Near Real-Time Vehicle Detection and Tracking in Highways

Ebrahim Soroush
Amirkabir University of Technology
Email: e.soroush@hotmail.com

Ali Mirzaei
Amirkabir University of Technology
Email: a.mirzaei69@gmail.com

Shiva Kamkar
Amirkabir University of Technology
Email: shv.kamkar@gmail.com

***Abstract*—In this paper we present an approach for detection and tracking of vehicles in highways. The Aggregated Channel Features (ACF) are used to detect vehicles and the Kalman filter is employed to track the detected objects. The proposed scheme enjoys high accuracy in both detection and tracking. Moreover, it can be run at near real-time speed on an ordinary computer (both detection and tracking take about 140ms for each frame). The proposed approach was the best algorithm in AUTCUP2015 competitions and it got the second rank in that competition (The first rank was not given to any team).**

I. Introduction

With daily increasing of number of vehicles and highway traffics, the Intelligent Traffic Systems (ITS) become more and more important than before. Obtaining the traffic parameters such as number of vehicles, average speed for any type of vehicles (light and heavy) and etc. can help the responsible organizations with better monitoring of the traffic. The vision-based systems are better than other existing methods (like laser guns for obtaining the speed or magnetic loops for counting and classification of vehicles) from several point of view: first, the cost of these kind of systems are far less than others. Secondly, the maintenance of cameras are much easier and thirdly all traffic parameters can be obtain with a single camera in a specific area.

In this paper we propose a scheme to detect and track vehicles in a video. Our proposed scheme works on a single camera which mounted on a high place and it is recording video from rear of vehicles. Moreover, it can be used in other views of camera if the detection model is trained based on that given view. The Aggregated Channel Features (ACF) is used for detection and a Kalman filter is employed to track the vehicles. The proposed approach can detect and track the vehicles in a near real-time speed (about 5 frame per second) on an ordinary computer[1]. The presented approach was the best method in AUTCUP2015 competitions in Amirkabir University of Technology and it ranked second in that competitions (no team ranked first).

The rest of this paper is organized as follow: Section II illustrates the existing algorithms. In section III the used detection algorithms (ACF) is described. In section IV the configuration of out kalman filter as a tracker is explained and section V concludes the paper.

II. Related Works

Detection and tracking of vehicles can be considered as a core of a vision-based intelligent system. In the following subsections the most well-known methods for both detection and tracking are reviewed and the reasons of selected methods are presented.

A. Detection

All detection algorithms can be classified into two categories: motion-based and appearance-based approaches. Motion-based methods [1],[2],[3] try to detect vehicles using their motions. However, these methods are fast and easy (they do not need to be trained), they are so vulnerable to shadows, luminance changes and shaking of the camera. In the other hand appearance-based approaches detect vehicles with a pre-trained model according to the intrinsic features of objects. Generally these kind of methods are more computationally demanding than motion-based approaches but they do not have the mentioned disadvantages of these methods.

As mentioned the appearance-based detections are more robust against luminance changed, camera shaking and shadows (problems that are common on highways). Because of these reasons we chose this family of methods for detection task.

One of the most well-known methods in object detection are part-based models such Deformable Part Model (DPM) [4]. Although DPM has a convincing performance for vehicle detection, it suffers from a high computations in test phase. The time consumption of DPM in test phase prevent us to have a real-time or even near real-time system for detection and tracking of vehicles. There are some methods which try to get fast DPM [] but all these algorithms reduce the performance or they are not real-time.

Recently the Convolutional Neural Networks (CNNs) presented very good results in object detection. These kind of methods are state-of-the-art algorithm in object detection.

TABLE I. Comparison of Detection Algorithms on Kitti Dataset

Criteria/Method	DPM	VGG	CNN-based	ACF
Accuracy				
Time				

B. Tracking

Generally there are two approaches to track objects. First, tracking by detection, in which objects are detected in each

[1] Intel i5-4460, 8GB RAM

frame and they are linked up in consecutive frames. Secondly, tracking by matching, that they tracked a predefined object in the next frames.

III. Detection: Aggregated Channels Features

In this section we provide a brief overview of ACF detection framework [5] and then demonstrate a very fast method for feature pyramids generation [6] in detection task.

A. Aggregated Channel Features (ACF)

First step includes computing several channels of given image I with $C = \Omega(I)$; one simple example of C could be the gray-level channel. In the second step, some blocks of pixels in the image is defined and then summed every block of pixels in C and smooth the resulting lower resolution channel feature. Next step would be the concatenating of all pixels in the aggregated channels. In the last step, a boosted trees of classifier is used to distinguish objects from background. Fig **??** is demonstrated this steps.

Main feature of ACF detector could be summarized as:

Channels: In this work same as [5] 10 channels of: LUV color channels (3 channels), normalized gradient magnitude(1 channel) and histogram of oriented gradients(6 channels) is used. After smoothing these channels with a $[1\ 2\ 1]/4$ filter, the channels divided into 4×4 blocks and pixels in each block are summed up. Finally the channels are smoothed again with the same filter.

Classifier: For classifying of cars and non-car detections, an AdaBoost [7] is used to train and combine 128 depth-two trees as weak classifiers. This classifier needs positive and negative samples of cars and non-car patches for training.

Sliding Window Detection: As traditional object detector, sliding window detector scheme employed for finding bounding boxes of car candidates. Features in multiple scales of these bounding boxes extracted and fed it to the boosted classifier. Extracting these features in multiple scales is very important in run-time efficiency and usually it is a bottleneck in detection. In the next section this subject will be discovered in detail.

B. Fast Feature Pyramids

One of the most challenging problems in object detection is: Different instances of one object could be appear in different sizes and pose. Therefor, it is very hard to have a classifier to detect all objects in an image. To tackle this problem detectors used sliding window method in a pyramid of images in multiple scales. In traditional methods we had to extract features in every scales that we called feature pyramids, it will will solve the different sizes problem by increasing the computational costs in number features and scales (up to multiple seconds for each frame).

For increasing the speed of the detector in building these feature pyramids, [6] propose a very simple and fast method for approximating features in different scales. For clarifying the subject, let $I_k(x, y) = I(x/k, y/k)$ is the up-sampled version of original image and suppose our feature is the gradient of the image: $h_k = \frac{\partial I_k}{\partial x}(x, y) = \frac{1}{k}\frac{\partial I}{\partial x}(x/k, y/k), h = \frac{\partial I_k}{\partial y}(x, y) = \frac{1}{k}\frac{\partial I}{\partial y}(x/k, y/k)$. With a simple calculus we can deduce that $h_k = k \times h$. Fig **??** showed an experiment with a dataset of natural images, upsampled using bilinear interpolation, this approximation shows $h_2 \approx 2 \times h$ for gradient histogram features. This experiments showed for 4280 images this approximation is unbiased and relatively small variance (0.061). For downsampled images because of lost information from original image this derivations is not true. Experiments showed this information lost is consistent and resulting approximation is biased but with small variance (0.059), fig **??** showed this discussions. [6] has more details about fast feature pyramids estimation.

IV. Tracking: Kalman Filter

We consider position (x, y) and size of bounding box (w, h) and their velocity as the state of kalman filter (constant velocity). So the state of the kalman filter will have 8 variable $(x, y, w, h, v_x, v_y, v_w, v_h)$. With this definition the dynamic model will be:

$$\begin{bmatrix} x \\ y \\ w \\ h \\ v_x \\ v_y \\ v_w \\ v_h \end{bmatrix}^{t+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \\ h \\ v_x \\ v_y \\ v_w \\ v_h \end{bmatrix}^{t} + n_d$$

,

where n_d is a vector and called model noise. The measurement model can be written as following:

$$\begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \\ h \\ v_x \\ v_y \\ v_w \\ v_h \end{bmatrix} + n_m$$

,

where n_m is a vector and is called the measurement noise.

V. Implementation and Performance Evaluation

In this section we demonstrate the performance of our method on AUTCUP dataset and explain some implementation details.

Training

We train ACF detector with open-source piotr-toolbox [8]. As explained in **??** for training, we need positive and negative samples from train dataset. For convenience we used a pre-trained DPM model on PASCAL VOC2012 on the train videos for extracting car patches. For training we extract 1657 patches of cars and 19188 non-car patches all resized in $[64\ 64]$ of from train videos. Training phase including reading all images, extracting feature pyramids with aforementioned method and training boosted classifiers takes 17 seconds.

AUTCUP Dataset

This competition includes 12 video of non-stable cameras with different views, fig **??** shows some sample of these videos. For training we captured 1000 random frames from each video and extracted car and non-car patches from these frames. Validation set of these competition was first 30 seconds of each videos and results showed in table **??**. For evaluating the method on this competition in the test set, all teams have to submit their code on an evaluation server, table **??** shows result of this competition.
Our method for detecting and tracking of all cars in video works in near real-time, about 145 ms for each frame.

VI. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank Mr. Agahi, member of technical committee of AUTCUP2015, for his help on evaluating of the method.

References

[1] S. C. Sen-Ching and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 881–892.

[2] N. Sirikuntamat, S. Satoh, and T. H. Chalidabhongse, "Vehicle tracking in low hue contrast based on camshift and background subtraction," in *Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on*. IEEE, 2015, pp. 58–62.

[3] X. Lu, T. Izumi, T. Takahashi, and L. Wang, "Moving vehicle detection based on fuzzy background subtraction," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 529–532.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[5] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west." in *BMVC*, vol. 2, no. 3. Citeseer, 2010, p. 7.

[6] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1532–1545, 2014.

[7] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[8] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.