Object Detection and Classification for Autonomous Vehicle

B. Rajesh, D. Ramakrishna, A. Ramakrishna Raju, Ameet Chavan

ECE Department, Sreenidhi Institute of Science and Technology, Hyderabad, TS – 501 301, INDIA
ramakrishnaraju@sreenidhi.edu.in, ameetchavan@sreenidhi.edu.in

Abstract: The proposed work presents efficient objection detection and tracking algorithm for autonomous vehicles, which is developed in MATLAB with Image Processing Computer Version Toolbox and Automated Driving Toolbox. The developed algorithm track and detects moving and stationary objects such as other vehicles, pedestrians, and traffic lanes. Accurate and efficient tracking are important to analyze object behavior. For this work various build in detectors from MATLAB tool box were tested and compared. The evaluation of algorithm was carried on for 19 short videos from 8 seconds to 23 seconds, and then it applied to K-dataset and full road experiments by the Automated Driving Lab. The full road tests are between one to five minutes, including CAR to CAR WEST, CAR WEST to CAR, and campus marked roads.

Keywords: Advanced Driver Assistance System (ADAS), Autonomous Vehicle, Region-based Convolutional Neural Networks(R-CNN), Gaussian Mixture Model (GMM), Aggregate Channel Feature (ACF).

1. Introduction

Real-time object detection, identification and tracking is a major challenge in developing computer vision algorithms. Accurate identification and recognition of objects in one or two image frames is critical for Advanced Driver Assistance Systems (ADAS) that are integrated in vehicles [1-4]. ADAS algorithms aid driver to identify and recognize objects which otherwise would be compromised due to lack of attention, anonymous objects, reflection of light etc. Inefficient algorithm design compromises safety and may result in accidents especially at high speed or under heavy traffic conditions. To overcome the said problem ADAS for autonomous vehicles require unique classification and detection of object under different scenarios [5]. Other applications of object detection are advanced robotics, surveillance systems, defense systems, face recognition and space research etc. [4] discusses the typical steps involved in object classification and identification algorithms for computer vision are – 1) Classification of objects in image, 2) Localization of objects in image, 3) Object detection in image and 4) Segmentation of image.

In addition to the above steps, ADAS in self-driving vehicles use advance sensor fusion techniques for detecting and extracting characteristics from moving or stationary objects [6]. There are many object detection algorithms have been proposed that are based on fuzzy, statistics, neural networks etc. Many of these algorithms have complex theory and require deeper understanding, implementation and experimentation before incorporating. Real-time images and videos of various objects and structures are bundled as datasets that provide information of the environment which is used for both training and testing. Using this information distance and direction of the object relative to the test vehicle is measured. Detection and reaction are achieved by combining extraction model with response model in the proposed algorithm. Counting is another important aspect of the algorithm, that involves estimating real-time the number of objects in the image. Once the objects are detected, classified, located and counted from the image, the object is tracked from the extracted features by the proposed algorithm. To perform all the afore mentioned tasks in real-time, it requires high performing hardware. In this paper image segmentation is employed to track without any errors using MATLAB automated tool box. The following section of the paper discusses object detection algorithm, road test results and conclusion.
2. Object detection algorithm

2.1 Camera Calibration

Camera calibration estimates camera intrinsic and extrinsic parameters. Intrinsic parameters are including focal length, principal point and image size. Extrinsic parameters are including pitch, yaw and roll, which can also write as a rotation matrix $R$ and translation vector $t$. Intrinsic parameters are to determine the relation between camera coordinate and pixel coordinate. Extrinsic parameters are to determine the relation between camera image coordinate and real-world coordinate. As figure 1 shows, in the full road tests nineteen figures have been collected with various angles and distances. Angles have to be less than forty-five degree [7].

![Camera Calibration Image](image_url)

Camera extrinsic parameters during all experiments are relative to sensor mounting location at the vehicle, which are mounting height, mounting pitch, mounting roll and mounting yaw [8], as figure 2 (a). Mounting height can be directly measured relative to ground, but other angles need to turn manually according to the bird’s eye view image. Figure 2 (b) and (c) indicates extrinsic parameters affections on bird’s eye view. Road and lanes in bird’s eye view should be parallel after successful turned with correct mounting angles. Although Camera Calibrator App can export camera extrinsic parameters, that is not same extrinsic parameters expected. As figure 3 shows, extrinsic parameters from the app are mapping matrix between image coordinate and camera coordinate. However, the expected extrinsic parameters are translation matrix between vehicle coordinate and camera coordinate as figure 2 shows.

2.2. Lane Boundaries Detection and Tracking

The algorithm follows a visual perception example by MATLAB document. After a physical defined region of interests, possible lane width and detection sensitivity, a built-in function Segment Lane Marker Ridge to detect lane points at grayscale bird’s eye view image, in figure 2 (d) shows [9]. A region of interests, detection sensitivity and lane approximate width need to adjust according to different scenarios. The x coordinate sampling rate is 0.0482 meter, so there are thousands x y lane boundary points in each frame. Image to vehicle and vehicle to image could transfer location between image coordinate pixels and vehicle coordinate meters. Coordinate systems show as figure 2. Image to vehicle first applied to translate bird’s eye view coordinate to vehicle coordinate. [10] To improve
point detection, find parabolic lane boundaries is to separate all vehicle coordinate points by the number of lanes as a parabolic model and x y points.

For each parabolic model includes 3 coefficients as a second-degree polynomial equation. In addition, that built-in function can also provide strengths of lanes, boundary type, x maximum position and x minimum position. Weak strength points, such as vehicle shadows and windscreen reflection, need to remove. Lane types as solids, dash etc. could reflect real road conditions. Lane length calculated by subtracting of x maximum and minimum points. After compared with a threshold length defined by a percentage length of maximum lane length in RoI, short length boundaries, such as zebra crossing, need to remove. In the last step, lane vehicle coordinates points would transfer to video frame image coordinate points by vehicle To Image function. Figure 4 shows the step to detect points in bird’s eye view and project them to frame.

![Vehicle and camera coordinate system](image1.png)

(a) Vehicle and camera coordinate system, (b) Bird’s eye view before (left), (c) after (right) correct mounting angles and (d) Lane point detection in grayscale bird’s eye image.

![Extrinsic parameters by apps](image2.png)

Figure 3 Extrinsic parameters by apps, based on image and world coordinate
2.3. Vehicle and Pedestrian Detection

Automated driving toolbox and computer vision toolbox provide various pre-trained detectors, including Aggregate Channel Feature (ACF), Gaussian Mixture Model (GMM), Region-based Convolutional Neural Networks(R-CNN), Fast R-CNN and Faster R-CNN [11-12]. The last three detectors are deep learning methods, which need more processing time compared with others in the test. GMM detector is not accurate compared with ACF detectors in nineteen short traffic videos experiments by Automated Driving Lab. Therefore, ACF is the main detector in this work. Although ACF detector has better work compared with other pre-trained detectors, it still has problems [13-14]. It is not always consistent tracking and some frame may miss label. Ground Truth Labeler app provides temporal interpolator algorithm, which can estimate label in the intermediate frame by interpolation. Another main problem is the rectangle label is not consistent with vehicle size, and the size may vary in different frames. Ground Truth Labeler app would also solve the problem, but it may need volume manual work.

2.4. Testing and Verification

The tracking algorithm proposed for vehicles and pedestrians is based on Kalman filter which in turn is built on constant velocity model [15]. The pedestrian detection system is a critical part of the system and mandates accuracy and reliability with real-time performance. The pedestrian detection system was tested and verifies taking samples of real traffic scenarios. Although the system provided correct results, it requires tuning and adaptive learning which can consume resources. Therefore, a simulation evaluation tool was used to develop the algorithm to test for real world situations and evaluate the response. Through this algorithm the pedestrian accident data was first analyzed against the other conventional algorithm to find and record its statistical distribution characteristics.

3. Road Test Results

The algorithm applied to these four tests does not show a great result. It has two main limitations. The first one is when lane width changes or separation, lane points would beyond pre-defined RoI. In complex road conditions, constant RoI will decrease detection accuracies and missing detection. Therefore, the lane detection algorithm has better to detect in constant road condition without lane width changes or division. Another limitation for lane detection algorithm is at the intersection. Since

Figure 4 Lane detection in Bird's eye view and video frame
it has a gap but does not reach the threshold, which is not enough strength or length, the detection lanes are not accurate neither able to remove. The detection points could include other direction lanes and all other shadows by vehicles. Theoretically, ground truth labeler app can remove inaccurate points in the frame. However, due to a volume of sampling points, physics work increases by selecting inaccurate points.

Besides hand-operated work by the app, single detection missing frame could update with last tracking points since each frame only takes 0.03 seconds. For a certain period, detection-missing frames, manual work need by ground truth labeler app. To reduce ground truth app processing time, only missing frames import to the app to adding detections. The camera frame rate is 30 frames per second so the frame number calculated as follows

\[ Frame = 30 \times time(seconds) + 1 \]  

(1)

3.1. Vehicle and pedestrian detection and tracking results and conclusion

The algorithm applied to these four tests does not show a great result. It has two main limitations. The one of important limitation is the windy and snow shower weather in testing day, as figure 5 shows. Image qualities are not satisfied as the previous test. The detector is only able to detect objects which close to the testing vehicle. Increasing threshold distance and sensitivity is unable to increase detection abilities. In addition, some frame was missing detection due to low image qualities. Therefore, mass manual work needed by ground truth labeler app.

Figure 5 Tracking with test video - Detection in heavy traffic

Another limitation is by ACF detector. ACF detector of vehicles and pedestrians may have inaccurate tracking since it defined as object size. Moreover, the detection rectangle size is not constant, and with small shakings. However, that will not affect object location calculation. ACF may miss detection in heavy traffic condition. Ground truth labeler app would help to manually add detections. Example as figure 5 shows. Overall, considering weather condition, the accuracy of vehicle and pedestrians’ detection at a short range is great, as figure 6 shows.
Figure 6. Tracking with test video - Detection in heavy traffic

3.2 Tracklet labeler

Tracklet labeler is created based on a timetable is shown in Table 1. For vehicles, pedestrians and stop signs which using rectangle label tracking, have an M x 4 matrix at a specific time. M presents the number of objects at the specific time and four presents image points of tracking rectangle in x, y, width and height. For Lane Marker, it would be 3 element coefficients to present a parabolic lane. Ground truth labeler app would export a group of image points for lanes. Since all these points are based on image coordinate, sensor calibration results are needed to transfer into vehicle coordinate when objects position needed and figure 8 shows output of object detection and line detection.

| Time          | 1 Cars  | 2 Pedestrian | 3 Stop | 4 LaneMarkers |
|---------------|---------|--------------|--------|---------------|
| 0 sec         | [ ]     | [ ]          | [ ]    | [ ]           |
| 0.03328 sec   | [ ]     | [ ]          | [ ]    | [ ]           |
| 0.066656 sec  | [ ]     | [ ]          | [ ]    | [ ]           |
| 0.099964 sec  | [ ]     | [ ]          | [ ]    | [ ]           |
| 0.13331 sec   | [140.01163... | [ ]          | [ ]    | [ ]           |
| 0.16664 sec   | [144.03483... | [ ]          | [ ]    | [ ]           |
| 0.19997 sec   | [151.8793... | [ ]          | [ ]    | [ ]           |
| 0.2333 sec    | [161.64683... | [ ]          | [ ]    | [ ]           |
| 0.26662 sec   | [166.06563... | [ ]          | [ ]    | [ ]           |
| 0.29995 sec   | [171.1142... | [ ]          | [ ]    | [ ]           |
| 0.33328 sec   | [176.49433... | [ ]          | [ ]    | [ ]           |
| 0.36661 sec   | [182.13923... | [ ]          | [ ]    | [ ]           |
| 0.39994 sec   | [187.7841... | [ ]          | [ ]    | [ ]           |
| 0.43326 sec   | [123.04792... | [ ]          | [ ]    | [ ]           |
| 0.46659 sec   | [96.380029... | [ ]          | [ ]    | [ ]           |
| 0.49992 sec   | [85.329252... | [ ]          | [ ]    | [ ]           |
| 0.53325 sec   | [74.259028... | [ ]          | [ ]    | [ ]           |
| 0.566658 sec  | [63.195528... | [ ]          | [ ]    | [ ]           |

Table 1 – Other Vehicle Trackerlet labeler
4. Conclusion

The proposed algorithm developed in MATLAB detects and tracks objects accurately on test videos for traffic lanes, other vehicles and pedestrians. The algorithm was verified for 19 short videos from Automated Driving Lab, including large curve, ramp and busy traffic scenarios. The tracking of vehicles, pedestrians and lane shows great result compared to ACF detector. Furthermore, the algorithm was applied to Vision Sensor Dataset and Full Road Test with longer than one minute. However, under dim lighting and extreme busy road conditions algorithm showed some limitations in object detection and tracking. This necessitates applying a sensor fusion model which integrates multiple sensors such as vision, radar, and IMU. This model would make the forward collision warning system of the vehicle more effective.

References

[1] K. Pranav and J. Manikandan, "Design and Evaluation of a Real-time Pedestrian Detection System for Autonomous Vehicles," 2020 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2020, pp. 155-159, doi: 10.1109/ZINC50678.2020.9161768.
[2] R. K. Rout, "A survey on Object Detection and Tracking Algorithms," 2013.
[3] B. Anand, V. Barsaiyan, M. Senapati and P. Rajalakshmi, "Region of Interest and Car Detection using LiDAR data for Advanced Traffic Management System," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020, pp. 1-5, doi: 10.1109/WF-IoT48130.2020.9221354.
[4] A. Naik, G. V. V. S. Naveen, J. Satardhan and A. Chavan, "LiEBiD - A LIDAR based Early Blind Spot Detection and Warning System for Traditional Steering Mechanism," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichi, India, 2020, pp. 604-609, doi: 10.1109/ICOSEC49089.2020.9215366.
[5] L. Koraqi and F. Idrizi, "Detection, identification and tracking of objects during the motion”," 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2019, pp. 1-6, doi: 10.1109/ISMSIT.2019.8932833.
[6] J. Kim, D. S. Han and B. Senucci, "Radar and Vision Sensor Fusion for Object Detection in Autonomous Vehicle Surroundings," 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, 2018, pp. 76-78, doi: 10.1109/ICUFN.2018.8436959.
[7] J. Ciberlin, R. Grbic, N. Teslic and M. Pilipovic, "Object detection and object tracking in front of the vehicle using front view camera," 2019 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2019, pp. 27-32, doi: 10.1109/ZINC.2019.8769367.
[8] Kyeong yuk min, Jun Wang. “Low-cost implementation of bird's-eye view system for camera-on-vehicle” IEEE Conference on consumer electronics 2010.
[9] C. Sun et al., "Proximity based automatic data annotation for autonomous driving," in IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 2, pp. 395-404, March 2020, doi: 10.1109/JAS.2020.1003033.
[10] M. Ramirez-Martinez, F. Sanchez-Fernandez, P. Brunet, S. M. Senouci and E. Bourennane, "Dynamic management of a partial reconfigurable hardware architecture for pedestrian detection in regions of interest," *2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, 2017, pp. 1-7, doi: 10.1109/RECONFIG.2017.8279787.

[11] Indrabayu, Bakti.R, Areni.I and Ais.A, "Vehicle detection and tracking using Gaussian Mixture Model and Kalman Filter," in IEEE, Makassar, Indonesia, 2016.

[12] M. Alavianmehr, A. Zahmatkesh and a. Sodagaran, "A New Vehicle Detect Method Based on Gaussian Mixture Model along with," in The 14th International Conference on Traffic and Transportation Engineering, 2014.

[13] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142-158, 2016.

[14] Shireen Y. Elhabian, Khaled M. El-Sayed, Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art, Recent Patents on Computer Science 2008, 1, 32-54.

[15] N. Y. Ko, G. Song, W. Youn, I. H. Choi and T. S. Kim, "Improvement of Extended Kalman Filter Using Invariant Extended Kalman Filter," *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, Daegwallyeong, 2018, pp. 948-950.