Computer Vision Introduction

Guido Gerig

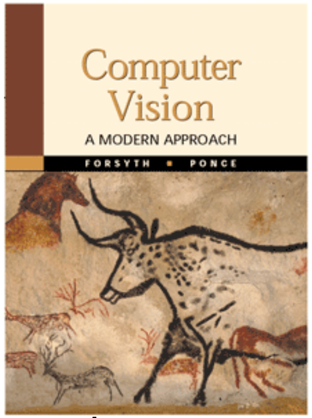
CS-GY 6643, Spring 2016

gerig@nyu.edu

Acknowledgements: some slides from Marc Pollefeys and Prof. Trevor Darrell, trevor@eecs.berkeley.edu

Administrivia

- Classes: Tue, 6 – 8.30pm
  Room RH 215
- Instructor: Guido Gerig
  gerig@nyu.edu
- TA: Rupanta Rwiteej Dutta
- Prerequisites: Graduate standing, CS-GY 5403 & MA-UY 2012, or equiv.
- Textbook:
  "Computer Vision: A Modern Approach" by Forsyth & Ponce



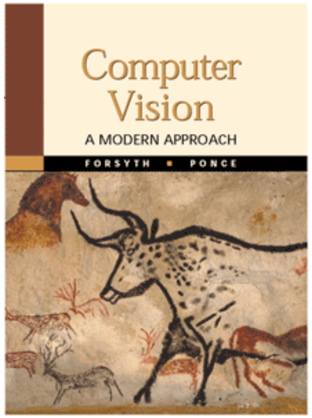
- Organization:
  Admin/Grading/Uploads:
  NYUClasses
  Slides, documents and assignments:
  Course Website

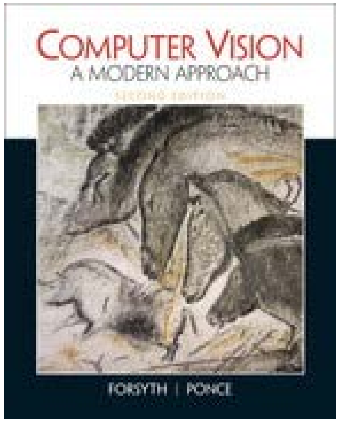
Administrivia

- Textbook:

  "Computer Vision: A Modern Approach" by Forsyth & Ponce

  Version 1 Version 2e 2012





- The Version 1 is sufficient for this course, but you can also buy the new updated version.

Web-Site

- Linked to NYUClasses CS-GY 6643 home page
- Linked to my home page (teaching page):

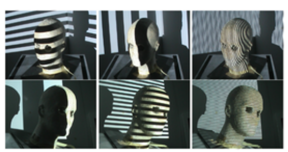
  http://engineering.nyu.edu/~gerig/CS-GY-6643-S2016/CS-GY_6643_Computer_Vision.html

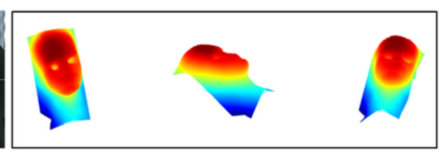
NYU TANDON SCHOOL OF ENGINEERING | Computer Science and Engineering

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

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

Syllabus, Guido Gerig (home)





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.

General Information:

Lecture: Tue 6 - 8pm, Rogers Hall, Rm 215
Instructor: Guido Gerig (gerig at nyu.edu)
Office: 2 MetroTech Center, 10.094. office hours tbd
TA: Rupanta Rwiteej Dutta, (rrd300 at nyu.edu)
Office XXXX, office hours: tbd

TA / SW Tools / Office Hours

- TA: Rupanta Rwiteej Dutta
- HW/SW: Matlab, Octave (free)
- Office Hours TA Hours/location: Tue 2-5pm, 2 MetroTech Center, 10th floor, cubicle space 10.098E (outside Gerig office)
- Office Hours instructor: Mon 2-5pm, 2 MetroTech Center, 10.094

MATLAB

- This course will support MATLAB for practical parts of assignments. C++ or other languages can be used but there will be no support and we need proof that code is fully developed by the student..
- If students want to purchase their own copy, Matlab for students is $50, or $99 (including 10 toolboxes.), link
- We will ***NOT USE Toolboxes*** but implement our own code.

NYU Honor Code

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 of Conduct:

- NYU Tandon Student Code of Conduct
- Student Code of Conduct (PDF)

Octave

About GNU Octave

GNU Octave is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with Matlab. It may also be used as a batch-oriented language.

Octave has extensive tools for solving common numerical linear algebra problems, finding the roots of nonlinear equations, integrating ordinary functions, manipulating polynomials, and integrating ordinary differential and differential-algebraic equations. It is easily extensible and customizable via user-defined functions written in Octave's own language, or using dynamically loaded modules written in C++, C, Fortran, or other languages.

GNU Octave is also freely redistributable software. You may redistribute it and/or modify it under the terms of the GNU General Public License (GPL) as published by the Free Software Foundation.

Octave was written by John W. Eaton and many others. Because Octave is free software you are encouraged to help make Octave more useful by writing and contributing additional functions for it, and by reporting any problems you may have.

Donate

Your donations help to fund continuing maintenance tasks, development of new features and the organization of Octave conferences.

Continue

Following the Continue link will take you to a Free Software Foundation page for payment processing.

Donate Bitcoins

Prerequisites

- General Prerequisites:
  - Data structures
  - A good working knowledge of programming (or willingness and time to pick it up quickly!)
  - Linear algebra
  - Vector calculus
- Assignments include theoretical paper questions and programming tasks (Matlab/Octave or C++).
- Students who do not have background in signal processing / image processing: We will provide support for most basic image processing methods.
- THIS IS NOT AN IMAGE PROCESSING COURSE BUT FOCUSES ON 3D VISION ASPECTS.

Grading - Weights

- Assignments (4-5 theory/prog.): 55%
- Midterm Exam: 15%
- Final Exam: 20%
- Class participation (active participation in summaries and discussions): 10%
- Short quizzes with in-class discussions to check understanding

Assignments & Projects

- Assignments: Theoretical and Practical Part: Example
- Strict deadlines: 10% deduction per day late, no more accepted after 3 days.
- Assignments solutions include:
  - Solutions to theoretical parts (can be handwritten and scanned)
  - Detailed report on practical solution (pdf document)
  - Code used to solve practical part
- Important:
  - Be creative with own images and experiments with your code. Try different scenarios and discuss pro's and con's.
  - Report needs to include description of what you did, description of results, and critical assessment of results, limitations, etc.
  - Code with image data submitted a separate tar/zip file.

Other Resources

- Cvonline: http://homepages.inf.ed.ac.uk/rbf/CVonline/
- A first point of contact for explanations of different image related concepts and techniques. CVonline currently has about 2000 topics, 1600 of which have content.
- See list of other relevant books in syllabus.

Some Basics

- Instructor and TA do not use email as primary communication tool.
- It will be your responsibility to regularly read the Announcements on NYUClasses.
- We don't need a laptop for the class, please keep them closed !!!!!
- Please interact, ask questions, clarifications, input to instructor and TA.
- Cell phones …., you surely know.

Syllabus

- See separate syllabus (linked to web-site and included on NYUClasses).
- Document

Goal and objectives

- To introduce the fundamental problems of 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 the student to make sense of the literature of computer vision.

CV: What is the problem?

Image Formation: From World to Image

- Camera model (optics & geometry): From points in 3D scene to points on 2D image.
- Photometry: From lights and surfaces in scene to intensity (brightness) and color in image.

Vision: From Image to (Knowledge of the) World

- Reconstruct scene (**world model**) from images.
- Extract sufficient **information** for detection/control **task**.

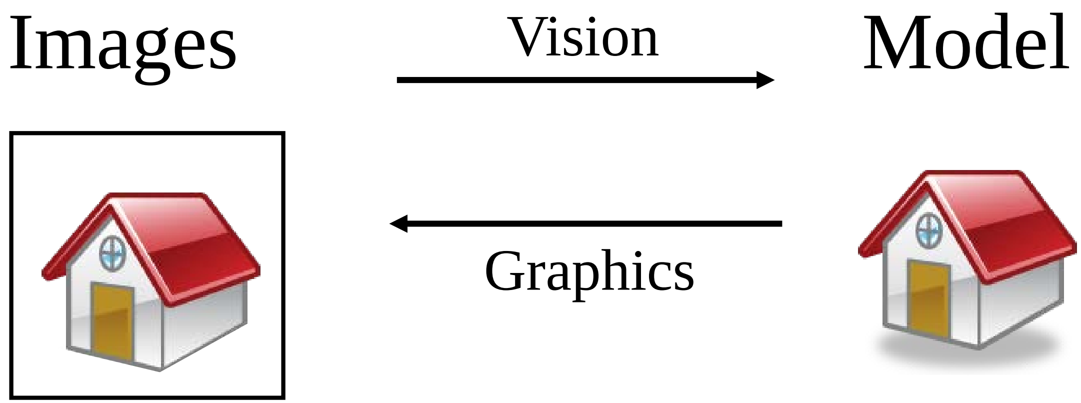
CV: A Hard Problem

- Under-constrained inverse problem – 3D world from 2D image.
- Images are **noisy** – shadows, reflections, focus, (ego-)motion blur – and noise is hard to model.
- Appearances – shape, size, color – of objects change with pose and lighting conditions.
- Image understanding requires cognitive ability ("AI-complete").
- Robotics & Control: massive data rate, real-time requirements.

What is Computer Vision?

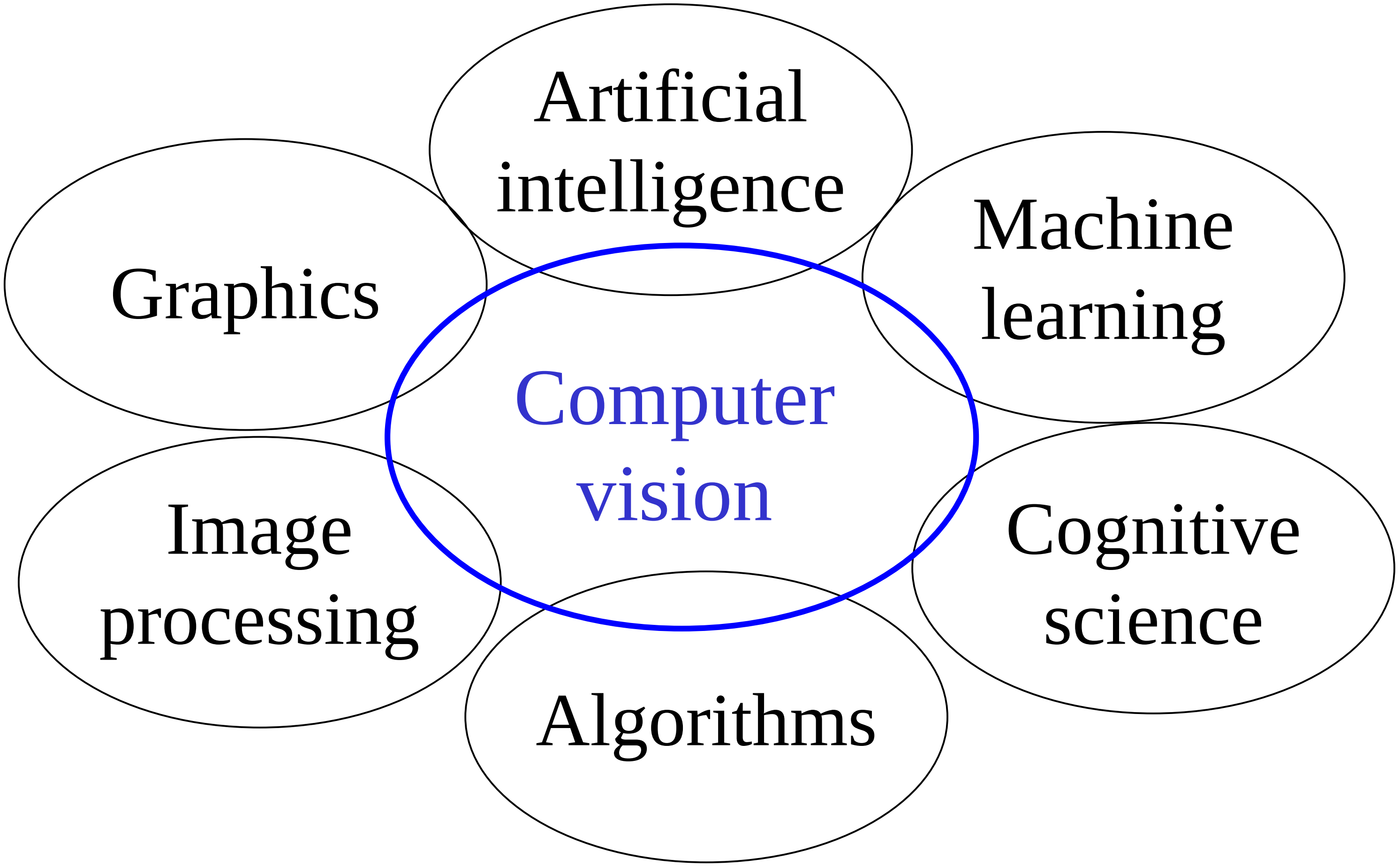
- Automatic understanding of images and video
  - Computing properties of the 3D world from visual data *(measurement)*
  - Algorithms and representations to allow a machine to recognize objects, people, scenes, and activities. *(perception and interpretation)*

Vision and graphics



Inverse problems: analysis and synthesis.

Related disciplines



Object recognition (in mobile phones)

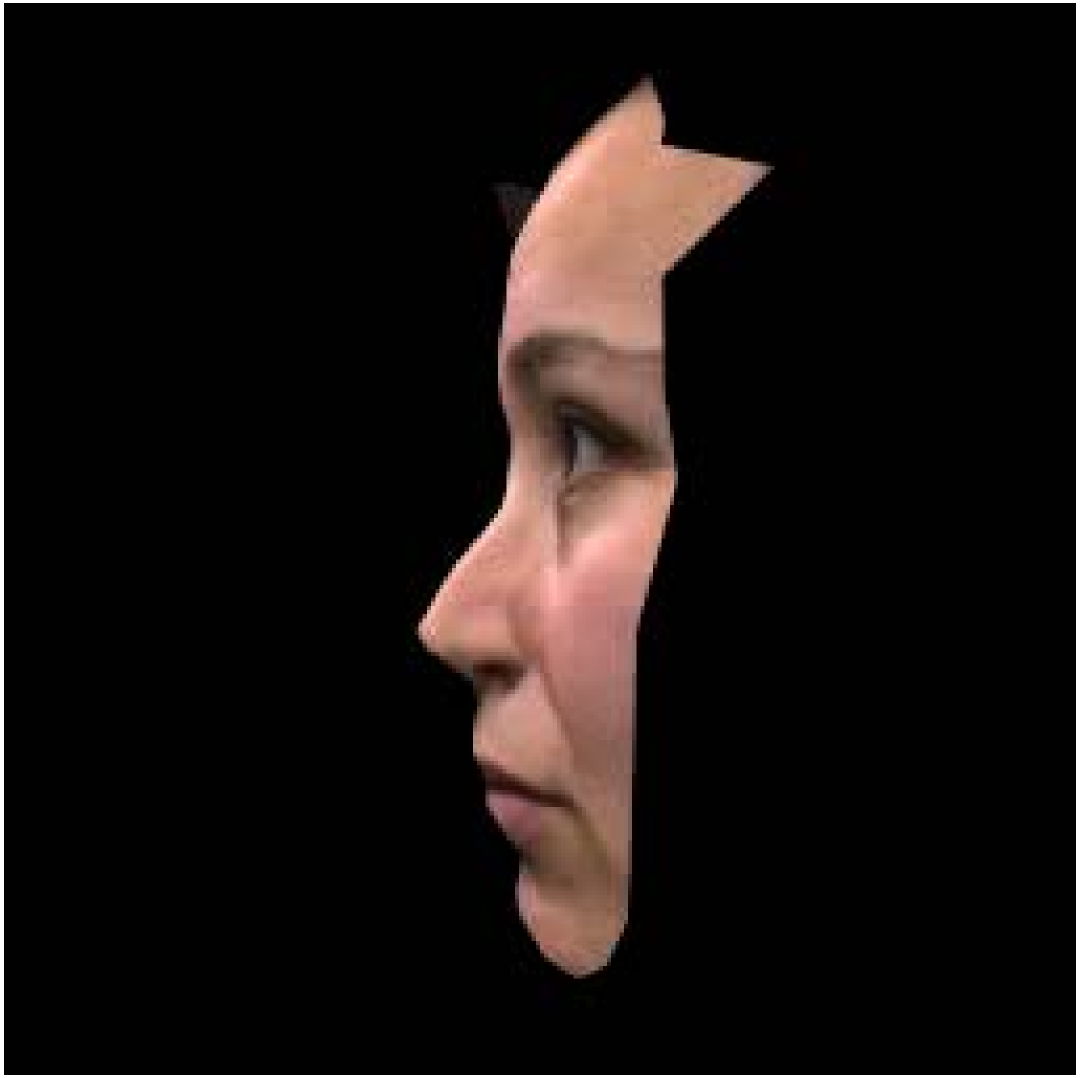
- This is becoming real:
  - Lincoln Microsoft Research
  - Point & Find, Nokia
  - SnapTell.com (now amazon)

Smart cars

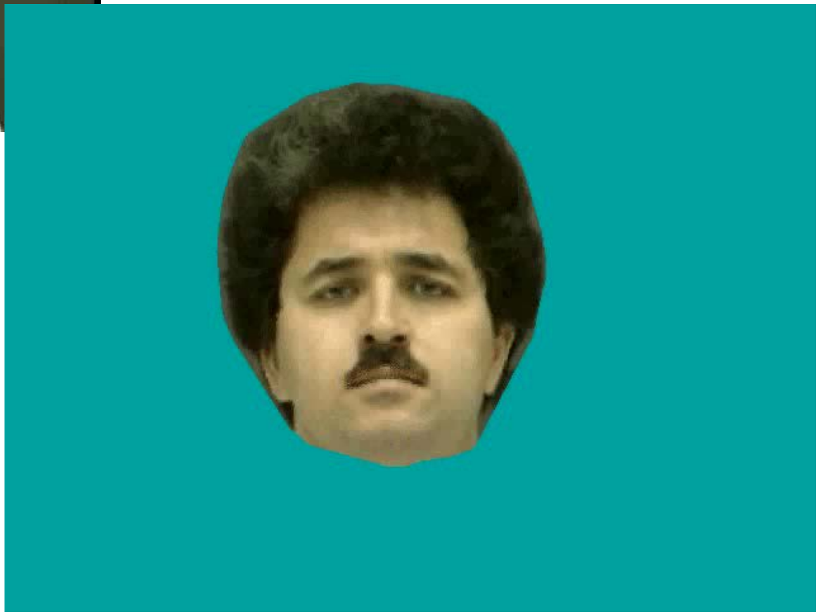
- [Mobileye](#)
  - Vision systems currently in high-end BMW, GM, Volvo models
  - By 2010: 70% of car manufacturers.
  - [Video demo](#)
  - [YouTube](#), [TestMovie](#)

Slide content courtesy of Amnon Shashua

Modeling 3D Structure from Pictures or 3D Sensors



Modeling ctd.



Main topics

- Shape (and motion) recovery

  "What is the 3D shape of what I see?"
- Segmentation

  "What belongs together?"
- Tracking

  "Where does something go?"
- Recognition

  "What is it that I see?"

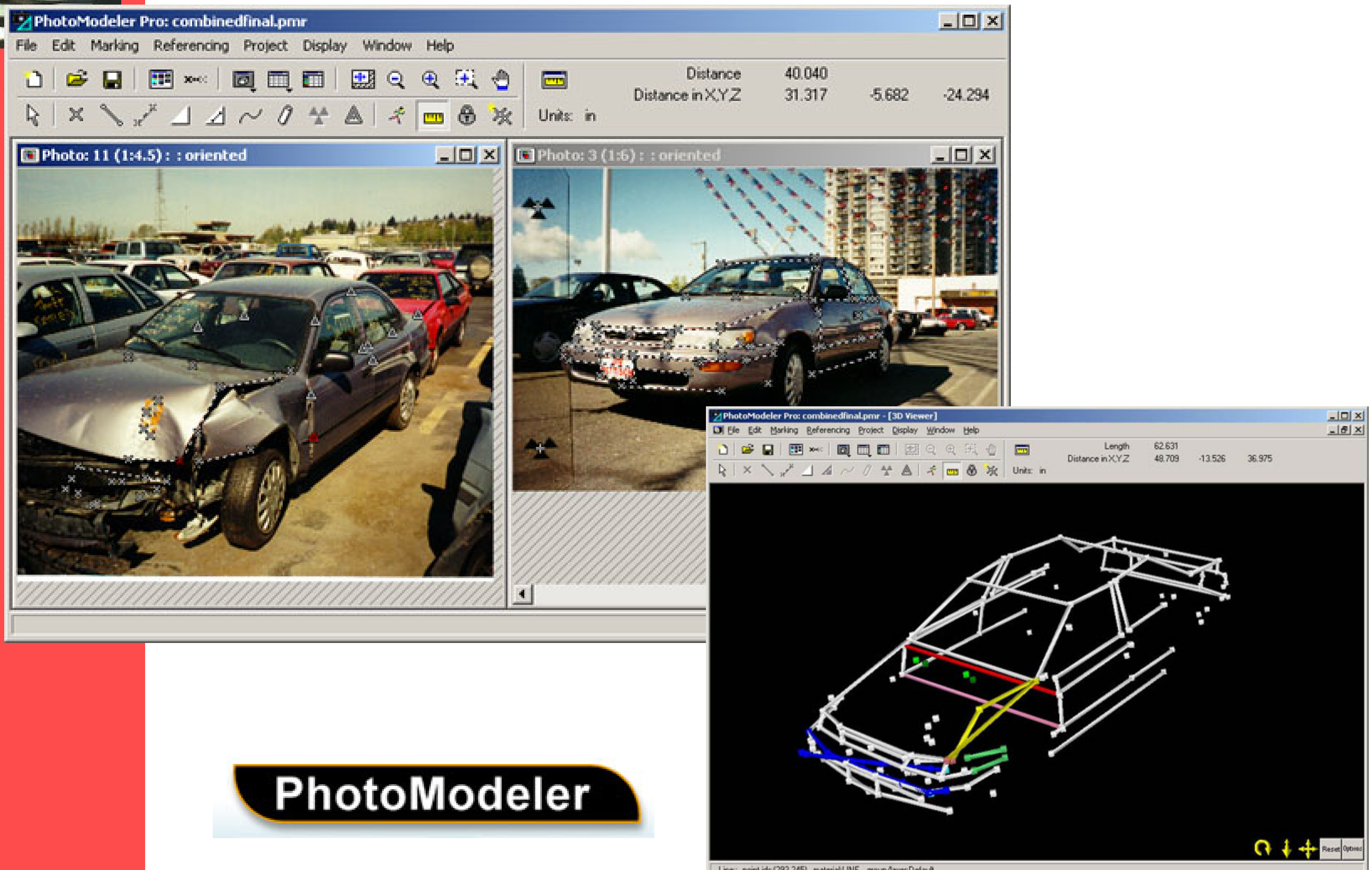
Why study Computer Vision?

- Images and movies are everywhere
- Fast-growing collection of useful applications
  - building representations of the 3D world from pictures
  - automated surveillance (who's doing what)
  - movie post-processing
  - CAM (computer-aided manufacturing
  - Robot navigation
  - face finding
- Various deep and attractive scientific mysteries
  - how does object recognition work?
- Greater understanding of human vision

Clothing

- Scan a person, custom-fit clothing

Forensics



PhotoModeler

3D urban modeling

drive by modeling in Baltimore

Earth viewers (3D modeling)

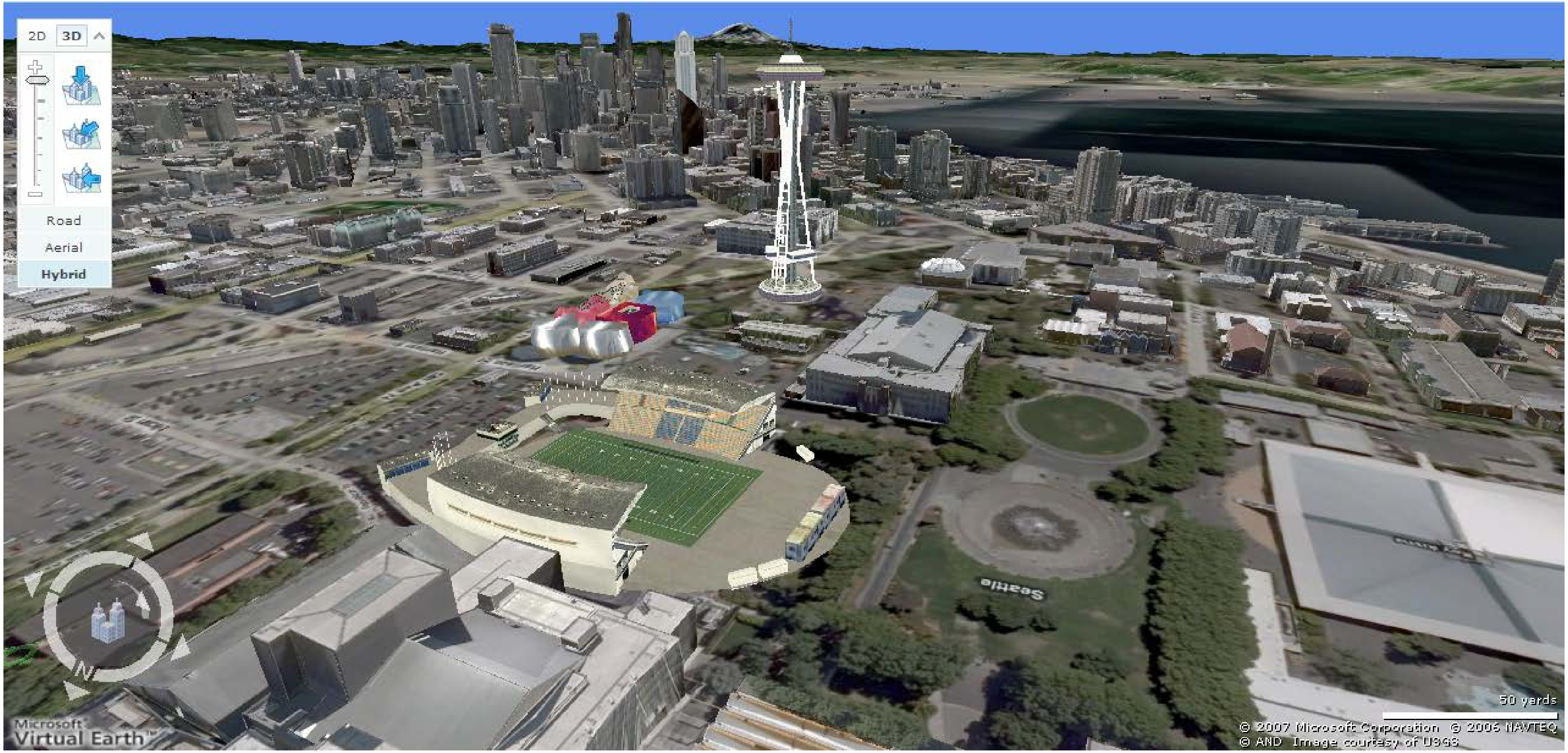
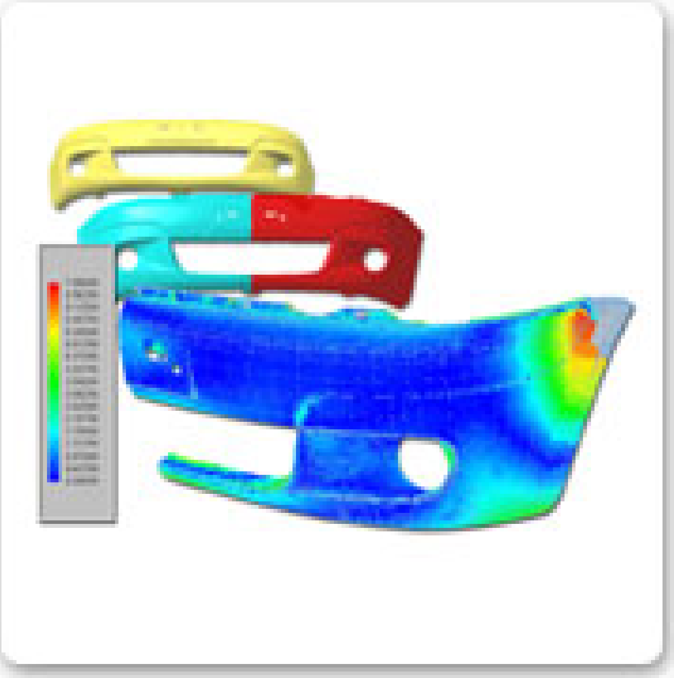


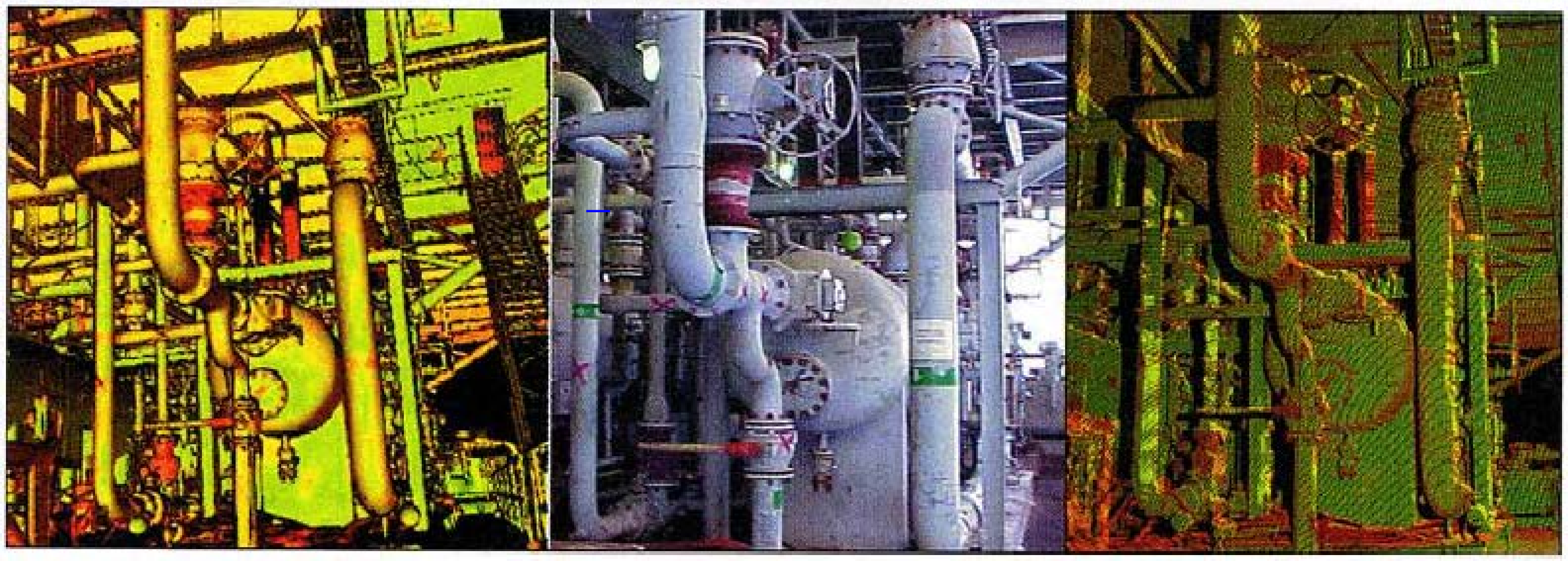
Image from Microsoft's Virtual Earth
(see also: Google Earth)

Industrial inspection

- Verify specifications
- Compare measured model with CAD



Scanning industrial sites



as-build 3D model of off-shore oil platform

Vision in space

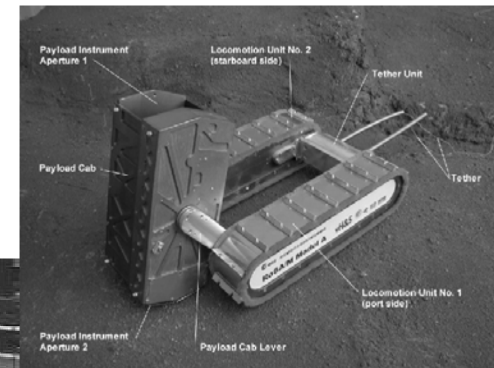
NASA'S Mars Exploration Rover Spirit captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.

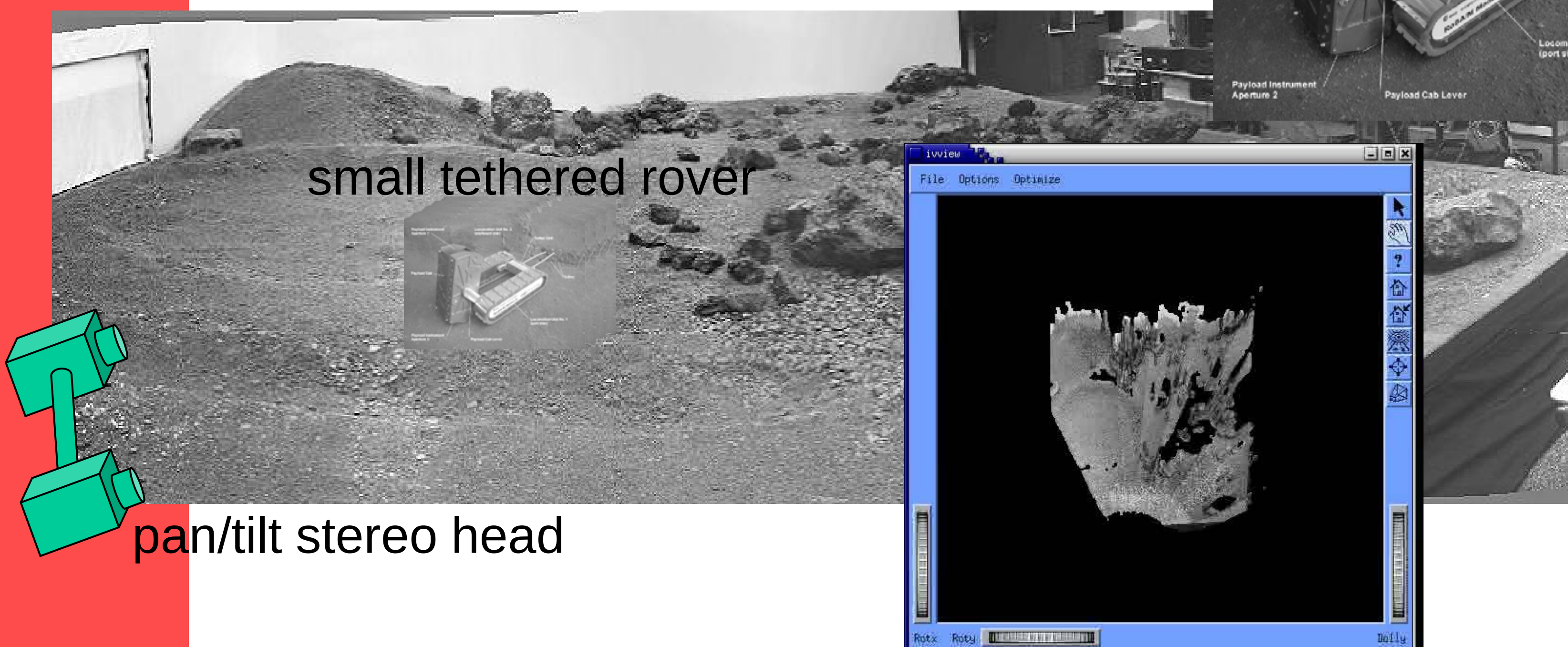
Vision systems (JPL) used for several tasks

- Panorama stitching
- 3D terrain modeling
- Obstacle detection, position tracking
- For more, read "Computer Vision on Mars" by Matthies et al.

Robot navigation

ESA project
our task: Calibration + Terrain modelling + Visualization

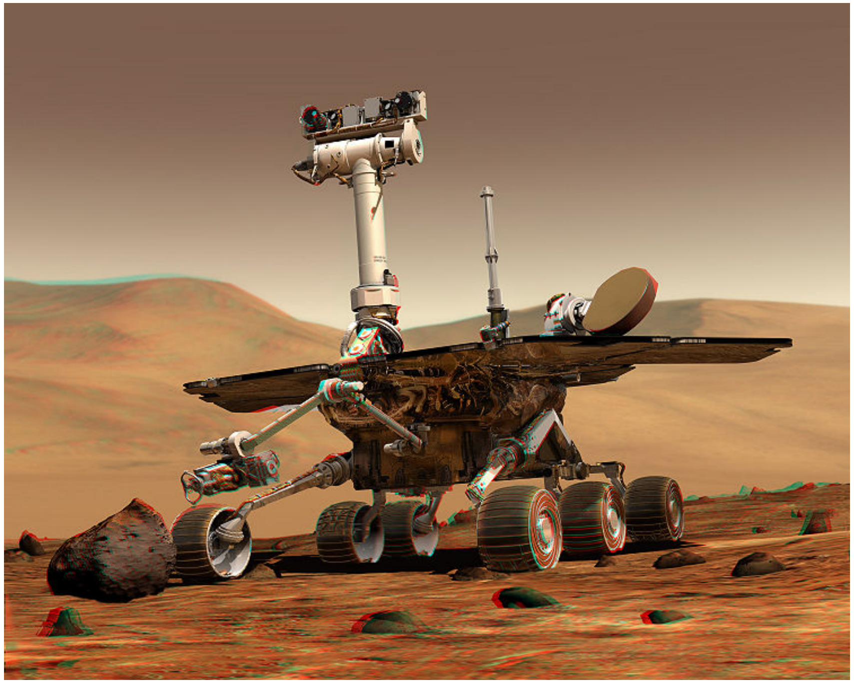




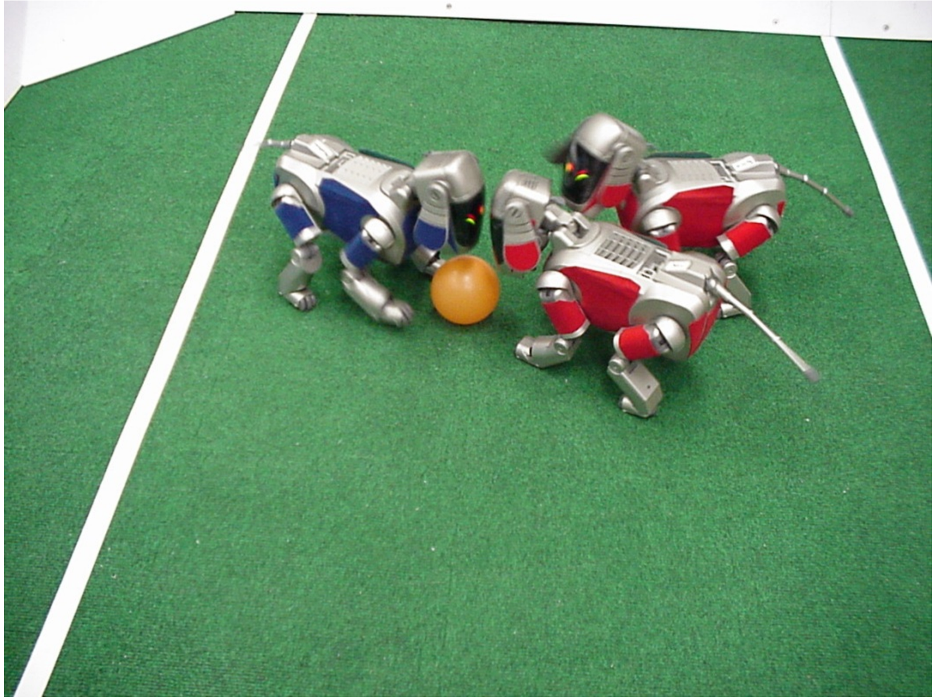
small tethered rover

pan/tilt stereo head

Robotics



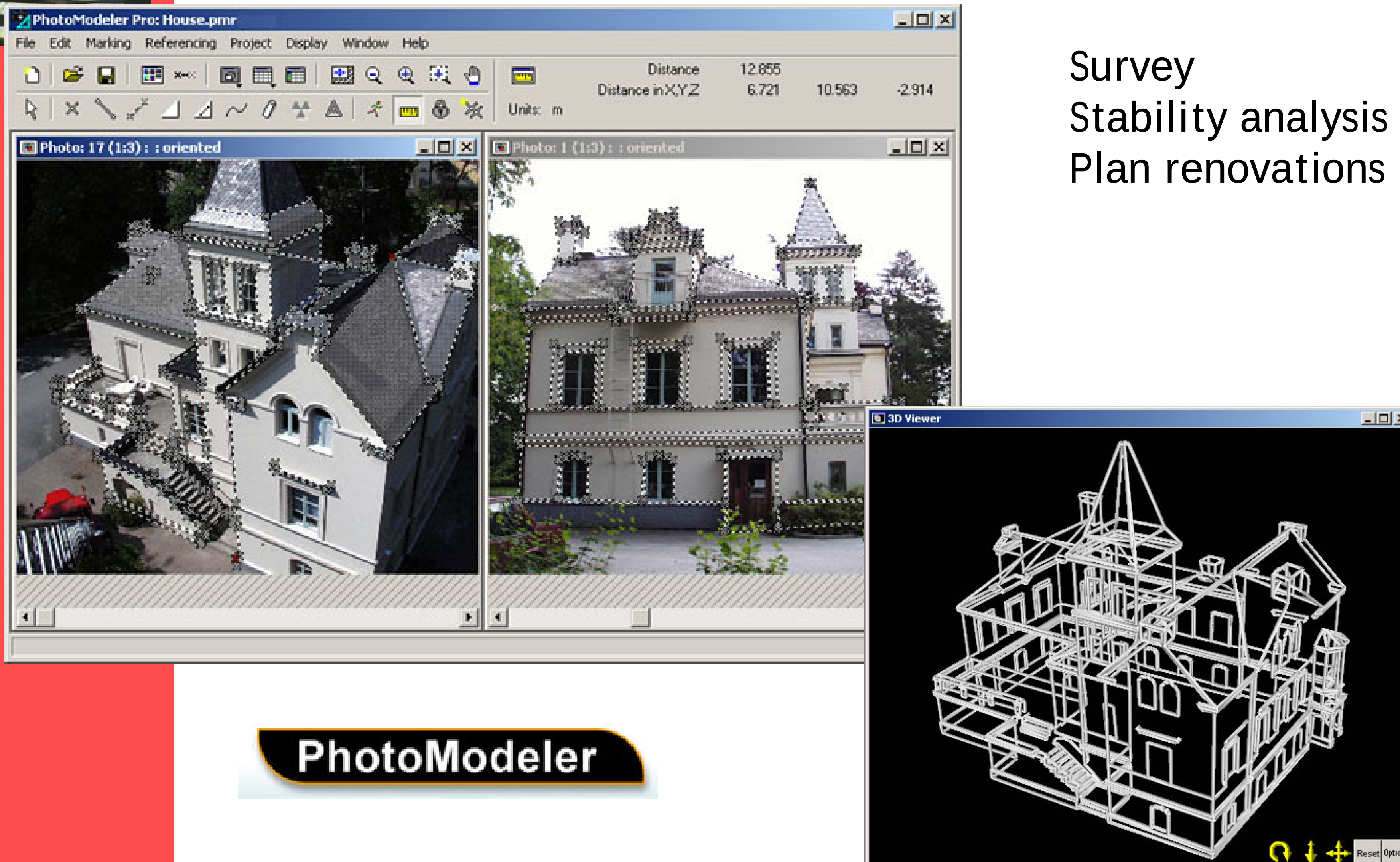
NASA's Mars Spirit Rover
http://en.wikipedia.org/wiki/Spirit_rover



http://www.robocup.org/

https://www.youtube.com/watch?v=GuXLvrBrO3s

Architecture



Survey
Stability analysis
Plan renovations

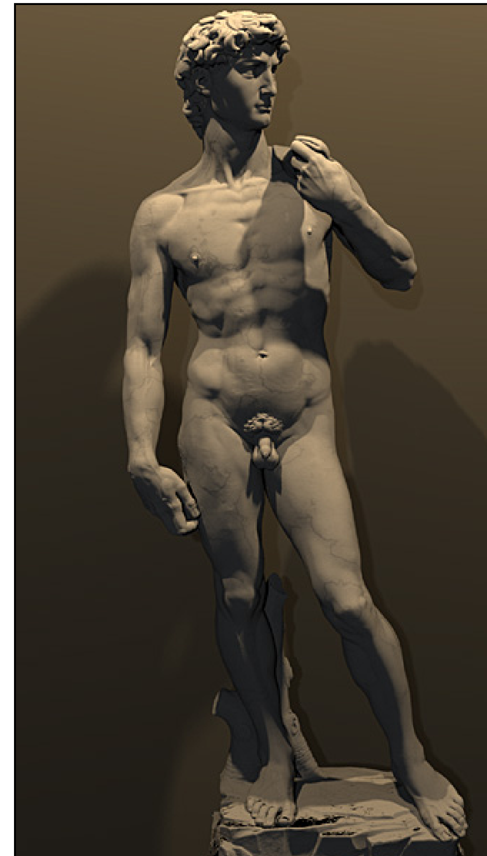
PhotoModeler

Architecture

Survey
Stability analysis
Plan renovations

Cultural heritage

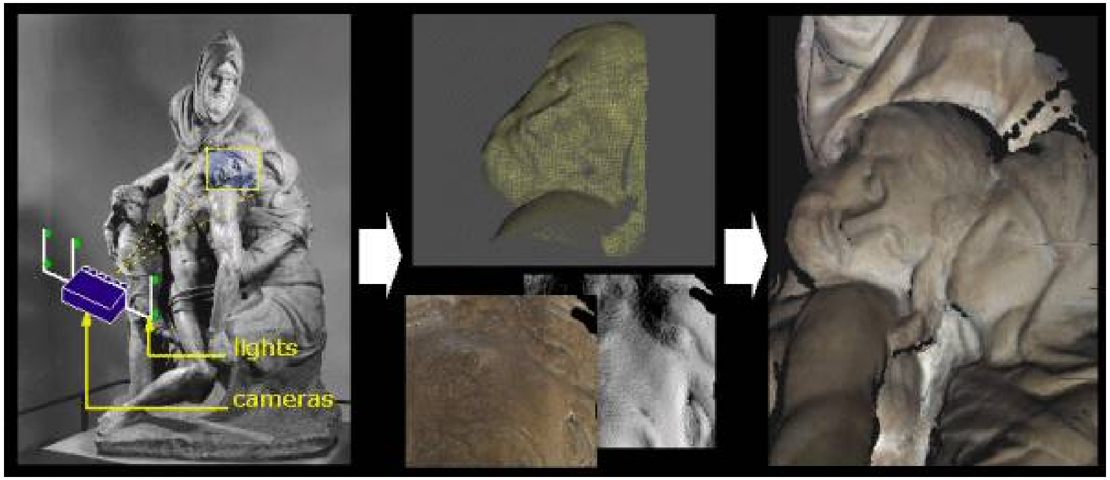
Stanford's Digital Michelangelo



Digital archive
Art historic studies

IBM's pieta project

Photometric stereo + structured light



more info:

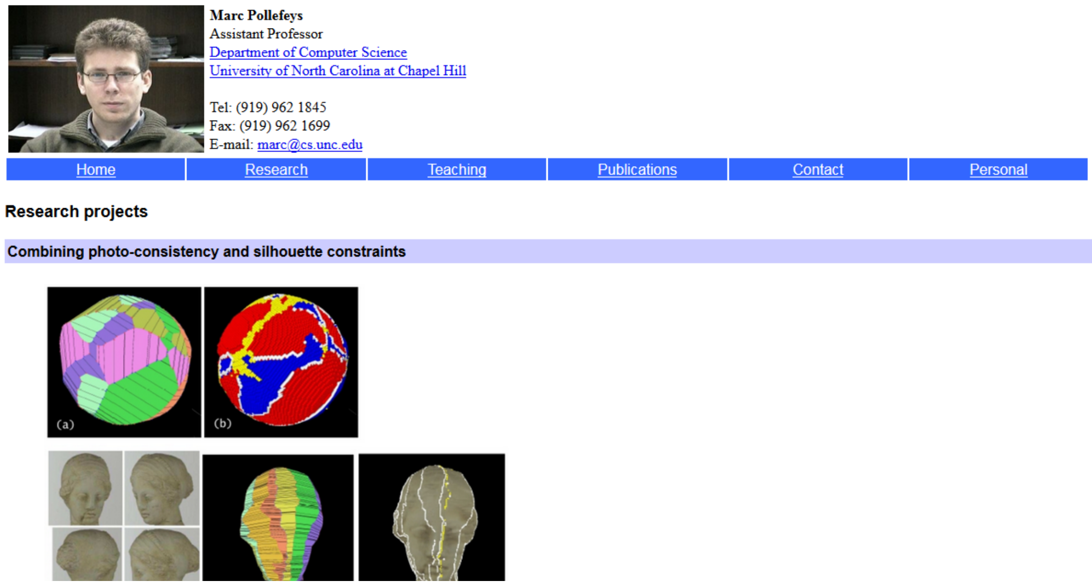
http://www.research.ibm.com/vgc/pdf/pieta-cga.pdf

Archaeology

accuracy ~1/500 from DV video
(i.e. 140kb jpegs 576x720)

link

Examples of Research Projects



http://www.cs.unc.edu/~marc/research.html

Visual Cues:
Stereo and Motion

Disparity map from Stereo

image I(x,y)

Disparity map D(x,y)

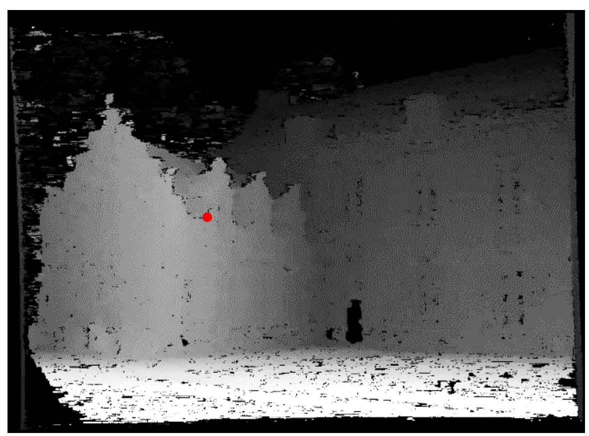
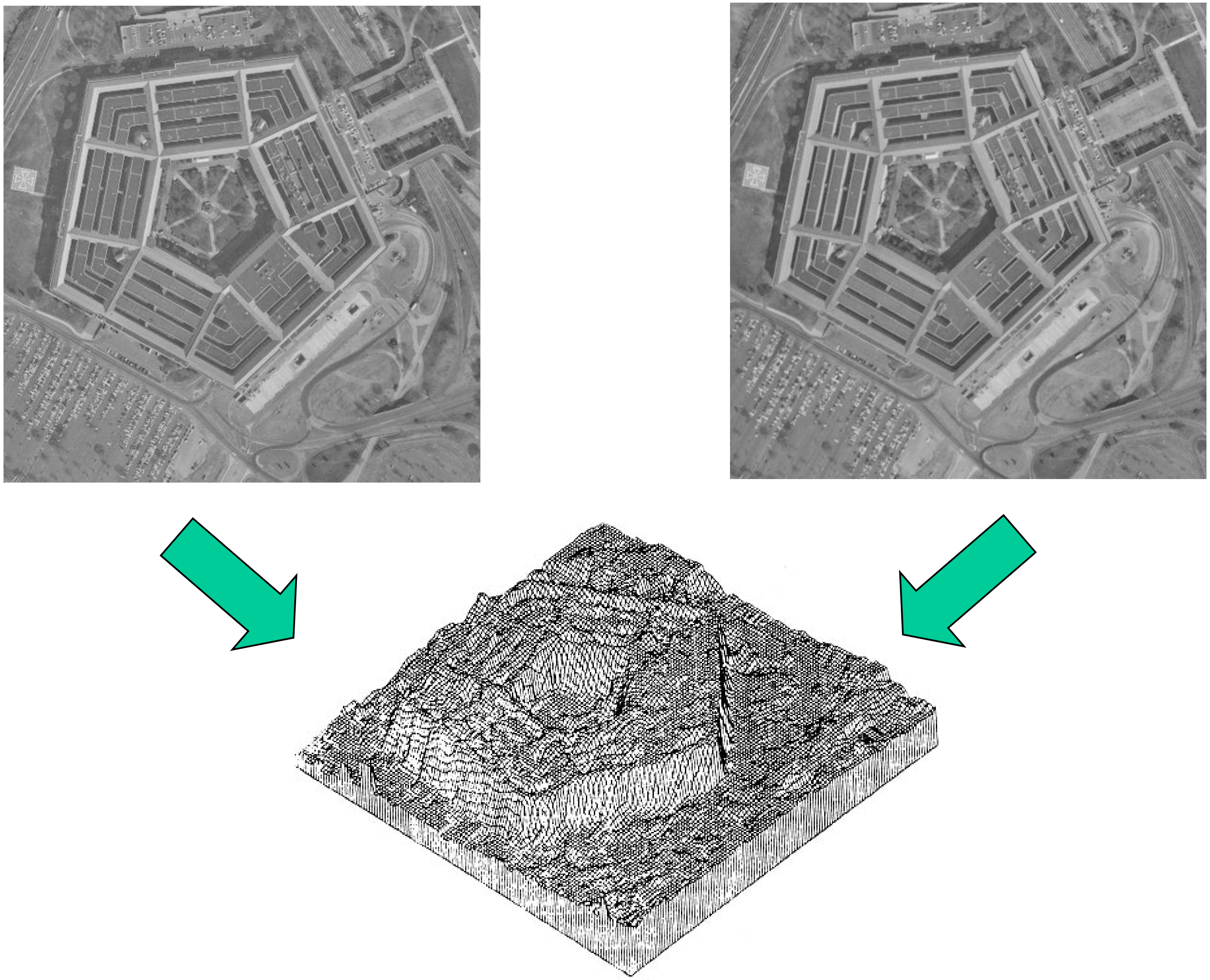


image I´(x´,y´)

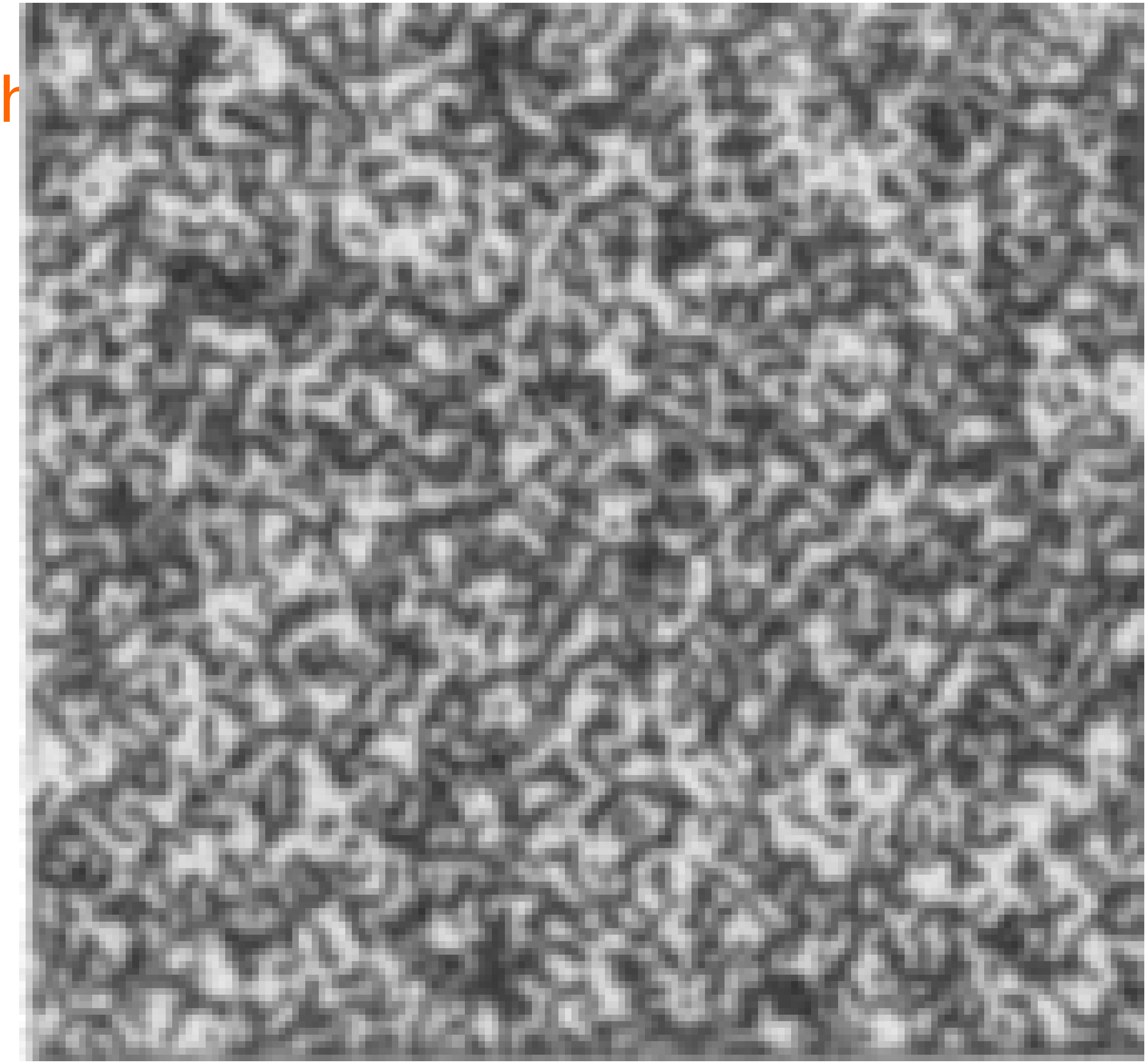
$$(x',y')=(x+D(x,y),y)$$

Dynamic Programming (Ohta and Kanade, 1985)



Optical flow

Wh

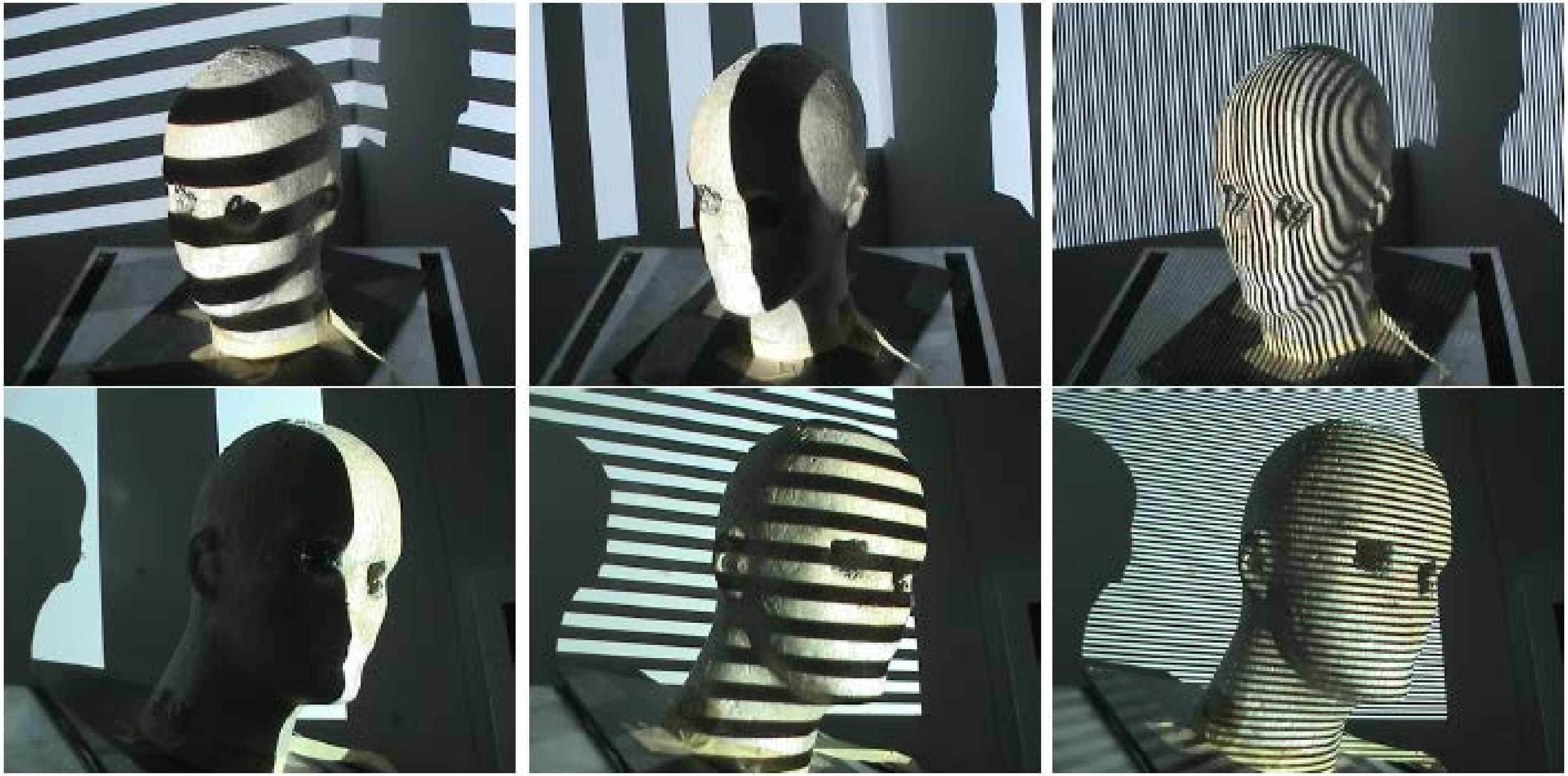


Optical flow

Results

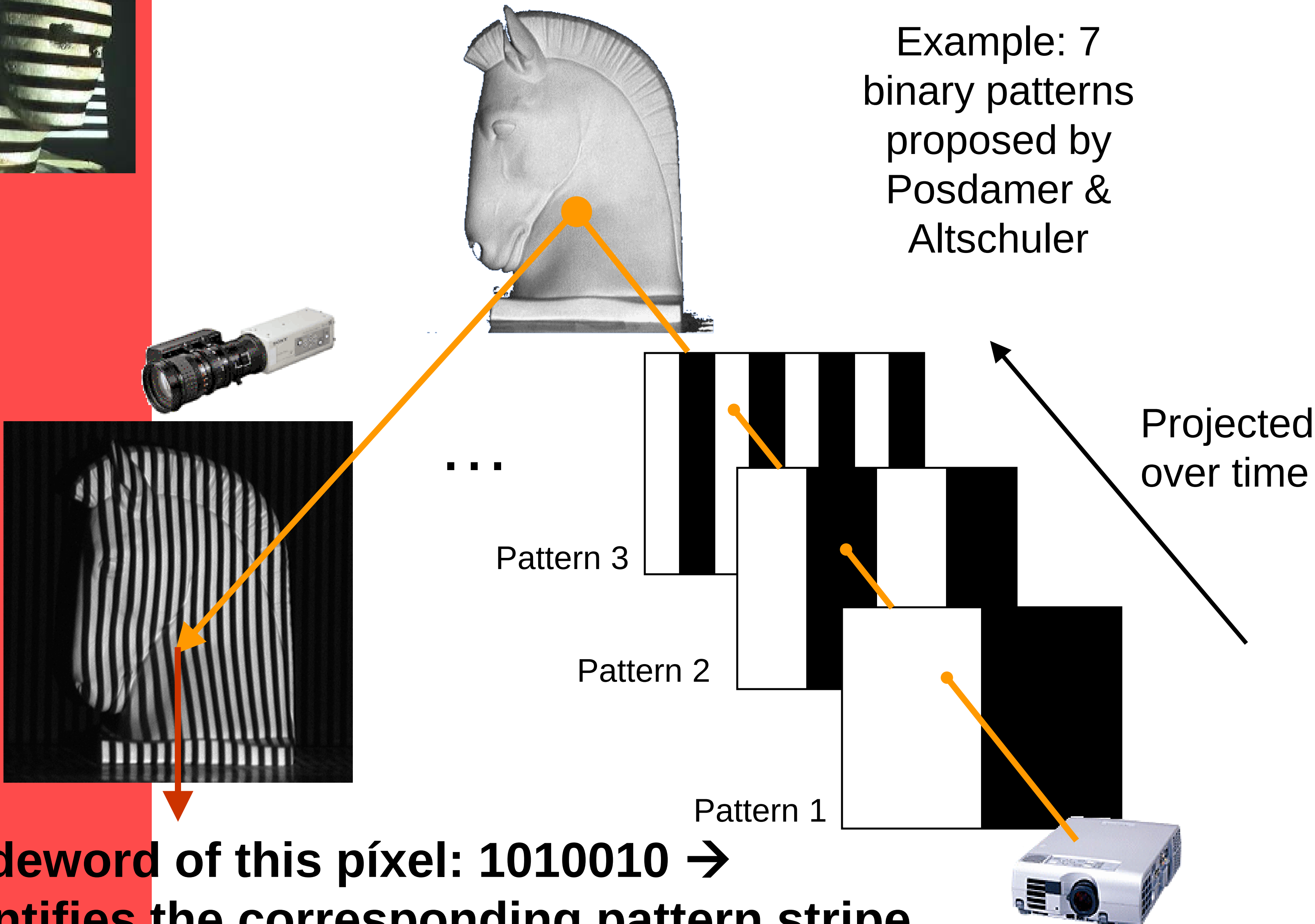
Active Vision:
Structured Light

Active Vision: Structured Light



Binary Coding

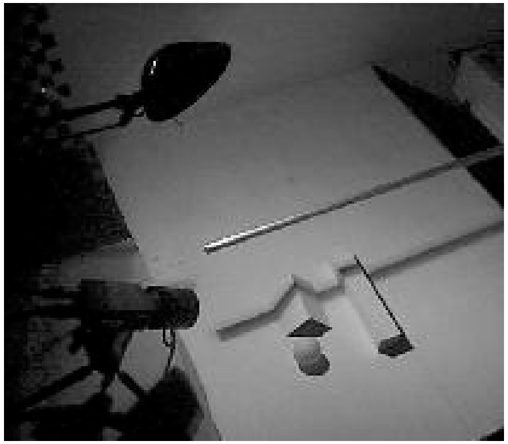
Example: 7 binary patterns proposed by Posdamer & Altschuler



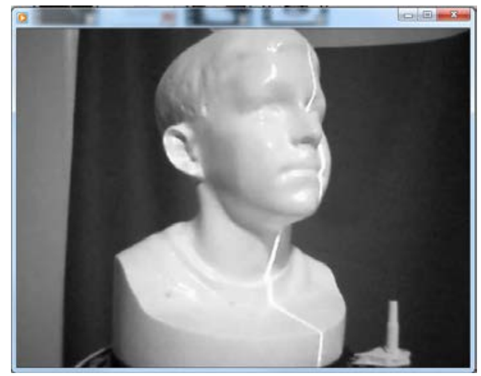
Codeword of this píxel: 1010010 → identifies the corresponding pattern stripe

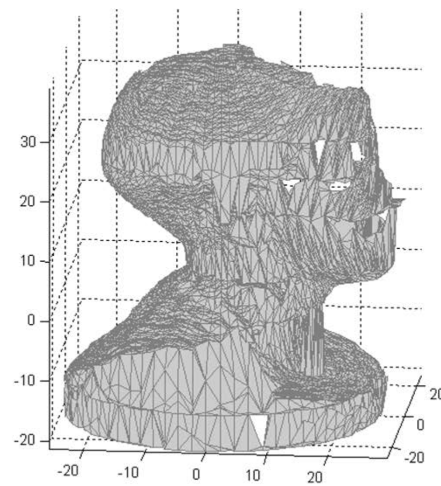
“Cheap and smart” Solution

Example: Bouguet and Perona, ICCV’98

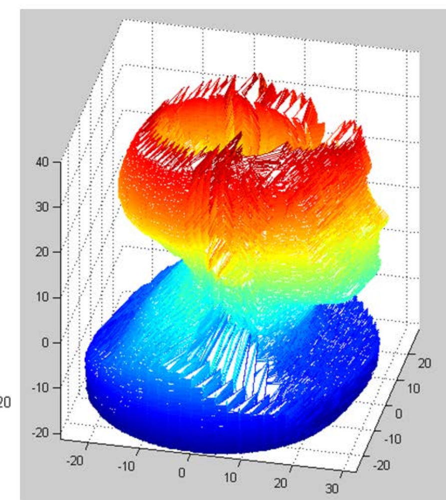


Structured Light Using a Rotating Table
James Clark, 3D CV F2009



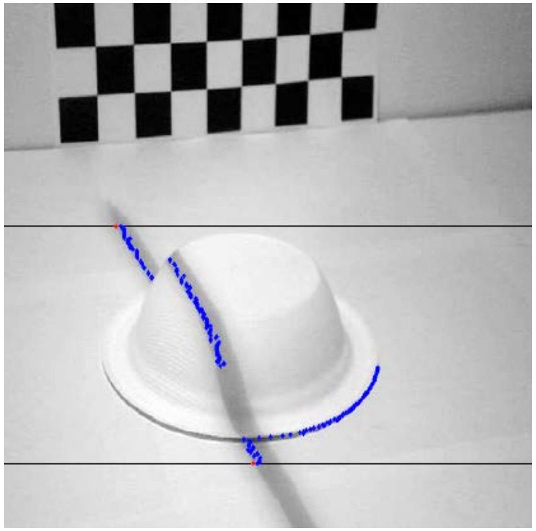


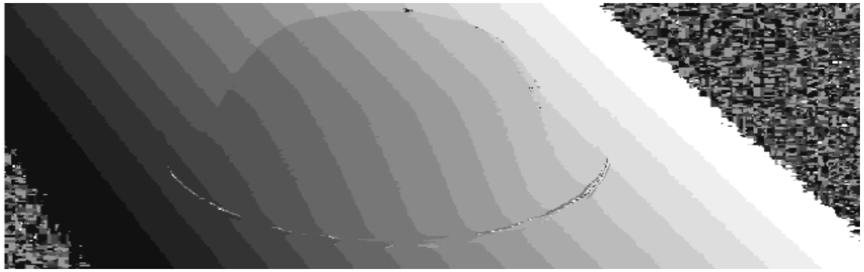
Height Strip Mesh



Localized Mesh

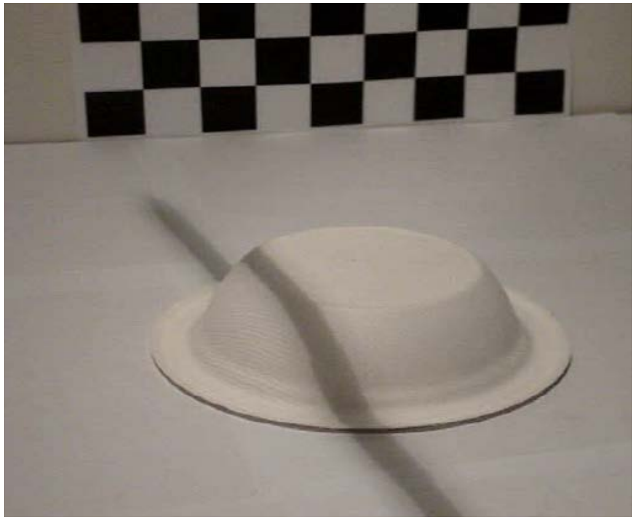
Structured Light
Anuja Sharma, Abishek Kumar

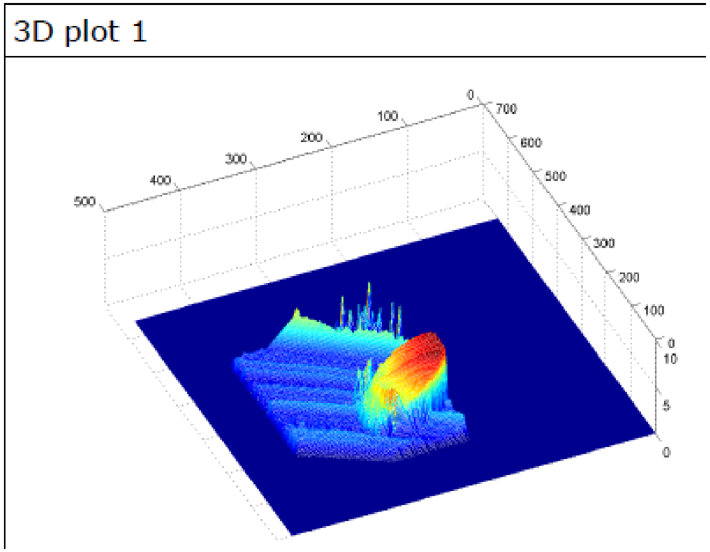




Structured Light

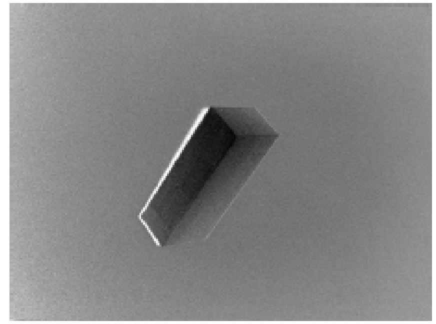
Anuja Sharma, Abishek Kumar

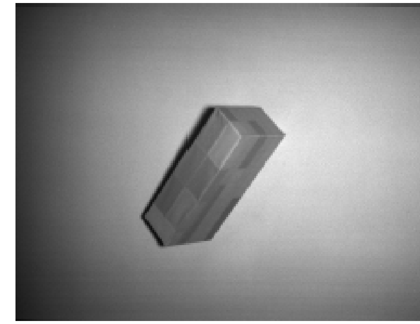




Range Sensor Data Processing to get 3D Shapes

Input Data: Depth Maps

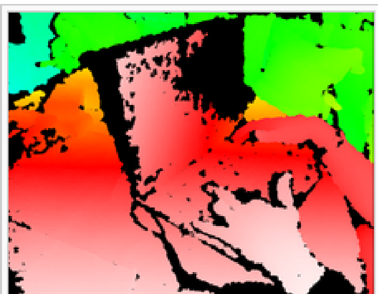




Range Image (left) and gray level image (right)

A slide from Microsoft's E3 Conference showing a diagram of the technologies in Kinect

This infrared image shows the laser grid Kinect uses to calculate depth



The depth map is visualized here using color gradients from white (near) to blue

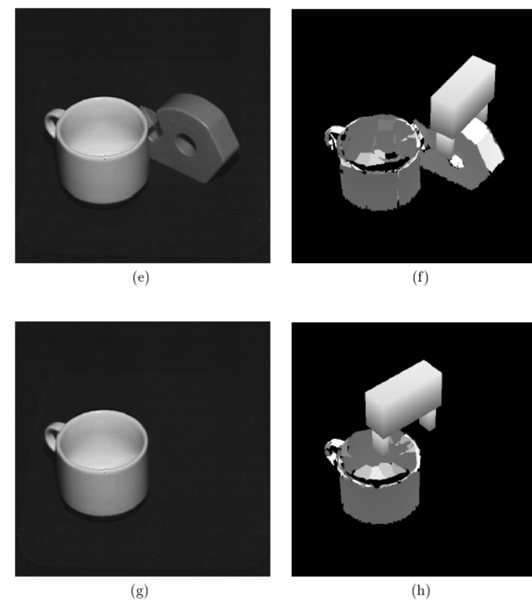
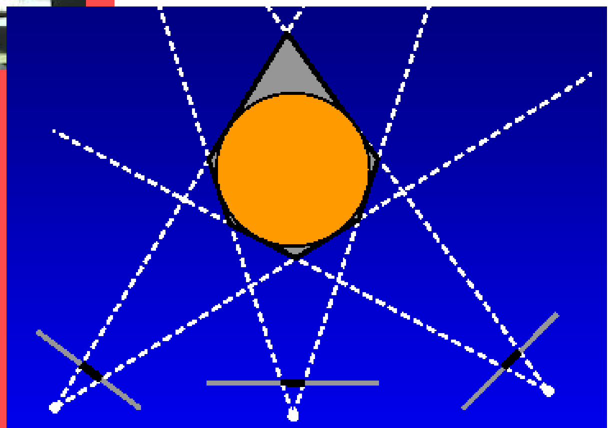
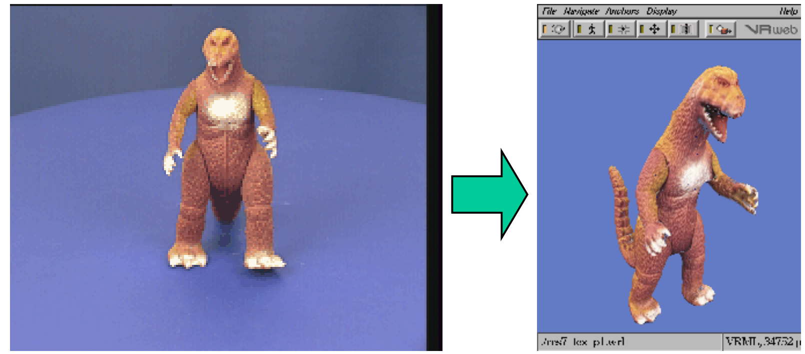


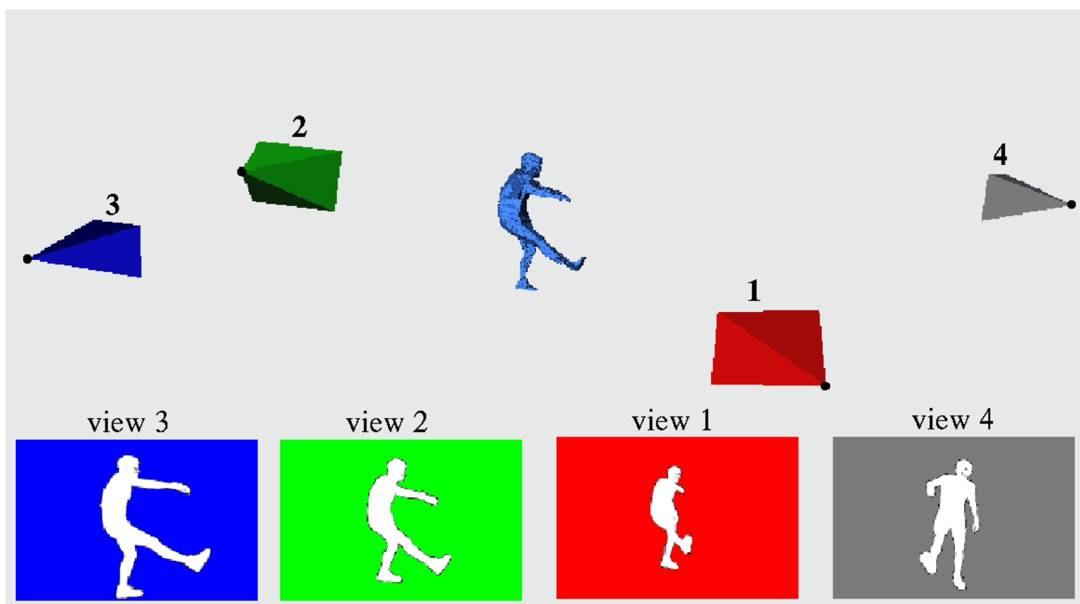
Figure 9: Continuation of the example scene consisting of four objects. (e) and (f) grasping the Scotch tape roller, and (g) and (h) grasping the coffee cup.

3D Shape Cues:
Shape from Silhouettes

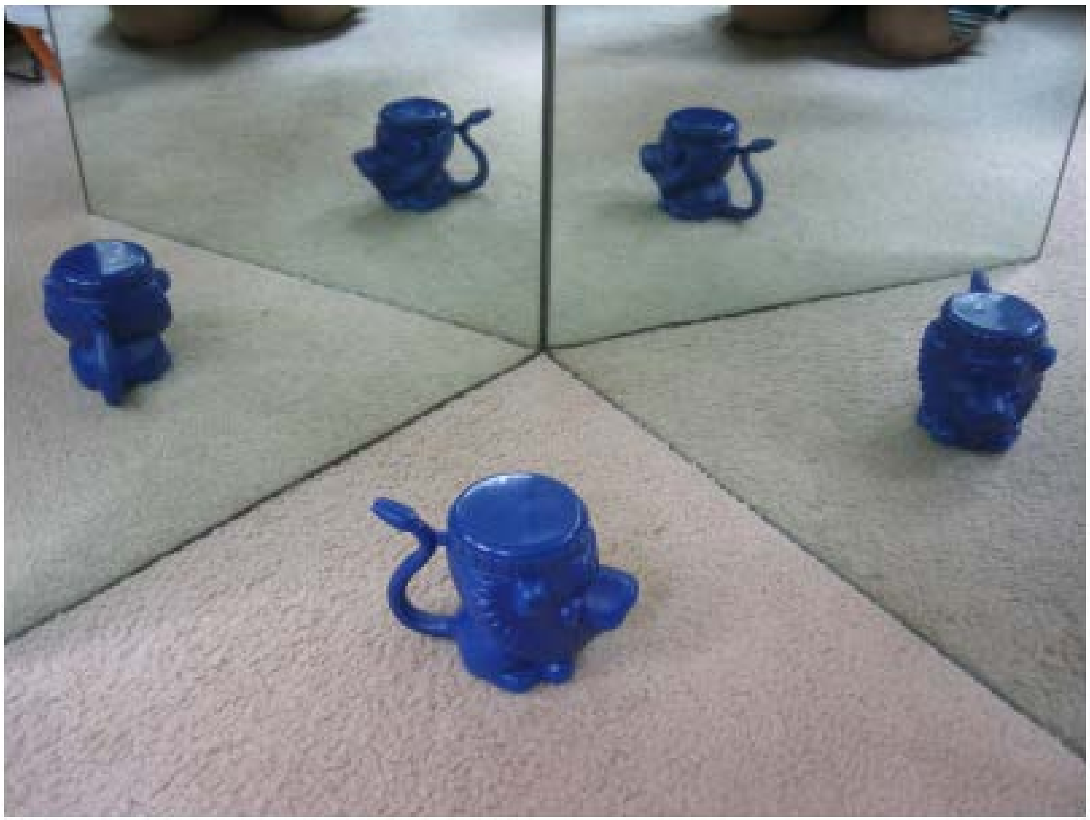
3D Shape from Silhouettes







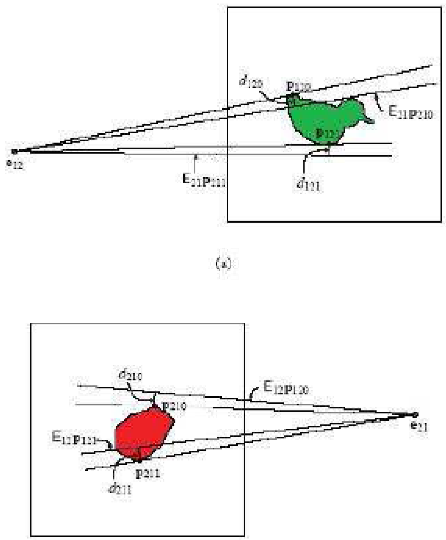
3D shape from silhouettes: Two Mirrors and uncalibrated camera



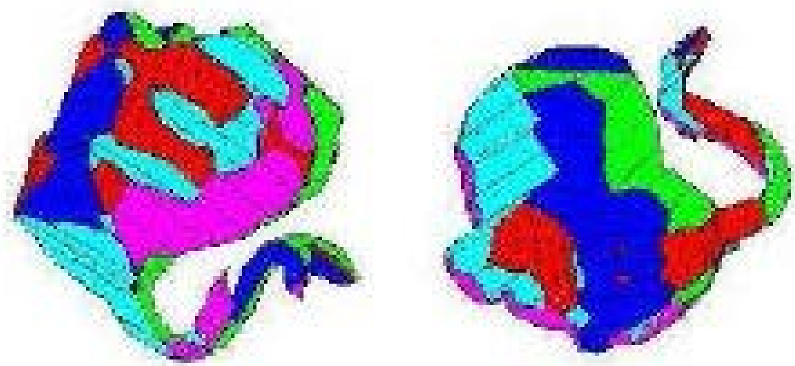
Forbes et al., ICCV2005

Christine Xu, Computer Vision Student Project

3D shape from silhouettes



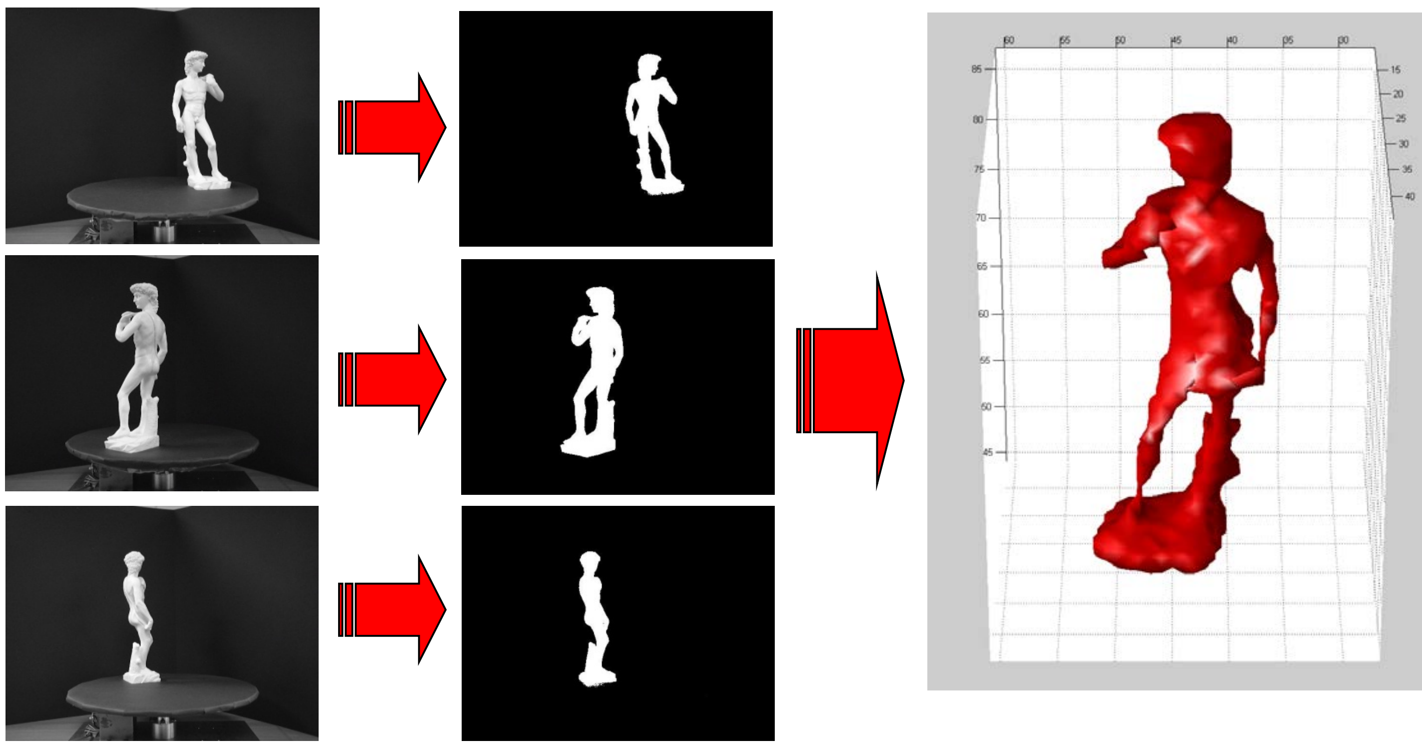
Build 3D model



Visualize 3D model from arbitrary viewing angles

Example

- Compute visual hull with silhouette images from multiple calibrated cameras
- Compute Silhouette Image
- Volumetric visual hull computation
- Display the result



Shape from Shading

Photometric Stereo
Christopher Bireley

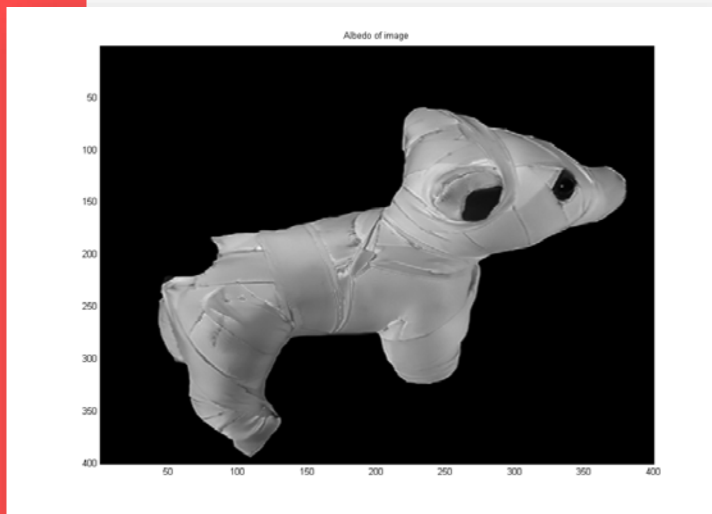
Bandage Dog

Imaging Setup

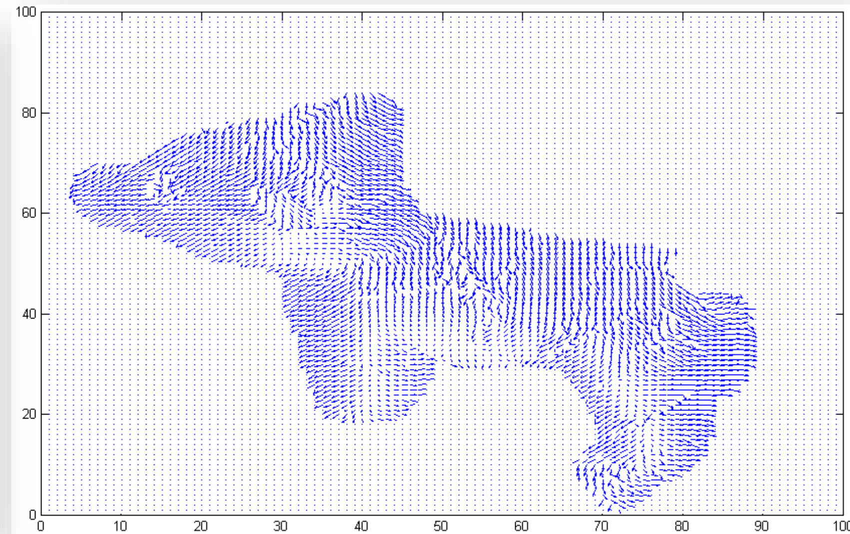
Preprocessing

- Remove background to isolate dog
- Filter with NL Means

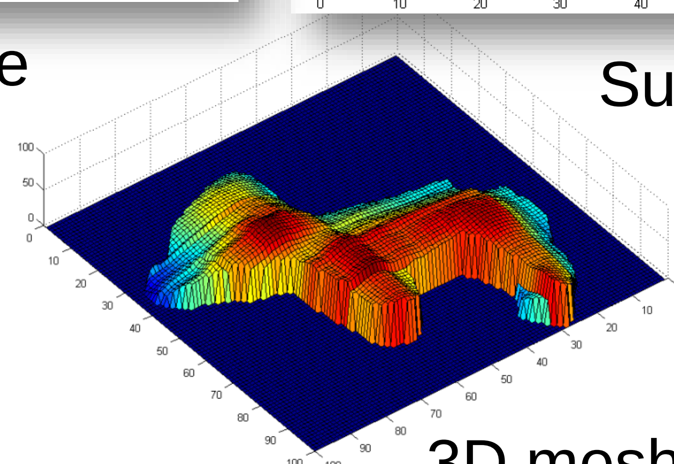
Photometric Stereo
Christopher Bireley



Albedo image



Surface Normals

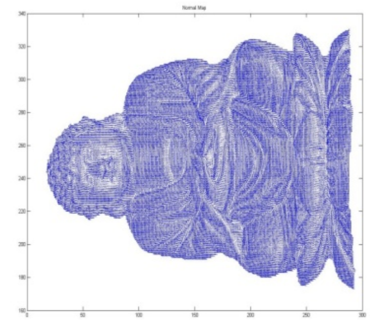


3D mesh

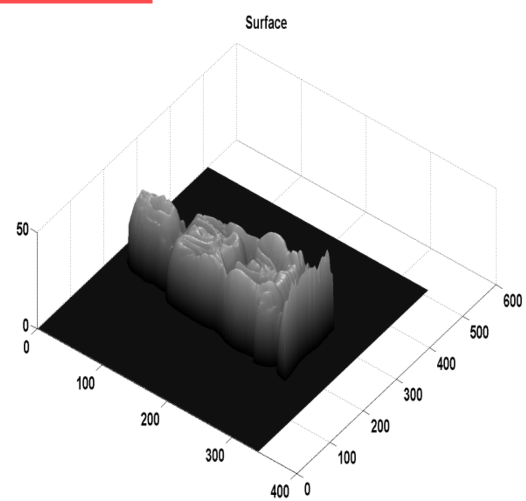
Results – Lord Buddha Images – Pre-Processed Images Guozhen Fan and Aman Shah

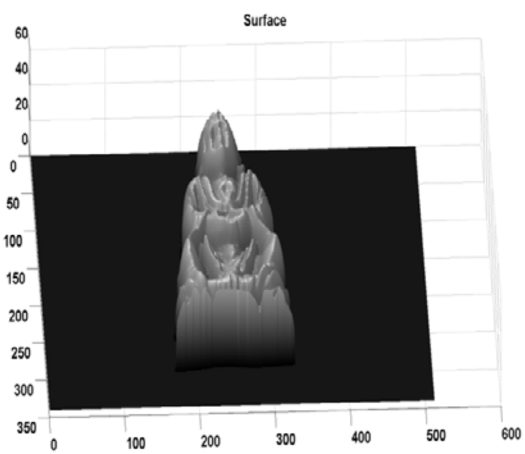
Original Image

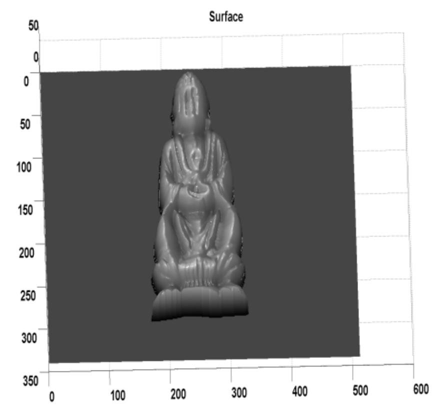
Albedo Map



Surface Normals



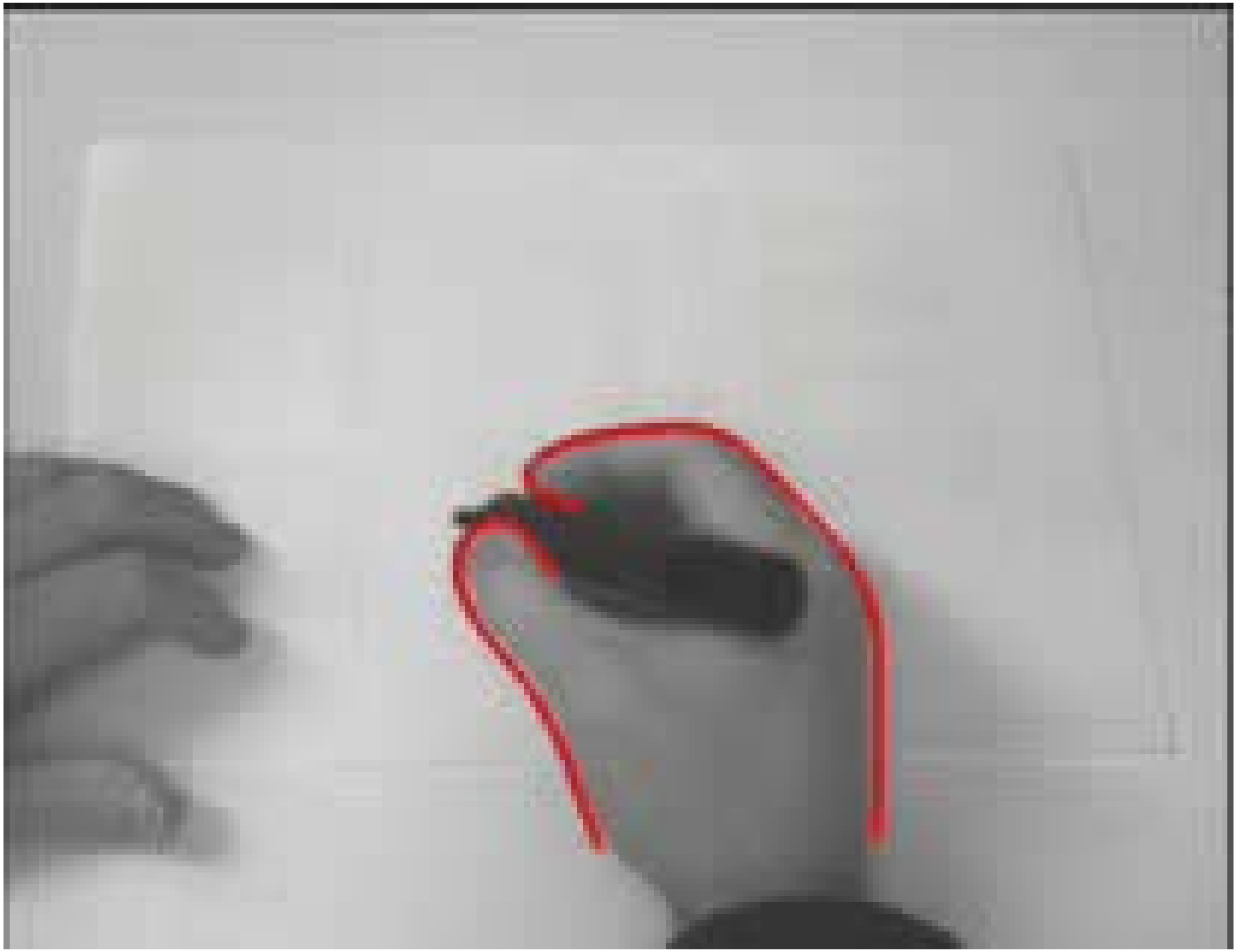




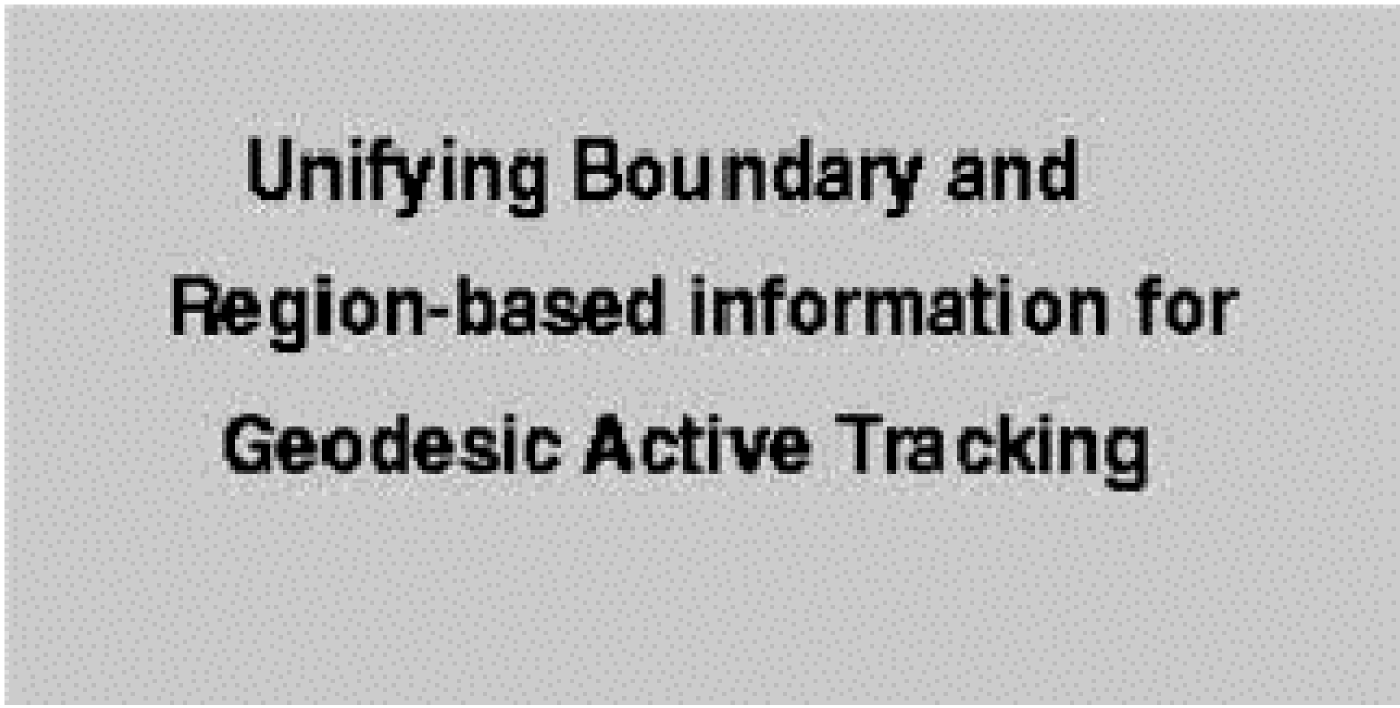
Obtained Surfaces from different angles

Object Tracking

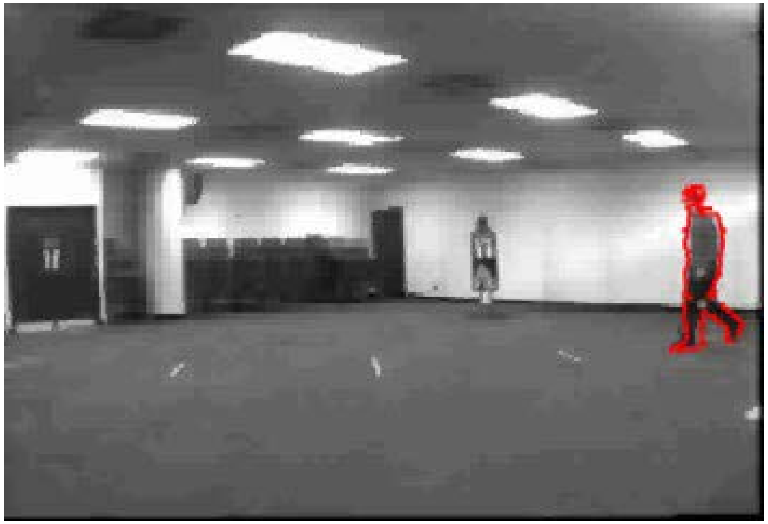
Object Tracking: Using Deformable Models in Vision



Object Tracking: Using Deformable Models in Vision: II



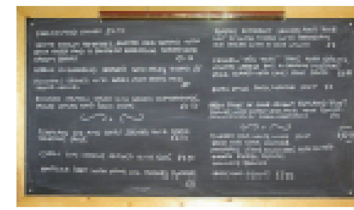
Object Tracking III

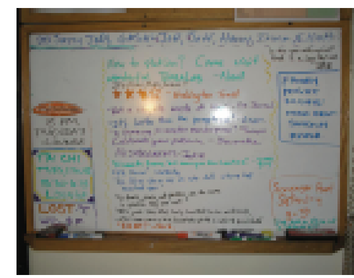


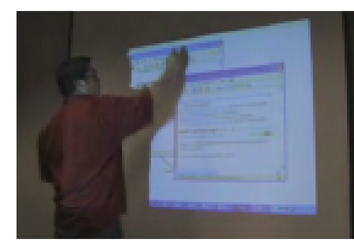
Computer Vision Systems

Webcam Based Virtual Whiteboard Student Project Jon Bronson James Fishbaugh

- Blackboards came first
- Whiteboards eventually followed
- Virtual Whiteboards are coming
- Basic Idea:
  - Write on any surface
  - Use no ink/chalk
  - Store all information to disk

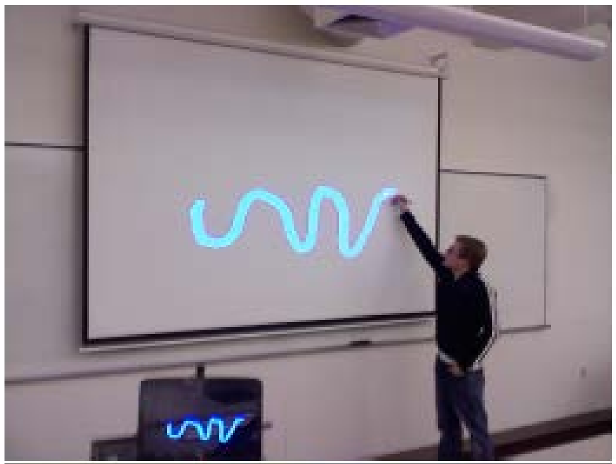




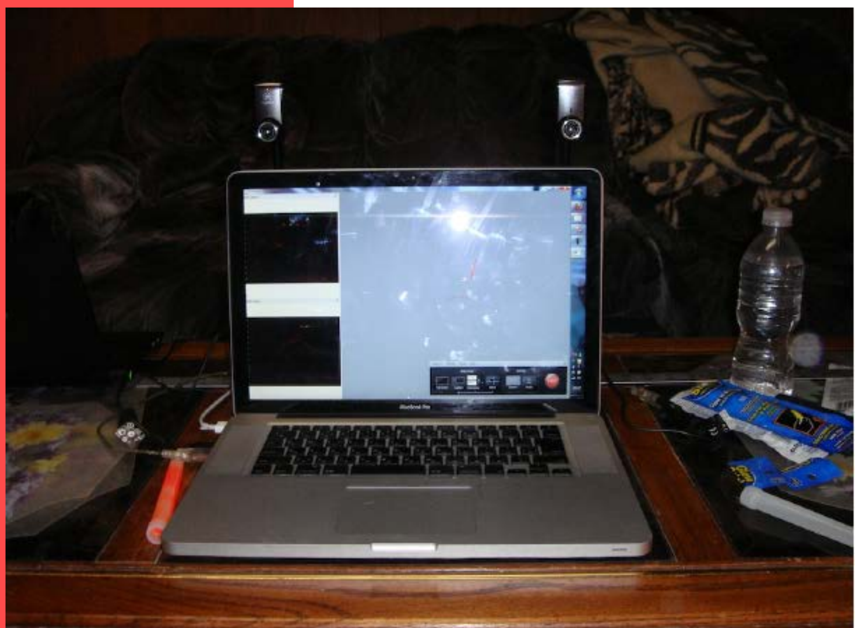


Webcam Based Virtual Whiteboard

Jon Bronson James Fishbaugh



Real-Time 3D Glowstick Detection
Computer Vision Project 2009
Andrei Ostanin



Detecting the 3D position of glowsticks in real-time using two cameras.

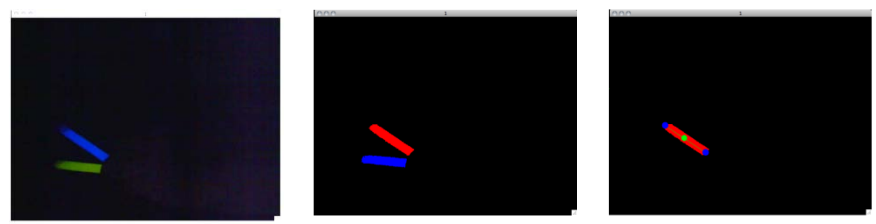
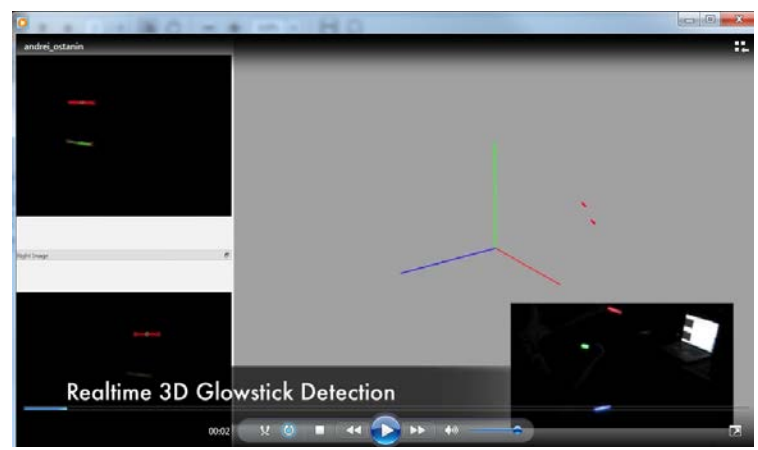
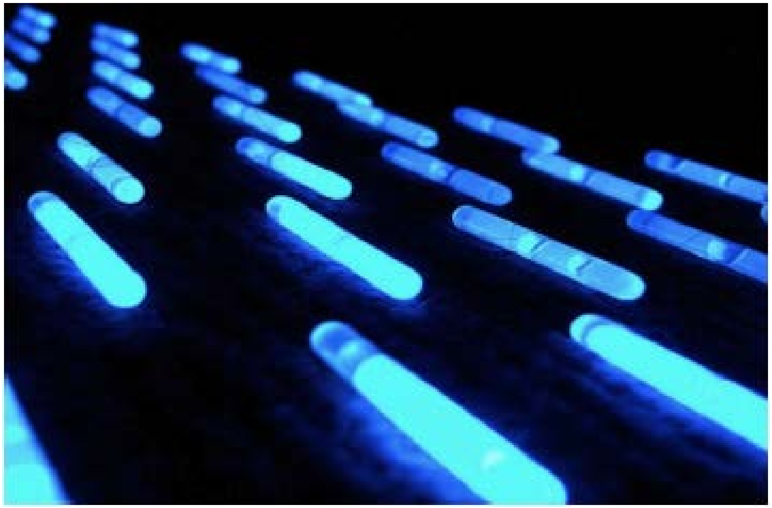
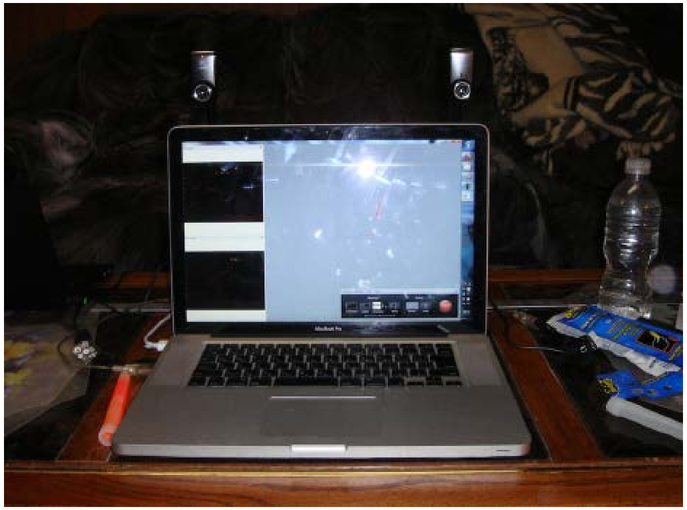


Figure 2: Camera input images.

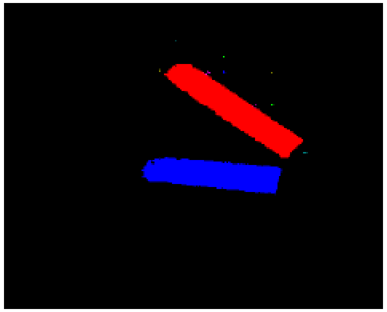


Realtime Glowstick Detection
Andrei Ostanin





- Capture the 3D position of glowsticks in real-time using two webcams
- Environment dark enough that glowsticks are easily segmented out
- Prefer speed over correctness

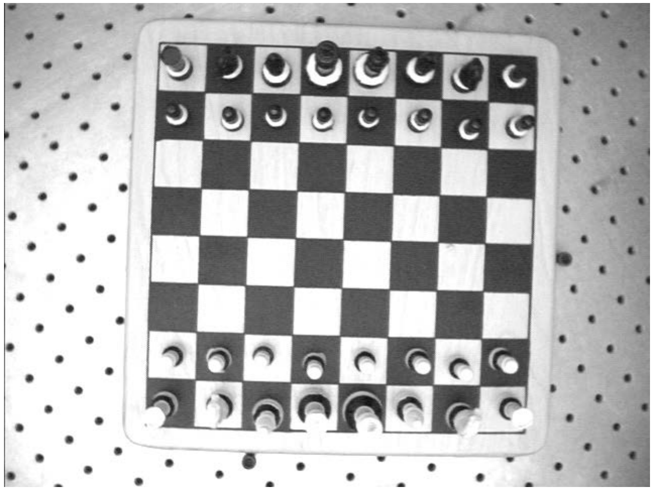


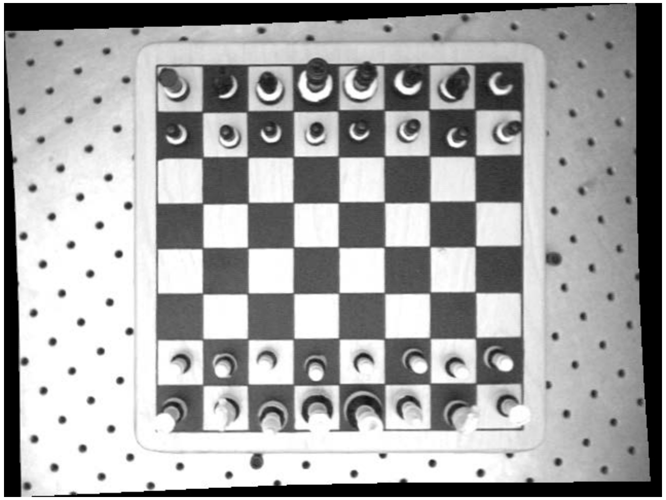
movie

Student Project: Playing Chess, Recognition and Simulation

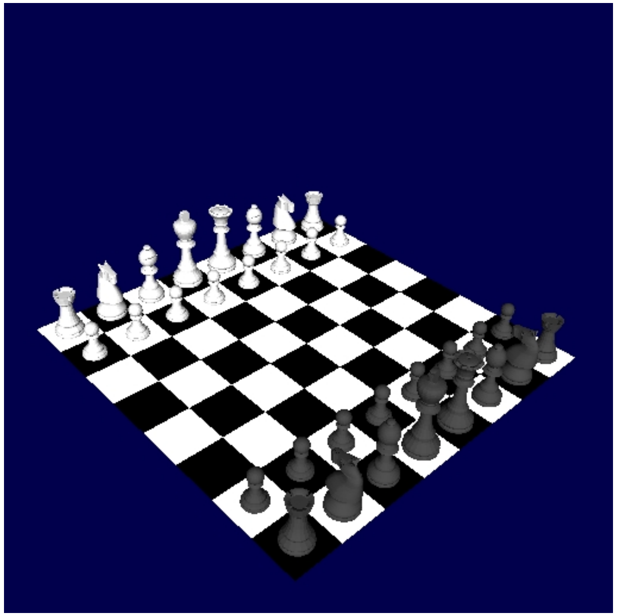
- Track individual chess pieces
- Maintain state of board
- Graphically represent state changes and state
- D. Allen, D. McLaurin UNC
- Major ideas:
  - 3D from stereo
  - detect and describe changes
  - Use world knowledge (chess)

Calibration, Rendering & Replay





Movie



Goal and objectives

From Snapshots, a 3-D View

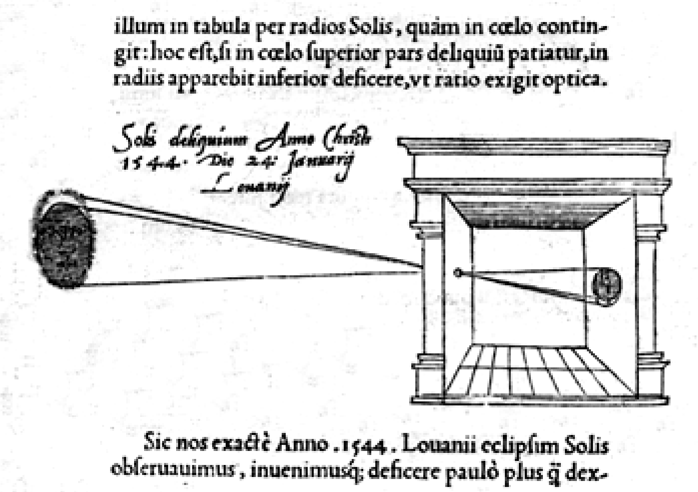
NYT, August 21, 2008, Personal Tech

http://www.nytimes.com/2008/08/21/technology/personaltech/21pogue.html

Stuart Goldenberg

Next: Image Formation
Chapter 1: Cameras

- Please find pdf copies of Chapters 1&2, Forsyth&Ponce, on the website.
- Purchase the course book on your own.



Assignment:

- Read Chapter 1: Cameras, Lenses and Sensors: See Course [home page]