Video Depth Anything: Consistent Depth Estimation for Super-Long Videos

Supplementary Material

1. More Qualitative Results

We present more qualitative comparisons among different approaches for static images and evaluation videos.

In-the-wild image results. Static image depth estimation results are shown in Fig. 1. DepthCrafter [5] and Depth Any Video [17] exhibit poor performance on oil paintings. DepthCrafter [5] also struggles with transparent objects such as glass and water. Compared with these methods, our model demonstrates superior depth estimation results in complex scenarios. Moreover, our model shows depth estimation results for static images that are comparable to those of Depth-Anything-V2 [18], demonstrating that we have successfully transformed Depth-Anything-V2 into a video depth model without compromising its spatial accuracy.

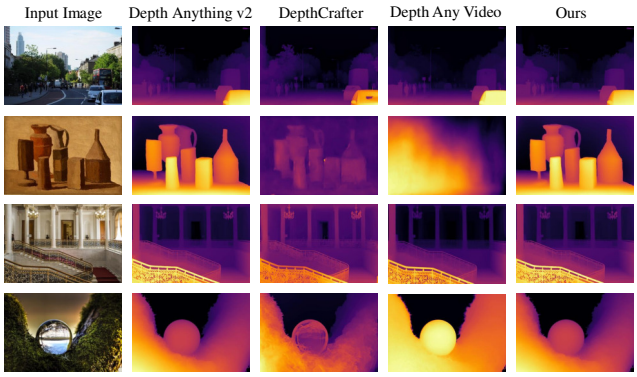

Figure 1. **Qualitative comparison for static image depth estimation.** We compare our model with Depth-Anything-V2 [18], DepthCrafter [5], and Depth Any Video [17] on static image depth estimation. Our model demonstrates visualization results comparable to those of Depth-Anything-V2 [18].

Evaluation video results. We showcase five video visualization results from the evaluation datasets Scannet [3] and Bonn [10] in Fig. 2. For enhanced visualization, all predicted video depths are aligned to the ground truth video depths using the same method as in the evaluation. DepthCrafter [5] exhibits depth drift in long videos, as indicated by the blue boxes. Moreover, our model demonstrates superior depth accuracy compared to DepthCrafter [5], as highlighted in the red boxes.

2. Short video depth quantitative results

We compare our model with DepthCrafter [5] and Depth Any Video [17] on the KITTI [4], Bonn [10], and Scannet [3] datasets, with frame lengths of 110, 110, and 90, respectively, corresponding to the settings in [5]. As shown in Tab. 1, our model demonstrates a significant advantage of approximately 7% over both DepthCrafter [5] and Depth Any Video [17] on the Scannet dataset [3]. On the KITTI dataset [4], our model significantly outperforms DepthCrafter [5] by about 7%. Additionally, our model achieves comparable results on Bonn [10] and KITTI [4] compared to Depth Any Video [17]. It is worth noting that the parameters of our model and the video depth data used for training are significantly smaller than those of DepthCrafter [5] and Depth Any Video [17], demonstrating the effectiveness and efficiency of our method.

3. Limitations and future work

Our model is trained primarily on publicly available video depth datasets, which may limit its capabilities due to the data quantity. We believe that with more data, the model's performance can be further improved, and the backbone network can be unlocked for fine-tuning. Additionally, although our model is significantly more computationally efficient than the baselines, it still faces challenges in handling streaming videos, which we leave as future work.

4. More Details of Pipeline

Spatiotemporal head details. Among the four temporal layers, two are inserted after the Reassemble layers at the two smallest resolutions, and the other two are inserted before the last two Fusion layers.

The shape of the feature is transformed into $(B \times H_f \times W_f) \times N \times C$ before each temporal layer and is transformed back to $(B \times N) \times C \times H_f \times W_f$ after each temporal layer. Here, B denotes the batch size, N represents the number of frames in the video clip, H_f and W_f are the height and width of the feature, respectively, and C represents the number of channels in the feature, as shown in Fig. 3

Image distillation details. We follow the approach in [18] and use a teacher model that comprises a ViT-giant encoder and is trained on synthetic datasets. The loss function used for distillation is identical to the spatial loss employed for video depth data.

Training dataset details. For video training, we utilize four synthetic datasets with precise depth annotations: TartanAir [15], VKITTI [2], PointOdyssey [19], and IRS [13], totally 0.55 million frames. The TartanAir [15], VKITTI [2], PointOdyssey [19], and IRS [13] datasets contain 0.31M,

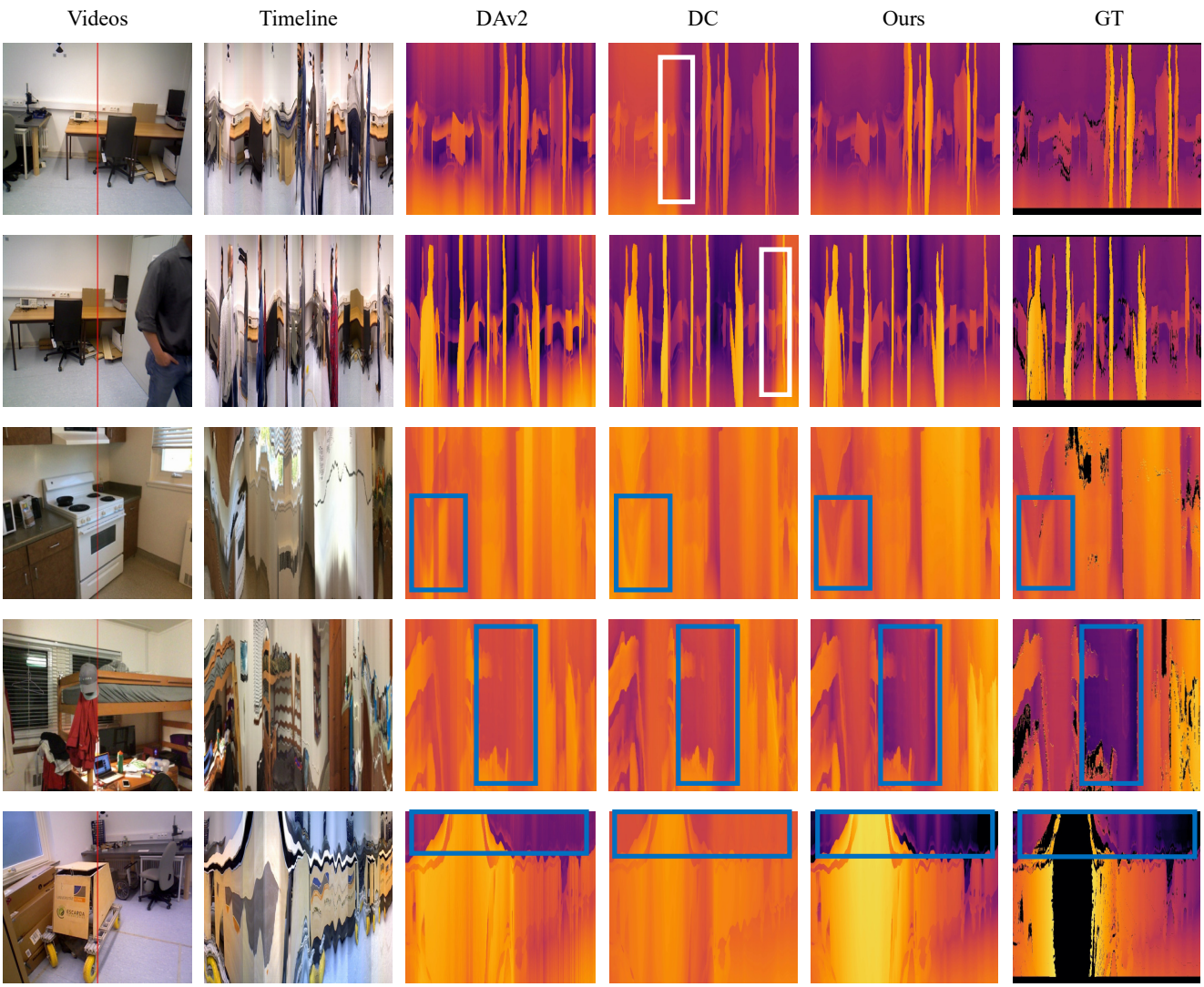

Figure 2. **Qualitative comparison for real-world long video depth estimation.** We compare with Depth-Anything-V2 [18] and DepthCrafter [5] on 500-frames videos from Scannet [3] and Bonn [10] . We show changes in color and depth over time at the vertical red line in videos. White boxes show inconsistent estimation. Blue boxes show our algorithm has higher accuracy.

Method / Metrics	Params(M)	# Video Training Data(M)	KITTI(110) [4]		Bonn(110) [10]		Scannet(90) [3]	
			AbsRel (↓)	δ_1 (↑)	AbsRel (↓)	δ_1 (↑)	AbsRel (↓)	δ_1 (↑)
DepthCrafter	2156.7	10.5~40.5	0.111	0.885	0.066	<u>0.979</u>	0.125	0.848
DepthAnyVideo	1422.8	6	**0.073**	**0.957**	**0.051**	**0.981**	<u>0.112</u>	<u>0.883</u>
VDA-L (Ours)	381.8	0.55	<u>0.079</u>	<u>0.950</u>	<u>0.053</u>	0.972	**0.075**	**0.954**

Table 1. **Zero-shot short video depth estimation results**. We compare with DepthCrafter [5] and DepthAnyVideo [17] in short video depth benchmark. "VDA-L" denotes our model with ViT-Large backbone. The default inference resolution of our model is set to 518 pixels on the short side, maintaining the aspect ratio. The **best** and the <u>second best</u> results are highlighted.

0.04M, 0.1M, and 0.1M frames, respectively. Additionally, 0.18 million frames from wild binocular videos labeled with [7] are included for training. We also incorporate a subset of real-world unlabeled datasets from [18] for single image supervision, totaling 0.62 million frames. Notably, we excluded 0.13M frames from PointOdyssey [19] that do not contain background depth ground truth, resulting in our usage of only half of the original dataset. Due to the uneven data distribution across the four training datasets, we employ a uniform sampler to ensure that each dataset contributes

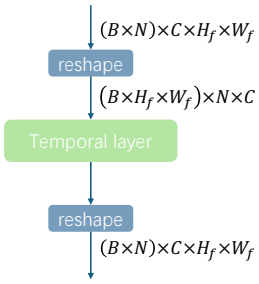

Figure 3. **Temporal layer.** The feature shape is adjusted for temporal attention.

equally during training.

Implementation Details The weights are initialized from Depth Anything V2 [18]. Training comprises two stages. In the first stage, synthetic and wild binocular data are used. In the second stage, synthetic videos and unlabeled single images are employed. Models labeled '-Syn' in the main paper are exceptionally trained using synthetic videos and unlabeled images in a single stage. Besides the loss defined in Equation 4 of the main paper used for synthetic videos, unlabeled single images are supervised using the same method as described in [18]. During training, we uniformly sample video clips of 32 frames from each dataset, resize the shorter edge of images to 518 pixels, and perform random center cropping, resulting in training clips with a resolution of $518 \times 518 \times 32$. We use the AdamW [8] optimizer with a cosine scheduler, setting the base learning rate to $1e^{-4}$. The batch size is set to 16 for video frames, each with a length of 32 frames, and 128 for image datasets. The loss weights for the single frame loss, TGM loss, and distillation loss are set to 1.0, 10.0, and 0.5, respectively.

5. More Details of Evaluation

Evaluation dataset details. We use a total of five datasets for video depth evaluation: KITTI [4], Scannet [3], Bonn [10], NYUv2 [9], and Sintel [1]. Specifically, we use Scannet [3] and NYUv2 [9] for static indoor scenes, Bonn [10] for dynamic indoor scenes, KITTI [4] for outdoor scenes, and Sintel [1] for wild scenes. For NYUv2 [9], we sample 8 videos from the original dataset, which contains 36 videos. Our evaluation comprises three different settings: long videos, long videos with different frame lengths, and short videos. For the long video evaluation, we use all five datasets and set the maximum frame length to 500 for each video. For the evaluation of long videos with different frame lengths, we select subsets of videos with frame lengths greater than 500 from Scannet [3], Bonn [10], and NYUv2 [9]. For the short video evaluation, we use KITTI [4], Bonn [10], and Scannet [3], setting the maximum frame lengths to 110, 110, and 90, respectively, in accordance with the settings in DepthCrafter [5]. In addition to video depth evaluation, we also assess our model's performance on static images. Following [18], we perform evaluations on five image benchmarks: KITTI [4], Sintel [1], NYUv2 [9], ETH3D [12], and DIODE [6]. To ensure a fair comparison, all evaluation videos and images are excluded from the training datasets.

Evaluation metric details. All video metrics we evaluated are based on ground truth depth. Specifically, we use the least squares method to compute the optimal scale and shift to align the entire inferred video inverse depth with the ground truth inverse depth. The aligned inferred video inverse depth is then transformed into depth, which is subsequently used to compute the video metrics with the ground truth depth. For geometric accuracy, we compute the Absolute Relative Error (AbsRel) and δ_1 metrics, following the procedures outlined in [5, 18]. To assess temporal stability, we use the Temporal Alignment Error (TAE) metric in [17], to measure the reprojection error of the depth maps between consecutive frames. We use Equation 1.

$$TAE = \frac{1}{2(N-1)} \sum_{k=1}^{N-1} AbsRel(f(\hat{x}_d^k, p^k), \hat{x}_d^{k+1}) + AbsRel(f(\hat{x}_d^{k+1}, p_-^{k+1}), \hat{x}_d^k) \quad (1)$$

Here, f represents the projection function that maps the depth $\hat{x}_d^k$ from the k-th frame to the $(k+1)$-th frame using the transformation matrix p^k. p_-^{k+1} is the inverse matrix for inverse projection. N denotes the number of frames.

Baseline implementations. We obtain the inferences of DepthCrafter [5], Depth Any Video [17], and NVDS [16] using the respective inference code provided by the authors. Specifically, DepthCrafter [5] employs different inference resolutions for different datasets. Depth Any Video [17] infers with a maximum dimension of 1024. NVDS [16] performs inference on a video twice, with a minimum dimension of 384, once in the forward direction and once in the backward direction, and computes the mean result from these two passes. For Depth-Anything-V2 [18], we obtain the video depth results by inferring each frame individually with a minimum dimension of 518.

6. Applications

Dense point cloud generation. By aligning single frame with metric depth, which can be obtained from a metric depth model or a sparse point cloud acquired through SLAM, our model can generate a depth point cloud for the entire environment using camera information. The generated point cloud can then be transformed into a mesh and utilized for 3D reconstruction, AR, and VR applications. We present

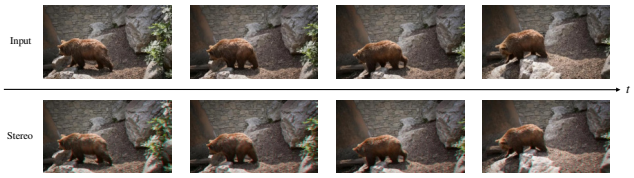

Figure 4. **3D Video Conversion.** A video from the DAVIS dataset [11] is transformed into a 3D video using our model.

a point cloud generation case in Fig. 5. Here, we sample 10 frames spanning approximately 5 seconds from the KITTI dataset [4]. After obtaining the inferred inverse depth, we compute the global scale and shift by aligning the first frame with the corresponding metric inverse depth. We then apply the affine transformation to the entire set of inverse depth frames and convert them to depth. The final point cloud is generated by merging the point clouds from each frame. As shown in Fig. 5, our model generates a clean and regular point cloud compared to DepthCrafter [5] and Depth Any Video [17]. Point cloud generation for wild videos is illustrated in Fig. 6. Compared to DepthCrafter [5] and DepthAnyVideo [17], our model produces more regular point clouds.

3D Video Conversion. Our model can be used to generate 3D videos. Compared to 3D videos generated by monocular depth models, those produced by our video depth model exhibit smoother and more consistent 3D effects. An example is presented in Fig.4.

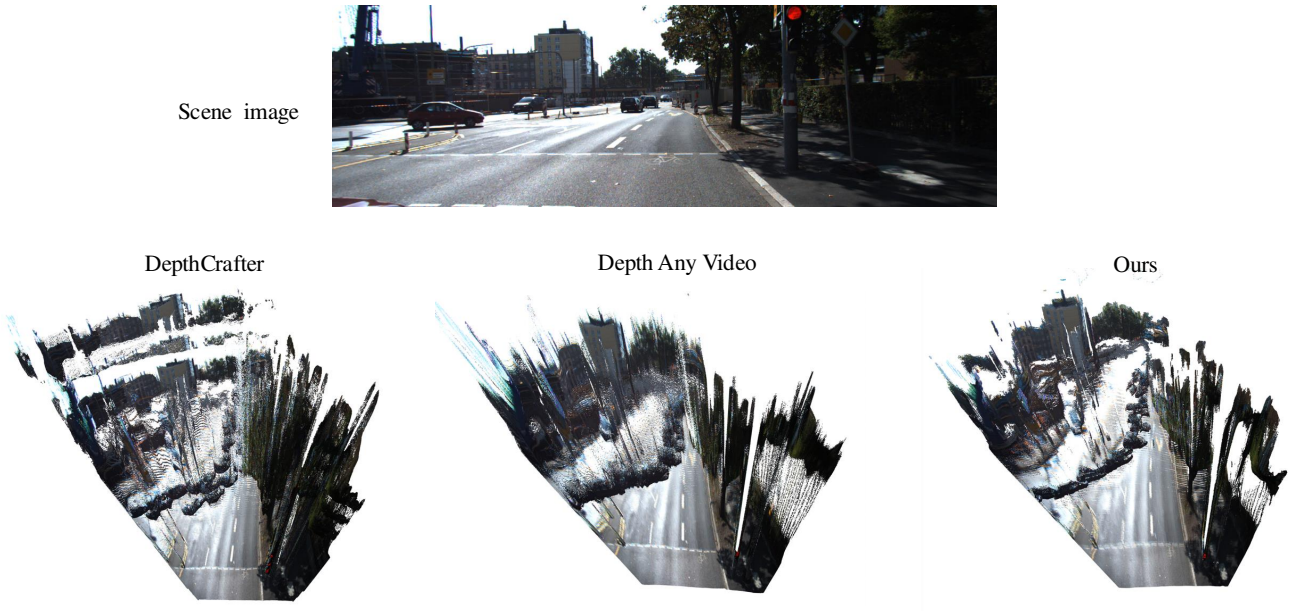

Figure 5. **Dense point cloud generation.** We compare our model with DepthCrafter [5] and DepthAnyVideo [17] for dense point cloud generation on the KITTI dataset [4]. Our model generates a clean and regular point cloud from multiple frames spanning approximately 5 seconds. In contrast, the point cloud generated by DepthCrafter [5] contains several obvious discontinuous layers. DepthAnyVideo [17] produces a point cloud with numerous noisy outliers and noticeable distortion in distant views.

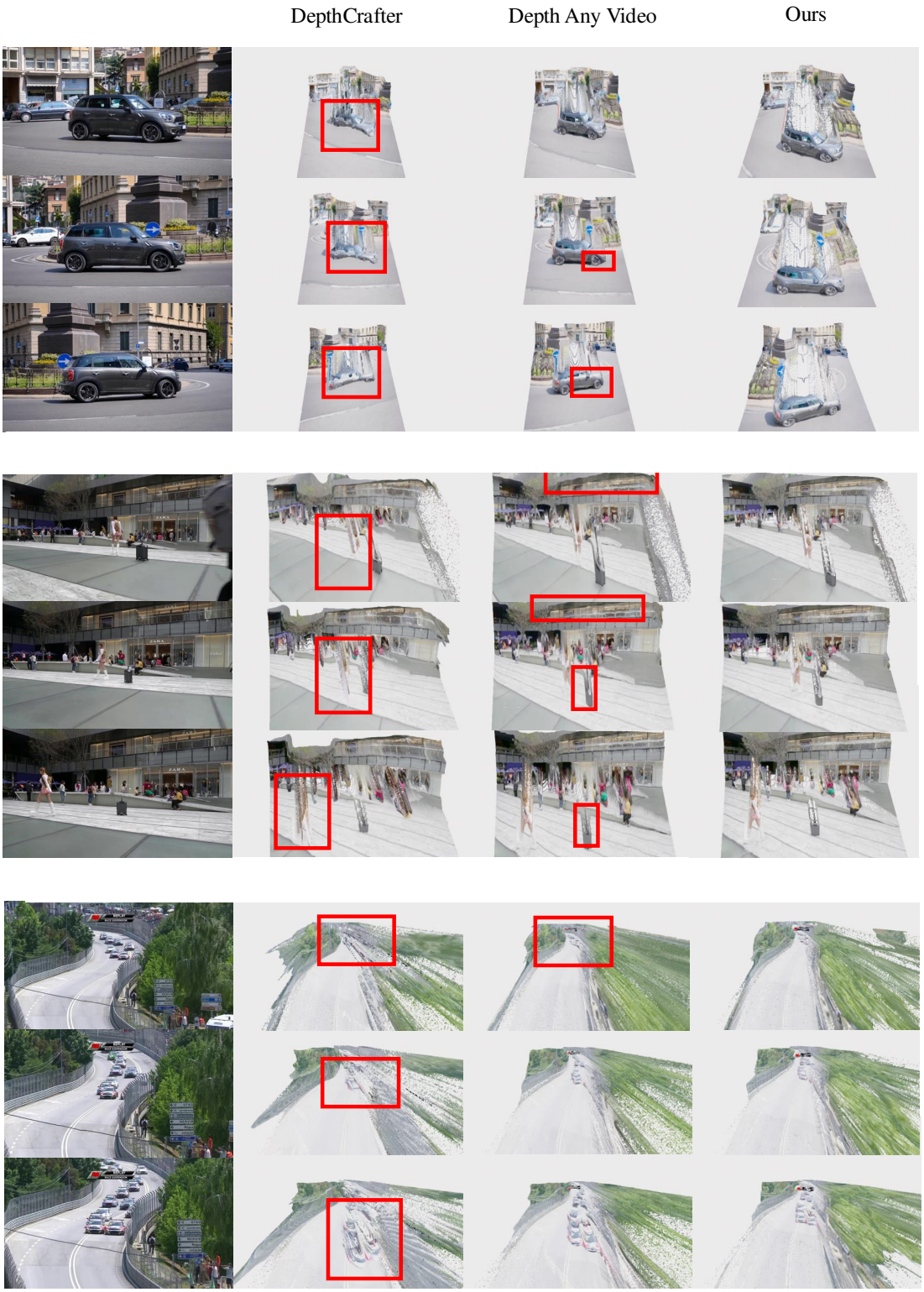

Figure 6. **Point cloud generation for wild videos.** We compare our method with DepthCrafter [5] and DepthAnyVideo [17] using three videos from DAVIS dataset [11]. Camera intrinsics, along with aligned scale and shift parameters, are derived from processing the first frame of each video through MoGe [14]. Point cloud distortions are highlighted with red boxes.

References

[1] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. *A Naturalistic Open Source Movie for Optical Flow Evaluation*, page 611–625. Jan 2012. 3

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 1

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 3

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1, 2, 3, 4, 5

[5] 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, 2, 3, 4, 5, 6

[6] Vasiljevic Igor, Kolkin Nicholas, Shanyi Zhang, Ruotian Luo, Haochen Wang, FalconZ. Dai, AndreaF. Daniele, Mohammadreza Mostajabi, Steven Basart, MatthewR. Walter, and Gregory Shakhnarovich. Diode: A dense indoor and outdoor depth dataset. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*, Aug 2019. 3

[7] Junpeng Jing, Ye Mao, and Krystian Mikolajczyk. Match-stereo-videos: Bidirectional alignment for consistent dynamic stereo matching. 2024. 2

[8] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[9] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3

[10] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. 2019. 1, 2, 3

[11] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 4, 6

[12] Thomas Schops, Johannes L. Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 3

[13] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation, 2021. 1

[14] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision, 2024. 6

[15] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 1

[16] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9466–9476, 2023. 3

[17] Honghui Yang, Di Huang, Wei Yin, Chunhua Shen, Haifeng Liu, Xiaofei He, Binbin Lin, Wanli Ouyang, and Tong He. Depth any video with scalable synthetic data. *arXiv preprint arXiv:2410.10815*, 2024. 1, 2, 3, 4, 5, 6

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 1, 2, 3

[19] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 1, 2