Computer Vision Report

Exercise 9

____

Amrollah Seifoddini

Task 1)

In this task, I followed the straightforward guides provided by exercise description for implementing the missing parts of Condensation algorithm. For color histogram, I loop through every pixel and divide its value in each color channel to the number of bins, then I increment by 1, the value in the bin that each color components fall into.

I chose matrix A, as [1 0; 0 1] for the mode 0 (just noise) and as [[1,0]',[0,1]',[1,0]',[0,1]'] for mode 1 (constant velocity). Because for the mode 0, we only have x,y and a A matrix should preserve particle locations, so the only option is identity matrix. And for the mode 1, particles should move with a constant velocity, so I chose A to add velocity components of each state to its current location.

In propagation part, I use normrnd function to generate random noise for position and velocity. Then I apply the A*p+noise formula to get the new particle location. If it falls outside of image, I will generate a new noise until the result fit into frame. In case of mode 1, I also set the velocity components with the sum of initial velocity and velocity_noise.

In observation part, I compute color histogram for each image patch with particle in its center, then I calculate the chi2-squred distant between each histogram and target histogram. Based on these distances, I set the weight of particles by using the formula provided by exercise slides. At the end, weight will be normalized so that their sum equals to 1 again.

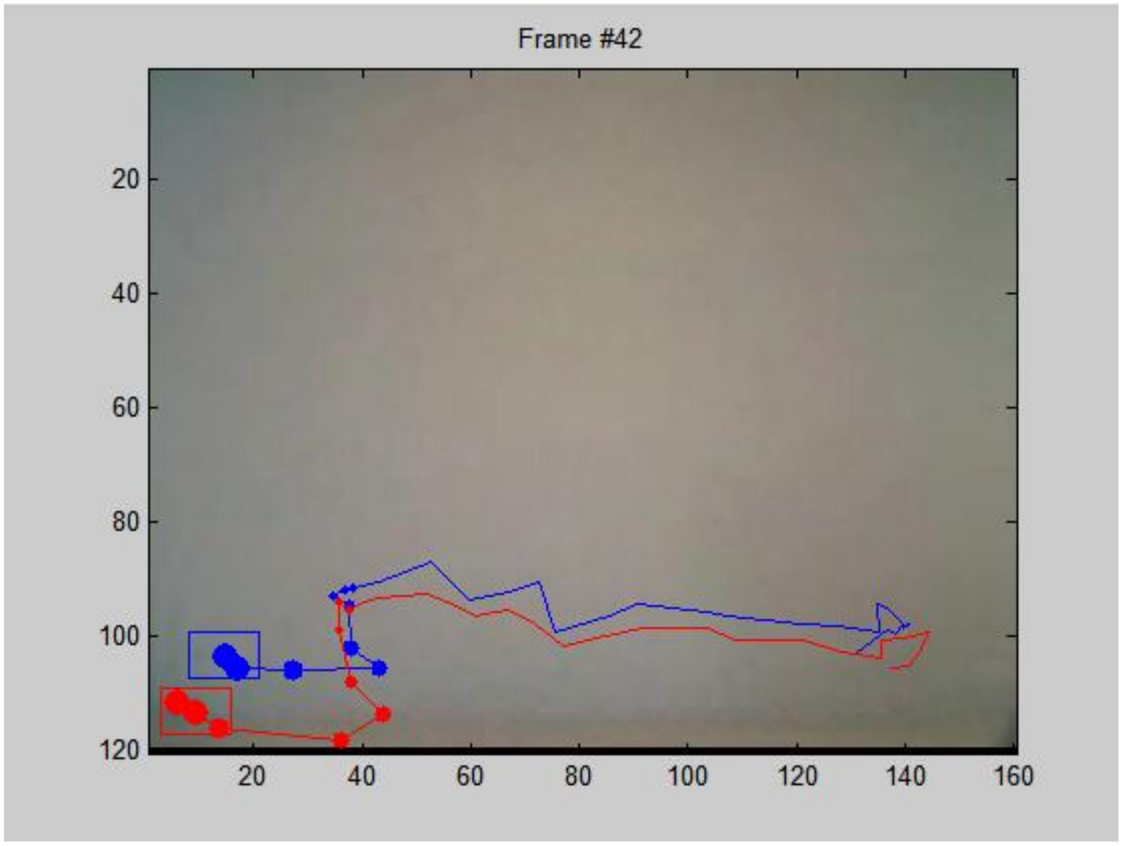
In estimate part, I just compute the weighted average of states by using their weights.

In resample part, I simply use randsample function of Matlab to draw a random sample out of current particles based on their weight as a probability distribution. When I have the index of selected particles, I form the new particle and particle weights matrixes from original ones. At the end, I normalize particle weights so that sum of them equals to one again.

Task 2)

In this task, I performed several experiments with different settings.

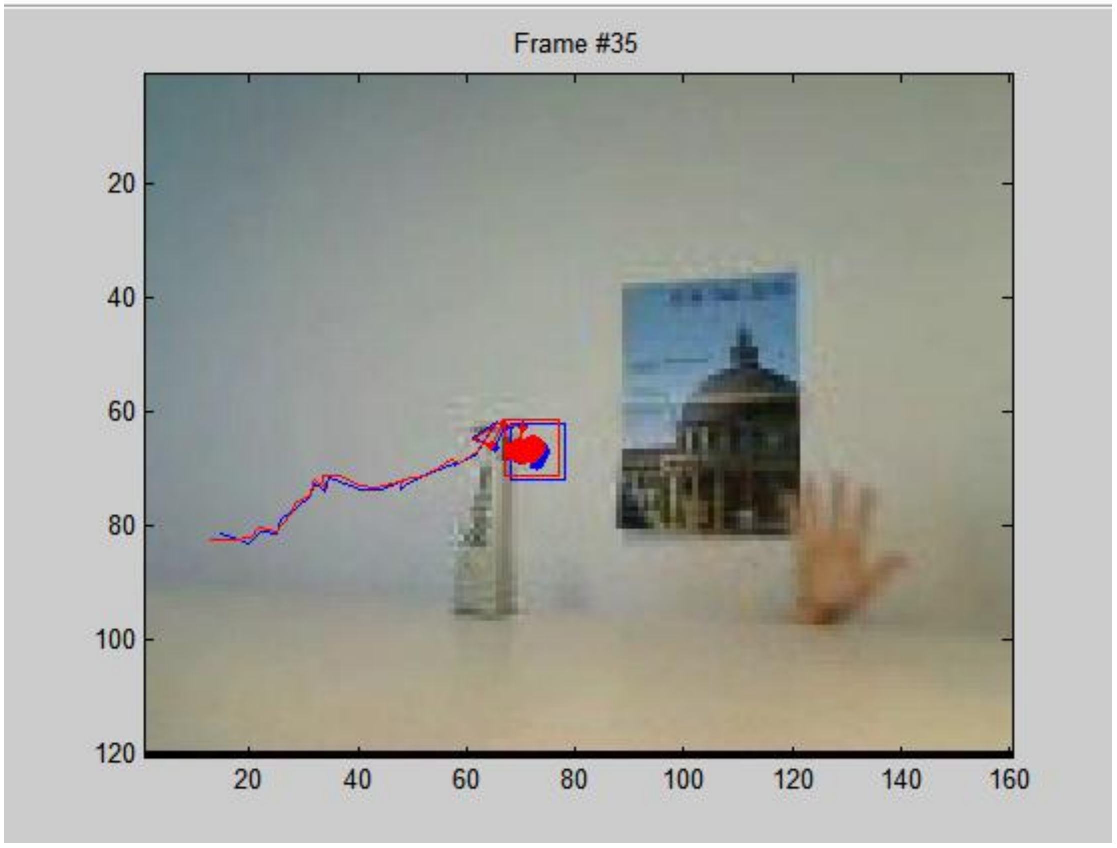
a: Here is my best result with video1:



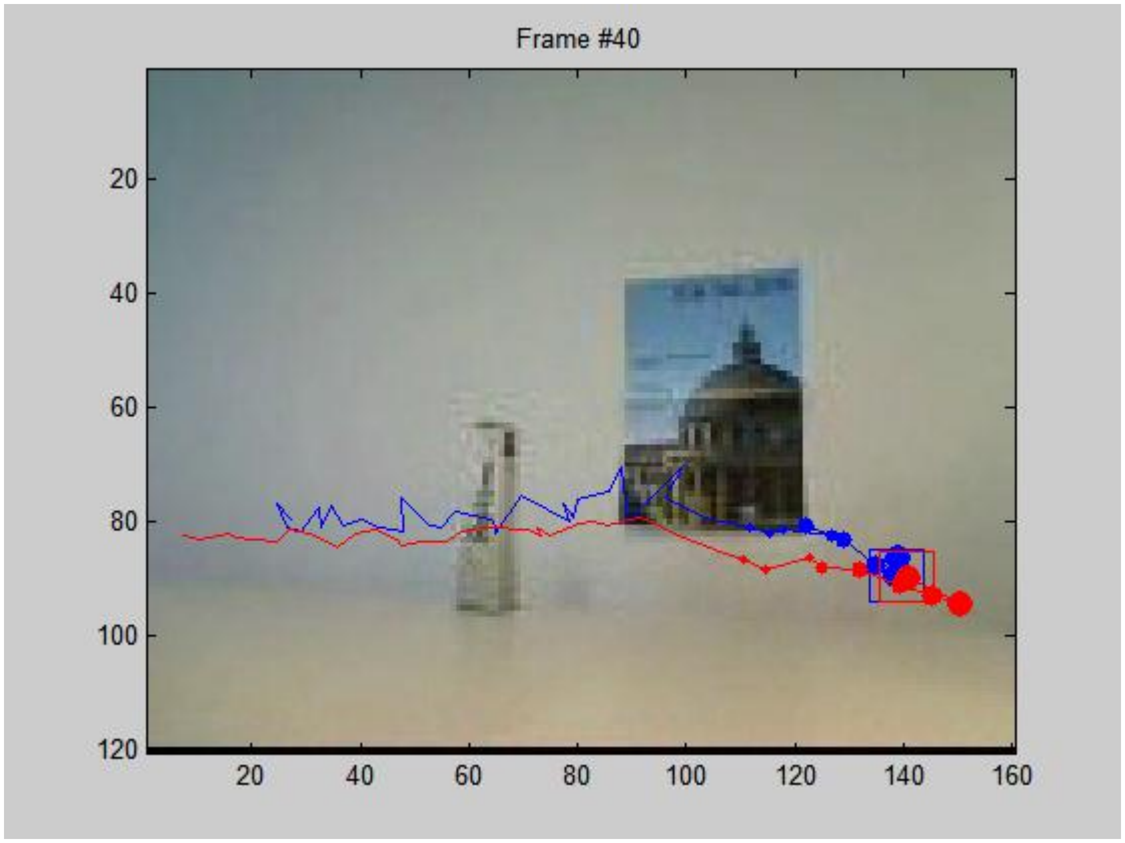
As you can see, the mode 0 (only noise) work very well here, and there was no need to use a velocity. I think it is because the hand moves very slow and smooth, so our limited noise can cover its movement area. At the end of video, when the hand is out of frame, the algorithm gets confused and make a weak hypothesis about location of hand which is not correct. My first attempt was successful, so I don't have an unsuccessful track to show.

b: Here is my result for tracking hand in video2:

At first, I used mode 0 for this video but it failed at the points where hand occluded and moves in front of ETH image on the wall.



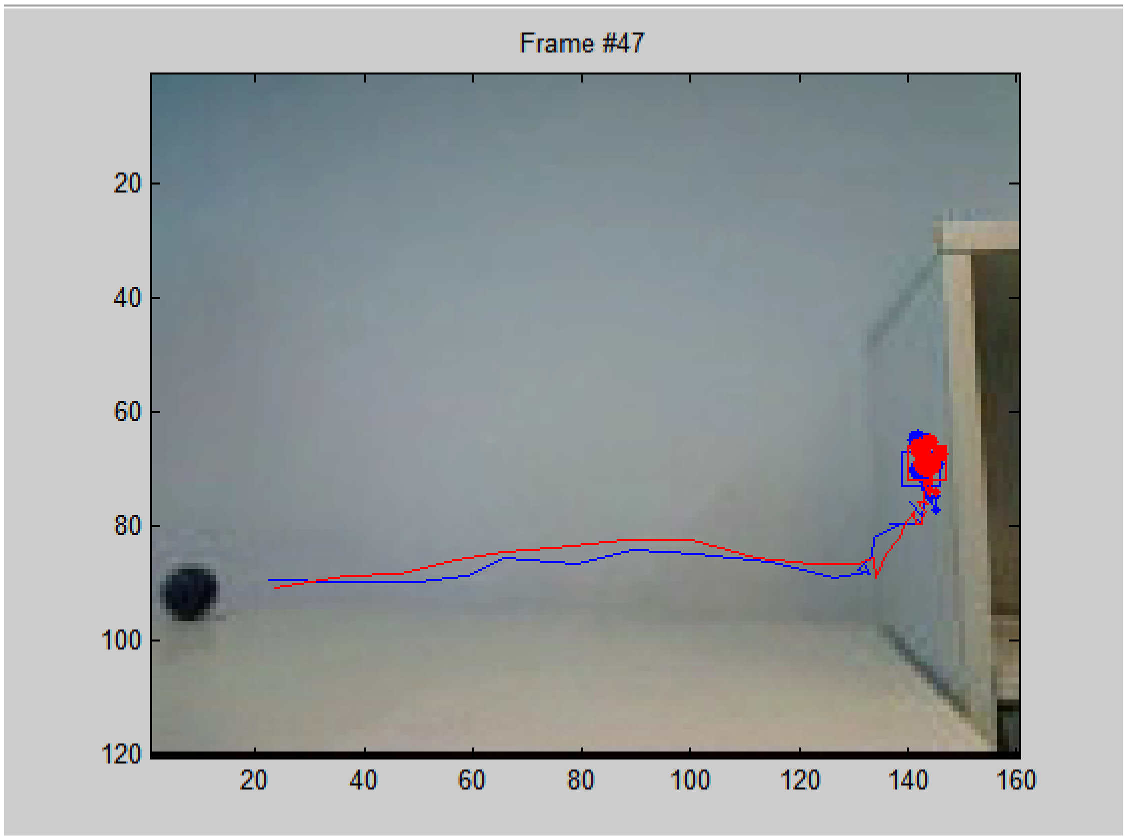
I guess the histogram of hand and bottom of ETH picture is almost the same, and makes the system to get confused about exact location of hand. But then I switched to mode 1 with an initial velocity of [8, 0] and sigma_velocity of 1. I also increased both of the system noise and measurement noise. Now, the algorithm can effectively track the hand even in case of occlusion and clutter background.



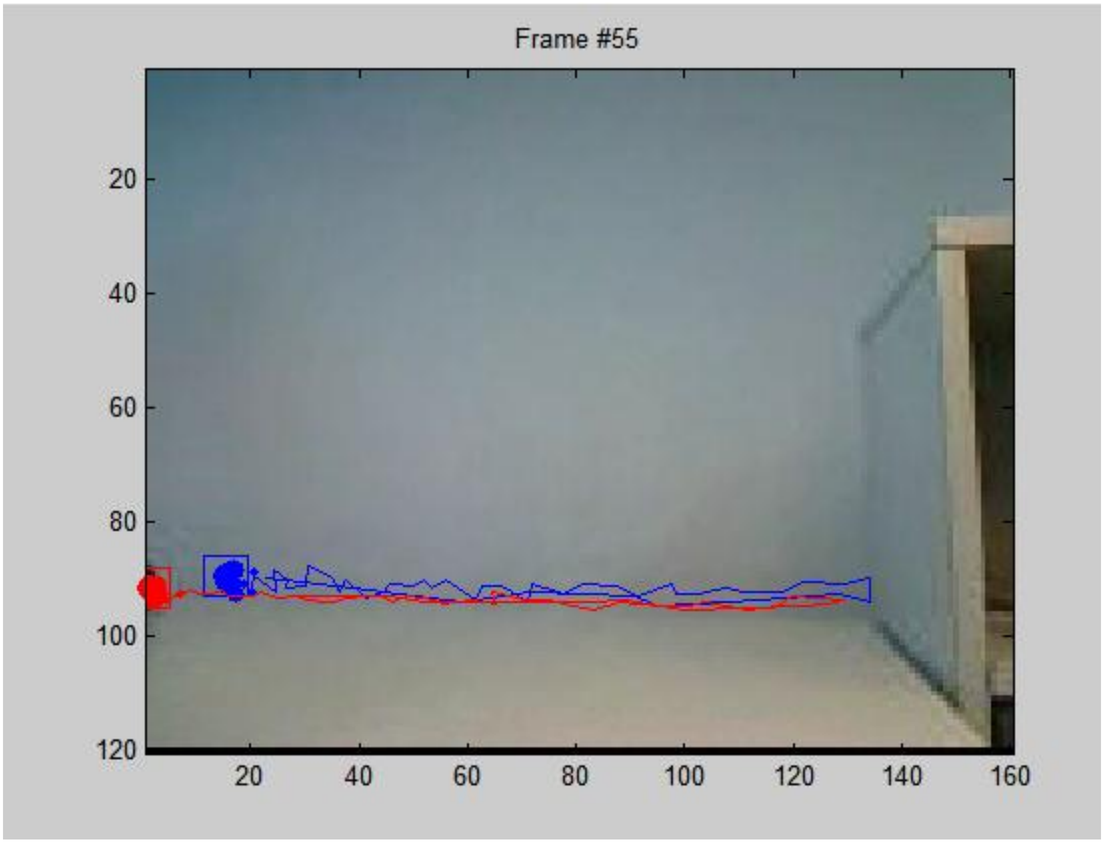
The effect of using constant velocity is somehow obvious, because now algorithm have a better estimate about where to look for hand. For example, even in front of ETH picture, system doesn't stalk. It will move in direction of velocity to find matches for histogram of hand. The increased system noise help system to recover from points that it loses the hand for a few frames. For example, in front of ETH picture, the histograms are very similar and particle weights are very close, but the high system noise allows considering some other points outside of this ambiguous patch. System hopes to find a strong hypothesis in those point and move towards them. The increased measurement noise has the same effect because it brings more randomness to setting particle weights. It means that we are less reliant on histogram distances for selecting new particles and this helps in situations where background is similar to target or the cases where appearance of target changes for a few frames. The decreased system noise and decreased measurement noise has the opposite effect than what I explained. It means less robustness to occlusion and clutter background. Of course it is a tradeoff and we cannot increase the noise too much, because then the noise take over system hypothesis and the results have highly random trajectory instead of smooth movement.

c: Here is my results for video3:

At the first, I tried to track ball with settings of video2, but it failed.

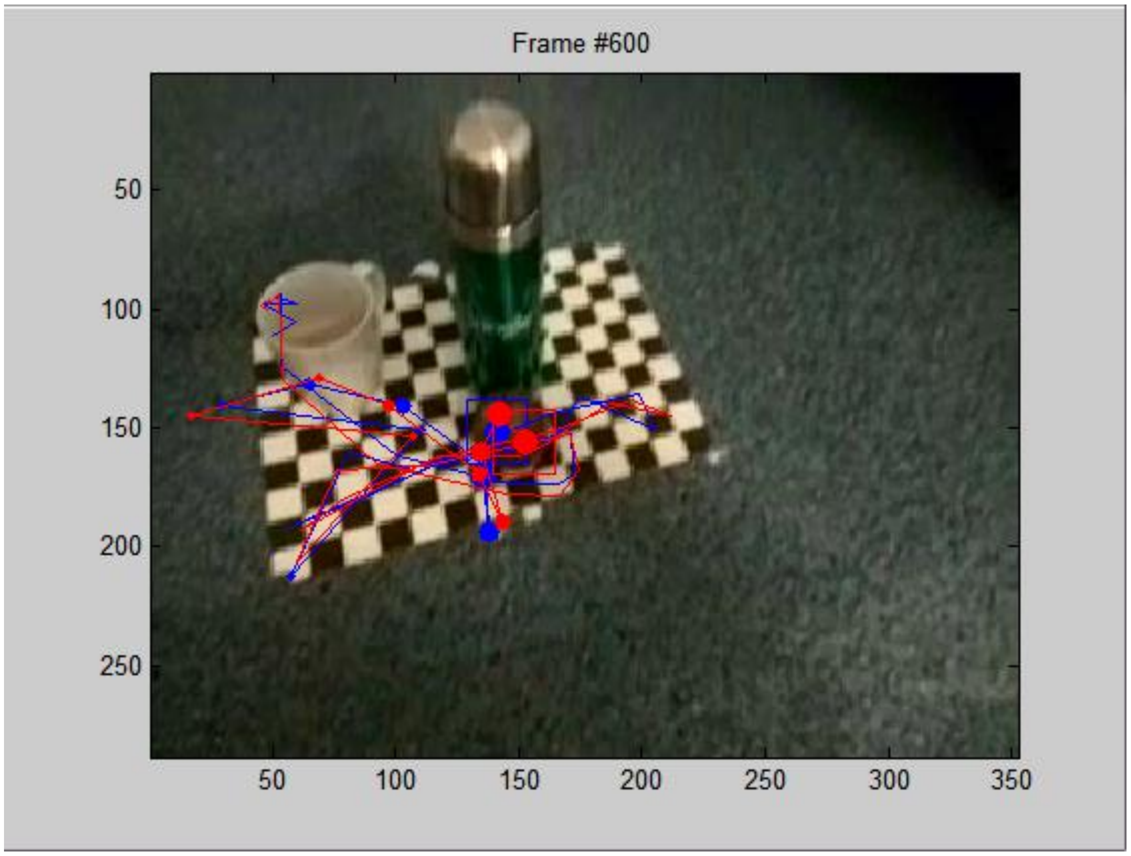


Then I increased the measurement noise, decreased the system noise a little bit and increased the sigma_velocity until I could track the ball nicely even in odd situations like hitting the wall and coming back.



I think the effect of system and measurement noise is to recover from situation of hitting the wall. But for locating the more exact position of ball, we should not increase system noise too much. The effect of using constant velocity in very important here because the ball moves fast and we cannot capture it with only noise. But the sigma_velocity should be high enough to allow for tracking ball when it hits wall and changes its velocity to other direction.

Task 3)



For this part, I used my own video which is recorded with my mobile camera. I am tracking the apple here with several occlusions and appurtenance changes in front of clutter background and shaky camera. The video is lengthy and I had to show trajectory of only a small part of it, but you can run the code and see the tracking for whole video.

What is the effect of using more or fewer particles?

The more particle means decreasing the effect of noise, because we are setting the new position based on average position of particles. But it is also a tradeoff. If we use very small number of particles, the estimation suffers from lack of good hypothesis sometimes, because there is a chance that good ones are not selected in previous step. On the other hand, if you use way too many particles, it will diminish the effect of goof particles in estimating the final hypothesis. I mean in the averaging procedure, the contributions of irrelevant particles may lead us to some bad estimation.

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

I tried several bin numbers. The effect of using more bins is that it will increase the precision of histograms. So, the distances of histograms which were close in 8 bins may be a little more in 16 bins. The effect of fewer bins is that it somehow, decreases the noise of appearance change for the target, but it is also a tradeoff, because with too few bins, most of histograms get close to each other and weigh of particles get almost the same in clutter background.

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

I also tried this out. It allows for recovering from illumination changes and some small appearance changes. But it should be kept very small, because otherwise there is risk for losing original model after a few frames.