The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

RTS3D: Real-time Stereo 3D Detection from 4D Feature-Consistency Embedding Space for Autonomous Driving

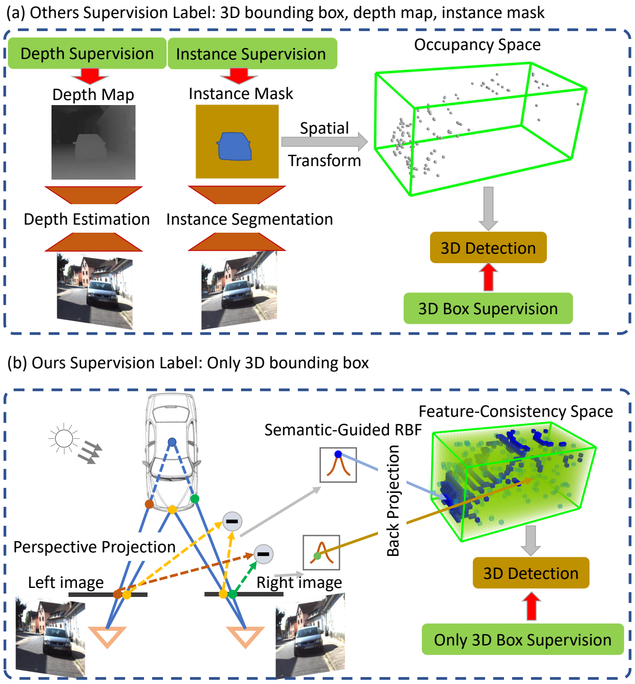
Peixuan Li[1,2,3,5,6], **Shun Su**[1,2,3,4, *], **Huaici Zhao**[1,2,5,6,†]
[1]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
[4]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[5]Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences
[6]Key Lab of Image Understanding and Computer Vision, Liaoning Province
lipeixuan@sia.cn, sushun@sia.cn, hczhao@sia.cn

Abstract

Although the recent image-based 3D object detection methods using Pseudo-LiDAR representation have shown great capabilities, a notable gap in efficiency and accuracy still exist compared with LiDAR-based methods. Besides, overreliance on the stand-alone depth estimator, requiring a large number of pixel-wise annotations in the training stage and more computation in the inferencing stage, limits the scaling application in the real world. In this paper, we propose an efficient and accurate 3D object detection method from stereo images, named RTS3D. Different from the 3D occupancy space in the Pseudo-LiDAR similar methods, we design a novel 4D feature-consistent embedding (FCE) space as the intermediate representation of the 3D scene without depth supervision. The FCE space encodes the object's structural and semantic information by exploring the multiscale feature consistency warped from stereo pair. Furthermore, a semantic-guided RBF (Radial Basis Function) and a structure-aware attention module are devised to reduce the influence of FCE space noise without instance mask supervision. Experiments on the KITTI benchmark show that RTS3D is the first true real-time system (FPS>24) for stereo image 3D detection meanwhile achieves 10% improvement in average precision comparing with the previous state-of-the-art method.

Introduction

3D object detection serves as an important role in many applications, such as augmented reality, robotics, and autonomous driving. Although recently developed LiDAR-based detection algorithms (Shi et al. 2020; He et al. 2020) show some excellent performance, the high price, low service life, and discordant appearance of the LiDAR system restrict its further development in practical applications. Alternatively, the solutions relying on cameras are very competitive for its low-cost, low-power consumption, and high flexibility in development. Therefore, it is grabbing much more attention in computer communities recently (Chen


(a) Others Supervision Label: 3D bounding box, depth map, instance mask

(b) Ours Supervision Label: Only 3D bounding box

Figure 1: Comparisons between occupancy space and proposed FCE space: (a) execute a depth generator to encode the structure of the object in a 3D occupancy space, and use instance mask to reduce the influence of noise in non-target areas. By contrast, our proposed RTS3D, as shown in (b), encode the structure of the object by estimate the consistency between warped left and right images for each 3D locations, and explicitly model the semantic cues for noise filtering, yielding superior accuracy and efficiency without additional label supervision.

[*] Peixuan Li and Shun Su contributed equally to this work
[†] Huaici Zhao is corresponding authors

et al. 2020; Sun et al. 2020; Li, Chen, and Shen 2019).

Image-based methods have two main tasks: 1) to find an appropriate and effective representation to recover the geometric structure of a 3D scene, and 2) to eliminate the interference of non-target areas. For the first task, Wang et al. have proposed a Pseudo-LiDAR representation (Wang et al. 2019), which expand 3D object detection from the 2D frontal view space to the 3D occupancy space. Recent methods try to tackle the second task by designing an instance-level Pseudo-LiDAR generator (Pon et al. 2019; Xu et al. 2020; Sun et al. 2020), which only estimate the depth map of the objects of interest. However, all these approaches heavily rely on extra sub-networks to perform CAD model generation (Sun et al. 2020), instance segmentation (Xu et al. 2020) or depth map estimation (Sun et al. 2020; Wang et al. 2019; Xu et al. 2020), as shown in Fig. 1 a. The additional pixel-wise labels for supervised learning required in these sub-networks become the biggest obstacle in collecting labor-intensive annotations, make it impractical in many real application scenarios. Moreover, reliance on stand-alone sub-networks makes an inherent disconnection in transmitting gradient while consuming plenty of computing resources in the training and inferring stages, limiting upper-bound of the detection accuracy and speed. Here, we tackle these two tasks without relying on additional labels while achieving true real-time detection with competitive accuracy against the state-of-the-art method.

The main contribution of our approach is a novel 4D intermediate representation of 3D object structure, named FCE space. This is different from the previous 3D occupancy space in Pseudo-LiDAR similar methods that represent object structure by estimating whether a location is occupied or not, as shown in Fig. 1 a). Here, we encode the structure of underlying objects by the feature consistency between warped left and right images for each 3D locations in latent space, as shown in Fig. 1 b. The rationalization behind the proposed representation comes from a typical assumption that the intensity of light projected onto the stereo image from the visible surface of a 3D object should be more consistent than from the non-object surface. The same assumption is also used in the plane-sweeping method (Collins 1996) to estimate the depth map, thus proving that the consistency space can encode structural information. We aim to establish such a consistency space and directly detect objects on it.

However, establishing FCE space is complex in the computation of the entire camera visual range, and this unsupervised space contains an enormous amount of noise due to the interference of non-Lambert properties, the textureless region, and nontarget surface.

We address these issues in four steps. First, we only compute feature consistency in the latent space of the target object. The initial latent space is predicted by monocular 3D detection at a high speed. Later it would be iteratively refined by the detection results of FCE space. Second, we compute the consistency from the multi-scale feature to make it more reliable in textureless and reflective regions. Predicting the required consistency only need local neighborhood, so a very simple convolutional neural network(e.g. ResNet18 (He et al. 2016)) is adopted to extract the multi-scale features. Third, we propose to encode the semantic information in an RBF to reduce the interference of the nontarget surface. This semantic-guided RBF explicitly modeling the semantic cues to 3D space is easier to converge than implicit learning possible relationships. Fourth, we propose a structure-aware attention (StrAA) module to further filter the spatial noise and capture local structure at a smaller computational cost than 3D CNN and PointNet++(Qi et al. 2017b).

To summarize, Our contributions are as follows: **1.)** An image-based 3D object detection approach predicts the 3D box of objects more efficiently and accurately. **2.)** A novel intermediate representation of object structure that bridges the performance gap between LiDAR-based and image-based methods without additional label supervision. **3.)** A semantic-guided RBF and a StrAA module to reduce the interference of noise and optimize the characterization of local structure in the FCE space. **4.)** Evaluation on the popular KITTI dataset shows that the proposed method is the first true real-time 3D detection approach using only images and achieves comparable detection accuracy against the other competitors.

Related Work

Monocular 3D Object Detection. Due to the lack of depth, 3D object detection is difficult given only a monocular image. A common theme of these methods is to employ sub-networks to generate extra 2.5D feature, such as depth map(Xu and Chen 2018; Ma et al. 2019), object mask(Chen et al. 2016), or CAD model(Chabot et al. 2017). Recent monocular-only works attempt to apply the geometry constrain as the post-processing (Mousavian et al. 2017; Li et al. 2020) or embedding knowledge to aid in detection. These methods explicitly model the relationship between 3D location and 2D feature, which enables them to be improved in both accuracy and running speed. However, their promised accuracy still not good enough comparing stereo approaches.

Stereo-based 3D Object Detection. Like monocular approaches, stereo methods can also be roughly divided into two ways by the type of training data. One is Pseudo-LiDAR similar pipeline. These methods (Wang et al. 2019; You et al. 2019; Chen et al. 2020) first use a SOTA disparity prediction with stereo processing to generate a depth map following to convert this depth map to occupancy space. Then apply a LiDAR-based framework (Shi, Wang, and Li 2019; Qi et al. 2017a) to detect object. In order to save computation and avoid streaking noise caused by non-target regions, the recent method aims to detect object only in potential area by introducing the instance mask (Sun et al. 2020; Pon et al. 2019; Xu et al. 2020; Dong et al. 2020). Intuitively, these methods containing more prior information, from extra-label supervision, would certainly improve the performance of detection. However, reliance on additional sub-networks and labels also leads to more time consumption and labor-intensive work. Another one, therefore, tries to fully explore the potency of stereo images. Stereo R-CNN (Li, Chen, and Shen 2019) associate left and right 2D box to generate rough 3D box that are later refined by dense

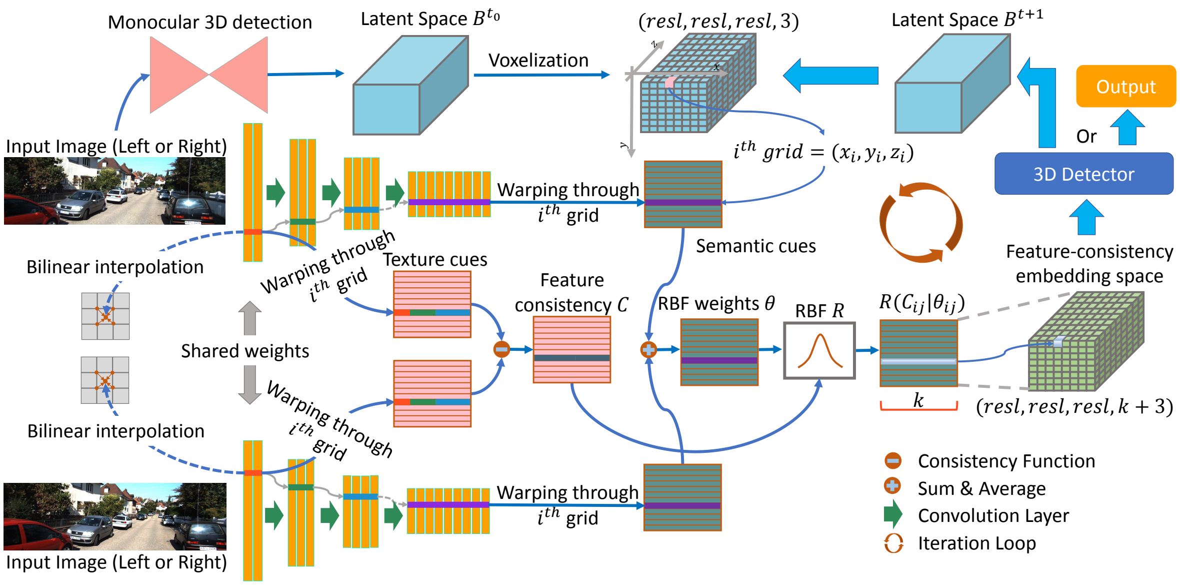


Figure 2: Overview of RTS3D architecture. Stereo images are first passed through a simple siamese network to generate the multi-scale feature. In parallel, a coarse latent space is predicted by fast monocular 3D detection and then is split to a regular grid. The FCE space is generated by warping the left and right multi-scale features to each location in latent space after the semantic-guided RBF. The 3D detector estimates the 3D box from the FCE space as the final output or generates a more refined hidden space for the next iteration.

3D box alignment. TLNet (Qin, Wang, and Lu 2019) enumerate a multitude of 3D anchors and then construct object-level correspondences to filter out dreadful proposals. However, these methods perform a lower accuracy and running speed comparing with the Pseudo-liDAR similar methods. By comparison, the proposed method has the fastest running and achieve a competitive accuracy comparing with the Pseudo-liDAR without extra labels help.

Proposed Method

Given a stereo pair (I_L, I_R), the goal is to estimate the 3D property of object typical represented by $B = (X, Y, Z, W, H, L, \theta)$, which denotes the 3D center position, width, height, length, and horizontal orientation respectively. Fig. 2 shows an overview of the proposed framework. It comprises four stages: 1) A very fast monocular 3D detector is leveraged to obtain initial latent space. 2) Multi-scale features are back-projected onto the grid of initial latent space to construct FCE space 3) A semantic-guided RBF and structure-aware attention module reduce the influence of FCE space noise and optimize the characterization of local structure. 4) A variant of the PointNet to predict the final 3D box with its confidence or generate more specific latent space for the next iteration.

Latent Space Generation. Instead of creating an entire viewable FCE space, we only computer the feature consistency in latent space containing the object of interest. Benefiting from the recent development of monocular 3D detection, we propose to employ an efficient one of them to generate an initial coarse cuboid B^{t_0} as the guidance of latent space. Later this coarse cuboid can be iteratively refined as B^{t+1} by the detection results of established FCE space. Here, we choose two monocular 3D object detection frameworks for the trade-off between speed and accuracy: KM3D-Net and CenterNet (Zhou, Wang, and Krähenbühl 2019). Both structures are one-stage 3D detectors and do not rely on extra annotation for the training.

Multi-scale Texture Cues Generation. Inspired by traditional stereo matching methods (Zhang et al. 2014), which process the correspondence by texture cues across multiple scales, we generate the consistency of a 3D location from pair images by extracting the hierarchical contextual information of low-level features. To ensure the real-time performance,the simple convolutional encoding structures, ResNet18(He et al. 2016), is adopted to output the multi-feature $\{F^{s}_{l\ r}\}^{S}_{s}$ with the downsampling stride $s = 2, 4, 8$. However, relying heavily on texture cues will unavoidably introduce noise from non-target objects, such as the ground or other objects. To overcome this issue, we add one high-level feature output with downsampling stride $/32$ to predict semantic cues. Nevertheless, without the instance mask and depth map annotation supervision, implicit fusion of this information makes the model difficult to converge. We, therefore, design a semantic-guided RBF to explicitly encode two cues for noise filtering.

Building the Feature-Consistency Embedding Space. Given the latent space, texture cues and semantic cues on object of interest, we convert them to the FCE space to encode geometric structure. We first split the latent space to regular grid with resolution ratio $resl$, which represent the latent space as $G = \{g_i = [x_i, y_i, z_i] \in \mathbb{R}^3\}_{i=1 \dots resl \times resl \times resl}$. After that, we project a voxel g_i into feature space $x_i^s = [u_i^s, v_i^s, 1]^T$ by using camera intrinsics K, extrinsic parameters T, consisting of a rotation matrix R and a translation matrix t of the left and right camera, and coordinate affine transformation h_s of the original image into the multi-scale features:

$$ {}^{l\,r}x_i^s = h_s K_{l\,r} \begin{bmatrix} R_{l\,r}^{3\times3} & t_{l\,r}^{1\times3} \\ \\ 0^T & 1 \end{bmatrix} g_i \tag{1}$$

where $l\,r$ indicates it belongs to the left or right image. The purpose of introducing affine transformation matrix h_s without uniform zooming parameters factor is to reduce the quantization error caused by different downsampling stride of original image scaling. Then the consistency of the 3D voxel g_i from the left and right image can be defined as:

$$C_i^s = f\left(\hat{F}_l({}^l x_i), \hat{F}_r({}^r x_i)\right), \hat{F}_l = Cat\left[\hat{F}_l^s \cdots\right]_{s=2}^{8} \tag{2}$$

Here, Cat means concatenation. Note that the projected coordinates x_s^i are continuous values and the feature vectors are all integer coordinates. We, therefore, use the differentiable bilinear sampling mechanism $\hat{F}$ inspired by spatial transformer networks (Jaderberg et al. 2015). f is a pair function of measures distance that represents the similarity of two signals. There are many existing choices for f, such as absolute difference, gaussian distance, cosine correlation, and concatenation. However, The first three methods are difficult to encode semantic cues and the uncertainty of each dimension. Concatenation implicitly encodes the uncertainty, but it is difficult to learn without the supervision of depth maps. We propose a novel semantic-guided RBF to explore pair signal relationship by combine texture cues and semantic cues:

$$\hat{C}_i^s = RBF(\hat{F}_l({}^l x_i) - \hat{F}_r({}^r x_i) | \alpha_i) \tag{3}$$

RBF denotes Radial Basis Function with parameters α that is normally the variance of a multi-scale feature in a given coordinate of voxel. Here, we consider $\alpha_i = \frac{1}{2}\left(\hat{F}_l^{32}({}^l x_i^{32}) + \hat{F}_r^{32}({}^r x_i^{32})\right)$ in the form of learnable parameters from semantic cues. By doing this, α will be reduced in unreliable channels and non-target location in the image feature.

3D Bounding Box Prediction. After generating the FCE space in the grid form, the common solution is to employ 3D convolution networks (3DCNNs) to extract local features for the estimation of 3D bounding. However, we set a very small resolution (min $resl = 10$ in our experiments) for the trade-off between accuracy and speed. This makes it difficult to determine the size of the 3D convolution kernel. For example, a large convolution kernel will introduce a lot of padding noise, while a small convolution kernel will increase computation but it is not obvious to extract local features. Here, we design a variant of Point-Net (Qi et al. 2016) with a StrAA module for 3D box prediction and confidence estimation, as shown in Fig.3

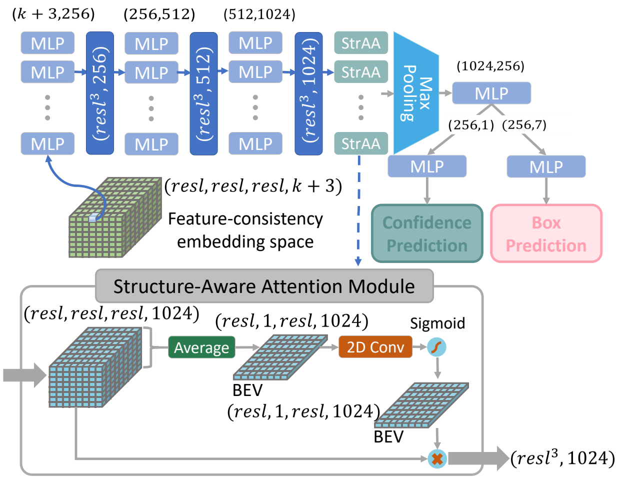


Figure 3: Overview of the proposed 3D object detector.

Structure-aware attention module. We first map the consistency of each voxel in FCE space to higher-dimensional vector $G_h \in \mathbb{R}^{1024 \times resl \times resl \times resl}$ by point-wise multi-layer perceptron (MLP). Although the semantic-guided RBF can reduce the interference of non-target area noise, the FCE space still has a lot of spatial noise because the vehicle is a typical Non-Lambert. To address this issue, we present a StrAA module to reduce the interference of unstructured spatial noise. Many recent LiDAR-based methods (Beltran et al. 2018) can detect 3D objects on a bird‘s eye view (BEV), indicating that the top view contains the structural information needed for detection. Therefore, to determine if a particular point belongs to the structure of the object, we can search the boundary of the object on BEV from the average value of the height direction. Specifically, StrAA first compute the average of G_h in the height dimension as $G_a \in \mathbb{R}^{1024 \times resl \times resl}$, and then apply a standard 2D convolution with 3×3 kernel size and $sigmoid$ to capture local structures. The output $G_m \in \mathbb{R}^{1024 \times resl \times resl}$ also can be regard as the attention map, inspired by self-attention (Vaswani et al. 2017). We obtain the final output $G_a \in \mathbb{R}^{1024 \times resl \times resl \times resl}$ by element-wise multiplication and summation. The overall process can be summarized as:

$$G_a = \sigma\left(Conv^{3\times3}\left(Avg(G_h, dim = 2)\right)\right) \otimes G_h + G_h \tag{4}$$

where $\otimes$ denotes element-wise multiplication. During multiplication, the attention map G_m are broadcasted (copied) along the object hight dimension. After StrAA module, the G_a are fed into the symmetric function following (Qi et al. 2016) to predict 3D box and its confidence.

Losses for box prediction. The box prediction head returns for each latent space with residual regression $\Delta B = (\Delta X, \Delta Y, \Delta Z, \Delta W, \Delta H, \Delta L, \Delta\theta)$ and its confidence P_B. Although these regression terms are independent, they are intrinsically related to the final box prediction. To sidestepping the issue of finding a proper weighting of each regression terms, we follow the $disentangling$ transformation (Simonelli et al. 2019) to decompose ΔB into 3 groups

Method	Extra	Time	$IoU > 0.5$ [$\boldsymbol{val}$]			$IoU > 0.7$ [$\boldsymbol{val/test}$]		
			Easy	Moderate	Hard	Easy	Moderate	Hard
3DOP	Mask	-	46.0	34.6	30.1	6.6 / -	5.1 / -	4.1 / -
MLF	Depth	-	-	47.4	-	- / -	9.8 / -	- / -
RT3DStereo	Depth+Mask	**92ms**	-	-	-	- / 28.5	- / 24.1	- / 20.32
PL: F-PointNet	Depth+Flow	670ms	89.5	75.5	66.3	59.4 / 39.7	39.8 / 26.7	33.5 / 22.3
PL: AVOD	Depth+Flow	510ms	88.5	76.4	61.2	61.9 / 55.40	45.3 / 37.17	39.0 / 31.37
PL++: AVOD	Depth+Flow	500ms	89.0	77.8	69.1	63.2 / -	46.8 / -	39.8 / -
PL++: P-RCNN	Depth+Flow	510ms	88.0	73.7	67.8	62.3 / -	44.9 / -	41.6 / -
OC-Stereo	Depth+Mask	350ms	89.65	**80.03**	70.34	64.07 / 55.15	48.34 / 37.60	40.39 / 30.25
ZoomNet	Depth+Mask	-	90.44	79.82	**70.47**	62.96 / 55.98	**50.47** / 38.64	**43.63** / 30.97
Disp R-CNN	Depth+Mask+CAD	425ms	**90.47**	79.76	69.71	**64.29 / 59.58**	47.73 / **39.34**	40.11 / **31.99**
TL-Net	None	-	59.51	43.71	37.99	18.15 / -	14.26 / -	13.72 / -
Stereo RCNN	None	417ms	85.84	66.28	57.24	54.11 / 49.23	36.69 / 34.05	31.07 / 28.39
Ours($iter$=0)	None	30.2ms	89.46	77.30	62.36	60.33 / 54.12	44.48 / 34.59	37.99 / 28.91
Ours ($iter$=1)	None	**39.4ms**	90.34 (-0.13)	79.67 (-0.36)	70.29 (-0.18)	64.76 / 58.51 (+0.47)/(-1.01)	46.70 / 37.38 (-3.77)/(-1.96)	39.27 / 31.12 (-4.36)/(-0.87)

Table 1: Comparison 3D detection methods for car category, evaluated by metric AP_{3D} on $\boldsymbol{val}$ / $\boldsymbol{test}$ set on KITTI. Mask means instance mask or segmantic mask.

(dimensions ϕ_1, position ϕ_2, and orientation ϕ_3) and unify the loss by the distance L_{dis} of eight corners and one center between prediction and ground-truth. In short, the unify loss is computed as:

$$L_{dis}(\phi, -\phi) = \frac{1}{9}\sum_{j=1}^{9} \|\pi(\Delta B(\phi, -\phi) + B_{ini}), \pi(B_{gt})\|_2$$

$$L_{reg} = \sum_{m=1}^{3} L_{dis}(\phi_m, \phi_{-m}^{gt}) \tag{5}$$

Here, $\pi : \mathbb{R}^7 \to \mathbb{R}^{3\times 9}$ transform property of box to 3D coordinate of its eight corners and one center. ϕ_m denotes the sub-vector corresponding to the mth group, and ϕ_{-m}^{gt} denotes the sub-vector in ground truth corresponding to all but the mth group.

The confidence classification loss L_{cls} aims to sort the quality of the target box. The label can be defined as:

$$\hat{p} = \begin{cases} 1 & IoU_{3D} > 0.75 \\ 0 & IoU_{3D} < 0.25 \\ 2IoU_{3D} - 0.5 & otherwise \end{cases} \tag{6}$$

where IoU_{3D} is the 3D intersection over union between prediction $\Delta B + B_{ini}$ and ground-truth B_{gt}. We then use the cross entropy loss to supervise the the predicted confidence. The overall training objective is:

$$L = L_{reg} + \omega(t)L_{cls} \tag{7}$$

Since the early training was unstable and the IoU_{3D} was generally small, the time function $\omega(t) = exp[-5(1 - t/100)^2]$ was used to balance the weight of the two losses.

Experimental

Implementation Details

We evaluate the proposed approach on the KITTI 3D detection benchmark, which consists of 7481 training stereo images and 7518 test stereo images. We follow the protocol in (Wang et al. 2019; Sun et al. 2020; Pon et al. 2019) to split the training set as $train$ set (3712 images) and val set respectively, and comprehensively compare proposed method with others on val set as well as test set. We report two official evaluation metrics in KITTI: average precision for 3D detection (AP_{3D}) and bird's eye view detection (AP_{BEV}). We train our model on the machine E5-2678 CPU with two 2080Ti GPUs and apply $Adam$ optimizer with an initial learning rate of 0.000125. We then train our model for 90 epochs and reduce the learning rate of $10\times$ at 80 epochs. Finally, $train$ set training takes 13 hours and the overall training set consumes 27 hours.

Establishing the FCE space needs the guidance of a coarse 3D box generated by the monocular-based methods. However, aligning the coarse 3D box to the ground truth is difficult. We, therefore, disturb the ground truth and then let our model predict this noise. We empirical set uniform noise in range $L = [-1.5, 1.5]$, $W = [-1.5, 1.5]$, $H = [-1.5, 1.5]$, $\theta = [-0.6, 0.6]$, $X = [-2, 2]$, $y = [-0.8, 0.8]$ and $Z = [-3, 3]$. 4.

Comparison with Other Methods

To fully evaluate the performance of the proposed method, we conduct our experiments in three regimes: easy, moderate, and hard, according to the occlusion and truncation levels. In addition to average precision, we also provide a comparison of runtime that is very important to the safety of autonomous driving or mobile robots. The results are shown in Table.1 and Table. 2, we default use KM3D-Net to generate the initial latent space. We can observe that our RTS3D is the fastest running speed while our accuracy outperforms all image-only methods. Specifically, without extra labels helping, RTS3D is 10 times faster than the existing SOTA work Stereo RCNN while achieves 10% improvement in AP_{3D} and AP_{BEV} for the moderate setting accuracy. Among other methods, the fastest is RT3DStereo, which requires depth and semantic masks for detection, while RTS3D only consumes $1/3$ of its runtime, and the accuracy in the easy set

Method	Accelerator	FPS	$IoU > 0.5$ [$\boldsymbol{val}$]			$IoU > 0.7$ [$\boldsymbol{val/test}$]		
			Easy	Moderate	Hard	Easy	Moderate	Hard
3DOP	-	-	55.0	41.3	34.6	12.6 / -	9.5 / -	7.6 / -
MLF	-	-	-	53.7	-	- / -	19.5 / -	- / -
RT3DStereo	TITAN X	**11.0**	25.19	18.20	15.52	- / 59.32	- / 49.48	- / 43.16
PL: F-PointNet	-	1.5	89.8	77.6	68.2	72.8 / 55.0	51.8 / 38.7	44.0 / 32.9
PL: AVOD	-	1.5	76.8	65.1	56.6	60.7 / -	39.2 / -	37.0 / -
PL++: AVOD	-	2.0	89.0	77.5	68.7	74.9 / 66.83	56.8 / 47.20	49.0 / 40.30
PL++: PIXOR	-	2.0	89.9	75.2	67.3	79.7 / 70.7	61.1 / 48.3	54.5 / 41.0
PL++: P-RCNN	-	2.0	88.4	76.6	69.0	73.4 / -	56.0 / -	52.7 / -
OC-Stereo	Titan Xp	2.9	90.01	80.63	71.06	77.66 / 68.89	65.95 / 51.47	51.20 / 42.97
ZoomNet	-	-	90.62	**88.40**	**71.44**	**78.68** / 72.94	**66.19** / **54.91**	**57.60** / **44.14**
Disp R-CNN	-	2.4	**90.67**	80.45	71.03	77.63 / **74.07**	64.38 / 52.34	50.68 / 43.77
TL-Net	-	-	62.46	45.99	41.92	29.22 / -	21.88 / -	18.83 / -
Stereo RCNN	-	2.4	87.13	74.11	58.93	68.50 / 61.67	48.30 / 43.87	41.47 / 36.44
Ours($iter$=0)	2080Ti	33.1	89.88	78.05	69.17	73.43 / 66.79	56.52 / 45.22	48.29 / 38.48
Ours ($iter$=1)	2080Ti	**25.4**	90.58 (-0.09)	80.72 (-7.68)	71.41 (-0.03)	77.50 /(72.17) (-1.18)/ (-1.9)	58.65 / 51.79 (-7.54)/(-3.13)	50.14 / 43.19 (-7.14)/(-0.95)

Table 2: Comparison 3D detection methods for car category, evaluated by metric AP_{BEV} on $\boldsymbol{val}$ / $\boldsymbol{test}$ set on KITTI.

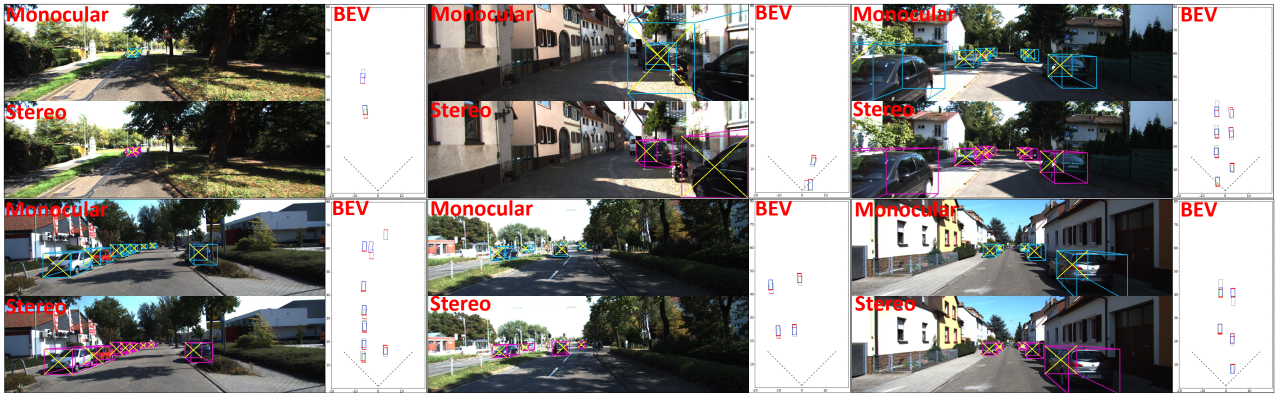


Figure 4: Overview of the proposed 3D object detector. In BEV, Ground truth boxes are in green, stereo predicted boxes in blue and monocular predicted boxes in gray.

is increased by 105%. Moreover, compared with Pseudo-LiDAR, Pseudo-LiDAR++, DispRCNN, ZoomNet, and OC-Stereo, each of them needs to establish a 3D occupancy space or instance occupancy space with the help of a depth map, instance mask, or other labels, we can still obtain competitive detection accuracy but with minimal time consumption. We visualize some qualitative results of object detection in Fig.

Running-time Analysis

In the case of a resolution $resl = 20$, our RTS3D takes 5.6ms for the multi-scale feature extraction from left and right image, 21ms for the latent space generation, 7.6ms for the FCE space building, and 1.6ms for the 3D detection from the FCE space. The latent space generation and the multi-scale feature extraction can be executed in parallel and therefore the overall runtime is 30.2ms with iteration=1, Note that these are the mean runtime over the val set and can vary accordingly the number of the objects in stereo images.

Ablation Study

In this section, we perform comprehensive ablation experiments to validate the contributions of different components in our approach. All experiments are conducted on the $train$ split and evaluated on the val split with the car category. If not specified, $resl$ for all experiments is set to 10 and $interation$ for 1.

Feature-Consistency Embedding Space VS Occupancy Space. For comparing the FCE space and occupancy space, we first generate the 3D occupancy space by using the PSMNet (Chang and Chen 2018). We train the PSMNet on KITTI stereo ground truth to generate depth maps and transform this depth maps to 3D points cloud like most previous Pseudo-LiDAR similar works do (Wang et al. 2019; You et al. 2019). We then use 3D points in the latent space to train our 3D detectors. In this case, the StrAA module is not suitable for dealing with irregular point clouds. We, therefore, removed the StrAA module in all tests for a fair comparison. In addition, for the fairness of the time comparison experiment, The number of the input point cloud are sampled (Vitter 1984) to 1000 which is the same number as the voxel of our FCE space. The results are shown in Table 3. Only PSMNet alone is more computationally intensive than all pipeline of our method and it also need the supervision of ground truth that is usually generated by expensive LiDAR system. Finally, using our FCE space obtains strong improvement in both accuracy and running speed.

Config	Runtime	Set	$\text{AP}_{\text{bev}}^{0.7}$	$\text{AP}_{\text{3d}}^{0.7}$
Occupancy Space	418.4ms	Easy	52.46	34.93
		Mode	38.87	23.53
		Hard	33.27	21.15
FCE Space	39.4ms	Easy	74.12	60.80
		Mode	56.08	43.54
		Hard	47.33	35.70
FCE Space with StrAA	39.4ms	Easy	76.29	62.92
		Mode	57.58	45.18
		Hard	48.99	38.13

Table 3: Comparisons of feature-consistency embedding space and occupancy space.

Config	Method	$\text{AP}_{\text{bev}}^{0.7}$	$\text{AP}_{\text{3d}}^{0.7}$
Channel-reducing	C-MLP	15.74	11.21
	Cosine Correlation	36.38	21.73
	Gaussion	38.87	**23.53**
	RBF with MLP	**39.39**	21.15
Channel-keeping	C-MLP	20.62	13.03
	Absolute Difference	53.08	39.54
	RBF	**57.18**	**45.18**

Table 4: Ablative analysis of the different methods for generating the feature-consistency space. Only the moderate sets are reported.C-MLP is short for Concatenation with MLP

en **Structure-Aware Attention Module.** We further evaluate the effect of the proposed StrAA module. The results are shown in Table 3. Without bells and whistles, we already outperform most of the image-based methods in both accuracy and running speed. StrAA module further enhances our method performance by several points in a slight computation increasing.

Semantic-guided RBF. Table 4 compares different types of generating feature-consistency space. With semantic-guided RBF, the proposed method obtains a better detection accuracy than the absolute difference. In channel-keeping methods, concatenation has the worst results. A possible reason is that the implicit model is difficult to learn effective knowledge without point-wise supervision. Channel-reducing methods average the channel for forcing the model to predict the occupancy of a point. All these methods have a poor performance, which also demonstrates the advantage of learning from the original image feature space.

Config	Data	Set	$\text{AP}_{\text{bev}}^{0.5}$	$\text{AP}_{\text{3d}}^{0.5}$
CenterNet	Mono	Easy	27.22	13.53
		Mode	22.91	10.37
		Hard	19.52	10.56
	Stero	Easy	90.03	89.32
		Mode	77.69	78.18
		Hard	70.39	68.13
KMNet	Mono	Easy	53.77	47.23
		Mode	40.58	34.12
		Hard	34.79	31.51
	Stero	Easy	90.44	90.27
		Mode	79.98	78.27
		Hard	70.75	69.06

Table 5: Ablative analysis of different methods to generate initial latent space.

Resolution of Feature-Consistency Embedding Space. We examine the accuracy and running speed of our method with respect to the resolution of the FCE space. The results are shown in Fig. 5. We observe that the accuracy will increase as the resolution of the FCE space increase until it reaches 20. A larger resolution will bring more details, but a small *batchsize* will also cause training instability. It is worth noting that even with a very small resolution, we still obtain a relatively good detection accuracy and have extremely fast running speed.

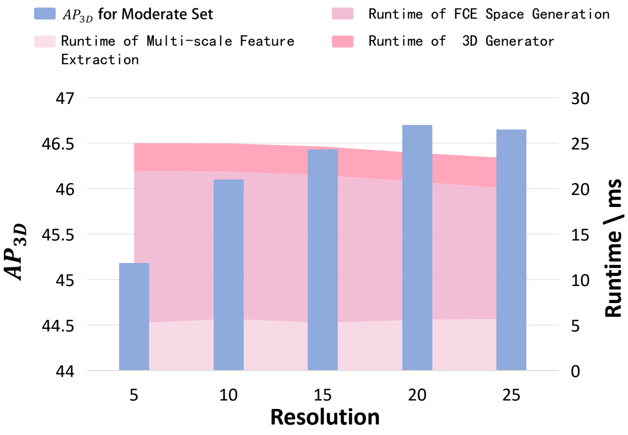


Figure 5: Comparison of different resolutions with their runtime.

Different Monocular 3D Detectors. We compare the impact of different monocular detectors on the generation of initial latent space. As shown in Table 5, even if the accuracy of the monocular detection method varies greatly, the final stereo accuracy is similar. This is likely due to two reasons. First, we use the iterative method to continuously modify this initial latent space. Second, RTS3D is trained on pseudo data, so it is not sensitive to the initial latent space generated by different monocular methods.

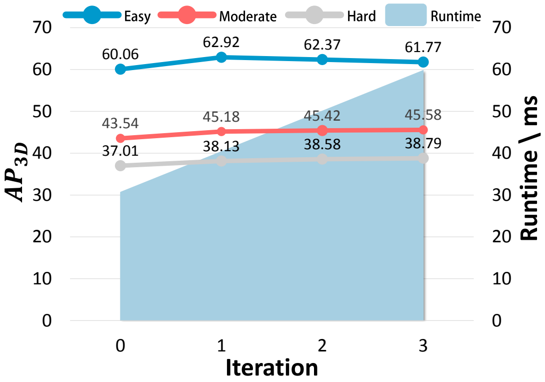


Figure 6: Ablative Analysis of Iterations.

Ablative Analysis of Iterations. We also compare the effects of different iterations on accuracy and runtime, as shown in Fig. 6. When iterations exceed 1, the effect is not significantly improved, but the running time will be greatly increased. Therefore, the number of iterations in our experiment is default set as 1 for the best speed-accuracy trade-off.

Conclusion and Discussion

We present a novel framework to perform faster and more accurate 3D object detection using stereo images. We design a novel 3D intermediate representation space which can encode the structural and semantic information of object without relying on additional annotation. We then propose a semantic-guided RBF and structure-aware attention module for reducing the influence of space noise. Extensive experiments show that our model achieves an unprecedented running speed while competing with the most advanced methods for accuracy.

Exploring the intermediate representation of a 3D scene has always been a meaningful thing. Pseudo-LiDAR transforms a front-view image with estimated depth map to 3D occupancy representation, bridging the gap between the LiDAR- and image-based detection accuracy. We propose a 4D feature-consistency representation to further bridge this gap and greatly improve the detection speed. We believe that the rapid progress in speed can not only greatly ensure the safety of autonomous driving, but also can further enhance accuracy in additional ways. One of the most straightforward methods conceivable is to smooth the detection results between adjacent frames. This is what we will do in future work.

References

Beltran, J.; Guindel, C.; Moreno, F. M.; Cruzado, D.; Garcia, F.; and De La Escalera, A. 2018. BirdNet: A 3D Object Detection Framework from LiDAR Information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 3517–3523. ISSN 2153-0017. doi:10.1109/ITSC.2018.8569311. URL https://ieeexplore.ieee.org/document/8569311.

Chabot, F.; Chaouch, M.; Rabarisoa, J.; Teulière, C.; and Chateau, T. 2017. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2040–2049.

Chang, J.-R.; and Chen, Y.-S. 2018. Pyramid Stereo Matching Network. In *CVPR*, 5410–5418. IEEE Computer Society. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2018.html#ChangC18.

Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; and Urtasun, R. 2016. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2147–2156.

Chen, Y.; Liu, S.; Shen, X.; and Jia, J. 2020. DSGN: Deep Stereo Geometry Network for 3D Object Detection. In (CVPR2020), 12533–12542. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2020.html#Chen0SJ20.

Collins, R. T. 1996. A Space-Sweep Approach to True Multi-Image Matching. In *CVPR*, 358–363. IEEE Computer Society. ISBN 0-8186-7258-7. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr1996.html#Collins96.

CVPR2019. 2019. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE. URL http://openaccess.thecvf.com/CVPR2019.py.

CVPR2020. 2020. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE. ISBN 978-1-7281-7168-5. URL https://ieeexplore.ieee.org/xpl/conhome/9142308/proceeding.

Dong, J.; Cong, Y.; Sun, G.; Liu, Y.; and Xu, X. 2020. CSCL: Critical Semantic-Consistent Learning for Unsupervised Domain Adaptation. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *European Conference on Computer Vision – ECCV 2020*, 745–762. Cham: Springer International Publishing. ISBN 978-3-030-58598-3.

He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020. Structure Aware Single-Stage 3D Object Detection From Point Cloud. In (CVPR2020), 11870–11879. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2020.html#HeZH0Z20.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *Advances in neural information processing systems*, 2017–2025.

Li, P.; Chen, X.; and Shen, S. 2019. Stereo R-CNN Based 3D Object Detection for Autonomous Driving. In (CVPR2019), 7644–7652. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2019.html#0001CS19.

Li, P.; Zhao, H.; Liu, P.; and Cao, F. 2020. RTM3D: Real-Time Monocular 3D Detection from Object Keypoints for Autonomous Driving. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 644–660. Cham: Springer International Publishing. ISBN 978-3-030-58580-8.

Ma, X.; Wang, Z.; Li, H.; Zhang, P.; Ouyang, W.; and Fan, X. 2019. Accurate monocular 3D object detection via Color-Embedded 3D reconstruction for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, 6851–6860.

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

Pon, A. D.; Ku, J.; Li, C.; and Waslander, S. L. 2019. Object-Centric Stereo Matching for 3D Object Detection. *CVRR* abs/1909.07566. URL http://dblp.uni-trier.de/db/journals/corr/corr1909.html#abs-1909-07566.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. URL http://arxiv.org/abs/1612.00593. Cite arxiv:1612.00593.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NIPS*, 5099–5108. URL http://dblp.uni-trier.de/db/conf/nips/nips2017.html#QiYSG17.

Qin, Z.; Wang, J.; and Lu, Y. 2019. Triangulation Learning Network: from Monocular to Stereo 3D Object Detection. *CoRR* abs/1906.01193. URL http://dblp.uni-trier.de/db/journals/corr/corr1906.html#abs-1906-01193.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In (CVPR2020), 10526–10535. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2020.html#ShiGJ0SWL20.

Shi, S.; Wang, X.; and Li, H. 2019. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In (CVPR2019), 770–779. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2019.html#ShiWL19.

Simonelli, A.; Bulo, S. R.; Porzi, L.; López-Antequera, M.; and Kontschieder, P. 2019. Disentangling monocular 3d object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1991–1999.

Sun, J.; Chen, L.; Xie, Y.; Zhang, S.; Jiang, Q.; Zhou, X.; and Bao, H. 2020. Disp R-CNN: Stereo 3D Object Detection via Shape Prior Guided Instance Disparity Estimation. In

(CVPR2020), 10545–10554. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2020.html#SunCXZJZB20.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need 30: 5998–6008. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vitter, J. S. 1984. Faster Methods for Random Sampling. *Commun. ACM* 27(7): 703–718. URL http://dblp.uni-trier.de/db/journals/cacm/cacm27.html#Vitter84.

Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. 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*, 8445–8453.

Xu, B.; and Chen, Z. 2018. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2345–2353.

Xu, Z.; Zhang, W.; Ye, X.; Tan, X.; Yang, W.; Wen, S.; Ding, E.; Meng, A.; and Huang, L. 2020. ZoomNet: Part-Aware Adaptive Zooming Neural Network for 3D Object Detection. *CVRR* abs/2003.00529. URL http://dblp.uni-trier.de/db/journals/corr/corr2003.html#abs-2003-00529.

You, Y.; Wang, Y.; Chao, W.-L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. *CVRR* abs/1906.06310. URL http://dblp.uni-trier.de/db/journals/corr/corr1906.html#abs-1906-06310.

Zhang, K.; Fang, Y.; Min, D.; Sun, L.; Yang, S.; Yan, S.; and Tian, Q. 2014. Cross-Scale Cost Aggregation for Stereo Matching. In *CVPR*, 1590–1597. IEEE Computer Society. ISBN 978-1-4799-5118-5. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2014.html#ZhangFMSYYT14.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. In *arXiv preprint arXiv:1904.07850*.