Condensation Tracker

Qi Ma, 20-960-225, qimaqi@student.ethz.ch

December 24, 2021

Abstract

In this task we use the CONDENSATION algorithm - CONDitional DENSity propagaTION over time, which is a sampled-based solution of the recursive Bayesian filter. There are several steps including "Prediction, Update". The underlying idea is to represent the probability distribution by a weighted sample set $S = (s(n), \pi(n)|n = 1, ..., N$. Consequently, the distribution of the samples constitutes a discrete approximation of the full probability distribution.

1 CONDENSATION Tracker Based On Color Histograms

Our first task is to implement a CONDENSATION tracker based on color histograms using python. The objects to be tracked are represented by bounding-boxes whose state is given as

$$s = \{x, y, \dot{x}, \dot{y}\} \tag{1}$$

where x,y represent the location of center of the bounding-box, and $\dot{x}, \dot{y}$ the velocities in the x and y direction. The x direction goes along with width of frame and y goes along with the height of frame.

1.1 Input

Sample set S_{t-1} and the target model color histogram CH_{target}

1.1.1 Color histrogram

The color histrogram calculate the normalized histogram of RGB colors occurring within the bounding box. This kind of descriptor can express the local feature in a compressed form. Instead of looking into 256^3 possible colors we only look $hist - bin^3$ possible distinct colors instead. We use χ^2 distance to calculate the distance between different color histogram.

1.2 Step 1: Sampling

Draw N samples from the set S_{t-1} with probabilities $\pi(n)$. The new set of samples forms the S'_{t-1}

1.3 Step 2: Prediction

Propagate each sample s'_{t-1} from the set S'_{t-1} by a linear stochastic differential forms the set S'_{t-1} equation:

$$s(n) = As'^{(n)}_{t-1} + w^{(n)}_{t-1} \tag{2}$$

where w(n) is the stochastic component. Thus we obtain $S = s(n)|n = 1...N$.

1.3.1 Derive matrix A

In this exercise we consider two prediction models: (i) no motion at all i.e. just noise; and (ii) constant velocity motion model. The A matrix of no motion at all is:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

Since

$$\begin{aligned} x_t &= x_{t-1} + w_{t-1}^1 \\ y_t &= y_{t-1} + w_{t-1}^2 \end{aligned} \tag{4}$$

Where for constant velocity motion model:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Since

$$\begin{aligned} x_t &= x_{t-1} + t * \dot{x}_{t-1} + w_{t-1}^1 \\ y_t &= y_{t-1} + t * \dot{y}_{t-1} + w_{t-1}^2 \\ \dot{x}_t &= \dot{x}_{t-1} \dot{x}_t = \dot{x}_{t-1} \end{aligned} \tag{6}$$

As above shows we can set time interval as 1 because we sample the velocity anyway so interval information can be included in the sampling process.

1.4 Step 3: Update

Observe the color distributions:

- For each sample s(n) compute the color histogram $CH_{(n)}$
- For each sample s(n) compute the χ^2 distance between its color histogram and the target color histogram
- Reweight each sample s(n)

1.5 Step 4: Estimate

We can get the estimation by weight different particles together with:

$$E[s_t] = \sum_{n=1}^{N} \pi_t^{(n)} s_t^{(n)} \tag{7}$$

After each iteration, the color histogram of the target (CH_{target}) is updated as a convex combination between the old color histogram of the target and the color histogram of the mean state:

$$CH_{target} = (1 - \alpha) * CH_{target} + \alpha * CH_{E[s_t]} \tag{8}$$

2 Experiment

The experiment include 3 parts of video1.wmv, video2.wmv and video3.wmv.

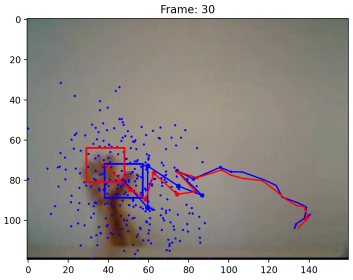
2.1 Video 1

The first video is relatively easy to track because no hiddence and no background interferce. I conduct four experiment to get good understanding of how different parameters influence the tracking performance

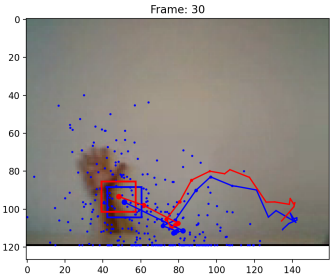
Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0	300	15	-	0.1	-
b	1	0	300	15	1	0.1	(1, 10)
c	0	0	600	15	-	0.1	-
d	1	0	600	15	1	0.1	(1, 10)
e	0	0.2	600	15	-	0.1	-
f	1	0.2	600	15	1	0.1	(1, 10)
g	0	0.5	600	15	-	0.1	-
h	1	0.5	600	15	1	0.1	(1, 10)
g	0	0.8	600	15	-	0.1	-
h	1	0.8	600	15	1	0.1	(1, 10)
I	0	1.0	600	15	-	0.1	-
J	1	1.0	600	15	1	0.1	(1, 10)

From the experiment in video 1 we can learn some ideas in parameter tuning.

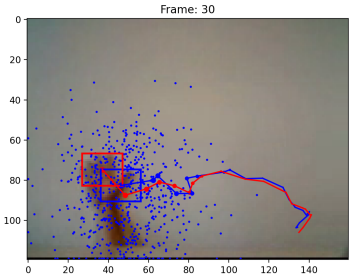
- Adding particle number will help tracking to some extend. This is easier to understood since more points will help the tracker to cover unknown movement so tracking will be easier. This also can be explained by more particles will better approximating the distribution of model movement.
- The constant velocity motion model do not always perform better than no motion prediction model. Especially in the case with small particle numbers. It can be explained by bad initial speed and also more sample space since more states means more combination.
- In the case of no occlusion and no complex background, allowing appearance model updating will help tracking a lot since the first color histogram can not well represent the object in later movement. So updating the color histogram will help tracking a lot.



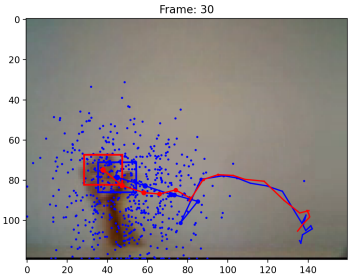
(a) Exp a.



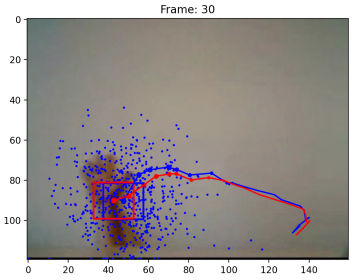
(b) Exp b.



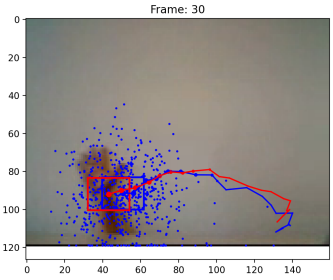
(c) Exp c.



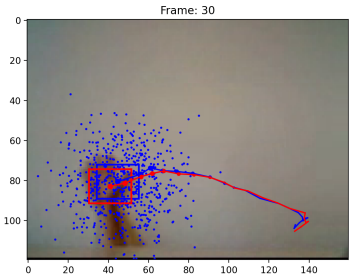
(d) Exp d.



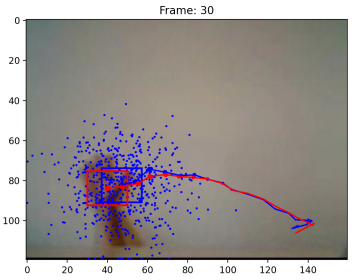
(e) Exp e.



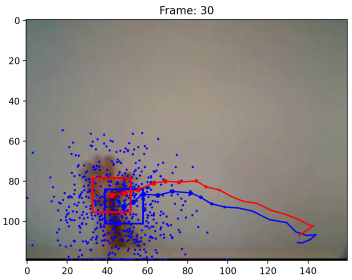
(f) Exp f.



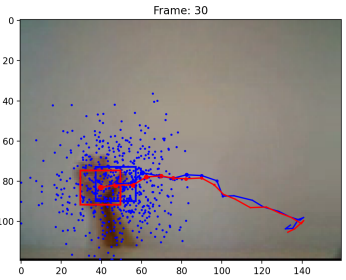
(g) Exp g.



(h) Exp h.



(i) Exp i.



(j) Exp j.

Figure 1: Experiment video 1: Test with parameters

2.2 Video 2

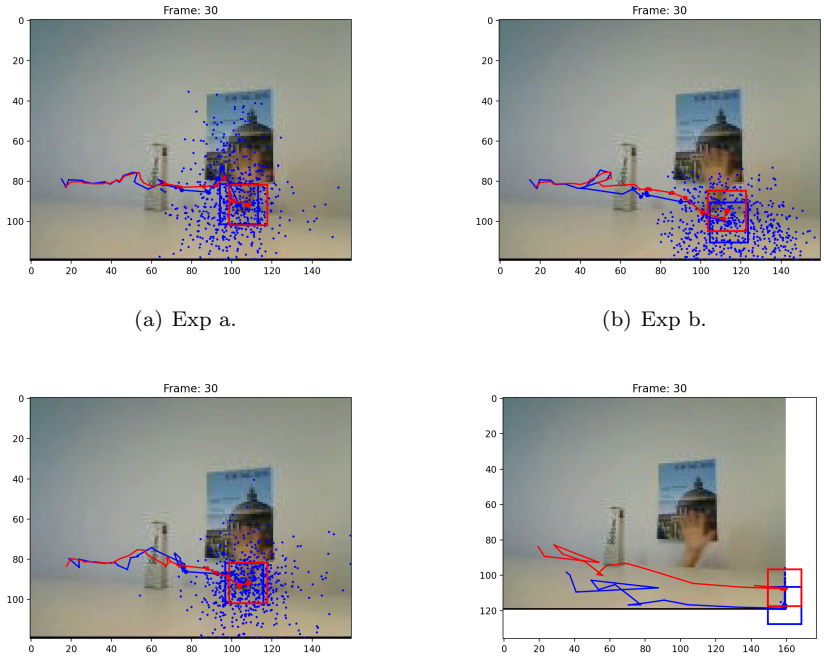
In this video we have occlusion and also background interference. The occlusion will inevitably lead to tracking loss. But the key is whether the tracker work after occlusion. Moreover the background will influence the color histogram greatly.

2.2.1 Motion model effect

At first we will analyze the effect of using a constant velocity motion model. We compare 4 different result with 1) no motion model, 2) with motion model but zero initial, 3) with motion model and good initial value. 4) With motion model but bad initial value.

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0.1	600	15	-	0.1	-
b	1	0.1	600	15	1	0.1	(0, 0)
c	1	0.1	600	15	1	0.1	(5,1)
d	1	0.1	600	15	1	0.1	(20, 20)

From the qualitative result I learned that if we use a motion model then compared to non-motion model, the trajectory is smoother than non-motion model. Most trajectory are parallel to each other and no abrupt jittering. This can be explained by the fact that movement by non-motion model depends most on position variance so all direction are possible. In contrast the constant velocity model assumed the model move in one direction in constant speed. And some turning is caused by velocity variance. I also use zero initial velocity, good initial velocity and bad initial velocity. Even initialize with 0 velocity, the trajectory is not so bad since the velocity variance will help to shift the zero initalize to better value. However if we initilize with a too bad value, the tracker will lose tracking.



(a) Exp a.

(b) Exp b.

(c) Exp c.

(d) Exp d.

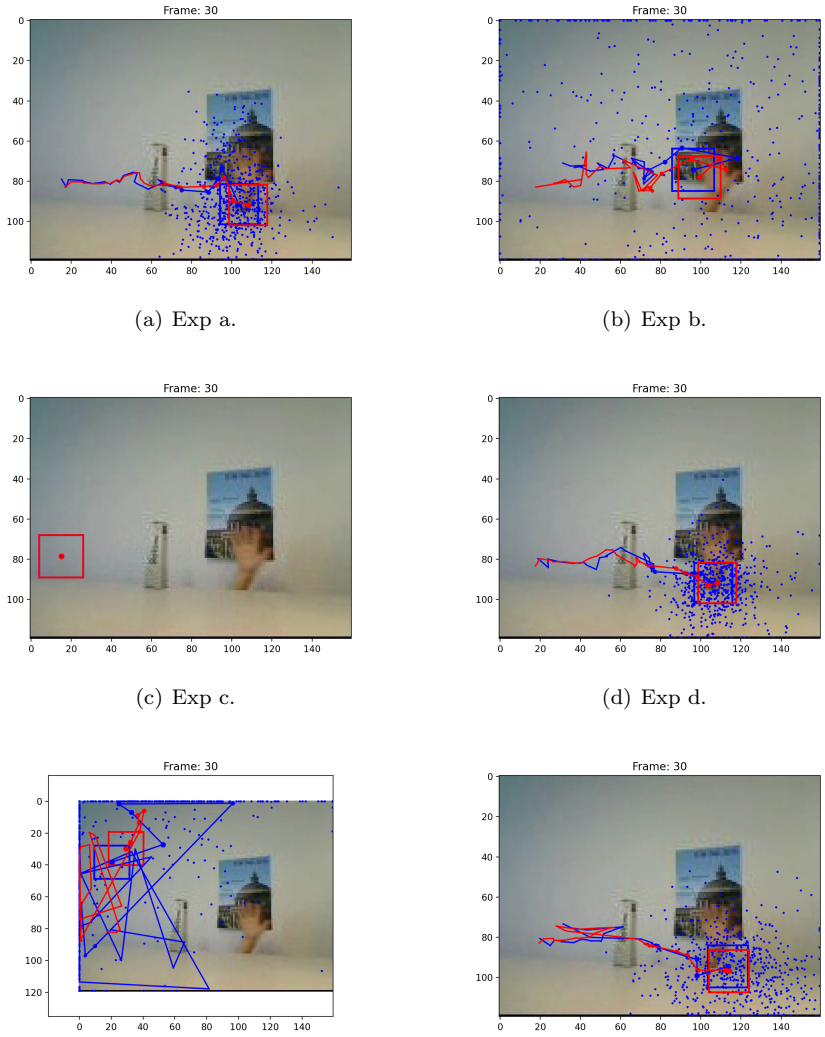
Figure 2: Experiment video 2: Model effect

2.2.2 System noise effect

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0.1	600	15	-	0.1	-
b	0	0.1	600	60	-	0.1	-
c	0	0.1	600	0	-	0.1	-
d	1	0.1	600	15	1	0.1	(5, 1)
e	1	0.1	600	15	50	0.1	(5, 1)
f	1	0.1	600	15	0	0.1	(5, 1)

The influence of position sigma variance is huge. If we use a small sigma, the tracker will stop at start point with non-motion model as we expected. If we use a large variance than the particle will go around whole image and most of particle are useless which result in bad trajectory.

For speed variance the result is similar. If we use a large variance the trajectory will be bad since the constant velocity is corrupted by large noise and the trajectory will look never like a line anymore like pic e shows. If we initialize with a 0 velocity noise then the trajectory looks like a line everywhere but due to position noise the whole trajectory will not be straight.



(a) Exp a.

(b) Exp b.

(c) Exp c.

(d) Exp d.

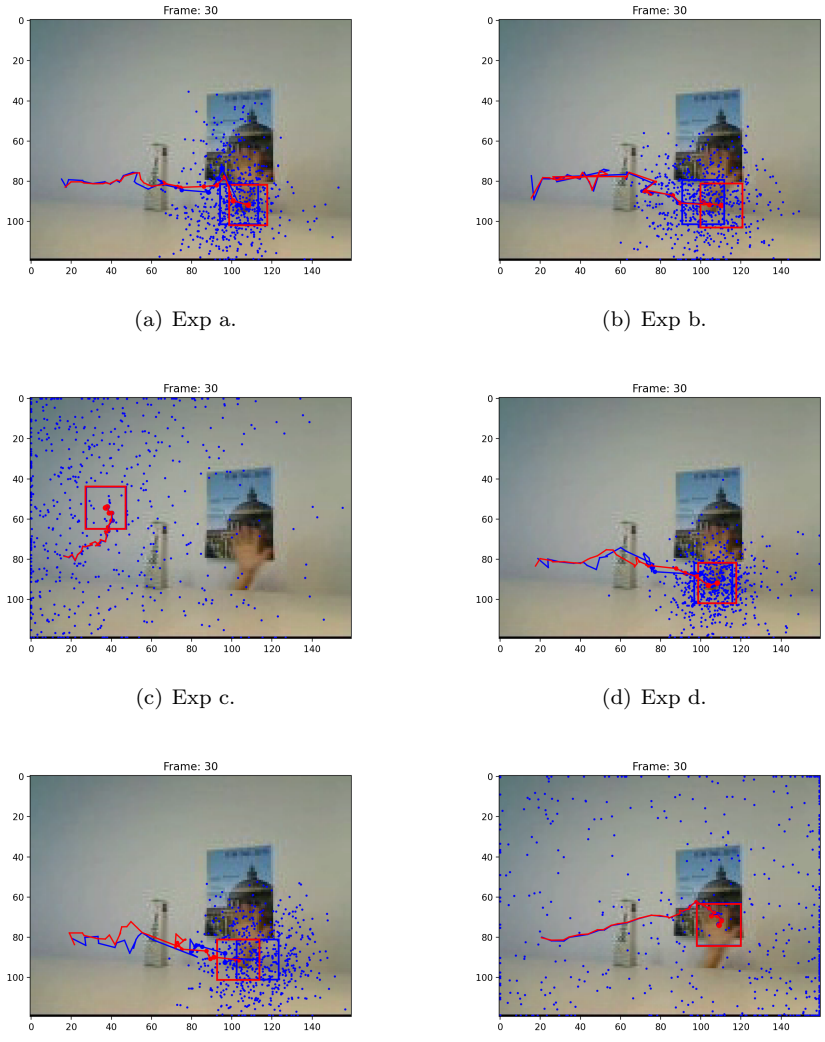
(e) Exp e.

(f) Exp f.

Figure 3: Experiment video 2: System noise effect

2.2.3 Measurement noise effect

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0.1	600	15	-	0.1	-
b	0	0.1	600	15	-	0.01	-
c	0	0.1	600	15	-	1	-
d	1	0.1	600	15	1	0.1	(5, 1)
e	1	0.1	600	15	1	0.01	(5, 1)
f	1	0.1	600	15	1	1	(5, 1)



(a) Exp a.

(b) Exp b.

(c) Exp c.

(d) Exp d.

(e) Exp e.

(f) Exp f.

Figure 4: Experiment video 2: Measurement noise effect

In this experiment we test the effect of different measurement noise. If the measurement noise is close 0, then according to the equation below:

$$\pi^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp - \frac{-\chi^2((CH_s), CH_{target})^2}{2\sigma^2} \tag{9}$$

Even if the color histogram differ with the target histogram a bit, the distance will be large so I can understand the tracker is very sensitive. In contrast when the σ is large then the tracker is less sensitive and the re-weighting will be useless because all distance are close.

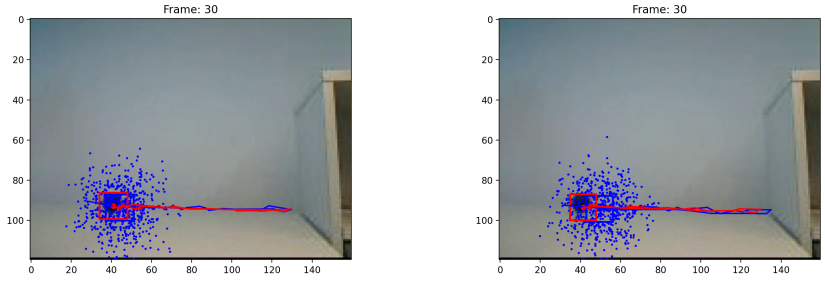
Therefore when σ set to 0.01 the tracker are more turbulent and when set to 1 the tracker lose the tracking. In the motion model case the large σ works better than non-motion case, but from the distribution of particiles we can still see the bad effect of large σ, the patch overall image have similar weight because large measurement noise

2.3 Video 3

In this experiment I will face a bouncing ball. This movement is complex and luckily we do not have occlusion and bad background.

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0.1	600	15	-	0.1	-
b	1	0.1	600	15	2	0.1	(5, 0)

It turns out the tracker works great in both motion model and non-motion model. As explained in experiment 2, even thought the direction of movement change totally but since we have σ_{pos} and σ_{vel} so we always have particle which will lead to right movement.



(a) Exp a.

(b) Exp b.

Figure 5: Experiment video 3

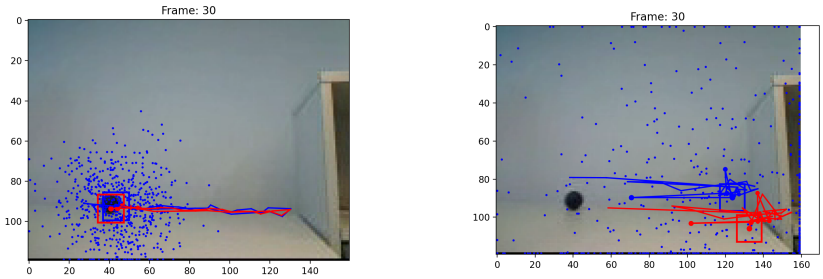
Similar to experiment 2 we do similar with video 3

2.3.1 System noise effect

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0.1	600	15	-	0.1	-
b	0	0.1	600	60	-	0.1	-
c	0	0.1	600	0	-	0.1	-
d	1	0.1	600	15	1	0.1	(5, 1)
e	1	0.1	600	15	50	0.1	(5, 1)
f	1	0.1	600	15	0	0.1	(5, 1)

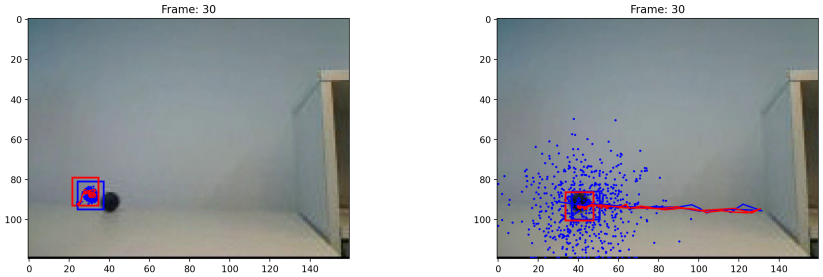
There are some similarity in the experiment 3, as large position variance like figure 6.b shows, the particles will go around all over the image and lead to bad tracking. small position variance will keep the tracker stuck at start point.

But in contrast the large variance in velocity will still bring good tracking in this case. So is the case with small variance. One possible explanation is that the motion-model fit the video so well and even with small or large variance, there are always particles successfully track the ball so the tracker is robust to noise.



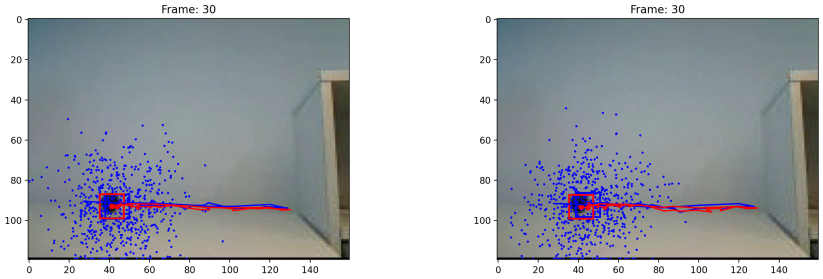
(a) Exp a.

(b) Exp b.



(c) Exp c.

(d) Exp d.



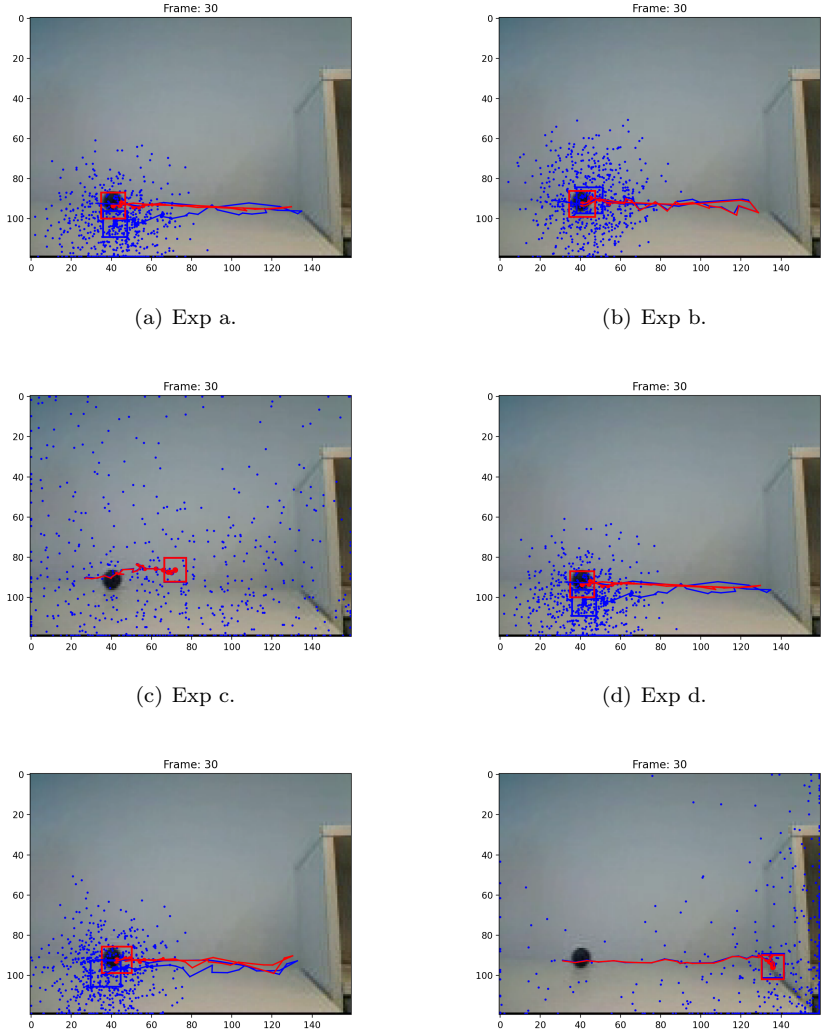
(e) Exp e.

(f) Exp f.

Figure 6: Experiment video 3, system noise

2.3.2 Measurement noise effect

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel
a	0	0.1	600	15	-	0.1	-
b	0	0.1	600	15	-	0.01	-
c	0	0.1	600	15	-	1	-
d	1	0.1	600	15	1	0.1	(5, 1)
e	1	0.1	600	15	1	0.01	(5, 1)
f	1	0.1	600	15	1	1	(5, 1)



(a) Exp a.

(b) Exp b.

(c) Exp c.

(d) Exp d.

(e) Exp e.

(f) Exp f.

Figure 7: Experiment video 3: Measurement noise effect

This experiment also support the result we got from experiment2. With small observation noise the tracker will be sensitive and with large measurment noise the tracking fails because tracker assume equal weight over all particles in whole image.

3 Discussion

In this section we will discussion and summerize what I learned from this task.

3.1 What is the effect of using more or fewer particles?

Particles number is the most important parameter I will say in this task. Since we use sample method to approximate the real distribution of random variable. So more particles we had, the better approximation we have. But also the computation will be heavy. I also learned that the higher dimension state will always need more particles since the state combination complex and harder to approximate.

3.2 What is the effect of using more or fewer bins in the histogram color model?

The bins in color histogram means the expression power of descriptor. We can conduct a easy example with video 3 by using different color bins.

Experiment	Model	alpha	particle-number	σ_{pos}	σ_{vel}	σ_{mea}	initial-vel	color bins
a video3	1	0.1	600	15	1	0.1	(5,1)	4
b video3	1	0.1	600	15	1	0.1	(5,1)	8
c video3	1	0.1	600	15	1	0.1	(5,1)	16
d video3	1	0.1	600	15	1	0.1	(5,1)	32
e video2	1	0.1	600	15	1	0.1	(5,1)	4
f video2	1	0.1	600	15	1	0.1	(5,1)	32

From the qualitive result we can learn how color bins number influence the tracking. If we use a simple model (color bins = 4) Then the simple color histogram can not distinguish the difference between table shadow and the ball (Like in figure 8.a and 8.e shows). If we use a color histogram with large color bins can express complex object as shown in figure 8.f, the tracker successfully tracking the hand in the poster background.

3.3 What is the advantage/disadvantage of allowing appearance model updating?

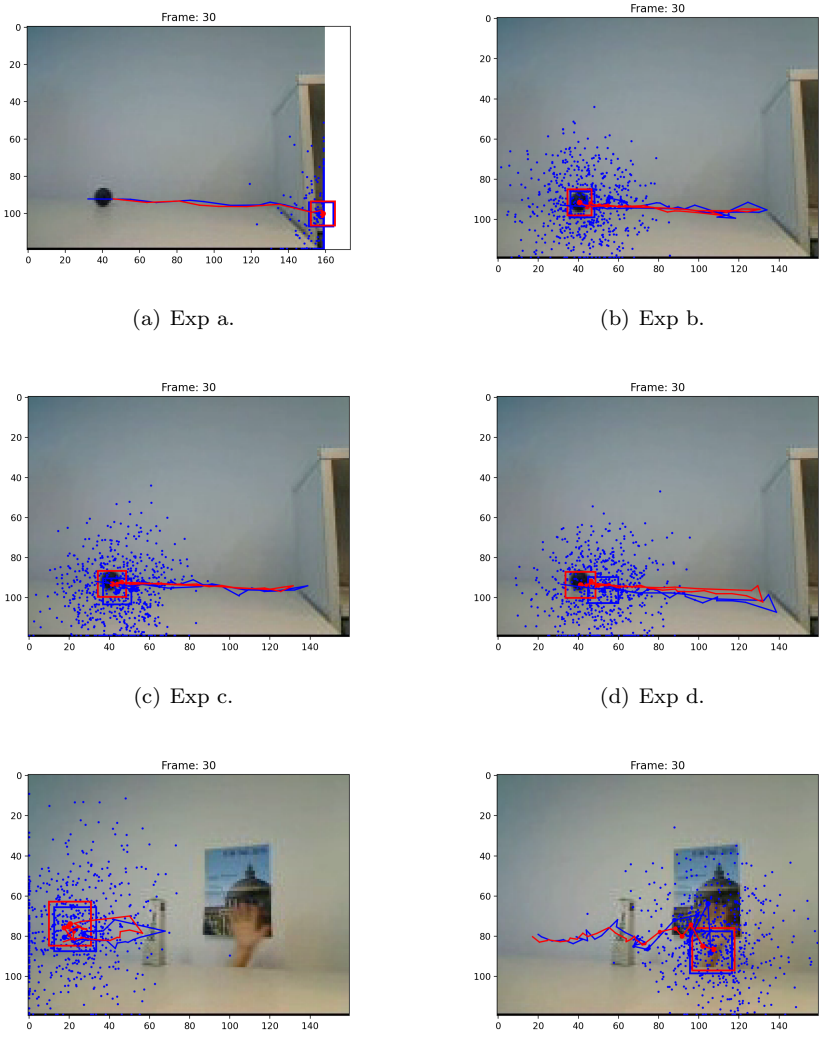
The model updating is good when the object may have some deformation or illumination change and then the original histogram can not describe the current object well, this advantages also work when the background of object changes like the poster in video 2. The disadvantage is that when we meet a occlusion, the update of model will consider the wrong object and hard for later tracking. So how to set the value is a trick. I will set small value of α if the occlusion happened and I will set large α for updating.

3.4 Prospecive future

I really learned a lot from this task. But due to time limit I only do some easy experiment and do not make own contribution, in the future I would like to add some feature in this tasks.

- Adapting alpha based on the occlusion and changing background
- Detecting whether the object leave the frame by rejecting large histogram difference

Thanks for all works from TA and Happy Chirismas!



(a) Exp a.

(b) Exp b.

(c) Exp c.

(d) Exp d.

(e) Exp e.

(f) Exp f.

Figure 8: Color bins effect