CS484 - CS555:
Introduction to Computer Vision

(Hand-crafted) Features: SIFT

Dr. Sedat Ozer

Overview

- Last lecture (before the presentations):
  - Hyper-parameter tuning
  - Optimization in Deep learning

- This week: How was it before the deep learning era!!
  - Features,
  - Hand-crafted features,
  - (Harris) Corner detection,
  - SIFT features,
  - Classification with SVM,

Overview

- Definitions for attributes, features,
- The importance of feature selection,
- Global vs. local features,
- Hessian Matrix and Harris Corner Detection
- SIFT (Scale-invariant feature transform):
    - Examples,
    - Gaussian scale-space,
    - Difference of Gaussians: Laplacian of a Gaussian
    - SIFT Key-point detection,
    - SIFT descriptor computation,
    - Other SIFT-like algorithms and applications,
- Conclusion

Definitions

Attribute:

- An object's **entity**,
- An **abstraction** of an entity,
- A **property**,
- A **characteristic** of a variable (or an object).

Feature:

- individual **measurable** properties of the phenomena being observed,
- A feature is a **distinct** property or piece.

Source: Wikipedia

Definitions

Attribute:

- An object's
- An **abstractio** ...ity,
- A **property**,
- A **characte**

Attribute = Feature
(describes - summarizes the image)

Feature:

- individual **m**
- A **featu** ... p

S...rce: Wikipedia

What does that mean for a classifier?

Feature-based Tasks

Features are good for:

- **Segmentation** tasks, (Example: separating the image into two: background and foreground, i.e., region of interest areas),
- **Recognition** tasks, (Example: assigning a label to the segmented region of interests),
- **Regression** tasks, (Examples: Fitting a function, estimation, prediction of a continuous value),
- **Tracking** tasks, (Example: Correlating objects over time),
- **Matching**, **retrieval** tasks, (Example: Searching through a database to list the closest samples),
- **Alignment, registration** tasks, (Example: scaling both images or objects to the same size and orientation),
- **Semantics** extraction.
- ...

A visual analysis example: What is in this image?

Image source: https://www.carthrottle.com/post/this-crazy-8000bhp-bmw-concept-is-the-coolest-thing-youve-ever-see/

Dream car of a 4 years old kid!

- 42 wheels, and of course 42-wheel drive
- Powered by 19 Porsche engines, each engine producing 459 horsepower.
- All engines linked to a single transmission.
- Three seats, three steering wheels – all operating simultaneously
- The trunk would be full of toys, and you can play in it!

Choosing the appropriate features is important: An Illustration

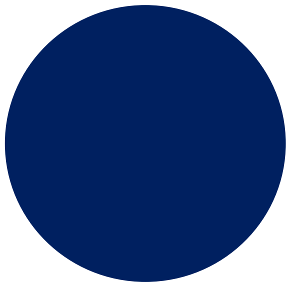
$$\text{Object\_A} = \begin{bmatrix} m_A \\ \sigma_A \end{bmatrix} \qquad \text{Object\_B} = \begin{bmatrix} m_B \\ \sigma_B \end{bmatrix}$$

Assume you have segmented two regions in an image and computed the mean and variance values of the interior pixel values for both of those regions:

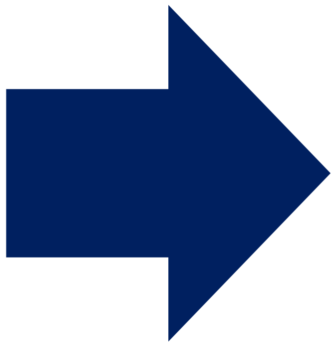
Choosing the appropriate features matters! An Illustration:

$$\text{Object\_A} = \begin{bmatrix} m_A \\ \sigma_A \end{bmatrix}$$

$$\text{Object\_B} = \begin{bmatrix} m_B \\ \sigma_B \end{bmatrix}$$



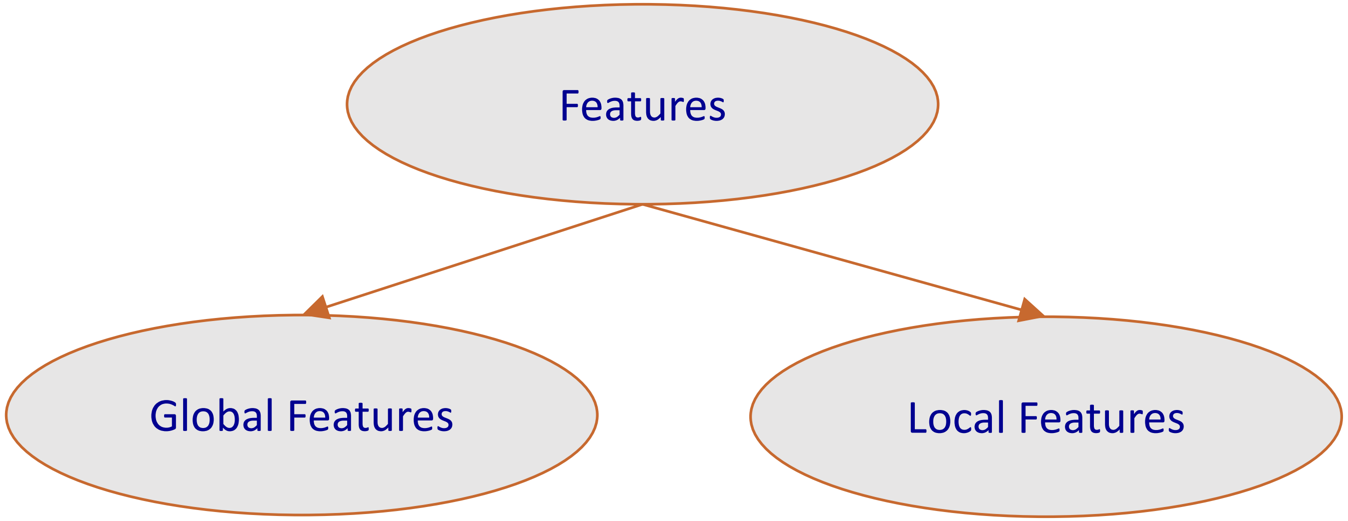
Object_A



Object_B

$$\left.\begin{array}{l} m_A = m_B \\ \sigma_A = \sigma_B \end{array}\right\} \text{Object\_A} \overset{?}{=} \text{Object\_B}$$

Global vs. Local Features:



Summarizes a global characteristic of the object/image.

Examples: shape-based: mean, variance, volume, Mass, Moments, Centroid, Orientation, image histogram, etc.

Summarizes a smaller (local) area.

Examples: (pixel) neighborhood based statistics, Local curvature, local min/max, local gradients, corner points, etc.

Image matching applications

Stereo correspondence and 3D reconstruction

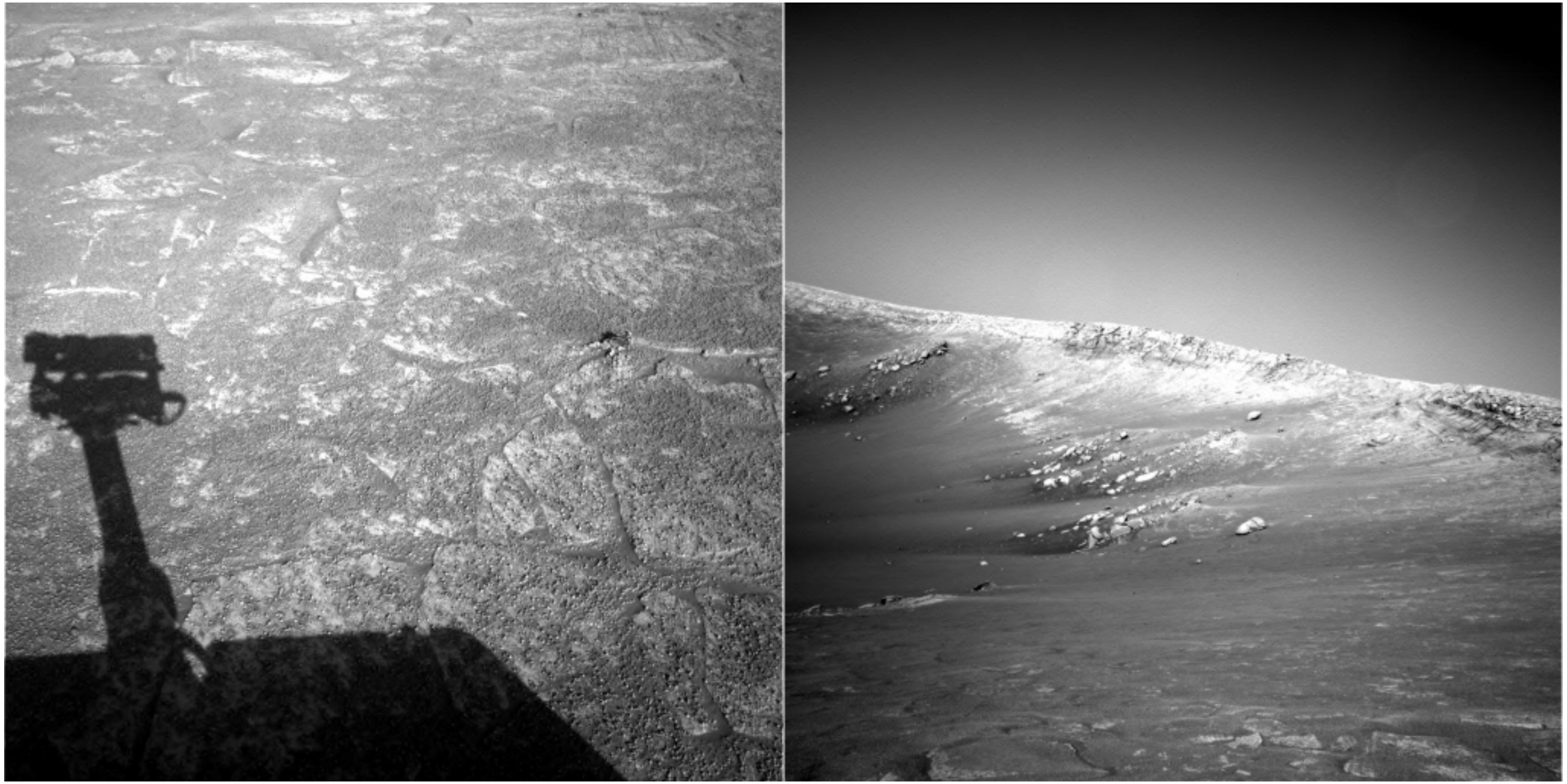
Image matching applications

Two images of Rome from Flickr

Image matching applications

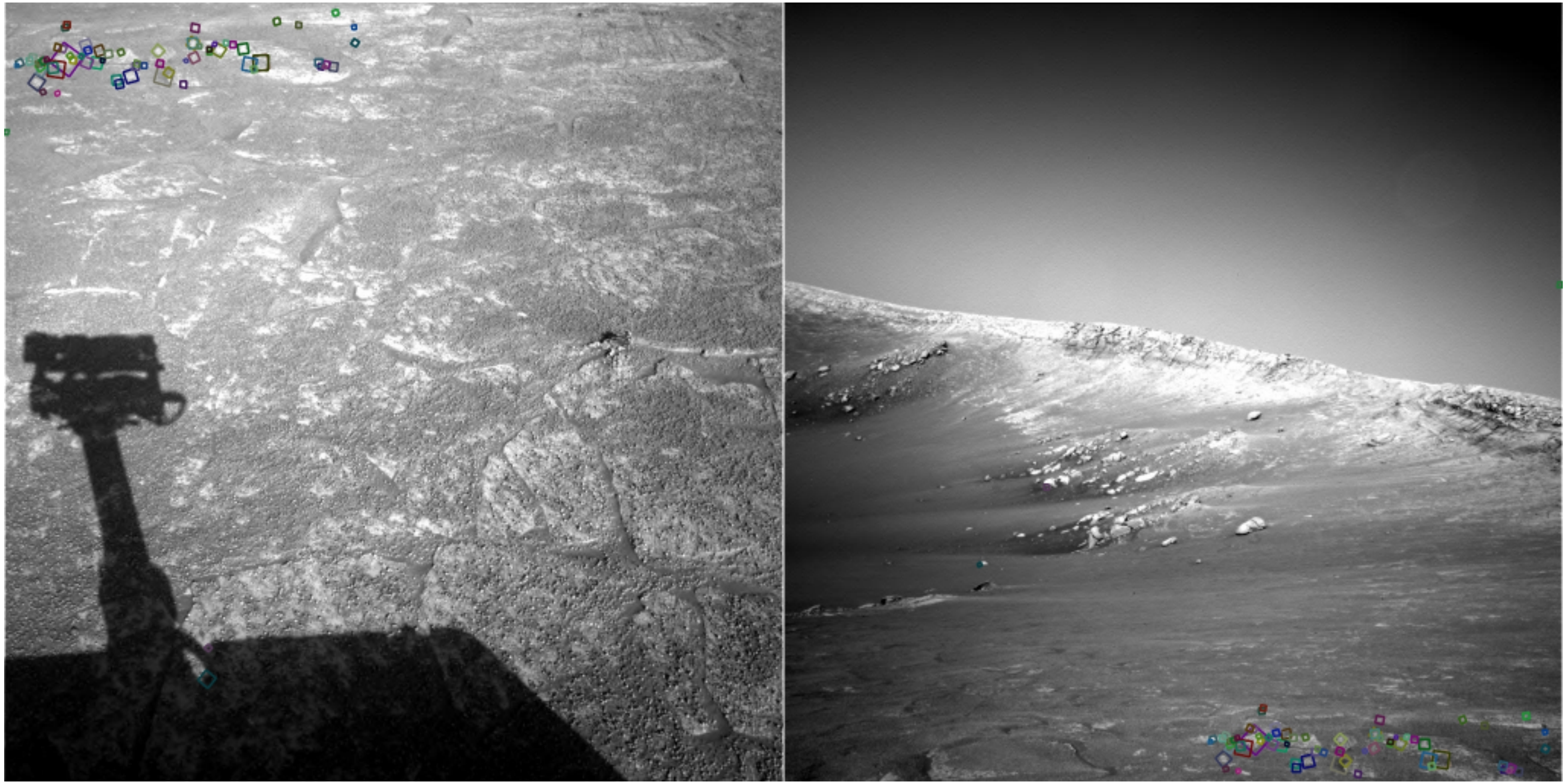
Two images of Rome from Flickr: harder case

Image matching applications



Two images from NASA Mars Rover: even harder case

Image matching applications



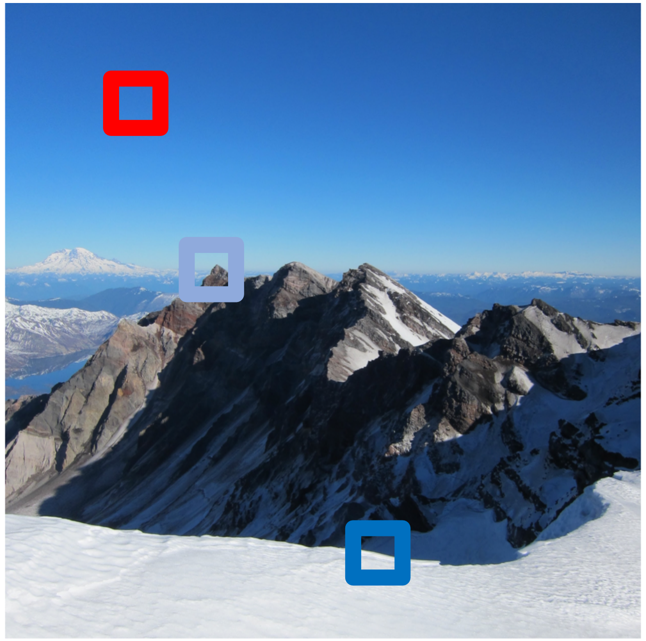
Two images from NASA Mars Rover: matching using local features

Advantages of local features

- Locality
  - features are local, so robust to occlusion and clutter
- Distinctiveness
  - can differentiate a large database of objects
- Quantity
  - hundreds or thousands in a single image
- Efficiency
  - real-time performance achievable
- Generality
  - exploit different types of features in different situations

Local features

- What makes a good feature?
- We want **uniqueness**.
  - Look for image regions that are **unusual**.
  - Lead to **unambiguous** matches in other images.
- How to define "unusual"?

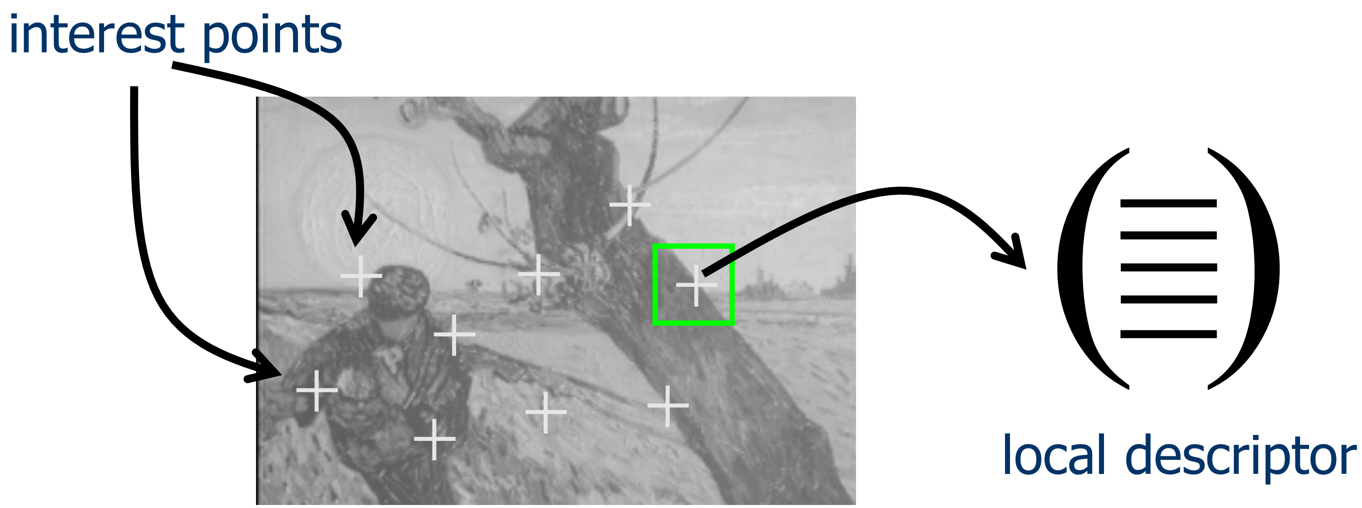


0D structure
not useful for matching

1D structure
edge, can be localized in 1D, subject to the aperture problem

2D structure
corner, can be localized in 2D, good for matching

Overview of the approach

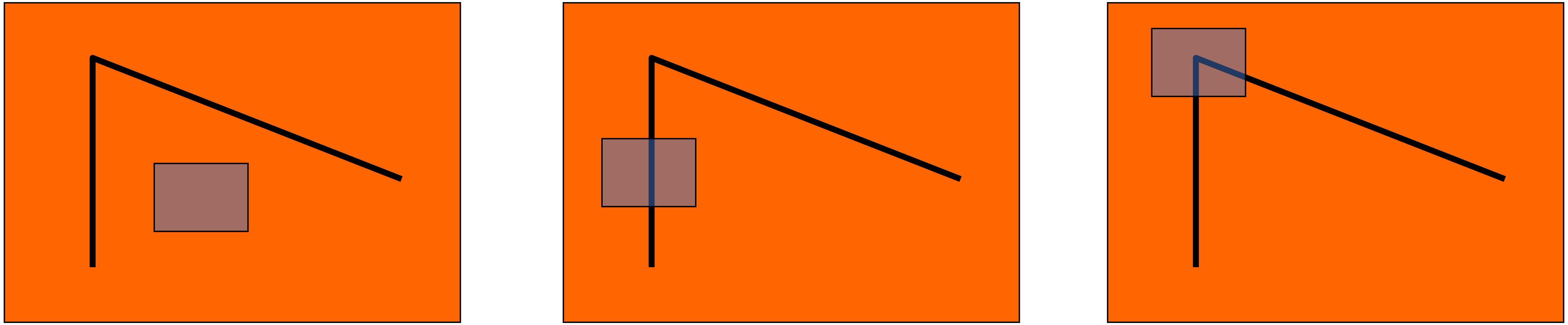


1. Extraction of interest points (characteristic locations).
2. Computation of local descriptors.
3. Determining correspondences.
4. Using these correspondences for matching/recognition/etc.

Local measures of uniqueness

Suppose we only consider a small window of pixels

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

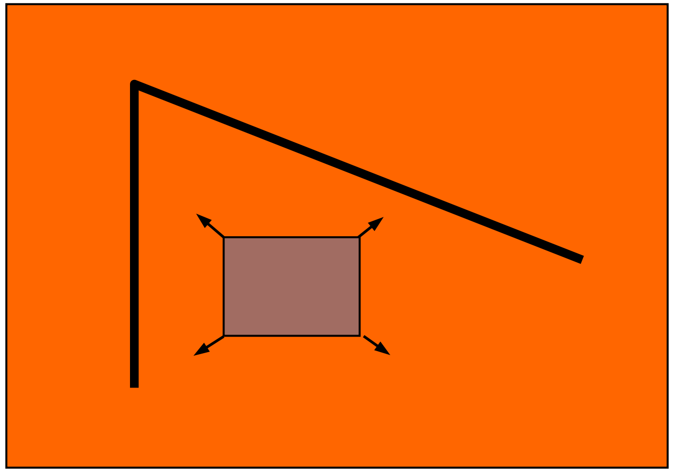


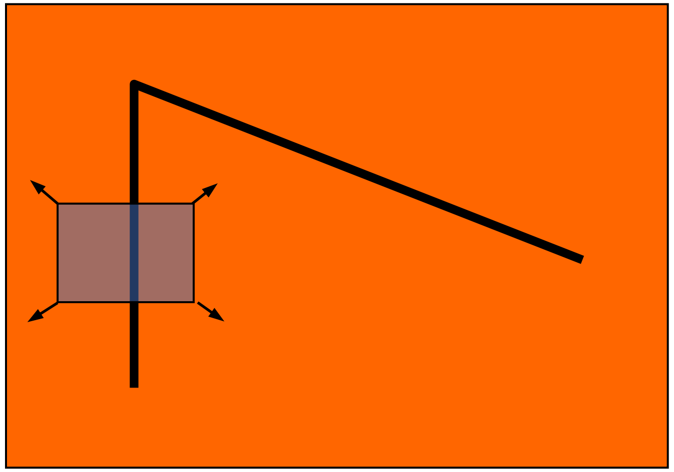
Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

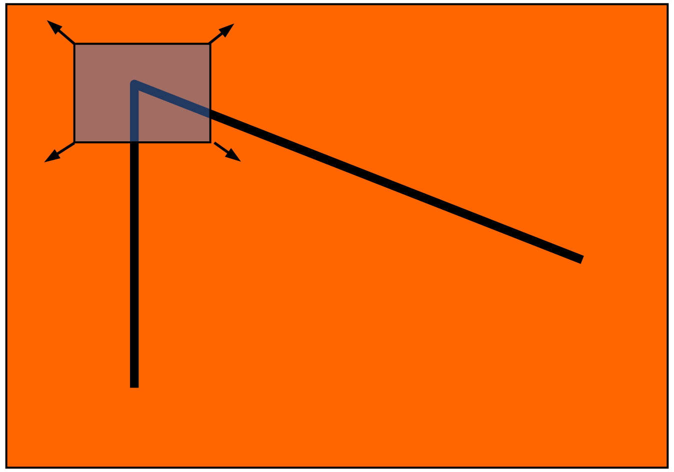
Feature detection

Local measure of feature uniqueness

- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*







"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

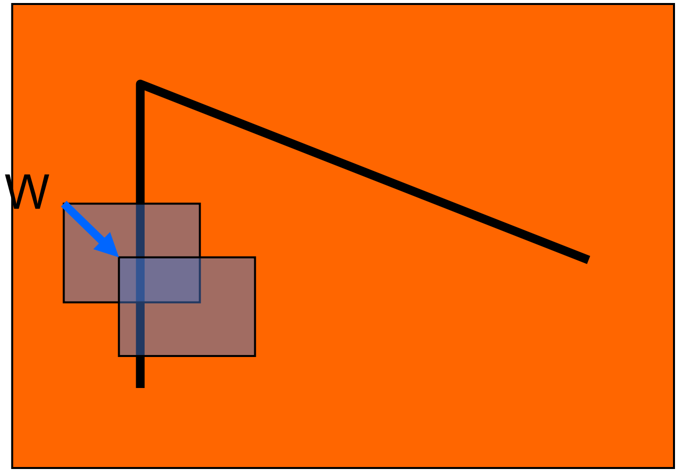
Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

Slide source: Ali Farhadi

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 (SSD)
- this defines an SSD "error" of *E(u,v)*:



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

Slide source: Ali Farhadi

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 enough

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

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

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

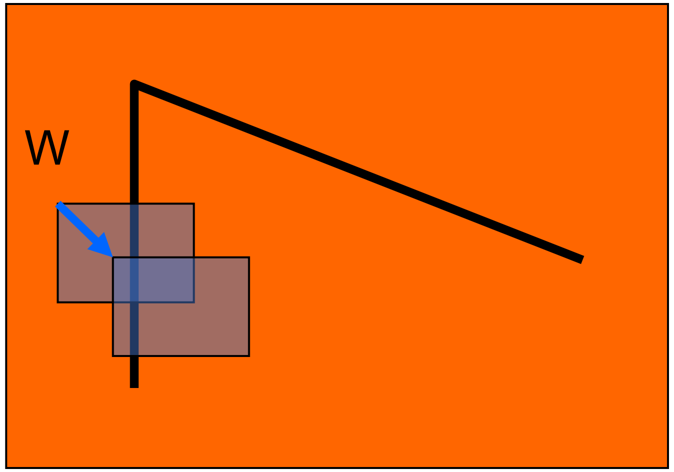
Plugging this into the formula on the previous slide…

Slide source: Ali Farhadi

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):



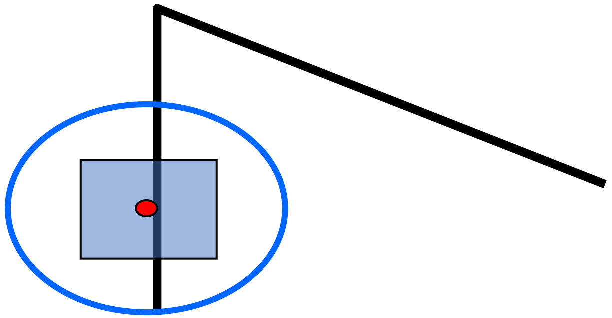
$$
\begin{aligned}
E(u,v) &= \sum_{(x,y)\in W} [I(x+u,y+v) - I(x,y)]^2 \\
&\approx \sum_{(x,y)\in W} \left[I(x,y) + [I_x \; I_y]\begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)\right]^2 \\
&\approx \sum_{(x,y)\in W} \left[[I_x \; I_y]\begin{bmatrix} u \\ v \end{bmatrix}\right]^2
\end{aligned}
$$

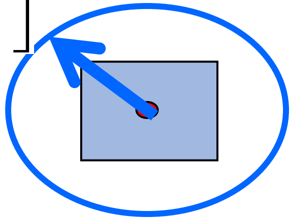
Slide source: Ali Farhadi

Feature detection: the math

This can be rewritten:

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





$$\begin{bmatrix} u \\ v \end{bmatrix}$$

For the example above

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

Slide source: Ali Farhadi

Quick eigenvalue/eigenvector review

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

A**x**=λ**x** → $$Ax = \lambda x$$

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:

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

Once you know λ, you find **x** by solving:

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

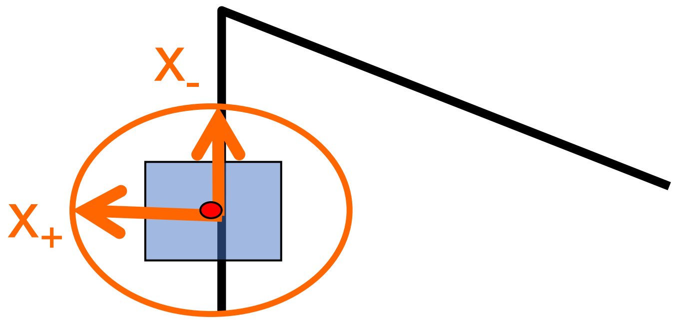
Slide source: Ali Farhadi

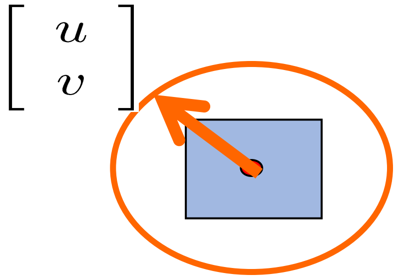
Feature detection: the math

This can be re-written:

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

(Also known as M matrix, instead of H)





Eigenvalues and eigenvectors of H

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

$$\begin{aligned} Hx_+ &= \lambda_+ x_+ \\ Hx_- &= \lambda_- x_- \end{aligned}$$

Classification of image points using eigenvalues of M:

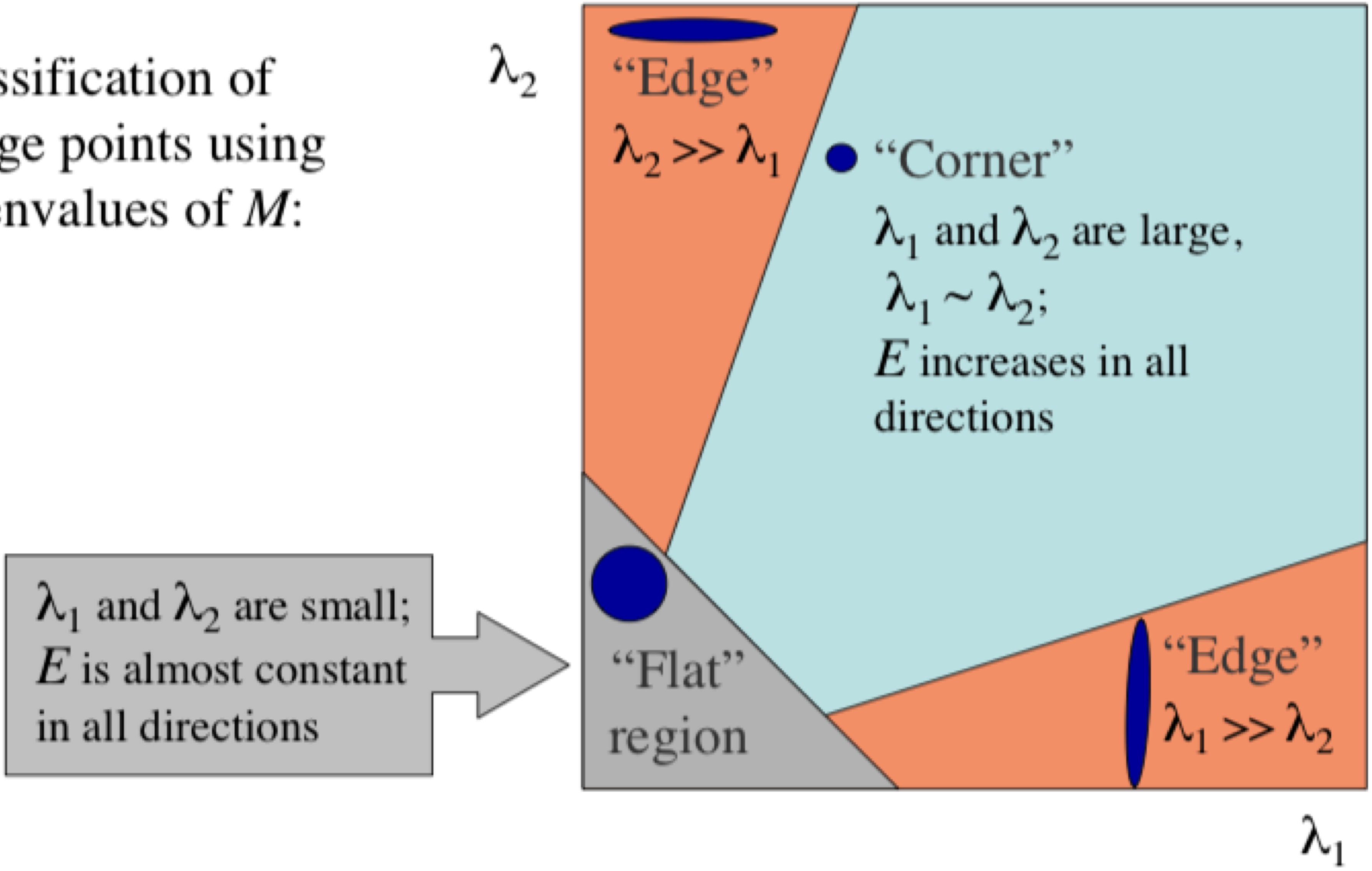


Image source: https://courses.cs.washington.edu/courses/cse576/06sp/notes/HarrisDetector.pdf

Feature detection: the math

How are λ_+, x_+, λ_-, and x_+ relevant for feature detection?

- What's our feature scoring function?

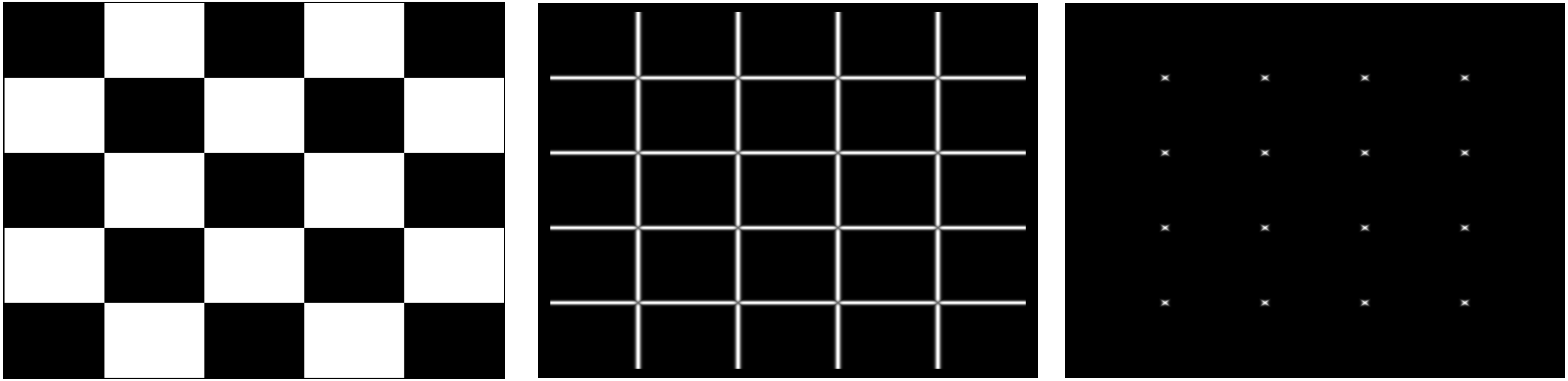
Feature detection: the math

How are λ_+, x_+, λ_-, and x_+ relevant for feature detection?

- What's our feature scoring function?

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*



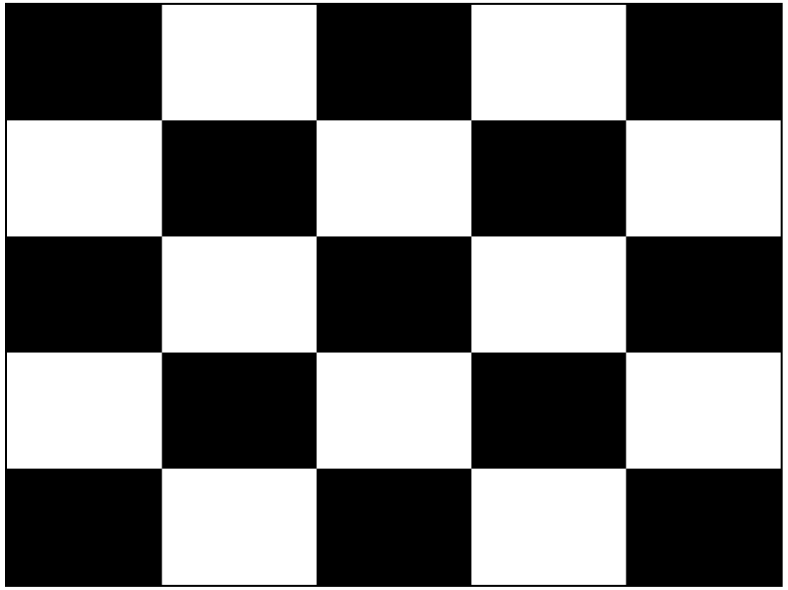
I λ_+ λ_-

Slide source: Ali Farhadi

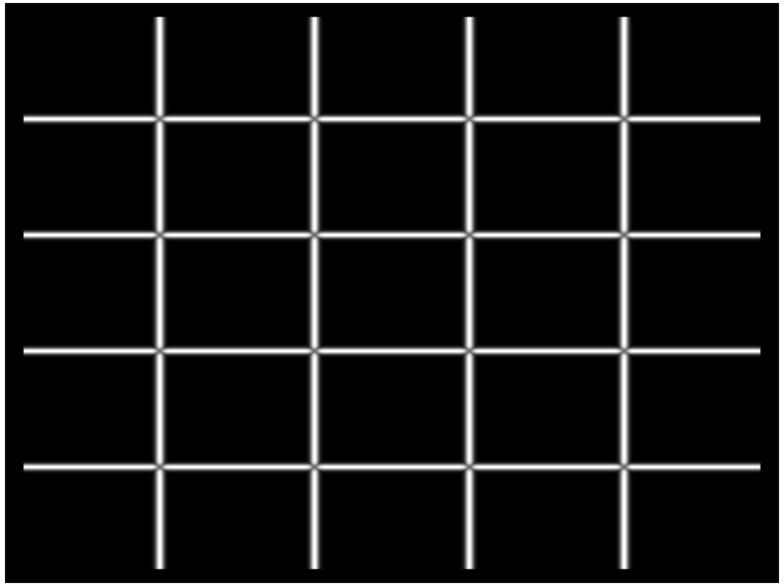
Feature detection summary

Here's what you do

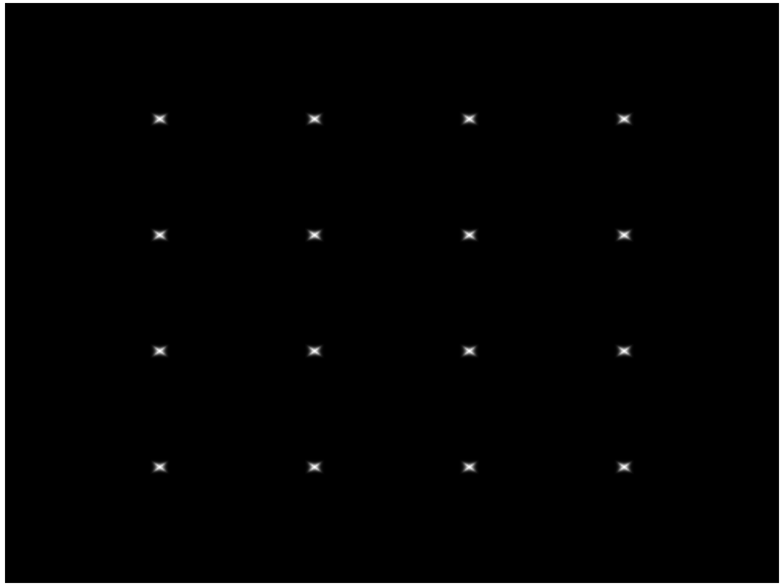
- Compute the gradient at each point in the image
- Create the *H* matrix from the entries in the gradient (for each pixel)
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



I



λ_+



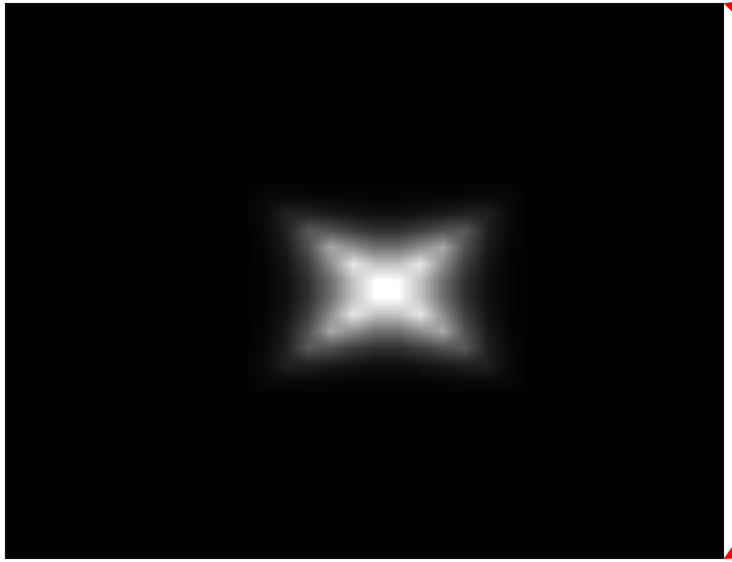
λ_-

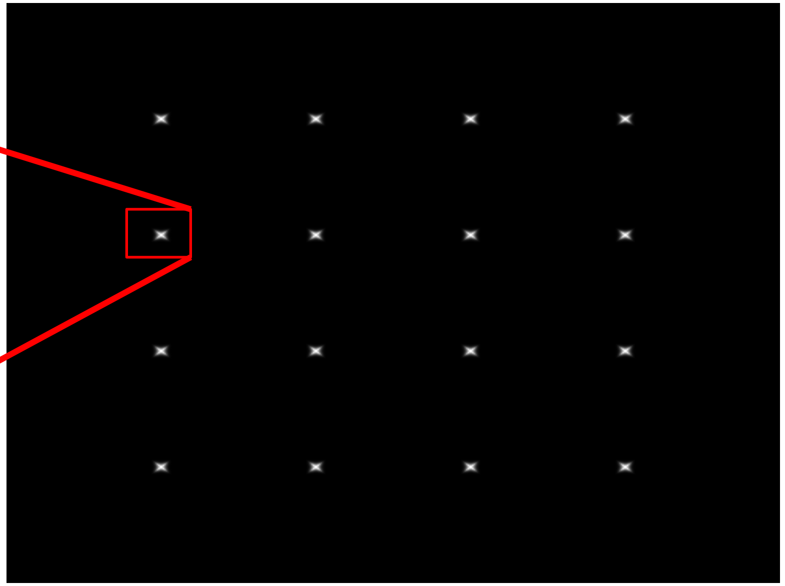
Slide source: Ali Farhadi

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





λ_-

Slide source: Ali Farhadi

Harris Corner Detection

Paper:

Chris Harris and Mike Stephens,

“A Combined Corner and Edge Detector”, 1988

Harris Corner (curvature) Response

Look at the Eigen values of H matrix only!

- $R = \det H - k\,(\mathrm{trace} H)^2$

$$\det H = \lambda_1 \lambda_2$$
$$\mathrm{trace} H = \lambda_1 + \lambda_2$$

(k is an empirical constant, choose k = 0.04 - 0.06)

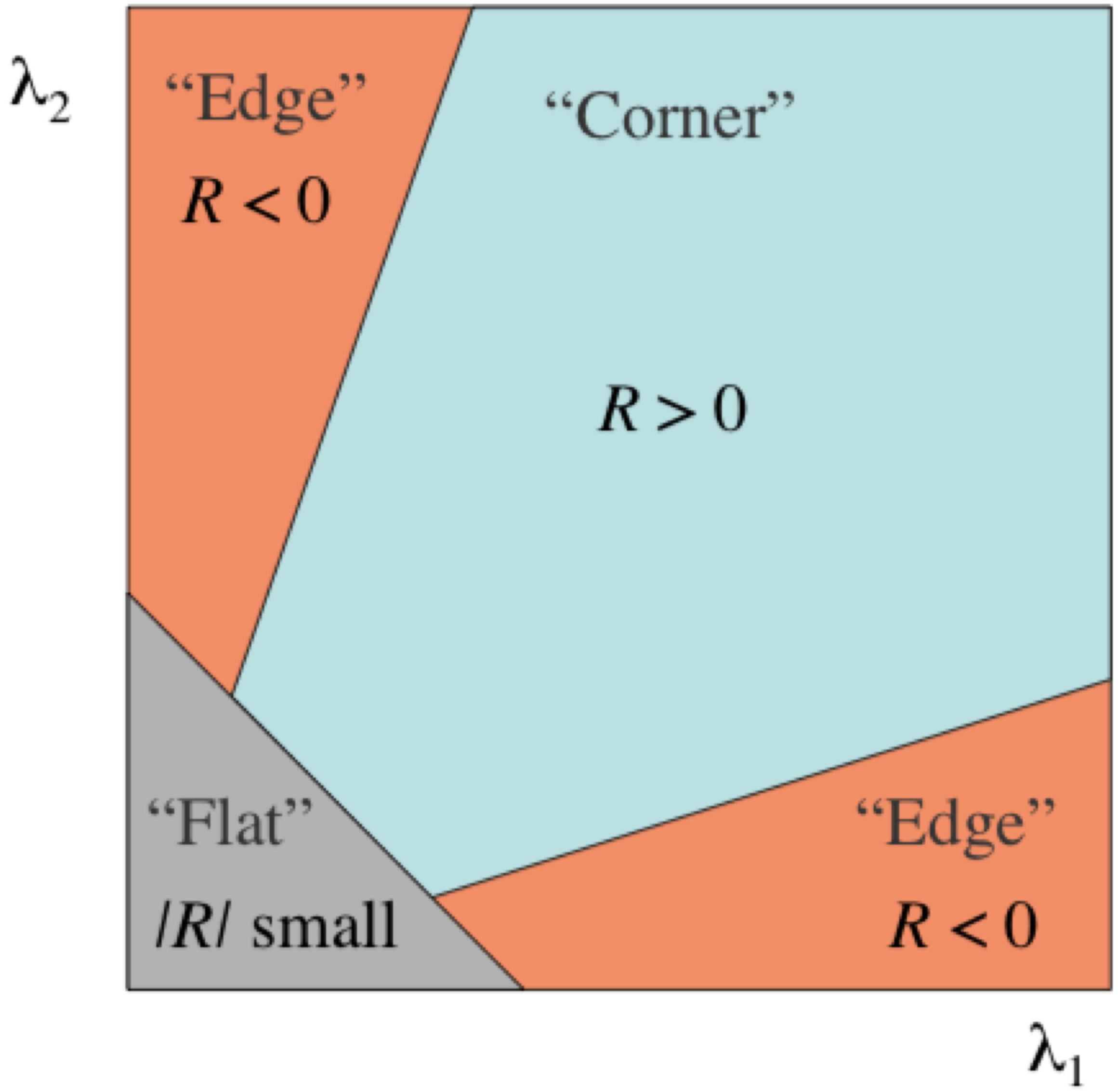
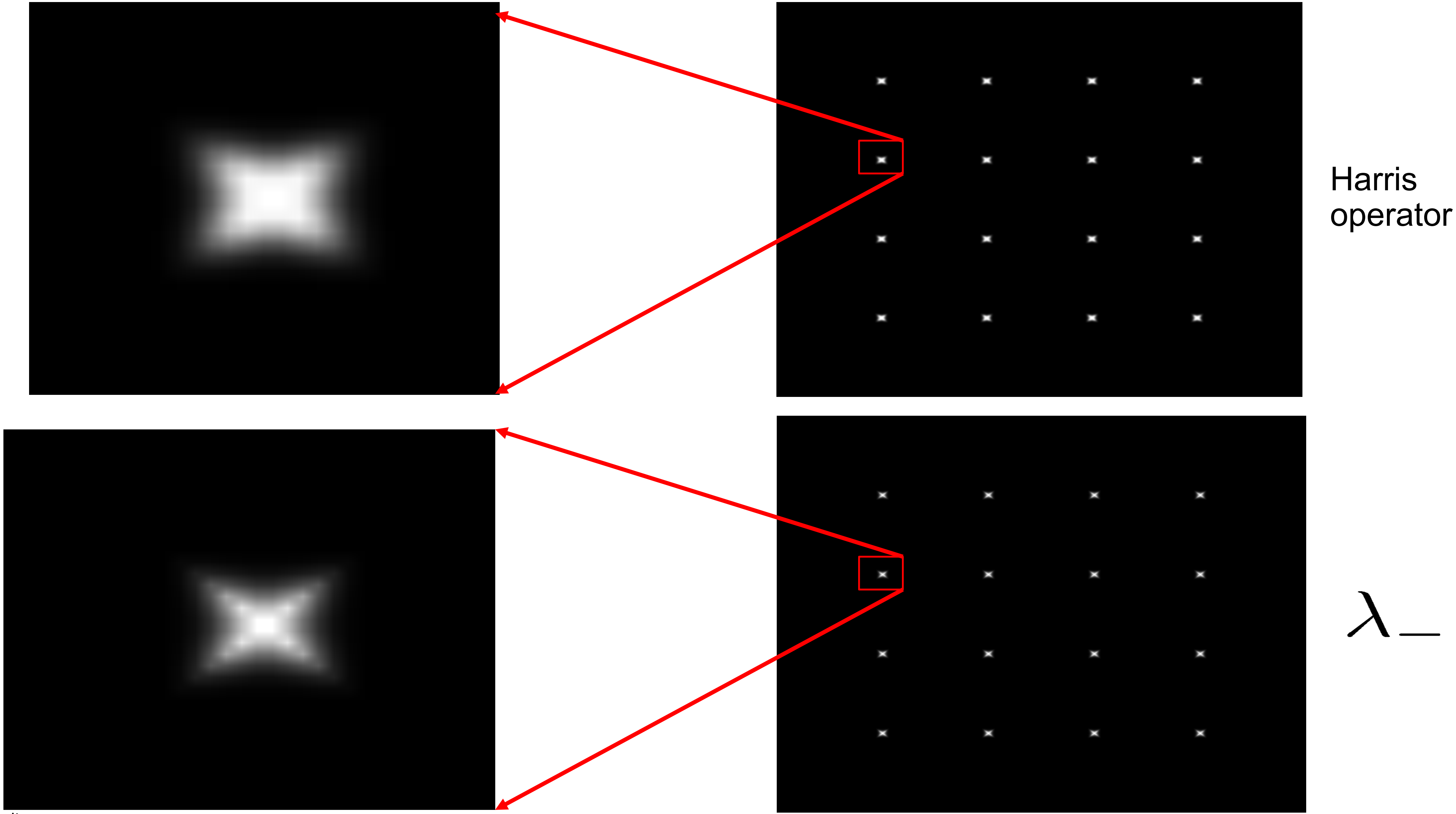


Image source: https://courses.cs.washington.edu/courses/cse576/06sp/notes/HarrisDetector.pdf

The “Harris operator”

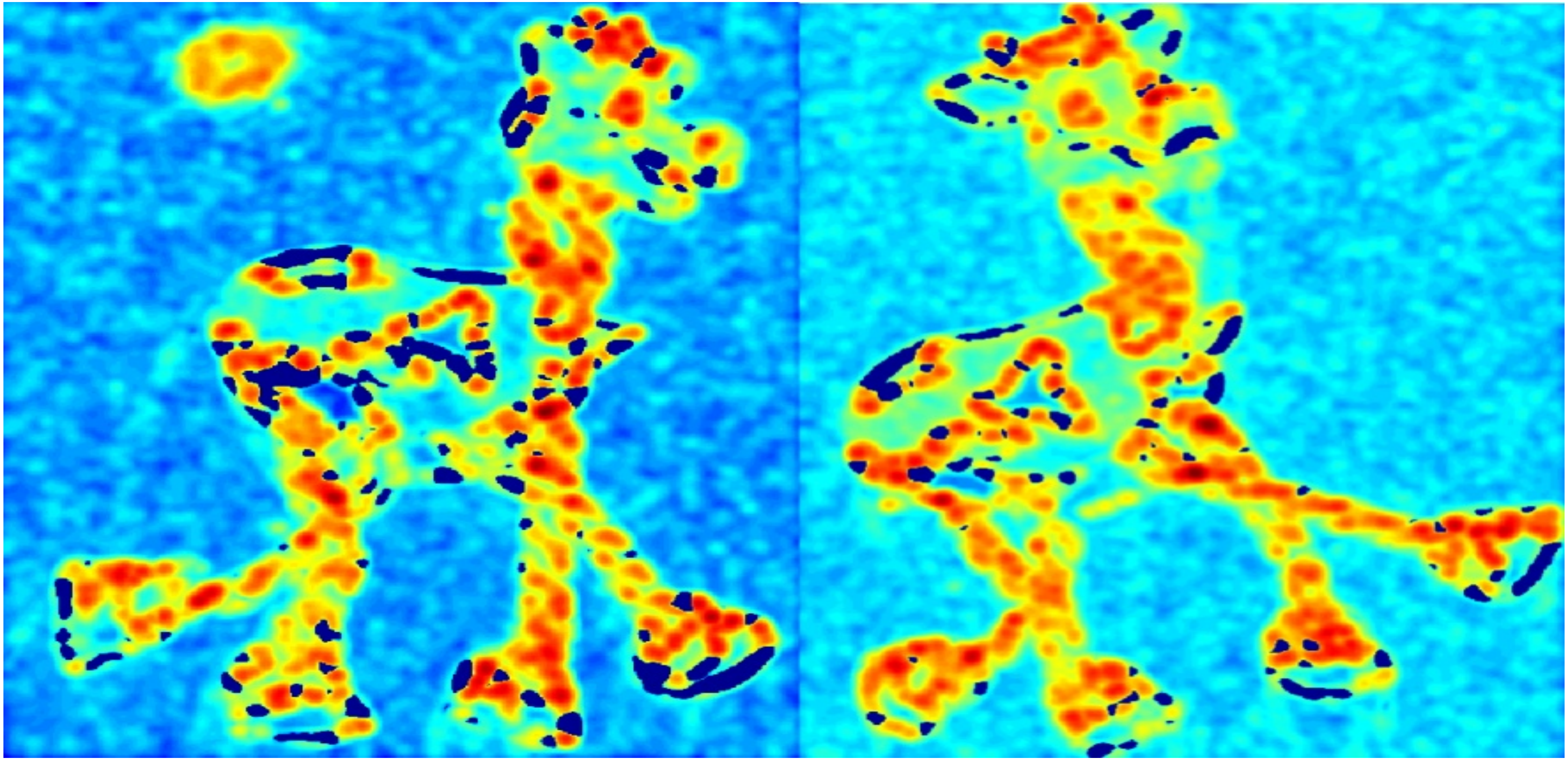


Slide source: Ali Farhadi

Harris (corner) detector example

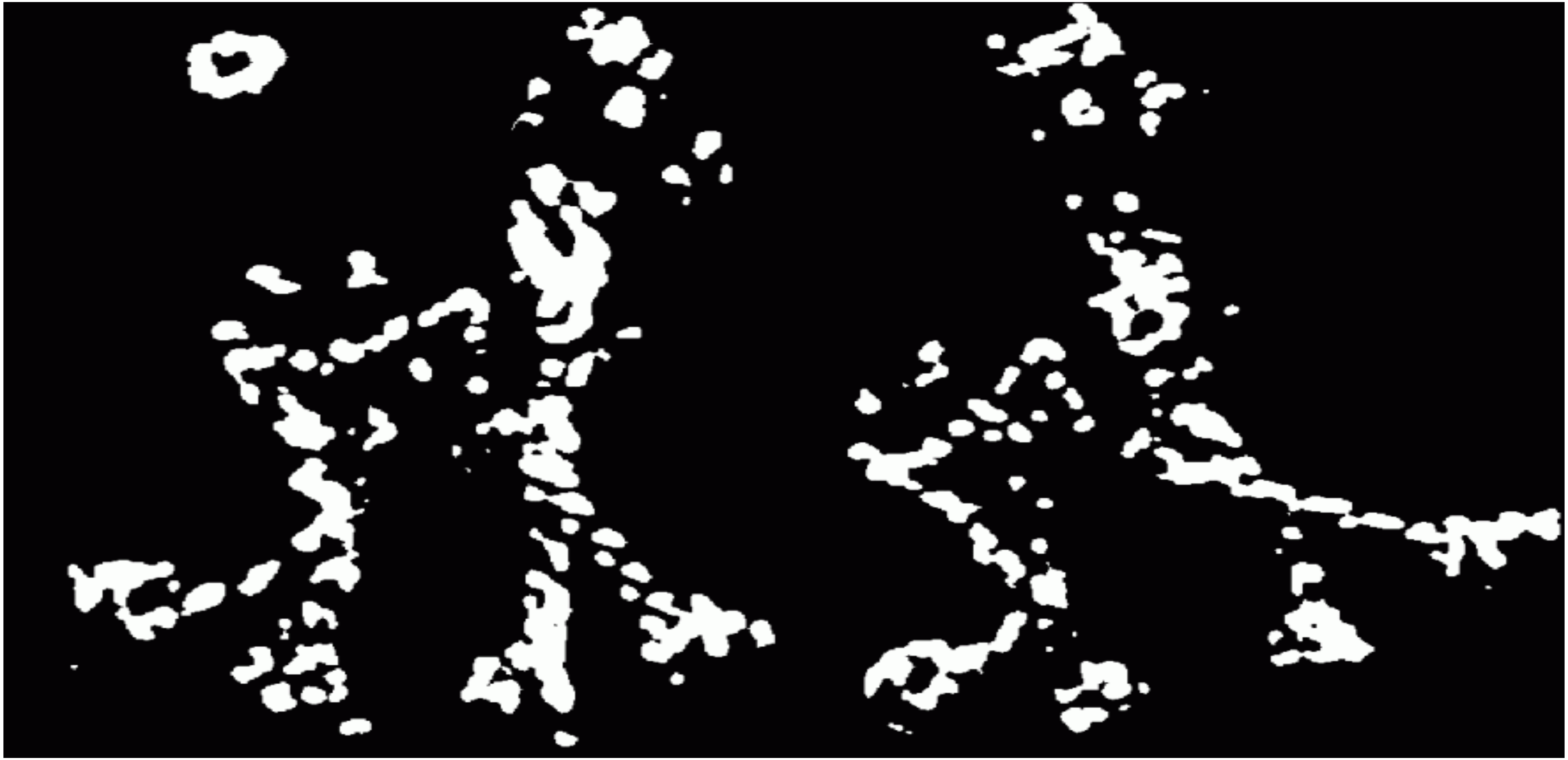
Slide source: Ali Farhadi

R value (red high, blue low)



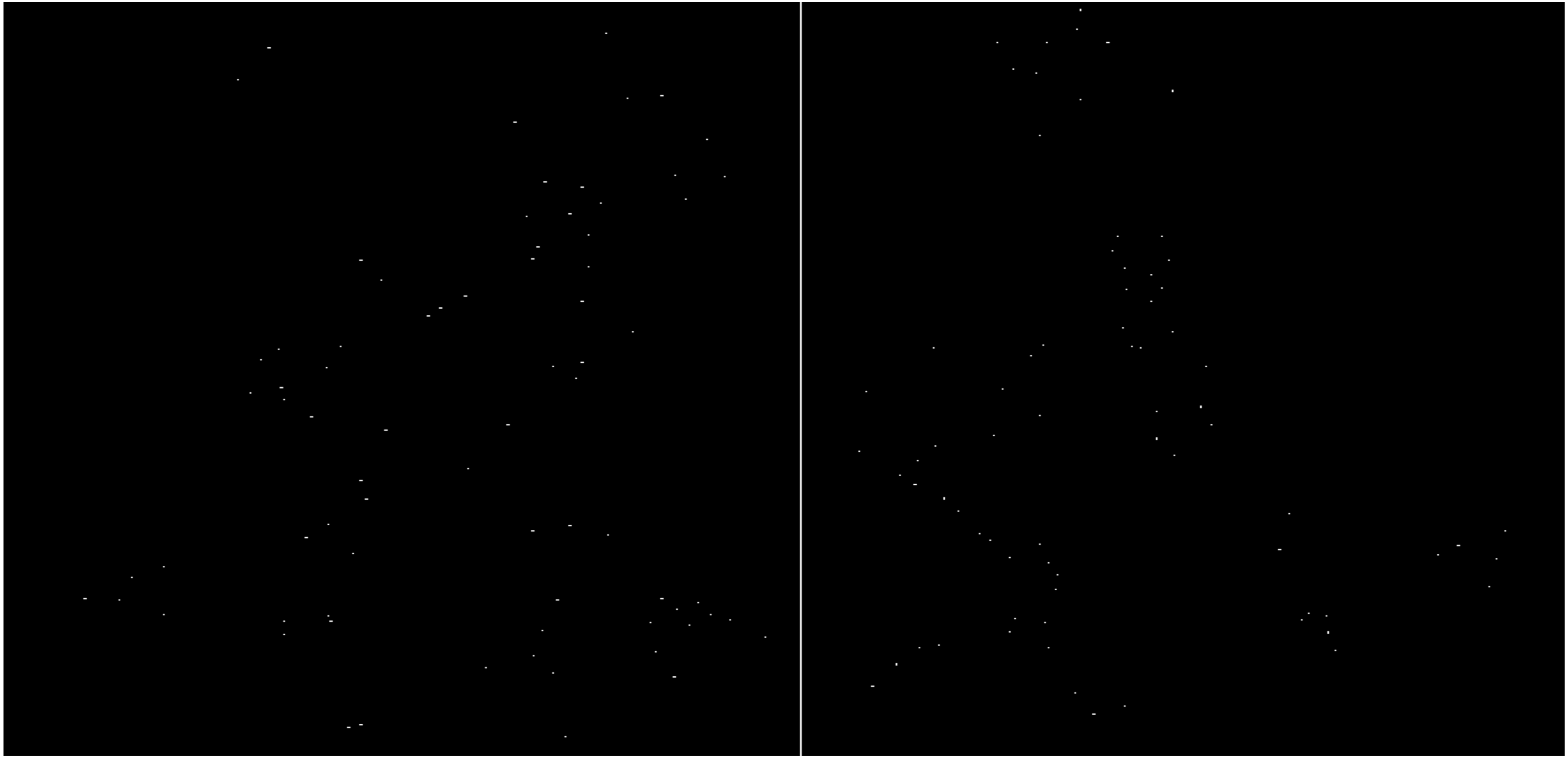
Slide source: Ali Farhadi

Threshold (R > ConstantValue)



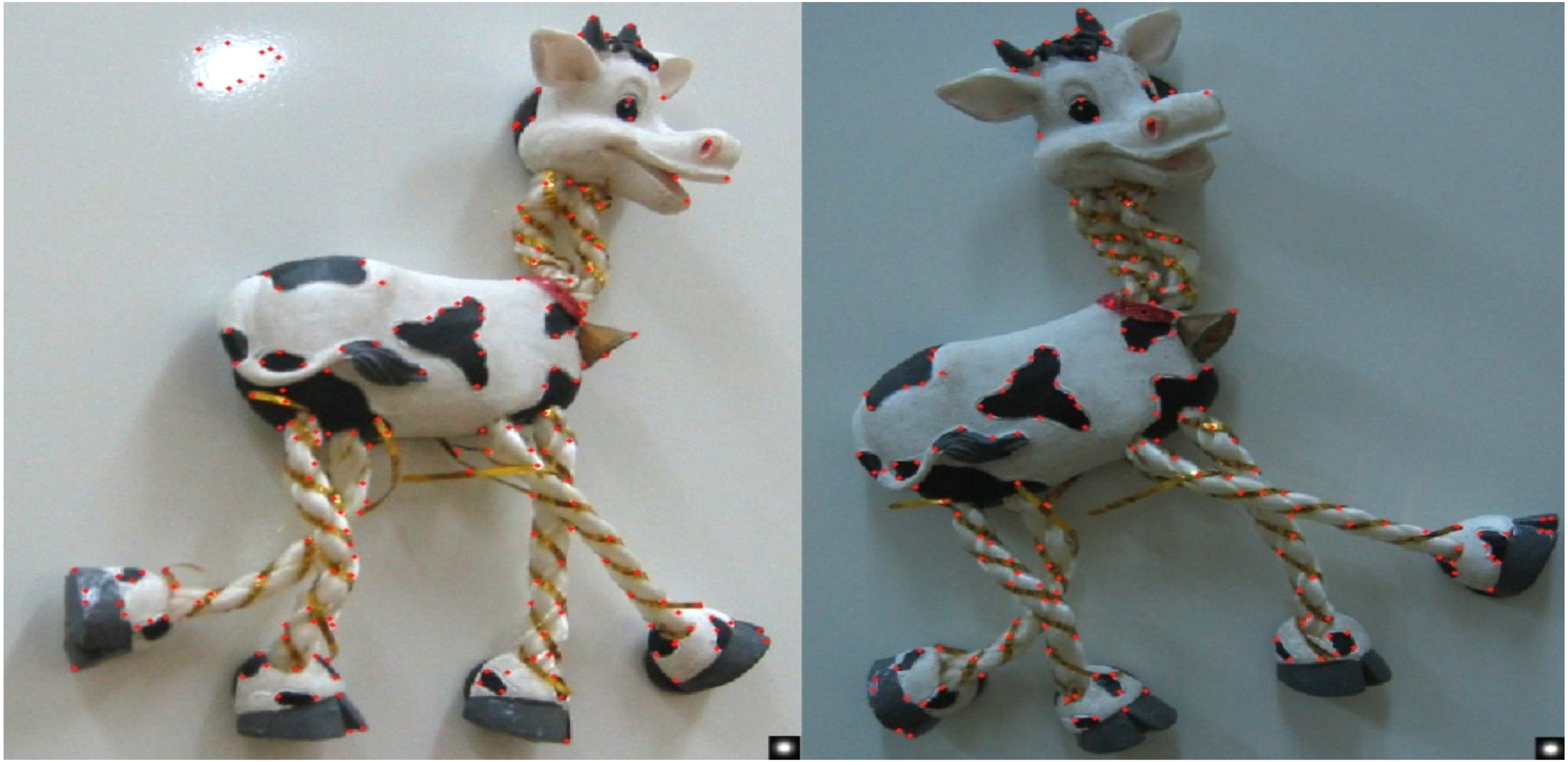
Slide source: Ali Farhadi

Find local maxima of R



Slide source: Ali Farhadi

Harris features (Shown in red)



Slide source: Ali Farhadi

Harris Corner Detection- Steps:

- Compute the Gaussian derivatives for each pixel
- Compute the Hessian matrix (also known as: second moment matrix *M)* in a Gaussian window around each pixel
- Compute the R values for each pixel
- Threshold *R*
- Apply **non-maximum suppression**

What about invariance?

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

- Will you still pick up the same features?

What if you change the brightness?

- Scale?
  - Key idea: find scale that gives local maximum of R

Some known problems:

Image 1

Rotation:

Image 2

Scaling:

200 x 150

400 x 300

800 x 600

Occlusion: