

# Improving 360° Monocular Depth Estimation via Non-local Dense Prediction Transformer and Joint Supervised and Self-supervised Learning

Ilwi Yun<sup>1</sup>, Hyuk-Jae Lee<sup>1</sup>, Chae Eun Rhee<sup>2</sup>

<sup>1</sup> Seoul National University, Korea

<sup>2</sup> Inha University, Korea

yuniw@capp.snu.ac.kr, hjlee@capp.snu.ac.kr, chae.rhee@inha.ac.kr

## Abstract

Due to difficulties in acquiring ground truth depth of equirectangular (360°) images, the quality and quantity of equirectangular depth data today is insufficient to represent the various scenes in the world. Therefore, 360° depth estimation studies, which relied solely on supervised learning, are destined to produce unsatisfactory results. Although self-supervised learning methods focusing on equirectangular images (EIs) are introduced, they often have incorrect or non-unique solutions, causing unstable performance. In this paper, we propose 360° monocular depth estimation methods which improve on the areas that limited previous studies. First, we introduce a self-supervised 360° depth learning method that only utilizes gravity-aligned videos, which has the potential to eliminate the needs for depth data during the training procedure. Second, we propose a joint learning scheme realized by combining supervised and self-supervised learning. The weakness of each learning is compensated, thus leading to more accurate depth estimation. Third, we propose a non-local fusion block, which can further retain the global information encoded by vision transformer when reconstructing the depths. With the proposed methods, we successfully apply the transformer to 360° depth estimations, to the best of our knowledge, which has not been tried before. On several benchmarks, our approach achieves significant improvements over previous works and establishes a state of the art.

## 1 Introduction

Recently, research interest in processing equirectangular (360°) images has increased as virtual reality enters the limelight. Equirectangular images (EIs) have advantages over traditional rectilinear images (RIs) in that they enable a 360° field of view. This benefit, however, complicates the acquisition of ground truth depths. Aside from the technical difficulties associated with 360° depth scanners, one practical difficulty is that sensors would be visible from the 360° RGB cameras, leading to partially obscured images (Matzen et al. 2017; Zioulis et al. 2018). Moreover, to acquire diverse and realistic synthesized data, numerous things should be set exquisitely, which often requires professional designers and tools (Zheng et al. 2020). Due to such problems, the quality and quantity of equirectangular depth data today is insufficient to represent fully the various scenes in the world.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

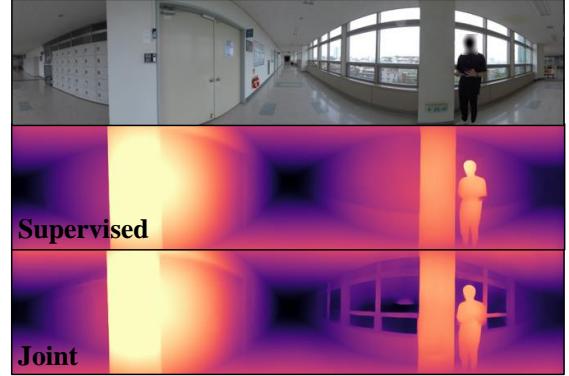


Figure 1: The effect of joint learning. Unlike the network trained via supervised learning only (middle row), the network trained via joint learning is able to distinguish windows from the walls (bottom row)

Therefore, learning 360° depths in a supervised manner is destined to produce unsatisfactory results because the performance of supervised learning is highly dependent on the dataset. To overcome the lack of data, learning 360° depths in a self-supervised manner has been attempted. However, previous methods require either calibrated stereo EI pairs (Payen de La Garanderie, Atapour Abarghouei, and Breckon 2018; Zioulis et al. 2019; Wang et al. 2020b) or conversion to cubemap projection (Wang et al. 2018a), both of which have limitations with regard to further improvements. Moreover, self-supervised learning often delivers incorrect or non-unique solutions (*e.g.*, light reflected object), which cause unstable performance.

In this paper, we propose 360° monocular depth estimation methods which improve on the areas that limited previous studies. First, we propose a self-supervised method for the learning of depth that only utilizes gravity-aligned video sequences, which has the potential to eliminate the needs of depth data during the training procedure. Similar to prior work (Zioulis et al. 2019), we utilize the relationships between consecutive scenes but improve it through consistency between depths. Second, we propose a joint learning scheme realized by combining supervised and self-supervised learning. Despite the limitations of each learning scheme, all pre-

vious works on the 360° depth estimation, to the best of our knowledge, have relied solely on either supervised or self-supervised learning. We show that the joint learning improves the unstable performance of self-supervised learning as well as the incorrect prediction of supervised learning caused by data scarcity, as visualized in Figure 1. Third, we propose a non-local fusion block which improves on the areas missed by vision transformers for dense prediction. Through non-local operations, global information encoded by a transformer can be further retained when reconstructing the depths. Under a challenging environment for a transformer (*i.e.*, lack of a large-scale dataset), we were able to train the vision transformer successfully using the features learned from depth of RIs. To the best of our knowledge, this is the first work applying transformers successfully to 360° depth estimation. Our approaches achieve significant improvements over previous works on several benchmarks, thus establishing a state of the art.

## 2 Background and Related work

### 2.1 EI geometry

Although EIs appear to be two dimensional (2D) images, EIs and RIs are different in many ways. EIs are generated by flattening the rays projected on a three dimensional (3D) sphere, whereas RIs are generated by directly projecting rays on a 2D plane. Therefore, EIs are 3D images despite their 2D structure. The spherical coordinates  $(\theta, \phi, \rho)$  are often used instead of the pixel coordinates  $(x, y)$  for this reason, and relationship between them is illustrated in Figure 2. Each value of  $\theta \in (0^\circ, 360^\circ)$  and  $\phi \in (0^\circ, 180^\circ)$  represents the latitude and longitude of an EI, and  $\rho$  denotes the radius of the sphere. Further, the spherical coordinates can be converted to Cartesian coordinates  $(X_c, Y_c, Z_c)$  by Eq. 1

$$\begin{cases} X_c = \rho \cdot \sin(\phi) \cdot \cos(\theta) \\ Y_c = \rho \cdot \sin(\phi) \cdot \sin(\theta) \\ Z_c = \rho \cdot \cos(\phi) \end{cases} \quad (1)$$

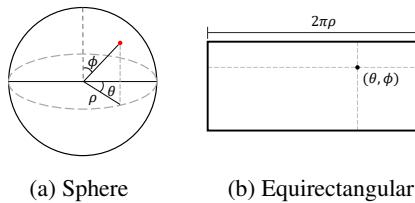


Figure 2: Equirectangular geometry

Meanwhile, rotations on EIs are defined as the yaw, pitch and roll. Due to undesirable visual changes (Sun, Sun, and Chen 2021; Davidson, Alvi, and Henriques 2020), a gravity-aligned structure (*i.e.*, with the roll and pitch set to 0°) is generally assumed in equirectangular depth benchmarks (Armeni et al. 2017; Chang et al. 2017; Zioulis et al. 2018; Zheng et al. 2020) and in recent studies (Pintore et al. 2021; Sun, Sun, and Chen 2021). If captured images/videos are not gravity-aligned, they can be calibrated afterwards (Xian et al. 2019; Davidson, Alvi, and Henriques 2020).

### 2.2 Supervised 360° depth estimation

Omnidepth (Zioulis et al. 2018) presents a 3D60 dataset (Matterport3D, Stanford3D and SunCG) by re-rendering previous 360° data [*e.g.*, Matterport (Chang et al. 2017), Stanford (Armeni et al. 2017)], which are now commonly used for training 360° depths. Bifuse (Wang et al. 2020a) jointly uses cubemap projected images with EIs to improve the performance. SliceNet (Pintore et al. 2021) splits the inputs and recovers them through long short-term memory (Xingjian et al. 2015) to retain the global information. HoHoNet (Sun, Sun, and Chen 2021) improves the performance and computational efficiency by focusing on the per-column information of the gravity-aligned EIs. Recently, multi-task learning among the depth, layout and semantics was attempted to improve performance outcomes. (Jin et al. 2020) regularizes the depth considering the layout, while (Zeng, Karaoglu, and Gevers 2020) train the layout, semantic and depth simultaneously.

### 2.3 Self-supervised 360° depth estimation

Self-supervised depth learning has been widely attempted for RIs based on the following intuition: The closer the object is to the camera, the greater the change in the object’s position when the camera moves (Garg et al. 2016; Godard, Mac Aodha, and Brostow 2017; Godard et al. 2019; Gordon et al. 2019). However, that intuition is not applied to EIs due to the different geometry, as shown in Figure 3. When the camera moves forward (denoted by the red arrows), the relative movements of objects in the scenes of RIs are represented by the dotted arrows in Figure 3 (a). The movement of objects only depends on the camera movements (direction) and corresponding depths (magnitude). The movement of objects in EIs, however, is also affected by the positions of the objects in EIs. As the camera moves forward, the objects in front of the camera become closer, while those of opposite side become further away, as shown in Figure 3 (b). Because more variables control the objects of EIs, learning 360° depths using a self-supervision becomes more difficult.

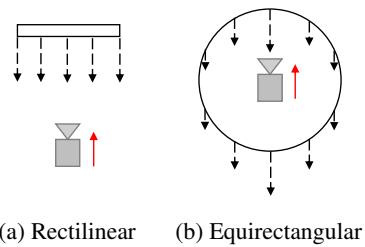


Figure 3: Difference in movements of a scene

In addition, self-supervised depth learning often has incorrect or non-unique solutions in some cases. Light reflected objects, which are not predictable using only the depth and camera motion, is one such example. These objects cause the network to output wrong depth values because the light reflection is controlled by light sources, not depths (Refer to Technical Appendix for more examples). Although several attempts have been made to remove those

kinds of intractable objects during the training process (Goddard et al. 2019; Gordon et al. 2019; Tan et al. 2021), there remain objects controlled by numerous variables and, not merely by depth and camera motions, which makes self-supervised learning a challenge.

360SD-Net (Wang et al. 2020b) and SvSyn (Zioulis et al. 2019) use stereo EI pairs as input for EI depth training data to simplify the relationship between the depth, image and camera motions. However, 360° field of view makes it difficult to acquire stereo EIs using two 360° cameras given that each camera is captured by others, which limits the use of this method. EBS (Payen de La Garanderie, Attapour Abarghouei, and Breckon 2018) proposes the method that uses relatively abundant RI stereo pairs. They distort RIs considering the EI geometry, and use them as training data. However, distorted data has a restricted field of view (< 90°), which cannot replace the EIs fundamentally. (Wang et al. 2018a) transforms EIs into cubemap-projected images to alleviate the difference in the geometry. However, cube map projection not only leads to discontinuity between each of the cubemap faces which results in large errors, but also requires additional computations (Cheng et al. 2018; Wang et al. 2020a).

## 2.4 Vision transformer

Recently, the vision transformer (ViT) network architecture was proposed for image classification (Dosovitskiy et al. 2020). In this architecture, the transformer (Vaswani et al. 2017), which has been widely used in natural language processing, is adopted instead of convolution block. Images are split into multiple flattened patches and are encoded by a transformer. ViT achieves results that are comparable to or even better than those of a convolutional neural network (CNN) on image classification tasks. Furthermore, it was recently demonstrated that ViT yields notable performance improvements on various vision tasks. The dense prediction transformer (DPT) successfully applies ViT to segmentation and depth estimation tasks by upsampling the encoded features via convolution-based reassemble and fusion blocks (FB) (Ranftl, Bochkovskiy, and Koltun 2021). Reassemble blocks reassemble the encoded features into 3D features  $R^s$ , whereas fusion blocks upsample the  $R^s$  into fused features  $F^s$ . Unlike a CNN, however, the transformer lacks inductive bias, necessitating large-scale dataset. Under an environment with an insufficient dataset, the performance of the transformer becomes worse than that of a CNN (Dosovitskiy et al. 2020). Therefore, several attempts have been made to alleviate data dependencies through multiple known techniques, such as knowledge distillation (Touvron et al. 2021).

## 3 Proposed method

### 3.1 Overall architecture

The overall structure of the proposed training procedure, as illustrated in Figure 4, is composed of two flows: self-supervised learning (black arrows) and supervised learning (brown arrows). For self-supervised learning flow, non-local DPT estimates the depth ( $D, D'$ ) for each consecutive video scene ( $V, V'$ ), while PoseNet predicts the cam-

era motion ( $\delta v, \delta r$ ) between them. Then, each scene and depth at different viewpoints are reconstructed through sampling function ( $f_s$ ). Image consistency ( $\mathcal{L}_I$ ), depth consistency ( $\mathcal{L}_D$ ) and pose consistency ( $\mathcal{L}_P$ ) losses are imposed by comparing reconstructed samples with corresponding video scenes and the estimated depths. For supervised learning flow, estimated depth ( $D_I$ ) for image input ( $I$ ) is compared with the ground truth ( $D_I^g$ ). Traditional pixel-wise loss ( $\mathcal{L}_{pix}$ ) and gradient loss ( $\mathcal{L}_{grad}$ ) are imposed. Overall, the objective function is constructed via Eq.2. Here,  $\lambda_I$  and  $\lambda_D$  are hyper-parameters balancing the supervised and self-supervised losses. Each element of Eq.2 will be described in the following sections.

$$\mathcal{L}_{total} = \lambda_I \cdot \mathcal{L}_I + \lambda_D \cdot \mathcal{L}_D + \mathcal{L}_P + \mathcal{L}_{pix} + \mathcal{L}_{grad} \quad (2)$$

### 3.2 Self-supervised losses

In this subsection, we introduce the self-supervised learning flow illustrated in Figure 4. First, we formulate the relationships among the depth, camera motion and gravity-aligned 360° video sequences. Then, the image consistency, depth consistency and pose consistency losses are explained.

**Relationships between consecutive scenes in video** Consecutive video scenes  $V \in \mathbb{R}^{3 \times H \times W}$  and  $V' \in \mathbb{R}^{3 \times H \times W}$  can be expressed as spherical coordinate  $(\theta, \phi, \rho)$  and  $(\theta', \phi', \rho')$ , respectively. Here, each of the  $\rho$  and  $\rho'$  values can be considered as estimated depths, which are denoted correspondingly as  $D \in \mathbb{R}^{1 \times H \times W}$  and  $D' \in \mathbb{R}^{1 \times H \times W}$ . When the video proceeds from  $V$  to  $V'$  (i.e.,  $V \rightarrow V'$ ), the translation and rotation of the camera between scenes are defined as  $\delta v \in \mathbb{R}^{3 \times 1 \times 1}$  and  $\delta r \in \mathbb{R}^{3 \times 1 \times 1}$ . Because we assume that the videos are gravity-aligned,  $\delta r_y \in \mathbb{R}^{1 \times 1 \times 1}$ ,  $\delta r_z \in \mathbb{R}^{1 \times 1 \times 1}$  can be set as a constant (i.e., 0). Therefore, the camera motion is simplified to four variables:  $\delta v_x, \delta v_y, \delta v_z$  and  $\delta r_x$ . Under this environment, Eq.3 is formulated, which denotes the movements of the 3D scene point between  $V$  and  $V'$  in Cartesian coordinates according to the camera motion and depth .

$$\begin{cases} \rho \cdot \cos(\theta - \delta r_x) \cdot \sin(\phi) - \delta v_x = \rho' \cdot \cos(\theta') \cdot \sin(\phi') \\ \rho \cdot \sin(\phi) \cdot \sin(\theta - \delta r_x) - \delta v_y = \rho' \cdot \sin(\phi') \cdot \sin(\theta') \\ \rho \cdot \cos(\phi) - \delta v_z = \rho' \cdot \cos(\phi') \end{cases} \quad (3)$$

By solving Eq.3, a closed-form expression of Eq.4 is obtained, representing the relationship between  $V (\theta, \phi, \rho)$  and  $V' (\theta', \phi', \rho')$  for the depth and camera motion. Eq.4 can be expressed simply using  $f_s$  in Eq.5.

$$\begin{cases} \theta' = \tan^{-1}\left(\frac{\rho \cdot \sin(\theta - \delta r_x) \cdot \sin(\phi) - \delta v_y}{\rho \cdot \cos(\theta - \delta r_x) \cdot \sin(\phi) - \delta v_x}\right) \\ \phi' = \tan^{-1}\left(\frac{\rho \cdot \sin(\theta - \delta r_x) \cdot \sin(\phi) - \delta v_y}{\sin(\theta') \cdot (\rho \cdot \cos(\phi) - \delta v_z)}\right) \\ \rho' = \frac{\cos(\phi) \cdot \rho - \delta v_z}{\cos(\phi')} \end{cases} \quad (4)$$

$$V' = f_s(V, D, \delta v, \delta r) \quad (5)$$

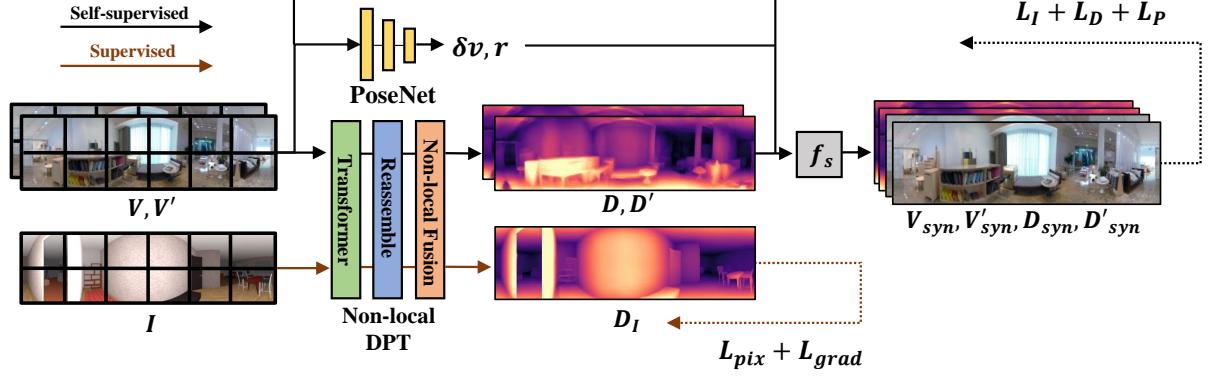


Figure 4: Overall architecture

**Image consistency loss** For consecutive scenes  $V \rightarrow V'$ , scenes at a different view point  $V'_syn$  can be synthesized from  $V$  with Eq.6 assuming the depth and camera motions are well estimated. In the same vein,  $V_{syn}$  can also be synthesized considering the reversely ordered sequences  $V' \rightarrow V$ . If the depth and camera motions are appropriately estimated, each synthesized frame should be identical to each corresponding scene in the video (*i.e.*,  $V' = V'_{syn}$  and  $V = V_{syn}$ ). Therefore, by regularizing the networks to synthesize images equal to scenes in the video, the network is indirectly trained to predict proper depth and camera motions. Therefore, the image consistency loss is constructed as Eq.7, similar to the previous works (Zioulis et al. 2019). Here,  $SM$  indicates structural similarity (Wang et al. 2004),  $\alpha$  represents the weight parameters,  $n$  indicates the number of pixels in an image.

$$\begin{aligned} V'_{syn} &= f_s(V, D, \delta v, \delta r) \\ V_{syn} &= f_s(V', D', -\delta v, -\delta r) \end{aligned} \quad (6)$$

$$\begin{aligned} L_I &= \frac{1}{n} \cdot \sum_{k=1}^n [\alpha \cdot (|V' - V'_{syn}| + |V - V_{syn}|) \\ &+ (1 - \alpha) \cdot (|1 - SM(V, V_{syn})| + |1 - SM(V', V'_{syn})|)] \end{aligned} \quad (7)$$

**Depth consistency loss** Previous self-supervised learning studies focusing on EIs only considered image consistency (Zioulis et al. 2019; Wang et al. 2020b). We argue that depth consistency can also be used for regularization, which can further strengthen the training. Here, we introduce the depth consistency loss for EIs, inspired by (Godard, Mac Aodha, and Brostow 2017). By regarding the depth as an image, the depths of different viewpoints  $D'_{syn}$  and  $D_{syn}$  can be synthesized using Eq.8. If the depth and camera motion are accurately estimated,  $D'_{syn}$  and  $D_{syn}$  become equal to  $D'$  and  $D$ , respectively. Therefore, similar to the image consistency loss,  $\mathcal{L}_D$  can be constructed with Eq.9. The estimated depths should be consistent across the scenes to minimize the loss, which causes the network to check the images in more detail.

$$\begin{aligned} D'_{syn} &= f_s(D, D, \delta v, \delta r) \\ D_{syn} &= f_s(D', D', -\delta v, -\delta r) \end{aligned} \quad (8)$$

$$L_D = \frac{1}{n} \cdot \sum_{k=1}^n (|D' - D'_{syn}| + |D - D_{syn}|) \quad (9)$$

**Pose consistency loss** If PoseNet  $P$  is properly trained, estimated camera motions for scenes in reverse order (*i.e.*,  $V \rightarrow V'$  and  $V' \rightarrow V$ ) should also have the opposite direction. Therefore, the pose consistency loss  $L_P$  can be established as Eq.10.

$$L_P = \frac{1}{n} \cdot \sum_{k=1}^4 |P(V, V') - P(V', V)| \quad (10)$$

### 3.3 Supervised losses for joint learning

Although self-supervised learning on videos has the potential to offer more accurate depth estimation, it has drawbacks that should be resolved (*i.e.*, non-optimal solutions). Supervised learning can alleviate those drawbacks because it has supervisions in the areas (*e.g.*, light reflection) which cause problems in self-supervised learning. Conversely, self-supervised learning can diversify the features learned from supervised learning through various video sequences, which makes the network perform well at the unexposed data. From this observation, we propose to use the supervised and self-supervised losses jointly. However, the scale of the depth value differs according to the depth-acquisition method used or the environment (Eigen, Puhrsch, and Fergus 2014; Chen et al. 2016; Wang et al. 2019; Ranftl et al. 2019). Therefore, scale and shift ambiguities regarding the depth must be resolved in advance to use both losses together. If this is not done, they conflict harshly and produce even worse performances. For this reason, we initially align the scale and shift of the depth via Eq.11 utilizing schemes proposed by (Ranftl et al. 2019). Considering the per-column characteristics of the gravity-aligned EIs (Sun, Sun, and Chen 2021), the depth is aligned in a column-wise manner. Here,  $D_I^g \in \mathbb{R}^{1 \times H \times W}$  indicates the ground truth depth,  $D_I^A \in \mathbb{R}^{1 \times H \times W}$  is the aligned depth, where

$s, t \in \mathbb{R}^{1 \times 1 \times W}$  represents the per-column scale and shift parameters.

$$\begin{aligned} s, t &= \underset{s, t}{\operatorname{argmin}}(s \cdot D_I + t - D_I^g) \\ D_I^A &= s \cdot D_I + t \end{aligned} \quad (11)$$

Then, we apply traditional pixel-wise loss expressed as Eq.12 and the gradient loss (Li and Snavely 2018) of Eq.13, while the gradient loss is calculated on four different scales. The gradient loss induces the network to place emphasis on the edges of the estimated depths.

$$\mathcal{L}_{pix} = \frac{1}{n} \cdot \sum_{k=1}^n |(D_I^A - D_I^g)| \quad (12)$$

$$\mathcal{L}_{grad} = \frac{1}{n} \cdot \sum_{k=1}^n |\nabla_x(D_I^A - D_I^g) + \nabla_y(D_I^A - D_I^g)| \quad (13)$$

### 3.4 Non-local dense prediction transformer

**Non-local fusion block** ViT has advantages over a CNN in that it can see the input images globally. For dense predictions, however, these advantages might be weakened. When upsampling encoded features via convolution-based FB (Ranftl, Bochkovskiy, and Koltun 2021), the receptive field is bounded by the convolution kernel size. Because EI contains a geometric structure (*e.g.*, a wall) which should be seen in a global manner (Pintore et al. 2021), losing the global outlook may result in non-accurate depth estimations. For these reasons, we propose the use of a non-local fusion block (NLFB), which performs non-local operations (Wang et al. 2018b) on each feature going into the FB, as shown in Figure 5. Here, we define  $F^s \in \mathbb{R}^{C \times H_s \times W_s}$  as the fused features at scale  $s$  and  $N^s \in \mathbb{R}^{C \times H_s \times W_s}$  as the non-local fused features. For the fused feature at the  $i$ th index  $F_i^s$ , the non-local fused feature  $N_i^s$  is calculated with Eq.14, where  $W_{\theta, \phi, g}^s \in \mathbb{R}^{C/2 \times 1 \times 1}$  and  $W_z^s \in \mathbb{R}^{C \times 1 \times 1}$  are the weight matrix to be learned and  $j$  denotes the array of all possible indexes. The features of each index  $i$  are reconstructed by seeing all other indexes  $j$ , which makes the network to continue seeing the features from a global perspective. Therefore, non-local DPT, which uses a NLFB instead of a FB, yields more accurate dense predictions.

$$\begin{aligned} f(F_i^s, F_j^s) &= e^{(W_\theta^s F_i)^T W_\phi^s F_j} \\ C(F^s) &= \sum_{\forall j} f(F_i^s, F_j^s) \\ N_i^s &= F_i^s + \frac{W_z^s}{C(F^s)} \sum_{\forall j} f(F_i^s, F_j^s) W_g^s F_j^s \end{aligned} \quad (14)$$

**Fine-tuning on EI depth** Due to lack of a inductive bias (*e.g.*, locality assumption), a large-scale dataset is needed to train a well-performing transformer (Dosovitskiy et al. 2020). Considering the small amount of EI depth data, this disadvantage is especially critical in EI depth estimation studies. To overcome this hardship, we utilize a pre-trained model which is learned from depth of RIIs based on the following observation: Depth estimations on RIIs and EIIs work

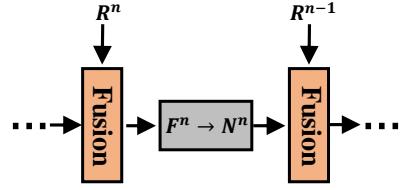


Figure 5: Non-local fusion block

similarly from the relative point of view. Here, we assume that two objects  $A$  and  $B$  are captured by RI and EI cameras, respectively. If it is perceived that  $A$  is closer than  $B$  in RIIs, it also should be the case in EIIs. This implies that features learned from depth of RIIs are also useful for EI depths. Therefore, we initialize the weights of our network using the transformer trained with RI depths and fine-tune the whole network using EI depths with some additional settings.

Although estimated depths are aligned via Eq.11 during the training procedure, we observe that scale mismatches in depth between two learning flows dominate the total loss. Before aligning depths via Eq.11, we thus robustly adjust the scales of ground truth and the estimated depth of supervised learning flows based on the scales of the depth learned from self-supervised losses. Further, we also pre-train the pose network to minimize the negative effects of incorrect pose estimations in the early phase of training. In this way, we successfully transfer the various features learned from large-scale RI depths to the equirectangular geometry. More details are described in Technical Appendix.

## 4 Experiments

Experiments are composed of four parts. First, we briefly explain the experiment environment in Section 4.1. Then, we evaluate our methods under the various settings in Sections 4.2 and 4.3. Finally, the effect of each proposed schemes is analyzed through an ablation study in Section 4.4.

### 4.1 Experimental setup

**Evaluation details** For fair and reproducible experiments, we compare our results with studies that provide a pretrained model in the open-source community. Omnidepth (Zioulis et al. 2018), SvSyn (Zioulis et al. 2019), Bifuse (Wang et al. 2020a) and HoHoNet (Sun, Sun, and Chen 2021), which are proven useful in numerous works, are compared. Stanford (Armeni et al. 2017) and 3D60 (Stanford3D, Matterport3D, SunCG) (Zioulis et al. 2018) datasets are used for evaluation. Matterport (Chang et al. 2017) dataset is not used since it requires additional data pre-processing which may provide different results according to how it is processed. The following standard depth evaluation metrics are used to compare methods: absolute relative error (Abs.rel), squared relative error (Sq.rel), root mean square error (RMS), root mean square log error (RMSlog) and relative accuracy measures ( $\delta$ ). Lower is better for Abs.rel, Sq.rel, RMS and RMSlog, whereas higher is better for  $\delta$ . Similar to (Ranftl et al. 2019; Ranftl, Bochkovskiy, and Koltun 2021), we align the predicted depths using Eq.11 for all methods before measuring

Table 1: Quantitative comparison on 3D60 dataset using the pre-trained baselines provided by each author. Numbers in **bold** indicate the best results.

Dataset	Method	AbsRel	Sq rel	RMS	RMSlog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Stanford3D	Omnidepth	0.1009	0.0522	0.3835	0.1434	0.9114	0.9855	0.9958
	SvSyn	0.1003	0.0492	0.3614	0.1478	0.9096	0.9822	0.9949
	Bifuse	0.1214	0.1019	0.5396	0.1862	0.8568	0.9599	0.9880
	HoHoNet	0.0901	0.0593	0.4132	0.1511	0.9047	0.9762	0.9933
	Ours w/ FB	0.0669	0.0249	0.2805	0.1012	0.9652	0.9944	0.9983
	Ours w/ NLFB	<b>0.0649</b>	<b>0.0240</b>	<b>0.2776</b>	<b>0.993</b>	<b>0.9665</b>	<b>0.9948</b>	<b>0.9983</b>
Matterport3D	Omnidepth	0.1136	0.0671	0.4438	0.1591	0.8795	0.9795	0.9950
	SvSyn	0.1063	0.0599	0.4062	0.1569	0.8984	0.9773	0.9934
	Bifuse	0.1330	0.1359	0.6277	0.2079	0.8381	0.9444	0.9815
	HoHoNet	<b>0.0671</b>	0.0417	0.3416	0.1270	0.9415	0.9838	0.9942
	Ours w/ FB	0.0729	0.0302	0.3089	0.1079	0.9574	0.9935	0.9980
	Ours w/ NLFB	0.0700	<b>0.0287</b>	<b>0.3032</b>	<b>0.1051</b>	<b>0.9599</b>	<b>0.9938</b>	<b>0.9982</b>
SunCG	Omnidepth	0.1450	0.1052	0.5684	0.1884	0.8105	0.9761	0.9941
	SvSyn	0.1867	0.1715	0.6935	0.2380	0.7222	0.9427	0.9840
	Bifuse	0.2203	0.2693	0.8869	0.2864	0.6719	0.8846	0.9660
	HoHoNet	0.0827	0.0633	0.3863	0.1508	0.9266	0.9765	0.9908
	Ours w/ FB	0.0740	0.0338	0.3475	0.1073	0.9584	0.9949	0.9986
	Ours w/ NLFB	<b>0.0715</b>	<b>0.0321</b>	<b>0.3401</b>	<b>0.1042</b>	<b>0.9625</b>	<b>0.9950</b>	<b>0.9986</b>

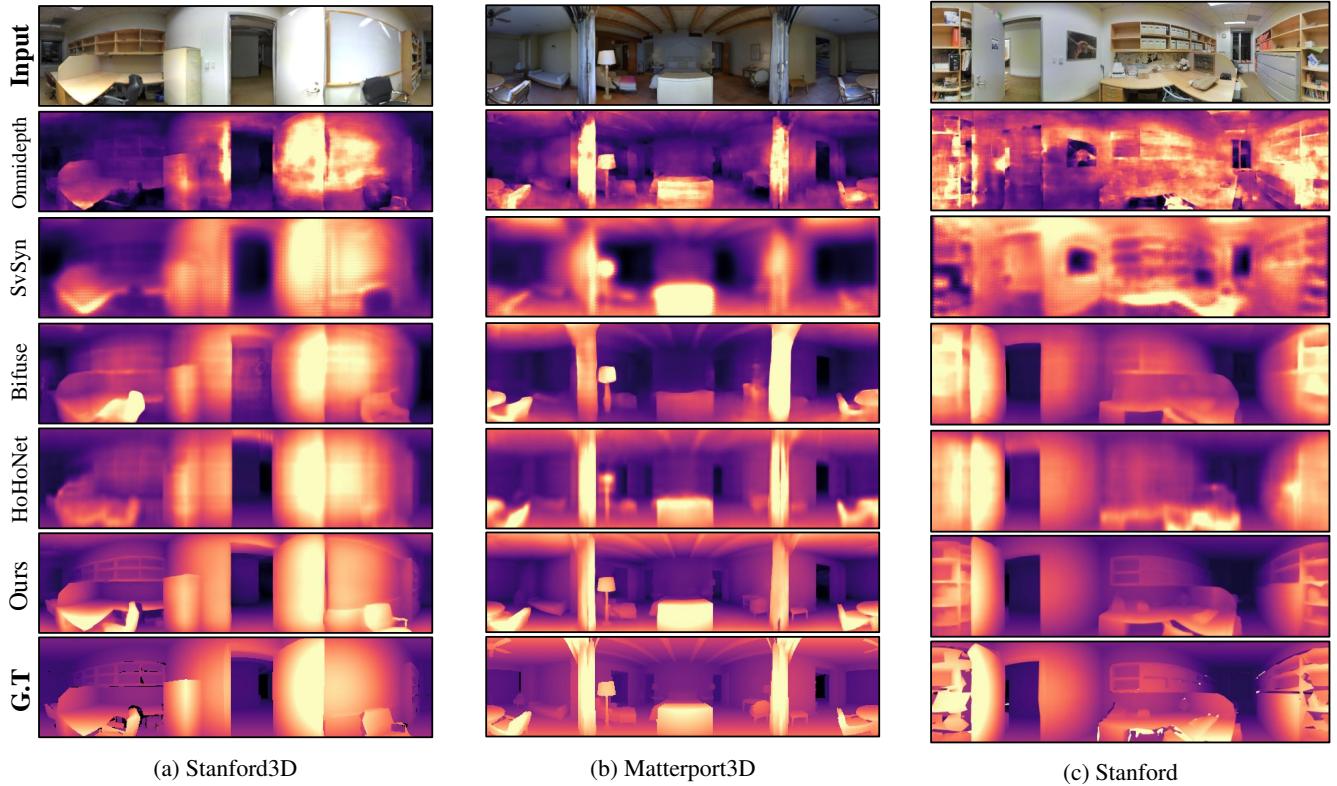


Figure 6: Qualitative comparison on Stanford3D, Matterport3D and Stanford dataset. Here, Ours indicates the model using NLFB module. Additional results are included in Technical and Multimedia Appendix.

the errors. For more details about the implementation, training, evaluation environment and additional experiments, refer to the the Technical, Code and Multimedia Appendix.

**Discussions on evaluation** Because training neural network is affected by numerous variables, it often becomes sensi-

tive even to the small changes in hyper-parameters. Therefore, unifying the training environment of the all previous works may not result in the fair comparison. Actually, the training setup of each previous study differs significantly ([Zioulis et al. 2018, 2019](#); [Wang et al. 2020a](#); [Sun, Sun, and](#)

Table 2: Quantitative comparison on Stanford dataset using the pre-trained baselines provided by each author. Numbers in **bold** indicate the best results.

Dataset	Method	AbsRel	Sq rel	RMS	RMSlog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Stanford	Omnidepth	0.1930	0.0042	0.0143	0.2691	0.7663	0.9140	0.9635
	Svsyn	0.1844	0.0039	0.0137	0.2596	0.7806	0.9220	0.9676
	Bifuse	0.1017	0.0019	0.0086	0.1783	0.9082	0.9722	0.9879
	HoHoNet	0.0801	0.0016	0.0074	0.1577	0.9355	0.9803	0.9902
	Ours w/ NLFB	<b>0.0666</b>	<b>0.0015</b>	<b>0.0066</b>	<b>0.1461</b>	<b>0.9531</b>	<b>0.9836</b>	<b>0.9910</b>

Table 3: Further quantitative comparison using the baselines re-trained under the same training environment. Numbers in **bold** indicate the best results.

Dataset	Method	AbsRel	Sq rel	RMS	RMSlog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Stanford3D	Bifuse	0.0421	0.0160	0.2199	0.0842	0.9752	0.9948	0.9983
	HoHoNet	0.0541	0.0237	0.2566	0.1030	0.9573	0.9915	0.9968
	Ours w/ NLFB	<b>0.0344</b>	<b>0.0116</b>	<b>0.1921</b>	<b>0.0709</b>	<b>0.9843</b>	<b>0.9965</b>	<b>0.9987</b>
Matterport3D	Bifuse	0.0455	0.0186	0.2368	0.0859	0.9744	0.9943	0.9981
	HoHoNet	0.0612	0.0319	0.2950	0.1142	0.9523	0.9887	0.9960
	Ours w/ NLFB	<b>0.0364</b>	<b>0.0125</b>	<b>0.1963</b>	<b>0.0700</b>	<b>0.9852</b>	<b>0.9966</b>	<b>0.9988</b>
SunCG	Bifuse	0.0323	0.0141	0.2067	0.0739	0.9811	0.9954	0.9985
	HoHoNet	0.0518	0.0291	0.2789	0.1082	0.9587	0.9898	0.9967
	Ours w/ NLFB	<b>0.0233</b>	<b>0.0076</b>	<b>0.1574</b>	<b>0.0534</b>	<b>0.9915</b>	<b>0.9979</b>	<b>0.9992</b>
Stanford	Bifuse	0.1237	0.0026	0.0114	0.2067	0.8684	0.9560	0.9823
	HoHoNet	0.1306	0.0028	0.0114	0.2138	0.8510	0.9511	0.9804
	Ours w/ NLFB	<b>0.0992</b>	<b>0.0018</b>	<b>0.0080</b>	<b>0.1717</b>	<b>0.9147</b>	<b>0.9764</b>	<b>0.9865</b>

Chen 2021), which makes it difficult to compare only the superiority of each method. For more fair and clear comparison, therefore, our methods are evaluated under the two settings. First, our method is compared with the pre-trained baselines provided by each author (Zioulis et al. 2018, 2019; Wang et al. 2020a; Sun, Sun, and Chen 2021) in the open-source community at Section 4.2. Because the performance of them is guaranteed by each author, this evaluation is fair and reproducible. Then, we compare our method with the baselines re-trained under the same training environment in Section 4.3, which further clarifies the superiority of each method.

## 4.2 Comparison with the pre-trained baselines

In this subsection, the pre-trained baselines provided by each author are used for evaluation. Our model is trained using 3D60 (Zioulis et al. 2018) and Stanford (Armeni et al. 2017) datasets, which are also used as training data in previous works.

**Quantitative results** Table 1 shows the quantitative depth prediction results on 3D60 testset. Ours w/ FB represents the model using normal fusion blocks (Ranftl, Bochkovskiy, and Koltun 2021), while Ours w/ NLFB is the model using the proposed non-local fusion blocks. Except for the Abs.rel metric on Matterport3D, our method achieves significant improvements over previous works. We observe that pre-trained Bifuse produces biased results on some specific test splits, which results in worse quantitative results than others. Meanwhile, Ours w/ NLFB provides better results than Ours w/ FB for all cases, which demonstrates the effectiveness of NLFB. Table 2 shows the quantitative results on Stanford testset. For all metrics, Ours w/ NLFB provides the

best results.

**Qualitative results** Figure 6 shows the qualitative depth estimation results. Omnidepth provides good results for small objects. Overall, however, unstable depth results are observed (*e.g.*, wall). HoHoNet produces stable depth results, but lacks detail. Small objects are not appropriately predicted, which is also reported as a weakness in their papers (Sun, Sun, and Chen 2021). Meanwhile, it is observed that Bifuse provides fine qualitative results compared to other previous works. However, the results of Ours (/w NLFB) are much more accurate than Bifuse, which are even better than the ground truth for some cases. Holes and inaccurate depths are observed among the ground truth depths, whereas Ours provides stable and accurate depth results.

## 4.3 Comparison with the re-trained baselines

In this subsection, we compare our method with the baselines re-trained under the same training environment to further clarify the superiority of our approaches. Bifuse and HoHoNet, which are the most recent studies and sharing similar training environment, are used for evaluation. Following the training environment of Bifuse and HoHoNet, each method including ours is re-trained using 3D60 dataset with  $512 \times 1024$  resolutions. Table 3 shows the quantitative results on the re-trained models. Compared to the results in Section 4.2, some improvements are observed except at the Stanford testset, which is expected considering that they are trained with 3D60 dataset only with higher resolutions. As similar to the results in Section 4.2, however, our approach provides the best results for all metrics at all testset. Consider-

Table 4: Ablation study on Structure3D dataset.  $\mathcal{L}_p$  is used for all methods. Numbers in **bold** indicate the best results.

ID	$\mathcal{L}_I$	$\mathcal{L}_D$	$\mathcal{L}_{pix} + \mathcal{L}_{grad}$	NLFB	AbsRel	Sq rel	RMS	RMSlog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
1	✓				0.1056	0.0120	0.0646	0.1611	0.9121	0.9792	0.9926
2	✓	✓			0.1048	0.0116	0.0631	0.1593	0.9151	0.9804	0.9928
3			✓		0.0867	0.0086	0.0520	0.1403	0.9451	0.9855	0.9941
4	✓		✓		0.0840	0.0079	0.0498	0.1360	0.9477	0.9870	0.9947
5	✓	✓	✓		0.0802	0.0071	0.0476	0.1302	<b>0.9540</b>	<b>0.9887</b>	0.9952
6	✓	✓	✓	✓	<b>0.0781</b>	<b>0.0071</b>	<b>0.0470</b>	<b>0.1294</b>	0.9537	0.9885	<b>0.9953</b>

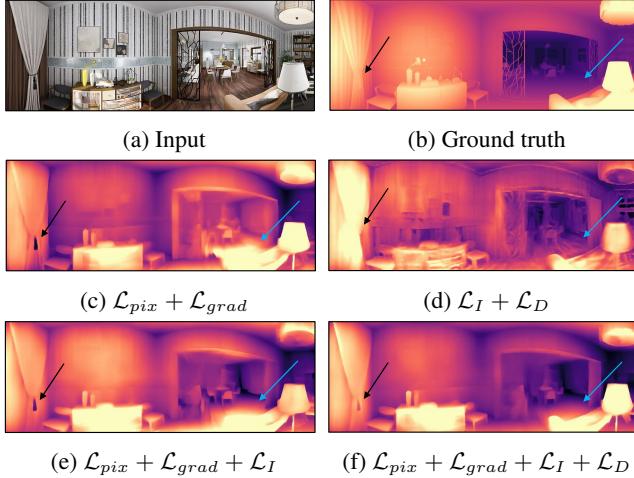


Figure 7: Studies on loss functions

ering the results of Tables 1,2 and 3 altogether, it is clear that our approach provides better results than others.

#### 4.4 Ablation study

In this section, we analyze the effect of each component of the proposed scheme through an ablation study on the Structure3D dataset (Zheng et al. 2020). Structure3D, a recently proposed dataset, is not used in the training procedure. Therefore, it is suitable to demonstrate our arguments on the proposed loss functions (e.g., perform well at unexposed data). Table 4 shows the quantitative results when proposed schemes are added gradually.  $\mathcal{L}_p$  is used in all cases, though it is omitted in Table 4 for better visualization. When only self-supervised losses are applied (ID 1 and 2), the results are not good as expected. However, when self-supervised losses are used with supervised losses (ID 4 and 5), the performance increases dramatically compared to the cases when only supervised losses are applied (ID 3). In this case,  $\mathcal{L}_D$  plays an important role when both losses are used together (ID 4 and 5). This results show that self-supervised learning actually improves the depth estimation results when combined with supervised learning. Also, it is observed that NLFB improves the performance further (ID 6).

Figure 7 shows the qualitative result of the schemes in Table 4. The model trained only with supervised losses in Figure 7 (c) produces unsatisfactory results for some areas. The black arrows indicate where the model predicts a decoration in front of the curtain as a defect. Therefore, these

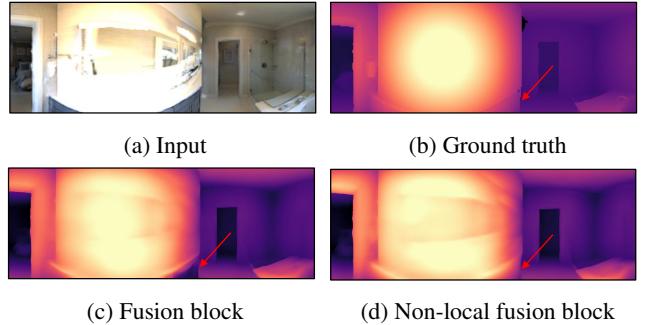


Figure 8: Effect of non-local fusion block

parts are predicted as holes, which are often found in ground truth depths. This indicates that the model is highly affected by inaccurate ground truth depths. The blue arrows indicate cases where the model fails to distinguish a sofa from the floor, which occurs because the model was not exposed to such cases during the training procedure. On the other hand, the model trained with self-supervised losses in Figure 7 (d) was able to recognize the black object at the curtain appropriately, and distinguish the sofa from the floor. However, it produces unstable depths overall. When  $\mathcal{L}_I$  is used with supervised losses as shown in Figure 7(e), the problems in Figure 7 (c) are mitigated according to the results shown in Figure 7 (d). When  $\mathcal{L}_D$  is applied in addition, as shown in Figure 7 (f), the model was able to distinguish the objects properly. Figure 8 shows the effect of a non-local fusion block in more detail. The fusion block fails to recognize the wall as a single object, and therefore, reconstructs undesirable depths (red arrows). On the other hand, the non-local fusion block reconstructs the depth well. This demonstrates that NLFB makes the network to continue to see the features with a wider view when reconstructing the depths.

## 5 Conclusion

In this paper, we introduce a self-supervised learning scheme, a joint objective function, and a non-local fusion block, in an effort to address the problems found in studies of EI depth estimations. Through the proposed scheme, significant improvements over prior works are achieved, and the benefits of each proposed method are also analyzed. We believe that each contribution not only affects the EI depth estimation research but also provides insight for those involved in studies of other vision tasks.

## Acknowledgment

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1702-54.

## References

- Armeni, I.; Sax, S.; Zamir, A. R.; and Savarese, S. 2017. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*. 2, 5, 7
- Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*. 2, 5
- Chen, W.; Fu, Z.; Yang, D.; and Deng, J. 2016. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29: 730–738. 4
- Cheng, H.-T.; Chao, C.-H.; Dong, J.-D.; Wen, H.-K.; Liu, T.-L.; and Sun, M. 2018. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1420–1429. 3
- Davidson, B.; Alvi, M. S.; and Henriques, J. F. 2020. 360° Camera Alignment via Segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, 579–595. Springer. 2
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. 3, 5
- Eigen, D.; Puhrsch, C.; and Fergus, R. 2014. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*. 4
- Garg, R.; Bg, V. K.; Carneiro, G.; and Reid, I. 2016. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, 740–756. Springer. 2
- Godard, C.; Mac Aodha, O.; and Brostow, G. J. 2017. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 270–279. 2, 4
- Godard, C.; Mac Aodha, O.; Firman, M.; and Brostow, G. J. 2019. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, 3828–3838. 2, 3
- Gordon, A.; Li, H.; Jonschkowski, R.; and Angelova, A. 2019. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8977–8986. 2, 3
- Jin, L.; Xu, Y.; Zheng, J.; Zhang, J.; Tang, R.; Xu, S.; Yu, J.; and Gao, S. 2020. Geometric Structure Based and Regularized Depth Estimation From 360 Indoor Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 889–898. 2
- Li, Z.; and Snavely, N. 2018. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2041–2050. 5
- Matzen, K.; Cohen, M. F.; Evans, B.; Kopf, J.; and Szeliski, R. 2017. Low-cost 360 stereo photography and video capture. *ACM Transactions on Graphics (TOG)*, 36(4): 1–12. 1
- Payen de La Garanderie, G.; Atapour Abarghouei, A.; and Breckon, T. P. 2018. Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 789–807. 1, 3
- Pintore, G.; Agus, M.; Almansa, E.; Schneider, J.; and Gobbetti, E. 2021. SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11536–11545. 2, 5
- Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*. 3, 5, 7
- Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; and Koltun, V. 2019. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*. 4, 5
- Song, S.; Yu, F.; Zeng, A.; Chang, A. X.; Savva, M.; and Funkhouser, T. 2017. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1746–1754.
- Sun, C.; Sun, M.; and Chen, H.-T. 2021. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2573–2582. 2, 4, 5, 6, 7
- Tan, J.; Lin, W.; Chang, A. X.; and Savva, M. 2021. Mirror3D: Depth Refinement for Mirror Surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15990–15999. 3
- Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR. 3
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008. 3
- Wang, C.; Lucey, S.; Perazzi, F.; and Wang, O. 2019. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, 348–357. IEEE. 4
- Wang, F.-E.; Hu, H.-N.; Cheng, H.-T.; Lin, J.-T.; Yang, S.-T.; Shih, M.-L.; Chu, H.-K.; and Sun, M. 2018a. Self-supervised Learning of Depth and Camera Motion from 360 Videos. In *Asian Conference on Computer Vision*, 53–68. Springer. 1, 3
- Wang, F.-E.; Yeh, Y.-H.; Sun, M.; Chiu, W.-C.; and Tsai, Y.-H. 2020a. BiFuse: Monocular 360 Depth Estimation via Bi-Projection Fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 462–471. 2, 3, 5, 6, 7
- Wang, N.-H.; Solarte, B.; Tsai, Y.-H.; Chiu, W.-C.; and Sun, M. 2020b. 360sd-net: 360 stereo depth estimation with learnable cost volume. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 582–588. IEEE. 1, 3, 4
- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803. 5
- Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612. 4

- Xian, W.; Li, Z.; Fisher, M.; Eisenmann, J.; Shechtman, E.; and Snavely, N. 2019. UprightNet: geometry-aware camera orientation estimation from single images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9974–9983. [2](#)
- Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, 802–810. [2](#)
- Zeng, W.; Karaoglu, S.; and Gevers, T. 2020. Joint 3d layout and depth prediction from a single indoor panorama image. In *European Conference on Computer Vision*, 666–682. Springer. [2](#)
- Zheng, J.; Zhang, J.; Li, J.; Tang, R.; Gao, S.; and Zhou, Z. 2020. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, 519–535. Springer. [1](#), [2](#), [8](#)
- Zioulis, N.; Karakottas, A.; Zarpalas, D.; Alvarez, F.; and Daras, P. 2019. Spherical view synthesis for self-supervised 360° depth estimation. In *2019 International Conference on 3D Vision (3DV)*, 690–699. IEEE. [1](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- Zioulis, N.; Karakottas, A.; Zarpalas, D.; and Daras, P. 2018. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 448–465. [1](#), [2](#), [5](#), [6](#), [7](#)