1 Implementation of the functions

1.1 Color histogram

I create a simple function that compute the histogram using np.histogram.

1.2 Derive matrix A

The derivation was almost trivial. Just create an array with numpy and the correct functions.

1.3 Propagation

The function generates the random part and the deterministic part. Then I combine both parts and I propagate the particles. I make sure the particles stay inside of the image.

1.4 Observe

For each particles I compute the bounding box, compute the histogram with the bounding box and compute the distance between the two histograms. The weights of the particles follow a gaussian distribution. Sometimes particles maybe get Nan as value. Hence I give them a low value. At the end I return the normalized values.

1.4 Mean state

Compute the mean. Nothing hard.

1.5 Resample

I resample the particles using the function np.random.choice and I make sure the samples are normalized at the end.

2 Experiments

2.1 Video 1

The tracking is working without problems. See screenshots below

2.2 Video 2

Using a constant velocity model, the a priori is less smooth than with the full noisy model, which makes sense.

Reducing the system noise make the a priori smoother.

Increasing the measurement noise makes the tracking less agressive.

2.3 Video 3

The model with only noise is working very fine. However we need to have sigma position bigger than ten because the ball move faster than the arm. See the screenshot below.

2.4 Conclusions

The number of particles is a trade off between performance precision. If we have more particles we have more precision but it takes longer during each epochs.

The number of bins is also a trade off between precision and performance. It takes less time to compute and compare histogram with less bins.

Using adaptive histogram could be a good solution to deal with some scene variation like light change. However we could after some step have a completely noisy histogram.

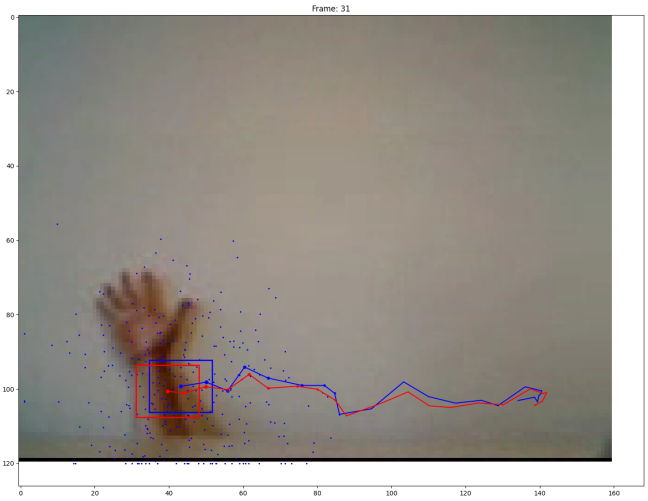

Figure 1: Figure 1

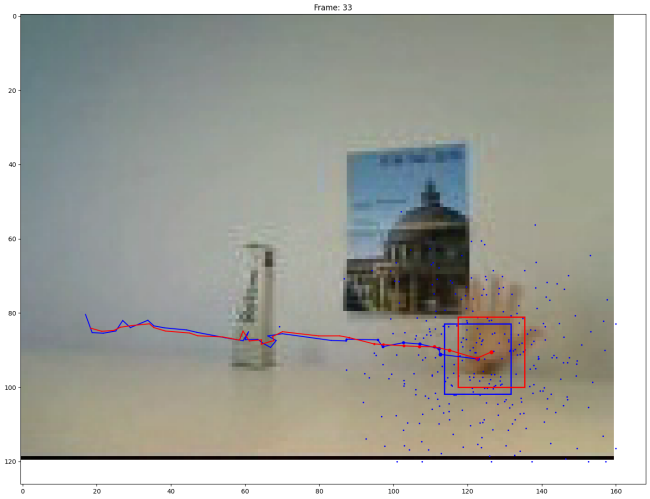

Figure 2: Figure 2_1

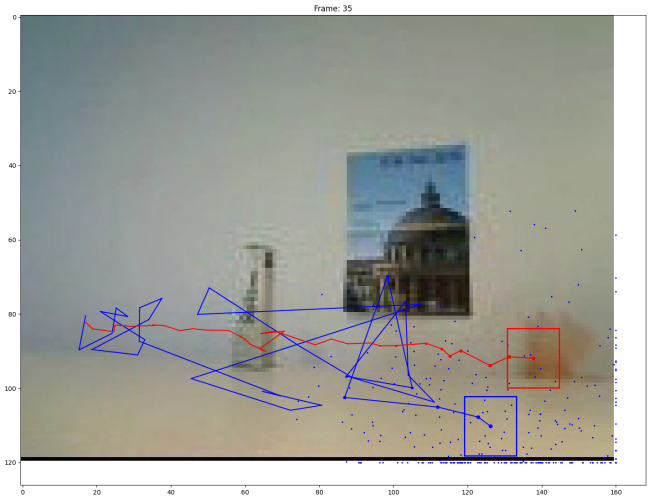

Figure 3: Figure 2_2

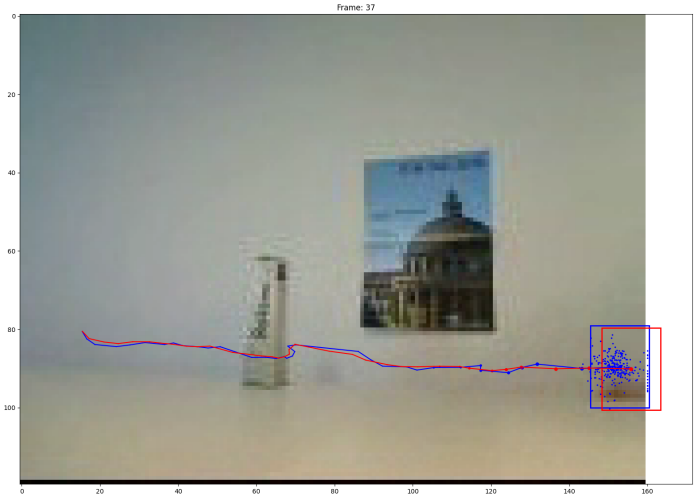

Figure 4: Figure 2_3

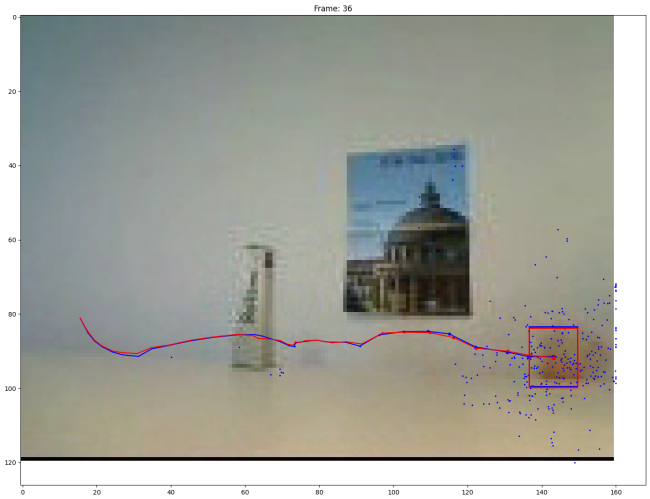

Figure 5: Figure 2_4

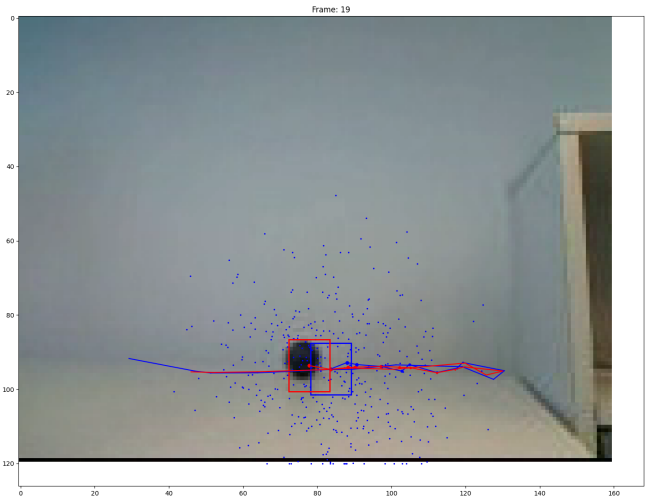

Figure 6: Figure 3