Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Vision
Autumn 2023

Object Tracking

Homework-5

December 7, 2023

Alpay Ozkan

aoezkan@student.ethz.ch

By submitting this work, I verify that it is my own. That is both implementation and report are my own work.

Contents

1 Condensation Algorithm

1.1 Method & Implementation

The method is motivated by some physics intuition. A probability density is created that moves with the object. The points are sampled around the corresponding particle states and noise accumulated. The similarity metric is in the RGB space and measured as histogram differences. Particles having higher overlap have higher weights, pushing the state density towards the target object. The overall pipeline looks like:

```
for each frame:
    # new particle states calculated from the current particle state
    particles = propagate(particles, frame_height, frame_width, params)

    # a priori state calculated with previous density observation
    mean_state_a_priori[i, :] = estimate(particles, particles_w)

    # current particle's density measured
    particles_w = observe(particles, frame, bbox_height, bbox_width, params["hist_bin"], hist, params

    # a posteriori particle state calculated with new density
    mean_state_a_posteriori[i, :] = estimate(particles, particles_w)

    # histogram of the estimated state is calculated
    hist_crrent = color_histogram(min(max(0, round(mean_state_a_posteriori[i, 0]-0.5*bbox_width)), fr
                                  min(max(0, round(mean_state_a_posteriori[i, 1]-0.5*bbox_height)), f
                                  min(max(0, round(mean_state_a_posteriori[i, 0]+0.5*bbox_width)), fr
                                  min(max(0, round(mean_state_a_posteriori[i, 1]+0.5*bbox_height)), f
                                  frame, params["hist_bin"])

    # anchoring: target histogram is updated, in case it changes over frames, to add some roboustness
    hist = (1 - params["alpha"]) * hist + params["alpha"] * hist_crrent

    # new particle state is sampled
    particles, particles_w = resample(particles, particles_w)

```

1.1.1 Color Histogram

3D histogram of the given region is calculated and then normalized to 1. The final object is of shape $hist\_bin^3$ where hist_bin is the number of histograms.

1.1.2 Propagation

The particles are updated to their new states and the transformation rule is as follows:

$s_t^{'(n)} = A.s_{t-1}^n + \epsilon$ where A is the system matrix ie. the rule that particles are expected to behave with, s_{t-1}^n is the previous state and ϵ is the noise vector with each dimension sampled from zero mean gaussian with some sigma value as one of the parameters of the model.

There are two approaches one with only position state and alternatively with position and velocity. The noise parameter for position and velocity are introduced separately. For the former case, a state is of the form

$$s = \begin{bmatrix} x \\ y \end{bmatrix}$$

without velocity

$$s = \begin{bmatrix} x \\ y \\ x^{'} \\ y^{'} \end{bmatrix}$$

with velocity

System transition matrices for both of these cases are as follows:

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

without velocity

The motion is assumed to be constant and only noise epsilon is added on top of x and y.

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

with velocity

There is motion and velocity is assumed to be per frame. The velocity is added on top of the coordinates x,y to include the shift from the motion per frame.

Explicit equations of these transformations correspond to:

$x_{t+1} = x_t + \epsilon_x$
$y_{t+1} = y_t + \epsilon_y$
with no motion

$x_{t+1} = x_t + x_t^{'} + \epsilon_x$
$y_{t+1} = y_t + y_t^{'} + \epsilon_y$
$x_{t+1}^{'} = x_t^{'} + \epsilon_{x'}$
$y_{t+1}^{'} = y_t^{'} + \epsilon_{y'}$
with motion

where $\epsilon_x, \epsilon_y \sim \mathcal{N}(0, \sigma_{pos}^2), \quad \epsilon_{x'}, \epsilon_{y'} \sim \mathcal{N}(0, \sigma_{velo}^2)$

In the implementation, the state transformation is applied with a matrix product and noise vector is added on top of that. Also I clipped/clamped the newly calculated state values for the position with respect to the valid range (0,0) and (frame_width, frame_height).

1.1.3 Observation

The point of this step is to observing the new probability density/weights for the recently calculated particles. For each particle, their corresponding histogram is calculated with the valid bounding box and chi-squared distance of this histogram with our target histogram is calculated. Then this distance is passed to a gaussian kernel and normalized to obtain a valid probability density for the particles. These probability scores indicate the level of closeness between the histogram stationed in the particle location and the target histogram. I added some shift in the exponential to have numerical stability otherwise I might receive very high problematic values due to the exponential function. For the chi-squared distance, I used the given function as suggested in the task description.

1.1.4 Estimation

The expectation of the states are calculated with respect to the state probability density to get the average state which is used as our prediction for tracking. Basically it is the dot-product between particles and particle weights.

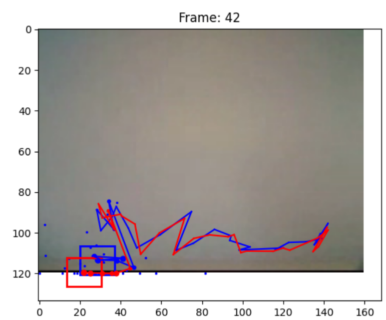
1.1.5 Re-sampling

Given the new particle density, new particles are sampled with respect to this distribution. The sampled particles' corresponding weights are re-normalized to form a valid distribution. The sampled particles and their normalized weights form our new particle and density state.

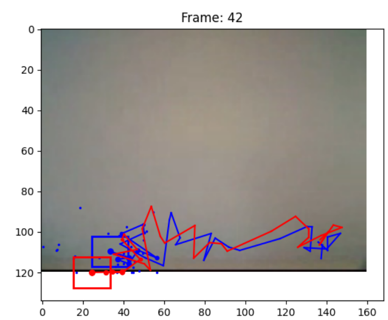
1.2 Experiments

1.2.1 Video-1

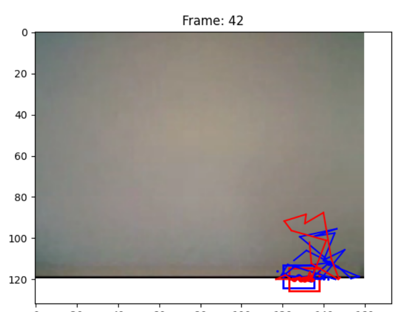
The default parameters are used. The experiment results are provided in Figure 1. Both models with and without velocity are able to reasonably well track the object. When number of samples are reduced significantly, the tracking fails.



((a)) With Position Only



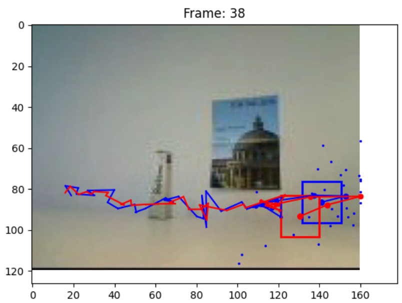
((b)) With Position and Velocity (1,10)



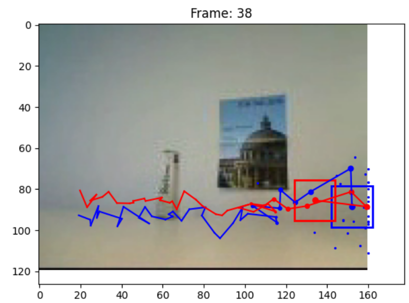
((c)) With Position and Velocity (1,10), Num Particles: 5 **(unsuccessful)**

Figure 1: Video-1 Results

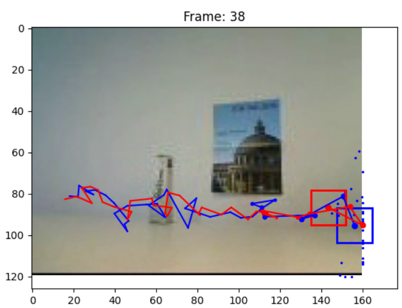
1.2.2 Video-2



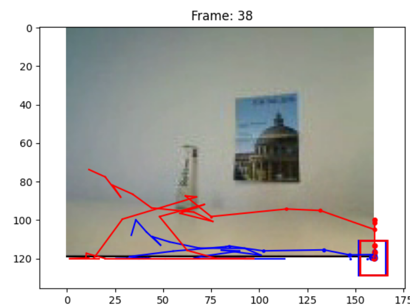
((a)) With Position Only



((b)) With Position and Velocity (1,10)



((c)) With Position and Velocity (4,0)



((d)) With Position and Velocity (20,30) **(unsuccessful)**

Figure 2: Video-2 Results

Even just using the default parameters does decent job and tracks the hand as in Figure 2. Introducing too much initial velocity fails the tracking procedure as in the case of (20,30). For the following questions and corresponding experiments the default parameters are used with initial veloctiy (4,0) and only the indicated parameter is changed. The defaults:

Initial Velocity: (4,0), Sigma Position: 15, Sigma Velocity: 1, Num Particles: 30, Sigma Observe: 0.1, Hist Bins: 16, Alpha: 0

In this section, the questions are answered more intuitively and more detailed answers are provided in section 1.2.3.

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

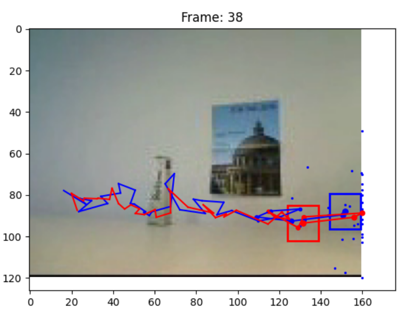
Constant velocity motion helps the tracker not to fall behind, state position is shifted with the velocity towards the initial velocity direction. Sampled particles are in the direction of the motion and tracker is given a useful bias in this sense.

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

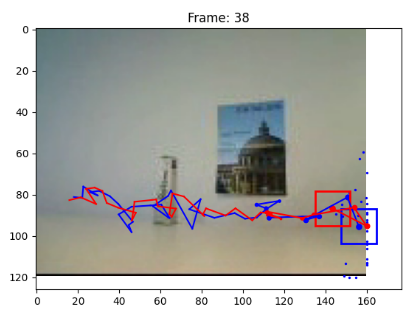
The effect of different system noises are illustrated in Figure 4. Sigma position and velocity have similar effects, the difference is that velocity noise is accumulated for the later time step but position noise takes the imminent effect for the new position calculation. Large sigma creates sparse and further samples from the center position and causes the model to deviate from the actual object. Very small noise also results a very concentrated area for sampled points and when the object moves fast it is very likely to loose the object.

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

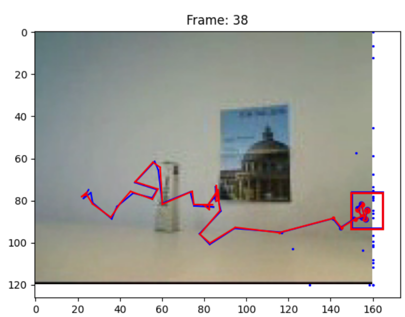
The effect of measurement noise is to adjust the significance of the chi-squared histogram difference in the calculation of the probability weights for the particles. Particles close to the target are assigned higher probabilities, but as the measurement noise (sigma) increases the the effect of the difference decreases and in the limit goes to zero, resulting equal probabilities for each particles regardless of the histogram difference. Also, as the sigma decreases the effect of the difference sharpens ie. particles close to the target get significantly higher weights. This effect is illustrated in Figure 3 where increased sigma results a loose tracking but smaller sigma values results tighter tracking with less fluctuations.



((a)) Sigma-Observation: 0.05

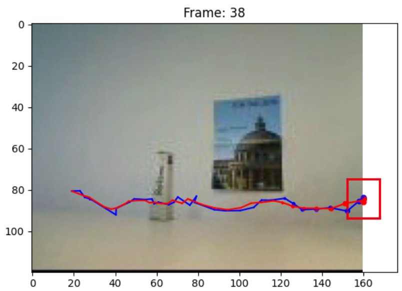


((b)) Sigma-Observation: 0.1

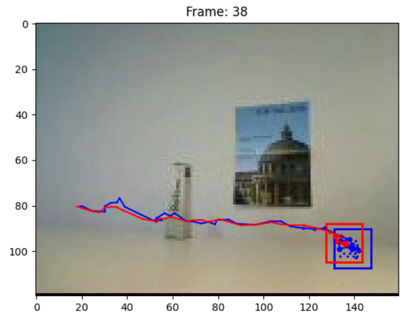


((c)) Sigma-Observation: 1

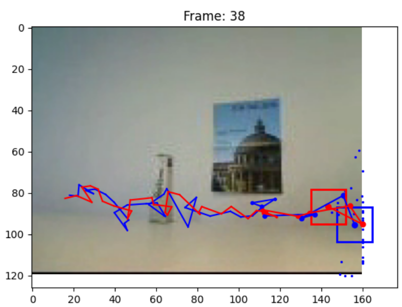
Figure 3: Video-2 Different Measurement Noises



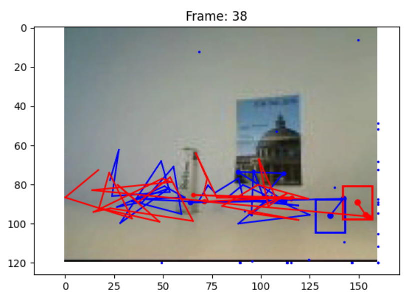
((a)) Sigma-Position: 0.1



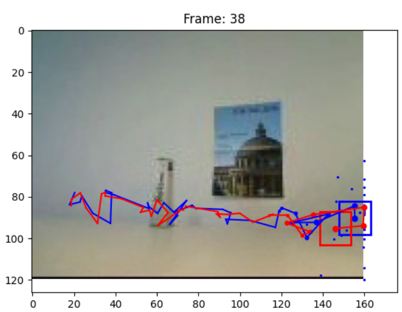
((b)) Sigma-Position: 1



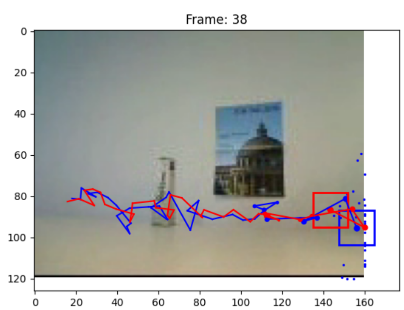
((c)) Sigma-Position: 15



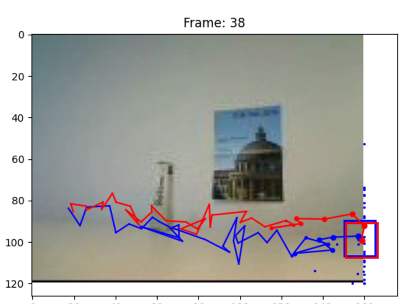
((d)) Sigma-Position: 50



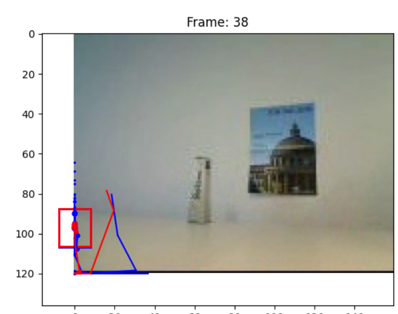
((e)) Sigma-Velocity: 0.1



((f)) Sigma-Velocity: 1



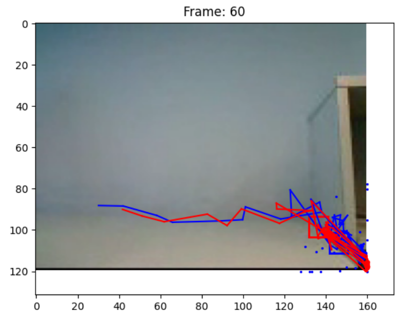
((g)) Sigma-Velocity: 3



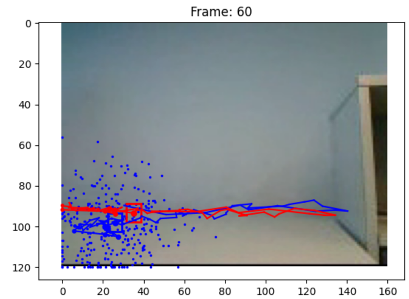
((h)) Sigma-Velocity: 10

Figure 4: Video-2 Different Sigma Positions and Velocities

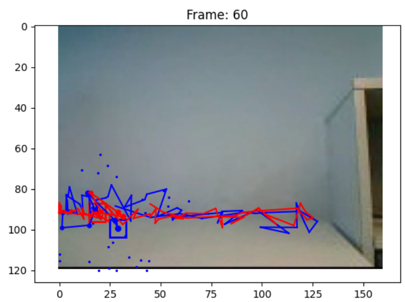
1.2.3 Video-3



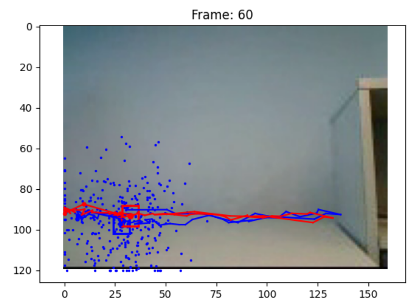
((a)) **Best Params of Video-2 (unsuccessful)**



((b)) Sigma Obsv: 0.05, Num Particles: 300



((c)) Sigma Obsv: 0.1, Num Particles: 30



((d)) Sigma Obsv: 0.1, Num Particles: 300

Figure 5: Video-3 Results

The best parameter configuration chosen from section 1.2.2 is:

Initial Velocity: (4,0), Sigma Position: 15, Sigma Velocity: 1, Num Particles: 30, Sigma Observe: 0.05, Hist Bins: 16, Alpha: 0

The best parameters of video-2 are illustrated in Figure 5 a and some other configurations with tweaked sigma observations and number of particles are given. The best configuration does not work for video-3. The reason is probably because the sigma observation is too small and when the object collides with the wall the dark edge on the wall confuses the model and it gets stuck on the right corner, but when the sigma is a bit larger it can attend to a larger area and increases the roboustness in this case. Another solution would be to increase the number of samples, this helps the model to observe larger neighborhood and increases the chance of overlap with the object. Another reason why it did not work could be because of the initial noise (4,0) which might be too much for this case and might have given significant bias towards right where it got stuck.

For the following questions, default parameters as assumed in section 1.2.2 are used.

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

To bias the shift in the tracker towards the motion direction. This is useful for objects moving fast since in the next frame the object will be further away and motionless particles will not be in the range. However, velocity extension provides robustness such that particle range is extended towards the the direction of movement.

Tracking steps between each frame gets larger as velocity is increased as seen from Figure 7. Too much velocity gives too much bias and model is stuck on the right as seen for velocities in +x higher than +2. Also, giving velocity in y direction does not make much sense since the object only moves in the x direction in this case and creates more deviations horizontally ie. y-direction.

Also, the tracking of the forward motion of the ball is more accurate than the backward motion after the collision since motion direction is reverted but the initial velocity is applied in the same way as it started. Hence we have sharper fluctuations on the way back, but with the right velocity it is still able to track and also the speed of the ball decreases after the collision hence easier to track.

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

The system noise has two components: position and velocity. Their combined effect is additive noise since velocity is added on top of the position to determine the new position and both of the previous velocity and position includes the noise defined with their corresponding sigma values. Having sigma value for velocity smaller than the position makes more sense since we do not want the velocity noise to dominate over the position noise since its point is to add some stochasticity for the velocity component.

The effect of different sigma positions are illustrated in Figure 6. As sigma position increases the range that particles are transformed are increased significantly. For the extreme case with sigma=50 the steps are too large it overshoots. For small sigma values 1 and 5, I observed that the range was too small such that the transformed points were not overlapped with the moving object and lost the track resulting random movements as shown in Figure 6 a and b.

The effect of different sigma velocities are also illustrated in Figure 6. Similarly, as sigma velocity increases the range of sampling is increased significantly resulting longer zig-zags and when it is too much then it gets stuck around edges and corners.

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

The measurement noise is used as the sigma parameter for the weight calculation in the Gaussian form, $\pi^n = \frac{1}{\sqrt{2\pi\sigma}} . \exp \frac{-X^2(CH_{S^n}, CH_{target})}{2\sigma^2}$. When we have less sigma then the effect of chi-squared histogram distance gets more significant and we give higher weights to the particles that have closer overlap with the target region in a way it concentrates more. On the other hand, having more sigma value makes the weights close to uniform case ie. the xi-distance gets less important. These theoretical analysis is also supported by our experiments; as seen in Figure 8, sigma=0.05 is more concentrated and we have less fluctuations, but when we have sigma=1, it is not able to track the object, it looks around at irrelevant locations.

What is the effect of using more or fewer particles?

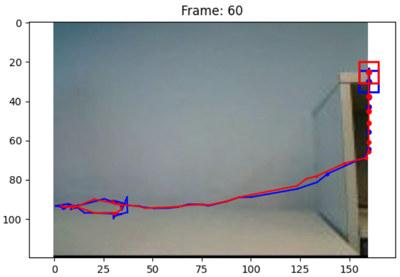
More particles gives more robustness as the expectation calculation will have less variance and particles with the right overlap on the object will have higher weights which increases the likelihood of tracking the object in expense of having some computational overhead. This effect is visible in Figure 5. Less particles means the opposite, the tracking will me more noisy and it is more likely to deviate from the object as some background might have some histogram values close to the target.

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

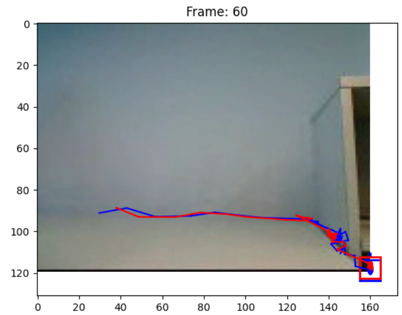
The effect of using different histogram bins are visualized in Figure 9. The increase in the number of histograms results in noisy and unsuccessful tracking since we will have much sparser histogram features and the difference between histograms will get significantly less as they are normalized. Hence, the model will not be able to attend the right part of the model. Furthermore, in this experiment the low histogram values works, but it does not have to. In this example, the ball is black and background is white, so the contrast difference is obvious, but if the object and the background was more colorful then we would need more bins to have the right histogram feature space to measure the distance. With less number of histograms we have more pooling or smoothing effect across the calculated pixel space.

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

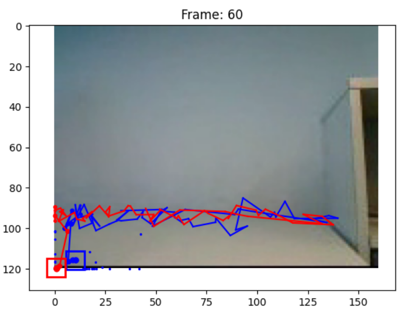
This method is useful for objects that might change during the tracking which could be due to orientation, lightening, occlusion etc. The following update equation is used to change the target histogram throughout the procedure for more robust tracking: $CH_{target} = (1 - \alpha).CH_{target} + \alpha.CH_{E[s_t]}$ Basically, the new histogram target is linearly interpolated between the previous target and the currently tracked bounding box's histogram. The advantage of this method is that it is iteratively adaptive to the object, but the disadvantage is that we assume the object being tracked correctly if there is high noise during tracking then we will feed to much noise in our target calculation in which case we might be lost in tracking especially on long frame lengths. Our experiments have low frame lengths, but if this method is used for long frame lengths the target object might deviate a lot from the intended object histogram features and we would have too much unstabilities.



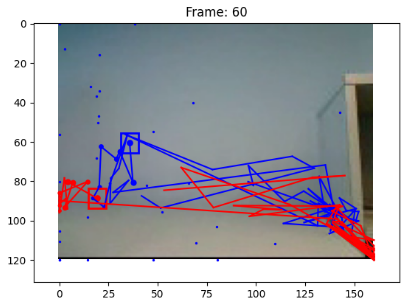
((a)) Sigma-Position: 1



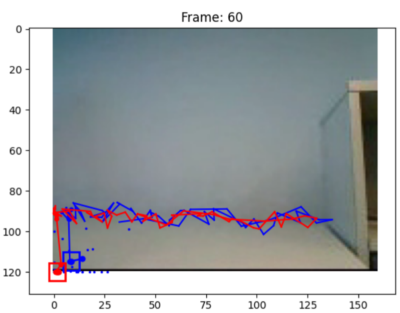
((b)) Sigma-Position: 5



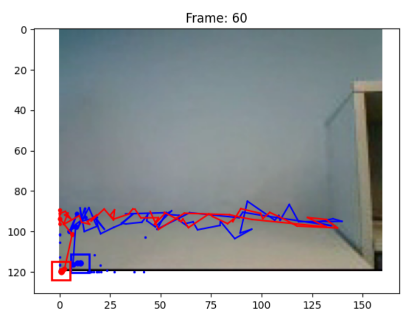
((c)) Sigma-Position: 15



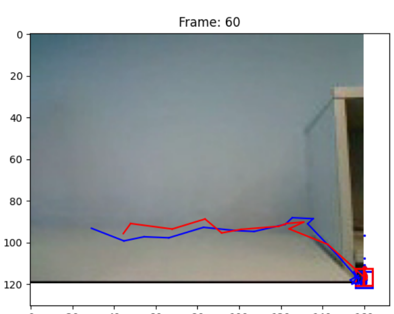
((d)) Sigma-Position: 50



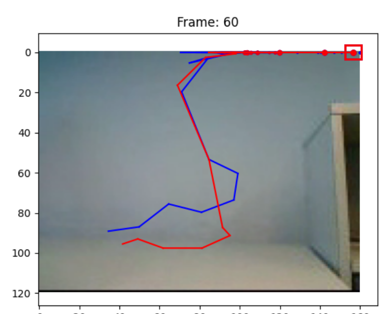
((e)) Sigma-Velocity: 0.1



((f)) Sigma-Velocity: 1

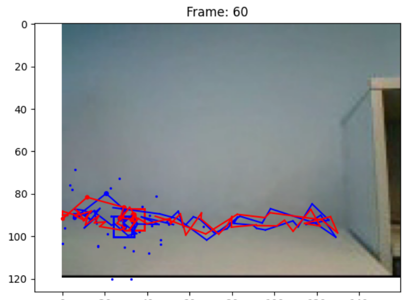


((g)) Sigma-Velocity: 3

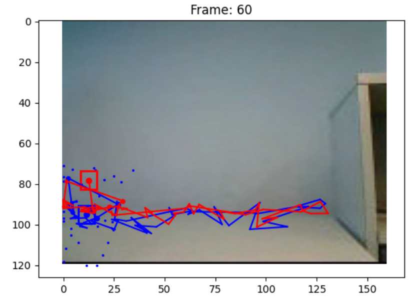


((h)) Sigma-Velocity: 10

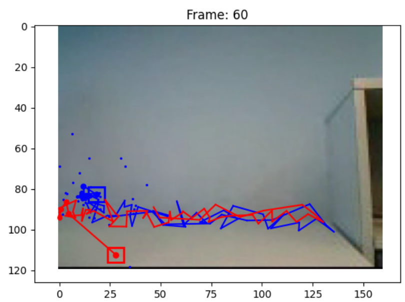
Figure 6: Video-3 Different Sigma Positions and Velocities



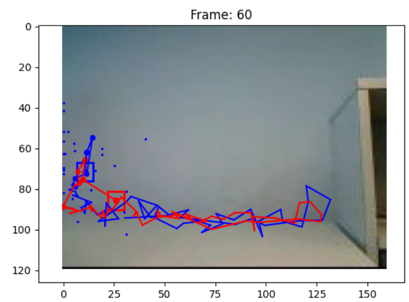
((a)) No Velocity



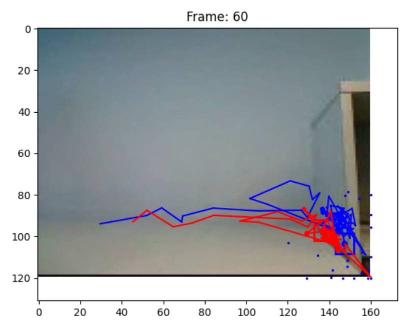
((b)) Initial Velocity: (1,0)



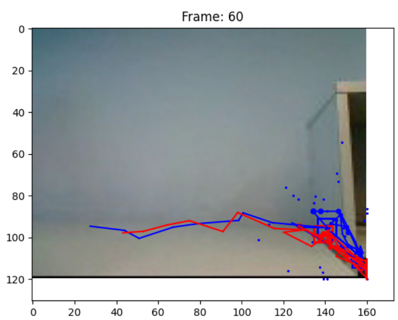
((c)) Initial Velocity: (1,1)



((d)) Initial Velocity: (2,0)

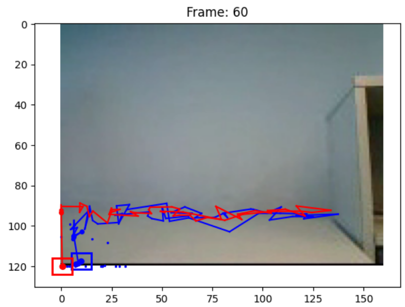


((e)) Initial Velocity: (2,2)

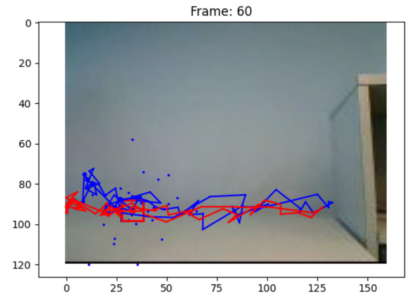


((f)) Initial Velocity: (4,0)

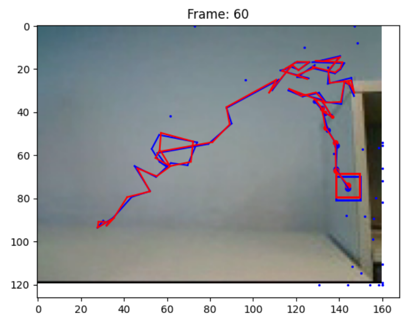
Figure 7: Video-3 Different Initial Velocities



((a)) Sigma-Observation: 0.05

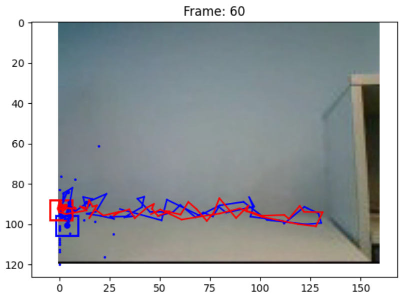


((b)) Sigma-Observation: 0.1

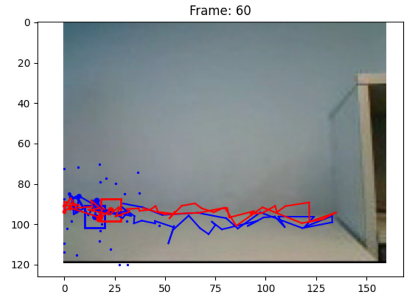


((c)) Sigma-Observation: 1

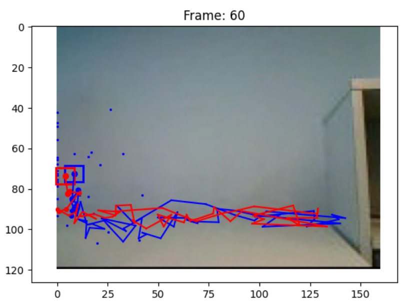
Figure 8: Video-3 Different Measurement Noises



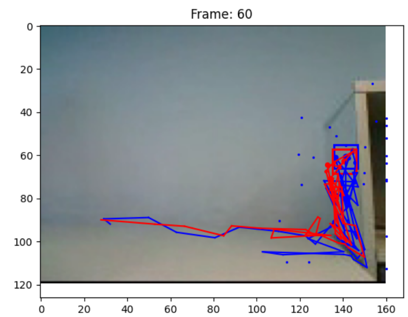
((a)) Histogram Bins: 4



((b)) Histogram Bins: 16



((c)) Histogram Bins: 32



((d)) Histogram Bins: 64

Figure 9: Video-3 Different Histogram Bins