Fundamentals of Computer Vision: Introduction

CMSC 828D

Larry Davis, Ramani Duraiswami,
Daniel DeMenthon, Yiannis Aloimonos

Introduction

- Policies and other administrativia
- What is this course about?
- Problems you should be able to solve at the end of the course
- Introduction to Matlab
- Homework 1

Tag Team Instruction ☺

- Larry Davis (lsd@umiacs.umd.edu)
- Ramani Duraiswami (ramani@umiacs.umd.edu) 301 405 6710
- Daniel DeMenthon (daniel@cfar.umd.edu) 301 405 6565
- Yiannis Aloimonos (yiannis@cfar.umd.edu)
- Office Hours (by appointment)

Course Policies

- Exams
  - Exam 1 (25%) November 1, 2000
  - Exam 2 (25%) handed out Dec 11. Due back Dec. 13
- Homework (50%)
  - Handed out last class of current week
  - Due first class of next week (almost no exceptions)
  - Late homework will not be graded
- Attendance/homework submission mandatory
- Counts as a Ph.D. qualifying course

Texts

1. *A Guided Tour of Computer Vision, by V. S. Nalwa, Addison-Wesley, 1993.*
2. Introductory Techniques for 3-D Computer Vision, by Emanuele Trucco, Alessandro Verri, Prentice-Hall, 1998
3. *Multiple View Geometry, by Richard Hartley, Andrew Zisserman, Cambridge University Press, 2000.*
4. *Numerical Recipes in C, by William Press et al., Cambridge Univ Press, 1992.*
5. *Pattern Classification and Scene Analysis, by Richard O. Duda, Peter E. Hart, John Wiley & Sons, 1973.*

Web

- Computer Vision home page
  - http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html
- CV Online texts
  - http://www.dai.ed.ac.uk/CVonline/
- Many others … any good ones please provide them to instructors
- Course home page
  http://www.umiacs.umd.edu/~ramani/cmsc828.html
  - Lecture Slides
  - Homework
  - Solutions (?)

Course outline

- Computer Vision
  - Focus is on basic computer vision skills
  - Tools for research
- Mathematical Background
  - Basic material
- Numerical/computational Skills
- Matlab
  - Quick prototyping
  - tool for checking out ideas

Computer Vision

- Computer vision basics
  - Image creation
  - Cameras, Eyes, Calibration
  - Features, correspondence
  - 3D vision
  - Optical Flow
  - Tracking
  - Compression, vision for content delivery

Mathematics

- Basic Mathematical tools
  - Linear systems, matrix decompositions
  - Vector space, Basis, eigenvalues, eigenvectors
  - Calculus: gradients, Taylor series, maxima
- Projective geometry
  - Homographies, projectivities, constraints
  - Fundamental matrix
  - Trifocal tensor
- Probability, Random Variables, Classification

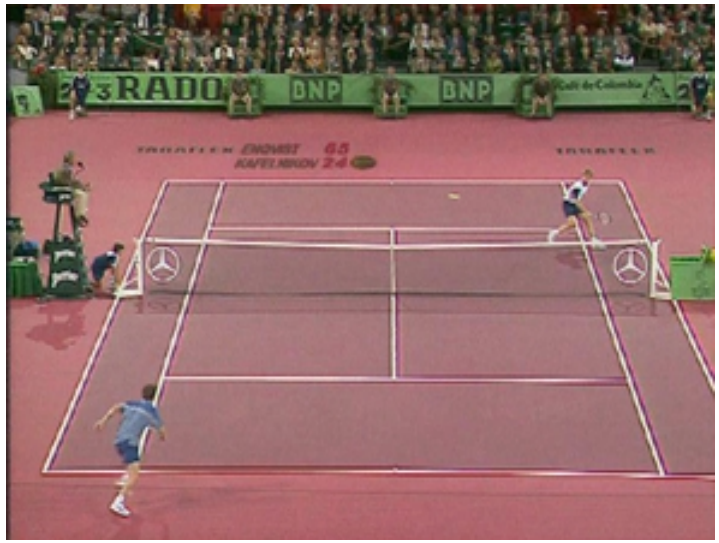
Numerical Methods

- Solving linear systems of equations
- Matrix decompositions
  - LU, Eigenvalue, SVD,
- Optimization
- Use Matlab as a tool to quickly try ideas
- Familiarity with image types
- “Practical computer vision”

Problems we should be able to solve

- Calibrate a camera
- Rectify an image
- Create a three dimensional reconstruction
- Insert model into image
- Track objects
- Track people
- Read papers on these topics and do thesis research
- Recommended course CMSC 828C
  - Computer vision seminar (for people doing research in vision)

Detect ground plane in video and introduce pictures on them



http://www.symah-vision.fr/vhtml/epsis.htm

Calibrate video sequences and insert new objects into them

Tracking

What is vision?

- Recognize objects
  - people we know
  - things we own
- Locate objects in space
  - to pick them up
- Track objects in motion
  - catching a baseball
  - avoiding collisions with cars on the road
- Recognize actions
  - walking, running, pushing

Vision is

- Deceivingly easy
- Deceptive
- Computationally demanding
- Critical to many applications

Vision is deceivingly easy

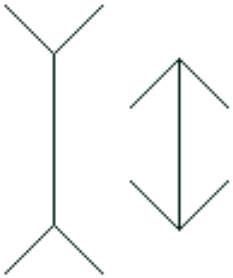
- We see effortlessly
  - seeing seems simpler than “thinking”
  - we can all “see” but only select gifted people can solve “hard” problems like chess
  - we use nearly 70% of our brains for visual perception!
- All “creatures” see
  - frogs “see”
  - birds “see”
  - snakes “see”

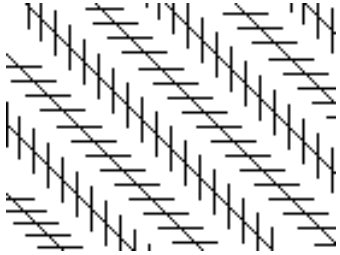
but they do not see alike

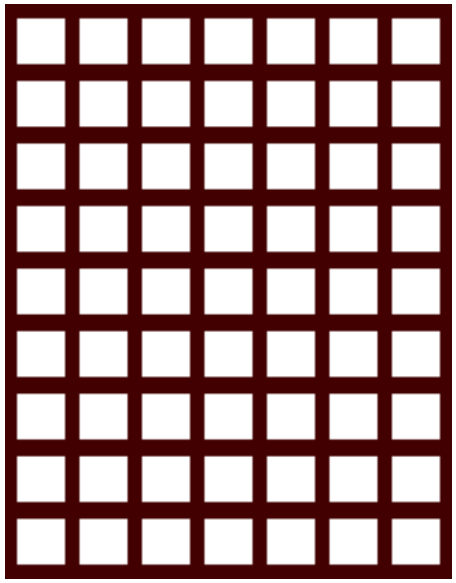
Vision is deceptive

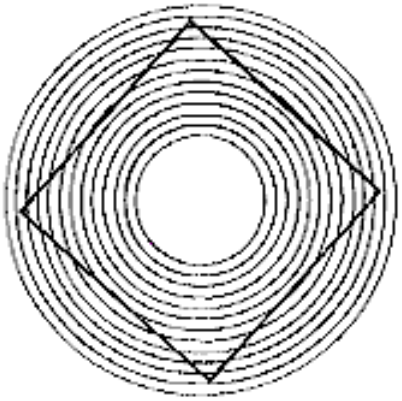
- Vision is an exceptionally strong sensation
  - vision is immediate
  - we perceive the visual world as external to ourselves, but it is a reconstruction within our brains
  - we regard how we see as reflecting the world “as it is;” but human vision is
    - subject to illusions
    - quantitatively imprecise
    - limited to a narrow range of frequencies of radiation
    - passive

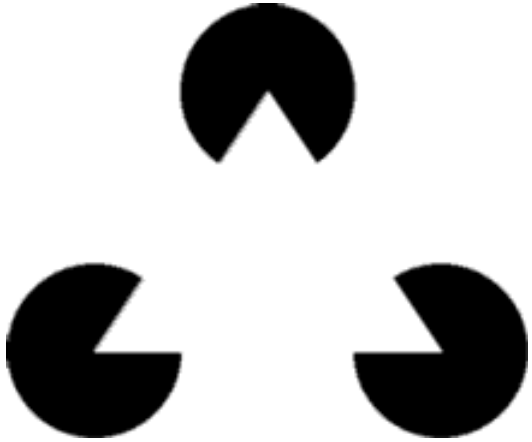
Some Illusions



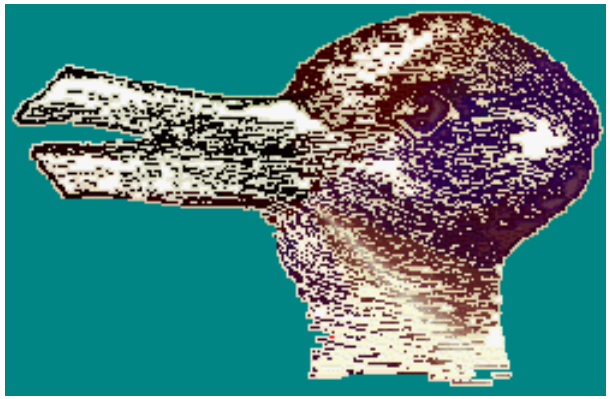


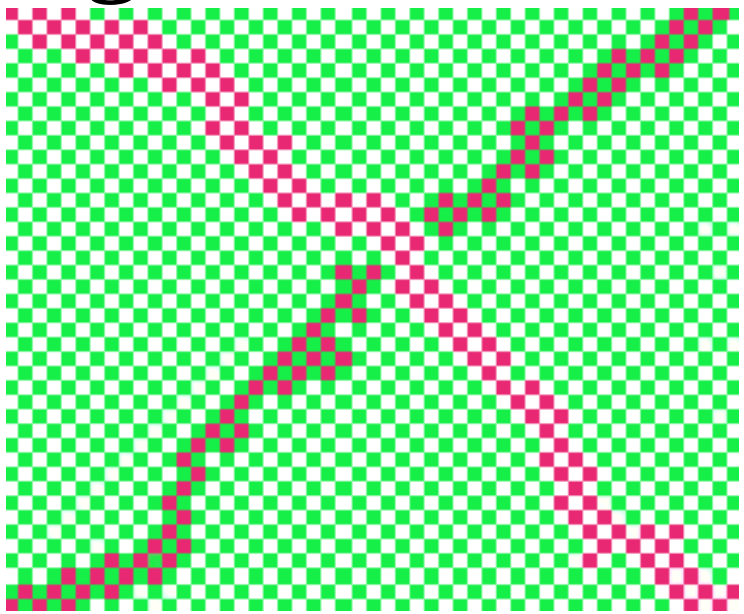


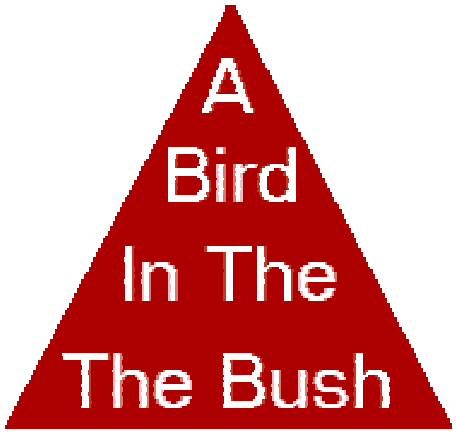




Higher illusions







Human vision is passive

- It relies on external energy sources (sunlight, light bulbs, fires) providing light that reflects off of objects to our eyes
- Vision systems can be “active” - carry their own energy sources
  - Radars
  - Bat acoustic imaging systems

Spectral limitations of human vision

- We “see” only a small part of the energy spectrum of sunlight
  - we don’t see ultraviolet or lower frequencies of light
  - we don’t see infrared or higher frequencies of light
  - we see less than .1% of the energy that reaches our eyes
- But objects in the world reflect and emit energy in these and other parts of the spectrum

Matlab

- Started out as “Matrix laboratory” a piece of freeware (80s)
- Written by mathematicians
- Integrated programming environment
- Includes a rich set of features
- Very good online help
  - Within program
  - On pdf and html files on disk
  - On the web
  - On newsgroups

Matlab

- Using Matlab allows programming without the need to do routine tasks
  - E.g., Reading and writing image files
- However Matlab is
  - Interpreted and hence slow
  - Does not support all data types and hence is a memory hog
  - natural data type is a double instead of a 8 bit uint
- Does not utilize multiprocessors
- **Prototyping Tool**

Matlab

- Vectorization can make code faster
  - But code then loses readability
- Some defects can be overcome
  - by interfacing with external compiled code in C/C++/Fortran
  - Active X and Mex files
- New developments
  - Parallel Matlab (http://www.ai.mit.edu/projects/ppserver)
  - New data types in new versions
- Good tool for learning and prototyping

Other tools

- Khoros (a little too old)
- Target Jr. (C++ library)
- DirectX graphedit with Intel Open Computer Vision Library (a little too new)
  - Encourage students pursuing computer vision research to work with this library
  - http://www.egroups.com/group/OpenCV