Computational Vision

Lecture 2.2: Motion

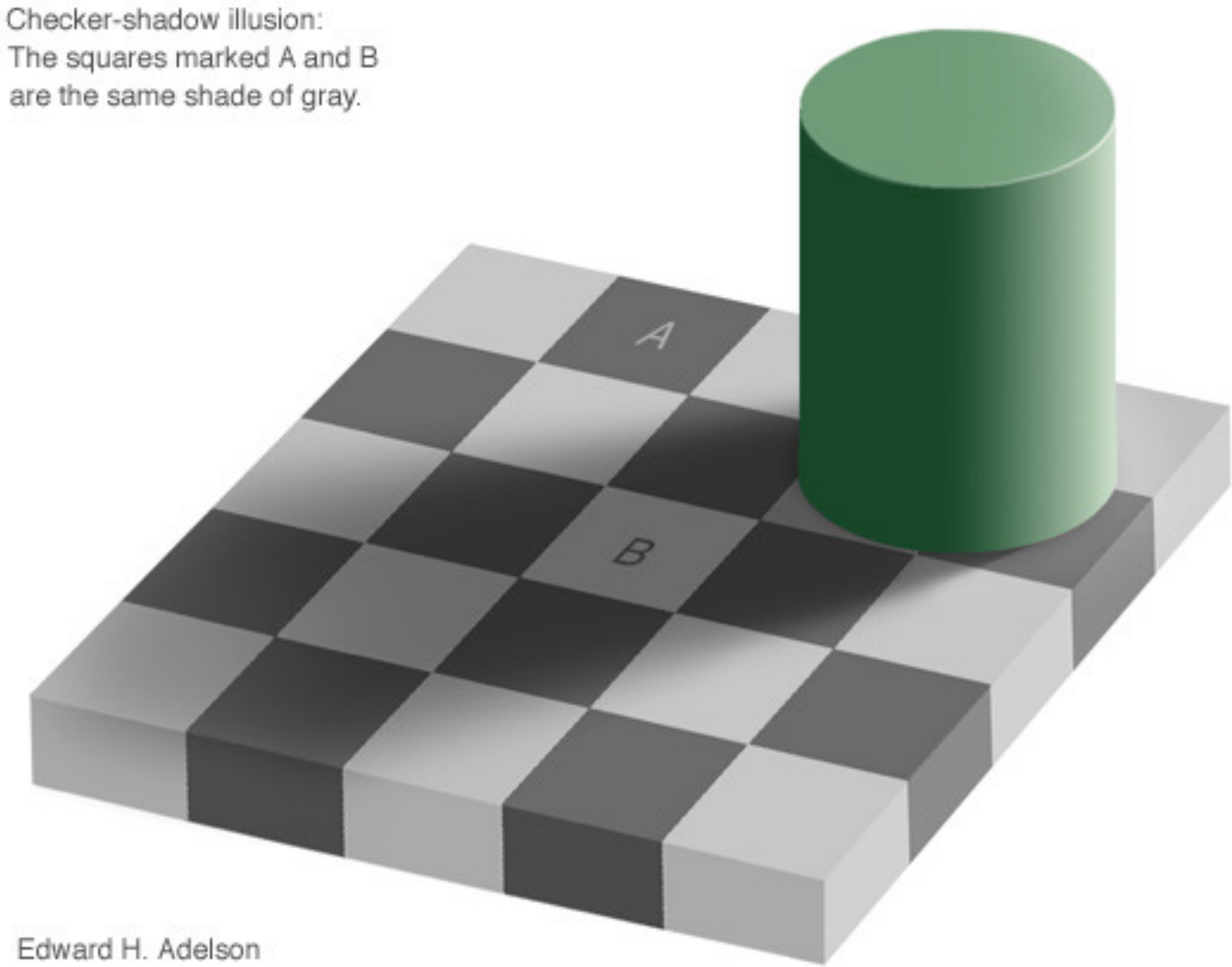
Hamid Dehghani

Office: CS 241

Why is Vision Interesting?

- Psychology
  - ~ 35% of cerebral cortex is for vision.
  - Vision is how we experience the world.
- Engineering
  - Want machines to interact with world.
  - Digital images are everywhere.

Vision is inferential: Light



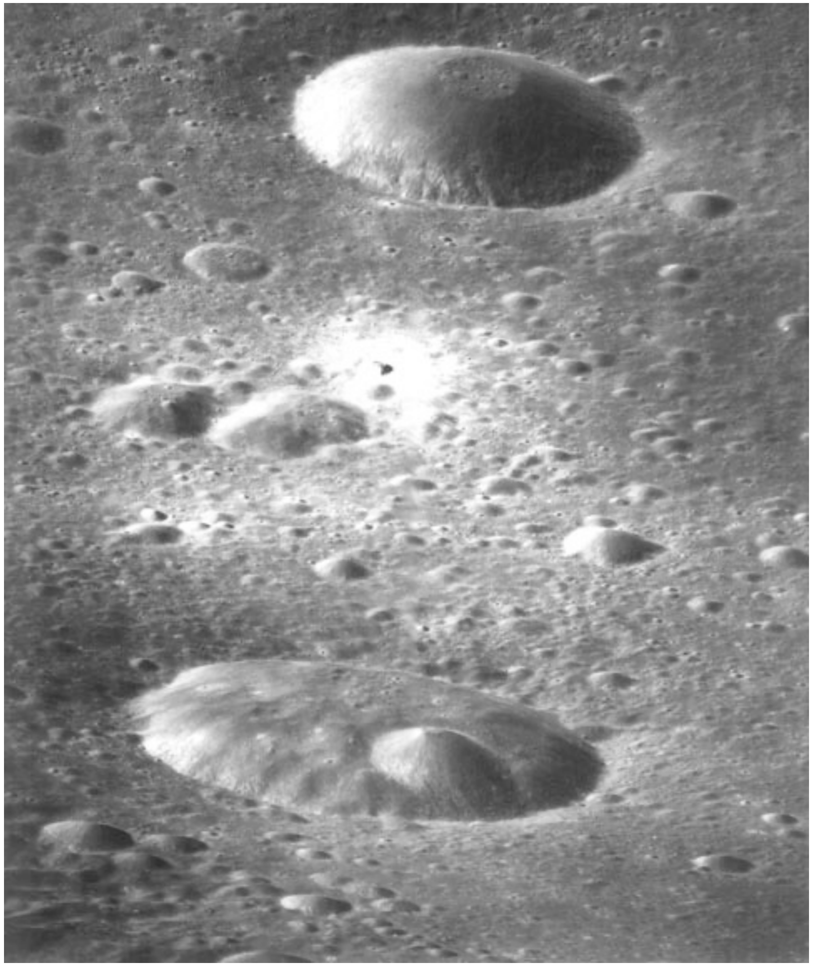
(http://www-bcs.mit.edu/people/adelson/checkershadow_illusion.html)

Vision is inferential: Light

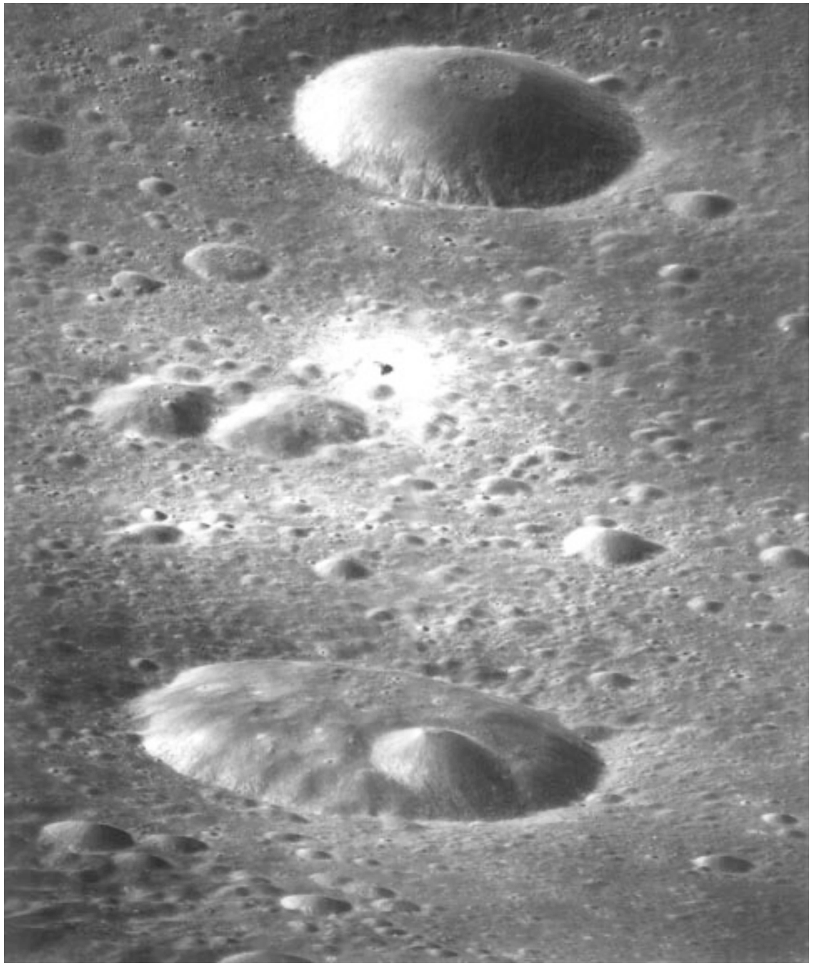
Checker-shadow illusion:
The squares marked A and B
are the same shade of gray.

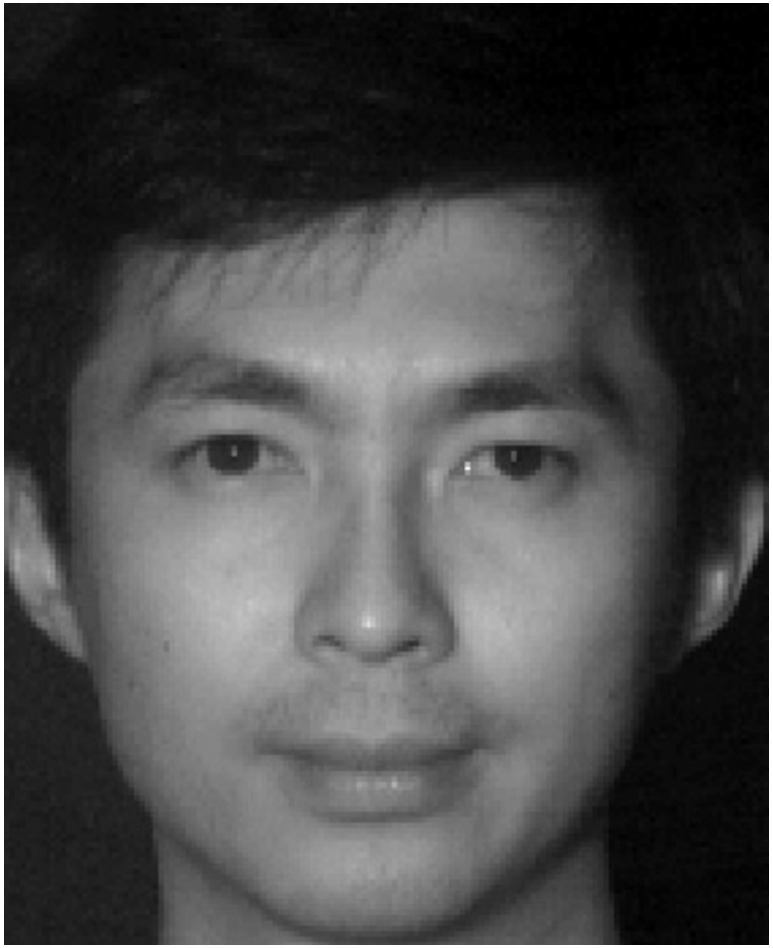
(http://www-bcs.mit.edu/people/adelson/checkershadow_illusion.html)

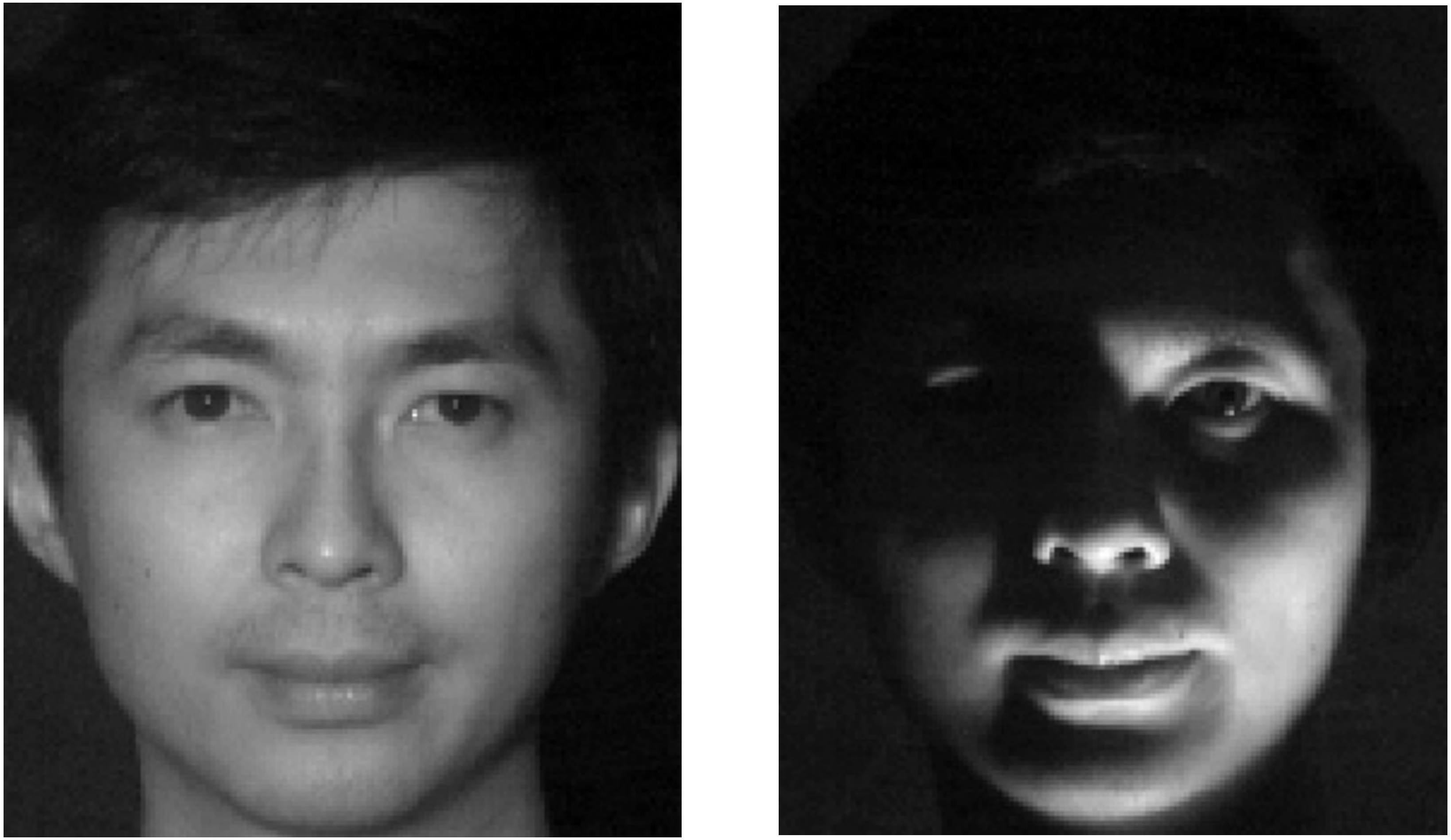
Vision is Inferential: Prior Knowledge



Vision is Inferential: Prior Knowledge

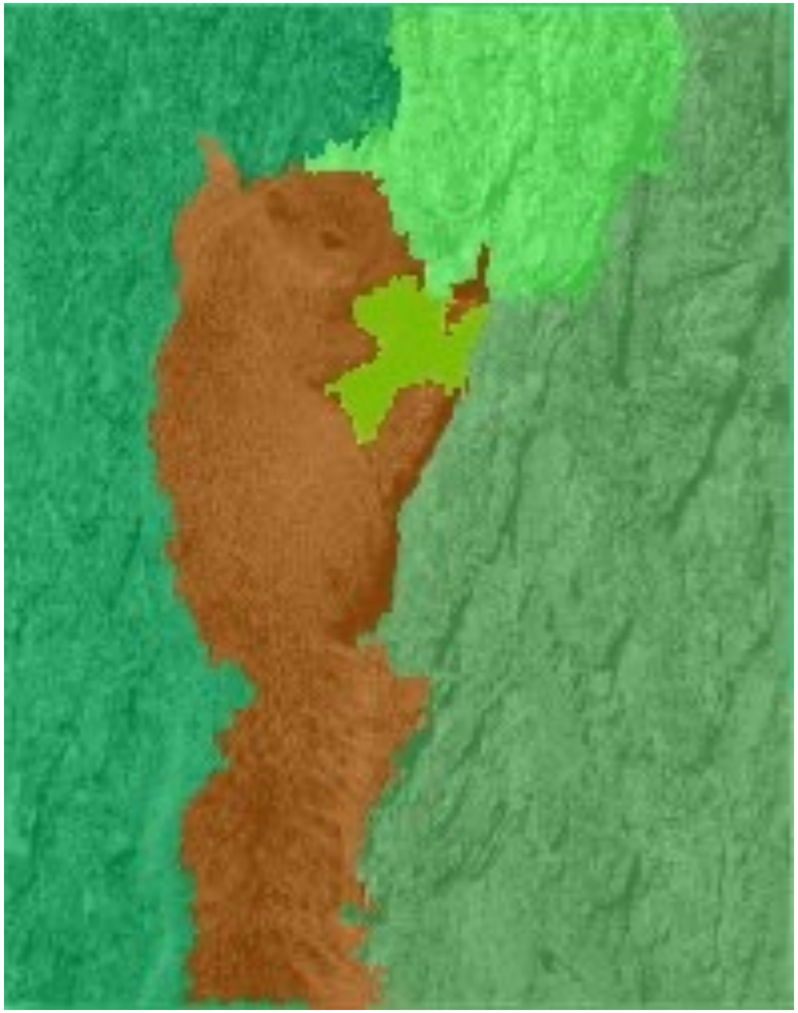




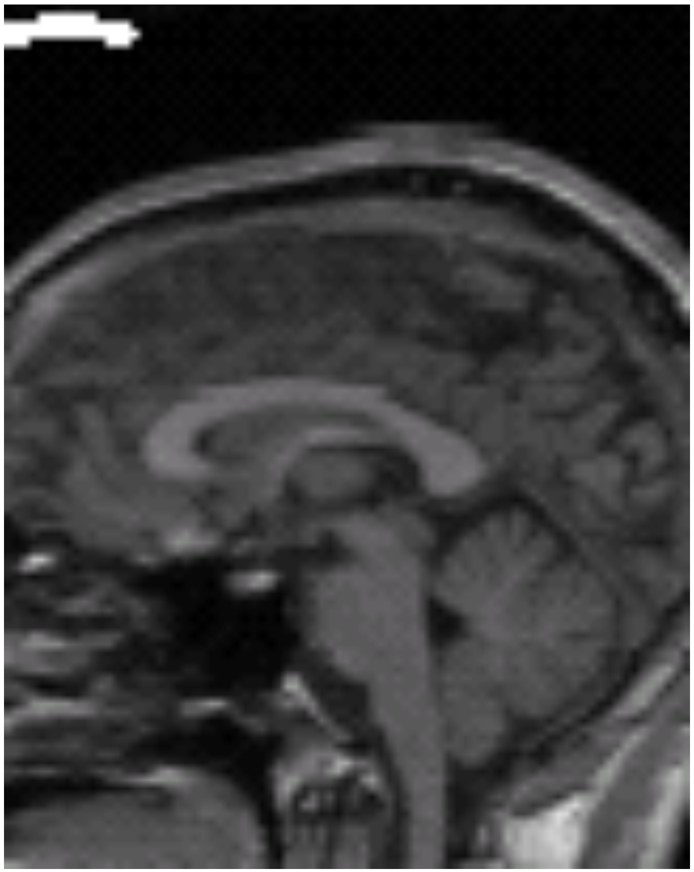


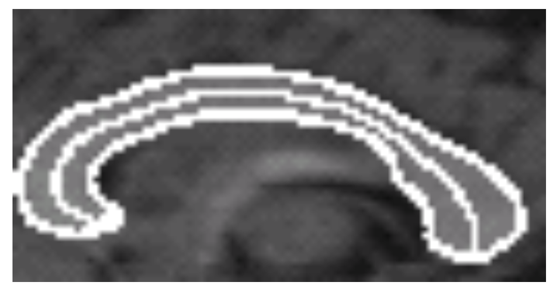
Boundary Detection

http://www.robots.ox.ac.uk/~vdg/dynamics.html



Boundary Detection

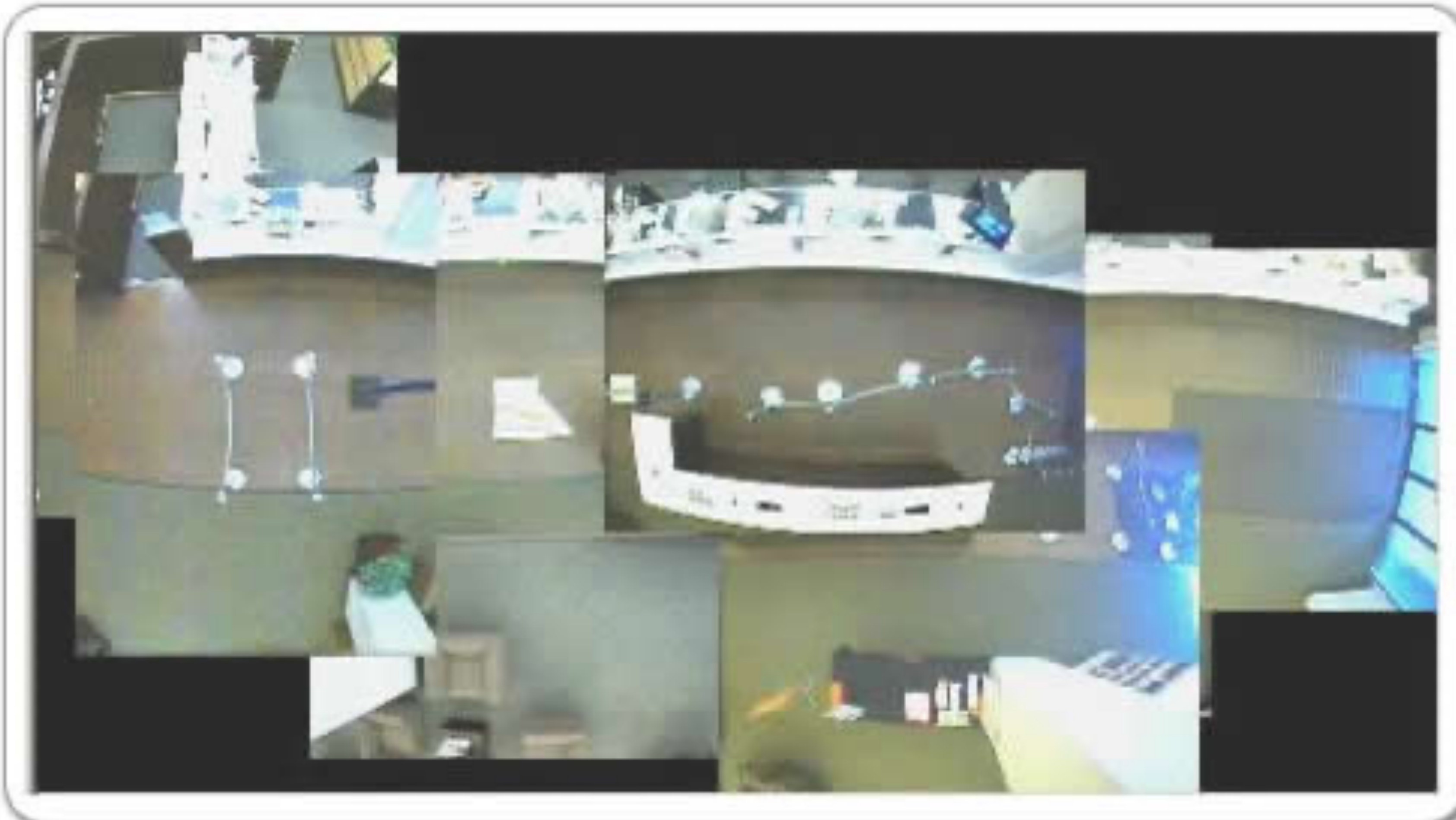




Finding the Corpus Callosum

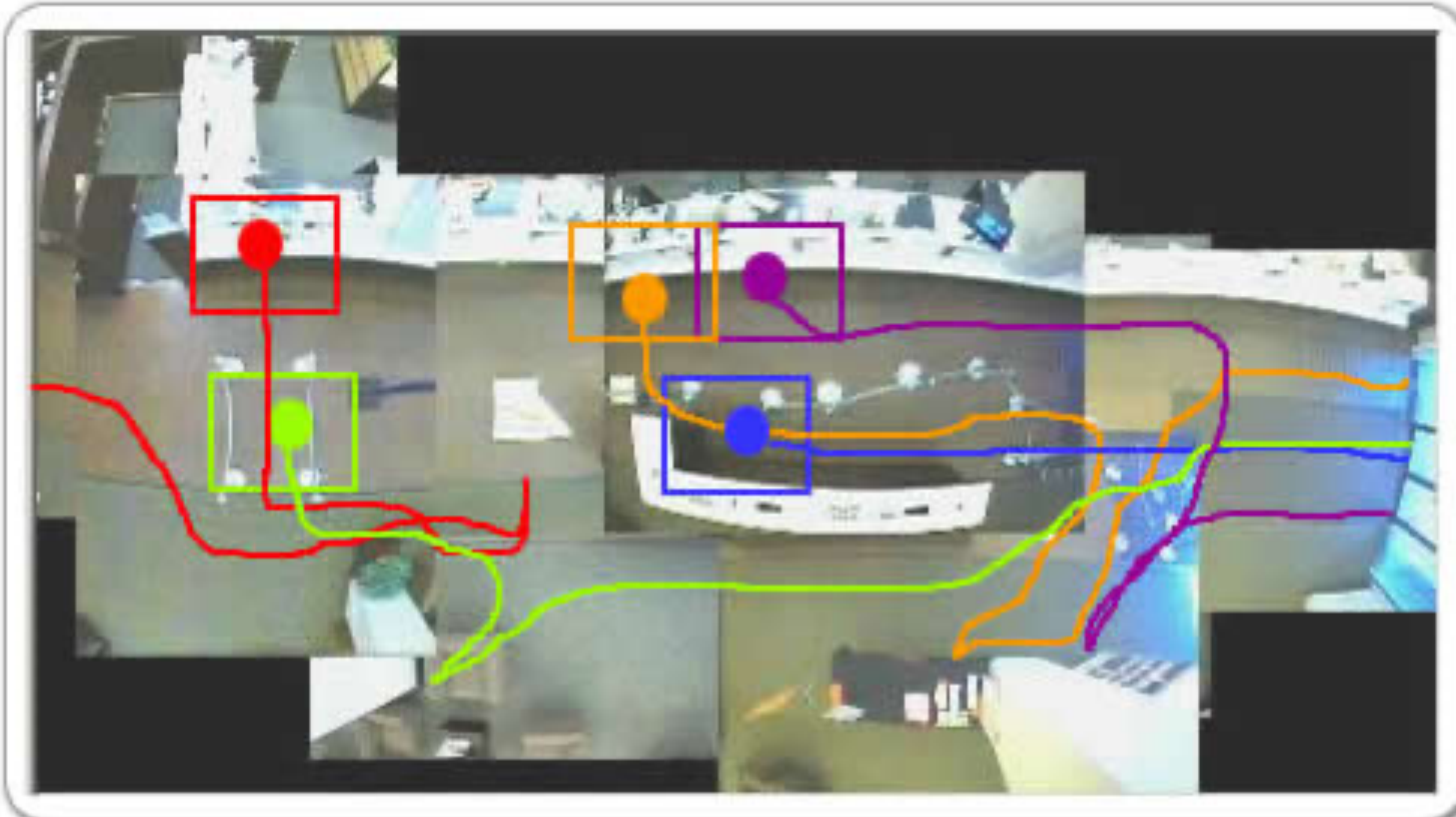
(G. Hamarneh, T. McInerney, D. Terzopoulos)

Tracking

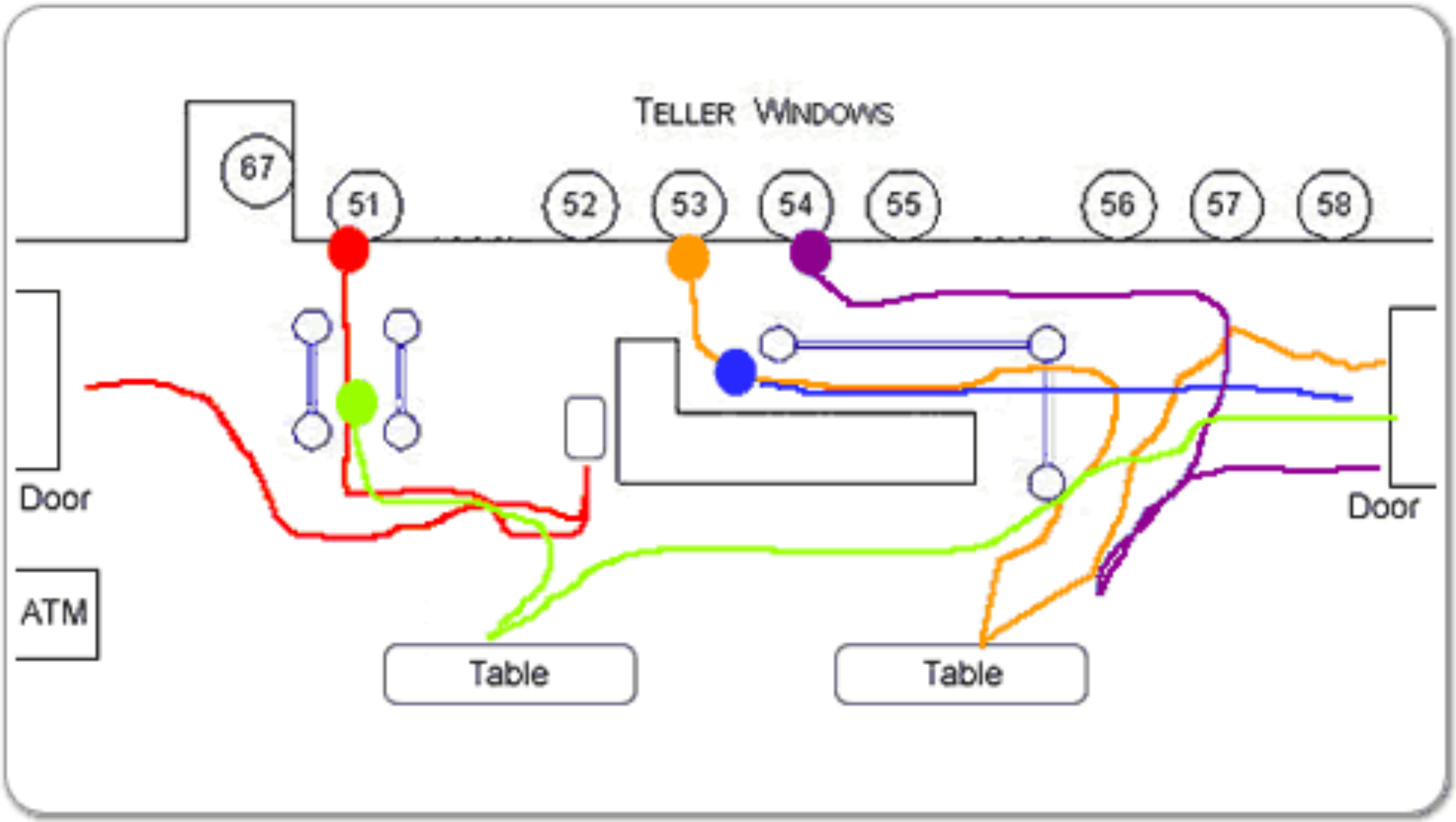


Tracking

Tracking

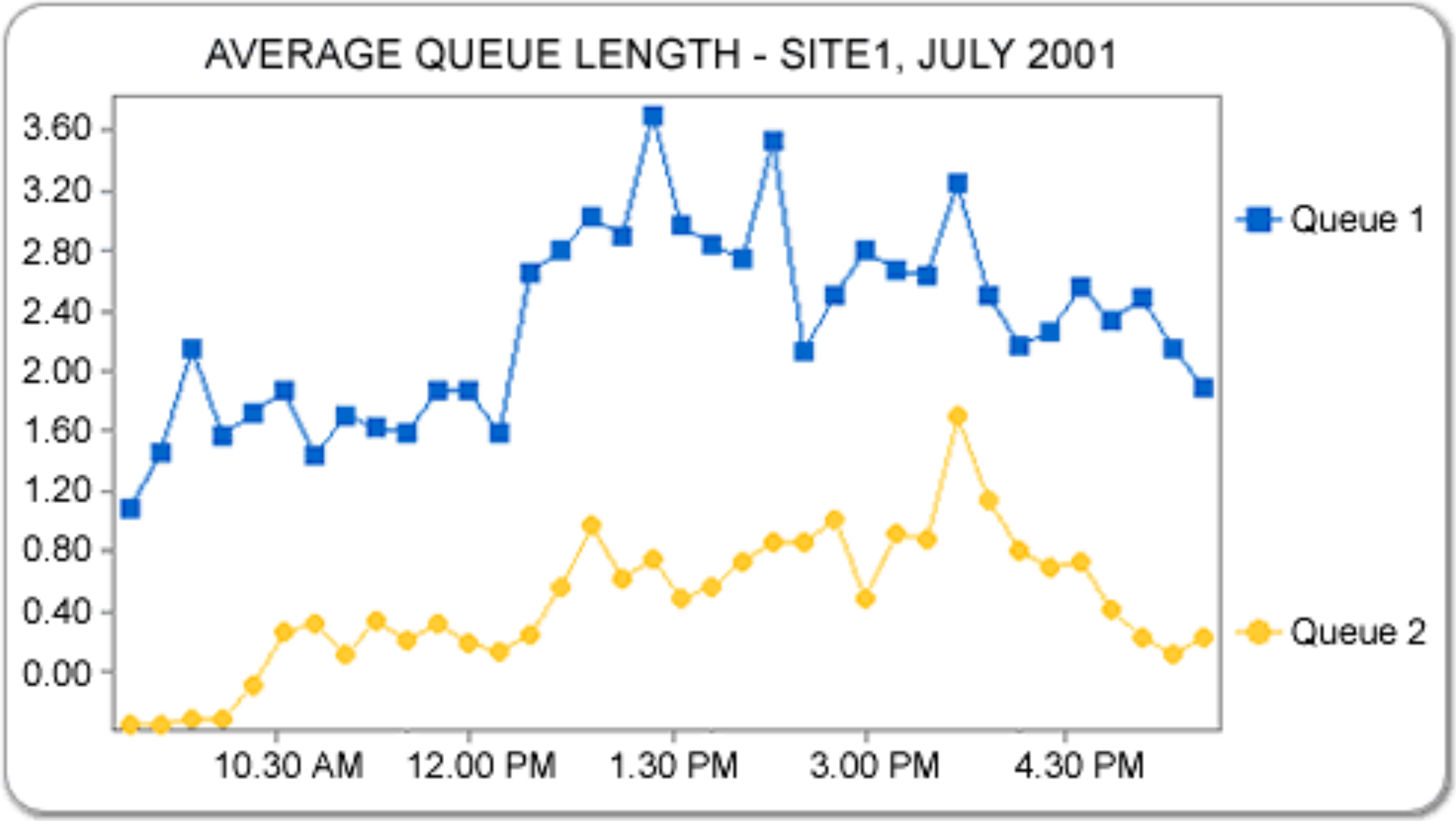


Tracking



TELLER WINDOWS
67
51
52
53
54
55
56
57
58
Door
Door
ATM
Table
Table

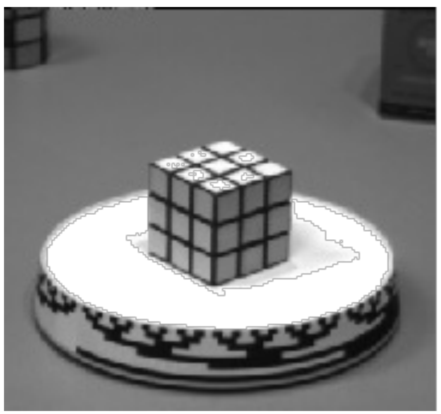
Tracking

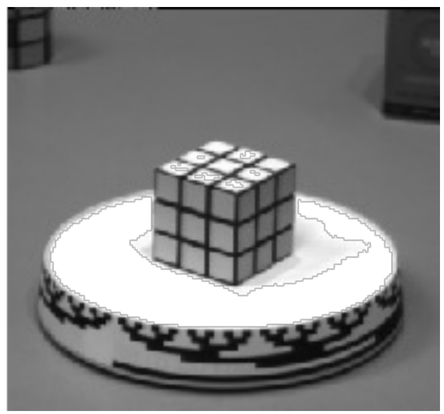


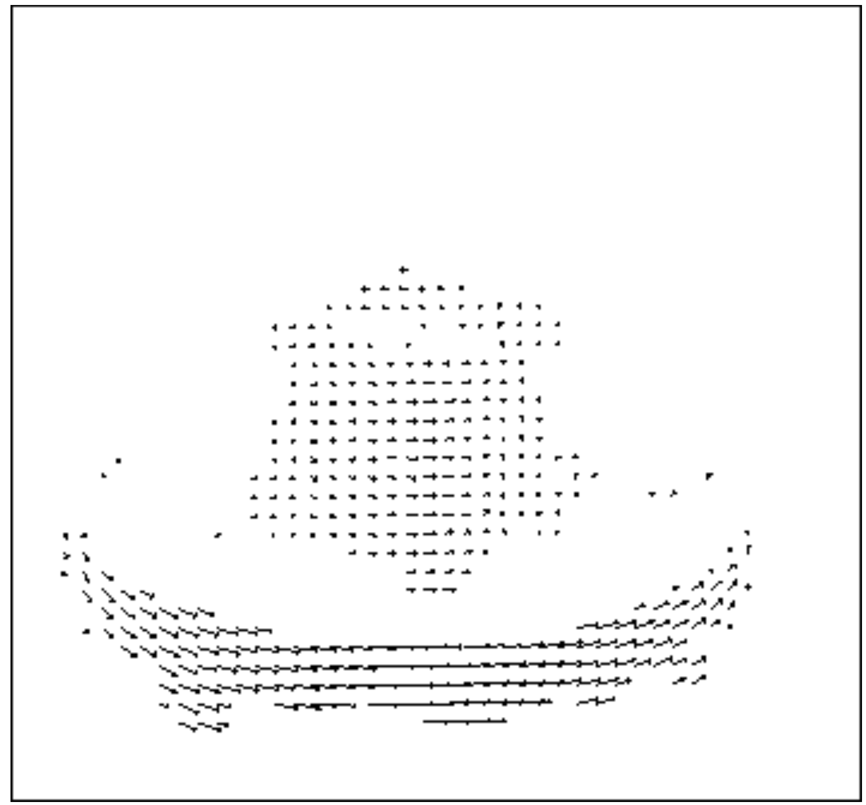
AVERAGE QUEUE LENGTH - SITE1, JULY 2001
3.60
3.20
2.80
2.40
2.00
1.60
1.20
0.80
0.40
0.00
10.30 AM
12.00 PM
1.30 PM
3.00 PM
4.30 PM
Queue 1
Queue 2

Optical flow

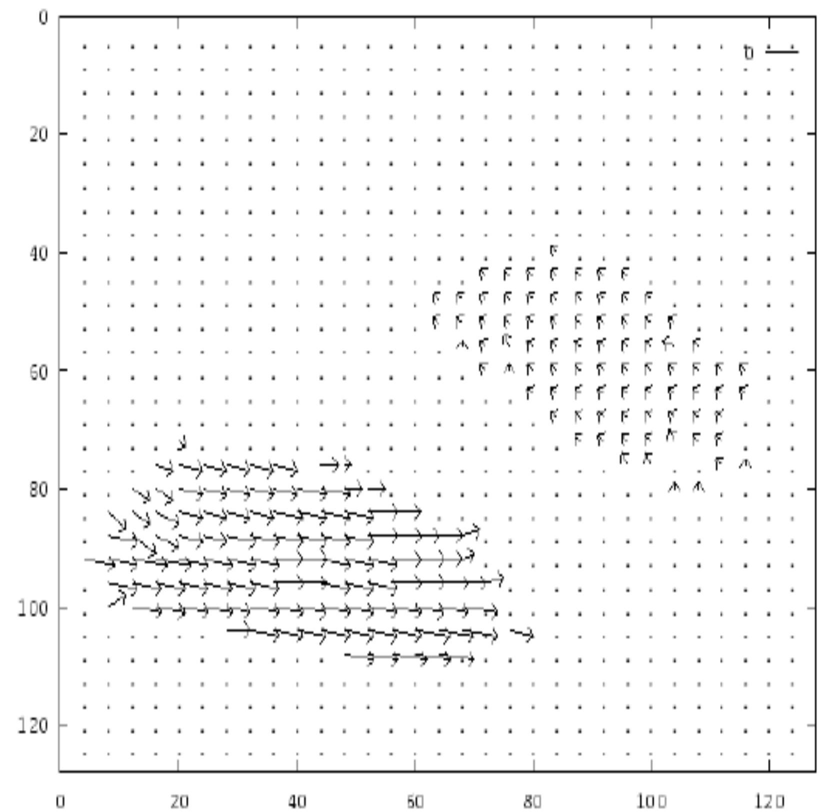
Measurement of motion at every pixel







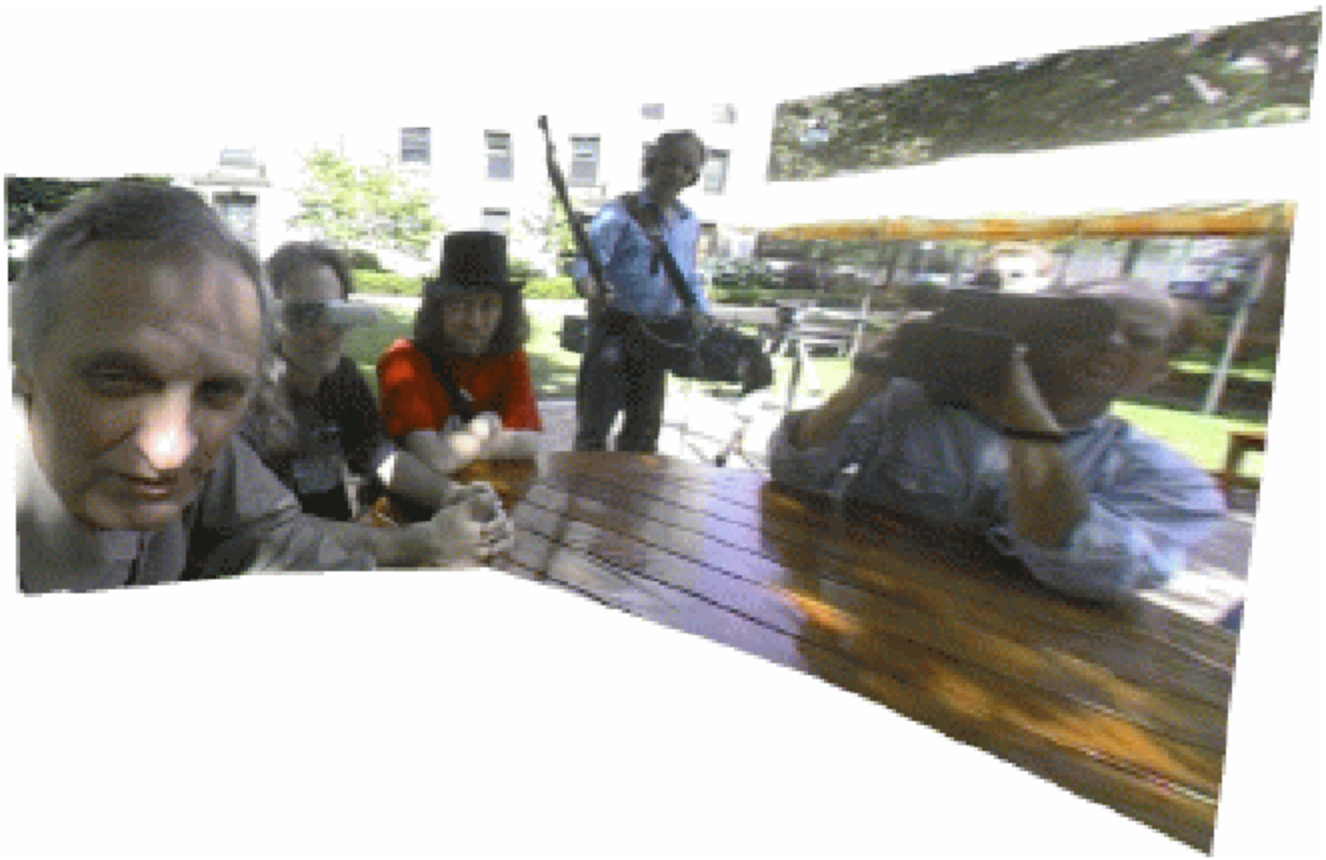
Optical flow



An image from Hamburg Taxi Sequence

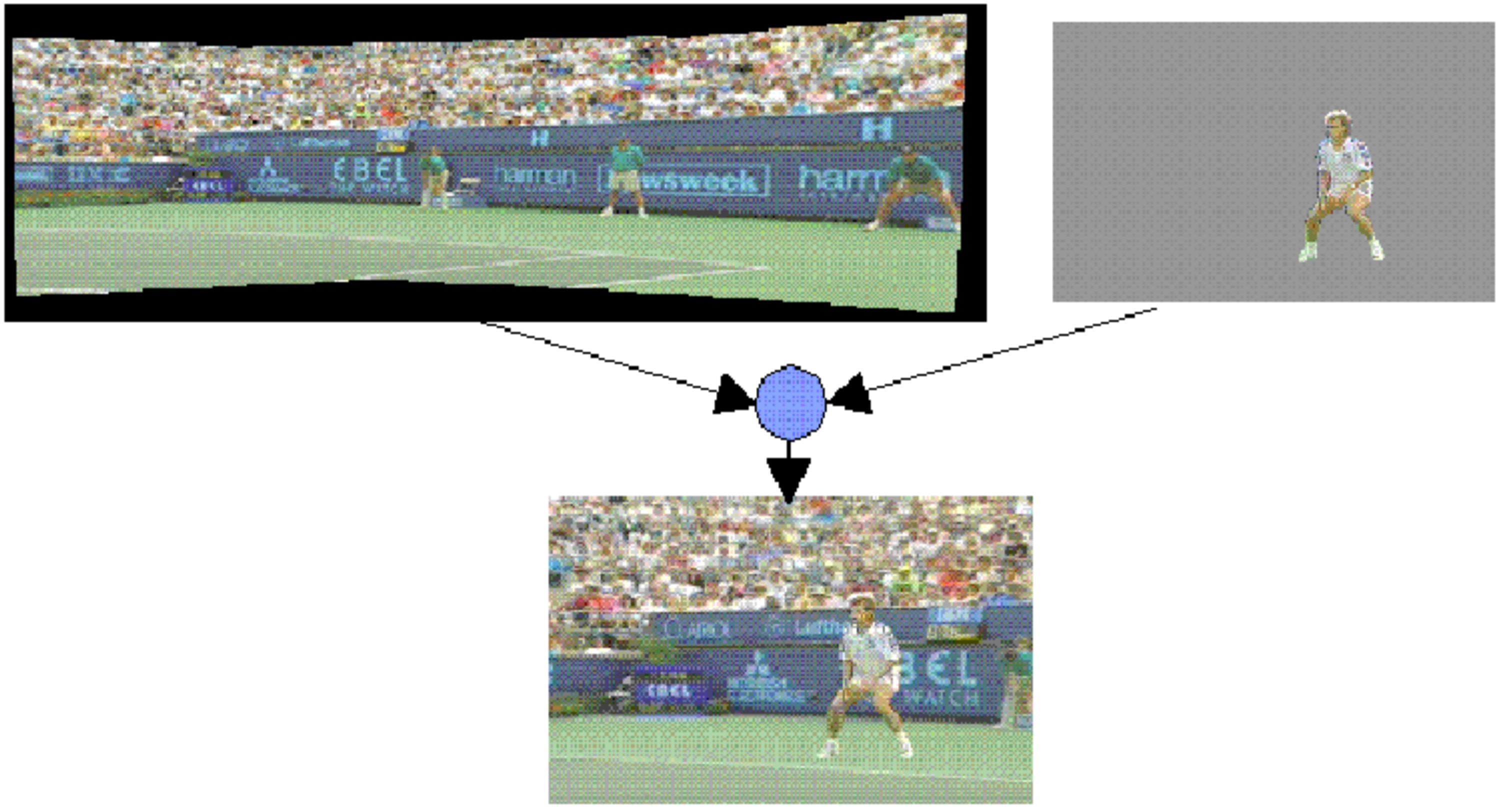
Video Mosaics

Video Mosaics



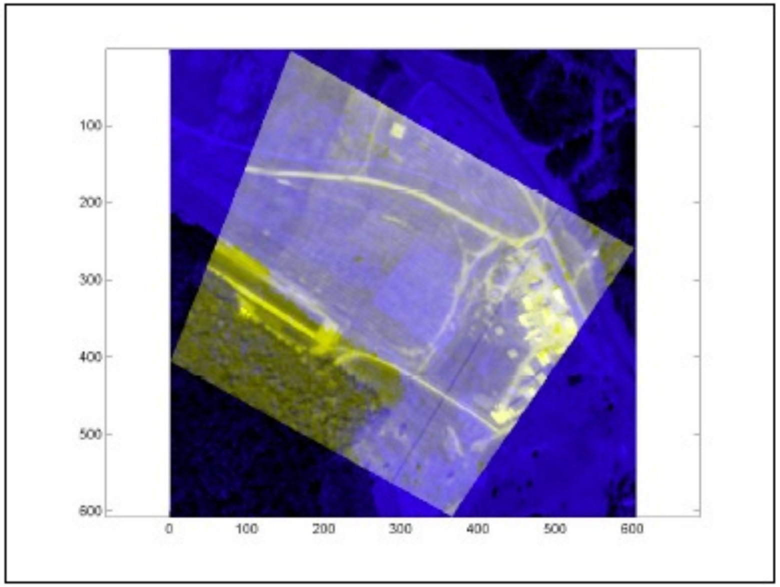
Video Mosaics

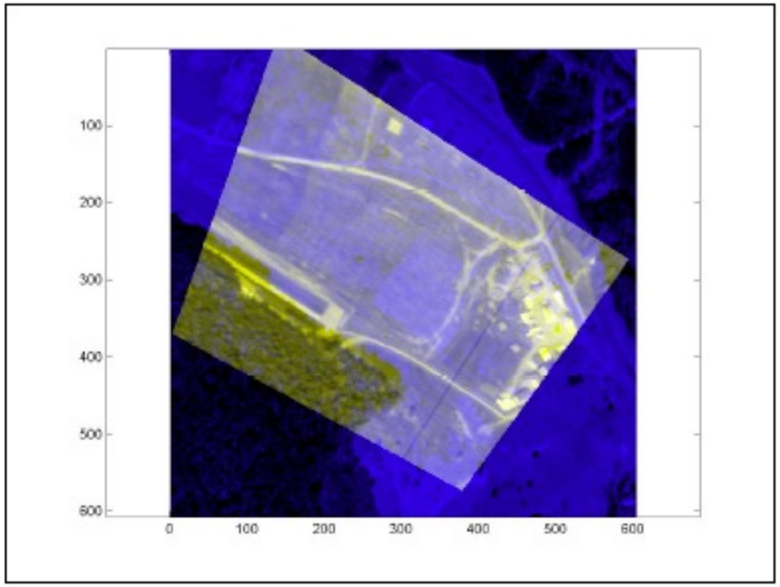
Video Compression



Geo Registration

Results superimposed with the reference image





Aims

- How to detect what changes have occurred
- How to determine what motions have induced those changes
- We will learn some other useful things on the way:
  - Notion of 4 and 8 neighbours
  - Notion of 4 and 8 connectedness
  - Using these for noise elimination
  - Moravec Interest Operator

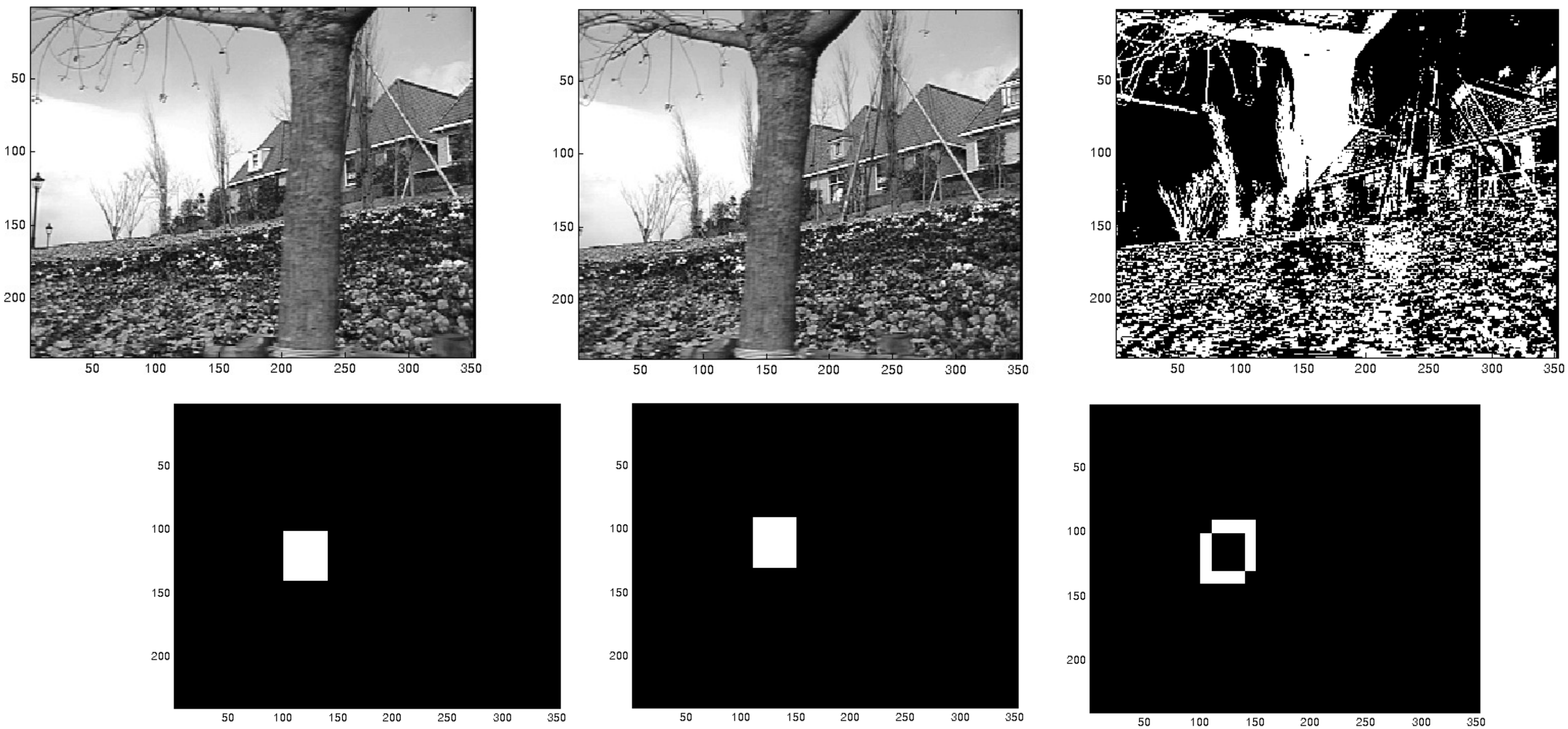
Event Code:

Dynamic Vision

- FOUR possibilities for dynamic nature of camera and world:
  - Stationary Camera, stationary Objects
  - Stationary Camera, moving Objects
  - Moving Camera, stationary Objects
  - Moving Camera, moving Objects

Detecting a change

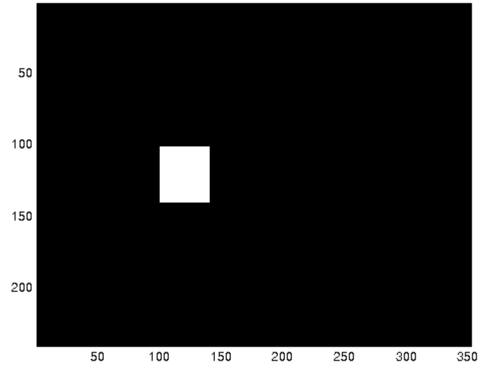
- Where has an image changed?

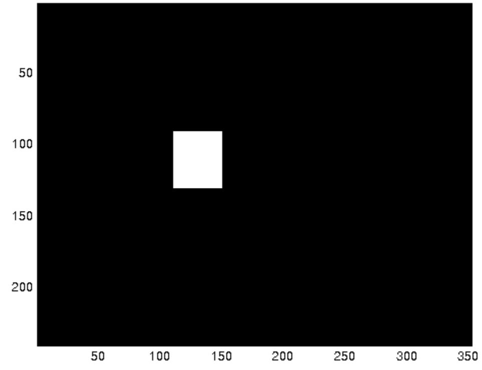


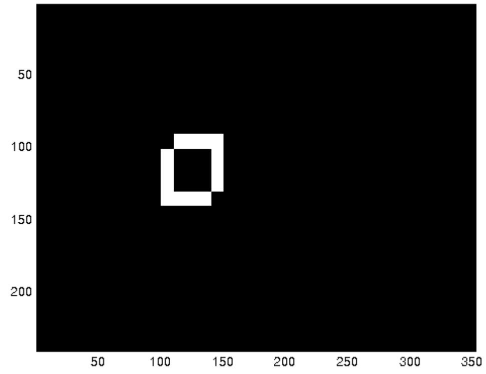
Detecting a change

- Where has an image changed?
- F(x,y,i) is the intensity of the image at time i, at point x,y
- Difference picture, DP

$$DP_{12}(x,y) = \begin{cases} 1 & if\ \left|F(x,y,1) - F(x,y,2)\right| > \tau \\ 0 & otherwise \end{cases}$$

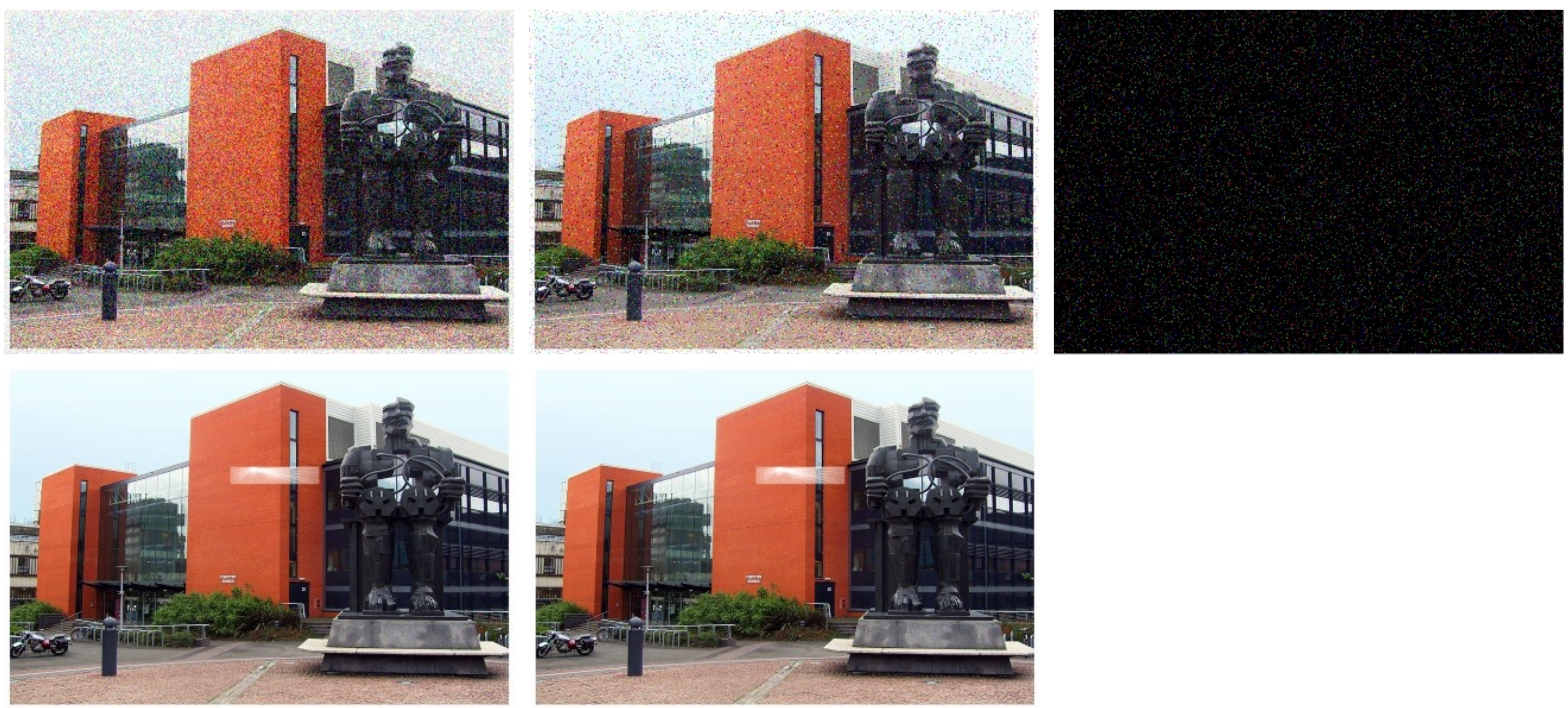






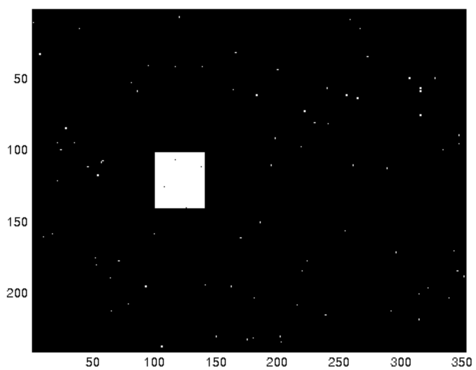
Detecting a change

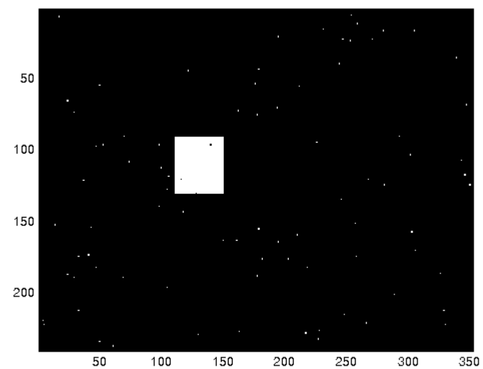
- If we have ‘random’ noise in image the difference picture will include all the noise points in both images

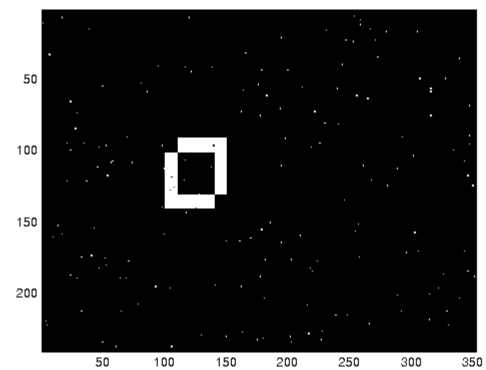


Detecting a change

- If we have ‘random’ noise in image the difference picture will include all the noise points in both images







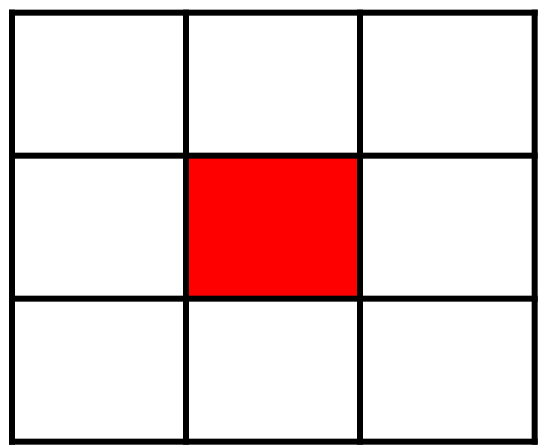
- Filter out all pixels that are not part of a larger structure---*how?*

Connectedness

- To filter out noise, we can use the idea of 8 or 4 connectedness:
  - Two pixels are 4-neighbours if they share a common boundary
  - Two pixels are 8-neighbours if they share at least a common corner
  - Two pixels are 8 connected if we can create a path of 8-neighbours from one to the other

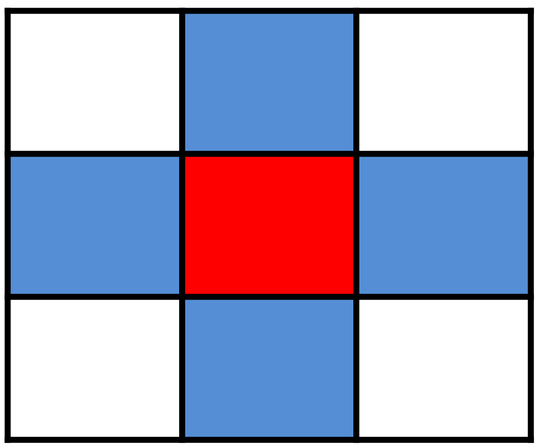
Connectedness

- To filter out noise, we can use the idea of 8 or 4 connectedness:
  - Two pixels are 4-neighbours if they share a common boundary



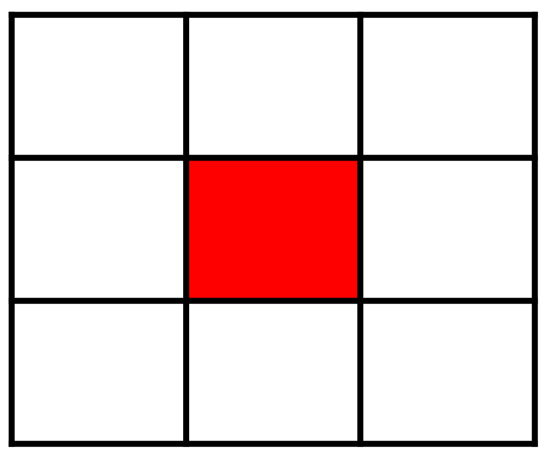
Connectedness

- To filter out noise, we can use the idea of 8 or 4 connectedness:
  - Two pixels are 4-neighbours if they share a common boundary



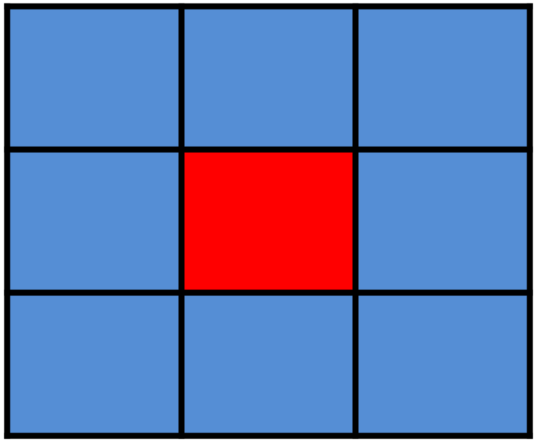
Connectedness

- To filter out noise, we can use the idea of 8 or 4 connectedness:
  - Two pixels are 8-neighbours if they share at least a common corner



Connectedness

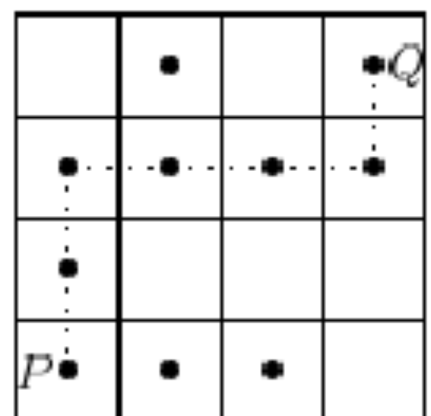
- To filter out noise, we can use the idea of 8 or 4 connectedness:
  - Two pixels are 8-neighbours if they share at least a common corner



Connectedness

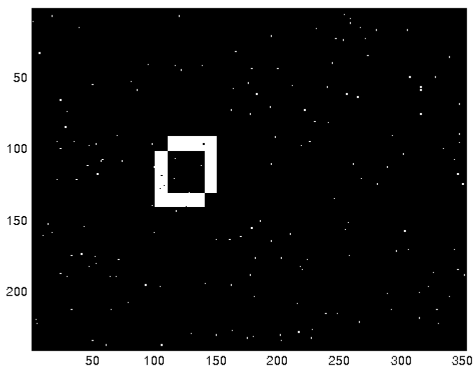
- To filter out noise, we can use the idea of 8 or 4 connectedness:
  - Suppose that P and Q are any two pixels (not necessarily adjacent), and suppose P and Q can be joined by a sequence of pixels as shown

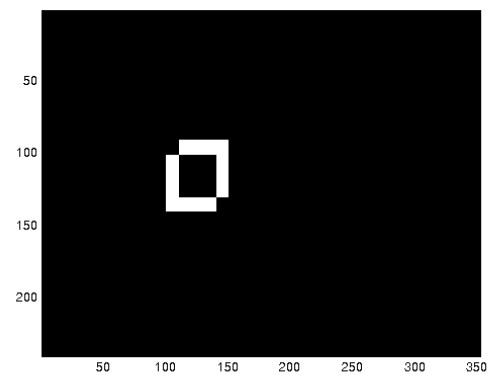
If the path contains 8-adjacent pixels, then P and Q are 8-connected.



Removing Noise

- Pixels not in a connected cluster of a certain size are removed from the difference image

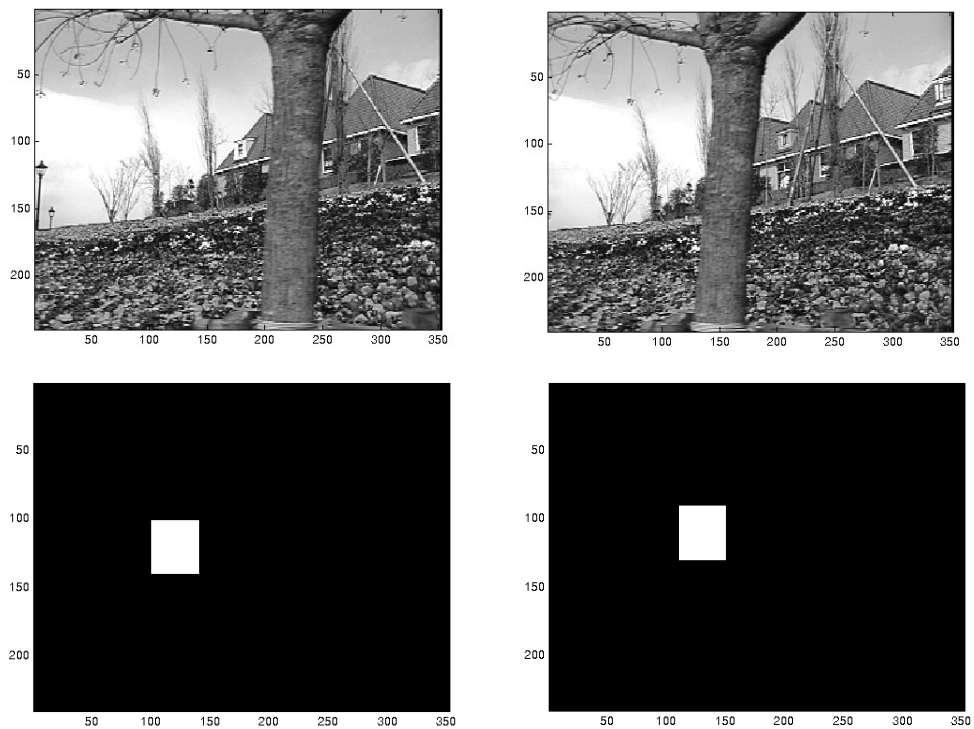




- *In MATLAB type 'help bwlabel'*

Determining motion

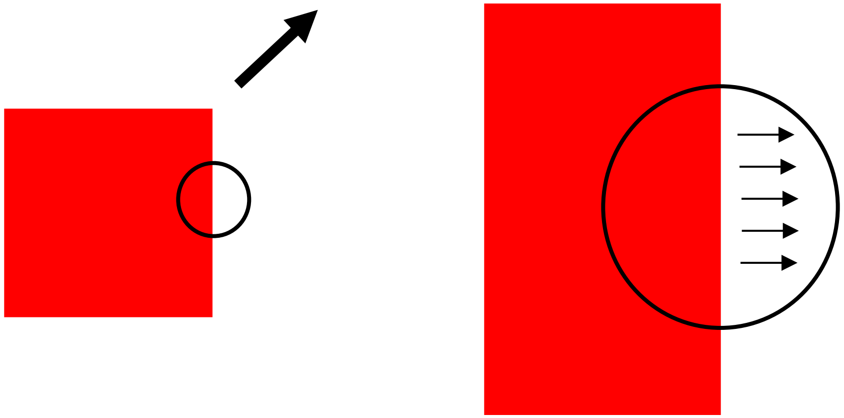
- We want to determine the motion of bodies in the image
- There is not enough information in the local intensity changes in an image to determine the motion



The aperture problem

- The grating appears to be moving down and to the right
- But it could be moving in many other directions
  - such as only down, or only to the right.

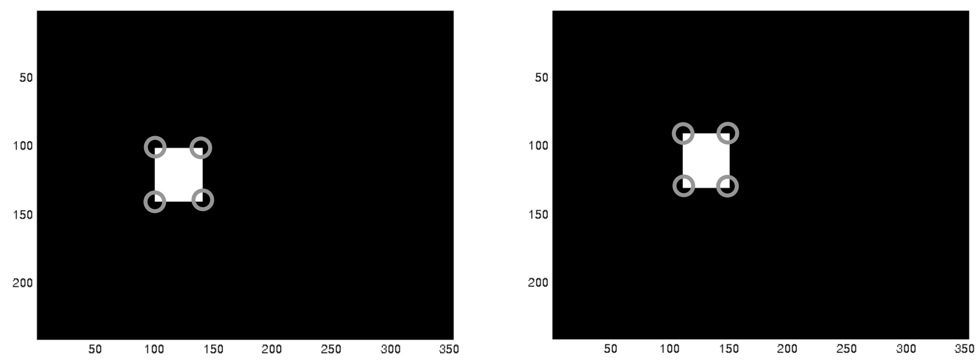
The aperture problem



- Locally we only see the horizontal component of the motion
- What ways round this are there?

Motion correspondence

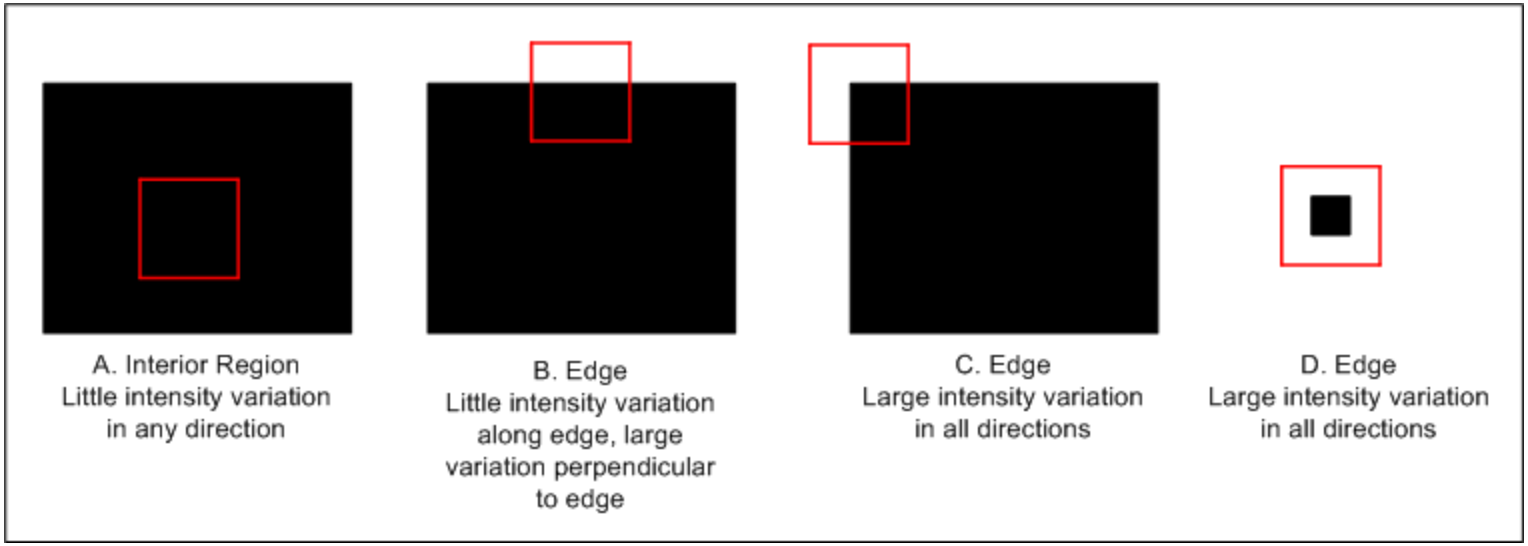
- One way is to pick a bunch of interesting points in each image
- And then match them (hoping they can be matched)



- e.g. pick corner points using a corner detector
- Or use a Moravec operator

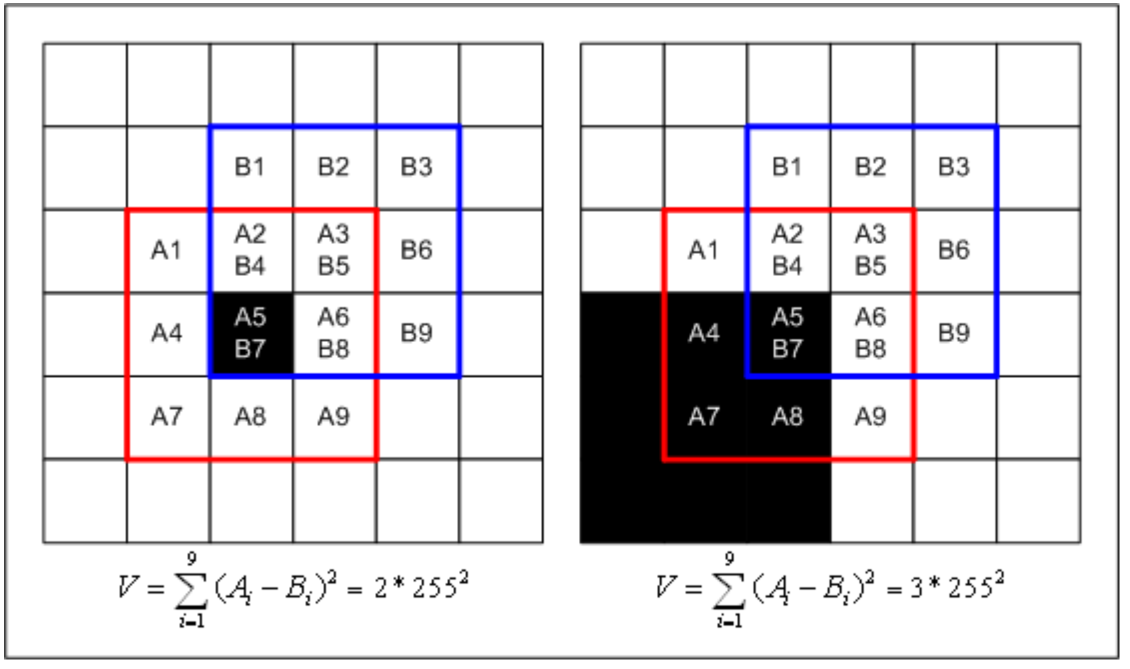
Moravec Operator

- The Moravec operator is considered a corner detector
  - it defines interest points as points where there is a large intensity variation in every direction.
  - This is the case at corners.
  - It may be sensitive to detecting isolated pixels as corners.



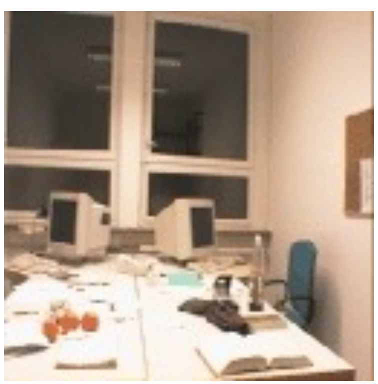
Moravec Operator

- measure the intensity variation by placing a small square window (typically, 3x3, 5x5, or 7x7 pixels) centered at P
  - then shifting this window by one pixel in each of the eight principle directions (horizontally, vertically, and four diagonals).



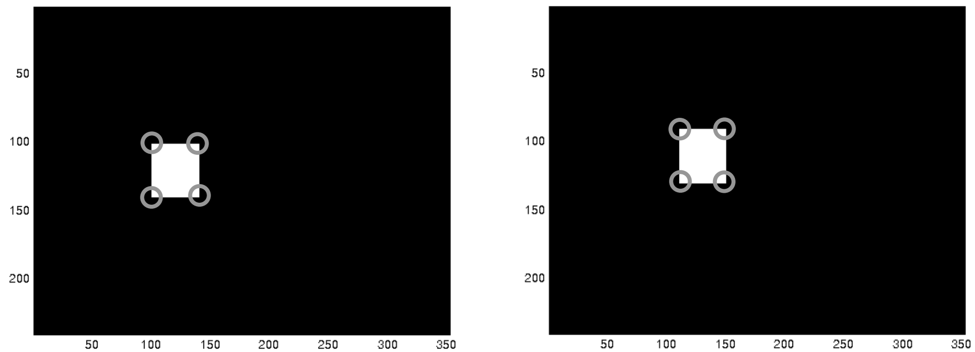
The Moravec Interest Operator

- We now apply the Moravec operator to every point (i,j) in the image
- This produces a new interest image
- We keep only the local maxima
  - (i.e. points that are locally most interesting)

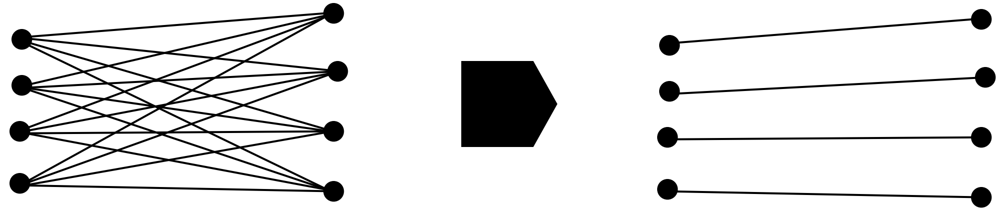


Motion Correspondence

- Now we have our points of interest



- We can try to match points in one image with those in another



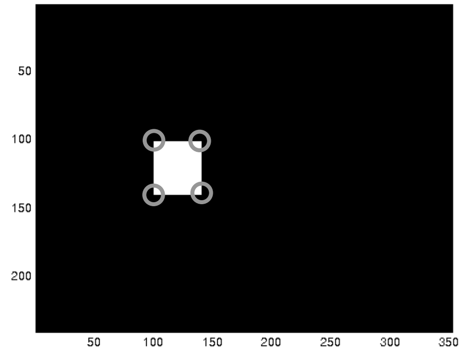
- By doing this we are able to estimate the motion

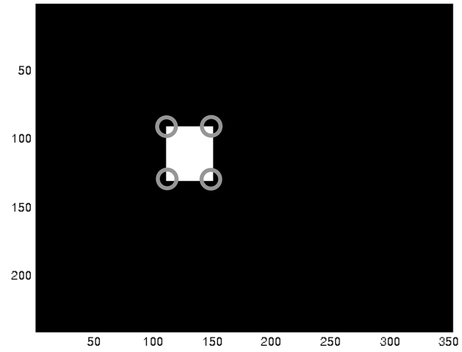
Motion Correspondence

- Motion correspondence (matching) is guided by three principles
  1. Discreteness: a measure of the distinctiveness of individual points
  2. Similarity: a measure of how closely two points resemble one another
  3. Consistency: a measure of how well a match conforms with nearby matches

Motion correspondence

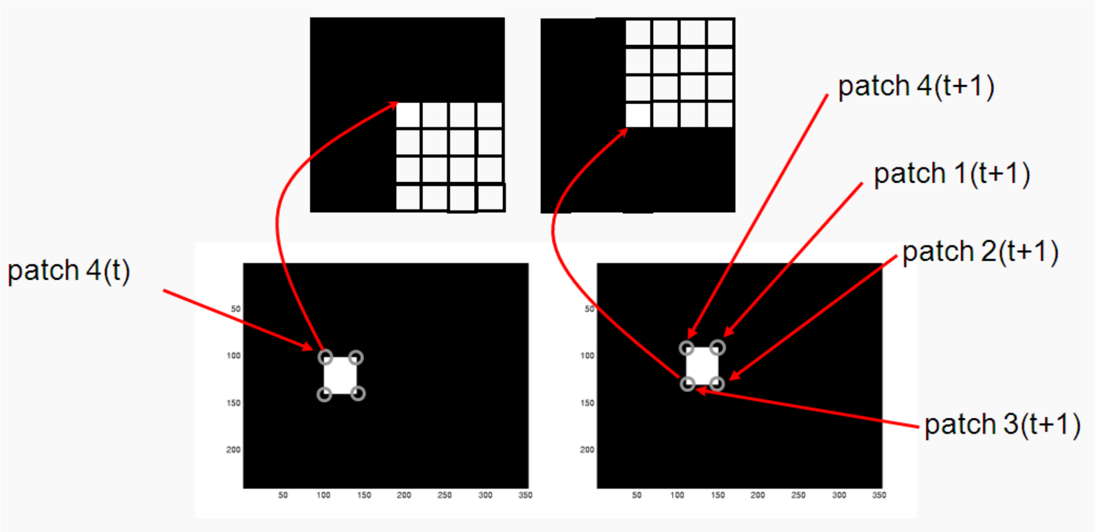
- A sketch of the algorithm
  - Get the interest points in each image
  - Pair each feature point in the first image with every feature point in the other image within some distance
  - Calculate the degree of similarity between the images for each possible match
  - Use this to calculate the likelihood of each match
  - Now revise the match likelihood for each point using the nearby matches





Calculating the degree of similarity

- To calculate the degree of similarity between two patches take the sum of the differences in the pixel values



i(t) is the patch in the first image

j(t+1) is a patch in the second image

s_{ij} is the similarity between i and j

w_{ij} is a weight

$$w_{ij} = \frac{1}{1 + \alpha\, s_{ij}}$$

Calculating the likelihood of a match

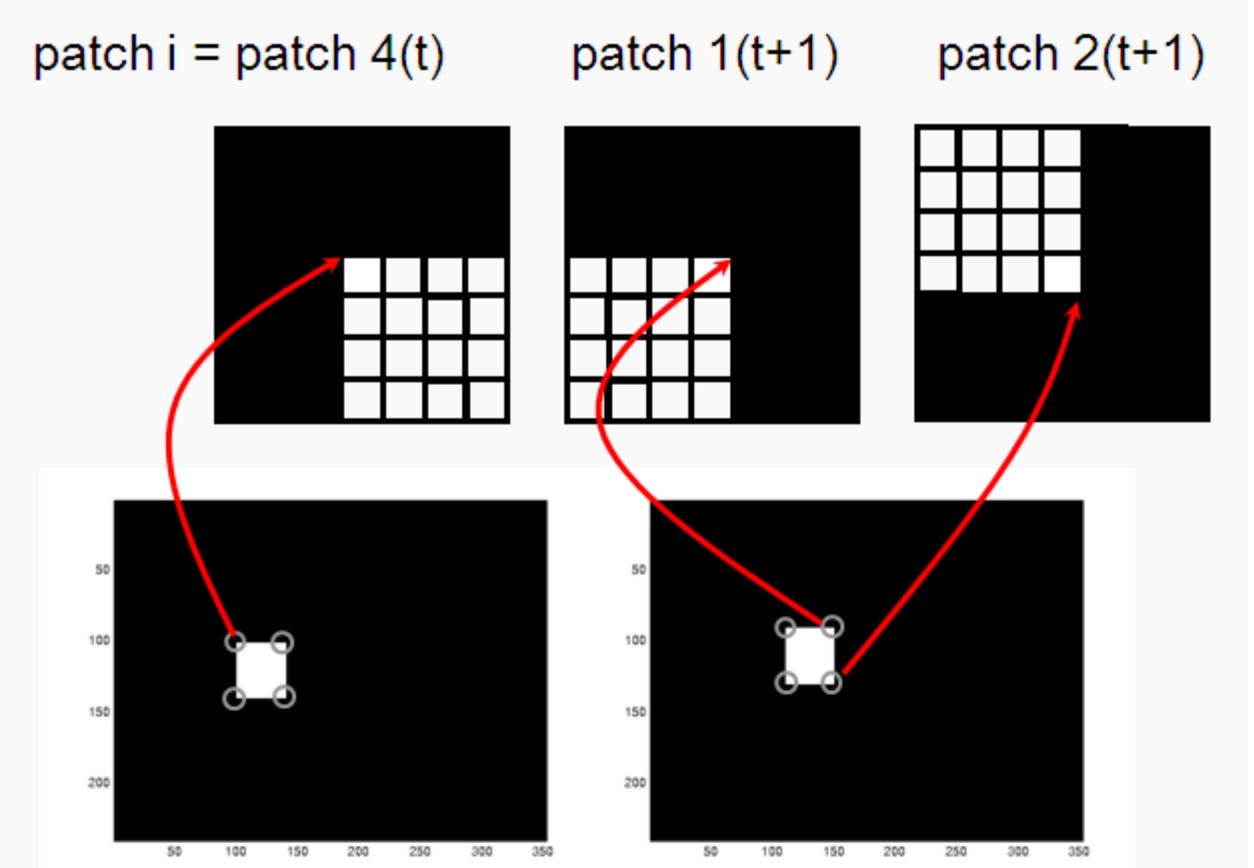
- Now we calculate the weights for all the possible matches for patch i(t)

$$S_{i1} = 24$$

$$S_{i2} = 30$$

$$w_{i1} = \frac{1}{1+\alpha 24}$$

$$w_{i2} = \frac{1}{1+\alpha 30}$$



Once we have the weights we will just normalise them to form a probability distribution over the matches

Calculating the likelihood of a match

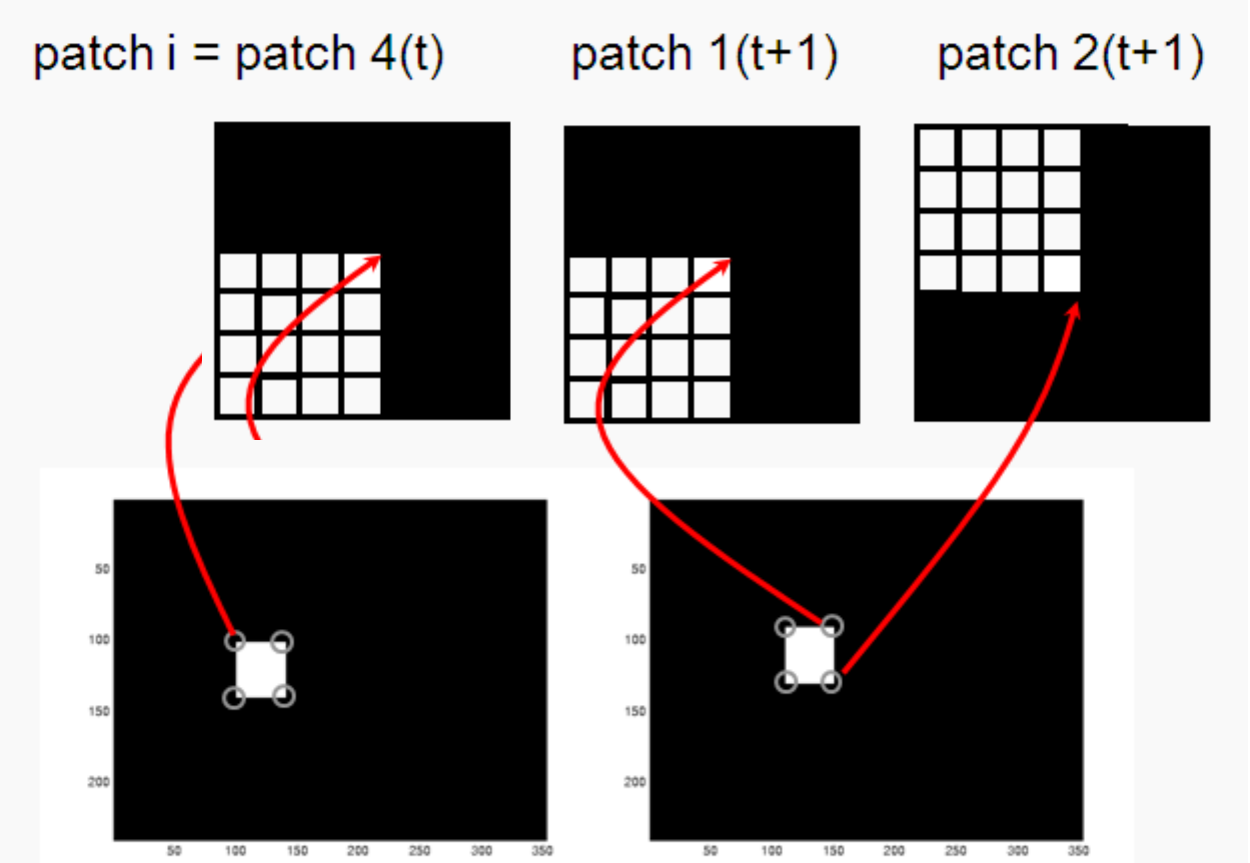
- Now we calculate the weights for all the possible matches for patch i(t)

$$S_{i1} = 24$$

$$S_{i2} = 30$$

$$w_{i1} = \frac{1}{1+\alpha 24}$$

$$w_{i2} = \frac{1}{1+\alpha 30}$$



Once we have the weights we will just normalise them to form a probability distribution over the matches

Calculating the likelihood of a match

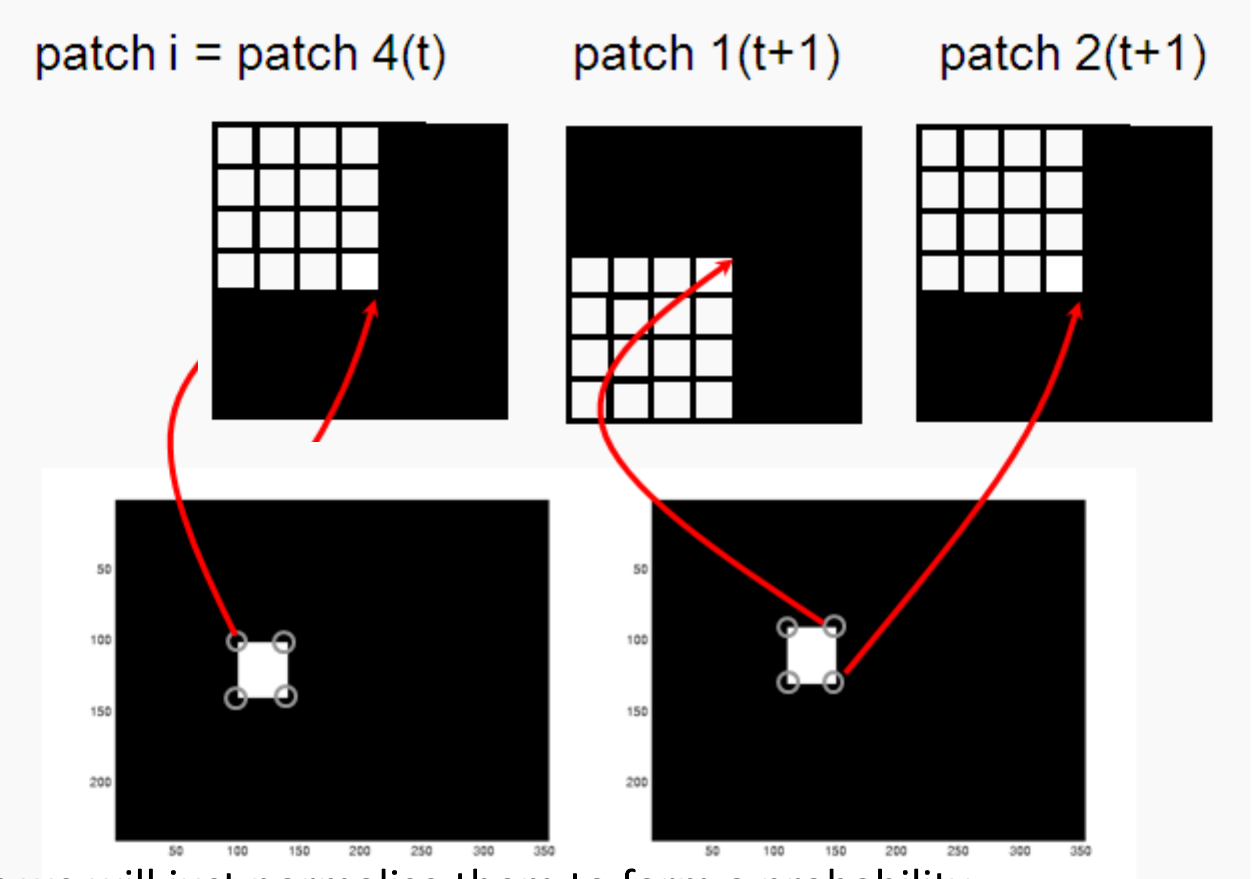
- Now we calculate the weights for all the possible matches for patch i(t)

$$S_{i1} = 24$$

$$S_{i2} = 30$$

$$w_{i1} = \frac{1}{1+\alpha 24}$$

$$w_{i2} = \frac{1}{1+\alpha 30}$$



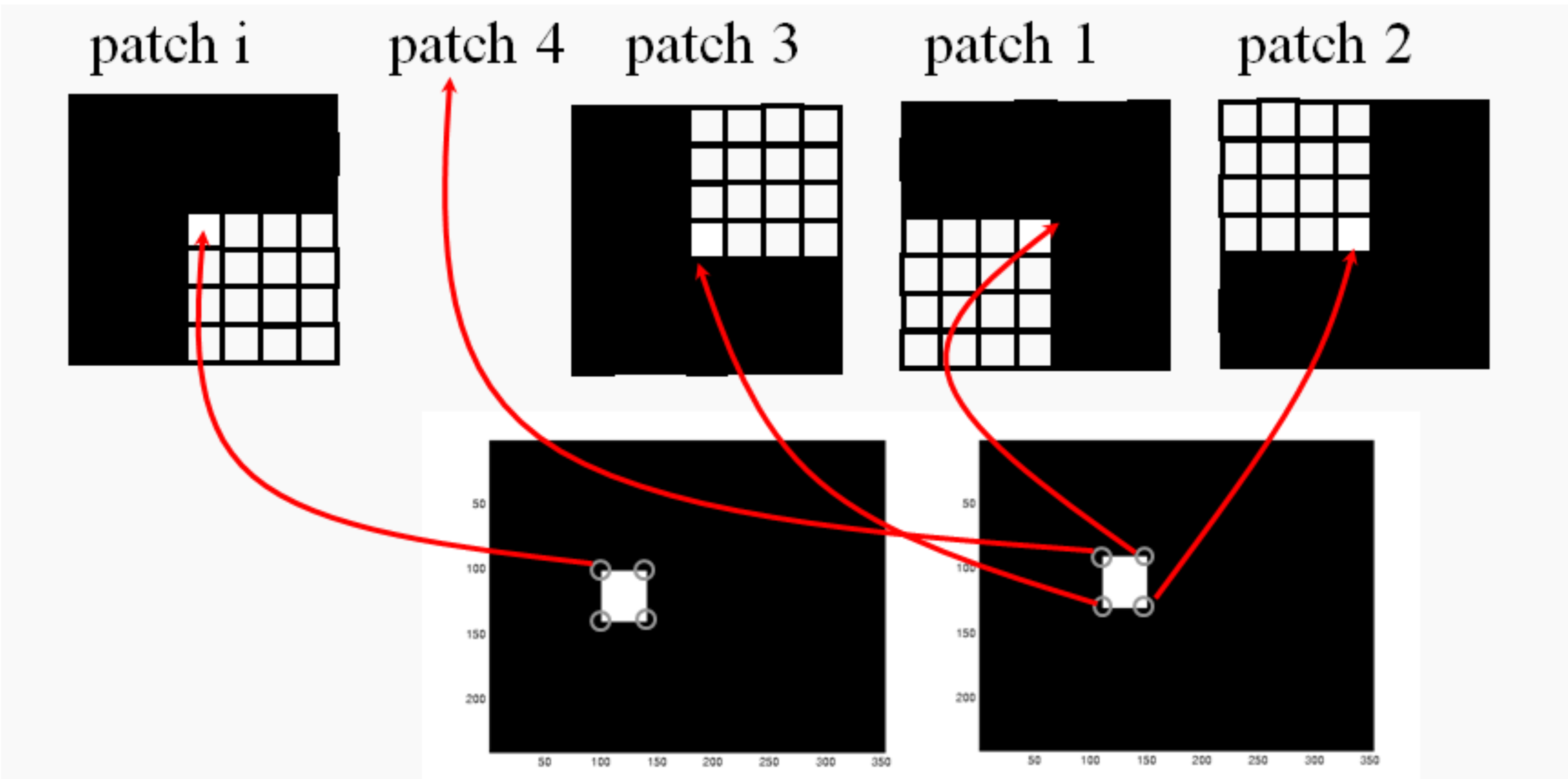
Once we have the weights we will just normalise them to form a probability distribution over the matches

The likelihood of a match

- Assume α = 1

$w_{i4} = 1/1$ $w_{i3} = 1/25$ $w_{i1} = 1/25$ $w_{i2} = 1/31$

$p_{i4} = 0.89$ $p_{i3} = 0.036$ $p_{i1} = 0.036$ $p_{i2} = 0.029$

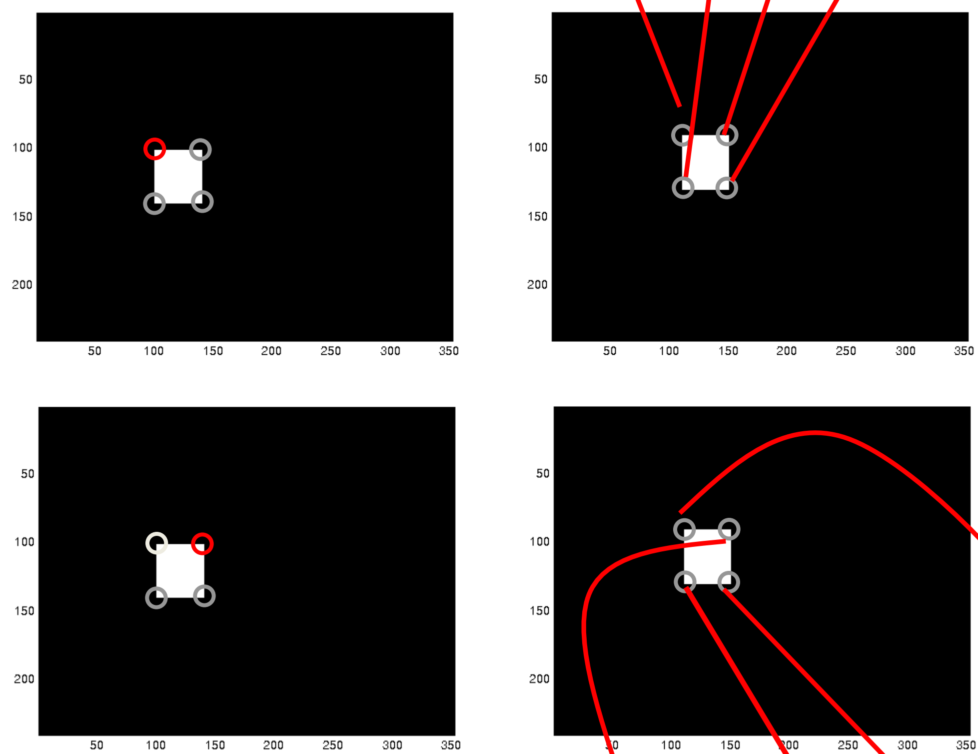


Adjusting probabilities

- So now we have a probability for each match

match probabilities for patch 4(t)= <.89 .04 .04 .03>

We now revise the probabilities for patch 4(t) using all the probabilities for its neighbours



match probabilities for patch 1(t) = <.89 .04 .03 .04>

Summary

- Change detection
- Making change detection more robust to noise
- Determining motion of objects
- Aperture problem
- Motion Correspondence