ROAD TRAFFIC MONITORING SYSTEM USING COMPUTER VISION TECHNIQUES

Arantxa Casanova, Sergio Castro, Belén Luque

Universitat Autònoma de Barcelona
{*arantxa.casanova, sergio.castro, belen.luque*}@*e-campus.uab.cat*

ABSTRACT

We present a road traffic surveillance system developed in Matlab that uses computer vision techniques to detect, track and estimate the speed of cars in a road, among other measures. After a foreground detection algorithm, a projective transformation is applied to the frames to obtain an aerial perspective and have a constant relation between meters and pixels. Then, vehicle detections are converted to blobs and tracked with Kalman filters, to finally extract some measures of the state of the road and the vehicles in it. The system has been tested with a video recorded using a smartphone in daylight conditions. The results provided include the number of cars per track, the density of the road, car speeds and images of the cars that surpass the speed limit.

Index Terms— traffic monitoring, vehicle detection, vehicle tracking, foreground detection, background subtraction, homography

1. INTRODUCTION

Intelligent systems that can monitor a road without the help of a human to extract conclusions are a growing demand nowadays. They can provide a vast surveillance, without almost any effort, to identify possible infractions, keep the roads safer and extract statistics.

We have developed a prototype that allows us to detect, track and estimate the speed of cars within a road. We also provide the density of the road at each frame, the number of cars in each track and photographies of vehicles that surpass the speed limit.

2. RELATED WORK

Given the aforementioned growing demand and great benefits of unsupervised traffic monitoring systems, the researching community has proposed many solutions to traffic monitoring problems in the past few years. The contributions range from monitoring the roads to adapt the traffic light timing [1] to using Unmanned Aerial Vehicles (UAV) to obtain faster and more flexible systems and larger areas to monitor [2]. Moreover, the also increasing interest in deep learning methods has favored the improvement of traffic monitoring systems by applying deep learning to both the foreground detection [3] and tracking [4]. Additionally, rich data such as 3D point clouds [5] or superpixels [6], has also been used to improve the tracking in the last few years.

Our work attempts to implement a cheap but robust monitoring system with diverse computer vision techniques that provides a good performance and fast computational time.

3. COMPLETE SYSTEM

The pipeline of our system, illustrated in Figure 1, has 4 different blocks. In the first step, the foreground objects of the grayscale input images are detected. The process is explained in subsection 3.1. Secondly, a filtering process and some corrections are applied, as detailed in Subsection 3.2, to clean the resulting binary mask from the previous block. Thirdly, cars are identified as blobs and tracked with Kalman filter, as documented in Subsection 3.3. Finally, a speed is estimated after applying homography -Subsection 3.4-. As output, the system has several measures, including car speeds, number of cars, density of the road and photographies of cars which surpassed speed.

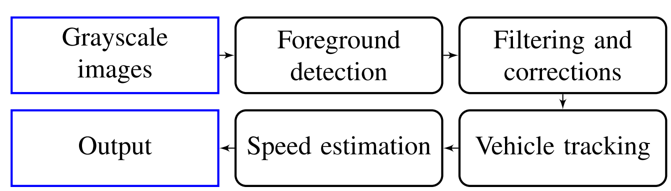


Fig. 1: Road traffic monitoring system pipeline

3.1. Foreground detection

The first step of the system is to detect the foreground objects of the scene. To do this, we first build a model of the background and then, for each pixel in a new frame, we assess if it belongs to the background model or, in the contrary, it belongs to a foreground object. In order to estimate the background, we used a single Gaussian adaptive model per pixel.

Comparing the performance of the background subtraction algorithm with different color spaces such as YUV, RGB and grayscale, the latter provided a better performance. Moreover, only one Gaussian per pixel is needed, unlike the 3 gaussians that YUV or RGB require, which results in less computational time.

To compute the background model in the test sequence, only 25 frames are used, to avoid any vehicles passing by the region of interest. The mean and the standard deviation obtained for the test sequence are shown in Figure 2.

Fig. 2: Mean (left) and standard deviation (right) of our sequence

A higher deviation in the upper regions of the image can be observed, since it is a fully operational road and it was not possible to obtain a sequence of frames without any vehicle in it. However, the road region of interest is not affected by this fact and does not affect further computations. The model used updates the values of the mean and standard deviation of the background model through time. This allows the model to adapt to changes in lighting throughout the day and small changes in the camera position or still objects left in the scene.

Once the background model is built, a pixel in a new frame is considered background if its value is comprehended between the mean minus/plus three standard deviations. Finally, a binary mask is created, as shown in Figure 3, were white pixels represent foreground and black pixels represent background.

Fig. 3: Input color frame (left) and resulting mask (right) after background removal

3.2. Filtering and corrections

The need of a filtering and correction of the obtained detection in order to eliminate noise and fill the shapes of the detected objects is evident in Figure 3.

Since this system aims to detect a vehicle and track its position with Kalman filter (see section 3.3), the shape is not as important as obtaining compact detections and avoiding the omission of small objects. For this purpose, we applied the following techniques:

- Filling the holes of the binary mask.
- Removal of connected components with less that 20 pixels.
- Closing and Opening with a structuring element of a disk of radius 5.
- Median filter of 15x15.

The output of this block is, as seen in Figure 4, a cleaned version of the binary mask.

Fig. 4: Original image (left), output mask with background removing (center), output filtered mask (right)

To improve detections and avoid false background/foreground labellings, we also used shadow removal [7] and video stabilization using block matching. These techniques were tested in two toy sequences of the Change Detection dataset [8]. We assessed that the video stabilization improved the detection when a scene was recorded with jittering. However, our test sequence did not have jitter, so the improvement is not noticeable. As for the shadow removal, some parts of gray vehicles were eliminated together with the shadows. For this reason, we decided not to use it as we are not interested in the real shape of the vehicles but cannot allow for small objects to be erroneously eliminated.

3.3. Vehicle Tracking

To be able to identify a vehicle throughout a sequence, we tested two different tracking algorithms: Mean Shift [9] and Kalman filter [10, 11]. The first one compares the color properties of the detections to decide whether a new detection corresponds to a tracked object or not. The second one takes blobs as input and uses measurements and predictions to determine the tracking of an object. With the implementations used, Kalman filter proved to be te one which is more robust to false detections, keeps a better track of detected vehicles, and has less false assignments of detected cars. Therefore, our final system is implemented using Kalman Filter.

3.4. Speed estimation

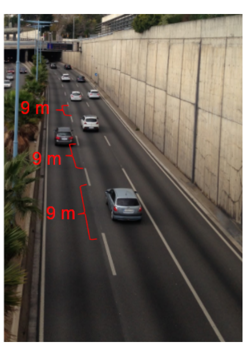
To have a consistent estimation of the speed of the tracked cars, the original geometry of the recording provided changing relations between pixels and meters, as seen in Figure 5a.

The original perspective of the tested video sequence does not allow for a constant relation between pixels and meters throughout the road, as seen in Figure 5a.

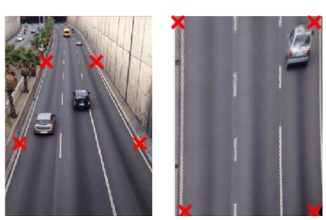
Our solution to this problem is to apply an homography transformation in order to change the perspective to an aerial view. Following the 8-point algorithm [12] to find the fundamental matrix, we identified 4 points in the original frame and mapped them into the 4 corners of the new perspective image. The fundamental matrix relates all pixels in the original image to the final one. After this step, considering 30fps and an empirical relation between meters and pixels, we can obtain a constant speed estimation in all the image using the basic speed formula (equation 1)

$$speed = \frac{displacement}{time} \tag{1}$$

Speed is updated every frame and computed using the displacement and time variation from the current to the past 3 frames. The displacement is calculated as the number of pixels that the upper left corner of the bounding boxes moves from one frame to another. In order to avoid bad estimations of the cars entering or leaving the road, we discard the estimations done at the top and bottom of the images.



(a) Pixel to meters relation across all image



(b) Homography mapping

Fig. 5: Homography applied to obtain a constant pixel to meter relation

3.5. Output of the system

Using the information obtained in the previous blocks, our system is able to provide an output such as Figure 6 for every frame in a sequence, where it is shown a qualitative measure of the density of the road, the number of cars in each lane, the instant speed of each car and a colored bounding box indicating if the speed surpasses the limit (in red, otherwise the bounding box is yellow).

Furthermore, for each vehicle that goes above the speed limit, a picture is taken and shown at the end of the sequence. This picture is obtained by applying an inverse homography to the bounding box of the car in the aerial view.

Finally, at the end of a video sequence, the total number of vehicles detected is indicated.

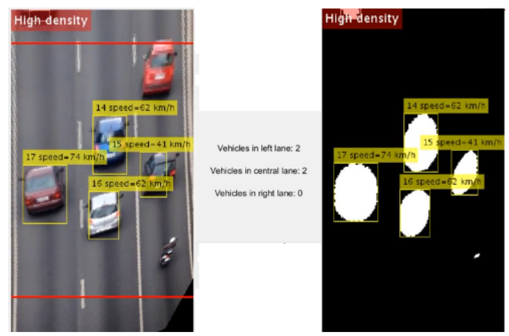


Fig. 6: Output example

4. EVALUATION

Given that our sequence does not have a ground truth to extract numerical results, we did a qualitative evaluation by visually inspecting the results obtained.

4.1. Tested data

The test video was recorded in the road *Ronda de Dalt* of Barcelona, where there is a speed limit of *80 km/h*. As we were only going to be using visual inspection for the assessment of our system for this sequence, we studied some aspects of the location to know what to expect as normal values in the output. We observed that in this particular stretch of the road, there is an entrance to the highway, which may cause the vehicles in the right lane to have an abnormally low speed if there happens to be a high density of cars that complicate the incorporation to the road. Furthermore, there is a traffic sign alerting of a speed control, so we expect the vehicles not to greatly exceed the maximum speed. In Figure 7, the recording scenario is depicted.

4.2. Results

The observed speed of the vehicles in the test sequence varies from 50 to 90km/h approximately, which we assume is correct

Fig. 7: Location where our test sequence was recorded

given the observations of the scenario previously done.
However, we also obtain some false estimations, visible in some of the pictures taken of the cars over the speed limit in Figure 8, which go over $150km/h$ and even $300km/h$. The rest of detections shown in the figure are assumed to be correct, as the speed is close to $80km/h$.

Fig. 8: Photography with cars that exceeded the speed limit

On another note, the system detected a total of 44 cars along the whole test sequence. By visual assessment, we know that the real number of cars is 43, so there was only one false detection.

5. CONCLUSIONS AND FUTURE WORK

The road traffic monitoring system we provide is robust to light changes and small camera jitter, due to the adaptive Gaussian background modelling and motion compensation respectively. Using homography we obtained a good transformation so we could make an assumption of constant velocity and simplify and improve the speed estimation.
This prototype sets the basis for a very complete road monitoring system. Some of the possible improvements or additions to the system would be to use deep learning techniques to segment the cars and track them, to implement a collision avoidance system with the information of the cars' position and speed, or to apply super resolution to the photographies of the cars that surpassed the speed limit to then apply OCR and identify the plate for further legal purposes.

6. REFERENCES

[1] N. V. Hung, L. C. Tran, N. H. Dung, T. M. Hoang, and N. T. Dzung, "A traffic monitoring system for a mixed traffic flow via road estimation and analysis," in *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, July 2016, pp. 375–378.

[2] R. Reshma, T. Ramesh, and P. Sathishkumar, "Security situational aware intelligent road traffic monitoring using uavs," in *2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA)*, Jan 2016, pp. 1–6.

[3] "Deep background subtraction with scene-specific convolutional neural networks," .

[4] Anton Milan, Seyed Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler, "Online Multi-Target Tracking Using Recurrent Neural Networks," *arXiv:1604.03635 [cs]*, Apr. 2016, arXiv: 1604.03635.

[5] Mahdieh Poostchi, Hadi Aliakbarpour, Raphael Viguier, Filiz Bunyak, Kannappan Palaniappan, and Guna Seetharaman, "Semantic Depth Map Fusion for Moving Vehicle Detection in Aerial Video," 2016, pp. 32–40.

[6] D. T. Lin and C. H. Hsu, "Crossroad Traffic Surveillance Using Superpixel Tracking and Vehicle Trajectory Analysis," in *2014 22nd International Conference on Pattern Recognition*, Aug. 2014, pp. 2251–2256.

[7] Ahmed Elgammal, David Harwood, and Larry Davis, *Non-parametric Model for Background Subtraction*, pp. 751–767, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[8] "Changedetection database, a video database for testing change detection algorithms," `http://changedetection.net/`.

[9] Jordi Frias Navarro Arcadi Llanza Carmona Sergi Canyameres Masip, Dèlia Fernández Canellas, "Road traffic monitoring," 2016.

[10] MathWorks, "Motion-based multiple object tracking," `https://es.mathworks.com/help/vision/ref/vision.histogrambasedtracker-class.html`.

[11] Greg Welch and Gary Bishop, "An introduction to the kalman filter," 1995.

[12] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, Sept. 1981.