Supplementary Material: MonoDGP: Monocular 3D Object Detection with Decoupled-Query and Geometry-Error Priors

Fanqi Pu[1] Yifan Wang[1] Jiru Deng[1] Wenming Yang[1*]
[1]Shenzhen International Graduate School, Tsinghua University
{pfq23, yf-wang23, djr23}@mails.tsinghua.edu.cn, yang.wenming@sz.tsinghua.edu.cn

A. Detailed Discussion on Depth Error

In our method, we regard the distance between the camera plane and the car's closest wheel point as the geometric depth. However, this assumption is appropriate when the camera has the same height of the object. The height inconsistency will lead to the bias l_{bias} between the actual geometric depth Z_{geo} and the wheel depth Z_w. We set the height ratio γ of the camera height H_{cam} to the object height H as follows:

$$\gamma = \frac{H_{cam}}{H} \tag{1}$$

We will discuss how the height ratio affects the distribution of the depth error Z_{err}.

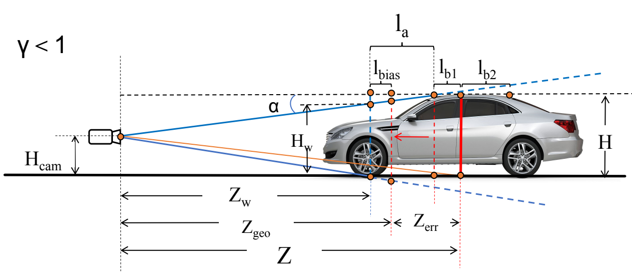


Figure 1. The perspective transformation when the camera height is lower than the object height.

The vehicle is treated as a trapezoid. The closest wheel locates in the lowest position of the 2D bounding box, while the highest position in the object surface will change with the height ratio. As shown in Fig. 1, when $\gamma < 1$, the wheel depth is shorter than the geometric depth, which can be expressed as:

$$Z_{geo} = Z_w + l_{bias} \tag{2}$$

To calculate the wheel bias, we first represent the height at the wheel point based on the similar triangle theory:

$$tan\alpha = \frac{H - H_{cam}}{Z_w + l_a} = \frac{H - H_w}{l_a} \tag{3}$$

$$H_w = H - \frac{(H - H_{cam}) \cdot l_a}{Z_w + l_a} \tag{4}$$

And then we utilize H_w to compute l_{bias} :

$$\frac{H}{Z_{geo}} = \frac{H_w}{Z_w} \tag{5}$$

$$l_{bias} = \frac{(H - H_{cam}) \cdot Z_w \cdot l_a}{H \cdot Z_w + H_{cam} \cdot l_a} = \frac{(1-\gamma) \cdot Z_w \cdot l_a}{Z_w + \gamma \cdot l_a} \tag{6}$$

We can express depth error as follows:

$$Z_{err} = l_{b1} + l_a - l_{bias} = l_{b1} + \sigma_1 \cdot l_a \tag{7}$$

$$\sigma_1 = \frac{\gamma}{1 - \frac{(1-\gamma) \cdot l_a}{Z_w + l_a}} \tag{8}$$

where $\gamma < \sigma_1 < 1$, $l_{bias} < (1-\gamma) \cdot l_a$. The original depth error, which should be perspective-invariant, is calculated by the formula $Z_{err} = l_{b1} + l_a$. Except for the vehicle's own attributes, γ and Z_w also affect the depth error. The closer σ_1 is to 1, the less effect it has. According to the Eq. (8), σ_1 reduces as Z_w increases and γ decreases.

To present the greatest impact of the height ratio, we take an extreme example based on the Fig. 2, and make σ_1 as smaller as possible. Specifically, we set $H_{cam} = 1.5m$, $H = 1.8m$, $\gamma = \frac{5}{6}$, $l_a = 1m$, $Z_w = 50m$. From the Eq. (6) and Eq. (8), we obtain $\sigma_1 \approx 0.84$ and $l_{bias} \approx 0.16m$. This extreme bias value is significantly lower than the whole depth value.

As shown in Fig. 3, when $\gamma > 1$, the wheel depth is larger than the geometric depth, which can be expressed as:

$$Z_{geo} = Z_w - l_{bias} \tag{9}$$

*Corresponding author.

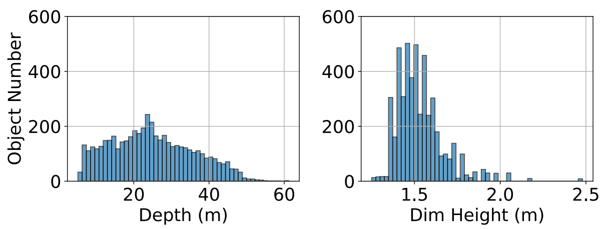


Figure 2. The data distribution of the object's central depth and dimension height on the KITTI training set.

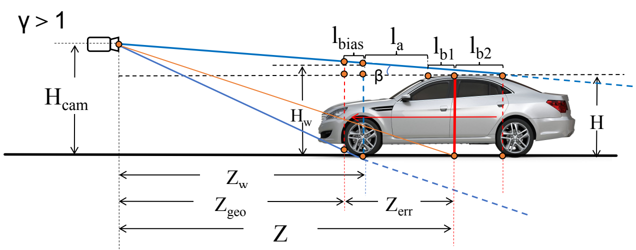


Figure 3. The perspective transformation when the camera height is higher than the object height.

We can achieve the wheel bias similar to the previous proof process:

$$tan\beta = \frac{H_{cam} - H}{Z_w + l_a + l_{b1} + l_{b2}} = \frac{H_w - H}{l_a + l_{b1} + l_{b2}} \tag{10}$$

$$H_w = \frac{H \cdot Z_w + H_{cam} \cdot (l_a + l_{b1} + l_{b2})}{Z_w + l_a + l_{b1} + l_{b2}} \tag{11}$$

$$l_{bias} = \frac{(\gamma - 1) \cdot Z_w \cdot (l_a + l_{b1} + l_{b2})}{Z_w + \gamma \cdot (l_a + l_{b1} + l_{b2})} \tag{12}$$

Homogeneously, we can express depth error as follows:

$$Z_{err} = l_{b1} + l_a + \sigma_2 \cdot (l_a + l_{b1} + l_{b2}) \tag{13}$$

$$\sigma_2 = \frac{(\gamma - 1)}{1 + \frac{\gamma \cdot (l_a + l_{b1} + l_{b2})}{Z_w}} \tag{14}$$

where $0 < \sigma_2 < \gamma - 1$, $l_{bias} < (\gamma - 1) \cdot (l_a + l_{b1} + l_{b2})$. The closer σ_2 is to 0, the less effect it has. According to the Eq. (14), σ_2 increases as Z_w and γ increase.

To show the height ratio's maximum impact, we also suppose an extreme case and make σ_2 as larger as possible. To be more specific, we set $H_{cam} = 1.5m$, $H = 1.25m$, $\gamma = \frac{6}{5}$, $l_a + l_{b1} + l_{b2} = 2m$, $Z_w = 50m$. Based on the Eq. (12) and Eq. (14), we achieve $\sigma_2 \approx 0.19$ and $l_{bias} \approx 0.38m$. This bias value is higher than the value calculated when $\gamma < 1$, but has a slight effect on the whole depth.

In most instances, the camera height is close to the vehicle height, which means $\gamma \approx 1$ and the depth error is roughly perspective-invariant for the car category. Even if the object height is obviously different from the camera height, the network can directly learn and predict this tiny bias compared with the whole depth. The error prediction is still a simple and effective method to replace the multi-depth prediction.

B. Discussion on Geometric Constraints

Previous works like Deep3DBox [12] and Shift R-CNN [13] enforce strict geometric constraints by tightly fitting projections of the 3D bounding box into the 2D box. While recent methods such as MonoGR2 [1] and GUP-Net [9] formulate constraints based on geometric similarity, where under vehicle-mounted camera perspectives and fixed focal length, the object's center depth can be uniquely determined through the proportional relationship between its 3D height and 2D projected height.

Projection-alignment constraints exhibit quadratic errors from 2D boundary localization inaccuracies, while height-ratio constraints demonstrate linear errors confined to height predictions. The former fails with truncated objects requiring full 2D contours, whereas the latter maintains functionality under partial occlusions through visible height segments. Height-ratio constraints surpass projection-alignment methods in stability (linear vs. quadratic errors), efficiency (closed-form vs. iterative), and robustness (partial vs. full contours), establishing them as core geometric priors for monocular 3D detection. Future frameworks could incorporate projection-alignment constraints as auxiliary regularizers within joint optimization.

C. Detailed Loss Function

The 2D loss L_{2D} adopts focal loss [7] to estimate the object categories, L1 loss to estimate the projected center (x_{3d}, y_{3d}) and 2D sizes (l, r, t, b), and GIoU loss for the bounding box. We can formulate the 2D object loss as:

$$L_{2D} = \lambda_1 L_{cls} + \lambda_2 L_{2dsize} + \lambda_3 L_{xy} + \lambda_4 L_{giou} \tag{15}$$

The 3D loss follows MonoDLE [11] to predict 3D sizes (h_{3d}, w_{3d}, l_{3d}) and orientation angle α. As for the depth prediction, an uncertainty regression loss based on the Laplacian distribution is defined as:

$$L_{depth} = \frac{\sqrt{2}}{\sigma_d} \left\| \frac{f \cdot H}{h_{bbox}} + Z_{err} - Z_{gt} \right\|_1 + \log(\sigma_d) \tag{16}$$

where σ_d is the standard deviation of the distribution.

We can formulate 3D object loss as:

$$L_{3D} = \lambda_5 L_{3dsize} + \lambda_6 L_{angle} + \lambda_7 L_{depth} \tag{17}$$

Methods	Val, IoU=0.5, $AP_{3D\|R40}$					
	Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDGP (Ours)	**13.77**	**10.06**	**7.96**	**12.21**	**6.61**	**5.95**
w/o Segment Embeddings	13.02	9.67	7.66	10.56	5.22	4.68
w/o RSH	12.50	9.42	7.34	9.16	4.34	4.18
w/o Depth Error	9.90	7.55	6.09	11.13	5.86	5.51

Table 1. Ablation study of the pedestrian and cyclist categories on the KITTI val set.

Methods	Extra data	Test, IoU=0.5, $AP_{3D\|R40}$					
		Pedestrian			Cyclist		
		Easy	Mod.	Hard	Easy	Mod.	Hard
CaDDN [15]	LiDAR	12.87	8.14	6.76	<u>7.00</u>	3.41	<u>3.30</u>
OccupancyM3D [14]		14.68	9.15	7.80	**7.37**	<u>3.56</u>	2.84
MonoPGC [19]	Depth	14.16	<u>9.67</u>	<u>8.26</u>	5.88	3.30	<u>2.85</u>
GUPNet [9]	None	<u>14.72</u>	9.53	7.87	4.18	2.65	2.09
MonoCon [8]		13.10	8.41	6.94	2.80	1.92	1.55
DEVIANT [5]		13.43	8.65	7.69	5.05	3.13	2.59
MonoDDE [6]		11.13	7.32	6.67	5.94	**3.78**	**3.33**
MonoDETR [21]		12.65	7.19	6.72	5.12	2.74	2.02
MonoDGP (Ours)	None	**15.04**	**9.89**	**8.38**	5.28	2.82	2.65

Table 2. Comparisons of the pedestrian and cyclist categories on the KITTI test set. We **bold** the best results and <u>underline</u> the second-best results.

The depth map loss L_{dmap} utilizes focal loss to predict categorical foreground depth map. More detailed information about L_{dmap} can be found in MonoDETR [21].

D. Experiments on Other Categories

Since segment embeddings are mainly trained to distinguish between background and target, they can easily handle multiple classes without modification. Ablation studies of other categories are shown in Tab. 1. In particular, error prediction significantly improves 3D pedestrian prediction compared to cars and cyclists. This can be explained that pedestrians have consistent depth errors across orientations, whereas cyclists have irregular shapes and greater geometric uncertainty. These spatial uncertainties may degrade the effectiveness of the projection transformation.

We also compare the pedestrian and cyclist detection results in Tab. 2. Specifically, our method achieves a superior performance on all levels of difficulty for pedestrian detection, benefiting from its simple and stable geometric structures. However, the performance for the cyclist category falls short of the best.

Notably, despite these geometric challenges, our cyclist detection performance remains competitive among methods without extra training data. This underscores the generalizability of RSH module for complex categories.

E. Sensitivity to Initial Features

Since error prediction mode heavily relies on good geometric features, the inaccuracies of initial features can significantly impact the convergence and performance of the proposed network. The Initial features, such as 3D dimension height (H_{3D}) and 2D bounding box height (h_{bbox}), are crucial for geometric depth calculation. To analyze their individual impacts, we conduct sensitivity experiments replacing predicted H_{3D} and h_{bbox} with ground truth values.

Geometric Depth ($f \times H_{3D}/h_{bbox}$)		Val, IoU=0.7, $AP_{3D\|R40}$		
Ground Truth H_{3D}	Ground Truth h_{bbox}	Easy	Mod.	Hard
×	×	30.76	22.34	19.01
✓	×	39.10	31.89	27.51
×	✓	33.62	25.04	21.97
✓	✓	**57.81**	**48.55**	**41.92**

Table 3. Sensitivity study on the KITTI val set for the car category.

As shown in Tab. 3, the perfectly accurate geometric depth improves moderate AP_{3D} by up to 26.21%, highlighting the significance of these features. Compared to h_{bbox}, the network is more sensitive to H_{3D} errors due to its inherent difficulty as a 3D property. There also exists a coupling relationship between h_{bbox} and H_{3D}. Simultaneously replacing both features with ground truth values performs much better than replacing them individually. Current limitations mainly arise from height prediction error accumulation in the perspective projection. Improvements in monocular features, particularly for H_{3D}, will further enhance the performance of error prediction in the future.

F. Initial Depth from Pre-trained MDE

Depth Prediction Mode	Val, IoU=0.7, $AP_{3D\|R40}$		
	Easy	Mod.	Hard
Direct Depth	24.33	18.87	15.31
DAv2-small (HS) + Depth Error	11.45	9.11	7.86
DAv2-small (VK2) + Depth Error	27.04	20.48	17.52
DAv2-base (VK2) + Depth Error	**27.86**	**21.15**	**18.10**

Table 4. Ablation Study of pre-trained MDE. 'DAv2' denotes Depth Anything V2 [20] method, 'HS' denotes pre-trained on indoor dataset Hypersim [16], 'VK2' denotes pre-trianed on outdoor dataset Virtual KITTI 2 [3].

Monocular depth estimation (MDE) models have developed for many years. We can also utilize the pre-trained MDE to provide a roughly approximate surface depth, similar to geometric depth, which may render the learning problem even simpler.

To explore this possibility, we exploit Depth Anything V2 [20] to generate depth maps. Based on the initial metric depth, error prediction can achieve better performance compared to direct prediction in Tab. 1. However, MDE heavily relies on pre-trained datasets, while geometric depth relies on its own attributes without additional parameters. This will limit the generalization of achieving initial depth from pre-trained MDE.

Difficulty	Methods	Extra	AP_{3D} All	AP_{3D} 0-30	AP_{3D} 30-50	AP_{3D} 50-∞	APH_{3D} All	APH_{3D} 0-30	APH_{3D} 30-50	APH_{3D} 50-∞
Level_1(IoU=0.7)	CaDDN [15]	LiDAR	5.03	15.54	1.47	0.10	4.99	14.43	1.45	0.10
	PatchNet [10] in [18]	Depth	0.39	1.67	0.13	0.03	0.39	1.63	0.12	0.03
	PCT [18]	Depth	0.89	3.18	0.27	0.07	0.88	3.15	0.27	0.07
	M3D-RPN [2] in [15]	None	0.35	1.12	0.18	0.02	0.34	1.10	0.18	0.02
	GUPNet [9] in [5]	None	2.28	6.15	0.81	0.03	2.27	6.11	0.80	0.03
	DEVIANT [5]	None	2.69	6.95	<u>0.99</u>	0.02	2.67	6.90	<u>0.98</u>	0.02
	MonoUNI [4]	None	<u>3.20</u>	<u>8.61</u>	0.87	<u>0.13</u>	<u>3.16</u>	<u>8.50</u>	0.86	<u>0.12</u>
	MonoDGP (Ours)	None	**4.28**	**10.24**	**1.15**	**0.16**	**4.23**	**10.10**	**1.14**	**0.16**
Level_2(IoU=0.7)	CaDDN [15]	LiDAR	4.49	14.50	1.42	0.09	4.45	14.38	1.41	0.09
	PatchNet [10] in [18]	Depth	0.38	1.67	0.13	0.03	0.36	1.63	0.11	0.03
	PCT [18]	Depth	0.66	3.18	0.27	0.07	0.66	3.15	0.26	0.07
	M3D-RPN [2] in [15]	None	0.35	1.12	0.18	0.02	0.33	1.10	0.17	0.02
	GUPNet [9] in [5]	None	2.14	6.13	0.78	0.02	2.12	6.08	0.77	0.02
	DEVIANT [5]	None	2.52	6.93	<u>0.95</u>	0.02	2.50	6.87	<u>0.94</u>	0.02
	MonoUNI [4]	None	<u>3.04</u>	<u>8.59</u>	0.85	<u>0.12</u>	<u>3.00</u>	<u>8.48</u>	0.84	<u>0.12</u>
	MonoDGP (Ours)	None	**4.00**	**10.20**	**1.13**	**0.15**	**3.96**	**10.08**	**1.12**	**0.15**
Level_1(IoU=0.5)	CaDDN [15]	LiDAR	17.54	45.00	9.24	0.64	17.31	44.46	9.11	0.62
	PatchNet [10] in [18]	Depth	2.92	10.03	1.09	0.23	2.74	9.75	0.96	0.18
	PCT [18]	Depth	4.20	14.70	1.78	0.39	4.15	14.54	1.75	0.39
	M3D-RPN [2] in [15]	None	3.79	11.14	2.16	0.26	3.63	10.70	2.09	0.21
	GUPNet [9] in [5]	None	10.02	24.78	4.84	0.22	9.94	24.59	4.78	0.22
	DEVIANT [5]	None	<u>10.98</u>	<u>26.85</u>	<u>5.13</u>	0.18	<u>10.89</u>	<u>26.64</u>	<u>5.08</u>	0.18
	MonoUNI [4]	None	<u>10.98</u>	26.63	4.04	<u>0.57</u>	10.73	26.30	3.98	<u>0.55</u>
	MonoDGP (Ours)	None	**12.36**	**31.12**	**5.78**	**1.24**	**12.18**	**30.68**	**5.71**	**1.22**
Level_2(IoU=0.5)	CaDDN [15]	LiDAR	16.51	44.87	8.99	0.58	16.28	44.33	8.86	0.55
	PatchNet [10] in [18]	Depth	2.42	10.01	1.07	0.22	2.28	9.73	0.97	0.16
	PCT [18]	Depth	4.03	14.67	1.74	0.36	4.15	14.51	1.71	0.35
	M3D-RPN [2] in [15]	None	3.61	11.12	2.12	0.24	3.46	10.67	2.04	0.20
	GUPNet [9] in [5]	None	9.39	24.69	4.67	0.19	9.31	24.50	4.62	0.19
	DEVIANT [5]	None	10.29	<u>26.75</u>	<u>4.95</u>	0.16	10.20	<u>26.54</u>	<u>4.90</u>	0.16
	MonoUNI [4]	None	<u>10.38</u>	26.57	3.95	<u>0.53</u>	<u>10.24</u>	26.24	3.89	<u>0.51</u>
	MonoDGP (Ours)	None	**11.71**	**31.02**	**5.61**	**1.17**	**11.56**	**30.58**	**5.54**	**1.15**

Table 5. Results on the Waymo val set for the vehicle category. Compared with methods without extra data, we **bold** the best results and <u>underline</u> the second-best results.

G. Experiments on Waymo Open Dataset

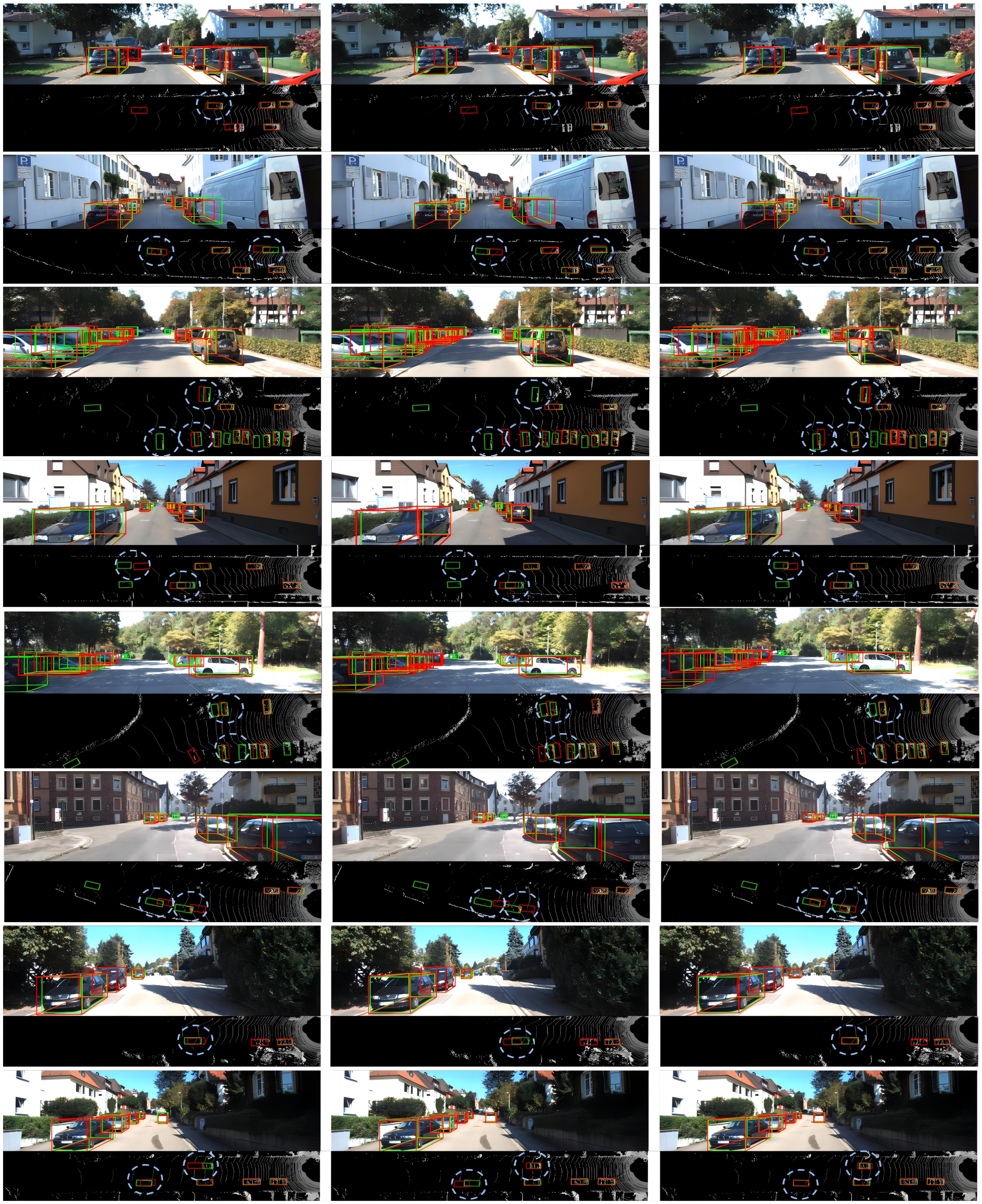
Waymo [17] evaluates objects at Level_1 and Level_2, which are determined by the number of LiDAR points within their 3D bounding boxes. The experiments is conducted across three distance ranges: [0, 30), [30, 50), and [50, ∞) meters. Performance on the Waymo dataset is assessed by average precision AP_{3D} and average precision weighted by heading APH_{3D}.

We follow the DEVIANT [5] split to generate 52,386 training and 39,848 validation images by sampling every third frame. For fairness, we mainly compare with methods using the same split in Tab. 5. Our method achieves state-of-the-art performance without extra data across all ranges, particularly for distant objects. These results further validate the effectiveness and generalizability of MonoDGP. It is worth noting that CaDDN [15]'s performance is better than MonoDGP, this discrepancy may be attributed to different dataset splits and introduction of LiDAR data.

H. Qualitative Discussion and Visualization

To provide a more intuitive comparison between our method and the baseline models, we visualize some 3D detection results from both the camera view and the bird's-eye view on the KITTI validation set. As shown in Fig. 4, our method demonstrates superior performance on distant and length-occluded objects.

However, since error prediction is affected by the initial accuracy of geometric depth, which is calculated from height relationships, height occlusion remains a challenge for our method. For the leftmost vehicle in the third example of Fig. 4, bushes block out its lower part, weakening the accuracy of height and consequently propagating errors to depth prediction. This failure case highlights the need for further improvements in handling height occlusion, potentially through the integration of additional contextual information or more robust occlusion-aware models.



(a) MonoCD (b) MonoDETR (c) MonoDGP(Ours)

Figure 4. Qualitative results on KITTI validation set. (a) MonoCD (b) MonoDETR (c) MonoDGP (ours). In each group of images, the first row shows the camera view, and the second row shows the bird's-eye view. Green represents the ground truth of boxes, while Red represents the prediction results. We also circle some objects to highlight the difference between the baseline model and our method.

References

[1] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3d object detection via geometric reasoning on keypoints. *arXiv preprint arXiv:1905.05618*, 2019. 2

[2] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, pages 9287–9296, 2019. 4

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 3

[4] Jia Jinrang, Zhenjia Li, and Yifeng Shi. Monouni: A unified vehicle and infrastructure-side monocular 3d object detection network with sufficient depth clues. *Advances in Neural Information Processing Systems*, 36:11703–11715, 2023. 4

[5] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. Deviant: Depth equivariant network for monocular 3d object detection. In *ECCV*, pages 664–683. Springer, 2022. 3, 4

[6] Zhuoling Li, Zhan Qu, Yang Zhou, Jianzhuang Liu, Haoqian Wang, and Lihui Jiang. Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection. In *CVPR*, pages 2791–2800, 2022. 3

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 2

[8] Xianpeng Liu, Nan Xue, and Tianfu Wu. Learning auxiliary monocular contexts helps monocular 3d object detection. In *AAAI*, pages 1810–1818, 2022. 3

[9] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, pages 3111–3121, 2021. 2, 3, 4

[10] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 4

[11] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, pages 4721–4730, 2021. 2

[12] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2

[13] Andretti Naiden, Vlad Paunescu, Gyeongmo Kim, Byeong-Moon Jeon, and Marius Leordeanu. Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. In *2019 IEEE international conference on image processing (ICIP)*, pages 61–65. IEEE, 2019. 2

[14] Liang Peng, Junkai Xu, Haoran Cheng, Zheng Yang, Xiaopei Wu, Wei Qian, Wenxiao Wang, Boxi Wu, and Deng Cai. Learning occupancy for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10281–10292, 2024. 3

[15] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, pages 8555–8564, 2021. 3, 4

[16] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 3

[17] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 4

[18] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3d object detection. *Advances in Neural Information Processing Systems*, 34:13364–13377, 2021. 4

[19] Zizhang Wu, Yuanzhu Gan, Lei Wang, Guilian Chen, and Jian Pu. Monopgc: Monocular 3d object detection with pixel geometry contexts. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4842–4849. IEEE, 2023. 3

[20] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2025. 3

[21] Renrui Zhang, Han Qiu, Tai Wang, Ziyu Guo, Ziteng Cui, Yu Qiao, Hongsheng Li, and Peng Gao. Monodetr: Depth-guided transformer for monocular 3d object detection. In *ICCV*, pages 9155–9166, 2023. 3