Center3D: Center-Based Monocular 3D Object Detection with Joint Depth Understanding

Yunlei Tang[1(✉)], Sebastian Dorn[2], and Chiragkumar Savani[2]

[1] Technical University of Darmstadt, Darmstadt, Germany
harryyunlei@gmail.com
[2] Ingolstadt, Germany

Abstract. Localizing objects in 3D space and understanding their associated 3D properties is challenging given only monocular RGB images. The situation is compounded by the loss of depth information during perspective projection. We present Center3D, a one-stage anchor-free approach and an extension of CenterNet, to efficiently estimate 3D location and depth using only monocular RGB images. By exploiting the difference between 2D and 3D centers, we are able to estimate depth consistently. Center3D uses a combination of classification and regression to understand the hidden depth information more robustly than each method alone. Our method employs two joint approaches: (1) **LID**: a classification-dominated approach with sequential **L**inear **I**ncreasing **D**iscretization. (2) **DepJoint**: a regression-dominated approach with multiple Eigen's transformations [6] for depth estimation. Evaluating on KITTI dataset [8] for moderate objects, Center3D improved the AP in BEV from 29.7% to **43.5**%, and the AP in 3D from 18.6% to **40.5**%. Compared with state-of-the-art detectors, Center3D has achieved a better speed-accuracy trade-off in realtime monocular object detection.

1 Introduction and Related Work

3D object detection is currently one of the most challenging topics for both industry and academia. Applications of related developments can easily be found in the areas of robotics, autonomous driving [4,18,21] etc. The goal is to have agents with the ability to identify, localize, react, and interact with objects in their surroundings. 2D object detection approaches [11,17,26] achieved impressive results in the last decade. In contrast, inferring associated 3D properties from a 2D image turned out to be a challenging problem in computer vision, due to the intrinsic scale ambiguity of 2D objects and the lack of depth information. Hence many approaches involve additional sensors like LiDAR [20,23] or radar [22] to measure depth. However, there are reasons to prefer monocular-based approaches too. LiDAR has reduced range in adverse weather conditions, while visual information of a simple RGB camera is more dense and also more

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-71278-5_21) contains supplementary material, which is available to authorized users.

Z. Akata et al. (Eds.): DAGM GCPR 2020, LNCS 12544, pp. 289–302, 2021.
https://doi.org/10.1007/978-3-030-71278-5_21

robust under rain, snow, etc. Another reason is that cameras are currently significantly more economical than high precision LiDARs and are already available in *e.g.* robots, vehicles, etc. Additionally, the processing of single RGB images is much more efficient and faster than processing 3D point clouds in terms of CPU and memory utilization.

These compelling reasons have led to research exploring the possibility of 3D detection solely from monocular images [1,2,9,12,14,16,19]. The network structure of most 3D detectors starts with a 2D region proposal based (also called anchor based) approach, which enumerates an exhaustive set of predefined proposals over the image plane and classifies/regresses only within the region of interest (ROI). MonoGRNet [16] consists of parameter-specific subnetworks. All further regressions are guided by the detected 2D bounding box. M3D-RPN [1] demonstrates a single-shot model with a standalone 3D RPN, which generates 2D and 3D proposals simultaneously. Additionally, the specific design of depth-aware convolutional layers improved the network's 3D understanding. With the help of an external network, Multi-Fusion [24] estimates a disparity map and subsequently a LiDAR point cloud to improve 3D detection. Due to multi-stage or anchor-based pipelines, most of them perform slowly.

Most recently, to overcome the disadvantages above, 2D anchor-free approaches have been used by researchers [2,5,11,26]. They model objects with keypoints like centers, corners or points of interest of 2D bounding boxes. Anchor-free approaches are usually one-stage, thus eliminating the complexity of designing a set of anchor boxes and fine tuning hyperparameters. Our paper is also an extension of one of these works CenterNet: Objects as Points [26], which proposed a possibility to associate a 2D anchor free approach with a 3D detection.

Nevertheless, the performance of CenterNet is still restricted by the fact that a 2D bounding box and a 3D cuboid are sharing the same center point. In this paper we show the difference between the center points of 2D bounding boxes and the projected 3D center points of objects, which are almost never at the same image position. Comparing CenterNet, our main contributions are as follows: 1. We additionally regress the 3D centers from 2D centers to locate the objects in the image plane and in 3D space more properly. 2. By examining depth estimation in monocular images, we show that a combination of classification and regression explores visual clues better than using only a single approach. An overview of our approach is shown in Fig. 1.

We introduce two approaches to validate the second conclusion: (1) Motivated by DORN [7] we consider depth estimation as a sequential classification with residual regression. According to the statistics of the instances in the KITTI dataset, a novel discretization strategy is used. (2) We divide the complete depth range of objects into two bins, foreground and background, either with overlap or associated. Classifiers indicate which depth bin or bins the object belongs to. With the help of Eigen's transformation [6], two regressors are trained to gather specific features for closer and farther away objects, respectively. For illustration see the depth part in Fig. 1.

Compared to CenterNet, our approach improved the AP of easy, moderate, hard objects in BEV from 31.5, 29.7, 28.1 to **56.7**, **43.5**, **41.2**, in 3D space from 19.5, 18.6, 16.6 to **52.5**, **40.5**, **34.9**, which is comparable with state-of-the-art

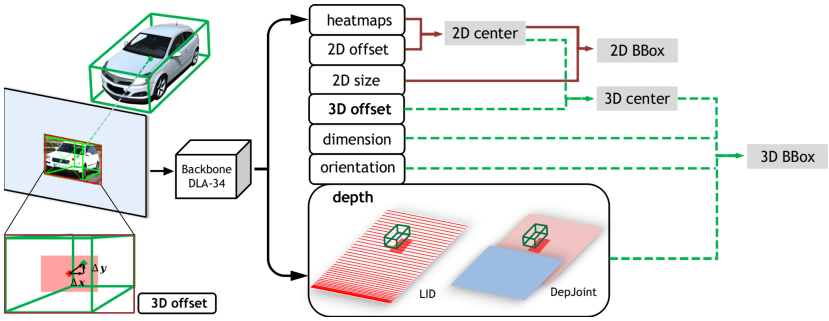

Fig. 1. Overview of Center3D. A monocular input image is fed to the backbone *DLA-34*, which generates feature maps. *Heatmaps* and *2D offset* are subsequently used to detect the *2D center* [26]. The latter is relocated by *3D offset* to propose the *3D center*, which is illustrated in the bottom-left of the figure. By applying a combination of regression and classification, *DepJoint* or *LID*, Center3D is inferring the *depth* of the associated *3D center*. *Depth*, together with regressed *dimensions*, *orientation*, and *3D center* are finally used to propose the *3D BBox*. Our contributions comparing CenterNet are indicated in bold or with dashed lines.

approaches. Center3D achieves a better speed-accuracy trade-off on the KITTI dataset in the field of monocular 3D object detection. Details are given in Table 1 and discussed in Sect. 3.

2 Center3D

2.1 CenterNet Baseline

The 3D detection approach of CenterNet described in [26] is the basis of our work. It models an object as a single point: the center of its 2D bounding box. For each input monocular RGB image, the original network produces a heatmap for each category, which is trained with focal loss [13]. The heatmap describes a confidence score for each location, the peaks in this heatmap thus represent the possible keypoints of objects. All other properties are then regressed and captured directly at the center locations on the feature maps respectively. For generating a complete 2D bounding box, in addition to width and height, a local offset will be regressed to capture the quantization error of the center point caused by the output stride. For 3D detection and localization, the additional abstract parameters, i.e. depth, 3D dimensions and orientation, will be estimated separately by adding a head for each of them. Following the output transformation of Eigen et al. [6] for depth estimation, CenterNet converts the feature output into an exponential area to suppress the depth space.

2.2 Regressing 3D Center Points

The 2D performance of CenterNet is very good, while the APs in 3D perform poorly, as the first row shown in Table 1. This is caused by the difference between

Fig. 2. 3D bounding box estimation on KITTI validation set. *first row:* the output of CenterNet. Projected 3D bounding boxes located around estimated 2D centers. The position of centers is generated by the peak of the Gaussian kernel on the heatmap. *second row:* the ground truth of input images. Here the 2D (red) and 3D (green) projected bounding boxes with their center points are shown. *third row:* the output of Center3D. The 3D cuboid is based on 3D center points shifted from 2D space with offset. More qualitative results can be found in the supplementary material. (Color figure online)

the center point of the visible 2D bounding box in the image and the projected center point of the complete object from physical 3D space. This is illustrated in the first two rows of Fig. 2. A center point of the 2D bounding box for training and inference is enough for detecting and decoding 2D properties, *e.g.* width and height, while all additionally regressed 3D properties, *e.g.* depth, dimension and orientation, should be consistently decoded from the projected 3D center of the object. The gap between 2D and 3D center points decreases for faraway objects and for objects which appear in the center area of the image plane. However the gap becomes significant for objects that are close to the camera or on the image boundary. Due to perspective projection, this offset will increase as vehicles get closer. Close objects are especially important for technical functions based on perception (e.g. in autonomous driving or robotics).

Hence we split the 2D and 3D tasks into separate parts, as shown in Fig. 1. Assuming that the centers of 2D bounding boxes is $\mathbf{c^i_{2D}} = (x^i_{2D}, y^i_{2D})$, and the 3D projected center points of cuboids from physical space is $\mathbf{c^i_{3D}} = (x^i_{3D}, y^i_{3D})$. We still locate an object with $\mathbf{c^i_{2D}}$, which is definitively included in the image, and determine the 2D bounding box of the visible part with w^i and h^i. For the 3D task we relocate $\mathbf{c^i_{3D}}$ by adding two head layers on top of the backbone and regress the offset $\mathbf{\Delta c^i} = (x^i_{3D} - x^i_{2D}, y^i_{3D} - y^i_{2D})$ from 2D to 3D centers. Given the projection matrix $\mathbf{P}$ in KITTI, we now determine the 3D location $\mathbf{C} = (X, Y, Z)$ by converting the transformation in homogeneous coordinates.

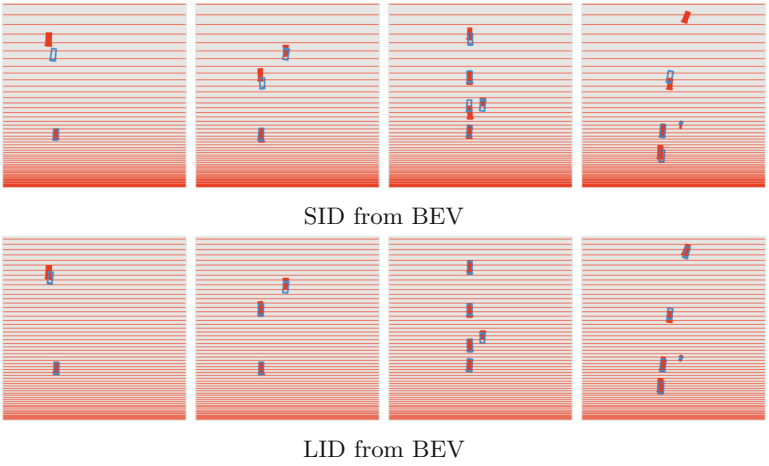

Fig. 3. The comparison of the discretization strategies LID (*first row*) and SID (*second row*) from BEV between 0 m and 54 m, with a setting of $d_{\text{min}} = 1\,\text{m}$, $d_{\text{max}} = 91\,\text{m}$ and $N = 80$. The solid red lines indicate the threshold of each bin, the solid rectangles represent the ground truth vehicles in BEV, while blue rectangles represent the estimations.

2.3 Enriching Depth Information

This section introduces two novel approaches to infer depth cues over monocular images: First, we adapt the advanced DORN [7] approach from pixel-wise to instance-wise depth estimation. We introduce a novel linear-increasing discretization (**LID**) strategy to divide continuous depth values into discrete ones, which distributes the bin sizes more evenly than spacing-increasing discretization (SID) in DORN. Additionally, we employ a residual regression for refinement of both discretization strategies. Second, with the help of a reference area (**RA**) we describe the depth estimation as a joint task of classification and regression (**DepJoint**) in exponential range.

LID Usually a faraway object with higher depth value and less visible features will induce a higher loss, which could dominate the training and increases uncertainty. On the other hand these targets are usually less important for functions based on object detection. To this end, DORN solves the ordinal regression problem by quantizing depth into discrete bins with SID strategy. It discretizes the given continuous depth interval $[d_{\text{min}}, d_{\text{max}}]$ in *log* space and hence down-weight the training loss in faraway regions with higher depth values, see Eq. 1. However, such a discretization often yields too dense bins within unnecessarily close range, where objects barely appear (as shown in Fig. 3 first row). According to the histogram in Fig. 4 most instances of the KITTI dataset are between 5 m and 80 m. Assuming that we discretize the range between $d_{\text{min}} = 1\,\text{m}$ and $d_{\text{max}} = 91\,\text{m}$ into $N = 80$ sub-intervals, 29 bins will be involved within just 5 m.

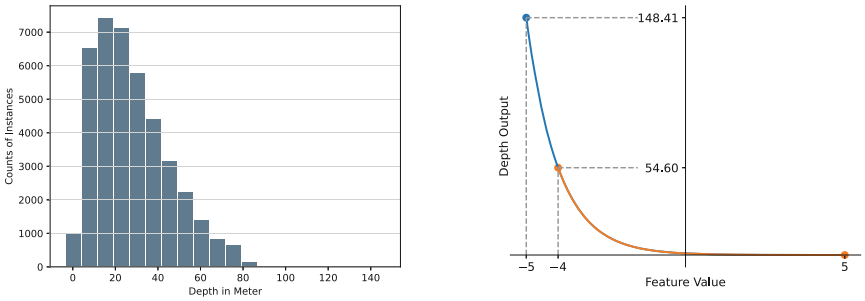

Fig. 4. *left*: Histogram of the depth. The analysis is based on instances in the KITTI dataset. *right*: Transformation of Eigen et al. [6] according to depth estimation. The x-axis indicates the feature output, and the y-axis is the depth output after transformation (given in meter).

Thus, we use the LID strategy to ensure the lengths of neighboring bins increase linearly instead of *log*-wise. For this purpose, assume the length of the first bin is δ. Then the length of the next bin is always δ longer than the previous bin. Now we can encode an instance depth d in $l_{\text{int}} = \lfloor l \rfloor$ ordinal bins according to LID and SID respectively. Additionally, we reserve and regress the residual decimal part $l_{\text{res}} = l - l_{\text{int}}$ for both discretization strategies:

$$\begin{aligned} &\text{SID:} \quad l = N \frac{\log d - \log d_{\max}}{\log d_{\max} - \log d_{\min}}, \\ &\text{LID:} \quad l = -0.5 + 0.5\sqrt{1 + \frac{8(d - d_{\min})}{\delta}}, \quad \delta = \frac{2(d_{\max} - d_{\min})}{N(1+N)}. \end{aligned} \tag{1}$$

During the inference phase, DORN counts the number of activated bins with probability higher than 0.5, as estimated by the ordinal label $\hat{l}_{\text{int}}$, and uses the median value of the $\hat{l}_{\text{int}}$-th bin as the estimated depth in meters. The notation of symbols with $\hat{\cdot}$ denotes the output of estimation. However, relying on discrete median values of bins only is not precise enough for instance localization. Hence we modify the training to be a combination of classification and regression. For classification we follow the ordinal loss with binary classification and add a shared layer to regress the residuals l_{res} additionally. Given an input RGB image $I \in \mathbb{R}^{W \times H \times 3}$, where W represents the width and H the height of I, we generate a depth feature map $\hat{D} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times (2N+1)}$, where R is the output stride. Backpropagation is only applied on the centers of 2D bounding boxes located at $\hat{\mathbf{c}}^{\mathbf{i}}_{\text{2D}}$, where $i \in \{0, 1, ..., K-1\}$ indicates the instance number of total K instances over the image. The final loss $\mathcal{L}_{\text{dep}}$ is defined as the sum of the residual loss $\mathcal{L}^i_{\text{res}}$ and ordinal loss $\mathcal{L}^i_{\text{ord}}$:

$$\begin{aligned}\mathcal{L}_{\text{dep}} &= \sum_{i=0}^{K-1}(\mathcal{L}_{\text{res}}^{i} + \mathcal{L}_{\text{ord}}^{i}), \quad \mathcal{L}_{\text{res}}^{i} = \text{SmL1}(\hat{l}_{\text{res}}^{i}, l_{\text{res}}^{i}), \\ \mathcal{L}_{\text{ord}}^{i} &= -\left(\sum_{n=0}^{l^i-1}\log \mathcal{P}_n^i + \sum_{n=l^i}^{N-1}\log(1-\mathcal{P}_n^i)\right), \quad \mathcal{P}_n^i = \mathcal{P}\left(\hat{l}^i > n\right),\end{aligned} \tag{2}$$

where $\mathcal{P}_n^i$ is the probability that the i-th instance is farther away than the n-th bin, and $SmL1$ represents the smooth L1 loss function [15]. During inference, the amount of activated bins will be counted up as $\hat{l}_{\text{int}}^i$. We refine the result by taking into account the residual part, $\hat{l} = \hat{l}_{\text{int}}^i + \hat{l}_{\text{res}}^i$, and decode the result by inverse-transformation of Eq. 1.

DepJoint The transformation described by Eigen et al. [6] converts the output depth to an exponential scale. It generates a depth feature map $\hat{D} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times 1}$. The output $\hat{d}$ at the estimated center point of a 2D bounding box $\hat{\mathbf{c}}_{\text{2D}}^{\mathbf{i}}$ is converted to $\Phi(\hat{d}) = e^{-\hat{d}}$. This enriches the depth information for closer objects by putting more feature values into smaller ranges. As shown on the right panel of Fig. 4, the feature map values between -4 and 5 correspond to a depth up to 54.60 m, while feature values corresponding to more distant objects up to 148.41 m account for only 10% of the feature output range $[-5, 5]$. The transformation is reasonable, since closer objects are of higher importance. Eigen's transformation shows an impressive precision on closer objects but disappoints on objects which are farther away. To improve on the latter, we introduce the DepJoint approach, which treats the depth estimation as a joint classification and regression. Compared to using Eigen's transformation solely, it also emphasizes the distant field. DepJoint divides the depth range $[d_{\min}, d_{\max}]$ in two bins with scale parameter α and β:

$$\begin{aligned}\text{Bin 1} &= [d_{\min},\ (1-\alpha)d_{\min} + \alpha d_{\max}], \\ \text{Bin 2} &= [(1-\beta)d_{\min} + \beta d_{\max},\ d_{\max}].\end{aligned} \tag{3}$$

Each bin will only be activated during training when the object lies within the appropriate interval. The first bin is used to regress the absolute value of depth d^i, while the second bin is used to regress the residual value of depth $\tilde{d}^i = d_{\max} - d^i$. With this transformation, a larger depth value will be supported with more features. We use the binary Cross-Entropy loss $CE_b(\cdot)$ for classification of each bin b and regress d^i and $\tilde{d}^i = d_{\max} - d^i$ with L1 loss $L1(\cdot)$ subsequent to an output transformation $\Phi(\cdot)$. Hence the output of the depth head is $\hat{D} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times 6}$ and the loss for training is defined as:

$$\begin{aligned}\mathcal{L}_{\text{dep}} &= \sum_{i=0}^{K-1}\left(\mathcal{L}_{\text{cls}}^{i} + \mathcal{L}_{\text{reg}}^{i}\right), \quad \mathcal{L}_{\text{cls}}^{i} = \sum_{b}\text{CE}_b(d^i), \\ \mathcal{L}_{\text{reg}}^{i} &= \mathbb{1}_1\left(d^i\right)\cdot \text{L1}\left(d^i, \Phi\left(\hat{d}_1^i\right)\right) + \mathbb{1}_2\left(d^i\right)\cdot \text{L1}\left(\tilde{d}^i, \Phi\left(\hat{d}_2^i\right)\right),\end{aligned} \tag{4}$$

where $\hat{d}_b^i$ represents the regression output for the b-th bin and i-th instance. The indicator function $\mathbb{1}_b\left(d^i\right)$ will only be activated, when d^i stays in b-th Bin. Training is only applied on 2D centers of bounding boxes. During inference the weighted average will be decoded as the final result:

$$\hat{d}^i = \mathcal{P}^i_{\text{Bin 1}}\left(\hat{d}^i\right) \cdot \Phi\left(\hat{d}^i_1\right) + \mathcal{P}^i_{\text{Bin 2}}\left(\hat{d}^i\right) \cdot \left(d_{\max} - \Phi\left(\hat{d}^i_2\right)\right), \tag{5}$$

where $P^i_{\text{Bin b}}$ denotes the normalized probability of $\hat{d}^i$.

2.4 Reference Area

Conventionally the regressed values of a single instance will be trained and accessed only on a single center point, which reduces the calculation. However, it also restricts the perception field of regression and affects reliability. To overcome these disadvantages, we apply the concept used by Eskil et al. [9] and Krishna et al. [10]. Instead of relying on a single point, a **R**eference **A**rea (**RA**) based on the 2D center point is defined within the 2D bounding box, whose width and height are set accordingly with a proportional value γ. All values within this area contribute to regression and classification. If RAs overlap, the area closest to the camera dominates, since only the closest instance is completely visible on the monocular image. During inference all predictions in the related RA will be weighted equally. Supplementary material contain additional details.

3 Experiments

3.1 Implementation Details

We performed experiments on the KITTI object detection benchmark [8], which contains 7481 training images and 7518 testing images. All instances are divided into easy, moderate and hard targets according to visibility in the image [8]. To numerically compare our results with other approaches we use intersection over union (IoU) based on 2D bounding boxes (AP), bounding boxes in Bird's-eye view (BEV AP) and in 3D space (3D AP). Most recently, the KITTI evaluation benchmark has been using 40 instead of 11 recalls. However, many methods only evaluated the average precision on 11 recalls (AP_{11}) in percentage. For fair comparison, we show here firstly AP_{11} on the validation set and then AP_{40} on the official test set.

Like most previous works, and in particular CenterNet, we firstly only consider the "Car" category and follow the standard training/validation split strategy in [3], which leads to 3712 images for training and 3769 images for validation. In particular, we keep the modified Deep Layer Aggregation (DLA)-34 [25] as the backbone. Regarding different approaches, we add specific head layers, which consist of one 3×3 convolutional layer with 256 channels, ReLu activation and a 1×1 convolution with desired output channels at the end. We trained the network from scratch in PyTorch [15] on 2 GPUs (1080Ti) with batch sizes 7 and

Table 1. $AP_{11}(\%)$ on KITTI validation set at 0.5 IoU threshold. We focus on the car detection result here. RT indicates runtime in *ms*. *ct3d* denotes CenterNet with 3D center points instead of 2D center points. *eigen* represents the original Eigen's transformation in CenterNet, while *lid* refers to the LID and *dj* represents the DepJoint approach. *ra* indicates a reference area supporting regression tasks. The best result is marked in bold, the second best is underlined. E, M and H indicate Easy, Moderate and Hard instances.

Approach	RT	2D AP			BEV / 3D AP		
	(ms)	E	M	H	E	M	H
CenterNet [26]	**43**	**97.1**	87.9	79.3	31.5 / 19.5	29.7 / 18.6	28.1 / 16.6
CenterNet(ct3d)	-	87.1	85.6	69.8	46.8 / 39.9	37.9 / 31.4	32.7 / 30.1
Mono3D [2]	-	92.3	**88.7**	79.0	30.5 / 25.2	22.4 / 18.2	19.2 / 15.5
MonoGRNet [16]	60	-	-	-	- / 50.5	- / 37.0	- / 30.8
Multi-Fusion [24]	120	-	-	-	55.0 / 47.9	36.7 / 29.5	31.3 / 26.4
M3D-RPN [1]	161	90.2	83.7	67.7	55.4 / 49.0	42.5 / 39.6	35.3 / 33.0
Center3D(+eigen)	47	96.7	88.0	79.4	47.6 / 38.0	37.6 / 30.8	32.4 / 29.4
Center3D(+lid)	53	96.9	87.5	79.0	51.3 / 44.0	39.3 / 35.0	33.9 / 30.6
Center3D(+dj)	54	96.1	86.8	78.2	55.4 / 49.7	41.7 / 38.1	35.6 / 32.9
Center3D(+dj+ra)	56	96.8	88.2	**79.6**	**56.7/ 52.5**	**43.5/ 40.5**	**41.2/ 34.9**

Table 2. $AP_{40}(\%)$ on KITTI test set at 0.7 IoU threshold. We show the car, pedestrain and cyclist detection results here. E, M and H indicate Easy, Moderate and Hard instances.

Approach	Car								
	2D AP			AOS			BEV/3D AP		
	E	M	H	E	M	H	E	M	H
M3D-RPN [1]	89.0	**85.1**	69.3	88.4	**82.8**	67.1	**21.0/14.8**	13.7/**9.7**	10.2/7.4
Center3D	**95.1**	**85.1**	**73.1**	**93.1**	82.5	**70.8**	18.9/12.0	**14.0**/9.3	**12.4/8.1**
BEV/3D AP	Pedestrain						Cyclist		
	E		M		H		E	M	H
M3D-RPN [1]	**5.7/4.9**		**4.1/3.5**		3.3/**2.9**		1.3/0.9	0.8/0.7	0.8/0.5
Center3D	**5.7/4.9**		3.7/3.4		**3.5**/2.8		**5.3/4.3**	**2.8/2.4**	**2.7/2.1**

9. We trained the network for 70 epochs with an initial learning rate of $1.25e^{-4}$ or $2.4e^{-4}$, which drops by a factor of 10 at 45 and 60 epochs if not specified otherwise.

3.2 Center3D

We can bridge the gap between 2D and 3D center points by adding 2 specific layers to regress the offset $\mathbf{\Delta c}^i$. For demonstration we perform an experiment, which is indicated as *CenterNet(ct3d)* in Table 1. It models the object as a projected 3D center point with 4 distances to boundaries. The visible object, whose 3D center point is out of the image, is ignored during training. As Table 1 shows, for easy targets *ct3d* increases the BEV AP by 48.6% and the 3D AP by 104.6% compared to the baseline of CenterNet. This is achieved by the proper

Table 3. Experimental results of LID. We show the comparison between SID and LID, the influence of different bins. *-res* indicates no regression of residuals as an ablation study. APs are given in percentage.

	Bin	BEV AP			3D AP		
		Easy	Mode	Hard	Easy	Mode	Hard
Eigen	–	47.6	37.6	32.4	38.0	30.8	29.4
SID	40	33.4	27.6	26.9	26.7	24.4	21.1
LID	40	31.5	25.6	24.9	24.7	22.7	19.4
SID	100	47.6	37.5	32.3	39.6	33.8	29.4
LID	100	50.2	39.2	**33.9**	41.5	**35.6**	**31.3**
SID	80	48.7	37.9	32.9	40.4	34.3	30.0
LID	80	**51.3**	**39.3**	**33.9**	**44.0**	35.0	30.6
LID/-res	80	37.1	33.0	29.2	31.9	26.7	25.8

decoding of 3D parameters based on an appropriate 3D center point. However, simply modeling an object with a 3D center will hurt 2D performance, since some 3D centers are not attainable, although the object is still partly visible in the image.

In contrast, the Center3D approach is able to balance the trade-off between a tightly fitting 2D bounding box and a proper 3D location. The regression of offsets is regularizing the spatial center, while also preserving the stable precision in 2D (the 7th row in Table 1). BEV AP for moderate targets improves from 29.7% to 37.6%, and 3D AP increases from 18.6% to 30.8%, which performs comparably to the state of the art. Since Center3D is also the basis for all further experiments, we treat the performance as our new baseline for comparison.

3.3 LID

We first implement and adjust the DORN [7] approach for depth estimation instance-wise rather than pixel-wise. Following DORN we add a shift ξ to both distance extremum d^*_{min} and d^*_{max} to ensure $d_{\text{min}} = d^*_{\text{min}} + \xi = 1.0$. In addition, we perform experiments for our LID approach to demonstrate its effectiveness. We set the number of bins to 80, and add a single head layer to regress the residuals of the discretization. Hence, for depth estimation, we add head layers to generate the output features $\hat{D} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 161}$, while CenterNet generates an output feature $\hat{D} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 1}$. Here the depth loss weight $\lambda_{\text{dep}} = 0.1$ yields the best performance. We compare the results with our new baseline Center3D with the same learning rate of $1.25e^{-4}$. The best result is shown as *Center3D(+lid)* in Table 1.

More detailed, Table 3 shows both LID and SID with different number of bins improved even instance-wise with additional layers for ordinal classification, when a proper number of bins is used. Our discretization strategy LID shows a considerably higher precision in 3D evaluation, comprehensively when

Table 4. Experimental results of DepJoint. The regression part of depth estimation is supported by RA. The dependence on α/β is shown, which represents the threshold scale of first/second bins regarding to $d_{\max}$. The left column shows the results of associated strategy, while the right column shows the results of overlapping strategy. APs are given in percentage.

	BEV AP			3D AP				BEV AP			3D AP		
α/β	E	M	H	E	M	H	α/β	E	M	H	E	M	H
0.7/0.3	52.9	40.6	35.0	44.5	**37.5**	**32.7**	0.2/0.2	48.6	37.5	33.3	42.1	32.9	30.9
0.6/0.4	49.9	39.6	34.1	44.7	35.5	30.8	0.3/0.3	**53.2**	**41.2**	**35.1**	**47.4**	36.4	32.3
0.8/0.2	47.9	38.7	33.6	40.3	31.9	30.9	0.4/0.4	51.5	40.3	34.1	45.5	36.0	31.2
0.9/0.1	48.4	37.8	32.7	41.4	31.2	29.4	0.5/0.5	42.9	35.9	31.9	37.7	29.3	28.3

80 and 100 bins are used. A visualization of inferences of both approaches from BEV is shown in Fig. 3. LID only preforms worse than SID in the 40 bin case, where the number of intervals is not enough for instance-wise depth estimation. Furthermore we verify the necessity of the regression of residuals by comparing the last two rows in Table 3. The performance of LID in 3D will deteriorate drastically if this refinement module is removed.

3.4 DepJoint and Reference Area

In this section, we evaluate the performance of DepJoint approach. Firstly, only the regression part of depth estimation is supported by RA, which is sensitive according to its size. We set $\gamma = 0.4$ as default for RA, which yields mostly the best result. The supporting experimental results can be found in the supplementary material. Additionally, we apply $d_{\min} = 0\,\text{m}$, $d_{\max} = 60\,\text{m}$ and $\lambda_{\text{dep}} = 0.1$ for all experiments. Table 4 shows the experimental results. As introduced in Sect. 2.3, we can divide the whole depth range into two overlapping or associated bins. For overlapping strategy, the overlapping area should be defined properly. When the overlapping area is too small, the overlapping strategy actually converts to the associated strategy. On the other hand, if the overlapping area is too wide, the two independent bins tend to capture the general feature instead of specific feature of objects in panoramic depth. In that case, the two bins would not focus on objects in foreground and background respectively anymore, since the input objects during training for both bins are almost the same. This can also explain why the threshold choices of 0.7/0.3 and 0.6/0.4 result in a better accuracy in 3D space comparing with 0.9/0.1 and 0.8/0.2. For the associated strategy, the thresholds α/β of 0.3/0.3 and 0.4/0.4 show the best performance for the following reason: usually more distant objects show less visible features in the image. Hence, we want to set both thresholds a little lower, to

assign more instances to the distant bins and thereby suppress the imbalance of visual clues between the two bins.

Furthermore, the ablation study (without RA) verifies the effectiveness of DepJoint in comparison with Eigen's transformation. For DepJoint approach, α/β are set to 0.7/0.3 and $\lambda_{\text{dep}} = 0.5$. As *Center3D(+eigen)* and *(+dj)* shown in Table 1, DepJoint approach has a considerably higher AP in both BEV and 3D space.

3.5 Comparison to the State of the Art

Table 1 shows the comparison with state-of-the-art monocular 3D detectors on the validation dataset. *Center3D(+lid)* and *(+dj)* follow the settings described above. However, *Center3D(+dj+ra)* is supported here by RA in the regression of 3D offset, dimension, rotation and 2D width/height comprehensively ($\gamma = 0.4$). The corresponding loss weightings are all set to 0.1, except $\lambda_{\text{dep}} = 0.5$ and $\lambda_{\text{rot}} = 1$. The learning rate is $2.4e^{-4}$.

As Table 1 shown, all Center3D models perform at comparable 3D performance with respect to the best approaches currently available. Both LID and DepJoint approach for depth estimation have a higher AP than simply applying the Eigen's transformation in 3D task. Especially, *Center3D(+dj+ra)* achieved state-of-the-art performance with BEV APs of 56.7% and 43.5% for easy and moderate targets, respectively. For hard objects in particular, it outperforms all other approaches with BEV AP of 41.2% and with 3D AP of 34.9%.

Table 2 shows the AP_{40} of *Center3D(+dj+ra)* on the KITTI test set. In comparison with M3D-RPN, Center3D outperforms in 2D AP and average orientation similarity (AOS) obviously with comparable BEV and 3D APs. Besides, Center3D performs particularly better for hard object detection and Cyclist detection. More qualitative results can be found in the supplementary material.

Center3D preserves the advantages of an anchor-free approach. It performs better than most other approaches in 2D AP, especially on easy objects. Most importantly, it infers on the monocular input image with the highest speed (around three times faster than M3D-RPN, which performs similarly to Center3D in 3D). Therefore, Center3D is able to fulfill the requirement of a real-time detection.

4 Conclusion

In this paper we introduced Center3D, a one-stage anchor-free monocular 3D object detector, which models and detects objects with center points of 2D bounding boxes. We recognize and highlight the importance of the difference between centers in 2D and 3D by regressing the offset directly, which transforms 2D centers to 3D centers. In order to improve depth estimation, we further explored the effectiveness of joint classification and regression when only monocular images are given. Both classification-dominated (LID) and regression-dominated (DepJoint) approaches enhance the AP in BEV and 3D space. Finally,

we employed the concept of RAs by regressing in predefined areas, to overcome the sparsity of the feature map in anchor-free approaches. Center3D performs comparably to state-of-the-art monocular 3D approaches with significantly improved runtime during inference. Center3D achieved a better trade-off between 3D precision and inference speed.

References

1. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9287–9296 (2019)
2. 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)
3. Chen, X., et al.: 3d object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems. pp. 424–432 (2015)
4. 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)
5. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6569–6578 (2019)
6. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. pp. 2366–2374 (2014)
7. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2002–2011 (2018)
8. 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)
9. Jörgensen, E., Zach, C., Kahl, F.: Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. arXiv preprint arXiv:1906.08070 (2019)
10. Krishnan, A., Larsson, J.: Vehicle detection and road scene segmentation using deep learning. Chalmers University of Technology (2016)
11. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018)
12. Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X.: Gs3d: An efficient 3d object detection framework for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1019–1028 (2019)
13. 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)
14. 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)
15. Paszke, A., et al.: Automatic differentiation in pytorch (2017)

16. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. Proceedings of the AAAI Conference on Artificial Intelligence. **33**, 8851–8858 (2019)
17. 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)
18. Rey, D., Subsol, G., Delingette, H., Ayache, N.: Automatic detection and segmentation of evolving processes in 3d medical images: Application to multiple sclerosis. Medical image analysis **6**(2), 163–179 (2002)
19. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018)
20. Shin, K., Kwon, Y.P., Tomizuka, M.: Roarnet: A robust 3d object detection based on region approximation refinement. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 2510–2515. IEEE (2019)
21. Surmann, H., Nüchter, A., Hertzberg, J.: An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. Robotics and Autonomous Systems **45**(3–4), 181–198 (2003)
22. Vasile, A.N., Marino, R.M.: Pose-independent automatic target detection and recognition using 3d laser radar imagery. Lincoln laboratory journal **15**(1), 61–78 (2005)
23. Wang, Z., Jia, K.: Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. arXiv preprint arXiv:1903.01864 (2019)
24. 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)
25. 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)
26. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. In: arXiv preprint arXiv:1904.07850 (2019)