Computer Vision

Condensation Tracker

Student: Yuchang Jiang

Date: 11 Dec. 2020

1. Implementation

1.1 Color histograms

$color\_histogram.m$ computes the color histogram of a defined bounding box in a frame. At first, the locations of bounding box (maximum and minimum of x and y) are checked to ensure they lie inside the image. Then histogram for each channel are generated given the number of bins and they are combined to form $histogram^3$.

1.2 Derive matrix

For prediction, the linear stochastic differential equation is defined as:

$$s_t = As_{st-1} + w_{t-1} \tag{1.1}$$

Here s_t, s_{t-1} are particles, A matrix is used to update the deterministic part of prediction model and w_{t-1} is the stochastic part of model. In this exercise, we defined two kinds of models: 1) no motion 2) with constant velocity. Different model have different corresponding A matrix.

For no motion model, there is no movement so $[x_{pos}, y_{pos}]_t = [x_{pos}, y_{pos}]_{t-1}$ for deterministic part. Then A can be derived:

$$A = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}. \tag{1.2}$$

For model with constant velocity, positions should be updated based on velocity: $[x_{pos}, y_{pos}]_t = [x_{pos} + v_x, y_{pos} + v_y]_{t-1}$ (here the time difference is set as 1). Therefore, A can be derived:

$$A = \begin{vmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}. \tag{1.3}$$

1.3 Propagation

$propagate.m$ tries to propagate particles based on derived A and defined noise for prediction model: $s_t = As_{st-1} + w_{t-1}$. For model with no motion, w_{t-1} is the noise generated by given $sigma\_position$ for x, y position. For model with constant velocity,w_{t-1} has more noises: the noise generated by given $sigma\_velocity$ for x, y velocity. With derived A matrix and computed random noise for position and velocity, particles can be propagated.

1.4 Observation

$observe.m$ computes the weights for particles based on color histogram. At first, ensure the bounding box for each particle lies inside the image. Then for each particle, compute the color histogram of the bounding box with the center at particle. The similarity between the color histogram of particle and target color histogram is computed by $\chi^2\ distance$. Then the probability for each particle is obtained with the following formula:

$$\pi = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\chi^2(s,target)^2}{2\sigma^2}} \tag{1.4}$$

Here $s, target$ means the color histogram of a certain particle and target.

1.5 Estimation

$estimate.m$ computes the mean states given the folloing formula:

$$E[s_t] = \sum \pi_t s_t \tag{1.5}$$

It means the mean state at a certain time is the weight average of all particles at this time (use weights from observation step).

1.6 Resampling

$resample.m$ tries to resample particles with replacement based on their weights from observation step. For resampled new particles, their weights are normalised.

To summarize, those steps explained above can be used in the following sequence to form CONDENSATION tracker:

1. compute the histogram of the selected bounding box as target histogram
2. initialize particles
3. propagate particles and estimate the mean state right now
4. observe the weights for current particles and update mean state based on weights

5. update color histogram based on histogram of target and mean state
6. resample particles based on weights
7. repeat step 3-6 until the end of video

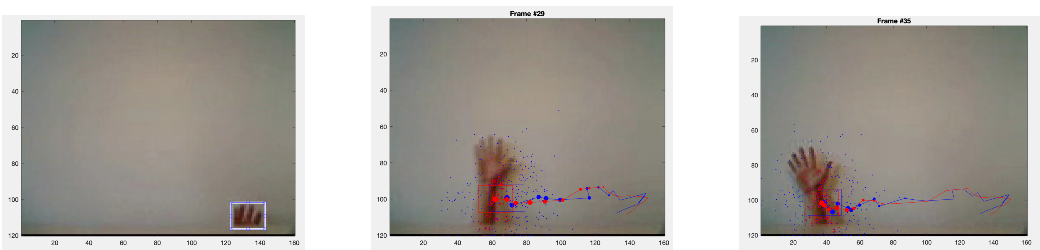
2. Experiments

2.1 Video 1

Here I tried video 1 for built tracker and results are shown in Figure 1 to Figure 3 (blue means prior and red mean posterior). The corresponding parameter settings are listed in Table 1. For the results of video, the mean trajectories from no motion model and constant velocity model are similar. Although the initial bounding box is about hand, all trajectories are linked to arm in the end. Because color histogram is used for measurement and colors of hand and arm are similar, tracker tends to follow arm in the end. This problem will be obvious when using a large α as shown in Figure 3. Considering the formula used for updating target histogram: $hist_{target} = (1-\alpha)hist_{target} + \alpha hist_{mean}$. A larger α will make the color histogram of arm play more important role during tracking process.

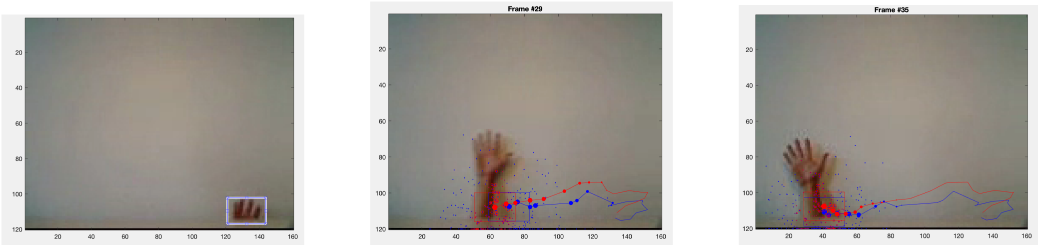
Figure	model	particle num	hist_bin	α	$\sigma_{observe}$	$\sigma_{position}$	$\sigma_{velocity}$
1	0	300	16	0	0.1	15	1
2	1	300	16	0	0.1	15	1
3	1	300	16	0.8	0.1	15	1

Table 1: Parameter for video 1



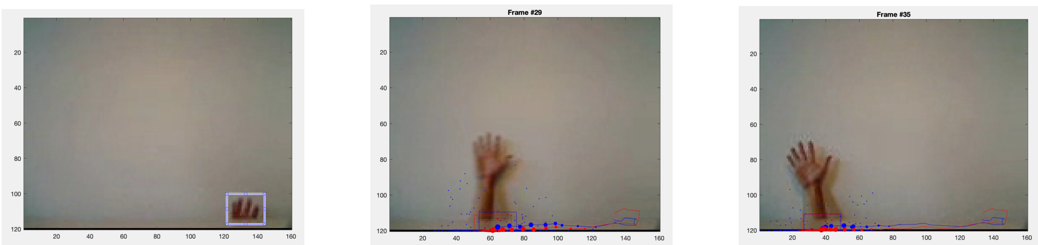
(a) (b) (c)

Figure 1: Result of video 1 (no motion, $\alpha = 0$)



(a) (b) (c)

Figure 2: Result of video 1 (constant velocity, $\alpha = 0$)



(a) (b) (c)

Figure 3: Result of video 1 (constant velocity, $\alpha = 0.8$)

2.2 Video 2

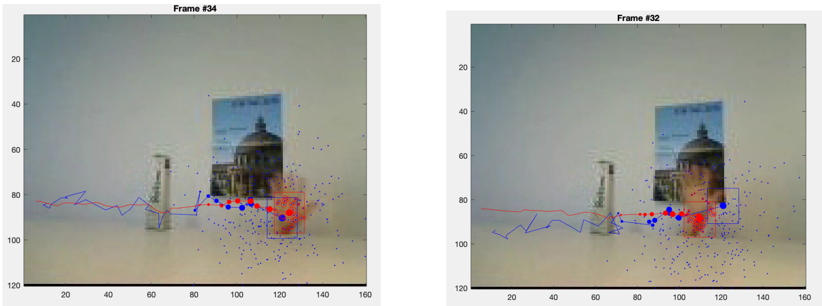
To check more complex scene with occlusion, video2 is used. Here the initial velocity is fixed at $[1\ 5]$. The following experiments will check the effect of model type, noise of dynamic model and noise of measurement. All settings of parameters are listed in Table 2.

As shown in Figure 4, results of two kinds of model are similar when using default parameters. However, as shown in Figure 5, results are different when using a small $\sigma_{position}$, no motion model totally fails while the constant velocity model can still work. This means constant velocity model is better in this case. On the other hand, if $\sigma_{position}$ is very large (as shown in Figure 6), both trajectories are rough which means the randomness dominates the tacker and it is hard to find the way back to hand. Thus, $\sigma_{position} = 15$ seems to be a good choice (neither too large or small).

For the influence of $\sigma_{observe}$, Figure 7 shows a large $\sigma_{observe}$ will make the observation too random and both models lose the tracking of hand. However, if $\sigma_{observe}$ is very small (as shown in Figure 8), the measurement of observation is too strict and it can be fooled by occlusion. Thus, $\sigma_{observe} = 1$ seems to be a good choice.

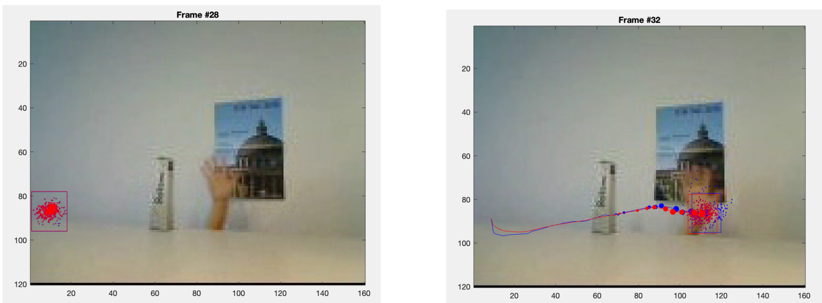
Figure	model	particle num	hist_bin	α	$\sigma_{observe}$	$\sigma_{position}$	$\sigma_{velocity}$
4(a)	0	300	16	0.5	0.1	15	1
4(b)	1	300	16	0.5	0.1	15	1
5(a)	0	300	16	0.5	0.1	1	1
5(b)	1	300	16	0.5	0.1	1	1
6(a)	0	300	16	0.5	0.1	30	1
6(b)	1	300	16	0.5	0.1	30	1
7(a)	0	300	16	0.5	1	15	1
7(b)	1	300	16	0.5	1	15	1
8(a)	0	300	16	0.5	0.01	15	1
8(b)	1	300	16	0.5	0.01	15	1

Table 2: Parameter for video 2



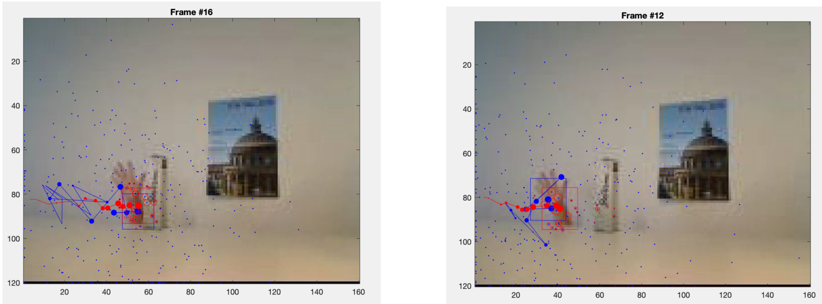
(a) (b)

Figure 4: Result of video 2 (left=model 0, right=model 1)



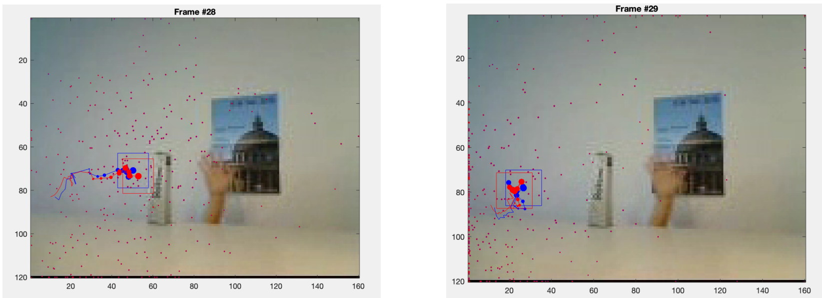
(a) (b)

Figure 5: Result of video 2 (left=model 0, right=model 1)



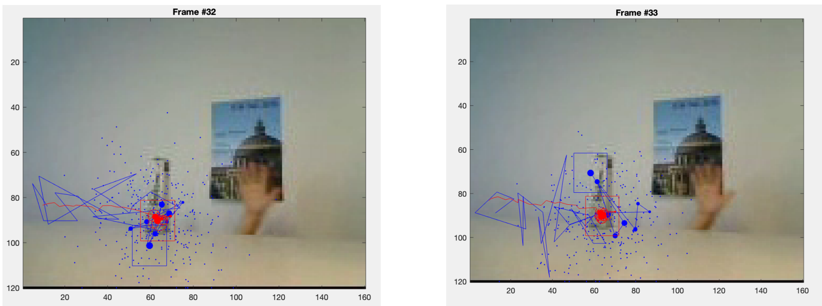
(a) (b)

Figure 6: Result of video 2 (left=model 0, right=model 1)



(a) (b)

Figure 7: Result of video 2 (left=model 0, right=model 1)



(a) (b)

Figure 8: Result of video 2 (left=model 0, right=model 1)

Based on previous experiments, suitable parameters are: constant velocity model, $\sigma_{position} = 15$ and $\sigma_{observe} = 0.1$. The corresponding trajectory is displayed in Figure 9.

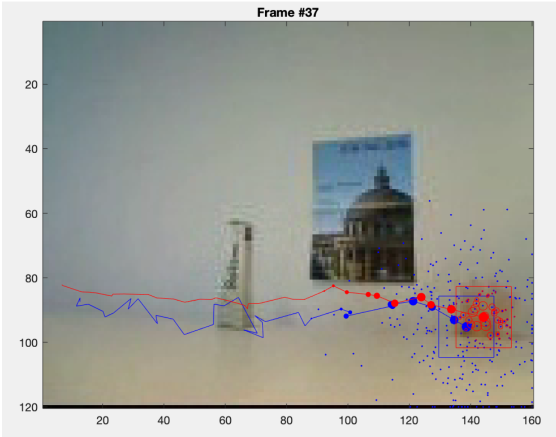


Figure 9: Result of video 2 (best parameters)

2.3 Video 3

Based on experiments of video 2, chosen parameters are used to test video 3, where a small ball is bouncing on the table. From Figure 10, it shows the tracker can track the ball successfully. The same experiments are conducted to check the influence of model type, noise of dynamic model and noise of measurement and the results are shown in Figure 11 to Figure 14. Based on results, similar conclusion of parameters can be drawn. In general, constant velocity model is better than no motion model. Large σ increases the randomness in prediction and measurement so an extremely large σ may make tracker lose control while an extremely small σ will make model stick to deterministic part and be sensitive to changes like occlusion or movement direction change.

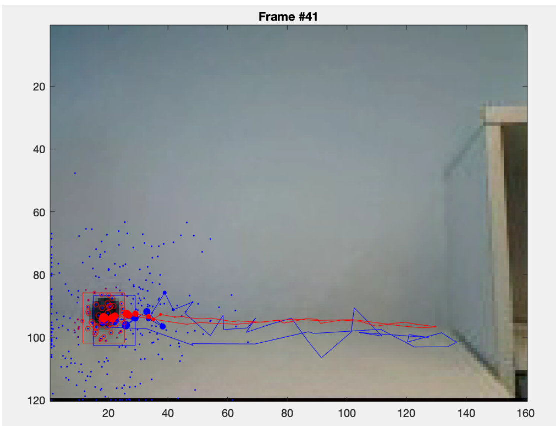
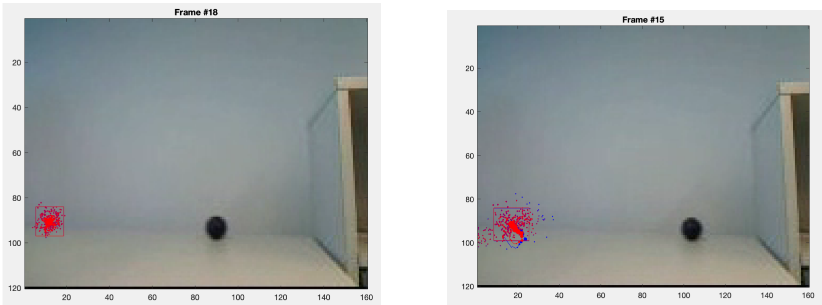


Figure 10: Result of video 3 (best parameters from video 2)

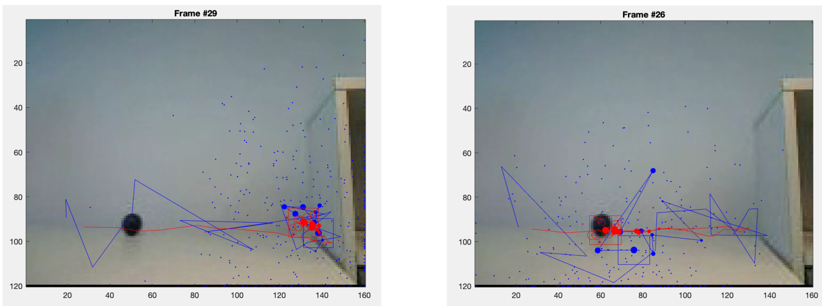
Figure	model	particle num	hist_bin	α	$\sigma_{observe}$	$\sigma_{position}$	$\sigma_{velocity}$
11(a)	0	300	16	0.5	0.1	1	1
11(b)	1	300	16	0.5	0.1	1	1
12(a)	0	300	16	0.5	0.1	30	1
12(b)	1	300	16	0.5	0.1	30	1
13(a)	0	300	16	0.5	1	15	1
13(b)	1	300	16	0.5	1	15	1
14(a)	0	300	16	0.5	0.01	15	1
14(b)	1	300	16	0.5	0.01	15	1

Table 3: Parameter for video 3



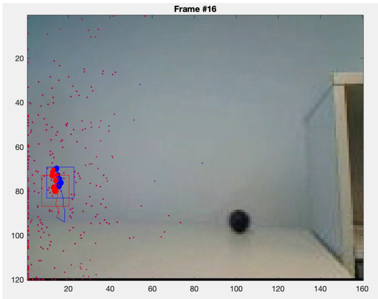
(a) (b)

Figure 11: Result of video 3 (left=model 0, right=model 1)

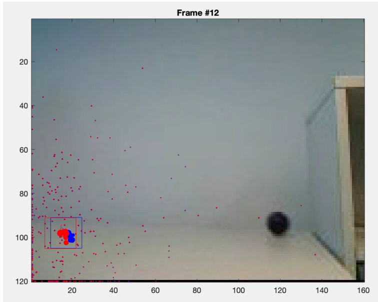


(a) (b)

Figure 12: Result of video 3 (left=model 0, right=model 1)

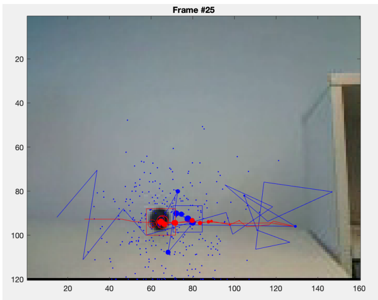


(a)

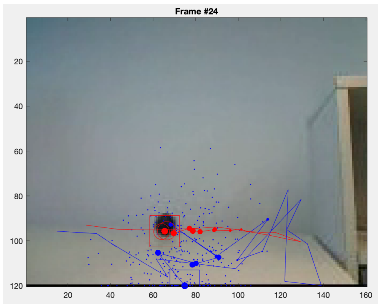


(b)

Figure 13: Result of video 3 (left=model 0, right=model 1)



(a)



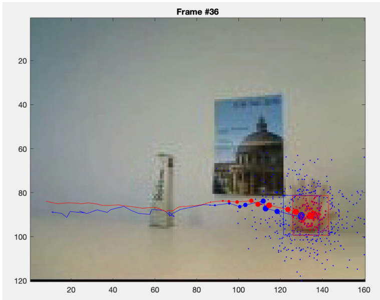
(b)

Figure 14: Result of video 3 (left=model 0, right=model 1)

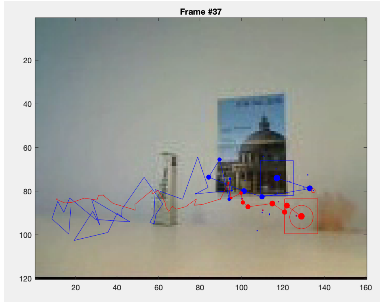
2.4 Influence of other parameters

After other experiments, the answers to the following questions are also figured out (here model with constant velocity is used).

1. Effect of the number of particles
With a large number of particles, tracker can perform well but it takes longer computation time. With a very small number of particles, tracker can still track the object but the trajectory looks rough. As shown in Figure 15, the trajectory of 500 particles looks more smooth than that of 20 particles.



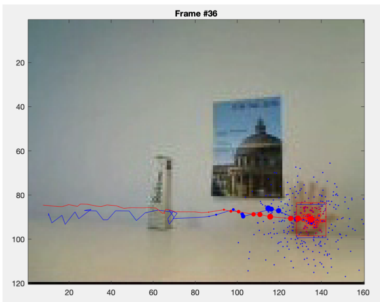
(a) 500 particles



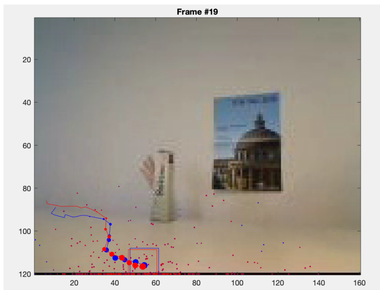
(b) 20 particles

Figure 15: Different number of particles

2. Effect of the number of histogram bins
With large number of bins, the description of color histogram is more accurate and detailed but it may be not necessary if a smaller number of histogram is already enough. However, if the number of bin is very small, it may be not enough to represent the object inside bounding box then tracker cannot find the object. As shown in Figure 16, 50 histogram bins can track hand successfully while 2 bins cannot.



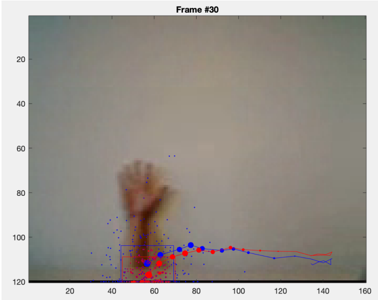
(a) 50 bins



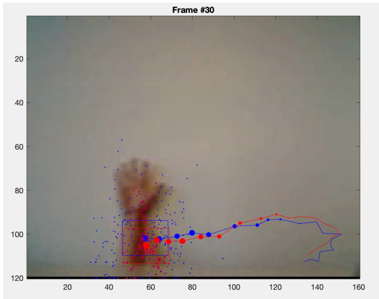
(b) 2 bins

Figure 16: Different number of histogram bins

3. Effect of the α for updating Based on the updating rule, $\alpha = 1$ means the updating term dominates while $\alpha = 0$ means no updating and the target histogram is always the initial histogram. As shown in Figure 17, if updating term dominates, the tracker will track the lower part of arm. The advantage of updating rule is the changes of object can be captured during tracking. The disadvantage is the error made during tracking will be accumulated.



(a) $\alpha = 1$



(b) $\alpha = 0$

Figure 17: Different alphas

2.5 Own Video

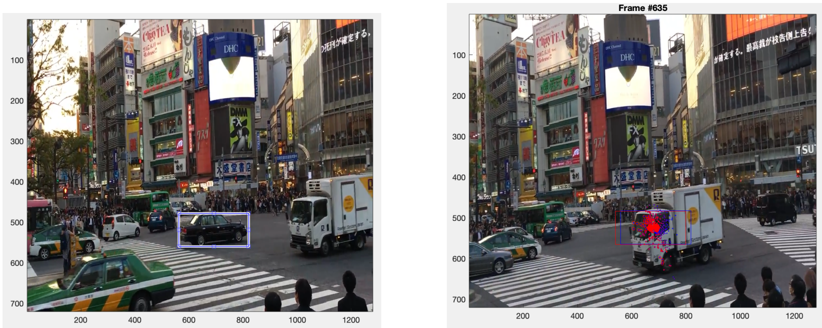
The used video is stored in data folder as 'myOwnVideo.mp4'. This is a video from Youtube showing the busy traffic in Japan. The busy traffic causes two problems for tracking a car:

1. Occlusion: from time to time, the car will by hidden by other cars
2. Similarity: many cars look similar to each other (e.g. car with similar color and size)
3. Camera movement: the camera is not fixed

The same parameters (best one from previous experiment) are used here: constant velocity model, $\sigma_{position} = 15$ and $\sigma_{observe} = 0.1$ and results are shown in Figure 18 and 19. The black car in initial bounding box is moving towards right and be occluded by a large white van. Because of this occlusion, tracker loses the target car (16a). Then there is another similar black car appears near the particles and the tracker turned to track this car instead.

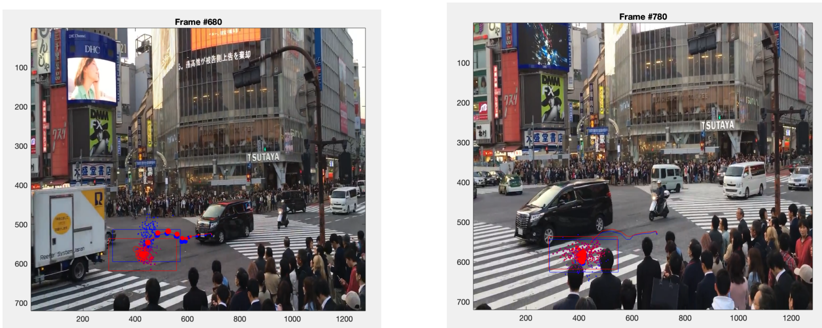
I tried to tune the parameter but the problem still exists. Unlike the occlusion in video 2, the occlusion, the large white van, in my own video hides the target car for quite a long time so it is harder to track. Besides, the angle of camera is changing during the video, which causes another problem for tracker.

For the large occlusion problem, the object may disappear for a long time. Maybe using momentum can help it: let particle remember the momentum of movement and use it when object disappears. For the similarity problem, we can use a more accurate descriptor: besides color histogram, descriptor also tries to describe the shape of object. For the problem of camera movement, if we assume camera moves with a constant velocity, extra terms describing the x, y velocity of camera can be used and the relative movement of particles can be derived.



(a) (b)

Figure 18: Result of my own video (left=frame 630, right=frame 635)



(a) (b)

Figure 19: Result of my own video (left=frame 680, right=frame 780)