Real-time vehicle detection and tracking in video based on faster R-CNN

Yongjie Zhang[1], Jian Wang[1, 3] and Xin Yang[2]

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China;
[2]Institute of Information Optics Engineering, Soochow University, Suzhou, Jiangsu 210056, China.
Email: jian-wang@tsinghua.edu.cn

Abstract. Vehicle detection and tracking is a significant part in auxiliary vehicle driving system. Using the traditional detection method based on image information has encountered enormous difficulties, especially in complex background. To solve this problem, a detection method based on deep learning, Faster R-CNN, which has very high detection accuracy and flexibility, is introduced. An algorithm of target tracking with the combination of Camshift and Kalman filter is proposed for vehicle tracking. The computation time of Faster R-CNN cannot achieve real-time detection. We use multi-thread technique to detect and track vehicle by parallel computation for real-time application.

1. Introduction

In auxiliary driving system, vehicle detection and tracking is a critical technology that plays an important role in vehicle active safety. The complex vehicle patterns in video frame from real scene make a challenge for real-time detection and tracking. The detection of vehicle using traditional image processing method is impossible in some complex conditions. Deep learning method is a promising alternative for natural scene target detection. Better results have been achieved using Faster R-CNN [1] proposed by Ren in some challenging datasets such as ImageNet and VOC, as well as professional datasets, such as KITTI [2].

In this paper, Faster R-CNN is applied to vehicle detection for auxiliary driving system. The neural network model is trained using partial KITTI dataset based on Faster R-CNN. An algorithm of target tracking with the combination of Camshift and Kalman filter is proposed for vehicle tracking. For real-time detection, multi-thread technique is used for vehicle detection and tracking. The experiment results validate our proposed method.

2. Overview of Faster R-CNN

Traditional methods for object detection based on deep learning employ sliding window or region proposal to generate object hypotheses, and then determine their classifications by a classifier. However, the process of region proposal computation is very time consuming, which is a bottleneck for real-time application.

To deal with this problem [3], Faster R-CNN proposes a Regional Proposal Network (RPN) [1], which shares the fully convolutional layers with a Fast R-CNN [4] object detection network. RPN is a fully convolutional network, which predicts both object proposals and objectness scores. The proposals and scores are fed into Fast R-CNN network for model training. The Faster R-CNN algorithm reduces time of computation and realizes good object detection performance with high mean Average Precision (mAP) at ILSVRC 2015 and the COCO 2015 competition.

3. The Tracking Algorithm

Camshift [5] algorithm is an extension of the meanshift [6] algorithm, based on color histogram. It can achieve the moving target tracking and adjust the size and position of the window in the simple background. However, in practice, camshift often fails in tracking when there are changes caused by illumination variation or partial occlusion, etc. A linear recursive Kalman filter using motion information including velocity and space direction to predict target position in the next frame.

In this paper, Camshift combined with Kalman filter is used for moving target tracking [7] [8]. Faster R-CNN is used for target detection. The detection results are used to initialize Kalman filter. The algorithm steps are as follows:

(1) To identify the target to be traced. For the first frame fetch the central coordinates of the target area (The coordinates of target areas coming from Faster R-CNN detection)., and then initialize the kalman filter with this coordinate;

(2) Based on the results of targets tracking in the first frame, use Kalman filter to predict the location in the current frame;

(3) With the coordinates acquired in the step(2) as the centre, set up the search window and calculate the back projection image, then use Camshift algorithm to do iteration, until it meets the conditions. Then the searched window coordinate is treated as the final coordinates, which are the real location of the tracking object in the current frame eventually;

(4) Correct kalman filter based on the coordinate from current frame, and set the current frame as first frame.

(5) If there is no new detection results from Faster R-CNN, making the program return to step (2), continue to track the target in the next frame. If the new detection results from Faster R-CNN are achieved, making the program return to step (2), continue to track the target in the next frame.

4. Experiment

4.1. Dataset

KITTI dataset is currently the largest computer vision dataset in the world, which is used to evaluate the performance of computer vision technology. Images were captured from real scenes such as urban, rural and highways. Figure 1 shows some images in KITTI dataset.

Figure 1. Some images from the KITTI dataset. Large variations in appearance and camera viewpoint, and severe occlusions are present in the dataset.

In this experiment, Subset from KITTI, including 7481 training images and 28742 labeled vehicle objects is used for model training. The car, van, truck, and tram are reclassified as part of the vehicle class. The test video was captured on the actual road using a usb camera on a car, with resolution 1280 * 720 and frame rate 30fts.

4.2. Training

A laptop with an Intel i7-6700 CPU, a GTX980M GPU with 8G memory, and operating system Ubuntu14.04 LTS is used for training and testing. The development language is python. Because of the limitation of computing power, a medium-sized VGG_CNN_M_1024 model is choosed for this experiment [9].

4.3. Detection and Tracking

The average computation time for target detection is 0.935s per image, and it is 0.0244s per image for tracking. Figure 2 and Figure 3 show the CDF (Cumulative Distribution Function) [10] of overhead time of detection and tracking. Figure 2 indicates that about 60% of the points fall within the range of [0.09, 0.11], while it's 75% of the points falling within the range of [0.003, 0.025] in Figure 3. The real-time vehicle detection and tracking system requires objects can be marked in the video frames without delay. Multi-thread technology can avoid blocking, while performing multiple tasks, maximizing multiprocessor performance. Multi-thread programming method is used, instead of the idea of "detection before tracking", to improve the real-time performance of the system and obtain good results.

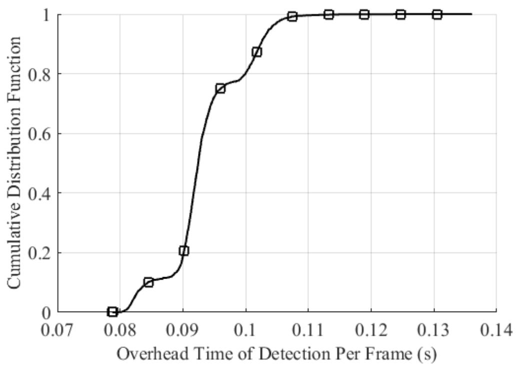

Figure 2. Overhead time of detection per frame.

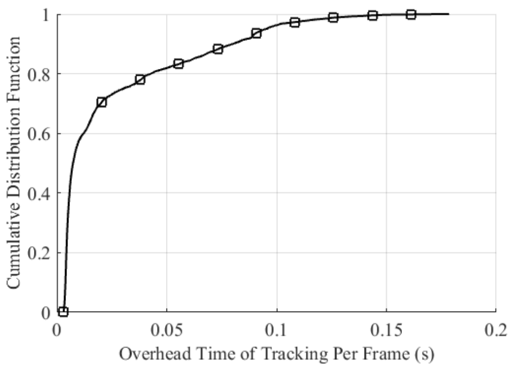

Figure 3. Overhead time of tracking per frame.

The flow chart of our algorithm for target detection and tracking is shown in Figure 4. Because the target detection computation is almost on GPU while the target tracking process uses CPU. We use two threads for real-time computation, detection thread and tracking thread, and define a global variable Flag to indicate whether new objects are detected or not. For the first frame, the detection thread detects targets using Faster R-CNN and the detected results are fed to tracking thread to start target tracking. The tracking thread uses new detection results for tracking if new detection results from detection thread are achieved. Two threads are running in parallel avoiding waiting for detection results.

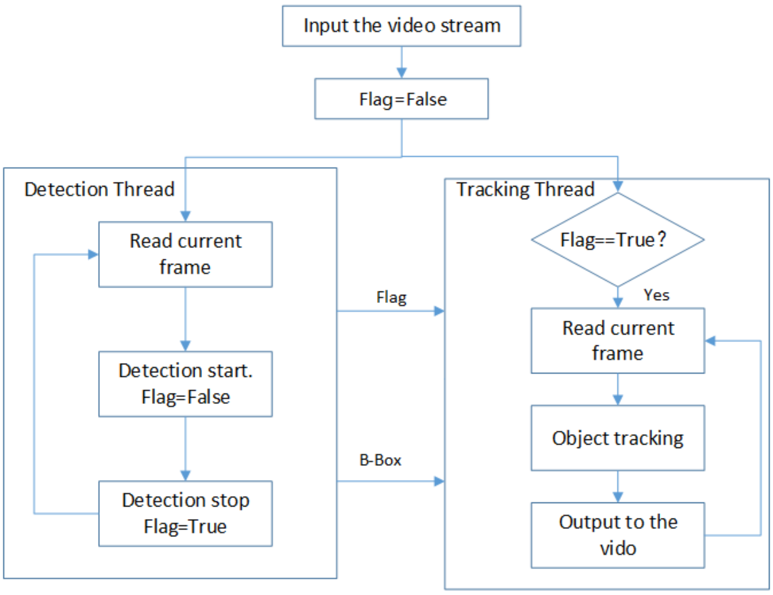

Figure 4. Flow chart of the multi-thread detection and tracking. Detection thread provides new object bounding-boxes and feed to the tracking thread.

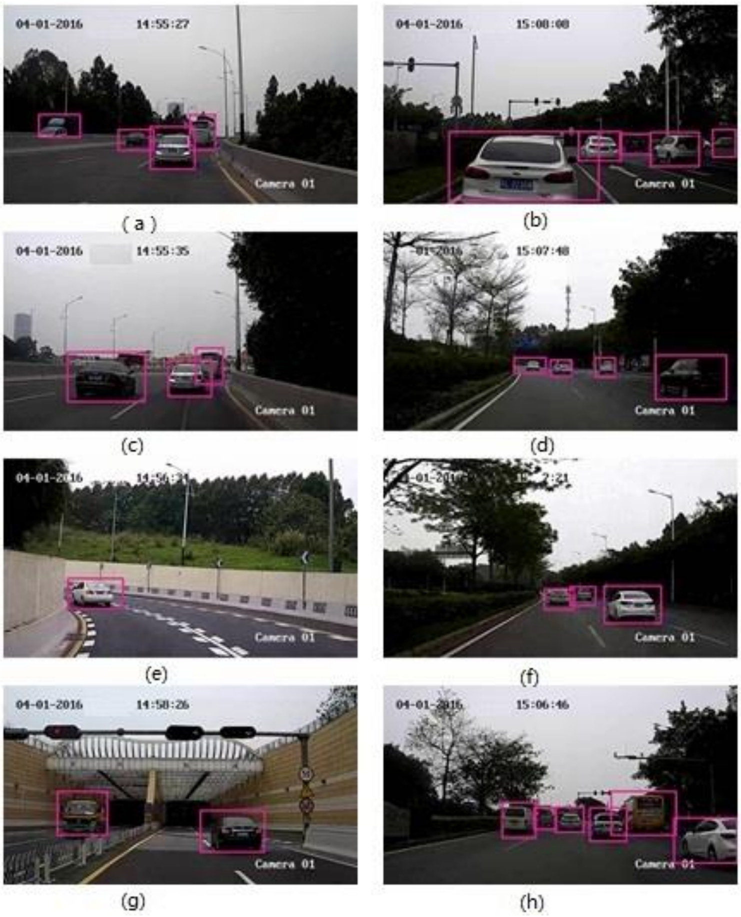

Figure 5. Samples of vehicle detection and tracking.

Figure 5 shows some results of vehicle detection and tracking. Faster R-CNN demonstrates a strong performance on vehicle detection. Although the illumination changed and occlusion occur, Camshift combined with Kalman filter algorithm overcomes the interference better and achieves good tracking effect.

5. Conclusion

In this article, Faster R-CNN method is adopted to conduct the training of deep learning on the KITTI sub-dataset, which is effective for object detection in complex background. The combination of Camshift and Kalman filter method is used for vehicle tracking, by passing on unknown space parameters to initial search process, is effective to overcome the interference of the background and improves the reliability of object tracking. This method can be used in real-time application.

6. **References**

[1] Ren S, He K, Girshick R, and Sun J 2015 Faster R-CNN: Towards real-time object detectionwith region proposal networks *In Advances in neural information processing systems* 91-99
[2] KITTI benchmark, http://www.cvlibs.net/datasets/kitti/eval_object.php
[3] Wang X and Li X 2010 The study of moving target tracking based on kalman-camshift in thevideo *IEEE Int. Conf. on Information Science and Engineering* 1-4
[4] Girshick R 2015 Fast r-cnn *Proc. Int. Conf. on computer vision* pp 1440-8
[5] Exner D, Bruns E, Kurz D, Grundhöfer A and Bimber O 2010 Fast and robust camshift tracking*Computer Vision and Pattern Recognition Workshops. IEEE*, 9-16
[6] Leichter I, Lindenbaum M and Rivlin E 2010 Mean shift tracking with multiple reference color histograms *Computer Vision and Image Understanding* **114(3)** 400-408

[7] Zhang C, Qiao Y, Fallon E and Xu C 2009 An improved camshift algorithm for target trackingin video surveillance *9th IT&T Conference* 1-10
[8] Huang S and Hong J 2011 Moving object tracking system based on camshift and Kalman filter*IEEE Int. Conf. on Consumer Electronics, Communications and Networks* 1423-6
[9] Lee C, Kim H J and Oh K W 2016 Comparison of faster R-CNN models for object detection *IEEE Int. Conf. on Control, Automation and Systems* 107-110
[10] Xue J, Titterington M The p-folded cumulative distribution function and the mean absolute deviation from the p-quantile *J. Statistics & Probability Letters*, 2011, 81(8):1179–1182.