

# Disp R-CNN: Stereo 3D Object Detection via Shape Prior Guided Instance Disparity Estimation

Jiaming Sun<sup>1,2</sup> \* Linghao Chen<sup>1\*</sup> Yiming Xie<sup>1</sup> Siyu Zhang<sup>3</sup>

Qinhong Jiang<sup>2</sup> Xiaowei Zhou<sup>1†</sup> Hujun Bao<sup>1†</sup>

<sup>1</sup>Zhejiang University <sup>2</sup>SenseTime <sup>3</sup>Southern University of Science and Technology

## Abstract

In this paper, we propose a novel system named *Disp R-CNN* for 3D object detection from stereo images. Many recent works solve this problem by first recovering a point cloud with disparity estimation and then apply a 3D detector. The disparity map is computed for the entire image, which is costly and fails to leverage category-specific prior. In contrast, we design an instance disparity estimation network (*iDispNet*) that predicts disparity only for pixels on objects of interest and learns a category-specific shape prior for more accurate disparity estimation. To address the challenge from scarcity of disparity annotation in training, we propose to use a statistical shape model to generate dense disparity pseudo-ground-truth without the need of LiDAR point clouds, which makes our system more widely applicable. Experiments on the KITTI dataset show that, even when LiDAR ground-truth is not available at training time, *Disp R-CNN* achieves competitive performance and outperforms previous state-of-the-art methods by 20% in terms of average precision. The code will be available at <https://github.com/zju3dv/disprcnn>.

## 1. Introduction

3D object detection plays an important role in many applications such as autonomous driving and augmented reality. While most methods work with the LiDAR point cloud as input, stereo image-based methods have significant advantages. RGB images provide denser and richer color information compared to the sparse LiDAR point clouds while requiring a very low sensor price. Stereo cameras are also able to perceive longer distances with customizable baseline settings. Recently, learning-based approaches like [11, 4, 34] tackled the stereo correspondence matching problem with Convolutional Neural Networks (CNNs) and

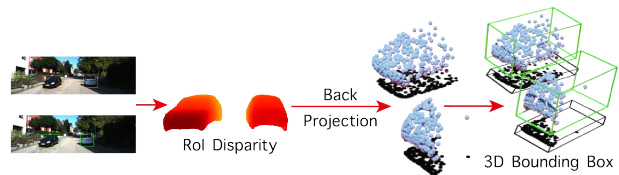


Figure 1. **The proposed system estimates an instance disparity map**, i.e., pixel-wise disparities only on foreground objects, for stereo 3D object detection. This design leads to better disparity estimation accuracy and faster run-time speed.

achieved impressive results. Taking an estimated disparity map as the input, 3D object detection methods [31, 30] convert it into a depth map or a point cloud to detect objects within it. However, since the disparity estimation network is designed for general stereo matching instead of the 3D object detection task, these pipelines have two major drawbacks. First, the disparity estimation process operates on the full image and often fails to produce accurate disparities on low textured or non-Lambertian surfaces like the surface of vehicles, which are exactly the regions we need to do successful 3D bounding boxes estimation. Moreover, since foreground objects of interest usually occupy much fewer space than the background in the image, the disparity estimation network and the 3D detector spend a lot of computation on regions that are not needed for object detection and lead to a slow running speed.

In this work, we aim to explore how we can solve these drawbacks with a disparity estimation module that is specialized for 3D object detection. We argue that estimating disparities on the full image is suboptimal in terms of network feature learning and runtime efficiency. To this end, we propose a novel system named *Disp R-CNN* that detects 3D objects with a network designed for instance-level disparity estimation. The disparity estimation is performed only on regions that contain objects of interest, thus enabling the network to focus on foreground objects and learn a category-specific shape prior that is suitable for 3D object detection. As demonstrated in the experiments, with the guidance of object shape prior, the estimated instance

\*The first two authors contributed equally. The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG.

†Corresponding authors: Hujun Bao and Xiaowei Zhou.

disparities capture the smooth shape and sharp edges of object boundaries while being more accurate than the full-frame counterpart. With the design of instance-level disparity estimation, the running time of the overall 3D detection pipeline is reduced thanks to the smaller number of input and output pixels and the reduced range of cost volume search in the disparity estimation process.

Another limitation of the full-frame disparity estimation is the lack of pixel-wise ground-truth annotation. In the KITTI dataset [9] for example, although it is possible to render disparity ground truth by manually selecting and aligning vehicle CAD models as in the KITTI Scene Flow benchmark [18], there is no such ground-truth provided in the KITTI Object Detection benchmark due to its difficulty in annotating on a massive scale. To make dense instance-level disparity supervision possible, we propose a pseudo-ground-truth generation process that can acquire accurate instance disparities and instance segmentation masks via object shape reconstruction and rendering. The object mesh is reconstructed by a PCA-based statistical shape model under several geometric constraints [13, 8]. The effort to manually annotate CAD models can be saved through this automated process since the basis of the statistical shape model can be learned directly from 3D model repositories like ShapeNet [3]. Different from some recent methods [30, 33, 7] that use the projected LiDAR point clouds as the sparse supervision for full-frame disparity estimation, our pseudo-ground-truth generation process can provide dense supervision even when LiDAR is not available at training time, which has a broader applicability in practice.

We evaluate our system on the KITTI dataset and provide ablation analysis of the different components of the proposed system. The experiments show that, with the guidance of the shape prior introduced by both the network design and the generated pseudo-ground-truth, the performance of instance-level disparity estimation surpasses the full-frame counterpart by a large margin. As a result, 3D object detection performance can be largely improved compared to baseline state-of-the-art 3D detectors that rely on full-frame disparities. When LiDAR supervision is not used at training time, our method outperforms the baseline methods by 20% in terms of average precision (27% vs. 47%).

In summary, our contributions are as follows:

- A novel framework for stereo 3D object detection based on instance-level disparity estimation, which outperforms state-of-the-art baselines in terms of both accuracy and runtime speed.
- A pseudo-ground-truth generation process that provides supervision for the instance disparity estimation network and guides it to learn the object shape prior that benefits 3D object detection.

## 2. Related Work

In this section, we briefly review the recent progress of 3D object detection with different modalities of input data and introduce the background of object shape reconstruction that is used in the proposed pseudo-ground-truth generation process.

**3D object detection with RGB images.** Several works concentrate on 3D object detection using a monocular image or stereo RGB images as input. Stereo R-CNN [14] designs a Stereo Region Proposal Network to match left and right Regions of Interest (RoIs), and refines 3D bounding boxes by dense alignment. On the monocular side, [19] proposes to estimate 3D bounding boxes with relation and constraints between 2D and 3D bounding boxes. [31] uses a depth map as an extra input channel to assist 3D object detection. Recently, Pseudo-LiDAR [30] converts the disparity map estimated from stereo images to point clouds as pseudo-LiDAR points, estimates 3D bounding boxes with LiDAR-input approaches and achieves state-of-the-art performance on both monocular and stereo input. It is worth noting that, there are two concurrent works OC-Stereo [21] and ZoomNet [32] that propose the similar idea of instance-level disparity estimation. OC-Stereo [21] uses depth completion results from sparse LiDAR points as object-centric disparity supervision, and ZoomNet [32] prepares a human-annotated CAD model dataset to achieve a similar purpose. Our method differs from these above-mentioned works in the disparity estimation region (on objects vs. on full images) and the automated dense instance disparity pseudo-ground-truth generation process.

**3D object detection with point clouds.** A majority of state-of-the-art 3D object detection methods are based on point clouds captured by depth sensors (LiDAR or RGB-D camera) [6, 22] as input. F-PointNet [23] segments the object point cloud within the 2D RoI frustum into foreground and background and later predicts 3D bounding boxes with PointNet++ [24]. Recently, PointRCNN [27] adapts this framework into a two-stage design as in the 2D object detection counterpart [26] and achieved impressive performance. The 3D object detector in the proposed pipeline is point cloud based and can be substituted to other methods that can achieve the similar purpose.

**Object shape reconstruction.** 3D object detection can benefit from shape reconstruction. [8] leverages the constraint that the point cloud must be lying on the object surface, and jointly optimizes the object pose and shape with the point cloud generated from stereo disparities and object shape prior model learned from the 3D shape repository with PCA. [20] further extends this pipeline with the temporal kinematic constraints of objects in dynamic scenes. [29] proposes a continuous optimization approach to jointly

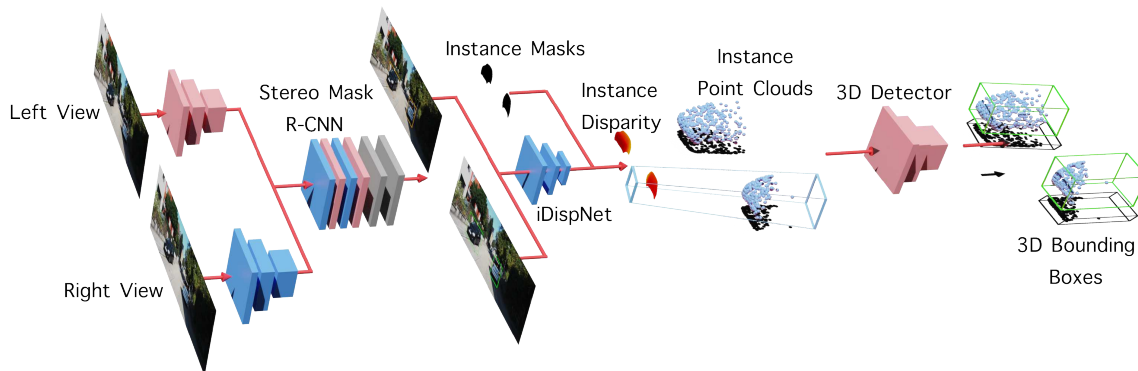


Figure 2. **Disp R-CNN Architecture.** Disp R-CNN has three stages. First, the input images are passed through a stereo variant of Mask R-CNN to detect 2D bounding boxes and instance segmentation masks. Then, the instance disparity estimation network (iDispNet) takes the cropped ROI images as input and estimates an instance disparity map. Finally, the instance disparity map is converted to an instance point cloud and fed into the 3D detector for 3D bounding box regression.

optimize object shape and pose with the photometric error. [17] proposes to use the object shape generated from a 3D auto-encoder in the data augmentation process during the training of monocular 3D object detection. For object categories other than vehicles (e.g., pedestrians and cyclists), shape reconstruction can be achieved similarly by fitting a statistical shape model (e.g., SMPL [15]) to point cloud data as demonstrated by the PedX dataset [12].

### 3. Methods

Given a pair of stereo images, the goal is to detect 3D bounding boxes of all the object instances of interest. As shown in Fig. 2, our detection pipeline consists of three stages: we first detect 2D bounding boxes and instance masks for each object, then estimate disparities only for pixels belonging to objects and finally use a 3D detector to predict 3D bounding boxes from the instance point cloud.

#### 3.1. Stereo Mask R-CNN

We start by briefly describing the base 2D detector that provides necessary input for the following modules of the pipeline. We extend the Stereo R-CNN [14] framework to predict the instance segmentation mask in the left image. Stereo Mask R-CNN is composed of two stages. The first stage is a stereo variant of the Region Proposal Network (RPN) as proposed in [14], where object proposals from the left and right images are generated from the same set of anchors to ensure the correct correspondences between the left and right Regions of Interest (ROIs). The second stage extracts object features from the feature map using RoIAlign as proposed in [10], followed by two prediction heads that produce 2D bounding boxes, classification scores, and instance segmentation masks.

#### 3.2. Instance Disparity Estimation Network

The disparity estimation module is responsible for recovering the 3D data in stereo 3D object detection and therefore its accuracy directly affects the 3D detection performance. Previous work [30] applies an off-the-shelf disparity estimation module that predicts the disparity map for all the pixels in the entire image. Since the area of the foreground objects only takes a small portion of the full image, most computation in both the disparity estimation network and the object detection network is redundant and can be reduced. Moreover, for the specular surfaces on most of the vehicles, the Lambertian reflectance assumption for the photometric-consistency constraint used in stereo matching cannot hold. To remedy these problems, we propose a learning-based instance disparity estimation network (iDispNet) that is specialized for 3D object detection. The iDispNet only takes the object ROI images as input and is only supervised on the foreground pixels, so that it captures the category-specific shape prior and thus produces more accurate disparity predictions.

Formally speaking, the *full-frame disparity* for a pixel  $p$  is defined as:

$$D_f(p) = u_p^l - u_p^r, \quad (1)$$

where  $u_p^l$  and  $u_p^r$  represent the horizontal pixel coordinates of  $p$  in the left and right views, respectively. With the 2D bounding boxes produced by the Stereo Mask R-CNN, we can crop the left and right ROIs out from the full images and align them in the horizontal direction. The width of each ROIs ( $w^l$ ,  $w^r$ ) are set to the larger value to make the two ROIs share the same size. Once ROIs are aligned, the disparity displacement for pixel  $p$  on the left image (reference) changes from *full-frame disparity* to *instance disparity*, which is defined as:

$$D_i(p) = D_f(p) - (b^l - b^r), \quad (2)$$

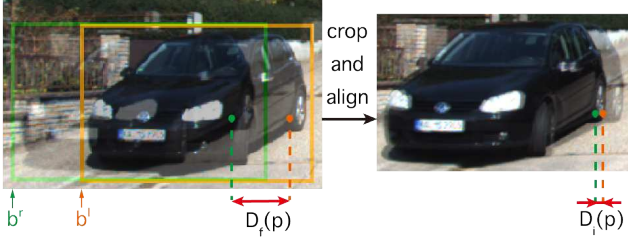


Figure 3. **The crop-and-align process** aligns the left and right ROIs by cutting off a *global offset*. As a result, the instance disparity  $D_i(p)$  distributes in a much narrower range compared to the full-frame disparity  $D_f(p)$ , which makes it possible to reduce the disparity search range when constructing the disparity cost volume and leads to faster inference.

where  $b^l$  and  $b^r$  stand for coordinates of the left border of bounding boxes in two views, respectively. Our goal is essentially to learn the instance disparity  $D_i(p)$  instead of  $D_f(p)$  for each  $p$  belonging to an object of interest. This crop-and-align process is visually illustrated in Fig. 3.

All the ROIs in the left and right images are resized to a common size  $H \times W$ . For all the pixels  $p$  that belong to an object instance  $O$  given by the instance segmentation mask, the loss function for the instance disparities is defined as:

$$L_{disp} = \frac{1}{|O|} \sum_{p \in O} L_{1;smooth}(\hat{D}'_i(p) - D'_i(p)), \quad (3)$$

$$D'_i(p) = \frac{D_i(p)}{\max(w^l, w^r)} W, \quad (4)$$

where  $\hat{D}'_i(p)$  is the predicted instance disparity for point  $p$ ,  $D'_i(p)$  is the instance disparity ground-truth,  $w^l$  and  $w^r$  represent the widths of 2D bounding boxes in two views, and  $|O|$  means the number of pixels belonging to the object  $O$ .

Once the iDispNet outputs instance disparity  $\hat{D}'_i(p)$ , we can compute the 3D location for each pixel  $p$  belonging to the foreground as the input of the following 3D detector. The 3D coordinate  $(X, Y, Z)$  is derived as follows:

$$X = \frac{(u_p - c_u)}{f_u} Z, \quad Y = \frac{(v_p - c_v)}{f_v} Z, \\ Z = \frac{B f_u}{\hat{D}'_i(p) + b^l - b^r},$$

where  $B$  is the baseline length between the left and right cameras,  $(c_u, c_v)$  is the pixel location corresponding to the camera center, and  $(f_u, f_v)$  are horizontal and vertical focal lengths, respectively.

### 3.3. Pseudo Ground-truth Generation

Training stereo matching network requires a large amount of dense disparity ground-truth, while most of the 3D object detection datasets [9, 2, 28] don't provide this

data due to its difficulties in the manual annotation. The full-frame disparity estimation module used in the recent works [30, 33] is first pre-trained on synthetic datasets and later fine-tuned on the real data with sparse disparity ground-truth converted from LiDAR points. Although the detection performance gained large improvements from this supervision, the requirement for LiDAR point cloud limits the scaling capability of stereo 3D object detection methods in the real world scenario due to the high sensor price.

Benefiting from the design of the iDispNet which only requires foreground supervision, we propose an effective way to generate a large amount of dense disparity pseudo-ground-truth (pesudo-GT) for the real data without the need of LiDAR points. The generation process is made possible by a category-specific shape prior model, from which the object shape can be reconstructed and later rendered to the image plane to obtain dense disparity ground-truth.

We use the volumetric Truncated Signed Distance Function (TSDF) as the shape representation. For some rigid object categories with relatively small shape variations (e.g. vehicles), the TSDF shape space for this category can be approximated by a low-dimensional subspace [13, 8]. Formally, denoting the basis of the subspace as  $V$ , which are obtained from the leading principal components of training shapes, and the mean shape as  $\mu$ , the shape  $\tilde{\phi}$  of an instance can be represented as:

$$\tilde{\phi}(z) = Vz + \mu, \quad (5)$$

where  $z \in \mathbb{R}^K$  is the shape coefficients and  $K$  is the dimension of the subspace.

Given the 3D bounding box ground-truth and the point cloud of an instance, we can reconstruct shape coefficients  $z$  for an instance by minimizing the following cost function:

$$L_{pc}(z) = \frac{1}{|P|} \sum_{x \in P} \phi(x, z)^2, \quad (6)$$

where  $\phi(x, z)$  is the interpolated value of a 3D point  $x$  in the TSDF volume defined by shape coefficients  $z$ ,  $P$  is the point cloud corresponding to the instance, and  $|P|$  is the number of points in the point cloud. Only  $z$  is being updated through the optimization process. Intuitively, this cost function minimizes the distance from the point cloud to the object surface defined by the zero crossing of the TSDF. The point cloud can be obtained from an off-the-shelf disparity estimation module or optionally LiDAR points.

Since the cost function above does not restrict the 3D dimension of object shape, we propose the following dimension regularization term to reduce the occurrence of objects overflowing the 3D bounding box:

$$L_{dim}(z) = \sum_{v \in V^{out}} \max(-\phi(v, z), 0)^2, \quad (7)$$

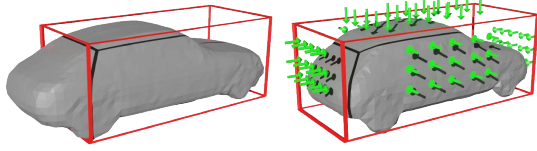


Figure 4. **The dimension regularization during Pseudo-GT generation** penalizes a voxel if it is outside of the 3D bounding box and has a negative TSDF value, thus enforcing the shape surface to stay inside the 3D bounding box. From left to right: object shapes without and with dimension regularization.

where  $V^{out}$  represents all the voxels that are defined outside of the 3D bounding box in a volume. A visualization of the dimension regularization is shown in Fig. 4.

To restrict the shape coefficients in an appropriate range, the following regularization term is used to penalize deviations of optimized shape from mean shape:

$$L_z(z) = \sum_{k=1}^K \left( \frac{z_k}{\sigma_k} \right)^2, \quad (8)$$

where  $\sigma_k$  is the  $k$ -th eigen value corresponding to the  $k$ -th principal component.

Combining the above terms, the total cost function is

$$L(z) = w_1 L_{pc}(z) + w_2 L_{dim}(z) + w_3 L_z(z). \quad (9)$$

Finally, instance disparity pseudo-GT  $D_i$  can be rendered based on the optimized object shape as follows:

$$D_i = \frac{Bf_u}{\pi(M(\tilde{\phi}(z)))} - (b^l - b^r), \quad (10)$$

where  $M$  represents the marching cubes [16] operation that converts the TSDF volume to a triangle mesh.  $\pi$  represents the mesh renderer that produces the pixel-wise depth map. Some examples of the rendered disparity pseudo-GT are visualized in the third line of Fig. 5.

### 3.4. Discussion

**Choices on network design.** There are two choices for the iDispNet design: **(1)** Using only the decoder part of the iDispNet as a prediction head similar to the mask head in Mask R-CNN. The RoI feature extracted from the backbone is reused in disparity estimation and the disparity head is trained end-to-end with the rest of the network; **(2)** Crop the RoI images from the original images, and then feed the cropped images to the encoder-decoder network of iDispNet. As shown in the Table 3 in the experiment section, the result of **(1)** is suboptimal compared to **(2)**, so we choose **(2)** as the proposed design. We believe the reason behind this result is related to the different requirements between

the tasks of instance segmentation and disparity estimation. Disparity estimation requires more fine-grained distinctive feature representation to make pixel-wise cost volume processing to be accurate, while instance segmentation is supervised to predict the same class probability for every pixel that belongs to the object. By jointly training the end-to-end version of the network, the backbone has to balance between these two different tasks and thus causes the sub-optimal result.

### Choices on the point cloud for Pseudo-GT generation.

In general, there are two choices of point cloud usage in the shape optimization process. The point cloud can be obtained from **(1)** the sparse LiDAR point clouds in the dataset with an optional depth completion step to improve density; **(2)** the prediction of an off-the-shelf disparity estimation network trained on other datasets (e.g. PSMNet trained on KITTI Stereo). **(1)** potentially gives a more accurate point cloud. But for datasets or application scenarios without the LiDAR points as optimization target in  $L_{pc}(z)$ , **(2)** is the only choice. We evaluate and present the results using both ways separately (titled by *Ours (velo)* and *Ours* relatively in Tab. 1 and 2). As later demonstrated in the results, **(2)** performs reasonably well without the usage of the LiDAR point cloud.

### 3.5. Implementation Details

**iDispNet.** Following the setting in [30], we use PSMNet [4] as the architecture for iDispNet. RoI images are cropped and resized to  $224 \times 224$  as the input. During stereo matching, we set the minimum and maximum instance disparities search range to -48 and 48 pixels, which cover 90% of the cases according to the statistics for the disparity distribution across the training set.

**3D detection network.** PointRCNN [27] is used as the 3D object detector in our implementation. Different from inputting point clouds of the entire scene in the conventional approach, we use the instance point cloud converted from instance disparity as the input to PointRCNN. The number of input point cloud subsamples is reduced to 768.

**Pseudo-GT generation.** To increase the stability of the pseudo-GT generation process, only points that sit inside of the ground-truth 3D bounding box are used for optimization. For objects with less than 10 points, the mean shape is directly used without further optimization. Following [8], we select the first five PCA components and set the volume dimension to  $60 \times 40 \times 60$ . The training shapes are obtained from [8], which are 3D models collected from the Google Warehouse website. During optimization, loss weights are set as  $w_1 = 10/3, w_2 = w_3 = 1$ . The optimization is achieved by a Levenberg–Marquardt solver implemented with Ceres [1].

Method	LiDAR Supervision	$AP_{bev}$ (IoU=0.7)			$AP_{3d}$ (IoU=0.7)			$AP_{bev}$ (IoU=0.5)			$AP_{3d}$ (IoU=0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
TL-Net [25]	N	29.22	21.88	18.83	18.15	14.26	13.72	62.46	45.99	41.92	59.51	43.71	37.99
S-RCNN [14]	N	68.50	48.30	41.47	54.11	36.69	31.07	87.13	74.11	58.93	85.84	66.28	57.24
PL (AVOD)	N	60.7	39.2	37.0	40.0	27.4	25.3	76.8	65.1	56.6	75.6	57.9	49.3
Ours	N	<b>76.51</b>	<b>58.63</b>	<b>50.26</b>	<b>63.57</b>	<b>47.15</b>	<b>39.73</b>	<b>90.60</b>	<b>80.53</b>	<b>71.16</b>	<b>90.38</b>	<b>79.77</b>	<b>69.81</b>
PL* (FP)	Y	72.8	51.8	44.0	59.4	39.8	33.5	89.8	77.6	68.2	89.5	75.5	66.3
PL* (AVOD)	Y	74.9	56.8	49.0	61.9	45.3	39.0	89.0	77.5	68.7	88.5	76.4	61.2
PL* (P-RCNN)	Y	73.4	56.0	<b>52.7</b>	62.3	44.9	<b>41.6</b>	88.4	76.6	69.0	88.0	73.7	67.8
Ours (velo)	Y	<b>77.63</b>	<b>64.38</b>	50.68	<b>64.29</b>	<b>47.73</b>	40.11	<b>90.67</b>	<b>80.45</b>	<b>71.03</b>	<b>90.47</b>	<b>79.76</b>	<b>69.71</b>
OC-Stereo	Y	77.66	65.95	51.20	64.07	48.34	40.39	90.01	80.63	71.06	89.65	80.03	70.34
ZoomNet	-	78.68	66.19	57.60	62.96	50.47	43.63	90.62	88.40	71.44	90.44	79.82	70.47
PL++ (P-RCNN)	Y	82.0	64.0	57.3	67.9	50.1	45.3	89.8	83.8	77.5	89.7	78.6	75.1

Table 1. **3D object detection results on the KITTI object validation set.** We report average precision of bird’s eye view ( $AP_{bev}$ ) and 3D boxes ( $AP_{3d}$ ) for the **car** category. LiDAR supervision indicates if the method uses the sparse LiDAR point cloud as a supervision signal during training. We report the reproduced result for PL (AVOD) since [30] didn’t provide full results on experiments without LiDAR supervision. Besides published state-of-the-art methods, we also present the results of concurrent works (grey background) for comparison.

Method	$AP_{bev}$ (IoU=0.7)			$AP_{3d}$ (IoU=0.7)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
S-RCNN	61.67	43.87	36.44	49.23	34.05	28.39
PL* (FP)	55.0	38.7	32.9	39.7	26.7	22.3
PL* (AVOD)	66.83	47.20	40.30	55.40	37.17	31.37
Ours	73.82	<b>52.34</b>	43.64	58.53	37.91	31.93
Ours (velo)	<b>74.07</b>	<b>52.34</b>	<b>43.77</b>	<b>59.58</b>	<b>39.34</b>	<b>31.99</b>
ZoomNet	72.94	54.91	44.14	55.98	38.64	30.97
OC-Stereo	68.89	51.47	42.97	55.15	37.60	30.25
PL++	75.5	57.2	53.4	60.4	44.6	38.5

Table 2. **3D object detection results on the KITTI object test set.** We report Average Precision of bird’s eye view ( $AP_{bev}$ ) and 3D boxes ( $AP_{3d}$ ) for **car** category. *Ours (velo)* and *Ours* indicates our method that uses and does not uses the sparse LiDAR point cloud as a supervision, respectively. Besides published state-of-the-art methods, we also present the results of concurrent works (grey background) for comparison.

**Training strategy.** We train the Stereo Mask R-CNN for 20 epochs with a weight decay of 0.0005, the iDispNet for 100 epochs with a weight decay of 0.01 and the PointRCNN 360 epochs with a weight decay of 0.0005. The learning rate is first warmed up to 0.01 and then decreases slowly in all the training processes.

## 4. Experiments

We evaluate the proposed approach on the 3D object detection benchmark of KITTI dataset [9]. First, we compare our method to state-of-the-art methods on the KITTI object detection benchmark in Sec. 4.1. Next, we conduct ablation studies to analyze the effectiveness of different components of the proposed method in Sec. 4.2. Then, we report the

running time of our method in Sec. 4.3. Finally, we provide some failure cases of our method in Sec. 4.4.

### 4.1. 3D Object Detection on KITTI

The KITTI object detection benchmark contains 7481 training images and 7518 testing images. To evaluate on the training set, we divide it into the training split and the validation split with 3712 and 3769 images following [5], respectively. Objects are divided into three levels: easy, moderate and hard, depending on their 2D bounding box sizes, occlusion, and truncation extent following the KITTI settings.

**Evaluation of 3D object detection.** We evaluate our method and compare it to previous state-of-the-art methods on the KITTI object 3D detection benchmark [9]. We perform the evaluation using Average Precision (AP) for 3D detection and bird’s eye view detection.

In Tab. 1, we compare our method with previous state-of-the-art methods on the validation split using 0.7 and 0.5 as the IoU threshold.

PL [30] estimates full-frame disparities, while our iDispNet predicts disparities only for pixels on objects. When LiDAR supervision is not used at training time, our method outperforms PL (AVOD) over 10% AP in all metrics. Specifically, our method gains over 23.57% improvement for  $AP_{bev}$  in the easy level with an IoU threshold of 0.7. This huge improvement comes from the pseudo-GT generation, which can provide a large amount of training data even if LiDAR ground-truth is not available at training time.

When LiDAR supervision is used at training time, our method still outperforms previous state-of-the-art methods in most of the metrics. PL\* (P-RCNN) and ours share the same 3D detector, but our method still obtains better results.

Method	GT	Pixel-wise		Object-wise	
		Disparity	Depth	Disparity	Depth
PSMNet	PGT	1.53	0.54	0.87	1.00
Ours (e2e)	PGT	1.22	0.41	0.76	0.86
Ours	PGT	<b>0.90</b>	<b>0.28</b>	<b>0.38</b>	<b>0.33</b>
PSMNet	LiDAR	1.01	0.64	1.27	1.28
GANet	LiDAR	<b>0.89</b>	0.63	<b>1.23</b>	1.24
Ours	LiDAR	1.32	<b>0.60</b>	1.27	<b>1.06</b>

Table 3. **Disparity EPE and Depth RMSE comparison**, evaluated on the KITTI *validation set*. We use our disparity pseudo-GT and sparse LiDAR as ground-truth for evaluation, denoted by PGT and LiDAR respectively.

Specifically, our method gains an 8.38% improvement in  $AP_{bev}$  at the moderate level with an IoU threshold of 0.7. The reason is that our iDispNet focuses on the foreground regions and we have much denser training data via the object shape rendering.

Tab. 2 compares our method with previous state-of-the-art methods and several concurrent works on the KITTI test set with an IoU threshold of 0.7. Comparing with previous methods, our method achieves the state-of-the-art performance in all metrics. Specifically, our method gains 7% and 5% improvement in  $AP_{bev}$  at the easy and moderate levels, respectively, and 4% improvement in  $AP_{3d}$  at the easy level, comparing to the previous state-of-the-art PL\* (AVOD). Among concurrent works, OC-Stereo [21] and ZoomNet [32] share a similar idea with ours. OC-Stereo utilizes LiDAR points after completion as supervision, and ZoomNet introduces fine-grained annotations to generate the ground-truth. Instead, our pseudo-GT is rendered from the optimized object shape, which is more accurate than OC-Stereo and more efficient than ZoomNet, and thus leads to better performance on the KITTI test set. More remarkably, our method achieves the state-of-the-art performance even if LiDAR supervision is not used at training time, which further shows that our method is robust and applicable in real-world applications.

We visualize some qualitative results of object detection, instance disparity estimation, and disparity pseudo-GT in Fig. 5.

## 4.2. Ablation Studies

In this section, we conduct extensive ablation experiments to analyze the effectiveness of different components in our method.

**Cost function for shape optimization.** To measure the effectiveness of the dimension regularization in the shape optimization process, we perform optimization processes with and without dimension regularization, and then compute the percentage of objects that have more than 70% vertices lo-

Method	S-RCNN	PL (AVOD)	PL (PRCNN)	PL (FP)	Ours
Time (s)	0.417	0.51	0.51	0.67	0.425

Table 4. **Running time comparison.** S-RCNN represents Stereo R-CNN [14].

cating inside the 3D bounding box. Our experiments show that the use of dimension regularization makes the percentage above rise from 71% to 82%, which proves that considering dimension regularization can reduce the occurrence of shape overflowing the 3D bounding box, thereby improving the quality of the object shape and the pseudo-GT.

**Instance disparity estimation.** To validate the benefit of instance disparity estimation, we compute the disparity end-point-error (EPE) and depth RMSE for our iDispNet and some full-frame deep stereo networks in the foreground area.

In addition to the pixel-wise error, we also calculate the object-wise error, which is defined as the average error within each instance, and then averaged among instances. We believe that the object-wise error is more suitable to reflect the quality of disparity estimation for each object because the pixel-wise error is dominated by objects with large areas.

The results are in Tab. 3. We use the pseudo-GT and sparse LiDAR as ground-truth separately, denoted by PGT and LiDAR. PSMNet and GANet are trained on the KITTI Stereo dataset, while our iDispNet is trained with the pseudo-GT. With the pseudo-GT as ground-truth, our iDispNet reaches smaller disparity and depth errors than the full-frame PSMNet by a large margin. With sparse LiDAR points as ground-truth, our iDispNet still performs better than the full-frame method PSMNet and the state-of-the-art deep stereo method GA-Net [34], especially for the object-wise depth RMSE error.

Comparing the second and third lines in Tab. 3 shows that re-using the features extracted from the RPN limits the quality of estimated disparity maps, which leading the end-to-end version of the iDispNet to give sub-optimal results, so we don't report results of the end-to-end version in other experiments.

Some qualitative results of instance disparity estimation and the comparison against the full-frame disparity estimation are shown in Fig. 6. The full-frame PSMNet cannot capture the smooth surfaces and sharp edges of vehicles, thus leading the following 3D detector to struggle to predict correct bounding boxes from inaccurate point clouds. In contrast, our iDispNet gives more accurate and stable predictions thanks to instance disparity estimation and the supervision from the disparity pseudo-GT.

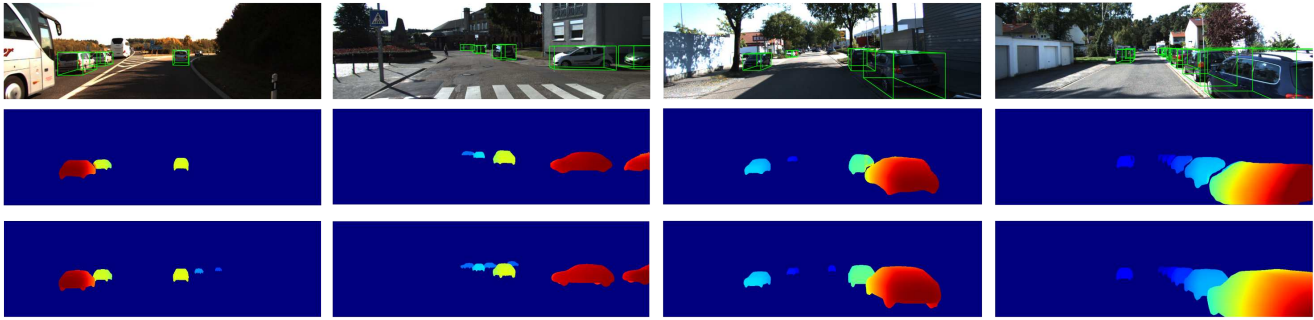


Figure 5. **Qualitative results.** The rows from top to bottom present 3D bounding box prediction, instance disparity estimation and our disparity pseudo-ground-truth, respectively.

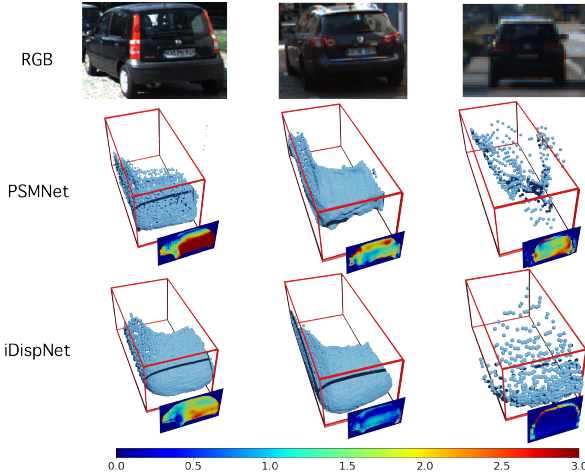


Figure 6. **Qualitative comparison of disparity estimation results between PSMNet and our iDispNet.** 3D ground-truth bounding boxes are shown in red. Disparity error maps are shown as well, where the larger value indicates the worse disparity.

### 4.3. Running Time

Tab. 4 shows the running time comparison of our method and other stereo methods. Our method takes 0.425s at inference time, surpassing almost all prior stereo methods. Specifically, our method takes 0.17s for the 2D detection and segmentation, 0.13s for the instance disparity estimation, and 0.125s for the 3D detection from the point cloud. The efficiency is attributed to estimating only the disparity in RoIs and only the 3D bounding boxes from the instance point clouds, which greatly reduces the search space.

### 4.4. Failure Cases

We visualize some failure cases in Fig. 7. Our 3D object detection method is most likely to fail on objects that are too far away as shown in Fig. 7(a), or under strong occlusion or truncation as shown in Fig. 7(b). The reason is that there are too few 3D points on these objects for the detector to predict the correct bounding boxes. Our pseudo-GT generation is most likely to fail on objects with unusual shapes, such as the car in Fig. 7(c) which is much shorter than other cars. Since there are very few examples with this kind of shape in

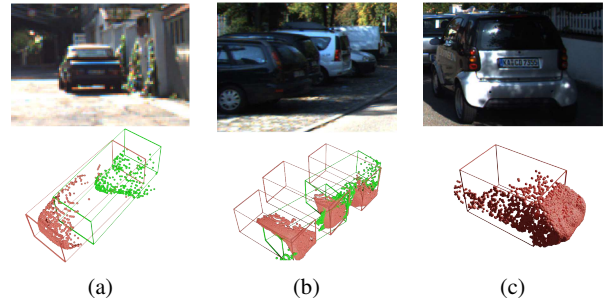


Figure 7. **Failure cases.** The ground-truth bounding boxes and the pseudo-GT point clouds are visualized in red, while the predictions are visualized in green.

the CAD model training set, so it is difficult to reconstruct this type of cars with the statistical shape model.

## 5. Conclusion

In this paper, we proposed a novel approach for 3D object detection from stereo images. The key idea is to estimate instance-level pixel-wise disparities only in detected 2D bounding boxes and detect objects based on the instance point clouds converted from the instance disparities. To solve the scarcity and sparsity of the training data, we proposed to integrate shape prior learned from CAD models to generate pseudo-GT disparity as supervision. Experiments on the 3D detection benchmark of the KITTI dataset showed that our proposed method outperformed state-of-the-art methods by a large margin, especially when LiDAR supervision was not available at training time. We believe that the proposed approach is also applicable to other object categories, e.g., pedestrians and cyclists, whose shapes can be reconstructed similarly by fitting a statistical shape model (e.g., SMPL [15]) to point cloud data, as demonstrated by the PedX dataset [12].

**Acknowledgements:** The authors would like to acknowledge support from NSFC (No. 61806176), Fundamental Research Funds for the Central Universities (2019XZZX004-09) and ZJU-SenseTime Joint Lab of 3D Vision.



## References

- [1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. <http://ceres-solver.org>. 5
- [2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 4
- [3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2
- [4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 1, 5
- [5] Xiaozi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015. 6
- [6] Xiaozi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2
- [7] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2
- [8] Francis Engelmann, Jörg Stückler, and Bastian Leibe. Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In *German Conference on Pattern Recognition*, pages 219–230. Springer, 2016. 2, 4, 5
- [9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 2, 4, 6
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3
- [11] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017. 1
- [12] Wonhui Kim, Manikandasriram Srinivasan Ramanagopal, Charles Barto, Ming-Yuan Yu, Karl Rosaen, Nick Goumas, Ram Vasudevan, and Matthew Johnson-Roberson. Pedx: Benchmark dataset for metric 3-d pose estimation of pedestrians in complex urban intersections. *IEEE Robotics and Automation Letters*, 4(2):1940–1947, 2019. 3, 8
- [13] Michael E Leventon, W Eric L Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *5th IEEE EMBS International Summer School on Biomedical Imaging, 2002.*, pages 8–pp. IEEE, 2002. 2, 4
- [14] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019. 2, 3, 6, 7
- [15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 3, 8
- [16] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 5
- [17] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2069–2078, 2019. 3
- [18] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 2
- [19] 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
- [20] J Krishna Murthy, Sarthak Sharma, and K Madhava Krishna. Shape priors for real-time monocular object localization in dynamic environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1768–1774. IEEE, 2017. 2
- [21] Alex D Pon, Jason Ku, Chengyao Li, and Steven L Waslander. Object-centric stereo matching for 3d object detection. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 2, 7
- [22] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 2
- [23] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 2
- [24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2
- [25] Zengyi Qin, Jinglu Wang, and Yan Lu. Triangulation learning network: from monocular to stereo 3d object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7607–7615. IEEE, 2019. 6
- [26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2
- [27] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointnet: 3d object proposal generation and detection from point

- cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. [2](#), [5](#)
- [28] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2019. [4](#)
- [29] R. Wang, N. Yang, J. Stueckler, and D. Cremers. Direct-shape: Photometric alignment of shape priors for visual vehicle pose and shape estimation. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. [2](#)
- [30] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#)
- [31] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018. [1](#), [2](#)
- [32] Zhenbo Xu, Wei Zhang, Xiaoqing Ye, Xiao Tan, Wei Yang, Shilei Wen, Errui Ding, Ajin Meng, and Liusheng Huang. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. [2](#), [7](#)
- [33] 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. In *International Conference on Learning Representations*, 2020. [2](#), [4](#)
- [34] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. [1](#), [7](#)