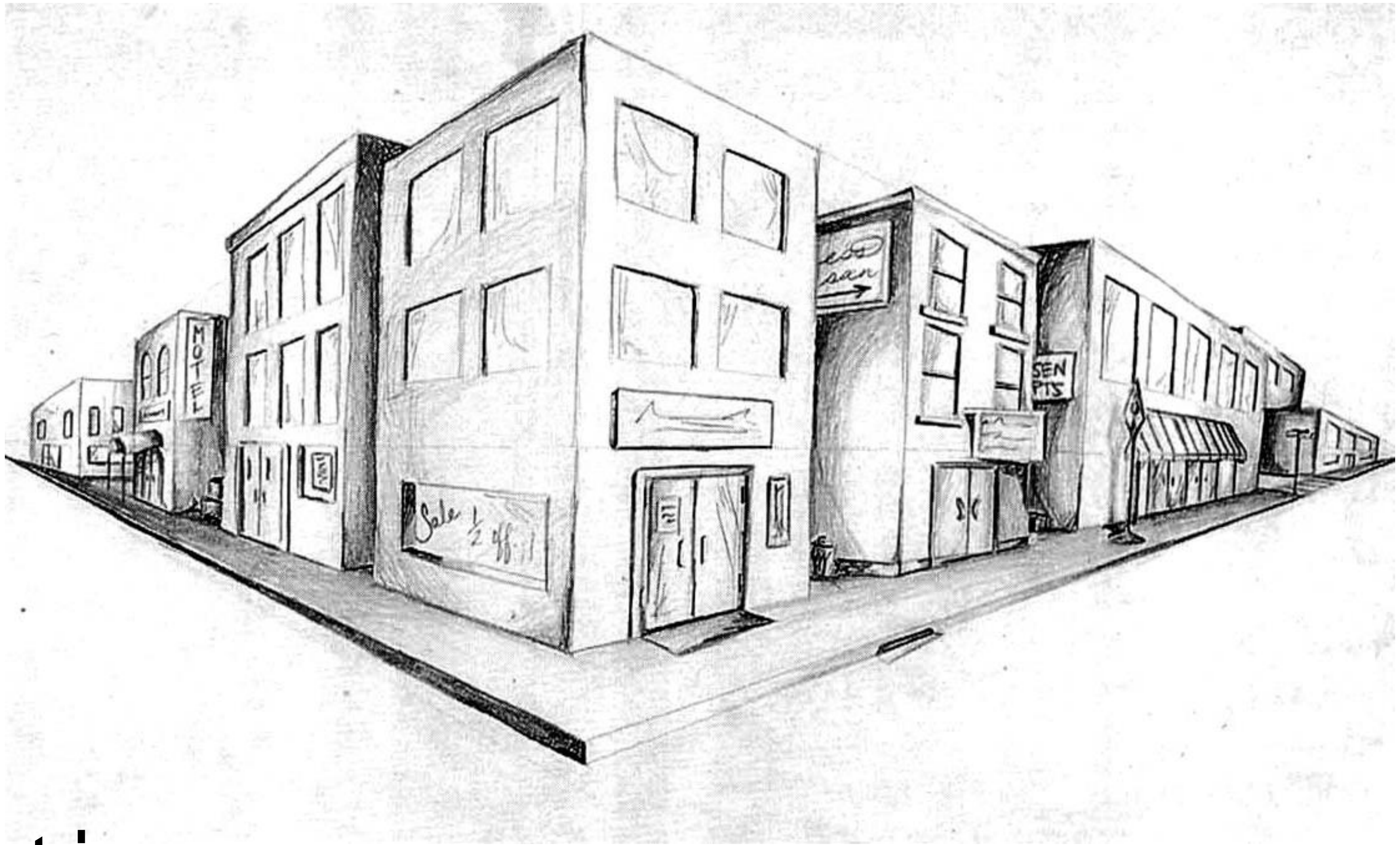
3D Vision and Machine Perception

Prof. Kyungdon Joo

3D Vision & Robotics Lab.

AI Graduate School (AIGS) & Computer Science and Engineering (CSE)

Some materials, figures, and slides (used for this course) are from textbooks, published papers, and other open lectures

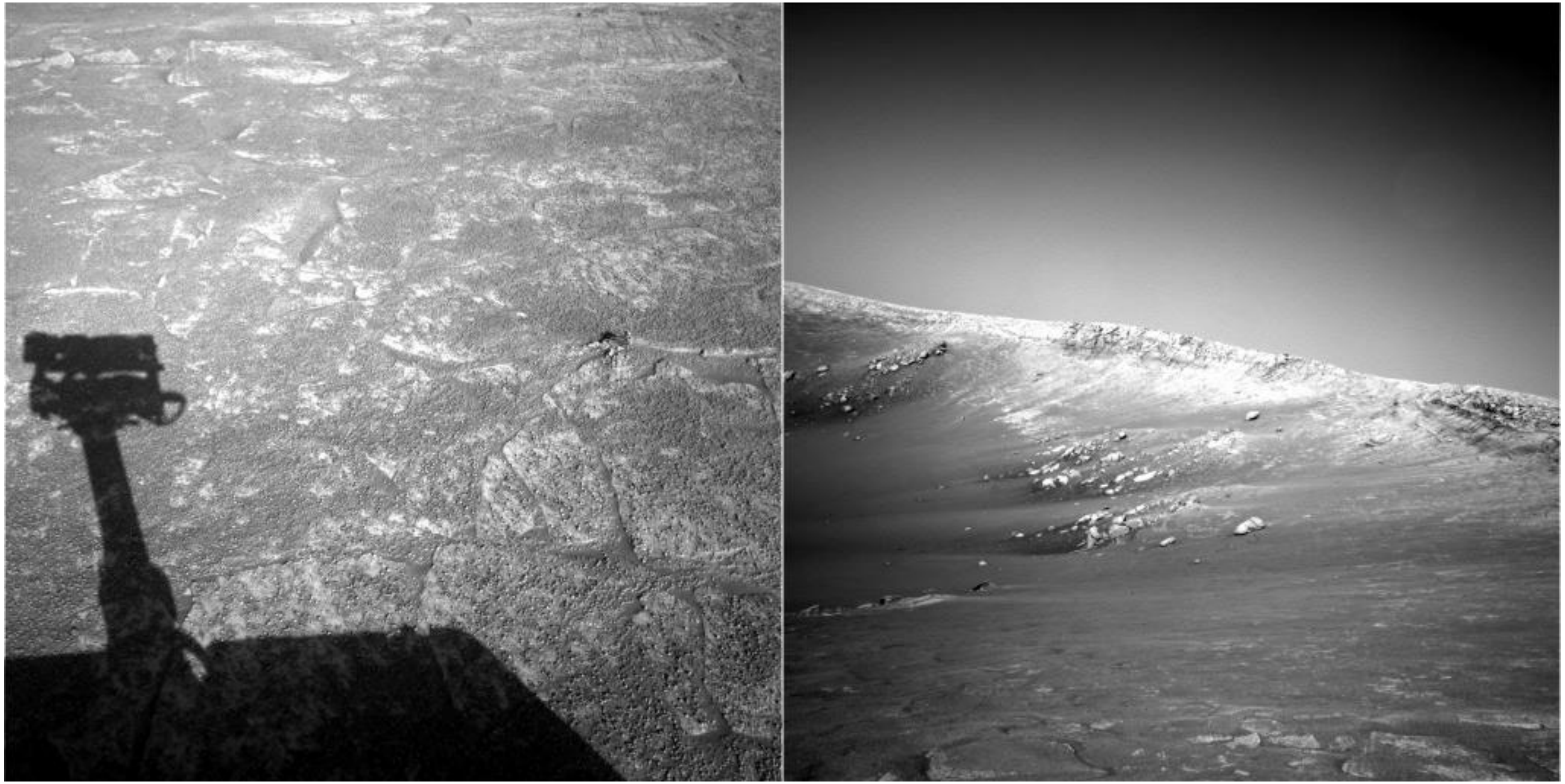


Detecting corners

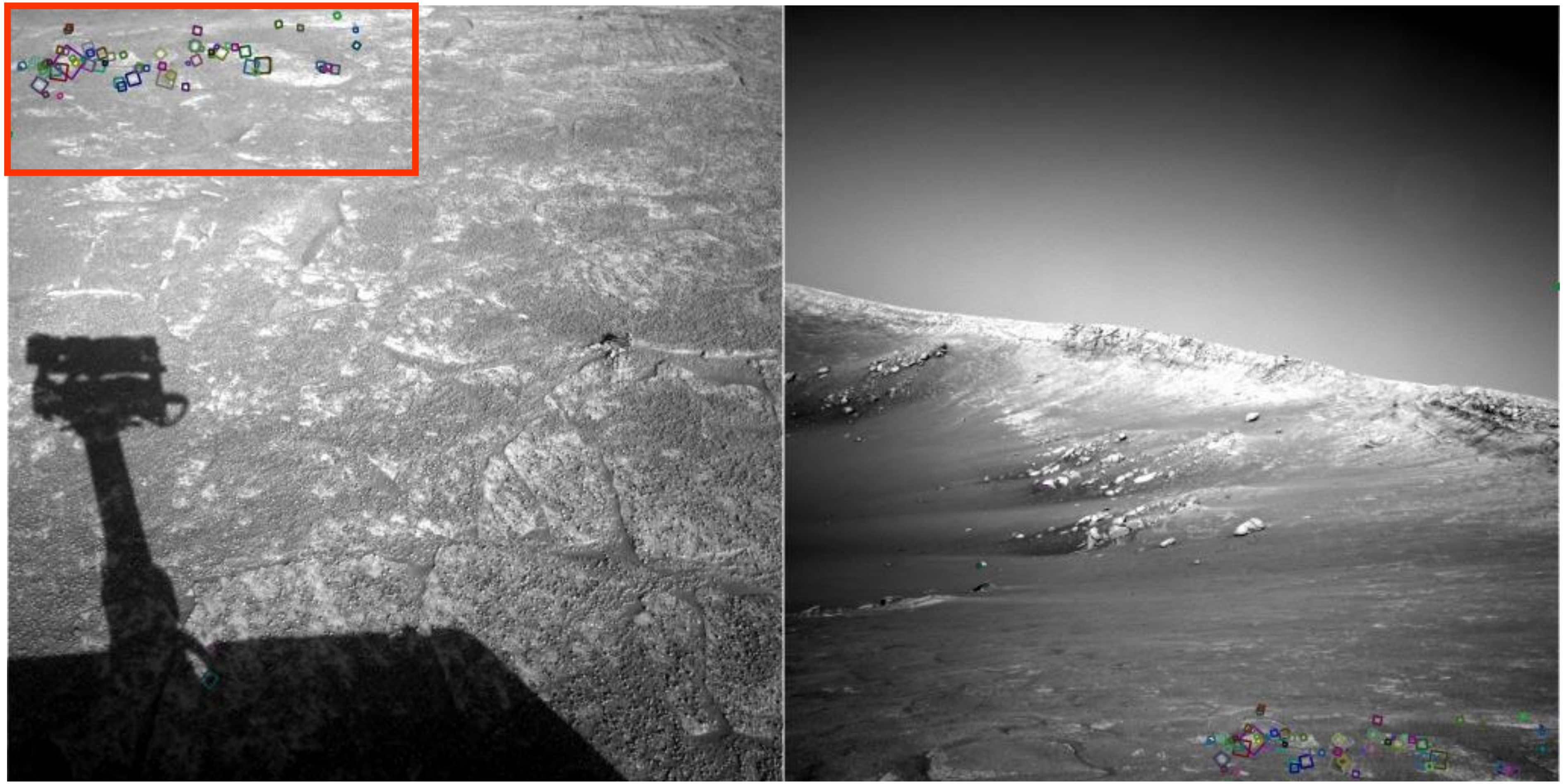
Why detect corners?

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

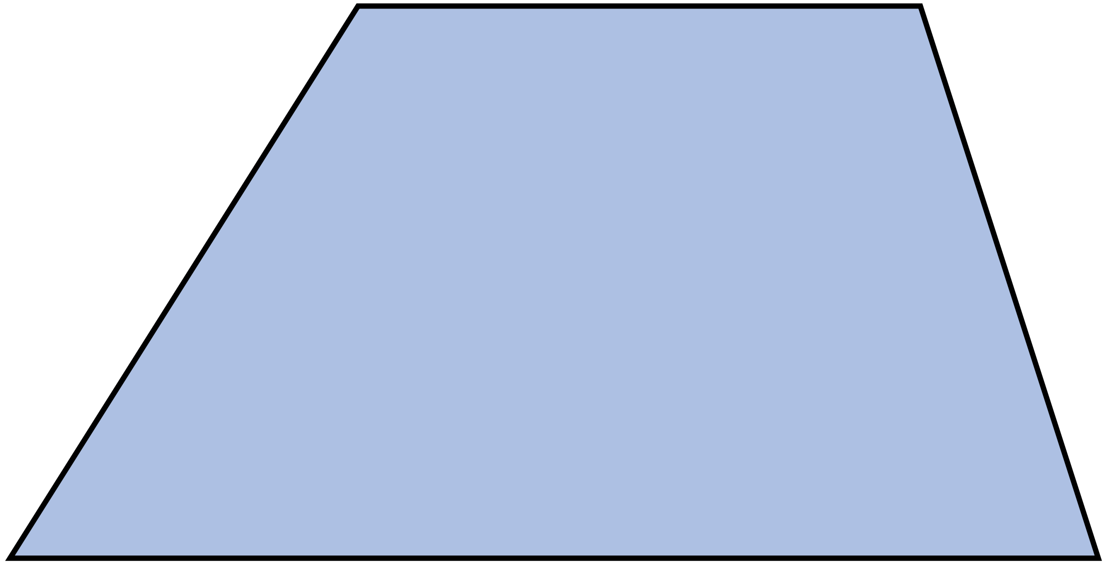
- Where are the corresponding points?



NASA Mars Rover images

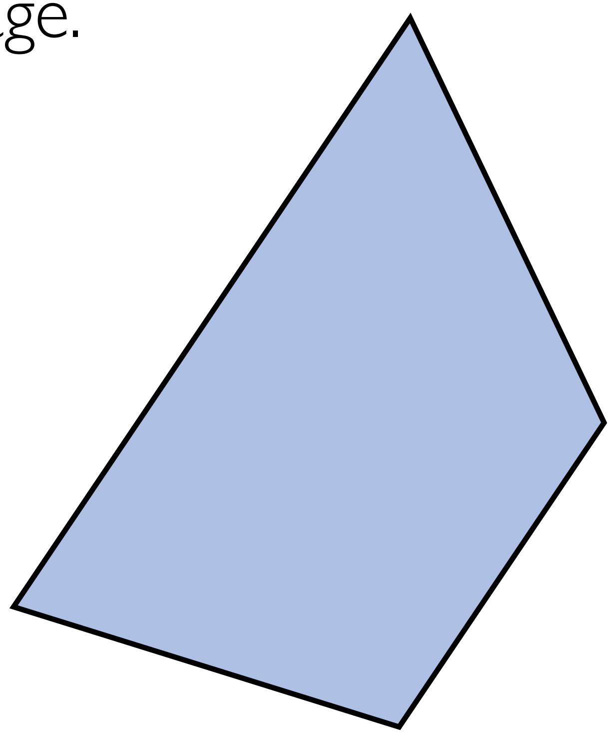


- Pick a point in the image.
- Find it again in the next image.



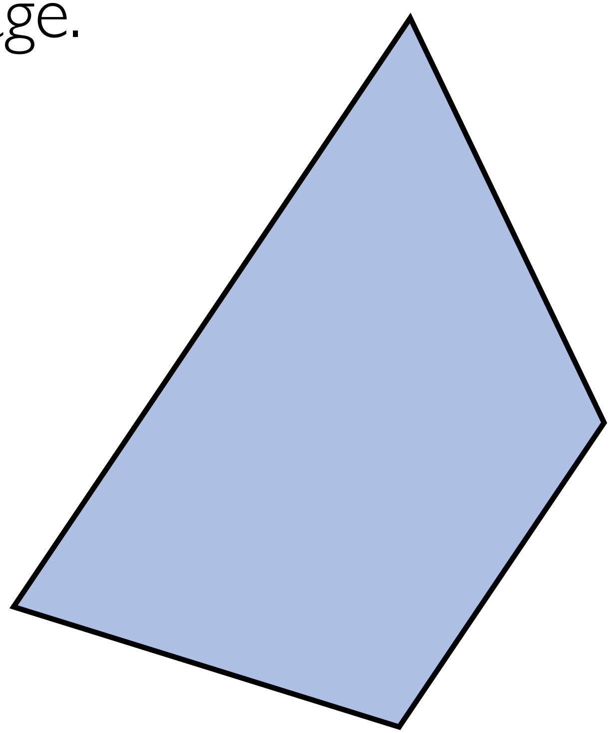
What type of feature would you select?

- Pick a point in the image.
- Find it again in the next image.



What type of feature would you select?

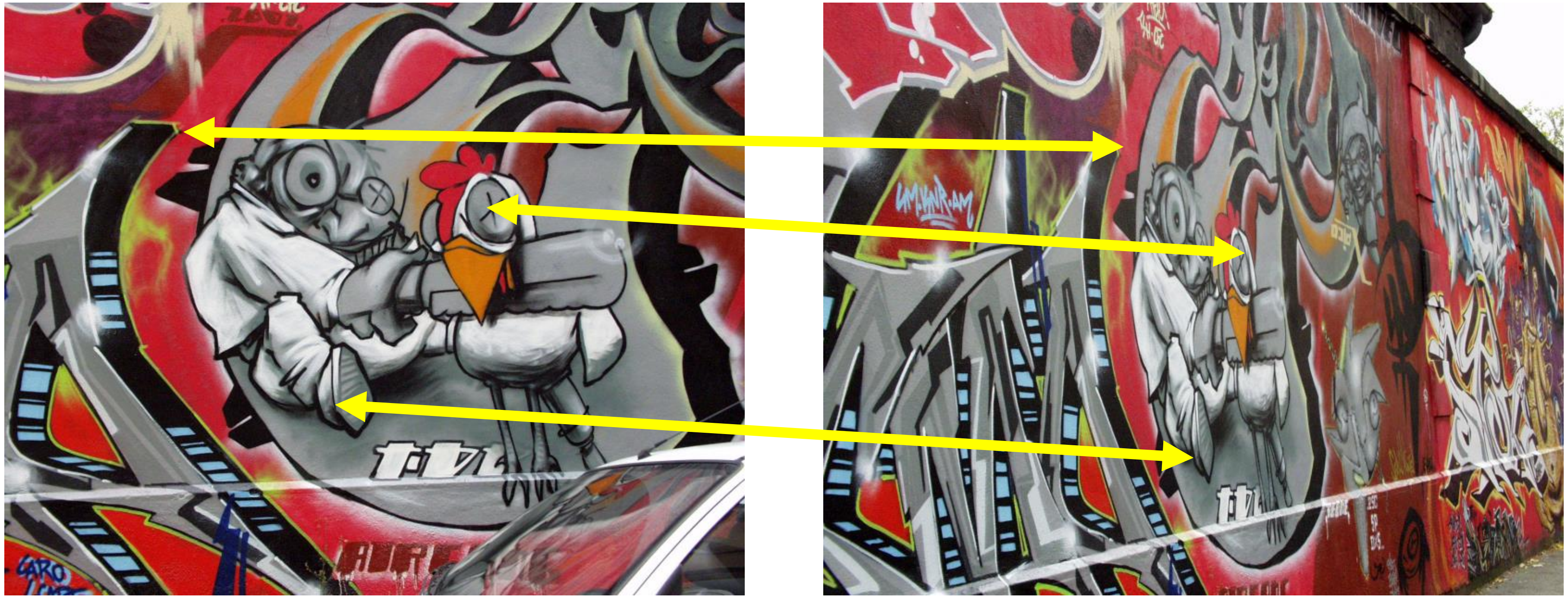
- Pick a point in the image.
- Find it again in the next image.



What type of feature would you select?

a corner

Image matching (or feature matching)

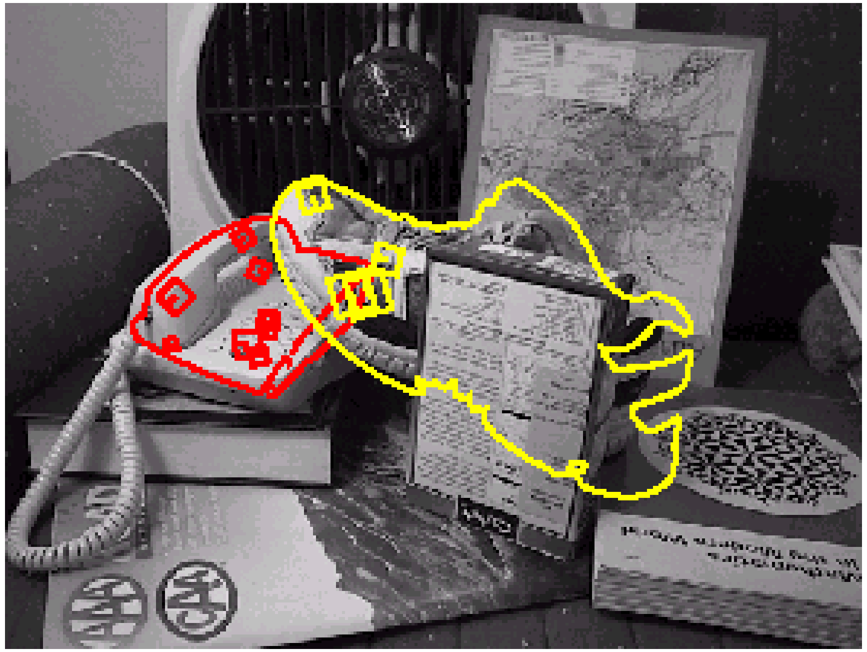


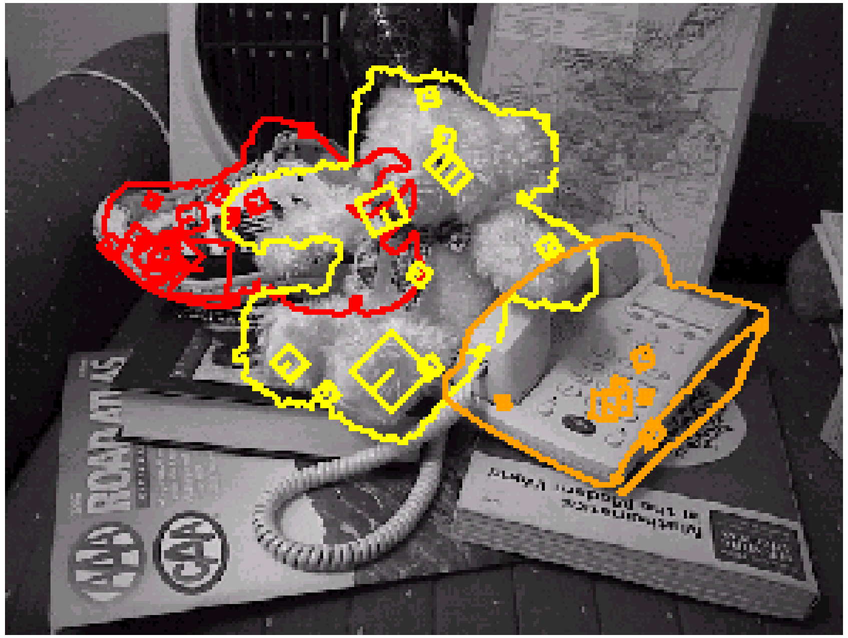
3D object recognition

Database of 3D objects

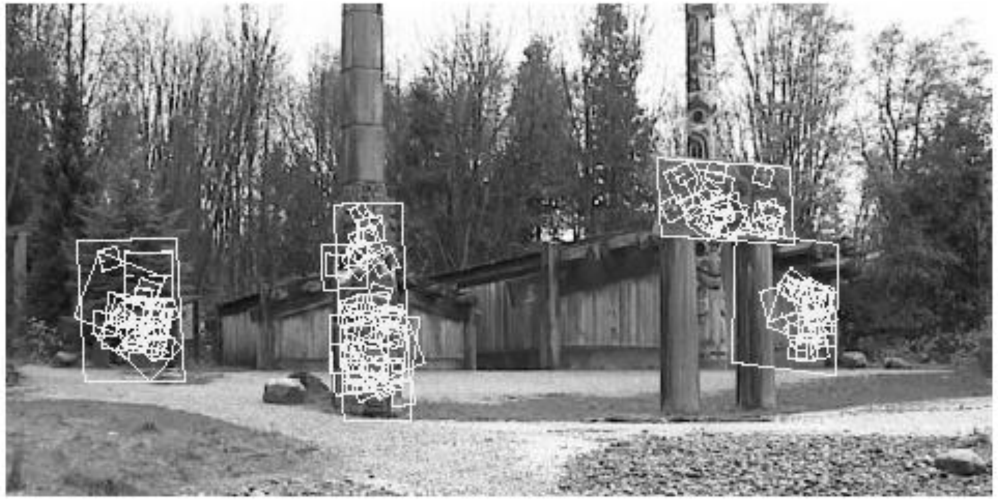
3D objects recognition

- Recognition under occlusion

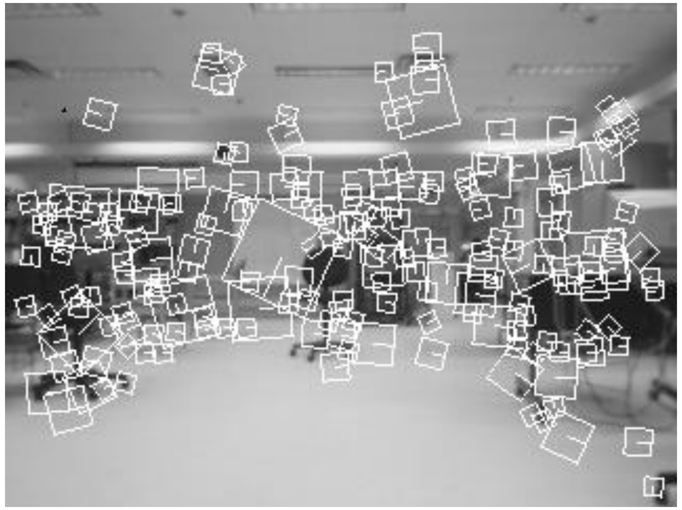




Location recognition

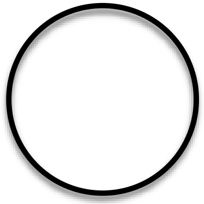


Robot localization



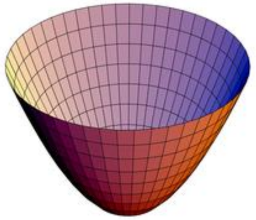
Visualizing quadratics

- Equation of a circle



$$1 = x^2 + y^2$$

- Equation of a 'bowl' (paraboloid)



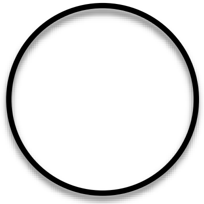
$$f(x, y) = x^2 + y^2$$

If you slice the bowl at

$$f(x, y) = 1$$

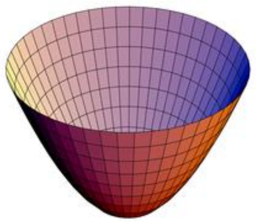
what do you get?

- Equation of a circle



$$1 = x^2 + y^2$$

- Equation of a 'bowl' (paraboloid)

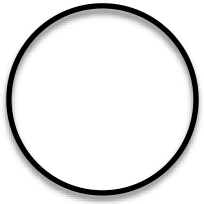


$$f(x, y) = x^2 + y^2$$

If you slice the bowl at

$$f(x, y) = 1$$

what do you get?



- The Equation

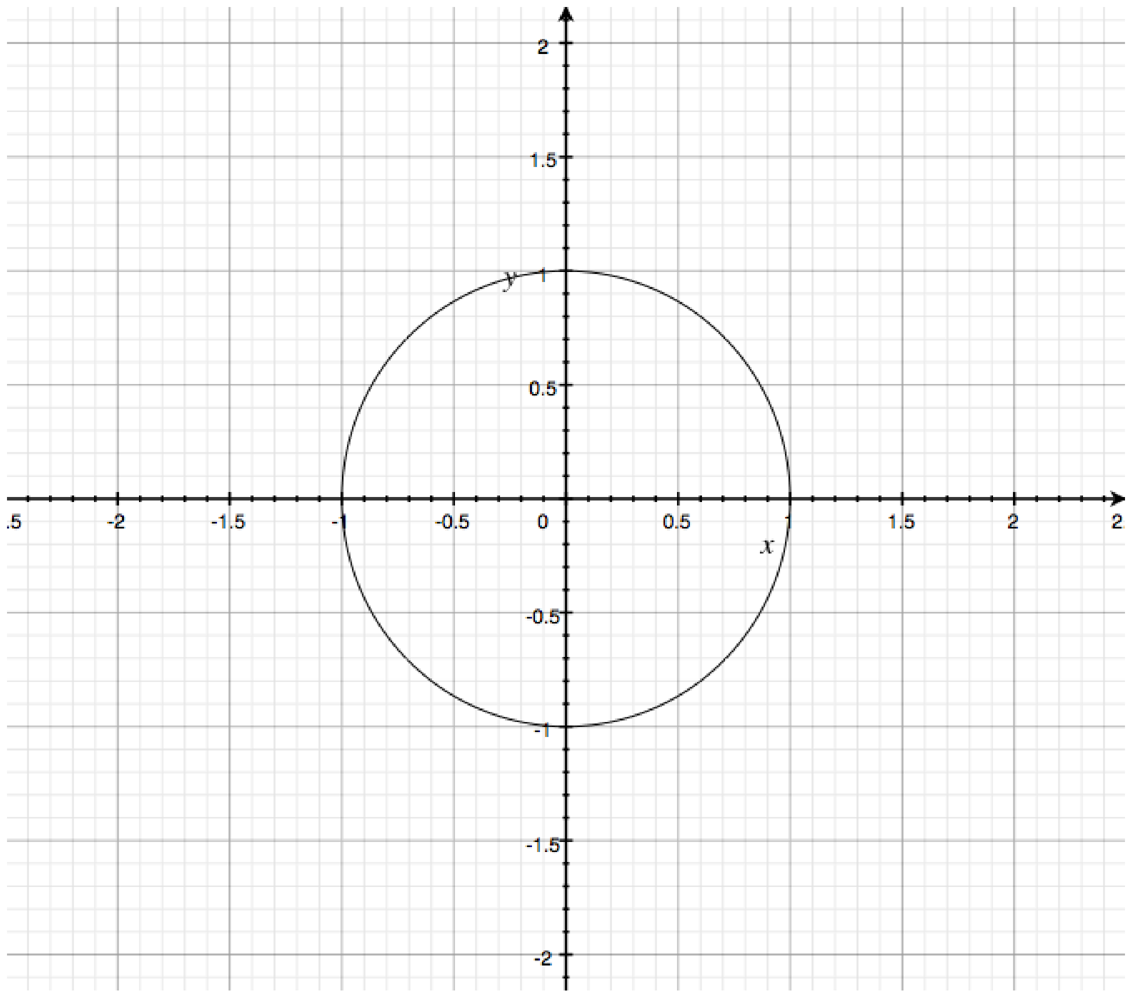
$$f(x, y) = x^2 + y^2$$

- can be written in matrix form like this…

$$f(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

'sliced at 1'



- What happens if you increase coefficient on x?

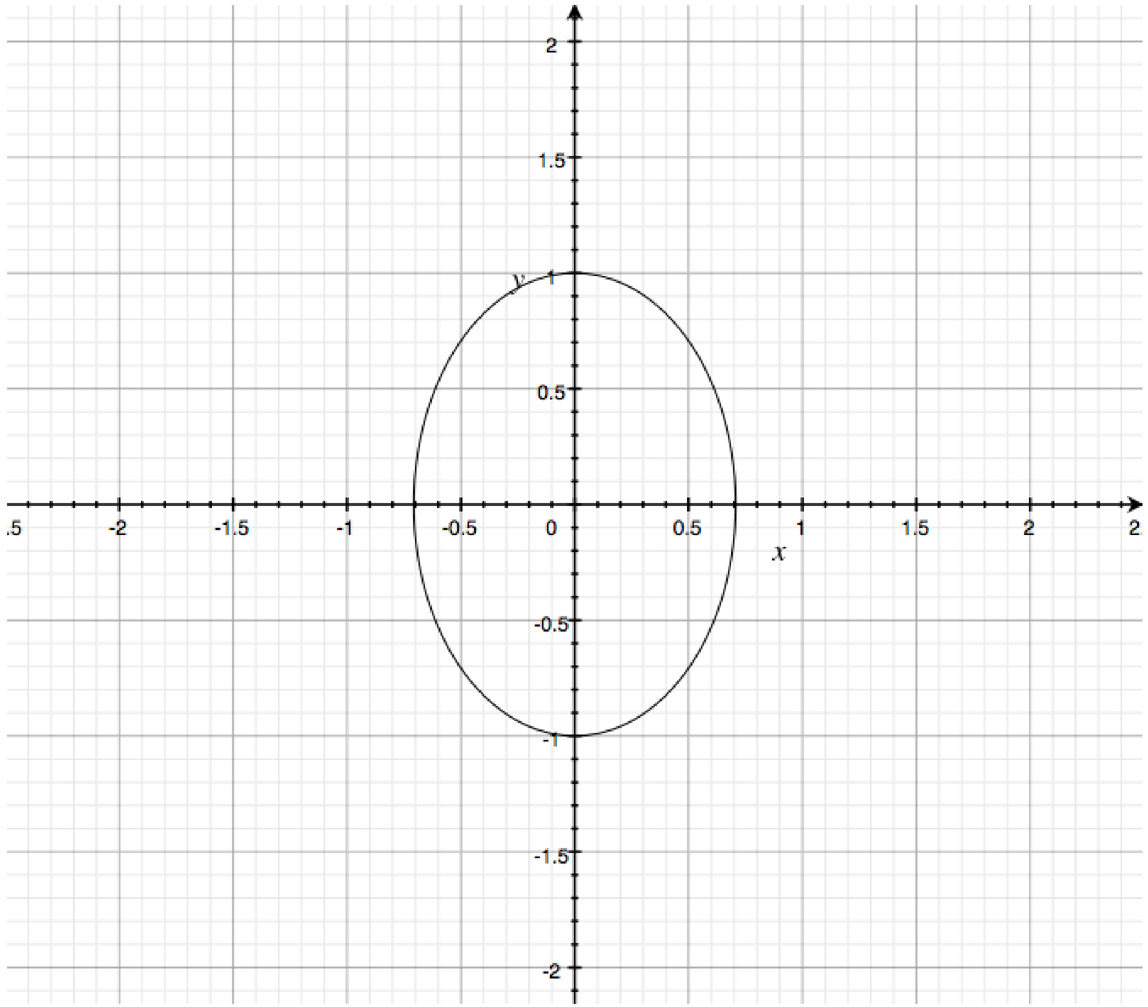
$$f(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

'sliced at 1'

- What happens if you increase coefficient on x?

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and slice at 1



decrease width in **x!**

- What happens if you increase coefficient on y?

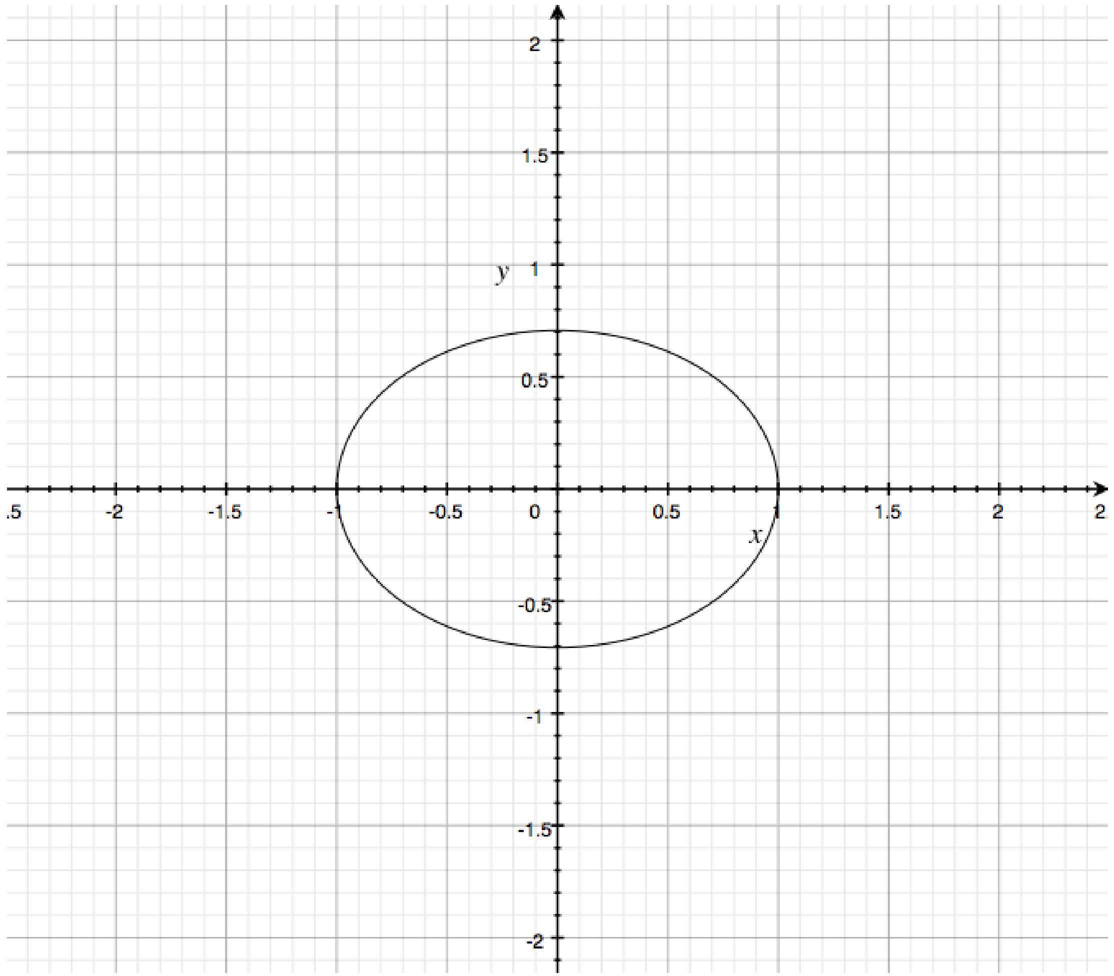
$$f(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and slice at 1

- What happens if you increase coefficient on y?

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and slice at 1



decrease width in y!

- The Equation

$$f(x, y) = x^2 + y^2$$

- can be written in matrix form like this…

$$f(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- What's the shape?
- What are the eigenvectors?
- What are the eigenvalues?

- The Equation

$$f(x,y) = x^2 + y^2$$

- can be written in matrix form like this…

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Result of Singular Value Decomposition (SVD)

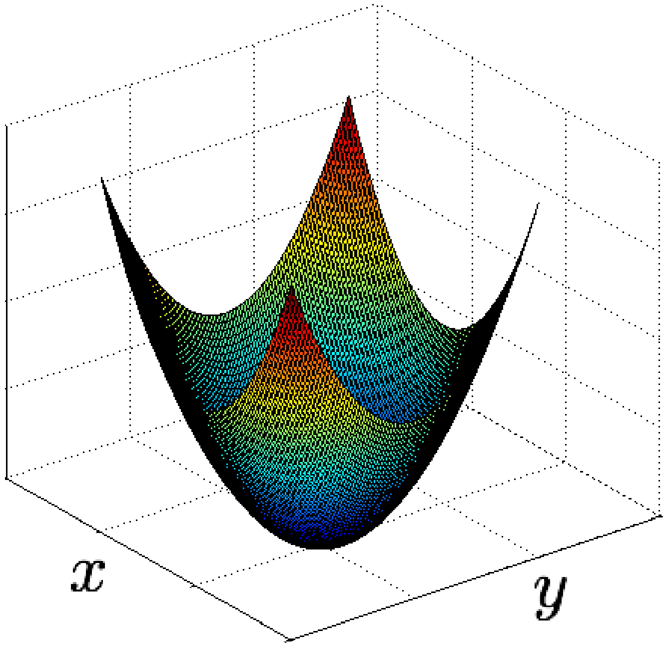
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \underbrace{\overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\text{eigenvectors}}}_{\text{axis of the 'ellipse slice'}} \underbrace{\overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\text{eigenvalues along diagonal}}}_{\text{Inverse sqr of length of the quadratic along the axis}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{\top}$$

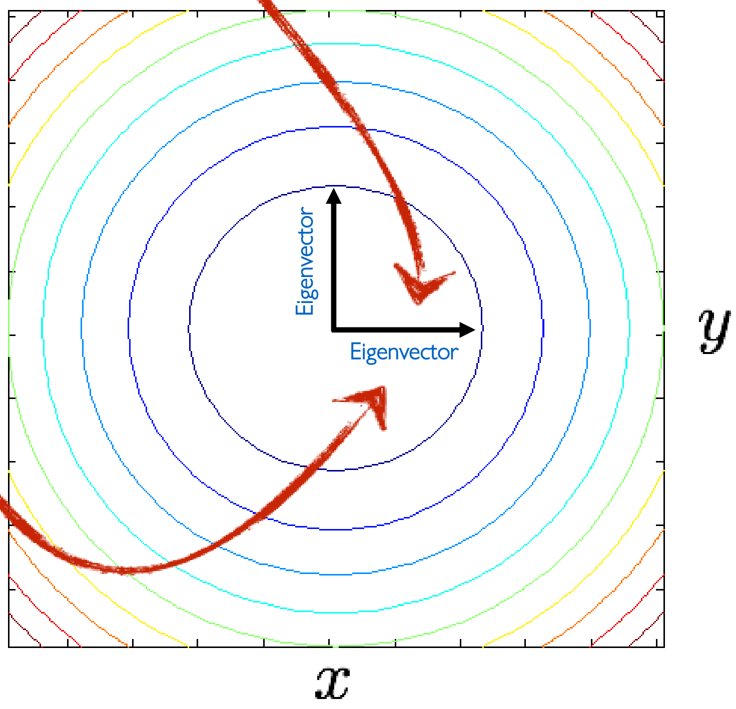
Eigenvectors Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

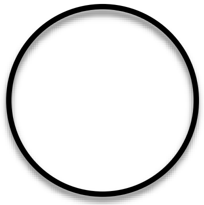
Eigenvectors

Inverse sqr of the size of the axis



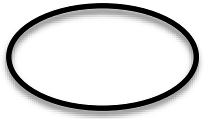


- Recall



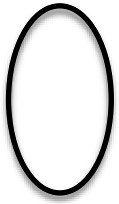
$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- you can smash this bowl in the y direction



$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- you can smash this bowl in the x direction



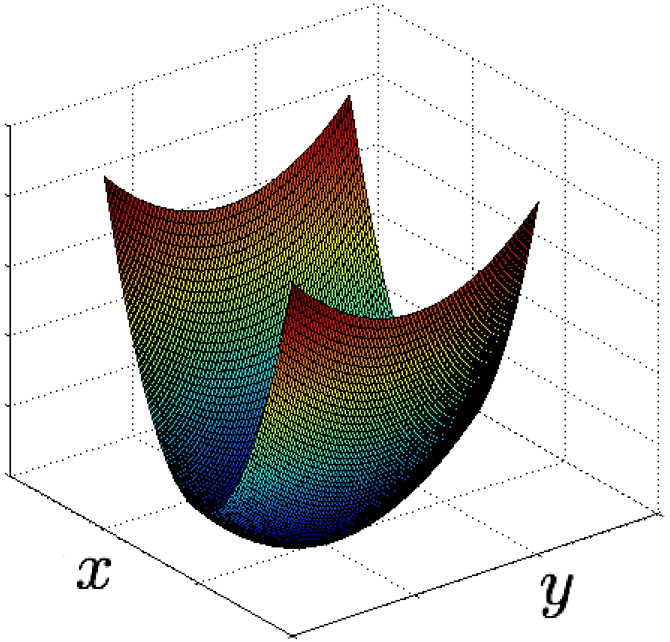
$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

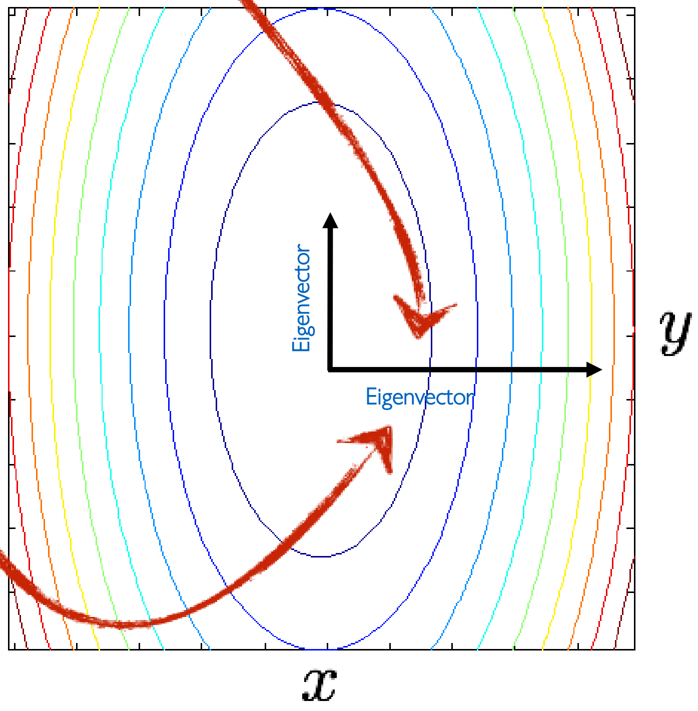
Eigenvectors Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

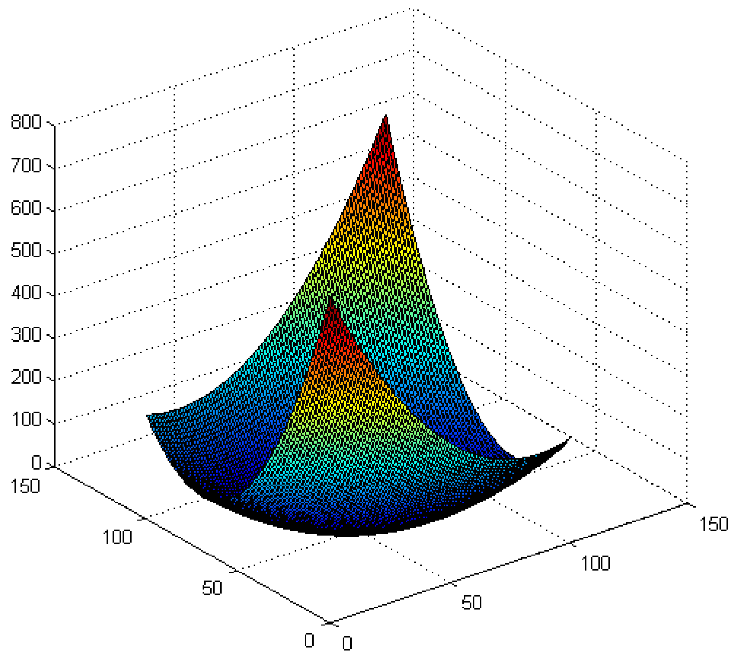
Eigenvectors

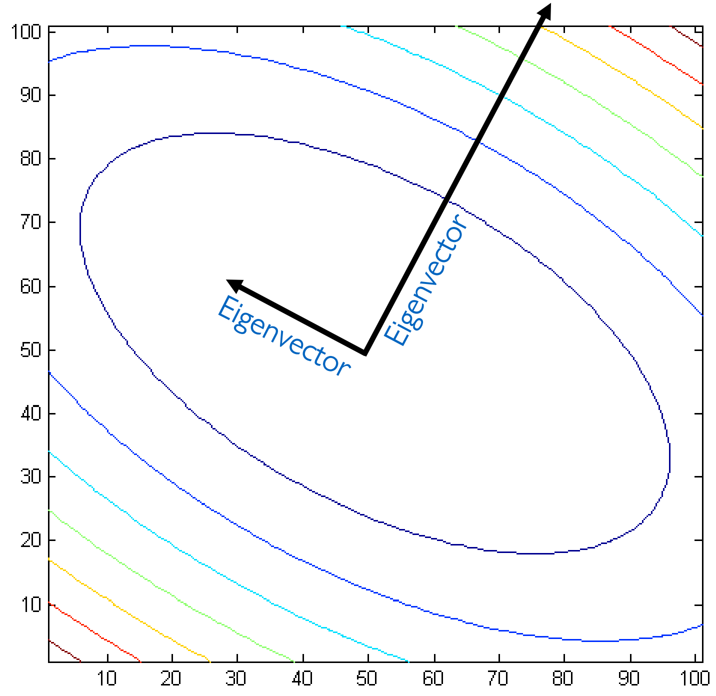
Inverse sqr of the size of the axis





$$\mathbf{A} = \begin{bmatrix} 3.25 & 1.30 \\ 1.30 & 1.75 \end{bmatrix} = \underbrace{\begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}}_{\text{Eigenvectors}} \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}}^{\text{Eigenvalues}} \underbrace{\begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}^T}_{\text{Eigenvectors}}$$

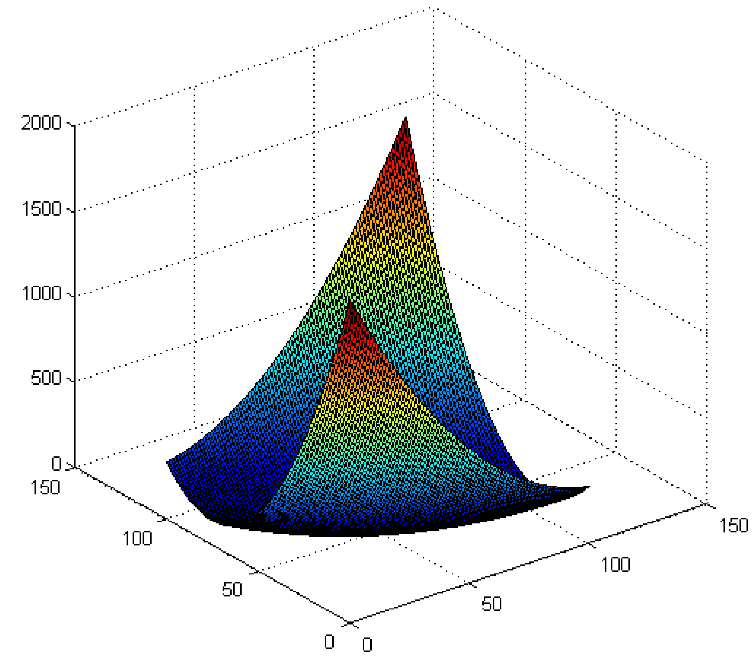


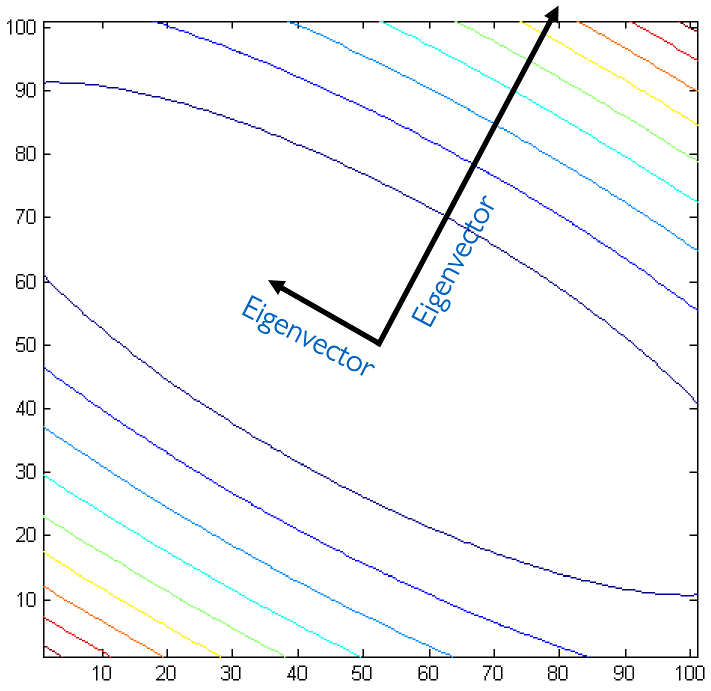


Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 7.75 & 3.90 \\ 3.90 & 3.25 \end{bmatrix} = \begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}^T$$

Eigenvectors Eigenvectors



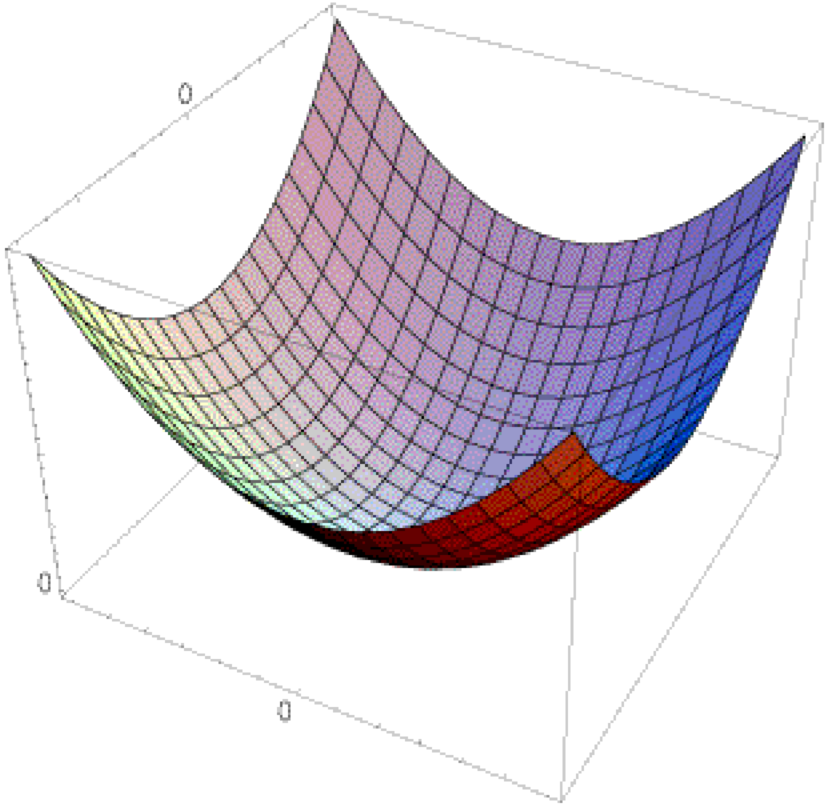


Error function for Harris corners

- We will need this to understand the…
  The surface *E(u,v)* is locally approximated by a quadratic form

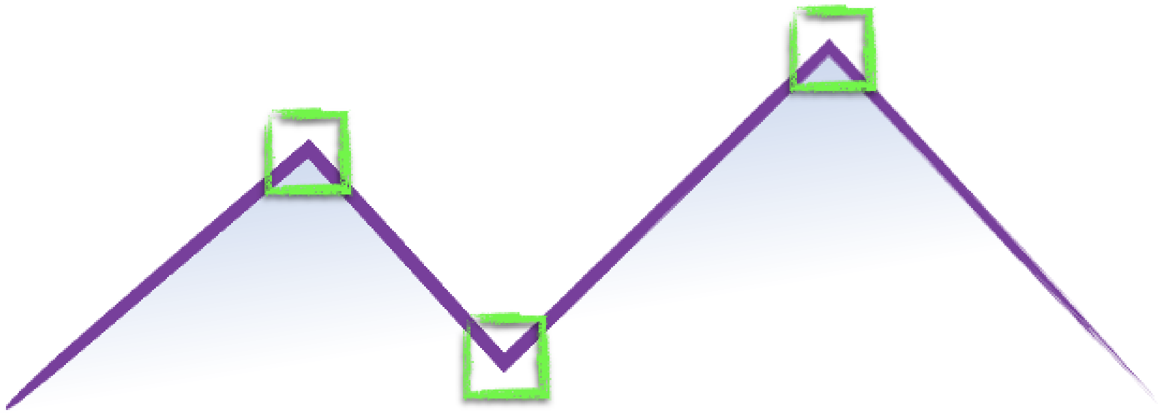
$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

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

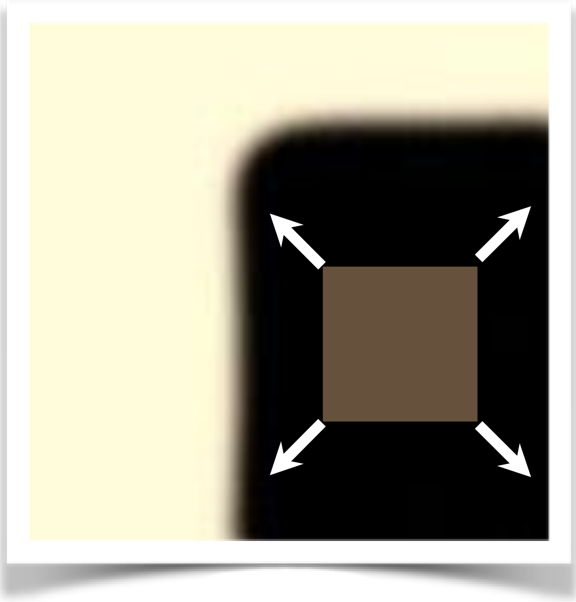


Harris corner detector

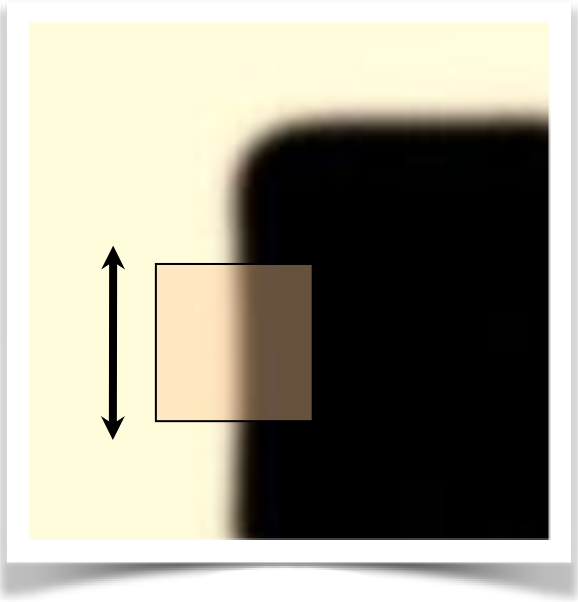
How do you find a corner?



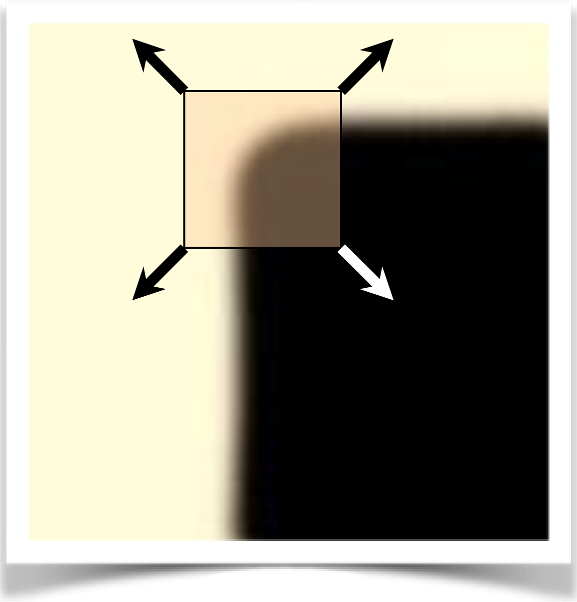
- Easily recognized by looking through a small window
- Shifting the window should give large change in intensity



"flat" region:
no change in all
directions



"edge":
no change along the edge
direction



"corner":
significant change in all
directions

[Moravec 1980]

- Design a program to detect corners
  (hint: use image gradients)

Finding corners (a.k.a. PCA)

1. Compute image gradients over small region

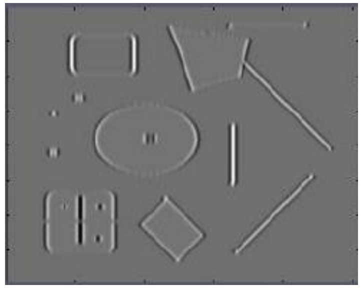
2. Subtract mean from each image gradient

3. Compute the covariance matrix

4. Compute eigenvectors and eigenvalues

5. Use threshold on eigenvalues to detect corners

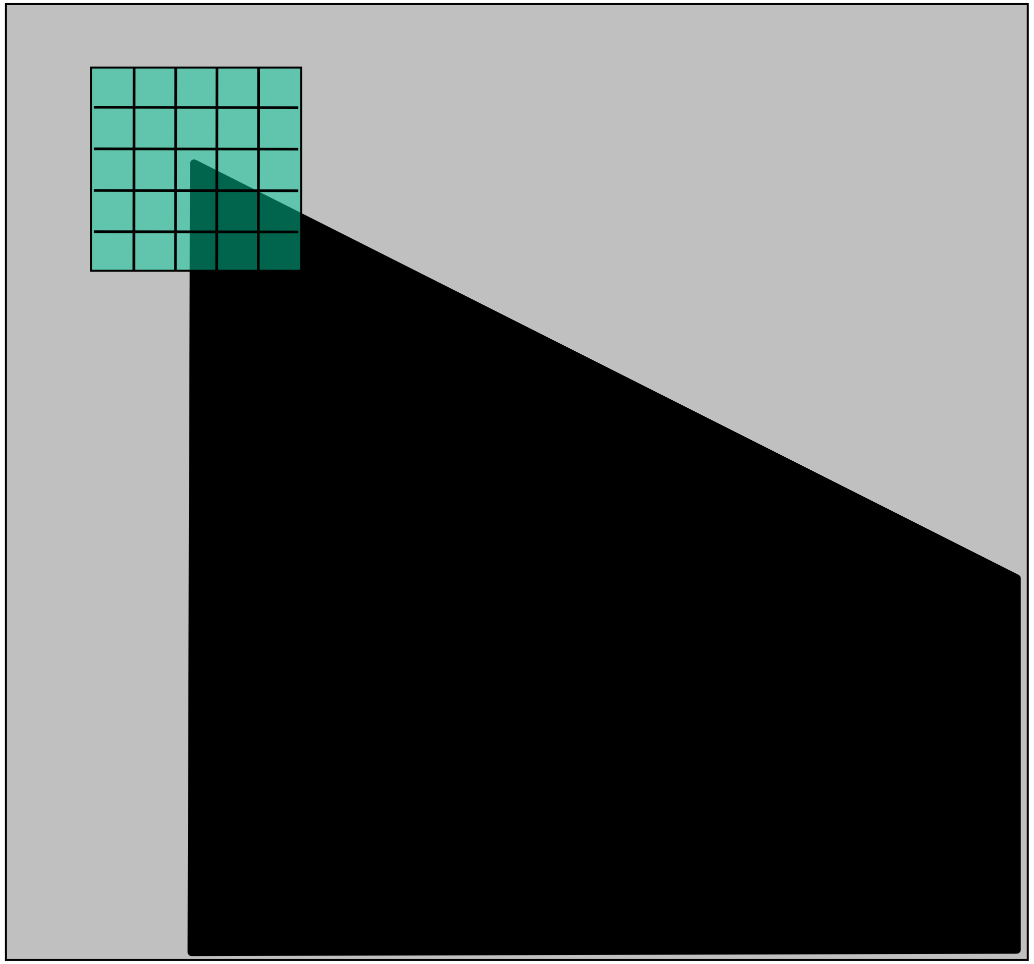
$$I_x = \frac{\partial I}{\partial x} \qquad I_y = \frac{\partial I}{\partial y}$$



$$\begin{bmatrix} \sum\limits_{p \in P} I_x I_x & \sum\limits_{p \in P} I_x I_y \\ \sum\limits_{p \in P} I_y I_x & \sum\limits_{p \in P} I_y I_y \end{bmatrix}$$

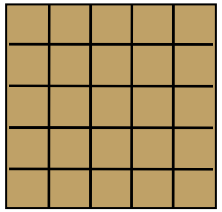
1. Compute image gradients over a small region
(not just a single pixel)

1. Compute image gradients over a small region



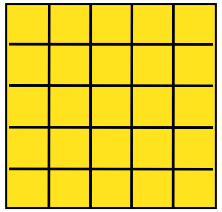
array of x gradients

$$I_x = \frac{\partial I}{\partial x}$$

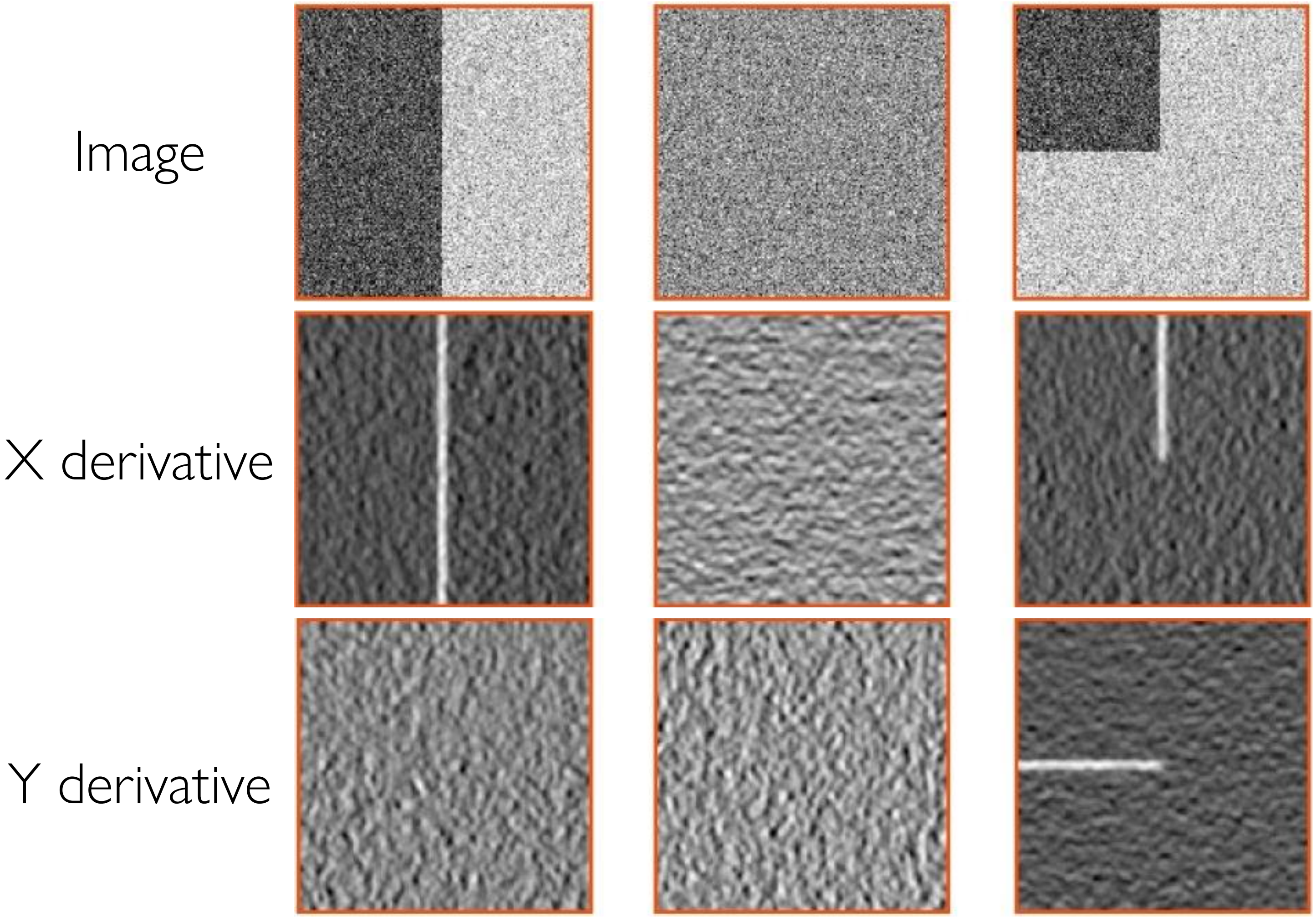


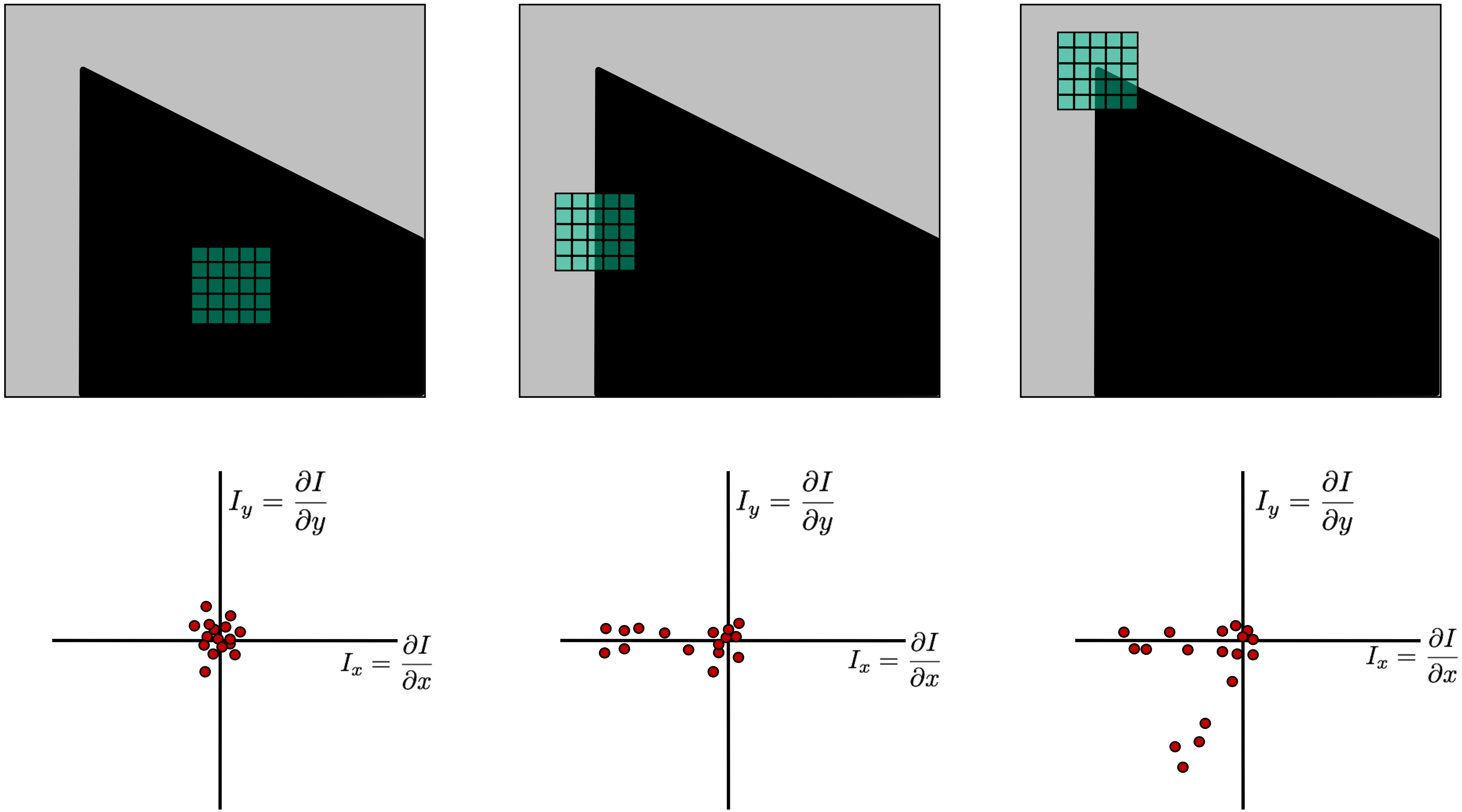
array of y gradients

$$I_y = \frac{\partial I}{\partial y}$$

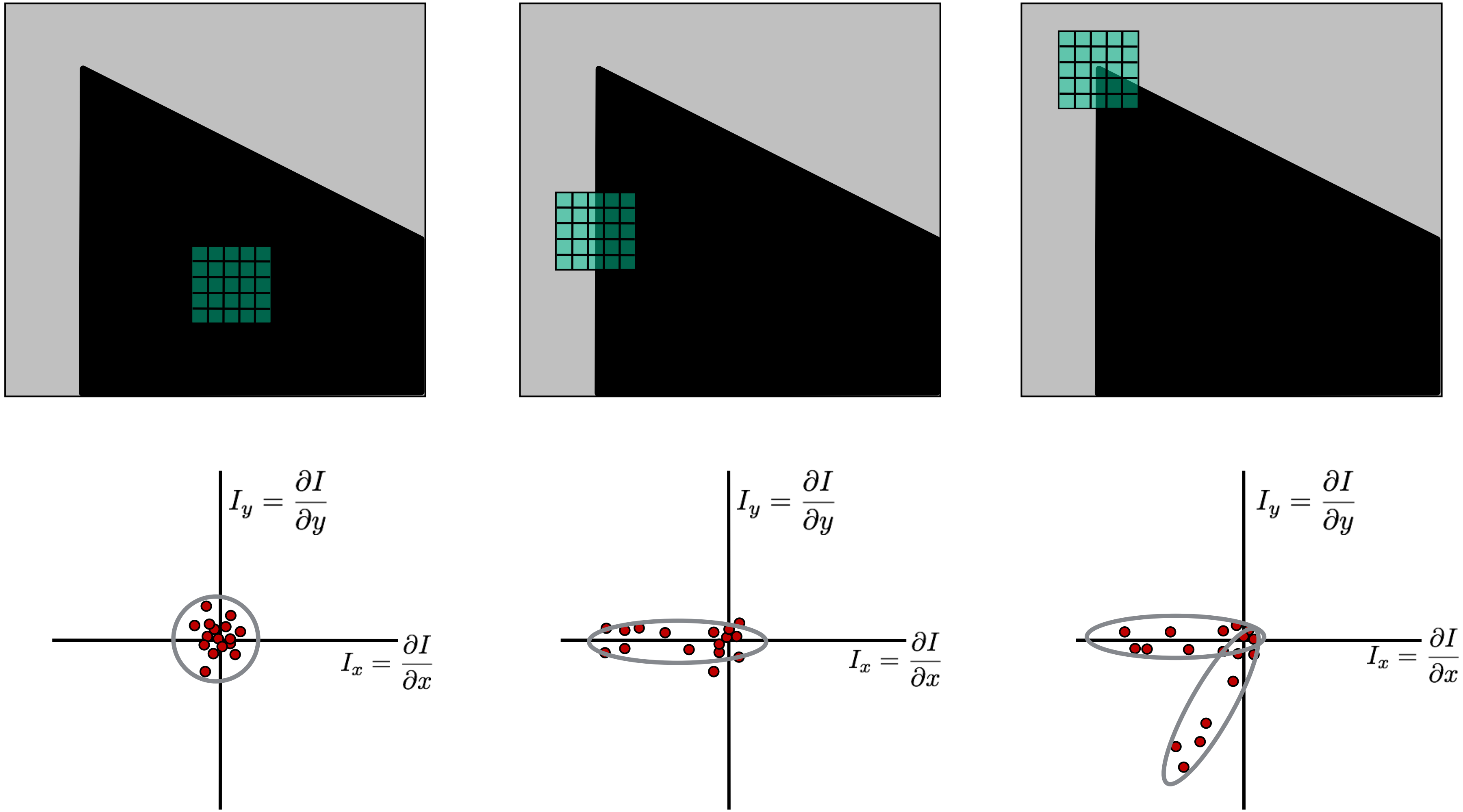


Visualization of gradients

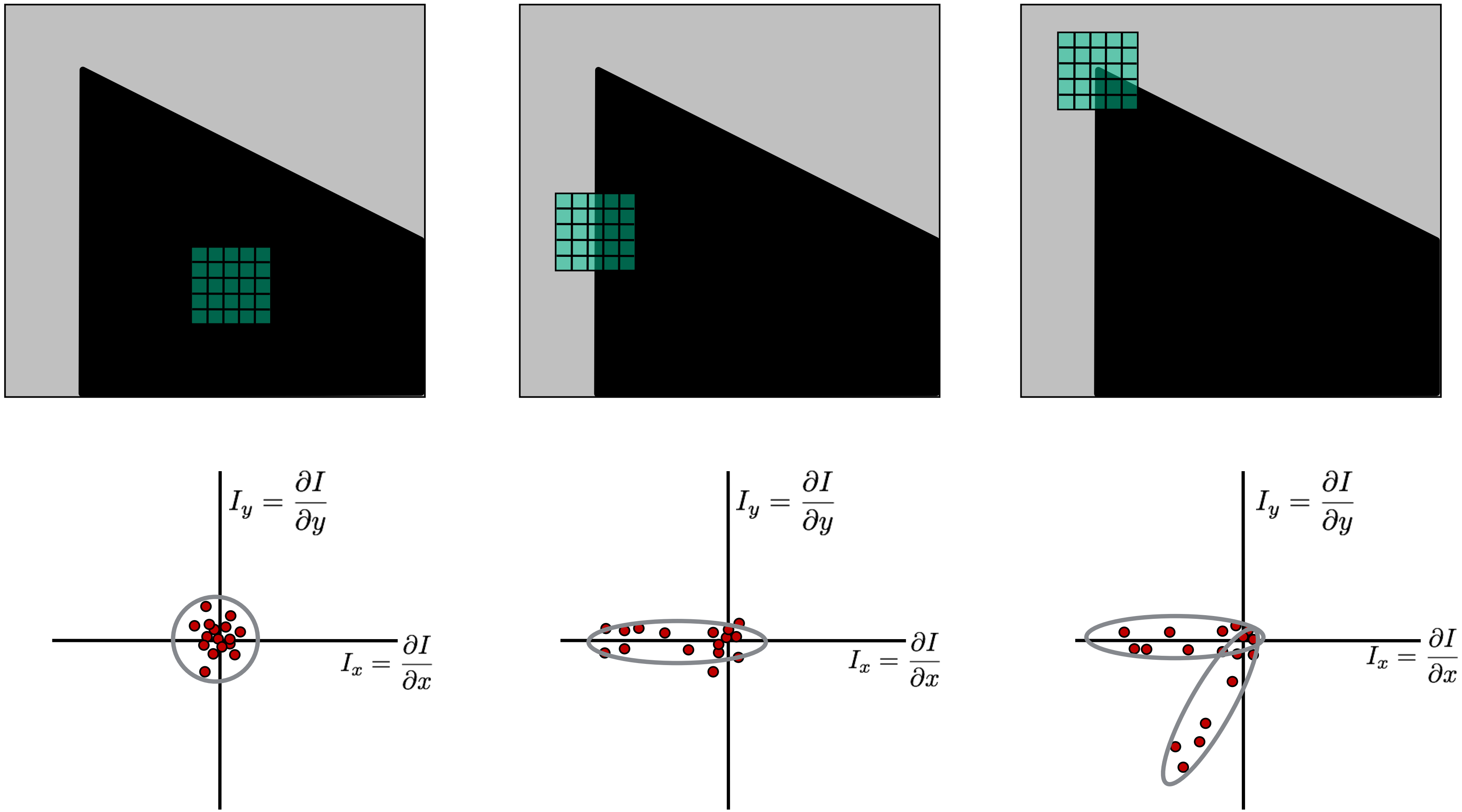




- What does the distribution tell you about the region?



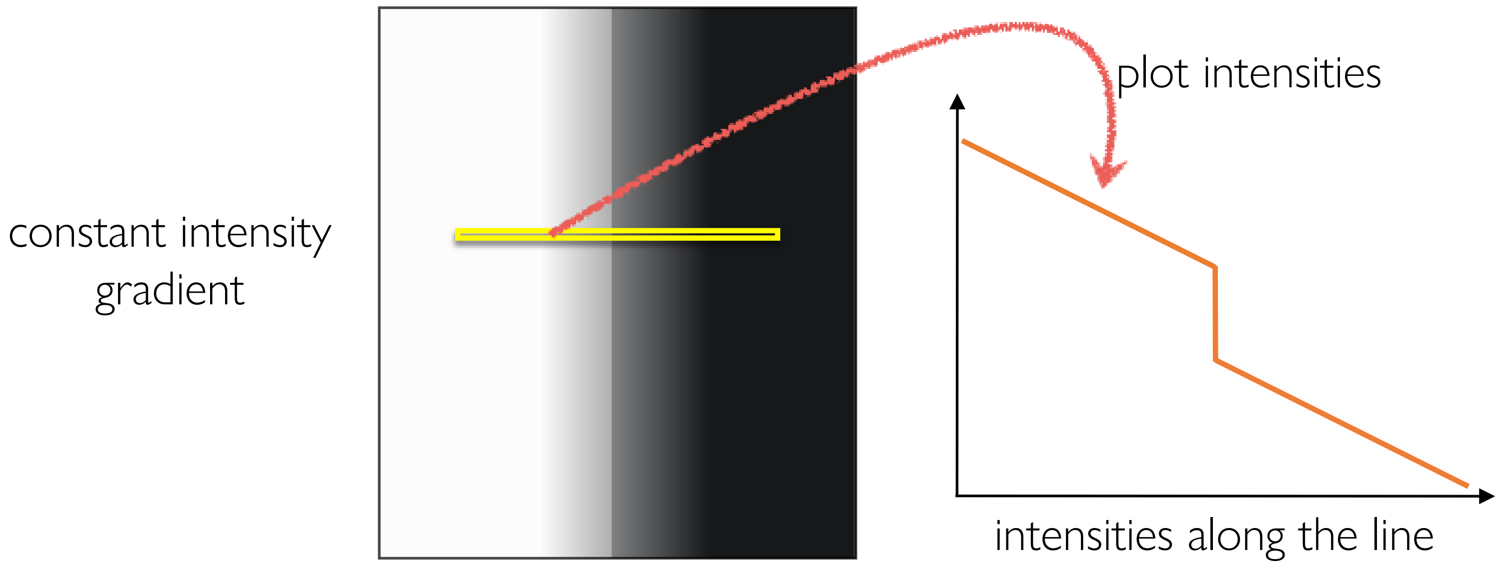
- Distribution reveals edge orientation and magnitude



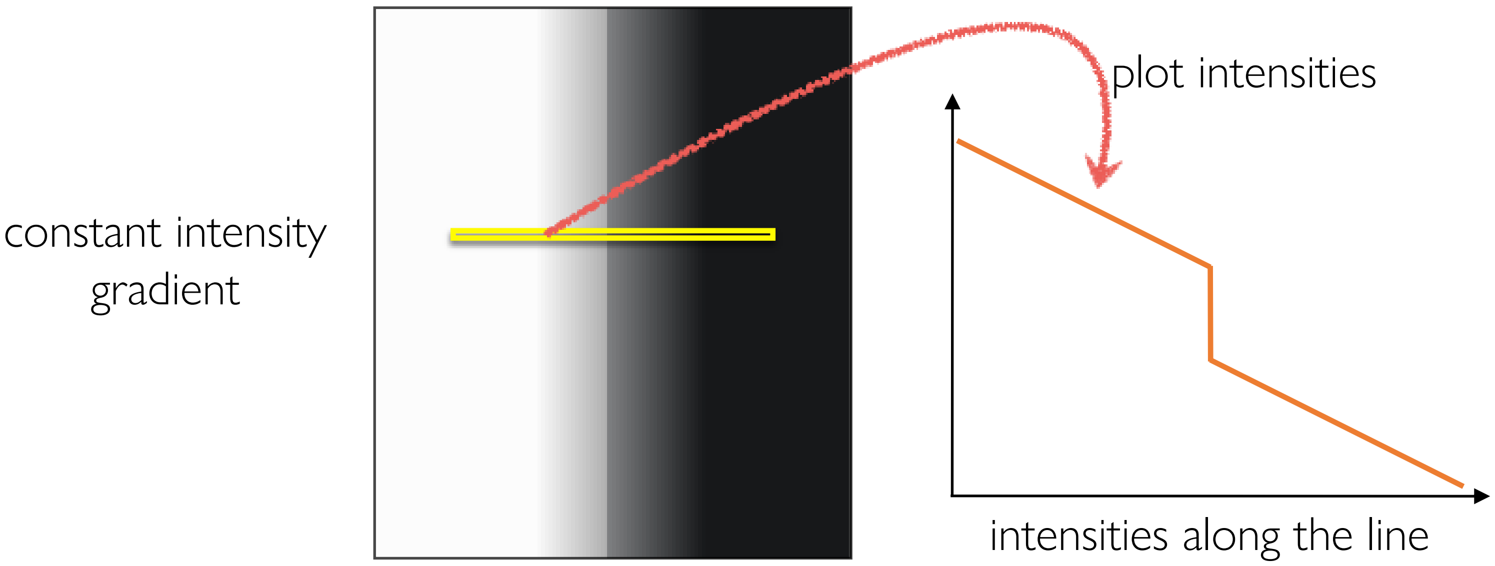
- How do you quantify orientation and magnitude?

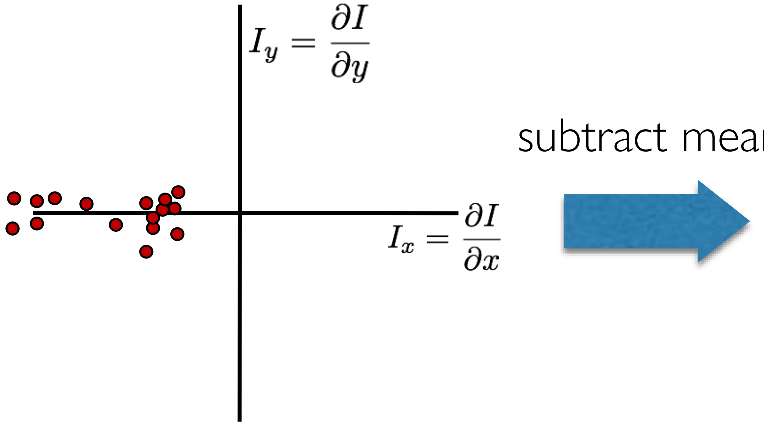
2. Subtract the mean from each image gradient

2. Subtract the mean from each image gradient



2. Subtract the mean from each image gradient

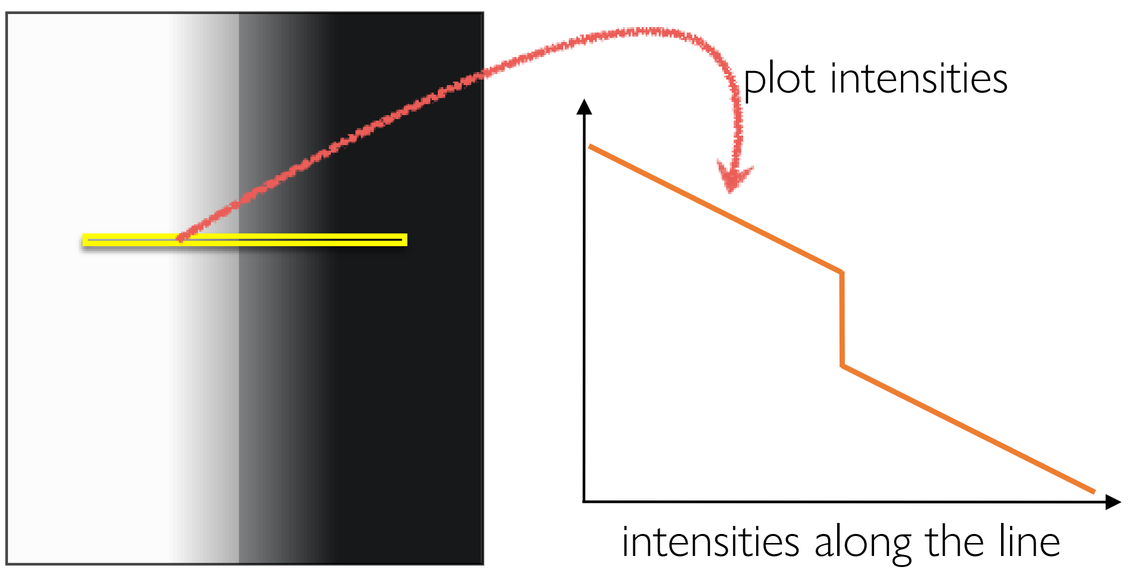


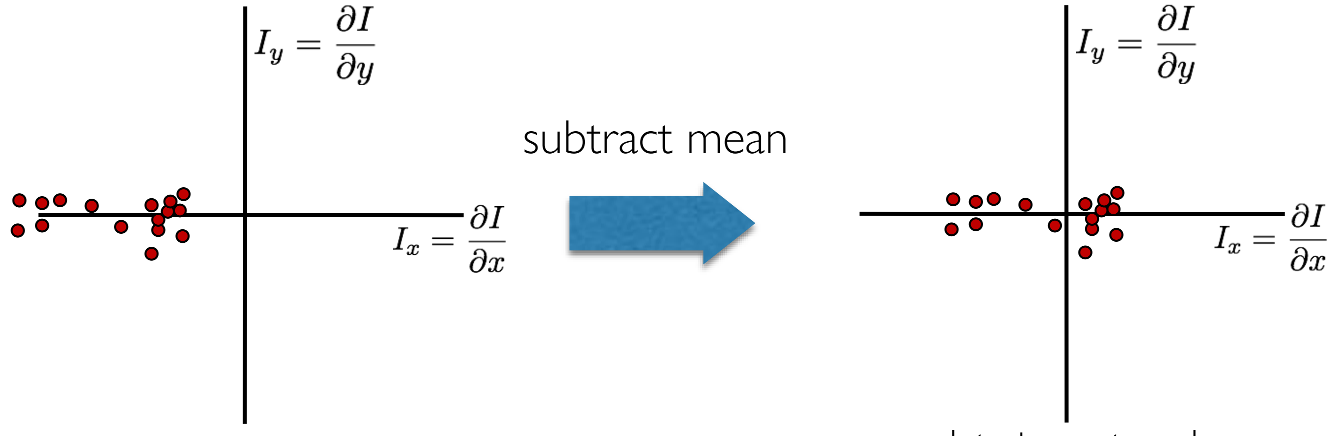


plot of image gradients

2. Subtract the mean from each image gradient

constant intensity gradient





plot of image gradients

data is centered
('DC' offset is removed)

3. Compute the covariance matrix

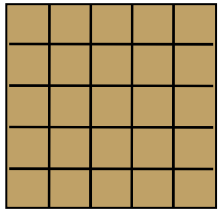
3. Compute the covariance matrix

$$\begin{bmatrix} \sum\limits_{p \in P} I_x I_x & \sum\limits_{p \in P} I_x I_y \\ \sum\limits_{p \in P} I_y I_x & \sum\limits_{p \in P} I_y I_y \end{bmatrix}$$

$I_x = \dfrac{\partial I}{\partial x}$

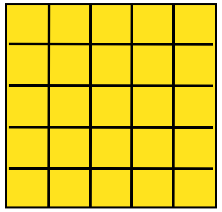
$I_y = \dfrac{\partial I}{\partial y}$

$\sum\limits_{p \in P} I_x I_y$ =sum(



array of x gradients

.*

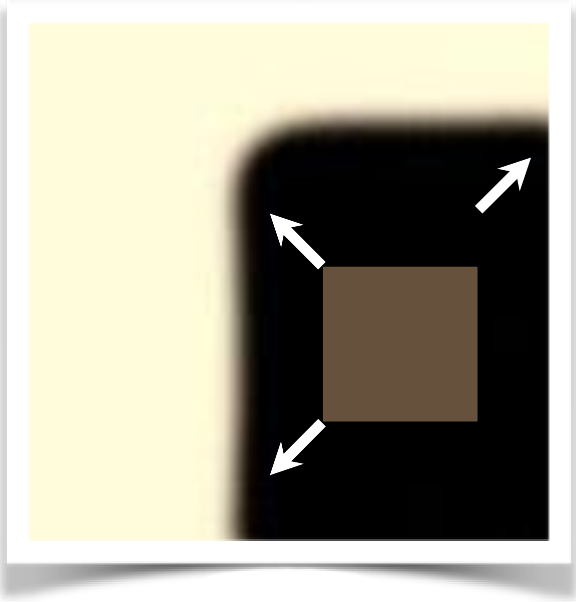


array of y gradients

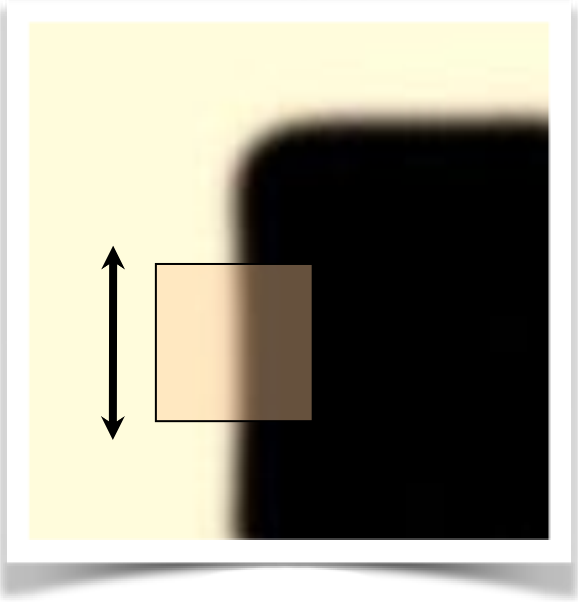
)

- Where does this covariance matrix come from?

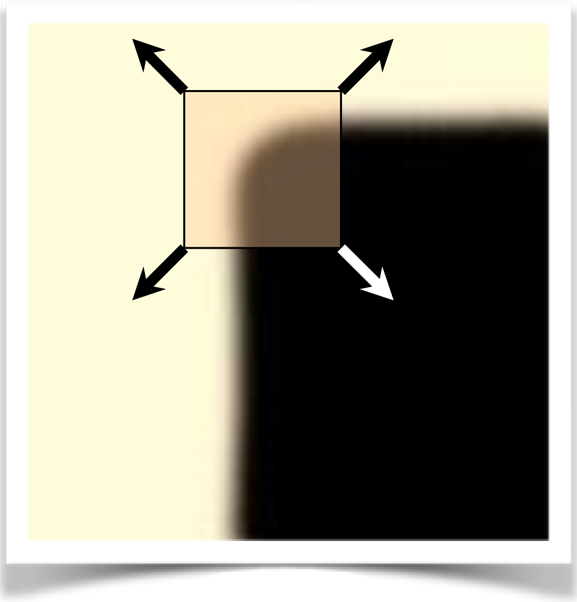
- Easily recognized by looking through a small window
- Shifting the window should give large change in intensity



"flat" region:
no change in all
directions



"edge":
no change along the edge
direction



"corner":
significant change in all
directions

[Moravec 1980]

Error function

- Some mathematical background…

Change of intensity for the shift [*u*,*v*]:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

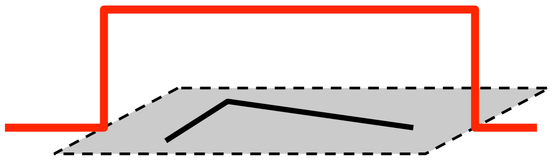
Error function

Window function

Shifted intensity

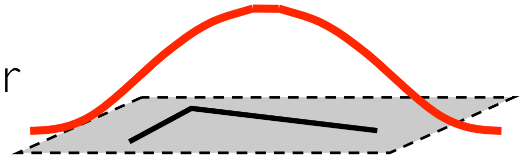
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Error function approximation

- First-order Taylor expansion of I(x,y) about (0,0)
  ((bilinear approximation for small shifts)

Change of intensity for the shift [*u*,*v*]:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Bilinear approximation

- For small shifts $[u,v]$ we have a 'bilinear approximation':

Change in appearance for a shift [u,v]

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

- where M is a 2×2 matrix computed from image derivatives:

'second moment' matrix
'structure tensor'

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- By computing the gradient covariance matrix…

$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

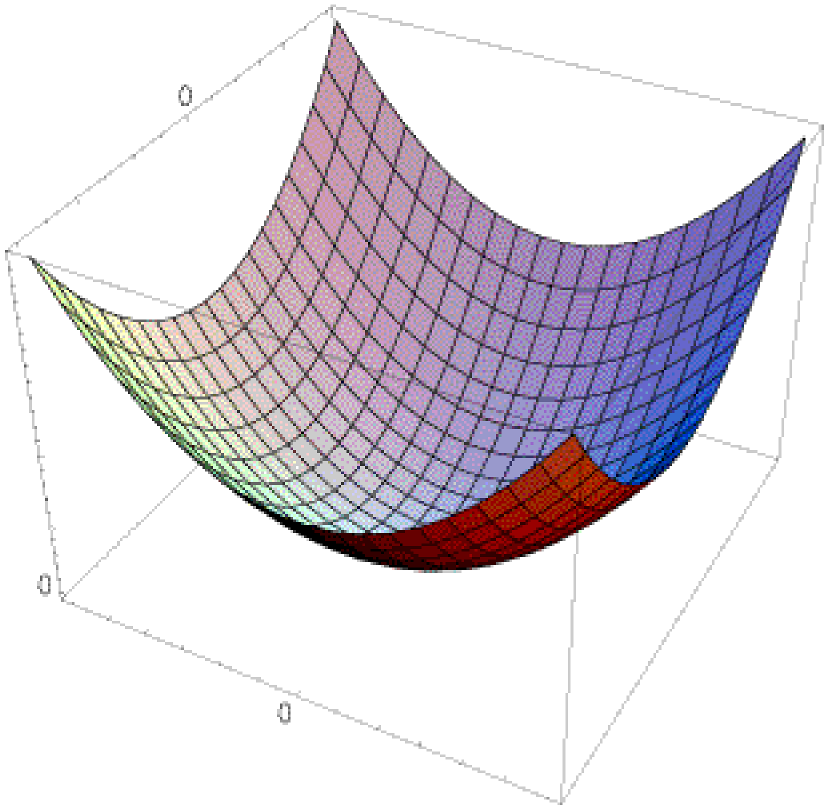
- We are fitting a quadratic to the gradients over a small image region

Visualization of a quadratic

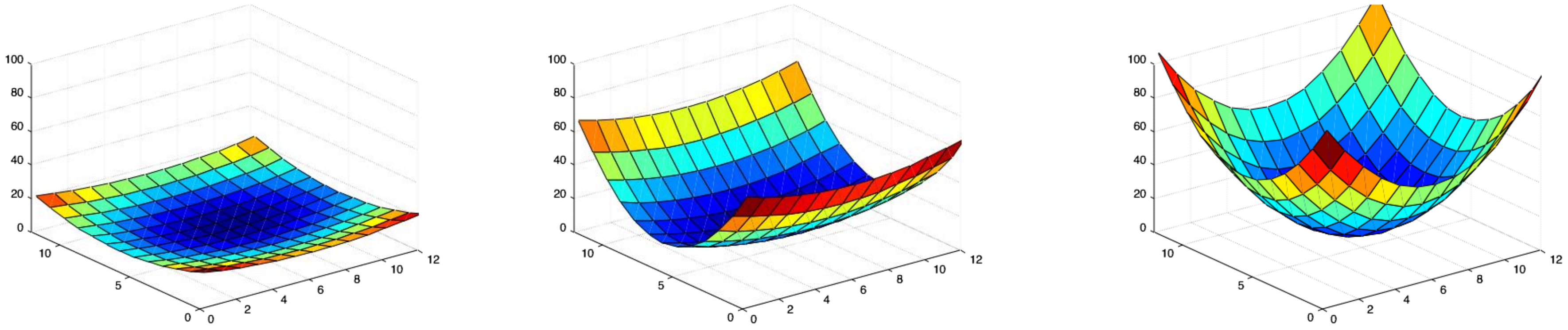
- The surface E(u,v) is locally approximated by a quadratic form

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

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

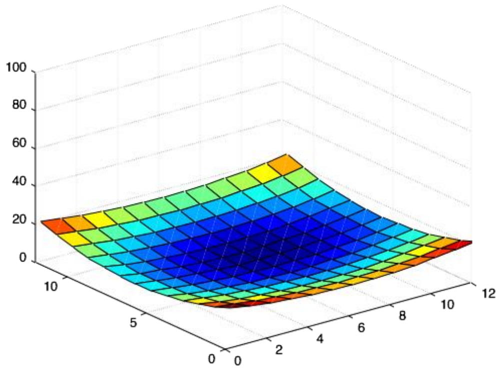


• Which error surface indicates a good image feature?

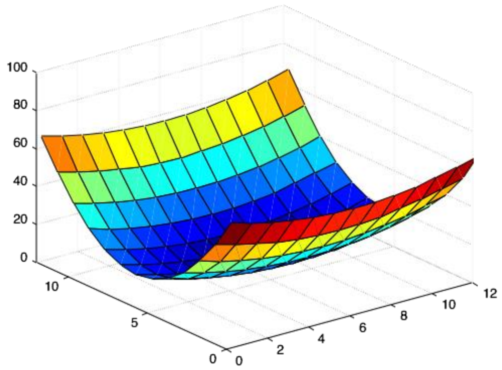


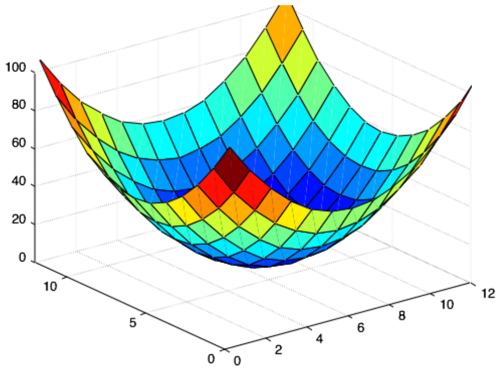
• What kind of image patch do these surfaces represent?

- Which error surface indicates a good image feature?

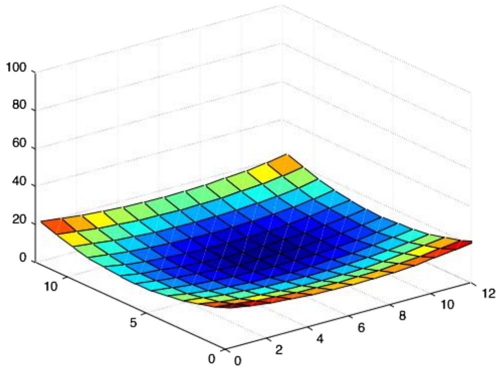


flat

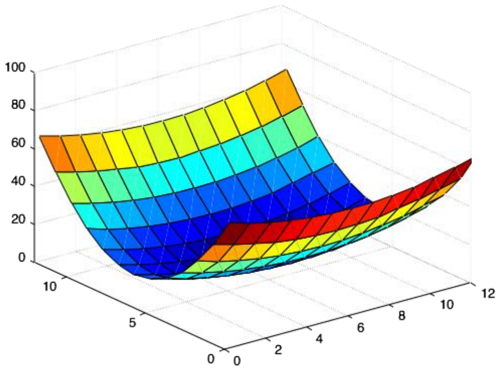




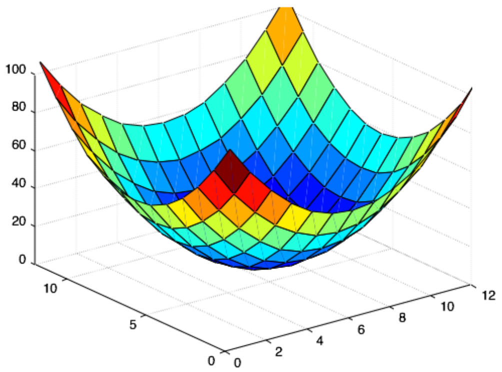
- Which error surface indicates a good image feature?



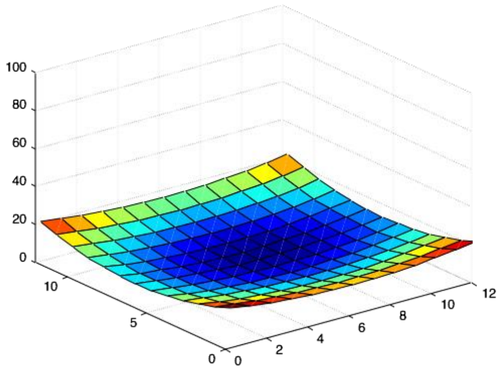
flat



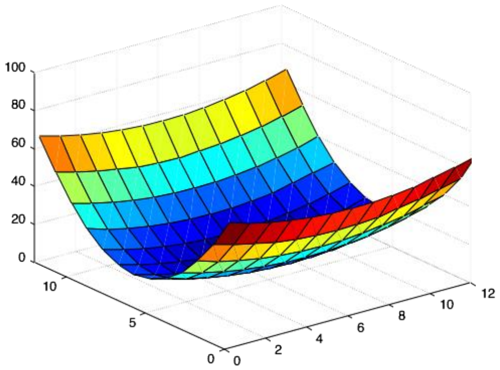
edge
'line'



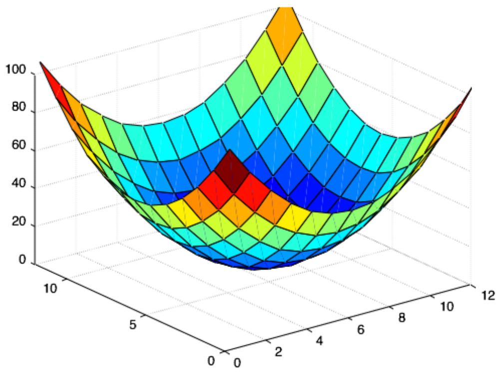
- Which error surface indicates a good image feature?



flat



edge
'line'



corner
'dot'

4. Compute eigenvalues and eigenvectors

4. Compute eigenvalues and eigenvectors

eigenvalue

$$\boldsymbol{M e} = \lambda \boldsymbol{e}$$

eigenvector

$$(M - \lambda I)\boldsymbol{e} = 0$$

4. Compute eigenvalues and eigenvectors

eigenvalue

$$Me = \lambda e$$

eigenvector

$$(M - \lambda I)e = 0$$

1. Compute the determinant of $M - \lambda I$

(returns a polynomial)

4. Compute eigenvalues and eigenvectors

eigenvalue

$$\boldsymbol{M}\boldsymbol{e} = \lambda\boldsymbol{e}$$

eigenvector

$$(M - \lambda I)\boldsymbol{e} = 0$$

1. Compute the determinant of $M - \lambda I$
(returns a polynomial)

2. Find the roots of polynomial $\det(M - \lambda I) = 0$
(returns eigenvalues)

4. Compute eigenvalues and eigenvectors

eigenvalue

$$Me = \lambda e$$

eigenvector

$$(M - \lambda I)e = 0$$

1. Compute the determinant of $M - \lambda I$

(returns a polynomial)

2. Find the roots of polynomial $\det(M - \lambda I) = 0$

(returns eigenvalues)

3. For each eigenvalue, solve $(M - \lambda I)e = 0$

(returns eigenvectors)

$eig(M)$

Visualization as an ellipse

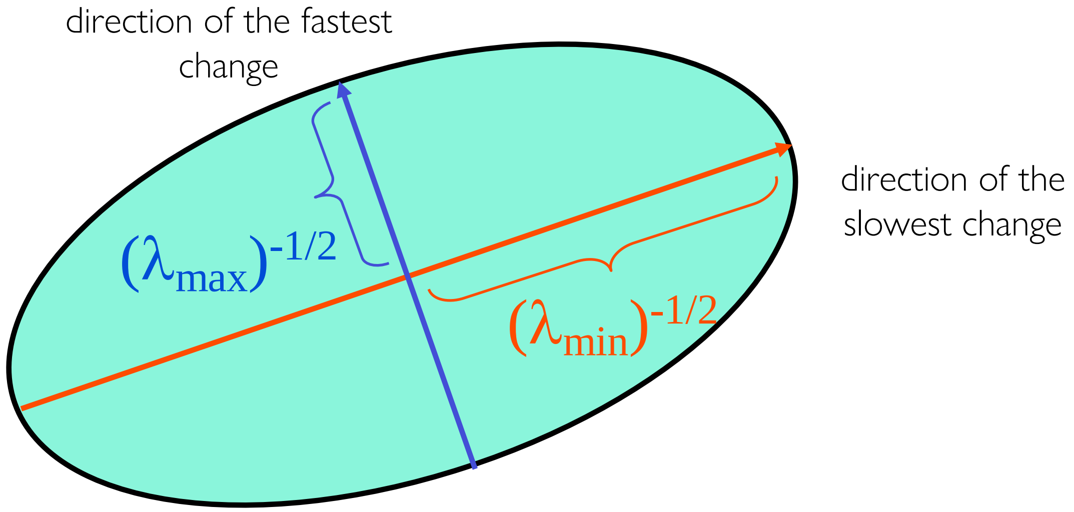
- Since M is symmetric, we have

$$M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

- We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*

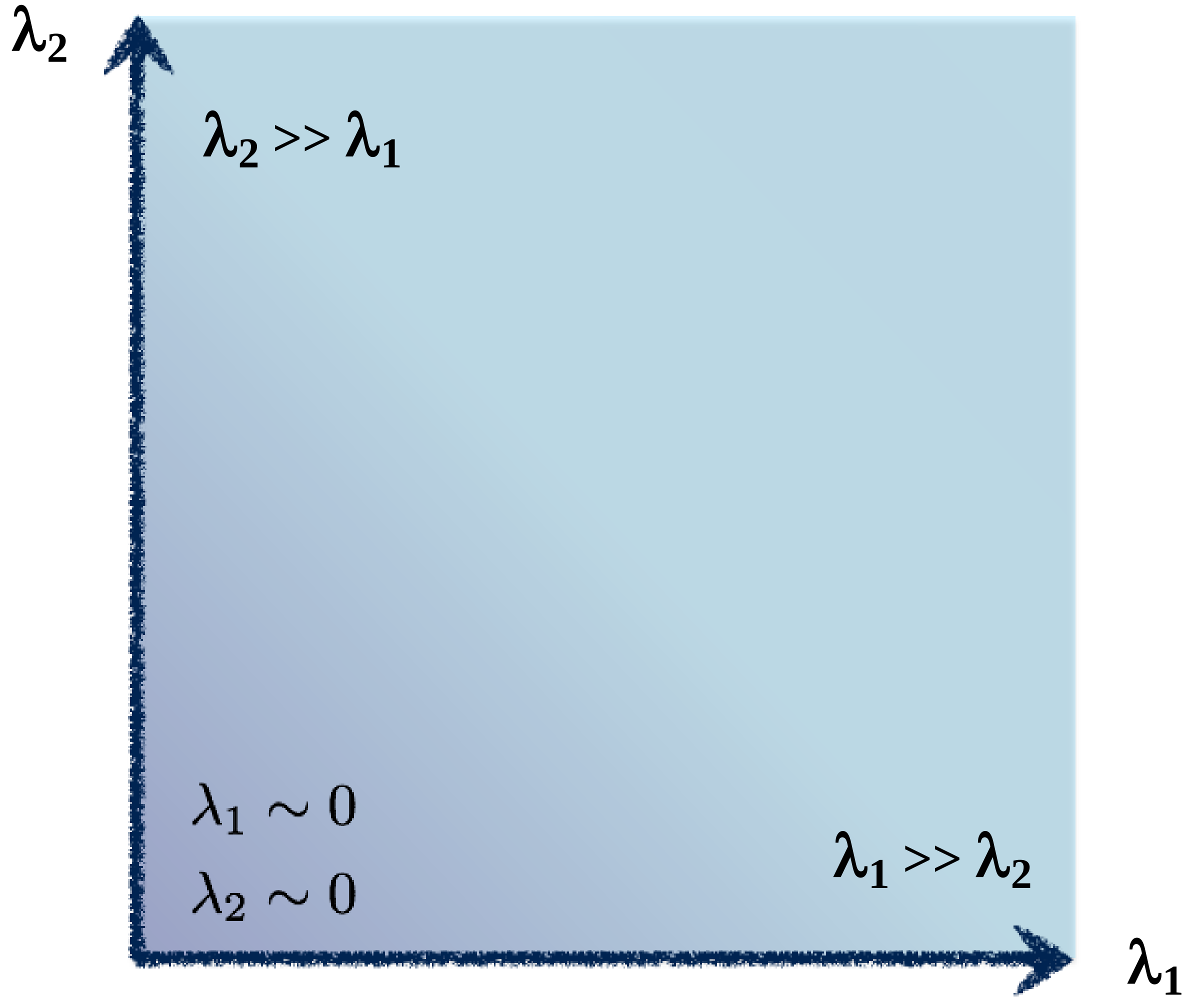
Ellipse equation:

$$[u \;\; v] \;\; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$



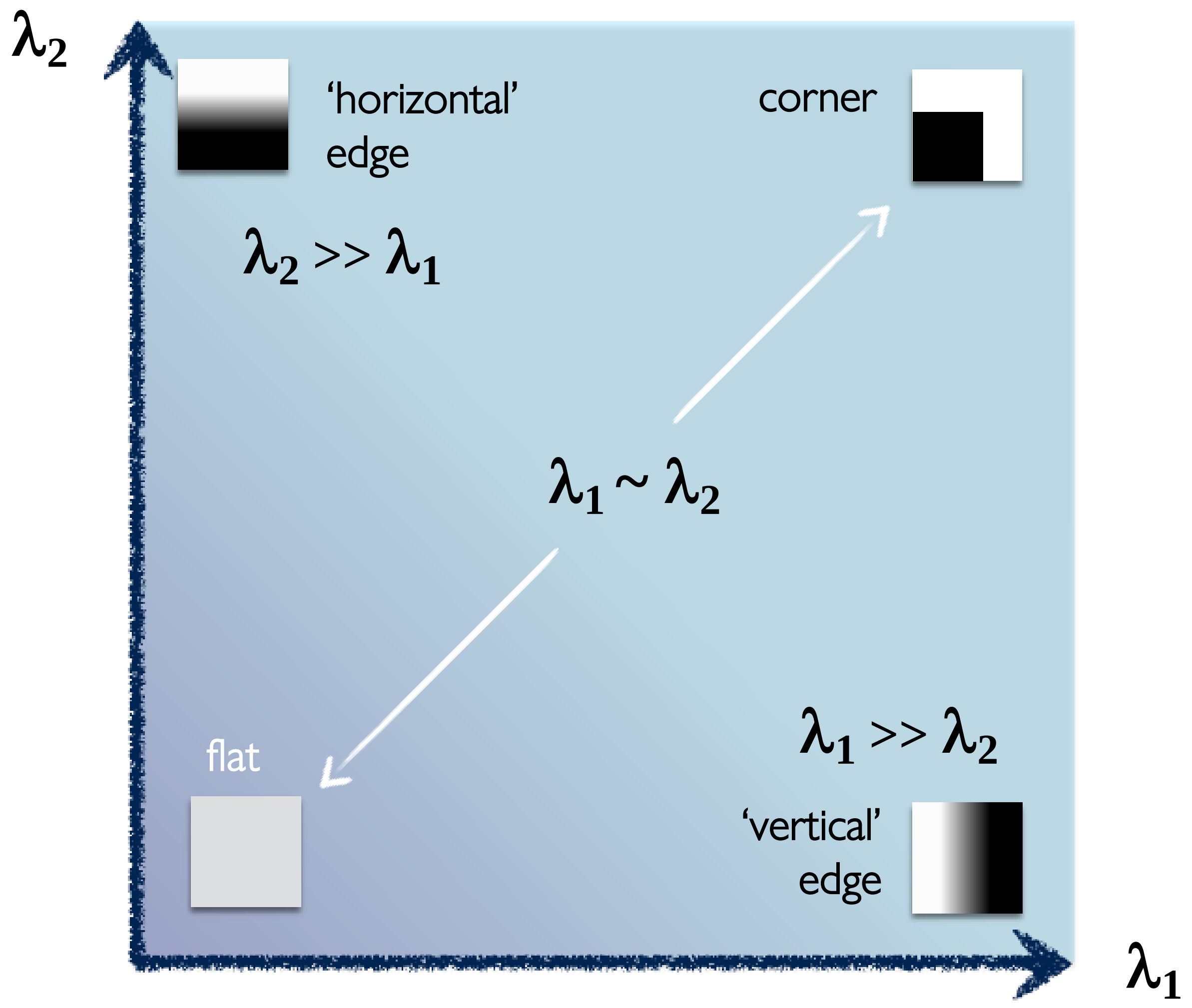
Interpreting eigenvalues

- What kind of image patch does each region represent?



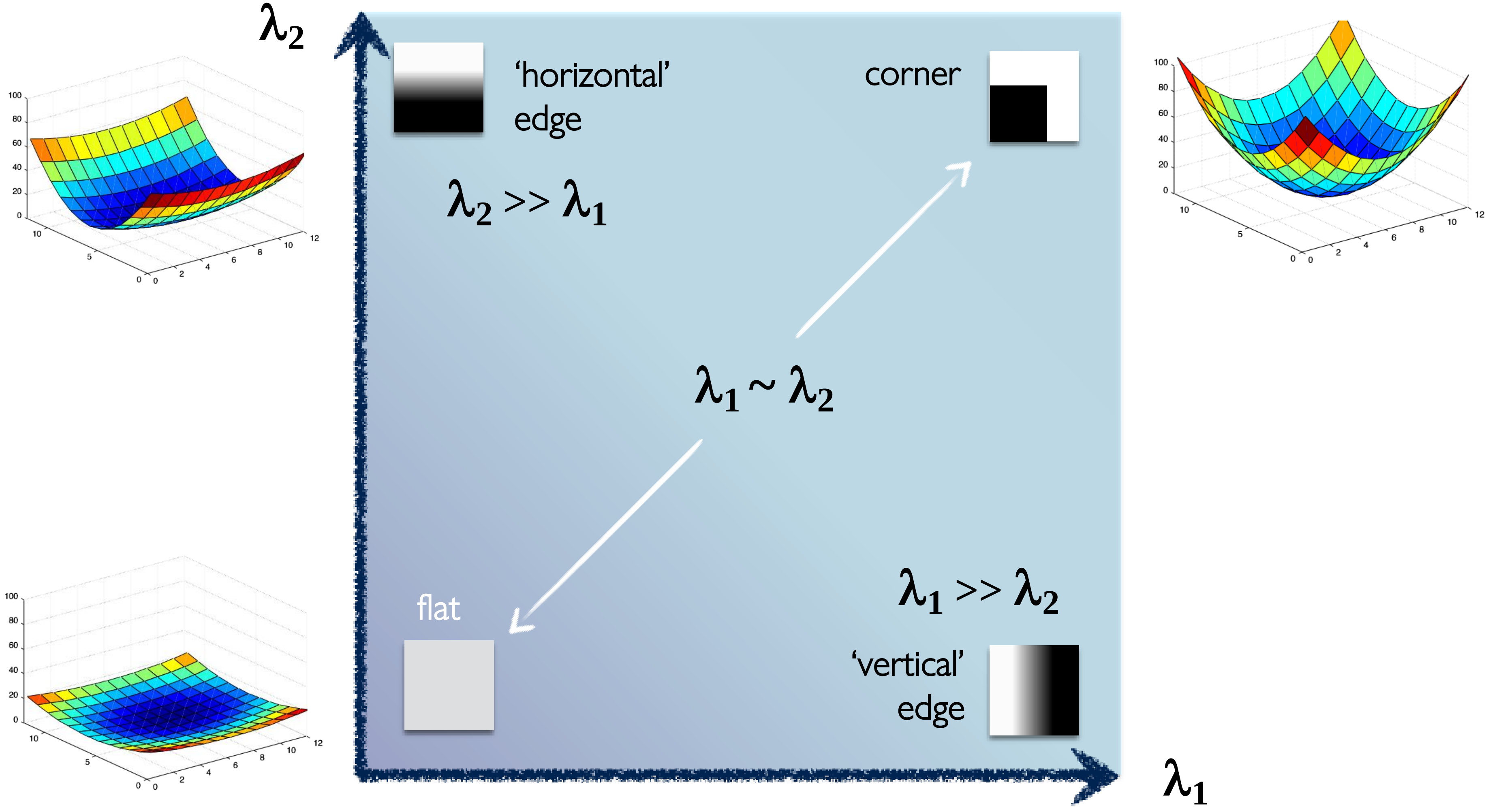
Interpreting eigenvalues

- What kind of image patch does each region represent?



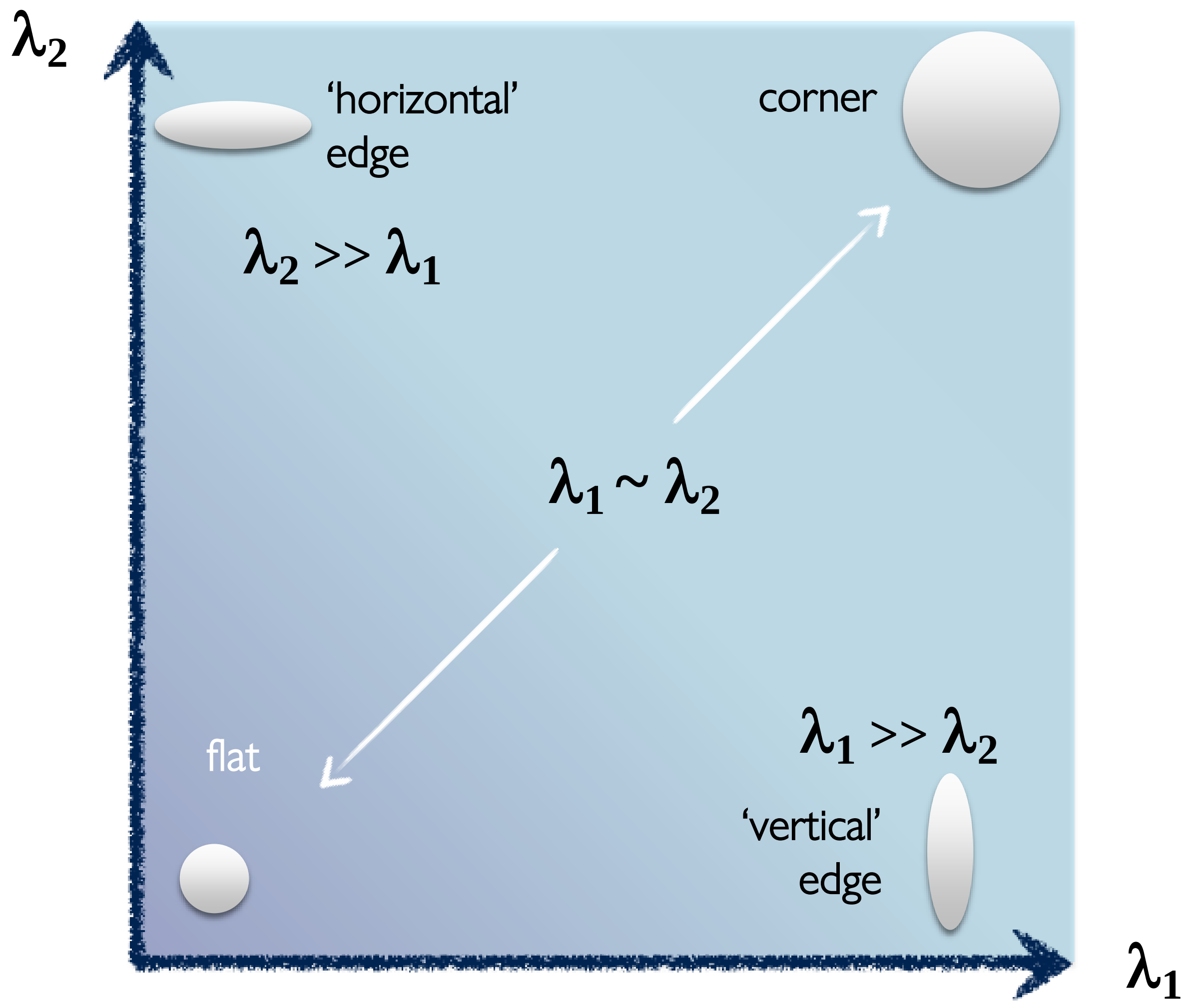
Interpreting eigenvalues

- What kind of image patch does each region represent?



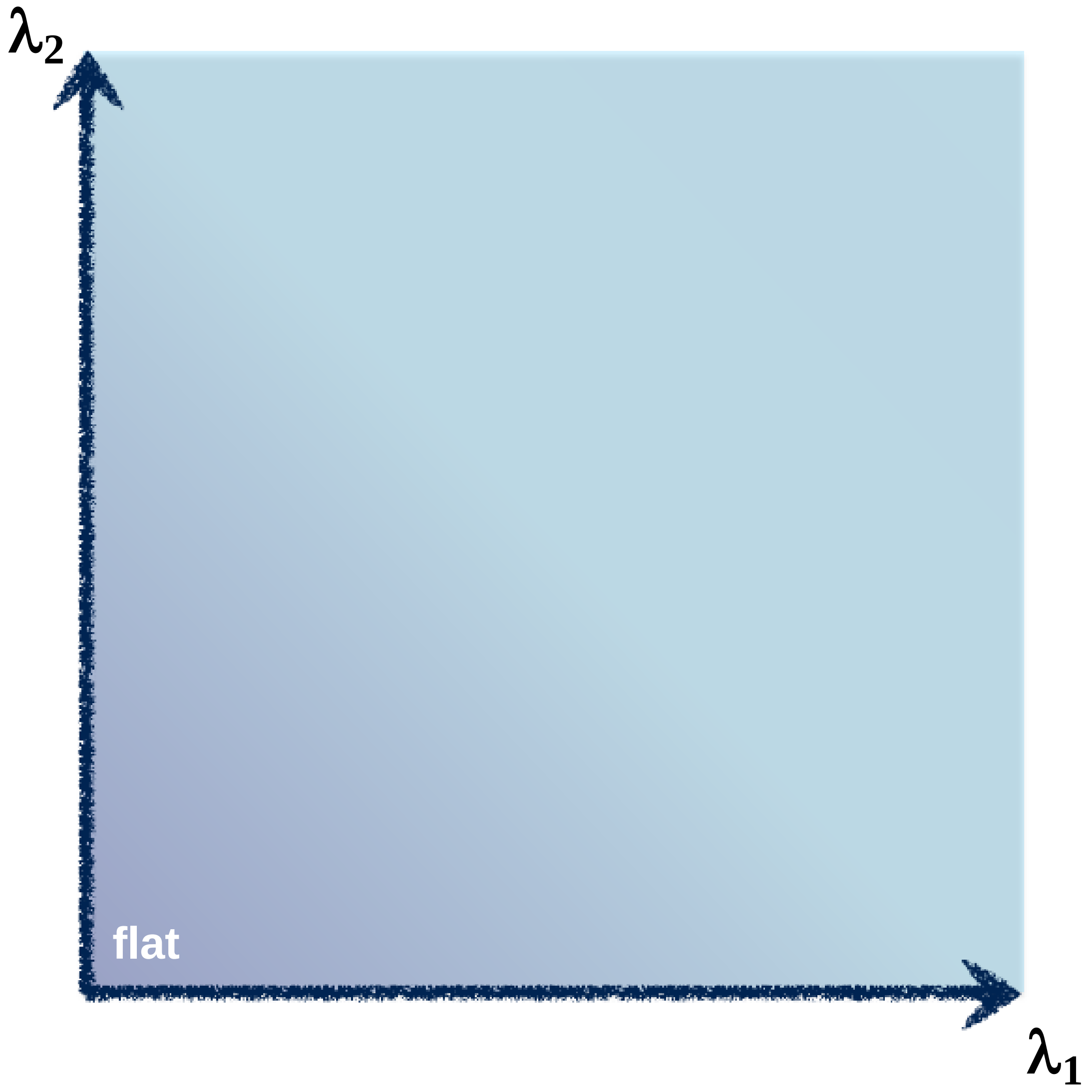
Interpreting eigenvalues

- What kind of image patch does each region represent?



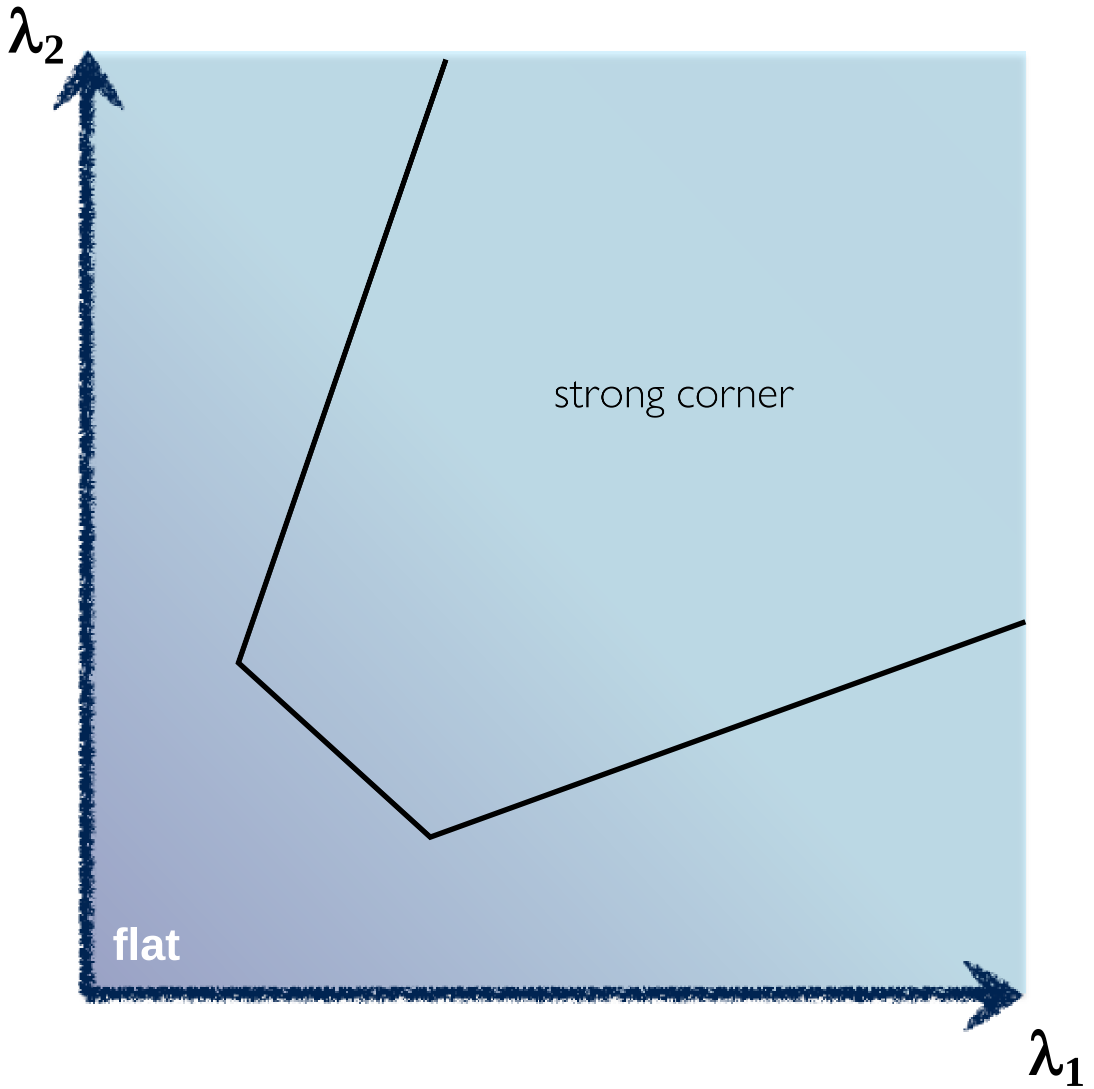
5. Use threshold on eigenvalues to detect corners

5. Use threshold on eigenvalues to detect corners



Think of a function to score 'cornerness'

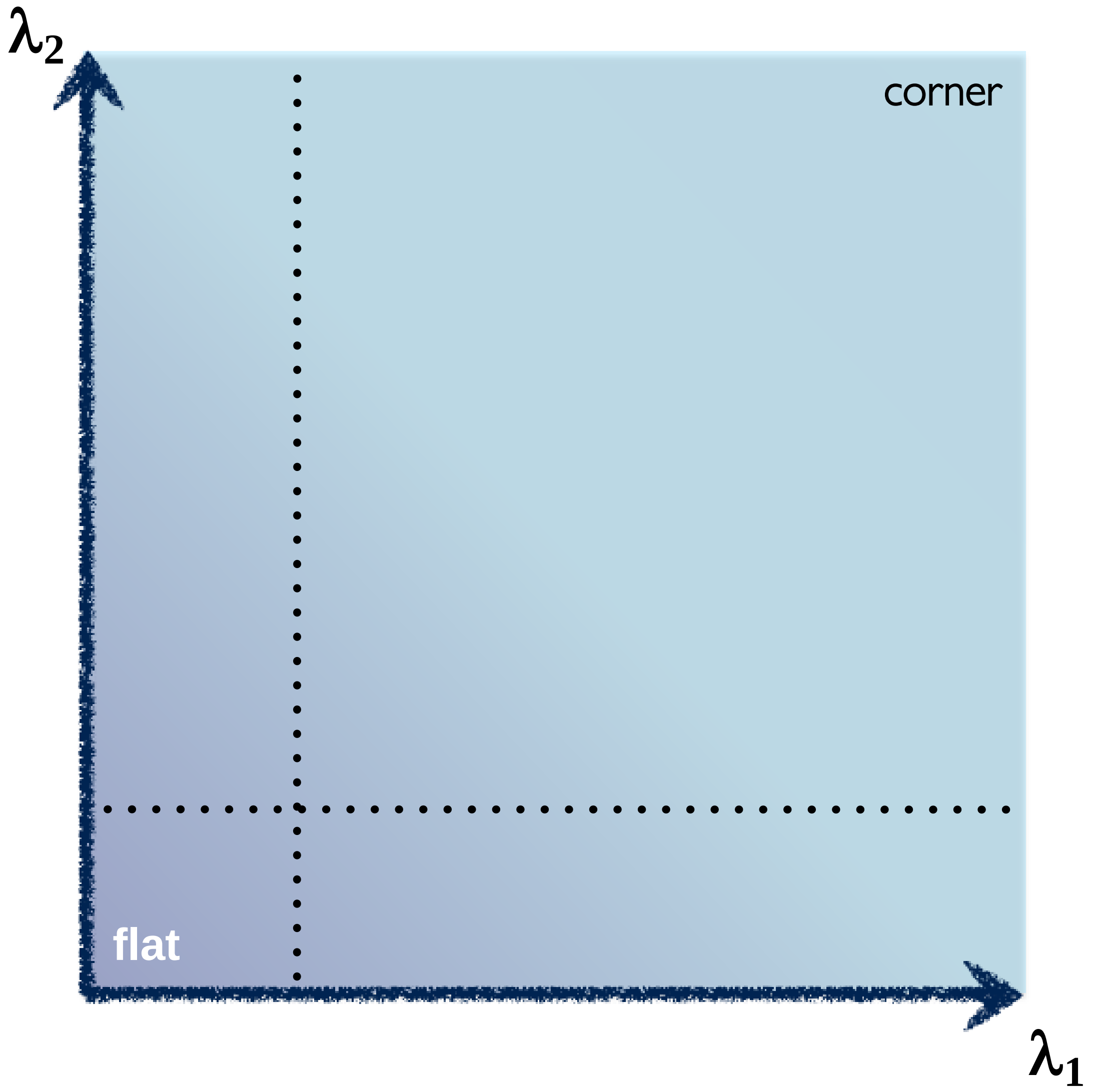
5. Use threshold on eigenvalues to detect corners



Think of a function to score 'cornerness'

5. Use threshold on eigenvalues to detect corners

(a function of)

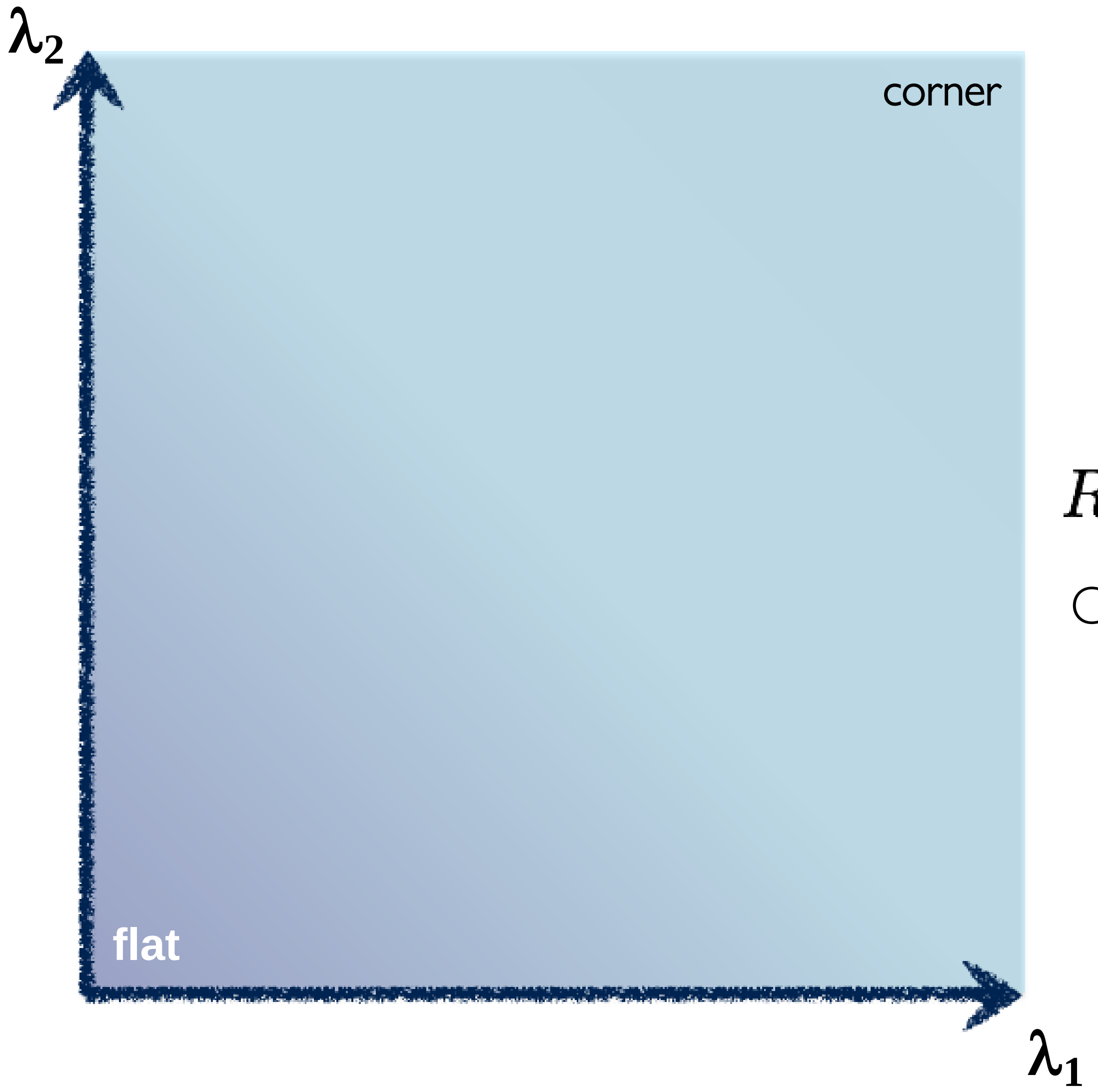


Use the smallest eigenvalue
as the response function

$$R = \min(\lambda_1, \lambda_2)$$

5. Use threshold on eigenvalues to detect corners

^
(a function of)



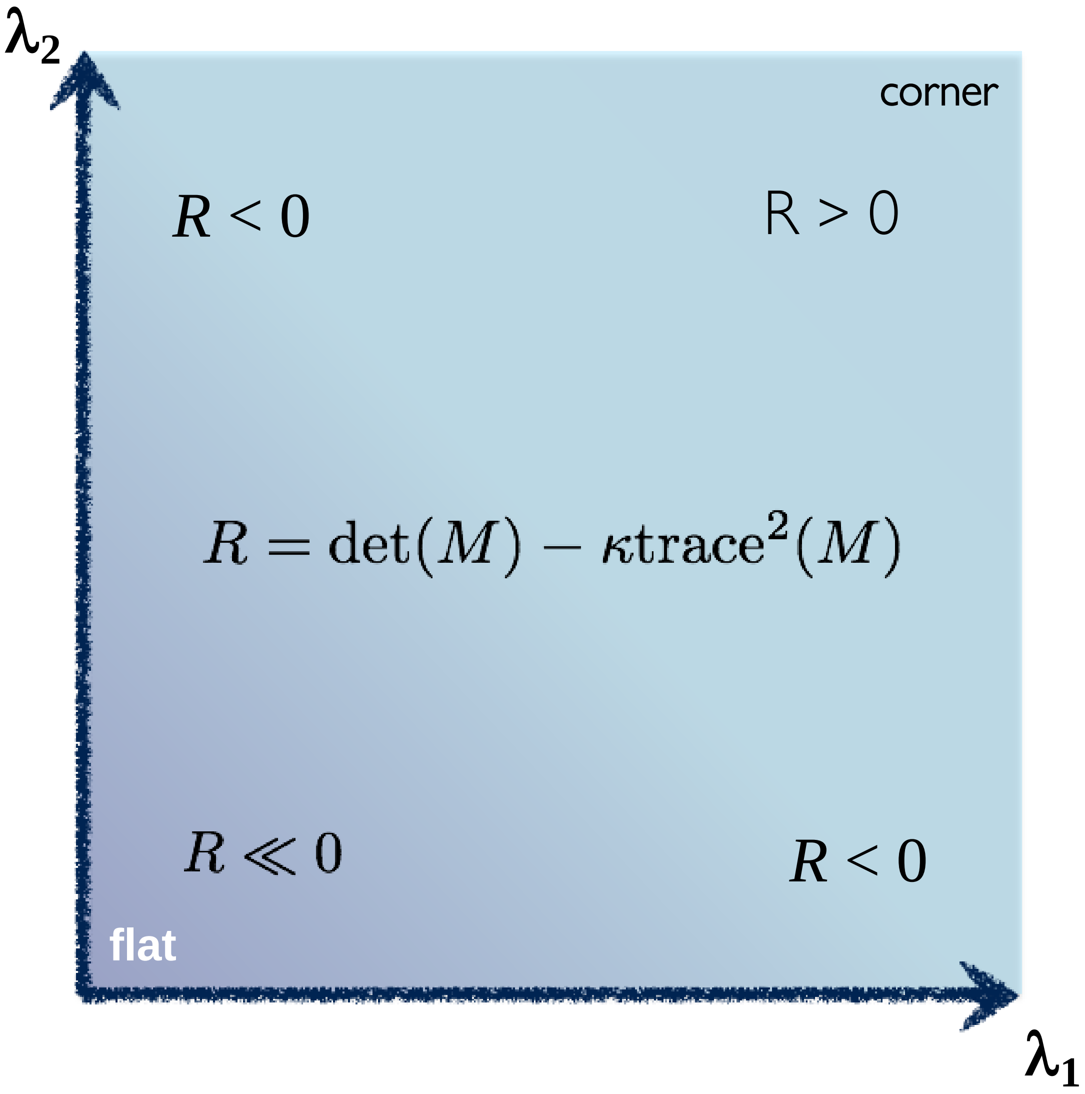
Eigenvalues need to be bigger than one.

$$R = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$$

Can compute this more efficiently…

5. Use threshold on eigenvalues to detect corners

^
(a function of)



$\det M = \lambda_1 \lambda_2$

$\operatorname{trace} M = \lambda_1 + \lambda_2$

$$det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

$$trace\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$$

- Harris & Stephens (1988)

$$R = \det(M) - \kappa \operatorname{trace}^2(M)$$

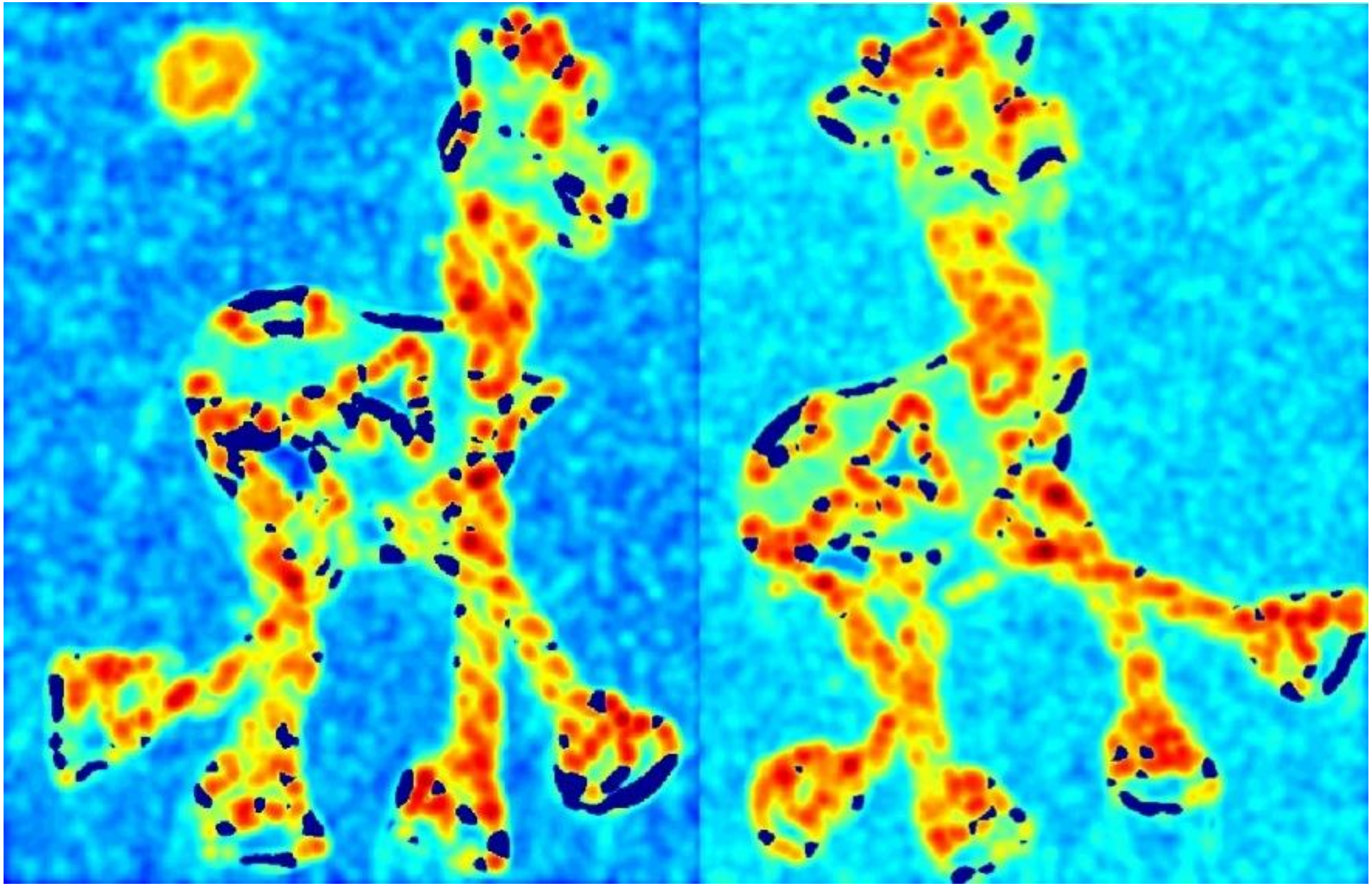
- Kanade & Tomasi (1994)

$$R = \min(\lambda_1, \lambda_2)$$

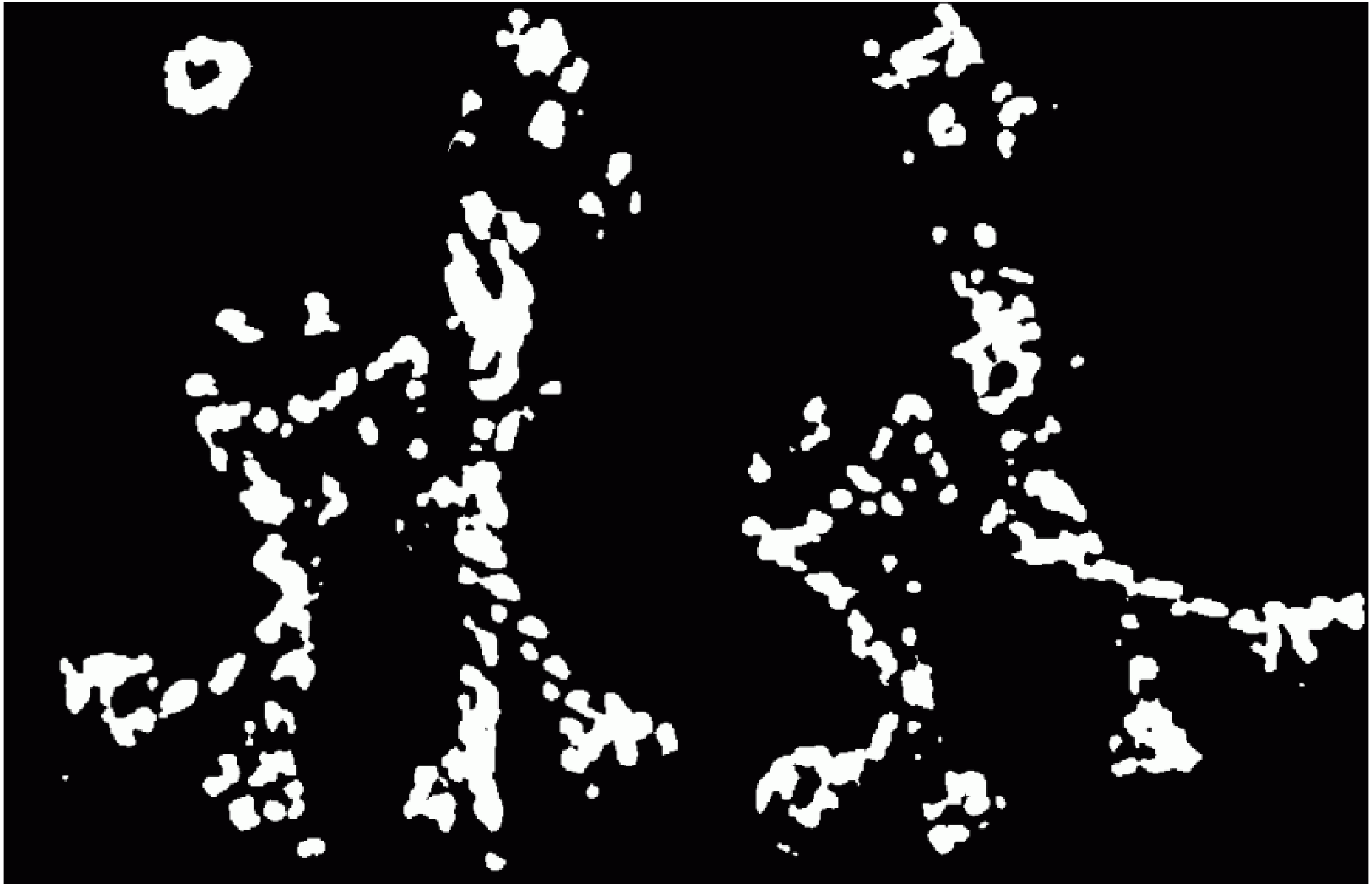
- Nobel (1998)

$$R = \frac{\det(M)}{\operatorname{trace}(M) + \epsilon}$$

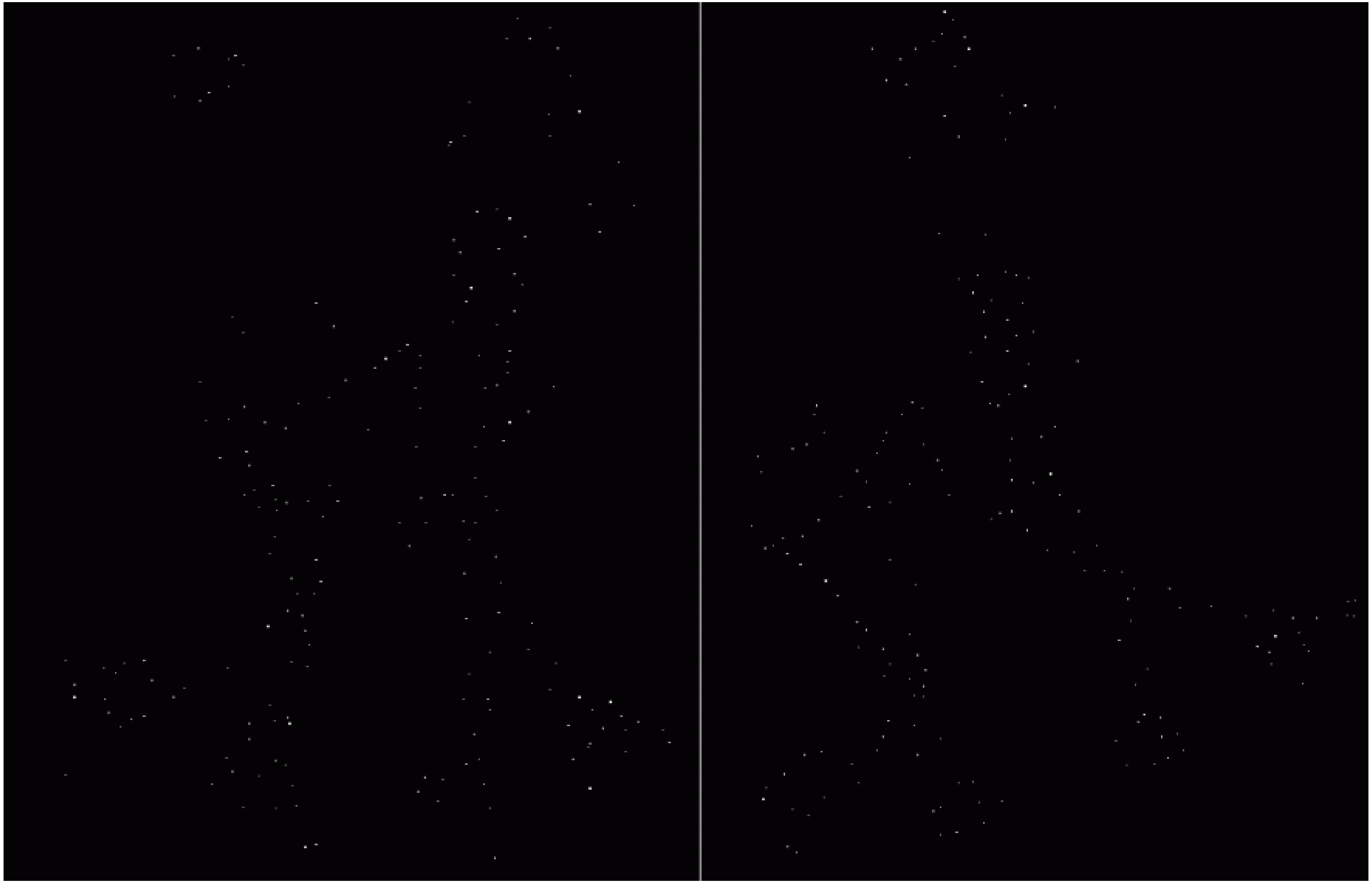

Corner response



Thresholded corner response

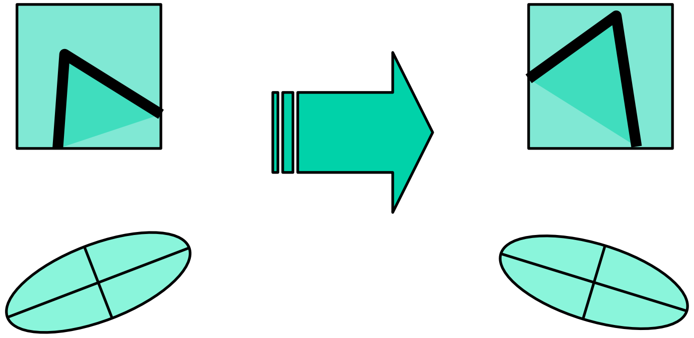


Non-maximal suppression



Harris corner response is invariant to rotation

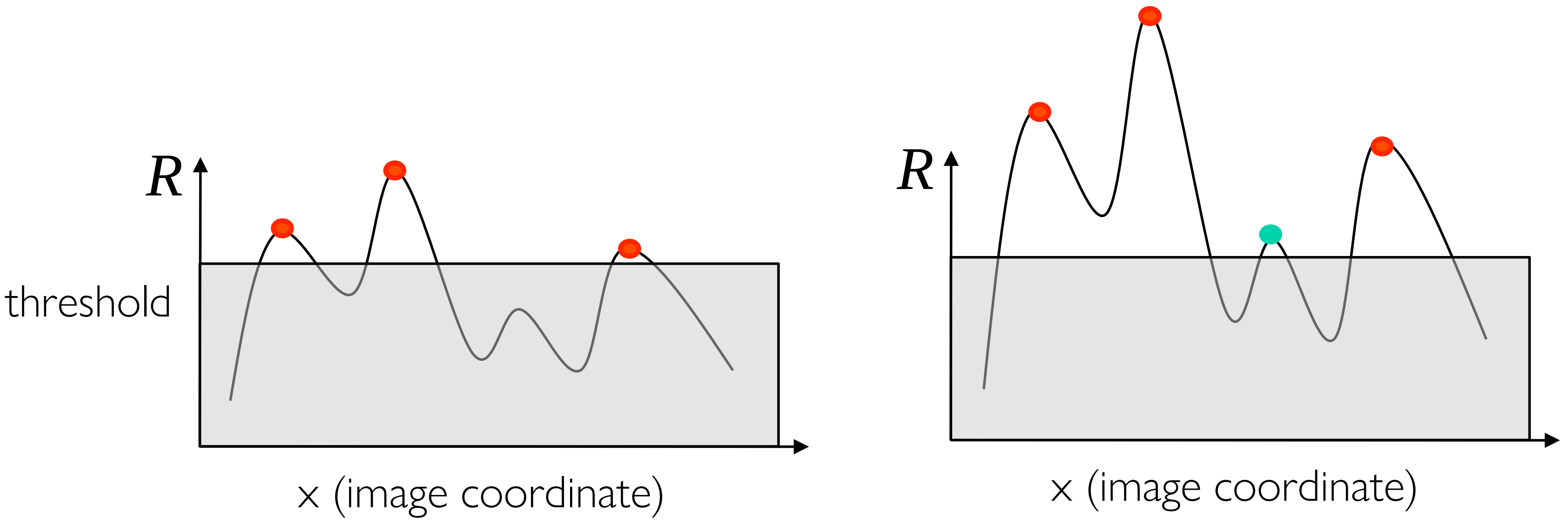
- Ellipse rotates but its shape (eigenvalues) remains the same



- Corner response R is invariant to image rotation

Harris corner response is invariant to intensity changes

- Partial invariance to **affine intensity** change
  - Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
  - Intensity scale : $I \rightarrow a\, I$



- The Harris detector is not invariant to changes in …

- The Harris detector is not invariant to changes in …

- The Harris corner detector is not invariant to scale