3D Bounding Boxes for Road Vehicles: A One-Stage, Localization Prioritized Approach Using Single Monocular Images

Ishan Gupta(✉), Akshay Rangesh, and Mohan Trivedi

University of California, San Diego, La Jolla 92093, USA
{i2gupta,arangesh,mtrivedi}@ucsd.edu

Abstract. Understanding 3D semantics of the surrounding objects is critically important and a challenging requirement from the safety perspective of autonomous driving. We present a localization prioritized approach for effectively localizing the position of the object in the 3D world and fit a complete 3D box around it. Our method requires a single image and performs both 2D and 3D detection in an end to end fashion. Estimating depth of an object from a monocular image is not as generalizable as pose and dimensions. Hence, we approach this problem by effectively localizing the projection of the center of bottom face of 3D bounding box (CBF) to the image. Later in our post processing stage, we use a look up table based approach to reproject the CBF in the 3D world. This stage is a single time setup and simple enough to be deployed in fixed map communities where we can store complete knowledge about the ground plane. The object's dimension and pose are predicted in multitask fashion using a shared set of features. Experiments show that our method is able to produce smooth tracks for surround objects and outperforms existing image based approaches in 3D localization.

Keywords: Single stage 3D object detection
Inverse perspective mapping · Effective near object localization

1 Introduction

Scene understanding is among the critical safety requirements to make an autonomous system learn and adapt based on his interactions with the surroundings. Works like [16] talk about the overall signal to semantics for surround analysis. [15] and [17] present complete vision based surround understanding systems. Taking inspiration from these works, our work proposes a complete vision based solution for estimating the location, dimension and pose of the surrounding objects. Complete 3D knowledge of the surround vehicles contributes to efficient path planning and tracking for autonomous systems. 3D object detection involves 9 degrees of freedom accumulated as pose, dimensions and location. In normal driving scenarios, we assume no roll and pitch of the objects and the visual yaw fluctuates around 0°, ±90° and 180°. Also, the dimensions of on road

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11133, pp. 626–641, 2019.
https://doi.org/10.1007/978-3-030-11021-5_39

objects like cars are highly invariant and have a high kurtosis. Effectively localizing the position of the object in 3D world become much more important for good 3D object detection.

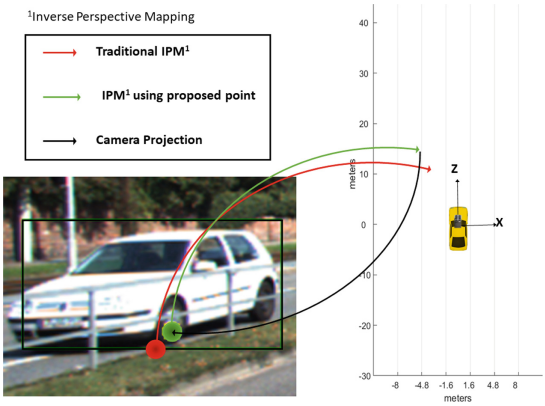

Fig. 1. Illustration of proposed approach: We train a detector to predict the keypoint (green circle) that would result in the desired 3D location after inverse perspective mapping (IPM). This is in contrast to traditional approaches where the bottom center of the 2D detection box (red circle) would be used to carry out the IPM. (Cropped image used from [3]) (Color figure online)

Most of the works in the domain of learning 3D semantics use expensive LiDAR systems to learn object proposals like [2] and [20]. In this work, we just use an input from a single camera and estimate the 3D location of the surround objects. We tackle the object localization by first estimating the projection of the center of the bottom face (CBF) on the image along with other parameters in an end to end fashion. Recent advances in the field of object detection can be broadly categorized into two stage and single stage architectures. The two stage architectures involve a pooling stage which takes input from the proposal network for all regions having the probability of an object. The detection architectures are further extended as in [5] to perfrom keypoint and instance mask prediction. On the other hand, architectures like [8,9,13] present a mechanism to learn the posterior distribution of each class given region in the image in a single stage. We take the inspiration from the success of these approaches and consider the 2D projection of the center of the bottom face as a keypoint. In driving scenarios, the position of this keypoint fluctuates a lot when the objects are in a certain range of the ego vehicle. Hence we focus on developing an efficient estimation scheme which prioritizes on localizing this keypoint against other learning tasks in the network.

All object detection architectures use anchors of different scales and ratios which are regressed over the whole feature map at different levels. The anchors are labeled as positive if they overlap above a threshold with the ground truth

location. Positive anchors are regressed to their corresponding ground truth match. The same regression approach can be applied for locating the projection of the 3D bounding box's center on the image plane which we refer as **CBF** in our work. However instead of creating a separate regression head for CBF, we change the anchor marking scheme to prioritize it's learning. This scheme reduces the total number of positive samples which might lead to heavy class imbalance. To avoid that, we use Focal loss [8] which helps in modulating the loss perfectly between the negative and positive examples. Our experiments show that change in anchor marking scheme does not effect the 2D detection task. Our modification implicitly helps in classifying those locations on the feature map which are close to the center projection. Hence, the network does all the task learning with reference to the keypoint's location which in our case is the projection of bottom face's center to the image plane.

Our main contributions presented in this paper can be summarized as follows - (1) We approach the 3D bounding box learning task in an end to end fashion and propose a complete image based solution. (2) We modify the single stage detection architecture to prioritize learning based on the keypoint location. (3) We demonstrate an alternative approach to traditional approaches which perform IPM (Inverse Perspective Mapping) on the center of the bottom edge of the 2D bounding box to find the corresponding location in the world coordinates. (4) We present a look up table based approach for reprojecting the center to the 3D world.

2 Related Research

We highlight some representative works in the 3D Object Detection in Autonomous Driving using different sensor modalities. Most approaches use depth sensors like LiDAR or a stereo setup. Chen *et al.* [2] learn proposals from the bird eye view of the LiDAR point cloud and use the corresponding region proposal in the image and the LiDAR front view to generate a pooled feature map from both LiDAR and camera modalities. The final 3D box regression and multi-class classification is performed after series of fusion operations. In [20], they distribute the complete LiDAR point cloud into voxels and perform learning upon the voxelized feature map. Each voxel's feature capture the local and global semantics for all the points inside that voxel. In [11], they run a 2D object detector over an image and seek for the LiDAR points corresponding to each object's frustum. Once, in the constrained LiDAR space, instance segmentation of 3D points is performed as done in [12]. All these techniques either learn proposals in the depth space or use it for post analysis. On the other hand, our approach just uses a single image and encourages a very cheap solution which can be deployed for near range scene perception. Our approach shows a happy marriage between Inverse Perspective Mapping(IPM) and deep network based predictions. Hence in a fixed map environment where there is complete knowledge of ground plane, our solution's performance becomes invariant to the range of the vehicle from the ego one.

Previous works which do 3D object detection using images, like [1] either rely on regressing 3D anchor boxes in the image using cues from complex features like segmentation maps, contextual pooling and location prior from the ground truth data. [10] learns dimensions and pose from cropped image features and uses projective constraints to compute the translation from the ego vehicle. They also analyzed how regressing the center of the 3D box against dimensions is sensitive to learning accurate 3D boxes. These approaches either compute complex features to regress the boxes in the 3D space or are not end to end learned. Our work shows a simple and efficient approach to compute the localization and a post processing stage to fit a 3D box over the object. We leverage upon works like [7] and present an end to end learning platform for 3D object detection.

3 Monocular 3D Localization

3.1 Problem Formulation

Given a single camera image, we have to estimate the location, dimensions and the pose of the all the objects in the field of view. The center of the bottom face of a 3D box lies on the ground plane. We use this constraint and design a supervised learning scheme which is able to localize the projection of the center on the image plane. Then we use the ground plane information by fitting a fixed number of planes on the ground surface and find the best plane which has the least inverse re-projection error. Note, this technique is only applicable for the points which lie on the ground plane. Hence, it is different from some other works which use the center as the intersection of the diagonals of the 3D box. We also extended our single stage architecture to predict the dimensions and the pose to fit a complete 3D box.

3.2 CBF Based Region Proposal

The original anchor based region proposal scheme takes as input a downscaled feature map and at each location on the feature map, we propose anchors of different scales and ratios. Assuming N anchors at each scale, only those anchors are marked as positive which have an intersection more than a threshold with any ground truth object. However we move slightly from this strategy. We project all the 3D center of the object to the image using camera projection matrices. The location of the projection is computed on each downscaled feature map which will be used for supervision. As the computed location will not be an integer, we mark all the nearest integer neighbors corresponding to that ground truth location in each feature map. Figure 2 shows the center of the positive anchors selected (red) and the location of the CBF projection (yellow). We perform regression on features maps which are downscaled by a factor of $1/2^i$, $\forall i = 3, 4, 5, 6, 7$ with respect to the original image size. Figure 3 shows how to determine the location of the positive anchors on any feature map. If both x and y coordinates of the center projection needs to be discretized, we choose the nearest 4 neighbors to it

on the feature map i.e $(x-1, y-1), (x+1, y+1), (x-1, y+1), (x+1, y-1)$. For cases, when either x or y coordinate is integer, we choose 6 neighbors by adding $((x, y+1), (x, y-1))$ or $((x-1, y), (x+1, y))$ in the two cases.

Fig. 2. The red circle shows the center of positive anchors selected by our approach and the yellow circle shows the projection of the center of the ground truth 3D bounding box. In comparison to IOU (Intersection Over Union) based anchor labeling approach, we label very few anchors as positive. Also depending upon the size of the anchor, IOU of the positive anchor with the object can be less than 0.5. (Color figure online)

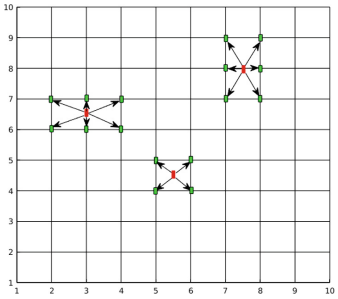

Fig. 3. The red dot shows the CBF projection in a feature map and the green dot shows the nearest integer neighbors. Depending on the data type of the ground truth, an object can not have more than six positive anchors. (Color figure online)

3.3 Regression Parameters

As described, our region proposal architecture marks only those anchors as positive which are around the CBF in the feature map. Simply classifying those anchors as positive will not suffice the purpose of accurate prediction of 3D translation. Hence, we attach a CBF regression head to the class body as shown in Fig. 4. The CBF head will help in accounting the problem caused by discretization of the CBF location in the feature map. We use the same approach as in [14] for regressing Δcbf_x and Δcbf_y. Apart from that we regress the Δx_c,

Δy_c, Δw, Δl for estimating the center and the dimensions of the 2D bounding box. As learning progresses, the classification head will learn to heat up only around the CBF location in the feature map. The shared pool of features learnt by the localization and the classification body can also be used to learn all the parameters for estimating an accurate 3D bounding box. Hence, we attach prediction heads for dimension and yaw in each prediction blob as shown in Fig. 4. For the classification head, we used the focal loss [8] which is excellent in handling the class imbalance between the positive and negative samples. Handling this imbalance is necessary because our location based anchor marking approach reduces the number of positive anchors per object. The regression targets for CBF and location head are learnt using Smooth-L1 loss, as in [4]. The regression loss is only computed for the positive anchors. Because of our new region proposal approach, we decrease the positive IOU threshold from 0.5, (as used in most of the cases) to 0.2. Anchors having a non zero IOU less than 0.2 are ignored while back propagation. Hence, the negative examples in our case will also include those anchors which are having a large overlap with the object of interest. The dimension head estimates the deviation from the mean dimensions of the dataset. This makes the learning easier because gradients will not be fluctuating heavily at the start of the training. The mean dimension (l,w,h) of cars in KITTI dataset is $(3.88, 1.63, 1.52)$ in meters. We use multibin loss to predict the camera yaw using 2 bins for classification, $(-\pi, 0)$ and $(0, \pi)$. Camera yaw can be defined as the angle made by the camera axis of the surround object with the light ray from ego camera. The overall loss function for all the predictions can be written as:-

$$L = L_{loc} + \alpha \cdot L_{class} + \beta \cdot L_{cbf} + \gamma \cdot L_{dim} + L_{\theta} \tag{1}$$

$$L_{\theta} = L_{\theta_{class}} + L_{\theta_{reg}} \tag{2}$$

We experiment with different weights for learning different tasks simultaneously. From our observations, using large weights during the start diverges the training. Hence, for the first 10 epochs, we use the same weight for all the tasks and eventually put α, β and γ to $8, 8$ and 2 respectively. All the loss functions are formulated as follows:-

$$L_{loc} = SmoothL1(t_x, t_{x^*}, t_y, t_{y^*}, t_w, t_{w^*}, t_h, t_{h^*}) \tag{3}$$

$$L_{CBF} = SmoothL1(t_{CBF}, t_{CBF^*}) \tag{4}$$

$$L_{dim} = 1/n \sum (d - d^*)^2 \tag{5}$$

$$L_{\theta_{class}} = SoftmaxLoss \tag{6}$$

$$L_{\theta_{reg}} = 1/n_{bins}((cos\theta - cos\theta^*)^2 + (sin\theta - sin\theta^*)^2) \tag{7}$$

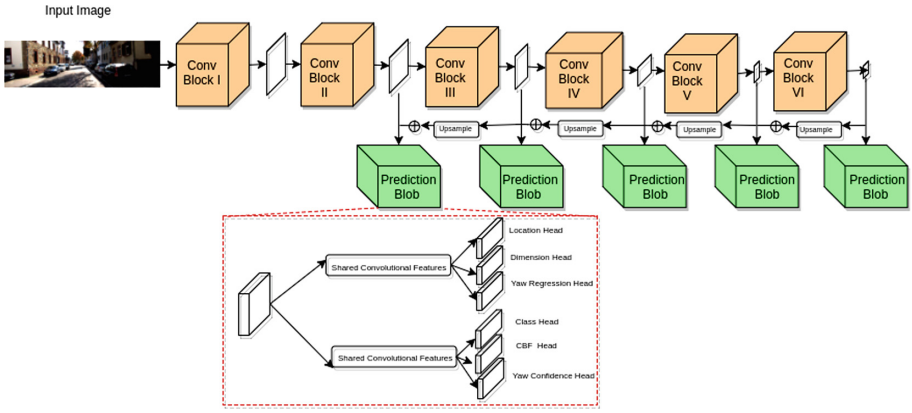

Fig. 4. Single stage multi-task learning framework for 3D bounding box estimation. Feature pyramid with resnet backbone is used to extract the features for all the prediction blobs. Each feature pyramid level predicts the location, dimension and pose of the object.

3.4 IPM Based Projection

The proposed network is capable to predict accurate location of the center projection on the image (CBF). Now we present a simple approach to map each CBF prediction to it's corresponding 3D location. The center of the 3D Box lies on the ground plane which allows approaches like Inverse Perspective Mapping to be applicable in our case. However instead of learning the transformation from ground plane to the image plane, we use a look-up table based approach which is easily extendable to more than one transformation. Multiple transformations will not restrict vehicles at different ranges to lie on a single ground plane. Also, the complete pipeline for reprojection of CBF is a one time setup. We use the ground LiDAR points for each scene in KITTI to kick start this one time setup. RANSAC is used to fit multiple planes to a given set of laser points. Upon a fixed 2D mesh grid, each plane equation will provide a different depth value. The 2D mesh grid includes points for which X ranges from 0 to 100 m and Y ranges from -40 to 40 m at a resolution of 0.01 m. Each 3D location is then projected to the image and stored in a separate KD-Tree for each plane. Also, we store the corresponding 3D location for each 2D location on the image. For each CBF prediction, we query all the KD-Trees to find the best possible solution. The 3D coordinates of the nearest neighbour are looked in the corresponding look up table and used as the center of the 3D box. The complete setup is summarized in the algorithm below:

Algorithm 1. IPM Setup Algorithm

procedure SETUPIPM($ground\_pts, tf\_img\_3d$) ▷ Returns possible ground planes
  $ground\_planes = RANSAC(ground\_pts)$
  $mesh\_2d \leftarrow get\_2d\_mesh(xmin, xmax, ymin, ymax, xres, yres)$
  $i \leftarrow 0$
  for all $plane \in ground\_planes$ **do**
    $pts\_3d[i] = get\_lidar\_mesh(plane, xmin, xmax, xres, ymin, ymax, yres)$
    $pts\_2d[i] = tf\_img\_3d.project(mesh\_3d[i])$
    $kd\_trees[i] = KDTREE(pts\_2d[i])$
    $i \leftarrow i + 1$
  end for
  return kd_trees, pts_3d, pts_2d
end procedure

3.5 Implementation

The complete architectural flow is shown in Fig. 4. We use the ResNet body [6] as our basenet and use feature pyramid as proposed in [7] to construct multi-scale feature maps. As shown in the architecture, each lower level of pyramid is formed by bi-linearly upsampling the upper level and adding the corresponding block's output from the basenet body. Each pyramid level is used to learn objects at different scales. Therefore, we chose anchor boxes of different sizes keeping number of aspect ratios to be constant at each level. We pull feature maps from five levels and use anchors boxes with sizes $(32 \times 32, 64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512)$ corresponding to each level. Anchor boxes are further changed to following aspect ratios $(1, 1/2, 2/1)$ at each level. The ResNet body is initialized with pretrained imagenet weights.

We use KITTI's 3D object detection dataset [3] for the training. The input resolution of the training data set is 1242×375, which is resized by changing the maximum dimension to 1024 keeping the aspect ratio constant. As different object scales are learnt efficiently using feature pyramid networks, we kept the input batch size as constant for entire training process. The KITTI training labels contain the translation for each labelled object which is transformed to the image using the LiDAR to camera and the rectified image projection matrices. We pad the image with zeros to take into account the cases where the CBF lies outside the image plane. We split the KITTI training data as proposed in [18] by ensuring that the same video sequence is not used in both training and validation set. The network is trained end to end with a batch size of 4 for 80 epochs. We use constant learning rate of 0.001 with a momentum of 0.9. Weight decay of 0.0001 is used to regularize the weights at each training step. During inference, the network will classify the regions surrounding the CBF projection as positive. We perform Non-Maximum Suppression (NMS) on the 2D bounding boxes by sorting the box predictions with the classification score. We use a NMS threshold of 0.3 and classification threshold of 0.5 during evaluation. The complete implementation can be summarized in an algorithm as follows.

Algorithm 2. Our Monocular 3D-BBOX Algorithm

```
procedure GET3DBBOX(img, kd_trees, meshes_3d)
    loc_preds, cls_preds, cbf_preds, dim_preds, yaw_preds ← net(image)
    bbox_2d, scores ← decode(loc_preds, cls_preds)        ▷ 2D Location of Object in Image
    for all pred ∈ cbf_preds do
        i ← 0
        min_dist ← ∞
        for all tree ∈ kd_trees do
            dist, loc ← tree.query(pred)
            if dist < min_dist then
                min_dist ← dist
                loc_3d[i] ← meshes_3d[loc]
            end if
        end for
        dim_l[i] ← mean_l + dim_preds[i][0]
        dim_w[i] ← mean_w + dim_preds[i][1]
        dim_h[i] ← mean_h + dim_preds[i][2]
        yaw[i] ← decode_multibin_pred(yaw_preds[i])
        i ← i + 1
    end for
    return loc_3d, dim_l, dim_w, dim_h, yaw
end procedure
```

4 Experimental Evaluation

We perform evaluation using the KITTI 3D object detection dataset. We are focusing our experiments only on the vehicle category in the KITTI. Figure 9 shows some qualitative results from our approach on KITTI cars in our test set.

4.1 Comparison with Direct CBF Regression

In this section, we compare our approach with the one where we keep the original IOU based region proposal methodology and add a regression head for CBF prediction. Our proposed positive anchor marking scheme gives better results than IOU based scheme. A variant of Chamfer Distance is used to evaluate and compare both the approaches. For each predicted CBF projection in the image, we find the closest ground truth correspondence to it. We also verify that the nearest neighbor should lie inside the region formed by expanding the predicted bounding box by factor of 1.5.

Figure 5 shows the improvement in pixel level estimation of the CBF with our proposed approach. Figure 6 illustrates some tracks picked from KITTI sequences. We can see how the flat ground plane assumption by IPM brings some jitters in the tracks. Next we also show that how our learning scheme is able to produce very similar tracks to the ones after applying IPM to ground trajectories. Figure 8 shows some visual examples where our proposed change helps in improving the CBF prediction.

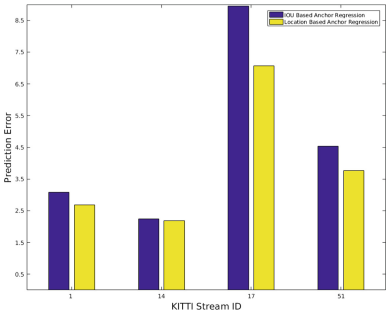

Fig. 5. We compare our change in the anchor labeling pipeline with IOU based anchor labeling. The blue bar shows the average prediction error for some KITTI streams used in the validation set. The yellow bar shows error for the case when the same architecture is trained with IOU based labeling. (Color figure online)

4.2 Effect of Range on Localization

In this section, we analyze how the 3D localization performance starts to degrade as the distance of the surround vehicle increases from the ego vehicle. We only analyze objects which are within a range of 50 m from the ego vehicle and show our performance at range interval of 10 m. Tables 1 and 2 show the 3D localization error after applying IPM over the predicted location of the center in the image and with/without applying IPM to the ground truth 3D location.

Table 1. 3D localization error variation with distance from ego vehicle after applying IPM to the ground truth annotations. We use only plane for our IPM based post processing. Multiple IPM planes can help in maintaining the same performance across all ranges.

Range (in meters)	C.D
[0–10)	0.312
[10–20)	0.668
[20–30)	1.103
[30–40)	1.582
[40–50)	2.212

4.3 Effect on the Detection Performance

The proposed change reduces the number of positive anchors in comparison to original anchor design. Also, the positive anchors are less overlapping with the objects because the CBF is most of the time near the bottom edge of 2D box.

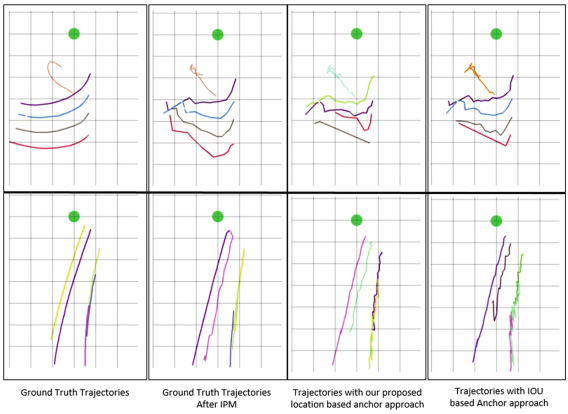

Fig. 6. We use the predicted center of the 3D box to form a complete trajectory for all the objects seen in the KITTI clip. Better object localization will remover the jitteriness from the tracks. Grid Resolution used is 2×2 m. The third column shows the trajectories formed using our approach. They are quite comparable to the ones in the second column which is formed after applying IPM on ground truth location and are much smoother than ones in the fourth column.

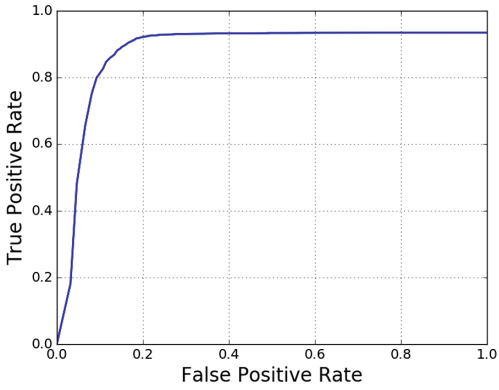

Fig. 7. ROC curve at IOU threshold of 0.5

Table 2. 3D localization error variation with distance from ego vehicle without applying IPM to the ground truth annotations. After comparison from Table 1, we can say that localization of the center on image plane is perfect and can be improved by using multiple IPM planes and better ground plane information.

Range (in meters)	C.D
[0–10)	0.454
[10–20)	1.446
[20–30)	2.358
[30–40)	4.532
[40–50)	7.823

Fig. 8. Illustration showing the improvements in pixel error (increase in concentric overlap) with the proposed approach. The red circles are the ground truth and yellow circles are the predictions. All circles have a radius of 5 pixels (Color figure online)

Table 3. Car detection results on the KITTI test set

Benchmark	Easy	Moderate	Hard
Car (detection)	79.87%	64.98%	49.31%

The results from the validation set on KITTI shows that our new design does not hamper the 2D localization. Figure 7 shows the ROC curve for the same.

As our main motivation was to analyze the quality of 3D bounding box, we ignored those samples which are heavily occluded and truncated from our training set. On the KITTI test dataset, we get reasonable recall at all distance ranges. Table 3 shows results obtained on KITTI test set for car detection.

Fig. 9. Illustration of the 2D detection boxes and the corresponding 3D projections estimated by our proposed approach.

Further improvements in the MAP can be obtained after performing padding on the image and including all truncated cases in the training.

4.4 3D Bounding Box Evaluation

To evaluate the accuracy of the predicted 3D bounding box, we compute the 3D Intersection over Union (IOU) and do a comparative analysis over surround objects from the ego vehicle. For objects which are in the range of [0–10] m, a good fitted 3D bounding box provides good scene understanding for near range perception activities. We compare our approach against [10] which also present a complete image based solution for 3D box estimation. In [10], first a 2D detector is ran over the image to obtain all the detections, whereas in contrast to that our approach learns the complete task of detection, 3D localization, orientation and dimension estimation in single step. Hence our evaluation is not variant to the performance of any component in our pipeline. Also, we evaluate the Average Orientation Similarity for KITTI Cars as shown in Table 4. The AOS score computes the cosine difference of the predicted yaw with the ground truth yaw and averages this over recall steps. We emulate KITTI's 3D bounding box

overlap strategy to compute the 3D IOU in our analysis. 3D recall at different ranges depends on the training samples which we include during training our architecture. On the other hand [10] are computing the mean 3D IOU after obtaining the cropped region from the 2D detector. Hence, even currently at lower recall from other approaches we are still able to outperform or match the 3D IOU across all distance ranges, as shown in Table 5. The recall of our approach for different distance ranges are shown in Table 6.

Table 4. Car orientation results on the KITTI test set

Benchmark	Easy	Moderate	Hard
Car (orientation)	50.26%	41.10%	32.03%

Table 5. 3D IOU variation with distance from ego vehicle

Method	[0–10)	[10–20)	[20–30)	[30–40)	[40–50)
SubCNN [19]	0.210	0.175	0.125	0.075	0.020
3D Bbox [10]	0.275	0.315	**0.200**	**0.152**	0.100
Our method	**0.487**	**0.324**	0.1958	0.143	**0.121**

Table 6. Recall for KITTI cars across distance ranges from ego vehicle

Range (in meters)	C.D
[0–10)	0.465
[10–20)	0.711
[20–30)	0.464
[30–40)	0.324
[40–50)	0.219

The large gain in 3D IOU for surround vehicles in the range of [0–10) should be credited to our localization prioritized approach. In Table 7 we compare the same localization error mentioned in Table 2 with the state of the art works selected for 3D IOU comparison. The single ground plane assumption suppresses our approach as the distance of surround vehicle increases from the ego.

Table 7. Localization error variation with distance from ego vehicle

Method	[0–10)	[10–20)	[20–30)
SubCNN [19]	1.449	1.887	2.437
3D Bbox [10]	1.447	**1.112**	**1.959**
Our method	**0.454**	1.446	2.358

5 Conclusions

In this paper, we propose a complete camera based solution to localize the surrounding objects in the 3D world. Our method helps in better estimation of the projection of the center in comparison to direct regression. For fixed map environments, the assumption of flat ground in IPM projection is resolved by learning a data dependent approach and choosing the best K fitting planes for all the points on the ground plane. This is a one time setup and the number of planes can be tuned without changing the inference pipeline. This learned module can be extended in future for learning the object maneuver and track prediction.

Acknowledgement. We would like to thank Nachiket Deo, Pei Wang and the anonymous reviewers for their useful inputs. We also gratefully acknowledge the continued support of our industry sponsors.

References

1. 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)
2. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3D object detection network for autonomous driving. In: IEEE CVPR, vol. 1, p. 3 (2017)
3. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
4. Girshick, R.: Fast R-CNN. arXiv preprint arXiv:1504.08083 (2015)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988. IEEE (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection (2018)
8. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. arXiv preprint arXiv:1708.02002 (2017)
9. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2

10. Mousavian, A., Anguelov, D., Flynn, J., Košecká, J.: 3D bounding box estimation using deep learning and geometry. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5632–5640. IEEE (2017)
11. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3D object detection from RGB-D data. arXiv preprint arXiv:1711.08488 (2017)
12. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5105–5114 (2017)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
14. 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)
15. Satzoda, R.K., Lee, S., Lu, F., Trivedi, M.M.: Vision-based front and rear surround understanding using embedded processors. IEEE Trans. Intell. Veh. **1**(4), 335–345 (2016)
16. Sivaraman, S., Trivedi, M.M.: Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Trans. Intell. Transp. Syst. **14**(4), 1773–1795 (2013)
17. Sivaraman, S., Trivedi, M.M.: Dynamic probabilistic drivability maps for lane change and merge driver assistance. IEEE Trans. Intell. Transp. Syst. **15**(5), 2063–2073 (2014)
18. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Data-driven 3D voxel patterns for object category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1903–1911 (2015)
19. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Subcategory-aware convolutional neural networks for object proposals and detection. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 924–933. IEEE (2017)
20. Zhou, Y., Tuzel, O.: VoxelNet: end-to-end learning for point cloud based 3D object detection. arXiv preprint arXiv:1711.06396 (2017)