

Align3R: Aligned Monocular Depth Estimation for Dynamic Videos

Supplementary Material

1. More implementation details

To estimate monocular depth, we employ Depth Anything V2 [13] and Depth Pro [1]. For Depth Anything V2, we use the large model variant to predict depth maps. During the global alignment of our method, we perform 300 iterations with the Adam optimizer, setting an initial learning rate of 0.05 and using a cosine learning rate schedule.

2. More qualitative results

2.1. Depth comparison

To provide a more vivid illustration, we perform visual comparisons on the PointOdyssey [15] validation set and the FlyingThings3D [6] test set, both containing numerous moving objects. In Fig. 2 and Fig. 3, we compare the Depth Pro version Align3R with two video depth estimation methods, ChronoDepth [10] and DepthCrafter [3]. It is worth noting that we visualize the depth after sequence alignment, with invalid areas replaced by white. These comparisons demonstrate that, after alignment, our approach achieves enhanced temporal consistency and finer detail by integrating the monocular depth estimator Depth Pro with DUST3R [12]. Additionally, in Fig. 2, the reason why some foreground objects predicted by DepthCrafter are shown in red is primarily due to certain regions in DepthCrafter having depth values less than 0 after sequence alignment. This indicates that the relative depth relationships between objects generated by DepthCrafter are not entirely accurate.

2.2. Camera pose comparison

In Fig. 4, we present qualitative results for camera pose estimation on the Sintel [2], Bonn [7], and TUM dynamics [11] datasets. We compare our model with the pose-only method COLMAP [9] and two joint depth and pose estimation methods, DUST3R [12] and MonST3R [14]. These comparisons show that our approach achieves improved camera pose estimation, demonstrating better consistency and closer alignment with the ground truth trajectory.

2.3. Dynamic point clouds

To further demonstrate the effectiveness of our method in depth and camera pose estimation, we present additional visualizations of the reconstructed point clouds. As illustrated in Fig. 5, the reconstructed point clouds exhibit strong geometric accuracy and temporal consistency, maintaining a clear structure for dynamic objects. This consistency across frames highlights our model’s ability to handle complex, real-world movements while preserving coherent geometry.

Optimization	Depth estimation	
	Abs Rel ↓	$\delta < 1.25$ ↑
Depth maps	0.306	0.613
Scale maps	0.419	0.604

Table 1. Analysis of the scale map optimization on the Sintel dataset.

Such results underline the robustness of our approach, effectively capturing and maintaining precise depth and pose information for improved 3D scene understanding in dynamic environments.

3. More ablation study

Ablation on point maps with depth maps. We have conducted an ablation study in Table 2, which demonstrates that point maps generally outperform depth maps.

Directly aligning the depth. An alternative to get consistent depth maps is to align the monocular depth map with scale factors. In Tab. 1, we align the monocular depth map I_v predicted by Depth Pro using a scale map. Instead of learning a depth map for each frame, we learn $\mathbf{S} := \{\mathbf{S}_v \in \mathbb{R}^{H \times W} | v = 1, \dots, N\}$ a set of scale maps to minimize the DUST3R target,

$$\arg \min_{\mathbf{S}, \pi, \sigma} \sum_{e \in \mathcal{E}} \sum_{v \in e} \mathbf{C}_v^e \left\| \mathbf{S}_v \hat{\mathbf{D}}_v - \sigma_e P_e(\pi_v, \mathbf{X}_v^e) \right\|_2^2. \quad (1)$$

The only difference here is that we do not learn a set of depth maps \mathbf{D} but we learn the scale map \mathbf{S}_v and compute the depth map as the product $\mathbf{D}_v = \mathbf{S}_v \hat{\mathbf{D}}_v$ where $\hat{\mathbf{D}}_v$ is the predicted Depth Pro depth map on the v -th view. This optimization process corresponds to traditional video depth optimization methods [4, 5]. However, since the initialized monocular depth maps predicted by Depth Pro are inconsistent across different frames, As shown in Fig. 1, solely optimizing the scale maps leads to inferior performances.

Flow loss of MonST3R [14]. We adopt the flow loss in MonST3R [14] because we find that flow loss does not affect the depth estimation too much but plays a crucial role in achieving accurate camera pose estimation. As shown in Table 3, we conduct an experiment on the three datasets (Sintel, TUM dynamics and Bonn) to analyze the effects of flow loss. Since camera poses can only be evaluated in 14 scenes of Sintel (as discussed in Section 4.3 of the main text), we also report the depth results of these same 14 scenes. For TUM dynamics and Bonn, we only perform the evaluation with the same 30 frames per scene for both camera pose and depth estimation. From the comparison,

Setting	Sintel (Depth)		TUM dynamics (Depth)		Sintel (Pose)			TUM dynamics (Pose)		
	Abs Rel↓	$\delta < 1.25\uparrow$	Abs Rel↓	$\delta < 1.25\uparrow$	ATE↓	RPE Trans↓	RPE Rot↓	ATE↓	RPE Trans↓	RPE Rot↓
Depth map	0.278	0.632	0.121	0.862	0.216	0.111	0.381	0.015	0.012	0.341
Point map	0.263	0.641	0.112	0.884	0.128	0.042	0.432	0.012	0.010	0.327

Table 2. Ablation on point maps with depth maps on the Sintel and TUM dynamics dataset.

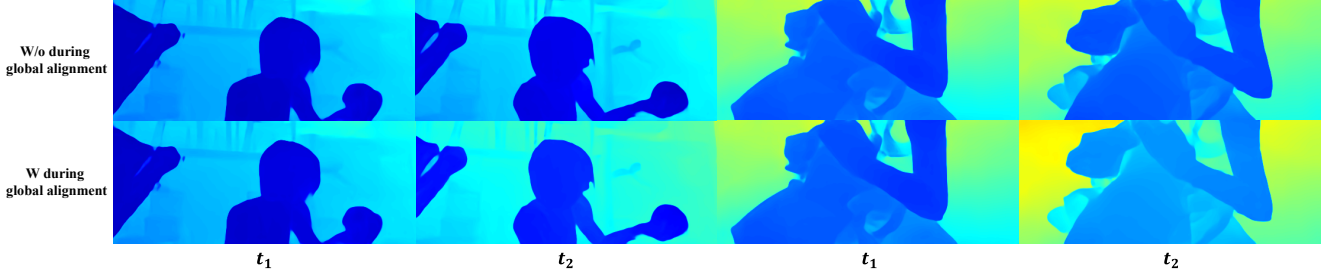


Figure 1. Visualization results with and without incorporating monocular depth estimation during global alignment.

we observe minimal differences in depth metrics but significant improvements in pose estimation. Meanwhile, we find that directly applying the flow loss to the original DUST3R greatly improve the pose estimation. The main reason is that the camera poses can be determined by several robust correspondences while being insensitive to the most depth values.

Sintel	Abs Rel↓	$\delta < 1.25\uparrow$	ATE↓	RPE Trans↓	RPE Rot↓
DUST3R w/o flow	0.515	0.533	0.601	0.214	11.426
DUST3R w flow	0.512	0.549	0.327	0.111	1.014
MonST3R	0.353	0.570	0.111	0.044	0.780
Ours w/o flow	0.314	0.562	0.204	0.164	2.305
Ours w flow	0.317	0.577	0.128	0.042	0.432

TUM-dynamics	Abs Rel↓	$\delta < 1.25\uparrow$	ATE↓	RPE Trans↓	RPE Rot↓
DUST3R w/o flow	0.172	0.766	0.093	0.035	1.708
DUST3R w flow	0.177	0.768	0.017	0.014	0.508
MonST3R	0.124	0.846	0.020	0.014	0.478
Ours w/o flow	0.099	0.879	0.043	0.025	0.630
Ours w flow	0.094	0.897	0.012	0.010	0.327

Bonn 5 scene	Abs Rel↓	$\delta < 1.25\uparrow$	ATE↓	RPE Trans↓	RPE Rot↓
DUST3R w/o flow	0.121	0.846	2.166	0.650	1.169
DUST3R w flow	0.113	0.874	0.754	0.700	0.598
MonST3R	0.060	0.971	0.686	0.595	0.593
Ours w/o flow	0.053	0.977	1.250	0.630	0.607
Ours w flow	0.052	0.977	0.673	0.570	0.576

Table 3. Analysis of the flow loss [14] for depth and pose estimation.

Runtime analysis. In Tab. 4, we provide an additional comparison of inference time using the same dataset setting as Tab.5 in the main text. Since the number of image pairs is a primary factor influencing inference time, we count the image pairs for each method to better understand the reasons behind these differences. In DUST3r, with a window

Method	#Pair	Avg. time (min)↓
DUST3R [12]	600	2.9
MonST3R [14]	250	2.6
Ours	138	1.8

Table 4. Comparison on inference time.

size of 10, for any given image i , the image pairs are:

$$\{(i, (i+1)\%30), ((i+1)\%30, i), \dots, (i, (i+10)\%30), ((i+10)\%30, i)\}, \quad (2)$$

resulting in a total of $10 \times 30 \times 2 = 600$ pairs. In MonST3r, the stride is set to 2 with a window size of 5, and explicit loop closure is not considered. So for any image i , the pairs are:

$$\{(i, i+1), (i+1, i), (i, i+1+2), (i+1+2, i), \dots, (i, i+2k+1), (i+2k+1, i)\}, \quad (3)$$

where $k = \min(5, \frac{30-1-i}{2})$. This configuration yields a total of 250 image pairs. In our method, we divide the 30 images into 3 groups, without explicit loop closure and symmetrical pairs. So the total image pairs are 3 (keyframe pairs) $+ \frac{10 \times 9}{2} \times 3 = 138$ (each group pairs) pairs. Thus, due to the significant difference in the number of image pairs, our method achieves the fastest inference speed, regardless of whether flow and trajectory smoothness losses are applied.

4. Relationship with MonST3R

Align3R is a concurrent work with MonST3R [14]. We started our project in June 2024 and the project is initially intended to improve the temporal consistency of monocular

depth estimation. Our initial idea is to adopt DUST3R [12] to align estimated depth maps of different frames. Thus, our codes are mainly based on DUST3R and we incorporate the estimated depth maps in fine-tuning DUST3R.

MonST3R [14] is released on arXiv in October 2024, which aims to extend the DUST3R model on dynamic videos without utilizing monocular depth estimation. Thus, our motivation is different from MonST3R but leads to a similar solution in the end. We find that the flow loss proposed in MonST3R is very important for pose estimation and thus we utilize the flow loss of MonST3R in our implementation. We sincerely thank the authors of DUST3R and MonST3R for sharing their codes of these two great works.

References

- [1] 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. 1, 5
- [2] 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. 1, 6
- [3] 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. 1, 4
- [4] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 1
- [5] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 1
- [6] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 1, 5
- [7] 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. 1, 6, 7
- [8] 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. 7
- [9] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1
- [10] 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. 1, 4, 5
- [11] 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. 1, 6
- [12] 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. 1, 2, 3
- [13] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiao-gang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 1
- [14] 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. 1, 2, 3
- [15] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 1, 4

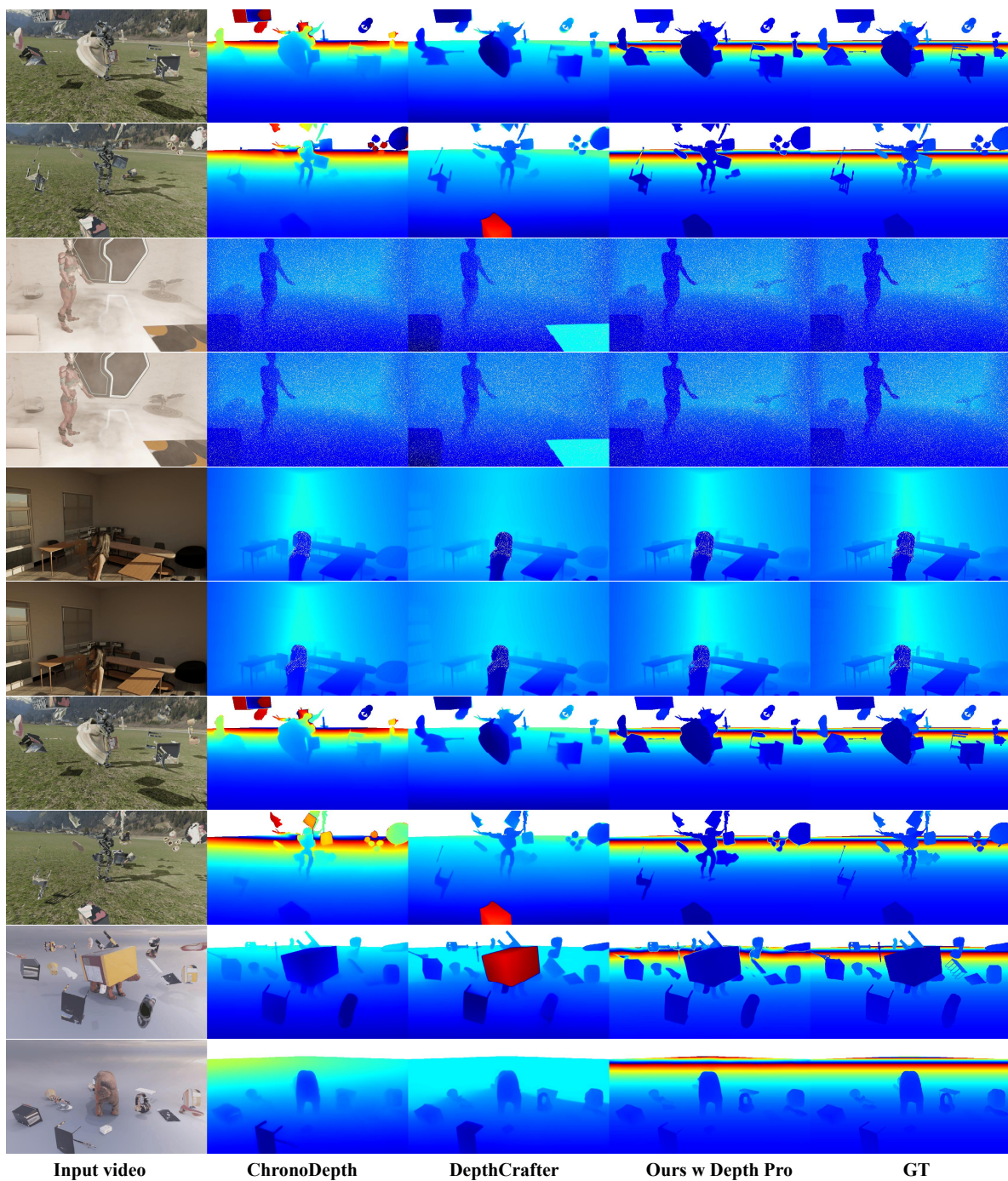


Figure 2. Qualitative comparison on the PointOdyssey [15] validation set with ChronoDepth [10] and DepthCrafter [3].

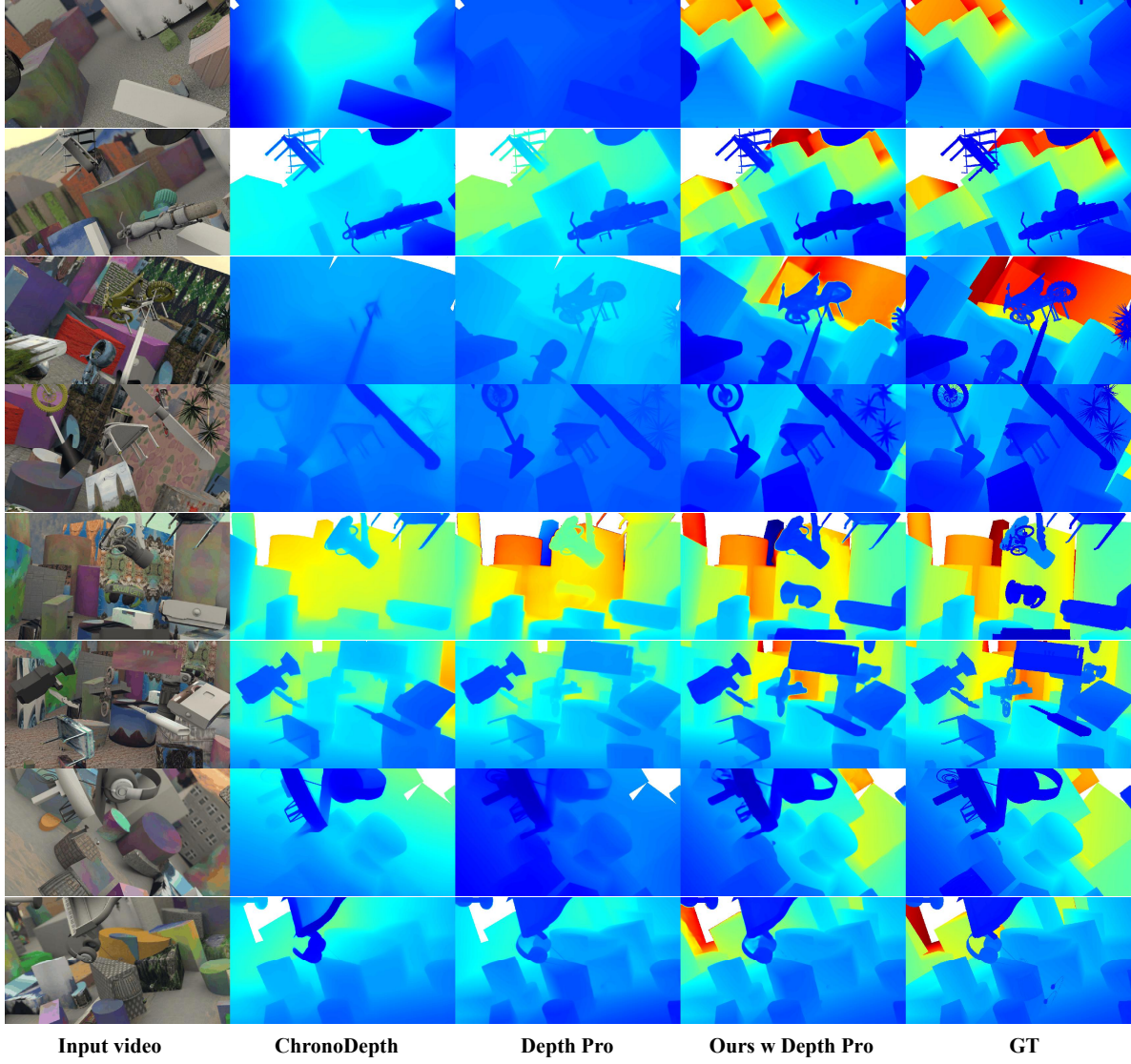


Figure 3. Qualitative comparison on the FlyingThings3D [6] test set with ChronoDepth [10] and Depth Pro [1].

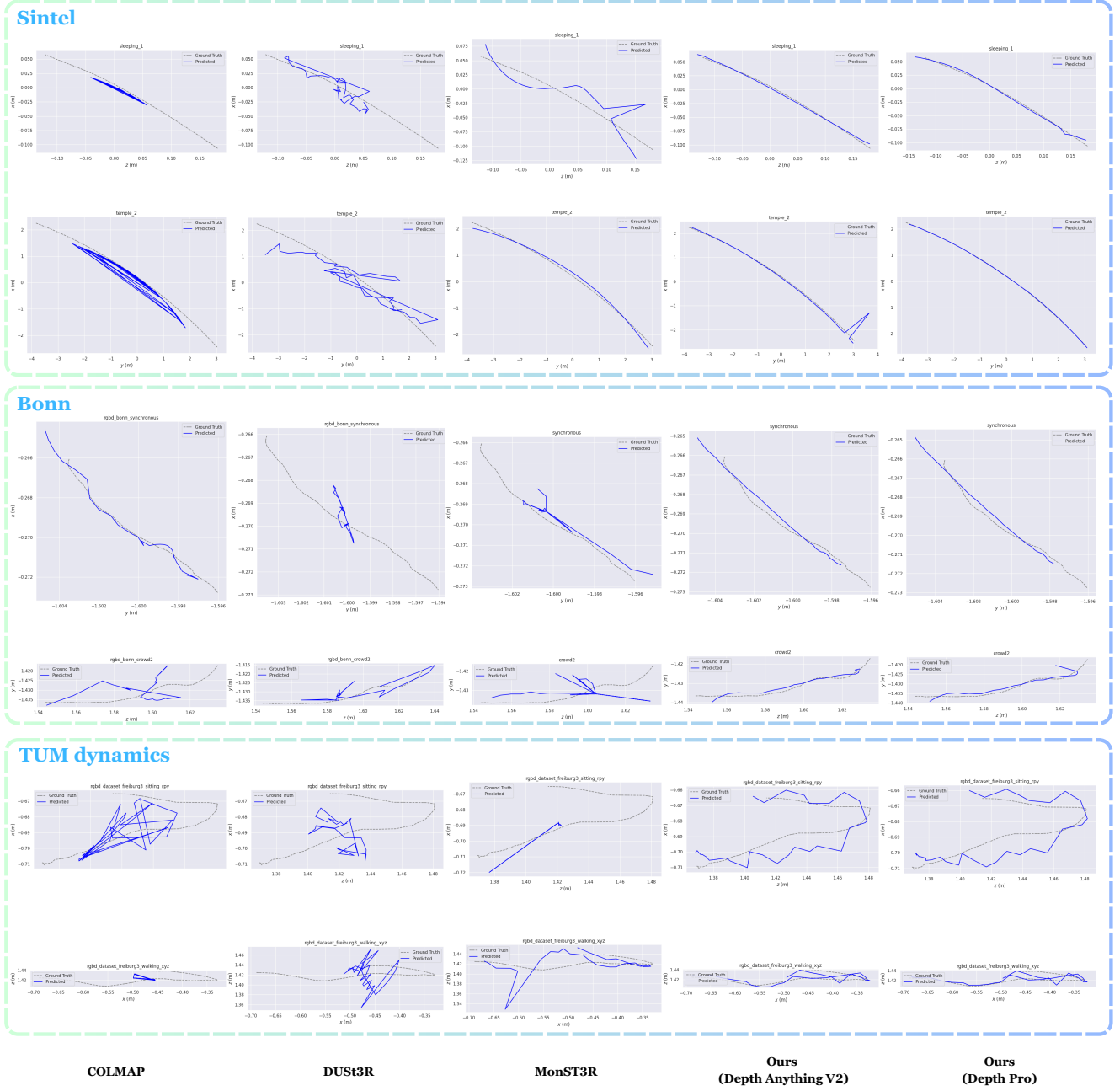


Figure 4. Camera pose estimation comparison on the TUM dynamics [11], Bonn [7], and Sintel [2] datasets.

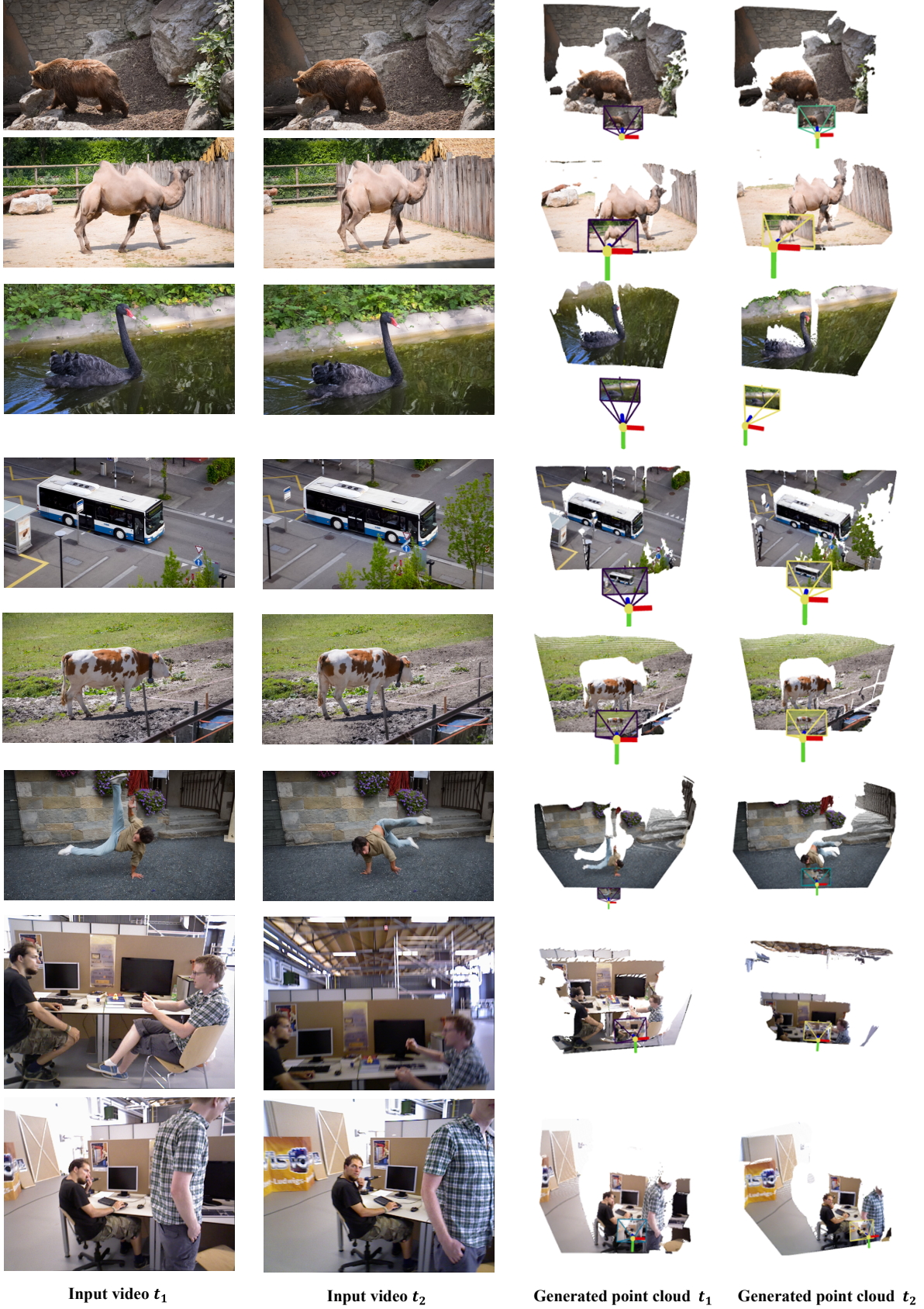


Figure 5. Visualization of point clouds on the DAVIS [8] and TUM dynamics [7] datasets.