EECS 442 – Computer Vision

University of Michigan, Electrical Engineering and Computer Science

Syllabus for Fall 2019

Administrivia

Instructor: David Fouhey (fouhey)

GSI: Mohamed El Banani (mbanani)

IAs: Zhengyuan Dong (dongzy), Shengyu Feng (shengyuf), Qichen Fu (fuqichen)

Canvas+Piazza link: TBD

Meeting Times, Location:
Lectures:
Monday/Wednesday 5:30PM – 7:00PM , 1571 GGBL
Discussion/Recitation:
Thursday 4:00PM – 5:00PM, 1018 DOW
Friday 10:30AM – 11:30AM, 1200 EECS
Friday 12:30PM – 1:30PM, 1012 FXB
Office Hours:
Tuesday 10:30AM - Noon, (Fouhey, BBB 3777)
Tuesday 3:00PM - 4:30PM, (Dong, TBD)
Wednesday 9:00AM - 11:00AM, (Fu, TBD)
Wednesday 3:00PM - 4:30PM, (El Banani, TBD)
Thursday 1:00PM - 2:30PM, (Feng, TBD)

Course Information: This course will use Piazza for announcements, updates, assignment releases, and discussions. **Please sign up and read what is posted**. The course will use gradescope for turning in assignments, and canvas for turning in code.

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* (**0.25**), *image formation/lighting/projective geometry* (**3**), *practical linear algebra* (**2**), *image processing / descriptors* (**3**), *image warping* (**2**), *linear models + basic optimization* (**2**), *neural networks* (**3**), *applications of neural networks* (**2**) *motion and flow* (**2**), *single-view geometry* (**2**) *multi-view geometry* (**3**), *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. *Linear algebra and calculus:* Computer vision is basically linear algebra + data. You will need knowledge of both linear algebra and calculus. 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, partial derivatives, the chain rule.

   I will include math refeshers. This is, however, meant to remind people of the linear algebra they forgot and to plug gaps that just aren't covered. However, this is a refresher course, not a first course. It is also not meant to be the only refresher if you are rusty with linear algebra.

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 **(3)** T. Hastie, R. Tibshirani, J. Friedman. *Elements of Statistical Learning.* Free copy here: https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf

Work and Evaluation

Your grade will be based on:

- *Practicals Mastery (5%):* At the beginning of the semester, we'll release an assignment that will teach you practical skills that you need to succeed. You are free to submit this as many times as you want until you get a 100. This aims to serve as a forcing function to ensure that everyone is caught up.

- *Homework (60%):* There will be six programming assignments over the semester, each worth 10%. All will be in Python. You are encouraged to discuss the homework with your classmates and read widely to solve the problems. However, your code and writeups must be your own.

- *Project (35%):* There will be one final project where groups of 3-5 will work together over the second half of the semester. We will have a piazza thread for helping people find project partners.

Interaction

- **Lecture, Discussion Section:** You are expected to attend lecture and discussion section. I do not take attendance in either because I believe you are adults and you are free to do what you want. While I record lectures as a convenience for students, digesting three of them in a sitting at $1.5\times$ speed is probably not a good strategy.

- **Office Hours:** You are highly encouraged to go to office hours. Office hours are more effective if you come prepared with what you have tried and visualizations of what works and doesn't work.

- **Study Groups:** You are highly encouraged to form study groups and there will be a piazza post for doing this.

- **Piazza:** This class uses piazza. You are expected to be signed up. If you have questions about logistics, the material, or assignments, please use piazza since it helps the staff respond to each question once. If you have privacy concerns, you are of course free to ask the course staff directly.

  The course staff will monitor piazza regularly but not necessarily respond immediately at all hours. Do not share answers or code on piazza.

Homework Policies

- **Formatting and Submission Policy:** We will not accept handwritten homework in any shape or form. You do not have to type stuff up in latex. You do not need to make things pretty.

  Type it up in any format and submit a pdf to gradescope, marking the document up for where the questions are on gradescope. Stuff that doesn't follow this simple formatting requirement will get a 0 that you can fix by submitting it in the right format.

- **Collaboration/Honor Code Policy:**

  This is a class with a lot more freedom and self-teaching compared to intro-sequence courses. **You are strongly encouraged to work on the homework and projects in teams.** My role is to provide you guidance and background knowledge, but the learning comes from you engaging with the material. It's often useful to have peers along this journey.

  - **With Students:** You should *never* know the specific implementation details of anyone else's homework or see their code. So working in teams and dispensing general advice about output or strategies is great and highly encouraged (e.g., "oh if the image is overly dark where it overlaps, you're doing averaging wrong"). But pair-programming is forbidden, as is sitting next to someone and debugging their code.

    We believe that *the overwhelming majority* of UM students are ethical and honorable. However, to change incentives for those who are less honorable, we will run data mining software periodically on the submitted homework.

  - **Consulting Outside Material:** In this class, you're going to get some fairly vague specifications, potentially at the level of turn that 40-word description on a slide into a functioning program. You can, and should, turn to other documentation (the textbooks I suggest, other professors' lecture notes, other professors' slides, documentation from various libraries)

    What you may not do is read a bunch of actual code (pseudocode is fine however). If you come across code, close the window, but don't worry about it.

  - **Things you should never worry about** (some adapted from here): reading documentation of publicly available libraries; clarifying ambiguities and mistakes in assignments, slides, handouts, or textbooks, or documentation; discussing the general material; helping with specific class-independent stuff like cryptic numpy errors; discussing the assignments to better understand what's expected and general solution strategies; discussing the starter code; discussing general strategies for writing and debugging code.

  - **Bonuses for actually reading:** I'm awarding 50 points of extra credit added to the final grade, to be divided (rounding up) among the people who fill out the form here (click it) and promise they have read the syllabus. This promise is important. Also – there may be another bonus somewhere in the syllabus.

  - **If you're panicking and think copying will solve your issues:** Don't panic. We've tried to structure the incentives so that doing the right thing is rewarded. Submit it late (that's why we have late days and a sliding late-penalty). Submit it half-done (that's why we have partial credit). In the absolute worst case, don't submit it, and take a 0. In a year or two, you will not be worried about doing poorly on any one particular homework.

- **Late Policy:** Stuff happens, so there's a generous late policy to account for the unexpected (e.g., food poisoning, a surprisingly difficult homework in this or another class, an interview, an alien abduction, etc, etc.).

  - **Penalties:** 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. This is applied as percentage points off. For instance, if you got a 90 and you were 20 hours late, it is now a 70.

  - **Late Hours:** Because unexpected things happen, you will have 72 late hours over the semester that you can use on **homework**. This will be done automatically using times recorded in gradescope. I will not keep track of lateness for you.

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 really exceptional circumstances that merit it.

- **Regrade Policy:** In a large class, it is likely that we are going to make a few mistakes. We are committed to making mistakes right. You are free to submit regrade requests, especially if you believe there is a serious error (although not all regrade requests will be approved).
  - **Regrade Deadline:** Submit your regrade request within **one week** of the grades being released. This regrade deadline is firm. We will announce when grades are released. It is your responsibility to check it.
  - **Small Judgment Calls:** Please do not quibble about small ambiguous decisions. Regrade requests for things that are off by ≤ 1 points on any particular problem, or which in total change the homework by ≤ 3 points will not be considered. If you are worried this will change your grade due to rounding, send an email to course staff titled `EECS 442F19 Minor Regrade Request` detailing your request and, at the end of the semester, if the points could change your grade, then we'll handle it then.
  - **Administrative Fs:** If you submit handwritten homework or don't mark the pages in gradescope, you will get a 0. These are subject to the same deadlines as other regrades. Don't lose tons of points out of laziness.

Project

This is an opportunity to explore a topic in depth and should involve substantial work. This is a group project, involving 3-5 people. 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 two homeworks' worth of work per-person. If you cannot find partners, there will be a piazza discussion for finding project partners. It will consist of:

1. **Proposal (2 pages, 5%):** 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.
2. **Progress report (2 pages, 10%):** The progress report should aim to explain what progress you've made and be a stepping stone to the report (i.e., almost all of your work for the progress report should be usable for the final report).
3. **Final Project Report (6 pages, 20%):** The final project report should explain what you have done.

All written work should be in CVPR format `.zip` file here for word and latex. **Overall, please remember that we do not see your hard work, we only see the products you deliver.**

General Remarks

1. *Doing well in class:* You are highly encouraged to read the course document on doing well: computer vision is a relatively difficult subject and requires simultaneous mastery of linear algebra, programming, and converting relatively vague specs into fairly specific code. If you approach the assignments from the wrong angle, they will be incredibly burdensome.
2. *Classroom etiquette:* Above all, please avoid being disruptive in class, for both me and for your classmates.
   (a) Don't talk during the lecture. I lose my train of thought, have to ask you to stop talking, and this wastes everyone's time. If you want to talk, be an adult and use gchat/wechat/whatsapp/email/slack/etc.
   (b) Please don't eat. Drinking's fine.
   (c) If you use a computer in class, you should really be only using it for taking notes. I'd discourage you from doing work for my class or other classes or browsing the internet. If you absolutely positively need to use your computer for other stuff, just be discreet and don't sit in the front row watching fail videos.
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. You should read the discussion in the homework policy of what constitutes acceptable collaboration.**