# Image Analysis: Mean-Shift Tracking

Guillaume Lemaître
*Heriot-Watt University, Universitat de Girona, Université de Bourgogne*
g.lemaitre58@gmail.com

## I. Introduction

Visual tracking is a very wide field in the computer vision domain and leads to many applications in the real world. In order to be able to apply these tracking methods, they will have to be fast enough in order to solve real-time problems and have to be robust. Tracking methods can be classified in two different types: *Filtering and Data Association* and *Target Representation and Localization*.

The first type of methods refers to all different filters methods: Bayes Filter, Kalman Filter, Extended Kalman Filter, Histogram Filter, Particle Filter or also Monte Carlo methods. The choice of the filter depends on the hypothesis of the framework. However, all filters have the same common idea. An estimation of position of a target at time $t$ have to be computed knowing the previous position of the target at time $t-1$ and given some measurements.

The second type of methods considers a different approach. The basic idea is to maximize a likelihood function to find the ”best” candidate which is matching with a target model.

Mean-Shift tracking algorithm belongs to the last kind of visual tracking methods. A target model is defined by its probability density function. Knowing the position of the target at time $t-1$, Mean-Shift algorithm [1], [2], [3] will allow to find the target candidate which will have the most similarities with the target model. The different steps to find the likelihood will be presented more in details in the followings parts.

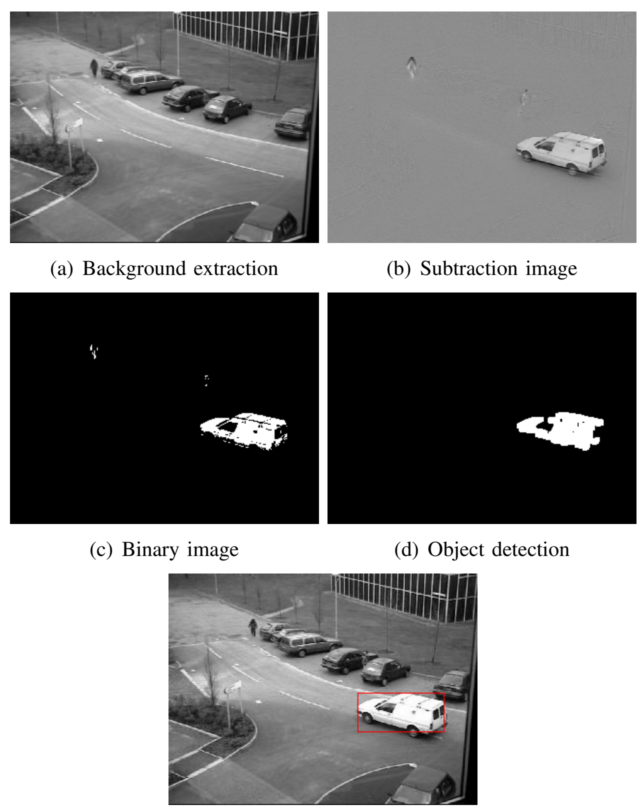
In this paper, we will present in details the Mean-Shift tracking but on a practical fashion. In the first section, we will present the pre-processing stage which allows to detect the target model. In the second section, we will present the Mean-Shift tracking where we will introduce the algorithm dedicated to grayscale and the extension for color images. In the fifth section, we will present two different tools which allow Mean-Shift algorithm to be more flexible. First we will present an improvement to deal with varying size of target and then a combination with the Kalman filter framework [1], [4] in order to deal with occlusions.

## II. Pre-processing

In this section, we will present how we detect the target model in order to track it in a sequence of images. The overview of the algorithm for the pre-processing stage is as follow:

**Algorithm 1** Detection of the Target Model

Convert all images from color space to grayscale
Background image $\leftarrow$ *Median Filter*(grayscale images)
Substraction image $\leftarrow$ First image $-$ Background image
Binary image $\leftarrow$ *Thresholding*(Substaction image, Otsu's threshold)
Erode image $\leftarrow$ *Erode*(Binary image)
**repeat**
  Tempory image $\leftarrow$ Binary image
  Dilate image $\leftarrow$ *Dilate*(Tempory image)
  Tempory image $\leftarrow$ Dilate image
**until** number of objects $> 1$
Object image $\leftarrow$ Tempory image



(a) Background extraction
(b) Subtraction image
(c) Binary image
(d) Object detection
(e) Object detected on real scene

Figure 1. Different steps to extract the object from the first image of the sequence

### A. Background extraction

The first step is to extract the background. The following steps will be achieved:

- Convert images of the sequence in grayscale space.
- For each pixel, compute the median value along time.

Assuming that the background does not change during the time, we can assume that the median value of a pixel along the time should be representative of the value of the background which is static. The result of this simple operation is shown on figure 1(a).

### B. Subtraction of the background to the first image of the sequence

Assuming again that the background is not changing during the time, a simple subtraction will allow to find roughly the object. The result of the subtraction is presented in the figure 1(b).

### C. Thresholding and morphological operations

*1) Thresholding using Otsu's method:* The image obtain after the subtraction of the background (figure 1(b)) is not a binary image due to the noise providing by the acquisition process. In order to detect the main objects of the scene, a thresholding will be applied. Otsu's method was used to find a threshold which will minimize the intraclass variance of the two classes object and not object. In the figure *fig1c* presents the result after thresholding the image obtained in the section II-B.

*2) Detection of the object applying morphological operations:* As shown on figure 1(c), after the thresholding, some false alarms can be detected. Assuming that the spatial size of these false alarms is is quite small; we will apply a morphological operation to discard these points. Moreover, the main object cannot be detected directly due to its non homogeneity. Erode and dilate operations will be used to solve these problems. The algorithm will be as follow:

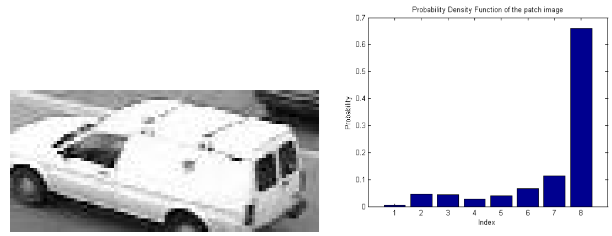
- Erode the image to remove the small alarm
- Dilate the image until to obtain only one object on the image.

Results are shown on figure 1(d). The detection on the real image is shown on figure 1(e).

## III. Mean-Shift tracking

### A. Mean-Shift tracking for grayscale images

*1) Target representation:* As said in the introduction, Mean-Shift tracking is based on maximizing a likelihood function which means that the aim is to find a candidate target which has the most similarities with a model target. In this first part, the Mean-Shift approach to characterize a target will be presented. A target can be characterized by




(a) Background extraction (b) Subtraction image

Figure 2. Example of target and associated density probability function computed

his color distribution. The number of bins of the probability density function will be reduced to a number chosen. A typical value is $m = 8$. According to [1], [2], [3], it is not a simple color distribution which will be computed. In fact, instead of checking the intensity of the pixel and increment the corresponding bin of the pdf, we will increment the bin by a value depending of the distance between the pixel and the center. Notice that locations pixel normalization is done by the size of the target model founded in the previous section. Two parameters are defined $h_x$ and $h_y$ which are defined respectively as half of the width and height of the target model. The color distribution can be formalized as:

$$h_u = C \sum_{k=1}^{n} k(\|\mathbf{x}_i^2\|)\delta[b(\mathbf{x}_i - u)] \quad (1)$$

where $\mathbf{x}_i$ is the normalized pixel location, $b(\bullet)$ is the bin corresponding to the intensity of the pixel and $C$ is the normalization coefficient to impose the condition $\sum_{u=1}^{m} h_u = 1$ hence this coefficient is defined as:

$$C = \frac{1}{\sum_{k=1}^{n} k(\|\mathbf{x}_i^2\|)} \quad (2)$$

In the equation 1, the value of the bin is weighted by a kernel function $k$. In our experimentation and as presenting in [1], [2], [3], a kernel with Epanechnikov is used:

$$k(x) = \begin{cases} 2\pi(1-x) & \text{if} \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Figure 2(b) shows a representation of the pdf of a target in the case of a grayscale image.

*2) Target Localization:* After defined how to describe a target, the problem is to find a target candidates which is similar to the target model. This problem can be solved as shown on [1], [2], [3], maximizing the Bhattacharya coefficient. The algorithm which allows the optimization of the Bhattacharya coefficient is presented in details in [1], [2], [3].

### B. Mean-Shift tracking for color images

The algorithm for color images is just an extension of the grayscale algorithm. The difference between color and

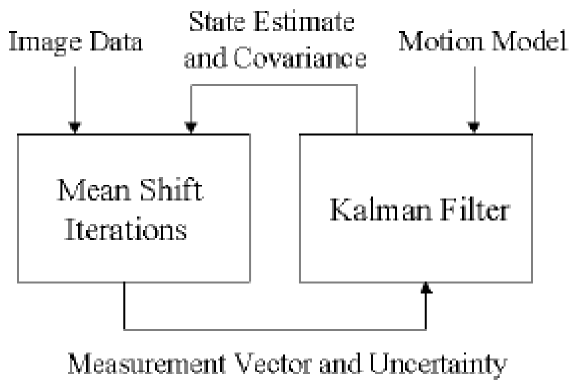



Figure 3. Framework integrated Kalman filter

grayscale implementation is regarding the probability density function. The complexity of the histogram is then $m \times m \times m$ and can be represented as a cube.

Any types of color space can be used as RGB, HSV, La*b*.

## IV. Tools to improve Mean-Shift tracking

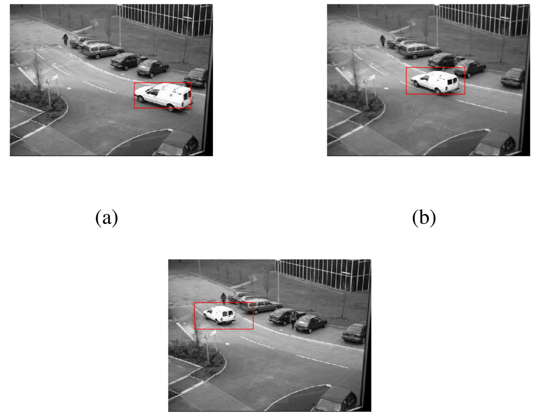
### A. Adaptive scale

As presenting inside the papers [1], [2], [3], an adaptation of scale can be performed between two successive frames. For each frame, the Mean-Shift algorithm is running three time but each time with different parameter $h_x$ and $h_y$. At the first iteration of the algorithm, the size will be reduced by 10 percent. At the second iteration, the size will be the same than at the previous iteration. Finally, the size will be increased by 10 percent at the third iteration. The largest Bhattcharya coefficient will give the scale at the next image noted $h_{new}$. However, to avoid oversensitive scale adaptation, the scale will be computed as follow:

$$h_{opt} = 0.1 h_{new} + 0.9 h_{prev} \tag{4}$$

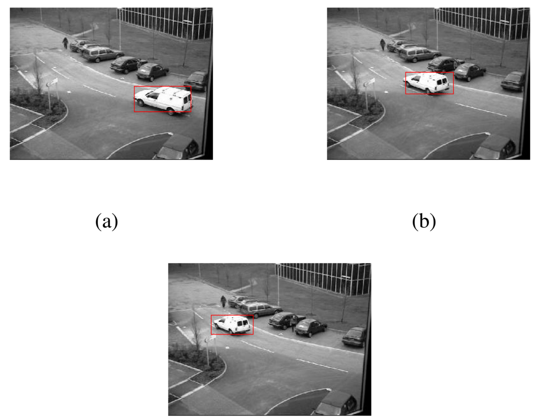
### B. Integration inside Kalman Filter framework

As proposed in [2], [3], Mean-Shift algorithm can be integrated inside a Kalman filter framework which will allow mainly to avoid problems with occlusions. The idea of the organization is presented in figure 3. The basic idea is to use the Kalman filter to make the prediction. Then this prediction will be used as starting point of the Mean-Shift algorithm. Thanks to the Kalman filter prediction, Mean-Shift algorithm will converge faster which will allow to speed-up the algorithm. Mean-Shift will be used as measurements to update the Kalman filter. Kalman will take in accounts these measurments only when the value of the Bhattacharya coefficient will be large enough. Inside our implementation, the threshold was put at $0.5$.



(a) (b) (c)

Figure 4. Car sequence using Kalman without scale adaptation



(a) (b) (c)

Figure 5. Car sequence using Kalman with scale adaptation

## V. Results

### A. Car sequence

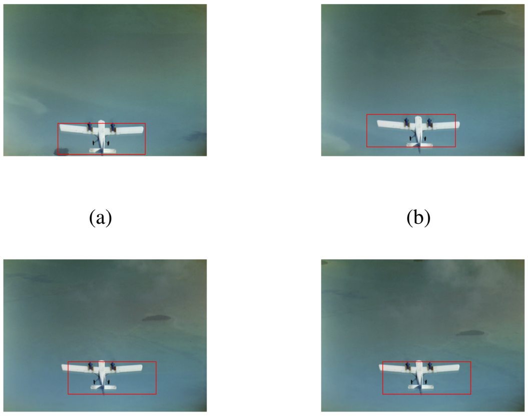
The interesting result of the car sequence is to see the importance of the scale adaptation. At the end of the second sequence, we can observe that the bounding box is covering only the car and not as in the first sequence, a part of the background.

### B. Plane sequence

In the plane sequence, we can observe the importance of the object detection. Considering the target as rectangle, we include a part of the background inside the target model. In the case of scale adaptation, it leads to a overcalling effect and bad tracking.

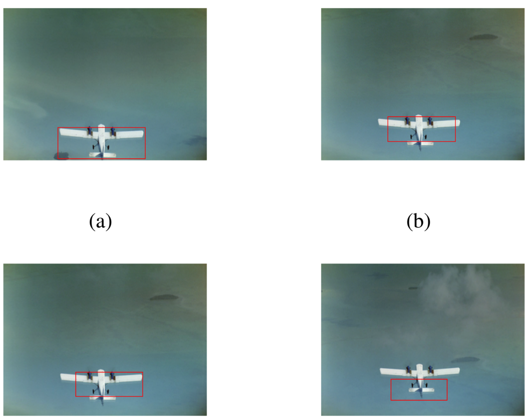
### C. Toy sequence

On the toy sequence, we observe the importance of the Kalman filter which allow to not loose the target after passing



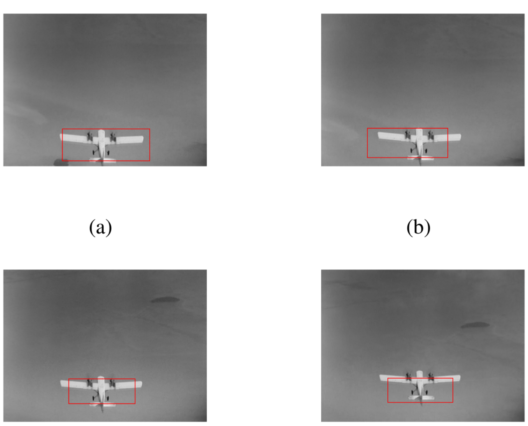
(a) (b) (c) (d)

Figure 6. Plane sequence using Kalman without scale adaptation in RGB



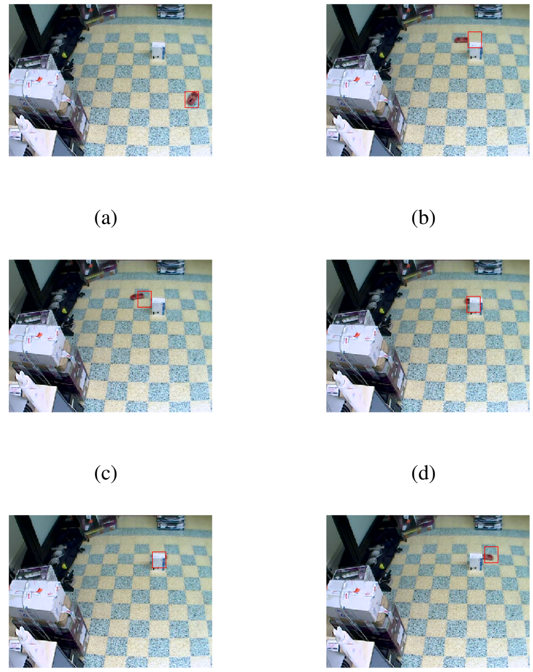
(a) (b) (c) (d)

Figure 7. Plane sequence using Kalman with scale adaptation in RGB



(a) (b) (c) (d)

Figure 8. Plane sequence using Kalman with scale adaptation in HSV



(a) (b) (c) (d) (e) (f)

Figure 9. Toy sequence using Kalman without scale adaptation in RGB

the white box. In the normal Mean-Shift algorithm, it is impossible to track an object which it is not appearing in the image. Kalman can make estimation at the moment where Mean-Shift is not working

## VI. Conclusion

We present on this report an implementation of the Mean-Shift tracking. We present this algorithm inside a Kalman filter framework which allow to fix the problem leads by occlusions.

## References


[1] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *Tracking, International Conference on Image Processing*, 2000, pp. 70–73.

[2] D. Comaniciu, V. Ramesh, P. Meer, S. Member, and S. Member, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–577, 2003.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Tracking, International Conference on Image Processing*, 2000, pp. 142–149.

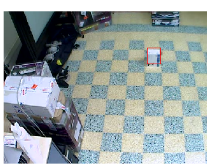
[4] Y. Bar-Shalom, *Tracking and data association*. San Diego, CA, USA: Academic Press Professional, Inc., 1987.

(a)

(b)

(c)

(d)



(e)

(f)

Figure 10. Toy sequence using Kalman with scale adaptation in RGB