Video Surveillance for Road Traffic Monitoring

Arnau Casadevall, Toni Perez, Adrian Salvador
{arnau.casadevall, antonijoan.perez, adrian.salvadorr}@e-campus.uab.cat *

Universitat Politecnica de Catalunya (UPC)
Master in Computer Vision

April, 2018

ABSTRACT

Nowadays, traffic surveillance applications are applied to improve the safety on the roads helping to reduce the accidents and control the traffic flow. In this paper, it is presented our Road Traffic Monitoring system based on Computer Vision techniques such as background subtraction, foreground segmentation, morphological operations and region tracking to track the cars and estimate their speed. With that, a cheap radar speed detector it has been implement with a fast computational time.

Index terms— Video traffic surveillance, road traffic monitoring, car tracking, speed estimation

1 Introduction

Surveillance applications monitors the behavior, activities, or other changing information and they are used for the purpose of influencing, managing, directing, or protecting people. Video traffic surveillance is important to control several actions on roads, for example, car speed and traffic flow. The main techniques of video processing will be applied in the context of video surveillance: moving object segmentation, motion estimation and compensation and video object tracking are basic components of many video processing systems.

Firstly, we will explain about the context of the video surveillance for road traffic monitoring in the last few years in the Section 2. Then, in the Section 3, the datasets we have used in this work. In the Section 4, we will present the techniques used to finally track the cars and estimate their speed and, afterwards, we will depicted our own results in Section 5.

Finally, in Section 6 we will end with future improvements and the conclusions.

2 Related Work

In the last years the monitoring systems in the road scene are increasing due to the growing demand and great benefits that can afford. Specially now, where the autonomous driving cars are becoming a reality.
A real-time traffic monitoring using virtual line analysis for determine the traffic volume automatically [1] is one of the examples as a contribution that has made along these years. All the information about the traffic in real-time can assists drivers to dynamically plan their trips more efficiently. More related in the speed estimation, we can found work about how to estimate a mean traffic speed using uncalibrated cameras [2].

Nowadays, and taking into account the also increasing interest in deep learning methods, several recent neural networks have achieved good results in traffic monitoring systems by applying deep learning. An estimation from a UAV (Unnamed Aerial Vehicle) video based on ensemble classifier and optical flow is a recent work that can monitor a real-time traffic networks from an aerial perspective using, mostly, a Haar Cascade + CNN [3].

3 Datasets

We have focused on two video sequences from outdoor scenarios, Highway and Traffic. [4]

- Highway sequence: category baseline. Simple video with 1700 frames. Example in Figure 1

- Traffic sequence: category camera jitter. video with heavy camera jitter with 1570 frames

*Github's source code: https://github.com/mcv-m6-video/mcv-m6-2018-team8

Each sequence contains a large number of input and ground truth images where ground truth images have five different labels: 0 (static), 50 (hard shadow), 85 (outside region of interest), 170 (unknown motion) and 255 (motion).

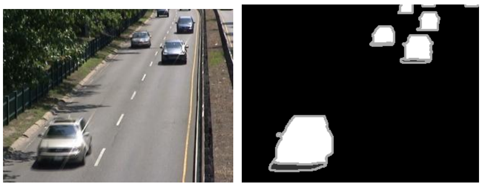

Figure 1: Input and ground truth image of Highway sequence

4 Car tracking pipeline

In the next Figure 2, we can see the pipeline of our final system. Before to achieve this pipeline we can experiment with several approaches trying to obtain the best performance in the background subtraction, specially.

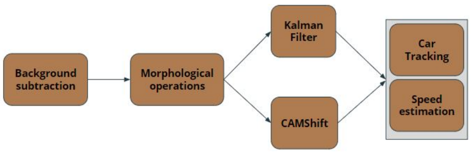

Figure 2: Pipeline of the system

4.1 Background subtraction

The idea is to separate the cars of the road, where the road is the static scene (background) and the cars the dynamic scene (the foreground). For this work we apply and adaptive model based on a Gaussian Mixture Model (GMM) [5], which depends basically on two parameters: α and ρ.

First, we compute the mean and the standard deviation of each pixel of the first 50% frames for each dataset: Highway and Traffic. From both values, a Gaussian is modeled for each pixel.

The second half of the dataset is used to segment the foreground. If the pixel does not fall in the Gaussian, it is considered as a foreground, otherwise, it is a background pixel.

In both datasets the most of amount of variation is found on the roads because of the movement of the dynamic cars. However, in the case of Traffic dataset, there is a high standard deviation due to the jitter movement of the camera.

Focusing on the GMM algorithm [5], when ρ is too low, μ and σ do not change when a foreground is detected (no weight for the previous frames), where the recall is too high but the precision decays drastically. In other hand, for a large values of ρ, the model takes into account the values of the previous frame too much and the results are not quite good.

Said that, it is expected that the best configuration be with a balanced numbers between α and ρ.

4.2 Morphological operations

Once the first background subtraction is done, it still keeps some noise observed, generally, as small dots. Therefore, it is a good strategy to apply some morphological operators to try to filter the noise, which can be a problem in further steps when we want to detect the dynamic objects, such the cars.

Erode the image to remove this dots and then dilate the image to recover the shape of the cars can be a good choice [6] [7]. Several tests have been done with different morphological operations and with different structure elements of different sizes. The best results have been obtained using opening operation, defined by the dilation of the erosion: removes small noise in the background and then dilate the image in order to increase the shrunken foreground. After that, an area filtering followed by a filling hole algorithm is applied to refine the background and obtain a better mask.

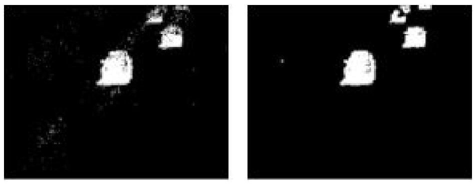

Figure 3: Morphology output

4.3 Tracking and speed estimation

In this part of the project the cars will be located in successive frames of the videos and its velocity will be estimated.

To do this tracking part two algorithms are used, both adapted from the OpenCV library [8]:

- Kalman Filter
- CAMShift

4.3.1 Kalman Filter

The Kalman filter has many uses, including applications in control, navigation, Computer Vision, and time series econometrics [9].

In this case, Kalman filter is used to track the cars on the roads. Kalman filter takes the current state of the system, and makes a prediction based on the current state and current uncertainty of the measurements, and make a prediction for the next state of the system with an uncertainty. Then, it compares its prediction with the received input and correct it self upon the error.

The main advantage of the Kalman Filter is its reliability compared with the other methods because it takes into account some noise into variables. Kalman filter is used to predict the location of a moving object based on prior motion information. Once a new detection is done, update this prediction with a real measurement.

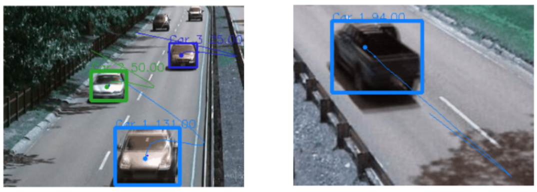

Figure 4: Tracking with Kalman Filter

4.3.2 CAMShift

With CAMShift (Continuously Adaptive Meanshift) we can adapt the window size and the rotation of the target.

The first step is to get a confidence map in the new image based on the color histogram (using HSV) of the object in the previous image: histogram backprojection [10]. Then, apply meanshift from OpenCV [8] and find the peak of the confidence map near the object you want o track.

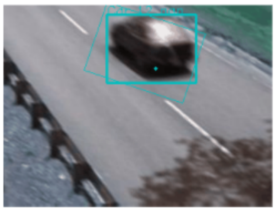

Figure 5: Tracking with CAMShift

The process is iterated up to the meanshift converges. It calculates the orientation of best fitting ellipse to it and then applies the meanshift again with new search window size. The idea is an adaptation from here [11].

CAMShift can obtain the next position with less parameters in comparison with Kalman Filter, but as it can be observed in the Figure 6 is easy to obtain a miss-prediction or conflicts in the tracking ID selection because of the similarity between the histograms of different cars.

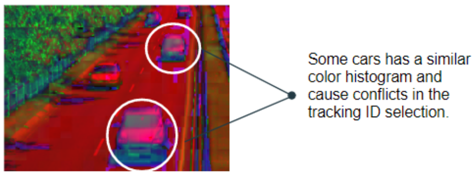

Figure 6: Frame in HSV

4.3.3 Speed estimation

The main idea is to calculate the distance between the old measurement and the current measurement as in the next equation 1. Generally, in the images we have the cars driving vertically, therefore we put more weight in the y-direction than in the x-direction:

$$D(x, y) = |\log(\sqrt{x^2 + y^2})| \tag{1}$$

Then, we approximate the speed with a *speed estimator* factor chosen empirically that multiplies the distance above. In addition, we do a mean between the last speed measured and the current one to avoid some outliers and have a kind of average value without abrupt variations.

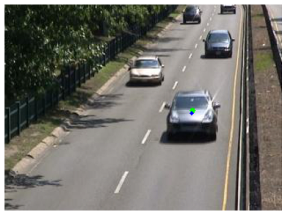

Figure 7: Speed estimation measurements. The green point represents the old measure, whereas the blue one is the new measure.

5 Evaluation: Our test

We have recorded our own video in Av. de Serragalliners (Cerdanyola del Vallès, UAB) from the bridge. In order to make an algorithm more stable, we have cropped the video avoiding outside noise.

We have carried out the same pipeline mentioned before in Section 4 applying a single Gaussian model: the mean μ and standard deviation σ model are obtained with frames where the road was almost empty. After that, we have applied car tracking with both Kalman Filter and CAMShift and we have estimate the speed using the same method used in Section 4.3.3. A qualitative demo is depicted in the Figure 8 and remark that works for both directions of the road.

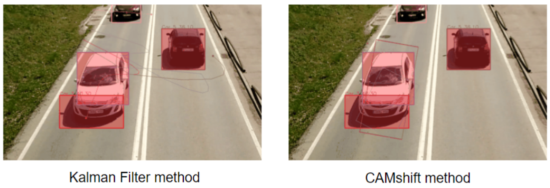

Figure 8: Left to Right: Radar speed detector with a Kalman Filter and CAMshift car tracking. A red filled rectangle means that the car is driving faster than the limit speed (30km/h in that case).

6 Conclusions and Future work

The system has been tested in three different datasets: Traffic, Highway and our own dataset. It is able to operate in real time (once the background model is obtained), track the cars with individual IDs and estimate the speed of them.

The paper shows a complete a road traffic monitoring system with a full pipeline robust to a dynamic background. Nevertheless, as a future work is desirable to improve some simple techniques used to increase the performance of the method. For example, in background subtraction it has been used an adaptive Gaussian, that works pretty well, but using a more complex algorithm like a Deep Convolutional Network could be increased the performance of the system.

An improved stabilization technique could also be applied to the system. We are not able to improve the results with the actual stabilization method and it can be appreciable especially in the Traffic sequence due to camera jitter. This improvement could increase considerably the quality of the foreground extraction. In addition, another interesting approach is to apply a shadow removal. We have not included it in the pipeline because in our case lots of TP had been removed and the performance dropped drastically.

Finally, apply Optical Flow for car tracking as well for speed detector could be another measure to take into account. Improving the car tracking system applying a Convolutional Neural Networks (CNN) as Real-Time Vehicle Detection and Tracking Using Deep Neural Networks [12]. The network architecture is composed by Convolutional layers (+9), Inception modules (+4), a Spatial Pyramid Pooling layer (+1) and Fully Connected layers (+2).

References

[1] B. L. Tseng, Ching-Yung Lin, and J. R. Smith. Real-time video surveillance for traffic monitoring using virtual line analysis. In *Proceedings. IEEE International Conference on Multimedia and Expo*, 2002.

[2] D. J. Dailey, F. W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, 2000.

[3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 2015.

[4] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[5] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999.

[6] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In David Vernon, editor, *Computer Vision — ECCV 2000*, 2000.

[7] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *CVPR 2011*, 2011.

[8] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[9] A. Kamann, J. B. Bielmeier, S. Hasirlioglu, U. T. Schwarz, and T. Brandmeier. Object tracking based on an extended kalman filter in high dynamic driving situations. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[10] Michael J. Swain and Dana H. Ballard. Indexing via color histograms. In Arun K. Sood and Harry Wechsler, editors, *Active Perception and Robot Vision*, 1992.

[11] John G. Allen, Richard Y. D. Xu, and Jesse S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing*, 2004.

[12] X. F. Gu, Z. W. Chen, T. S. Ma, F. Li, and L. Yan. Real-time vehicle detection and tracking using deep neural networks. In *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2016.