HRDFuse: Monocular 360° Depth Estimation by Collaboratively Learning Holistic-with-Regional Depth Distributions –Supplementary Material–

Hao Ai[1] Zidong Cao[1] Yan-Pei Cao[2] Ying Shan[2] Lin Wang[1,3*]
[1]AI Thrust, HKUST(GZ) [2]ARC Lab, Tencent PCG [3]Dept. of CSE, HKUST
hai033@connect.hkust-gz.edu.cn, caozidong1996@gmail.com
caoyanpei@gmail.com, yingsshan@tencent.com, linwang@ust.hk

1. Abstract

Due to the lack of space in the main paper, we provide more details of the proposed method and experimental results in the supplementary material. Sec. 2 adds more details of tangent projection. Sec. 3 provides the detailed calculation progress of the Collaborative Depth Distribution Classification (CDDC) module. Sec. 4 introduces our loss function, and Sec. 5 presents a detailed description of the used benchmark datasets and metrics. In the Sec. 6 and Sec. 7, we show additional comparison results and visual results about experiments. Furthermore, we discuss the rationality of SFA module in the Sec. 8 and show some comparison results on real data in the Sec. 9.

2. More Details of Tangent Projection

We start by introducing an example of the tangent projection (TP) [1]. As shown in Fig. 1, P_s is a point on the sphere surface, O is the center of the sphere, P_c is the center of the tangent plane, and P_t is the intersection point of the tangent plane and the extension line of $\overrightarrow{OP_s}$. As both P_s and P_c are on the sphere surface, we represent their spherical coordinates as (θ_s, ϕ_s) and (θ_c, ϕ_c), respectively. Then, we can obtain the planar coordinate (u_t, v_t) of the point P_t on the tangent plane as follows:

$$\begin{aligned} u_t &= \frac{\cos(\phi_s)\sin(\theta_s-\theta_c)}{\cos(c)}, \\ v_t &= \frac{\cos(\phi_c)\sin(\phi_s)-\sin(\phi_c)\cos(\phi_s)\cos(\theta_s-\theta_c)}{\cos(c)}, \\ \cos(c) &= \sin(\phi_c)\sin(\phi_s)+\cos(\phi_c)\cos(\phi_s)\cos(\theta_s-\theta_c). \end{aligned} \tag{1}$$

And the inverse transformations are:

$$\begin{aligned} \theta_s &= \theta_c + \tan^{-1}\left(\frac{u_t\sin(\sigma)}{\gamma\cos(\phi_c)\cos(\sigma)-v_t\sin(\phi_c)\sin(\sigma)}\right), \\ \phi_s &= \sin^{-1}\left(\cos(\sigma)\sin(\phi_c)+\frac{1}{\gamma}v_t\sin(\sigma)\cos(\phi_c)\right), \end{aligned} \tag{2}$$

where $\gamma = \sqrt{u_t^2+v_t^2}$ and $\sigma = \tan^{-1}\gamma$. With Eq.1 and Eq. 2, we can convert the points on the sphere and pixels in TP patches to each other. In addition, we can convert the spherical points into pixels in the ERP image with $(u_e, v_e) = (\frac{\theta_s * w}{2\pi}, \frac{\phi_s * h}{\pi})$, where w and h are the width and height of the ERP image, respectively. Therefore, given the spherical coordinate of a TP patch center, we can achieve the mapping between the pixels in the ERP images and those in the corresponding TP patches.

The number of TP patches projected from a 360° spherical image depends on the sampling latitudes (the range of latitude is from -90° to 90°) and the sampling number at each latitude. For instance, in Omnifusion [12], TP patches are sampled

*Corresponding author (e-mail: linwang@ust.hk)

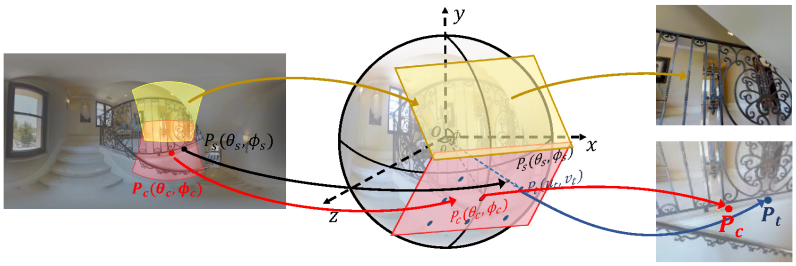


Figure 1. An example of TP and ERP. Two TP patches are projected from two different areas (red area and yellow area).

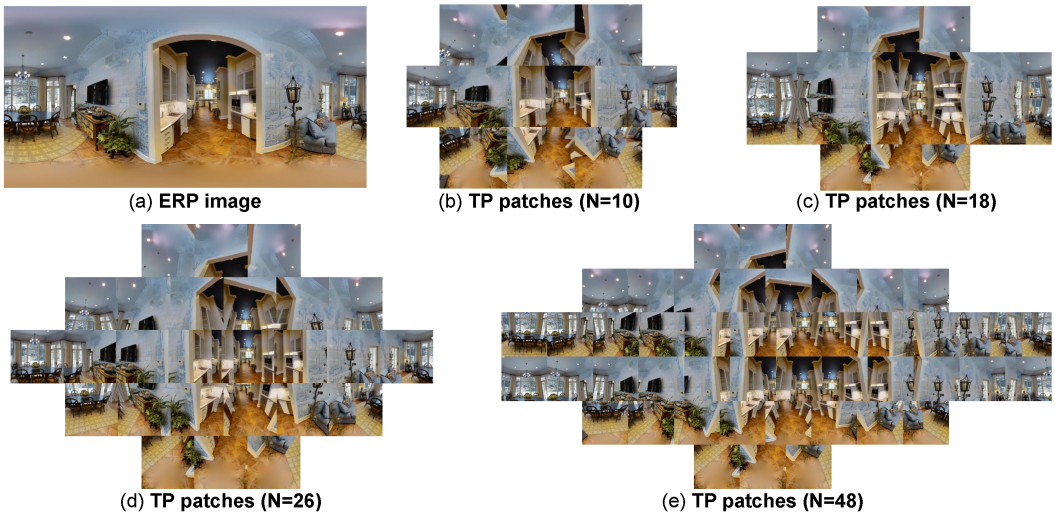


Figure 2. (a) An ERP image; (b) TP patches with the patch number $N = 10$, which are sampled at three latitudes; (c) TP patches with $N = 18$, which are sampled at four latitudes; (d) TP patches with $N = 26$, which are sampled at five latitudes; (e) TP patches with $N = 46$, which are sampled at six latitudes

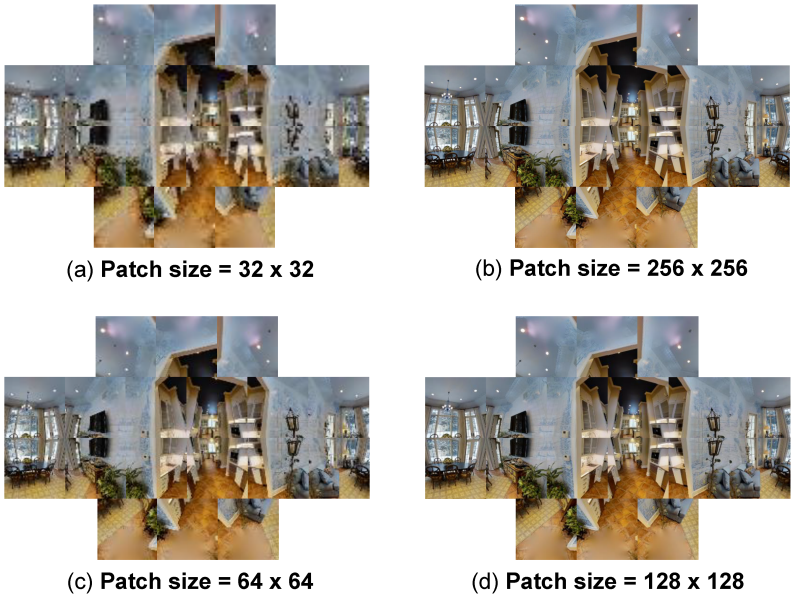


Figure 3. TP patches with different patch sizes.

from four latitudes: -67.5°, -22.5°, 22.5°, 67.5°, with 3, 6, 6, 3 patches on each latitude, respectively (see Fig 2c). Besides, for one more case, as shown in Fig 2d, the sampled latitudes can be set: -72.2°, -36.1°, 0°, 36.1°, 72.2°, while the sampled patch numbers are 3, 6, 8, 6, 3, respectively. From Fig 2, we can see that with the patch number increased, the area of the

Figure 4. TP patches with different patch FoVs.

overlapping regions increased correspondingly. As shown in Table. 2, too few patches can not provide sufficient regional structural information, while too many patches lead to the redundancy of regional information. As a result, we chose to use a relatively small patch number of 18.

We fix the patch FoV to 80° and compare TP patches with different patch sizes of 32×32, 64×64, 128×128, and 256×256 in Fig. 3, and it demonstrates that different patch sizes do not affect the content in each TP patch, but a large patch size does produce TP patches with more details. However, as shown in Table. 3 of the main paper, too large patch size will increase computational costs and the redundancy of regional structural information (the amount of pixels in the overlapping regions), which may further influence the prediction from holistic contextual information and decrease the overall performance. As a result, we chose to use a relatively large patch size of 128×128.

For the patch FoV, we fix the patch size to 128×128, and change the patch FoVs to obtain a set of TP patches, as shown in Fig. 4. Compared with the complete view of Fig. 2a, too small FoV causes the loss of the scene information, while too large FoV causes the redundancy of information in the overlapping areas. As a result, we chose to use patch FoV 80°.

3. More Details of Collaborative Depth Distribution Classification (CDDC)

In this section, we introduce the calculation process of the collaborative depth distribution classification (CDDC) module in detail.

First, given an ERP image with the size of $H_e \times W_e \times 3$, we follow the gnomonic projection to obtain N TP patches with the size of $H_t \times W_t \times 3$. Through the feature extractors, we can obtain the ERP feature map f^E with the size $H_e/2 \times W_e/2 \times C_e$ and TP feature maps $\left\{f_n^T\right\}, n = 1, \ldots, N$ with the size of $H_t/2 \times W_t/2 \times C_t \times N$, as the inputs of CDDC module. Then we summarize the detailed layer-by-layer network configurations in Table. 1. Especially, we introduce network configurations in four parts: holistic depth distribution classification, holistic depth prediction, regional depth distribution classification, and regional depth prediction.

In the holistic depth distribution classification, given the output of the transformer encoder, embedding tokens Tk_{out}^H, we select the first token $Tk_{out}^H[0]$ to calculate the bin center vector $\mathbf{c}^H$ as

$$\mathbf{c}_i^H = D_{min} + (\mathbf{w}_i^H/2 + \sum_{j=1}^{i-1} \mathbf{w}_j^H), \tag{3}$$

$$\mathbf{w}_i^H = (D_{max} - D_{min}) \frac{(\mathrm{mlp}(Tk_{out}^H[0]))_i + \epsilon}{\sum_{j=1}^{B} (\mathrm{mlp}(Tk_{out}^H[0]))_j + \epsilon}, \tag{4}$$

where $i, j = 1, \ldots, B$, $\mathbf{w}^H$ is the bin widths of the holistic distribution histogram, mlp denotes a multi-layer perceptron (MLP) head with a ReLU activation, (D_{min}, D_{max}) is the depth range of the dataset, B denotes the number of depth distribution bins, and ϵ is a small constant to ensure that each value of $\mathbf{w}^H$ is positive. For the holistic depth prediction, the bin centers $\mathbf{c}^H$ are linearly blended with a probability score map P^H to predict the depth value at each pixel (i, j):

$$D^H(i,j) = \sum_{b=1}^{B} P^H(i,j)_b \cdot \mathbf{c}_b^H. \tag{5}$$

For the regional depth distribution classification, as illustrated in the Table. 1, we collect regional depth bin center vectors from the collection of TP feature map $\left\{F_n^{\mathrm{TP}}\right\}$ and concatenate the center vectors to obtain the tensor $\mathbf{c}^R$ with the size of

Collaborative Depth Distribution Classification (CDDC)							
Input	InpRes	Kernel	Stride	Ch I/O	Opt.	OutRes	Output
Holistic Depth Distribution Classification							
F^{ERP}	$H_e/2 \times W_e/2 \times C_e$	8	8	C_e/C_1	Flatten	$(\frac{H_e * W_e}{256}) \times C_1$	Tk_{in}^H
Tk_{in}^H	$(\frac{H_e * W_e}{256}) \times C_1$	-	-	C_1/C_1	Transformer Encoder	$(\frac{H_e * W_e}{256}) \times C_1$	Tk_{out}^H
$Tk_{out}^H[0]$	$1 \times C_1$	-	-	C_1/B	Eq. 3, Eq. 4	$1 \times B$	$\mathbf{c}^H$
Holistic Range Attention Map							
F^{ERP}	$H_e/2 \times W_e/2 \times C_e$	3	1	C_e/C_1		$H_e/2 \times W_e/2 \times C_1$	F^H
F^H & $Tk_{out}^H[1:C_2+1](T^H)$	$H_e/2 \times W_e/2 \times C_1$ & $C_2 \times C_1$	- -	- -	- -	$\odot$	$H_e/2 \times W_e/2 \times C_2$	R^H
$\mathcal{R}^H$	$H_e/2 \times W_e/2 \times C_2$	-	-	-	Up-sample	$H_e \times W_e \times C_2$	$\mathcal{R}^{H'}$
$\mathcal{R}^{H'}$	$H_e \times W_e \times C_2$	1	1	C_2/B	Softmax	$H_e \times W_e \times B$	P^H
Holistic Depth Prediction							
$\mathbf{c}^H \& P^H$	$1 \times B \& H_e \times W_e \times B$	-	-	$B/1$	Eq. 5	$H_e \times W_e \times 1$	D^H
Regional Depth Distribution Classification							
$\left\{F_n^{\mathrm{TP}}\right\}$	$\frac{H_t}{2} \times \frac{W_t}{2} \times C_t \times N$	4	4	C_t/C_1	Flatten	$(\frac{H_t * W_t}{64}) \times C_1 \times N$	Tk_{in}^R
Tk_{in}^R	$(\frac{H_t * W_t}{64}) \times C_1 \times N$	-	-	C_1/C_1	Transformer Encoder	$(\frac{H_t * W_t}{64}) \times C_1 \times N$	Tk_{out}^R
$Tk_{out}^R[0]$	$1 \times C_1 \times N$	-	-	C_1/B	Similar to Eq. 3, Eq. 4	$1 \times B \times N$	$\mathbf{c}^R$
$\mathbf{c}^R \& M$	$1 \times B \times N \& \frac{H_e}{2} \times \frac{W_e}{2} \times N$	-	-	-	Eq. 6	$\frac{H_e}{2} \times \frac{W_e}{2} \times B$	M_c
Regional Range Attention Map							
$Tk_{out}^R[1:C_2+1]$	$C_2 \times C_1 \times N$	-	-	-	Mean	$C_1 \times N$	T^R
$T^R \& M$	$C_1 \times N \& \frac{H_e}{2} \times \frac{W_e}{2} \times N$	-	-	-	Similar to Eq. 6	$\frac{H_e}{2} \times \frac{W_e}{2} \times C_1$	M_{key}
$M_{key} \& T^H$	$\frac{H_e}{2} \times \frac{W_e}{2} \times C_1 \& C_2 \times C_1$	-	-	-	$\odot$	$\frac{H_e}{2} \times \frac{W_e}{2} \times C_2$	R^R
R^R	$\frac{H_e}{2} \times \frac{W_e}{2} \times C_2$	-	-	-	Up-sample	$H_e \times W_e \times C_2$	$R^{R'}$
$R^{R'}$	$H_e \times W_e \times C_2$	1	1	C_2/B	Softmax	$H_e \times W_e \times B$	P^R
Regional Depth prediction							
M_c	$\frac{H_e}{2} \times \frac{W_e}{2} \times B$	-	-	-	Up-sample	$H_e \times W_e \times B$	$M_c^{'}$
$M_c^{'} \& P^R$	$H_e \times W_e \times B \& H_e \times W_e \times B$	-	-	$B/1$	Similar to Eq. 6	$H_e \times W_e \times 1$	D^R

Table 1. Network summary of the CDDC module ($\odot$ denotes the dot-production).

$B \times N$. Moreover, with the spatial guidance of index map M, we can obtain an ERP format bin center map M_c based on $\mathbf{c}^R$ as follows:

$$M_c(i,j) = \sum_{n=1}^{N} M(i,j)_n \cdot \mathbf{c}_n^R \tag{6}$$

where (i, j) is the pixel coordinate, and n is the patch index. The bin center map M_c represents the depth distribution of each pixel with the regional structural information. Meanwhile, we concatenate the collection of selected tokens and reduce the first dimension of the concatenation with the average operation, to obtain the tensor T^R. Then we combine T^R with the spatial locations of index map M to obtain a feature map M_{key}. Moreover, we introduce the embedding vectors T^H of the ERP branch. With M_{key} as the "keymap" and T^H as the "queries", we can predict the probability score map P^R and further output the ERP format regional depth map D^R.

4. More Details of Loss Functions

As introduced in the main paper, our loss consists of two terms: the pixel-wise depth loss and the holistic distribution loss. For the pixel-wise depth loss, following existing works [9, 12], we adopt BerHu loss [10] for pixel-wise depth supervision, which is formulated as

$$\mathcal{L}_{depth} = \sum_{i \in P} \mathcal{B}(D^i - D_{GT}^i), \tag{7}$$

$$\mathcal{B}(x) = \begin{cases} |x|, |x| \leq c \\ \dfrac{x^2 + c^2}{2c}, |x| > c \end{cases} \tag{8}$$

where D_{GT} is the ERP format ground truth, P indicates pixels which are valid in the ground truth depth map. c is a threshold hyper-parameter and set to 0.2 empirically [9, 12].

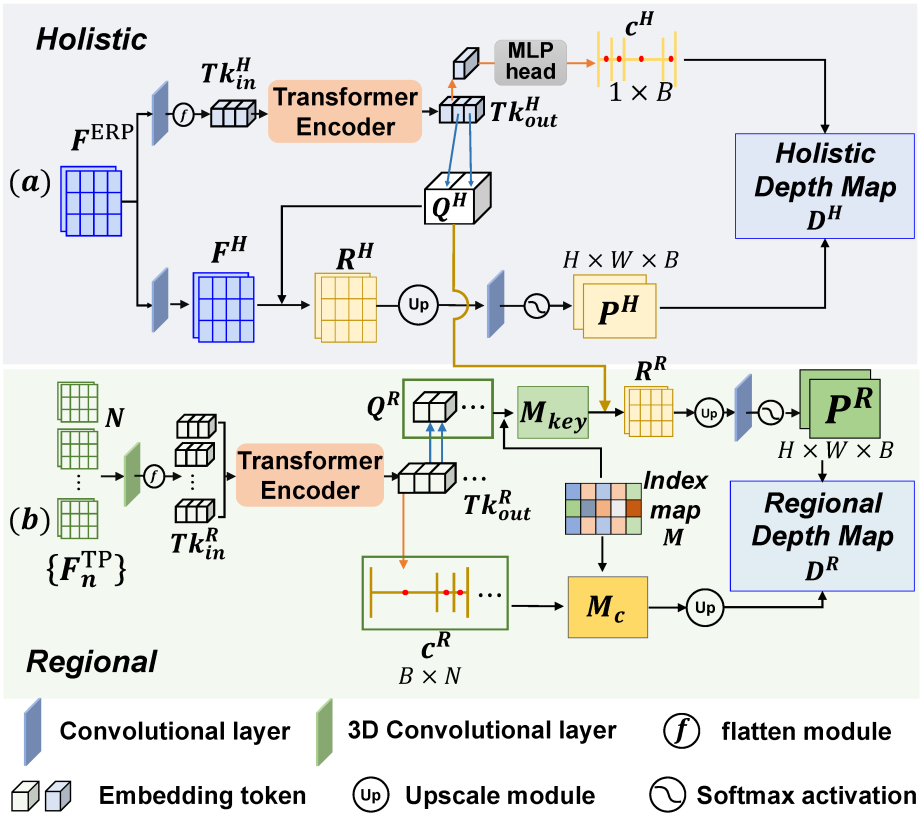


Figure 5. Overview of the CDDC module with two steps.

Furthermore, following [3], we employ the bi-directional Chamfer Loss [7] to encourage the holistic depth bin centers $\mathbf{c}^H(b)$ to be consistent with the distribution of all depth values (X) in the ground truth map as:

$$\mathcal{L}_{H_{bin}} = Cha(X, \mathbf{c}^H(b)) \tag{9}$$

$$Cha(X_1, X_2) = \sum_{x \in X_1} \min_{y \in X_2} \|x - y\|_2^2 + \sum_{y \in X_2} \min_{x \in X_1} \|x - y\|_2^2 \tag{10}$$

Finally, the total loss is the summation of both two terms:

$$\mathcal{L}_{total} = \mathcal{L}_{depth} + \lambda \mathcal{L}_{H_{bin}}. \tag{11}$$

For the balance weight λ, we follow [3] and set $\lambda = 0.1$ for all our experiments.

5. More Details of Datasets and Metrics

We conduct experiments on three benchmark datasets: Stanford2D3D [2], Matterport3D [4] and 3D60 dataset [16]. Note that Stanford2D3D dataset and Matterport3D dataset are real-world datasets, while 3D60 dataset is composed of two synthetic datasets (SunCG [14] and SceneNet [8]) and two real-world datasets (Stanford2D3D and Matterport3D). Stanford2D3D contains 1413 panoramic samples and we split it into 1,000 samples for training, 40 samples for validation and 373 samples for testing. Matterport3D is the largest real-world dataset for indoor panorama scenes containing 10,800 panoramas and we follow the official split to split it into 33875 samples for training, 800 samples for validation, and 1298 samples for testing. As the largest 360° depth estimation dataset, 3D60 totally contains 35973 panoramic samples where 33875 of them are used for training, 800 samples for validation, and 1298 samples for testing. During training and testing, we resize the resolution of the panorama and depth map in the former two datasets into 512×1024. For 3D60, we set the input size into 256×512.

6. Additional Comparison Results

As shown in the open source code of PanoFormer [13], the authors applied the masking strategy for the Stanford2D3D dataset:

```
mask = torch.ones([512, 1024])
mask[0:int(512*0.15), :] = 0
mask[512-int(512*0.15):512, :] = 0
```

Therefore, we apply the same masking strategy for the Stanford2D3D dataset and compare with PanoFormer in Table. 2. With the masking strategy, our HRDFuse outperforms PanoFormer [13] by a significant margin, *e.g.*, 5.8% (Abs Rel), 11.3% (Sq Rel), 5.3% (RMSE). Furthermore, we compare our method with the PanoFormer on 3D60 dataset in Table. 3. Note that PanoFormer did not provide the pre-trained models on the 3D60 dataset, we re-train the PanoFormer for 60 epochs with the official hyper-parameters and same experiment setting as OmniFusion and UniFuse. Our HRDFuse outperforms PanoFormer by a large margin.

Datasets	Method	Patch size/FoV	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE(log) ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Stanford2D3D	PanoFormer* [13]	−/−	0.1131	0.0723	0.3557	0.2454	0.8808	0.9623	0.9855
	PanoFormer† [13]	−/−	0.0721	0.0506	0.3187	0.1949	0.9260	0.9766	0.9922
	HRDFuse*,Ours	128 × 128 / 80°	0.0984	0.0530	0.3452	0.1465	0.8941	0.9778	0.9923
	HRDFuse†,Ours	128 × 128 / 80°	0.0730	0.0469	0.3265	0.1311	0.9213	0.9807	0.9934
	HRDFuse*,Ours	256 × 256 / 80°	0.0935	0.0508	0.3106	0.1422	0.9140	0.9798	0.9927
	HRDFuse†,Ours	256 × 256 / 80°	**0.0679**	**0.0449**	**0.3017**	**0.1271**	**0.9327**	**0.9826**	**0.9935**

Table 2. Quantitative comparison with the SOTA methods. $*$ represents that the model is re-trained following the official setting. † represents that the model is evaluated with the masking strategy in PanoFormer [13].

Datasets	Method	Patch size/FoV	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE(log) ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
3D60	FCRN [11]	−/−	0.0699	0.2833	-	-	0.9532	0.9905	0.9966
	RectNet [16]	−/−	0.0702	0.0297	0.2911	0.1017	0.9574	0.9933	0.9979
	Mapped Convolution [6]	−/−	0.0965	0.0371	0.2966	0.1413	0.9068	0.9854	0.9967
	BiFuse with fusion [15]	−/−	0.0615	-	0.2440	-	0.9699	0.9927	0.9969
	UniFuse with fusion [9]	−/−	0.0466	-	0.1968	-	0.9835	0.9965	0.9987
	ODE-CNN [5]	−/−	0.0467	0.0124	0.1728	0.0793	0.9814	0.9967	0.9989
	OmniFusion (1-iter) [12]	128 × 128 / 80°	0.0469	0.0127	0.1880	0.0792	0.9827	0.9963	0.9988
	OmniFusion (2-iter) [12]	128 × 128 / 80°	0.0430	0.0114	0.1808	0.0735	0.9859	0.9969	0.9989
	PanoFormer* [13]	−/−	0.0442	0.0124	0.1691	0.0676	0.9861	0.9966	0.9987
	HRDFuse,Ours	128 × 128 / 80°	0.0363	0.0103	0.1565	0.0594	0.9888	**0.9974**	0.9990
	HRDFuse,Ours	256 × 256 / 80°	**0.0358**	**0.0100**	**0.1555**	**0.0592**	**0.9894**	0.9973	**0.9990**

Table 3. Quantitative comparison with the SOTA methods. $*$ represents that the model is re-trained following the official setting.

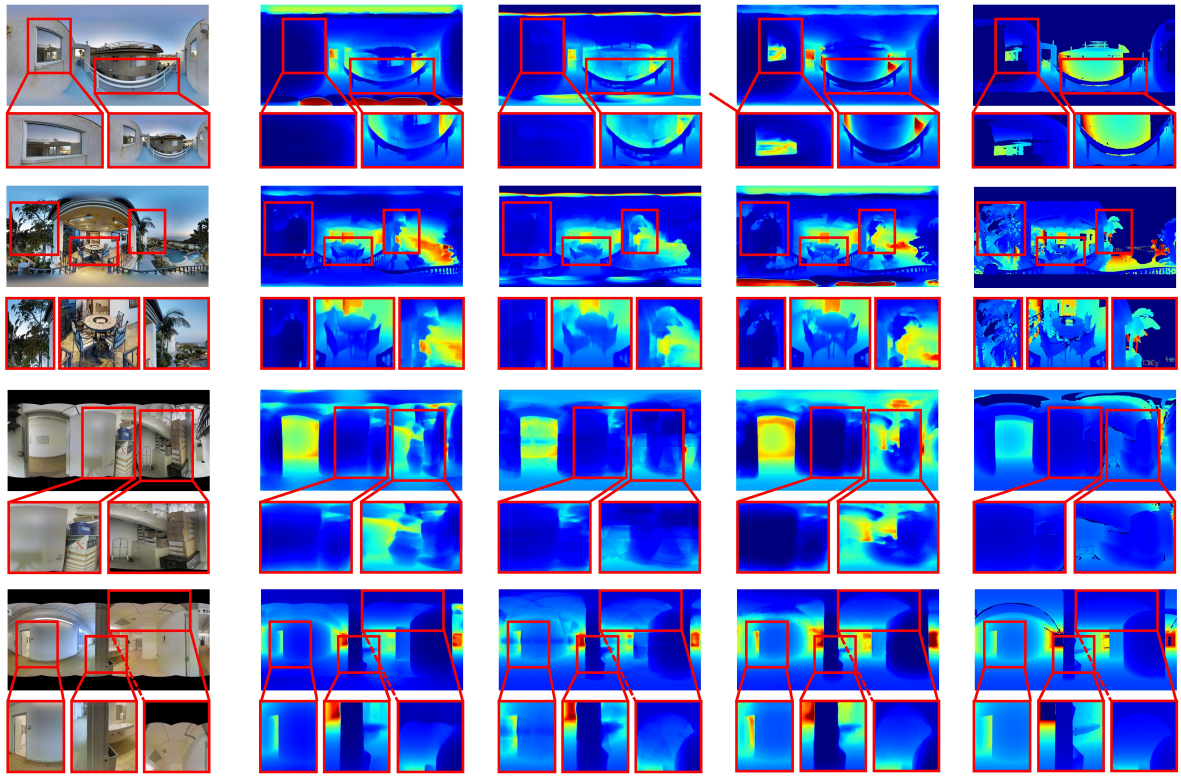
7. Additional Visual Results

More visual comparisons on Stanford2D3D and Matterport3D. In Fig. 6a, we perform qualitative comparisons with the SOTA methods, UniFuse. [9] and OmniFusion [12], on the Stanford2D3D dataset and Matterport3D dataset, whose samples are from real-world scenes. From the visual results, we confirm that our HRDFuse predicts the depth maps which are more precise and contain more structural details than other methods.

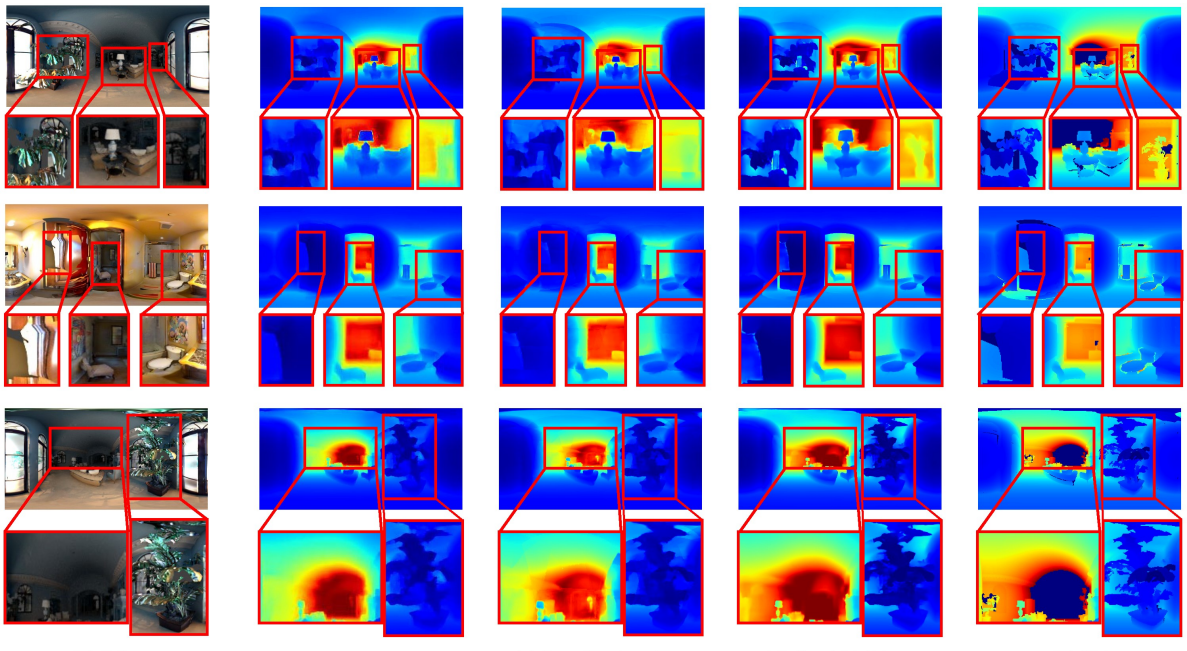
More visual comparisons on 3D60. In Fig. 6b, we perform qualitative comparisons with the SOTA methods, UniFuse. [9] and OmniFusion [12], on the 3D60 dataset, which contains both real-world and synthetic samples. From the visual results, we further confirm the superiority of our HRDFuse.

8. Discussion on the rationality of SFA module

As shown in the Fig .7, for a scene with simple structure, our SFA module make the index map centralized to several representative TP patches with higher frequency of index, *e.g.*, 4, 6, 10 (Fig. 7(c)). Combined with the Fig. 7(d), we can observe that the feature alignment based on the feature similarity in the SFA module tends to employ the most representative



(a) Visual comparisons on Stanford2D3D and Matterport3D.



(b) Visual comparisons on 3D60 dataset.

Figure 6. More visual comparison results.

regional depth distributions to avoid the redundant usage. Meanwhile, facing the special depth values, SFA module will

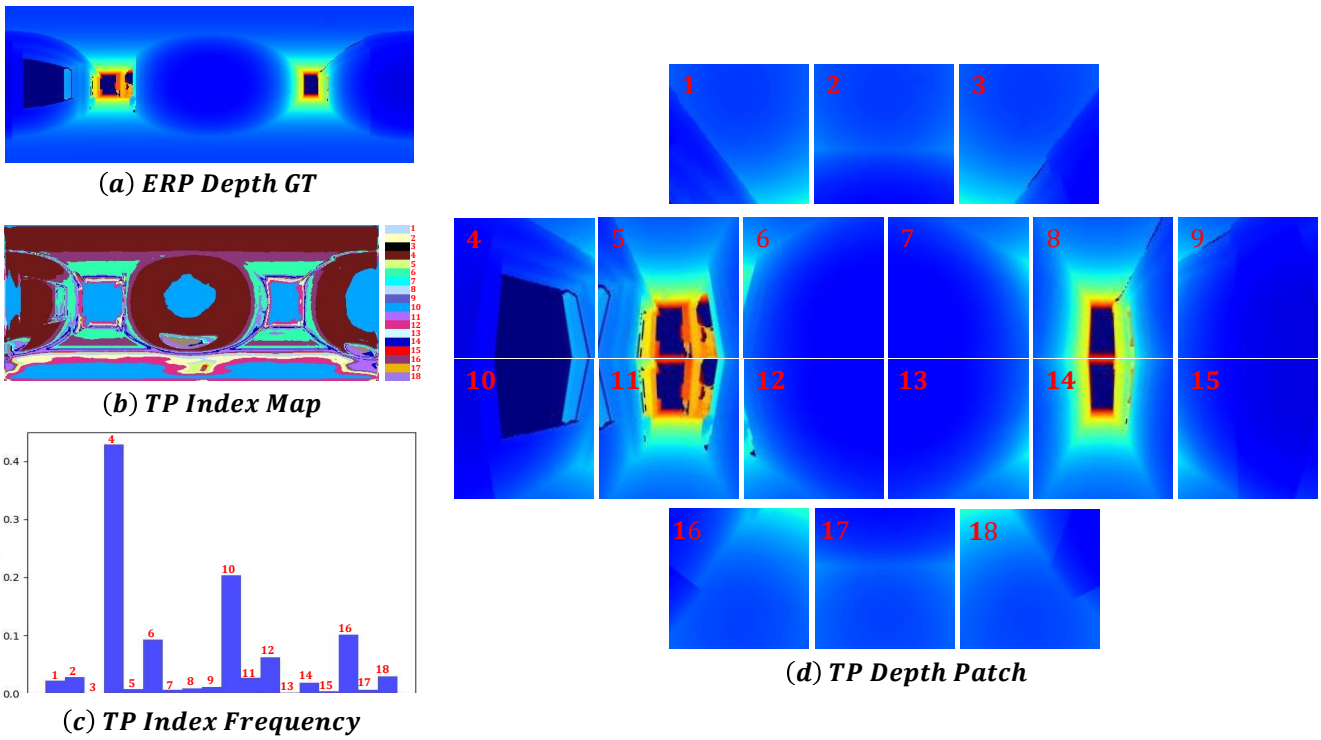


Figure 7. The visualization of (a) ERP depth ground truth, (b) TP index map (colored according to the attached color card), (c) TP index frequency and (d) TP depth patches with the corresponding index numbers from a scene with simple structure.

introduce the corresponding the regional depth distributions to predict them (*e.g.*, with index 11, 14). Especially, with the scene structure becoming more complex, the more TP patches are needed to describe the holistic depth information, as shown in the Fig. 8. The frequency of TP index is more balanced.

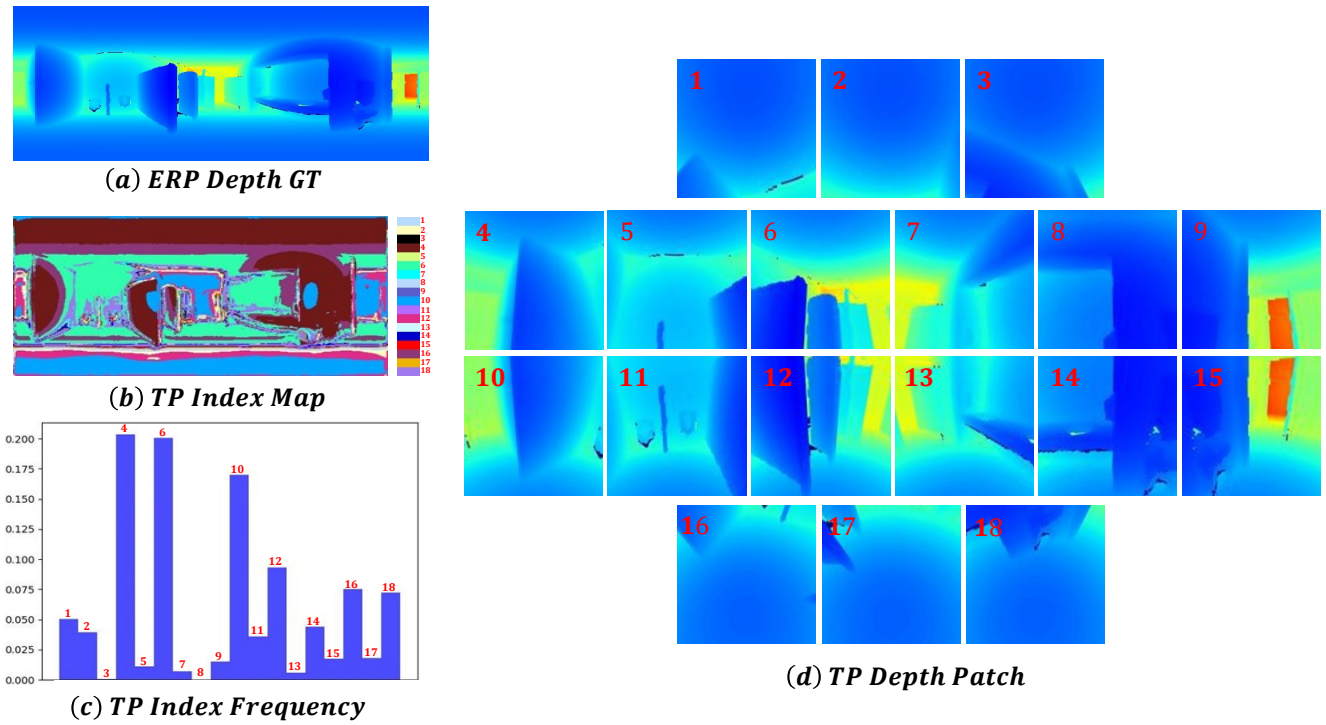


Figure 8. The visualization of (a) ERP depth ground truth, (b) TP index map (colored according to the attached color card), (c) TP index frequency and (d) TP depth patches with the corresponding index numbers from a scene with complex structure.

9. Visual comparisons and discussion on real data.

To better compare the generation capability of our HRDFuse and other SoTA methods, we capture the two real images which records the indoor scene (considering the limited max depth value, we ignore the outdoor scene) and directly use the models trained on Matterport3D training dataset to predict their depth maps. As shown in the Fig. 9, we can observe that our HRDFuse predicts more precise depth maps for the captured scenes. By contrast, the results of PanoFormer [13] tend to be blurry and over-smooth on unseen scenes.

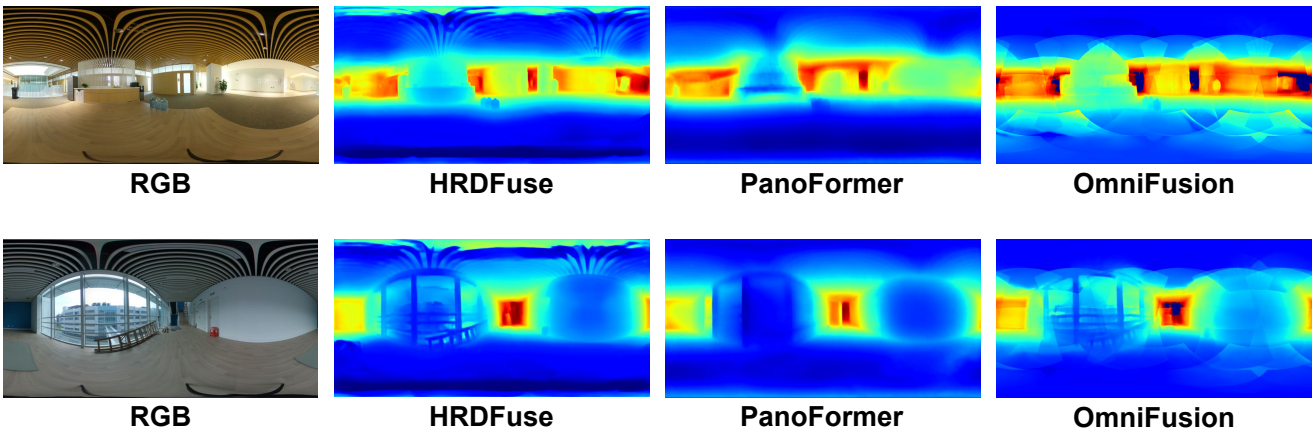


Figure 9. Visual comparisons on real data (captured by Ricoh Theta Z1).

References

[1] Hao Ai, Zidong Cao, Jinjing Zhu, Haotian Bai, Yucheng Chen, and Lin Wang. Deep learning for omnidirectional vision: A survey and new perspectives. *CoRR*, abs/2205.10468, 2022. 1

[2] Iro Armeni, Sasha Sax, Amir Roshan Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017. 5

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, pages 4009–4018. Computer Vision Foundation / IEEE, 2021. 5

[4] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *3DV*, pages 667–676. IEEE Computer Society, 2017. 5

[5] Xinjing Cheng, Peng Wang, Yanqi Zhou, Chenye Guan, and Ruigang Yang. Omnidirectional depth extension networks. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 589–595, 2020. 6

[6] Marc Eder, True Price, Thanh Vu, Akash Bapat, and Jan-Michael Frahm. Mapped convolutions. *ArXiv*, abs/1906.11096, 2019. 6

[7] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *computer vision and pattern recognition*, 2016. 5

[8] Ankur Handa, Viorica Patraucean, Simon Stent, and Roberto Cipolla. Scenenet: An annotated model generator for indoor scene understanding. *international conference on robotics and automation*, 2016. 5

[9] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. Unifuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics Autom. Lett.*, 6(2):1519–1526, 2021. 4, 6

[10] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *international conference on 3d vision*, 2016. 4

[11] Iro Laina, C. Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248, 2016. 6

[12] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. *CoRR*, abs/2203.00838, 2022. 1, 4, 6

[13] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: Panorama transformer for indoor 360° depth estimation. In *ECCV*, 2022. 6, 10

[14] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *arXiv: Computer Vision and Pattern Recognition*, 2016. 5

[15] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *CVPR*, pages 459–468. Computer Vision Foundation / IEEE, 2020. 6

[16] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *ECCV (6)*, volume 11210 of *Lecture Notes in Computer Science*, pages 453–471. Springer, 2018. 5, 6