HomeNet: Layout Generation of Indoor Scenes from Panoramic Images Using Pyramid Pooling

Krisha Mehta Yash Kotadia
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
{krisha.mehta, yash.kotadia}@djsce.edu.in

Abstract

We propose HomeNet, an end-to-end approach to generate 3D layouts for indoor scenes. It employs a Fully Convolutional Network (FCN) along with pyramid pooling to predict the main structure of the room using only a single 360 degree panorama. When provided with the input image, the FCN outputs the boundary and corner maps, which are further optimized using a significantly faster algorithm than previous approaches. Using this information, the main structure of the room is obtained, which undergoes affine transformations to generate the 3D layout. We find that using global prior representation obtained through pyramid pooling helps improve accuracy. When evaluated on the PanoContext and 2D-3D Stanford dataset, we find our model is more accurate than state-of-the-art methods while also being faster.

Figure 1. Examples of indoor layouts generated using HomeNet.

1. Introduction

Layout estimation of indoor scenarios has been an active field for more than a decade now. As humans, a 3D model can provide a clearer and more intuitive understanding of the structure of a room as compared to 2D images. It has various applications in the field of robotics, indoor navigation [13][15], augmented reality [17] and gaming [16]. The problem of layout estimation is an inverse problem where the goal is to recreate the original 3D structure as closely as possible, as shown in Fig. 1. While various solutions using perspective images have been developed in the past, we observe that a single perspective image fails to provide complete information of the room. To solve this problem, panoramic images can be used instead.

Our goal in this paper is to delineate the walls, ceiling, and floor of a room as accurately as possible based on a single panorama image. A panorama image ensures the availability of a larger field-of-view(FoV), which in turn helps contain more information about the room. Our method improves in accuracy with the state of the art methods while being faster. It works well on panoramic images. Our model works in the following way, as shown in Fig. 2. We use a panorama of the room as input. The input image is fed to a PSPNet [20] inspired network with a DenseNet-121 feature extractor that predicts the corner map and boundary map of the room. Accurate prediction of corners and boundaries is extremely crucial for an accurate layout generation of the room. We observe that an increase in the accuracy of corner prediction, increases overall accuracy. This is facilitated through pyramid pooling that provides a combination of local and global cues. In the following sections, we will describe our algorithm in greater detail as well demonstrate a comparative study we have drawn by developing different variants of our model.

2. Related Work

3D layout generation of indoor scenes from a single image is one of the oldest problems in computer vision. Solutions to the problem are being developed for more than

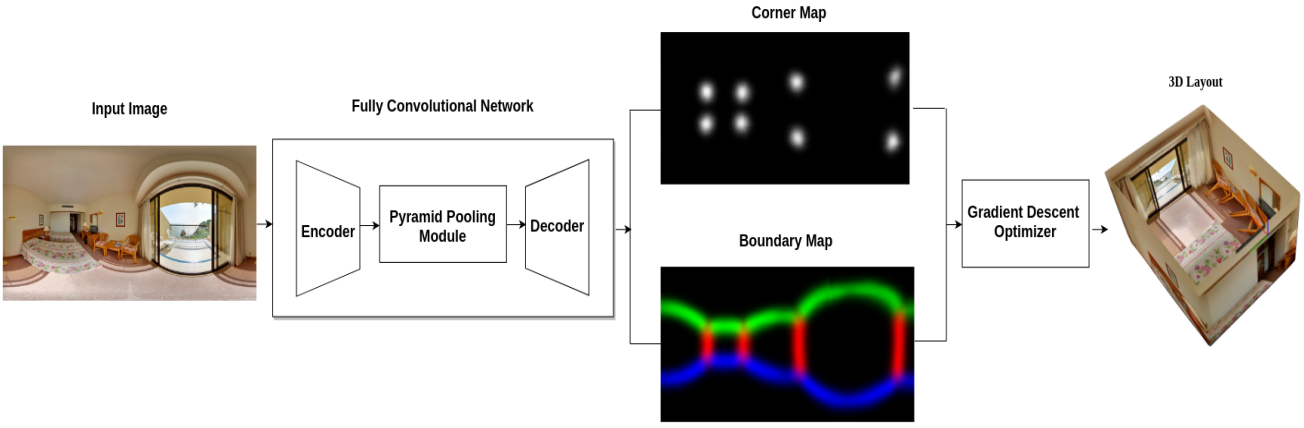

Figure 2. **HomeNet Architecture:** HomeNet uses an FCN with a pyramid pooling module. The FCN predicts a boundary map and corner map based on the panorama image provided. These maps, after optimization, are used to generate the 3D layout.

two decades now. However, learning and generating room layouts using different methods has been gaining much attention lately. Various methods involve the use of geometry, deep learning, or a combination of both to produce accurate layouts. Inputs can be in the form of perspective images, panoramas, or point clouds while the outputs can be 3D layouts of rooms [12, 21, 10, 5, 18] , floorplans [3] or 3D CAD models [9].

Replacing traditional methods of using geometric context and orientation maps to predict room layouts, Mallya et al. [12] instead use informative edge maps predicted using Structured Forests for Edge Detection [6] and FCN [11]. These maps, along with other information, ranked various box layouts. Following this, DeLay [5] uses an FCN for directly predicting per pixel semantic label. The output was further optimized to produce valid layouts. Both the approaches demonstrate that an FCN is more robust to clutter than previous techniques. This is very significant as most indoor scenes involve varying amount of clutter obstructing relevant edges and corners.

RoomNet [10] is one of the first ones to develop an end-to-end learning model. It uses perspective images to generate the room layout and the corresponding segmentation, by classifying rooms into 11 layouts based on the locations of an ordered set of key points. Meanwhile, Zhao et al. [19] developed a physics-inspired optimization technique as a new inference scheme, which is based on the mechanics concepts to estimate room layouts.

Perspective images have a smaller FoV as compared to panorama images. Hence, the amount of information contained in a perspective image can be quite less. The first approach extended from perspective images to use 360 panorama images for layout generation is PanoContext [18]. Using panorama images as input, it generates 3D bounding boxes of the scene as well as the objects contained within the room. The generation of different 3D hypotheses, including a 3D cuboid hypotheses for significant objects present in the scene, are then ranked based on Orientation Maps and geometric context. The hypothesis that provides the best holistic representation of the whole room is finally chosen. Approaches [21, 7] that followed PanoContext were, in general, more direct. LayoutNet [21] is one of the first methods to generate 3D layout using panorama as well as perspective images. Along with the input image, a Manhattan line map is fed to an encoder-decoder arrangement with skip connections similar to that in U-Net [14]. The decoder has two branches that predict the boundary map and the corner map simultaneously. The joint prediction of the two maps helps increase the accuracy of the model, but it also leads to an increase in model size and number of redundant parameters learned. More recently, PanoRoom [7] predicts edge and corner maps as the output of an FCN to generate 3D layout. Using ResNet-50 that is pretrained on ImageNet allows PanoRoom to converge faster than previous models. Unlike LayoutNet [21] it consists of a single decoder whose output has two channels which reduce the number of parameters needed for training as well as the computing time. Though our end goal is similar to that of LayoutNet, our approach is different. We adopt pyramid pooling that helps us capture global contextual information more effectively. This, in turn, helps improve the accuracy of the boundary map and the corner map that we generate.

3. Approach

In this section, we describe our approach for generating the layout for cuboid rooms. We adopt an end-to-end fully convolutional network with a pyramid pooling layer for predicting the boundary map and corner map of the room given its panoramic image. We further optimize the obtained lay-

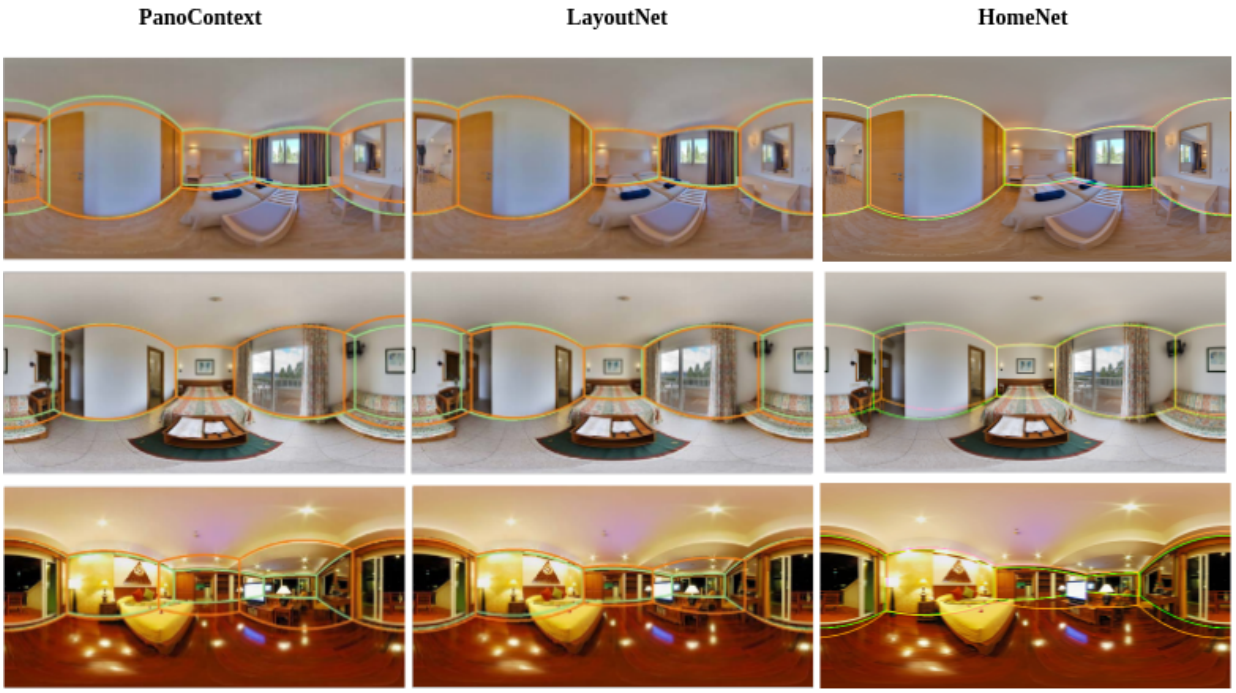

Figure 3. **Qualitative results (randomly sampled) for layout prediction based on the PanoContext dataset [18]:** We compare our model HomeNet with two previous approaches PanoContext [18] and LayoutNet [21]. Each image consists predicted layout from given method (orange lines) and ground truth layout (green lines). Best viewed in color.

out by post-processing it for better performance. Finally, the layout thus obtained is projected in 3D space for visualization.

3.1. Capturing the input image

To capture maximum details of the target room, we use the panoramic image of the room as proposed by [18]. The panoramic image captured must cover 360 horizontal field of view under equilateral projection thus covering the entire room. Such panoramic images can be easily captured with a smartphone camera by using the Google Street View application [1] which employs image stitching to obviate the requirement of a camera array. The proposed pyramid pooling layer vastly benefits from the rich contextual information captured in the panoramic images. The contextual information improves the predictions of the boundary map and the corner map. It handles occlusions better and makes use of logic such that a chair would be over the floor and a fan would be on the ceiling, thus improving the overall accuracy.

3.2. Network Architecture

We design a Fully Convolutional Network to predict the corner probability map and the boundary probability map. We adopt the PSPNet [20] architecture which is an encoder-decoder architecture enhanced with a pyramid pooling module.

3.2.1 PSPNet

The input to the network is a 3 channel RGB panoramic image of the target room. The resolution of the images used is 512 x 1024, however, since HomeNet is a fully convolutional network, it supports images of varying sizes. We extract the feature maps of the image using the pretrained DenseNet121 [8] network. These feature maps are then fed to the Pyramid Pooling Module inspired by [20]. The most naive pooling operation would be global pooling for global contextual understanding. The pyramid pooling module, however, enhances the performance by using smaller filters and essentially dividing the scene into a grid. It captures features under multiple pyramid scales to extract the most important contextual information in each grid. This pooling mechanism allows for more information to be captured than simple global pooling. The pooling strategy leads to a significant reduction in corner error compared to the previous state-of-the-art approach as shown in Table 1. As the output of the pooling at multiple scales is of varying sizes, it is upsampled to a common size using bilinear interpolation. These maps are then concatenated with the extracted feature maps followed by a deconvolutional layer to output the desired boundary map and the corner map.

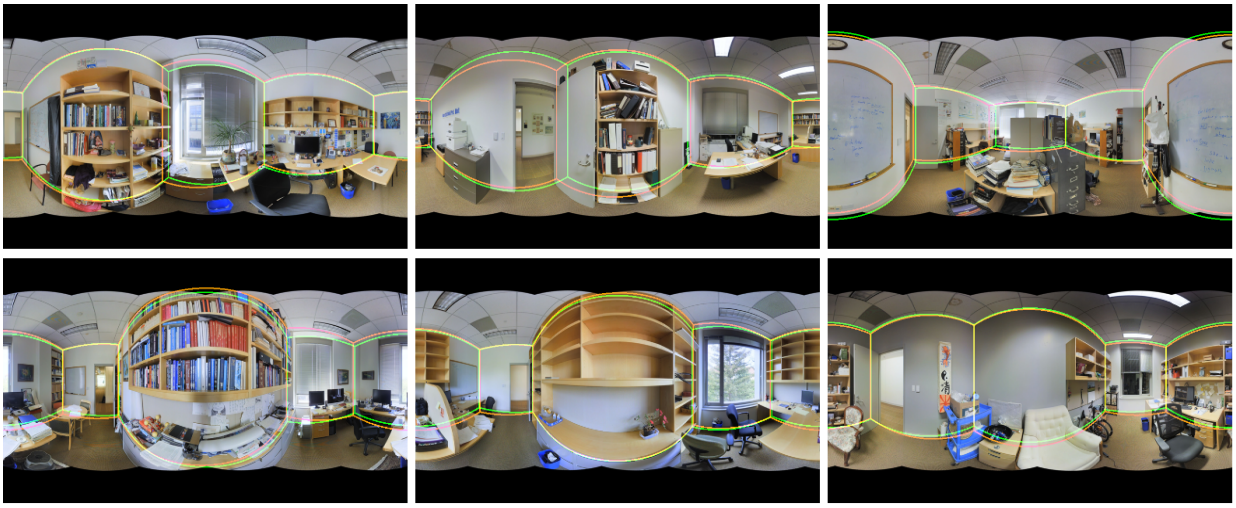

Figure 4. **Qualitative results (randomly sampled) on the Stanford 2D-3D annotation dataset.** The Stanford Dataset is more complex. Here, orange lines show HomeNet's layout prediction while green lines show the ground truth. Best viewed in color.

3.2.2 Loss Function

The loss function is calculated as shown in Eq. 1:

$$L(B_p, C_p) = -\sum_{\overline{y_b} \in B_p} [y_b . \log \sigma(\overline{y_b}) + (1 - y_b) . \log(1 - \sigma(\overline{y_b}))] - \sum_{\overline{y_c} \in C_p} [y_c . \log \sigma(\overline{y_c}) + (1 - y_c) . \log(1 - \sigma(\overline{y_c}))] \quad (1)$$

The loss combines sigmoid layer and binary cross entropy of the predicted probability maps B_p and C_p. Here, $\overline{y_b}$ is the predicted probability of a pixel in B_p and y_b is the ground truth of the same pixel. Similarly, $\overline{y_c}$ is the predicted probability of a pixel in C_p and y_c is the ground truth of the same pixel. The combination of the sigmoid layer with binary cross entropy, as opposed to sequential operations, is numerically more stable since it takes advantage of the log-sum-exp trick for numerical stability. Note that [21] uses an additional term calculating the Euclidean distance of the 3D cuboid parameters in the 3D parameter regressor. We do not employ it since the 3D parameter regressor leads to marginal improvement.

3.3. Gradient Descent Optimizer

The layout is expected to be cuboid and follow the Manhattan World assumptions [4]. Based on the predicted boundary map and corner map, we first extract the 3D parameters $(l, w, h, tx, tz, \theta)$ where l is the length of the cuboid, w is the width, h is the height, (tx, tz) determine the translation of floor w.r.t. x-z axis and θ is the rotation of the cuboid in the x-z axis. After sampling boundary and corner points from the obtained cuboid, we apply affine transformations to project the points to the original equirectangular image. We then calculate the loss as the error in resampling the boundary and corners from the cuboid layout and apply the gradient descent optimizer to minimize the loss w.r.t. the 3D parameters.

4. Experiments

4.1. Datasets

We evaluate our approach on two benchmark datasets and show their results in Fig. 3 and Fig. 4:

4.1.1 PanoContext DataSet

The PanoContext dataset [18] contains 500 panoramic room images with annotated corners and boundaries. The dataset contains images of indoor environments such as living rooms and bedrooms.

4.1.2 Stanford 2D-3D Annotation Dataset

The Stanford 2D-3D dataset [2] is a large-scale indoor environment dataset containing extensive annotations including depth, surface normals, semantic annotations, camera metadata. It contains 1413 equirectangular images of indoor environments from either offices or educational campuses. The dataset does not, however, contain the applicable annotations and hence we use the 571 images annotated by LayoutNet [21]. All in all, our total dataset contains 1063 images carefully split into training-validation-testing as 75-10-15%. Since CNN's require a huge amount of data to train from scratch, we employ following augmentation techniques to improve the performance, 1) Gamma adjustment 2) Horizontal Flip 3) Horizontal Rotation 4) Contrast changes 5) Noise addition.

Test Dataset	Method	3D IoU(%)	CE(%)	PE(%)
PanoContext	LayoutNet	75.12	1.02	3.18
PanoContext	Ours	**84.18**	**0.69**	**2.06**
Stanford 2D-3D	LayoutNet	77.51	0.92	**2.42**
Stanford 2D-3D	Ours	**81.16**	**0.79**	2.6

Table 1. Comparitive study of Layoutnet and HomeNet

Method	3D IoU(%)	CE(%)	PE(%)
PanoContext [18]	67.23	1.60	4.55
LayoutNet [21]	74.48	1.06	3.34
Ours	**82.7**	**0.76**	**2.29**

Table 2. Performance on Panocontext

4.2. Training Details

The input to the network is the 360HFOV panoramic image of resolution 512 x 1024 and the output is the boundary map and corner map of the same resolution. We apply backpropagation across the network. We use PyTorch, an open source deep-learning framework for training the architecture. The network is trained using Adam optimization algorithm. The initial learning rate is set to 0.0001 with decay rate for first moment set to 0.9 and second moment to 0.999. The network is trained on Google Colab which deploys the NVIDIA Tesla K-80 GPU. Training is done for 50 epochs and requires less than 10 hours.

4.3. Results

We evaluate our model on three metrics:

1) Corner Error(CE): Mean of the L2 error between predicted corners and ground truth normalized by the diagonal of the image. It gives the accuracy of the predicted corners.

2) Pixel Error(PE): Mean of the pixel-wise error between the ground truth and the layout prediction across the image. Evaluates the pixel surface error across the ceiling, floor, and walls.

3) 3D Intersection over Union(3D IoU): Calculates the intersection of the predicted surfaces over their union between the ground truth and predicted layout.

When trained and tested on the PanoContext dataset, our 3D IoU accuracy is 82.7% while the corner error is 0.76%. When trained on both the datasets,and tested on the PanoContext dataset, our 3D IoU accuracy is 84.18%. When tested on the Stanford dataset, our 3D IoU accuracy is 81.16%. On both the datasets, we perform better than state of the art.

Method	Approx. CPU Time (s)
PanoContext [18]	>300
LayoutNet [21]	44.73
Ours[1]	11.72 + 3.63 + 3.4 + 2 = **20.75**

Table 3. CPU Runtime Performance

4.4. Accuracy

The performance of the proposed approach on the PanoContext dataset is evaluated in Table 2. The current state-of-the-art LayoutNet [21] uses double decoders with skip connections between the corner decoder and boundary decoder. Comparatively, our model, by incorporating both global and local context as well as using a pretrained model performs much better on both datasets as seen in Table 1.

4.5. Runtime and Complexity

As shown by LayoutNet [21], the majority of processing time in the pipeline is taken up by optimization post inference by the deep learning model. As shown in Table 3, using only CPU with PyTorch, the alignment takes 11.72s, loading the encoder-decoder weights 3.63s, forward pass 3.4s and optimization 2s overall.

5. Discussion

We provide an evaluation of different variants of our model which include (i) ablation study of optimization,(ii) ablation study of Pooling Layer Size. The gradient descent optimizer we use improves our results significantly. Table 4 shows how using the optimizer helps improve the 3D IoU accuracy by more than 8% in the Stanford 2D-3D dataset and by more than 6% in the PanoContext dataset.

Dataset	Model	3D IoU(%)	CE(%)	PE(%)
PanoContext	w/ gradient descent optimizer	**84.18**	**0.69**	**2.06**
PanoContext	w/o gradient descent optimizer	77.55	0.95	2.75
Stanford 2D-3D	w/ gradient descent optimizer	**81.16**	**0.79**	**2.6**
Stanford 2D-3D	w/o gradient descent optimizer	72.66	0.92	3.44

Table 4. Ablation study of Gradient Descent Optimizer

The original PSPNet [20] suggests the use of (1, 2, 3, 6) sizes of pyramid pooling. As shown in Table 5, we find that

Dataset	Pooling Layer Size	3D IoU(%)	CE(%)	PE(%)
PanoContext	1,2,3,6	**84.18**	**0.69**	**2.06**
PanoContext	1,2,3,6,10	81.08	1.15	2.65
Stanford 2D-3D	1,2,3,6	81.16	**0.79**	**2.6**
Stanford 2D-3D	1,2,3,6,10	**81.91**	0.81	2.65

Table 5. Ablation study of Pooling Layer Size

the use of finer grid, that is, pooling sizes (1, 2, 3, 6, 10) lead to marginal changes. It performs slightly better on the Stanford 2D-3D Dataset because finer contextual information aids with the more cluttered environment present in the dataset.

6. Conclusion

We propose HomeNet, an indoor layout generation algorithm using a single panorama image. Our approach is different as it incorporates global prior offering higher accuracy as compared to previous works. The post-processing optimization strategy proposed significantly increases the efficiency of the pipeline. Our approach works on different room layouts. Future work includes implementation of object recognition for improved local contextual understanding.

References

[1] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010. 3

[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. 4

[3] Ricardo Cabral and Yasutaka Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 628–635. IEEE, 2014. 2

[4] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 941–947. IEEE, 1999. 4

[5] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 616–624, 2016. 2

[6] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1841–1848, 2013. 2

[7] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cedric Demonceaux, and Jose J Guerrero. Panoroom: From the sphere to the 3d layout. *arXiv preprint arXiv:1808.09879*, 2018. 2

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[9] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *CVPR*, 2017. 2

[10] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4865–4874, 2017. 2

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[12] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 936–944, 2015. 2

[13] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016. 1

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[15] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 1

[16] Tim Tutenel, Rafael Bidarra, Ruben M Smelik, and Klaas Jan De Kraker. Rule-based layout solving and its application to procedural interior generation. In *CASA Workshop on 3D Advanced Media In Gaming And Simulation*, 2009. 1

[17] Jianxiong Xiao and Yasutaka Furukawa. Reconstructing the worlds museums. *International journal of computer vision*, 110(3):243–258, 2014. 1

[18] 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*, pages 668–686. Springer, 2014. 2, 3, 4, 5

[19] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 870–878, 2017. 2

[20] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In

[1]The runtime for other methods is evaluated on Intel Xeon 3.5GHz(6 cores CPU). Ours is computed on Intel Xeon 2.2GHz(1 core).

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2881–2890, 2017. 1, 3, 5

[21] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, 2018. 2, 3, 4, 5