Multi-Monocular Camera Object Detection

R10922045 Ching-Yu Tseng,
R10922066 Yi-Rong Chen,
R10922192 Ya-Ching Hsu

National Taiwan University

Abstract. Perceiving 3D scenes is a critical problem for autonomous driving, as it provides information from other vehicles such as their accelerations, velocities, and locations. The sensors used for perception include lidar, radar, and camera. In general, lidar-based 3D object detection methods perform better than camera-based methods. However, since the lidar sensor is quite expensive, we are searching for a solution for a pure camera solution. To this end, we propose a new end-to-end architecture that combines multi-monocular perspective view images for 3D object detection. This model uses a convolution neural network to generate depth distribution and thus provides explicit geometry information. Then, we project the camera feature into bird-eye-view to aggregate global information. Moreover, we warp the bird-eye-view feature from different timestamps to the current frame and use 3D convolution to extract the temporal information between each timestamp. Eventually, we build up a baseline 3D object detection model. In the experiments, our model achieve a competitive result and provided a new perspective for the 3D object detection task. In addition, we show the effectiveness of temporal information for occlusion. Our model can be a potential baseline for further research.

Keywords: Intelligent Vehicles, Bird-Eye-View, Multi-Monocular Camera, 3D Object Detection

1 Introduction

In recent years, autonomous vehicles have received considerable attention due to the prevalence of Artificial Intelligence. Autonomous driving can be divided into 3 critical tasks including perception, prediction, and planning. It is important to perceive the surroundings precisely for autonomous vehicles when generating controls for prediction and planning. In practice, the perception of autonomous vehicles mainly depends on 3D object detection.

3D object detection refers to the localization and the classification of specific objects in the 3D world coordinate systems, which may generate many cuboids to indicate the position of objects. Most previous works use lidar data to generate 3D bounding box representations in world coordinate systems. Lidar data

represent clusters of points in the 3D world coordinates. Those methods generally partition the 3D world coordinate into multiple voxels or pillars and extract features of points in each voxel to generate object proposals on world coordinate systems. The lidar-based methods [1, 2] empirically provide better performance. Apart from previous methods, other works focus on monocular camera solutions in the camera coordinate systems, which can be divided into 2 kinds of methods. The first camera-based method [3–6] is to use a feature extractor to extract directly on perspective view image and generate object proposals on the camera coordinate system. The other way [4, 6, 7] is to project perspective features into world coordinates with intrinsic and extrinsic matrices and perform object detection on the world coordinate systems as the lidar-based method.

Nevertheless, the cost of lidar data is extremely high. Thus, it is a trend to use camera data as an alternative to lidar data. In general, to achieve the perception of the surrounding environment with camera data, researchers conduct detection on each perspective view sequentially and aggregate all predictions at the end of inference time. It is called single-camera 3D object detection. For the purpose of using camera data as an alternative to lidar data, we have to provide the detection of surrounding objects with a global view. That is, combining camera data from different perspectives views may be an indispensable solution. However, few studies have provided this novelty. From this viewpoint, the motivation of the paper is to construct a model that conducts detection with multi-camera fusion during training time.

The aim of this work is to propose a camera-based method [3–5, 7–10] with competitive performance of lidar-based method. Common techniques of lidar-based method [1,2] utilize point cloud as input and operate detector on bird-eye view. To bridge the gap between the lidar model and camera data input, the pseudo-lidar method [4, 6–9] is utilized.

This paper provides the following contributions: (a) Transform camera data to pseudo-lidar format to fit lidar-based detector and conduct early fusion to generate more precise feature representation in bird-eye-view. (b) Use depth-guided method to improve the accuracy of pseudo-lidar. (c) Propose cross-view attention during early fusion to correct noise data. (d) Extract temporal features and aggregate during early fusion to optimize data performance. The remainder of this paper is divided into five sections including related works, methodology, results and evaluation, discussion, and conclusion. Numerical results show that the proposed method is able to achieve performance very close to the optimum.

2 Related Work

2.1 Multi-camera to Bird-Eye-View segmentation

Many works have dealt with the 2D perspective image to Bird-Eye-View transformation. In [8], propose a directly end-to-end architecture to extract Bird-

Eye-View representation from an arbitrary number of cameras. Three steps of the idea are used in the model: 1) Lift: transform each input perspective-view image into 3D frustum-shaped point cloud. 2) Splat: conduct sum pooling vertically to generate Bird-Eye-View feature maps. 3) Shoot: choose predefined plans according to Bird-Eye-View segmentation.

For the motion prediction task, [9] introduce a model that combines perception, sensor fusion, and prediction using a monocular camera. Provide a probabilistic future distribution to generate Bird-Eye-View segmentation.

2.2 Monocular 3D objection

Although a camera seems to be a better option than Lidar, a single camera [3, 5, 10] can not provide depth information of scenes directly. To eliminate the downside of a camera, existing works have been explored to predict depth information. In [4], it proposed the Categorical Depth Distribution Network(CaDDN) to predict appropriate depth from provided perspective-view image. Furthermore, [7] uses camera data to conduct 3D object detection by Lidar-Based algorithm [1, 2]. Combining monocular depth estimation and geometry transformation to generate pseudo-Lidar, and train a Lidar-based 3D detection network to eliminate noise in pseudo-Lidar.

3 Methodology

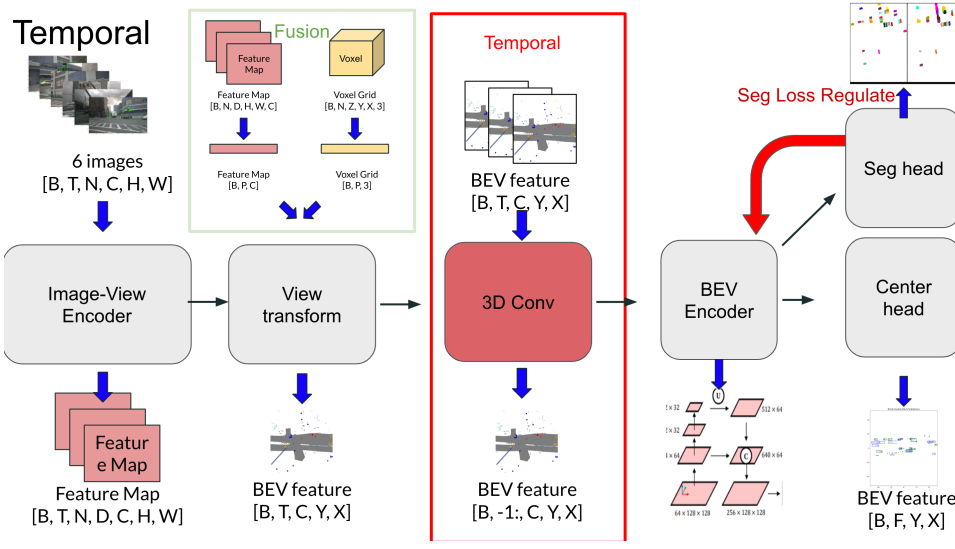

Fig. 1. The figure contain several subtasks to conduct 3D object detection according to given image.

This section delivers our proposed architecture for 3D object detection in the Bird-Eye-View from multiple perspective views. In this task, N monocular views consisting of perspective images, camera intrinsic matrices, and extrinsic matrices are given. The goal of our model is to generate 7 degrees of freedom to represent 3D object bounding boxes in the world coordinate.

3.1 Overall Architecture

We use modular design to build up our model, which is composed of 5 modules: An image-view encoder, a view-transform module, a temporal module, a BEV encoder, and a detection head. The architecture of our model is illustrated in 1.

3.2 Image-View Encoder

To extract features from images and map them to corresponding 3D positions, we follow the steps of [8, 9]. To begin with, we use EfficientNet [11] as a feature extractor. It generates channels C of feature maps from monocular images as well as predicts additional channels D which represents depth distribution. Then, we use the outer product to generate tensors that represent features of 3D positions in the camera coordinates. The tensor $\in \mathbb{R}^{B\times N\times D\times C\times H\times W}$.

3.3 View transform

Given feature maps from 6 monocular perspective-view images, we need to transform features into the ego-vehicle coordinate. First, we generate frustums as the size of grids in perspective-view(camera coordinate). Second, we use both intrinsic and extrinsic to calibrate frustums to voxels in ego-vehicle coordinate. In this step, the feature from images may be distorted in the ego-vehicle coordinate.

3.4 3D Conv

In this step, we feed the Bird-Eye-View features, with the timestamp is referred to, into a 3D convolution neural network. After conducting the weighted sum of each timestamp, the bird-eye-view features with temporal information are extracted.

3.5 BEV Encoder

With a similar method of the image-view encoder, we use Feature Pyramid Network (FPN) to extract features on different scales. The bird-eye-view features from the previous step are fed into the ResNet101 backbone and aggregated into a single feature map. The final feature map is the input of the next step.

3.6 Center head

To generate 3D bounding boxes in the world coordinate, 7 degrees of freedom containing values representing positions, scales, and orientations of both movable and static objects in the world are regressed. In order to avoid the difficulties regressing orientations of objects, we follow the previous study [1, 2] to use the detection head of CenterNet [2] to generate our targets.

4 Experiments and Results

4.1 Dataset and Implementsation Details

Dataset We use the Nuscene [12] dataset, which provides both camera and Lidar sensor data with 3D bounding box annotations, as our benchmark. It contains 1000 scenes and is divided to 700 / 150 / 150 scenes for training / validation / testing. Each scene is 20 seconds in length and annotated at 2HZ. The official evaluation metrics contain scenes Detection Score (NDS), mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error(mAOE), mean Average Velocity Error(mAVE), and mean Average Attribute Error(mAAE).

Implementation Details We use adam optimizer with a learning rate of 1e-4 to train our model. We train our model for 30 epochs on Tesla V100 for 48 hours. We use input images at resolution 256×704 for a static model. For the temporal model, we use the image size of 224×480 due to the computational cost. The size of the BEV map is 200×200 with range $[-50\text{m},\ 50\text{m}]$ and resolution 0.5m per BEV grid.

4.2 Quantitative Results

Results of Nuscene benchmark Table 1 shows our result on the Nuscene benchmark. As we can see, we build a model with competitive performance in comparison to different modalities. The mAP of PointPillar [1] is 0.305, BEVDet [13] is 0.288, and our work is 0.218. Other metrics like mASE, mAAE, and NDS score are also close to PointPillar. Note that PointPillar uses LiDAR to detect, but BEVDet [13] and our model only use the camera modality. This shows that camera-only perception has bridged the gap between LiDAR. However, the mAVE is not comparable with the LiDAR method. We believe that temporal information plays an important role in estimating velocity since it is hard to predict the correct velocity just from static images.

4.3 Ablation Study

Table 2 shows the results of ablation studies. We use SECOND [14] as the backbone and SECOND FPN to refine the BEV features. The baseline mAP is 19.14. After enabling the segmentation loss guiding module, the mAP increases to

Table 1. The table shows the result of Nuscene benchmark comparison with SOTA method.

Methods	Image Size	Modality	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
PointPillar	-	LiDAR	0.305	0.5170	0.290	0.500	0.316	0.368	0.453
BEVDet	256 * 704	Camera	0.288	0.722	0.269	0.538	0.911	0.270	0.373
Ours	256 * 704	Camera	0.218	0.781	0.365	0.7993	1.3034	0.349	0.2799

21.76. This indicates the effectiveness of using segmentation task as the regularization on BEV feature maps. Moreover, we deploy the image augmentation and increases the score to 21.86.

Effectiveness of the temporal module. Table 3 shows the effectiveness of utilizing temporal information. The mAP score increases when we take more timestamps at once. It is worthwhile to note that the mean Average Velocity Error (mAVE) decreases after using the temporal module. This improvement indicates that the model learns to predict velocity better with the information of previous frames.

Table 2. The table shows the ablation study; B + FPN means the components in the BEV encoder, we use SECOND [14] as the backbone and SECOND FPN as Feature Pyramid Network; SL means that we use segmentation loss to regularize the output of the bird-eye-view heat map; IA means the usage of front-view image augmentation.

Method: Pseudo Lidar(PL)	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
PL + B + FPN	19.14	0.7866	0.369	0.8615	1.3526	**0.346**	0.2594
PL + B + FPN + SL	21.76	**0.7534**	**0.3617**	0.8204	1.3428	0.3536	0.2799
PL + B + FPN + SL + IA	**21.86**	0.781	0.365	**0.7993**	1.3034	0.349	**0.2799**

Table 3. The table shows the performance of different frames. 3 Frames means that we take 3 timestamps as a single sample. Note that the image size of these experiments is 224×480 due to the computational cost.

Method	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
Static	0.1723	0.8146	0.3678	0.9341	1.2616	0.3537	0.2392
2 Frames	0.1767	0.8117	**0.3742**	**0.8235**	1.1778	0.3637	**0.251**
3 Frames	**0.1786**	**0.8018**	0.3762	0.8949	**1.1407**	**0.3534**	0.2467

4.4 Qualitative Results

Temporal model Figure 2 delivered the qualitative results of the bounding box in bird-eye-view. Referring to the figure 2, we suggest that the temporal information provides much more clear information for detection as the bounding box predictions are more certain.

Velocity Regularization Figure 3 deliberated the qualitative results of velocity regularization. As we can see the heat map of the model with velocity regularization can generate the prediction of invisible objects. Consequently, we assume that the velocity regularization and temporal information can resolve the occlusion problem.

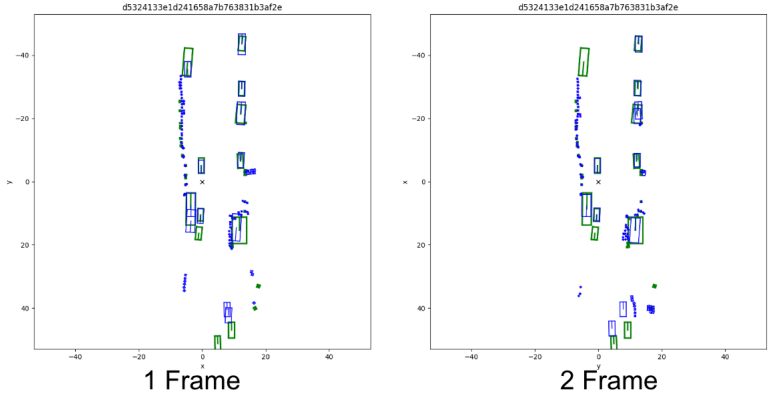

Fig. 2. The figure contain qualitative results of model with different frames; Green bounding box: GT, Blue bounding box: Predictions.

5 Conclusions

In this project, we designed an end-to-end model based on the framework of [8], [9]. The model predicts the information of the 3D bounding box according to the given 6 monocular camera images. This is the first object detection work using the multi-camera sensor on the Nuscenes dataset. we conduct the baseline experiments and bring the novel idea for further work in 3D object detection.

6 Contribution

Table 4. The table show division of work.

Member	Work	Contribution(%)
R10922045 Ching-Yu Tseng	Research.	40
R10922066 Yi-Rong Chen	Implementation.	40
R10922192 Ya-Ching Hsu	Survey.	20

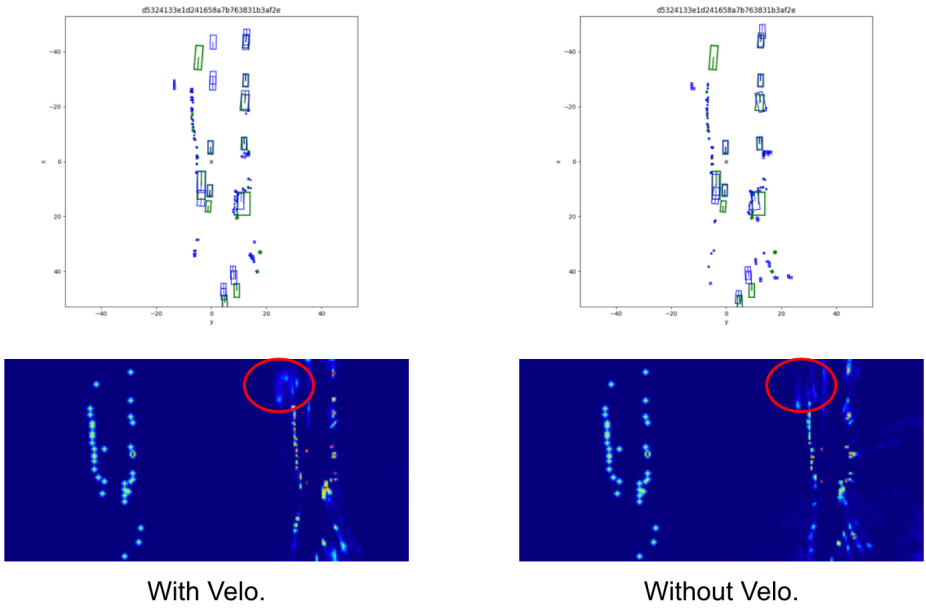

Fig. 3. The figure contain qualitative results of model with and without velocity; Green bounding box: GT, Blue bounding box: Predictions.

References

1. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 12697–12705
2. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2021) 11784–11793
3. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 913–922
4. Reading, C.: Categorical depth distribution network for monocular 3d object detection. arXiv (2021)
5. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning, PMLR (2022) 180–191
6. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018)
7. Weng, X.: Monocular 3d object detection with pseudo-lidar point cloud. arXiv (2019)
8. Philion, J.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. arXiv (2020)

9. Hu, A.: Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. ICCV (2021)
10. Wang, T., Xinge, Z., Pang, J., Lin, D.: Probabilistic and geometric depth: Detecting objects in perspective. In: Conference on Robot Learning, PMLR (2022) 1475–1485
11. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, PMLR (2019) 6105–6114
12. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
13. Huang, J., Huang, G., Zhu, Z., Yun, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
14. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors (2018)