CSci 4270 and 6270
Computational Vision,
Spring Semester, 2022
Homework 7
Due: Wednesday, April 27 at 11:59pm

Overview

This is the final homework of the semester, and it is worth **100 points** toward your overall homework grade. You may use up to two late days on this assignment. If you hand in the assignment with beyond your available late days you will be charge 7.5 points for each extra late day. **No homework will be accepted after 11:59:59 pm on Friday, April 29..**

We focus on an important problem for self-driving cars that you already have all of the tools to solve. This is the detection of independently moving objects. For this assignment, do not expect perfect results — you will not get them. Instead, please do your best in the time you have available and implement a solution to each stage of the algorithm.

Problem Description

Consider two images, I_0 and I_1, taken by a single camera while it is either stationary or moving through a scene. Importantly, other objects may be moving as well. The technical problems you must solve are:

1. estimate the image motion vectors at a sparse set of points
2. determine whether or not the camera is moving through estimating the "focus of expansion" induced by the camera's motion
3. determine which points are moving independent of the camera's movement
4. cluster such points into coherent objects

Looking in more detail:

- You may choose the points where you estimate motion based on the Harris criteria, the KLT criteria, or any other method you wish (i.e. SIFT or ORB). You may use OpenCV algorithms to do this, but you already have the tools to do this going back to early in the semester.
- Estimate the image motion at these points. You may use the algorithm discussed in the Lecture 22 (and 23) notes, but you may also use descriptor matching. Once again there is OpenCV code to do this (`calcOpticalFlowPyrLK`), but you can also "roll your own". Regardless of what you do, be sure you can handle non-trivial image motion distances.
- Determine the motion of the camera, if any, by estimating the focus of expansions (FOE) using the algorithm sketched in the end of the lecture notes on motion. To do this, you must:
  1. At each chosen point where motion is estimated, convert the point location and motion vector to a line. Call these the "motion lines"

2. Use RANSAC to estimate the FOE. This is the point within a distance tolerance of the largest number of motion lines. Remember, this algorithm is essentially the same as our original RANSAC algorithm with the roles of points and lines reversed. (Again, see the lecture notes and recording.)
3. If there are enough motion lines within the distance tolerance of the FOE then the camera should be considered to be moving.

- Once the FOE is found, motion lines that do not come close to this point correspond either to errors in motion estimation or independently moving objects. Similarly, if the camera is not moving, all non-trivial estimated motion vectors correspond to errors or independently moving objects. In either case, see if you can figure out a way to identify the points from independently moving objects and and group them, throwing out groups that are too small. Our class discussion on segmentation and clustering will help here. One challenge to doing so is that the motion vectors of points with small apparent motions are fairly unstable — so that the orientations of the lines you generate can have a great deal of error.

For each pair of input images, I_0 and I_1 please generate two output images:

- An image with the points, the motion vectors, and the focus of expansion drawn over top of image I_1. If there is no motion of the camera, show no focus of expansion, but make sure the fact that there is no motion is documented and justified in your diagnostic output (see below).
- A second image, similar to the first, but this one showing the independently moving objects you detected. For each independently moving object, select a random color and use this to color the points and motion vectors determined to be part of that cluster and to draw a bounding box around the points.

In addition to the image output, please show brief but clear diagnostic output from your program.

What to Submit

Submit just two documents. The first is your python program. The second is a description of your algorithm and results. This should (a) explain your design decisions and any trade-offs involved, (b) demonstrate your results, and (c) evaluate the strengths and weaknesses of your algorithm as highlighted by these results. How many results will be needed? My answer is that this should be enough to justify each of the claims in your write-up without being redundant. You should clearly illustrate how well each step works and when and why might it fail (or at least produce lower quality results).

Evaluation (for Part 2)

We will use the following rubric in grading your submission, so be sure your submission highlights them

- (12 points) Selection of points to estimate motion
- (12 points) Estimation of apparent motion
- (20 points) Estimation of the focus of expansion, or deciding that the camera did not move.
- (12 points) Clustering of independently moving objects

- (10 points) Quality of code
- (12 points) Clarity of explanation
- (12 points) Highlight of strengths and weaknesses
- (10 points) Selection of illustrative examples.