BEYOND A SMART VIDEO SURVEILLANCE APPROACH FOR ROAD TRAFFIC MONITORING

Mónica Alfaro, Andrea Calafell, Martín Matilla, Jordina Torrents

Universitat Autònoma de Barcelona

ABSTRACT

This paper tries to solve the problem of road traffic monitoring. The main goal is to count the amount of vehicles on a road and estimate their speed using visual cues. Our solution presents an optimized approach of the typical adaptive Gaussian modelling, in order to subtract the background. Then, morphological filters and shadow removal are applied to refine the solution. Stabilization methods have been studied to remove jitter effect caused by the wind. Finally, a Kalman filter is used to track each vehicle appearing in the sequence, and a homography matrix is applied in the sequences to obtain the correct perspective for computing the speed.

***Index Terms*—** traffic, vehicles, tracking, speed, surveillance, stabilization

1. MOTIVATION

With thousands of road kilometers, traffic monitoring has become very important to increase security in roads. As speeding is one of the main causes of car accidents, being able to automatically track the speed of vehicles can be critical to save lives.

2. RELATED WORK

Video road sequences scenes are increasingly used in several contexts with an emphasis on automation to estimate vehicular speed, and notably for tracking moving objects in a static background. Therefore, several well-known researchers have proposed a wide-ranging state-of-the-art.

In [1] the authors have proposed a method of vehicular speed estimation based on spherical projection. Moreover, researchers in [2] have been carried out speed estimation using optical flow for a side view. Although those algorithms work with un-calibrated cameras, they are limited to vehicles travelling in 1D and the camera must be placed in-line with the vehicle motion. A part from that, the aforementioned methods feature a weak background subtraction technique that yields poor results in the presence of noise. Moreover, in [3] only average speeds are estimated restricted to 1D motion. Authors in [4] also develop a speed estimation method suppressing camera vibrations using background compensation. However, due to the Interframe difference technique, the error rate in low speed estimations are high.

3. METHODOLOGY

3.1. Background Estimation

The background estimation is done by modelling it using random Gaussian variables. However these Gaussian variables can be used in different approaches: using a non adaptive model, using an adaptive model to adapt the changes in the background, and using several Gaussian to model each pixel.

3.1.1. Non adaptive model

In this approach we model each pixel as a single Gaussian function, and we estimate the mean and the variance of each one using the training dataset. Then, for each image of the testing set we classify as foreground the pixels that satisfy the following condition:

$$|I_p - \mu_p| \geq \alpha(\sigma_p + 2) \tag{1}$$

Where I_p is the value of the pixel in the current image, μ_p and σ_p are the mean and the standard deviation of this pixel, and α is the threshold used to classify the pixel.

3.1.2. Adaptive model

In this case, we also estimate the initial values of the mean and the variance of each pixel using the training set. However, during the classification part, these values are updated in each frame, in order to adapt them to the different time instances of the background scene, or to new objects that could remain in the scene for a long period. Thus, the pixels are classified using the same approach explained in the non adaptive model 3.1.1. Then, the pixels classified as background are used to update the mean and the variance following these equations:

$$\mu_p(t) = (1-\rho)\mu_p(t-1) + \rho I_p(t) \tag{2}$$

$$\sigma^2(t) = (1-\rho)\sigma^2(t-1) + \rho(I_p(t)\mu_p(t))^2 \tag{3}$$

Where I_p is the value of the pixel in the current image, μ_p and σ_p^2 are the mean and the variance of this pixel, and ρ is the

memory of the system, it means that is the value to establish how quickly the system is updated.

In order to obtain the best parameters for both ρ and α, we have developed a method to optimize both parameters together as we can see in Figure 1. Thus, our system starts estimating the best α in a non adaptive approach. Then, iteratively, ρ is estimated given the best α at current time instant and then, again, α is estimated given the best ρ at current time instant. Finally, the system stops when it converge to an optimal solution.

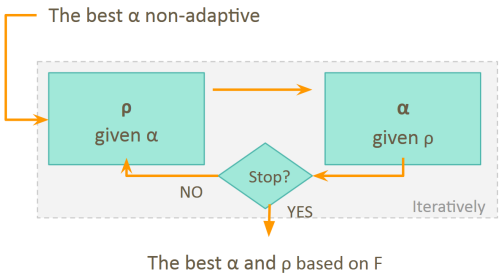


Fig. 1: Scheme of the adaptive approach

3.1.3. Gaussian Mixture Model

This approach uses multiples Gaussians to model both the background and the foreground in order to take into account the variability of the background due to switch between various states that it could have. Each Gaussian represents either a foreground pixel or some time instances of the background. Thus, Gaussians with high weight and low variances are considered as background pixels. By contrast, Gaussians with high variance and low weight are considered as foreground pixels.

In this case we have also develop a method, as we can see in Figure 2, to optimize the number of Gaussians that we have to use, the learning rate and the minimum background ratio, which represents the threshold to determine which Gaussians are considered as background.

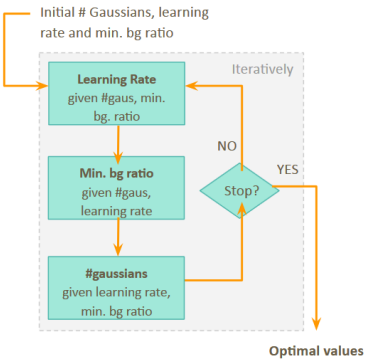


Fig. 2: Scheme of the gmm approach

3.2. Morphological Filtering

The background estimation step may include undesirable noise. In order to clean the mask some morphological operation have been made. First, our system applies a reconstruction by erosion to remove noise while preserving the shape of the objects that remains. Then, a closing is performed to enclose objects. Finally, a filling holes approach is used. However, when the objects were still not bounded, the system finds the convex hull of the blobs in the mask instead of doing a filling holes. Moreover, different structural elements have been used depending on each sequence.

3.3. Stabilization

In traffic monitoring, some sequences can not be perfectly stabilised due to jitter effect produced by the wind. This effect can lead to errors in object tracking. Due to the shaking, those pixels that should not move, appear to be moving uniformly from one frame to another. Also, those pixels that represents moving objects in the scene will have different motion vectors. Therefore, the motion vector for any pixel is the summation of the camera movement and the object movement. In order to compensate the sequence is necessary to separate the camera motion from the action within the scene. We studied different ways of estimating the motion.

On one hand, a block matching approach using backward compensation was used. Briefly, in the backward compensation, the current image is divided in macroblocks and the block matching algorithm tries to find similar blocks in a reference image within a search area. We use Mean Square Error to find the best matches. This approach makes two assumptions: (1) pixels belonging to the same block moves in the same way; (2) the motion of a block is determined by the search window.

In an exhaustive search, the cost function is computed at each possible location in the search window. The obtained results is the best possible match but the computational cost is expensive. The 3 steps algorithm is a suboptimal solution that reduce the computational cost. As we can see in Figure 3 It is an iterative algorithm that only computes the cost function in 8 points around the search window and in the center. From these 9 points, it selects the point with least cost and makes it the new search center. The step size is reduced by half at each iteration.

On the other hand, the Lucas-Kanade technique is an exhaustive algorithm that calculates the motion vector for every pixel in the image assuming slow motion. Since, the sequences in traffic monitoring have large motion, this assumption is not guaranteed. Thus, a Hierarchical Lucas-Kanade is used. The low levels of the hierarchy detects slow motion whereas the high levels detects large and global motion.

Once the motion vectors has been estimated, a translation model is fit to all the motion vectors and the values t_x and t_y which best fits the data are found. Finally, this values are used

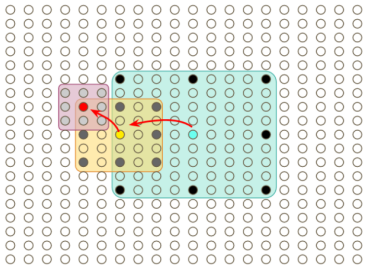


Fig. 3: 3-step approach

to reverse the shaking effect.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

3.4. Shadow Removal

In the previous sections we are detecting vehicles using those pixels whose color change more that our expected mean value for the background. However these methods presents a problem, as we are detecting as part of the vehicles their own shadows.

In order to detect properly the vehicles in the road, we have implemented the algorithm proposed in [5]. It consists on using the HSV colorspace and create a submask only for those pixels already selected as foreground in the previous steps. The pixels in this submask are considered as shadows if the following conditions are satisfied in each channel, otherwise their are considered as truly foreground:

$$\beta_1 < (I_v/\mu_v) < \beta_2 \tag{5}$$

$$|I_s - \mu_s| < \tau_s \tag{6}$$

$$(I_h - \mu_h) < \tau_h \tag{7}$$

Where I_v, I_s and I_h corresponds to the value of the pixel in each channel.

3.5. Vehicle Tracker

In order to finally track the vehicles appearing in the sequence we use the Kalman filter algorithm using the masks obtained in the previous steps. Since we are monitoring traffic, we defined a ROI on the road and all processing is perform only on the ROI. This algorithm processes measurement to deduce the optimal estimate of the state of a linear system using a set of measurements and a statistical model of the system, as we can see in Figure 4. It means that, this algorithm estimate in a first step the centroid of each detected vehicle using measurements observed over time, and in a second step it updates the predicted centroid using the real observed measures.

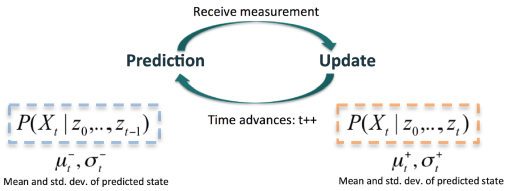


Fig. 4: Scheme of the kalman filter

3.6. Speed Estimation

The speed estimation can be calculated using the centroids obtained in 3.5 as the number of pixels travelled by the object during a certain number of frames scaled by the frame rate and pixels to meters ratio.

$$speed(km/h) = 3.6 \times \frac{frame\,rate}{num.\,frames} \times \frac{distance}{px\,to\,m} \tag{8}$$

For accurate speed estimation, the centroids obtained from the camera view in 3.5 have to convert to the top view. This is done using the concept of planar homography. Some assumptions, which are based in the current road traffic regularization, have been made such as the width and height of the road. In this way, we also calculated the number of pixels representing one meter in the image.

4. EVALUATION

In order to evaluate all the methods, we have used four dataset.

- *Highway:* this is the baseline dataset. The frame range used is 1050- 1350.
- *Fall:* this dataset presents the dynamic background problem. The frame range used is 1460-1560.
- *Traffic:* this dataset presents the camera jitter problem. the frame range used is 950-1050.
- *Ours:* this is the dataset that we have recorded in order to test all the final approach.

4.1. Results

Our final system uses an adaptive approach for the background estimation explained in 3.1. This approach has shown its efficacy in challenging sequences like Traffic. Although, we expected to obtain better results using Gaussian Mixture Model in sequences where the pixels of the background can take different values. However, it seems that it is not the case, as we can see in 1. Furthermore, we have also tested with

Sequence	Non Adaptive	Adaptive	GMM
Highway	0.4543	0.7496	0.7864
Fall	0.6707	0.7595	0.6524
Traffic	0.4843	0.6672	0.5804

Table 1: F-measure for each approach in each sequence

color images in YUV and RGB colorspace but the results obtained did not improve the results.

The morphological filtering described in 3.2 improves the results in 3 as we expected.

Sequence	AUC	Total Gain
Highway	0.8163	+0.1751
Fall	0.8856	0.1581
Traffic	0.8491	0.2703

Table 2: AUC obtained before use morphological filters and the gain with respect to the previous step

The shadow removal approach presented in 3.4 We have not used shadow removal technique in the test sequences because we observed that the results were slightly worse after applying it. It is caused because even if the thresholds have been chosen to detect correctly the shadows, there are some errors when the color of the cars are similar to the background (dark colors). The shadows are more difficult to detect in the Traffic sequence, probably due to the color of the cars. However, this technique shows better results in our sequence. This is because the shadows in this sequence are smoother.

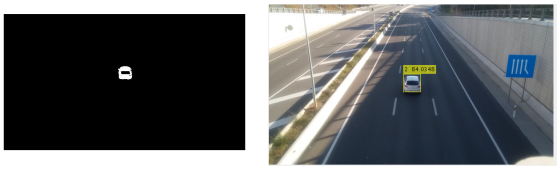
Sequence	With shadows	Without shadows
Highway	0.7496	0.7313
Fall	0.7595	0.7467
Traffic	0.6672	0.5570

Table 3: F-measure obtained before and after removing shadows

In the Highway sequence 20 cars are detected but only appears 11 cars. In the Traffic sequence 18 cars of 17 are detected. After adjust all the parameters of the tracker system, we observed that in some cases, the blobs belonging to the same object have different track paths due to they are recognized as different objects. Also, two different objects represented as separate blobs are viewed as a single object by the tracker, so it assigns only one track path.

We estimate a range of speeds between 80-100 kmph for Highway and Traffic sequences. On the other hand, we estimate that the speed of the car of our sequence was 84kmph when the real speed was 90 kmph.

(a) Mask after background esti- (b) Mask after shadow removal
mation



(c) Mask after morphological fil-(d) Final car detection with speed
ters

Fig. 5: Final results step by step in our sequence

5. CONCLUSIONS

We can conclude that, in order to perform a good vehicle tracker, the system should take into account several problems and solve them. For instance, the camera has to be well oriented through the road, it also has to be stabilized, the shadows and the sun reflex have to avoided.

In our approach we have perform a system that can deal with some of these problems, as it can stabilized the camera but it can not remove the shadows if they are as dark as the cars, and we can not compute correctly the speed of the vehicles if the camera is well oriented.

All the information presented in this paper, as well as information of their authors and the code can be found in this page

6. REFERENCES

[1] Vamsi Krishna Madasu and Madasu Hanmandlu, "Estimation of vehicle speed by motion tracking on image sequences," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 185–190.

[2] Sedat Doğan, Mahir Serhan Temiz, and Sıtkı Külür, "Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera," *Sensors*, vol. 10, no. 5, pp. 4805–4824, 2010.

[3] Daniel J Dailey, Fritz W Cathey, and Suree Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 98–107, 2000.

[4] Thuy Tuong Nguyen, Xuan Dai Pham, Ji Ho Song, Seunghun Jin, Dongkyun Kim, and Jae Wook Jeon, "Compensating background for noise due to camera vibration

in uncalibrated-camera-based vehicle speed measurement system," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 1, pp. 30–43, 2011.

[5] R Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with hsv color information," *Intelligent Transportation Systems*, 2011.