Corner Detection

COMP 4900D

Winter 2006

Motivation: Features for Recognition

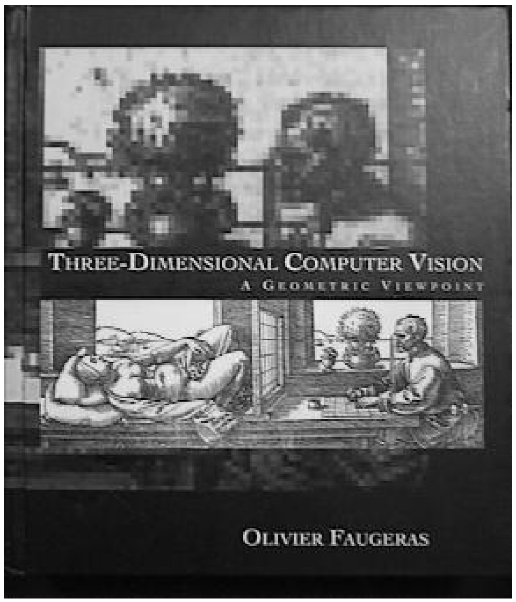


Image search: find the book in an image.

Motivation: Build a Panorama

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

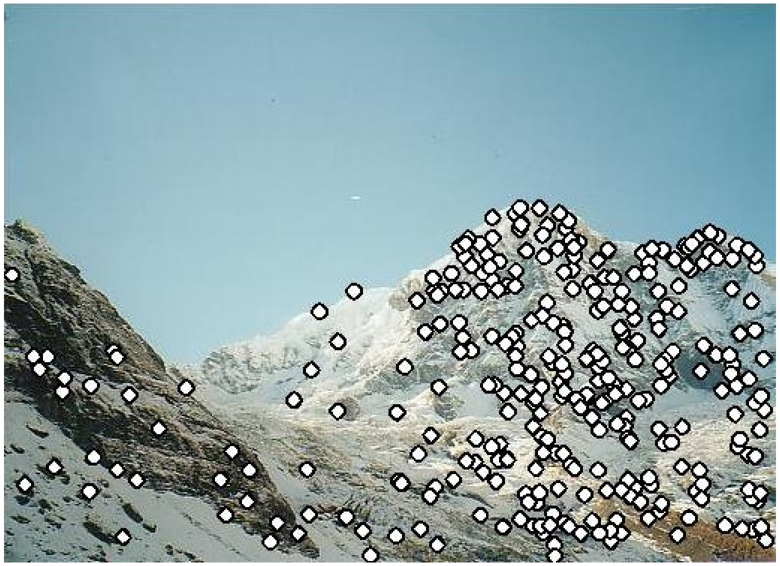
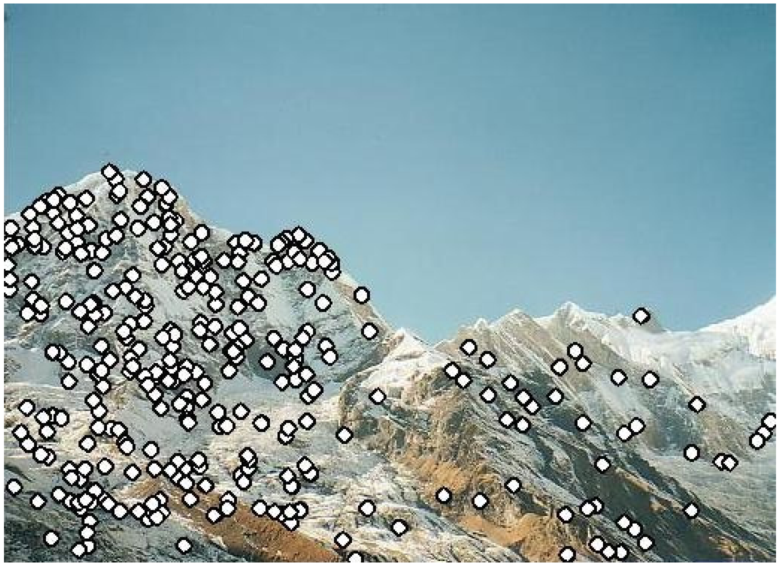
How do we build panorama?

We need to match (align) images

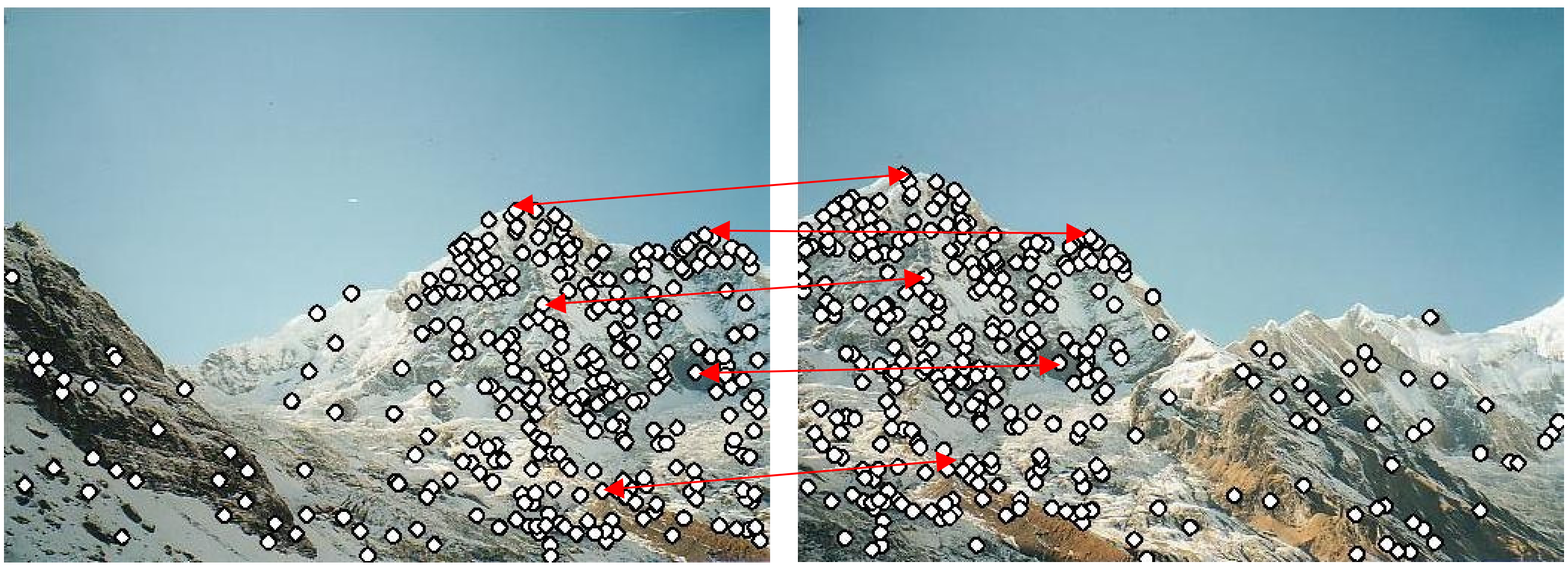
Matching with Features

- Detect feature points in both images

Matching with Features

- Detect feature points in both images
- Find corresponding pairs



Matching with Features

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images

More motivation…

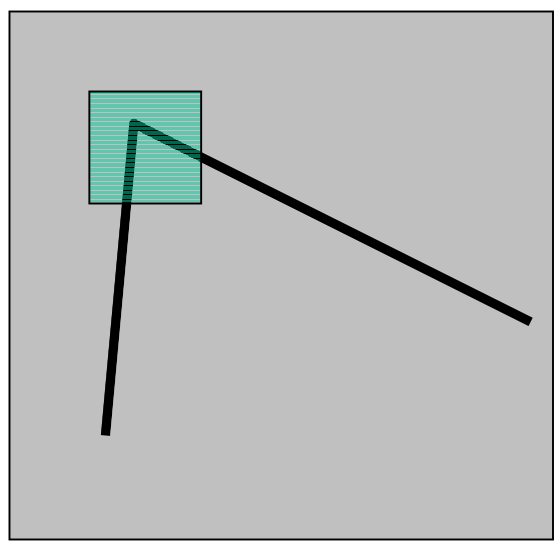
Feature points are used also for:

- Image alignment (homography, fundamental matrix)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- … other

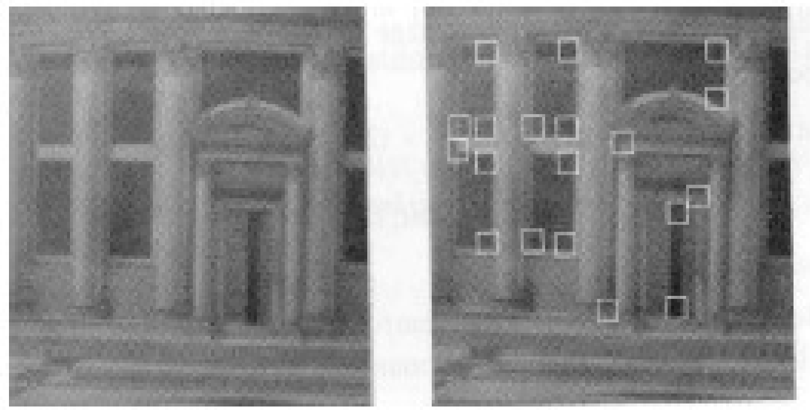
Corner Feature

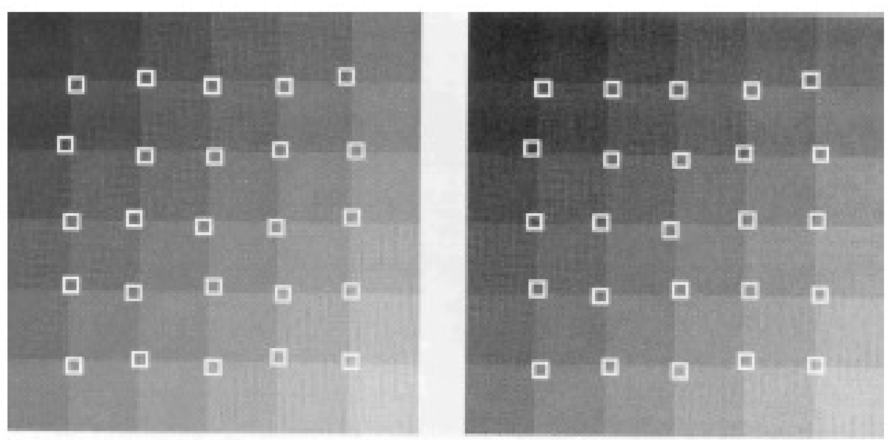
Corners are image locations that have large intensity changes in more than one directions.

Shifting a window in *any direction* should give *a large change* in intensity

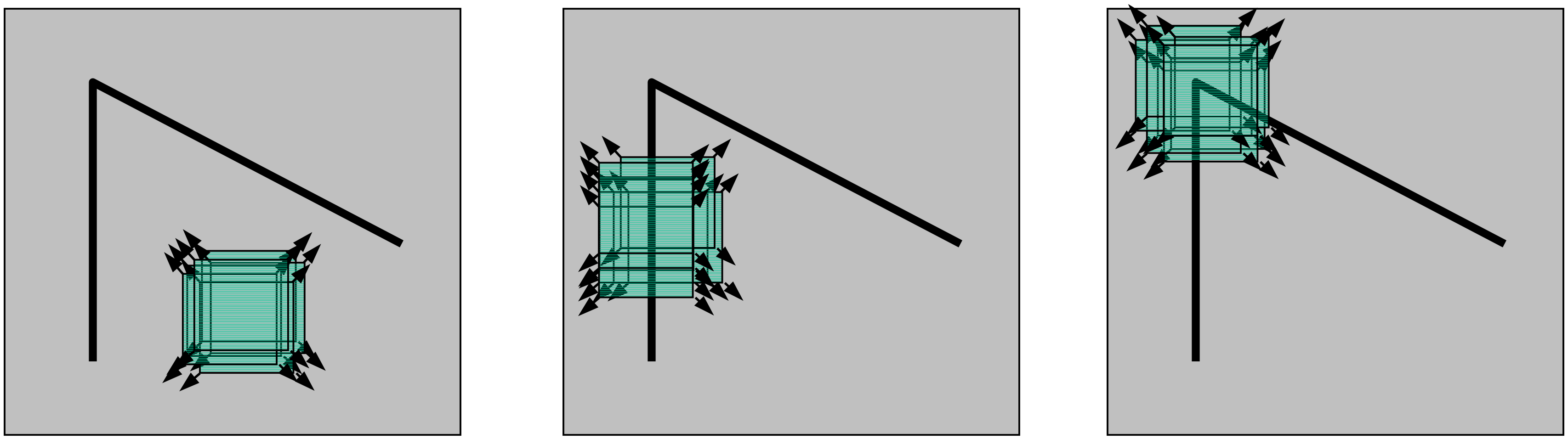


Examples of Corner Features





Harris Detector: Basic Idea



"flat" region:
no change in
all directions

"edge":
no change along
the edge direction

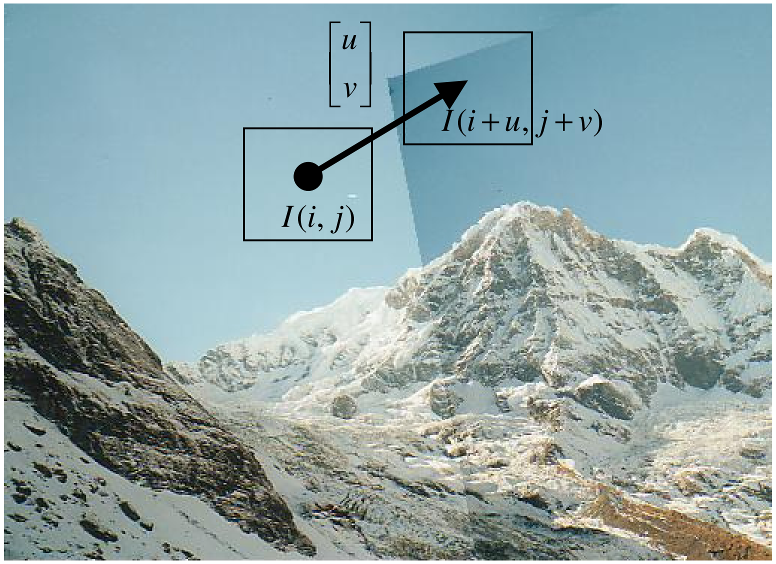
"corner":
significant change
in all directions

C.Harris, M.Stephens. "A Combined Corner and Edge Detector". 1988

Change of Intensity

The intensity change along some direction can be quantified by sum-of-squared-difference (SSD).

$$D(u,v) = \sum_{i,j} \left(I(i+u, j+v) - I(i,j) \right)^2$$



Change Approximation

If u and v are small, by Taylor theorem:

$$I(i+u, j+v) \approx I(i, j) + I_x u + I_y v$$

where $I_x = \dfrac{\partial I}{\partial x}$ *and* $I_y = \dfrac{\partial I}{\partial y}$

therefore

$$\begin{aligned}
\big(I(i+u, j+v) - I(i, j)\big)^2 &= \big(I(i, j) + I_x u + I_y v - I(i, j)\big)^2 \\
&= \big(I_x u + I_y v\big)^2 \\
&= I_x^2 u^2 + 2 I_x I_y uv + I_y^2 v^2 \\
&= \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}
\end{aligned}$$

Gradient Variation Matrix

$$D(u,v) = \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

This is a function of ellipse.

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

Matrix C characterizes how intensity changes in a certain direction.

Eigenvalue Analysis – simple case

First, consider case where:

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

This means dominant gradient directions align with x or y axis

If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

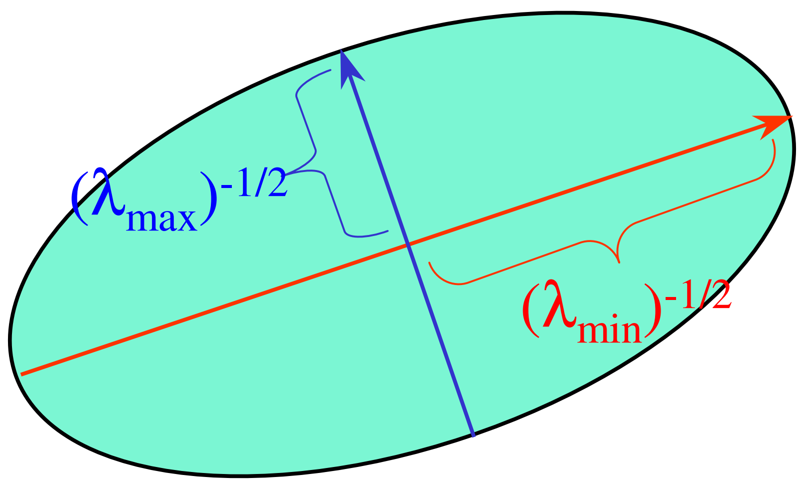
Slide credit: David Jacobs

General Case

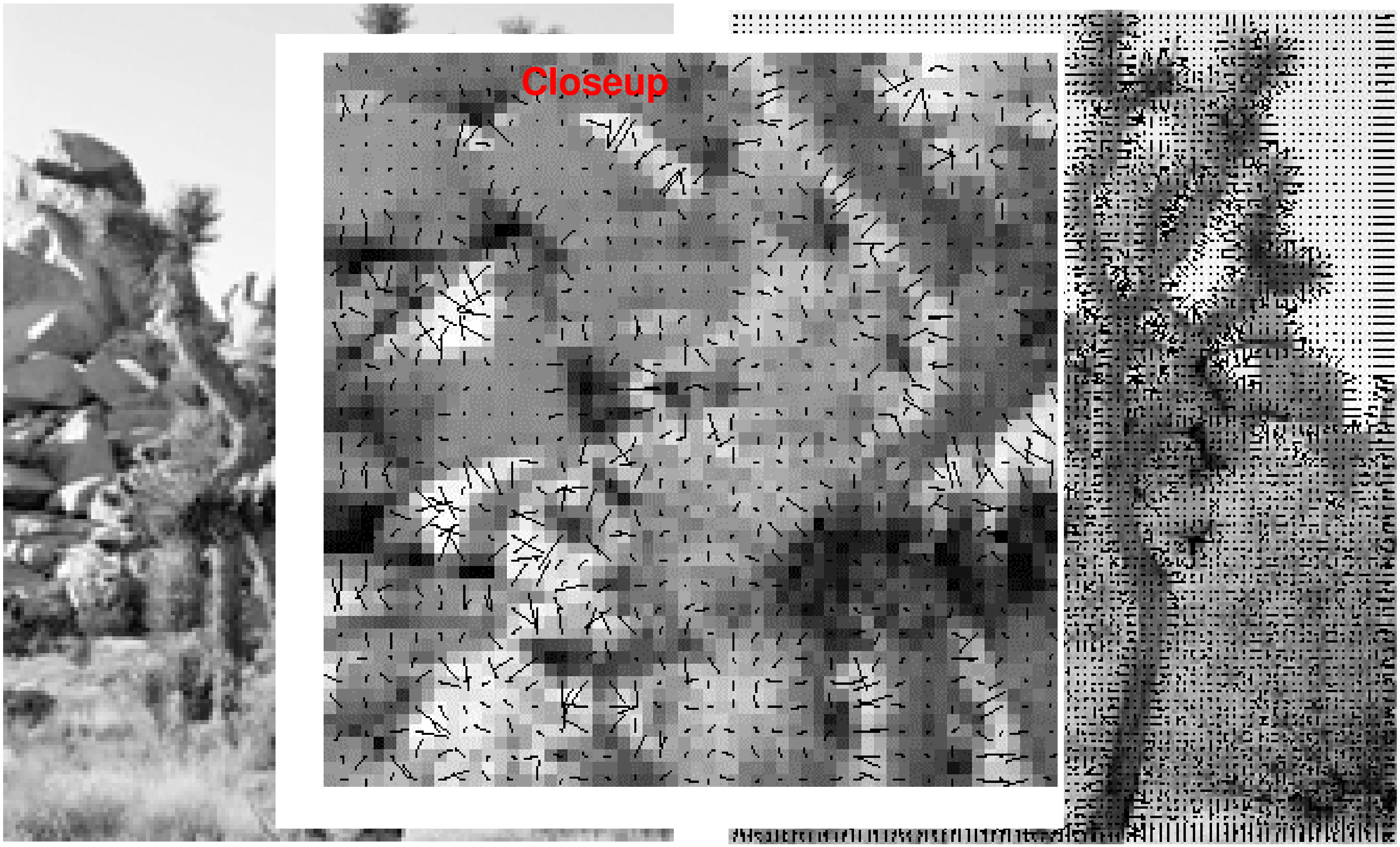
It can be shown that since C is symmetric:

$$C = Q^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} Q$$

So every case is like a rotated version of the one on last slide.

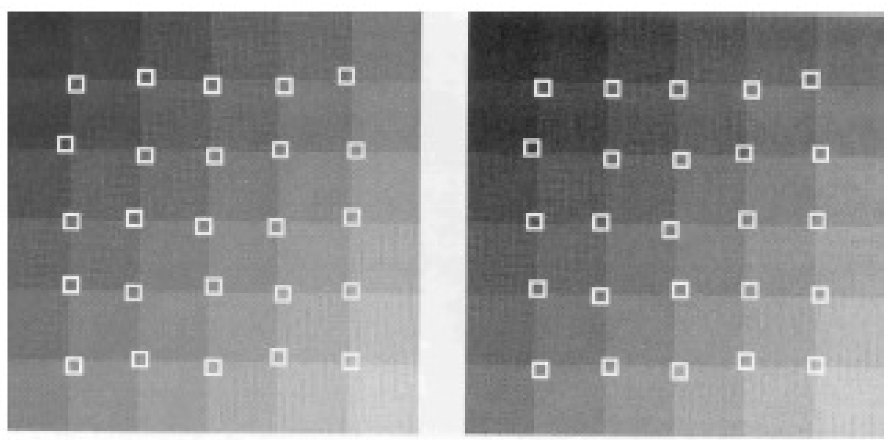


Gradient Orientation



Corner Detection Summary

- if the area is a region of constant intensity, both eigenvalues will be very small.
- if it contains an edge, there will be one large and one small eigenvalue (the eigenvector associated with the large eigenvalue will be parallel to the image gradient).
- if it contains edges at two or more orientations (i.e., a corner), there will be two large eigenvalues (the eigenvectors will be parallel to the image gradients).



Corner Detection Algorithm

Algorithm

Input: image f, threshold t for λ_2, size of Q

(1) Compute the gradient over the entire image f

(2) For each image point p:

(2.1) form the matrix C over the neighborhood Q of p
(2.2) compute λ_2, the smaller eigenvalue of C
(2.3) if $\lambda_2 > t$, save the coordinates of p in a list L

(3) Sort the list in decreasing order of λ_2

(4) Scanning the sorted list top to bottom: delete all the points that appear in the list that are in the same neighborhood Q with p