HRDFUSE: MONOCULAR 360° DEPTH ESTIMATION BY COLLABORATIVELY LEARNING HOLISTIC-WITH-REGIONAL DEPTH DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

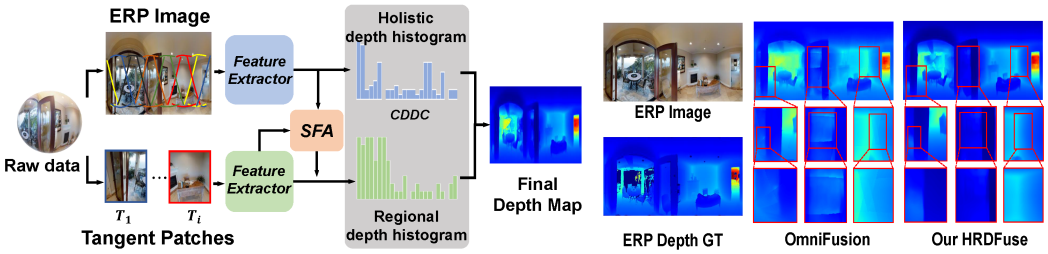
Depth estimation from a monocular 360° image is a burgeoning problem as a 360° image provides holistic sensing of a scene with a wide field of view. Recently, some methods, *e.g.*, OmniFusion, have applied the tangent projection (TP) to represent a 360° image and predicted depth values via patch-wise regressions, which are merged to get a depth map with equirectangular projection (ERP) format. However, these methods suffer from 1) non-trivial process of merging a large number of patches; 2) less smooth and accurate depth results caused by ignoring the holistic contextual information contained only in the ERP image and directly regressing the depth value of each pixel. In this paper, we propose a novel framework, **HRDFuse**, that subtly combines the potential of convolutional neural networks (CNNs) and transformers by collaboratively learning the *holistic* contextual information from the ERP and the *regional* structural information from the TP. Firstly, we propose a spatial feature alignment (**SFA**) module that learns feature similarities between the TP and ERP to aggregate the TP features into a complete ERP feature map in a pixel-wise manner. Secondly, we propose a collaborative depth distribution classification (**CDDC**) module that learns the **holistic-with-regional** histograms capturing the ERP and TP depth distributions. As such, the final depth values can be predicted as a linear combination of histogram bin centers. Lastly, we adaptively combine the depth predictions from two projections to obtain the final depth map. Extensive experiments on three benchmark datasets show that *our method achieves more smooth and accurate depth results while favorably surpassing the SOTA methods by a significant margin*.

1 INTRODUCTION

The 360° camera is becoming increasingly popular as a 360° image provides holistic sensing of a scene with a wide field of view (FoV) Ai et al. (2022). Therefore, the ability to infer the 3D structure of a 360° camera's surroundings has sparked the research for monocular 360° depth estimation Wang et al. (2020). Generally, raw 360° images are transmitted into 2D planar representations while preserving the omnidirectional information Yoon et al. (2022). Equirectangular projection (ERP) is the most commonly used projection format and can provide a complete view of a scene. Cubemap projection (CP) Cheng et al. (2018) projects 360° contents into six discontinuous faces of a cube to reduce the distortion; thus, the pre-trained 2D convolutional neural networks (CNNs) can be applied. However, ERP images suffer from severe distortions in the polar regions, while CP patches are hampered by geometric discontinuity and limited FoV.

For this reason, some works Zioulis et al. (2018); Zhuang et al. (2021) have proposed distortion-aware convolution filters to tackle the ERP distortion problem for depth estimation. BiFuse Wang et al. (2020) and UniFuse Jiang et al. (2021) explore the complementary information from the ERP image and CP patches to predict the depth map.

Recently, research has shown that it is promising to use tangent projection (TP) because TP patches have less distortion, and many pre-trained CNN models designed for perspective images can be directly applied Eder et al. (2020). Accordingly, 360MonoDepth Rey-Area et al. (2021) predicts the patch-wise depth maps from a set of TP patches using the state-of-the-art (SOTA) perspective depth estimators, which are aligned and merged to obtain an ERP format depth map. OmniFusion Li et al. (2022) proposes a framework leveraging CNNs and transformers to predict depth maps from



(a) The overview of our HRDFuse (b) Monocular 360° depth estimation

Figure 1: (a) Our HRDFuse employs the SFA module to align the regional information in discrete TP patches and holistic information in a complete ERP image. The CDDC module is proposed to estimate ERP format depth outputs from both the ERP image and TP patches based on holistic-with-regional depth histograms. (b) Compared with OmniFusion Li et al. (2022), our depth predictions are more smooth and more accurate.

the TP inputs and merges these patch-wise predictions to the ERP space based on geometric prior information to get the final depth output with ERP format. However, these methods suffer from two critical limitations because: 1) geometrically merging a large number of patches is computationally heavy; 2) they ignore the holistic contextual information contained only in the ERP image and directly regress the depth value of each pixel, leading to less smooth and accurate depth estimation results.

To tackle these issues, we propose a novel framework, called **HRDFuse**, that subtly combines the potential of convolutional neural networks (CNNs) and transformers by collaboratively exploring the ***holistic*** contextual information from the ERP and ***regional*** structural information from the TP (See Fig. 1(a) and Fig. 2). Compared with previous methods, our method achieves more smooth and more accurate depth estimation results while maintaining high efficiency with three key components. Firstly, for each projection, we employ a CNN-based feature extractor to extract spatially consistent feature maps and a transformer encoder to learn the depth distribution with long-range feature dependencies. In particular, to efficiently aggregate the individual TP information into an ERP space, we propose a spatial feature alignment (**SFA**) module to learn a spatially aligned index map based on feature similarities between ERP and TP. With this index map, we can efficiently measure the spatial location of each TP patch in the ERP space and achieve pixel-level fusion of TP information to obtain a smooth output in ERP format. Secondly, we propose a collaborative depth distribution classification (**CDDC**) module to learn the *holistic* depth distribution histogram from the ERP image and *regional* depth distribution histograms from the collection of TP patches. Consequently, the pixel-wise depth values can be predicted as a linear combination of histogram bin centers. Lastly, the final depth map is the adaptive fusion of two ERP format depth predictions from ERP and TP.

We conduct extensive experiments on three benchmark datasets: Stanford2D3D Armeni et al. (2017), Matterport3D Chang et al. (2017), and 3D60 Zioulis et al. (2018). The results show that our method can predict more smooth and more accurate depth results while favorably surpassing the existing methods by a significant margin (See Fig. 1(b) and Tab. 1). In summary, our main contributions are three-fold: (**I**) We propose HRDFuse that combines the holistic contextual information from the ERP and the regional structural information from the TP. (**II**) We introduce the SFA module to efficiently aggregate the TP features into the ERP format, relieving the need for expensive re-projection operations. (**III**) We propose the CDDC module to learn the holistic-with-regional depth distributions and estimate the depth value based on the histogram bin centers. (**IV**) Our experimental results on three benchmark datasets show that our method achieves ***new*** SOTA performance. *Our project code will be publicly available upon acceptance.*

2 Related Work

Monocular 360 Depth Estimation *ERP-based methods:* To address the spherical distortion in the ERP images, endeavours have been made to leverage the characteristics of convolutional filters. OmniDepth Zioulis et al. (2018) applies row-wise rectangular filters to cope with the distortions in different latitudes, while ACDNet Zhuang et al. (2021) leverages a group of dilated convolution

filters to rectify the receptive field. Tateno et al. Tateno et al. (2018) explored the standard convolution filters trained with the perspective images, and deformed the shape of sampling grids based on spherical distortion accordingly during the inference. SliceNet Pintore et al. (2021) partitions an ERP image into vertical slices and directly applies the standard convolutional layers to predict the ERP depth map. *Combination of CP and ERP:* BiFuse Wang et al. (2020) proposes to bidirectionally fuse the ERP and CP features at both encoding and decoding stages. By contrast, UniFuse Jiang et al. (2021) fuses the features only at the encoding stage as it is argued that ERP features are more important for final ERP format depth prediction. Differently, Bai et al. (2022) employs CNNs to extract ERP features and a transformer block Dosovitskiy et al. (2021) to extract CP features, which are fused to predict the final depth map. *TP-based methods:* TP is recently shown to suffer less from distortion, and the pre-trained CNN models designed for perspective images can be directly applied. Accordingly, 360MonoDepth Rey-Area et al. (2021) and OmniFusion Li et al. (2022) build their frameworks based on the TP patches. For more details, we refer readers to a recent survey Ai et al. (2022). Compared with these methods, our HRDFuse combines the potential of CNNs and transformers by collaboratively learning the holistic contextual information from the ERP image and regional structural information from the TP patches. Our method achieves new SOTA performance on three benchmark datasets.

Distribution-based Perspective Depth Estimation Many methods estimate depth by directly regressing the depth values; however, they suffer from slow convergence, and deficiency of global analysis Laina et al. (2016b); Lee et al. (2021). For this reason, Fu et al. (2018) discretized the depth range into several pre-determined intervals and recast depth prediction as an ordinal regression problem, which accounts the depth distributions depending on the located intervals. Adabins Bhat et al. (2021) divides the depth range into many adaptive bins whose widths are computed from the scene information, and the depth values are a linear combination of the bin centers, showing better performance over previous methods. Our HRDFuse is the *first* to explore the idea of depth distribution classification for 360° depth estimation. The proposed CDDC module learns the holistic depth distributions from the ERP image and regional depth distributions from the collection of TP patches. As such, the final depth values are predicted as a linear combination of bin centers.

Vision Transformer Transformers are capable of modeling the long-range dependencies for computer vision tasks Dosovitskiy et al. (2021). Recently, it has been shown that the combination of convolutional operations and self-attention mechanisms further enhance the representation learning. For instance, DeiT Touvron et al. (2021) employs a CNN as the teacher model to distill the tokens to the transformer, while DETR Carion et al. (2020) models the global relationship via serially feeding the features extracted by CNNs to the transformer encoder-decoder. Moreover, some works,*e.g.*, Peng et al. (2021); Chen et al. (2021) attempted to concurrently fuse the features from CNNs and transformers. Our HRDFuse framework is also built based on the combination of CNNs and transformers; however, it shares a different spirit as we focus on ensuring network efficiency. For this reason, we extract the high-resolution feature maps using a CNN-based encoder-decoder and feed them to a smaller transformer encoder Dosovitskiy et al. (2021) to estimate distributions.

3 METHODOLOGY

Overview. As depicted in Fig. 2, to exploit the complementary information from holistic context and regional strcuture, our framework simultaneously takes two projections of a 360° image, an ERP image and N TP patches, as inputs. For the ERP branch (See Fig. 2 Top), an ERP image with the resolution of $H \times W$ is fed into a feature extractor, comprised of an encoder-decoder block, to produce a decoded ERP feature map F^{ERP}. For the TP branch (See Fig. 2 Bottom), N TP patches are first obtained with gnomonic projection Eder et al. (2020). Then, these TP patches are passed through the TP feature extractor to obtain 1-D patch feature vectors $\{V_n, n = 1, \ldots, N\}$, which are passed through the TP decoder to obtain the TP feature maps $\left\{F_n^{\mathrm{TP}}, n = 1, \ldots, N\right\}$.

To determine and align the spatial location of each TP patch in the ERP space, we propose the spatial feature alignment (SFA) module (See Fig. 2) to learn feature correspondences between pixel vectors in the ERP feature map F^{ERP} and patch feature vectors $\{V_n\}$. This way, we can obtain the spatially aligned index map M, recording the location of each patch in the ERP space.

Next, the index map M, ERP feature map F^{ERP}, and TP feature maps $\left\{F_n^{\mathrm{TP}}\right\}$ are fed into the proposed collaborative depth distribution classification (CDDC) module that accordingly outputs

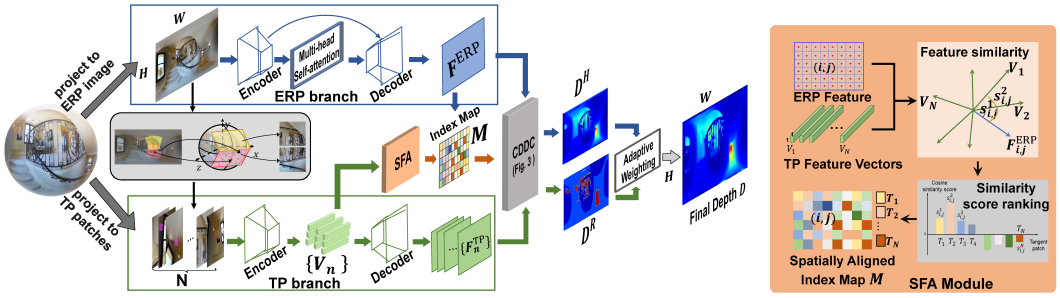


Figure 2: Overview of our HRDFuse, consisting of three parts: feature extractors for both ERP and TP inputs, spatial feature alignment (SFA) module, and collaborative depth distribution classification (CDDC) module (See Fig. 3 for details).

two ERP format depth predictions (See Fig. 3). In principle, the CDDC module first learns holistic-with-regional histograms to simultaneously capture depth distributions from the ERP image and a set of TP patches. Consequently, the depth distributions are then converted to depth values through a linear combination of bin centers. Lastly, the two depth predictions from the CDDC module are adaptively fused to output the final depth result. We now describe these modules in detail.

3.1 Feature Extraction

Overall, taking the ERP image and a collection of TP patches as inputs, the feature extractor of the ERP branch outputs the decoded feature map F^{ERP}, and the feature extractor of the TP branch produces encoded patch feature vectors $\{V_n\}$ and decoded TP feature maps $\left\{F_n^{\mathrm{TP}}\right\}$.

Specifically, for the ERP branch (Fig. 2 Top), we design the feature extractor with an encoder-decoder network, following the design of OmniFusion Li et al. (2022). It consists of an encoder built with the pre-trained ResNet34 He et al. (2016), a multi-head self-attention block Vaswani et al. (2017), and a decoder with commonly used up-sampling blocks. This way, we obtain the decoded feature map F^{ERP}.

For the TP branch, we first sample TP patches from the sphere via gnomonic projection Ai et al. (2022); Eder et al. (2020). *The details can be found in the appendix.* Secondly, we feed the patches simultaneously into the feature extractor, similar to the ERP branch but without the multi-head self-attention block, which helps to maintain the independence of each patch feature vector for spatial feature alignment. As such, we extract the patch feature vectors $\{V_n\}$ through the encoder and obtain the decoded patch feature maps $\left\{F_n^{\mathrm{TP}}\right\}$. The resolutions of the ERP feature map F^{ERP} and TP feature maps $\left\{F_n^{\mathrm{TP}}\right\}$ are set to half of the corresponding input resolutions for efficiency.

3.2 Spatial Feature Alignment

With ERP feature map F^{ERP} and patch feature vectors $\{V_n\}$, our SFA module outputs the spatially aligned index map M. *It determines the spatial relations between TP patches and pixel positions in the ERP space according to the feature similarity score ranking (See Fig. 2) and can be applied to achieve smooth pixel-wise fusion of individual TP information.* Existing works aggregate the discrete TP information into the complete ERP space via geometric fusion Li et al. (2022); Rey-Area et al. (2021). However, they are less capable of predicting smooth equirectangular depth outputs without holistic contextual information. For instance, as shown in Fig. 1(b), depth predictions in OmniFusion suffer from severe artifacts along the edges of the merged regions. For this reason, we propose the SFA module to measure, rank, and record the pixel-wise similarities between the ERP feature map F^{ERP} and patch feature vectors $\{V_n\}$. The pixel-wise similarity can be formulated as

$$s_{(i,j),k} = \frac{\overrightarrow{F^{\mathrm{ERP}}(i,j)} \cdot \overrightarrow{V_k}}{\|F^{\mathrm{ERP}}(i,j)\| \, \|V_k\|}, \tag{1}$$

where (i, j) is the coordinate of a pixel in the ERP feature map and k is the TP patch index. As depicted in Fig. 2, for each feature vector $F^{\mathrm{ERP}}(i, j)$ in the ERP feature map, our SFA module

calculates the cosine similarity score $s_{(i,j),k}$ between $F^{\mathrm{ERP}}(i,j)$ and each patch feature vector V_k. Then, it ranks the scores, selects the m-th patch that satisfies:

$$m = \arg\max_k s_{(i,j),k}, \tag{2}$$

and records the index m of the pixel location (i,j) on the spatially aligned index map M. For convenience, we extend each index into an N-dimension one-hot vector and transform the resolution size of index map M to $h_e \times w_e$, where (h_e, w_e) is the resolution size of ERP feature map F^{ERP}. Note that this spatially aligned index map is produced with the guidance of the holistic contextual information only contained in the ERP image. With this index map, we can efficiently aggregate the TP features into an ERP format feature map while maintaining spatial consistency.

3.3 Collaborative Depth Distribution Classification

The proposed CDDC module replaces the pixel-wise depth value regression with depth distribution classification, inspired by the works for perspective images Bhat et al. (2021); Fu et al. (2018). Importantly, to fully exploit the complete view in the ERP image and structural details in the less-distorted TP patches, we marry the potential of CNNs and transformers to learn the holistic-with-regional histograms capturing the ERP and TP depth distributions simultaneously.

In the following, we introduce our CDDC module in three parts: generic depth distribution classification, depth prediction based on the holistic depth distribution, and depth prediction based on the regional depth distributions.

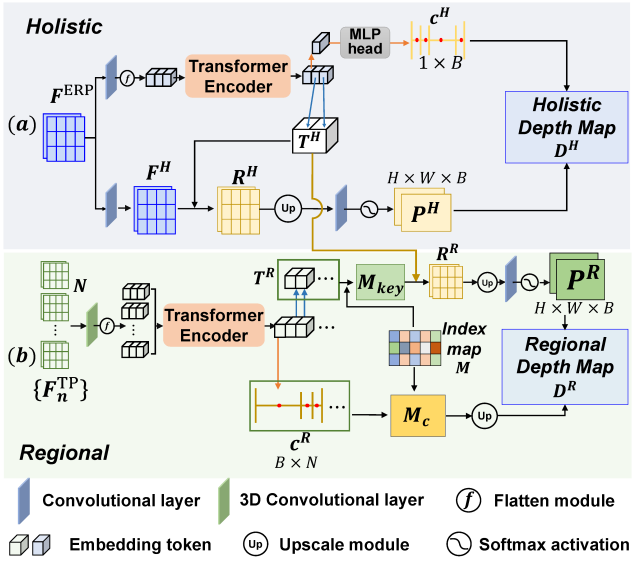


Figure 3: Overview of the CDDC module with two steps: depth distribution histogram classification, and depth prediction combination based on the range attention maps.

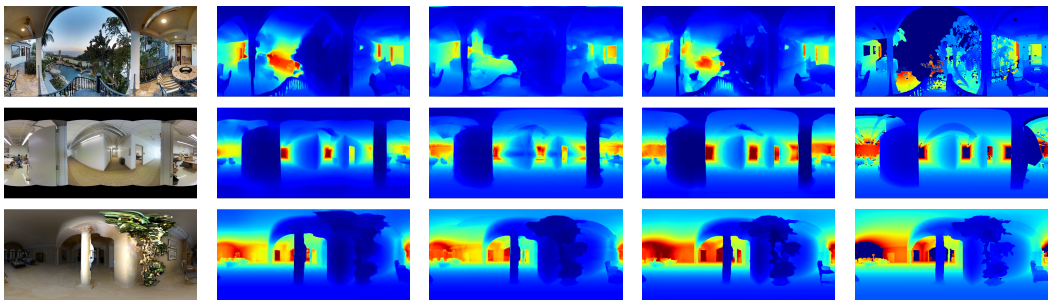
Generic depth distribution classification. Following previous works Bhat et al. (2021); Fu et al. (2018), given an extracted feature map $F \in \mathbb{R}^{H \times W \times C_{in}}$ (*e.g.*, F^{ERP} in Fig 3(a)), a sequence of embedding tokens T_{in} is obtained from F by a convolutional layer followed by a spatially flattening module. A transformer encoder then encodes the embedding tokens T_{in}, producing processed tokens T_{out}. Note that the processed tokens T_{out} now benefit from the global context and thus can accurately capture the depth distribution. Then the first token $T_{out}[0]$ from T_{out} is selected to predict the bin centers $\mathbf{c}$ of depth distribution histograms (*e.g.*, $\mathbf{c}^H$ in Fig 3(a)) as:

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

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

where $i, j = 1, \ldots, B$, $\mathbf{w}$ is the bin widths of the 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 $\mathbf{w}$ is positive. Finally, the bin centers $\mathbf{c}$ are linearly blended with a probability score map P (*e.g.*, P^H in the Fig 3(a)) to predict the depth value at each pixel (i,j):

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



(a) RGB (b) UniFuse (c) OmniFusion (d) HRDFuse(ours) (e) GT

Figure 4: Qualitative results on Matterport3D (top), Stanford2D3D (middle), and 3D60 (bottom).

We will detail the computation of P respectively in the following subsections.

Holistic distribution-based depth prediction. As depicted in Fig. 3a, we follow the process of generic depth distribution classification to predict the holistic depth bin centers $\mathbf{c}^H$. We then perform the following steps to obtain the holistic probability score map P^H. First, we select a part of processed tokens, which are the output of the transformer encoder and contain global context, as the "query" embedding T^H. At the same time, we encode a spatially consistent feature map F^H containing local pixel-wise information as the "keymap". Next, we calculate the dot-production between the query T^H and pixel features in F^H to obtain a range attention map R^H. This range attention map R^H thus contains global context and is spatially aligned with the ERP feature map. Then R^H is passed through a 1×1 convolutional layer with a softmax activation to predict the probability score map P^H. Given holistic depth bin centers and probability score map, we can now calculate the holistic depth map following Eq. 5. Note that the ERP feature map is with the half resolution of the input ERP image to limit GPU memory usage. Therefore, we additionally employ an up-sampling module to upscale the probability score map to the desired resolution (i.e., $H \times W$).

Regional distribution-based depth prediction. Compared with the ERP branch, predicting an ERP format depth map from TP patches based on corresponding regional depth distributions meets two critical difficulties: 1) accurate and smooth fusion of individual TP patches; 2) capturing the holistic information for the ERP format depth output. To address them, we utilize the spatially aligned index map M from the SFA module and the holistic query embedding T^H from the ERP branch (See Fig. 3b). We first follow the generic depth distribution classification to collect regional depth bin centers from the collection of TP feature maps $\{F_n^{\mathrm{TP}}\}$ and concatenate them to obtain the tensor $\mathbf{c}^R$ with the size $B \times N$. Then, with the spatial guidance of index map M, we can obtain an ERP format bin center map M_c from $\mathbf{c}^R$ as:

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

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 aggregated regional structural information.

Meanwhile, we concatenate and average a collection of processed regional tokens, which record the regional structural information of each individual TP patch, to a tensor T^R. Similarly, the index map M then helps to aggregate the regional structure in T^R to a regional feature map M_{key}. Next, with M_{key} as the "keymap" and T^H as the "query", we can predict the regional probability score map P^R and further output the ERP format regional depth map D^R. Note that the query embedding T^H from the ERP branch provides necessary and favorable holistic guidance. *Due to the page limit, more details can be found in the appendix.*

3.4 The Final Output and Loss Function

To obtain the final depth map, we adaptively fuse the depth prediction D^H from the holistic contextual branch and depth prediction D^R from the regional structural branch, which can be formulated as follows:

$$D = w_0 D^H + w_1 D^R, \tag{7}$$

Table 1: Quantitative comparison with the SOTA methods. $*$ represents that the model is re-trained following the official setting.

Datasets	Method	Patch size/FoV	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE(log) ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Stanford2D3D	FCRN Laina et al. (2016a)	−/−	0.1837	-	0.5774	-	0.7230	0.9207	0.9731
	RectNet Zioulis et al. (2018)	−/−	0.1996	-	0.6152	-	0.6877	0.8891	0.9578
	BiFuse with fusion Wang et al. (2020)	−/−	0.1209	-	0.4142	-	0.8660	0.9580	0.9860
	UniFuse with fusion Jiang et al. (2021)	−/−	0.1114	-	0.3691	-	0.8711	0.9664	0.9882
	OmniFusion (1-iter) Li et al. (2022)	256 × 256 / 80°	0.0961	0.0543	0.3715	0.1699	0.8940	0.9714	0.9900
	OmniFusion (2-iter) Li et al. (2022)	256 × 256 / 80°	0.0950	**0.0491**	0.3474	0.1599	0.8988	0.9769	0.9924
	HRDFuse,Ours	128 × 128 / 80°	0.0984	0.0530	0.3452	0.1465	0.8941	0.9778	0.9923
	HRDFuse,Ours	256 × 256 / 80°	**0.0935**	0.0508	**0.3106**	**0.1422**	**0.9140**	**0.9798**	**0.9927**
Matterport3D	FCRN Laina et al. (2016a)	−/−	0.2409	-	0.6704	-	0.7703	0.9714	0.9617
	RectNet Zioulis et al. (2018)	−/−	0.2901	-	0.7643	-	0.6830	0.8794	0.9429
	BiFuse with fusion Wang et al. (2020)	−/−	0.2048	-	0.6259	-	0.8452	0.9319	0.9632
	UniFuse with fusion Jiang et al. (2021)	−/−	0.1063	-	0.4941	-	0.8897	0.9623	0.9831
	OmniFusion (1-iter) Li et al. (2022)	256 × 256 / 80°	0.0980	0.0611	0.4536	0.1587	0.9040	0.9757	0.9919
	OmniFusion (2-iter) Li et al. (2022)	256 × 256 / 80°	0.0900	0.0552	0.4261	0.1483	0.9189	0.9797	0.9931
	OmniFusion (1-iter) *	256 × 256 / 80°	0.1054	0.0992	0.4548	0.1713	0.9061	0.9650	0.9834
	OmniFusion (2-iter) *	256 × 256 / 80°	0.1007	0.0969	0.4435	0.1664	0.9143	0.9666	0.9844
	HRDFuse,Ours	128 × 128 / 80°	**0.0967**	**0.0936**	**0.4433**	**0.1642**	**0.9162**	**0.9669**	**0.9844**
	HRDFuse,Ours	256 × 256 / 80°	0.0981	0.0945	0.4466	0.1656	0.9147	0.9666	0.9842
3D60	FCRN Laina et al. (2016a)	−/−	0.0699	0.2833	-	-	0.9532	0.9905	0.9966
	RectNet Zioulis et al. (2018)	−/−	0.0702	0.0297	0.2911	0.1017	0.9574	0.9933	0.9979
	Mapped Convolution Eder et al. (2019)	−/−	0.0965	0.0371	0.2966	0.1413	0.9068	0.9854	0.9967
	BiFuse with fusion Wang et al. (2020)	−/−	0.0615	-	0.2440	-	0.9699	0.9927	0.9969
	UniFuse with fusion Jiang et al. (2021)	−/−	0.0466	-	0.1968	-	0.9835	0.9965	0.9987
	ODE-CNN Cheng et al. (2020)	−/−	0.0467	0.0124	0.1728	0.0793	0.9814	0.9967	0.9989
	OmniFusion (1-iter) Li et al. (2022)	128 × 128 / 80°	0.0469	0.0127	0.1880	0.0792	0.9827	0.9963	0.9988
	OmniFusion (2-iter) Li et al. (2022)	128 × 128 / 80°	0.0430	0.0114	0.1808	0.0735	0.9859	0.9969	0.9989
	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**

where w_0 and w_1 are learnable parameters and $w_0 + w_1 = 1$ (superiority of adaptive weighting is shown in Table. 6). Following previous works Li et al. (2022); Jiang et al. (2021), we adopt BerHu loss Laina et al. (2016b) for pixel-wise depth supervision, denoted as $\mathcal{L}_{depth}$. Furthermore, to encourage the holistic distribution to be consistent with all depth values in the ground truth depth map, we adopt the commonly used bi-directional Chamfer loss Fan et al. (2016) as the holistic distribution loss $\mathcal{L}_{H_{bin}}$. Therefore, the total loss $\mathcal{L}_{total}$ can be written as:

$$\mathcal{L}_{total} = \mathcal{L}_{depth} + \lambda \mathcal{L}_{H_{bin}}, \tag{8}$$

where λ is a weight factor and set to 0.1 for all experiments empirically Bhat et al. (2021).

4 Experiments

Datasets and Metrics. We conduct experiments on three benchmark datasets: Stanford2D3D Armeni et al. (2017), Matterport3D Chang et al. (2017), and 3D60 Zioulis et al. (2018). Note that Stanford2D3D and Matterport3D are real-world datasets, while 3D60 is composed of two synthetic datasets (SUNCG Song et al. (2016) and SceneNet Handa et al. (2016)) and two real-world datasets (Stanford2D3D and Matterport3D).

Following previous works Li et al. (2022); Wang et al. (2020), we evaluate our method with the standard metrics: Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE), Root Mean Squared Logarithmic Error (RMSE (log)), as well as a percentage metric with a threshold δ_t, where $t \in \{1.25^1, 1.25^2, 1.25^3\}$. *Due to the lack of space, the details of datasets and metrics can be found in the appendix.*

Implementation Details. We implement our method using Pytorch and train it on a single NVIDIA 3090 GPU. We use ResNet-34 He et al. (2016), pre-trained on ImageNet Deng et al. (2009), as the encoder. Following Li et al. (2022), we use Adam Kingma & Ba (2014) optimizer with cosine annealing Loshchilov & Hutter (2016) learning rate policy and set the initial learning rate to 10^{-4}. The default TP patch number is $N = 18$, and the batch size is 4. We train 80 epochs for Stanford2D3D Armeni et al. (2017) and 60 epochs for Matterport3D Chang et al. (2017), and 3D60 Zioulis et al. (2018). The input images are augmented only by horizontal translation and vertical flipping.

Table 2: The ablation results for individual components.

Methods	FPS	#Params	Abs Rel ↓	Sq Rel ↓	RMSE ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
ERP branch only	2.88	**33.57M**	0.1028	0.0985	0.4543	0.9086	0.9658	0.9841
TP branch only	2.56	37.09M	0.1018	0.0982	0.4492	0.9104	0.9662	0.9842
ERP branch + TP branch+ geometric fusion	2.82	70.66M	0.0986	0.0944	0.4466	0.9141	0.9664	0.9843
ERP branch +TP branch + SFA	**6.21**	49.95M	0.0991	0.0956	0.4479	0.9132	0.9666	0.9843
ERP branch +TP branch + SFA + CDDC	5.52	53.77M	**0.0967**	**0.0936**	**0.4433**	**0.9162**	**0.9669**	**0.9844**

4.1 Comparison with the state-of-the-arts

In Table. 1, we compare our HRDFuse with the SOTA methods on three benchmark datasets. For a fair comparison, we do not discuss self-supervised methods Vaswani et al. (2017); Lai et al. (2021). Note that OmniFusion did not provide the pre-trained models on the Matterport3D dataset, so we re-trained them with the official hyper-parameters. For all the datasets, we show the results of the proposed HRDFuse with TP patch sizes of 128×128 and 256×256.

As shown in Table. 1, **our HRDFuse performs favorably against** the SOTA methods Li et al. (2022); Jiang et al. (2021); Wang et al. (2020); Zioulis et al. (2018); Laina et al. (2016a) **by a significant margin on all three datasets.** Specifically, for the Stanford2D3D dataset, our HRDFuse with the patch size of 256×256 outperforms UniFuse Jiang et al. (2021) by 16.07% (Abs Rel), 15.85% (RMSE), and 4.29% (δ_1). Compared with OmniFusion (2-iter), our HRDFuse improves RMSE(log) by 11.07% and δ_1 by 1.52%.

For Matterport3D and 3D60 datasets, which contain more samples, our HRDFuse is more advantageous and surpasses the compared methods for all metrics. On the Matterport3D dataset, our HRDFuse with the patch size 128×128 outperforms UniFuse by 2.65% (δ_1), 9.03% (Abs Rel), and outperforms OmniFusion (2-iter) by 3.97%(Abs Rel), 3.41% (Sq Rel). On the 3D60 dataset, HRDFuse with the patch size 256×256 outperforms UniFuse by 23.18% (Abs Rel) and 20.99% (RMSE), and outperforms OmniFusion (2-iter) by 16.74% (Abs Rel) and 13.99% (RMSE).

In Fig. 4, we present the qualitative comparison with UniFuse 4(b) and OmniFusion 4(c). As demonstrated in the figure, our HRDFuse 4(d) can recover more regional structural details (e.g., leaves and seats) and suffer less from artifacts caused by the discontinuity among TP patches (see red boxes). *More qualitative comparisons can be found in the appendix.*

4.2 Ablation Study and Analyses

The effectiveness of each module. We verify the effectiveness of each module in our HRDFuse by adding one module each time (Table. 2). We form our baselines in three ways. Firstly, for the ERP branch-only baseline, we directly employ the feature extractor to obtain the decoded feature map and add a convolutional layer, followed by a commonly used up-sampling block, to regress the depth map. Secondly, with only the TP branch, we add the geometric fusion, as done in Li et al. (2022), to the feature extractor to obtain the ERP format depth map. Thirdly, we combine the ERP branch and TP branch, followed by the geometric fusion mechanism in Li et al. (2022). Based on this, we then add the SFA module. Here, we directly leverage the spatially aligned index map to aggregate the patch feature vectors V_n into an ERP feature map and predict the depth map, without employing the decoder (see Fig. 2) or the geometric fusion module of Li et al. (2022) in the TP branch. Lastly, we add the CDDC module to learn the holistic-with-regional depth distributions.

Table 3: The ablation results for the TP patch size and FoV.

Patch FoV	Patch size	Abs Rel ↓	Sq Rel ↓	RMSE ↓
60	128×128	**0.0986**	0.0961	0.4454
	256×256	0.0986	**0.0942**	**0.4448**
80	128×128	**0.0967**	**0.0936**	**0.4433**
	256×256	0.0981	0.0945	0.4466
100	128×128	**0.0970**	**0.0938**	**0.4453**
	256×256	0.0979	0.0940	0.4458

As shown in Table. 2, with the ERP branch alone, it is difficult to alleviate the projection distortion, thus leading to the worst depth estimation performance. The performance improves when using the TP branch only due to less distortion, and is further improved by the fusion of the ERP branch and TP branch (with the geometric fusion mechanism). Furthermore, by introducing the SFA module,

the network parameters are significantly reduced by 29.31%, leading to more than three frames per second (fps) gain in inference speed. When the CDDC module is finally added, the performance is further boosted by 2.42%(Abs Rel) and 2.09%(Sq Rel), although the parameters slightly increase.

Patch size, FoV, and the number of patches of TP. They are essential parameters and can directly affect the accuracy and efficiency of our method. Therefore, we study their impact and find an optimal balance between efficiency and performance. Following Li et al. (2022), we fix the patch number as 18 and study how TP patch size affects the learning under multiple patch FoVs. As shown in Table 3, on the Matterport3D dataset, all the results with the patch size of 128×128 perform better than those of 256×256, which indicates that too large patch size may cause the redundancy of regional structural information and degrade the accuracy of the final ERP format output. Meanwhile, we can observe the influence of patch FoV in Table 3: either too small patch FoV or too large patch FoV degrades the performance. When FoV is too small, the regional information in each TP patch would be insufficient; in contrast, too large FoV will increase the inconsistency in the overlapping areas between adjacent TP patches.

Table 4: The ablation results for the number of TP patches.

Number	Patch size/FoV	Abs Rel ↓	Sq Rel ↓	RMSE ↓	δ_1 ↑	δ_2 ↑
10	128×128 / 80°	0.0996	0.0965	0.4491	0.9130	0.9664
18		**0.0967**	**0.0936**	**0.4433**	**0.9162**	0.9669
26		0.0978	0.0945	0.4444	0.9151	**0.9670**
46		0.1232	0.1178	0.4996	0.8780	0.9563
10	256×256 / 80°	0.0976	0.0948	0.4447	0.9152	0.9668
18		0.0981	0.0945	0.4466	0.9147	0.9666
26		0.0974	0.0953	0.4478	0.9147	0.9662
46		**0.0966**	**0.0938**	**0.4432**	**0.9168**	**0.9668**

Furthermore, as the number of TP patches and the computational memory cost are directly related, we fix the patch size and FoV to compare the depth results with different patch numbers such that we can find the most cost-effective patch number. As shown in Table. 4, too few patches can not provide sufficient regional structural information, while too many patches lead to the redundancy of regional information, thus degrading the role of holistic contextual information. We find that $N = 18$ performs best in our experiments.

Table 5: The ablation study for the number B of depth distribution histogram bins.

Number of bins	Abs Rel ↓	Sq Rel ↓	RMSE ↓	δ_1 ↑	δ_2 ↑
20	0.0997	0.0963	0.4502	0.9132	0.9661
50	0.0971	0.0939	0.4454	0.9159	0.9665
100	**0.0967**	**0.0936**	**0.4433**	**0.9162**	**0.9669**
150	0.0997	0.0948	0.4497	0.9121	0.9662
300	0.0999	0.0959	0.4476	0.9131	0.9666

Number of bins. In this study, we compare the performance with various numbers of depth distribution histogram bins. As observed from Table. 5, starting from B=20, the depth accuracy first improves with the increase of B, and then drops significantly. The result indicates that too many bins lead to difficulty in classification. For this reason, we choose 100 as the number of bins for experiments.

Table 6: The ablation study for the final fusion.

ERP branch	TP branch	Abs Rel ↓	Sq Rel ↓	RMSE ↓	δ_1 ↑	δ_2 ↑
1	0	0.0976	0.0948	0.4450	0.9153	0.9664
0	1	0.0975	0.0944	0.4459	0.9149	0.9670
0.5	0.5	0.0969	0.0942	0.4442	0.9157	0.9668
Adaptive weighting		**0.0967**	**0.0936**	**0.4433**	**0.9162**	**0.9669**

Weights of fusion. Table. 6 lists the estimated depth results under four groups of fusion weights with the patch number set as $N = 18$, patch size as 128×128, and FoV as 80°. Overall, our adaptive weighting achieves the best performance.

5 CONCLUSION

This paper proposed a novel solution for monocular 360° depth estimation, which predicts an ERP format depth map by collaboratively learning the holistic-with-regional depth distributions. To address the two issues: 1) challenges in pixel-wise depth value regression; 2) boundary discontinuities brought by the geometric fusion, our HRDFuse introduced the SFA module and the CDDC module, whose contributions allow HRDFuse to efficiently incorporate ERP and TP, and significantly improve the depth prediction accuracy and achieve new SOTA performance.

Limitations and future work Our work focused on the supervised monocular 360° depth estimation task and did not cover self-supervised methods. In the future, we will further explore the potential of TP, e.g., contrastive learning for TP patches. In addition, our task and 360° semantic segmentation task are closely related, as they are both dense scene understanding tasks. Therefore, joint 360° monocular depth estimation and semantic segmentation based on the combination of ERP and TP is a promising research direction.

REFERENCES

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. doi: 10.48550/arXiv.2205.10468. URL https://doi.org/10.48550/arXiv.2205.10468.

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

Jiayang Bai, Shuichang Lai, Haoyu Qin, Jie Guo, and Yanwen Guo. Glpanodepth: Global-to-local panoramic depth estimation. *CoRR*, abs/2202.02796, 2022.

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

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pp. 213–229. Springer, 2020.

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*, pp. 667–676. IEEE Computer Society, 2017.

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. *CoRR*, abs/2108.05895, 2021.

Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. In *CVPR*, pp. 1420–1429. Computer Vision Foundation / IEEE Computer Society, 2018.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *computer vision and pattern recognition*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

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

Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent images for mitigating spherical distortion. In *CVPR*, pp. 12423–12431. Computer Vision Foundation / IEEE, 2020.

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.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pp. 2002–2011. Computer Vision Foundation / IEEE Computer Society, 2018.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778. IEEE Computer Society, 2016.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ziye Lai, Dan Chen, and Kaixiong Su. Olanet: Self-supervised 360° depth estimation with effective distortion-aware view synthesis and l1 smooth regularization. *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2021.

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)*, pp. 239–248, 2016a.

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

Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In *AAAI*, pp. 1873–1881. AAAI Press, 2021.

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.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, 2016.

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *ICCV*, pp. 357–366. IEEE, 2021.

Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti. Slicenet: Deep dense depth estimation from a single indoor panorama using a slice-based representation. In *CVPR*, pp. 11536–11545. Computer Vision Foundation / IEEE, 2021.

Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 360monodepth: High-resolution 360° monocular depth estimation. *CoRR*, abs/2111.15669, 2021.

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.

Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *ECCV (16)*, volume 11220 of *Lecture Notes in Computer Science*, pp. 732–750. Springer, 2018.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, pp. 459–468. Computer Vision Foundation / IEEE, 2020.

Youngho Yoon, Inchul Chung, Lin Wang, and Kuk-Jin Yoon. Spheresr: 360deg image super-resolution with arbitrary projection via continuous spherical image representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5677–5686, 2022.

Chuanqing Zhuang, Zhengda Lu, Yiqun Wang, Jun Xiao, and Ying Wang. Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation. *CoRR*, abs/2112.14440, 2021.

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*, pp. 453–471. Springer, 2018.

A APPENDIX

A.1 MORE DETAILS OF TANGENT PROJECTION

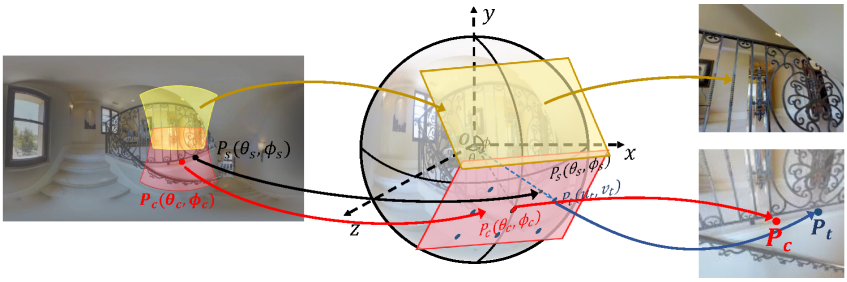


Figure 5: An example of TP and ERP. Two TP patches are projected from two different areas (red area and yellow area).

We start by introducing an example of the tangent projection (TP) Ai et al. (2022). As shown in Fig. 5, 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:

$$
\begin{aligned}
u_t &= \frac{\cos(\phi_s)\sin(\theta_s - \theta_c)}{\cos(c)}, \\
v_t &= \frac{\cos(\phi_c)\sin(\phi_s) - \sin(\phi_c)\cos(\phi_s)\cos(\theta_s - \theta_c)}{\cos(c)}, \\
\cos(c) &= \sin(\phi_c)\sin(\phi_s) + \cos(\phi_c)\cos(\phi_s)\cos(\theta_s - \theta_c).
\end{aligned} \tag{9}
$$

And the inverse transformations are:

$$
\begin{aligned}
\theta_s &= \theta_c + \tan^{-1}\left(\frac{u_t \sin(\sigma)}{\gamma\cos(\phi_c)\cos(\sigma) - v_t\sin(\phi_c)\sin(\sigma)}\right), \\
\phi_s &= \sin^{-1}\left(\cos(\sigma)\sin(\phi_c) + \frac{1}{\gamma}v_t\sin(\sigma)\cos(\phi_c)\right),
\end{aligned} \tag{10}
$$

where $\gamma = \sqrt{u_t^2 + v_t^2}$ and $\sigma = \tan^{-1}\gamma$. With Eq.9 and Eq. 10, 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 Li et al. (2022), TP patches are sampled from four latitudes: -67.5°, -22.5°, 22.5°, 67.5°, with 3, 6, 6, 3 patches on each latitude, respectively (see Fig 6c). Besides, for one more case, as shown in Fig 6d, 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 6, we can see that with the patch number increased, the area of the overlapping regions increased correspondingly. As shown in Table. 6, 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. 7, 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

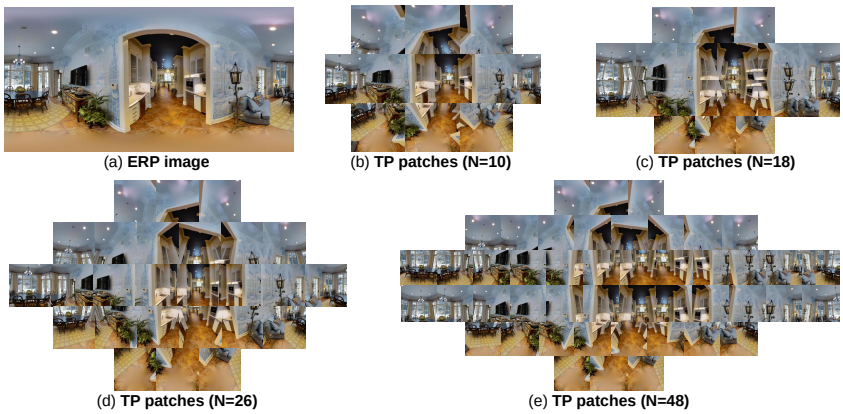


Figure 6: (a) An ERP image; (b) TP patches with the patch number $N = 10$, which are sampled at three latitudes; (c) TP patches with $N = 18$, which are sampled at four latitudes; (d) TP patches with $N = 26$, which are sampled at five latitudes; (e) TP patches with $N = 46$, which are sampled at six latitudes

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.

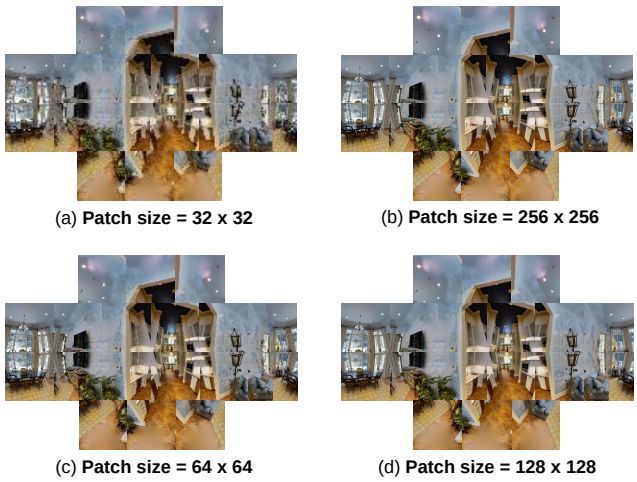


Figure 7: TP patches with different patch sizes.

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. 8. Compared with the complete view of Fig. 6a, 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°.

Figure 8: TP patches with different patch FoVs.

B MORE DETAILS OF COLLABORATIVE DEPTH DISTRIBUTION CLASSIFICATION

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. 7. 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

$$\mathbf{c}_i^H = D_{min} + (\mathbf{w}_i^H/2 + \sum_{j=1}^{i-1} \mathbf{w}_j^H), \tag{11}$$

$$\mathbf{w}_i^H = (D_{max} - D_{min}) \frac{(\mathrm{mlp}(Tk_{out}^H[0]))_i + \epsilon}{\sum_{j=1}^{B} (\mathrm{mlp}(Tk_{out}^H[0]))_j + \epsilon}, \tag{12}$$

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):

$$D^H(i, j) = \sum_{b=1}^{B} P^H(i, j)_b \cdot \mathbf{c}_b^H. \tag{13}$$

For the regional depth distribution classification, as illustrated in the Table. 7, 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 $B \times N$. 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:

$$M_c(i, j) = \sum_{n=1}^{N} M(i, j)_n \cdot \mathbf{c}_n^R \tag{14}$$

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 Q^R. Then we combine Q^R with the spatial locations of index map M to obtain a feature map M_{key}. Moreover, we introduce the embedding vectors Q^H of the ERP branch. With M_{key} as the "keymap" and Q^H as the "queries", we can predict the probability score map P^R and further output the ERP format regional depth map D^R.

B.1 MORE DETAILS OF DATASETS AND METRICS

We conduct experiments on three benchmark datasets: Stanford2D3D Armeni et al. (2017), Matterport3D Chang et al. (2017) and 3D60 dataset Zioulis et al. (2018). Note that Stanford2D3D dataset and Matterport3D dataset are real-world datasets, while 3D60 dataset is composed of two synthetic datasets (SunCG Song et al. (2016) and SceneNet Handa et al. (2016)) 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

Table 7: Network summary of the CDDC module ($\odot$ denotes the dot-production).

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. 11, Eq. 12	$1 \times B$	$\mathbf{c}^H$
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& $Tk_{out}^H[1:C_2+1](Q^H)$	$H_e/2 \times W_e/2 \times C_1$& $C_2 \times C_1$	- -	- -	- -	$\odot$	$H_e/2 \times W_e/2 \times C_2$	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. 13	$H_e \times W_e \times 1$	D^H
Regional Depth Distribution Classification							
$\{F_n^{\mathrm{TP}}\}$	$\frac{H_t}{2} \times \frac{W_t}{2} \times C_t \times N$	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. 11, Eq. 12	$1 \times B \times N$	$\mathbf{c}^R$
$\mathbf{c}^R$&M	$1 \times B \times N$&$\frac{H_e}{2} \times \frac{W_e}{2} \times N$	-	-	-	Eq. 14	$\frac{H_e}{2} \times \frac{W_e}{2} \times B$	M_c
Regional Range Attention Map							
$Tk_{out}^R[1:C_2+1]$	$C_2 \times C_1 \times N$	-	-	-	Mean	$C_1 \times N$	Q^R
Q^R&M	$C_1 \times N$&$\frac{H_e}{2} \times \frac{W_e}{2} \times N$	-	-	-	Similar to Eq. 14	$\frac{H_e}{2} \times \frac{W_e}{2} \times C_1$	M_{key}
M_{key}&Q^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	$H_e \times W_e \times B$&$H_e \times W_e \times$	-	-	$B/1$	Similar to Eq. 14	$H_e \times W_e \times 1$	D^R

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.

B.2 Additional Visual Results

More visual comparisons on Stanford2D3D and Matterport3D. In Fig. 9, we perform qualitative comparisons with the SOTA methods, UniFuse. Jiang et al. (2021) and OmniFusion Li et al. (2022), 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. 10, we perform qualitative comparisons with the SOTA methods, UniFuse. Jiang et al. (2021) and OmniFusion Li et al. (2022), on the 3D60 dataset, which contains both real-world and synthetic samples. From the visual results, we further confirm the superiority of our HRDFuse.

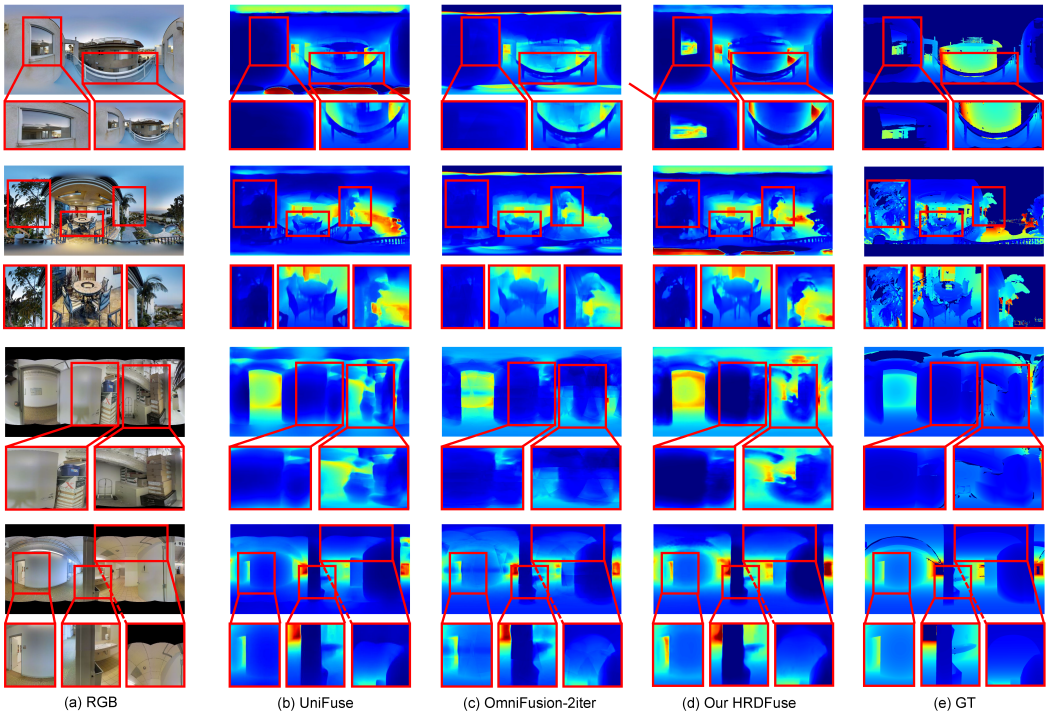


Figure 9: Visual comparisons on Stanford2D3D and Matterport3D.

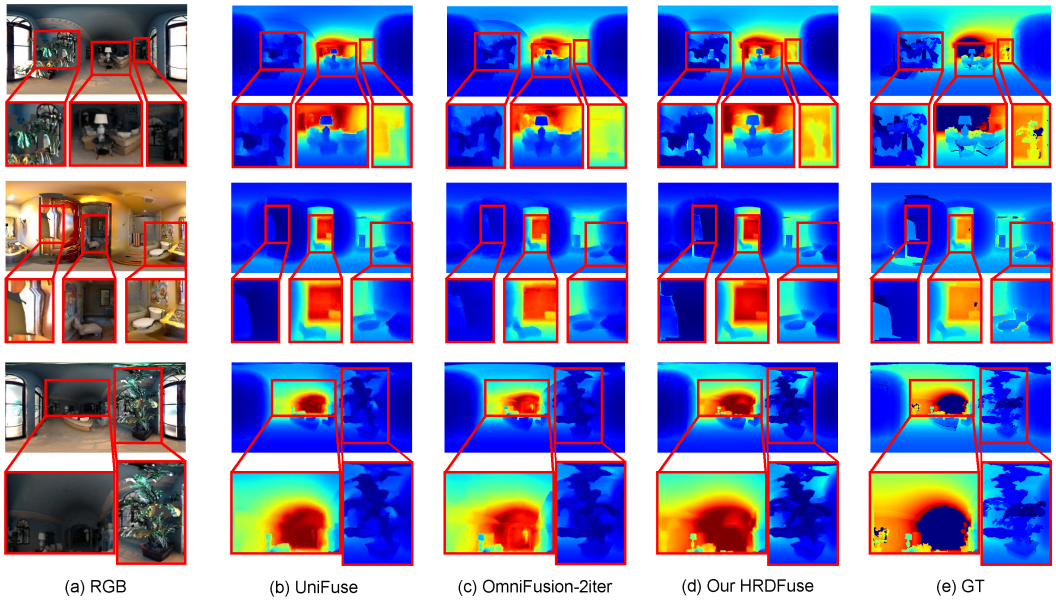


Figure 10: Visual comparisons on 3D60 dataset.