[CV21] Assignment 6 - Di Zhuang

Implementation

Color histogram

First, we select all pixels within the bounding box. Then we apply the function numpy.histogramdd() to the pixels to obtain a histogram of the colors of all pixels with d = 3. Finally, we normalize and return the histogram.

Derive matrix A

(i) no motion at all

$$s = \begin{pmatrix} x \\ y \end{pmatrix} \qquad As = s \Rightarrow A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

(ii) constant velocity.

$$s = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \qquad As = \begin{pmatrix} x+\dot{x} \\ y+\dot{y} \\ \dot{x} \\ \dot{y} \end{pmatrix} \Rightarrow A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Propagation

First, we check the model of the algorithm. If model = 0, there is no velocity. So for each particle, we add a 2d noise drawn from a normal distribution with (0, sigma_position) to the center of the particle. If model = 1, there is a constant velocity. So for each article, we add the velocity (x’, y’) and a 2d noise drawn from a normal distribution with (0, sigma_position) to the center (x, y). In addition, we add a 2d noise drawn from a normal distribution with (0, sigma_velocity) to the velocity (x’, y’).

Finally, for those particles with a new center out of the frame, we clip it to the boundary of the frame.

Observation

Given the width and height of the bounding box, for each particle, compute the color histogram of the bounding box that is centered at the particle's center. Compute the distance between the particle's histogram and the target histogram, based on which the particle's weight can be computed. Compute and save the weights for all particles.

Estimation

Given the states and weights of the particles, compute the normalized weighted states of the particles.

Resample

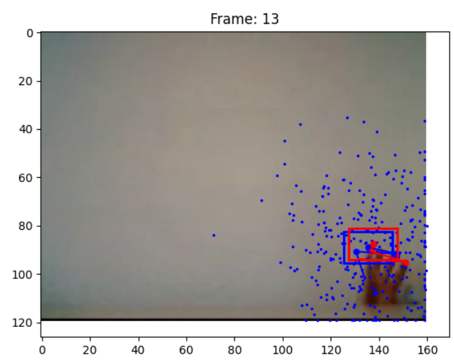
Use numpy.random.choice() to to resample the particles and the corresponding weights.

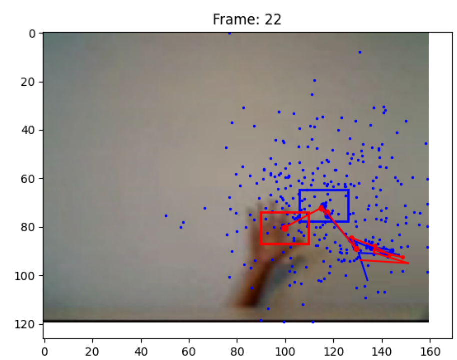
Experiments

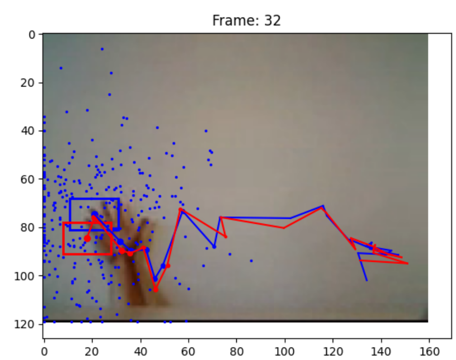
Video 1

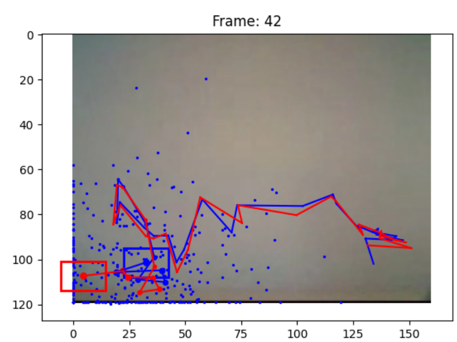
Successful tracking

```
params = {
        "draw_plots": 1,
        "hist_bin": 64,
        "alpha": 0,
        "sigma_observe": 0.1,
        "model": 0,
        "num_particles": 300,
        "sigma_position": 20,
        "sigma_velocity": 1,
        "initial_velocity": (1, 10)
    }
```



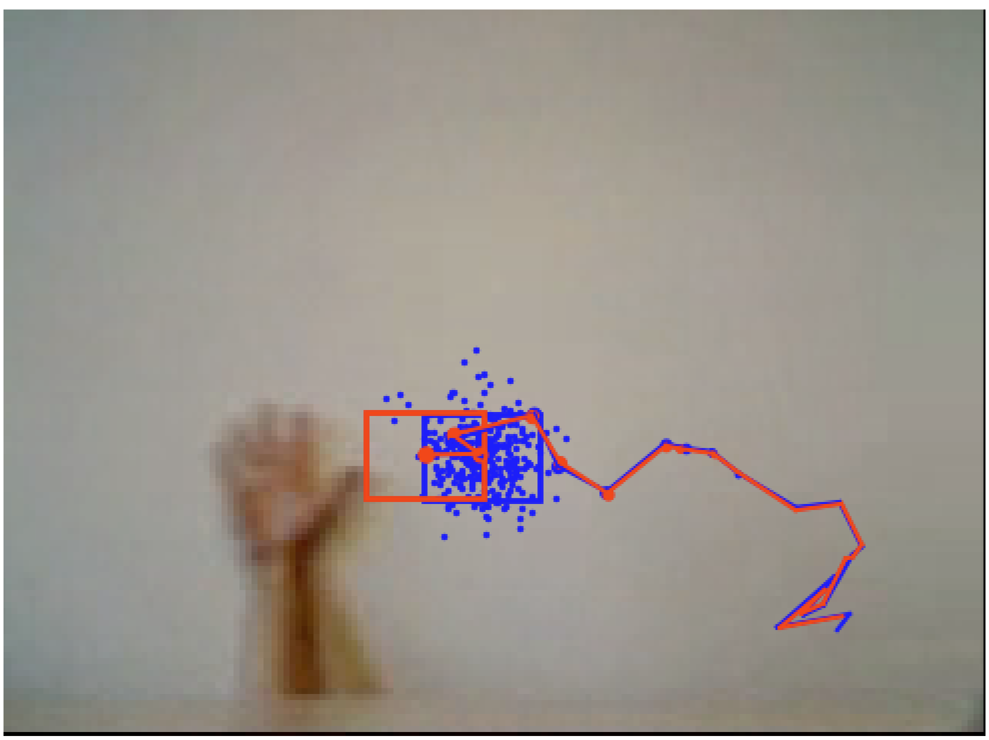






Unsuccessful tracking

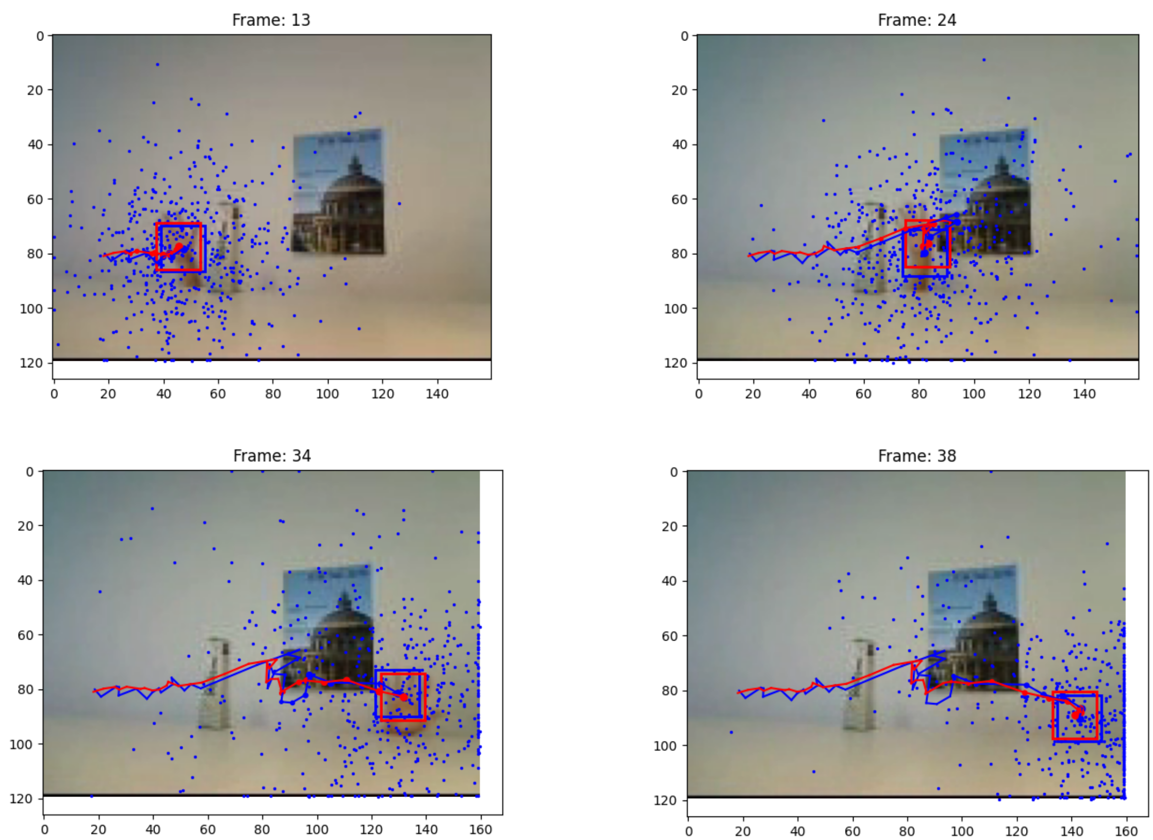
```
params = {
    "draw_plots": 1,
    "hist_bin": 64,
    "alpha": 0,
    "sigma_observe": 0.1,
    "model": 0,
    "num_particles": 300,
    "sigma_position": 5,
    "sigma_velocity": 1,
    "initial_velocity": (1, 10)
}
```



Video 2

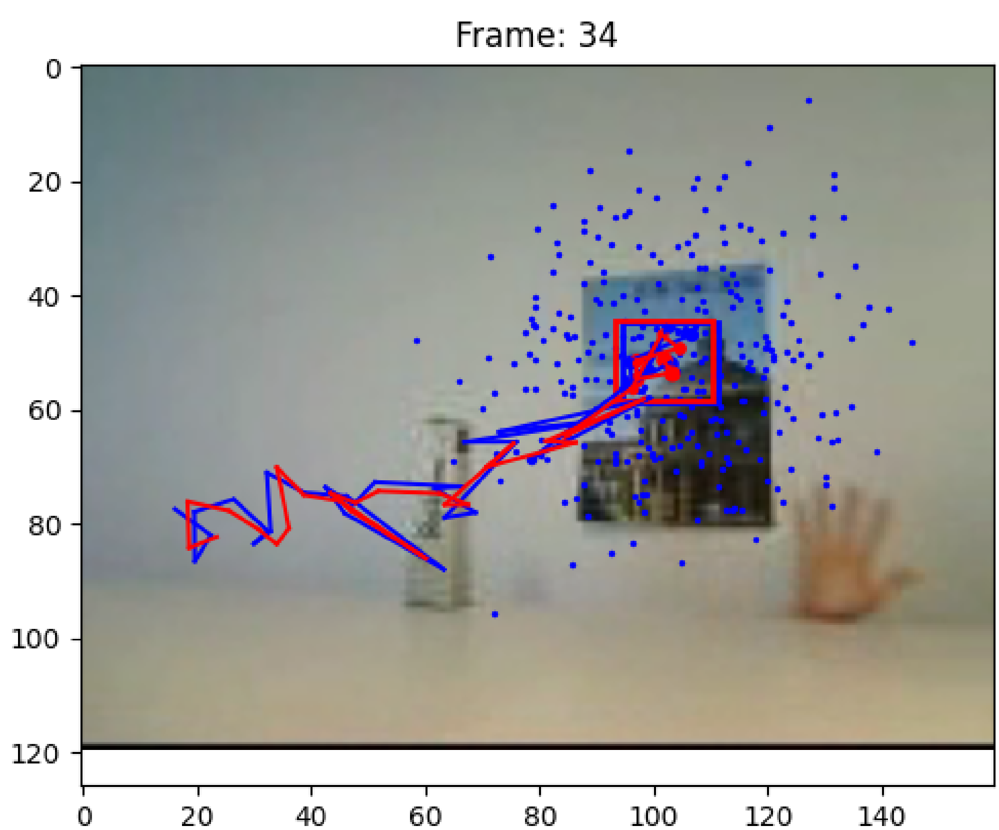
Successful tracking

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.5,
    "model": 1,
    "num_particles": 500,
    "sigma_position": 15,
    "sigma_velocity": 3,
    "initial_velocity": (5, 1)
}
```



Unsuccessful tracking

```
params = {
    "draw_plots": 1,
    "hist_bin": 64,
    "alpha": 0,
    "sigma_observe": 0.1,
    "model": 0,
    "num_particles": 300,
    "sigma_position": 15,
    "sigma_velocity": 1,
    "initial_velocity": (1, 10)
}
```

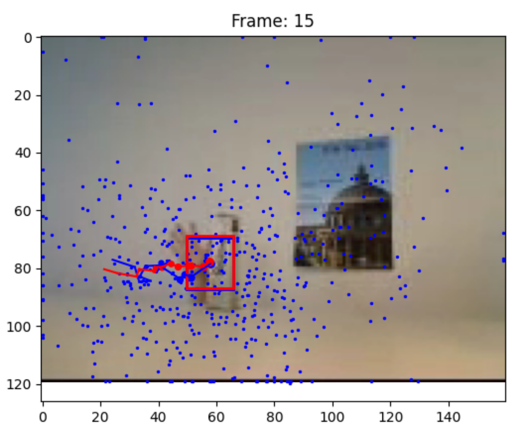


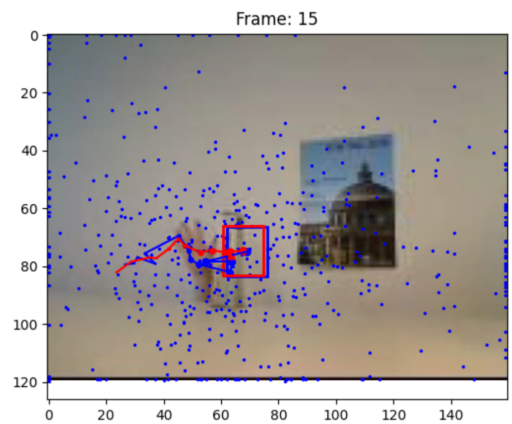
Questions

- **What is the effect of using a constant velocity motion model?**

In a constant velocity motion model, it is assumed that the object is moving with a constant velocity. For each time step, the particles move in the direction and magnitude of the constant velocity (plus some noise) regardless of the actual movement of the object.

If the object is moving with a uniform background, then adding a constant velocity greater (less) than the actual velocity of the object might cause the bounding box moving faster (slower) than the object, detecting only part of the object. Here is an example:



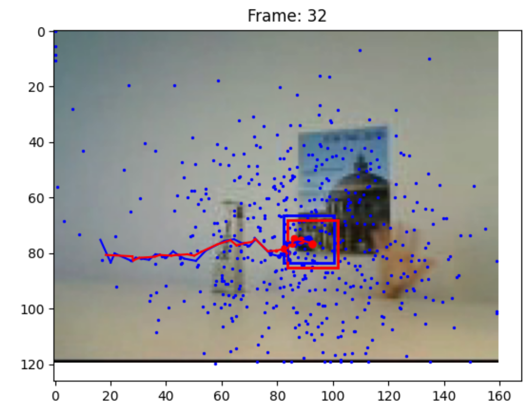


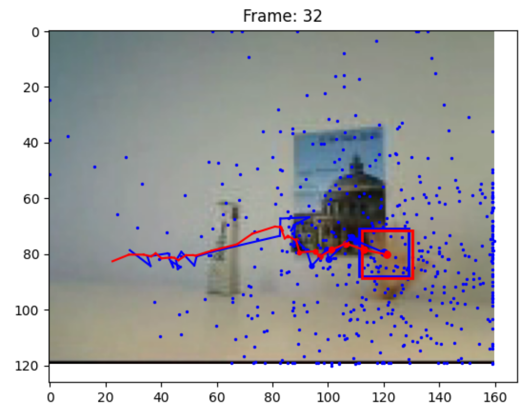
The left frame has model = 0, and the right frame has model = 1 with {sigma_velocity = 3, initial velocity = (15, 1)}, with other parameters:

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.5,
    "num_particles": 500,
    "sigma_position": 15,
}
```

It can be seen that in the left frame, the bounding box keeps tracking the moving hand, while in the right frame, the bounding box moves faster than the bounding box.

If the object passes through a background that has a color histogram similar to the object, then, without a constant velocity, the bounding box might be interfered by the background, falling behind the object. Here is an example:





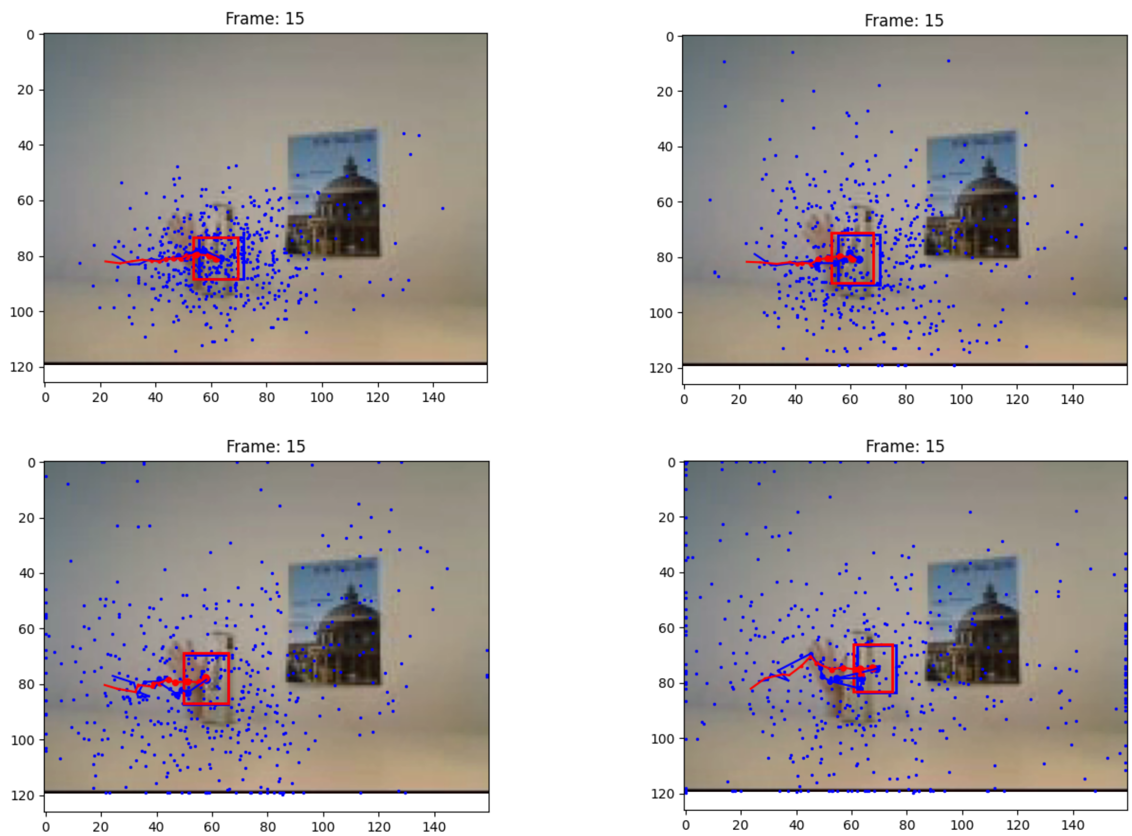
The left frame has model = 0, and the right frame has model = 1 with {sigma_velocity = 3, initial velocity = (15, 1)}, with other parameters:

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.5,
    "num_particles": 500,
    "sigma_position": 15,
}
```

It can be seen that in the left frame, the bounding box falls behind the moving hand, while in the right frame, the bounding box keeps tracking the moving hand.

- **What is the effect of assuming decreased/increased system noise?**

Increased (decreased) system noise leads to increased (decreased) sparsity of the distribution of the particles, making the particles spread over a larger (smaller) area. In particular, increased (decreased) position noise increases (decreases) the range of the particles' centers, while increased (decreased) velocity noise makes the distribution of the particles' centers more (less)



evenly.

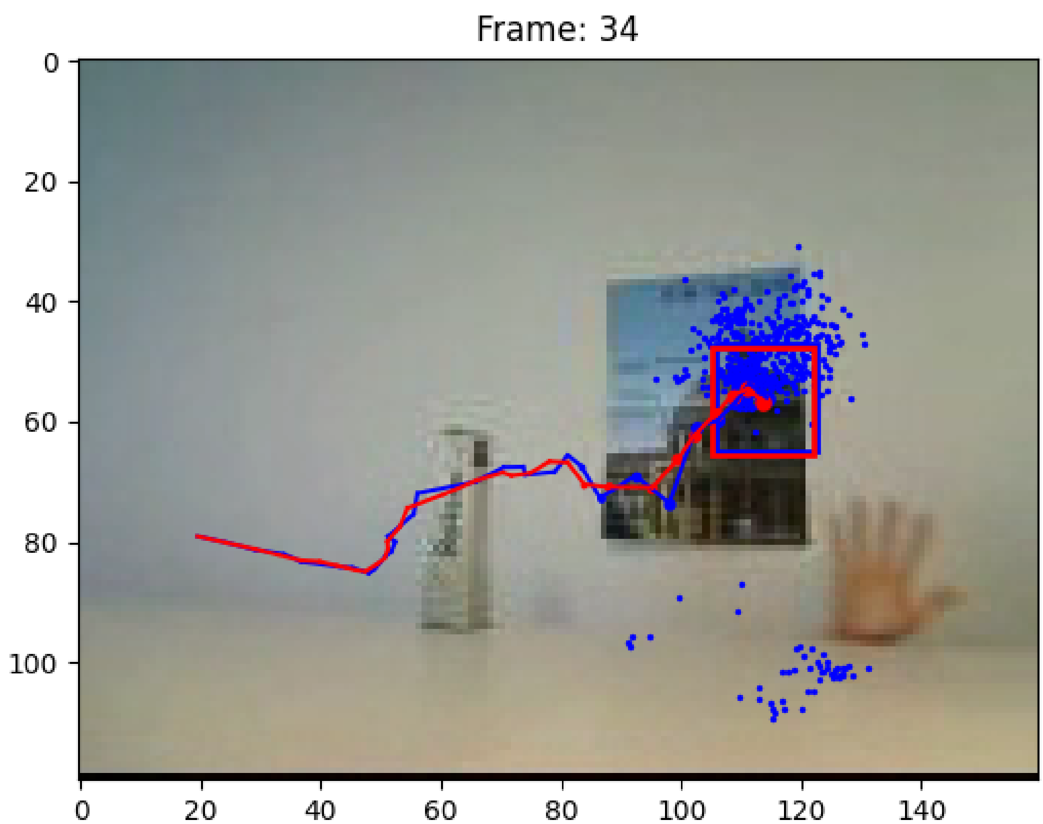
For illustration, in the above four frames, the top left frame has parameters {sigma_position = 10, sigma_velocity = 1}, the top right frame has parameters {sigma_position = 15, sigma_velocity = 1}, the bottom left frame has parameters {sigma_position = 15, sigma_velocity = 5}, the bottom right frame has parameters {sigma_position = 15, sigma_velocity = 10}, while all the other parameters of the frames are:

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.5,
```

```
    "model": 1,
    "num_particles": 500,
    "initial_velocity": (10, 1)
}
```

It can be seen that the range of the particles' centers is larger in the top right frame than in the top left frame, and the distribution of the particles centers are more evenly in the bottom right frame than in the bottom left frame or the top right frame.

If the system noise is too low, then the particles may be so concentrated that it could fail to cover the range of motion of the object, losing track of the object, especially when occlusion / clutter occurs:



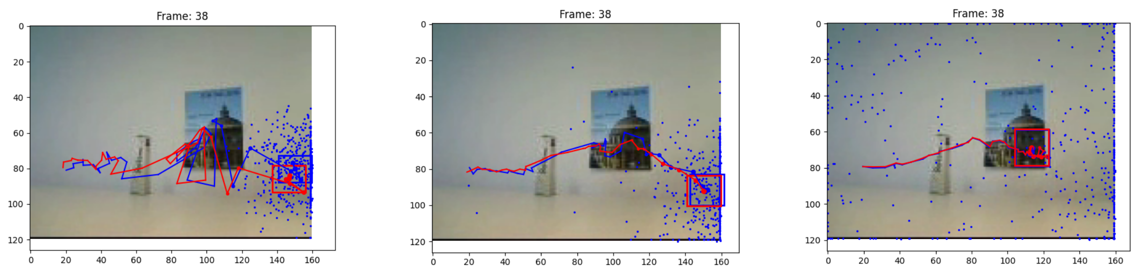
This frame has {sigma_position = 1, sigma_velocity = 0.5}, with other parameters:

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.5,
```

```
    "model": 1,
    "num_particles": 500,
    "initial_velocity": (5, 1)
}
```

- **What is the effect of assuming decreased/increased measurement noise?**

Increased (decreased) measurement noise gives higher (lower) weights to candidate particles with large costs. The higher the measurement noise, the more particles with large costs are preserved during resampling. In addition, the higher the measurement noise, the more similar the a priori mean state and the a posteriori mean state. Here is an example:



The left frame has sigma_observe = 0.1, the middle frame has sigma_observe = 0.5, and the right frame has sigma_observe = 2, with all the other parameters:

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "model": 1,
    "num_particles": 500,
    "sigma_position": 15,
    "sigma_velocity": 1,
    "initial_velocity": (5, 1)
}
```

When measurement noise is small, for example, sigma_observe = 0.1 as in the left frame, particles with large cost have small weights, and thus do not contribute much to the mean state. This is good when the background is uniform, as the color histogram around the object does not change much, and thus large cost is a reasonable indication of wrong estimation.

When the background is not uniform, as in video 2 with a poster on the wall, small measurement noise, such as sigma_observe = 0.1 as in the left frame, is not enough. As the color histogram around the object changes a lot when the object passes through the poster, if measurement noise is small, then the particles centered at the object will have high cost and small weights, contributing little to the mean state. This could result in the mean state deviating a lot from the object. See the trajectory in the left frame.

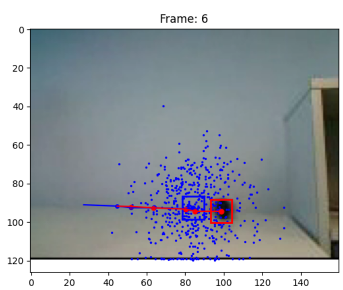
However, measurement noise cannot be too large. If the measurement noise is too large, for example, sigma_observe = 2 as in the right frame, the weights of particles with large cost will be so large that even those that are not centered around the object could contribute a lot to the mean state. This could also result in the mean state deviating a lot from the object. See the trajectory in the right frame.

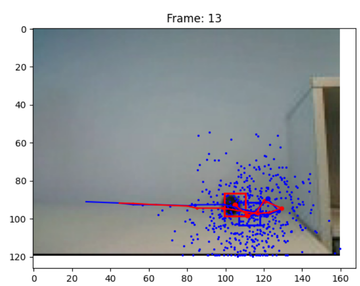
In the case of video 2, an appropriate measurement noise we chose is sigma_observe = 0.5.

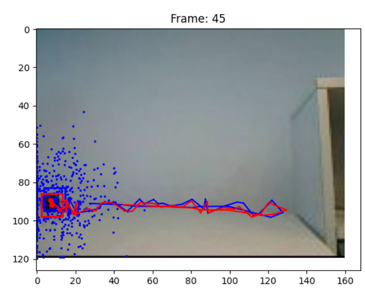
Video 3

Successful tracking

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.1,
    "model": 0,
    "num_particles": 500,
    "sigma_position": 15,
    "sigma_velocity": 3,
    "initial_velocity": (5, 1)
}
```

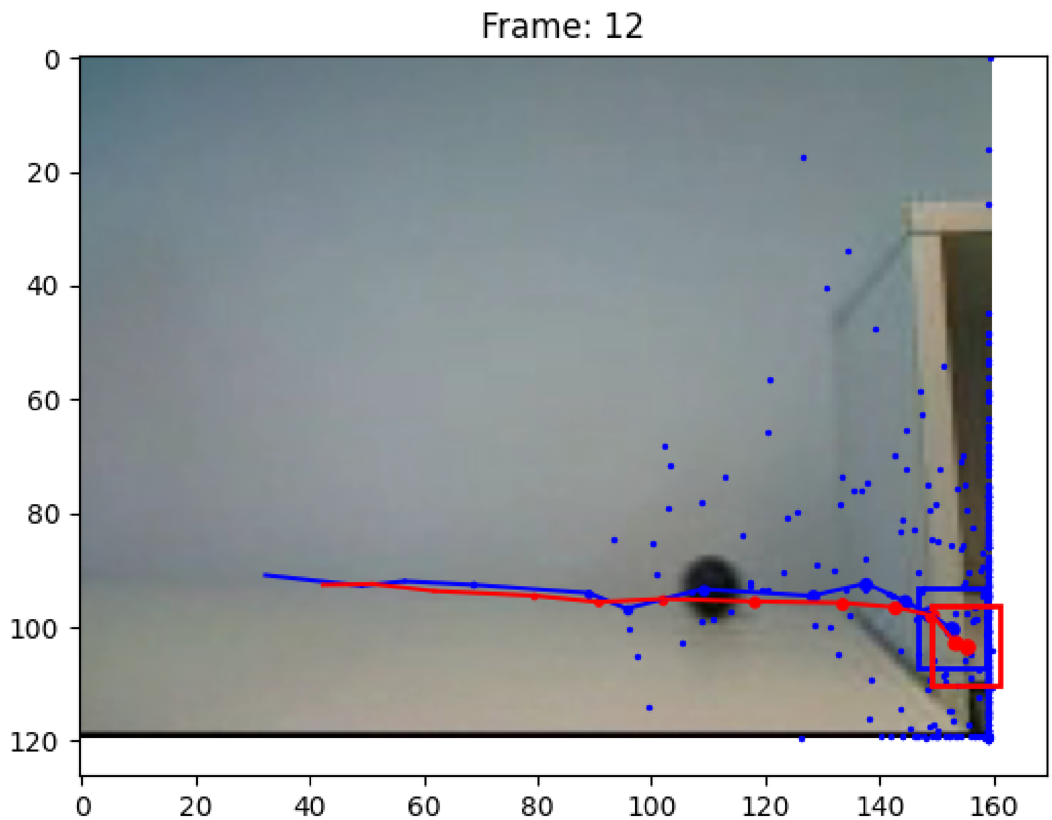






Unsuccessful tracking

```
params = {
    "draw_plots": 1,
    "hist_bin": 8,
    "alpha": 0,
    "sigma_observe": 0.5,
    "model": 1,
    "num_particles": 500,
    "sigma_position": 15,
    "sigma_velocity": 3,
    "initial_velocity": (5, 1)
}
```



Tracking with best params of video 2:

No, I cannot track the bouncing ball with the best params of video 2, because the sigma_observe used for video 2 is too large for video 3.

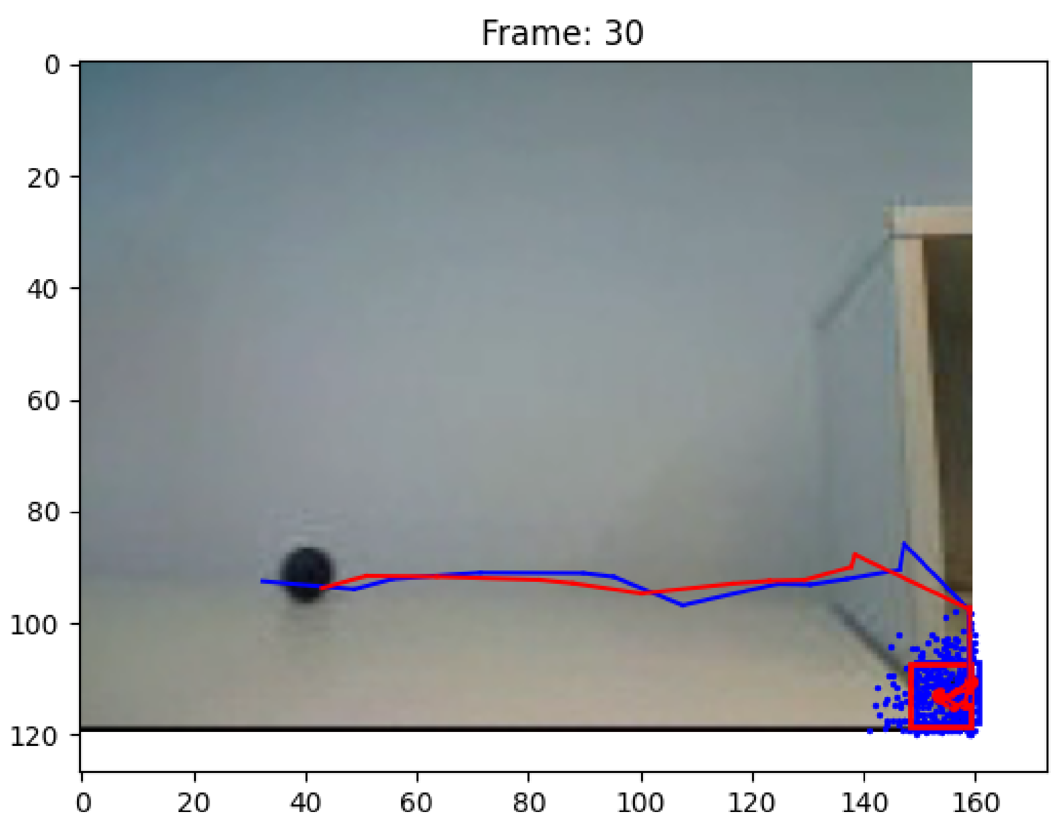
The result is presented in the unsuccessful case. By changing sigma_observe = 0.5 to sigma_observe = 0.1, the tracking is successful.

Questions

- **What is the effect of assuming decreased/increased system noise?**

The general effect has been described in Video 2, here only the effect specific to video 3 is discussed.

For video 3, the system noise cannot be too small. As shown in the successful tracking case, {sigma_position = 15, sigma_velocity = 3} works well. However, when we decrease the system noise to {sigma_position = 5, sigma_velocity = 1} with the other parameters unchanged, the tracker fails at tracking the ball when the ball bounces back. The reason could be that, when the system noise is too small, the particles are so concentrated that the model fails at covering the range of bouncing of the ball with sufficiently many particles.



- **What is the effect of assuming decreased/increased measurement noise?**

The general effect has been described in Video 2, here only the effect specific to video 3 is discussed.

As discussed in the previous sections, for video 3, the measurement noise needs to be small (in our case, we choose sigma_observe = 0.1). Since the background is uniform, if the measurement noise is large, the weights of particles with large cost will be so large that even those that are not centered around the object could contribute a lot to the mean state. This could also result in the mean state deviating a lot from the object. See the trajectory in the unsuccessful case with sigma_observe = 0.5.

Questions

- **What is the effect of using more or fewer particles?**

Using more (less) particles gives a more (less) stable solution, as the mean state is determined by averaging over all particles' states.

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

Using more bins does **not** always give better tracking. It depends on what elements are there in the frame.

When the background is uniform, but there are elements similar to the object, it is better to use more bins. For example, in video 1, to distinguish the hand from the arm, it is better to use more bins.

When the background is not uniform, such as in video 2, it is better to use fewer bins. When the hand passes through the poster, the color histogram around the hand changes a lot. In this case, if we use more bins, the color histogram of the hand with the poster in the background may differ much from the target histogram, and thus will be assigned a low weight.

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

Advantage: less complexity in time and space.
Disadvantage: feature not representative enough in complicated cases such as occlusion, clutter etc.