2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Object-Aware Centroid Voting for Monocular 3D Object Detection

Wentao Bao, Qi Yu, and Yu Kong

***Abstract*— Monocular 3D object detection aims to detect objects in a 3D physical world from a single camera. However, recent approaches either rely on expensive LiDAR devices, or resort to dense pixel-wise depth estimation that causes prohibitive computational cost. In this paper, we propose an end-to-end trainable monocular 3D object detector without learning the dense depth. Specifically, the grid coordinates of a 2D box are first projected back to 3D space with the pinhole model as 3D centroids proposals. Then, a novel object-aware voting approach is introduced, which considers both the region-wise appearance attention and the geometric projection distribution, to vote the 3D centroid proposals for 3D object localization. With the late fusion and the predicted 3D orientation and dimension, the 3D bounding boxes of objects can be detected from a single RGB image. The method is straightforward yet significantly superior to other monocular-based methods. Extensive experimental results on the challenging KITTI benchmark validate the effectiveness of the proposed method.**

I. INTRODUCTION

Object detection has been achieving remarkable progress in recent years with the help of deep learning models [1], [2], [3]. Though 2D objects can be accurately detected in an image, detecting 3D objects from visual data is much more difficult while its applications are increasingly demanded in autonomous driving, robotic navigation, etc. 3D object detection aims to recover the measurements of objects in a 3D physical world, including the 3D locations, 3D dimensions, and 3D orientations. In this paper, we focus on 3D object detection from only a single (monocular) image for the autonomous driving scenario.

Recent methods with high 3D object detection performance such as [4], [5], [6], [7], [8] heavily rely on the expensive LiDAR devices to provide 3D depth information. Also, LiDAR point cloud data brings challenge to process extremely sparse and noisy 3D points [9]. As an alternative, a monocular camera is much cheaper and the dense pixels can be effectively processed and perceived with recent deep neural networks [10]. Though 3D depth information is lost during the imaging process, recent advances in monocular-based depth estimation [11], [12], [13] and 3D object detection [14], [15], [16] demonstrate the potential to detect the complete 3D objects from a single image.

However, existing monocular methods such as [14], [15], [17] still resort to standalone *dense pixel-wise* depth estimator to achieve leading performance. For these methods, the unshared features of all image pixels would result in prohibitive computational cost. Different from these methods, recent work such as GS3D [18], FQNet [19] and Shift R-CNN [20] directly detect 3D objects by incorporating the geometric constraints into existing 2D object detectors without learning the pixel-wise depth. However, these methods are not designed as single model with an end-to-end learning process, achieving relatively low performance.

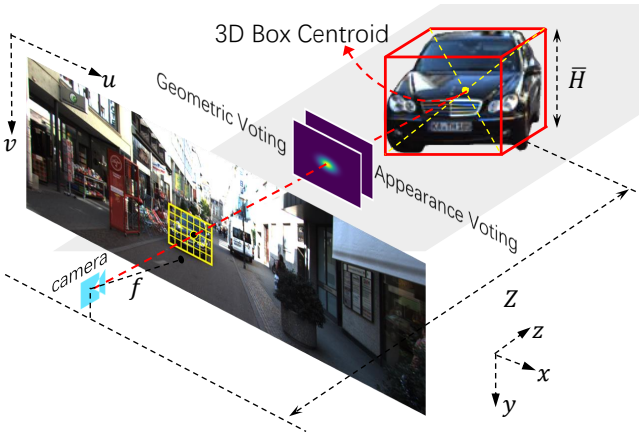


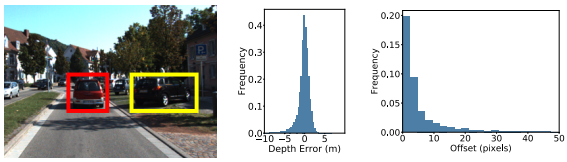
Fig. 1. **3D Object Detection Pipeline.** Given an image with predicted 2D region proposals (yellow box), the regions are divided into grids. Each grid point with (u, v) coordinate is projected back to 3D space by leveraging the pinhole model and the class-specific 3D height $\bar{H}$, resulting in 3D box centroid proposals. With the novel voting method inspired by both appearance and geometric cues, 3D object location is predicted.

In this paper, we show that it is feasible and effective to fulfill end-to-end 3D object detection without explicitly learning the dense 3D depth or using the handcraft post-processing. To this end, we re-visit the inherent constraints for autonomous driving scenario and obtain the following findings that are exploited in this paper.

First, the apparent heights of objects in an image are approximately invariant for the same class when the objects are with the same depth (as shown in Fig. 2(a)). Thus, by additionally leveraging the typical 3D height of object and the camera intrinsic matrix, 3D centroid proposals can be estimated with sufficient quality (Fig. 2(b)) from the grid coordinates of 2D region of interest (RoI) (see Sec. III-C). Second, the 3D object centroids are not exactly projected at corresponding 2D box center on image plane. We find the distribution of their geometric offset (Fig. 2(c)) is informative to vote for 3D object location. Third, region-wise appearance attention indicates the foreground objects within RoIs so that object awareness could be crucial to 3D object localization.

Based on these findings, we propose a 3D object detection method from a single image without densely predicting the depth of all image pixels. The general pipeline is depicted

Wentao Bao, Qi Yu, and Yu Kong are with the Golisano College of Computing and Information Sciences (GCCIS), Rochester Institute of Technology, Rochester, NY 14623, USA. {wb6219, qi.yu, yu.kong}@rit.edu



(a) (b) (c)

Fig. 2. **Examples and statistics on KITTI training set.** (a) The invariance of apparent height on image can be utilized to roughly infer the object depth. (b) The errors of 3D depth from the pinhole model are sufficiently small ($\mu_{\Delta Z} = -0.12$) with low variance ($\sigma_{\Delta Z} = 2.53$). (c) The offset distances on image are informative (small variance) to vote for the projection of 3D object centroids.

in Fig. 1. Given an image and the predicted 2D RoI, the grid coordinates of each RoI are projected back to 3D space as 3D centroid proposals through the pinhole model, where the apparent height h and typical 3D height $\bar{H}$ of objects are utilized. By dynamically learning the appearance attention map (AAM) and geometric projection distribution (GPD), an object-aware voting strategy is found effective to generate high-qualified 3D object centroid proposals, which eventually lead to accurate 3D object detection. The complete pipeline is presented in Fig. 3 and Sec. III.

Our method is straightforward and achieves better 3D localization performance than other methods like Pseudo LiDAR [17] and even the PointRCNN [7] on the KITTI dataset. The main contributions are summarized as follows.

- We propose an end-to-end learnable framework for monocular 3D object detection without dense pixel-wise depth estimation.
- A novel object-aware voting method is found effective by learning the knowledge from both 2D image appearance and 3D geometric projection.
- Our method exhibits superior 3D object detection and localization performance for small 3D objects.

II. RELATED WORK

Monocular 3D object detection receives much focus in recent years while a big performance gap exits when compared with LiDAR-based methods [8], [7]. To this end, deep learning-based depth estimation could greatly improve 3D object detection performance [14], [15], [17], [21]. Considering the high-cost of pixel-wise depth estimation for monocular 3D object detection, related work could be categorized into two types in terms of whether the depth information is densely learned or not.

With the help of monocular depth estimation, recent monocular 3D object detection methods have achieved leading performance [14], [22], [17], [23]. For these methods, 3D objects can be detected from the depth-based pseudo-LiDAR by using point cloud deep networks [9], [5], [7]. In addition to pseudo-LiDAR, birds' eye view (BEV) maps can also be estimated by the recent generative adversarial networks (GAN) for 3D object detection [21]. However, all these methods essentially leave the challenge of monocular 3D object detection to the pixel-wise depth estimation without providing a clean solution through an end-to-end single model. Furthermore, depth estimation for all image pixels by [11], [12] consumes much computational resource, which is essentially unnecessary for object detection tasks.

Instead, re-thinking the inherent constraints of 2D and 3D objects, the methodology design without dense pixel estimation is more promising. Existing literature such as [24], [25], [26] have demonstrated the effectiveness of utilizing monocular cues in autonomous driving scenarios. For 3D object detection, Chen et al. [27] proposed the Mono3D in which the 3D box proposals are generated by exhaustively placing 3D bounding boxes on the ground plane and exploits multiple priors to score the proposals. FQNet is recently proposed by Liu et al. [19] addressing the problem of fitting degree between the 3D projections and the objects. However, the performance is limited due to the ambiguity of the 3D projections caused by the unknown depth. Similar to our method, Li et al. [18] proposed GS3D method which also leverages the 2D region proposals for 3D object detection. GS3D utilizes the guidance from the orientation estimation to extract surface visual features of visible object parts. Naiden et al.proposed Shift R-CNN [20] to detect 3D objects based on [2]. In their method, 3D object location is optimized by the least square iteration with the constraints of 3D orientation, 3D dimension and 2D detection.

Similar to the methods without depth estimation, in this paper, we propose to exploit the geometric projection constraints between 2D and 3D to roughly infer the object's depth and introduce a novel voting method for 3D object localization. Experimental results on the KITTI benchmark [28] shows that our method can outperform existing state-of-the-arts at the time of submission.

III. APPROACH

A. Overview

The architecture of our proposed method is depicted in Fig. 3. It is designed based on the two-stage object detection framework proposed in [2], where the first stage generates region of interests (RoIs) through region proposal network (RPN) and the second stage detects the 2D bounding box from the RoIs and the shared convolutional features.

The complete pipeline of 2D object detection is maintained to guide the learning of 3D object detection. With the RPN module, 2D RoIs are first predicted. Then, each RoI is divided into grid cells and the grid coordinates (u_i, v_i) are projected back to the 3D space by the pinhole model, resulting in grid-formatted 3D centroids (X_i, Y_i, Z). By considering the object-awareness from both the appearance attention map (AAM) and the geometric projection distribution (GPD), 3D centroid proposals are voted and further fed into a fully-connected layer for 3D object localization. The dimension and orientation of 3D objects are predicted with the shared features in 2D object detection head. All of the 2D and 3D tasks are jointly trained in a multi-task loss function. Each module is introduced in detail in following sections.

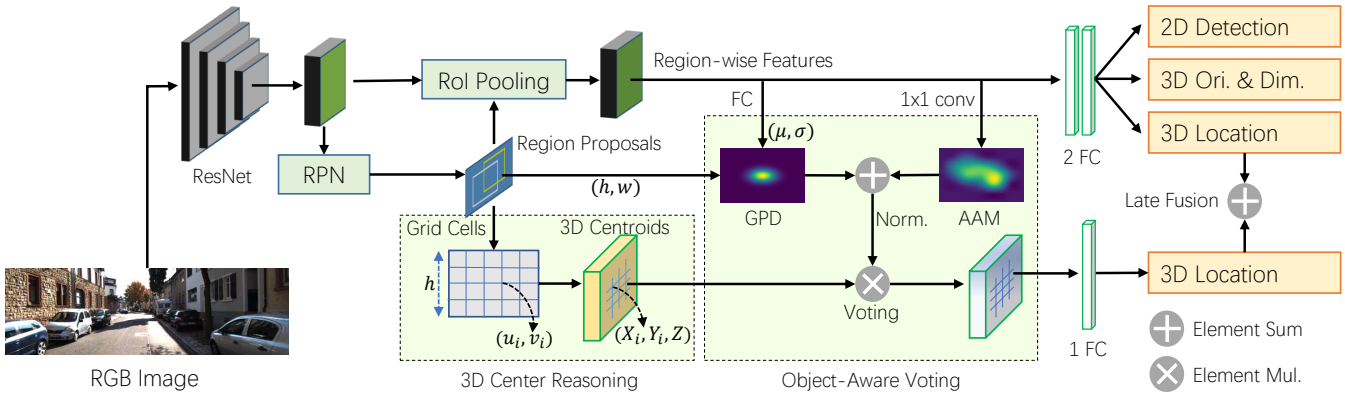


Fig. 3. **The Proposed Architecture**. 2D region proposals are first obtained from the RPN module introduced in [2]. Then, with the proposed *3D Center Reasoning* (the left dashed box), multiple 3D centroid proposals are estimated from the 2D RoI grid coordinates. Followed by the proposed *Object-Aware Voting* (the right dashed box), which consists of geometric projection distribution (GPD) and appearance attention map (AAM), the 3D centroid proposals are voted for 3D localization. For the 3D dimension and orientation, they are estimated together with 2D object detection head.

B. Region Proposal Generation

The two-stage object detection is to first generate a set of region proposals (candidate 2D bounding boxes) and then use the second stage to detect the objects [2]. Based on the learned convolutional feature maps of an input image, a small network predicts the region proposals at each location of the feature maps. This network consists of two fully-connected network branches for bounding box regression and objectiveness score prediction, respectively. Followed by the non-maximum-suppression (NMS), the region proposals are selected from the predictions.

In this paper, we maintain the complete 2D object detection pipeline so that the loss function of the 2D task is used:

$$L_{2d} = L_{cls}\left(t_{rpn}, t^*_{rpn}\right) + w_{2d} \cdot L_{reg}\left(t_{rcnn} - t^*_{rcnn}\right), \qquad (1)$$

where the t and t^* represent the predicted and the target boxes parameterized with the pre-defined *anchors*, a set of 2D boxes with pre-defined aspect ratios and scales. The coefficient w_{2d} is the hyper-parameter. The classification loss L_{cls} is the binary cross-entropy loss and the bounding box regression loss L_{reg} is fulfilled by a smooth L_1 loss:

$$L_{reg}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \qquad (2)$$

The definitions of loss functions L_{cls} and L_{reg} are used for all the tasks discussed in the following sections.

C. 3D Centroid Reasoning

The 3D object localization is the most challenging sub-task for monocular 3D object detection since the depth information is already lost during imaging. To this end, recent state-of-the-arts [14], [17] utilize the monocular depth estimation to predict the depth of the whole scene. However, since the image appearance of 3D objects and their local context are sufficient to infer the 3D information via deep neural networks, there is no need to predict the depth of all pixels in 3D object detection system.

In this paper, to infer the object depth, we re-visit the geometric constraints for the autonomous driving scenario based on the *pinhole model*. For the objects on driving road, they are horizontally placed without the pose angles of raw and pitch [27] with respect to the camera. Besides, the 3D dimension variance of each class of objects (such as *Car*) is quite small [14]. These constraints lead to our idea that the apparent heights of objects on image are approximately invariant when objects are in the same depth. Recent survey [13] also points out that the positions and apparent size of object in an image are applicable to infer the depth on KITTI dataset. Therefore, we believe that the 3D object centroid can be roughly inferred with the simple pinhole camera model. Therefore, the Z coordinate of the 3D object center can be approximately inferred by

$$Z \approx \frac{f \cdot \bar{H}}{h}, \qquad (3)$$

where the $\bar{H}$ is the average 3D height of objects for each class, and f is the constant focal length of camera.

In this way, we could anticipate that the depth Z may be greatly affected due to the possible tiny error of the predicted apparent height h since the numerator term $f \cdot \bar{H}$ is generally a large scalar value. To address this problem, instead of using only a single 2D coordinate, we utilize multiple grid coordinates of each RoI to infer 3D object centroid.

Specifically, we divide each 2D region proposals into $s \times s$ grid cells and project the grid coordinates back onto 3D space (see Fig. 3). Since each grid point indicates the *probable* projection of the corresponding 3D object centroid, we can get multiple 3D centroid proposals P_{3d} where the i-th centroid proposal $P_{3d}(X_i, Y_i, Z)$ is computed by

$$X_i = (u_i - p_x)Z/f, \quad Y_i = (v_i - p_y)Z/f, \qquad (4)$$

where (u_i, v_i) is the i-th grid point and (p_x, p_y) is the principal point given by the camera intrinsic matrix.

To verify the quality of the 3D centroid proposals, we compute the histogram of the depth error ΔZ between the estimated 3D centroids (X, Y, Z) and the corresponding ground truth (X^*, Y^*, Z^*) on KITTI dataset [28]. Results are shown in Fig. 2(b). The statistics ($\mu_{\Delta Z} = -0.12, \sigma_{\Delta Z}$=2.53)

(a) Background Grids (b) Occluded Grids (c) Inaccurate Grids

Fig. 4. **Motivations for object-aware voting.** Fig. 4(a): grid points locate on background point due to object deformation. Fig. 4(b): grid points locate on occluded objects. Fig. 4(c): grid points locate on background point due to the inaccurate 2D RoI.

indicates that the 3D centroid estimations are close to the ground truth, revealing the sufficient quality of the generated 3D centroid proposals.

Note that the straightforward pinhole model in this paper has the merit of long-range 3D localization as verified in Fig. 6(a), which is intractable for LiDAR-based 3D object detectors due to the sparsity of point cloud data. Though the occlusion and truncation could affect the accuracy of the 3D centroids, the proposed object-aware voting method could handle for this problem (see Sec. III-D).

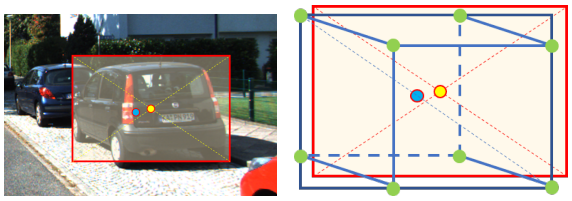
D. Object-aware Centroid Voting

Since the 3D object centroid has larger probability to be projected onto the center area of the RoI than other sub-regions, the estimated 3D centroid proposals from 2D grid coordinates should be applied with different confidence scores by considering the objectiveness. To this end, we propose the object-aware voting by considering two aspects, i.e., the appearance attention map (AAM) and the geometric projection distribution (GPD). The motivation and the methodology of them are introduced as follows.

Appearance Attention Map. This component impacts the voting confidence from three aspects. First, not all 2D coordinates within each RoI indicate the foreground object even when the 2D bounding boxes are accurately detected. Take the *Pedestrian* as an example (see Fig. 4(a)), the object deformation results in a relatively large 2D bounding box while only a few grid points locate on the foreground objects. Second, due to the object occlusion, the projected 3D centroid of one object could locate at another object region (see Fig. 4(b)). Third, the inaccurate 2D region proposals lead to meaningless background region (see Fig. 4(c)). To address these problems, we propose to introduce object-aware voting by leveraging the appearance attention.

Specifically, we use a single 1×1 convolution followed by sigmoid activation to generate appearance attention M_{app} from the feature maps of RoI pooling layer. The activated convolution feature map from the image indicates the foreground semantic objects due to the classification supervision in 2D objection detection, leading to the object-ware voting in our method.

Geometric Projection Distribution. This voting component comes from the distribution of the offset between the projected 3D centroid (denoted as $P_{3D\to 2D}$) and the 2D box center (denoted as P_{2D}). The offset is illustrated in Fig. 5. This offset results from the misalignment between the 2D bounding box annotation (red box in Fig. 5(b)) and the minimum rectangular bounding box of 3D box projections (blue rectangular box in Fig. 5(b)). To address the impact of this offset on the voting, we introduce to use the two-dimensional distribution of geometric projection $P_{3D\to 2D}$. Recently, He et al. [29] demonstrates that the 2D box center can be modeled as Gaussian distribution with ground truth as expectation. Similarly, for the offset $\Delta = P_{3D\to 2D} - P_{2D}$, it is formulated to follow Gaussian such that $\Delta \sim N\left(\mu, \sigma^2\right)$. where the parameters μ and σ^2 are two-dimensional and needs to be learned. Then, the geometric voting confidence map is given as



(a) Offset on Image (b) Geometric Description

Fig. 5. **Projection offset.** The projection of the 3D object centroid (blue point) may not exactly locate on the 2D bounding box center (yellow point) as shown in Fig. 5(a). Their offset is mainly caused by the uncertain 2D bounding box annotation (which cause misalignment between the blue and red box in Fig. 5(b)).

$$M_{geo} = \hat{N}(\hat{\mu}, \hat{\sigma}^2). \tag{5}$$

To dynamically learn the distribution, in this paper, the 2D grid coordinates and image features of RoI are concatenated together as input of a fully-connected layer to predict the offset Δ. And we exploit Kullback–Leibler (KL) divergence as loss function to supervise the learning:

$$L_{kld}\left(\hat{N}, N\right) = \frac{1}{2}\left[\log\hat{\sigma}^2 - \log\sigma^2 + \frac{\sigma^2 + (\mu - \hat{\mu})^2}{\hat{\sigma}^2} - 1\right], \tag{6}$$

where the prediction $\hat{\Delta}$ and ground truth Δ follow the Gaussians $\hat{N}(\hat{\mu}, \hat{\sigma}^2)$ and $N(\mu, \sigma^2)$, respectively. The minimization of KL divergence ensures that the distribution of predicted offsets approximates the distribution of offsets from ground truth data. Note that this is different from the method in [29], whose objective is to minimize the KL divergence between the Gaussian distribution of the predicted 2D box and Dirac distribution of the ground truth. Since the projection distribution is predicted from the image features, the proposed geometric projection voting is dynamic and should be adaptive for variant scenarios.

Eventually, the object-aware voting can be formulated as the element-wise multiplication with both the normalized probability maps M_{app} and M_{geo} as follows:

$$\tilde{P}_{3d} = P_{3d} \cdot G\left(M_{app} + M_{geo}\right), \tag{7}$$

where P_{3d} are grid-formatted 3D centroids proposals with shape $s \times s \times 3$, and the function $G(\cdot)$ is to normalize the

input with element-wise sum and sample the values at the center of each grid cell. In this equation, we use the element-wise summation as the voting probability map, indicating that either the appearance attention or the geometric projection distribution has an impact on the voting. This voting method is demonstrated critical to achieve good performance for 3D localization in our experiments (as shown in Table IV).

Followed by this step, an intuitive way is to average $\tilde{P}_{3d}$ to get the 3D location prediction. However, we believe that using the learnable fully-connected (FC) layer can adaptively regress the target. In practice, we only use 64 units for FC layer to regress the 3D locations.

Besides the geometric features from the FC layer, our method uses the shared appearance features from RoI pooling layer for 3D localization and introduces the late fusion (element-sum) to enhance the performance. This is similar to the late fusion method in [14], [15]. In the training stage, the 3D localization pipeline is trained with smooth L_1 loss:

$$L_{loc} = L_{reg}\left((P_{loc}^{(g)} + P_{loc}^{(a)}) - P_{loc}^{*}\right), \tag{8}$$

where the $P_{loc}^{(g)}$ and $P_{loc}^{(a)}$ are 3D location predictions from geometric and appearance features, respectively. P_{loc}^{*} represents the corresponding ground truth. The late fusion enforces $P_{loc}^{(a)}$ to be the residuals of P_{loc}^{*}, which takes the advantage of residual learning. This has been successfully verified in our experiments (see Fig. 6(b)).

E. 3D Dimension and Orientation Estimation

Similar to recent image-based 3D object detection methods [14], [30], we directly use the region-wise image features to predict the 3D dimension and orientation angle with the fully-connected layer.

For the 3D dimension prediction, the loss function comparing predictions and the ground truth are defined in the logarithm space through the smooth L_1 loss, which is the typical practice in existing literature [14], [12], such that

$$L_{size} = L_{reg}\left(\log\left(P_{size}\right) - \log\left(P_{size}^{*}\right)\right), \tag{9}$$

where the P_{size} and the P_{size}^{*} are the predicted 3D dimension and the corresponding ground truth.

For 3D orientation estimation, we use Multi-Bin [30] to disentangle it into residual angle prediction and angle bins classification. Specifically, the orientation angles are categorized into N overlapped bins, resulting in a N-dimensional classification sub-task. For each angle bin, the residual angles with respect to the bin center are regressed, leading to a N-dimensional regression sub-task. Therefore, the 3D orientation estimation loss is formed as

$$L_{angle} = L_{cls}\left(\sigma\left(P_{bin}\right), \sigma\left(P_{bin}^{*}\right)\right) + w_{ang} L_{reg}\left(P_{res} - P_{res}^{*}\right), \tag{10}$$

where P_{bin} and P_{res} are predictions of the bins classification and residuals regression. P_{bin}^{*} and P_{res}^{*} are corresponding ground truth. Function $\sigma(\cdot)$ represents the sigmoid function and w_{ang} is the constant coefficient.

F. Multi-task Training

In this paper, the loss functions for the tasks of 2D and 3D object detection are added together to form a multi-task learning objective:

$$L = aL_{2d} + bL_{loc} + cL_{size} + dL_{angle} + eL_{kld}, \tag{11}$$

where coefficients a, b, c, d and e are hyper-parameters and could be obtained by using validation set in training. The joint training ensures that 2D and 3D object detection can benefit from each other.

IV. Experiments

To validate the proposed method for the driving scenario, we conducted the experiments on the KITTI benchmark [28], which provides the most widely used 3D object detection dataset. It contains 7,481 RGB images with both 2D and 3D bounding box annotations and 7,518 unlabeled images for testing. Following *train/val* split proposed by [33], there are 3,712 and 3,769 training and validation samples, respectively. Though there are other driving datasets such as nuScenes [34], existing monocular 3D object detection methods are far from competitive than LiDAR-based methods so that evaluation only on KITTI is sufficient for monocular tasks due to large performance gap between *val* and *test* sets.

We use the KITTI official toolkit to evaluate both 3D object detection (3D AP) and 3D localization accuracy (BEV AP), both of which are based on the intersection-over-union (IoU) threshold. Results from three difficulty regimes are provided, i.e., *Easy*, *Moderate*, and *Hard*, with both 0.5 and 0.7 as the IoU thresholds. By default, the results are evaluated on the *Car* category.

A. Implementation Details

We implement the proposed method with MXNet [35] framework. The ResNet-101 [10] and Deformable RoI Pooling [36] are used as the network backbone and the RoI pooling layer, respectively. The grid size of RoI is set to 7. To handle the large variance of object appearance size, we use 2, 4, 8, 16, and 32 as anchor scales and 0.5, 1.0, and 2.0 as the aspect ratios during the RPN stage. We use data augmentation with random brightness during training.

In the training phase, the coefficients of Eqn. 1 and Eqn. 10 are set to 10, and the coefficients of the multi-task loss Eqn. 11 are set to 1, 5, 0.5, 5 and 1 by using the validation set, respectively. We use the pre-trained ResNet-101 model trained from COCO dataset [37] to initialize the weights of our proposed model. The online hard example mining (OHEM) method is used. We use the stochastic gradient descent (SGD) with initial learning rate 0.0001 and step-wise decay strategy with decay steps 30K and 45K iterations for total 20 epochs.

B. Comparison with State-of-the-arts

We compare the proposed method with the recent methods OFT-Net [31], FQNet [19], ROI-10D [23], GS3D [18], Shift R-CNN [20], and A3DODWTDA [32]. Similar to our method, they only use the single image without pixel-wise

TABLE I

KITTI *val* set results. The results are evaluated with both 3D object detection (3D AP) and 3D object localization (BEV AP). All of the compared methods do not learn the pixel-wise depth. (IoU threshold = 0.5 / 0.7)

Method	3D AP (%)			BEV AP (%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3D [27]	25.19 / 2.53	18.20 / 2.31	15.52 / 2.31	30.50 / 5.22	22.39 / 5.19	19.16 / 4.13
OFT-Net [31]	– / 4.07	– / 3.27	– / 3.29	– / 11.06	– / 8.79	– / 8.91
FQNet [19]	28.16 / 5.98	21.02 / 5.50	19.91 / 4.75	32.57 / 9.50	24.60 / 8.02	21.25 / 7.71
ROI-10D [23]	– / 9.61	– / 6.63	– / 6.29	– / 14.50	– / 9.91	– / 8.73
GS3D [18]	32.15 / 13.46	29.89 / 10.97	26.19 / 10.38	– / –	– / –	– / –
Shift R-CNN [20]	– / **13.84**	– / 11.29	– / **11.08**	– / 18.61	– / 14.71	– / 13.57
A3DODWTDA [32]	40.31 / 10.13	30.77 / 8.32	26.55 / 8.20	45.46 / 15.64	33.83 / 12.90	31.78 / 12.30
Ours	**44.68** / 13.65	**32.76** / **11.47**	**28.27** / 10.70	**51.23** / **20.65**	**38.33** / **16.35**	**34.30** / **14.21**

depth estimation. Recent monocular 3D object detection approaches like MF3D [14], MonoFENet [15], Pseudo LiDAR [17] all benefit from the pixel-wise depth estimation, thus they are not included for comparison. For both of 3D object detection and localization tasks, we report the results on *val* set in Table I and *test* set in Table II. In Table II, except for OFT-Net, results of other methods are obtained from KITTI official website.

3D Object Detection and Localization. As shown in Table I, the proposed method outperforms OFT-Net, FQNet, and A3DODWTDA by a large margin with most metrics ($2 \sim 5\%$ better than A3DODWTDA on *Moderate* regime). For the recent methods GS3D and Shift R-CNN, our method is on par with these methods with 3D AP (IoU threshold 0.7). However, for the 3D AP (IoU threshold 0.5) and all BEV AP metrics, our results are better than these methods about $1.5 \sim 2\%$. Moreover, from the KITTI *test* set results (see Table II), our method outperforms the best model Shift R-CNN with $1 \sim 4\%$, showing good generalization capability of the proposed model. Note that on *test* set, A3DODWTDA achieves better 3D AP in *Moderate* and *Hard* regimes. Since it predicts the 2D coordinates of eight-corners of a 3D box, occlusion and truncation are handled by fine-grained supervision from eight-corners, so that it is reasonable for A3DODWTDA to get better results on hard examples.

For small 3D objects category, we present the results for *Pedestrian* subset on KITTI *val* split as shown in Table III. We see that our method outperforms Shift R-CNN by $4 \sim 5\%$ and even better than MonoPSR $1 \sim 3\%$, which takes advantage of predicting object point cloud by shape reconstruction. In addition, the stable results on different difficulty regimes demonstrate the superiority of our method to handle occlusion and truncation, which leads to our better results on small deformable objects compared with MonoPSR and Shift R-CNN for 3D pedestrian detection.

Performance of 3D Object Centroid. Since our method is built upon the pinhole model, in which 3D object location could be affected by object distance (or apparent height), it is critical to figure out how 3D localization error changes with distance from object to camera. We analyze the mean centroid error (MCE) of detected 3D boxes with respect to object distance on KITTI *val* set as shown in **Fig. 6(a)**. Each

TABLE II

KITTI *test* set results. (IoU threhsold = 0.7).

Method	3D AP (%)			BEV AP (%)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
OFT-Net [31]	2.50	3.28	2.27	9.50	7.99	7.51
FQNet [19]	2.77	1.51	1.01	5.40	3.23	2.46
ROI-10D [23]	4.32	2.02	1.46	11.84	6.82	5.27
GS3D [18]	4.47	2.90	2.47	8.41	6.08	4.94
A3DODWTDA [32]	6.88	**5.27**	**4.45**	10.37	8.66	7.06
Shift R-CNN [20]	6.88	3.87	2.83	11.84	6.82	5.27
Ours	**8.13**	4.77	3.78	**16.24**	**10.13**	**8.28**

TABLE III

KITTI *Pedestrian val* set. (IoU threhsold = 0.5)

Method	3D AP (%)			BEV AP (%)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoPSR [16]	10.64	8.18	7.18	11.68	10.05	8.14
Shift R-CNN [20]	7.55	6.80	6.12	8.24	7.50	6.73
Ours	**11.55**	**10.93**	**10.04**	**13.10**	**12.33**	**11.70**

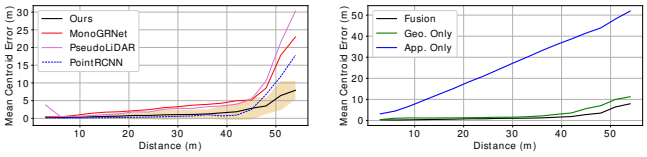
MCE value is computed by averaging the Euclidean distance from ground truth 3D box within a 3-meter interval to the nearest detected 3D box. The confidence region (shaded region) corresponds to one standard deviation around the mean value. In this experiment, our method is compared with two monocular-based methods MonoGRNet [22] and Pseudo LiDAR [17], and one LiDAR-based method PointRCNN [7]. The sharp decrease of PseudoLiDAR in the first interval is caused by outliers.

We can see that the mean errors of our method (black curve) are less than 2 meters when object distances are less than 40 meters. More importantly, as a monocular-based method, our method achieves competitive mean centroid error with the powerful LiDAR-based method PointRCNN (blue curve) within about 45 meters, and even outperforms all the others when objects locate more than 45 meters away. These results indicate that our method is superior to *look* faraway objects. Note that for faraway small objects in autonomous driving scenario, the capability of long-distance localization is even more important than accurate 3D bounding box detection.

TABLE IV

Ablation Studies. 3D / BEV results (IoU threshold = 0.5)

Fusion	AAM	GPD	Easy (%)	Mod. (%)	Hard (%)
✓	✓	✓	**44.68 / 51.23**	**32.76 / 38.33**	**28.27 / 34.30**
✓	✓		40.50 / 47.96	29.23 / 35.19	24.95 / 31.22
✓		✓	25.32 / 40.29	17.77 / 28.58	15.58 / 25.50
	✓	✓	41.20 / 49.49	30.22 / 37.03	25.95 / 32.82



(a) Methods Comparison (b) Late Fusion Validation

Fig. 6. **3D MCE vs. Object Distance.** X axis means the distance from 3D object centroids to camera. Smaller values indicate better results.

C. Ablation Study

We compare our full method with different variants to validate our method design. The experiments were conducted on KITTI *val* set and the results are shown in Table IV.

Effects of Different Voting Components. The first three rows in Table IV present the results of different voting components, i.e., the appearance attention map (AAM) and geometric projection distribution (GPD). We can see that the GPD voting contributes consistently to 3D object localization with different IoU thresholds and difficulty regimes, leading to more than 3% performance gain. In addition, the AAM voting serves as the most important component since removing it leads to significant performance degradation. These results are reasonable because the strong supervision such as gradients from 2D object detection head are totally ignored if removing AAM voting module. We can also compare the results of our proposed model without GPD with those of A3DODWTDA, showing that we can still achieve comparable 3D AP performance and even better than A3DODWTDA by 3% for BEV AP. Therefore, both the appearance-based voting AAM and geometric-based voting GPD are practically effective.

Effects of Late Fusion. The last row of Table IV shows the results when the late fusion is not used by removing the 3D localization branch from the *RoI pooling layer*. We can see that without the late fusion, the performance degrades $1.5 \sim 2\%$ in all metrics. This demonstrates that the region-wise features are informative to predict 3D object location. We can attribute the merit of the late fusion to the residual learning. That means the residual part of 3D location can be learned from image feature and is essentially the difference of ground truth and the relatively accurate 3D centroid.

To verify the residual learning by late fusion for 3D object localization, we use 3D mean centroid error (MCE) to evaluate the performances of 3D localization with only RoI appearance features (*App Only*), only geometric features from object-aware voting (*Geo Only*), and the late fusion from them (*Late Fusion*), respectively. The results are presented in **Fig. 6(b)**. It clearly shows that the MCE

TABLE V

3D Localization Methods. Evaluation results with BEV AP (IoU threshold = 0.5 / 0.7)

	Easy (%)	Mod. (%)	Hard (%)
Mean	50.12 / 17.42	38.02 / 13.29	33.34 / 11.05
FC Layer	**51.23 / 20.65**	**38.33 / 16.35**	**34.30 / 14.21**

performance gap between *Geo Only* and *Late Fusion* are significantly smaller than the gap between *App Only* and *Late Fusion*. Therefore, the late fusion enforces RoI image features to predict the residuals of 3D object locations.

Different 3D Localization Heads. Although an intuitive way to get the final 3D location results is to average the 3D centroid proposals, we believe that a small fully-connected layer (FC) can get better 3D location predictions. To validate this, we replace the FC layer of 3D object localization head with the average of 3D centroids (Mean). Results are reported in Table V. We can see that 3D object localization results by FC layer are significantly better than those by the average of 3D centroid proposals. This can be explained that the *mean* operation could be easily fitted by the nonlinear FC layer.

D. Qualitative Results

We visualize the 3D object detection and localization results on examples from KITTI *val* set (as shown in Fig. 7). It shows that our method can handle various driving scene such the the crowded highway (right of the first row). Besides, even far objects more than 40 meters can be accurately localized (see the second row). However, the orientation angle prediction is not quite accurate for faraway objects (see the last image). More vivid visualization results can be found in our video supplementary.

V. CONCLUSIONS

In this paper, we propose an end-to-end monocular 3D object detection method for autonomous driving scenario. Our method exploits the pinhole model for 3D centroids proposals generation. Followed by an object-aware voting which considers both the appearance attention map and the geometric projection distribution, the 3D centroid proposals are voted and used for 3D object localization. Our method gets rid of pixel-wise depth estimation in existing approaches while still keeps superior performance on KITTI benchmark. Furthermore, the proposed 3D centroid reasoning and voting modules can be easily integrated into cutting-edge two-stage object detectors or even instance segmentation models.

Acknowledgement. We thank NVIDIA for GPU donation. This research is supported by an ONR Award N00014-18-1-2875. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

REFERENCES

[1] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *CVPR*, 2016.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

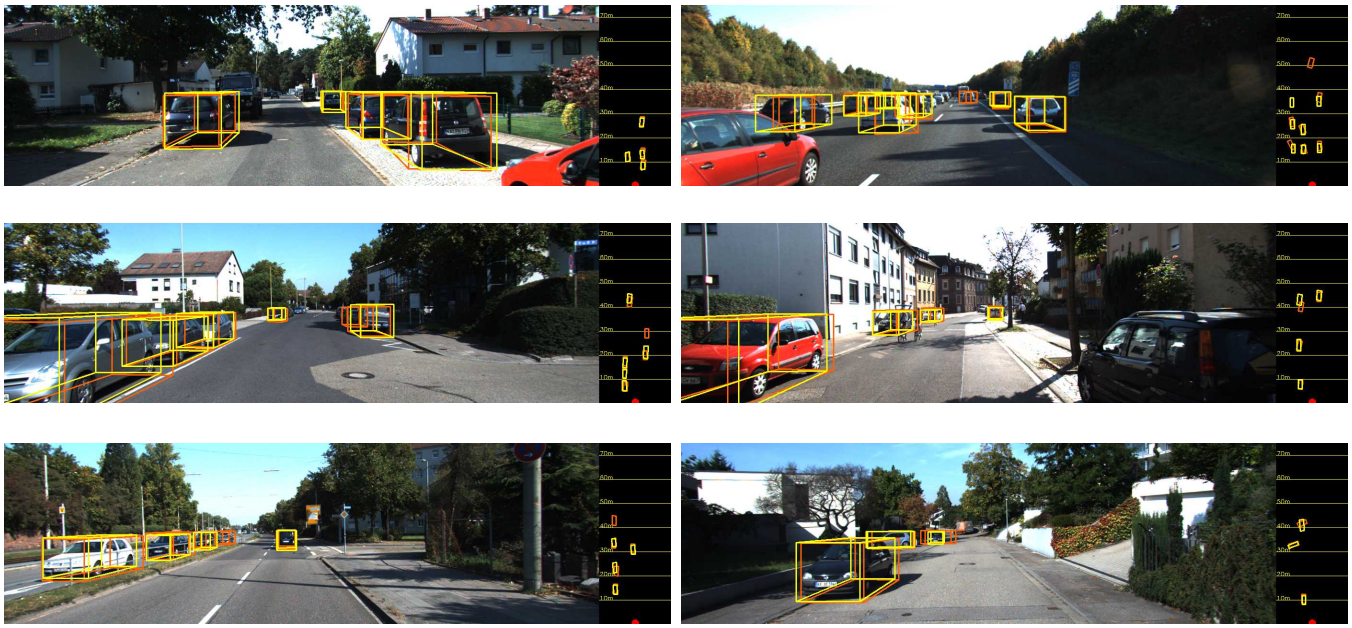


Fig. 7. **Qualitative results**. Red: detected 3D boxes. Yellow: ground truth. Right: birds' eye view (BEV) results. Zoom in for better visualization.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017.

[4] B. Li, "3D fully convolutional network for vehicle detection in point cloud," in *IROS*, 2017.

[5] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *CVPR*, 2018.

[6] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *IROS*, 2018, pp. 1–8.

[7] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *CVPR*, 2019.

[8] Z. Wang and K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," in *IROS*, 2019.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *CVPR*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[11] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018.

[13] T. v. Dijk and G. d. Croon, "How do neural networks see depth in single images?" in *ICCV*, 2019.

[14] B. Xu and Z. Chen, "Multi-Level fusion based 3D object detection from monocular images," in *CVPR*, 2018.

[15] W. Bao, B. Xu, and Z. Chen, "MonoFENet: Monocular 3D object detection with feature enhancement networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 2753–2765, 2020.

[16] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3D object detection leveraging accurate proposals and shape reconstruction," in *CVPR*, 2019.

[17] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *CVPR*, 2019.

[18] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3D object detection framework for autonomous driving," in *CVPR*, 2019.

[19] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3D object detection," in *CVPR*, 2019.

[20] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu, "Shift R-CNN: Deep monocular 3D object detection with closed-form geometric constraints," in *ICIP*, 2019.

[21] S. Srivastava, F. Jurie, and G. Sharma, "Learning 2D to 3D lifting for object detection in 3D for autonomous vehicles," in *IROS*, 2019.

[22] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *AAAI*, 2019.

[23] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *CVPR*, 2019.

[24] G. P. Stein, O. Mano, and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy," in *IVS*, 2003.

[25] S. Song and M. Chandraker, "Joint sfm and detection cues for monocular 3d localization in road scenes," in *CVPR*, 2015.

[26] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *ICRA*, 2018.

[27] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *CVPR*, 2016.

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.

[29] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *CVPR*, 2019.

[30] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *CVPR*, 2017.

[31] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," in *BMVC*, 2019.

[32] F. Gustafsson and E. Linder-Norén, "Automotive 3D object detection without target domain annotations," Master's thesis, Linköping University, 2018.

[33] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals for accurate object class detection," in *NeurIPS*, 2015.

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

[35] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems," in *Learning Systems*, 2015.

[36] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017.

[37] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.