VIDEO SURVEILLANCE FOR ROAD TRAFFIC MONITORING

Vacit Oguz Yazici, Akhil Gurram,Xialei Liu

UAB UPC UPF UPO

ABSTRACT

The goal of this project is to monitor road traffic using video surveillance techniques. The main ideas we use in this paper are: background estimation, foreground segmentation,video stabilization and region tracking.In foreground segmentation,the shadow removing technique is applied to improve the precision.In video stabilization, the block matching methods are used to compute the optical flow. In region tracking, the deep learning method are applied to track cars except using Kalman filter. The results evaluated by using pixels evaluation method which provides the AUC and F1 measurements. finally, the system we implemented can track the multiple cars in the road and draw a bounding box in each car with ID counter,and estimate the speed of car in real time. The results showed that the precision of foreground segmentation and tracking is quiet high in Highway sequence and Traffic sequence.

Index Terms— foreground segmentation, shadow removing, video stabilization, tracking, deep learning

1. INTRODUCTION

In this paper, we focus on video surveillance for road traffic monitoring, firstly, the Gaussian statistic method is used to estimate the background which involves single gaussian and Stauffer and Grimson multiple gaussian approachs. Secondly, the foreground is segmented using a threshold. Secondly, the filling hole and morphological methods are used to remove the noises and fill the hole in the foreground. Thirdly, the shadow removing method is applied to improve the precision because the shadow in ground truth seen as the background. Fourthly, the block matching method is used to estimate the optical flow which can be used as a video stabilization method. Finally, the different tracking methods are used to track the car involving the Kalman filter, Mean shift and Deep learning. And then the simple assumption is made to estimate the velocity of the vehicle.

Thanks to the master of Computer Vision.

2. BACKGROUND ESTIMATION AND FOREGROUND SEGMENTATION

The first step for tracing the car in the video sequence is background estimation, after obtaining the background, the foreground segmentation method is used to get the position of the car.

2.1. Background estimation

In this part, the Gaussian function is used to model each background pixels. the first 50 percent is used to obtain the mean and variance of the video sequence, and the second 50 percent is used to adapt the background. the background images in highway sequence before and after adapting are showed as following:

(a) Before adapting (b) After adapting

Fig. 1. The mean of the background in Highway sequence.

2.2. Foreground segmentation

In this part, the following equation is used to segment the foreground.

$$|I_i - \mu_i| \geq \alpha \cdot (\sigma_i + 2) \tag{1}$$

After segmentation, the closing and opening are used to remove the noise of small areas, and fill the holes in the foreground.

This statistics define the F1 measure assuming ground truth as foreground plus shadow region which is defined in column 4, column 5 defines the foreground region, the percentage of total number of background, foreground pixel are being properly detected with comparison of GT pixels and the last column define the noise pixel being detected.

Statistics							
This statistics is without using morphological operations							
Highway	**phi ranges from 0 to 1 diff of 0.1**				**GTPhiAlphaValLoc**		
	Mean input		F1 measure comapred with BFG	F1 measure comapred with GT	Only BG percent	Only FG percent	Only Noise pixels
	GT		0.5829/0.3/7	0.5571/0.3/7	0.5433	0.4963	40273
	BFG		0.8575/0.1/5	0.7647/0.1/6	0.9966	0.8469	15654
This statistics is with morphological operations (imclose and imopen with 7x7 square)							
Highway	**phi ranges from 0 to 1 diff of 0.1**				**GTPhiAlphaValLoc**		
	Mean input		F1 measure comapred with BFG	F1 measure comapred with GT	Only BG percent	Only FG percent	Only Noise pixels
	GT		0.6143/0.3/8	0.6314/0.4/13	0.5276	0.5282	560
	BFG		0.9088/0.2/7	0.8493/0.3/10	0.9835	0.9699	2089

Fig. 2. The result obtained in Highway

Statistics							
This statistics is without using morphological operations							
Traffic	**phi ranges from 0 to 1 diff of 0.1**				**GTPhiAlphaValLoc**		
	Mean input		F1 measure comapred with BFG	F1 measure comapred with GT	Only BG percent	Only FG percent	Only Noise pixels
	GT		0.802/0.5/6	0.7032/0.1/3	0.6082	0.7457	15748
	BFG		0.8778/0.1/3	0.7340/0.1/4	0.873	0.8394	4163
This statistics is with morphological operations (imclose and imopen with 7x7 square)							
Traffic	**phi ranges from 0 to 1 diff of 0.1**				**GTPhiAlphaValLoc**		
	Mean input		F1 measure comapred with BFG	F1 measure comapred with GT	Only BG percent	Only FG percent	Only Noise pixels
	GT		0.8898/0.8/10	0.7617/0.1/3	0.61003	0.78358	798
	BFG		0.9240/0.1/3	0.7669/0.1/3	0.9463	0.9284	1531

Fig. 3. The result obtained in Traffic

3. SHADOW REMOVING

Shadows are evaluated as part of foregrounds which cause $F1$ measure to decrease approximately 10%. To detect shadow, we used HSV color space [1]. According to that algorithm, pixel intensity of saturation and hue channels of shadow and background are approximately same. While the value channel intensity is different.

The selection of parameters depend on the environment. For example, small a value can be used for sunny environments where the intensity difference is high. To prevent division by zero, a small epsilon value was used in the first statement.

4. VIDEO STABILIZATION

4.1. Block-matching for optical flow

The block matching algorithm is used in order to estimate the optical flow of the traffic sequence. We used New Tree Step Search algorithm [2]. Given a block, the algorithm searches for a motion vector which minimizes the energy function. The window size NxN and the algorithm checks W step in each direction. According to that the energy function,

$$E(u,v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_{t-1}(i+u, j+v) - f_t(i,j)| \quad (2)$$

u,v are the motion vector between consecutive frames. We limit them to restrict the matching of the block in unrelated regions.

Fig. 4. The results of optical flow

- Size of the blocks We choose 16-by-16 window size for the block matching. It is an ideal size for not losing the information and catching the motion.
- Size of the blocks We choose 7 as an area of search. This value caused the highest $F1$ score.

The stabilization of the traffic sequence by extracting affine transformation from the optical flow (block matching).

5. TRACKING

5.1. Kalman Filter

In this part, the Kalman filter[3] is used to track the car. the model we use is 'ConstantAcceleration',so the state equation is as following:

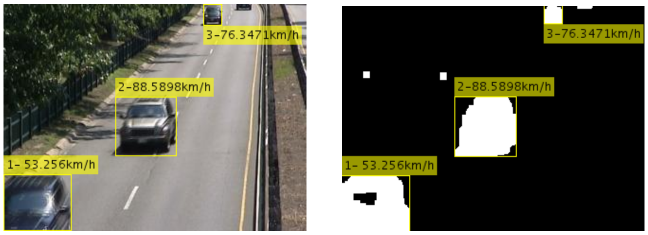
$$\begin{bmatrix} x_t \\ v_t \\ a_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_{t-1} \\ v_{t-1} \\ a_{t-1} \end{bmatrix} + Noise \quad (3)$$

The measurement equation is as following:

$$Z_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_t \\ v_t \\ a_t \end{bmatrix} + Noise \quad (4)$$

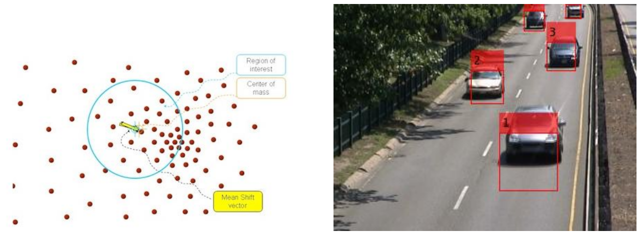
5.2. Continuously Adaptive Mean Shift

CAMS (Continuously Adaptive Mean Shift) .[4] differs from the mean shift algorithm by dynamically calculating a color distribution of an object. As all tracking algorithms, the objects have to be initialized. The initialization gives current



(a) Original image tracking (b) Foreground tracking

Fig. 5. The tracking in Highway sequence.



(a) Updating Window Centers (b)Tracking image

Fig. 6. The tracking results using CMAS

centers and window locations. After initialization, the histograms of the objects are computed. After that, for each object the probability distributions of colors are computed.

The purpose is to get the similar probability distribution for each object in consecutive frames. For that the following steps are done:

- Calculate the mean in the window. I is the probability distribution of colors for each pixel

$$M_{00} = \sum_{x \in S_i} I(x,y) \tag{5}$$

- Calculate the moment in both directions

$$M_{10} = \sum_{x \in S_i} x * I(x,y) \tag{6}$$

$$M_{01} = \sum_{x \in S_i} y * I(x,y) \tag{7}$$

- Move the window to its new location

$$x_c = \frac{M_{10}}{M_{00}} \tag{8}$$

$$y_c = \frac{M_{01}}{M_{00}} \tag{9}$$

The tracking results are showed in Fig .6.

5.3. Deep learning for tracking [5]

- Preprocessing

  The main goal of Preprocessing is to eliminate noise in the input data by applying Threshold on region Properties.

  Assuming very small objects which are less than 120 pixels to be noise. we eliminate them using connected component properties of the object from foreground extraction module.

- Detection and Tracking New object

  For the first frame we extract all the object which are in the RoI and create new track for each object. From the second frame we create a new track for the objects which doesnt exist in the previous frames and which are in the RoI. For updating the tracklist whether is there any new object being detected or any object are lost we are updating the status of each track simultaneously. We have assigned a field called continue to each track and checks whether is it true or not.

  We have given one more condition saying if the object is not in the RoI then we neglect that object as we lose that track. For each track we define the center, bounding box of the object.

- Multi object tracking using DLT

  The object to track is specified by the location of its bounding box in the first frame. Positive samples are collected from the Bounding box with a change of one pixel in all the directions and Some negative examples are collected from the background at a short distance from the object. A sigmoid classification layer is then added to the encoder part of the SDAE obtained from offline training. When a new video frame arrives, we first draw 1000 particles according to the particle filter approach. The confidence pi of each particle is then determined by making a simple forward pass through the network. An appealing characteristic of this approach is that the computational cost of this step is very low even though it has high accuracy.

  If the maximum confidence of all particles in a frame is below a predefined threshold , it may indicate significant appearance change of the object being tracked. To address this issue, the whole network can be tuned again in case this happens. We note that the threshold should be set by maintaining a tradeoff. If is too small, the tracker cannot adapt well to appearance changes. On the other hand, if is too large, even an occluding object or the background may be mis-treated as the tracked object and hence leads to drifting of the target.

We use the same process for all the object to track it in a parallel way for each frame as it doesn't have any relation to the other objects, So it is good approach for multi object tracking simultaneously, Though processing is slow. we can achieve high accuracy.

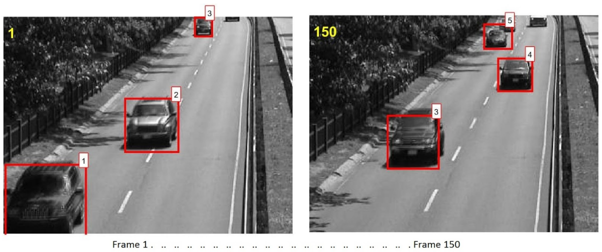


Fig. 7. The result obtained by using deep learning

- DLT Implementation Details

  We use the gradient method with momentum for optimization. The momentum parameter is set to 0.9. For online tuning, we use a larger value of 0.002 to avoid overfitting and a smaller mini-batch size of 10. The threshold is set to 0.8. The particle filter uses 1000 particles. For other parameters such as the affine parameters in the particle filter and the search window size in the other methods, we performed grid search to determine the best values. The same setting is applied to all other methods compared if applicable.

5.4. Speed estimation [6]

In this part, the following assumption is used to estimate the speed of the car. By using the Kalman filter, the velocity of the cars in the frame can be estimated but the metric is in pixel way, therefore the assumption can help us to transform the pixel into meters in real world. Considering the different resolution on the top and bottom of the images in Highway sequence, we use the function in Fig 8 to fit the distance in real world.

The Tab .1 shows that the speed estimation for different cars in both Highway and Traffic sequence. The reason why the speed varies with the frames is because of Kalman model we use and assumption.

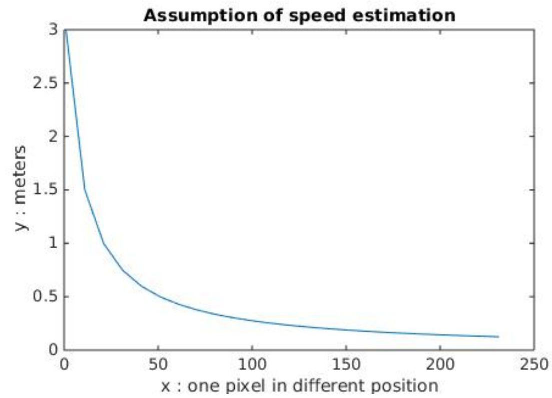


Fig. 8. The assumption for speed estimation in Highway sequence.

Table 1. The results of speeding estimation

	Speed-Highway	Speed- Traffic
Vehicle 1	60-80 km/h	30-50 km/h
Vehicle 2	60-80 km/h	30-50 km/h
Vehicle 3	40-60 km/h	–
Vehicle 3	40-60 km/h	–
Vehicle 3	60-80 km/h	–

6. CONCLUTIONS

In speed estmation part, if we have a marker in the image, it will be more accurate.

7. REFERENCES

[1] Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Prati, and Stefano Sirotti, "Improving shadow suppression in moving object detection with hsv color information," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 334–339.

[2] Renxiang Li, Bing Zeng, and Ming L Liou, "A new three-step search algorithm for block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 4, no. 4, pp. 438–442, 1994.

[3] Gene F Franklin, J David Powell, and Michael L Workman, *Digital control of dynamic systems*, vol. 3, Addison-wesley Menlo Park, 1998.

[4] Jicai Ning, Leiqi Zhang, Dejing Zhang, and Chunlin Wu, "Scale and orientation adaptive mean shift tracking," *Computer Vision, IET*, vol. 6, no. 1, pp. 52–61, 2012.

[5] Naiyan Wang and Dit-Yan Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, 2013, pp. 809–817.

[6] Daniel J Dailey, Fritz W Cathey, and Suree Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 98–107, 2000.