Supplemental Document:
Egocentric Scene Reconstruction from an Omnidirectional Video

This supplemental document includes an extended review and discussion of the related work on spherical depth estimation (Section 1). In Section 2 and Figure 4, we present additional results for our method, including quantitative and qualitative comparisons as well as its computational performance. Table 1 summarizes the main symbols used in the paper.

1 FURTHER RELATED WORK ON SPHERICAL DEPTH ESTIMATION

Depth maps for spherical images can be estimated from monocular, binocular or multi-view input.

Monocular. Deep learning has facilitated monocular depth estimation, also for spherical images. Supervised approaches are mostly trained on synthetic datasets due to the difficulty of acquiring ground-truth spherical depth maps [Eder et al. 2019; Zioulis et al. 2018]. Accuracy can be improved by fusing predictions for equirectangular and cubemap projections [Jiang et al. 2021; Wang et al. 2020b], leveraging the geometric structure of indoor scenes [Jin et al. 2020; Pintore et al. 2021; Sun et al. 2021; Zeng et al. 2020], or self-supervised training via view synthesis [Zioulis et al. 2019]. Tateno et al. [2018] introduce an approach for adapting pre-trained monocular depth estimation for perspective images [Godard et al. 2017; Li et al. 2021; Ranftl et al. 2021] to spherical images using distortion-aware convolutional filters. However, the performance of these learning-based approaches highly depends on their training data. Most spherical image datasets are synthetic and only consider indoor scenes, and the methods therefore tend to perform poorly on real and/or outdoor scenes. Rey-Area et al. [2022] address this domain gap by aligning and blending more robust perspective monocular depth estimation [Ranftl et al. 2021] on tangent images [Eder et al. 2020]. However, monocular depth estimation is not consistent in scale across multiple views, which makes it difficult to reconstruct scene geometry from multiple monocular depth maps.

Binocular Stereo. Learning-based spherical stereo methods either assume a known, fixed camera baseline [Lai et al. 2019; Wang et al. 2020a], or also estimate the relative camera pose [Wang et al. 2018]. They again mostly rely on synthetic training data, making them unsuitable for real outdoor scenes. Spherical rectification [Li 2008; Matzen et al. 2017] aligns epipolar lines between a pair of spherical stereo images. This allows the processing of spherical stereo images with existing stereo correspondence methods that are designed for pinhole images. Learning-based correspondence techniques [Saikia et al. 2019; Teed and Deng 2020] have shown great performance for perspective images, and they can now also be applied to a rectified spherical image pair. However, their performance is limited by the distortion of rectified spherical images compared to perspective images. To overcome this problem, we create a synthetic spherical RGBD video dataset for fine-tuning a state-of-the-art perspective optical flow network [Teed and Deng 2020] on rectified spherical stereo image pairs. This adapts the optical flow network to the

Table 1. Main symbols used in the main paper.

Symbol	Description
α	angular extent of a node, used for calculating the solid angle of a node (Equation 9)
δ	angular disparity (Equation 6)
θ	polar angle in spherical binoctree (Equation 8)
ϕ	azimuth angle in spherical binoctree (Equation 8)
ϕ_{neigh}	angle between baseline and a line from the neighbor camera C_{neigh} to the point P (Figure 3)
ϕ_{rect}	azimuth angle in transverse equirectangular projection (Equation 12)
ϕ_{ref}	angle between baseline and a line between a world point and the center of the reference camera (Figure 3)
σ_{c}	parameter controlling the color consistency weight w_{c} (Equation 14)
σ_{d}	parameter controlling the depth consistency weight w_{d} (Equation 13)
σ_{p}	parameter controlling the proximity weight w_{p} (Equation 11)
Δ	disparity in pixels (Section 3.1)
Ω	solid angle (Equation 9)
b	baseline between two cameras (Equation 6)
d	radial distance (*aka* depth), e.g. of a point to a camera (Equation 6)
e_{m}	slope of the TSDF truncation threshold function (Equation 10)
e_{n}	offset of the TSDF truncation threshold function (Equation 10)
d_{node}	distance between the center point of a node and the center of a camera (Equation 9)
p	a pixel point corresponding to the 3D point P (Equation 11)
t	length used for solid angle calculation in Equation 9
w	width of a 2D image (Section 3.2)
w_{c}	color consistency term of TSDF weight (Equation 14)
w_{d}	depth consistency term of TSDF weight (Equation 13)
w_{p}	proximity term of TSDF weight (Equation 11)
w_{update}	total TSDF weight (Equation 15)
C_{neigh}	camera center of the neighbor camera (Figure 3)
C_{ref}	camera center of the capturing/reference camera (Figure 6)
D^{est}	estimated depth map (Equation 10)
D^{rend}	rendered depth map (Equation 16)
D^{tri}	depth map from triangle center to camera center (Equation 16)
I_i	input image for camera/frame i (Equation 14)
K	number of neighboring views used for spherical depth estimation (Section 3.1.5)
M	depth consistency mask (Equation 17)
M^*	blurred depth consistency mask (Equation 18)
M'	soft depth consistency mask (Equation 18)
$N(i)$	set of neighboring views of view i (Equation 15)
O_{tree}	origin of the octree (Figure 6)
P	a 3D point (Section 3.1.3, Section 3.2.2)
S	visibility score (Equation 19)
T_{solid}	solid angle threshold that decides the size of each node (Section 3.2.2)
$T_{\text{trunc}}(d)$	TSDF truncation threshold, a function of depth d (Equation 10)
$V(p)$	visibility ratio for a pixel p (Equation 16)
V_{node}	volume of a spherical frustum (Equation 8)

specific distortions in rectified spherical images and achieves state-of-the-art performance, as we demonstrate in Section 2.1.

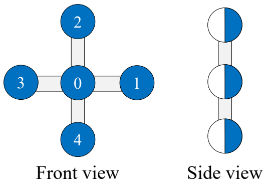
Multi-view Stereo. Im et al. [2016] pioneered sphere sweeping for computing depth maps from multiple input views in analogy to plane sweeping for perspective multi-view stereo [Collins 1996]. Sphere sweeping creates a spherical cost volume from a set of concentric virtual spheres that are used to align all input views; winner-take-all then determines the optimal depth per pixel. Several methods have extended this traditional cost volume approach to use learned, deep features, and to regress the output depth map from the deep cost volume to increase performance [Komatsu et al. 2020; Won et al. 2019a,b]. However, these methods are limited to depth map resolutions of only 640×360 pixels, which is insufficient for high-quality scene geometry reconstruction. da Silveira and Jung [2019] reconstruct spherical depth maps from multiple pairwise flow fields, whose contributions are weighted according to the reprojection error, followed by a guided image filtering post-process. Recently, Meuleman et al. [2021] proposed a real-time sphere sweeping stereo method using four fisheye images as input. Although this method produces spherical RGBD videos in real time, it does not account for temporal coherence of reconstructed RGBD frames. None of these methods is specialized for 3D reconstruction from spherical input.

2 ADDITIONAL RESULTS

2.1 Spherical Depth Estimation

In Table 2, we compare the mean absolute error (MAE), RMSE, and the percentage of bad pixels with an error of more than 0.1 and 0.4 in inverse depth (lower is better for all metrics). Our method outperforms all other methods in every measure for every baseline. Table 3 shows a quantitative comparison of constant/adaptive truncation thresholds and our confidence weights used for TSDF fusion.

We compare multi-view depth estimation accuracy with Parra Pozo et al. [2019]. We chose this method as it supports unstructured camera setups and can estimate large field-of-view depth maps. For fair comparison, we placed five spherical cameras looking in the same direction in a cross-shaped camera layout (see right). We rendered 25 sets of equirectangular images at 1024×512 resolution (5 positions × 5 scenes), but evaluate the depth accuracy only on the frontal hemisphere (shown in blue). We feed Parra Pozo et al.'s pipeline with 180° fisheye images of the frontal hemispheres and ground-truth camera poses, and estimate the depth map for the central camera. For our method, we estimate the depth map of the central camera based on stereo pairs formed with each of the other four cameras, as described in Section 3.1.4 in the main paper. Figure 1 shows that our method performs best.



Real scenes often contain dynamic objects, reflections and textureless regions, which can be problematic when integrating depth estimates from different viewpoints or timestamps (see an example in Figure 2). To handle these problems, we introduce Gaussian-like weights for proximity weight, depth consistency, and color consistency, as described in Section 3.2.3 in the main paper.

Table 2. Comparison of two-view spherical depth estimation methods 360SD-Net [Wang et al. 2020a], SGBM [Hirschmüller 2008], GPU-SGM [Hernandez-Juarez et al. 2016], RAFT [Teed and Deng 2020], and our method. The columns '>0.1' and '>0.4' show the percentage of bad pixels that exceed an absolute inverse depth error of 0.1/0.4 [m^{-1}]. Our method significantly outperforms all others.

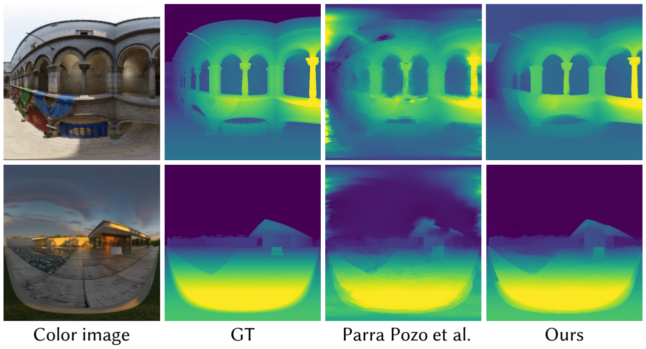
Baseline	Method	>0.1	>0.4	MAE	RMSE
10 cm	360SD-Net	73.79	45.09	0.699	0.955
	SGBM	16.02	3.98	0.070	0.172
	GPU-SGM	17.92	4.18	0.080	0.180
	RAFT	13.52	2.57	0.067	0.200
	Ours	**10.43**	**0.70**	**0.041**	**0.081**
20 cm	360SD-Net	52.94	25.53	0.332	0.544
	SGBM	14.33	4.23	0.066	0.180
	GPU-SGM	15.04	4.60	0.077	0.200
	RAFT	10.96	2.21	0.057	0.184
	Ours	**9.38**	**0.71**	**0.036**	**0.076**
30 cm	360SD-Net	44.25	15.59	0.211	0.369
	SGBM	14.74	4.91	0.070	0.191
	GPU-SGM	15.26	5.34	0.081	0.219
	RAFT	10.93	2.43	0.056	0.183
	Ours	**8.39**	**0.63**	**0.032**	**0.073**
40 cm	360SD-Net	39.45	10.69	0.163	0.291
	SGBM	15.74	5.53	0.076	0.202
	GPU-SGM	16.44	6.35	0.090	0.240
	RAFT	10.45	2.51	0.055	0.182
	Ours	**7.97**	**0.55**	**0.030**	**0.070**
mean	360SD-Net	52.61	24.22	0.351	0.540
	SGBM	15.21	4.66	0.070	0.186
	GPU-SGM	16.17	5.12	0.082	0.210
	RAFT	11.46	2.43	0.059	0.187
	Ours	**7.97**	**0.55**	**0.035**	**0.075**

Table 3. Quantitative geometric error for each case in Figure 7 of the main paper: constant vs adaptive truncation threshold, with and without our confidence weights. We evaluate the quality of just the mesh pixels ('Mesh') and all pixels ('Mesh+Skybox'). Completeness ('Comp.') is defined as the proportion of pixels that see the mesh compared to the ground truth. The confidence weights used for TSDF integration increase the accuracy of the reconstructed mesh, and the adaptive truncation threshold increases the mesh completeness.

	Mesh only		Mesh+Skybox		Comp.
Variants	MAE	RMSE	MAE	RMSE	%
(b) constant w/o weight	0.022	0.056	0.044	0.099	86.2
(c) adaptive w/o weight	0.025	0.057	0.024	0.057	98.7
(d) constant w/ weight	**0.017**	**0.051**	0.019	0.056	96.9
(e) adaptive w/ weight	0.018	**0.051**	**0.017**	**0.052**	**98.8**

2.2 Performance

We implemented our method on a desktop computer equipped with an Intel Core i9-10920X processor at 3.5 GHz with 128 GB RAM, and an NVIDIA Titan RTX graphics card. To produce the results of the Dinosaur scene in Figure 1 and Figure 10 in the main paper, we used as input a 17-second, handheld spherical video containing 512 frames at a resolution of 5760×2880 pixels. Our implementation took



Method	>0.1	>0.4	MAE	RMSE
Parra Pozo et al. [2019]	6.37	0.18	0.033	0.057
Ours	**5.56**	**0.11**	**0.026**	**0.044**

Fig. 1. Comparison of hemispherical depth maps estimated from five input views. Our method produces cleaner depth maps with fewer artifacts.

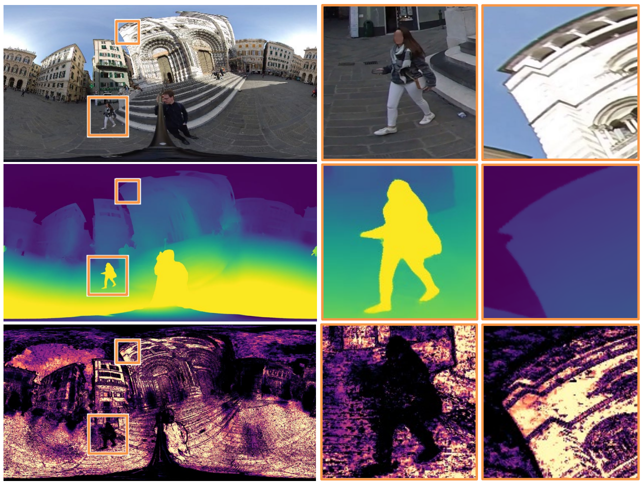
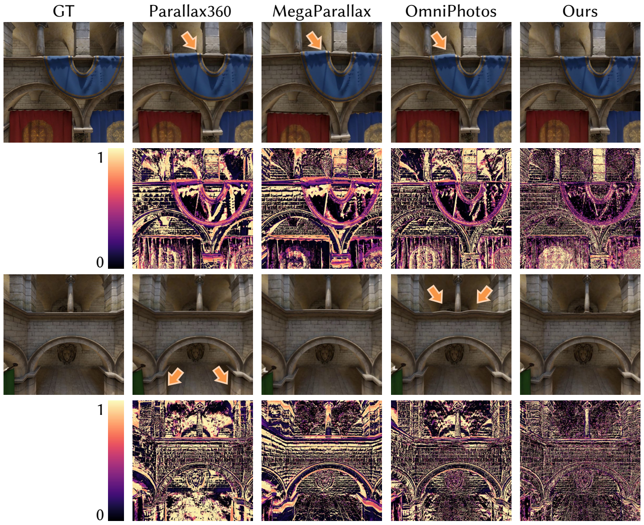


Fig. 2. Consistency-based weight map. **Top:** Input color image. **Middle:** Estimated inverse depth map. Incorrect depth from dynamic objects will be inconsistent with depths from neighbor frames (see highlighted person). **Bottom:** Visualization of our depth and color consistency ($w_\mathrm{d} \times w_\mathrm{c}$). Inconsistent depths and/or colors results in small weights (e.g., dark person outline in the left closeup), while consistent depths and colors results in high weights (e.g., bright regions in the right closeup).

on average 3.15 seconds to estimate one depth map at a resolution of 1728×864, using $K = 11$ neighbor frames. In total, it took about 27 minutes to estimate all depth maps. Fusing the 512 depth maps into a triangle mesh with 3.6 million faces took 75 seconds overall: 60 seconds are spent on generating the spherical binoctree with 9.8 million nodes using OpenMP, 10 second for updating the TSDF values on the GPU, and 5 seconds for extracting the mesh using dual marching cubes. The texture map with 8-pixel triangles has a size of 9,893×8,543 pixels and is reconstructed on the GPU in 45 seconds. The total run time is less than 30 minutes end-to-end.



Method	PSNR	SSIM	LPIPS
Parallax360 [Luo et al. 2018]	22.47	0.814	0.445
MegaParallax [Bertel et al. 2019]	20.78	0.789	0.450
OmniPhotos [Bertel et al. 2020]	24.99	0.851	0.333
Ours	**26.91**	**0.897**	**0.259**

Fig. 3. Synthesized views and error maps compared to the ground truth. **Top:** Rows 1 and 3: Ground-truth color image and synthesized novel views. Arrows indicate visual artifacts. Rows 2 and 4: Absolute color difference map between the synthesized novel views and ground truth. **Bottom:** Quality of novel-view synthesis, as measured using the PSNR, SSIM [Wang et al. 2004], and LPIPS [Zhang et al. 2018] image similarity metrics.

2.3 Novel-View Synthesis Evaluation

We next evaluate novel-view synthesis results using our textured meshes compared to state-of-the-art image-based rendering methods: Parallax360 [Luo et al. 2018], MegaParallax [Bertel et al. 2019], and OmniPhotos [Bertel et al. 2020]. As input, we rendered a 200-frame 3-loop horizontal circular trajectory with a radius of 55 cm in the Sponza scene. To compare the methods, we rendered views at the center of the circular trajectory. We create our texture map from 10 frames.

The comparison in Figure 3 shows that our method produces more accurate novel views with lower errors compared to the other methods. While Parallax360 and MegaParallax show fewer visual artifacts due to their smooth proxy geometry, they still introduce visual distortions that are noticeable in the error maps in Figure 3. OmniPhotos's proxy geometry approximates the scene geometry more closely, which results in better image metrics, but some straight lines in the test scene end up warped. Our method shows the best results, both visually and quantitatively in Figure 3, as our method is based on more accurate reconstruction of 3D geometry and texture.

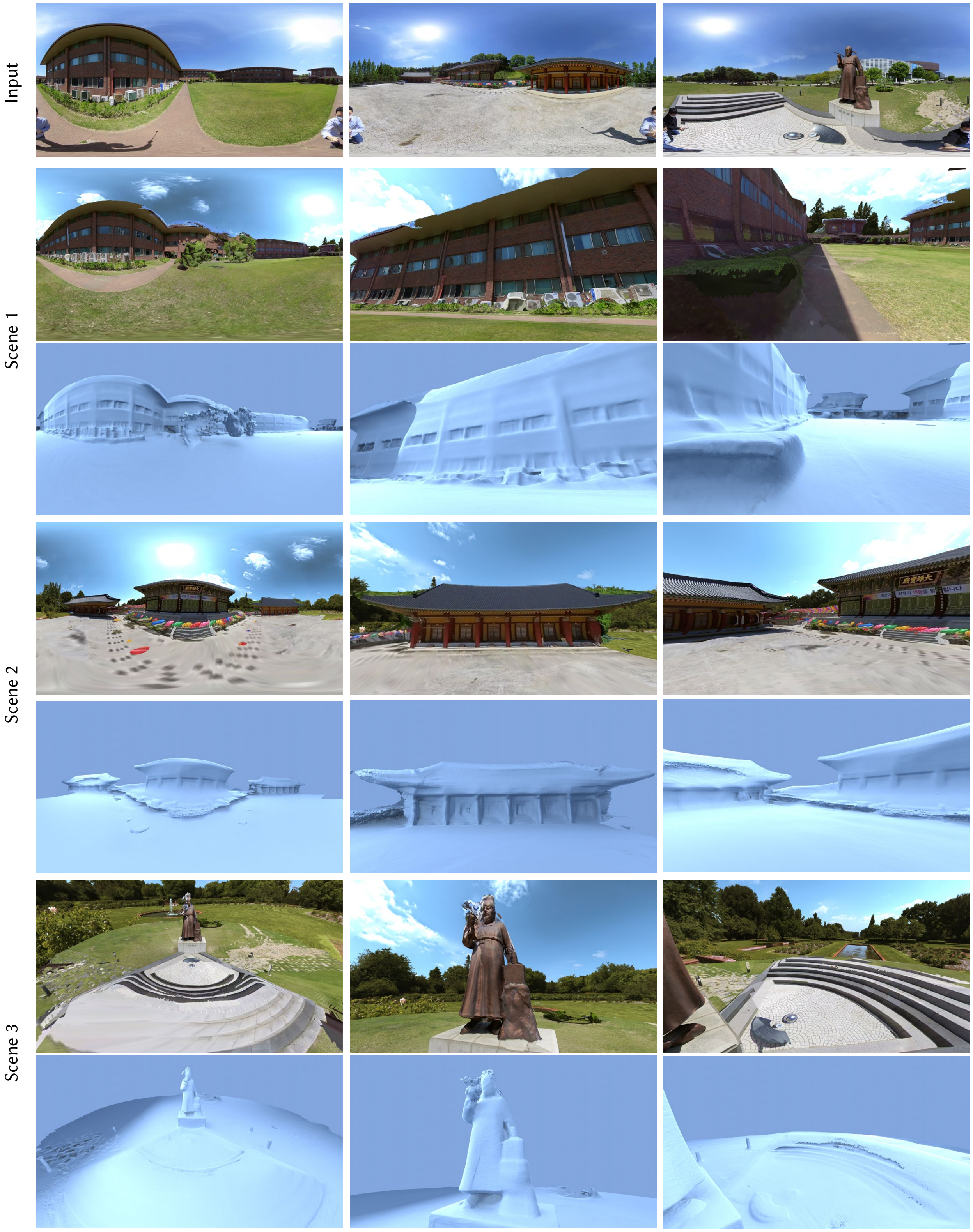


Fig. 4. Additional results from real scenes captured with a large camera trajectory. **First row**: A frame of each input video. **For each scene—Top:** Our reconstructed textured mesh with cube-map sky-box. **Bottom:** Our reconstructed geometry.

REFERENCES

Tobias Bertel, Neill D. F. Campbell, and Christian Richardt. 2019. MegaParallax: Casual 360° Panoramas with Motion Parallax. *IEEE Trans. Vis. Comput. Graph.* 25, 5 (2019), 1828–1835. doi: 10.1109/TVCG.2019.2898799

Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. 2020. OmniPhotos: Casual 360° VR Photography. *ACM Trans. Graph.* 39, 6 (2020), 267:1–12. doi: 10.1145/3414685.3417770

Robert T. Collins. 1996. A space-sweep approach to true multi-image matching. In *CVPR*. 358–363. doi: 10.1109/CVPR.1996.517097

Thiago Lopes Trugillo da Silveira and Claudio R. Jung. 2019. Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications. In *IEEE VR*. 9–18. doi: 10.1109/VR.2019.8798281

Marc Eder, Pierre Moulon, and Li Guan. 2019. Pano Popups: Indoor 3D Reconstruction with a Plane-Aware Network. In *3DV*. 76–84. doi: 10.1109/3DV.2019.00018

Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. 2020. Tangent Images for Mitigating Spherical Distortion. In *CVPR*. doi: 10.1109/CVPR42600.2020.01244

Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. 2017. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *CVPR*. 6602–6611. doi: 10.1109/CVPR.2017.699

Daniel Hernandez-Juarez, Alejandro Chacón, Antonio Espinosa, David Vázquez, Juan Carlos Moure, and Antonio M. López. 2016. Embedded Real-time Stereo Estimation via Semi-Global Matching on the GPU. In *International Conference on Computational Science*. 143–153. doi: 10.1016/j.procs.2016.05.305

Heiko Hirschmüller. 2008. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal.* 30, 2 (2008), 328–341. doi: 10.1109/TPAMI.2007.1166

Sunghoon Im, Hyowon Ha, François Rameau, Hae-Gon Jeon, Gyeongmin Choe, and In So Kweon. 2016. All-around Depth from Small Motion with A Spherical Panoramic Camera. In *ECCV*. doi: 10.1007/978-3-319-46487-9_10

Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. 2021. UniFuse: Unidirectional Fusion for 360° Panorama Depth Estimation. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1519–1526. doi: 10.1109/LRA.2021.3058957

Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. 2020. Geometric Structure Based and Regularized Depth Estimation From 360 Indoor Imagery. In *CVPR*. 886–895. doi: 10.1109/CVPR42600.2020.00097

Ren Komatsu, Hiromitsu Fujii, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama. 2020. 360° Depth Estimation from Multiple Fisheye Images with Origami Crown Representation of Icosahedron. In *IROS*. doi: 10.1109/IROS45743.2020.9340981

Po Kong Lai, Shuang Xie, Jochen Lang, and Robert Laganière. 2019. Real-time panoramic depth maps from omni-directional stereo images for 6 DoF videos in virtual reality. In *IEEE VR*. 405–412. doi: 10.1109/VR.2019.8798016

Shigang Li. 2008. Binocular Spherical Stereo. *IEEE Transactions on Intelligent Transportation Systems* 9, 4 (2008), 589–600. doi: 10.1109/TITS.2008.2006736

Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. 2021. MannequinChallenge: Learning the Depths of Moving People by Watching Frozen People. *IEEE Trans. Pattern Anal.* 43, 12 (2021), 4229–4241. doi: 10.1109/TPAMI.2020.2974454

Bicheng Luo, Feng Xu, Christian Richardt, and Jun-Hai Yong. 2018. Parallax360: Stereoscopic 360° Scene Representation for Head-Motion Parallax. *IEEE Trans. Vis. Comput. Graph.* 24, 4 (2018), 1545–1553. doi: 10.1109/TVCG.2018.2794071

Kevin Matzen, Michael F. Cohen, Bryce Evans, Johannes Kopf, and Richard Szeliski. 2017. Low-cost 360 Stereo Photography and Video Capture. *ACM Trans. Graph.* 36, 4 (2017), 148:1–12. doi: 10.1145/3072959.3073645

Andréas Meuleman, Hyeonjoong Jang, Daniel S. Jeon, and Min H. Kim. 2021. Real-Time Sphere Sweeping Stereo from Multiview Fisheye Images. In *CVPR*. doi: 10.1109/CVPR46437.2021.01126

Albert Parra Pozo, Michael Toksvig, Terry Filiba Schrager, Joyse Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. 2019. An Integrated 6DoF Video Camera and System Design. *ACM Trans. Graph.* 38, 6 (2019), 216:1–16. doi: 10.1145/3355089.3356555

Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti. 2021. SliceNet: Deep Dense Depth Estimation From a Single Indoor Panorama Using a Slice-Based Representation. In *CVPR*. 11531–11540. doi: 10.1109/CVPR46437.2021.01137

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2021. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. *IEEE Trans. Pattern Anal.* (2021). doi: 10.1109/TPAMI.2020.3019967

Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 2022. 360MonoDepth: High-Resolution 360° Monocular Depth Estimation. In *CVPR*. https://manurare.github.io/360monodepth/

Tonmoy Saikia, Yassine Marrakchi, Arber Zela, Frank Hutter, and Thomas Brox. 2019. AutoDispNet: Improving Disparity Estimation With AutoML. In *ICCV*. 1812–1823. doi: 10.1109/ICCV.2019.00190

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2021. HoHoNet: 360 Indoor Holistic Understanding with Latent Horizontal Features. In *CVPR*. 2573–2582. doi: 10.1109/CVPR46437.2021.00260

Keisuke Tateno, Nassir Navab, and Federico Tombari. 2018. Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images. In *ECCV*. 732–750. doi: 10.1007/978-3-030-01270-0_43

Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*. doi: 10.1007/978-3-030-58536-5_24

Fu-En Wang, Hou-Ning Hu, Hsien-Tzu Cheng, Juan-Ting Lin, Shang-Ta Yang, Meng-Li Shih, Hung-Kuo Chu, and Min Sun. 2018. Self-Supervised Learning of Depth and Camera Motion from 360° Videos. In *ACCV*. doi: 10.1007/978-3-030-20873-8_4

Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. 2020b. BiFuse: Monocular 360 Depth Estimation via Bi-Projection Fusion. In *CVPR*. 462–471. doi: 10.1109/CVPR42600.2020.00054

Ning-Hsu Wang, Bolivar Solarte, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 2020a. 360SD-Net: 360° Stereo Depth Estimation with Learnable Cost Volume. In *ICRA*. 582–588. doi: 10.1109/ICRA40945.2020.9196975

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 4 (2004), 600–612. doi: 10.1109/TIP.2003.819861

Changhee Won, Jongbin Ryu, and Jongwoo Lim. 2019a. OmniMVS: End-to-End Learning for Omnidirectional Stereo Matching. In *ICCV*. 8986–8995. doi: 10.1109/ICCV.2019.00908

Changhee Won, Jongbin Ryu, and Jongwoo Lim. 2019b. SweepNet: Wide-baseline Omnidirectional Depth Estimation. In *ICRA*. doi: 10.1109/ICRA.2019.8793823

Wei Zeng, Sezer Karaoglu, and Theo Gevers. 2020. Joint 3D Layout and Depth Prediction from a Single Indoor Panorama Image. In *ECCV*. doi: 10.1007/978-3-030-58517-4_39

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*. doi: 10.1109/CVPR.2018.00068

Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. 2019. Spherical View Synthesis for Self-Supervised 360° Depth Estimation. In *3DV*. 690–699. doi: 10.1109/3DV.2019.00081

Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. 2018. OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas. In *ECCV*. 448–465. doi: 10.1007/978-3-030-01231-1_28