
Computer vision

Computer Vision

Lecture 2: Features I

Dr. Dina Khattab

dina.khattab@cis.asu.edu.eg

Scientific Computing Department

Instructor:	Dr. Dina Khattab
Email:	dina.khattab@cis.asu.edu.eg
Office:	Main Building – 4th floor – Room 302
Office Hours:	Monday 12:00 - 2:00 PM Thursday 11:00 AM to 12:00 PM

Agenda

- Features
  - Harris (Feature Detector)

FEATURES

Image Stitching

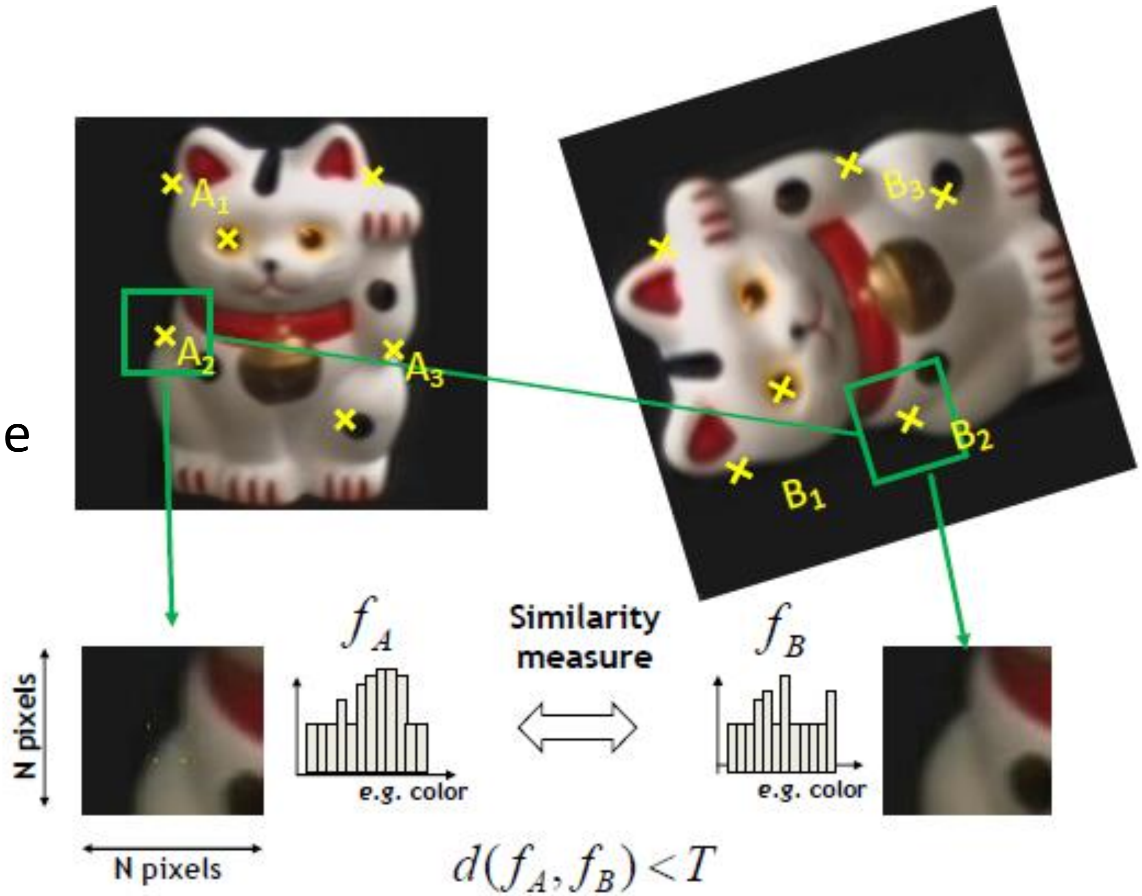
Image matching: a challenging problem

Image matching: a challenging problem

Harder Case

General Approach

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors



Features / Interest Points

- Some information computed by finding local distinctive structures in the image (*feature*) and describe them by surrounding region (e.g., a small patch centered on the detected feature).

Feature Descriptor

- Is a representation of an image that simplifies the image by extracting useful information and throwing away extraneous information.

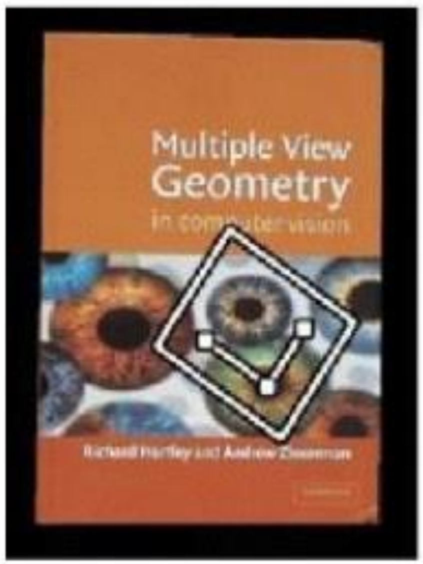
- converts an image of size width x height x 3 (channels) to a feature vector / array of length n

Challenges of designing a feature descriptor

- Invariant to photometric transformations (lighting changes, noise, and blur)

Challenges of designing a feature descriptor

- Invariant to geometric transformations

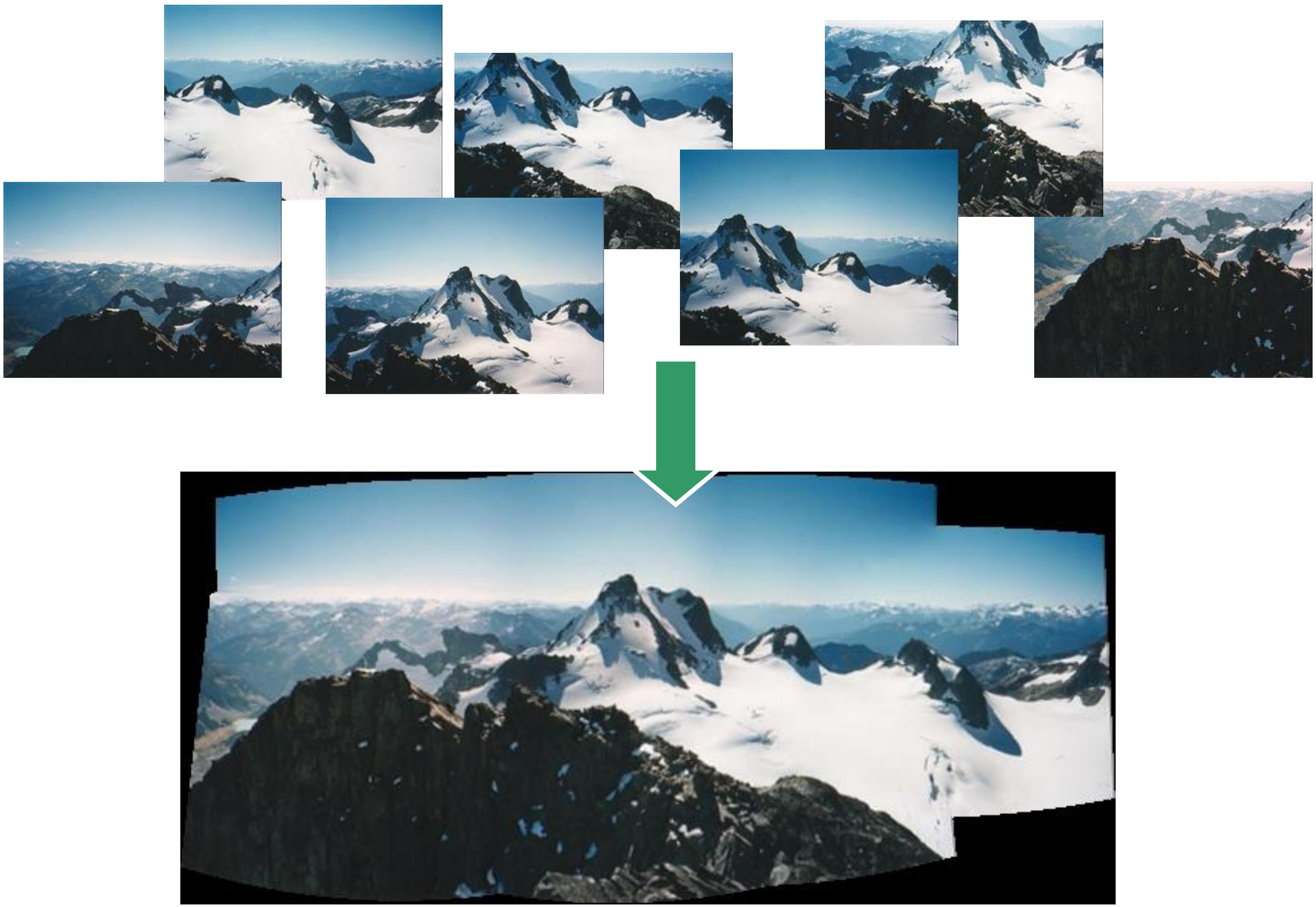


objects will appear at different scales, translation and rotation

More motivation…

- Feature points are used also for:
  - Object detection
  - Object recognition
  - Motion tracking
  - Indexing and database retrieval
  - Robot navigation
  - … other

Image Stitching
Finding a Panorama



[Microsoft Digital Image Pro version 10]

How do we build panorama?

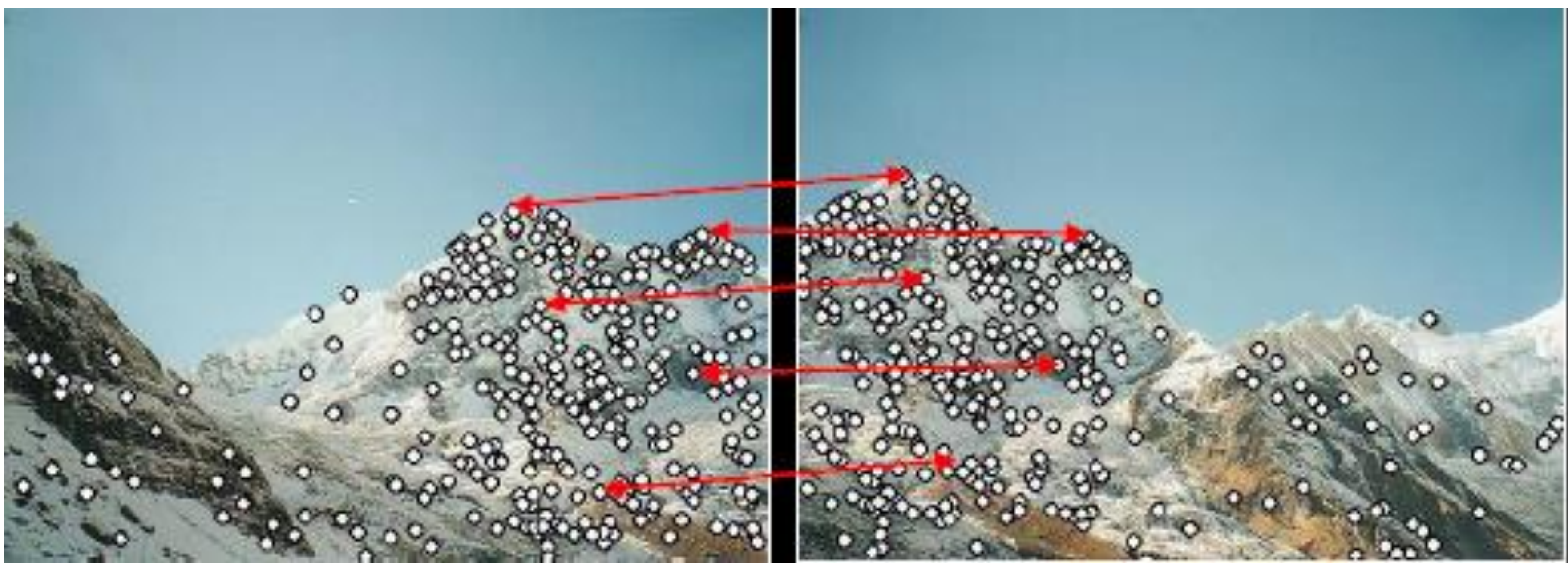
- We need to match (align) images

Detecting Features

- Detect features (feature points) in both images.

Matching with Features

- Detect features (feature points) in both images.
- Match features - find corresponding pairs.



Feature Alignment

- Detect features (feature points) in both images.
- Match features - find corresponding pairs.
- Use these pairs to align images

Detecting Features

- Problem 1:

Detect the same point independently in both images

no chance to match!

We need a repeatable detector

Characteristics of good features

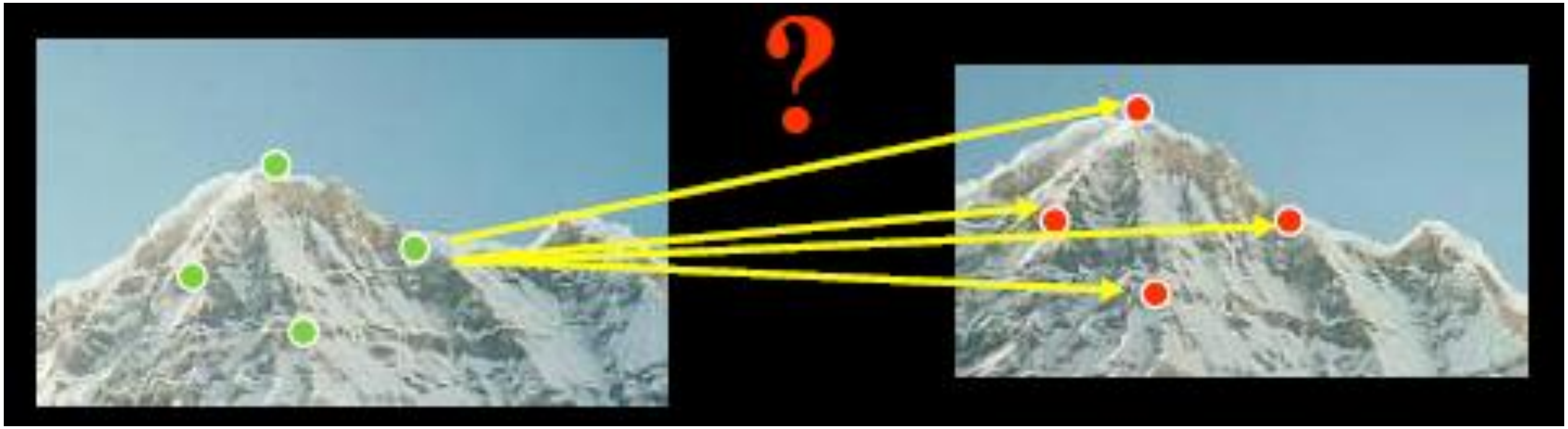
 

- *Repeatability/Precision*
- The same feature can be found in several images of the same scene despite geometric and photometric transformations

Matching with Features

- Problem 2:

For each point correctly recognize the corresponding one



We need a reliable and distinctive descriptor

Characteristics of good features

- *Saliency/Matchability*
- Each feature has a distinctive description and show a large amount of variation; this ensures that the detected features can be distinguished from each other and properly matched between different images.

Characteristics of good features

- *Quantity and efficiency*
- Many fewer features than image pixels.
- Enough features should be chosen to ensure the accuracy of technique.
- Feature matching in new images should be conducive to real-time applications.

Characteristics of good features

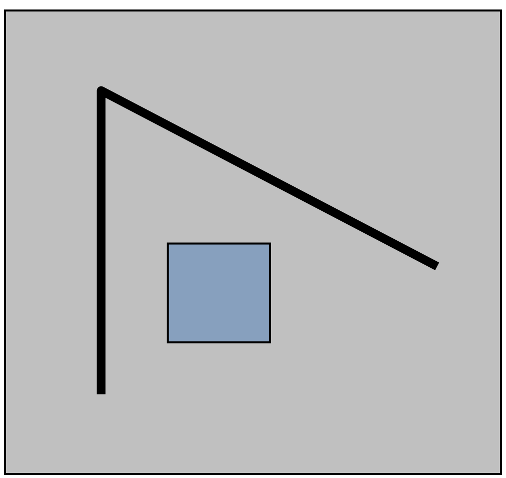
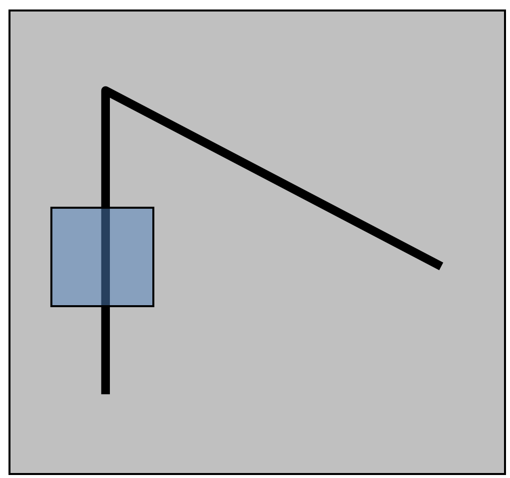
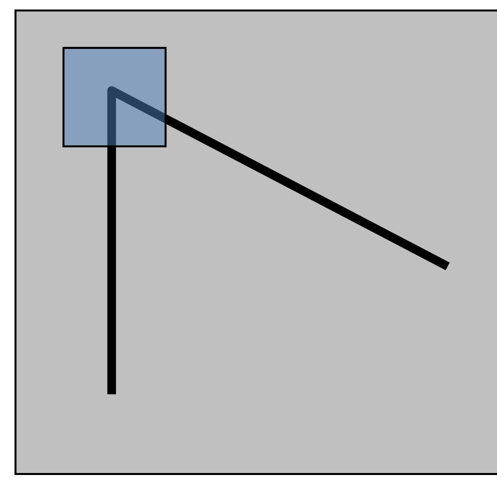
 

- *Locality*
- A feature occupies a relatively small area of the image; robust to clutter and occlusion

HARRIS DETECTOR

Local Measures of Uniqueness

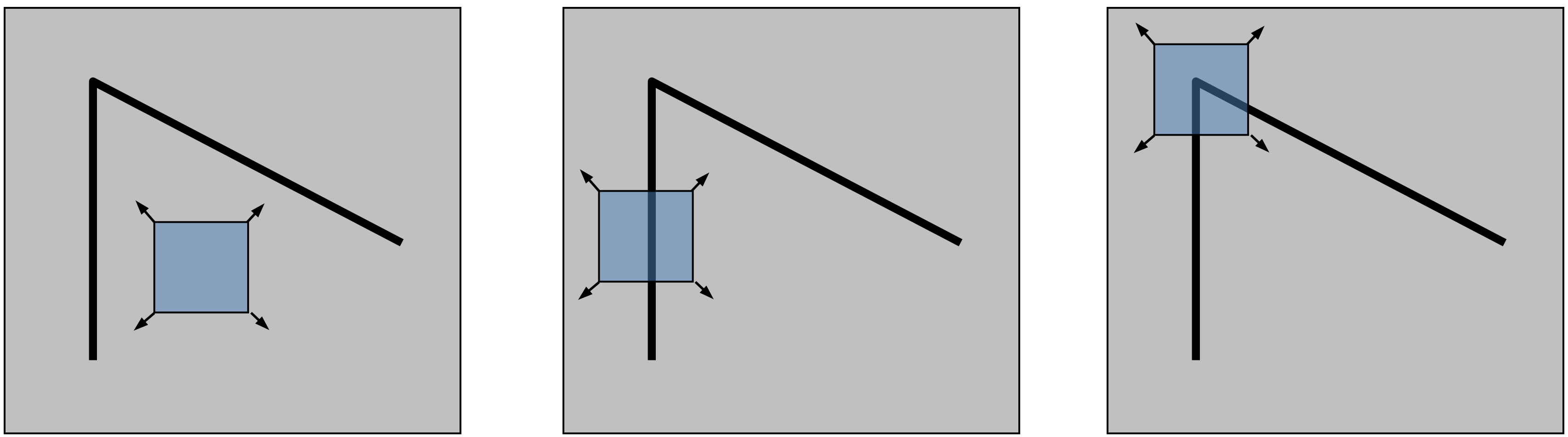
- Suppose we only consider a small window of pixels
  - What defines whether a feature is a good or bad candidate?

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*

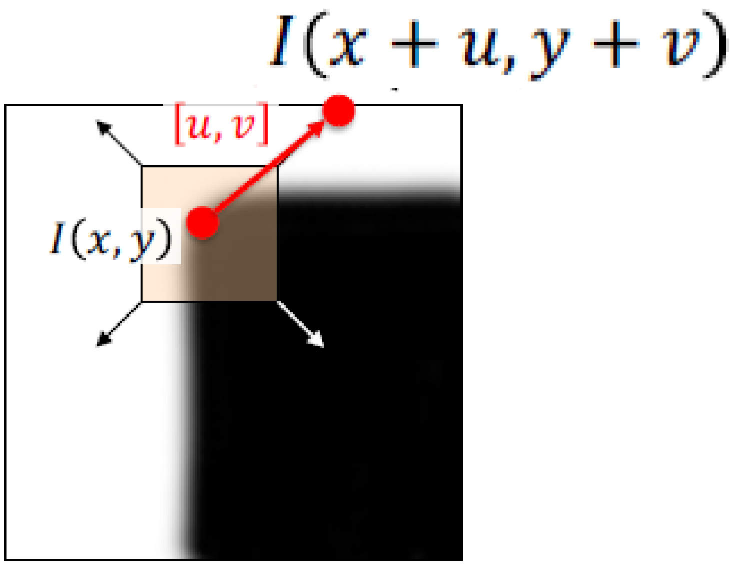


"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Measure change in all directions [u, v]



"corner":
significant change
in all directions

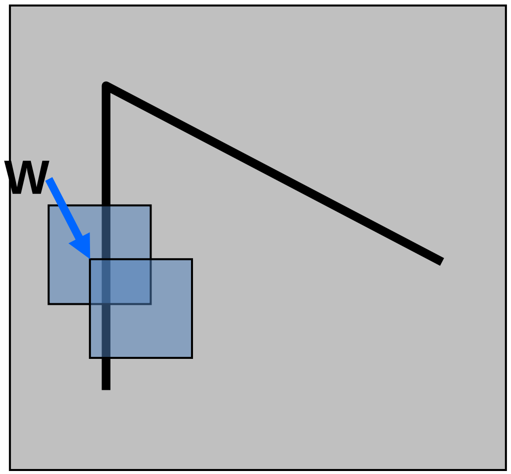
- Measure change as intensity difference:

$$I(x+u, y+v) - I(x, y)$$

- That's for a single point, but we have to accumulate over a "small window" around that point…

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)*:



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

- The intuition behind Harris detector is that if a window (w) slides over an image, the change in the intensity of pixel values caused by the shift is highest at corners

Corner Detection: Mathematics

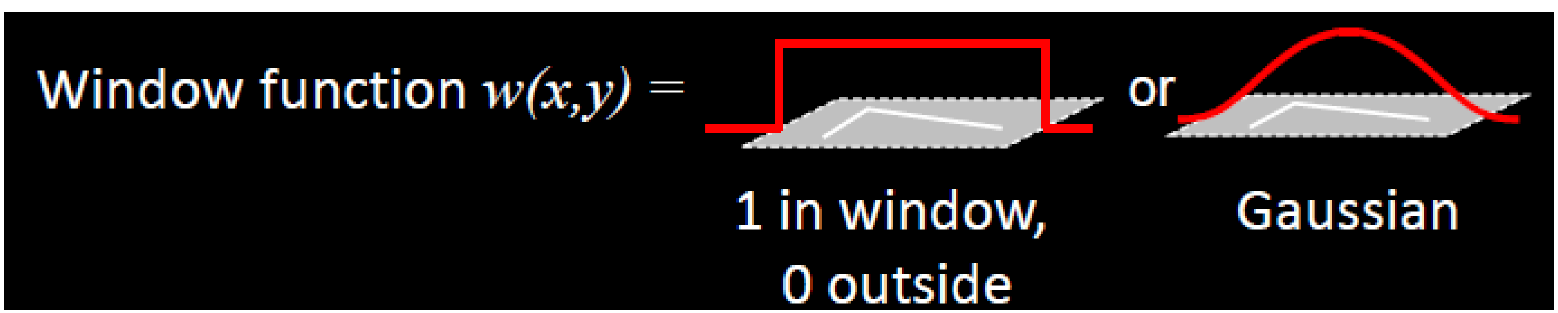
Change in appearance for the shift [*u*,*v*]:

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

Window Function

Shifted Intensity

Intensity



Corner Detection: Mathematics

Change in appearance for the shift [*u,v*]:

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

We want to find out how this function behaves for ***small*** shifts (u,v near 0,0)

Second-order Taylor expansion of *E*(*u,v*) about (0,0) (local quadratic approximation for small *u,v*):

Corner Detection: Mathematics

- The quadratic approximation simplifies to

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

- where *M* is a *2x2 second moment matrix* computed from *image derivatives* and smoothed by *Gaussian weights*.

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

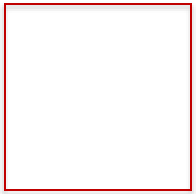
Gradient with respect to x, times gradient with respect to y

Sum over image region – the area we are checking for corner

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y]$$

What Does This Matrix Reveal?

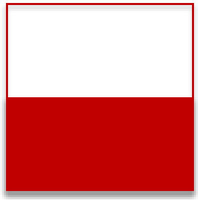
$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$



$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Small

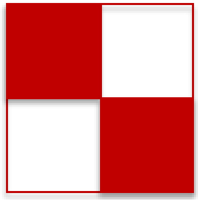
Nothing



$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Large

Edge



$\sum I_x^2 \longrightarrow$ Large

$\sum I_y^2 \longrightarrow$ Large

Corner

What Does This Matrix Reveal?

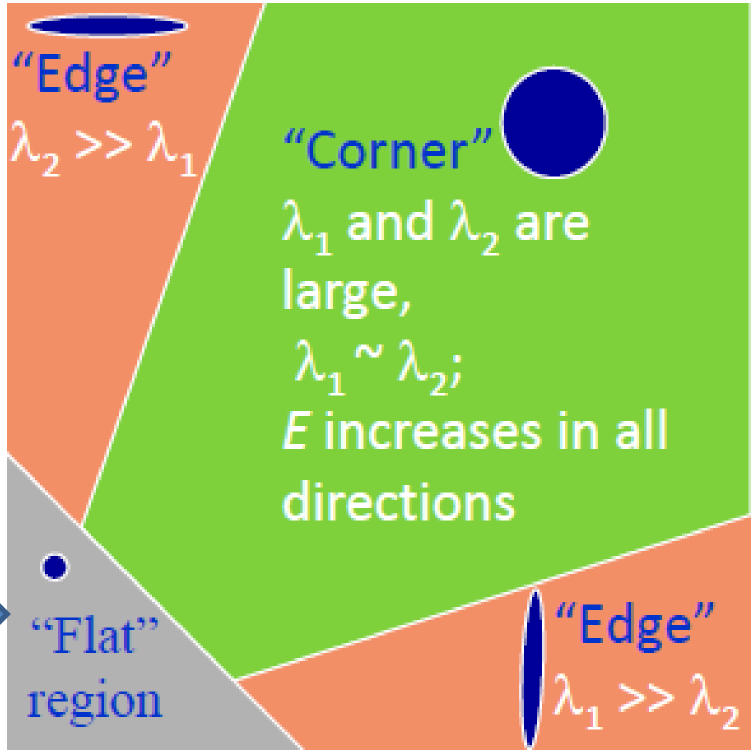
$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R \qquad M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

(Eigenvalue decomposition)

- This means:
  - If either λ is close to 0, then this is not a corner, so look for locations where both are large.

Interpreting the eigenvalues

- Classification of image points using eigenvalues of M:



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

Corner response function

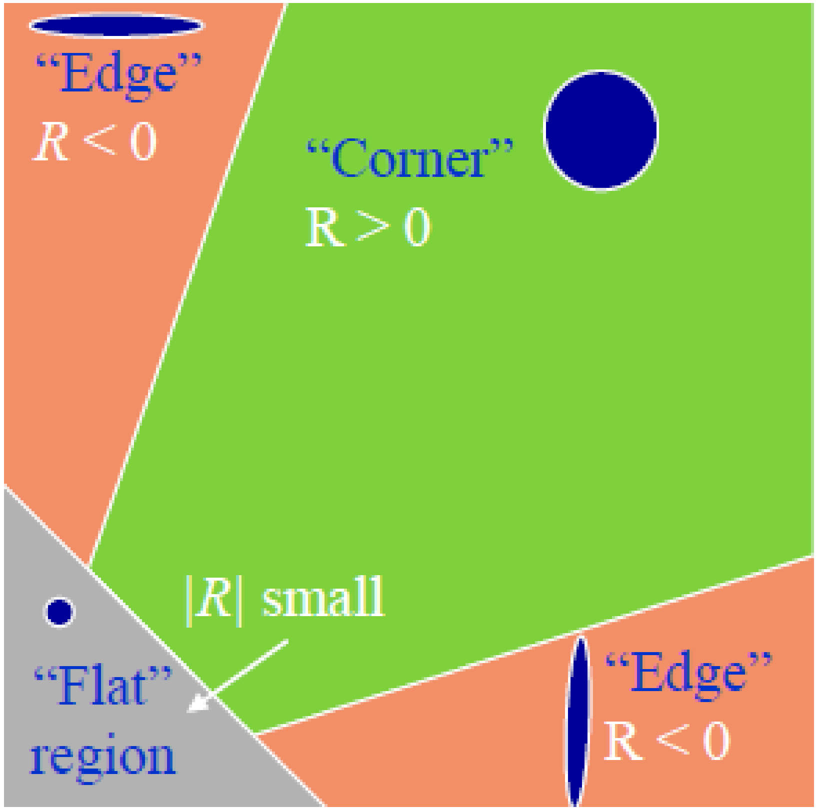
R = det(M) - α trace $(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$

α : constant (0.04 to 0.06)

$$det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

$$trace\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$$

- *R* is <u>large</u> for a corner
- *R* is <u>negative</u> with large magnitude for an edge
- |*R*| is <u>small</u> for a flat region



Harris Detector: Algorithm

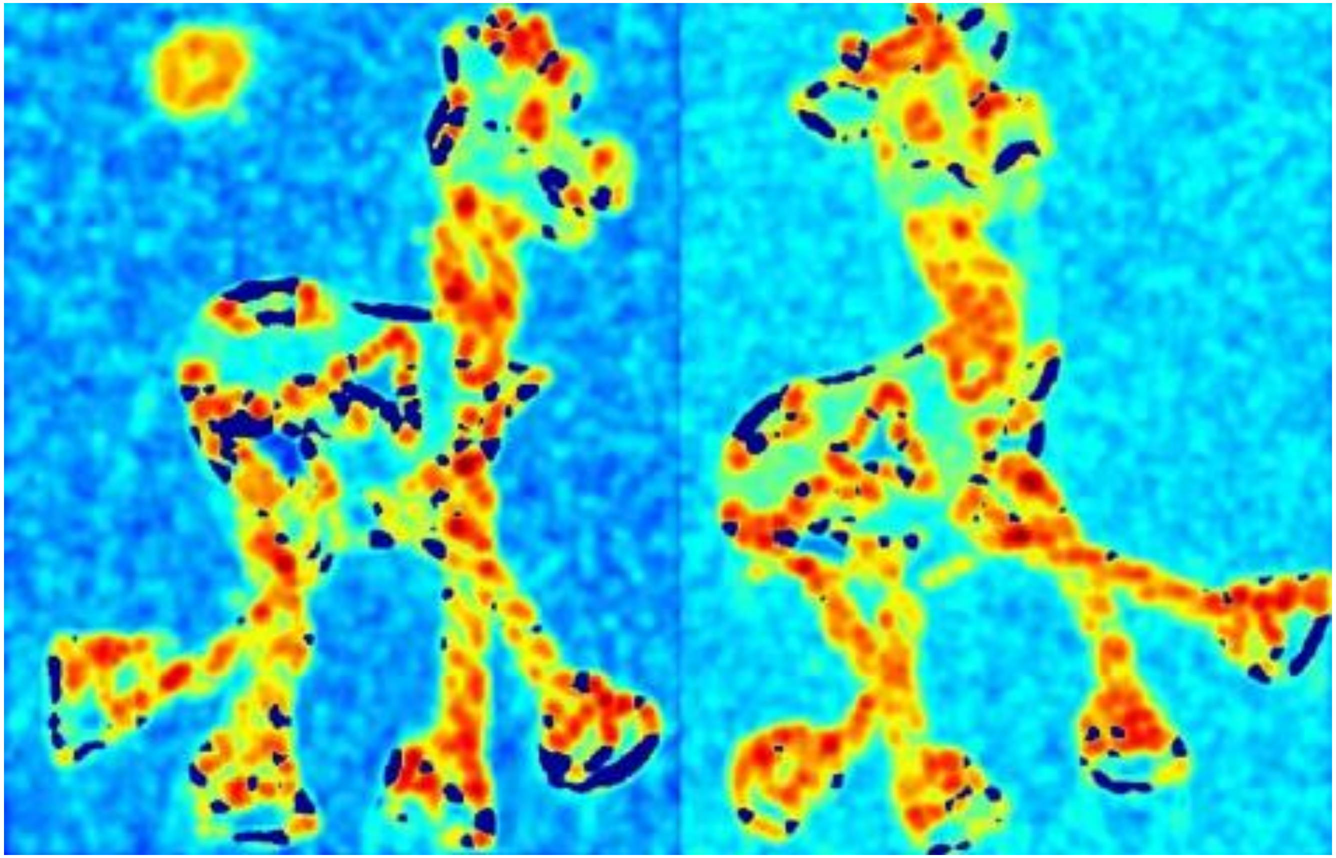
1. Compute derivatives at each pixel
2. Compute second moment matrix M in a Gaussian window around each pixel
3. Compute corner response function R
4. Threshold R
5. Find local maxima of response function (non-maximum suppression)

C. Harris and M. Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris Detector: Workflow

Harris Detector: Workflow

- Compute corner response R

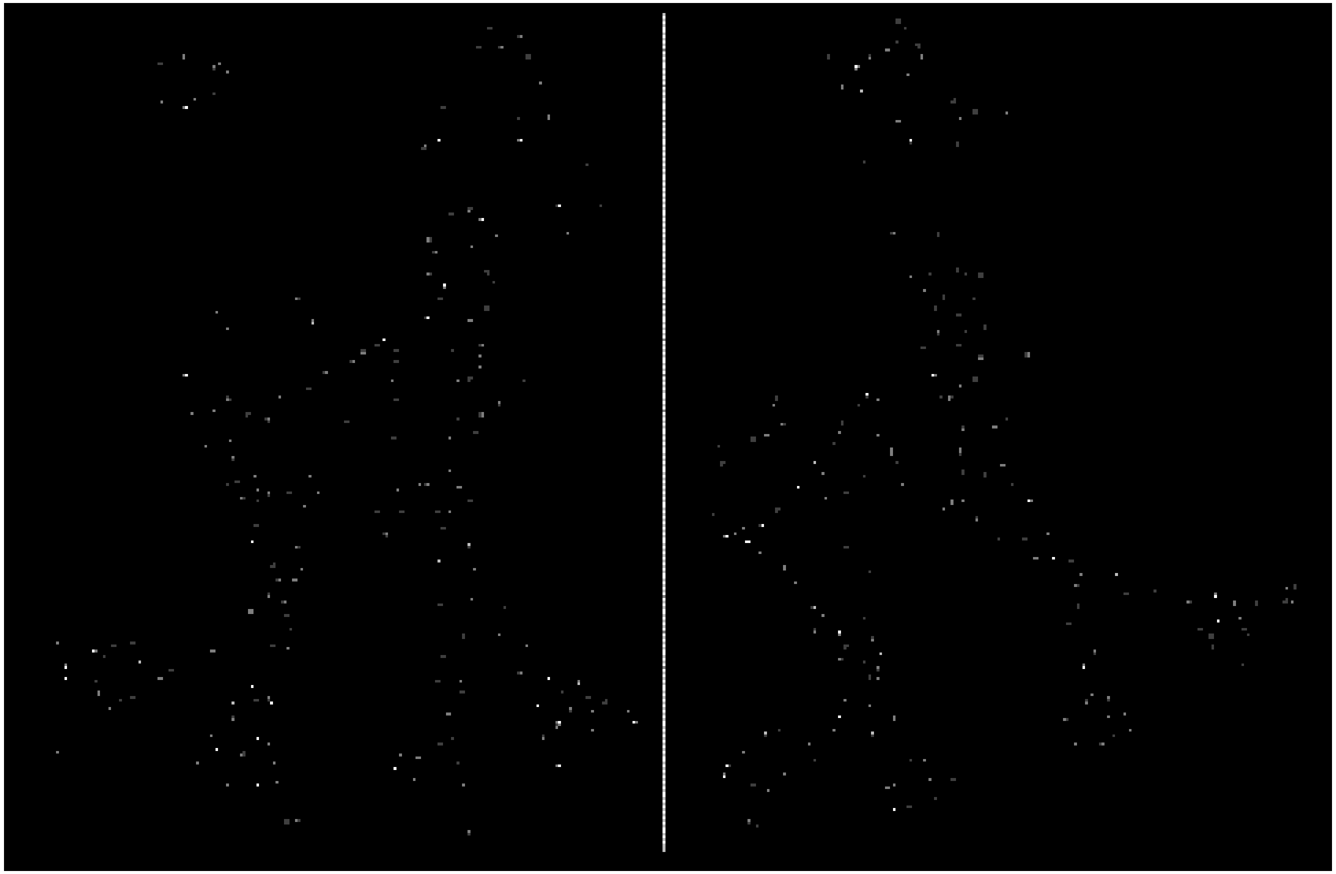


Harris Detector: Workflow

- Find points with large corner response: R>threshold

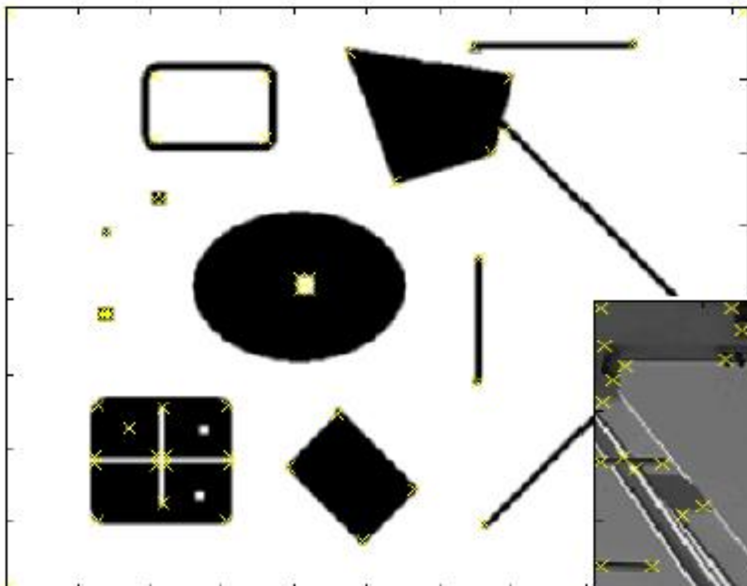
Harris Detector: Workflow

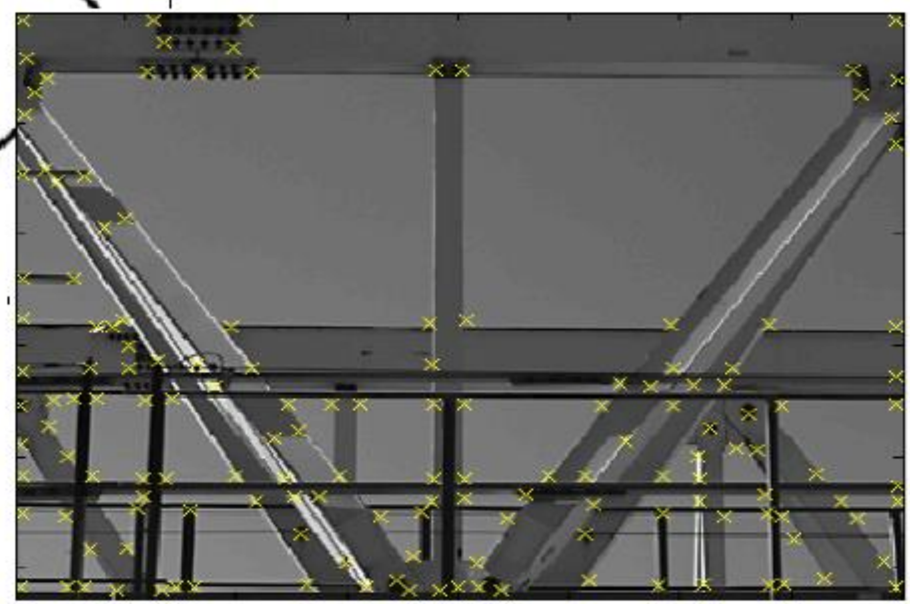
- Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector – Responses [Harris88]



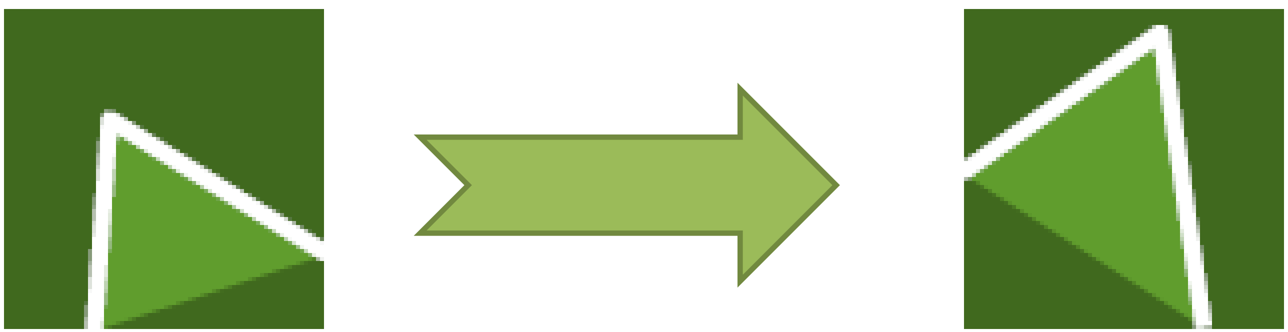


Effect: A very precise corner detector.

Harris Detector – Responses [Harris88]

Harris Detector: Some Properties

- Rotation invariance?

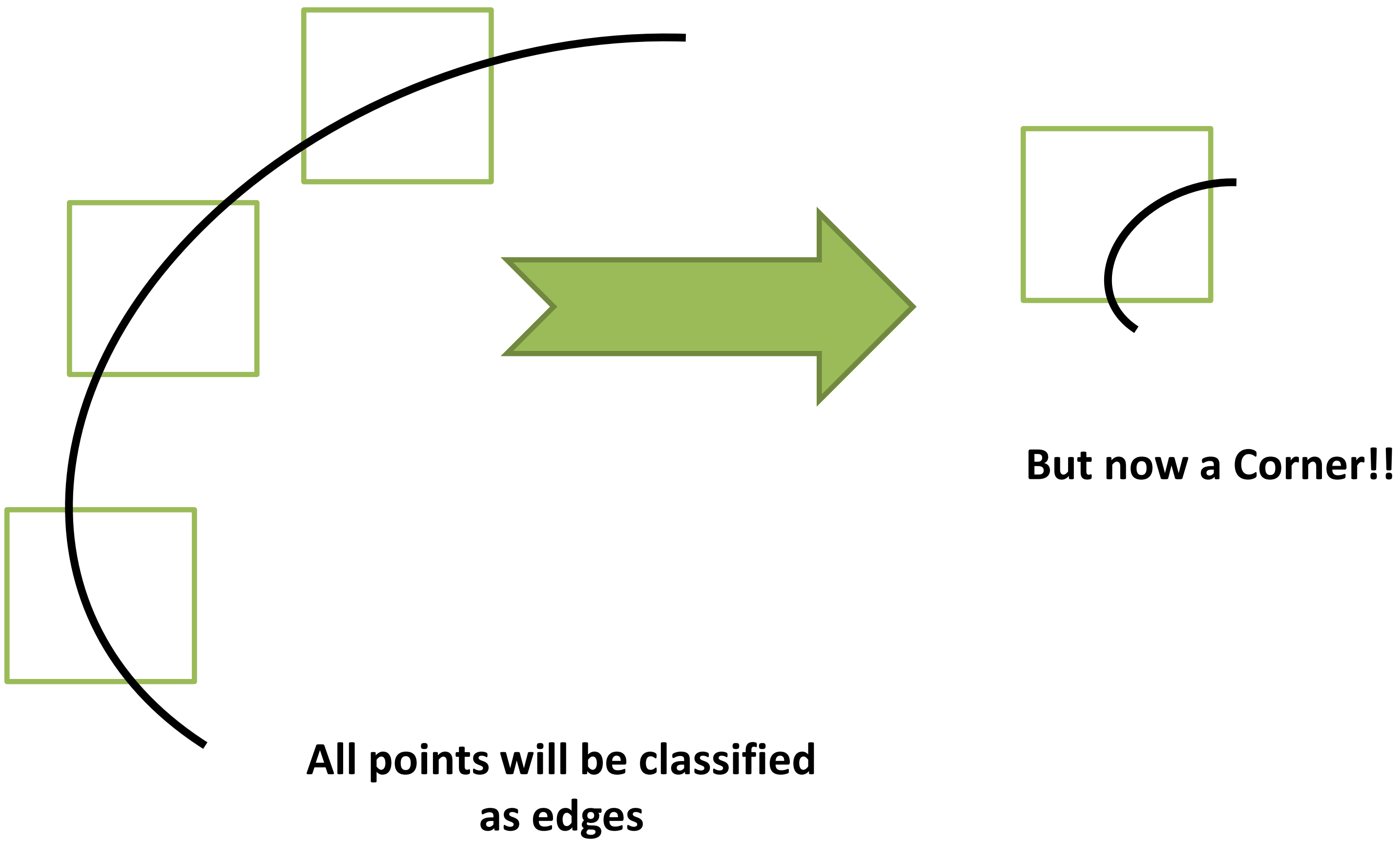


- The shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Harris Detector: Some Properties

- Not invariant to *image scale*!



Credit for

CS 4495 "Computer Vision" (Spring 2015) by A. Bob - College of Computing, Georgia Tech

CS131 "Computer Vision: Foundations and Applications" by University of Stanford (Fall 2019)

"Advanced Machine Learning Specialization" by National Research University Higher School of Economics, Russia