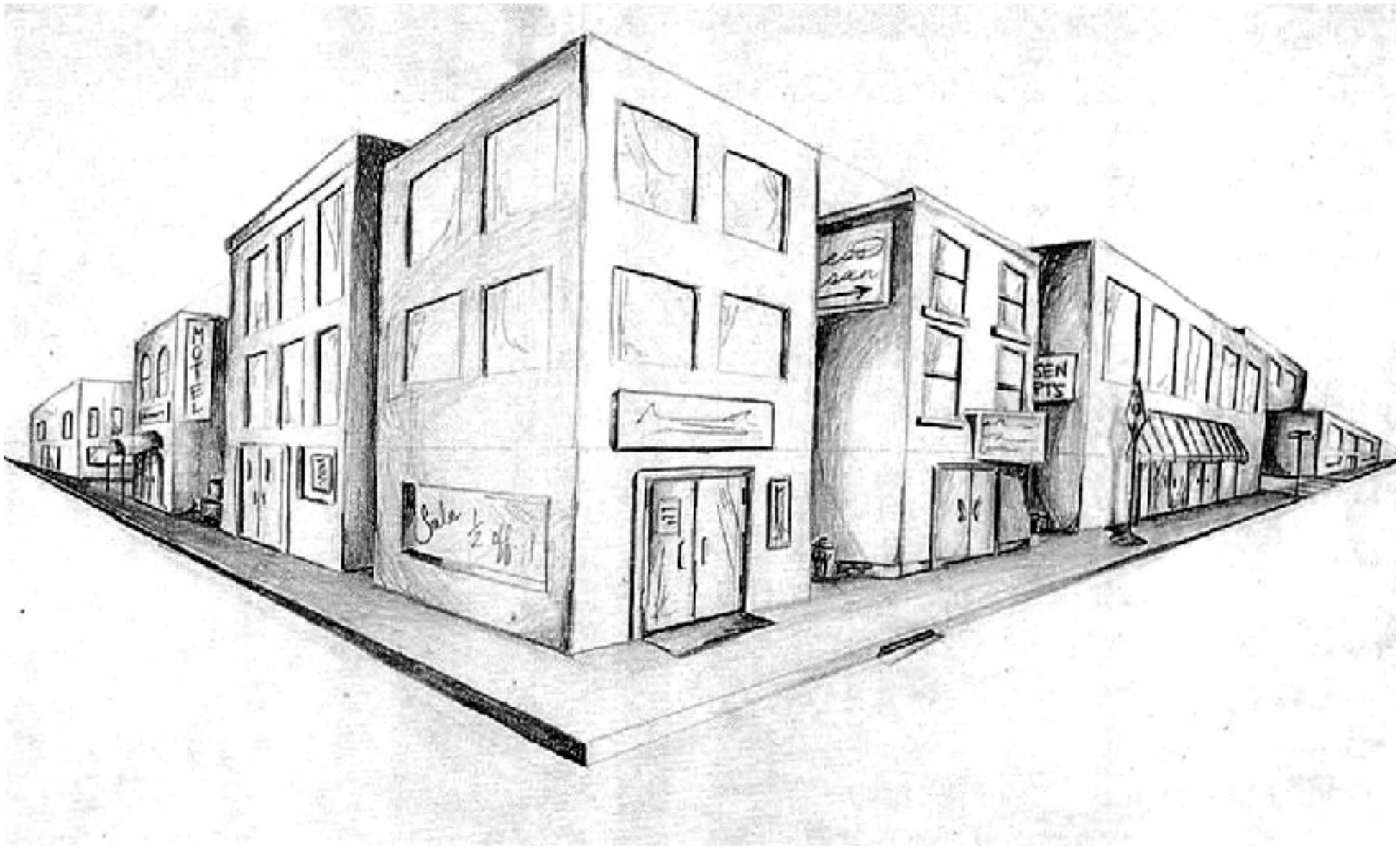


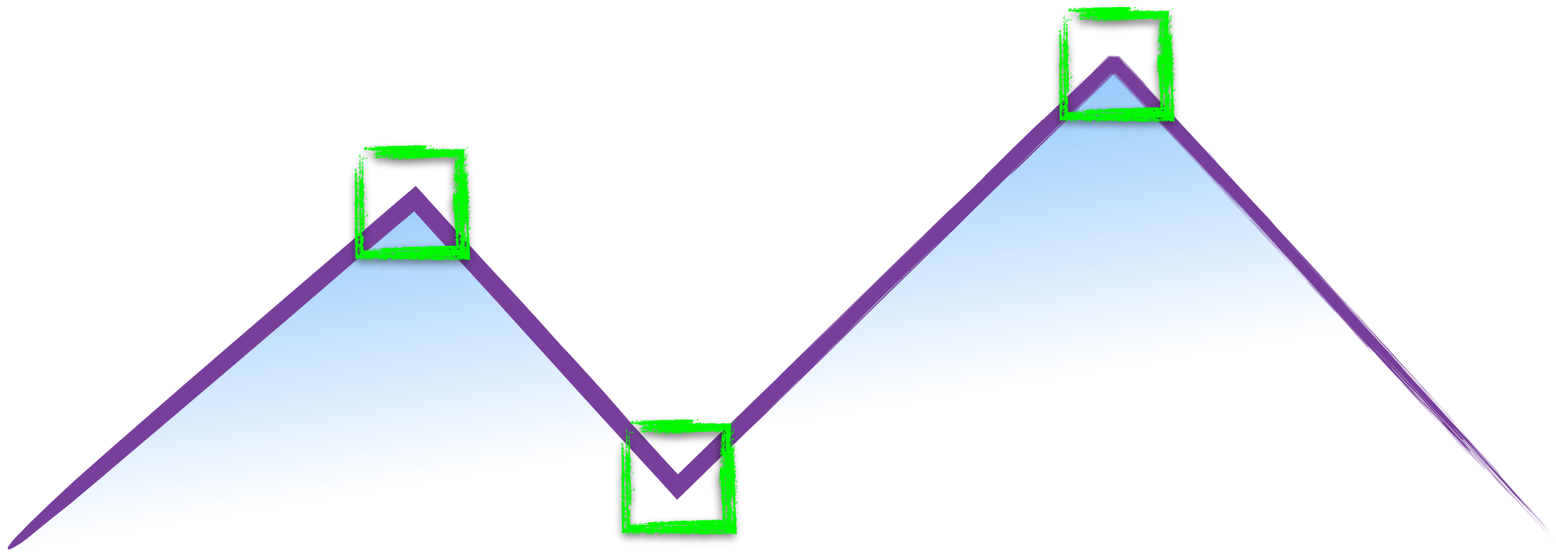

# Harris Corners

Computer Vision

**Carnegie Mellon University (Kris Kitani)**

# How do you find a corner?

[Moravec 1980]



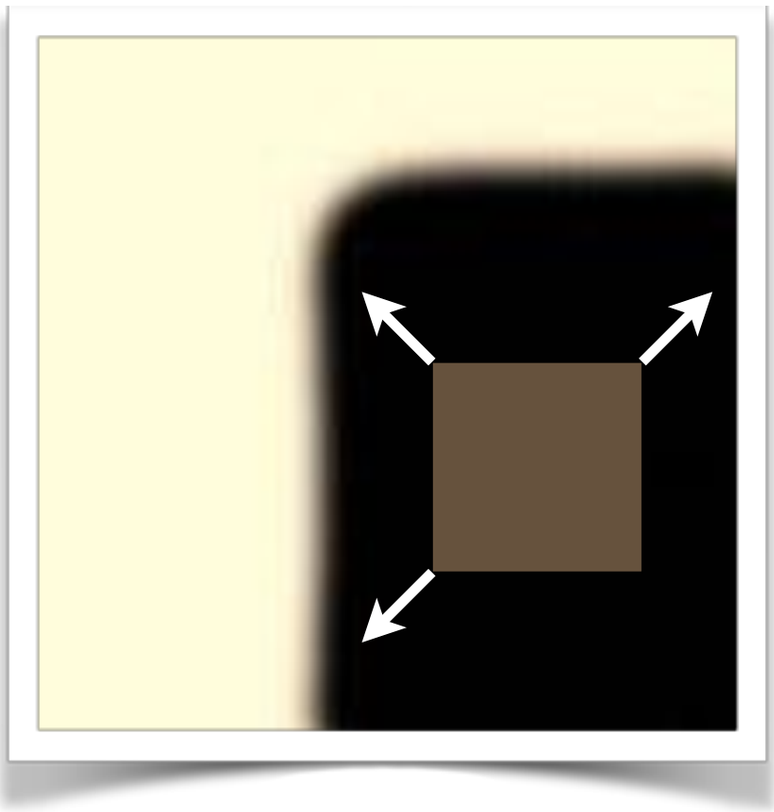
Easily recognized by looking through a small window

Shifting the window should give large change in intensity

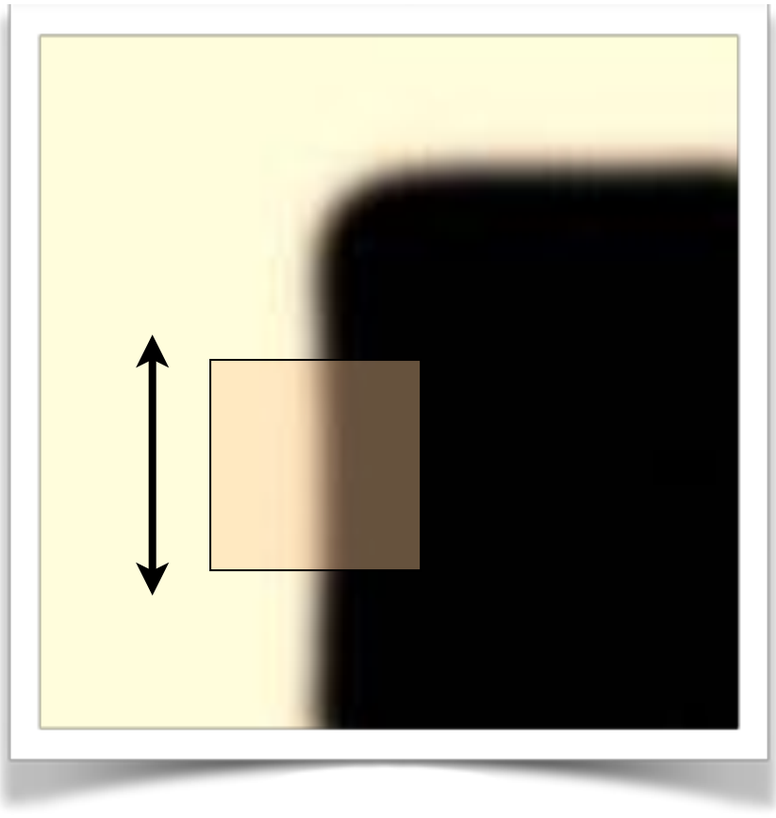
[Moravec 1980]

Easily recognized by looking through a small window

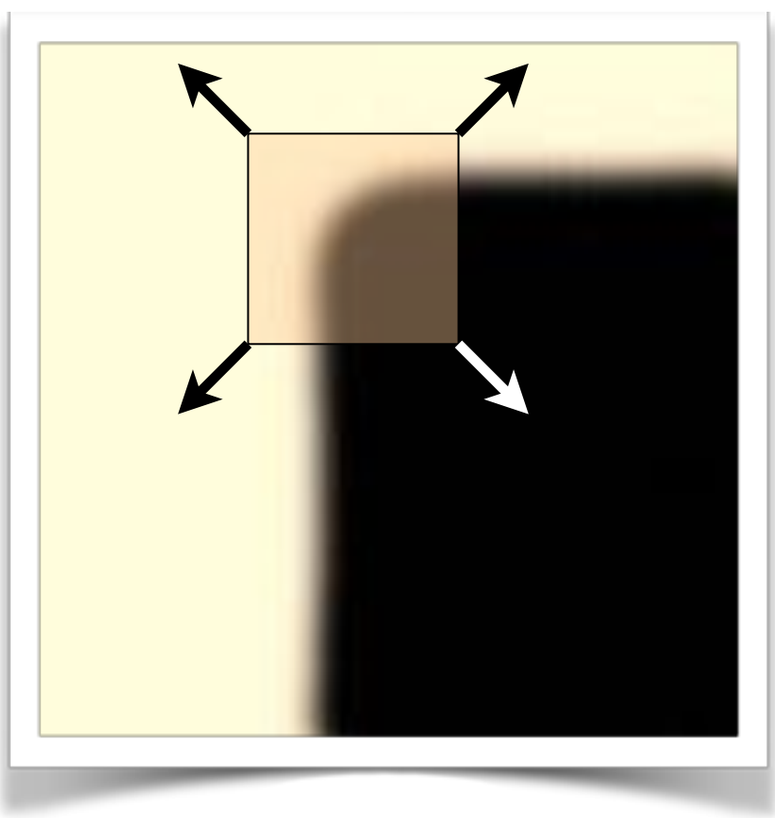
Shifting the window should give large change in intensity



"flat" region:
no change in all directions



"edge":
no change along the edge direction



"corner":
significant change in all directions

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

# A COMBINED CORNER AND EDGE DETECTOR

Chris Harris & Mike Stephens

Plessey Research Roke Manor, United Kingdom


*Consistency of image edge filtering is of prime importance for 3D interpretation of image sequences using feature tracking algorithms. To cater for image regions containing texture and isolated features, a combined corner and edge detector based on the local auto-correlation function is utilised, and it is shown to perform with good consistency on natural imagery.*

they are discrete, reliable and meaningful[2]. However, the lack of connectivity of feature-points is a major limitation in our obtaining higher level descriptions, such as surfaces and objects. We need the richer information that is available from edges[3].

## THE EDGE TRACKING PROBLEM

# Harris Corner Detection

1. Compute image gradients over small region

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

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

2. Compute the covariance matrix

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

3. Compute eigenvectors and eigenvalues

4. Use threshold on eigenvalues to detect corners

# Harris Corner Detection

**1. Compute image gradients over small region**

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

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

2. Compute the covariance matrix

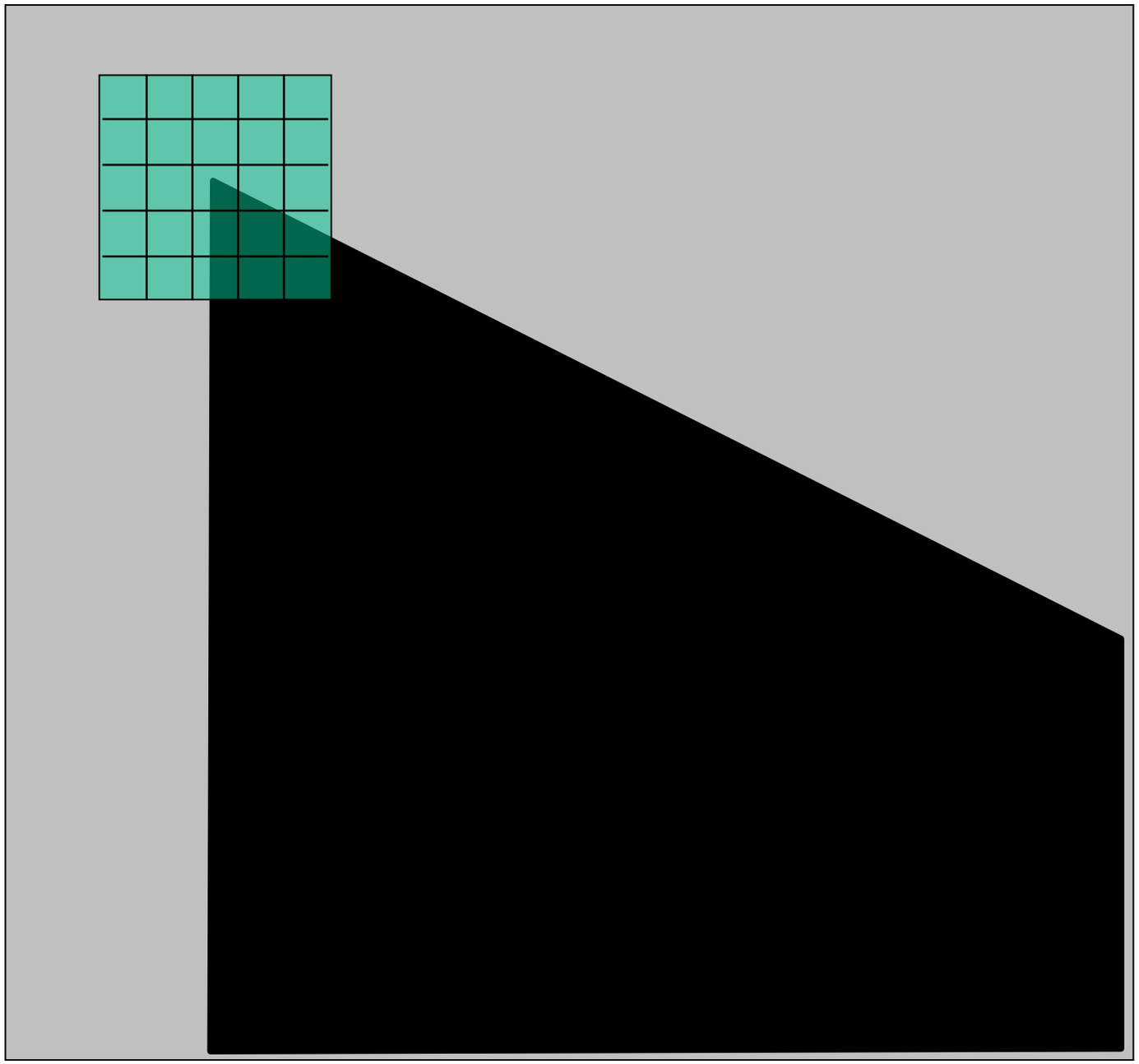
3. Compute eigenvectors and eigenvalues

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

4. Use threshold on eigenvalues to detect corners

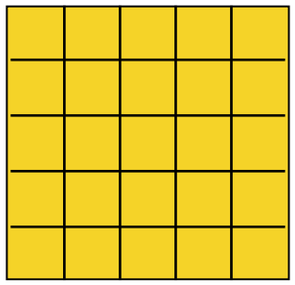
# 1. Compute image gradients over a small region

(not just a single pixel)



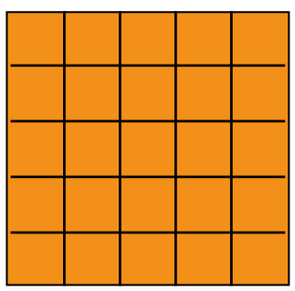
array of x gradients

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



array of y gradients

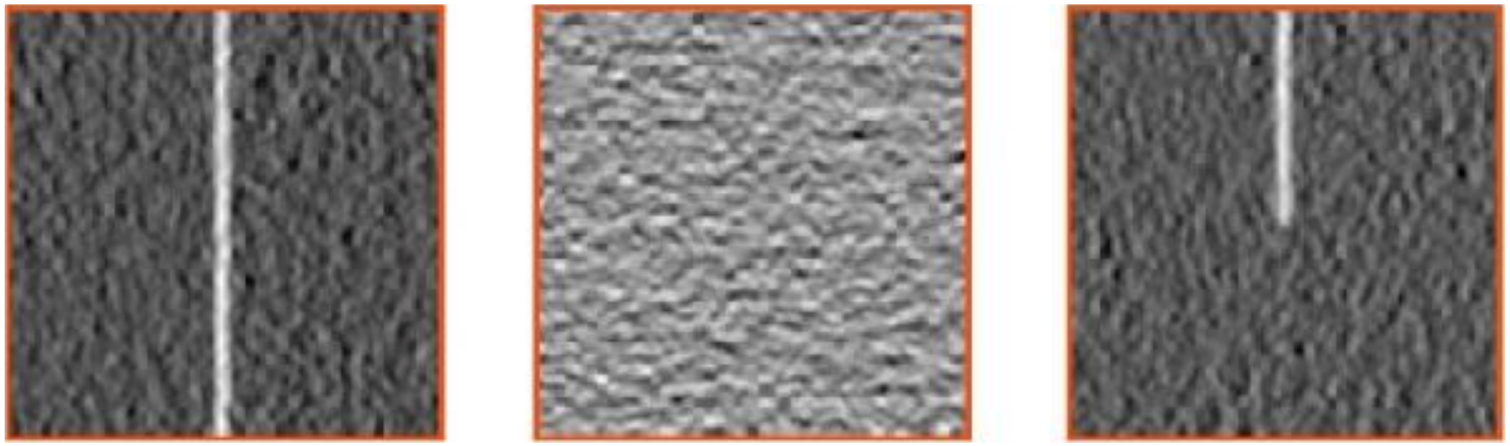
$$I_y = \frac{\partial I}{\partial y}$$



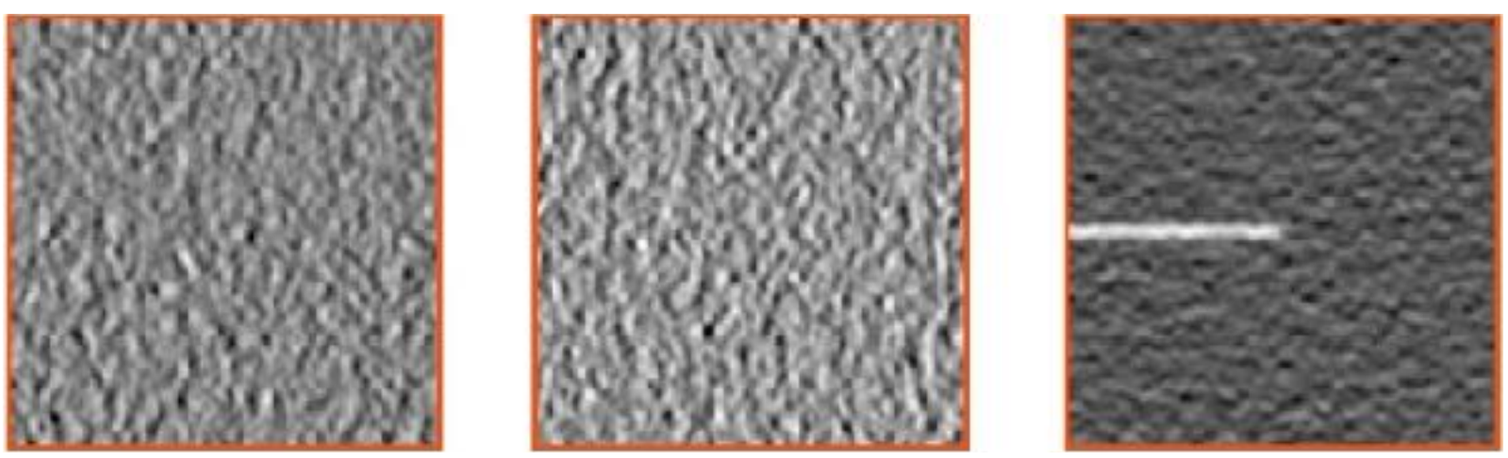
# visualization of gradients

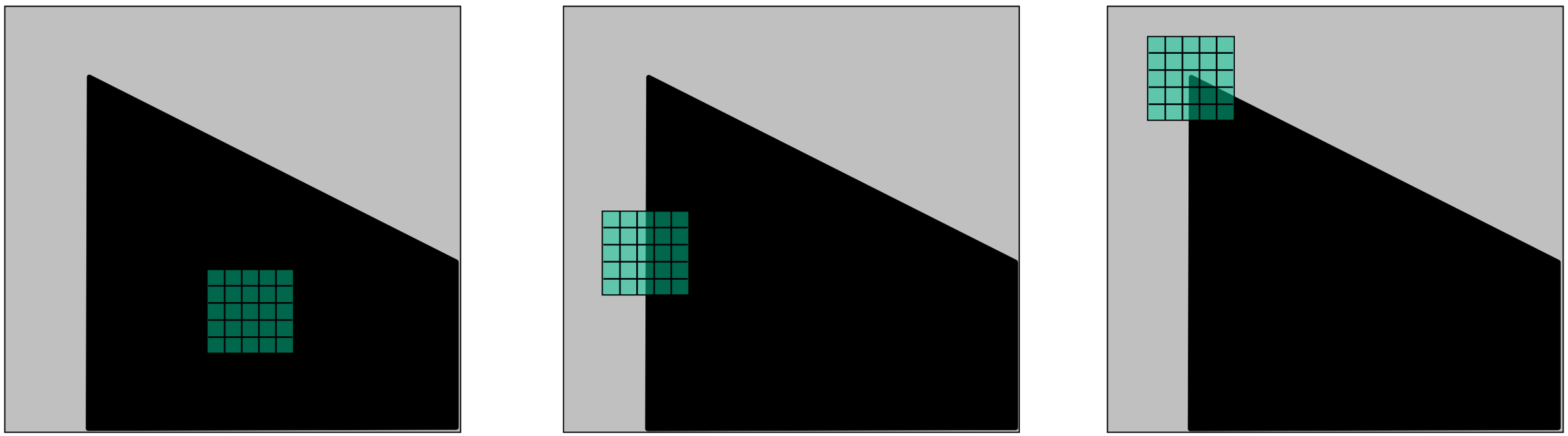
image

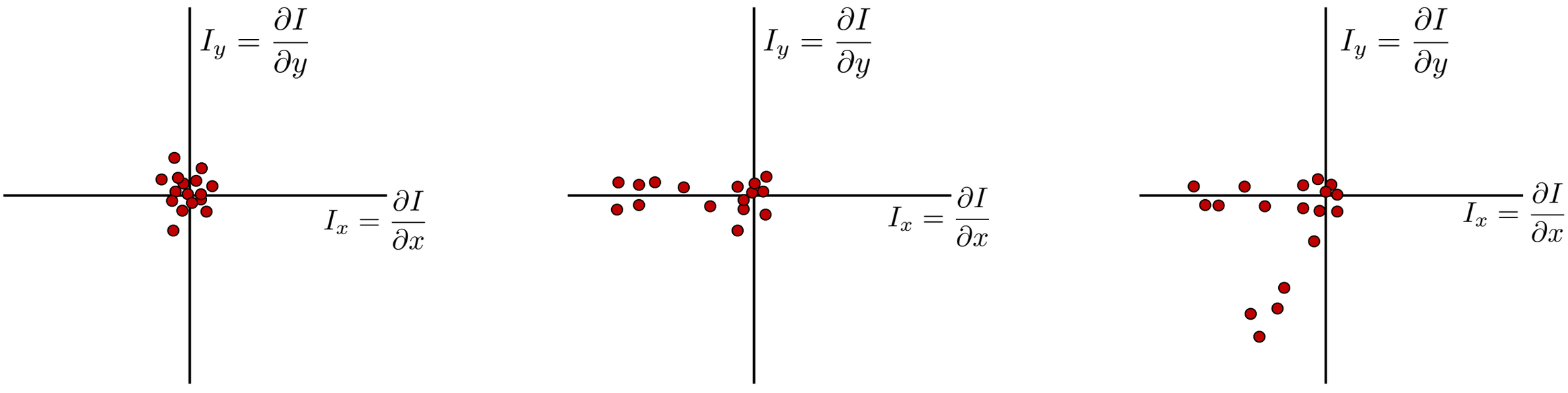
X derivative



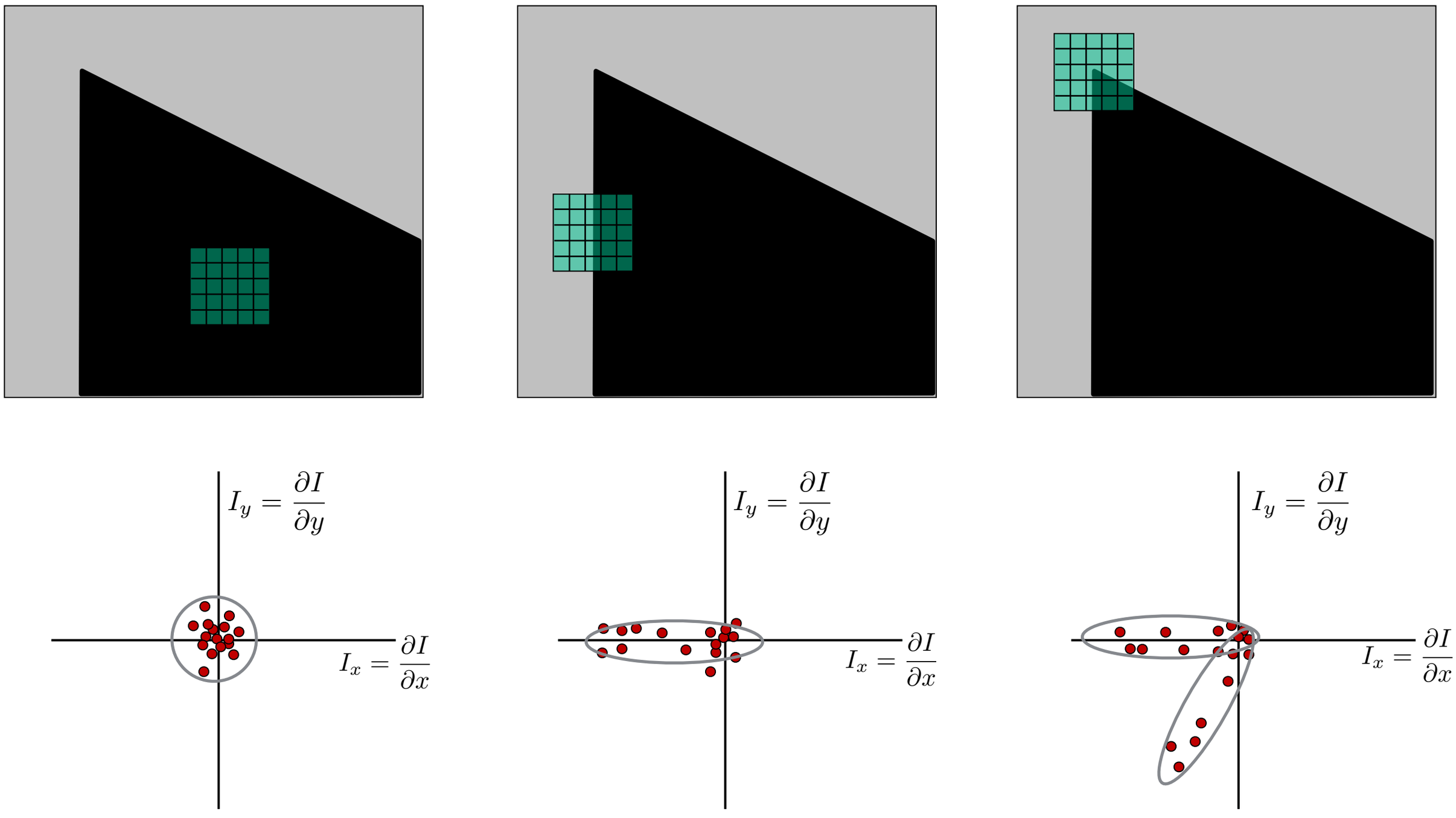
Y derivative



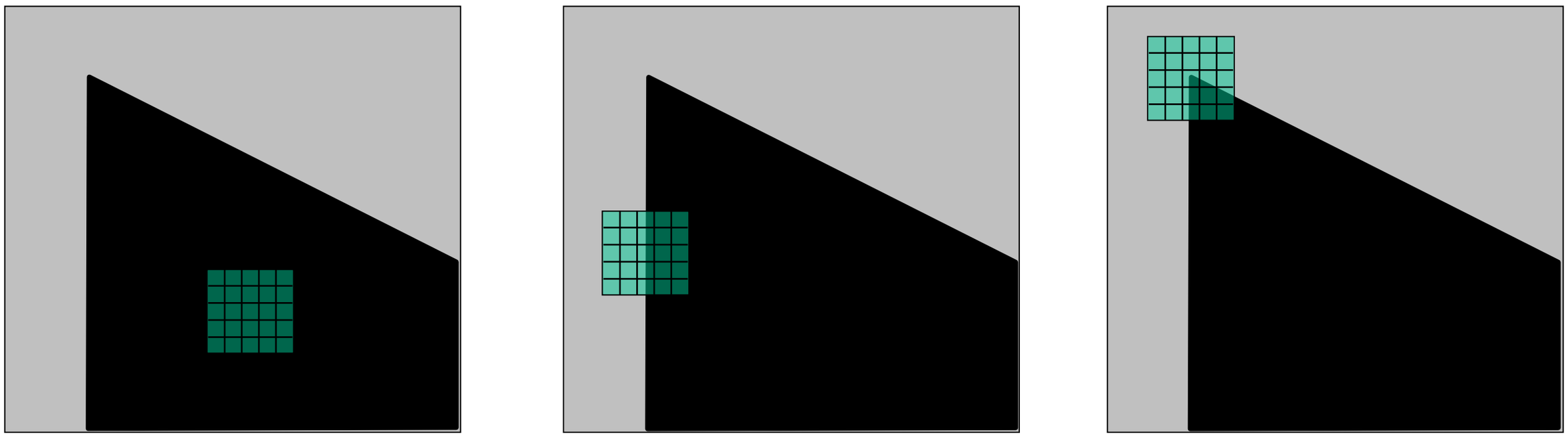


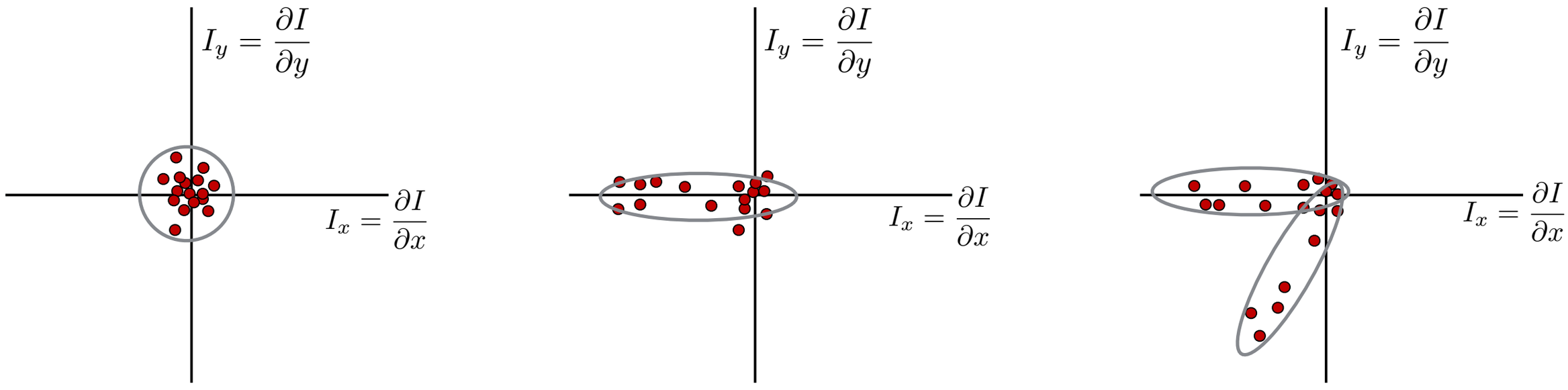



*What does the distribution tell you about the region?*




distribution reveals edge orientation and magnitude






*How do you quantify orientation and magnitude?*

# Harris Corner Detection

1. Compute image gradients over small region

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

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

**2. Compute the covariance matrix**

3. Compute eigenvectors and eigenvalues

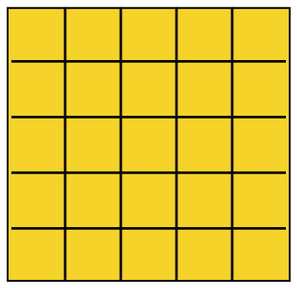
$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

4. Use threshold on eigenvalues to detect corners

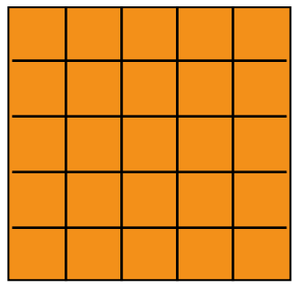
# 2. Compute the covariance matrix

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

$$\sum_{p \in P} I_x I_y = \text{sum}( \; I_x = \frac{\partial I}{\partial x} \; .* \; I_y = \frac{\partial I}{\partial y} \; )$$



array of x gradients



array of y gradients

*Where does this covariance matrix come from?*

# Error Function

Change of intensity for different shifts

Pixel-wise difference between two images

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

Shifted image

Original image

# Error Function

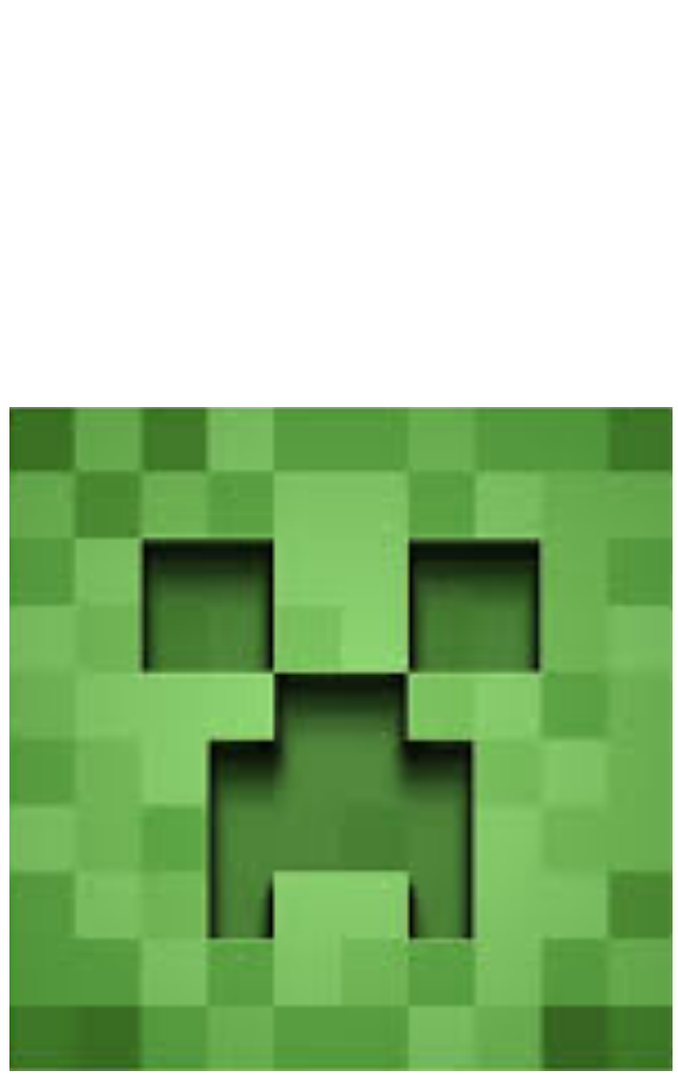
Change of intensity for different shifts

Pixel-wise difference between two images

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

Shifted image

Original image



# Error Function

Change of intensity for different shifts

Pixel-wise difference between two images

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

Shifted image

Original image

$E(-1,-1)$

# Error Function

Change of intensity for different shifts

Pixel-wise difference between two images

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

Shifted image

Original image

Difference value
for one shift value

$E(-1,-1)$

# Error Function

Change of intensity for different shifts

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

$E(-1,-1)$

# Error Function

Change of intensity for different shifts

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

$E(0,-1)$

# Error Function

Change of intensity for different shifts

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

$E(1,-1)$

# Error Function

Change of intensity for different shifts

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

# Error Function

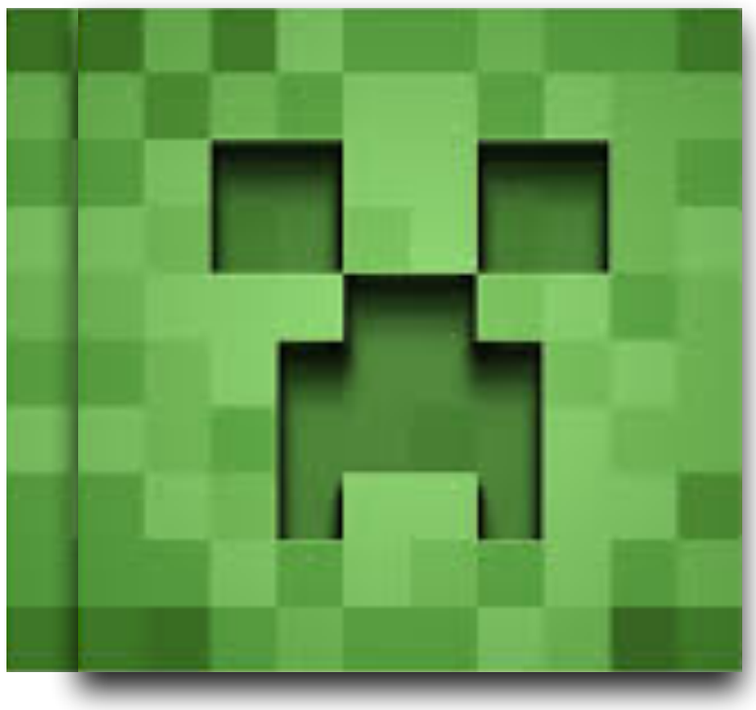
Change of intensity for different shifts

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

# Error Function

Change of intensity for different shifts

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



# Error Function

Change of intensity for different shifts

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

# Error Function

Change of intensity for different shifts

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

# Error Function

Change of intensity for different shifts

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

# Error Function

Change of intensity for different shifts

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

*What does it mean if all error values are **zero**?*

*What does it mean if all error values are **large**?*

*How can we characterize how 'flat' the image is using the error function?*

We can approximate the error function as a quadratic…

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

We can approximate the error function as a quadratic…

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

$$\approx \sum_{x,y \in W} [I(x,y) + [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)]^2$$

Taylor series approximation

We can approximate the error function as a quadratic…

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

$$\approx \sum_{x,y \in W} [I(x,y) + [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)]^2$$

$$= \sum_{x,y \in W} \left[ [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$$

We can approximate the error function as a quadratic…

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

$$\approx \sum_{x,y \in W} [I(x,y) + [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)]^2$$

$$= \sum_{x,y \in W} \left[ [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$$

$$= \sum_{x,y \in W} [u \; v] \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

This is a quadratic (bilinear) function.

$$E(u,v) = \sum_{x,y \in W} [u\ v] \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= [u\ v] \sum_{x,y \in W} \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= [u\ v]\ \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}$$

For small shifts,
error function is a bilinear (quadratic) function

By computing the gradient covariance matrix…

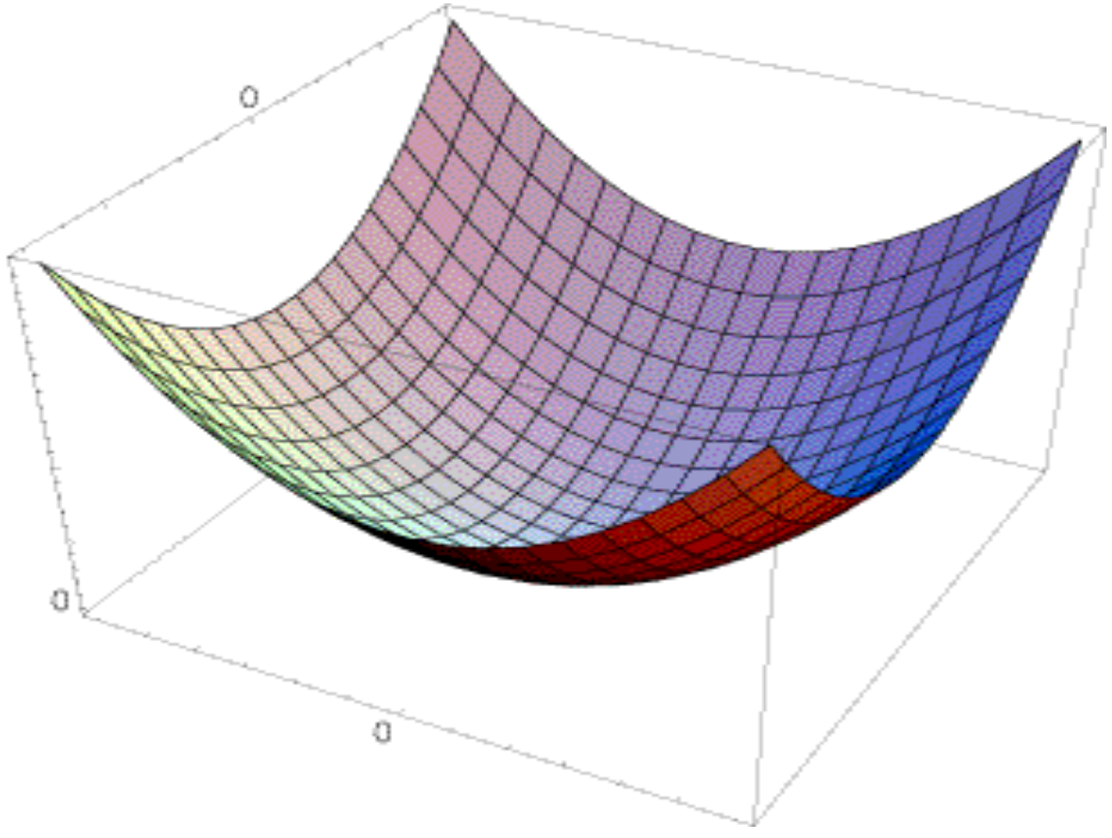
$$\mathbf{M} = \begin{bmatrix} \sum_{x,y \in W} I_x I_x & \sum_{x,y \in W} I_x I_y \\ \sum_{x,y \in W} I_y I_x & \sum_{x,y \in W} I_y I_y \end{bmatrix}$$

we are fitting a quadratic to the error function
over a  small image region

*How do we interpret this covariance matrix?*

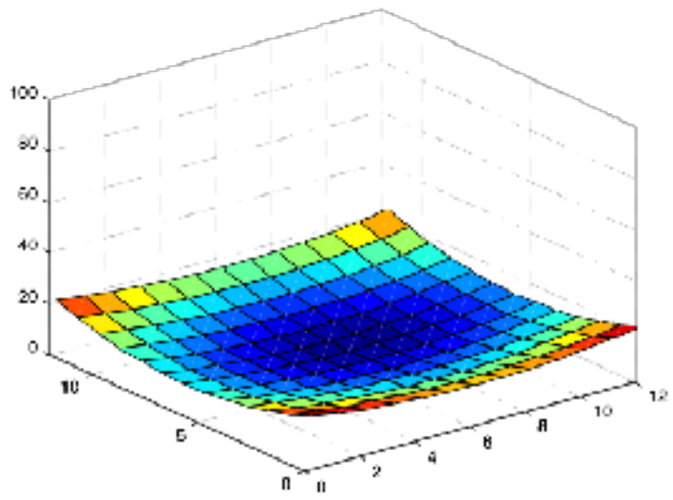
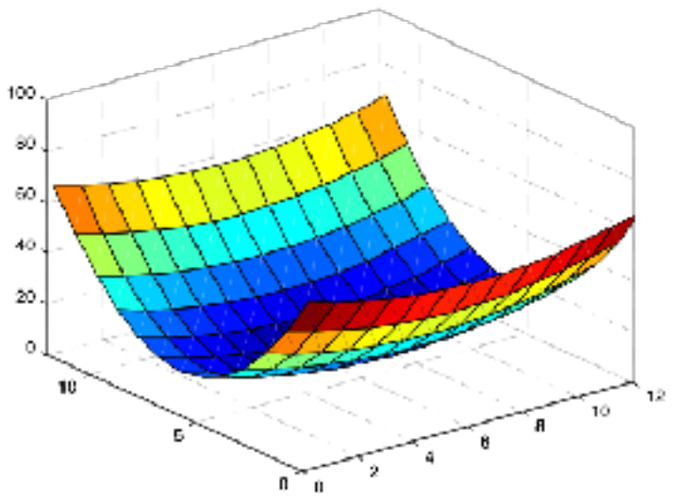
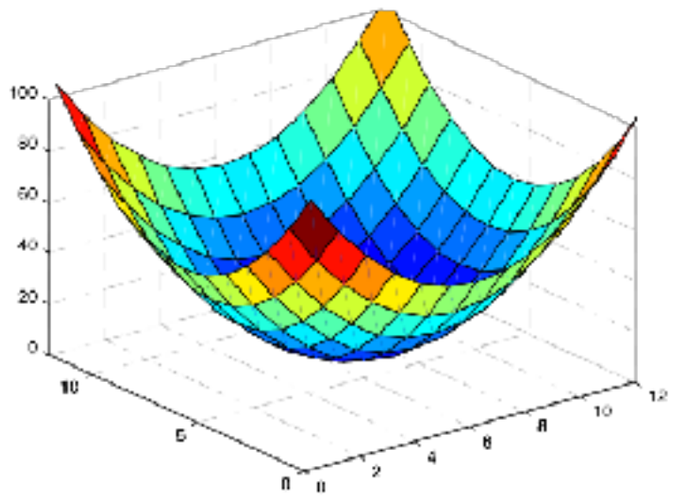
The surface $E(u,v)$ is locally approximated by a quadratic form

$$[u \ v] \ \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}$$



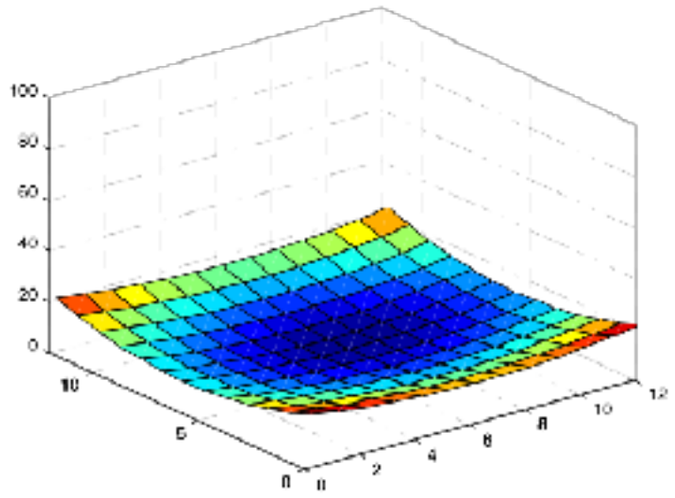
What does the curvature of the quadratic tell us about the original image?

$$[u\ v]\ \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}$$

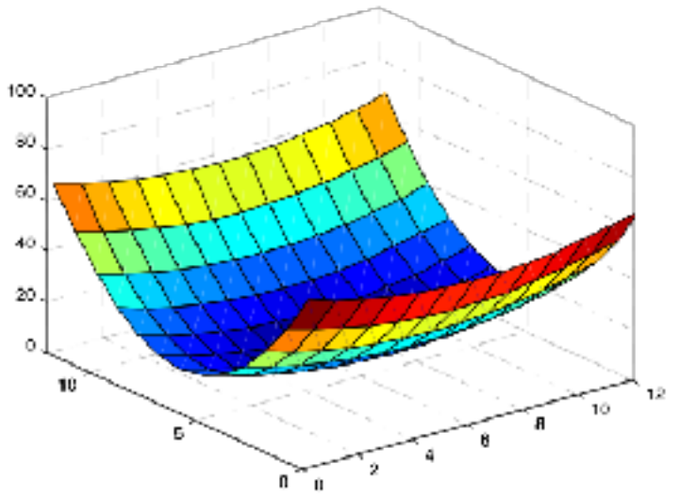




*What kind of image patch do these surfaces represent?*

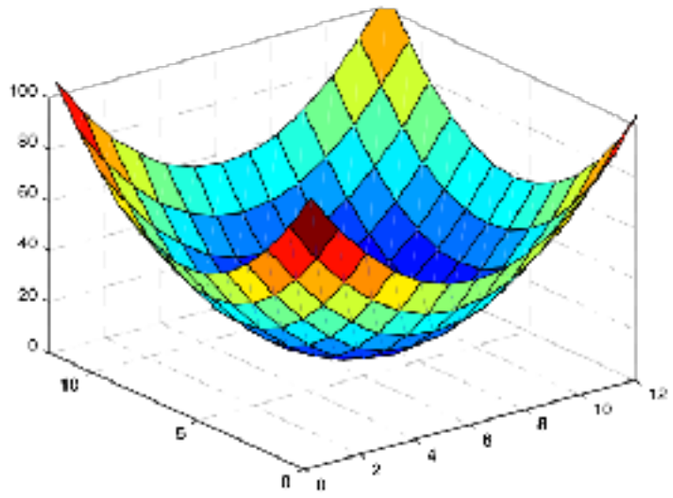
$$[u\ v]\ \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}$$




flat




edge
‘line’




corner
‘dot’

# Harris Corner Detection

1. Compute image gradients over small region

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

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

2. Compute the covariance matrix

**3. Compute eigenvectors and eigenvalues**

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

4. Use threshold on eigenvalues to detect corners

eig(M)

eigenvalue

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

eigenvector

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

eigenvalue

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

eigenvector

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

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

(returns a polynomial)

eigenvalue

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

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)

eigenvalue

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

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)

3. For each eigenvalue, solve $(M - \lambda I)\boldsymbol{e} = 0$
(returns eigenvectors)

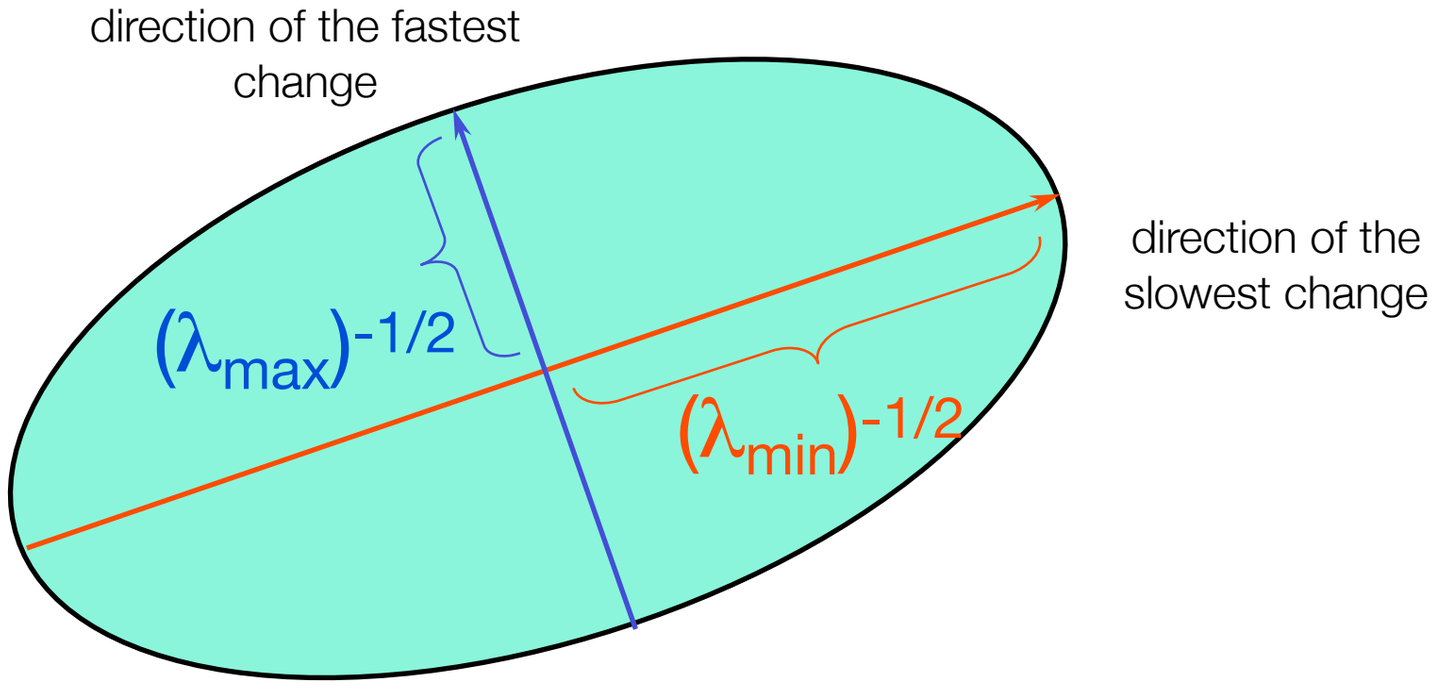
# Visualization as an ellipse

Since M is symmetric, we have $M = R^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}R$

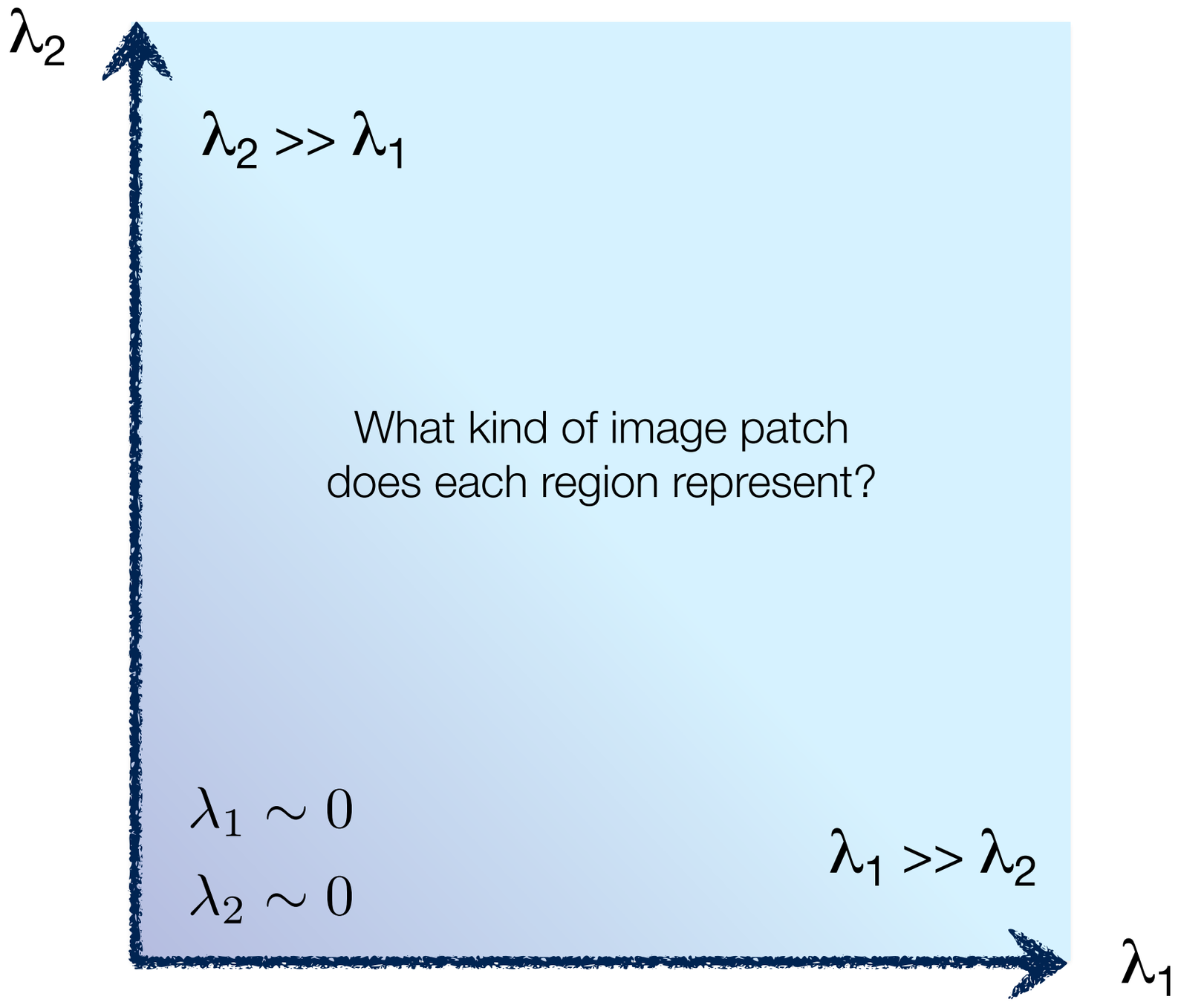
We can visualize *M* as an ellipse with axis lengths determined by the eigenvalues and orientation determined by *R*

Ellipse equation:

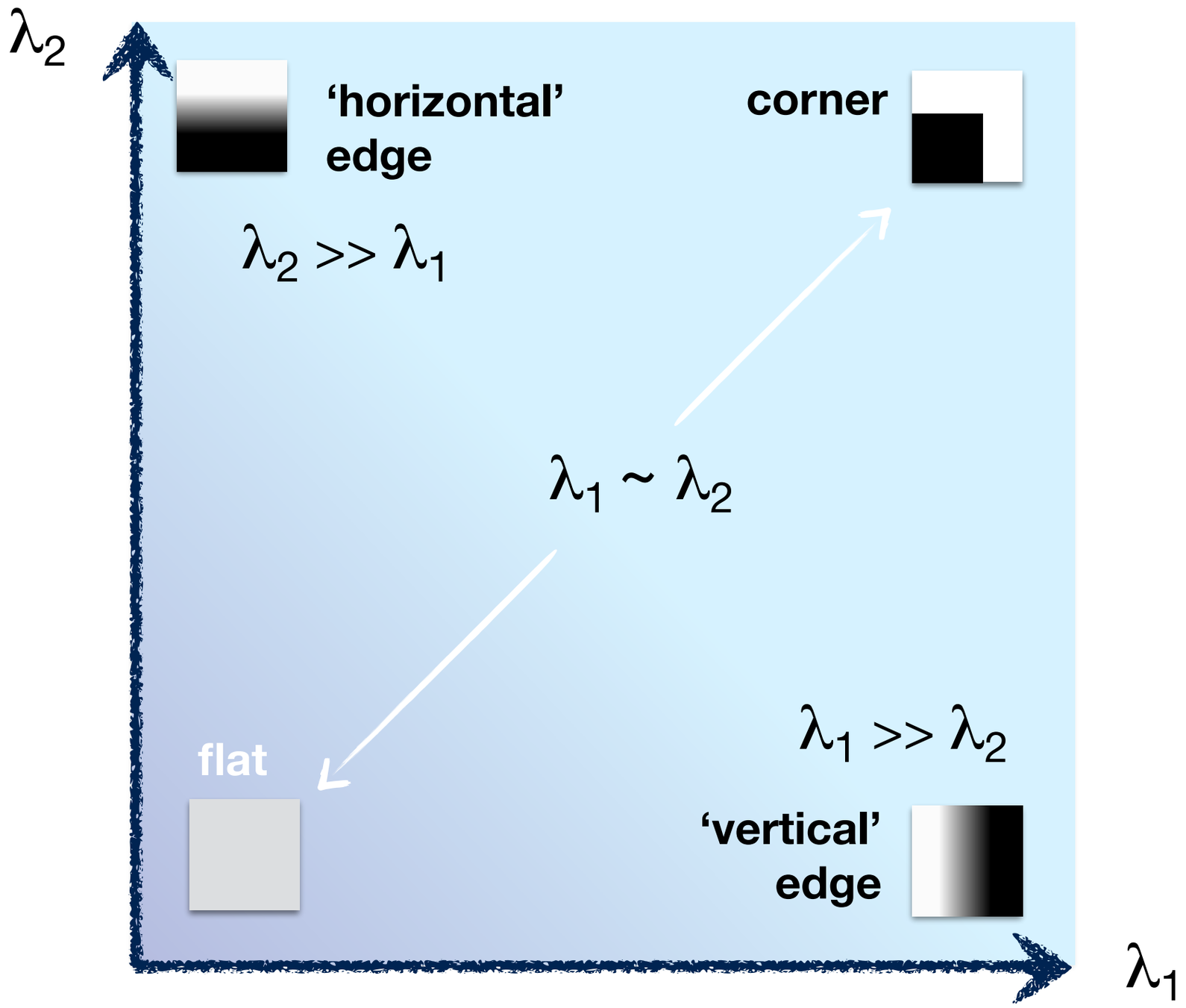
$$[u \quad v]\ M \begin{bmatrix}u\\ v\end{bmatrix} = \text{const}$$



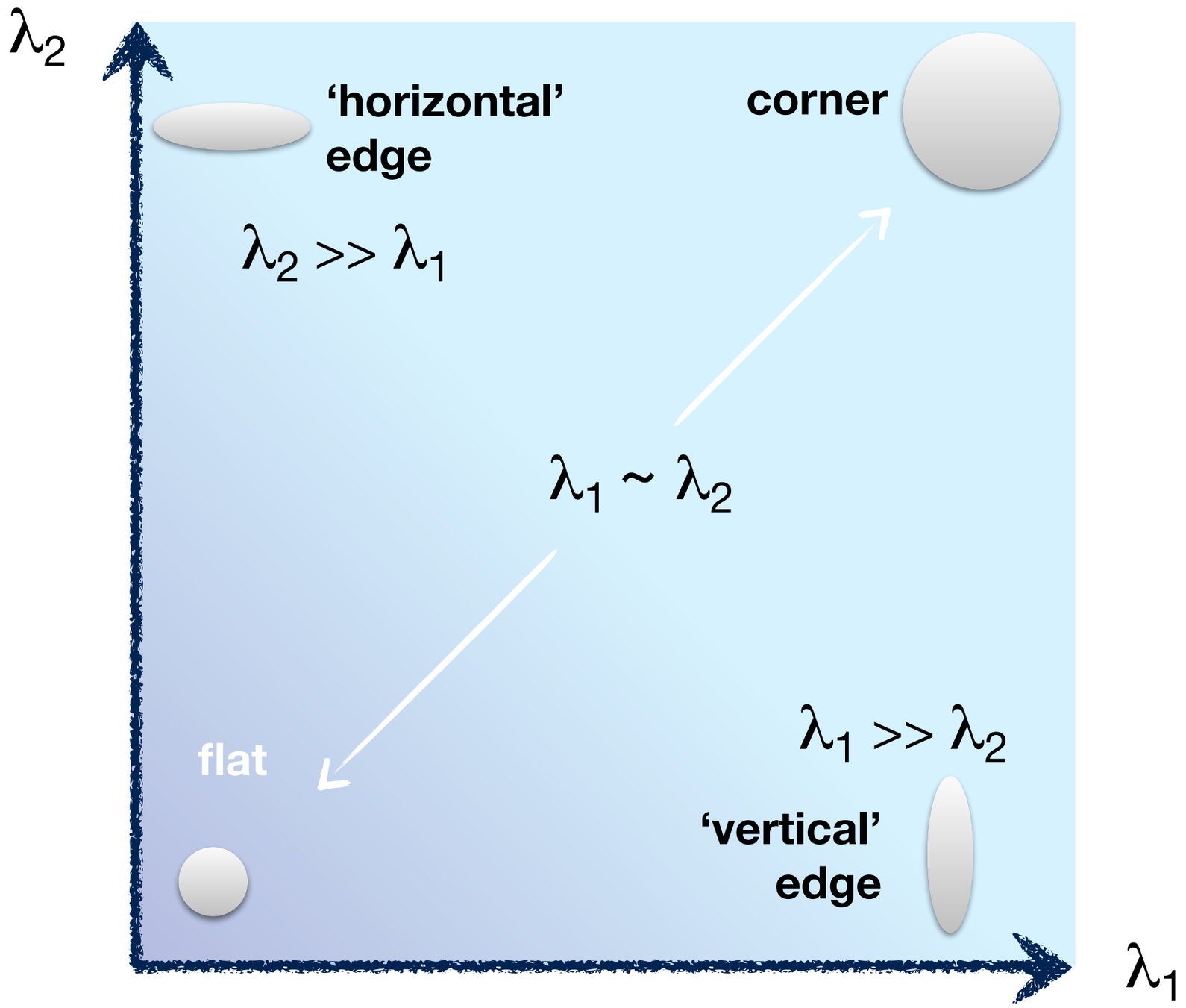
# interpreting eigenvalues



# interpreting eigenvalues



# interpreting eigenvalues



# Harris Corner Detection

1. Compute image gradients over small region

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

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

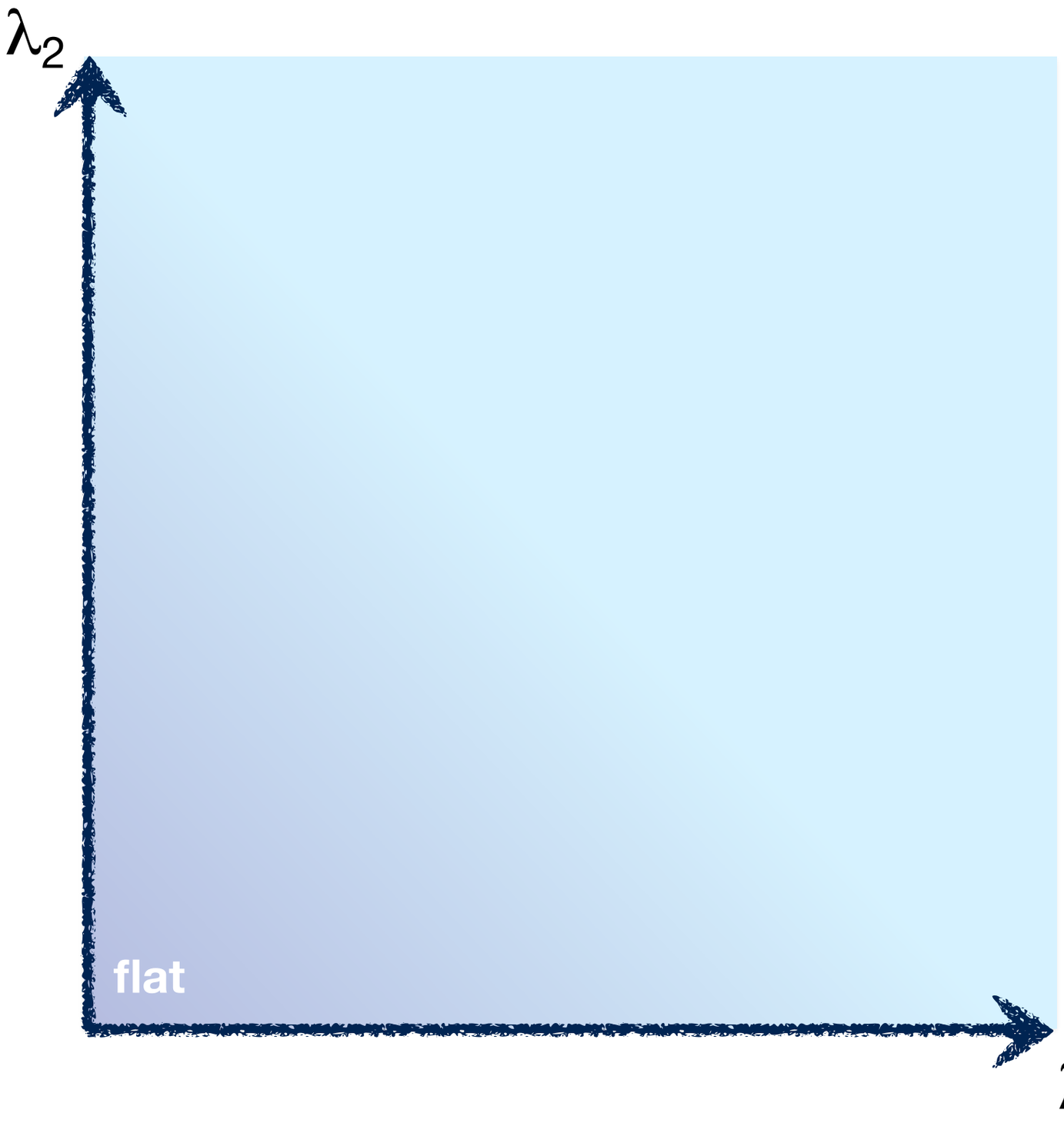
2. Compute the covariance matrix

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

3. Compute eigenvectors and eigenvalues

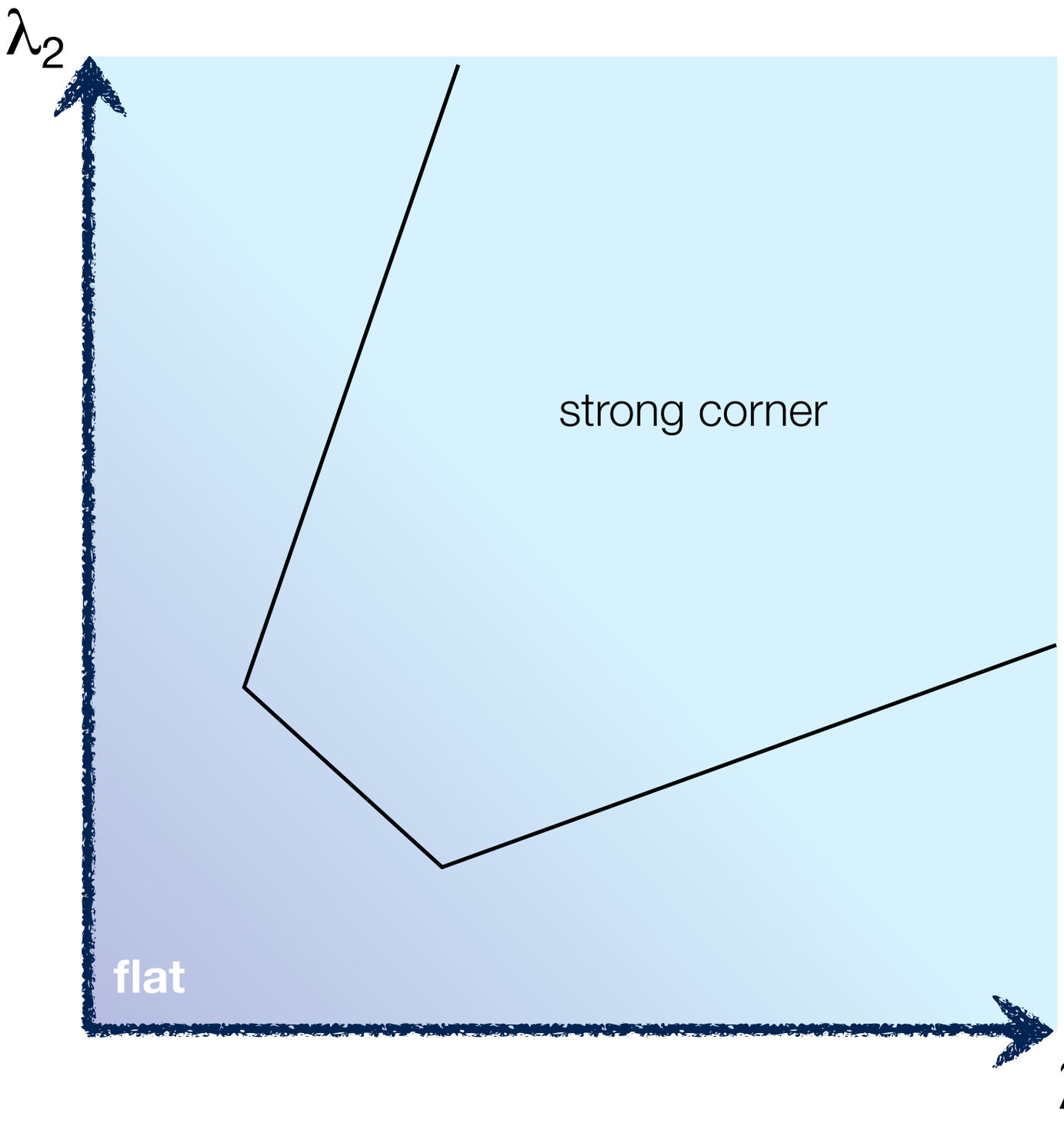
**4. Use threshold on eigenvalues to detect corners**

# 5. Use threshold on eigenvalues to detect corners




Think of a function to score ‘cornerness’

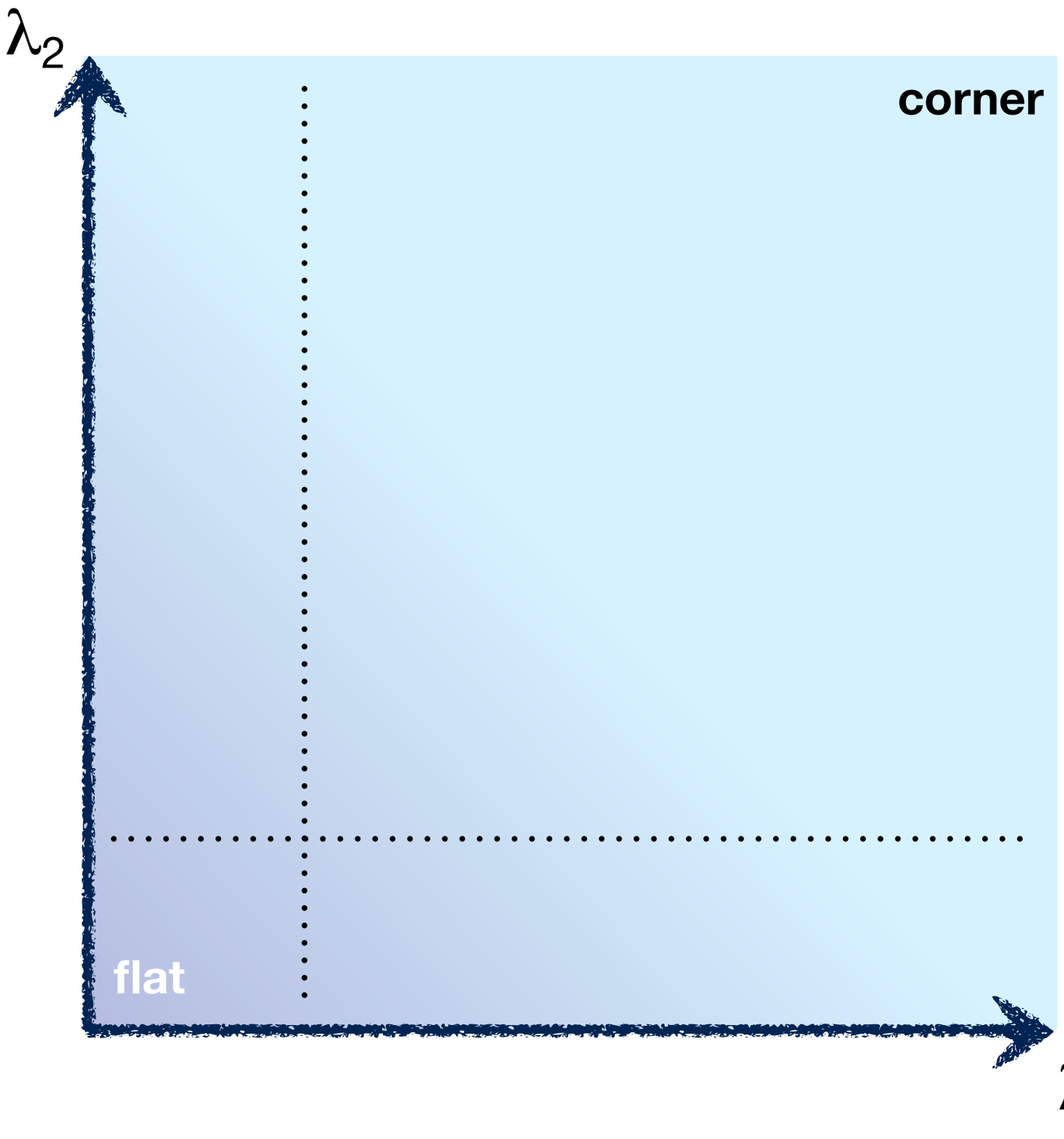
# 5. Use threshold on eigenvalues to detect corners




Think of a function to score ‘cornerness’

# 5. Use threshold on eigenvalues to detect corners

(a function of $\hat{}$)

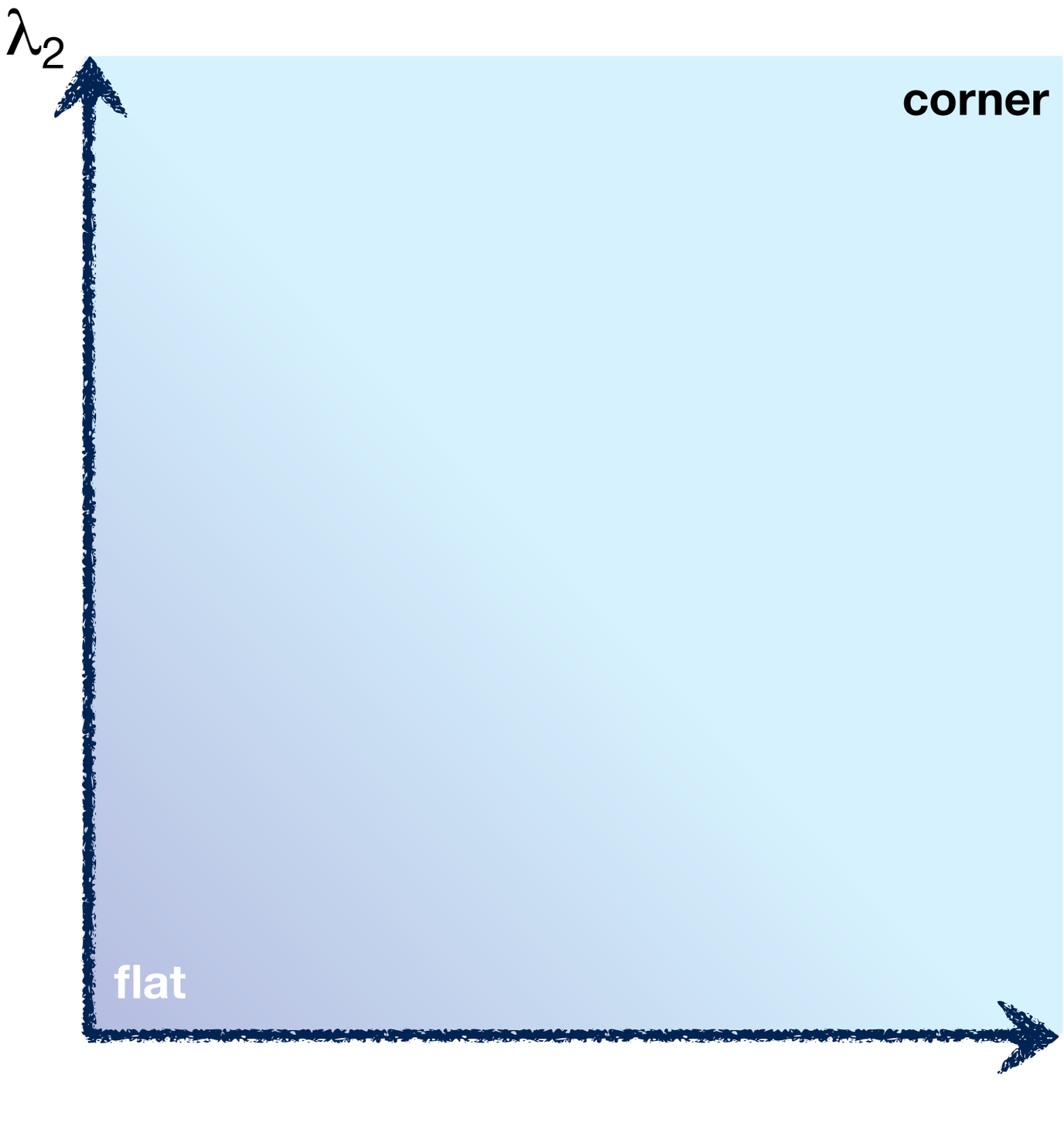



Use the smallest eigenvalue as the response function

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

# 5. Use threshold on eigenvalues to detect corners

(a function of $\hat{}$)



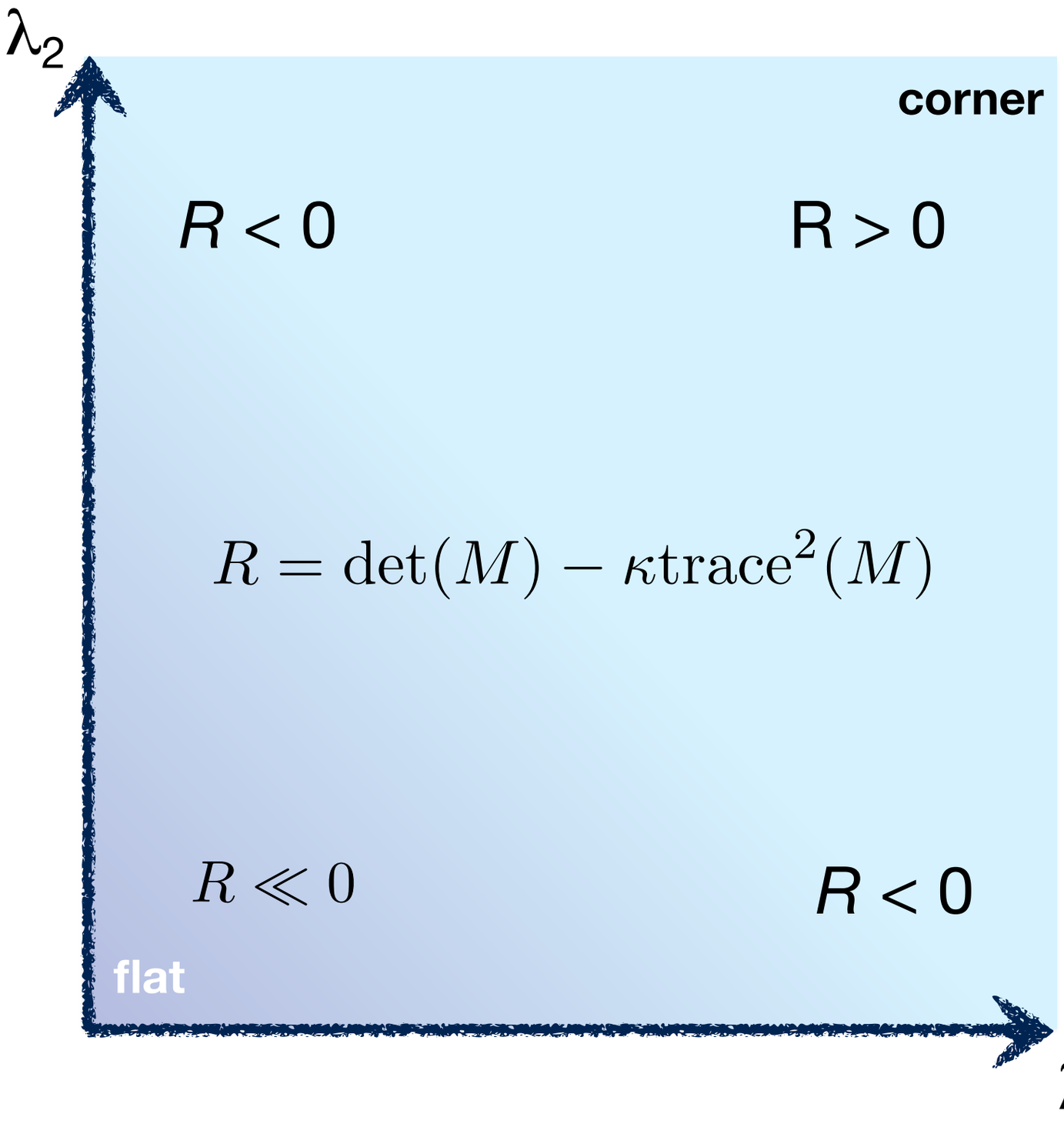

Eigenvalues need to be bigger than one.

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

Can compute this more efficiently…

# 5. Use threshold on eigenvalues to detect corners

(a function of $\hat{}$)




$$\det M = \lambda_1 \lambda_2$$

$$\mathrm{trace}\, M = \lambda_1 + \lambda_2$$

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

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

Harris & Stephens (1988)

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

Kanade & Tomasi (1994)

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

Nobel (1998)

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

# Summary

C.Harris and M.Stephens. “A Combined Corner and Edge Detector.”1988.

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \qquad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x^2} = I_x \cdot I_x \qquad I_{y^2} = I_y \cdot I_y \qquad I_{xy} = I_x \cdot I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x^2} = G_{\sigma'} * I_{x^2} \qquad S_{y^2} = G_{\sigma'} * I_{y^2} \qquad S_{xy} = G_{\sigma'} * I_{xy}$$

# Summary

C.Harris and M.Stephens. “A Combined Corner and Edge Detector.”1988.

4. Define the matrix at each pixel

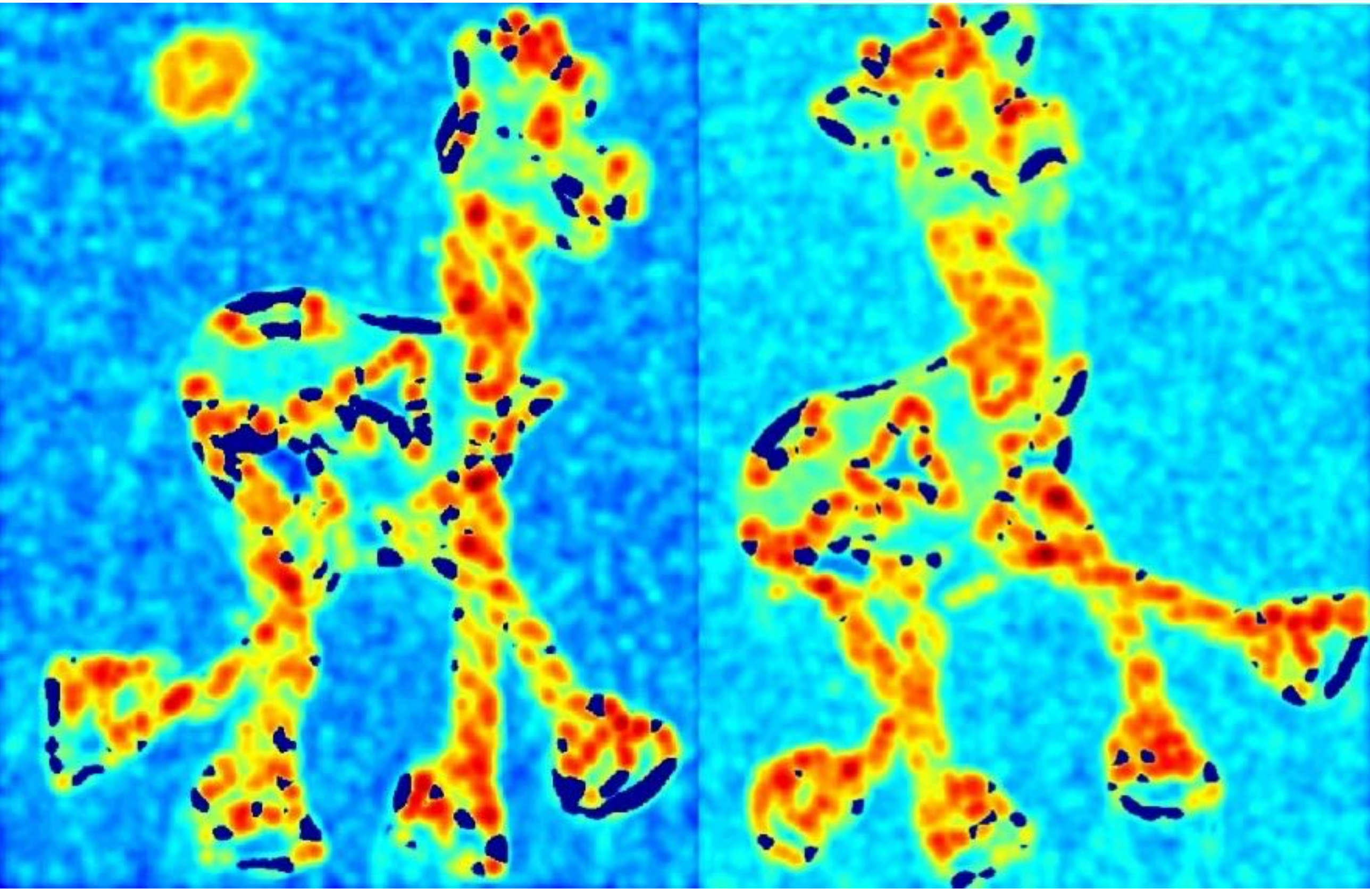
$$M(x,y) = \begin{bmatrix} S_{x^2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y^2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

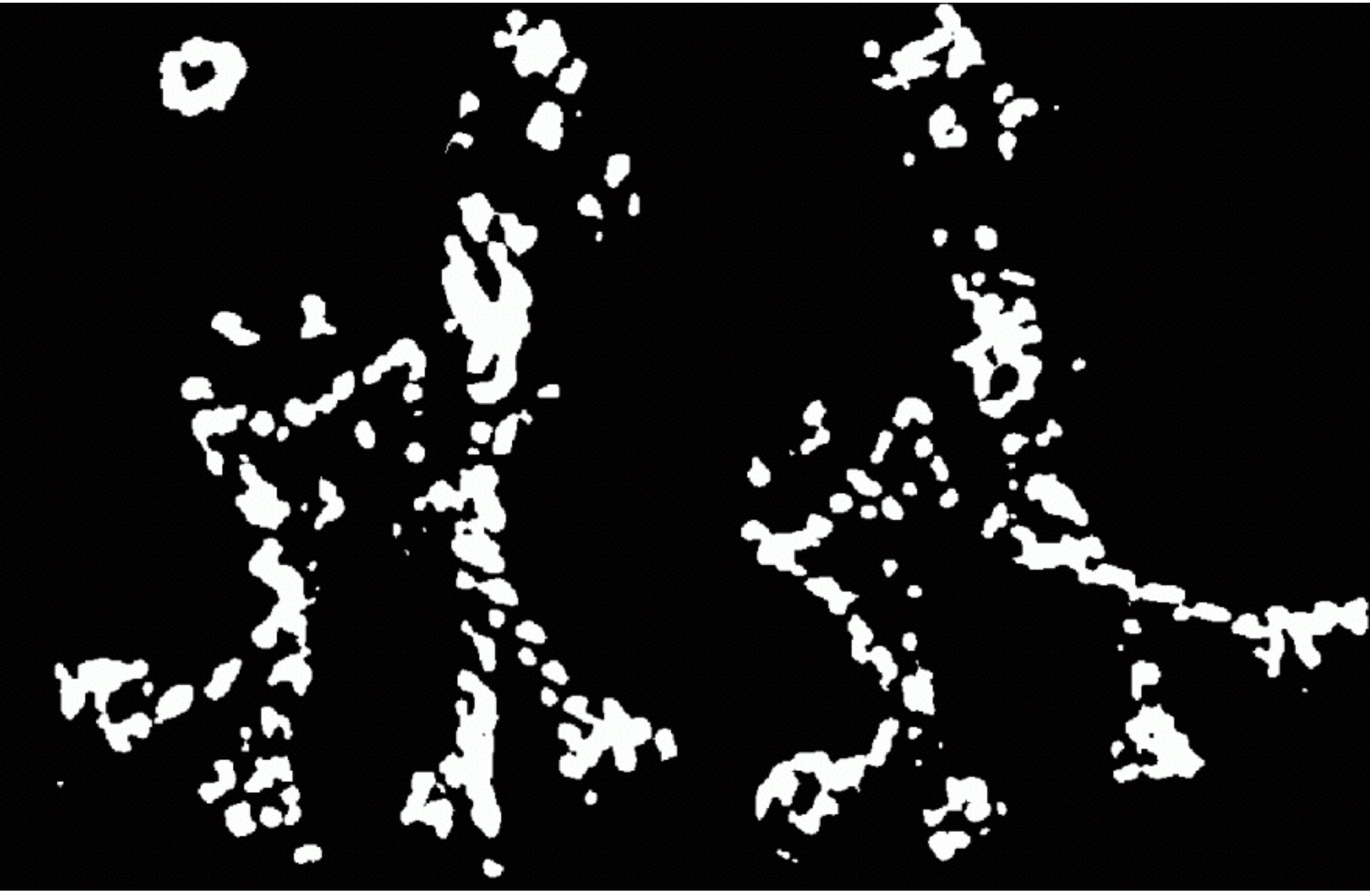
$$R = \det M - k\left(\text{trace}M\right)^2$$

6. Threshold on value of R; compute non-max suppression.

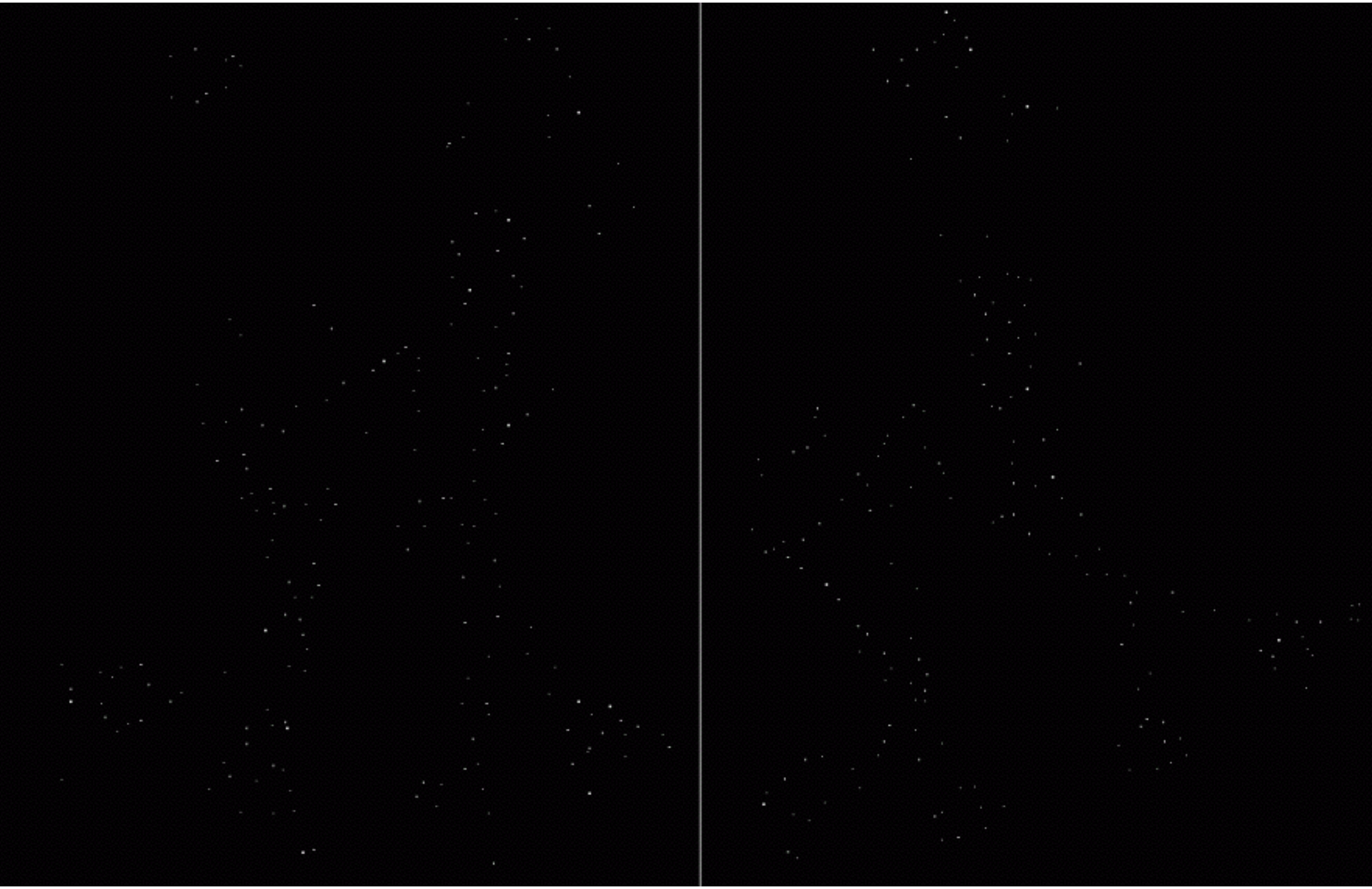
# Corner response



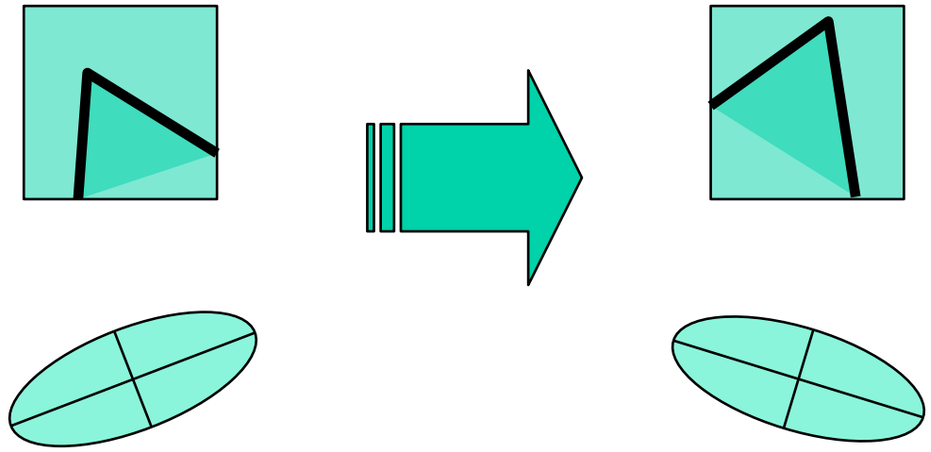
# Thresholded corner response



# Non-maximal suppression



# Harris corner response is rotation invariant



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

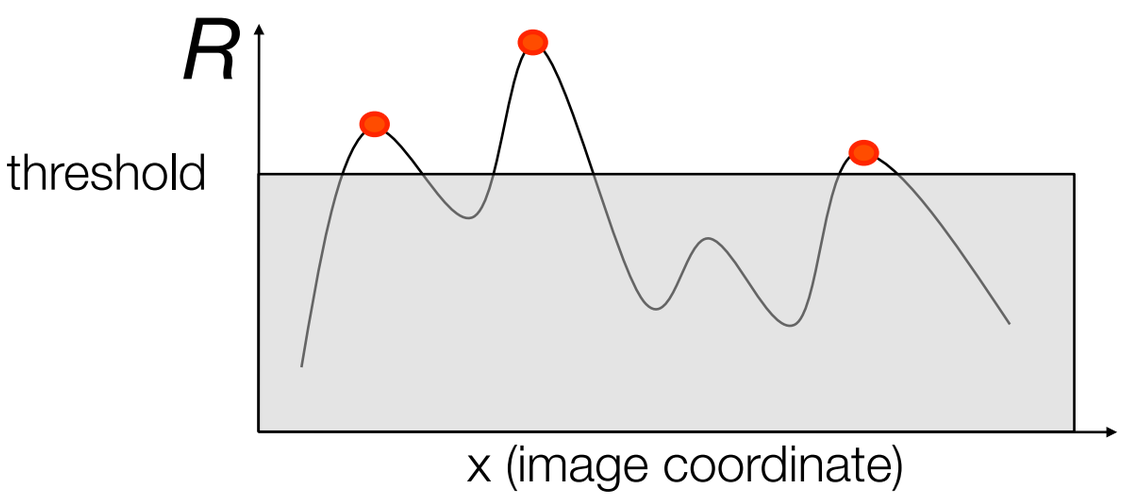
**Corner response R is invariant to image rotation**

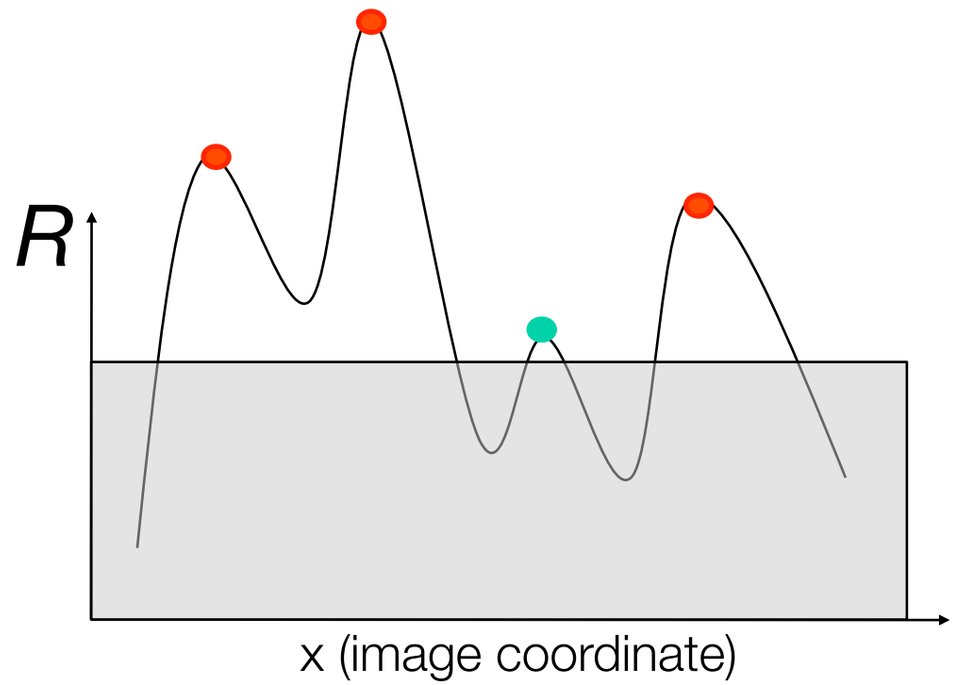
# intensity changes

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 not invariant to changes in …