Learning Icosahedral Spherical Probability Map Based on Bingham Mixture Model for Vanishing Point Estimation (Supplementary Material)

Haoang Li[1,*] Kai Chen[1,*] Pyojin Kim[2] Kuk-Jin Yoon[3] Zhe Liu[4] Kyungdon Joo[5,†] Yun-Hui Liu[1,†]

[1]The Chinese University of Hong Kong, Hong Kong, China [2]Sookmyung Women's University, South Korea

[3]KAIST, South Korea [4]University of Cambridge, United Kingdom [5]UNIST, South Korea

Overview

In this supplementary material, we provide additional contents that are not included in the main paper due to the space limit:

- Details of the encoder of our network (see Section 1).
- Additional information of datasets and evaluation criteria (see Section 2).
- Additional comparisons with state-of-the-art methods (see Section 3).
- Details of a baseline method using the equi-angular discretization on the sphere (see Section 4).
- Additional tests of loss function (see Section 5).

1. Encoder of Our Network

As introduced in Section 3.2 of the main paper, we follow DCGAN [7] to design the encoder of our network. As shown in Fig. 1, we present the details. Our encoder works on the image domain. The height and width of an input image are 480 and 640 pixels, respectively. We first use a series of convolutions to extract features. The stride of convolution is 2, and the size of convolution kernel is 5×5. Each convolution is followed by bias adding, batch normalization, and leaky ReLU function. Then we reshape the feature map from 15×20 to 1×300 pixels. After that, we multiply this map by a 300×320 matrix, obtaining a 1×320 code. Note that the entries of this matrix are network parameters to optimize. Finally, we reshape the code from 1×320 to 20×4^2 pixels, and treat the result as the input spherical map of our decoder.

*Haoang Li and Kai Chen contributed equally to this work.
†Kyungdon Joo and Yun-Hui Liu are corresponding authors.

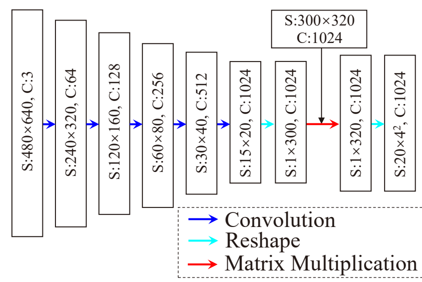


Figure 1. Our encoder works on the image domain. "S" and "C" denote the image size and the number of channels, respectively. We conduct matrix multiplication on each channel independently.

(a) (b) (c)

Figure 2. Representative images that satisfy (a) Manhattan world, (b) Atlanta world and (c) MMF, respectively.

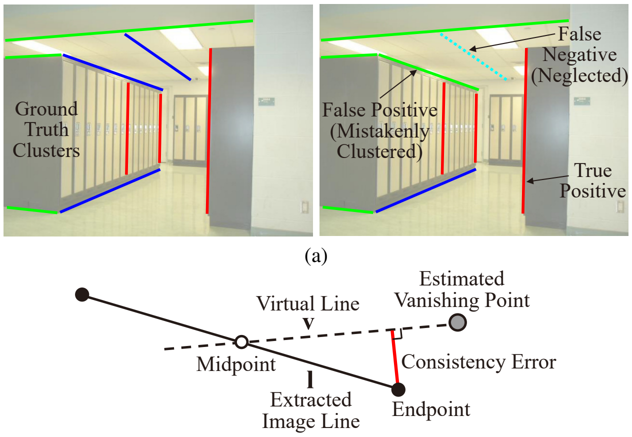
Table 1. Structure models used to express the scenes of datasets.

	Manhattan World [1]	Atlanta World [8]	MMF [10]
YUD+ [2]	√	√	√
VSD [5]	-	√	-
NYU-VP [9]	√	√	√
SU3 [13]	√	-	√

2. Dataset and Evaluation Criteria

Datasets. As shown in Fig. 2, three well-known structure models (i.e., Manhattan world [1], Atlanta world [8], and mixture of Manhattan frames (MMF) [10]) hold for various man-made environments. Table 1 summaries the models used to express the scenes of SU3 [13], YUD+ [2], VSD [5], and NYU-VP [9] datasets.

Precision, Recall, and F_1-score. As shown in Fig. 3(a),



(b)

Figure 3. (a) Illustration of true positive, false positive, and false negative to compute the precision and recall of image line clustering. (b) Illustration of consistency error.

we follow [5, 6] to define the true positive, false positive, and false negative of image line clustering. We compute the precision, recall, and F_1-score by

$$\begin{aligned} \text{precision} &= \frac{N(\text{true positive})}{N(\text{true positive}) + N(\text{false positive})}; \\ \text{recall} &= \frac{N(\text{true positive})}{N(\text{true positive}) + N(\text{false negative})}; \\ F_1\text{-score} &= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \end{aligned} \tag{1}$$

where $N(\cdot)$ denotes the cardinality.

Consistency Error. As shown in Fig. 3(b), an estimated vanishing point and the midpoint of an image line $\mathbf{l}$ associated with this vanishing point define a virtual line $\mathbf{v}$. Consistency error represents the distance from an endpoint of the line $\mathbf{l}$ to the virtual line $\mathbf{v}$. Given a set of image lines, we follow [11, 12] compute the root mean square of consistency errors.

3. Comparisons with State-of-the-art Methods

As shown in Fig. 4, we present additional comparisons with state-of-the-art methods. Overall, these results are similar to the results reported in Fig. 7 and Table 1 of the main paper. Specifically, the generality of TR-L-3 is low since this method assumes Manhattan world. TR-L-auto leads to low efficiency due to high-dimensional search space and highly non-linear cost function. Moreover, it cannot handle sloping DDs, which results in unsatisfactory generality. DL-nL-3 provides low generality due to the assumption of three DDs. Moreover, it is efficient but inaccurate due to its coarse-to-fine sampling strategy on the sphere. The accuracy of DL-L-auto is unsatisfactory since the sampled image lines may be affected by noise. In contrast, our DL-nL-auto simultaneously achieves high generality, satisfactory accuracy, and high efficiency. Therefore, it is more practical than the other methods.

Table 2. Comparison between various combinations of MSE, AS, and L_0 losses on all the datasets. We report the mean.

Coefficients		Cons. Error	F_1-score
Fitting	Regularization		
$\lambda_{\mathrm{MSE}} = 1$	$\lambda_{\mathrm{AS}} = 0.25$, $\lambda_1 = 0.05$	2.298 pix.	87.01%
	$\lambda_{\mathrm{AS}} = 0.25$, $\lambda_1 = 0.15$	2.243 pix.	87.23%
	$\lambda_{\mathrm{AS}} = 0.75$, $\lambda_1 = 0.05$	2.125 pix.	87.36%
	$\lambda_{\mathrm{AS}} = 0.75$, $\lambda_1 = 0.15$	2.084 pix.	87.94%
$\lambda_{\mathrm{MSE}} = 2$	$\lambda_{\mathrm{AS}} = 0.25$, $\lambda_1 = 0.05$	1.793 pix.	89.58%
	$\lambda_{\mathrm{AS}} = 0.25$, $\lambda_1 = 0.15$	1.752 pix.	89.67%
	$\lambda_{\mathrm{AS}} = 0.75$, $\lambda_1 = 0.05$	1.683 pix.	90.31%
	$\lambda_{\mathrm{AS}} = 0.75$, $\lambda_1 = 0.15$	1.697 pix.	90.18%
$\lambda_{\mathrm{MSE}} = 3$	$\lambda_{\mathrm{AS}} = 0.25$, $\lambda_1 = 0.05$	1.916 pix.	88.54%
	$\lambda_{\mathrm{AS}} = 0.25$, $\lambda_1 = 0.15$	1.895 pix.	88.79%
	$\lambda_{\mathrm{AS}} = 0.75$, $\lambda_1 = 0.05$	1.796 pix.	89.33%
	$\lambda_{\mathrm{AS}} = 0.75$, $\lambda_1 = 0.15$	1.801 pix.	89.15%
$\lambda_{\mathrm{MSE}} = 2$, $\lambda_{\mathrm{AS}} = 0.5$, $\lambda_1 = 0.1$ (Our)		**1.644** pix.	**90.75%**

4. Baseline Using Equi-angular Discretization

As introduced in Section 6.2 of the main paper, we design a baseline method using the equi-angular discretization on the sphere. As shown in Figs. 5(a) and 5(b), the input of baseline is the same as the input of our network. For the output spherical map based on equi-angular discretization of baseline, we set its resolution as $104 \times 208 = 21,632$ pixels[1]. This resolution is similar to the resolution of icosahedral spherical representation, i.e., $20 \times 4^5 = 20,480$ pixels, which contributes to a fair comparison. The reason why the second dimension (i.e., 208) is two times the first dimension (i.e., 104) is that the range $[-\pi, \pi]$ of longitude is two times the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ of latitude. In the following, we present details.

As shown in Fig. 5(a), except for the image sizes in the last two layers, the encoder of baseline is the same as the encoder of our network. Recall that the encoder of our network outputs a code whose length is $20 \times 4^2 = 320$. In contrast, the encoder of baseline outputs a code whose length is $13 \times 26 = 338$. These lengths are of the same magnitude, which contributes to a fair comparison (similar to the aforementioned resolution of the spherical map).

As shown in Fig. 5(b), overall, the decoder of baseline is similar to the decoder of our network. We summarize only three differences as follows. First, image sizes in corresponding layers are slightly different. For example, $26 \times 52 = 1352$ pixels of baseline correspond to $20 \times 4^3 = 1280$ pixels of our network. Second, the decoder

[1]The resolution $100 \times 200 = 20,000$ pixels mentioned in the main paper refers to the rough magnitude but not exact value.

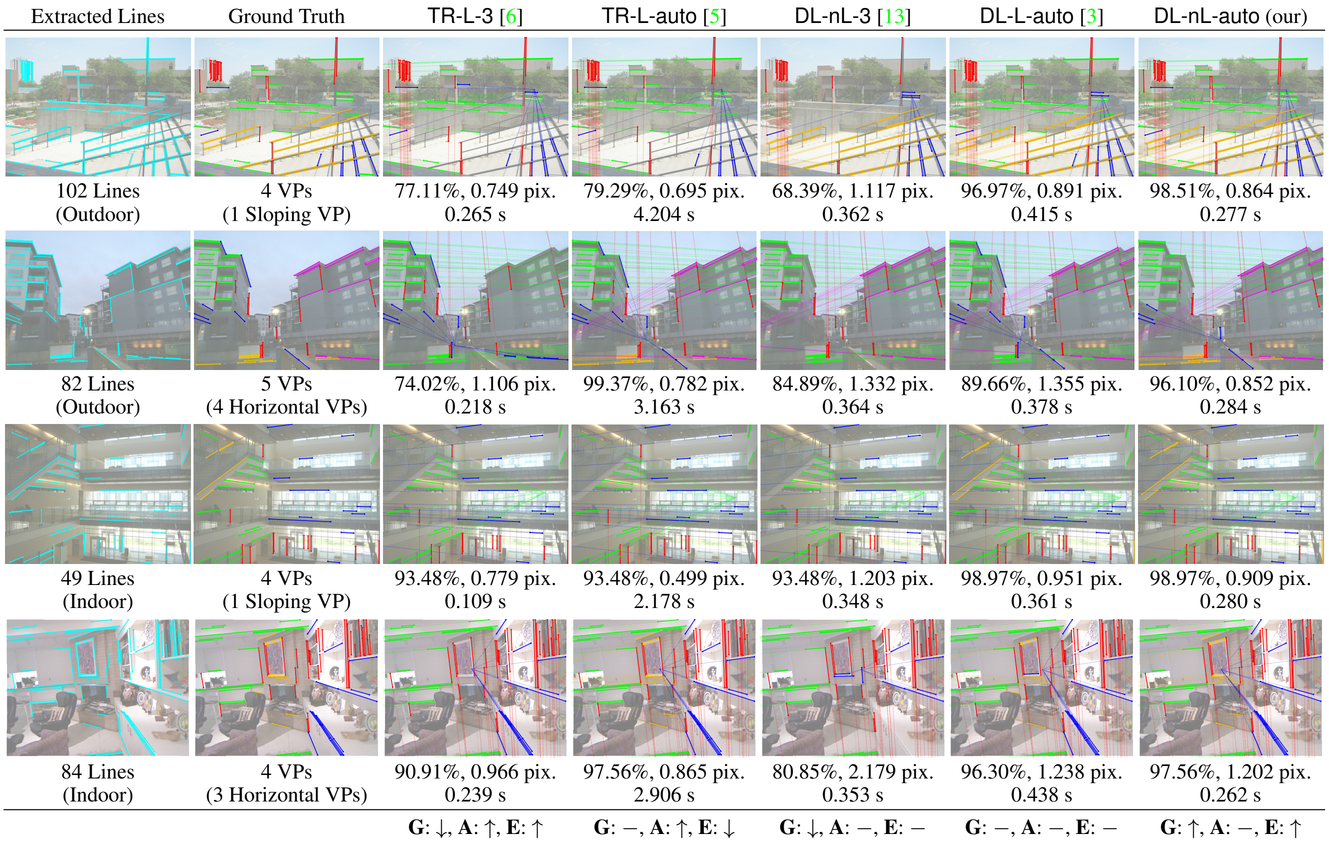


Figure 4. Additional generality ("**G**"), accuracy ("**A**") and efficiency ("**E**") comparisons on four representative images. "↑", "−" and "↓" represent high, middle and low, respectively. We use image lines to compute F_1-score and consistency error, regardless of whether a method requires image lines for vanishing point (VP) estimation. In the 3-rd to 7-th columns, a dotted line in the image represents the connection between the midpoint of a clustered image line and an estimated VP. A triplet of numbers below an image represents F_1-score, consistency error, and run time.

of baseline follows [4] to use traditional 2D image convolution for the equi-angular discretization on the sphere. The stride of convolution is 1, and the size of convolution kernel is 5×5. In contrast, the decoder of our network uses the spherical convolution (see Fig. 3(b) of the main paper). Third, as shown in Fig. 5(c), the up-sampling used by baseline is different from the spherical up-sampling used by our network (see Fig. 3(c) of the main paper). Despite differences, both strategies pad *three* neighbors of a pixel with zero for a fair comparison.

5. Tests of Loss Function

As introduced in Section 3.3 of the main paper, we empirically set the coefficients of MSE, AS, and L_0 sub-losses as 2, 0.5, and 0.1, respectively. In the following, we conduct tests by varying these coefficients. As shown in Table 2, we vary the coefficient λ_{MSE} of MSE loss from 1 to 3. We vary the coefficient λ_{AS} of AS loss from 0.25 to 0.75. We vary the coefficient λ_1 of L_0 loss from 0.05 to 0.15. Since the coefficient combination $\{\lambda_{\text{MSE}} = 2, \lambda_{\text{AS}} = 0.5, \lambda_1 = 0.1\}$ leads to the highest accuracy, we treat it as our fine-tuned combination.

References

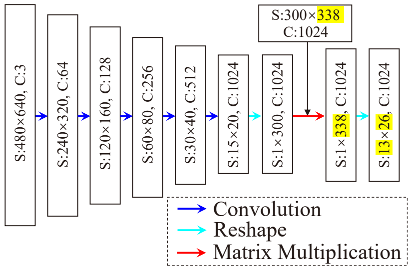
[1] James Coughlan and Alan Yuille. Manhattan world: Compass direction from a single image by Bayesian inference. In *ICCV*, 1999. 1

[2] Patrick Denis, James Elder, and Francisco Estrada. Efficient edge-based methods for estimating Manhattan frames in urban imagery. In *ECCV*, 2008. 1

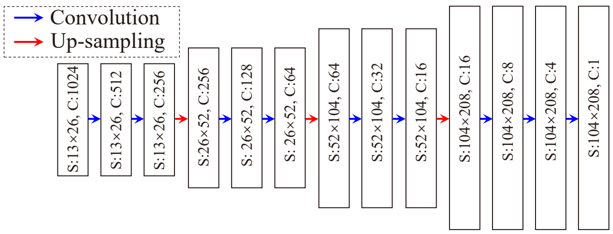
[3] Florian Kluger, Eric Brachmann, Hanno Ackermann, Carsten Rother, Michael Ying Yang, and Bodo Rosenhahn. CONSAC: Robust multi-model fitting by conditional sample consensus. In *CVPR*, 2020. 3

[4] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. SpherePHD: Applying CNNs on 360° images with non-Euclidean spherical polyhedron representation. *TPAMI*, 2020. 3

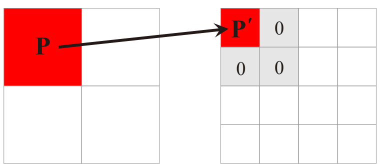
[5] Haoang Li, Pyojin Kim, Ji Zhao, Kyungdon Joo, Zhipeng Cai, Zhe Liu, and Yun-Hui Liu. Globally optimal and efficient vanishing point estimation in Atlanta world. In *ECCV*, 2020. 1, 2, 3



(a) Encoder on Image Domain



(b) Decoder on Sphere Domain



(c) Up-sampling

Figure 5. Baseline method using the equi-angular discretization on the sphere. (a) The encoder of baseline is analogous to the encoder of our network in Fig. 1. Their differences in image sizes are highlighted in yellow. (b) The decoder of baseline is analogous to the decoder of our network in Fig. 3(a) of the main paper. (c) The up-sampling of baseline is analogous to the up-sampling of our network in Fig.3(c) of the main paper. We transfer a pixel $\mathbf{p}$ in the lower-resolution map to a pixel $\mathbf{p}'$ in the higher-resolution map, and then pad three gray neighbors of $\mathbf{p}'$ with 0.

[6] Haoang Li, Ji Zhao, Jean-Charles Bazin, and Yun-Hui Liu. Quasi-globally optimal and near/true real-time vanishing point estimation in Manhattan world. *TPAMI*, 2020. 2, 3

[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1

[8] G. Schindler and F. Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *CVPR*, 2004. 1

[9] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 1

[10] Julian Straub, Oren Freifeld, Guy Rosman, John Leonard, and John Fisher. The Manhattan frame model—Manhattan world inference in the space of surface normals. *TPAMI*, 2018. 1

[11] Jean-Philippe Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *ICCV*, 2009. 2

[12] Lilian Zhang, Huimin Lu, Xiaoping Hu, and Reinhard Koch. Vanishing point estimation and line classification in a Manhattan world with a unifying camera model. *IJCV*, 2016. 2

[13] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. NeurVPS: Neural vanishing point scanning via conic convolution. In *NeurIPS*, 2019. 1, 3