Neural Geometric Parser for Single Image Camera Calibration – Supplementary Material –

Jinwoo Lee[1], Minhyuk Sung[2], Hyunjoon Lee[3], and Junho Kim[1]

[1]Kookmin University [2]Adobe Research [3]Intel

S.1 Comparisons on SUN360 [15] Dataset

Similar to the experiment in Sec. 4.1 in the paper, we also compare our method with the baseline methods using SUN360 [15] dataset. We selected indoor and outdoor scenes in the SUN360 dataset that satisfied the Manhattan/Atlanta assumptions, and generated the training and test images in the same way as with Google Street View dataset, described in Sec. 4 in the paper. 30,837 and 878 images are generated for the training and test sets, respectively. Details of the evaluation metrics and baseline methods are provided in Sec. 4.1 in the paper.

The quantitative results with SUN360 dataset [15] are reported in Table S1. The trends of the results are similar to those of the Google Street View experiment in Table 2 in the paper. Our method provides the best performance for most of the evaluation metrics, and the second-best for the remaining evaluation metrics, such as the median roll error and mean FoV error. Our method has a very marginal difference with the best AUC. The qualitative results are presented in Fig. S1.

Table S1. Quantitative evaluations with SUN360 dataset. Bold represents the best result, while an underscore represents the second-best result. Note that for DeepHorizon [13]*, we use the GT FoV to calculate the camera up vector (angle, pitch, and roll errors) from the predicted horizon line. In addition, for UprightNet [14]**, we use a pretrained model on ScanNet [4] due to the lack of required supervision in the SUN360 dataset.

Method	Angle (°) ↓		Pitch (°) ↓		Roll (°) ↓		FoV (°) ↓		AUC (%) ↑
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	
Upright [9]	3.43	<u>1.43</u>	3.03	<u>1.13</u>	6.85	**0.47**	8.62	<u>3.21</u>	79.16
A-Contrario [11]	5.77	1.53	4.91	1.19	6.93	0.66	-	-	72.75
DeepHorizon [13]*	2.87	2.12	2.36	1.64	1.16	0.85	-	-	<u>80.65</u>
Perceptual [8]	<u>2.54</u>	1.93	<u>2.11</u>	1.49	<u>1.06</u>	0.77	**5.29**	3.93	**80.85**
Perceptual [8] +**L**	2.86	2.17	2.45	1.76	<u>1.06</u>	0.75	6.29	4.37	78.38
UprightNet [14]**	34.72	34.67	35.31	33.72	4.92	2.88	-	-	-
Ours	**2.33**	**1.27**	**1.97**	**0.96**	**0.97**	<u>0.51</u>	<u>5.66</u>	**3.16**	80.07

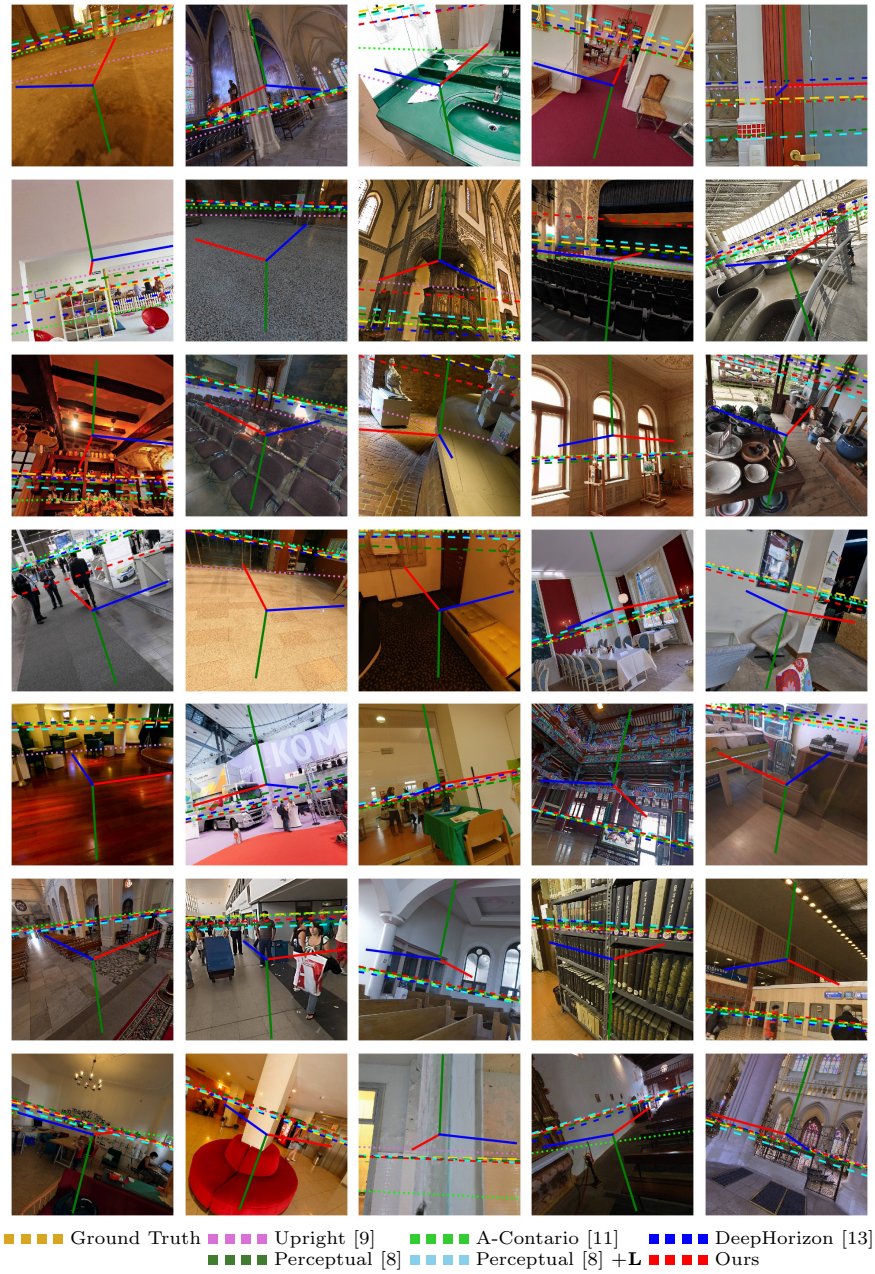

Fig. S1. Examples of horizon line prediction on the SUN360 test set. Each example also displays the Manhattan direction of the highest score candidate.

S.2 Additional Results on Google Street View [1] Dataset

Fig. S2 presents additional results on the Google Street View [1] dataset, as in Fig. 5 in the paper, visualizing horizon line predictions and weakly supervised

Manhattan directions. In each example, we illustrate the Manhattan directions of the highest score candidate (Eq. (22) in the paper). In most cases, our method provides better horizon prediction results than those of previous state-of-the-art methods. Note that we only use the supervision of horizon lines and focal lengths (3DoF), yet we can further estimate the camera rotation and focal lengths (4DoF) based on the Manhattan world assumption.

S.3 Visualization of Our Network I/Os

Fig. S3 illustrates how geometric cues are processed and utilized in the proposed method. Each row of Fig. S3(a)-(d) shows the input image, rasterized line segment map $\mathbf{L}$, grouped horizon line segments used for sampling candidates of Manhattan directions (as in Fig. 3(d) in the paper), and the set of sampled candidates of Manhattan directions, respectively.

Fig. S3(e) displays the prediction results of the horizons and their corresponding ground truths, as well as the Manhattan direction of the highest score candidate. Activation maps $\mathbf{A}$ with respect to the Manhattan directions are presented in Fig. S3(f). Notice that activation maps $\mathbf{A}$ in Fig. S3(f) explain much of their respective line segment maps $\mathbf{L}$ in Fig. S3(b), exemplifying how our method incorporates with the Manhattan world assumption.

Fig. S3(g) superimposes the eight Manhattan directions of the top-8 high-scoring candidates over the input images. As illustrated in Fig. S3(g), the zenith directions are almost the same between candidates, as the man-made scenes usually satisfies either the Manhattan or Atlanta world assumption [3, 10]. For scenes satisfying the Manhattan world assumption (rows 1–4), the axes of eight frames almost overlap. For the last two scenes (rows 5 and 6) that follow the Atlanta world assumption, all the frames have zenith directions that are very close to each other. By utilizing these frames we can robustly and accurately estimate horizon lines and focal lengths of given scenes.

S.4 Comparison of Manhattan Direction Prediction on YUD [5] and ECD [12]

We report the accuracy of Manhattan direction prediction using our method on YUD [5] and ECD [12] datasets and compare the result with those of the other methods. In the experiment, we took the network model trained on the Google Street View dataset and tested it on YUD [5] and ECD [12] datasets. For the evaluation, the Manhattan direction of the high score candidate is used.

YUD [5] dataset contains 102 images under the Manhattan assumption, where each image is annotated with three VPs and a focal length. ECD [12] dataset contains 103 images under the Atlanta assumption, where each image is annotated with a zenith VP and more than two horizontal VPs on a horizon line. For ECD [12] dataset, the direction which is the closest to the prediction is used for comparisons.

Table S2 shows the quantitative comparisons, in terms of the relative rotation angle, differences of FoV, and AUC. For FoV and AUC, we used the same setting

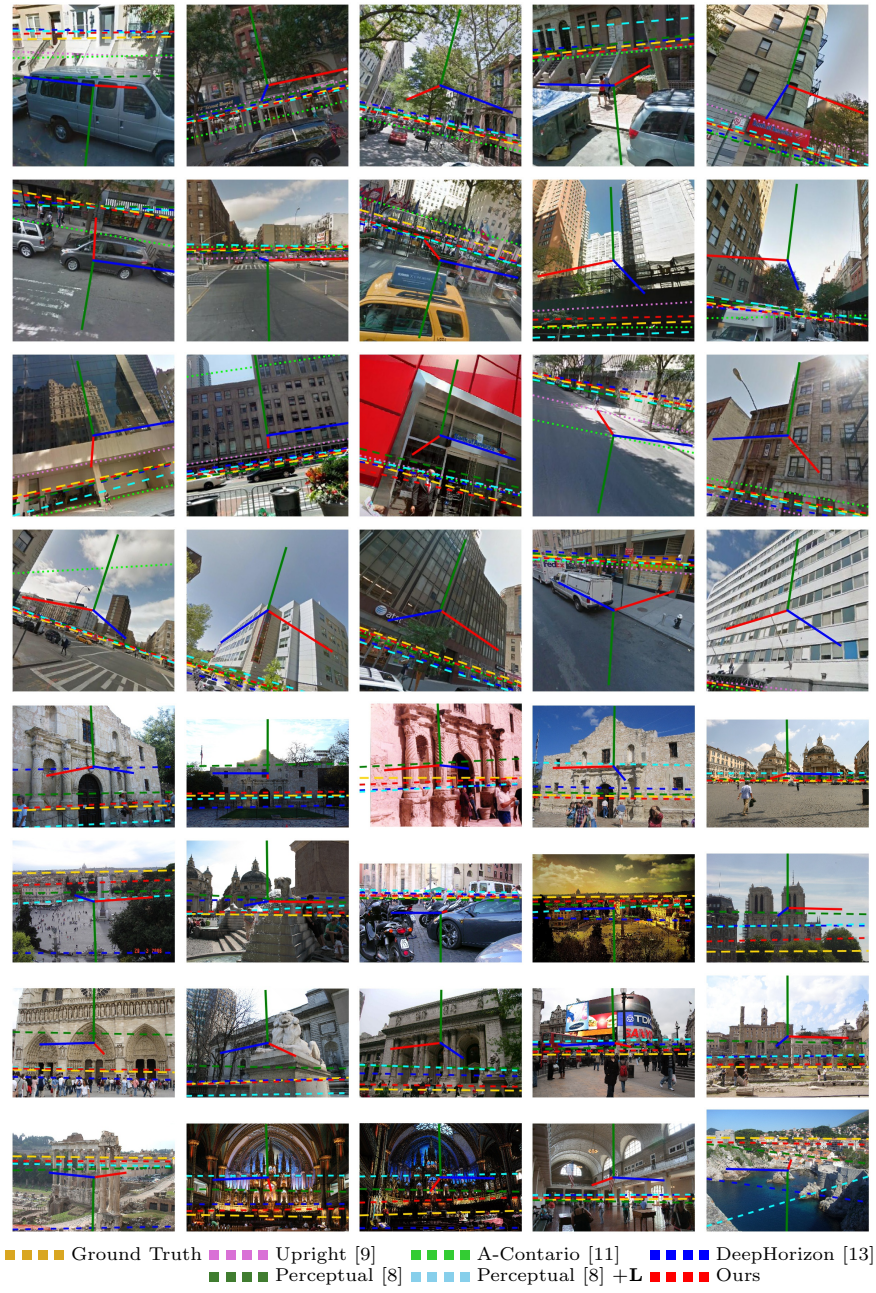

Fig. S2. Examples of horizon line prediction on the Google Street View test set (top four rows) and on the HLW test set (bottom four rows). Each example also shows the Manhattan direction of the highest score candidate.

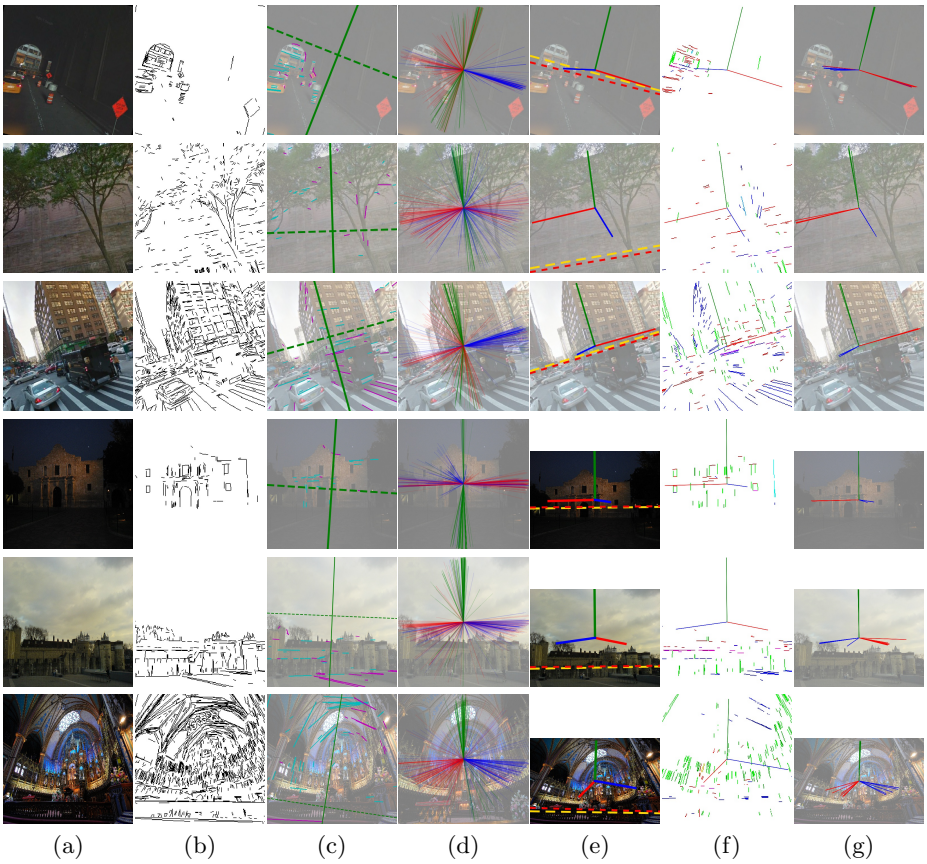

Fig. S3. Sampled images from the Google Street View test set (top three rows) and the HLW test set (bottom three rows). For each row, we show: (a) input image; (b) rasterized line segment map **L**; (c) two groups of horizontal line segments (cyan & magenta) used for sampling candidates of Manhattan directions; (d) sampled candidates of Manhattan directions; (e) ground truth and predicted horizon lines (yellow & red dashed) as well as the estimated Manhattan directions of the highest score candidate; (f) activation map **A** of the Manhattan directions shown in (e); and (g) Manhattan directions of the top-8 high score candidates.

as depicted in Sec. 4.1 in the paper. As shown Table S2, our results are comparable to the ones of non-neural-net methods [9, 11], which are highly optimized for YUD [5] and ECD [12] datasets. We remark that our networks are trained on a different dataset and also with weak and indirect supervision (horizon lines and focal lengths).

Table S2. Quantitative evaluations of Manhattan direction prediction with YUD [5] and ECD [12] datasets.

Dataset	Method	Rotation Angle (°) ↓		FoV (°) ↓		AUC (%) ↑
		Mean	Med.	Mean	Med.	
YUD [5]	Upright [9]	**0.46**	**0.27**	3.41	1.18	87.26
	A-Contrario [11]	1.03	0.78	3.42	**1.17**	**95.35**
	Ours	0.52	0.33	**2.73**	1.47	83.21
ECD [12]	Upright [9]	**2.91**	**1.02**	**12.44**	**6.31**	76.71
	A-Contrario [11]	3.11	1.44	16.73	10.22	**91.10**
	Ours	3.14	1.34	13.61	7.73	77.61

S.5 Parameter Sensitivity Test

We tested parameter sensitivity by varying our parameters including: the angle threshold for vertical lines (δ_z in Eq. (4)), the angle thresholds for deciding the positive and negative samples of zenith candidates (δ_p and δ_n in Eq. (8)), the score threshold for ZSNet (δ_c in Eq. (12)), the score threshold for FSNet (δ_s in Eq. (19)), and the numbers of line segments and intersection points used in the network ($|L_z|$ and $|Z|$). Also, we tested sensitivity to line detection algorithm by varying the LSD algorithm parameter, $-\log(\text{NFA})$, where NFA is the number of false alarms and also by replacing the line detection algorithm with MCMLSD [2]. We used the network model trained with the Google Street View [1] dataset with *default* parameters and tested the model by varying the parameters, except for δ_p and δ_n in Eq. (8) and δ_s in Eq. (19); these parameters change either the ground truth labels or the loss function. For those parameters, we finetuned our network from the pretrained model. All results are reported in Table S3. The highlighted rows show the results with default parameters. The results demonstrate that our method is robust to the change of the parameters.

In our implementation, 1,024 for both lines and points was the maximum number to train the network with 11 GB GPU memory. However, more numbers of lines and points also significantly increase training time and GPU memory usage. For the sake of simplicity, all results reported in this paper were obtained with $|L_z| = |Z| = 256$ both at training time and test time.

S.6 Visualization of FSNet Focus

In Fig. S4, we show more visualizations of the weights of the second last convolution layer in FSNet, as shown in Fig. 6 in the paper. The network mostly focuses on the lines that pass the vanishing points.

S.7 Failure Cases

Fig. S5 shows failure cases of our framework. The failure cases occur when the computation of the focal length is unstable, such that: i) the scene is far from the Manhattan assumption, ii) only short or noisy line segments are detected in the scene, iii) the scene is almost perpendicular to the center of projection.

Table S3. Parameter sensitivity test results. The highlighted rows show the result with the default parameters. Bold is the best result, and underscore is the second-best result in each experiment.

		Angle (°) ↓		Pitch (°) ↓		Roll (°) ↓		FoV (°) ↓		AUC
		Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	(%) ↑
δ_z (Eq. (4))	58.5°	**2.04**	1.67	**1.84**	1.44	**0.64**	**0.46**	**5.67**	**3.52**	83.01
	67.5°	2.12	**1.61**	1.92	**1.38**	0.75	0.47	6.01	3.72	**83.12**
	76.5°	2.83	1.97	2.30	1.67	1.62	0.57	6.47	3.97	79.70
δ_p, δ_n (Eq. (8))	1°, 2°	3.45	1.98	2.73	1.87	1.53	0.62	7.43	4.02	75.22
	2°, 5°	**2.12**	**1.61**	**1.92**	**1.38**	**0.75**	**0.47**	**6.01**	**3.72**	**83.12**
	5°, 10°	2.54	1.97	2.10	1.71	**0.75**	0.57	6.64	4.21	79.01
δ_c (Eq. (12))	0.4	2.32	1.84	2.28	1.44	0.84	0.52	6.82	4.37	80.23
	0.5	**2.12**	**1.61**	**1.92**	**1.38**	0.75	0.47	6.01	3.72	**83.12**
	0.6	2.17	1.71	1.96	1.48	**0.65**	**0.46**	**5.76**	**3.42**	82.94
δ_s (Eq. (19))	0.4	3.02	1.80	2.71	1.61	1.04	**0.47**	6.70	4.05	80.82
	0.5	**2.12**	**1.61**	**1.92**	**1.38**	0.75	**0.47**	6.01	3.72	**83.12**
	0.6	2.19	1.64	1.93	1.43	**0.74**	**0.47**	**5.88**	**3.43**	83.10
top-k	$k = 1$	2.23	1.72	1.97	1.49	0.75	0.55	6.71	3.84	82.12
	$k = 4$	**2.10**	1.70	**1.89**	1.48	**0.65**	0.49	6.01	**3.66**	83.05
	$k = 8$	2.12	**1.61**	1.92	**1.38**	0.75	0.47	6.01	3.72	**83.12**
	$k = 16$	2.24	1.71	2.04	1.52	**0.65**	**0.46**	**5.61**	**3.66**	82.70
$\|L_z\|$, $\|Z\|$	256, 256	2.12	**1.61**	1.92	**1.38**	0.75	0.47	6.01	3.72	83.12
	1024, 1024	**2.05**	1.65	**1.86**	1.46	**0.63**	**0.45**	**5.66**	**3.45**	**83.80**
$-\log$ (NFA) in LSD [7], MCMLSD [2]	0	2.12	**1.61**	2.09	**1.38**	0.80	0.47	6.15	3.72	83.12
	0.01×1.75^0	2.12	1.74	**1.91**	1.54	**0.65**	0.47	6.02	3.77	82.38
	0.01×1.75^5	**2.11**	1.72	**1.91**	1.51	**0.65**	0.48	6.07	3.90	**83.36**
	0.01×1.75^{10}	2.19	1.75	1.95	1.55	0.71	0.49	6.25	3.65	82.97
	0.01×1.75^{15}	2.17	1.70	1.95	1.46	0.67	**0.46**	**5.53**	3.16	83.34
	MCMLSD [2]	2.31	1.65	2.02	1.46	0.81	0.50	5.85	**3.01**	83.05

Please notice that the estimated zenith directions are still reasonable in Fig. S5, thanks to the semantic information learned by ResNet, the backbone of our FSNet. Therefore, even in the cases of Fig. S5, our framework is still applicable to image rotation corrections as shown in Fig. 1(a).

S.8 Experiment on KITTI [6] Dataset

We conducted an additional experiment with KITTI [6] dataset. The KITTI dataset contains wide-images captured by driving around urban cities and rural areas. We sample 8,675 images of urban scenes from the KITTI dataset and feed them to finetune our network from the pretrained model with the Google Street View dataset. We test our finetuned model to 481 images of urban and rural scenes from the KITTI dataset.

Fig. S6 shows some examples of horizon predictions with the KITTI test set. Unfortunately, the GT horizons of the dataset are geometrically inaccurate due to the large influence of the vehicle's tilting angle during cornering. Nevertheless, we obtained interesting results where the estimated horizons of our framework do not deviate significantly from the GT horizons in urban areas. We believe the results come from the KITTI dataset, since there are little changes in the horizontal line and focal length. Another reason seems to be that the ResNet,

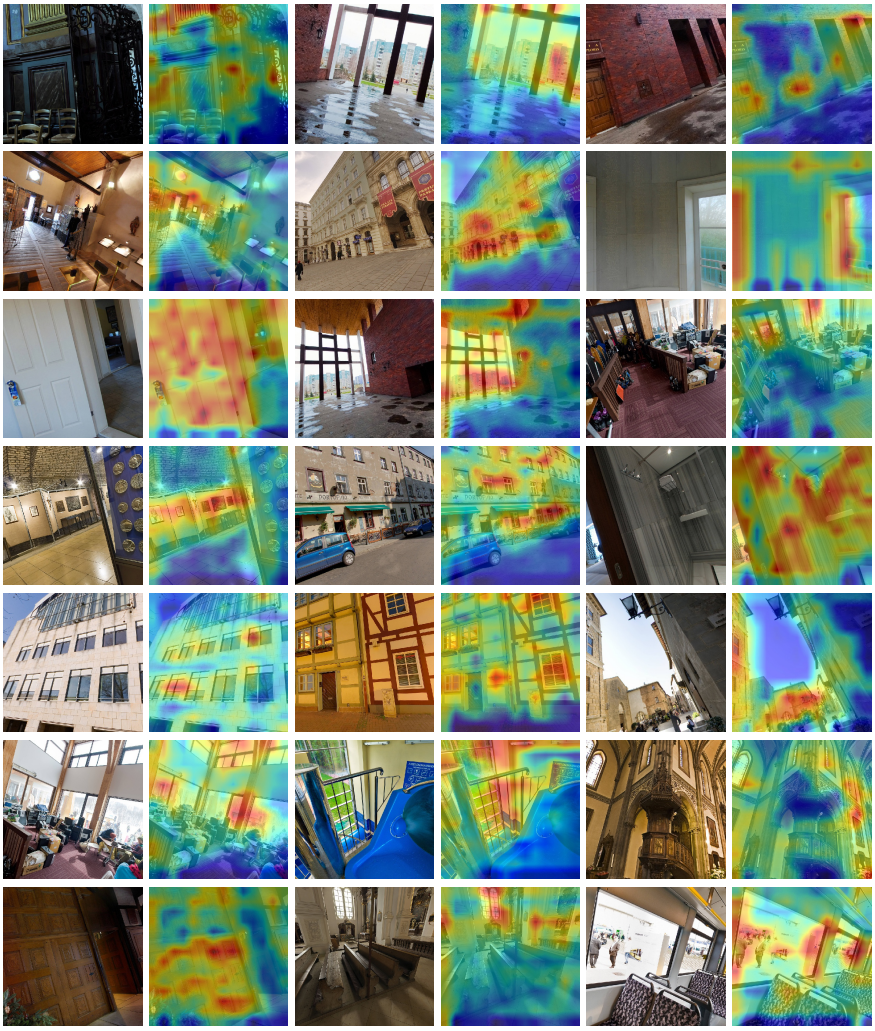

Fig. S4. More visualizations of FSNet focus: Left is input; right is feature highlight.

Table S4. Quantitative evaluations with KITTI dataset.

Method	Angle (°) ↓		Pitch (°) ↓		Roll (°) ↓		FoV (°) ↓		AUC (%) ↑
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	
Ours ($k = 8$)	3.38	3.06	2.50	2.12	1.87	1.88	17.68	16.49	78.34

the backbone of our FSNet, learned the scene context from the KITTI dataset. Table S4 reports the quantitative evaluations with KITTI dataset.

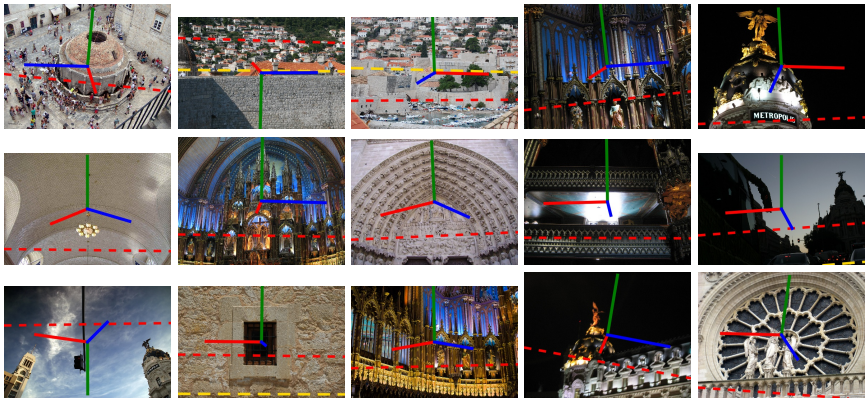

Fig. S5. Failure cases.

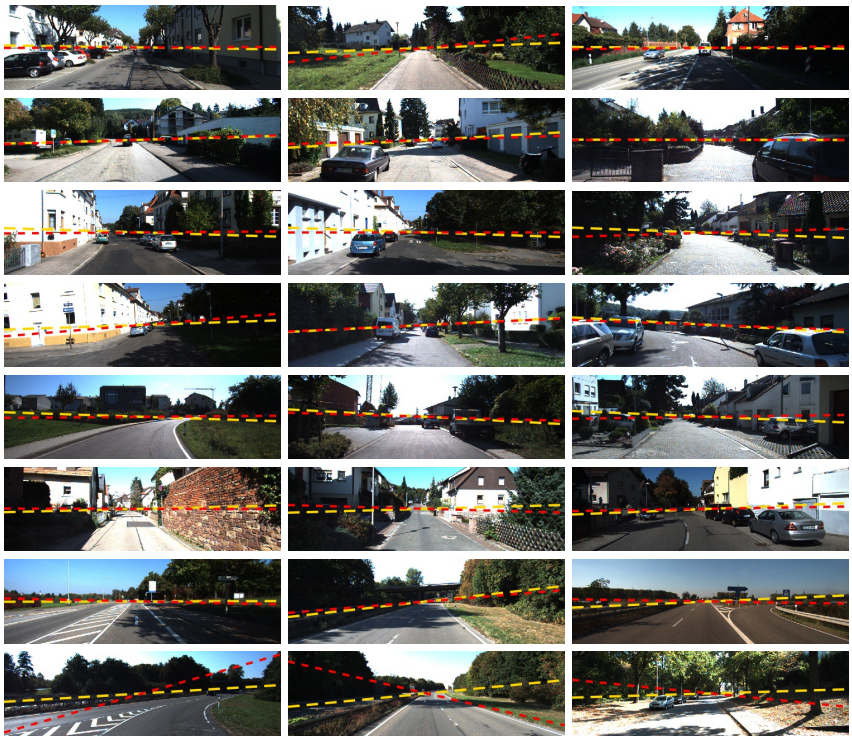

Fig. S6. Examples of horizon line prediction on the KITTI test set.

References

1. Google Street View Images API. https://developers.google.com/maps/
2. Almazàn, E.J., Tal, R., Qian, Y., Elder, J.H.: MCMLSD: A Dynamic Programming Approach to Line Segment Detection. In: Proc. CVPR. pp. 2031–2039 (2017)
3. Coughlan, J.M., Yuille, A.L.: Manhattan World: Compass Direction from a Single Image by Bayesian Inference. In: Proc. ICCV. pp. 941–947 (1999)
4. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In: Proc. CVPR. pp. 5828–5839 (2017)
5. Denis, P., Elder, J.H., Estrada, F.J.: Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery. In: Proc. ECCV. pp. 197–210 (2008)
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proc. CVPR. pp. 3354–3361 (2012)
7. von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: A Fast Line Segment Detector with a False Detection Control. IEEE Trans. Pattern Analysis Machine Intelligence **32**(4), 722–732 (2010)
8. Hold-Geoffroy, Y., Sunkavalli, K., Eisenmann, J., Fisher, M., Gambaretto, E., Hadap, S., Lalonde, J.F.: A Perceptual Measure for Deep Single Image Camera Calibration. In: Proc. CVPR. pp. 2354–2363 (2018)
9. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic Upright Adjustment of Photographs with Robust Camera Calibration. IEEE Trans. Pattern Analysis Machine Intelligence **36**(5), 833–844 (2014)
10. Schindler, G., Dellaert, F.: Atlanta World: An Expectation Maximization Framework for Simultaneous Low-level Edge Grouping and Camera Calibration in Complex Man-made Environments. In: Proc. CVPR (2004)
11. Simon, G., Fond, A., Berger, M.O.: A-Contrario Horizon-First Vanishing Point Detection Using Second-Order Grouping Laws. In: Proc. ECCV. pp. 318–333 (2018)
12. Tretyak, E., Barinova, O., Kohli, P., Lempitsky, V.: Geometric Image Parsing in Man-Made Environments. International Journal of Computer Vision **97**(3), 305–321 (2012)
13. Workman, S., Zhai, M., Jacobs, N.: Horizon Lines in the Wild. In: Proc. BMVC. pp. 20.1–20.12 (2016)
14. Xian, W., Li, Z., Fisher, M., Eisenmann, J., Shechtman, E., Snavely, N.: UprightNet: Geometry-Aware Camera Orientation Estimation From Single Images. In: Proc. ICCV. pp. 9974–9983 (2019)
15. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing Scene Viewpoint using Panoramic Place Representation. In: Proc. CVPR. pp. 2695–2702 (2012)