Monocular 3D Object Detection with Decoupled Structured Polygon Estimation and Height-Guided Depth Estimation

Yingjie Cai\textsuperscript{1}, Buyu Li\textsuperscript{1}, Zeyu Jiao\textsuperscript{2}, Hongsheng Li\textsuperscript{1}, Xingyu Zeng\textsuperscript{2}, Xiaogang Wang\textsuperscript{1}

\textsuperscript{1}The Chinese University of Hong Kong \hspace{1em} \textsuperscript{2}Sensetime Group Limited

caiyingjie@link.cuhk.edu.hk, \{jiaozeyu, zengxingyu\}@sensetime.com, \{byli, hsli, xgwang\}@ee.cuhk.edu.hk

Abstract

Monocular 3D object detection task aims to predict the 3D bounding boxes of objects based on monocular RGB images. Since the location recovery in 3D space is quite difficult on account of absence of depth information, this paper proposes a novel unified framework which decomposes the detection problem into a structured polygon prediction task and a depth recovery task. Different from the widely studied 2D bounding boxes, the proposed novel structured polygon in the 2D image consists of several projected surfaces of the target object. Compared to the widely-used 3D bounding box proposals, it is shown to be a better representation for 3D detection. In order to inversely project the predicted 2D structured polygon to a cuboid in the 3D physical world, the following depth recovery task uses the object height prior to complete the inverse projection transformation with the given camera projection matrix. Moreover, a fine-grained 3D box refinement scheme is proposed to further rectify the 3D detection results. Experiments are conducted on the challenging KITTI benchmark, in which our method achieves state-of-the-art detection accuracy.

1 Introduction

3D object detection is an important computer vision task since it is an essential component of autonomous driving and robot perception to avoid collisions with surrounding objects. Most existing 3D object detection methods heavily rely on LiDAR devices to obtain accurate and direct depth measurements. However, such sensors can not widely adopted due to the expensive cost and limited perception range (\(\sim\)100m). The farther away the objects are, the fewer and sparser depth measurements would be on the objects. In contrast, cameras are much cheaper and can be installed on any vehicles. This paper mainly focuses on 3D detection with monocular images. In general, a 3D bounding box can be described by 7 parameters in autonomous driving scenarios, i.e. the location \((x, y, z)\), size \((l, w, h)\) and orientation \(\theta\) on the ground. For 3D detection from monocular images, recovering the location in 3D space is challenging on account of the absence of the accurate depth measurements. As illustrated in Fig. \[\text{(a)}\], given an accurate 2D bounding box of an object (Fig. \[\text{(a)}\]), its 3D location is still difficult to recover because one 2D box has an infinite number of corresponding 3D boxes (Fig. \[\text{(b)}\]) according to 2D-to-3D projection. However, the projection of the 3D bounding box on the 2D image plane is unique and much easier to estimate with features in a 2D image. Since the projected 3D box follows prior knowledge of being a polygon consisting of several quadrilaterals (corresponding to the visible surfaces of the 3D box), we refer it as \textit{structured polygon} for convenience. The 3D box can be completely recovered given the structured polygon, depth and the projection matrix. As the camera projection matrix is generally known in auto-driving scenarios, the only additional information required is the depth of the object.

Inspired by the analysis above, we propose a novel framework that decomposes the 3D object detection task into a structured polygon prediction task and a depth estimation task. Different from the commonly used 2D bounding boxes in most previous works (Chen et al. 2016), (Mousavian et al. 2017), (Xu and Chen 2018) and (Qi et al. 2018), the structured polygon can provide richer information for the 3D box recovery. Since the challenging depth estimation task is de-
Figure 2: The overall framework (Decoupled-3D) decouples the monocular 3D object detection problem into sub-tasks. The overall network consists of three parts. (Top row) The 2D structured polygons are generated with a stacked hourglass network. (Middle row) Object depth stage utilizes 3D object height as a prior to recover missing depth of the object. (Bottom row) 3D box refine stage rectifies coarse 3D boxes using bird’s eye view features in 3D-ROIs. Best viewed in color.

coupled, the specific module can be designed to accurately tackle it. Moreover, the prediction of other parameters, e.g., the orientation of the object and the aspect ratio, would not be affected by depth estimation. Therefore, our framework has significant superiority over existing works that estimate all the 3D factors simultaneously (Chen et al. 2018), (Chen et al. 2017), (Xu and Chen 2018), (Qi et al. 2018).

To obtain the structured polygon, we regress the projection points of the eight vertices of the cuboid via a stacked hourglass network. With the predicted structured polygon, we propose an efficient method to estimate the depth. Specifically, we use the object height as a prior to compute the inverse projection transformation from the 2D image plane to 3D space with the given camera projection matrix. With the estimated depth and the obtained structured polygon, we can consequently recover the complete 3D box.

The 3D box obtained from previous steps is actually a coarse estimation. We further propose a 3D box refinement scheme to rectify the coarse 3D boxes. We use the local features around and within the coarse 3D box for the refinement. The features are extracted from the bird’s eye view map of the estimated depth map by a monocular depth estimation algorithm DORN (Fu et al. 2018). In contrast to previous works that refine the 3D boxes with 2D image-level features such as MGR (Qin, Wang, and Lu 2019) and MLP (Xu and Chen 2018), we refine the deviations of coarse boxes with bird’s eye view map containing direct spatial information. The ingenious design can align the coarse boxes adaptively and substantially enhances the accuracy.

To validate the effectiveness of our method, we perform thorough experiments on the challenging 3D object detection benchmark KITTI (Geiger, Lenz, and Urtasun 2012) and achieve new state-of-the-art performance under both AP_{BEV} and AP_{3D} metrics. The contributions are summarized as following three-fold:

- A novel framework, which decomposes the challenging 3D detection problem into sub-tasks of image based structured polygon prediction and object depth estimation, is proposed. The two decomposed sub-tasks can be better tackled.
- An efficient object depth estimation approach is proposed, which uses the object height as a prior. Combined the depth with the structured polygon, coarse 3D boxes can be obtained.
- A fine-grained 3D box refinement scheme is proposed. Different from the existing methods, we rectify the coarse boxes with bird’s eye view map, which significantly improves the accuracy of the 3D boxes.

2 Related work

We briefly review recent works based on LiDAR data, stereo images and monocular images.

LiDAR-based 3D Object Detection. Most state-of-the-art 3D object detection methods reconstruct 3D bounding boxes using point clouds from LiDAR. (Luo, Yang, and Urtasun 2018) and (Zhou and Tuzel 2018) quantize the raw point cloud by using voxel grid and then feed the structured voxel grid to 2D or 3D CNN to detect 3D objects. (Qi et al. 2018) and (Shi, Wang, and Li 2019) respectively uses 2D bounding box and segmentation to lock effective point cloud to generate 3D bounding boxes instead of quantizing to voxel grid with less information lossing. They respectively uses 2D bounding box and segmentation to lock effective point cloud and both encode point cloud via PointNet++ (Qi et al. 2017). Our method focuses on monocular data setting and unavoidably suffers from the lack of accurate and direct depth measurements.

Stereo-based 3D Object Detection. There are several
works are based on stereo vision. Stereo R-CNN \cite{Li, Chen, and Shen 2019} utilizes stereo RPN to detect 3D objects on left and right images simultaneously and tries novel pixel-level refinement based on stereo matching to refine 3D boxes. 3DOP \cite{Chen et al. 2015} assumes enormous 3D candidates and exploits ground plane prior and object size to filter the candidates. Stereo images provide more information than monoculars. However, stereo setting has high requirements during the camera installation while our approach only needs a single image, which can be more flexible in real cases.

### Monocular-based 3D Object Detection

More and more recent works are based on monocular images even though it is the most difficult. MGR \cite{Qin, Wang, and Lu 2019} and Mono3D \cite{Chen et al. 2016} encode RGB image feature to 2D CNN to regress 3D proposals and further refine the proposals with superimposed 2D features. Mono3D generates a diverse set of 3D candidate boxes first and exploits ground plane prior and 2D cues including segmentation and object size to filter the candidates. Pseudo-LiDAR \cite{Wang et al. 2019} transforms the depth map into pseudo point clouds and feeds the points into LiDAR based methods. \cite{Kehl et al. 2017}, \cite{Pepik et al. 2015} and \cite{Chabot et al. 2017} adopt CAD models to build templates for better supervision meaning that more objects poses are prepared. Deep3Dbox \cite{Mousavian et al. 2017} leverages the geometry constrains between 3D and 2D bounding box to recover the 3D poses. These methods use more information from the superimposed 2D image level or additional CAD models to constrain. However, our method decomposes the problem in a novel approach and designs specific strategies for each subtasks to achieve better 3D object detection.

## 3 Our Approach

In this section, we present our proposed framework for 3D object detecting from monocular images. First, we introduce the overall formulation of our architecture. We then introduce 3D coarse box estimation with structured polygon and depth estimation. Finally, we demonstrate a 3D box refinement stage to rectify coarse boxes. We name our method 3D object detection via decoupled tasks as Decoupled-3D, as illustrated in Fig. 2.

### 3.1 Decoupled Tasks

As introduced, since the estimation of depth is the most strenuous part for monocular based 3D object detection, we decouple the depth from complicated 3D box estimation and decompose the task into structured polygon prediction and coarse-to-fine depth estimation sub-tasks.

In 3D object detection, each object can be covered by a minimal cuboid in the 3D space, denoted by $B_{3d}$. A cuboid contains eight vertices $P_i = [X_i, Y_i, Z_i]^T \in \mathbb{R}^3$, $i = 1, \cdots, 8$, as the left blue cuboid shown in Fig. 3. An object corresponds to a special structured polygon on the 2D image plane via 3D-to-2D projection. A structured polygon contains the eight vertices $\{p_i = [u_i, v_i]^T | i = 1, \cdots, 8\}$ as the 2D vertices shown in Fig. 3(right). Given the camera intrinsic matrix $K$, the projection of a 3D vertex $P_i$ on the image plane is formulated as the following equation:

$$K \cdot [X_i, Y_i, Z_i]^T = [u_i, v_i, 1]^T \cdot Z_i, \quad (1)$$

where $Z_i$ is the depth of the vertex $P_i$. Given $p_i$ and $Z_i$, $P_i$ can be obtained as:

$$[X_i, Y_i, Z_i]^T = K^{-1} \cdot [u_i, v_i, 1]^T \cdot Z_i. \quad (2)$$

According to this equation, to estimate $B_{3d}$, we just need $K$, projected 2D vertices $p_i$ and the corresponding depth $Z_i$. As the camera intrinsic matrix $K$ is generally known, the remaining problem is the estimation of the 2D vertices of the structured polygon and their corresponding depths.

### 3.2 2D Structured Polygon Estimation

In order to first obtain the locations of objects in the 2D image, we predict the 2D bounding box of each object by Faster RCNN \cite{Ren et al. 2015}.

For structured polygon estimation, we propose to regress the 2D image coordinates of 8 vertices based on the features extracted from the object area. However, it is still difficult to find accurate position in occlusion areas, texture-less regions and reflective surfaces. As shown in Fig. 3, the vertices are projected on texture-less background such as ground plane and wall without strong physical meaning. Solely applying the local feature is generally insufficient for accurate estimation in such challenging regions. Therefore, global context information should be incorporated to infer accurate positions of the vertices.

**Figure 3**: 3D bounding box (left) and structured polygon (right). Best viewed in color.
Yang, and Deng 2016). The fusion of features from multiscales can integrate both local and global features to obtain accurate vertices positions. Each of the eight projected vertices has a corresponding output heatmap from the network, as shown in Fig. 4. We use $M_i$ to denote the heatmap of the projected vertex $p_i$, and $M_i(u,v)$ is the value of the pixel $(u,v)$ on the heatmap, which represents the probability of the vertex locating at the this location. The supervision of each vertex is a label map $\hat{M}_i$ with the ground truth position being one and others being zeros. We use the Euclidean loss function for training, which can help the model converge faster:

$$L_{sp} = \sum_i ||M_i - \hat{M}_i||^2_2$$  \hspace{1cm} (3)

During testing, the vertex position is estimated as the location with the highest probability.

$$\hat{p}_i = \arg\max_{(u,v)} M_i(u,v)$$  \hspace{1cm} (4)

The label of eight projected vertices can be obtained via 3D-to-2D projection with Eq. (1) from 3D coordinates of vertices.

### 3.3 Height-Guided Depth Estimation

The depth of an object is the most challenging parameter to estimate due to the fact that this information is missing after 3D-to-2D projection. Therefore, instead of directly regressing the strenuous depth from image-level features, we choose to recover the depth via camera projection principle. Based on the projection principle, we adopt a simple, but effective strategy for the missing depth via structured polygon and a 3D physical prior, object height.

As shown in Fig. 5, $f$ is the camera focal length, $H$ represents the 3D height of object and $h_j$ for $j=1, 2, 3, 4$ is the projected height of one vertical edge of the cuboid. The height values of the four vertical edges in 3D space are the same, while the projected height of the four vertical edges are different due to their different depths in the 3D space. Fig. 5 clearly shows the 3D-to-2D projection process of one vertical edge (i.e., $h_2$). Therefore, the corresponding depth ($Z_j$) of each vertical edge of the cuboid can be expressed as

$$Z_j = f \cdot H/h_j \hspace{0.5cm} for \hspace{0.5cm} j = 1, 2, 3, 4$$  \hspace{1cm} (5)

where $h_j$ can be directly obtained from the estimated structured polygon, which is the pixel distance of two projected vertices. For object height $H$, an intuitive idea is to use the average value $A_H$ obtained from the statistics of the height values in data set. However, the average height is not accurate enough for each instance. So we estimate the height of each object. Specifically, we pool the RoI feature of an object on the feature map from Res4, and then use 2 fully connected layers to predict the height. Instead of regressing the ground truth height $G_H$ directly, our regression target $t_H$ is the scale change:

$$t_H = \log(G_H/A_H)$$  \hspace{1cm} (6)

The Smooth-L1 (Girshick 2015) loss function is adopted for the training of the regressor, since Smooth-L1 is less sensitive to outliers. Further, according to Eq. (4), the coordinates of eight vertices in 3D space can be obtained via generated projected vertices and depths.

With the eight 3D vertices, we use an average operation to obtain a coarse 3D box. Specifically, in KITTI dataset the location of an object is defined as the position of the bottom center of its 3D bounding box, so we use the average of the midpoints of the diagonal $P_2P_7$ and $P_3P_8$ to estimate the location $(x, y, z)$. The $l$ can be calculated using the average of distances of $P_2P_3$, $P_5P_7$, $P_1P_4$ and $P_2P_8$. $h$ and $w$ are calculated using the similar way. Orientation $\theta$ comes from the average of four vertors $P_3P_2$, $P_7P_6$, $P_1P_4$ and $P_5P_8$.

### 3.4 3D Box Refinement

The 3D box obtained from previous steps is actually a coarse estimation. But the error is usually minor and the ground-truth is just located nearby. As shown in the bird’s eye view of Fig. 6 (left), the predictions have deviations about 1m.

Based on this fact, the proposed method tailors a fine-grained refinement scheme for 3D detection. Different from existing methods exploiting image-level features or front-view depth map, we leverage bird’s eye view map, which contains direct spatial information, to rectify coarse boxes. The well-designed scheme can adjust coarse 3D boxes to better locations in nearby region. For convenience, we refer this Fine-Grained refinement strategy as FG.

Bird’s eye view maps are transformed from predicted monocular depth maps using DORN (Fu et al. 2018). The details for the transformation process are outlined in section 4.1. We take an entire BEV map and a set of coarse
boxes as input. The whole map is processed by a CNN to produce a feature map. Then, for each object we adopt an warping operation of the region on the feature map to extract a fixed-length feature vector. The region has 2x size of the coarse box to ensure that targets are within this region. Since this region contains direct 3D information, we named it 3D-ROI for convenience. Each feature vector is fed into a sequence of convolution layers followed by a fully-connected layer that outputs residual values \((\delta x, \delta y, \delta z, \delta l, \delta w, \delta h, \delta \theta)\) based on the coarse 3D box. A Smooth-L1 loss is used for the training of this network.

In this way, the deviations of coarse boxes are removed, which surges the performance. We argue that bird’s eye view map with direct 3D spatial information is much more suitable for fine-grained refinement in 3D task than 2D image-level features or front-view depth map. Detailed analysis and comparisons are in section 4.3.

4 Experiments

We evaluate the proposed method on the KITTI object detection benchmark (Geiger, Lenz, and Urtasun 2012) and split training images into training set and validation set following the commonly used train/val split mentioned in 3DOP (Chen et al. 2018). In the KITTI dataset, objects are divided into three difficulty regimes: easy, moderate and hard, according to the 2D box height, occlusion and truncation degrees following the KITTI official standard. For all experiments, we follow most previous methods to focus on vehicle category as it has the majority of samples in the dataset.

4.1 Implementation Details

ResNet-50 (He et al. 2015) is selected as our basic backbone to extract features, which is initialized by pre-trained weights on ImageNet (Russakovsky et al. 2015). The initial learning rate is 0.001 for the previous 30K interactions and then 0.0001 for another 10K iterations. For the 3D box refinement, we keep depth points in the following range of the bird’s eye view:

\[-25m \leq X \leq 25m, -1.5m \leq Y \leq 4.09m, 0 \leq Z \leq 50m\]

The predicted depth map is from a monocular based method DORN (Fu et al. 2018). The row and column represent left-right (i.e., \(X\)) direction and the depth (i.e., \(Z\)) direction respectively, and the top-down view, \(Y\) is mapped to a slice at each \((X, Y)\) location on bird’s eye view maps. The width and height of 3D-ROI are set as 256 and 456 respectively. The size is computed from the statistics to ensure targets are within this region. For the object height, we use 1.46m as the average value.

4.2 Comparison with Other Methods

We compare with state-of-the-art monocular based methods including Mono3D (Chen et al. 2016), Deep3Dbox (Mousavian et al. 2017), MLP (Xu and Chen 2018), ROI-10D (Manhardt, Kehl, and Gaidon 2019), GS3D (Li et al. 2019), MGR (Qin, Wang, and Lu 2019), MonoPSR (Ku, Pon, and Waslander 2019) and Pseudo-LiDAR (Wang et al. 2019).

Metrics. The proposed method is evaluated by Average Precision on both bird’s eye view \((AP_{BEV})\) and 3D detection \((AP_{3D})\) metrics. \(AP_{BEV}\) evaluates whether the location of prediction is accurate by calculating the intersection over union (IoU) with ground-truth boxes from bird’s eye view. This performance of this metric is critical for autonomous driving to avoid collision. \(AP_{3D}\) counts the intersection of two cuboids (i.e., predicted box and ground truth) and adds object height and up-down information based on location.

Bird’s Eye View Evaluation. \(AP_{BEV}\) evaluates the projection of the 3D box on bird’s eye view. For comprehensive comparison, we experiment with two IoU thresholds (0.5 and 0.7) following existing methods. Just as shown in Tab. 1 our method outperforms state-of-the-art monocular based methods at both IoUs and surpasses Pseudo-LiDAR (Wang et al. 2019) by 4.91\% for 0.5 IoU and 3.39\% for 0.7 IoU under moderate level respectively.

3D Detection Evaluation. For \(AP_{3D}\), we also perform evaluations under the two IoUs. Compared to \(AP_{BEV}\), \(AP_{3D}\) expands from bird’s eye view plane to 3D space and calculates the intersection over union with 3D ground-truth boxes. As shown in Tab. 1 our method outperforms state-of-the-art monocular methods in all difficulties for 0.5 IoU and surpasses Pseudo-LiDAR (Wang et al. 2019) by 8.20\% for moderate level. The results of 0.7 IoU have certain gap with Pseudo-LiDAR.

Results on Test Set. We submit our results to the KITTI test server for evaluation and compare with all monocular based published methods on the test set. As shown in Tab. 1 the results show that our method outperforms the previous methods by significant margins in terms of all metrics, which proves the effectiveness of our method. Compared to the latest state-of-the-art method, our \(AP_{BEV}\) has an average improvement of 3.31\%, and \(AP_{3D}\) increases by 1\% at easy level.

4.3 Ablation Study

In this section, we conduct ablation experiments to validate the effectiveness of different components of our overall framework. All comparison are engaged on validation set.

Benefits of Decoupled Tasks. As introduced, we decompose the 3D box estimation problem into sub-tasks. To better evaluate the contribution of decoupled tasks, we compare our coarse and final results with regressing all variables simultaneously. As shown in Tab. 1 the performance of jointly regressing all parameters is far worse than our decoupled strategy. Even compared with the coarse results, there is a drop of 10\% in terms of \(AP_{BEV}\) and \(AP_{3D}\).

Benefits of Structured Polygon. We add an experiment of utilizing bird’s eye view (BEV) features to regress coarse 3D boxes instead of using structured polygon (SP). As shown in Tab. 1 the model with structured polygon outperforms the one with BEV by 15.70\% and 22.62\% in terms of \(AP_{BEV}\) and \(AP_{3D}\) at moderate level, which demonstrates the contribution of structured polygon.
Table 1: Bird’s eye view localization and 3D detection performance: Average Precision in bird’s eye view (\(AP_{BEV}\)) and Average Precision of 3D boxes (\(AP_{3D}\)) on KITTI validation set.

| Method       | IoU=0.5 \(AP_{BEV}\) | IoU=0.7 \(AP_{BEV}\) | IoU=0.5 \(AP_{3D}\) | IoU=0.7 \(AP_{3D}\) |
|--------------|------------------------|------------------------|------------------------|------------------------|
|              | Easy       | Mod.       | Hard      | Easy       | Mod.       | Hard      | Easy       | Mod.       | Hard      |
| Mono3D       | 30.50      | 22.93      | 19.16     | 25.19      | 18.20      | 15.52     | 5.22       | 5.19       | 4.13      |
| Deep3Dbox    | 31.12      | 22.53      | 18.12     | 26.15      | 19.42      | 14.62     | 9.33       | 6.71       | 5.11      |
| MLF          | 55.02      | 36.73      | 31.27     | 47.88      | 29.48      | 26.42     | 22.03      | 13.63      | 11.60     |
| ROI-10D      | -          | -          | -         | -          | -          | -         | 14.76      | 9.55       | 7.57      |
| GS3D         | -          | -          | -         | 33.11      | 27.16      | 23.57     | -          | -          | -         |
| MGR          | 53.29      | 37.55      | 30.46     | 50.51      | 36.97      | 30.82     | 21.67      | 14.93      | 12.34     |
| MonoPSR      | 56.97      | 43.39      | 36.00     | 49.65      | 41.71      | 29.95     | 20.63      | 18.67      | 14.45     |
| Pseudo-LiDAR | 70.8       | 49.4       | 42.7      | 66.3       | 42.3       | 38.5      | 40.6       | 26.3       | 22.9      |
| Ours         | 73.22      | 54.31      | 45.97     | 69.40      | 50.50      | 42.46     | 44.42      | 29.69      | 24.60     |

Table 2: Average precision for bird’s eye view localization and 3D detection on KITTI test set.

| Method       | IoU=0.7 \(AP_{BEV}\) | IoU=0.7 \(AP_{3D}\) |
|--------------|------------------------|------------------------|
|              | Easy       | Mod.       | Hard      | Easy       | Mod.       | Hard      |
| ROI-10D      | 9.78       | 4.91       | 3.74      | 2.02       | 4.32       | 1.46      |
| GS3D         | 8.41       | 6.08       | 4.94      | 2.90       | 4.47       | 2.47      |
| MGR          | 18.19      | 11.17      | 8.73      | 9.61       | 5.74       | 4.25      |
| MonoPSR      | 18.33      | 12.58      | 9.91      | 10.76      | 7.25       | 5.85      |
| Ours         | 24.62      | 14.66      | 11.46     | 11.68      | 7.28       | 5.69      |

Benefits of Height-Guided Depth Estimation. As introduced in section 3.3, we propose a simple, but effective height-guided depth recovery strategy. To verify the effectiveness, we regress the depth directly and use mean depth error to compare the two strategies. As shown in Tab. 3, the impacts of 2D image-level features (i.e., \(+img\)) and front-view depth map (i.e., \(+fv\)) are basically the same. The results of 3D detection are all better than our coarse results and \(AP_{3D}\) improves about 1.3% at hard level. The performance of localization is basically unchanged. While with the help of fine-grained 3D box refinement (i.e., FG), \(AP_{BEV}\) and \(AP_{3D}\) have been substantially improved, which proves the effectiveness of FG successfully capturing bird’s eye view map.

Table 3: Ablation study of decoupled tasks. Jointly represents all variables regressed simultaneously.

| Method     | IoU=0.5 \(AP_{BEV}\) | IoU=0.5 \(AP_{3D}\) |
|------------|------------------------|------------------------|
|            | Easy       | Mod.       | Hard      | Easy       | Mod.       | Hard      |
| Jointly    | 16.44      | 12.05      | 9.90      | 6.84       | 4.50       | 4.13      |
| Coarse     | 26.42      | 20.91      | 17.93     | 19.67      | 16.36      | 13.87     |
| Ours       | 73.22      | 54.31      | 45.97     | 69.40      | 50.50      | 42.46     |

Table 4: Ablation study of structured polygon. BEV represents coarse 3D boxes regressed with bird’s eye view map.

| Method     | IoU=0.5 \(AP_{BEV}\) | IoU=0.5 \(AP_{3D}\) |
|------------|------------------------|------------------------|
|            | Easy       | Mod.       | Hard      | Easy       | Mod.       | Hard      |
| BEV        | 56.19      | 38.61      | 31.94     | 40.03      | 27.88      | 22.70     |
| SP         | 73.22      | 54.31      | 45.97     | 69.40      | 50.50      | 42.46     |

Benefits of 3D box refinement. As mentioned in section 3.4, we propose a tailored refinement scheme for 3D task and argue that bird’s eye view map with 3D spatial information is much more suitable than 2D image-level features and front-view depth map for fine-grained refinement in the 3D space. For comprehensive comparison, we refine coarse boxes with 2D image features and front-view depth map respectively. As shown in Tab. 4, the impacts of 2D image-level features (i.e., \(+img\)) and front-view depth map (i.e., \(+fv\)) are basically the same. The results of 3D detection are all better than our coarse results and \(AP_{3D}\) improves about 1.3% at hard level. The performance of localization is basically unchanged. While with the help of fine-grained 3D box refinement (i.e., FG), \(AP_{BEV}\) and \(AP_{3D}\) have been substantially improved, which proves the effectiveness of FG successfully capturing bird’s eye view map.

Table 5: Ablation study of height-guided depth estimation. The lower the mean depth error value, the better the results.

| Method           |
|------------------|
| mean depth error |
| regress directly |
| height-guided    |
| 2.41m            |
| 1.21m            |

Table 6: Ablation study of 3D box refinement. Comparison of different refinement strategies. (+\(img\)) and (+\(fv\)) represent refining the coarse results via image-level features and front-view depth map respectively.

| Method     | IoU=0.5 \(AP_{BEV}\) | IoU=0.5 \(AP_{3D}\) |
|------------|------------------------|------------------------|
|            | Easy       | Mod.       | Hard      | Easy       | Mod.       | Hard      |
| Coarse     | 26.42      | 20.91      | 17.93     | 19.67      | 16.36      | 13.87     |
| +\(img\)   | 26.36      | 20.74      | 17.83     | 21.82      | 16.95      | 15.17     |
| +\(fv\)    | 26.32      | 20.70      | 17.80     | 21.81      | 16.93      | 15.14     |
| +FG        | 73.22      | 54.31      | 45.97     | 69.40      | 50.50      | 42.46     |

We also conduct an additional experiment to verify the contribution of FG. As shown Tab. 6, when applied on other methods like Mono3D (Chen et al. 2016) and MGR (Qin, Wang, and Lu 2019), FG can also improve them with significant margins.

The remarkable margins for our coarse to final results is due to following two reasons. One is about the elaborately designed refinement scheme, which leverages bird’s
4.4 Qualitative Results

We show some quantitative results in Fig. 7 where structured polygons are visualized in top row and the 3D boxes are showed in bottom on bird’s eye view. The pink and green boxes represent ground-truth boxes and predicted results respectively. As seen from the visualization results, our method can accurately predict the boxes in different locations and orientations only based on monocular images.

5 Conclusion

In this paper, we propose an efficient monocular 3D object detection framework which decomposes the complicated 3D detection problem into a structured polygon prediction task and a following depth recovery task. The former task uses a stacked top-down/bottom-up hourglass network to build eye view map. Compared to other formats, our design with direct spatial information is more suitable for 3D detection.

Another is that most coarse boxes are actually not far from the corresponding targets. As shown in Fig. 8 most deviations of coarse X and Z are within 1 m. As long as coarse boxes are shifted suitably, the accuracy can be increased substantially.

6 Acknowledgment

This work is supported in part by SenseTime Group Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants CUHK14202217, CUHK14203118, CUHK14207319 and CUHK14208619.
