DuLa-Net: A Dual-Projection Network for Estimating Room Layouts from a Single RGB Panorama

Shang-Ta Yang[1,2] sundadenny@gapp.nthu.edu.tw
Fu-En Wang[1] fulton84717@gapp.nthu.edu.tw
Chi-Han Peng[2] pchihan@asu.edu
Peter Wonka[2] pwonka@gmail.com
Min Sun[1] sunmin@ee.nthu.edu.tw
Hung-Kuo Chu[1] hkchu@cs.nthu.edu.tw
[1]National Tsing Hua University [2]KAUST

Abstract

We present a deep learning framework, called DuLa-Net, to predict Manhattan-world 3D room layouts from a single RGB panorama. To achieve better prediction accuracy, our method leverages two projections of the panorama at once, namely the equirectangular panorama-view and the perspective ceiling-view, that each contains different clues about the room layouts. Our network architecture consists of two encoder-decoder branches for analyzing each of the two views. In addition, a novel feature fusion structure is proposed to connect the two branches, which are then jointly trained to predict the 2D floor plans and layout heights. To learn more complex room layouts, we introduce the Realtor360 dataset that contains panoramas of Manhattan-world room layouts with different numbers of corners. Experimental results show that our work outperforms recent state-of-the-art in prediction accuracy and performance, especially in the rooms with non-cuboid layouts.

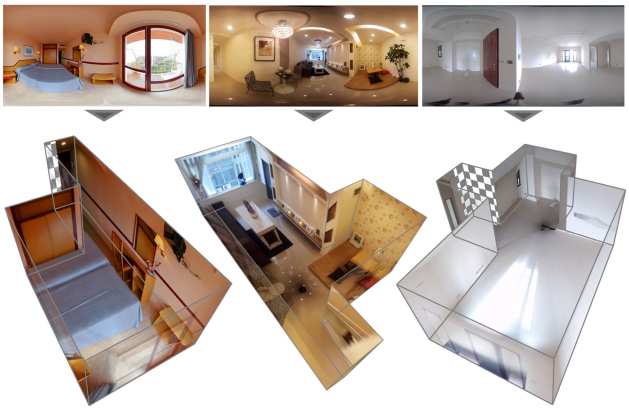

Figure 1. 3D room layouts with different complexity are estimated from a single RGB panorama using our system. (*Left to right*) Room layout with a floor plan of 6 corners, 8 corners, and 10 corners. The checkerboard patterns on the walls indicate the missing textures due to occlusion.

1. Introduction

Inferring high-quality 3D room layouts from indoor panoramic images plays a crucial role in indoor scene understanding and can be beneficial to various applications, including virtual/augmented reality and robotics. To that end, recent methods recover 3D room layouts by using deep learning to predict the room corners and boundaries on the input panorama. For example, LayoutNet [33] achieved impressive reconstruction accuracy for Manhattan world-constrained rooms. However, the clutter in the room, e.g. furniture, poses a challenge to extract critical corners and edges that are occluded in the input panorama. In addition, estimating 3D layouts from 2D corner and edge maps is an ill-posed problem and thus imposing extra constraints in the optimization. Therefore, it remains challenging to process complex room layouts.

In this work, we present a novel end-to-end framework to estimate a 3D room layout from a single RGB panorama. By the intuition that a neural network may extract different kinds of features given the same panorama but in different projections, we propose to predict the room layouts from two distinct views of the panoramas, namely the equirectangular *panorama-view* and the perspective *ceiling-view*. The network architecture follows the encoder-decoder scheme and consists of two branches, the *panorama-branch* and the *ceiling-branch*, for respectively analyzing images of the panorama-view and the ceiling-view. The outputs of panorama-branch include a *floor-ceiling probability map* and a *layout height*, while the ceiling-branch outputs a *floor plan probability map*. To share information between

branches, we employ a feature fusion scheme to connect the first few layers of decoders through a *E2P* conversion that transforms intermediate feature maps from equirectangular projection to perspective ceiling-view. We find that better prediction performance is achieved by jointly training the two connected branches. The final 2D floor plan is then obtained by fitting an axis-aligned polygon to a *fused floor plan probability map* (see Figure 3 for details) and then extruded by the estimated layout height.

To learn from panoramas with complex layouts, we need a proper dataset for network training and testing. However, existing public datasets, such as PanoContext [30] dataset, provide mostly labeled 3D layouts with simple cuboid shapes. To learn more complex layouts, we introduce a new dataset, *Realtor360*, which includes a subset of SUN360 [24] dataset (593 living rooms and bedrooms) and 1980 panoramas collected from a real estate database. We annotated the whole dataset with a custom-made interactive tool to obtain the ground-truth 3D layouts.

A key feature of our dataset is that it contains rooms with more complex shapes in terms of the numbers of the corners. The experimental results demonstrate that our method outperforms the current state-of-the-art method ([33]) in prediction accuracy, especially with rooms with more than four corners. Our method also takes much less time to compute the final room layouts. Fig. 1 shows some room layouts estimated by our method. Our contributions are summarized as follows:

- We propose a novel network architecture that contains two encoder-decoder branches to analyze the input panorama in two different projections. These two branches are further connected through a *feature fusion* scheme. This *dual-projection* architecture can infer room layouts with more complex shapes beyond cuboids and L-shapes.

- Our neural network is an important step towards building an *end-to-end* architecture. Our network directly outputs a probability map of the 2D floor plan. This output requires significantly less post-processing to obtain the final 3D room layout than the output of the current state of the art.

- We introduce a new data set, called Realtor360, that contains 2573 panoramas depicting rooms with 4 to 12 corners. To the best of our knowledge, this is largest data set of indoor images with room layout annotations currently available.

2. Related Work

There are multiple papers that propose a solution to estimate room layouts from a single image taken in an indoor environment. They mainly differ in three aspects: 1) the assumptions of the room layouts, 2) the types of the input images, and 3) the methods. In terms of room layout assumptions, a popular choice is the "Manhattan world" assumption [4], meaning that all walls are aligned with a global coordinate system [4, 23]. To make the problem easier to solve, a more restrictive assumption is that the room is a cuboid [8, 5, 13], i.e., there exist exactly four room corners. Our method adopts the Manhattan world assumption but allows for arbitrary numbers of corners.

In terms of types of input images, the images may differ in the FoV (field of view) - ranging from being monocular (i.e., taken from a standard camera) to 360° panoramas, and whether depth information is provided. The methods then largely depend on the input image types. The problem is probably most difficult to solve when only a monocular RGB image is given. Typically, geometric (e.g., lines and corners) [14, 8, 22] and/or semantic (e.g., segmentation into different regions [9, 10] and volumetric reasoning [7]) "cues" are extracted from the input image, a set of room layout hypotheses is generated, and then an optimization or voting process is taken to rank and select one among the hypotheses. Recently, neural network-based methods took stride in tackling this problem. A trend is that the neural networks generate higher and higher levels of information - starting from line segments [17, 31], surface labels [5], to room types [13] and room boundaries and corners [33], to make the final layout generation process increasingly easier to solve. Our method pushes this trend one step further by using neural networks to directly predict a 2D floor plan probability map that requires only a 2D polygon fitting process to produce the final 2D room layout.

If depth information is provided, there exist methods that estimate scene annotations including room layouts [28, 15, 29]. A deeper discussion is beyond the scope of this paper.

Closely related problems include depth estimation from a given image [32, 21] and scene reconstructions from point clouds [19, 18, 16]. Note that neither estimated depths nor reconstructed 3D scenes necessarily equate a clean room layout as such inputs may contain clutters.

360° panorama: The seminal work by Zhang et al. [30] advocates the use of 360° panoramas for indoor scene understanding for the reason that the FOV of 360° panoramas is much more expansive. Work in this direction flourished, including methods based on optimization approaches over geometric [6, 21, 26] and/or semantic cues [25, 27] and later based on neural networks [13, 33]. Except for LayoutNet [33], most methods rely on leveraging existing techniques for single perspective images on samples taken from the input panorama. We believe that this is a major reason of LayoutNet's superior performance since it performs predictions on the panorama as a whole, thus extracting more global information that the input panorama might contain.

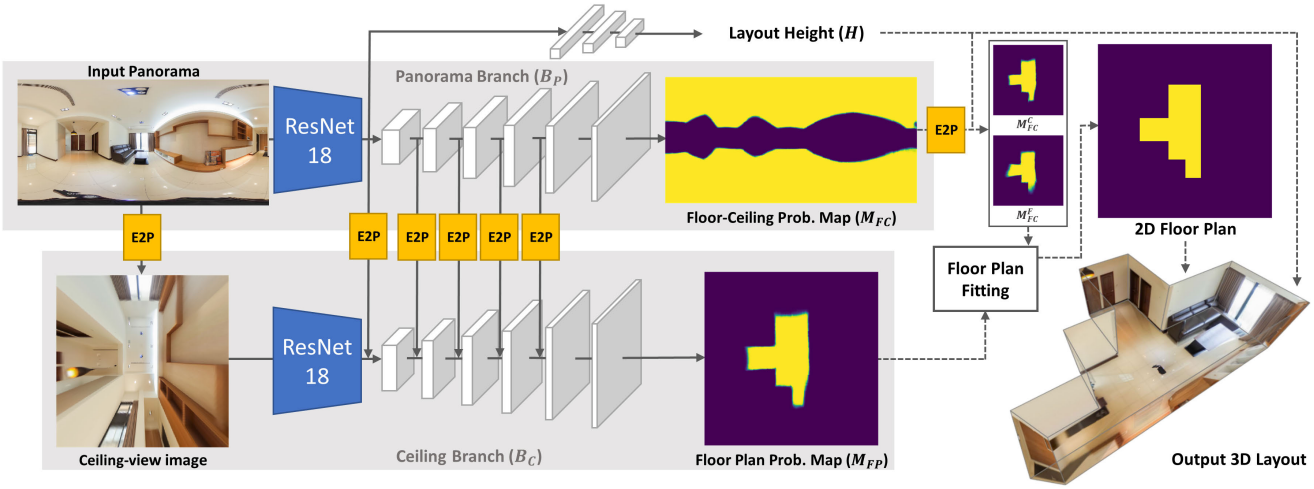

Figure 2. Our network architecture follows the encoder-decoder scheme and consists of two branches. Given a panorama in equirectangular projection, we additionally create a perspective *ceiling-view* image through a equirectangular-to-perspective (E2P) conversion. The panorama and the ceiling-view images are then fed to the *panorama-view* (upper) and *ceiling-view* (lower) branches. A E2P-based feature fusion scheme is employed to connect the two branches, which are jointly trained by the network to predict: 1) probability maps of the floor and ceiling in panorama view, 2) a floor plan in ceiling view, and 3) a layout height. Then, our system estimates a 2D floor plan by fitting a Manhattan-world aligned polygon to a weighted average of the three floor plans, which is further extruded using the predicted layout height to obtain the final 3D room layout.

A further step in this direction can be found in [21], in which the input panorama is projected to a 2D "floor" view in which the camera position is mapped to the center of the image and the vertical lines in the panorama become radial lines emanated from the image center. An advantage of this approach is that the room layout becomes a 2D closed loop that can be extracted more easily. We derived our "ceiling" view idea here - instead of looking *downward* toward the floor in which all the clutter in the room is included, we look *upward* toward the ceiling and got a more clutter-free view of the room layout.

3. Overview

Fig. 2 illustrates the overview of our framework. Given the input as an equirectangular panoramic image, we follow the same pre-processing step used in PanoContext [30] to align the panoramic image with a global coordinate system, i.e. we make a Manhattan world assumption. Then, we transform the panoramic image into a perspective ceiling-view image through an equirectangular to perspective (E2P) conversion (Sec. 4). The panorama-view and ceiling-view images are then fed to a network consisting of two encoder-decoder branches. These two branches are connected via a E2P-based feature fusion scheme and jointly trained to predict a floor plan probability map, a floor-ceiling probability map, and the layout height (Sec. 5). Two intermediate probability maps are derived from the floor-ceiling probability map using E2P conversion and combined with floor plan probability map to obtain a fused floor plan probability map. The final 3D Manhattan layout is determined by extruding a 2D Manhattan floor plan estimated on the fused floor plan probability map using the predicted layout height (Sec. 6).

4. E2P conversion

In this section, we explain the formulation of E2P conversion that transforms an equirectangular panorama to a perspective image. We assume the perspective image is square with dimension $w \times w$. For every pixel in the perspective image at position (p_x, p_y), we derive the position of the corresponding pixel in the equirectangular panorama, (p'_x, p'_y), $-1 \leq p'_x \leq 1, -1 \leq p'_y \leq 1$, as follows. First, we define the field of view of the pinhole camera of the perspective image as FoV. Then, the focal length can be derived as:

$$f = 0.5 * w * \cot(0.5 * \text{FoV}) \, .$$

(p_x, p_y, f), the 3D position of the pixel in the perspective image in the camera space, is then rotated by 90° or -90° along the x-axis (counter-clockwise) if the camera is looking upward (looking at the ceiling) or downward (looking at the floor), respectively.

Next, we project the rotated 3D position to the equirectangular space. To do so, we first project it onto a unit sphere by vector normalization, (s_x, s_y, s_z), and apply the following formula:

$$(p'_x, p'_y) = (\frac{arctan_2(\frac{s_x}{s_z})}{\pi}, \frac{arcsin(s_y)}{0.5\pi}), \qquad (1)$$

to project (s_x, s_y, s_z), the 3D position on the unit sphere, back to (p'_x, p'_y), the corresponding 2D position in the equirectangular panorama. Finally, we use (p'_x, p'_y) to interpolate a pixel value from the panorama. We note that this process is differentiable so it can be used in conjunction with backpropagation.

5. Network architecture

Our network architecture is illustrated in Fig. 2. It consists of two encoder-decoder branches, for the panorama-view and the ceiling-view input images. We denote the panorama-view branch as B_P and the ceiling-view branch as B_C . The encoder and decoder of B_P are denoted as E_{B_P} and D_{B_P} and for B_C they are denoted as E_{B_C} and D_{B_C} . A key concept is that our network predicts the floor plan and the layout height. With these two predictions, we can reconstruct a 3D room layout in a post-process (Sec. 6).

5.1. Encoder

We use ResNet-18 as the architecture for both E_{B_P} and E_{B_C} . The input dimension of E_{B_P} is $512 \times 1024 \times 3$ (the dimension of the input panorama) and the output dimension is $16 \times 32 \times 512$. For E_{B_C} , the input and output dimensions are $512 \times 512 \times 3$ and $16 \times 16 \times 512$. Note that the input of E_{B_C} is a perspective ceiling-view image generated by applying E2P conversion to the input panorama with FoV set to 160° and w set to 512. We also tried other more computationally expensive network architectures such as ResNet-50 for the encoders. However, we find no improvements in accuracy so we chose to work with ResNet-18 for simplicity.

5.2. Decoder

Both D_{B_P} and D_{B_C} consist of six convolutional layers. The first five layers are 3×3 resize convolutions [1] with ReLU activations. The last layer is a regular 3×3 convolution with sigmoid activation. The numbers of channels of the six layers are 256, 128, 64, 32, 16, and 1. To infer the layout height, we add three fully connected layers to the middlemost feature of B_P . The dimensions of the three layers are 256, 64, and 1. To make the regression of the layout height more robust, we add dropout layers after the first two layers. To take the middlemost feature as input, we first apply global average pooling along both x and y dimensions, which produces an 1-D feature with 512 dimension, and take it as the input of the fully connected layers.

The output of B_P is a probability map of the floor and the ceiling in the equirectangular projection, denoted as the *floor-ceiling probability map* (M_{FC}). For B_C , the output is a probability map of the floor plan in the ceiling view, denoted as the *floor plan probability map* (M_{FP}). Note that B_P also outputs a predicted layout height (H).

5.3. Feature fusion

We find that applying fusion techniques to merge the features in both B_P and B_C increases the prediction accuracy. We conjecture a reason as follows. In a ceiling-view image, the areas near the image boundary (where some useful visual clues such as shadows and furniture arrangements exist) are more distorted, which can have a detrimental effect for the ceiling-view branch to infer room structures. By fusing features from the panorama-view branch (in which distortion is less severe), performance of the ceiling-view branch can be improved.

We apply fusions before each of the first five layers of D_{B_P} and D_{B_C} . For each fusion connection, a E2P conversion (Sec. 4) with FoV set to 160° is taken to project the features in D_{B_P} , which are originally in the equirectangular view, to the perspective ceiling view. Each fusion works as follows:

$$f^*_{B_C} = f_{B_C} + \frac{\alpha}{\beta^i} \times f_{B_P},\ i \in \{0, 1, 2, 3, 4\}, \tag{2}$$

where f_{B_C} is the feature from B_C and f_{B_P} is the feature from B_P after applying the E2P conversion. α and β are the decay coefficients. i is the index of the layer. After each fusion, the merged feature, $f^*_{B_C}$, is sent into the next layer of D_{B_C} . The performance improvement of this technique is discussed in Sec. 8.

5.4. Loss function

For M_{FC} and M_{FP}, we apply binary cross entropy loss:

$$E_b(x, x^*) = -\sum_i x^*_i \log(x_i) + (1 - x^*_i) \log(1 - x_i). \tag{3}$$

For H (layout height), we use L1-loss:

$$E_{L1}(x, x^*) = \sum_i |x_i - x^*_i|. \tag{4}$$

The overall loss function is:

$$L = E_b(M_{FC}, M^*_{FC}) + E_b(M_{FP}, M^*_{FP}) + \gamma E_{L1}(H, H^*), \tag{5}$$

where M^*_{FC}, M^*_{FP} and H^* are the ground truth of M_{FC}, M_{FP}, and H.

5.5. Training details

We implement our method with PyTorch[20]. We use the Adam[11] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is 0.0003 and batch size is 4. Our training loss converges after about 120 epochs. For each training iteration we augment the input panorama with random flipping and horizontal rotatations by 0°, 90°, 180°, and 270°. For

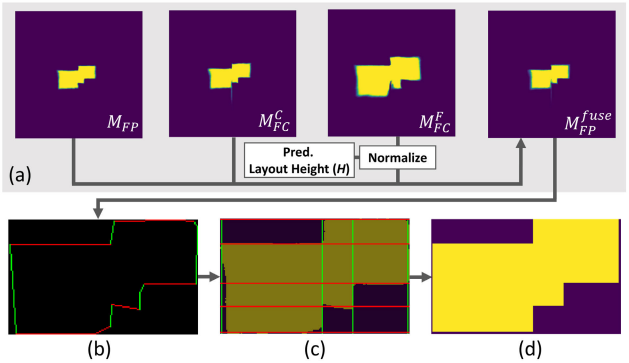

Figure 3. **2D floor plan fitting.** (a) The probability maps that our network outputs are fused to a floor plan probability map M_{FP}^{fuse}. (b) We apply image thresholding to M_{FP}^{fuse} and fit a polygon shape to the floor plan region. (c) The polygon edges are regressed and clustered into two sets of horizontal lines (red) and vertical lines (green). (d) The final floor plane shape is defined by grids in (c) where the ratio of floor plan area is greater than 0.5.

fusion, we set α and β in Eqn. 2 to be 0.6 and 3. We set the γ in Eqn. 5 to be 0.5. Because we estimate the floor plan probability map in the ceiling view, we assume the distance between the camera and the ceiling to be 1.6 meters, and use this constant to normalize the ground truth.

6. 3D layout estimation

Given the probability maps (M_{FC} and M_{FP}) and the layout height (H) predicted by the network, we reconstruct the final 3D layout in the following two steps:

1. Estimating a 2D Manhattan floor plan shape using the probability maps.
2. Extruding the floor plan shape along its normal according to the layout height.

For step 1, two intermediate maps, denoted as M_{FC}^{C} and M_{FC}^{F}, are derived from ceiling pixels and floor pixels of the floor-ceiling probability map using the E2P conversion. We further use a scaling factor, $1.6/(H-1.6)$, to register the M_{FC}^{F} with M_{FC}^{C}, where the constant 1.6 is the distance between the camera and the ceiling. Finally, a fused floor plan probability map is computed as follows:

$$M_{FP}^{fuse} = 0.5 * M_{FP} + 0.25 * M_{FC}^{C} + 0.25 * M_{FC}^{F}. \quad (6)$$

Fig. 3 (a) illustrates the above process. The probability map M_{FP}^{fuse} is binarized using a threshold of 0.5. A bounding rectangle of the largest connected component is computed for later use. Next, we convert the binary image to a densely sampled piece-wise linear closed loop and simplify it using the Douglas-Peucker algorithm (see Fig. 3 (b)). We run a regression analysis on the edges and cluster them into sets of axis-aligned horizontal and vertical lines. These lines divide the bounding rectangle into several disjoint grid cells (see Fig. 3 (c)). We define the shape of the 2D floor plan as the union of grid cells where the ratio of floor plan area is greater than 0.5 (see Fig. 3 (d)).

Table 1. Statistics of the Realtor360 dataset.

4 corners	6 corners	8 corners	10+ corners	Total
1246	950	316	61	2573

Figure 4. Few example panoramas in Realtor360. The annotated 3D room layouts are drawn as blue wireframes.

7. Realtor360 dataset

A dataset that contains a sufficient number of 3D room layouts with different numbers of corners is crucial for training as well as testing our network. Unfortunately, existing public domain datasets, such as the PanoContext [30] dataset and the Stanford 2D-3D dataset labeled by Zou *et al.* [33], contain mostly layouts with a simple cuboid shape. To prove that our framework is flexible enough to deal with rooms with an arbitrary number of corners, we introduce a new dataset, named Realtor360, that contains over 2500 indoor panoramas and annotated 3D room layouts. We classify each room according to its layout complexity measured by the number of corners in the floor plan. Table 1 shows the statistics of the dataset and a few visual examples can be found in Fig. 4. The source panoramic images in the Realtor360 dataset are collected from two sources. The first one is a subset of the SUN360 dataset [24], which contains 593 living rooms and bedrooms panoramas. The other source is a real estate database with 1980 indoor panoramas acquired from a real-estate company. We annotate the 3D layouts of these indoor panoramas using a custom-made interactive tool as explained below.

Annotation tool. To annotate the 2D indoor panoramas with high-quality 3D room layouts, we developed an interactive tool to facilitate the labeling process. The tool first leverages existing automatic methods to extract a depth map [12] and line segments [30] from the input panorama.

Method	Average		4 corners		6 corners		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 [33]	65.84	62.77	80.41	76.6	60.5	57.87	41.16	38.61	22.35	21.52
ours (fc-only)	75.2	72.02	76.75	73.27	76.04	73.06	70.8	67.89	56.42	54.2
ours (fp-only)	75.75	72.18	79.66	75.54	75.42	72.23	70.51	67.39	51.03	48.57
ours (w/o fusion)	78.52	74.8	81.77	77.57	78.5	75.1	73.61	70.37	57.01	54.12
ours (full)	**80.53**	**77.2**	**82.63**	**78.91**	**80.72**	**77.79**	**78.12**	**74.86**	**63.1**	**59.72**

Table 2. **Quantitative evaluation on the Realtor360 dataset.** We compare our method with the LayoutNet [33], and conduct an ablation study using different configurations of our method. Bold numbers indicate the best performance.

Method	Average		4 corners		6 corners		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 [33]	71.31	67.91	80.69	76.82	68.95	65.83	50.31	47.23	44.53	42.51
Ours (full)	**77.87**	**74.16**	**82.42**	**78.3**	**77.19**	**73.74**	**70.81**	**67.55**	**54.05**	**50.96**

Table 3. **Quantitative evaluation on the subset of Realtor360 dataset.** We compare with LayoutNet [33] using a training set that contains only rooms with cuboid layout (4 corners). Bold numbers indicate the best performance.

Then, an initial 3D Manhattan-world layout is created by sampling the depth along the horizontal line in the middle of the panorama. The tool allows the users to refine the initial 3D layout through a set of intuitive operations, including (i) pushing/pulling a wall; (ii) merging multiple walls; and (iii) splitting a wall. It also offers a handy function to snap the layout edges to the estimated line segments during the interactive editing to improve the accuracy.

8. Experiments

We compare our method to LayoutNet [33], a state-of-the-art method in room layout estimation, through a series of quantitative and qualitative experiments on our Realtor360 dataset and the PanoContext [30] dataset. We also conduct ablation study with several alternative configurations of our method. We adopt 2D and 3D Intersection over Union (IoU) to evaluate the accuracy of the estimated 2D floor plans and 3D layouts, which is a standard metric in similar tasks [3]. All the experiments used the same hyper-parameter discussed in Sec. 5.5. Fig. 5 shows a few 3D room layouts with different numbers of corners estimated using our method. Please refer to the supplementary materials for more results in the following experiments.

Evaluation on the Realtor360 dataset. To train both LayoutNet [33] and our DuLa-Net on the Realtor360 dataset, we randomly selected 2169 panoramas for training and took the remaining 404 panoramas for testing. We further classify the testing panoramas according to their numbers of corners. We run LayoutNet using the codes and default hyper-parameter released by the authors. The quantitative comparison with LayoutNet is shown in Table 2. We observe that LayoutNet delivers good performance on cuboid-shaped rooms (4 corners), similar to the numbers reported in their paper. However, the accuracy drops significantly as the number of corners increases. In comparison, Our DuLa-Net not only outperforms LayoutNet on cuboid-shaped rooms by a small margin (around 2%), but also performs well on rooms with larger numbers of corners. This leads to an overall performance gain of $\sim 14\%$ in both 2D and 3D metrics when compared to LayoutNet.

Since the 3D layout optimization and the hyper-parameter of LayoutNet were tuned on a dataset that contains mostly cuboid-shaped rooms, we conducted another experiment by training both networks on a revised training set that excludes rooms of non-cuboid layouts, while keeping the testing set untouched. Table 3 shows the quantitative results. Note that while the performance of LayoutNet improves, our method still outperforms on all kinds of rooms.

From the qualitative comparison shown in Fig. 6, we can observe a strong tendency of LayoutNet to predict the rooms to be cuboid-shaped, possibly due to the constraints imposed in their 3D layout optimization. In comparison, our method simplifies the problem by directly predicting a Manhattan-world floor plan without any assumptions about the numbers of corners. We conjecture that this is a main reason why our method outperforms LayoutNet, especially with rooms with more than four corners.

We also conducted an ablation study that evaluates the performance of our method in different configurations as follows: 1) *ours(fc-only)*: only panorama-view branch, 2) *ours(fp-only)*: only ceiling-view branch, and 3) *ours(w/o fusion)*: our full model but without feature fusion. The quantitative results in Table 2 shows that jointly training both branches leads to better performance than training only one of them. In addition, adding feature fusion between the two branches further improves the performance.

Evaluation on the PanoContext and Stanford 2D/3D datasets. LayoutNet provided quantitative results on the PanoContext [30] dataset with 414 panoramas for training

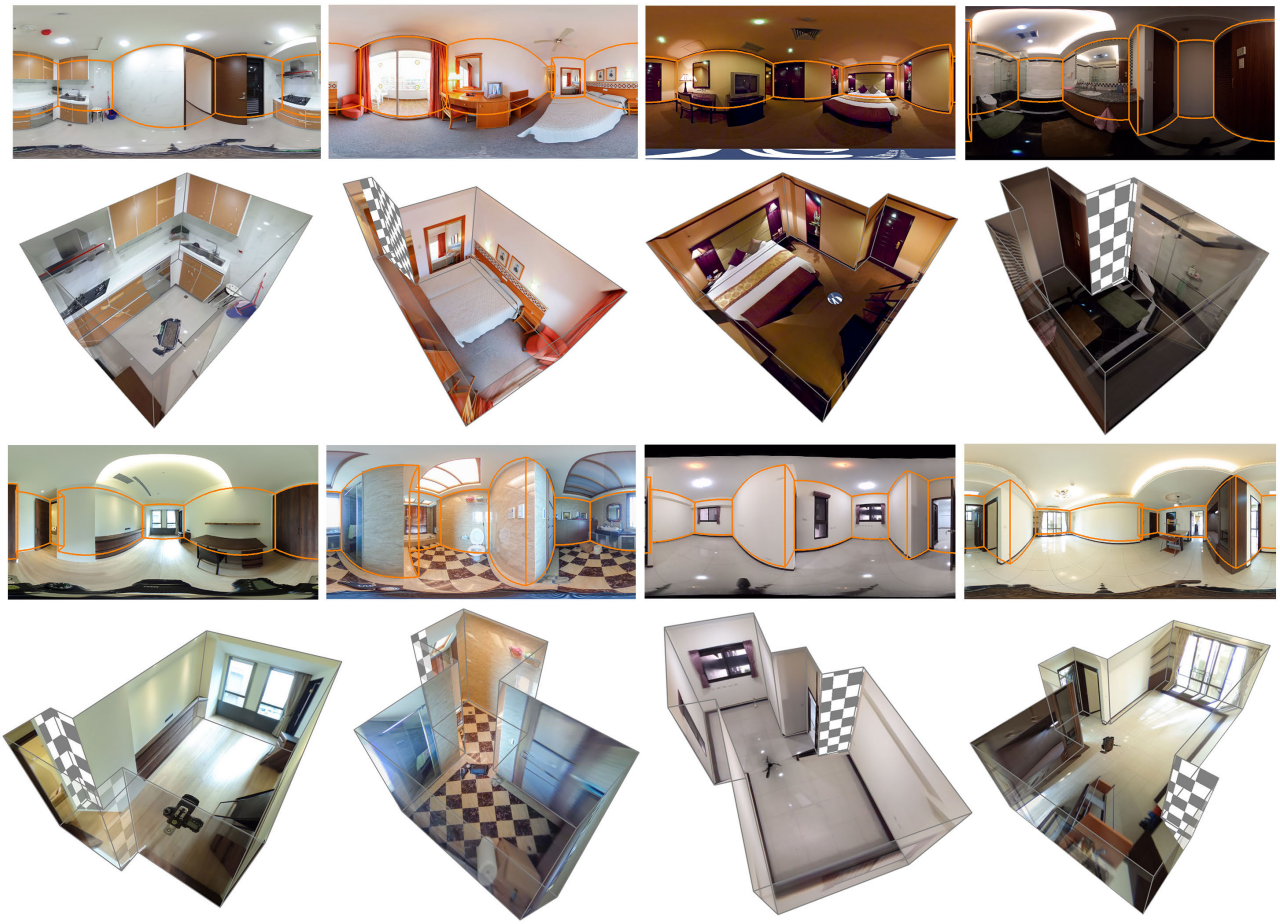

Figure 5. **Visual results.** Given a single RGB panorama, our method automatically estimates the corresponding 3D room layout. Our method is flexible to handle more complex room layout beyond the simple cuboid room. The checkerboard patterns on the walls indicate the missing textures due to occlusion.

and 53 panoramas for testing. All rooms are labeled as cuboid-shape. To compare, we trained our network on the same dataset. The quantitative comparison is shown in Table 4. Our model outperforms LayoutNet by a small margin.

We also evaluate our model on the Stanford 2D-3D [2] dataset with annotations labeled by LayoutNet [33]. The dataset includes 404 panoramas for training and 113 panoramas for testing. The last column in Table 4 shows the quantitative result on the Stanford 2D-3D [2] dataset.

Table 4. **Quantitative evaluation on the PanoContext [30] and Stanford 2D/3D [2] datasets in 3D IoU (%)**

Method	PanoContext	Stanford 2D-3D
LayoutNet [33]	74.48	76.33
Ours (full)	**77.42**	**79.36**

Timing. An end-to-end computation takes three main steps - 1) an alignment process to align the input panorama with a global coordinate system, 2) floor plan probability map prediction by our neural network, and 3) 2D floor plan fitting. Step 1) is most time-consuming, which takes about 13.37s measured on a machine with a single NVIDIA 1080ti GPU and Intel i7-7700 3.6GHZ CPU. Step 2) takes only 34.68ms and step 3) takes only 21.71ms.

Compared to LayoutNet, they carry out the same alignment process and their neural network prediction is also very fast (39ms). However, they needed another very time-consuming 3D layout optimization step in the end, which takes 30.5s. In summary, an end-to-end computation by LayoutNet takes about 43.9s while our method takes about 13.4s, a speed up of 3.28X.

9. Conclusion

We present an end-to-end deep learning framework, called DuLa-Net, for estimating 3D room layouts from a

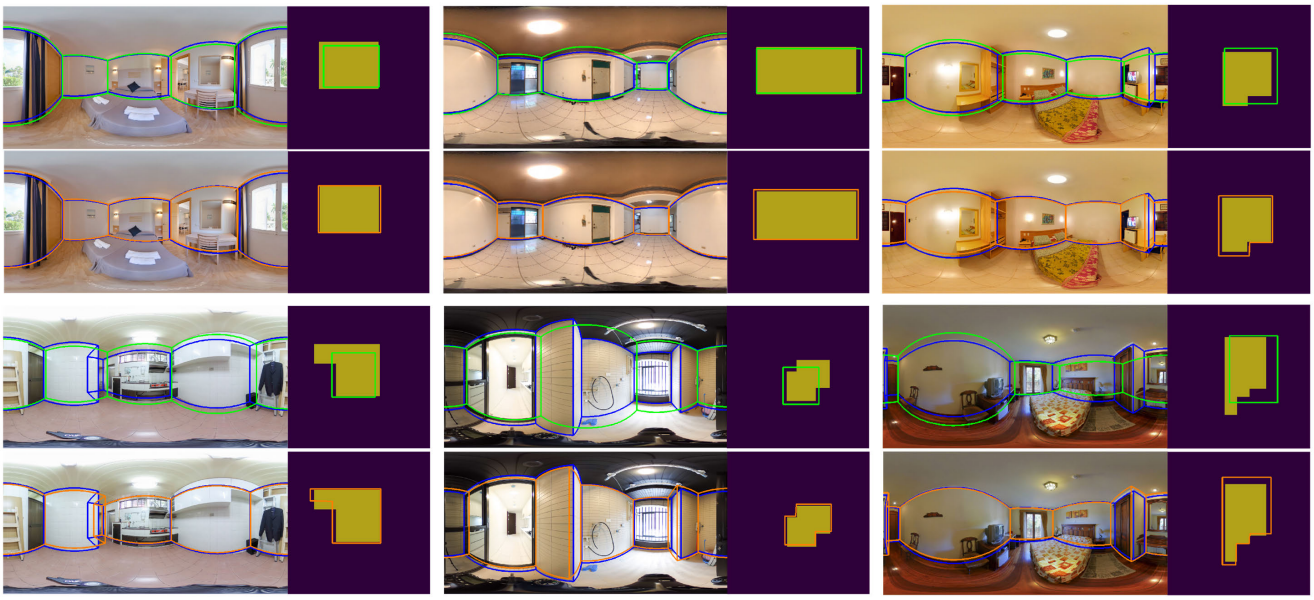

Figure 6. **Qualitative comparison with LayoutNet [33].** The 3D room layouts generated by LayoutNet[33] (green lines) and our method (orange lines). Results are displayed on both the equirectangular panorama-view (*left*) and floor plan view (*right*), where the blue lines and yellow solid shapes represent the ground truth, respectively.

single RGB panorama. We propose a new network architecture that consists of two encoder-decoder branches for analyzing features from two distinct views of the input panoramas, namely the equirectangular panorama-view and the perspective ceiling-view. The two branches are connected through a novel feature fusion scheme and jointly trained to achieve the best accuracy in the prediction of 2D floor plan and layout height. To learn from complex layouts, we introduce a new dataset, Realtor360, which contains 2573 indoor panoramas of Manhattan-world room layouts with various complexity. Both the quantitative and qualitative results demonstrate that our method outperforms the current state-of-the-art in prediction accuracy, especially with rooms with more than four corners, and take much less time to compute the final 3D room layouts.

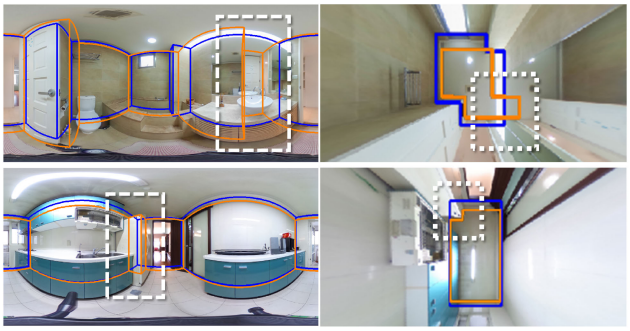

Figure 7. **Limitations.** Two failure cases generated by our method (orange lines) due to the lack of object semantics. (*Top*) Our method is misled by the reflection of mirror. (*Bottom*) The boundary of floor plan is occluded by the refrigerator. The ground truth layout is rendered in blue.

Limitations and future work. Our method has the following limitations: i) without knowing the object semantics, our network might get confused with the rooms that contains mirrors or large occluding objects as shown in Fig. 7; and ii) our approach of 3D layout estimation involves heuristics and assumptions that might over- or underestimate the underlying floor plan probability map and also restrain the results to Manhattan world. We propose to explore the following directions in the near future. First, introducing the object semantics, i.e., segmentation and labels, to the network architecture could potentially improve the accuracy by ignoring those distracting and occluding objects from the floor plan prediction. Second, designing a principled algorithm for a more robust 3D layout estimation, e.g., no Manhattan-world assumption and support rooms with curve shapes. Last but not the least, we believe that even better results can be achieved by experimenting with a larger range of encoders for our network architecture.

Acknowledgements. The project was funded in part by the KAUST Office of Sponsored Research (OSR) under Award No. URF/1/3426-01-01, and the Ministry of Science and Technology of Taiwan (107-2218-E-007-047- and 107-2221-E-007-088-MY3).

References

[1] Andrew P. Aitken, Christian Ledig, Lucas Theis, Jose Caballero, Zehan Wang, and Wenzhe Shi. Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. *CoRR*, abs/1707.02937, 2017. 4

[2] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, Feb. 2017. 7

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 6

[4] James M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. pages 941–, 1999. 2

[5] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 616–624, June 2016. 2

[6] Kosuke Fukano, Yoshihiko Mochizuki, Satoshi Iizuka, Edgar Simo-Serra, Akihiro Sugimoto, and Hiroshi Ishikawa. Room reconstruction from a single spherical image by higher-order energy minimization. *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1768–1773, 2016. 2

[7] Abhinav Gupta, Martial Hebert, Takeo Kanade, and David M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1288–1296. Curran Associates, Inc., 2010. 2

[8] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1849–1856, Sept 2009. 2

[9] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 654–661 Vol. 1, Oct 2005. 2

[10] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, Oct 2007. 2

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[12] 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 Computer Society, 2016. 5

[13] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. *CoRR*, abs/1703.06241, 2017. 2

[14] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143, June 2009. 2

[15] C. Liu, P. Kohli, and Y. Furukawa. Layered scene decomposition via the occlusion-crf. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–173, June 2016. 2

[16] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. *European Conference on Computer Vision (ECCV), 2018*, 2018. 2

[17] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 936–944, Washington, DC, USA, 2015. IEEE Computer Society. 2

[18] Aron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. Rapter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4):103:1–103:12, July 2015. 2

[19] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011. 2

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 4

[21] G. Pintore, V. Garro, F. Ganovelli, E. Gobbetti, and M. Agus. Omnidirectional image capture on mobile devices for fast automatic generation of 2.5d indoor maps. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, March 2016. 2, 3

[22] Srikumar Ramalingam and Matthew Brand. Lifting 3d manhattan lines from a single image. *2013 IEEE International Conference on Computer Vision*, pages 497–504, 2013. 2

[23] S. Ramalingam, J. K. Pillai, A. Jain, and Y. Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3065–3072, June 2013. 2

[24] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702, June 2012. 2, 5

[25] J. Xu, B. Stenger, T. Kerola, and T. Tung. Pano2cad: Room layout from a single panorama image. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 354–362, March 2017. 2

[26] H. Yang and H. Zhang. Efficient 3d room shape recovery from a single panorama. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5422–5430, June 2016. 2

[27] Yang Yang, Shi Jin, Ruiyang Liu, Sing Bing Kang, and Jingyi Yu. Automatic 3d indoor scene modeling from single panorama. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[28] J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sen-

sors. In *2013 IEEE International Conference on Computer Vision*, pages 1273–1280, Dec 2013. 2

[29] Yinda Zhang, Mingru Bai, Pushmeet Kohli, Shahram Izadi, and Jianxiong Xiao. Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding. *International Conference on Computer Vision (ICCV 2017)*, 2017. 2

[30] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pages 668–686, 2014. 2, 3, 5, 6, 7

[31] Hao Zhao, Ming Lu, Anbang Yao, Yiwen Guo, Yurong Chen, and Li Zhang. Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

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

[33] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 5, 6, 7, 8