CSc 8830: Computer Vision

Assignment 3

Submission in Classroom:

Manage all your code in a github repo for each assignment. Provide a link to the repo in the PDF document. You can choose to program in either C/C++ or Python. Submit the script with clear commenting and ReadMe documentation on top of each script to execute the script.

Create a working demonstration of your application and record a screen-recording or a properly captured footage of the working system.

Upload the PDF document and video in the Google classroom submission. (copying the script in the document is not required; GitHub repo must be accessible)

For parts that require or ask for "solve by hand" or "show by example" methods:

convert your problem solving by hand into a digital format (typed or scanned only. You can use camera scanner apps) and embedded/appended into the final PDF documentation. Camera images of paper worksheets will NOT be accepted

ALL REFERENCES MUST BE CITED IN YOUR ASSIGNMENT REPORTS

Consider the object images that you had used for measurement of dimensions. Collect 10 different images (same object from different angles and distances). This set will be called a dataset in this assignment. Unless otherwise specified you will use this dataset for evaluation. **Note: All of your exercises from 1-4 must work real-time with visualization on a web application. You must have already built this from Module 1. Keep adding the features as you keep developing over the course.**

1. Find the gradient image for each of the images in your dataset: this will be a set of two images, one is the magnitude of gradient and other is the angle. Save both these as images for visualization. Then, find the Laplacian of Gaussian filtered version of your original images. Compare it with the gradient images.

2. Implement a simple algorithm based on your learning from the lecture videos for detecting keypoints that can be marked as an EDGE. Implement a simple algorithm based on your learning from the lecture videos for detecting keypoints that can be marked as a CORNER.

3. Implement a script to find the exact boundaries of the object of interest in your image. You can use the functions and libraries available in opencv. Deep learning and machine learning based code is not allowed for this problem.

4. Implement object segmentation (find boundary of an object) of a NON RECTANGULAR object. Stick ArUco markers anywhere on the boundary of the object and capture the image using your camera of interest from various distances and angles. You need to use at least 10 images for evaluation. You can use OpenCV's built-in ArUco marker detection or any reference you may find appropriate.

5. Compare your results from (4) with the SAM2 segmentation model.