# SYLLABUS

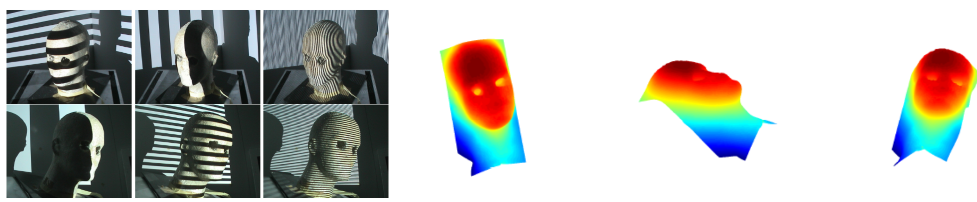
# 3D Computer Vision

## Computing properties of our 3-D world from passive and active sensors

## CS Catalog Number 6320
## Spring Semester 2012
## M,W 1.25 - 2.45 WEB L126


## Guido Gerig (gerig@sci.utah.edu)


**Class web-page:**
`http://www.sci.utah.edu/~gerig/CS6320-S2012/CS6320_3D_Computer_Vision.html`



## Syllabus

**Goal and Objectives**:

- 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 solutions for reasonably complex problems.
- To enable participants to understand basic methodology that is discussed in the computer vision literature.

**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 in learning the fundamental concepts of 3D Computer Vision or desire to use compute vision techniques in their research. Research in 3D computer vision is closely related to computer graphics, to image analysis, but also to areas like machine/robot vision, surgical planning and more.

**Prerequisites:** Most of the knowledge required should be part of the standard background in Computer Science and undergraduate/graduate Mathematics and Geometry. **It is a prerequisite that students have knowledge in fundamental image processing procedures and techniques.** The graduate course *CS6640 Image Processing* or an equivalent graduate level image analysis or graphics/imaging course are highly recommended since this course builds on some fundamental image processing methodology and Matlab or C++ code development. Students with different background and curriculum need to discuss suitability and options with the teacher.
It is important to note that this is **NOT AN IMAGE PROCESSING COURSE** but a course about 3D vision which uses image processing as basic methodology but 3D geometry and mathematical methods of optimization to extract 3D information from single or multiple 2D image or other sensor data.

## Overview of the Course

- Introduction
- Image Formation and Image Models
  - Projective Geometry, Modeling Cameras, Projection Matrix
  - 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 textbook for this course is " Computer Vision: Algorithms and Applications" by Richard Szeliski, published by Sprnger Verlag **http://szeliski.org/Book/**. This is a most recent book on computer vision, authored by a most respected researcher in the field. We will also use materials from "Computer Vision: A Modern Approach" by David Forsyth and Jean Ponce, 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. It is not expected that students purchase another book as all additional material (scientific papers, chapters from other textbooks etc.) will be distributed during the course and made available via the UofU webct/blackboard system **(https://webct.utah.edu/)**.

## 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.
- There will be assignments with theoretical/programming questions to provide students with practical experience of some computer vision techniques.
- There ill also be a final programming project where students will solve a real world problem using computer vision techniques.

## Organization

| | |
|---|---|
| Teaching: | Guido Gerig<br>email: gerig@sci.utah.edu<br>Tel: (801) 585-0327 |
| Lecture time and place: | M,W 1.25 - 2.45 / room WEB L126 |
| Office hours: | tbd, office WEB 4893 |
| TA: | Kathlea Quebbeman<br>quebbeka@cs.utah.edu<br>TA Office hours: M,W 1.25 - 2.45<br>room: tbd. |
| Material: | Book R. Szelisky **(http://szeliski.org/Book/)** and handouts on WebCT |
| 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. |

## Honor Policy

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 use of sources (for projects or tests) that are not specifically given to the student by the Professor or TA, must be discussed with the Professor or TA or documented in the report. 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 (`http://www.admin.utah.edu/ppmanual/8/8-10.html`).

## Assignments

Assignments will be done by individuals on topics assigned approximately every 2-3 weeks by the professor (i.e. there will be approximately 4-5 assignments). 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 (with Imaging toolkit, CADE lab) or other SW environments as preferred by the student. Each practical assignment will have to be documented by a 5-6 page pdf report.

## Late Policy

There will be 10 points (out of 100) deducted for each day of late submission of assignments. Assignments will no longer be accepted 3 days after the deadline and would afterwords count as an F.

## How to submit your electronic files

The CADE facility has a program called handin for handing in files. To use this, you need to log into a CADE machine, and if necessary, transfer your files to that account. Here are some examples of how this program works:

```
handin cs6968
```

will show you the available subdirectories, one for each of the different assignments.

```
handin cs6968 project1
```

will show you the list of files you have submitted for the first project, in the project1 directory.

```
handin cs6968 project1 filename1 filename2 ...
```

will submit your files filename1, filename2 etc. You can't unsubmit files, but if you submit a file again, the old file is overwritten. If you need more information about handin, type

```
man handin
```

for proper documentation. You can check for uploaded files yourself to be sure that they are submitted. If you don't understand these handin directions, please ask TA or instructor well in advance of the due date.

## Grading

| | |
|---|---|
| Assignments/Projects (3) | 50% |
| Final project (programming/demo/presentation/report) | 40% |
| Class participation | 10% |

Final project is required to get a passing grade

| | | | | |
|---|---|---|---|---|
| | 87-89 B+ | 77-79 C+ | 67-69 D+ | |
| 93-100 A | 83-86 B | 73-76 C | 63-66 D | 0-59 E |
| 90-92 A- | 80-82 B- | 70-72 C- | 60-62 D- | |

## Some useful links

- The Computer Vision Homepage (http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html)
- CVonline (http://www.dai.ed.ac.uk/CVonline/)
- CVonline Wikipedia (http://en.wikipedia.org/wiki/CVonline)
- Middlebury Stereo Vision Page (http://cat.middlebury.edu/stereo/)
- CV publications (http://iris.usc.edu/Vision-Notes/bibliography/contents.html)

## Bibliography


1. ***Computer Vision: Algorithms and Applications*** by Richard Szeliksy (http://szeliski.org/Book/)
2. ***Computer Vision: A Modern Approach*** by David Forsyth and Jean Ponce
3. *Computer and Robot Vision, Vol II* by Haralick & Shapiro
4. *Emerging topics in computer vision*, by Gérard Medioni and Sing Bing Kang
5. *Introductory Techniques for 3-D Computer Vision*, Emanuele Trucco and Allessandro Verri, Prentice Hall, 1998.
6. *Three-Dimensional Computer Vision*, Olivier Faugeras, The MIT Press, 1993.
7. *Computer Vision: Three-Dimensional Data from Images*, Reinhard Klette, Karsten Schlüns, Andreas Koschan, Springer 1998.
8. *Invariant and Calibration-Free Methods in Scene Reconstruction and Object Recognition*, Richard Hartley and Joe Mundy, ARPA No. 8228, 1998. (http://www.balltown.cma.com/hartley/index.html, ftp://ftp.balltown.cma.com/pub/hartley/arpa-report.ps.gz)
9. *Geometric Computation for Machine Vision*, Kenichi Kanatani, Oxford Science Publications, 1995.
10. *Machine Vision: Theory, Algorithms, Practicalities*, E.R. Davies, Academic Press, 1997.
11. *Robot Vision*, B.K.P. Horn, MIT Press 1986.
12. *Multiple View Geometry in computer vision*, R. Hartley and A. Zisserman, Cambridge University Press


Draft syllabus, subject to change, Guido Gerig Oct. 10 2011