# Object Recognition

# +

# Gesture Recognition

Matt Loper

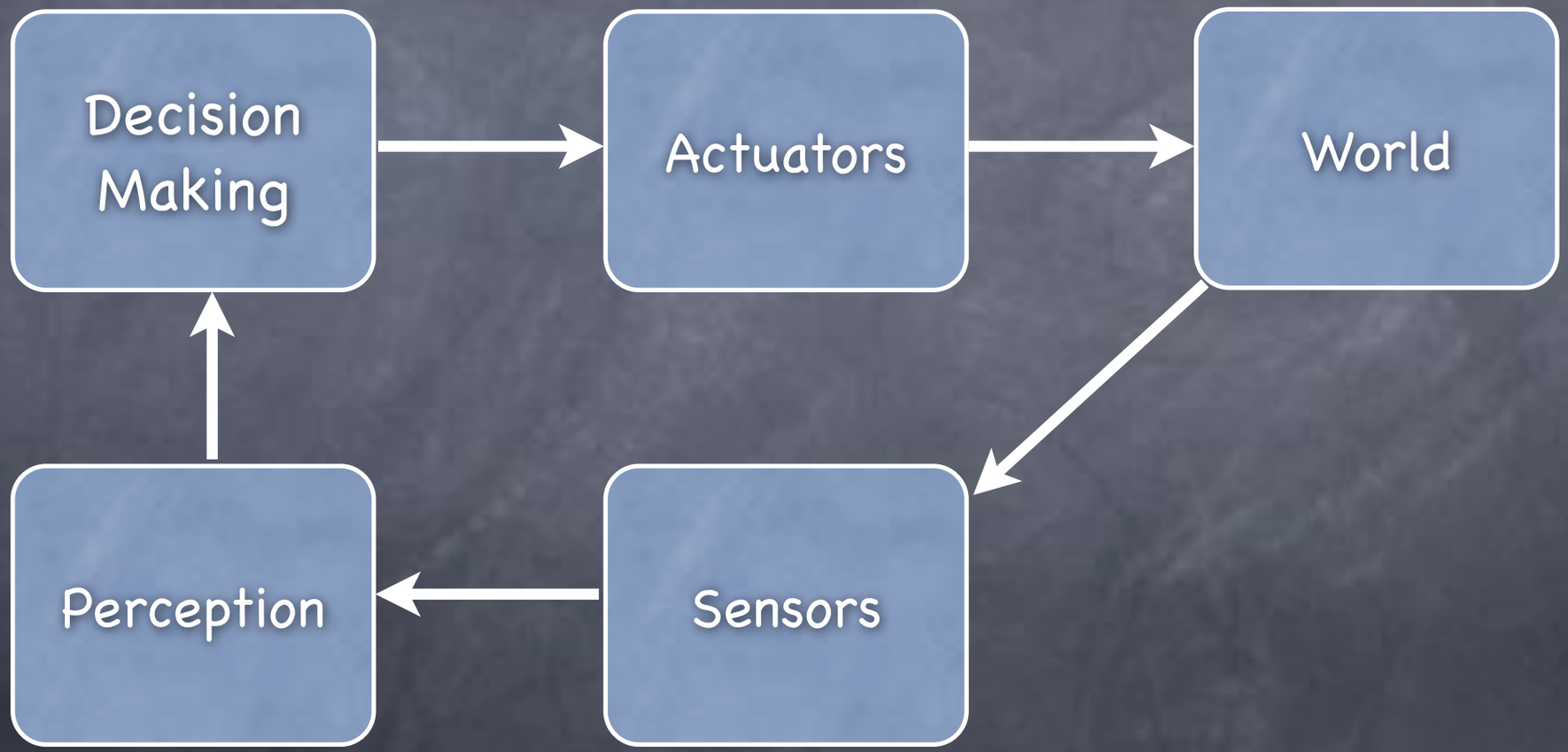
CS148

Nov 1st, 2007

BROWN

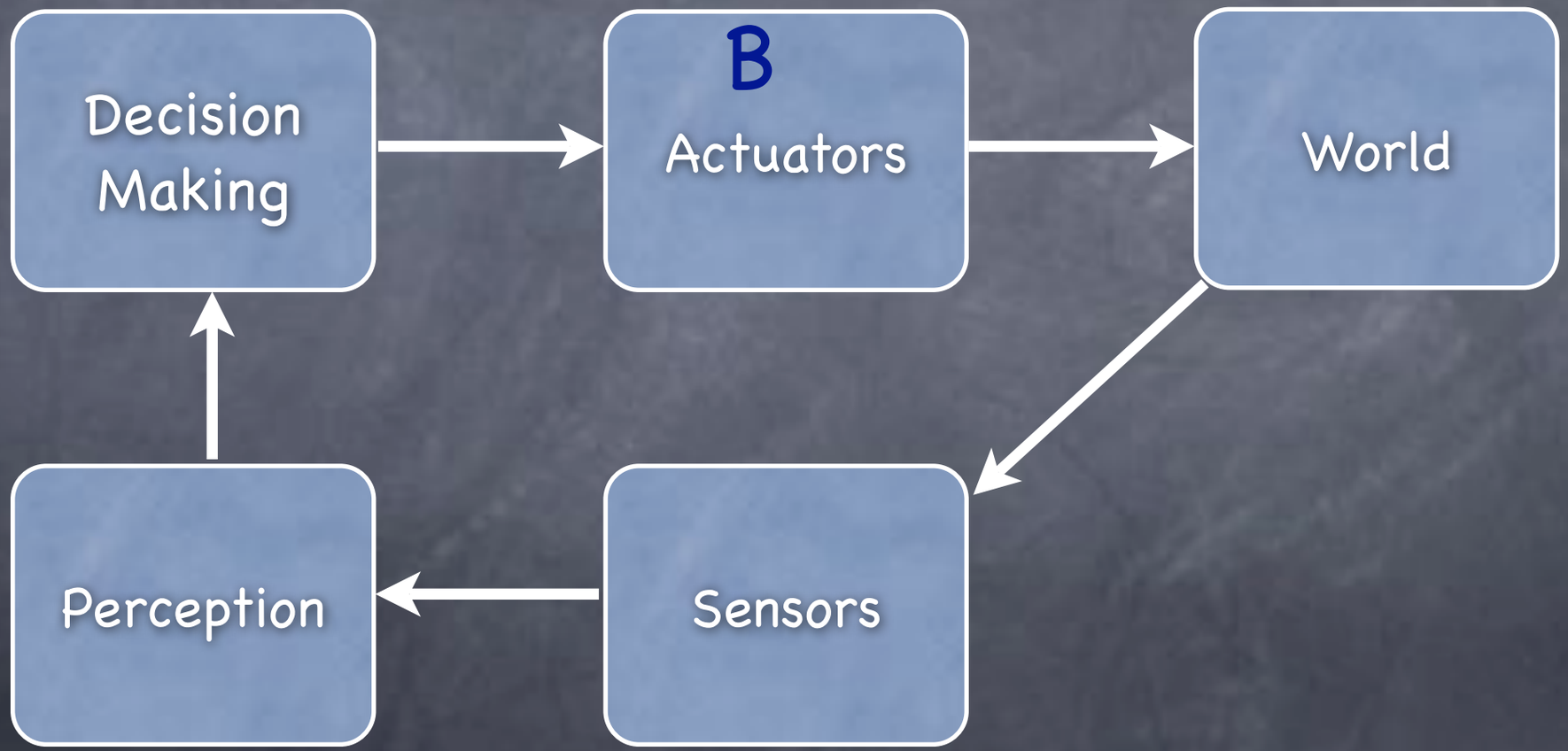
# Motivation

- Consider the robot control loop
- Compare it to a human



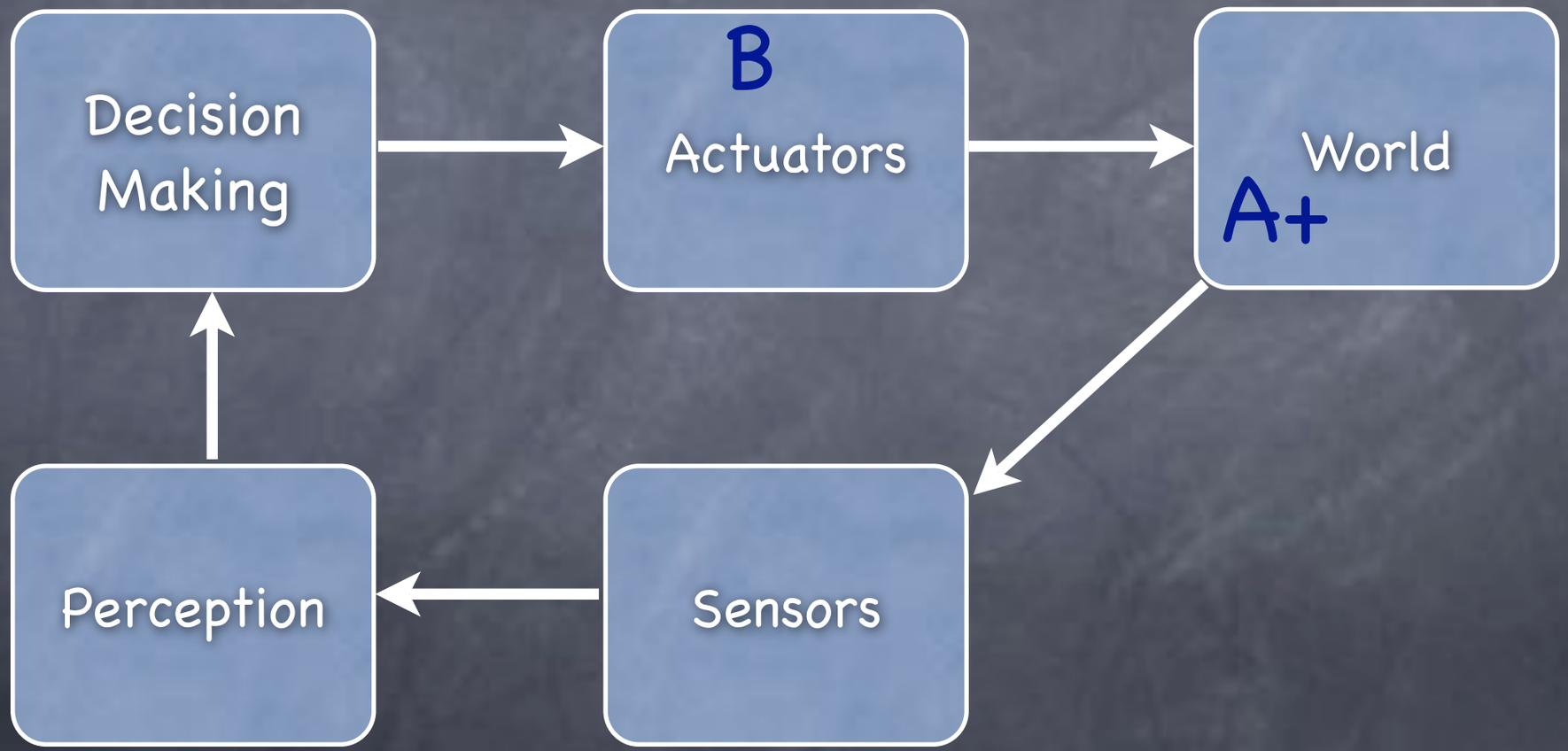
# Motivation

- Consider the robot control loop
- Compare it to a human



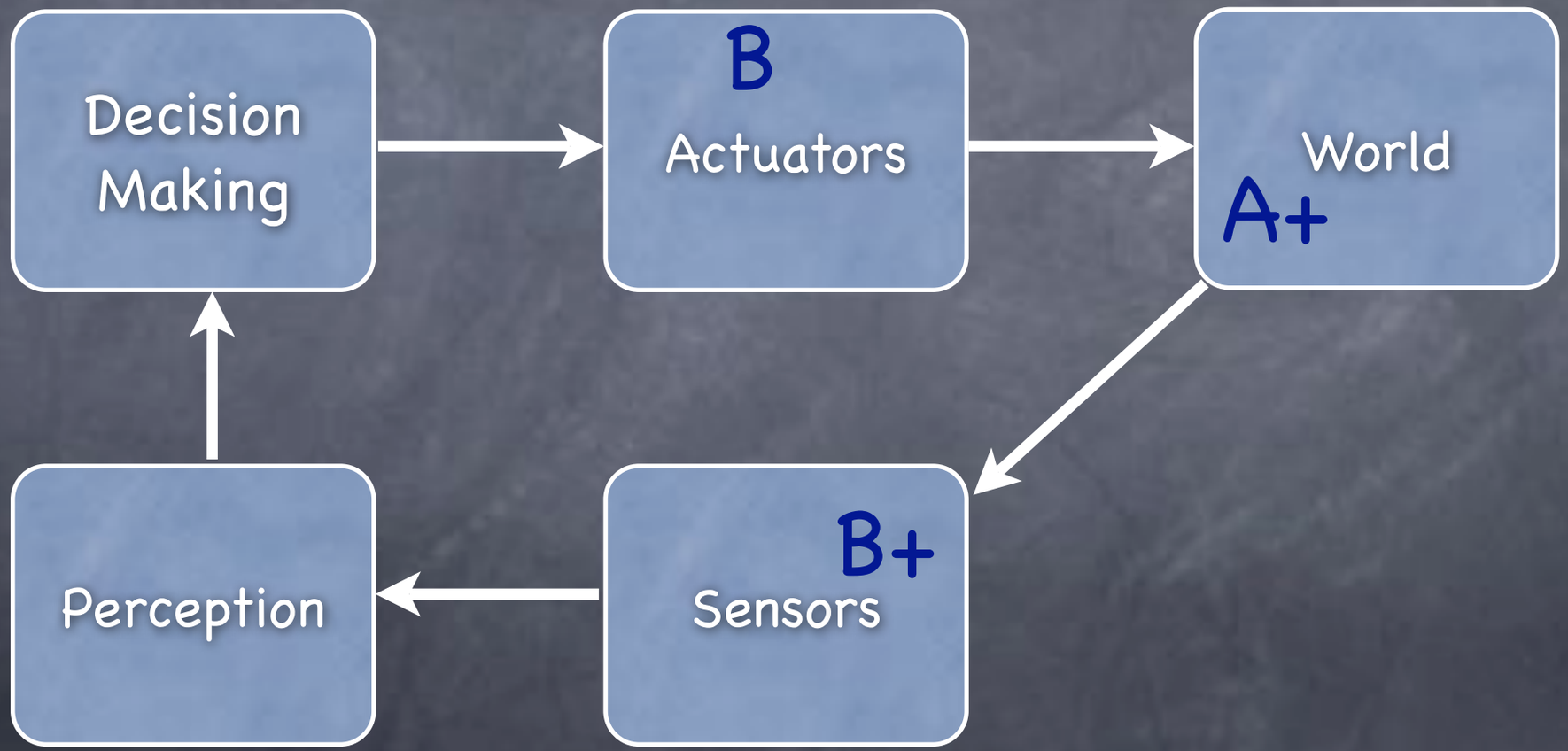
# Motivation

- Consider the robot control loop
- Compare it to a human



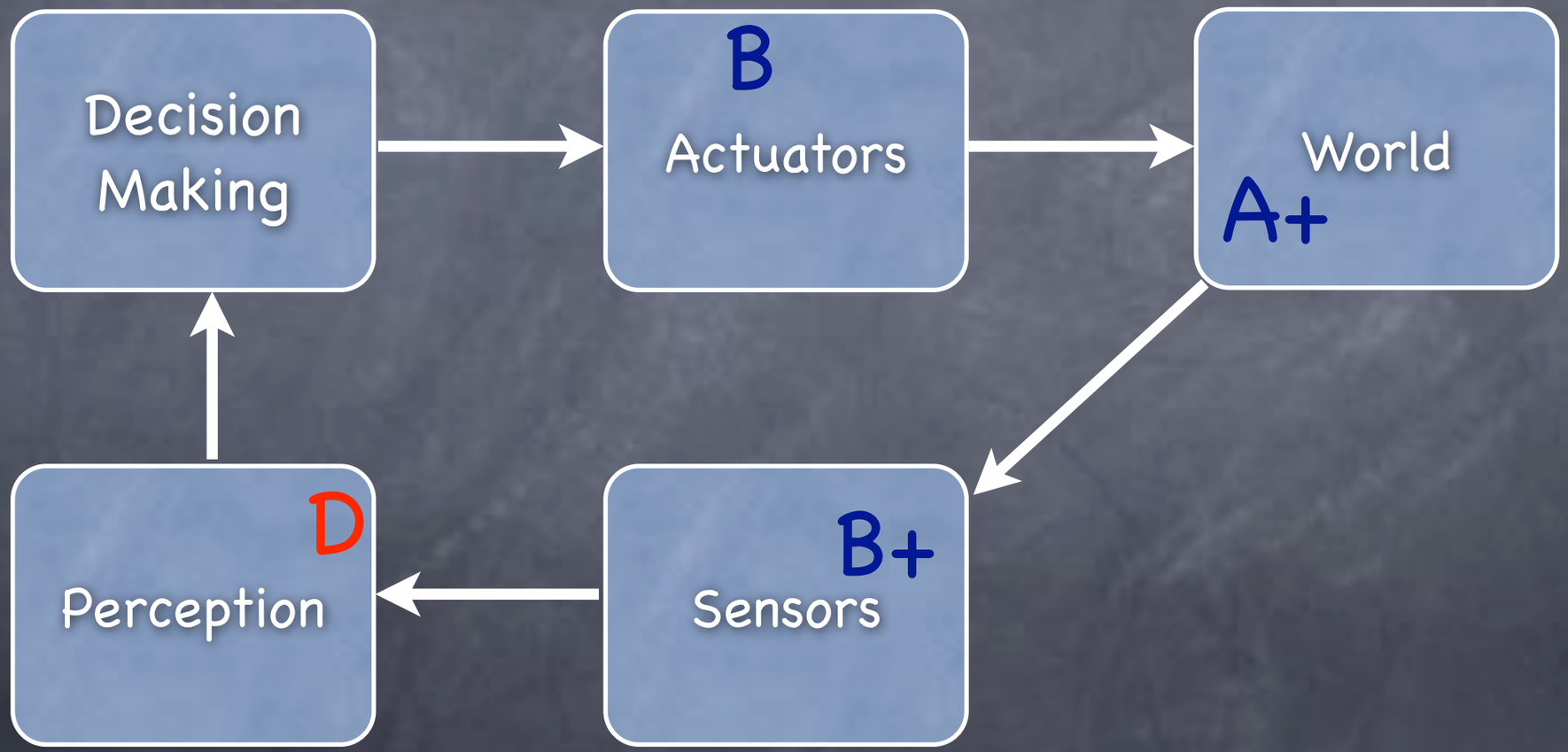
# Motivation

- Consider the robot control loop
- Compare it to a human



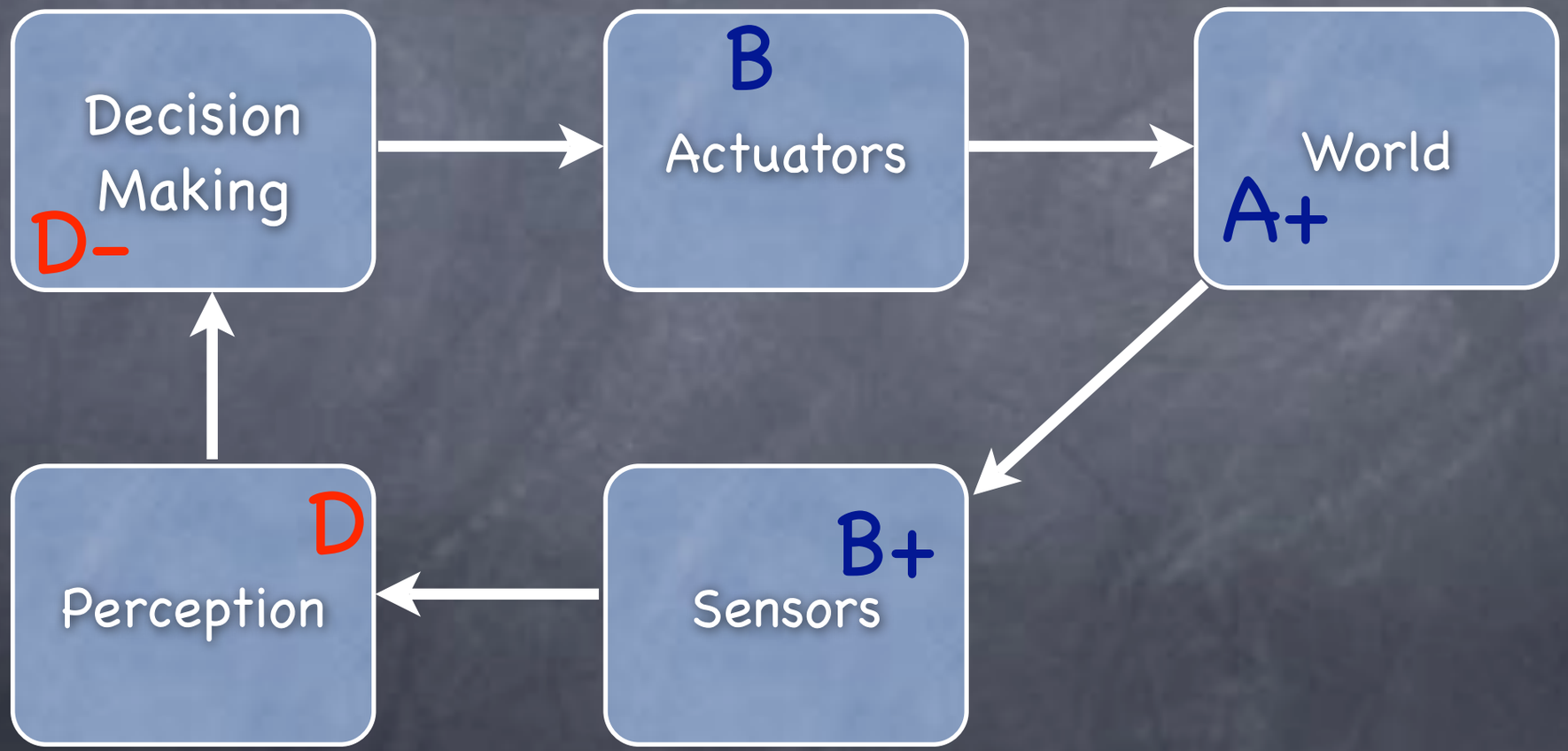
# Motivation

- Consider the robot control loop
- Compare it to a human



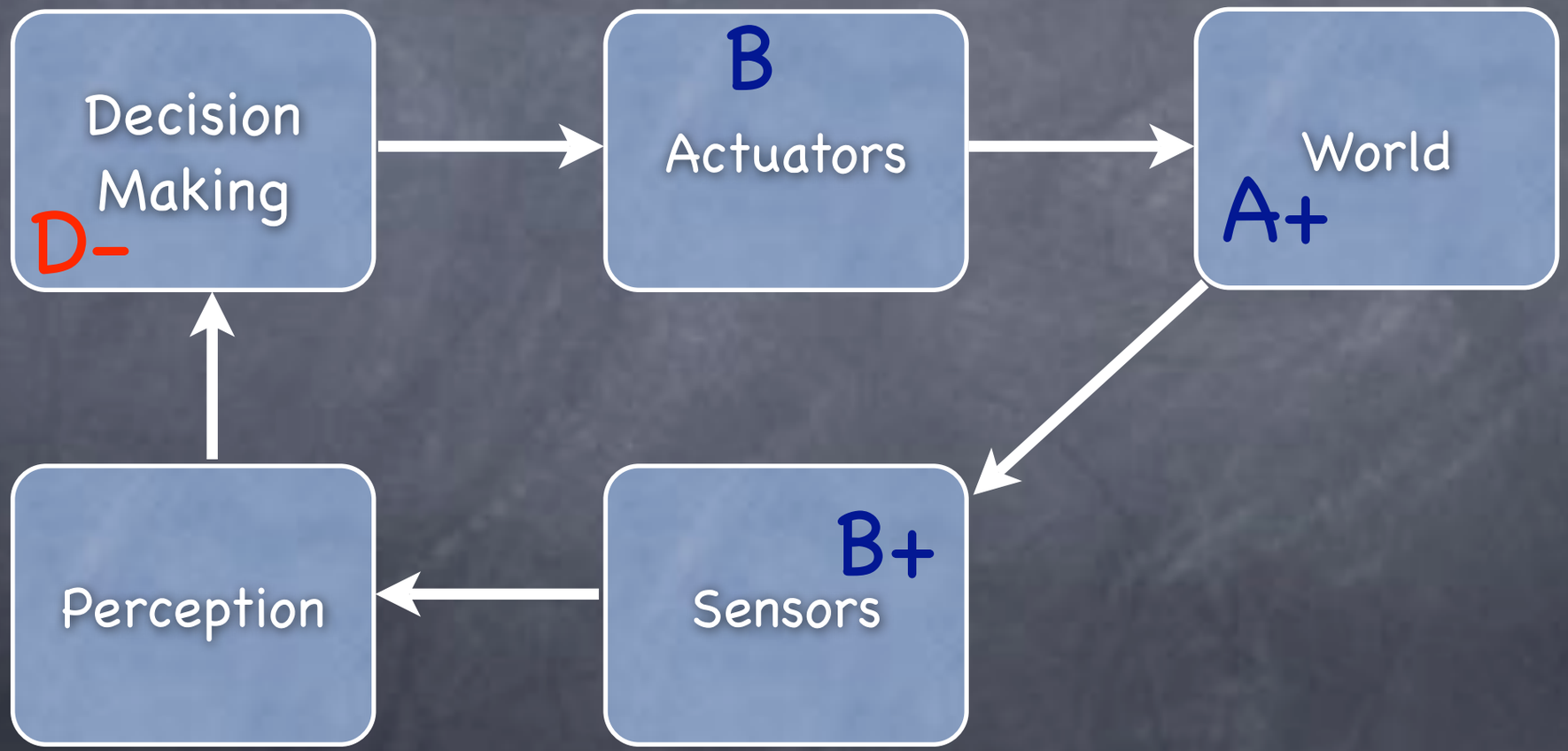
# Motivation

- Consider the robot control loop
- Compare it to a human



# Motivation

- Consider the robot control loop
- Compare it to a human



# Motivation

- Consider the robot control loop
- Compare it to a human

**Perception**

Localization

People Detection

Object Recognition

Action Recognition

Others...

# Motivation

- Consider the robot control loop
- Compare it to a human

Perception

Localization

People Detection

**Object Recognition**

**Action Recognition**

Others...

# What is object recognition?

# Two philosophies

Model based

Object Centric

Top Down

Template based

View Centric

Bottom up

# Two philosophies

## Goal: Find the horse in the picture

Model based

Object Centric

Top Down

Template based

View Centric

Bottom up

# Two philosophies

## Goal: Find the horse in the picture

Model based

Object Centric

Top Down

**"I have a geometric model of the horse."**

Template based

View Centric

Bottom up

**"I have pictures of parts (hooves etc) of the horse."**

# Two philosophies

## Goal: Find the horse in the picture

Model based

Object Centric

Top Down

**"I have a geometric model of the horse."**

Template based

View Centric

Bottom up

**"I have pictures of parts (hooves etc) of the horse."**

# Two philosophies

## Goal: Find the horse in the picture

Model based

Object Centric

Top Down

**"I have a geometric model of the horse."**

Template based

View Centric

Bottom up

**"I have pictures of parts (hooves etc) of the horse."**

# Two steps to O.R.

- In theory, only two things to do:

  1. Collect features
  2. Perform classification, using those features

# Two steps to O.R.

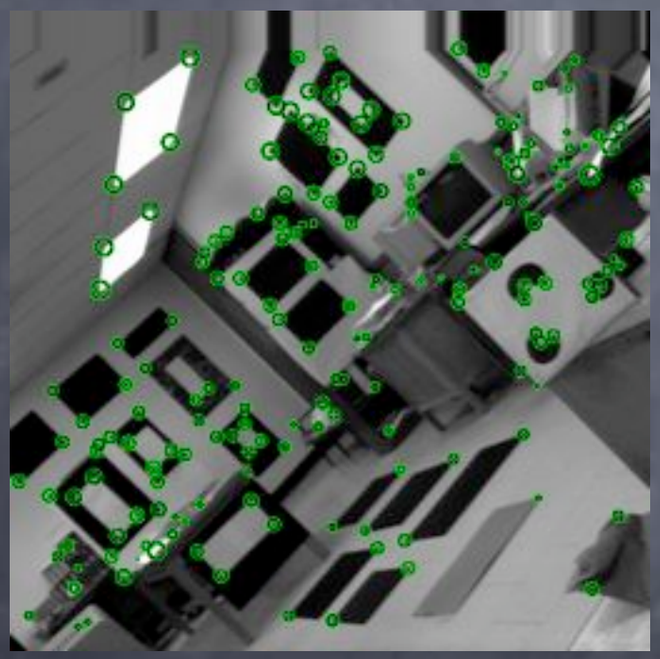
- In theory, only two things to do:

1. Collect features
   - Tool: image processing
2. Perform classification, using those features
   - Tool: machine learning approach

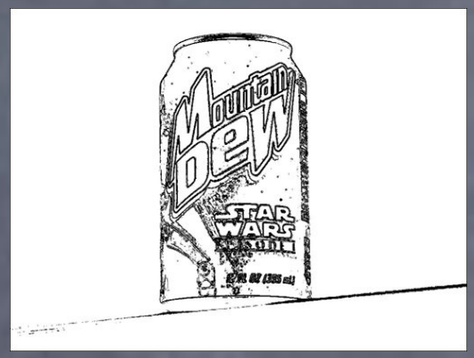
# Feature Collection

- Image processing brings us closer...

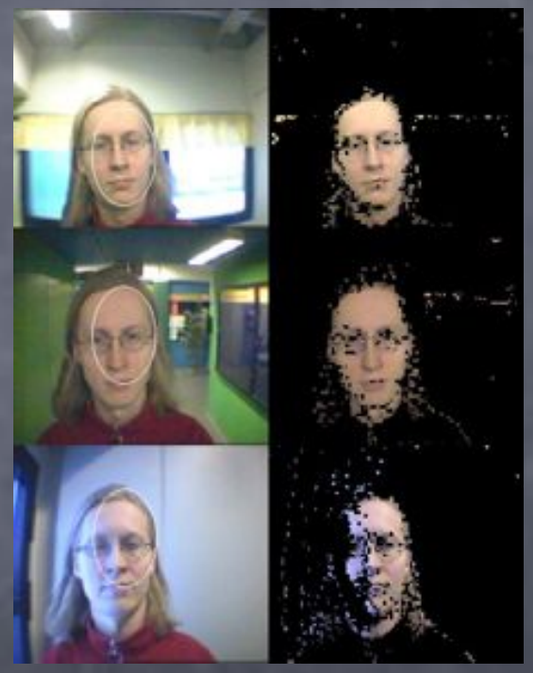
Corners



Edges



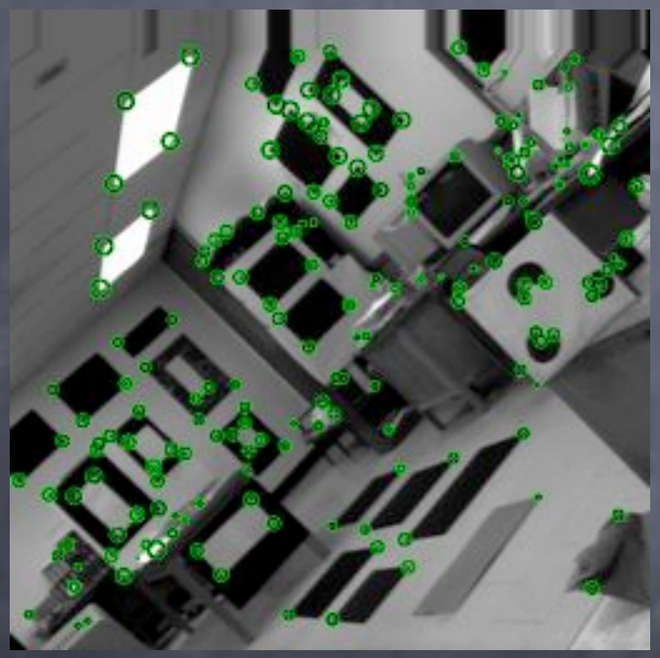

Blobs



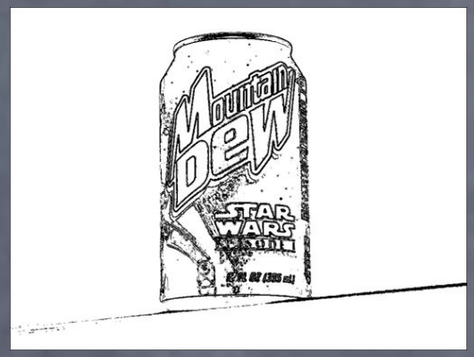
- But now we must formalize these into features

# Feature Collection

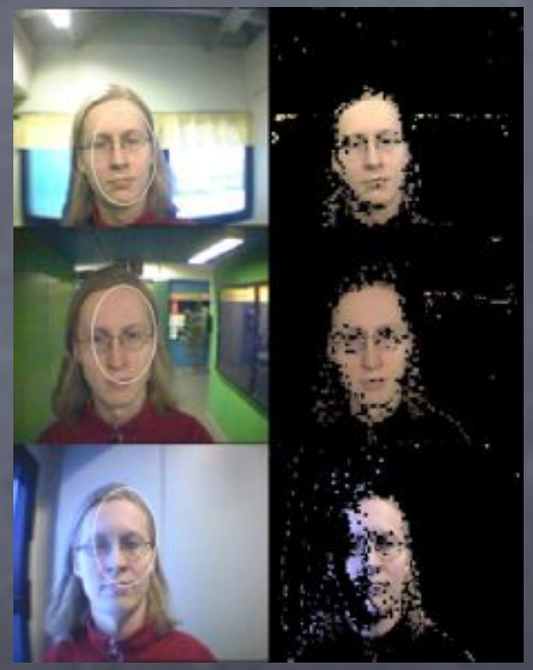
## Corners



## Edges

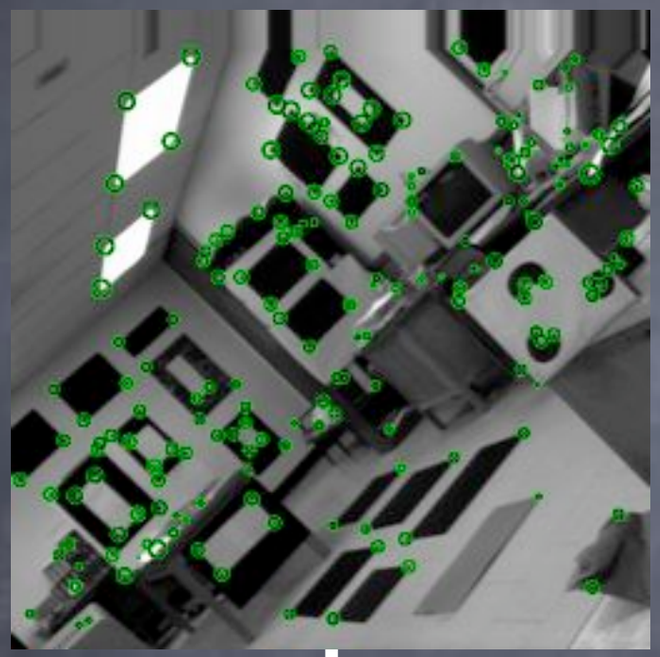



## Blobs



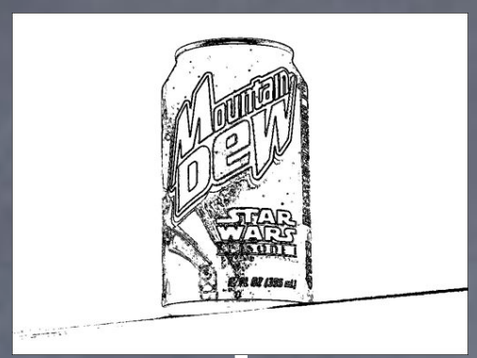
# Feature Collection

## Corners



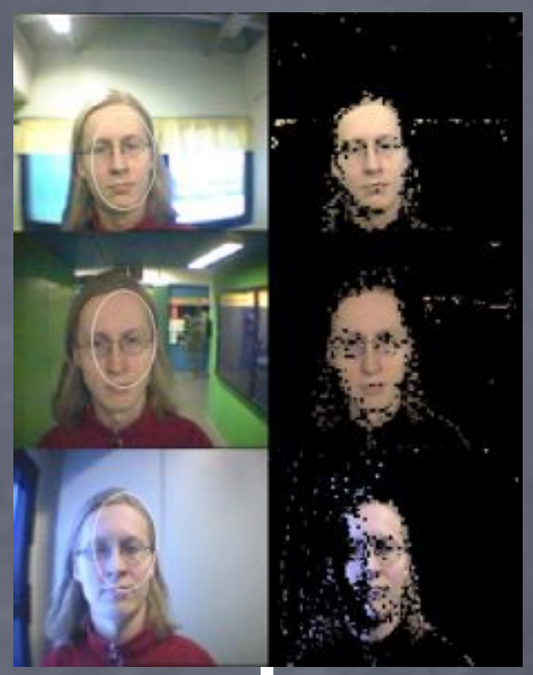
x
y
local brightness

## Edges




begin x,y
end x,y
curve polynomial

## Blobs



mean blob x,y
color distribution

# Feature Limitations

- General limitations
  - Rotational invariance
  - Scale invariance
- Specific feature limitations
  - Corners: Many objects are soft
  - Edges: Aperture problem
  - Blobs: subject to lighting changes, surface reflectance

# SIFT

- Introduced in 1999 by David Lowe
- (S)cale (I)nvariant (F)eature (T)ransform
- Galvanized the field
- Three attributes
  1. Scale invariance
  2. Rotation invariance
  3. Corner detection

# Scale Invariance

- How do you look for a variable sized object?
- Looking for a needle in a haystack...

Needle

Haystack

# Scale Invariance

- How do you look for a variable sized object?
- Looking for a needle in a haystack...

Needle

Match!

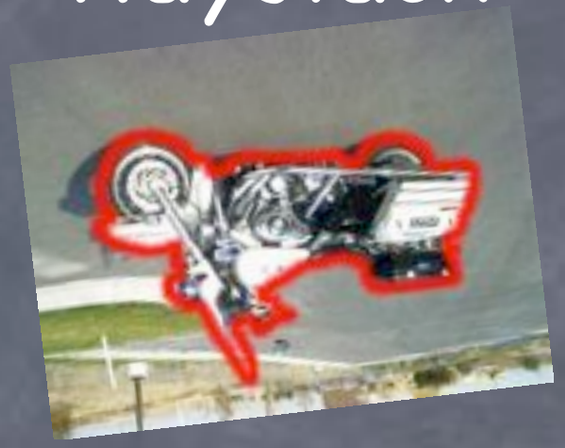
Haystack

# Rotation Invariance

- What if the object is rotated?

Needle

Haystack

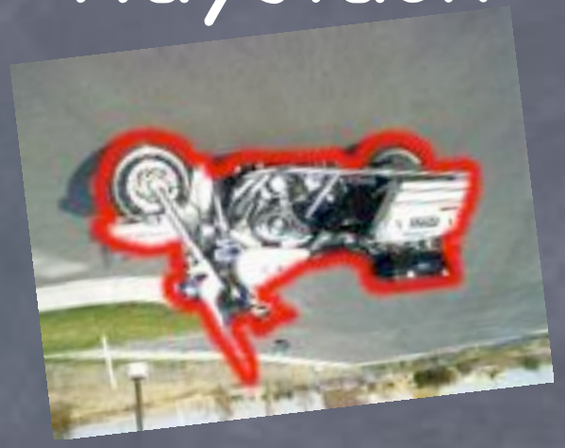


# Rotation Invariance

- What if the object is rotated?

Needle

Haystack



- Rotate to canonical orientation before comparison…

# Corner detection

- Corner detection: misnomer
- Finds locally unique regions
- What is NOT a corner?

# Corner detection

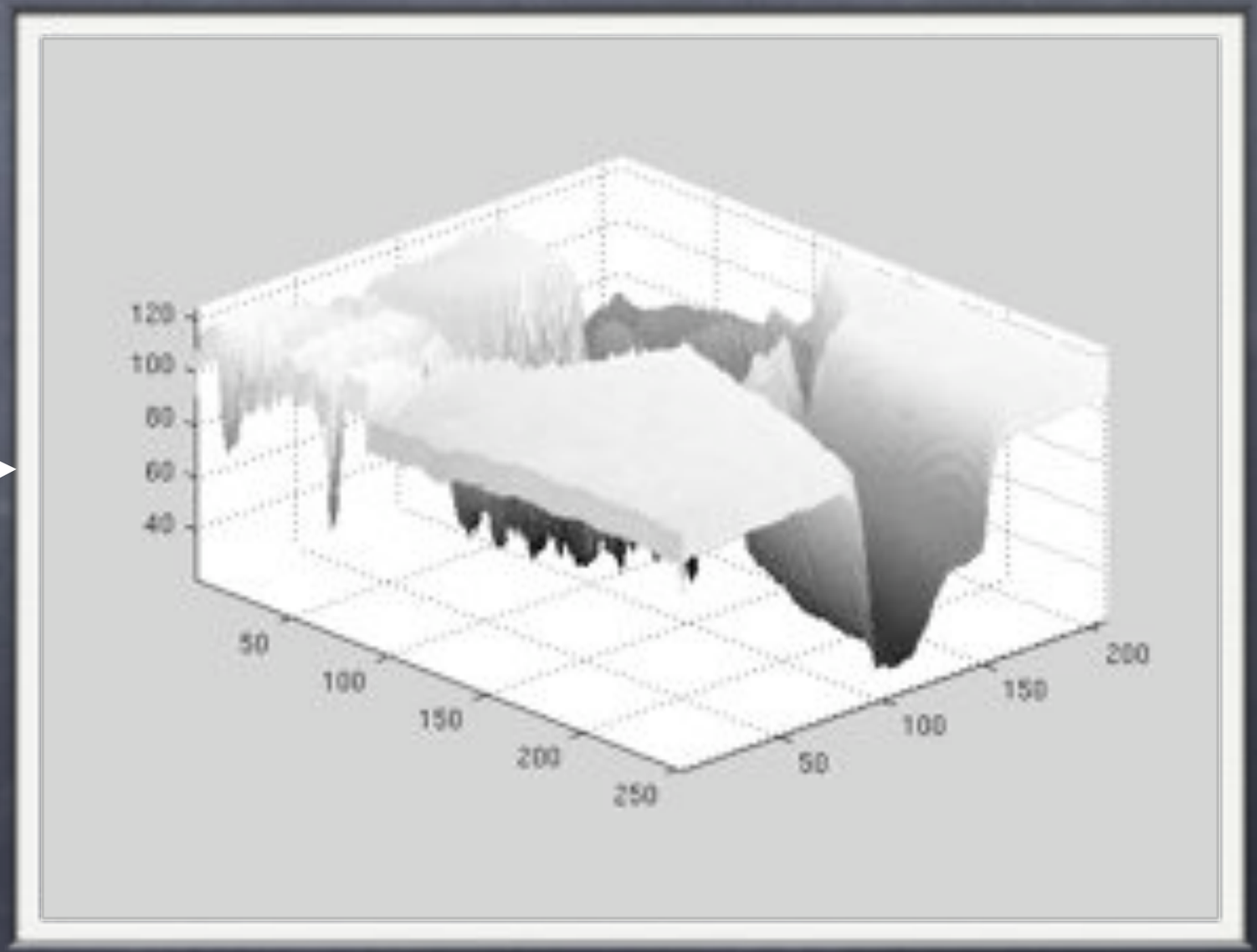
- Corner detection: misnomer
- Finds locally unique regions
- What is NOT a corner?

# Background: corner detection

- Think of an image as a mountainous landscape
- White is high, black is low

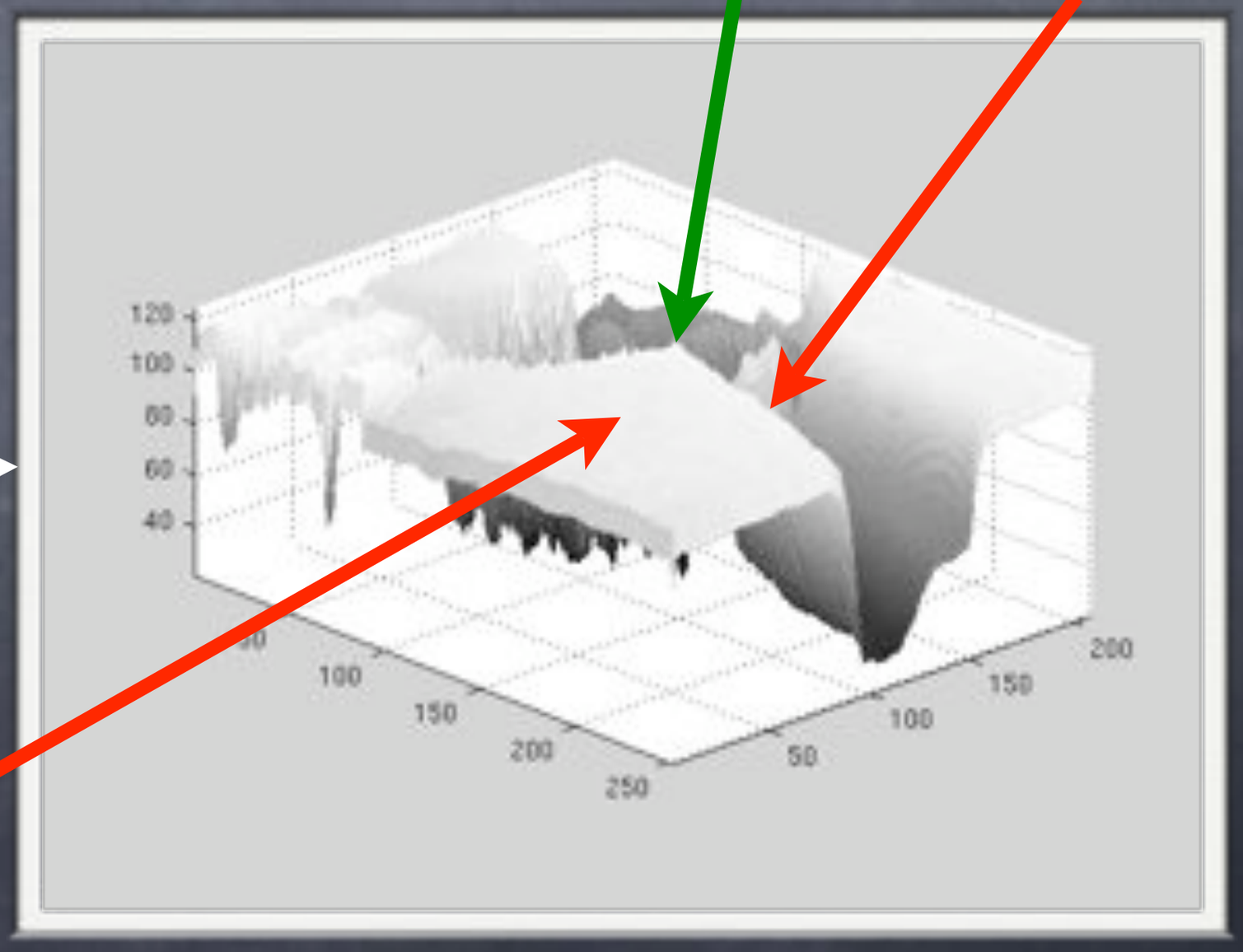
# Visualizing gradients

- Images can be seen as heightmaps
- Where are corners?



# Visualizing gradients

- Images can be seen as heightmaps
- Where are corners?



# Multiscale corners

- Different corners at different scales

# Multiscale corners

- Different corners at different scales

# Step 1 of 3: Get features

- In training image (needle)
  - Find corners in different scales
  - For each corner...
    - Find local image patch
    - Rotate it to canonical orientation
- In testing image (haystack)
  - Do the same thing!!

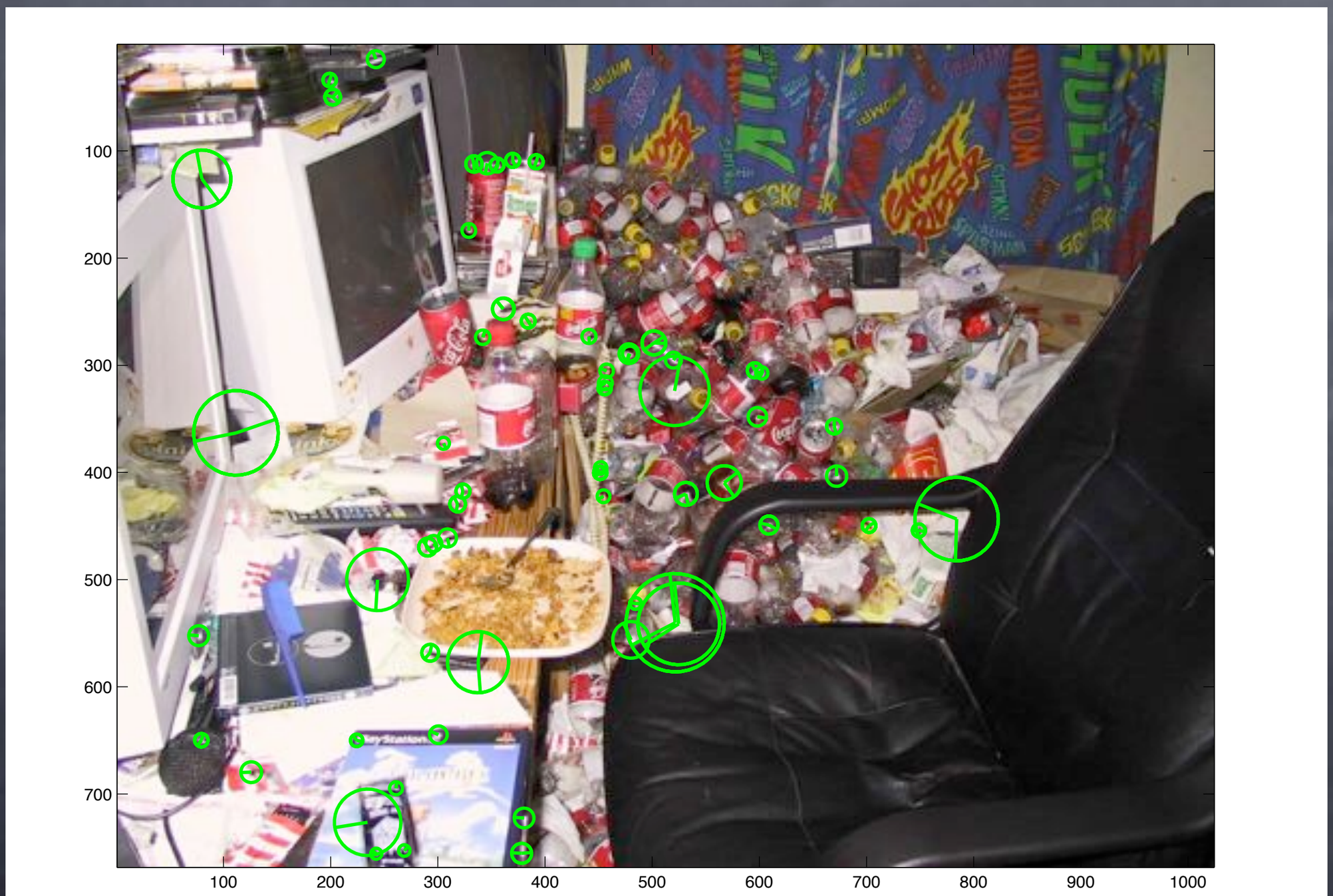
# Step 2 of 3: match features

- What do we have now?
  - Corner features in training
  - Corner features in testing
- Now we just have to match them up
  - We have M x N comparisons
  - M is # training corners, N is # testing corners
  - Compare image regions

# Step 3 of 3: match objects

- Find collections of matches that make sense
- Training images show spatial relationships
- Testing images should retain those for objects found

# SIFT corners found

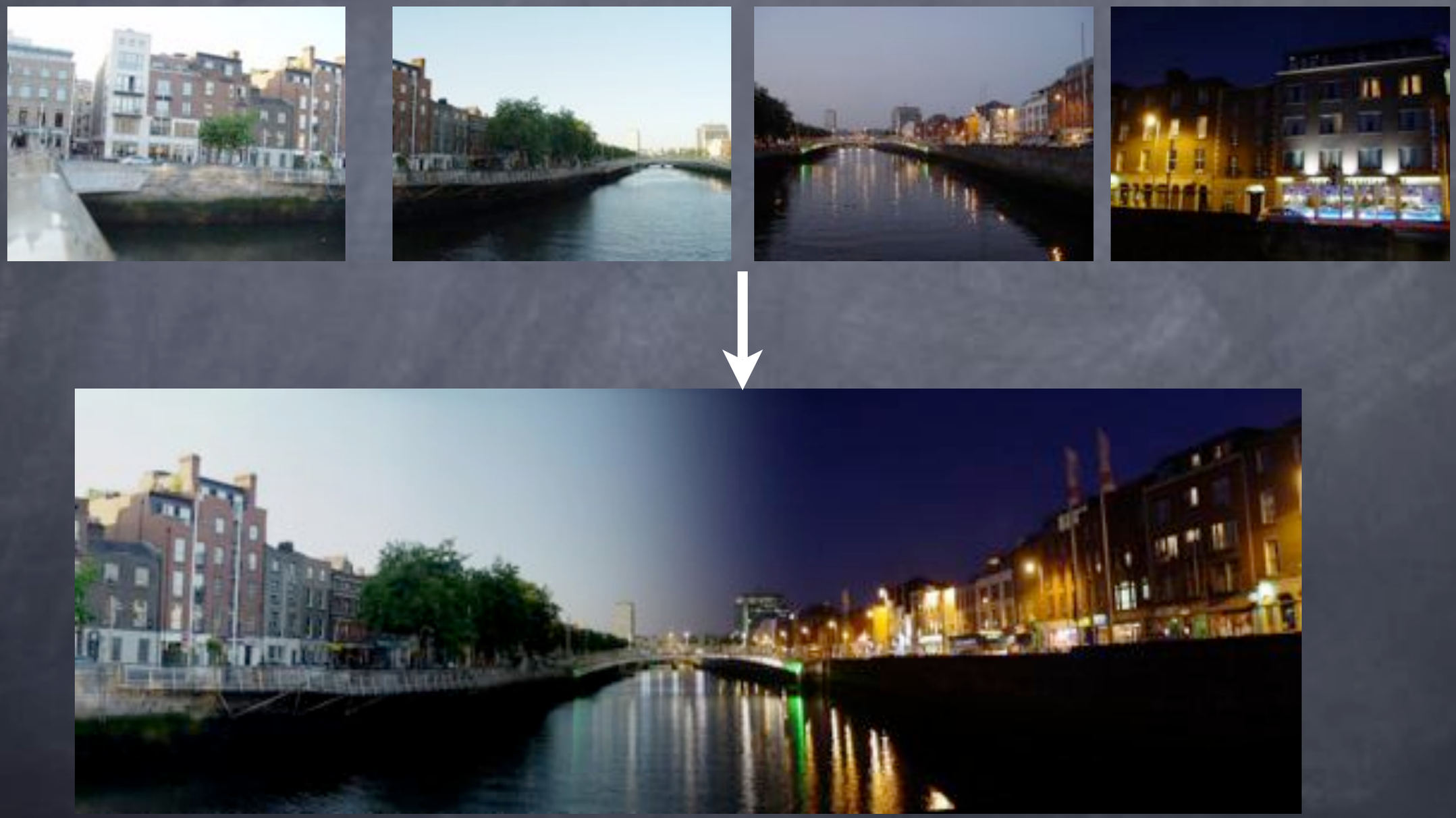


# Applications: Blackjack

- Train dog on each card
- Look for features that match known spatial arrangement

# Applications: Panoramas

- AutoStitch (google it, it's fun and easy)



# SIFT at home

- http://vision.ucla.edu/~vedaldi/code/sift/sift.html



AndreaVedaldiUCLAVisionLab

Home

Research

Publications

Code

VLFeat

Bag of features

MDoc

Autorights

VLPov

Anaview

SIFT++

SIFT

Custom keypoints

MSER

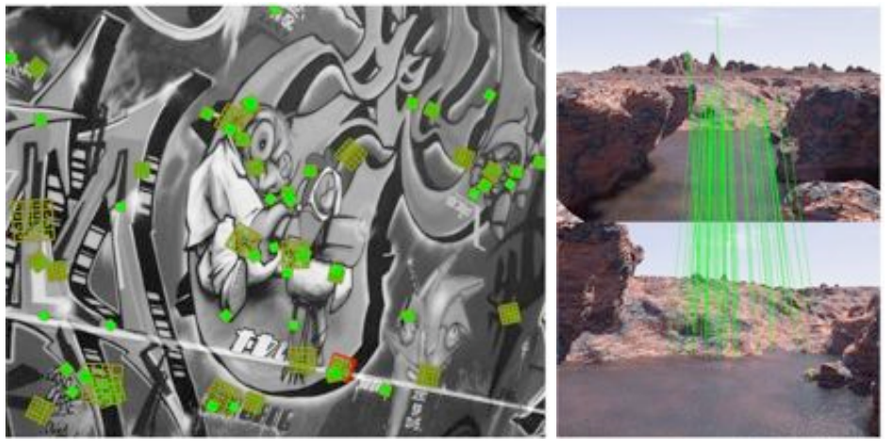
VLUtil

Code Snippets

Restricted

## SIFT

### An open implementation of SIFT



This is a MATLAB/C implementation of SIFT detector and descriptor. It is fairly customizable and features a decomposition of the algorithm in several reusable M and MEX files. This implementation produces interest points and descriptors which are very similar to David Lowe's implementation.

Remark. This code is well suited to study, understand and modify SIFT, but it is not particularly fast. If you need to compute lots of features, you might be interested in this lightweight C++ version, which does not require MATLAB and comes with a flexible command line interface.

## Copyright

# SIFT Drawbacks

- Rotation
  - If object rotates around camera's z-axis, that's ok
  - But if object rotates around other axes, that's a potential problem
  - SIFT tolerates only 15% to 20% rotation
  - Can be worked around by generating features at other rotations
- Floppy objects don't generate reproducible features

# Object Recognition Summary

- Why: because current robot perception is not good
- What: feature selection+classification
- How: SIFT (or Viola & Jones, or color-based segmentation)

# Motivation

- Automation: for the 3 d's
  - dirty
  - dangerous
  - dull
- Great! What's done already?

# What's automated today?

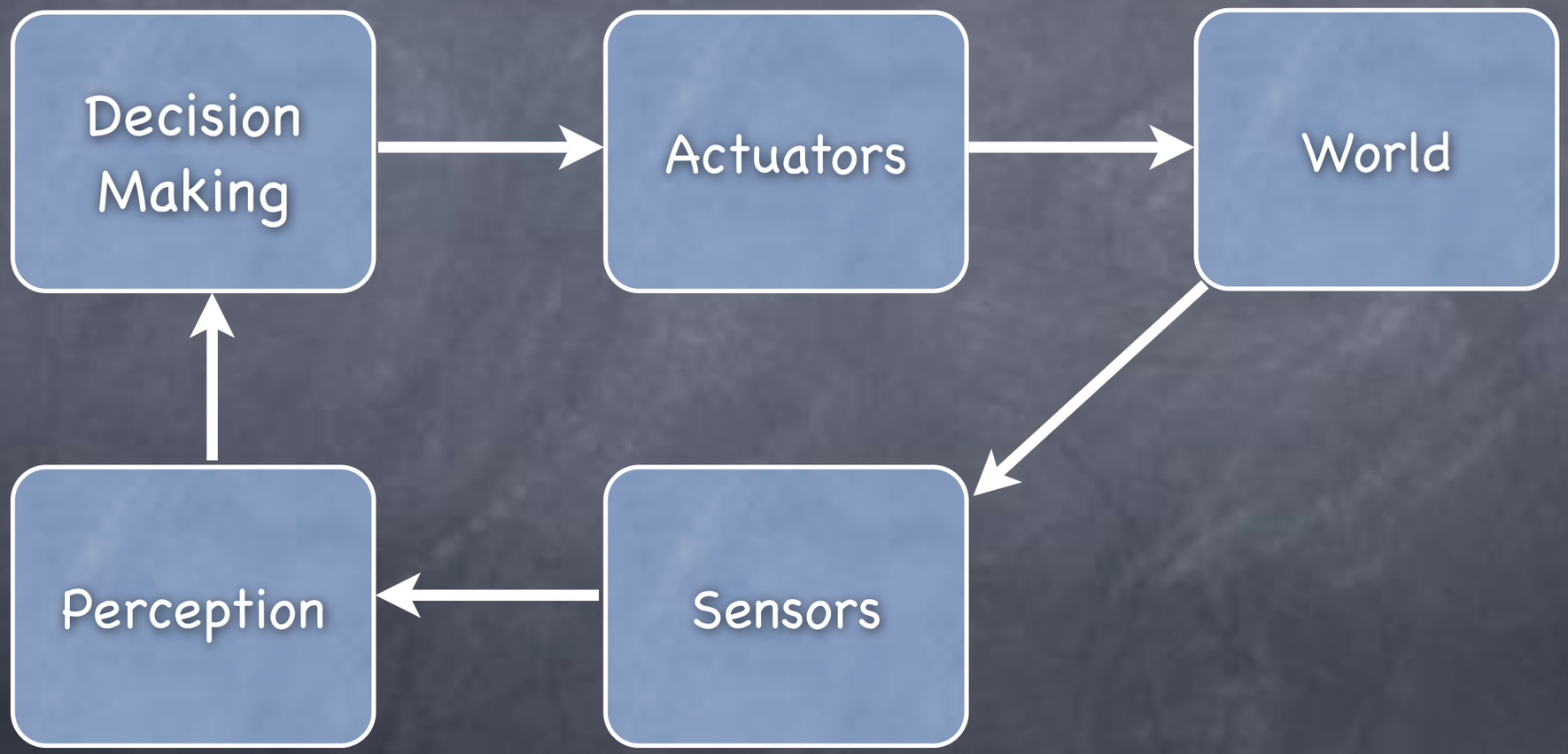
- Today's automation
  - ATM's
  - Automatic/assisted assembly
    - cars
    - electronics
    - packaging
  - Vending
- What is in common?

# Motivation

- Today's automation: ATM's, assembly, vending, automated phone assistance
- What's in common?
  - They are **immobile**
  - Communicating with them is **unnatural**
- Many dull, dangerous tasks need mobility!
- ...and can benefit from natural communication!

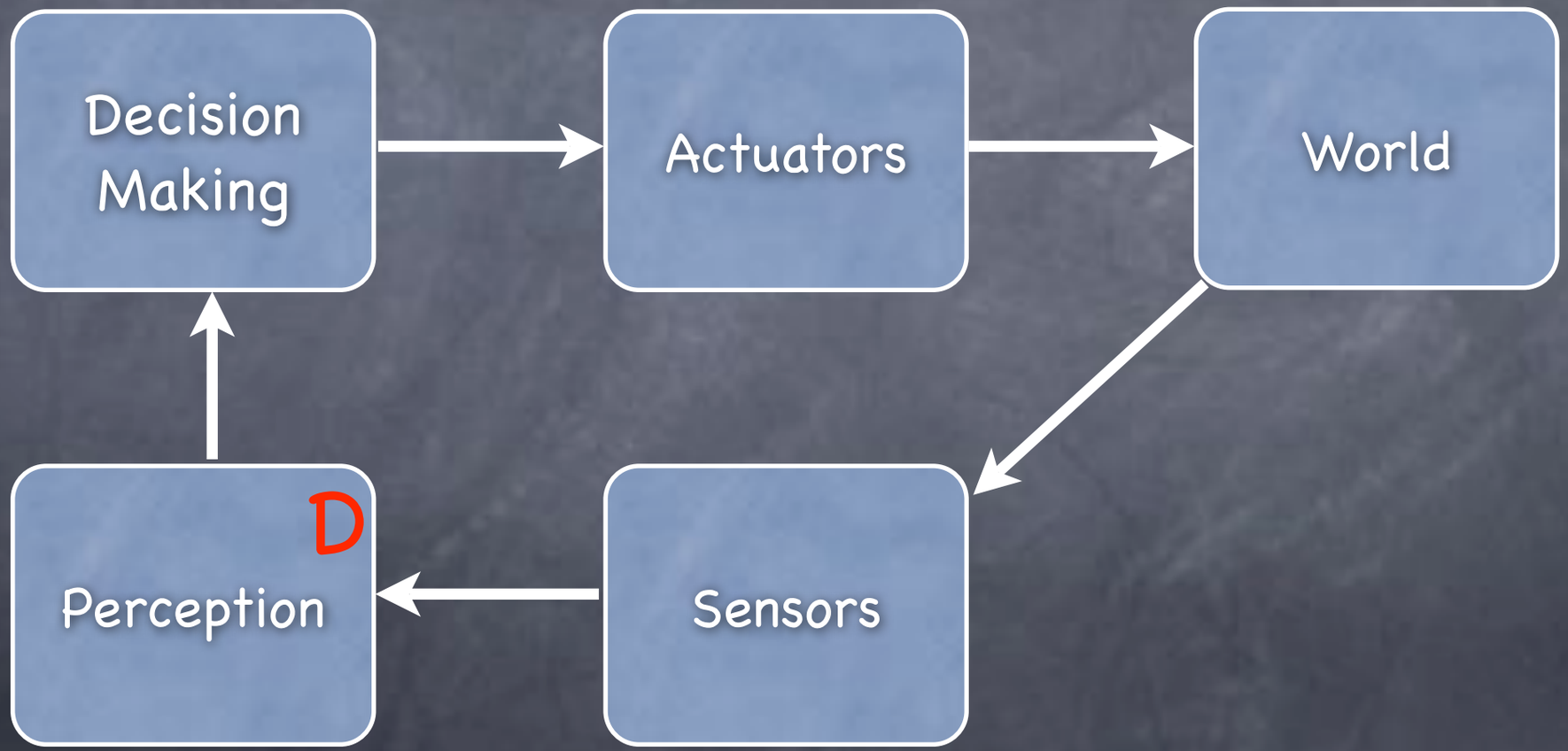
# Immobility: why?

- Why don't robots truck around on thayer?



# Immobility: why?

- Why don't robots truck around on thayer?



# Immobility: why?

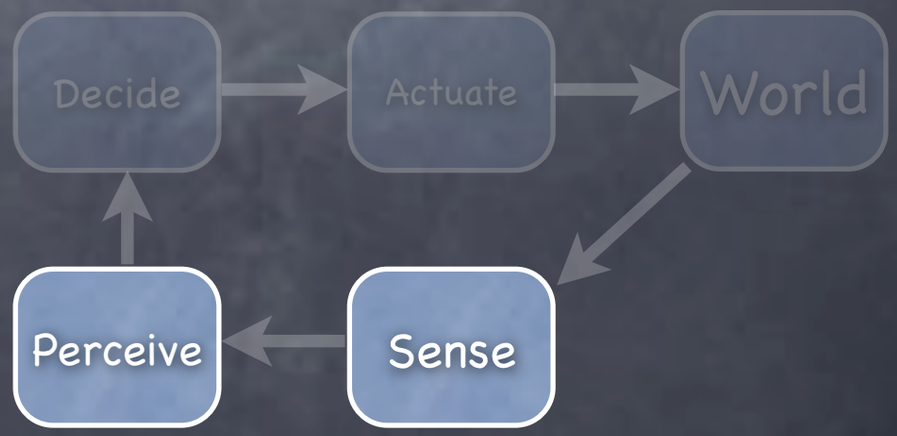
- Danger to us, and/or themselves
- Can't "see" things or people

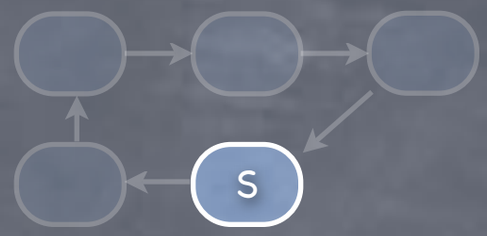
# Natural communication

- Hard to communicate when you can't "see" people
- Speech recognition works alright when trained on one person, or on simple words
  - Hard to hear when it's noisy, different accents are being used, etc
- I will focus on Action Recog.

# Goals

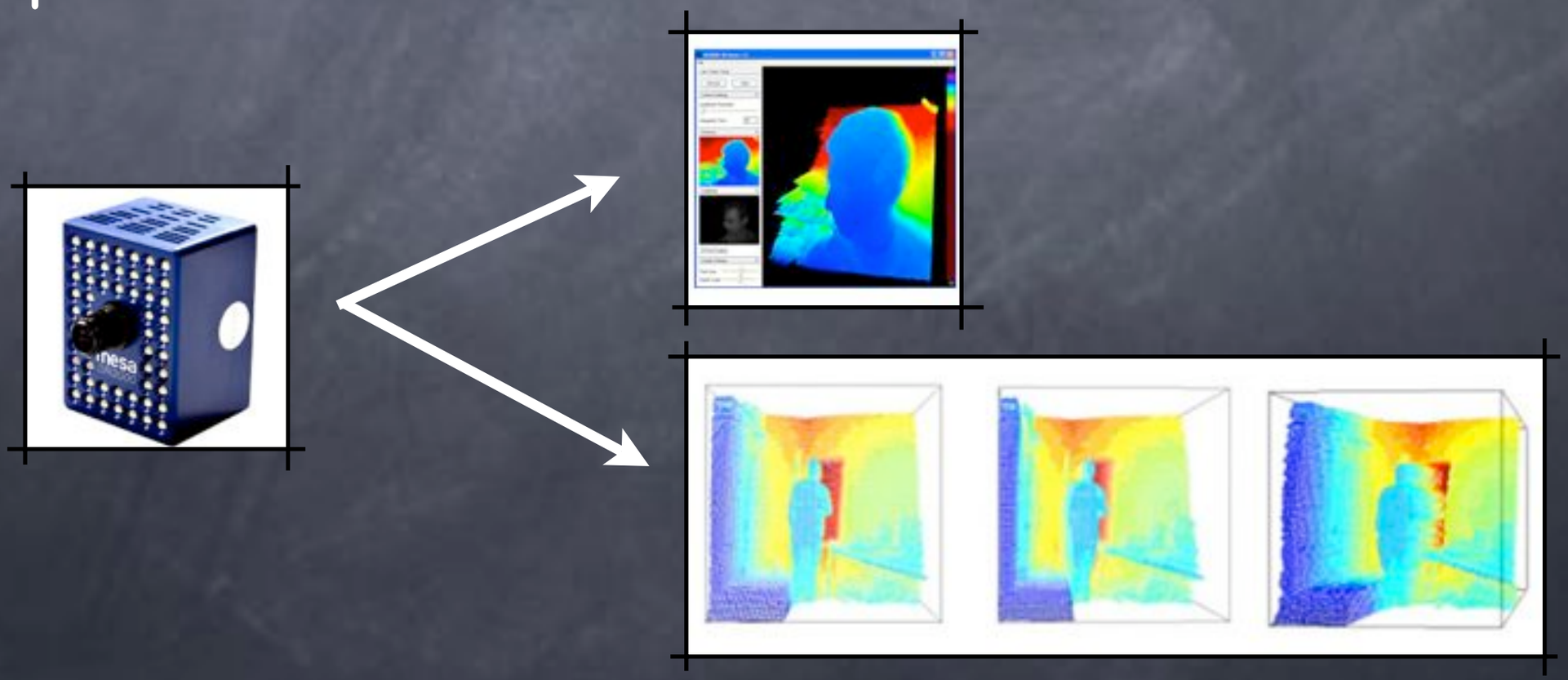
- Goals
  - Person detection
  - Gesture-based communication
  - A 3d view of the world
- Means
  - Improve sensors
  - Improve perception



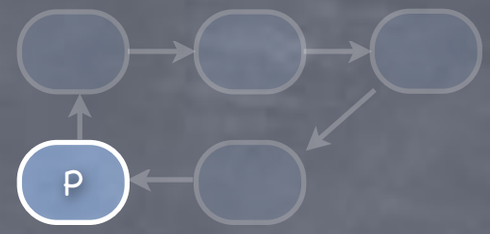



# Sensors

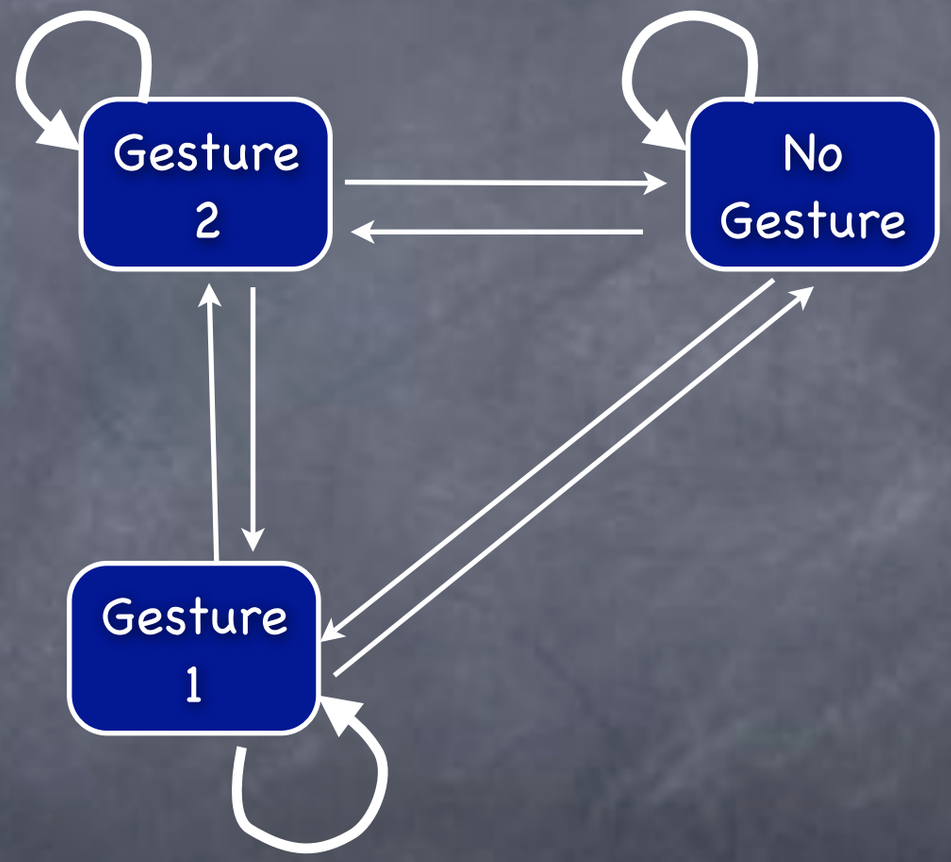
- CSEM SwissRanger
- Emits nonvisible light
- Recovers a depth image
- Expensive



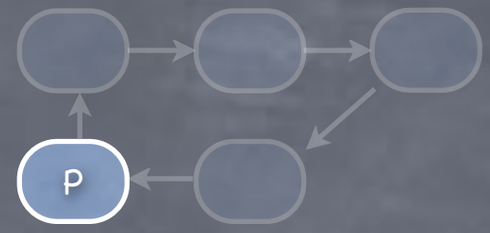
# Perception



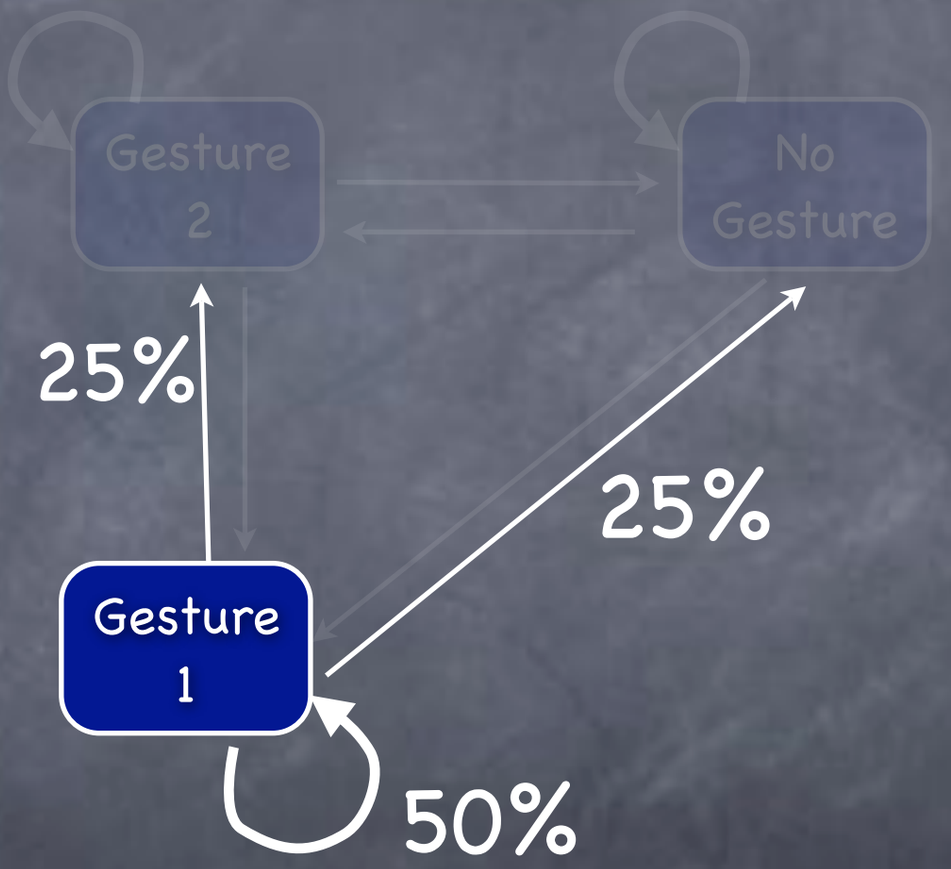

- Hidden Markov Model



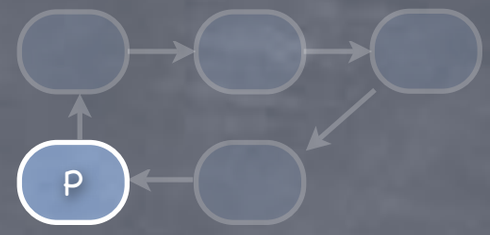
# Perception



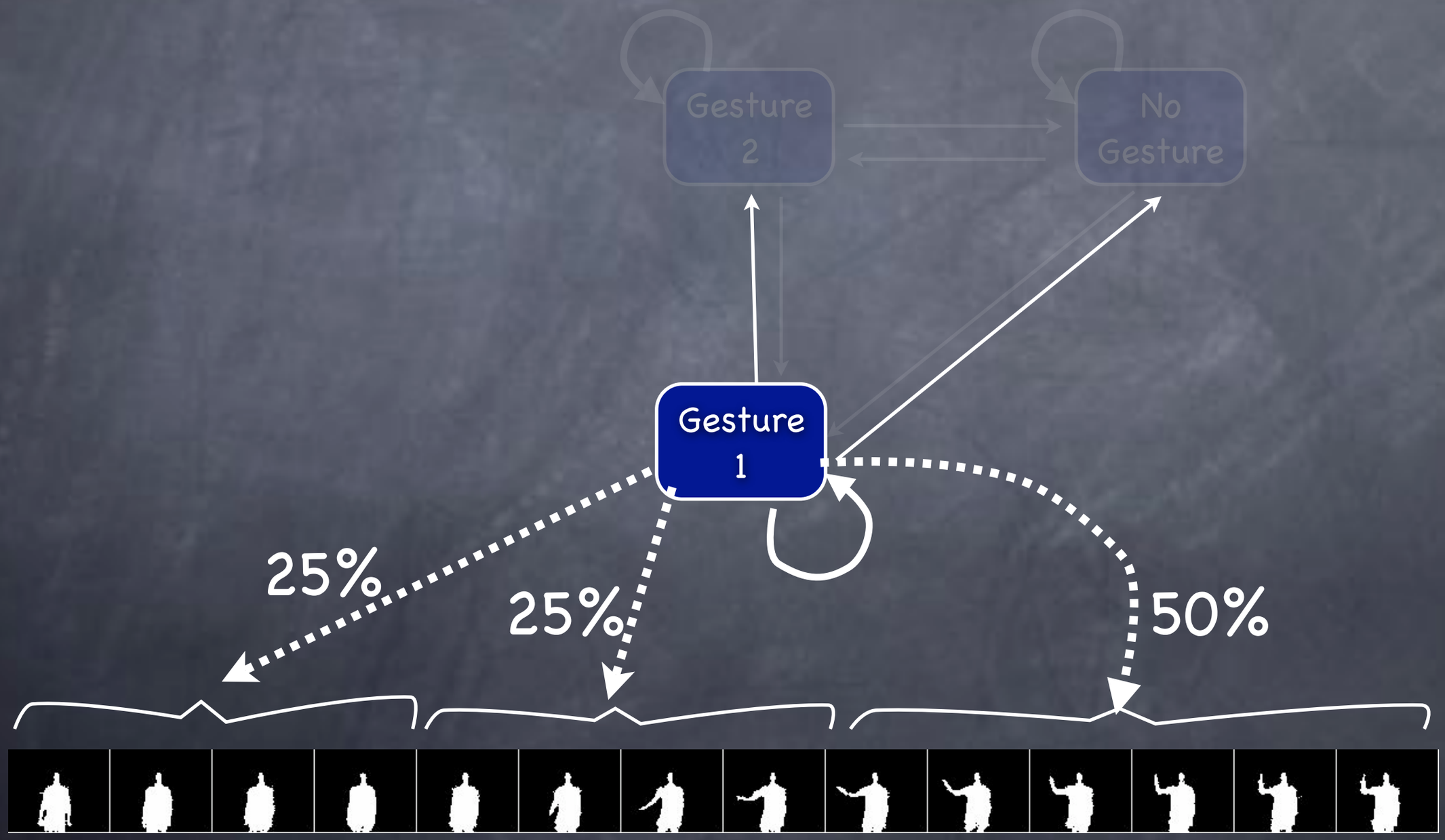

- Hidden Markov Model



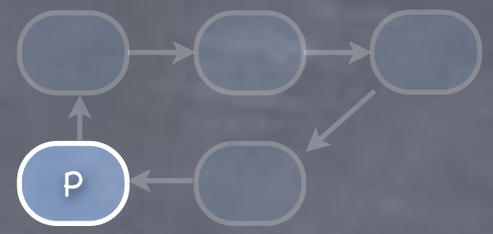
# Perception



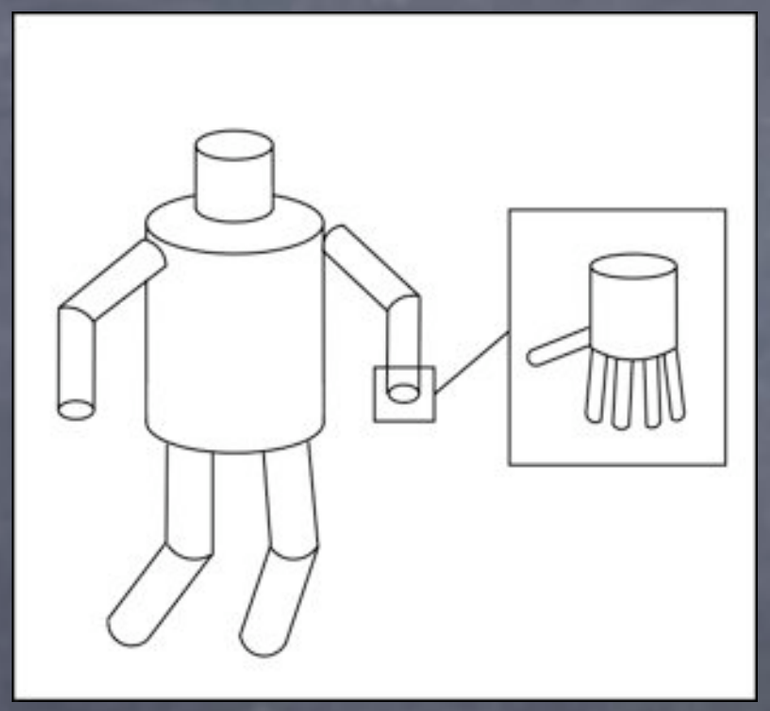

- Hidden Markov Model



# Getting features




- I use a cylindrical body model (ok, not this one)



- And I use "guess and check", aka Bayesian reasoning
- Gives me poses, which give me features

# Demo