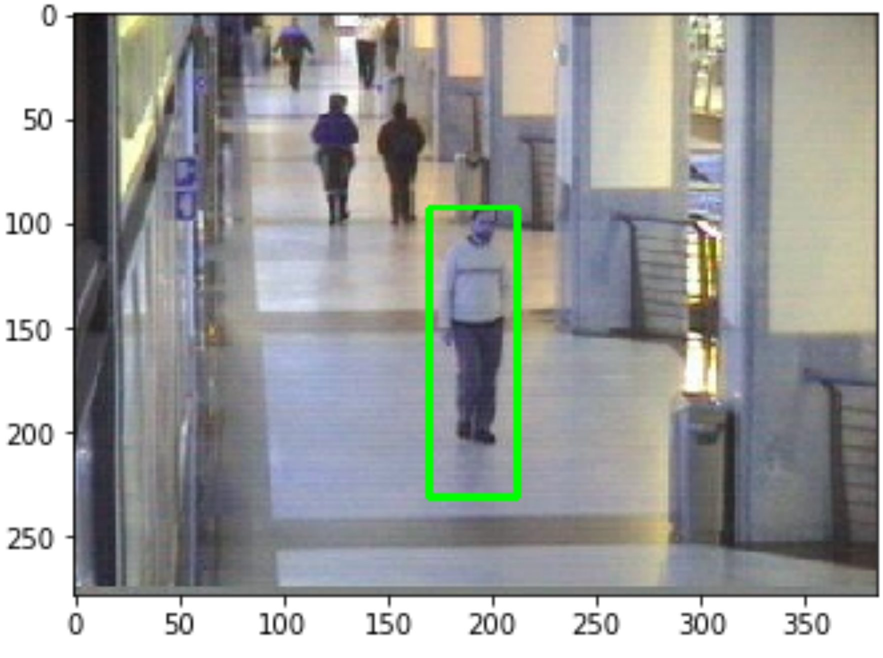
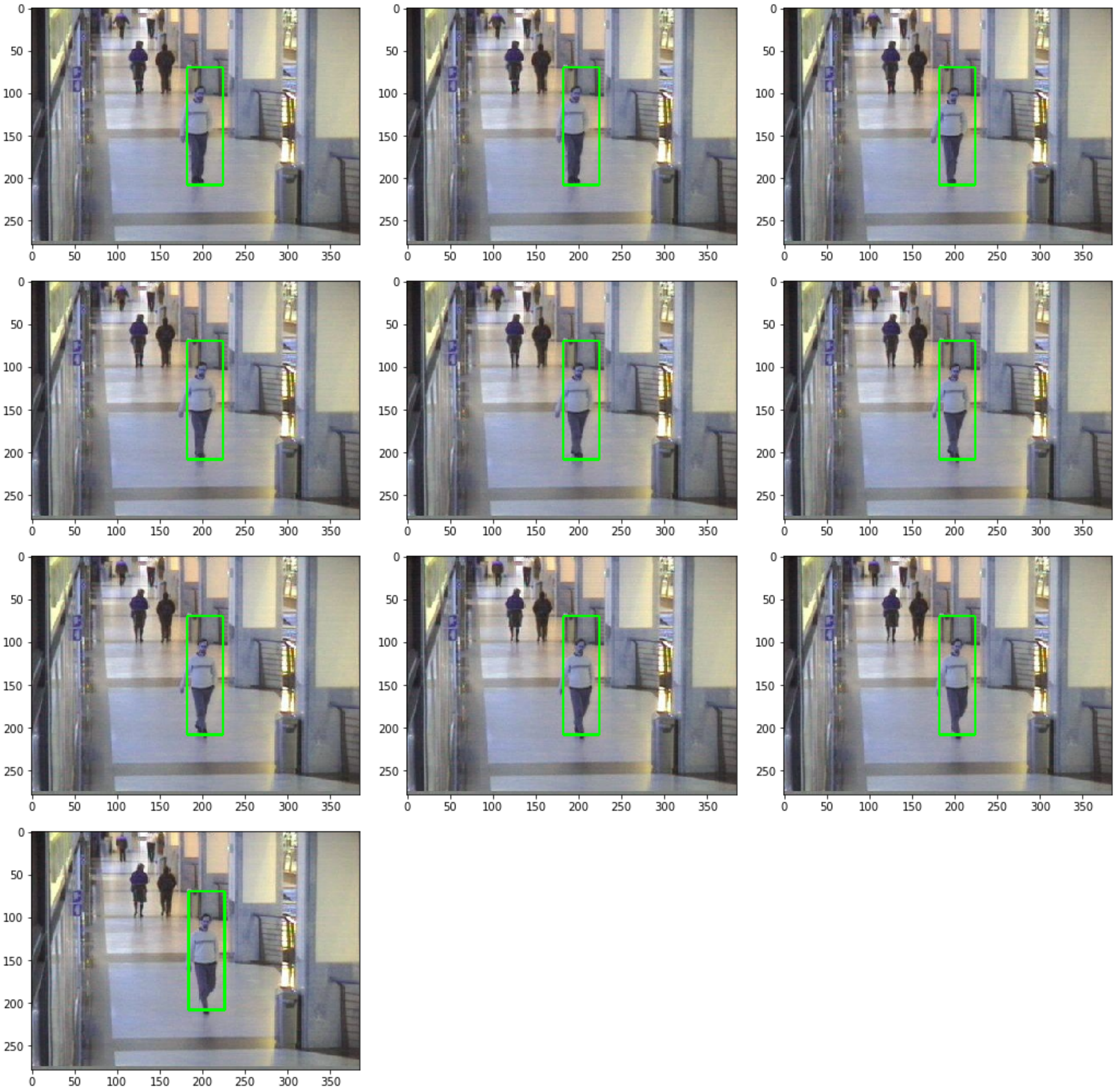
Problem1

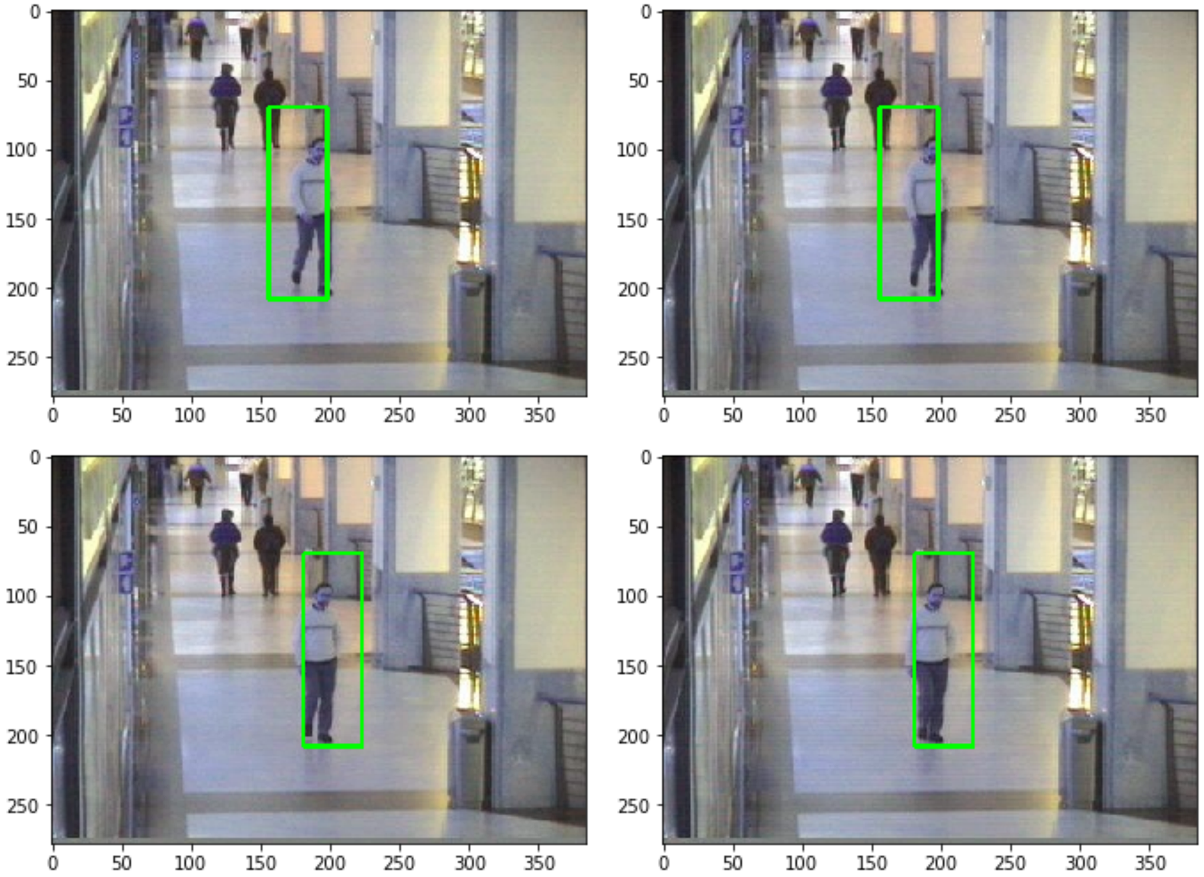
We have created our own method of background subtraction and object tracking. Using the function cv2.BackgroundSubtractorMOG() only gives a crude result (also on mac this function does not work so we replaced it with cv2.createBackgorundSubtractorMOG2()). So we applied morphological post processing to get rid of isolated points (not relevant points). Thus we were able to get contours for all the people in the frame. To get the contour for the man, we observed that his contour was the biggest and hence used this observation to extract contour corresponding to him. The following image shows the result of our algorithm.



We then used this block as the reference for an exhaustive block searching algorithm for the next 10 frames of the video. We used grayscale intensity values of the block as the features to calculate similarity. The algorithm proved to be robust and gave good results for the next 10 frames.



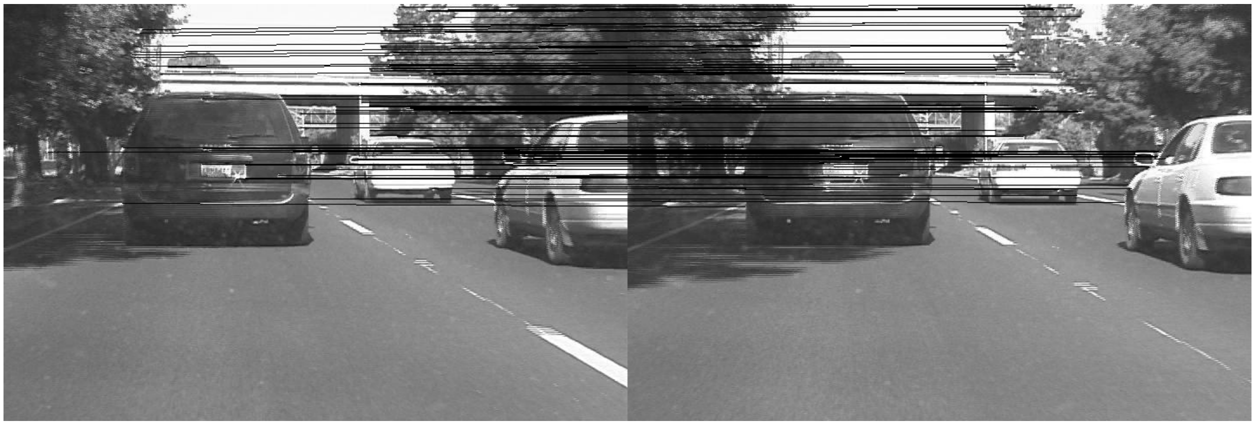
To check the robustness of our algorithm we decided to match the blocks starting from frame 1 and see if the algorithm works for frames far away from the 10th frame. Turns out, our algorithm worked perfectly in detecting the person, even when there was a sudden shift, as is evident from the images below.



So we see that when the person starts moving out of the frame, the block matching algorithm is able to find the person in the next frame.

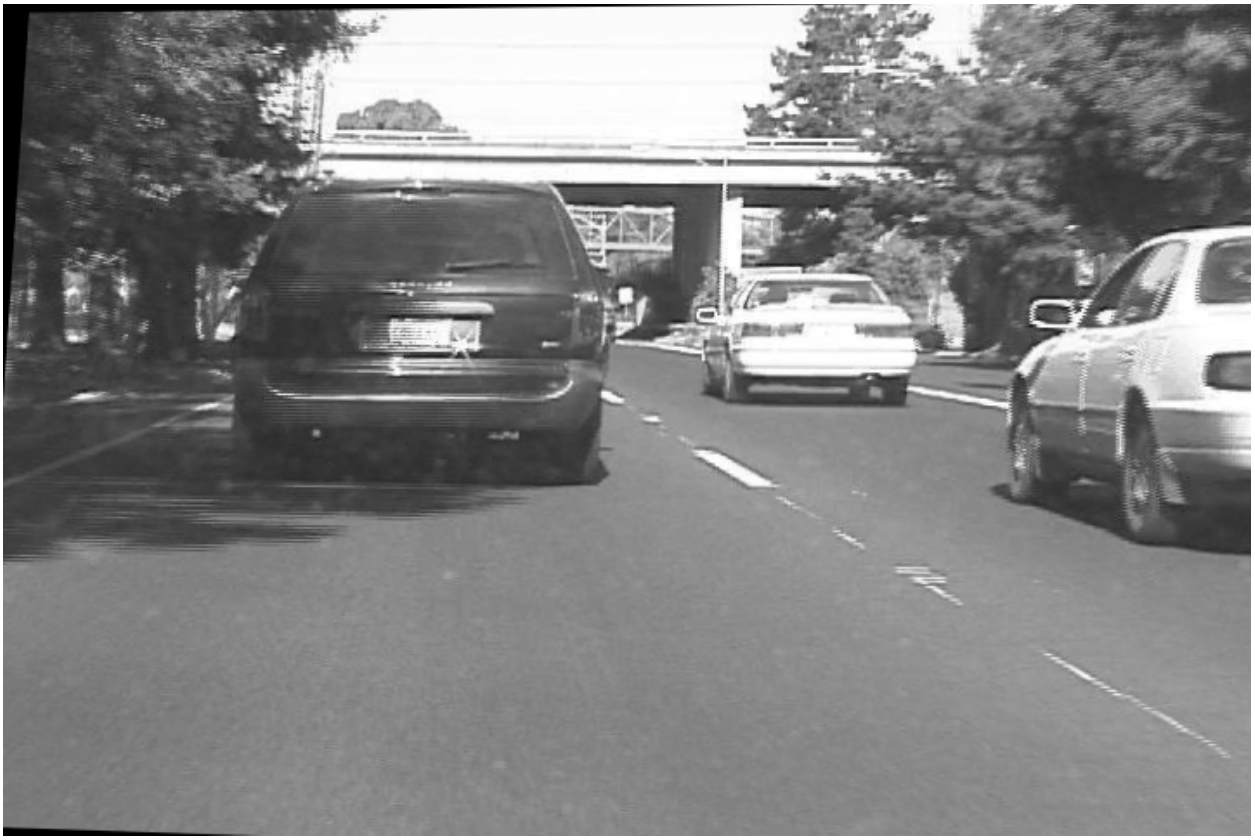
Problem 2

In this problem we address the issue of moving camera. We detect Harris feature points in Frame 1 and find the corresponding points in Frame 4 using Optical Flow Lucas Kanade Method (We used frame images provided by the TA). We then draw lines to see the matching points in both the images.



Thus we see that the corresponding points are well connected with the lines. Next we calculate the Homography mapping between Frames 1 and 4 using these corresponding points using the least square method. The original images are

The warped Frame 4 after warping it using the Homography mapping found is as shown below



Finally we detect the moving object (a binary image) using cv2.BackgroundSubtractorMOG() and the result is like this