VIDEO SURVEILLANCE FOR ROAD TRAFFIC MONITORING

Adrià Ciurana, Guim Perarnau and Pau Riba

Universitat Politècnica de Catalunya
Master in Computer Vision, Computer Science

ABSTRACT

This article presents an algorithm for video surveillance devoted to road traffic monitoring. The proposed approach starts from the raw video of the road and ends with an estimation of the velocity for each car appearing in the images. Here, we present a detailed analysis for every step of the proposed solution and an evaluation for the whole pipeline. Moreover, we will face different problems such as camera jittering or dynamic background.

***Index Terms*—** Video Analysis, Video Surveillance, Background Estimation, Optical Flow, Kalman Filter

1. INTRODUCTION AND MOTIVATION

Nowadays, video surveillance is a hot topic in the field of computer vision. Using this kind of techniques allows not only to get lots of visual data but to analyze it automatically. In this work we have faced the problem of road traffic monitoring. Lots of other applications have been proposed for video surveillance in other scenarios (such as [1]), but as Computer Vision Master students, we wanted to apply all the theoretical knowledge learned during this course to make our own system. In our case we have chosen a straightforward approach that is based on strong assumptions and constraints (explained in Section 2) so as to simplify the implementation of the video surveillance.

The rest of this paper is organized as follows. First, Section 2 describes all the steps done for creating our system. In Section 3 we show the obtained results. Finally Section 4 draws the conclusions and future work.

2. METHODOLOGY

The proposed approach is constructed using different modules that we will explain in detail. Firstly, the foreground that contains the objects that we want to monitor must be segmented. Afterwards, we can detect the objects – cars in our case – and track them. From the tracking – and given a real measure of the distance introduced by the user – we can estimate the real velocity of the vehicles. The assumptions made for our applications are the following:

- The user has to mark the lines of the road in order to calculate the homography based on the vanishing point. Also, the user has to indicate the relationship between a pixel and the real distance. That is, for example, how many pixels equals a meter (used in Subsection 2.5).
- It is assumed that there won't be any occlusions in the sequences. If there are, our algorithm could track inaccurately.

2.1. Background estimation

Estimating the background is of key importance to the proposed technique. Three main approaches have been tested: two statistical models based on one Gaussian per pixel (Subsection 2.1.1) and a third based on a mixture of Gaussians (Subsection 2.1.2). Then, color information has been added to improve the previous estimation. Thus, until Section 2.1.3 we will be talking about gray-scale images.

2.1.1. Single Gaussian per pixel

The previously mentioned statistical models share the same idea, which consists of one Gaussian function to model each background pixel. From the first frames of our sequences we can estimate the Gaussian parameters μ and σ for each pixel. For every frame, we define a pixel I_i as foreground if the following condition is fulfilled:

$$|I_i - \mu_i| \geq \alpha(\sigma_i + 2) \text{ for all pixel } i \qquad (1)$$

At this point, this model can be divided between a non adaptive or adaptive approach.

- *Non Adaptive*: μ and σ are static – they do not adapt. They are calculated only once during a training process. This approach has some drawbacks.
- *Adaptive or recursive*: the Gaussians are adapted using the segmented background in the previous frame. Therefore, we will adapt to slow changes in the background, such as illumination. This technique uses a parameter ρ to control the adaptation speed. Then, the

adaption is performed for all $\text{pixel}\, i \in \text{Background}$ following the equations below:

$$\begin{aligned} \mu_i &= \rho I_i + (1-\rho)\mu_i \\ \sigma_i &= \rho(I_i - \mu_i)^2 + (1-\rho)\sigma_i^2 \end{aligned} \tag{2}$$

2.1.2. *Stauffer and Grimson*

Stauffer and Grimson [2] have proposed a technique for background estimation that uses a Gaussian Mixture Model (GMM) to model foreground objects. As the single Gaussian method, S&G is based on the temporal observation of the pixels. Each pixel is assigned to its nearest Gaussian. If the Gaussian models foreground (background), the pixel will be detected as foreground (background). It is necessary to manually set the number of Gaussians used (normally between 3 and 6) to adapt to data. The authors claim that this technique reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes (for ex. if a car parks and remains still will eventually detected as background).

2.1.3. *Color space*

Originally, we only used gray-scale images. However, the original input images are in color, meaning that if we convert them to gray-scale, we are losing information that we could benefit from. Thus, we have tested three different color spaces: RGB, YUV and CIE-Lab [3]. The first one is the most common color space but it has some drawbacks that other spaces try to correct. For instance, YUV space is focused on taking human perception into account whereas CIE-Lab aims to separate luminescence from the chromatic channels.

2.2. Foreground segmentation

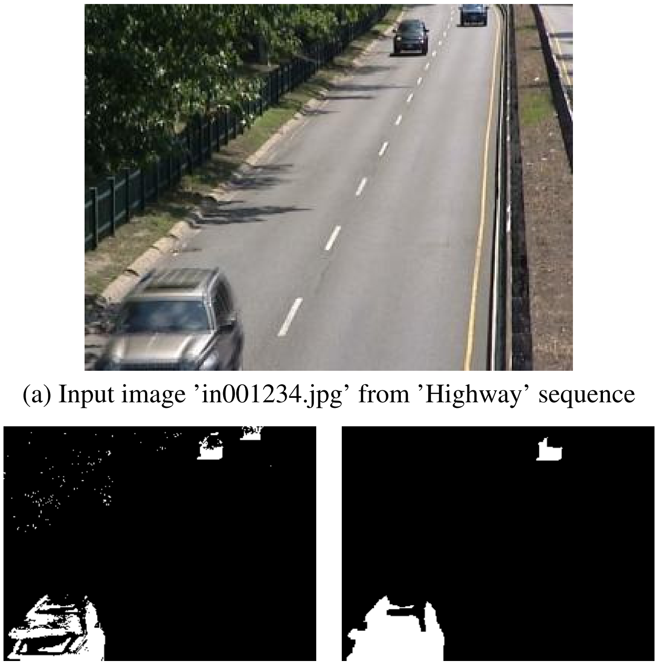
Once the background has been estimated, we want to avoid the parts that are miss-classified as foreground due to noise (camera jittering produced by the wind) or dynamic background (rustling leaves). The correction proposed in this Section differs from Section 2.1 because it uses neighbor pixels information, whereas the background estimation uses locally per-pixel information in the time domain. We propose to apply mathematical morphology to correctly eliminate miss-classifications using three operations:

1. *Area filtering*: in order to delete noise from the background produced by camera movements, wind, etc. However, it could mistakenly delete foreground blobs.

2. *Closing*: in order to join blobs of the same object that could be separated.

3. *Hole filling*: fill holes of foreground objects in the segmentation.

Figures 1 shows the pipeline for a frame from the input image to the foreground segmentation. Figure 1(a) represents one frame of a video sequence from which μ and σ for background estimation are known. Thus, the background can be estimated giving the mask showed in Figure 1(b). Afterwards, the mask is refined performing a foreground estimation by means of morphological filtering (Figure 1(c).


(a) Input image ’in001234.jpg’ from ’Highway’ sequence

(b) Background estimation

(c) Foreground segmentation

Fig. 1. Example of the background estimation (section 2.1) and foreground segmentation (Section 2.2) for an input image of the ’Highway’ sequence (database info. in Section 3.1).

2.3. Stabilization

An estimation of the optical flow of the sequence using either block matching or Lucas-Kanade [4] can be applied in order to perform an image stabilization so as to avoid the miss-segmentation and noise produced by camera jittering. For the block matching algorithm, the block division size is a very important parameter. There can happen that pixels that are moving in different directions fall in the same block, making the estimated flow to be inaccurate. On the other hand, L-K does not divide the image by blocks but allows each pixel to have his own direction using the Taylor approximation. However, the performance of the background segmentation falls and it has been discarded.

2.4. Tracking

For tracking purposes, two main approaches have been tested. First an algorithm based on Kalman filter [5] is used.

This algorithm is optimal for linear dynamical models under the assumption of Gaussian noise. Then another technique based on a stack denoising autoencoder for representing features and neural networks for classification [6] has also been tested.

Both techniques have been implemented as the tracking of a bigger system of lives. There, each blob that is tracked by one of these algorithms have their own life-cycle. For instance, a blob that do not move will be classified as inactive and will lose lives for each new frame. Afterwards, if this object loses all his lives, it is deleted. Therefore, the proposed system avoids the background noise or miss classified blobs due to a camera movement in one frame.

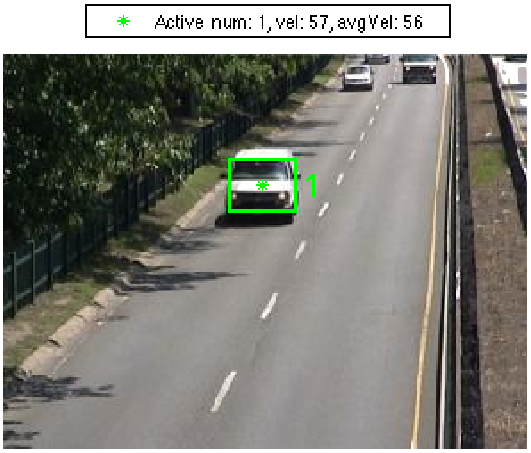


Fig. 2. Car tracked in a frame from the 'Highway' sequence.

2.5. Velocity estimation

Firstly, let us remark that in non orthogonal point of view images the objects will move in different velocities depending on their distance from the camera, no matter if their real velocity is constant. To solve that problem, an homography matrix can be defined. The proposed approach asks the user for two lines that are parallel in the real world and computes their vanishing point. Once the homography is computed, the image can be projected in a bird's-eye perspective, where the estimation of the speed will not be dependent on the car position. Taking the trackers from the previous section, we can easily estimate the velocity of the moving object in pixels per frame through the proposed homography $\mathrm{norm}(\vec{cd})$. Then, taking a fix reference from the real world $\vec{ab}$ and knowing their length ($\mathrm{dist}(a, b)$ given by the user) the scale factor is computed:

$$\mathrm{dist}(c, d) = \mathrm{norm}(\vec{cd}) \frac{\mathrm{dist}(a, b)}{\mathrm{norm}(\vec{ab})} \tag{3}$$

In road sequences, we can use an approximation of the length of the road central lines. This length can be easily obtained from [7]. Hence, it is defined that the mentioned lines measure 10 feet or 3.04 meters. Using this information and a scale factor α the real speed in kilometers per hour is estimated. Figure 3 shows an example for the homography.

Table 1. Set of frames for evaluation purposes.

Sequence	**Frame Range**	**Type**
Highway	1050 - 1350	Baseline
Traffic	950 - 1050	Camera jitter

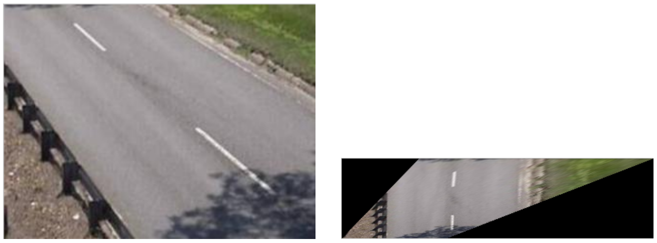


Fig. 3. Example of the homography transformation for the Traffic sequence.

Figure 2 shows an example of the proposed tracker with the computed velocity. The system keeps a tracking of the velocity in each point and their average. The velocity estimated can produce bigger jumps due to segmentation problems or other cars or shadows. Hence, take into account the average speed can give

3. EVALUATION

3.1. Database

For evaluation purposes, we used the Change Detection Benchmark Dataset [8] which is an open database for foreground segmentation purpose. From this dataset, three sequences have been used. Table 1 shows the frames that have been used for different types of evaluation.

3.2. Results

The results for all the modules will be presented. Moreover, the final estimation of the speed will be presented for different sequences[1].

3.2.1. Background Estimation

As it has been explained in subsection 2.1, the idea is to distinguish between background and foreground. Figure 4 shows a comparison in terms of f1-score for three sequences setting the parameters α and ρ. It is clear, that the best technique (in average) for our case of study is the adaptive single gaussian per pixel.

[1] `https://youtu.be/5w7WQJvfClA`

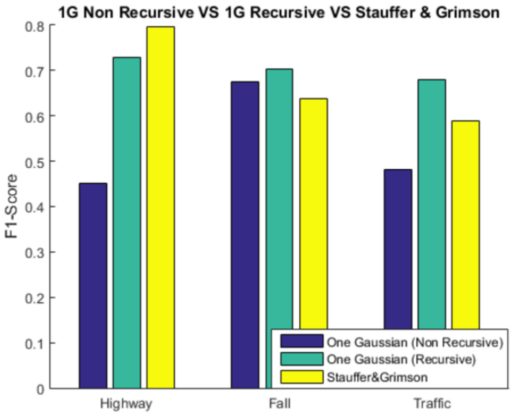


Fig. 4. Comparison of background segmentation techniques for three sequences.

Hence, this work has been devoted to improve this segmentation using color and foreground segmentation. Table 2 shows a global measure (AUC) depending of α and setting ρ to the best value obtained for each sequence (0.2 for Highway and Traffic and 0.1 for Fall sequence). Using a global measure gives us more information about the stability of the proposed technique.

For the foreground segmentation, also shown in table 2, it has been shown that the best parameters are $\mathrm{connectivity} = 4$ for the hole filling, $\mathrm{p} = 5$ for the closing and $\mathrm{area} = 300$ for the area filtering. Shadow removal algorithm has been tested but have been shown to decrease the performance of the system deleting foreground instead of background. Figure 5 shows the shadow issues where it is segmented as foreground. Moreover, for the background colors and shadows, it is spitted in two blobs and both are classified as a car.

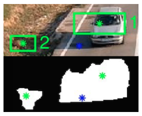


Fig. 5. Shadow are still an issue for the proposed approach.

3.2.2. *Road Traffic Monitoring*

The final objective of this work is to monitor, track and count the vehicles in some roads. Hence, we have to identify each car and be able to get an estimation of the speed along the road stretch.

Table 3 shows some examples of detected vehicles for each sequence and their corresponding speed. For the Highway sequence 29 cars have been detected (26) whereas for the Traffic sequence 23 cars have been detected (30).

The speed provided in the table is the average of all instant speed where the car is detected. For instance, some outliers can appear due to blobs that are miss classified.

Table 2. Area under the precision-recall curve for three sequences and their average. Column 2 corresponds to the AUC for the background estimation and the last one indicates the AUC for the foreground estimation and the improvement with respect the previous column.

	Back. Est.	**Fore. Seg.**
Highway	0.7539	0.8423 (+0.0884)
Fall	0.7954	0.9526 (+0.1572)
Traffic	0.7081	0.7807 (+0.0726)
Average	0.7525	0.8585 (+0.1061)

Table 3. Relations between vehicles and their speed (km/h) in two sequences

Sequence	**Highway**	Sequence	**Traffic**
	Est. speed		**Est. speed**
Car 1	62.0090	Car 1	69.5027
Car 2	88.1962	Car 2	68.4294
Car 3	54.2807	Car 3	57.3601
Car 4	53.8650	Car 4	61.688

In order to test the accuracy of the velocity estimation, a new sequence have been recorded with one of the cars moving at constant speed (40km/k or 50km/h depending on the sequence).

4. CONCLUSIONS AND FUTURE WORK

During this work, a video surveillance framework have been presented. Each module have been discussed separately and his performance evaluated. Finally, the whole system have been presented to be able to count and track the cars and estimate their velocity.

The work proposed have been shown to get good results under controlled assumptions. However, the problem of video surveillance is not solved. Future work should be focused on improve the techniques for video stabilization and shadow removal. Moreover, another key point to increase the performance of the proposed pipeline can be a module to segment objects in the foreground mask. For instance, a problem come from vehicles that are overlapped and the tracker is not capable to divide. Furthermore, other state of the art stabilization techniques can give an important increase of the performance [9].

5. REFERENCES

[1] C.H. Heartwell and A.J. Lipton, "Critical asset protection, perimeter monitoring and threat detection using automated video surveillance," in *Proceedings IEEE 36th International Carnahan Conference*, 2002.

[2] Chris Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 2, p. 252 Vol. 2.

[3] Richard Sewall Hunter, "Photoelectric color-difference meter," in *JOSA*, 1948, vol. 38.

[4] Bruce D Lucas, Takeo Kanade, et al., "An iterative image registration technique with an application to stereo vision.," in *IJCAI*, 1981, vol. 81, pp. 674–679.

[5] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[6] Naiyan Wang and Dit-Yan Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, Eds., pp. 809–817. Curran Associates, Inc., 2013.

[7] U.S. Department of Transportation, "U.s. federal transportation guide," `http://mutcd.fhwa.dot.gov/htm/2003r1/part3/part3a.htm`, Accessed: 26-02-2016.

[8] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 393–400.

[9] Johannes Kopf, Michael F Cohen, and Richard Szeliski, "First-person hyper-lapse videos," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 78, 2014.