ENGR 27/CPSC 72 – COMPUTER VISION
COURSE SYLLABUS

Course Description

Computer vision studies how computers can analyze and perceive the world using input from imaging devices. This course introduces topics in computer vision of particular relevance to engineers, with an emphasis on hands-on applications. We will also learn about state of the art techniques and sensors, including the Microsoft XBOX Kinect sensor. The course is divided into three broad areas of investigation:

- **Appearance based methods** including filtering, morphological operators, convolutions, frequency domain methods, edge and feature detection, correlation, and template tracking.
- **Geometry based methods** including multiple view geometry, structure from motion, visual odometry, stereo and structured light, and shape from shading.
- **Probabilistic and learning based methods** such as segmentation, classification, and object recognition.

Instructor Information

Prof. Matt Zucker
Office: Hicks 219 – office hours: Mon 2:00-4:00 PM
Phone: (610) 328-8636
Email: mzucker1@swarthmore.edu

Meeting Times

Lecture: Hicks 211, Tuesday, Thursday 11:20 AM - 12:35 PM
Labs: self-scheduled

Prerequisites

One of ENGR 15, CPSC 21 or CPSC 35 is required; MATH 27 or 28 is strongly recommended. In practice, I expect you to understand elementary programming concepts, including basic loops, functions, and array processing. I also expect you to be comfortable with linear algebra concepts such as solving linear systems, matrix inverses, rank, rotations and translations, dot products and cross products, and eigenvalues/vectors.

Textbook


The book is available in its entirety online at [http://szeliski.org/Book/](http://szeliski.org/Book/). I have not ordered copies at the bookstore, but if you want a hard copy you should be able to find it for purchase online.

Assignments and grading

Homework consisting of math, short answer questions, and small programming exercises will be assigned weekly. There will be four larger projects/labs which are both more open ended and more programming intensive, as well as a self-directed final project. Projects and labs are self-scheduled. The course has two midterm exams and a final exam. Grading will follow approximately the divisions shown below:
• Homework: 20%
• Projects/labs: 35%
• Midterm exams: 2 x 15%
• Final exam: 15%

Collaboration policy

• Homework should be completed individually.
• Projects and labs should be completed in groups of two.
• Although you may discuss the homeworks and labs with your other classmates, I expect that the work you turn in is your own. You are not permitted to directly copy answers or source code under any circumstances.
• If you do discuss your solutions with your classmates, I expect you to disclose any such collaboration clearly in your writeups and/or reports. Err on the side of caution – it’s the best way to avoid awkward conversations about suspicious similarities between assignments with no attribution of credit.
• Cite any external sources used, including the textbook, web sites, discussions with other professors, etc.

Late policy

Homework will generally be assigned on Thursday, and due at the start of class the following Thursday. An assignment may be turned in up to four days late (i.e. on the following Monday) for half credit. Students get two free late turn-ins without penalty.

Due to the collaborative nature of the work, labs and projects will be strongly penalized for lateness.

I will do my best to accommodate you in extraordinary circumstances. Bear in mind that advance notice of such circumstances is always better.

Webpage and mailing list

The course webpage is at http://www.swarthmore.edu/NatSci/mzucker1/e27_s2015/index.html. This page will be regularly updated with assignments, projects and reading. You are expected to be responsible for checking for webpage updates in a timely fashion.

You should be automatically subscribed to the course mailing list, which you are also responsible for checking. Throughout the semester, we will use the list to communicate about course information and technical help for programming. The mailing list should not, however, be used to share solutions to homework assignments, projects, or labs. If you’re unsure about whether to post something, feel free to email me first.