

# Video Surveillance for Road Traffic Monitoring

Team 7

C. Carmona, A. Flores, A. Hernández, A. Imbernon, A. Mosella

January 29, 2016

## Abstract

Computer vision systems can be applied to a wide variety of tasks, but some of the most interesting are those related with security and surveillance. Within this group, our application for Video Surveillance for Road Traffic Monitoring can be placed. We propose a solution based on machine learning and video analysis techniques that involves the whole process: database evaluation, background estimation, foreground segmentation, video stabilization and object tracking. As a result of this, our system will be able to monitorize some basic parameters of traffic flow as vehicles counting or speed estimation.

**Key words:** video surveillance, traffic monitoring, computer vision, video analysis

## 1 Motivation

As this is an academic project, the main motivation is to understand the pipeline of a project based on video analysis and apply all this knowledge to a specific task. Video surveillance and monitoring is one of the most popular applications, and also has some peculiarities that make it very interesting, as the presence of stable background for an easier segmentation.

## 2 Related work

The ability of tracking objects has been used to implement applications for different tasks. One of them is people tracking, where different techniques as adaptive background subtraction [6], Bayesian

tracking [9] or methods based on image intensity and object modeling segmentation [8] have been applied. Also in the field of assisted surveillance these techniques can be very useful since a high number of process can be automatized increasing the system reliability and performance. Many of this implementations are based on adaptive background estimation [10][11] and that is the technique that we have also chosen for motion estimation because it has reported good results and it is a good starting point given the academic nature of our project. Even so, there exist another implementations that improve the results thrown by background estimation, for example at [7] with a framework based on a multilayer hierarchy of features or [1] where applying edge-detection techniques to the key regions the irregularities due to ambient light conditions are overcome.

## 3 System description

As it was written before, our project covers all the stages of the process, from the analysis of the input sequences to the obtain of the statistics of traffic flow, passing by video stabilization, background estimation, foreground estimation and region tracking. A complete pipeline can be seen at 1.

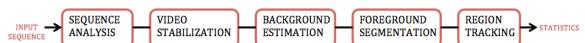


Figure 1: Project pipeline including all stages

### 3.1 Sequence Analysis

The first step is to analyze the kind of video sequences that we are going to work with. We have

selected two sequences (available at [2]) recorded from a camera located on the top of a mast pointing to a one-way dual lane. The first one has been called highway and does not need any additional correction. The second one has been called traffic and presents a high aberration due to camera jittering (as can be appreciated at 3) that should be corrected to reduce its negative impact on the results.

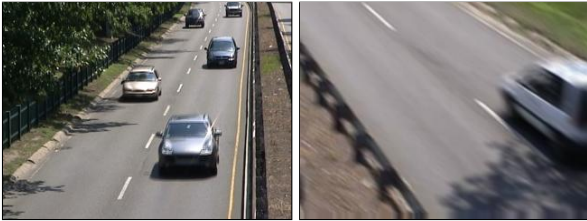


Figure 2: Frame from high-way sequence



Figure 3: Frame from traffic sequence

### 3.2 Video Stabilization

There exist different ways to face video stabilization, but we have chosen one based on optical flow. One of the most widely used technique to compute optical flow is Lucas-Kanade [4] that assumes that flow is nearly constant in neighbours pixels and computes it by applying the least square criterion to affine pixels. For the final system, we have implemented another method based on block matching. This method estimates motion between two frames by searching a block of pixels extracted from the reference frame into the current one within a defined search area. The algorithm looks for matches moving the reference block over the search area of the current image, computing the MSE for every location and choosing the value that minimizes the MSE function. As it can be seen at 1, we have got an improvement of 36,53% at PEPN using block matching instead of Lucas-Kanade (but also an increasing of computation time has been produced).

	Mean Magnitude Error (MMEN)	Percentage of Erroneous Pixels (PEPN)
Block Matching	4,3173	42,03 %
Lucas-Kanade	10,6271	78,56 %

Table 1: Results for optical flow estimation at traffic dataset

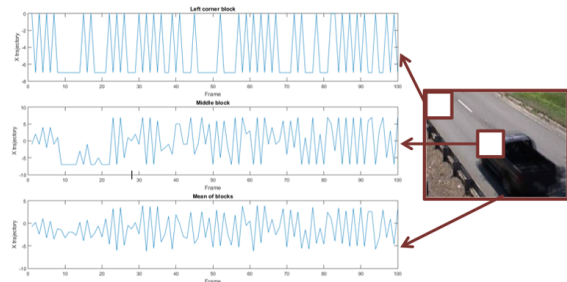


Figure 4: Trajectory of pixel blocks for traffic sequence

Once the optical flow is computed, this information will be used to align the scene by calculating the difference between the motion vectors of two successive frames and applying the affine transform. As it can be observed at 4, the trajectory of the pixel blocks is different attending to its situation over the scene: there exist areas with a low range of movement (like the upper-left corner) and others with a great variety (like the center).

### 3.3 Background Estimation

Once we have our sequence stabilized, is time to separate and classify the items of the scene in two different groups: background and foreground. A first attempt to this is Gaussian modeling, where every pixel of the background is modeled as a random variable with its own mean and variance. Once the statistical models for background and foreground have been established, changes in the scene can be detected by means of a classification process.

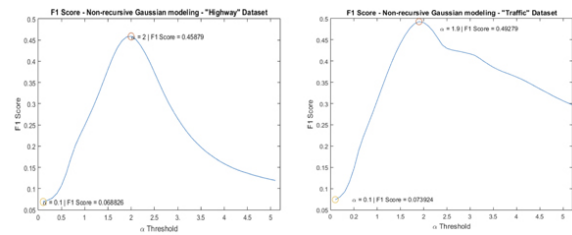


Figure 5: F1 vs  $\alpha$  for highway and traffic sequences

At 5 the curves for F1 vs threshold after evaluating the system are shown. The threshold  $\alpha$  corresponds to the width of the Gaussian curve, that is, how much standard deviation is allowed. In the case of traffic sequence it can be appreciated

a less marked fall due to a higher variance as a result of jittering.

Another possible step in our system development is update Gaussian modelling to adaptive modelling. To implement it, the first 50 % frames are devoted for training and the rest for background adaptation so the algorithm looks for the best pair of values  $\alpha$   $\rho$  that maximizes F1-score. We have been able to improve F1 scores for lower  $\rho$  values.

### 3.4 Foreground Segmentation

To enhance the foreground object's shape it would be necessary to apply some morphological techniques as hole filling or area filtering. First of all we try hole filling with 4 and 8 connectivity parameter, so we would be able to close those objects containing gaps. After that, we add area opening to the chain, so we try to remove small objects from the image. The results of applying this can be observed at 6. We have got a small improvement at F1 score.

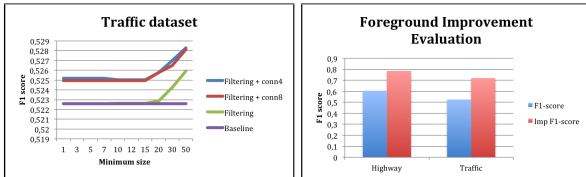


Figure 6: Foreground segmentation

Figure 7: Extra foreground evaluation

We have also found interesting trying to improve the way of compute F1 score following the implementation at [5]. This method applies weighting functions to the errors, taking into consideration the dependency between the pixels and the locations of errors. An improvement of 29,8% at highway sequence and 37,1% at traffic sequence has been reached as it can be shown at 7.

### 3.5 Region Tracking

The last step of our system consists on tracking the objects previously segmented. This process has been implemented using Kalman filter [3]. It involves two main process: predicting the next location of the object and minimizing the location error. Different parameters have been tuned to get the best results. The main problem that we

have found when working with Kalman filter is its performance when tracking more than one object at the same time as result of a strong dependency with background subtraction, what can suppose a problem for very close objects that can be confused (Kalman filter does not have into account the motion direction).



Figure 8: Using of homography to estimate distances

To estimate the speed of the vehicles we have used homography to project the image and eliminate the perspective aberration as it is reflected at 8. To obtain the homography first a vanishing point has been defined and following the expression at 9 we can estimate distances. We have made a couple of assumptions: the dashed line segments of the road are 5 meters long (Standard 8.2-IC, Ministerio de Fomento), and the time lapse between frames is 0.033 seconds. Combining all of this we can now estimate the number of vehicles on the road and its speed.

$$Homography = \begin{bmatrix} 1 & \frac{v_1}{v_2} & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{v_2} & 1 \end{bmatrix}$$

Figure 9: Expression for computing Homography.  $v_1$  and  $v_2$  corresponds to x and y directions respectively.

## 4 Evaluation

The evaluation consists on testing whether or not the system can successfully perform the desired tasks, that is, count the number of vehicles of the video sequence and estimate its speed. At 10 and 11 can be seen that in general terms the system has a good performance, even in the presence of jittering as in the case of traffic sequence.

The number of vehicles recognized is 11 for highway and 18 for traffic dataset. The real number of vehicles appearing is lower, and it is due the limitation described before for Kalman filter when there



Figure 10: Highway sequence results



Figure 11: Traffic sequence results

exist more than one object to track on the scene, at certain moments one single vehicle is perceived as two, so the counter increase in one when should not do it.

## 5 Conclusions

We have realized during the development of the project that to obtain a good tracking it is first necessary to examine the dataset to explore its peculiarities and apply as many previous techniques as needed to get a robust foreground segmentation. Also, Kalman filter would not be the better choice for this application with multiple objects to follow at the same time. Exploring another methods like mean-shift or deep learning would throw better results. We have implemented our system with simple and well known methods and our results have been satisfactory, but any kind of improvement will revert on a better performance.

## References

- [1] M Fathy and MY Siyal. A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters. *Vehicular Technology, IEEE Transactions on*, 47(4):1342–1349, 1998.
- [2] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection.net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.
- [3] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [4] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [5] Ran Margolin, Lihi Zelnik-Manor, and Avishay Tal. How to evaluate foreground maps. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 248–255. IEEE, 2014.
- [6] Stephen J McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, and Harry Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.
- [7] Kostia Robert. Video-based traffic monitoring at day and night vehicle features detection tracking. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pages 1–6. IEEE, 2009.
- [8] Akio Shio and Jack Sklansky. Segmentation of people in motion. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 325–332. IEEE, 1991.
- [9] Martin Spengler and Bernt Schiele. Automatic detection and tracking of abandoned objects. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. Citeseer, 2003.
- [10] Chris Stauffer and W Eric L Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747–757, 2000.
- [11] Jie Zhou, Dashan Gao, and David Zhang. Moving vehicle detection for automatic traffic monitoring. *Vehicular Technology, IEEE Transactions on*, 56(1):51–59, 2007.