Indoor Scene Modelling and Graph Generation from Single Panorama (TPA-1)

Aghin Shah Alin *CS17B109,* Mahima Raut *CS17B112*

Abstract

This term project is done as a part of the course on Computer Vision taught by Professor Sukhendu Das. In this report, we present a whole-room 3D context model to address the indoor scene understanding problem from a single panorama. The output is a 3D cuboid room layout with recognized scene objects represented by their 3D bounding boxes. For Indoor Scene 3-D Modelling, we've used Princeton's PanoContext dataset which provides 360° full-view panoramas for scene understanding and run inference on it to get a whole-room context model in 3D with bounding boxes of the room and all major objects inside, together with their semantic categories.

I. Introduction

CONTEXT is more powerful than we think. But the field of vision of cameras are too small, therefore it makes sense to use 360° panoramic pictures for indoor scene modelling. This way more contextual information can be used to recover spatial layouts. In this project we try to infer 3D structure of an indoor scene from a single 2D panorama by finding floor, ceiling, walls and recover shapes of typical indoor objects such as furniture. Depth information computed is used to perform 3D reconstruction of the scene. The resultant 3D mesh is used to generate scene graph. The dataset used here for this purpose is PanoContext [2]. We've implemented LayoutNetv2 in Pytorch for this purpose.

II. ALGORITHMIC DESCRIPTION

The entire algorithm can be divided into 2 parts mainly :

- Pre-processing of 360° Panoramic Images
- LayoutNet

*We used LayoutNet[4] to accomplish the task as described above.

A. *Pre-processing*

Given the input as a panorama that covers a 360° horizontal field of view, the first step of is to align the image to have a horizontal floor plane.

For this we estimate the orientation of floor plan under spherical projection using Zhang et al.'s approach [2] (i.e., selecting long line segments using the Line Segment Detector (LSD) in each overlapping perspective view), then we vote for three mutually orthogonal vanishing directions using the Hough Transform. Afterward, we rotate the scene and re-project it to the 2D equi-rectangular projection. This is the final aligned panoramic image and it is used as input to the LayoutNet.

Apart from this, the LayoutNet also takes Manhattan line feature map as an input in addition to the input panoramic picture to produce boundary maps and corner maps which when put through Manhattan Layout Optimizer, produces a 3D layout reconstruction. Therefore, LayoutNet additionally concatenates a 512x1024 Manhattan line feature map lying on three orthogonal vanishing directions using the alignment method.

B. *LayoutNet*

LayoutNet follows the encoder-decoder strategy. The network input is a concatenation of a single RGB panorama and Manhattan line map. The network jointly predicts layout boundaries and corner positions. The 3D layout parameter loss encourages predictions that maximize accuracy. The final prediction is a Manhattan layout reconstruction. The network structure is shown in Fig 1.

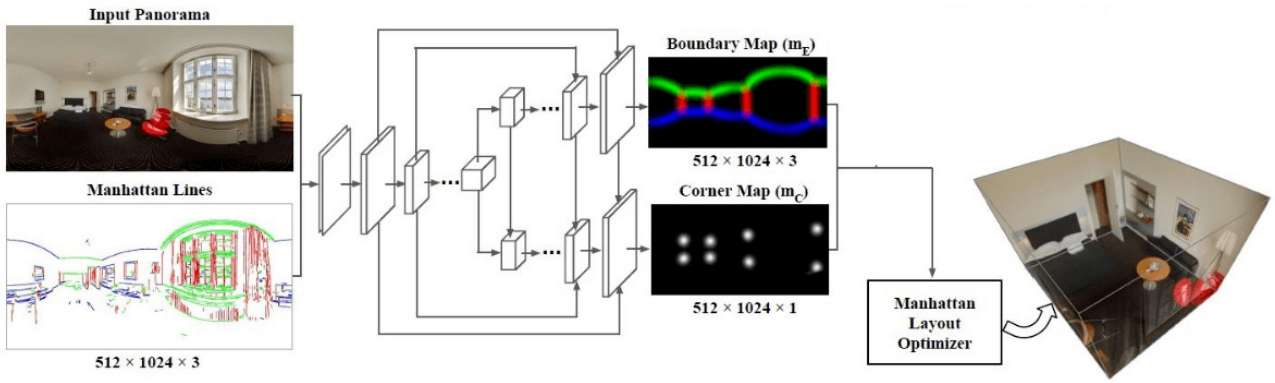


Fig. 1: Network architecture of LayoutNet. LayoutNet follows the encoder-decoder strategy. The network input is a concatenation of a single RGB panorama and Manhattan line map. The network jointly predicts layout boundaries and corner positions.The final prediction is a Manhattan layout reconstruction. Best viewed in color.

The Manhattan line feature map provides additional input features.

We use ResNet uniformly because we find that it shows better performance in capturing layout features than SegNet[3].

The encoder part of the LayoutNet is basically ResNet minus the last fully connected layer and the average pooling layer. This encoder receives a 512x1024 RGB panoramic image under equi-rectangular view as input.

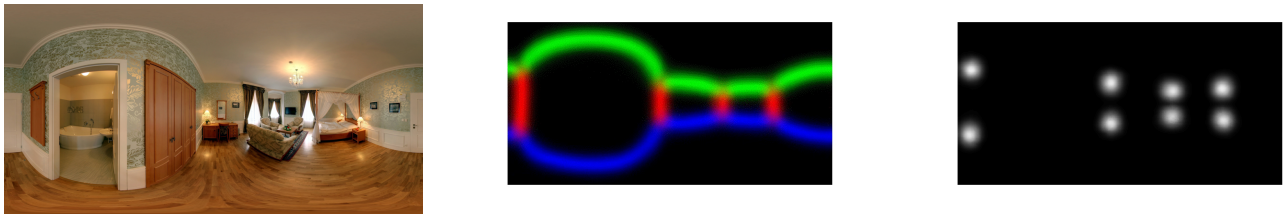
The decoder's output is predictions of layout pixels in the form of corner and boundary positions under equi-rectangular view.

1) Decoder structure: The decoder consists of two branches.

- The top branch, the layout boundary map (mE) predictor, decodes the bottleneck feature into a 2D feature map with the same resolution as the input. mE is a 3-channel probability prediction of wall-wall, ceiling-wall and wall-floor boundary on the panorama, for both visible and occluded boundaries.
- The lower branch, the 2D layout corner map (mC) predictor, follows the same structure as the boundary map predictor and additionally receives skip connections from the top branch for each convolution layer. This stems from the intuition that layout boundaries imply corner positions, especially for the case when a corner is occluded.

It's shown in [4] that the joint prediction helps improve the accuracy of the both maps, leading to a better 3D reconstruction result.

III. OUTPUT



(a) Input

(b) Boundary

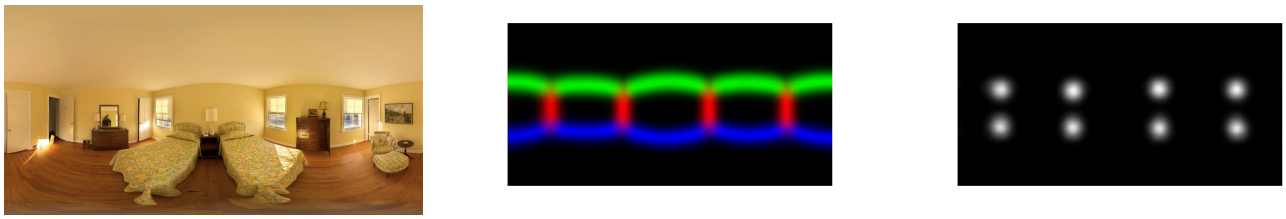
(c) Corners

Fig. 2

A. *Observation*

Describe your observation while doing the experiments.

- Figures 2 and 3 show the output of LayoutNet producing Boundary and Corner maps for the corresponding panoramic input image.
- Fig. 4 shows the 3D maps from 2 different angles of an aligned panoramic picture.

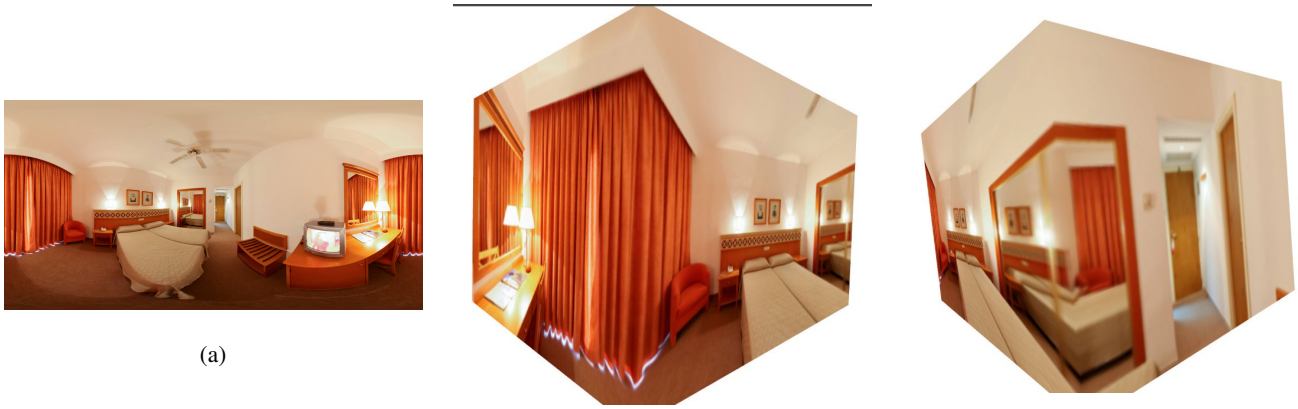


(a) Input

(b) Boundary

(c) Corners

Fig. 3



(a)

(b)

(c)

Fig. 4

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] Zhang Y, Song S, Tan P, Xiao J (2014) Panocontext: A whole-room 3d context model for panoramic scene understanding. In: European conference on computer vision, Springer, pp 668–686

[3] Manhattan Room Layout Reconstruction from a Single 360◦ image: A Comparative Study of State-of-the-art Methods

[4] Zou C, Colburn A, Shan Q, Hoiem D (2018) Layoutnet: Reconstructing the 3d room layout from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2051–2059