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

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. However, lots of other applications have been proposed for video surveillance in other scenarios, for instance, (SOME EXAMPLES WITH REFERENCES). 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 the methodology used to obtain the needed information from an input video. In section .... 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 (used in Subsection 2.5).
- The user has to indicate the relationship between a pixel and the real distance. That is, for example, how many pixels equals a meter (necessary 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:

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

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. For instance, if something stops or starts moving in some region of our frame, it will be always detected as foreground.
- *Adaptive*: in order to solve the problems of the non adaptive approach, 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:

$$\mu_i = \rho I_i + (1-\rho)\mu_i$$
$$\sigma_i = \rho(I_i - \mu_i)^2 + (1-\rho)\sigma_i^2$$

2.1.2. *Stauffer and Grimson*

Stauffer and Grimson [1] 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). Also, each Gaussian is weighted: every time a current pixel is assigned to a Gaussian, its weight increases, while the other Gaussians weights decrease. In fact, a Gaussian can be eliminated if its weight is too low. If that is the case, a new Gaussian will be created. 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*

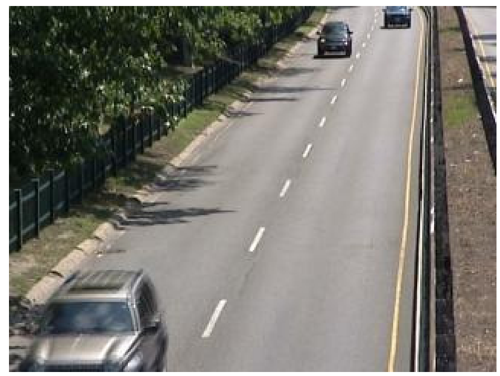
Originally, we only used gray-scale images. However, the original input images are in color, meaning that if we convert them to gray-scale, we are losing information that we could benefit from. Thus, we have tested three different color spaces: RGB, YUV and CIE-Lab [2]. The first one is the most common 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 caused by 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 by means of the following three different operations:

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

Estimation of the optical flow of the sequence using either block matching or Lucas-Kanade [3] is 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 is a very important step. 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.

Table 1. Set of frames for evaluation purposes.

Sequence	Frame Range	Type
Highway	1050 - 1350	Baseline
Fall	1460 - 1560	Dynamic background
Traffic	950 - 1050	Camera jitter

2.4. Tracking

For tracking purposes, two main approaches have been tested. First an algorithm based on Kalman filter [4] is used. This algorithm is optimal for linear dynamical models under the assumption of Gaussian noise. Then another technique [5] have been tested that uses deep learning for tracking purposes.

2.5. Velocity estimation

Taking the trackers from the previous section, we can easy estimate the velocity of the moving object in pixels per frame. However, taking a fix reference from the real world, the real velocity can be approximated. In traffic sequences, we can use an approximation of the length of the central lines of the road. These length can be obtained easily from [6]. Hence, it is defined that the dashed lines measure 10 feet or 3.04 meters. Using this information and a scale factor α the speed is guessed.

3. EVALUATION

3.1. Database

For evaluation purposes, we used the Change Detection Benchmark Dataset [7] 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. Metrics

The results will be presented in terms of F1-score, precision-recall curve and area under the curve (AUC). This metrics are computed pixel-wise using the ground truth masks.

$$\text{F1-score} = \frac{2\,\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

3.3. Results

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

[1] linkaunvideoambeltracking

Sequence	**Highway**
	Est. speed
Car 1	

Sequence	**Traffic**
	Est. speed
Car 1	

Table 2. Relations between vehicles and their speed in two sequences

3.3.1. Background Estimation

3.3.2. 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 2 shows the number of detected vehicles for each sequence and their corresponding speed.

4. CONCLUSIONS AND FUTURE WORK

5. REFERENCES

[1] 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.

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

[3] 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.

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

[5] 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.

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

[7] 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.