

CAP 6415: Computer Vision Assignment 2

Due Date: Sunday 02/09/2020 11:59pm

Submission Type: Upload onto Canvas

Problem Statement: In this assignment you have to implement a point feature tracker that detects point features in the first frame of a video and tracks them through the video. Instead of finding feature points independently in multiple images and then matching them, find features in the first image of a video or image sequence and then re-locate the corresponding points in the next frames using either search and gradient descent (Shi and Tomasi 1994) or learned feature detectors (Lepetit, Pilet, and Fua 2006; Fossati, Dimitrijevic, Lepetit et al. 2007). When the number of tracked points drops below a threshold or new regions in the image become visible, find additional points to track.

(Extra Credit - 50% of earned grades) – Augment the tracker you have implemented with Kalman tracking.

As output, overlay the entire tracks on the first frame of the video and show them as one image per video in your report.

You may use any video data available at <https://motchallenge.net/> for your assignment.

Submission Requirements:

Please upload a ZIP file containing the following files:

1. All your code files, including any helper files/dependencies.
2. A README file detailing how to run your code along with any compilation instructions.
3. A 2-Page technical report containing the following sections:
 - a. A short description of the algorithm
 - b. A description of any code/algorithms that were used/re-used by you for your implementation.
 - c. A few examples of results from your implementations, comparison with the original implementation (if needed).
 - d. A general discussion of lessons learned based on your experiments with the algorithm. E.g. What did you struggle with, issues faced while implementing the code, scopes for and/or proposed improvements, etc.
4. Your code will be tested on different test inputs and graded based on the progress of your approach on these test inputs. A demo session will be scheduled for evaluating your implementation.

Grading:

Each assignment will be graded out of 100:

Code (out of 30) quality of coding, readability, understandability (comments, variable names, etc.)

Report (out of 40), point 3a, 3b, 3c, and 3d above

Demo and Results (out of 30) Compilations, execution, correctness of results on test cases,

Solutions to your **programming assignments** have to be self-sufficient and **not dependent on other computer vision code, such as OpenCV or Matlab vision package**. You may use packages for display graphics or mathematics packages, such as for linear algebra (numpy, for example, matlab (but not computer vision module)) or graphs or optimization.

All reuse of code has to be clearly acknowledged in the source code, any README files, and also in the report. Failure to do so will be considered plagiarism.