BGDNet: Background-guided Indoor Panorama Depth Estimation

Jiajing Chen*†1, Zhiqiang Wan*2, Manjunath Narayana[2], Yuguang Li[2], Will Hutchcroft[2],
Senem Velipasalar[1], and Sing Bing Kang[2]

[1]Syracuse University
[2]Zillow Group

Abstract

Depth estimation from single perspective image has received significant attention in the past decade, whereas the same task applied to single panoramic image remains comparatively under-explored. Most existing depth estimation models for panoramic images imitate models proposed for perspective images, which take RGB images as input and output depth directly. However, as demonstrated by our experiments, model performance drops significantly when the training and testing datasets greatly differ, since they overfit the training data. To address this issue, we propose a novel method, referred to as the Background-guided Network (BGDNet), for more robust and accurate depth estimation from indoor panoramic images. Different from existing models, our proposed BGDNet first infers the background depth, namely from walls, floor and ceiling, via background masks, room layout and camera model. The background depth is then used to guide and improve the output foreground depth. We perform within dataset as well as cross-domain experiments on two benchmark datasets. The results show that BGDNet outperforms the state-of-the-art baselines, and is more robust to overfitting issues, with superior generalization across datasets.

1. Introduction

Depth estimation is a classic computer vision task with wide-ranging applications, including autonomous driving, 3D reconstruction, and simultaneous localization and mapping (SLAM). Most existing work on single image-based depth estimation focus on perspective images, while panoramic images remain under-explored. Panoramic images can capture more scene information with a wider Field of View (FoV) than perspective images, and have become more popular and accessible with the availability of 360° cameras. Recently, research on panoramic images has attracted increasing attention from both industry and academia.

Current state-of-the-art (SOTA) deep learning-based models designed for single panoramic image depth estimation, adopt an end-to-end network structure, similar to the depth estimation models developed for perspective images. In these end-to-end models, the network takes an RGB image as input and directly outputs the depth prediction [1, 19]. Although different methods have been proposed focusing mainly on the distortion problem of panoramic images [13, 17, 20, 22], they still suffer from severe overfitting when the training and test sets differ significantly. *Depth estimation is a pixel-wise regression task, and the network is prone to failure especially when unseen foreground/furniture or room layout variations appear in testing scenarios*. Existing panoramic image datasets for depth estimation are either rendered from Computer-Aided Design (CAD) models [23] or collected from a small number of buildings (around tens of buildings) and/or cover limited types of rooms in the real world [2, 3, 16]. Thus, when a model trained on these datasets is used in real-world applications, where testing data is highly likely to be dissimilar to training data, the overfitting problem gets emphasized.

In order to address the aforementioned issues, we propose a novel method, referred to as the background-guided network (BGDNet) for panoramic image depth estimation. *Different from the conventional methods of directly estimating each pixel depth from RGB images, our method first infers the background depth through segmented background masks and the panoramic camera model.* The pipeline of the proposed BGDNet is shown in Fig. 1. we leverage the Segment Anything Model (SAM) [10] in one of the branches to obtain raw segmentation masks of the scene. Since SAM has been trained on 11 million images and over 1B masks, it has strong generalizability to broader test-

*Equal contribution.
†This work was done when Jiajing Chen was an intern at Zillow.
[1] {jchen152,svelipas}@syr.edu
[2] {zhiqiangw,manjun,yuguangl,willhu,singbingk}@zillowgroup.com

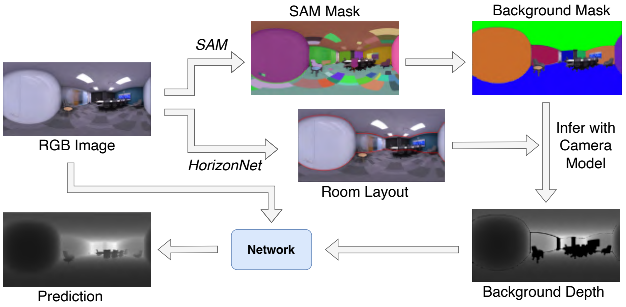

Figure 1. The pipeline of our approach. Instead of letting the network perform depth estimation directly on the whole image, our method first estimates the background depth, and then uses it to guide the final prediction of the panorama depth map.

ing scenarios. Then, our method takes SAM masks as input and extracts the background (floor, ceiling, and wall) masks. In a second branch, we use HorizonNet [18] to obtain the room layout, specified by ceiling-wall and floor-wall boundaries. Compared to depth estimation directly from RGB images, which requires performing regression on each pixel, room layout estimation is relatively easier and more robust to variations in test images. Given the background mask and the room layout, our proposed approach directly computes the background depth by using the panoramic camera model. The computed background depth and RGB image are fed into a network to predict the depth for the whole image. During testing, we replace a part of network output with depth from Background depth map obtained from SAM and HorizonNet, to further improve our method performance. Our experiments on different datasets show that our proposed model provides better and more stable results than four SOTA baselines in terms of commonly used metrics, such as Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). The main contributions of this work are:

- We provide a motivation for our approach by analyzing the overfitting problem of the existing methods with a set of experiments.
- We propose a novel network, BGDNet, which first infers the background depth by strategic use of segmented masks and room layout, and then uses the background depth to guide the depth estimation. During testing, the network performance is further improved with our proposed Background Depth Replacement module.
- We perform within dataset and cross-domain experiments on two different datasets. The results show that our proposed BGDNet outperforms four SOTA baselines, namely HRDFuse [1], HoHoNet [19], FCRN [11] and OmniFusion [13], in terms of commonly used metrics.
- We perform a series of ablation studies to further show the effectiveness of different components of our approach. These studies also show that the inferred background depth can indeed alleviate the overfitting problem.

2. Related Work

Depth Estimation from Single Perspective Images. An early Convolutional Neural Network (CNN)-based depth estimation work [7] employs two branches to perform depth estimation from single perspective image. In one branch, a series of convolutional layers are used to predict a coarse depth map. Another branch uses the coarse depth map as input and outputs a refined depth map. Since then, different methods have been proposed using end-to-end deep learning for depth estimation. A CNN architecture is used in [6] to perform depth prediction, surface normal estimation, and semantic labeling. A two-streamed network is presented in [12] to estimate fine-scaled depth maps. This method predicts depth and depth gradients, which are then fused to obtain a detailed depth map. Other methods [9, 14] adopt Generative Adversarial Networks (GANs) for depth estimation. Very recently, iDisc [15] was proposed to perform depth estimation by partitioning the scene into a set of parts sharing similar features.

Depth Estimation from Single Panoramic Images. Bifuse [21] uses features from equirectangular and cube map projections for monocular 360° depth estimation. Different from Bifuse, Unifuse [8] fuses the equirectangular and cube map features only at the encoding stage, and claims that equirectangular features are more important than cube map features for depth prediction. HoHoNet [19] claims that column features play an important role in panoramic image feature representation, and uses an approach based on proposed Latent Horizontal Feature for depth estimation. OmniDepth [24] addresses the distortion problem in panoramic images with row-wise rectangular filters. OmniFusion [13] first transforms a panoramic image into less-distorted perspective patches, and then performs depth estimation on each patch. The final output is obtained by merging each patch's output. A more recent, SOTA method, HRDFuse [1] employs both CNNs and transformers to learn features from equirectangular projection and tangent projection for final depth estimation.

3. Motivation

Taking a single RGB image as input, the depth estimation task requires the model to predict the distance of each pixel to the camera. In this section, we first perform experiments on the Replica dataset [16] to illustrate the overfitting problem of four different baseline depth estimation models. The Replica dataset contains 3D indoor meshes collected from the real-world, and 3554 panoramic images rendered from these 3D meshes. As shown in Fig. 2, we use two different test sets, Test Set 1 and Test Set 2, generated from *Room_0* and *Apartment_0*, respectively. Data from the remaining rooms (office, hotel, apartment, etc.) in the dataset are used

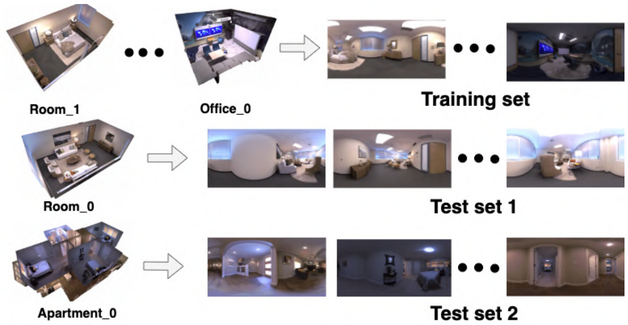

Figure 2. Two different test sets are generated from *Room_0* and *Apartment_0*, respectively. *Room_0* is similar to several other rooms in the training set, such as *Room_1*, in terms of foreground and furniture layout. Yet, *Apartment_0* is more different from rooms in the training set.

for training. *Room_0* is a single room that is similar to several other rooms (which have been used to generate the training images) in terms of foreground and furniture layout. However, *Apartment_0* contains multiple rooms, which are different from most rooms used to generate the training set. Thus, Test Sets 1 and 2 constitute a good example to measure the severity of overfitting, and evaluate generalization ability by comparing the performance of different well-trained models on these test sets.

The performances of four different benchmarks on Test Set 1 and Test Set 2 are listed in Table 1. We use three established metrics, namely Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and δ_1 value to evaluate the performance. As can be seen, the performance of all the benchmark approaches drops significantly on Test Set 2 compared to Test Set 1. Taking the SOTA method HRDFuse [1] as an example, the MAE value increases by **298%** from 0.0584 to 0.2325. A similar trend is observed with the other metrics as well. This is a typical problem for real-world applications, since it is difficult to ensure that the training set spans all possible variations in layout and room type, as well as furniture type and arrangement. As a result, pixel-wise depth estimation is prone to overfit to a given training set, leading to significant performance degradation for uncovered test scenarios.

Method	Set	MAE↓	RMSE ↓	δ_1↑
FCRN[11]	Test Set 1	0.0755	0.1758	0.9718
	Test Set 2	0.2851	0.4586	0.6229
HoHoNet[19]	Test Set 1	0.0588	0.1653	0.986
	Test Set 2	0.2295	0.3925	0.7893
OmniFusion[13]	Test Set 1	0.1242	0.2526	0.9608
	Test Set 2	0.2761	0.5125	0.6917
HRDFuse[1]	Test Set 1	0.0584	0.1695	0.9889
	Test Set 2	0.2325	0.4108	0.7469

Table 1. Performance of several benchmark models on two different test sets. Performance of all baselines degrade significantly when they are tested on Test Set 2, which is more different than anything in the training data, compared to Test Set 1.

4. Proposed Method

To address the aforementioned issue, we propose Background-guided Network (BGDNet). The pipeline of our proposed method is shown in Fig. 3. Our method first estimates the background depth D_{Bg} by our proposed background depth estimation module. Then, with the guidance of background depth, our method performs depth estimation task on the whole image. Finally, a background depth replacement module is designed to further improve the performance by replacing network output with an accurate part of D_{Bg}.

4.1. Background Depth Estimation Module

We now explain how we directly compute the background depth with panoramic camera model. We assume known camera height, which can easily be obtained during image capture [4].

4.1.1 Background Segmentation

In a panoramic image, the background, which is composed of ceiling, floor, and wall, occupies a significant portion of the image. The backgrounds of indoor scenes are usually composed of horizontal or vertical planes. Thus, their depths can be directly computed from background masks with the panoramic camera model. In order to segment the background region in a panoramic image, we adopt the Segment Anything Model (SAM)[10]. We employ SAM, since it was trained on 11 million images, and with over 1B masks, giving the model the ability to provide stable prediction under various test scenarios.

However, SAM can only output masks without labels. To solve this problem, we first feed an RGB image into SAM to obtain raw masks. In panoramic images of indoor scenes, the floor and ceiling cover a significant portion of the image and are bordered by the bottom and top image boundaries, respectively. Based on this observation, we iterate through each of the SAM output masks and mark a mask as a 'potential floor mask' if it contains at least f-many pixels, that are on or at most d-pixels away from the bottom image boundary. We set $f = W/6$ and d=20 pixels in our experiments, where W denotes the image width. Then, we designate the potential floor mask with the largest area as the final floor mask. We obtain the ceiling mask in a similar way, this time using the top image boundary. Subsequently, we go over the remaining masks, and classify a mask as a wall mask if it satisfies one of the following conditions:

- The mask connects to *both* floor and ceiling masks.
- The mask only connects to the floor or ceiling mask, but it crosses the horizon line of the image.

Although some large furniture items, such as refrigerators, may also satisfy these conditions, our Background Depth Estimation Module, introduced in Sec. 4.1, is applicable to them, since such large furniture items are typically

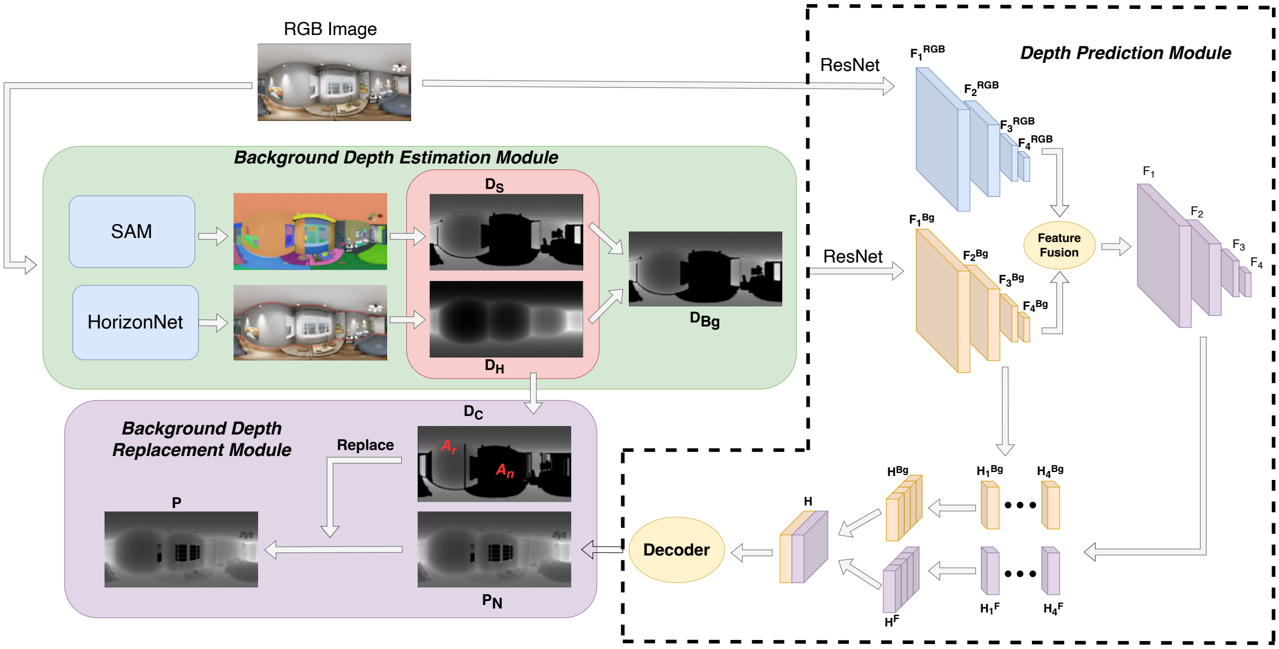

Figure 3. The pipeline of the proposed Background-guided Network (BGDNet). The RGB image is firstly fed into the background depth estimation module to obtain the background depth map D_S and D_H respectively. Then, taking advantage of D_S and D_H, we obtain D_{Bg}. The RGB image and D_{Bg} are sent to two separate backbones, and their features are fused to perform the prediction to obtain P_N. Then, we average the depth of D_S and D_H in the area A_r wherein their difference is less a threshold, to obtain D_C. The depth value in A_r of network prediction P_N is replaced with corresponding value in D_C, to obtain the final prediction P.

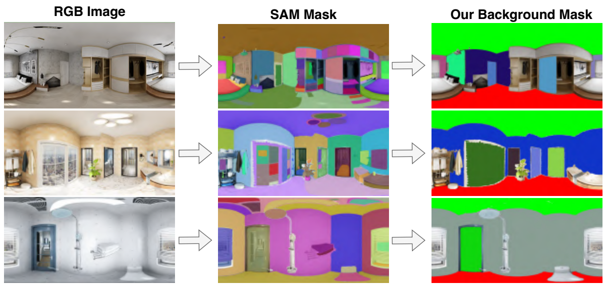

Figure 4. **Left:** Input RGB images, **mid:** masks outputs by SAM, **right:** background masks obtained by our method. As examples show, our background segmentation module can segment ceilings, floors, and most of the walls successfully.

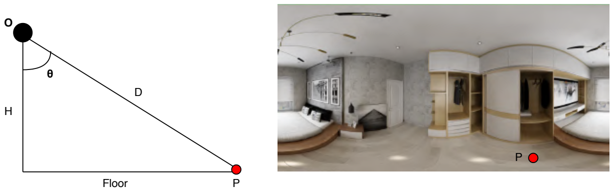

Figure 5. For any point P on the floor, the distance D from camera O to P can be calculated from the camera height H and angle θ.

composed of planes. The segmentation pipeline and results are shown in Fig. 4, where different colors on the walls represent different masks classified as wall segments.

4.1.2 Floor Point Depth Estimation

Given any point P on the floor, the longitude and latitude angle of point P can be inferred based on its image coordinates. Then, angle θ, shown in Fig. 5, can be calculated as described in detail in [18]. Since we assume the camera height H is known, the absolute depth D of P could be calculated as $D = \frac{H}{cos(\theta)}$.

4.1.3 Ceiling Point Depth Estimation

To calculate the depth of ceiling points, we first need to infer the distance from the camera to the ceiling plane. Once we have this distance, we can calculate the depth of any point on the ceiling similarly to how we obtain floor point depth. In Sec. 4.1.1, we described how to segment the wall masks. For all wall masks that connect to *both the ceiling and floor masks*, we scan through all the pixel columns in that mask. Fig. 6 shows a sample pixel column AB connecting to both ceiling and floor masks, where A and B are on the wall-ceiling and wall-floor boundary, respectively. Angles ϕ_1 and ϕ_2 are obtained from the pixel coordinates of points A and B. Given the camera height BC, we can first calculate OC as $OC = \frac{BC}{tan(\phi_2)}$, and then the ceiling distance as $AC = OC \cdot tan(\phi_1)$. In a panoramic image, we can compute multiple ceiling distances from multiple pixel columns, connecting to both floor and ceiling masks. In this case, we take the median of these inferred ceiling distances as the final distance value from the camera to the ceiling plane. Once we obtain the ceiling distance, the ceiling point depth could be inferred similarly to how we obtain the floor point depth, described in Sec. 4.1.2.

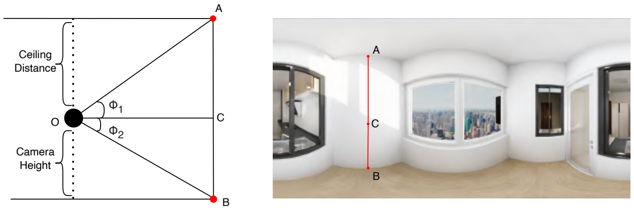

Figure 6. AC is the distance between the camera horizon point and ceiling. Since the image coordinates of A and C are known, angles ϕ_1 and ϕ_2 can be inferred by their longitude and latitude angles.

4.1.4 Wall Point Depth Estimation

Wall masks connect to either the floor, the ceiling, or both. In Sections 4.1.2 and 4.1.3, we described how to obtain the depth of each point on the floor and ceiling, respectively, and the connection points, such as A and B shown in Fig. 6. Given this information, we can infer the depth of each point on the pixel columns of all wall masks, based on the assumption that the wall is perpendicular to the ceiling and floor. For example, with known lengths of OA, OC and OB (Fig. 6), we can easily calculate the distance from any point on line AB to camera O.

4.1.5 Combined Background Depth Estimation

Since we now have the inferred depths of the wall, ceiling, and floor, based on the masks output of SAM, one can simply merge them into the background depth map D_S of the panoramic image (as shown in the top branch of Fig. 7). However, although SAM provides great segmentation output for most parts, the wall-ceiling and wall-floor boundary pixel coordinates, such as Points A and B shown in Fig. 6, may still suffer from errors according to our experiment outputs. These errors propagate into the calculation of the distance from the camera to the ceiling, and can further cause inaccurate estimation of the depth of ceiling points and ceiling-connected wall points.

To solve this problem, we adopt HorizonNet to predict wall-ceiling and wall-floor boundary pixel locations. As shown in the bottom branch of Fig. 7, taking RGB image as input, HorizonNet outputs wall-floor and wall-ceiling boundaries of the room. Then, based on the output of HorizonNet, we could further obtain the empty room's background mask. Based on the background mask, the empty room background depth D_H is obtained. D_S leaves empty/undefined pixels in the foreground depth area but has more errors on the ceiling and ceiling-connected wall area. D_H has a more accurate estimation of the ceiling and ceiling-connected wall, but has a much greater error in the area taken by furniture. To obtain the inferred background depth map D_{Bg}, we take the background depth value from D_H directly, but leave the foreground depth value as zero/empty. We explain how we make use of D_{Bg} with more detail in section 4.2, and visualize their error respectively in the supplementary material.

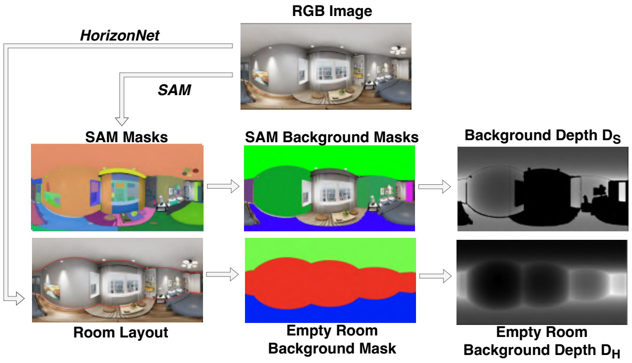

Figure 7. The upper branch shows the background depth D_S obtained from SAM outputs, while the bottom branch shows the empty room background depth D_H obtained through HorizonNet.

4.2. Depth Prediction Module

With the background depth map D_{Bg} and RGB image, a depth prediction module is proposed to predict the depth for the whole image. By incorporating the features of the background depth map, we aim to reduce the burden on the network and mitigate the overfitting problem when testing cases differ significantly from the training dataset.

As shown in Fig. 3, the RGB image and D_{Bg} are fed into two separate ResNet backbones, to obtain a series of feature maps F_i^{RGB} and F_i^{Bg}, where $i \in \{1, 2, 3, 4\}$. Then, F_i^{RGB} and F_i^{Bg} are fused by the Iterative Attentional Feature Fusion (IAFF) [5]. The structure of IAFF is shown in Fig 8. F_i^{RGB} and F_i^{Bg} are first fused by summation to obtain $F_1^{sum} \in \mathbb{R}^{B \times C \times H \times W}$, where B, C, H, W denote the batch size, channel number, feature map height and width, respectively. Then, F_1^{sum} is fed into two branches. In the left branch, it is processed by a series of convolutional layers and an adaptive average pooling and is squeezed into a feature vector $F_1^{glo} \in \mathbf{R}^{B \times C \times 1 \times 1}$ to represent a global feature. In the right branch, convolutional layers are applied to F_1^{sum}, to obtain the local feature $F_1^{loc} \in \mathbf{R}^{B \times C \times H \times W}$, which has the same size as F_1^{sum}. We sum up global feature F_1^{glo} and local feature F_1^{loc}, and obtain the attention weight map F_1^{att} with the sigmoid function. With the attention map, we obtain the first round fused feature F_2^{sum} by $F_2^{sum} = F_i^{RGB} \cdot F_1^{att} + F_i^{Bg} \cdot (1 - F_1^{att})$. Then F_2^{sum}

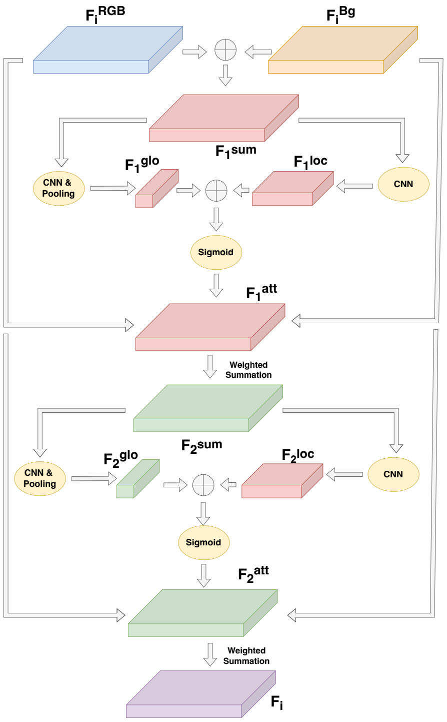

Figure 8. F_i^{RGB} and F_i^{Bg} are fused with attention mechanism to obtain the fused feature F_i.

is used as input to the second round of feature fusion with attention mechanism to obtain the final fused feature F_i.

According to the experiments in [19], latent horizontal features have shown success in several panoramic image-related tasks, considering both accuracy and speed. For this reason, once we have F_i and F_i^{Bg}, we compress them into latent horizontal features H_i^F and H_i^{Bg} to represent features in different receptive fields. Then, we concatenate all of them and feed them into a decoder to restore the depth map P_N. The decoder pipeline is the same as HoHoNet [19]. Once we have P_N, the L1 loss can be calculated for the network training.

4.3. Background Depth Replacement Module

As discussed in Sec. 3, depth estimation is a pixel-wise regression task. Yet, networks trained on a certain dataset may have poor performance when applied in real-world scenarios. To address this issue, we replace a part of depth values in P_N with the values obtained from initial background depth maps D_S and D_H, since they are more robust to variations in test data. Specifically, if the difference of a pixel's depth on D_S and D_H is less than a threshold α, we conclude that D_S and D_H are in agreement, and we average their depth values to obtain a confident depth map D_C. As shown in Fig. 3, the depth value in the grey area A_r of D_C is used directly in the final prediction. As for the black area A_n, where D_S and D_H are in disagreement, we use the network output P_N in the final prediction. In this paper, we set threshold α as 0.42.

5. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness and generalizability of our network. We have used the Replica and Structured3D datasets, and all models are evaluated with error measured in meters. *Please note that these datasets have been selected, since they (i) provide full access for both academic and non-academic use; and (ii) allow for evaluating the cross-domain performance, one being a real-world and the other being a synthetic dataset.*

- **Replica** dataset is captured using an RGB-D capture rig with an IR projector from real-world indoor scenes. We render 3554 images from these 3D models by setting the camera in random positions on the floor.
- **Structured3D** contains 196K rendered panoramic images and corresponding depth labels, which covers 12835 rooms from 3500 scenes. Each room is created manually using CAD models of furniture, which are in real-world dimensions and used in real production.

To evaluate robustness to the variations between training and testing sets and across different domains, we conducted experiments with three settings: (1) training on Replica and testing on Replica, (2) training on Structured3D and testing on Replica, and (3) training on Replica and testing on Structured3D. We did not perform an experiment with training and testing on Structured3D due to the high similarity between the training and testing splits. The similarly high performance of different baselines on this dataset does not indicate an overfitting issue, and does not allow evaluating their generalization ability.

5.1. Training and Testing on Replica

We perform a comparison with four SOTA baselines in terms of commonly used metrics for depth estimation. The results are summarized Tab. 2. As can be seen, HRDFuse [1] (CVPR 2023) and HoHoNet [19] (CVPR 2021) provide very similar performance. Our BGDNet significantly outperforms all the baselines in terms of all the metrics. Taking MAE as an example, our proposed method outperforms HRDFuse and HoHoNet by 21.62% and 27.59%, respectively. To qualitatively show the superiority of our method, we visualize the predicted depth map error of HRDFuse, HoHoNet and our BGDNet in Fig 9. A significant portion of the HoHoNet and HRDFuse depth error map is filled with red color, indicating that the error in these areas is high. Compared to these baselines, our method has better depth estimation in most areas, thanks to estimating the depth of the background, which is composed of planes and occupies a significant area, and using it for guidance, rather than relying on a network that overfits specific training datasets.

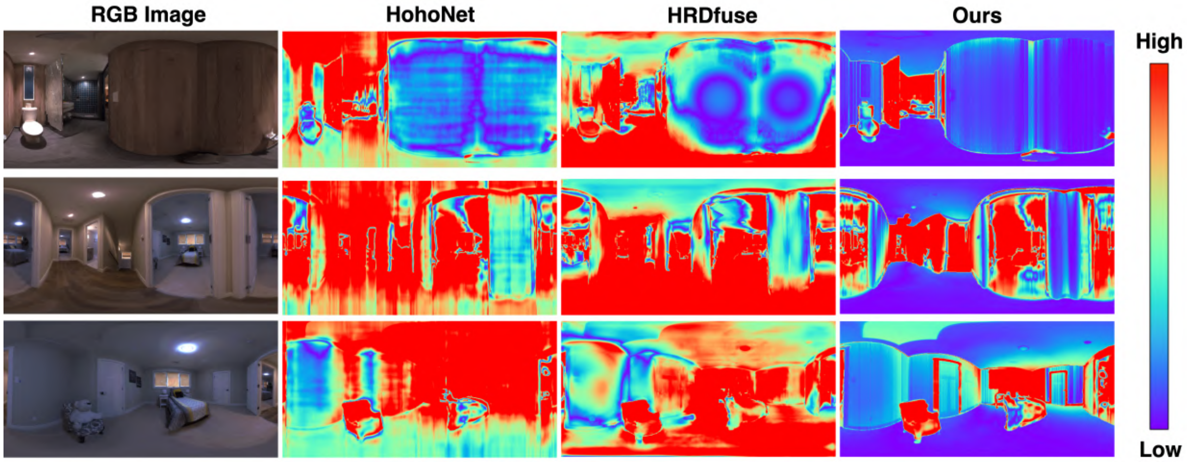

Figure 9. Qualitative results of predicted depth errors between HRDFuse, HoHoNet, and our method. Our approach shows generally smaller errors, particularly in the floor, ceiling, and wall regions, when compared to the benchmark methods.

Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
FCRN[11]	0.2871	0.4531	0.1511	0.6211	0.8891	0.9581
OmniFusion[13]	0.2721	0.5041	0.1571	0.6961	0.8911	0.9421
HoHoNet[19]	0.2141	0.3701	0.1441	0.8081	0.9261	0.9551
HRDFuse[1]	0.2171	0.3891	0.1421	0.7711	0.9221	0.9541
BGDNet (Ours)	**0.1678**	**0.3456**	**0.1334**	**0.8554**	**0.9365**	**0.9624**

Table 2. Experiment results when training and testing are both performed on the Replica dataset,

5.2. Training on Structured3D, Testing on Replica

In this experiment, we train the model on the Structured3D and test it on the Replica dataset. The Structured3D dataset consists of images rendered from manually created CAD models with consistent lighting and camera position (at the center of the room). The Replica dataset, on the other hand, contains images rendered with varying lighting and camera positions, using 3D meshes from real-world rooms. Thus, the difference between training and testing images is much larger than the one in Sec. 5.1, and the models are being tested on more challenging data than the training set. The results are shown in Tab. 3. Compared to Tab. 2, performances of all baselines degrade to various degrees in Tab. 3. Taking MAE as an example, the error of HRDFuse increases from 0.2171 to 0.2731, and HoHoNet experiences an error increase from 0.2141 to 0.3511. These baselines perform pixel-wise depth estimation solely based on RGB images. Consequently, due to the significant variations in image lighting, room layout, and camera position, these models have an unstable performance. In contrast, our method adopts a different approach by first estimating the depth of the background, which comprises a significant portion of images. The background depth estimation is performed using SAM and HorizonNet, which exhibit better stability and adaptability for various testing scenarios. Afterwards, the final prediction is made with the guidance of the estimated background depth. It can be seen that our method's MAE value in Tab. 3 is even less than the one in Tab. 2.

Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
FCRN[11]	0.3621	0.5471	0.2081	0.5411	0.7611	0.8621
OmniFusion[13]	0.2851	0.4421	0.1251	0.6971	0.9291	0.9661
HoHoNet[19]	0.3511	0.5231	0.2241	0.5691	0.7811	0.8701
HRDFuse[1]	0.2731	0.4101	0.1371	0.6971	0.9041	0.9501
BGDNet	**0.1632**	**0.3495**	**0.1062**	**0.8613**	**0.9434**	**0.9707**

Table 3. Experiment result when training on Strucutred3D dataset with testing on Replica dataset

5.3. Training on Replica, Testing on Structured3D

In this experiment, we conduct training on images rendered from the Replica dataset, and evaluate the models on the Structured3D dataset. While the images rendered from the Replica dataset exhibit variations in lighting and camera positions, the Structured3D dataset presents a much larger number of scenes and rooms compared to the Replica dataset. The results are shown in Tab. 4. Again, compared to Tab. 2, performances of all baselines degrade. The reason behind this is that the testing images, generated from a much larger variety of rooms compared to the training set, contain foreground objects and furniture layouts that have never appeared in the training set, posing a significant challenge for the deep learning-based prediction task. As a result, there is a significant drop in accuracy when it comes to predicting foreground objects. However, despite the considerable differences between the foregrounds of the training and testing sets, our proposed background depth estimation module can consistently and accurately predict the depth of the background, which includes the walls, ceiling, and floor. Thanks to this, our proposed method, which performs depth estimation based on background depth output, outperforms baseline models by significant margins.

Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
FCRN[11]	0.3431	0.5061	0.1501	0.6121	0.8421	0.9331
OmniFusion[13]	0.2981	0.4951	0.1411	0.6921	0.8831	0.9501
HoHoNet[19]	0.2721	0.4341	0.1271	0.6991	0.9011	0.9621
HRDFuse[1]	0.2451	0.4061	0.1201	0.7561	0.9161	0.9631
BGDNet	**0.1656**	**0.349**	**0.1001**	**0.8366**	**0.9377**	**0.9731**

Table 4. Experiment results when training on Replica and testing on Structured3D dataset.

6. Ablation Studies

We perform a series of ablation studies to further show the effectiveness of our proposed method. The training and testing are all performed on the Replica dataset.

6.1. Performance without Replacement Module

As explained in Sec. 4.3, we replace a part of the network prediction with depth from D_{Bg}. To demonstrate that our network provides better depth prediction than baselines, we evaluate the performance of BGDNet without depth replacement. The results in Tab. 5 show that the depth map directly output by BGDNet, without any replacement, still has better performance than the SOTA baselines. This shows that our proposed pipeline, which involves D_{Bg} as the input to the network, improves the network itself for the depth estimation task. With replacement component, the performance of our proposed method improves further.

Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
HoHoNet	0.2141	0.3701	0.1441	0.8081	0.9261	0.9551
HRDFuse	0.2171	0.3891	0.1421	0.7711	0.9221	0.9541
BGDNet w/o Repl.	0.2079	0.358	0.142	0.8161	0.9279	0.9562
BGDNet w/ Repl.	**0.1678**	**0.3456**	**0.1334**	**0.8554**	**0.9365**	**0.9624**

Table 5. The performance of our BGDNet with and without replacement module. **Bold** and underline show the best and second-best performances.

6.2. Effectiveness of Background Guidance

As described in Sec. 4.3, the depth value in area A_r of D_C, where D_S and D_H are in agreement, is used to replace the corresponding area in P_N and obtain final prediction. The depth value corresponding to other areas A_n, where D_S and D_H are in disagreement, the output of the network P_N is used for the final prediction. We investigate the performance of our method in these two areas A_r and A_n, and perform comparison with the SOTA baselines. The results are shown in Tab. 6. As seen in Tab. 6 (a), the MAE of our method on area A_r is half of HoHoNet and HRDFuse, which indicates the robustness of D_C to variations in test cases. As shown in Tab. 6 (b), with the guidance from D_{Bg} obtained from SAM and HorizonNet, our network also offers better performance on area A_n, compared to networks, which perform depth estimation by solely relying on RGB images.

Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
HoHoNet	0.1435	0.2024	0.0958	0.8667	0.9581	0.9741
HRDFuse	0.1471	0.2157	0.0986	0.8114	0.9541	0.9751
BGDNet	**0.0718**	**0.1455**	**0.0680**	**0.9388**	**0.9741**	**0.9861**

(a) Performance on A_r

Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
HoHoNet	0.3260	0.5317	0.1856	0.7175	0.8771	**0.9271**
HRDFuse	0.3291	0.5651	**0.1798**	0.7095	0.8751	0.9241
BGDNet	**0.3218**	**0.5231**	0.1859	**0.7235**	**0.8781**	0.9261

(b) Performance on A_n

Table 6. (a) and (b) show the performance of HoHoNet, HRDFuse and our method on area A_r and A_n respectively.

6.3. Effect of Different Training Data Size

In this section, we perform experiments on the Replica dataset by using 50% and 100% of the training data to observe the stability of models. We compare BGDNet with HoHoNet and HRDFuse, and show the results in Tab. 7. Our method outperforms the baselines in these two settings. Moreover, when 50% of training data are used, the MAE values of HoHoNet and HRDFuse drop by 6.54% and 6.46%, respectively. However, our proposed method has a variation of only 0.4% of the MAE value as the training dataset size changes.

Training Data Size	Model	MAE↓	RMSE↓	RMSE (log)↓	δ^1↑	δ^2↑	δ^3↑
50%	HoHoNet	0.2291	0.3761	0.1551	0.7961	0.9191	0.9501
	HRDFuse	0.2321	0.3881	0.1401	0.7701	0.9251	0.9581
	BGDNet	**0.1671**	**0.3441**	**0.1331**	**0.8581**	**0.9381**	**0.9621**
100%	HoHoNet	0.2141	0.3701	0.1441	0.8081	0.9261	0.9551
	HRDFuse	0.2171	0.3891	0.1421	0.7711	0.9221	0.9541
	BGDNet	**0.1678**	**0.3456**	**0.1334**	**0.8554**	**0.9365**	**0.9624**

Table 7. Performance of BGDNet, HoHoNet and HRDFuse with different training data sizes.

6.4. Other Experiments

We also performed experiments to show how the error from D_S obtained from SAM, and threshold value α affect the final prediction. These results and the discussion are presented in the supplementary material.

7. Conclusion

In this paper, we have first performed experiments to analyze the performance of the existing panoramic depth estimation models when indoor testing scenes greatly differ from the training data, and showed their performance significantly degrade indicating an overfitting problem. To address this problem, we have presented a new approach, BGDNet, which first estimates the room layout and the background depth, and then estimates the scene depth with guidance from background depth. Our proposed method BGDNet provides more robust and improved depth estimation, despite variances between training and testing cases. We have performed within dataset and cross-domain experiments on the Replica and Structured3D datasets, and shown that our method consistently outperforms SOTA baselines with significant margins in all experiments, and provides a better cross-domain performance.

References

[1] Hao Ai, Zidong Cao, Yan-Pei Cao, Ying Shan, and Lin Wang. Hrdfuse: Monocular 360deg depth estimation by collaboratively learning holistic-with-regional depth distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13273–13282, 2023. 1, 2, 3, 6, 7, 8

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1

[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. *International Conference on 3D Vision (3DV)*, 2017. 1

[4] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360º panoramas and 3d room layouts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2133–2143, 2021. 3

[5] Yimian Dai, Fabian Gieseke, Stefan Oehmcke, Yiquan Wu, and Kobus Barnard. Attentional feature fusion. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3560–3569, 2021. 5

[6] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 2

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 2

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

[9] Hyungjoo Jung, Youngjung Kim, Dongbo Min, Changjae Oh, and Kwanghoon Sohn. Depth prediction from a single image with conditional adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721. IEEE, 2017. 2

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 3

[11] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 2, 3, 7, 8

[12] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 3372–3380, 2017. 2

[13] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2801–2810, 2022. 1, 2, 3, 7, 8

[14] Kin Gwn Lore, Kishore Reddy, Michael Giering, and Edgar A Bernal. Generative adversarial networks for depth map estimation from rgb video. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1258–12588. IEEE, 2018. 2

[15] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21477–21487, 2023. 2

[16] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 2

[17] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2019. 1

[18] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019. 2, 4

[19] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2573–2582, 2021. 1, 2, 3, 6, 7, 8

[20] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–722, 2018. 1

[21] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020. 2

[22] Qiang Zhao, Chen Zhu, Feng Dai, Yike Ma, Guoqing Jin, and Yongdong Zhang. Distortion-aware cnns for spherical images. In *IJCAI*, pages 1198–1204, 2018. 1

[23] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. 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*, pages 519–535. Springer, 2020. 1

[24] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for

indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018. 2