Programming Assignment-2 on course CSL442_IVP_S22

Instructions:

1. Solve any five questions out of six. Each question carries 3 marks.
2. Required two items for submission:
    - 2.1. Solutions Jupiter notebook / MATLAB file.
    - 2.2. Recorded video of your three programs explanation and output demos. Try to keep the video duration minimum (preferably 10-15min).
3. All the submissions should be submitted only through Gradescope portal. Assignment due date is 16-04-2022 before 5:00PM. Late submission is accepted till 17.04.2022 before 5:00PM with 2 marks penalty from the acquired marks.
4. Copying assignment solutions from others is strongly discouraged and will violate the academic code of conduct. If such acts are discovered, the entire score will be set to zero.

Assignment Questions:

Q1. Develop a function to implement Harris corners detection algorithm. Use the function to detect corner points in the given **corner_test.png** image. Compare your function implementation result with built in cv2.cornerHarris() / detectHarrisFeatures() function result to test your code.

Q2. Implement your own program to detect lines in a given image using the given hough transform algorithm. Use **roadlane.jpg** to test your code. Compare your code output with inbuilt hough line model fitting function.

Algorithm:

1. Decide on the range of ρ and θ. Often, the range of θ is [0, 180] degrees and ρ is [-d, d] where d is the length of the edge image’s diagonal. It is important to quantize the range of ρ and θ meaning there should be a finite number of possible values.
2. Create a 2D array called the accumulator representing the Hough Space with dimension (num_rhos, num_thetas) and initialize all its values to zero.
3. Perform edge detection on the original image. This can be done with any edge detection algorithm of your choice.
4. For every pixel on the edge image, check whether the pixel is an edge pixel. If it is an edge pixel, loop through all possible values of θ, calculate the corresponding ρ, find the θ and ρ index in the accumulator, and increment the accumulator base on those index pairs.
5. Loop through all the values in the accumulator. If the value is larger than a certain threshold, get the ρ and θ index, get the value of ρ and θ from the index pair which can then be converted back to the form of y = ax + b.

Q3. Write a program to perform robust image matching using RANSAC algorithm and Harris Corner features. Use **FM_img1.jpg** and **FM_img2.jpg** to test the program.

Q4. Write a program to perform image stitching for two images. Use SIFT features (use builtin library function) and homography based Feature Matching technique. Use **scene1.jpg** and **scene2.jpg** to test the program. NOTE: install "pip install opencv-contrib-python" to use builtin SIFT descriptor.

Note: CV2 SIFT function: `cv2.xfeatures2d.SIFT_create()`
Matlab SIFT function: `detectSIFTFeatures()`

Q5. Write a program to implement Otsu’s algorithm for global threshold-based image segmentation. Use **Boat_Otsu.png** image to test your code.

Q6. Implement a program to detect moving vehicles by using median differencing background subtraction technique and mention your observations on the result. Use **traffic.3gp** video clip to test your code.