Supplementary Material for Deep Stereo Geometry Network (DSGN)

A. Appendix

A.1. More Experiments

Correlation between stereo (depth) and 3D object detection accuracy. Following KITTI stereo metric, we consider a pixel as being correctly estimated if its depth error is less than an *outlier_thresh*. The percentage of non-outlier pixels inside the object box is deemed as depth estimation precision.

With several *outlier_threshes* (0.1, 0.3, 0.5, 1.0, 2.0 meters), scatter plots of stereo and detection precision are drawn. We observed that when *outlier_thresh* is 0.3 meter, the strongest linear correlation is yielded.

Outlier_thresh	$>$ 2m	$>$ 1m	$>$ 0.5m	$>$ 0.3m	$>$ 0.1m
PCC	0.249	0.353	0.438	0.450	0.417

Table 1. Pearson's correlation coefficients (PCC) for a set of *outlier_threshes* between depth estimation precision and detection accuracy. *Outlier_thresh*$= 0.3m$ yields the strongest linear correlation.

Figure 1 shows the scatter plots with the *outlier_thresh* 0.3m and 0.1m. Quite a few predictions get over 0.7 detected precision within a certain range of depth estimation error. This reveals that the end-to-end network gives rise to its ability to detect 3D objects even with a larger depth estimation error. The following 3D detector enables compensation for the stereo depth estimation error by 3D location regression with back-propagation.

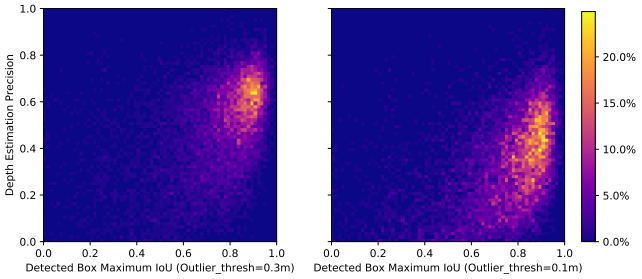

Figure 1. BEV Detection precision versus depth precision for all predicted boxes for *Car* on KITTI *val* set. Only TPs with IoU$>$ 0.01 and score $>$ 0.1 are shown.

Relationship between distance and detection accuracy. Figure 2 illustrates, as distance increases, all detection accuracy indicators have a shrinking tendency. The average accuracy maintains 80%+ within 25 meters. In all indicators, 3D AP decreases the fastest, followed by BEV AP and last 2D AP. This observation suggests that the 3D detection accuracy is determined by the BEV location precision beyond 20 meters.

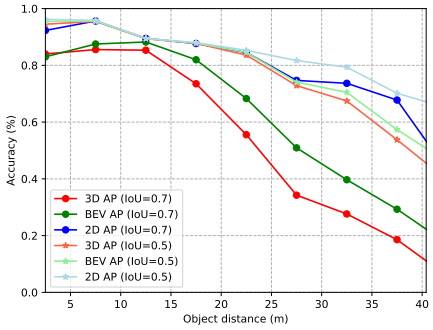

Figure 2. Detection accuracy versus distance. We separate the range [0, 40] (meters) into 8 intervals, each with 5 meters. All evaluations are conducted within each interval.

Influence of different features for 3D geometry. We discuss the efficiency of different geometric representations for volumetric structure. Most depth prediction approaches [3, 5, 2, 4] apply the strategy of "depth classification" (such as cost volume) instead of "depth regression". Thus, we have several choices for encoding the depth information of cost volume into a 3D volume. The intuitive one is to use a 3D voxel occupancy (denoted by "Occupancy"). An advanced version is by keeping the probability of voxel occupancy (denoted by "Probability"). They both have explicit meaning for 3D geometry and can be easily visualized. Another one is by using the last feature map for cost volume as geometric embedding for 3D volume (denoted by "Last Features").

Table 2 reveals the performance gap between "Occupancy" / "Probability". "Last Features" indicates the latent feature embedding (64D) that enables the network to extract more 3D latent geometric information and even semantic

Voxel Features	AP_{3D} / AP_{BEV} / AP_{2D}
Occupancy	37.86 / 50.64 / 70.79
Probability	43.24 / 54.87 / 74.93
Last Features	**54.27 / 63.91 / 83.59**

Table 2. Ablation study on 3D geometric representation. "Occupancy" indicates only using binary feature for 3D volume. It is 1 for voxel of minimum cost along the projection ray and is 0 otherwise. "Probability" denotes keeping the probability of voxel occupancy instead of quantizing it to 0 or 1. "Last Features" represents transforming the last features of cost volume to 3D volume.

JOINT	IMG	ATT	Depth	HG	Flip	AP_{3D}/AP_{BEV}/AP_{2D}
						40.71 / 53.71 / 76.11
✓						45.51 / 56.65 / 78.31
✓	✓					44.79 / 56.24 / 81.58
✓	✓	✓				46.52 / 57.44 / 82.41
✓			✓			45.79 / 56.89 / 78.49
✓	✓	✓		✓		51.73 / 61.74 / 83.6
✓	✓	✓		✓	✓	54.27 / 63.91 / 83.59

Table 3. Ablation study evaluated in moderate level. "JOINT" indicates using joint optimization instead of separable optimization for bounding boxes regression. "IMG" denotes concatenating the mapped left image feature to 3DGV for retrieving more 2D semantics. "ATT (Attention)" represents concatenating the mapped image feature weighted by the corresponding depth probability. "Depth" indicates warping the final matching cost volume to 3DGV. "HG (Hourglass)" represents applying an hourglass module in 3DGV. "Flip" means using random horizontal flipping augmentation.

cues than the explicit voxel occupancy. It aids learning of 3D structure.

Technical details We explore several technical details used in DSGN and discuss their importance in the pipeline in Table 3. Joint optimization of bounding box regression improves accuracy ($+4.80$ AP) than the separable optimization. The intermediate 3D volume representation enables the network to naturally retrieve image feature for more 2D semantic cues. However, 3DGV cannot directly benefit from the concatenation of mapped 32D image features and warped predicted cost volume. Instead, the image feature weighted by the depth probability achieves $+1.01$ AP gain. Further, Involving more computation by an extra hourglass module on 3D object detector and flip augmentation, DSGN finally achieves 54.27 AP on 3D object detection.

***Pedestrian* and *Cyclist* detection.** The main challenges for detecting *Pedestrian* and *Cyclist* are the limited data (about only 1/3 of images are annotated) and the difficulty to estimate their poses in an image even for human. As a result, most image-based approaches yield poor performance or are not validated on *Pedestrian* and *Cyclist*. Since the evaluation metric is changed on the official KITTI leaderboard, We only report the available results from original papers and the KITTI leaderboard.

Experimental results in Table 4 shows that our approach achieves better results on *Pedestrian* but worse ones on *Cyclist* compared with PL: F-PointNet. We note that PL: F-PointNet used Scene Flow dataset [8] to pre-train the stereo matching network. Besides, PL: F-PointNet achieves the best result on *Pedestrian* and the model PL: AVOD works best on *Car* and *Cyclist*. Table 5 shows the submitted results on the official KITTI leaderboard.

A.2. More Implementation Details

Network Architecture. We show the full network architecture in Table 6, including the networks for 2D feature extraction, constructing plane-sweep volume and 3D geometric volume, stereo matching and 3D object detection.

Implementation Details of 3D Object Detector. Given the feature map $\mathcal{F}$ on bird's eye view, we put four anchors of different orientation angles $(0, \pi/2, \pi, 3\pi/2)$ on all locations of $\mathcal{F}$. The box sizes of pre-defined anchors used for respectively *Car*, *Pedestrian*, *Cyclist* are $(h_{\mathbf{A}} = 1.56, w_{\mathbf{A}} = 1.6, l_{\mathbf{A}} = 3.9)$, $(h_{\mathbf{A}} = 1.73, w_{\mathbf{A}} = 0.6, l_{\mathbf{A}} = 0.8)$, and $(h_{\mathbf{A}} = 1.73, w_{\mathbf{A}} = 0.6, l_{\mathbf{A}} = 1.76)$.

The horizontal coordinate $(x_{\mathbf{A}}, z_{\mathbf{A}})$ of each anchor lies on the center of each grid in bird's eye view and its center along the vertical direction locates on $y_{\mathbf{A}} = 0.825$ for *Car* and $y_{\mathbf{A}} = 0.74$ for *Pedestrian* and *Cyclist*. We set $\gamma = 1$ for *Car* and $\gamma = 5$ for *Pedestrian* and *Cyclist* for balancing the positive and negative samples. The classification head of 3D object detector is initialized following RetinaNet [7]. NMS with IoU threshold 0.6 is applied to filter out the predicted boxes on bird's eye view.

Implementation Details of Differentiable Warping from PSV to 3DGV. Let $\mathcal{U} \in \mathbb{R}^{H_I \times W_I \times D_I \times C}$ be the last feature map of PSV, where C is the channel size of features. We first construct a pre-defined 3D volume $\in \mathbb{R}^{H_V \times W_V \times D_V \times 3}$ to store the center coordinate (x, y, z) of each voxel in 3D space (Section **??**). Then we get the projected pixel coordinate (u, v) by multiplying the projection matrix. z is directly concatenated to pixel coordinate to get (u, v, z) in *camera frustum space*.

As a result, we get a coordinate volume $\in \mathbb{R}^{H_V \times W_V \times D_V \times 3}$, which stores the mapped coordinates in *camera frustum space*. By *trilinear interpolation*, we fetch the corresponding feature of $\mathcal{U}$ at the projected coordinates to construct the 3D volume $\mathcal{V} \in \mathbb{R}^{H_V \times W_V \times D_V \times C}$, *i.e.*, 3D geometric volume. We ignore the projected coordinates outside the image by setting these voxel features to 0. In backward operations, the gradient is passed and computed using the same coordinate volume.

Modality	Method	**3D Detection AP (%)**			BEV Detection AP (%)			2D Detection AP (%)		
		Easy	**Moderate**	Hard	Easy	**Moderate**	Hard	Easy	**Moderate**	Hard
Pedestrian										
Mono	M3D-RPN [1]	–	11.09	–	–	11.53	–	–	–	–
Stereo	PL: F-PointNet* [10]	33.8	27.4	24.0	41.3	34.9	30.1	–	–	–
	DSGN	**40.16**	**33.85**	**29.43**	**47.92**	**41.15**	**36.08**	**59.06**	**54.00**	**49.65**
Cyclist										
Mono	M3D-RPN [1]	–	2.81	–	–	3.61	–	–	–	–
Stereo	PL: F-PointNet* [10]	**41.3**	**25.2**	**24.9**	**47.6**	**29.9**	**27.0**	–	–	–
	DSGN	37.87	24.27	23.15	41.86	25.98	24.87	49.38	33.97	32.40

Table 4. Comparison of results for *Pedestrian* and *Cyclist* on KITTI *val* set. PL: F-PointNet* uses extra Scene Flow dataset to pretrain the stereo matching network.

Modality	Method	**3D Detection AP (%)**			BEV Detection AP (%)			2D Detection AP (%)		
		Easy	**Moderate**	Hard	Easy	**Moderate**	Hard	Easy	**Moderate**	Hard
Pedestrian										
Mono	M3D-RPN [1]	4.92	3.48	2.94	5.56	4.05	3.29	56.64	41.46	37.31
Stereo	RT3DStereo [6]	3.28	2.45	2.35	4.72	3.65	3.00	41.12	29.30	25.25
	DSGN	**20.53**	**15.55**	**14.15**	**26.61**	**20.75**	**18.86**	**49.28**	**39.93**	**38.13**
Cyclist										
Mono	M3D-RPN [1]	0.94	0.65	0.47	1.25	0.81	0.78	61.54	41.54	35.23
Stereo	RT3DStereo [6]	5.29	3.37	2.57	7.03	4.10	3.88	19.58	12.96	11.47
	DSGN	27.76	18.17	16.21	31.23	21.04	18.93	49.10	35.15	31.41

Table 5. Comparison of results for *Pedestrian* and *Cyclist* on KITTI test set (official KITTI Leaderboard).

A.3. Future Work

More further studies on stereo-based 3D object detection are recommended here.

The Gap with state-of-the-art LiDAR-based approaches. Although our approach achieves comparable performance with some LiDAR-based approaches on 3D object detection, there remains a large gap with state-of-the-art LiDAR-based approaches [14, 9, 15, 12]. Besides, an obvious problem is the accuracy gap on bird's eye view (BEV) detection. As shown in the table of main results, there is almost 12 AP gap on the moderate and hard level in BEV detection.

One possible solution is high-resolution stereo matching [13], which can help obtain more accurate depth information to increase the robustness for highly occluded, truncated and far objects.

3D Volume Construction. Table **??** shows basic comparison of volume construction in DSGN. We expect a more in-depth analysis of the volume construction from multi-view or binocular images, which serves as an essential component design for 3D object understanding. Besides, the effectiveness of 3D volume construction methods still requires more investigation since it needs to balance and provide both depth information and semantic information.

Computation Bottleneck. The computation bottleneck of DSGN locates on the computation of 3D convolutions for computing cost volume. Recent stereo matching work [16, 11] focused on accelerating the computation of cost volume. Another significant aspect of constructing cost volume is that current cost volume [5, 2] is designed for regressing disparity but not depth. Further research might explore more efficient feature encoding for the plane-sweep cost volume.

Network Architecture Design. There is a trade-off between stereo matching network and 3D detection network for balancing the feature extraction of pixel- and high-level features, which can be conducted by recent popular Network Architecture Search (NAS).

Application on Low-speed Scenario. Our approach shows comparable performance with the LiDAR-based approach on 3D and BEV detection in the close range in the KITTI easy set. Most importantly, it is affordable even with one strong GPU Tesla V100 ($11,458 (USD)) compared with the price of a 64-beam LiDAR $75,000 [17]. It is a promising application of image-based autonomous driving system for low-speed scenarios.

A.4. Qualitative Results

We provide a video demo [1] for visualization of our approach, which shows both the detected 3D boxes on front view and bird's eye view. The ground-truth LiDAR point cloud is shown on bird's eye view. The detection results are obtained by DSGN trained on KITTI training split only. The unit of the depth map is meter.

Some noise observed in the predicted depth map is mainly caused by the implementation details. (1) Noise in the near and far part: 3D volumes are constructed in [2,

[1] https://www.youtube.com/watch?v=u6mQW89wBbo

40.4] (meters). (2) Noise and large white zone in the higher region (>3m): The stereo branch is trained with a sparse GT depth map (64 lines around [-1,3 (meters) along the gravitational z-axis, captured by a 64-ray LiDAR).

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. 2019.

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018.

[3] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[4] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019.

[5] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, pages 66–75, 2017.

[6] Hendrik Königshof, Niels Ole Salscheider, and Christoph Stiller. Realtime 3D Object Detection for Automated Driving Using Stereo Vision and Semantic Information. In *Proc. IEEE Intl. Conf. Intelligent Transportation Systems*, Auckland, NZ, Oct 2019.

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

[8] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016.

[9] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. 2018.

[10] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, pages 8445–8453, 2019.

[11] Yan Wang, Zihang Lai, Gao Huang, Brian H Wang, Laurens van der Maaten, Mark Campbell, and Kilian Q Weinberger. Anytime stereo image depth estimation on mobile devices. In *ICRA*, pages 5893–5900. IEEE, 2019.

[12] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. 2018.

[13] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *CVPR*, June 2019.

[14] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, 2019.

[15] Chen Yilun, Liu Shu, Shen Xiaoyong, and Jia Jiaya. Fast point r-cnn. In *ICCV*, 2019.

Layers	Kernel Size	Chl	Output Size
Image Feature Extractor			
Input Image		3	$H_I \times W_I$
conv1_x	$(3\times3)\times3$	64	$H_I/2 \times W_I/2$
conv2_x	$(3\times3)\times3$	32	$H_I/2 \times W_I/2$
conv3_x	$(3\times3)\times6$	64	$H_I/4 \times W_I/4$
conv4_x	$(3\times3)\times12$	128	$H_I/4 \times W_I/4$
conv5_x	$(3\times3)\times4$ (dila=2)	192	$H_I/4 \times W_I/4$
conv5_4 → SPP Module			
branch_1	64×64 *avgpool* 3×3 bilinear interpolation	32	$H_I/4 \times W_I/4$
branch_2	32×32 *avgpool* 3×3 bilinear interpolation	32	$H_I/4 \times W_I/4$
branch_3	16×16 *avgpool* 3×3 bilinear interpolation	32	$H_I/4 \times W_I/4$
branch_4	8×8 *avgpool* 3×3 bilinear interpolation	32	$H_I/4 \times W_I/4$
Fusion of shadow and deep layers			
concat [conv3_6, conv4_12, conv5_4, branch1~4]		512	$H_I/4 \times W_I/4$
fusion_1	3×3	256	$H_I/4 \times W_I/4$
fusion_2	3×3	32	$H_I/4 \times W_I/4$
(fusion_2 (left), fusion_2 (right)) → Constructing Plane-Sweep Volume			
Plane-Sweep Volume		64	$H_I/4 \times W_I/4 \times D_I/4$
PS_conv1_x	$(3\times3\times3)\times2$	64	$H_I/4 \times W_I/4 \times D_I/4$
PS_conv2_x	$(3\times3\times3)\times2$ *add* PS conv1_2	64	$H_I/4 \times W_I/4 \times D_I/4$
PS_stack_1x	$(3\times3\times3)\times2$	128	$H_I/8 \times W_I/8 \times D_I/8$
PS_stack_2x	$(3\times3\times3)\times2$	128	$H_I/16 \times W_I/16 \times D_I/16$
PS_stack_3	deconv $3\times3\times3$ *add* PS_stack_12	128	$H_I/8 \times W_I/8 \times D_I/8$
PS_stack_4	deconv $3\times3\times3$ *add* PS_conv2_2	64	$H_I/4 \times W_I/4 \times D_I/4$
PS_stack_4 → Stereo Matching			
Depth conv_1	$3\times3\times3$	64	$H_I/4 \times W_I/4 \times D_I/4$
Depth conv_2	$3\times3\times3$	1	$H_I/4 \times W_I/4 \times D_I/4$
Upsample	*trilinear interpolation*	1	$H_I \times W_I \times D_I$
***soft argmin* function**		1	$H_I \times W_I$
PS_stack_4 → 3D Geometric Volume			
3D Geometric Volume		64	$H_V \times W_V \times D_V$
3DG_conv	$(3\times3\times3)$	64	$H_V \times W_V \times D_V$
3DG_ stack_1x	$(3\times3\times3)\times2$	128	$H_V/2 \times W_V/2 \times D_V/2$
3DG_ stack_2x	$(3\times3\times3)\times2$	128	$H_V/4 \times W_V/4 \times D_V/4$
3DG_ stack_3	deconv $3\times3\times3$ *add* 3DG_stack_12	128	$H_V/2 \times W_V/2 \times D_V/2$
3DG_ stack_4	deconv $3\times3\times3$ *add* 3DG_conv	64	$H_V \times W_V \times D_V$
3DG_stack_4→3D Geometric Volume on BEV			
	$4\times1\times1$ *avgpool* and reshape to bev	$64\times H_V/4$	$W_V \times D_V$
3DGVbev_ conv_x	$(3\times3)\times2$	128	$W_V \times D_V$
3DGVbev_conv_2 → Classification			
cls_conv_x	$(3\times3)\times4$	128	$W_V \times D_V$
bbox_cls	3×3	4×3	$W_V \times D_V$
3DGVbev_conv_2 → Regression			
reg_conv_x	$(3\times3)\times4$	128	$W_V \times D_V$
bbox_cls	3×3	10×3	$W_V \times D_V$
reg_conv_4 → Centerness			
bbox_centerness	3×3	4	$W_V \times D_V$

Table 6. Full network architecture of DSGN. The color of the table highlights different components.

[16] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *CVPR*, pages 6044–6053, 2019.

[17] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. 2019.