RTM3D: Real-time Monocular 3D Detection from Object Keypoints for Autonomous Driving

Peixuan Li[1,2,3,4,5][0000−0003−0931−1932], Huaici Zhao[1,2,4,5][0000−0002−7772−8652], Pengfei Liu[1,2,3,4,5][0000−0002−8432−711X], and Feidao Cao[1,2,3,4,5][0000−0002−9000−3963]

[1] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China
[2] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences, Shenyang, China
[5] Key Lab of Image Understanding and Computer Vision, Liaoning Province, Shenyang, China

Abstract. In this work, we propose an efficient and accurate monocular 3D detection framework in single shot. Most successful 3D detectors take the projection constraint from the 3D bounding box to the 2D box as an important component. Four edges of a 2D box provide only four constraints and the performance deteriorates dramatically with the small error of the 2D detector. Different from these approaches, our method predicts the nine perspective keypoints of a 3D bounding box in image space, and then utilize the geometric relationship of 3D and 2D perspectives to recover the dimension, location, and orientation in 3D space. In this method, the properties of the object can be predicted stably even when the estimation of keypoints is very noisy, which enables us to obtain fast detection speed with a small architecture. Training our method only uses the 3D properties of the object without any extra annotations, category-specific 3D shape priors, or depth maps. Our method is the first real-time system (FPS>24) for monocular image 3D detection while achieves state-of-the-art performance on the KITTI benchmark.

Keywords: Real-time Monocular 3D Detection, Autonomous Driving, Keypoint Detection

1 Introduction

3D object detection is an essential component of scene perception and motion prediction in autonomous driving [2, 9]. Currently, most powerful 3D detectors heavily rely on 3D LiDAR laser scanners for the reason that it can provide scene locations [8, 46, 41, 29]. However, the LiDAR-based systems are expensive and not conducive to embedding into the current vehicle shape. In comparison, monocular camera devices are cheaper and convenient which makes it drawing

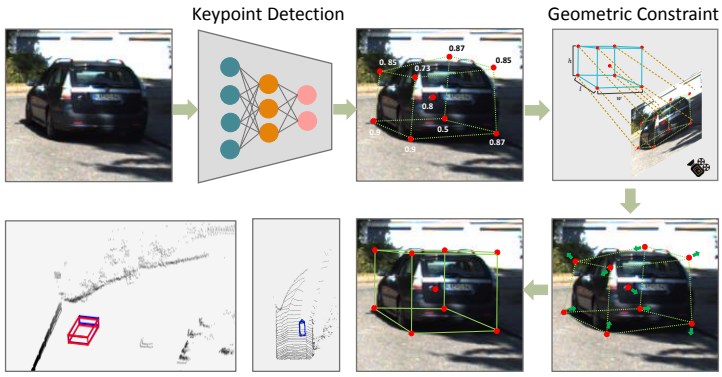

Fig. 1. Overview of proposed method: We first predict ordinal keypoints projected in the image space by eight vertexes and a central point of a 3D object. We then reformulate the estimation of the 3D bounding box as the problem of minimizing the energy function by using geometric constraints of perspective projection.

an increasing attention in many application scenarios [6, 26, 40]. In this paper, the scope of our research lies in 3D object detection from only monocular RGB image.

Monocular 3D object detection methods can be roughly divided into two categories by the type of training data: one imposes complex features, such as instance segmentation, category-specific shape prior and even depth map to select best proposals in multi-stage fusion module [6, 7, 40]. These features require additional annotation work to train some stand-alone networks which will consume plenty of computing resources in the training and inferring stages. Another one only employs 2D bounding box and properties of a 3D object as the supervised data [33, 3, 20, 42]. In this case, the most straightforward way build a deep regression network to directly predict the 3D information of the object, which caused the performance bottlenecks due to the large search space. To address this challenge, recent works have clearly pointed out that apply geometric constraints from 3D box vertexes to 2D box edges to refine or directly predict object parameters [28, 23, 3, 20, 26]. However, four edges of a 2D bounding box provide only four constraints on recovering a 3D bounding box while each vertex of a 3D bounding box might correspond to any edges in the 2D box, which will takes 4,096 of the same calculations to get one result [26]. Meanwhile, the strong reliance on the 2D box causes a sharp decline in 3D detection performance when predictions of 2D detectors even have a slight error. Therefore, most of these methods take advantage of two-stage detectors [11, 10, 32] to ensure the accuracy of 2D box prediction, which limit the upper-bound of the detection speed.

In this paper, we propose an efficient and accurate monocular 3D detection framework in the form of one-stage, which be tailored for 3D detection without relying on extra annotations, category-specific 3D shape priors, or depth maps.

Table 1. Comparison of the real-time status and the requirements of additional data in different image-based detection approaches.

Method	Real Time	Stereo	Depth	Shape/CAD	Segmentation
Mono3D [6]					✓
3DOP [7], stereoRCNN [20]		✓			
MF3D [40], Peseudo-LiDAR [37], MonoPSR [16] AM3D[25]			✓		
Mono3D++ [13],Deep-MANTA [5], 3DVP [38]				✓	
Deep3DBox [26],GS3D [19],MonoGRNet [31], FQNet[23], M3D-RPN[3] Shift-RCNN [28], MonoDIS [34]					
Ours(RTM3D)	✓				

The framework can be divided into two main parts, as shown in Fig. 1. First, we perform a one-stage fully convolutional architecture to efficiently predict 9 of the 2D keypoints which are projected points from 8 vertexes and central point of 3D bounding box. This 9 keypoints provides 18 geometric constrains on the 3D bounding box. Inspired by CenterNet [45], we model the relationship between the eight vertexes and the central point to solve the keypoints grouping and the vertexes order problem. The SIFT, SUFT and other traditional keypoint detection methods [24, 1]computed an image pyramid to solve the scale-invariant problem. A similar strategy was used by CenterNet as a post-processing step to further improve detection accuracy, which slows the inference speed. Note that the Feature Pyramid Network(FPN) [21] in 2D object detection is not applicable to the network of keypoint detection, because adjacent keypoints may overlap in the case of small-scale prediction. We propose a novel multi-scale pyramid of keypoint detection to generate a scale-space response. The final activate map of keypoints can be obtained by means of the soft-weighted pyramid. Given the 9 projected points, the next step is to minimize the reprojection error over the perspective of 3D points that parameterized by the location, dimension, and orientation of the object. We formulate the reprojection error as the form of multivariate equations in $\mathfrak{se}_3$ space, which can generate the detection results accurately and efficiently. We also discuss the effect of different prior information, such as dimension, orientation, and distance, predicted in parallel from our keypoint detection network. The prerequisite for obtaining this information is not to add too much computation so as not to affect the final detection speed. We model these priors and reprojection error term into an overall energy function in order to further improve 3D estimation.

To summarize, our main contributions are the following:

- We formulate the monocular 3D detection as the keypoint detection problem and combine the geometric constrains to generate properties of 3D objects more efficiently and accurately.
- We propose a novel one-stage and multi-scale network for 3D keypoint detection which provide the accurate project points for multi-scale object.
- We propose an overall energy function that can jointly optimize the prior and 3D object information.

– Evaluation on the KITTI benchmark, We are the first real-time 3D detection method using only images and achieves better accuracy under the same running time in comparing other competitors.

2 Related Work

Extra Data or Network for Image-based 3D Object Detection. In the last years, many studies develop the 3D detection in an image-based method for the reason that camera devices are more convenient and much cheaper. To complement the lacked depth information in image-based detection, most of the previous approaches heavily relied on the stand-alone network or additional labeling data, such as instance segmentation, stereo, wire-frame model, CAD prior , and depth, as shown in Table. 1. Among them, monocular 3D detection is a more challenging task due to the difficulty of obtaining reliable 3D information from a single image. One of the first examples [6] enumerate a multitude of 3D proposals from pre-defined space where the objects may appear as the geometrical heuristics. Then it takes the other complex prior, such as shape, instance segmentation, contextual feature, to filter out dreadful proposals and scoring them by a classifier. To make up for the lack of depth, [40] embed a pre-trained stand-alone module to estimate the disparity. The disparity map concatenates the front view representation to help the 2D proposal network and the 3D detection can be boosted by fusing the extracted feature after RoI pooling and point cloud. As a followup, [25] combines the 2D detector and monocular depth estimation model to obtain the 2D box and corresponding point cloud. The final 3D box can be obtained by the regression of PointNet [30] after the aggregation of the image feature and 3D point information through attention mechanism, which achieves the best performance in the monocular image. Intuitively, these methods would certainly increase the accuracy of the detection, but the additional network and annotated data would lead to more computation and labor-intensive work.

Image-only in Monocular 3D Object Detection. Recent works have tried to fully explore the potency of RGB images for 3D detection. Most of them include geometric constraints and 2D detectors to explicitly describe the 3D information of the object. [26] uses CNN to estimate the dimension and orientation extracted feature from the 2D box, then it proposes to obtain the location of an object by using the geometric constraints of the perspective relationship between 3D points and 2D box edges. This contribution is followed by most image-based detection methods either in refinement step or as direct calculation on 3D objects [20, 3]. All we know in this constraint is that certain 3D points are projected onto 2D edges, but the corresponding relationship and the exact location of the projection are not clear. Therefore, it needs to exhaustively enumerate $8^4 = 4096$ configurations to determine the final correspondence and can only provide four constraints, which is not sufficient for fully 3D representation in 9 parameters. It led to the need to estimate other prior information. Nevertheless, possible inaccuracies in the 2D bounding boxes may result in a grossly

inaccurate solution with a small number of constraints. Therefore, most of these methods obtain more accurate 2D box through a two-stage detector, which is difficult to get real-time speed.

Keypoints in Monocular 3D Object Detection. It is believed that the detection accuracy of occluded and truncated objects can be improved by deducing complete shapes from vehicle keypoints [5, 44, 27]. They represent the regular-shape vehicles as a wire-frame template , which is obtained from a large number of CAD models. To train the keypoint detection network, they need to re-label the data set and even use depth maps to enhance the detection capability. [13] is most related to our work, which also considers the wire-frame model as prior information. Furthermore, It jointly optimizes the 2D box, 2D keypoints, 3D orientation, scale hypotheses, shape hypotheses, and depth with four different networks. This has limitations in run time. In contrast to prior work, we reformulate the 3D detection as the coarse keypoints detection task. Instead of predicting the 3D box based on an off-the-shelf 2D detectors or other data generators, we build a network to predict 9 of 2D keypoints projected by vertexes and center of 3D bounding box while minimize the reprojection error to find an optimal result.

3 Proposed Method

In this section. We first describe the overall architecture for keypoint detection and prior property prediction. Then we detail how to estimate the 3D bounding box of the object by maintaining 2D-3D consistency.

3.1 Keypoint Detection Network

Our keypoint detection network takes an only RGB image as the input and simultaneously generates 2D-related information, such as perspective points and 2D size, and 3D-related information, such as dimension, orientation and distance. As shown in Fig. 2, it consists of three components: backbone, keypoint feature pyramid, and detection head. The main architecture adopts a one-stage strategy that shares a similar layout with the anchor-free 2D object detector [36, 15, 45, 18], which allows us to get a fast detection speed. Details of the network are given below.

Backbone. For the trade-off between speed and accuracy, we use two different structures as our backbones: ResNet-18[12] and DLA-34 [43]. All models take a single RGB image $I \in \mathbb{R}^{W \times H \times 3}$ and downsample the input with factor $S = 4$. The ResNet-18 and DLA-34 build for image classification network, the maximal downsample factor is $\times 32$. We upsample the bottleneck thrice by three bilinear interpolations and 1×1 convolutional layer. Before the upsampling layers, we concatenate the corresponding feature maps of the low level while adding one 1×1 convolutional layers for channel dimension reduction. After three upsampling layers, the channels are 256, 128, 64, respectively.

Keypoint Feature Pyramid. Keypoint in the image have no difference in size.

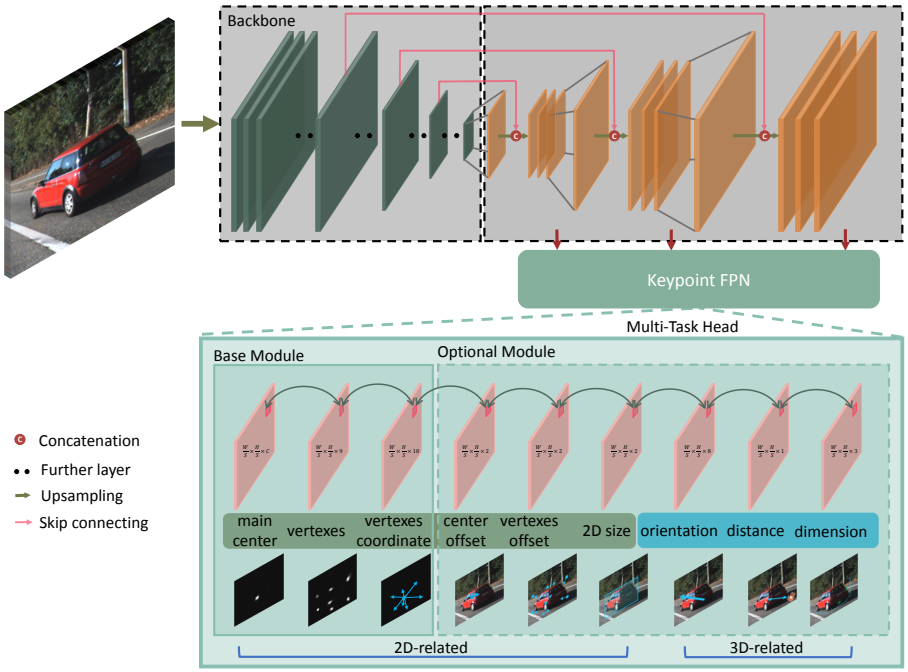

Fig. 2. An overview of proposed keypoint detection architecture: It takes only the RGB images as the input and outputs main center heatmap, vertexes heatmap, and vertexes coordinate as the base module to estimate 3D bounding box. It can also predict other alternative priors to further improve the performance of 3D detection.

Therefore, the keypoint detection is not suitable for using the Feature Pyramid Network(FPN) [21], which detect multi-scale 2D box in different pyramid layers. We propose a novel approach Keypoint Feature Pyramid Network (KFPN) to detect scale-invariant keypoints in the point-wise space, as shown in Fig. 3. Assuming we have F scale feature maps, we first resize each scale $f, 1 < f < F$ back to the size of maximal scale, which yields the feature maps $\hat{f}_{1<f<F}$. Then, we generate soft weight by a softmax operation to denote the importance of each scale. The finally scale-space score map S_{score} is obtained by linear weighing sum. In detail, it can be defined as:

$$S_{score} = \sum_{f} \hat{f} \odot softmax(\hat{f}) \tag{1}$$

where $\odot$ denote element-wise product.

Detection Head. The detection head is comprised of three fundamental components and six optional components which can be arbitrarily selected to boost the accuracy of 3D detection with a little computational consumption. Inspired by CenterNet [45], we take a keypoint as the maincenter for connecting all fea-

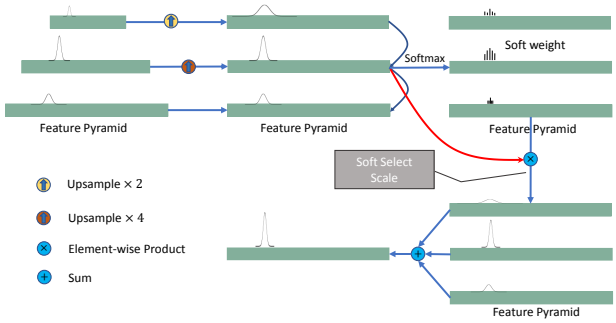

Fig. 3. Illustration of our keypoint feature pyramid network(KFPN).

tures. Since the 3D projection point of the object may exceed the image boundary in the case of truncation, the center point of the 2D box will be selected more appropriately. The heatmap can be define as $M \in [0,1]^{\frac{H}{S} \times \frac{W}{S} \times C}$, where C is the number of object categories. Another fundamental component is the heatmap $V \in [0,1]^{\frac{H}{S} \times \frac{W}{S} \times 9}$ of nine keypoints projected by vertexes and center of 3D bounding box. For keypoints association of one object, we also regress an local offset $V_c \in R^{\frac{H}{S} \times \frac{W}{S} \times 18}$ from the maincenter as an indication. Keypoints of V closest to the coordinates from V_c are taken as a group of one object.

Although the 18 constraints by the 9 keypoints have an ability to recover the 3D information of the object, more prior information can provide more constraints and further improve the detection performance. We offer a number of options to meet different needs for accuracy and speed. The center offset $M_{os} \in R^{\frac{H}{S} \times \frac{W}{S} \times 2}$ and vertexes offset $V_{os} \in R^{\frac{H}{S} \times \frac{W}{S} \times 2}$ are discretization error for each keypoint in heatmaps. The dimension $D \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S} \times 3}$ of 3D object have a smaller variance, which makes it easy to predict. The rotation $R(\theta)$ of an object only by parametrized by orientation θ (yaw) in the autonomous driving scene. We employ the *Multi-Bin* based method [26] to regress the local orientation. We classify the probability with cosin and sine offset of the local angle in one bin, which generates feature map of orientation $O \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S} \times 8}$ with two bins. We also regress the depth $Z \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S} \times 1}$ of 3D box center, which can be used as the initialization value to speed up the solution in Sec .3.2.

Training. All the heatmaps of keypoint and maincenter training strategy follow the [45, 18]. The loss solves the imbalance of positive and negative samples with focal loss [22]:

$$L_{kp}^{K} = -\frac{1}{N} \sum_{k=1}^{K} \sum_{x=1}^{H/S} \sum_{y=1}^{W/S} \begin{cases} (1-\hat{p}_{kxy})^{\alpha} log(\hat{p}_{kxy}) & if \quad p_{kxy} = 1 \\ (1-p_{kxy})^{\beta} \hat{p}_{kxy} log(1-\hat{p}_{kxy}) & otherwise \end{cases} \tag{2}$$

where K is the channels of different keypoints, $K = C$ in maincenter and $K = 9$ in keypoints. N is the number of maincenter or keypoints in an image, and α and β are the hyper-parameters to reduce the loss weight of negative and easy

positive samples. We set is $\alpha = 2$ and $\beta = 4$ in all experiments following [45, 18]. p_{kxy} can be defined by Gaussian kernel $p_{xy} = exp\left(-\frac{x^2+y^2}{2\sigma}\right)$ centered by ground truth keypoint $\tilde{p}_{xy}$. For σ, we find the max area A_{max} and min area A_{min} of 2D box in training data and set two hyper-parameters σ_{max} and σ_{min}. We then define the $\sigma = A(\frac{\sigma_{max}-\sigma_{min}}{A_{max}-A_{min}})$ for a object with size A. For regression of dimension and distance, we define the residual term as:

$$\begin{aligned} L_D &= \frac{1}{3N}\sum_{x=1}^{H/S}\sum_{y=1}^{W/S} \mathbb{1}_{xy}^{obj}\left(D_{xy} - \Delta\widetilde{D}_{xy}\right)^2 \\ L_Z &= \frac{1}{N}\sum_{x=1}^{H/S}\sum_{y=1}^{W/S} \mathbb{1}_{xy}^{obj}\left(log(Z_{xy}) - log(\widetilde{Z}_{xy})\right)^2 \end{aligned} \tag{3}$$

We set $\Delta\widetilde{D}_{xy} = log\frac{\widetilde{D}_{xy}-\bar{D}}{D_\sigma}$, where $\bar{D}$ and D_σ are the mean and standard deviation dimensions of training data. $\mathbb{1}_{xy}^{obj}$ denotes if maincenter appears in position x, y. The offset of maincenter, vertexes are trained with an L1 loss following [45]:

$$\begin{aligned} L_{off}^{m} &= \tfrac{1}{2N}\sum_{x=1}^{H/S}\sum_{y=1}^{W/S} \mathbb{1}_{xy}^{obj}\left|M_{os}^{xy} - \left(\tfrac{p^m}{S} - \tilde{p}_m\right)\right| \\ L_{off}^{v} &= \tfrac{1}{2N}\sum_{x=1}^{H/S}\sum_{y=1}^{W/S} \mathbb{1}_{xy}^{ver}\left|V_{os}^{xy} - \left(\tfrac{p^v}{S} - \tilde{p}_v\right)\right| \end{aligned} \tag{4}$$

where p^m, p^v are the position of maincenter and vertexes in the original image. The regression coordinate of vertexes with an L1 loss as:

$$L_{ver} = \frac{1}{N}\sum_{k=1}^{8}\sum_{x=1}^{H/S}\sum_{y=1}^{W/S} \mathbb{1}_{xy}^{ver}\left|V_c^{(2k-1):(2k)xy} - \left|\frac{p^v - p^m}{S}\right|\right| \tag{5}$$

The finial multi-task loss for keypoint detection define as:

$$\begin{aligned} L =& \omega_{main}L_{kp}^{C} + \omega_{kpver}L_{kp}^{8} + \omega_{ver}L_{ver} + \omega_{dim}L_D + \\ & \omega_{ori}L_{ori} + \omega_Z L_{dis} + \omega_{off}^{m}L_{off}^{m} + \omega_{off}^{v}L_{off}^{v} \end{aligned} \tag{6}$$

We empirical set $\omega_{main} = 1, \omega_{kpver} = 1, \omega_{ver} = 1, \omega_{dim} = 1, \omega_{ori} = 0.5, \omega_{dis} = 0.1, \omega_{off}^{m} = 0.5$ and $\omega_{off}^{v} = 0.5$ in our experimental.

3.2 3D Bounding Box Estimate

We estimate the 3D bounding box by enforcing the 2D-3D consistency between estimated 2D-related and 3D-related information, given by our keypoint detection network. Considering an image I, a set of $i = 1...N$ object are represented by 9 keypoints and other optional prior, keypoints as $\widehat{kp}_{ij}$ for $j \in 1...9$, dimension as $\widehat{D}_i$, orientation as $\hat{\theta}_i$, and distance as $\widehat{Z}_i$. The corresponding 3D bounding box B_i can be defined by its rotation $R_i(\theta)$, position $T_i = [T_i^x, T_i^y, T_i^z]^T$, and

dimensions $D_i = [h_i, w_i, l_i]^T$. Our goal is to estimate the 3D bounding box B_i, whose projections of 3D center and vertexes on the image space best fit the corresponding 2D keypoints $\widehat{kp}_{ij}$. This can be solved by minimize the reprojection error of 3D keypoints and 2D keypoints. We formulate it and other prior errors as a nonlinear least squares optimization problem:

$$\begin{aligned} R^*, T^*, D^* = &\arg\min_{\{R,T,D\}} \sum_{R_i,T_i,D_i} \left\| e_{cp}\left(R_i, T_i, D_i, \widehat{kp}_i\right)\right\|^2_{\Sigma_i} \\ &+\omega_d \left\| e_d\left(D_i, \widehat{D}_i\right)\right\|^2_2 + \omega_r \left\| e_r\left(R_i, \hat{\theta}_i\right)\right\|^2_2 \end{aligned} \tag{7}$$

where $e_{cp}(..), e_d(..), e_r(..)$ are measurement error of camera-point, dimension prior and orientation prior respectively. We set $\omega_d = 1$ and $\omega_r = 1$ in our experimental. Σ is the covariance matrix of keypoints projection error. It is the confidence extracted from the heatmap corresponding to the keypoints:

$$\Sigma_i = diag(softmax(V(\widehat{kp}_i)) \tag{8}$$

In the rest of the section, we will first define this error item, and then introduce the way to optimize the formulation.

Camera-Point. Following the [9], the homogeneous coordinate of eight vertexes and 3D center can be parametrized as:

$$\begin{aligned} P^i_{3D} &= diag(D_i)Cor \\ Cor &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 & 1/2 & -1/2 & -1/2 & 1/2 & 0 \\ 1/2 & 1/2 & -1/2 & -1/2 & 1/2 & 1/2 & -1/2 & -1/2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{aligned} \tag{9}$$

Given the camera intrinsics matrix K, the projection of these 3D points into the image coordinate is:

$$kp_i = \frac{1}{s_i} K \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} diag(D_i)Cor = \frac{1}{s_i} K \exp(\xi^\wedge) diag(D_i)Cor \tag{10}$$

where $\xi \in \mathfrak{se}_3$ and exp maps the $\mathfrak{se}_3$ into SE_3 space. The projection coordinate should fit tightly into 2D keypoints detected by the detection network. Therefore, the camera-point error is then defined as:

$$e_{cp} = \widehat{kp}_i - kp_i \tag{11}$$

Minimizing the camera-point error needs the Jacobians in $\mathfrak{se}_3$ space. It is given by:

$$\begin{aligned} \frac{\partial e_{cp}}{\partial \delta\xi} &= - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} 0 & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \cdot \left[I, -P'^\wedge\right] \\ \frac{\partial e_{cp}}{\partial D_i} &= -\frac{1}{9} \sum_{col=1}^{9} \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} 0 & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \cdot R \cdot Cor_{col} \end{aligned} \tag{12}$$

where $P^{'} = [X^{'}, Y^{'}, Z^{'}]^T = (\exp(\xi^{\wedge} P))_{1:3}$.
Dimension-Prior: The e_d is simply defined as:

$$e_d = \widehat{D}_i - D_i \tag{13}$$

Rotation-Prior: We define e_r in $SE3$ space and use log to map the error into its tangent vector space:

$$e_r = \log(R^{-1} R(\hat{\theta}))^{\vee}_{\mathfrak{se}_3} \tag{14}$$

These multivariate equations can be solved via the Gauss-newton or Levenberg-Marquardt algorithm in the g2o library [17]. A good initialisation is mandatory using this optimization strategy. We adopt the prior information generated by keypoint detection network as the initialization value, which is very important in improving the detection speed.

4 Experimental

4.1 Implementation Details

Our experiments were evaluated on the KITTI 3D detection benchmark [9], which has a total of 7481 training images and 7518 test images. We follow the [7] and [39] to split the training set as $train1, val_1$ and $train2, val_2$ respectively, and comprehensively compare our framework with other methods on this two validation as well as test set.

Our deep neural network implemented by using PyTorch with the machine i7-8086K CPU and 2 1080Ti GPUs. All the original image are padded to 1280×384 for training and testing. We project the 3D bounding box of Ground Truths in the left and right images to obtain Ground Truth keypoints and use the horizontal flipping as the data augmentation, which makes our dataset is quadruple with the origin training set. We run Adam [14] optimizer with a base learning rate of 0.0002 for 300 epochs and reduce $10\times$ at 150 and 180 epochs. For standard deviation of Gaussian kernel, we set $\sigma_{max} = 19$ and $\sigma_{min} = 3$. Based on the statistics of KITTI dataset, we set $\tilde{l} = 3.89, \tilde{w} = 1.62, \tilde{h} = 1.53$ and $\sigma_{\tilde{l}} = 0.41, \sigma_{\tilde{w}} = 0.1, \sigma_{\tilde{h}} = 0.13$. In the inference step, after 3×3 max pooling, we filter the maincenter and keypoints with threshold 0.4 and 0.1 respectively, and only keypoints that in the image size range are sent into the geometric constraint module. The backbone networks are initialized by a classification model pre-trained on the ImageNet data set. Finally, The ResNet-18 takes about three days with batch size 30 and DLA-34 for four days with batch size 16 in the training stage.

4.2 Comparison with Other Methods

To fully evaluate the performance of our keypoint-based method, for each task three official evaluation metrics be reported in KITTI: average precision for 3D

Table 2. Comparison of our framework with current image-based 3D detection methods for car category evaluated using metric AP_{3D} on val_1 / val_2 of KITTI data set. "Extra" means the extra data used in training. Red denotes the highest result, blue for the second highest, and cyan for the third.

Method	Extra	Time	AP_{3D} (IoU=0.5)			AP_{3D} (IoU=0.7)		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3D [6]	Mask	4.2 s	25.19 / -	18.20 / -	15.52 / -	2.53 / -	2.31 / -	2.31 / -
3DOP [7]	Stereo	3 s	46.04 / -	34.63 / -	30.09 / -	6.55 / -	5.07 / -	4.10 / -
MF3D [40]	Depth	-	47.88 / 45.57	29.48 / 30.03	26.44 / 23.95	10.53 / 7.85	5.69 / 5.39	5.39 / 4.73
Mono3D++ [13]	Depth+Shape	>0.6s	42.00 / -	29.80 / -	24.20 / -	10.60 / -	7.90 / -	5.70 / -
GS3D [19]	None	2.3s	32.15 / 30.60	29.89 / 26.40	26.19 / 22.89	13.46 / 11.63	10.97 / 10.51	10.38 / 10.51
Deep3DBox [26]	None	-	27.04 / -	20.55 / -	15.88 / -	5.85 / -	4.10 / -	3.84 / -
MonoGRNet [31]	None	**0.06s**	**50.51**/ -	**36.97**/ -	30.82 / -	13.88 / -	10.19 / -	7.62 / -
FQNet[23]	None	3.33s	28.16 / 28.98	21.02 / 20.71	19.91 / 18.59	5.98 / 5.45	5.50 / 5.11	4.75 / 4.45
M3D-RPN [3]	None	0.16s	**48.96**/**49.89**	**39.57**/**36.14**	**33.01**/ 28.98	**20.27**/**20.40**	**17.06**/**16.48**	**15.21**/**13.34**
Ours (ResNet18)	None	**0.035s**	47.43 /**46.52**	33.86 /**32.61**	**31.04**/**30.95**	**18.13**/**18.38**	**14.14**/**14.66**	**13.33**/**12.35**
Ours (DLA34)	None	**0.055s**	**54.36**/**52.59**	**41.90**/**40.96**	**35.84**/**34.95**	**20.77**/**19.47**	**16.86**/**16.29**	**16.63**/**15.57**

Table 3. Comparison of our framework with current image-based 3D detection frameworks for car category, evaluated using metric AP_{BEV} on val_1 / val_2 of KITTI data set.

Method	Extra	Time	AP_{BEV} (IoU=0.5)			AP_{BEV} (IoU=0.7)		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3D [6]	Mask	4.2 s	30.50 / -	22.39 / -	19.16 / -	5.22 / -	5.19 / -	4.13 / -
3DOP [7]	Stereo	3 s	55.04 / -	41.25 / -	34.55 / -	12.63 / -	9.49 / -	7.59 / -
MF3D [40]	Depth	-	55.02 / 54.18	36.73 / 38.06	31.27 / 31.46	22.03 / 19.20	13.63 / 12.17	11.60 / 10.89
Mono3D++ [13]	Depth+Shape	>0.6s	46.70 / -	34.30 / -	28.10 / -	16.70 / -	11.50 / -	10.10 / -
GS3D [19]	None	2.3s	- / -	- / -	- / -	- / -	- / -	- / -
Deep3DBox [26]	None	-	30.02 / -	23.77 / -	18.83 / -	9.99 / -	7.71 / -	5.30 / -
MonoGRNet [31]	None	**0.06s**	- / -	- / -	- / -	- / -	- / -	- / -
FQNet[23]	None	3.33s	32.57 / 33.37	24.60 / 26.29	21.25 / 21.57	9.50 / 10.45	8.02 / 8.59	7.71 / 7.43
M3D-RPN [3]	None	0.16s	**55.37**/**55.87**	**42.49**/**41.36**	**35.29**/**34.08**	**25.94**/**26.86**	**21.18**/**21.15**	**17.90**/**17.14**
Ours(ResNet18)	None	**0.035s**	**52.79**/**41.91**	**35.92**/**34.28**	**33.02**/**28.88**	**20.81**/**21.34**	**16.60**/**16.48**	**15.80**/**15.45**
Ours (DLA34)	None	**0.055s**	**57.47**/**56.90**	**44.16**/**44.69**	**42.31**/**41.75**	**25.56**/**24.74**	**22.12**/**22.03**	**20.91**/**18.05**

intersection-over-union (AP_{3D}), average precision for Birds Eye View (AP_{BEV}), and Average Orientation Similarity (AOS) if 2D bounding box available. We evaluate our method at three difficulty settings: easy, moderate, and hard, according to the object's occlusion, truncation, and height in the image space [9]. ***AP_{3D}* and *AP_{BEV}***. We compare our method with current image-based SOTA approaches and also provide a comparison about running time. However, it is not realistic to list the running times of all previous methods because most of them do not report their efficiency. The results AP_{3D}, AP_{BEV} and running time are shown in Table 2 and 3, respectively. ResNet-18 as the backbone achieves the best speed while our accuracy outperforms most of the image-only method. In particular, it is more than 100 times faster than Mono3D [6] while outperforms over 10% for both AP_{BEV} and AP_{3d} across all datasets. In addition, our ResNet-18 method is more than 75 times faster while having a comparable accuracy than 3DOP [7], which employs stereo images as the input. DLA-34 as the backbone achieves the best accuracy while having relatively good speed. It is faster about 3

times than the recently proposed M3D-RPN [3] while achieves the improvement in most of the metrics. Note that comparing our method with this all approaches is unfair because most of these approaches rely on extra stand-alone network or data in addition to monocular images. Nevertheless, we achieve the best speed with better performance.

Fig. 4. Qualitative results of our 3D detection. From top to bottom are keypoints, projections of the 3D bounding box and bird's eye view image, ground truths in green and our results in blue. The crimson arrows, green arrows, and red arrows point to occluded, distant, and truncated objects, respectively.

Results on the KITTI testing set. We also evaluate our results on the KITTI testing set, as shown in Table. 4.

4.3 Qualitative Results

Fig. 4 shows some qualitative results of our method. We visualize the keypoint detection network outputs, geometric constraint module outputs and BEV images. The results of the projected 3D box on image demonstrate than our method can handle crowded and truncated objects. The results of the BEV image show that our method has an accuracy localization in different scenes.

Table 4. Comparing 3D detection AP_{3D} on KITTI testing set. We use the DLA-34 as the backbone.

Method	Accelerater	Time	AP_{3D}(IoU=0.7)		
			Easy	Mode	Hard
GS3D[19]	-	2.3s	7.69	6.29	6.16
MonoGRNet[31]	Tesla P40	**0.06s**	9.61	5.74	4.25
M3D-RPN[3]	1080Ti	0.16s	**14.76**	**9.71**	**7.42**
FQNet[23]	1080Ti	3.33s	2.77	1.51	1.01
MonoDIS[34]	V100	-	10.37	7.94	6.40
Ours(DLA34)	1080Ti	**0.055s**	**14.41**	**10.34**	**8.77**

4.4 Ablation Study

Effect of Optional Components. Three optional components be employed to enhance our method: dimension, orientation, distance and keypoints offset. We experiment with different combinations to demonstrate their effect on 3D detection. The results are shown in Table. 5, we train our network with DLA-34 backbone and evaluate it using AP_{3D} and AP_{BEV}. The combinations of dimension, orientation, distance and keypoints offset achieve the best accuracy meanwhile have a faster running speed. This is because we take the output predicted by our network as the initial value of the geometric optimization module, which can reduce the search space of the gradient descent method.

Table 5. Ablation study of different optional selecting results on val_1 set. We use the DLA-34 as the backbone.

dim	ori	dist	off	time(s)	AP_{3D}(IoU=0.5)			AP_{3D}(IoU=0.7)		
					Easy	Mode	Hard	Easy	Mode	Hard
√				0.058	51.21	40.73	35.00	18.23	17.05	15.94
	√			0.061	25.35	22.33	21.18	3.12	3.43	2.97
√	√			0.057	54.18	41.34	34.89	20.23	16.02	15.94
√	√	√		0.055	54.20	41.56	35.13	20.76	16.80	16.25
√	√	√	√	0.055	54.36	41.90	35.84	20.77	16.86	16.36

Effect of Keypoint FPN. We propose keypoint FPN as a strategy to improve the performance of multi-scale keypoint detection. To better understand its effect, we compare the AP_{3D} and AP_{BEV} with and without KFPN. The details are shown in Table. 6, using KFPN achieves the improvement across all sets while no significant change in time consumption. **2D Detection and Orientation.** Although our focus is on 3D detection, we also report the performance of our methods in 2D detection and orientation evaluation in order to better understand the comprehensive capabilities of our approach. We report the AOS and AP with a threshold IoU=0.7 for comparison. The results are shown in Table. 7,

Table 6. Comparing 3D detection AP_{3D} of w/o KFPN and w/ KFPN for car category on val_1 set. We use the DLA-34 as the backbone.

KFPN	time	AP_{3D}(IoU=0.7)			AP_{3D}(IoU=0.5)		
		Easy	Mode	Hard	Easy	Mode	Hard
w/o	0.054	50.14	40.73	34.94	17.47	15.99	15.36
w/	0.055	54.36	41.90	35.84	20.77	16.86	16.36

Table 7. Comparing of 2D detection AP_{2D} with IoU=0.7 and orientation AOS results for car category evaluated on val_1 / val_2 of KITTI data set. Only the results under the moderate criteria are shown. Ours(2D) represents the results from the keypoint detection network, and Ours(3D) is the 2D bounding box of the projected 3D box.

Method	AP_{2D}	AOS
Mono3D [6]	88.67 / -	86.28 / -
3DOP [7]	88.07 / -	85.80 / -
Deep3DBox [26]	- /**97.20**	- /**96.68**
DeepMANTA [5]	**90.89**/ 91.01	**90.66**/ 90.66
GS3D [19]	88.85 / 90.02	87.52 / 89.13
Ours(2D)	90.14 / 91.85	89.58 / 89.22
Ours(3D)	90.41 / 92.08	89.95 / 89.40

the Deep3DBox train MS-CNN [4] in KITTI to produce 2D bounding box and adopt VGG16 [35] for orientation prediction, which gives him the highest accuracy. Deep3Dbox takes advantage of better 2D detectors, however, our AP_{3D} outperforms it by about 20% in moderate sets, which emphasize the importance of customizing the network specifically for 3D detection. Another interesting finding is that the 2D accuracy of back-projection 3D results is better than the direct prediction, thanks to our method that can infer the occlusive area of the object.

5 Conclusion

In this paper, we have proposed a faster and more accurate monocular 3D object detection method for autonomous driving scenarios. We reformulate 3D detection as the keypoint detection problem and show how to recover the 3D bounding box by using keypoints and geometric constraints. We specially customize the point detection network for 3D detection, which can simultaneously predict keypoints of the 3D box and other prior information of the object using only images. Our geometry module formulates this prior to easy-to-optimize loss functions. Our approach generates a stable and accurate 3D bounding box without containing stand-alone networks, additional annotation while achieving real-time running speed.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision. pp. 404–417. Springer (2006)
2. Behl, A., Hosseini Jafari, O., Karthik Mustikovela, S., Abu Alhaija, H., Rother, C., Geiger, A.: Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2574–2583 (2017)
3. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. Seoul, South Korea (2019)
4. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: european conference on computer vision. pp. 354–370. Springer (2016)
5. Chabot, F., Chaouch, M., Rabarisoa, J., Teulière, C., Chateau, T.: 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. pp. 2040–2049 (2017)
6. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2147–2156 (2016)
7. Chen, X., Kundu, K., Zhu, Y., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals using stereo imagery for accurate object class detection. IEEE transactions on pattern analysis and machine intelligence **40**(5), 1259–1272 (2017)
8. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)
10. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. He, T., Soatto, S.: Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. arXiv preprint arXiv:1901.03446 (2019)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv: Learning (2014)
15. Kong, T., Sun, F., Liu, H., Jiang, Y., Shi, J.: Foveabox: Beyond anchor-based object detector. arXiv preprint arXiv:1904.03797 (2019)
16. Ku, J., Pon, A.D., Waslander, S.L.: Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11867–11876 (2019)
17. Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G 2 o: A general framework for graph optimization pp. 3607–3613 (2011)

18. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018)
19. Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X.: Gs3d: An efficient 3d object detection framework for autonomous driving. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
20. Li, P., Chen, X., Shen, S.: Stereo r-cnn based 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7644–7652 (2019)
21. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Liu, L., Lu, J., Xu, C., Tian, Q., Zhou, J.: Deep fitting degree scoring network for monocular 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1057–1066 (2019)
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
25. Ma, X., Wang, Z., Li, H., Zhang, P., Ouyang, W., Fan, X.: Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6851–6860 (2019)
26. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7074–7082 (2017)
27. Murthy, J.K., Krishna, G.S., Chhaya, F., Krishna, K.M.: Reconstructing vehicles from a single image: Shape priors for road scene understanding. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 724–731. IEEE (2017)
28. Naiden, A., Paunescu, V., Kim, G., Jeon, B., Leordeanu, M.: Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. 2019 IEEE International Conference on Image Processing (ICIP) pp. 61–65 (2019)
29. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 918–927 (2018)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
31. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8851–8858 (2019)
32. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
33. Rubino, C., Crocco, M., Del Bue, A.: 3d object localisation from multi-view image detections. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1281–1294 (2017)
34. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1991–1999 (2019)

35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
36. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. arXiv preprint arXiv:1904.01355 (2019)
37. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: 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. pp. 8445–8453 (2019)
38. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Data-driven 3d voxel patterns for object category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1903–1911 (2015)
39. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Subcategory-aware convolutional neural networks for object proposals and detection pp. 924–933 (2017)
40. Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2345–2353 (2018)
41. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7652–7660 (2018)
42. Yang, S., Scherer, S.: Cubeslam: Monocular 3-d object slam. IEEE Transactions on Robotics (2019)
43. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2403–2412 (2018)
44. Zeeshan Zia, M., Stark, M., Schindler, K.: Are cars just 3d boxes?-jointly estimating the 3d shape of multiple objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3678–3685 (2014)
45. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. In: arXiv preprint arXiv:1904.07850 (2019)
46. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)