LED^2-Net: Monocular 360° Layout Estimation via Differentiable Depth Rendering

Fu-En Wang*[1]
fulton84717@gapp.nthu.edu.tw

Yu-Hsuan Yeh*[2]
yuhsuan.eic08g@nctu.edu.tw

Min Sun[1,4]
sunmin@ee.nthu.edu.tw

Wei-Chen Chiu[2]
walon@cs.nctu.edu.tw

Yi-Hsuan Tsai[3]
ytsai@nec-labs.com

Abstract

Although significant progress has been made in room layout estimation, most methods aim to reduce the loss in the 2D pixel coordinate rather than exploiting the room structure in the 3D space. Towards reconstructing the room layout in 3D, we formulate the task of 360° layout estimation as a problem of predicting depth on the horizon line of a panorama. Specifically, we propose the Differentiable Depth Rendering procedure to make the conversion from layout to depth prediction differentiable, thus making our proposed model end-to-end trainable while leveraging the 3D geometric information, without the need of providing the ground truth depth. Our method achieves state-of-the-art performance on numerous 360° layout benchmark datasets. Moreover, our formulation enables a pre-training step on the depth dataset, which further improves the generalizability of our layout estimation model.

1. Introduction

Inferring the geometric structure such as depth, layout, etc. from a single image has been studied for years. With the advance of deep learning, convolutional neural networks are widely used in these tasks. In addition, with the increasing popularity of consumer-level 360° cameras, approaches dealing with 360° panoramas start to play a crucial role in virtual and augmented reality (VR/AR) and robotic vision. In order to support the indoor use case of these applications, the task of room layout estimation from a single 360° panorama becomes important.

Generally, the room layout can be constructed by connecting the adjacent room corners or directly finding the boundary between walls, floor, and ceiling. Hence, most methods directly estimate the layout boundary and corners from the input panorama, e.g., HorizonNet [18]. Despite significant progress being achieved, the 3D reconstruction of the room layout is often not as good as expected from observing the results overlaid on the 2D panorama. The main issue is that these methods are trained with the loss in the pixel coordinates of the 2D panorama rather than in the coordinate of the 3D reconstruction. In particular, 2D pixel loss disregards the fact that pixels with different depths from the camera should contribute differently to the loss in the 3D coordinate (see Figure 2). Additional losses such as binary segmentation loss in the ceiling and floor perspective views have been introduced [24]. However, segmentation loss tends to focus on the correctness of the majority of the segment rather than the boundary of the segment. On the other hand, although several progresses have been made for monocular 360° depth estimation given a single 2D panorama [11, 25, 20, 28] where the loss is defined to reduce errors in 3D, none of the existing works aims at applying depth-based constraints to layout estimation frame-

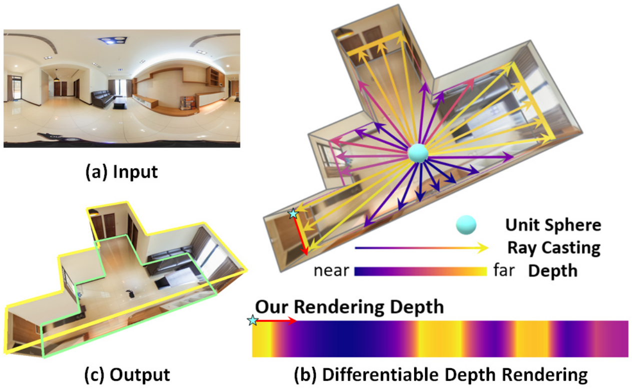

Figure 1. Our LED^2-Net takes the (a) single panorama as input and infers the (c) 3D room layout. We propose the (b) Differentiable Depth Rendering technique to incorporate the geometry-aware information into our model.

[1] National Tsing Hua University
[2] National Chiao Tung University
[3] NEC Labs America
[4] MOST Joint Research Center for AI Technology and All Vista Healthcare
* The authors contribute equally to this paper.

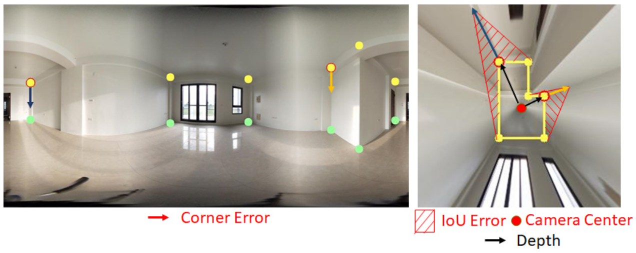

Figure 2. For the panorama shown on the left, we visualize several corners/boundary points where the layout estimation methods generally aim to find, in which their objective is mostly based on the errors in the 2D pixel coordinate on the equirectangular image (e.g., two arrows indicate the same error). However, as illustrated in the ceiling perspective view on the right, these two corner errors actually associate with different depth values (denoted as black arrows) from the camera, and such depth difference cannot be reflected in the intersection-over-union (IoU) error metric.

work. Hence, we are inspired to leverage the depth prediction loss to improve room layout estimation, which provides us the geometric information in the 3D space.

To this end, we re-formulate the 360° layout estimation into a unique 360° depth estimation problem. First, instead of trying to estimate the full depth map of the panorama, we only estimate the depth values on the horizon line of a panorama, which we call "horizon-depth" (see Figure 3), which is already sufficient to recover the layout. To this end, we propose a differentiable *L2D* (Layout-to-Depth) procedure to transform the layout into a "layout-depth". As a result, we can adopt the widely-used objective functions in depth estimation to train our model on layout estimation datasets, which also enables the possibility to pre-train our model on depth estimation datasets and further improve the model generalizability.

Our proposed layout-to-depth procedure is based on ray-casting (i.e., casting the rays from a unit sphere as illustrated in Figure 1(b)). The depth is recovered by computing the distances of each ray. Ideally, we can predict the depth for every pixel on the horizon line, but it would reduce the model efficiency. Also, for layout estimation, we simply need to know at least the depth values of corner points in the room. To consider the balance between efficiency and accuracy, we propose a "Grid Re-sample" strategy which is able to approximate the horizon-depth map by a flexible number of casting rays (see Figure 1(b)). We name our method **LED^2-Net**, which can be efficiently trained in an end-to-end fashion.

To demonstrate the effectiveness of our proposed model based on the novel technique of Differentiable Depth Rendering for 360° layout estimation, we conduct extensive experiments on four benchmark datasets, including Matterport3D [30], Realtor360 [24], PanoContext [26], and Stanford2D3D [1]. We show that our method performs favorably against state-of-the-art approaches in both the within-dataset and cross-dataset settings. More interestingly, we leverage the property of our depth estimation objective to enable depth pre-training using a synthetic dataset, Structure3D [27], which further improves the generalization ability of our model. Our supplementary material, source code, and pre-trained models are available to the public on our project website[1]. We summarize our contributions as follows:

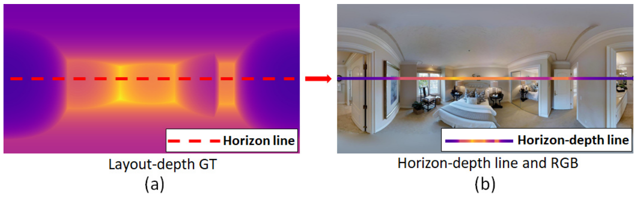

Figure 3. (a) The layout-depth generated from layout annotation. The horizontal red line indicates the horizon-depth, in which we use it as the supervisory signal for the network. (b) The horizon-depth aligned with the RGB panorama.

1. We reformulate the task of 360° layout estimation to a unique 360° depth estimation problem that optimizes a loss in 3D while maintaining the simplicity of layout estimation.

2. We propose a differentiable layout-to-depth procedure to convert the layout into horizon-depth through ray-casting of a few points, which enables the end-to-end training on layout estimation datasets.

3. We show that our framework can be seamlessly pre-trained by 360° depth datasets, which further improves the generalizability on cross-dataset evaluations.

2. Related Work

With the popularity of virtual/augmented reality, inferring the geometric context from 360° images becomes an important topic in recent years. In this section, we discuss the literature relevant to 360° depth and layout estimation.

360° Depth and Layout Estimation. One of the pioneering works for 360° perception is proposed by Cheng *et al.* [3]. They use cubemap projection and cube padding to avoid the distortion of equirectangular images while keeping the connection between each adjacent face of the cubemap. Wang *et al.* [19] then adopt the cubemap representation and unsupervisedly learn monocular 360° depth estimation. To capture distortion-aware context, several approaches of spherical CNNs are proposed [17, 16, 4, 23, 5, 7]. With a supervised scheme, Zioulis *et al.* [28] incorporate [17] and propose two network variants to estimate monocular 360° depth. Following [19, 28], Wang *et al.* [20] pro-

[1] https://fuenwang.ml/project/led2net

pose a framework consisting of equirectangular and cubemap projections to estimate the 360° depth map.

While the depth map contains details of a scene, the layout provides a rough room structure. Since most rooms have walls that are perpendicular to each other, many approaches follow the assumption of Manhattan World [6]. By using the Line Segment Detection [8] and extracting the vanishing points, Zhang *et al*. [26] generate the layout hypothesis and infers the layout from a single 360° panorama. Recently, CNN-based approaches have come in handy for layout estimation. Zou *et al*. [29] first incorporate a U-Net-like [14] architecture to predict the corners/boundary of the room, and then apply the optimization based on the Manhattan assumption. Sun *et al*. [18] propose HorizonNet which simplifies the layout into a horizontal representation and improves it via a recurrent neural network. On the other hand, as the bird's eye view of the layout can be considered as the floor plan, several approaches consider this property and instead infer the floor plan of a room. Yang *et al*. [24] propose DuLa-Net to predict a binary segmentation map as the floor plan. Pintore *et al*. [13] predict the floor plan by combining the benefits of both DuLa-Net and HorizonNet.

Different from the above-mentioned approaches, we find that the geometric cues across the layout and depth are tightly relevant to each other, and combining the two information becomes an important topic recently. Jin *et al*. [11] propose to use the layout information (i.e. boundary, corners, and depth) to improve 360° depth estimation. Zeng *et al*. [25] propose a two-stage framework to estimate both depth and layout-depth. However, since the annotation of layout involves only monocular images, the room scale is unknown. Thus, direct regression for up-to-scale layout-depth suffers from unknown scale issues. This motivates us to design a representation which is differentiable, geometric-aware, scale-invariant, and efficient to optimize, in a way that approximates the dense depth map.

3. Approach

As motivated previously, in this paper we aim to use the layout-depth as the training objective for 360° layout estimation, which is realized by our "L2D" (Layout-to-Depth) transformation. Basically, our layout estimation network takes a panorama as input and learns to predict the spherical coordinates of boundary points on the equirectangular image. Note that boundary points outlining both the floor and ceiling are estimated, as shown in Figure 4(b). Afterward, the L2D transformation is applied to the predicted boundary points to establish the horizon-depth map, as shown in Figure 4(c-g). The errors on such a horizon-depth map with respect to the ground truth become the objective for training our layout estimation network. In the following, we first introduce the layout representation and spherical projection in Section 3.1, followed by elaborating the L2D transformation in Section 3.2. Finally, we provide details of our loss function and network architecture in Section 3.3 and Section 3.4, respectively.

3.1. Preliminary

Given an equirectangular image taken in a room, we represent its layout with a sequence of boundary positions in the spherical coordinate system, as shown in Figure 4(b). Basically, a pixel q on the equirectangular image positioned by longitude and latitude, i.e., (θ, ϕ), can be easily converted to a 3D point $p \in \mathbb{R}^3$ on a unit sphere by the function $S(\theta,\ \phi)$, as shown in Figure 4(c):

$$\begin{aligned}
S(\theta,\ \phi) &= (p_x,\ p_y,\ p_z)\ , \\
p_x &= \cos(\phi) \cdot \sin(\theta)\ , \\
p_y &= \sin(\phi)\ , \\
p_z &= \cos(\phi) \cdot \cos(\theta)\ .
\end{aligned} \tag{1}$$

As θ and ϕ indicate a location on the sphere, the range of θ is from $-\pi$ to $+\pi$, while the range of ϕ is from -0.5π to $+0.5\pi$. In the following, we use such function S for converting the layout boundary to 3D points on a unit sphere.

3.2. L2D Transformation

Before we delve into the details of our proposed L2D transformation, we first introduce two important assumptions used in most prior works of 360° layout estimation, where these two assumptions help to tackle the absolute scale issue that typically cannot be derived from a monocular image:

1. The height of the camera for taking panoramas (i.e., the perpendicular distance from the floor) is normalized to a fixed number [24, 18]. Following [18], it is set to 1.6 for all the experiments in this paper.

2. The ceilings, floors, and walls are flat planes, where the walls are perpendicular to each other (i.e., Manhattan World assumption [6]).

Our proposed method also follows these two assumptions for transforming the layout into depth prediction. In addition, the two assumptions are applied to generate the ground truth horizon-depth. In the following, we introduce three main steps of our L2D transformation.

Layout Plane Recovering. Our layout representation predicted from the input panorama is composed of two sets of spherical coordinates, denoted respectively as $\mathbb{Q}^f$ and $\mathbb{Q}^c$, where $\mathbb{Q}^f = \{q_i^f\}_{i=1}^N,\ q_i^f = (\theta_i^f, \phi_i^f)$ and $\mathbb{Q}^c = \{q_i^c\}_{i=1}^N,\ q_i^c = (\theta_i^c, \phi_i^c)$. The set $\mathbb{Q}^f$ (respectively $\mathbb{Q}^c$) represents the boundary points sampled from the boundaries between the walls and the floor (respectively the ceiling), where N is the number of boundary points and the smallest N is equal to the number of walls. Particularly, q_i^f and

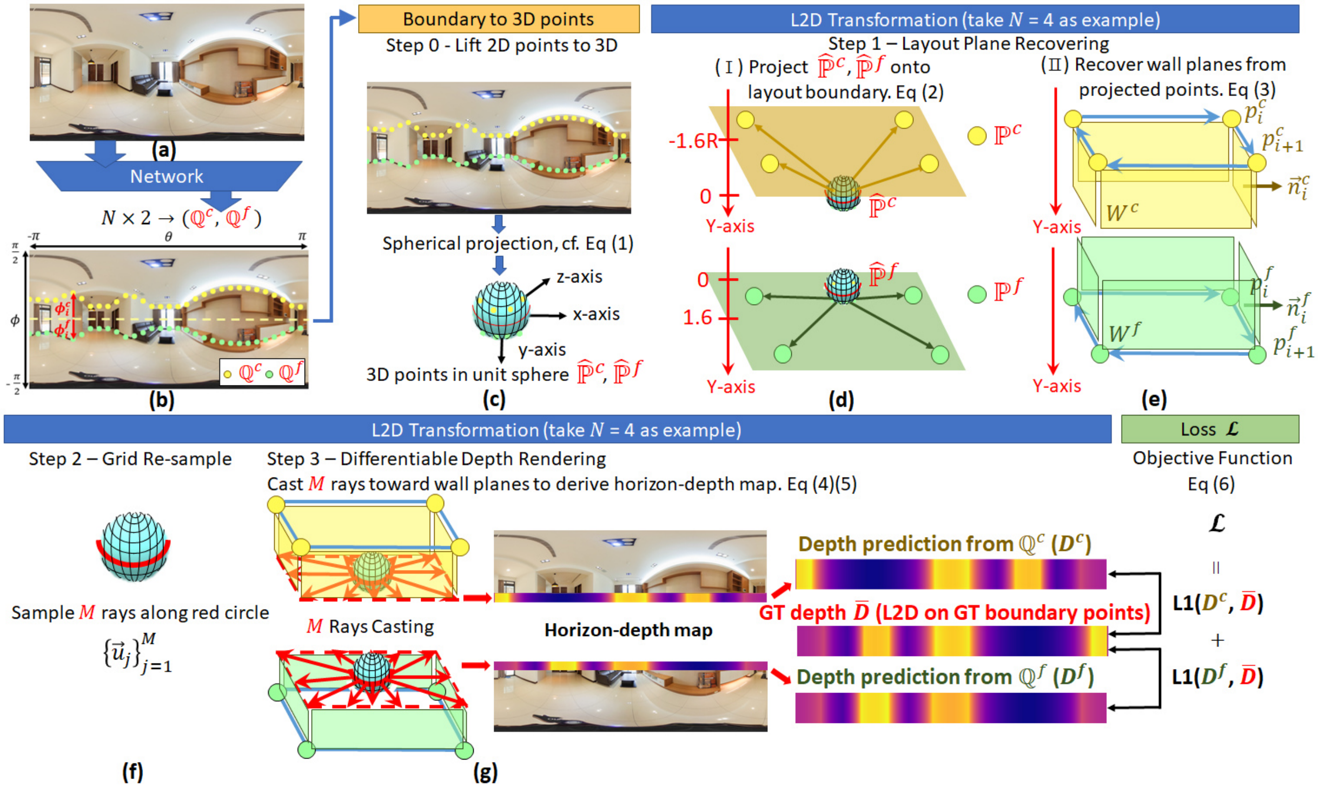
(a)
Network
$N \times 2 \rightarrow (\mathbb{Q}^c, \mathbb{Q}^f)$
$-\pi$
θ
π
$\frac{\pi}{2}$
ϕ
$-\frac{\pi}{2}$
ϕ_i^c
ϕ_i^f
$\mathbb{Q}^c$ $\mathbb{Q}^f$
(b)
Boundary to 3D points
Step 0 - Lift 2D points to 3D
Spherical projection, cf. Eq (1)
z-axis
x-axis
y-axis
3D points in unit sphere $\hat{\mathbb{P}}^c, \hat{\mathbb{P}}^f$
(c)
L2D Transformation (take N = 4 as example)
Step 1 – Layout Plane Recovering
(Ⅰ) Project $\hat{\mathbb{P}}^c, \hat{\mathbb{P}}^f$ onto layout boundary. Eq (2)
-1.6R
0
Y-axis
$\hat{\mathbb{P}}^c$
$\mathbb{P}^c$
0
1.6
Y-axis
$\hat{\mathbb{P}}^f$
$\mathbb{P}^f$
(d)
(Ⅱ) Recover wall planes from projected points. Eq (3)
p_i^c
p_{i+1}^c
$\vec{n}_i^c$
W^c
Y-axis
p_i^f
$\vec{n}_i^f$
W^f
p_{i+1}^f
Y-axis
(e)
L2D Transformation (take N = 4 as example)
Step 2 – Grid Re-sample
Sample M rays along red circle
$\{\vec{u}_j\}_{j=1}^M$
(f)
Step 3 – Differentiable Depth Rendering
Cast M rays toward wall planes to derive horizon-depth map. Eq (4)(5)
M Rays Casting
Horizon-depth map
(g)
Depth prediction from $\mathbb{Q}^c$ (D^c)
GT depth $\bar{D}$ (L2D on GT boundary points)
Depth prediction from $\mathbb{Q}^f$ (D^f)
Loss $\mathcal{L}$
Objective Function
Eq (6)
$\mathcal{L}$
=
L1(D^c, $\bar{D}$)
+
L1(D^f, $\bar{D}$)

Figure 4. The overall framework of our proposed LED^2-Net. Our layout estimation network takes **(a)** an RGB panorama as input, and outputs **(b)** the layout representation which is composed of two sets of layout boundary points ($\mathbb{Q}^c, \mathbb{Q}^f$) on the ceiling and the floor respectively. Our novel L2D (layout-to-depth) transformation is applied to transform the layout representation, which is first **(c)** lifted to 3D space (cf. Section 3.1), into the horizon-depth maps (i.e. D^c and D^f). The L2D transformation is composed of three main steps: **(d)(e)**"Layout Plane Recovering" (cf. Section 3.2) to recover the plane equation (i.e. W^c and W^f), **(f)** "Grid Re-sample" (cf. Section 3.2), and **(g)**"Differentiable Depth Rendering" (cf. Section 3.2). The training objective of our LED^2-Net is thus defined by the errors between our estimated horizon-depth maps and the corresponding ground truth $\bar{D}$ (cf. Section 3.3).

q_j^c with $i = j$ share the same value of longitude θ. Note that, the points in $\mathbb{Q}^f$ or $\mathbb{Q}^c$ are already arranged by the ascending order according to their values of longitude θ. We can convert these two point sets from spherical coordinate system onto a unit sphere by using function S via (1), and obtain $\hat{\mathbb{P}}^f = \{\hat{p}_i^f = S(q_i^f)\}_{i=1}^N$ and $\hat{\mathbb{P}}^c = \{\hat{p}_i^c = S(q_i^c)\}_{i=1}^N$, where each $\hat{p}$ is a 3-dimensional vector in the Cartesian coordinate system with $\|\hat{p}\| = 1$. This step is illustrated in Figure 4(c).

Next, based on the two aforementioned assumptions of Manhattan layout, i.e., the camera height is fixed to 1.6 and the floor (respectively the ceiling) is a flat plane, we project all the points in $\hat{\mathbb{P}}^f$ (respectively $\hat{\mathbb{P}}^c$) from the unit sphere onto the boundary between the walls and the floor (respectively the ceiling):

$$
\begin{aligned}
p_i^f &= \hat{p}_i^f * \frac{1.6}{\hat{p}_i^f(y)}, \\
p_i^c &= \hat{p}_i^c * \frac{-1.6R}{\hat{p}_i^c(y)},
\end{aligned}
\tag{2}
$$

where $\hat{p}_i^f(y)$ denotes the coordinate of $\hat{p}_i^f$ in the $y-$axis, and similarly for $\hat{p}_i^c(y)$. R denotes the ratio between the height of the camera and the distance from the camera center to the ceiling. Figure 4(d) presents the step of this projection. Note that the $y-$axis in this Cartesian coordinate system is perpendicular to the ground plane and point to the floor.

After obtaining $\mathbb{P}^f = \{p_i^f\}_{i=1}^N$ and $\mathbb{P}^c = \{p_i^c\}_{i=1}^N$, based on every pair of adjacent points in $\mathbb{P}^f$, we derive N walls where their plane equations $\{W_i^f\}_{i=1}^N$ are obtained by:

$$
\begin{aligned}
\vec{n}_i^f &= \hat{y} \times (p_{i+1}^f - p_i^f) \,, \\
t_i^f &= -\vec{n}_i^f \cdot p_i^f \,, \\
W_i^f &= (\vec{n}_i^f, t_i^f) \,.
\end{aligned}
\tag{3}
$$

where $\times$ denotes the outer product operation, $\hat{y}$ denotes the unit vector along $y-$axis, $\vec{n}$ is the 3-dimensional normal vector of the wall, and t_i^f is the offset in the plane equation. This plane recovering step is shown in Figure 4(e). Another set of walls based on $\mathbb{P}^c$ with plane equations $\{W_i^c\}_{i=1}^N$ can be also derived in the same way.

Grid Re-sample. After having the wall equations in the "layout plane recovering" step, we aim at casting the rays from the unit sphere (i.e., camera center) towards these walls in order to obtain the information related to depth. As motivated in the introduction, we take the computational efficiency into consideration, in which the number of casting rays is less than the size of the complete horizon-depth map (i.e., covering the width of the input panorama). In other words, the horizon-depth map generated by our ray-casting process is an approximation of the complete one.

Here, we approximate the horizon-depth map with a size M by sampling M rays casting from the unit sphere (see Figure 4(f)), in which these rays are denoted as M unit vectors $\{\vec{u}_j\}_{j=1}^{M}$, $\|\vec{u}_j\| = 1$. Specifically, these rays are obtained by $\vec{u}_j = S(\theta_j, \phi_j),\ j = \{1, ..., M\}$, where $\phi_j = 0\ \forall j$ and $\{\theta_j\}_{j=1}^{M}$ are equiangularly sampled from $[-\pi, \pi]$. That is, these rays form a 360° horizontal radiation pattern. Note that, the zero latitude is aligned with the height of the camera, which is a general setting in the prior work [18, 24].

Differentiable Depth Rendering. Now, as we already have the wall planes and the casting rays, we can compute the intersections of them, and then the horizon-depth map can be easily obtained from the distances between these intersections and the camera center. To be detailed, given wall planes $\{W_i^f\}_{i=1}^{N}$ and casting rays $\{\vec{u}_j\}_{j=1}^{M}$, for each $\vec{u}_j$, we obtain N candidate depth values $\{d_{j,i}^f\}_{i=1}^{N}$ as:

$$d_{j,i}^f = -\frac{t_i^f}{\vec{u}_j \cdot \vec{n}_i^f}. \tag{4}$$

In particular, we use two conditions to filter out inappropriate candidates: (1) $d_{j,i}^f$ must be ≥ 0; and (2) since the wall W_i^f is derived from p_{i+1}^f and p_i^f via (3), which are connected to the longitude θ_{i+1}^f and θ_i^f, the longitude of the intersection of $\vec{u}_j$ and W_i^f must be within the range $[\theta_i^f, \theta_{i+1}^f]$. After filtering out the candidates, we obtain the corresponding depth value d_j^f of the $j-$th pixel on the resultant horizon-depth map of size M via:

$$d_j^f = \min_i d_{j,i}^f, \tag{5}$$

where we use the $\min$ function to find d_j^f, since other larger values indicate the occluded area. The same computation procedure can be applied to another set of wall planes $\{W_i^c\}_{i=1}^{N}$. We denote the final horizon-depth maps by concatenating d_j^f (respectively d_j^c) as D^f (respectively D^c) in Figure 4(g).

3.3. Objective Function

As the entire procedure of our proposed L2D (layout-to-depth) transformation is differentiable, the training objective of our model can be directly defined upon the errors of our two predicted horizon-depth maps (i.e., D^f and D^c) with respect to the ground truth horizon-depth map $\bar{D}$. Moreover, the model is end-to-end trainable, where we adopt the L1 loss to measure the errors between depth maps:

$$\mathcal{L} = \|D^f - \bar{D}\|_1 + \|D^c - \bar{D}\|_1. \tag{6}$$

It is worth noting that, the ground truth depth map $\bar{D}$ can be obtained via applying the same L2D transformation on the ground truth layout, which consists of the ground truth layout boundary points. Also, as the depth maps obtained from the ground truth boundary points either on the floor or the ceiling should be identical to each other, here we only use a single ground truth horizon-depth map $\bar{D}$. However, using the depth objective may encounter the scaling issue caused by the unknown scale in a room. In our method, since the point sets $\mathbb{Q}^f$ and $\mathbb{Q}^c$ only represent the angles, which are scale-invariant, this effect does not exist anymore.

Another benefit of our learning objective based on a depth loss is that we can not only use the horizon-depth map derived from the layout, but also the ones acquired from the laser scanner or virtual environments. Later in our experimental section, we demonstrate that using a virtual 360° dataset with depth ground truths to pre-train our layout estimation network can further improve its generalization ability on the cross-dataset evaluation setting.

3.4. Network Architecture

We follow the architecture of the HorizonNet [18] to construct our layout estimation network. First, a ResNet-50 [9] based encoder is adopted to extract feature maps of the input equirectangular image, where the feature maps at different scales are further fused together via several convolution layers followed by concatenation [12]. Then, a bidirectional Long Short-Term Memory (bi-LSTM) module [15, 10] is applied to smooth the fused feature map along the width, followed by one fully-connected layer and a sigmoid function to obtain our final output. In our formulation, the layout representation estimated by our network is composed of two point sets in the spherical coordinate, i.e., $\mathbb{Q}^f$ and $\mathbb{Q}^c$. In our implementation, since we distribute the N boundary points along the axis of longitude sampled from $[-\pi, \pi]$, our layout estimation network only needs to predict the latitude values of the boundary points. Thus, the second dimension of the output size $N \times 2$ indicates the two point sets related to the ceiling and the floor (N is set to 256 in our experiments). To further constrain the layout prediction by the Manhattan assumption and infer the layout height to create a clean room layout, we adopt the post processing procedure of HorizonNet [18]. Please refer to the supplementary material for more detailed descriptions of our network architecture and the post processing step.

Table 1. The quantitative experimental results on Realtor360 [24] dataset.

Method	Overall		4 corners		6 corenrs		8 corners		10+ corners	
	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)
LayoutNet	65.84	62.77	80.41	76.60	60.50	57.87	41.16	41.16	22.35	22.35
DuLa-Net	80.53	77.20	82.63	78.91	80.72	77.79	78.12	74.86	63.10	59.72
HorizonNet	86.69	83.66	87.83	84.73	87.63	84.78	81.27	78.44	78.49	73.64
AtlantaNet	80.36	74.59	83.42	77.05	80.67	75.01	73.72	69.31	59.43	55.51
Ours	**88.19**	**85.21**	**89.25**	**86.33**	**88.80**	**85.97**	**83.70**	**80.81**	**81.67**	**76.20**

Table 2. The quantitative experimental results on Mattertport3D [30] dataset.

Method	Overall		4 corners		6 corenrs		8 corners		10+ corners	
	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)	2D IoU (%)	3D IoU (%)
LayoutNet	78.73	75.82	84.61	81.35	75.02	72.33	69.79	67.45	65.14	63.00
DuLa-Net	78.82	75.05	81.12	77.02	82.69	78.79	74.00	71.03	66.12	63.27
HorizonNet	81.24	78.73	83.54	80.81	82.91	80.61	76.26	74.10	72.47	70.30
AtlantaNet	82.09	80.02	84.42	82.09	83.85	82.08	76.97	75.19	**73.19**	**71.62**
Ours	**83.91**	**81.52**	**86.91**	**84.22**	**85.53**	**83.22**	**78.72**	**76.89**	71.79	70.09

Table 3. The qualitative experimental results on both PanoContext [26] and Stanford2D3D [1] datasets.

Method	3D IoU (%)				
	LayoutNet	DuLa-Net	HorizonNet	AtlantaNet	**Ours**
PanoContext	74.48	77.42	82.17	78.76	**82.75**
Stanford2D3D	76.33	79.36	79.79	82.43	**83.77**

4. Experiments

We conduct extensive experiments on four 360° layout datasets, which are Realtor360 [24] and Matterport3D [30] with more complicated scenes, and two cuboid datasets, PanoContext [26] and Stanford2D3D [1]. We compare our proposed method with several state-of-the-art baselines of monocular 360° layout estimation, including LayoutNet [29], DuLa-Net [24], HorizonNet [18], and AtlantaNet [13]. Moreover, since our method allows us to use the datasets that contain the depth ground truth as a pre-training step, we conduct additional experiments by leveraging a synthetic dataset (i.e., Structure3D [27]), in which the depth annotation is free to collect. Note that such pre-training is an additional benefit but not the requirement of our framework, in which none of the existing layout estimation methods is equipped with this ability. We follow the same protocols of [24] to calculate the 2D and 3D intersection-over-union (IoU). We also investigate the model sensitivity on the ray-casting number (i.e., M). More results are provided in the supplementary material.

PanoContext and Stanford2D3D. There are around 500 panoramas along with ground truth layout annotations in PanoContext dataset, which are collected from SUN360 [22] and labeled by Zhang *et al.* [26]. In order to extend the available training samples of layout estimation, Zou *et al.* [29] additionally collect 571 panoramas from the original Stanford2D3D dataset [1] and label the corresponding layout ground truths. We adopt the same train/val/test splits as used in [29] for all the experiments on these two datasets. Please note that, as PanoContext and Stanford2D3D datasets primarily consist of cuboid-shape layouts, only adopting these two datasets is not enough for well evaluating the capacity of different models for tackling layout estimation on full 360° panoramas. We, therefore, consider other datasets such as Realtor360 and Matterport3D, which contain more complicated cases of layouts.

Realtor360. This dataset is proposed and annotated by Yang *et al.* [24], where they collect 593 panoramas from the subsets of SUN360 dataset (composed of scenes of living rooms and bedrooms) as well as 1980 panoramas from a real estate database. We follow the official train/test split as [24] to conduct experiments on Realtor360.

Matterport3D. The original Matterport3D [2] dataset contains 10,800 panoramas along with the depth ground truths obtained from laser scanners. Zou *et al.* [30] and Wang *et al.* [21] remove the cases that do not satisfy the Manhattan World assumption and use the annotation tool provided by DuLa-Net [24] to label the layout ground truth. Eventually, there are 2295 panoramas in total, including the complicated cases with the different number of layout corners. We adopt the official train/val/test split of [30] to conduct the experiments.

4.1. Experimental Results

Datasets with Challenging Cases. We first conduct experiments on the datasets which contain sufficiently complicated layouts (i.e., Realtor360 and Matterport3D) for making comparisons among different models in terms of their ability to deal with difficult cases. Table 1 and Table 2 provide the quantitative results on the Realtor360 and Matterport3D datasets, respectively. On these two datasets, the proposed method performs favorably against other state-of-the-art methods. In particular, compared with HorizonNet [18] that has quite a similar architecture to our layout estimation network, our model consistently produces better performance and thus verifies the contribution of our novel L2D (layout-to-depth) transformation building upon

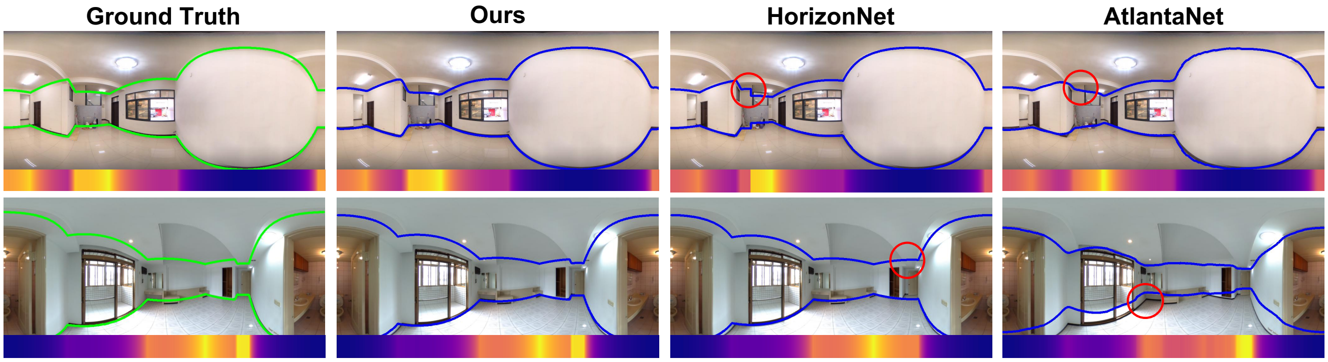

Figure 5. The qualitative results of layout boundary and horizon-depth on the Realtor360 [24] dataset. The red circles highlight the errors produced by the baselines.

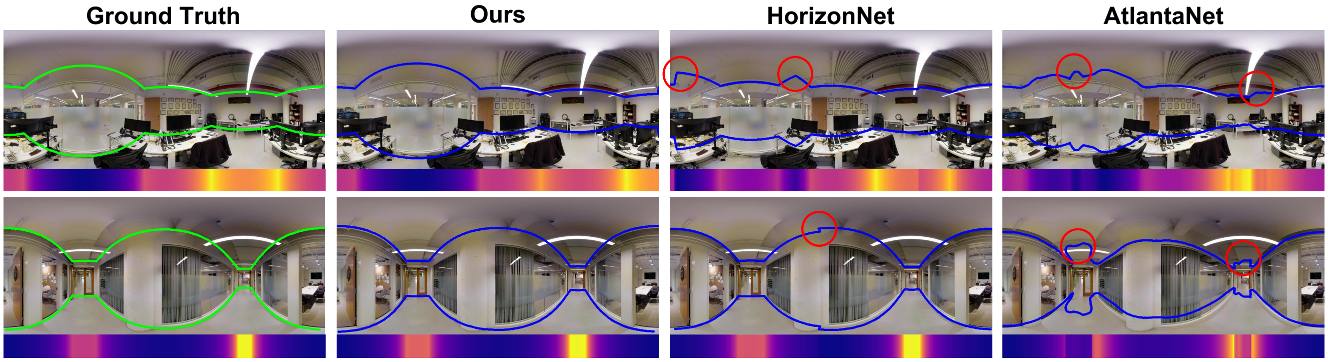

Figure 6. The qualitative results of layout boundary and horizon-depth on the Matterport3D [30] dataset. The red circles highlight the errors produced by the baselines.

the "Differentiable Depth Rendering". To further visualize the effectiveness of our approach, we provide several qualitative example results in Figure 5 and Figure 6, showing that our proposed method is able to infer precise layouts for complicated cases (i.e., rooms with many layout corners), while the other approaches instead produce noisy layout estimations. In addition, we provide more 3D layout visualizations in Figure 7, in order to demonstrate our capability on the task of monocular 360° layout estimation.

Datasets with Cuboid-shape Layouts. In addition to more complicated layouts, for the experiments conducted on the PanoContext and Stanford2D3D datasets, which are primarily composed of cuboid-shape layouts (i.e., the rooms have four layout corners) and considered to be simpler cases, we provide the quantitative results in Table 3. We show that our proposed method consistently outperforms all the state-of-the-art methods.

4.2. Generalizability

The generalizability of room layout estimation (e.g., cross-dataset setting) has not been widely studied, yet it is an important task to validate whether the models can generalize to unseen room layouts with different dataset distribution. To investigate this problem, we first perform cross-dataset evaluations, as shown in the top three rows of Table 4. Here, we provide two settings: 1) train the model on Matterport3D and test on Realtor360 and Stanford2D3D (the left part in Table 4), and 2) train the model on Realtor360 and test on Matterport3D and Stanford2D3D (the right part in Table 4). Results show that our model consistently performs better than the other approaches, HorizonNet and AtlantaNet, which validate that our method is more robust to the cross-dataset setting.

Moreover, we aim to demonstrate that using the 360° depth datasets for pre-training is able to improve the generalizability of our proposed network. However, obtaining depth ground truth from laser scanners is much more expensive than labeling the layout ground truth, and hence we focus on adopting the synthetic Structure3D [27] dataset to perform our model pre-training, in which this dataset is collected from a virtual environment and the ground truths are in high-quality and are easy to obtain. In total, Structure3D contains 21,835 room scenes and 196,515 photo-realistic panoramas along with the corresponding ground truth depth maps, where we can extract the horizon-depth maps to pretrain our model.

In the last row of Table 4, we show the results of finetuning on the training dataset and testing on the cross-dataset setting with such a pre-training scheme. We find that this strategy significantly improves some of the settings, e.g.,

Table 4. The quantitative results of the cross-dataset evaluation scheme (cf. Section 4.2).

Method	IoU (%)	Train-Dataset	Cross-Dataset		Train-Dataset	Cross-Dataset	
		Matterport3D	Realtor360	Stanford2D3D	Realtor360	Matterport3D	Stanford2D3D
HorizonNet	2D 3D	81.24 78.73	80.01 76.37	84.91 81.74	86.69 83.66	78.00 75.24	84.84 80.92
AtlantaNet	2D 3D	73.11 68.09	72.26 66.88	81.97 75.22	80.36 74.59	73.21 67.76	83.48 77.38
Ours [w/o pretrained]	2D 3D	**83.91** **81.52**	80.74 77.17	85.25 80.54	88.19 85.21	79.78 76.89	86.51 83.50
Ours [w/ pretrained]	2D 3D	83.59 81.24	**83.73** **80.52**	**88.37** **85.20**	**89.00** **86.31**	**79.92** **76.99**	**86.81** **83.69**

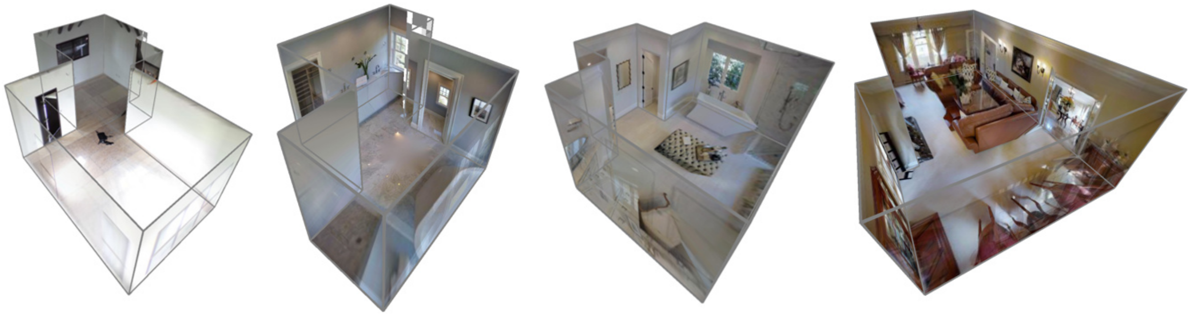

Figure 7. 3D layout visualizations based on the layout estimation produced by our LED^2-Net, for Matterport3D and Realtor360 datasets.

Table 5. The ablation study for the sensitivity of our model performance with respect to the number of casting rays (i.e., M).

Corner Number	16	64	256	1024
2D IoU (%) 3D IoU (%)	84.72 81.89	86.74 83.58	88.19 85.21	88.12 85.19

training on Matterport3D and testing on Realtor360 and Stanford2D3D, which demonstrates the benefit of designing a depth-based objective. Moreover, the result in the within-dataset ("Train-Dataset" in Table 4) setting for Realtor360 is also improved by around 1%. However, the performance on Matterport3D ("Train-Dataset") is slightly worse than the one without pre-training, and we argue that it is due to some specific characteristics (e.g., containing some outdoor scenes) in Matterport3D, which is less compatible with the scenes in Structure3D. While pre-training achieves improvement in most cases, we show the potential of our Differentiable Depth Rendering framework that involves depth pre-training to obtain more 3D prior information, which may inspire more future research on studying the cross-dataset setting or depth pre-training.

4.3. Effect of Ray-Casting Number

To study the effect of the model sensitivity with respect to the number of casting rays (i.e., M) used in the "Grid Re-sample" procedure, we conduct experiments using $M = 16, 64, 256,$ and 1024 in Table 5. From the results, while having more casting rays do provide a better approximation on the horizon-depth map (as the model performance increases from $M = 64$ to $M = 256$), the IoU starts to saturate when M grows up to be larger than 256. Taking the computational cost into consideration (where higher M costs more), we choose to adopt $M = 256$ as the default setting for all our experiments.

5. Conclusions

In this paper, we propose a differentiable L2D (layout-to-depth) procedure to convert the 360° layout representation into the 360° horizon-depth map, thus enabling the training objective for our layout estimation network to take advantage of 3D geometric information. We conduct extensive experiments on various datasets and achieve superior performance in comparison to several state-of-the-art baselines of monocular 360° layout estimation. Furthermore, as our proposed method is capable of adopting 360° depth datasets for model pre-training, it shows better generalizability for the cross-dataset evaluation scheme.

Acknowledgements. We thank iStaging for providing the Realtor360 dataset for research purpose. This project is funded by Ministry of Science and Technology of Taiwan (MOST 109-2634-F-007-016, MOST 110-2634-F-007-016, MOST 110-2636-E-009-001, MOST 110-2634-F-009-018, MOST Joint Research Center for AI Technology and All Vista Healthcare, and Taiwan Computing Cloud).

References

[1] 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. 2, 6

[2] 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 Virtual Conference on 3D Vision (3DV)*, 2017. 6

[3] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[4] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[5] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[6] J. M. Coughlan and A. L. Yuille. Manhattan world: compass direction from a single image by bayesian inference. In *IEEE International Conference on Computer Vision (ICCV)*, 1999. 3

[7] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so(3) equivariant representations with spherical cnns. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[8] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: a line segment detector. In *Image Processing On Line*, 2012. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 5

[11] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[12] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[13] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. Atlantanet: Inferring the 3d indoor layout from a single 360 image beyond the manhattan world assumption. In *European Conference on Computer Vision (ECCV)*, 2020. 3, 6

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 3

[15] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997. 5

[16] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[17] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360° imagery. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 5, 6

[19] Fu-En Wang, Hou-Ning Hu, Hsien-Tzu Cheng, Juan-Ting Lin, Shang-Ta Yang, Meng-Li Shih, Hung-Kuo Chu, and Min Sun. Self-supervised learning of depth and camera motion from 360° videos. In *Asian Conference on Computer Vision (ACCV)*, 2018. 2

[20] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[21] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Layoutmp3d: Layout annotation of matterport3d. *arXiv:2003.13516*, 2020. 6

[22] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6

[23] Qin Yang, Chenglin Li, Wenrui Dai, Junni Zou, Guo-Jun Qi, and Hongkai Xiong. Rotation equivariant graph convolutional network for spherical image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[24] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 5, 6, 7

[25] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Joint 3d layout and depth prediction from a single indoor panorama image. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[26] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *European Conference on Computer Vision (ECCV)*, 2014. 2, 3, 6

[27] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 6, 7

[28] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2

[29] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6

[30] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. 3d manhattan room layout reconstruction from a single 360 image. *arXiv preprint arXiv:1910.04099*, 2019. 2, 6, 7