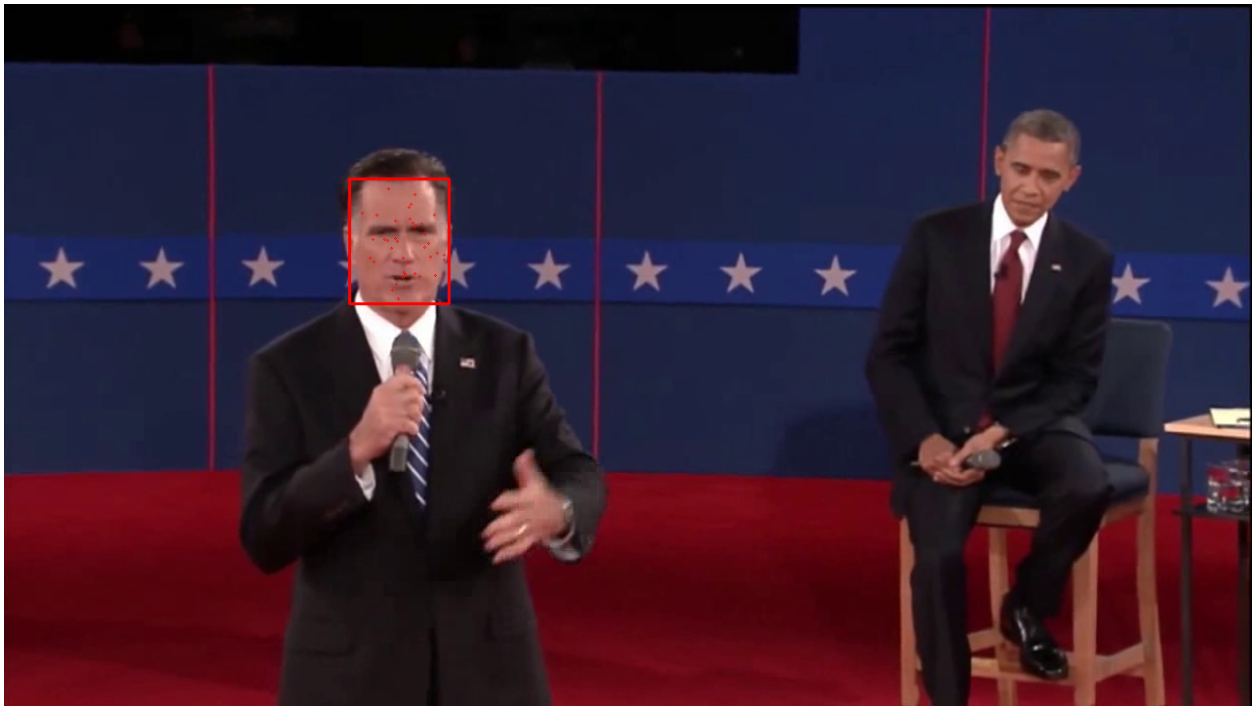
Jonathan Hudgins (jhudgins8@gatech.edu)
GTID: 903050550
CS4495 Spring 2015 OMS
ps6

For this problem set I did all calculations in the color space.

Question 1a:
Patch (ps6-1-a-1.png)

Frame 28 Simple Face Tracking (ps6-1-a-2.png)



Frame 84 Simple Face Tracking (ps6-1-a-3.png)

Frame 144 Simple Face Tracking (ps6-1-a-4.png)

Question 1b:
I was a little surprised that even when I reduced the window size to 1/32nd of the original, the tracking never lost the face. This is probably because the colors in the face are not really present anywhere near (or even anywhere at all) in the image.

Advantages of larger window size:

- Less stochastic because entire face is captured
- Centered -- instead of moving around to different parts of the face
- Better chance of not getting confused by other objects (like the hands)

Advantages of smaller window size:

- Improved performance
- Will not be as confused by changing background
- Easier to debug

Question 1c:
The σ_{MSE} parameter adjusts how heavily to weight the particles with the best mean-squared-error (MSE). That is a lower σ_{MSE} will concentrate the probability density on the particles with the lowest MSE, while a higher σ_{MSE} will spread out the distribution more evenly among particles so that even those with high MSE have only slightly less probabilities than particles with low MSE.

This means that a high σ_{MSE} (for color a σ_{MSE} of 16) will fail to track the face even at frame 28 because the randomness of the particles dominates. A very low σ_{MSE}, however, will essentially drop all particles except the best particle. While this low σ_{MSE} can still perform decent for a case like the presidential debate video, it will have difficulty if the face is occluded or moves farther than the limits of random particle motion.

Question 1d:
Optimized number of particles for this image: 10
There are surprisingly good results with *num_samples* even as low as 10. The resulting window contains only slightly less face than with *num_samples* at 640. However, I believe this is due to the fact that other parameters are tuned well for this image.

The advantage of higher number of particles is more robust results when motion between frames was greater or more random, or when the object being tracked is less distinguishable from the background.

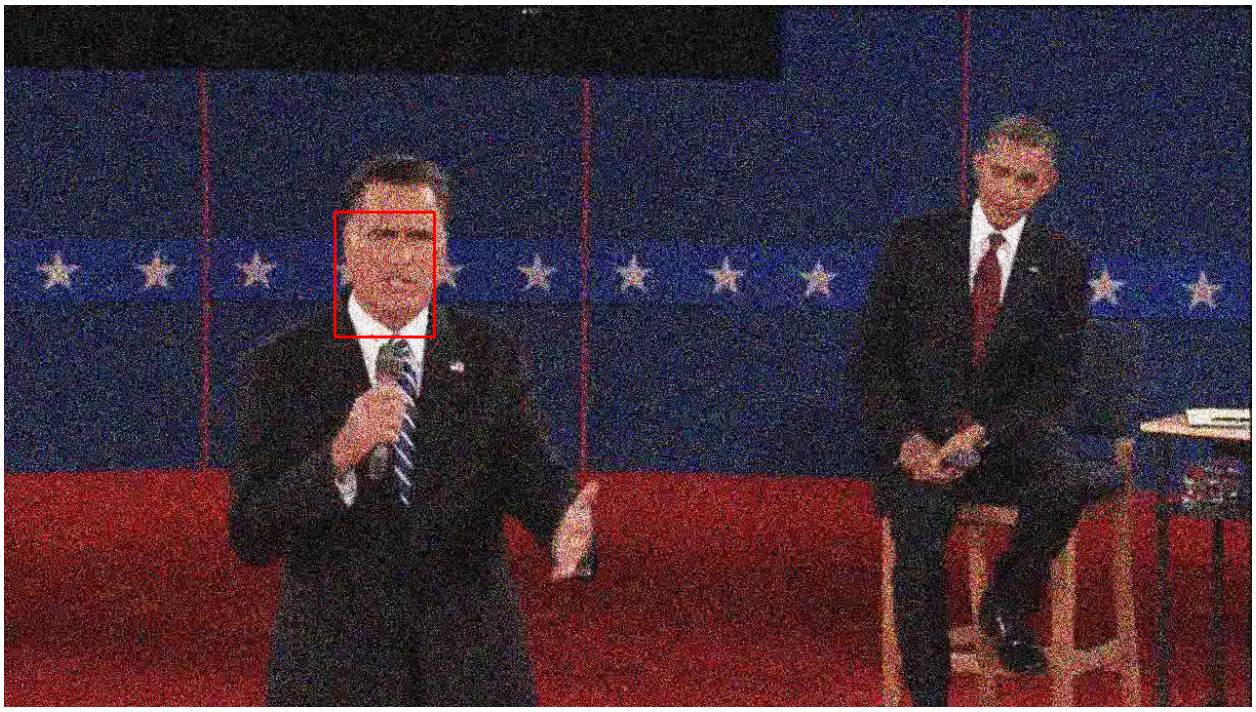
The advantage of smaller number of particles is the algorithm can run much faster. For runtime applications, the right number of particles is important.

In generic applications I would probably start with 50-100 particles and test many examples to see what lowest value still provides good results.

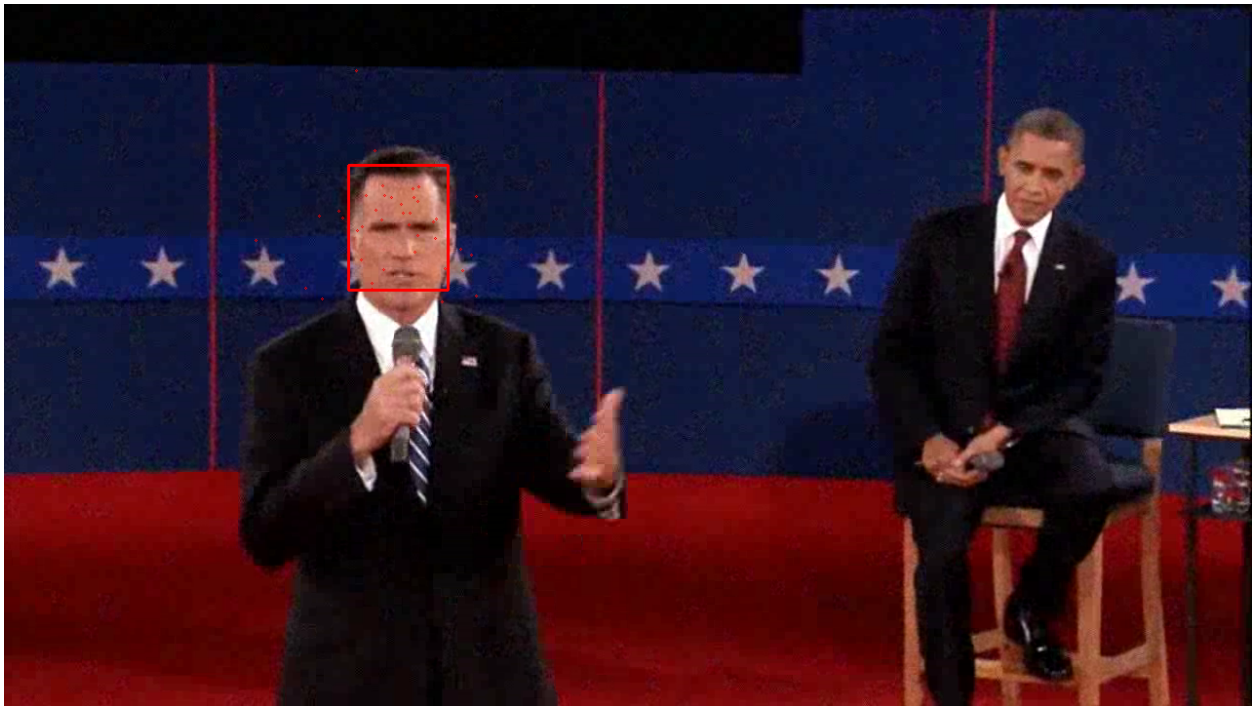
Question 1e:

I have 5 parameters in code which I tuned one at a time trying a range of values. After examining the result of a parameter, I used the best value for that parameter for subsequent experiments. The values almost identically matched the values I had tuned for in Question 1a. The only difference is using σ_{MSE} = 4.0 instead of the value of 2.0 that produced better results without the noise.

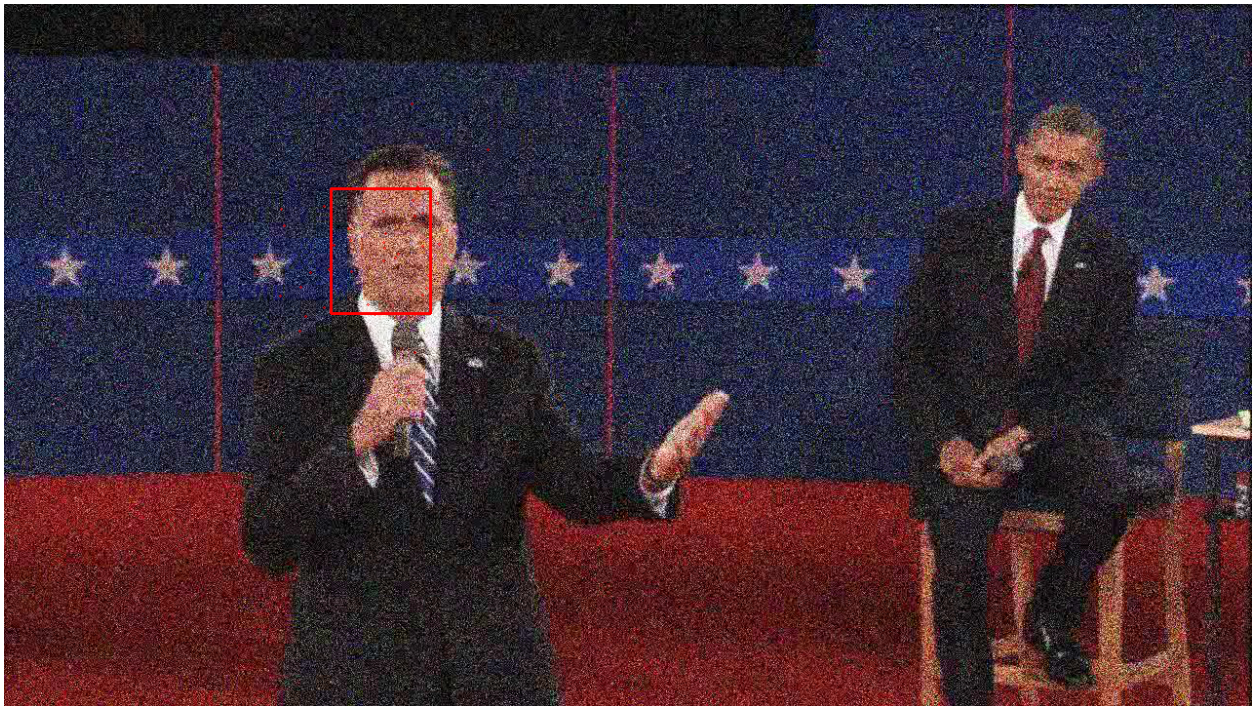
Frame 14 Noisy Face Tracking (ps6-1-e-1.png)



Frame 32 Noisy Face Tracking (ps6-1-e-2.png)



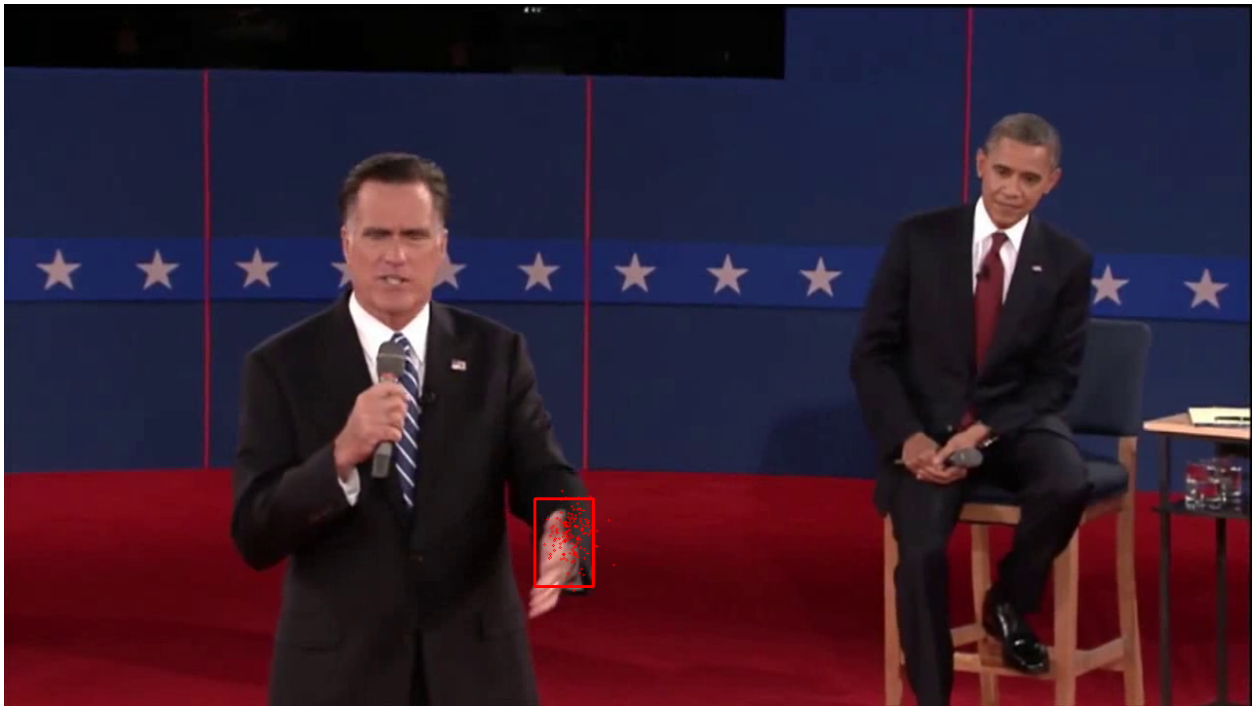
Frame 46 Noisy Face Tracking (ps6-1-e-3.png)



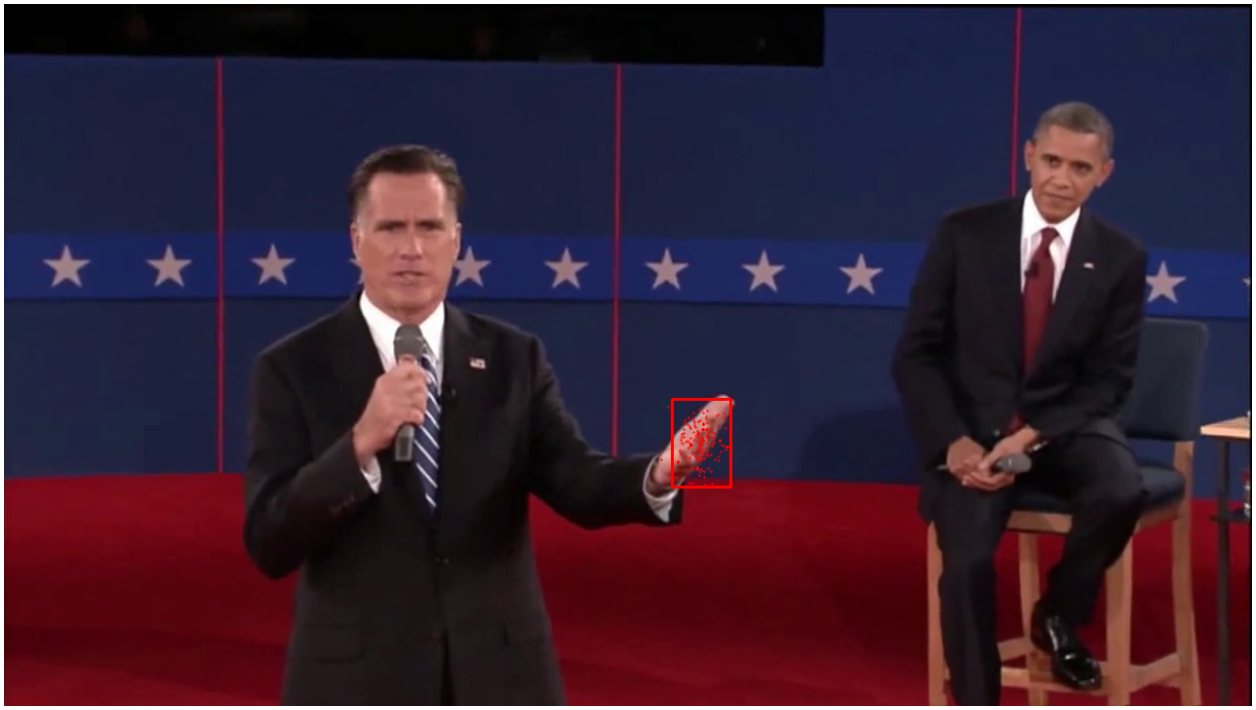
Question 2a:

Hand Tracking Update Patch (ps6-2-a-1.png)

Frame 15 Hand Tracking Update Image (ps6-2-a-2.png)



Frame 50 Hand Tracking Update Image (ps6-2-a-3.png)

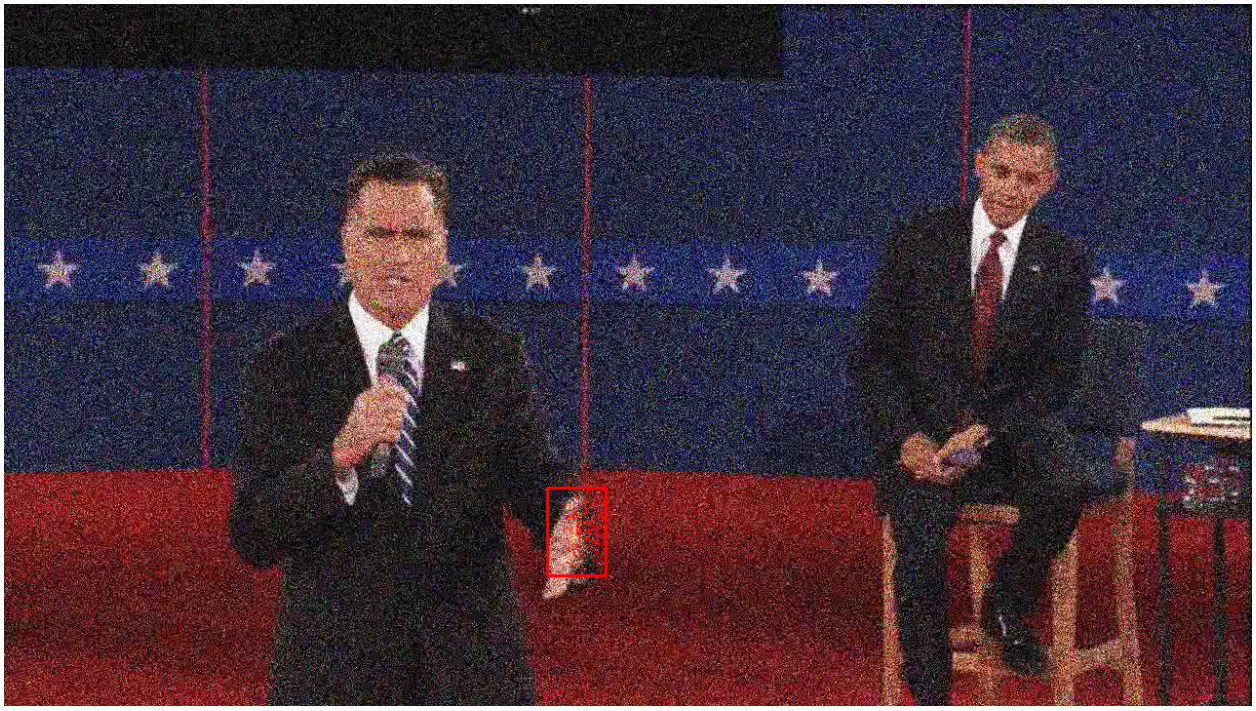


Frame 140 Hand Tracking Update Image (ps6-2-a-4.png)

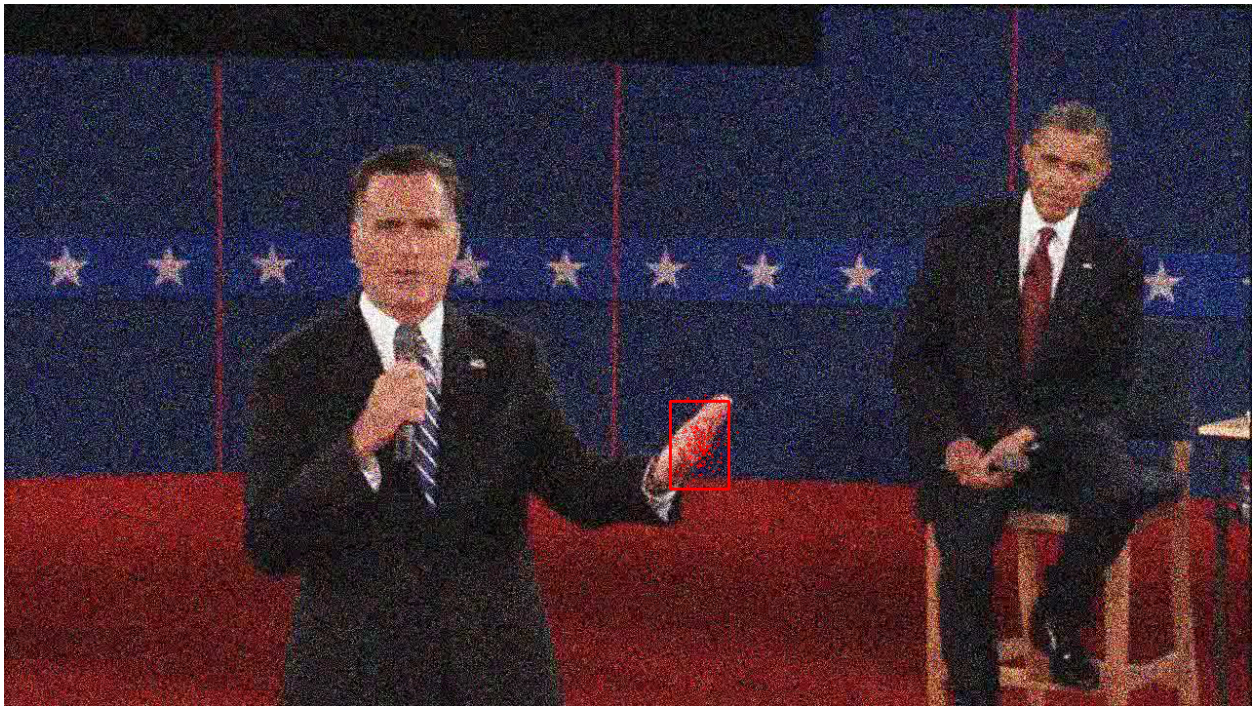
Question 2b:

Patch Hand Tracking Update Noisy (ps6-2-b-1.png)

Frame 15 Hand Tracking Update Noisy (ps6-2-b-2.png)



Frame 50 Hand Tracking Update Noisy (ps6-2-b-3.png)



Frame 140 Hand Tracking Update Noisy (ps6-2-b-4.png)

I did not have to change any parameters from part 2.a to part 2.b. The parameters with the given algorithm were robust enough to get great results. I believe this is the case, because even with noise, updating the patch enables the code to maintain a valid concept of the object.