

Example: Harris Corner Detector

1. Problem Statement

Apply the Harris corner detector to the following binary image I of size 5×5 , which contains a bright 3×3 block on a dark background (Table 1). Pixels are denoted (r, c) with r the row and c the column, both starting at 0. The block has a corner at pixel $(2, 2)$, which we select to carry out the step-by-step calculation.

Table 1: The binary input image I .

	c=0	c=1	c=2	c=3	c=4
r=0	0	0	0	0	0
r=1	0	0	0	0	0
r=2	0	0	1	1	1
r=3	0	0	1	1	1
r=4	0	0	1	1	1

The following choices are made to keep the numbers simple:

- derivatives by central differences **without the factor** $\frac{1}{2}$: $I_x(r, c) = I(r, c + 1) - I(r, c - 1)$ and $I_y(r, c) = I(r + 1, c) - I(r - 1, c)$; the pixels outside the image are replaced by the nearest border pixel (replication);
- window w : a 3×3 box (all weights equal to 1), centred on the pixel; the pixels of the window that fall outside the image are ignored (zero contribution);
- sensitivity constant $k = 0.04$.

2. Principle

The Harris detector measures how much the grey levels of a small window change when it is shifted by (u, v) in any direction. A first-order Taylor expansion shows that this change is a quadratic form of the **structure tensor** M , built from the image derivatives summed over the window:

$$\begin{aligned} E(u, v) &\approx (u \quad v) M \begin{pmatrix} u \\ v \end{pmatrix}, & M &= \sum_{(r,c) \in w} \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \\ &= \begin{pmatrix} A & C \\ C & B \end{pmatrix} \end{aligned} \quad (1)$$

The two eigenvalues λ_1 and λ_2 of M describe the change in the two principal directions: both small means a flat region, one large and one small means an edge, both large means a corner. To avoid computing the eigenvalues, the Harris response is used:

$$R = \det(M) - k [\text{trace}(M)]^2 = (AB - C^2) - k (A + B)^2 \quad (2)$$

A large positive R indicates a corner, a negative R an edge, and R close to 0 a flat region.

3. Step 1: Image Derivatives

The derivatives are computed at every pixel. For the selected pixel (2,2):

$$I_x(2,2) = I(2,3) - I(2,1) = 1 - 0 = 1 \quad (3)$$

$$I_y(2,2) = I(3,2) - I(1,2) = 1 - 0 = 1 \quad (4)$$

The same calculation applied to all the pixels gives the two maps of Tables 2 and 3. The horizontal derivative is non-zero only along the vertical edge of the block (columns 1 and 2), and the vertical derivative only along its horizontal edge (rows 1 and 2).

Table 2: Horizontal derivative I_x .

	c=0	c=1	c=2	c=3	c=4
r=0	0	0	0	0	0
r=1	0	0	0	0	0
r=2	0	1	1	0	0
r=3	0	1	1	0	0
r=4	0	1	1	0	0

Table 3: Vertical derivative I_y .

	c=0	c=1	c=2	c=3	c=4
r=0	0	0	0	0	0
r=1	0	0	1	1	1
r=2	0	0	1	1	1
r=3	0	0	0	0	0
r=4	0	0	0	0	0

4. Step 2: Products of Derivatives in the Window of Pixel (2, 2)

The window of the selected pixel covers the rows 1 to 3 and the columns 1 to 3 (red frame of Figure 1a). We extract the derivatives in this window and compute the three products I_x^2 , I_y^2 and $I_x I_y$ pixel by pixel. The values are:

I_x in the window

	c=1	c=2	c=3
r=1	0	0	0
r=2	1	1	0
r=3	1	1	0

I_y in the window

	c=1	c=2	c=3
r=1	0	1	1
r=2	0	1	1
r=3	0	0	0

I_x^2 in the window

	c=1	c=2	c=3
r=1	0	0	0
r=2	1	1	0
r=3	1	1	0

I_y^2 in the window

	c=1	c=2	c=3
r=1	0	1	1
r=2	0	1	1
r=3	0	0	0

$I_x \cdot I_y$ in the window (only pixel (2,2) has both derivatives non-zero)

	c=1	c=2	c=3
r=1	0	0	0
r=2	0	1	0
r=3	0	0	0

5. Step 3: Sums over the Window and Matrix M

Summing each product over the nine pixels of the window gives the elements of M :

$$A = \sum I_x^2 = 1 + 1 + 1 + 1 = 4 \quad (5)$$

$$B = \sum I_y^2 = 1 + 1 + 1 + 1 = 4 \quad (6)$$

$$C = \sum I_x I_y = 1 \cdot 1 = 1 \quad (7)$$

$$M = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} \quad (8)$$

6. Step 4: Harris Response at Pixel (2, 2)

$$\det(M) = AB - C^2 = 4 \cdot 4 - 1^2 = 15, \quad \text{trace}(M) = A + B = 8 \quad (9)$$

$$R = 15 - 0.04 \cdot 8^2 = 15 - 2.56 = 12.44 \quad (10)$$

As a check, the eigenvalues of M are $\lambda_1 = 5$ and $\lambda_2 = 3$ (the roots of $\lambda^2 - 8\lambda + 15 = 0$): both are large and positive, and $\lambda_1 \lambda_2 = 15 = \det(M)$. Since $R = 12.44$ is large and positive, the pixel (2,2) is a **corner**.

To see the difference with other types of pixels, the same operations are applied to an edge pixel and a flat pixel (Table 4). For the edge pixel (4,2), the window (rows 3 to 5, columns 1 to 3) contains only the vertical edge of the block: $I_x = 1$ at four pixels and $I_y = 0$ everywhere, so $A = 4$, $B = 0$, $C = 0$. For the flat pixel (0,0), all the derivatives of the window are zero.

Table 4: Harris computation for three pixels ($k = 0.04$).

Pixel	A	B	C	$\det(M)$	$\text{trace}(M)$	R	Decision
(2,2)	4	4	1	15	8	+12.44	Corner
(4,2)	4	0	0	0	4	-0.64	Edge
(0,0)	0	0	0	0	0	0	Flat

7. Step 5: Response Map, Threshold and Corner Detection

Applying the same procedure to the 25 pixels gives the response map of Table 5 (Figure 1b). The largest response is at (2,2), the corner of the block; the pixels next to the corner also have a positive response because their window overlaps the corner, while the pixels far from the block have $R \approx 0$ and the pixels that see only a straight edge have a small negative response.

Table 5: Harris response R for all the pixels.

	c=0	c=1	c=2	c=3	c=4
r=0	0	-0.04	-0.16	-0.36	-0.16
r=1	-0.04	2.36	5.56	3.04	-0.64
r=2	-0.16	5.56	12.44	8.44	-0.64
r=3	-0.36	3.04	8.44	6.56	-0.16
r=4	-0.16	-0.64	-0.64	-0.16	0

Two final operations isolate the corners:

1. **Thresholding:** keep the pixels with $R > T$, with for example $T = 0.5 R_{max} = 0.5 \times 12.44 = 6.22$. Four pixels remain: (2,2) with 12.44, (2,3) and (3,2) with 8.44, and (3,3) with 6.56.
2. **Non-maximum suppression:** keep a pixel only if its response is the largest in its 3×3 neighbourhood. The pixels (2,3), (3,2) and (3,3) have the neighbour (2,2) with a larger response and are discarded.

The only remaining pixel is (2,2): the detector finds **one corner at pixel (2, 2)**, which is the top-left corner of the bright block (Figure 1c).

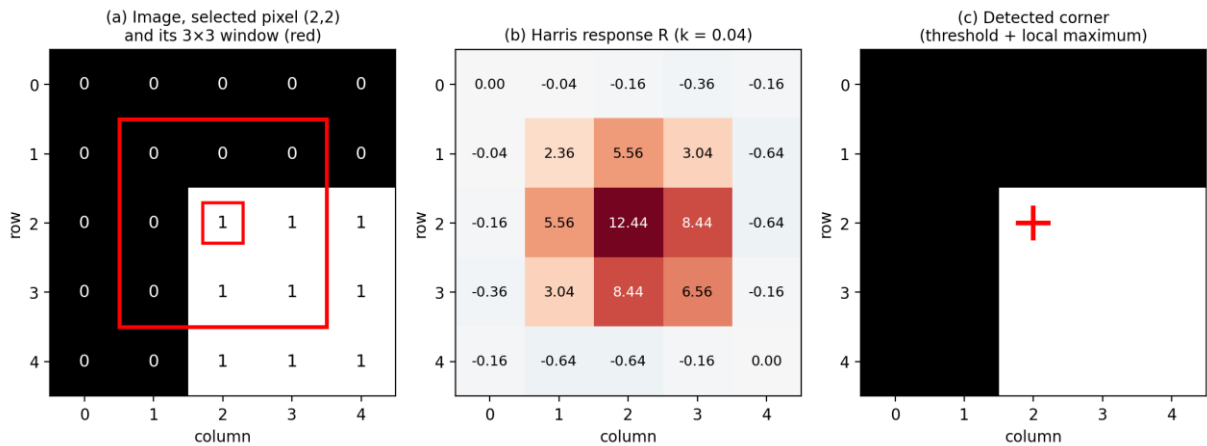


Figure 1: Selected pixel and window (a), Harris response map (b) and detected corner (c).

Remarks

- **Scale of the values:** using the factor $\frac{1}{2}$ in the derivatives would divide A , B , C by 4 and R by 16; the classification (sign of R and position of the maximum) would not change, only the value of the threshold.
- **Other corners of the block:** the other three corners of the block lie on the border of the image, where the replication of the border pixels removes the intensity change; the detector therefore only sees the corner (2,2) in this small image.
- **Role of k :** the constant k (typically 0.04 to 0.06) sets the balance between corner and edge responses: the larger k , the more negative R is along edges.

Table 6: Summary of the steps of the example.

Step	Operation	Result for pixel (2,2)
1	Derivatives I_x, I_y	$I_x = 1, I_y = 1$ (maps in Tables 2 and 3)
2	Products $I_x^2, I_y^2, I_x I_y$ in the 3×3 window	Non-zero at 4, 4 and 1 pixels respectively
3	Window sums, matrix M	$A = 4, B = 4, C = 1$
4	Response $R = \det(M) - k \text{trace}(M)^2$	$R = 12.44$ (corner)
5	Threshold and non-maximum suppression	Corner detected at (2,2)