Supplementary Material: Video Depth without Video Models

Bingxin Ke[1] Dominik Narnhofer[1] Shengyu Huang[1] Lei Ke[2] Torben Peters[1]
Katerina Fragkiadaki[2] Anton Obukhov[1] Konrad Schindler[1]
[1]ETH Zurich [2]Carnegie Mellon University

This supplementary material includes additional implementation details and experimental results.

A. Implementation Details

A.1. Depth Co-Alignment

As discussed in Sec. 3.3 in the main paper, let $k(i,j)$ denote an indexing function that returns the snippet index k corresponding to the j-th depthmap of i-th frame. To make the optimization more robust, we include an additional loss term in depth space while predicting inverse depth. We further scale the loss terms by their respective mean absolute value per frame to increase the numerical stability. Additionally, soft constraints on s_k, t_k are applied:

$$\min_{s_k>0,t_k} \left(\sum_{i=1}^{N_F} \sum_{j=1}^{N^i} \left| \frac{\widehat{\mathbf{d}_j^i} - \overline{\mathbf{d}^i}}{\widehat{\mu}_i} \right| + \left| \frac{\widehat{\mathbf{d}_j^i}^{-1} - \widetilde{\mathbf{d}}^i}{\widetilde{\mu}_i} \right| \right) + \lambda_1 \max(0, 1 - s_{k(i,j)})^2 + \lambda_2 t_{k(i,j)}, \quad (1)$$

where $\widehat{\mathbf{d}_j^i} = s_{k(i,j)} \mathbf{d}_j^i + t_{k(i,j)}$. The mean depth and mean inverse depth are defined as

$$\overline{\mathbf{d}^i} = \frac{1}{N^i} \sum_{j=1}^{N^i} \widehat{\mathbf{d}_j^i} \qquad \widetilde{\mathbf{d}}^i = \frac{1}{N^i} \sum_{j=1}^{N^i} \widehat{\mathbf{d}_j^i}^{-1}, \quad (2)$$

with the corresponding mean absolute values per frame given by

$$\widehat{\mu}_i = \frac{1}{HW} \sum^{HW} \left| \overline{\mathbf{d}^i} \right| \qquad \widetilde{\mu}_i = \frac{1}{HW} \sum^{HW} \left| \widetilde{\mathbf{d}}^i \right|. \quad (3)$$

We found that $\lambda_1 = 10^{-1}$, $\lambda_2 = 10^1$ work well in practice.

A.2. Additional Training and Inference Details

During training, we follow Marigold to use MSE loss on the latents. We apply gradient accumulation to increase the effective batch size, to 32. To better mix the samples with varying snippet lengths, every mini-batch is sampled randomly and can have different snippet lengths. For the initial depth prediction, we apply the same random Gaussian noise to all frames. When applying refinement, the same noise is used to perturb the (encoded) co-aligned depth sequence. The denoising process then starts from timestep $T/2$.

A.3. Evaluation Datasets

PointOdyssey [11] contains several sequences that feature overly simplified toy scenes, as well as some with smoke, for which depth estimation is ambiguous (*cf*. Fig. S1). We exclude these sequences from the test dataset, a detailed list of selected frames will be provided with the code. For evaluation, pixels on windows are excluded due to inconsistent depth labels.

In **ScanNet [1]**, the RGB images and depth labels include a thin black border. Following DepthCrafter [4], we crop the RGB images by removing 8 pixels from the top and bottom and 12 pixels from the left and right. Similarly, we crop the depth maps by removing 4 pixels from the top and bottom and 6 pixels from the left and right.

For **DyDToF [8]**, we exclude depth values beyond 23m, corresponding to less than 1% of the depth values.

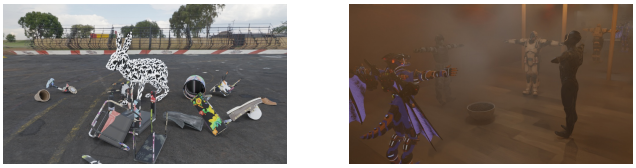

Figure S1. Examples of PointOdyssey toy scenes (*left*) and scenes with smoke (*right*).

A.4. Baseline Methods

We evaluate baseline methods using their recommended default settings. For DepthCrafter [4], the inference is performed with 25 diffusion steps, using an overlap of 25 frames for videos longer than 110 frames. For ChronoDepth [7], inference comprises 10 denoising steps, with a window size of 10 (referred to as "num-frames" in the code) and a stride of 9 (referred to as "denoise-steps" in the code).

For Marigold [5], we retrained an inverse depth version using the trailing scheduler setting [2, 6]. Under this configuration, 1-step inference with a single model achieves performance comparable to the original configuration with multi-step inference and ensembling, so we utilize the former, more efficient setting.

B. Additional Experiment Results

B.1. Temporal smoothness evaluation

We further quantitatively evaluate the temporal smoothness using optical-flow-based warping loss (OPW) [9] on PointOdyssey and ScanNet datasets and report the results in Tab. S1. The optical flow is estimated using GMFlow [10].

Table S1. Temporal smoothness (OPW↓) comparison. All values are $\times 10^3$, lower is better. * denotes catastrophic failures on some sequences. Numbers in brackets are evaluated on subsets that exclude those cases.

	PointOdyssey	ScanNet
Marigold	3.52 (4.00)	0.48
DepthAnything	3.92 (4.21)	0.32
NVDS	<u>3.50</u> (2.97)	0.29
ChronoDepth	8.98* (2.99)	0.29
DepthCrafter	7.75* (**1.30**)	<u>0.25</u>
RollingDepth (ours)	**1.42** (<u>1.63</u>)	**0.20**

We notice that ChronoDepth and DepthCrafter have catastrophic failure in some cases of PointOdyssey (*cf*. Sec. B.4), leading to large errors, as denoted by *. We manually exclude these failure cases. The re-calculated average OPW is reported in the brackets. Overall, RollingDepth shows good smoothness, on par with DepthCrafter, while being more robust than DepthCrafter and ChronoDepth against occasional failures.

We point out that OPW only evaluates the "smoothness" between adjacent frames while ignoring the long-term smoothness and geometric consistency. As shown in Tab. S2, with larger dilation rates, the geometric accuracy shows a clear improving trend, while the trend of OPW is unclear. We hypothesize that with a larger dilation rate, geometric accuracy is improved at a cost of minor local smoothness decrease when merging the aligned snippets.

Table S2. Extended table of dilation rate ablation study (Tab. 2). Values are $\times 10^3$.

	PointOdyssey			ScanNet		
Dilation rates	Abs Rel↓	$\delta 1$ ↑	**OPW** ↓	Abs Rel↓	$\delta 1$ ↑	**OPW** ↓
{1}	16.7	75.5	1.22	12.8	83.2	0.24
{1, 25}	10.2	89.5	2.06	10.6	88.8	0.29
{1, 10, 25}	10.2	89.6	1.98	9.9	90.1	0.29

B.2. Evaluation on DDAD dataset

We further evaluate the model performance on the DDAD [3] dataset, which is a driving-scene dataset with sparse depth annotation. We use the 100-frame sequences on the test set.

As shown in Tab. S3, RollingDepth outperforms other methods in terms of accuracy and smoothness.

Table S3. Evaluation on DDAD dataset.

	Abs Rel↓ $\times 10^{-2}$	$\delta 1$ ↑ $\times 10^{-2}$	OPW ↓ $\times 10^{-3}$
NVDS	30.8	57.2	0.39
ChronoDepth	34.2	46.9	<u>0.21</u>
DepthCrafter	<u>19.3</u>	<u>74.8</u>	0.28
RollingDepth (ours)	**12.8**	**83.2**	**0.19**

B.3. Inference efficiency

We report the inference efficiency comparison in Tab S4. The benchmarking is done on the same machine with a single RTX3090 GPU. For each method, we run 10 repeated inferences after a warm-up iteration, with the model loaded on GPU, and calculated the mean run time and peak memory footage of each iteration.

Table S4. Inference speed and peak GPU memory usage comparison on a 768×432 video of 250 frames. By increasing the batch size of processing, RollingDepth† can trade memory for speed.

	Time (s)	Peak GPU Memory (GB)
NVDS	284	<u>7.6</u>
ChronoDepth	121	15.0
DepthCrafter	284	13.6
RollingDepth (ours)	<u>105</u>	**6.2**
RollingDepth† (ours)	**81**	40.1

B.4. Failure cases of video models on PointOdyssey

We provide further examples from the PointOdyssey dataset where video-based methods struggle. Figure S2 features scenes with large depth changes, such as hand gestures in front of the camera or objects entering the near field. These sudden changes require rapid alterations of the depth range, both before and after the event. Video models tend to produce incorrect overall scene layout in such cases, we hypothesize that they "try too hard" to equalize the depth range throughout the scene.

B.5. Failure Cases of RollingDepth

While our proposed method handles changing depth range more robustly than video models, it also has certain limitations. Two examples are shown in Fig. S3. RollingDepth sometimes misjudges the depth of cloudy skies. Another source of error is transparent surfaces such as glass windows, where subtle variations of transparency or reflections may cause the depth to oscillate between the glass and the scene behind it – a common issue of depth estimators.

References

[1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1

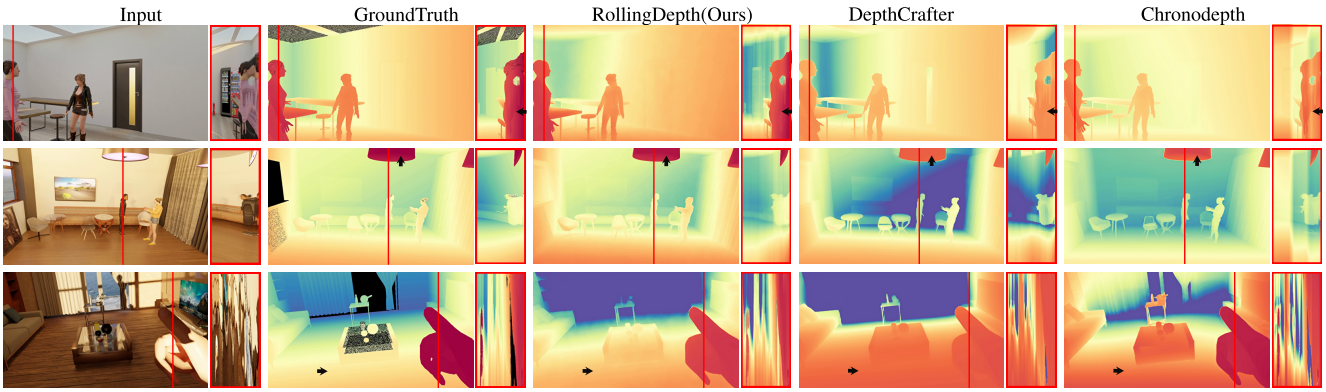

Figure S2. Examples of PointOdyssey samples that challenge video models. In the cases above, the (inverse) depth range varies significantly across frames. The arrows highlight situations where video models yield distorted depth maps. In the first two rows, this occurs in regions where the depth deviates significantly from the surrounding scene. In the last row, the depth predictions get drawn towards the near plane to match the object close to the camera, biasing the depth in the far field.

Figure S3. The two samples on the left show incorrect depth predictions in the cloudy sky. The two samples on the right show inconsistencies between different frames of the same video, where the depth at the glass windows fluctuates between the solid and transparent states.

[2] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. *arXiv:2409.11355*, 2024. 1

[3] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[4] 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:2409.02095*, 2024. 1

[5] 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*, 2024. 1

[6] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *WACV*, 2024. 1

[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:2406.01493*, 2024. 1

[8] Zhanghao Sun, Wei Ye, Jinhui Xiong, Gyeongmin Choe, Jialiang Wang, Shuochen Su, and Rakesh Ranjan. Consistent direct time-of-flight video depth super-resolution. *arXiv:2211.08658*, 2022. 1

[9] Yiran Wang, Zhiyu Pan, Xingyi Li, Zhiguo Cao, Ke Xian, and Jianming Zhang. Less is more: Consistent video depth estimation with masked frames modeling. In *ACM MM*, 2022. 2

[10] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022. 2

[11] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. PointOdyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 1