By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

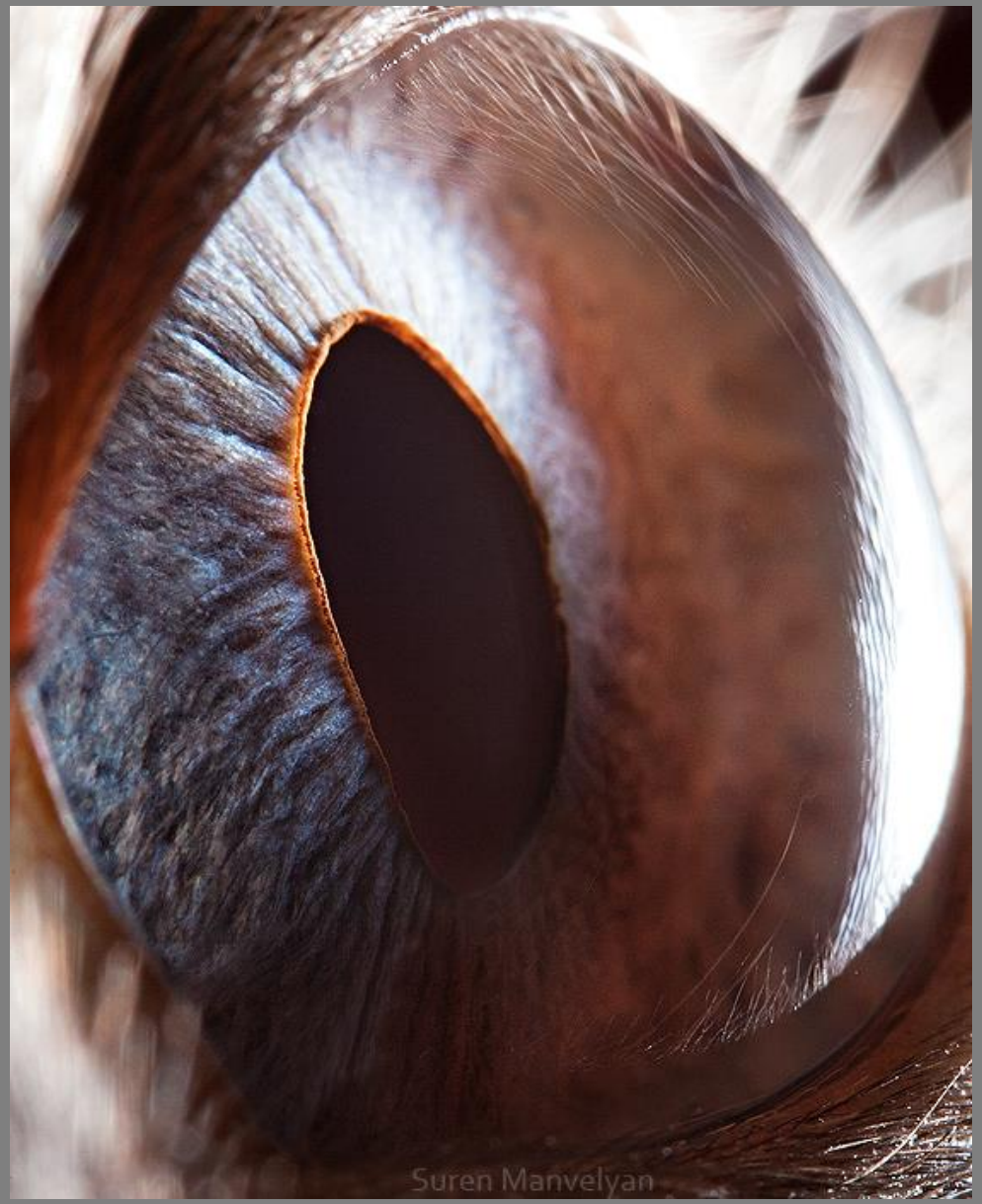
By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

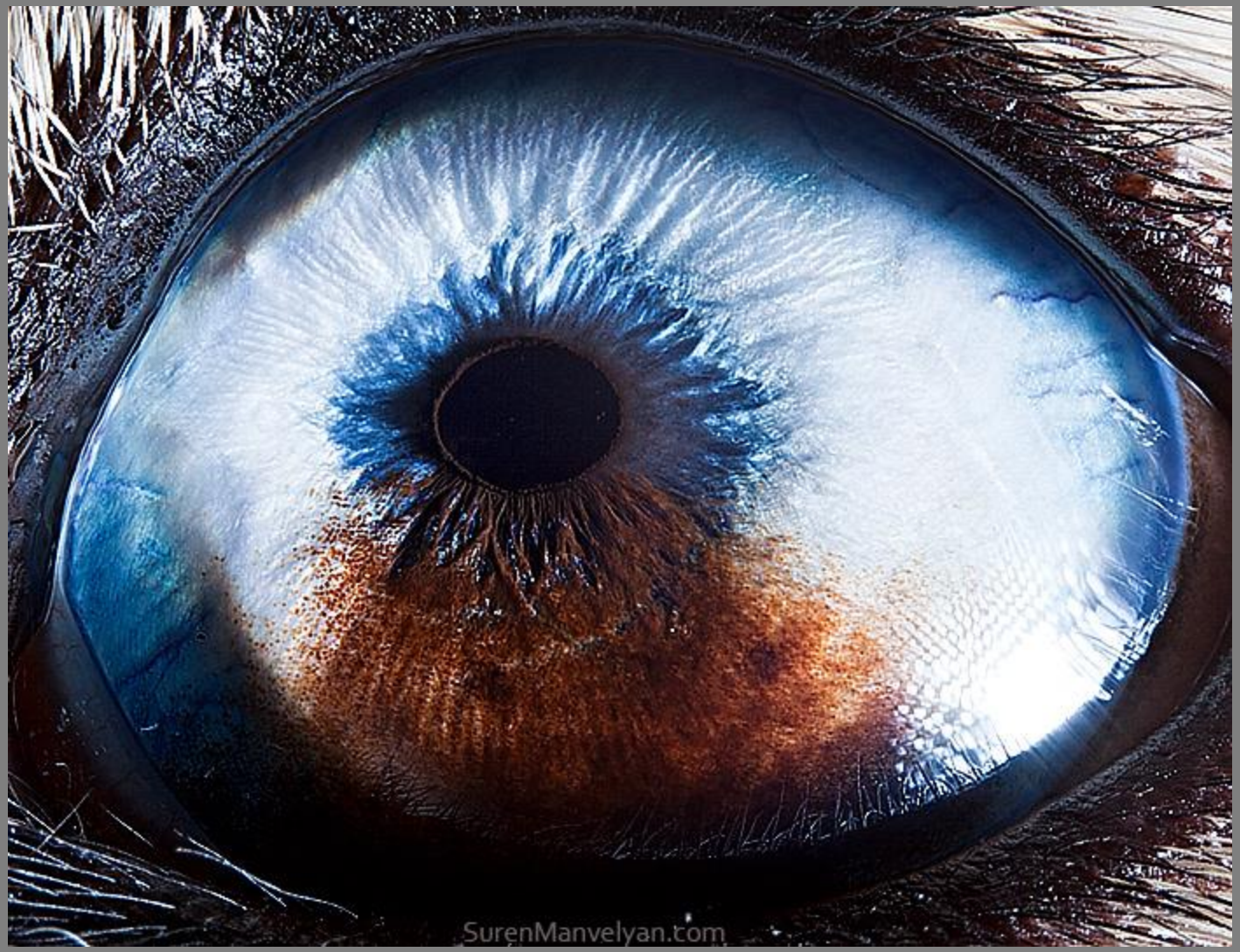
By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116



By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116



By Suren Manvelyan, http://www.surenmanvelyan.com/gallery/7116

Indy

Indy

Heterochromia iridum

From Wikipedia, the free encyclopedia

Not to be confused with Heterochromatin or Dichromatic (disambiguation).

In anatomy, **heterochromia** (ancient Greek: ἕτερος, *héteros*, different + χρώμα, *chróma*, color[1]) is a difference in coloration, usually of the iris but also of hair or skin. Heterochromia is a result of the relative excess or lack of melanin (a pigment). It may be inherited, or caused by genetic mosaicism, chimerism, disease, or injury.[2]

Heterochromia of the eye (***heterochromia iridis*** or ***heterochromia iridum***) is of three kinds. In *complete heterochromia*, one iris is a different color from the other. In *sectoral heterochromia*, part of one iris is a different color from its remainder and finally in "central heterochromia" there are spikes of different colours radiating from the pupil.

Heterochromia

Complete heterochromia in human eyes: one brown and one green/hazel

Classification and external resources

Specialty	ophthalmology
ICD-10	Q13.2, H20.8, L67.1
ICD-9-CM	364.53
OMIM	142500
DiseasesDB	31289

The Spectrum of Biological Inspiration

Radar

Lidar

Traditional cameras

Tesla autopilot

Humanoid and Quadruped robots

Less biologically inspired

More biologically inspired

Interest Points and Corners

Computer Vision

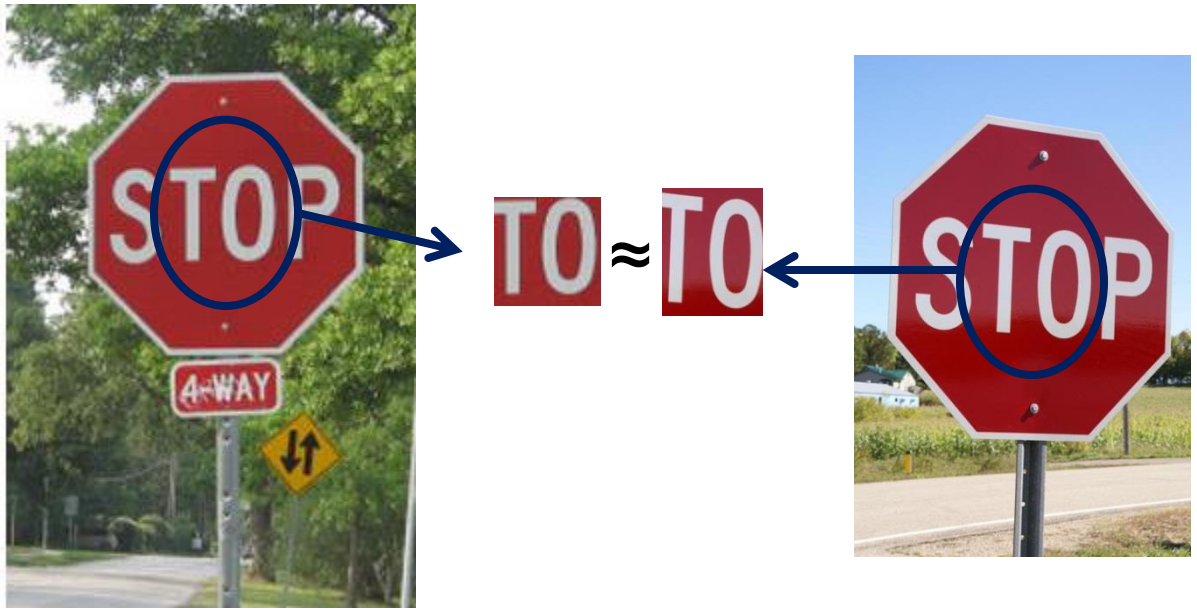
James Hays

Read Szeliski 7.1.1 and 7.1.2

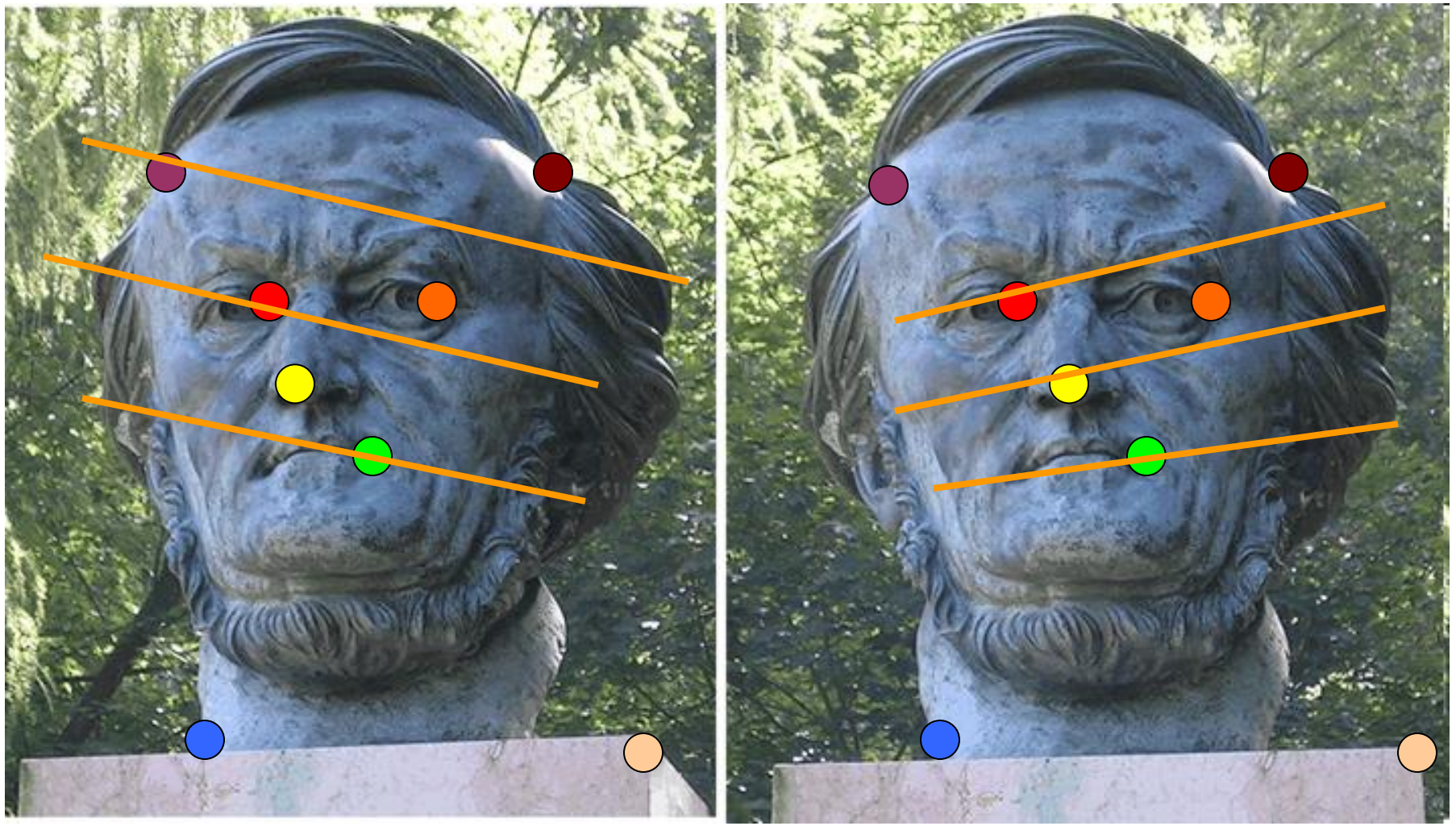
Slides from Rick Szeliski, Svetlana Lazebnik, Derek Hoiem and Grauman&Leibe 2008 AAAI Tutorial

Correspondence across views

- Correspondence: matching points, patches, edges, or regions across images



Example: estimating “fundamental matrix” that corresponds two views

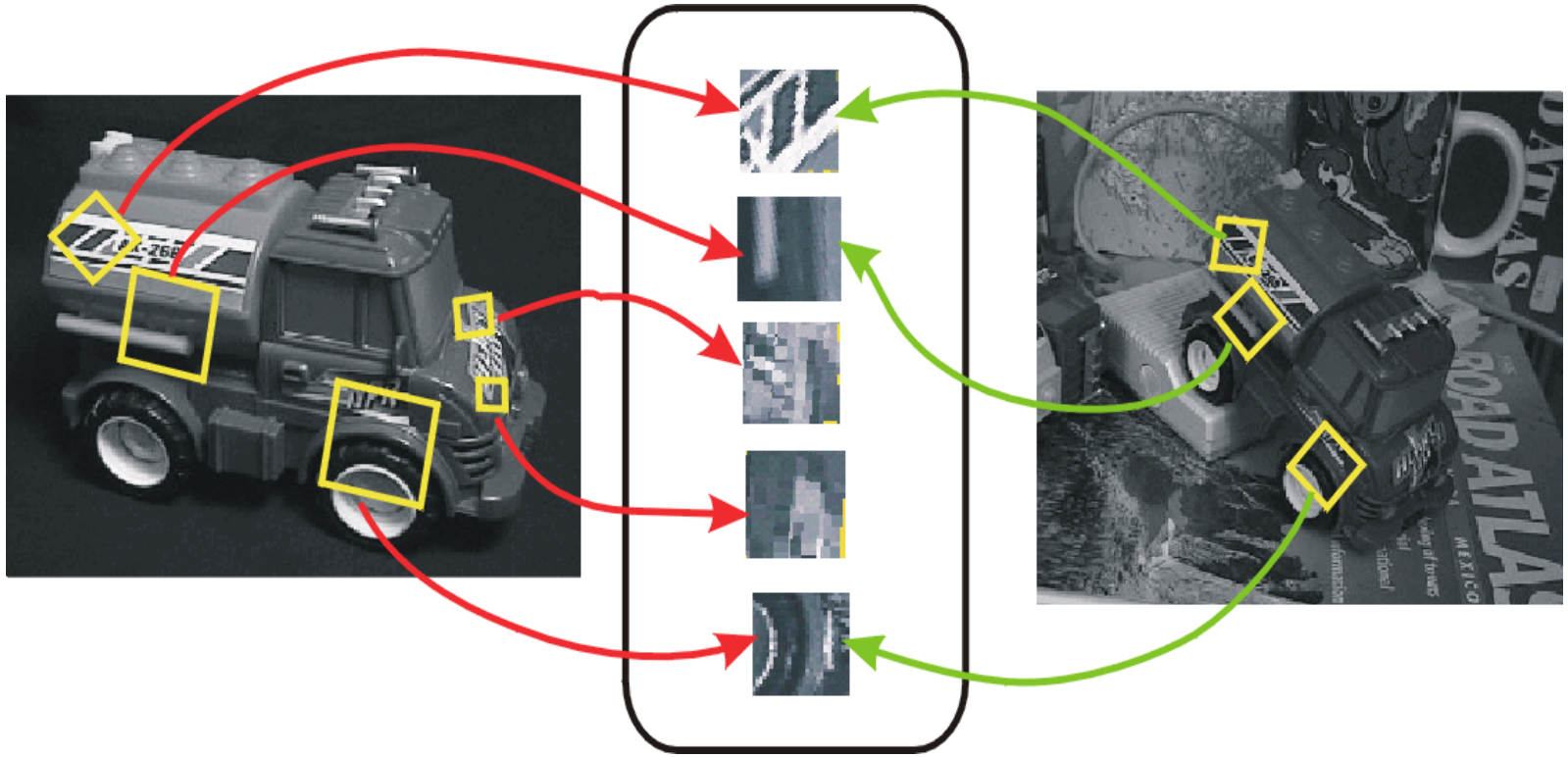


Slide from Silvio Savarese

Application: structure from motion

Invariant Local Features

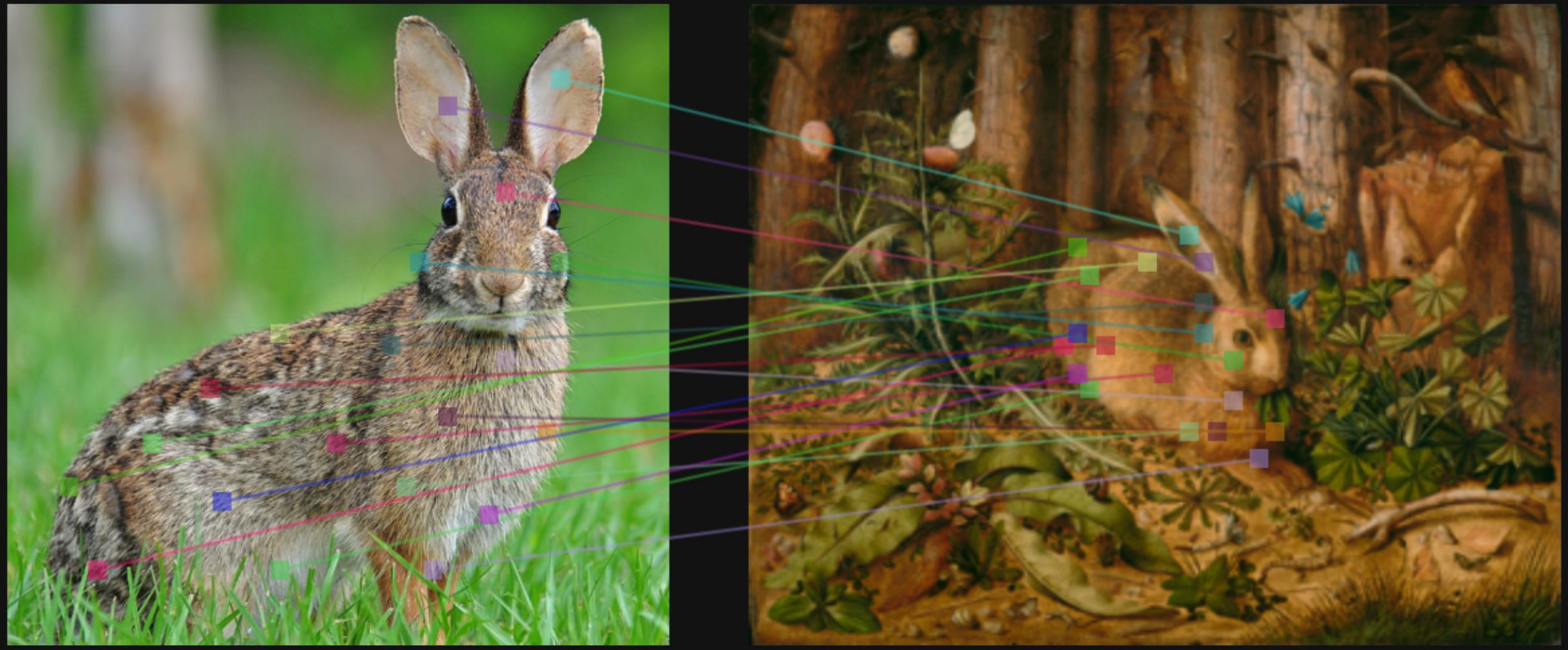
Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Features Descriptors

Sparse Matching

DINOv2's frozen features are relevant to recognize the main objects in an image and to consistently encode similar parts across images. Here we match most similar patches across two images.

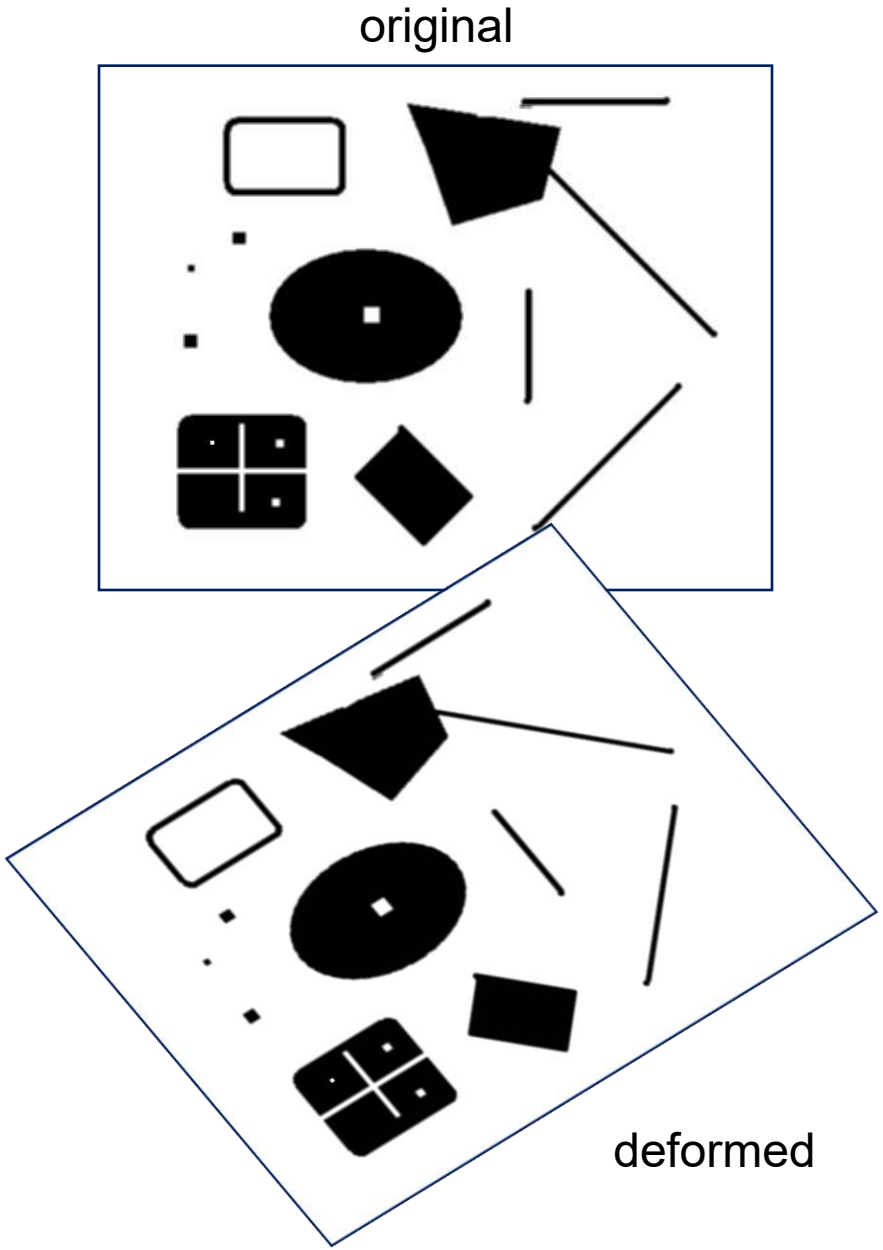


Project 2: interest points and local features

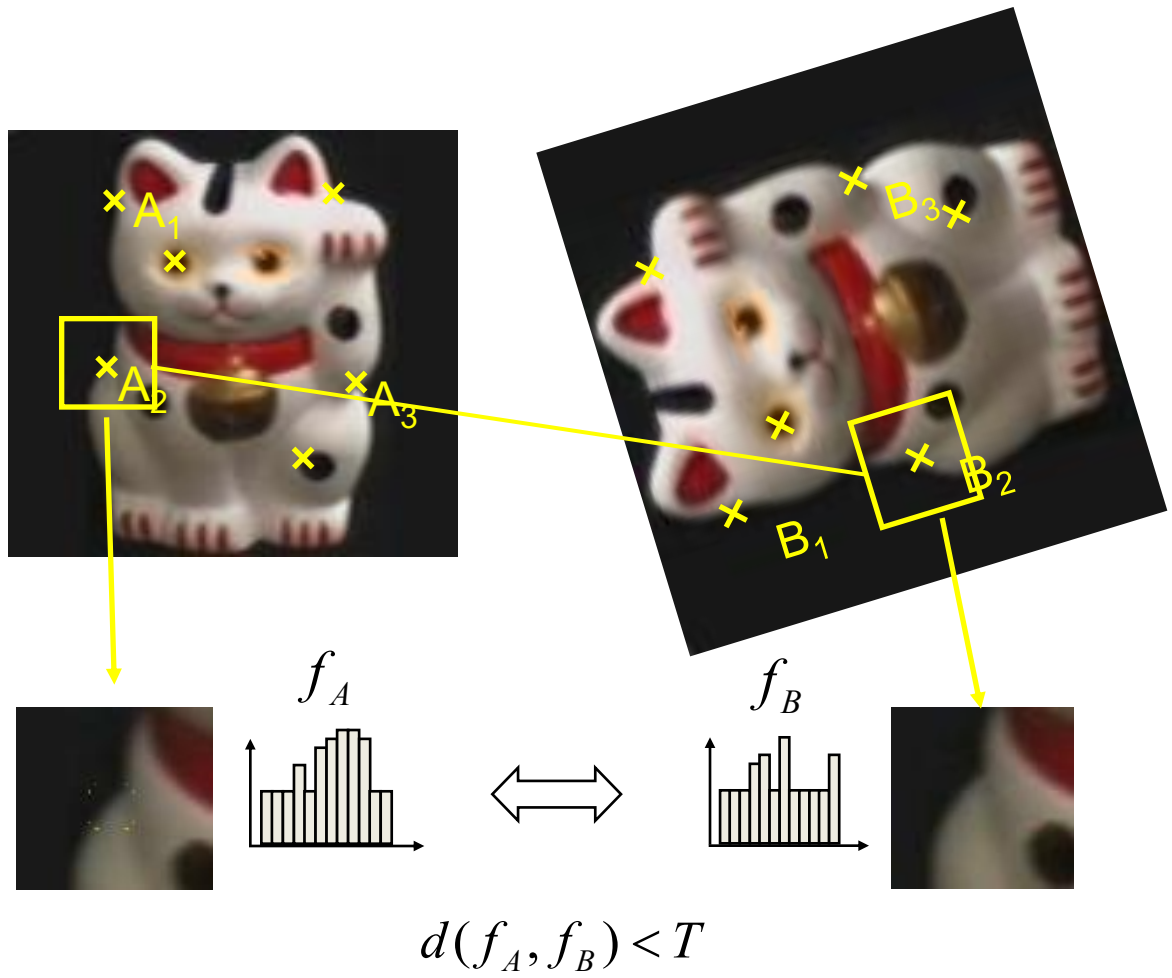
I use “interest points” and “keypoints” and “SIFT features” interchangeably

This class: interest points

- Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
  - Which points would you choose?



Overview of Keypoint Matching



1. Find a set of distinctive keypoints

2. Compute a local descriptor from the region around each keypoint

3. Match local descriptors

Goals for Keypoints

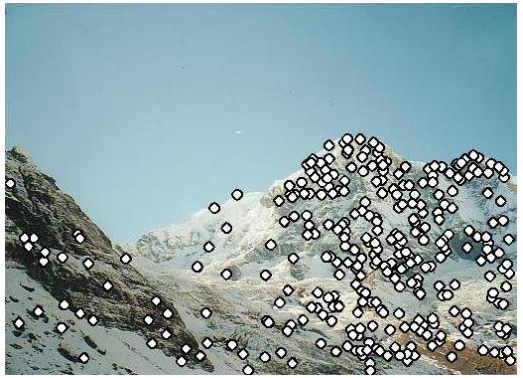
Detect points that are *repeatable* and *distinctive*

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

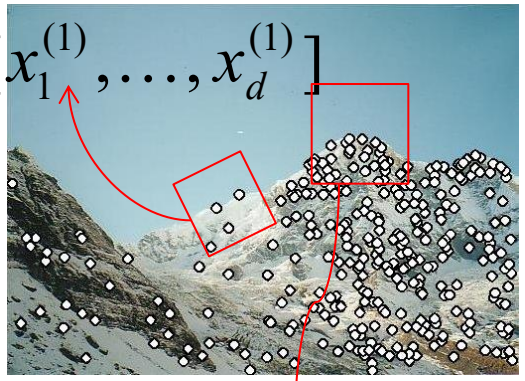
Local features: main components

1) Detection: Identify the interest points



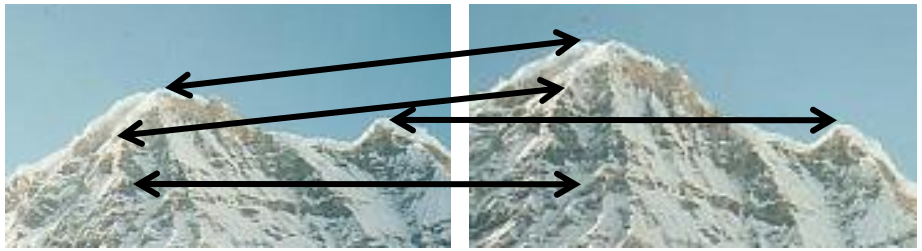
2) Description: Extract vector feature descriptor surrounding each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

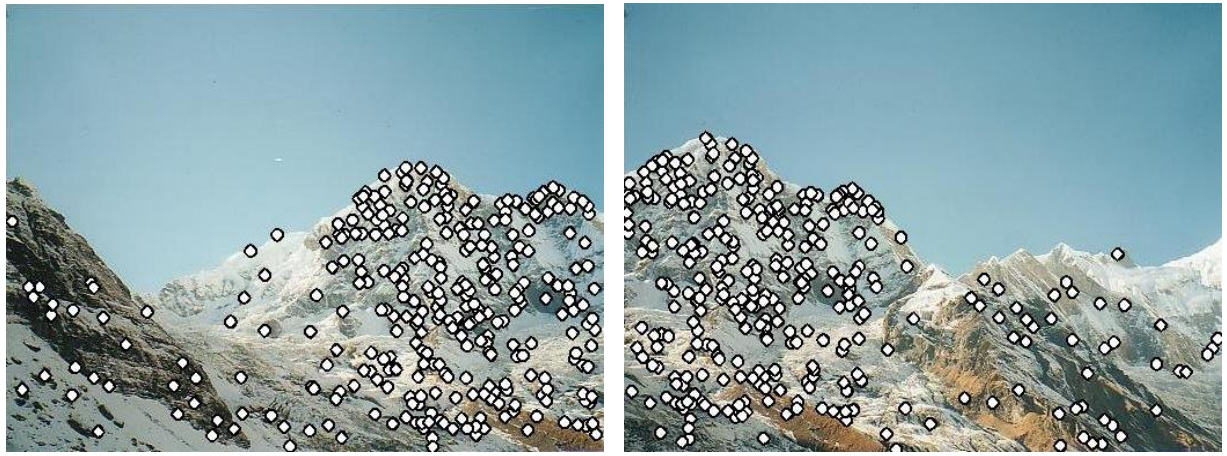


$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) Matching: Determine correspondence between descriptors in two views



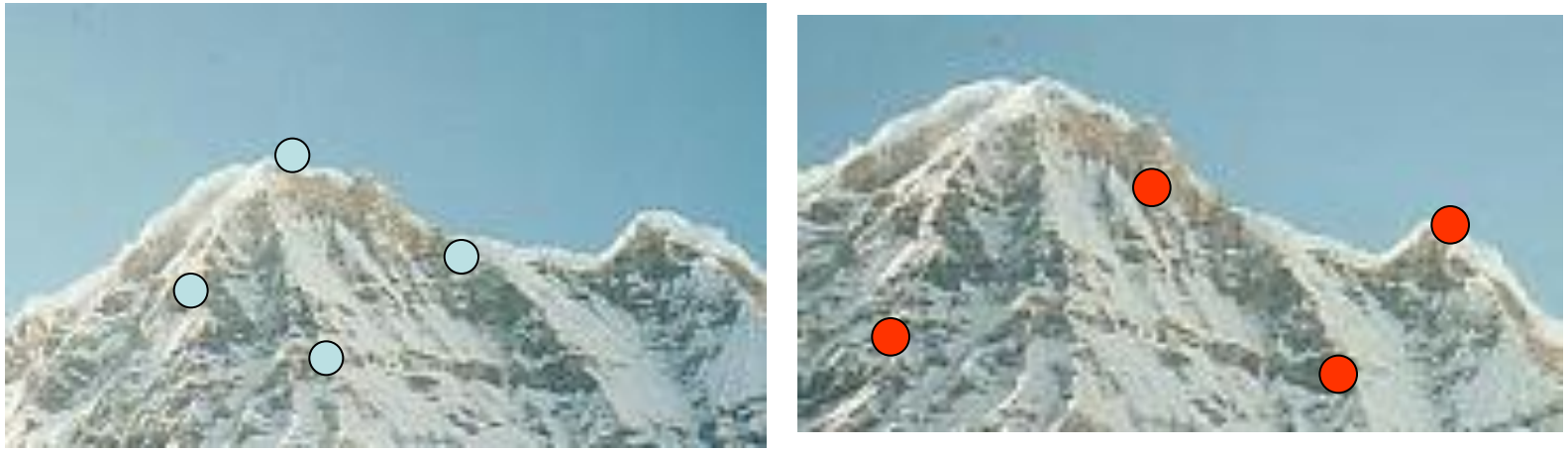
Characteristics of good features



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

Goal: interest operator repeatability

- We want to detect (at least some of) the same points in both images.

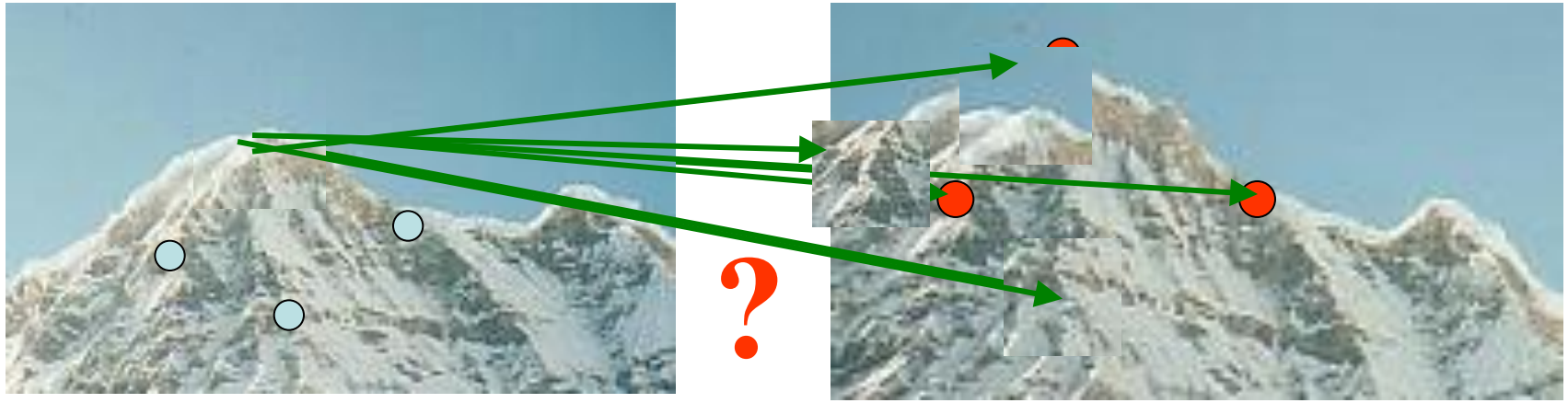


No chance to find true matches!

- Yet we have to be able to run the detection procedure *independently* per image.

Goal: descriptor distinctiveness

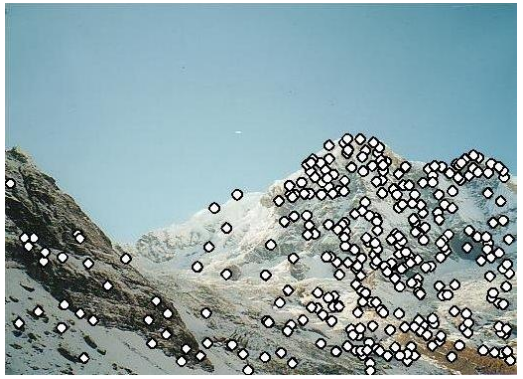
- We want to be able to reliably determine which point goes with which.



- Must provide some invariance to geometric and photometric differences between the two views.

Local features: main components

1) Detection: Identify the interest points



2) Description:Extract vector feature descriptor surrounding each interest point.

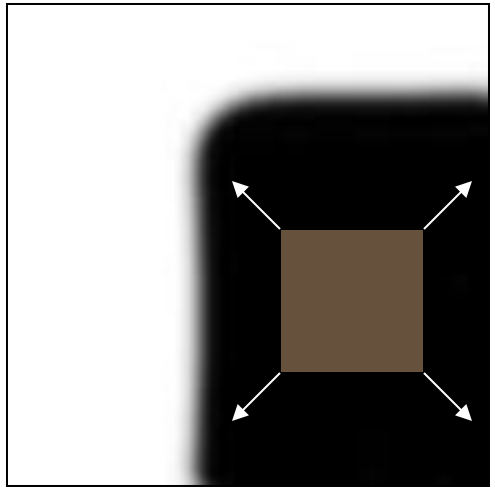
3) Matching: Determine correspondence between descriptors in two views

Many Existing Detectors Available

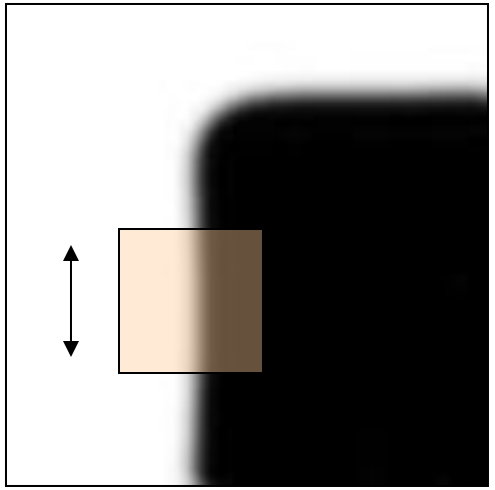
Hessian & Harris	[Beaudet ‘78], [Harris ‘88]
Laplacian, DoG	[Lindeberg ‘98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid ‘01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid ‘04]
EBR and IBR	[Tuytelaars & Van Gool ‘04]
MSER	[Matas ‘02]
Salient Regions	[Kadir & Brady ‘01]
Others…	

Corner Detection: Basic Idea

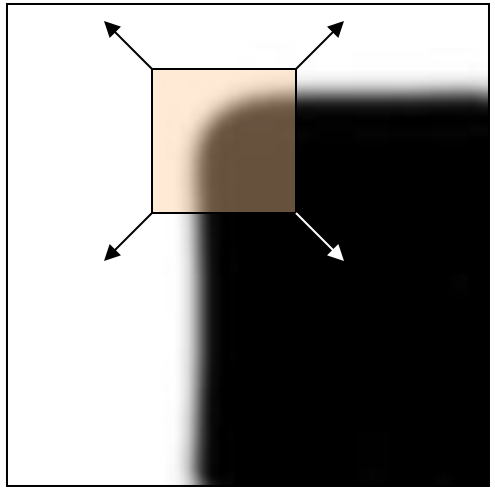
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region:
no change in all directions



"edge":
no change along the edge direction

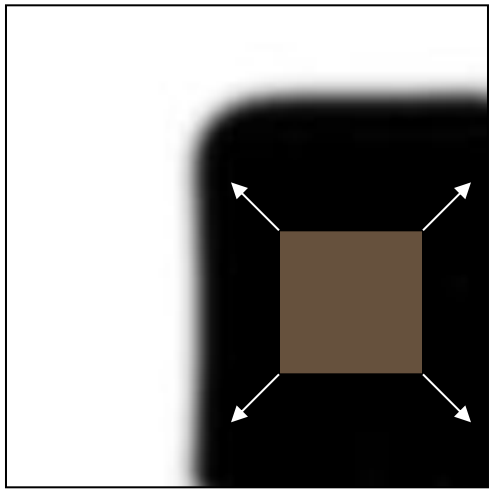


"corner":
significant change in all directions

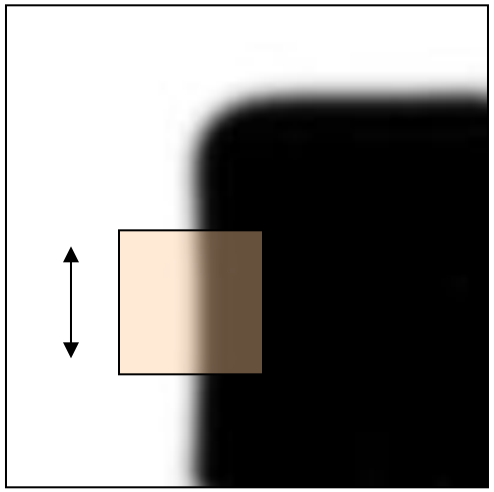
Source: A. Efros

Corner Detection: Baseline strategies

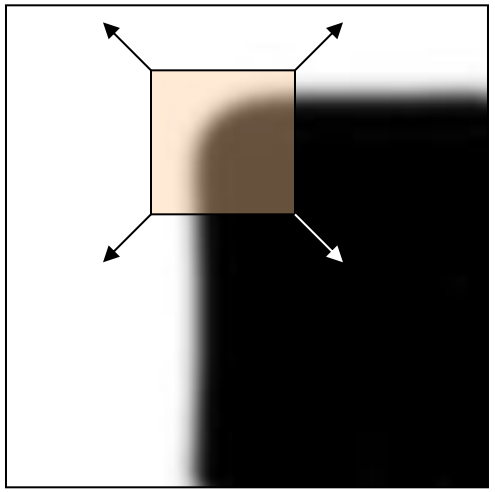
- First, cornerness is a property of a “patch”, not a single pixel
- Let’s look for patches that have high gradients in the x and y directions.



“flat” region: no gradients



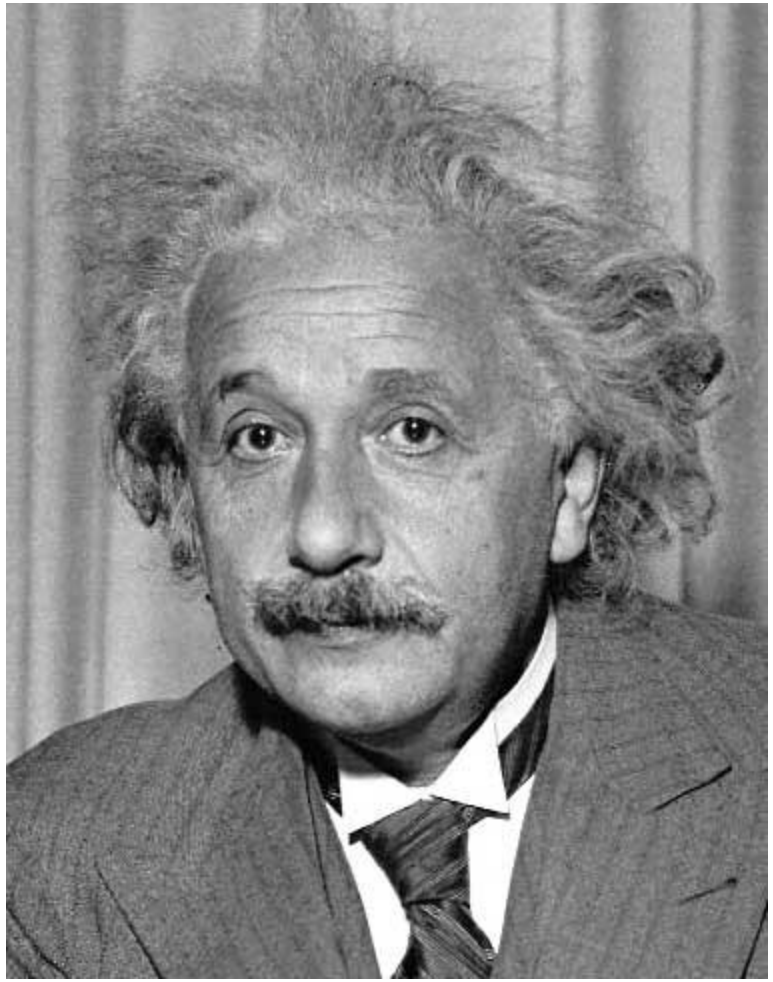
“edge”: gradients in one direction



“corner”: gradients in both directions

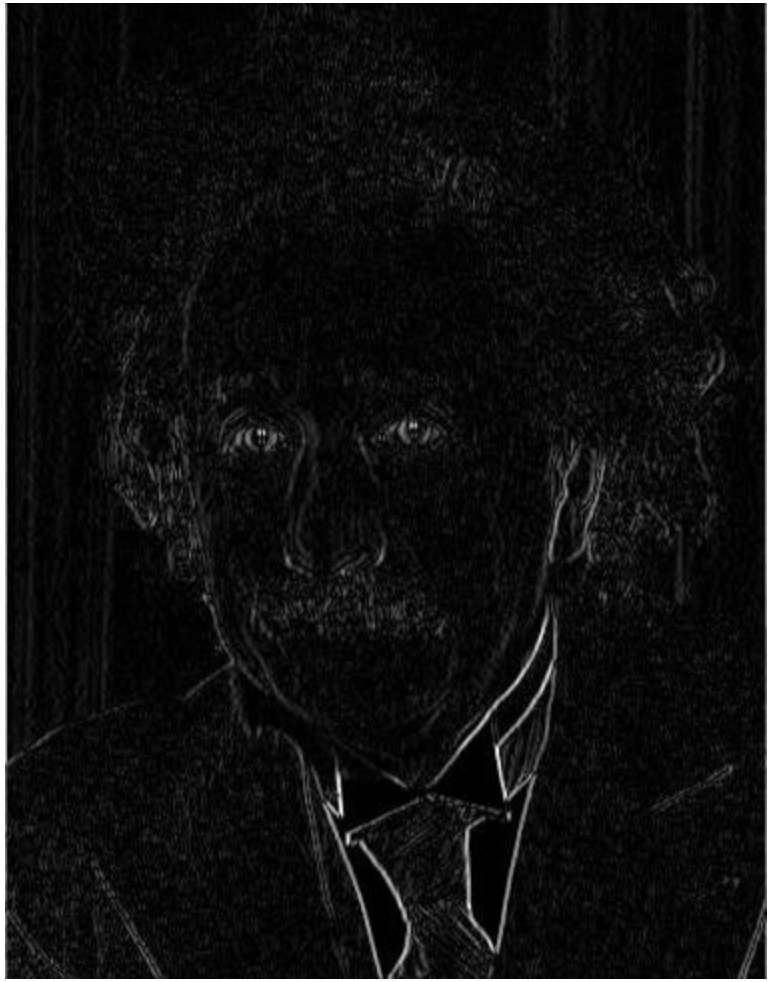
Source: James Hays

Reminder: gradients measured with filtering



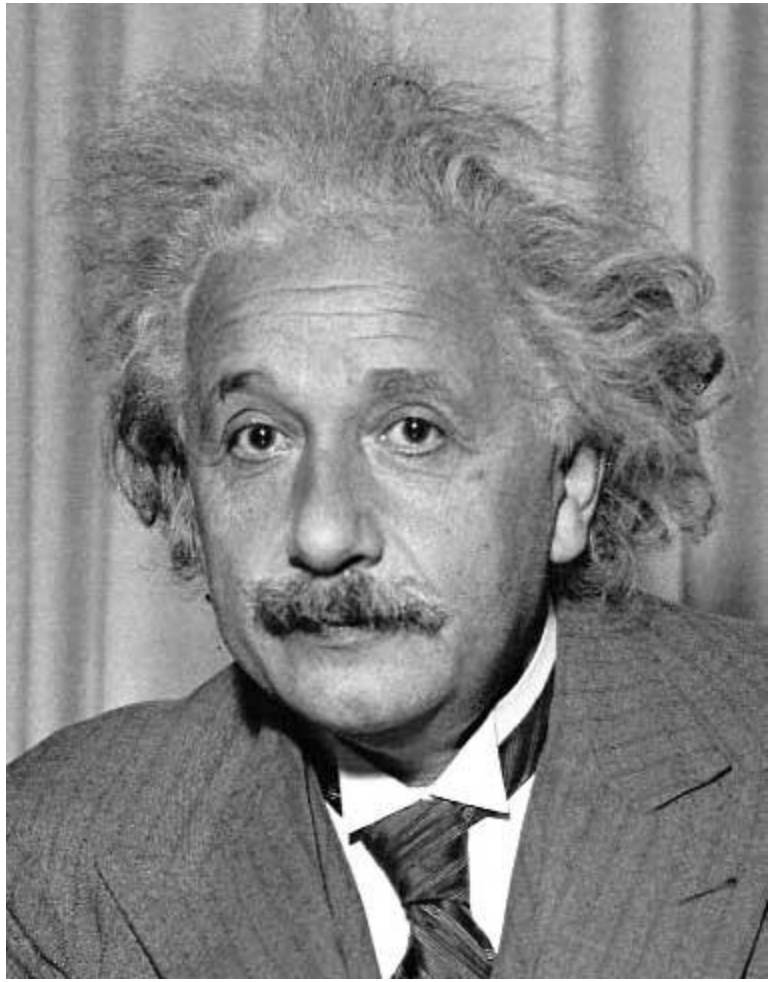
1	0	-1
2	0	-2
1	0	-1

Sobel



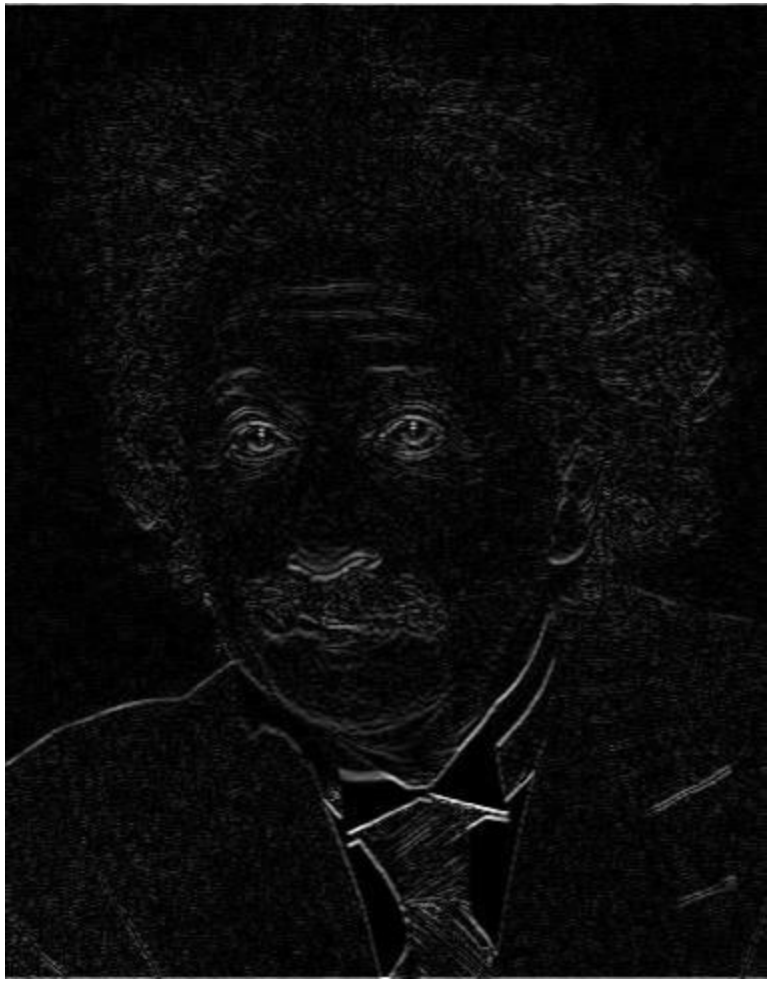
Vertical Edge
(absolute value)

Reminder: gradients measured with filtering



1	2	1
0	0	0
-1	-2	-1

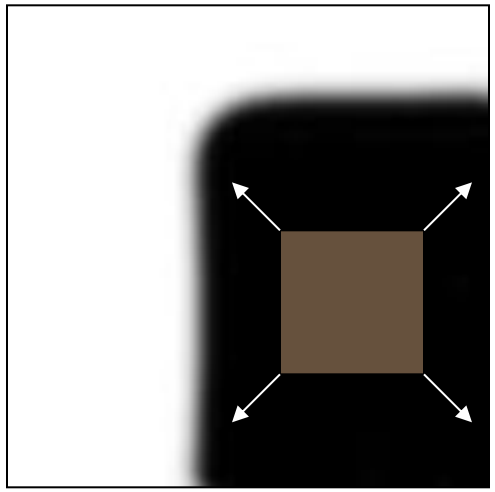
Sobel



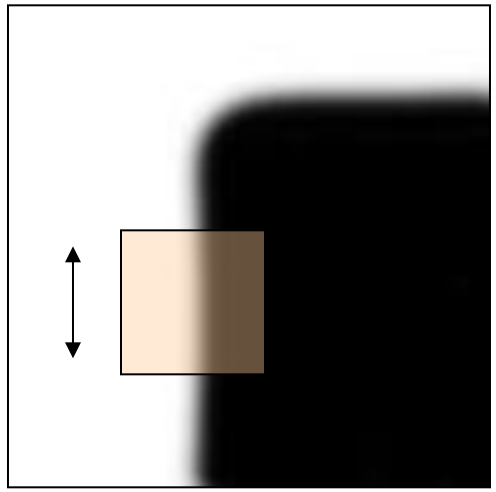
Horizontal Edge
(absolute value)

Corner Detection: Baseline strategies

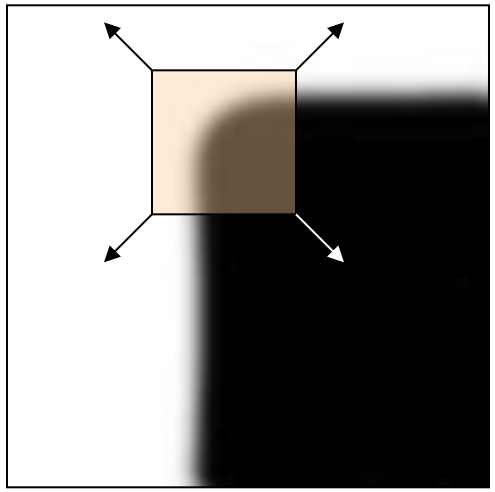
- First, cornerness is a property of a “patch”, not a single pixel
- Let’s look for patches that have high gradients in the x and y directions.



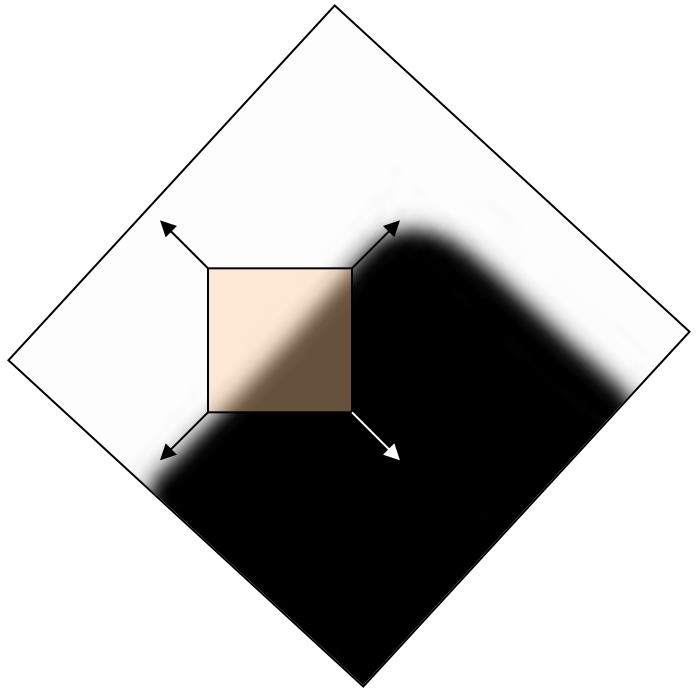
“flat” region: no gradients



“edge”: gradients in one direction



“corner”: gradients in both directions



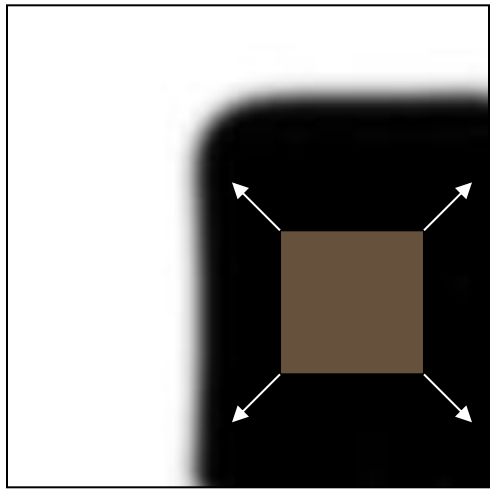
“edge”: gradients in both directions

Source: James Hays

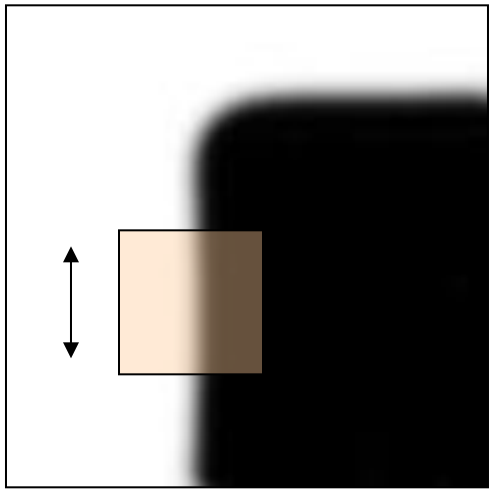
Corner Detection: Baseline strategies

- ~~Let's look for patches that have high gradients in the x and y directions.~~

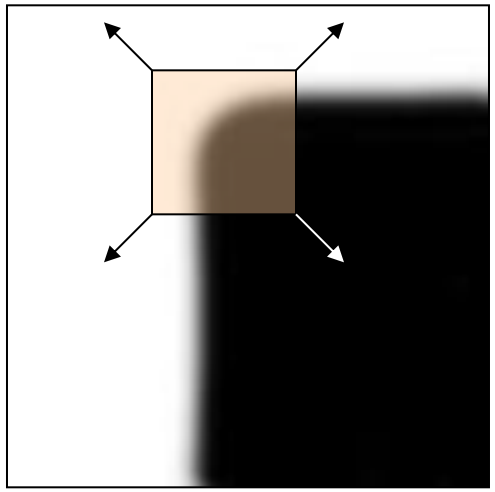
Not a sufficient strategy



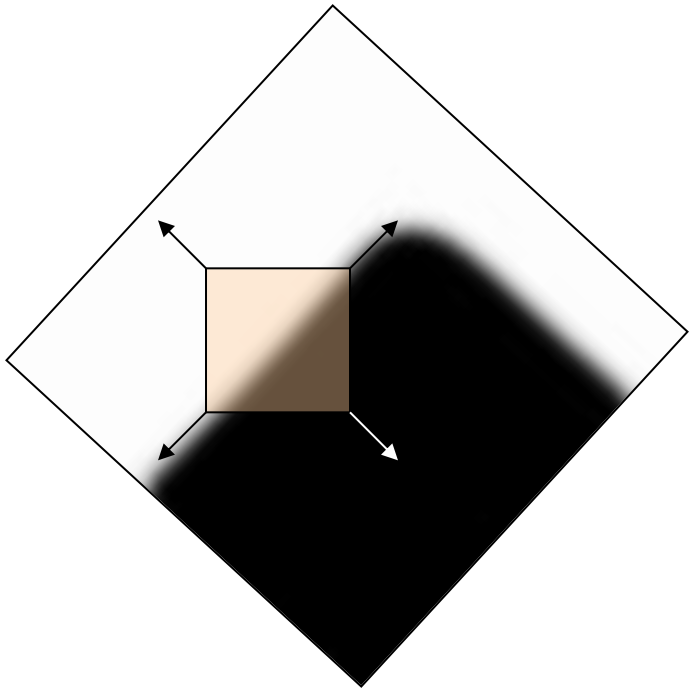
"flat" region: no gradients



"edge": gradients in one direction



"corner": gradients in both directions

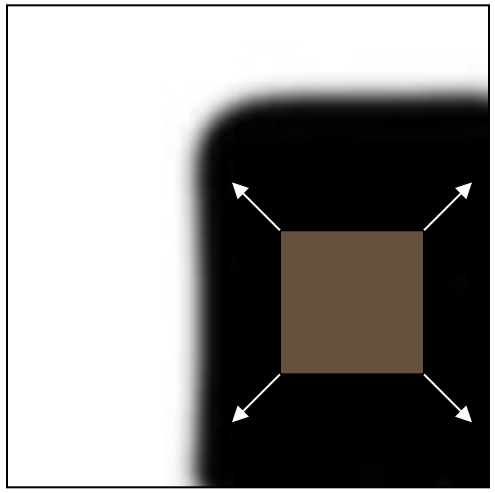


"edge": gradients in both directions

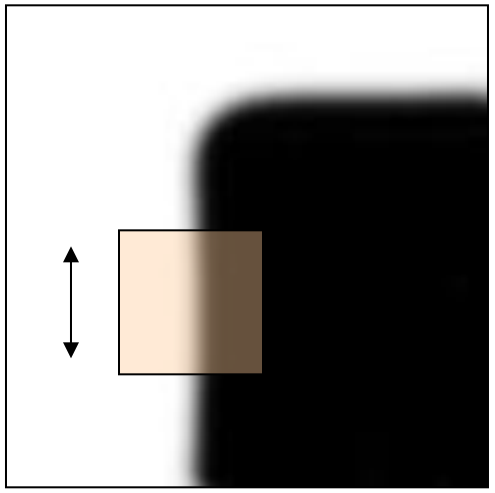
Source: James Hays

Corner Detection: Baseline strategies

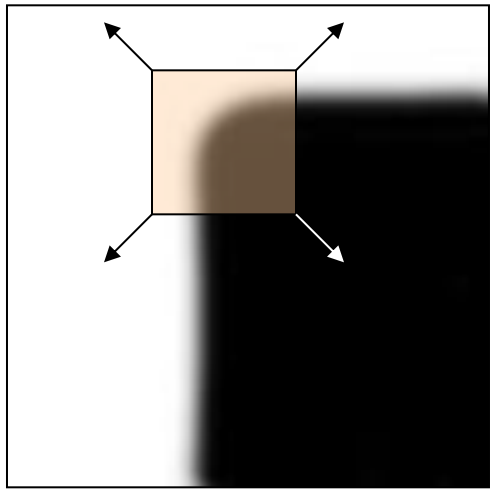
- Let's write down what the gradients actually look like in different scenarios



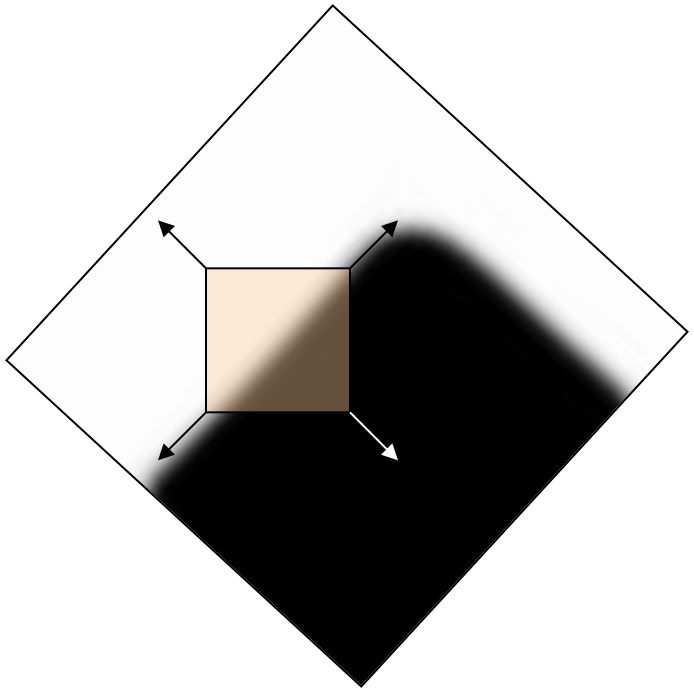
"flat" region:
no gradients



"edge":
gradients in one direction



"corner":
gradients in both directions

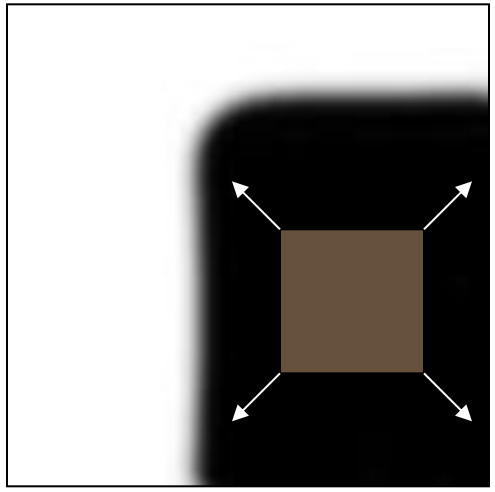


"edge":
gradients in both directions

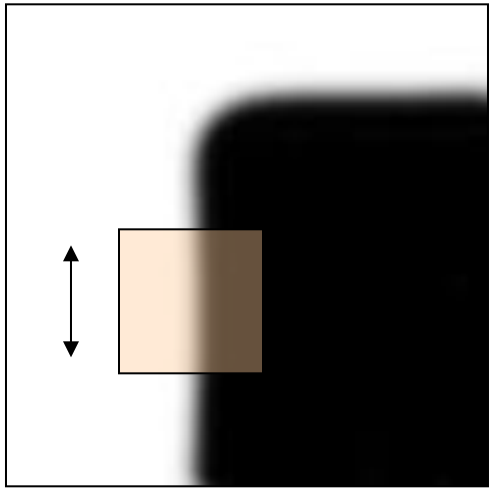
Source: James Hays

Corner Detection: Baseline strategies

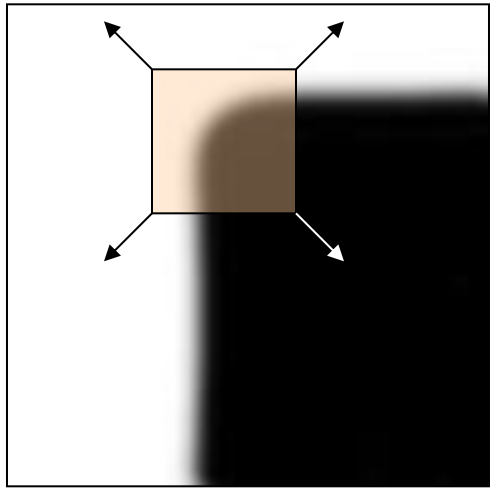
- For a patch to be a corner, the gradient distribution needs to be full rank
- We should check more than 2 pixels
- How do we measure this rank?



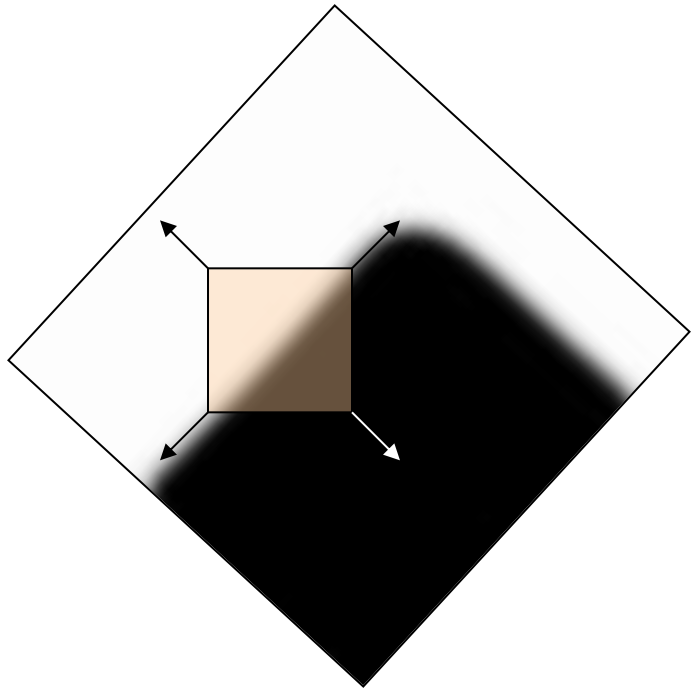
"flat" region:
no gradients



"edge":
gradients in one direction



"corner":
gradients in both directions

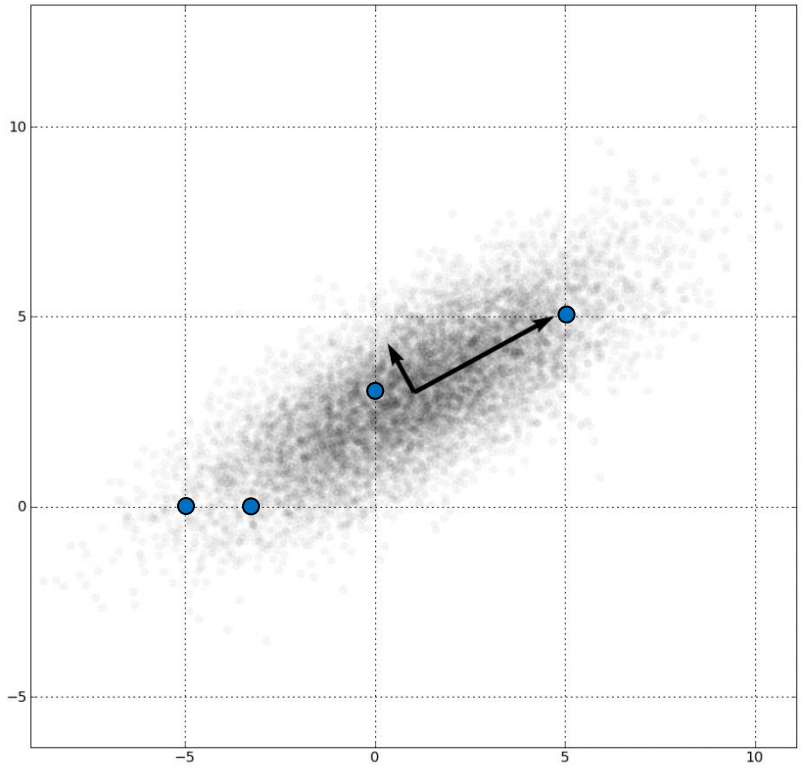


"edge":
gradients in both directions

Source: James Hays

Eigenvalues tell us the rank

I = [-5, 0
0, 3
-3, 0
5, 5
…
...]

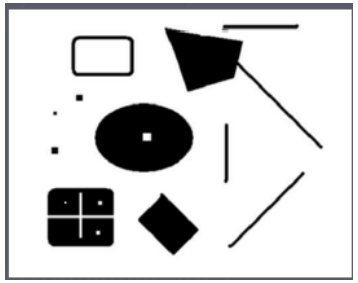


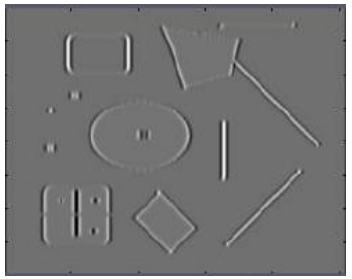
https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

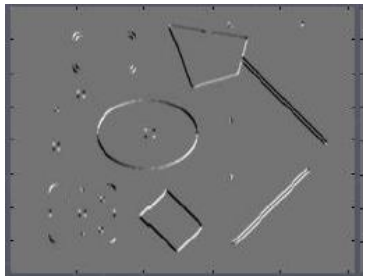
Corners as distinctive interest points

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

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).







Notation:

$$I_x \Leftrightarrow \frac{\partial I}{\partial x} \qquad I_y \Leftrightarrow \frac{\partial I}{\partial y} \qquad I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Using a Taylor Series expansion of the image function $I_0(\mathbf{x}_i + \Delta\mathbf{u}) \approx I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u}$ (Lucas and Kanade 1981; Shi and Tomasi 1994), we can approximate the auto-correlation surface as

$$E_{\mathrm{AC}}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2 \tag{7.3}$$

$$\approx \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u} - I_0(\mathbf{x}_i)]^2 \tag{7.4}$$

$$= \sum_i w(\mathbf{x}_i)[\nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u}]^2 \tag{7.5}$$

$$= \Delta\mathbf{u}^T \mathbf{A} \Delta\mathbf{u}, \tag{7.6}$$

where

$$\nabla I_0(\mathbf{x}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y}\right)(\mathbf{x}_i) \tag{7.7}$$

is the *image gradient* at $\mathbf{x}_i$. This gradient can be computed using a variety of techniques (Schmid, Mohr, and Bauckhage 2000). The classic "Harris" detector (Harris and Stephens 1988) uses a [-2 -1 0 1 2] filter, but more modern variants (Schmid, Mohr, and Bauckhage 2000; Triggs 2004) convolve the image with horizontal and vertical derivatives of a Gaussian (typically with $\sigma = 1$).

The auto-correlation matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{7.8}$$

Different derivations exist.

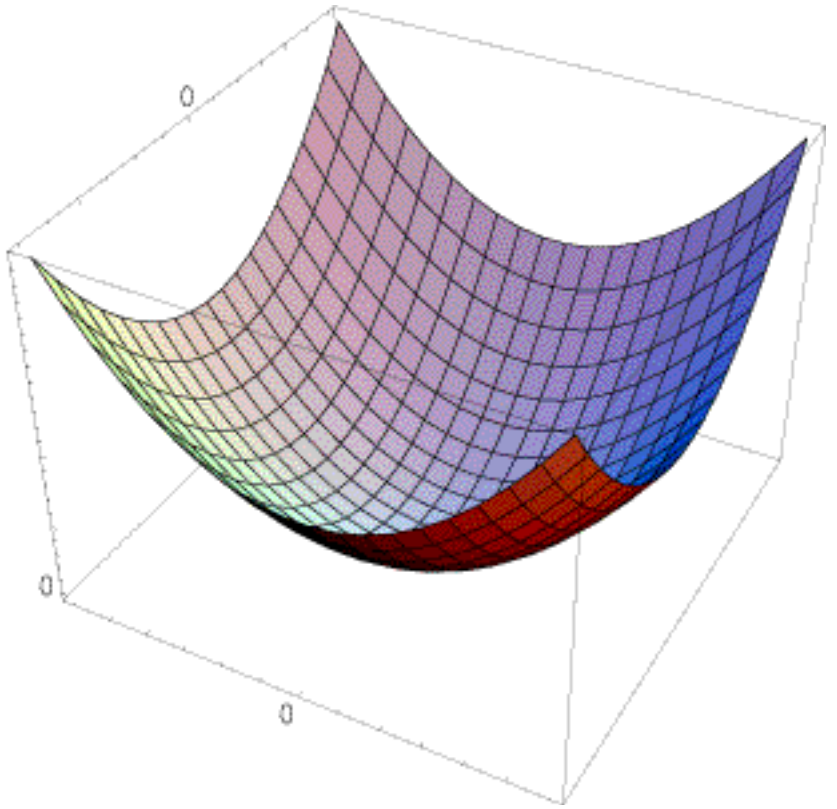
This is the textbook version.

Interpreting the second moment matrix

The surface $E(u,v)$ is locally approximated by a quadratic form. Let's try to understand its shape.

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

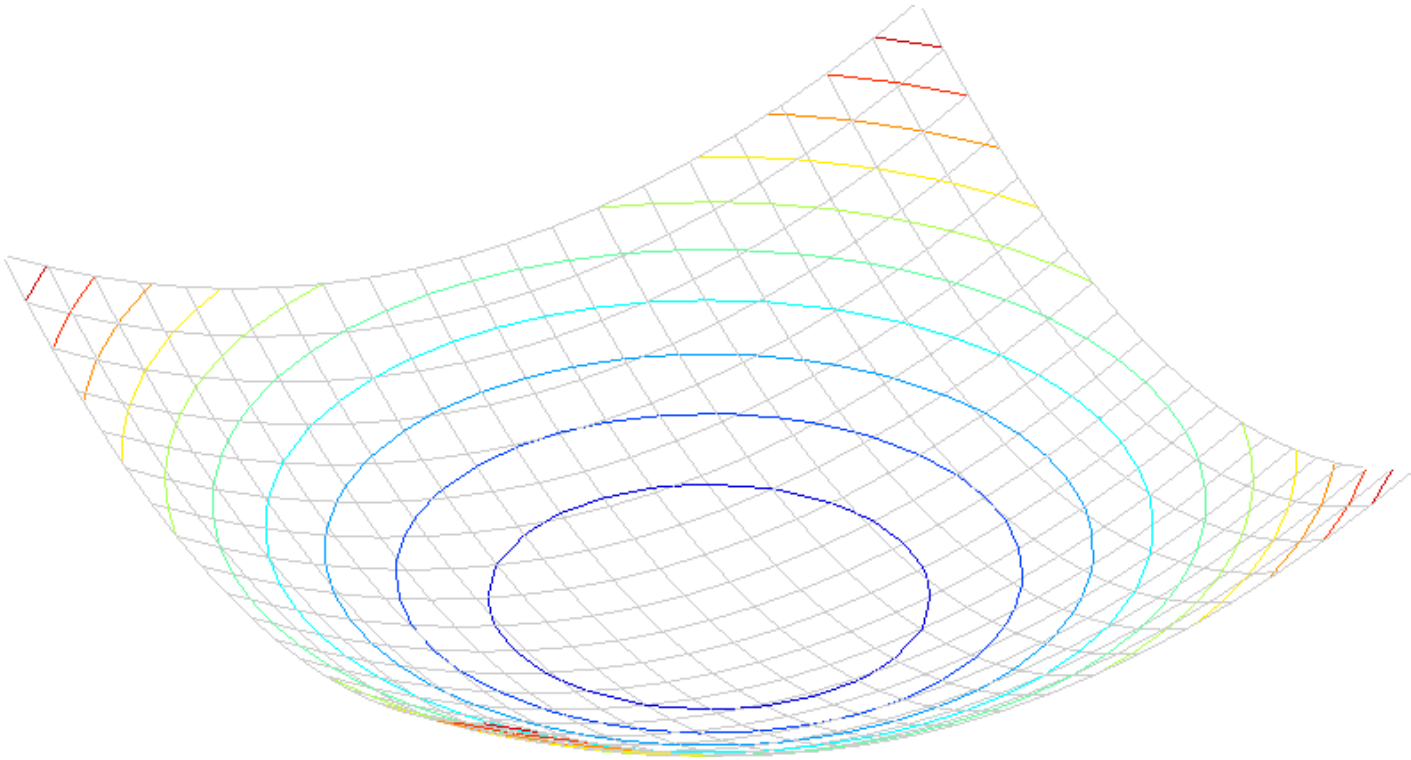
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $\begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.



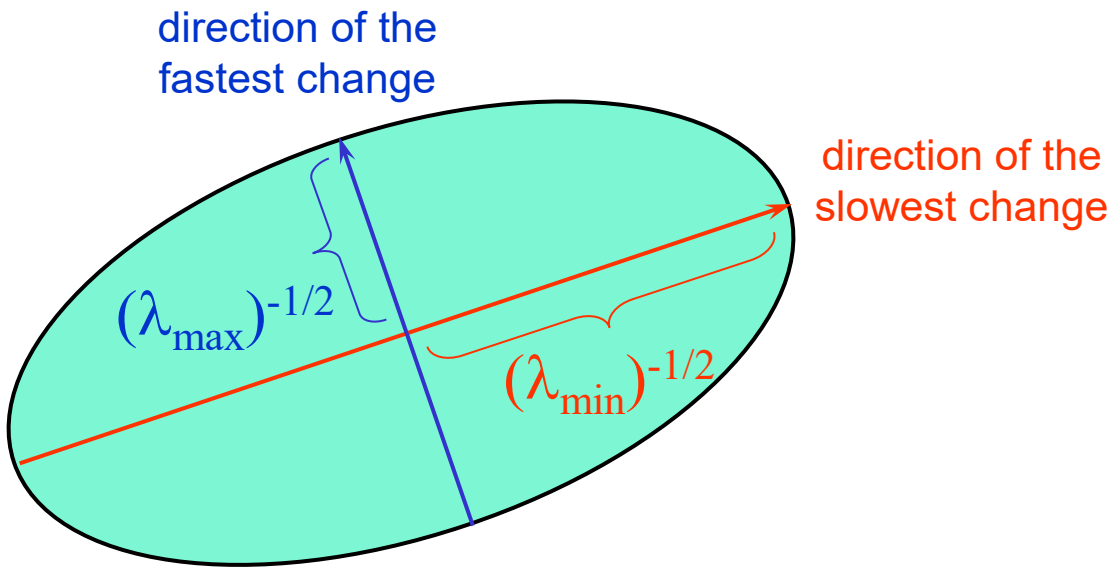
Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $\begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

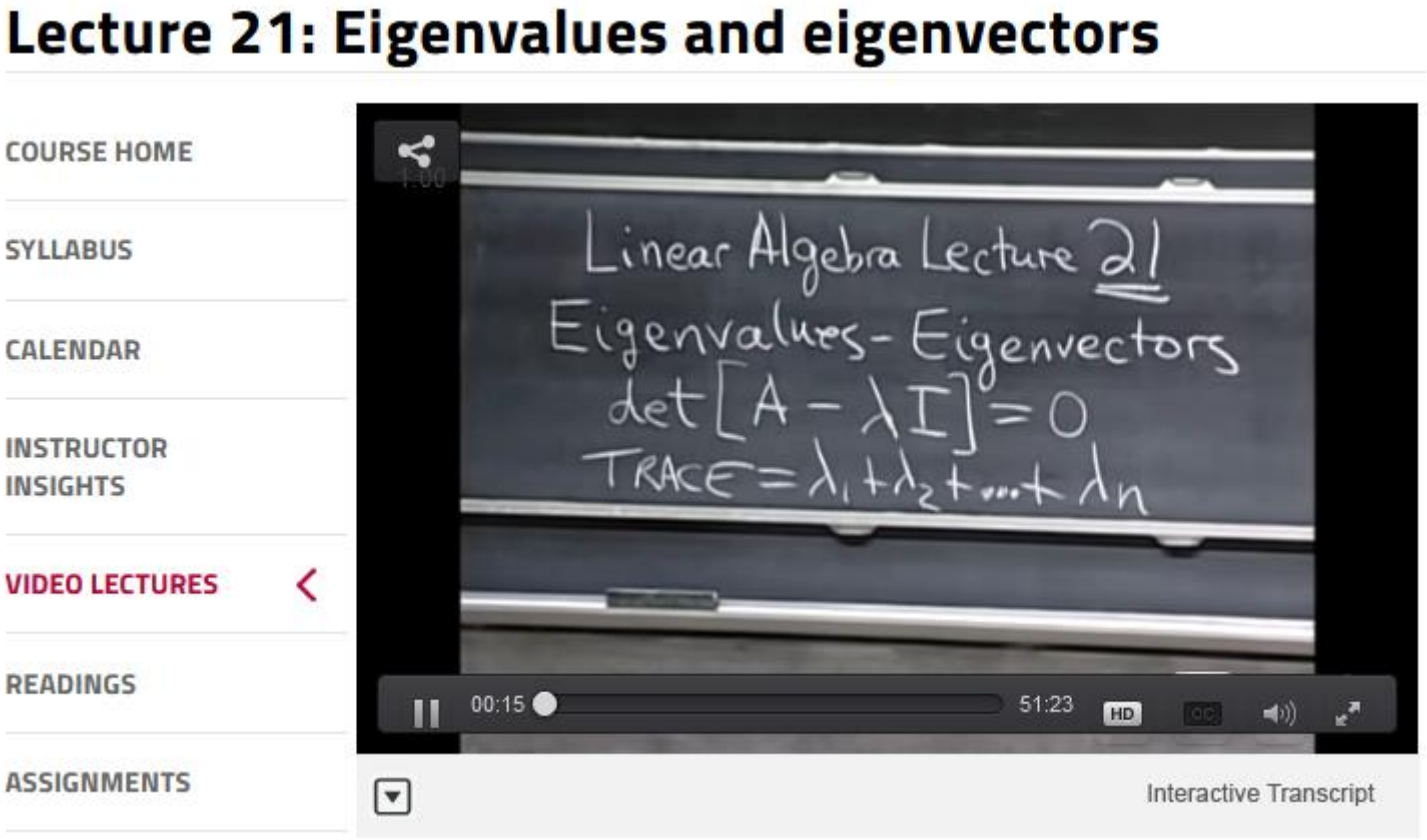
This is the equation of an ellipse.

Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by R



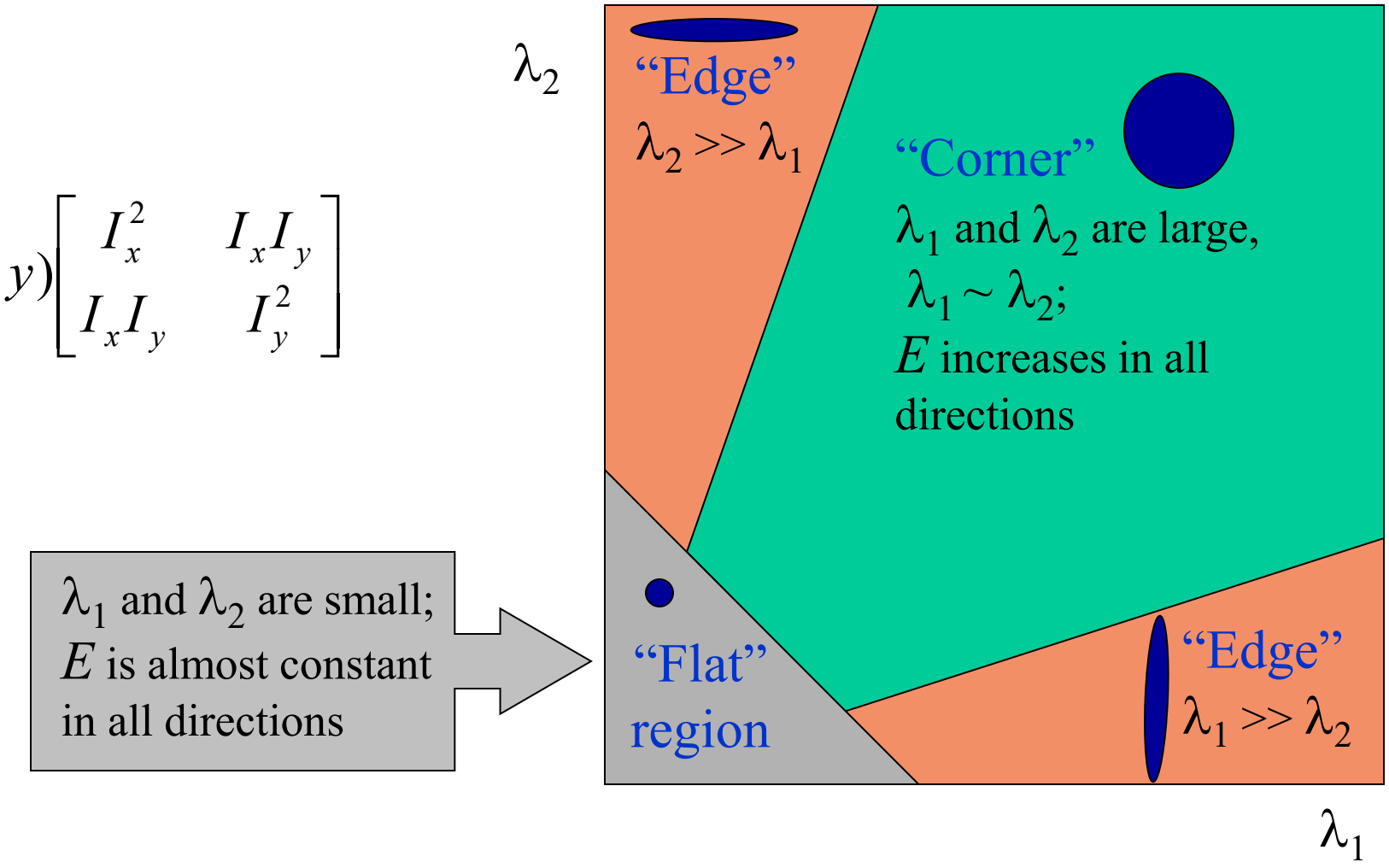
If you’re not comfortable with Eigenvalues and Eigenvectors, Gilbert Strang’s linear algebra lectures are linked from the course homepage



Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:

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

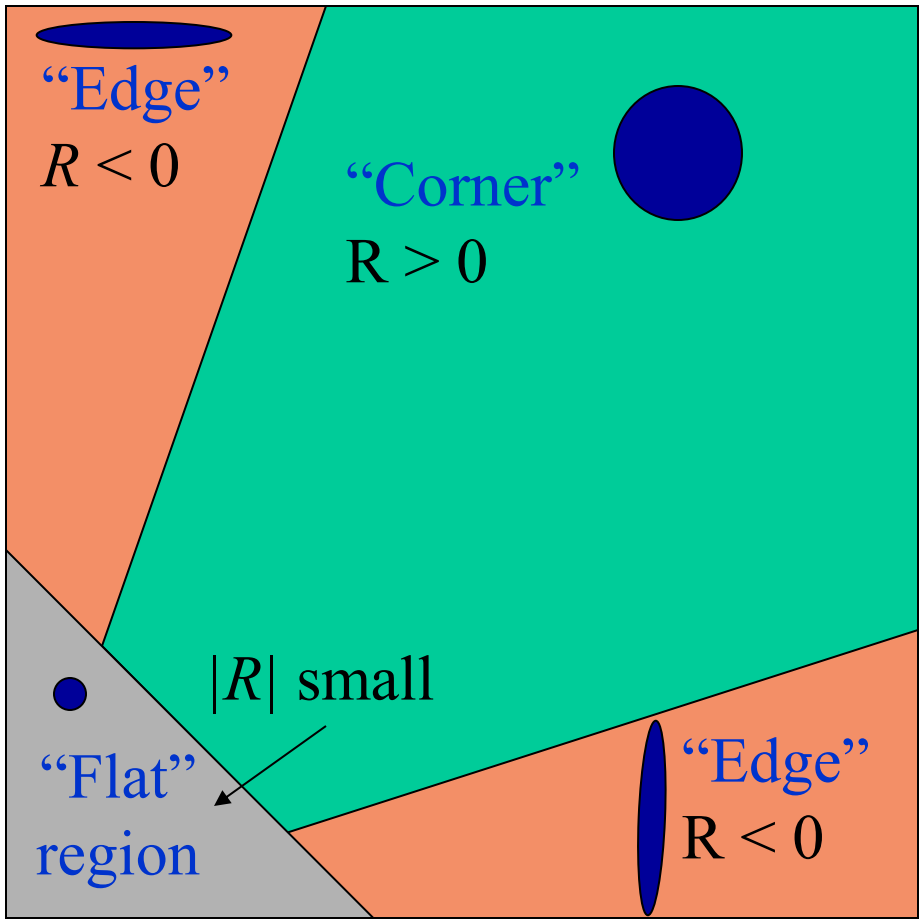


Corner response function

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

α: constant (0.04 to 0.06)

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



Harris corner detector

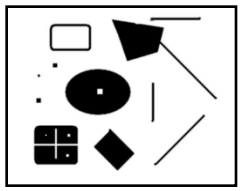
1) Compute *M* matrix for each image window to get their *cornerness* scores.
2) Find points whose surrounding window gave large corner response (*f*> threshold)
3) Take the points of local maxima, i.e., perform non-maximum suppression

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

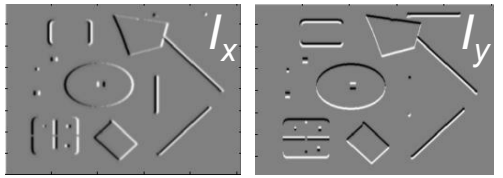
Harris Detector [Harris88]

- Second moment matrix

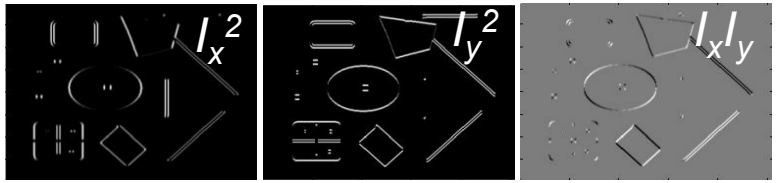
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



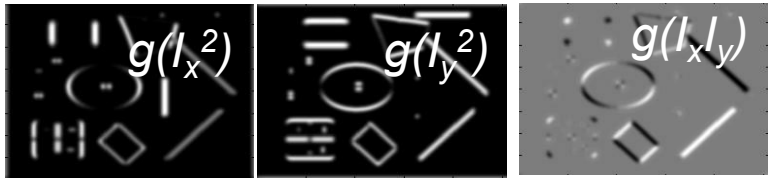
1. Image derivatives (optionally, blur first)



2. Square of derivatives



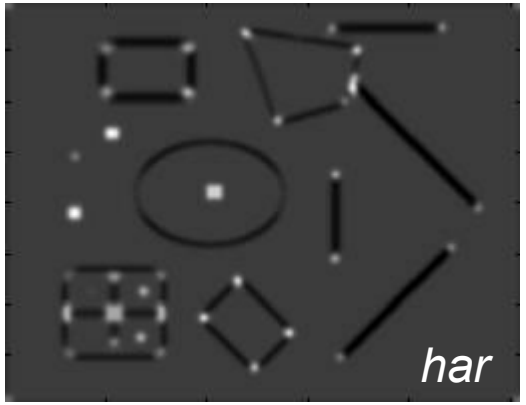
3. Gaussian filter $g(\sigma_l)$



$$\det M = \lambda_1 \lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

4. Cornerness function – both eigenvalues are strong

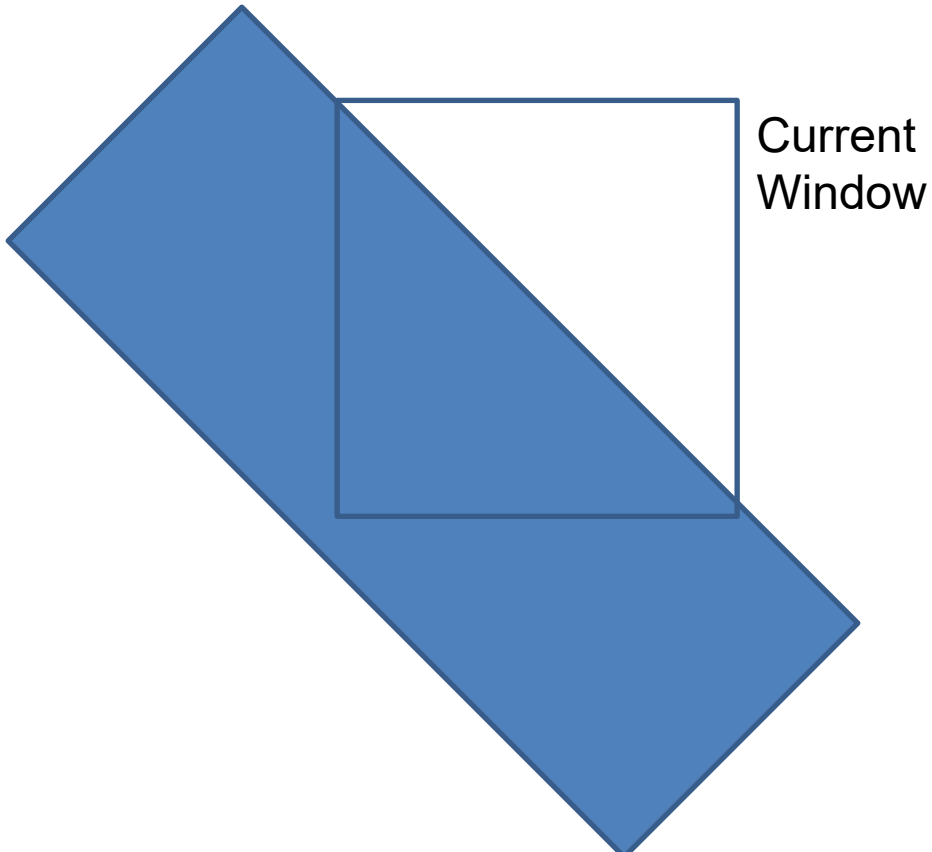
$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))^2] =$$
$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$



5. Non-maxima suppression

Harris Corners – Why so complicated?

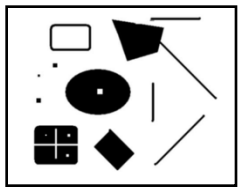
- Can’t we just check for regions with lots of gradients in the x and y directions?
  - No! A diagonal line would satisfy that criteria



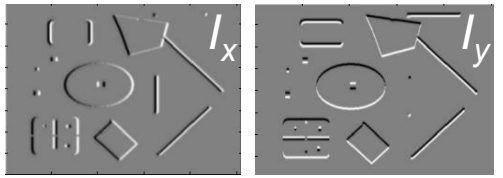
Harris Detector [Harris88]

- Second moment matrix

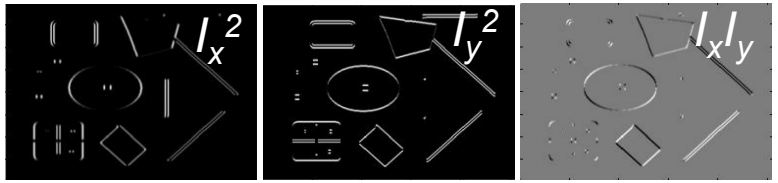
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



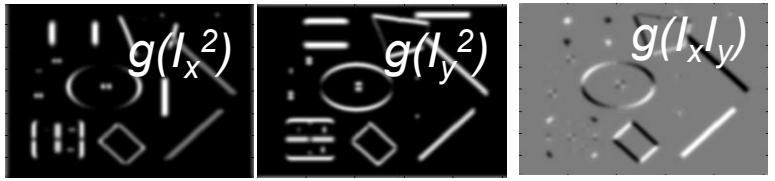
1. Image derivatives (optionally, blur first)



2. Square of derivatives



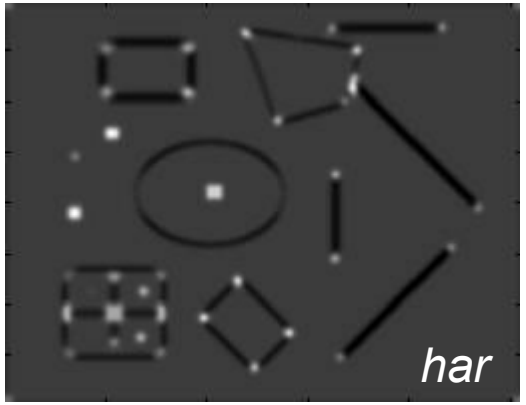
3. Gaussian filter $g(\sigma_l)$



$$\det M = \lambda_1 \lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

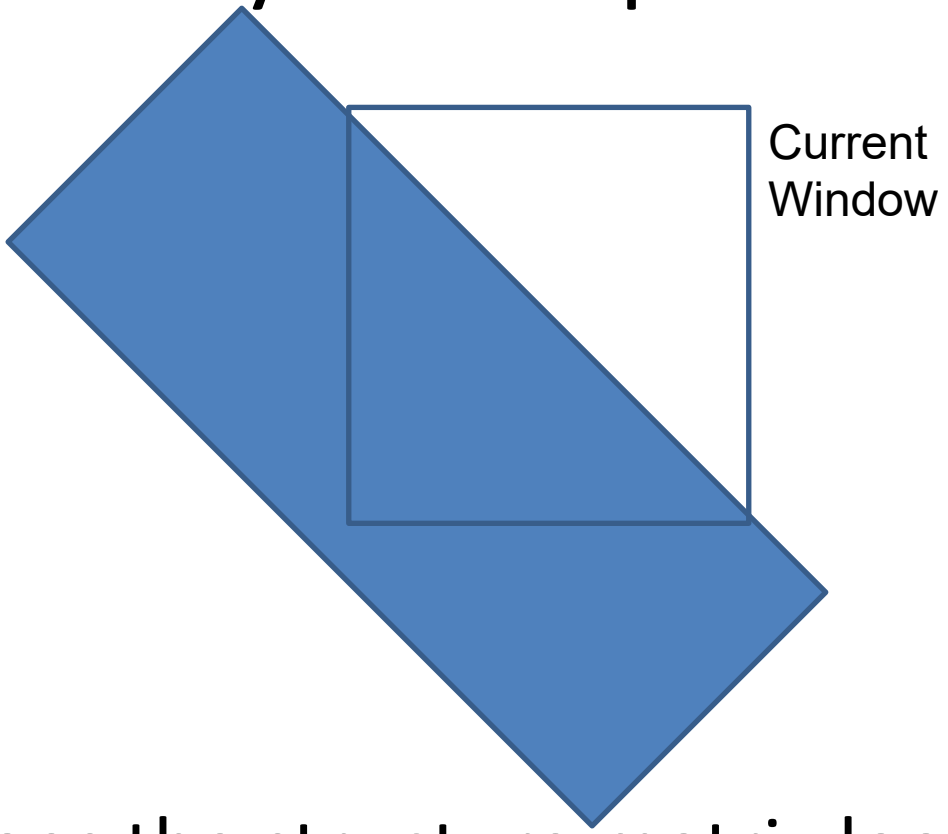
4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))^2] =$$
$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$



5. Non-maxima suppression

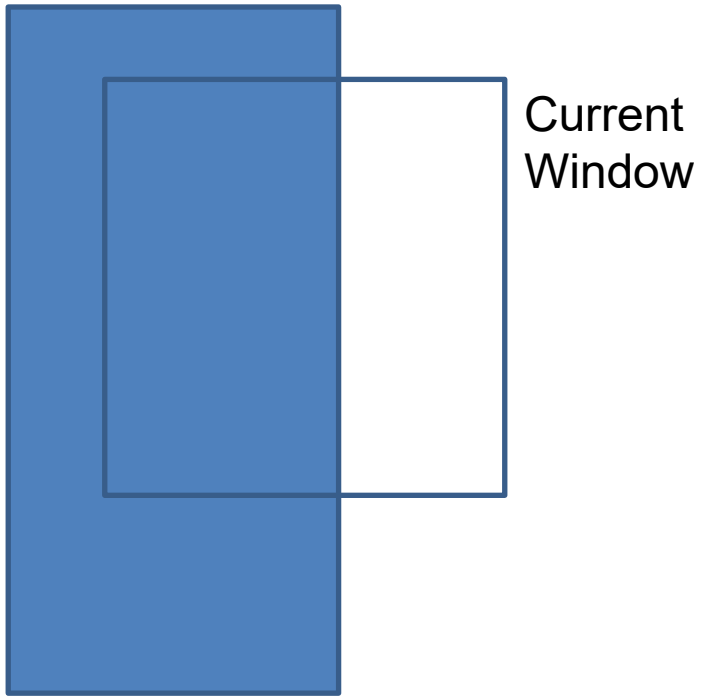
Harris Corners – Why so complicated?



- What does the structure matrix look here?

$$\begin{bmatrix} C & -C \\ -C & C \end{bmatrix}$$

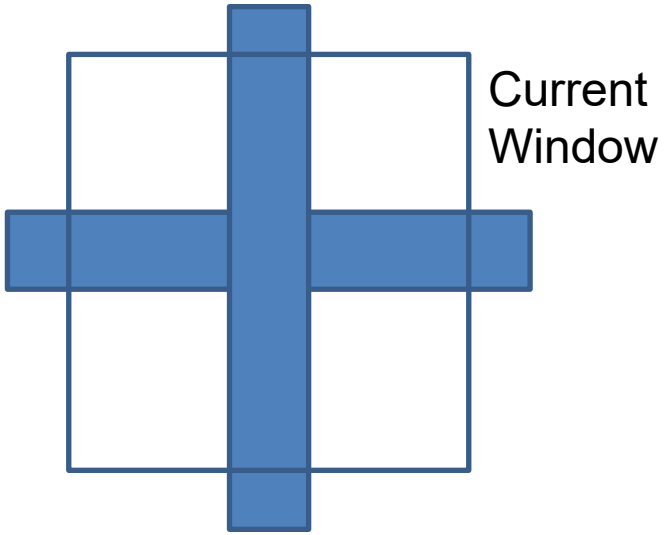
Harris Corners – Why so complicated?



- What does the structure matrix look here?

$$\begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix}$$

Harris Corners – Why so complicated?



- What does the structure matrix look here?

$$\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$$

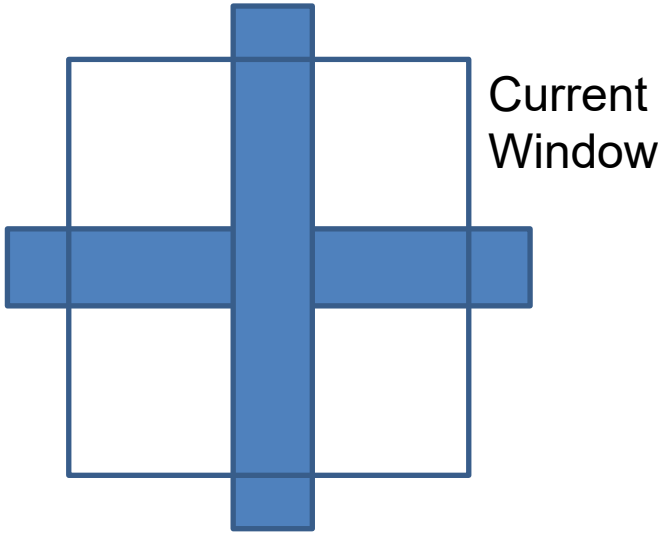
Harris Corners – Why so complicated?

We will measure x gradients by filtering with:

1	0	-1
2	0	-2
1	0	-1

We will measure y gradients by filtering with:

1	2	1
0	0	0
-1	-2	-1



- What does the structure matrix look here?

$$\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$$

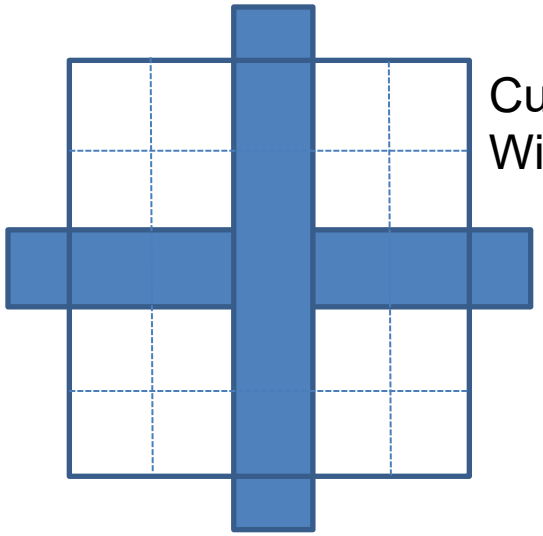
Harris Corners – Why so complicated?

We will measure x gradients by filtering with:

-1	0	1
-2	0	2
-1	0	1

We will measure y gradients by filtering with:

-1	-2	-1
0	0	0
1	2	1



Current Window

0	-4	0	4	0
0	-4	0	4	0
0	-2	0	2	0
0	-4	0	4	0
0	-4	0	4	0

$$I_x \Leftrightarrow \frac{\partial I}{\partial x}$$

0	0	0	0	0
-4	-4	-2	-4	-4
0	0	0	0	0
4	4	2	4	4
0	0	0	0	0

$$I_y \Leftrightarrow \frac{\partial I}{\partial y}$$

$$I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

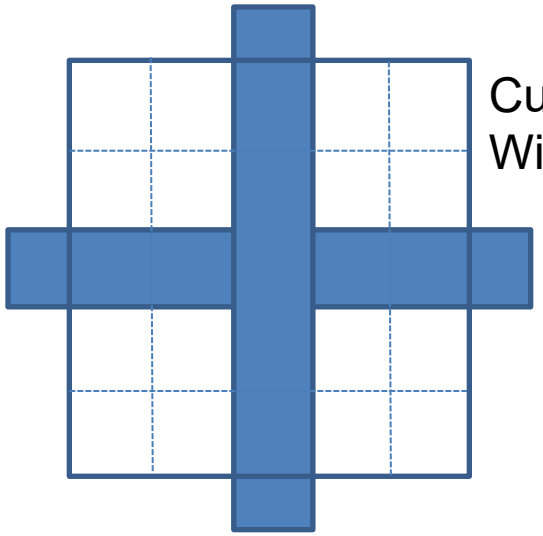
Harris Corners – Why so complicated?

We will measure x gradients by filtering with:

-1	0	1
-2	0	2
-1	0	1

We will measure y gradients by filtering with:

-1	-2	-1
0	0	0
1	2	1



Current Window

0	-4	0	4	0
0	-4	0	4	0
0	-2	0	2	0
0	-4	0	4	0
0	-4	0	4	0

$$I_x \Leftrightarrow \frac{\partial I}{\partial x}$$

0	0	0	0	0
-4	-4	-2	-4	-4
0	0	0	0	0
4	4	2	4	4
0	0	0	0	0

$$I_y \Leftrightarrow \frac{\partial I}{\partial y}$$

0	0	0	0	0
0	16	0	-16	0
0	0	0	0	0
0	-16	0	16	0
0	0	0	0	0

$$I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

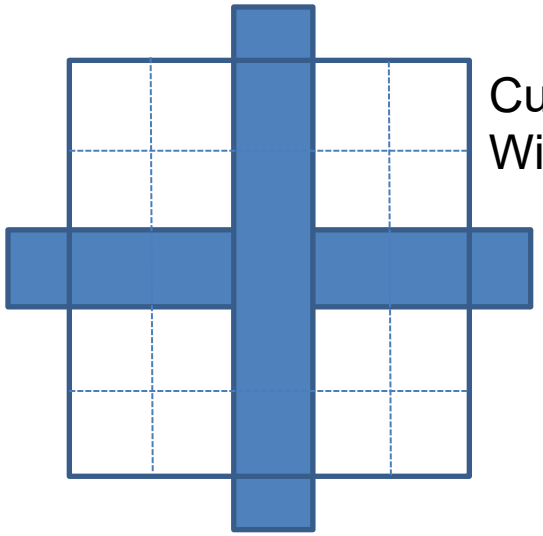
Harris Corners – Why so complicated?

We will measure x gradients by filtering with:

-1	0	1
-2	0	2
-1	0	1

We will measure y gradients by filtering with:

-1	-2	-1
0	0	0
1	2	1



Current Window

0	-4	0	4	0
0	-4	0	4	0
0	-2	0	2	0
0	-4	0	4	0
0	-4	0	4	0

$$I_x \Leftrightarrow \frac{\partial I}{\partial x}$$

0	0	0	0	0
-4	-4	-2	-4	-4
0	0	0	0	0
4	4	2	4	4
0	0	0	0	0

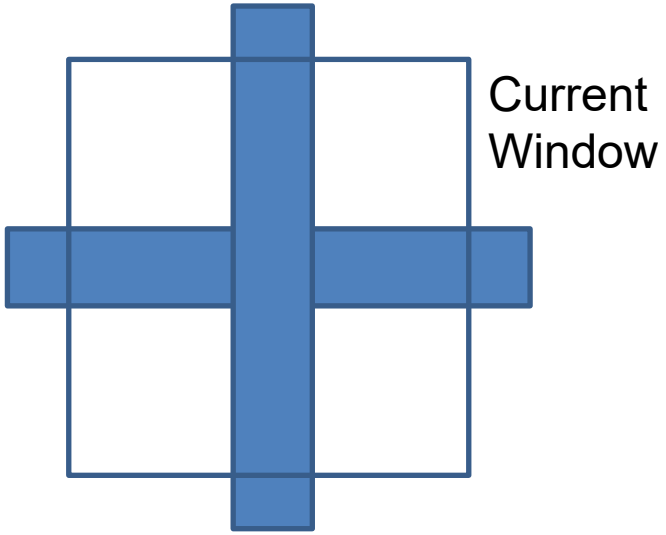
$$I_y \Leftrightarrow \frac{\partial I}{\partial y}$$

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

0	0	0	0	0
0	16	0	-16	0
0	0	0	0	0
0	-16	0	16	0
0	0	0	0	0

$$I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

Harris Corners – Why so complicated?



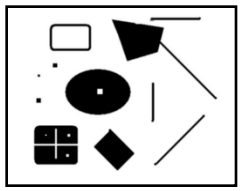
- What does the structure matrix look here?

$$\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$$

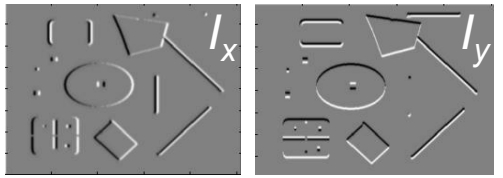
Harris Detector [Harris88]

- Second moment matrix

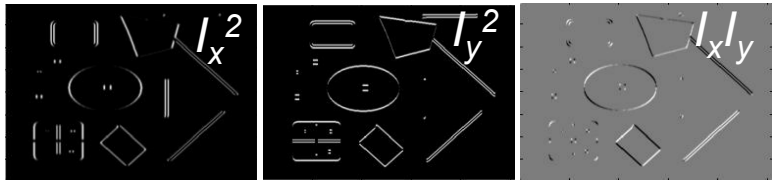
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



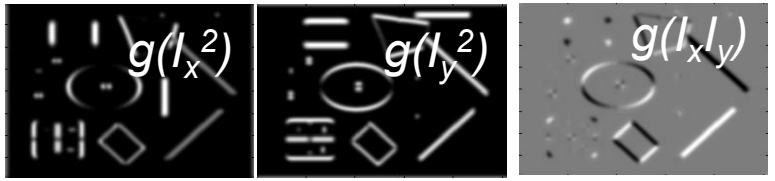
1. Image derivatives (optionally, blur first)



2. Square of derivatives



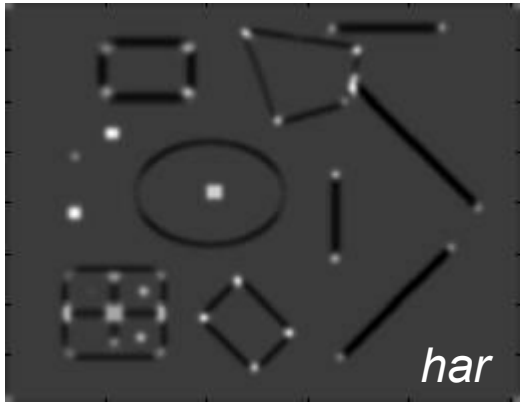
3. Gaussian filter $g(\sigma_l)$



$$\det M = \lambda_1 \lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

4. Cornerness function – both eigenvalues are strong

$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))^2] =$$
$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

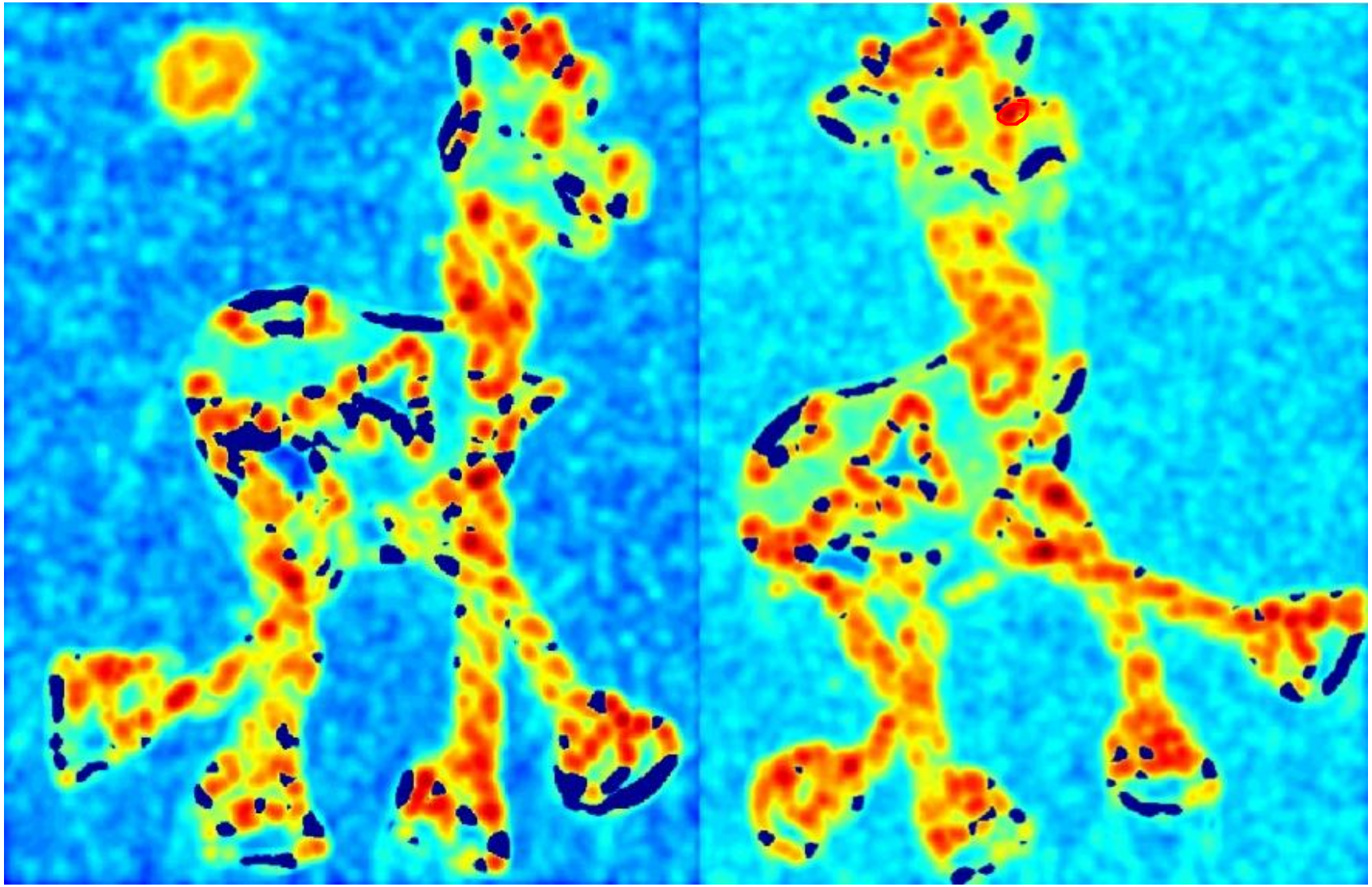


5. Non-maxima suppression

Harris Detector: Steps

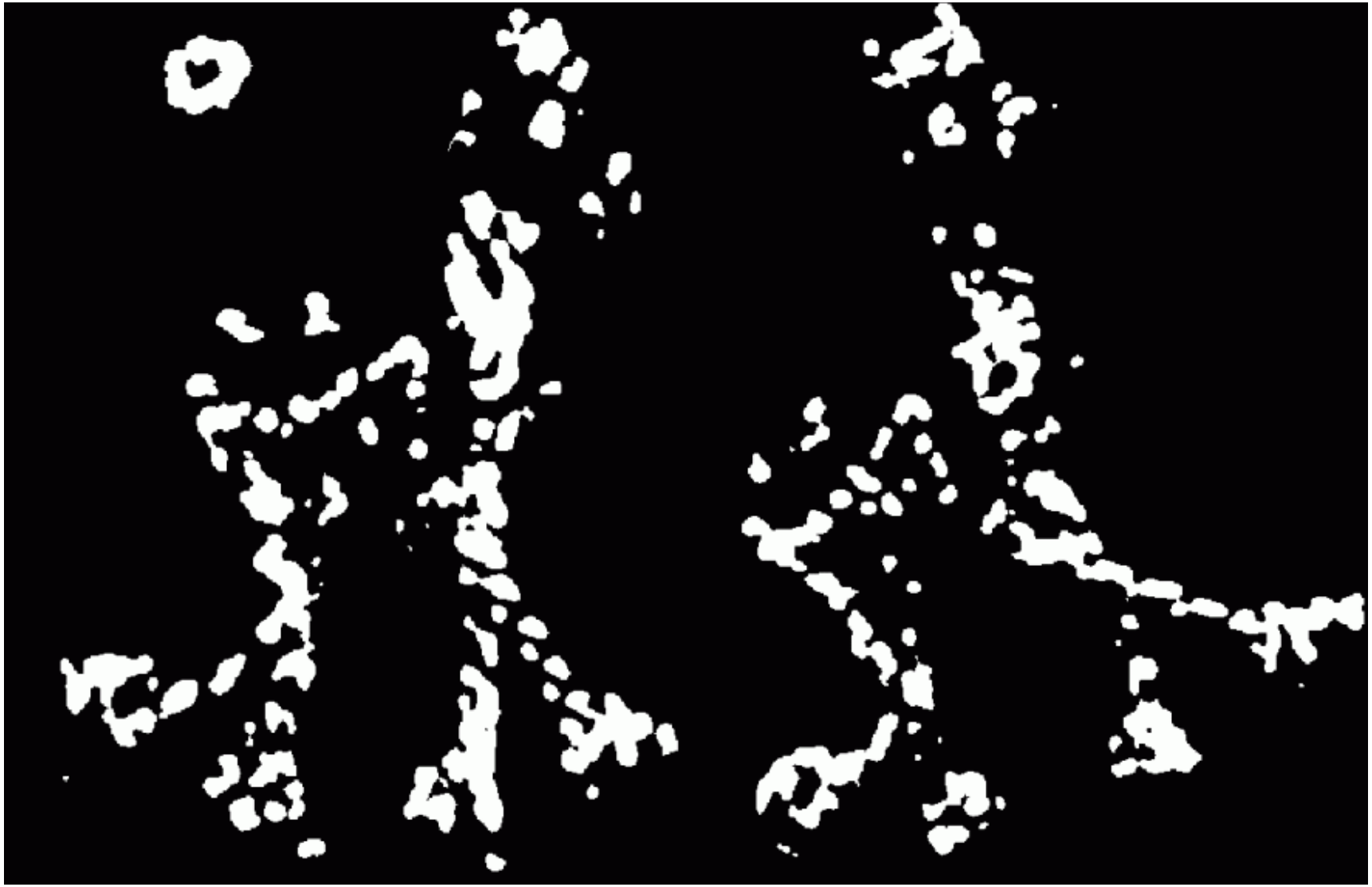
Harris Detector: Steps

Compute corner response R



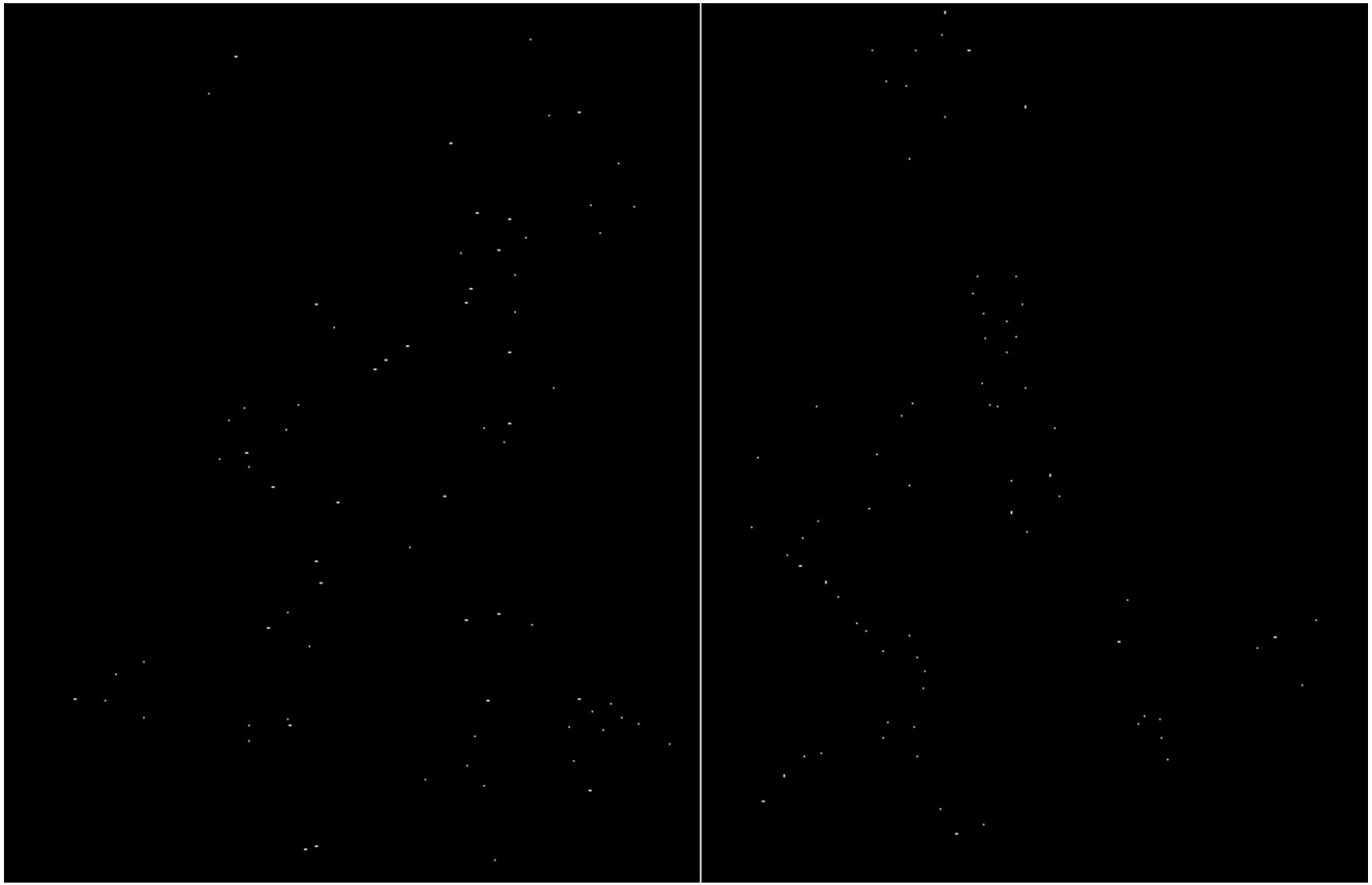
Harris Detector: Steps

Find points with large corner response: R>threshold



Harris Detector: Steps

Take only the points of local maxima of R



Harris Detector: Steps