LGT-Net: Indoor Panoramic Room Layout Estimation with Geometry-Aware Transformer Network

Zhigang Jiang[1,2] Zhongzheng Xiang[2] Jinhua Xu[1*] Ming Zhao[2]
[1]East China Normal University [2]Yiwo Technology
zigjiang@gmail.com even_and_just@126.com jhxu@cs.ecnu.edu.cn zhaoming@123kanfang.com

1. Geometrical Relationship

Since the corners of PanoContext [9] and Stanford 2D-3D [1] datasets are annotated with longitude and latitude values and need to convert 3D points. Meanwhile, the prediction results of our network need to visualize on the panorama. To clearly illustrate the geometrical relationship of points between panorama and 3D space, we describe their transformation formula.

Conversion from 2D to 3D We convert a 2D point (θ, ϕ) on panorama to a 3D point $q = (x^q, y^q, z^q)$ on a unit sphere:

$$\begin{aligned} x^q &= \cos(\phi)\sin(\theta), \\ y^q &= \sin(\phi), \\ z^q &= \cos(\phi)\cos(\theta), \end{aligned} \tag{1}$$

where θ denotes longitude and is in the range $[-\pi, \pi]$, ϕ denotes latitude and is in the range $[-0.5\pi, 0.5\pi]$, as shown in Fig. 1.

Then, we convert the 3D point q to a 3D point $p = (x^p, y^p, z^p)$ on the floor (*i.e.*, $y = h^f$) plane or ceiling (*i.e.*, $y = -h^c$) plane:

$$\begin{aligned} x^p &= x^q \times \frac{y}{y^q}, \\ y^p &= y, \\ z^p &= z^q \times \frac{y}{y^q}. \end{aligned} \tag{2}$$

Conversion from 3D to 2D We convert a 3D point $p = (x, y, z)$ to a 2D point (θ, ϕ) on panorama:

$$\begin{aligned} \theta &= \arctan2(x, z), \\ \phi &= \arctan(\frac{y}{\sqrt{x^2 + z^2}}), \end{aligned} \tag{3}$$

where $\arctan2$ is 2-argument arctangent and returns a value in the range $[-\pi, \pi]$.

*Corresponding author.

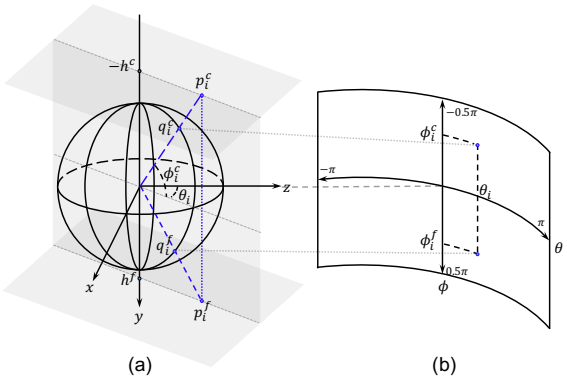
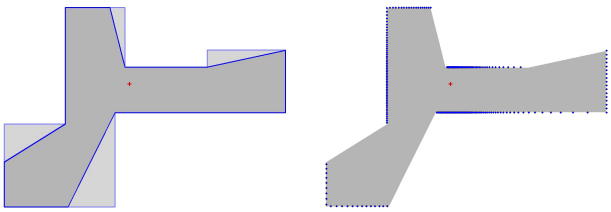


Figure 1. Illustration of the geometrical relationship between panorama and 3D space.



(a) Visibility polygon. (b) Sampling points.

Figure 2. Visualization of sampling visibility points.

2. Preprocessing

To get the ground truth of horizon-depth (mentioned in Section 3.2 of our main paper), we first get the 3D points of ordered floor corners from a label and connect them to get a polygon. Next, we use the algorithm proposed by Asano *et al*. [2] to obtain visible polygon, as shown in Fig. 2a. Then, increasing by $\frac{2\pi}{N}$ each time to get rays, meanwhile, calculating the intersection of the rays with the visible polygon to get sampling points $\{\bar{p}_i\}_{i=1}^{N}$, as shown in Fig. 2b. Finally, we use Eq. (4) to convert the sampled points to the ground truth of the horizon-depth sequence $\{\bar{d}_i = D(\bar{p}_i)\}_{i=1}^{N}$. Each depth of the horizon-depth sequence corresponds to $\frac{1024}{N}$

Method	Overall		4 corners		6 corners		8 corners		10+ corners	
	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU
LayoutNet v2 [10]	78.73	75.82	84.61	81.35	75.02	72.33	69.79	67.45	65.14	63.00
DuLa-Net v2 [10]	78.82	75.05	81.12	77.02	82.69	78.79	74.00	71.03	66.12	63.27
HorizonNet [6]	81.71	79.11	84.67	81.88	84.82	82.26	73.91	71.78	70.58	68.32
AtlantaNet [5]	82.09	80.02	84.42	82.09	83.85	82.08	76.97	75.19	**73.19**	**71.62**
LED^2-Net [7]	82.61	80.14	84.93	82.26	84.71	82.29	78.43	76.09	72.01	70.34
Ours	**83.52**	**81.11**	**85.67**	**82.95**	**86.24**	**83.97**	**79.55**	**77.64**	72.50	70.82

Table 1. Quantitative results(%) of general layout estimation evaluated on MatterportLayout [10] dataset.

Method	Overall		4 corners		6 corners		8 corners		10+ corners	
	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU
HorizonNet [6]	90.44	88.59	91.70	89.72	90.53	88.89	86.49	84.81	83.08	81.09
LED^2-Net [7]	90.36	88.49	91.63	89.64	90.40	88.71	86.26	84.50	83.80	81.93
Ours	**91.78**	**89.95**	**93.22**	**91.38**	**91.63**	**89.91**	**87.68**	**85.77**	**84.18**	**81.97**

Table 2. Quantitative results(%) of general layout estimation evaluated on ZInd [3] dataset.

Method	Train-Dataset MatterportLayout		Cross-Dataset ZInd		Train-Dataset ZInd		Cross-Dataset MatterportLayout	
	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU	2DIoU	3DIoU
HorizonNet [6]	81.71	79.11	-	-	90.44	88.59	67.04	64.74
LED^2-Net [7]	82.61	80.14	**80.54**	**78.18**	90.36	88.49	69.67	67.34
Ours	**83.52**	**81.11**	80.36	77.60	**91.78**	**89.95**	**69.73**	**67.38**

Table 3. Quantitative results(%) of cross-dataset evaluation scheme.

Method	2DIoU	3DIoU	Precision	Recall	F_1-score
HorizonNet [6]	90.17	88.32	72.57	80.31	74.53
LED^2-Net [7]	90.06	88.17	72.23	79.59	73.94
Ours	**91.50**	**89.69**	**82.85**	**82.80**	**82.00**

Table 4. Other quantitative results(%) **with post-processing** on ZInd [3] dataset.

columns in panoramic image.

$$D(p) = \sqrt{x^2 + z^2}. \tag{4}$$

3. More Quantitative Results

Corner Number We report the performance of room layouts with different numbers of corners on MatterportLayout [10] and ZInd [3] datasets, as shown in Tab. 1 and Tab. 2. Our approach offers better performance than all baselines.

Cross-Data We train our network and baselines on MatterportLayout [10] dataset and test on ZInd [3] dataset (the left part in Tab. 3). In addition, we train on ZInd dataset and test on MatterportLayout dataset (the right part in Tab. 3). We observed that the performance of all approaches was low on cross-dataset. We argue that there is a large domain gap between MatterportLayout dataset and ZInd dataset, and the main difference lies in annotation standard and furniture occlusion. In the future we will try to use combined multi-room room layout type of ZInd dataset to train our network.

F_1-Score ZInd dataset consists of 14.35% non-Manhattan layouts, and it is not suitable to use the current single post-processing method. Nevertheless, we reported the results with post-processing of Dula-Net [8] in Tab. 4. Meanwhile, We reported corner metrics (Precision, Recall and F_1-score) with 10 pixels as threshold in Tab. 4.

4. More Qualitative Results

To clearer comparison with other approaches and demonstrate the performance of our proposed approach, we show more qualitative results of MatterportLayout [10] and ZInd [3] datasets in Fig. 3 and Fig. 5, respectively. Meanwhile, we show more 3D layout visualizations of our approach, as shown in Fig. 4 and Fig. 6.

Figure 3. Qualitative comparison of general layout estimation on MatterportLayout [10] dataset. We show the room layouts without post-processing by HorizonNet [6], LED^2-Net [7], and ours. We show the boundaries of the room layout on panorama (left) and the floor plan (right). The blue lines are ground truth, and the green lines are prediction. Moreover, we visualize the predicted horizon-depth, normal, and gradient below each panorama, and the ground truth is at the top. The dashed white lines highlight the errors generated by the baselines.

Figure 4. The 3D visualization results of our approach on MatterportLayout [10] dataset. The green lines are predicted boundaries by our network, and the red lines are results with post-processing of the prediction.

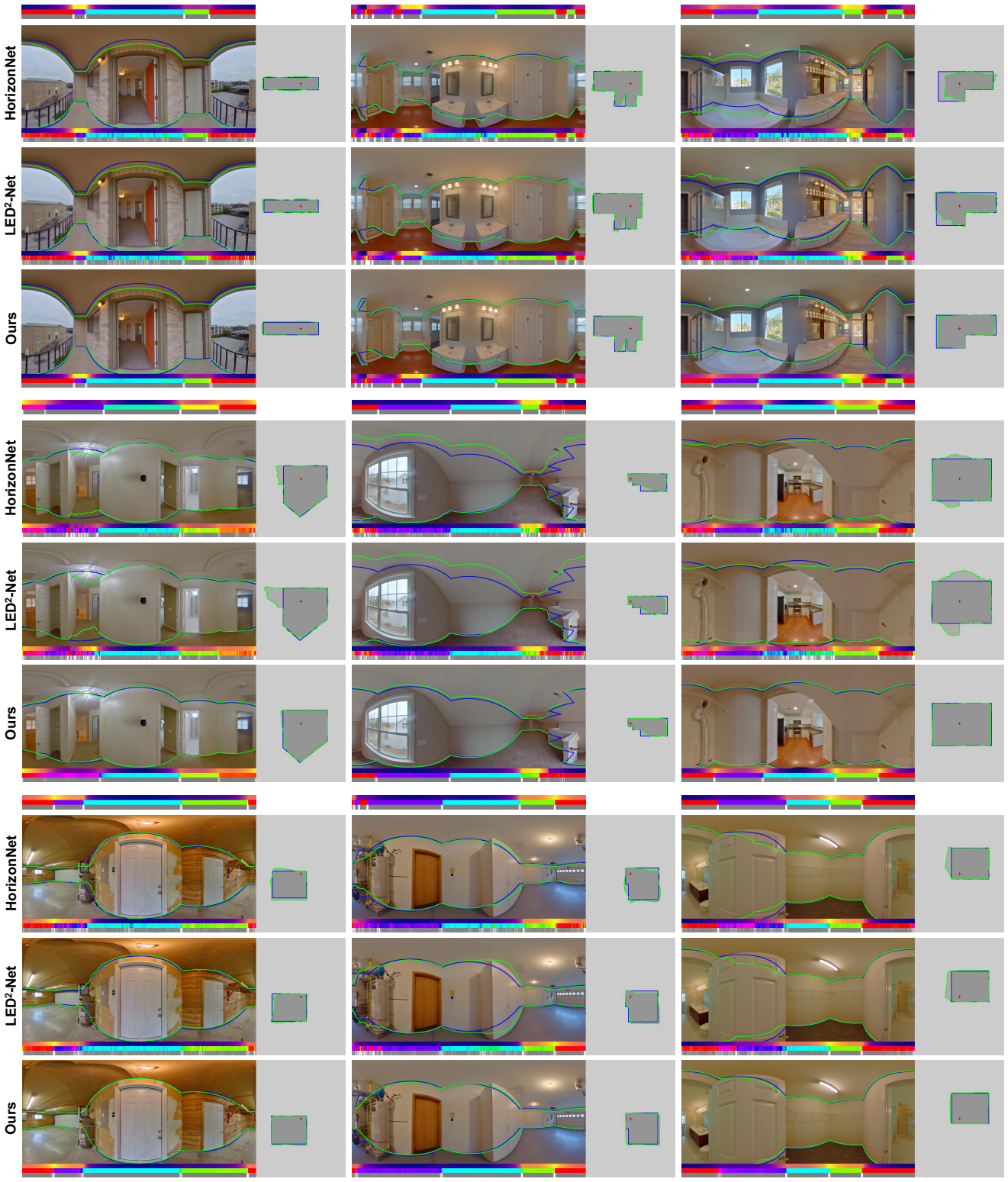


Figure 5. Qualitative comparison of general layout estimation on ZInd [3] dataset. We show the room layouts without post-processing by HorizonNet [6], LED^2-Net [7], and ours. We show the boundaries of the room layout on panorama (left) and the floor plan (right). The blue lines are ground truth, and the green lines are prediction. Moreover, we visualize the predicted horizon-depth, normal, and gradient below each panorama, and the ground truth is at the top. The dashed white lines highlight the errors generated by the baselines.

Figure 6. The 3D visualization results of our approach on ZInd [3] dataset. The green lines are predicted boundaries by our network, and the red lines are results with post-processing of the prediction. The room layouts are non-Manhattan at last row, we use the algorithm proposed by Douglas *et al*. [4] to simplify boundary directly.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1

[2] Tetsuo Asano. An efficient algorithm for finding the visibility polygon for a polygonal region with holes. *IEICE TRANSACTIONS (1976-1990)*, 68(9):557–559, 1985. 1

[3] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360° panoramas and 3d room layouts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2133–2143, 2021. 2, 5, 6

[4] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973. 6

[5] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. Atlantanet: Inferring the 3d indoor layout from a single 360° image beyond the manhattan world assumption. In *European Conference on Computer Vision*, pages 432–448. Springer, 2020. 2

[6] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019. 2, 3, 5

[7] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Led2-net: Monocular 360° layout estimation via differentiable depth rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12956–12965, 2021. 2, 3, 5

[8] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3363–3372, 2019. 2

[9] 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. 1

[10] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360° image: A comparative study of state-of-the-art methods. *International Journal of Computer Vision*, 129(5):1410–1431, 2021. 2, 3, 4