Advancing Self-supervised Monocular Depth Learning with Sparse LiDAR Supplementary Materials

This document contains the supplementary materials for "Advancing Self-supervised Monocular Depth Learning with Sparse LiDAR".

1 More Results for Depth Prediction

Depth Prediction on KITTI Dataset: Due to the space limitation, we only compared with part of the sate-of-the-art methods for dense depth prediction task in the main paper. Here, Table 1 shows the complete comparison with the state-of-the-art methods on KITTI [1] dataset. By effectively using low-cost sparse LiDAR points, our method achieves more accurate dense depth predictions than all state-of-the-art counterparts including the sparse-LiDAR based methods.

Statistics by Semantic Categories: Fig. 1 shows the comparison of depth prediction error by different semantic categories in the KITTI [1] dataset, while Fig 2 shows the average number of pixels per image by different semantic categories. Our proposed model consistently and significantly improves the depth quality for all the semantic categories.

Depth Error Qualitative Analysis: Fig. 5 shows the complete qualitative comparison of the depth errors of our method and the image-based depth prediction model monodepth2 [2]. Leveraging low-cost sparse LiDAR information, our method produces much better results on all of the objects.

Effectiveness of RefineNet: To better understand the effectiveness of our RefineNet, we show the qualitative comparison between the initial depth and the refined depth in Fig. 6. These results show that our proposed RefineNet can significantly reduce the depth error on all these objects.

Computational Efficiency of RefineNet: The existing depth correction / refinement methods [3, 4] conducts iterative optimization on the testing data, and normally they have higher accuracy but is extremely slow (1-2 FPS). Table 2 shows the comparison between our RefineNet and the existing method including GDC [3] and PnP-Depth [4]. The comparison shows that our RefineNet is more efficient, achieving real-time speed (139 FPS) on single Nvidia RTX-2080Ti GPU.

LiDAR Sparsity. As shown in Fig. 3, our proposed method can consistently improve the depth prediction even when only one beam of sparse LiDAR points is used. And our method significantly outperforms other methods [5, 3, 6] when the same amount of sparse LiDAR points are used.

2 More Results for Monocular 3D Object Detection

Comparison With the State-of-the-Art: In addition to the concise quantitative comparison for 3D detection in the main paper, here we show a more comprehensive quantitative evaluation of how our advanced depth prediction improves downstream tasks. We employ the PatchNet [7] to perform the 3D monocular object detection on the KITTI [1] dataset using the depth maps generated from our model. Table. 3 shows the full comparison between our method and state-of-the-art methods on the KITTI testing set, using the KITTI official testing server. Our method significantly outperforms all counterparts, including the Pseudo LiDAR++ [3] that also uses the sparse LiDAR points.

Qualitative Comparison: Fig. 4 shows the qualitative comparison between our method and the state-of-the-art monocular depth prediction model Monodepth2 [2] on the KITTI validation set. The PatchNet [7] is employed for detection which takes the depth maps generated by our model and the Monodepth2 as inputs respectively. Note that the Monodepth2 also needs 4-beams LiDAR points

5th Conference on Robot Learning (CoRL 2021), London, UK.

Method	Train	The lower the better				The higher the better		
		Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
SfMLearner [8]	M	0.208	1.768	6.958	0.283	0.678	0.885	0.957
DNC [9]	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Vid2Depth [10]	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
LEGO [11]	M	0.162	1.352	6.276	0.252	0.783	0.921	0.969
GeoNet [12]	M	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DF-Net [13]	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
DDVO [14]	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
EPC++ [15]	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2Depth [16]	M	0.141	1.036	5.291	0.215	0.816	0.945	0.979
SIGNet [17]	M	0.133	0.905	5.181	0.208	0.825	0.947	0.981
CC [18]	M	0.140	1.070	5.326	0.217	0.826	0.941	0.975
LearnK [19]	M	0.128	0.959	5.230	0.212	0.845	0.947	0.976
DualNet [20]	M	0.121	0.837	4.945	0.197	0.853	0.955	0.982
SuperDepth [21]	M	0.116	1.055	-	0.209	0.853	0.948	0.977
Monodepth2 [2]	M	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Guizilini *et al.* [22]	M	0.111	0.785	4.601	0.189	0.878	-	-
PackNet-SfM [23]	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
FeatDepth [24]	M	0.104	0.729	4.481	0.179	0.893	0.965	0.984
MonoDepth [25]	S	0.133	1.142	5.533	0.230	0.830	0.936	0.970
MonoDispNet [26]	S	0.126	0.832	4.172	0.217	0.840	0.941	0.973
MonoResMatch [27]	S	0.111	0.867	4.714	0.199	0.864	0.954	0.979
MonoDepth2 [2]	S	0.107	0.849	4.764	0.201	0.874	0.953	0.977
RefineDistill [28]	S	0.098	0.831	4.656	0.202	0.882	0.948	0.973
UnDeepVO [29]	M+S	0.183	1.730	6.570	0.268	-	-	-
DFR [30]	M+S	0.135	1.132	5.585	0.229	0.820	0.933	0.971
EPC++ [15]	M+S	0.128	0.935	5.011	0.209	0.831	0.945	0.979
MonoDepth2 [2]	M+S	0.106	0.818	4.750	0.196	0.874	0.957	0.979
DepthHint [31]	M+S††	0.100	0.728	4.469	0.185	0.885	0.962	0.982
FeatDepth [24]	M+S	0.099	0.697	4.427	0.184	0.889	0.963	0.982
Dorn [32]	M+Sup	0.099	0.593	3.714	0.161	0.897	0.966	0.986
BTS [33]	M+Sup	0.091	0.555	4.033	0.174	0.904	0.967	0.984
Guizilini *et al.* [22]*	M+L	0.082	0.424	3.73	**0.131**	0.917	-	-
Ours (Initial Depth)	M+L	0.078	0.515	3.67	0.154	0.935	0.973	0.986
Ours (Refined Depth)	M+L	**0.074**	**0.423**	**3.61**	0.150	**0.936**	**0.973**	**0.986**
Struct2Depth [16]	M†	0.109	0.825	4.750	0.187	0.874	0.958	0.983
GLNet [34]	M†	0.099	0.796	4.743	0.186	0.884	0.955	0.979
FeatDepth [24]	M†	0.088	0.712	4.137	0.169	0.915	0.965	0.982
FeatDepth [24]	M+S†	0.079	0.666	3.922	0.163	0.925	0.970	0.984
Pseudo LiDAR++ (GDC) [3]**	M+L†	0.098	0.714	4.30	0.176	0.899	0.967	0.984
Ours (Initial Depth + GDC)	M+L†	0.067	0.423	3.42	0.144	0.941	0.977	0.988
Ours (Refined Depth + GDC)	M+L†	**0.063**	**0.364**	**3.291**	**0.139**	**0.945**	**0.978**	**0.988**

Table 1: **Depth prediction on KITTI original dataset:** Methods are ranked by absolute relative error. The best results are in bold. All methods are using a resolution of 640x192 pixels. Due to the exceptional time-consume (around 1-2 FPS), we rank methods with and without iterative refinement separately. M, S, and L respectively indicates Monocular, Stereo, and Sparse LiDAR data, with Sup and † respectively indicating supervised training and iterative correction in testing phase.
* Only use LiDAR data in training phase, but tested on the KITTI improved dataset, which usually has a much lower error value.
** For a fair comparison, we replace the supervised stereo depth module with monodepth2 [2].

Method	Iterative	Abs Rel	Speed (FPS)
Without Refinement	—	0.078	-
Ours (Refine Net + GDC)	Yes	0.064	2.00
GDC [3]	Yes	0.067	2.01
PnP Depth [4]	Yes	0.077	15.2
Ours (Refine Net)	No	0.074	139.0

Table 2: **Speed comparison between our RefineNet and other conventional iterative refinement methods:** The other methods use LiDAR point cloud to iteratively refine the predictions and it usually results with higher accuracy but super low speed (around 1-2 FPS). As its replacement, our newly designed RefineNet is an efficient feed-forward network that achieves real-time performance (139 FPS) on single Nvidia RTX-2080Ti GPU.

to retrieve the absolute metric scale of the depth map before detection. With more accurate depth predictions, our method leads to much better detection results than the Monodepth2.

3 More Implementation Details

Dense Depth Prediction: The proposed framework is trained on KITTI Depth Prediction dataset with an Adam optimizer [35] with a learning rate starting at $1e-4$ and reduced by 90% every 15 epochs. Our model takes images of resolution 640×192 as input and outputs predictions of same

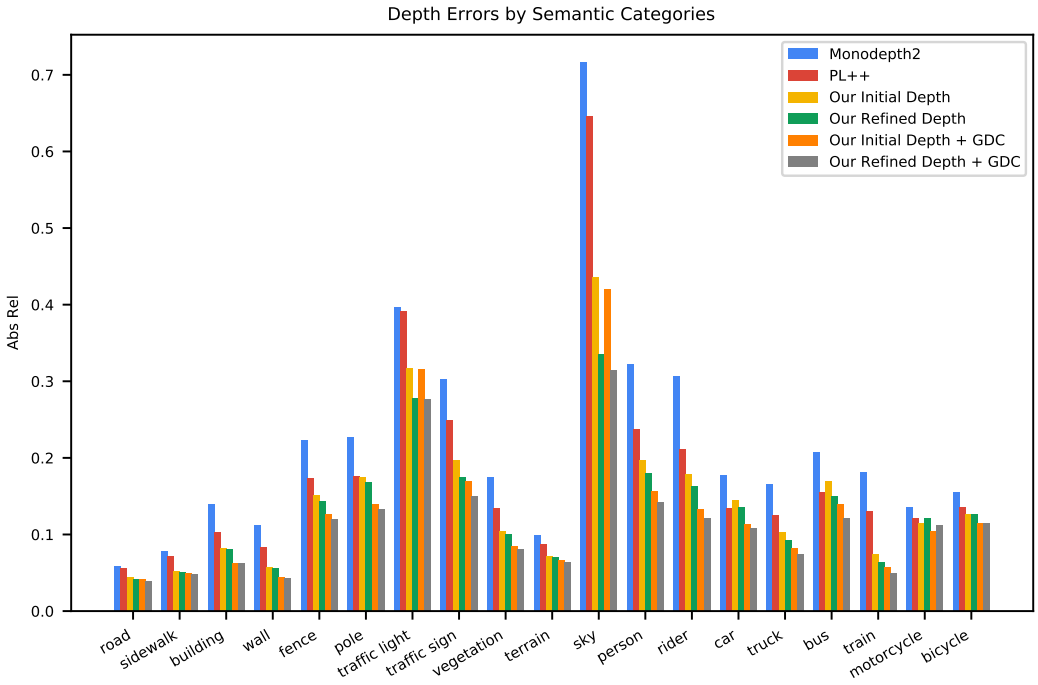

Figure 1: **Depth Error by Semantic Categories:** The depth absolute relative error analysis by different semantic categories in the KITTI test set. Our proposed model consistently improves the depth quality for all the semantic categories.

resolution. All the models are trained with a batch size of 8 on a single NVIDIA Tesla V100 GPU for around 15 hours.

Depth Completion: Our framework is trained with an Adam optimizer [35] with a learning rate starting at $1e-4$ and reduced by 90% every 8 epochs. Our model takes images of resolution 1216×352 as input and outputs predictions of the same resolution. All the models are trained with a batch size of 4 on a single NVIDIA Tesla V100 GPU for 15 epochs, and the training takes around 20 hours.

Monocular 3D Object Detection: For monocular 3D object detection, the most recent state-of-the-art model PatchNet [7] is employed as detector to evaluate the performance based on our predicted depth. The PatchNet is trained on the KITTI detection dataset with pseudo-LiDAR patches as input, which is lifted from our predicted depth. The model is optimized with an Adam optimizer [35] with a learning rate starting at $1e-3$ and reduced by 90% every 40 epochs. The entire optimization is done with 100 epochs, and it takes around 10 hours on a single NVIDIA Tesla V100 GPU.

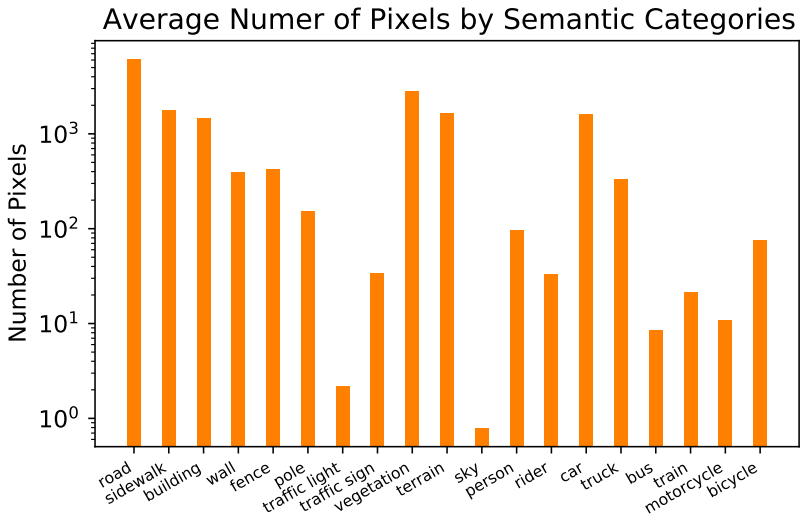

Figure 2: **Number of Pixels by Semantic Categories:** The average number of pixels per image by different semantic categories in the KITTI test set.

Method	Supervised Depth	KITTI Testing ($AP|_{40}$) Easy	Mod.	Hard
OFTNet [36]	-	1.61	1.32	1.00
FQNet [37]	-	2.77	1.51	1.01
ROI-10D [38]	-	4.32	2.02	1.46
GS3D [39]	-	4.47	2.90	2.47
Shift R-CNN [40]	-	6.88	3.87	2.83
Multi-Fusion [41]	✗	7.08	5.18	4.68
MonoGRNet [42]	✓	9.61	5.74	4.25
Decoupled-3D [43]	✓	11.08	7.02	5.63
MonoPSR [44]	-	10.76	7.25	5.85
MonoPL [45]	✓	10.76	7.50	6.10
SS3D [46]	-	10.78	7.68	6.51
MonoDIS [47]	-	10.37	7.94	6.40
M3D-RPN [48]	-	14.76	9.71	7.42
AM3D [49]	✓	16.50	10.74	9.52
PatchNet [7]	✓	15.68	11.12	10.17
Pseudo LiDAR++ [3]	✗	14.93	10.85	9.50
Ours	✗	**25.21** **+68.9%**	**18.99** **+75.0%**	**16.53** **+74.0%**

Table 3: **3D detection performance evaluation** for the **Car** category on the *testing* set of KITTI dataset [1]. IoU threshold is set to 0.7. For fair comparison, we replace the supervised stereo depth module of Pseudo LiDAR++ with Monodepth2 [2]. Our method significantly outperforms all counterparts, including the Pseudo LiDAR++ [3] that also uses the sparse LiDAR points.

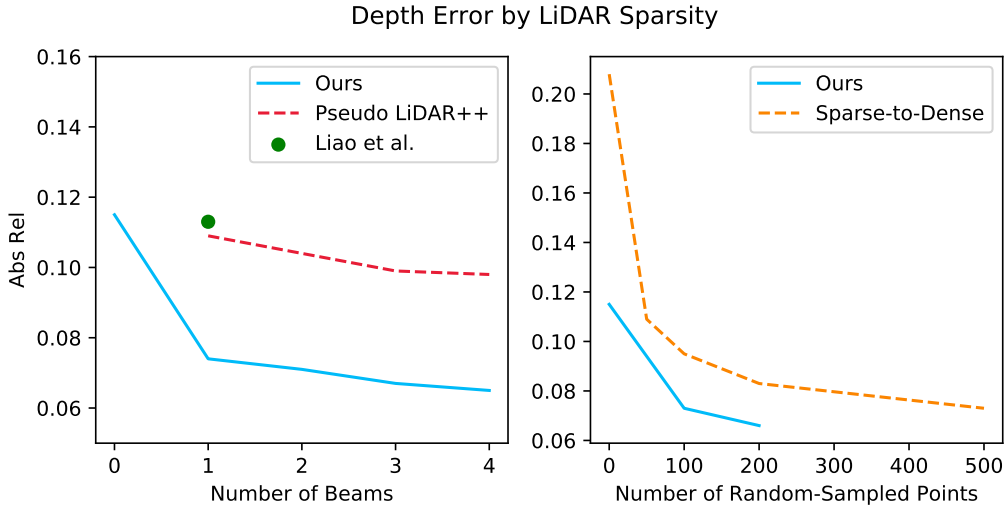

Figure 3: **Depth Error With Different LiDAR Sparsity:** The depth absolute relative errors on the KITTI depth prediction dataset.

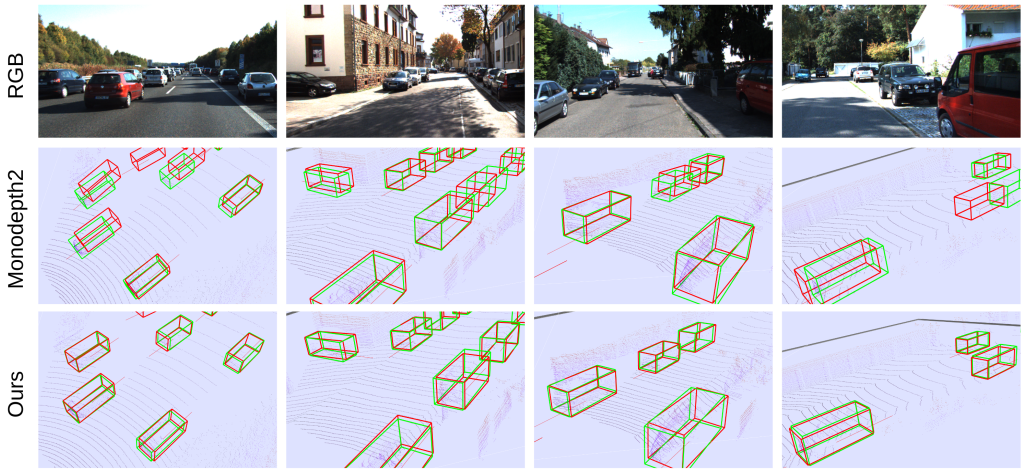

Figure 4: **Monocular 3D Object Detection:** The qualitative comparison of monocular 3D detection by PatchNet [7] based on the depth from our model and the Monodepth2 [2]. With the accurate dense depth prediction, our method produces much better detection results than the Monodepth2. Green boxes are the ground truth boxes while red boxes are the detection results. The LiDAR points in this figure are only used for visualization purpose.

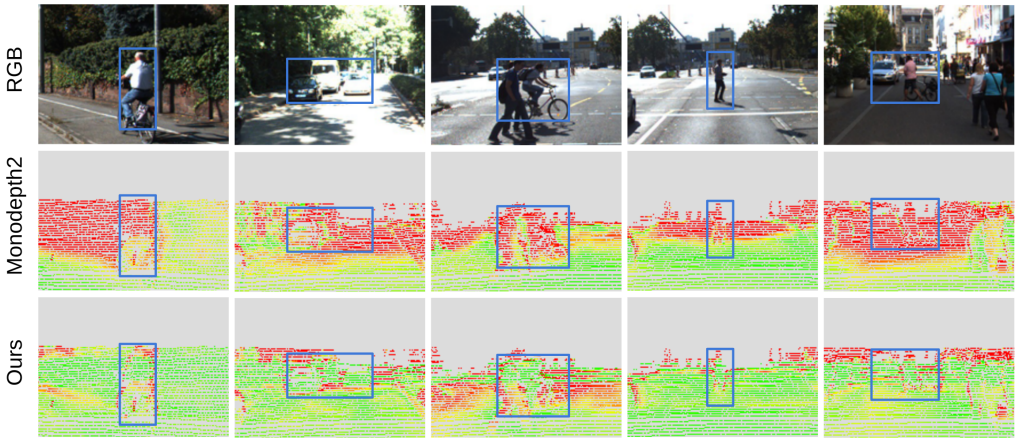

Figure 5: **Depth Error Qualitative Analysis:** The depth absolute error. The first to third rows are: the input RGB image, the prediction of Monodepth2 [2], and the predictions by our method respectively. The Red, yellow, and green indicate the depth error from high to low (best viewed in color). By fusing the low-cost sparse LiDAR information, our method generates much better results than the baseline which only rely on image features for all these objects.

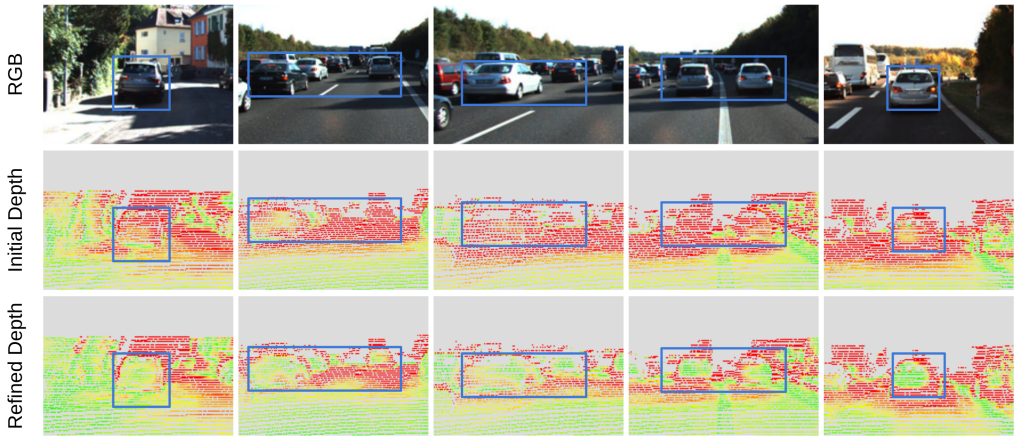

Figure 6: **Effectiveness of RefineNet:** The depth absolute error. The first to third rows are: the input RGB image, our initial depth prediction, and our refined depth prediction. The red, yellow, and green indicate the depth error from high to low (best viewed in color). These results show that our proposed RefineNet can significantly reduce the depth error on all these objects.

References

[1] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. IEEE Computer Society.

[2] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. IEEE.

[3] Y. You, Y. Wang, W. Chao, D. Garg, G. Pleiss, B. Hariharan, M. E. Campbell, and K. Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. OpenReview.net.

[4] T.-H. Wang, F.-E. Wang, J.-T. Lin, Y.-H. Tsai, W.-C. Chiu, and M. Sun. Plug-and-play: Improve depth prediction via sparse data propagation. IEEE.

[5] F. Ma and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. IEEE.

[6] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu. Parse geometry from a line: Monocular depth estimation with partial laser observation. IEEE.

[7] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang. Rethinking pseudo-lidar representation. Springer.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. IEEE Computer Society.

[9] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency.

[10] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. IEEE Computer Society.

[11] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. LEGO: learning edge with geometry all at once by watching videos. IEEE Computer Society.

[12] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. IEEE Computer Society.

[13] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency.

[14] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. IEEE Computer Society.

[15] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv:1810.06125*.

[16] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. AAAI Press.

[17] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, and D. Bharadia. Signet: Semantic instance aided unsupervised 3d geometry perception. Computer Vision Foundation / IEEE.

[18] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. Computer Vision Foundation / IEEE.

[19] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. IEEE.

[20] J. Zhou, Y. Wang, K. Qin, and W. Zeng. Unsupervised high-resolution depth learning from videos with dual networks. IEEE.

[21] S. Pillai, R. Ambrus, and A. Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation.

[22] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon. Robust semi-supervised monocular depth estimation with reprojected distances. PMLR, .

[23] V. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon. Packnet-sfm: 3d packing for self-supervised monocular depth estimation. *arXiv:1905.02693*, .

[24] C. Shu, K. Yu, Z. Duan, and K. Yang. Feature-metric loss for self-supervised learning of depth and egomotion. Springer.

[25] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. IEEE Computer Society.

[26] A. Wong and S. Soatto. Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. Computer Vision Foundation / IEEE.

[27] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. Computer Vision Foundation / IEEE.

[28] A. Pilzer, S. Lathuilière, N. Sebe, and E. Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. Computer Vision Foundation / IEEE.

[29] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning.

[30] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. D. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. IEEE Computer Society.

[31] J. Watson, M. Firman, G. J. Brostow, and D. Turmukhambetov. Self-supervised monocular depth hints. IEEE.
[32] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. IEEE Computer Society.
[33] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv:1907.10326*.
[34] Y. Chen, C. Schmid, and C. Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. IEEE.
[35] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization.
[36] T. Roddick, A. Kendall, and R. Cipolla. Orthographic feature transform for monocular 3d object detection. BMVA Press.
[37] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou. Deep fitting degree scoring network for monocular 3d object detection. Computer Vision Foundation / IEEE.
[38] F. Manhardt, W. Kehl, and A. Gaidon. ROI-10D: monocular lifting of 2d detection to 6d pose and metric shape. Computer Vision Foundation / IEEE.
[39] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang. GS3D: an efficient 3d object detection framework for autonomous driving. Computer Vision Foundation / IEEE.
[40] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu. Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. IEEE.
[41] B. Xu and Z. Chen. Multi-level fusion based 3d object detection from monocular images. IEEE Computer Society.
[42] Z. Qin, J. Wang, and Y. Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. AAAI Press.
[43] Y. Cai, B. Li, Z. Jiao, H. Li, X. Zeng, and X. Wang. Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10478–10485, 2020.
[44] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3d proposal generation and object detection from view aggregation. IEEE.
[45] X. Weng and K. Kitani. Monocular 3d object detection with pseudo-lidar point cloud.
[46] E. Jörgensen, C. Zach, and F. Kahl. Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. *CoRR*.
[47] A. Simonelli, S. R. Bulò, L. Porzi, M. Lopez-Antequera, and P. Kontschieder. Disentangling monocular 3d object detection. IEEE.
[48] G. Brazil and X. Liu. M3D-RPN: monocular 3d region proposal network for object detection. IEEE.
[49] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. IEEE.