Lab Assignment - Condensation Tracker

Kaiyue Shen

December 10, 2020

1 Pipeline of Condensation Tracker

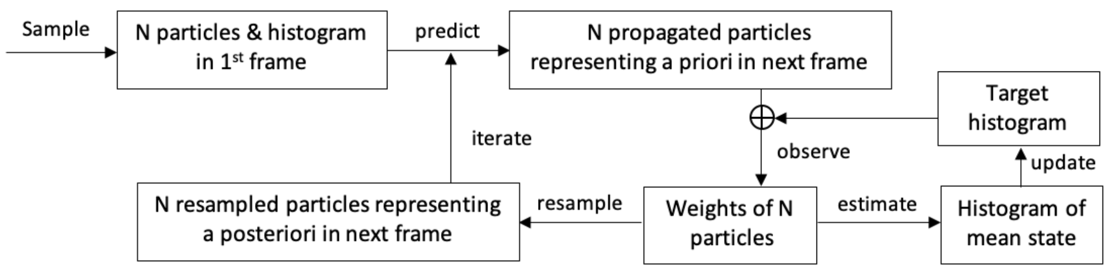


Figure 1: Pipeline of Shape Matching

Based on my understanding, the Condensation Tracker is consisted of a sequence of operations:

1. **Sample**. We use the center of manually selected bounding box and initial velocity(if we use constant velocity model) to initialize the particles. We also compute the normalized histogram of bounding box in the 1st frame and use it to initialize the the target histogram, which is needed in the "observe" step.

2. **Predict**. We build two prediction models $s_t^{(n)} = As_{t-1}^{(n)} + w_{t-1}^{(n)}$: (i) no motion at all, just noise; (ii) constant velocity motion model. For (i), the state of particles are in the form of (x, y), the dynamic model is shown in formula 1. Because the velocity in both directions are zero, $x_t^{(n)} = x_{t-1}^{(n)}$+noise, $y_t^{(n)} = y_{t-1}^{(n)}$+noise. For (ii), the state of particles are $(x, y, \dot{x}, \dot{y})$, the dynamic model is shown in formula 2. Since the velocity is constant, $\dot{x}_t^{(n)} = \dot{x}_{t-1}^{(n)}$+noise, $\dot{y}_t^{(n)} = \dot{y}_{t-1}^{(n)}$+noise, displacement equals velocity times time interval, $x_t^{(n)} - x_{t-1}^{(n)} = \dot{x}_{t-1}^{(n)}\Delta t$, similarly, $y_t^{(n)} - y_{t-1}^{(n)} = \dot{y}_{t-1}^{(n)}\Delta t$. In our case, Δt equals 1.

3. **Observe**. For each sample, We compute the Chi-squared distance between current histogram and target histogram. And we use the distance to further get the weight of each sample. From formula 3, we can easily see that better tracking, smaller the distance, higher the weight.

4. **Estimate + Update**. We simply use the weighted average to compute the mean state. Then we use the histogram of mean state and old target histogram to update the target histogram. The update rule is shown in formula 4, where α is the parameter we can modify, which we will discuss in the experiment part.

5. **Resample**. Given N propagated particles and corresponding weights, we can use the resampling wheel as in the lab assignment 3 to get N resampled particles.

6. **Iterate**. We iterate from "Predict" to "Resample" until the last frame.

$$\begin{pmatrix} x_t^{(n)} \\ y_t^{(n)} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} x_{t-1}^{(n)} \\ y_{t-1}^{(n)} \end{pmatrix} + w_{t-1}^{(n)} \tag{1}$$

$$
\begin{pmatrix} x_t^{(n)} \\ y_t^{(n)} \\ \dot{x}_t^{(n)} \\ \dot{y}_t^{(n)} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} x_{t-1}^{(n)} \\ y_{t-1}^{(n)} \\ \dot{x}_{t-1}^{(n)} \\ \dot{y}_{t-1}^{(n)} \end{pmatrix} + w_{t-1}^{(n)} = \underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} x_{t-1}^{(n)} \\ y_{t-1}^{(n)} \\ \dot{x}_{t-1}^{(n)} \\ \dot{y}_{t-1}^{(n)} \end{pmatrix} + w_{t-1}^{(n)} \tag{2}
$$

$$
\pi^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{\chi^2(CH_s(n), CH_{target})^2}{2\sigma^2}} \tag{3}
$$

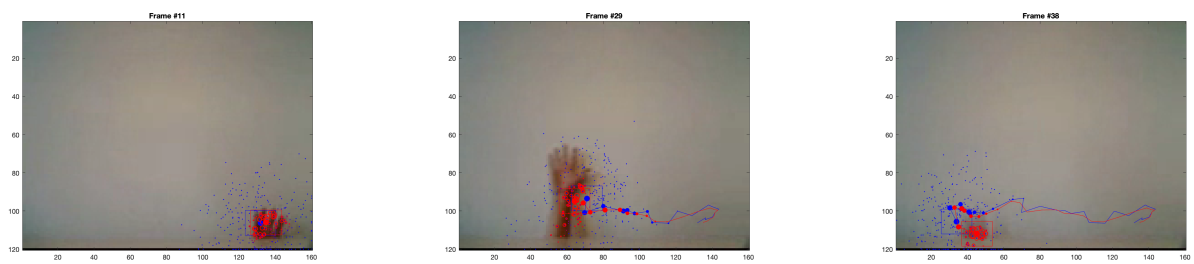
$$
CH_{\text{target}} = (1-\alpha) \cdot CH_{\text{target}} + \alpha \cdot CH_{E[s_t]} \tag{4}
$$

I implement the whole pipeline as shown in figure 1 and experiment results are in the following section.

2 Experiment results

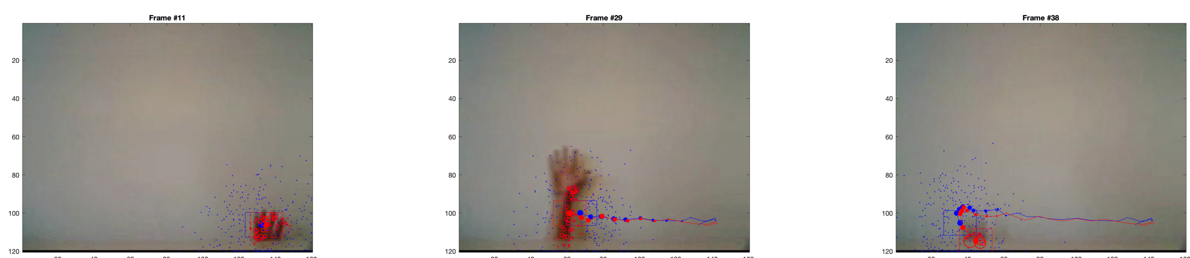
2.1 Video 1: Hand, uniform background

For video 1, I try two prediction models and vary α in formula 4. We set other parameters as default. The tracking results are shown in figure 2, 3, 4, 5. The blue line refers to a priori mean state trajectory and the red line refers to a posteriori mean state trajectory (same for other figures).



(a) Frame #11 (b) Frame #29 (c) Frame #38

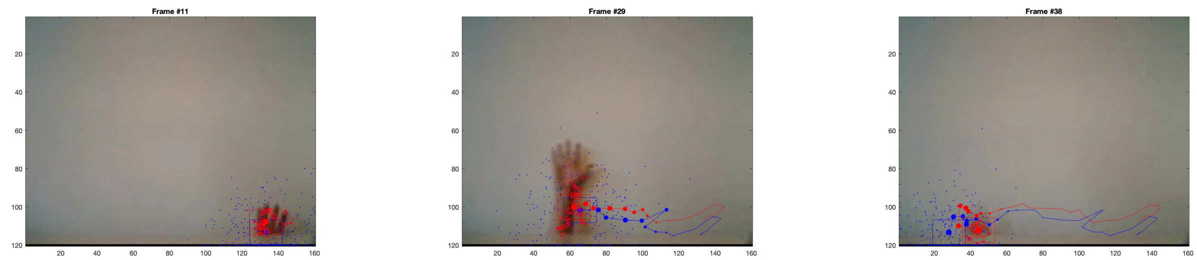
Figure 2: Tracking results of video 1, prediction model (i): no motion, $\alpha = 0$



(a) Frame #11 (b) Frame #29 (c) Frame #38

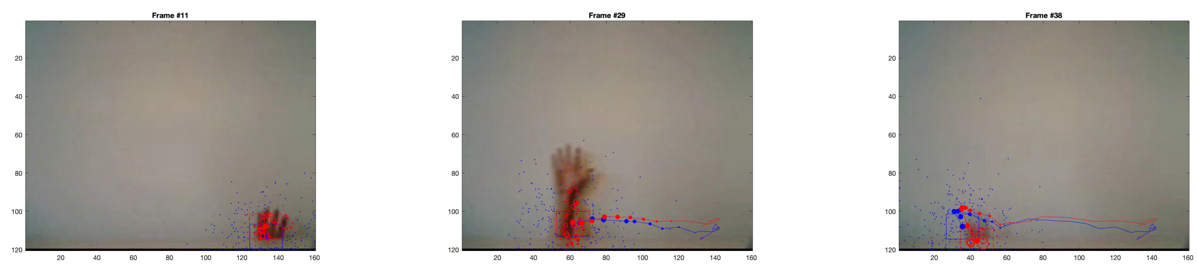
Figure 3: Tracking results of video 1, prediction model (i): no motion, $\alpha = 0.5$

Comparing four figures, we find the results look very similar and all capture the target object quite well. The only problem is that as the forearm has the same color as the hand, the bounding box will gradually move to the forearm.



(a) Frame #11 (b) Frame #29 (c) Frame #38

Figure 4: Tracking results of video 1, prediction model (ii): constant velocity, $\alpha = 0$

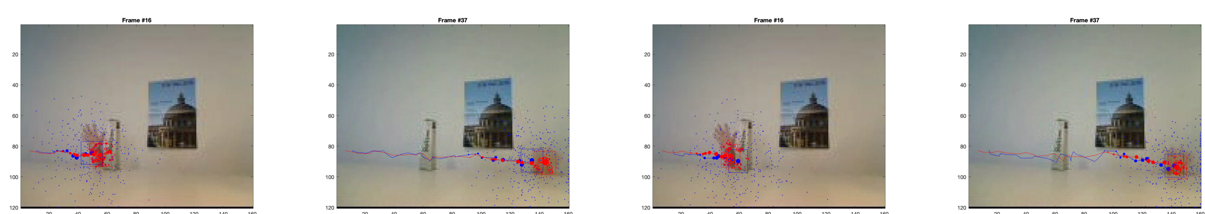


(a) Frame #11 (b) Frame #29 (c) Frame #38

Figure 5: Tracking results of video 1, prediction model (ii): constant velocity, $\alpha = 0.5$

2.2 Video 2: Hand, clutter, occlusions

For video 2, I try two prediction models, and vary the system noise and measurement noise to see their effect. Besides, we set all $\alpha = 0.5$ and the initial velocity for constant velocity model to be $[5, 0]$. Because when the system noise is small, the precision of dynamic model will have huge influence on the tracking result. By observing the video 2, we find the hand movement is mostly in the x direction. And by experiment, we find 5 is a proper velocity along the x axis. The tracking results are shown in figure 6, 7, 8.

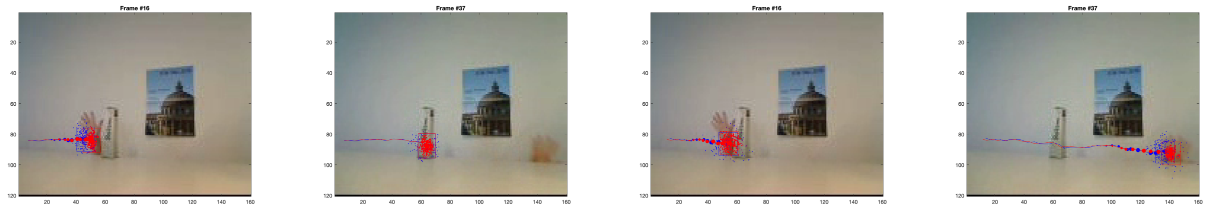


(a) Frame #16 (b) Frame #37 (c) Frame #16 (d) Frame #37

Figure 6: Tracking results of video 2, $\alpha = 0.5$, $\sigma_{position} = 15$, (a)(b): no motion model, (c)(d): constant velocity model

- What is the effect of using a **constant velocity motion model**?
  By comparing figure 6a, 6b and figure 6c, 6d, we find that when the system noise is large enough, both the no motion model and constant velocity model can track the hand in the case of occlusion. I think the reason is that large system noise let the particles fully spread out and some particles can capture the reappeared hand.

- What is the effect of assuming decreased/increased **system noise**?
  By comparing the figure 6 and 7, we find that when the system noise decrease, the no motion model

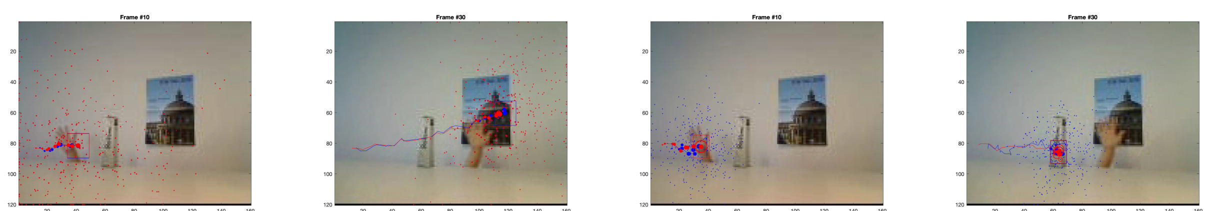


(a) Frame #16 (b) Frame #37 (c) Frame #16 (d) Frame #37

Figure 7: Tracking results of video 2, $\alpha = 0.5$, $\sigma_{position} = 3$, (a)(b): no motion model, (c)(d): constant velocity model

will get stuck in the case of occlusion while the constant velocity model can still track well. I think the reason is the particles are concentrated, so they will fail to keep track of the hand when it is occluded by something else. While the constant velocity model allow particles move forward, increasing the chance of capturing the hand again.

- What is the effect of assuming decreased/increased **measurement noise?**
  By looking at figure 8a, 8b, we can see that if the measurement noise is too large, the blue and red bounding box almost align with each other and the red spots are spread out. This means we rely on prediction model too much and the observation does not provide much information. Even the particle is far away from truth, it is considered as a good prediction. So if the prediction fails, the tracking fails as well.
  By looking at figure 8c, 8d, we can see that if the measurement noise is too small, the red spots are concentrated. In this case, we rely on observation too much, even the prediction differs little from the target, that particle will have very small weight. So in the case of occlusion, the tracking will fail.



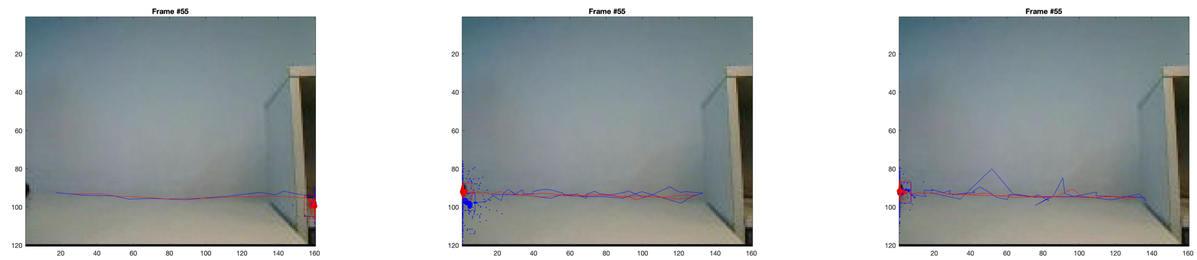
(a) Frame #10 (b) Frame #30 (c) Frame #10 (d) Frame #30

Figure 8: Tracking results of video 2, $\alpha = 0.5$, $\sigma_{position} = 15$, constant velocity model, (a)(b): $\sigma_{observe} = 1$, (c)(d): $\sigma_{observe} = 0.01$

2.3 Video 3: Ball bouncing

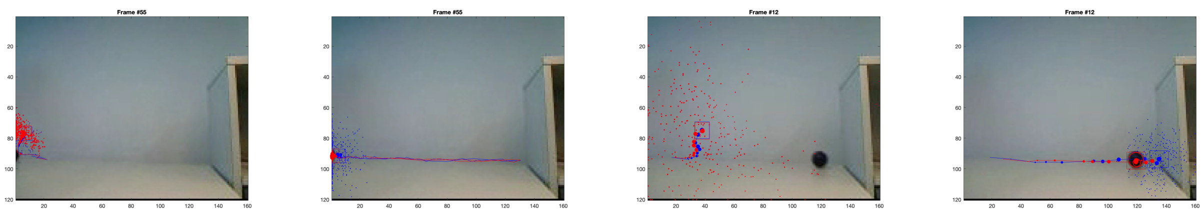
For video 3, I first use the best parameters for video 2 to try to track the ball. These parameters are: prediction model (ii), $\alpha = 0.5, \sigma_{position} = 3, \sigma_{observe} = 0.1, \sigma_{velocity} = 1$, initial velocity=$[5, 0]$. The tracking result is shown in figure 9a. We can see when the ball bounce, its velocity change a lot. If we use the constant velocity model, the tracking will fail after the rapid velocity change.

- What is the effect of using a **constant velocity motion model**?
  What is the effect of assuming decreased/increased **system noise**?
  These two questions can be answered together.
  If we keep the $\sigma_{position}$ as small as 3, both the no motion model and constant velocity model will fail, as shown in figure 9a and 10a. If we increase the $\sigma_{position}$ to 10, both models can track the ball well, as can be seen from figure 9b and 10b. But the tracking trajectory of no motion model is smoother



(a) $\sigma_{position} = 3, \sigma_{velocity} = 1$ (b) $\sigma_{position} = 10, \sigma_{velocity} = 1$ (c) $\sigma_{position} = 3, \sigma_{velocity} = 5$

Figure 9: Tracking results of video 3, $\alpha = 0.5$, constant velocity model, Frame #55

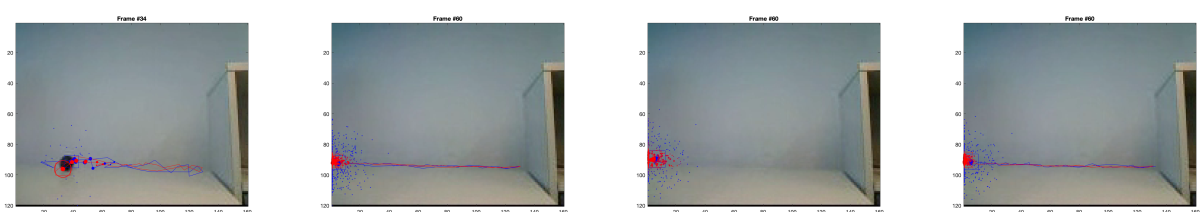


(a) $\sigma_{position} = 3$ (b) $\sigma_{position} = 10$ (c) $\sigma_{observe} = 1$ (d) $\sigma_{observe} = 0.05$

Figure 10: Tracking results of video 3, $\alpha = 0.5$, (a)(b) no motion model, Frame #55, $\sigma_{observe} = 0.1$, (c)(d) constant velocity model, Frame #12, $\sigma_{position} = 10$

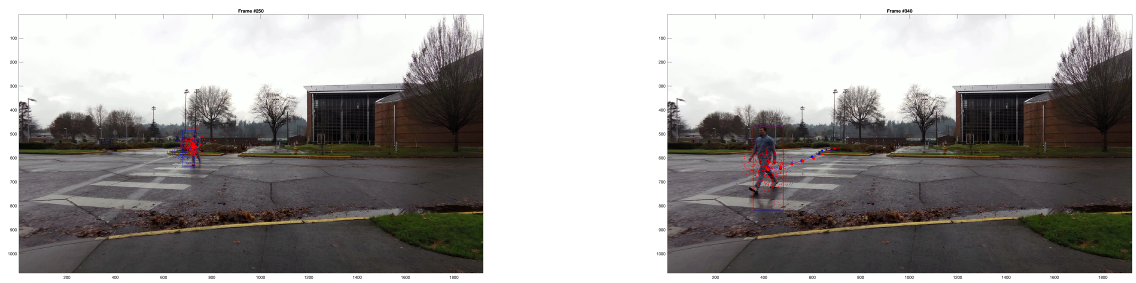
than that of constant velocity model. The reason of success is that large $\sigma_{position}$ allows particles fully spread out and keep track of the ball. The same idea can be applied to $\sigma_{velocity}$. If we increase the $\sigma_{velocity}$, the constant velocity model will also be able to track the ball. The only problem is large $\sigma_{velocity}$ leads to large variance of trajectory, as can be seen in figure 9c.

- What is the effect of assuming decreased/increased **measurement noise?**
  By looking at figure 10c, we can see that if the measurement noise is too large, the red spots are spread out. Even the particle is far away from truth, it is considered as a good prediction. So if the prediction fails, the tracking fails as well.
  By looking at figure 10d, we can see that if the measurement noise is too small, the red spots are concentrated. In this case, we rely on observation too much, even the prediction differs little from the target, that particle will have very small weight. So in the case of bounce, the tracking will fail.



(a) num_particles=30 (b) num_particles=500 (c) hist_bin=2 (d) hist_bin=30

Figure 11: Tracking results of video 3, no motion model, (a)(b) different num_particles, (c)(d) different hist_bin



(a) Frame #250 (b) Frame #340

Figure 12: Tracking results of my own video, constant velocity model

2.4 Additional Questions

- What is the effect of using more or fewer particles?
  Comparing figure 11a and 11b, we find with more particles, although the computation increase a little, the tracking result get improved. While with fewer particles, the tracking will easily fail, in our case, it gets stuck on the half way and the variance of tracking trajectory is large.

- What is the effect of using more or fewer bins in the histogram color model?
  Comparing figure 11c and 11d, we find that with more bins in the histogram model, the tracking is more precise. While with fewer bins, histogram provide little information for weight computing, leading to the failure of tracking.

- What is the advantage/disadvantage of allowing appearance model updating?
  In theory, if we use formula 4 to update appearance model, the tracking will be more precise and robust. The reason is that the appearance of the target object may change during the movement, e.g. illumination changes. Allowing it gradually change can provide a more robust target histogram. However, in our experiments, we do not see much difference.

2.5 My own video: Pedestrian tracking

For my own video, I choose parameters as following: prediction model (ii): constant velocity model, $\alpha = 0.5$, num_particles=500, $\sigma_{position} = 30$, initial velocity=[-50,10], other parameters as default. The tracking result is shown in figure 12. By observing the video, I find that in my selected frames, the man is walking closer to the camera, so the body shape becomes larger. Therefore, instead of keeping the size of bounding box fixed, I expand the width and height of bounding box frame by frame, which improves the tracking result. The success lies in the constant pace and direction of his movement. However, this trick has its limitation. If the man turns back and keeps moving backward, the bounding box keeps expanding when the man is actually move far away from the camera.