CSc 4980/6980: Computer Vision Assignment 2

Submission in Classroom:

Camera images of paper worksheets will NOT be accepted

Python: submit a jupyter notebook and the .py files

associated

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-script s.html

Manage all your code in a github repo for each assignment. Provide a link to the repo in the documentation workspace (jupyter notebook or mlx file).

Create a working demonstration of your application and record a screen-recording or a properly captured footage of the working system. Upload the video in the Google classroom submission.

Hardware: Unless otherwise specified, CAMERA refers to the OAK-D Lite camera provided to you.

Software:

MATLAB: Either of the following will work: Use MATLAB R2018b or later version as installed in your machine (installation instructions already provided) **OR** Use MATLAB Online (https://www.mathworks.com/products/matlab-online.html).

For OAK-D you can implement your solutions in either Python or C/C++: https://docs.luxonis.com/en/latest/

1. Capture a 10 sec video footage using a camera of your choice. The footage should be taken with the camera in hand and you need to pan the camera slightly from left-right or right-left during the 10 sec duration. For all the images, operate at grayscale

a. Pick any image frame from the 10 sec video footage. Find the boundary of any object in the scene. You can pick regular shapes. You must show usage of Harris corner and Canny edge detection function.

b. Pick another image frame from the set which also has the same object in view. Find all corresponding points of the object under consideration between these two images. Find the homography matrix between the images.

2. Implement the image stitching application in MATLAB (not necessary to be real-time). Test your application for any FIVE of a set of 3 image-set available in the gsu_building_database. That is, your stitching application should stitch 3 images. You must test the performance of your application for FIVE such sets.

https://drive.google.com/drive/folders/1cgVYdrzn9yUpYYi14mgvNyQUv8Ym5gui?usp=sharing

3. Implement an application that will compute and display the INTEGRAL image feed along with the stereo and RGB feed. You **cannot** use a built-in function such as

“output = integral_image(input)”

4. Implement the image stitching, for at least 1 pair of images. Use SIFT features. If using Depth AI API this should function in real-time. You **can** use built-in libraries/tools provided by the DepthAI API.

If available, you can also simply call any built-in function “image_stitch(image1, image1)”. **However,** in that case, you need to show a 180 or 360degree panoramic output.

5. Repeat (4) using ORB features.

You can make assumptions as necessary, however, justify them in your answers/description.