Supplementary Material for GeometryCrafter

Tian-Xing Xu[1] Xiangjun Gao[3] Wenbo Hu[2 †] Xiaoyu Li[2] Song-Hai Zhang[4,1 †] Ying Shan[2]
[1] Tsinghua University [2] ARC Lab, Tencent PCG [3] HKUST [4] Qinghai University
Project Page: https://geometrycrafter.github.io

Table 1. **An overview of the training datasets.**

Dataset	Domain	#Frames	#Videos
3DKenBurns [22]	In-the-wild	76K	526
DynamicReplica [14]	Indoor/Outdoor	145K	1126
GTA-SfM [28]	Outdoor/In-the-wild	19K	234
Hypersim [25]	Indoor	75K	×
IRS [29]	Indoor	103K	722
MatrixCity [17]	Outdoor/Driving	452K	3029
MidAir [7]	Outdoor/In-the-wild	357K	2433
MVS-Synth [13]	Outdoor/Driving	12K	120
Spring [21]	In-the-wild	5K	49
Structured3D [35]	Indoor	71K	×
Synthia [26]	Outdoor/Driving	178K	1276
TartanAir [31]	In-the-wild	306K	2245
UrbanSyn [10]	Outdoor/Driving	7K	×
VirtualKitti2 [3]	Driving	43K	320
Total	-	1.85M	12K

1. Datasets

1.1. Training Datasets

We collect 14 open-source synthetic RGBD datasets to facilitate the training of GeometryCrafter, among which 11 can be composited into video sequences. To construct the training video dataset, we extract non-overlapping segments with a sequence length not exceeding 150 frames. An overview of the training datasets is provided in Tab. 1, categorized into four distinct domains: indoor, outdoor, in-the-wild and driving scenarios. It is noteworthy that the frame count may slightly differ from the original datasets, owing to the exclusion of invalid frames. To ensure computational efficiency and adhere to GPU memory constraints, we preprocess all images and videos to a standardized resolution of 320 × 640. Specifically, we apply cover resizing while preserving the original aspect ratio, followed by center cropping to achieve the desired resolution. Additionally, we implement random resizing as a technique for augmenting camera intrinsics.

1.2. Evaluation Datasets

We exhaustively evaluate GeometryCrafter and previous state-of-the-art methods using seven datasets with ground truth labels that remain entirely unseen during the training phase. Notably, to ensure compatibility with most baselines, such as MoGe [30] and UniDepth [24], which necessitate an input image aspect ratio of less than 2, we preprocess the evaluation datasets in the following manner:

- **GMU Kitchens** [9]: All scenarios are employed for evaluation. For each scenario, we extract 110 frames with a stride of 2 to ensure extensive spatial coverage while preserving temporal coherence. To reduce memory usage during evaluation, we downsample the generated 1920p videos and ground truth depth maps to a resolution of 960 × 540.
- **ScanNet** [5]: Following DepthCrafter [12], we select 100 scenes from the test split for evaluation, wherein each video comprises 90 frames with a frame stride of 3. Due to the discrepancy in resolutions between the RGB images and depth maps, we first resize the RGB images to align with the depth maps, followed by center cropping to remove the black space around RGB images, yielding videos of resolution 624 × 464 .
- **DDAD** [11]: All 50 sequences from the validation split of the DDAD dataset are utilized for evaluation, with sequence lengths of either 50 or 100 frames. Owing to the high memory demands of the raw resolution 1936 × 1216, we apply center cropping to reduce the resolution to 1920 × 1152, followed by downsampling to 640 × 384 for evaluation. The ground truth depth maps, acquired via LiDAR sensors, are inherently sparse; consequently, the preprocessing has negligible influence on the comparative analysis of various methods.
- **KITTI** [8]: All sequence in the valid split of depth annotated dataset are used evaluation. For excessively long video sequences, we extract the initial 110 frames, resulting in 13 videos with sequence lengths ranging between 67 and 110 frames. Given that the original resolution of 1242 × 375 fails to conform to the aspect ratio requirements of most baseline methods, we apply center cropping to achieve a resolution of 736 × 368.

- **Monkaa** [20]:We select 9 scenes from the original dataset for evaluation, truncating each video sequence to 110 frames while maintaining the original resolution of 960×540. To derive valid masks, we manually annotate the sky regions within each sequence.
- **Sintel** [2]: All sequences within the training split are employed for evaluation, with sequence lengths ranging between 21 and 50 frames. Given the original resolution of 1024×436 for each image, we apply cropping to achieve a standardized resolution of 872×436.
- **DIODE** [27]: We utilize all 771 images from the validation split of DIODE for evaluation purposes. To address the noisy values along the edges of objects within the depth maps, we employ a Canny filter to detect edge regions, subsequently refining the valid masks based on the filtering outcomes.

2. Loss Functions of VAE and UNet

To train the point map VAE, we define the loss function $\mathcal{L}_{\text{pmap}}$ to measure the reconstruction errors of point maps. The reconstruction loss $\mathcal{L}_{\text{recon}}$ for each valid pixel is defined as the L_1 norm

$$\mathcal{L}_{\text{recon}} = \sum_{p \in \mathcal{M}} ||z_p - \widehat{z}_p||_1 + \sum_{p \in \mathcal{M}} ||\theta_{\text{diag}} - \widehat{\theta}_{\text{diag}}||_1 \quad (1)$$

where $\mathcal{M} = \{p | \mathbf{m}(p) = 1\}$ and $\widehat{z}_p, \widehat{\theta}_{\text{diag}}$ are the reconstructed values at pixel p. To enhance surface quality, we additionally supervise the normal maps derived from the reconstructed point maps and the ground truth:

$$\mathcal{L}_{\text{n}} = \sum_{p \in \mathcal{M}} (1 - n_p \cdot \widehat{n}_p) \quad (2)$$

To enhance supervision for local geometry, we draw inspiration from MoGe [30] and propose a multi-scale depth loss function that measures the alignment between reconstructed and ground truth depth maps within local regions $\mathcal{H}_\alpha$, parameterized by scale α

$$\mathcal{L}_{\text{ms}} = \sum_{\mathcal{H}_\alpha} \sum_{p \in \mathcal{H}_\alpha \& p \in \mathcal{M}} ||(z_p - \overline{z}_{p,\mathcal{H}_\alpha}) - (\widehat{z}_p - \widetilde{z}_{p,\mathcal{H}_\alpha})||_1 \quad (3)$$

Here, $\overline{z}_{p,\mathcal{H}_\alpha}$ and $\widetilde{z}_{p,\mathcal{H}_\alpha}$ are the mean value of predicted and ground truth depth map defined on local region $\mathcal{H}_\alpha$. In practice, we split video frames into non-overlapped patches of size $\frac{W}{\alpha} \times \frac{H}{\alpha}$ to define the local regions. The reconstruction objective $\mathcal{L}_{\text{pmap}}$ is thus given by

$$\mathcal{L}_{\text{pmap}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{ms}} + \lambda_{\text{n}} \mathcal{L}_{\text{n}} \quad (4)$$

Following LayerDiffuse [33], we apply the frozen decoder $\mathcal{D}_{\text{SVD}}$ to measure the extent to which the latent offset disrupts the modified latent distribution during training, given by

$$\mathcal{L}_{\text{identity}} = ||\mathbf{x}_{\text{disp}} - \widehat{\mathbf{x}}_{\text{disp}}||_2^2 = ||\mathbf{x}_{\text{disp}} - \mathcal{D}_{\text{SVD}}(\mathbf{z}_{\text{pmap}})||_2^2 \quad (5)$$

where $|| \cdot ||_2^2$ denotes the mean square loss function. Additionally, we introduce a mask loss to regularize the reconstructed valid mask:

$$\mathcal{L}_{\text{mask}} = ||\widehat{\mathbf{m}} - \mathbf{m}||_2^2 \quad (6)$$

where $\mathbf{m} \in \mathbb{R}^{T \times H \times W}$ is the ground truth valid mask. The final training objective of VAE is defined as

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{identity}} + \mathcal{L}_{\text{pmap}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}} \quad (7)$$

To finetune the UNet D_θ parameters on the adjusted latent space obtained by our proposed point map VAE, we employ the objective $\mathcal{L}_{\text{UNet}}$ for supervision, written as

$$\mathbb{E}_{\mathbf{z}_t \sim p(\mathbf{z}, \sigma_t), \sigma_t \sim p(\sigma)} [\lambda_{\sigma_t} ||D_\theta(\mathbf{z}_t; \sigma_t, \mathbf{z}_{\mathbf{v}}, \mathbf{z}_{\text{prior}}) - \mathbf{z}_{\text{pmap}}||_2^2] \quad (8)$$

Here the noisy latent input $\mathbf{z}_t$ is generated by adding Gaussian noise n to the latent code $\mathbf{z}_{\text{pmap}}$. $\mathbf{z}_{\mathbf{v}}$ is the conditional latent code of input video. $\mathbf{z}_{\text{prior}}$ denotes the per-frame geometry priors provided by MoGe [30]. σ_t denotes noise level at time t, satisfying $\log \sigma_t \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}})$ with $P_{\text{mean}} = 0.7$ and $P_{\text{std}} = 1.6$ adopted in the EDM [15] noise schedule and λ_{σ_t} is a weight parameter at time t.

3. More Implementation Details

For the point map VAE design, we reuse the architecture of SVD's VAE with minor modification: we adopt zero convolution initialization [34] to the output convolution layer of encoder and apply a scale factor of 0.1 to ensure that latent offsets do not disrupt the latent distribution during the initial stage of training. Inspired by the training strategy of SVD, we first train the model from scratch with an AdamW [19] optimizer on RGBD images, with a fixed learning rate of 1e-4 for 40K iterations. Then, we finetune the temporal layers in the decoder for another 20K iterations on video data. The batch sizes are set to be 64 and 8 for the respective stages, with sequence lengths randomly sampled from $[1, 8]$ for video data in the second stage. For the UNet denoiser, we initialize UNet with the pretrained parameters provided by DepthCrafter [12], finetuning it with a learning rate of 1e-5 and a batch size of 8. We train our diffusion UNet in two stages, where we first train it on videos with sequence lengths sampled from [1, 25] frames to adapt the model to our generation task, and then solely finetune the temporal layers with the sequence length randomly sampled from [1, 110] frames due to the limitation of GPU memory. After training, the UNet can process videos with varying lengths (e.g., 1 to 110 frames) at a time. Both components are

Table 2. **Inference time.of different components on** 448×768 **videos with 110 frames.**

Method	Per-frame Prior	Encoder	UNet	Decoder	Total
Ours(G)	0.1	0.04	0.04	0.08	0.27s/frame
Ours(D)	0.1	0.04	0.01	0.08	0.24s/frame

trained on 320×640 images or videos for efficiency, with random resizing and center cropping applied for data augmentation and resolution alignment. All trainings are conducted on 8 GPUs, with the entire process requiring about 3 days.

4. Camera Pose Estimation

To recover camera poses from point maps, we need to establish correspondences of the static background across frames. We first obtain the dynamic object masks by annotating the first frame using SegmentAnything [16], and then apply XMem [4], a robust method for video object segmentation, to generate the dynamic target masks for the subsequent frames. Given the dynamic masks, we adopt SuperPoint [6] to detect reliable points of interest in the first frame and filter out those points that belong to the dynamic objects. After that, we employ SpaTracker [32] to generate the 2D trajectory of each point, which is subsequently used to form the constraints for the camera pose optimization. Let p_t denote the XY coordinate of a 2D trajectory at time step t, the 2D point p_t can be lifted to the world coordinate $\widetilde{p}_t$ using the following transformation

$$\widetilde{p}_t = W_t^{-1} \pi_{K_t}^{-1}(p_t, D(p_t)) \tag{9}$$

Here W_t denotes the camera pose at time step t, $D(\cdot)$ denotes the scale-invariant depth value obtained from our predicted point maps and $\pi_{K_t}^{-1}$ refers to the back-projection of the 2D point to camera coordinate with camera intrinsic K, which can also be estimated from the point maps. For time step t', the 2D projected coordinate should align with the trajectory position at timestep t'. Therefore, we formulate the camera pose estimation as the following problem

$$\min_{W_1 \ldots W_T} \sum_{i,j \in [1 \ldots T]} ||\pi_{K_j} W_j W_i^{-1} \pi_{K_i}^{-1}[p_i, D(p_i)] - [p_j, D(p_j)]||_2^2 \tag{10}$$

Due to the sequence length limitation of SpaTracker (12 for each segment), we apply a shifted window strategy with 6 overlapping frames to regularize the optimization of all camera poses. The optimization process for each scene takes from less than 1 minute to several minutes, depending on the number of frames.

5. Limitations

The major limitation of our method is the expensive computation and memory cost, primarily attributing to the large model size inherent in both the VAE and U-Net architectures. As shown in Tab. 2, we provide a comparison of the inference times of different components in GeometryCrafter. Our experiments are conducted on a single GPU, revealing that the decoder of the point map VAE is the bottleneck during inference. How to design a lightweight decoder capable of producing temporally consistent outputs will be a focal point of our future works.

6. More results

In the following pages, we provide more visual results of our method. We provide more results on Sora [1]-generated videos to demonstrate the temporal consistency and geometry quality of our method, as shown in Fig. 1. For comprehensive comparison with MGE methods, we provide a visual analysis in Fig. 2. Our method achieves robust and sharp point map estimation compared to other methods. In contrast, UniDepth [24] fails to segment the sky region from the input frames, while MoGe [30] struggles to handle fine-grained structure. Fig. 3 and Fig. 4 shows the point maps aligned with the optimized camera poses, where the rows from left to right are 4 input frames uniformly sampled from the whole video and two views of aligned point maps in the world coordinates. We only provide the results of concatenating 8 point maps sampled from the predicted point sequences for better visualization.

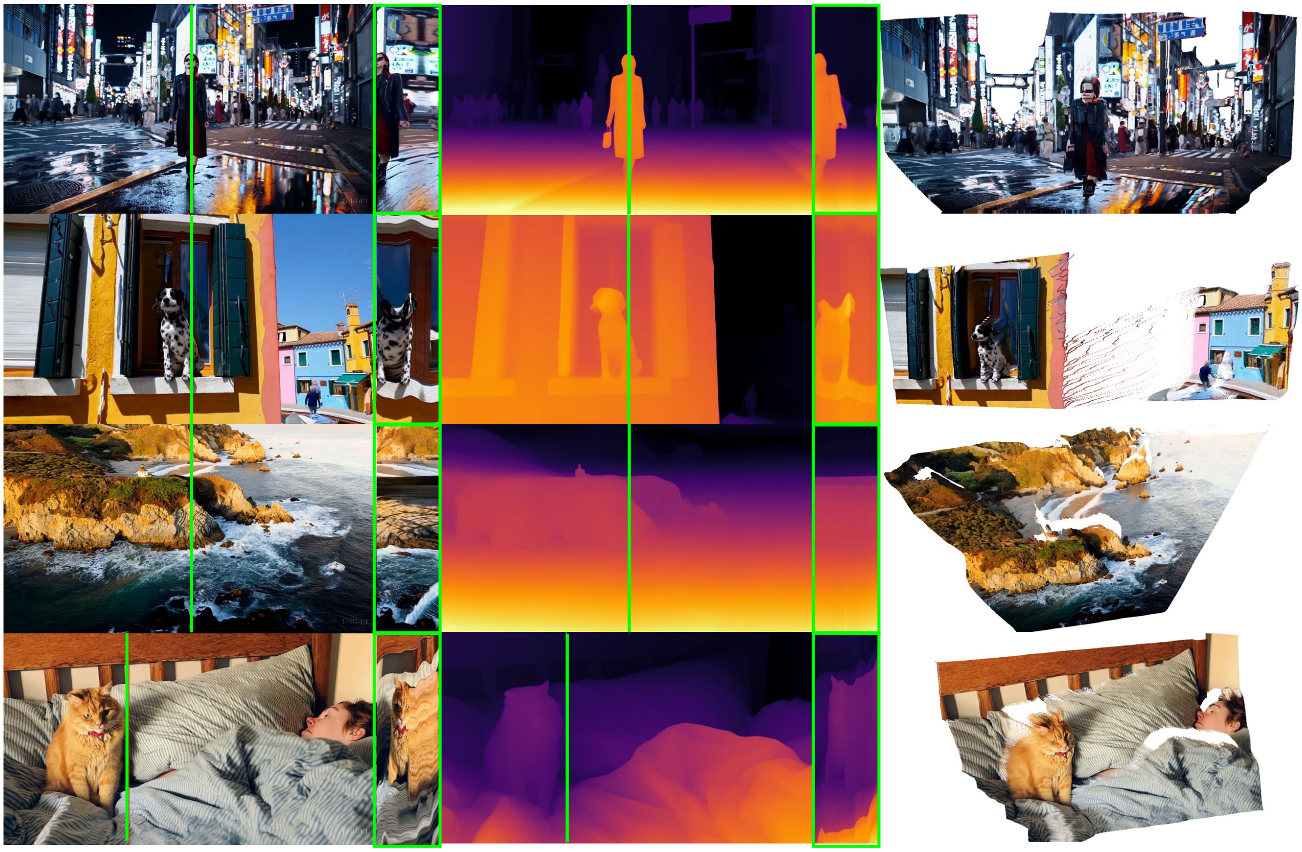

Figure 1. **Visual results on Sora-generated videos.** The rows from left to right are the input videos, the disparity maps and the point cloud of the first frame.

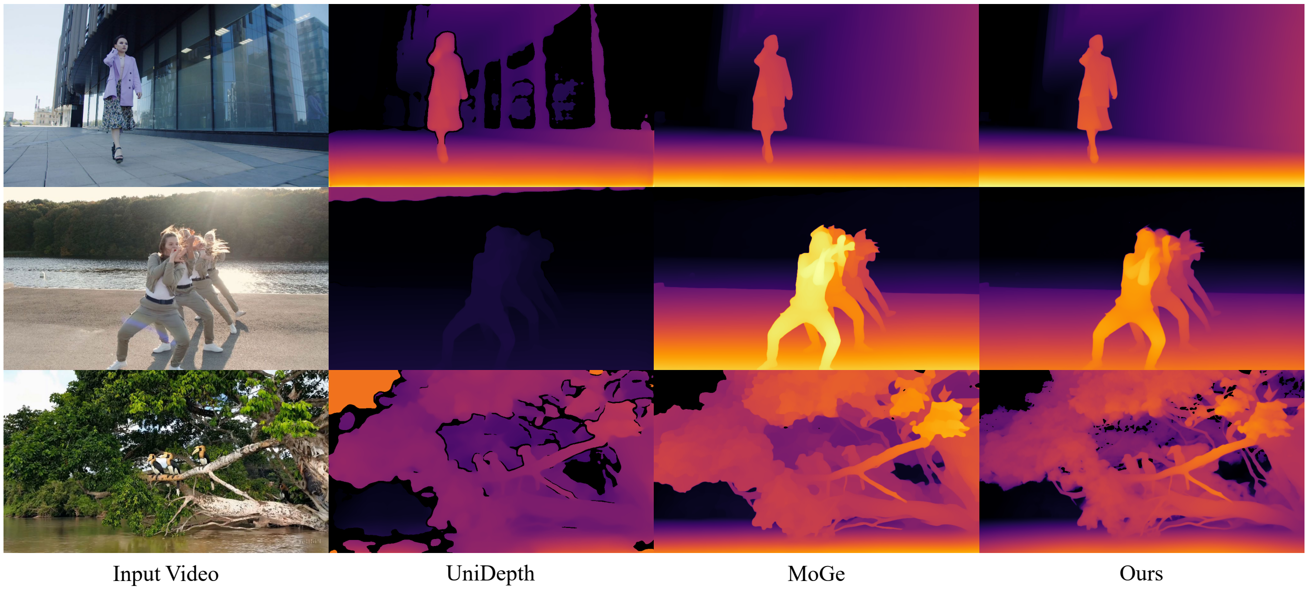

Figure 2. **Visual comparison with monocular geometry estimation methods.** All point maps are converted to disparity maps for better visualization the sharpness of depth prediction.

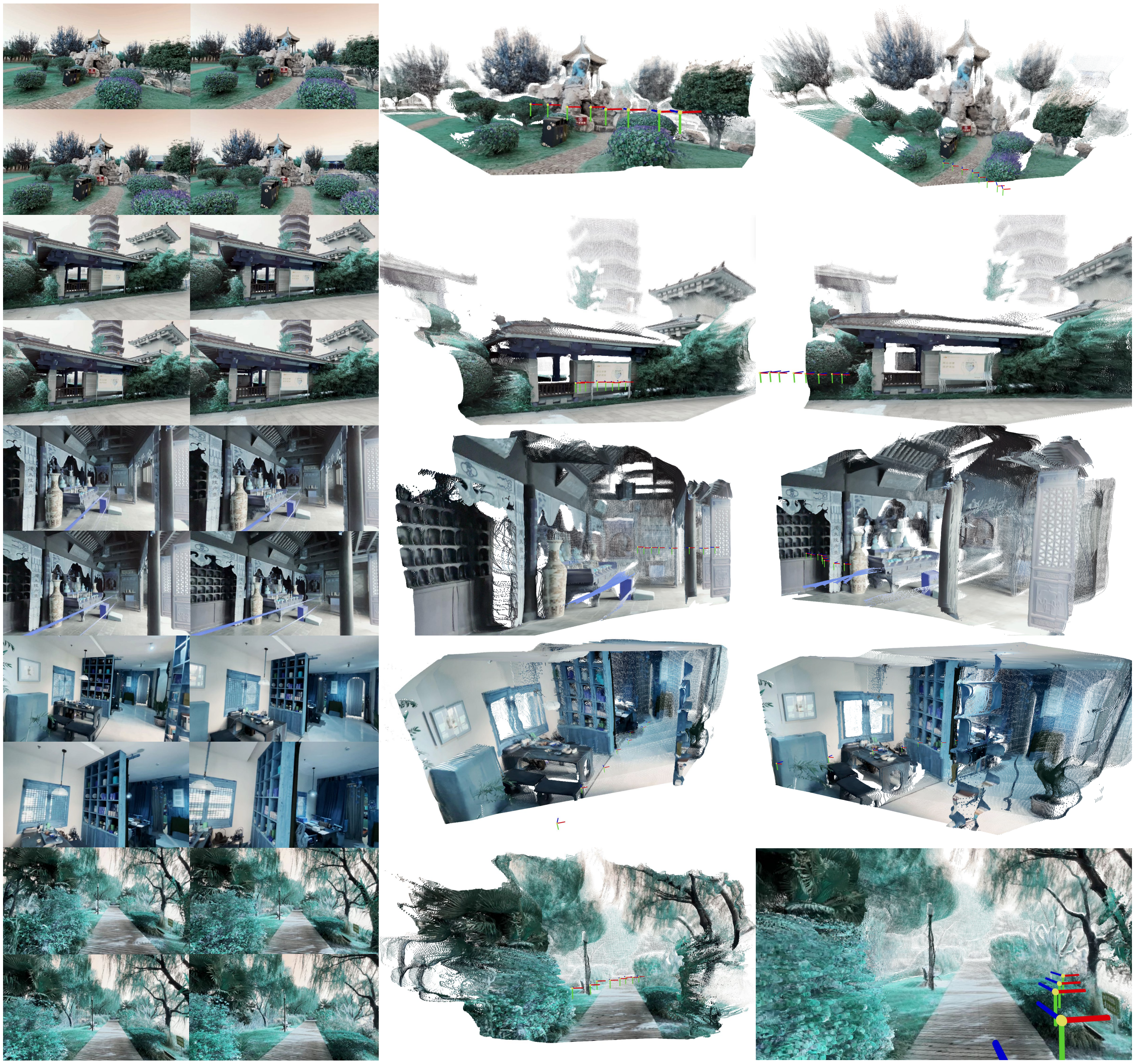

Figure 3. **Visual results on DL3DV [18] with camera poses estimated from the output point maps.** We concatenate 8 aligned point maps from the original point map sequence for visualization.

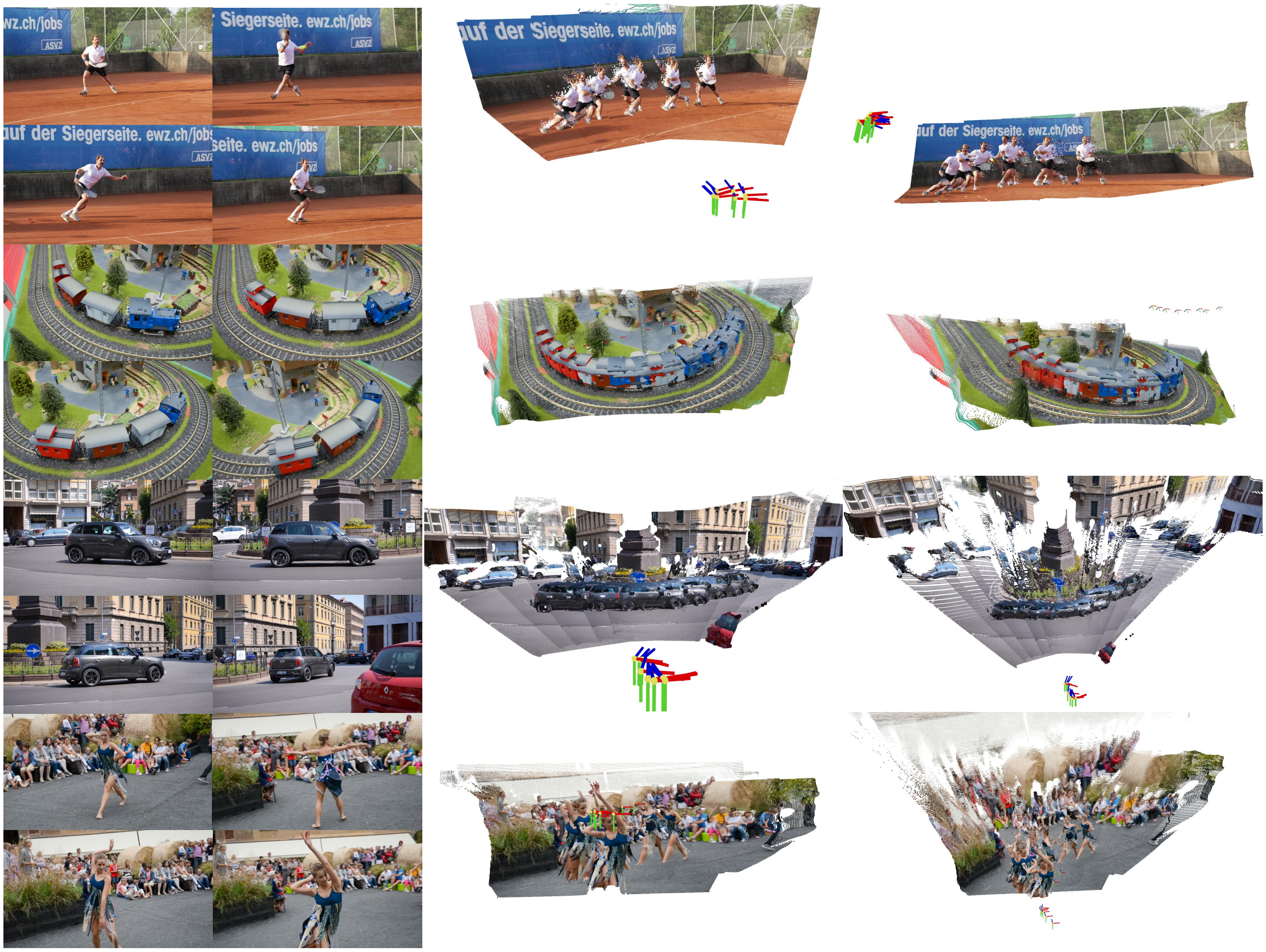

Figure 4. **Visual results on DAVIS [23] with camera poses estimated from the output point maps.** We concatenate 8 aligned point maps from the original point map sequence for visualization.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. 3

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012. 2

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

[4] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 3

[5] 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

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 3

[7] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 1

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1

[9] Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, and Jana Košecká. Multiview rgb-d dataset for object instance detection. In *2016 Fourth international conference on 3D vision (3DV)*, pages 426–434. IEEE, 2016. 1

[10] Jose L Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A Iglesias-Guitian, and Antonio M López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *arXiv preprint arXiv:2312.12176*, 2023. 1

[11] 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. 1

[12] 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*, 2025. 1, 2

[13] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018. 1

[14] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. 1

[15] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 2

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 3

[17] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023. 1

[18] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 5

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[20] 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. 2

[21] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023. 1

[22] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (ToG)*, 38(6):1–15, 2019. 1

[23] 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. 6

[24] 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. 1, 3

[25] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 1

[26] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[27] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019. 2

[28] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020. 1

[29] 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. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 1

[30] 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. *arXiv preprint arXiv:2410.19115*, 2024. 1, 2, 3

[31] 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

[32] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024. 3

[33] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024. 2

[34] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 2

[35] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020. 1