LOW COST VIDEO SURVEILLANCE SOLUTION FOR ROAD TRAFFIC MONITORING

Luis Lebron Casas , Jose Luis Gomez Zurita

Master Computer Vision, Universitat Politecnica de Catalunya

ABSTRACT

Road safety is an important topic in order to reduce the number of mortal victims yearly and increase the security for drivers. In order to promote it, a video surveillance that detects object in motion and estimate its speed can be helpful in order to control which drivers respects speed limits. In this paper, we propose a low cost solution where using a road recording with a cheap monocular camera, we can perform a vehicle tracking on roads and estimate the speed with simple references. Our approach use well known techniques in order to stabilize the recording using optical flow and extract the background with Gaussian models in order to detect moving object. Also, we apply techniques to increase the robustness of the detection to finally apply tracking systems to this detections in order to estimate the speed. We evaluate the system in KITTI datasets and our own road recording.

***Index Terms*—** Video surveillance, Object detection, Velocity control, Traffic control

1. INTRODUCTION

Nowadays, the numbers of vehicles and roads dont stop increasing, making the road safety an essential matter, because it is alarming the number of mortal victims yearly and the imprudences that some drivers commit. Even if it tries to raise awareness on the population, it still necessary use the actual technology systems in order to support all the possible measures to increase the road security. One of this systems is video surveillance, that it helps to control the traffic flow and detect imprudences, like drive over speed limit or dangerous actions. In order to control this actions, its needed to detect and know which vehicles are committing it and Its here, where the computer vision techniques are applied.

Actually, there is a vast number of publications related to object detection, using the most advanced techniques in order to perform detection and tracking systems and the numerous utilities that it have. This field is becoming more and more relevant due the research in autonomous driving systems and the necessity of object detection and tracking in order to understand the environment for this systems.

In our case we implement a simple and reliable system, based in well known visual techniques following related work, like [1], [2] where some approaches are implement in order to estimate the speed of the vehicles. Our framework consist in the next pipeline [fig 1]: a first step performing a video stabilization using optical flow approach, followed by adaptive Gaussian mixture model to obtain a mask with the background subtracted. Morphological operations and shadow removal techniques to the mask in order to increase the robustness of our vehicle detection, to finally apply tracking techniques in order to count the vehicles and estimate the speed during the video sequence.

2. METHODOLOGY

In the following section we will explain the different goals and methods that were studied to solve the problem. At the end of each part we will expose which method give use the best result for our on video. This final solution could vary depending in which video is applied.

2.1. Background Estimation

The first part of the project is to obtain a background estimation in order to be able to split the foreground and background. We studied three different algorithms to represent the background: non adaptive model, adaptive model and Stauffer and Grimson.

2.1.1. Non adaptive model

The first approach was to use a single Gaussian to represent the background. In this model the background image is computed using the mean and standard deviation of a subset of images define as train.

Then to applied the background estimation we use the following algorithm for each pixel of one image:

For all *pixel i* if $|I_i - \mu_i| \geq \alpha(\sigma_i + 2)$ **then** *pixel* $\rightarrow$ Foreground **else** *pixel* $\rightarrow$ Background,where μ is the mean

luis.lebron@e-campus.uab.cat
JoseLuis.GomezZ@e-campus.uab.cat

computed at train, σ is the standard deviation also computed at train and α is a constant.

The result is an image were each pixel is defined as foreground or background.

2.1.2. Adaptive model

The previous methods doesn't give a good result. To improve this one thing that could be done is update the values of the mean an variance (or standard deviation) in each step. The process is similar, first we computed a mean and standard deviation from a subset of train and the for each image of test we computed if it is foreground and background with the same equation but this time we update the background model with the pixels we classify as background in the new image with the following equations 1, 2 .

$$\mu_i = \rho I_i + (1-\rho))\mu_i \tag{1}$$

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

This is the method that give us better results and will use in posterior algorithms.

2.1.3. Stauffer and Grimson

The last approach consists in using a Mixture of Gaussians to model the background and foreground. Each Gaussian only represent the foreground or the background, and each pixel of an image is label depending the nearest Gaussian. The result is supposed to produce a stable result for lighting changes, repetitive motions from clutter and long-term scene changes.[3]

2.2. Improve foreground

The result of the previous methods, even the best algorithm, gave a poor segmentation where many cars have holes or some noise is preserved. To solve it and make our result more robust, in this section we study two methods to improve the binary image obtain from the background estimation. The first one is shadow removal to eliminated the shadow that are detected as a foreground and morphology operations to suppress noise and restore the objects of the scene.

2.2.1. Shadow Removal

As a first step, in order to make our result more robust, a shadow removal technique is applied if it's needed. There are different methods to remove shadow where the color space is involved, our approach follow the steps in [4] [5] and use RGB color space. Following the idea of a shadow being a region of the background darker, we separate the color information from the lightness, calculating the chromaticity coordinates r, g and lightness s using the equation 3.

$$r = \frac{R}{R+G+B},\ g = \frac{G}{R+G+B},\ s = R+G+B \tag{3}$$

Fig. 1: Shadow removal process from an input image. Top right lightness image $\frac{s_t}{s}$, bottom left shadow detected and bottom right final mask removing shadows.

Then, we have a region background with a pixel value $[r, g, s]$. If this pixel is covered by a shadow, it is expected a value $[r_1, g_1, s_\mathrm{t}]$, where $r \approx r_1$, $g \approx g_1$ and $\alpha \leq \frac{s_t}{s} \leq \beta$. Then, s_t must be lower than s, because it's luminance value is lower for a shadow. Finally, we only need to find the proper values for α (≈ 0.5) and β (≈ 1) in order to detect only the part that correspond to the shadow and remove it from the background detection, process shown at figure 1.

2.2.2. Morphology

Other way to improve the result is use some morphology operators to reduce the noise of fill the missing parts. The main disadvantage is that the best convention of operators isnt general and a fine tune for each sequence is required. However, in our case we try to produce a method which goal is not to be so aggressive and could be applied to all the sequences although it does not accomplish the best result. The process has the following steps: eliminated the small areas, use a closing to join the disconnected parts of a figures and fill the holes in the objects

With this sequences of operators, we get a binary image with the missing information restore and improve the validation measures of all the sets in test.

2.3. Video stabilization

Other errors in the scene were cause for the shake of the camera. This produce that the scene recorded has not the same position of the road in all the frames. The result of apply background estimation on this kind of sequence contains some discrepancies because the road does not coincide. To work in this problem, we try to applied to different approach of video stabilization: optical flow with Block Matching and Feature Points Matching.

2.3.1. Optical Flow with Block Matching

In this approach we use try to get a vector of two components that represent the general optical flow between two frames, then to correct the displacement we apply a translation following this vector.

We use Block Matching backward with a size of 17 for the block size and the search area. The Block Matching consist in for all the blocks of the current frame search for the matching block on the previous frame in a limited area. When we obtain all the vectors which represent the matching blocks between the current and the previous frame, we computed the vector that is repeated more time in the scene and then we use it to transform the current image. In order to chain all the transformation, we always use the motion compensated image as the previous frame and not the actual one.

2.4. Feature Points Matching

In this method we want to extract some features from both frames, the current and the previous one, and then match them and obtain a transformation between the two frames. We use SURF as the feature descriptor, then we match the features and computed a geometry transformation. In the moment that we obtain the transformation we apply it to the current image and as in the block matching we use the compensated image as a previous frame for the next iteration.[6]

This solution has a problem respect to block matching and its that at some times the feature descriptor doesnt get enough points to do the match and the resulting transformation is erroneous. That why it could not be use, although it has a better result than Block Matching for the segments of the video where it could have computed enough features.

2.5. Region tracking

When we have the best result for the foreground and background segmentation, the next step is to follow the objects in the sequences. To do so a tracking algorithm is needed, we have studied two: the Kalman filters and the Particle filters. Both of the follow the same steps: get the foreground, detect the interest objects, predict the new location, detect the tracks, update all the tracks assigned and unassigned, delete the lost tracks and create the new tracks

The important steps are the first four.The first one is the previous studied of how to separate the foreground from the background. Then we use assigned to each connected region with a minimum size a bounding box.The one which change in the different algorithm is the prediction step and will be explain for each one in the following sections. The last one is to search for the tracks which prediction are near the detected objects or if it fails created a new track.

2.5.1. Kalman filter

Kalman filters is a model which use the measures over time of some variable with the assumption of Gaussian noise. Then with a Bayesian inference and estimating a joint probability distribution the next position of the car could be predict. [7]

One of the main advantage of the Kalman filter is its reliability compared to other methods because it use various measures and take into account some noise in the values.

2.5.2. Particle filter

The particle filter is a Monte Carlo method base on the use Bayesian state estimator with particles to generated a distribution of the predicted positions. As a Monte Carlo method in each step the model predicts the position and is updated with the real prediction.[8]

The advantage of this method over the Kalman filter is that it could recover a occlude track or a lost track because the big number of particles use makes easier to track a single object even it is only detected partially.

2.6. Speed estimation and cars count

The final part of our algorithm is focus on extract information from the frames. In this case we want the speed of each car and count the number of car that drive through these roads.

2.6.1. Speed estimation

We use a simple approach to computed the velocity propose by Team 2 of the previous year [9]. The formula in equation 4 explain how it is computed. The distance use is the Euclidean distance between the centroid of the first time the object is see in the video and the centroid in the current frame. Then the number of frames is the sequence of frames when the track has follow the car, divide by the fps of the video gets the real measure of time in seconds. The last constant is referring to the conversion from pixels to meters, this is estimated by hand. In the traffic sequence we use as a referent the white line that divide the road which following the traffics rules must measure 4.5 meters. In the other sequence this measure is adapt to match the speed of one car, which we need to know or suppose.

$$Speed = \frac{distance * pixels\_to\_meters}{\frac{number\_of\_frames}{fps}} \tag{4}$$

One consideration for this system is the perspective of the camera. In order to get all the measure correctly we need that all the pixels of the image represent the same measure in the reality. To do it we use a holography from four points from the road, we use the four extremes, to four imaginary points to create a simulation of an image capture from a flying perspective. Then we want to computed the speed we transform each centroid to the new spaces.

Table 1: Table evaluating the F1-score and AUC for the CD-NET dataset using pixel by pixel evaluation with the Ground Truth (F1-score / AUC).

	Highway	**Fall**	**Traffic**
Normal	0.53 / 0.46	0.69 / 0.69	0.51 /0.46
Shadow removal	0.39 /0.3	0.54 / 0.46	0.33 / 0.2
Morphology	0.67 / 0.65	0.83 / 0.86	0.53 / 0.49
Shadow removal + morphology	0.56 / 0.53	0.68 / 0.64	0.38 / 0.25

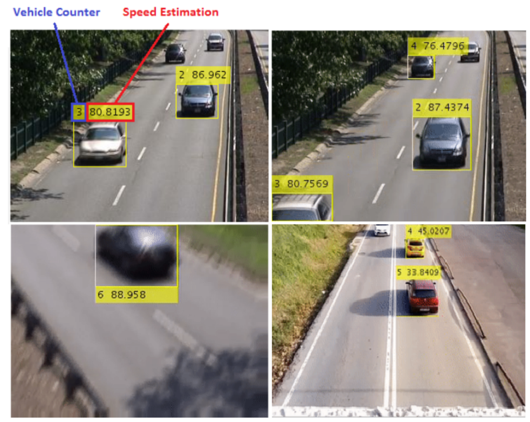

Fig. 2: Car tracking detection from frames of the different datasets tested (Highway at top, traffic at bottom left) and our own video (bottom right), showing the estimated speed and the vehicle counting

2.6.2. *Cars count*

To obtain this measure we use the identifications assigned by the tracking algorithm. For each identification we need to decide if it's have lived enough to show it in the result. The condition to do it is if the number of frames that we have keep track of it. Then we it is decide to show the track a new identification is assigned representing the cars that where show. This new identification could also be used to count the number of car.

3. EXPERIMENTAL RESULTS

In this section we analyze each step of the proposed system using adaptive Gaussian model, the implications of use the shadow removal method proposed, the addition of morphological operators and the final identifications of the vehicles and their tracking, estimating their speed. In the experiments, we consider the CDNET dataset which offer different datasets to test object tracking approaches and we selected three of them related to road traffic: **Highway** baseline dataset that consist in 1700 frames where we use the frames 1050 to 1350 to evaluate. **Fall** dynamic background dataset of 4000 frames, where we use the frames 1460 to 1560. **Traffic** camera jitter dataset of 1570 frames, where we use the frames 950 to 1050.

In order to evaluate our system, we use a first stage using the ground truth provided by the datasets, we evaluate pixel a pixel our results, taking into account the shadows as a false positive, obtaining F1-score and AUC. Then, a last stage consist in a qualitative evaluation where the object tracking is performed using a bounding box that include the vehicle tracked, a vehicle counter and the speed estimation.

The first stage results, are showed in table 1, where we can see the differences between the datasets, where the adaptive model works better in fall, that have dynamic background and have more difficulty in a dataset with camera jitter, even applying the video stabilization first. Then, analyzing the different methods, we can see how shadow removal provoke a drop in the scores due the aggressive removal that affect also part of the actual vehicle in the mask, making the pixel by pixel evaluation worst. Applying only morphology operators, we achieve a good increasing in the scores for each dataset, showing the necessity and robustness that this method apply in the masks.

Finally, the second stage results is a qualitative analysis where we can see in the figure 2, the bounding box covering the vehicles tracking, showing the speed estimation and counting. For the CDNET datasets, the results are positive, performing the tracking properly in practically all the sequences, with minor outliers. But, in our own video, that was took at UAB at evening, the shadows are really huge and we need to use the shadow removal approach, that it works but not really well, making really difficult the tracking and in show frames, the vehicles are splitted due the aggressiveness of the method. Then, shadow removal method need to be improved.

4. CONCLUSIONS

We proposed a system that detect and track vehicles in a road and estimate their speed using vision-based techniques and a low cost monocular camera. We analyzed the different techniques applied and the final results obtained for different datasets in a qualitative analysis. Our system works and is useful as we can see in the results section if we avoid undesirable conditions, like too high brighten scenarios or daytime with high shadow projections, were the performance decrease.

As a future work, is desirable to improve some simple techniques used in order to increase the robustness of the system, like shadow removal that the method is not enough good and the video stabilization algorithm to extend the functionality of the system when the recording have huge camera shakes.

5. REFERENCES

[1] Abbas Dehghani Arash Gholami Rad and Mohamed Rehan Karim, "Vehicle speed detection in video image sequences using cvs method," .

[2] Dolley Shukla and Ekta Patel, "Speed determination of moving vehicles using lucas- kanade algorithm," .

[3] Chris Stauffer and W Eric L Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* IEEE, 1999, vol. 2, pp. 246–252.

[4] Ahmed M. Elgammal, David Harwood, and Larry S. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision-Part II*, London, UK, UK, 2000, ECCV '00, pp. 751–767, Springer-Verlag.

[5] Shailaja Surkutlawar, Prof Ramesh, and K Kulkarni, "Shadow suppression using rgb and hsv color space in moving object detection," .

[6] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung, "Video stabilization using robust feature trajectories," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1397–1404.

[7] Rudolph Emil Kalman et al., "A new approach to linear filtering and prediction problems," 1960.

[8] Pierre Del Moral, "Non-linear filtering: interacting particle resolution," *Markov processes and related fields*, vol. 2, no. 4, pp. 555–581, 1996.

[9] Martn Matilla Jordina Torrents Mnica Alfaro, Andrea Calafell, "Beyond a smart video surveillance approach for road traffic monitoring," 2016.