VIDEO SURVEILLANCE FOR ROAD TRAFFIC DETECTION

Adriana Fernández, Gerard Martí, Eric López, Magí Andorrà

Master in Computer Vision
Centre de Visió per Computador, Barcelona

ABSTRACT

Road traffic detection and monitorization is an important problem in object tracking. In this paper we tackle the problem of tracking vehicles that are moving along a road. We propose a model based on foreground segmentation using and adaptative gaussian grayscale model, with postprocessing hole filling, area filtering and morphological closing. Then, after a block-matching based stabilization, we use a Kalman filter and a particle filter to detect the vehicles, with a speed estimation based on linear regression. Our model achieves good results both in background segmentation, with an average AUC of 0.83, and in traffic tracking, with great visual results both in vehicle counting and speed estimation.

***Index Terms*—** Road monitoring, Background modeling, Background Estimation, Foreground Segmentation, Video Stabilization, Region tracking.

1. INTRODUCTION

Detecting moving vehicles from a video source has many interesting applications. Detecting and tracking a moving object in a video sequence is not an easy task, but thanks advances in technology, and large amount on related research, several methods for solving this problem have been proposed. Due to the scope of this paper, it is unfeasible to do an in depth review of the state of the art in the field. A good overview of research in urban traffic can be found in [1]. More recently, in [2], Li, X et al. presents a good method for traffic detection and tracking using a particle filter. We will use a similar, but simpler approach later on.

In this paper we will try to implement Kalman filter [3] for object tracking, applied to road traffic. Kalman filters for vehicle tracking has already been proposed [4] [5]. We will implement a full pipeline for vehicle tracking: computing a background estimation, using area filtering to improve the results, video stabilization, and finally region tracking using Kalman filter and speed estimation. We will also present a different approach by using a multitracking particle filter, and we will visually compare the results obtained between both methods.

2. METHODS AND ALGORITHMS

Here the algorithms and methods used for each stage of our system will be described and explained.

2.1. Background Estimation and Foreground Segmentation

To create our foreground-background segmentation system we assume that every background pixel of the video follows a **gaussian distribution**. For every pixel i we will define two values, the mean (μ_i) and the standard deviation (σ_i), following a gaussian distribution. Every pixel is labeled as foreground or background depending on the distance to the mean of our gaussian model. The threshold is controlled by the parameter α, as seen in Algorithm 1. We will take the first 50% of the sequences as a training set to compute μ_i and σ_i for every pixel.

```
forall the pixels $i$ in image $I$ do
    if $|I_i - \mu_i| \geq \alpha * (\sigma_i + 2)$ then
        $I_i \rightarrow$ Foreground;
    else
        $I_i \rightarrow$ Background;
    end
end
```

Algorithm 1: Foreground Segmentation

2.1.1. *Adaptative modeling*

To improve and extend our model, we implemented an **adaptative** modeling. After the initial training, our model adapts to each frame and updates the values of the gaussians in the pixels, following Algorithm 2. The degree of adaptation of the model is controlled by the parameter ρ.

2.1.2. *Stauffer and Grimson*

Stauffer and Grimson model [6] is an adaptive background mixture model very robust to illumination changes and to repetitive motion from clutter. The value of a pixel is modeled as a mixture of adaptive Gaussians. This method is already implemented in the MATLAB Computer Vision Toolbox, so we tried to optimize its parameters: Number of Gaussians, Learning Rate and minimum background ratio.

```
forall the pixels $i$ in image $I$ do
    if $I_i \in Background$ then
        $\mu_i = \rho * I_i + (1-\rho) * \mu_i$;
        $\sigma_i = \rho * (I_i - \mu_i)^2 + (1-\rho) * \sigma_i^2$;
    end
end
```

Algorithm 2: Adaptative Modeling

2.1.3. *Color Extension*

We can extend the system to color images by modeling the image using one gaussian per channel and per pixel. For segmentation we do an independent decision for each channel: adaptability is independent, and a pixel is only classified as foreground if all three are classified as foreground. We implemented RGB and YUV color spaces.

2.2. Post-processing filtering

After an initial estimation and foreground segmentation, we apply several morphological filters and operations to improve the results.

2.2.1. *Morphological filter*

We considered the following filters:

- **Hole filling**: After the segmentation, some of the zones detected may have some holes of not-detected pixels. Filling those holes should improve the results. We tested 4-connectivity and 8-connectivity.
- **Area filtering**: As the objects we want to detect are quite large, we can eliminate any connected components smaller than a set threshold. The value of the threshold is very dependent of the quality of the background detection.
- **Dilate**: In some instances, vehicles were not completely detected. We implemented a dilation, but the resulting AUC was worse, so this method was discarded.
- **Close and Open**: Both morphological operators were tested. For close, the motivation was to fill some holes that may not have been filled. And for Open, the idea is to remove small objects that are not part of any foreground object. However, opening did not improve the results further than area filtering already did, so we discarded it.

2.2.2. *Shadow Removal*

Foreground-background segmentations based on moving objects have the issue of detecting shadows as foreground. To avoid that, we tried to implement the method described by R. Cucchiara et al. [7] and a RGS implementation described by Elgammal et al. [8] Those methods consists in performing a simple thresholding in HSV or RGS colorspace to remove the shadows. Shadow removal did not offer any substantial improvement, so we discarded it.

2.3. Video stabilization

Sometimes cameras are not stable due to human action or natural phenomena (like wind). This instability needs to be solved in order for our algorithms to work properly. We propose a video stabilization by **motion estimation**. The idea is to compute the optical flow in order to align the frames of a shaking video and stabilize it. As we are only dealing with fixed cameras we will choose a window in the first frame of the video and use it to align the rest of frames with respect to this one.

2.3.1. *Block matching algorithm*

For computing the optical flow, we implemented a block-matching algorithm. We decided to implement backward flow compensation. For the area of search, we decided on to an area of $3 * Block\_size$, with the original block as center. After further testing, we decided to have a $Block\_size = 16$.

2.4. Region tracking and speed estimation

As the last part of our pipeline, we want to use the segmentation of the sequence obtained to track the vehicles that appear and estimate their velocity.

2.4.1. *Kalman Filter*

For tracking, we use Kalman Filtering [3]. This algorithm is widely used on signal processing. In our case it will be very useful to soft the trajectory of our objects to track and even to keep tracking an object when it is momentously occluded. We will use a modified implementation of a Multi-Object Kalman Filter tracking [1].

2.4.2. *Speed Estimation*

For vehicle speed estimation, we base our approach on the known length of the road lines and the movement over the image. We will use a corner of the bounding box as point to track (in our case the bottom right corner as due to luminance reasons, this point is more robust to shadows), and we will measure the displacement on every frame. After this we need to convert this pixel displacement to a real world length. To

[1] Motion-Based Multiple Object Tracking http://es.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html

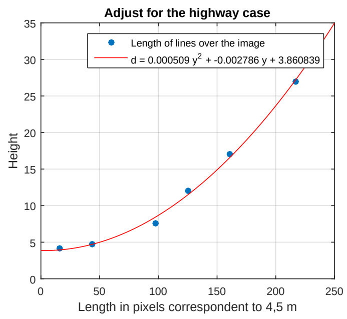


Fig. 1. Linear regression performed on the highway case.

do it we use the fact that we know that the length of the discontinuous lines is about 4.5 meters long. Therefore, we will measure the length of every line and we will plot it against the y-position over the image. We use a linear regression in order to adjust it to a 2^{nd} grade polynomial (see Figure 1). Thus, depending on the y-position of the the car to track we apply a different factor to transform from pixel length to meters.

2.4.3. Particle Filter

As an alternative method to Kalman filtering, we have implemented a Multi-Object Particle Filter algorithm [9]. This method was chosen because Kalman filter have difficulties to recover an object when it have been occluded. One of the properties of particle filter is that a lot of "particles" are tracking the same object and when it is occluded it is easier to find it when it reappears. We have modified a MATLAB implementation of Single-Object Particle Filter tracking to Multi-Object (See the link below). [2]

3. EVALUATION AND RESULTS

In this section we will explain how we evaluate the system and present the results obtained.

3.1. Datasets

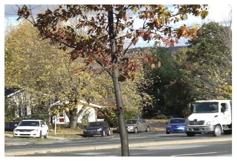
We use three different sequences[3]. We use the sequence **highway**, a baseline sequence, **fall**, a sequence with dynamic background, and **traffic**, a sequence with camera jitter. In figure 2 there is an example of a frame from each dataset.

We have also recorded two videos of vehicle traffic in a well-known highway in Barcelona, one in the morning and one at night, from the same place. This way, we will be able to assess and compare the different luminance conditions and see the impact they have in our system's performance. For testing our velocity estimator, we have recorded a car driving along a road, with a groundtruth of the real velocity.

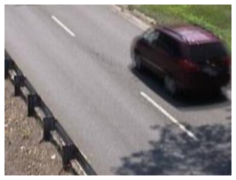
[2]Simple Particle Filter http://www.mathworks.com/matlabcentral/fileexchange/33666-simple-particle-filter-demo

[3]available in http://changedetection.net

(a) Frame from highway dataset



(b) Frame from fall dataset



(c) Frame from traffic dataset

Fig. 2. Example of frames from our datasets.

The training of parameters has been done in a small subset of each dataset. For the highway, we have done the training with frames 1050 to 1350, for fall, 1460 to 1560, and for traffic, 950 to 1050. The final results will be shown on the full sequence.

3.2. Results

For foreground segmentation, we tested the algorithms described in section 2. The parameters used for the adaptive model are shown in 1. The best results for each method can be found in table 2. The comparison between methods is done using F1-measure.

Dataset	ρ	α
Highway	0.2	2.75
Fall	0.05	3.25
Traffic	0.175	3.75

Table 1. Table of parameters for adaptive modeling

Dataset	Non-adapt.	Adapt.	S.G.	RGB	HUV
Highway	0.43	0.73	**0.79**	0.70	0.54
Fall	0.66	**0.71**	0.64	0.63	0.64
Traffic	0.47	**0.68**	0.56	0.65	0.67

Table 2. Table of results (F1-measure) for foreground segmentation

In table 3 there are the different filters applied and the improvement in each step(we compare AUC). For each method in the table: Hole filling is 4-connectivity, area filtering is 230,20 and 290 for highway, fall and traffic, respectively, and a closing of (disk,8). Each filter is applied is over the previous filter, and all the percentages are with respect to the initial version.In figure 3 the final precision recall curves are shown.

Dataset	No Filt.	Hole Fill	Area Filt.	Closing
Highway	0.6713	0.7145 (+6.4%)	0.7663 (+14.1%)	0.8050 (+19.9%)
Fall	0.7407	0.8070 (+8.4%)	0.9238 (+24.7%)	0.9446 (+27.5%)
Traffic	0.6358	0.6421 (+0.9%)	0.7267 (+14.3%)	0.7544 (+18.6%)
Average	0.6826	0.7212 (+5.4%)	0.8056 (+18%)	0.834 (+22.7%)

Table 3. Table of results (AUC) for filters used

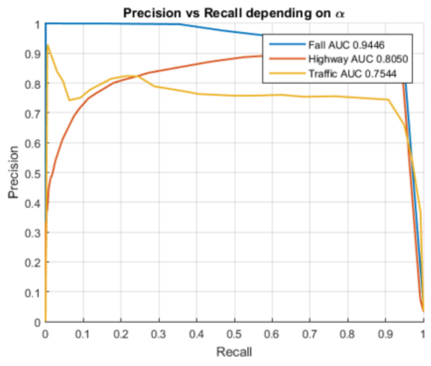


Fig. 3. Precision-Recall curves for the datasets

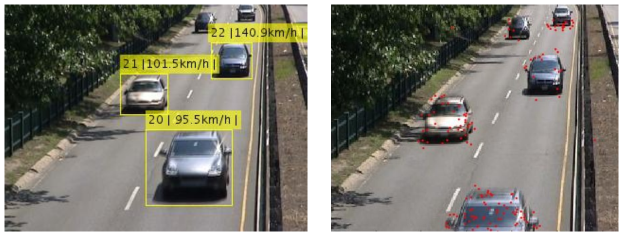
For the results obtained with the Kalman and the particle filter, we show the detection rate of vehicles against the real count. The shaky sequences have been previously stabilized by our block matching algorithm. In table 4 the results for each dataset are shown, as well as the average of the speed for all the vehicles detected for that specific dataset and the expected estimated speed for that road. The videos of the full results can be found on the website of the project. In figure 4 examples of the detection are shown.

	Highway	Traffic	RDalt-Day	RDalt-Night
gt	31	33	14	19
n^o cars	39	34	14	14
av. speed (km/h)	108.7	83.2	86.5	91.14
max speed (km/h)	100-120	70-90	80	80

Table 4. Vehicles detected for Kalman Filter

4. DISCUSSION

The method that gets best results overall is the adaptive model, over Non-adapt and Stauffer and Grimson. Regarding color model, RGB surpass HUV, but that overall the results does not show a clear improvement to justify using a color model. Regarding the filtering, we obtain a net improvement of more than 20% in average for our training datasets, which is considerably large. For the tracking, the method



(a) Kalman filter (b) Particle filter

Fig. 4. Example of Kalman and Particle filter working.

that provides a better performance is the Kalman filter. One of the restrictions of the particle filter implemented is that is very dependent on the foreground detection, same as the Kalman filter. But when objects disappear Kalman Filtering performs the best. We found that, even applying our stabilization method, tracking becomes very challenging with sequences that has a lot of jitter. The reason of that is that jitter introduce artifacts that are interpreted as foreground by the alghorithm. This was fixed partially by just showing the bounding boxes that appeared more than 3 frames.

Our ’length lines’ approach to compute the speed show really good results in almost every sequence (See again Table 4). The only problem of this approach is that will only work with a fixed camera and is scenario dependent. That last problem will be very difficult to solve without another 3D information.

Regarding luminance factors we found that when the light conditions turn darker the overall performance of the system decreases (See again Table 4), mainly due to shadows and low contrast.

5. CONCLUSIONS

Our results show that a Kalman filter with the described approach to calculate the velocity provides good results, both numerically and visually. The main limitation of the model is that it only works with a fixed camera, and with a reference in the scene which a known length. Future work could be focused on improving the segmentation (adding a good shadow removal) and updating the tracker to work with moving cameras, as well as reducing the impact that jitter has on the system.

6. REFERENCES

[1] N. Buch, S.A. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011, cited By 123.

[2] X. Li, S. Qu, and Y. Xia, “A method of multi-targets

detection and tracking from traffic video based on particle filtering," 2013, pp. 8120–8124, cited By 0.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering*, 1960.

[4] Roya Rad and Mansour Jamzad, "Real time classification and tracking of multiple vehicles in highways," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1597 – 1607, 2005.

[5] Jen-Chao Tai, Shung-Tsang Tseng, Ching-Po Lin, and Kai-Tai Song, "Real-time image tracking for automatic traffic monitoring and enforcement applications," *Image and Vision Computing*, vol. 22, no. 6, pp. 485 – 501, 2004.

[6] Chris Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," 1999, vol. 2, pp. 246–252, cited By 2317.

[7] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with hsv color information," 2001, pp. 334–339, cited By 258.

[8] 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.

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