A Overview

This document provides additional analysis and extra experiments to the main paper. Specifically, in Sec. B, we analyse the latency of proposed method and compare it with some pseudo-LiDAR based methods. Sec. C shows the performance of stereo images while Sec. D gives the results of Pedestrian and Cyclist detection. Finally, Sec. E presents more visualization examples.

B Runtime Analysis

In this section, we will analyze the latency of our PatchNet and compare it with some existing methods [24,35,36] based on pseudo-LiDAR representation. In general, all the four methods can be divided into three main stages. In ours designs, the processing flows of PatchNet-vanilla and pseudo-LiDAR are the same, but the representations of inputs are different. So the runtime of these two methods are almost the same, which is shown as follow (tested on a single 1080 GPU):

Table 8. Runtime of PachNet-vanilla and pseudo-LiDAR.

2D detection	Depth estimation	3D detection
60ms	400ms	28ms

PatchNet shares the same 2D detector and depth estimator (note the runtime of different depth estimators varies greatly, see **KITTI Benchmark** for details), and we show its runtime of 3D detection stage for different backbone models as follows:

Table 9. Runtime of PatchNet in 3D detection stage.

Backbone	PointNet-18	ResNet-18	ResNeXt-18	SE-ResNet-18
runtime	12ms	23ms	18ms	26ms

Although we add some extra operations in PatchNet, the runtime of the baseline model (PointNet-18) is 12ms while the runtime of PatchNet-vanilla is 28ms. This is mainly because we remove the foreground segmentation net and use a dynamic threshold to segment the foreground, which can save about 18ms. For the best backbone, the runtime is only 26ms, which has similar runtime of pseudo-LiDAR for 3d detection.

Besides, although PatchNet and [24] use the same segmentation method, [24] add another ResNet-34 to extract image features. For [36], it adds a 2D instance segmentation net, which will bring lots of computing overhead (e.g., about 200ms for Mask RCNN [14]).

In summary, PatchNet is more efficient than [24,36] and has the similar run time as [35].

C Stereo Images

Pseudo-LiDAR representation is also widely used in the field of stereo 3D detection task. In order to verify that the proposed method is still work with binocular images, we replace the monocular depth maps with the stereo ones (we use PSMNet [5] as our stereo depth estimator and get the pre-trained model from [35]) and test the performance on KITTI *validation* set using $AP|_{R_{11}}$ for better comparison with previous works. As shown in the Tab. 10 , PatchNet-vanilla has almost the same accuracy as pseudo-LiDAR, while PatchNet achieves better performances. We also report the $AP|_{R_{40}}$ for reference.

Table 10. Stereo 3D detection performance of the **Car** category on KITTI *validation* dataset. IoU threshold is set to 0.7. We highlight the best results in **bold**.

Method	3D Detection			BEV Detection		
	Easy	Moderate	Hard	Easy	Moderate	Hard
3DOP [7]	6.55	5.07	4.10	12.63	9.49	7.59
Multi-Fusion [38]	-	9.80	-	-	19.54	-
Stereo-RCNN [20]	54.1	36.7	31.1	68.5	48.3	41.5
Pseudo-LiDAR [35]	59.4	39.8	33.5	72.8	51.8	44.0
PatchNet-vanilla	60.8	40.1	33.6	72.7	51.2	43.8
PatchNet	**65.9**	**42.5**	**38.5**	**74.5**	**52.9**	**44.8**
PatchNet-vanilla@$AP\|_{R_{40}}$	61.4	37.6	31.6	73.5	49.8	41.7
PatchNet@$AP\|_{R_{40}}$	**66.0**	**41.1**	**34.6**	**76.8**	**52.8**	**44.3**

D Pedestrian and Cyclist

For better comparison, we also report **Pedestrian/Cyclist** detection performance for 3D detection task on KITTI *validation* set in this part. Specifically, we conduct these experiments using both monocular and stereo images with $AP|_{R_{11}}$ as metric. It can be seen from Tab. 11 that the proposed model also get better performance than [35] with each setting. Note that results of pseudo-LiDAR are evaluated by ourselves using its official code, since pseudo-LiDAR did not provide Pedestrian/Cyclist detection results for monocular images.

Besides, the accuracy of **Pedestrian/Cyclist** detection fluctuate greatly compared with **Car** detection. This fluctuation of performance is mainly caused by insufficient training samples (there are only 2,207/734 training samples for **Pedestrian/Cyclist** in KITTI *training* set, while it provides 14,357 **Car** instances). This problem can be reduced by introducing more training data or more effective data augmentation strategies.

Table 11. 3D detection performance of the **Pedestrian/Cyclist** category on KITTI *validation* dataset. Metric is $AP|_{R_{11}}$ and IoU threshold is set to 0.5. We highlight the best results in **bold**.

Method	Category	Monocular			Stereo		
		Easy	Moderate	Hard	Easy	Moderate	Hard
Pseudo-LiDAR [35]	Pedestrian	7.32	6.19	5.64	33.8	27.4	24.0
PatchNet	Pedestrian	**9.82**	**7.86**	**6.84**	**38.8**	**30.1**	**26.5**
Pseudo-LiDAR [35]	Cyclist	5.49	3.85	3.82	41.3	25.2	24.9
PatchNet	Cyclist	**8.14**	**4.84**	**4.62**	**46.8**	**29.0**	**26.8**

E More Qualitative Examples

In this part, we compare the monocular images and stereo pairs by some representitive qualitative results in Fig. 6. First, we can find that stereo images can detect objects more accurately, which is generally reflected to the better depth estimation, instead of size or heading estimation. Then, for most of close range objects, in terms of visual experience, monocular images are not inferior to stereo images (although there are still some failure case among those instances).

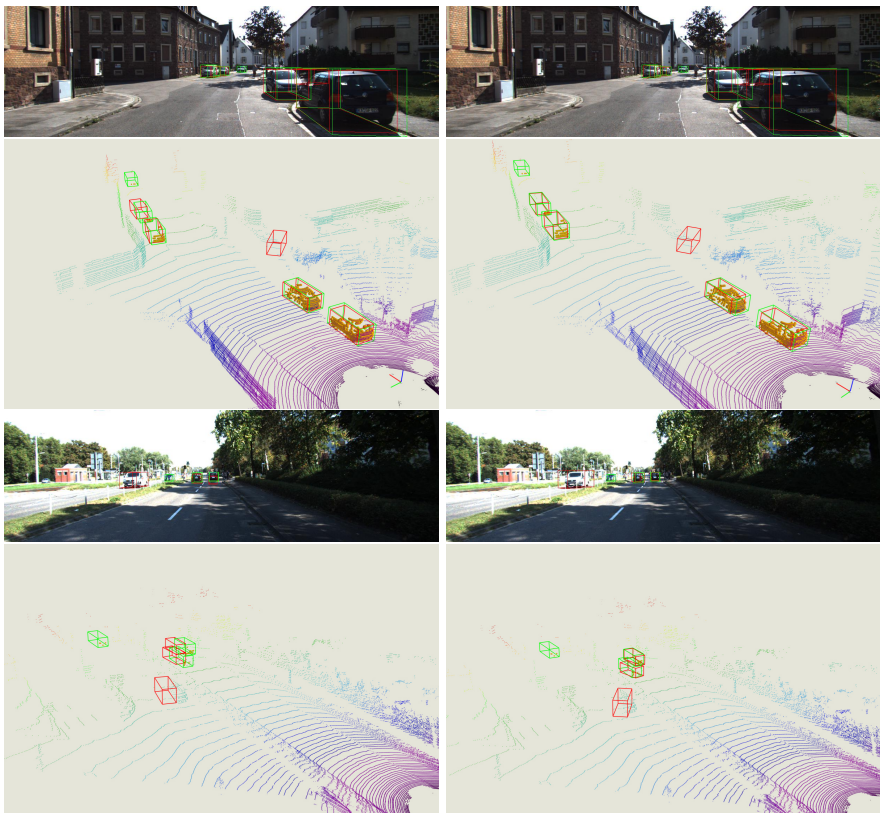

Fig. 6. Qualitative results on KITTI *validation* set. *Left*: monocular detection results. *Right*: stereo detection results.Red boxes represent our predictions, and green boxes come from ground truth. LiDAR signals are only used for visualization. Best viewed in color with zoom in.