Lecture 7

Keypoints and Corners

Administrative

A1 due today!!!
- You can use up to 2 late days

A2 is out
- Due April 25th

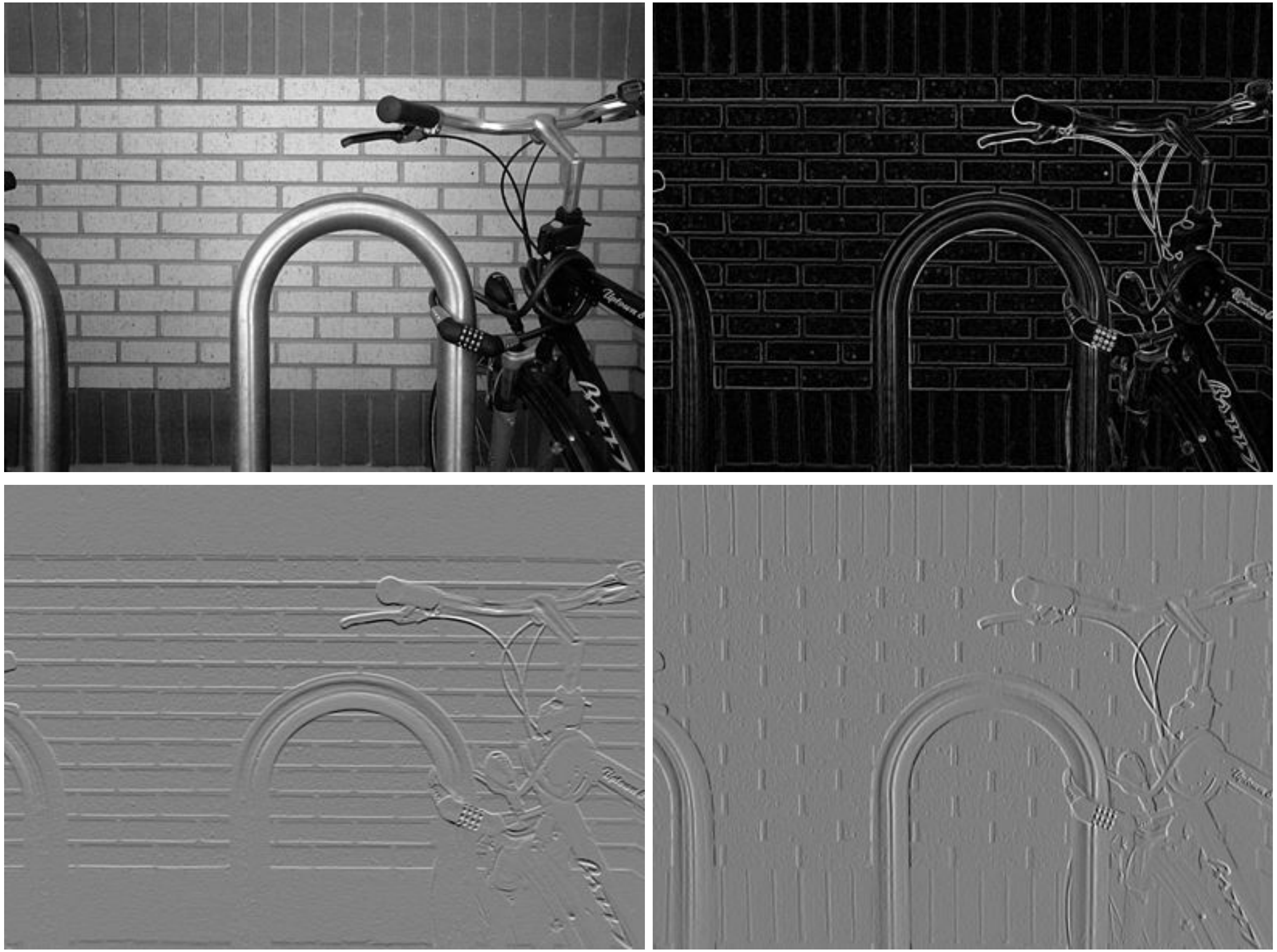
Administrative

- Recitation this Friday
- Fatemah
- Geometric transformations

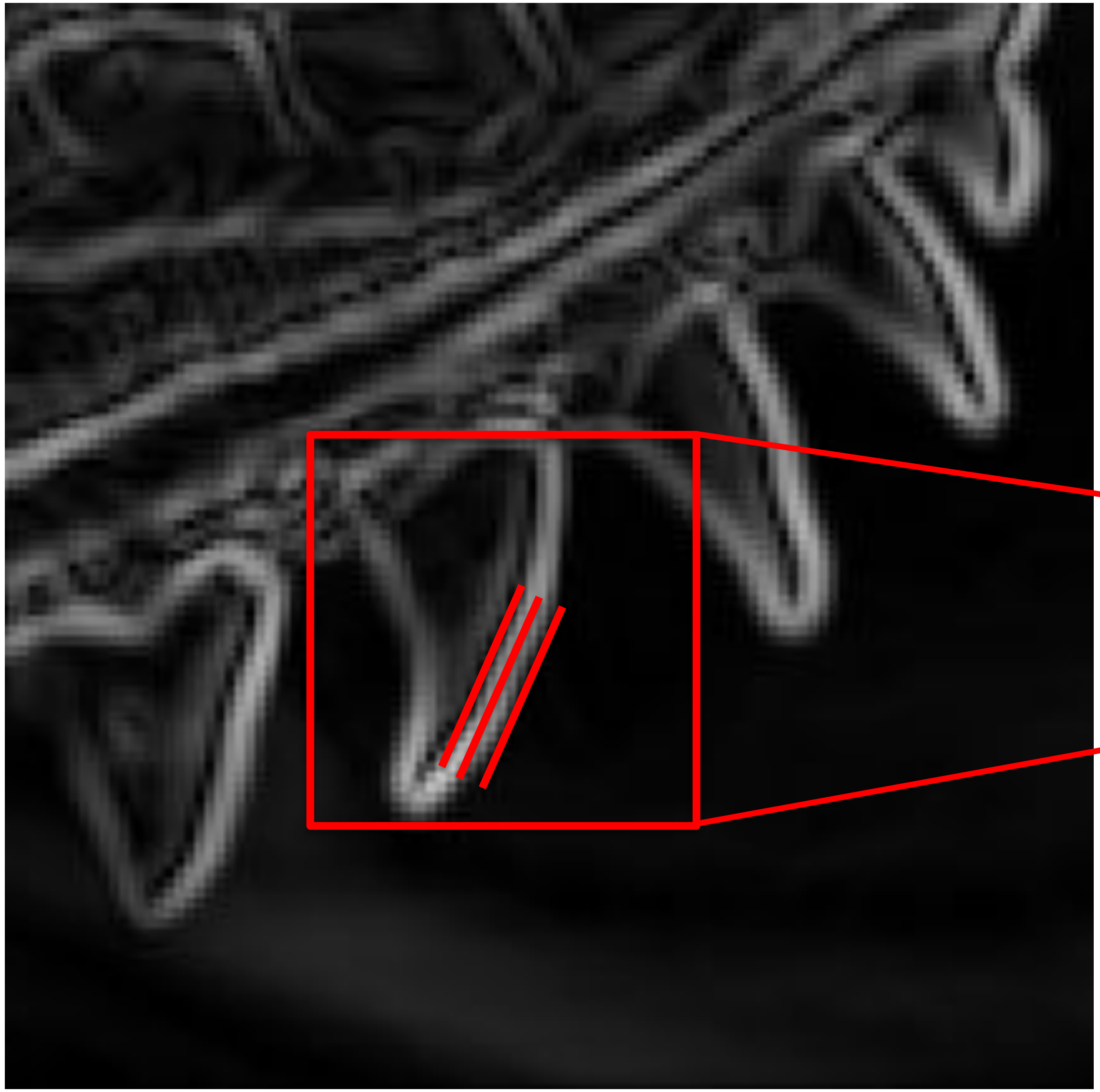
So far: Sobel Filter

Step 1: Calculate the gradient magnitude at every pixel location.

Step 2: Threshold the values to generate a binary image

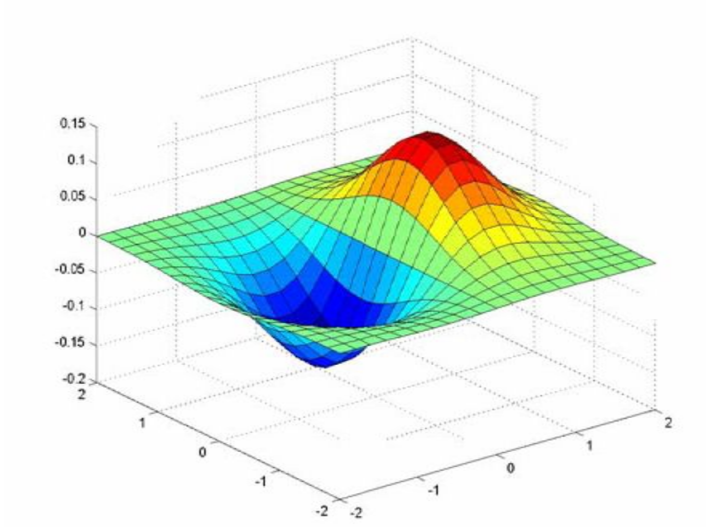


So far: challenges multiple disconnected edges

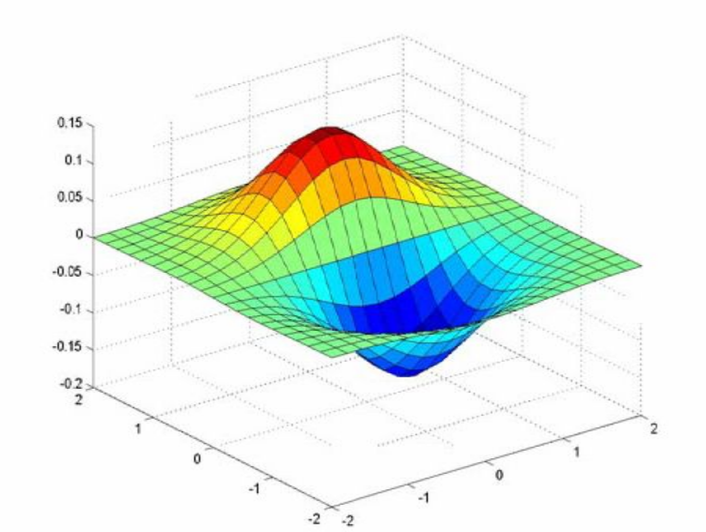


Gradient Magnitude

So far: Canny edge detector
Use Sobel filters to find line estimates

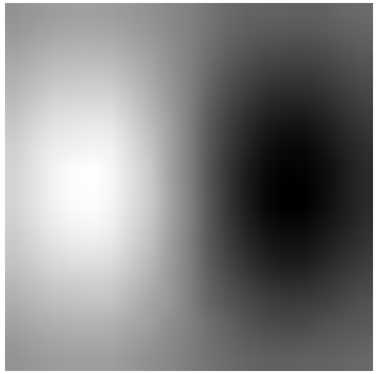


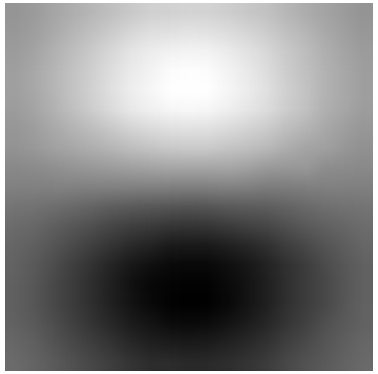
x-direction



y-direction

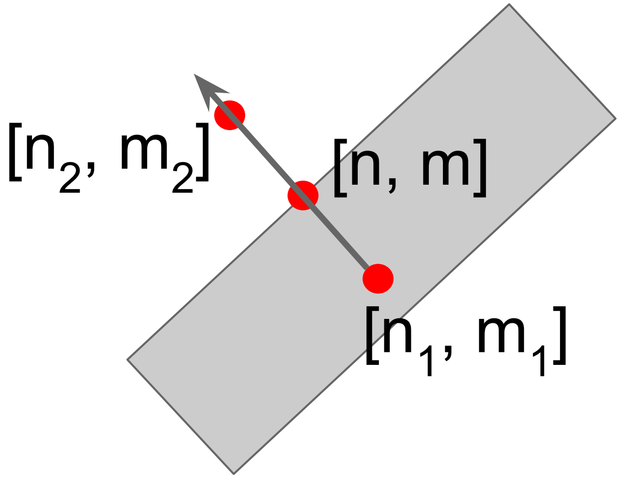
$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$





$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

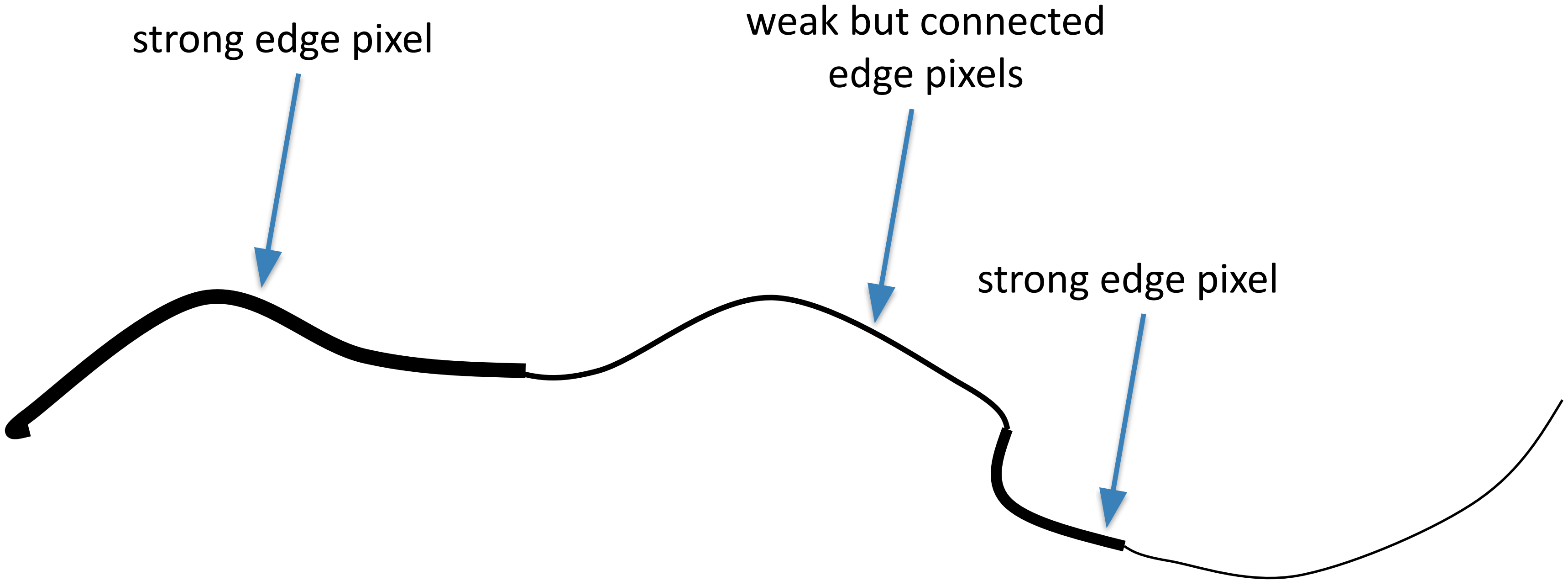
So far: Non-maximum suppression



$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

$$\text{If } G[n,m] = \begin{cases} G[n,m] & \text{if } G[n,m] > G[n_1,m_1] \text{ and } G[n,m] > G[n_2,m_2] \\ 0 & \text{otherwise} \end{cases}$$

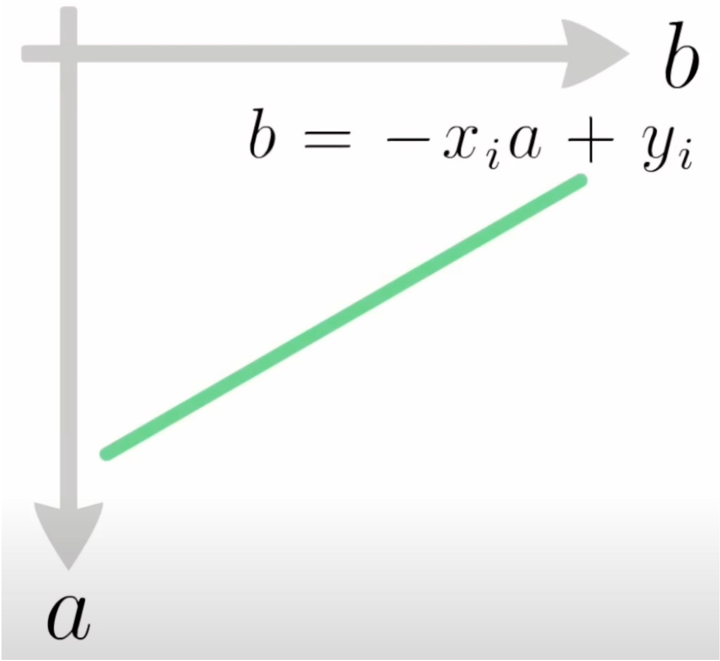
So far: Hysteresis thresholding
Strong and weak edges

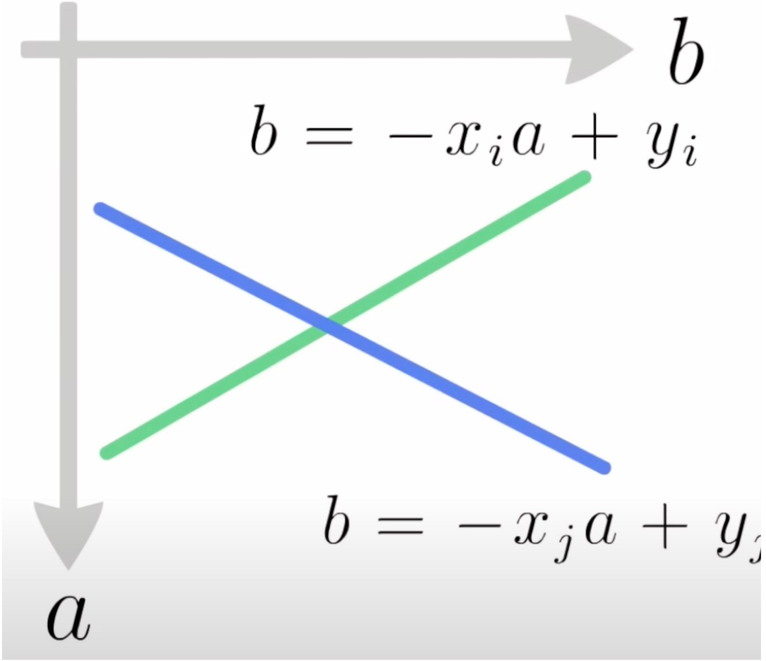


Source: S. Seitz

So far: The Hough transform

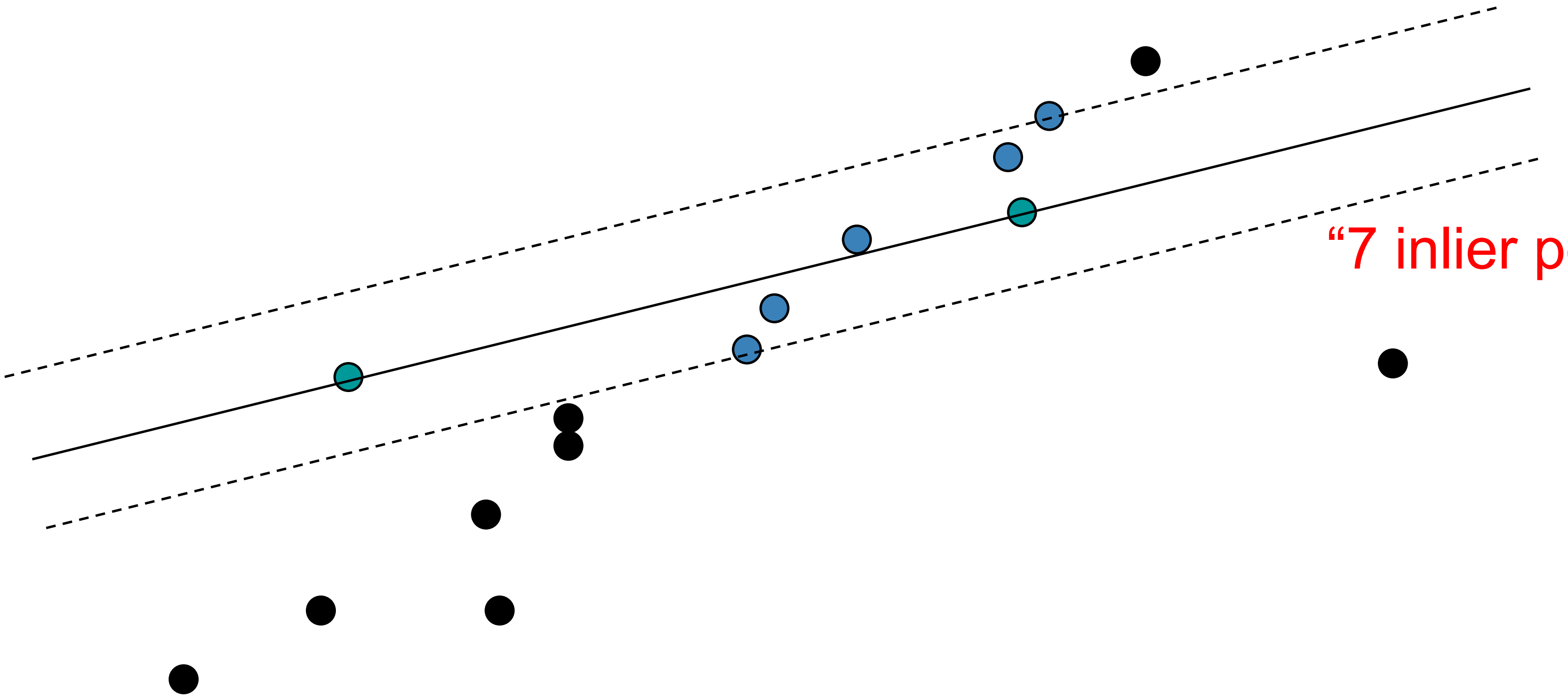
- So: one point (x_i,y_i) gives a line in (a,b) space.
- Another point (x_j,y_j) will give rise to another line in (a,b)-space.
- Iterate over pairs of points, to vote for buckets of intersection in (a,b)-space





So far: RANSAC

- Sample seed points, calculate line, count # of inliers, repeat



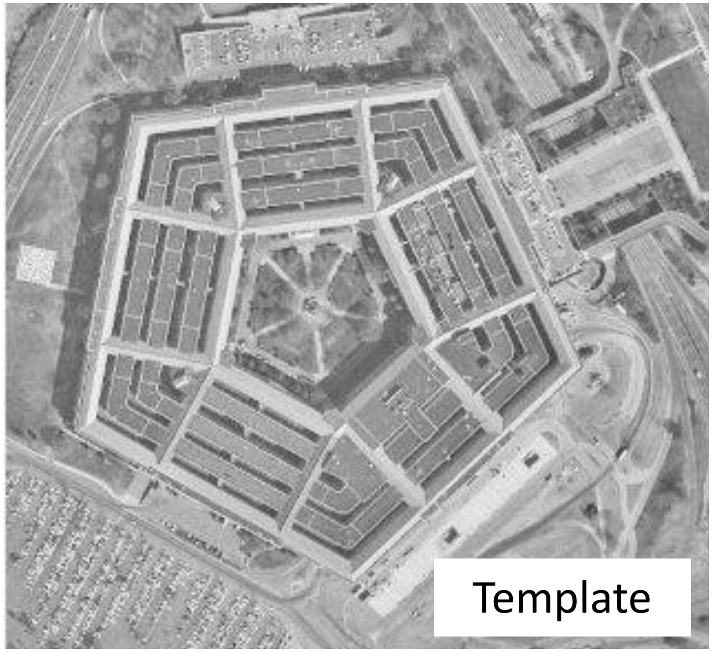
Today's agenda

- Local Invariant Features
- Harris Corner Detector

Today's agenda

- Local Invariant Features
- Harris Corner Detector

Image matching: a challenging problem



Q1. Will cross-correlation work?

Q2. Can we use match the lines?

Q. How would you build a system that can detect this movie in the pile?

Challenge: Perspective / viewpoint changes

by Diva Sian

by swashford

Slide credit: Steve Seitz

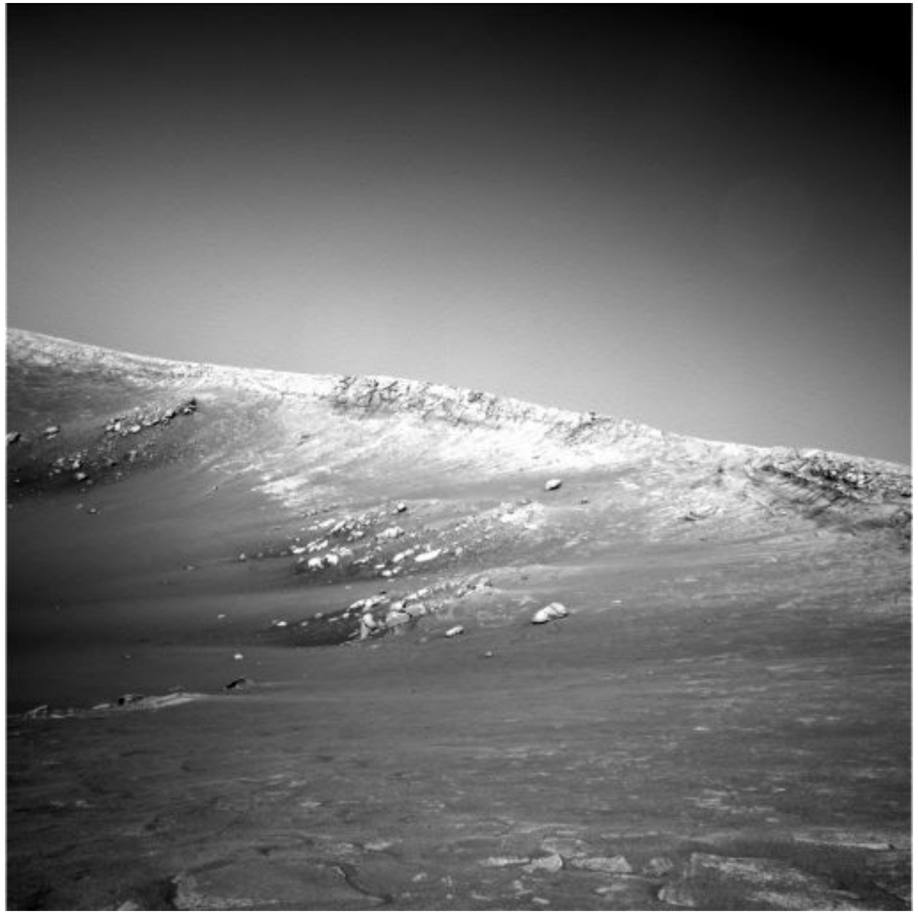
Challenge: partial observability

by Diva Sian

by scgbt

Slide credit: Steve Seitz

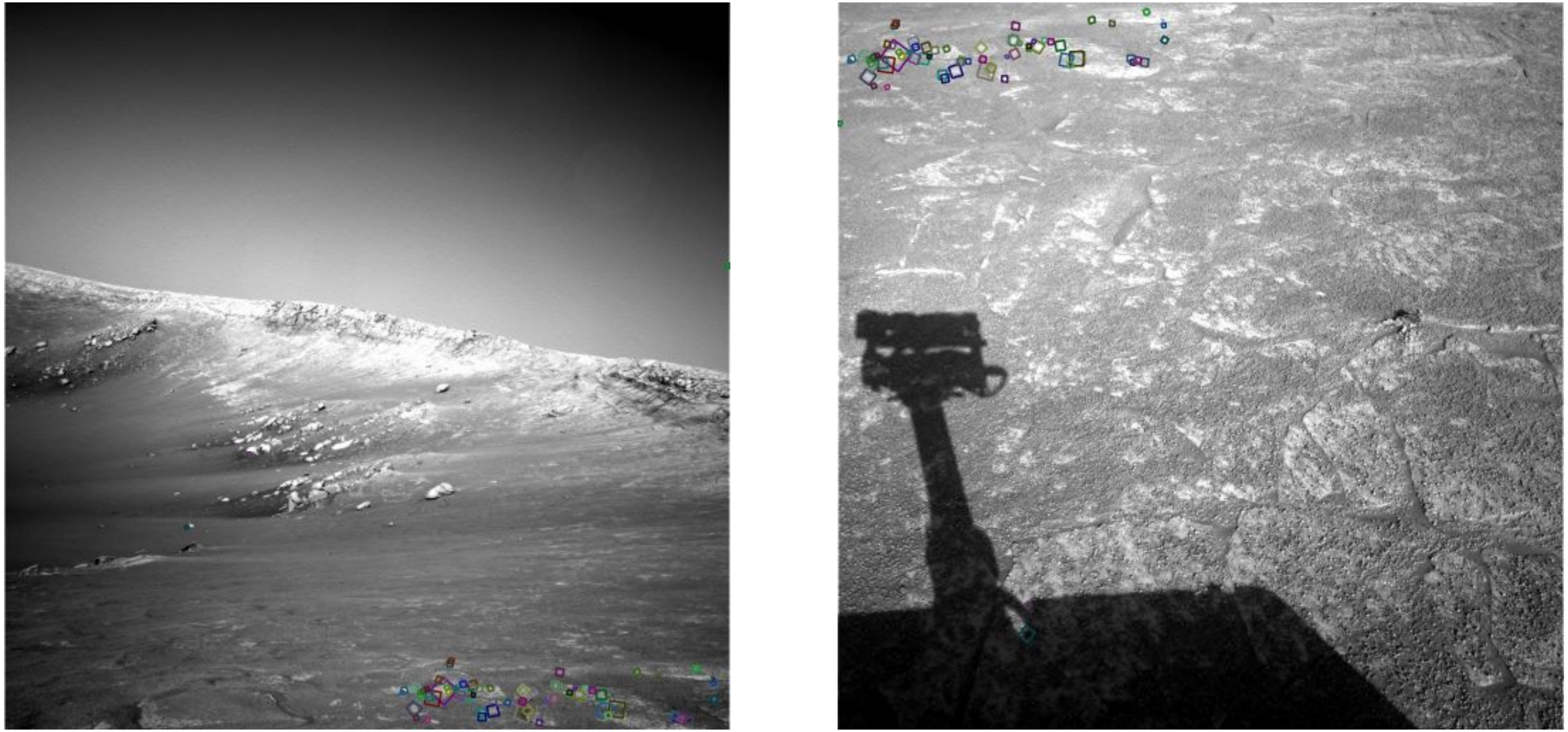
Challenge even for us



NASA Mars Rover images

Slide credit: Steve Seitz

Answer Below (Look for tiny colored squares)

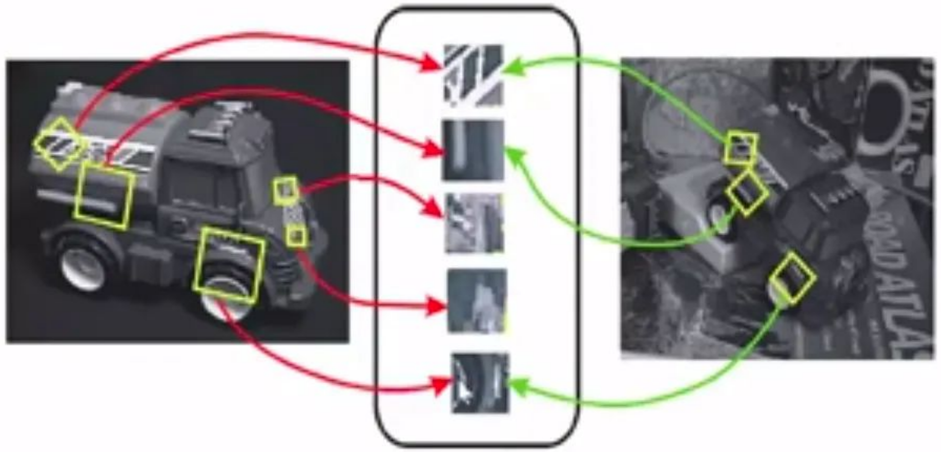


NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)

Slide credit: Steve Seitz

Intuition behind how to match images

- Find matching patches
- Check to make sure enough patches



Intuition behind how to match images

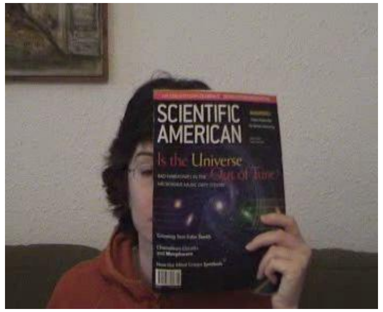
- Find matching patches
- Check to make sure enough patches

What do we need?

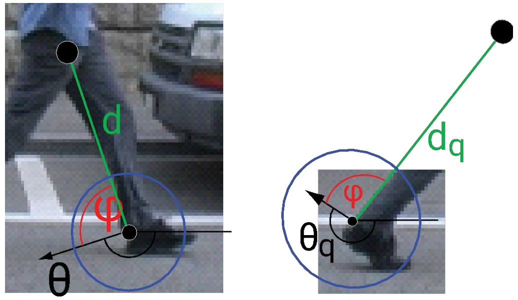
- We need to identify patches
- We need to learn to a way to describe each patch
- We need an algorithm to match the description between two patches

Motivation for using local features

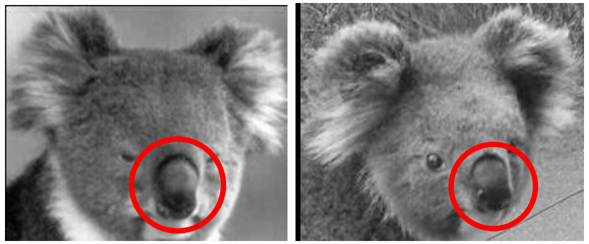
- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions



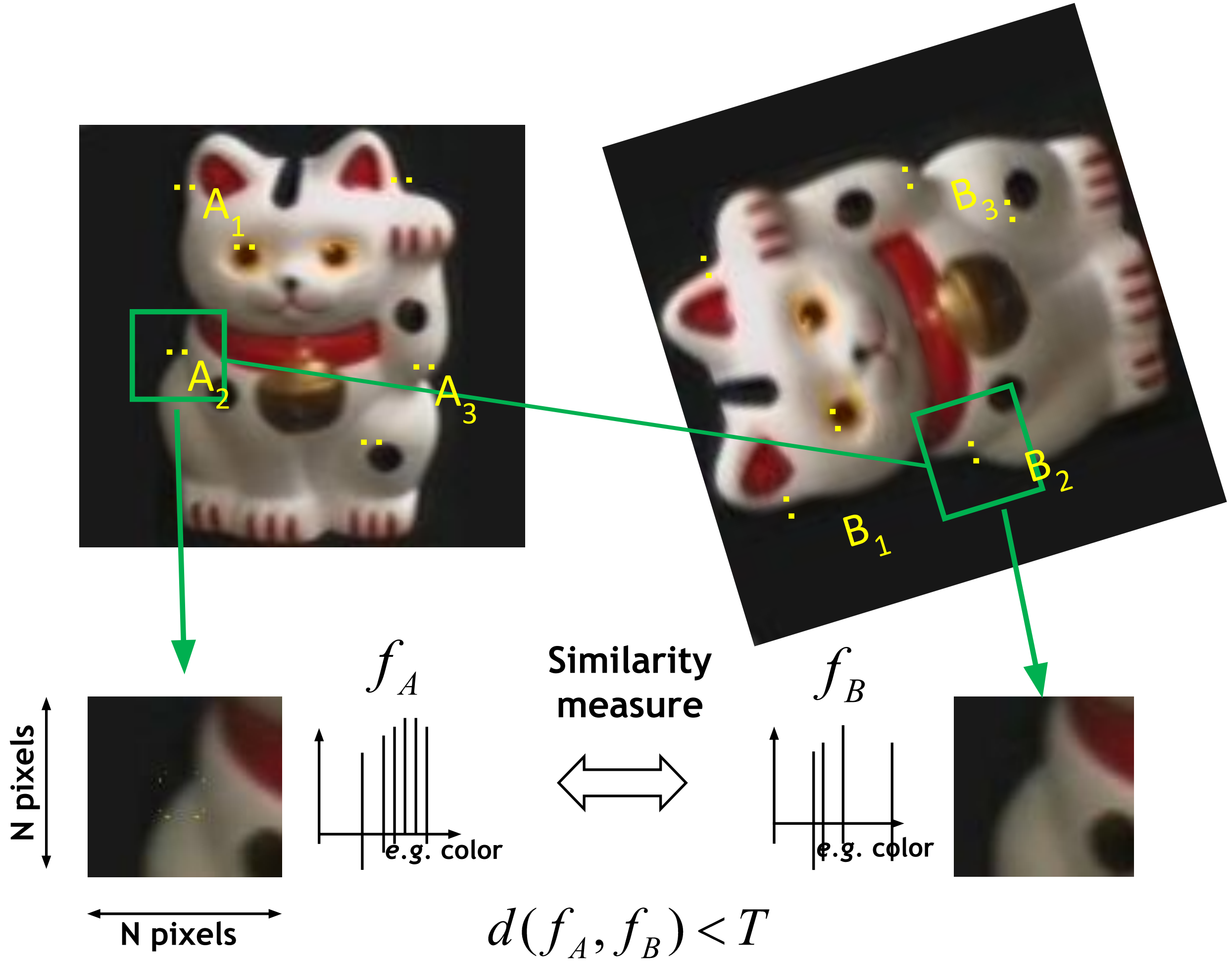
  - Articulation



  - Intra-category variations



General Approach



$$d(f_A, f_B) < T$$

1. Find a set of distinctive key-points

2. Define a region/patch around each keypoint

3. Describe and normalize the region content

4. Compute a local descriptor from the normalized region

5. Match local descriptors

Common Requirements

- Problem 1: How should we choose the key-points?
  - We want to detect the same points independently in both images

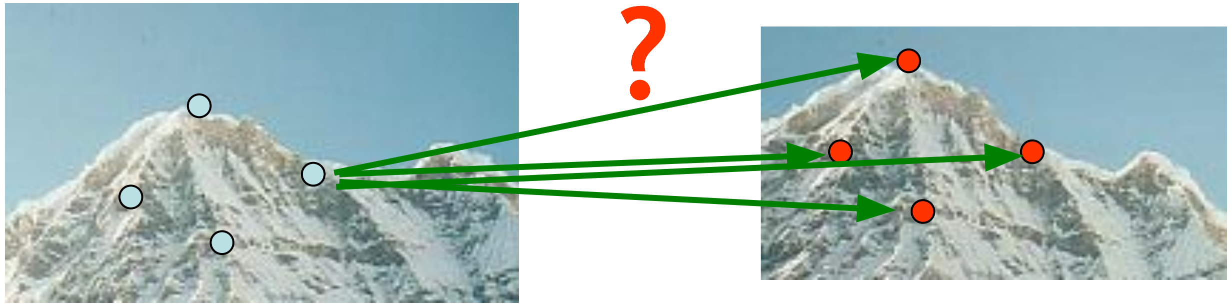
No chance to match if the key-points aren't the same

We need a repeatable detector!

Slide credit: Darya Frolova, Denis Simakov

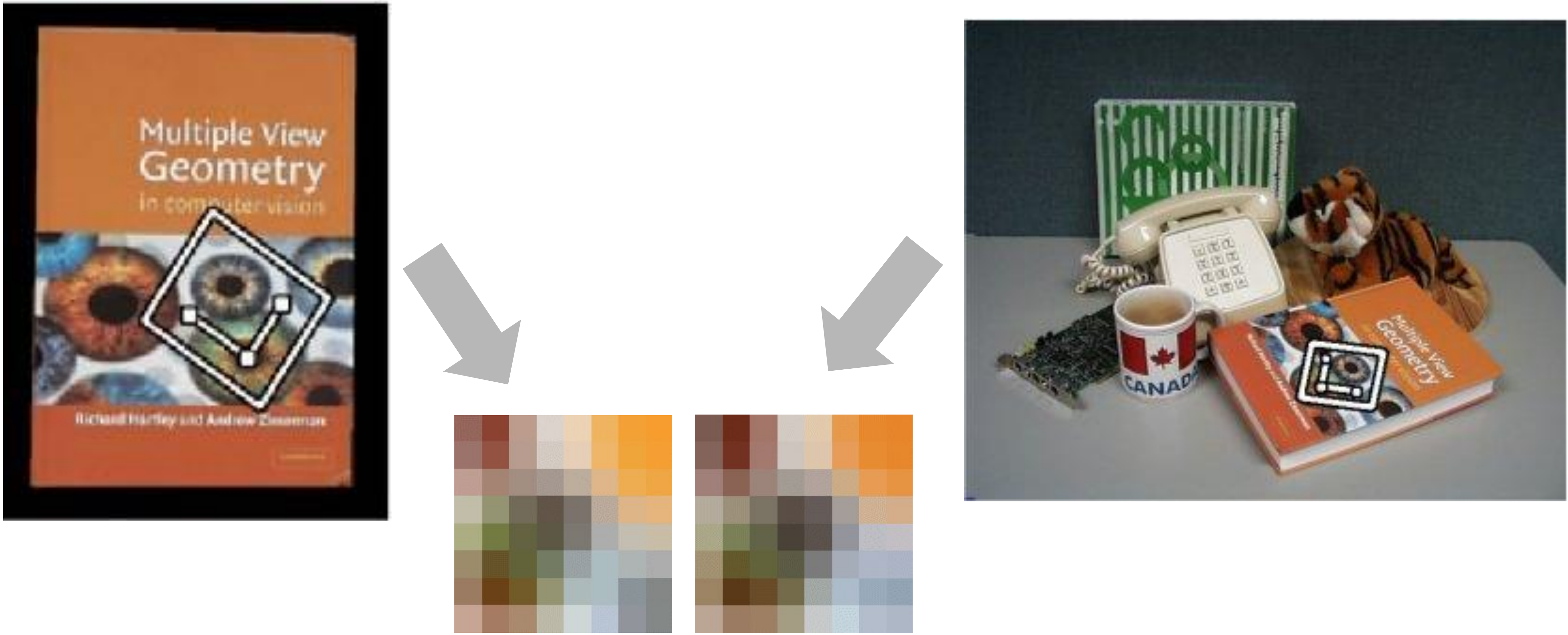
Common Requirements

- Problem 1: How should we choose the key-points?
  - Detect the same point independently in both images
- Problem 2: How should we describe each patch?
  - For each point correctly recognize the corresponding one

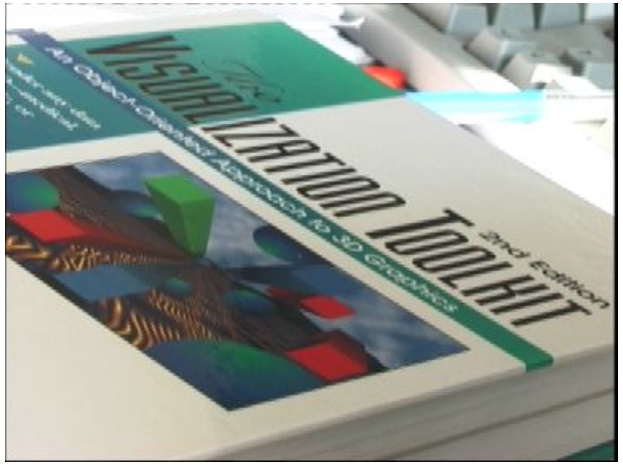


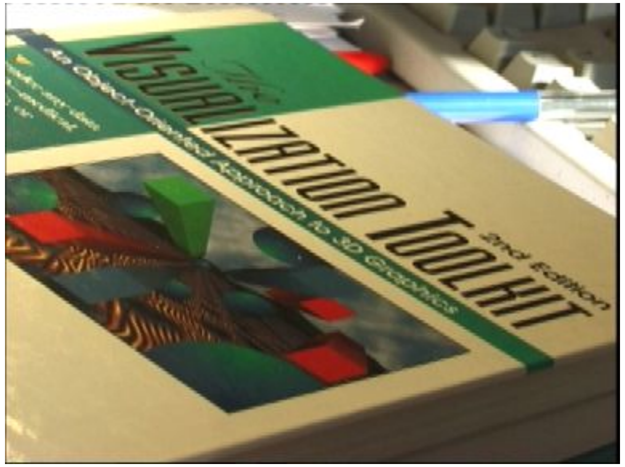
We need a reliable and distinctive descriptor!

Descriptions should be invariant to rotation and translation



Descriptions should be invariant to photometric transformations

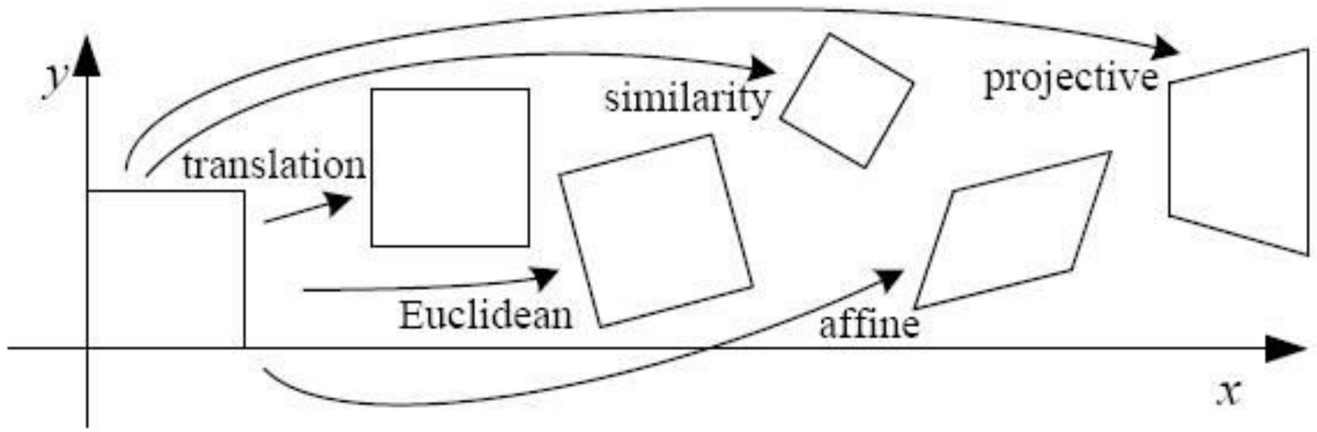


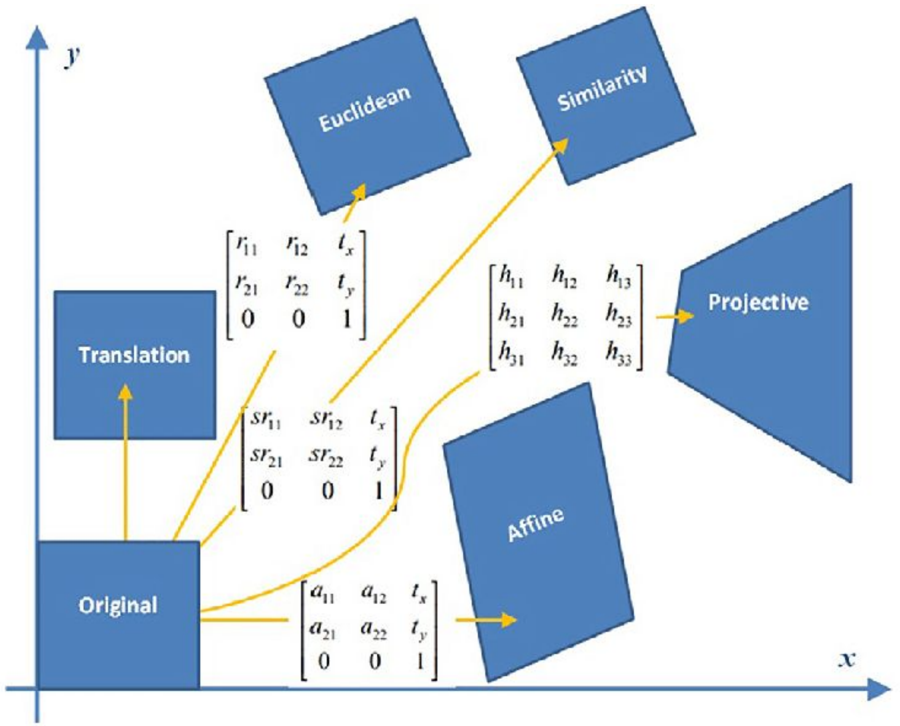


- Often modeled as a linear transformation:
  - Scaling + Offset

Slide credit: Tinne Tuytelaars

Levels of geometric transformations





Slide credit: Bastian Leibe

Requirements for Local Features

- Patch selection needs to be **repeatable** and **accurate**
  - **Invariant** to translation, rotation, scale changes
  - **Robust** to out-of-plane (≈affine) transformations
  - **Robust** to lighting variations, noise, blur, quantization
- **Locality**: Features are local, therefore robust to occlusion and clutter.
- **Quantity**: We need a sufficient number of regions to cover the object.
- **Distinctiveness**: The regions should contain "unique" structure.
- **Efficiency**: Close to real-time performance.

Slide credit: Bastian Leibe

Many existing feature detectors available

- Hessian & Harris [Beaudet '78], [Harris '88]
- Laplacian, DoG [Lindeberg '98], [Lowe '99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid '01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid '04]
- EBR and IBR [Tuytelaars & Van Gool '04]
- MSER [Matas '02]
- Salient Regions [Kadir & Brady '01]
- Neural networks [Krichevsky '12]

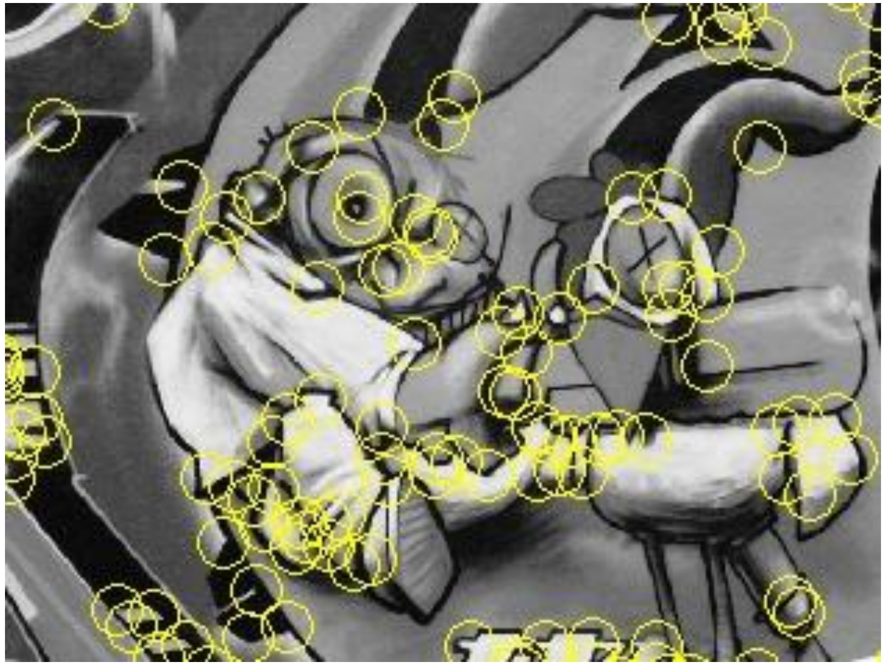
- *Those detectors have become a basic building block for many applications in Computer Vision.*

Slide credit: Bastian Leibe

Today's agenda

- Local Invariant Features
- Harris Corner Detector

Keypoint Localization



- **Goals:**
  - Repeatable detection
  - Precise localization
  - Interesting content

intuition ⇒ *Look for 2D signal changes* (LSI systems strike again)

Slide credit: Bastian Leibe

What are good patches?

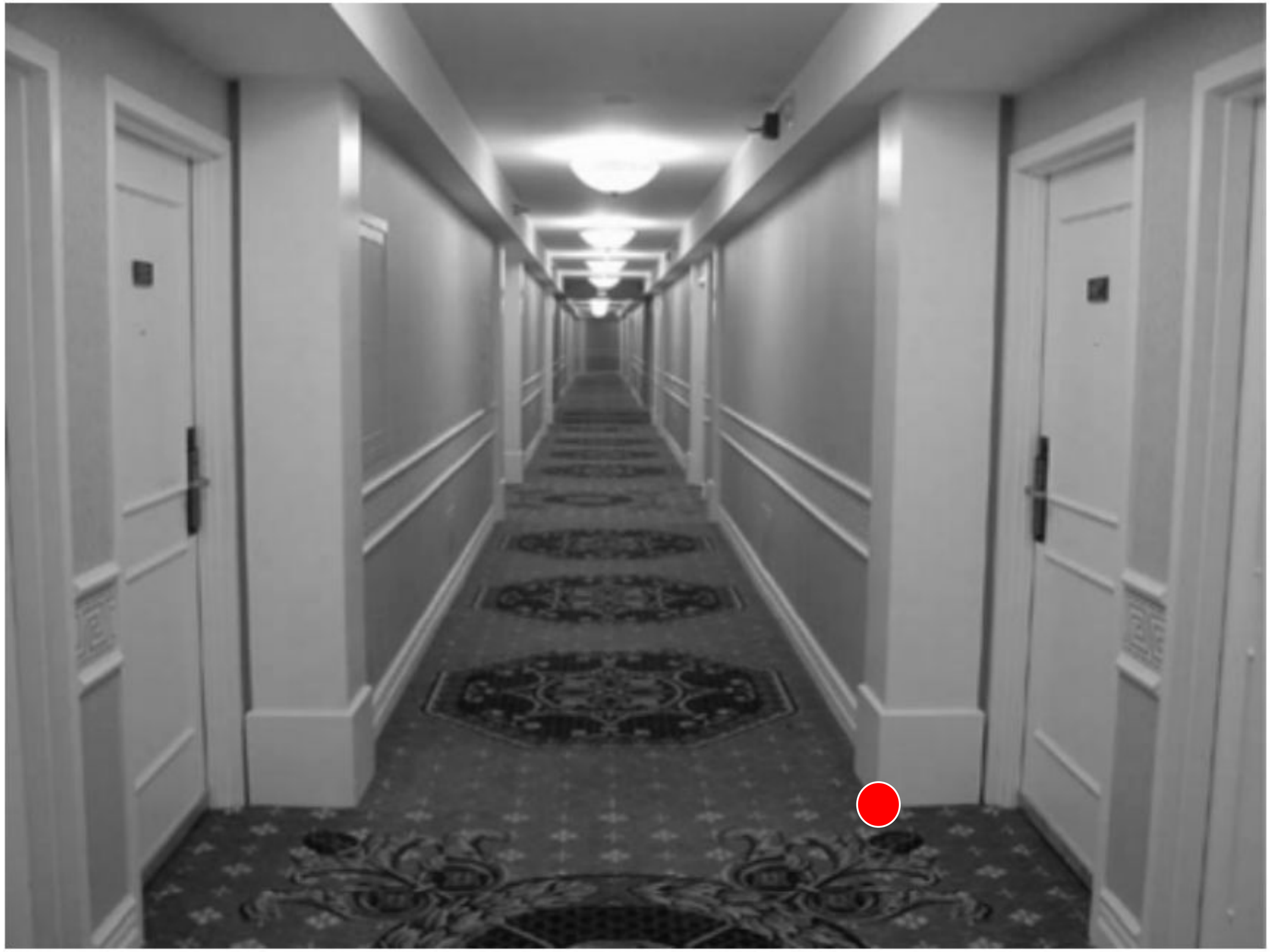
Q. Is this a good patch for image matching?

What are good patches?

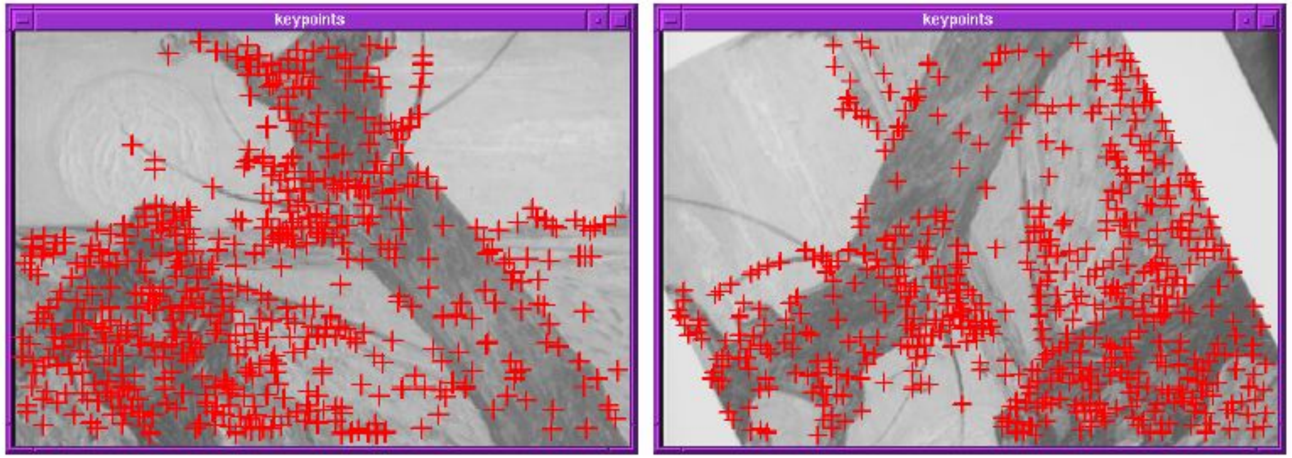
Q. What about this one?

What are good patches?

Q. Let's try another one?



Finding Corners



How do we find corners using LSI systems?

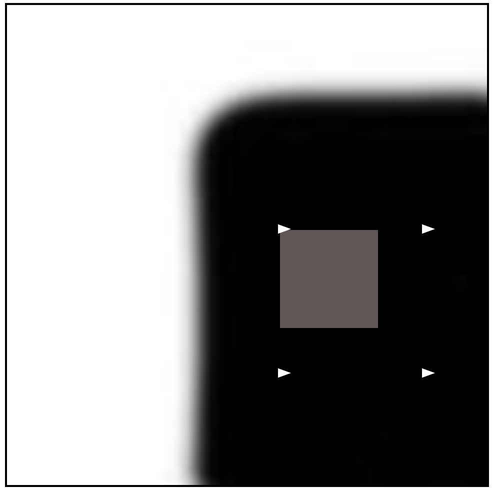
- The image gradient around a corner has two or more dominant directions

Corners are **repeatable** and **distinctive**

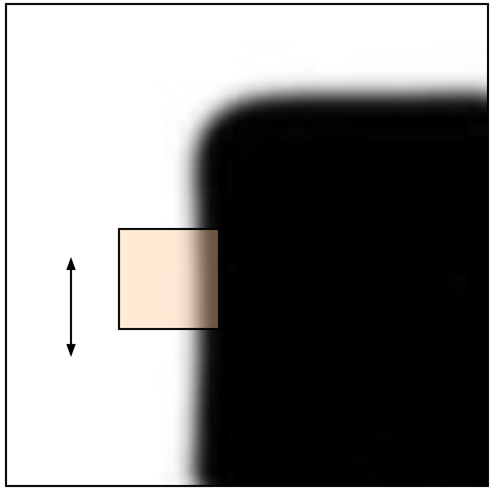
C.Harris and M.Stephens. "A Combined Corner and Edge Detector."
Proceedings of the 4th Alvey Vision Conference, 1988.

Corners are distinctive key-points

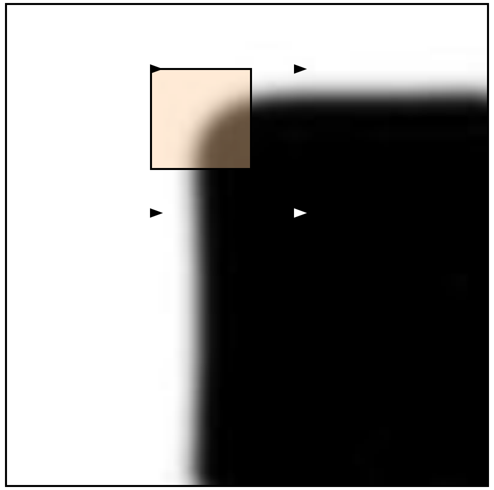
- We should easily recognize the corner point by looking through a small image patch (*locality*)
- Shifting the window in *any direction* should give *a large change* in intensity (*good localization*)



"flat" region: no change in all directions



"edge": no change along the edge direction



"corner": significant change in all directions

Slide credit: Alyosha Efros

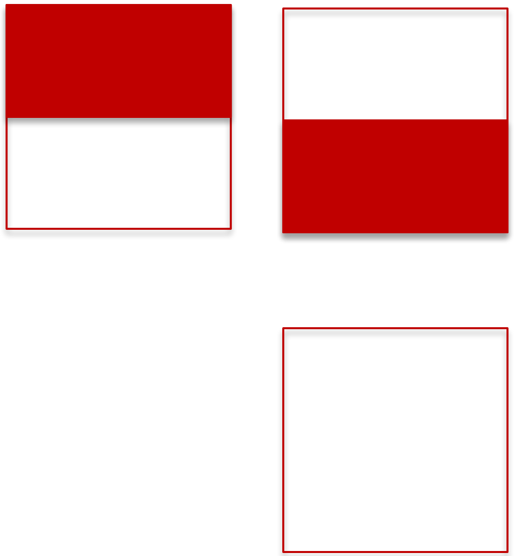
Flat patches have small image gradients

$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Small

Flat

Edges have high gradient in one direction



$\sum I_x^2 \rightarrow$ Small

$\sum I_y^2 \rightarrow$ Large

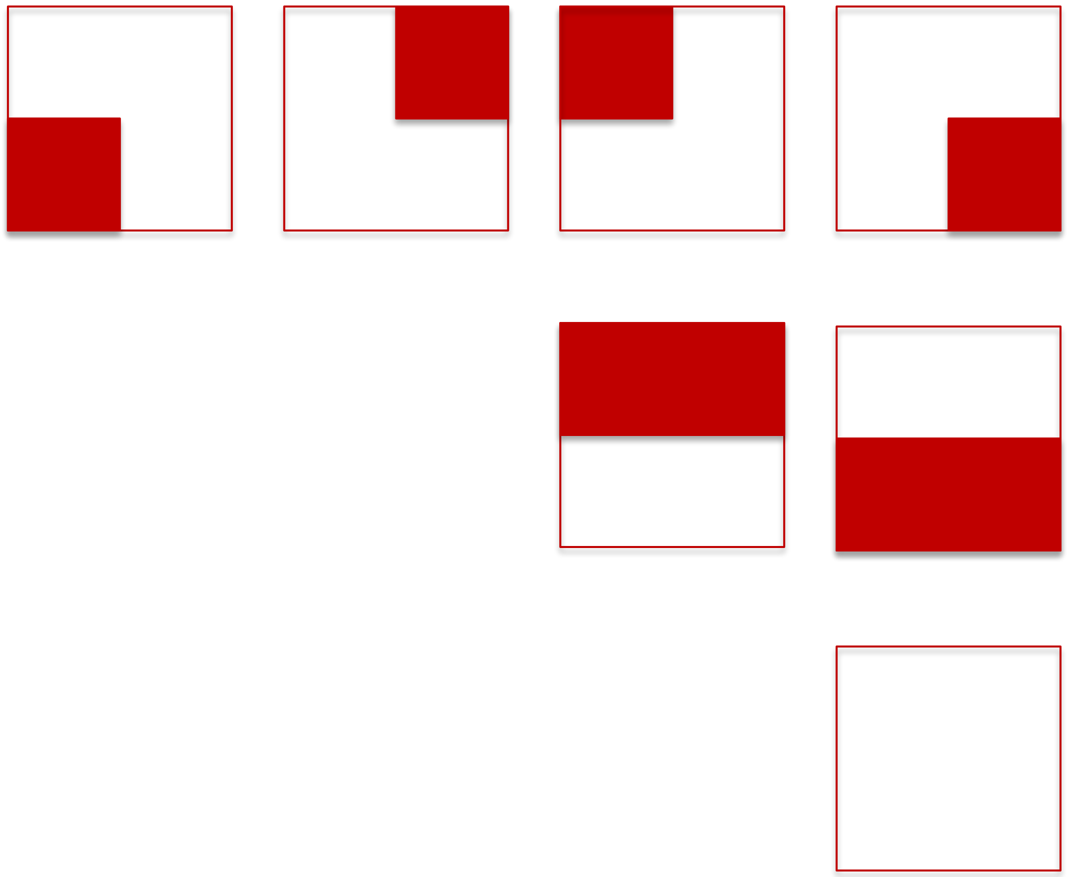
Edge

$\sum I_x^2 \rightarrow$ Small

$\sum I_y^2 \rightarrow$ Small

Flat

Corners versus edges



$$\sum I_x^2 \rightarrow \text{Large}$$
$$\sum I_y^2 \rightarrow \text{Large}$$

Corner

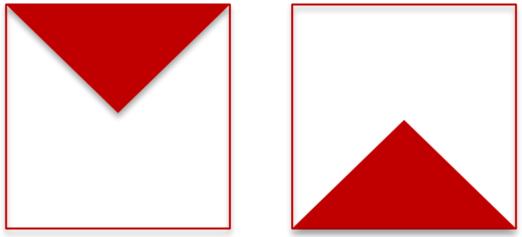
$$\sum I_x^2 \rightarrow \text{Small}$$
$$\sum I_y^2 \rightarrow \text{Large}$$

Edge

$$\sum I_x^2 \rightarrow \text{Small}$$
$$\sum I_y^2 \rightarrow \text{Small}$$

Flat

Generalizing to corners in any direction



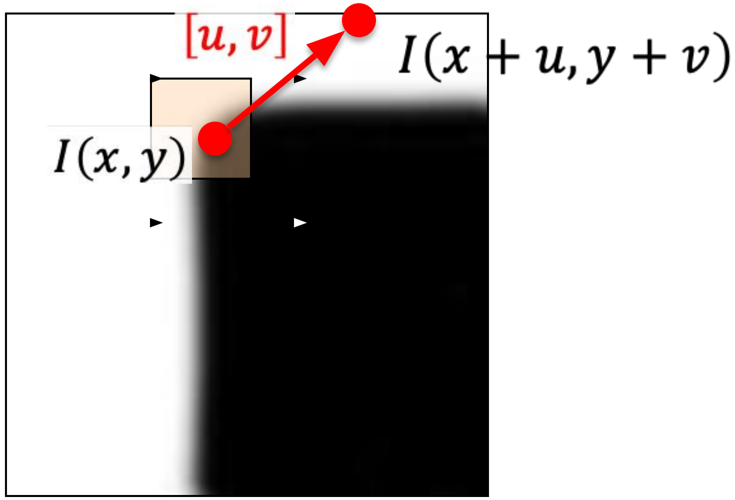
$\sum I_x^2 \longrightarrow$??

$\sum I_y^2 \longrightarrow$??

Corner

Harris Detector Formulation

- Find patches that result in large change of pixel values when shifted in *any direction.*
- When we shift by $[u, v]$, the intensity change at the center pixel is:



"corner":
significant change in all directions

- Measure change as intensity difference:

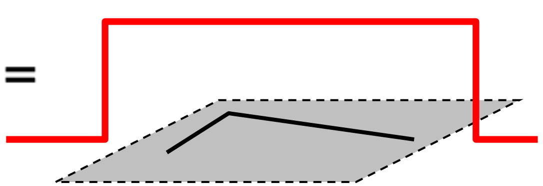
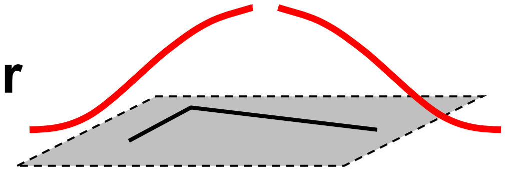
$$(I(x + u, y + v) - I(x, y))$$

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

Harris Detector Formulation

- When we shift by $[u, v]$, the change in intensity for the "small window" is:

$$E(u,v) = \underbrace{\sum_{x,y}}_{\text{Sum over window}} \underbrace{w(x,y)}_{\text{Window function}} \underbrace{[\underbrace{I(x+u, y+v)}_{\text{Shifted intensity}} - \underbrace{I(x,y)}_{\text{Intensity}}]^2}_{\text{Intensity change}}$$

Window function $w(x,y) =$  or 

1 in window, 0 outside

Gaussian

Slide credit: Rick Szeliski

Change in intensity function

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

We can rewrite the shifted intensity using Taylor's expansion:

$$I(x+u, y+v) \approx I(x,y) + I_x u + I_y v$$

Substituting it back into E(u, v):

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

Re-writing E:

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

Re-writing E:

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

$$= \sum_{x,y} w(x,y) \begin{bmatrix} I_x u & I_y v \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

Re-writing E:

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

$$= \sum_{x,y} w(x,y) \begin{bmatrix} I_x u & I_y v \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

$$= \sum_{x,y} w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

Re-writing E:

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

$$= \sum_{x,y} w(x,y) \begin{bmatrix} I_x u & I_y v \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

$$= \sum_{x,y} w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

$$= \sum_{x,y} w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Re-writing E:

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

$$= \sum_{x,y} w(x,y) \begin{bmatrix} I_x u & I_y v \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

$$= \sum_{x,y} w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x u \\ I_y v \end{bmatrix}$$

$$= \sum_{x,y} w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

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

Re-writing E:

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

Re-writing E:

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

Re-writing E:

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

$$= w(x,y) \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

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

Re-writing E:

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

$$= w(x,y) \boxed{\begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}}$$

Does anyone know what this part of the equation is?

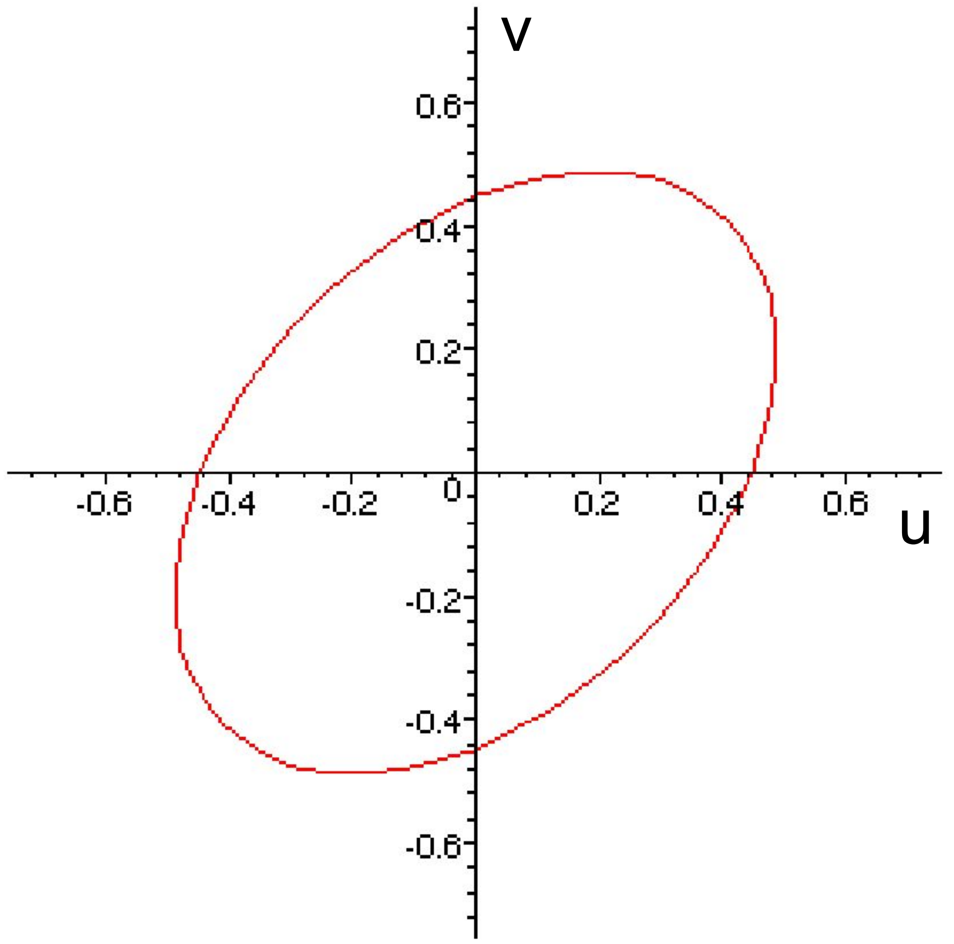
$$M = \begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_y^2 \end{bmatrix}$$

It's the equation of an ellipse

$$5u^2 - 4uv + 5v^2 = 1$$

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

$$M = \begin{bmatrix} 5 & -2 \\ -2 & 5 \end{bmatrix}$$



Change in intensity in a patch

- So, using Taylor's expansion, the change in intensity in an image patch:

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

where M is a 2×2 matrix computed from image derivatives:

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

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

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

Harris Detector Formulation

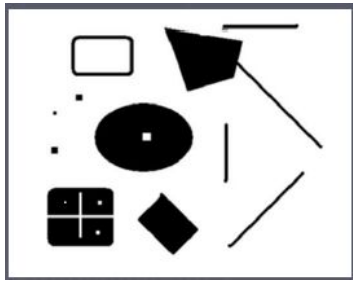
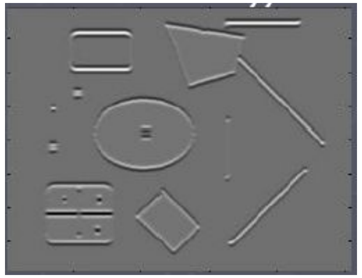
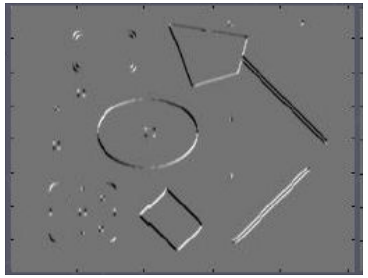


Image I

I_x



I_y



$I_x I_y$

where M is a 2×2 matrix computed from image derivatives:

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

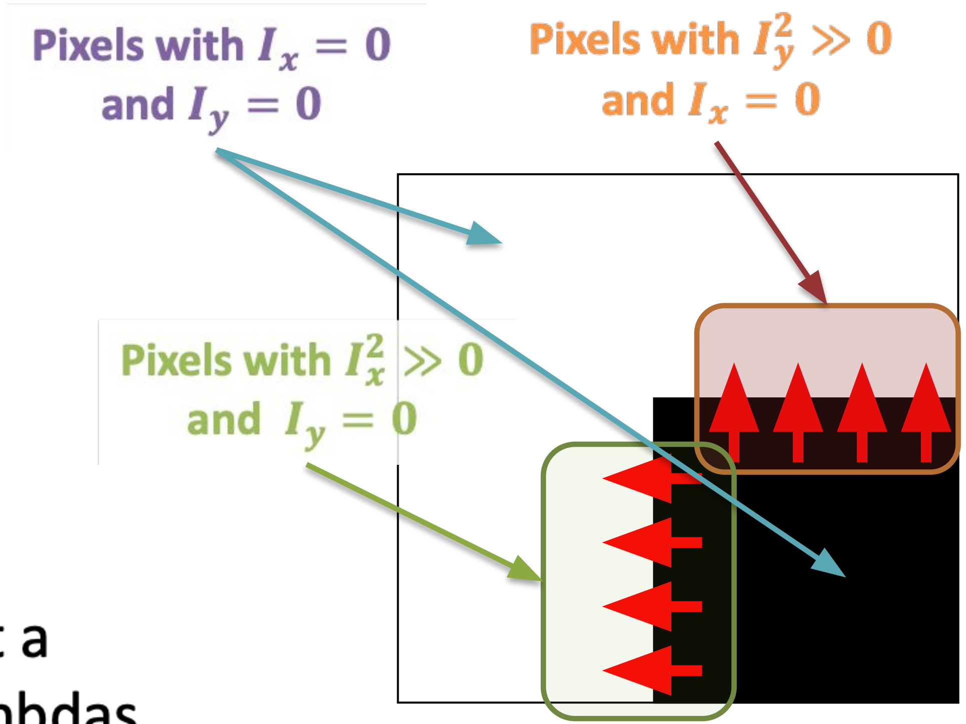
Gradient with respect to x, times gradient with respect to y

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

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

What Does This Matrix Reveal?

- First, let's consider an axis-aligned corner.
- In that case, the dominant gradient directions align with the x or the y axis
- $M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$
- This means: if either λ is close to 0, then this is not a corner, so look for image windows where both lambdas are large.
- What if we have a corner that is not aligned with the image axes?

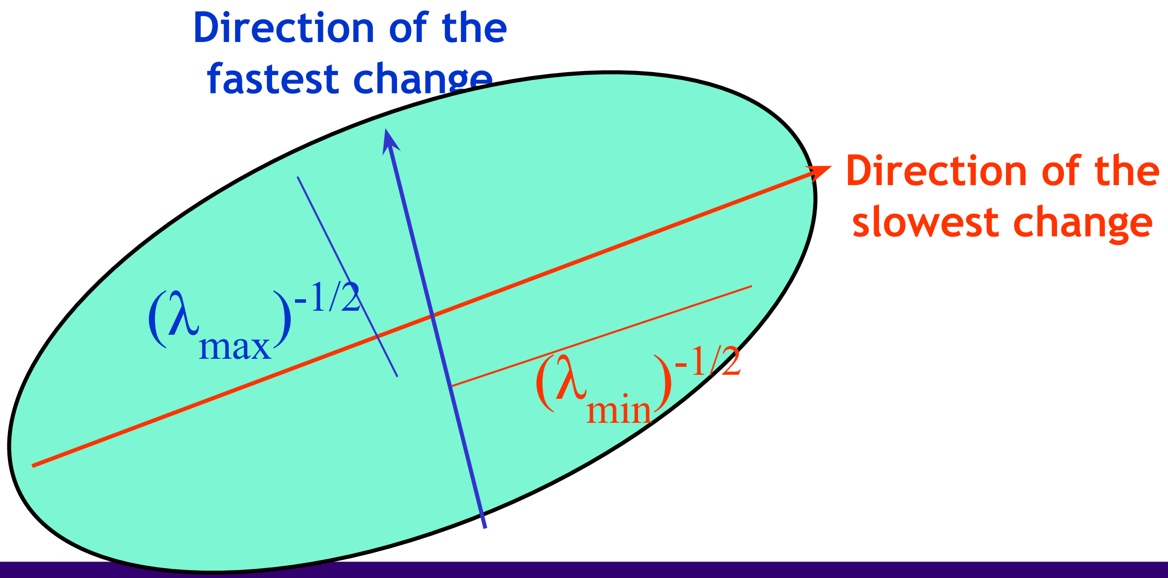


General Case

- Since $M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$ is symmetric, we can re-rewrite $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

(Eigenvalue decomposition)

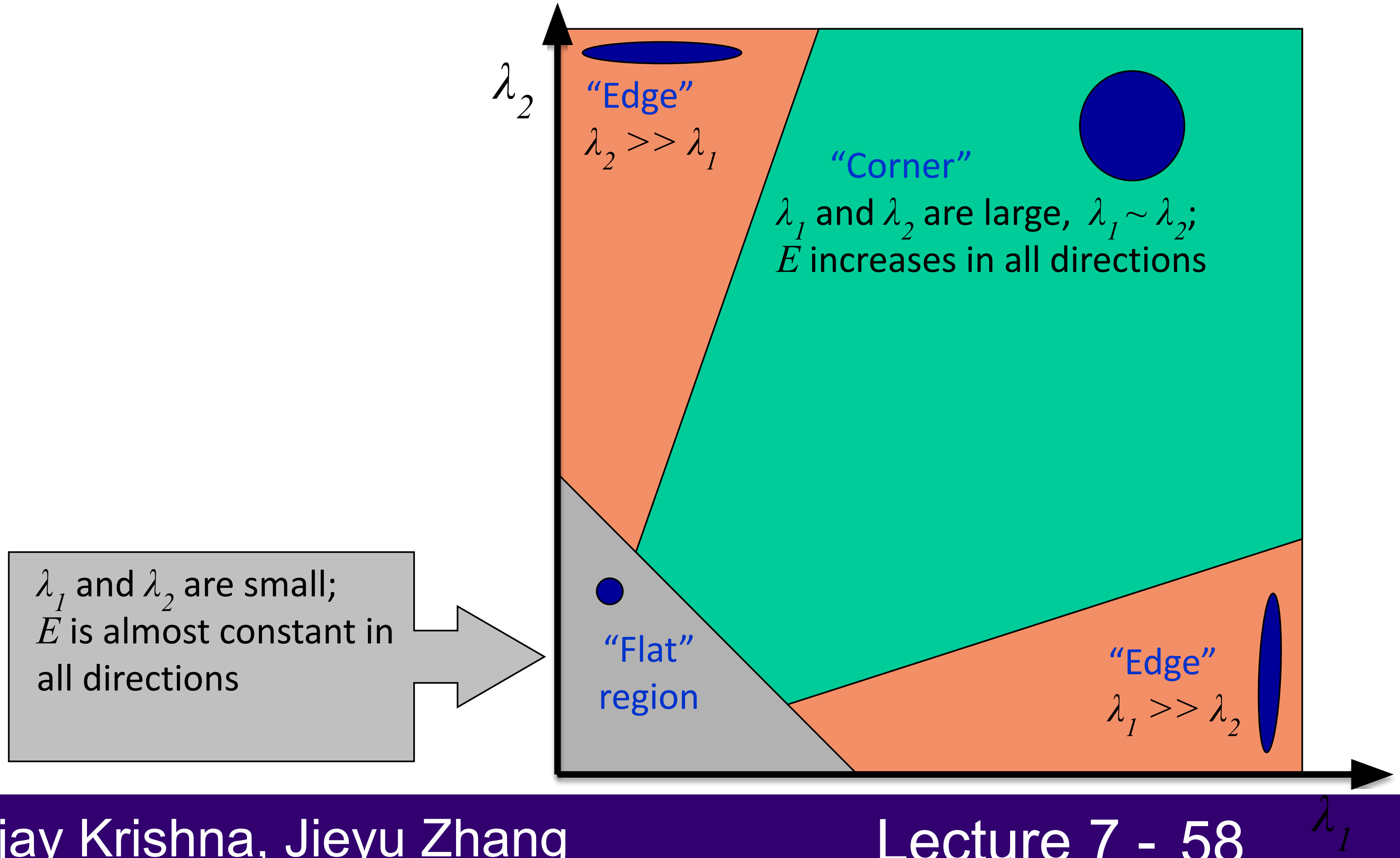
- We can think of M as an ellipse with its axis lengths determined by the eigenvalues λ_1 and λ_2; and its orientation determined by R



- A rotated corner would produce the same eigenvalues as its non-rotated version.

Interpreting the Eigenvalues

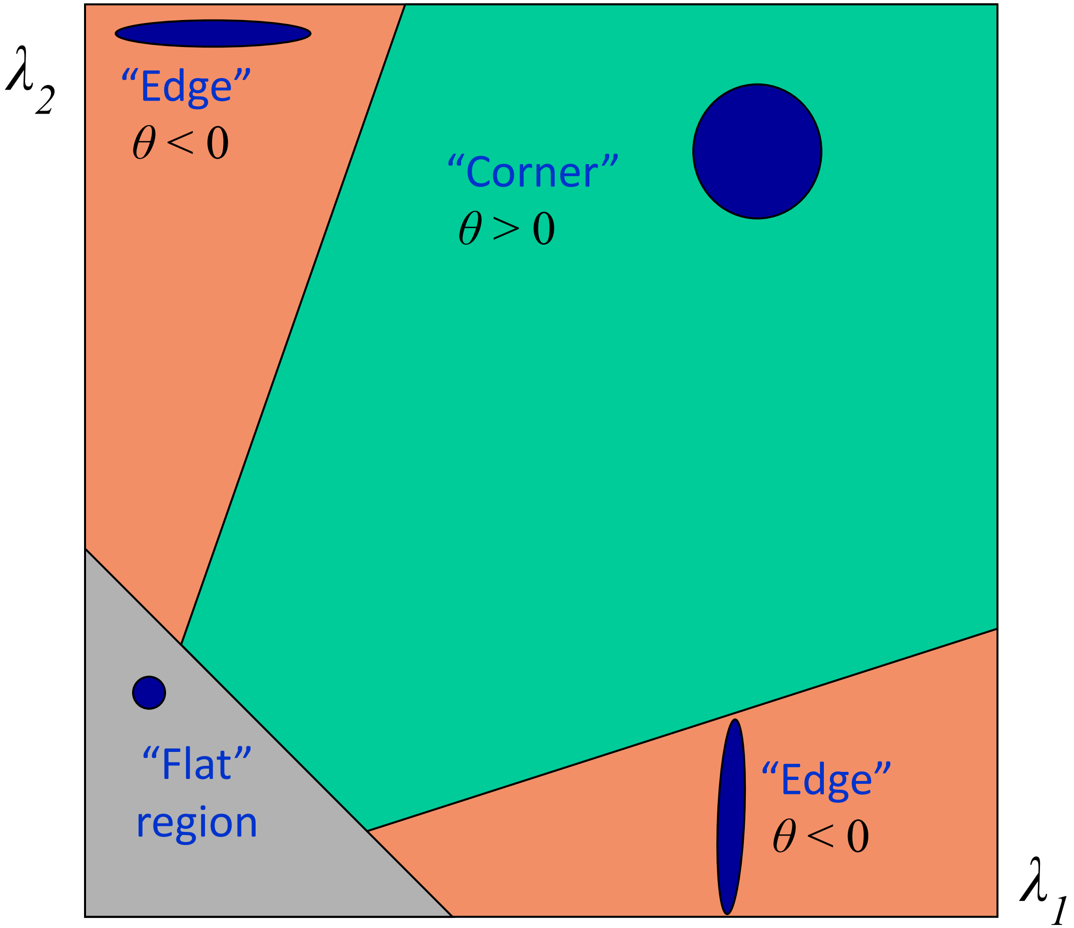
- Classification of image points using eigenvalues of M:



Slide credit: Kristen Grauman

Corner Response Function

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



- Fast approximation
  - Avoid computing the eigenvalues
  - α: constant (0.04 to 0.06)

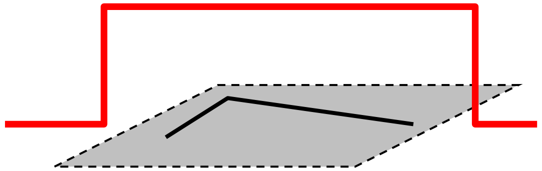
Slide credit: Kristen Grauman

Window Function $w(x,y)$

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

- Option 1: uniform window
  - Sum over square window

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

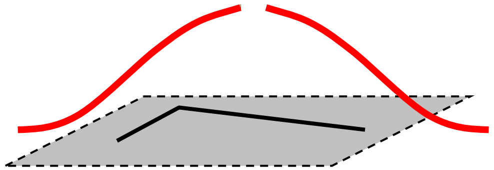


1 in window, 0 outside

  - Problem: not rotation invariant

- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum

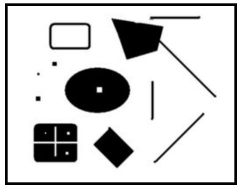
$$M = g(\sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



**Gaussi
an**

  - Result is rotation invariant

Summary: Harris Detector [Harris88]



- Compute second moment matrix (autocorrelation matrix)

$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

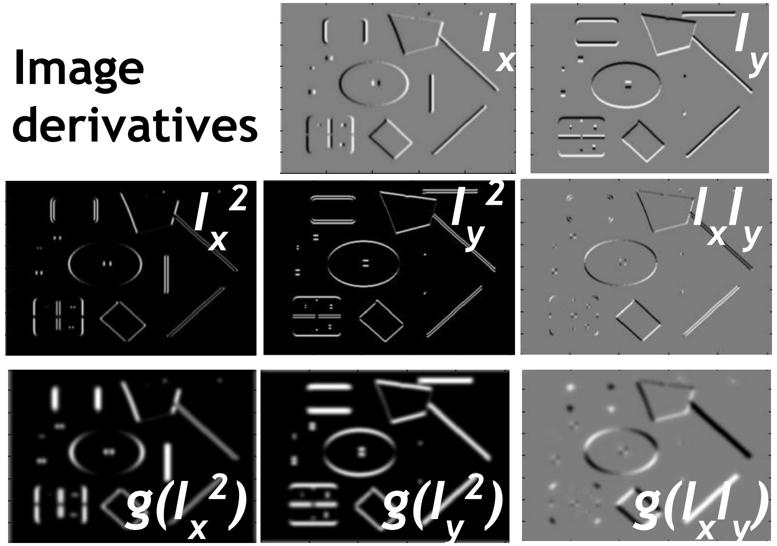
σ_D: for Gaussian in the derivative calculation

σ_I: for Gaussian in the windowing function

1. Image derivatives

2. Square of derivatives

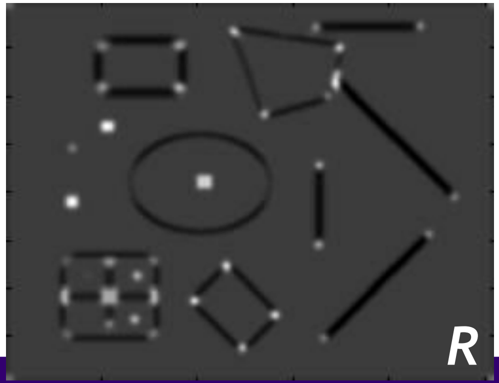
3. Gaussian filter $g(\sigma_I)$



4. Cornerness function – two strong eigenvalues

$$\theta = \det[M(\sigma_I,\sigma_D)] - \alpha[\mathrm{trace}(M(\sigma_I,\sigma_D))]^2$$
$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Perform non-maximum suppression

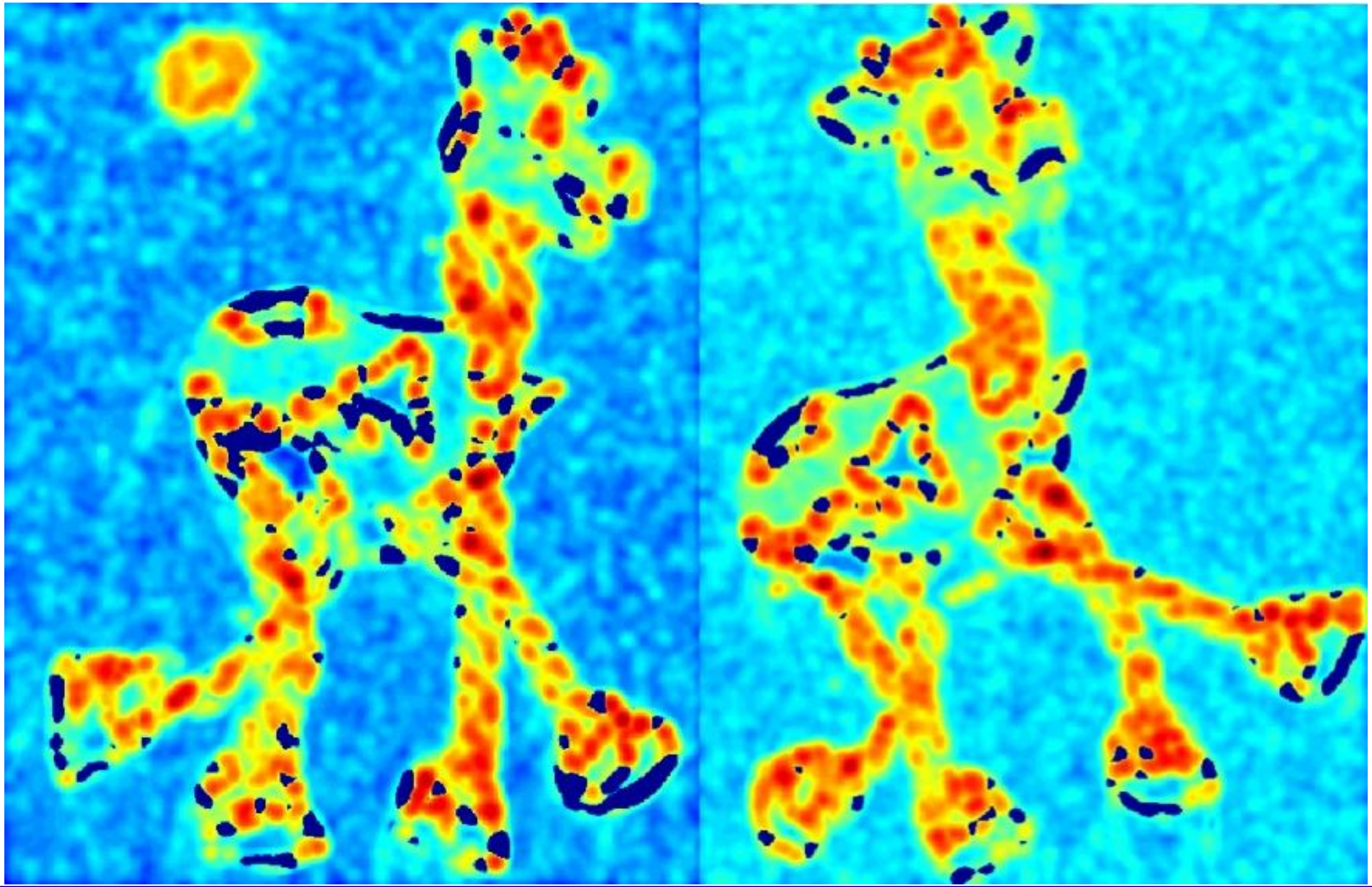


Harris Detector: Example

- Input Image

Harris Detector: Example

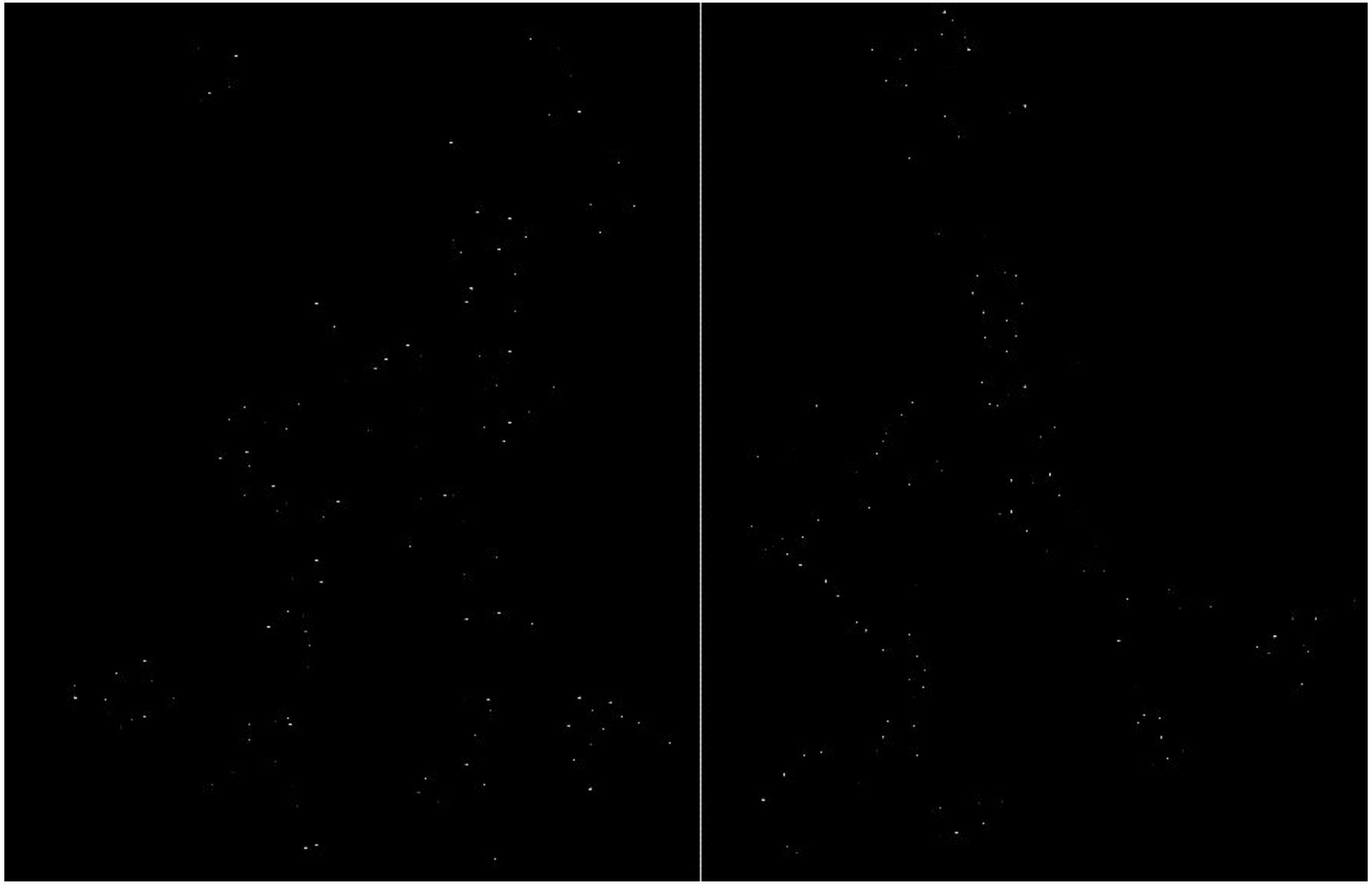
- Input Image
- Compute corner response function θ



Slide adapted from Daya Frolova, Denis Simakov

Harris Detector: Example

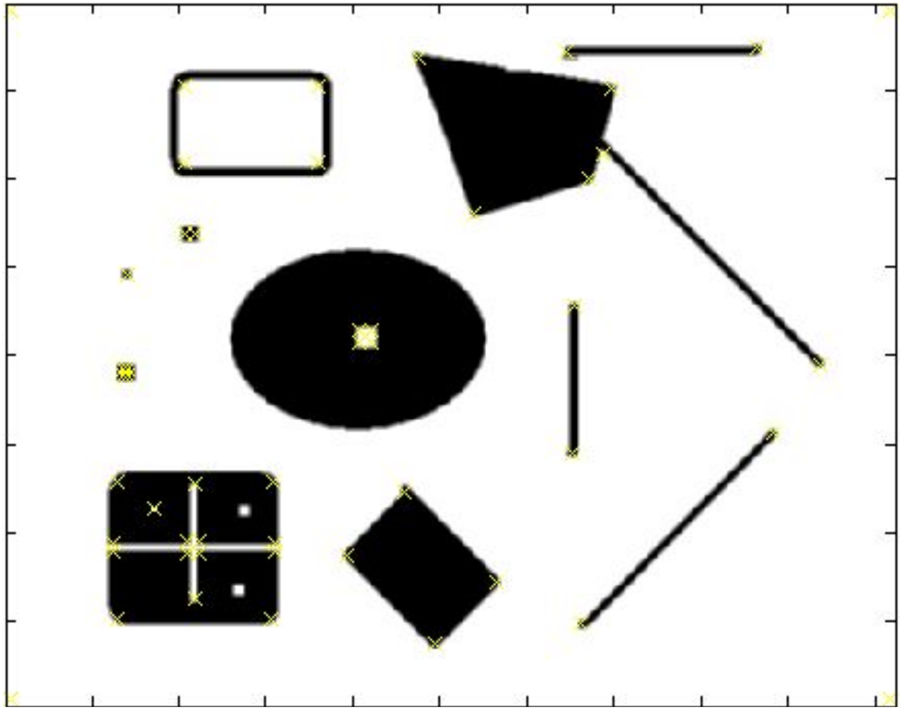
- Input Image
- Compute corner response function θ
- Take only the local maxima of θ, where θ > threshold



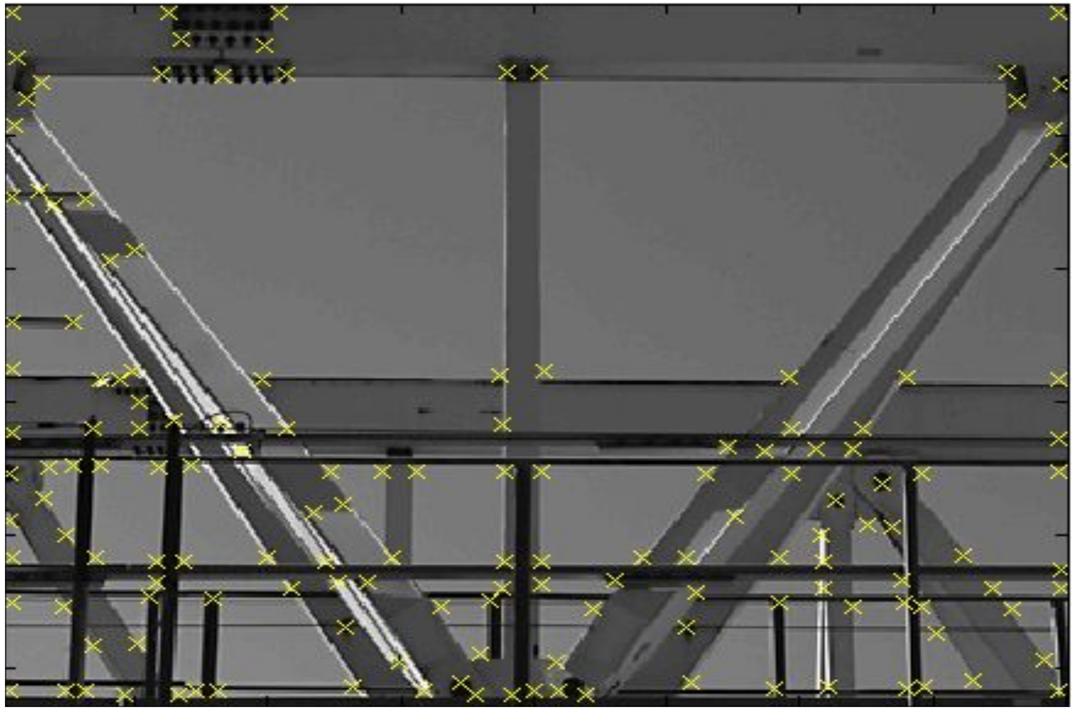
Harris Detector: Example

- Input Image
- Compute corner response function θ
- Take only the local maxima of θ, where θ > threshold

Harris Detector – Responses [Harris88]



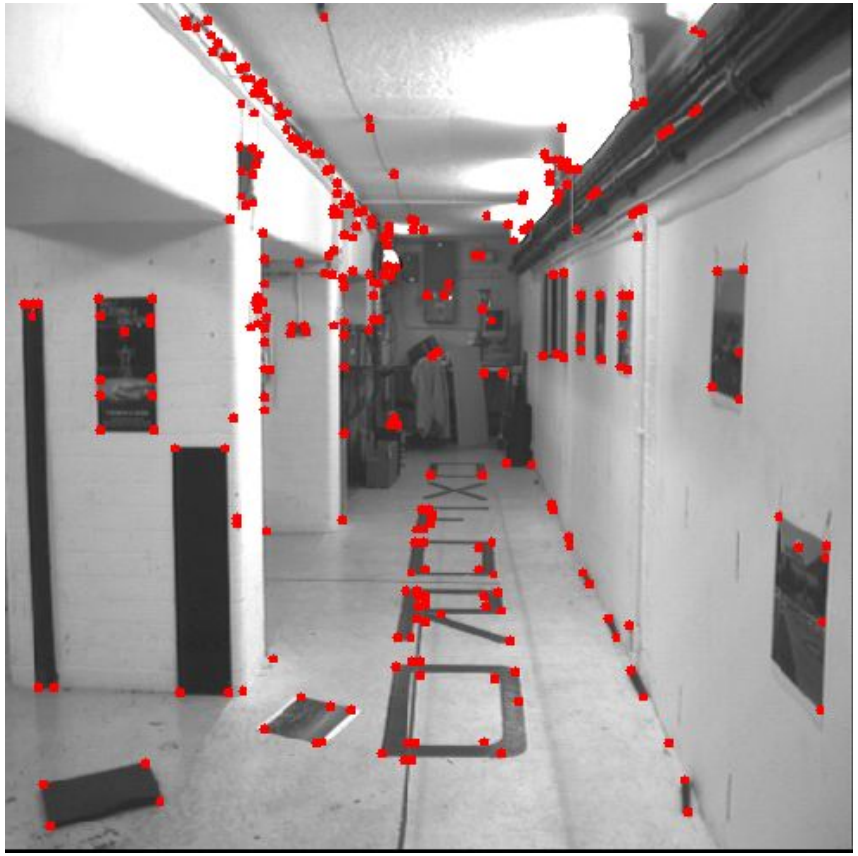
Effect: A very precise corner detector.



Harris Detector – Responses [Harris88]

Slide credit: Krystian Mikolajczyk

Harris Detector – Responses [Harris88]



● Results are great for finding correspondences matches between images

Slide credit: Kristen Grauman

Summary

- Local Invariant Features
- Harris Corner Detector

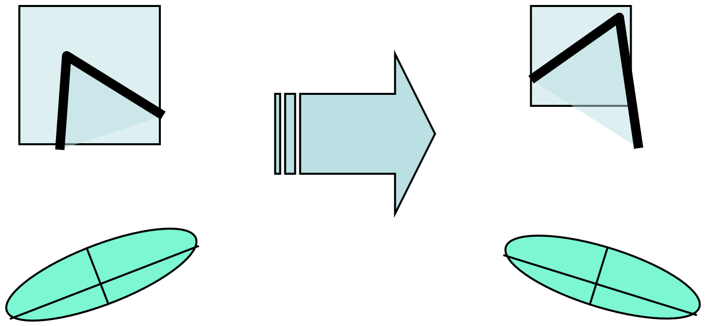
Harris Detector: Properties

- Translation invariance?

Slide credit: Kristen Grauman

Harris Detector: Properties

- Translation invariance
- Rotation invariance?

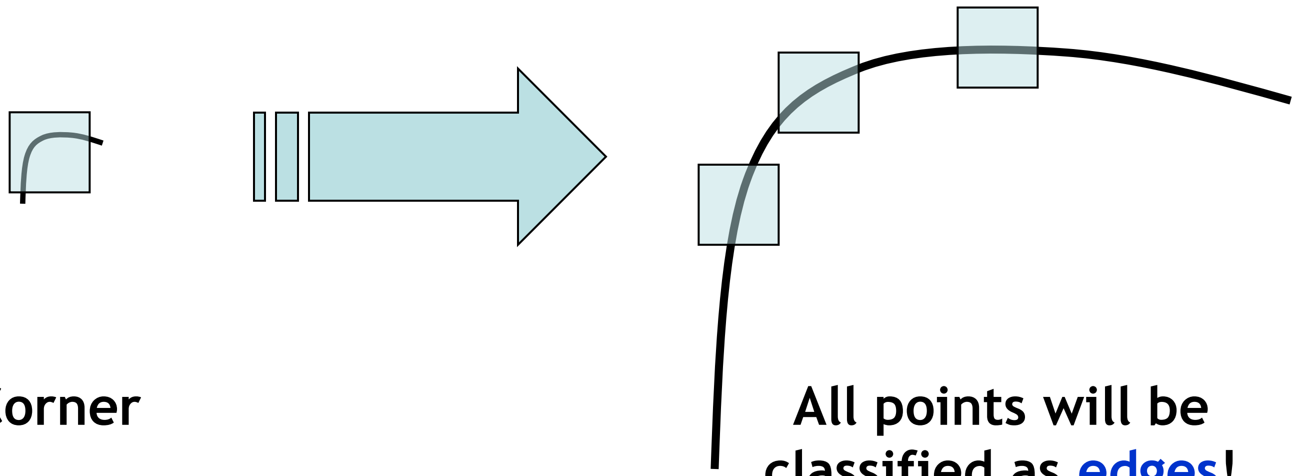


Ellipse rotates but its shape (i.e. eigenvalues) remains the same

***Corner response* θ is invariant to image rotation**

Harris Detector: Properties

- Translation invariance
- Rotation invariance
- Scale invariance?



Corner

All points will be classified as edges!

Not invariant to image scale!

Slide credit: Kristen Grauman

Next time

Detectors and Descriptors