ECE 847 Digital Image Processing
Fall 2009

Instructor: Stan Birchfield
Office: 207-A Riggs Hall, 656-5912, stb at clemson
Office Hours: 2:00-4:00 F, or by appointment
Course website: **http://www.ces.clemson.edu/~stb/ece847**
Text (recommended):

- Sonka, Hlavac, Boyle, *Image Processing, Analysis, and Machine Vision*, 3rd ed., 2008
- Forsyth and Ponce, *Computer Vision: A Modern Approach*, Prentice-Hall, 2003
- Gonzalez and Woods, *Digital Image Processing*, 2nd ed., Prentice-Hall, 2002

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

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, 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 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).
- *Course project.* Determine a topic to investigate and research it by finding and reading relevant research papers. 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 a written report.

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)

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 biweekly quizzes given during class. 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 miss no more than two 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 per person.
- *Grading.* Grades will be determined by the following formula: programming assignments (60%), project (20%), quizzes (20%).