Features

Joshi Pratik

What is a feature?

- A **feature** is the property of an object. Eg: Size of an object, Shape of an object.
- In Image Processing and Computer Vision, features may be specific structures in the image such as points, edges or objects.

Goal

- When the multiple images of the same scene are provided, find points in an image that can be:
  - Found in other images as well
  - Found precisely – well localized
  - Found reliably – well matched

Why do we need them?

- Want to compute a fundamental matrix to recover geometry.
- Robotics/Vision: See how a bunch of points move from one frame to another. Allows computation of how camera moved-> depth->moving objects.
- Build a panorama...

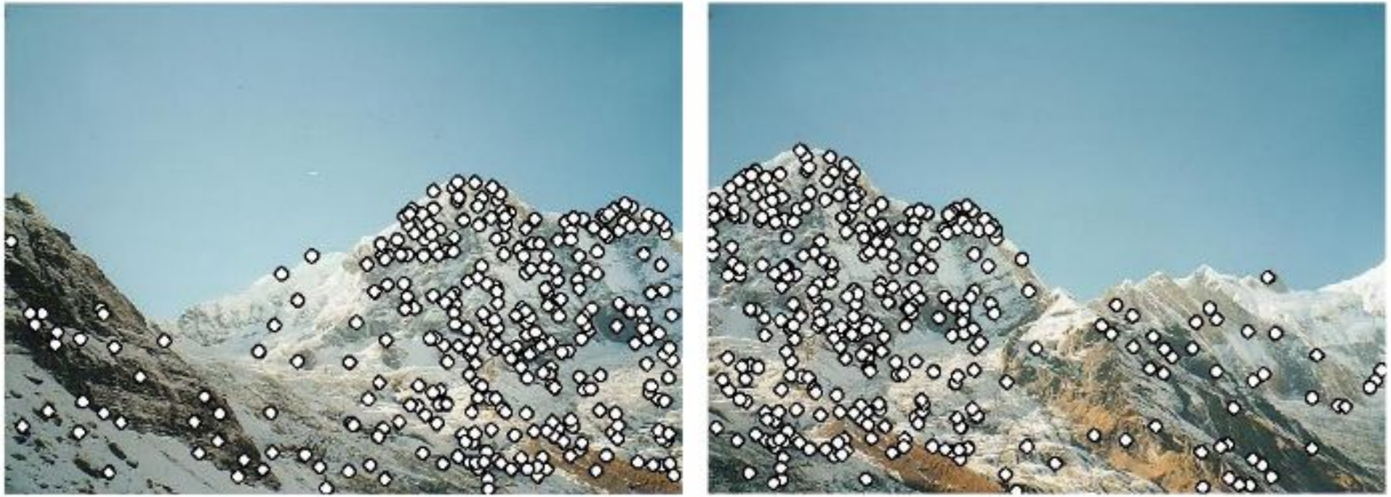
Suppose you want to build a panorama

How do we build panorama?

- We need to match(align) the images.

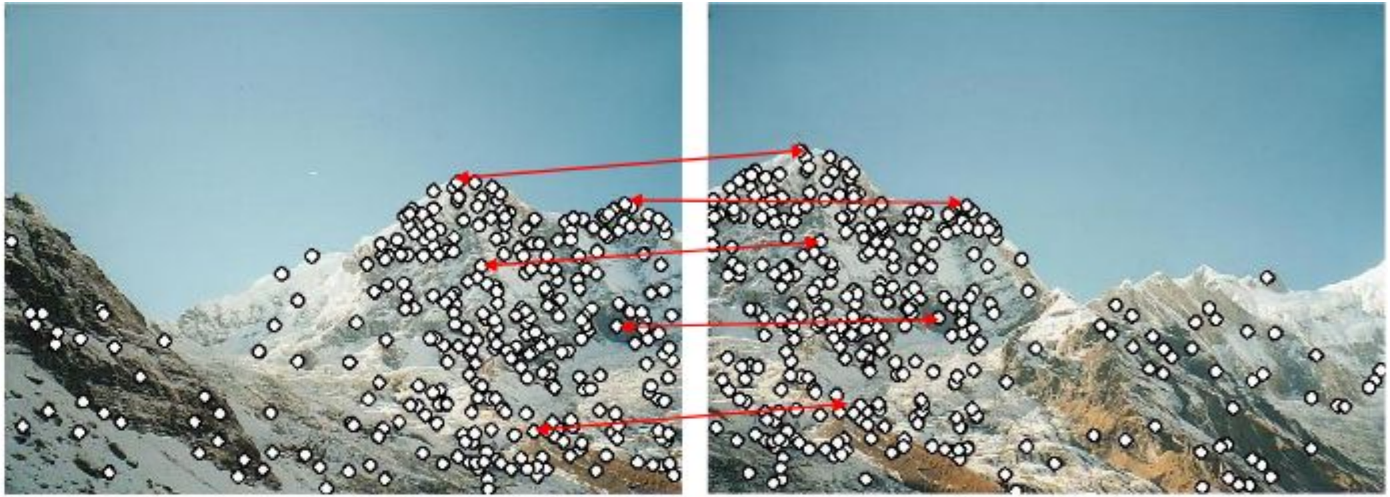

Matching with Features

- Detect features (feature points) in both images.



Matching with Features

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



Matching with Features

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

Matching with Features

- Problem 1:
  - Detect the same point independently in both the images.

no chance to match!

Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding one

More motivation...

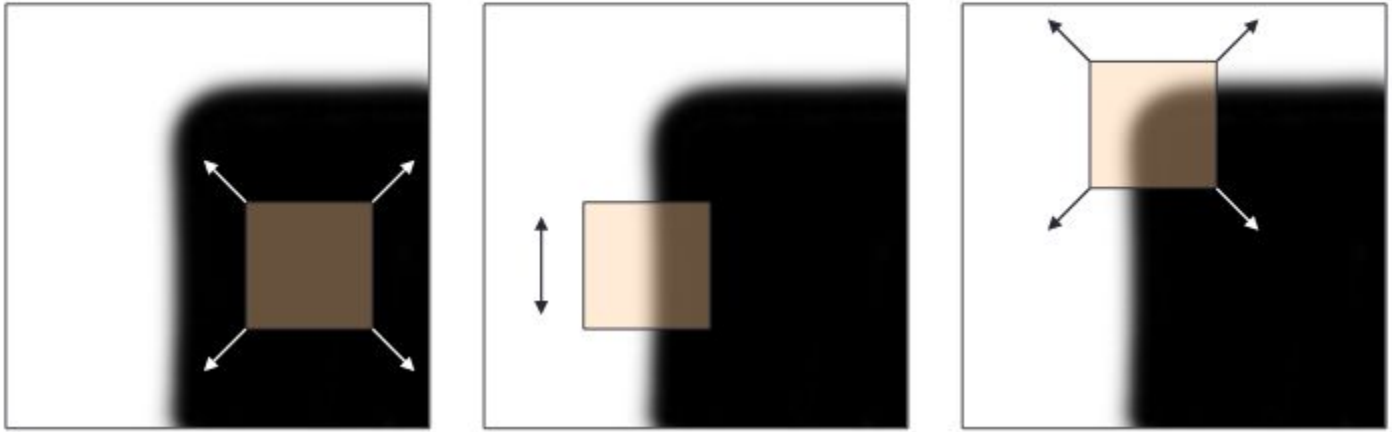
- Feature points are used also for:
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Robot navigation
  - Other...

Characteristics of good features

- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Matchability
  - Each feature has a distinctive description
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to occlusion

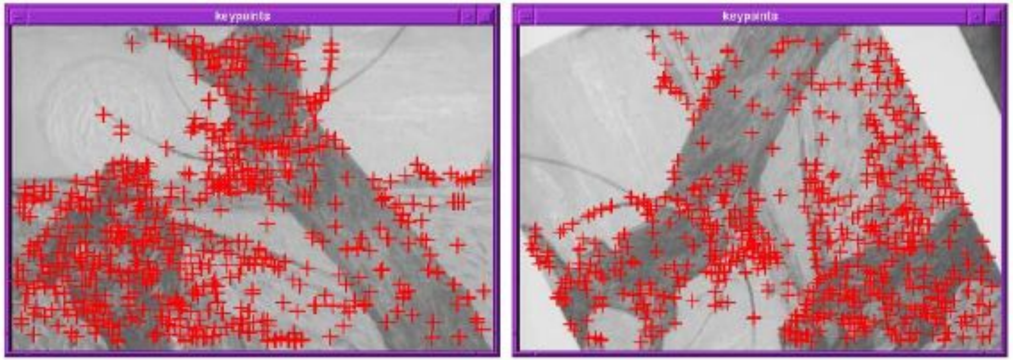
Corner Detection: Basic Idea

- We should be able to recognize the point by looking through a small window.
- Shifting a window in any direction should give a large change in intensity.



Finding Harris corners

- *Key property*: in the region around a corner, image gradient has two or more dominant directions.



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

Corner Detection

Change in appearance for the shift [u,v]

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

Window function

Shifted intensity

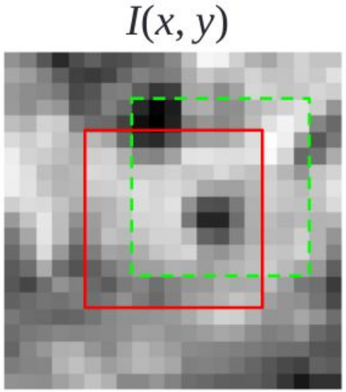
Intensity

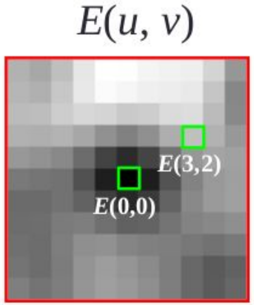
We're looking for windows that produce a large E value.

Corner Detection

Change in appearance for the shift [u,v]

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





Second order taylor series

For 1-D : $F(\delta x) \approx F(0) + \delta x \cdot \frac{dF(0)}{dx} + \frac{1}{2}\delta x^2 \cdot \frac{d^2F(0)}{dx^2}$

Here

$$E(u,v) \approx E(0,0) + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Corner Detection

Second-order Taylor expansion of E(u,v) about (0,0):

$$E(u,v) \approx E(0,0) + [u \quad v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \quad v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Calculate:

- $E_u(u,v)$
- $E_v(u,v)$
- $E_{uv}(u,v)$

And put E(0,0)

Corner detection

$$E_u(u,v) = \sum_{x,y} 2w(x,y)\left[I(x+u,y+v) - I(x,y)\right] I_x(x+u,y+v)$$

$$E_{uu}(u,v) = \sum_{x,y} 2w(x,y) I_x(x+u,y+v) I_x(x+u,y+v)$$

$$+ \sum_{x,y} 2w(x,y)\left[I(x+u,y+v) - I(x,y)\right] I_{xx}(x+u,y+v)$$

$$E_{uv}(u,v) = \sum_{x,y} 2w(x,y) I_y(x+u,y+v) I_x(x+u,y+v)$$

$$+ \sum_{x,y} 2w(x,y)\left[I(x+u,y+v) - I(x,y)\right] I_{xy}(x+u,y+v)$$

Corner detection

Evaluate at (u,v) = **(0,0)**:

$$E(u,v) \approx \overset{=0}{\boxed{E(0,0)}} + [u \;\; v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \;\; v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

$$E_u(u,v) = \sum_{x,y} 2w(x,y)\underset{=0}{\boxed{[I(x+u,y+v) - I(x,y)]}} I_x(x+u,y+v)$$

$$E_{uu}(u,v) = \sum_{x,y} 2w(x,y) I_x(x+u,y+v) I_x(x+u,y+v)$$

$$+ \sum_{x,y} 2w(x,y)\underset{=0}{\boxed{[I(x+u,y+v) - I(x,y)]}} I_{xx}(x+u,y+v)$$

$$E_{uv}(u,v) = \sum_{x,y} 2w(x,y) I_y(x+u,y+v) I_x(x+u,y+v)$$

$$+ \sum_{x,y} 2w(x,y)\underset{=0}{\boxed{[I(x+u,y+v) - I(x,y)]}} I_{xy}(x+u,y+v)$$

Corner detection

$$E(u,v) \approx E(0,0) + [u \ \ v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \boxed{\frac{1}{2}[u \ \ v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}}$$

$$E(0,0) = 0$$

$$E_u(0,0) = 0$$

$$E_v(0,0) = 0$$

$$E_{uu}(0,0) = \sum_{x,y} 2w(x,y)I_x(x,y)I_x(x,y)$$

$$E_{vv}(0,0) = \sum_{x,y} 2w(x,y)I_y(x,y)I_y(x,y)$$

$$E_{uv}(0,0) = \sum_{x,y} 2w(x,y)I_x(x,y)I_y(x,y)$$

Corner Detection

We get,

$$E(u,v) \approx [u \ \ v] \begin{bmatrix} \sum_{x,y} w(x,y) I_x^2(x,y) & \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y) \\ \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y) & \sum_{x,y} w(x,y) I_y^2(x,y) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Corner Detection

The quadratic expression simplifies to

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

Where M is the second moment matrix, given computed by image derivatives

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

Interpreting the second moment matrix

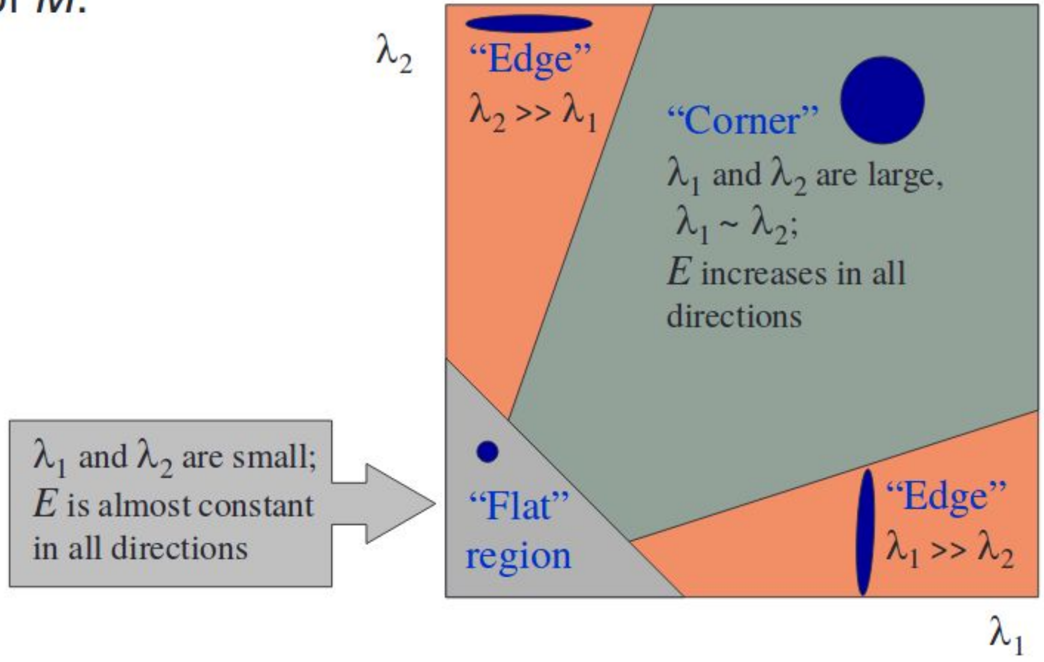
Find the eigenvalues of M. Consider the axis aligned case where gradients are either horizontal or vertical

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

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

Interpreting the eigenvalues

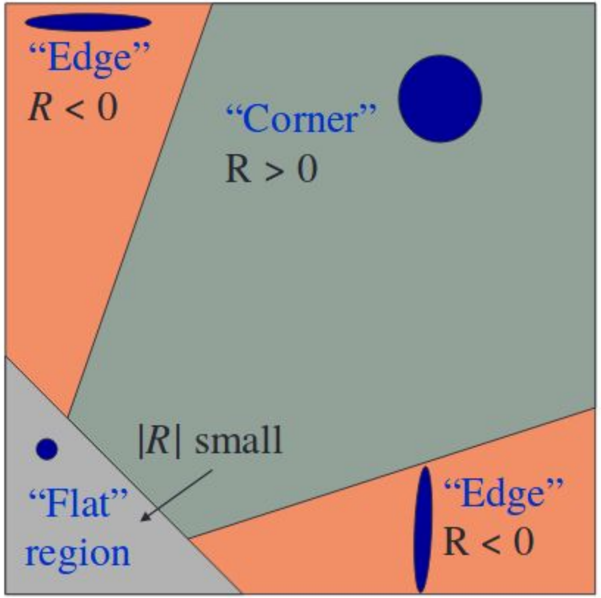
Classification of image points using eigenvalues of *M*:



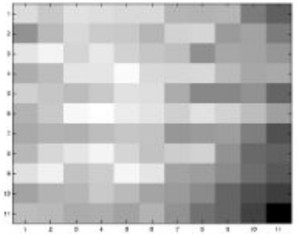
Harris corner response function

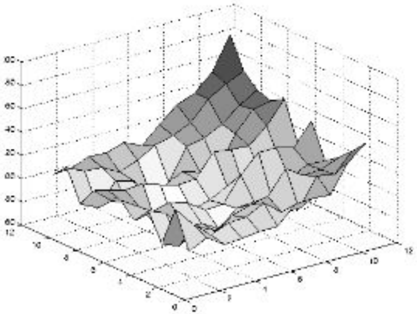
$$R = \det(M) - \alpha\,\mathrm{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

- R is large for a corner
- R is negative with large magnitude for an edge
- |R| is small for a flat region



Low texture region



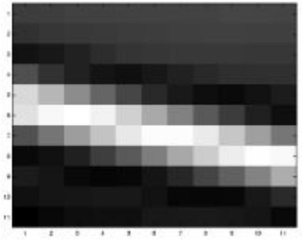


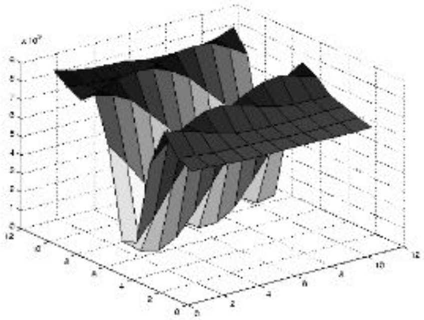
$$\sum \nabla I (\nabla I)^T$$

– gradients have small magnitude

– small λ_1, small λ_2

Edge



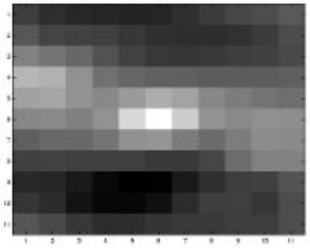


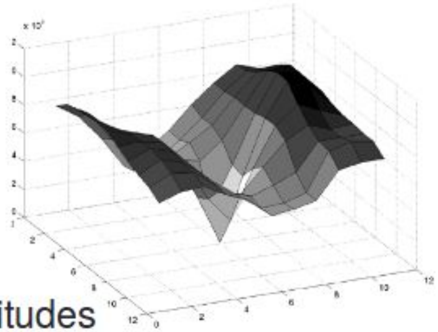
$$\sum \nabla I (\nabla I)^T$$

– large gradients, all the same

– large λ_1, small λ_2

High textured region





$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

– large λ_1, large λ_2

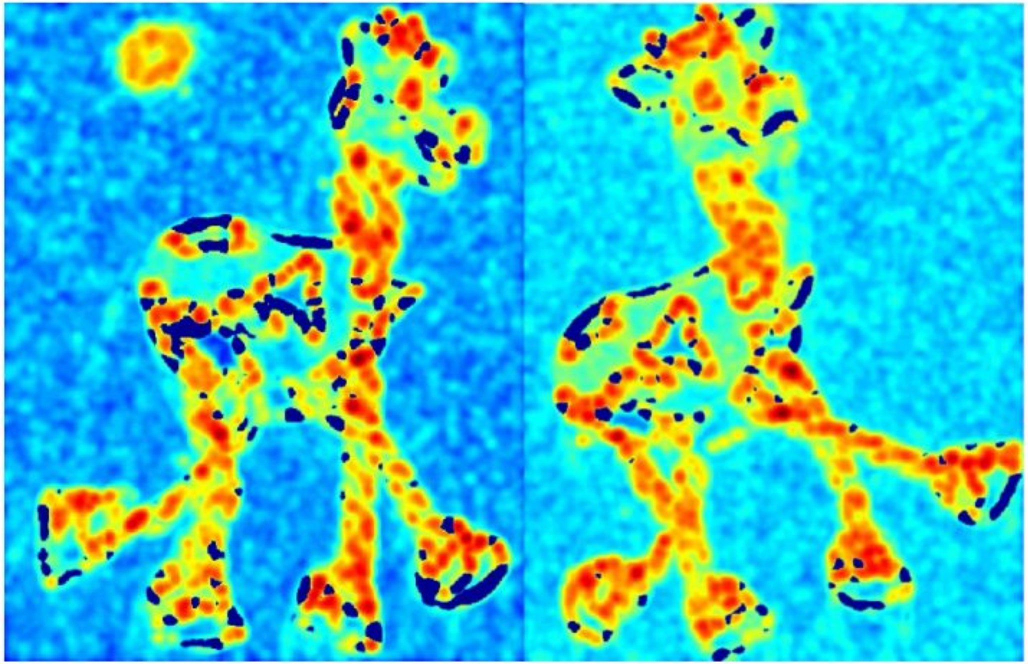
Harris detector: Algorithm

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

Harris Detector: Workflow

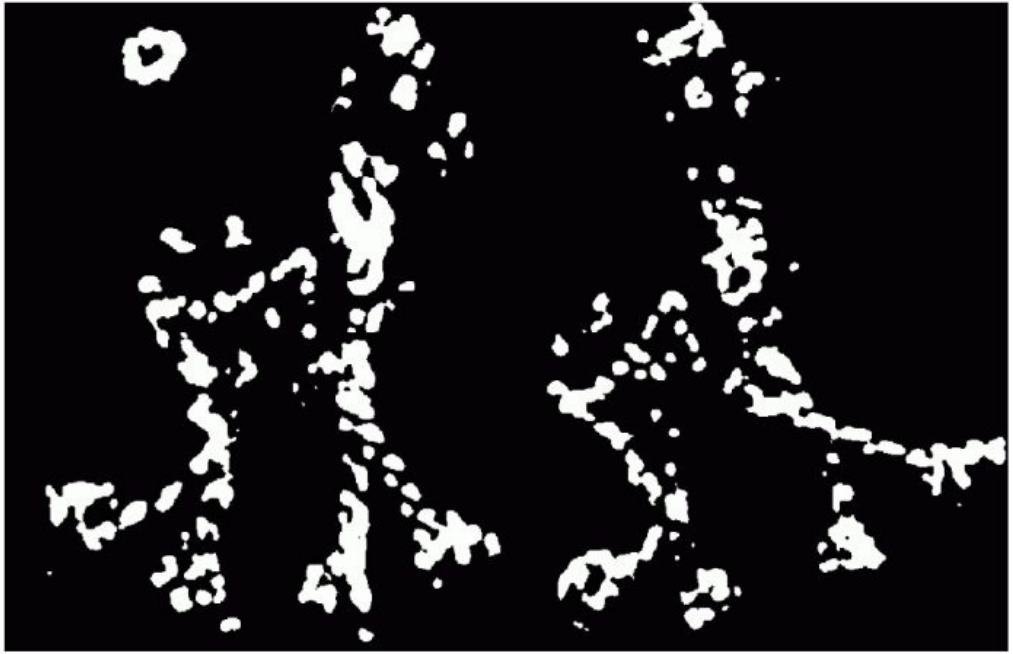
Harris Detector: Workflow

Compute corner response R



Harris Detector: Workflow

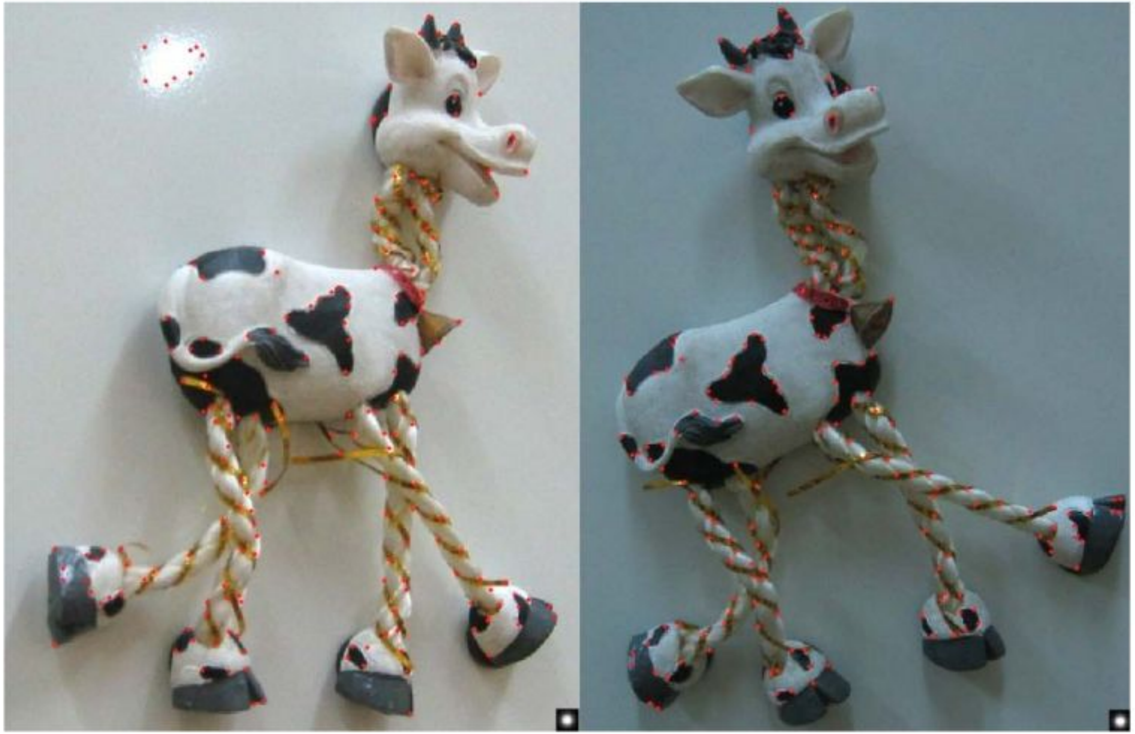
Find points with large corner response: R>threshold



Harris Detector: Workflow

Take only the points of local maxima of R

Harris Detector: Workflow



References

CS 4495 Computer Vision – A. Bobick

Thank You!