Stereo Any Video: Temporally Consistent Stereo Matching
Supplementary Material

In this supplementary material, we provide additional details and more qualitative results. We highly recommend referring to the **video** we provided, since the visual quality of the disparities can be better accessed with videos.

1. Settings on Table 1.

We compare the temporal consistency of video disparity estimated by DepthCrafter [3] and RAFTStereo [8] by collecting ratings from 10 recruited participants. Each participant's rating is determined by the average of two scores: flickering region f_r and flickering strength f_s out of three possible levels: 0, 1, and 2. A higher f_r indicates a larger flickering region within the video disparity, while a higher f_s reflects a greater depth discrepancy between video frames.

2. Feature Extraction

In the feature extraction stage, when processing a video sequence with the monocular video depth model, we first resize it to ensure its dimensions are divisible by 14, maintaining consistency with the model's pretrained patch size. After obtaining the feature maps, we resize the image back to its original dimensions. Unlike previous methods [6, 8], which normalize the image directly to the range [-1,1], we apply mean and standard deviation normalization based on ImageNet pre-trained models. This ensures better alignment with the VDA framework. The monocular depth model produces feature maps with 32 channels, while the CNN encoders extract both image and context features with 96 channels each. These feature maps are concatenated to form a 128-channel representation, which serves as input to the subsequent correlation module.

3. Temporal Convex Upsampling

We develop temporal convex upsampling that extends traditional spatial upsampling techniques to the temporal dimension, enabling precise disparity interpolation across both spatial and temporal dimensions. As shown in Figure 1, our implementation leverages a convex combination approach, which ensures that the upsampled disparity maintains physical consistency while preserving intricate motion patterns. The method operates by first reshaping and normalizing mask weights through a softmax operation, creating a probability distribution across neighboring elements. Subsequently, we extract local patches from the input disparity field using a 3D unfold operation typically with a kernel size of 3, scaling the disparity vectors by the upsampling factor to maintain velocity magnitudes. The core of the algorithm lies in computing the weighted sum of these patches using the normalized mask weights, effectively performing a learned interpolation that respects the underlying motion structure. Our implementation can efficiently handle arbitrary batch sizes and integrate with existing deep learning architectures, making it suitable for video understanding tasks with high-resolution disparity fields.

```
# F[n, 2, t, h, w]      - image flow field
# M[n, c, t, h, w]      - learnable weights
# r                     - upsampling factor

# reshape and normalize mask weights into
# M[n, 1, 27, 1, r, r, t, h, w]
M = softmax(reshape(M), dim=2)

# Patch and reshape the scaled flow
# F[n, 2, 27, 1, 1, 1, t, h, w]
F = unfold_3d(r*F, kernel=[3,3,3])

# Compute the weighted (convex) combination
# F_upsampled[n, 2, 1, r, r, t, h, w]
F_upsampled = sum(M*F, dim=2)

# permute and upsample the flow
# F_upsampled[n, 2, t, r*h, r*w]
F_upsampled = reshape(permute(F_upsampled))

return F_upsampled
```

Figure 1. Pythonic pseudo-code for the implementation of temporal convex upsampling with a kernel size of $3 \times 3 \times 3$.

4. Datasets

4.1. SceneFlow (SF)

SceneFlow [9] consists of three subsets: FlyingThings3D, Driving, and Monkaa.

- FlyingThings3D is an abstract dataset featuring moving shapes against colorful backgrounds. It contains 2,250 sequences, each spanning 10 frames.
- Driving includes 16 sequences depicting driving scenarios, with each sequence containing between 300 and 800 frames.
- Monkaa comprises 48 sequences set in cartoon-like environments, with frame counts ranging from 91 to 501.

4.2. Sintel

Sintel [1] is generated from computer-animated films. It consists of 23 sequences available in both clean and final rendering passes. Each sequence contains 20 to 50 frames. We use the full sequences of Sintel for evaluation.

4.3. Spring

Spring [10] is a high-fidelity synthetic dataset rendered from the open-source animated short film *Spring* by the Blender Foundation. It features long, continuous sequences with realistic human and animal motion in natural environments. The dataset provides dense frame-wise rendering with cinematic quality, ground-truth camera parameters, depth maps, and optical flow among diverse scenes.

4.4. Dynamic Replica

Dynamic Replica [5] is designed with longer sequences and the presence of non-rigid objects such as animals and humans. The dataset includes:

- 484 training sequences, each with 300 frames.
- 20 validation sequences, each with 300 frames.
- 20 test sequences, each with 900 frames.

Following prior methods [4, 5], we use the entire training set for model training and evaluate on the first 150 frames of the test set.

4.5. Infinigen SV

Infinigen SV [4] is a synthetic dataset designed for outdoor natural environments. It consists of 226 photorealistic videos, each lasting between 3 and 20 seconds, recorded at 24 fps. The dataset is divided into:

- 186 training videos
- 10 validation videos
- 30 testing videos

Similar to Dynamic Replica, we use the full training set for training and evaluate on the first 150 frames of the test set.

4.6. Virtual KITTI2

Virtual KITTI2 [2] is a synthetic dataset that simulates outdoor driving scenarios. It consists of five sequence clones from the KITTI tracking benchmark, with variations in weather conditions (e.g., fog, rain) and camera configurations.

Since this dataset is being used for video stereo matching evaluation for the first time, we randomly select 10% of the dataset as the test set. The selected test sequences are:

- `Scene01_15-deg-left`
- `Scene02_30-deg-right`
- `Scene06_fog`
- `Scene18_morning`
- `Scene20_rain`

4.7. KITTI Depth

KITTI Depth [12] is a real-world outdoor dataset collected for autonomous driving applications. It provides sparse depth maps captured using a LiDAR sensor. Following prior work [7], we use the following test sequences:

- `2011_09_26_drive_0002_sync`
- `2011_09_26_drive_0005_sync`
- `2011_09_26_drive_0013_sync`
- `2011_09_26_drive_0020_sync`
- `2011_09_26_drive_0023_sync`
- `2011_09_26_drive_0036_sync`
- `2011_09_26_drive_0079_sync`
- `2011_09_26_drive_0095_sync`
- `2011_09_26_drive_0113_sync`
- `2011_09_28_drive_0037_sync`
- `2011_09_29_drive_0026_sync`
- `2011_09_30_drive_0016_sync`
- `2011_10_03_drive_0047_sync`

4.8. South Kensington SV

South Kensington SV [4] is a real-world stereo dataset capturing daily life scenarios for qualitative evaluation. It consists of 264 stereo videos, each lasting between 10 and 70 seconds, recorded at 1280×720 resolution and 30 fps. We conduct qualitative evaluations on this dataset.

5. Comparison Methods

We compare our approach against representative image-based stereo matching methods—RAFTStereo [11], IGEVStereo [14], and Selective-IGEV [13]—as well as video-based stereo matching methods—DynamicStereo [5] and BiDAStereo [4]. For the evaluations presented in Table 2, we use the official model checkpoints provided in the open-source implementations. For Table 3, we employ the following specific checkpoints:

- RAFTStereo: Robust Vision Challenge checkpoint
- IGEVStereo & Selective-IGEV: Middlebury fine-tuned checkpoints
- DynamicStereo & BiDAStereo: Mixed-dataset fine-tuned checkpoints from [4]

6. Application

Figure 2 demonstrates additional applications that benefit from our method's ability to produce both accurate and temporally consistent depth sequences, including adding atmospheric fog effects and the adjustment of lighting conditions. Specifically, we implement these effects by blending the input video frames with supplementary color maps, with the blending parameters determined by the estimated depth values to simulate varying transparency at the pixel level. As illustrated in the figure, the resulting frames exhibit high consistency in color without perceptible flickering, further corroborating the robust temporal consistency achieved by our method.

7. Qualitative Results on Real-world Datasets

Figure 3 gives another demo on a dynamic real world video predicted using our method. Figure 4 and Figure 5 demonstrate comparison on real world indoor scenes. Figure 6, Figure 7, and Figure 8 give more examples on real world outdoor scenes.

8. Qualitative Results on Synthetic Datasets

Figure 9 presents the comparison results on Virtual KITTI2 dataset. Figure 10, Figure 11, and Figure 12 give comparison results on Infinigen SV dataset. Figure 13 and Figure 14 show visualization comparisons on Sintel dataset, and Figure 15 shows the results on Dynamic Replica.

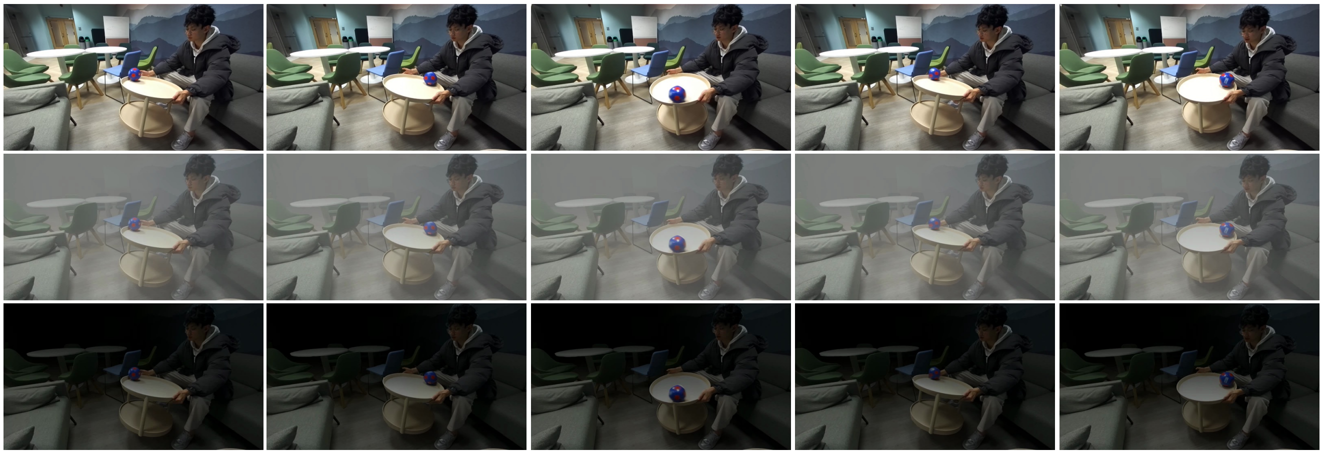

Figure 2. Examples of visual effects that could benefit from using our method, including adding fog effects and adjusting light conditions.

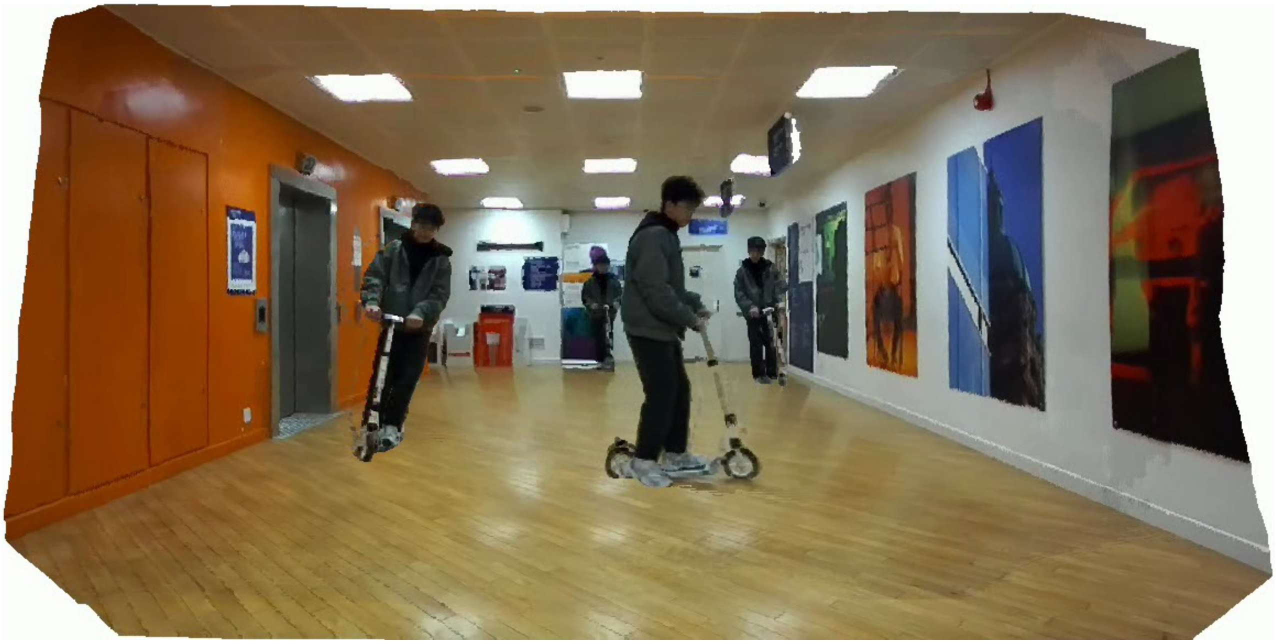

Figure 3. Another demo prediction on a dynamic real-world stereo video using our method.

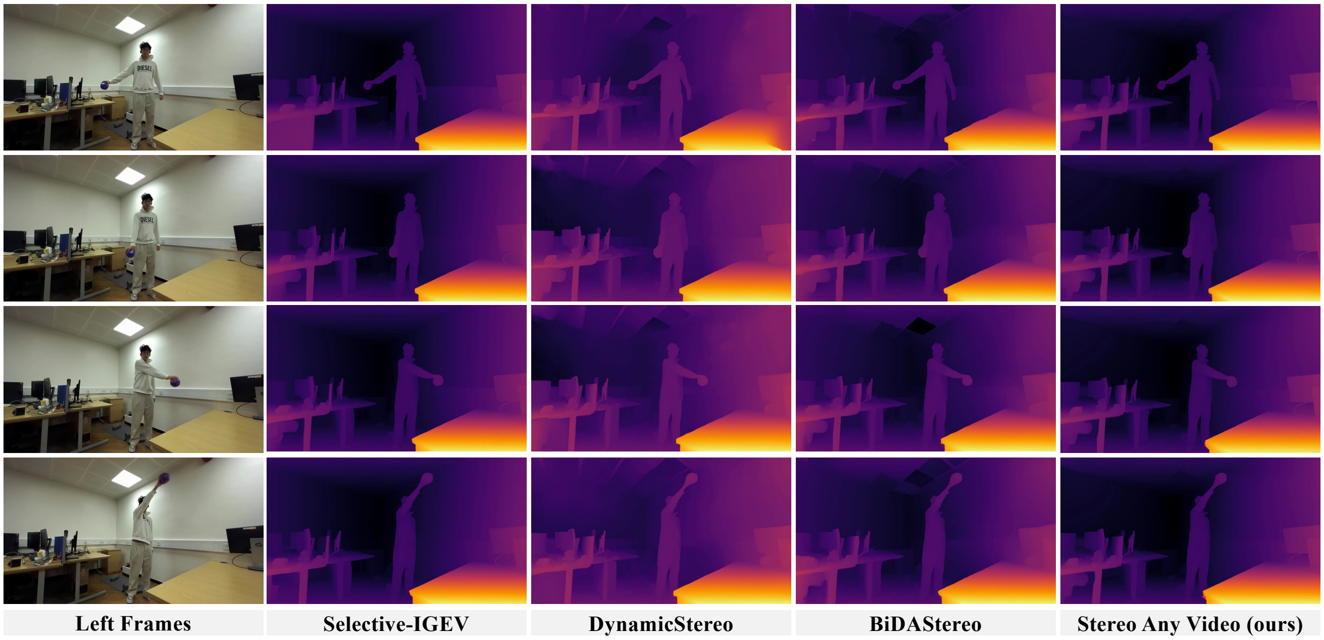

Figure 4. Qualitative comparison on a dynamic indoor scenario from the South Kensington SV dataset [4].

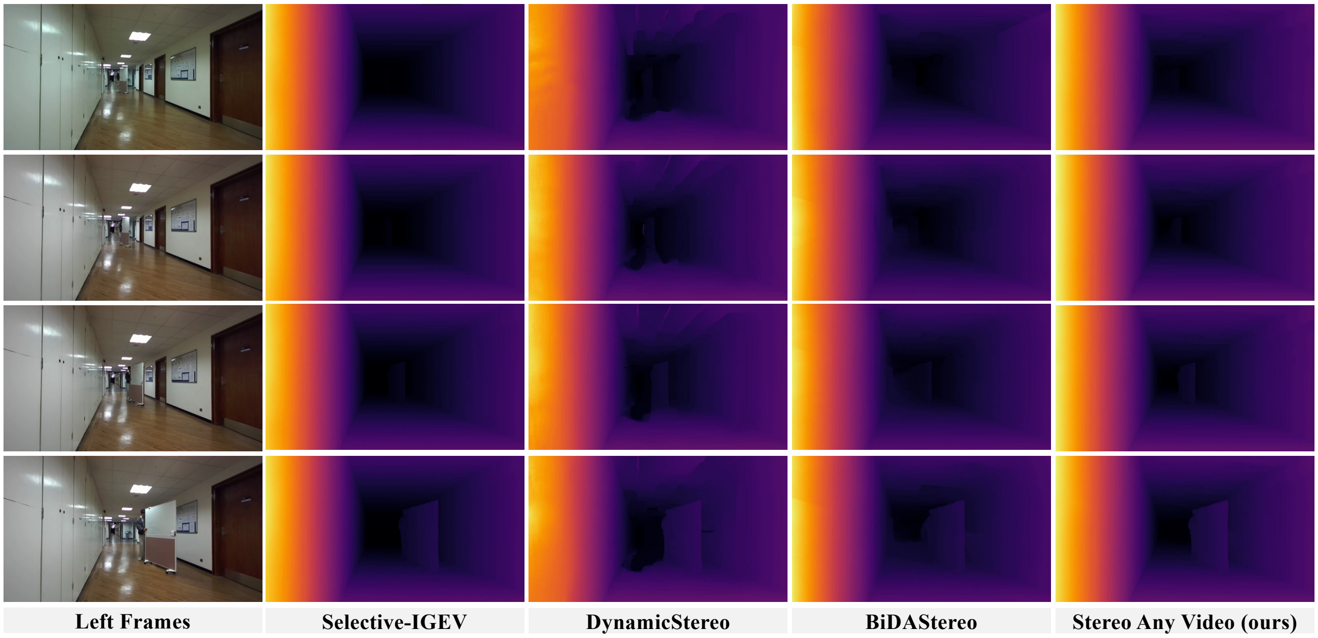

Figure 5. Qualitative comparison on a dynamic indoor scenario from the South Kensington SV dataset [4].

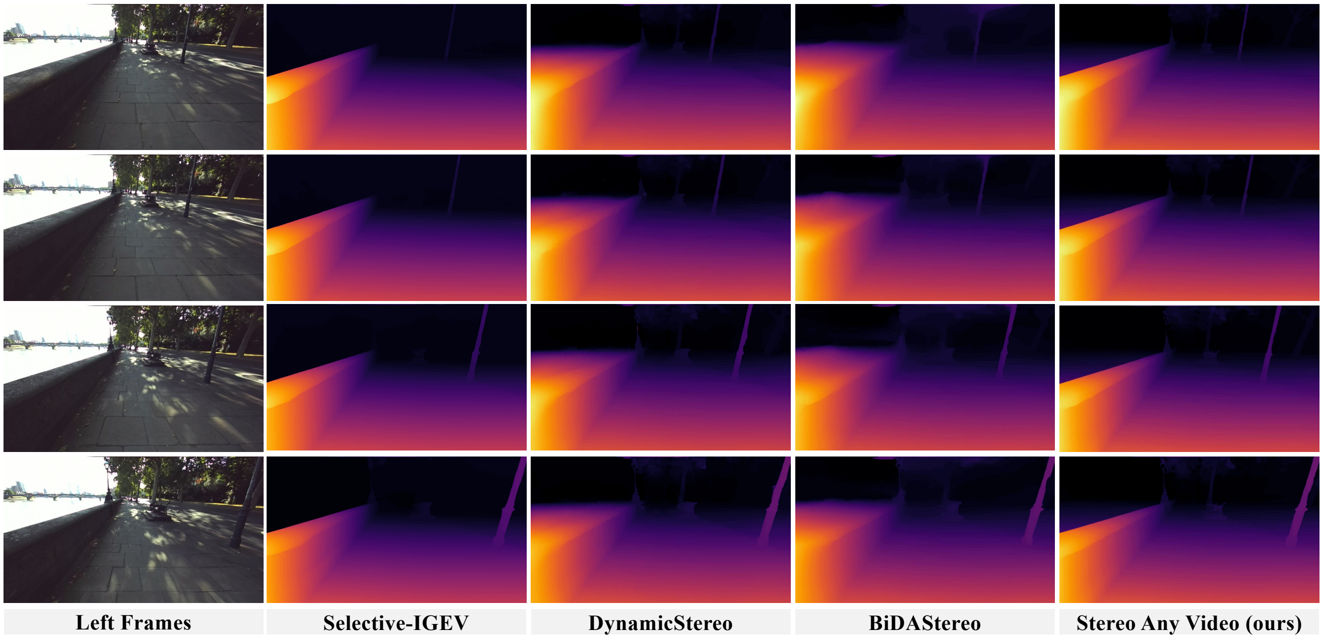

Figure 6. Qualitative comparison on a dynamic outdoor scenario from the South Kensington SV dataset [4].

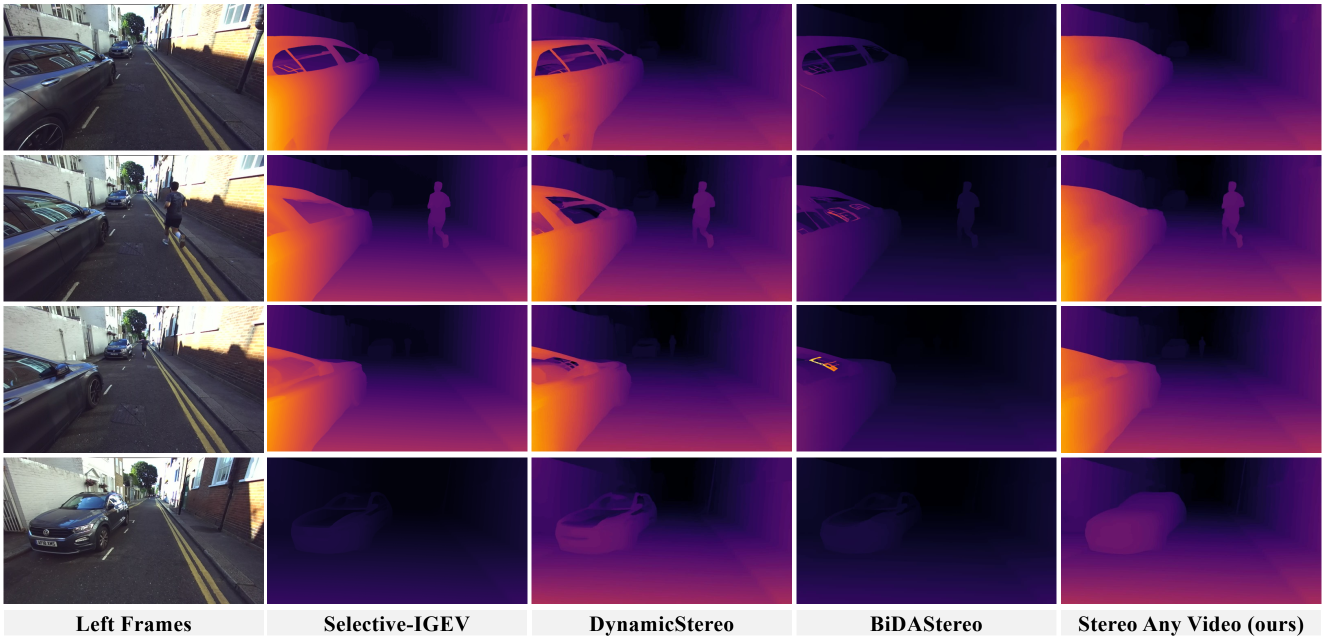

Figure 7. Qualitative comparison on a dynamic outdoor scenario from the South Kensington SV dataset [4].

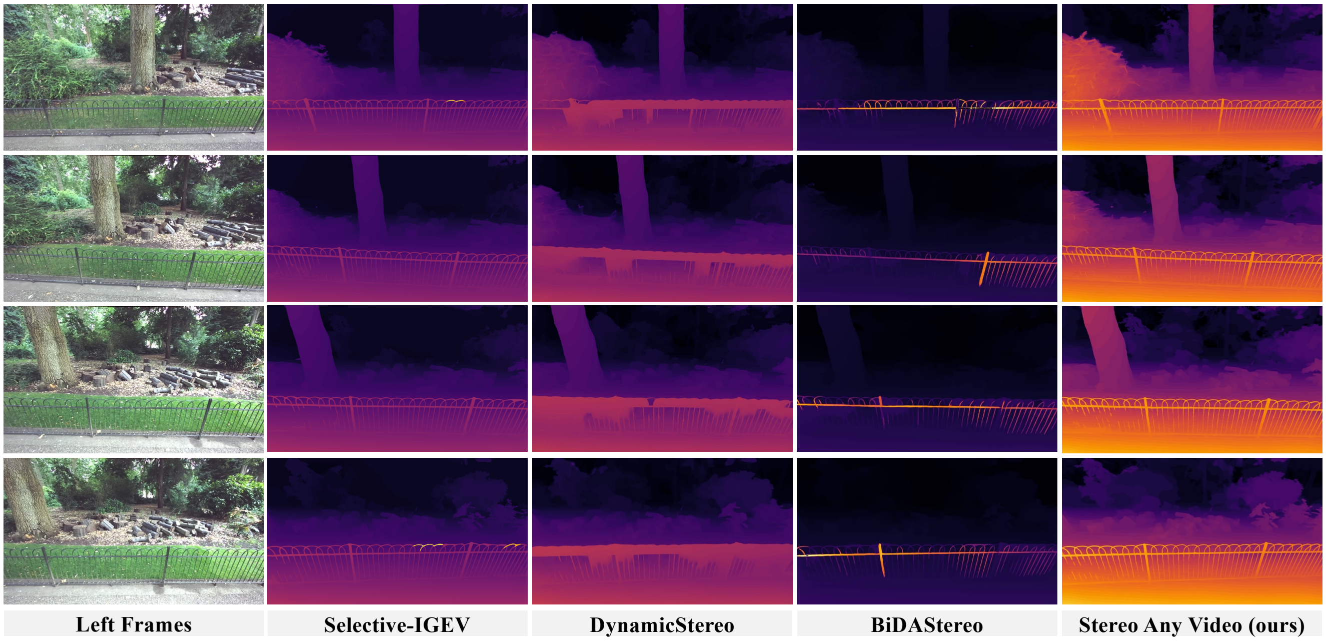

Figure 8. Qualitative comparison on a dynamic outdoor scenario from the South Kensington SV dataset [4].

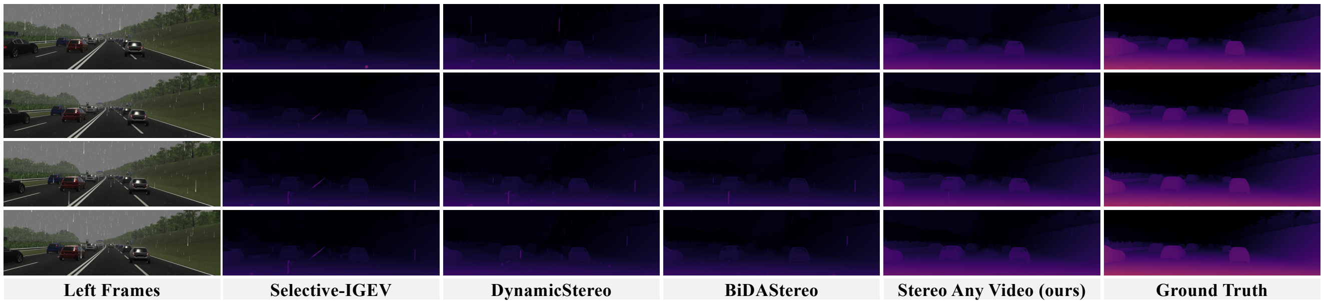

Figure 9. Qualitative comparison on Virtual KITTI2 dataset [2].

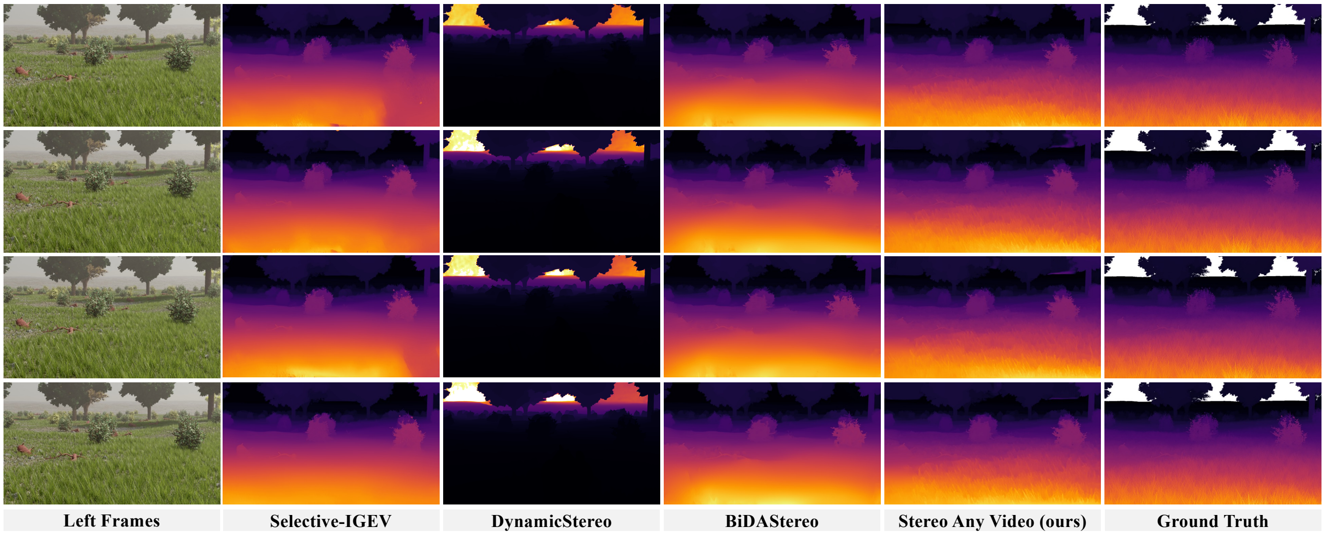

Figure 10. Qualitative comparison on Infinigen SV dataset [4].

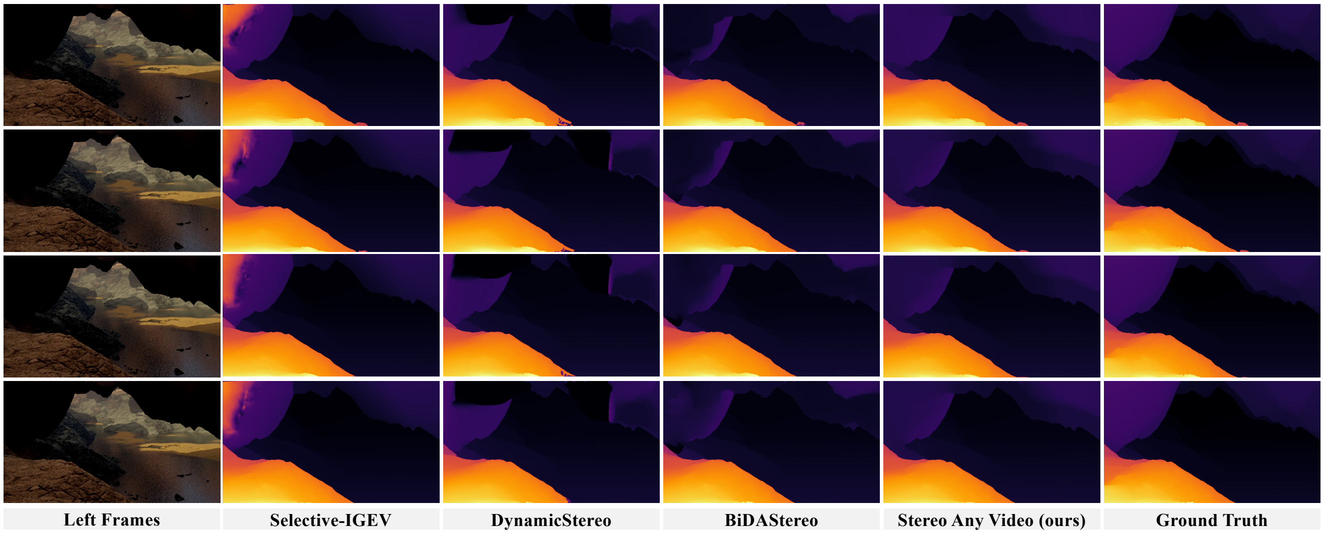

Figure 11. Qualitative comparison on Infinigen SV dataset [4].

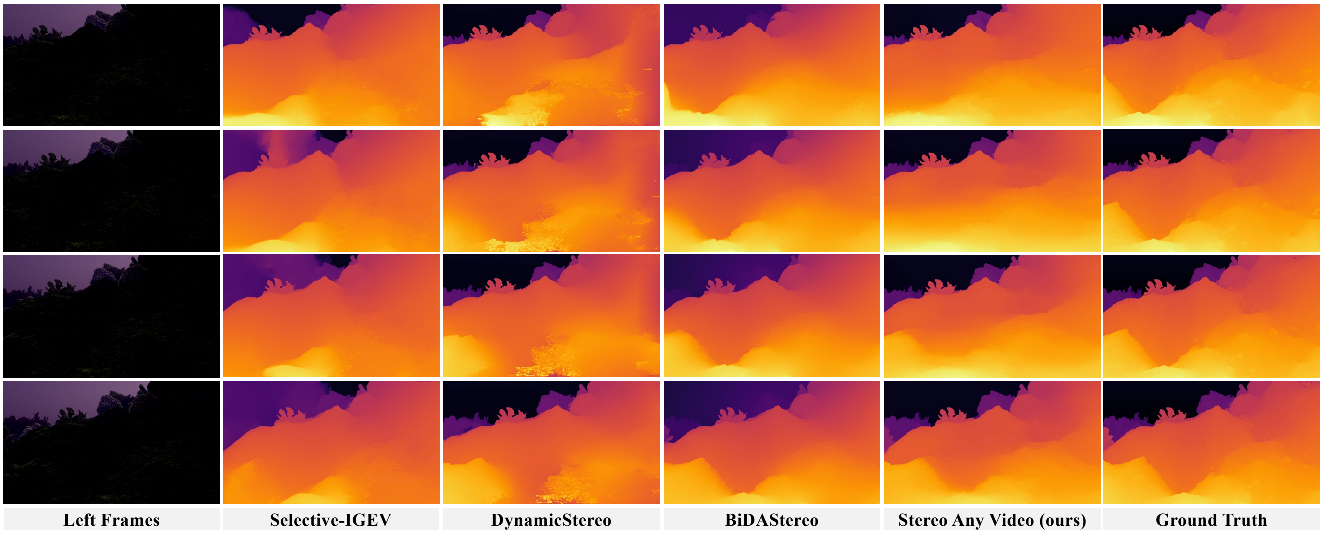

Figure 12. Qualitative comparison on Infinigen SV dataset [4].

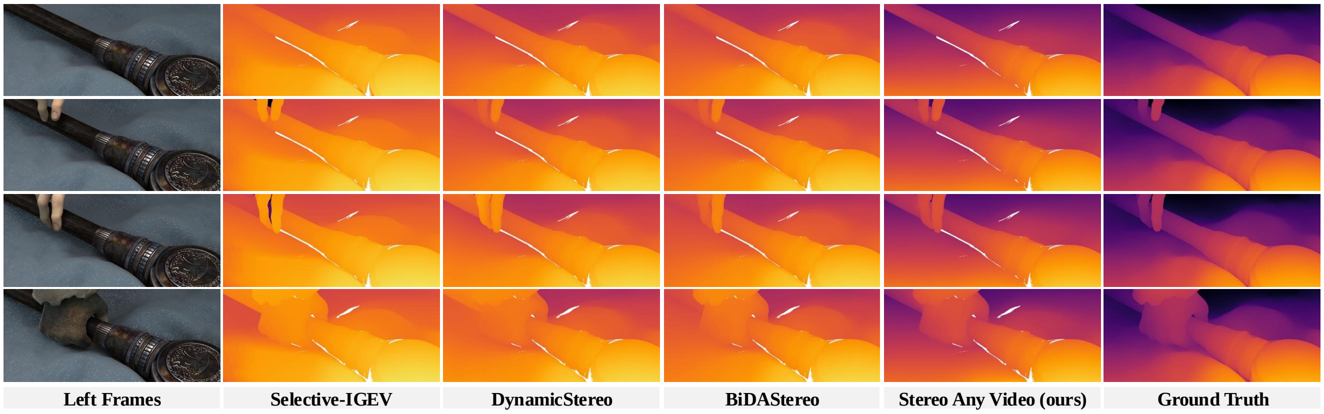

Figure 13. Qualitative comparison on Sintel dataset [1].

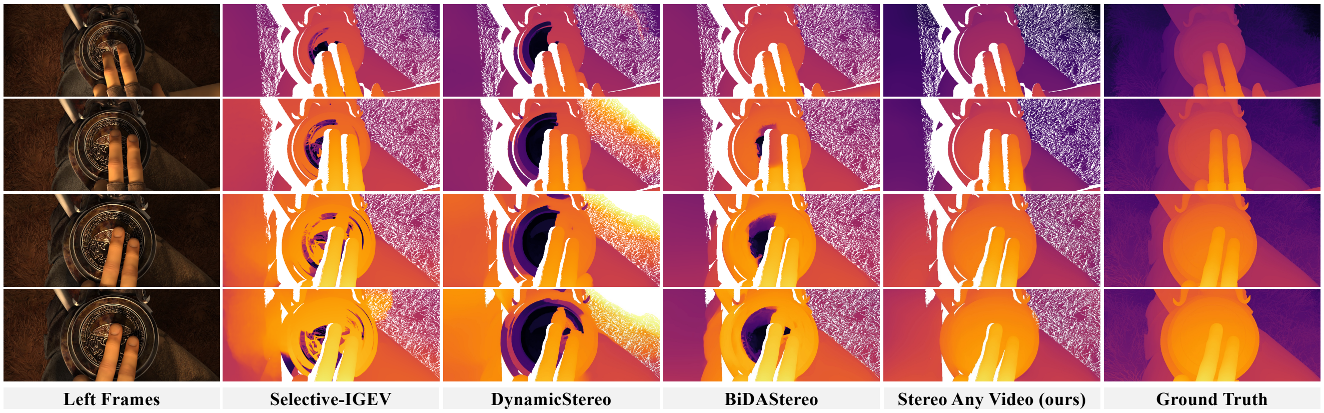

Figure 14. Qualitative comparison on Sintel dataset [1].

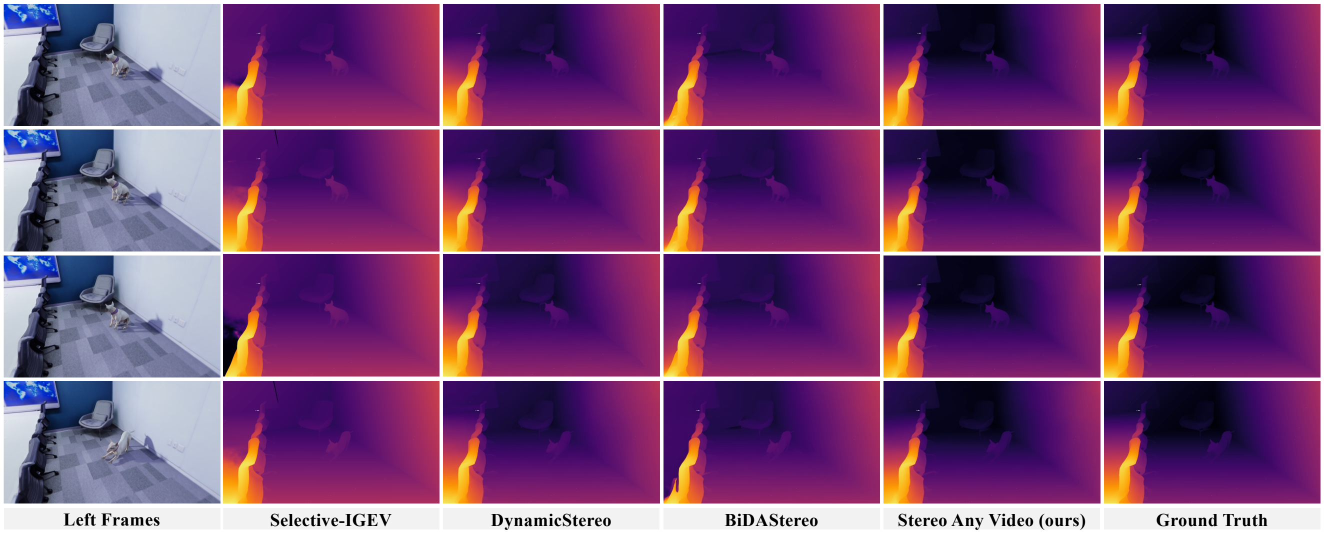

Figure 15. Qualitative comparison on Dynamic Replica dataset[5].

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625, 2012. 2, 8, 9

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 2, 7

[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] Junpeng Jing, Ye Mao, Anlan Qiu, and Krystian Mikolajczyk. Match stereo videos via bidirectional alignment, 2024. 2, 5, 6, 7, 8

[5] 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. 2, 9

[6] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022. 1

[7] Zhaoshuo Li, Wei Ye, Dilin Wang, Francis X Creighton, Russell H Taylor, Ganesh Venkatesh, and Mathias Unberath. Temporally consistent online depth estimation in dynamic scenes. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3018–3027, 2023. 2

[8] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. *arXiv preprint arXiv:2109.07547*, 2021. 1

[9] 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 *CVPR*, pages 4040–4048, 2016. 2

[10] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[11] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 2

[12] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 2

[13] Xianqi Wang, Gangwei Xu, Hao Jia, and Xin Yang. Selective-stereo: Adaptive frequency information selection for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19701–19710, 2024. 2

[14] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023. 2