CS 696i, Spring 2005, part III

Computer Vision

(Making Machines See and telling J.P.)

and other miscellaneous stuff

Kobus Barnard

Computer Science, University of Arizona

Administrivia

You need to get set up to use Matlab ASAP. Some possibilities:

Matlab is available on CS linux machines. Public workstations are available in GS228. Remote access is fine, but you have to go via “lecture” and then logon to cy01, cy02, ... If you don’t have a CS account, then in theory you can get one via:

http://www.cs.arizona.edu/~apply/

However, if you did not register using CS 696i, you may need to send me E-mail regarding what course you are registered under before this will work.

Matlab is available on Windows machines in the CCIT “OSCR” lab in ECE 206.

Section Outline

Programming assignments:

One warm-up, due Tuesday, March 22
One real one, due Tuesday, March 29 (under construction)

Written assignment:

One written, due Tuesday, April 5 (under construction)

Outline for first two lectures:

What is intelligence and computational intelligence?
A few examples from the human visual system.
A very brief introduction to the theory of computation.

Section Outline

Topics after the break:

- Statistical methods / machine learning
- An introduction to machine vision
- Ambiguity in vision
- Top down and bottom up processing
- Representation in vision
- Re-connecting vision to artificial intelligence

Pop Quiz

Pop Quiz (1)

Programming: I have written a program in a general purpose language that does something interesting (T/F).

Pop Quiz (2)

Cardinality: I know that the number of real numbers exceeds that of the integers in a mathematically well defined way, and I have some vague recollection of how to prove it (T/F).

Pop Quiz (3)

Computability: I know that it is futile to attempt to write a program that reliably determines whether another program halts, and I have some vague recollection of how to prove it (T/F).

Pop Quiz (4)

Tractability: I know (roughly) what NP complete means (T/F).

Pop Quiz (5)

Statistics: I am familiar with Bayes rule (T/F).

Characterizing computational intelligence

Perspectives on intelligence from last section:

Intelligence according to the American Heritage Dictionary:

“The capacity to acquire and apply knowledge.”

Redefined for Chuck’s domain:
“The ability to acquire and apply information
from the environment to modify behavior”

Characterizing computational intelligence

Perspectives on intelligence from last section:

Intelligence according to the American Heritage Dictionary:

“The capacity to acquire and apply knowledge.”

Redefined for Chuck’s domain:
“The ability to acquire and apply information from the environment to modify behavior”

In this section we will attempt to characterize more abstractly, partly so that we can focus on computational intelligence.

Characterizing computational intelligence

Perspectives on intelligence from last section:

Something can be smart, and not know it.

Interesting/intelligent behavior can emerge from the interaction of simple things with a complex world.

Characterizing computational intelligence

Perspectives on intelligence from last section:

Something can be smart, and not know it.
Interesting/intelligent behavior can emerge from the interaction of simple things with a complex world.

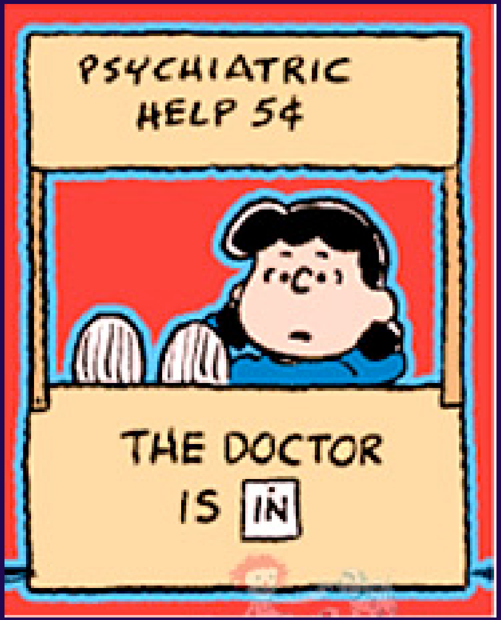
Lets “psychoanalyze” our low level vision system.

Characterizing computational intelligence

Color constancy demo ommitted from WWW version.

Characterizing computational intelligence

And the psychiatrist says ....



Characterizing computational intelligence

Second example: Stereo vision

First set of examples are from:

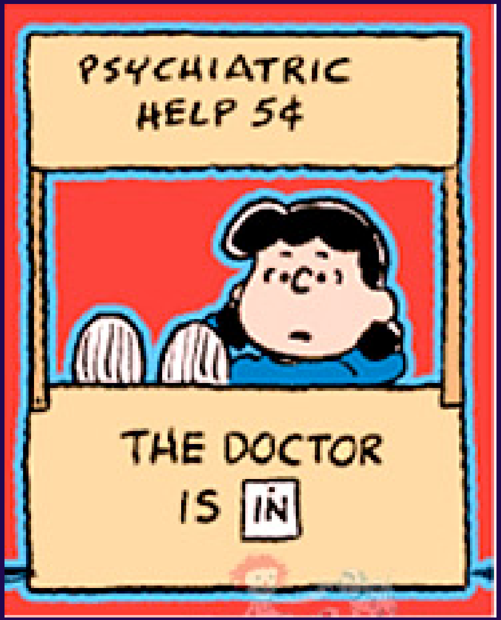
http://astronomy.swin.edu.au/~pbourke/opengl/redblue

Characterizing computational intelligence

Stereo demo ommitted from WWW version.

Characterizing computational intelligence

And the psychiatrist says ....



Characterizing computational intelligence

General principal: Understanding an intelligent system requires knowing and understanding what problem “it” is “trying” to solve.

Characterizing computational intelligence

General principal: Understanding an intelligent system requires knowing and understanding what problem “it” is “trying” to solve.

General approach: Focus on the problems to be solved.

What information to solve it is available?
Can it be done efficiently?
Can it be done robustly?
How to define/measure efficacy?
How to optimize efficacy?

Characterizing computational intelligence

General principal: Understanding an intelligent system requires knowing and understanding what problem “it” is “trying” to solve.

General approach: Focus on the problems to be solved.

What information to solve it is available?
Can it be done efficiently?
Can it be done robustly?
How to define/measure efficacy?
How to optimize efficacy?

Notice that while we are always keen to get architectural hints from biological systems, the approach is architecture neutral.