CSci 4270 and 6270
Computational Vision,
Fall Semester, 2019
Homework 7
Due: Thursday, December 12, at 5 pm

Overview

This is the final homework of the semester, and it is worth **125 points** toward your overall homework grade. The assignment is divided into two parts. Part 1, worth 25 points, is a short writing assignment about social and ethical issues of computer vision applications. No late days may be used for Part 1, meaning that no write-ups submitted after 5 pm on December 12 will be accepted without a formal excuse. Part 2, worth 100 points, is programming assignment. You may use at most a single late day for the first part — if you have one remaining.

Part 1: Should You Build It?

Our class discussion (Monday 12/2) of ethical and social issues of computer vision focuses on face recognition and self-driving vehicles. For this assignment, pick a **different** application of computer vision and answer all of the following questions:

1. What application did you choose, and what are some technical problems associated with it?
2. What else might this technology be used for?
3. How would someone collect training data for this application?
4. What social and ethical issues might arise with this data collection?
5. What are some positive and negative consequences of building and deploying this application?
6. Based on current technology should we build this application? Should we restrict its use if we do build it? Why and how?

This problem requires a different approach to the homework we have done so far. Specifically, you may have to do some background investigation on a topic we have not covered. You are being asked to think critically about the topic you choose, and so there is not necessarily one "correct" answer.

Please provide short, crisp answers to these questions, with the total write-up occupying less than one-page, single-sided, 11 point font. A shorter write-up is fine if you provide clear answers. You may (and probably will) use resources you find on-line, but please be sure to cite them.

You will be graded on the quality of your analysis and the clarity of your write-up. You may choose any topic you wish within the field of computer vision (except facial recognition and self-driving vehicles). In your writing, please use complete sentences, and run it through a spell-checker before submission.

Part 2: Independent Motion

We have not had a lot of time to deeply investigate optical flow, motion estimation and stereo. Therefore 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 anywhere near. 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
3. if it is moving, find the "focus of expansion" induced by the camera's motion
4. determine which points are moving independent of the camera's movement
5. 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 19 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 if the camera is moving. For this you need to assume that the significant majority of the points in the image are not parts of other moving objects. Develop **simple** criteria to detect this.
- If the camera is moving, estimate the image position of the focus of expansion. If you consider each of the sparse image points and its motion vector as a line, then all correctly-estimated lines for stationary points in the scene (i.e. all motion vectors induced solely by camera motion) will intersect at a single point. For simplicity, we'll assume the camera is roughly pointing in the direction of motion, so the focus of expansion is in the image, or perhaps just outside of it. You should be able to adapt RANSAC to estimate this point, including a least-squares estimate of the point's position once all "inliers" are found.
- Once the focus of expansion point is found, motion vector 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 points in groups that are too small. 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.