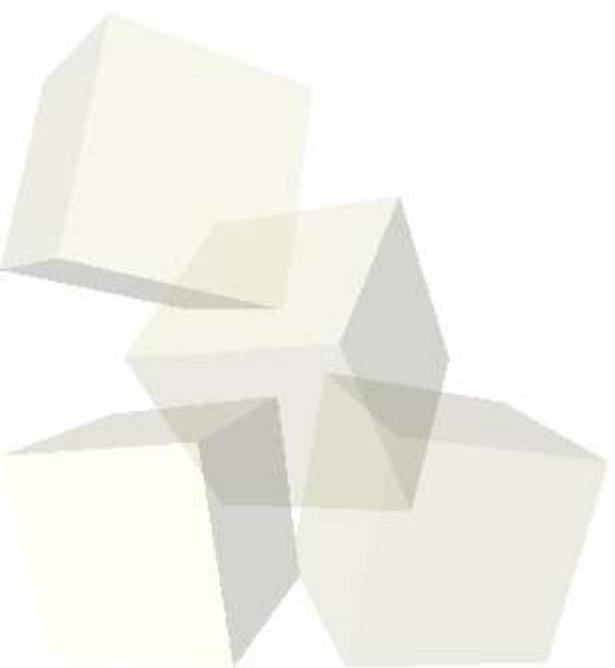




# Recognizing Action At A Distance

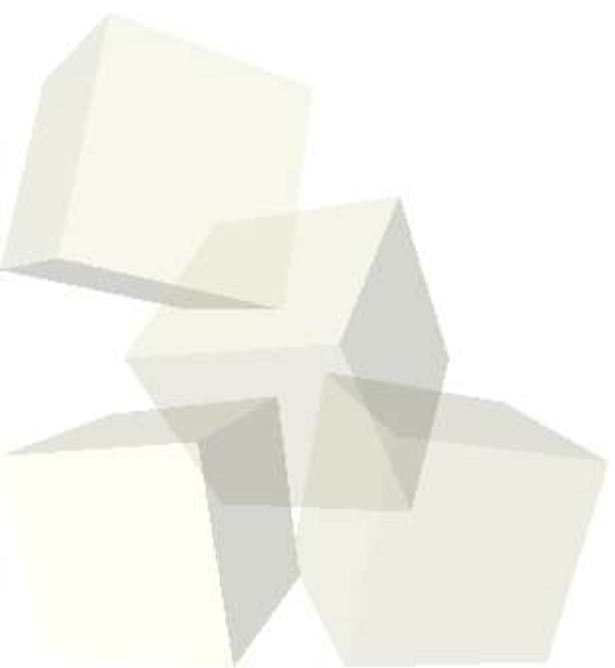
Alexei A. Efros, et Al.

Presented by: Sunny Chow



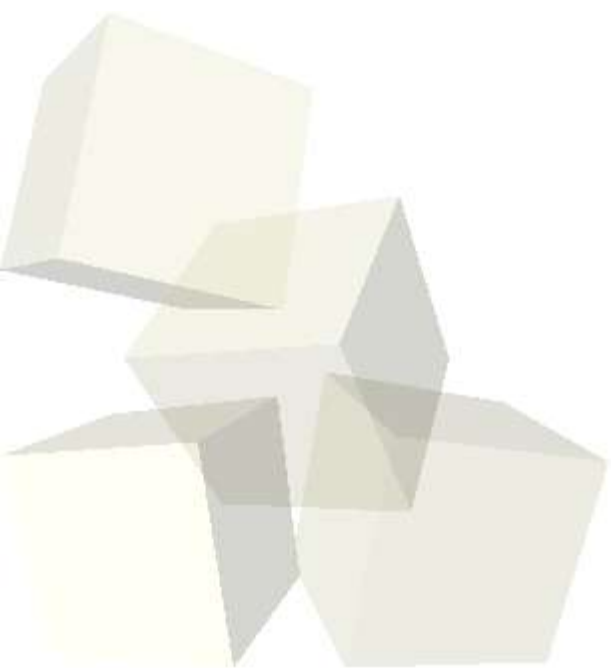


- We are adept at classifying actions.
  - ◆ Easily categorize even with noisy and small images
- Want computers to do just as well
- How do we do it?



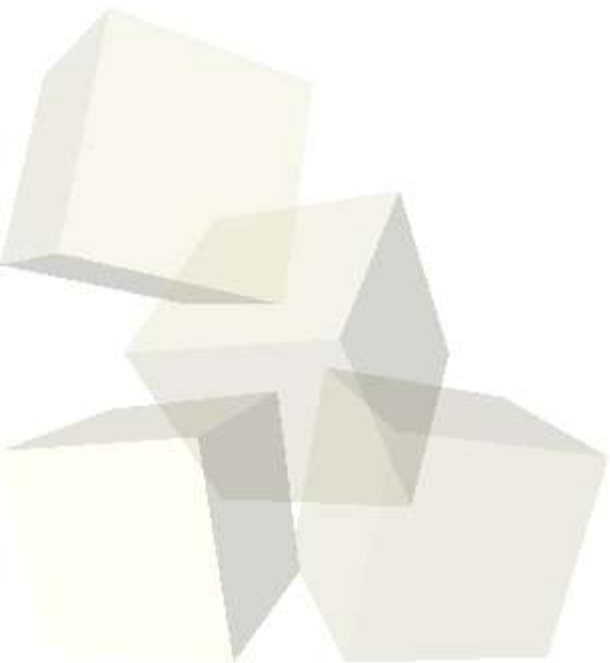


- Possible applications for action recognition
  - ◆ Obvious
    - Tracking people's activities in public places
  - ◆ Less obvious
    - Use classification to solve a harder problem
      - Put a skeletal model over the novel sequence
      - Synthesize actions





- Action classification has been attempted in the past, with different assumptions
  - ◆ Most work in nearfield
    - Shah and Jain – Track Body Works
  - ◆ Motion periodicity
    - Cutler and Davis – Poor quality moving footage





## ■ Assumptions

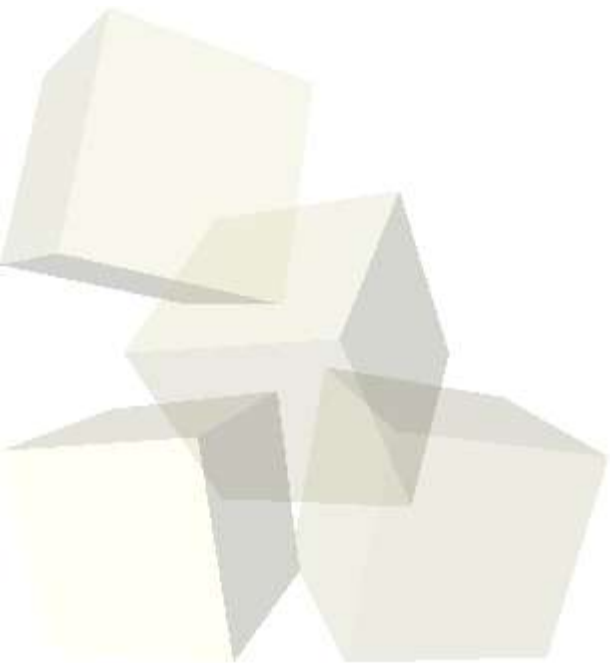
- ◆ Tracking and Image Stabilization is taken care of.
- ◆ Figure-centric sequence of images as input
- ◆ Human actions

## ■ Conditions

- ◆ Image sequence from mid-field
- ◆ Different start and End points
- ◆ Different rate of motions
- ◆ Independence of appearance
  - Actor
  - background



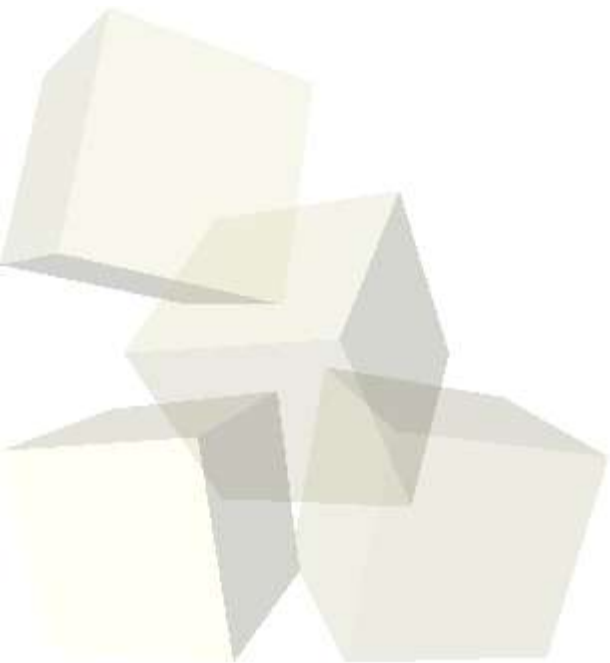
- Comparison between novel and classified, stored images
- Need to choose representation
- Based on optical flow
- Spatial-Temporal Descriptor



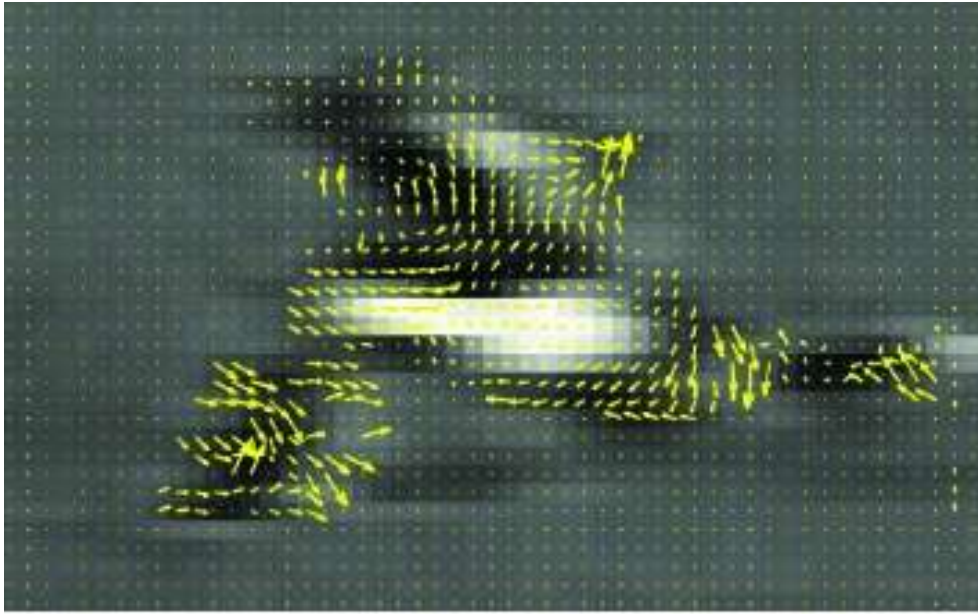


# Quick Review of Optical Flow

- Given: two frames of a video scene closely separated in time.
- Goal: Get motion of each pixel.
- Motion field, noisy.
  - ◆ Certain measurements are better than others.



# Quick Review of Optical Flow 2



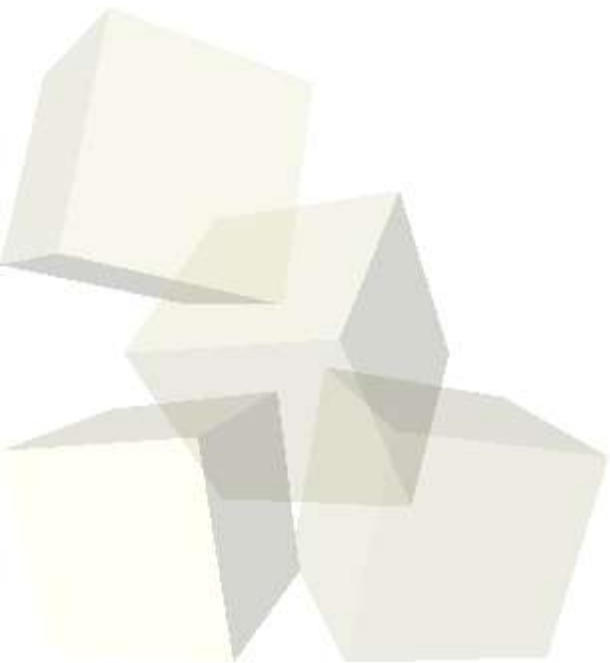
(b) optical flow  $F_{x,y}$

- Measure only relative motion between frames.
- Indifferent to actual appearance.
- Failure modes
  - ◆ Specularities sit still
  - ◆ Large displacements



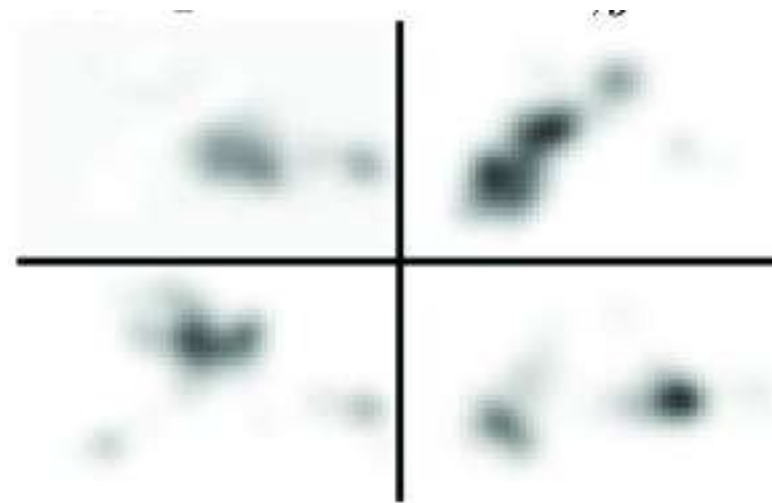
# Problems with Optical Flow

- 1. Data is noisy
  - ◆ Novel idea: Treat vectors as “noisy measurements” which can be added up later
- 2. Data may not be properly aligned in space/time
  - ◆ Just blur.
  - ◆ Treat positive values and negative values separately.

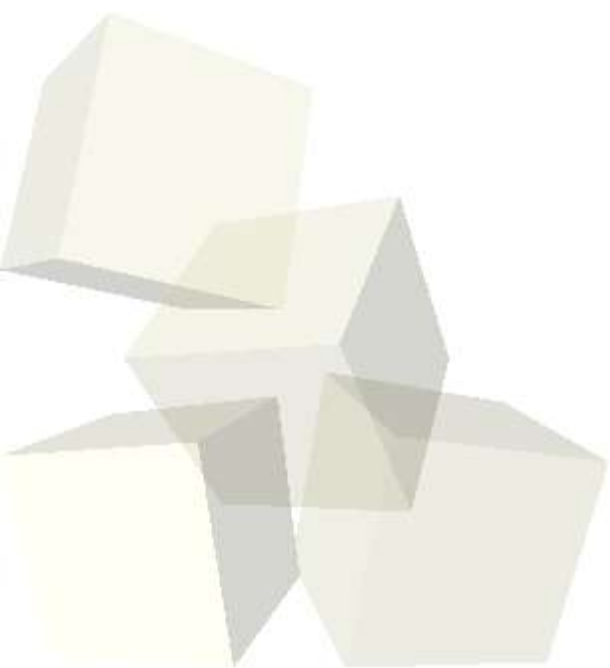




- Spatial-Temporal descriptor
  - ◆ 4 channels per image in a sequence
    - Gradients in X and Y separated into positive and negative channels.



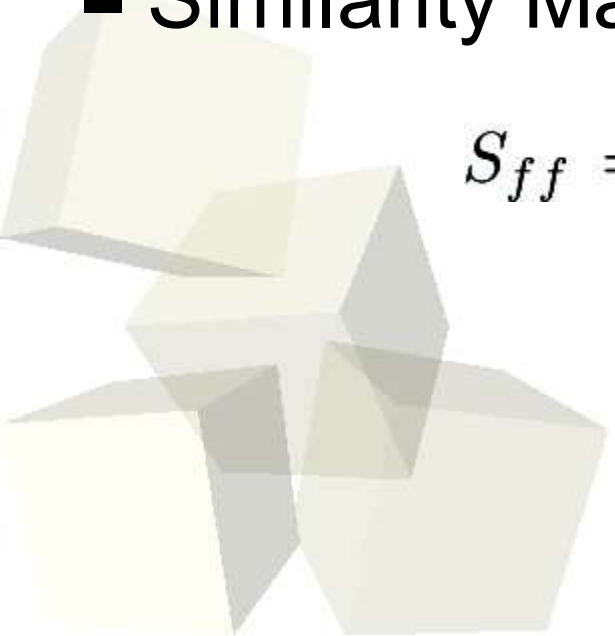
(e)  $Fb_x^+$ ,  $Fb_x^-$ ,  $Fb_y^+$ ,  $Fb_y^-$





- Use normalized correlation to compare motion descriptors
- Interested in sequence of images.
  - ◆ Start and end of novel sequence unknown
  - ◆ Rate of action unknown
- String channels from the sequence together
- Similarity Matrix:

$$S_{ff} = A_1^T B_1 + A_2^T B_2 + A_3^T B_3 + A_4^T B_4$$

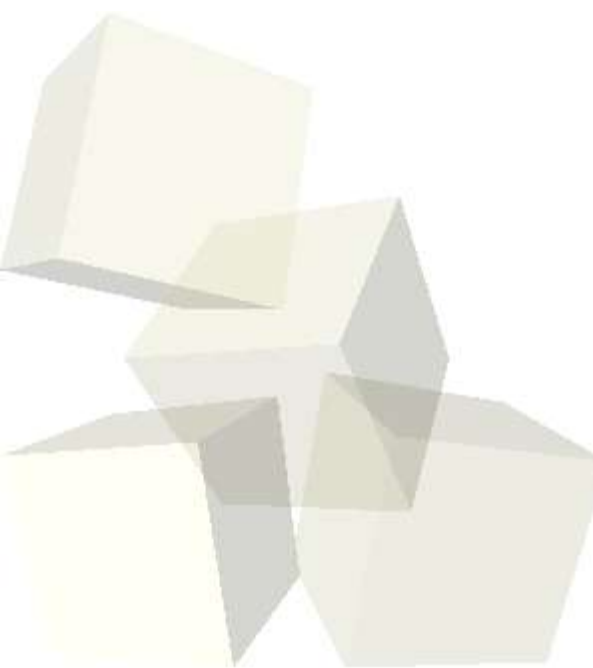




# Comparison Intuition

- Consider one channel at a time.
  - ♦ Same rate, different starting times.
  - ♦ Suppose a started at 1, b started at 2

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{21} & b_{31} & b_{41} & b_{51} \\ b_{12} & b_{22} & b_{32} & b_{42} & b_{52} \\ b_{13} & b_{23} & b_{33} & b_{43} & b_{53} \\ b_{14} & b_{24} & b_{34} & b_{44} & b_{54} \\ b_{15} & b_{25} & b_{35} & b_{45} & b_{55} \end{bmatrix} = \begin{bmatrix} (1, 1) & (1, 2) & (1, 3) & (1, 4) & (1, 5) \\ (2, 1) & (2, 2) & (2, 3) & (2, 4) & (2, 5) \\ (3, 1) & (3, 2) & (3, 3) & (3, 4) & (3, 5) \\ (4, 1) & (4, 2) & (4, 3) & (4, 4) & (4, 5) \\ (5, 1) & (5, 2) & (5, 3) & (5, 4) & (5, 4) \end{bmatrix}$$



$$\begin{bmatrix} 0 & S & 0 & 0 & 0 \\ 0 & 0 & S & 0 & 0 \\ 0 & 0 & 0 & S & 0 \\ 0 & 0 & 0 & 0 & S \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



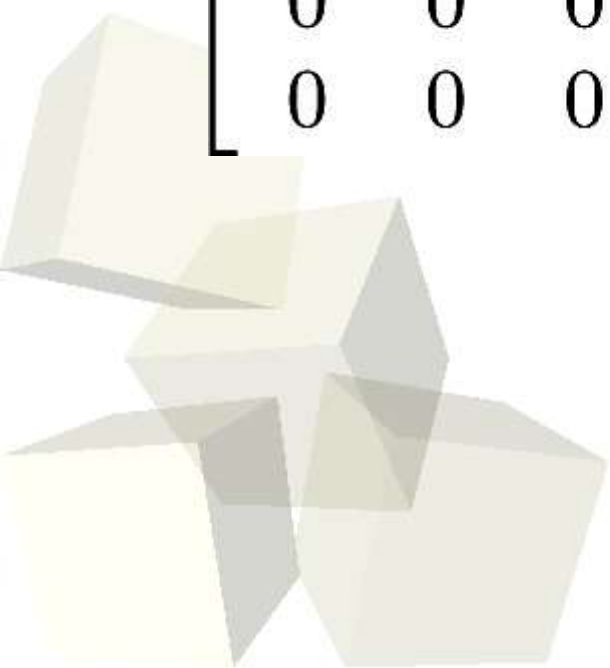
# Comparison Intuition 2

- Different rates, use “Blurry Identity” kernel

$$\begin{bmatrix} S & 0 & 0 & 0 & 0 \\ 0 & S & S & 0 & 0 \\ 0 & 0 & 0 & S & S \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



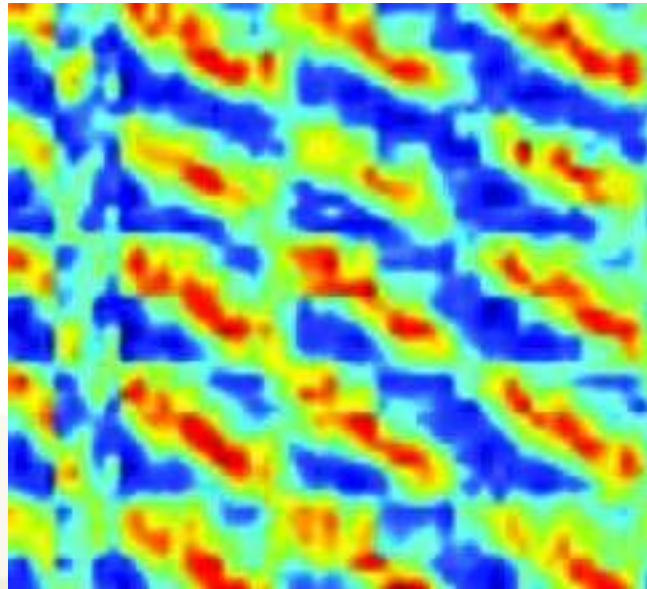
$$K(i, j) = \sum_{r \in R} w(r) \chi(i, rj)$$



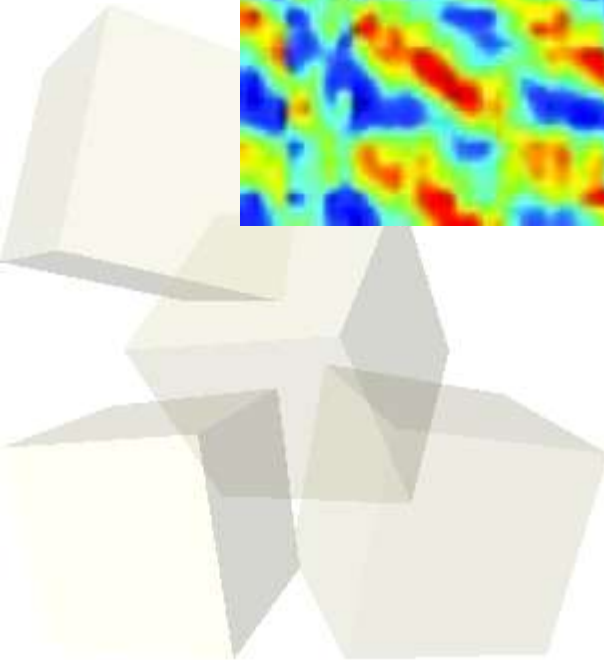
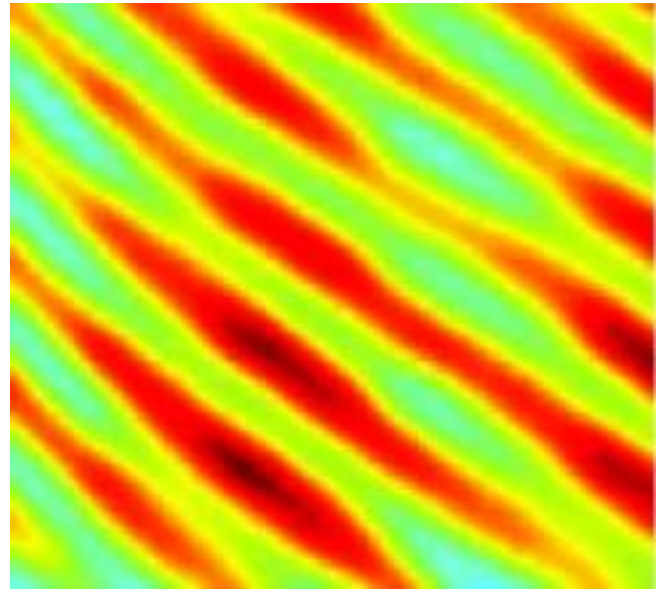


# Comparison

■ S\_ff

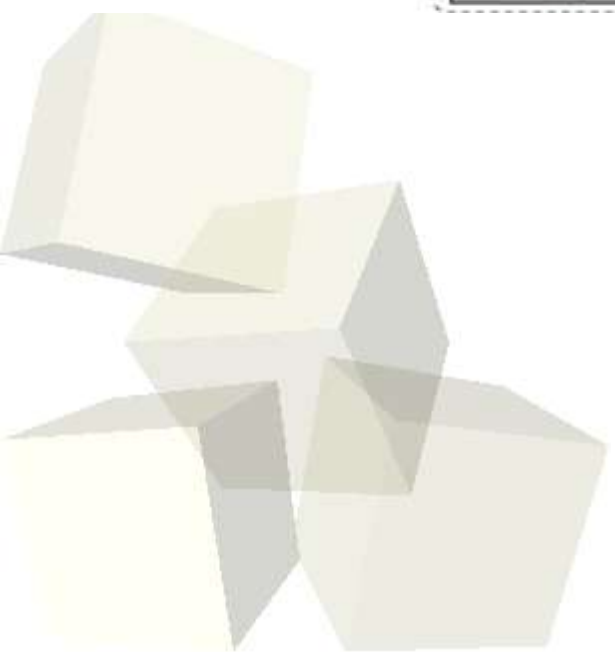
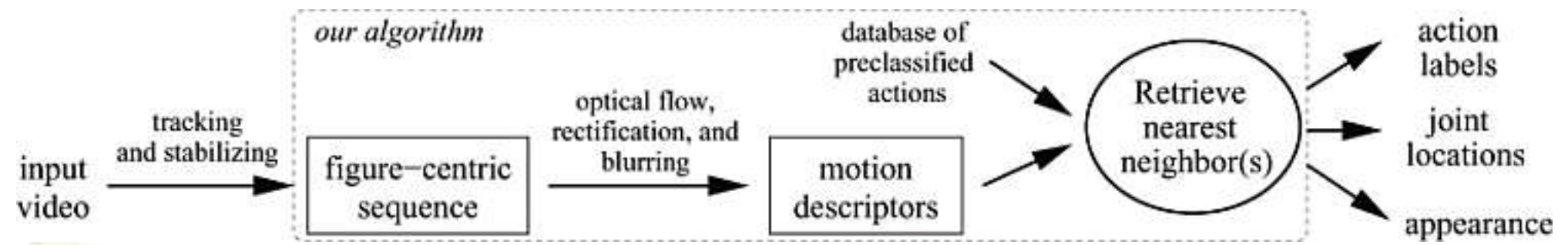


■ Final Similiarity Matrix



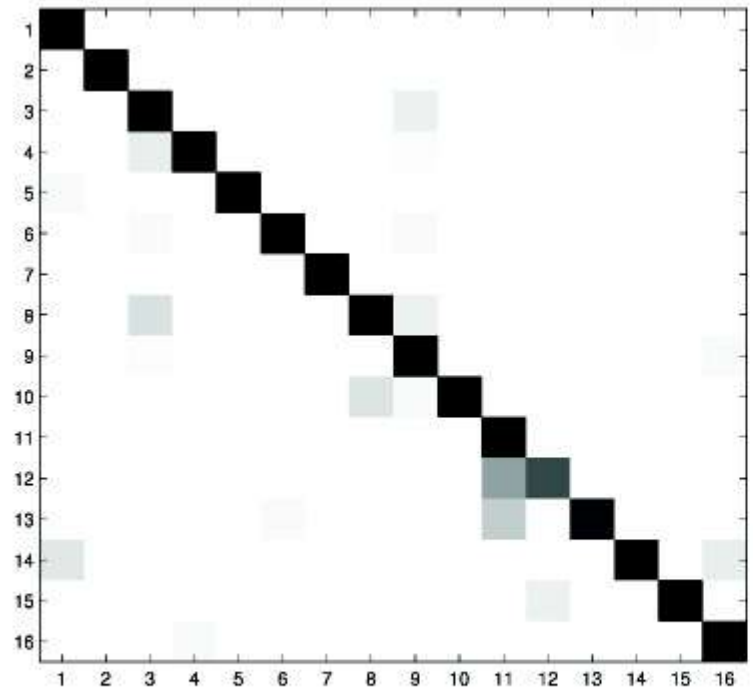


# Algorithm Outline

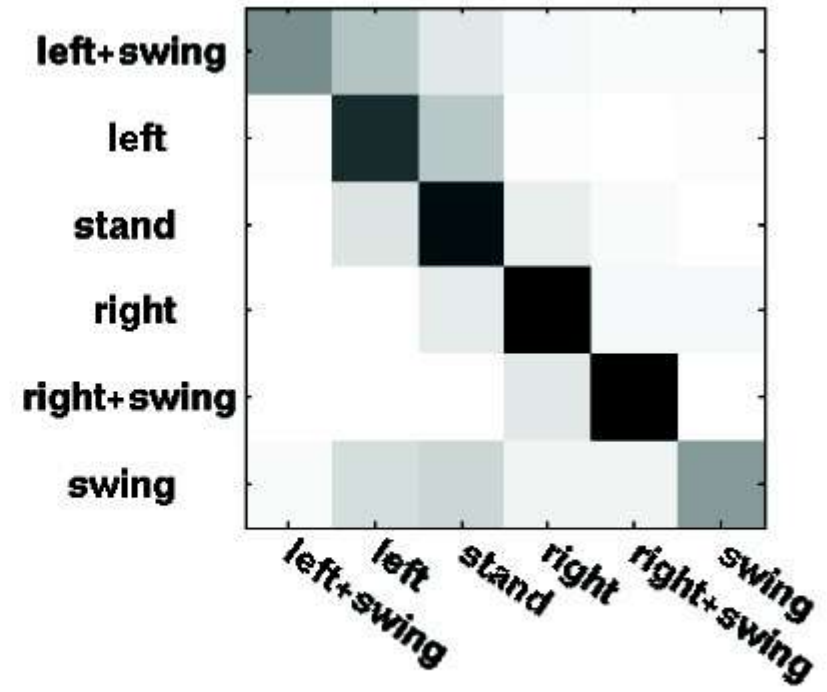




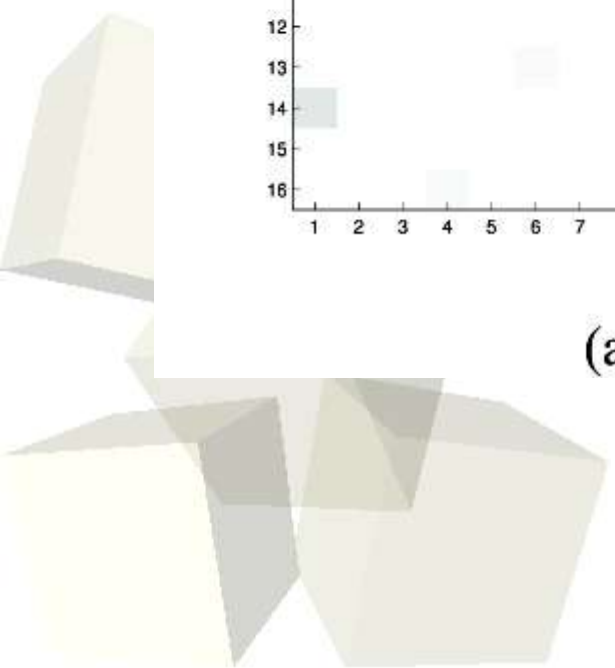
## ■ Test Sequences for Ballet and Tennis



(a) Ballet

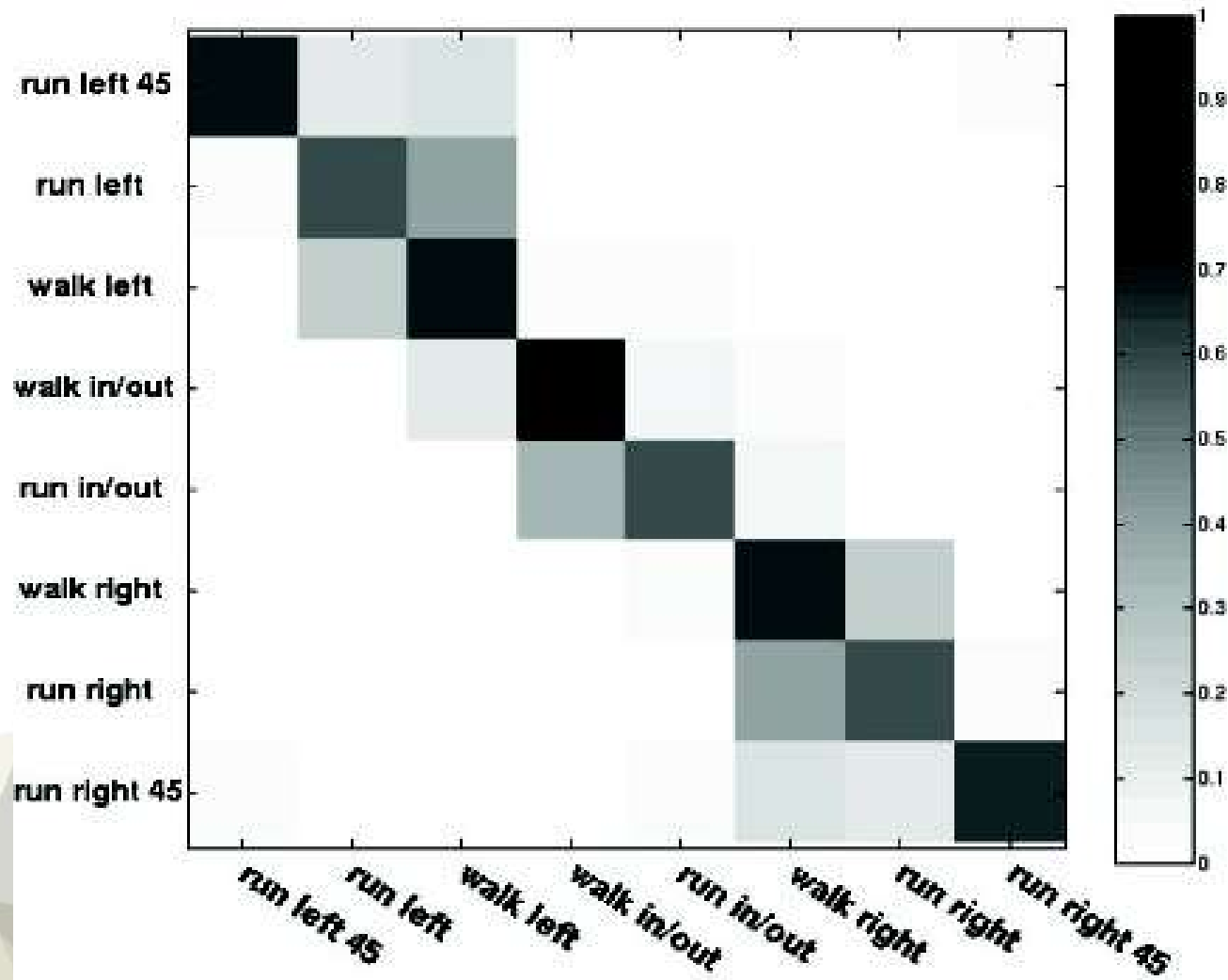


(b) Tennis





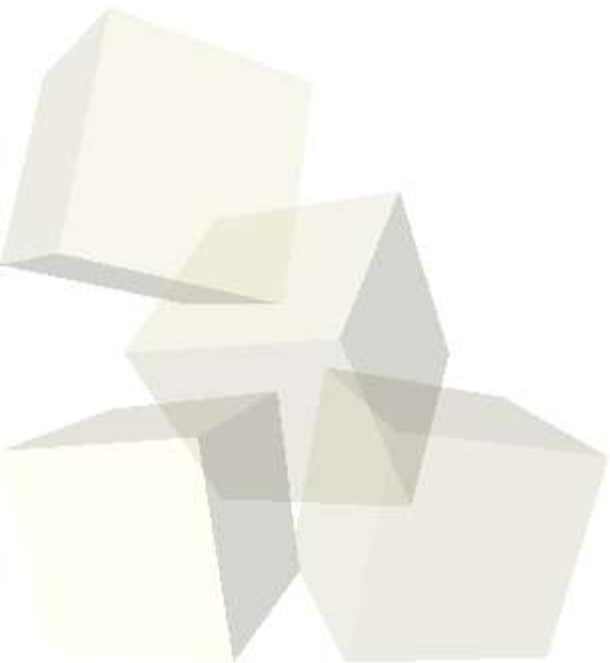
## ■ Test Sequence for Football



(c) Football



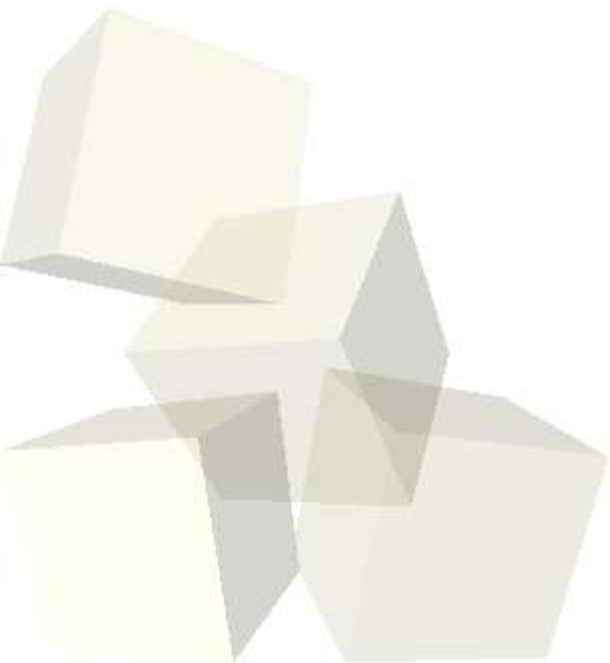
- Do as I do...
  - ◆ Query with novel action sequence, create a similar sequence using stored data





## ■ Do as I Say

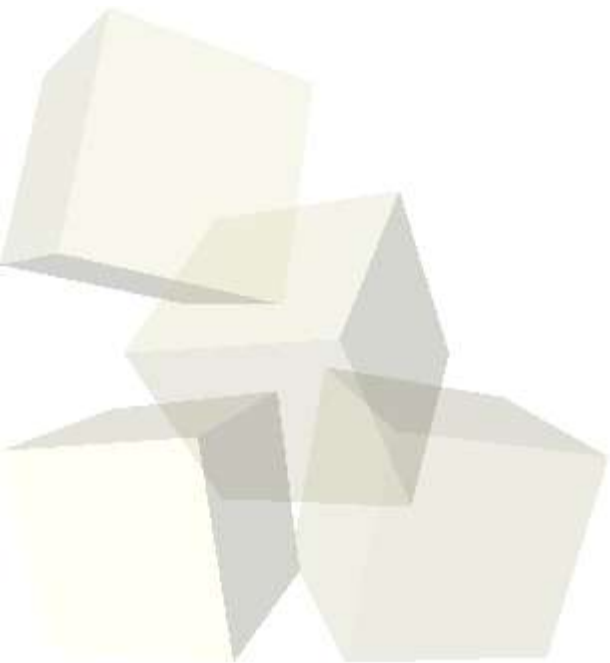
- ◆ Query with action identifier (english description), create an action sequence.
- ◆ Think Mortal Kombat





# Additional Applications

- Skeletal Model
- Figure Correction
  - ◆ Find stored motion descriptor closest to data
  - ◆ Common parts: what we're interested in
  - ◆ Variations: noise occlusion. Use to correct





- Novel observation, optical flow can be treated as noisy measurements
- Create spatial-temporal descriptor to represent action
- Use descriptor as a query into a database of classified actions to classify novel action
- Use database to solve harder representation problems



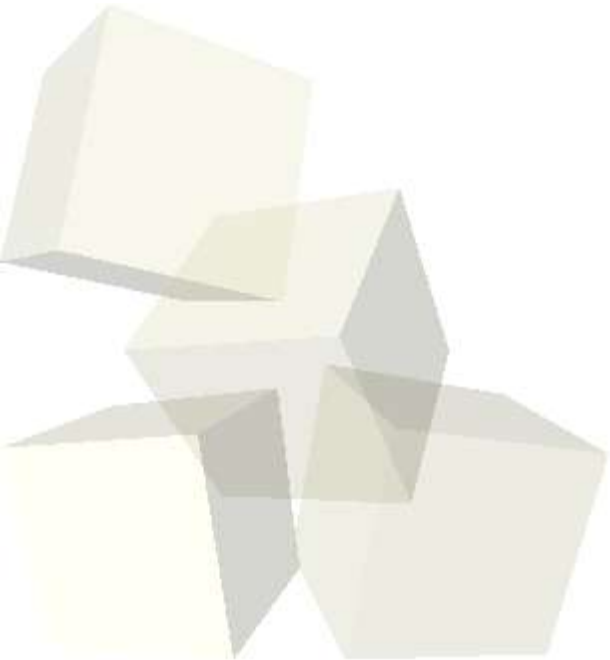


# Unanswered Questions

- Querying into database seems computationally expensive.
- Unclear on granularity of representation of the motion descriptors
- How well does this algorithm compare to a human's ability to classify actions?
- How to determine the size of temporal window?
- How much does background movement affect the results?



But that's not all, folks, wait and see what else you will get!





# 2 for 1 special, today only!

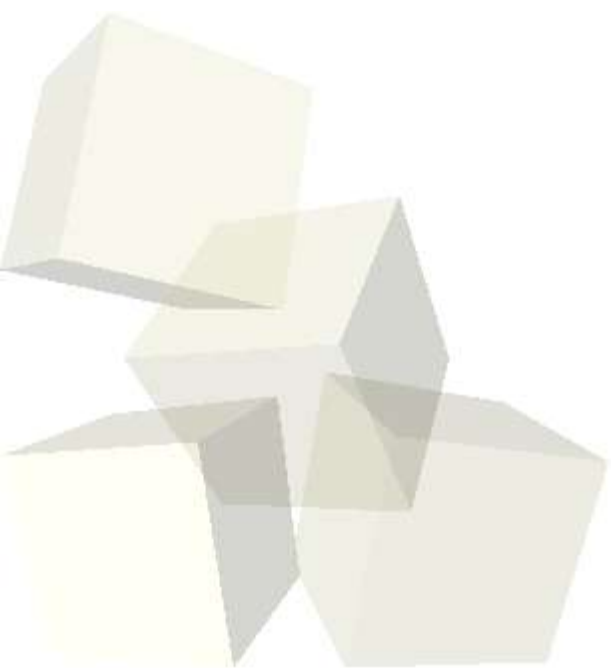
- Detecting Pedestrians Using Patterns of Motion and Appearance
  - ◆ Paul Viola, Michael Jones, et al.





# Huh? What is this about?

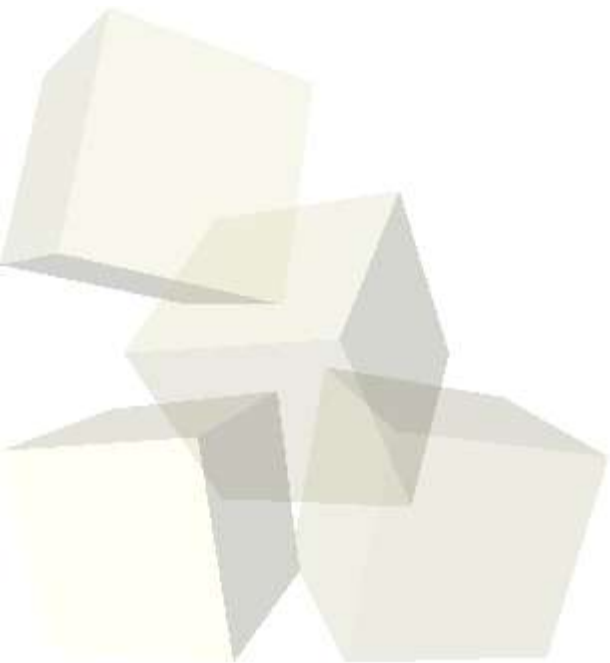
- Allows detection of specific features in an image
- Feature of interest: moving pedestrians
  - ◆ Detects pedestrian as small as 20x15
- Extremely fast, 15 fps





# So what's different?

- No tracking or stabilization assumptions
- Will detect only moving pedestrians
- Static image
- Uses only short term patterns of motion





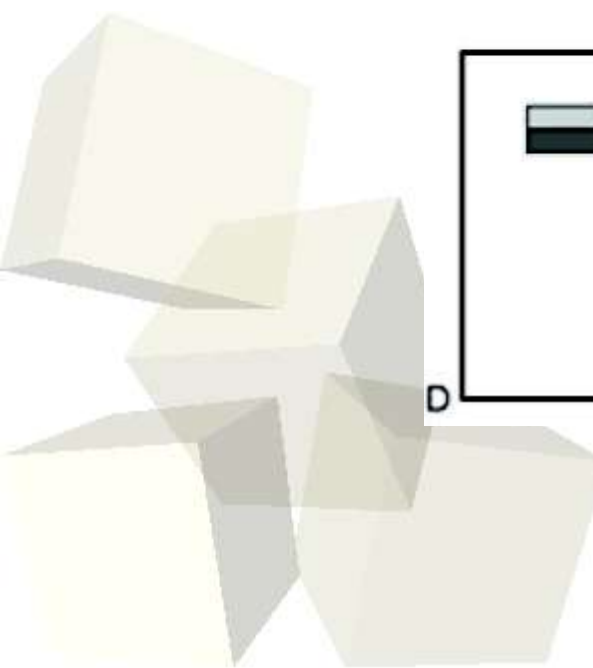
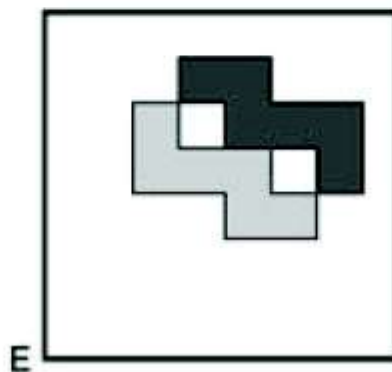
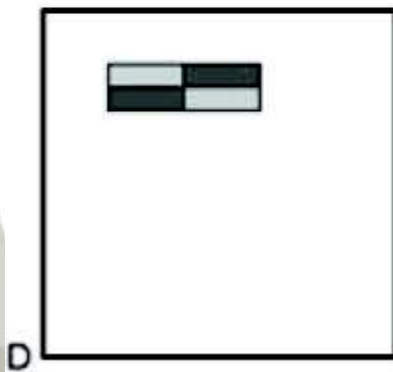
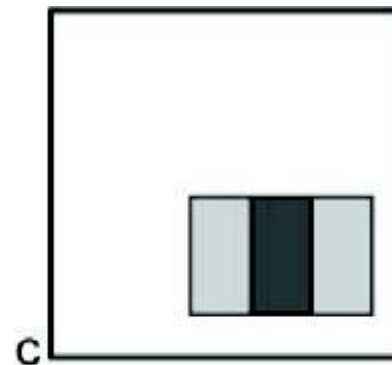
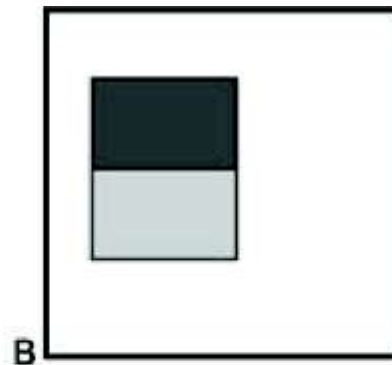
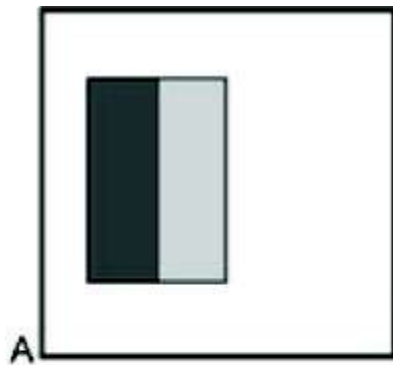
# High level summary of methods

- Based largely on previous work, "Rapid Object Detection using a Boosted Cascade of Simple Features"
  - ◆ Primary purpose: detecting faces from a picture
- 3 parts:
  - ◆ "Integral Image"
  - ◆ Learning algorithm based on "AdaBoost"
  - ◆ Combining increasingly complex classifiers into a cascade.



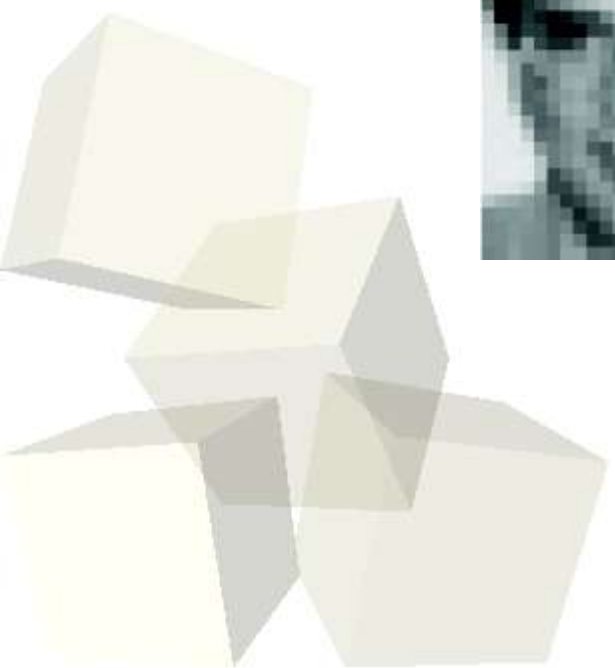
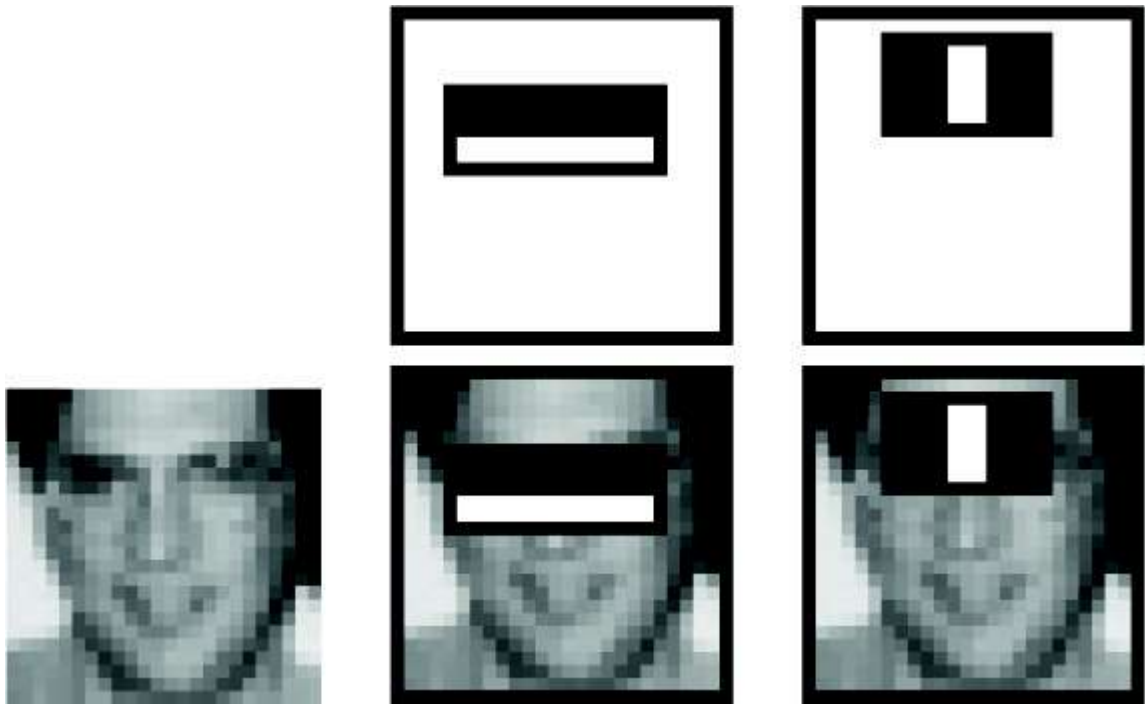
## ■ Features represented as filters

- ◆ Simple
- ◆ Scale easily





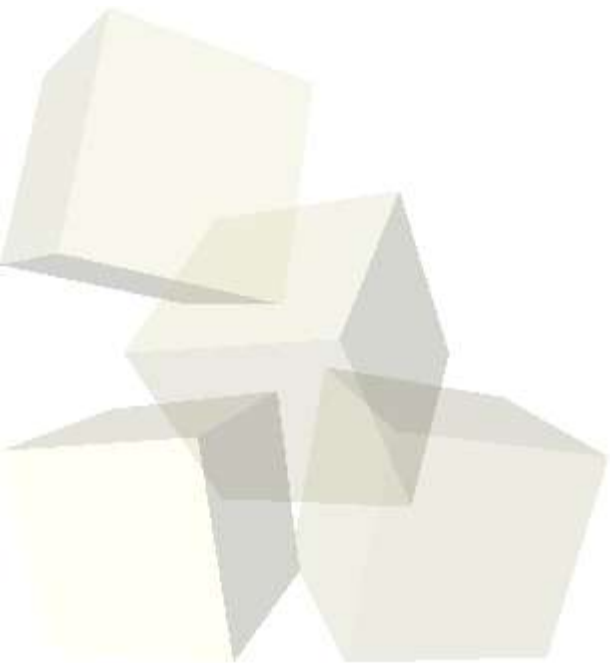
- Filter intuition





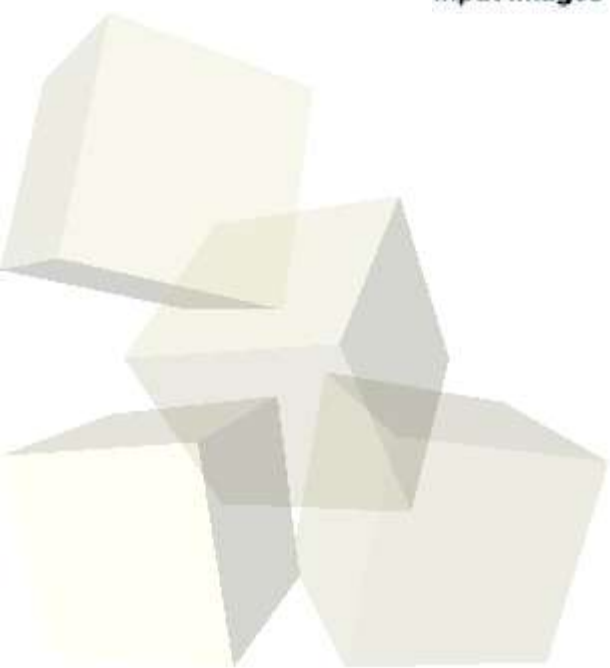
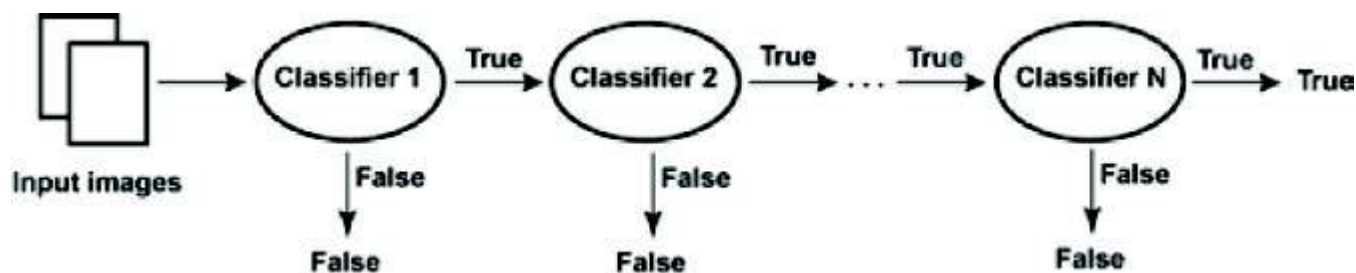
# Filter application

- Use these filters to classify both motion & intensity
- Use AdaBoost to combine various filters into classifiers
  - ◆ Goal: balance intensity, motion information, maximize detection rates



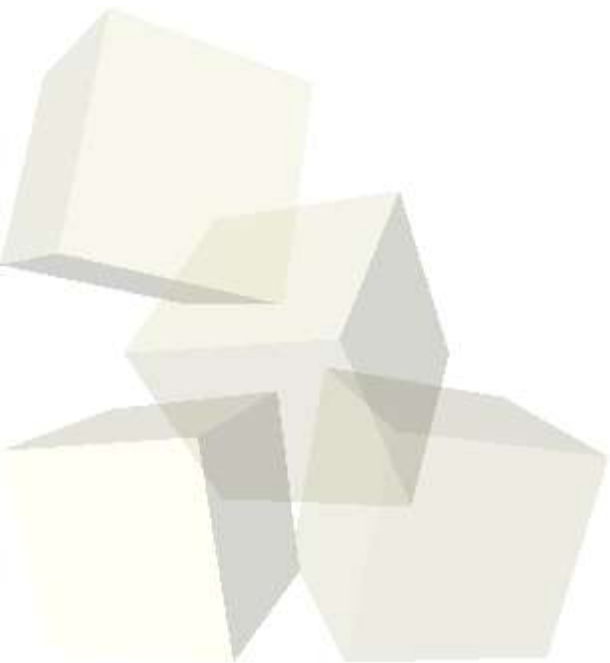


- String classifiers together
- Simple to Complex
- Simple: weed out things that look nothing like what we're interested in.



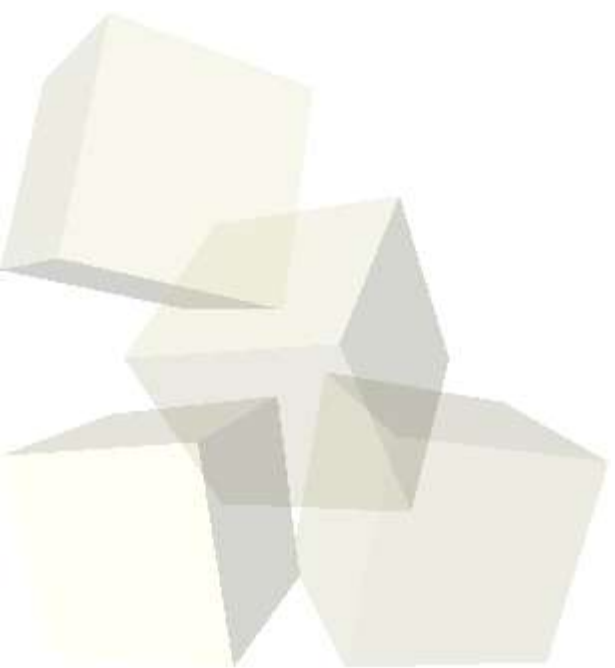
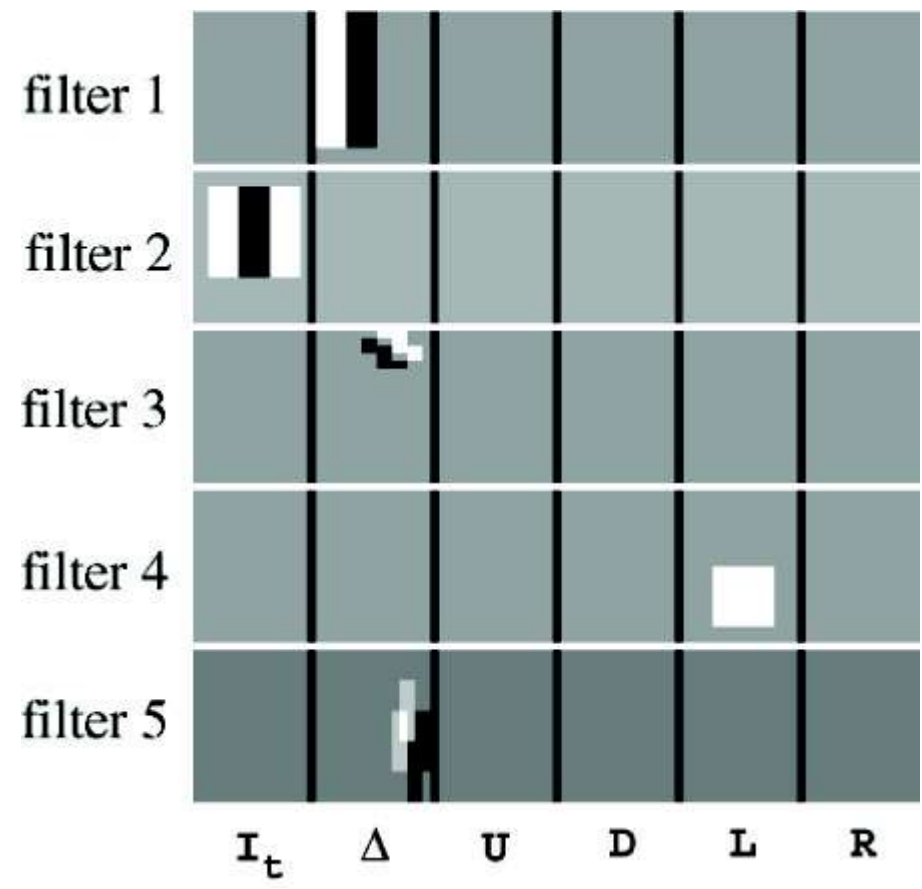


- For each stage, since simple to complex
- Both false positive rates and detection rates decrease
- Trick: get false positive rates to decrease faster than detection rate.



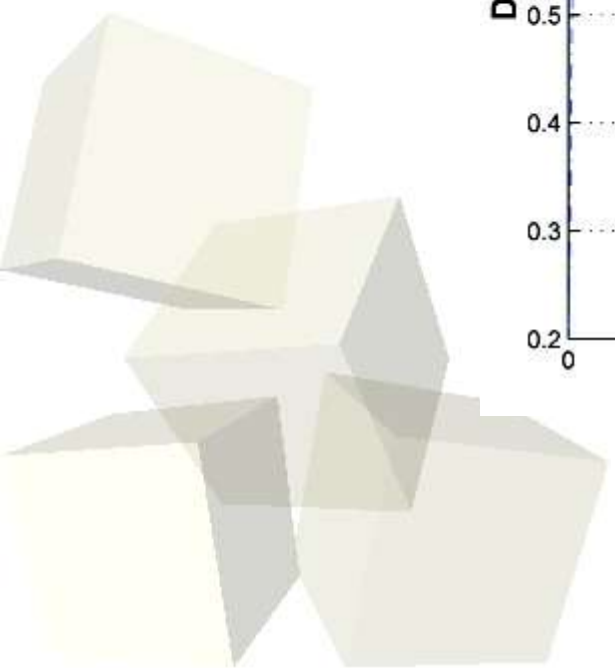
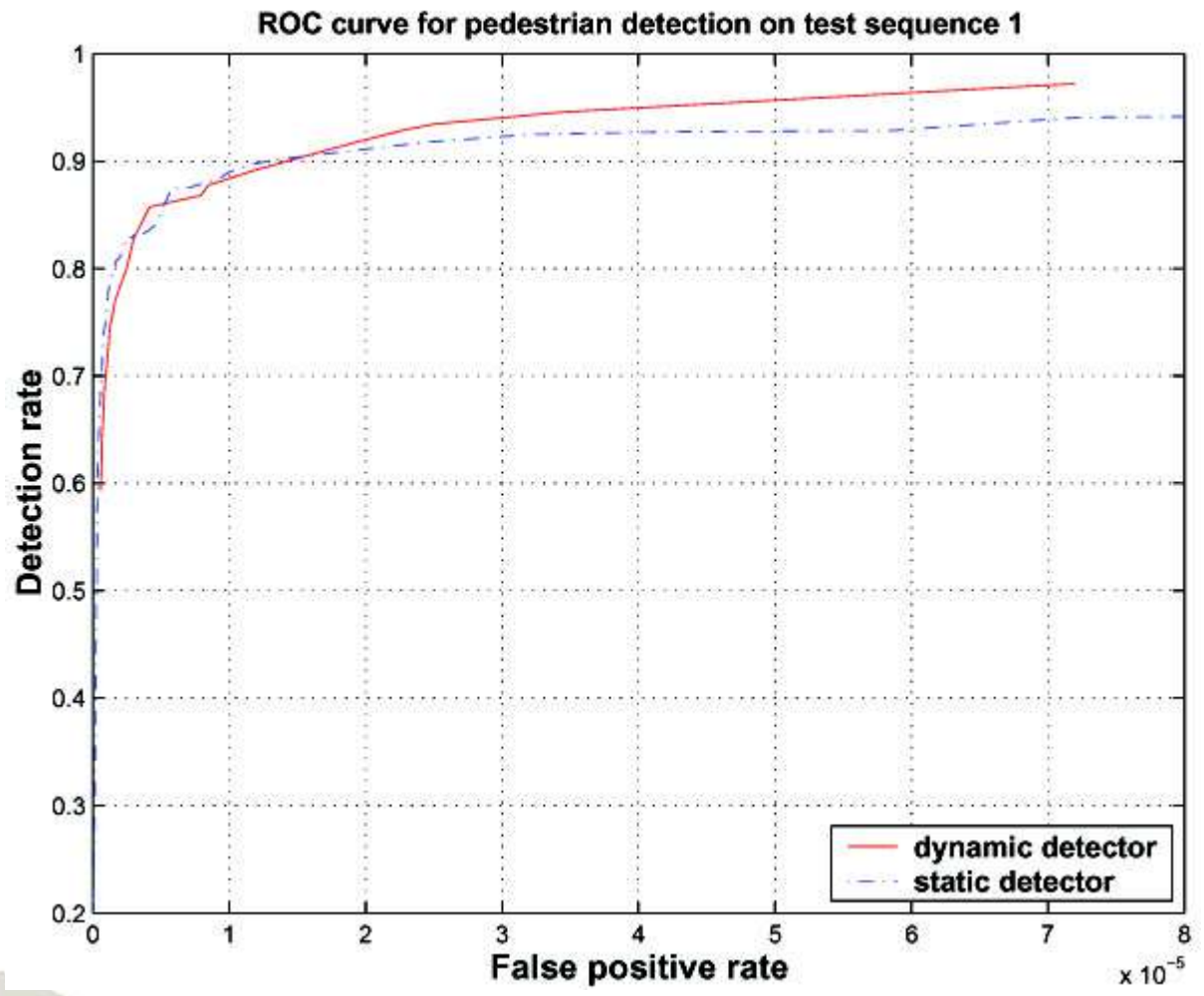


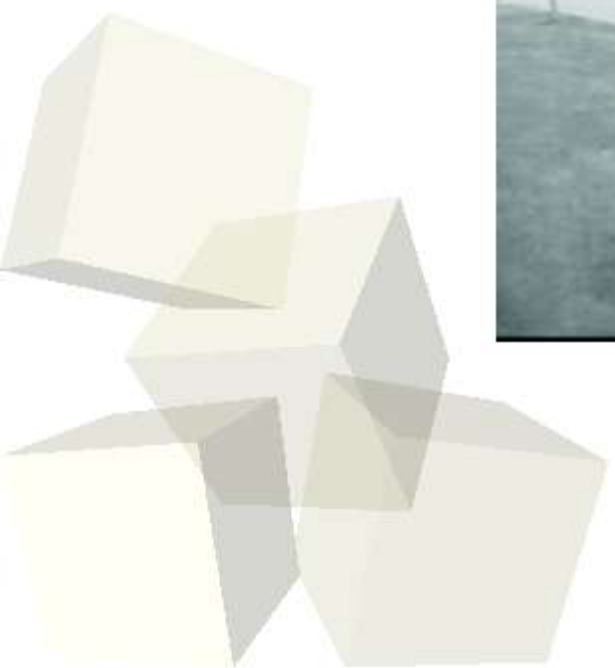
# Classifier Intuition





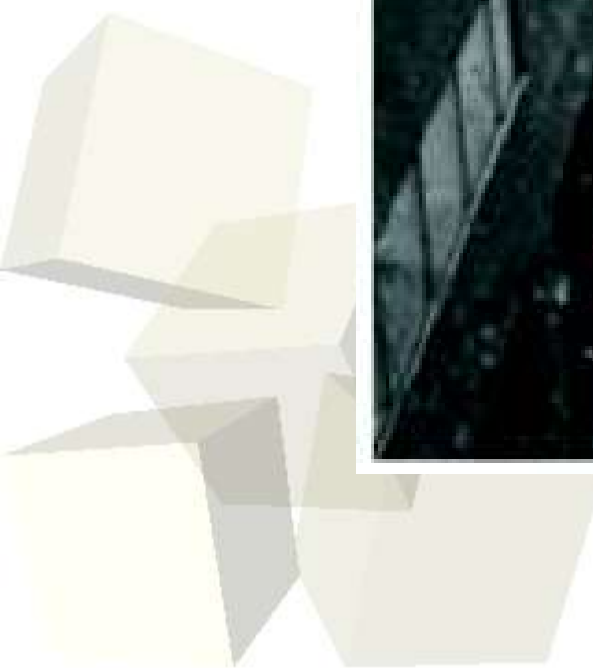
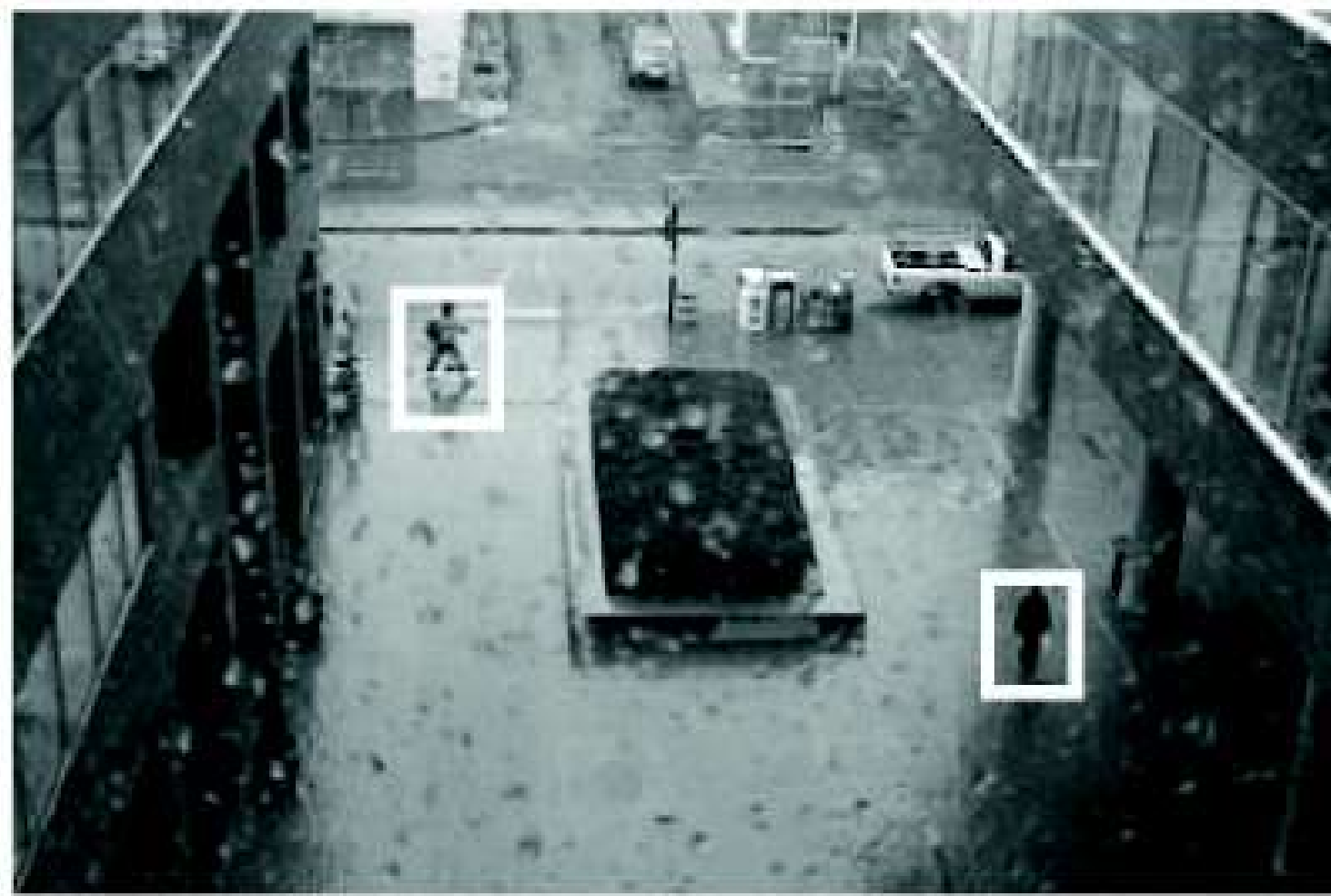
# Accuracy







- Through rain or snow...





Thanks for your time!

