ECE 877 Computer Vision
Spring 2012

Instructor: Stan Birchfield
Office: 209 Riggs Hall, 656-5912, stb at clemson
Office Hours: Wed 4-5, or by appointment
Lectures: MWF 1:25 – 2:15, Riggs 223
Course website: **http://www.ces.clemson.edu/~stb/ece877**

Text (recommended):

- 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; ECE 847

Overview: This course builds upon ECE 847 by exposing students to fundamental concepts, issues, and algorithms in digital image processing and computer vision. Topics include classification, shape, texture, model fitting, camera calibration, multi-view geometry, 3D reconstruction, and function optimization. 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, evaluate the strengths and weakness of various approaches, and explore a topic of their own choosing in a course project.

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

- *Fundamental concepts.* Define the problems of classification, shape representation, texture, model fitting, camera calibration, multi-view geometry, 3D reconstruction, and function optimization. Explain the various algorithms and concepts proposed to solve these problems and discuss the relationships and tradeoffs between them.
- *Computation.* Write code to implement standard algorithms (such as template matching, segmentation, snakes / level sets, texture synthesis, RANSAC, mosaicking, camera calibration, perspective projection, epipolar geometry calculation, color discrimination, structure from motion, 3D reconstruction).
- *Course project.* Research a topic by finding and reading relevant research papers, writing a survey of past research, and identifying holes in the current approaches. Develop an approach to solving the problem, implement and test the solution, and critically evaluate the results. Effectively communicate the steps and conclusions of the investigation in an oral presentation and written report.

Topics:

- classification (Bayesian decision theory, statistical pattern recognition, linear methods, architectures, boosting, ROC curve)

- shape (active contours, B-splines, shape context, generalized cylinder, deformable template, Mumford-Shah, dual contour, level sets)
- texture (stuff vs. things, co-occurrence matrix, Fourier transform, filter banks, wavelets, Gabor filters, steerable pyramids, texture synthesis, feature detection and matching, scale space)
- model fitting (least squares, line fitting, robust estimators, Hough transform, Douglas-Peucker algorithm, RANSAC, mosaicking)
- camera calibration (Hildreth's radial calibration, Tsai's calibration model, Zhang's planar algorithm)
- multiple view geometry (homography matrix, camera projection matrix, essential and fundamental matrices)
- 3D reconstruction (SVD, structure from motion, rank constraint)
- function optimization (expectation maximization (EM), Newton-Raphson, Gauss-Newton, gradient descent, conjugate gradient, Levenberg-Marquardt, multiway cut)

Grading:

- *Programming Assignments.* There will be six programming assignments. Late assignments 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 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.
- *Quizzes.* There will be six or seven short quizzes during class, which may be announced or unannounced. A student who keeps up with lectures and reading assignments should do well on these quizzes. Unless agreed upon in advance by the instructor, missed quizzes will receive a grade of zero. To receive a passing grade in the course, a student must take at least four quizzes. The grade of the lowest quiz may be dropped, subject to instructor discretion.
- *Final project.* During the last few weeks of the semester, students will explore a topic of their own choosing in more depth. Students are expected to investigate the topic, define the scope, implement and test an algorithm (whether novel or existing), and report their conclusions in both an oral presentation and a written report. For the project, students may work alone or in pairs. The amount of work invested in the project should be approximately equal to two programming assignments.
- *Grading.* Grades will be determined by the following formula: programming assignments (60%), project (20%), quizzes (20%).