Université Concordia University
FACULTY OF ENGINEERING AND COMPUTER SCIENCE
Department of Computer Science and Software Engineering

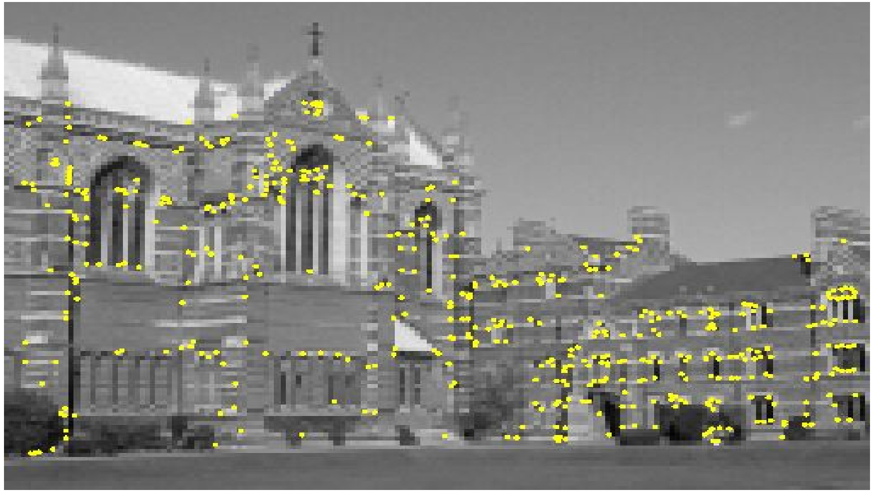
COMP498G/691G COMPUTER VISION

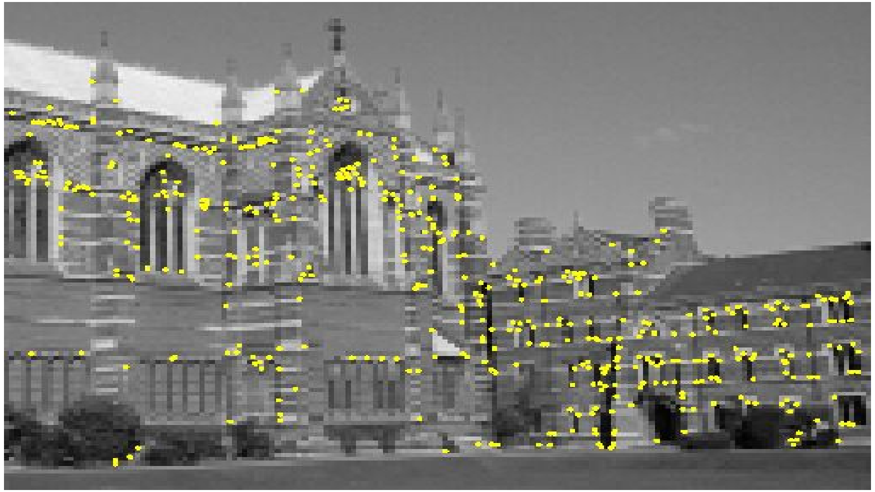
LECTURE 6
INTEREST OPERATORS

Today’s Lecture

- Interest Operator
  - Slides acknowledgment: L. Shapiro
- Questions

Preview: Harris detector

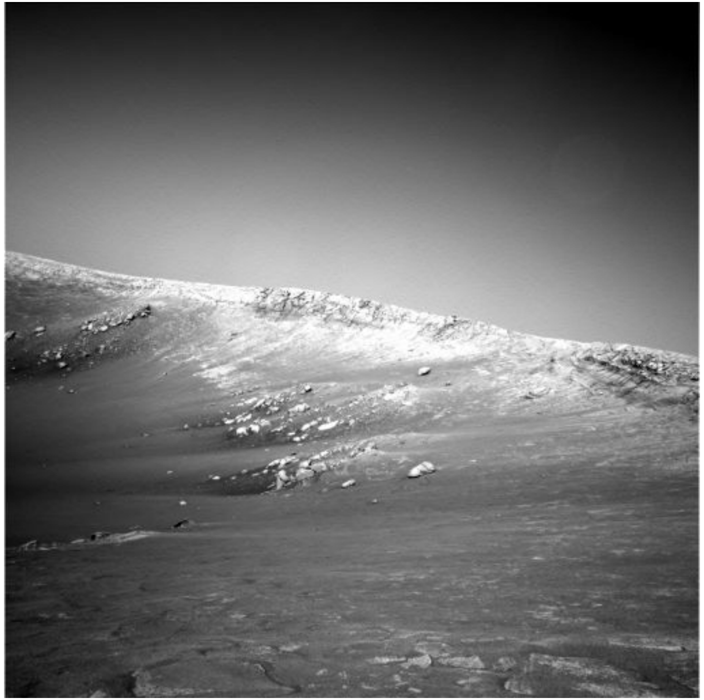




Interest points extracted with Harris (~ 500 points)

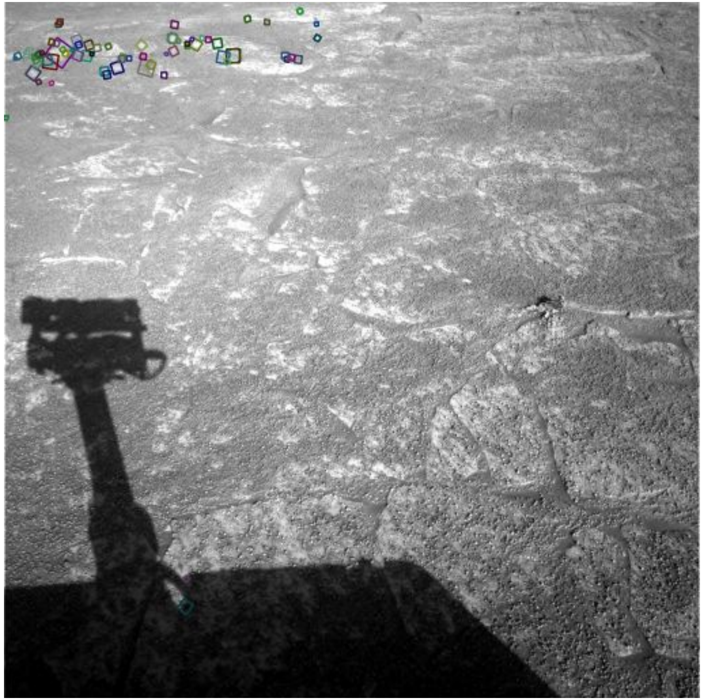
How can we find corresponding points?

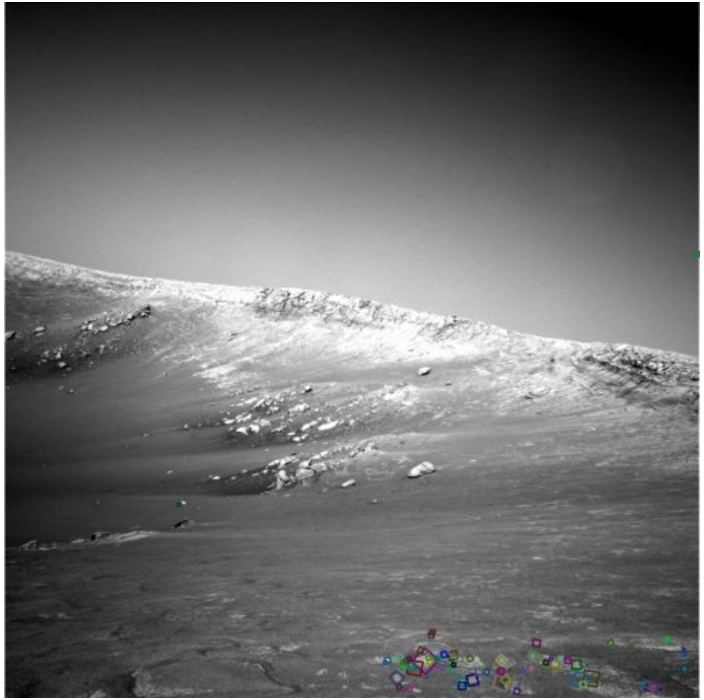
Not always easy



NASA Mars Rover images

Answer below (look for tiny colored squares…)



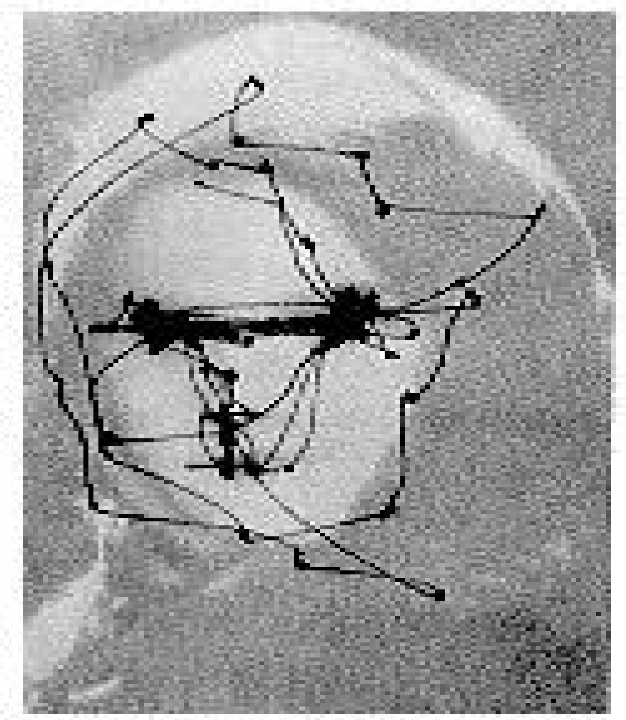


NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely

Human eye movements

What catches your interest?

Human eye movements



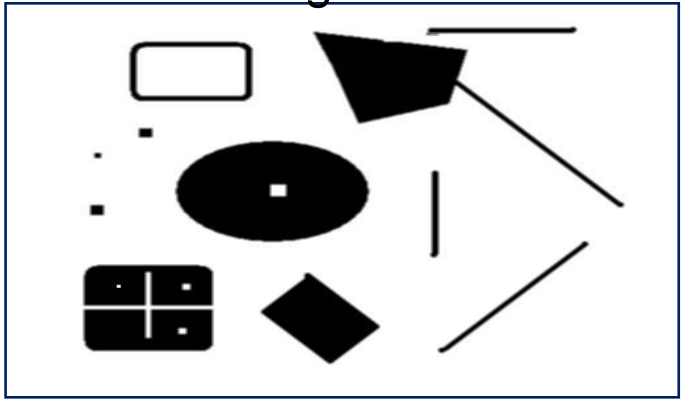
Yarbus eye tracking

What catches your interest?

Interest points

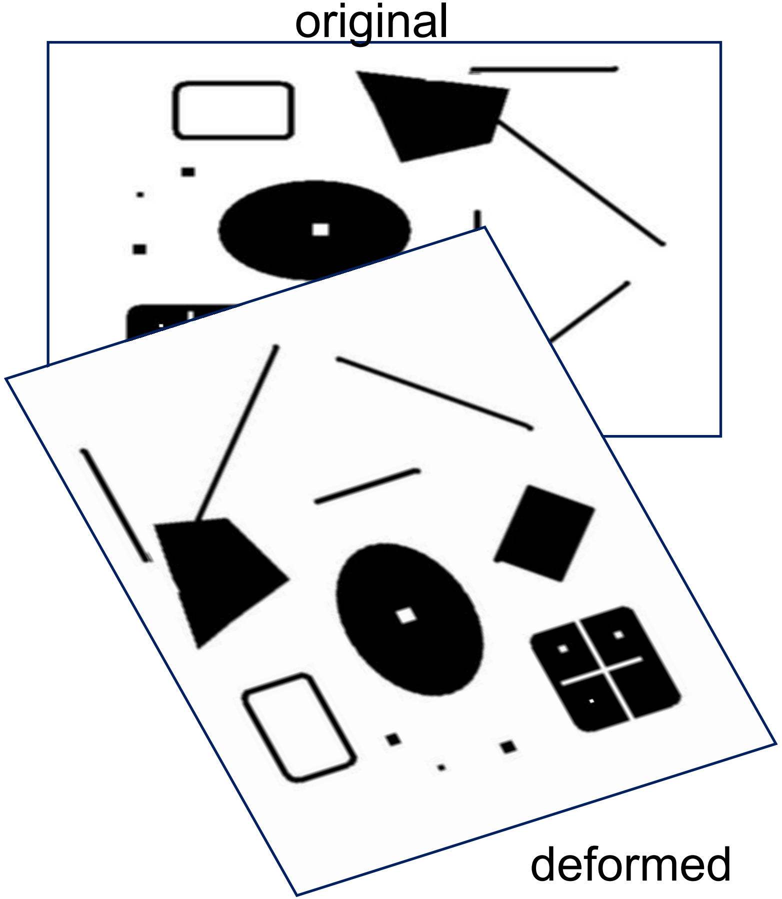
- Suppose you have to click on some points, go away and come back after I deform the image, and click on the same points again.
  - Which points would you choose?

original



Interest points

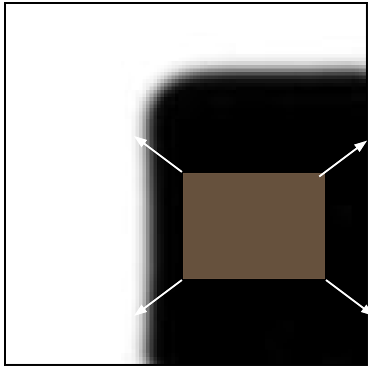
- Suppose you have to click on some points, go away and come back after I deform the image, and click on the same points again.
  - Which points would you choose?



Intuition

Corners

- 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

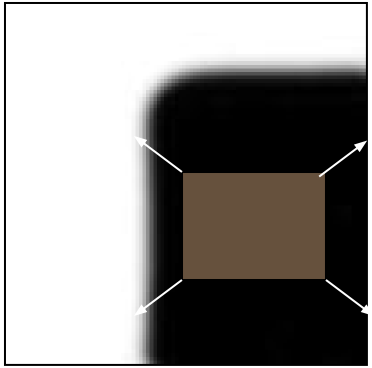


"flat" region: no change in all directions

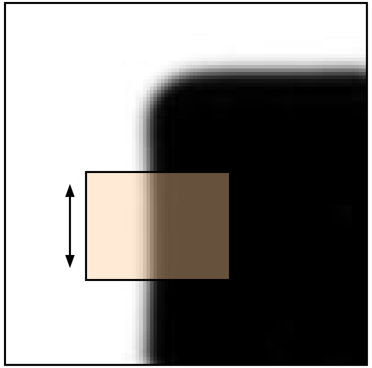
Source: A. Efros

Corners

- 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

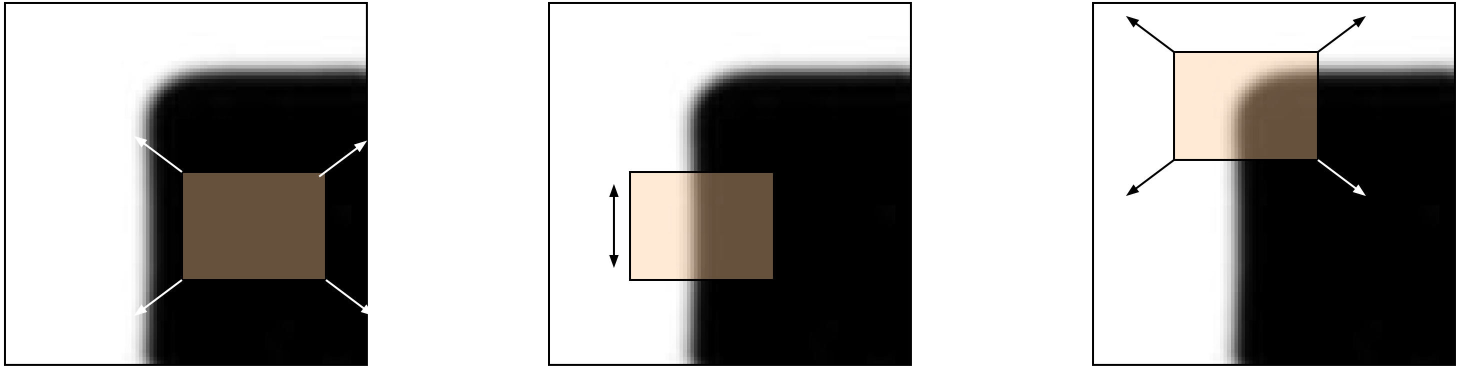


"edge": no change along the edge direction

Source: A. Efros

Corners

- 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



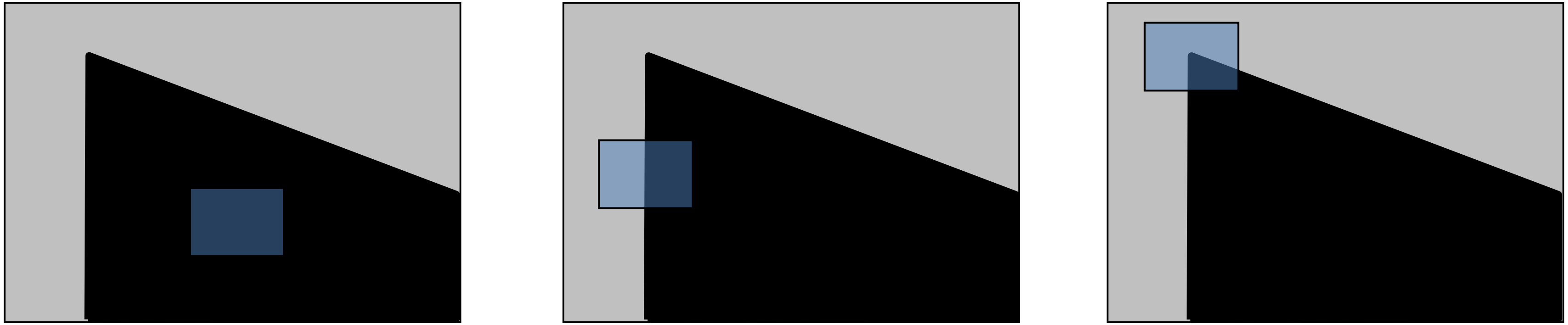
"flat" region: no change in all directions

"edge": no change along the edge direction

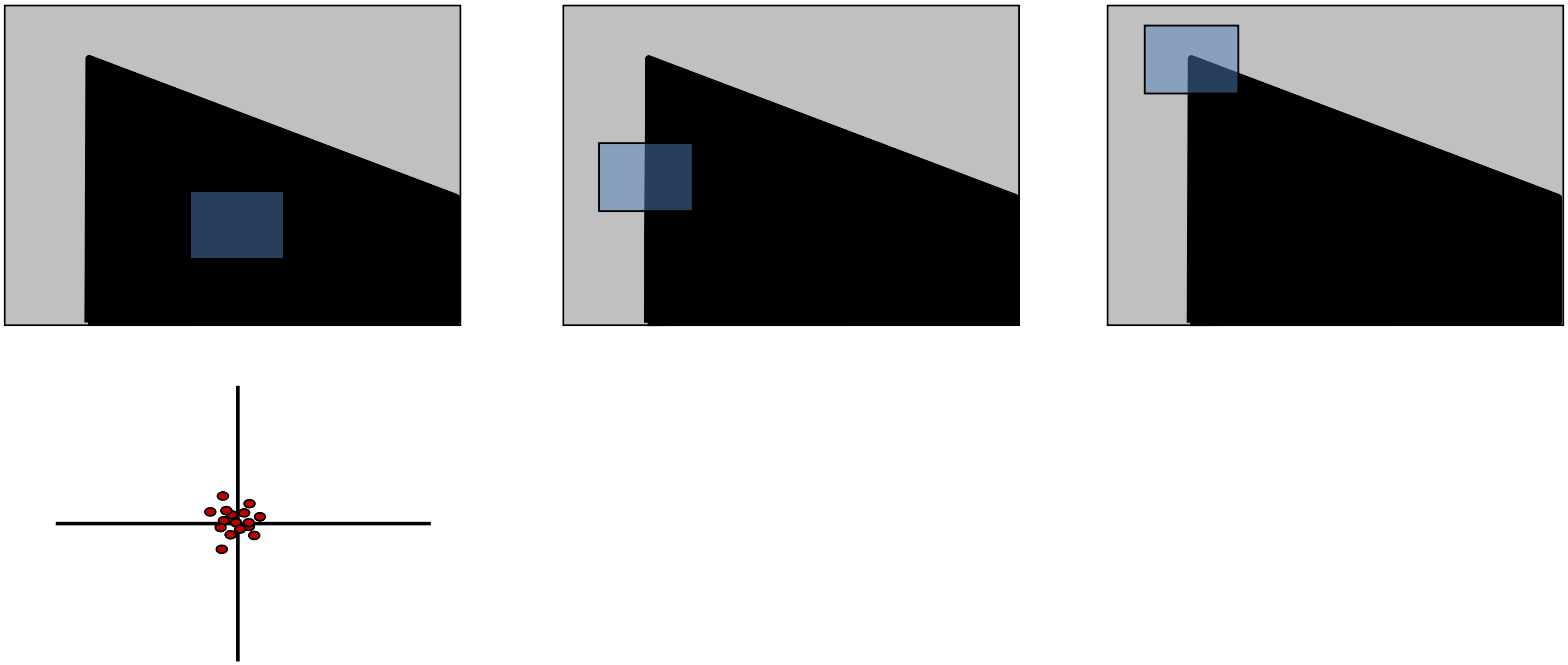
"corner": significant change in all directions

Source: A. Efros

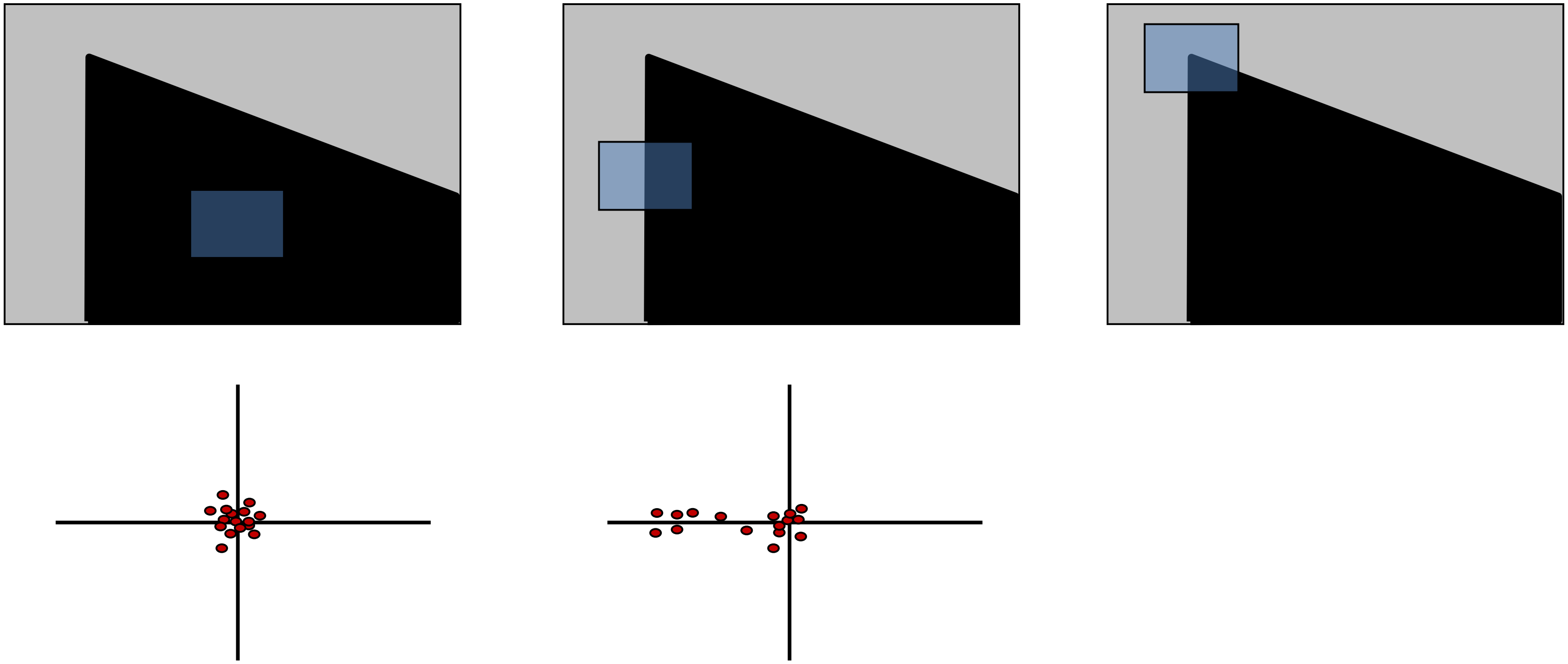
Let’s look at the gradient distributions



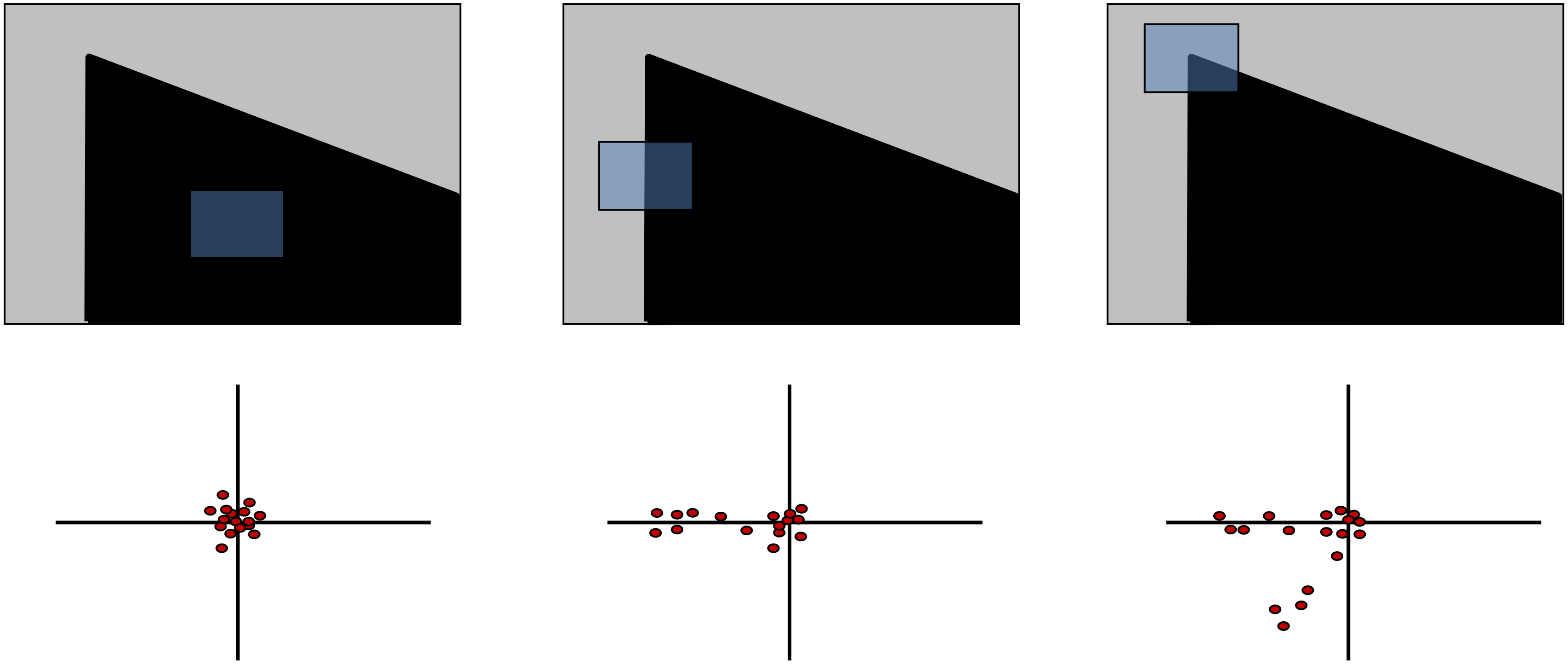
Let’s look at the gradient distributions



Let's look at the gradient distributions

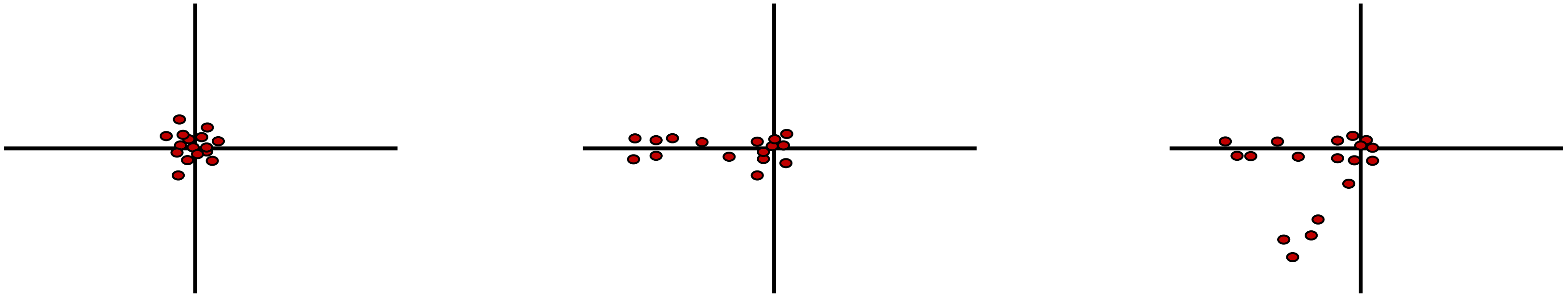


Let's look at the gradient distributions



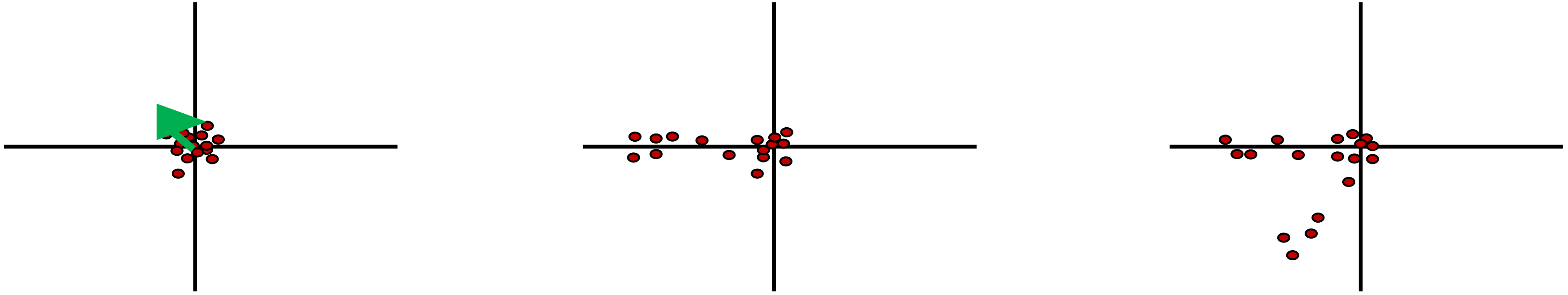
Principle Component Analysis

Principal component is the direction of highest variance.



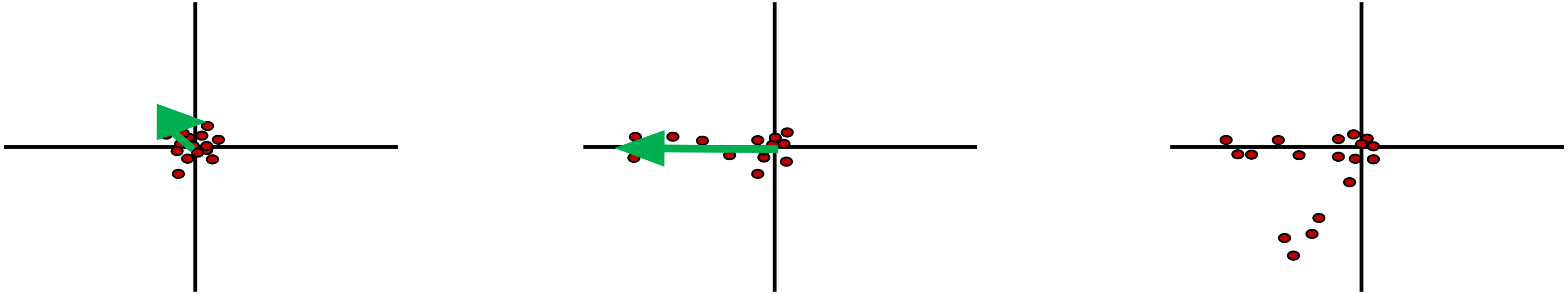
Principle Component Analysis

Principal component is the direction of highest variance.



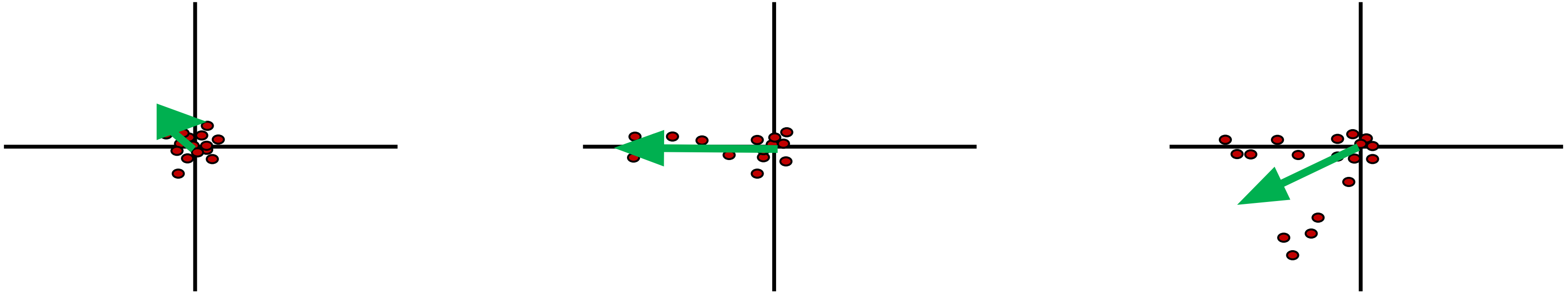
Principle Component Analysis

Principal component is the direction of highest variance.



Principle Component Analysis

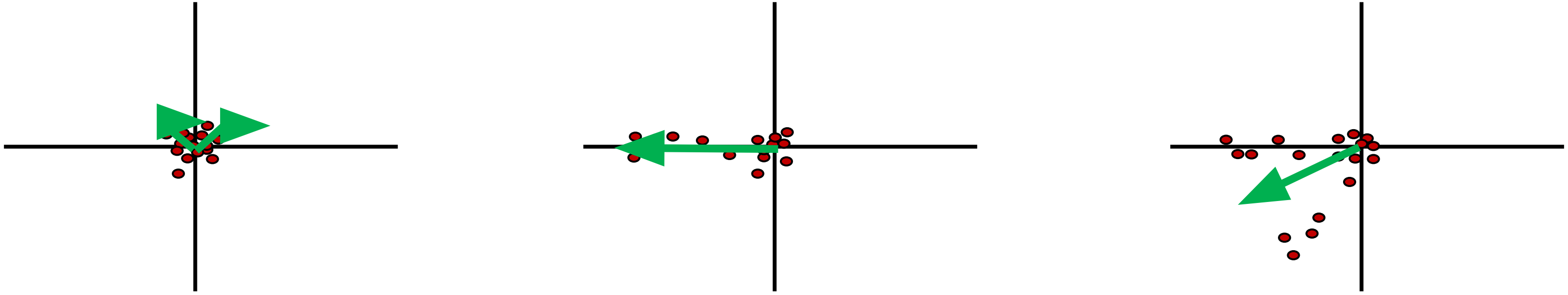
Principal component is the direction of highest variance.



Principle Component Analysis

Principal component is the direction of highest variance.

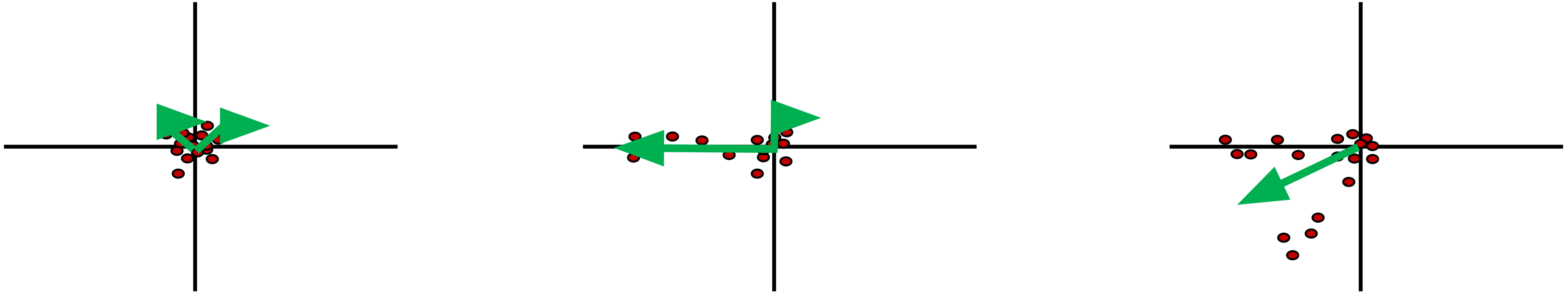
Next, highest component is the direction with highest variance *orthogonal* to the previous components.



Principle Component Analysis

Principal component is the direction of highest variance.

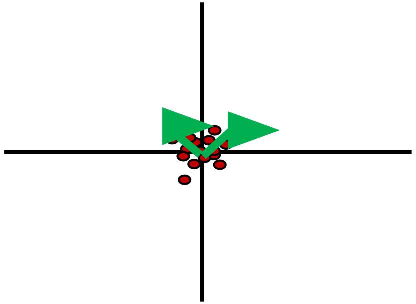
Next, highest component is the direction with highest variance *orthogonal* to the previous components.

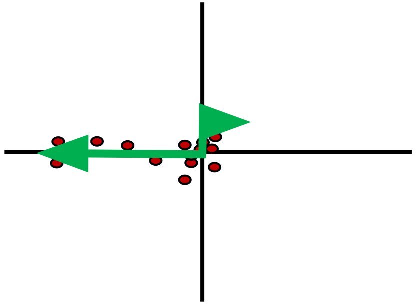


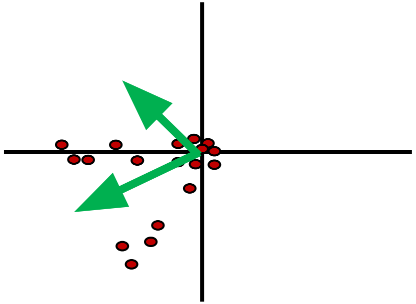
Principle Component Analysis

Principal component is the direction of highest variance.

Next, highest component is the direction with highest variance *orthogonal* to the previous components.







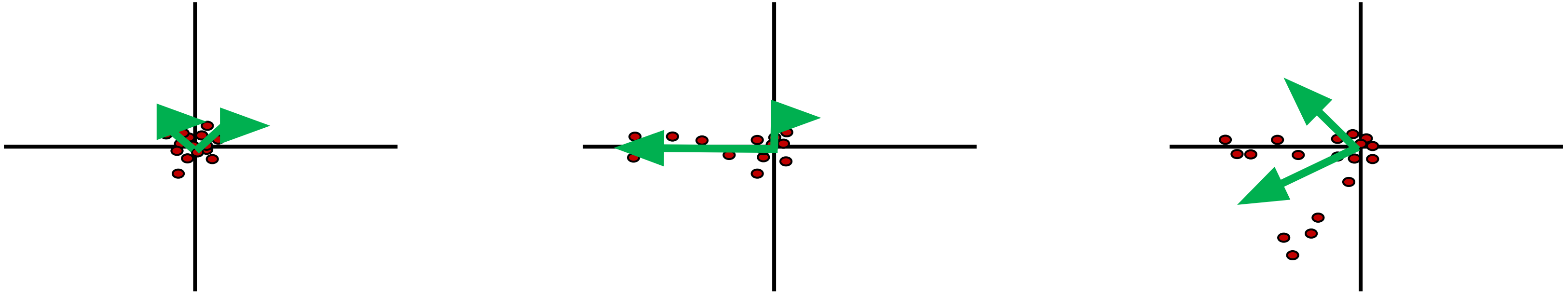
Principle Component Analysis

Principal component is the direction of highest variance.

Next, highest component is the direction with highest variance *orthogonal* to the previous components.

How to compute PCA components:

1. Subtract off the mean for each data point.
2. Compute the covariance matrix.
3. Compute eigenvectors and eigenvalues.
4. The components are the eigenvectors ranked by the eigenvalues.



<u>Covariance matrix:</u>

$$\operatorname{cov}(X,Y) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)\cdot(y_i - y_j) = \frac{1}{n^2}\sum_{i}\sum_{j>i}(x_i - x_j)\cdot(y_i - y_j).$$

Principle Component Analysis

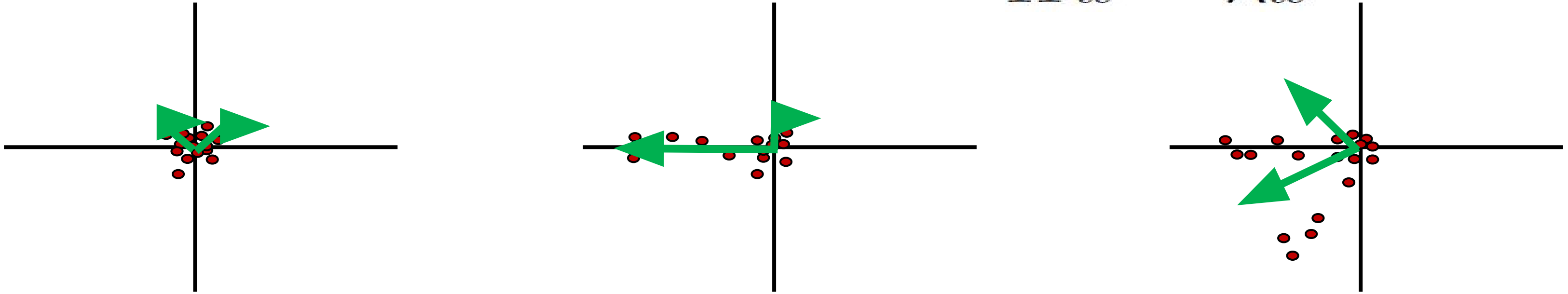
Principal component is the direction of highest variance.

Next, highest component is the direction with highest variance *orthogonal* to the previous components.

How to compute PCA components:

1. Subtract off the mean for each data point.
2. Compute the covariance matrix.
3. Compute eigenvectors and eigenvalues.
4. The components are the eigenvectors ranked by the eigenvalues.

$$Hx = \lambda x$$



Covariance matrix:

$$\mathrm{cov}(X,Y) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)\cdot(y_i - y_j) = \frac{1}{n^2}\sum_{i}\sum_{j>i}(x_i - x_j)\cdot(y_i - y_j).$$

Quick eigenvalue/eigenvector review

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

$$Ax = \lambda x$$

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

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

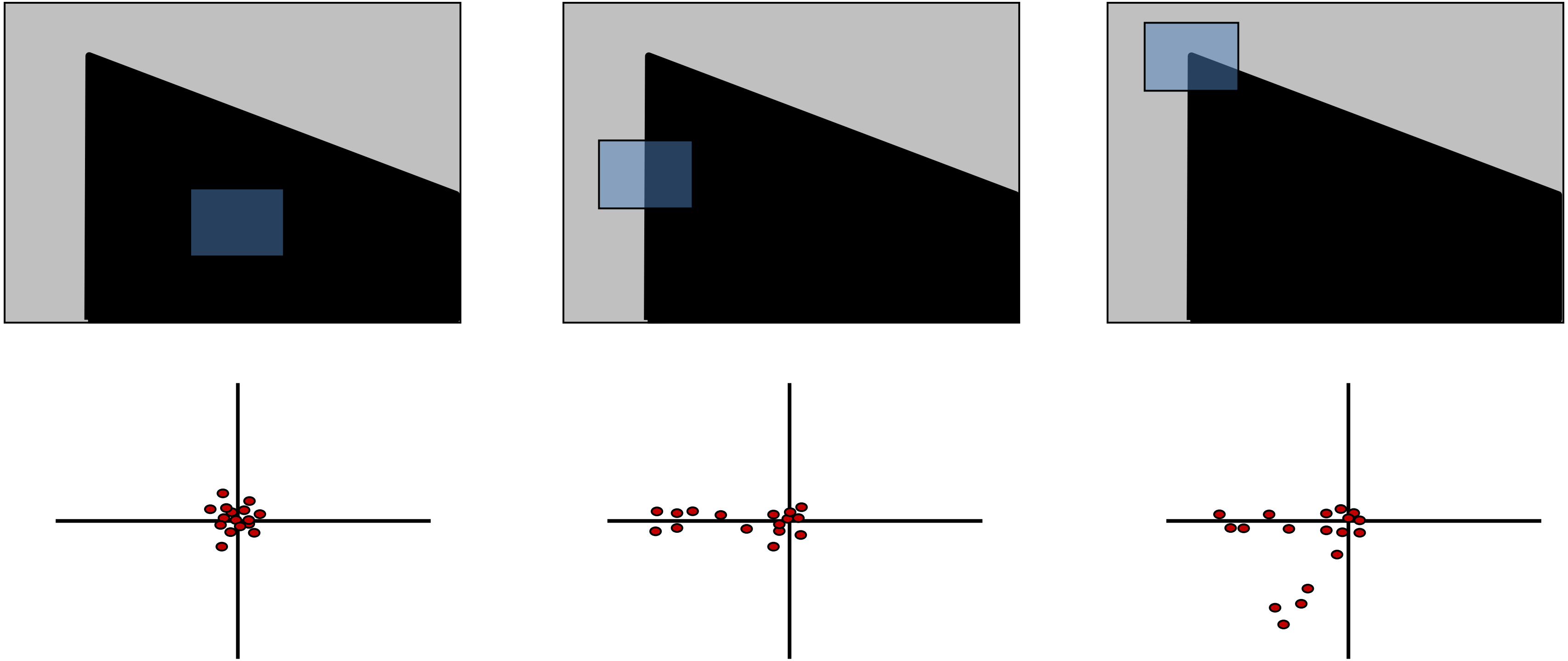
- The solution:

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

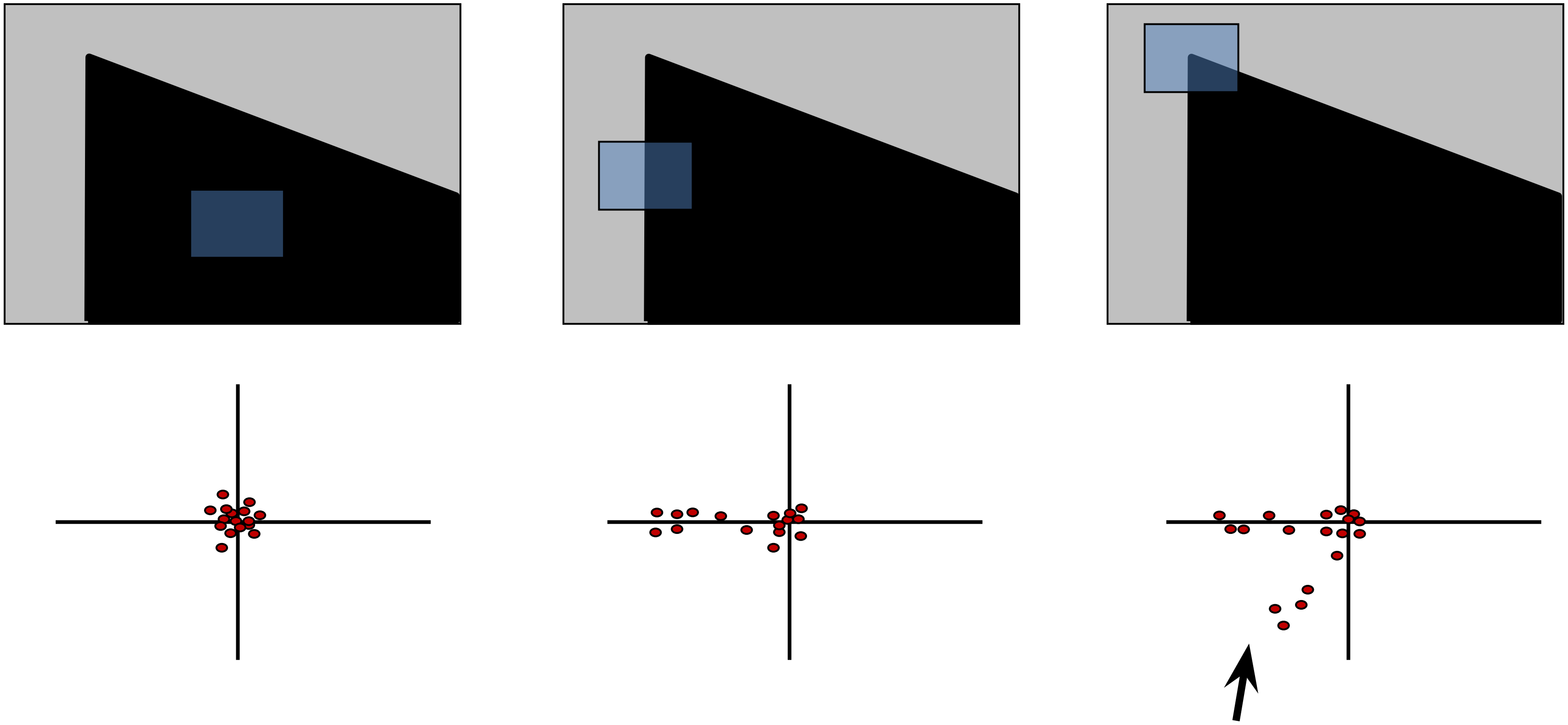
Once you know λ, you find x by solving

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

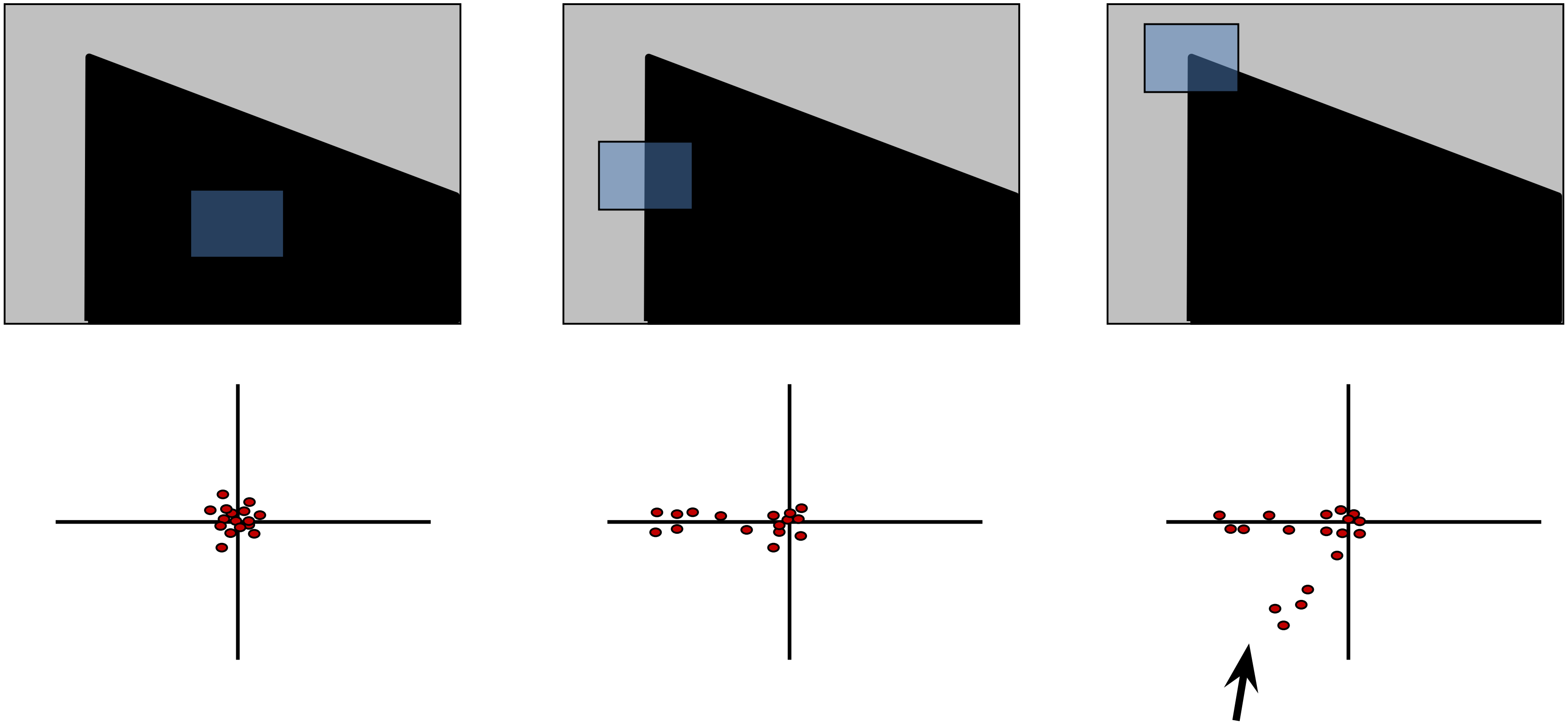
Corners have …



Corners have …



Corners have …



Both eigenvalues are large!

Second Moment Matrix or Harris Matrix

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

2 x 2 matrix of image derivatives smoothed by Gaussian weights.

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

First compute I_x, I_y, and I_xI_y as 3 images; then apply Gaussian to each.

The math

To compute the eigenvalues:

1. Compute the Harris matrix over a window.

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

Typically Gaussian weights

$$I_x = \frac{\partial f}{\partial x}, I_y = \frac{\partial f}{\partial y}$$

2. Compute eigenvalues from that.

$$H = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \lambda_{\pm} = \frac{1}{2}\left((a+d) \pm \sqrt{4bc + (a-d)^2}\right)$$

Corner Response Function

- Computing eigenvalues are expensive

Corner Response Function

- Computing eigenvalues are expensive
- Harris corner detector used the following alternative

$$R = det(M) - \alpha \cdot trace(M)^2$$

Reminder:

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

Harris detector: Steps

1. Compute derivatives I_x^2 , I_y^2 and $I_x I_y$ at each pixel and smooth them with a Gaussian.

C.Harris and M.Stephens. *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris detector: Steps

1. Compute derivatives I_x^2 , I_y^2 and $I_x I_y$ at each pixel and smooth them with a Gaussian.
2. Compute the Harris matrix H in a window around each pixel

C.Harris and M.Stephens. *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris detector: Steps

1. Compute derivatives I_x^2 , I_y^2 and $I_x I_y$ at each pixel and smooth them with a Gaussian.
2. Compute the Harris matrix H in a window around each pixel
3. Compute corner response function *R*

C.Harris and M.Stephens. *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris detector: Steps

1. Compute derivatives I_x^2 , I_y^2 and $I_x I_y$ at each pixel and smooth them with a Gaussian.
2. Compute the Harris matrix H in a window around each pixel
3. Compute corner response function *R*
4. Threshold *R*

C.Harris and M.Stephens. *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris detector: Steps

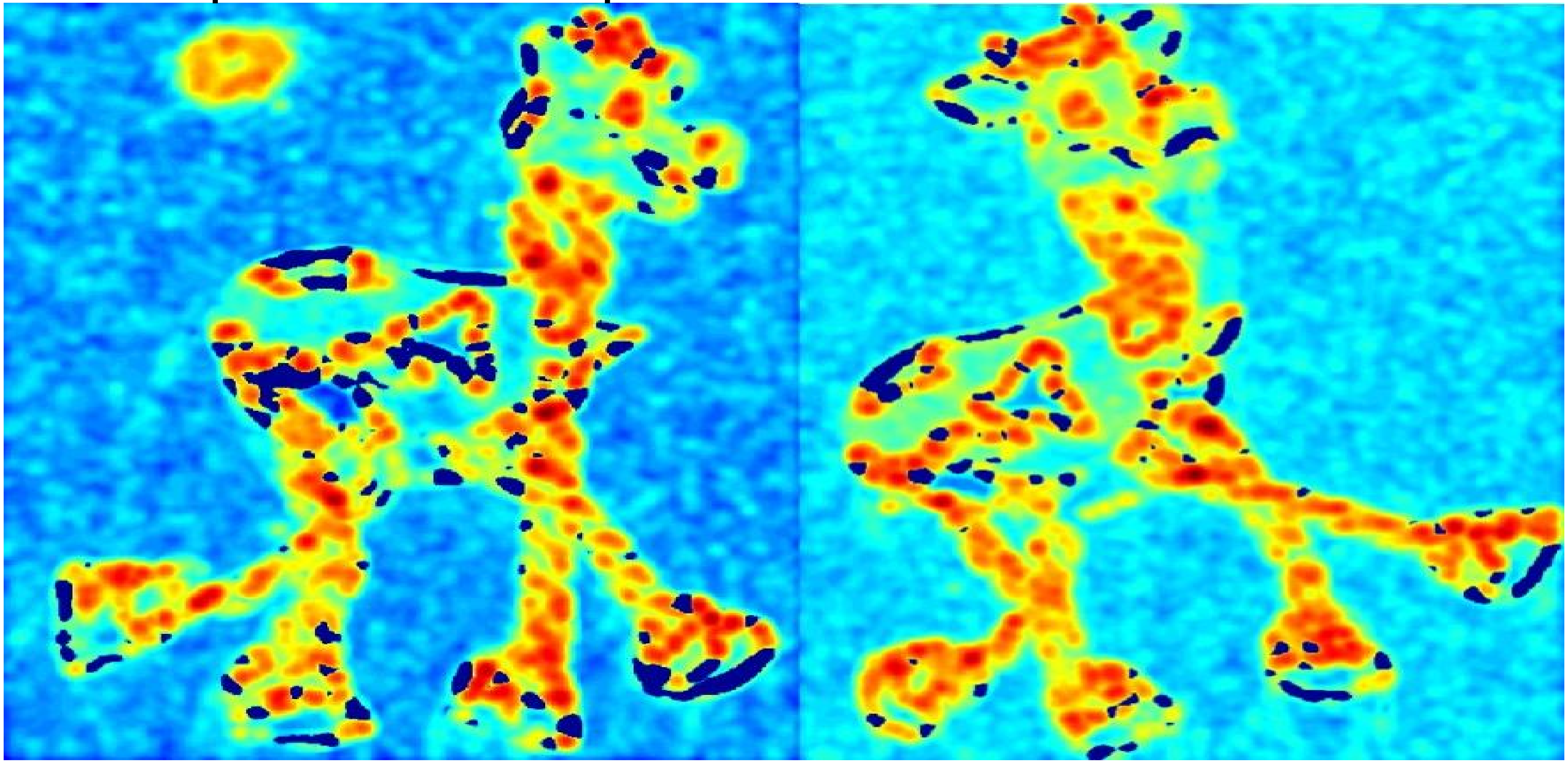
1. Compute derivatives I_x^2 , I_y^2 and I_xI_y at each pixel and smooth them with a Gaussian.
2. Compute the Harris matrix H in a window around each pixel
3. Compute corner response function *R*
4. Threshold *R*
5. Find local maxima of response function (non-maximum suppression)

C.Harris and M.Stephens. *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris Detector: Steps

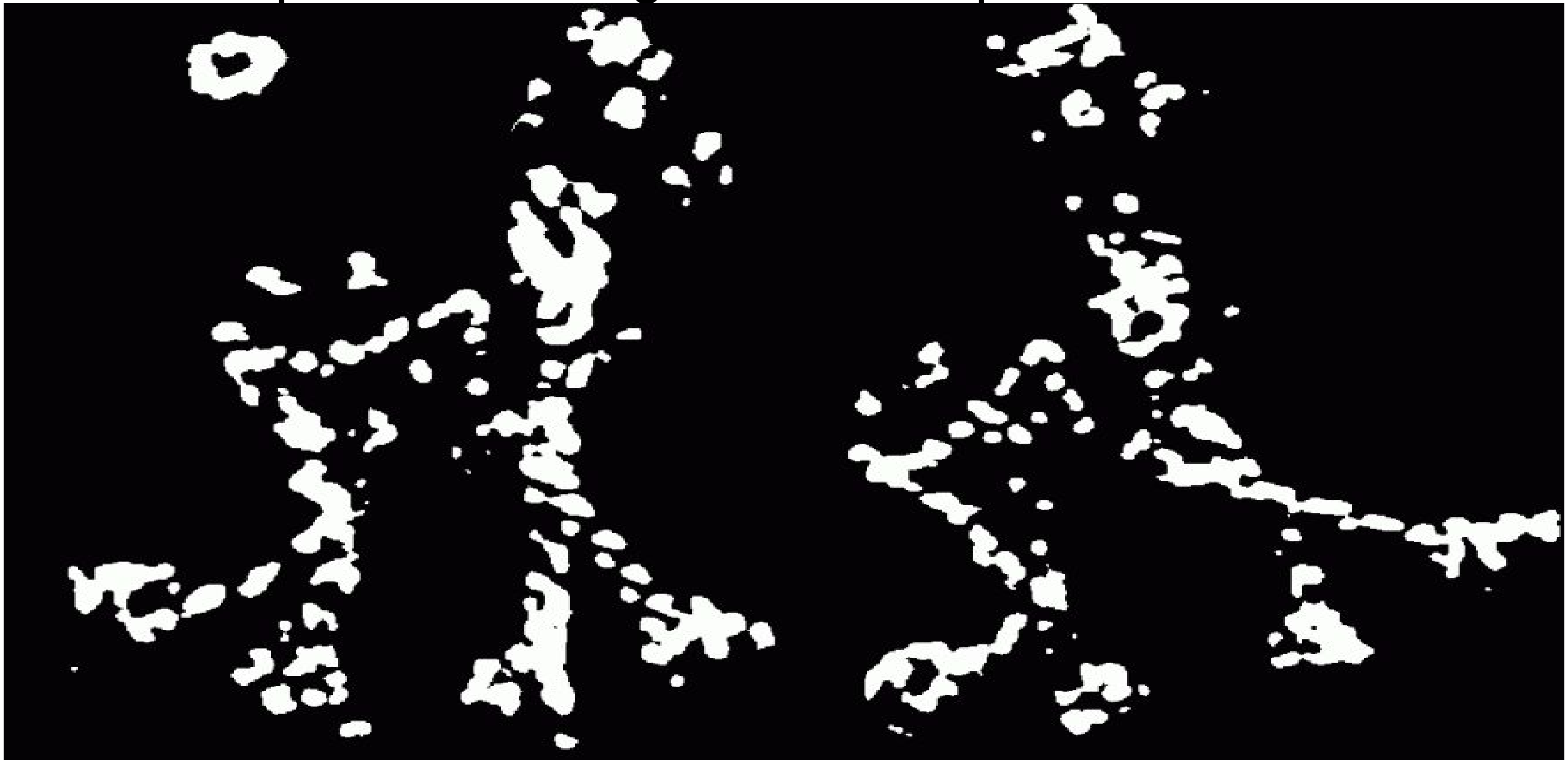
Harris Detector: Steps

Compute corner response R



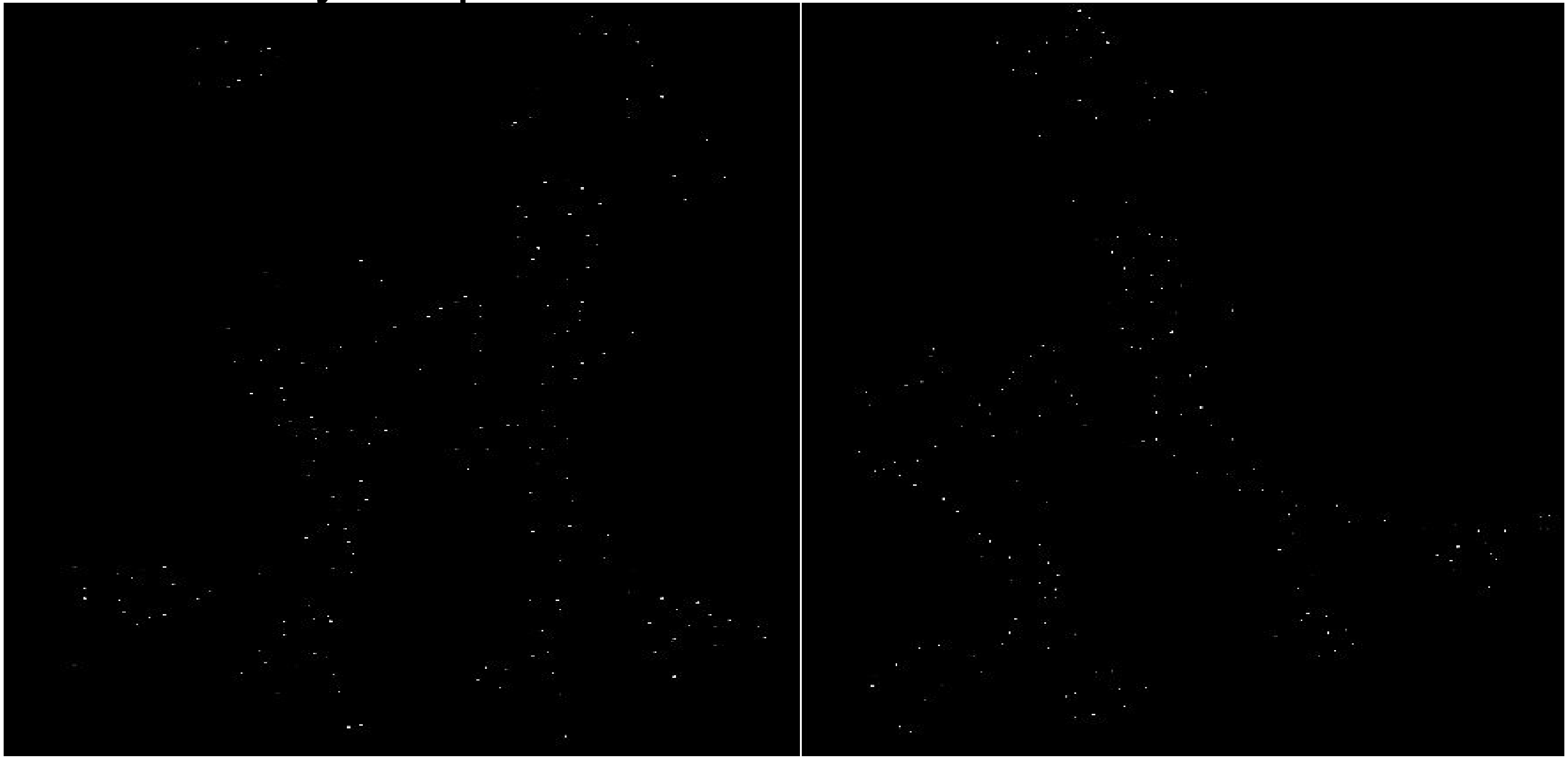
Harris Detector: Steps

Find points with large corner response: $R >$

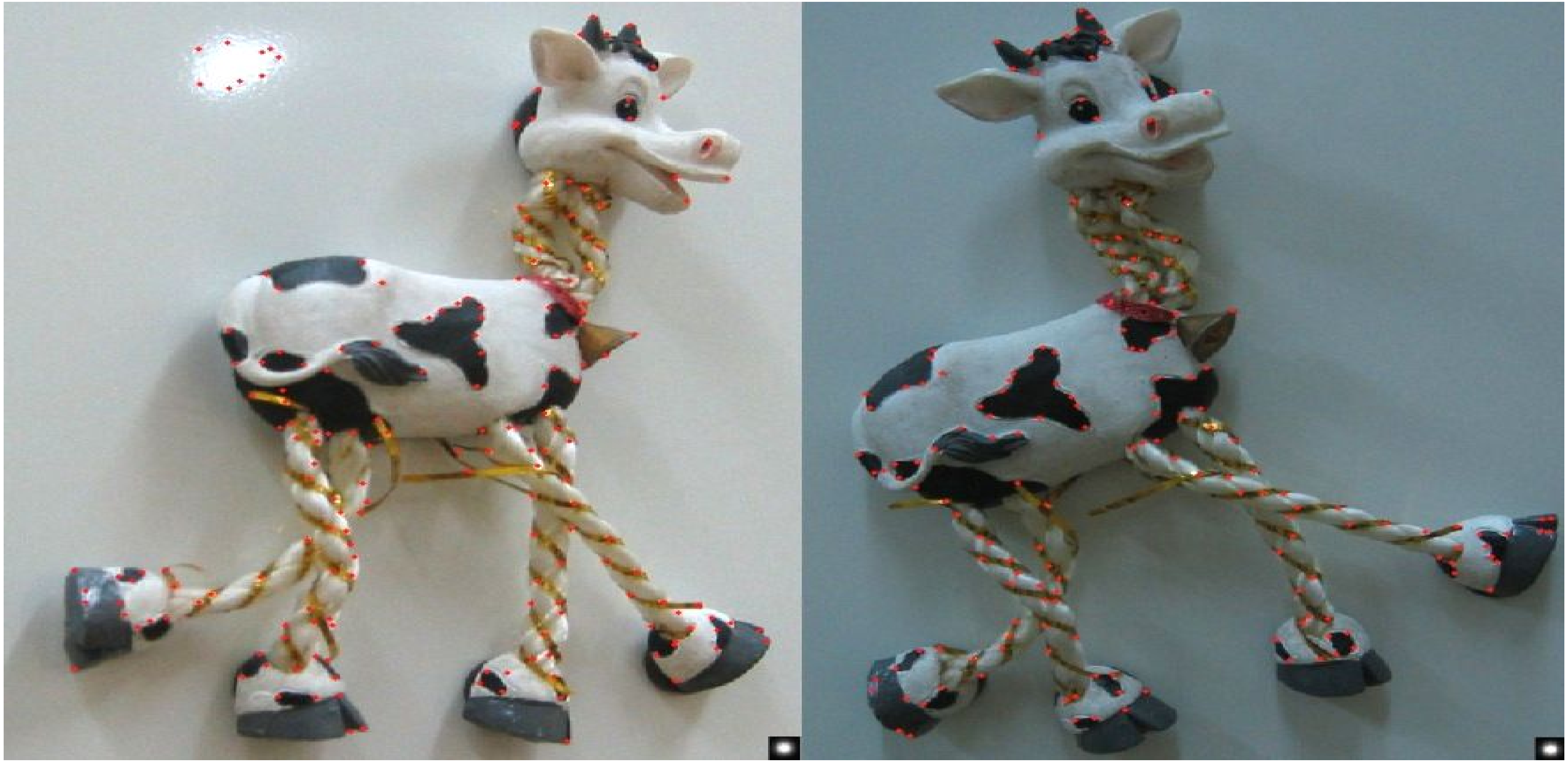


Harris Detector: Steps

Take only the points of local maxima of R



Harris Detector: Results



Simpler Response Function

Instead of

$$R = det(M) - \alpha \cdot trace(M)^2$$

We can use

$$f = \frac{1}{\frac{1}{\lambda_1} + \frac{1}{\lambda_2}} = \boxed{\frac{Det(H)}{Tr(H)}}$$

Properties of the Harris corner detector

- Translation invariant?

Properties of the Harris corner detector

- Translation invariant? Yes

Properties of the Harris corner detector

- Translation invariant? Yes
- Rotation invariant?

Properties of the Harris corner detector

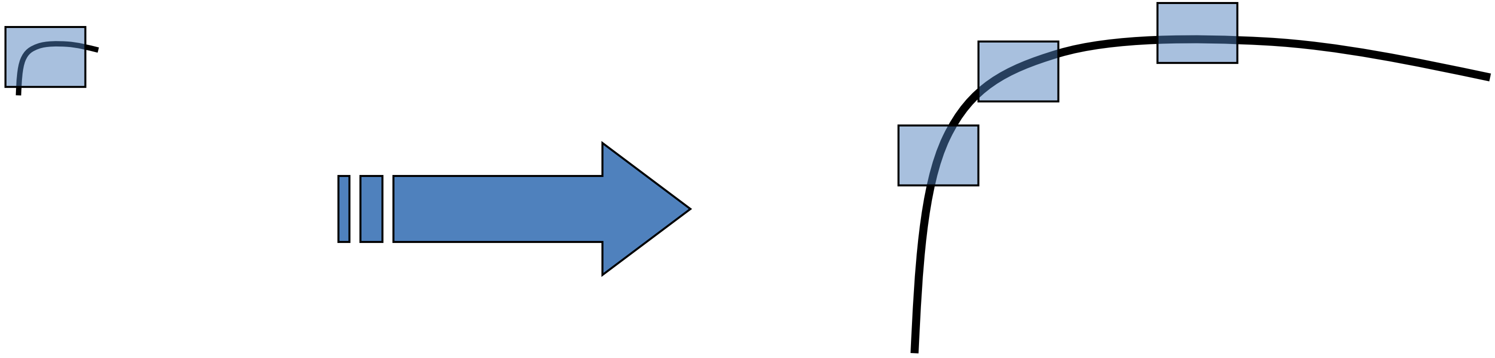
- Translation invariant? Yes
- Rotation invariant? Yes

Properties of the Harris corner detector

- Translation invariant? Yes
- Rotation invariant? Yes
- Scale invariant?

Properties of the Harris corner detector

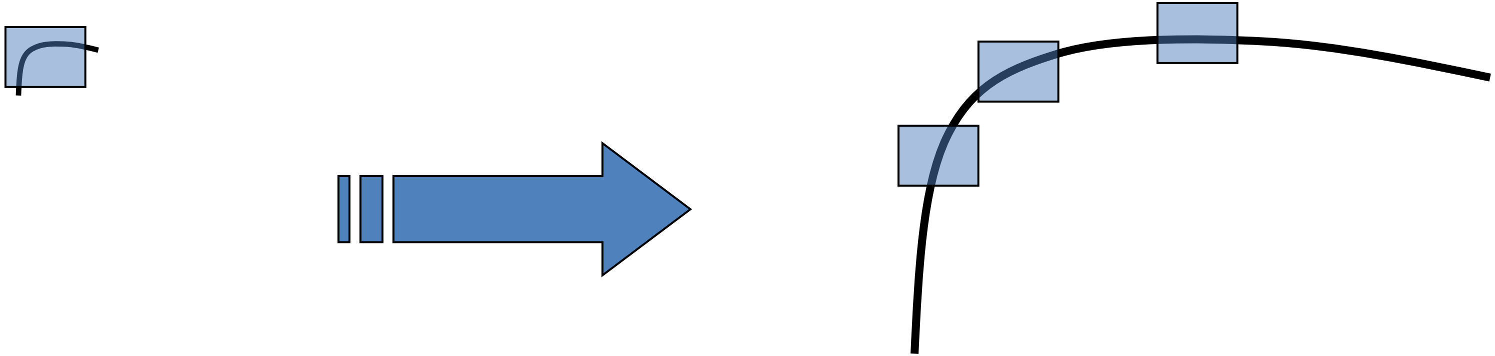
- Translation invariant? Yes
- Rotation invariant? Yes
- Scale invariant?



Properties of the Harris corner detector

- Translation invariant? Yes
- Rotation invariant? Yes
- Scale invariant?

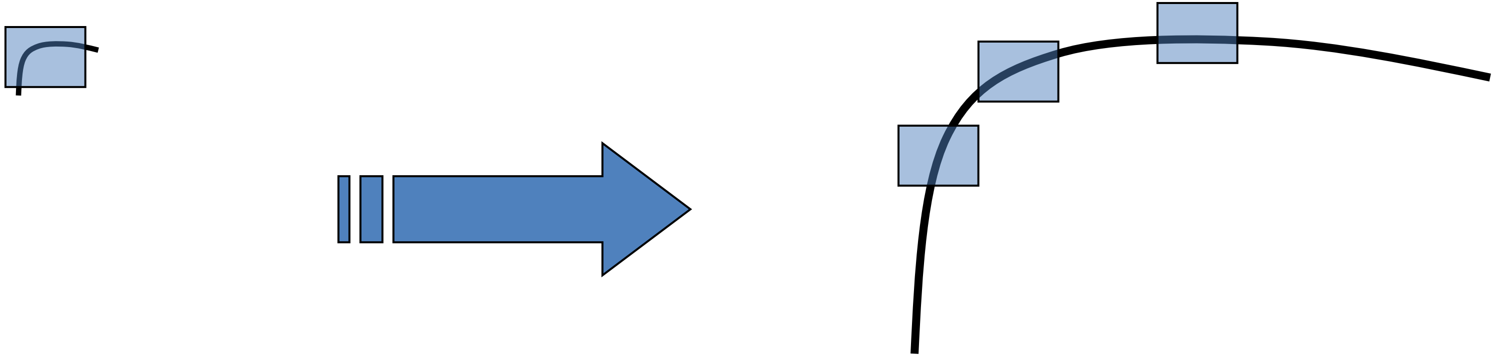
What's the problem?



Properties of the Harris corner detector

- Translation invariant? Yes
- Rotation invariant? Yes
- Scale invariant? No

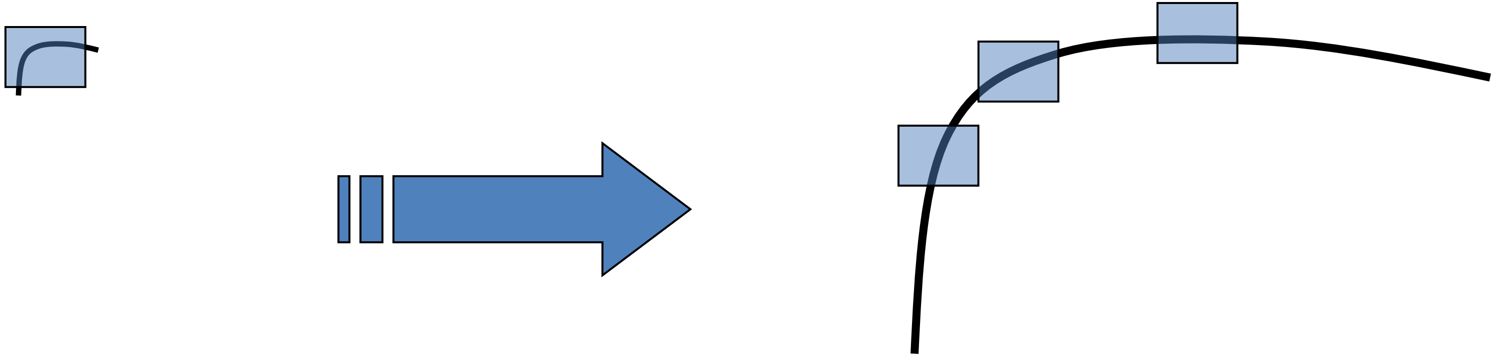
What's the problem?



Properties of the Harris corner detector

- Translation invariant? Yes
- Rotation invariant? Yes
- Scale invariant? No

What's the problem?



Corner !

All points will be classified as edges

Scale

Let's look at scale first:

What is the "best" scale?

Scale

Let’s look at scale first:

What is the “best” scale?

Scale

Let’s look at scale first:

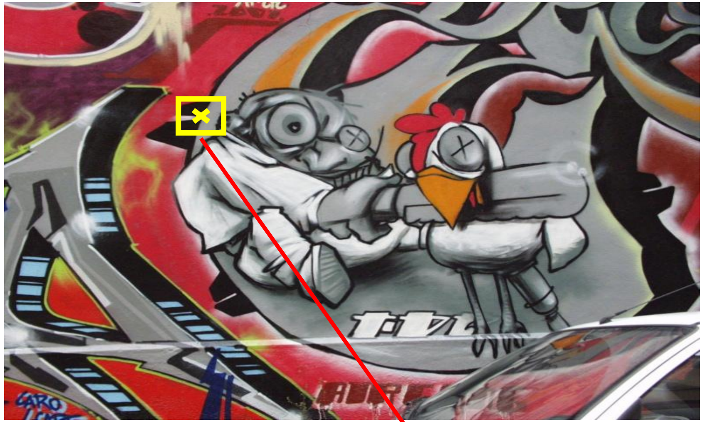
What is the “best” scale?

Scale

Let’s look at scale first:

What is the “best” scale?

Scale Invariance

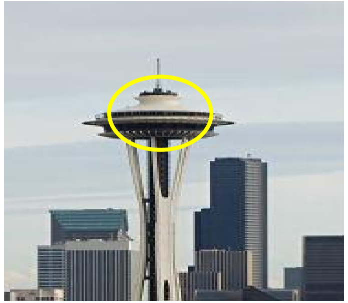


$$f(I_{i_1\ldots i_m}(x,\sigma)) = f(I_{i_1\ldots i_m}(x',\sigma'))$$

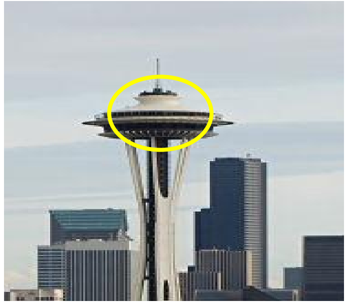
How can we independently select interest points in each image, such that the detections are repeatable across different scales?

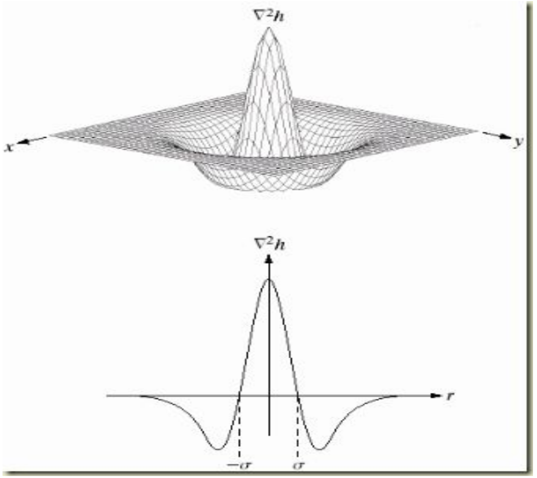
K. Grauman, B. Leibe

Differences between Inside and Outside



Differences between Inside and Outside





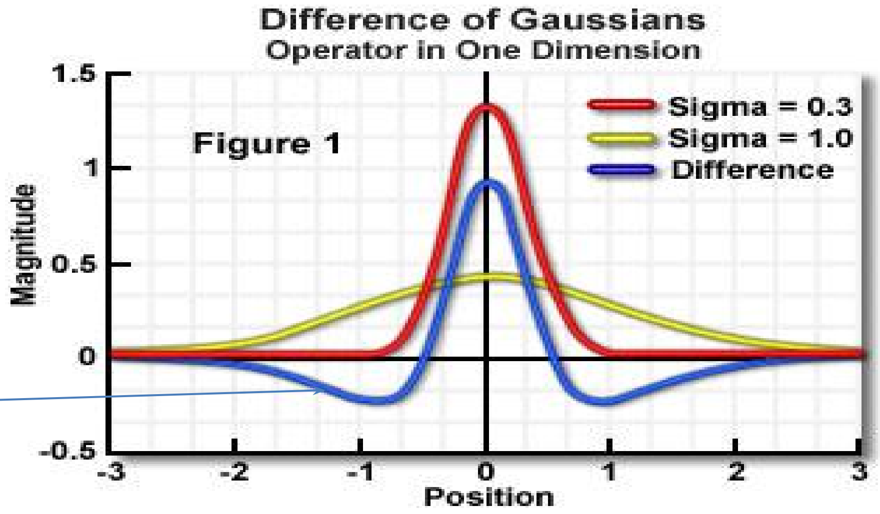
1. We can use a Laplacian function

Scale

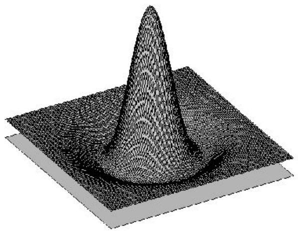
But we use a Gaussian.
Why Gaussian?

It is invariant to scale change, i.e., $f * \mathcal{G}_\sigma * \mathcal{G}_{\sigma'} = f * \mathcal{G}_{\sigma''}$ and has several other nice properties. Lindeberg, 1994

In practice, the Laplacian is approximated using a Difference of Gaussian (DoG).



Difference-of-Gaussian (DoG)

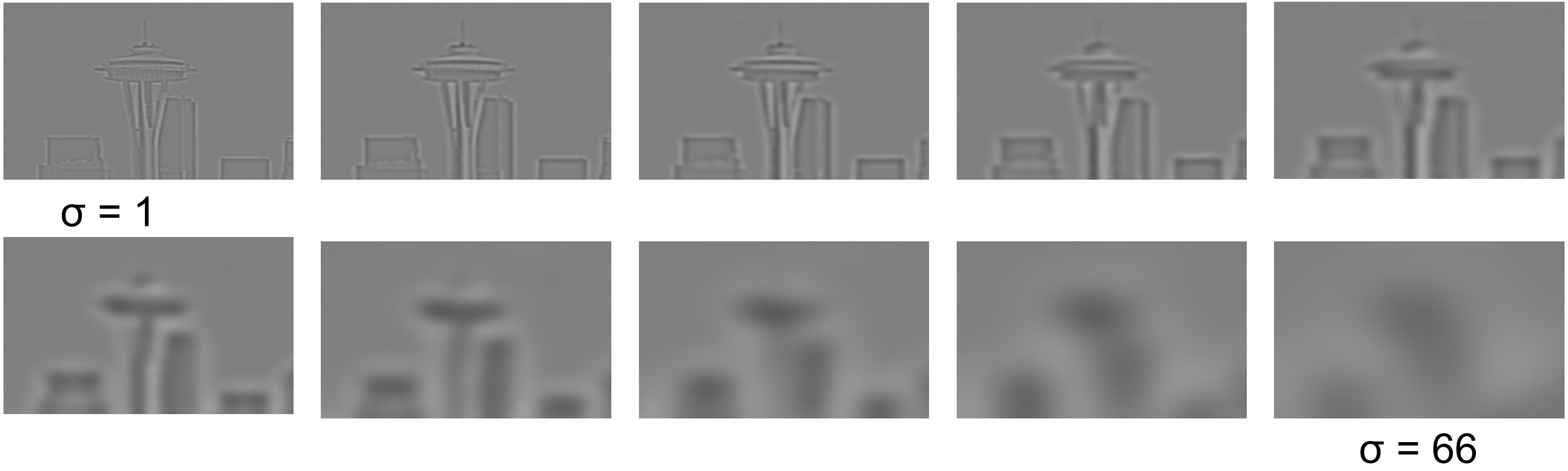


G1 - G2 = DoG

K. Grauman, B. Leibe

DoG example

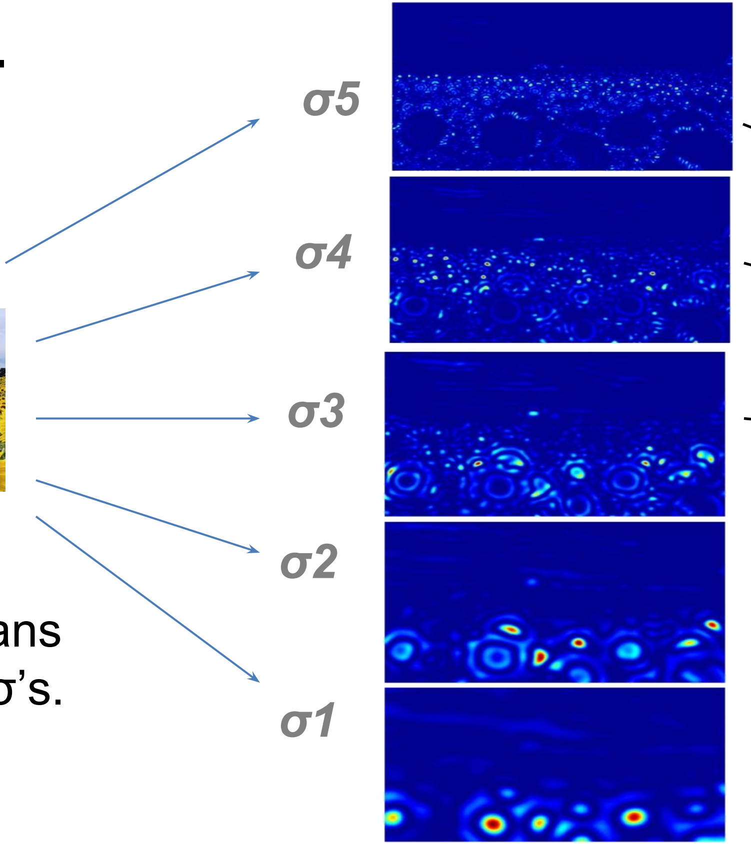
Take Gaussians at multiple spreads and uses DoGs.



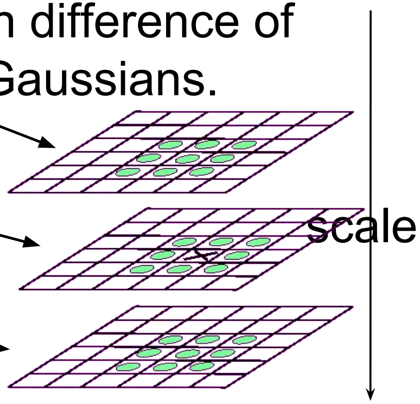
Scale invariant interest points

Interest points are local maxima in both position and scale.

Apply Gaussians with different σ's.



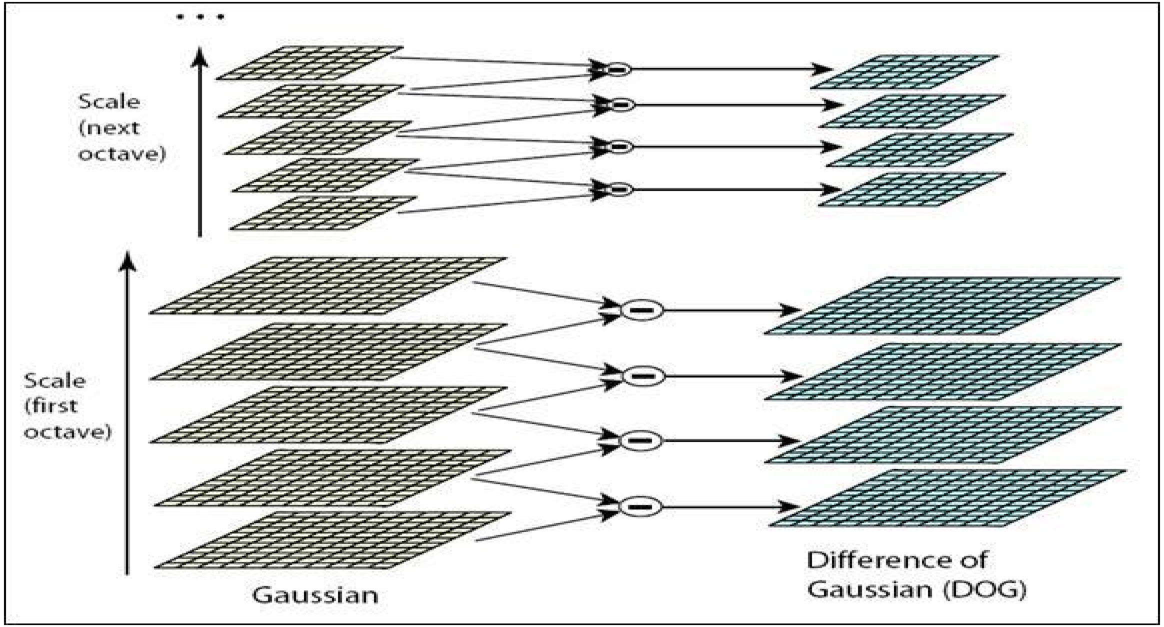
Look for extrema in difference of Gaussians.



⇒ **List of**
(*x*, *y*, σ)

Scale

In practice the image is downsampled for larger sigmas.



Lowe, 2004.

Lowe's Pyramid Scheme

s+2 filters

$\sigma_{s+1}=2^{(s+1)/s}\sigma_0$

.

.

$\sigma_i=2^{i/s}\sigma_0$

.

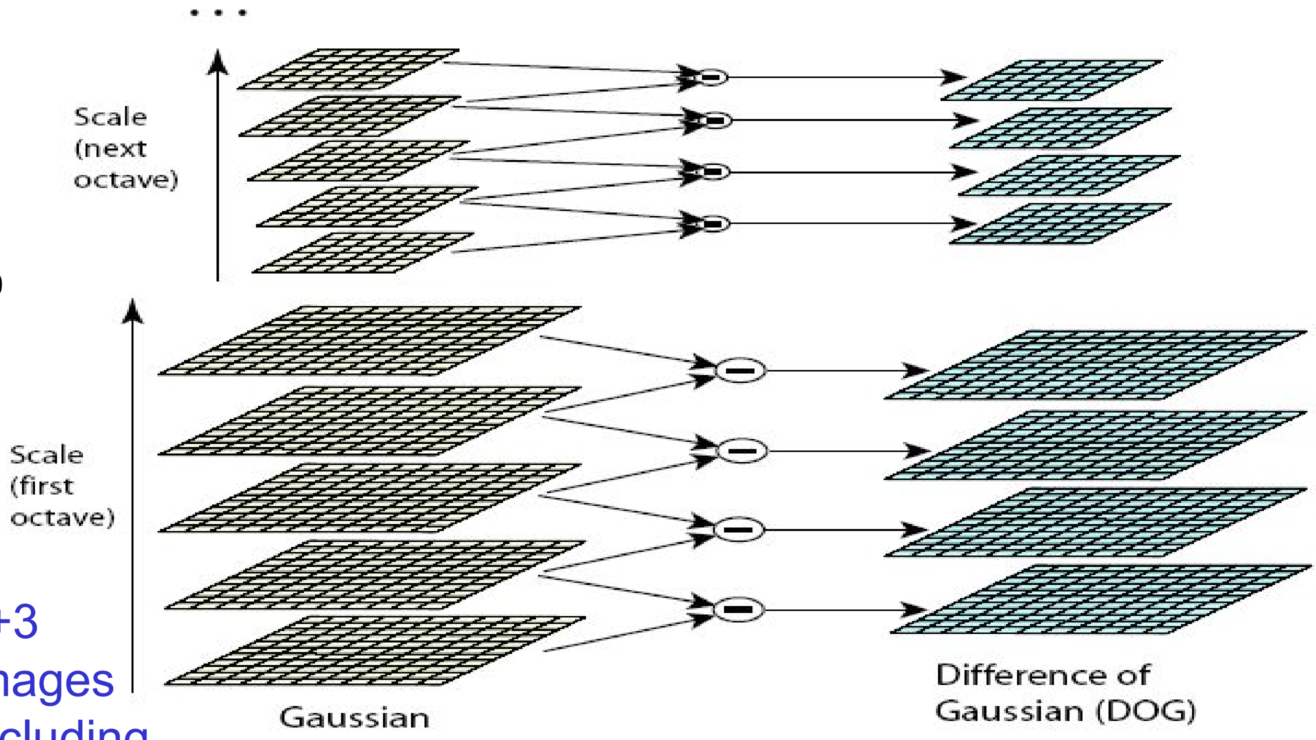
.

$\sigma_2=2^{2/s}\sigma_0$

$\sigma_1=2^{1/s}\sigma_0$

σ_0

s+3 images including original



s+2 difference images

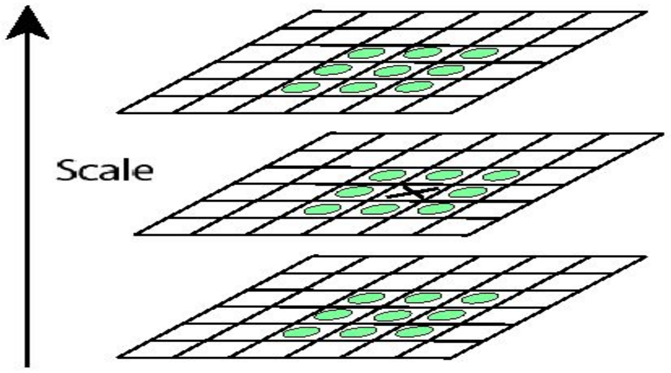
The parameter **s** determines the number of images per octave.

Key point localization

Detect maxima and minima of difference-of-Gaussian in scale space

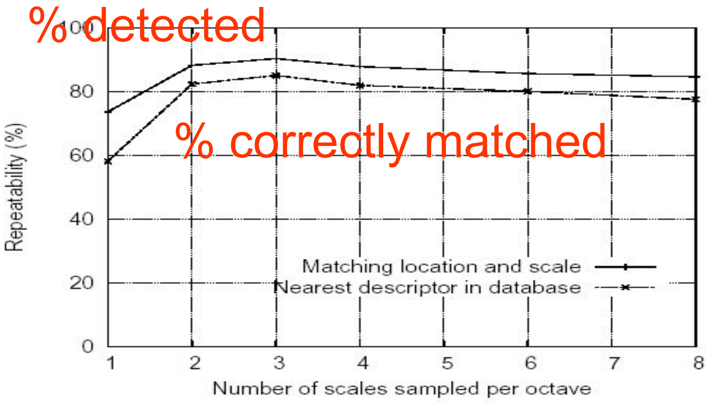
Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

s+2 difference images.
top and bottom ignored.
s planes searched.

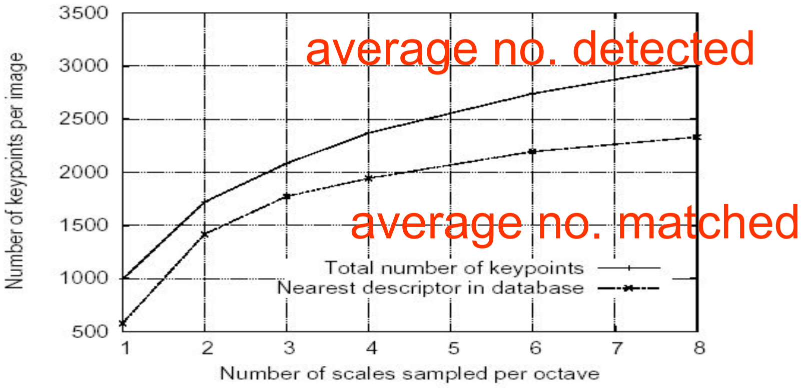


For each max or min found, output is the **location** and the **scale**.

Scale-space extrema detection: experimental results over 32 images that were synthetically transformed and noise added.



Stability



Expense

Sampling in scale for efficiency

How many scales should be used per octave? S=?

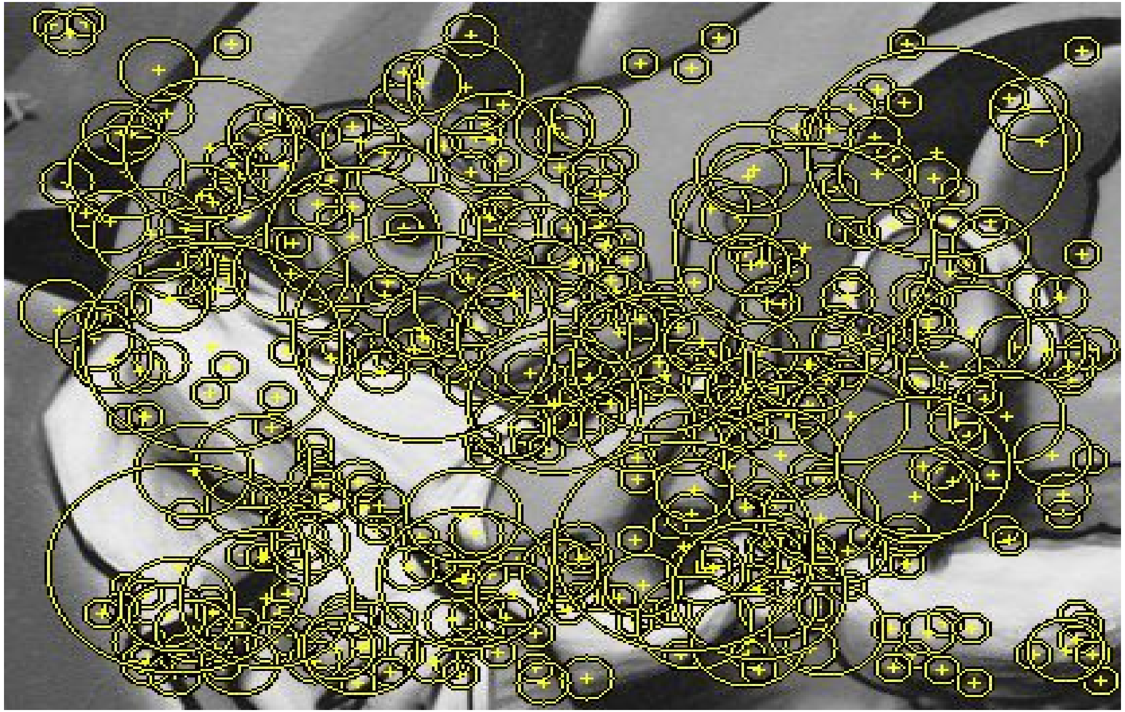
More scales evaluated, more keypoints found

S < 3, stable keypoints increased too

S > 3, stable keypoints decreased

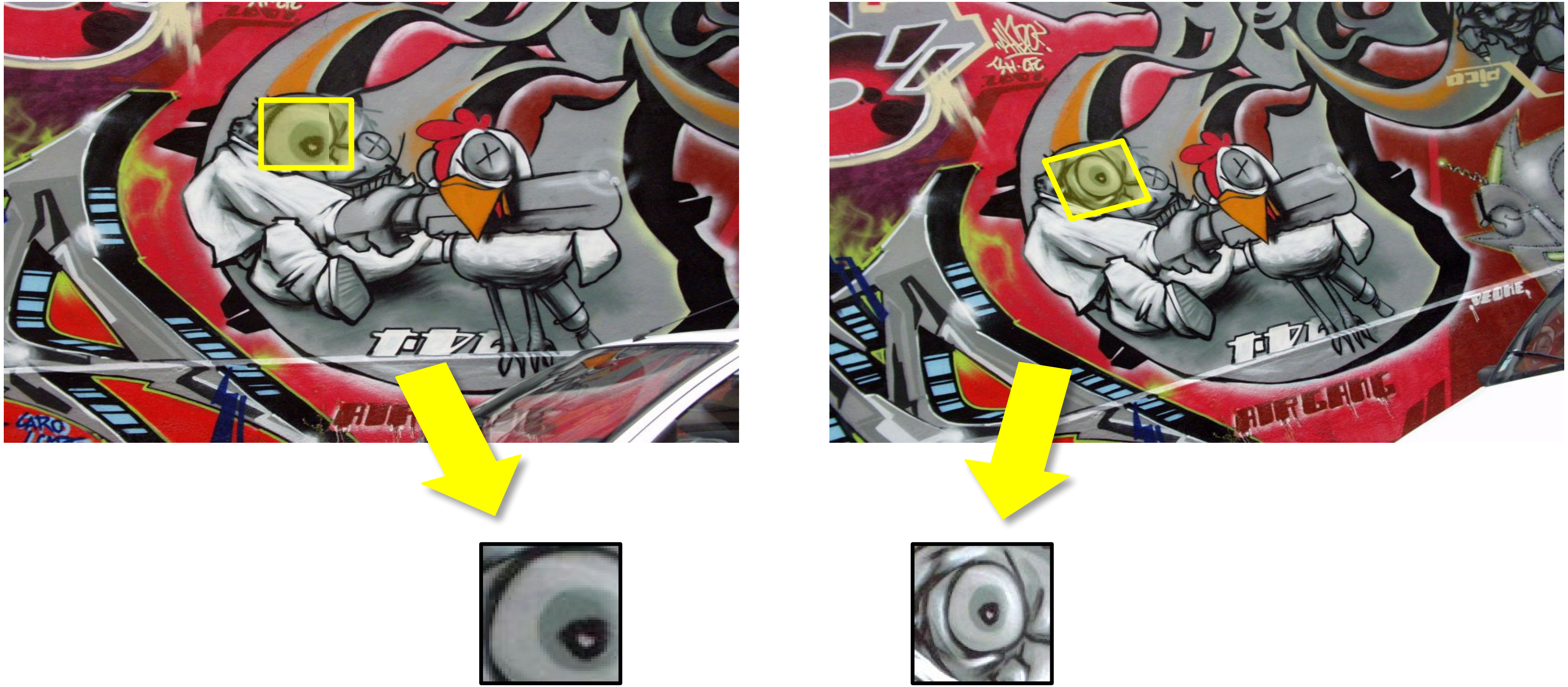
S = 3, maximum stable keypoints found

Results: Difference-of-Gaussian



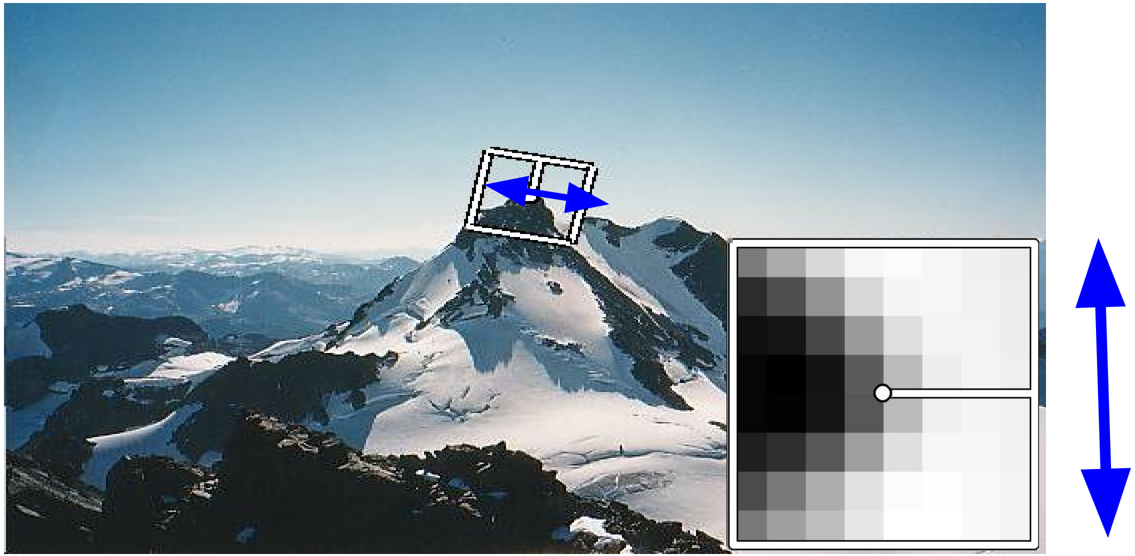
K. Grauman, B. Leibe

How can we find correspondences?



Similarity transform

Rotation invariance



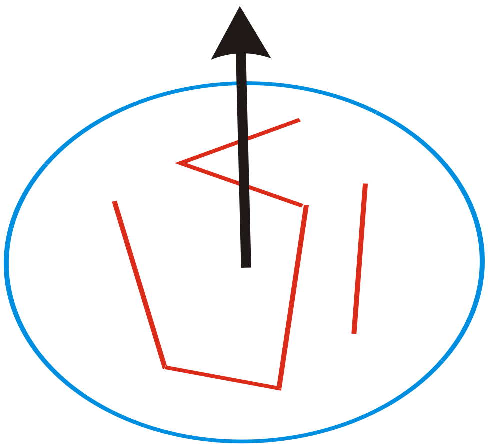
- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

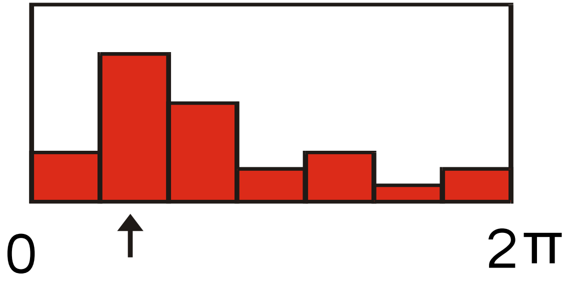
Image from Matthew Brown

Orientation Normalization

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

[Lowe, SIFT, 1999]

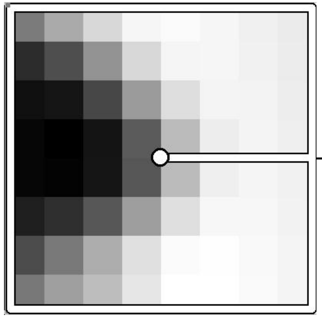




What's next?

Once we have found the keypoints and a dominant orientation for each,

we need to describe the (rotated and scaled) neighborhood about each.



128-dimensional vector

Review

- Interest points
  - Harris corner detector
  - invariance
    - scale
    - rotation

CONCORDIA.CA