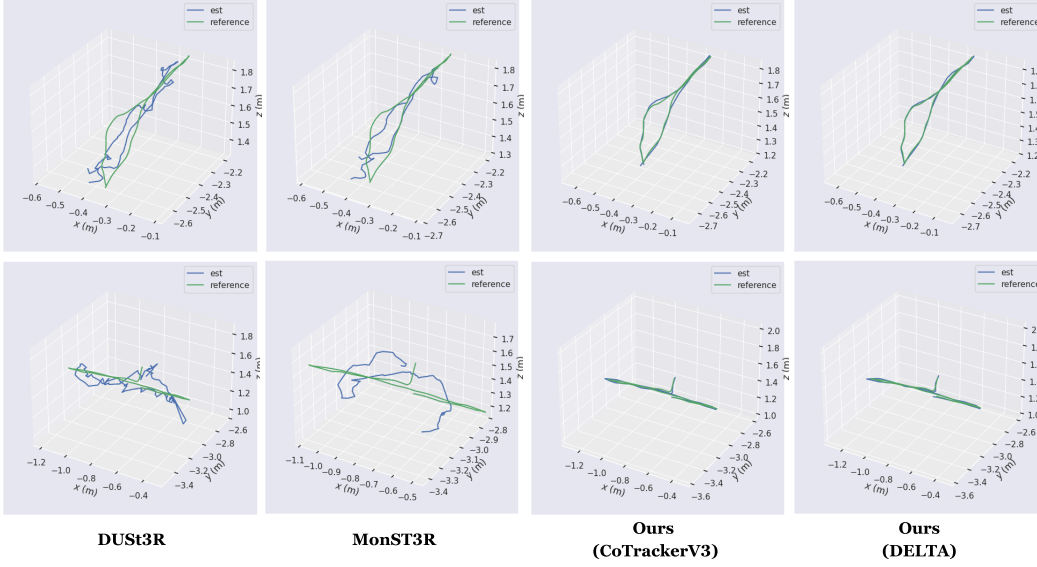


Figure 2: **Camera pose estimation comparison** on the Sintel [14] Bonn [15] and TUM-D [16] datasets.

- **CVO Dataset** [22]: Available at <https://github.com/mulns/AccFlow/blob/main/data/README.md>, licensed under the MIT License.

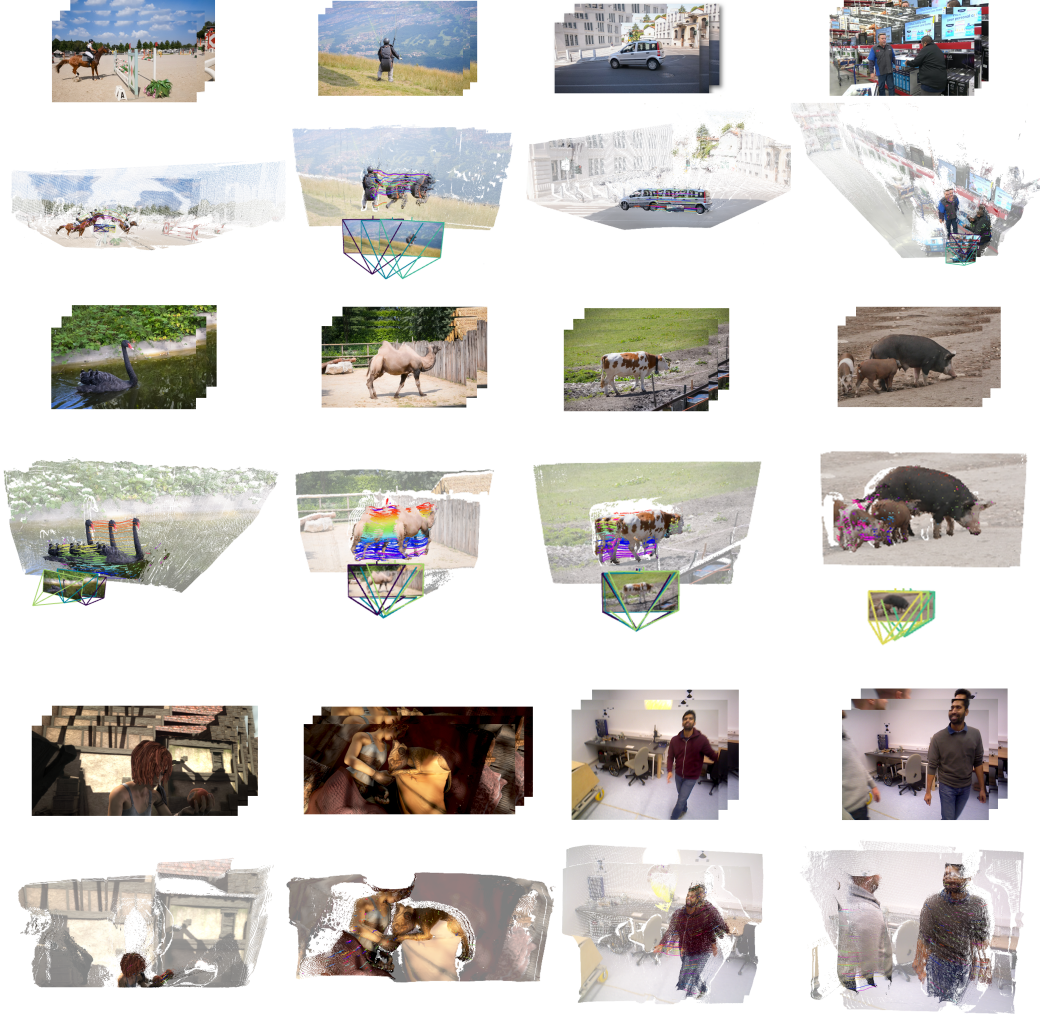


Figure 3: **More Qualitative results.** Our method can output 3D tracks in a world-centric coordinate system.

## References

- [1] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.
- [2] David Yifan Yao, Albert J Zhai, and Shenlong Wang. Uni4d: Unifying visual foundation models for 4d modeling from a single video. *arXiv preprint arXiv:2503.21761*, 2025.
- [3] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.
- [4] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.
- [5] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.
- [6] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024.

- [7] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. *arXiv preprint arXiv:2406.01493*, 2024.
- [8] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024.
- [9] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.
- [10] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.
- [11] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. *arXiv preprint arXiv:2412.03079*, 2024.
- [12] Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. *arXiv preprint arXiv:2410.24211*, 2024.
- [13] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.
- [14] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012.
- [15] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7855–7862. IEEE, 2019.
- [16] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.
- [17] 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. *arXiv preprint arXiv:2504.13152*, 2025.
- [18] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025.
- [19] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.
- [20] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20133–20143, 2023.
- [21] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015.
- [22] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. Accflow: Backward accumulation for long-range optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12119–12128, 2023.