Report of Anton Maksimov (antonma, 16-952-137), Task 11 "Condensation Tracker" on ETHZ course "Computer Vision".

Description of implementation of parts which aren't directly stated in the task description:

Derivation of matrix A

When we consider static model, x and y are the same as they were (without noise), so $A\mathbf{x}(t-1) = \mathbf{x}(t) = \mathbf{x}(t-1) \Rightarrow A = Id_2$.
With motion the same is for velocities (so correspondent submatrix is identical), but now x and y are changed due to velocities, so we should put ones to the correspondent to input of correspondent velocity to coordinate (in Matlab matrix coordinates we should change Id_4 in (1, 3) and (2, 4) positions, putting there ones).

Resampling

For resampling we used written for «Particle filtering» exercise on this course code, whic uses «rotating wheel» resampling. It is known to be good working when there is a big range of presented sampling probablitiies.

Histogram construction

We use `imcrop` and `imhist` functions to choose pixels of the estimated area and to construct corresponding histogram (initially we wrote it all from scratch, but even with optimization for Matlab – removing «for» loops it worked much slower than this implementation with Matlab–functions: ≈ 1 minute instead of ≈ 0.1 seconds).

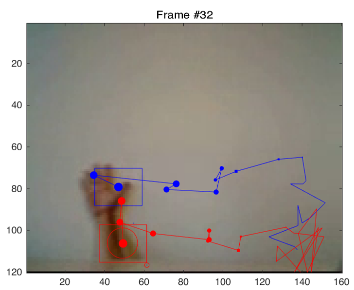

Figure 1: Static model, though works pretty well.

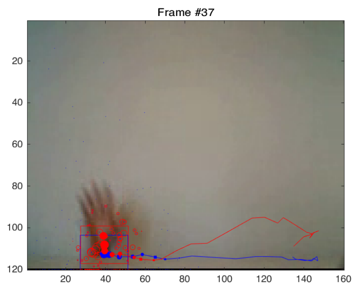

Figure 2: Dynamic model with initial velocity (1,10), as provided in `params`. Because we constrained positions it just stacks on the maximal y and folows hand, because noise and selection of particles easier changes wrong velocity 1 in x than 10 in y.

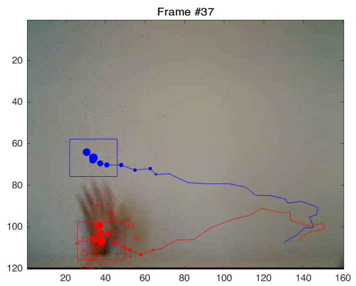

Figure 3: Dynamic model with initial velocity (-3, 0), so it corresponds to real movement of the hand and produces therefore better results (at the previous frame hand was at the last blue point position).

Figure 4: video1.wmv

Number of particles

Number of bins

Updating of appearance model

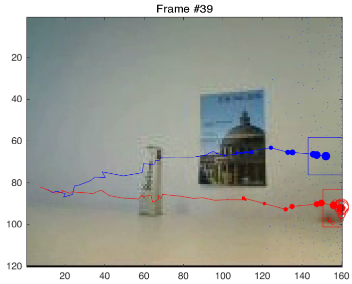

Figure 5: Static model.

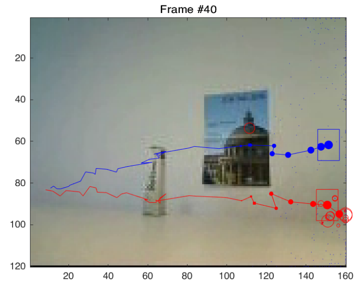

Figure 6: Dynamic model with initial velocity (0, 0).

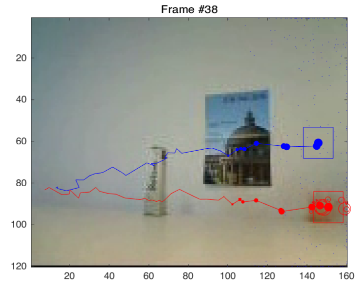

Figure 7: Dynamic model with initial velocity (5, 0).

Figure 8: video2.wmv

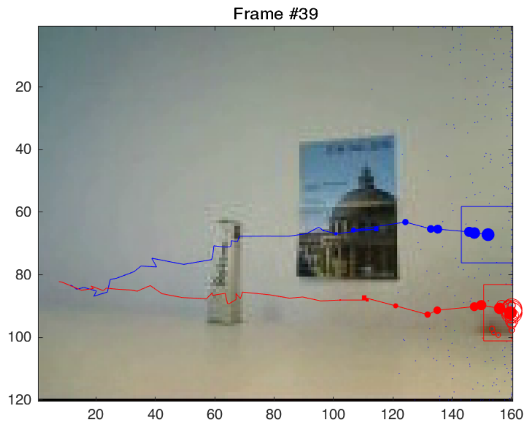

Figure 9: Static model.

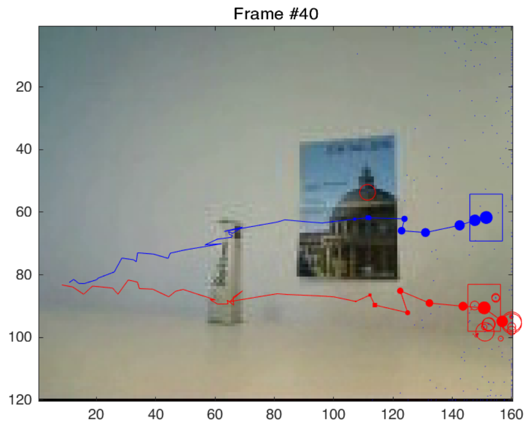

Figure 10: Dynamic model with initial velocity (0, 0).

Figure 11: video2.wmv

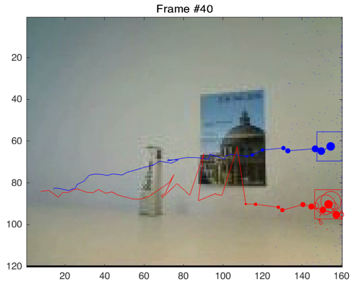

Figure 12: Dynamic model with $\sigma_{vel} = 0.1$

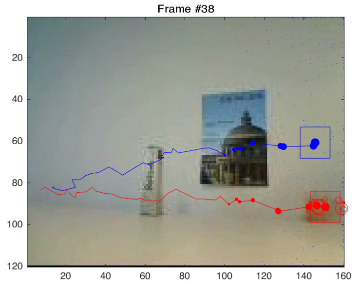

Figure 13: Dynamic model with $\sigma_{vel} = 1$

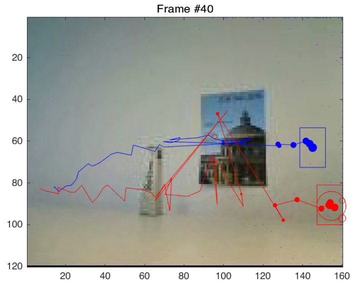

Figure 14: Dynamic model with $\sigma_{vel} = 5$

Figure 15: video2.wmv, initial velocity (5, 0). It can be seen that too big or too small noise in velocity causes unstable behaviour of trajectory.

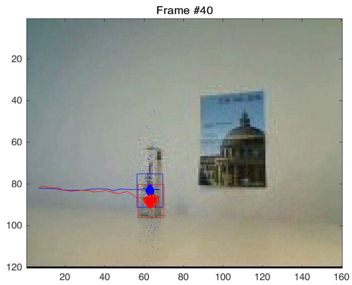

Figure 16: Dynamic model with $\sigma_{pos} = 1$

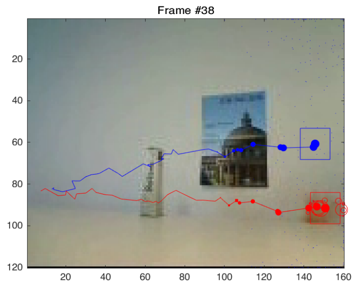

Figure 17: Dynamic model with $\sigma_{pos} = 15$

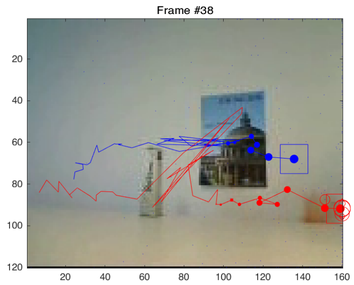

Figure 18: Dynamic model with $\sigma_{pos} = 50$

Figure 19: video2.wmv, initial velocity (5, 0). It can be seen that too big noise in position causes unstable behaviour of trajectory, to small makes it stop when occlusion appears.

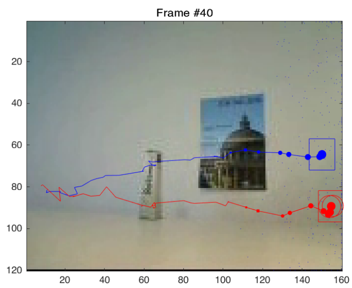

Figure 20: Static model with $\sigma_{obs} = 0.03$

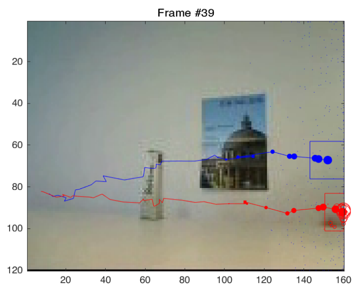

Figure 21: Static model with $\sigma_{obs} = 0.1$

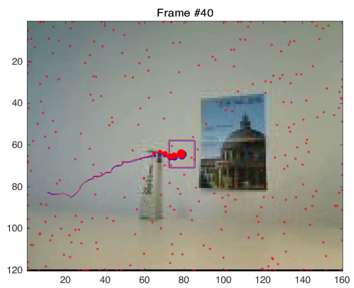

Figure 22: Static model with $\sigma_{obs} = 3$

Figure 23: video2.wmv, static. It can be seen that too big noise in position causes better following of the trajectory, but stucking when situation becomes more complicated.

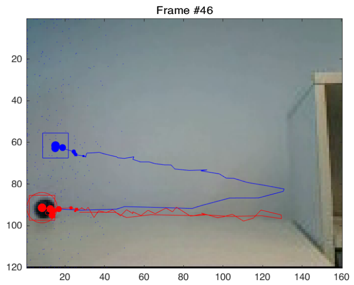

Figure 24: Linear velocity model.

```
params.initial_velocity = [5, 0];
params.sigma_velocity = 3;
params.sigma_position = 15;
params.sigma_observe = 0.05;
```

Figure 25: Used parameters

Figure 26: video3.wmv. The effect of changing parameters is similar to the previous video, with the change in the type of movement - sharp change of velocity of the ball, which is harder to account for with smaller sigma for velocity, so we increased it. Also noise in observation was made smaller in order to get more stable trajectory.