ENPM 673, Robotics Perception
Project 2: Visual Odometry.
Due on: **Wednesday** – **March 28**, 2018

1 Visual Odometry - 100Pts

Visual Odometry is a crucial concept in Robotics Perception for estimating the trajectory of the Robot (camera to be precise). The concepts involved in Visual Odometry are quite the same for SLAM which needless to say is an integral part of Perception.

2 Details

You are given frames from a driving car, scripts to extract intrinsic parameters and the output should be a plot for the trajectory of the camera - DATASET.

2.1 Pipeline

- The input images are in Bayer format from which you can recover the color images using the demosaic function with `GBRG` alignment.
- Extract the camera parameters using `ReadCameraModel.m` as follows:
  `[fx, fy, cx, cy, G_camera_image, LUT] = ReadCameraModel('./stereo/centre', './model');` // *top line of* `stereo_narrow_left.txt` *gives fx, fy, cx, cy*
- Undistort the current frame and next frame using `UndistortImage.m`.
- Find point correspondences using ANY keypoint operator of your choice.
- Estimate the Fundamental matrix using these point correspondences.
- Recover the Rotation and Translation matrices from the Fundamental matrix.
- Plot the position of the camera center (for each frame) based on the rotation and translation parameters between successive frames.

You should implement the functions to **estimate Essential Matrix** and also to **recover rotation/translation matrices** and **NOT** use MATLAB's Computer Vision Toolbox or any third party code.

3 Extra Credit - upto 40 Pts

Compare your result against the rotation/translation parameters recovered using `relativeCameraPose`, `estimateFundamentalMatrix` from MATLAB's Computer Vision Toolbox. You can plot both the trajectories and report the accumulated drift in trajectory from your implementation versus the one where you use MATLAB's in-built functions.

4 Submission Guidelines

Your submission **SHOULD be a ZIP folder** (no other file extensions) with the nam-ing convention `YourDirectoryID proj2.zip` on to ELMS/Canvas. Additionally follow the guidelines given below:

1. You will have a parent directory `P2_Submission`.
2. Under `P2_Submission/VisualOdometry` you will have three sub-folders `code`, `input` and `output`.
3. **You SHOULD also submit a report** (`Report.pdf`) under `P2_Submission/VisualOdometry` folder.

5 Useful Resources

- Check out relativeCameraPose from MATLAB's Computer Vision Toolbox.
- Watch this excellent video lecture to understant *How to Estimate Essential Matrix*

6 Acknowledgement

Dataset used is by courtesy of Oxford's Robotics Institute.

7 Collaboration Policy

You are allowed to discuss the ideas with fellow students, but you need to give credits in the report. But the code you turn-in should be your own and if you **DO USE** (try not to and it is not permitted) other external codes - do cite them and you might get partial credits. For other honor code refer to the University of Maryland Honor Pledge.

DISCLAIMER: *You should take the effort to search online for any help regarding function usage, however, any concept related queries can be discussed in TA Office Hours.*