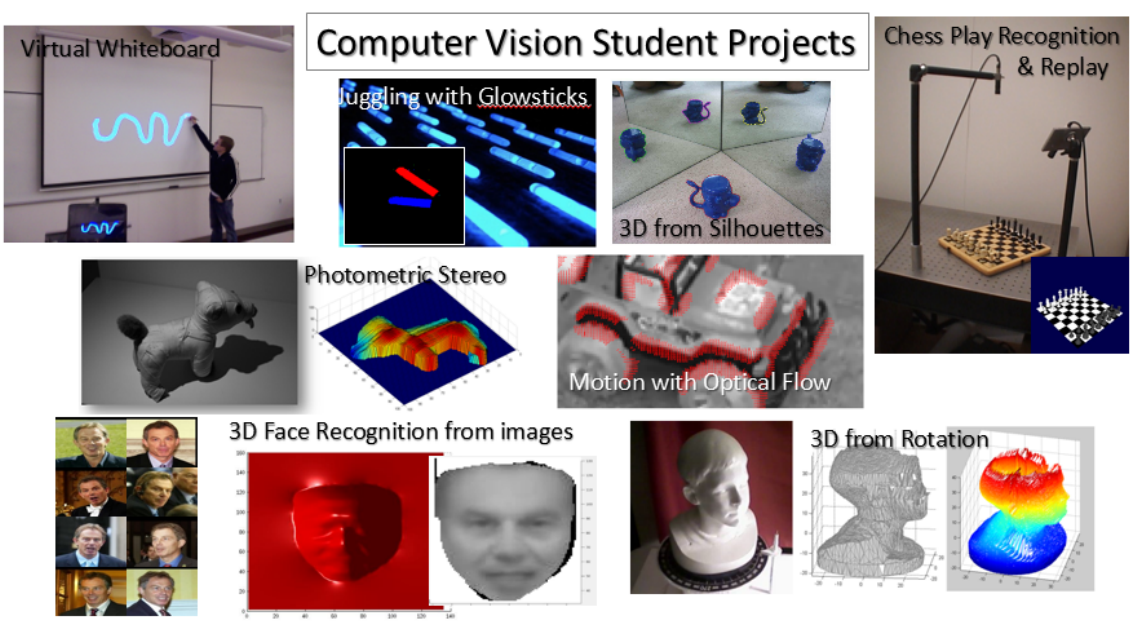
CS-GY 6643 Computer Vision (class#23724) Spring 2016
Guido Gerig

Goals and Objectives

- Main Goal: Computing properties of our 3-D world from passive and active sensors.
- To introduce the fundamental problems of 3D computer vision.
- To introduce the main concepts and techniques used to solve those.
- To enable participants to implement own solutions to applications of their choice.
- To understand basic CV methodologies as used in industrial and entertainment systems.

Computer Vision (*following Tomaso Poggio, MIT*) : Computer Vision, formerly an almost esoteric corner of research and regarded as a field of research still in its infancy, has emerged to a key discipline in computer science. Vision companies have emerged and commercial applications become available, ranging from industrial inspection and measurements to security database search, surveillance, multimedia and computer interfaces. Computer Vision is still far from being a solved problem, and most exciting developments, discoveries and applications still lie ahead of us. Understanding the principles of vision has implications far beyond engineering, since visual perception is one of the key modules of human intelligence.

Who should attend this course? Graduate students who are interested and motivated in learning the fundamental concepts of 3D Computer Vision. Assignments encourage students to use own cameras and smartphones and to be creative in regard to applying methodologies to 3D scenes and applications of their choice. Research in 3D Computer Vision is closely related to computer graphics and image analysis, but also to areas like machine and robot vision, digital media and arts.



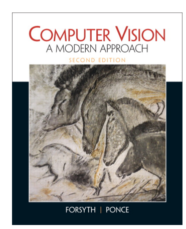
Prerequisites: Proficiency in programming, and familiarity with matrix arithmetic and geometry. Most of the knowledge required should be part of the standard background in Computer Science, fundamentals of programming and undergraduate/graduate Mathematics and Geometry. The teacher will provide introductions into algorithmic and computational concepts of image processing where necessary. It is important to note that this is **not an image processing course** but a course on 3D vision which uses image processing as basic methodology but 3D geometry and mathematical methods to recover 3D information from single or multiple 2D images, thus being complementary to Image Processing courses.

Course Overview (subject to change)

- Introduction
- Image Formation and Image Models
  - Projective Geometry, Cameras
  - Camera distortions and artifacts
  - Camera calibration
- Early Vision: Multiple Images
  - The geometry of multiple views
  - Stereo Vision, epipolar constraints, disparity
  - Shape from stereo, correspondence
- Shape from X
  - Reflectance map
  - Shape from shading
  - Photometric stereo
  - Shape from optical flow (moving camera, moving objects)
  - Rotating camera, Silhouettes, Space carving
  - Light stripe encoding
  - Laser range systems (TOF)
- High Level Vision
  - Model-based Vision
  - Aspect graphs
  - Tracking
  - Finding Templates and Recognition

Textbook

The textbooks for this course is "Computer Vision: a modern approach" by David Forsyth and Jean Ponce [1], authored by two very well respected researchers in the field. Additional material (scientific papers, chapters from other textbooks etc.) will be distributed during the course and made available via the course web-site. We will also use materials from from "Computer and Robot Vision" by Haralick&Shapiro, from "Multiple View Geometry in CV" by Hartley&Zisserman, from "Robot Vision" by P. Horn, from "Three-Dimensional Computer Vision" by O. Faugeras, from "3-D Computer Vision" by Trucco&Verri, and from "Computer Vision" by Klette, all classic textbooks by leaders in the field.



Learning approach

- Students should read the relevant chapters of the books and/or reading assignments before the class.
- In the course, the material will then be discussed in detail and motivated with real world examples and applications.

[1] (http://www.pearsonhighered.com/educator/product/Computer-Vision-A-Modern-Approach-2E/9780136085928.page)

- There will be assignments with theoretical/programming questions to provide students with practical experience of some computer vision techniques.
- We will have a midterm and final exam where the methods will be reviewed.

Organization

Teaching:	Guido Gerig email: gerig@nyu.edu
Lecture time and place:	Tuesday, 6pm - 8.30pm, location tbd
Office hours:	tbd, office 10.094, 10th floor, 2 Metrotech Center
Material:	Computer Vision, David Forsyth and Jean Ponce (see link above)
Assignments:	Programming work with Matlab or C++, practical examples with real multi-camera images, range image data and images generated by students using own pictures.

Policy on Academic Dishonesty

The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness. Please see the school's policy on academic dishonesty at our school's website: `http://engineering.nyu.edu/academics/code-ofconduct/` academic-dishonesty. Students are expected to work *on their own*, as instructed by the Professor. Students may discuss projects with other individuals either in the class or outside the class, but they may not receive code or results electronically from any source that is not documented in their report. *Students must write their own code, conduct their own experiments, write their own reports, and take their own tests.* Any student who is found to be violating this policy will be given a failing grade for the course and will be reported to the authorities as described in the University's Student Code.

Assignments

There will be 4 to 5 assignments done by individuals on topics assigned approximately every 2-3 weeks by the professor. The theoretical part can be written by hand or with a text system. The practical part should be written with a text system and should include equations, hints to the programming solution, outline of solution strategy, graphs and results, and a critical discussion of results and eventual obstacles. Programming will be done in either MATLAB or other software environments as preferred by the student. Each practical assignment will have to be documented by a 5-6 page pdf report. There will also be regular short quizzes in class, followed by a discussion, in order to review the materials.

Late Policy

Assignments/projects will be accepted up to 7 days late but will be subject to 2% grade penalty (of the total points of the assignment) each day it is late (weekends included). Homework will no longer be accepted 7 days after the deadline and would afterwords count as an F.

Grading: tentative

Assignments/Projects (4-5)	40%
Exam I: Midterm (theoretical)	25%
Exam II: Final (theoretical)	25%
Active class participation (incl. Quizzes)	10%