ECE 847 Digital Image Processing
Fall 2013

Overview: This course introduces students to the basic concepts, issues, and algorithms in digital image processing and computer vision. Topics include image formation, projective geometry, convolution, Fourier analysis and other transforms, pixel-based processing, segmentation, texture, detection, stereo, and motion. The goal is to equip students with the skills and tools needed to manipulate images, along with an appreciation for the difficulty of the problems. Students will implement several standard algorithms, and evaluate the strengths and weakness of various approaches.

Instructor: Stan Birchfield, 864-656-5912, stb at clemson

Office hours: Online by appointment. Note that technical and administrative questions will be handled primarily via an online forum to ensure that all students have the maximum benefit of the answers provided. Email is discouraged as a form of communication because it does not provide all students with access to the answer. Emails and individual appointments should be used only if questions cannot be resolved via the online forum. Details regarding the forum will be in a separate communication.

Class time: Mon. – Fri. (online)

TA / Grader: TBD

Text (may be helpful but not required):

- Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010 **http://szeliski.org/Book/**
- Sonka, Hlavac, Boyle, *Image Processing, Analysis, and Machine Vision*, 3rd ed., 2008
- Umbaugh, *Digital Image Processing and Analysis*, 2nd ed., 2010
- Forsyth and Ponce, *Computer Vision: A Modern Approach*, Prentice-Hall, 2nd ed., 2011
- Gonzalez and Woods, *Digital Image Processing*, 3rd ed., Prentice-Hall, 2008

Prerequisites: Probability and statistics, linear algebra, signals and systems, programming skills, creativity and enthusiasm.

Course materials: Online lectures and lecture slides, as well as any other material for the course, are available on Blackboard.

Additional (optional) materials: **http://www.ces.clemson.edu/~stb/ece847**

Attendance policy: This is an online course, so attendance is not required. Nevertheless, it is expected that you will view all the online lectures and study the lecture notes (slides), as well as perform independent reading to fill in any unanticipated gaps.

You will be responsible for all the material covered in these lectures. You are also encouraged to seek out your fellow students, so that you can learn together in a mutually supportive way, subject to the restrictions regarding source code mentioned below.

Objectives: By the end of the course, students should be able to do the following:

- *Fundamental concepts.* Define the problems of compression, restoration, segmentation, detection, recognition, segmentation, reconstruction, and tracking. Explain the relationship between image processing, machine vision, computer vision, and computer graphics. Explain the concepts of regions, edges, filters, transforms, photometry, and geometry.
- *Computation.* Write C/C++ code to implement standard algorithms (such as region analysis, edge detection, template matching, segmentation, stereo correspondence, perspective projection, epipolar geometry calculation, color discrimination, and/or compression).

Grading:

- *Programming Assignments.* There will be seven programming assignments. All assignments are due at 11:59pm on the due date; but an 8-hour grace period is automatically granted so that no points will be deducted for anything submitted before 8:00am the next morning. Late assignments (after the grace period) will be accepted at a penalty of 10 points per day (according to a six-day work week), up to a maximum of 35 penalty points; assignments turned in more than one week late will receive a score of zero. Students are allowed 3 late days total throughout the semester. Although students are encouraged to discuss the assignments with their colleagues, they must turn in their own work. Looking at the written work or code of another student (including former students or students at other universities), or copying that work is strictly prohibited. Similarly, students may seek information on the web to aid their understanding of the assignments, but any such work may not be copied. Students who violate university rules on academic dishonesty will be subject to disciplinary penalties, such as failure in the course and/or dismissal from the University.
- *Midterm and Final.* There will be a written midterm and final exam. Both of these will be open book (if you have one) and open notes (including the lecture slides). The instructor and/or TA should be notified at least one week in advance of any conflict that would prevent the student from taking an exam, so that alternative arrangements can be made. Without prior approval, missed exams cannot be made up except in cases of extreme urgency and importance (e.g., sudden illness, death in the immediate family, etc.).
- *Grading.* Grades will be determined by the following formula: programming assignments (70%), midterm (10%), final (20%).

Academic integrity: As members of the Clemson University community, we have inherited Thomas Green Clemson's vision of this institution as a “high seminary of learning.” Fundamental to this vision is a mutual commitment to truthfulness, honor, and responsibility, without which we cannot earn the trust and respect of others. Furthermore, we recognize that academic dishonesty detracts from the value of a Clemson degree. Therefore, we shall not tolerate lying, cheating, or stealing in any form. This includes looking at and copying the code of others, as mentioned above.

Disability Access: It is University policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have disabilities. Students are encouraged to contact Student Disability Services to discuss their individual needs for accommodation. In order to obtain accommodations, the student must notify the instructor at least one week before any exam for which accommodations are needed.

Topics:

- pixel-based processing (edge and region analysis, distance measures, histograms, morphological operations)
- filters and edge detection (convolution, Gaussian, Laplacian of Gaussian, noise types, simple edge detection methods, scale-space)
- pattern detection (Hough transform, matched filter, ROC curve)
- segmentation (region growing, split-and-merge algorithm, Gestalt, watershed algorithm)
- texture (co-occurrence matrices, autocorrelation, entropy, filter banks, pyramids)
- transforms (Fourier, cosine, Gabor, and wavelet transforms; basis functions; PCA)
- projective geometry (stratification of geometry, homography and collineation, homogeneous points, projection models)
- image formation (geometry, radiometry, photometry, color, sensors)
- stereo (geometry, correspondence, constraints, rectification)
- motion (optical flow and motion field, aperture problem, feature detection and tracking)