Stereo CenterNet based 3D Object Detection for Autonomous Driving

Yuguang Shi\textsuperscript{a,b,c}, Yu Guo\textsuperscript{a,d,*}, Zhenqiang Mi\textsuperscript{a}, Xinjie Li\textsuperscript{a}

\textsuperscript{a}School of Computer and Communication Engineering, University of Science and Technology Beijing, China
\textsuperscript{b}Beijing Key Laboratory of Knowledge Engineering for Materials Science, University of Science and Technology Beijing, China
\textsuperscript{c}Beijing Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, China
\textsuperscript{d}Shunde Graduate School, University of Science and Technology, Guangdong, China

Abstract

Recently, three-dimensional (3D) detection based on stereo images has progressed remarkably; however, most advanced methods adopt anchor-based two-dimensional (2D) detection or depth estimation to address this problem. Nevertheless, high computational cost inhibits these methods from achieving real-time performance. In this study, we propose a 3D object detection method, Stereo CenterNet (SC), using geometric information in stereo imagery. SC predicts the four semantic key points of the 3D bounding box of the object in space and utilizes 2D left and right boxes, 3D dimension, orientation, and key points to restore the bounding box of the object in the 3D space. Subsequently, we adopt an improved photometric alignment module to further optimize the position of the 3D bounding box. Experiments conducted on the KITTI dataset indicate that the proposed SC exhibits the best speed-accuracy trade-off among advanced methods without using extra data.

Keywords: 3D object detection, stereo imagery, photometric alignment

1. Introduction

Three-dimensional (3D) object detection is a fundamental but challenging task in several fields such as robotics and autonomous driving [1, 2]. Several current mainstream 3D detectors rely on light detection and ranging (LiDAR) sensors to obtain accurate 3D information [3, 4, 5, 6], and the application of LiDAR data has been considered crucial to the success of 3D detectors. Despite their substantial success and emerging low-cost LiDAR studies, it is important to note that LiDAR still faces a few challenges, such as its high cost, short life, and limited perception. Conversely, stereo cameras, which work in a manner resemble human binocular vision [7], cost less and have higher resolutions; hence, they have garnered significant attention in academia and industry.

The basic theoretical knowledge of 3D detection based on stereo images can be traced back to triangulation [8] and the perspective n-point problem (pnp) [9]. Owing to the introduction of cumbersome datasets [10, 11, 12, 13], 3D pose estimation entered the era of object detection. To date, machine learning-based methods have been widely adopted in practical engineering [2, 14, 15]. However, these methods have limited the ability to search for information in 3D space without requiring additional information; hence, it is difficult for its accuracy to exceed that of deep learning-based methods.

Recently, a few high-precision 3D detection methods for stereo images have emerged [16, 17, 18, 19] and have divided the 3D detection task into two subtasks: depth prediction and two-dimensional (2D) object detection. Regarding the depth prediction subtask, a number of methods adopt high-performance disparity estimation networks to calculate disparity map of the entire image. Other methods apply instance segmentation to solely predict disparity for pixels on objects of interest. However, a stereo 3D detector, e.g., Disp R-CNN [20], has a disparity estimation network that takes more than one third of the entire detection process time. Regarding the 2D object detection subtask, it supports the conditions in which the objects in KITTI images are typically small and most of them, which are heavily occluded, limit the performance of most present bottom-up detectors [21]. To ensure accuracy, almost all stereo-based 3D detectors rely on anchor-based 2D detectors and the association approach.

However, anchor-based detectors and association approaches are faced by three primary limitations: 1) Hyperparameters of the anchor boxes, such as its size, aspect ratio, and number required to be carefully designed, that not only require relevant prior knowledge, but also influences the performance of the 2D detector. The existing method assigns the union of left and right GT boxes as the target for objectness classification. However, no research has been conducted on the parameters associated with stereo anchor boxes, and existing methods simply follow the parameters associated with monocular 2D detection. 2) Stereo anchor boxes incur additional processing steps and computation, such as scaling the stereo region of interest (ROI) using stereo ROI Align [22], utilizing non-maximal suppression (NMS) to suppress overlapping 2D bounding boxes, and adopting real bounding boxes [23] to compute the intersection of sets (IoU) scores. 3) Anchor-based detectors position the anchor...
boxes densely on the input image; however, only a part of the anchor boxes contain objects, triggering an imbalance between positive and negative samples, which does not only slow down the training, but may also lead to degenerate models. These factors limit the upper bound of the 3D object detection speed, which makes it difficult for existing stereo-based 3D object detection methods to achieve real-time performance.

Considering the above challenges, this study achieves the main task of stereo-based 3D detectors that are difficult to infer- ence in real-time. We apply the semantic and geometric information in the stereo images to propose an efficient 3D object detection method that combines deep learning and geometry named Stereo CenterNet (SC), as presented in Fig. 1. This method adopts an anchor-free single-stage structure. It solely adopts stereo RGB images as input, and does not rely on pixel-level mask annotations and LiDAR data as supervision. We propose a novel stereo objects association method after experimenting with various combinations to address the problem that existing stereo images 3D detectors are difficult to achieve in actual time. The method circumvents redundant information as much as possible, solely adds two detection heads, returns the information required for object association more directly, eliminates the anchor box constraint, simplifies the overall framework, and improves the general operational efficiency. In addition, we optimize the perspective key points classification strategy to improve the accuracy without network classification and reduce the computational effort of the framework. Finally, in our experiments, we inferred that the conventional dense alignment approach does not work optimally in the case of heavily occluded and truncated objects. We deduce that the occluded objects have negligible amounts of effective image information, which is not sufficient for optimizing the depth information. Therefore, we redesigned the objects screening strategy of the conventional dense alignment module that adaptively selects different 3D box estimation methods according to the object occlusion level, to further improve the detection accuracy of hard samples. Owing to these ideas, the proposed SC can achieve a faster inference time while ensuring accuracy.

Our contributions are summarized as follows:

- **An anchor-free 2D box association method.** To reduce calculation, we solely detect objects in the left image and perform left-right association by predicting a left-right distance.
- **An adaptive object screening strategy, which selects different 3D box alignment methods according to the occlusion level of the object.**
- **Evaluation on the KITTI dataset.** We propose a novel stereo-based 3D object detection method that does not require depth estimation and anchor boxes. We achieved a better speed-accuracy trade-off than other methods without using extra data.

2. Related Work

In this section, we briefly review recently presented literature on 3D object detection using monocular and stereo images. **Monocular image based methods:** GS3D [24] extract features from the visible surface of the 3D bounding box to solve the feature blur problem. Shi Xuepeng et al. [25] adopted scale information to formulate different detection tasks, and also proposed a fully convolutional cascade point regression method that restores the spatial size and direction of the object via the loss of projection consistency. RTM3D [26] predicts the nine perspective key points of the 3D bounding box of the image space, and adopts geometric relationships to restore the information of the object in the 3D space. MonoPair [27] considered the relationship between paired samples to improve monocular 3D object detection. SMOKE [28] detects the 3D center point of the object, and proposes a multi-step separation method to construct a 3D bounding box. According to Beker et al. [29], the 3D position and grid of each object in the image can be self-supervised using shape priors and instance masks. However, the monocular-image-based method finds it difficult to obtain accurate depth information.
Stereo images based methods: According to the type of training data, stereo-imagery-based methods can be generally divided into three types. The first type solely requires stereo images with corresponding annotated 3D bounding boxes. TLNet [30] enumerates several 3D anchors between the ROI in stereo images to construct object-level correspondences, and introduces a channel reweighting strategy that facilitates the learning process. Stereo R-CNN [16] converted the 3D object detection problem to left and right 2D box, key point detection, dense alignment, including other tasks, and adopted geometric relationship generation constraints to develop a 3D bounding box; however, it did not solve the detection problem of occluded objects. IDA-3D [18] predicts the depth of the object midpoint via the instance depth perception, parallax adaptation, and matching cost weighting methods, which can detect the 3D box end-to-end. Although these methods attempt to harness the potential of stereo images, its accuracy is the lowest among the three training types. The second type requires an additional depth map to train data. These methods [17, 31, 32] convert the estimated disparity map of the stereo images into pseudo LiDAR points, and adopt the LiDAR input method to estimate the 3D bounding box. DSGN [19] re-encodes 3D objects and can detect 3D objects end-to-end; however, although it exhibits the highest performance among the three methods, it has a slow inference speed. Another one adds the instance segmentation annotation on the basis of the Pseudo-LiDAR training data. To minimize the amount of calculation and eliminate the disparity estimation network triggering streaking artifacts, these methods [20, 33, 7] adopt the stereo instance segmentation network to extract the pixels on the object of interest and predict disparity. It is evident that more information will definitely facilitate higher detection performance. However, the pixel-level mask annotations are significantly heavier than the frame-level annotations required for object detection, and the high training cost hinders the deployment of stereo systems in practical applications.

3. Approach

The overall network, which is built on CenterNet [34], uses a weight-share backbone network to extract consistent features on left and right images architecture, as illustrated in Fig. 1. The network outputs 10 sub-branches behind the backbone network to complete two tasks: (A) stereo 2D detection and (B) stereo 3D Components. In task A, we used five sub-branches to estimate the left objects center, left objects center offset, left objects bounding box size, left and right center distances, and the width of the right objects box. The five sub-branches in task B estimated the orientation, dimension, bottom vertices, bottom vertices offset, vertices, and left objects center distance of the 3D bounding box. Subsequently, the objects in the image were dichotomized according to the level of occlusion, based on this information. Different methods are adopted to obtain the coordinates of objects in 3D space for heavily occluded objects and those that are not heavily occluded.

Figure 2: Center point heatmaps generated via different gaussian kernels. From left to right: original image, heatmap without aspect ratio, and heatmap with aspect ratio.

3.1. Stereo CenterNet

In this section, the backbone network of feature extraction is briefly discussed, and then the 2D and 3D detection modules are comprehensively introduced. Finally, the specific details of the implementation are introduced. Backbone. For testing, we adopted two different backbone networks: ResNet-18 [35] and DLA-34 [36]. The two RGB images \( i \) \( \in \mathbb{R}^{3 \times h \times w} \) of the stereo cameras were input into the model, and the down-sampled feature map was obtained four times, where \( w_i \) and \( h_i \) represent the width and height of the input image, respectively. Regarding ResNet-18, we adopted the keypoint feature pyramid network (KFPN) structure in RTM3D [26] to increase the key point feature extraction capability. For DLA-34, the same structure used in CenterNet [34] was adopted, and all hierarchical aggregation connections were replaced with deformable convnets networks (DCN) [37]. We connected the stereo feature maps and added a 1x1 convolutional layer to reduce the channel size. The number of channels output by ResNet-18 and DLA-34 were 128 and 256, respectively. Each of the 10 output sub-branches became connected to the backbone network with two convolutions of sizes \( 3 \times 3 \) (128/256) and \( 1 \times 1 \times n \), where \( n \) denotes the characteristic channel of the relevant output branch. All 10 sub-branches maintained the same characteristic width and height. Stereo 2D Detection. CenterNet [34] is a bottom-up anchor-free object detection method. The 2D bounding box of the object was constructed by predicting the center point of the object and the object’s width and height. CenterNet has three regression branches: center point coordinates, 2D size, and center offset. Similar to CenterNet, we adopted the center point of the left image as the main center connecting all functions, and by introducing new branches to detect and associate the 2D bounding boxes of the left and right images simultaneously, our stereo 2D detection has five branches: \( [(u_i, v_i), (w_i, h_i), off_f, dis, w_r] \), where \( (u_i, v_i) \) represent the coordinates of the center point of the left image 2D box; in addition, \( (w_i, h_i), off_f, dis, \) and \( w_r \) represent the width and height of the left 2D bounding box, local offset of the left image center point, distance between the right and left center points, and width of the 2D box on the right image, respectively. We adopted the corrected stereo images, and the left and right objects shared the height of the same 2D box; hence, it was not necessary to predict the right box height.
The difference is that we chose a Gaussian kernel with aspect ratio \(a \times b\) to replace the Gaussian kernel in CenterNet [34], as illustrated in figure 2. The new heat map agrees more with people’s intuitive judgments of objects, and can also provide more positive samples that can improve the detection accuracy of small objects. Regarding the main center point of the left object, we adopted a 2D Gaussian kernel \(Y_{\text{ave}} = \exp \left( -\frac{(u-v)^2}{2\sigma^2} \right) \) to generate a heat map \(Y \in [0, 1]^W \times H \times C \times N\), where \(\sigma_x = \frac{a}{6}, \sigma_y = \frac{b}{6}\), \(\alpha\) is an object size adaptive standard deviation [39]. \(C, R,\) and \(N\) denote the number of categories, downsampling multiple, and batch size, respectively. Generate a heat map of key points during the prediction \(\hat{Y} \in [0, 1]^W \times H \times C \times N\), and consider the predicted value \(\hat{Y}_{\text{ave}} = 1\) as a positive sample, and \(\hat{Y}_{\text{ave}} = 0\) as a negative sample. To address the imbalance problem of positive and negative samples, we adopted focal loss [40] to train.

\[
L_{\text{fa}} = -\frac{1}{N} \sum_{a} \left\{ \begin{array}{ll}
\ln \left( \frac{1}{1 + \exp(-\alpha \hat{Y}_{\text{ave}})} \right) & \text{if } Y_{\text{ave}} = 1 \\
\ln \left( 1 + \exp(-\beta (1 - \hat{Y}_{\text{ave}})) \right) & \text{otherwise}
\end{array} \right.
\]

(1)

Where \(N\) represents the number of key points in the image, while \(\alpha\) and \(\beta\) are hyperparameters set to 2 and 4, respectively, in the experiment.

We regressed the local offset of the left center point \(\hat{F} \in \mathbb{R}^{W \times H \times 2}\) for each left ground truth center point \((u_l, v_l)\) in \(\mathbb{R}^2\), and the offset and distance were trained with an L1 loss.

\[
L_{\text{off}} = \frac{1}{N} \sum_{(u_l, v_l)} \left\| \hat{F}_{(u_l, v_l)} - \frac{(u_l, v_l)}{R} - (u_l, v_l) \right\|
\]

(2)

Where \((u_l, v_l)\) denotes the left center point, which was downsampled and then rounded to the integer value. Next, we used L1 loss to predict a left 2D bounding box size \(S_k = (w_{\text{left}}^{(k)}, h_{\text{left}}^{(k)})\) for each left object \(k\). We adopted a single size prediction \(\hat{S} \in \mathbb{R}^{W \times H \times 2}\) for all object categories.

\[
L_{\text{size}} = \frac{1}{N} \sum_{k=1}^{N} |\hat{S}_k - S_k|
\]

(3)

To detect the right object, a feasible approach is to directly regress multiple identical detection heads after the left object detection. However, we decided to adopt another method to solely regress the information required to construct the 3D box, i.e., solely the left-right distance and width of the right object. Owing to perspective, the distance between the center of the left and right objects appears closer with the increase in depth. The primary motivation behind our approach is the idea that the left-right distance can be considered a special case of the left object center offset, thereby circumventing the increased computational effort of regressing other detection heads to obtain the right object center. In the ablation study (sect. 4.2) we provide experimental results to prove the validity of this idea. Hence, we regressed the distance \(D \in \mathbb{R}^{W \times H \times 2}\) between the left and right object centers when the ground truth center \((\bar{u}_r, \bar{v}_r)\) in \(\mathbb{R}^2\) of the right image object was known.

\[
L_{\text{dis}} = \frac{1}{N} \sum_{(u_r, v_r)} \left\| D_{(u_r, v_r)} - \frac{(u_r, v_r)}{R} - (u_r, v_r) \right\|
\]

(4)

To return to the width of the right-image object, similar to [41], we added an output transformation \(w_r = 1/\sigma (\bar{w}_r) - 1\) before L1 loss, where \(\sigma\) denotes the sigmoid function.

\[
L_{w_r} = \frac{1}{N} \sum_{k=1}^{N} \left| \frac{1}{\sigma (\bar{w}_r)} - 1 - w_r \right|
\]

(5)

For the inference, use the \(3 \times 3\) max pooling operation instead of NMS.

**Stereo 3D Components** To build a 3D bounding box, we added three regression branches \([d, o, v]\), where \(d\) denotes the length, the width and height of the 3D box of \([L, W, H]^T\), \(o\) represents the orientation of each object, and \(v\) denotes the bottom vertex of the 3D box as the key point. Although the constraints of these three branches can restore the 3D information of the object, the impact of key points on the result is crucial. To further improve the accuracy of the results, two additional optional branches are provided: bottom vertex offset and main center point-vertex distance.

Regarding the 3D dimensions of the object, we pre-calculated the average of the 3D dimensions of a category in the entire data set \([L, W, H]^T\), regressing to the true value and the offset between the prior size \(o f_{\text{dim}} = 2 \times ([L, W, H]^T - [L, W, H]^T)\). We trained the dimensions estimator using an L1 loss, and adopted the predicted size offset to restore the size of each object during inference.

\[
L_{\text{dim}} = \frac{1}{N} \sum_{k=1}^{N} \left| \alpha \hat{f}_k^{\text{dim}} - of_{\text{dim}} \right|
\]

(6)

For orientation, as illustrated in Figure 4, we regressed the car’s local direction \(\alpha\) instead of yaw rotation \(\theta\). Similar to [42], we generated a feature map that uses eight scalars to indicate orientation \(O \in \mathbb{R}^{W \times H \times 8}\), the orientation were trained with the L1 loss. Subsequently, we utilized \(\alpha\) and the object position to restore the yaw angle \(\theta\).

\[
\theta = \alpha + \arctan \left( \frac{x}{z} \right)
\]

(7)

To construct more stringent constraints, we predicted the four vertices at the bottom of the 3D bounding box, following [16],
and solely performed keypoint detection on the left image. We adopted the Gaussian kernel $V_{xy} = \exp\left(-\frac{(x-x')^2+(y-y')^2}{2\sigma^2}\right)$ to generate a ground truth vertex heat map $V \in [0, 1]^{\frac{w}{2} \times \frac{h}{2}}$, which is the same as the main center point of the left image, trained by focal loss. To improve the accuracy of the key points, we regressed the downsampling offset $F_y \in \mathbb{R}^{\frac{w}{2} \times \frac{h}{2}}$ of each vertex.

To correlate the vertices with the center of the left image, we also returned the distance $D_{xy} = \omega_{ff} + \omega_{oo} + \omega_{off} + \omega_{dim}$ of each vertex. For the parameter $\omega$ before each item, we adopted uncertainty weighting\cite{43} instead of manual tuning.

$$L = \omega_m L_{on} + \omega_{off} L_{off} + \omega_{dist} L_{dist} + \omega_{size} L_{size} +$$
$$+ \omega_2 L_{Wv} + \omega_{dim} L_{dim} + \omega_1 L_v + L_v$$
where $L_{on}$, $L_v$, $L_{off}$, and $L_{dist}$ represents orientation, vertex coordinate, vertex coordinate offset, and vertex coordinate distance, respectively. For the parameter $\omega$ before each item, we adopted uncertainty weighting\cite{43} instead of manual tuning.

$$u_1 = \begin{cases} x - \frac{w}{2} \cos \theta - \frac{h}{2} \sin \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_1 = \begin{cases} y - \frac{h}{2} \sin \theta - \frac{w}{2} \cos \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_2 = \begin{cases} x + \frac{w}{2} \cos \theta + \frac{h}{2} \sin \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_2 = \begin{cases} y + \frac{h}{2} \sin \theta + \frac{w}{2} \cos \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_3 = \begin{cases} x - \frac{w}{2} \cos \theta - \frac{h}{2} \sin \theta \left/ \left( z + \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_3 = \begin{cases} y - \frac{h}{2} \sin \theta + \frac{w}{2} \cos \theta \left/ \left( z + \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_4 = \begin{cases} x + \frac{w}{2} \cos \theta + \frac{h}{2} \sin \theta \left/ \left( z + \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_4 = \begin{cases} y + \frac{h}{2} \sin \theta - \frac{w}{2} \cos \theta \left/ \left( z + \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_p = \begin{cases} x + \frac{w}{2} \cos \theta - \frac{h}{2} \sin \theta \left/ \left( z - \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_p = \begin{cases} y + \frac{h}{2} \sin \theta + \frac{w}{2} \cos \theta \left/ \left( z - \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_{pp} = \begin{cases} x - \frac{w}{2} \cos \theta + \frac{h}{2} \sin \theta \left/ \left( z - \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_{pp} = \begin{cases} y - \frac{h}{2} \sin \theta - \frac{w}{2} \cos \theta \left/ \left( z - \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$

Where $b$ represents the baseline length of the stereo cameras, $w$, $h$, and $l$ denote the regression size, and $x, y, z$ represent the coordinates of the 3D bounding box’s center point. These multivariate equations were solved by the Gauss-Newton method.

**3.2. Stereo 3D Box Estimation**

**Geometric construction** After predicting the left and right 2D boxes, key points, angles, and 3D dimensions, following\cite{16}, the center position and horizontal direction $K = [x, y, z, \theta]$ of the 3D box can be solved by minimizing the reprojection error of the 2D box and key points.

As presented in the figure 1, among the four bottom vertices, they can be accurately projected to the middle of the 2D bounding box, and the bottom vertices used to construct the 3D box are called perspective keypoints\cite{16}. To identify the perspective key points in the four bottom vertices, we adopted logical judgment instead of image classification. As illustrated in Figure 4, we defined these four vertices as 0, 1, 2, 3, respectively, and the key points of perspective observed in different perspectives of the same car were different. However, in the same perspective, the key points of the perspective observed under the same car angle were the same. We deduce that the perspective key points are the same as the bottom vertex closest to the camera; hence, the perspective key points can be distinguished from the four vertices based on the predicted angle to estimate the 3D bounding box. The two key points of the boundary will be used for dense alignment. This method enables us to detect the keypoint of occluded and truncated objects, and improve the accuracy of hard samples.

According to the predicted results in Sect. 3.1, seven key values $Z = \{(u_l, v_l), (u_r, v_r), u_{p1}, u_{p2}, \ldots\}$ represent the upper left and lower right coordinates of the 2D box of the left image, the left and right abscissas of the 2D box in the right image, and the abscissa of the perspective key vertex. As presented in the figure 5, according to the sparse geometric relationship between 2D and 3D, a set of constraint equations can be constructed:

$$u_1 = \begin{cases} x - \frac{w}{2} \cos \theta - \frac{h}{2} \sin \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_1 = \begin{cases} y - \frac{h}{2} \sin \theta - \frac{w}{2} \cos \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_2 = \begin{cases} x + \frac{w}{2} \cos \theta + \frac{h}{2} \sin \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_2 = \begin{cases} y + \frac{h}{2} \sin \theta + \frac{w}{2} \cos \theta \left/ \left( z + \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_3 = \begin{cases} x - \frac{w}{2} \cos \theta - \frac{h}{2} \sin \theta \left/ \left( z - \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_3 = \begin{cases} y - \frac{h}{2} \sin \theta + \frac{w}{2} \cos \theta \left/ \left( z - \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_4 = \begin{cases} x + \frac{w}{2} \cos \theta + \frac{h}{2} \sin \theta \left/ \left( z - \frac{w}{2} \sin \theta + \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_4 = \begin{cases} y + \frac{h}{2} \sin \theta - \frac{w}{2} \cos \theta \left/ \left( z - \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$u_p = \begin{cases} x + \frac{w}{2} \cos \theta - \frac{h}{2} \sin \theta \left/ \left( z - \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$
$$v_p = \begin{cases} y + \frac{h}{2} \sin \theta + \frac{w}{2} \cos \theta \left/ \left( z - \frac{w}{2} \sin \theta - \frac{h}{2} \cos \theta \right) \right. \\
\end{cases}$$

**3D box estimation with stereo key-point constraints.**

**Dense Alignment** After obtaining the rough 3D bounding box in Sect 3.2, to further optimize the position of the 3D box in space, we adopt the dense alignment module of Peiliang Li et al.\cite{16}. Fig. 6 presents the detection result on a scene in the KITTI dataset. As illustrated in Fig. 6(b), when the object is severely occluded, the dense alignment module exhibits the wrong result. However, as presented in Fig. 6(a), the same phenomenon does not emerge in Stereo R-CNN. After checking the code, we inferred that a possible reason for this phenomenon is that Stereo R-CNN discarded the severely occluded samples during training. Therefore, it is challenging to maintain the complete training samples and circumvent the negative optimization of the dense alignment module.

To address this challenge, after the geometric construction module estimates the 3D bounding box, we classified these ob-
Algorithm 1: 3D Object Classification Strategy

Input: Number l of objects; The vertical coordinate of the upper left corner y1 and the lower right corner y2 of the box; A zero list depth_line with a length of 1280;
Output: Id of the occluded object o and the unoccluded object u

1. for i = 0; i ≤ l; i ++ do
   2. Read object depth coordinates z;
   3. for j = y1[i]; j ≤ y2[i]; j ++ do
      4. pixel = depth_line[j];
      5. if pixel = 0 then
         6. depth_line[j] = z;
      7. else if z < depth_line[j] then
         8. depth_line[j] = (z + pixel)/2;
   9. for k = 0; k ≤ l; k ++ do
      10. Read object depth coordinates z;
      11. left_y = right_y = True;
      12. if depth_line[y1[i]] < z then
          13. left_y = False;
      14. if depth_line[y2[i]] < z then
          15. right_y = False;
      16. if right_y = False and left_y = True then
          17. o ← k;
      18. else
          19. u ← k;

3.3 Implementation Details

Data Augmentation. We filled the original image 1280 × 384 for training and testing, then flipped the training set image horizontally, and swapped the images left and right to obtain a new stereo images; hence, our dataset was twice the original training number. The training data adopted the random scaling data enhancement method. Because the 3D information was inconsistent with the data enhancement, the proportional enhancement method was not suitable for the length, width, and height regressions.

Hyperparameter. By default, we set the downsampling factor $R = 4$. According to previous studies [44], we adopted the average size of the car as $[L, W, H] = [3.88, 1.63, 1.53]^\top$. Set $\alpha = 0.6$ in all the experiments. To prevent excessive vertex coordinates from overflowing the image during random scaling, we set the probability of random scaling to 0.5. In the inference process, we set thresholds 0.25 and 0.1 to filter the main center and perspective key points of the left image, respectively.

Training. We adopted the AdamW [45] optimizer on a NVIDIA Tesla V100 GPU for 45 epochs of training. The initial learning rate was set to $1.5 \times 10^{-4}$. The learning rate was reduced 10 times at the 40-th epoch. The backbone network was initialized by a classification model pre-trained on the ImageNet dataset. Regarding ResNet-18, we trained with 16 batch sizes for approximately 15 h. For DLA-34, we trained with 8 batch sizes for approximately 24 h.

4. Experiments

We evaluated the performance of our method on the popular KITTI 3D object detection dataset [11], which contains 7481 training images and 7518 testing images. Based on 3DOP [21], we split the training data into training set(3712 images) and validation set(3769 images). KITTI classifies objects into three levels: Easy, Moderate and Hard, according to the occlusion/truncation and object size of each object category in the 2D image. The methods were further evaluated using different IoU criteria per class(IoU ≥ 0.7 for Car and IoU ≥ 0.5 for Pedestrian and Cyclist). We mainly performed comparisons with other stereo images input methods in the car class, and a fully ablation study. We also benchmarked our results on the online KITTI test server.

4.1 Detection on KITTI

2D Detection Performance. We compared the proposed SC with other stereo 3D detectors that contain left and right associate components, where the stereo AP3D metric is defined by Stereo R-CNN [16]. As presented in Table 1, the proposed SC...
exhibits a higher detection precision on the left image than the CenterNet [34] in Easy/Moderate/Hard cases. In addition, the proposed SC correlated the left and right objects without extra computation. In the validation set, the performance of our left-right objects correlation method outperforms or has comparable results to Stereo R-CNN, while the AP\(_D\) of the moderate and hard cases in the left image increased by 1.35% and 9.57%, respectively.

It is important to note that it is difficult to accurately measure the performance of the box association method by solely reporting the stereo AP\(_D\). We observed that the Left AP\(_D\) is usually higher than the right and stereo AP\(_D\) in these methods; hence, we calculated the sum of the gap between the stereo and left AP\(_D\) in the three cases. As presented in Table 1, the proposed box association method has the smallest gap. In the test set, with DLA-34, the proposed SC obtains 96.61%/91.27%/83.50% and outperforms those of existing stereo SOTA studies. The detailed performance can be found online. Accurate stereo detection provides sufficient constraints for 3D box estimation.

### 3D Detection Performance

We compared SC with other stereo 3D detectors in Table 2. In addition to the average precision for the aerial view(AP\(_{BEV}\)) and 3D box(AP\(_{3D}\)), we also provided comparison of inference time. Specifically, our method outperforms all monocular image based 3D detector. For the validation set, the proposed SC surpasses all stereo methods, except those with extra labels. Among stereo 3D detectors without additional labels, using ResNet-18 backbone, the SC provides the best inference time and runs at 37 FPS. DLA34 runs at 23 FPS with 55.25%/41.44%/35.13% AP\(_{3D}\). This is twice as fast as IDA-3D and reports 0.32%/3.99%/2.90% AP\(_{3D}\), which is more accurate in the moderate case at 0.7 IoU.

As presented in Table 3, the proposed SC exhibits better accuracy than IDA-3D [18] on the KITTI test set. The proposed SC adaptively switches between geometric constraints and sub-pixel disparity estimation to achieve better depth estimation for occluded and truncated objects. This explains why AP\(_{BEV}\) and AP\(_{3D}\) improved better in moderate and hard cases. We also compared the proposed SC with stereo detectors that contain extra labels. It is inappropriate to perform comparisons with these methods; hence, we only list them for reference. Compared with the Pseudo-LiDAR [17] using F-PointNet [46] as the detector, the proposed SC does not only achieve comparable detection performance, but also its inference time is only 1/10.

We also present the results of the comparison with baseline, as illustrated in Figure 7. We report the single-model speed-accuracy of the proposed SC with different backbones. The variant with DLA-34 outperforms the best stereo key-point 3D detector, Stereo R-CNN (41.4% vs 36.6%) while running more than 4x faster. The variant with the same backbones far outperforms the accurate monocular keypoint 3D detector, RTM3D (41.4% AP\(_{3D}\) in 43 ms vs 16.9% AP\(_{3D}\) in 38 ms), although it is only approximately 0.2x the increase in reasoning time.

### 4.2. Ablation Study

In this section, we present the results of the experiments conducted by comparing different left-right association, training, key point, and dense alignment strategies. These experiments...
are performed in the car category of training/validation in the KITTI dataset. If not specified, the backbone for all experiments adopts the DLA-34 network.

**Left-Right Association Strategy.** CenterNet [34] adopts the center heatmap to detect objects; hence, it is difficult to associate left-right objects and circumvent additional computation. We tested two anchor-free associate strategies for two different Gaussian kernels: regress all detection heads on the right image (including rights center heatmap, rights center offset, right objects width and left-right distances) and solely regress right objects width and L-R distances. We experimented with different combinations to demonstrate their effects on detection performance, and the obtained results are presented in Table 4. The heatmap considering the aspect ratio improves the detection accuracy. No significant difference in accuracy exists between the two correlation methods, and the inference speed is increased by 2 FPS by solely using the L-R distance.

The result obtained from selecting all components is unsatisfactory. A possible reason is that the loss function contains several components from sub-branches; hence, the training procedure was complex and difficult to learn effective knowledge. Therefore, we solely predicted the right objects width and L-R distances to correlate the object, thereby reducing the amount of network calculations.

![Image](https://example.com/image.png)

**Figure 7:** Stereo 3D detection performance on KITTI (Moderate: IOU=0.7) validation set. The results of the comparison with baseline; the proposed SC achieves the best speed-accuracy trade-off.

**Table 3:** Car Localization and Detection. AP_{BEV}/AP_{3D} on test set. The italicizations in the method column indicates that the methods require extra labels, while the others are without extra labels.

| Sensor | Method                  | 3D Detection AP (%) | BEV Detection AP (%) |
|--------|-------------------------|--------------------|----------------------|
|        | Easy        | Moderate       | Hard     | Easy        | Moderate       | Hard     |
| LiDAR  |             |                |          |             |                |          |
| Mono   | M3D-RPN     | 14.76          | 9.71     | 7.42       | 21.02          | 13.67    | 10.23    |
| Mono   | RT3D        | 14.41          | 10.34    | 8.77       | 19.17          | 14.2     | 11.99    |
| Stereo | RT3D/Stereo | 29.9           | 23.28    | 18.96      | 58.81          | 46.82    | 38.38    |
| Stereo | PL-J3SD     | 54.53          | 34.05    | 29.25      | 67.3           | 45       | 38.4     |
| Stereo | TLNet       | 7.54           | 4.57     | 3.74       | 13.77          | 7.69     | 6.73     |
| Stereo | IDA-3D      | 45.09          | 29.32    | 23.13      | –              | –        | –        |
| Stereo | Stereo R-CNN| 47.58          | 30.23    | 23.72      | 61.92          | 41.31    | 33.42    |
| Stereo | SC(Ours)    | 49.84          | 31.3     | 25.62      | 65.97          | 42.12    | 35.37    |

**Table 4:** Results are evaluated at a moderate level with IOU = 0.7. Aspect Ratio stands for considering the aspect ratio of the heatmap in the kernel, and Right stands for the detection head containing the right object heatmap and center offset.

| w/ Aspect Ratio | w/ Right | FPS | AP_{3D}/AP_{BEV}/AP_{3D} |
|-----------------|----------|-----|--------------------------|
| × | × | 21 | 83.45 / 46.94 / 37.87   |
| √ | × | 23 | 88.65 / 50.68 / 38.44   |
| × | √ | 21 | 89.99 / 52.52 / 40.93   |
| √ | × | 23 | 89.83 / 53.27 / 41.44   |

**Training Strategy.** During training, we adopted three strategies to enhance model performance: flip enhancement, uncertainty weight [43], and AdamW [45] optimizer. Subsequently, we discussed their importance in our method, as presented in the Table 5. We conducted different combinations of experiments on the three strategies. Uncertain weight circumscribed the manual adjustment of weights while achieving better accuracy. The stereo flip data enhancement doubles the number of training set and alleviates the imbalance between positive and negative samples. The AdamW optimizer improves accuracy (+0.57% AP_{3D}) better than the Adam [47] optimizer, and improves training speed. These strategies have played positive roles in ensuring that the proposed SC achieved better performance.

**Table 5:** Evaluate the results at a moderate IOU level = 0.7. “Flip” implies adopting random horizontal flipping augmentation. “Uncert” means using uncertain weight. “AdamW” means applying AdamW optimizer. All experiments are conducted in the same settings.

| Flip | Uncert | AdamW | AP_{3D}/AP_{BEV}/AP_{3D} |
|------|--------|-------|--------------------------|
|      | √      |       | 79.10 / 42.97 / 30.20    |
|      | ×      |       | 81.24 / 42.72 / 32.88    |
| √    |        |       | 89.80 / 52.23 / 40.86    |
| √    |        |       | 89.80 / 52.23 / 40.87    |
| √    | √      |       | 89.83 / 53.27 / 41.44    |

**Keypoint Strategy.** We tested three detection methods for the key points at the bottom of the 3D box: direct regression to the center point depth, direct detection of the perspective key points, and detection and classification of the four vertices at the bottom. We compared AP_{3D} in the two IOU criteria. As presented in Table 6, we performed experiments on the three strategies. Four-point detection can provide higher accuracy, and is more conducive to the construction of 3D box. In addition to the perspective key points, the points can also provide pixel-level constraints for the dense alignment module, which is more appropriate for the optimization of the 3D box position.

**Table 6:** Detect different numbers of key points in our geometric construction module. “Without key points” means direct regression to the center point depth.

| method      | AP_{3D}(IOU=0.5) | AP_{3D}(IOU=0.7) |
|-------------|------------------|------------------|
|             | Easy | Moderate | Hard   | Easy | Moderate | Hard   |
| w/o kp      | 79.85 | 63.12    | 55.08  | 41.11 | 30.22    | 25.19  |
| w/ 1kp      | 86.41 | 73.57    | 59.16  | 52.10 | 39.01    | 32.76  |
| w/ 4kp      | 86.54 | 73.98    | 65.70  | 55.25 | 41.44    | 35.13  |
We conducted experiments in three cases: no dense alignment, dense alignment, and our dense alignment strategy. As shown in the Table 7, our dense alignment module strategy exhibits the highest accuracy and a better accuracy improvement on hard samples.

Table 7: Evaluations of no-dense, dense and our dense alignment modules in the KITTI validation set. We use ResNet-18 as the backbone.

| Method          | AP_{2D(INU=0.7)} | AP_{3D(INU=0.7)} |
|-----------------|------------------|------------------|
| w/o Alignment   | Easy             | Moderate         | Hard             |
|                 | 13.41            | 11.28            | 10.34            |
| w/ Alignment    | 59.21            | 43.87            | 36.59            |
| w/ Alignment(Ours) | 59.34          | 43.99            | 36.86            |

Pedestrian and Cyclist detection. In the KITTI object detection benchmark, the training samples of Pedestrian and Cyclist are limited; hence, it is more difficult than detecting car category. Because most image-based methods do not exhibit the evaluation results of Pedestrian and Cyclist, we solely report the available results of the original paper. We present the pedestrian and cyclist detection results on KITTI validation set in Table 8. In fact, PL-F-PointNet and DSGN [19] are both methods that utilize extra labels. The proposed SC exhibits more accurate results than these methos on AP_{2D}, but worse results on AP_{BEV} and AP_{3D}.

Table 8: Comparison of results for Pedestrian and Cyclist on KITTI validation set.

| Method          | AP_{2D} | AP_{BEV} | AP_{3D} |
|-----------------|---------|----------|---------|
| Pedestrian      | E / M / H | E / M / H | E / M / H |
| PL-F-PointNet   | 59.06 / 54.00 / 49.65 | 41.30 / 34.90 / 30.10 | 33.80 / 27.40 / 24.00 |
| DSGN            | 68.17 / 59.59 / 51.25 | 47.92 / 41.15 / 36.08 | 40.16 / 33.85 / 29.43 |
| SC(Ours)        | 76.16 / 51.10 / 50.39 | 47.60 / 39.90 / 27.00 | 41.50 / 25.20 / 24.90 |

5. Conclusion and Future Work

This research proposed SC, a faster 3D object detection method for stereo images. We addressed the 3D detection problem as a key point detection problem via a combination of learning and geometry. The proposed SC achieved the best speed-accuracy trade-off without requiring anchor-based 2D detection methods, depth estimation, and LiDAR monitoring. We infer that the image-based methods have substantial potential in the 3D field. However, the proposed framework only learned a few right-images information to facilitate geometric calculations, and the stereo feature network was not carefully designed to be slightly time-consuming. Therefore, we can attempt to further refine and simplify the framework by learning stereo information from single images while ensuring detection performance. Another main limitation of our method is that SC has not been tested in other scenarios except for the autonomous driving scenario; hence, the performance of the application in other scenarios remains unknown. In the future, we will test the application and deployment of this method in other scenarios such as indoor simultaneous localization and mapping (SLAM) [48] and remote surgery [49, 50, 51].

Acknowledgment

The authors acknowledge financial support from the National Natural Science Foundation of China under Grant No.161772068, the Finance science and technology project of Hainan province (No. ZDYF2019009), the China Postdoctoral Science Foundation under Grant No.2020M680352, the Guangdong Basic and Applied Basic Research Foundation under Grant No.2020A151510463, the Scientific and Technological Innovation Foundation of Shunde Graduate School, USTB under Grant No.2020BH011. The computing work is partly supported by USTB MatCom of Beijing Advanced Innovation Center for Materials Genome Engineering.
References

[1] Peiliang Li, Tong Qin, et al. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In Proceedings of the European Conference on Computer Vision (ECCV), pages 646–661, 2018.

[2] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3d object slam. IEEE Transactions on Robotics, 35(4):925–938, 2019.

[3] Shaoshuai Shi, Chaoxu Guo, Li Ji, Jie Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. P-vrcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10529–10538, 2020.

[4] Zetong Yang, Yanan Sun, Shu Liu, and Jiajia Jia. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11040–11048, 2020.

[5] Xiaozi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1907–1915, 2017.

[6] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiajia Jia. S3d: Sparse-to-dense 3d object detector for point cloud. In Proceedings of the IEEE International Conference on Computer Vision, pages 1951–1960, 2019.

[7] Zhenbo Xu, Wei Zhang, Xiaoqing Ye, Xiao Tan, Wei Yang, Shilei Wen, Errui Ding, Aijun Meng, and Liusheng Huang. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In AAAI, pages 12557–12564, 2020.

[8] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis., 47(1-3):7–42, 2002.

[9] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate ( Onion) solution to the pnp problem. Int. J. Comput. Vis., 81(2):155–165, 2009.

[10] Yu Xiang, Wenhui Kim, Wei Chen, Jingwei Li, Christopher B. Choy, Hao Su, Rooshbeh Mottaghi, Leonidas J. Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII, volume 9912 of Lecture Notes in Computer Science, pages 160–176. Springer, 2016.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8445–8453, 2019.

[12] Wanhui Peng, Hao Pan, He Liu, and Yi 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, pages 13015–13024, 2020.

[13] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12536–12545, 2020.

[14] Jiangwei Sun, Linghao Chen, Yiming Xie, Siyu Zhang, Qinhong Jiang, Xiaowei Zhou, and Hujun 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, pages 10548–10557, 2020.

[15] Xiaozi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. IEEE transactions on pattern analysis and machine intelligence, 40(5):1259–1272, 2017.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2980–2988. IEEE Computer Society, 2017.

[17] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9627–9636, 2019.

[18] Buyu Li, Wanli Ouyang, La Sheng, Xinrui Geng, and Xiaogang Wang. G3dn: An efficient 3d object detection framework for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1004–1012, 2019.

[19] Xuepeng Shu, Zhixiang Chen, and Tae-Kyun Kim. Distance-normalized unified representation for monocular 3d object detection. In European Conference on Computer Vision, pages 91–107. Springer, 2020.

[20] Peixuan Li, Huiuai Zhao, Pengfei Liu, and Fendao Cao. R3tn3d: Real-time monocular 3d detection from object keypoints for autonomous driving. arXiv preprint arXiv:2001.03343, 2020.

[21] Yongjiang Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12093–12102, 2020.

[22] Zechen Liu, Zizhong Wu, and Roland Töth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 996–997, 2020.

[23] Deniz Beker, Hiroharu Kato, Mihai Adrian Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Ahnld Gaidon. Monocular differentiable rendering for self-supervised 3d object detection. arXiv preprint arXiv:1904.14524, 2020.

[24] Zengyi Qin, Jingliu Wang, and Yan Lu. Triangulation learning network: from monocular to stereo 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7615–7623, 2019.

[25] Yurong You, Yan Wang, Wei-Lun Chao, Divyangsh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudolidar+: Accurate depth for 3d object detection in autonomous driving. arXiv preprint arXiv:1906.06310, 2019.

[26] Rui Qian, Divyangsh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudolidar for image-based 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7681–7690, 2020.

[27] Alex D Pon, Jason Ku, Chengyao Li, and Steven L. Waslander. Object-centric stereo matching for 3d object detection. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 8383–8389. IEEE, 2020.

[28] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. arXiv preprint arXiv:1904.07850, 2019.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
[36] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2403–2412, 2018. 3

[37] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9308–9316, 2019. 3

[38] Zili Liu, Tu Zheng, Guodong Xu, Zheng Yang, Haifeng Liu, and Deng Cai. Training-time-friendly network for real-time object detection. In AAAI, pages 11685–11692, 2020. 4

[39] Hei Law and Jia Deng. CornerNet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750, 2018. 4

[40] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017. 4

[41] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems, pages 2366–2374, 2014. 4

[42] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7074–7082, 2017. 4

[43] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7482–7491, 2018. 5, 8

[44] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 1991–1999, 2019. 6

[45] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 6, 8

[46] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 918–927. Computer Vision Foundation / IEEE Computer Society, 2018. 7

[47] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 8

[48] Yanmin Wu, Yunzhou Zhang, Delong Zhu, Yonghui Feng, Sonya Coleman, and Dermot Kerr. EAO-SLAM: monocular semi-dense object SLAM based on ensemble data association. In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 22-26, 2020, pages 1953–1960. IEEE, 2020. 9

[49] Hang Su, Andrea Mariani, Salih Ertug Ovar, Arianna Menciassi, Giancarlo Ferrigno, and Elena De Momi. Toward teaching by demonstration for robot-assisted minimally invasive surgery. IEEE Trans Autom. Sci. Eng., 18(2):484–494, 2021. 9

[50] Hang Su, Wen Qi, Chenguang Yang, Juan Sebastián Sandoval Arévalo, Giancarlo Ferrigno, and Elena De Momi. Deep neural network approach in robot tool dynamics identification for bilateral teleoperation. IEEE Robotics Autom. Lett., 5(2):2943–2949, 2020. 9

[51] Hang Su, Wen Qi, Yinghai Hu, Hamid Reza Karimi, Giancarlo Ferrigno, and Elena De Momi. An incremental learning framework for human-like redundancy optimization of anthropomorphic manipulators. IEEE Transactions on Industrial Informatics, 2020. 9