DepthSync: Diffusion Guidance-Based Depth Synchronization for Scale- and Geometry-Consistent Video Depth Estimation

Supplementary Material

1. Implementation Details

1.1. Hyperparameters

We employ the Euler scheduler [8] with 5 denoising steps, as per DepthCrafter [4], we apply both guidance terms sequentially in the last two steps: geometry guidance first to refine geometry within each window, followed by scale guidance to synchronize scales across windows.

Scale guidance uses forward guidance [1], backpropagating the MSE loss gradients (Eqn. (4) in the main paper) to the noise prediction. This process iterates 1000 times in the final denoising step. The base learning rate lr (Eqn. (2) in the main paper) is set to 6e4 for the second-to-last step and 1e6 for the final step, decaying as:

$$s(t) = \mathrm{lr} \times 0.98^{\mathrm{iter}//100} \tag{1}$$

Early termination occurs if the loss falls below $5e-4$.

Geometry guidance uses backward guidance, optimizing the predicted depth decoded from the clean latent (Eqn. (1) in the main paper). We set the base learning rate to 0.01 for the second-to-last step and 0.02 for the final step, with 80 optimization iterations per step. We use the AdamW optimizer (betas = (0.9, 0.999), weight decay = 1e-4) and a cosine annealing scheduler (eta min = base lr * 0.01, T max = 80).

1.2. Loss Terms in Geometry Guidance

We employ geometric optimization based on multiple geometric constraints as the geometry guidance, specifically, comprising of global *reconstruction constraints* and local *detail constraints* to ensure geometric consistency. In each denoising step, we first obtain the current prediction of the clean latent using Eqn. (1) in the main paper. We then decode it back to the image space to obtain the predicted depth maps $\mathbf{d}_i$ on input video frames $\mathbf{I}_i$, which serve as inputs for the guidance loss computation.

1.2.1. Global Geometric Constraints constraints

A monocular video, along with its corresponding depth maps per frame, provides a comprehensive representation of the scene geometry. Specifically, each frame can generate a segment of the scene's point cloud when projected with its depth. Different frames should form aligned projections, establishing an inherent constraint between depths. Aligning these projections requires access to the camera pose of each frame. To determine the camera poses, we employ an off-the-shelf tracking prediction network [7] to generate pairwise dense tracking pixel correspondences between adjacent frames and estimate the camera poses $\mathbf{P}_i$ of each frame using Perspective-n-Point (PnP) [2]. We employ the solvePnPRansac inferface from opencv-python library for PnP computation. With the determined poses, we apply depth reprojection loss and tracking loss to guarantee global consistency, ensuring that the 3D structure formed with the video depths within sliding window represents a geometrically consistent scene.

Depth Reprojection Loss. Given a frame $\mathbf{I}_i$, camera intrinsics $\mathbf{K}$ and relative pose transformation $\mathbf{T}_{i\to j}$ to another frame $\mathbf{I}_j$, we project pixels in $\mathbf{I}_i$, denoted as p_i, to 3D points, transform them to the coordinates of j and project to $\mathbf{I}_j$ to get the corresponding pixels p_j, with a pixel correspondence between two frames as:

$$p_j \sim \mathbf{K}\mathbf{T}_{i\to j}\mathbf{d}_i(p_i)\mathbf{K}^{-1}p_i, \tag{2}$$

Consequently, depth map $\mathbf{d}_j$ can be warped $\mathbf{d}_{j\to i}$ under this pixel correspondence relationship, and its difference to $\mathbf{d}_i$ forms the depth reprojection loss $\mathcal{L}_r$, which is computed as an L1 difference between the two depth maps:

$$\mathcal{L}_d^{i,j} = |\mathbf{d}_i - \mathbf{d}_{j\to i}| \tag{3}$$

The depth reprojection loss is computed and averaged across all pairs within frame distance of 3 in a sliding window. As for the case where camera intrinsic matrix K is unknown, we can approximate it using image height (h) and width (w), with focal length as $(w+h)/2$ and principal point as $(w/2, h/2)$ as common practice.

Tracking Loss. Tracking correspondences between $\mathbf{I}_i$ and frame $\mathbf{I}_j$ allow us to project pixels to 3D using corresponding depth predictions, obtaining points P_i and P_j. Transforming P_j to coordinate i using derived camera poses from PnP, the L2 difference to P_i forms a tracking loss:

$$\mathcal{L}_t^{i,j} = ||P_i - P_{j\to i}||_2 \tag{4}$$

The tracking loss is computed for adjacent pairs only for simplicity.

1.2.2. Local Geometric Constraints

Surface Normal Loss. To improve the local structure of the generated depth, we employ an offline surface normal generation network [13] to pre-compute surface normals $\mathbf{n}_i^{pre}$ for each video frame. We compute the surface normal $\mathbf{n}_i$ from the predicted depth map and align it to the direction of the pre-computation result with surface normal angular loss:

$$\mathcal{L}_n^i = 1 - \cos(\mathbf{n}_i^{pre}, \mathbf{n}_i) \tag{5}$$

Smoothness Loss. We also use edge-aware smoothness loss $\mathcal{L}_s$ to encourage local smoothness by compute an L1 penalty on depth gradients with image gradients as weight [3]:

$$\mathcal{L}_s^i = |\partial_x \mathbf{d}_i| e^{-|\partial_x \mathbf{I}_i|} + |\partial_y \mathbf{d}_i| e^{-|\partial_y \mathbf{I}_i|} \tag{6}$$

The detail constraints are computed from each frame respectively and averaged over the batch.

1.2.3. Overall Loss

The overall guidance function comprises a weighted sum of the multiple loss terms above:

$$\mathcal{L} = \alpha_d \mathcal{L}_d + \alpha_t \mathcal{L}_t + \alpha_n \mathcal{L}_n + \alpha_s \mathcal{L}_s$$

We assign the largest loss weight to the depth reprojection loss term as it plays the major role in aligning the geometry between frames. We set $\alpha_d = 35, \alpha_n = 0.1, \alpha_s = 1, \alpha_t = 2$ in all the experiments in the paper.

2. Comparison with Reconstruction Methods

We compare our method with representative reconstruction methods on the first 10 ScanNet test scenes. COLMAP[9] and 2DGS [5] (uses COLMAP poses) fail on 3 scenes due to failed COLMAP pose estimation, while our method reconstructs all scenes successfully. For shared scenes, our approach achieves more accurate depth and poses (Table 1).

Method	COLMAP [9]	2DGS[6]	**DepthSync**
# Failure Cases	3	3	**0**
AbsRel↓	0.464	0.136	**0.098**
$\delta < 1.25$ ↑	0.479	0.823	**0.910**
ATE↓	0.214		**0.088**
RPE t↓	0.0914		**0.0198**
RPE r↓	6.65		**0.611**

Table 1. **Comparison with Reconstruction Methods.** Our method demonstrates better geometric accuracy as well as better robustness compared to conventional reconstruction methods.

3. Inference Cost and Supplementary Evaluation

We report time and peak GPU memory usage during inference in Table 2. To balance performance, we provide a lightweight variant (denoted as Ours-S in Tables 2, 3): omit geometry guidance and apply scale guidance only at the penultimate denoising step. We further supplement a comprehensive evaluation of this lightweight version with current latest depth estimation methods in Table 3. Our system offers a favorable trade-off between accuracy and cost, supporting both efficient online alignment and more accurate but costlier offline optimization.

Method	Latency (s)	Max Memory (MiB)
DepthCrafter [4]	9.01	13449
DepthAnyVideo [12]	14.6	24097
DUSt3R [11]	430	9207
MonST3R [14]	572	30559
Post Opt.	652	14191
Ours:		
Scale Guidance Only	33.3	18223
Geometry Guidance Only	952	26833
Ours	961	26833
Ours-S	18.1	18223

Table 2. **Inference Cost Evaluation** for 90 frames (Resolution: 640× 448) on a 40GB A100 GPU.

Cost Comparison with Post Optimization. Our full system has significantly better results than post optimization with comparable cost, and our scale-only version is both faster (Table 2) and superior (main paper Table 3) than post optimization.

Comparison with DUSt3R [11] and MonST3R [14] (3Rs). We follow the depth estimation paradigm, taking a video as input and predicting video depths, while 3Rs use image pairs to predict point maps, adopting a fundamentally different technical route. For comparison, we use sliding window strategy and align windows via overlap area to enable long video inference. We use 3Rs to adjacent frames in each window and derive depths from predicted point maps. Results are shown in Table 3 (DUSt3R fails on 2 of 26 Bonn scenes, so it's averaged over 24 scenes; others use all 26). While 3Rs perform better on indoor scenes, likely due to training on ScanNet++, our method achieves better results on KITTI and Bonn.

4. Supplementary Ablation Study

We present supplementary ablation studies on the guidance function, including the starting step, guidance order, and geometry guidance loss terms. These studies are conducted on the first 25 scenes of the ScanNet test set, with video length as 150, which is consistent with the ablation settings in the main paper.

Starting Step of Guidance. We conduct experiments on when to start guidance. As shown in Table 4, starting guidance early does not always improve performance, as noise in the predicted clean latent during early denoising stages can mislead the guidance process. Starting from the third-to-last step yields the highest depth accuracy but the same relative error with second-to-last and adds 50% inference latency. Thus, starting from the second-to-last step strikes the best balance between efficiency and effectiveness.

Ablation on Guidance Term Order. As shown in Table 4, applying the geometry guidance first and then the scale guidance leads to better depth estimation results, in-

	Dataset	ScanNet		GMU Kitchen		KITTI		Bonn	
	Metrics	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
①	ChronoDepth [10]	0.172	0.749	0.196	0.650	0.178	0.733	0.092	0.932
	DepthCrafter [4]	0.141	0.799	0.143	0.795	0.114	0.879	0.095	0.917
	DUSt3R [11]†	**0.059**	**0.972**	**0.069**	**0.968**	0.155	0.777	0.075	0.935
	MonST3R [14]†	0.067	0.959	0.087	0.920	0.192	0.692	**0.046**	**0.975**
	Ours-S	0.128	0.834	0.126	0.844	0.112	0.882	0.070	0.968
	Ours	0.113	0.870	0.113	0.881	**0.110**	**0.887**	0.069	0.972
②	ChronoDepth [10]	0.193	0.699	0.225	0.612	0.193	0.691	0.132	0.853
	DepthCrafter [4]	0.171	0.716	0.160	0.748	0.173	0.727	0.143	0.801
	DUSt3R [11]†	**0.090**	**0.934**	0.127	0.828	0.174	0.731	0.115	0.889
	MonST3R [14]†	0.099	0.911	0.140	0.817	0.232	0.620	**0.086**	0.918
	Ours-S	0.157	0.757	0.138	0.807	0.119	0.863	0.094	0.926
	Ours	0.154	0.769	0.131	0.837	**0.117**	**0.869**	0.093	**0.929**

Table 3. **Supplementary Depth Evaluation**. Ours-S: a lightweight version of our method. ① and ②: the shortest and longest video length settings in our main paper. *: trained on Hypersim. †: trained on ScanNet++.

Step	AbsRel	$\delta < 1.25$
5	0.146	0.779
4	0.144	0.787
3	0.137	**0.803**
2	**0.137**	0.801
1	0.151	0.757
Order	**AbsRel**	**$\delta < 1.25$**
Scale First	0.145	0.782
Geometry First	**0.137**	0.801

Table 4. **Supplementary Ablation Study on Guidance Implementations.** The ”Step” column indicates the number of final steps in the diffusion loop at which the guidance is employed. For example, if the ”Step” value is 5, it means that the guidance starts from the last five steps.

dicating that a geometric aligned intra-window depth help a better synchronization of depth scale between windows.

Ablation on Each Loss Term in Geometry Guidance. We present ablation results on the effectiveness of each loss term in geometry guidance in Table 5. Applying global geometric constraints (depth reprojection loss and tracking loss) individually yield greater improvements than local constraints (surface normal loss and smoothness loss). Depth reprojection loss outperforms tracking loss, as it operates on every pixel rather than sparse tracked points. Combining local constraints with global constraints further enhances the performance, as global constraints alone just align the depths between frames but may not converge to the optimal solution, while local constraints introduces supplementary information about the geometric structure to the optimization process and help avoid local optima.

Constraints	Metrics	
	AbsRel	$\delta < 1.25$
Baseline 90	0.136	0.816
loss d only	0.122	0.856
loss n only	0.121	0.851
loss s only	0.122	0.847
loss t only	0.121	0.850
all loss	**0.108**	**0.880**

Table 5. **Ablation Study on Guidance Terms** on ScanNet first 25 Test Scenes.

5. Supplementary Qualitative Examples

We provide additional qualitative examples to supplement the main paper. Figure 1, Figure 2, Figure 3, Figure 4 and Figure 5 demonstrate that our DepthSync achieves better scale synchronization between windows for long video and more accurate geometric consistency within each window.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023. 1

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[3] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for

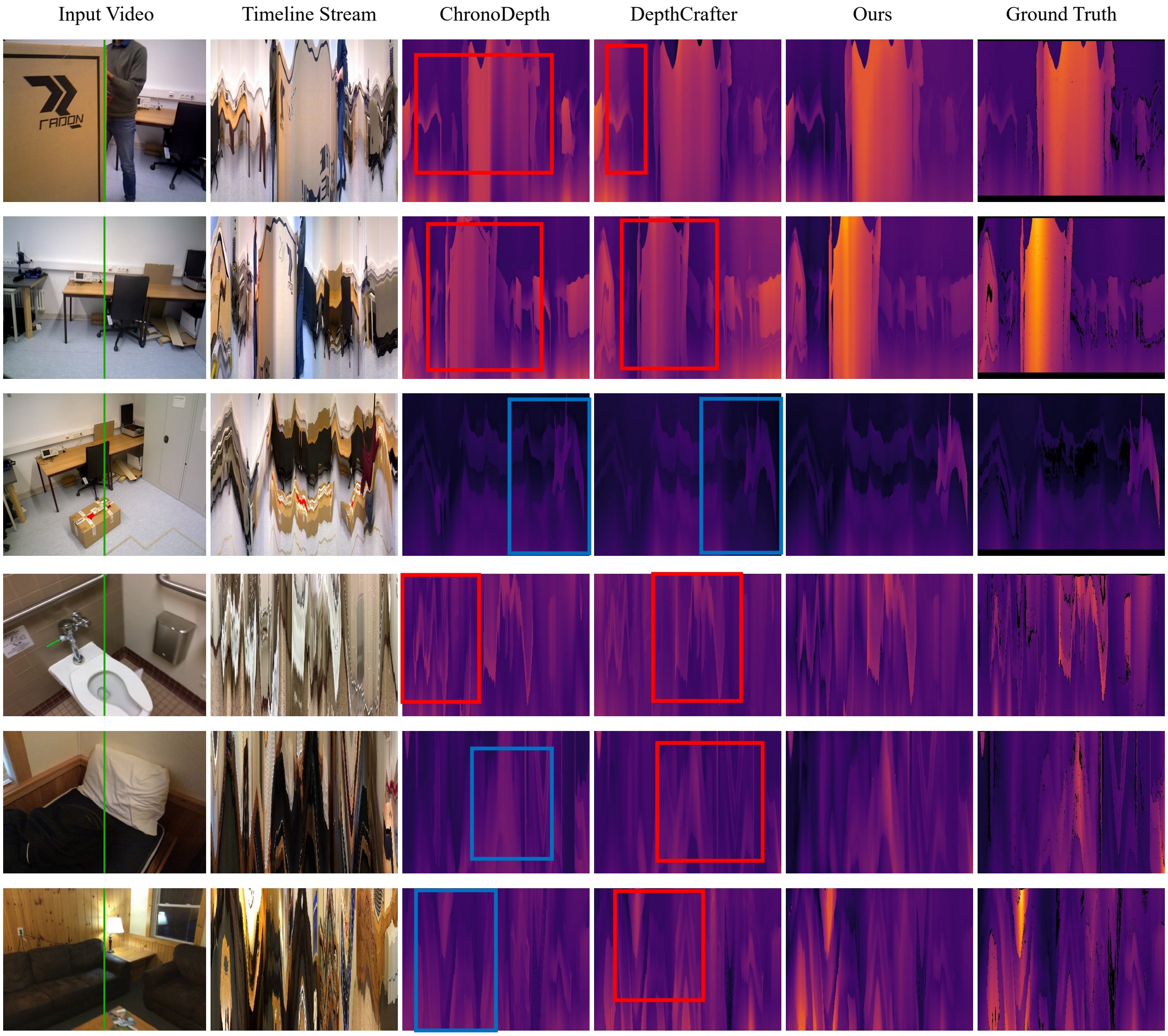

Figure 1. **Supplementary Qualitative Comparison on Long Video Depth Consistency.** We compare predictions on 450-frame ScanNet videos and 590-frame Bonn videos. Slicing along the timeline at the green-line position (first column), we concatenate results to visualize temporal changes in color and depth. Red boxes highlight depth scale inconsistencies, while blue boxes mark depth inaccuracies in previous methods. Our method shows superior depth accuracy and scale consistency over time.

stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013. 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 preprint arXiv:2409.02095*, 2024. 1, 2, 3

[5] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2

[6] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[7] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 1

[8] 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. 1

[9] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[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. 3

[11] 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. 2, 3

[12] 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. 2

[13] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *arXiv preprint arXiv:2406.16864*, 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. 2, 3

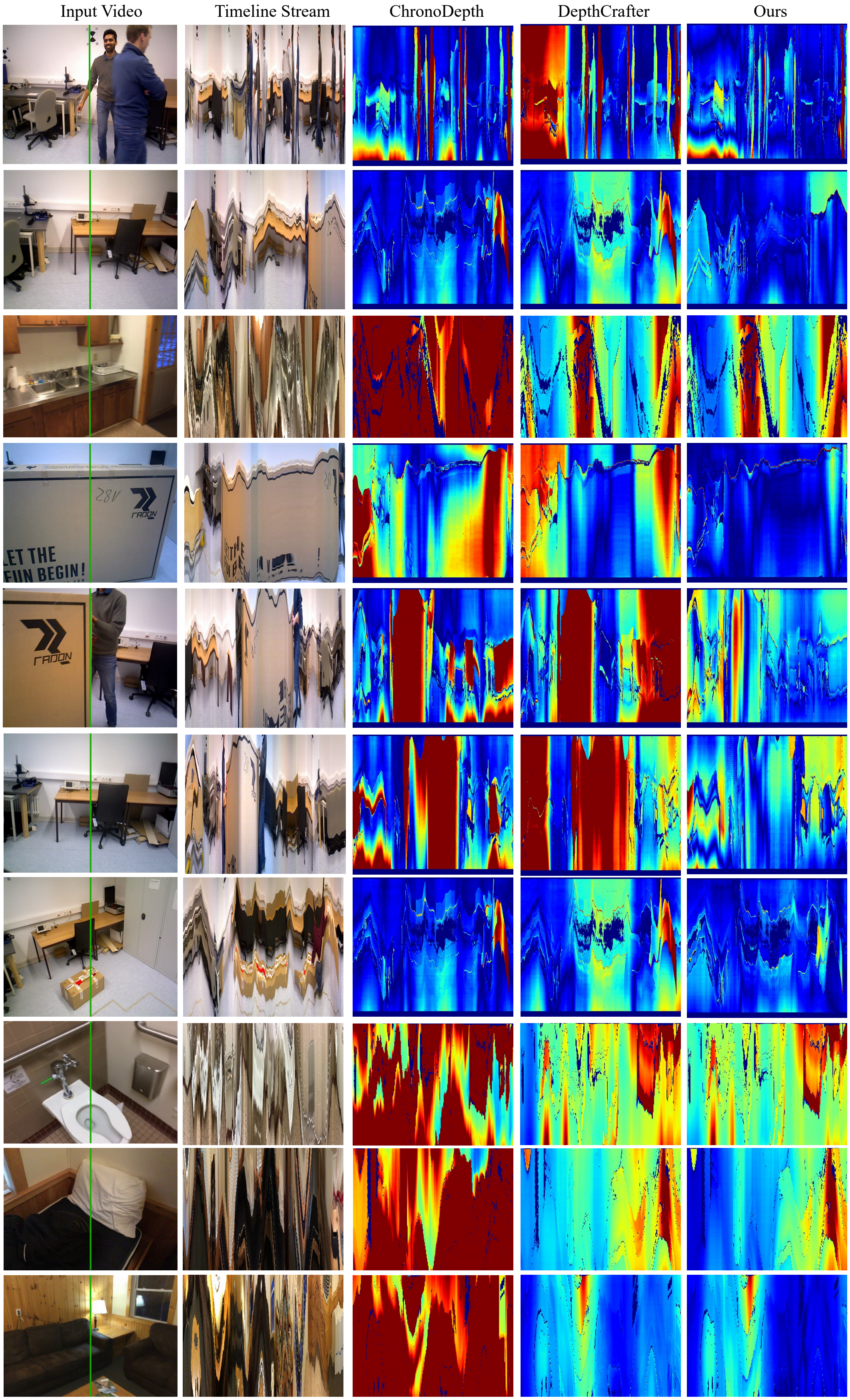

Figure 2. **Depth Error Map Comparison on Long Video Depth Consistency.** Supplementary visualization of absolute error maps for qualitative analysis (see Figure 4 in the main paper and Figure 1). Blue indicates low error, while red corresponds to high error.

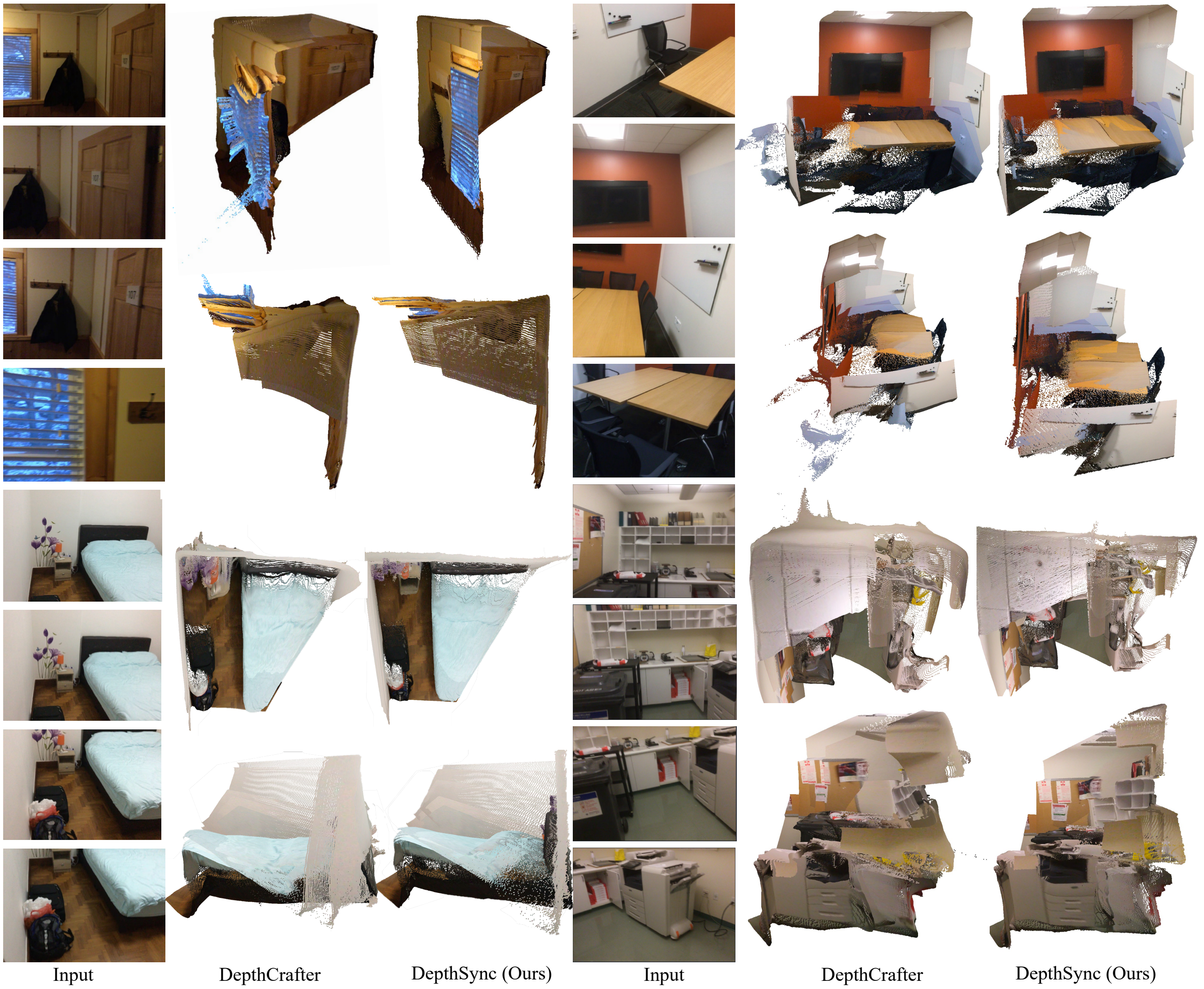

Figure 3. **Supplementary Qualitative Examples for 3D Reconstruction Comparisons**.

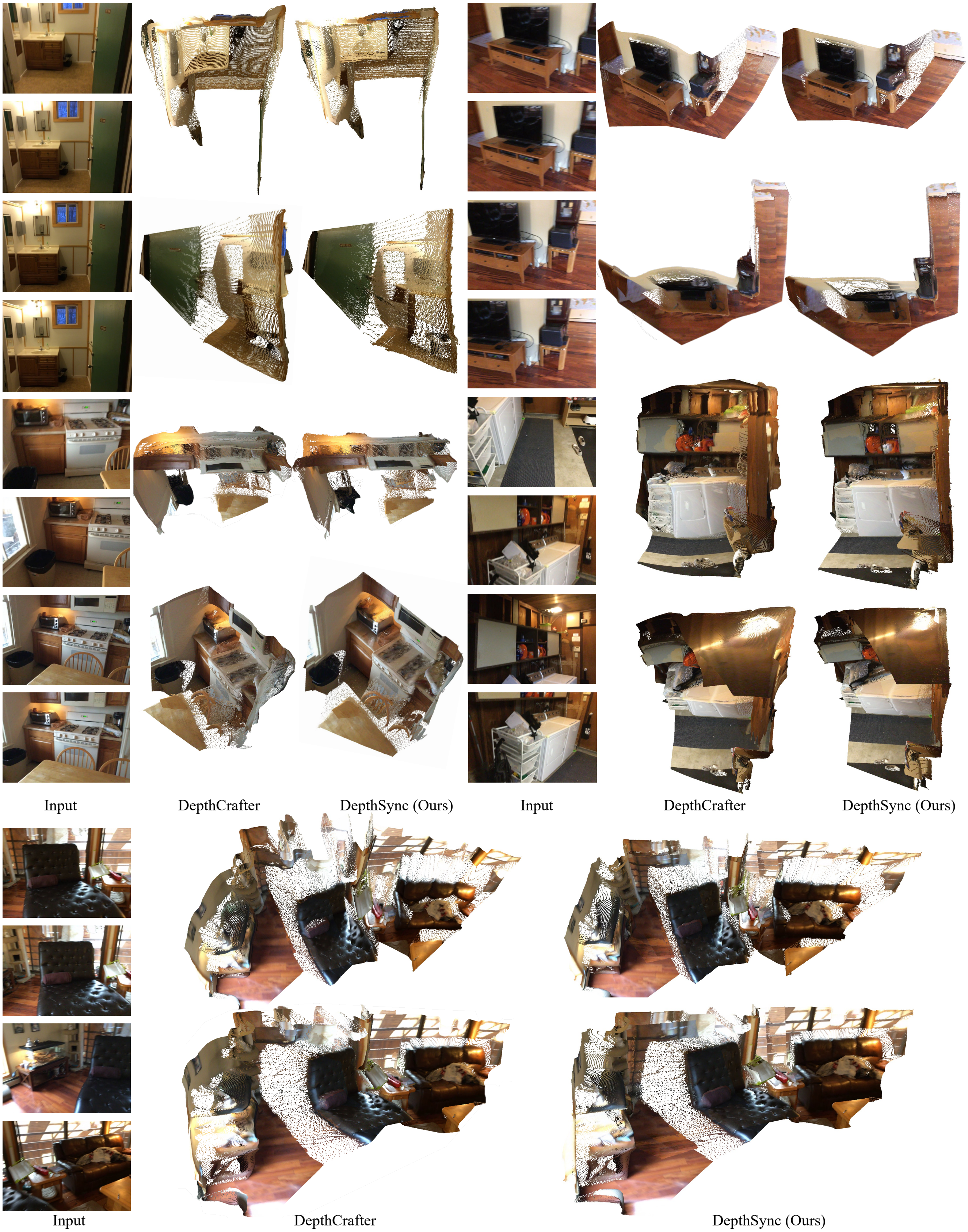

Figure 4. **Supplementary Qualitative Examples for 3D Reconstruction Comparisons**.

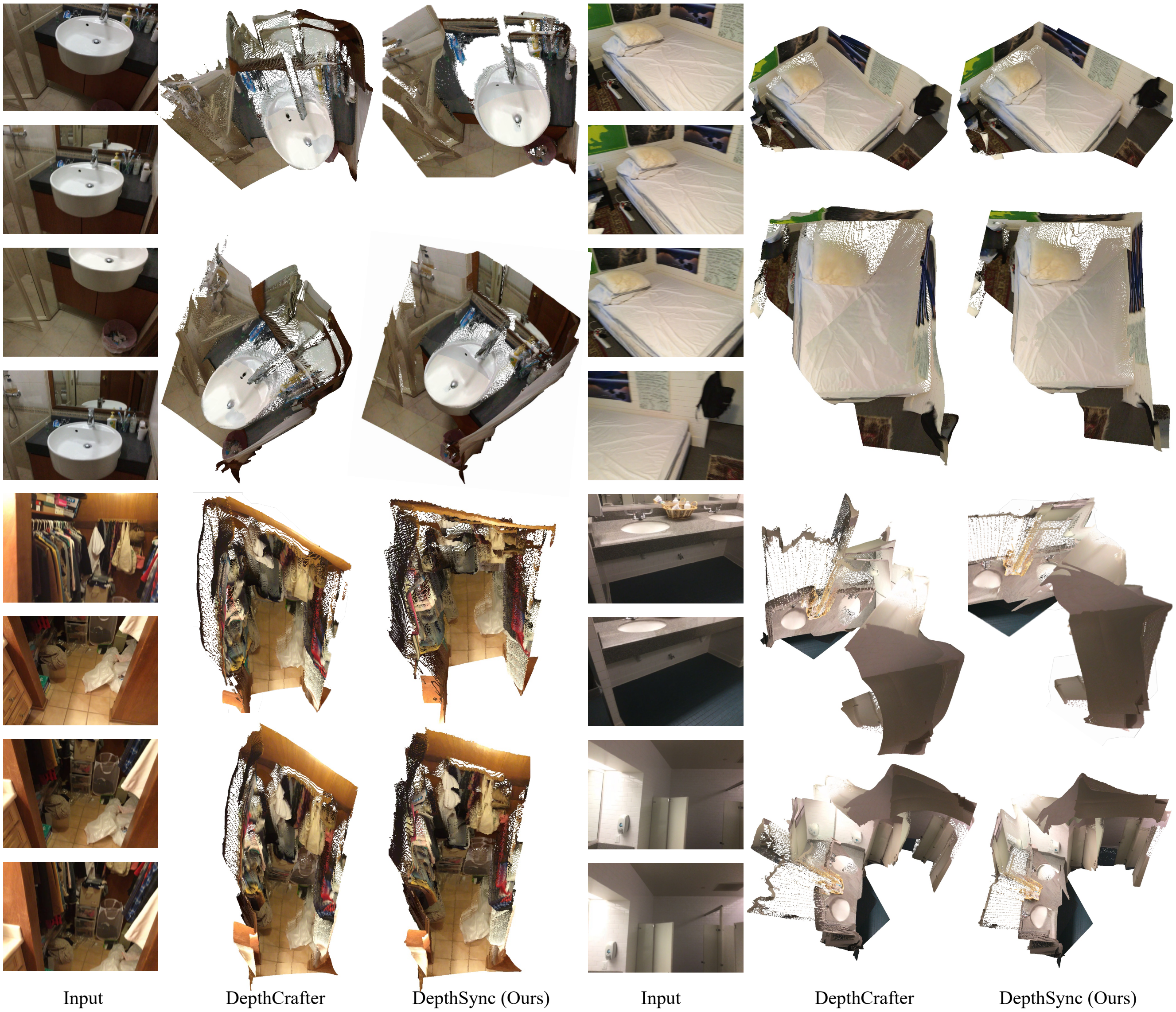

Figure 5. **Supplementary Qualitative Examples for 3D Reconstruction Comparisons**.