

Assignment 1

OpenCV setup and point operations

1. The processing time for one video frame or image:

Processing Time per Frame: 0.013908 seconds
Processing Time per Frame: 0.037657 seconds
Processing Time per Frame: 0.009016 seconds
Processing Time per Frame: 0.009179 seconds
Processing Time per Frame: 0.010675 seconds
Processing Time per Frame: 0.008801 seconds
Processing Time per Frame: 0.016500 seconds
Processing Time per Frame: 0.028313 seconds
Processing Time per Frame: 0.005509 seconds
Processing Time per Frame: 0.015015 seconds

The mean processing time per frame is 0.01546 seconds

2. How does the processing time change when you add the bright spot detection?

Processing Time per Frame: 0.014111 seconds
Processing Time per Frame: 0.011123 seconds
Processing Time per Frame: 0.012158 seconds
Processing Time per Frame: 0.035148 seconds
Processing Time per Frame: 0.009239 seconds
Processing Time per Frame: 0.011433 seconds
Processing Time per Frame: 0.012107 seconds
Processing Time per Frame: 0.008553 seconds
Processing Time per Frame: 0.004290 seconds
Processing Time per Frame: 0.016893 seconds

The mean processing time per frame is 0.01351 seconds So it is approximately the same.

3. Is the processing time identical when you do not display the image?

Processing Time per Frame: 0.021795 seconds
Processing Time per Frame: 0.020392 seconds
Processing Time per Frame: 0.020389 seconds
Processing Time per Frame: 0.020990 seconds
Processing Time per Frame: 0.021437 seconds
Processing Time per Frame: 0.021623 seconds
Processing Time per Frame: 0.021343 seconds
Processing Time per Frame: 0.026160 seconds

Processing Time per Frame: 0.021535 seconds

Processing Time per Frame: 0.014725 seconds

The mean of the processing times is 0.02104 seconds. That is a little slower than displaying the image but still minimal change in processing time.

4. How does your for-loop implementation compare to the built-in function?

Processing Time per Frame: 0.479856 seconds

Processing Time per Frame: 0.475770 seconds

Processing Time per Frame: 0.464804 seconds

Processing Time per Frame: 0.604436 seconds

Processing Time per Frame: 0.484238 seconds

Processing Time per Frame: 0.472094 seconds

Processing Time per Frame: 0.464571 seconds

Processing Time per Frame: 0.469880 seconds

Processing Time per Frame: 0.462424 seconds

Processing Time per Frame: 0.538742 seconds

The mean is 0.4916815 seconds. So the frame rate is approximately 2 FPS while the built-in function's mean time (around 0.01546 seconds per frame) corresponds to approximately 65 FPS.

5. Moving your hand in front of the camera, estimate the latency between image capture and display. The latency for the built in function is ~0.03 seconds and the latency for the for loop implementation is ~1 second.
6. Figures 1 and 2 show screenshots of the video captured using the phone camera connected to the code. I did not notice any significant change in latency compared to the computer's built-in camera for the built-in function implementation. In the for-loop implementation the latency increased. The processing time per frame was approximately 0.6 seconds, and the FPS was most frequently around 2.10.



Figure 1: Video feed from the phone camera showing the detection of the brightest and reddest spots in the frame with the built in cv2.minMaxLoc() function.



Figure 2: The for-loop implementation of the function