This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# Rethinking Supervised Depth Estimation for 360° Panoramic Imagery

Lu He, Bing Jian, Yangming Wen, Haichao Zhu, Kelin Liu, Weiwei Feng, Shan Liu
Tencent America, Palo Alto, USA
lhluhe, bingjian, yangmingwen, haichaozhu, kelinliu, wfeng, shanl@tencent.com

## Abstract

*Depth estimation from a single 360° panorama image is a difficult task. It is an ill-posed problem to estimate depth maps from an RGB panorama image due to the intrinsic scale ambiguity issue. To mitigate the scale inconsistency issue in the ground truth depth map, we propose a simple yet effective method to normalize the depth data based on estimated camera height. In addition, we design a multiple head planar-guided depth network, to provide more geometric constraints for depth estimation. Experimental results show that our relative depth estimation task is more accurate than the absolute depth estimation task, and our proposed model produces state-of-the-art performance on both Matterport3D and Stanford2D3D datasets.*

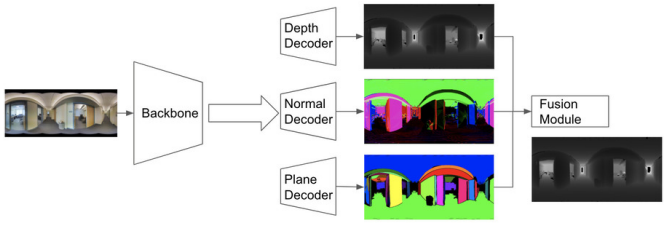



Figure 1. An overview of the proposed system. For the supervised learning of depth estimation for 360 images, we propose to preprocess the ground truth depth map to address the scale inconsistency issue. In addition, we derive geometry information such as planes, normals from ground truth depth data, then integrate those information into a multi-task neural network to jointly train a depth estimation model in an end-to-end fashion.

## 1. Introduction

Dense depth estimation is a fundamental task in 3D computer vision, as it aims to provide the essential information towards general 3D scene understanding from 2D images. Applications of depth estimation range from object and scene reconstruction, robotic navigation, augmented reality, and autonomous driving, etc. Most existing depth estimation methods take input images captured from conventional perspective cameras, which only offer a limited field of view (FOV). On the other hand, the ability to record and sense surrounding views is instrumental in various applications, to name a few, autonomous navigation and virtual reality.

With the emergence of compact 360° cameras in the consumer market, 360° images (or spherical images) that provide a 360° field of view can now be generated and acquired much more easily. Together with the popularization of VR technology and the trend of creating so-called digital twins, these 360° images have now been increasingly used in a wide variety of industries, e.g., real estate, tourism, entertainment, building and construction, and so forth. This rapidly increasing trend of production and consumption of 360° images in turn motivates many recent research works in processing and understanding 360° images and videos.

The focus of this paper is to estimate the depth of a single panorama image, in particular, the equirectangular projection (ERP) of the spherical imagery. Note that unlike in normal perspective settings where depth map is defined as the distance to the imaging plane, i.e., the z coordinate, here in spherical imagery, the depth is defined as the radial distance $d = \sqrt{x^2 + y^2 + z^2}$ to the camera center which is used as the origin $(0, 0, 0)$.

Similar to the depth estimation in perspective views, depth estimation in 360° images also has the intrinsic scale ambiguity issue, even though they possess a whole field of view. Imagine two rooms with an identical layout except that one is 1.5 times bigger (further, taller, deeper) than the other in all directions. The resulting panoramas will be identical. Related to this scale ambiguity issue is the ability to generalize on different camera heights. Panoramas captured with different camera heights may have similar visual looks in some regions, e.g., the nearby floor region, but the ground truth distances could be quite different. This creates two issues. First, mixing image data captured with different camera heights may cause difficulty in training. Second, training only on data with a fixed camera height will not generalize well when testing on data acquired with different camera heights. In this paper, we propose a depth normalization approach to mitigate these issues. More details

about the normalization are given in Section 3.1

In addition, we derive geometry information such as planes and normals from ground truth depth data, then integrate that information into a multi-task neural network to jointly train a depth estimation model in an end-to-end fashion. An overview of the proposed idea is shown in Figure 1. Experimental results are shown in Section 4.

The key merits and contributions of this work are:

1). We design a multi-head network for depth estimation from a single 360° panorama image by exploiting the underlying geometry relation between depth values, point normals, and planar structures. Our proposed network achieves the state-of-the-art results on two standard benchmark datasets: Matterport3D [3] and Stanford2D3D dataset [1].

2). We dive into the depth inconsistency issue in the benchmark panorama datasets that may negatively affect any supervised learning based method for the depth estimation task. We also point out the need to have additional evaluation metrics that are more indicative for distinguishing the performance discrepancy when the desired accuracy level is high.

3). We propose a simple yet effective depth GT normalization method to mitigate the scale ambiguity issue. The experiments show that training on our normalized depth map could significantly improve the accuracy of panorama depth estimation.

## 2. Related work

### 2.1. Monocular depth estimation

Monocular depth estimation is the task of predicting a dense depth map for a given single RGB image (by default, a normal perspective image). Most traditional methods leverage the multi-view geometry and hence require image pairs or image sequences to calculate depth values of sparse correspondences. Recently, with the rapid development of deep learning and the collection of large amount of annotated data [1,10,22], promising results for monocular depth estimation, an traditional ill-posed problem, have been reported in [2,9,11,16,17], to just name a few. Eigen et al. [9] introduce a first deep learning based method for monocular depth estimation with a multi-scale network. MonoDepth [11] proposes an end-to-end unsupervised monocular depth estimation algorithm with left-right depth consistency loss, which requires binocular images during the training stage. BTS [17] proposes hidden local planar guidance layers located at multiple stages in the decoding phase.

### 2.2. Normal estimation and plane detection

Normal estimation and planar structure detection are two tasks closely related to depth estimation due to the underlying geometric relations. Wang et al. [27] use convolutional networks for the task of estimating surface normal from a single image. Eigen and Fergus [8] extend their multi-scale convolutional network [9] to surface normal estimation and semantic segmentation tasks.

PlaneNet [20] presents the first end-to-end neural architecture for piece-wise planar reconstruction from a single RGB image. PlaneRCNN [19] uses the object detection architecture Mask RCNN [12] to assist the depth estimation.

### 2.3. Panorama depth estimation

Garanderie et al. [6] proposed a domain adaptation approach by retraining existing architectures on panoramic images. They transformed the KITTI dataset [10] into partial panoramic images and then adapted the self-supervised learning method [11] in panoramatic settings. Tateno et al. [25] trained a CNN using perspective datasets with GT depth annotations but replaced the standard convolution with a distortion-aware convolution when running inference on panorama images.

Zioulis et al. [32] synthesized a large-scale dataset for indoor panorama depth estimation via scene rendering from four existing realistic datasets and computer generated datasets. Using the synthesized depth map as ground truth, they implemented an end-to-end supervised learning approach for panorama depth prediction with two kinds of encoder-decoder networks: UResNet with strided convolutions and RectNet with dilated convolutions. Eder et al. [7] presented a CNN for predicting depth, surface normal, and planar boundaries from a single indoor panorama image, assuming each scene is piecewise-planar. Jin et al. [14] proposed to predict structure information and depth jointly where structure information such as corners, boundaries, and planes are used as both a prior and a regularizer for indoor depth estimation.

Wang et al. [26] designed a two-branch neural network by fusing features from both equirectangular and cubemap projections. Zeng et al. [29] proposed to first predict a so-called layout depth map through an intermediate semantic segmentation and coarse depth estimation, followed by 3d layout estimation and depth refinement. Sun et al. [24] presented a multi-task framework for layout prediction, depth estimation, and semantic segmentation of an indoor 360-degree panorama using a Latent Horizontal Feature representation.

## 3. Methodology

In this section, we first introduce the proposed planar-constraint model in Section 3.1. Then, we describe in more details on how we extract and segment planes from panoramic depth data in Section 3.2. Section 3.3 introduces the multiple loss function. In Section 3.4, we elaborate the

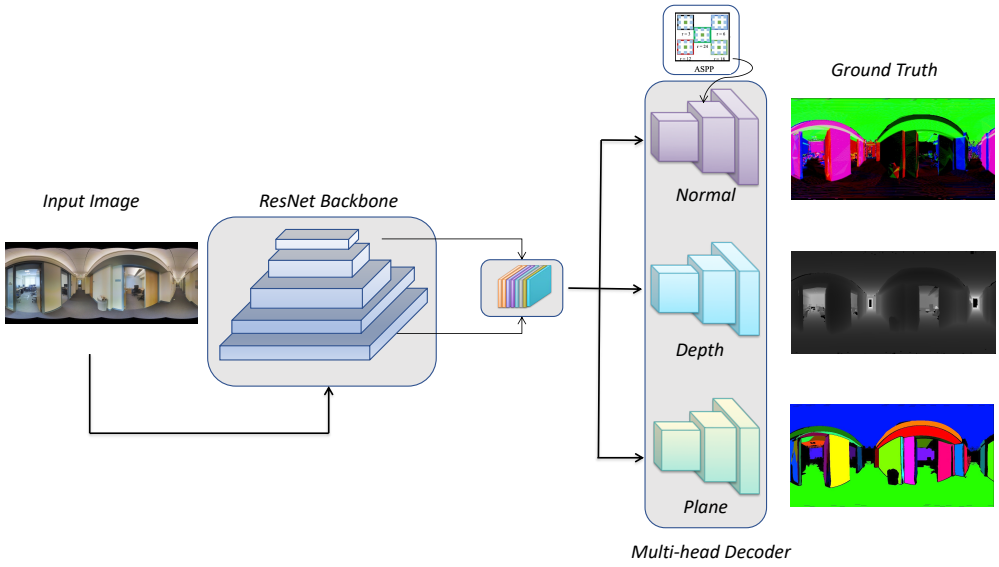



Figure 2. An overview of the whole system with normal and plane constraints. The input panorama is resized to $(1024 \times 512)$. A shared feature pyramid is extracted from the different stages of the backbone (e.g., ResNet50). A shared feature map goes to different decoders for three tasks: depth, normal, and plane.

depth map inconsistency issue for panorama images, and provide a normalization method to mitigate this issue.

### 3.1. System Overview

The detailed overview of the whole system is shown in Figure 2. It consists of 3 parts: 1) input 360° panorama image, 2) backbone, 3) multiple head architecture with geometric constraints.

The input image of our system is the standard equirectangular projection (ERP) of a panorama image. In the ERP projection, the $x$ coordinate of the image pixel represents the longitude in sphere coordinates, and the $y$ coordinate represents the latitude in sphere coordinates. We expect the input panoramas have the full $360° \times 180°$ view (2:1 ratio) to capture the full spherical projection. In particular, the panorama images fed to our system are resized to $1024 \times 512$, which is the same as in other papers [24, 26].

ResNet [13] with fully pyramid network (FPN) [18] is selected as our backbone for a fair comparison with other methods. Some intermediate features from different ResNet stages are selected as the multiple scale encoding features, and then the multiple scale encoding features are used as the inputs of the shared multiple-head features. Also, we adapt atrous convolution in DeepLab V3 [4] as the typical decoder.

Our multi-head system consists of a normal segmentation subsystem, a planar regression subsystem, and a depth estimation subsystem.The normal segmentation subsystem segments the normal into three classes, namely, 'horizontal' class, 'vertical' class, and 'other' class. The planar mask subsystem is a bottom-up anchor-free detection network [5] that includes planar center head, planar offset head. The planar center head estimates the center of planar, and the planar offset head estimates the boundary of each mask. The details are introduced in Section 3.3. The depth subsystem estimates the depth map.

### 3.2. Normal and plane generation

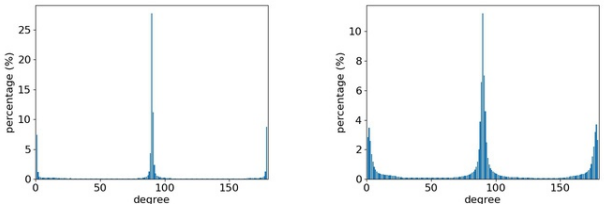
In this section, we describe how the planar annotations are generated. We first convert the depth map to a point cloud. Then we estimate the surface normals from the point cloud data. Finally, planes are segmented by an iterative robust plane fitting procedure similar to [21].

Note that the depth map here is also stored in the ERP representation as to the input panorama RGB image. The depth value represents the distance $d$ of each 3D world point to the camera center $(0, 0, 0)$. Given depth value $d$, the 3D coordinate of the world point corresponding to the spherical coordinate $(\varphi, \theta)$ can be calculated by Equation 1.

$$\begin{cases} x &= d * \sin\varphi * \cos\theta \\ y &= d * \sin\varphi * \sin\theta \\ z &= d * \cos\varphi \end{cases} \tag{1}$$

where $\vec{p} = (x, y, z)$ is the point in the point cloud, and $\varphi$ and $\theta$ are the latitude and the longitude in the spherical coordinate system.

After obtaining the point cloud, we estimate the vertex normals by fitting a local plane at the neighborhood of each




(a) Stanford2D3D (b) Matterport3D

Figure 3. The histogram of angles between normal vectors and gravity direction. The horizontal planes have angles close to 0 degree, and the vertical planes have angles close to 90 degree.

point. With the normal information, robust plane fitting is conducted, and neighboring points are assigned to the same plane instance. This procedure is iterated until all points have been assigned or the maximum number of plane instances have been detected. We generate up to 40 plane instances (sorted by the number of points) per each panorama in our practice.

### 3.3. Multiple head subsystem

A multiple head subsystem is used in our depth estimation system. They are normal decoder, planar decoder, and depth decoder. In this section, we will describe those decoders.

**Normal Head**: We calculate the histogram of angles between normal vectors and gravity direction. The results on two benchmark datasets are shown in Figure 3. Planes with an angle in the degree range of [85,95] are classified as horizontal planes, and planes with an angle in the degree range of [0,5] or [175,180] are labeled as vertical planes. We observe that 25.9% of the points lie in vertical planes, 42.2% are in horizontal planes, and 31.9% are others in the Matterport3D [3] dataset. A similar distribution is found in the Stanford2D3D [1] dataset. Based on the distribution of the normal angles, we split planes into three classes, horizontal, vertical, and others. A three-label classification is used to train the normal classification decoder. The loss function for the normal classification decoder is multi-class cross-entropy loss defined in Equation 2.

$$l_n = -\sum_i y_{n,i} \log y^*_{n,i}. \tag{2}$$

Note that nearly all the 360° image data in these benchmark datasets are already well-aligned with gravity. It has been observed that the model could have degraded performance on input 360° images that are not well-aligned. We do recommend that in practice, a preprocessing step is employed to ensure the gravity alignment as done in [30, 33].

**Plane Head**: The plane regression decoder is an anchor-free plane proposal network [5]. The generated plane mask with plane instances information is required for this head.

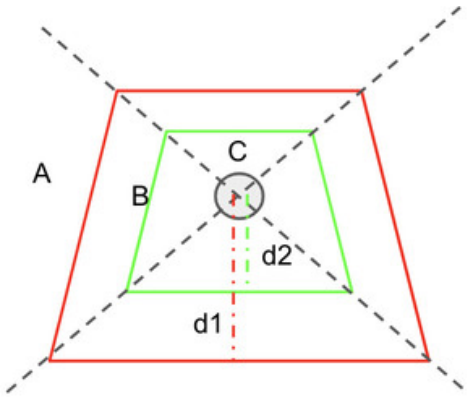



Figure 4. Illustration of the depth ambiguity issue in panoramic images. (better viewed in color)

We convert the plane instances to two parts on-the-fly. One is the center of each mask, and the other is the offsets of each pixel in the mask to its own center. The center of each mask is defined as the center of the bounding box enclosing that mask, and the bounding box is obtained by finding the minimum rectangle to cover the plane mask. For the offset, we calculate the offset between each pixel to its mask center. The loss of the plane regression is defined in Equation 3.

$$l_p = w_c \sum_i ||y_{c,i} - y^*_{c,i}||_2 + w_o \sum_j ||y_{o,j} - y^*_{o,j}||_1. \tag{3}$$

The plane regression loss is a combination of a center loss and an offset loss. The center loss is the mean square loss, and the offset loss is the L1 loss. $w_*$ are the weights.

**Depth Head**: The depth estimation decoder is a regression head. The ground truth is the depth map. We ignore the pixels that are greater than 10 meters during training. L1 loss is selected for depth estimation:

$$l_d = \sum_i ||y_{d,i} - y^*_{d,i}||_1 \tag{4}$$

**Combined Loss**: The total loss of the whole system is the weighted sum of normal loss, plane loss, and depth loss:

$$loss = w_n l_n + w_p l_p + w_d l_d. \tag{5}$$

We train the system in an end-to-end manner.

### 3.4. Normalized annotation

This section explains the rationale behind the normalized depth annotation and how we prepare the depth normalization annotation.

Similar to the monocular depth estimation for perspective images, depth estimation from a single panorama image is also an ill-posed problem.

| GT Distance (meter) | $\delta^{0.25}$ | $\delta^{0.5}$ | $\delta^{1}$ |
|---|---|---|---|
| 1.0 m | (0.94, 1.06) | (0.89, 1.12) | (0.8, 1.25) |
| 2.0 m | (1.89, 2.11) | (1.79, 2.24) | (1.6, 2.5) |
| 3.0 m | (2.84, 3.17) | (2.68, 3.35) | (2.4, 3.75) |
| 4.0 m | (3.78, 4.23) | (3.58, 4.47) | (3.2, 5) |

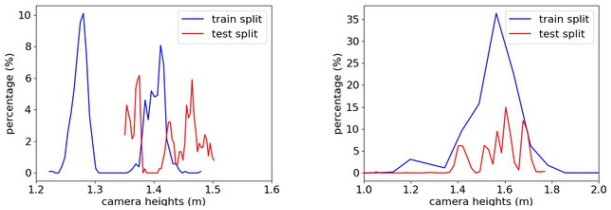
Table 1. Distance ranges of $\delta^k$ with various thresholds ($\delta^{0.25}$, $\delta^{0.5}$, $\delta^1$) for ground truth distances at 1 m, 2 m, 3 m, 4 m respectively.

Figure 4 is a toy example to illustrate the depth ambiguity issue. Center $C$ is a panorama camera center. The green trapezoid $B$ is smaller than the red trapezoid $A$. Both trapezoids are centered at $C$. Assuming that any two points on these two trapezoids have the same color when they are collinear with the camera center $C$. Then green trapezoid $B$ and red trapezoid $A$ will have the same panorama image if captured individually without blocking each other, but it is clear that the actual depth values are very different.

Therefore it is very likely that two panoramas have similar visual looking in some regions, but the ground truth distances could be quite different. Because of this data inconsistency issue, the learning procedure may have difficulty converging or learning the average value of the inconsistent depths to minimize the regression loss. In order to mitigate this data inconsistency issue, we propose to normalize the ground truth depth annotation based on the camera height.

Given a ground truth depth map, the camera height is estimated by the following algorithm: 1) find the horizontal planes at the bottom of the panorama. 2) calculate the vertical distance $z$ for each point in the horizontal planes found in 1) by Equation 1. 3) perform a robust estimate method to get the camera height $h$. Finally, the ground truth depth map is normalized by dividing $h$ such that the normalized depth maps have a consistent unit camera height. Note that there is no need to recompute normal and plane segmentation after the depth normalization.

It is worth noting that normalization with respect to the camera height may slightly affect the depth estimation quality for the ceiling area in the indoor setting, which we also noticed during error analysis. We choose to normalize the depth using the camera height (the distance from the camera to the ground) instead of the distance from the camera to the ceiling because 1) it is applicable in both indoor and outdoor settings; 2) for many applications, a lower region usually attracts more attention than the upper region of the scene. Therefore the accuracy gain of depth estimation in the lower region outweighs the potential accuracy degradation in the upper region; 3) camera height can be more easily recorded and provided to the downstream 3D reconstruction tasks to recover the actual scale. /




(a) Stanford2D3D (b) Matterport3D

Figure 5. Histogram of camera heights in Matterport3D and Stanford2D3D dataset.

## 4. Experimental Results

In this section, we conduct experiments on Matterport3D [3] and Stanford2D3D [1]. Quantitative and qualitative comparisons with other state-of-the-art algorithms are presented in Section 4.4. Then we report results from models trained on our new normalized annotation in Section 4.5.

### 4.1. Datasets

Matterport3D [3] dataset is a large-scale real-world dataset that contains 10,800 indoor panoramas from 90 houses. Following BiFuse [26] and HoHoNet [24], we use 7829 panoramas from 61 houses for training, and the remaining 2971 panoramas from 29 houses are reserved for testing. Stanford2d3D [1] is a relatively smaller dataset with 1413 panorama images. The training set includes 1040 panorama images, and the 373 panorama images in area-5 are used for testing. All panorama image and depth maps are resized to $512 \times 1024$. The depth maps are truncated at 10 meters.

### 4.2. Evaluation Metrics

Standard depth estimation metrics, including mean relative error (MRE), mean absolute error (MAE), root-mean-square error (RMSE), log-based root-mean-square error (RMSE-log), and threshold based precision $\delta^k$ are used for evaluation. These metrics are defined as follows:

$$\begin{cases} MRE &= \sum |(y-\hat{y})/y|/N \\ MAE &= \sum |(y-\hat{y})|/N \\ RMSE &= \sqrt{\sum (y-\hat{y})^2/N} \\ RMSE(log) &= \sqrt{\sum (\log(y)-\log(\hat{y}))^2/N} \\ \delta^k &= \sum \frac{\max(y/\hat{y}, \hat{y}/y) < 1.25^k}{N} \end{cases} \quad (6)$$

It is worth noting that we introduce two additional $\delta^k$ evaluation metrics, namely $\delta^{0.25}$ and $\delta^{0.5}$, in addition to the conventional metrics $\delta^1$ and $\delta^2$. As shown in Table 1, $\delta^1$ and $\delta^2$ are not able to indicate the accuracy of depth estimation when the desirable error threshold is tight. For example, with a distance of 2 meters, the range of $\delta^1$ is (1.6 m, 2.5 m), which means the predicted distance that is greater than 1.6

| Datset | Method | MRE | MAE | RMSE | RMSE (log) | $\delta^{0.25}$ | $\delta^{0.5}$ | $\delta^{1}$ | $\delta^{2}$ |
|---|---|---|---|---|---|---|---|---|---|
| Matterport3D | FCRN [16] | 0.241 | 0.401 | 0.670 | 0.124 | - | - | 0.770 | 0.917 |
| | OmniDepth(bn) [32] | 0.290 | 0.483 | 0.764 | 0.145 | - | - | 0.770 | 0.879 |
| | Equi [26] | 0.207 | 0.370 | 0.654 | 0.118 | - | - | 0.830 | 0.925 |
| | Cube [26] | 0.250 | 0.393 | 0.663 | 0.128 | - | - | 0.756 | 0.914 |
| | BiFuse [26] | 0.205 | 0.347 | 0.626 | 0.113 | - | - | 0.845 | 0.932 |
| | HoHoNet [24] | 0.149 | 0.286 | 0.514 | 0.087 | - | - | 0.879 | 0.952 |
| | Ours | **0.097** | **0.248** | **0.443** | **0.063** | 0.486 | 0.739 | **0.906** | **0.971** |
| Stanford2D3D | FCRN [16] | 0.184 | 0.343 | 0.577 | 0.110 | - | - | 0.723 | 0.921 |
| | OmniDepth(bn) [32] | 0.200 | 0.374 | 0.615 | 0.121 | - | - | 0.688 | 0.889 |
| | Equi [26] | 0.143 | 0.271 | 0.464 | 0.091 | - | - | 0.826 | 0.946 |
| | Cube [26] | 0.133 | 0.259 | 0.441 | 0.084 | - | - | 0.835 | 0.952 |
| | BiFuse [26] | 0.121 | 0.234 | 0.414 | 0.079 | - | - | 0.866 | 0.958 |
| | HoHoNet [24] | 0.101 | **0.203** | **0.383** | **0.067** | - | - | **0.905** | 0.969 |
| | Ours | **0.098** | 0.209 | 0.394 | **0.067** | 0.464 | 0.728 | 0.903 | **0.974** |

Table 2. Comparison with state-of-the-art methods on Matterport3D dataset (top) and Stanford2D3D dataset (bottom). The evaluation metrics, MRE, MAE, RMSE, RMSE (log), are the lower the better. The $\delta^k$ metrics are the higher the better.

meters and less than 2.5 meters is counted as correct in $\delta^1$ metric. On the other hand, the distance range corresponding to $\delta^{0.25}$ metric is tightened to (1.89 m, 2.11 m), which is definitely more desirable in many downstream tasks.

### 4.3. Implementation details

We implement our model in Detectron2 [28] framework. ResNet-50 is used as backbone for fair comparison with other methods. We train the whole network with a batch size of 16 in 4 GPUs. The learning rate is set to 0.01, and the polynomial learning rate decay is applied.

### 4.4. State-of-the-art comparison using standard datasets

We compare our method with other state-of-the-art algorithms on Matterport3D [3] and Stanford2D3D [1] datasets. Evaluation results are shown in Table 2. Our method produces significant improvement on Matterport3D dataset and competitive results on Stanford2D3D dataset. Note that Matterport3D dataset contains more diversified indoor scenes than Stanford2D3D dataset which was mainly captured in the campus building setting.

It is not surprising that in Table 2 the $\delta^{0.25}$ and $\delta^{0.5}$ metrics are significantly lower than those conventional $\delta$ metrics. We speculate that the large performance gap is partially due to the scale inconsistency issue as described in Section 3.4. We plot the statistics of camera heights in Figure 5. Take Matterport3D dataset as an example. The camera heights in the training set are in the range of (1.2 m,1.5 m), but are in the range of (1.35 m, 1.5 m) in the testing split. Using the mean value of training camera heights 1.35m as the standard camera height, note that the range $\delta^1$ (1.08 m, 1.687 m) covers (1.35, 1.5) but is not the case for $\delta^{0.25}$ (1.277 m, 1.427 m). A similar analysis could also be made on Matterport3D dataset. This probably correlates with the large performance drop from $\delta^1$ to $\delta^{0.25}$ and supports our proposed approach of normalizing depth and computing stricter metrics.

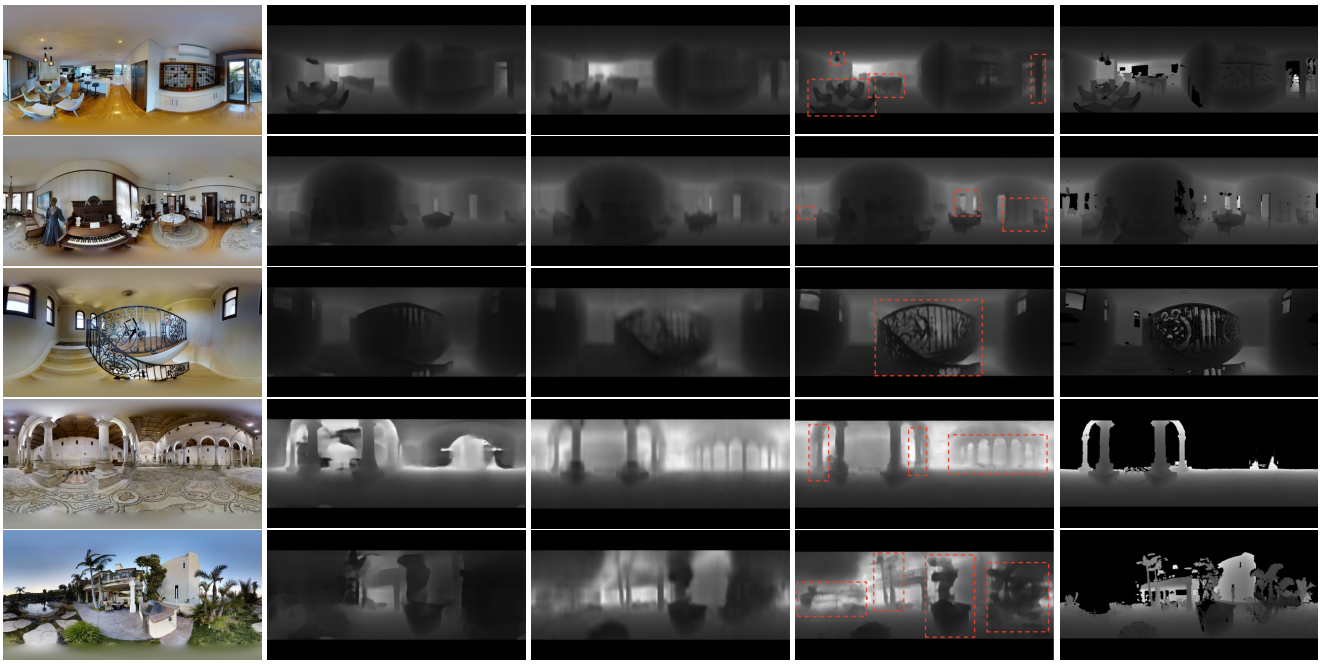
Qualitative results compared with BiFuse [26] and HoHoNet [24] on both Matterport3D dataset and Stanford2D3D datasets are shown in Figure 6 and Figure 7. A few merits of our method are observed in the qualitative comparison: 1) Our predicted depth maps are sharper around the edges. 2). The planar attribute is obvious in our predicted depth map. 3) The depth estimation of main objects in the foreground region is more accurate than other methods. 4) Our method is able to predict the distant depth reasonably well where even the ground truth depth might be missing. More high-resolution depth maps can be found in the supplementary.

We also collect some panorama images from web to check the generalization ability of our method. Note that those data are not in the benchmark dataset and do not have ground truth depth. Qualitative results from our method and other methods are shown in Figure 8.

### 4.5. Results on normalized depth annotations

#### 4.5.1 Comparison with the original datasets

We train our network on our normalized datasets with the same setting on original datasets. For fair comparison, we evaluate our results on $\delta^{0.25}$, $\delta^{0.5}$, and $\delta^1$ as the $\delta$ metrics are scale-invariant. Table 3 shows the results on both original datasets and normalized datasets. As expected, training on normalized datasets leads to better results on $\delta^{0.25}$, and $\delta^{0.5}$.



(a) Input (b) BiFuse (c) HoHoNet (d) Ours (e) Ground Truth

Figure 6. Qualitative results of our method compared to Bifuse [26] and HoHoNet [24] on Matterport3D dataset. We discard distance values greater than 10 meters before mapping them to gray scale for visualization. The red boxes in column(d) highlight regions where our results excel. The first two rows come from indoor scenes; the third row is on stairs; the fourth row is from a grand hall; the last row is from an outdoor scene. More high-resolution depth maps can be found in the supplementary.

(a) Input (b) BiFuse (c) HoHoNet (d) Ours (e) Ground Truth

Figure 7. Qualitative results of our method compared to Bifuse [26] and HoHoNet [24] on Stanford2D3D dataset. We discard distance values greater than 10 meters before mapping them to gray scale for visualization. The red boxes in column(d) highlight regions where our results excel. More high-resolution depth maps can be found in the supplementary.

| Dataset | Method | $\delta^{0.25}$ | $\delta^{0.5}$ | $\delta^{1}$ |
|---|---|---|---|---|
| Matterport3D | Original | 0.486 | 0.739 | 0.906 |
| | Normalized | 0.578 | 0.785 | 0.909 |
| Stanford2D3D | Original | 0.464 | 0.728 | 0.903 |
| | Normalized | 0.600 | 0.787 | 0.910 |

Table 3. Results on original datasets and our normalized counterparts.

#### 4.5.2 Ablation study with normal and planar constraints

We evaluate the improvement with normal head and planar head on our normalized Stanford2D3D dataset in Table 4 using ablation study. Results show that the planar head helped gain an absolute 3.3% improvement (from 0.754 to 0.786) on metric $\delta^{0.5}$, and an absolute 6.9% improvement (from 0.531 to 0.600) on metric $\delta^{0.25}$. The results show that our model with the planar head has a better improvement on a higher accuracy bar than without the planar head.

(a) Input (b) BiFuse (c) HoHoNet (d) Ours

Figure 8. Qualitative results of our method compared to Bifuse [26] and HoHoNet [24] on a few indoor panoramas collected online (not in standard datasets and no ground truth). Predicted depths greater than 10 meters are discarded before mapped to gray scale for visualization. The red boxes in column(d) highlight regions where our results excel. More high-resolution depth maps can be found in the supplementary.

| Method | $\delta^{0.25}$ | $\delta^{0.5}$ | $\delta^{1}$ |
|---|---|---|---|
| Depth baseline | 0.531 | 0.754 | 0.893 |
| Depth w/ normal | 0.574 | 0.779 | 0.904 |
| Depth w/ normal & planar | 0.600 | 0.787 | 0.910 |

Table 4. Ablation study on results with normal head and planar head on our normalized Stanford2D3D dataset.

| Dataset | | $\delta^{0.25}$ | $\delta^{0.5}$ | $\delta^{1}$ |
|---|---|---|---|---|
| Matterport3D | w/o 3D60 | 0.578 | 0.785 | 0.909 |
| | w/ 3D60 | 0.690 | 0.865 | 0.969 |
| Stanford2D3D | w/o 3D60 | 0.600 | 0.787 | 0.910 |
| | w/ 3D60 | 0.689 | 0.861 | 0.953 |

Table 5. Results with 360 datasets.

#### 4.5.3 Benefit of more training data

We also analyze the improvement when more data are available for training. 3D60[1] is a collective dataset generated in recent 360° vision research works [15, 31, 32]. It comprises multi-modal stereo renders of scenes from realistic and synthetic large-scale 3D datasets (Matterport3D [3], Stanford2D3D [1], SunCG [23]). In this experiment, we select the right position and left down position from Matterport3D part in 3D60 dataset, perform same normalization, and append them to the training set. The evaluation results in Table 5 indicate that the more training data are used, the bigger improvement our depth normalization approach could achieve.

## 5. Conclusion

In this paper, we design a multi-head convolutional neural network for panoramic depth estimation. Our model achieves state-of-the-art results on benchmark datasets in terms of standard depth estimation metrics. In the meantime, by adding two stricter metrics $\delta^{0.25}$ and $\delta^{0.5}$, we illustrate that current state-of-the-art models do not work well when the desirable accuracy bar is high. We believe that the performance gap could partially attribute to the scale inconsistency issue presented in the ground truth depth data. Thus, we propose a simple yet effective depth normalization method to address the depth scale inconsistency problem. Experimental results show that models trained on the new normalized depth data yield better performance than those trained on the depth data without normalization.

[1] https://vcl3d.github.io/3D60/

## References


[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016. 2, 4, 5, 6, 8

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, pages 4009–4018, 2021. 2

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017. 2, 4, 5, 6, 8

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for seman-

tic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[5] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 3, 4

[6] Greire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P Breckon. Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery. In *ECCV*, pages 789–807, 2018. 2

[7] Marc Eder, Pierre Moulon, and Li Guan. Pano popups: Indoor 3d reconstruction with a plane-aware network. In *3DV*, pages 76–84, 2019. 2

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015. 2

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 2

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[14] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *ECCV*, pages 886–895, 2020. 2

[15] Antonis Karakottas, Nikolaos Zioulis, Stamatis Samaras, Dimitrios Ataloglou, Vasileios Gkitsas, Dimitrios Zarpalas, and Petros Daras. 360° surface regression with a hyper-sphere loss. In *3DV*, 2019. 8

[16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248. IEEE, 2016. 2, 6

[17] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 2

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[19] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *CVPR*, pages 4450–4459, 2019. 2

[20] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, pages 2579–2588, 2018. 2

[21] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum*, 26(2):214–226, 2007. 3

[22] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 2

[23] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 190–198, 2017. 8

[24] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *CVPR*, pages 2573–2582, 2021. 2, 3, 5, 6, 7, 8

[25] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *ECCV*, pages 732–750, 2018. 2

[26] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *CVPR*, pages 462–471, 2020. 2, 3, 5, 6, 7, 8

[27] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, pages 539–547, 2015. 2

[28] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[29] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Joint 3d layout and depth prediction from a single indoor panorama image. In *ECCV*, pages 666–682, 2020. 2

[30] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *ECCV*, pages 668–686, 2014. 4

[31] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. Spherical view synthesis for self-supervised 360° depth estimation. In *3DV*, 2019. 8

[32] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *ECCV*, pages 448–465, 2018. 2, 6, 8

[33] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360 image: A comparative study of state-of-the-art methods. *Int. J. Comput. Vis.*, 129(5):1410–1431, 2021. 4