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 for Part A. 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

For programming, you can choose to program in either MATLAB or Python

If MATLAB: Submit a MATLAB Live script (.mlx file) and also convert the .mlx file to PDF and append to PDF from Part A.

The MATLAB Live Script document must contain all the solutions, including graphs. The file must be saved as ".mlx" format. See here for live scripts:
https://www.mathworks.com/help/matlab/matlab_prog/create-live-scripts.html

If Python, you can submit it as a Jupyter notebook or the .py file. You should include clear commenting and ReadMe

1. Implement an application (must run on web or as an app on mobile device) using the stereo camera where it will recognize, track and estimate dimensions (at least 2D) of any object within 3m distance and inside field-of-view to the camera. You can use barcodes or text recognition tools for identification. However, the entire object must be tracked (not just the barcode or text). Machine/Deep learning tools are NOT allowed.

2. Use the DepthAI SDK or use ORB3-Visual SLAM (https://github.com/UZ-SLAMLab/ORB_SLAM3) to execute the scripts on your depth camera and run experiments in two different locations. Provide snapshots of your SLAM output and what limitations/corner cases do you observe.