VIDEO PROCESSING

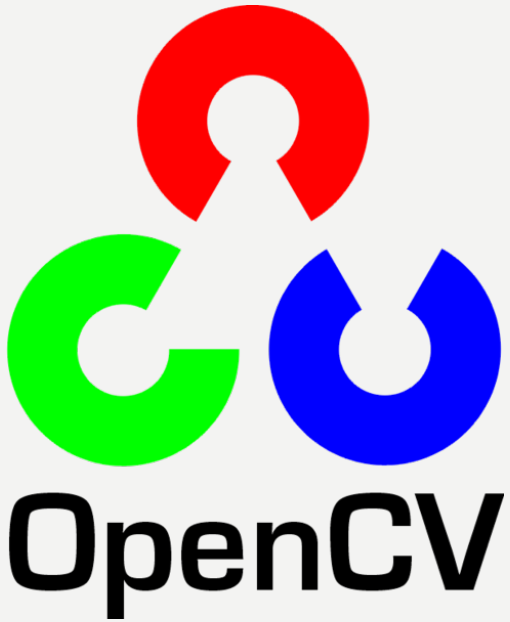
ACTIVITY 05
PHYSICS 301

KENNETH M. LEO

THE BASICS

In this activity, we explore the basics of video processing. I will be processing two type of video in this presentation: 1) live video and 2) recorded video.

For the live image acquisition. I will be using opencv in Python to do this. Actually, most (if not all) of the things I will be doing in this activity are all done using opencv.



So, the things I will be exploring are as follows:

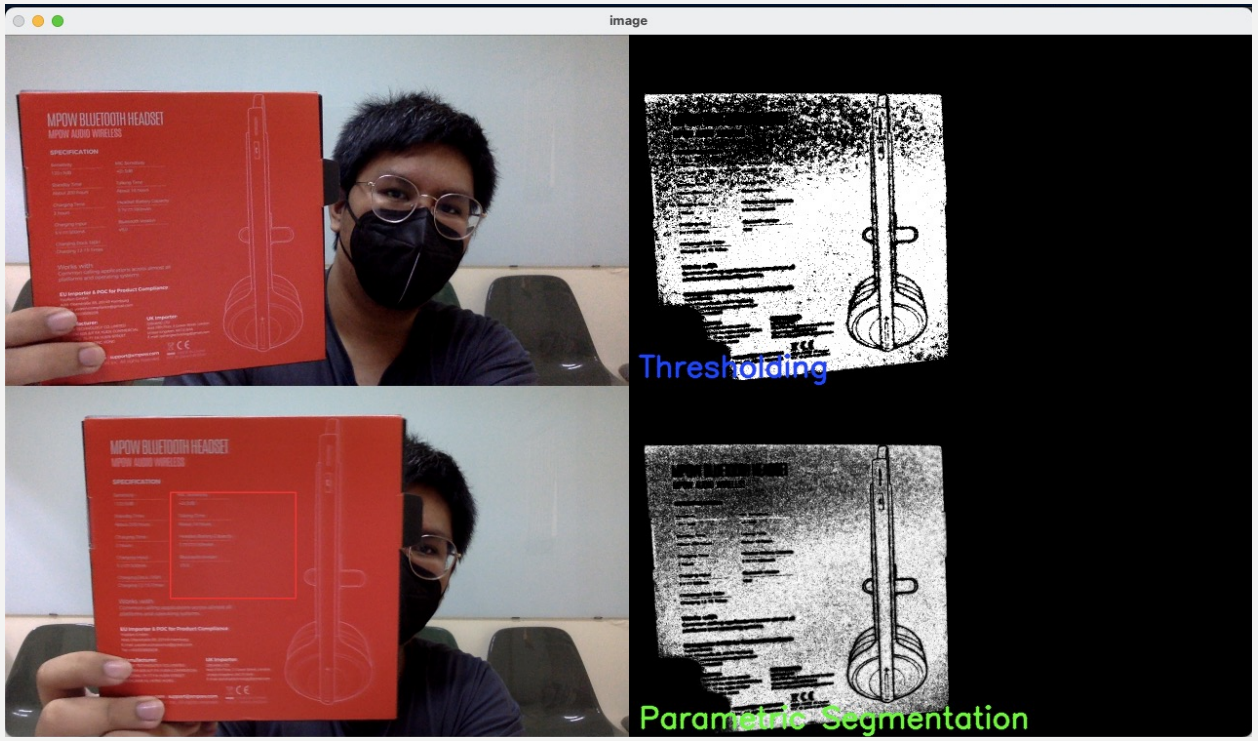
1. Live tracking colored object using color segmentation
2. Optical flow
3. Face detection
4. Others

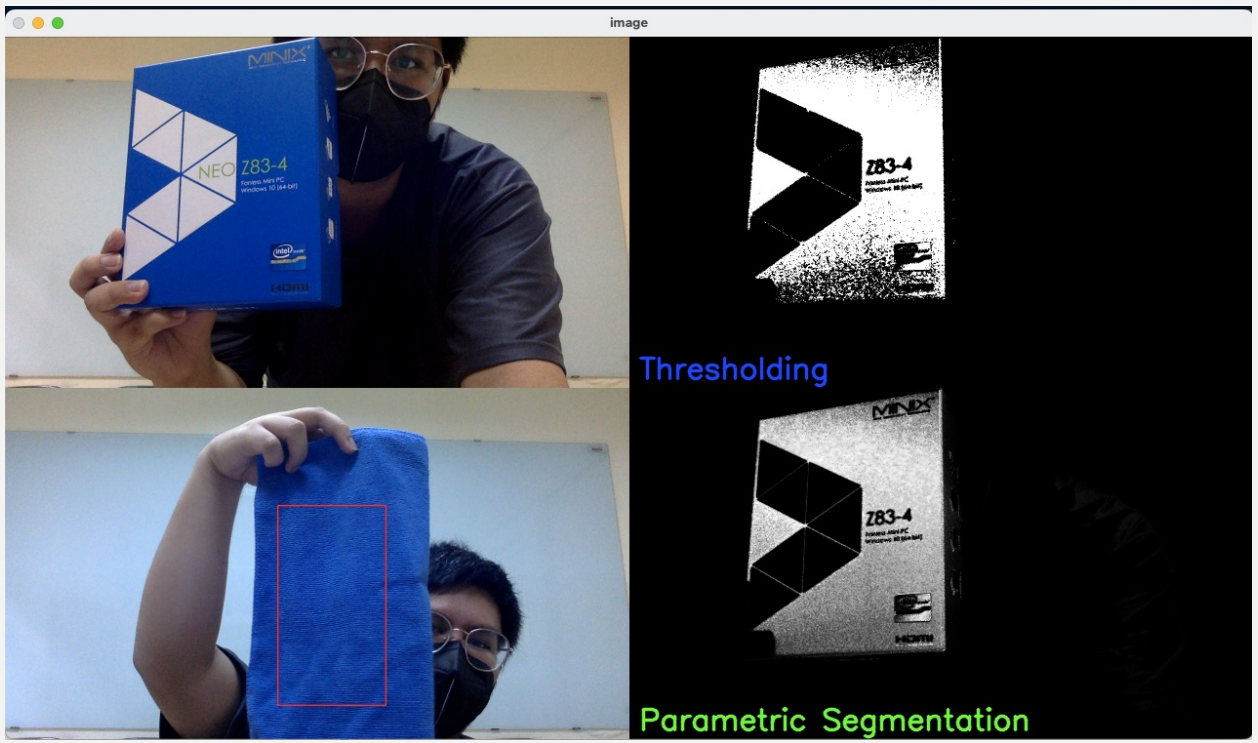
COLOR SEGMENTATION

For this part, I used my code from Activity 01 for segmentation by thresholding and parametric segmentation. I did not use the non-parametric segmentation in this case because that requires manually adjusting some parameters which defeats the purpose of live video color segmentation.

We see from the gif on the right that since we selected the black mask as our ROI, only the black colored regions (hair, mask, some parts of shirt) can be seen in both segmentation techniques.

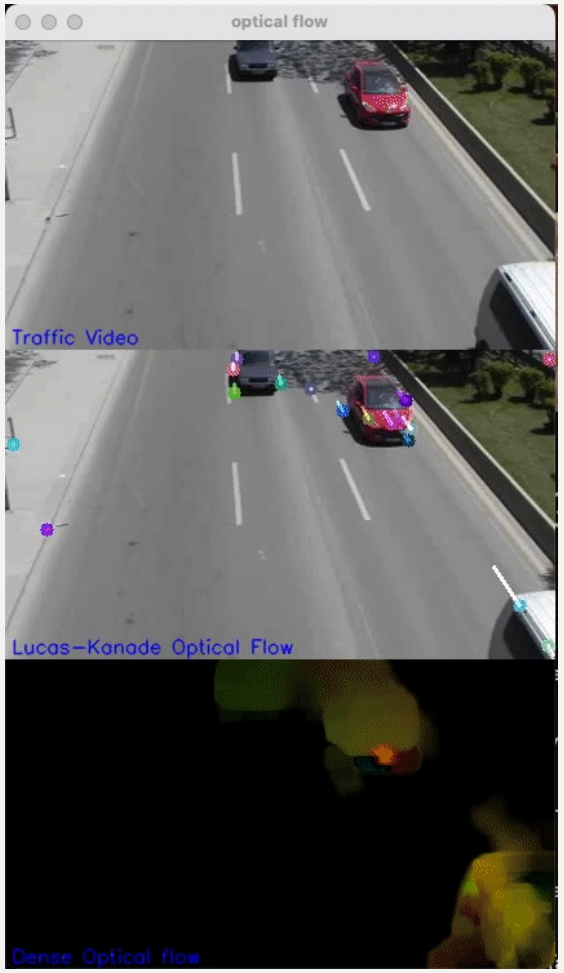
COLOR SEGMENTATION: OTHER SAMPLES





We see here that if we use ROI with distinct colors (different with the background), we can get better segmentation. For the figure on the right, we see that the object I used my ROI from and the object I was testing is different but the color segmentation is still successful because both images have the same color profile.

TRY: OPTICAL FLOW



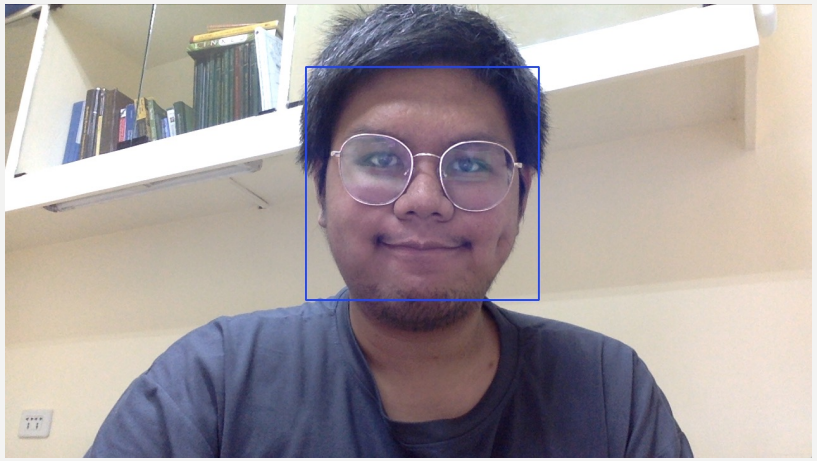
Optical flow, in its simplest form is the pattern of apparent motion of image objects between two consecutive frames. This motion can either be caused by the movement of your object or the camera.

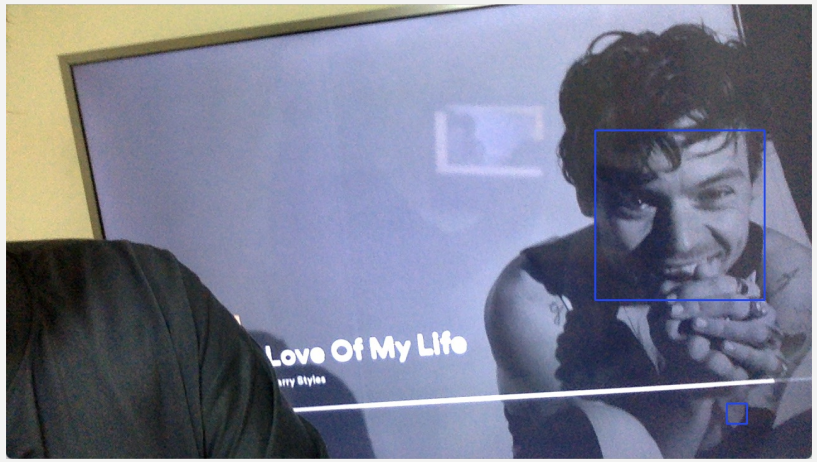
From this, we can infer the direction of your object (assuming camera is stable) and using additional algorithms, we can also estimate the speed of the object.

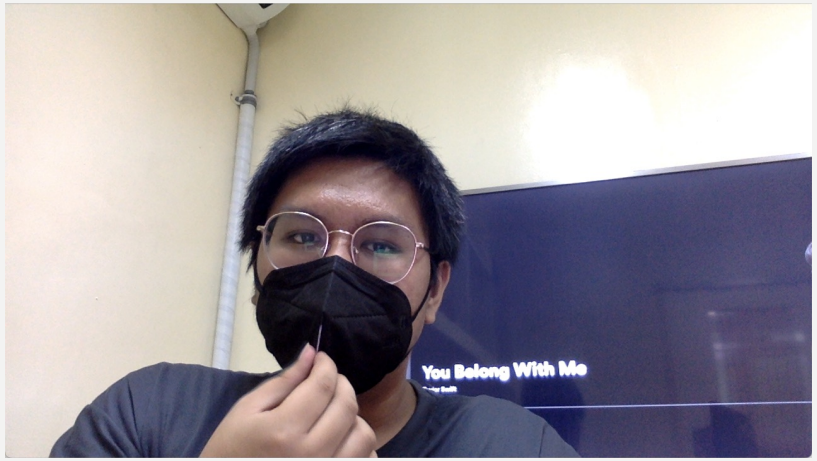
The figure on the left shows two different types of optical flow algorithm used: the Lucas-Kanade optical flow which tracks specific points on the video. The second algorithm is the Dense Optical Flow which computes optical flow of all points in the video (can run slower than the first algorithm.

TRY: FACE DETECTION

Next, I play with face detection in opencv. With this one, we first need a dataset of faces in order to make it work. Luckily, opencv has a pre-trained machine learning model for detecting faces called the "Haar cascades". In this classifier, we can also detect eyes, lower body, etc., but for demonstration purposes I will just be detecting faces.







We can see from the first two figures that the classifier can successfully detect faces, no matter if you're wearing glasses or the face is just taken from the screen. But sadly, we see that it fails to detect a face wearing face mask. This is probably because the dataset used for the model did not include faces wearing face masks.

SUMMARY PLUS REFERENCES

Summary

I was actually really happy about this experiment! I was able to learn about live video processing using opencv in Python. For the color segmentation, the code was purely written by me and that simple implementation of opencv shows how good this package actually is. It was also cool to try optical flow and face detection. I can see how optical flow can be useful especially when working with traffic networks/dataset.

References

[1] https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html#lucas-kanade
[2] https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81

Score:

Technical Correctness – 30
Quality of Presentation – 30
Reflection – 30
Ownership – 10

Total – 100/100