Image Features (Part II)

Goal: We introduce and motivate several types of image features. These are each designed for specific vision tasks.

We consider features to support the following tasks:

I. Matching image patches between images with significantly different viewpoints,

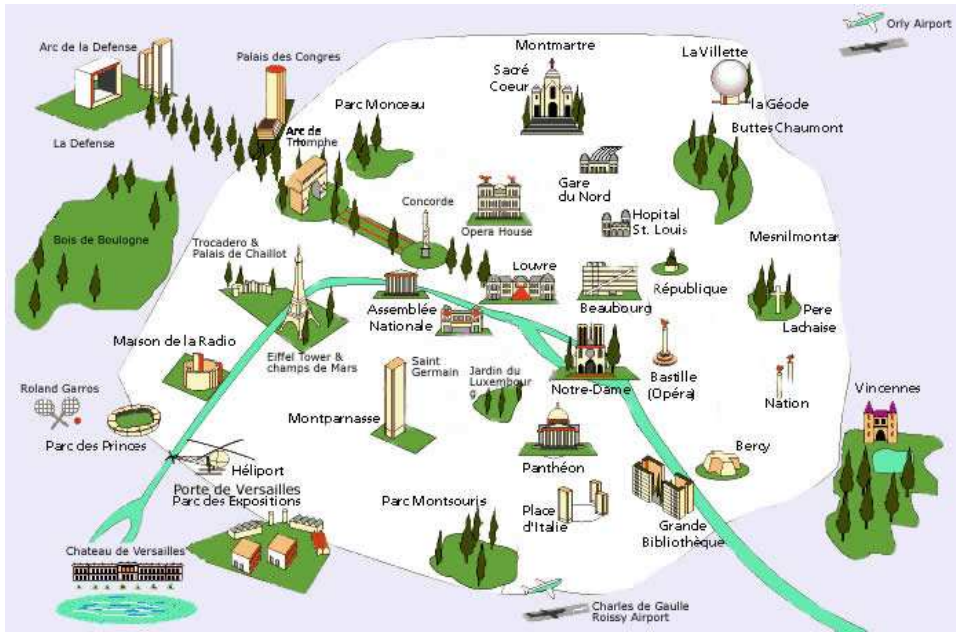
II. Extracting image landmarks; a) their (x, y) position, $\Leftarrow$ **Today**

III. Extracting image landmarks; b) their scale, and c) their orientation.

Readings: Szeliski, Section 4.1 and 4.2.

Part II: Image Landmarks

A **landmark** is defined as an point of reference that helps orienting in a familiar or unfamiliar environment (paraphrased from Wikipedia). By "orienting" oneself we mean answering both where you are and which direction you are facing.



An **image landmark** is similar. Given a set of pictures of the same object, an image landmark is designed to help orient a vision system with respect to the content of the image.

Ideally, an image landmark should identify the image centers (x, y) of corresponding image patches in different images, along with their image orientations and scales.

Design Criteria of Image Landmarks

When we move about in new surroundings we **choose** to remember certain places, to help us orient ourselves later.

Similarly, our task here is to **choose** specific image landmarks. We seek landmarks that are:

- distinctive,
- specific in terms of image location, orientation, and scale,
- repeatedly identifiible (i.e., across many images, viewpoints, etc.),
- commonly occuring in images (i.e., we want lots of good landmarks).

Our image landmarks will be at a much more basic level than using views of the CN tower to help navigate around Toronto. (Although, we may be able to use sets of these image landmarks to reliably detect the CN tower.)

We next describe one popular method for selecting (x, y)-locations for image landmarks. The image scale and orientation of landmarks will be dealt with later.

Image Landmark Locations

Good image landmarks should be reliably localized across different views and lighting conditions.

We describe the Harris/Forstner interest point. (For other options, see Corner detection in Wikipedia.)

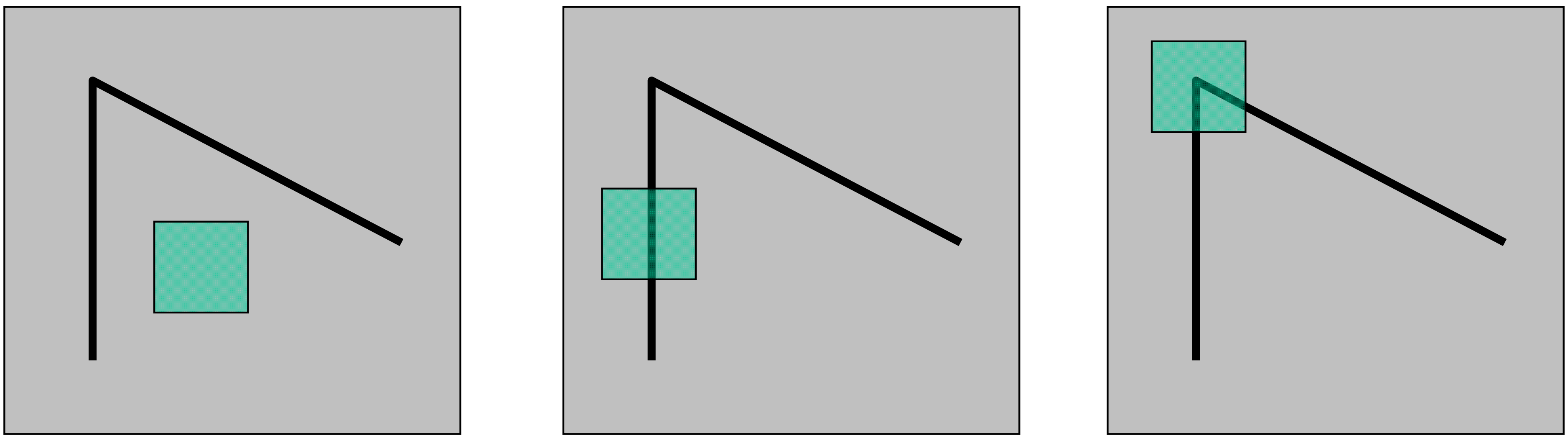
Harris/Forstner provides a localization in pixel coordinates only, we address localization in scale and orientation separately.

Slides adapted (slightly) from those of Darya Frolova and Denis Simakov, Weizmann Institute, and also from those of Steve Seitz and Rick Szeliski, University of Washington.

Local measures of uniqueness

Suppose we only consider a small window of pixels centered on a chosen point (i.e., an image patch).

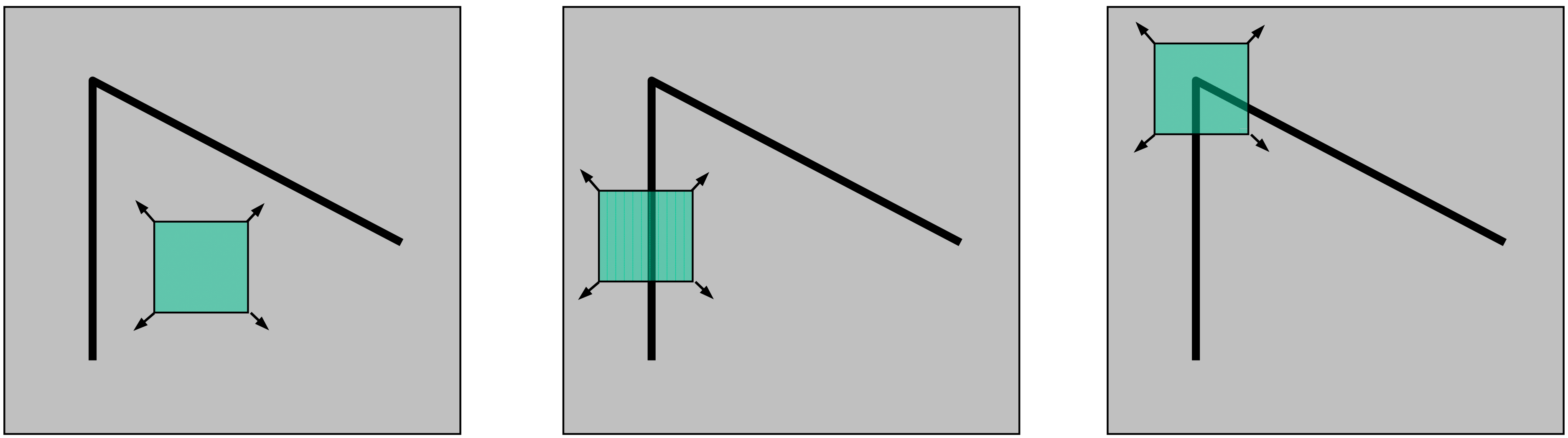
- What defines whether a patch is a good or bad candidate?



Interest point detection

Local measure of the uniqueness of the image patch

- How does the window change when you shift it?
- Looking for a *big change* when shifted in *any* direction.



"flat" region:
no change in all directions

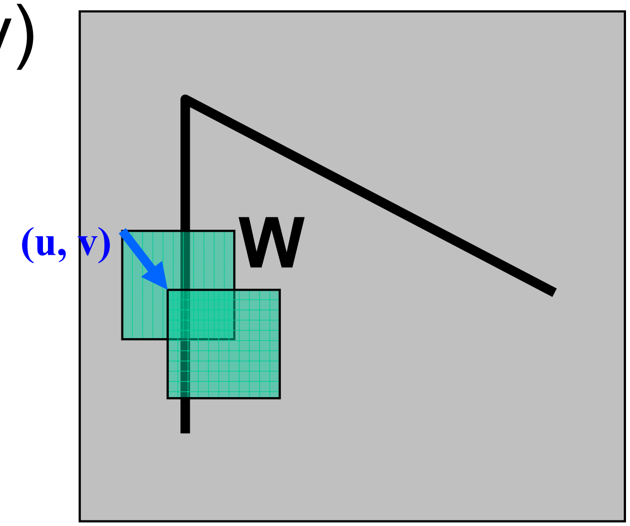
"edge":
no change along the edge direction

"corner":
significant change in all directions

Interest point detection: the math

Consider shifting the window **W** by (u,v)

- how do the pixels in **W** change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD "error" of *E(u,v)*, for a patch centered at (x_0, y_0).

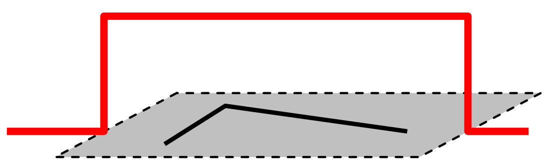


$$E(u,v) = \sum_{(x,y)\in W} w(x-x_0, y-y_0)\left[I(x+u,y+v) - I(x,y)\right]^2$$

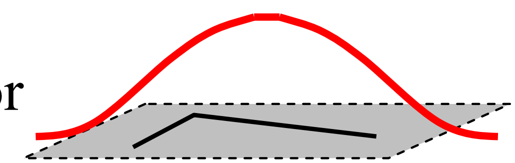
$$w(\vec{x}-\vec{x}_0)$$

Here w(x,y) is a window function centered at (0,0), e.g., a Gaussian.

$w(x,y) =$



or



1 in window, 0 outside

Gaussian

Small motion assumption

Taylor Series expansion of I:

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

If the motion (u,v) is small, then first order approx is good

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x,y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

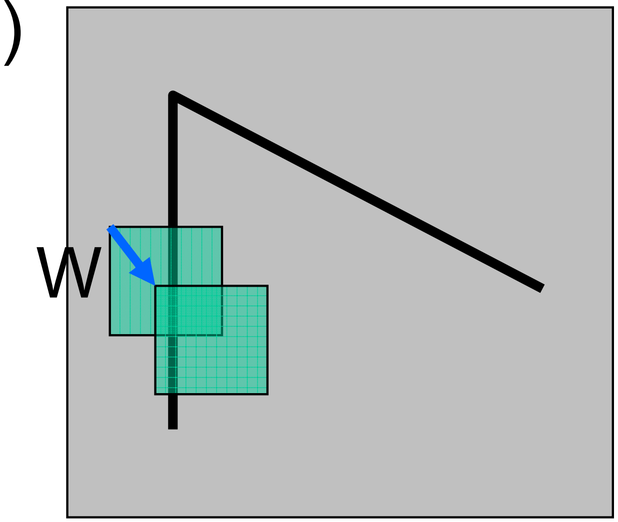
shorthand: $I_x = \frac{\partial I}{\partial x}$

Plugging this into the formula on the previous slide…

Interest point detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an "error" of E(u,v):



$$E(u,v) = \sum_{(x,y)\in W} w(\vec{x}-\vec{x}_0)[I(x+u,y+v) - I(x,y)]^2$$

$$\approx \sum_{(x,y)\in W} w(\vec{x}-\vec{x}_0)\left[I(x,y) + [I_x \;\; I_y]\begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)\right]^2$$

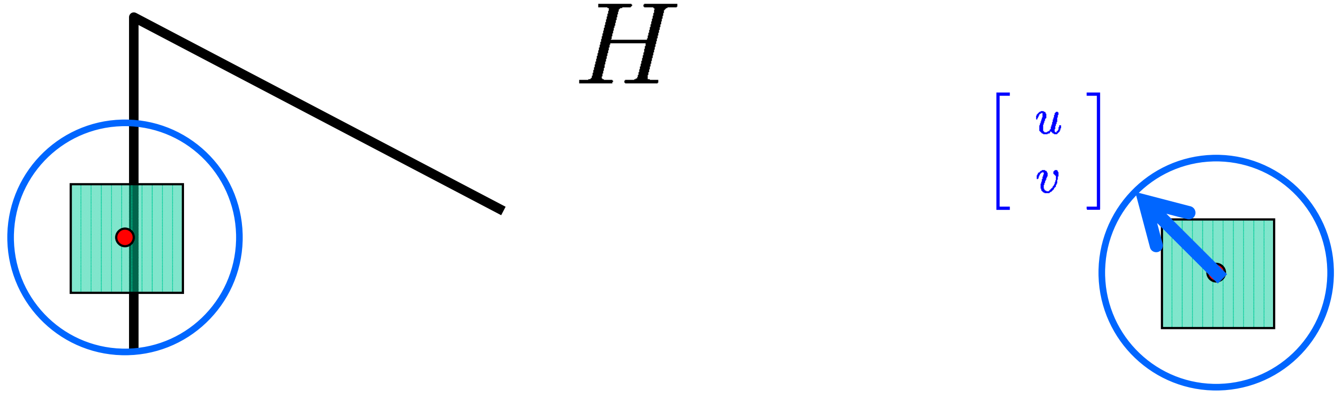
$$= \sum_{(x,y)\in W} w(\vec{x}-\vec{x}_0)\left[[I_x \;\; I_y]\begin{bmatrix} u \\ v \end{bmatrix}\right]^2$$

Interest point detection: the math

This can be rewritten:

$$E(u,v) = [u\ v] \left\{ \sum_{(x,y)\in W} w(\vec{x} - \vec{x}_0) \left[\begin{array}{cc} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{array} \right] \right\} \left[\begin{array}{c} u \\ v \end{array} \right]$$

$$H$$



For the example above

- You can move the center of the green window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest values of E?
- We can find these directions by looking at the eigenvectors of ***H***

Quick eigenvalue/eigenvector review

The **eigenvectors** of a matrix **A** are the vectors **u** that satisfy:

$$Au = \lambda u$$

The scalar λ is the **eigenvalue** corresponding to **u.**

The eigenvalues are found by solving:

$$det(A - \lambda I) = 0$$

- In our case, ***A*** = ***H*** is a 2x2 matrix, so we have

$$det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$$

- The solution:

$$\lambda_{\pm} = \frac{1}{2} \left[(h_{11} + h_{22}) \pm \sqrt{4 h_{12} h_{21} + (h_{11} - h_{22})^2} \right]$$

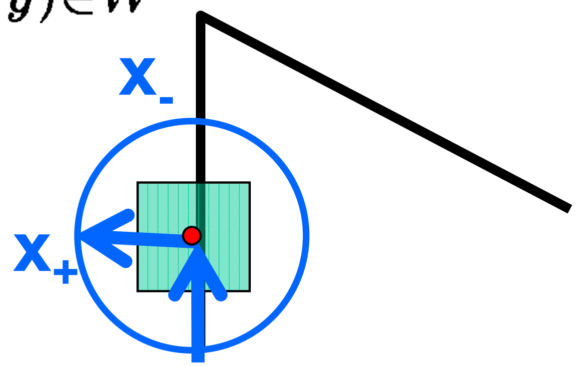
Once you know λ, you find **u** by solving

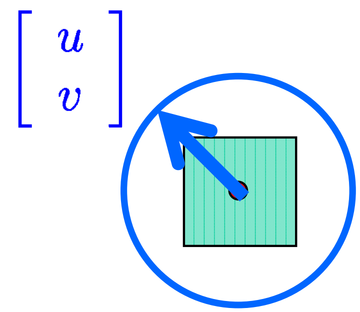
$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

Interest point detection: the math

$$E(u,v) = [u\ v]H \begin{bmatrix} u \\ v \end{bmatrix},$$

$$H \equiv \sum_{(x,y)\in W} w(\vec{x} - \vec{x}_0) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}.$$





Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- u_+ = direction of **largest** increase in E.
- λ_+ = amount of increase in direction u_+
- u_- = direction of **smallest** increase in E.
- λ- = amount of increase in direction u_+

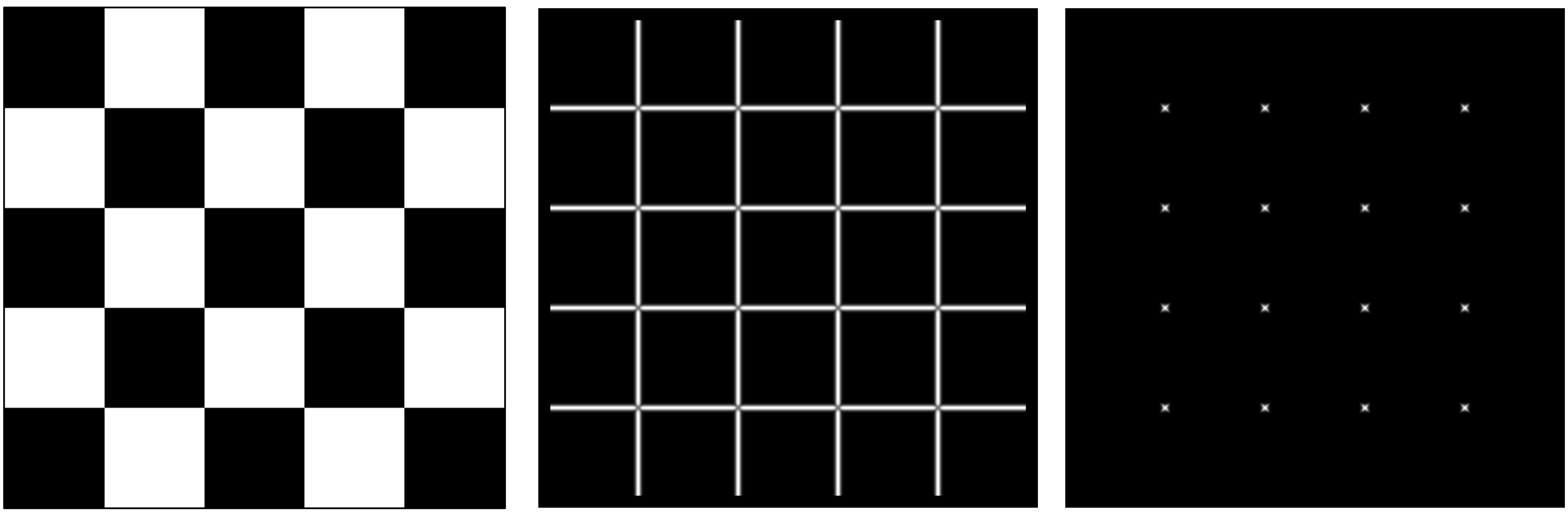
$$\begin{aligned} Hu_+ &= \lambda_+ u_+ \\ Hu_- &= \lambda_- u_- \end{aligned}$$

Interest point detection: the math

How are λ_+, $\mathbf{u}_+$, λ_-, and $\mathbf{u}_-$ relevant for point detection?

Want *E(u,v)* to be ***large*** for small shifts in ***all*** directions

- the *minimum* of *E(u,v)* should be large, over all unit vectors [u v]
- this minimum is given by the smaller eigenvalue (λ_-) of ***H***



I λ_+ λ_-

Interest point detection: the math

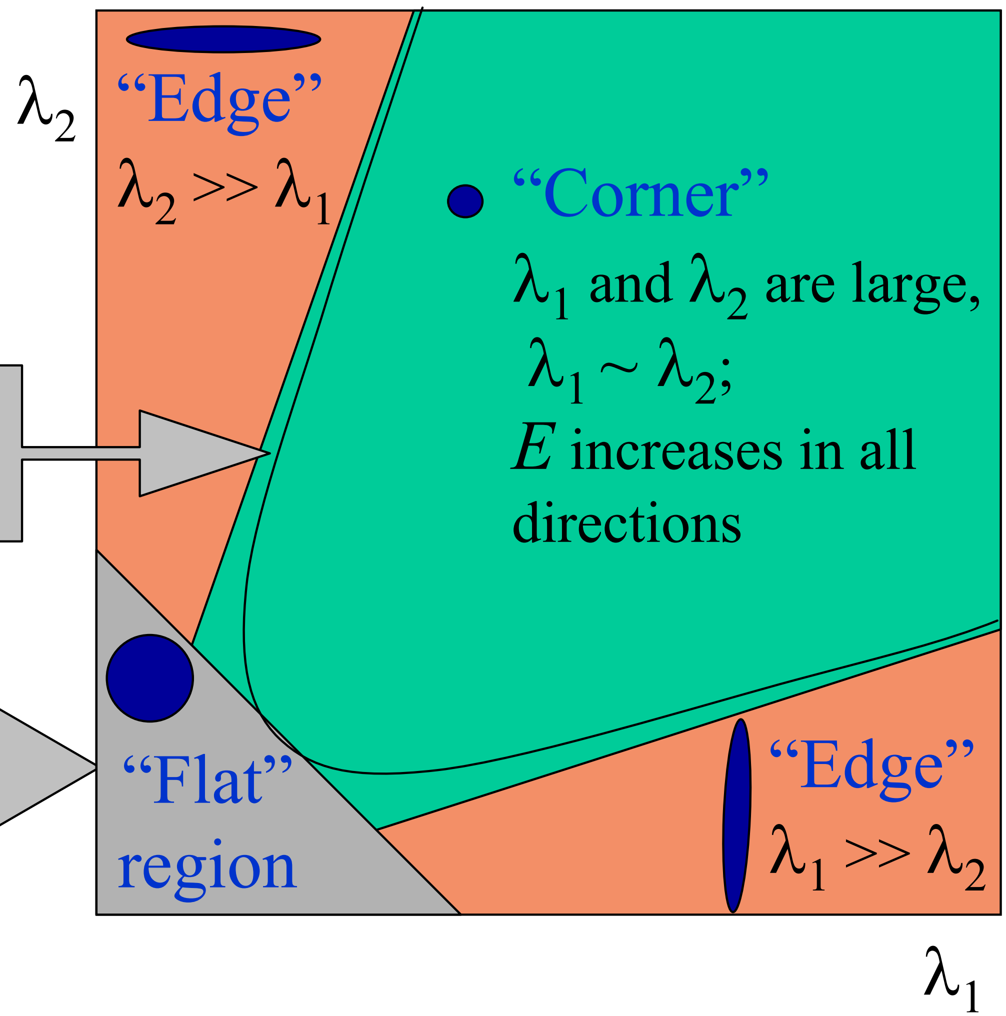
Classification of image points using eigenvalues of *H*:

Seek $\lambda_+ > \tau$, $\lambda_- / \lambda_+ > k'$ (e.g., k' = 0.05 for $\lambda_- > 5\% \lambda_+$).

Replacing $\lambda_\pm$ by λ_1 and λ_2 (in any order), we have:

Hyperbola:
$(\lambda_1 - k'\lambda_2)(\lambda_2 - k'\lambda_1) = c > 0$

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



The Harris operator

Criteria for interest point detection

$$(\lambda_1 - k'\lambda_2)(\lambda_2 - k'\lambda_1) = (1+k')^2\lambda_1\lambda_2 - k'(\lambda_1+\lambda_2)^2,$$
$$= (1+k')^2 det(H) - k'\left[trace(H)\right]^2 \geq c'.$$

The “Harris operator” for interest point detection

$$R \equiv det(H) - k\left[trace(H)\right]^2$$

- The *trace* is the sum of the diagonals, i.e., *trace(H)* = $h_{11} + h_{22}$
- R > c identical to $(\lambda_1 - k'\lambda_2)(\lambda_2 - k'\lambda_1)$ > c’, with k = k’/(1+k’)
- Called the “Harris Corner Detector” or “Harris Operator”
- Lots of other detectors, this is one of the most popular

Harris interest points defined to be local maxima of R with R > c.

Harris Interest Points: Workflow

- Compute image gradients (I_x, I_y).
- For each pixel (x_0,y_0), compute:

$$H(x_0, y_0) = \sum_{(x,y)\in W} w(\vec{x} - \vec{x}_0) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}.$$

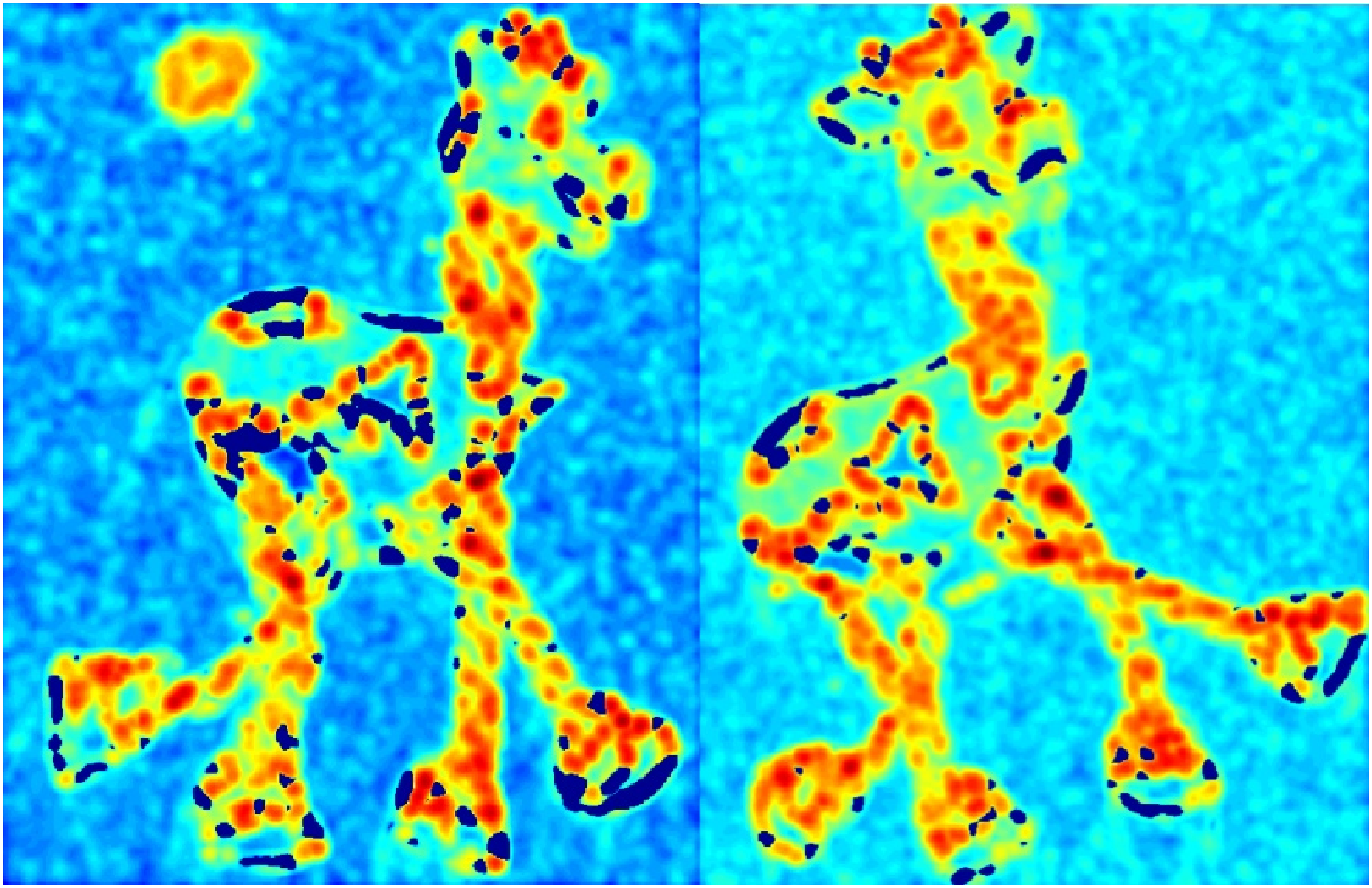
- Compute the Harris operator:

$$R(x,y) \equiv det(H(x,y) - k\,[trace(H(x,y)]^2$$

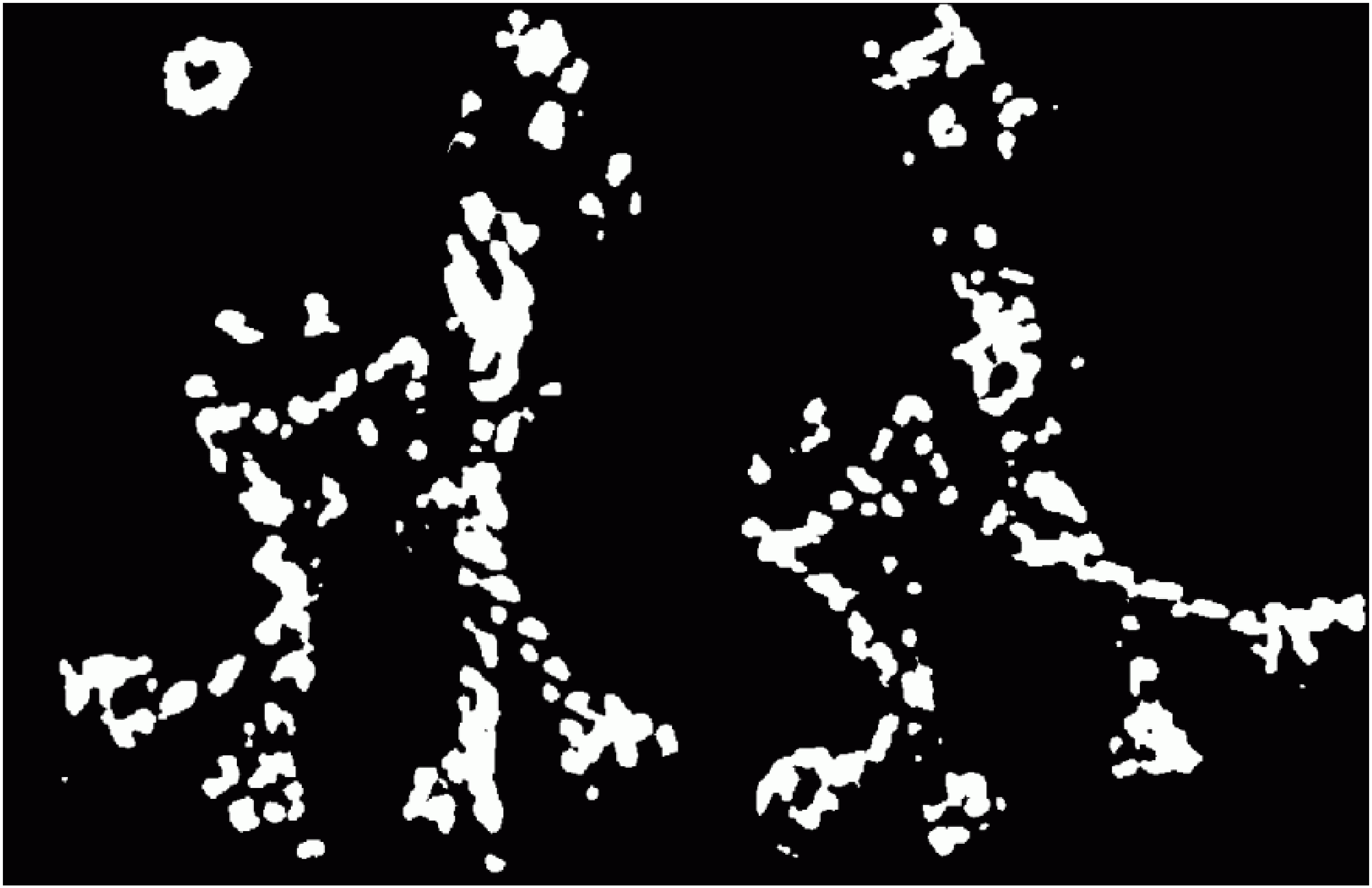
- Identify local maxima of R(x,y) with R > c.

Harris detector example

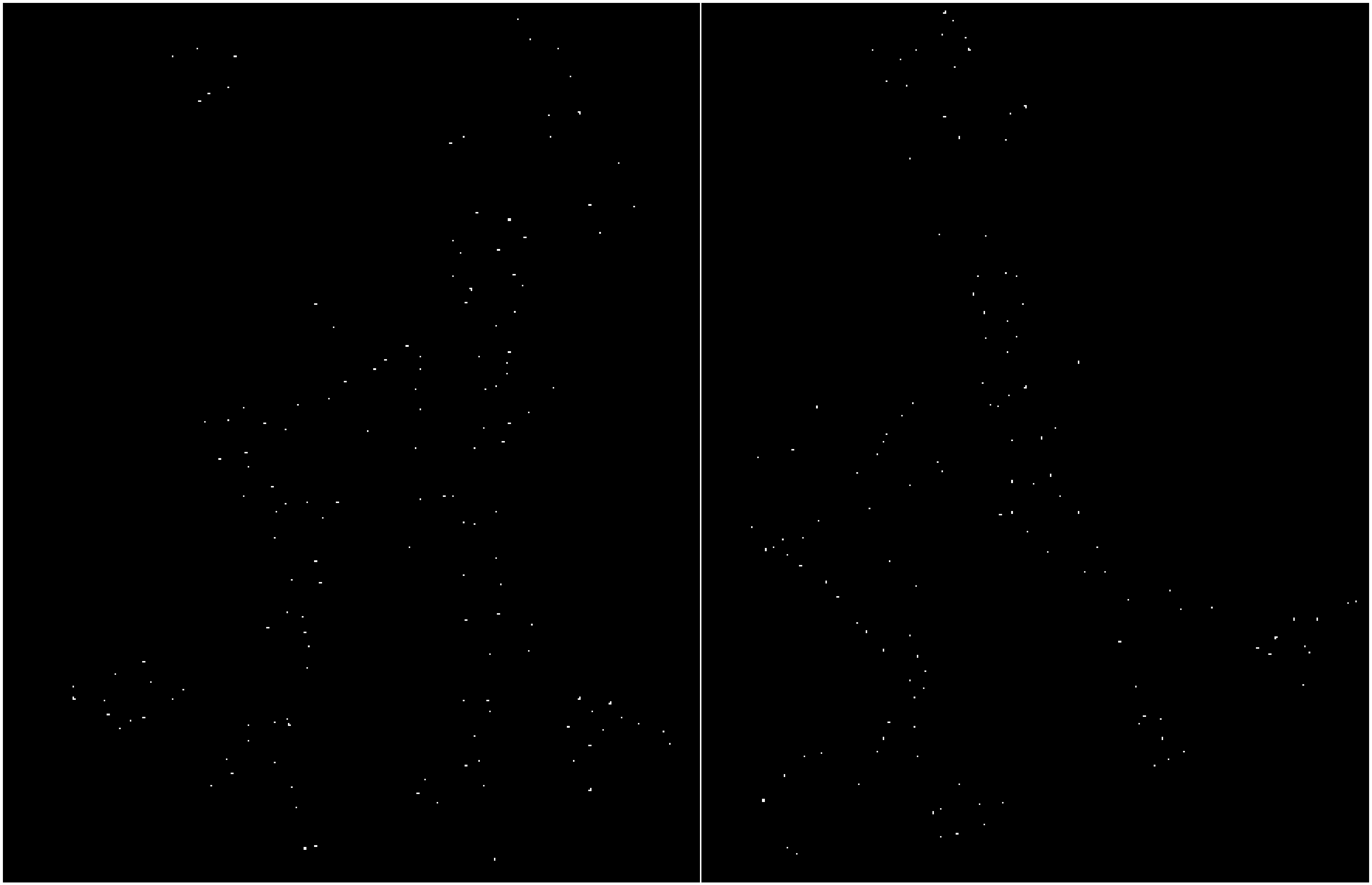
R value (red high, blue low)



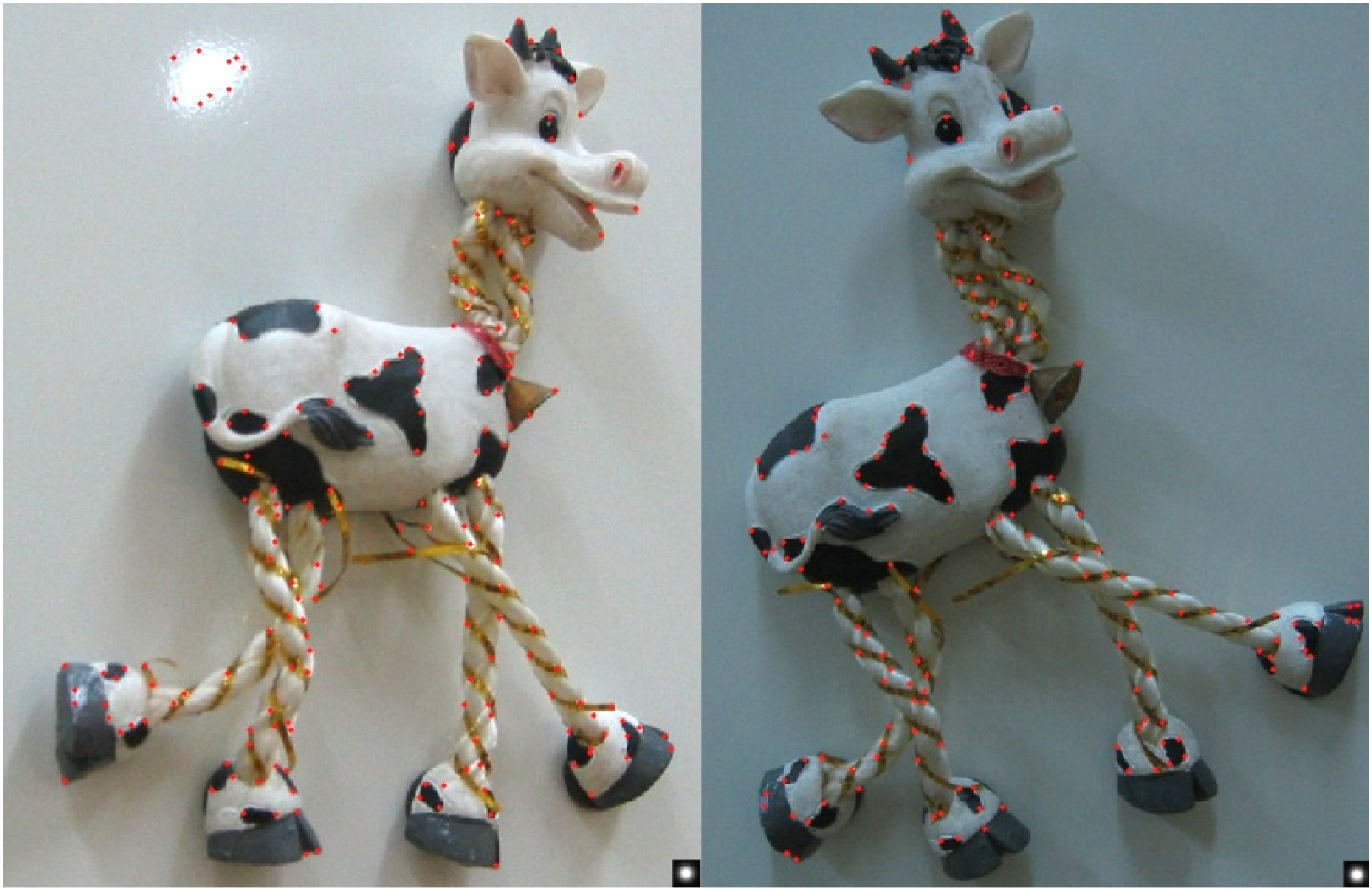
Threshold (R > value)



Find local maxima of R



Harris features (in red)



Invariance

Suspose you **rotate** the image by some angle

- Will you still pick up the same features?

What if you change the brightness?

Scale?

Viewpoint?

References

C. Harris and M. Stephens, A combined corner and edge detector, In Alvey Vision Conference, pp.147-151, 1988.

David G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008.

T. Tuytelaars and K. Mikolajczyk , Local Invariant Feature Detectors - Survey, Foundations and Trends in Computer Graphics and Vision , 3(3):177-280, 2008.

K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. In PAMI 27(10), pp.1615-1630, 2005.