Condensation Tracking

Chao Ni

December 9, 2020

In this exercise, we are going to implement conditional density propagation over time algorithm. Essentially this is a sampled-based solution of the recursive Bayesian, pretty much similar to the assignment three, where particle filtering is implemented. We are going to split the exercise into four parts: implementation, results, discussion and a try with my own video.

1 Implementation

1.1 Color histogram

We will use a bounding box as a descriptor for the feature. For all three channels, we will compute a color histogram within a given number of histogram bins. All histograms will be normalized. We assume it would be normalized based on the sum of all three channels.

1.2 Derive Matrix

The processing model depends on the model we will be using. It we just consider pure noise, the we have two states, namely the position of the object, and this object will center around its last step position after receiving some noise. $x_{t+1} = x_t + \epsilon$, therefore the matrix would be $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. If the object has a constant velocity, then $x_{t+1} = x_t + \dot{x}_t dt$. Assuming a unit time step and augmenting the state results in the processing model $A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

1.3 Propagation

The propagation of the particles follows the processing model. We need to generate random noise for each states based on the standard deviation given by the exercise.

1.4 Observation

We will need to update the wights for each particle based on the observation. The observation is represented by the position of the particle and a bounding box, by comparing this box with the target object box, we can compute the weight for this particle. All weights will also be normalized afterwards.

1.5 Estimation

Based on the particles and its weights, we compute its mean for all states.

1.6 Resampling

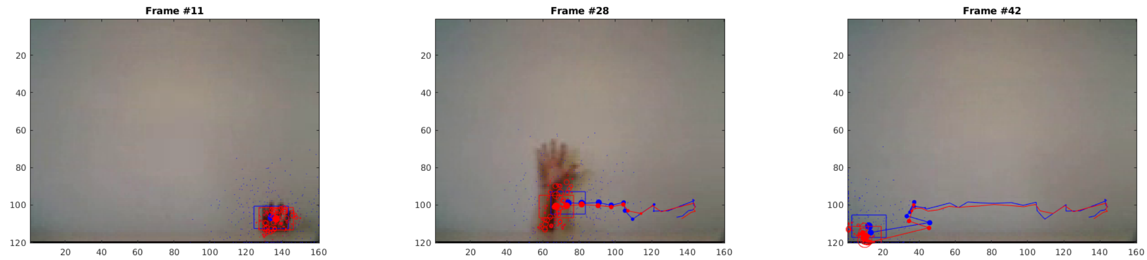
Like we did in particle filtering, we resample the particles based on their weight. There are multiple methods to do that, we simply call matlab nested function *datasample* to realize that and compute the corresponding weights for each resampled particle.

2 Experiments

In this section, we will first show some plotting results for all three videos with carefully selected parameters.

2.1 Video1

Video1 is relatively easy to track, with default setting and no motion model, we plot the trajectory of mean state (prior and posterior) as shown in Fig. 1. Blue bounding box represents a priori mean state and red a posteriori mean state.



(a) frame 11 (b) frame 28 (c) frame 42 (last frame)

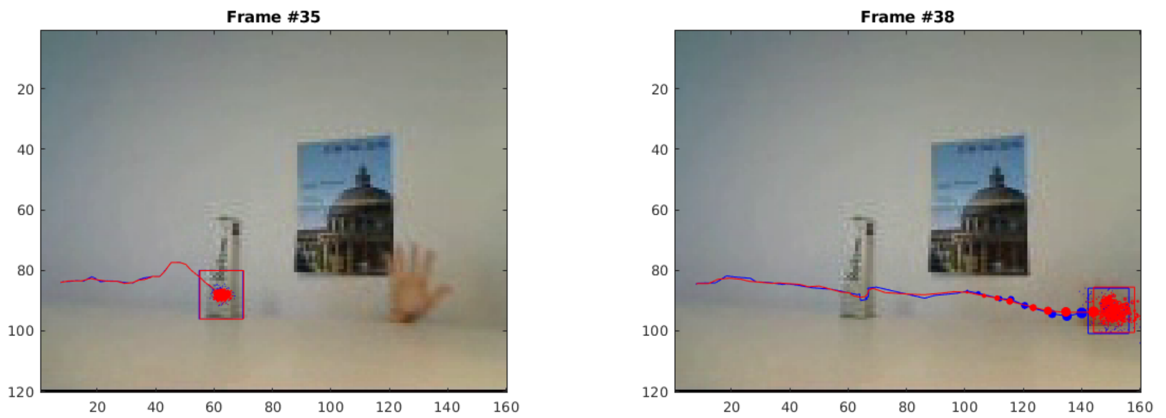
Figure 1: No motion model, default parameter setting, it is able to track the object correctly.

2.2 Video2

In this subsection, we are going to explore the effects of parameters on the tracking. We will consider different parameter settings and reason the phenomenon.

2.2.1 Motion model

If we used the default setting, we will find without motion, the algorithm is also able to track the object. This is due to the relatively large position variance, therefore there always some particles that are able to track the object very well. In order to gain more insights of the effect of motion model, we set the position standard deviation to unit. We then have the results shown in Fig. 2.

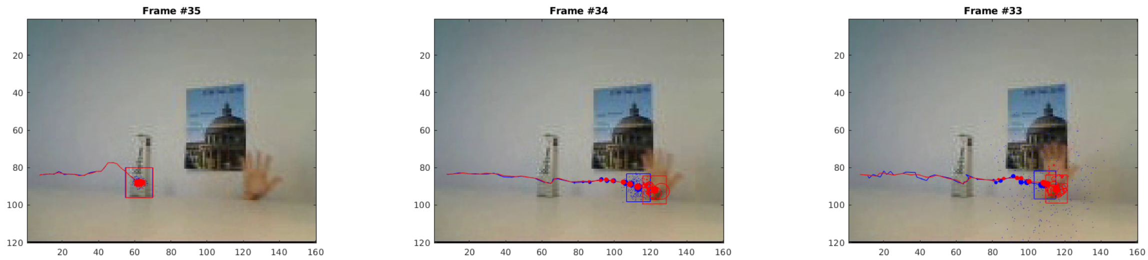


(a) no motion model: the object is not tracked due to occlusion (b) constant velocity model: zero initial velocity

Figure 2: Motion model helps with occlusion case, it is able to predict the object even if the object is not visible. We use unit standard deviation for positions.

2.2.2 System noise

Some insights are already given in the previous subsection, we will explore in detail how system noise effect the tracking performance. We take the no motion model for example, and vary the standard deviation for position, the results are shown in Fig. 3. We can see when increasing the variance, the sampled particles are much scattered apart, and the chances of tracking the object increases. With first image, the tracking failed.

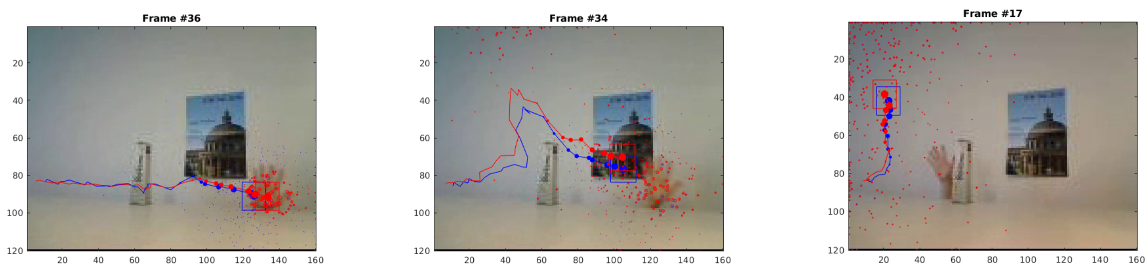


(a) $\sigma_{position}$: 1 (b) $\sigma_{position}$: 3 (c) $\sigma_{position}$: 15

Figure 3: When increasing the system noise, the sampled particles are much scattered apart, and the chances of tracking the object increases. With first image, the tracking failed.

2.2.3 Measurement noise

The measurement noise matters to the weight computation for each particles. If the measurement noise is too big, i.e. the variance is too large, then the distribution will be flattened, which means some particles that are far away to the target would also be given relatively large weights. This would lead to failure of successful tracking. We will verify this with some results. We set $\sigma_{position}$ as default value 15 and choose no motion model for this experiment.

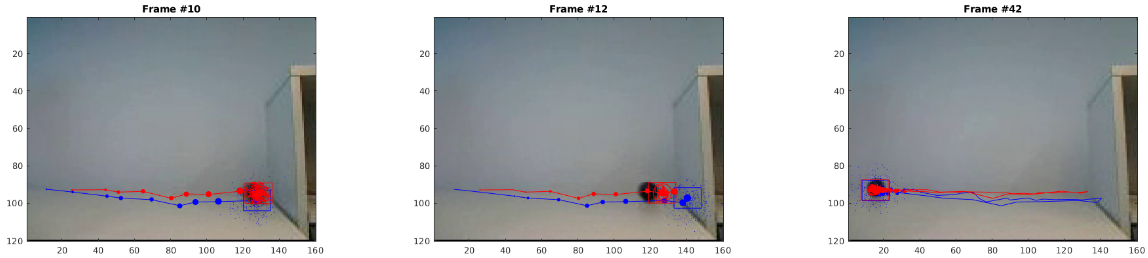


(a) $\sigma_{observe}$: 0.3 (b) $\sigma_{observe}$: 0.65 (c) $\sigma_{observe}$: 1.0

Figure 4: When increasing the measurement noise, the sampled particles are much scattered apart because their weights are more evenly distributed. This, however, would potentially decrease the chances of giving some closer particles priority and therefore leads to failure.

2.3 Video3

With careful selection, we choose motion model and set $\sigma_{position} = 5$, with zero initial velocity. With this setting, we are able to track the ball. The results are shown in Fig. 5. The tracking performance is not good as in the second video, partly because the ball is not strictly moving constantly. One observation can be made from the second image, where the priori and posteri differs much. This is due to the fact that the ball bounces and changes its moving direction. At that moment, the priori doesn't anticipate it and it corrects itself afterwards. Generally, all effects applying to the second video also apply for the third video.



(a) frame: 10 (b) frame: 12 (c) frame: 42

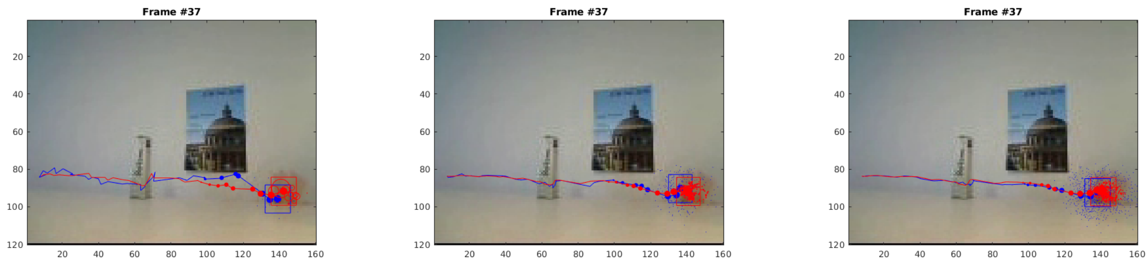
Figure 5: The tracking performance is not good as in the second video, partly because the ball is not strictly moving constantly. One observation can be made from the second image, where the priori and posteri differs much. This is due to the fact that the ball bounces and changes its moving direction. At that moment, the priori doesn't anticipate it and it corrects itself afterwards.

3 Discussion

In this section we will research on the effects of number of particles, number of bins and analyze the pro and con of allowing appearance model updating.

3.1 Number of particles

We can see from Fig. 6 that with more particles, the tracking is more smooth as the noise is eliminated.



(a) Number of particles: 30 (b) Number of particles: 300 (c) Number of particles: 1000

Figure 6: With more particles, the tracking is more smooth, potentially more accurate and unbiased.

3.2 Number of bins

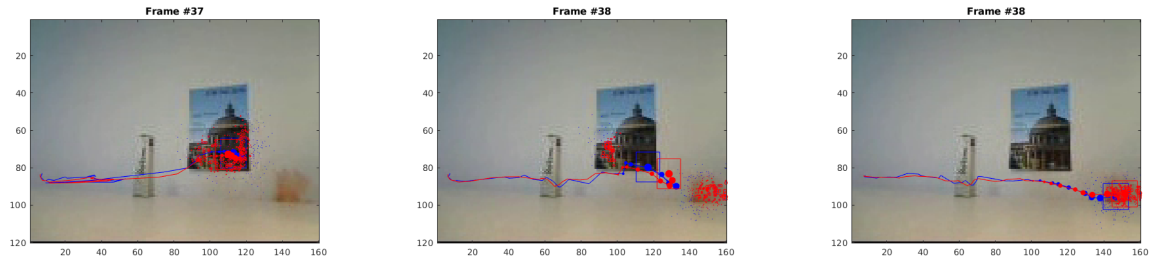
If we decrease the number of bins, there is a chance the algorithm cannot compare different patches correctly, as shown in Fig. 7. The picture on the wall will distract the algorithm and leads it to a failure of tracking.

3.3 Appearance model

To update the model means a compromise between the initial target and the current status. We illustrate the effect by showing a comparison as shown in Fig. 8. AS to speak of pros and cons, we can list several based on intuitions:

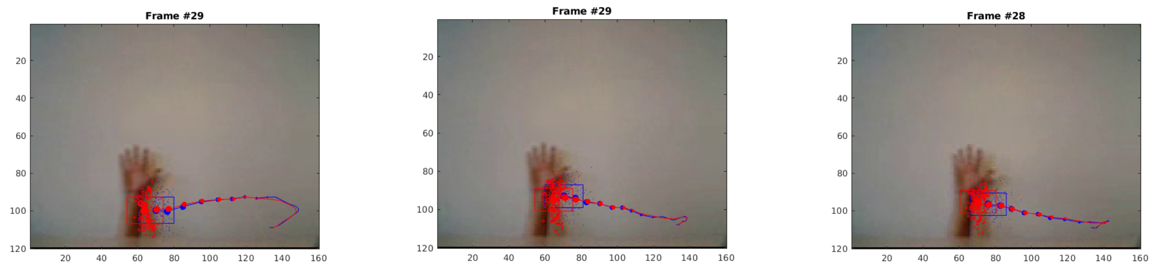
3.3.1 Pros

- Adjusting to the actual scene makes the tracking smooth;
- If the initial bounding box is not good, sticking to this target might lead to failure.



(a) Number of bins: 2 (b) Number of bins: 4 (c) Number of bins: 8

Figure 7: Decreasing number of bins would potentially lead to failure of tracking because the picture will distract the algorithm from making the correct comparison.



(a) No update (b) Update: 0.5 (c) Update: 1

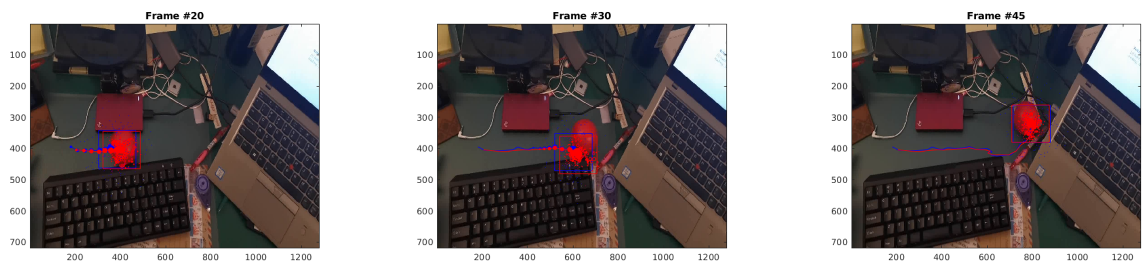
Figure 8: The initial bounding box is mainly on the fingers, and the color histogram is basically half dark and half light, which looks much like the wrist. If we don't include update, then it will end up tracking the wrist as (a) shows, if we introduces update, then it will also consider the actual scene we have seen so far and make a compromise.

3.3.2 Cons

- Adjusting to the current status might also amplify the mistake and lead to failure tracking.

4 My own video

I happen to have the ETH pressure ball on my desk, so I decide to record a short video and track the ball. I use no motion model and have to increase the position variance to $\sigma = 50$ such that the algorithm works. The result is pretty impressive and it shows the algorithm is robust as the scene I used has a complex background.



(a) frame: 20 (b) frame: 30 (c) frame: 45

Figure 9: Because this image is pretty large compared to other images seen before, we need to increase the variance as well, otherwise the bounding box just stay at the same place.