結合深度學習與圖形最佳化方法之多視角室內全景影像三維格局重建

Sio-Keong Si
National Tsing Hua University
siokeongshi@gmail.com

Jheng-Wei Su
National Tsing Hua University
jhengweisu@gapp.nthu.edu.tw

Chi-Han Peng
National Chiao Tung University
pchihan@asu.edu

Kuo-Wei Chen
National Taiwan University of Science and Technology
chen51202@gmail.com

Felix Chang
iStaging Corp.
felix@istaging.com

Chih-Yuan Yao
National Taiwan University of Science and Technology
cyuan.yao@csie.ntust.edu.tw

Hung-Kuo Chu
National Tsing Hua University
hkchu@cs.nthu.edu.tw

ABSTRACT

We propose a novel framework to estimate room layouts from multiple panoramas taken inside the same room with registration. Our solution consists of the following major components. First, we propose a boxification line prediction network that can predict boxification lines for each panorama in the same room. Second, we propose a graph-cut based binary segmentation that produces room layouts with sharp corners and straight walls. Third, we also annotated one multi-view consistent layout dataset for this new layout prediction framework. Our quantitative results show an improvement over single-view room layout estimation algorithms.

CCS Concepts

•**Computing methodologies** → **Reconstruction;**

Keywords

Layout Reconstruction; Multi-view; Graph Cut; Deep learning

1. INTRODUCTION

The problem of indoor layout reconstruction is currently one of the popular issues in computer vision. The goal of indoor layout reconstruction is to predict the information including walls, ceilings, floors, etc., for the indoor space. Such information can be used for many different applications, such as creating VR or AR scenes, indoor navigation, floor plan estimation, etc.

CGW '21, July 12–13, 2021, Keelung City, Taiwan

DOI:

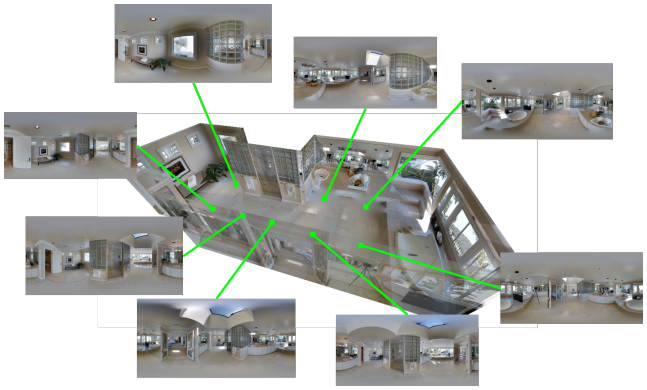

Figure 1: Our framework takes several panoramas as input, we can thus output one single 3D layout that is consistent among different views.

Many related jobs choose to use a single panorama to reconstruct the indoor layout, and there are many benefits to use a single panorama. One of the benefits is that panoramic images have more information. This feature has many advantages in terms of visualizing the results and the reconstruction problem itself.

However, using only a single panoramic image for reconstruction still encounters many problems. First, it is often impossible to see the full picture of the indoor layout from a single view because of the occlusion of the walls in some complex indoor scenes. Second, it is difficult to handle the large indoor scene, in which the distant wall usually only occupies a small area on the panoramic image, so it is difficult to reconstruct the entire details.

To achieve more accurate reconstruction, we use multiple views for layout reconstruction. There are two challenges in using multi-view panoramic images for Layout reconstruction: 1) how to estimate the camera extrinsic between multiple panoramic images; 2) how to integrate information from multiple panoramas. The main issue discussed in this paper is the second question. Assuming that we can obtain camera extrinsic from multiple panoramas, how to correctly

combine the information from each perspective to predict an accurate indoor layout? A direct integration method is to perform single-view predictions separately from each view and then align these single-view prediction results through camera extrinsic, and then perform a simple shape overlay. However, the disadvantage of using this method is that the prediction error comes from every single view are also accumulated, so the final results may not be more accurate than the single-view prediction.

We propose an algorithm that can better integrate the predictions from multiple panoramas and generates an accurate multi-view layout prediction. Our method is inspired by the post-processing in DuLa-Net [11] that predicts the layout from the ceiling view, and then derives the rectangular dividing lines from the layout segmentation, called boxification lines, and uses boxification lines on the ceiling view. We then divide the space into multiple small blocks, and then calculate the coverage of each grid on the ceiling viewing angle by layout segmentation, and finally pick out the grids with sufficient coverage. We converted this method to a multi-view scenario, redesigned the entire problem into a graph-cut problem, and added additional image feature information to remove the overlay error.

To verify the effectiveness of our method, an accurate dataset is needed. This dataset must have a consistent and accurate layout among different views. However, most of the current datasets are labeled on a single view. To this end, we have developed a set of multi-view indoor layout labeling tools. This system allows users to switch between different views to edit the same indoor layout. It can annotate accurate and consistent indoor layouts with multiple views in the complex indoor environment within minutes. Using this annotation tool, we have annotated 535 rooms on the Matterport3D [2] dataset, which can be used for verification or other related purposes. We tested our method on the multi-view dataset, showing that the performance of our proposed method outperforms the single view methods.

In summary, our contributions are as follows:

- We propose a framework that can reconstruct the layout from multiple panoramas.
- We annotated one multi-view consistent layout dataset for this new layout prediction framework from the annotation tool developed by ourselves.
- Our quantitative results show an improvement over single-view room layout estimation algorithms on our newly proposed multi-view dataset.

2. RELATED WORK

Perspective images layout reconstruction.

Using perspective images (compared to panoramic photos) for layout reconstruction has always been an important research problem in computer vision. Lee et al. [5] used deep learning to predict several layout corner distributions and positions on the image to reconstruct the indoor layout. Dasgupta et al. [3] proposed to use deep learning to predict which part of the layout each pixel belongs to, such as wall, ceiling, etc., and then find the most suitable pattern through optimization methods. Because of the limited viewing area of a single view, some related studies used multiple perspective images to reconstruct the multi-view indoor layout. Jenkins et al. [4] used deep learning to predict the position of the plane from multiple images which were clustered and analyzed to find the indoor layout without using the Manhattan hypothesis.

Single-view panorama layout reconstruction.

In recent years, many studies use deep learning for layout prediction on a single panoramic image. In addition to using panoramic images, Zou et al. [13] also added line features on panoramic images to assist in layout prediction. They use two branch networks to predict the indoor layout. The first branch network predicts the position of the ceiling and floor in the panoramic image, and the second branch network predicts the position of the corner on the panoramic image. Finally, they obtain the 3D indoor layout by a post-processing. Yang et al. [11] chose to obtain additional information from the ceiling view images, dividing the network into two branches, and predicting the difference between the panoramic view and the ceiling view images on the two branches. Two layout segmentation maps, and then combine the two segmentation maps to generate the final prediction. The advantage of this method is that a lot of unnecessary information, such as furniture, can be subtracted from the ceiling view images, so they could generate a more complete indoor layout shape. Sun et al. [10] proposed a one-dimensional representation for indoor layout. They trained the network to learn how to predict this one-dimensional representation through a panoramic image and then converted the one-dimensional representation into the indoor layout through post-processing. This method reduces the spatial complexity on the one hand compared to the method of directly predicting the entire panoramic image, on the other hand, it also makes the network easier to train.

Multi-view panorama layout reconstruction.

Many researches use multiple panoramas to reconstruct the indoor environment, such approaches can handle much more complex or large scenes. Cabral et al. [1] use Multi-View Stereo to predict point cloud from multiple panoramas, the point cloud is then projected to top-down view and discretized as a grid, find a closed path on the grid to forms a layout. Such an approach highly relies on the quality of point cloud and the camera position of panoramas can be improved by adding external point cloud. Pintore et al. [8] proposed to predict the superpixel segmentation on every single view, labeling the superpixel as floor, ceiling, or wall, predicts the height of each superpixel, then transform the superpixel to a common coordinate space, using the overlapping of superpixel from different views to estimate the shape of layout, Pintore et al. [7] extents this approach with object detection, which provide more cues of the interior object such as furniture, to get a more accurate boundary of the indoor layout. Another approach proposed by Pintore et al. [9] is to mix the corner points of multiple single view layout, the single view layout is overlapped, then the intersection of walls are detected to create new corner points, and the closed corner points are merged. Through this process multiple single view layouts are integrated into one multi-view layout.

3. METHOD

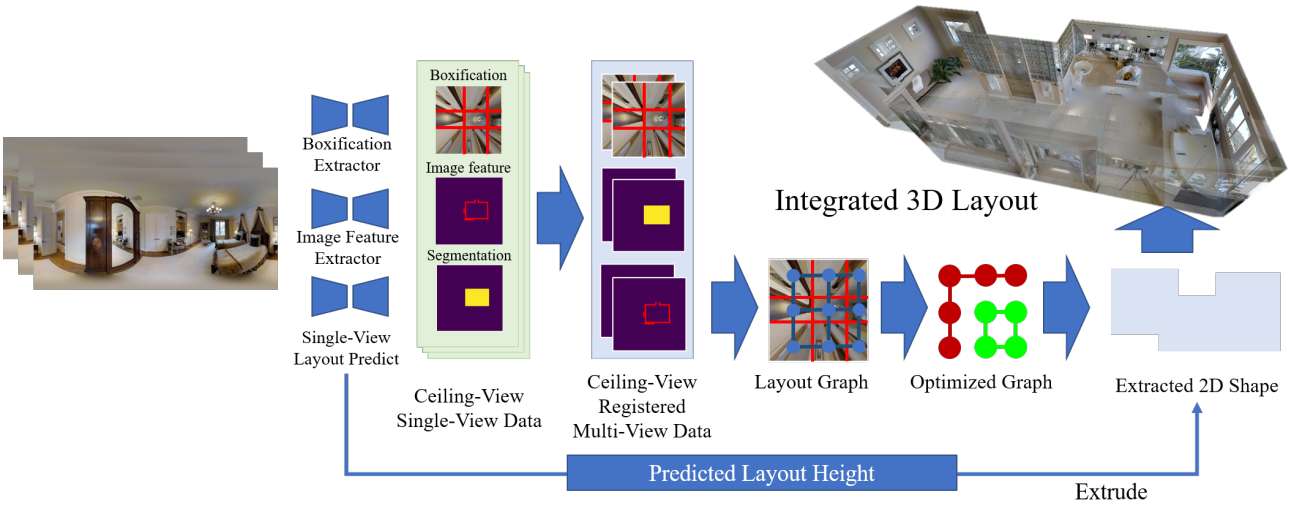

Figure 2: **Overview.**This system uses multiple aligned panoramas as input. First, we extract three different features from the panoramas of different views through deep learning. These features will be converted to ceiling view images. We use these features to build a graph, we optimize this graph to select vertices that are suitable for the layout and calculate the 2D layout from these vertices. We then convert the 2D layout into a 3D layout through the predicted height from a single view.

3.1 Overview

Our goal is to reconstruct a layout with Manhattan's assumption which implies all walls, ceilings, and floors are parallel to the three main coordinate axes. We first apply the same pre-processing as DuLa-Net[11], where the pre-processing can ensure that the panoramic image meets the Manhattan assumption, and then we take 160 degrees FoV, resampling the upper half of the panoramic image to obtain the ceiling view image that helps to remove information that does not belong to the layout, such as furniture, etc. In this paper, our methods are all operated from the ceiling view images. We first obtain segmentation, boxification, and image features from a single view. These data are all obtained or converted to the ceiling perspective after the ceiling view image is acquired. We obtain these three types of information for each view. After obtaining the single-view information, we separately convert the information of each view through the camera extrinsic matrix of each view and convert all the information to a common coordinate system for alignment. Next, we treat this process as a graph-cut optimization to obtain the final 2D layout. Finally, we add a predicted height to this 2D layout to output the 3D layout.

3.2 Single-view feature extraction

Segmentation and Image feature.

Segmentation refers to finding the pixels belonging to the ceiling from the ceiling view image. In the Manhattan layout, finding the shape of the ceiling means finding the layout. We use a state-of-the-art single-view layout prediction method, HorizonNet[10], to obtain single view layout prediction, and then convert the layout into ceiling view segmentation. In addition, in a single view, we will also obtain additional image features (LCNN[12]) to help us predict a more accurate layout.

Boxification line.

We use a deep neural network to predict boxification lines in a single view. Boxification lines can cut the ceiling into multiple squares aligned with the horizontal and vertical axes. We select the appropriate squares to form the best layout, just like DuLaNet[11] mentioned in their paper. To be able to predict accurate boxification, we designed a deep neural network architecture to help us make boxification line predictions. The boxification lines representation we use is similar to HorizonNet[10]. For each element on the vector, we calculate the distance d from the element to the nearest boxification lines and then calculate the value of the element c^d, where c Is a smoothing constant. In the experiment, we tried 0.96, 0.8, 0.7, 0.6, etc., and finally found that c taking 0.8 would have the best effect.

We design a neural network to predict the boxification lines. This network takes the ceiling view image as input and uses ResNet-50 to extract features. We follow feature pyramid network[6], by upsampling the high-level features in ResNet and concat them with the low-level features, we can mix the features at various scales. The 128x128 feature map is extracted through this modified ResNet-50, and then the feature map is average pooled in the two dimensions of width and height. This step is to distinguish the horizontal and vertical boxification information. After this network, we obtain these two one-dimensional vectors containing information in the horizontal and vertical directions respectively. Then, we refer to the decoder design in HorizonNet[10] and connect a fully connected layer for the features of each row on the one-dimensional vector. The one-dimensional vector in the horizontal direction or the one-dimensional vector in the vertical direction of this fully connected layer has common weights. Through this fully connected layer, we can start from the horizontal and vertical directions. We use binary cross entropy as our model's training loss.

3.3 Graph optimization

After we extract the LCNN line feature of the ceiling view images, we then add the line feature of the ceiling view im-

Table 1: **Ablation Study Result.** We replaced some components in our approach and compare the performance. The result shows our approach has the highest accuracy than other combinations of the method.

Boxification Prediction		Graph Optimization		MatterportLayout v2	
HorizonNet[10]	Ours	Coverage Only	Graph Cut	2D IoU ↑	3D IoU ↑
✓		✓		0.8427	0.8122
✓			✓	0.8540	0.8221
	✓	✓		0.8494	0.8198
	✓		✓	0.8676	0.8360

ages to the graph. We first use the boxification line to divide our ceiling view images into many vertices and edges. The edge represents the boundary between the regions represented by the vertices, and the line segment information represents the boundary information. We can obtain the boundary weight of this boundary by detecting the line segment information around the boundary represented by the edge. The boundary weight represents the possibility that a boundary is a true boundary, and when we calculate the boundary weight, we only consider the line segments that are in the same direction as the boundary.

We calculate the boundary weight $W_{boundary}$ by the following method: First, we define an influence distance $2D$ around the boundary, all line segments outside this distance will not affect the result, and then we remove the line segment beyond the boundary. Then, we calculate the distance d from the midpoint of the line segment to the boundary, as well as the length of the line segment projected to the boundary l_1 and the length of the rest of the boundary l_2.Fig 3 demonstrates an example of this process. Then, we calculate the boundary weight as follows:

$$W_{boundary} = (1 - \frac{d}{D}) * \frac{l_1}{(l_1 + l_2)} \quad (1)$$

For the weight of each vertex, we define it as the ratio of segmentation to the area represented by the vertex. And for the vertex area where the camera is, we set its weight to 1000. After passing the Ford-Fulkerson algorithm, we get the min-cut on the map, and divide the map into two parts, the inside and the outside of the layout. Usually, the vertices in this layout will be connected, but in rare cases(less than 2%), these vertices may be divided into more than two connected graphs, then we will pick one of the largest areas as the final output.

After we have selected the vertices in the layout, we can connect the areas represented by the selected vertices to predict the shape of the integrated layout. After adding a height to the layout as input, the 2D layout can be converted into a 3D layout. In our experiment, we averaged the predicted height of the single-view method to obtain the height of the entire room to predict the 3D layout.

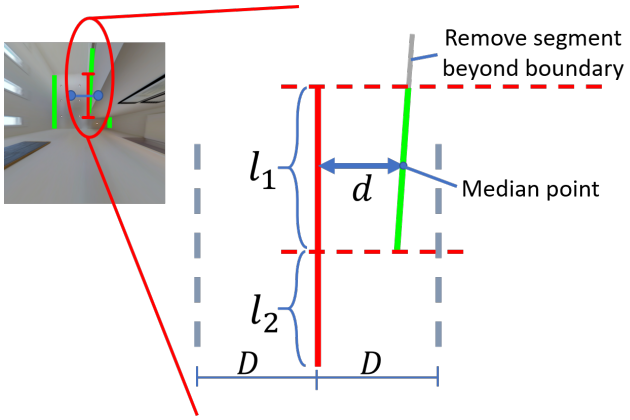

Figure 3: An example of boundary weight calculation. The image on top-left is the ceiling view image, the green lines in the image line segments predicted by LCNN, and the red lines are boxification line segments, we remove the unnecessary part of the image line segment(denoted as the grey line), then use the rest to calculate a boundary weight.

4. EXPERIMENTS

4.1 Dataset

To verify the effect of our method, an accurate dataset is needed. This dataset must have a consistent and accurate layout among different views. However, most of the current datasets are labeled from a single view. Such data is accurate enough in a single viewing angle, but in multiple viewing angles, it is often not accurate enough in other viewing angles due to occlusion or insufficient resolution.

To be able to quickly generate consistent and accurate multi-view layout data, we have developed a multi-view indoor layout labeling tool. This system allows users to switch between multiple viewing angles to edit the same indoor layout, allowing users to mark accurate and consistent indoor layouts within a few minutes. We annotated a new dataset called *MatterportLayoutv2* which consist of total of 535 rooms on the Matterport3D[2] dataset and a total of 2340 panoramic images. These multi-view indoor layouts can be used for verification or other related purposes. We used part of the data when training the network. We used 294 rooms with a total of 1213 views as the training set, and 141 rooms with a total of 582 views were used as the test set and 35 rooms with a total of 164 views were used as the validation set.

4.2 Quantitative Layout Estimation Evaluation

We compare our method with the current state-of-the-art single-view layout prediction method HorizonNet[10] and directly use its predicted layout for simple integration. We cal-

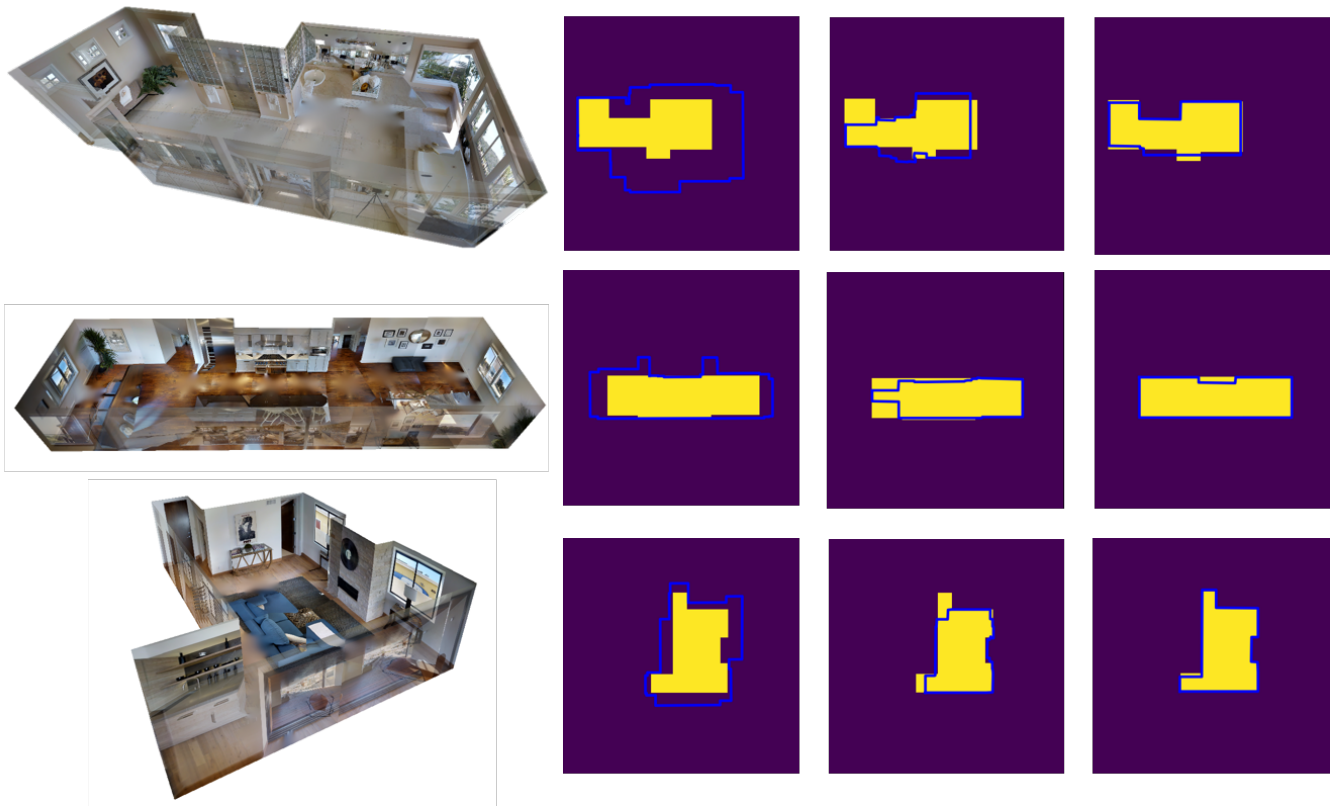

Figure 4: Results from left column to right column: Our 3D layout results, Union of HorizonNet [10], Thresholding on coverage, Our 2D layout results. The blue lines are the predicted layout and the yellow area is ground truth.

Table 2: **Layout Prediction Accuracy.** We compare our multi-view layout reconstruction with other baseline

Method	MatterportLayout v2	
	2D IoU ↑	3D IoU ↑
HorizonNet [10]	0.8045	0.7753
Union of HorizonNet [10]	0.8079	0.7786
Thresholding on coverage	0.8282	0.7990
Our method	**0.8676**	**0.8360**

culate the 2D IoU and 3D IoU between the predicted result and the ground truth. For a fair comparison, we also train HorizonNet[10] on our *Matterportlayoutv*2 dataset, and the comparison result is shown in Table 2. Among them, we compared three kinds of baselines. HorizonNet[10] refers to direct prediction on a single perspective and compares it with ground truth. Union of HorizonNet [10] refers to the comparison through simple layout unions, while Thresholding on coverage simply selects the vertex through the coverage of each vertex. If the coverage of the vertex is > 0.5, the vertex is selected as a part of the layout.

4.3 Ablation Study

To evaluate the several methods we proposed: single-view boxification prediction, graph optimization, we set up several ablation studies. We replaced the single-view boxification line prediction with boxification line extracted from the layout predicted by HorizonNet[10], and only use *coverage* $>$ 0.5 for vertex selection. We test the effect of adding each method on layout reconstruction accuracy. In Table 1, it shows that adding our boxification line or using Graph Cut can improve the overall accuracy. Similarly, using two methods at the same time can further improve the accuracy. In summary, the ablation study shows that the methods we proposed have a positive effect.

4.4 Qualitatively Layout Estimation Comparison

Figure 4 shows the comparison of our method with the different methods in Table 2 and the visual results of the large or occluded layout. The indoor layout predicted by our method is the best, showing that our process can handle this complex and large indoor layout well.

5. CONCLUSION

In this paper, we propose an algorithm that can predict the 3D layout from multiple panoramas taken in the same room. Our proposed algorithm can deal with complex large-scale layouts very well. Because of the occlusion problem and resolution limitation, it is impossible to reconstruct indoor details. By combining information from multiple images, we can produce a more accurate prediction of this large-scale layout than a single image.

7. REFERENCES

[1] R. Cabral and Y. Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, page 628–635, USA, 2014. IEEE Computer Society.

[2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[3] 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, 2016.

[4] H. Howard-Jenkins, S. Li, and V. Prisacariu. Thinking outside the box: Generation of unconstrained 3d room layouts. In *ACCV*, 2018.

[5] C. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. Roomnet: End-to-end room layout estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4875–4884, 2017.

[6] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[7] G. Pintore, F. Ganovelli, A. Jaspe, and E. Gobbetti. Automatic modeling of cluttered multi-room floor plans from panoramic images. *Computer Graphics Forum*, 38, 09 2019.

[8] G. Pintore, F. Ganovelli, R. Pintus, R. Scopigno, and E. Gobbetti. 3d floor plan recovery from overlapping spherical images. *Computational Visual Media*, 4:367–383, 2018.

[9] G. Pintore, R. Pintus, F. Ganovelli, R. Scopigno, and E. Gobbetti. Recovering 3d existing-conditions of indoor structures from spherical images. *Computers & Graphics*, 77, 09 2018.

[10] C. Sun, C. Hsiao, M. Sun, and H. Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1047–1056, 2019.

[11] S.-T. Yang, F.-E. Wang, C.-H. Peng, P. Wonka, M. Sun, and H.-K. 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*, pages 3363–3372, 2019.

[12] Y. Zhou, H. Qi, and Y. Ma. End-to-end wireframe parsing. In *ICCV 2019*, 2019.

[13] C. Zou, A. Colburn, Q. Shan, and D. 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.