PanoPose: Self-supervised Relative Pose Estimation for Panoramic Images

Supplementary Material

1. Evaluation Metrics

In the experiment section of the manuscript, various evaluation metrics are used to assess different methods. In this section, we give a detailed calculation of these metrics. The relative rotation error (RRE) and relative translation angle error (RTAE) are computed as

$$RRE = \arccos \frac{tr(\mathbf{R}_{pre}\mathbf{R}_{gt}^T) - 1}{2} \tag{1}$$

$$RTAE = \arccos \frac{\mathbf{t}_{pre}^T \mathbf{t}_{gt}}{\|\mathbf{t}_{pre}\|_2 \|\mathbf{t}_{gt}\|_2} , \tag{2}$$

where $(\mathbf{R}_{pre}, \mathbf{t}_{pre})$ is the estimated relative pose and $(\mathbf{R}_{gt}, \mathbf{t}_{gt})$ is the ground truth. $tr(\cdot)$ computes the trace of a matrix. Since the relative translation obtained from the image has scale singularity, a direct comparison between the scale of the predicted relative translation and the scale of ground truth is unreliable and cannot measure the effectiveness between different methods. Thus, the relative scale error (RSE) is computed as

$$RSE = \frac{s\|\mathbf{t}_{pre}\|_2 - \|\mathbf{t}_{gt}\|_2}{\|\mathbf{t}_{gt}\|_2} , \tag{3}$$

where s is a scaling factor to make the estimated scale as close as possible to the true scale. s is not a constant number and it is computed as

$$\arg\min_s \frac{1}{n} \sum_{i=0}^{n} \left(s\|{}^i\mathbf{t}_{pre}\|_2 - \|{}^i\mathbf{t}_{gt}\|_2\right)^2 . \tag{4}$$

Here, n is the number of total validation pairs, $\|{}^i\mathbf{t}_{pre}\|$ and $\|{}^i\mathbf{t}_{gt}\|$ is the estimated relative translation and ground truth relative translation of i-th pair, respectively. After an epoch of training, the relative translation of the network output will change, and therefore the scaling factor s will also change.

As for the global pose error, the predicted poses are aligned with the ground truth before evaluation. The computation of absolute rotation error (ARE) is the same as RRE (Eq. (1)) and the difference is that $(\mathbf{R}_{pre}, \mathbf{R}_{gt})$ is the estimated global rotation and the ground truth. The absolute translation error (ATE) is computed as

$$ATE = \|\mathbf{t}_{pre} - \mathbf{t}_{gt}\|_2^2 . \tag{5}$$

Here, $(\mathbf{t}_{pre}, \mathbf{t}_{gt})$ is the estimated global translation and the ground truth.

2. Details of Datasets

In this section, we give more detail of the datasets used in the experiment, namely PanoSUNCG, Mapillary Metropolis, 360VO, Building, and Campus. PanoSUNCG and 360VO are synthetic datasets for indoor and outdoor environments, respectively. Mapillary Metropolis dataset is collected in real-world scenes. However, this dataset has a limitation because it samples images uniformly at a fixed distance of 6 meters, resulting in a sparsely connected pose graph that makes absolute pose estimation challenging. To address this limitation, we have collected our own datasets using an Insta 360 ONE X2 panoramic camera, named Building and Campus datasets. The visualization of these datasets is shown in Fig. 1, where the first and third rows are the RGB images, and the second and fourth rows are the corresponding depth images. The depth map of the 360VO dataset is not available and we use NA to represent it. Fig. 2 shows the camera trajectory of the different datasets. Since PanoSUNCG and 360VO datasets have multiple scenes, we only show one of them.

3. Additional Experiment Result

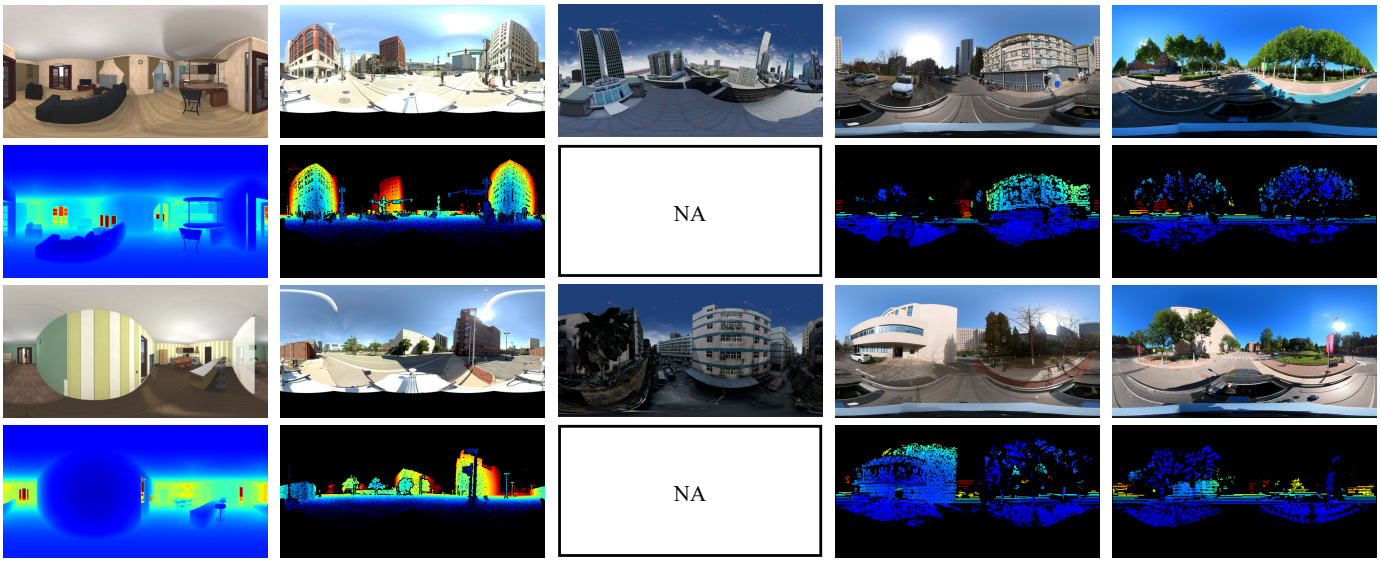
In this section, we demonstrate the additional experimental results of our PanoPose, in terms of relative pose and depth estimation.

3.1. Relative Pose Estimation Result

On the 360VO dataset, sequences 1, 4, and 9 are used for relative pose evaluation. Due to the space limitation, only the results from sequence 1 are reported in the manuscript. In this section, we present the relative pose estimation result on 360VO-Seq4 and 360VO-Seq9 in Tab. 1. From the table, it is clear that the traditional five-point method [2] achieves the best result in relative rotation estimation (RRE) and relative translation direction estimation (RTAE). However, the five-point method is unable to estimate the scale of relative translation, thus leading to the largest RSE in all methods. Our PanoPose outperforms other self-supervised methods in most evaluation metrics. An exception is the mean RSE on 360VO-seq4, where PanoPose's error is only 0.03 higher than the best result, achieving the second-best in learning-based methods.

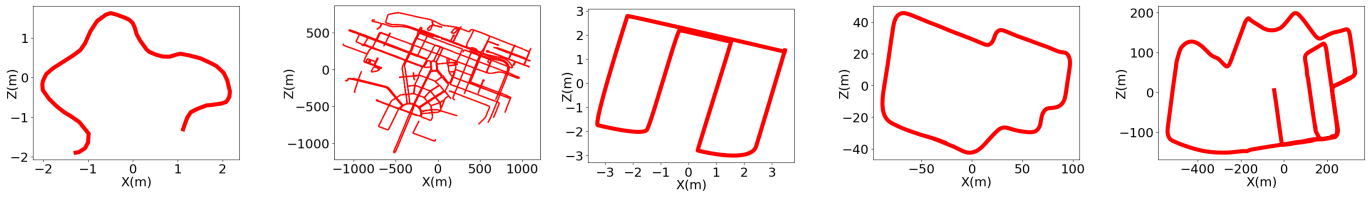
3.2. Cross-data Generalization

In this section, we demonstrate the generalization ability of our PanoPose by training and testing the network with various datasets, as summarized in Tab. 2.



(a) PanoSUNCG (b) Mapillary Metropolis (c) 360VO (d) Building (e) Campus

Figure 1. Visualization of the dataset used in the experiments. The first and third rows are the panoramic images. The second and the fourth rows are corresponding depth maps, in which red indicates large depth and blue is small depth.



(a) PanoSUNCG (b) Mapillary Metropolis (c) 360VO (d) Building (e) Campus

Figure 2. Camera trajectory of different datasets.

Dataset	Method	Mean RRE	Med RRE	Mean RTAE	Med RTAE	Mean RSE	Med RSE
360VO-Seq4	SfMLearner [6]	0.2195	0.0981	1.2846	0.8834	**0.1498**	0.0784
	MonoDepth2 [1]	0.1612	0.1047	0.7302	0.4817	0.1577	0.0851
	NonLocal-DPT [5]	0.1744	0.0992	0.5572	0.4211	0.2009	0.1135
	BiFuse++ [4]	0.7986	0.1119	9.0365	4.0132	0.4031	0.2003
	Five-point [2]	**0.0383**	**0.0396**	**0.4238**	**0.2027**	2.0326	0.4711
	PanoPose	0.0656	0.0626	0.4809	0.3316	0.1770	**0.0775**
360VO-Seq9	SfMLearner [6]	0.4891	0.2588	4.5872	0.9805	0.3187	0.1559
	MonoDepth2 [1]	0.2263	0.1532	3.8979	0.4528	0.2296	0.0949
	NonLocal-DPT [5]	0.2514	0.1765	3.0507	0.8773	0.3097	0.2138
	BiFuse++ [4]	0.5993	0.1312	**2.3713**	1.1305	0.3313	0.0937
	Five-point [2]	**0.1133**	**0.0791**	3.1101	**0.3642**	2.1149	0.4426
	PanoPose	0.1833	0.1219	2.9274	0.4302	**0.2181**	**0.0834**

Table 1. Relative pose estimation error. The unit of relative rotation error (RRE) and relative translation angle error (RTAE) is degree, and relative scale error (RSE) is unitless. The best result is shown in **bold** and the second best is shown with under line.

From the table, it is evident that PanoPose exhibits generalization capabilities in relative rotation estimation. It outperforms other methods in both mean RRE and median RRE across most dataset combinations. Notably, the only exceptions are when trained on the Campus dataset and evaluated on the Building and Mapillary Metropolis

T. Dataset	E. Dataset	Method	Mean RRE	Med RRE	Mean RTAE	Med RTAE	Mean RSE	Med RSE
Mapillary Metropolis	Building	SfMLearner [6]	1.8728	0.6991	168.2549	172.5138	0.4867	0.4236
		MonoDepth2 [1]	1.9971	0.7209	173.3312	175.1398	0.4899	0.4147
		Bifuse++ [4]	2.0330	0.8648	175.8732	176.7391	0.4546	0.3029
		PanoDepth	**0.7531**	**0.5581**	**149.0472**	**151.3470**	**0.4543**	**0.3020**
Mapillary Metropolis	Campus	SfMLearner [6]	1.2987	0.6774	8.9657	6.2198	278.2934	0.3593
		MonoDepth2 [1]	1.0625	0.4298	7.5949	5.0971	**145.7589**	0.3801
		Bifuse++ [4]	1.3831	0.9700	7.9631	6.1899	387.6580	0.3456
		PanoDepth	**0.5458**	**0.4121**	**3.9245**	**2.0934**	201.5932	**0.3335**
Mapillary Metropolis	PanoSUNCG	SfMLearner [6]	15.8735	11.2958	50.9814	45.9223	4.3297	0.4394
		MonoDepth2 [1]	16.4948	12.6478	48.0064	45.0384	4.3719	0.4409
		Bifuse++ [4]	16.5660	12.8155	8.8364	**4.6106**	**2.7913**	**0.3626**
		PanoDepth	**8.6318**	**2.2702**	**8.3023**	5.6434	3.6319	0.3755
PanoSUNCG	Building	SfMLearner [6]	1.3758	1.1867	173.8752	175.9671	0.6786	0.3492
		MonoDepth2 [1]	1.2936	1.0596	172.8793	174.0039	0.6297	0.3557
		Bifuse++ [4]	1.1450	0.9810	174.0099	174.8455	**0.3293**	**0.2519**
		PanoDepth	**0.7244**	**0.4339**	**93.0434**	**96.5969**	0.7734	0.5003
PanoSUNCG	Campus	SfMLearner [6]	0.8816	0.6729	**4.8551**	**2.5934**	5.6829	0.4365
		MonoDepth2 [1]	0.9145	0.7354	4.9385	2.7738	**5.5644**	**0.4159**
		Bifuse++ [4]	1.2086	0.8214	5.2916	3.2601	187.9531	0.3806
		PanoDepth	**0.6490**	**0.4597**	7.7042	4.4586	21.2679	0.5890
PanoSUNCG	Mapillary Metropolis	SfMLearner [6]	2.4873	1.2550	4.2543	2.3491	0.4294	0.3927
		MonoDepth2 [1]	2.4178	1.1610	4.1021	2.2534	0.4960	0.3524
		Bifuse++ [4]	2.6883	1.3865	**3.7817**	**2.0923**	**0.3199**	0.2807
		PanoDepth	**1.8237**	**0.6495**	3.9221	2.1309	0.4133	**0.2423**
Campus	Building	SfMLearner [6]	1.6728	0.7381	173.2588	173.8723	0.3869	0.3594
		MonoDepth2 [1]	1.5945	0.7138	174.43263	174.8271	0.3946	0.3618
		Bifuse++ [4]	1.8882	**0.6196**	174.9197	175.1363	**0.3078**	**0.2217**
		PanoDepth	**0.8816**	0.7100	**51.9322**	**49.8283**	0.4638	0.2928
Campus	PanoSUNCG	SfMLearner [6]	16.9816	13.0087	11.3654	5.9156	5.3394	0.4638
		MonoDepth2 [1]	16.5622	12.7941	10.1081	5.5776	5.1357	0.4332
		Bifuse++ [4]	16.4711	12.6300	**9.5208**	**5.1788**	5.3989	0.4140
		PanoDepth	**13.3785**	**7.2264**	72.7482	71.5257	**4.8017**	**0.3847**
Campus	Mapillary Metropolis	SfMLearner [6]	3.4874	0.9005	4.2564	**2.2837**	0.3956	0.3795
		MonoDepth2 [1]	3.3949	0.8472	4.0109	2.4365	0.3486	0.2977
		Bifuse++ [4]	3.2004	**0.6091**	**3.9176**	2.7633	0.2177	0.1740
		PanoDepth	**2.6575**	0.6915	115.4759	115.7581	**0.1117**	**0.0893**

Table 2. Generalization ability across different datasets. T. Dataset and E. Dataset represent the dataset for training and evaluating, respectively. The best results are shown in **bold**.

datasets. In these cases, PanoPose achieves the second-best median RRE. Regarding relative translation angle error (RTAE), it is observed that all methods, including PanoPose, generate large errors. In the Building dataset, the RTAE of competitive methods approaches 180°. Such a high error suggests that the direction of the predicted relative translation is almost inverse to the ground truth. PanoPose performs better than other methods on this dataset, but its RTAE is still relatively high. On the other hand, PanoPose generates large RTAE when training on Campus and evaluating on PanoSUNCG and Mapillary Metropolis. Focusing on the relative scale error (RSE), we observed that all methods yield large errors, except for PanoPose training on Campus and evaluating on Mapillary Metropolis (last row in Tab. 2). This indicates the poor generalization ability in relative translation scale estimation.

From Tab. 2, we can observe that the network generalizes better in estimating relative rotation than relative translation. In most experiments, relative rotation errors are lower than relative translation errors. We attribute this phenomenon to the relatively short baselines between the input images. In situations with limited baseline distances, tradi-

Dataset	Method	MRE ↓	RMSE ↓	RMSElog ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
PanoSUNCG	SfMLearner [6]	0.2379	0.6115	0.4920	0.7211	0.7808	0.8182
	MonoDepth2 [1]	0.2047	0.7313	0.2663	0.7908	0.9111	0.9564
	NonLocal-DPT [5]	0.1864	0.7010	0.2547	0.8127	0.9136	0.9567
	BiFuse++ [4]	**0.1176**	**0.4321**	**0.0790**	**0.8974**	**0.9546**	**0.9773**
	PanoPose	0.1578	0.5773	0.2051	0.8469	0.9341	0.9622
Metropolis	SfMLearner [6]	0.2478	6.2927	0.3264	0.6481	0.8654	0.9396
	MonoDepth2 [1]	0.2345	6.2108	0.3179	0.6508	0.8705	0.9430
	NonLocal-DPT [5]	0.2546	6.8725	0.3521	0.5933	0.8387	0.9281
	BiFuse++ [4]	**0.1999**	**5.6440**	**0.2773**	**0.7309**	**0.9047**	**0.9586**
	PanoPose	0.2068	5.7837	0.2871	0.7173	0.8965	0.9550

Table 3. Depth estimation error. The best result is shown in **bold**, and the second-best is shown with under line. ↑ indicates that the higher the result, the better. ↓ is just the opposite.

tional geometric methods also exhibit a similar trend, where relative rotation estimation tends to be more accurate than relative translation. This can be explained by the fact that the reprojection error of 3D points is inherently more sensitive to changes in rotation, prompting a greater emphasis on optimizing rotation parameters during the pose optimization process. The process of utilizing a network to estimate relative pose can be viewed as the alignment of two images, which is also more sensitive to rotation. Consequently, the neural network exhibits superior performance in estimating relative rotation as compared to relative translation. Our PanoPose has better relative rotation estimation than other competitive methods, and we attribute it to the larger receptive field of transformers than CNN and the proposed rotation-only pre-training strategy. Generally, we believe that the current network for relative pose estimation based on self-supervision does not generalize well. To address this limitation, augmenting the training dataset with additional data may prove to be a crucial step.

3.3. Depth Estimation Result

Our PanoPose consists of a pose-net and a depth-net. While most of our experiments focus on pose estimation, in this section, we compare our depth estimation result against other self-supervised methods on PanoSUNCG and Mapillary Metropolis. We use the standard depth evaluation protocols, including mean relative error (MRE), root mean square error (RMSE), root mean square log error (RMSElog), and relative accuracy measures (δ). Before evaluation, the estimated depth map is aligned with ground truth using the median depth, which can be expressed as

$$\hat{D}_{pred} = D_{pred} \cdot \frac{Med(D_{gt})}{Med(D_{pred})} . \tag{6}$$

Here, D_{pred} and D_{gt} are the predicted depth map and ground truth, $\hat{D}_{pred}$ is the aligned predicted depth, and $Med(\cdot)$ is computing the median value of a matrix. The depth evaluation result is summarized in Tab. 3. From the table, it is clear that BiFuse++ exhibits the best depth estimation performance in both datasets and our PanoPose achieves the second-best result. Comparing Tab. 1 in the main text and Tab. 3 in the supplementary material, we can observe that the accuracy of relative pose estimation and depth estimation do not improve simultaneously. Across the experiment datasets, BiFuse++ reduces depth error by 15% compared to PanoPose, but PanoPose reduces relative rotation and relative translation angle errors by 80% and 62% compared to BiFuse++. This indicates that PanoPose is slightly weaker in depth estimation compared to BiFuse++, but it significantly outperforms BiFuse++ in relative pose estimation. This phenomenon underscores the effectiveness of the pose-net structure we have designed and the rotation only pre-training strategy in enhancing the accuracy of scaled relative pose estimation, which is a crucial aspect of the SfM problem.

3.4. Different Depth-net

In our proposed PanoPose, the depth-net is based on a ResNet-18 backbone. In this experiment, we explore the impact of different backbone architectures on the depth-net and assess their performance in terms of depth estimation and relative pose estimation accuracy. The results are summarized in Tab. 4. For this experiment, we consider four distinct ResNet architectures: ResNet-18, ResNet-34, ResNet-50, and ResNet-101. Furthermore, we explore the transformer-based depth-net architectures. Thus, PanoFormer[3], which is a network designed for panoramic image depth estimation, is selected as the depth-net. Additionally, we experiment with an alternative approach where we employ the Croco backbone network for both depth estimation and relative pose estimation simultaneously. In Tab. 4, this setup is denoted as "Croco". In essence, it dispenses with the use of an independent depth-net and instead relies on the features generated by Croco for both

Dataset	Backbone	MRE	Mean RRE	Med RRE	Mean RTAE	Med RTAE	Mean RSE	Med RSE
PanoSUNCG	ResNet-18	0.1578	**0.1559**	**0.0560**	**0.4253**	**0.2874**	**1.2295**	**0.0115**
	ResNet-34	0.1528	0.6736	0.3322	2.9858	1.8511	6.4084	0.0818
	ResNet-50	0.1689	0.2291	0.1740	0.6399	0.3877	7.6150	0.0154
	ResNet-101	0.6879	16.8719	12.7980	83.3537	84.0831	4.3481	0.3650
	PanoFormer[3]	**0.1400**	0.2731	0.1371	0.7220	0.4752	7.3984	0.0298
	Croco	0.6074	16.5305	12.7444	9.3661	5.3680	4.3150	0.3764
Mapillary Metropolis	ResNet-18	**0.2068**	**1.7228**	0.2683	**1.7661**	0.4006	**0.0217**	**0.0101**
	ResNet-34	0.5156	2.5370	0.2770	2.3079	0.5412	0.1074	0.0456
	ResNet-50	0.5681	2.6718	0.2987	2.2058	**0.3406**	0.0522	0.0384
	ResNet-101	0.3715	2.2884	**0.2273**	2.2953	0.5356	0.1061	0.0418
	PanoFormer[3]	0.6745	3.2925	0.7058	6.1243	4.6472	0.0389	0.0268
	Croco	0.6140	3.2656	0.6626	26.0129	11.6665	0.4056	0.3540
Building	ResNet-18	**0.1061**	**0.2009**	**0.1427**	**0.4653**	**0.3892**	**0.0935**	**0.0733**
	ResNet-34	0.4006	1.6412	0.4846	22.2070	1.4090	0.4566	0.3495
	ResNet-50	0.4858	2.1615	0.9891	3.1206	1.5235	0.4084	0.3238
	ResNet-101	0.4237	1.7336	0.6091	10.1709	1.5105	0.4240	0.3461
	PanoFormer[3]	0.5401	1.5849	0.4528	1.3362	0.7363	0.1231	0.0788
	Croco	0.5056	0.3956	1.7267	1.1607	0.4024	0.3749	0.4682
Campus	ResNet-18	**0.0979**	**0.1094**	**0.0862**	**2.2683**	**0.4644**	**0.2563**	**0.0519**
	ResNet-34	0.1965	0.8868	0.3986	5.1364	0.6287	0.6619	0.1136
	ResNet-50	0.2219	0.9300	0.4145	4.3781	0.6501	0.7187	0.1568
	ResNet-101	0.3791	0.6741	0.2712	8.9667	0.8924	1.1539	0.2517
	PanoFormer[3]	0.8724	1.0029	0.4178	2.5871	0.7609	8.9828	0.0803
	Croco	0.4592	0.8703	0.4481	2.5200	0.6685	2.0695	0.3740

Table 4. Depth estimation error and relative pose estimation error with different backbone. For brevity, only MRE (mean relative error) is used for depth evaluation. The best result is shown is **bold**.

depth prediction and pose estimation. From the table, it is clear that using ResNet-18 as the backbone yields the best result in most cases, except for the Mapillary Metropolis dataset. In terms of depth estimation evaluation (assessed by the MRE metric), from ResNet-18 to ResNet-101, the depth estimation error grows larger as the backbone becomes larger. Furthermore, introducing PanoFormer as the depth-net generally results in larger errors compared to ResNet-based networks, except for the PanoSUNCG dataset. It can be attributed to PanoFormer's training for indoor depth estimation, which aligns with the indoor environment of the PanoSUNCG dataset, yielding improved depth estimation results. Utilizing the Croco backbone for both depth and relative pose estimation is also infeasible. This configuration leads to significant errors, particularly in the PanoSUNCG dataset, and results in the second-worst depth estimation across the other three datasets. This experiment shows that as the network becomes more complex, the performance does not exhibit improvement but deteriorates. This can be attributed to the challenges of training larger networks and the optimization is easy to fall into local optima. Therefore, our depth-net uses the most lightweight ResNet-18 as the backbone, ensuring network convergence and reliable performance.

References

[1] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *IEEE International conference on Computer Vision (ICCV)*, pages 3828–3838, 2019. 2, 3, 4

[2] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(6):756–770, 2004. 1, 2

[3] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: Panorama transformer for indoor 360° depth estimation. In *European Conference on Computer Vision (ECCV)*, pages 195–211, 2022. 4, 5

[4] Fu-En Wang, Yu-Hsuan Yeh, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. Bifuse++: Self-supervised and efficient bi-projection fusion for 360 depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(5):5448–5460, 2022. 2, 3, 4

[5] Ilwi Yun, Hyuk-Jae Lee, and Chae Eun Rhee. Improving 360 monocular depth estimation via non-local dense prediction transformer and joint supervised and self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3224–3233, 2022. 2, 4

[6] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017. 2, 3, 4