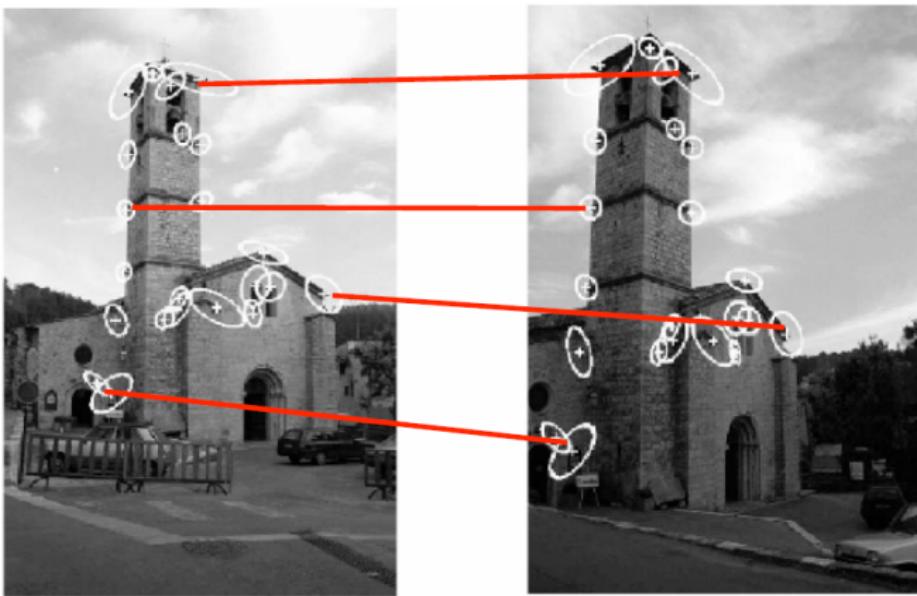


Harris Corner Detector



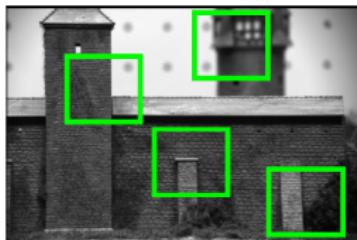
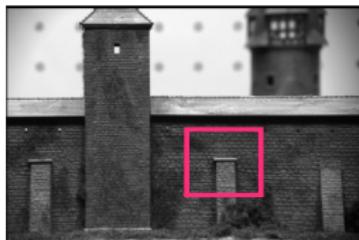
Motivation: Matching Problem

- Vision tasks such as stereo and motion estimation or tracking require finding corresponding features across two or more images.



Motivation: Patch Matching

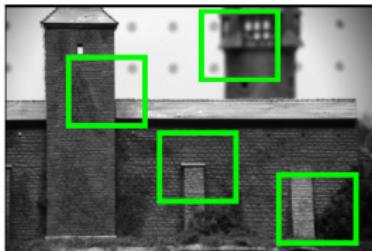
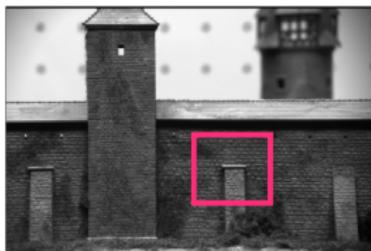
Elements to be matched are image patches of fixed size



Task: find the best (most similar) patch in a second image



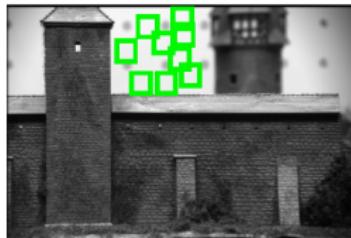
Not all patches are created equal



Intuition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar).



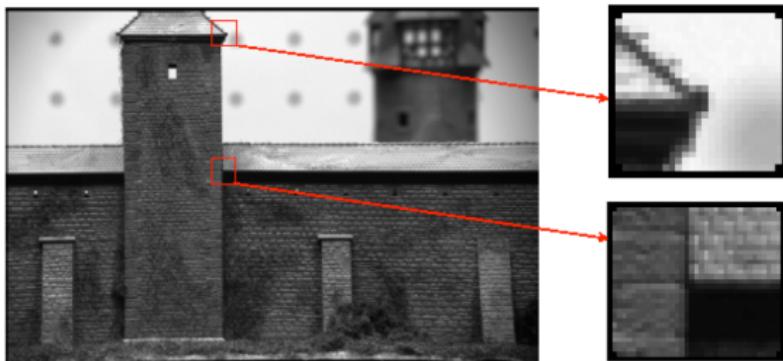
Not all patches are created equal



Intuition: this would be a BAD patch for matching, since it is not very distinctive (there are many similar patches in the second frame)

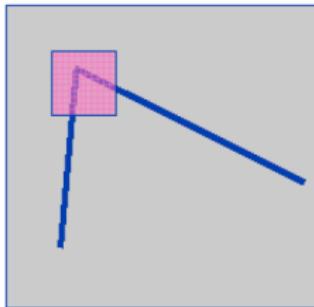
$$\begin{array}{c} \textcolor{red}{\square} \\ ? \end{array} = \begin{array}{c} \textcolor{green}{\square} \\ \textcolor{green}{\square} \\ \textcolor{green}{\square} \\ \textcolor{green}{\square} \end{array}$$

What are corners?



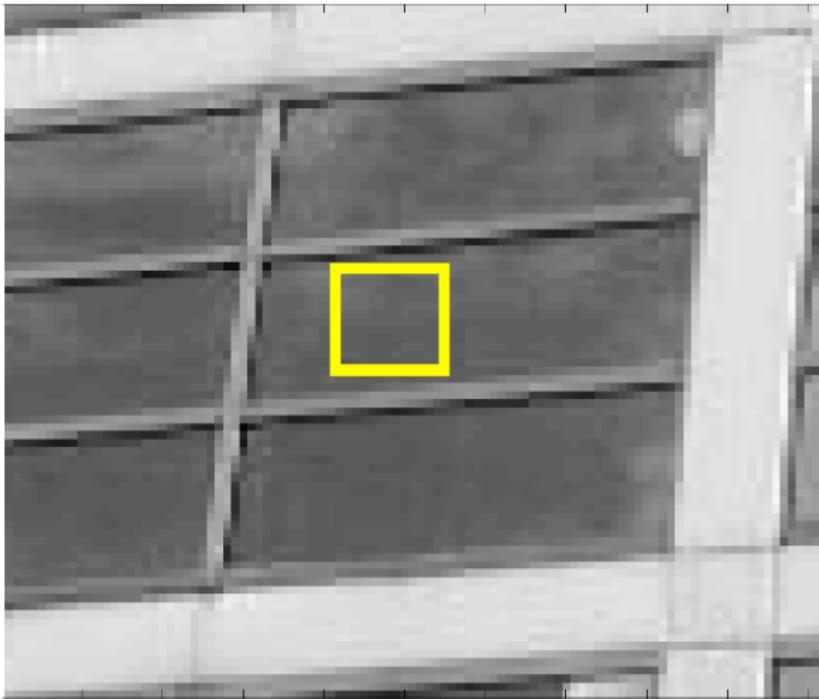
- Intuitively, junctions of contours.
- Generally more stable features over changes of viewpoint
- Intuitively, large variations in the neighborhood of the point in all directions
- They are good features to match!

- We should easily recognize the point by looking at intensity values within a small window
- Shifting the window in *any direction* should yield a *large change* in appearance.

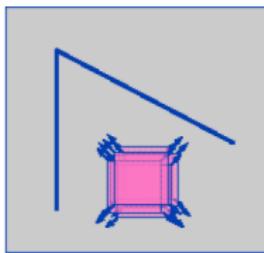


Appearance Change in a Local Neighbourhood

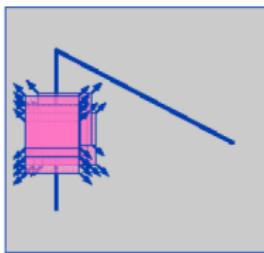
UCL



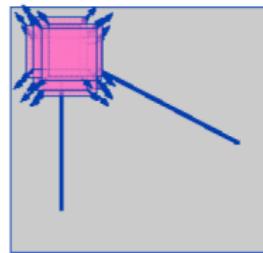
Harris Corner Detector: Basic Idea



“flat” region:
no change in
all directions

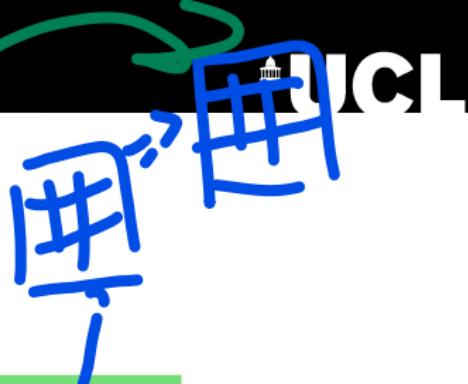


“edge”:
no change along
the edge direction



“corner”:
significant change
in all directions

Harris Corner Detector: Maths



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

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

Window function

Shifted intensity

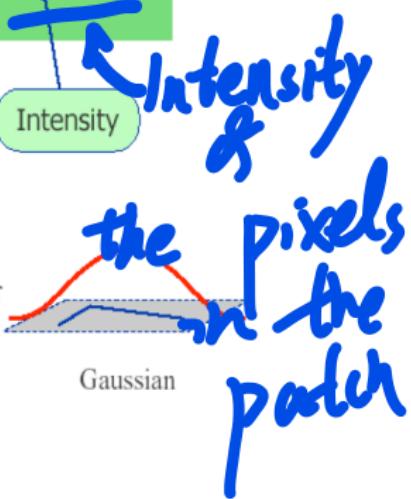
Intensity

Window function $w(x, y) =$

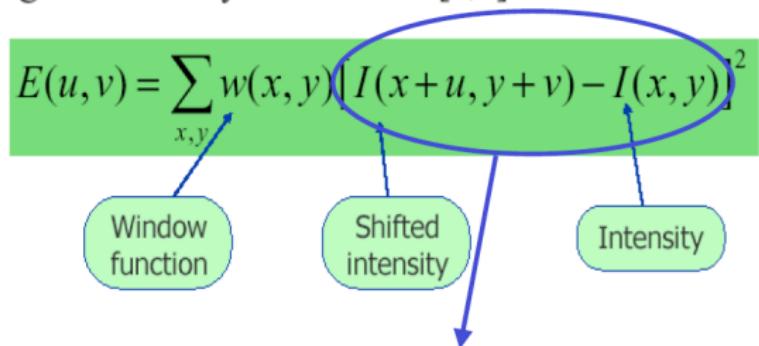


1 in window, 0 outside

or



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

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


Window function Shifted intensity Intensity

For nearly constant patches, this will be near 0.
For very distinctive patches, this will be larger.
Hence... we want patches where $E(u, v)$ is LARGE.

$$f(x+u, y+v) = f(x, y) + u f_x(x, y) + v f_y(x, y) +$$

First partial derivatives

$$\frac{1}{2!} [u^2 f_{xx}(x, y) + uv f_{xy}(x, y) + v^2 f_{yy}(x, y)] +$$

Second partial derivatives

$$\frac{1}{3!} [u^3 f_{xxx}(x, y) + u^2 v f_{xxy}(x, y) + uv^2 f_{xyy}(x, y) + v^3 f_{yyy}(x, y)]$$

Third partial derivatives

+ ... (Higher order terms)

First order approx

$$f(x+u, y+v) \approx f(x, y) + u f_x(x, y) + v f_y(x, y)$$

Harris Corner Derivation

$$\sum [I(x+u, y+v) - I(x, y)]^2$$

$$\approx \sum [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad \text{First order approx}$$

$$= \sum u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2$$

$$= \sum \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{Rewrite as matrix equation}$$

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

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

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

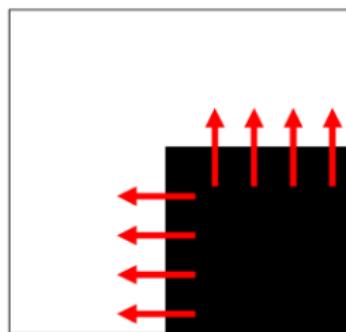
where M is a 2×2 matrix computed from image derivatives:

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

Widening function - computing a weighted sum (simplest case, $w=1$)

Note: these are just products of components of the gradient, I_x, I_y

First, consider an axis-aligned corner:



- First, consider an axis-aligned corner:

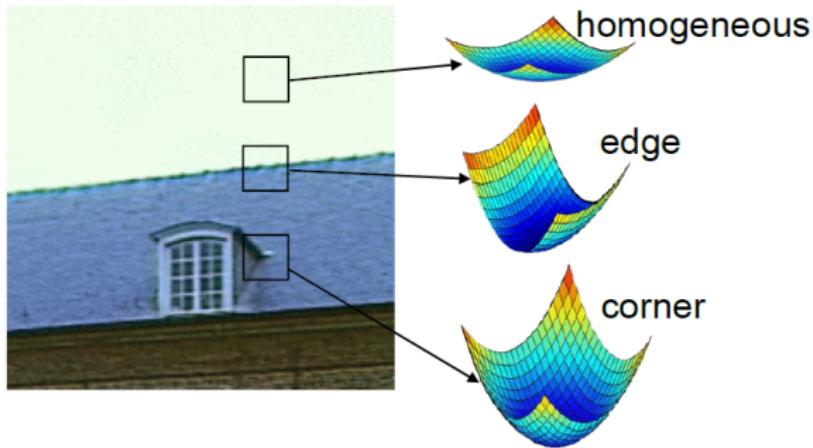
$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

- 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.

- M captures the curvature of the local autocorrelation surface
- The eigenvalues of M are the principal curvatures
- They are the solutions to:

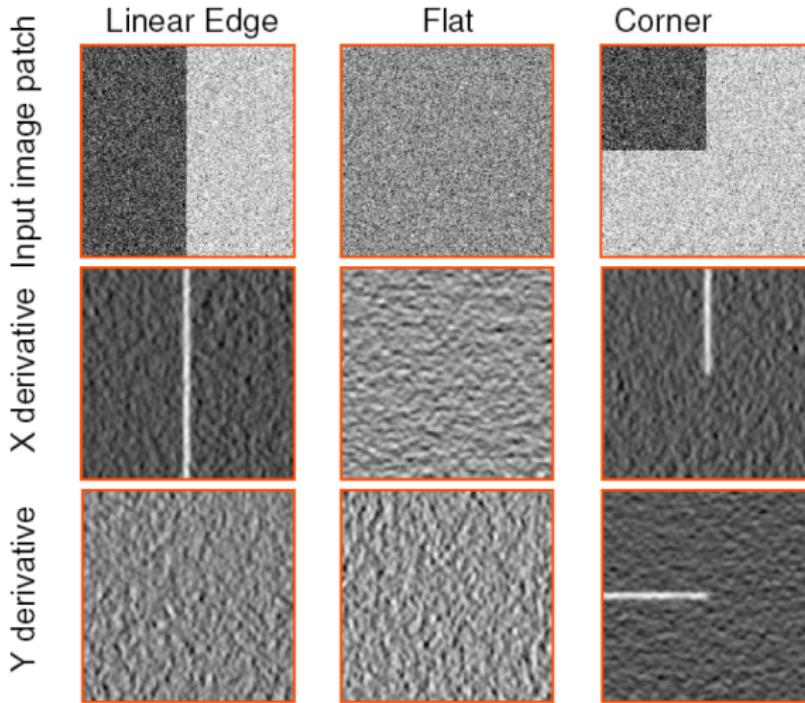
$$\lambda^2 - \lambda(\sum I_x^2 + \sum I_y^2) + (\sum I_x)^2(\sum I_y)^2 - (\sum I_x I_y)^2 = 0$$

Principal Axes (Eigenvectors)



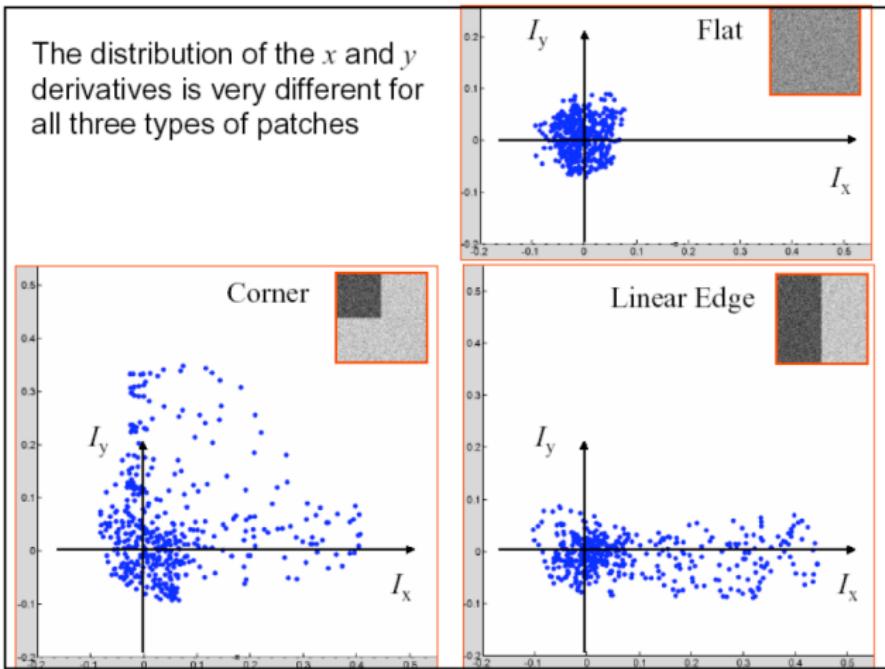
Corner = both eigenvalues are large

Example: Cases and 2D Derivatives



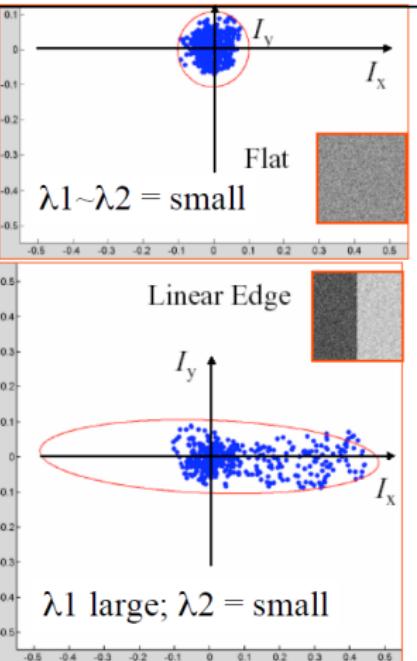
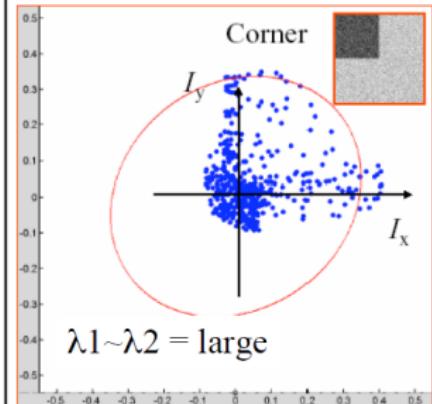
Plotting Derivatives as 2D Points

The distribution of the x and y derivatives is very different for all three types of patches



Fitting an Ellipse to each Set of Points

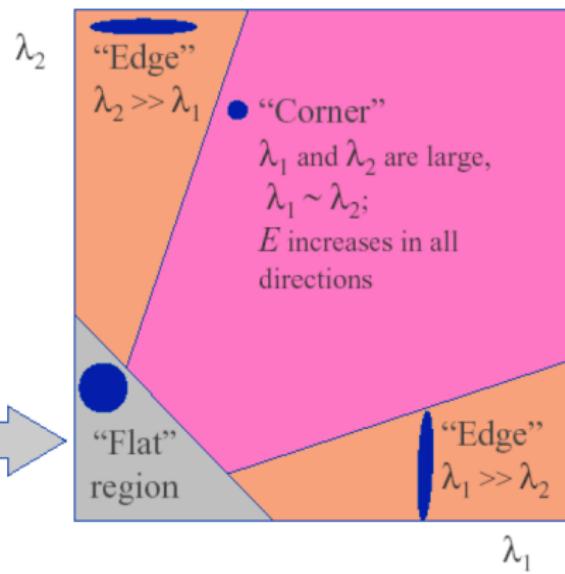
The distribution of x and y derivatives can be characterized by the shape and size of the principal component ellipse



Classification via Eigenvalues

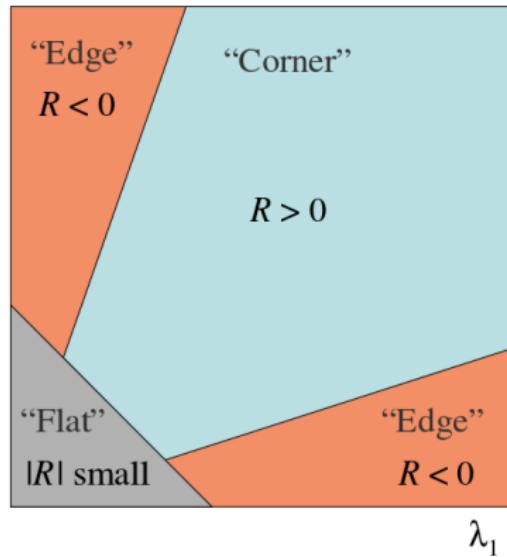
Classification of
image points using
eigenvalues of M :

λ_1 and λ_2 are small;
 E is almost constant
in all directions



Classification via Eigenvalues

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Measure of corner response:

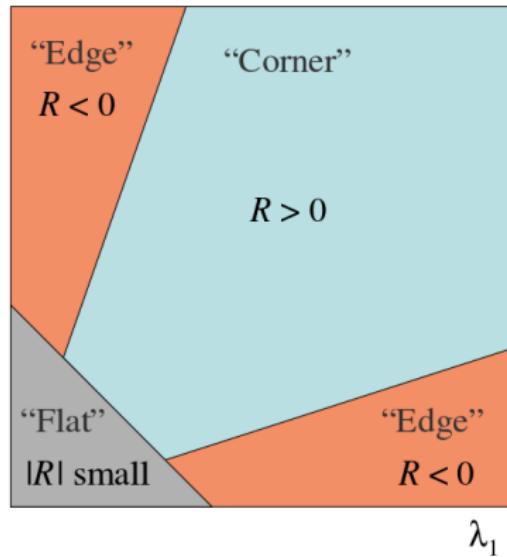
$$R = \det M - k (\operatorname{trace} M)^2$$

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

(k is an empirically determined constant; $k = 0.04 - 0.06$)

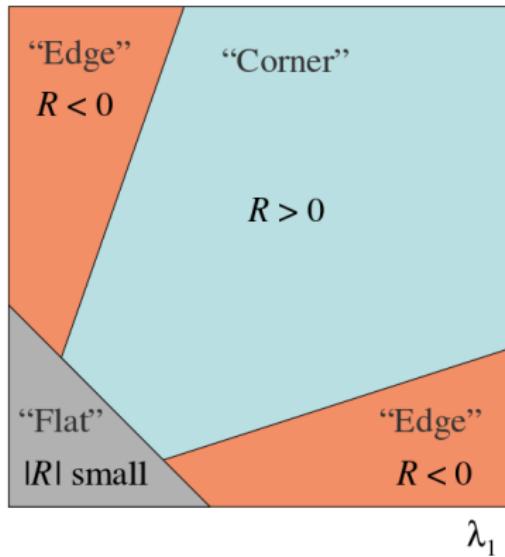
Corner Response Measure

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Harris Corner Detector: Algorithm

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region

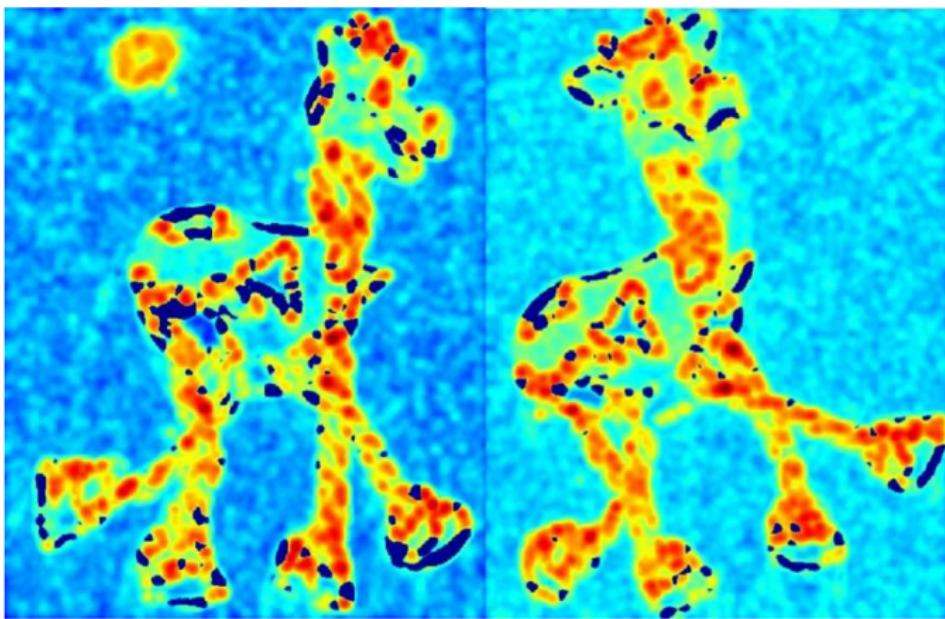


Harris Detector: Workflow



Compute corner response R

Compute corner response R



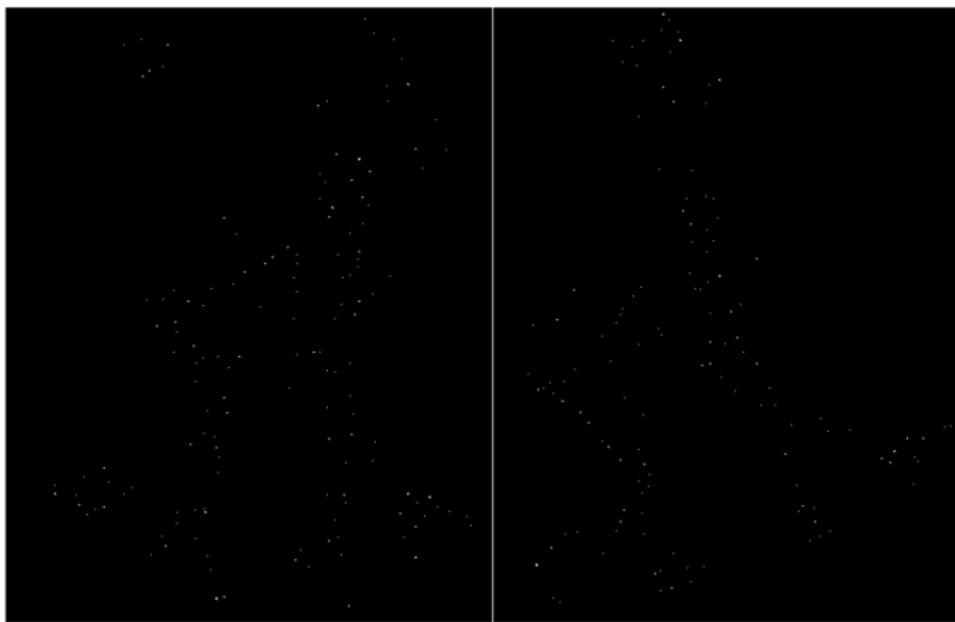
Find points with large corner response $R_i > \text{threshold}$

Find points with large corner response: $R > \text{threshold}$



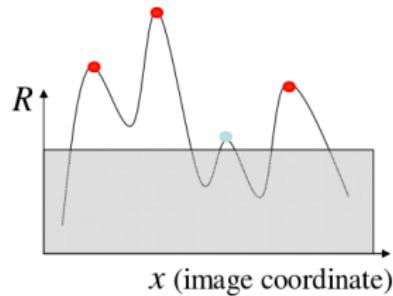
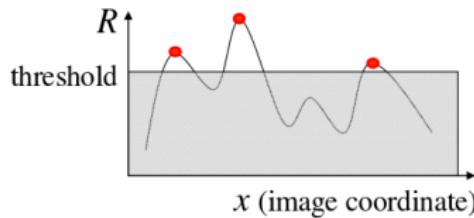
Non-maxima suppression

Take only the points of local maxima of R



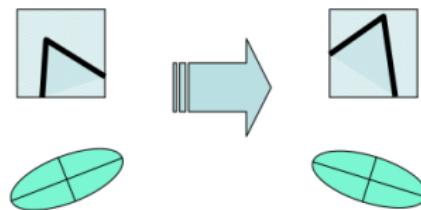
Harris Detector: Properties

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



Invariant to rotations

- Rotation invariance

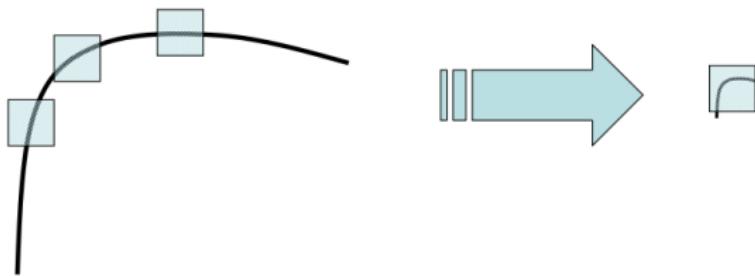


Ellipse rotates but its shape (i.e. eigenvalues)
remains the same

Corner response R is invariant to image rotation

Not invariant to image scale!

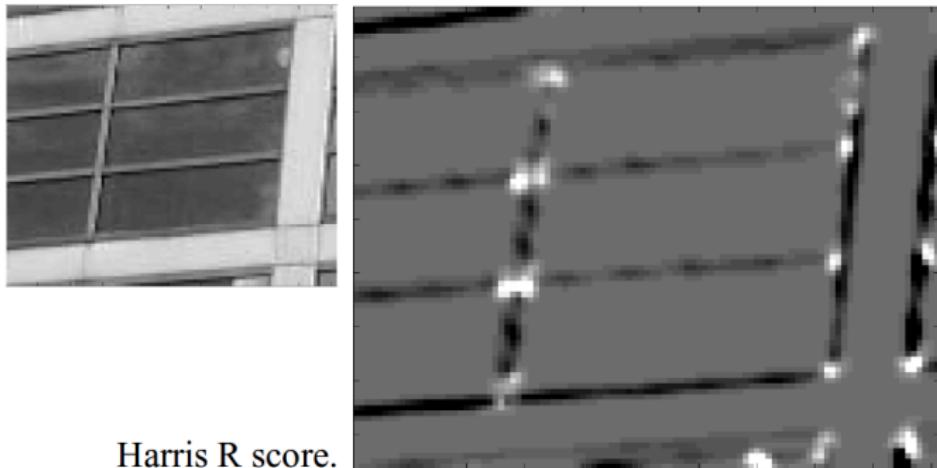
- But: non-invariant to *image scale*!



All points will be
classified as edges

Corner !

Harris Detector: Results

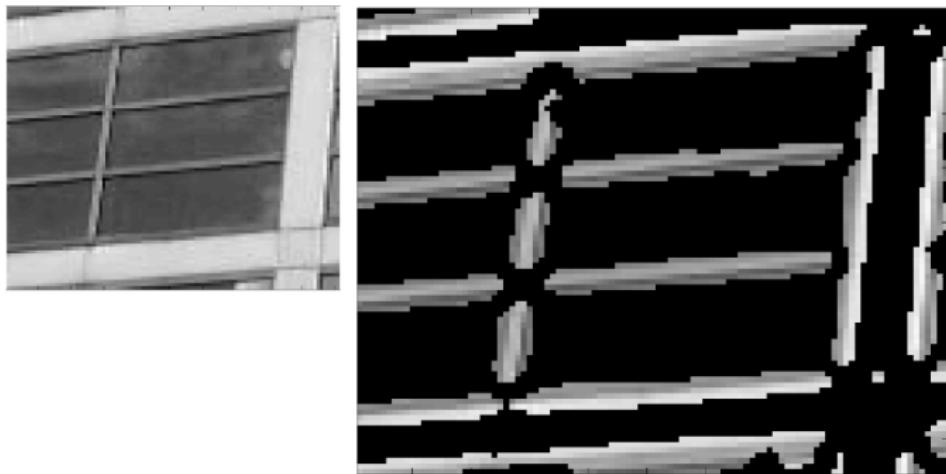


Harris R score.

I_x, I_y computed using Sobel operator

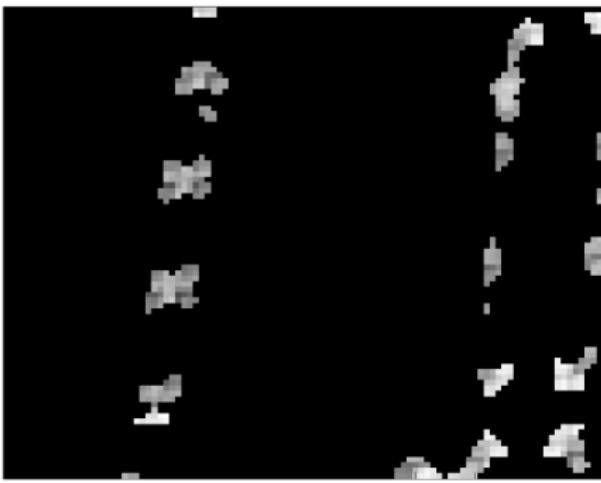
Windowing function $w = \text{Gaussian}$, $\sigma=1$

Harris Detector: Results



Threshold: $R < -10000$
(edges)

Harris Detector: Results



Threshold: > 10000
(corners)

Harris Detector: Algorithm Summary

1. Compute x and y derivatives of image

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

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x \cdot I_x \quad I_{y2} = I_y \cdot I_y \quad I_{xy} = I_x \cdot I_y$$

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

$$S_{x2} = G_{\sigma t} * I_{x2} \quad S_{y2} = G_{\sigma t} * I_{y2} \quad S_{xy} = G_{\sigma t} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x, y) = \begin{bmatrix} S_{x2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{y2}(x, y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = \text{Det}(H) - k(\text{Trace}(H))^2$$

6. Threshold on value of R. Compute nonmax suppression.