

A Framework for Wrong Way Driver Detection Using Optical Flow

Gonçalo Monteiro, Miguel Ribeiro, João Marcos, and Jorge Batista *

Institute for System and Robotics
Dep. of Electrical Engineering and Computers
University of Coimbra - Portugal

Abstract. In this paper a solution to detect wrong way drivers on highways is presented. The proposed solution is based on three main stages: Learning, Detection and Validation. Firstly, the orientation pattern of vehicles motion flow is learned and modelled by a mixture of gaussians. The second stage (Detection and Temporal Validation) applies the learned orientation model in order to detect objects moving in the lane's opposite direction. The third and final stage uses an Appearance-based approach to ensure the detection of a vehicle before triggering an alarm. This methodology has proven to be quite robust in terms of different weather conditions, illumination and image quality. Some experiments carried out with several movies from traffic surveillance cameras on highways show the robustness of the proposed solution.

1 Introduction

In order to ensure a safe and efficient driving, it is important to classify the behaviors of vehicles and to understand their interactions in typical traffic scenarios. Until recently, this task was performed by human operators at traffic control centers. However, the huge increase of available cameras requires automatic traffic surveillance systems [1–8].

In the last decades, one of the most important efforts in Intelligent Transportation Systems (ITS) research has been the development of visual surveillance systems that could help reduce the number of traffic incidents and traffic jams in urban and highway scenarios. Although the large number of systems based on different types of sensors and their relative performance, vision-based systems are very useful to collect very rich information about road traffic.

The work presented in this paper is part of an automatic traffic surveillance system [9]. The primary goal of the system is to detect and track potentially anomalous traffic events along the highway roads. By anomalous events it is meant the detection of vehicles that stop on the highway, vehicles driving in the lane's opposite direction and also vehicles that are constantly switching between lanes.

* This work was supported by BRISA, Auto-estradas de Portugal, S.A.

Vehicles driving on the wrong way represent a serious threat. An immediate detection of a vehicle driving on the wrong direction could help prevent serious accidents by warning the oncoming vehicles (via traffic telematic systems or radio announcements) and by warning the police.

The proposed project aims at the automatic detection of drivers circulating on the wrong way and consequently triggering an alarm on the highway traffic telematic system. The system must be robust to illumination changes and small camera movements, being able to robustly track vehicles against occlusions and crowded events.

A simple way to detect the wrong way drivers is using a segmentation process to distinguish the vehicle from the background, and then tracking all segmented cars and verify if the direction of its trajectory is the correct one for the lane or if it is a vehicle circulating on the wrong side. This is not a hard process to implement, but it has some disadvantages, namely the lack of robustness to the variation of light and weather conditions, and the difficult task of tracking vehicles in crowded situations without grouping those circulating near each other. After taking all this into account we opted to use the optical flow obtained by two consecutive frames. This process is more robust and accurate as regards light and weather conditions variation.

The solution presented in the paper is based mainly on three stages. Firstly, the orientation pattern of vehicles motion flow is learned and modelled by a mixture of Gaussians (Learning Stage). Then, there is a Detection and Temporal Validation using the learned orientation model to detect objects moving on the lane’s opposite direction. On both stages, a Block Median Filtering is applied to the motion flow in order to remove noisy data. The temporal validation is applied through a kalman filter, tracking over a stack of images the blocks marked as belonging to a driver’s wrong way event. Finally, an appearance-based approach is used to validate the existence of a vehicle as an object that triggers the event, sending an alert sign in case the temporal and appearance validation succeeds.

2 Image Motion Estimation

The algorithm proposed uses motion information that can be provided by different motion estimation algorithms. A valuable comparison of different techniques is presented in [10]. Satisfactory results were obtained in several experiments by applying the method proposed by Lucas and Kanade in [11] and modified according to the work of Simoncelli et al. [12]. Furthermore, this optical flow estimation algorithm also provides an objective measurement of the local level of reliability of the motion information. Shi and Tomasi adopted this criterion of reliability in order to evaluate the texture properties of pictures areas, and achieved improved tracking performances [13]. We will not go into a detailed description of the method, but we will just report here the results of the discussion in [10]. The reliability of the estimates for a given pixel can be evaluated using the eigenvalues $\lambda_1 \geq \lambda_2$ of the matrix C (1).

$$C = \begin{bmatrix} \sum_{x \in \Omega} W^2(x) I_x^2(x) & \sum_{x \in \Omega} W^2 I_x(x) I_y(x) \\ \sum_{x \in \Omega} W^2(x) I_x(x) I_y(x) & \sum_{x \in \Omega} W^2 I_y^2(x) \end{bmatrix} \quad (1)$$

Where the summations are intended over a small spatial neighborhood Ω of the pixel, $W(x)$ is a window function that gives more influence to pixels in the center of the neighborhood, and I_x and I_y are the spatial gradients of the gray levels in directions x and y respectively. The method proposed in [11], [12] sets a condition $\lambda_2 \geq \delta$ on the smallest eigenvalue for a velocity to be evaluated; otherwise, no velocity value is assigned to the pixel.

The result of the optical flow detection contains some disturbance (see Fig. 1), which is mainly caused by motion flow discontinuity regions and noise. To reduce the disturbance and the volume of information analyzed, the image is divided into blocks of 8×8 pixels. For each block the median of the directions obtained by the optical flow is calculated (*Block Median Filtering*). From this moment on, all the references made to the movement direction in the image are related to the median motion flow of the block. Likewise, the flow detected in the image is analysed for each block instead of a pixel by pixel analysis.

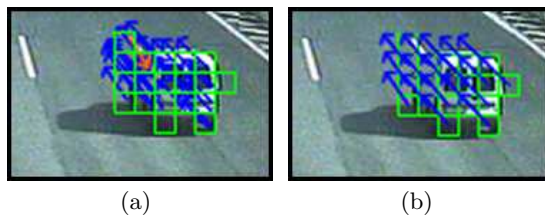


Fig. 1. Results of the Block Median Filtering to reduce the disturbance in the optical flow estimation. a) output of the estimated flow with the Lucas-Kanade Algorithm. b) block median filtering applied to a)

3 Traffic Flow Direction Learning

The basic idea when learning the patterns of vehicles' motion direction flow on the different lanes is that vehicles circulating on these, during the learning period, are moving in the correct direction along the lane.

The estimation of each lane's motion orientation on the image is learned through the analysis of a large amount of frames. A *Gaussian mixture* is modelled to learn the image motion flow orientation of each block in the image by the analysis of the vehicle's movement (see Fig. 2).

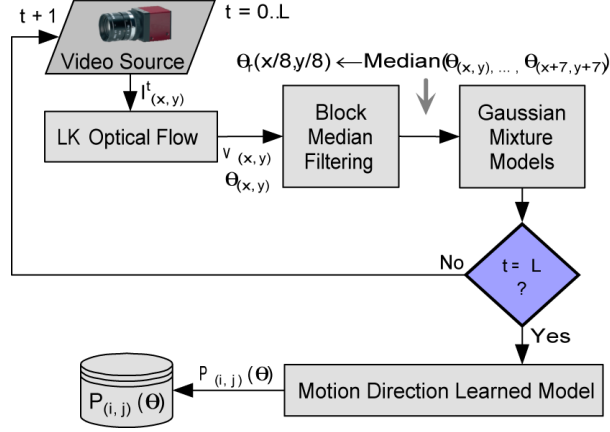


Fig. 2. Flow chart of the traffic flow direction learning process.

If it is assumed that the directions of the vehicles (θ) have a Gaussian distribution, then the direction can be modelled by a *Gaussian Mixture Model* (GMM) [14], which is given by (2), in which ω_i is the prior of the Gaussian distribution \mathcal{N}_i with mean μ_i and standard deviation σ_i , and θ is the block image direction of the movement. In practice, the number of kernels was limited to a certain maximum, namely $K = 3$.

$$p(\theta) = \sum_{i=1}^K \omega_i \mathcal{N}_i(\theta; \mu_i, \sigma_i) \quad (2)$$

The mixture model is dynamically updated. Each block direction is updated as follows: i) The algorithm checks if each incoming direction angle θ can be ascribed to a given mode of the mixture, this is the match operation. ii) If the direction angle occurs inside the confidence interval with ± 2.57 standard deviation (for the 99% confidence interval), a match event is verified. The parameters of the corresponding distributions (matched distributions) for that pixel are updated according to

$$\mu_i^t = (1 - \alpha_i^t) \mu_i^{t-1} + \alpha_i^t \theta^t \quad (3)$$

$$\sigma_i^t = (1 - \alpha_i^t) \sigma_i^{t-1} + \alpha_i^t (\theta^t - \mu_i^t)^2 \quad (4)$$

where

$$\alpha_i^t = \tau \mathcal{N}(\theta^t, \mu_i^{t-1}, \sigma_i^{t-1}) \quad (5)$$

The weights are updated by

$$\omega_i^t = (1 - \tau) \omega_i^{t-1} + \tau (M_i^t) \quad (6)$$

$$\text{with } M_i^t = \begin{cases} 1 & \text{matched models} \\ 0 & \text{remaining models} \end{cases}$$

where τ is the learning rate. The non match components of the mixture are not modified. If none of the existing components match the direction angle, the least probable distribution is replaced by a normal distribution with mean equal to the current value, a large covariance and small weight. iii) The next step is to order the distributions in the descending order of ω . This criterion favours distributions which have more weight (most supporting evidence) and less variance (less uncertainty). iv) Finally, the algorithm models each direction as the sum of the corresponding updated distributions.

Finally, after the models learning, these are filtered to eliminate some noise. All the models with a weight lower than a threshold are discarded (see Fig. 3(d)).

The main advantage of the Gaussian mixture modelling in this situation is that it can embrace a variety of directions for the same block, which is very useful in lanes with exits or bifurcations, modelling also the movements of vehicles that are changing between lanes.

The number of necessary frames to obtain a correct estimation of the GMM's depends on the number of vehicles circulating on the road. In our experiments a stack of 1000 learning frames was used. Obviously, if there are no vehicles circulating on one of the lanes, the direction of that lane will not be learned.

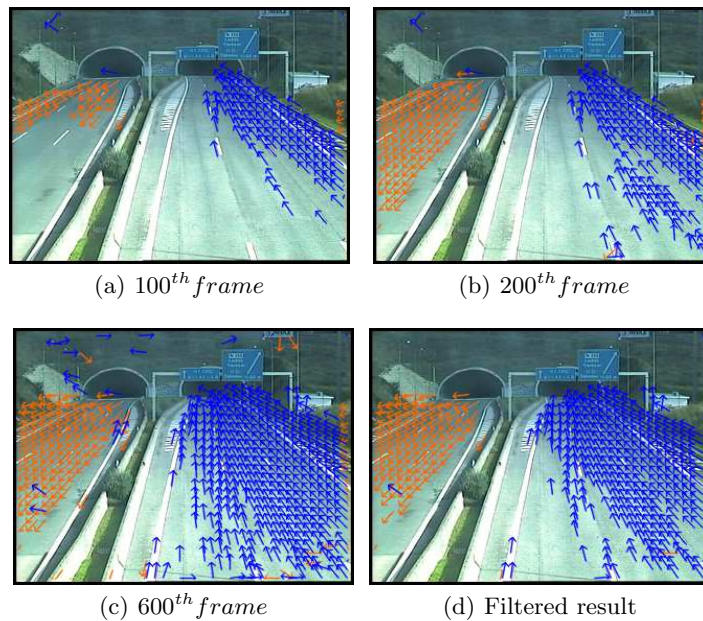


Fig. 3. a), b) and c) three frames showing the evolution of the orientation pattern modelled by the first Gaussian of the GMM on a highway scenario. d) the result of the learning process after the model weight filtering.

4 Wrong Way Drivers Detection

In this section it is described the methodology used to detect and validate the vehicles circulating on the wrong way (see Fig. 4). In each new frame the optical motion flow is computed and the median of the flow direction for each block is calculated. An object is defined as circulating on the wrong direction when the difference between both the direction of the flow in the present frame and the estimated means of the corresponding block learned are larger than 2.57σ for the 99% confidence interval.

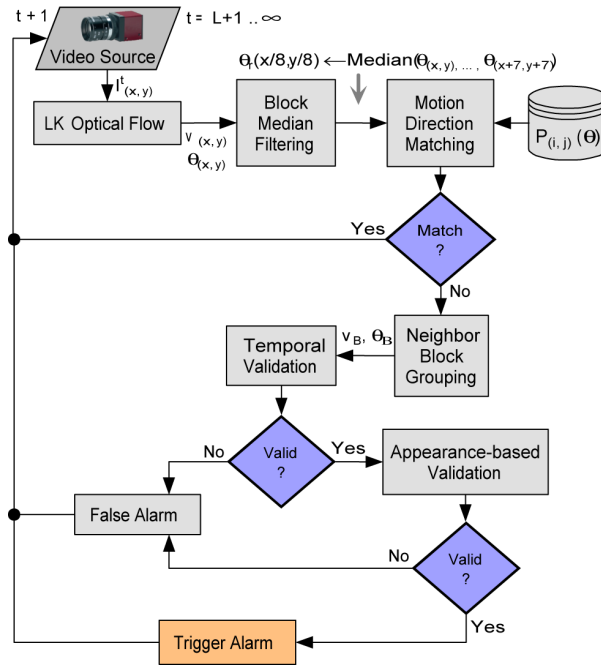


Fig. 4. Flow chart of the wrong way drivers detection system proposed here.

Due to the vibration of the surveillance camera pole and noisy motion flow estimation, it is possible that a vector or a set of vectors of flow are detected even if there is no real motion in those blocks of the image. Thus, it is necessary to validate all the objects detected in the wrong way before triggering an alarm. Two types of validation were used, namely a temporal validation, to verify if the detected objects make a coherent trajectory, and an appearance-based validation to check if that object is, in fact, a car.

4.1 Temporal Validation

The temporal validation consists of tracking all the objects detected as circulating on the wrong side of the road and verifying if they appear in consecutive frames. If an object is detected more than n times in m consecutive frames and makes some coherent trajectory, then it will be considered as an object circulating on the wrong side of the road, namely $n = 4$ and $m = 6$. A second order Kalman filter is used to track and predict the position of the vehicles in consecutive frames.

When a detected flow does not match with the learned motion direction model, a new *tracker* is initiated. The object image position, P , is given by the center of mass of all neighbor blocks detected as being part of an object moving in a wrong direction. The velocity, ν , of the object is obtained in 2 parts: the direction is obtained as the median of the direction of all grouped blocks, and the module is computed by the average motion of all grouped blocks. When tracking the object it is only necessary to save P , ν , and the area of the grouped blocks. The m frames used for temporal validation are stored and used in the appearance-based validation.

4.2 Appearance-based Validation

The appearance-based sub-system described here receives as input all frames used to validate the vehicle in the temporal validation sub-system. The appearance-based sub-system is applied to the sub-windows where flow has been detected in m frames and it verifies if it is a vehicle or a false positive.

This system uses a set of *Haar-Like* features (see Fig. 5) to extract the information from the given image [15]. The detection of the objects is performed using these features as an input to an AdaBoost classifier. The AdaBoost classifier is then trained to perform the detection of the vehicles on the road. The main goal of this learning algorithm is to find a small set of *Haar-Like* features that best classifies the vehicles, rejecting most of the background objects, and to construct a robust classifier function. To support this purpose, a weak learning algorithm is designed to select the single feature which best separates the positive and negative examples. For each feature, the weak learner determines the optimal threshold classification function, so that the minimum number of examples are misclassified. A weak classifier $h_j(x)$ consists of a feature f_j , a threshold θ_j and a parity p_j indicating the direction of the inequality sign (7). The value 1 represents the detection of the object class and 0 represents a non-object. Each of these classifiers alone is not able to detect the object category. Rather, it reacts to some simple feature in the image that may be related to the object. The final classifier $H(x)$ is constructed with the weighted sum of the T weak classifiers and is represented by (8).

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$H(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

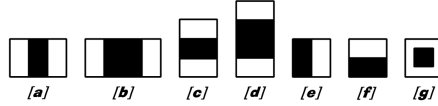


Fig. 5. Subset of the Haar-like features prototypes used in the object detection. *a*, *b*, *c* and *d* are the line features, *e* and *f* the edge features and *g* is the center-surround feature.

To construct a robust and accurate cars classifier it is necessary to gather a large amount of labelled cars in the scene that we want to use the classifier. A segmentation process [9] was used to obtain the labelled cars in the scene images.

The detection of the objects is done by sliding a search window through each sub-image and checking whether an image region at a certain location is classified as a car (see Fig. 6). Initially, the detecting window is of the same size of the classifier (30×30), then the window's size is increased by β until the size of the window is equal to the sub-image size ($\beta = 1.05$).

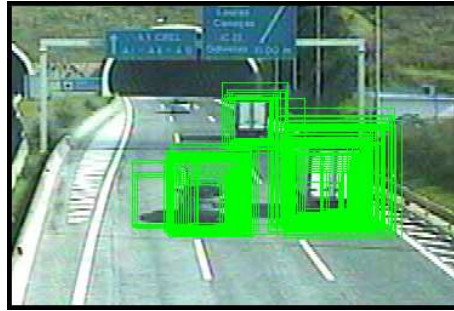


Fig. 6. Cars classification using AdaBoost classifier at different scales and positions of the search window on the image.

The appearance-based validation is carried out after the temporal validation. In order to validate an object as a vehicle it should be positively classified at least q times in m consecutive frames. A value of $q = 4$ was used.

5 Experimental Results

The system described here was tested by using a real set of image sequences from highways traffic surveillance cameras with different weather conditions, illumination, image quality and fields of view.

This set of image sequences is composed by real and simulated wrong way events. In fact, in some of the simulated video sequences the vehicles were not circulating on the wrong side of the road. These situations are scarce and it is difficult to obtain videos of these events when they happen. To test the system, all the directions learned during the training phase were increased by π and, therefore, all vehicles in the road should be considered as circulating on the wrong direction (see Fig. 7). All vehicles detected as wrong way drivers are bounded by a red box. A simulated wrong way driver event was also tested with this algorithm. In this situation, a vehicle is entering the highway through an exit lane, and the system was able to detect the event correctly (see Fig. 8). Another simulated situation was also tested, including frames from a video in a tunnel with a wrong way driver event (see Fig. 9). It is worth noting that in the experimental result image sequences, the vehicle seen in the first frames as going in the wrong direction is not detected due to the temporal validation.



Fig. 7. Wrong way drivers detection in real event video sequence. All the moving vehicles were validated as wrong way drivers.

A set of 300 frames of highway tunnels in two different scenarios and 600 frames of outdoor video sequences in five different scenarios were used to test the system. All the directions learned during the training phase were increased by π , and, therefore, all vehicles in the road should be considered as circulating on the wrong direction of the lane. The results of this experiments are presented in Table 1. In the first column of the table represents the rate of cars detected



Fig. 8. Wrong way driver detection on simulated event video sequence. In the first three frames of the sequence, the vehicle entering the highway through the exit lane is being validated and then it is detected as a wrong way driver.

with flow by each frame in the experimental image set. The hit rate is obtained by number of vehicles detected in the wrong direction of the lane divided by the total number of vehicles circulating on the road and the false alarm rate is the number of wrong way drivers events falsely detected divided by the total number of frames.

Table 1. Performance of system here proposed tested in various scenarios.

	Flow Detection (%)	Hit Rate(%)	False Alarm Rate(%)
Tunnels	89.85	92.31	0.03
Outdoor	89.55	89.86	0.24

The system’s performance in the tunnels is higher because its illumination is controlled, unlike the outdoor situations, where the illumination depends on the weather conditions, which are unstable. An additional problem in the outdoor scenarios is the vibration of the camera supporting poles, which induces a false optical flow in the image and, usually, the vehicles in the image are considerably smaller than those in tunnel situations.

The system can detect vehicles moving on the wrong direction of a lane over a 320×240 pixel image at 33 frames/s on a 3.2 GHz *P4 Intel Processor* under Linux OS. This approach is part of a traffic surveillance system that is currently being tested in some of the Brisa’s Highway roads.



Fig. 9. Wrong way driver detection in a simulated event video sequence in a tunnel.

6 Conclusions

In this paper, a methodology to detect vehicles circulating on the wrong side of the highway using optical flow is proposed. In the learning phase, the direction of each lane is modelled by a Gaussian Mixture. The optical flow is calculated to detect the moving objects in every frame. If the calculated direction does not match the Gaussian Mixture Model, then a temporal and an appearance-based validation are initiated. After all these procedures, if the vehicle is validated, an alarm will be triggered.

The experiments conducted on a large number of scenes demonstrate that the proposed system has the following properties: a) it is able to detect vehicles circulating on the wrong side of the road with good accuracy; b) it runs in real-time; and c) it is robust to variation of weather conditions, illumination and image quality.

References

1. G. Foresti, "Object detection and tracking in time-varying and badly illuminated outdoor environments," in *SPIE Journal on Optical Engineering*, 1998.
2. M. Piccardi R. Cucchiara, C. Grana and A. Prati, "Detecting moving objects, ghosts and shadows in video streams," in *IEEE Trans. Pattern Anal. Machine Intell.*, 2003, pp. 1337–1342.
3. B. Coifman J. Malik D. Beymer, P. McLauchlan, "A real-time computer vision system for measuring traffic parameters," in *IEEE CVPR*, 1997.
4. et. al. D. Koller, "Towards robust automatic traffic scene analysis in real-time," in *Int. Conference on Pattern Recognition*, 1994.
5. K. Ikeuchi M. Sakauchi S. Kamijo, Y. Matsushita, "Occlusion robust vehicle detection utilizing spatio-temporal markov random filter model," in *7th World Congress on ITS*, 2000.

6. D. Magee, “Tracking multiple vehicles using foreground, background and motion models,” in *Image and Vision Computing*, 2004, pp. 43–155.
7. T. Ebrahimi A. Cavallaro, O. Steiger, “Tracking video objects in cluttered background,” in *IEEE Transactions on Circuits and Systems for Video Technology*, 2005, pp. 575–584.
8. R. Collins et al., “A system for video surveillance and monitoring,” in *CMU-RI-TR-00-12*, 2000.
9. C. Fernandes J. Batista, P. Peixoto and M. Ribeiro, “A dual-stage robust vehicle detection and tracking for real-time traffic monitoring,” in *IEEE Int. Conference on Intelligent Transportation Systems*, 2006.
10. D. J. Fleet J. L. Barron and S. S. Beauchemin, “Performance of optical flow techniques,” in *Int. J. Comput. Vision*, 1994, p. 4377.
11. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *DARPA Image Understanding Workshop*, 1981, p. 121130.
12. E. H. Adelson E. P. Simoncelli and D. J. Heeger, “Probability distribution of optical flow,” in *IEEE Conf. Comput. Vision and Pattern Recognition*, 1991, p. 310315.
13. G. Foresti, “Object detection and tracking in time-varying and badly illuminated outdoor environments,” in *SPIE Journal on Optical Engineering*, 1998.
14. G. Stijnman and R. van den Boogaard, “Background extraction of colour image sequences using a gaussian mixture model,” Tech. Rep., ISIS - University of Amsterdam, 2000.
15. P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *IEEE Conf. Comput. Vision and Pattern Recognition*, 2001.