Object Detection and Localization Based on Binocular Vision for Autonomous Vehicles

Mingchi Feng¹*, Yibo Liu¹, Panpan Jiang¹, Jingshu Wang²

¹ Advanced Manufacturing Engineering School, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
² College of Mechanical Engineering, Chongqing University of Technology, Chongqing 400054, China

* Corresponding author’s e-mail: donotbreeze@163.com

Abstract. Environment perception based on vision plays an important role in autonomous driving technology. Although vision perception has achieved certain results in recent years, many methods can not solve the contradiction between speed and precision. In this paper, we propose a system for fast and accurate object detection and localization based on binocular vision. For object detection, a neural network model based on YOLOv3 is proposed. Specifically, MobileNet is employed in the backbone of YOLOv3 to improve the speed of feature extraction. Then the corresponding ORB feature points are extracted from continuous stereo images which take from the binocular cameras on the moving car. Thus, the disparity of each ORB feature point is calculated. After that, we use the result of object detection to screen the ORB feature points. Finally, the depth of the targets in the traffic scene can be estimated. Experiments on the KITTI dataset show the efficacy of our system, as well as the accuracy and robustness of our object localization relative to ground truth and prior works.

1. Introduction
Understanding the complex traffic scene in presence of objects such as vehicles and pedestrian is crucial for autonomous vehicle. In order to get the target information, autonomous vehicles use sensors to sense their surroundings, such as LiDAR, millimetre wave radar and cameras. Zeng Yiming et al. [1] present a 3D vehicle detection method that utilizes pure LiDAR point cloud to predict the location, orientation, and size of vehicles. S. Sugimoto et al. [2] propose an object detection and location method based on Millimetre Wave Radar. But the high cost of LiDAR is not conducive to the popularization of automatic driving system, and fog and rain will weaken the detection capability of millimetre wave radar. Cameras are particularly suited for this task as they are much cheaper than the other sensors, and can provide, through dedicated fast algorithms, both object detection and depth estimation. In addition, using a binocular camera enables 3D reconstruction of traffic scene from each images, which can be used for the targets localization. In the design of an embedded object detection process, three main constraints have to be accounted for: real-time processing, high reactivity, precise management of measurement and estimation errors to assess the reliability of the decisions. We address these three constraints in our work.

To detect the object in traffic scene, [3] presented a method for detecting objects in images using a single deep neural network. This method discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction stage, the network generates scores for the presence of each object category in each default box and produces
adjustments to the box to better match the object shape. Choi et al. [4] improved YOLOv3 [5] with the help of gaussian distribution so that the network can output the uncertainty of each detection box, thus improving the precision of the network. Choi’s method achieved 83.61% mAP (mean average precision) at the speed of 43.13 FPS on KITTI dataset on NVIDIA Titan V. But tested on NVIDIA Jetson TX2, the speed of detection can only reach 8 FPS. Markus Enzweiler et al. [6] proposed a novel attention mechanism to improve stereo-vision based object recognition systems in terms of recognition performance and computational efficiency at the same time. Markus Enzweiler’s method can detect the obstacle in front of the car at the speed of 50 FPS on KITTI dataset on NVIDIA Jetson TX2. Although they can get the location of the obstacles at a high speed, they can not get the type of the obstacles. Andreas Geiger et al. [7] proposed a method which provides three-dimensional position information of moving objects in a world coordinate system. But it costs too much computing resources for scene flow clustering based on Delaunay triangulation. Therefore, there are still many problems in object detection and localization, we need to take into account both detection speed and measurement accuracy.

In summary, many existing methods can only handle small areas for the object detection and localization. Whether the algorithm can balance the real-time processing and accuracy has become a major problem. To solve this problem, we propose an object detection and localization method based on lightweight detection neural network and feature points cluster.

2. System Overview

As shown in Figure 1, the method proposed in this paper is divided into three parts: object detection, feature points extraction and feature points clustering. Firstly, the bounding boxes of targets in front of the vehicle are detected in left image based on MobileNet-YOLOv3 neural network. Then, feature points are detected in two consecutive stereo image pairs and the coordinates of every feature point are calculated. Finally, we delete the feature points that are not within the bounding boxes, and cluster the left feature points based on Euclidean Cluster algorithm. The classification and the center coordinate of the targets are available after this progress.

Figure 1. Schematic diagram of the proposed system.

2.1. Object Detection

In order to make the target detection algorithm run in real time on the embedded device, we replaced the backbone network in YOLOv3 with MobileNet [8] to accelerate feature extraction time. The basic convolution unit of MobileNet called depthwise separable convolution, which consists of depthwise convolution and pointwise convolution.

Standard convolutions have the computational cost of:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

where the computational cost depends multiplicatively on the number of input channels $M$, the number of output channels $N$ the kernel size $D_K \times D_K$ and the feature map size $D_F \times D_F$.

Depthwise separable convolutions have the computational cost of:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

which is the sum of the depthwise and $1 \times 1$ pointwise convolutions.
By expressing convolution as a two steps process of filtering and combining we get a reduction in computation of:

$$\frac{D_k \cdot D_k \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_k \cdot D_k \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} \frac{1}{D_k^2}$$  \hspace{1cm} (3)

In the feature extraction phase of the network, MobileNet will reduce the cost of computation obviously.

As shown in Figure 2, the last three network layers in MobileNet: average layer, pooling layer and softmax layer were discarded. Darknet-53, which is the original backbone in YOLOv3, was replaced with the rest of MobileNet. The 11th and 13th pointwise convolution layer in MobileNet and the final output of MobileNet were selected as the feature graphs of the three scales to carry out the multi-scale prediction of YOLOv3.

It has excellent target detection performance in different traffic scenarios, the Figure 3 shows that MobileNet-YOLOv3 can accurately detect the bounding boxes of objects, such as cars, pedestrians and traffic signs.

Table 1 shows the performance of the proposed algorithm and other methods using the KITTI validation set on NVIDIA Jetson TX2. Although the mAP of MobileNet-YOLOv3 with a 512 × 512 resolution is 2.20 lower than that of YOLOv3, which has the highest accuracy among the previous methods, it is noteworthy that the fps of the proposed method is 3 times better than that of YOLOv3. After the process of object detection based on MobileNet-YOLOv3, the classification and bounding boxes of the targets are available.

Table 1. Performance comparison using KITTI validation set

| Detection algorithm | mAP% | FPS | Input size |
|---------------------|------|-----|------------|
| SSD                 | 61.29| 5.1 | 512×512    |
| YOLOv3              | 80.52| 8.3 | 512×512    |
| MobileNet-YOLOv3    | 78.32| 24  | 512×512    |

2.2. Feature points extraction

In this section, we extracted the feature points by ORB algorithm\(^9\) and compute the coordinate of every feature point. The configuration of the binocular camera is shown in Figure 4. Assumed that the binocular camera has completed calibration and polar alignment, the input images for our system have been rectified. And set the left camera frame as the world frame.
The feature points \( x(u, v) \) were extracted by ORB algorithm in left and right frame at time \( k, k - 1 \). As shown in Figure 4, the robust feature points were left after loop match \( (x_{L,k-1} \rightarrow x_{R,k-1} \rightarrow x_{R,k} \rightarrow x_{L,k} \rightarrow x_{L,k-1}) \).

The world coordinates \( P(X, Y, Z) \) of feature points were calculated by the model of binocular camera, and the model is given by:

\[
X = \frac{(u_L - c_u) \cdot b}{d} \quad (4)
\]

\[
Y = \frac{(v_L - c_v) \cdot b}{d} \quad (5)
\]

\[
Z = \frac{b \cdot f}{d} \quad (6)
\]

where \( b \) represents the baseline of the binocular camera, \((c_u, c_v)\) is the principal of the camera, and \( f \) is the focal length.

2.3. Feature points cluster

The current feature points extracted from last section which include all feature points of the traffic scene in front of the vehicle. And it is impossible to directly distinguish the feature points which belong to the specific targets. Therefore, the target detection algorithm was used to screen the feature points in the left image, deleted the feature point which not in the bounding boxes. Feature points were preserved in the bounding boxes area. The results are as shown in Figure 5, the redundant feature points were deleted by the object detection result.
It is convenient to reject the points which do not belong to the targets, but there are some background feature points remain. These points were close to the targets in image. After these points were projected on the ground plane, we found that they were far from the object centre. So we applied the Euclidean Cluster algorithm\(^{[10]}\) to delete these points: For a certain point \(P(x, y, 0)\) in the space, the nearest \(k\) points to \(P\) were found by the KD-Tree nearest neighbor search algorithm. And those points whose distance was less than the threshold are clustered into set \(Q\). If the number of elements in \(Q\) did not increase, the whole clustering process ended. Otherwise, we selected another point in set \(Q\) and the above process was repeated until the number of elements in \(Q\) did not increase. Finally, we regarded the geometric centre of the point set \(Q\) as the centre of the target. And we constructed the minimum enclosing rectangle of the set \(Q\) as the shape of the target. The results are as shown in the Figure 6.

![Figure 6. The result of Euclidean Cluster](image)

### 3. Experiments and Discussion

We tested our algorithm on KITTI dataset, which provides binocular camera parameters, RGB images taken by the binocular camera and the ground truth of targets. So the model of binocular camera can be built according to the camera parameters, and the results of our algorithm can compare with the targets ground truth, which includes the type and location of the targets. We deployed the algorithm on NVIDIA Jetson TX2.

Our algorithm was tested in different sequence of KITTI dataset. The visualization of the experimental results is shown in the Figure 7. We found that the detection at close and far range up to 60m works well in the sparse traffic scene. But when there was occlusion between the objects, some targets were lost.

![Figure 7. The result of object detection and location in image and top view](image)

As for the positioning error, we calculated the distance from the geometric centre of the output target to the origin of the vehicle coordinate system and compared it with the real value. We extracted the
result of target localization error in a certain frame from the test sequence. As shown in the Table 2, we found that the positioning error was about 4.1% from range 0 to 60m.

| Sequence ID | No. Frame | Error % | Object ID | FPS |
|-------------|-----------|---------|-----------|-----|
| 0004        | 1         | 2.1     | 1         | 44  |
| 0047        | 169       | 5.7     | 4         | 18  |
| 0056        | 222       | 6.3     | 8         | 12  |
| Total       | 4.1       |         | 13        | 24  |

4. Conclusion and future work

In this paper, we proposed a system for fast and accurate object detection and localization based on binocular vision. The method of object detection based on MobieNet-YOLOv3 got the targets bounding boxes in left image. Feature points were extracted by ORB algorithm, and their vehicle coordinates were calculated. The centre of clustered points and their minimum enclosing rectangle described the final system output. Our method worked well in 0 to 60m at 24 FPS.

But our method was flawed in detecting targets that are too far away and blocking each other. Our next step will build an end-to-end neural network to solve the problems existing in our algorithm. We will further improve the robustness and the detection range of the algorithm.

Acknowledgments

This work supported by the National Natural Science Foundation of China (51505054), the Chongqing Science and Technology Commission (cstc2019jcyjA0538), and the Chongqing Municipal Education Commission (KJ1500405).

References

[1] Zeng, Y., Hu, Y., Liu, S., Ye, J., Han, Y., Li, X., & Sun, N. (2018). Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving. IEEE Robotics and Automation Letters, 3(4): 3434-3440.

[2] Wang, X., Xu, L., Sun, H., Xin, J., & Zheng, N. (2016). On-road vehicle detection and tracking using MMW radar and monovision fusion. IEEE Transactions on Intelligent Transportation Systems, 17(7): 2075-2084.

[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In: European conference on computer vision. Las Vegas. pp. 21-37.

[4] Choi, J., Chun, D., Kim, H., & Lee, H. J. (2019). Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. Seoul. pp. 502-511.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Amsterdam. pp. 779-788.

[6] Enzweiler, M., Hummel, M., Pfeiffer, D., & Franke, U. (2012). Efficient stixel-based object recognition. In: 2012 IEEE Intelligent Vehicles Symposium. Alcalá de Henares. pp. 1066-1071.

[7] Lenz, P., Ziegler, J., Geiger, A., & Roser, M. (2011). Sparse scene flow segmentation for moving object detection in urban environments. In: 2011 IEEE Intelligent Vehicles Symposium. Baden-Baden. pp. 926-932.

[8] Kim, W., Jung, W. S., & Choi, H. K. (2019). Lightweight driver monitoring system based on multi-task mobilenets. Sensors, 19(14): 3200.

[9] Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5):1255-1262.

[10] Assem, I., & Dupont, G. (2011). Friezes and a construction of the Euclidean cluster variables. Journal of Pure and Applied algebra, 215(10): 2322-2340.