

HÁSKÓLINN Í REYKJAVÍK



Assignment 1: OpenCV setup and point operations

T-869-COMP - Computer Vision

Student

Magnús Benedikt Magnússon

Teacher

Torfi Þórhallsson

November 28, 2023

- The processing time for one video frame or image.
 - The processing time for one video frame or image seems to be somewhere between 0.9-1.9 μ s, as can be seen below in Figure 1.

```
(.venv) magnusbenediktmagnusson@Magnuss-MacBook-Pro T-869-COMP % "/Users/magnusbenediktmagnusson/Library/CloudStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/.venv/bin/python" "/Users/magnusbenediktmagnusson/Library/CloudStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/assignment1.py"
Processing Time: 0.0009536743ms
Processing Time: 0.0019073486ms
Processing Time: 0.0009536743ms
Processing Time: 0.0011920929ms
Processing Time: 0.0011920929ms
Processing Time: 0.0009536743ms
Processing Time: 0.0009536743ms
Processing Time: 0.0011920929ms
Processing Time: 0.0019073486ms
Processing Time: 0.0009536743ms
Processing Time: 0.0009536743ms
Processing Time: 0.0011920929ms
Processing Time: 0.0009536743ms
Processing Time: 0.0011920929ms
Processing Time: 0.0009536743ms
Processing Time: 0.0009536743ms
Processing Time: 0.0009536743ms
```

Figure 1: Snapshot of a printout of the processing time for one video frame or image.

- How does the processing time change when you add the bright spot detection?
 - The processing time for one video frame or image jumped up a but by adding the bright spot detection, or around 0.3 - 0.9 ms as can be seen in Figure 2. Adding the red spot detection as well bumped the processing time up even further, up to about 5-10 ms as can be seen in Figure 3

```
(.venv) magnusbenediktmagnusson@Magnuss-MacBook-Pro T-869-COMP % "/Users/magnusbenediktmagnusson/Library/CloudStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/.venv/bin/python" "/Users/magnusbenediktmagnusson/Library/CloudStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/assignment1.py"
Processing Time: 0.4808902740ms
Processing Time: 0.7529258728ms
Processing Time: 0.5738735199ms
Processing Time: 0.6060600281ms
Processing Time: 0.4060268402ms
Processing Time: 0.4661083221ms
Processing Time: 0.8718967438ms
Processing Time: 0.4899501801ms
Processing Time: 0.4220008850ms
Processing Time: 0.6661415100ms
Processing Time: 0.5128383636ms
Processing Time: 0.3118515015ms
Processing Time: 0.4742145538ms
Processing Time: 0.7071495056ms
Processing Time: 0.5114078522ms
Processing Time: 0.7190704346ms
Processing Time: 0.3609657288ms
Processing Time: 0.3919601440ms
Processing Time: 0.9181499481ms
Processing Time: 0.3628730774ms
Processing Time: 0.4768371582ms
```

Figure 2: Snapshot of a printout of the processing time for one video frame or image after adding the bright spot detection.

```

• (.venv) magnusbenediktmagnusson@iot-research T-869-COMP % "/Users/magnusbenediktmagnusson/Library/CloudStorage/Goog
leDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/.venv/bin/python" "/Users/magnusbenediktmagnusson/Library/Clo
udStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/Assignment 1/assignment1.py"
Processing Time: 10.2851390839ms
Processing Time: 9.6418857574ms
Processing Time: 9.7739696503ms
Processing Time: 9.0138912201ms
Processing Time: 9.3266963959ms
Processing Time: 6.5290927887ms
Processing Time: 6.1218738556ms
Processing Time: 7.0099830627ms
Processing Time: 6.4609050751ms
Processing Time: 9.2189311981ms
Processing Time: 6.7119598389ms
Processing Time: 6.4871311188ms
Processing Time: 6.7090988159ms
Processing Time: 5.2740573883ms
Processing Time: 7.0929527283ms
Processing Time: 6.9861412048ms
Processing Time: 5.8610439301ms
Processing Time: 8.0718994141ms

```

Figure 3: Snapshot of a printout of the processing time for one video frame or image after adding the bright spot and reddest pixel detection.

- Is the processing time identical when you do not display the image?
 - Yes, the processing time seems to be identical when not displaying the image, as can be seen by comparing Figures 3 and 4.

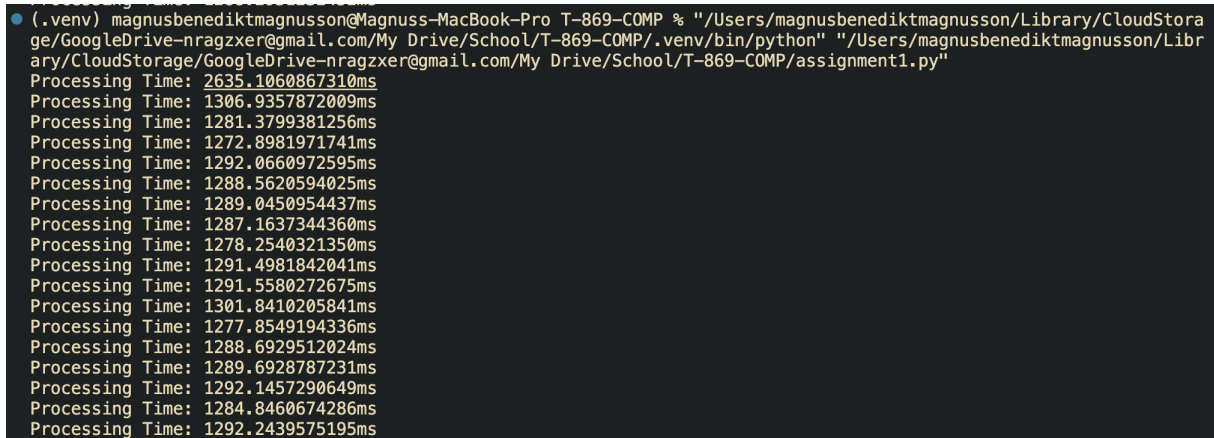
```

Ⓢ (.venv) magnusbenediktmagnusson@iot-research T-869-COMP % "/Users/magnusbenediktmagnusson/Library/CloudStorage/Goog
leDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/.venv/bin/python" "/Users/magnusbenediktmagnusson/Library/Clo
udStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/Assignment 1/assignment1.py"
Processing Time: 8.4168910980ms
Processing Time: 10.7948780060ms
Processing Time: 9.2799663544ms
Processing Time: 8.1269741058ms
Processing Time: 6.0582160950ms
Processing Time: 8.7618827820ms
Processing Time: 5.5410861969ms
Processing Time: 6.8008899689ms
Processing Time: 6.4158439636ms
Processing Time: 6.9179534912ms

```

Figure 4: Snapshot of a printout of the processing time for one video frame or image after adding the bright spot and reddest pixel detection while not displaying the image.

- How does your for-loop implementation compare to the built-in function?
 - The for-loop implementation completely kills the FPS (going to sub 1 levels) as well as shooting up the processing time by a huge margin, as can be seen in Figure 5. This is due the the fact that we are processing each pixel individually instead of relying on optimized algorithms and functions that extract only the needed information for computations etc.



```

(.venv) magnusbenediktmagnusson@Magnuss-MacBook-Pro T-869-COMP % "/Users/magnusbenediktmagnusson/Library/CloudStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/.venv/bin/python" "/Users/magnusbenediktmagnusson/Library/CloudStorage/GoogleDrive-nragzxer@gmail.com/My Drive/School/T-869-COMP/assignment1.py"
Processing Time: 2635.1060867310ms
Processing Time: 1306.9357872009ms
Processing Time: 1281.3799381256ms
Processing Time: 1272.8981971741ms
Processing Time: 1292.0660972595ms
Processing Time: 1288.5620594025ms
Processing Time: 1289.0450954437ms
Processing Time: 1287.1637344360ms
Processing Time: 1278.2540321350ms
Processing Time: 1291.4981842041ms
Processing Time: 1291.5580272675ms
Processing Time: 1301.8410205841ms
Processing Time: 1277.8549194336ms
Processing Time: 1288.6929512024ms
Processing Time: 1289.6928787231ms
Processing Time: 1292.1457290649ms
Processing Time: 1284.8460674286ms
Processing Time: 1292.2439575195ms

```

Figure 5: Snapshot of a printout of the processing time using the double for-loop implementation.

- Moving your hand in front of the camera, estimate the latency between image capture and display.
 - By using a stopwatch, I set a timer immediately after moving my hand in front of the camera, and stopped when I saw the hand appear moving on the displayed video. The latency seems to be about 2.5 seconds.
- Is the latency different when capturing from a mobile phone?
 - By using an online stopwatch, I set a timer immediately after moving my hand (holding the mouse so I go start the timer) in front of the camera, and stopped when I saw the hand appear moving on the displayed video. The latency seems to be about 5 seconds, so yes it's quite different. I also noticed that sometimes the image buffer seemed to skip the frames where my hand moved in front of the camera but that could be attributed to the fact that the frame rate was so slow due to the double for-loop implementation.
- Below is a link to the GitHub repo
 - https://github.com/magnusbenedikt/T-869-COMP_Computer-Vision/tree/main/Assignment%201