MSFT lab - Background Subtraction

Yannick PORTO

***Index Terms*—Tracking methods, Background Subtraction, Gaussian, Eigen**

I. Introduction

THE goal of this *Visual Tracking* module is to learn about, and, more importantly, to learn how to use basic tracking algorithms and evaluate their performances.
A very simple and effective technique is called *Background Subtraction* and can be used to initialize the tracker, i.e. to find the target's position in the first frame of a sequence, or to track the target through this sequence. In this lab, we will see the differences between three Background Subtraction methods : Frame differencing, Running average gaussian and Eigen background.
These methods are applied here on two different sequences. In order to visualize the effects of the algorithm, we first test it on a sequence with few frames, only one car to detect and two persons walking. And a numerical comparison is done with a sequence including multiple cars and a groundtruth.

II. Frame differencing

In this algorithm, the background is modeled with the median of the previous frames :

$$B_i(x,y) = median(I_{i-n+1}(x,y), I_{i-n}(x,y), ..., I_{i-2}(x,y), I_{i-1}(x,y)) \quad (1)$$

A pixel belongs to the foreground if

$$|I_i(x,y) - B_i(x,y) > T| \quad (2)$$

where T is a defined threshold. We used in this experiment the global image threshold using Otsu's method.

When the sequence of images starts, only few frames are used to compute the background. For instance, when the algorithm is trying to detect the car in the frame 3, the median computed from the only first and second frames is taken as the background. What gives a blurred shape in the background estimation.

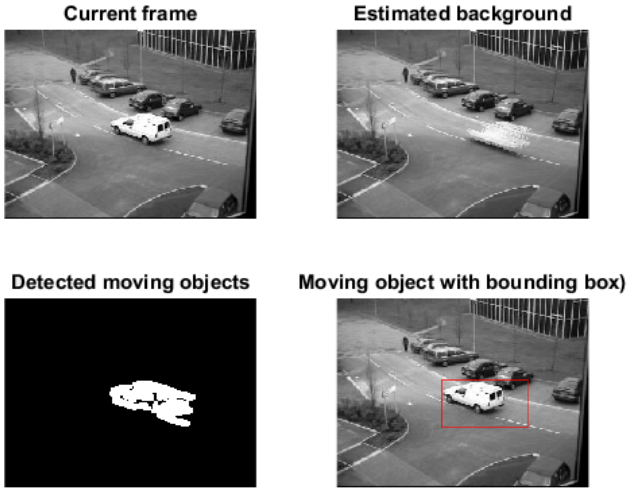

Fig. 1. Car detection in the first frames with the frame differencing method

The blurred remains of the car come to disappear after some time and the background is properly estimated without any noise.

An opening morphological operation helps to improve the result of the detected foreground. A small rectangle is actually used for the erosion in order to remove the noise, what is followed by a dilatation processed with a bigger rectangle for merging the different part of the car.
The bounding box is drawn by detecting the biggest area in the foreground image.

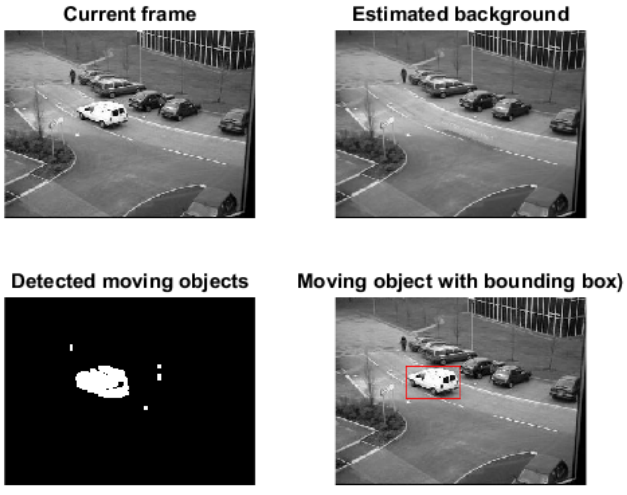

Fig. 2. Proper detection of the car with the frame differencing method

Unfortunately the algorithm and the preprocessing opening operation leads to merge the too close moving shapes.

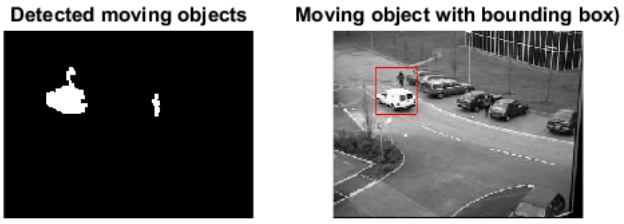

Fig. 3. Detection of the moving person as the car with the frame differencing method

III. Running average Gaussian

The running average Gaussian models each pixel's recent history as a Gaussian probability density function with a mean μ_t and a variance σ_t^2 :

$$\begin{aligned} \mu_t(x,y) &= \alpha I_t(x,y) + (1-\alpha)\mu_{t-1}(x,y) \\ \sigma^2(x,y) &= d^2\alpha + (1-\alpha)\sigma^2_{t-1}(x,y) \end{aligned} \tag{3}$$

where $d = |I_t(x,y) - \mu_t(x,y)|$, and α determines the size of the temporal window that is used to fit the pdf.

This method behave a bit as the previous one at the beginning. It takes a large bounding box because of the remains of the car from the previous frames for which the background pixels are poorly estimated due to their tiny history.

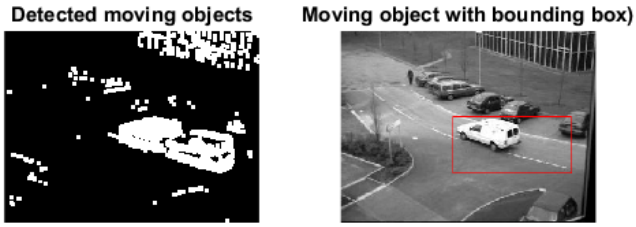

Fig. 4. Car detection in the first frames with the running average gaussian method

The difference in the results come at the end of the sequence, when the car meet the moving person. Those one are not merged anymore and the bounding is properly fitting the car thanks to an individual estimation of the pixels.

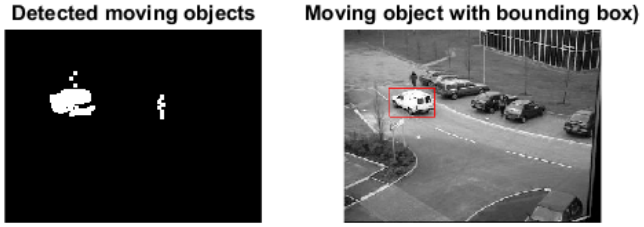

Fig. 5. Detection of the car without the moving person with the running average gaussian method

The pixels are here classified according to their intensity value with regard to a confidence interval of its distribution's mean. Changing the threshold value is really sensitive and leads to really different results.

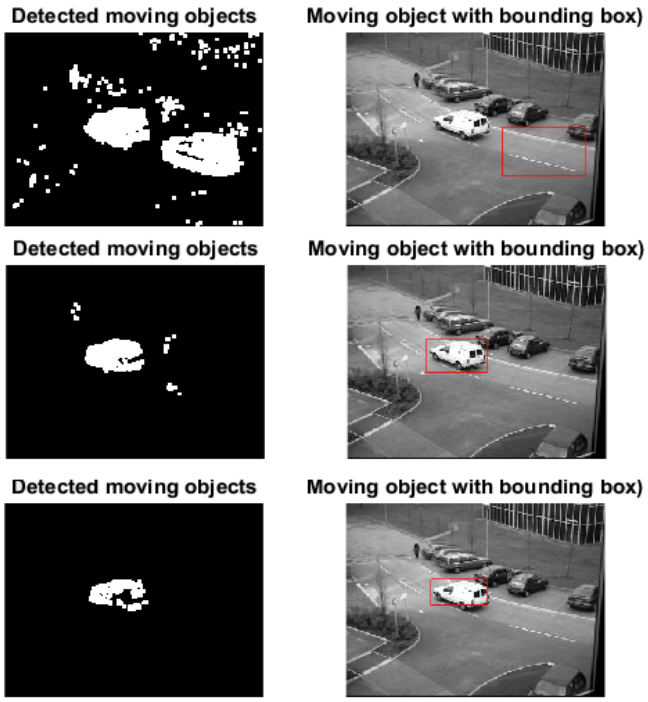

Fig. 6. Comparison of the results according to the threshold value (first row T = 1.5, second row T = 2.5, third row T = 3.5) with the running average gaussian method

By taking the pixels above a small threshold, the foreground gives a lot of importance to the previous frames to such a point that it detects the car where it was at the beginning. A too strict threshold provides a really narrow bounding box of the detected car. The value of 2.5 seems to be a good trade off between both.

IV. Eigen Background

This method uses an eigenspace that models the background. It first build mean-normalized image vectors in a matrix *X*, where each image is substracted by a mean image *m* computed from a sample of N images.

$$m = \frac{1}{N} * \sum_{i=1}^{N} x_i \tag{4}$$

$$X = \begin{bmatrix} x_1 - m & x_2 - m & \dots & x_N - m \end{bmatrix} \tag{5}$$

A singular value decomposition (SVD) is then performed on this matrix X.

$$X = U\Sigma V^T \tag{6}$$

And a projected image is computed thanks to the principal component analysis (PCA) by keeping the first k columns of U.

$$\hat{y} = U_k p + m \tag{7}$$

with $p = U_k^T(y - m)$

The foreground is therefore given by all the absolute difference between the input image and the projected image that are higher than a certain threshold T.

$$|\hat{y} - y| > T \tag{8}$$

This method provides a correct detection of the car, without taking into consideration the moving persons. The result is however quite sparse and does not return a single and proper shape, what result in a detected car of split regions. Therefore, only the back of the car is often surrounded by the bounding box.

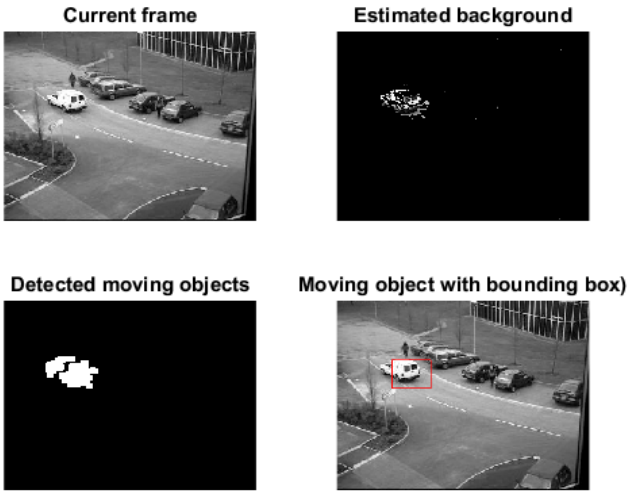

Fig. 7. Car detected in the last frame with the eigen background method

V. Comparisons

All the comparison numbers have been computed from the second sequence which is an highway, and especially at the end of sequence where we can see a lot of cars.

Precision and recall have been computed thanks to the ground truth. The following results are the one obtained on a same image with the thresholds giving the best F-score :

	$Precision$	$Recall$	$Fscore$
$F.diff.$	0.86	0.73	0.79
$R.A.Gaussian$	0.77	0.78	0.77
$EigenB.$	0.68	0.66	0.67

No post processing have been used so that it does not confuse the results of the method itself.

The First and second method show a bit more advanced score than the last one. It can be explained by the fact that they are making computations at every frame in order to model the background whereas the eigen background method makes the estimation with a certain number of frames and apply it directly to the frame we want, without any update.

Precision and recall have also been computed with a moving threshold so that we check the algorithm is behaving continuously :

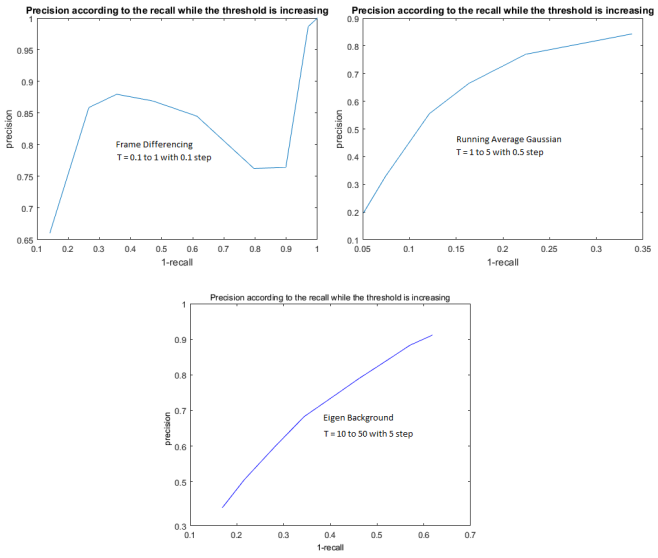

Fig. 8. Precision according the recall for all the methods while the threshold is changing. Top Left = Frame differencing with T = 0.1 to 1. Top Right = Running Average Gaussian with T = 1 to 5. Bottom = Eigen Background with T = 10 to 50.

A observe a proper linearity in the behavior of the second and third algorithm according to their threshold, while the first one is showing some discontinuity in its thresholding.

We can therefore deduce than the frame differencing method shows good scores when it uses an optimized threshold (for example with the help of the Otsu's method) but is not really stable when using an other threshold. The running Gaussian average method stay quite stable and show good precision and recall. It also output the result quite rapidly thanks to the few number of computations. At last, the eigen background method appear to be really continuous in its thresholding but shows a bit less precision than the other methods. It is also quite long to compute the background model from the initial frames with the svd technique.