VISION - TME 7
Object Tracking in Videos

Lucrezia Tosato & Marie Diez

Contents

1 Introduction

In this piratical we will understand the challenges and difficulties of visual tracking, to experiment and develop solutions based on Mean Shift and General Hough Transform algorithms.

2 Mean Shift

2.1 Experiment

This algorithm employs an iterative approach based on the calculation of the mean shift vector. We want to locate the concentrated region in the search space by estimating the mode of a distribution. Given an initial location in this space, we track the mean shit vector v's direction by doing the following at each iteration:

$$v(x_t) = m(x_{t-1}) - x_{t-1} \tag{1}$$

where m(x) is the average around the current point x:

$$m(x) = \frac{1}{n}\sum_{i}^{n} x_i \tag{2}$$

Following the vector v is thus equal to moving towards the place in space where the distribution's mean is located (1). The mean m(x) is frequently computed as a weighted mean, with the weights defined by a kernel function $K : \epsilon \to R^+$ with ϵ the set of points. This function K will determine the importance of each point in the vicinity of x. Then we may write m(x) as:

$$m(x) = \frac{\sum_i^n K(x - x_i)x_i}{\sum_i^n K(x - x_i)} \tag{3}$$

The Mean Shift is thus an algorithm with a straightforward implementation, which is one of its key merits. After working with the films provided to us, we discovered the following drawbacks: the approach is overly sensitive to initialization, if the ROI is not exact enough, the algorithm might easily lose tracking. Another downside is that it is impossible to resume tracking if the object is lost. In Figure 1, we can see how the algorithm works in iterations:

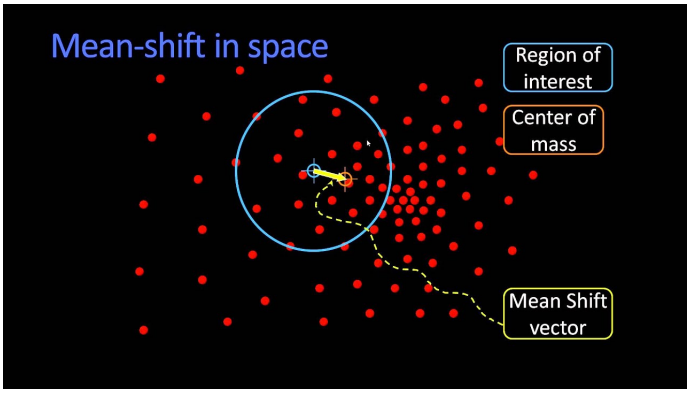


Figure 1: Mean Shift

Once a region corresponding to the object is selected, the associated histogram is computed and normalized to obtain a kind of probability density. The backprojection consists in estimating the probability of a pixel to belong to the object, for that we look at the value corresponding to the intensity of this pixel on the normalized histogram, which will give us the probability of this pixel to belong to the object.

Here are some results :

Figure 2: H

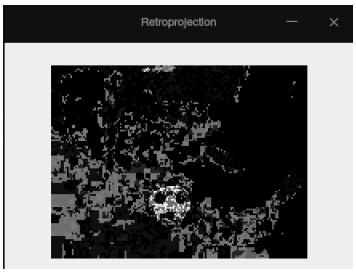
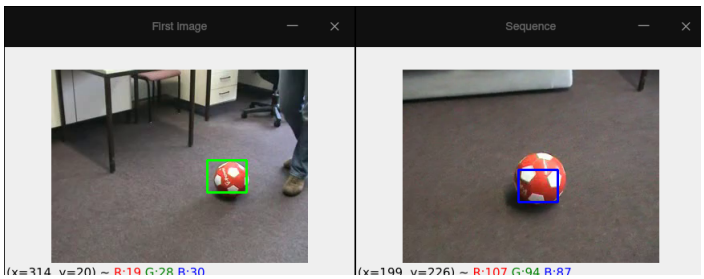
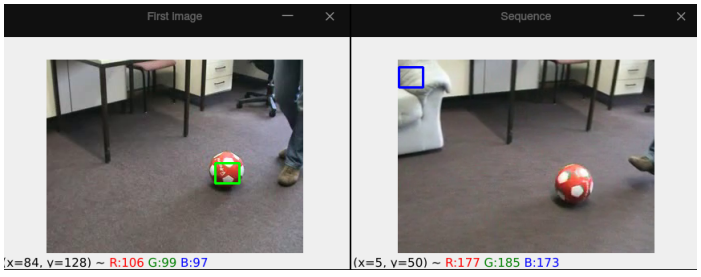


Figure 3: Backprojection



(a) frame 1



(b) frame 2

Figure 4: Tracking

We can see that the results of the tracking are not very powerful, indeed we look at the following display (we look at the hue H by saturating S and V) we observe that the balloon has the same colour as the background :

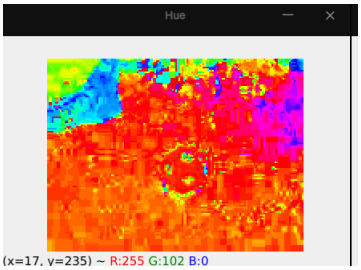


Figure 5: Hue image

We can experiment to see how the algorithm behaves, for example change the attribute on which the histogram is calculated, rather than using the hue H we can use the saturation S:

Figure 6: S

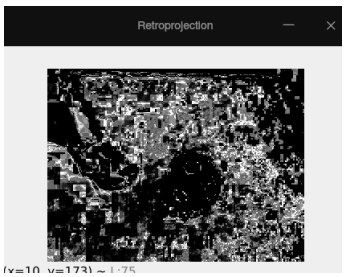


Figure 7: Backprojection

Figure 8: Tracking

We can also change the attribute on which the histogram is calculated, rather than using the hue H we can use the value V :

Figure 9: V

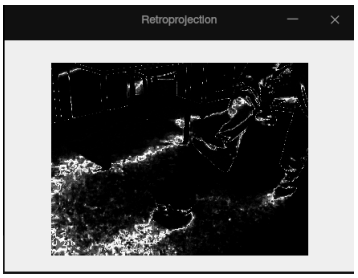


Figure 10: Backprojection

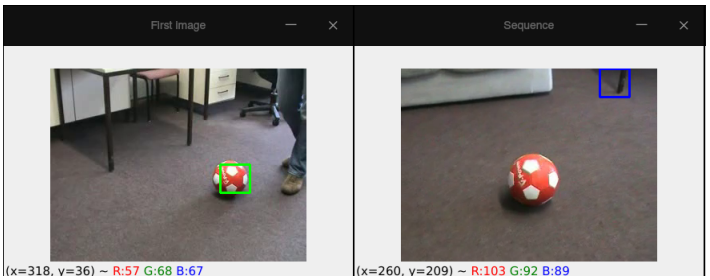
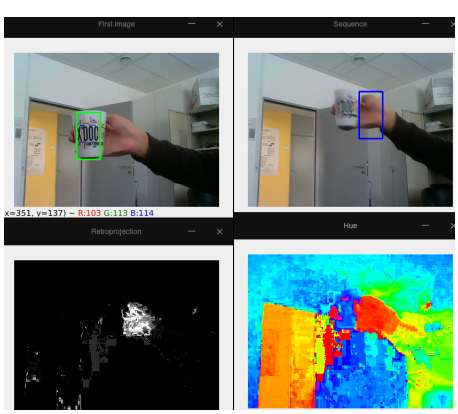


Figure 11: Tracking

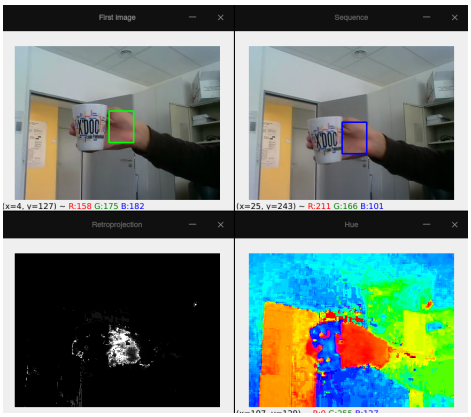
We can also see that the tracking does not perform well for the saturation or the value, the object is very poorly recognised.

2.2 Analyse

As we seen before for the Hue plot, we saw that if the object have the same color as the background the detection can't work well. When evaluating areas with the same colors, such as the mug in Figure 13, the system performs badly. It is more efficient for more contrasting pictures,such as the hand in the video like we can see below :



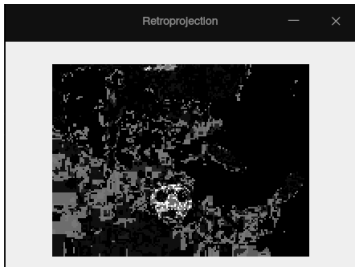
(a) Goblet selection



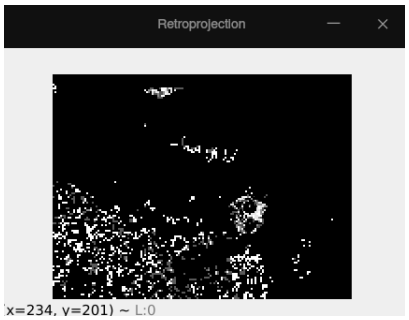
(b) Hand selection

Figure 12: Tracking

For the previous exemple we can threshold the backprojection like : $dst[dst[:,:] < 80] = 0$



(a) Backprojection without thresholding



(b) Backprojection with thresholding

Figure 13: Tracking

It's relatibly difficult to get the exact same frame, but globaly we can see that the thresholding seems to improve the detection of the object. It reduce a little to have to much false positive pixel in the backprojection map.

However, thresholding will deplete the information in the backprojection even more than it already is. The problem can be seen as limiting the high probability on the pixels that do not belong to the object (the noise), for this we can think of smoothing the original images frames with a gaussian. We can't expect this method to improve of the results, indeed the smoothing is not enough to solve the problem of colours between the ball and the background. It can be a solution in case where the noise or too much texture are present in the image, but here it's not the main problem so the gaussian blurring can't improve the result.

Let's look at the results :

Figure 14: Results with gaussian blurring $\sigma = 1$

Figure 15: Results with gaussian blurring $\sigma = 5$

Indeed we can't see an improvement of our tracking.

Another idea would be to change the criterium on what the histogram is computing, as we did in the previous section with S and V, but we couldn't succed to have good result, we would have to chose another criterium, for exemple the hough accumulator as we will see later.

Mean shift robustness to change of illumination
We try to track the pull of the guy in the video *Sunshade*, the mean shift algorithm don't work well in this exemple, after that the gui goes into the shadow the tracking is lost. It can be due to the lake of presision on change illumation scene, and/or also we have the same problem as for the ball in the shadow side, the object as the same hue color as the background.

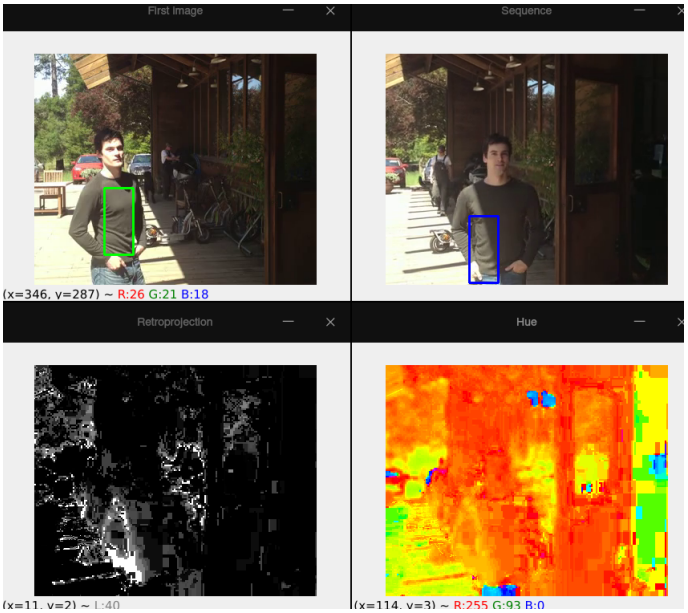


Figure 16: Mean shift H saturation criterium to compute the histogram

3 Hough Transform

3.1 Calculate

The local orientation is determined for each frame using OpenCV's Sobel function. Using a pre-defined kernel, this method returns the image's derivative in each direction (x and y). The gradient vector can then be constructed by taking the two derivatives, and its norm can be calculated by calculating $||G|| = \sqrt{g_x^2 + g_y^2}$. The norm will act as a mask, with a threshold determining which pixels are relevant. Figure 18 shows non-significant pixels in red. The following steps were taken: The norm is first adjusted to be between 0 and 255, and then only pixels with a gradient and a norm greater than 25 are taken. This threshold value was discovered through testing and analyzing the findings.

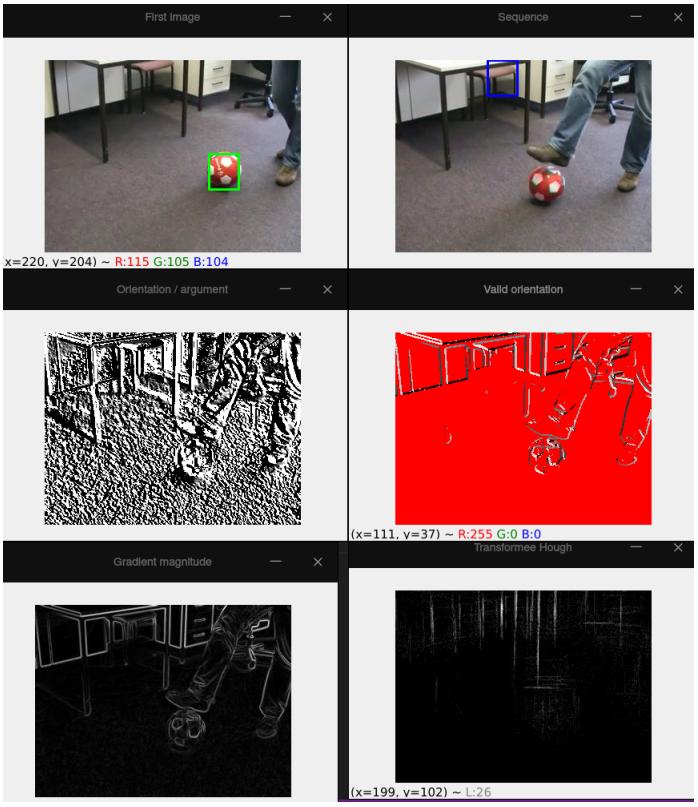


Figure 17: Implementation of the pre-techniques for the Hough Transform

3.2 Build

A Generalized Hough Transform with an R-table is used to construct an object model for this section. The R-table is built from a prototype, with each point indexed with regard to a geometric property. In this case, we use the orientation with regard to the prototype's center. The generalized transform concept is to have a list of "votes" that indicate the hypothesis of object center localization (a more voted point is considered as being more probable to correspond to the real centre of the object). Experimenting reveals that the tracking of the object is sometimes lost. Here the object is lost earlier than mean shift method.

Indeed, in circumstances where there are multiple edges competing with the chosen ROI, this approach, which takes into account the picture contour (as realized by the norm) and uses the most significant contours to be based on, can be difficult to stabilize. However, depending on the implementation, this technique performs similarly to the mean shift algorithm.

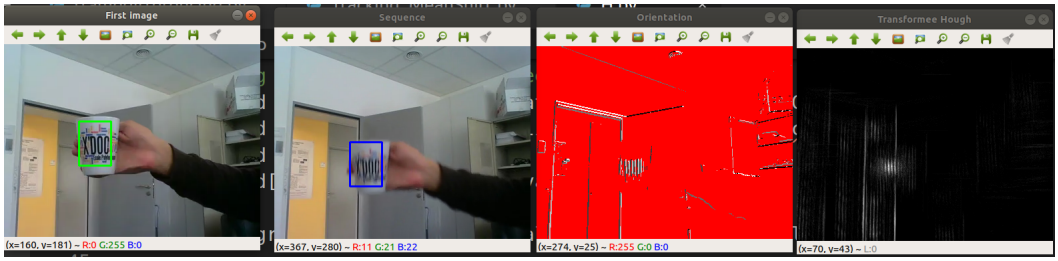


Figure 18: Hough transform

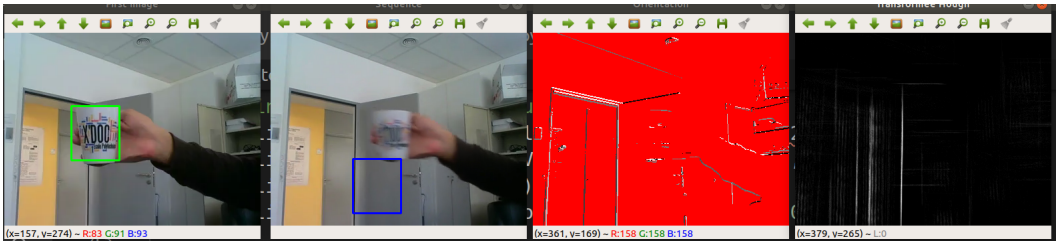


Figure 19: Hough transform

3.3 Replace

As mentioned in the previous question, using the Hough transform with Mean Shift requires a significant amount of processing. However, this change improves the method's stability. When the highest value represents the location of the object, a tracking window that is too short may not be able to recover a lost tracking.
In this scenario, we provide to the Mean Shift a more complicated reaction that seeks the appropriate return on investment needed to follow the object in question by employing the two linked techniques, where we insert the most voted values chosen by the Hough algorithm. As a consequence, the strong components of the two methods have been integrated.

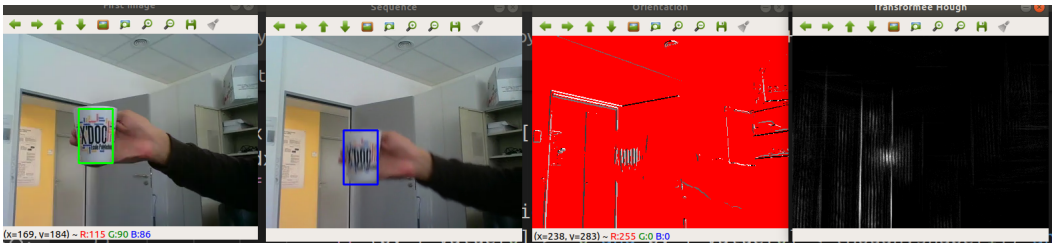


Figure 20: Hough transform plus Mean Shift

In the case of the Antoine mug the lines are not as much a problem as before, because the center of the object will no longer be the argmax of the accumulator. The shift in the direction of the more voted pixels enable to be a little less subject to this problem. But it's not sufficient to track perfectly the glass.

For the example of the ball we have better result with this algorithm that only each separately. Indeed we seen the problem of colors, and for the Hough transform the center of object diverge immediately :

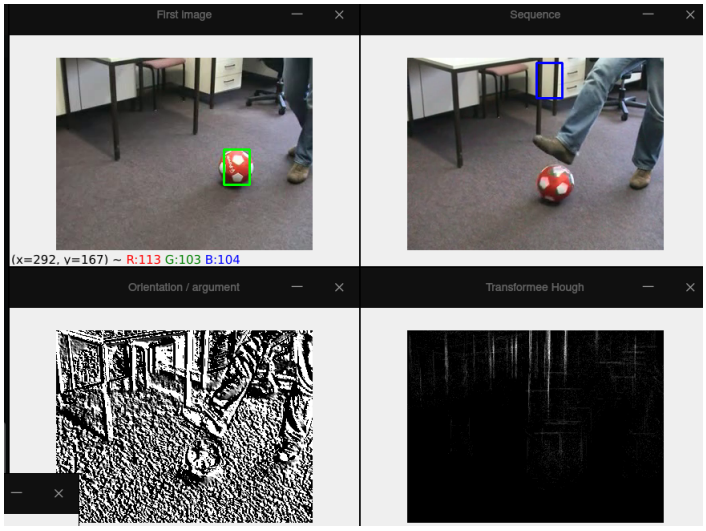


Figure 21: Hough transform

With the Hough transform with Mean Shift, the ball is lost with the line of the couch, but not at the start as in the simple Hough transform:

Figure 22: Hough transform plus Mean Shift

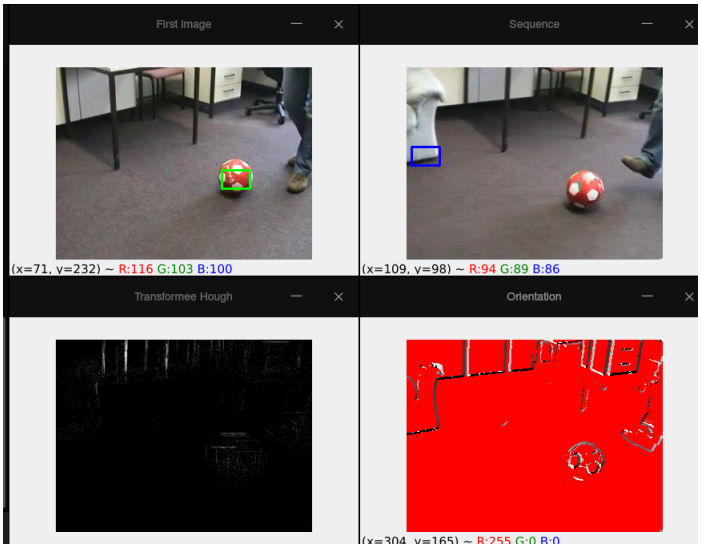


Figure 23: Hough transform plus Mean Shift

4 Bibliography

[1] Georgia Tech, “Introduction to Computer Vision: Mean Shift in Space”
Udacity. Available: ://www.youtube.com/watch?v=TMPEujQrY70.

ML | Mean-Shift Clustering available at : https://www.geeksforgeeks.org/ml-mean-shift-clustering/