Shape Prior Guided Instance Disparity Estimation for 3D Object Detection

Linghao Chen*, Jiaming Sun*, Yiming Xie, Siyu Zhang, Qing Shuai,
Qinhong Jiang, Guofeng Zhang, Hujun Bao, Xiaowei Zhou

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 point clouds 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, when LiDAR ground-truth is not used at training time, Disp R-CNN outperforms previous state-of-the-art methods based on stereo input by 20% in terms of average precision for all categories. The code and pseudo-ground-truth data are available at the project page: https://github.com/zju3dv/disprcnn.

Index Terms—Autonomous Driving, 3D Detection, Stereo Matching.

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 [1], [2], [3] tackled the stereo correspondence matching problem with Convolutional Neural Networks (CNNs) and achieved impressive results. Taking an estimated disparity map as the input, 3D object detection methods [4], [5] 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 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.

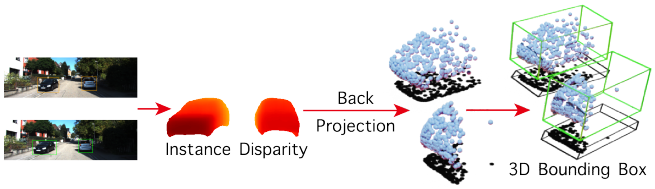


Fig. 1. **The proposed system estimates an instance disparity map,** i.e., pixel-wise disparities only on foreground objects, for stereo 3D object detection. This design leads to better disparity estimation accuracy and faster run-time speed.

Another limitation of the full-frame disparity estimation is the lack of pixel-wise ground-truth annotation. In the KITTI dataset [6] 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 [7], there is no such ground-truth provided in the KITTI Object Detection benchmark due to

- *The first two authors contributed equally.*
- *Linghao Chen, Yiming Xie, Qing Shuai, Guofeng Zhang, Hujun Bao and Xiaowei Zhou are affiliated with the State Key Lab of CAD&CG, Zhejiang University, China.*
- *Jiaming Sun, Siyu Zhang and Qinhong Jiang are with SenseTime.*
- *Corresponding author: Xiaowei Zhou.*

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 statistical shape model under several geometric constraints [8], [9]. 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 [10].

Different from some recent methods [5], [11], [12] 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 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.

This article is the extension of our conference work [13]. In the conference version, we performed the experiments only on the car category in the KITTI dataset. Here, we extend our method to deal with the more challenging pedestrian and cyclist categories. Pedestrians and cyclists tend to have thinner structures than cars. Moreover, human bodies are non-rigid and more difficult to reconstruct. Thus, we propose to fit a statistical human body model (SMPL [14]) to point clouds and 2D keypoints in images, yielding 3D mesh models of pedestrians and cyclists, and then perform the proposed Disp R-CNN pipeline for the pedestrians and cyclists in the KITTI dataset. The proposed approach achieves state-of-the-art performance for pedestrians and cyclists, proving that our method can be generalized to multiple categories.

In summary, our contributions are as follows:

- We propose 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 for all categories of the KITTI dataset.
- We propose 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.
- We release our fitted 3D models of cars, pedestrians, and cyclists on the KITTI dataset, which provide richer information than original bounding box annotations. The code and data are available at https://github.com/zju3dv/disprcnn.

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 [15] 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, [16] proposes to estimate 3D bounding boxes with relation and constraints between 2D and 3D bounding boxes. [4] uses a depth map as an extra input channel to assist 3D object detection. Recently, Pseudo-LiDAR [5] 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 [17] and ZoomNet [18] that propose the similar idea of instance-level disparity estimation. OC-Stereo [17] uses depth completion results from sparse LiDAR points as object-centric disparity supervision, and ZoomNet [18] 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) [19], [20] as input. F-PointNet [21] segments the object point cloud within the 2D RoI frustum into foreground and background and later predicts 3D bounding boxes with PointNet++ [22]. Recently, PointRCNN [23] adapts this framework into a two-stage design as in the 2D object detection counterpart [24] 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. [9] 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. [25] further extents this pipeline with the temporal kinematic constraints of objects in dynamic scenes. [26] proposes a continuous optimization approach to jointly optimize object shape and pose with the photometric error. [27] 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. humans, SMPLify [28] proposes to reconstruct the human body by fitting the SMPL [14] model to 2D keypoints. [29] further utilizes silhouette consistency to constraint the SMPL parameters. The PedX dataset [30] proposes to fit the SMPL model to point cloud data, further improving the reconstruction quality.

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.

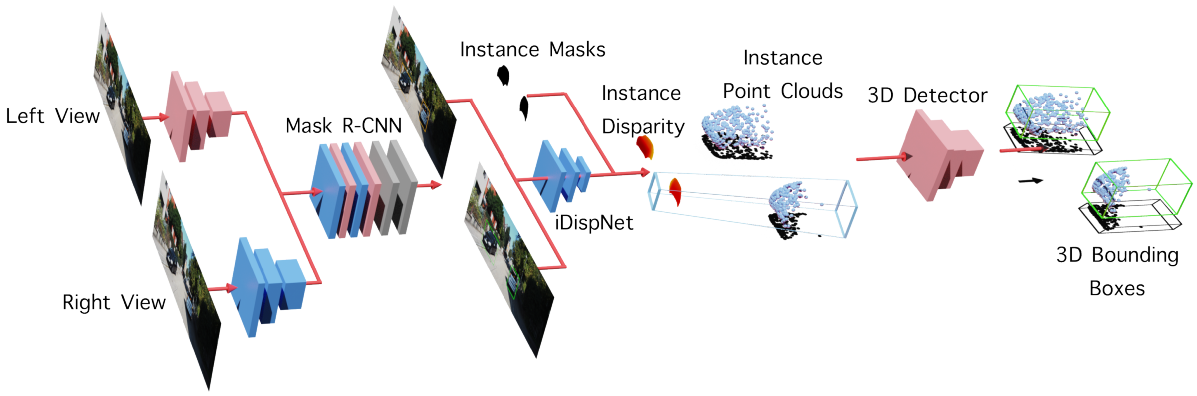


Fig. 2. **Disp R-CNN Architecture.** Disp R-CNN has three stages. First, the input images are passed through a Mask R-CNN to detect 2D bounding boxes and instance segmentation masks. Then, the instance disparity estimation network (iDispNet) takes the cropped RoI images as input and estimates an instance 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.

3.1 2D Detection and Association

We start by briefly describing the base 2D detector that provides necessary input for the following modules of the pipeline. Previous work [15] designed a stereo variant version of Region Proposal Network (Stereo RPN) to generate proposals from the same set of anchors. However, the Stereo RPN requires objects in different views to have considerable overlaps, thus cannot work well for objects with thin structures, such as pedestrians and cyclists.

To this end, we propose to detect 2D objects in the left and right views separately and then perform association. We use the Mask R-CNN [31] as our 2D detector. Mask R-CNN is composed of two stages. The first stage is a Region Proposal Network (RPN) that generates object proposals from a collection of anchors. The second stage extracts image features using RoIAlign, followed by prediction heads that produce 2D bounding boxes, classification scores, and instance segmentation masks.

After 2D objects in the left and right views are detected, we compute the Structural SIMilarity index (SSIM) [32] between left and right RoIs to perform association. More information about this process is given in Sec. 3.5.

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 [5] 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.

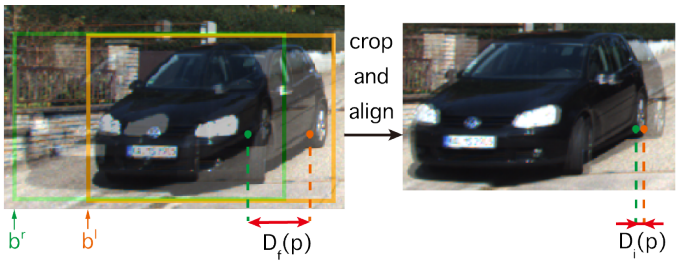


Fig. 3. **The crop-and-align process** aligns the left and right RoIs by cutting off a *global offset*. As a result, the instance disparity $D_i(p)$ distributes in a much narrower range compared to the full-frame disparity $D_f(p)$, which makes it possible to reduce the disparity search range when constructing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

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

$$D_f(p) = u_p^l - u_p^r, \tag{1}$$

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 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:

$$D_i(p) = D_f(p) - (b^l - b^r), \tag{2}$$

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:

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

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

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:

$$X = \frac{(u_p - c_u)}{f_u} Z, \quad Y = \frac{(v_p - c_v)}{f_v} Z, \tag{5}$$

$$Z = \frac{B f_u}{\hat{D}_i(p) + b^l - b^r}, \tag{6}$$

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 [6], [33], [34] don't provide this data due to its difficulties in the manual annotation.

The full-frame disparity estimation module used in the recent works [5], [11] 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 (pseudo-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.

3.3.1 Rigid objects

For some rigid object categories with relatively small shape variations (e.g. vehicles), we use the volumetric Truncated Signed Distance Function (TSDF) as the shape representation, and approximate the TSDF shape space by a low-dimensional subspace [8], [9]. 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 μ, the shape $\tilde{\phi}$ of an instance can be represented as:

$$\tilde{\phi}(z) = Vz + \mu, \tag{7}$$

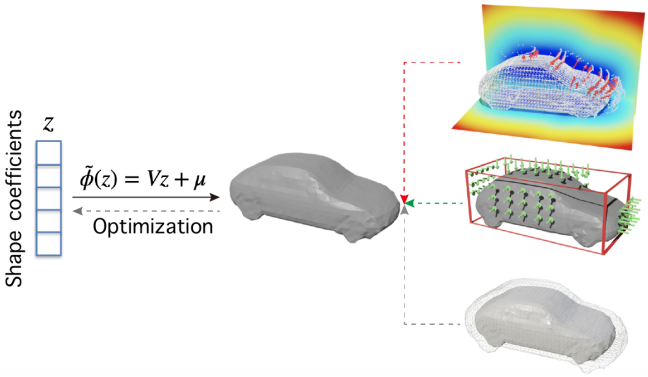


Fig. 4. **The object shape reconstruction process for the car category.**

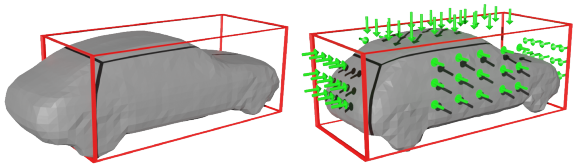


Fig. 5. **The dimension regularization during pseudo-GT generation** penalizes a voxel if it is outside of the 3D bounding box and has a negative TSDF value, thus enforcing the shape surface to stay inside the 3D bounding box. From left to right: object shapes without and with dimension regularization.

where $z \in \mathbb{R}^K$ 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:

$$L_{pc}(z) = \frac{1}{|P|} \sum_{x \in P} \phi(x, z)^2, \tag{8}$$

where $\phi(x, z)$ is the trilinear 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 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:

$$L_{dim}(z) = \sum_{v \in V^{out}} \max(-\phi(v, z), 0)^2, \tag{9}$$

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. 5.

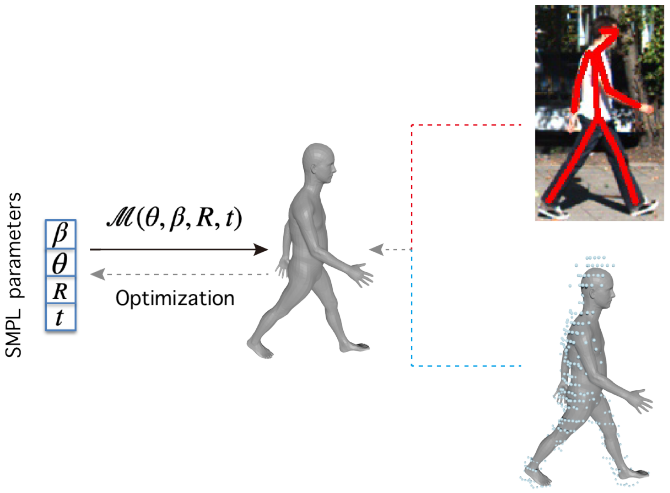


Fig. 6. **The human body is reconstructed by jointly fitting the SMPL model to point clouds and minimizing the reprojection error of 2D joints.**

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

$$L_z(z) = \sum_{k=1}^{K} (\frac{z_k}{\sigma_k})^2, \tag{10}$$

where σ_k is the k-th eigen value corresponding to the k-th principal component.

Combining the above terms, the total cost function is

$$L(z) = w_1 L_{pc}(z) + w_2 L_{dim}(z) + w_3 L_z(z). \tag{11}$$

The overall pipeline for reconstructing cars is visualized in Fig. 4.

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

$$D_i = \frac{B f_u}{\pi(M(\tilde{\phi}(z)))} - (b^l - b^r), \tag{12}$$

where M represents the marching cubes [35] operation that converts the TSDF volume to a triangle mesh. π 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. 7.

3.3.2 Non-rigid human bodies

For some non-rigid object categories such as human bodies, we propose to use the SMPL [14] model as the category-level shape prior to perform the reconstruction. The reconstruction pipeline is visualized in Fig. 6.

The SMPL model provides a function $\mathcal{M}(\theta, \beta, R, t)$ that takes as input the pose parameters θ, the shape parameters β, the rotation parameters R and the translation parameters t, and returns the body mesh $M \in \mathbb{R}^{N\times 3}$, and body joints $X \in \mathbb{R}^{k\times 3}$.

Given an image with 2D keypoints $x \in \mathbb{R}^{k\times 2}$, point clouds $P \in \mathbb{R}^{N\times 3}$, we aim to perform the reconstruction by fitting the SMPL model to point clouds and minimizing the reprojection error of 3D and 2D keypoints simultaneously. The corresponding cost functions are defined as follows:

$$L_{kpts}(\theta, \beta, R, t) = \sum_{i=1}^{k} (\pi(X_i) - x_i)^2 \tag{13}$$

$$L_{pc}(\theta, \beta, R, t) = \sum_{j\in C_{cor}} (M_j - P_j)^2, \tag{14}$$

where π represents projection operation, and C_{cor} is the set of corresponding indices between mesh vertices and point clouds, found by the nearest neighbor between their projections on the image plane.

Combining the above terms produces the following cost function:

$$L(\theta, \beta, R, t) = w_1 L_{kpts} + w_2 L_{pc} \tag{15}$$

After optimization, instance disparity pseudo-GT D_i can be rendered as follows:

$$D_i = \frac{B f_u}{\pi(M(\theta, \beta, R, t))} - (b^l - b^r), \tag{16}$$

where π represents the mesh renderer that produces the pixel-wise depth map, and M is the body mesh.

Some examples of the rendered disparity pseudo-GT are visualized in the third line of Fig. 8.

3.4 Discussion

Choices on network design. There are two choices for the iDispNet design: **(1)** Use 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 Tab. 7 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 suboptimal 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 the KITTI-Stereo dataset). **(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, Tab. 2, Tab. 4 and 5). As later demonstrated in the results, **(2)** performs reasonably well without the usage of the LiDAR point cloud.

Choices on the cost volume search space. Some works [11], [36] build depth cost volume in the stereo matching network instead of traditional disparity cost volume [2], [5], based on the fact that uniformly sampling in the disparity space is equal to non-uniformly sampling in the depth space, resulting in sparser sampling for further pixels and decreased detection accuracy for further objects. However, we still choose disparity cost volume in our method as uniformly sampling in the disparity space on the resized RoIs has a similar effect as the depth cost volume, especially for further objects. As shown in Sec. 4, our method produces even better performance than methods using depth cost volume.

3.5 Implementation Details

2D Detection and Segmentation Network. We use Mask R-CNN [31] as the 2D detection and segmentation network. The segmentation mask head is supervised by the pseudo-ground-truth masks rendered from the object shape. To further encode the geometry information into the left-right bounding box association, we model the center disparity and the height of 2D bounding boxes by linear regression and reject the association between RoIs when the difference between the observed center disparity and expected center disparity is over three standard deviations.

iDispNet. Following the setting in [5], we use PSMNet [2] as the architecture for iDispNet. RoI images are cropped and resized to 224×224 as the input. During stereo matching, -48 to 48 pixels is set to be the instance disparities search range, which covers 98% of the cases according to the statistics for the disparity distribution across the training set.

3D detection network. PointRCNN [23] 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. We do not use GT-AUG described in the PointRCNN paper, as each sample is the point cloud of the entire scene in PointRCNN, while each sample is an instance point cloud in our method. Other settings of our network remain the same as the original PointRCNN.

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.

For the **car** category, we select the first five PCA components and set the volume dimension to $60 \times 40 \times 60$ following [9]. The training shapes are obtained from [9], 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 [37].

For the **pedestrian** category, we use an off-the-shelf keypoint detector HR-Net [38] to detect 2D keypoints. To ensure the reconstruction quality, we initialize SMPL parameters by performing inference using the pretrained SPIN [39] model. During optimization, loss weights are set as $w_1 = 1, w_2 = 0.05$.

For the **cyclist** category, we reconstruct the human body and bicycle separately and then combine them. We use the same method as the pedestrian category to reconstruct the human body, and manually select a CAD model for bicycles from ShapeNet [10] to represent all bicycles in the KITTI dataset. Since point clouds for the human body and bicycle cannot be separated trivially, we don't use point clouds for cyclists. Instead, we simply put the human body and the bicycle model at the top center and the bottom center of the 3D bounding box, respectively. Thus we don't differ 'Ours' and 'Ours (velo)' for the cyclist category in all experiments.

Training strategy. We train the 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 the KITTI dataset [6]. 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 [43], 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 [6]. 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 for the **car** category.

PL [5] 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 26.34% improvement for AP_{bev} in the moderate 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. Specifically, our method gains a 10.21% 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.

We also evaluate our method for the **pedestrian** and **cyclist** categories on the KITTI validation set using IoU=0.5. As shown

TABLE 1
3D object detection results for the car category on the KITTI object validation set. 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 [5] didn't provide full results on experiments without LiDAR supervision. The 'Ours' version is using PSMNet trained on the KITTI-Stereo dataset to generate pseudo-GT, 'Ours (velo)' is using LiDAR points to generate pseudo-GT, 'Ours (ms)' is using the mean shape for all objects to generate pseudo-GT, and 'Ours (dob)' is using PSMNet trained on the DrivingStereo dataset to generate pseudo-GT. We also report results that use segmentation mask annotations in the KINS dataset (grey background), where 'Ours (flb)' is training Disp-RCNN using fused LiDAR points instead of pseudo-GT, and 'Ours (vkb)' is training Disp-RCNN using the same disparity pseudo-GT as 'Ours (velo)'.

Method	LiDAR Supervision	KINS mask	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 [40]	N	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 [15]	N	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
IDA-3D [41]	N	N	70.68	50.21	42.93	54.97	37.45	32.23	88.05	76.69	67.29	87.08	74.57	60.01
PL (AVOD)	N	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 (ms)	N	N	75.15	62.90	49.64	59.40	44.95	38.12	90.49	80.37	71.15	89.72	79.03	69.79
Ours (dob)	N	N	76.15	63.33	55.28	60.20	44.78	42.46	90.21	86.91	70.90	89.79	78.82	69.90
Ours	N	N	**77.89**	**65.54**	**57.01**	**63.97**	**51.50**	**44.15**	**90.65**	**87.75**	**78.82**	**90.27**	**79.83**	**70.69**
PL* (FP)	Y	N	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	N	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	N	73.4	56.0	52.7	62.3	44.9	41.6	88.4	76.6	69.0	88.0	73.7	67.8
OC-Stereo	Y	N	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	-	N	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	N	**82.0**	64.0	57.3	67.9	50.1	45.3	89.8	83.8	77.5	89.7	78.6	**75.1**
Ours (velo)	Y	N	77.85	**66.21**	**57.78**	**69.92**	**53.81**	**46.70**	90.58	87.74	**78.86**	90.31	79.95	70.91
Ours (flb)	Y	Y	77.47	66.06	57.76	70.11	54.43	47.40	90.49	87.57	71.52	90.30	80.07	70.99
Ours (vkb)	Y	Y	83.29	66.18	57.60	70.18	54.72	46.99	97.20	87.92	71.71	90.33	87.06	71.15

TABLE 2
3D object detection results for the pedestrian category on the KITTI object validation set. LS stands for LiDAR supervision.

Method	LS	AP_{bev} (IoU=0.5)			AP_{3d} (IoU=0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
PSMNet+AVOD	Y	36.68	30.08	23.76	27.39	26.00	20.72
PL*(FP)	Y	41.3	34.9	30.1	33.8	27.4	24.0
OC-Stereo	Y	44.00	37.20	30.39	34.80	29.05	28.06
Ours	N	48.46	36.60	33.99	40.43	33.03	27.05
Ours (velo)	Y	**50.70**	**38.33**	**35.50**	**43.87**	**36.26**	**29.81**

TABLE 3
3D object detection results for the cyclist category on the KITTI object validation set.

Method	LS	AP_{bev} (IoU=0.5)			AP_{3d} (IoU=0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
PSMNet+AVOD	Y	36.12	22.99	22.11	35.88	22.78	21.94
PL*(FP)	Y	47.6	29.9	27.0	41.3	25.2	24.9
OC-Stereo	Y	48.2	27.9	26.96	45.59	25.93	24.62
Ours	N	**61.60**	**36.89**	**35.07**	**55.98**	**33.46**	**29.51**

in Tab. 2 and Tab. 3, our method outperforms the previous state-of-the-art method for all metrics. Note that even if LiDAR supervision is not available, our method still achieves comparable results.

Several works such as [15] and [18] didn't evaluate their methods on these two categories because their methods are specially designed for the car category. However, our method is not restricted by the category as long as an appropriate statistical shape prior model is available.

TABLE 4
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 the **car** category. 'Ours (velo)' uses the sparse LiDAR point cloud as supervision, while 'Ours' doesn't.

Method	AP_{bev} (IoU=0.7)			AP_{3d} (IoU=0.7)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
TLNet	13.71	7.69	6.73	7.64	4.37	3.74
S-RCNN	61.67	43.87	36.44	49.23	34.05	28.39
IDA-3D	61.87	42.47	34.59	45.09	29.32	23.13
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
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++	78.31	58.01	**51.25**	61.11	42.43	36.99
Ours	79.61	57.98	47.09	67.02	43.27	36.43
Ours (velo)	**79.76**	**58.62**	47.73	**68.21**	**45.78**	**37.73**

Tab. 4, Tab. 5, and Tab. 6 compare our method with previous state-of-the-art methods and several concurrent works on the KITTI test set. Comparing with previous methods, our method achieves state-of-the-art performance in many metrics. Specifically, for the **car** category, our method gains 12.93% and 11.42% improvement in AP_{bev} at the easy and moderate levels, respectively, and 12.81% improvement in AP_{3d} at the easy level, comparing to the previous state-of-the-art PL* (AVOD). Among concurrent works, OC-Stereo [17] and ZoomNet [18] share a similar idea with ours. OC-Stereo utilizes LiDAR points after

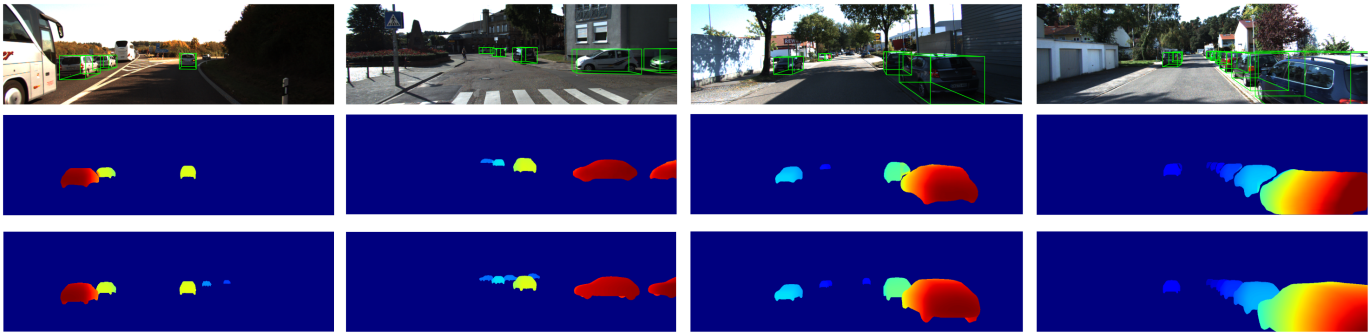


Fig. 7. **Qualitative results for the car category on the KITTI object validation set.** The rows from top to bottom present 3D bounding box prediction, instance disparity estimation, and our disparity pseudo-ground-truth, respectively.

TABLE 5
3D object detection results for the pedestrian category on the KITTI object test set.

Method	LS	AP_{bev} (IoU=0.5)			AP_{3d} (IoU=0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
RT3DStereo	N	4.72	3.65	3.00	3.28	2.45	2.35
PL*(FP)	Y	31.3	24.0	21.9	29.8	22.1	18.8
PL* (AVOD)	Y	27.5	20.6	19.4	25.2	19.0	15.3
OC-Stereo	Y	35.12	23.23	22.56	28.14	21.85	20.92
Ours	N	**42.72**	**29.12**	**25.09**	35.75	25.40	21.79
Ours (velo)	Y	40.21	28.34	24.46	**37.12**	**25.80**	**22.04**

TABLE 6
3D object detection results for the cyclist category on the KITTI object test set.

Method	AP_{bev} (IoU=0.5)			AP_{3d} (IoU=0.5)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
RT3DStereo [42]	7.03	4.10	3.88	5.29	3.37	2.57
PL*(FP)	4.1	3.1	2.8	3.7	2.8	2.1
PL* (AVOD)	13.5	9.1	9.1	13.3	9.1	9.1
OC-Stereo	34.77	22.26	21.36	32.66	21.25	19.77
Ours	**44.19**	**27.04**	**23.58**	**40.05**	**24.40**	**21.12**

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, leading to better performance on the KITTI test set. Compared to PL++ [11] which uses depth cost volume, our method still outperforms it for nearly all metrics. Specifically, our method outperforms PL++ [11] by 7.1% and 3.35% at the easy and moderate levels in AP_{3d}, respectively. This is because our iDispNet can predict more accurate disparity with the guide of object shape prior. More remarkably, our method achieves 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. For the **pedestrian** category, our method gains 15.22% improvement in AP_{bev} at the easy level comparing to previous state-of-the-art PL* (AVOD). For the **cyclist** category, our method gains 30.69% improvement in AP_{bev} at the easy level. Note that 'Ours' outperforms 'Ours (velo)' for the pedestrian

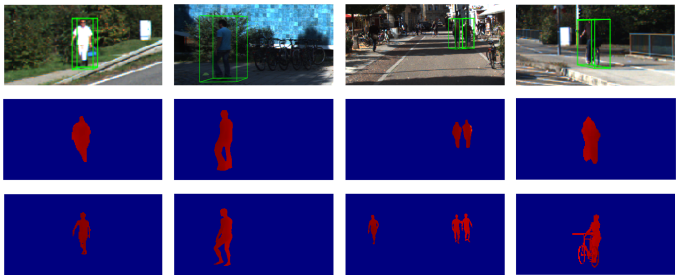


Fig. 8. **Qualitative results for pedestrian and cyclist categories on the KITTI object validation set.** The rows from top to bottom present 3D bounding box prediction, instance disparity estimation, and our disparity pseudo-ground-truth, respectively.

TABLE 7
Disparity EPE and Depth RMSE comparison, evaluated on the KITTI validation set for the car category. We use our disparity pseudo-GT and sparse LiDAR as ground-truth for evaluation, denoted by PGT and LiDAR respectively.

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	**0.90**	**0.28**	**0.38**	**0.33**
PSMNet	LiDAR	1.01	0.64	1.27	1.28
GANet	LiDAR	**0.89**	0.63	**1.23**	1.24
Ours	LiDAR	1.32	**0.60**	1.27	**1.06**

category on the test set by a small margin, especially for AP_{bev}. The reason is that the LiDAR points on the human body are very sparse and inaccurate.

We visualize some qualitative results of object detection, instance disparity estimation, and disparity pseudo-GT in Fig. 7 and Fig. 8.

4.2 Ablation Studies

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

2D Detection. To analyze the effectiveness of our association method in the 2D detection process, we compare our results and Stereo R-CNN [15] for the pedestrian category on the KITTI

TABLE 8
2D Detection AP, evaluated on the KITTI validation set for the pedestrian and cyclist categories using 0.5 as IoU threshold. We report the reproduced results for Stereo R-CNN since [15] does not perform experiments for the pedestrian and cyclist categories.

Method	Pedestrain			Cyclist		
	Easy	Mode.	Hard	Easy	Mode.	Hard
S-RCNN	58.90	51.43	44.59	65.73	46.92	45.16
Ours	**68.80**	**64.11**	**56.13**	**71.17**	**52.54**	**50.82**

validation set. As shown in Tab. 8, the proposed association method outperforms Stereo R-CNN by a large margin in terms of 2D detection AP. As claimed in Sec. 3.1, Stereo R-CNN is designed for wide objects, such as cars, thus misses lots of objects with thin structures, such as pedestrians and cyclists.

Dimension regularization 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 locating inside the 3D bounding box. Our experiments show that the use of dimension regularization makes the above percentage 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 shown in Tab. 7. 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 [3], especially for the object-wise depth RMSE error.

Comparing the second and third lines in Tab. 7 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. 9. 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.

TABLE 9
Running time comparison. S-RCNN represents Stereo R-CNN [15].

Method	S-RCNN	PL (AVOD)	PL (PRCNN)	PL (FP)	Ours
Time (ms)	417	510	510	670	**387**

The necessity of supervising the iDispNet using rendered disparity maps. To validate the necessity of supervising the iDispNet using rendered disparity maps, we fuse multi-frame LiDAR point clouds, produce denser disparity maps, and train the Disp R-CNN. We also use segmentation mask annotations provided by the KINS dataset [44] in this experiment, represented as 'Ours (flb)'. To make a fair comparison, we also train the Disp R-CNN using segmentation mask annotations provided by the KINS dataset and rendered disparity pseudo-GT, represented as Ours '(vkb)' in Tab. 1. As shown in Tab. 1, training with fused LiDAR points reaches lower performance than 'Ours (vkb)'. The reasons are as follows. 1) Fusing LiDAR point clouds requires 3D bounding box tracklet annotations. However, only 61.8% images in the KITTI object dataset are annotated with tracklets. 2) LiDAR points on transparent surfaces such as glasses cannot be captured in any frame and thus remain low-density after fusion. Thus, we find it better to supervise the iDispNet using rendered disparity maps instead of fused LiDAR point clouds.

The necessity of using instance-specific shape. To validate the necessity of using a different shape for each instance, we generate pseudo-ground-truth by using the mean shape for all the instances in the car category as an ablation, represented by 'Ours (ms)'. The results are shown in Tab. 1, where 'Ours (ms)' reaches a lower performance than 'Ours', especially at the hard level and when 0.7 is used as the IoU threshold. If we use the average shape to supervise the iDispNet, the predicted disparity will not be accurate enough, thus making the following 3D detector to struggle to predict correct 3D bounding boxes from inaccurate point clouds.

Point cloud generation for pseudo-GT. In the 'Ours' version, we generate pseudo-GT using a PSMNet trained on the KITTI-Stereo dataset, which is in the same domain as the KITTI object dataset. To validate the cross-domain generalization of our method, we generate pseudo-GT using point cloud predicted by a PSMNet trained on the DrivingStereo dataset [45] as the supervision to the iDispNet. The results are listed in Tab. 1, represented by 'Ours (dob)'. Compared to the 'Ours' version, the results drop a little, but still outperform previous methods by a large margin.

4.3 Running Time

Tab. 9 shows the running time comparison of our method and other stereo methods. Our method takes 387ms at inference time on average on the KITTI dataset, surpassing all prior stereo methods. Specifically, our method takes 190ms for the 2D detection and segmentation, 87ms for the instance disparity estimation, and 110ms 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.

TABLE 10
Detection precision using iDispNet with different disparity search ranges and RoI input sizes. AP_{bev} and AP_{3d} are reported for the car category on the KITTI validation set using 0.7 as IoU threshold.

Disparity range	Input size	AP_{bev} (IoU=0.7)			AP_{3d} (IoU=0.7)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
-48 ∼ 48	224 × 224	77.85	66.21	57.78	69.92	53.81	46.70
-16 ∼ 32	224 × 224	76.64	63.76	55.73	63.28	51.31	44.05
-24 ∼ 24	112 × 112	76.49	63.26	55.41	63.92	47.05	44.14

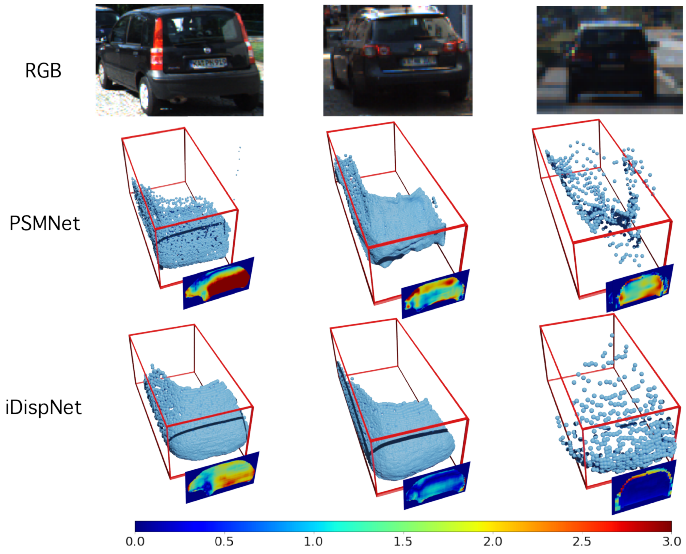


Fig. 9. **Qualitative comparison of disparity estimation results between PSMNet and our iDispNet.** 3D ground-truth bounding boxes are shown in red. Disparity error maps are shown as well, where the larger value indicates the worse disparity.

To further boost the running speed of our method, we perform experiments using different disparity search ranges and RoI input sizes for the iDispNet. The running time of the iDispNet and the detection performance are in Fig. 11 and Tab. 10, respectively. Although the running time depends on the number of detected objects, the iDispNet runs faster than the full-frame PSMNet, especially using a narrower disparity search range and smaller RoI input size, while maintaining comparable performance at the same time as visualized in Fig. 12.

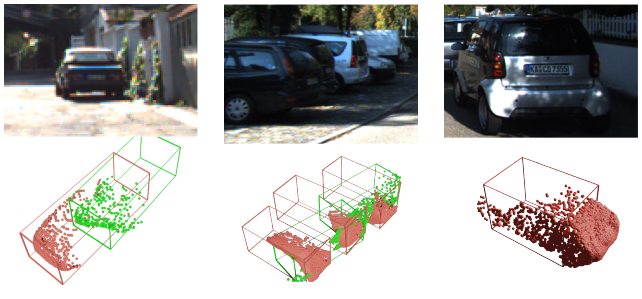


Fig. 10. **Failure cases.** The ground-truth bounding boxes and the pseudo-GT point clouds are visualized in red, while the predictions are visualized in green.

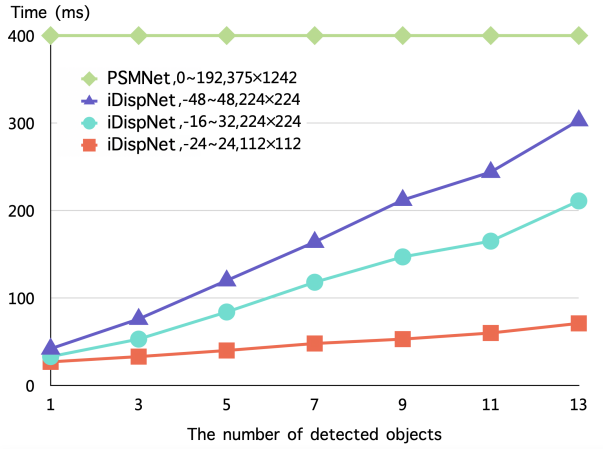


Fig. 11. **Running time comparison between PSMNet and iDispNet.** iDispNet is tested with different disparity search ranges and RoI input sizes. Texts in the legend from left to right: model, disparity search range and input size.

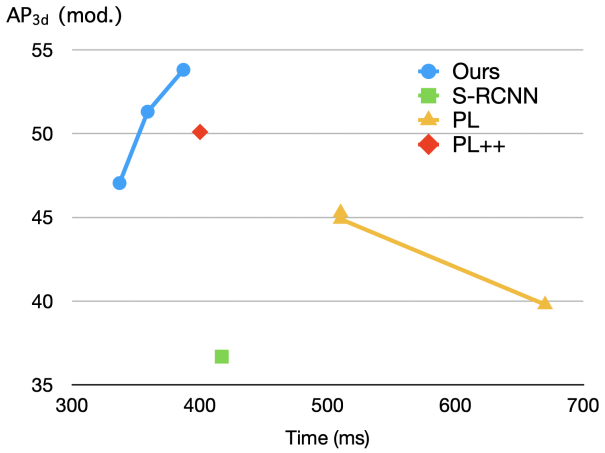


Fig. 12. **Speed-accuracy trade-off on the KITTI validation set.** AP_{3d} using 0.7 as IoU threshold is reported in the moderate level for the car category on the KITTI validation set. Ours uses different disparity search ranges and RoI input sizes, while PL uses different 3D detectors.

4.4 Failure Cases

We visualize some failure cases in Fig. 10. Our 3D object detection method is most likely to fail on objects that are too far away as shown in Fig. 10(a), or under strong occlusion or truncation as shown in Fig. 10(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. 10(c) which is much shorter than other cars. Since there are very few examples with this kind of shape in the CAD model training set, so it is difficult to reconstruct these types 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 for all categories, especially when LiDAR supervision was not available at training time.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support from the National Key Research and Development Program of China (No. 2020AAA0108901), NSFC (No. 61806176), and ZJU-SenseTime Joint Lab of 3D Vision.

REFERENCES

[1] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 66–75.

[2] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[3] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 185–194.

[4] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2345–2353.

[5] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. 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*, 2019, pp. 8445–8453.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[7] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.

[8] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *5th IEEE EMBS International Summer School on Biomedical Imaging, 2002.* IEEE, 2002, pp. 8–pp.

[9] F. Engelmann, J. Stückler, and B. Leibe, "Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors," in *German Conference on Pattern Recognition*. Springer, 2016, pp. 219–230.

[10] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[11] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," in *International Conference on Learning Representations*, 2020.

[12] Y. Chen, S. Liu, X. Shen, and J. Jia, "Dsgn: Deep stereo geometry network for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[13] J. Sun, L. Chen, Y. Xie, S. Zhang, Q. Jiang, X. Zhou, and H. Bao, "Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 548–10 557.

[14] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[15] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7644–7652.

[16] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.

[17] A. D. Pon, J. Ku, C. Li, and S. L. Waslander, "Object-centric stereo matching for 3d object detection," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[18] Z. Xu, W. Zhang, X. Ye, X. Tan, W. Yang, S. Wen, E. Ding, A. Meng, and L. Huang, "Zoomnet: Part-aware adaptive zooming neural network for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[19] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[20] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9277–9286.

[21] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[23] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[25] J. K. Murthy, S. Sharma, and K. M. Krishna, "Shape priors for real-time monocular object localization in dynamic environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1768–1774.

[26] R. Wang, N. Yang, J. Stueckler, and D. Cremers, "Directshape: 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.

[27] F. Manhardt, W. Kehl, and A. 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*, 2019, pp. 2069–2078.

[28] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *European Conference on Computer Vision*. Springer, 2016, pp. 561–578.

[29] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 459–468.

[30] W. Kim, M. S. Ramanagopal, C. Barto, M.-Y. Yu, K. Rosaen, N. Goumas, R. Vasudevan, and M. Johnson-Roberson, "Pedx: Benchmark dataset for metric 3-d pose estimation of pedestrians in complex urban intersections," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1940–1947, 2019.

[31] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[33] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[34] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," 2019.

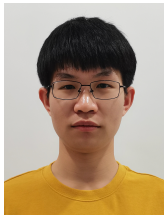
[35] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[36] R. Qian, D. Garg, Y. Wang, Y. You, S. Belongie, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, "End-to-end pseudo-lidar for image-based 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5881–5890.

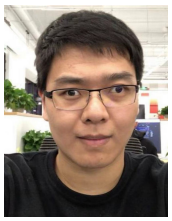
[37] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[38] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[39] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2252–2261.
[40] Z. Qin, J. Wang, and Y. Lu, "Triangulation learning network: from monocular to stereo 3d object detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 7607–7615.
[41] W. Peng, H. Pan, H. Liu, and Y. Sun, "Ida-3d: Instance-depth-aware 3d object detection from stereo vision for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 015–13 024.
[42] H. Königshof, N. O. Salscheider, and C. Stiller, "Realtime 3d object detection for automated driving using stereo vision and semantic information," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1405–1410.
[43] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 424–432.
[44] L. Qi, L. Jiang, S. Liu, X. Shen, and J. Jia, "Amodal instance segmentation with kins dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3014–3023.
[45] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 899–908.



Linghao Chen is currently a Ph.D. student of computer science at Zhejiang University, advised by Dr. Xiaowei Zhou. He received a B.E. degree in computer science from Zhejiang University in 2019. His research interests include 3D object detection and 3D reconstruction.

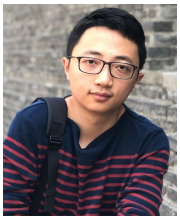


Jiaming Sun is currently working as Computer Vision Researcher at SenseTime. He received his B.E. degree in computer science from the Southern University of Science and Technology (SUSTech) in 2017. His research interests include 3D object detection and 3D reconstruction.

Yiming Xie is currently a research intern at Zhejiang University, advised by Dr. Xiaowei Zhou. He received a B.E. degree in opto-electronics information engineering from Zhejiang University in 2019. His research interests include 3D object detection and 3D reconstruction.

Siyu Zhang is currently a Research Engineer at SenseTime. He received a B.E. degree in Computer Science and Engineering from Southern University of Science and Technology in 2020. His research interests include object pose estimation, visual localization, and 3D reconstruction.

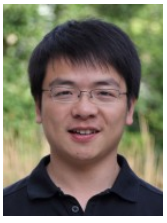
Qing Shuai is currently a master student of computer science at Zhejiang University, advised by Dr. Xiaowei Zhou. He received a B.E. degree from Zhejiang University in 2019. His research interests include 3D human pose estimation, human mesh recovery.



Qinhong Jiang is currently a Senior R&D Manager at SenseTime, leading the perception model&application team in the intelligent driving group. His research interests are mainly focused on autonomous driving, including but not limited to visual perception, LiDAR perception, multi-sensor fusion, intention understanding, trajectory prediction and software system optimization.

Guofeng Zhang now is a Professor at State Key Lab of CAD&CG, Zhejiang University. He received his BS and Ph.D. degrees in Computer Science from Zhejiang University, in 2003 and 2009, respectively. His research interests include structure-from-motion, SLAM, 3D reconstruction, augmented reality, video segmentation, and editing.

Hujun Bao is currently a professor in the State Key Laboratory of CAD&CG and College of Computer Science and Technology, Zhejiang University. He graduated from Zhejiang University in 1987 with a B.Sc. degree in mathematics and obtained his Ph.D. degree in applied mathematics from the same university in 1993. His research interests include computer graphics and mixed reality.



Xiaowei Zhou is a Research Professor of Computer Science at Zhejiang University, China. He obtained his Ph.D. degree from The Hong Kong University and Science and Technology, after which he was a postdoctoral researcher at the GRASP Lab, University of Pennsylvania. His research interests include 3D reconstruction and scene understanding.