EECS 442 – Computer Vision

University of Michigan, Electrical Engineering and Computer Science

Syllabus for Winter 2019

Administrivia

Instructor: Prof. David Fouhey (`fouhey@umich.edu`)

GSI: Richard Higgins (`relh`)

IAs: Linyi Jin (`jinlinyi`), Shengyi Qian (`syqian`), Yi Wen (`wennyi`)

Canvas+Piazza link: https://umich.instructure.com/courses/269680

Meeting Times, Location:
Lectures:
Tuesday/Thursday 10:30AM – 12:00 Noon, 1571 GGBL
Discussion/Recitation:
Monday 12:30PM - 1:30PM, 2246 SRB
Wednesday 5PM – 6PM, 1571 GGBL
Office Hours:
(BBB Learning Center) Tuesday 3:00-4:30pm
(BBB Learning Center) Thursday 2:30-4:00pm
(3777 BBB) Friday 10:30AM – 12:00AM

Course Information: This course will use Canvas and Piazza for announcements, updates, assignment releases, and disussions. Please sign up. If you have questions about logistics, the material, or assignments, please use piazza since it helps the staff respond to each question once, unless you have privacy concerns in which case feel free to ask the course staff directly.

Description

This is an introduction to computer vision. Topics include: camera models, multi-view geometry, reconstruction, some low-level image processing, and high-level vision problems like object and scene recognition. A rough outline of *topics* and **number** of classes spent on them is: *introduction* (**1**), *image formation/lighting/projective geometry* (**3**), *image processing / filtering* (**4**), *linear models + basic optimization* (**3**), *neural networks* (**3**), *motion and flow* (**4**), *geometry* (**5**), *applications* (**3**).

Prequisites

While some of you may have some experience in image processing, machine learning, or computer vision, none of these are required. However, you are expected to have a basic level of programming, computer science, and mathematics in order to succeed and derive satisfaction from this course.
The course will be taught assuming (i.e., we will not cover) that you have:

1. computer science knowledge at the level of EECS 281 (data structures) and corresponding programming ability;
2. the ability to program in Python, or if not, the ability to learn to program in a new language quickly.

It would be **extremely** helpful to know the following topics (i.e., we will provide a brief refresher when introducing it, but will not explain the details):

1. *Programming:* Some knowledge of numpy would be useful.
2. Basic knowledge of basic linear algebra, calculus. Learning and geometry depend on both.

   If you are unfamiliar with linear algebra or basic calculus, please consider taking both: without these tools, you will likely be lost for large parts of the course. If you are rusty, do not worry but do make an effort to refresh your memory of both at the start of the course.

   In particular, in addition to knowing basic matrix operations, it would be good to know least-squares, Eigen- and singular-value decompositions, some notion of continuous optimization, partial derivatives, the chain rule.

Textbooks

There is no textbook. However, you may find these two books useful:

1. R. Szeliski. *Computer Vision: Algorithms and Applications.* Free copy here: `http://szeliski.org/Book/`
2. D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach (2nd Edition).* `http://luthuli.cs.uiuc.edu/~daf/CV2E-site/cv2eindex.html`

Coursework and Evaluation

- *Homework (50%):* There will be five programming assignments over the semester, each worth 10%. All will be in Python. You are encouraged to discuss the homework with your classmates. However, your code and writeups must be your own. In particular, you should not copy code from anywhere and over-the-shoulder debugging of others' code is not allowed.

  Late homework will be penalized at a rate of 1% per hour, rounding to the nearest hour, ties away from zero. For example, if the homework is due Monday at 11:59:59pm, homework submitted until Tuesday 12:30am will be accepted with no penalty. After 12:30, there is a 1% penalty. If you submit it Wednesday 6:30am, there is a 30% penalty. Because unexpected things happen, you will have three late days over the semester that you can use on homework. **Given this lenient policy (as opposed to late homework is 0), please do not ask course staff to wave late deductions or give extra late days, except for circumstances that merit it.**

- *Project (35%):* There will be one final project where groups of two will work together over the second half of the semester. With the instructor's permission, you can form groups of three or more.

  We will have a piazza thread for helping people find project partners.

- *Mid-term Exam (15%):* There will be one mid-term exam.

Project

This is an opportunity to explore a topic in depth and should involve substantial work. This can be in implementation (e.g., implementing an existing algorithm), applications (e.g., applying computer vision to an existing problem), or research (e.g., trying something new in computer vision). Your project should amount to three homeworks' worth of work per-person.

If you cannot find a partner, there will be a piazza discussion for finding project partners. All written work should be in CVPR format `.zip` file here. **Overall, please remember that we do not see your hard work, we only see the products you deliver.**

Proposal (2 pages): The proposal should aim to explain what the problem is, why it's feasible to solve in the given timeline, and how you plan to achieve it. **While this is ungraded, this is important to get right.** In particular, your project proposal should list:

1. What specific problem are you trying to solve? Are you appling computer vision, trying to extend computer vision, or implementing something? All are reasonable options. But please make it clear which you are trying to accomplish.

2. How are you going to solve the problem? You don't need to have everything already planned out, but you should have a clear idea. What parts are likely to work, and how can you still get work done if your plan fails.

3. What do you need that you don't currently have access to? This could be data, knowledge of certain machine learning techniques, or particular code.

4. What do you expect as an outcome and how will you identify and quantify success?

Progress Report (2 pages, 5%): What progress have you made? **Think of this as an opportunity to get feedback**. Please be specific. in particular, your report should list:

1. What has been accomplished **concretely**? This may be finding a good formulation for your problem, scouring the Internet for data, building a proof of concept, or implementing a core part of the algorithm.

2. What is left to do? What hurdles do you anticipate and how will you overcome them? What will you do if you overcome them in the given time? Please be realistic.

Final Project Report (8 pages, 20%) and Code (5%): Please write up your project in the style of a CVPR paper (see samples here). Please remember to claim credit for everything you did and be clear what part of the project is yours and what part comes from others (e.g., did the data come from somewhere else? the initial implementation that you improved? a 3rd party library?). Your final report should list:

1. The problem that you are trying to solve. Describe it concretely and explain an intuition about why it should be solvable.

2. Your approach. Please include diagrams – math is useful once the high-level idea is established, but diagrams will help us more quickly get up to speed.

3. Your results. Please explain not only the results but also *why* you ran the experiments you did. What do the experiments actually test? Please include both quantitative results (i.e., measuring numerically what's happening) as well as qualitative results (i.e., describing in words and images what's happening).

4. Details about your implementation. If you are building on the code of others, please indicate what is yours and what was others' work.

Your code should be a `.zip` file and should be reasonably well documented and make clear what is yours and what is others' code.

Poster (5%): There will be a poster presentation. Please include the main idea, main method, and main results. Be sure to include visual results and diagrams, and refrain from large blocks of text. The best posters simultaneously: (a) let the presenter communicate the main result of the work to a reasonably attentive and educated person in about 2 minutes; (b) let the presenter explain more details of the project and discuss results and similar work for about 5 minutes; (c) enable a person who is not interacting with the presnter figure out the main idea of the work (but perhaps not all the details – that's what the paper is for).

General Remarks

1. *Doing well in class:* You are highly encouraged to start assignments and projects early. Computer vision code has a nasty tendency of working reasonably well (but not correctly!) even if is wrong, being fairly difficult to debug, and can also can take a long time to run in order to debug. It is a good idea to bake in time for a few screwups. Late homework and assignments will not be accepted.

2. *Classroom etiquette:* Above all, please avoid being disruptive in class. (a) Please don't eat. (b) Please don't talk while the lecture or recitation is in progress. (c) If you use a computer in class, please use it only for taking notes and not for doing work for other classes or browsing the internet.

3. *Accommodation:* If you think that you need an accommodation for a learning disability, please let me know. We will work with the Office of Service for Students with Disabilities (https://ssd.umich.edu/ to make proper accommodations.

4. *Counseling Center:* The Counseling Center staff (https://caps.umich.edu/) are trained to help you deal with a wide range of issues, such as how to deal with exam-related stress and other academic and non-academic issues. Services are free and confidential and do not impact student records.

Academic Integrity

All students in the class are: (a) presumed to be decent and honorable; (b) bound by the College of Engineering Honor Code; and (c) expected to read, understand, and follow the honor code. Information about this can be found here: https://elc.engin.umich.edu/honor-council/.

You are highly encouraged to discuss the course materials, assignments, and projects. However, **programming assignments must be done individually (or in a team for projects)**. In particular, downloading or copying code from anywhere/over-the-shoulder coding/sitting at the debugger for others is strictly prohibited. Googling for solutions is also prohibited. If you are not sure whether something is permitted, please ask the course staff. Any instance of cheating will be reported to the honor council and may result in an F for the entire course.