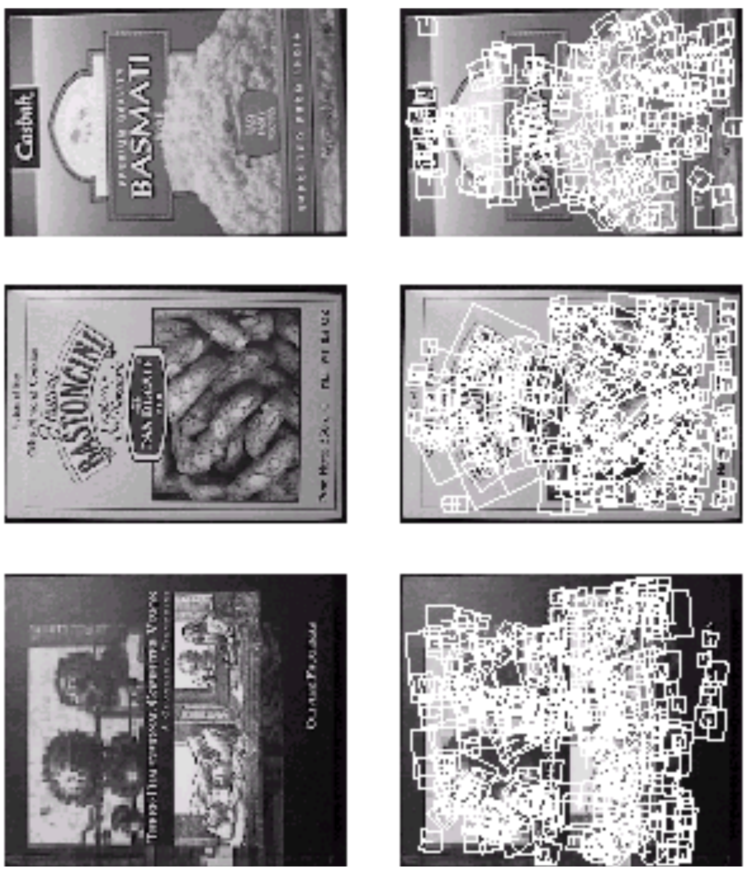
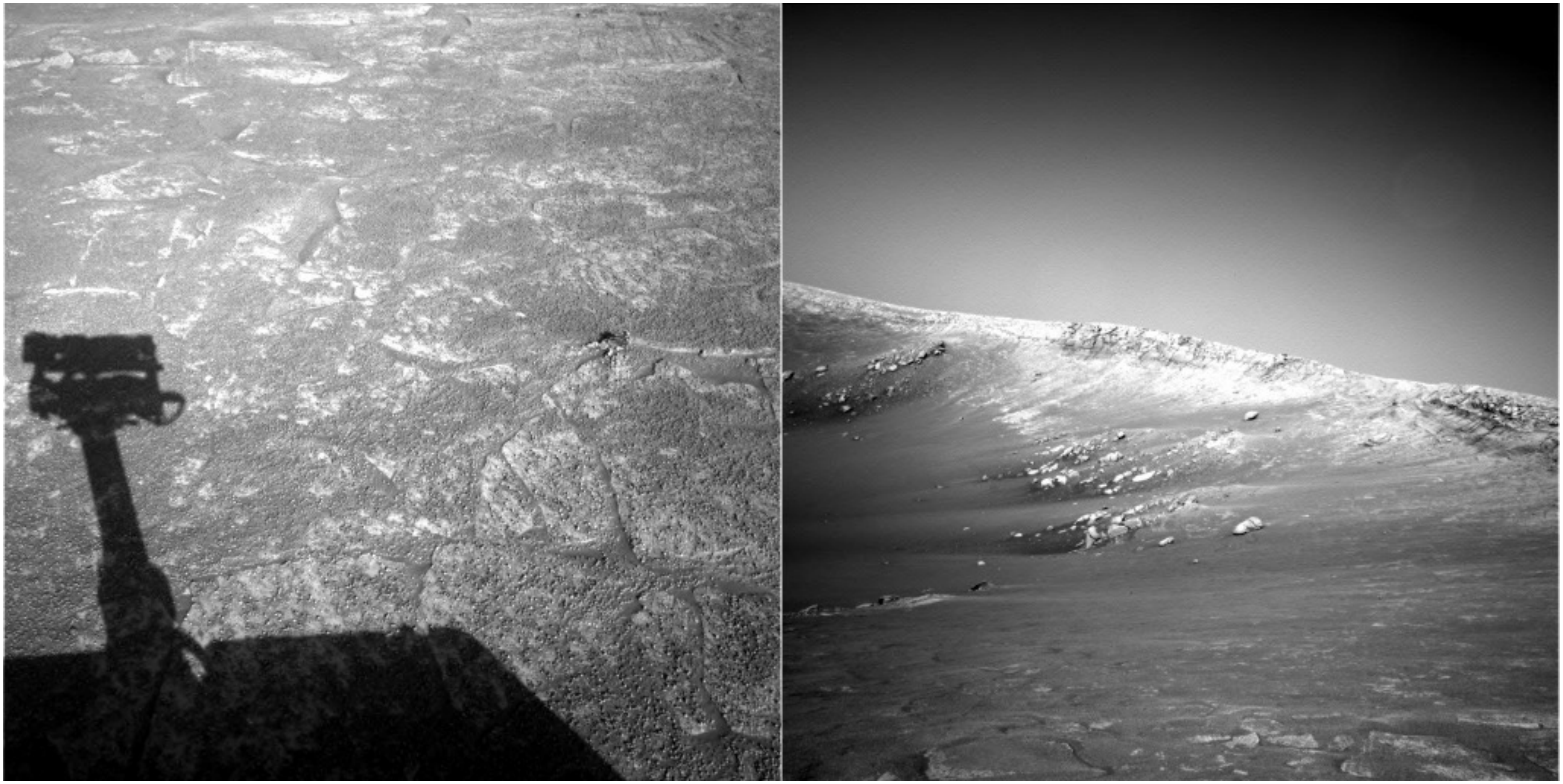
Feature Detection



Overview

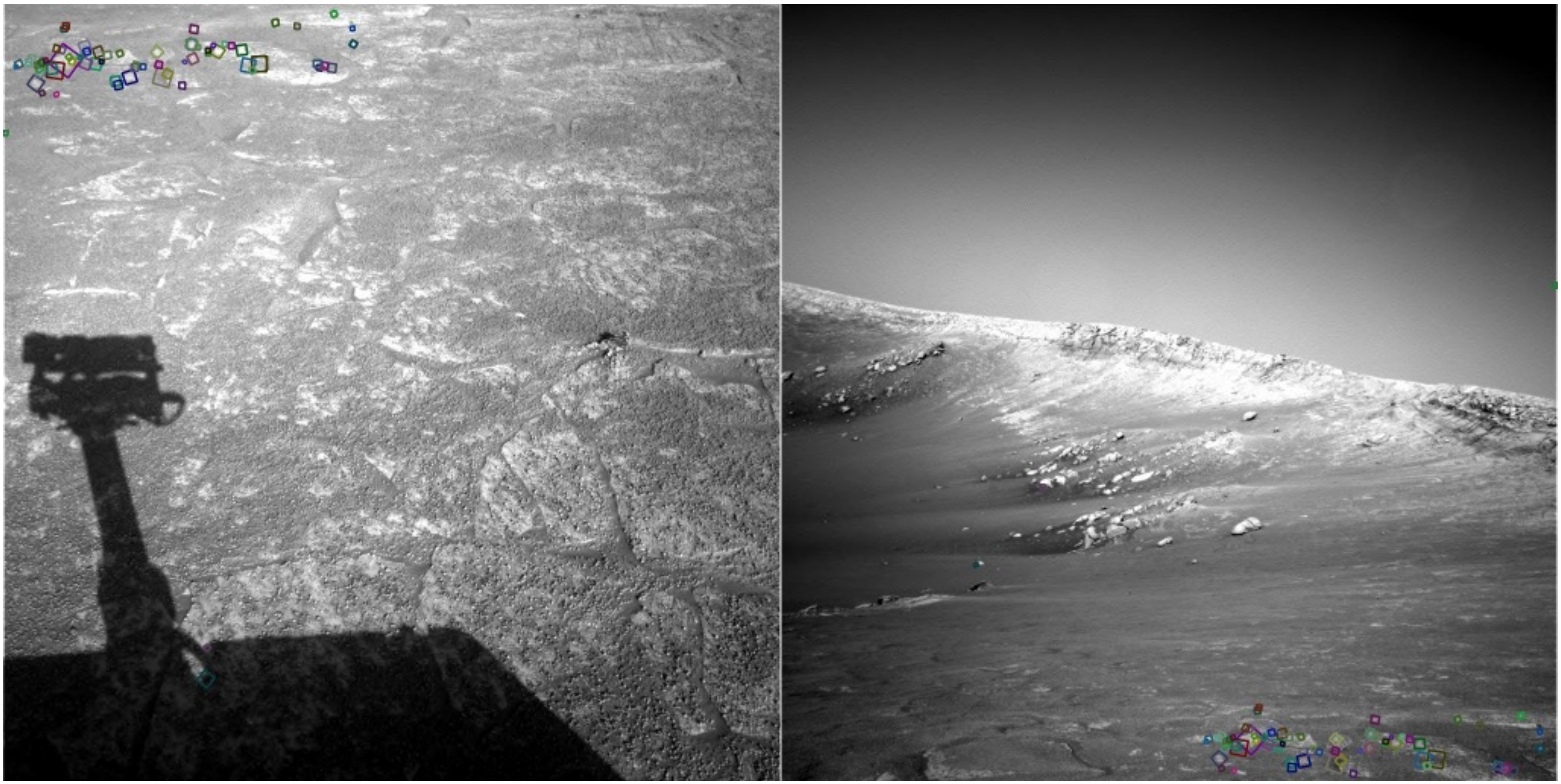
Classical approaches (before ConvNets):

- Why you want to find features
- Harris corner detector
- Multi-scale detection
- Multi-scale blob detection



NASA Mars Rover images

Where are the corresponding points?



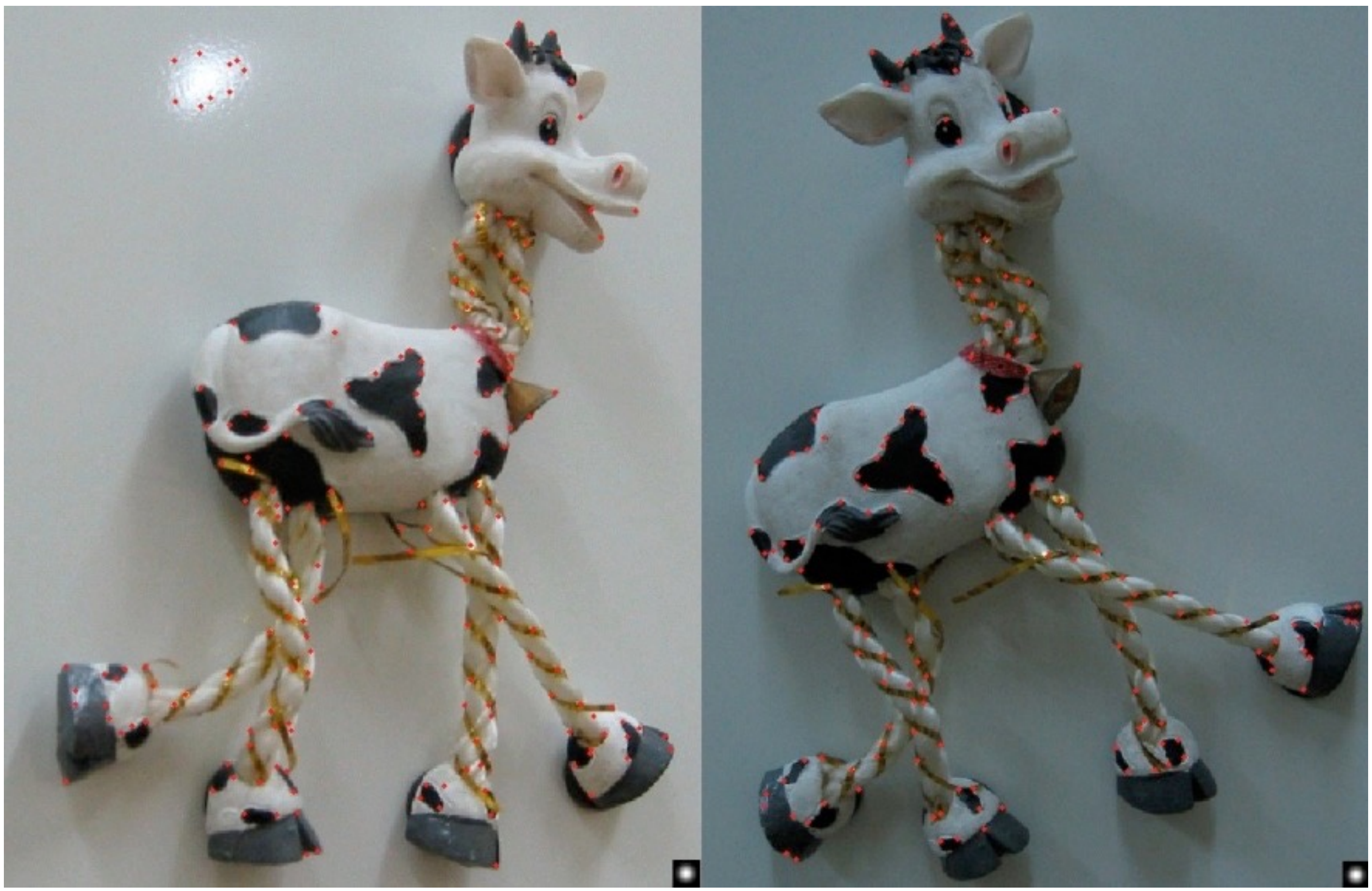
Challenges: Invariance

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …

Two Problems for Features

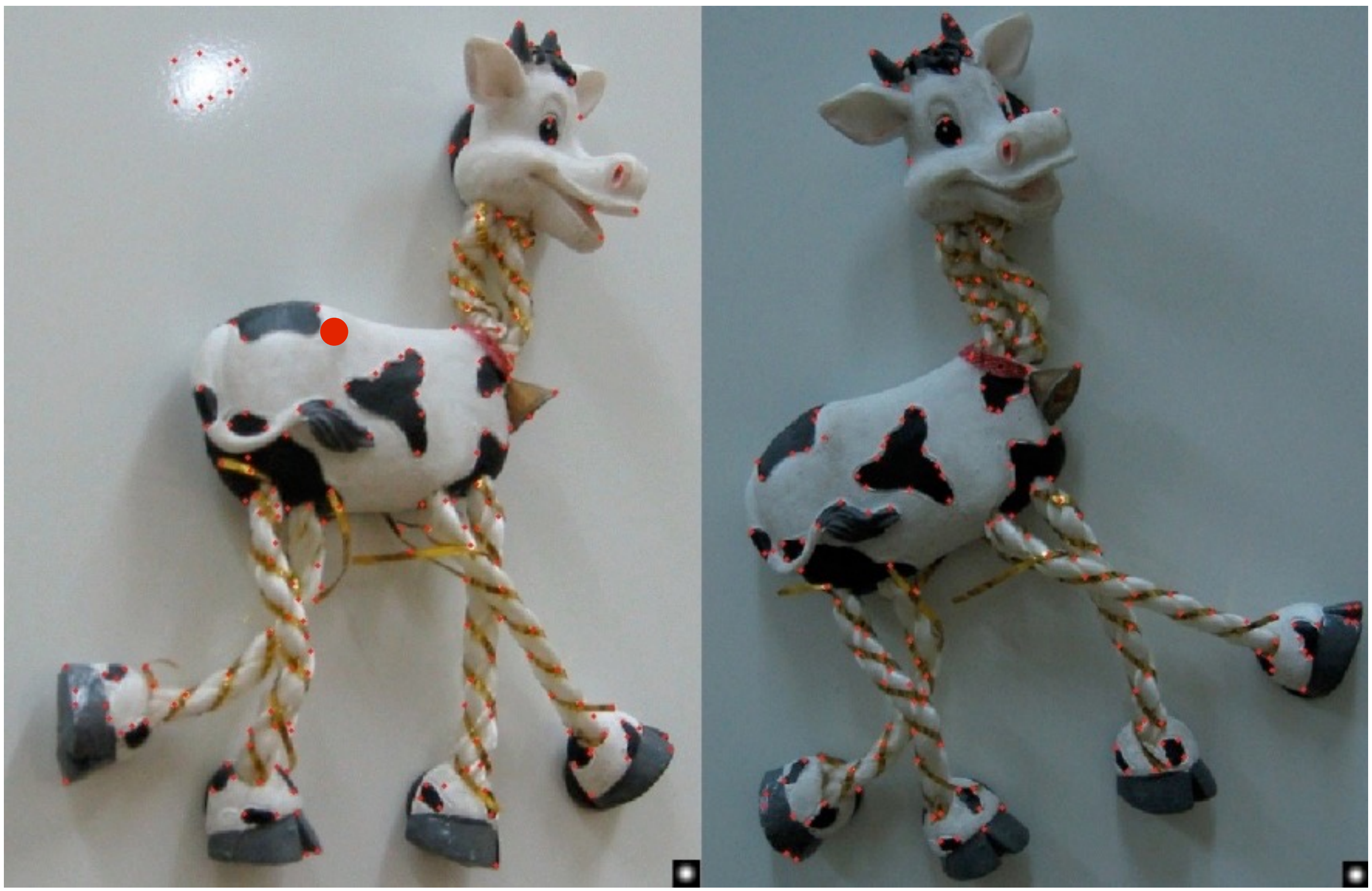
Feature detection



Two Problems for Features

Feature detection

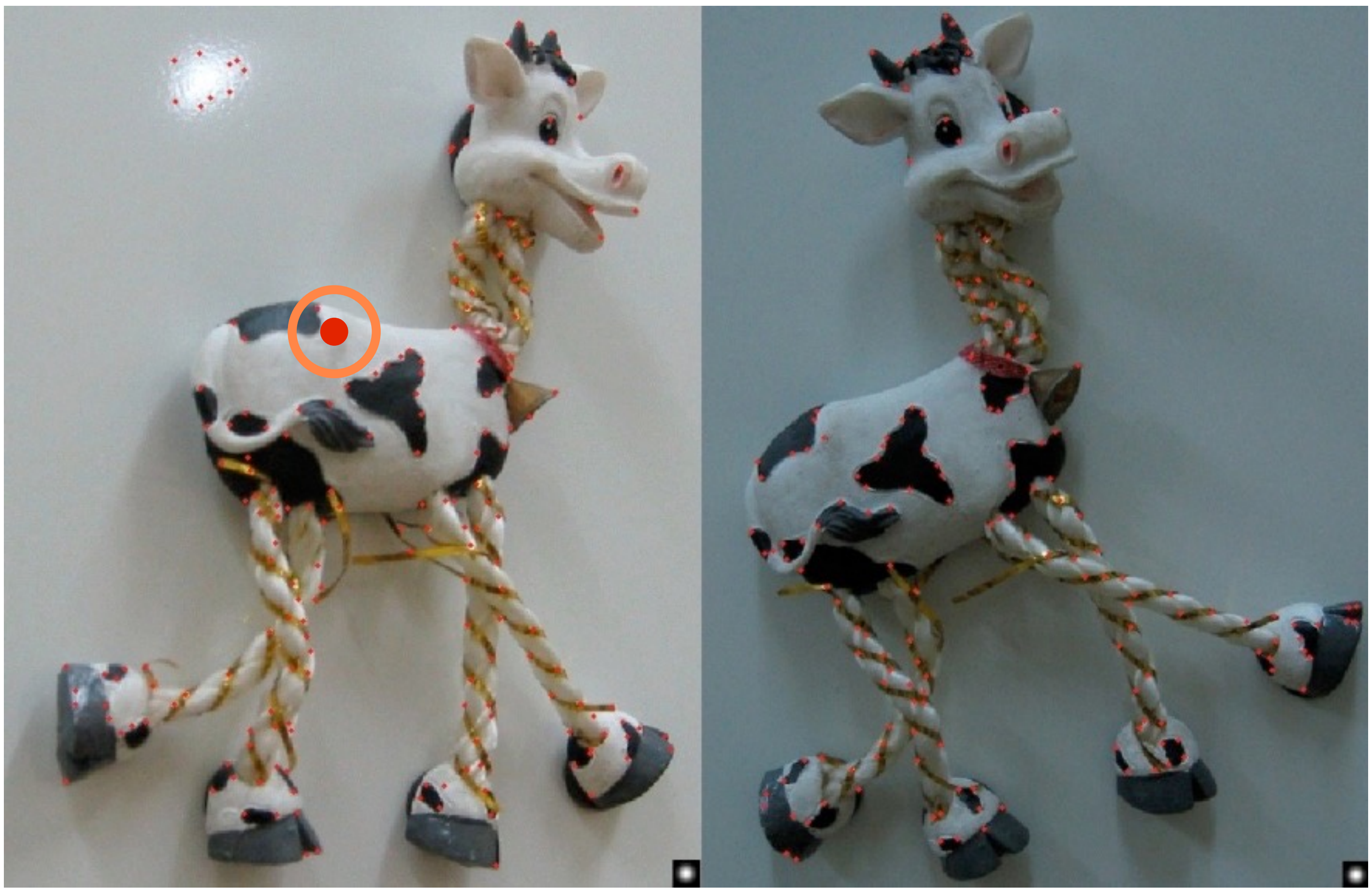
Feature descriptor



Two Problems for Features

Feature detection

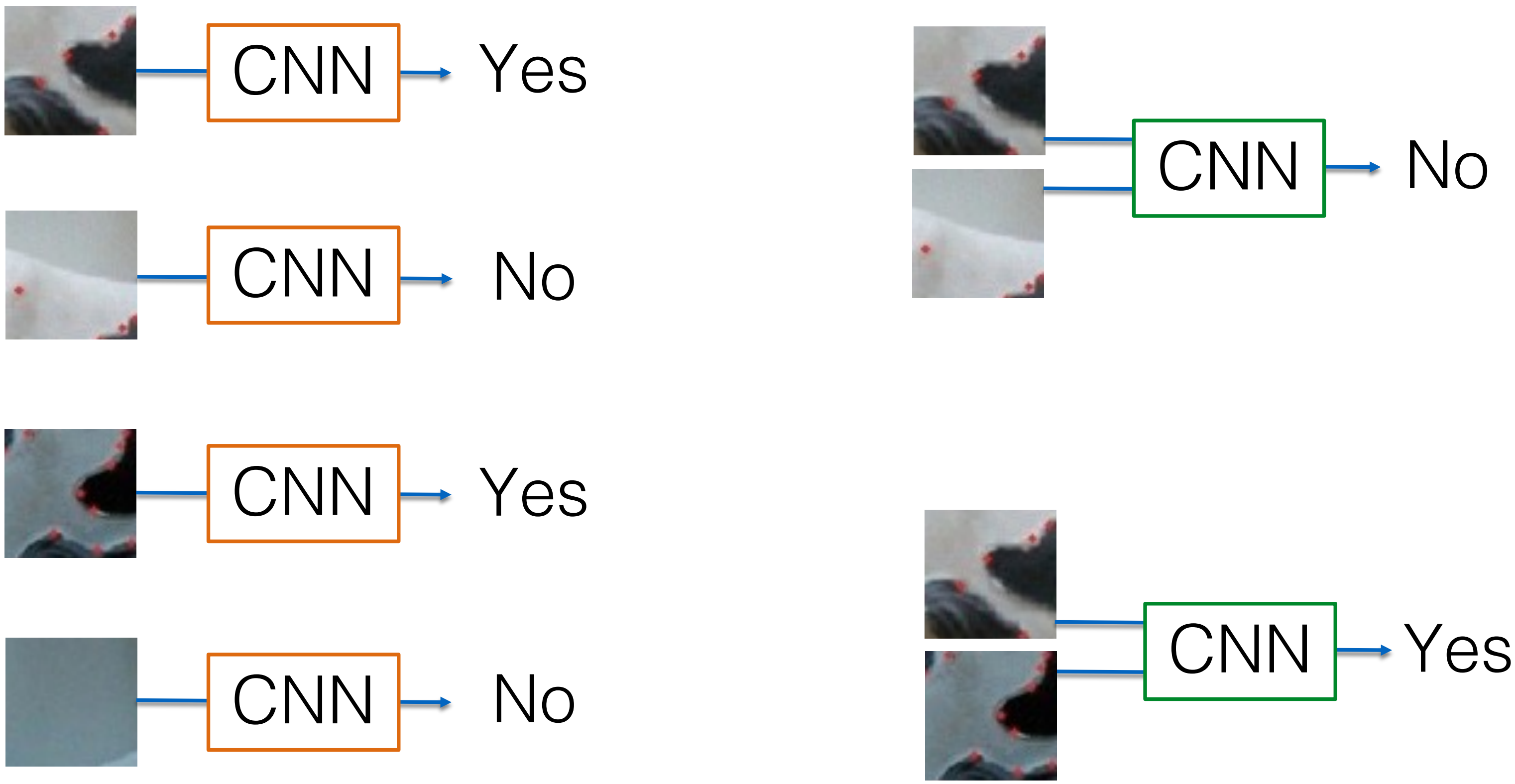
Feature descriptor



How do you solve
feature detection & matching in CNN

?

How do you solve feature detection & matching in CNN



*ignore the red dots in these images

What makes a good feature?

Want uniqueness

Look for unusual / unique image regions

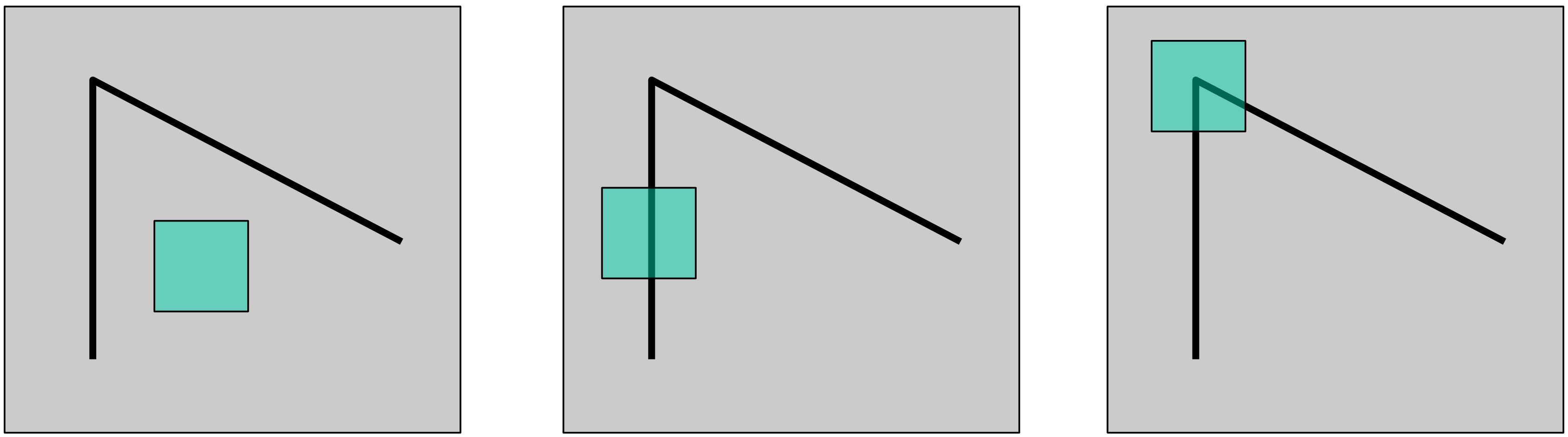
- Lead to unambiguous matches in other images

How to define "unusual"?

Local measures of uniqueness

Consider a small window of pixels

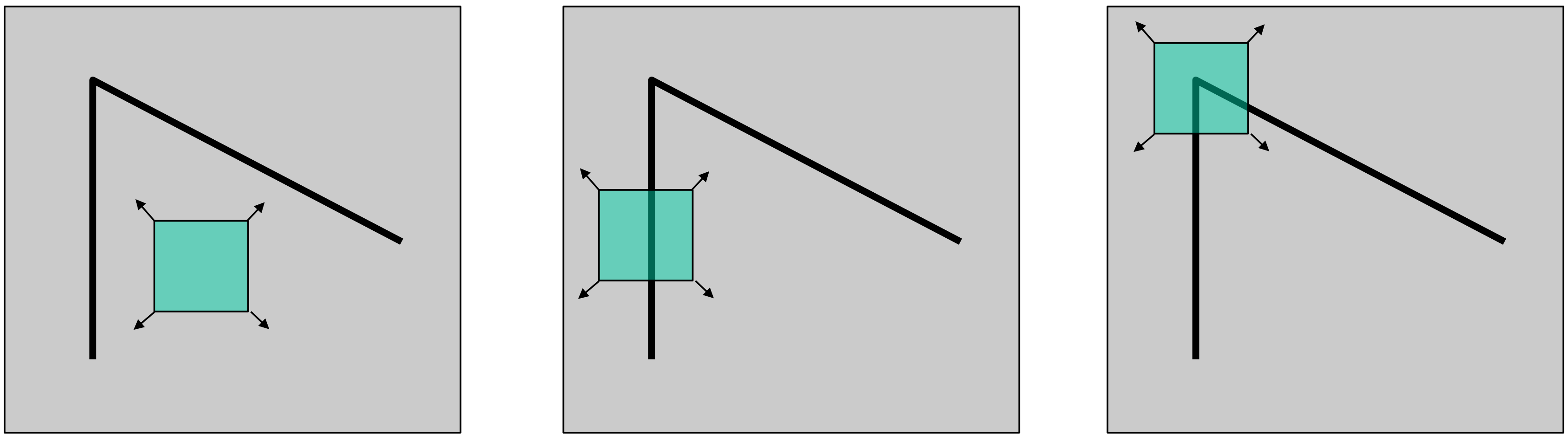
- Where are features good and bad?



Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

Feature detection

Uniqueness = change when shifted by a *small amount*



“flat” region:
no change in all directions

“edge”:
no change along the edge direction

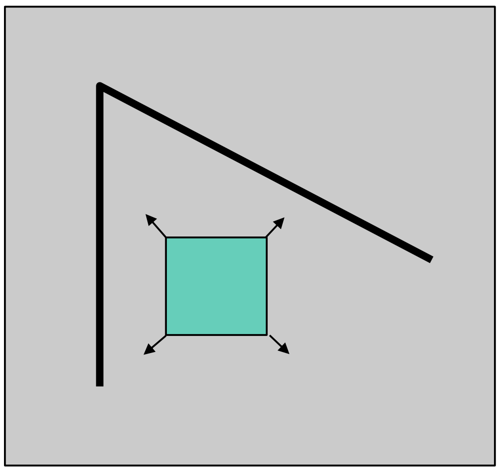
“corner”:
significant change in all directions

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

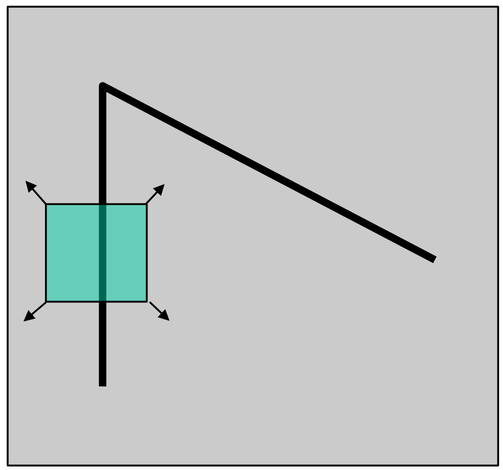
Feature detection

Define

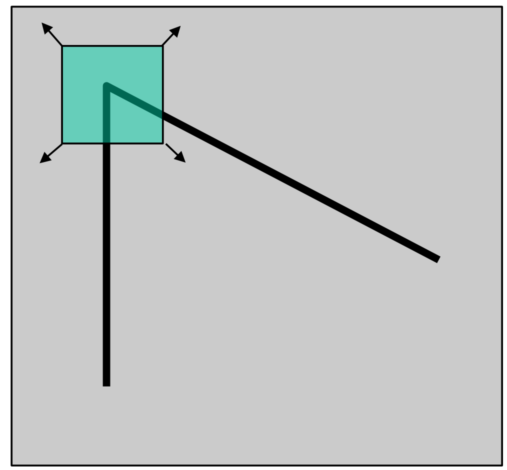
E(u,v) = amount of change when you shift the window by (u,v)



E(u,v) is small
for **all** shifts



E(u,v) is small
for **some** shifts



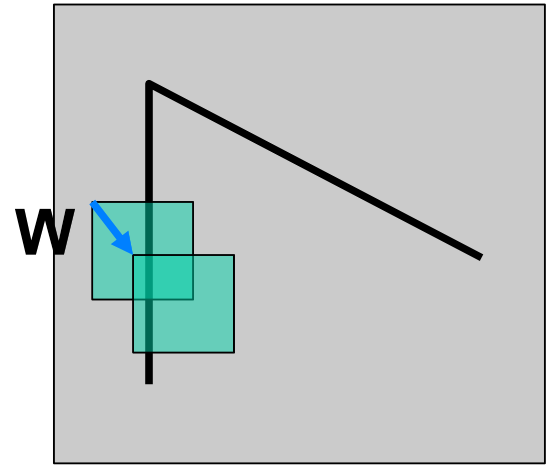
E(u,v) is small
for **no** shifts

We want $\min_{(u,v)} E(u,v)$ to be ______

Feature 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 Sum of the Squared Differences (SSD)
- this defines an SSD "error" *E(u,v)*:



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

Small motion assumption

Taylor 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 approximation is good!

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

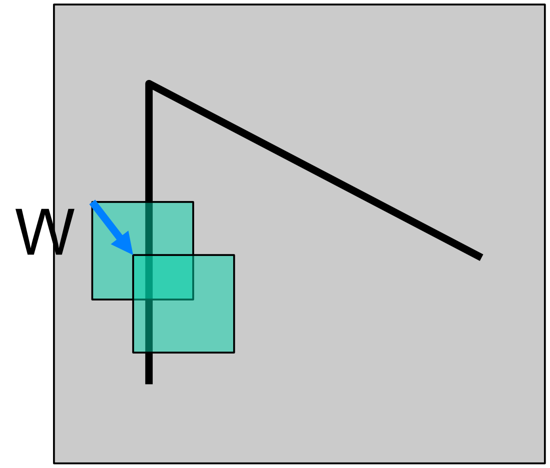
shorthand: $I_x = \frac{\partial I}{\partial x}$

Plugging this into the formula on the previous slide…

Feature 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):



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

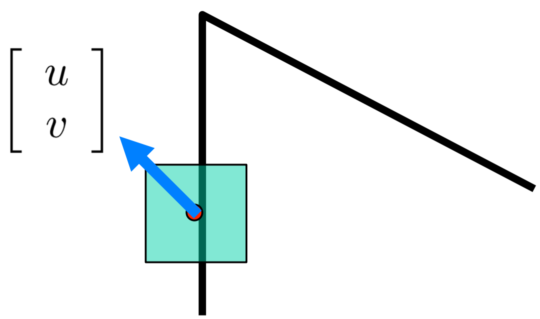
$$\approx \sum_{(x,y)\in W} \left[I(x,y) + [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)\right]^2$$

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

Feature detection: the math

This can be rewritten:

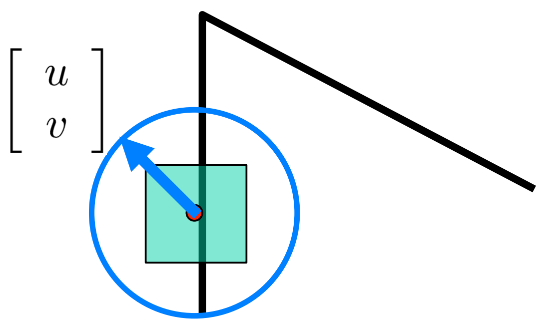
$$E(u,v) = [u\ v] \underbrace{\left(\sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right)}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



Feature detection: the math

This can be rewritten:

$$E(u,v) = [u\ v] \underbrace{\left(\sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right)}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



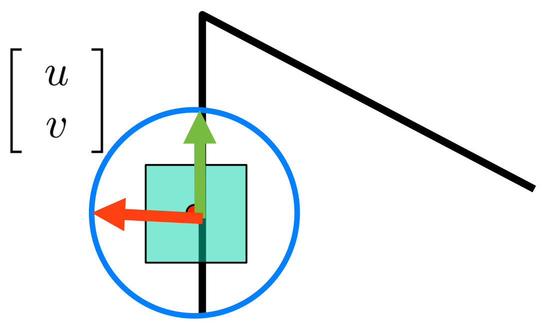
Which [u v] maximizes E(u,v)?

Which [u v] minimizes E(u,v)?

Feature detection: the math

This can be rewritten:

$$E(u,v) = [u\ v] \underbrace{\left(\sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right)}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



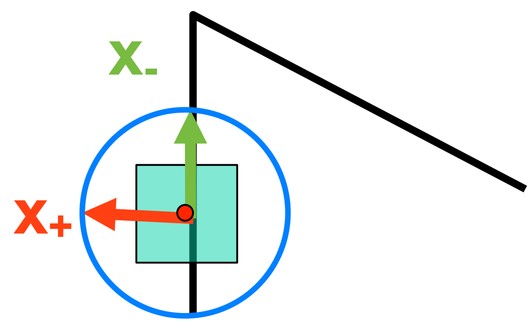
Which [u v] maximizes E(u,v)?

Which [u v] minimizes E(u,v)?

Feature detection: the math

This can be rewritten:

$$E(u,v) = [u \;\; v] \underbrace{\left(\sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right)}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



Eigenvector x_+ with the largest eigen value

Eigenvector x_- with the smallest eigen value

Quick eigenvalue/eigenvector review

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

$$Ax = \lambda x$$

The scalar λ is the **eigenvalue** corresponding to **x**

- The eigenvalues are found by solving:

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

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

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

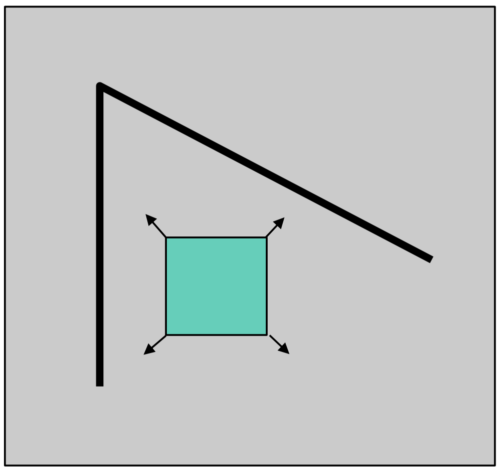
- The solution:

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

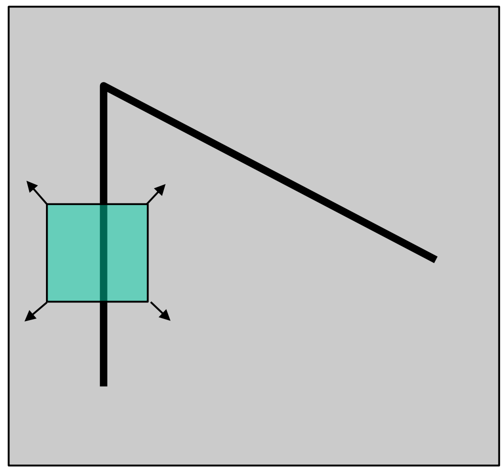
Feature detection

Local measure of feature uniqueness

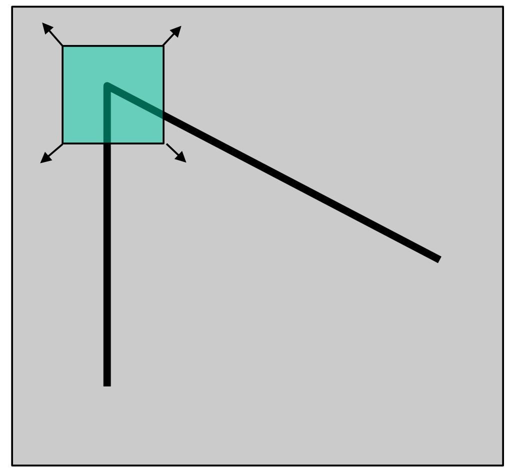
- E(u,v) = amount of change when you shift the window by (u,v)



E(u,v) is small
for **all** shifts



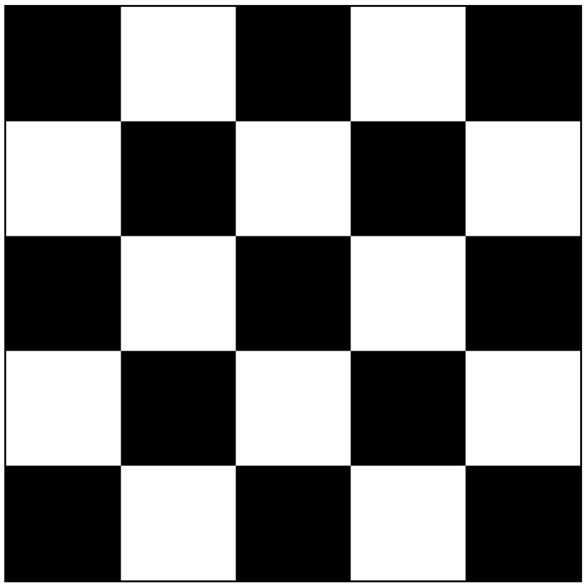
E(u,v) is small
for **some** shifts



E(u,v) is small
for **no** shifts

We want $\min_{(u,v)} E(u,v)$ to be large

Eigenvalues of **H**



I

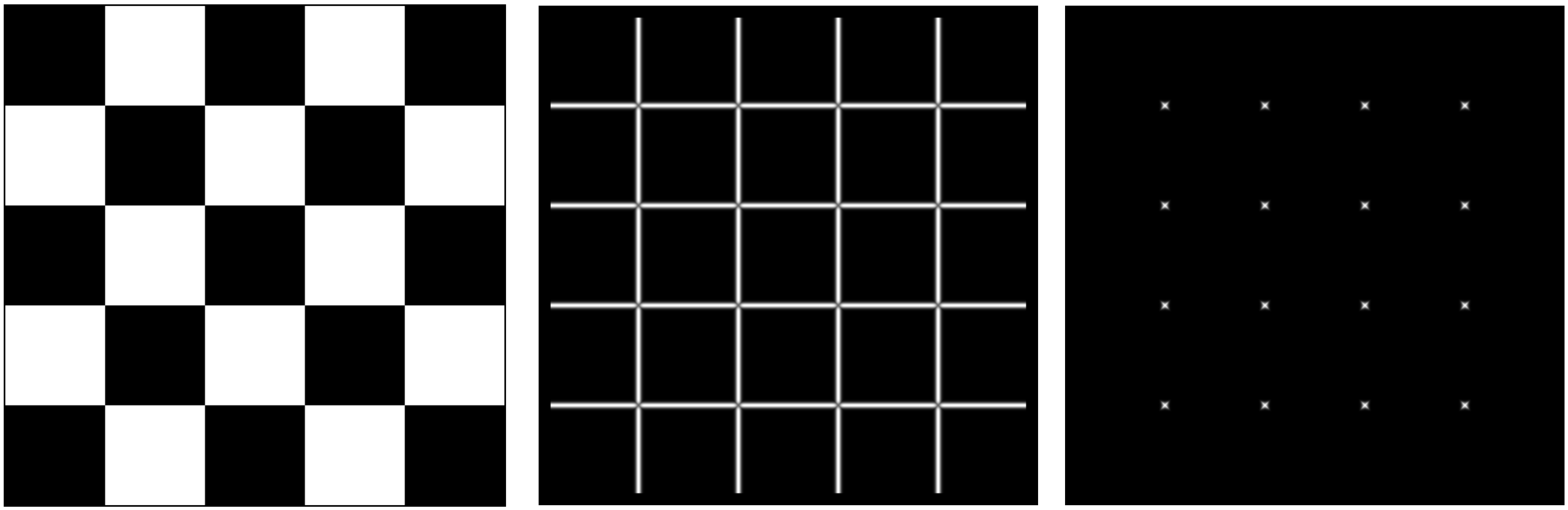
?

λ_+

?

λ_-

Eigenvalues of **H**

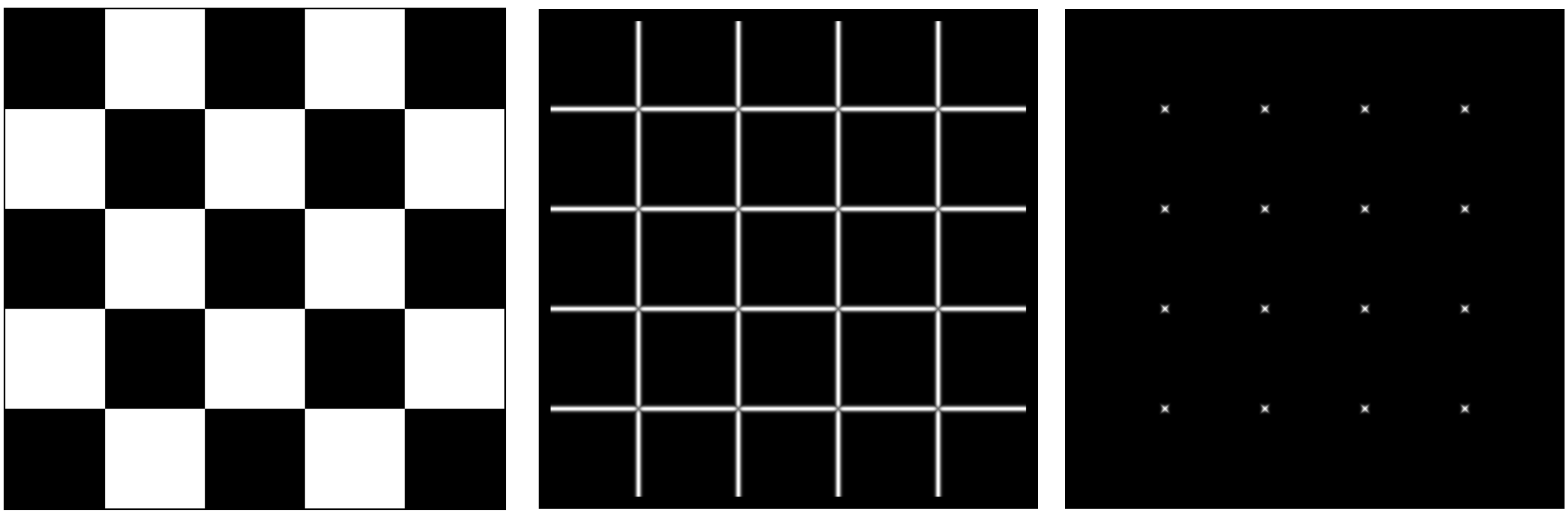


I λ_+ λ_-

Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the ***H*** matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



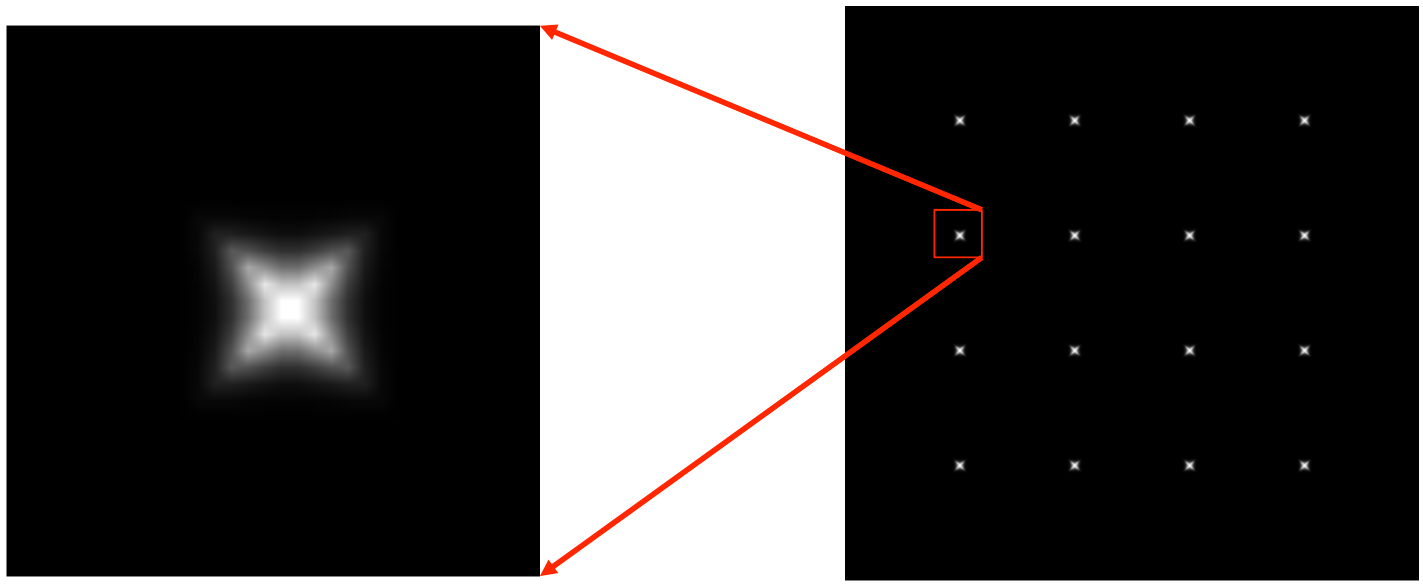
I λ_+ λ_-

Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the ***H*** matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features

Called "non-max suppression"



λ_-

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f
0.03	0.02	

Flat

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f
0.03	0.02	0.012

Flat

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f	
0.03	0.02	0.012	Flat
3	0.02	0.02	?

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f	
0.03	0.02	0.012	Flat
3	0.02	0.02	Edge

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f	
0.03	0.02	0.012	Flat
3	0.02	0.02	Edge
2.5	3	1.36	?

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f	
0.03	0.02	0.012	Flat
3	0.02	0.02	Edge
2.5	3	1.36	Corner

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f	
0.03	0.02	0.012	Flat
3	0.02	0.02	Edge
2.5	3	1.36	Corner
5	6	2.73	?

The Harris operator

λ_- is a variant of the “Harris operator” for feature detection

$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

λ_+	λ_-	f	
0.03	0.02	0.012	Flat
3	0.02	0.02	Edge
2.5	3	1.36	Corner
5	6	2.73	Strong corner

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

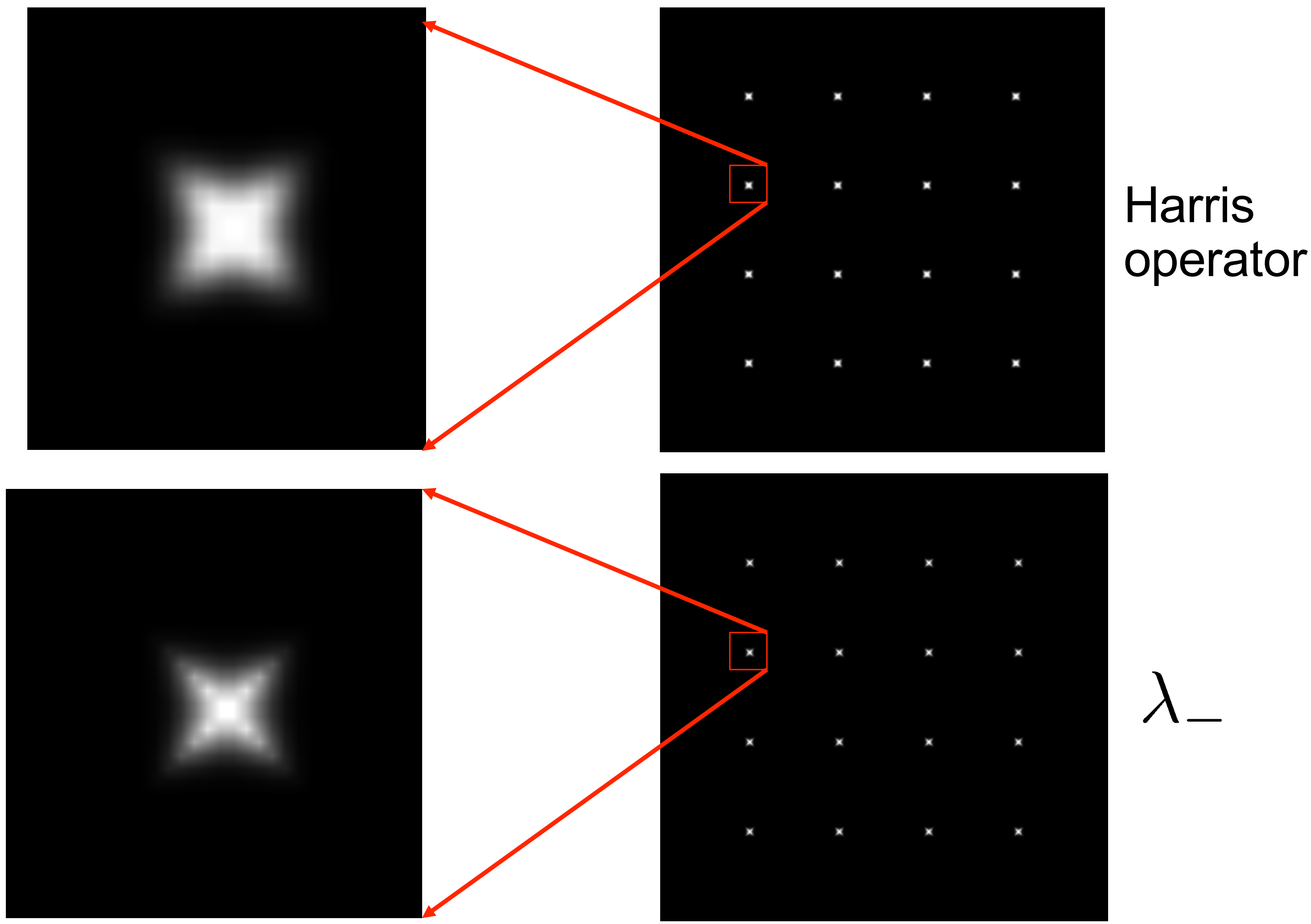
$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$= \frac{h_{11} \times h_{22} - h_{12} \times h_{21}}{h_{11} + h_{22}}$$

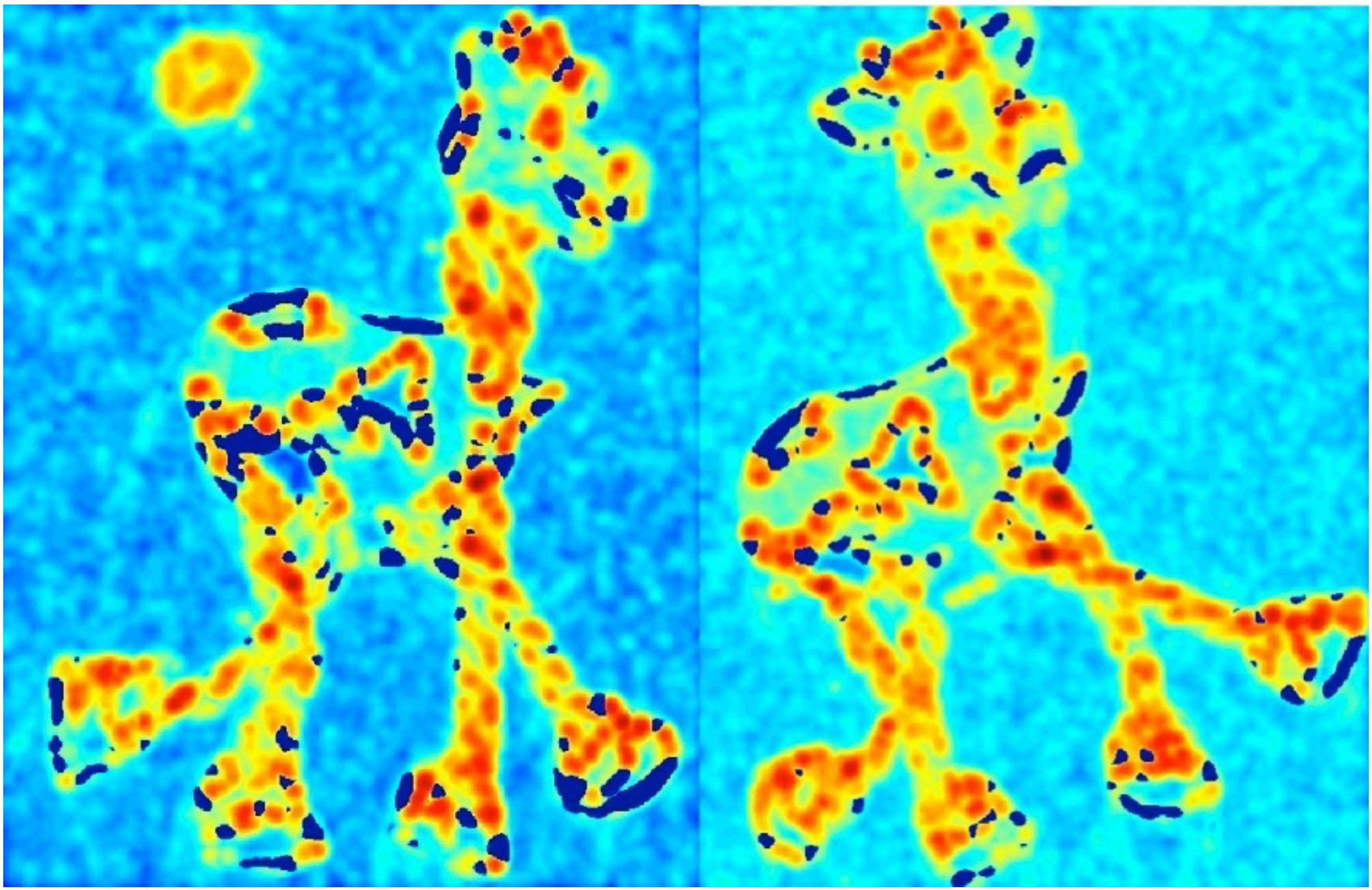
- **The trace is the sum of the diagonals, i.e., trace(H) = h_{11} + h_{22}**
- **Very similar to λ_- but less expensive (no square root)**
- **Called the "Harris Corner Detector" or "Harris Operator"**
- **Lots of other detectors, this is one of the most popular**

The Harris operator

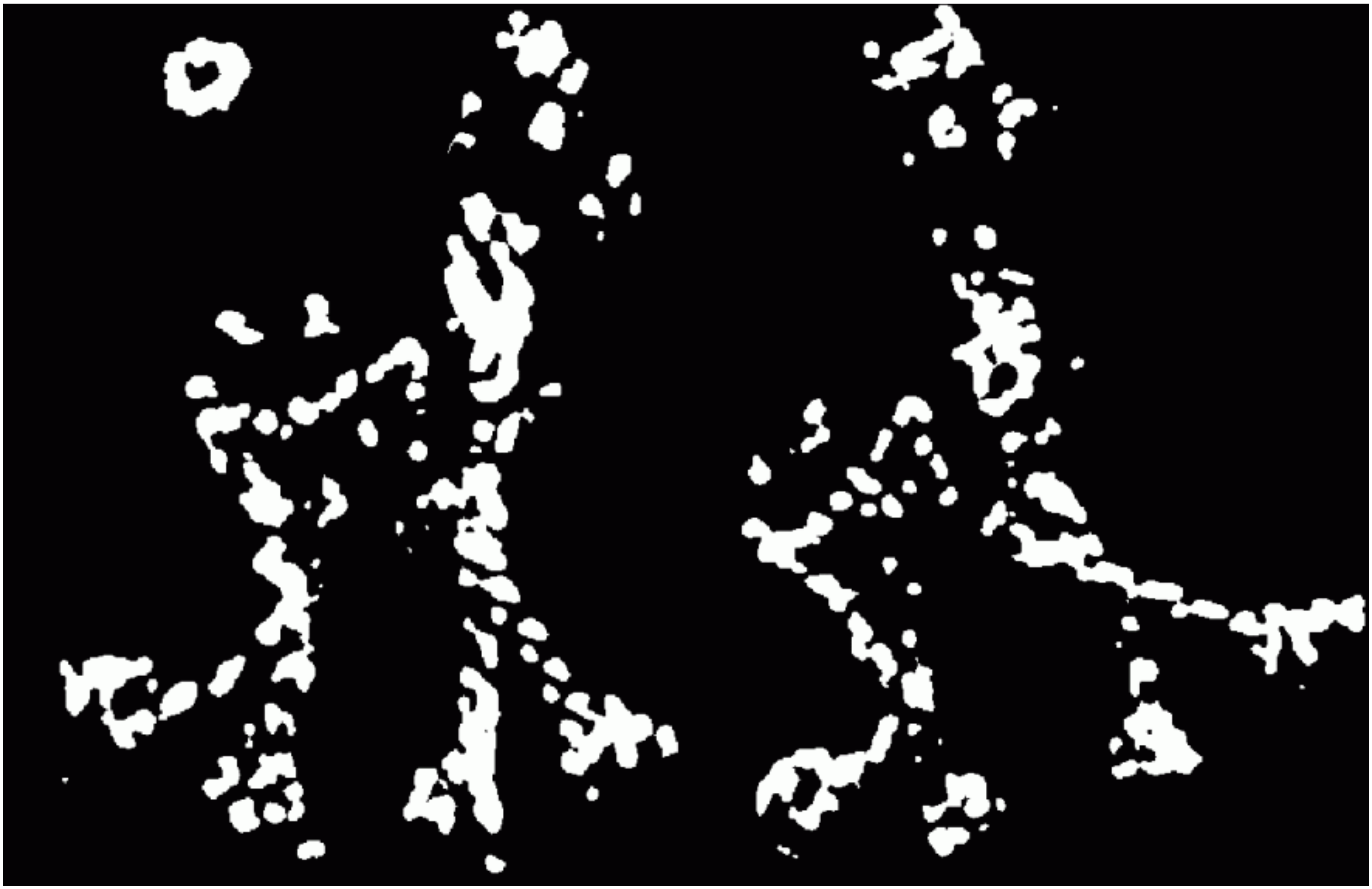


Harris detector example

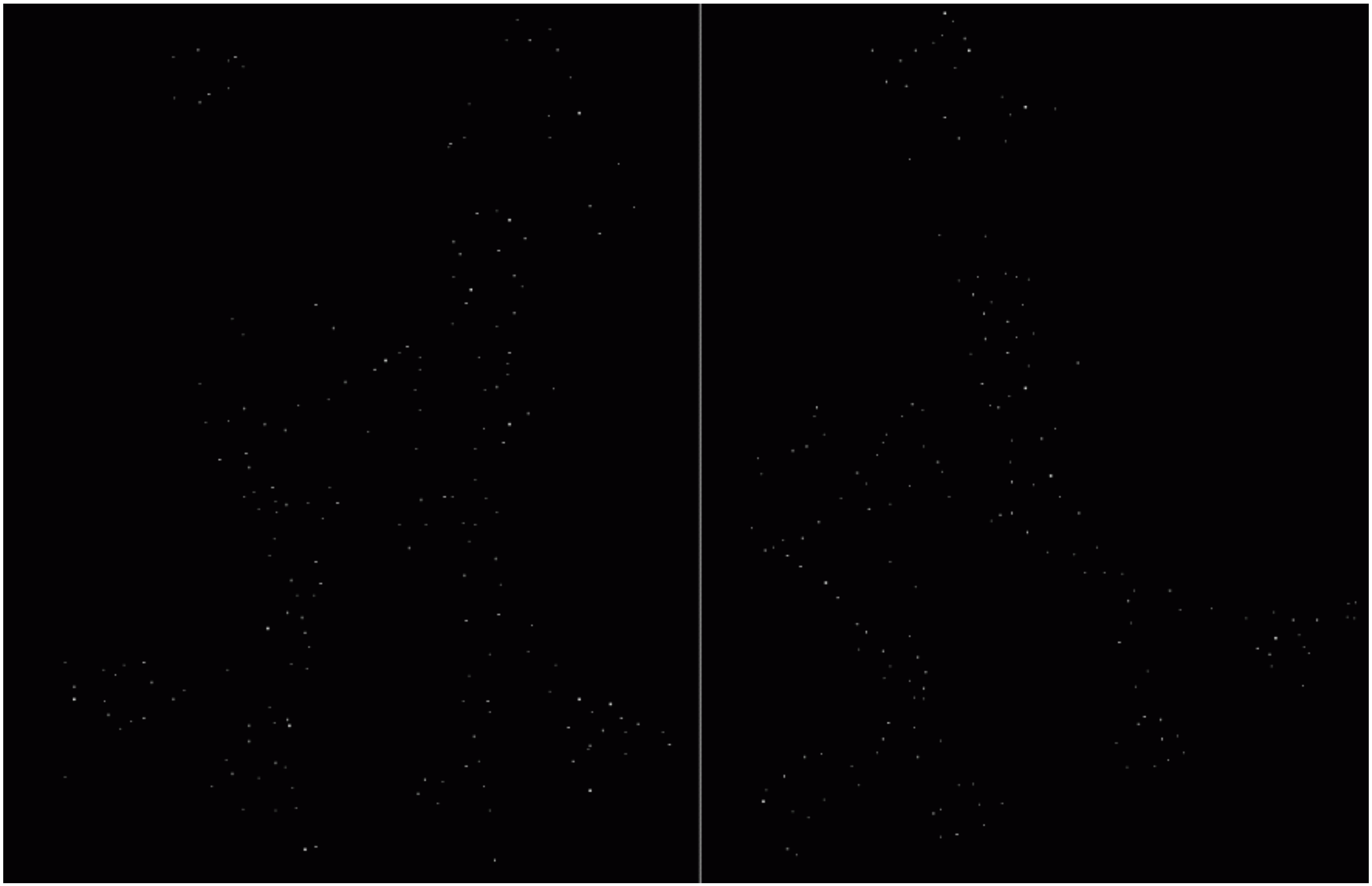
f value (red high, blue low)



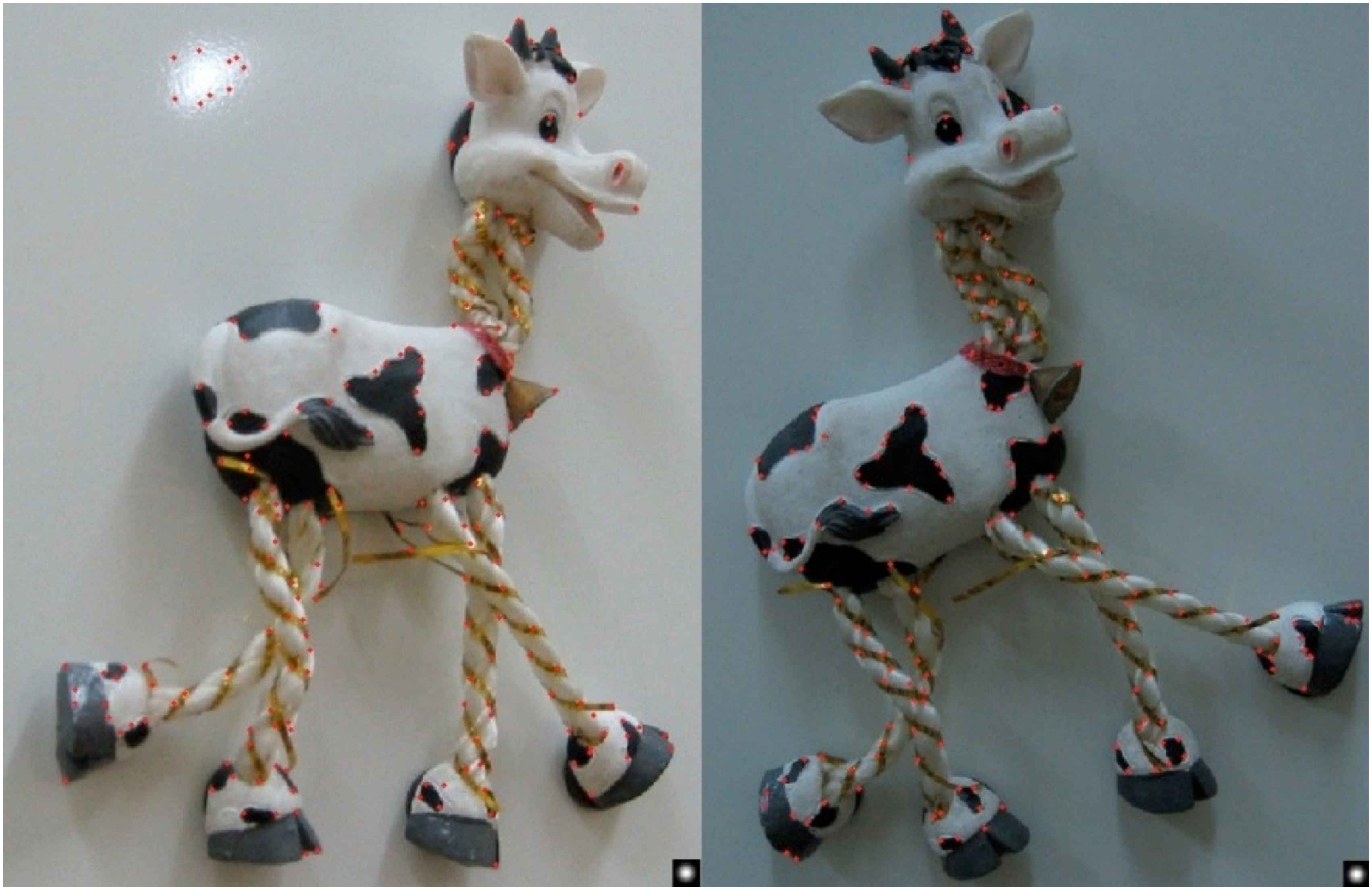
Threshold (f > value)



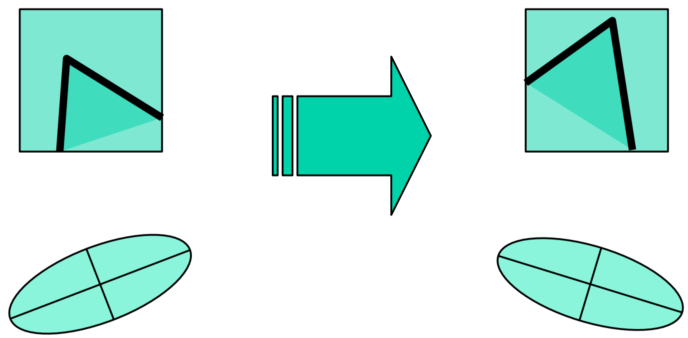
Find local maxima of f



Harris features (in red)



Harris corner response is invariant to rotation



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

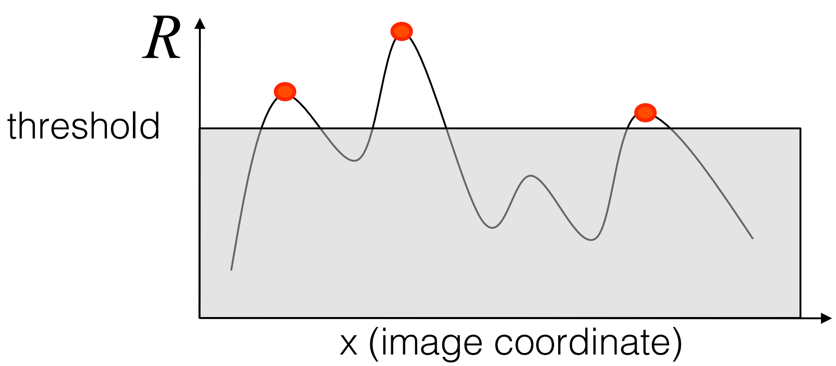
Corner response R is invariant to image rotation

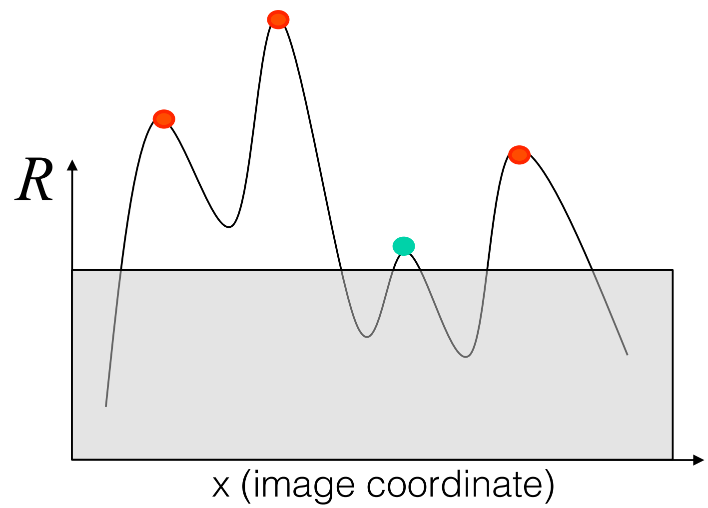
Harris corner response is invariant to intensity changes

Partial invariance to *affine intensity* change

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

- Intensity scale: $I \rightarrow a\, I$





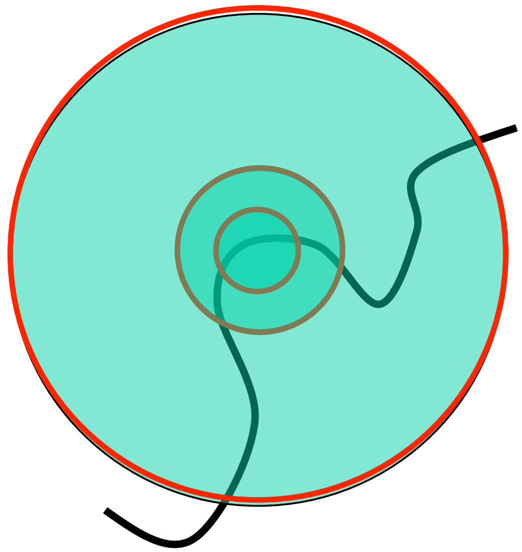
The Harris detector is not invariant to changes in …

The Harris corner detector is <u>not</u> invariant to scale

Multi-scale detection

Scale invariant detection

Suppose you're looking for corners

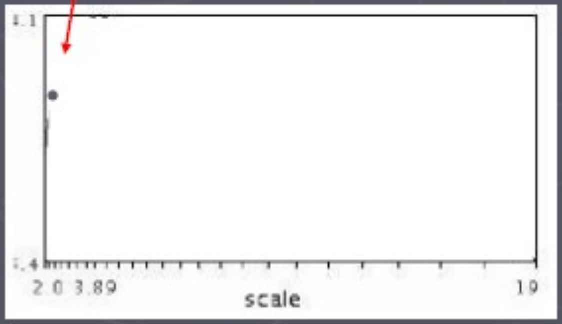


Key idea: find scale that gives local maximum of f

- f is a local maximum in both position and scale

Automatic scale selection

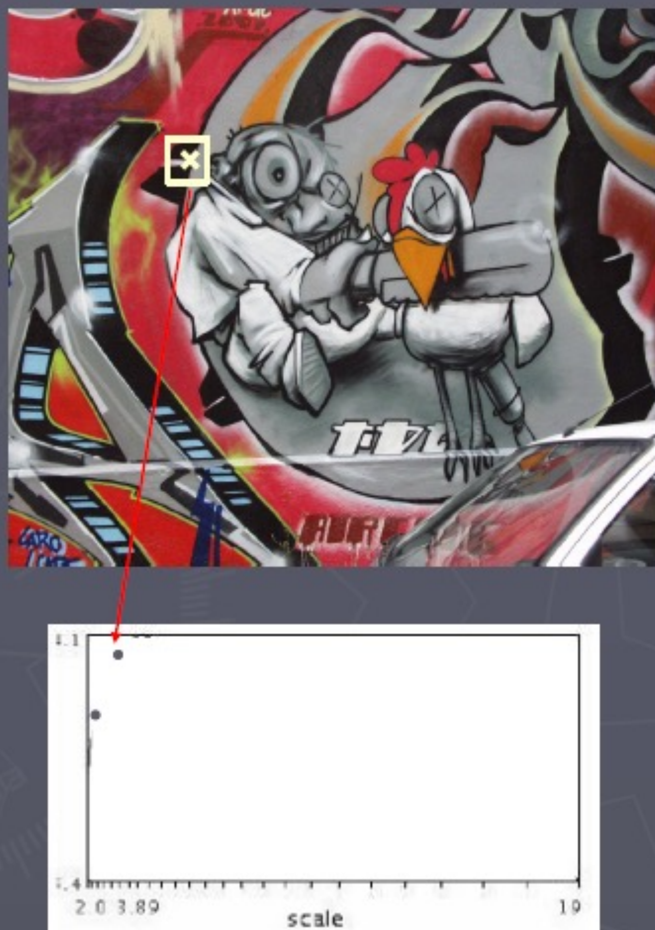
Lindeberg et al., 1996



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Slide from Tinne Tuytelaars

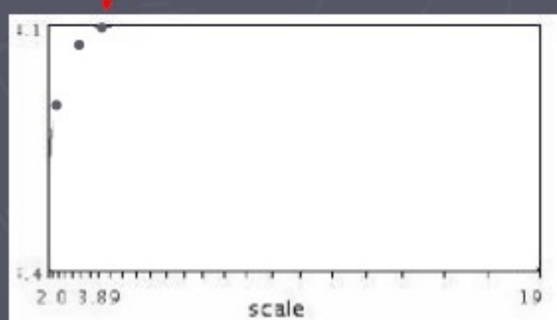
Automatic scale selection



$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection

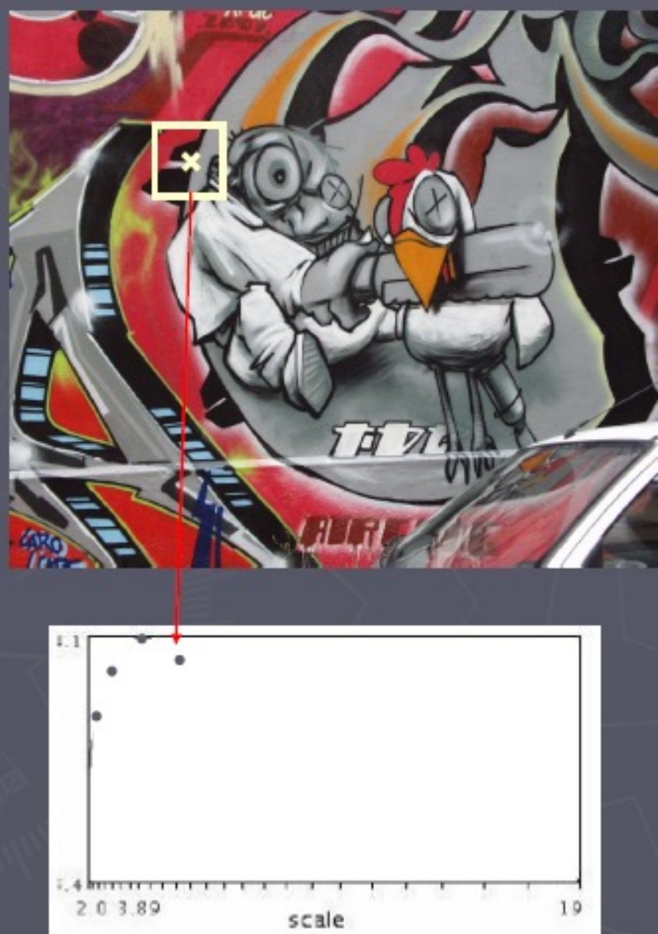
Function responses for increasing scale
(Scale trace (signature))



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

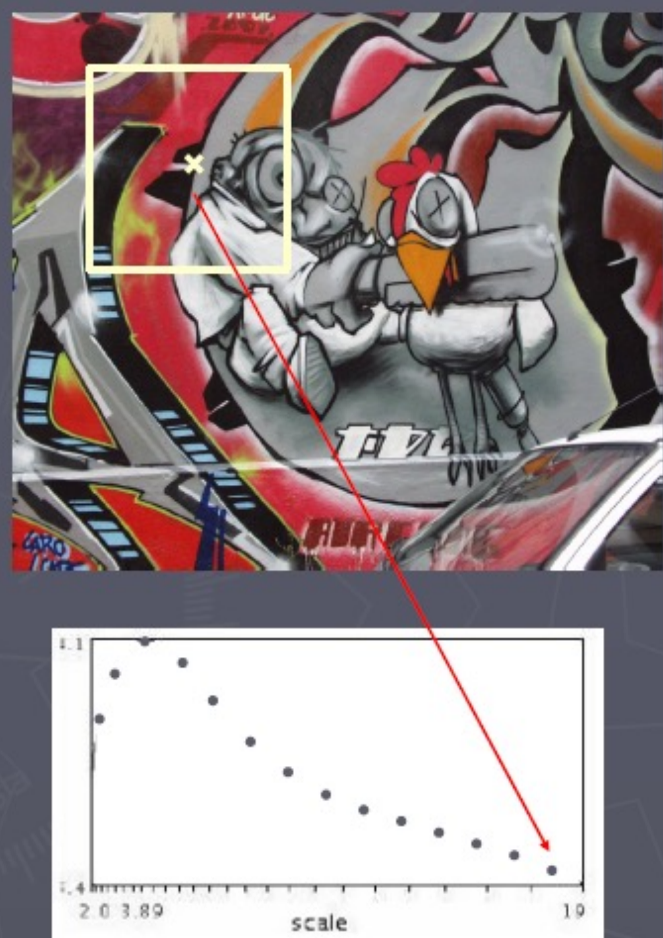
Function responses for increasing scale (scale signature)



$$f(I_{i_1 \ldots i_m}(x, \sigma))$$

Automatic scale selection

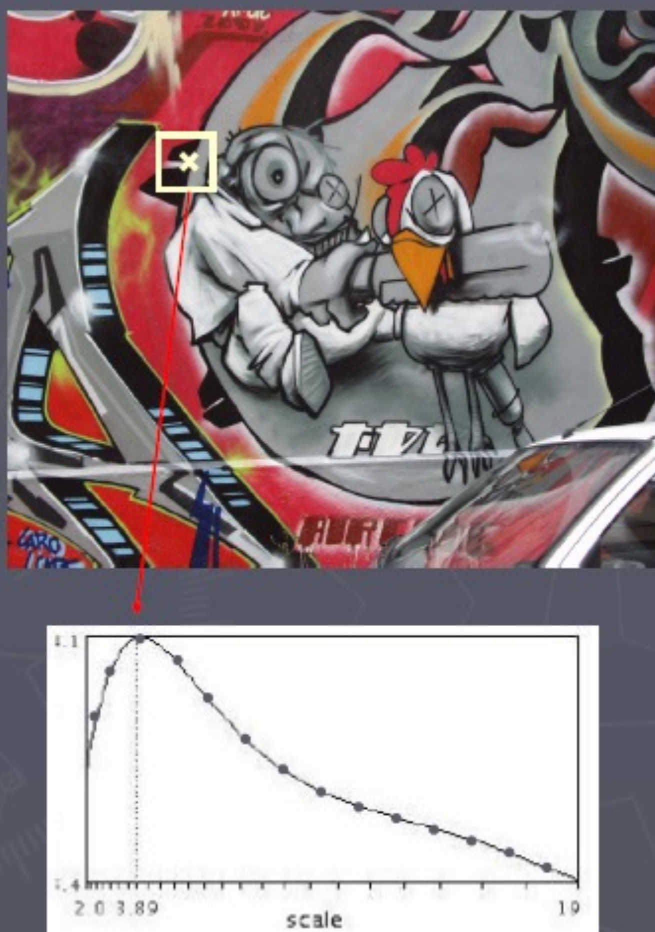
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

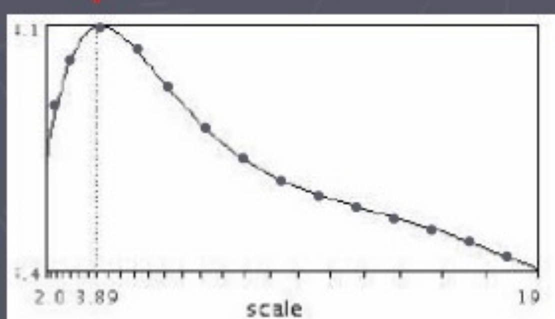
Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

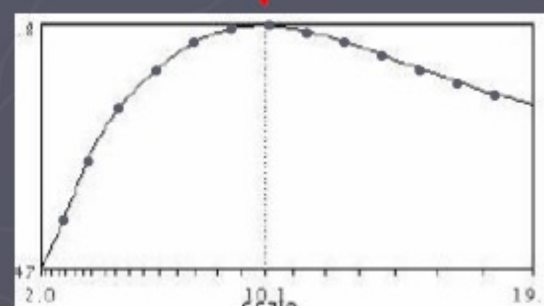


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection



$f(I_{i_1 \ldots i_m}(x,\sigma))$



$f(I_{i_1 \ldots i_m}(x',\sigma'))$

Readings

Szeliski textbook (2nd edition), Chapter 7