VISION COURSE
Pratical work – Object Tracking in Videos

Victor PIRIOU
Nathan GALMICHE

1 Mean Shift

Q1

The Mean-Shift algorithm seeks *modes* or local maxima of a density function given a discrete set $S = \{\mathbf{x}_i\}_i \subset \mathcal{E}$ of data that are assumed to be sampled from that function.

In order to find the mode of a new point $\mathbf{x}$, the method starts from it and, at each iteration, computes the gradient that it has to follow in order to get closer to a local maxima. The latter is finally defined as the mode of $\mathbf{x}$.

The gradient v of the first iteration is given by:

$$v(\mathbf{x}) = m(\mathbf{x}) - \mathbf{x},$$

where m is mean of the density in the window that is weighted by a kernel $K : \mathcal{E} \longrightarrow R_+$ which is typically a Gaussian kernel. The mean is defined by:

$$m(\mathbf{x}) = \frac{\sum_{x_i \in N(\mathbf{x})} K(x_i - \mathbf{x}) x_i}{\sum_{x_i \in N(\mathbf{x})} K(x_i - \mathbf{x})}.$$

Here N denotes the neighbours of $\mathbf{x}$ that are within the window. Once it is computed, this gradient is used to update $\mathbf{x}$:

$$\mathbf{x} \leftarrow m(\mathbf{x}) + \mathbf{x}$$

The algorithm keeps iterating until it converges to the position, *i.e.* $\mathbf{x} = m(\mathbf{x})$.

The advantages of this algorithm are the following:

- it does not assume any predefined shape on data clusters.
- it is capable of handling arbitrary feature spaces (HSV, embedding produced by neural networks, etc).
- unlike the multiple hypothesis algorithm (MHT) or the Hungarian method, that are some of the most used methods to link detections, the Mean-Shift algorithm allows an *online* tracking.

The drawbacks of this algorithm are the following:

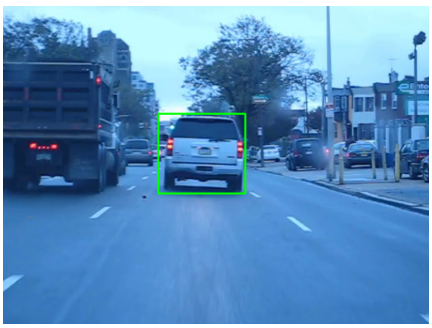
- the selection of the window's size is not trivial because it should be adaptive due to change of scale.
- the performance depends on the quality of the extracted features that have to be robust to small perturbations (*e.g.* blur due to sudden movement, acquisition noise, etc) and changes (*e.g.* rotation, deformation, etc).
- when there is a lack of information (for instance due to poor contrast, occlusion or the high speed of the movement), the object to be tracked can get lost.

(a) First frame

(b) A frame taken halfway through the video

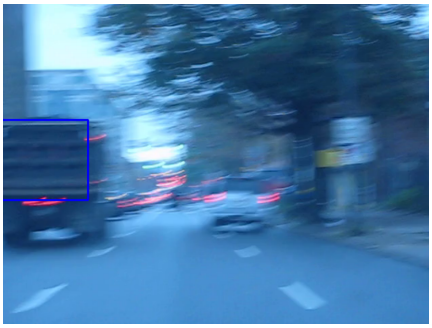
(c) One of the last frames

Figure (1) **Example that illustrates the fact that the algorithm cannot manage occlusion and that sometimes the window's size should not be fixed.** In this case, the movement of the player changes the distribution within the window.



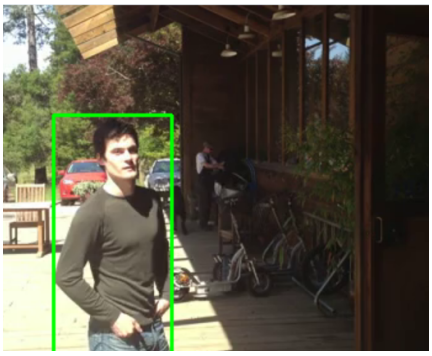
(a) First frame

(b) A frame taken halfway through the video

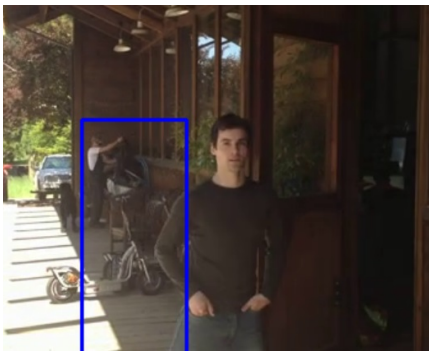


(c) One of the last frames

Figure (2) Example that illustrates the fact that the algorithm is not robust to shaking.



(a) First frame



(b) A frame taken halfway through the video

(c) One of the last frames

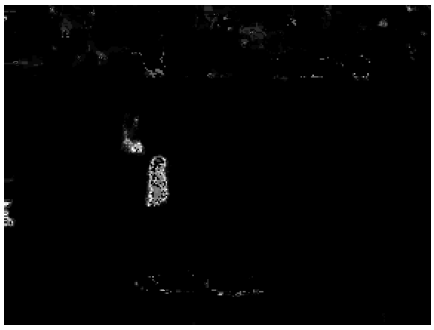
Figure (3) Example that illustrates the fact that the algorithm does not work well when the contrast is poor.

(a) First frame

(b) A frame taken halfway through the video

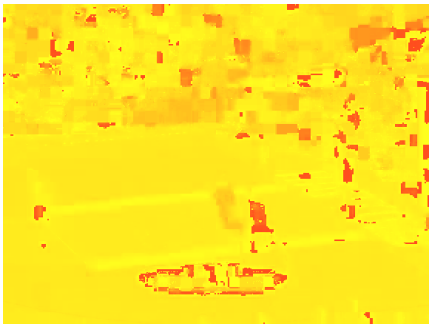
(c) One of the last frames

(d) Weight image corresponding to the back-projection R_H



(e) Weight image corresponding to the back-projection R_H

(f) Weight image corresponding to the back-projection R_H

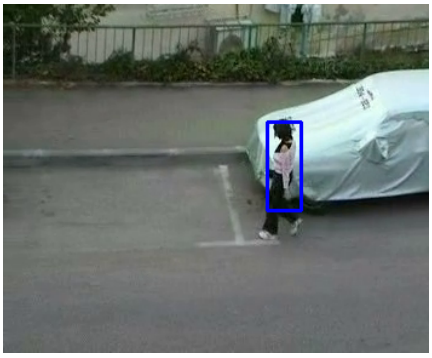


(g) Hue histogram f_H

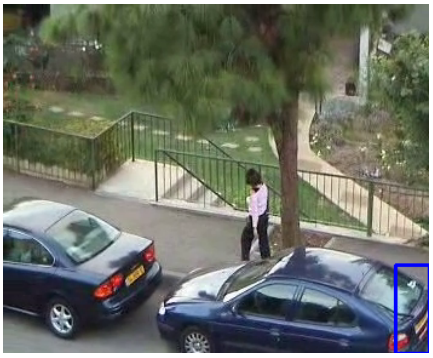
(h) Hue histogram f_H

(i) Hue histogram f_H

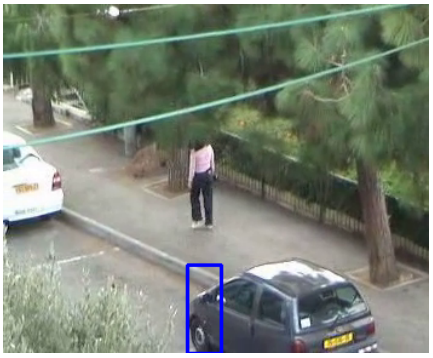
Figure (4) **Example of tracking that succeeded.** Each row of images corresponds to a particular time step.



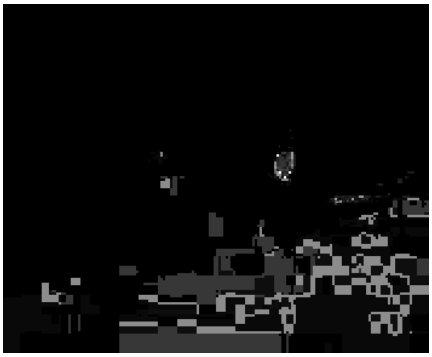
(a) First frame



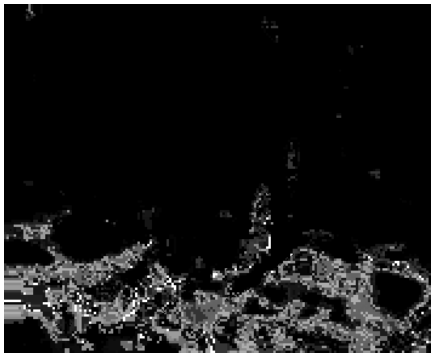
(b) A frame taken halfway through the video



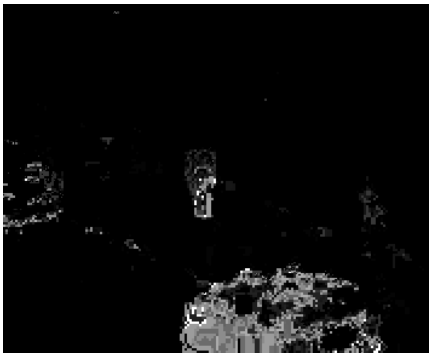
(c) One of the last frames



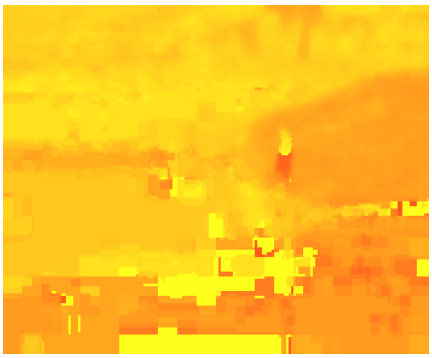
(d) Weight image corresponding to the back-projection R_H



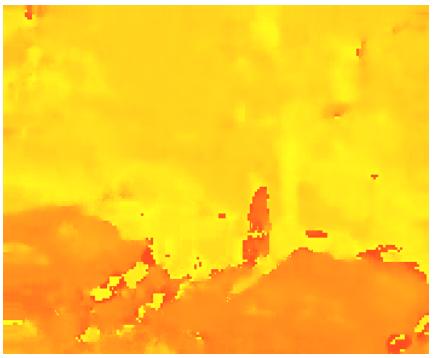
(e) Weight image corresponding to the back-projection R_H



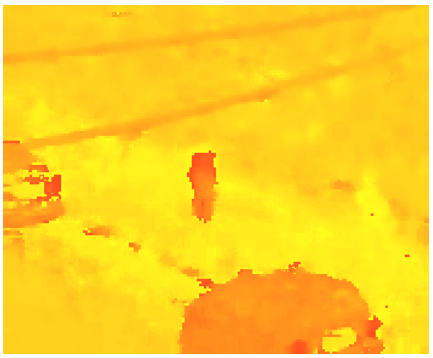
(f) Weight image corresponding to the back-projection R_H



(g) Hue histogram f_H



(h) Hue histogram f_H

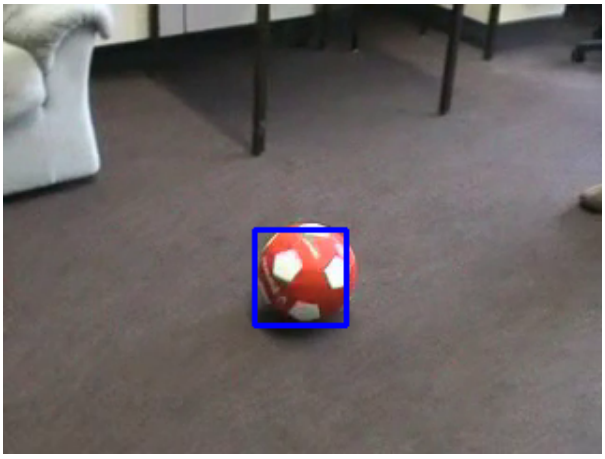


(i) Hue histogram f_H

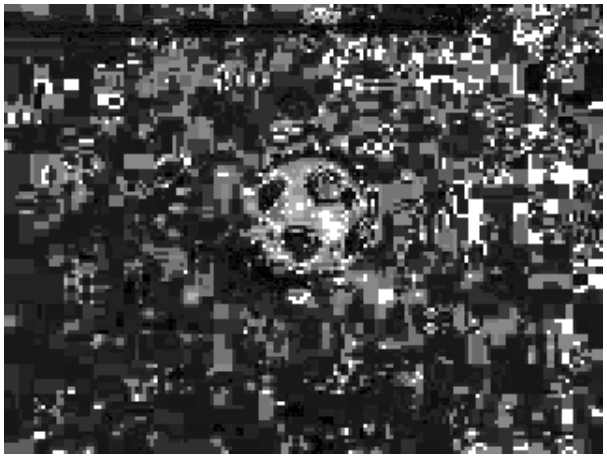
Figure (5) **Example of tracking that failed.** Each row of images corresponds to a particular time step.

Q2

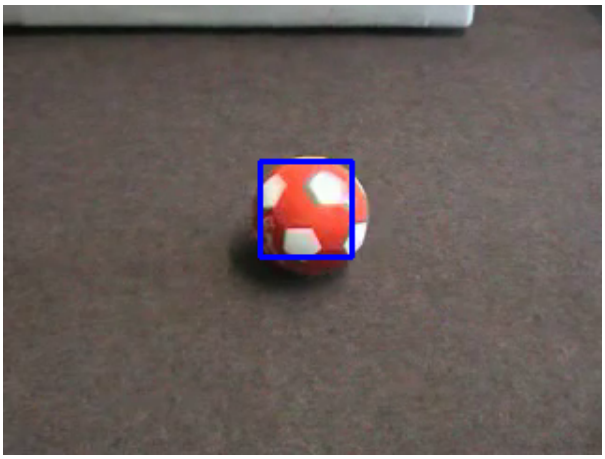
In order to combine the three weight maps computed from the H, S and V components, we performed mean and max pooling.



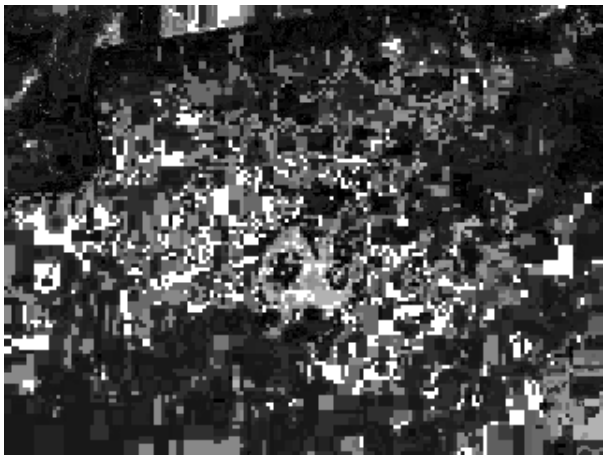
(a) One of the first frames



(b) Backpropagation map obtained with max-pooling

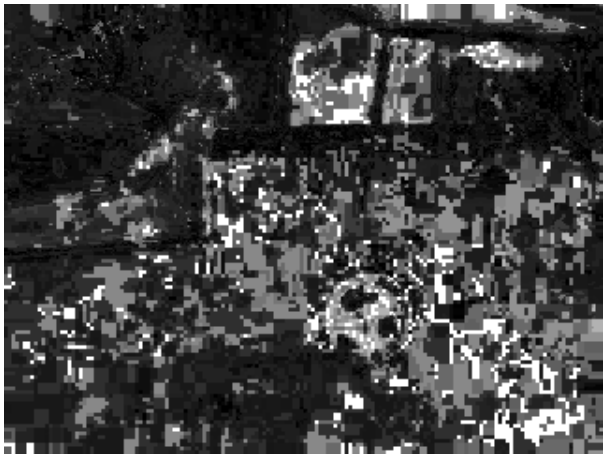


(c) A frame taken halfway through the video



(d) Backpropagation map obtained with max-pooling

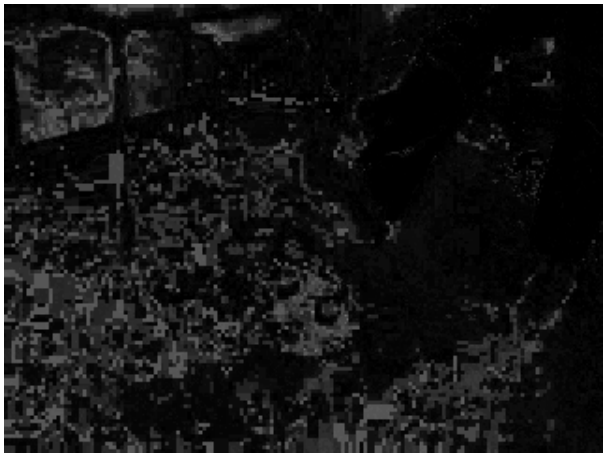
(e) One of the last frames



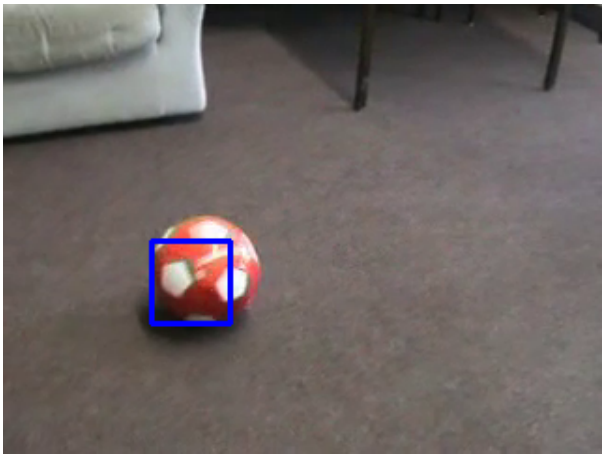
(f) Backpropagation map obtained with max-pooling

Figure (6)

(a) One of the first frames

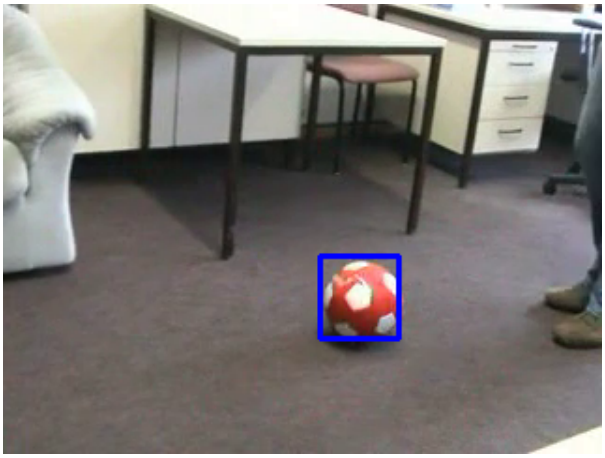


(b) Backpropagation map obtained with mean-pooling



(c) A frame taken halfway through the video

(d) Backpropagation map obtained with mean-pooling



(e) One of the last frames

(f) Backpropagation map obtained with mean-pooling

Figure (7)

2 Hough transform

Q3

Figure (8) Orientations of the gradient where the non-significant ones are shown in red.

Q4

The Hough transform method seems to work pretty well with objects that are not changing too much. Since we initialize the R-table with the first region of interest, if the object we want to track somehow changes from one frame to the other, the method does not manage to isolate the object and other edges take as much importance in the Hough transfrom.

Figure (9) Hough transfrom and good detection on the mug video.

Here, we put the region of interest on the writing. Since it is held upright and it is only being translated, the writings do not change and the method manages to follow the cup pretty well.

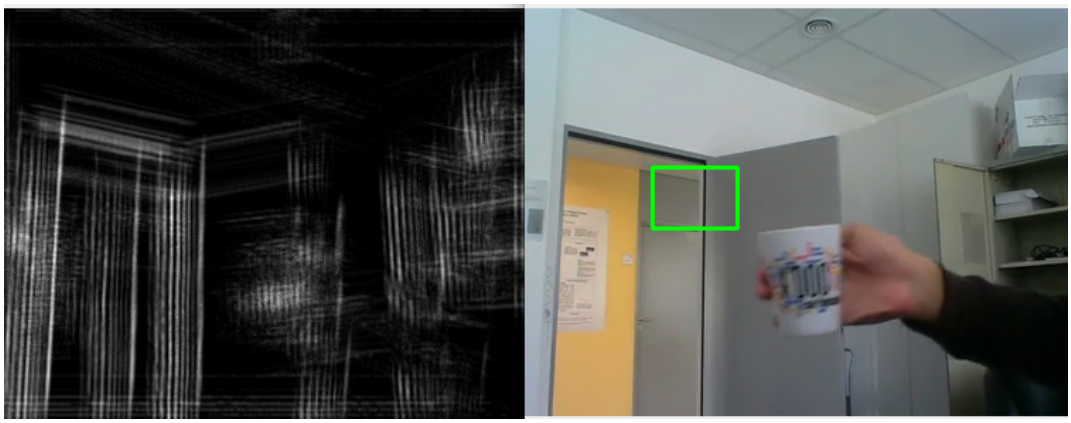


Figure (10) Hough transfrom and bad detection on the mug video.

However, when looking at another frame, the cup is being moved too fast so it is blurry. The effect is that the edges are smoothed and the gradient does not recognize them. The Hough method thus cannot find it while the other edges such as the door frame are still visible. The method does not manage to differentiate the objects and we can see on the Hough transform that the weights are more balanced on the whole frame.

Figure (11) Hough transform and poor detection on the basket video.

An extreme case where the Hough transform is not efficient is the basket-ball video where there are many details, which means many edges. We find that the Hough method does not manage to differentiate the object we are trying to track, we cannot even see it on the map since there are weights everywhere.

Figure (12) Hough transfrom and good detection on the car video.

The Hough method also works pretty well on the car video where the object we are tracking is the car which is pretty big.

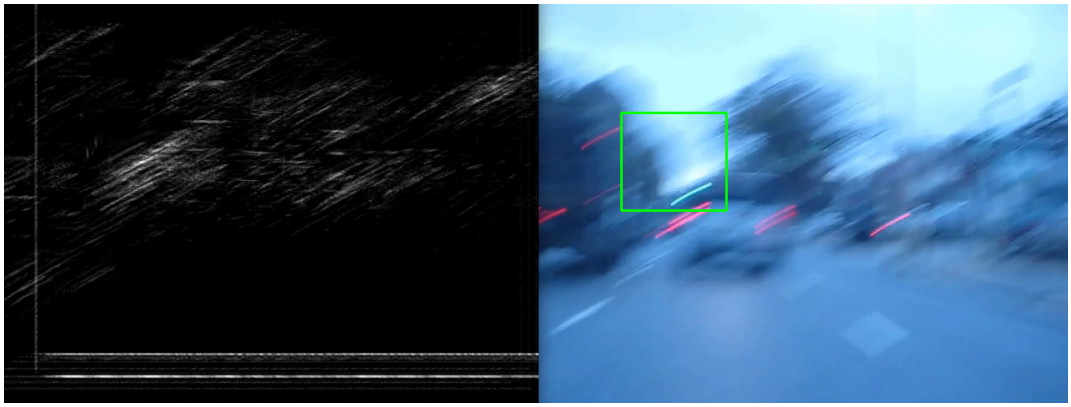


Figure (13) Hough transform and poor detection on the car video.

Of course this video has shaking which means we also find the same phenomenon as the cup where the object cannot be found when it is blurry.

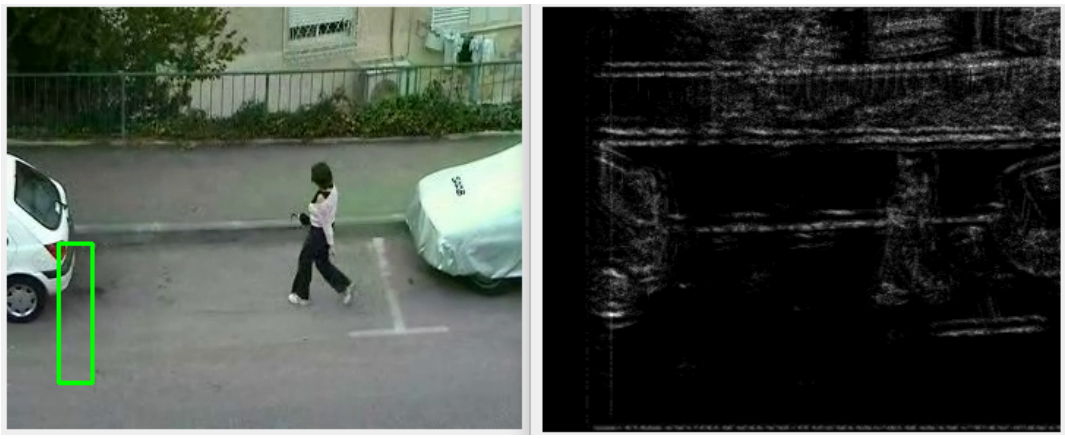


Figure (14) Hough transfrom and poor detection on the woman video.

In the woman video, the problem is that we have a small object that changes shape when moving. Since the region of interest is small, the R-table is smaller and the method obviously does not work as well since there are less orientations for it to relate to. And with the object changing shape, we can see the woman on the Hough map but the weights are close to other weights on the map and the object is not tracked.

3 Synthesis

Q5

Is it possible to combine these two methods ? If yes, how ?
To combine the methods, we can simply use the Hough map as the kernel function for the mean shift method. This uses the number of votes in the Hough transform to evaluate the mean around a pixel and then finding the shift.

This is interesting because it improves the Hough transform method by limiting the displacement of the region of interest from one frame to the next. We previously found that it is not rare that the Hough method loses the object and the region is all of sudden moved to another object somewhere else on the image. Using the mean shift allows to keep the region around its original position, limiting large unrealistic displacements. It also improves the mean shift because it allows to focus more on a model of the object.

However we found some issues when using this technique:

Figure (15) Hough transform and poor detection on the mug video.

In the mug video, we find that when the first fast movement happens, the cup is blurry and the Hough transform does not identify its position well. The problem is that it happens when the cup is in front of the door frame which is one of the part where the Hough method votes the most. This results in the region of interest staying in this area while the cup is being moved somewhere else, hence the region losing the cup. And since we added the mean shift characteristics, the method cannot find the cup again, unless it goes in the same zone again.

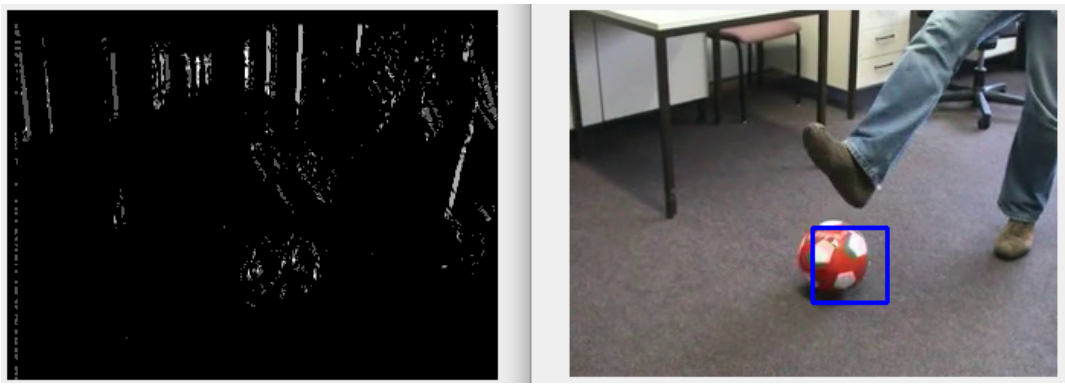


Figure (16) Hough transform and poor detection on the ball video.

The Hough method was not effective with the ball video but it is when combining with the mean shift. This is because with only the Hough method, the region of interest used to get lost in the other objects of the video but mean shift allows to keep the region around the ball which is pretty isolated. It does not help as much on videos where the tracked object is not as isolated such as the basketball one.

Figure (17) Hough transform and poor detection on the basket video.

Combining the methods on the basket video is completely inefficient since the method does not manage to differentiate the objects and all of the edges have weights. This means that the mean shift method will barely move

the region of interest since the weights are almost homogeneous.

How to update the model to make the tracking more robust to appearance changes, or occlusions ?

In order to deal with occlusions we can combine Mean-Shift tracking with Kalman filtering which allows to predict the target's position of the occluded object [1]. We can also use a particle filter.

How to exploit deep features learned by a neural network, in association with (one of) the two previous methods ?

In order to produce discriminative features of a patch we can use a siamese Convolutional Neural Network (CNN) that has to be trained *offline*. This architecture is said to be siamese because we use two CNN with shared weight. The learning is done in a contrastive fashion that consists of training the network to produce embeddings such that the distance between them is maximized if they correspond to different objects, otherwise it is minimized.

We can then perform the tracking via the Mean Shift algorithm in the space of deep features.

References

[1] Oscar Efrain Ramos Ponce, Mohammad Ali Mirzaei, and Frédéric Merienne. "Tracking in Presence of Total Occlusion and Size Variation using Mean Shift and Kalman Filter". In: *2011 IEEE/SICE International Symposium on System Integration*. Kyoto, Japan, Dec. 2011. URL: https://hal.science/hal-00749631.