Report: CONDENSATION Tracker

Computer Vision - Assignment 6

Ian SCHOLL

ischoll@student.ethz.ch

1. CONDENSATION Tracker based on Color Histograms (50%)

1.1. Color histogram (10%)

To calculate the histogram I have tried different versions and methods, like *histogramdd* in the *numpy* library, and *calcHist* in the *cv2* library. However, in the end they calculate the same as my proper implementation, which creates a *hist_bin x hist_bin x hist_bin* 3D matrix, containing the counts of all $hist\_bin^3$ possible colors in the patch of interest. In the end, I normalize the histogram for it to be numerically stable and scalable.

1.2. Derive Matrix A (5%)

i. *no motion at all i.e. just noise*

In this case we completely ignore the velocities and the state vector s reduces to its first two elements as: $s = [x, y]^T$. Therefore, we expect for the patch to just stand still and have the equation: $s_t' = Is_{t-1} + w_{t-1}$, where w is the added stochastic noise. Therefore we have in this case:

$$A = I_{2x2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

ii. *constant velocity motion model*

This time we don't consider the patch stationary and therefore have to take into account the velocities in the state vector of form: $s = [x, y, \dot{x}, \dot{y}]^T$. For each timestep, x and y now have to move by the quantity given in $\dot{x}$, $\dot{y}$. Therefore, for instance in x we want at each timestep: $x_t = x_{t-1} + \dot{x}_{t-1}$, and the same in y. This can be achieved for the total state vector by using an overall A matrix of:

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

1.3. Propagation (10%)

This step is now where we implement the equation $s_t' = As_{t-1} + w_{t-1}$. For w I use the *np.random.normal* method in order to sample elements following the gaussian distributions given by zero mean and the standard deviations on position and velocity. One just has to differentiate between the case with no motion, where the state vector is reduced to 2 elements, and the one with constant velocity and 4 elements in the state vector.

Finally, I also implemented a vectorized check to make sure each new patch center will also end up within the frame of the picture; if that's not the case I move it to its closest border of the picture.

1.4. Observation (10%)

Here again I have tried several implementations. For me it has worked best to take over the already present call to *color_histogram* in the main loop of the *condensation_tracker* method, but my other proper implementation achieves very similar results. Then, one has to calculate the weights, which is just implementing the already given formula *(6)* for $\pi^{(n)}$ of the assignment instructions.

1.5. Estimation (5%)

This part again was very straightforward. Formula *(8)* of the assignment instructions can be calculated as a matrix multiplication between the particle coordinates and their respective weights, in order to receive a weighted average estimate on the new position of the object to be tracked.

1.6. Resampling (10%)

To re-sample, we want to choose more weighted particles more often. Therefore, I used the *np.random.choice* method to choose indices at random with the given particle weights as a probability distribution. Then I replace both the particles and their weights by their respective counterparts at those random indices. Most likely, particles weighted more will be picked several times and particles weighted less might never be picked, this shifts the average even more towards the highly weighted region where the object to be tracked most probably is to be found.

2. Experiments (50%)

2.1. video1.wmv (10%)

The following experiments have been conducted:

Exp. #	**hist**$_{bin}$	**α**	**σ**$_{observe}$	**model**	**#particles**	**σ**$_{position}$	**σ**$_{velocity}$	**initial vel.**		**Observation**
								x	**y**	
1	16	0	0.1	0	300	15	-	-	-	Hand gets roughly tracked, but not very closely and the estimates drift off in between
2	16	**0.8**	0.1	0	300	15	-	-	-	Almost perfect results, the hand gets tracked very well, *a posteriori* mean always a bit more accurate and 'ahead' of *a priori* one
3	16	0.8	**1**	0	300	15	-	-	-	Not successful, hand only gets sort of tracked in the beginning, then lost; the sampling here is probably too sparse now
4	16	0.8	0.1	0	**600**	15	-	-	-	Medium successful, as the motion gets tracked, but the many particles cause for the model to be distracted and track the arm rather than the hand itself
5	16	0.8	0.1	0	**150**	15	-	-	-	Not successful, hand only gets sort of tracked in the beginning, then lost; the sampling here is probably too sparse now
6	16	0.8	0.1	0	300	**20**	-	-	-	Also successful, but no clear improvement
7	16	0.8	0.1	0	300	**10**	-	-	-	Also successful, but no clear improvement
8	**64**	0.8	0.1	0	300	15	-	-	-	Medium successful, as the motion gets tracked, but the more colors cause for the model to be distracted and track the arm rather than the hand itself
9	**8**	0.8	0.1	0	300	15	-	-	-	Still successful, but a bit more 'laggy'
10	16	0.8	0.1	**1**	300	15	**1**	**2**	**-5**	Also successful, but no clear improvement

11	16	0.8	0.1	**1**	300	15	**10**	2	-5	Also successful, but a lot more ‘shaky’
10	16	0.8	0.1	**1**	300	15	**0.1**	2	-5	Also successful, but no clear improvement

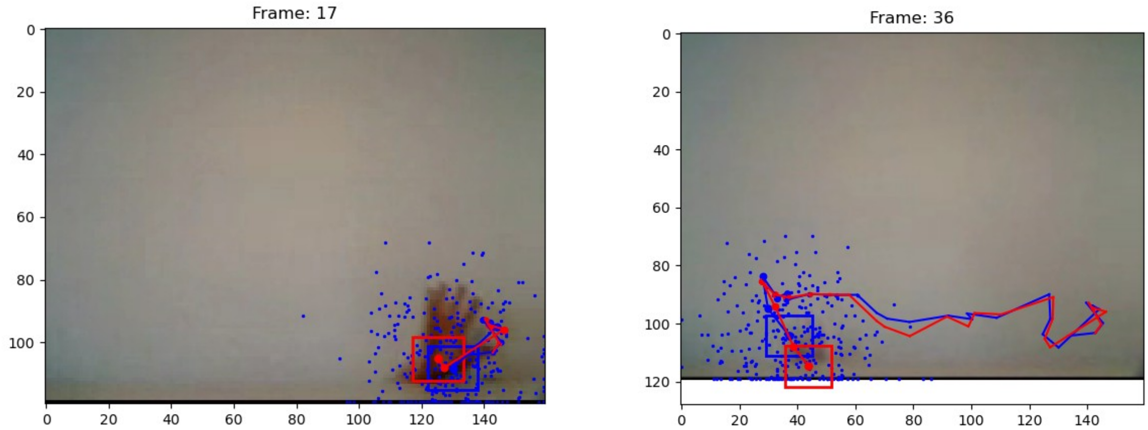

Figure 1: Experiment 1, medium successful, with parameters given in initial skeleton code

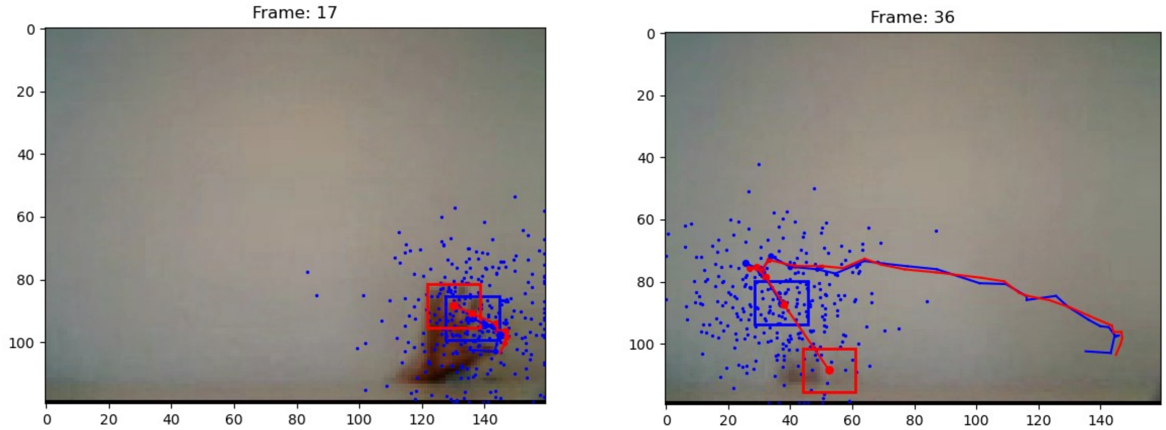

Figure 2: Experiment 2, very successful after increasing α

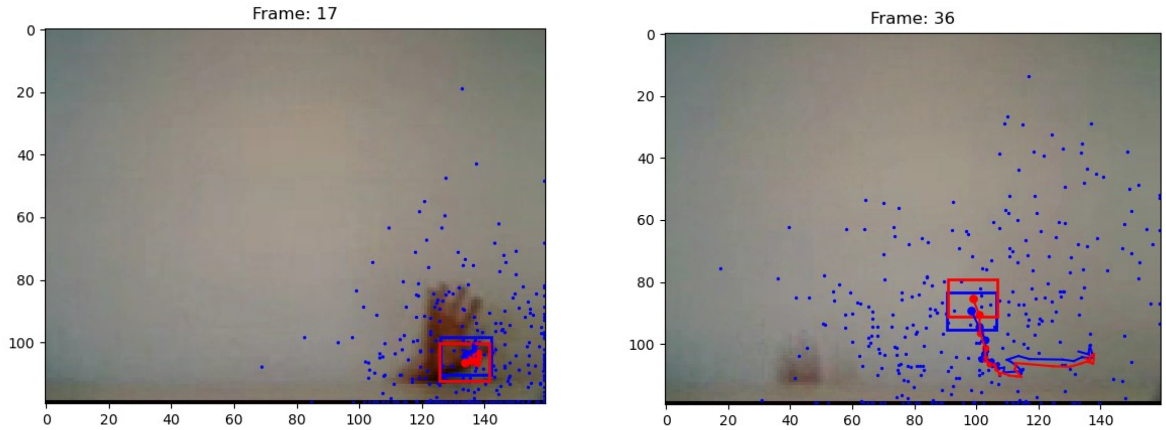

Figure 3: Experiment 3, not successful after increasing $\sigma_{observe}$

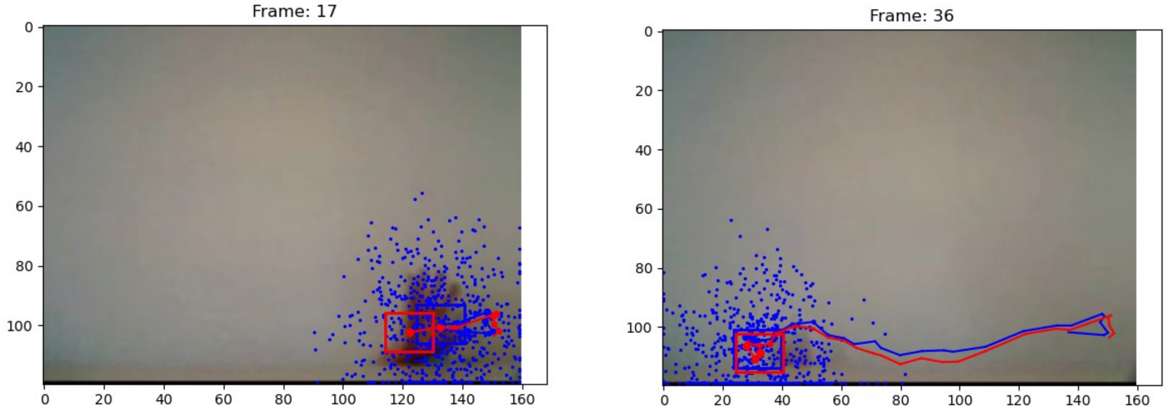

Figure 4: Experiment 4, medium successful after increasing number of particles

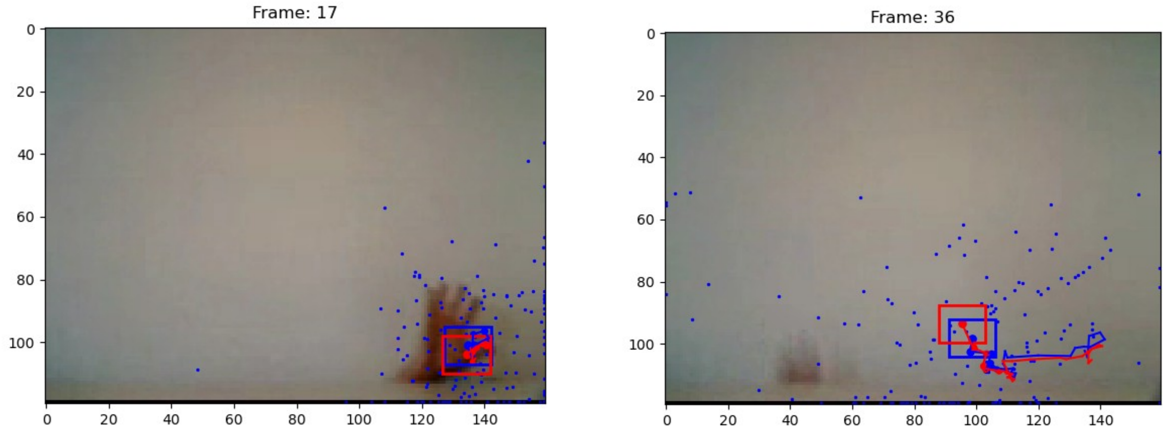

Figure 5: Experiment 5, not successful after decreasing number of particles

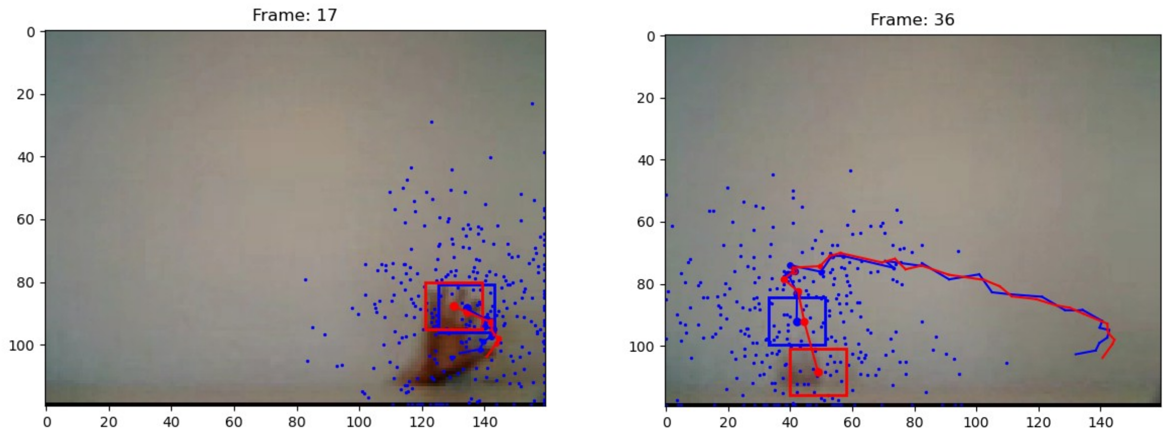

Figure 6: Experiment 6, still successful after increasing $\sigma_{position}$

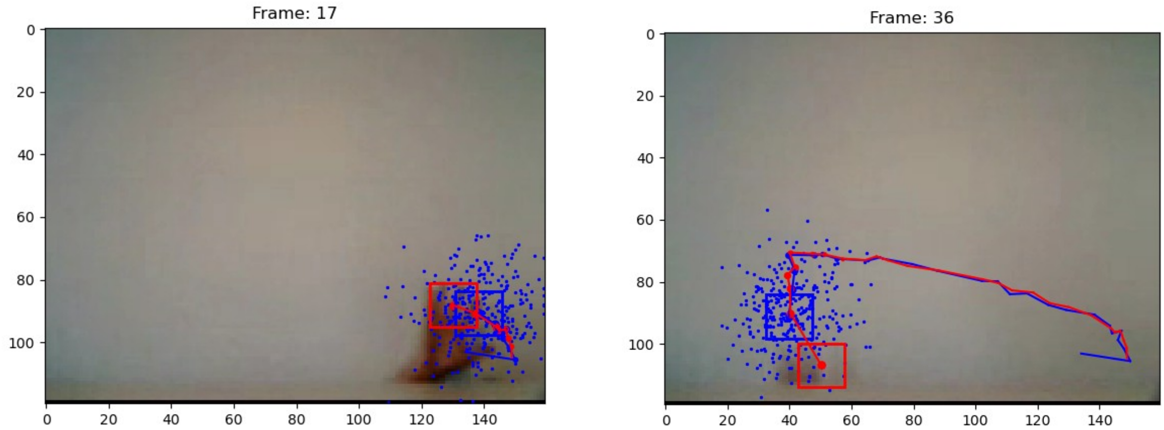

Figure 7: Experiment 7, still successful after decreasing $\sigma_{position}$

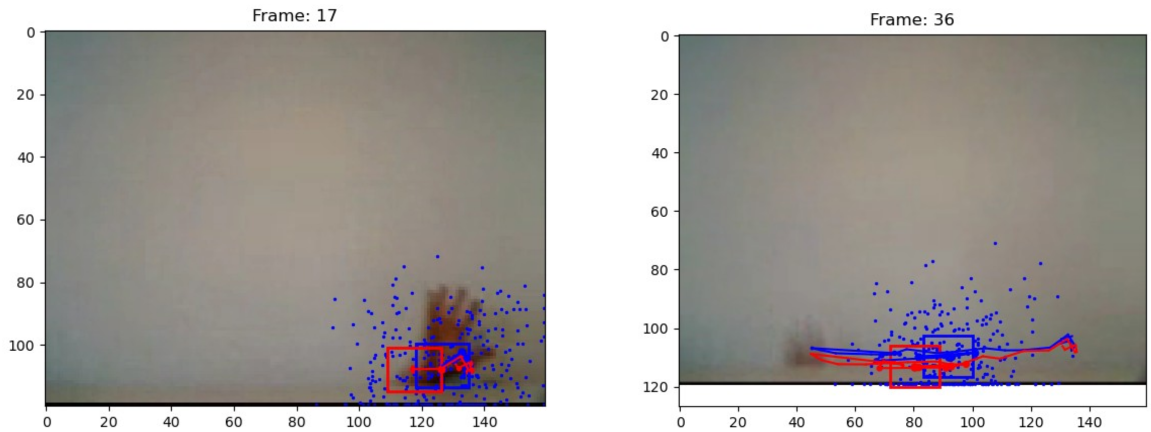

Figure 8: Experiment 8, medium successful after increasing $hist_{bin}$

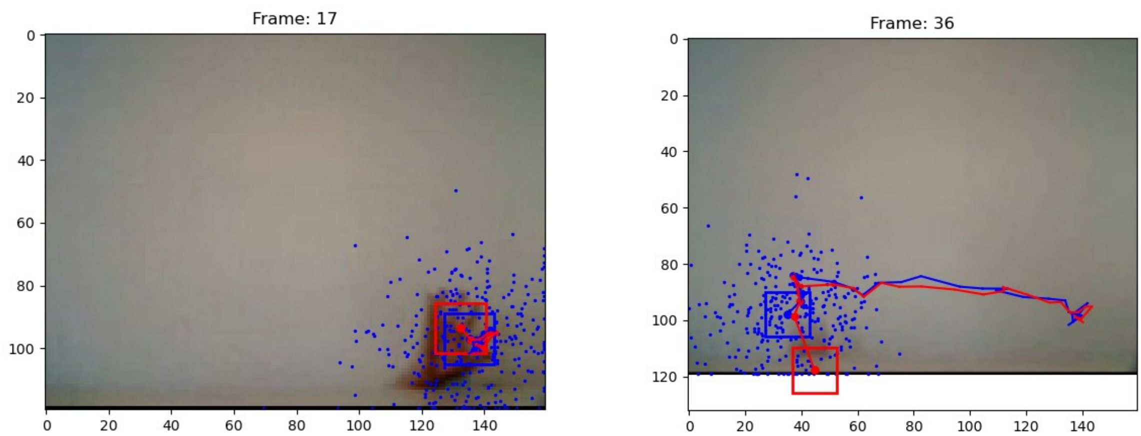

Figure 9: Experiment 9, medium successful after increasing $hist_{bin}$

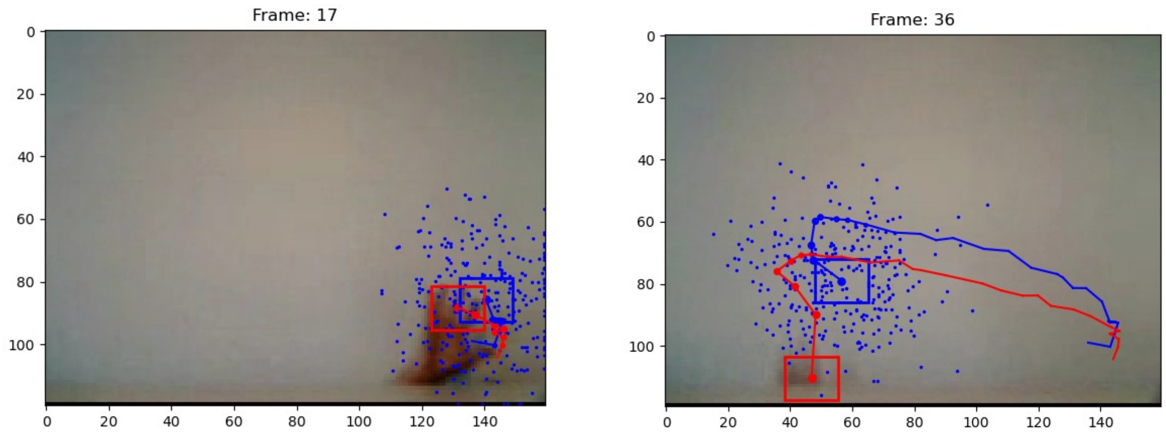

Figure 10: Experiment 10, still successful using constant velocity model

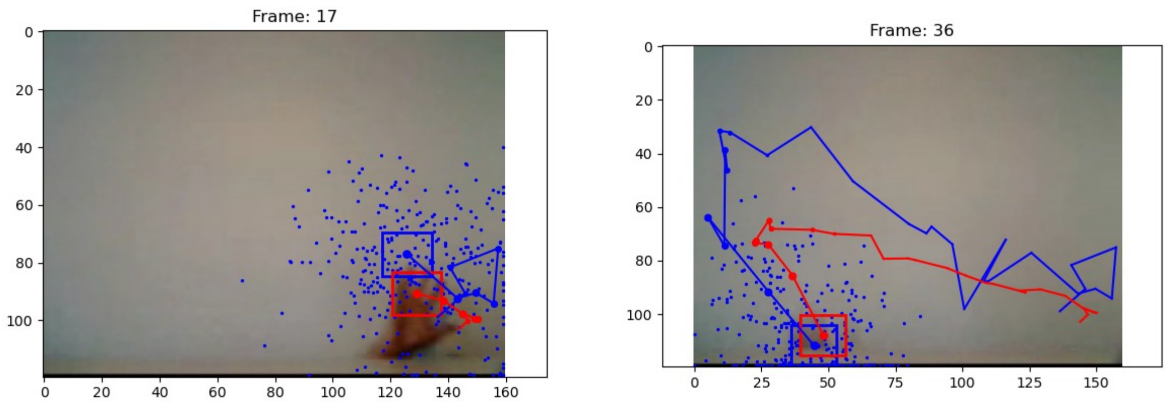

Figure 11: Experiment 11, medium successful after increasing $\sigma_{velocity}$

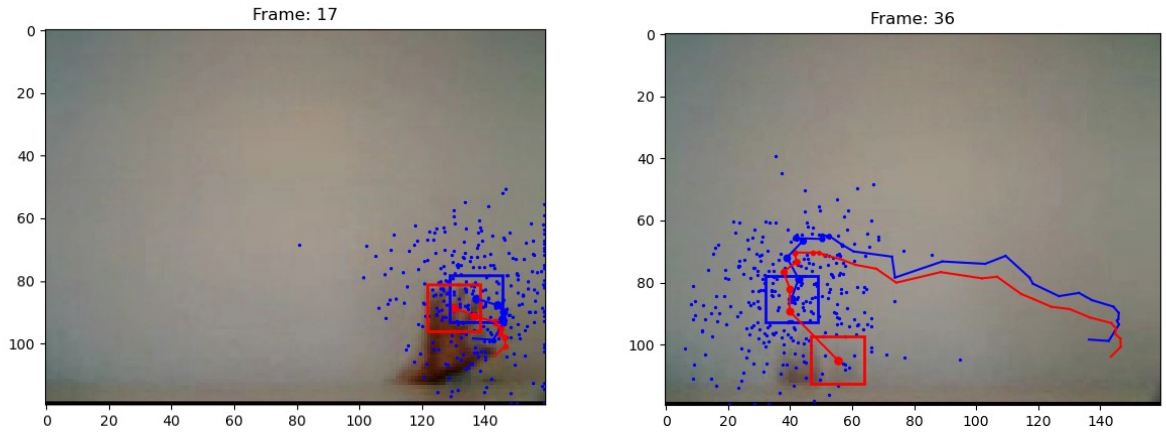

Figure 12: Experiment 12, still successful after decreasing $\sigma_{velocity}$

2.1.1. Conclusions

The hand can be tracked very well just with minor tuning of the parameters. Therefore, the code seems to be working without bugs and to be delivering the expected results.

2.2. video2.wmv (20%)

I started the experiments with the best parameters from the previous video 1:

Exp. #	**hist$_{bin}$**	**α**	**$\sigma_{observe}$**	**model**	**#particles**	**$\sigma_{position}$**	**$\sigma_{velocity}$**	**initial vel.**		**Observation**
								x	**y**	
13	16	0.8	0.1	0	300	15	-	-	-	Hand gets tracked in the beginning, but lost after the occlusion behind the milk carton
14	16	**0.2**	0.1	0	300	15	-	-	-	Better, medium successful, finds it after milk carton but still loses the hand eventually
15	16	0.2	**1**	0	300	15	-	-	-	No improvement, still can't overcome occlusion and background
16	16	0.2	0.1	0	**600**	15	-	-	-	Successfully overcomes the occlusion and background and tracks hand to the end
17	16	0.2	0.1	0	**150**	15	-	-	-	No improvement, can't fully overcome occlusion and background
18	16	0.2	0.1	0	600	**20**	-	-	-	Also successful, a bit smoother
19	16	0.2	0.1	0	600	**10**	-	-	-	No improvement, can't fully overcome occlusion and background
20	**64**	0.2	0.1	0	600	20	-	-	-	Not successful, gets distracted with background and loses hand
21	**8**	0.2	0.1	0	600	20	-	-	-	Not successful, gets distracted with background and loses hand
22	16	0.2	0.1	**1**	600	20	**1**	**5**	**0**	Also successful, but more 'shaky'
23	16	0.2	0.1	**1**	600	20	**10**	5	0	Also successful, but even more 'shaky'
24	16	0.2	0.1	**1**	600	20	**0.1**	5	0	Also successful, but more 'shaky'

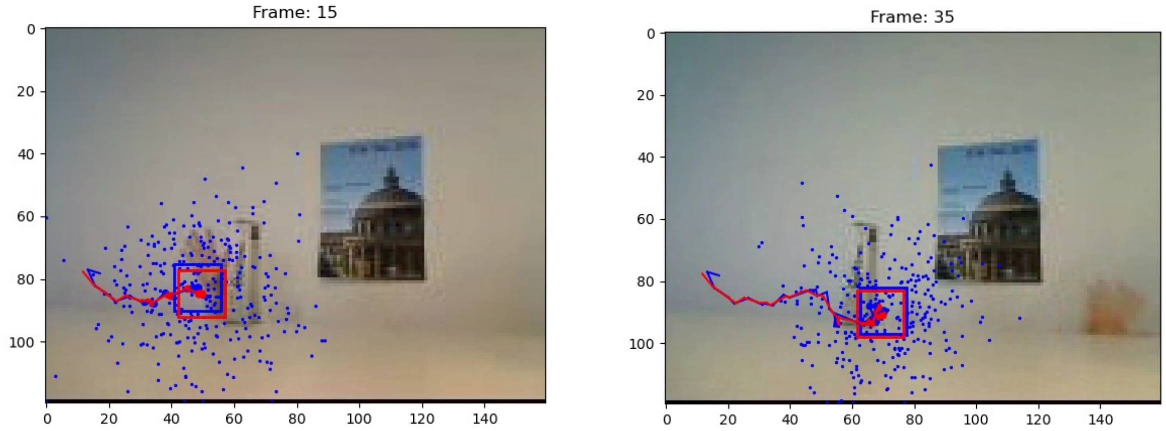

Figure 13: Experiment 13, not successful after the occlusion

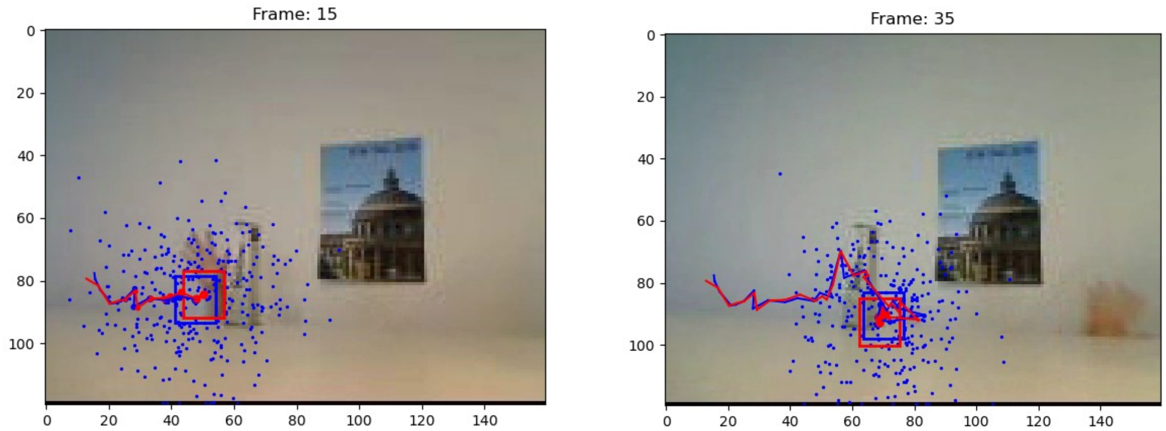

Figure 14: Experiment 14, more successful after the occlusion with decreased α

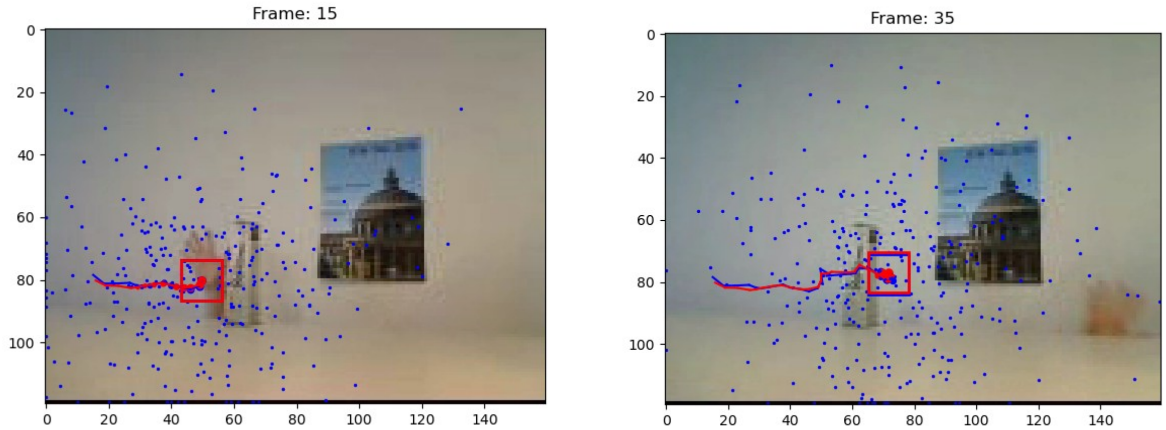

Figure 15: Experiment 15, not more successful after the occlusion with increased $\sigma_{observe}$

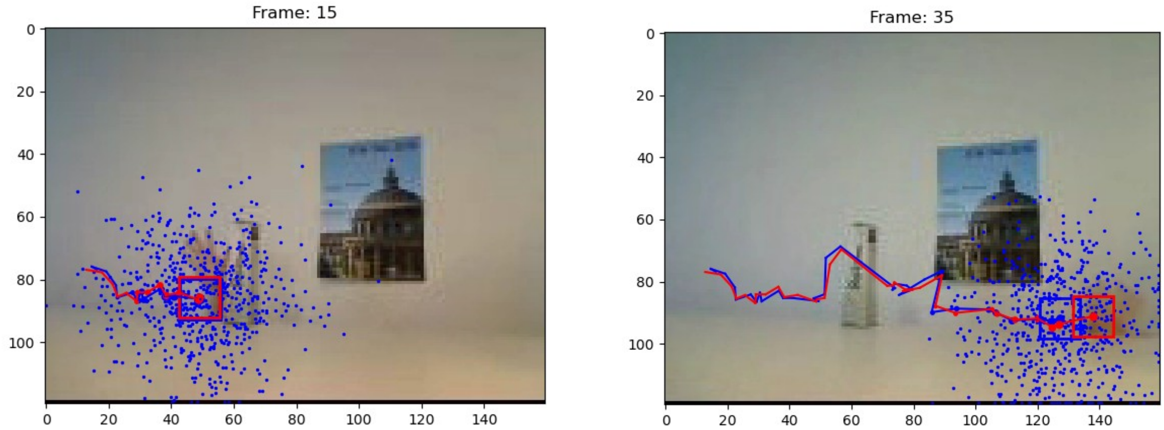

Figure 16: Experiment 16, successful until end with increased number of particles

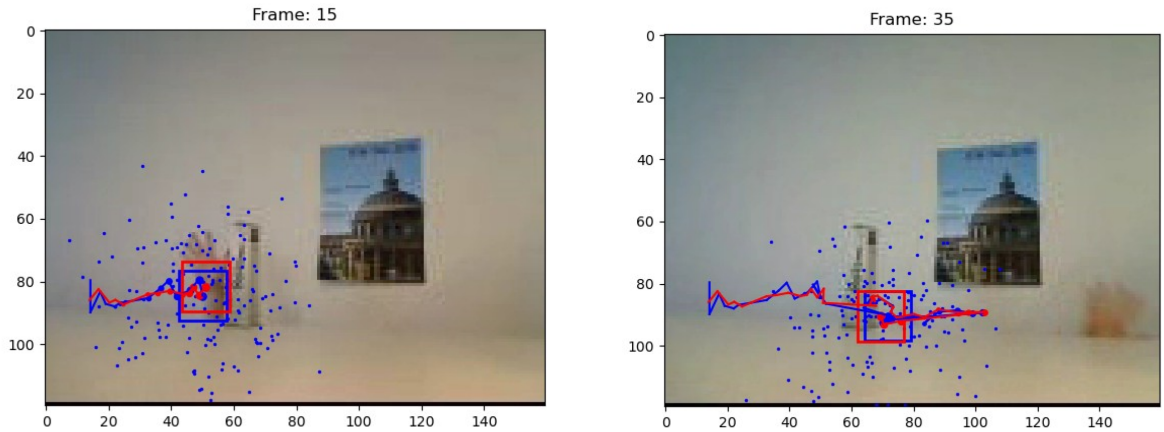

Figure 17: Experiment 17, not successful with decreased number of particles

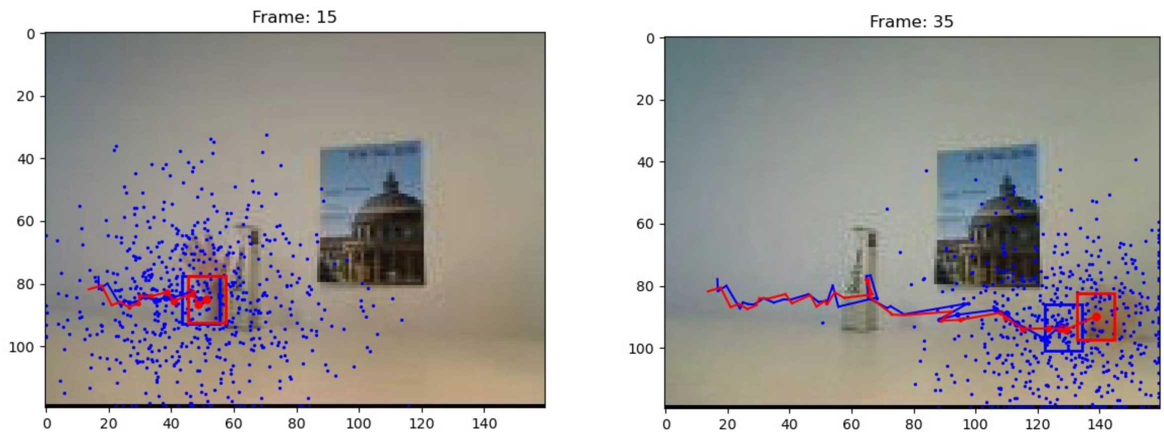

Figure 18: Experiment 18, slightly more successful with increased $\sigma_{position}$

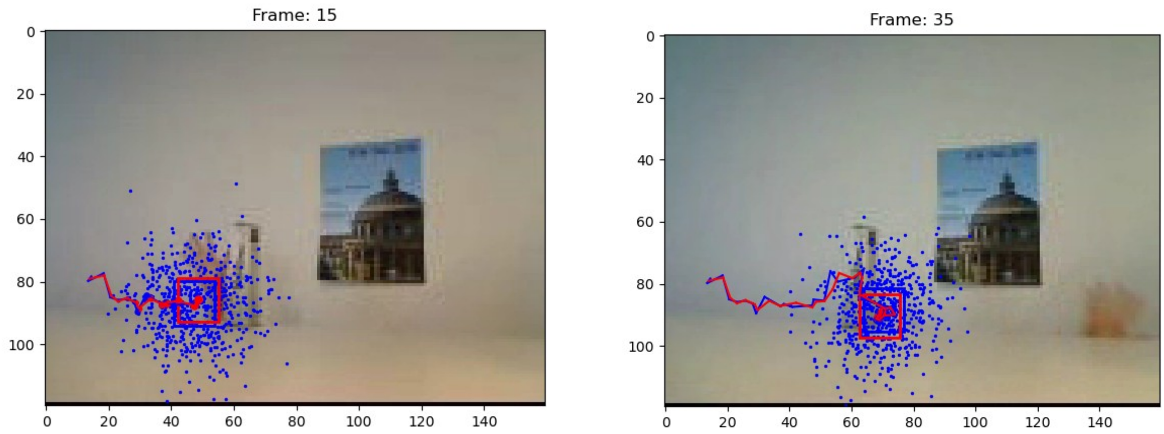

Figure 19: Experiment 19, not successful with decreased $\sigma_{position}$

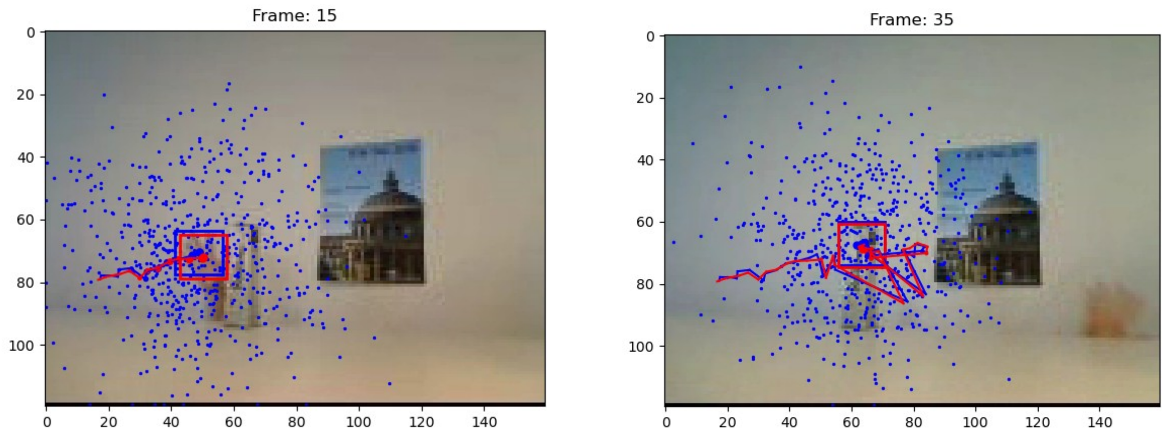

Figure 20: Experiment 20, not successful with increased $hist_{bin}$

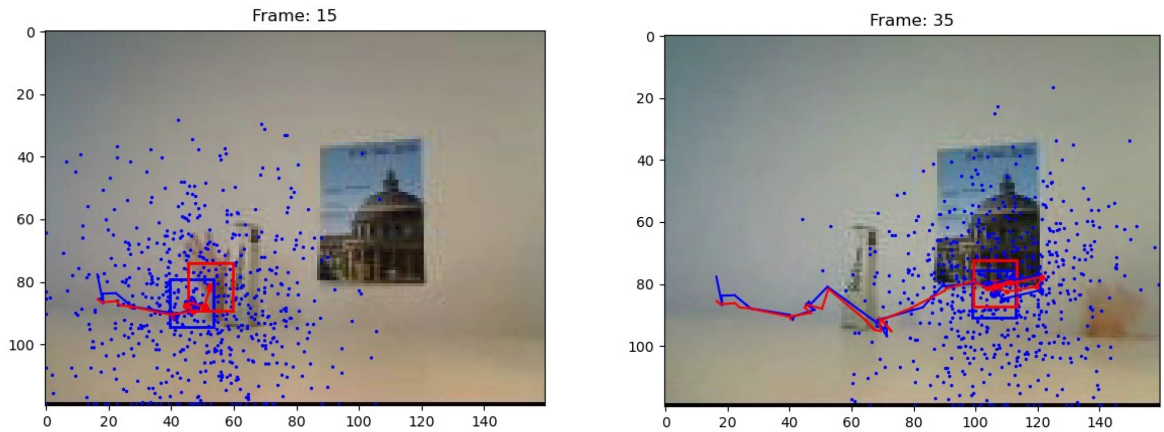

Figure 21: Experiment 21, not successful with decreased $hist_{bin}$

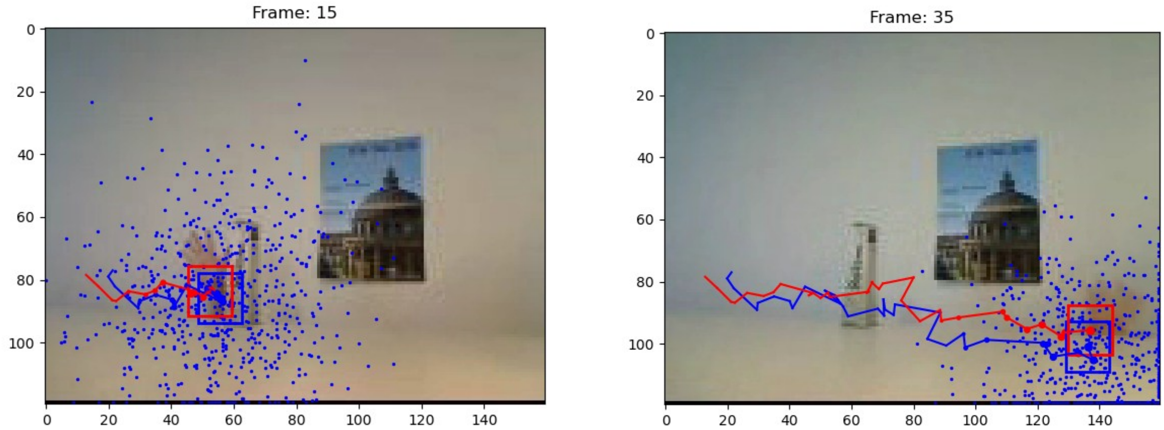

Figure 22: Experiment 22, successful but shaky using constant velocity model

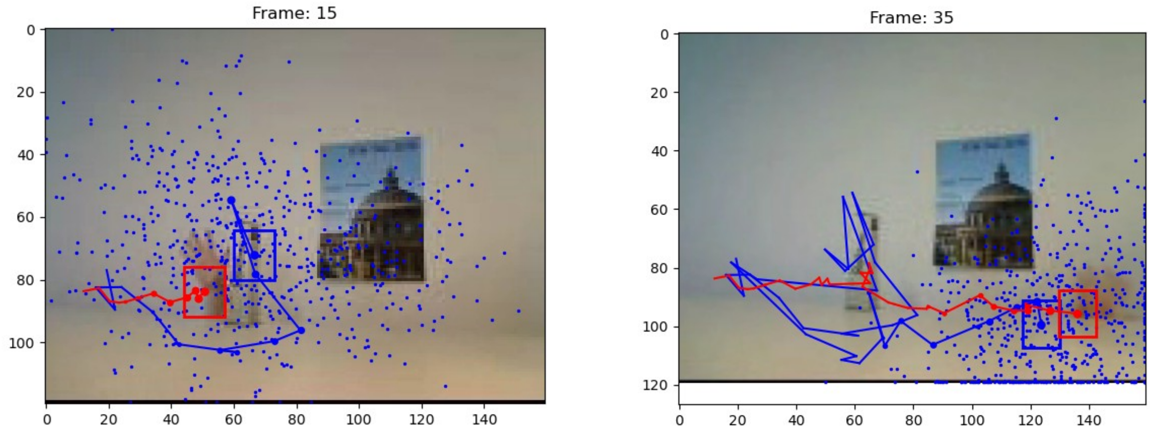

Figure 23: Experiment 23, successful but very shaky using increased $\sigma_{velocity}$

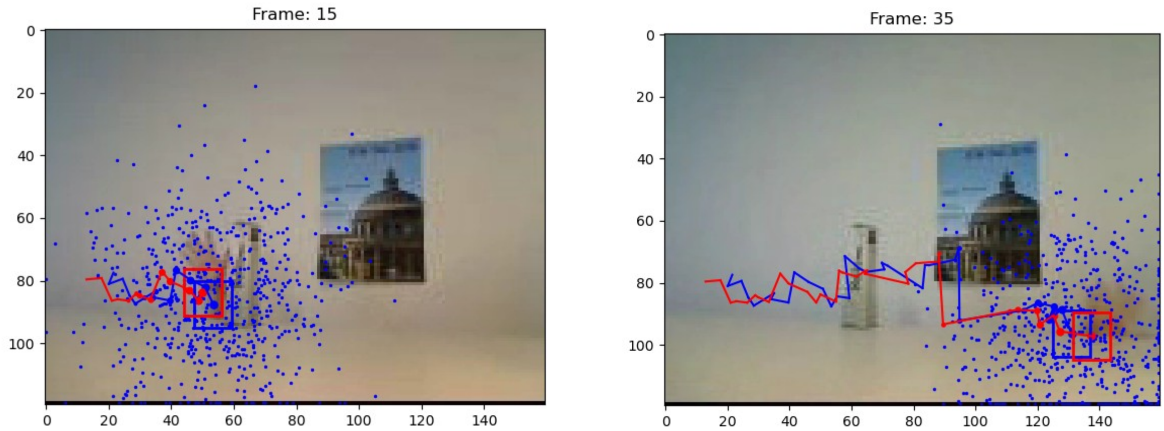

Figure 24: Experiment 24, successful but shaky using decreased $\sigma_{velocity}$

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

As can be seen in Experiments 22-24 (Figure 22-24), introducing the constant velocity model works, but doesn't seem to be bringing any improvement in my case. Especially the a priori estimates tend to get more shaky, even more so with high $\sigma_{velocity}$, and the tracking therefore seems less smooth and accurate. This is also surely due to the fact, that the hand is part of a human, and thus does not follow a perfect constant velocity motion. The model therefore has to constantly adapt and correct for the small changes in velocity. However, some further tuning of all the parameters while using the constant velocity model still could show improved results very similar or maybe even better than the ones from Experiment 18 (Figure 18).

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

We can observe the effect of adapting the position noise assumption with parameter $\sigma_{position}$ in experiments 18 & 19 (Figure 18, 19). We see that increasing it in my case caused for the whole system to act a bit smoother, as it seems to allow an estimate to take in more sparse point clouds and therefore also more easily detect the hand on the other end of the occlusion or the whole arm in front of the background. So therefore it makes the model better. Decreasing it however, makes the whole model less stable and struggling more after the occlusion etc, with opposite reasoning as before.

The effect of changing the assumption on velocity noise with parameter $\sigma_{velocity}$ in the constant velocity model can be studied in experiments 23 & 24 (Figure 23, 24). It shows that increased velocity noise makes for more shaky predictions and therefore a less smooth model. Therefore, it is advisable to keep it reasonably low, while still allowing for some change in velocity as can be expected for a human hand moving.

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

This time we have to look at experiment 15 (Figure 15), where I play with the assumption on measurement noise with parameter $\sigma_{observe}$. In my case, it has very little effect on the results when increased, and even low levels seem to be producing good results in later stages of the tuning. However, one can state that similarly as with $\sigma_{position}$, increasing it makes the observation area a bit more sparse, and therefore eventually easier to overcome some occlusions and backgrounds. However, the same effects can already be achieved with tuning just $\sigma_{position}$, too.

2.3. video3.wmv (20%)

Again, I started the experiments using the best settings from video 2 before:

Exp. #	$hist_{bin}$	α	$\sigma_{observe}$	model	#particles	$\sigma_{position}$	$\sigma_{velocity}$	initial vel.		Observation
								x	y	
25	16	0.2	0.1	0	600	20	-	-	-	Very successful
26	16	**0.8**	0.1	0	600	20	-	-	-	Also successful, but slightly more ‘shaky’
27	16	0.2	**1**	0	600	20	-	-	-	Not successful, gets distracted too much by the border
28	16	0.2	0.1	0	**1200**	20	-	-	-	Successful, but no improvement
29	16	0.2	0.1	0	**300**	20	-	-	-	Also successful, but slightly more ‘shaky’
30	16	0.2	0.1	0	600	**30**	-	-	-	Successful, but no improvement
31	16	0.2	0.1	0	600	**10**	-	-	-	Also successful, a bit smoother
32	**64**	0.2	0.1	0	600	10	-	-	-	Successful, but no improvement, slightly drifts off to the bottom
33	**8**	0.2	0.1	0	600	10	-	-	-	Also successful, a bit smoother
34	16	0.2	0.1	**1**	600	10	**1**	**5**	**0**	Very similarly successful
35	16	0.2	0.1	**1**	600	10	**10**	5	0	Quite successful, but more ‘shaky’, loses it in the end
36	16	0.2	0.1	**1**	600	10	**0.1**	5	0	Very successful and smooth too

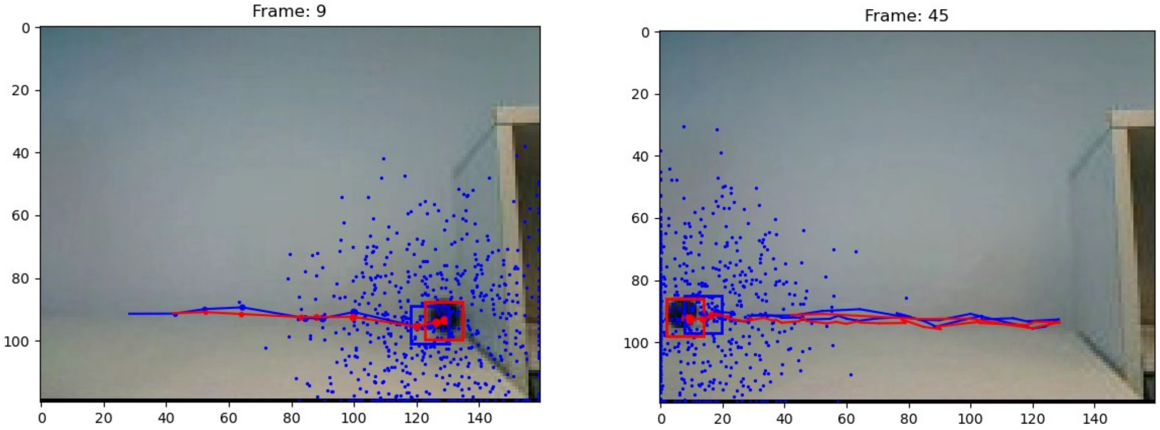

Figure 25: Experiment 25, successfully tracks the ball with same parameters as video 2

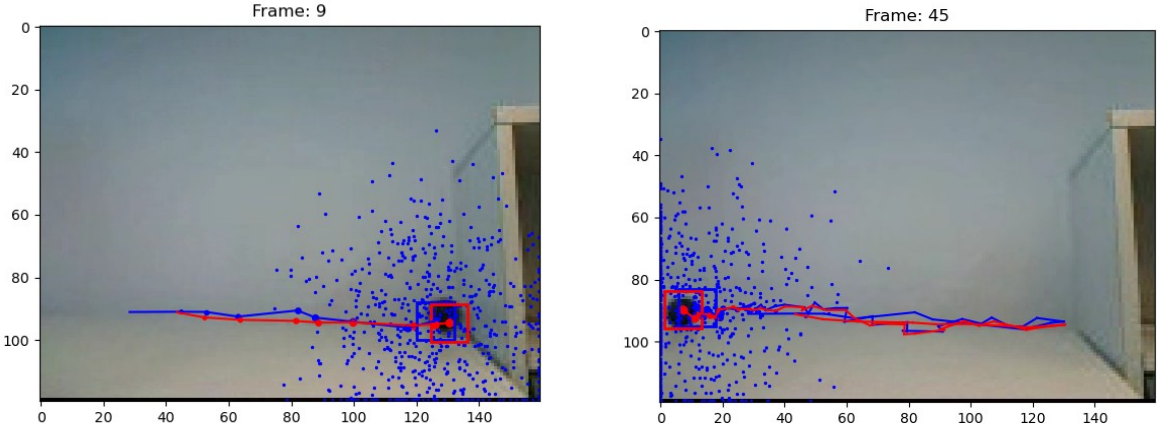

Figure 26: Experiment 26, successful but a bit shaky with increased α

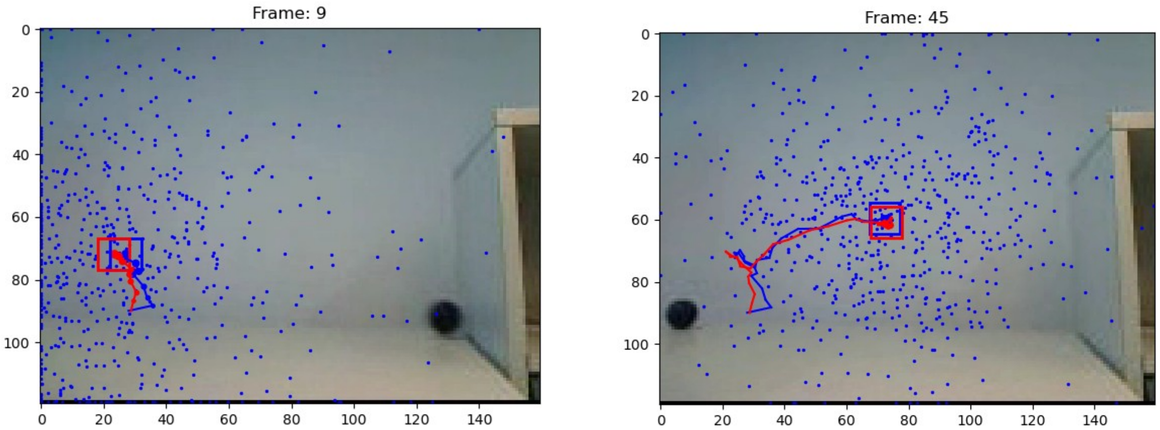

Figure 27: Experiment 27, not successful with increased $\sigma_{observe}$

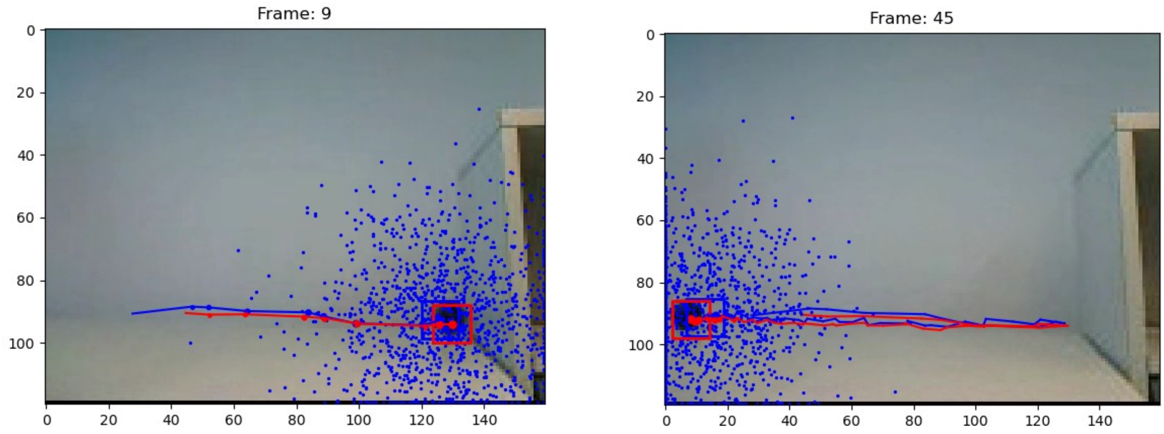

Figure 28: Experiment 28, similarly successful with increased number of particles

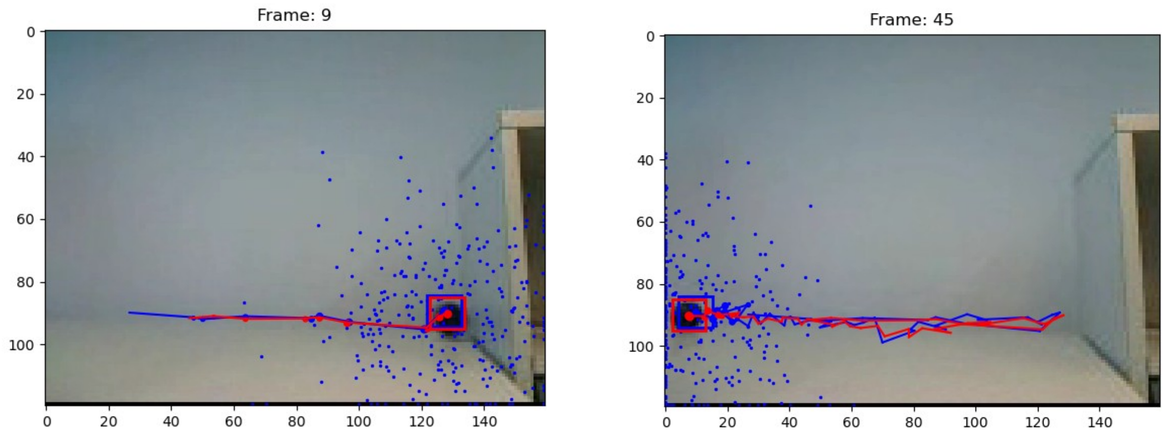

Figure 29: Experiment 29, successful but slightly more shaky with decreased number of particles

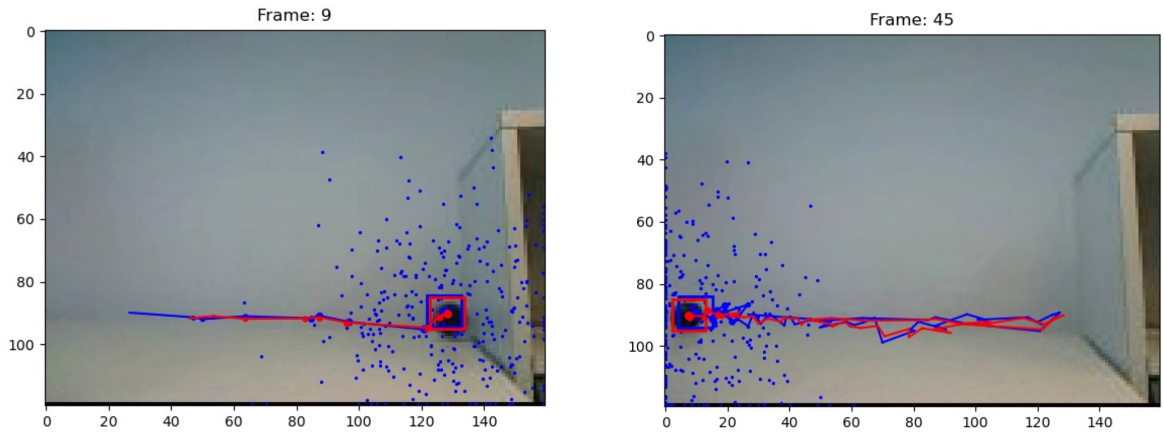

Figure 30: Experiment 30, similarly successful with increased $\sigma_{position}$

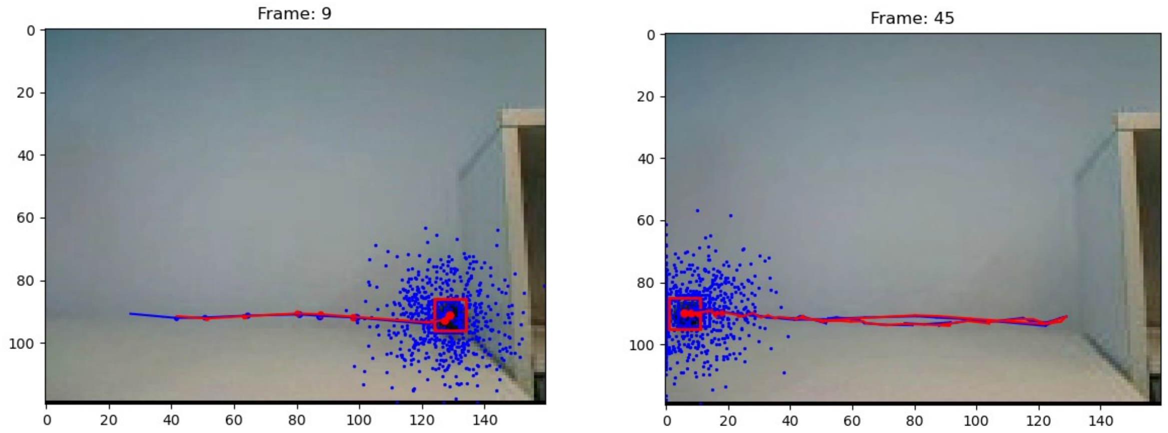

Figure 31: Experiment 31, similarly successful but smoother with decreased $\sigma_{position}$

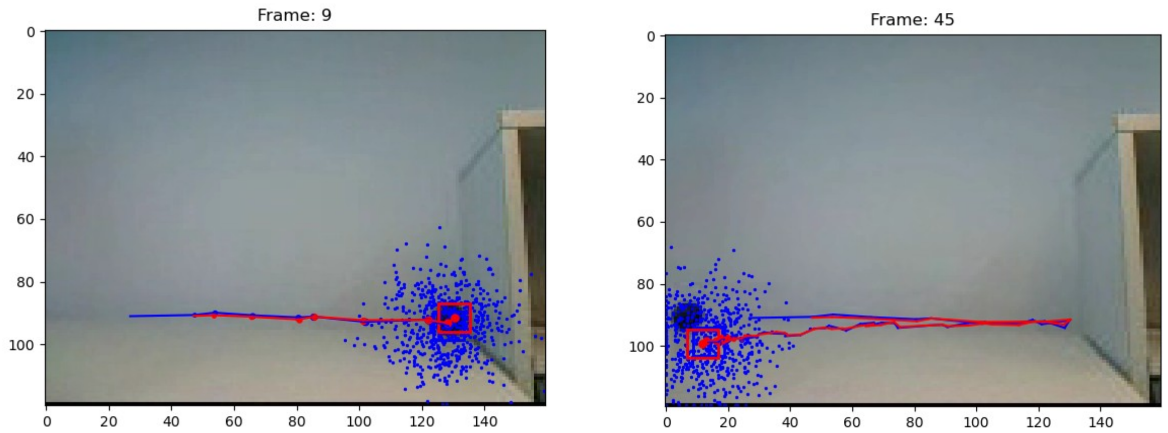

Figure 32: Experiment 32, similarly successful with increased $hist_{bin}$

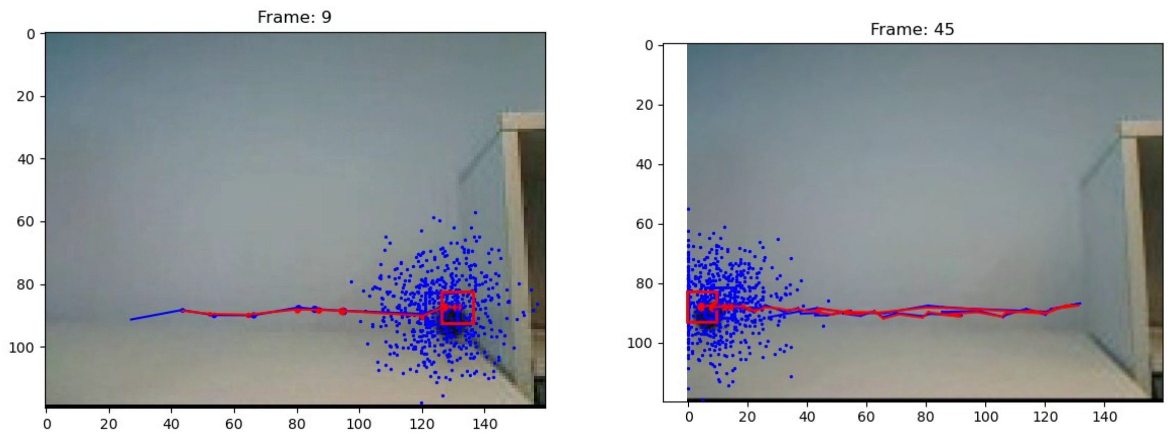

Figure 33: Experiment 33, similarly successful with decreased $hist_{bin}$

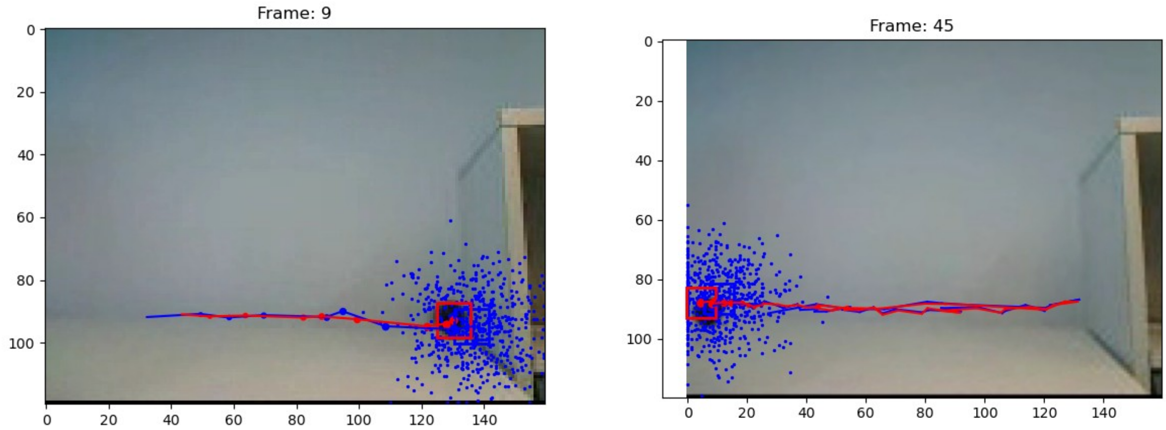

Figure 34: Experiment 34, very similarly successful with constant velocity model

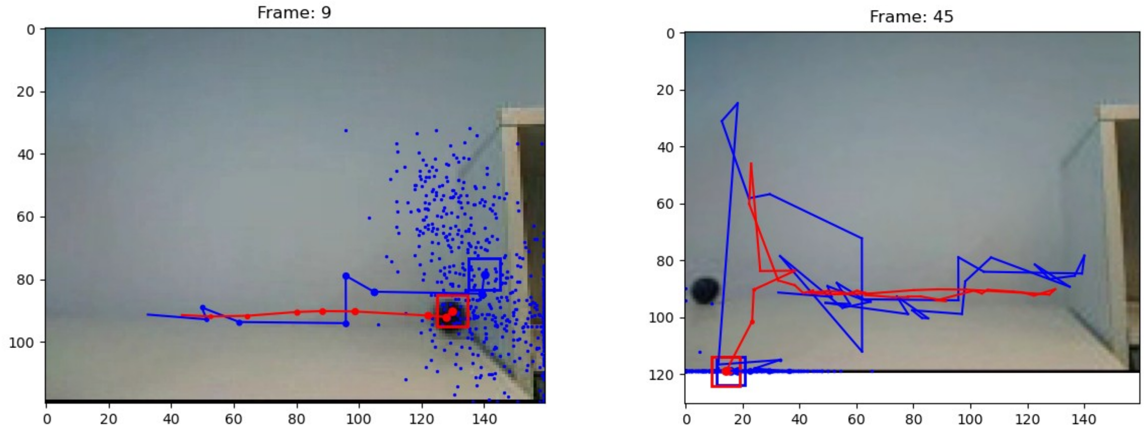

Figure 35: Experiment 35, quite successful but more shaky with increased $\sigma_{velocity}$

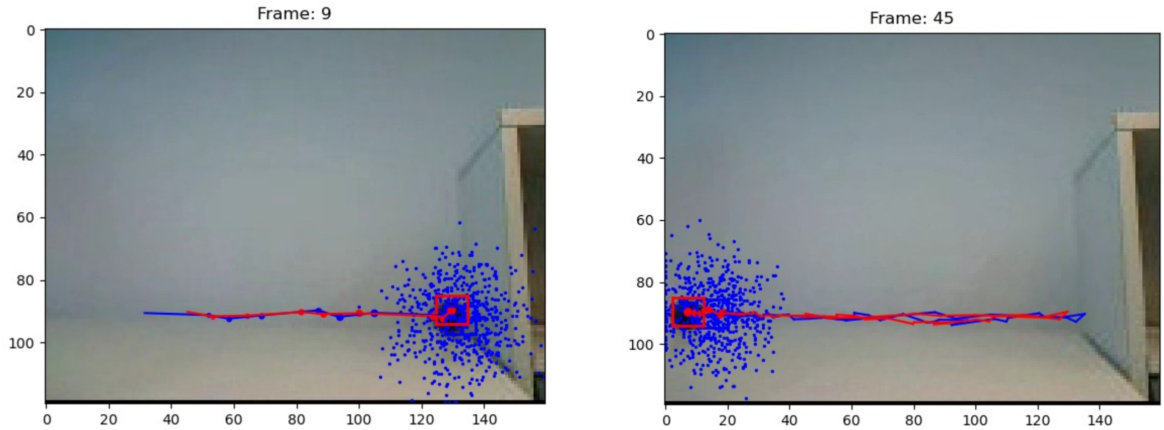

Figure 36: Experiment 36, very successful with decreased $\sigma_{velocity}$

i. *Are you able to track the ball? Why or why not?*

Yes, I indeed seem to be able to track the ball very well. This is because the ball clearly stands out from the background with its black color, and has some rather steady movement. Both the model with no velocity assumption but enough level of noise assumption and the model with constant velocity can produce smooth and expected results. This is because they both allow for enough room to find the ball in the next frame and shifting the estimate towards its new position.

ii. *Consider the questions from point b) again, for video3.wmv. What are the effects for this video?*

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

As can be seen in Experiments 34-36 (Figure 34-Figure 36), introducing the constant velocity model works, and seems to probably even produce the very best results of all experiments in number 36. In some configurations, the *a priori* estimates tend to get more shaky, even more so with high $\sigma_{velocity}$, and the tracking therefore seems less smooth and accurate. This is also surely due to the fact, that the ball physically only should be changing very little in velocity and therefore that noise part should only be very small. The model therefore has to only constantly adapt and correct for very small changes in velocity. However, some further tuning of all the parameters while using the constant velocity model still could show even more improved results very similar or maybe even better than the ones from Experiment 36 (Figure 36).

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

We can observe the effect of adapting the position noise assumption with parameter $\sigma_{position}$ in experiments 30 & 31 (Figure 30, 31). We see that decreasing it in my case caused for the whole system to act a bit smoother, as there seems to be only very little noise on the position evolution of the ball. So therefore this makes the model better. Increasing it however, makes the whole model a bit less stable and just a bit shaky.
The effect of changing the assumption on velocity noise with parameter $\sigma_{velocity}$ in the constant velocity model can be studied in experiments 35 & 36 (Figure 35, 36). It shows that increased velocity noise makes for more shaky predictions and therefore a less smooth model. This is probably, as mentioned before, because the ball has a very constant velocity and therefore barely any noise on the velocity. Therefore, it is advisable to keep its standard deviation reasonably low.

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

This time we have to look at experiment 27 (Figure 27), where I play with the assumption on measurement noise with parameter $\sigma_{observe}$. In my case, it has a significant effect on the results when increased: the more sparse observations seem to distract the model with the picture frame, which leads to a non-successful tracking of the ball. Therefore, keeping it reasonably low, just as $\sigma_{position}$, seems to be the best conclusion.

2.4. Further questions

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

This depends on the setup. For the simple hand tracking in video 1, using more or less particles has cosiderable effect: as the hand seems to need a certain density in points to be tracked well, too few particles seem to fail it (experiment 5). On the other hand, when there’s too many particles (experiment 4) the whole arm rather than just the hand gains a lot of weight, and therefore the arm is tracked rather than the hand itself.

In video 2, the effect is even more crucial: more particles and therefore a more dense modeling of the surroundings as in experiment 16 helps overcoming an occlusion such as the milk carton and distraction such as the image background. Experiment 15 and its decreased performance with less particles can be explained with the adverse effect, as here it’s even harder to bias the model over an occlusion or background.

Finally, for video 3 I did not observe any effect really when adapting the number of particles (experiments 28 & 29), this is most probably because the ball is moving out in the open and with a very constant velocity, which can be captured well with both dense and sparse modeling.

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

For all videos, changing this number of bins only had a minor effect on the histogram (experiments 8, 9, 20, 21, 32 & 33). The most notable exception is experiment 8 on video 1 with increased bins (and therefore more colors in the model), where that seems to have slightly distracted the model towards tracking the whole arm rather than just the hand. It can also be stated here that using more bins in my case caused the code to need slightly more time to compute, however still staying within reasonable amounts of time.

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

The effect is different depending on the videos. For video 1, experiment 2 has shown that allowing for more update helps improving the result. This is probably because the hand is very easy to track and changes quite a bit in appearance at over the course of the video, as its finger positions etc always slightly change out in the free air.

In video 2 however, we have a serious occlusion, so if we update the model too much, the searched for appearance would have changed way too much once the hand comes back out behind the milk carton and would not be recognized as the target anymore. So experiment 14 shows that a lower α clearly improves the results.

For video 3 then, the effect of the value of α is minor. In my particular case, increasing α shows a bit worse results rather than to just always mostly stick with the previous appearance (low α). But in any case the ball as a physical object should not at all be changing in appearance over the whole course of such a movement, so this is a negligible parameter in this case.

2.5. Conclusions

Even though this code is a rather basic approach, with some parameter fine tuning the results are surprisingly good and track the objects as wanted. For tracking hands in videos 1 & 2, the most basic model performs best in my setup, and even some serious occlusions and background distractions can be overcome without a problem when assuming enough noise. For the ball rolling through video 3, assuming a constant velocity model works very well and again shows some impressive results. We can therefore conclude that the CONDENSATION tracker indeed works well when implemented wisely and taking into account the circumstances of a scene.

2.6. Hand in

See together with this report:

- All my code (excluding the *ex6_data* folder)
- Some images of results in the folder *plots*

3. Sources

[1] Isard, M., Blake, A. - *'CONDENSATION - conditional density propagation for visual tracking'*, IJCV 1998.