Overall Approach:

Our approach began by deciding what we wanted to center our project around. Both of us being runners, we decided it could be a cool idea to track a professional runner's foot while they run. This initially seemed a lot easier than it turned out to be. We thought that because runner's typically wear bright colored shoes, thresholding on RGB values would be simple, but we forgot to account for the numerous colors present in the crowd in most videos. Also, the trend of bright shoes is a more recent phenomenon. A lot of the videos we found of runners that seemed like they could be good candidates were wearing white shoes, which would be heard to threshold given that parts of the actual track were white. Luckily we found a video of Jakob Ingebrigtsen in the Olympic 5k, where his shoe color was very distinct from the track, and he was far away from any other runners, so we could isolate him. We then began the coding portion by thresholding on the color of his shoes, but ran into difficulties because there was a bright strip in the crowd of similar color caused by a strong reflection from the sun. Thus, we decided to crop our video to include just his lower half, allowing us to remove the crowd and most of the annoying parts of the background completely. From there, we were able to relatively successfully threshold his shoe. Then we ran the frames through morphological operators. We decided to dilate the image to try to get a nice shoe outline first and fill in any holes in the shoe, and then we would erode away the extraneous artifacts. It initially seemed more logical to erode away the tiny specks of salt noise first and then dilate, but this ended up making the outline of the shoe far worse. Thus, we kept our original order despite it not being perfect. We then found the contours of the feet, but had to do some manipulating to remove any contours it found of some flaws in our morphological operators. We decided to order the contours by area and keep only the two largest which always happened to be the runner's two feet. This allowed us then to store the centroid of each of the two feet in each frame and plot them to get a nice plot showing the trace of both of the runner's feet over the entire video. Our 'cool factor' was sort of two fold. We wanted to have something that the user could change to affect the output of the code, but we also wanted to tie the project a little more closely to our own experiences in running. Joey is a long distance runner, while Vir is more of a middle distance/sprinter, so we decided to use our code on a second athlete rather than another similar video of Jakob Ingebrigtsen. Thus, we chose to use our code for Matthew Hudson Smith, a 400m athlete, and compare his stride to the best long distance runner in the world, Jakob Ingebrigtsen. The 400 runner's shoe was not exactly the same as Jakob's, so the mask isn't great, but it still is able to capture his stride path really well. Although the camera angle of the two videos is not identical, their strides appear to be very similar which we thought was interesting. However, as expected, the 400 runner has a slightly longer stride, it appeared. Then adding further, we allowed the user to input the size of the structuring element to see how it would change the results of the plot. See images below for a comparison of frames before and after morphological operators, different structuring element size, and a comparison of the two athletes' strides.

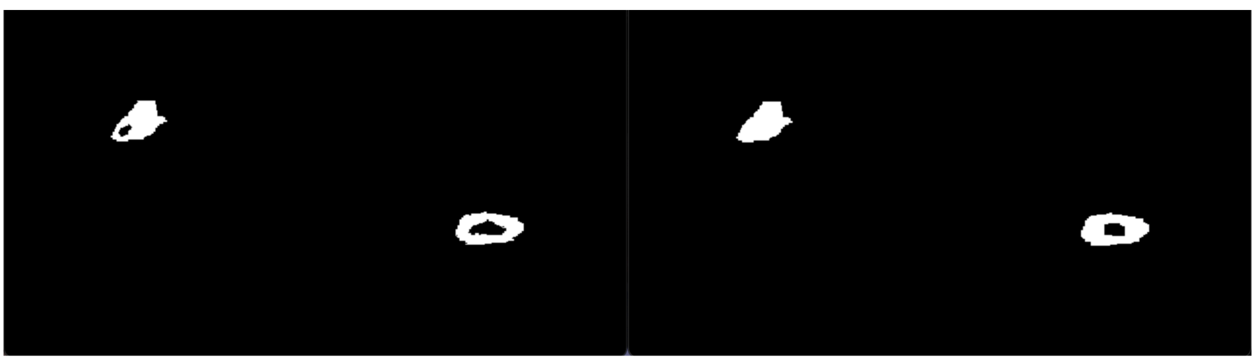
Before vs After morphological with default structuring element size

Before After

On the left, we see that there are gaps in his shoe that are filled by the morphological operators.

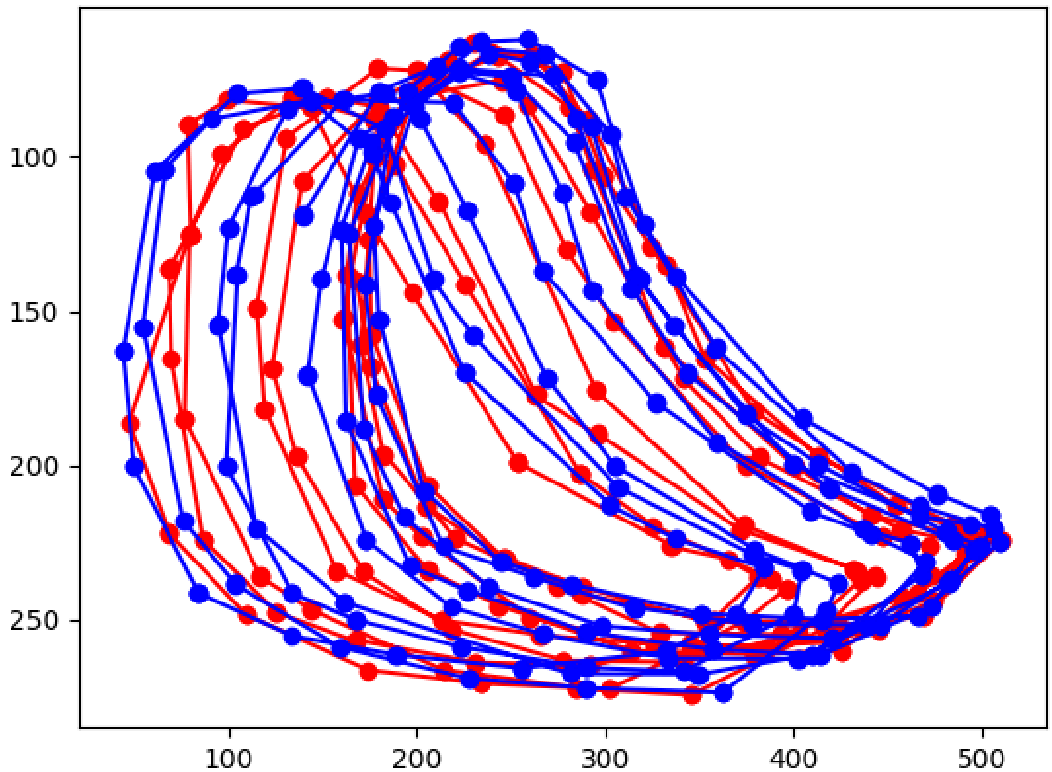
Before vs After morphological operator with structuring element specified at 10x10 rectangle

Before After

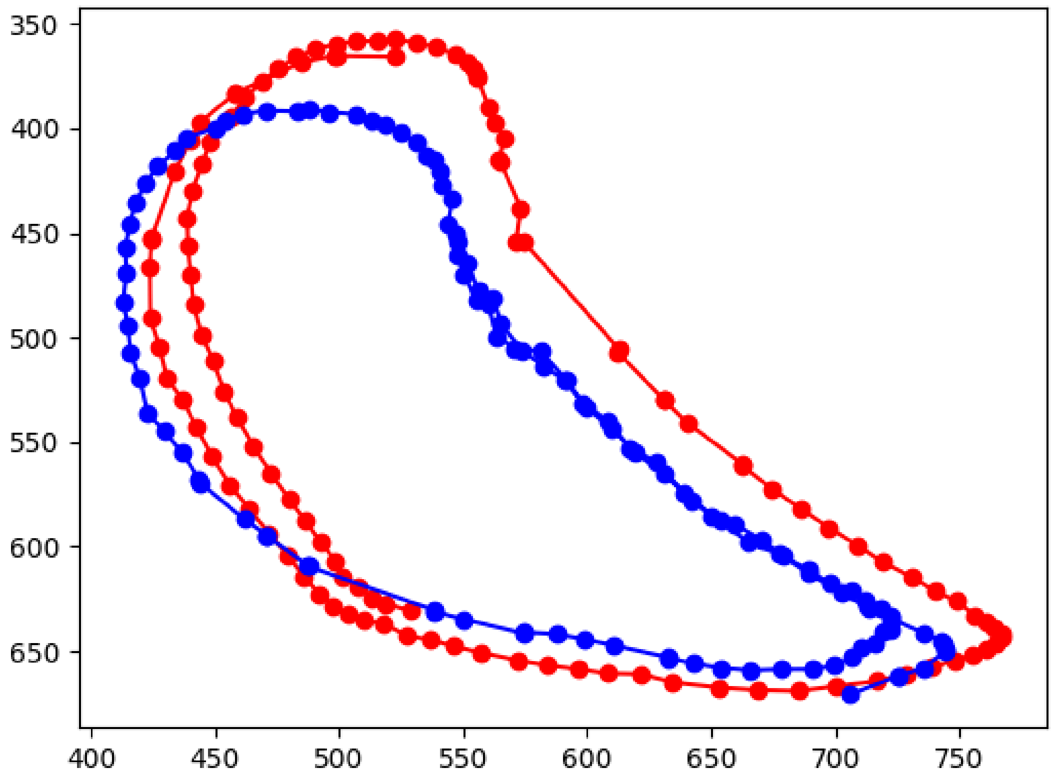


We see here, that with a smaller structuring element, we don’t completely remove gaps in his shoes.

Jakob Ingebrigtsen stride path: Blue = right foot   Red=left foot



Matthew Hudson Smith (400 runner) stride path: Blue = right foot Red = left foot



PSEUDO Code

1. process_image ()

- **Input:** A single frame from the video, a list of previous centroids, and the last detected "left" foot position.
- **Steps:**
    1. Define a color to detect the shoe.
    2. Compute the difference between each pixel in the frame and the target shoe color.
    3. Create a binary mask to isolate pixels that are close to the shoe color.
    4. Apply noise reduction using morphological operations (dilation and erosion).
    5. Find contours
    6. Keep only the two largest contours
        - If more than two contours are found, keep only the largest two.
        - If fewer than two contours are found, return the frame as-is.
    7. Assign the correct contour to the left foot:
        - If it's the first frame, assume the first detected contour is the left foot.
        - Otherwise, compare both contours to the previous left foot position.
        - Assign whichever contour is closer to the previous left foot centroid as the first contour in the list.
    8. Draw the contours on the frame and store the foot centroids.
    9. Return the processed frame and the updated left foot position.

---

2. main ()

- **Input:** A video file.
- **Steps:**
    1. Open the video file.
    2. Reset the video to the first frame.
    3. Initialize an empty list to store foot positions.
    4. Initialize a variable to track the left foot’s position.
    5. Loop through each frame:
        - Read the frame from the video.
        - Process the frame to extract foot contours.
        - Display the processed frame with contours drawn.
        - Exit when the user presses 'q'.
    6. Plot the foot trajectory:
        - Extract the x, y coordinates of the centroids.
        - Plot the trajectory of the foot over time.

Input video credit:

Matthew Hudson Smith: https://www.youtube.com/watch?v=F6qcv_jaS8I

Jakob Ingebrigtsen: https://www.youtube.com/watch?v=pn7TrcRbZpY