Real-Time Sphere Sweeping Stereo from Multiview Fisheye Images Supplemental Document

Andreas Meuleman Hyeonjoong Jang Daniel S. Jeon Min H. Kim

KAIST

1. Inter-scale Bilateral Upsampling Formulation

In addition to Section 3.2.1 in the main paper, we describe in more details the upsampling process. First, we define the inter-scale bilateral weights as follows:

$$w^{\uparrow}_{x',y'}(I, I', x, y) = \exp\left(\frac{\|I(x,y) - I'(x',y')\|^2}{2\sigma_I^2}\right), \quad (1)$$

where σ_I is the edge preservation parameter as mentioned in the main paper. We use those weights to smooth the higher resolution image in an edge aware manner:

$$\tilde{I}_l(x,y) = \left(1 - w^{\uparrow}_l\right) I_l(x,y) + w^{\uparrow}_l \sum_{(x',y') \in \mathcal{N}^{\uparrow}_{x,y}} w^{\uparrow}_{x',y'} \left(\tilde{I}_{l+1}, I_l, x, y\right), \quad (2)$$

where $w^{\uparrow}_l$ are the scale blending weights defined in Section 3.2.1 in the main paper. $\mathcal{N}^{\uparrow}_{x,y}$ are the direct neighbor pixels of $(x/2, y/2)$. Note that there can be four, six or nine direct neighbors depending on the sampling location, as Figure 4 in the main paper shows. Our algorithm outputs the smoothed image $\tilde{I}_0$.

2. Camera Selection with a Different Camera Rig

Figure 1 shows the selected camera with a perfect planar rig. As expected, this layouts leads to a vertically symmetrical camera selection, as opposed to our real setup in Figure 3 of the main paper.

3. Filter Kernel and Smoothness

Figure 5 in the main paper shows the impact of the edge preservation parameter σ_I. Figure 2 shows additional filtering examples with different smoothness parameters σ_s. As expected, higher smoothness propagates more aggressively.

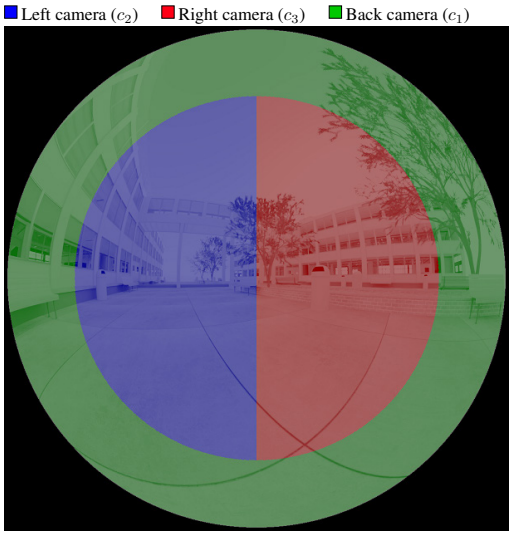

Figure 1: Selected camera for each pixel in c_0 with a perfect planar rig as our test dataset.

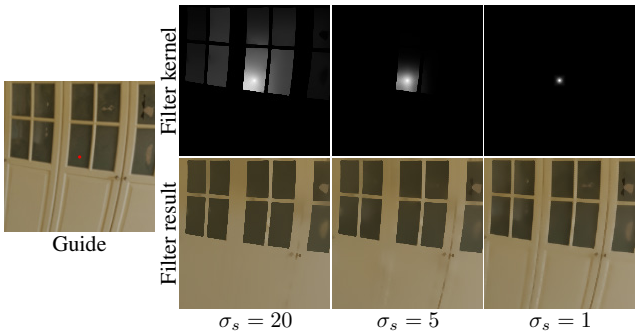

Figure 2: Filter shape and results with different smoothness parameters.

4. Additional Comparisons

4.1. Impact of directional inpainting

When replacing the naïve diffusion kernel [3] by the directed version, the >0.1 bad pixel ratio and RMSE of the inpainted areas fall from 23.98 % and 0.123 m^{-1} to 14.03 % and 0.089 m^{-1}, showing that our directional inpainting is

significantly more reliable. This method brings the error in the inpainted version comparable to error in the overall distance map (12.51 % and 0.079 m^{-1}).

4.2. Number of distance candidates

Table 1 shows that, thanks to a stable cost volume and sub-candidate interpolation, only a small number of candidates allows for good accuracy on our dataset.

#candidates	8	16	24	32	40	48
>0.1 bad pixel ratio (%)	17.90	12.75	12.52	12.51	12.49	**12.47**
1/distance RMSE (m^{-1})	0.093	0.079	0.080	**0.079**	0.080	0.080
Runtime on Xavier board (ms)	**15**	21	28	34	40	46

Table 1: Impact of the number of distance candidates.

4.3. Robustness

To evaluate the depth estimation's robustness, we insert Gaussian noise of standard deviation 0.02 to our synthetic dataset. Table 2 shows that our method is barely affected while some other are. In particular, SweepNet [5] and Lin et al. [2], due to their weaker cost aggregation, are highly impacted by non-ideal capture conditions. Figure 7 shows the results on an example scene from our dataset, and can be compared with non noisy inputs in Figure 8.

4.4. Real Camera Layouts

In the main paper, we ran comparisons with a planar camera rig to follow the configuration of the other works and allow for fair comparison. We additionally rendered our dataset with our real camera rig positions. Table 3 shows that a vertical and horizontal camera rig seems to grant noticeable improvement to all other methods. It it worth noting that CrownConv [1] improved significantly more than OmniMVS [4]. This can be explained by its greater capabilities to adapt to different camera rigs.

4.5. Qualitative Comparison

Figures 3, 4, 5, 6 show the real scenes from the main paper in full resolution with comparison against Other depth from fisheye images methods. We run SweepNet and OmniMVS at 1024×512 px due to memory requirements.

4.6. Real-Time Prototype Demo

Refer to the supplemental video for real-time prototype demo.

	Inverse distance (% | m^{-1})				Panorama		Runtime
	>0.1	>0.4	MAE	RMSE	PSNR	SSIM	(ms)
CrownConv* [1]	57.34	4.45	0.149	0.194	33.77	0.953	$5.2{\cdot}10^{2}$
OmniMVS† [4]	43.31	7.83	0.144	0.219	34.60	0.955	$1.3{\cdot}10^{3}$
Sweepnet‡ [5]	53.15	6.80	0.161	0.255	32.94	0.948	$1.0{\cdot}10^{5}$
Lin et al.† [2]	39.96	12.36	0.168	0.277	35.21	0.963	$2.5{\cdot}10^{3}$
Ours	**22.66**	**0.69**	**0.073**	**0.102**	**37.06**	**0.979**	$\mathbf{2.8{\cdot}10^{0}}$

Table 2: Comparison of our spherical RGB-D results against other methods on our rendered dataset with additive Gaussian noise. All methods are run on our desktop test system and output a $1024{\times}512$ px distance map.

	Inverse distance (% | m^{-1})				Panorama		Runtime
	>0.1	>0.4	MAE	RMSE	PSNR	SSIM	(ms)
CrownConv* [1]	27.11	1.98	0.086	0.121	37.31	0.988	$5.2{\cdot}10^{2}$
OmniMVS† [4]	28.72	4.83	0.101	0.165	36.99	0.986	$1.3{\cdot}10^{3}$
Sweepnet‡ [5]	29.61	2.85	0.088	0.125	35.20	0.980	$1.0{\cdot}10^{5}$
Lin et al.† [2]	33.44	5.69	0.130	0.304	36.05	0.980	$2.5{\cdot}10^{3}$
Ours	**14.35**	**0.25**	**0.052**	**0.076**	**37.75**	**0.989**	$\mathbf{2.8{\cdot}10^{0}}$

Table 3: Comparison of our spherical RGB-D results against other methods on our rendered dataset following our camera positions. Note that our method reproject at the center of the camera rig instead of at the center of the two reference cameras to compare fairly against the other methods. This harms the quality of our stitched panorama.

* CrownConv inference runs with a fixed number of vertices (10242).
† no reference implementation is provided.
‡ part of the method runs on CPU.

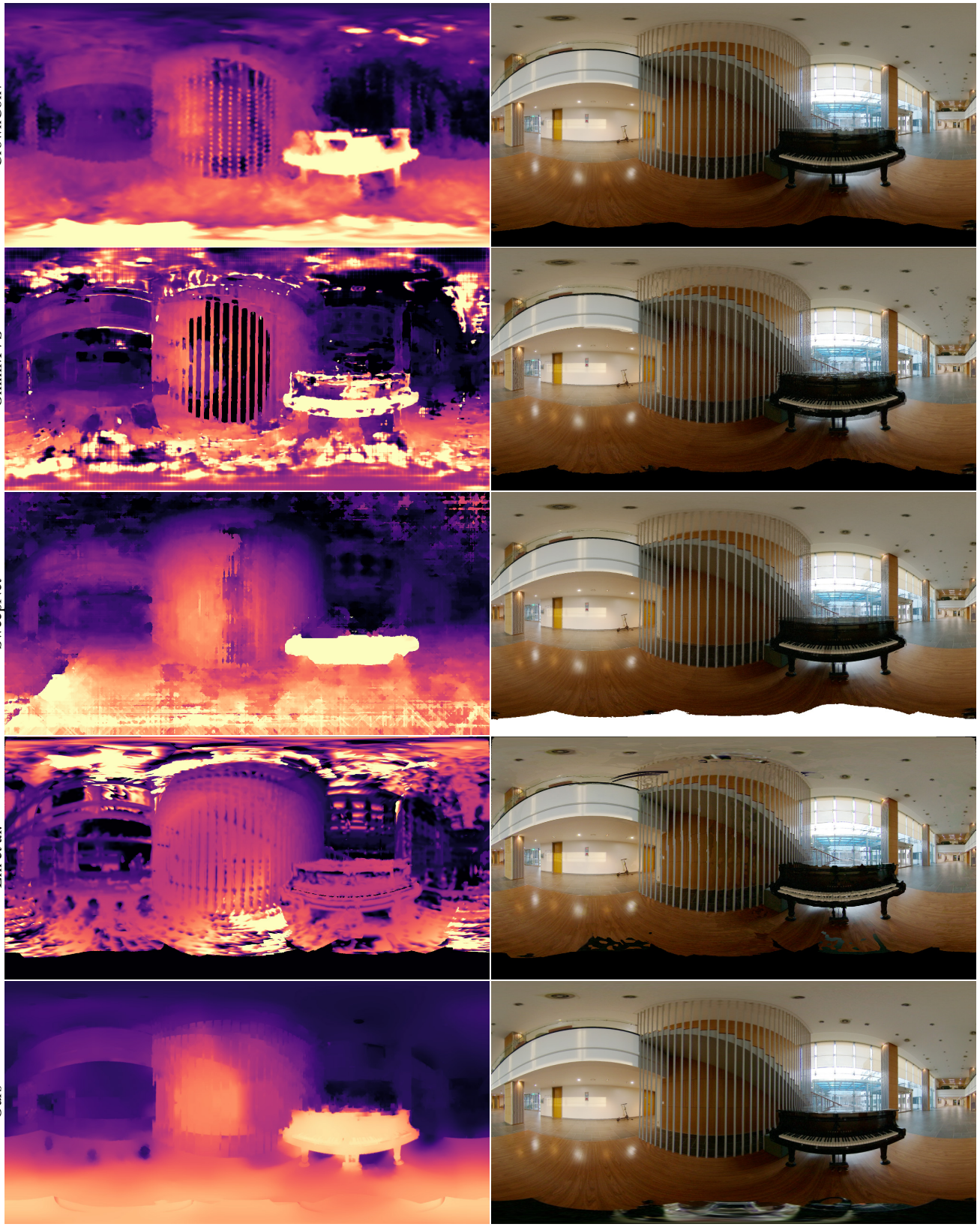

Figure 3: Results on real scene.

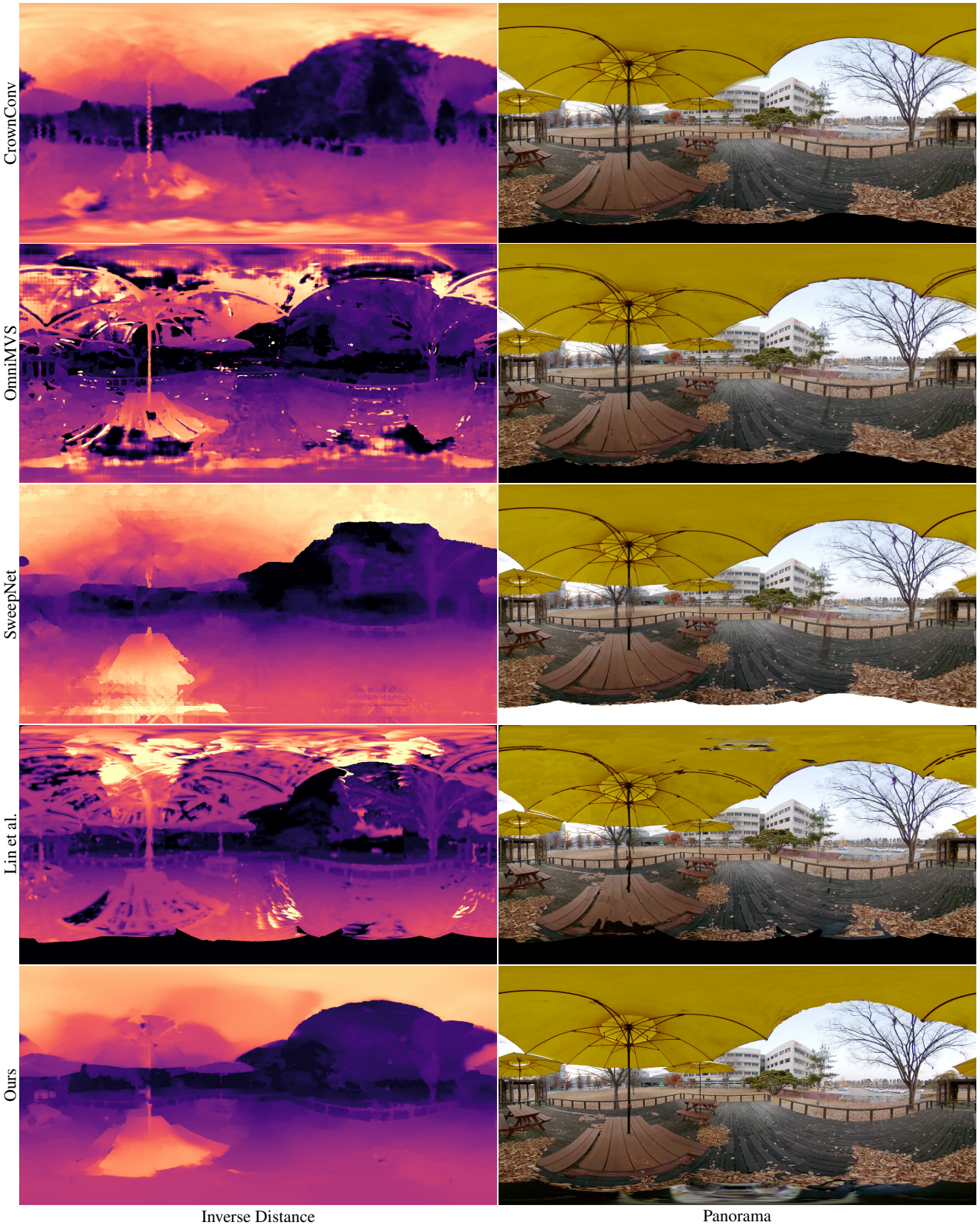

Figure 4: Results on real scene.

Figure 5: Results on real scene.

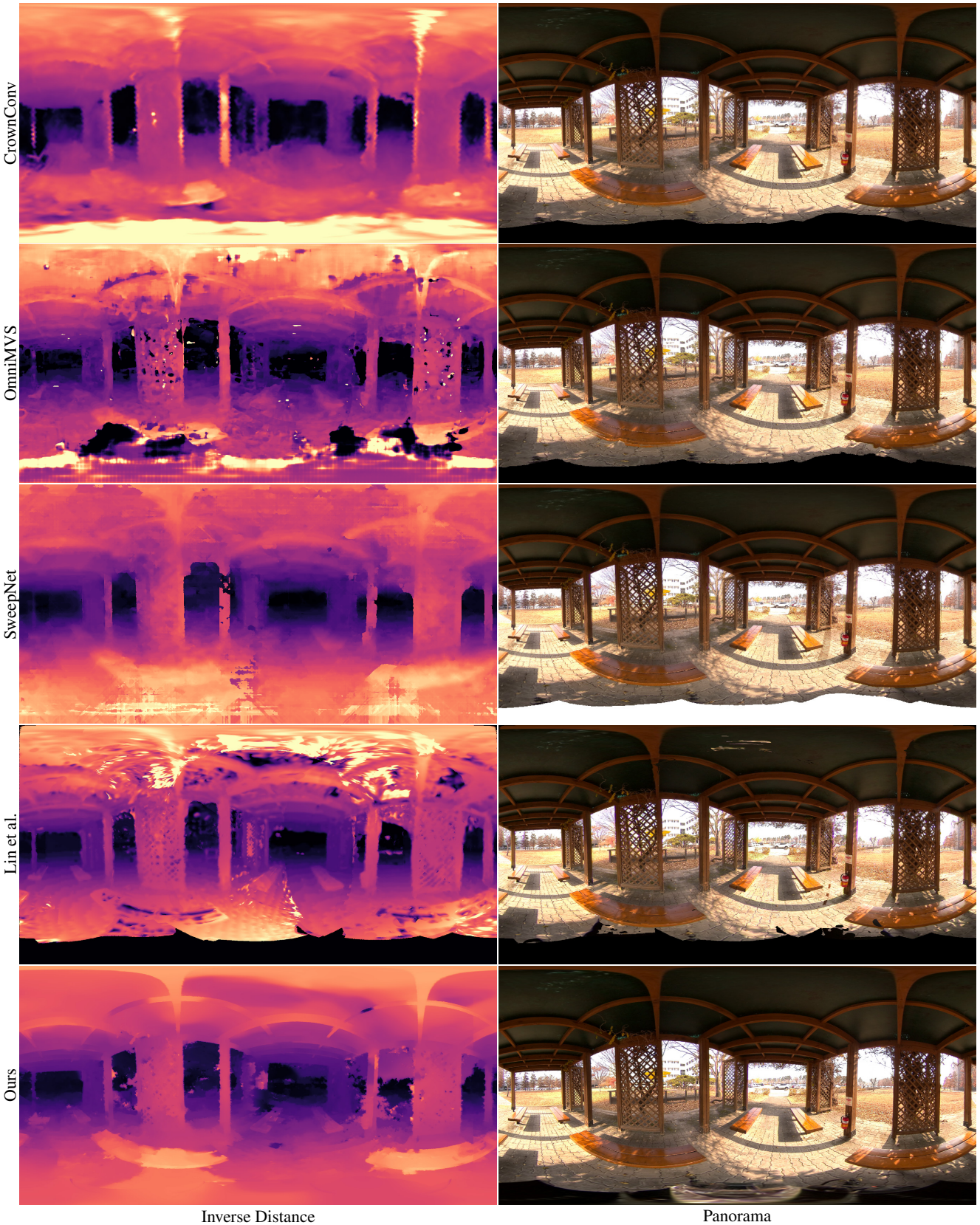

Figure 6: Results on real scene.

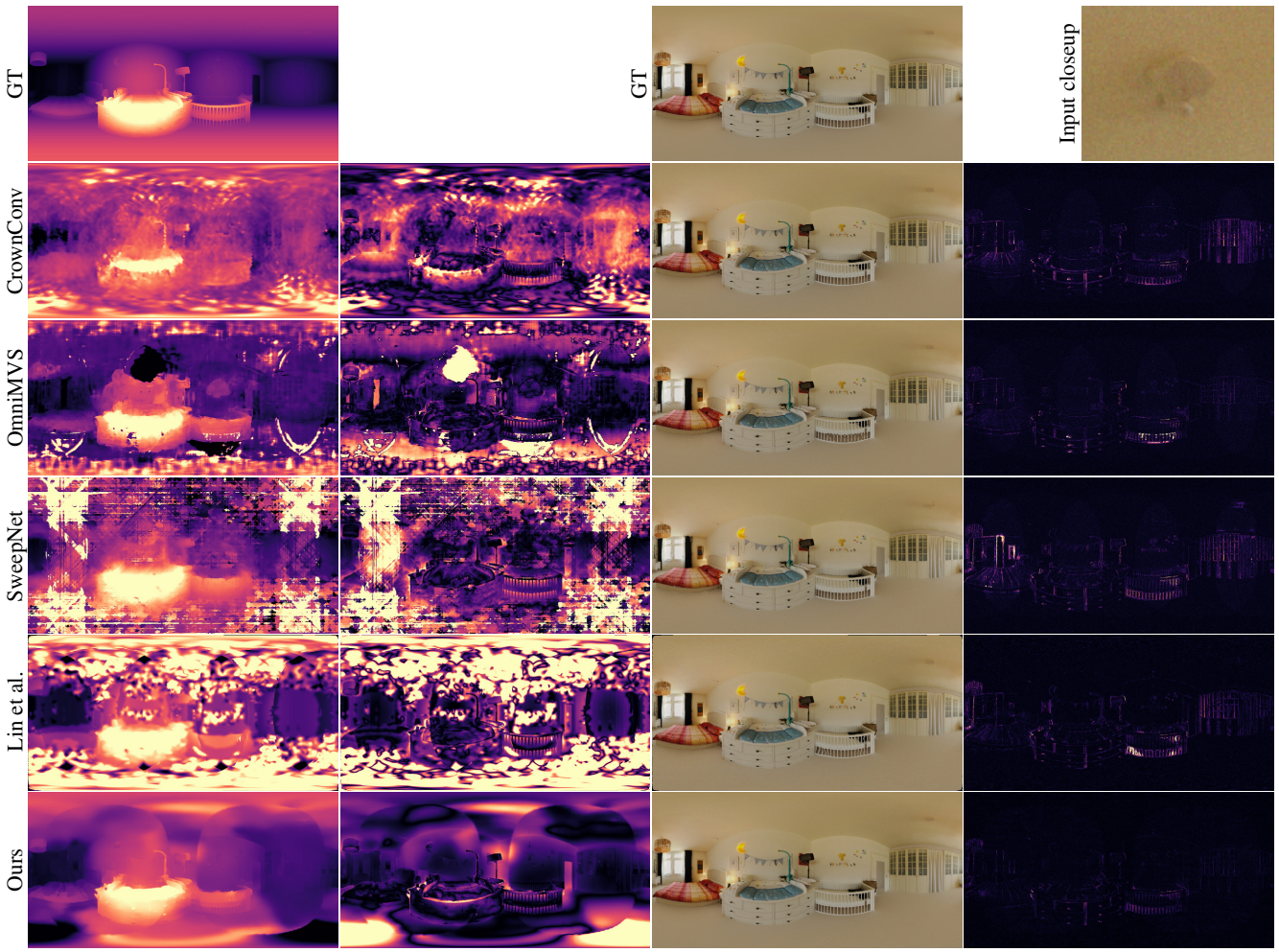

Figure 7: Results on our noisy synthetic dataset.

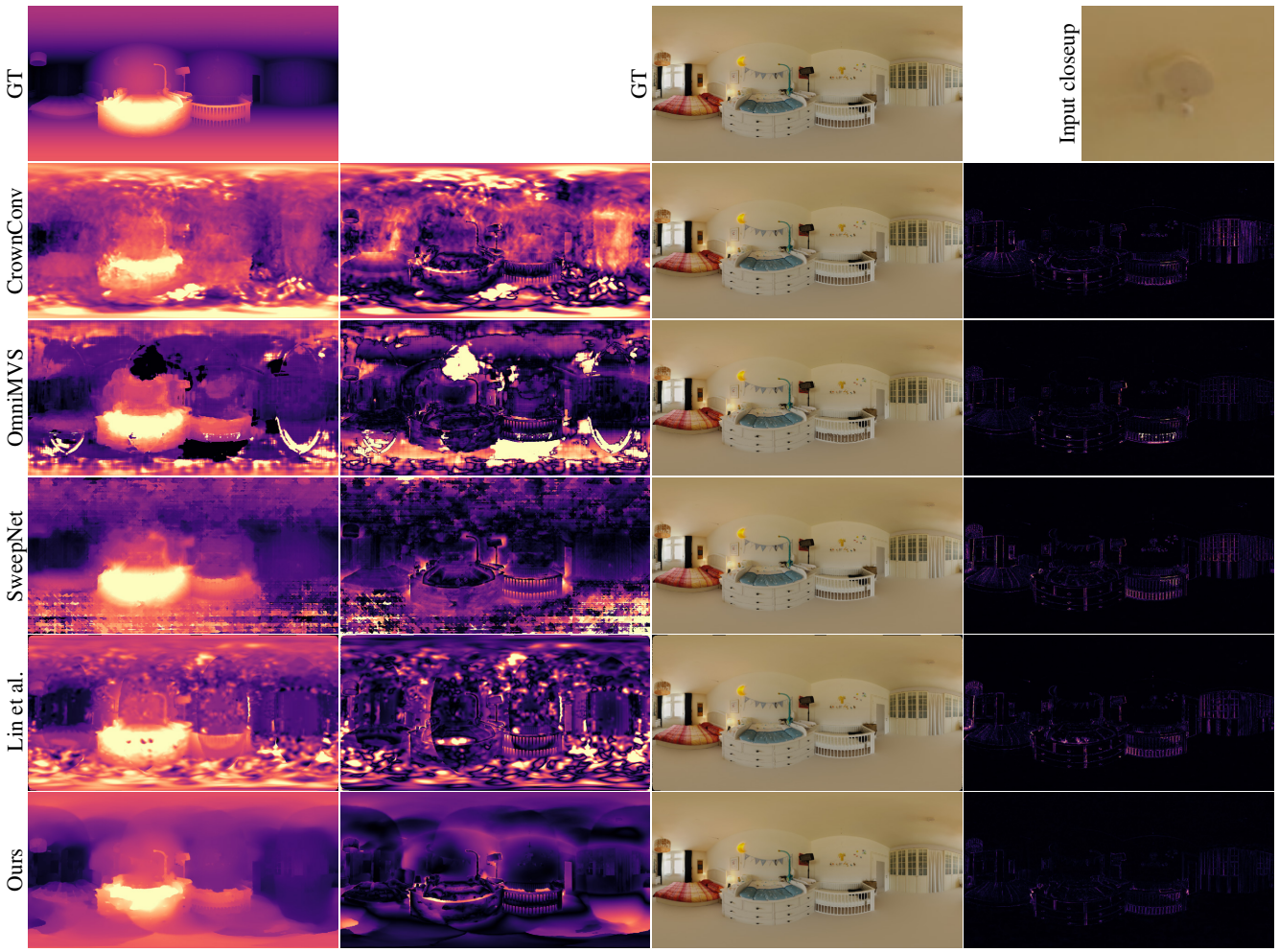

Figure 8: Results on synthetic dataset without noise.

References

[1] Ren Komatsu, Hiromitsu Fujii, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama. 360° depth estimation from multiple fisheye images with origami crown representation of icosahedron. In *IROS*, 2020. 2

[2] Hong-Shiang Lin, Chao-Chin Chang, Hsu-Yu Chang, Yung-Yu Chuang, Tzong-Li Lin, and Ming Ouhyoung. A low-cost portable polycamera for stereoscopic 360° imaging. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2

[3] Manuel Oliveira, Brian Bowen, Richard McKenna, and Yu-Sung Chang. Fast digital image inpainting. *VIIP*, pages 261–266, 01 2001. 1

[4] Changhee Won, Jongbin Ryu, and Jongwoo Lim. OmniMVS: End-to-end learning for omnidirectional stereo matching. In *ICCV*, 2019. 2

[5] Changhee Won, Jongbin Ryu, and Jongwoo Lim. SweepNet: Wide-baseline omnidirectional depth estimation. In *ICRA*, 2019. 2