**Introduction to Computer Science, CSC 589, Spring 2014**

Instructor: [Prof. Bei Xiao](https://sites.google.com/site/beixiao/), American University.

Contact: [bxiao@american.edu](mailto:bxiao@american.edu), Office: SCAN 110.

Lecture Time: Mon/Thu 11:45am-1pm

Location: SCAN

Office hours: Monday 4-5:15pm, Thursday 2-3pm (SCAN 110)

Website: http://nw08.american.edu/~bxiao/CSC589

**Textbooks and references: (all books are in library reserve and in the bookstore)**

1. (Required) R. Szeliski, Computer Vision: Algorithms and Applications available at

<http://research.microsoft.com/en-us/um/people/szeliski/Book/>

1. (Required) Computer vision with Python. O’Reilly.
2. OpenCV -Python tutorial <http://docs.opencv.org/trunk/doc/py_tutorials/py_tutorials.html>
3. (Recommended) Multiple View Geometry in Computer Vision, 2nd Edition, by R. Hartley, and A. Zisserman, Cambridge University Press, 2004.

**Computer and software:**

Python 2.8 and Numpy, Scipy, OpenCV should be downloaded. All of these libraries must be downloaded and tested to run during the first two weeks of classes. This is a hands-on and technical class. You are extremely encouraged to bring your own laptop to class to follow in-class demos. It is strongly recommend that students also obtain and download MATLAB. Some exercises and demos might need to use MATLAB.

**Perquisites:** CSC280 is strongly recommended. If you haven’t taken any programming classes before, you might consider take this class **after** you obtain some programming skills. Without programming skills, you will have a hard time complete the homework. At least one semester of Calculus is required. Linear algebra and probability are strongly recommended.

**Course Description:**  This course is an introduction to current algorithms used in computer vision and computational photography (automatic image editing and manipulations). We will start from low-level image processing (edges), and then move to mid-level feature analysis (texture, color, motion), and eventually to high-level image and video understanding (objects, faces, scene, human activity). The topics include basic image processing and image analysis, camera models, texture synthesis, motion analysis, automatic image editing, object and scene recognition, face and pose recognition and a gentle survey of machine learning methods for computer vision. Students will learn the state-of-the art tools in computer vision and also hands-on experiences on image manipulations.

We will use Python as our primary programming tool in this course. Codes written in C and MATLAB might be demoed if needed. Working models of computer vision will be demoed in class and homework will be programming exercises with tools such as Numpy, SciPy, and OpenCV.

**Tentative Schedule:**

**January**

Introduction, install and review of Numpy, Scipy, MATLAB.

Image formation and camera models, projective geometry

Light and color

Image processing and filtering, image statistics, image gradients

Image pyramids

**February**

Edge detection

Interest points and corners

Texture analysis

Image Warping

Image panoramas, Correspondence

Local image features

**March**

Multiple view geometry

Stereo-vision

HDR Images, Augmented reality

Panorama images

Motion Analysis, Optical flow

**April**

Machine learning intro: clustering

Object recognition

Face detection

Scene understanding (image database)

Video analysis, human activity monitoring, surveillance.

Human computation and crowd sourcing

Final project presentations and write-ups

**May**

Final project due.

**Expected Working hours outside class**

Please refer to regulations regarding credit hours and expected workloads:

http://www.american.edu/provost/undergrad/undergrad-rules-and-regulations.cfm

**Expected Learning outcomes:**

1. Knowing how to use programming tools (Python or MATLAB) to process, manipulate and make simple inferences from images.
2. Having a basic understanding of the history, state-of-the-art methods, and applications of computer vision.
3. Being able to apply off-the-shelve computer vision algorithms to solve practical problems in computer vision.
4. Gaining exposure to common mathematical tools in engineering such as frequency analysis, convolution, linear algebra, numerical methods, and basic probability theory.
5. Being able to learn advanced classes in computer vision, computer graphics, data sciences, signal processing, and machine learning in the future.

**Grading:** 60% homework projects, 15% mid-term exam, 20% Final project, 5% attendances including quizzes. Quizzes sometime need to turn in.

Late policy:

 • 5 free late days for homework only (**not final project**)– use them in your ways;  
 • After you have finished using all of your late days, there is a 25% off per day late penalty;  
 • No submission is accepted after 3 late days per homework (zero points).

A final letter grade will be converted from the percentage you receive through out the course.

Grading Scale listed below:

93-100 A 4.0

90-93 A- 3.67

86-89 B+ 3.33

81-85 B 3.0

76-80 B- 2.67

71-75 C+ 2.33

66-70 C 2.00

61-65 C- 1.67 cut off for CS major in order to receive credits for major requirement.

51-60 D 1.00

0-50 F 0.00

Please refer to university academic rules for converting grading scale and the corresponding honors degree requirements:

<http://www.american.edu/provost/undergrad/undergrad-rules-and-regulations.cfm#5.1>

**Homework Policy (late policy):**

Late policy: Assignments are due midnight of the due date. Each student will have a total of **five** free late (calendar) days to use for the assignments. Once these late days are exhausted, any assignments turned in late will be penalized 25% per late day. However, no assignment will be accepted more than **3 days** after its due date therefore will receive ZERO points. Late days cannot be used for the final project.

**Exam Policy:**

Mid-term exams will be announced at least one week ahead of time. If you have special needs, you need to notify me at least 5 days before to arrange the test be performed off-class in the exam center. Missed exams **cannot** be made up.

**Attendance policy:**

Because the teamwork nature of the class, attendance is mandatory. 10% of the final grade depends on in-class discussions, quizzes and exercises. Absence from the class more than 3 times without a legitimate reason will be reported to Dean of students Office. See university academic rules for absence (http://www.american.edu/provost/undergrad/undergrad-rules-and-regulations.cfm#5.1)

**Academic Integrity (please read):**

Plagiarism and academic misconduct are defined in the University Academic Integrity Code. You should be familiar with what constitutes academic dishonesty. In particular, you should observe the following rules: Collaboration on projects is restricted (if you have high-level discussion with another person, please **write down the name of the person**). If I found two people’s codes are extremely similar, both people will be reported to the university.

Any information taken from the Internet, books, or anywhere else for use on your assignments must be **cited**. If you need help with your homework/project, you can discuss the ideas you read from the internet/paper. But line-by-line copying codes are completely forbidden. This is extremely important!! Your code must be entirely your own work. Any plagiarism (**including copying codes form online-forums or other course websites**) will be reported to the university.