Supplemental Material

Anonymous AAAI submision
Paper ID 3014

In this supplemental material, we provide additional technical details, extra analysis experiments to the main paper.

Network Architecture

We choose $VGG16$ (Matthew and Rob 2014) as our CNN backbone. For the 2D detector, we mainly follow KittiBox (Teichmann et al. 2016). The 6 output channels consist of the objectiveness confidence (2 channels), the offsets of 2D bounding box center to the grid center (2 channels) and the size of the bounding box (2 channels). In the refinement stage, $RoiAlign$ is applied to the region of interest in early feature maps to regress the delta values.

The sub-network for the instance-level depth estimation, 3D localization and corner offsets regression shares the same buffer zone following the backbone network. Notice that there are two buffer zones, with one extending from $conv4\_3$ and another from $pool5$. Layers with 64 output channels are designed as bottlenecks to enforce the network to encode minimal sufficient information and prevent over-fitting, and also reduce the computational cost. Inverted residual connections (Sandler et al. 2018) are applied to neighboring bottlenecks to provide shortcuts for gradient propagation.

The branch (i.e., depth encoder) for instance-level depth estimation follows the architecture introduced in DORN (Fu et al. 2018) that stacks along the channel axis the outputs of a fully-connected global information encoder and 3 parallel dilated convolution layers (Yu and Koltun 2015) with different dilated rates, which for the early features are 6, 12, 24 and 2, 4, 8 for deep features. The branches for location estimation and corner regression both contain 4 convolution layers with 3×3 kernels and 1×1 stride. While there are 96 weighted layers in total, the deepest path i.e., from the input to IDE output, only contains 26 weighted layers, since the 3D reasoning branches are parallel. Detailed network configuration is shown in Table A

Results Visualization

In Fig. A, we compare our 3D detection results with 3DOP (Chen et al. 2015) and Mono3D (Chen et al. 2016) by visualizing in the 3D space and on the image. We also present the instance-level depth estimation outputs in Fig. B.

	Layer Description	Output Tensor Dim.
	Input image	$384 \times 1248 \times 3$
	Backbone	
13	VGG 16, conv 4	$48 \times 156 \times 512$
18	VGG 16, pool 5	$12 \times 39 \times 512$
	2D Detector	
19-20	1×1 conv	$12 \times 39 \times 128$
21-22	1×1 conv, softmax	$12 \times 39 \times 2$
23	from 20, 1×1 conv, bbox offsets	$12 \times 39 \times 4$
24	from 13, 7×7 RoiAlign	$12 \times 39 \times 7 \times 7 \times 512$
25	1×1 conv, reduce channels	$12 \times 39 \times 7 \times 7 \times 32$
26	fully connected, offset deltas	$12 \times 39 \times 4$
27	add layer 26 and 23, refined offsets	$12 \times 39 \times 4$
	Buffer Zone	
28	from 18, 3×3 conv	$12 \times 39 \times 64$
29	3×3 conv	$12 \times 39 \times 256$
30	3×3 conv, add with layer 28	$12 \times 39 \times 64$
31-34	(repeat layers 29 and 30) $\times 2$	$12 \times 39 \times 64$
35	3×3 conv	$12 \times 39 \times 128$
36	from 13, 3×3 conv	$48 \times 156 \times 64$
37-43	repeat layers 29-35, $4\times$ resolution	$48 \times 156 \times 256$
	Instance-level Depth Estimation	
44	3×3 conv	$48 \times 156 \times 64$
45	3×3 conv	$48 \times 156 \times 256$
46	3×3 conv, add with 44	$48 \times 156 \times 64$
47-48	repeat layer 45 and 46	$48 \times 156 \times 64$
49	3×3 conv	$48 \times 156 \times 128$
50	3×3 conv, stride 2	$24 \times 78 \times 64$
51	2×2 max pooling, stride 2	$12 \times 39 \times 64$
52	fully connected	64
53	copy to every pixel	$48 \times 156 \times 64$
54	from 49, 3×3 conv, $6\times$atrous	$48 \times 156 \times 64$
55	from 49, 3×3 conv, $12\times$atrous	$48 \times 156 \times 64$
56	from 49, 3×3 conv, $24\times$atrous	$48 \times 156 \times 64$
57	stack 53, 54, 55 and 56 along channels	$48 \times 156 \times 256$
58	1×1 conv	$48 \times 156 \times 128$
59	from 35, 3×3 conv	$12 \times 39 \times 64$
60-75	repeat layers 45-58, $1/4$ resolution	$12 \times 39 \times 128$
76	1×1 conv, coarse instance depth	$12 \times 39 \times 1$
77	from 58, 7×7 RoiAlign	$12 \times 39 \times 7 \times 7 \times 128$
78	fully connected, delta instance depth	$12 \times 39 \times 1$
79	add 76 and 78, refined instance depth	$12 \times 39 \times 1$
	3D Center Localization	
80	from 43, 3×3 conv	$48 \times 156 \times 64$
81-85	repeat layers 45-49	$48 \times 156 \times 128$
86	from 35, 3×3 conv	$12 \times 39 \times 64$
87-91	repeat layers 45-49, $1/4$ resolution	$12 \times 39 \times 128$
92	1×1 conv, projected 3D center	$12 \times 39 \times 2$
93	extension, outputs 3D location	$12 \times 39 \times 3$
94	from 85, 7×7 RoiAlign	$12 \times 39 \times 7 \times 7 \times 128$
95	fully connected, delta 3D location	$12 \times 39 \times 3$
96	add layer 93 and 95, refined location	$12 \times 39 \times 3$
	Corner Offsets Regression	
97	from 43, 3×3 conv	$48 \times 156 \times 64$
98-102	repeat layers 45-49	$48 \times 156 \times 128$
103	7×7 RoiAlign	$12 \times 39 \times 7 \times 7 \times 128$
104	fully connected, 3×8 offsets	$12 \times 39 \times 24$

Table A: Network configuration.

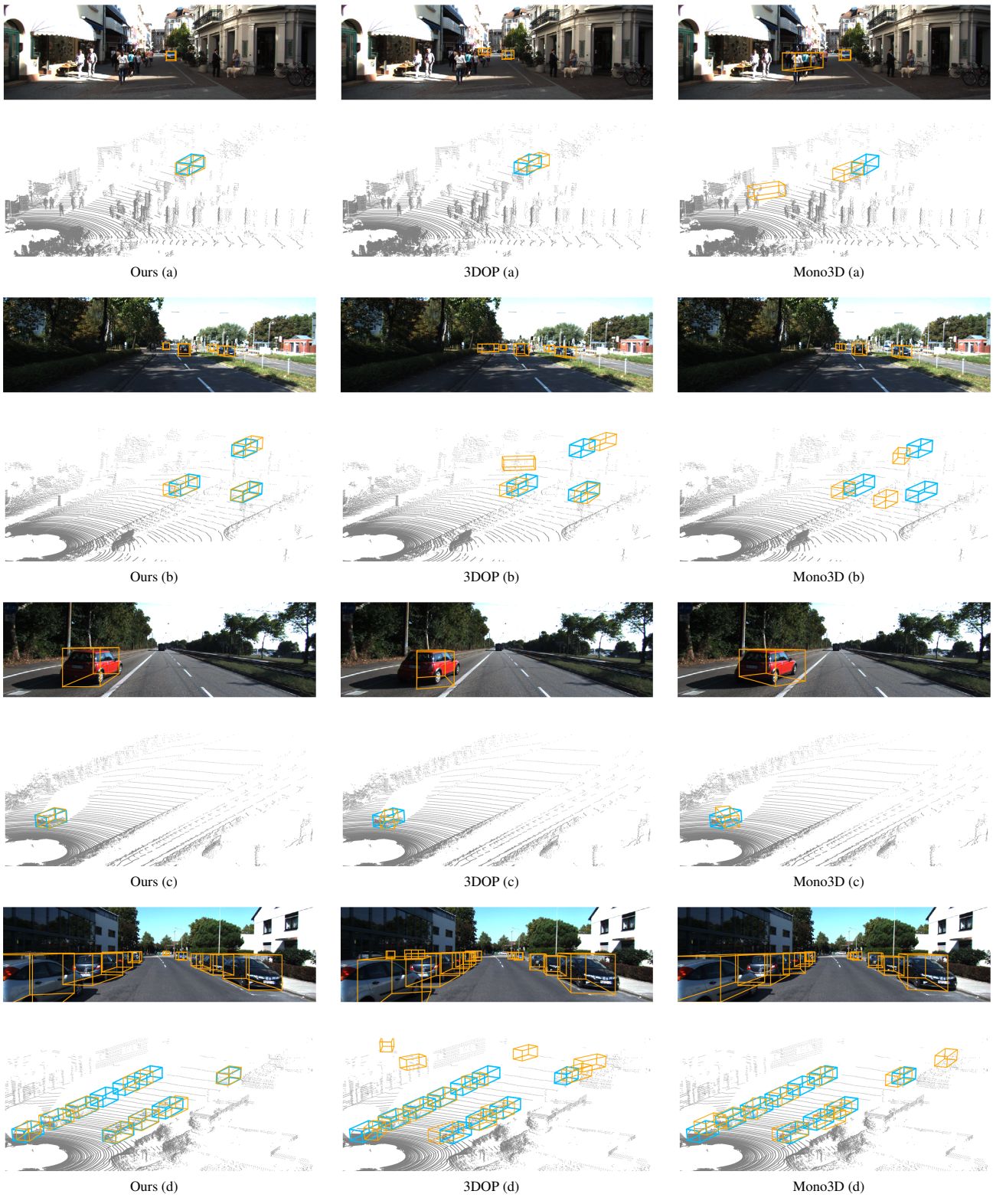

Figure A: **Qualitative comparison.** Blue boxes indicate ground truths and orange ones are predictions. It can be seen from (a) that our method is the most stable when dealing with far objects. In corner-cases when the object is truncated by the image boundaries, i.e., in (d), our method can still localize the whole ABBox-3D.

Figure B: **Instance-level depth.** Each grid cell predicts the 3D centric depth of its nearest instance. Cells with objectiveness confidence (provided by the 2D detector) no less than 0.1 are kept for visualization

References

Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A. G.; Ma, H.; Fidler, S.; and Urtasun, R. 2015. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, 424–432.

Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; and Urtasun, R. 2016. Monocular 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2147–2156.

Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; and Tao, D. 2018. Deep ordinal regression network for monocular depth estimation. In *Computer Vision and Pattern Recognition (CVPR)*.

Matthew, D. Z., and Rob, F. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 818–833.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Computer Vision and Pattern Recognition (CVPR)*, 4510–4520.

Teichmann, M.; Weber, M.; Zoellner, M.; Cipolla, R.; and Urtasun, R. 2016. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*.

Yu, F., and Koltun, V. 2015. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*.