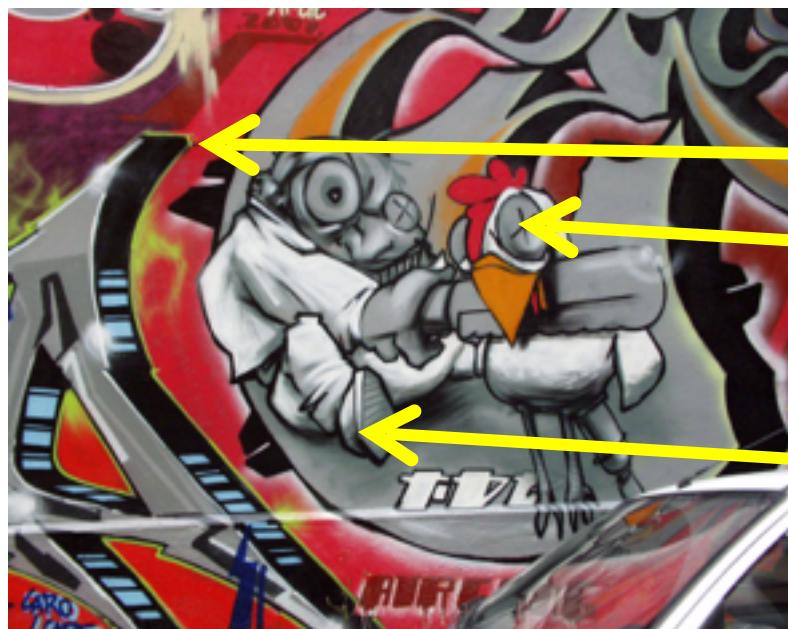


Interest points

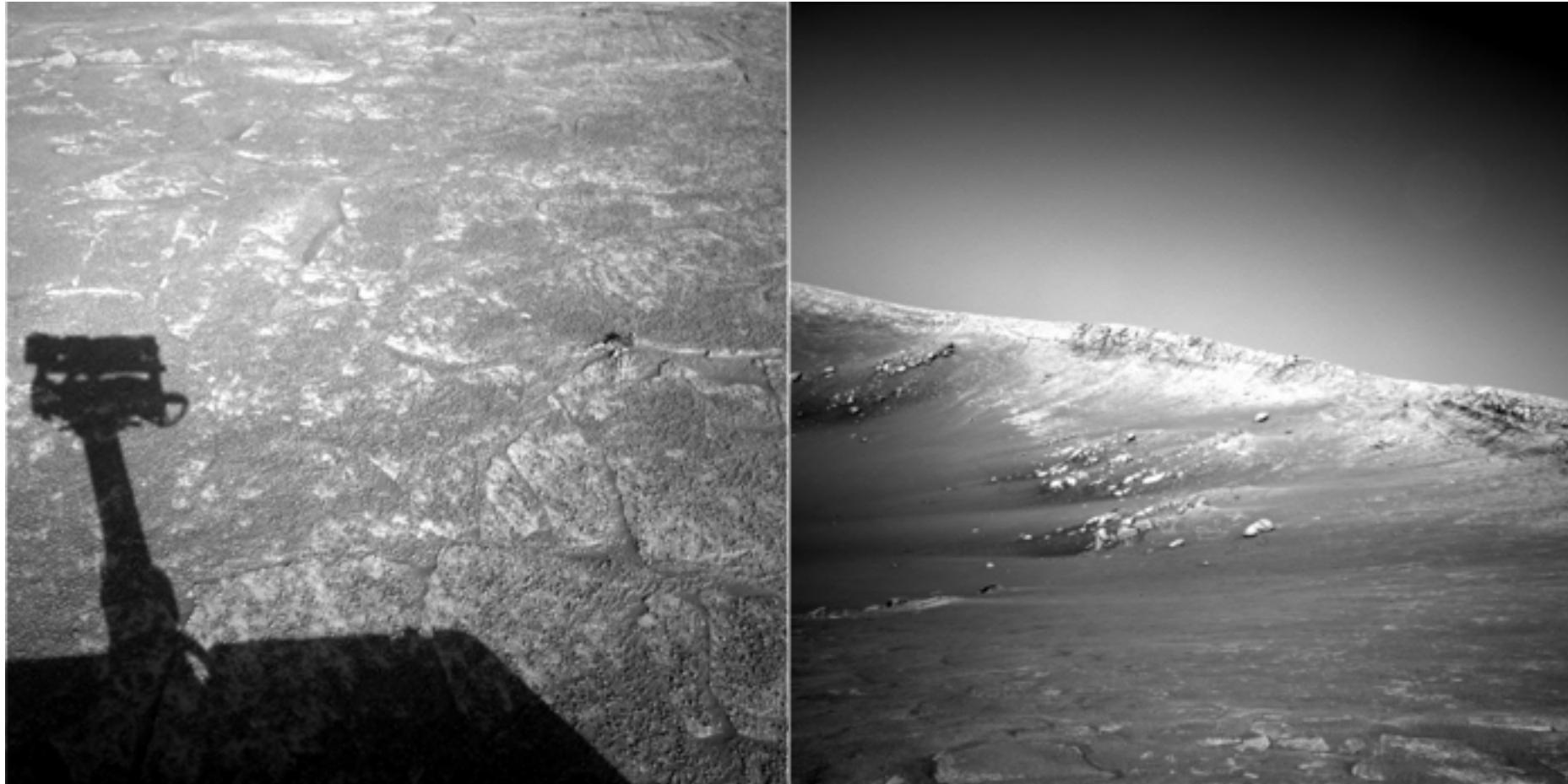
CSE 576
Ali Farhadi

Many slides from Steve Seitz, Larry Zitnick

How can we find corresponding points?

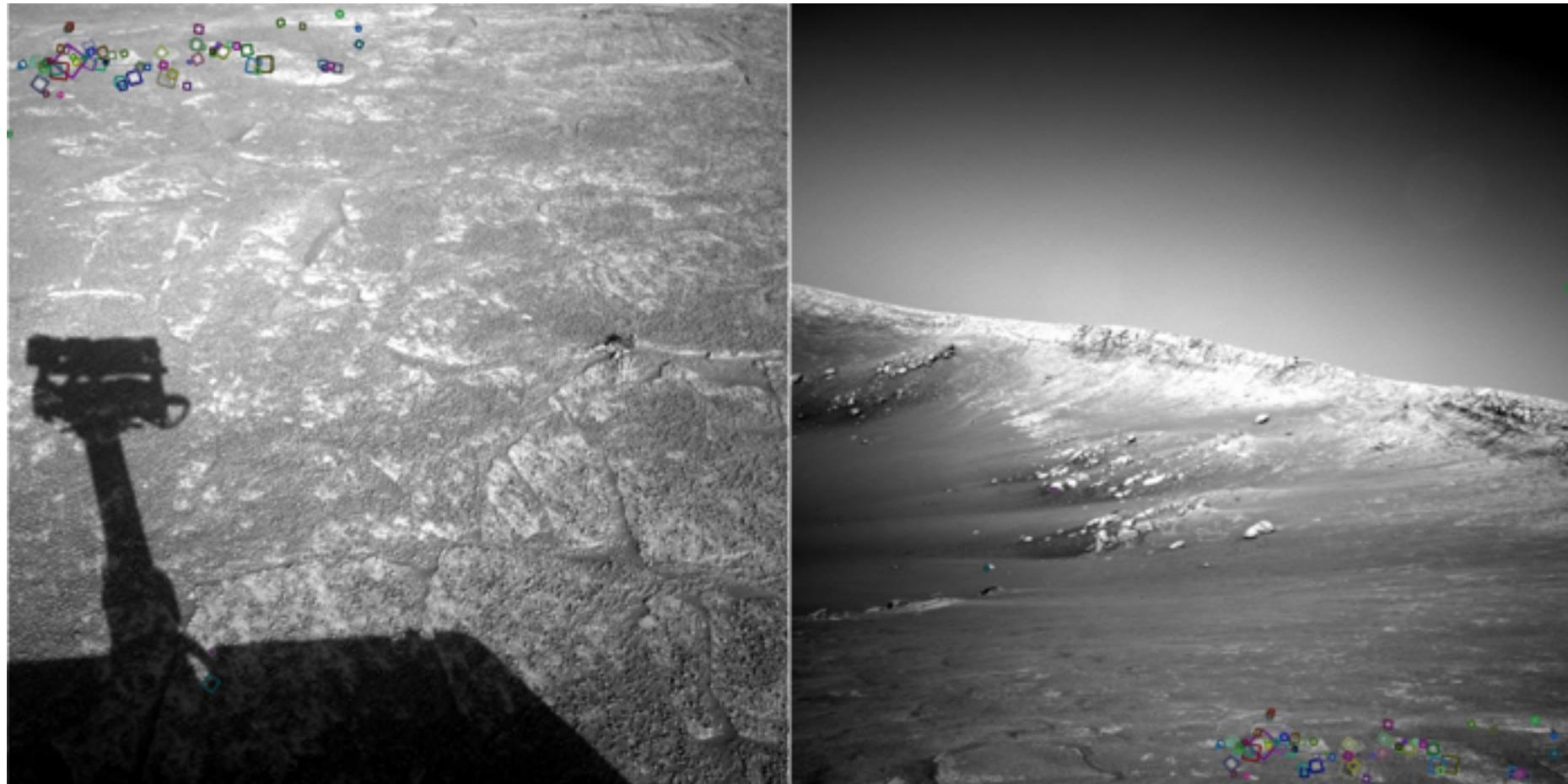


Not always easy



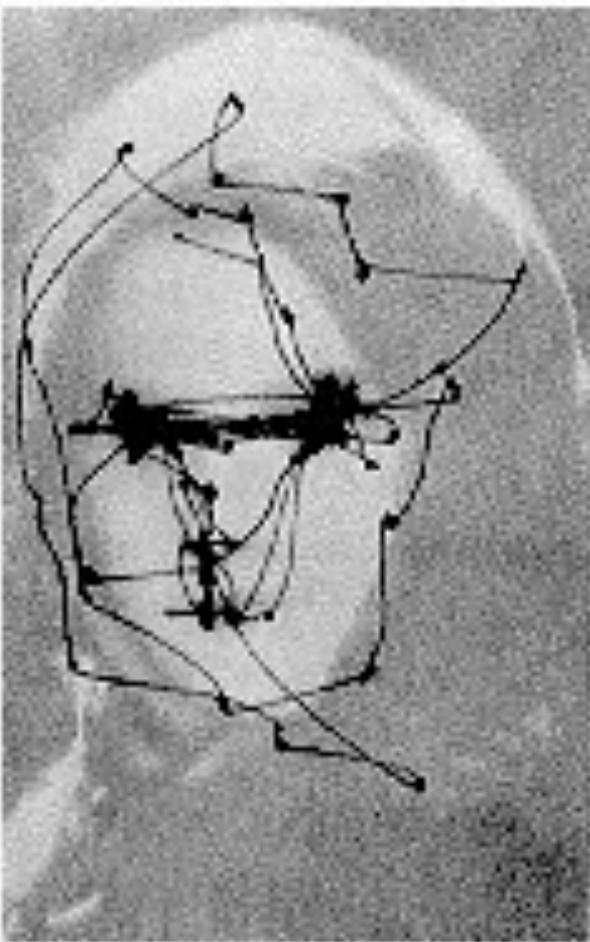
NASA Mars Rover images

Answer below (look for tiny colored squares...)



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

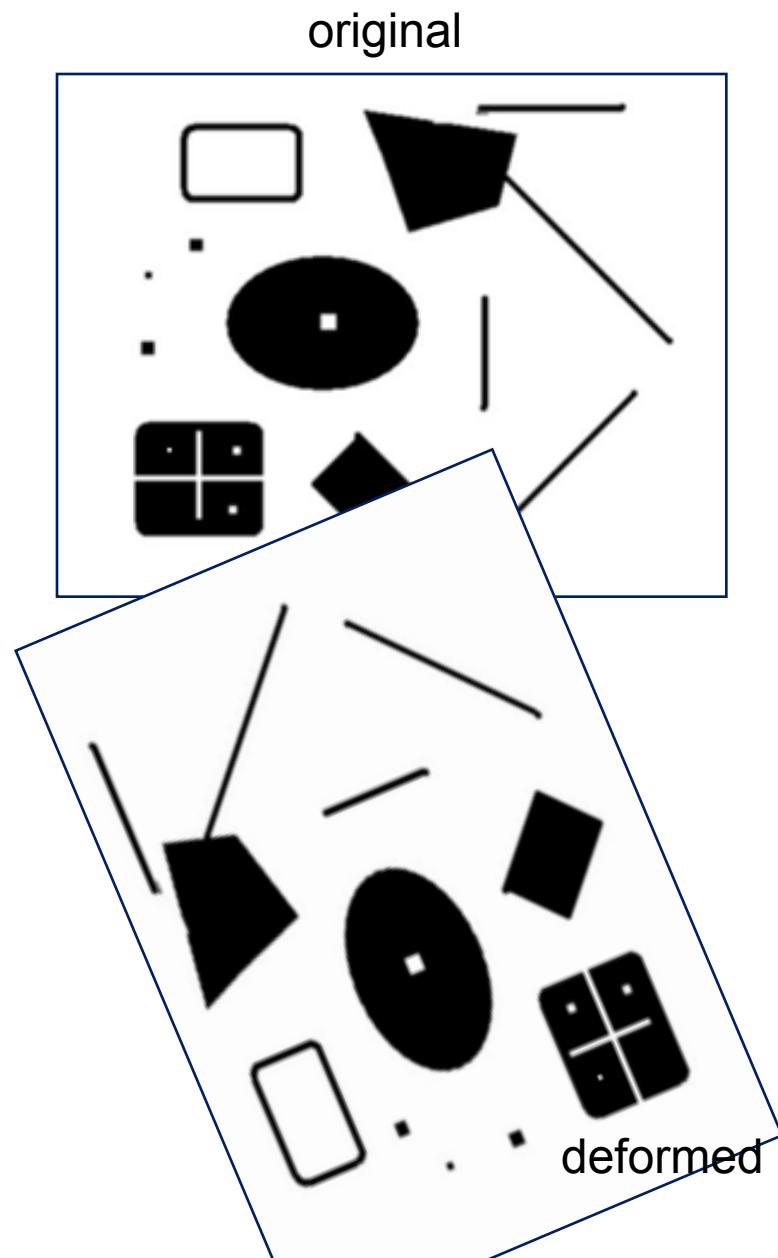
Human eye movements



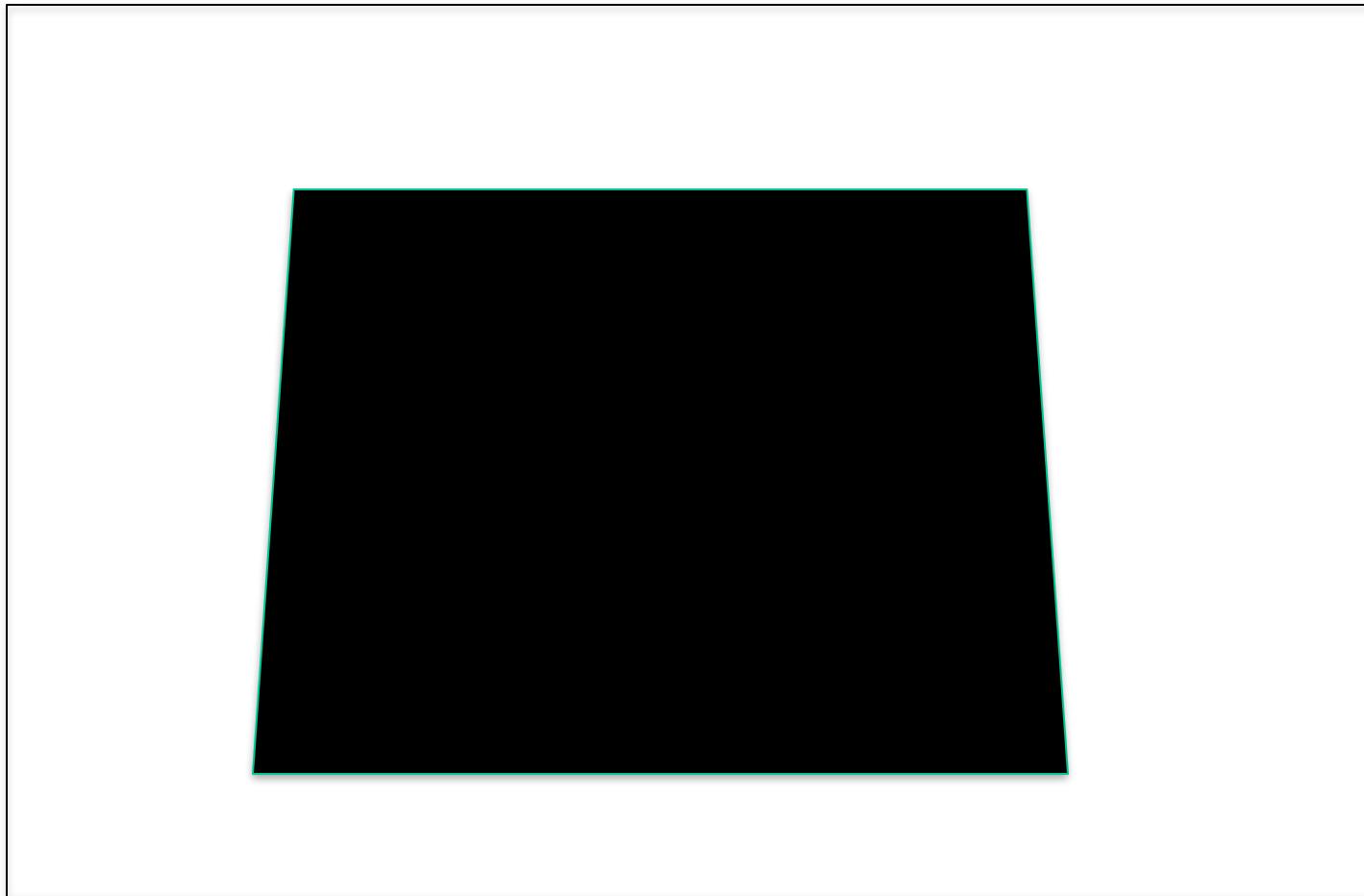
Yarbus eye tracking

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?

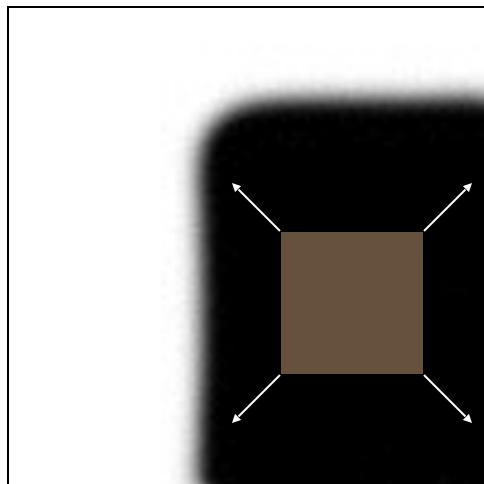


Intuition

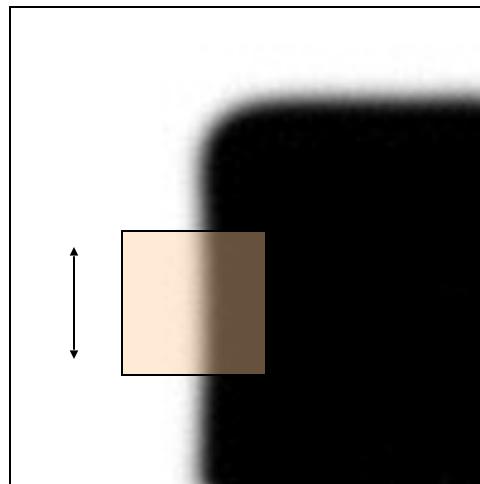


Corners

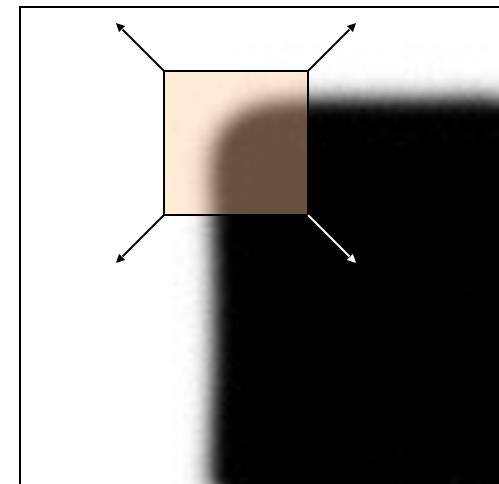
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity



“flat” region:
no change in
all directions

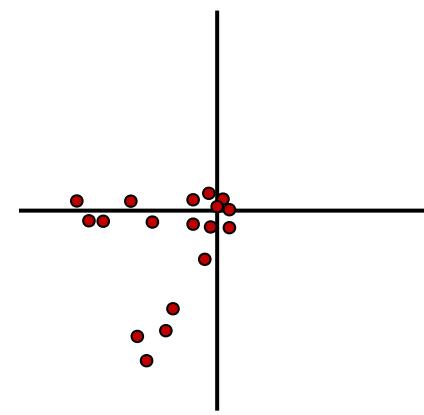
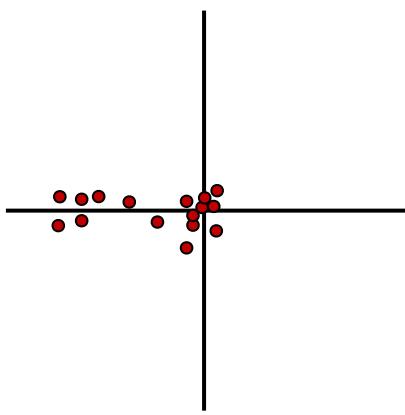
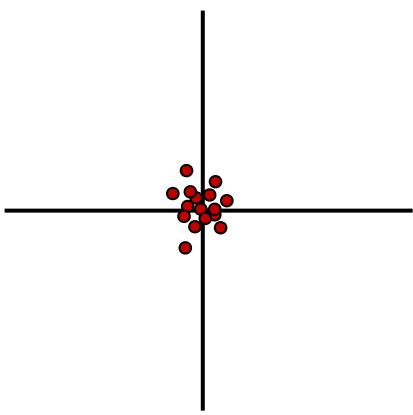
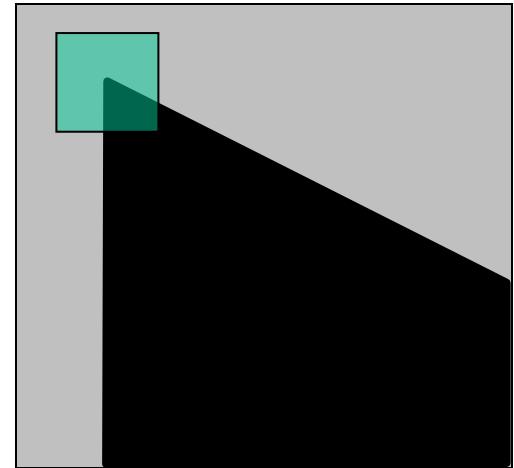
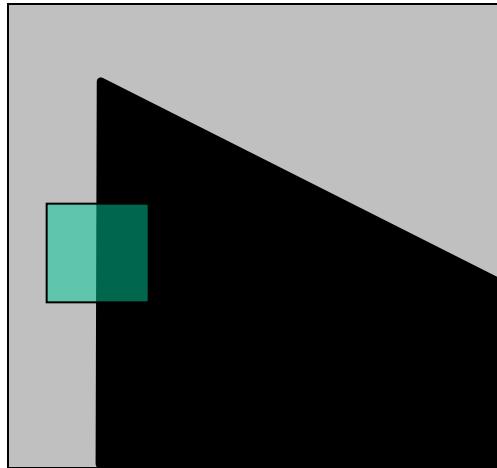
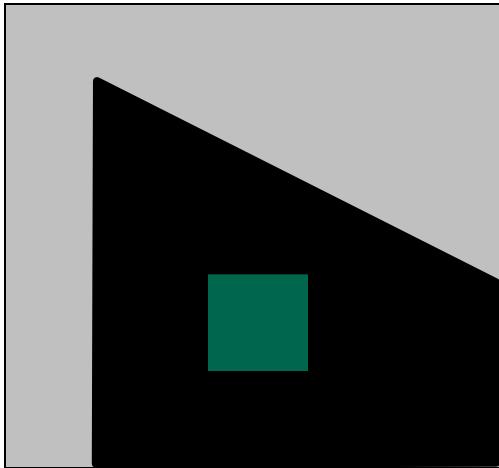


“edge”:
no change along
the edge
direction



“corner”:
significant
change in all
directions

Let's look at the gradient distributions



Principle Component Analysis

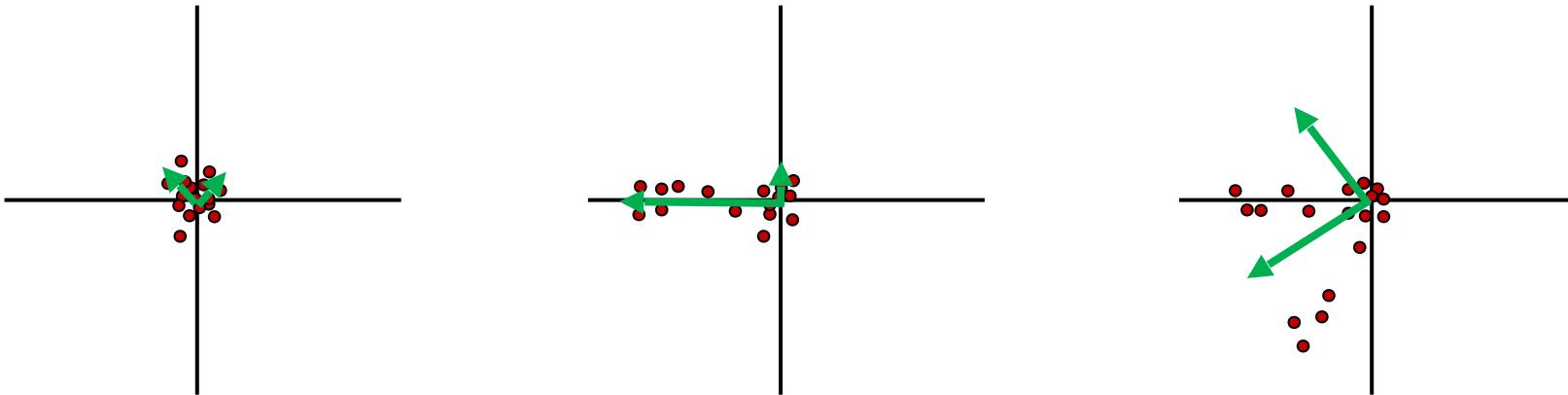
Principal component is the direction of highest variance.

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

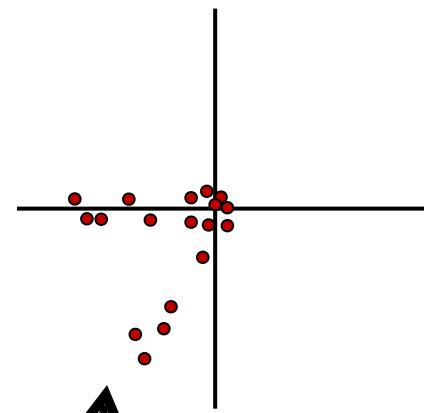
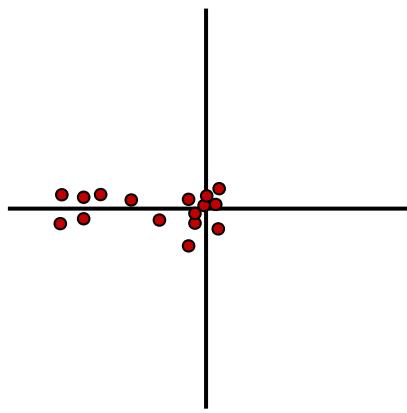
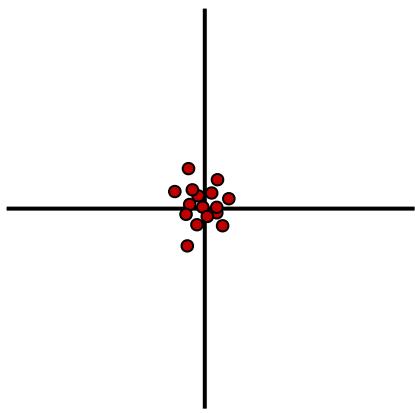
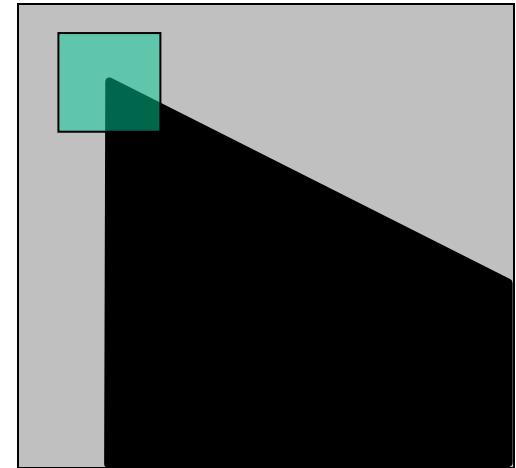
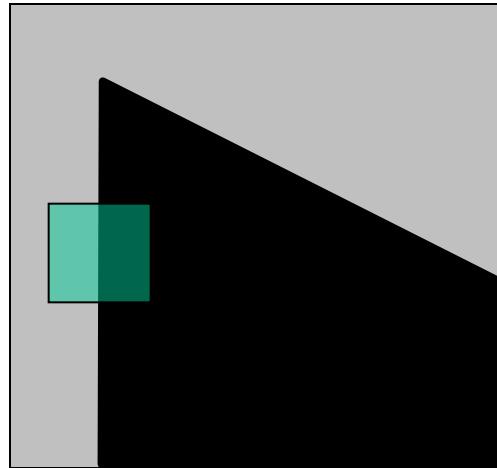
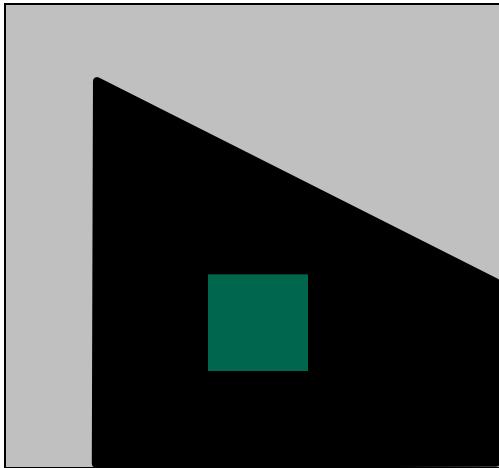
How to compute PCA components:

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.

$$Hx = \lambda x$$



Corners have ...

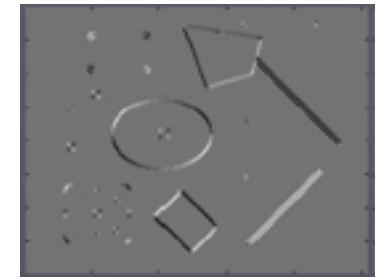
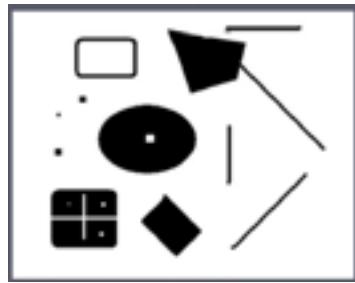


Both eigenvalues are large!

Second Moment Matrix

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

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



Notation:

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

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

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

The math

To compute the eigenvalues:

1. Compute the covariance matrix.

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

↑
Typically Gaussian weights

2. Compute eigenvalues.

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

Corner Response Function

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

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

Reminder:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc \quad \text{trace} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = a + d$$

Harris detector: Steps

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

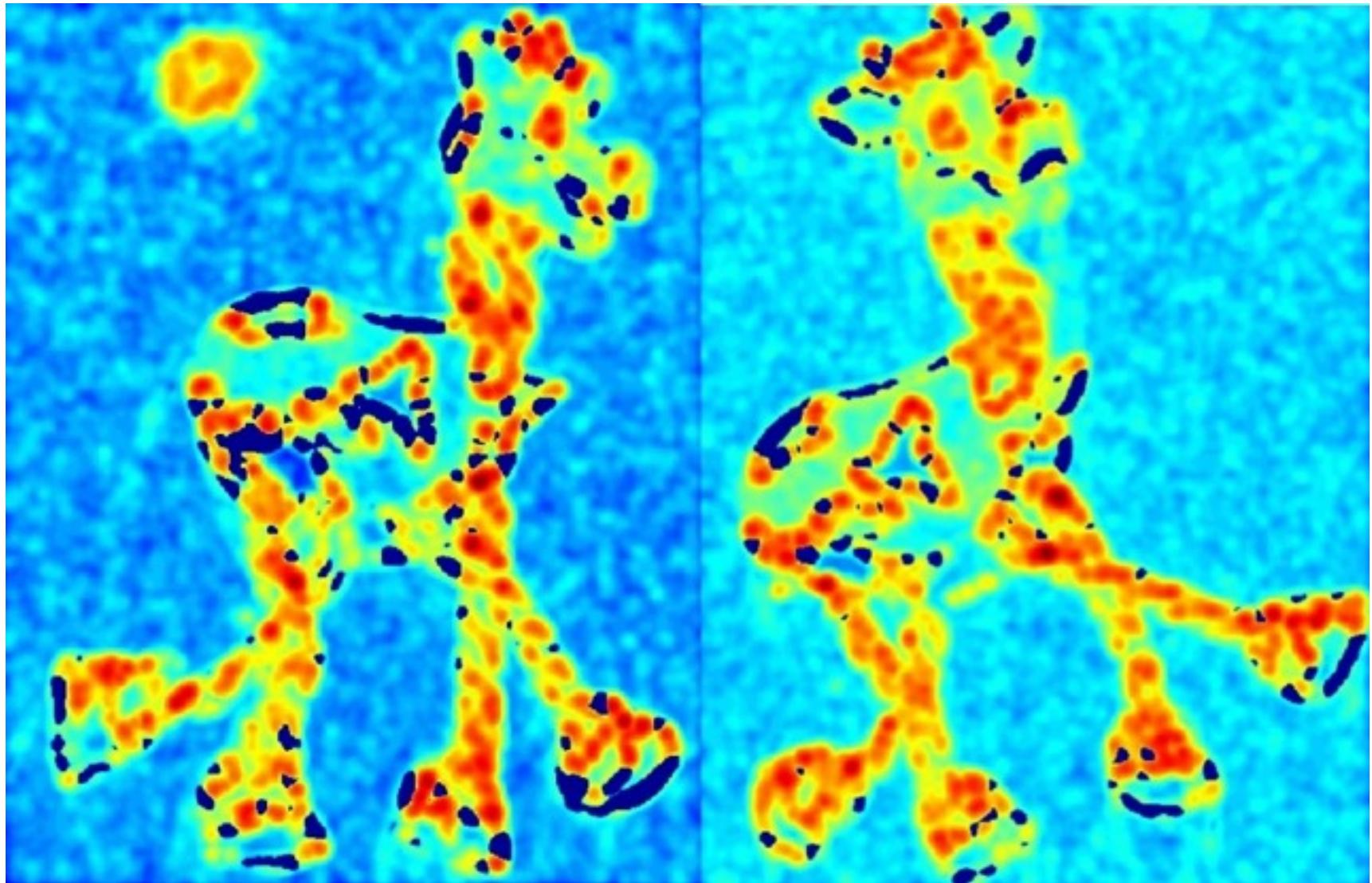
C.Harris and M.Stephens. "[A Combined Corner and Edge Detector.](#)"
Proceedings of the 4th Alvey Vision Conference: pages 147—151, 1988.

Harris Detector: Steps



Harris Detector: Steps

Compute corner response R



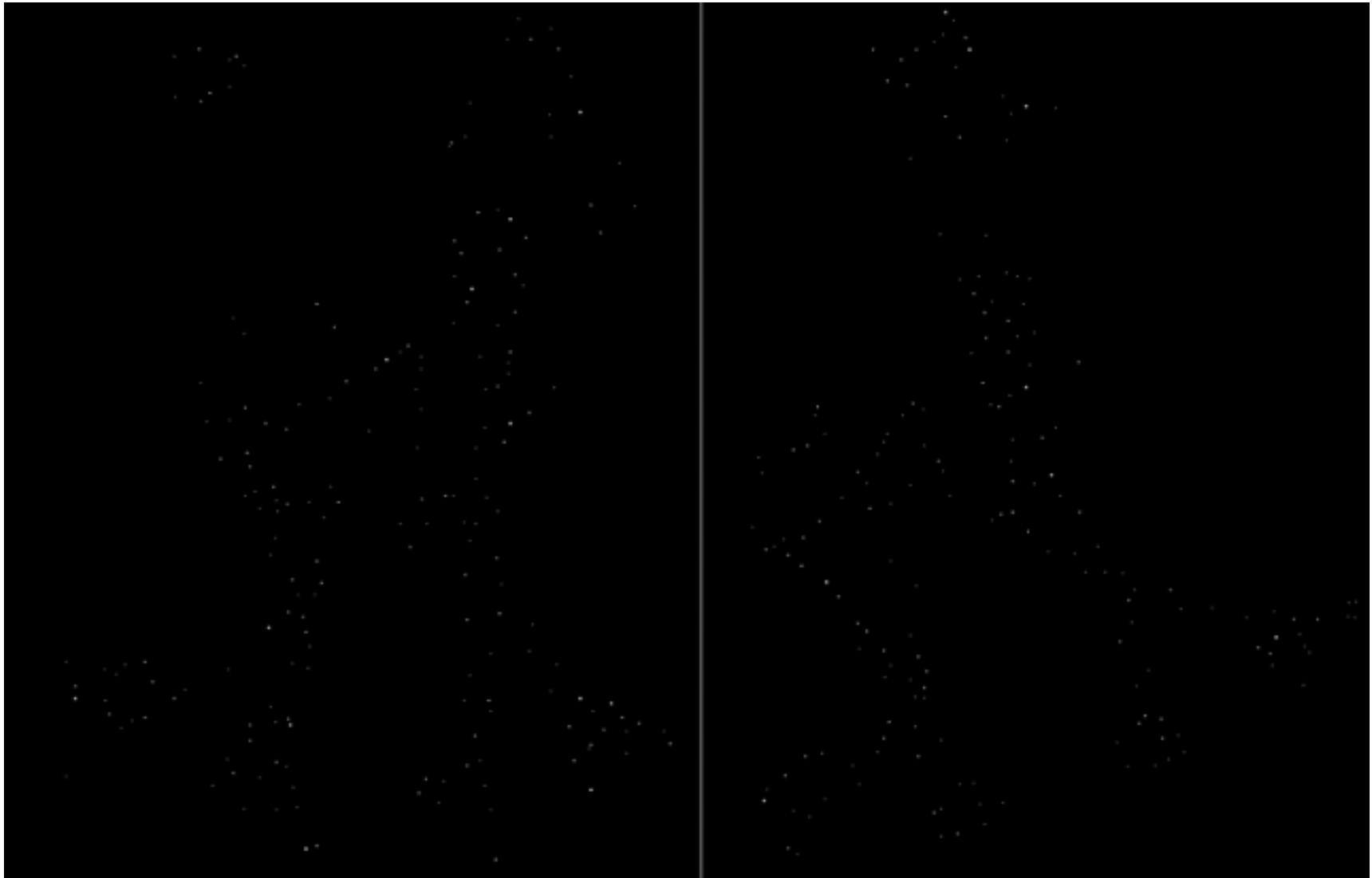
Harris Detector: Steps

Find points with large corner response: $R > \text{threshold}$



Harris Detector: Steps

Take only the points of local maxima of R



Harris Detector: Steps



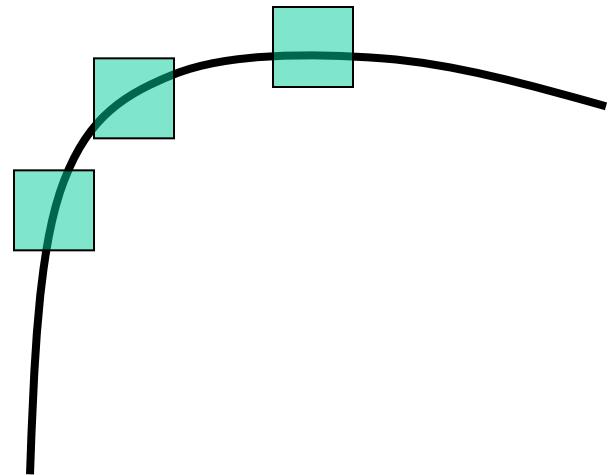
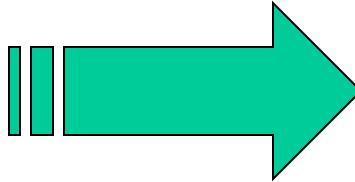
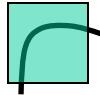
Simpler Response Function

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

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

Properties of the Harris corner detector

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



Corner !

All points will be
classified as edges

Scale

Let's look at scale first:



What is the “best” scale?

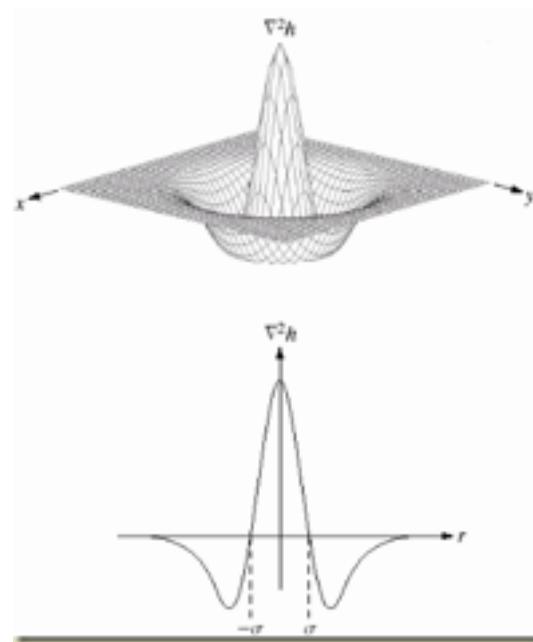
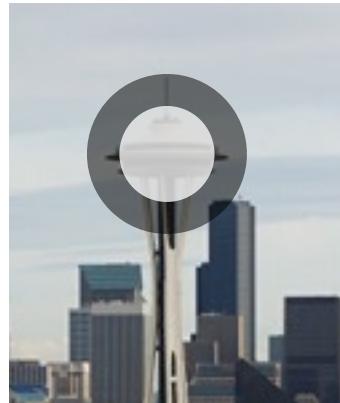
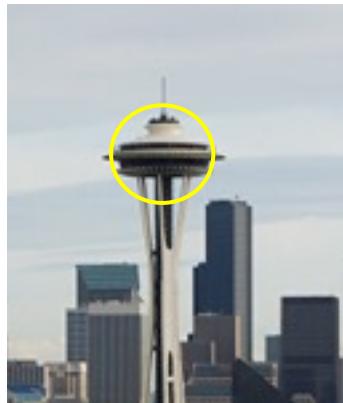
Scale Invariance



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

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

Differences between Inside and Outside

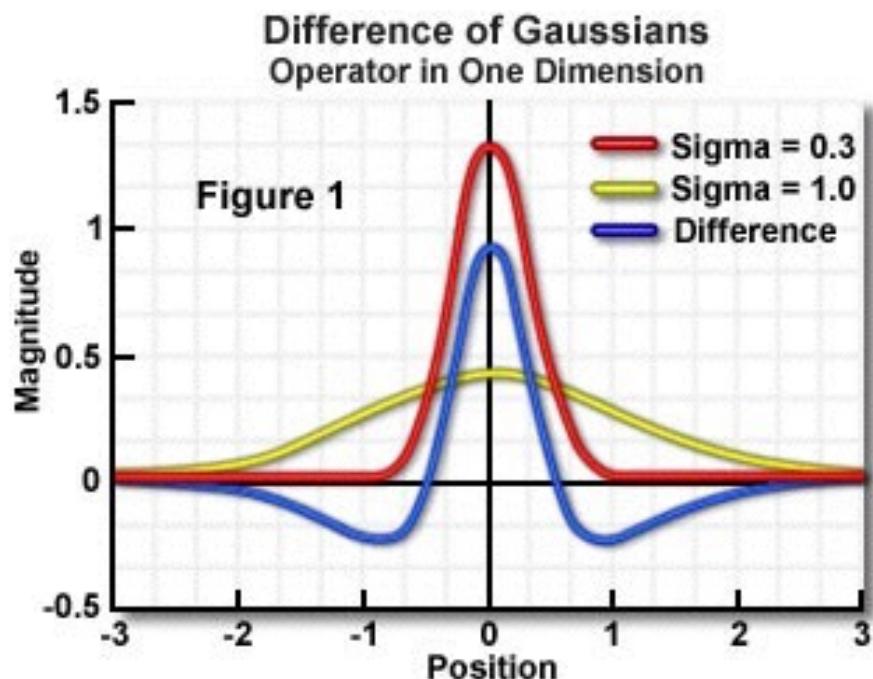


Scale

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)



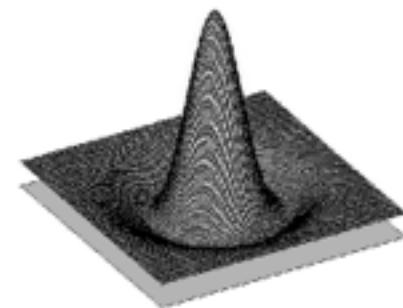
-



=



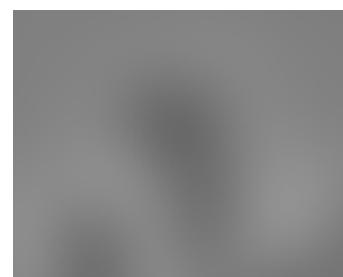
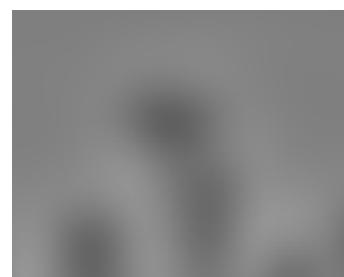
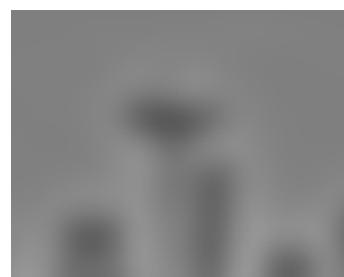
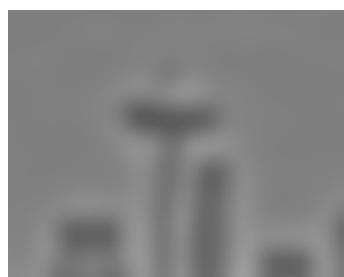
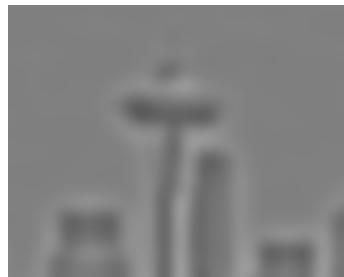
K. Grauman, B. Leibe



DoG example



$\sigma = 1$



$\sigma = 66$

Scale invariant interest points

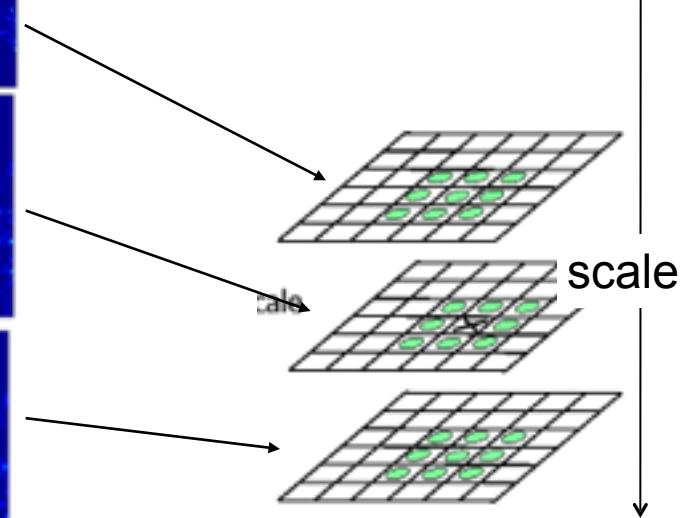
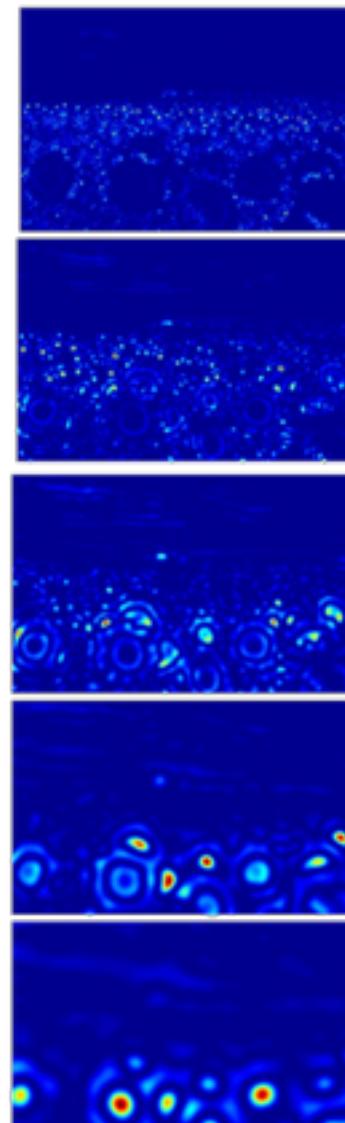
Interest points are local maxima in both position and scale.



$$L_{xx}(\sigma) + L_{yy}(\sigma)$$

↓
σ₁
↓
σ₂
↓
σ₃
↓
σ₄
↓
σ₅

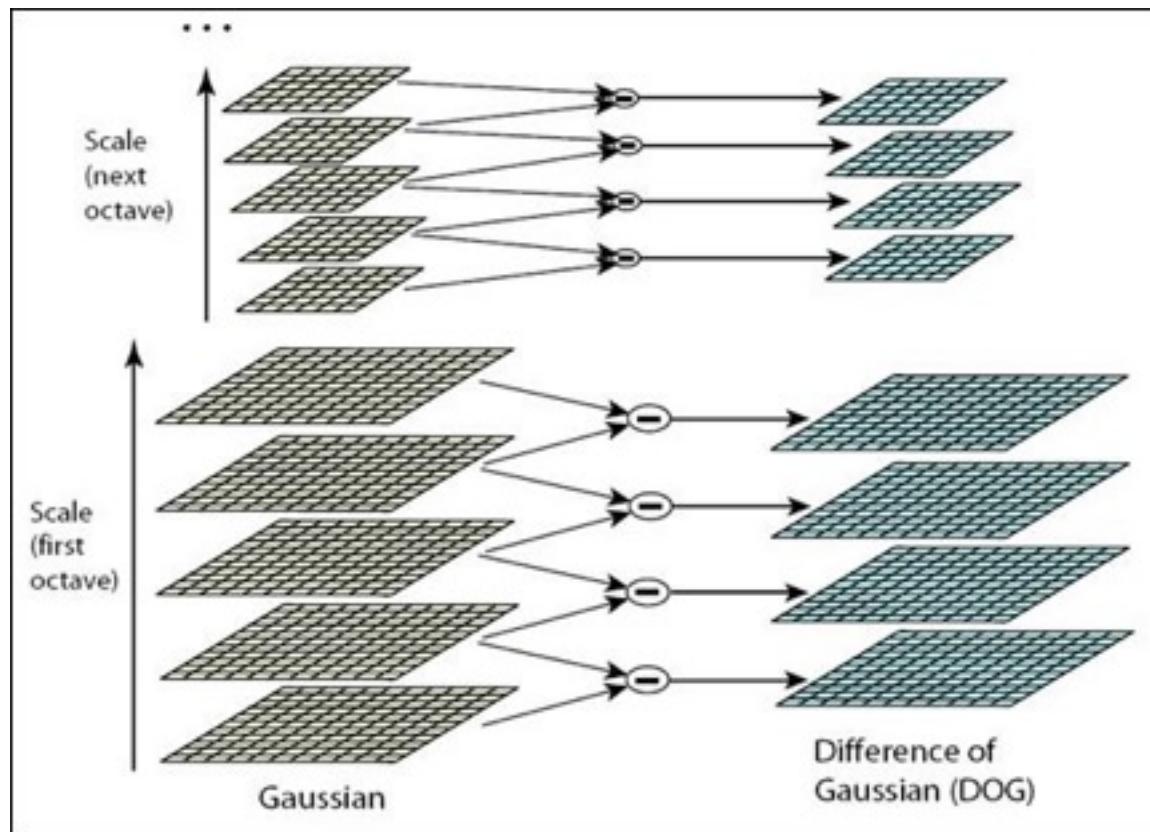
Squared filter response maps



⇒ List of
 (x, y, σ)

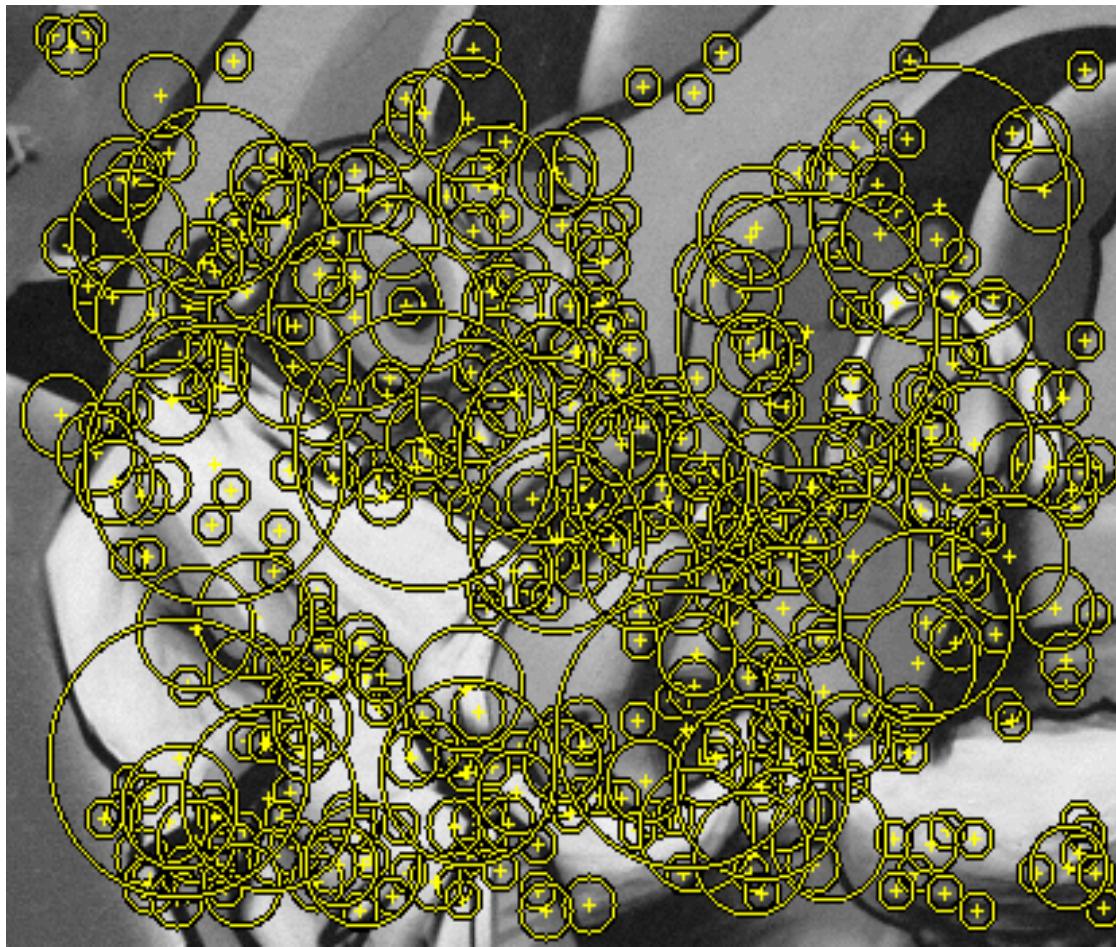
Scale

In practice the image is downsampled for larger sigmas.



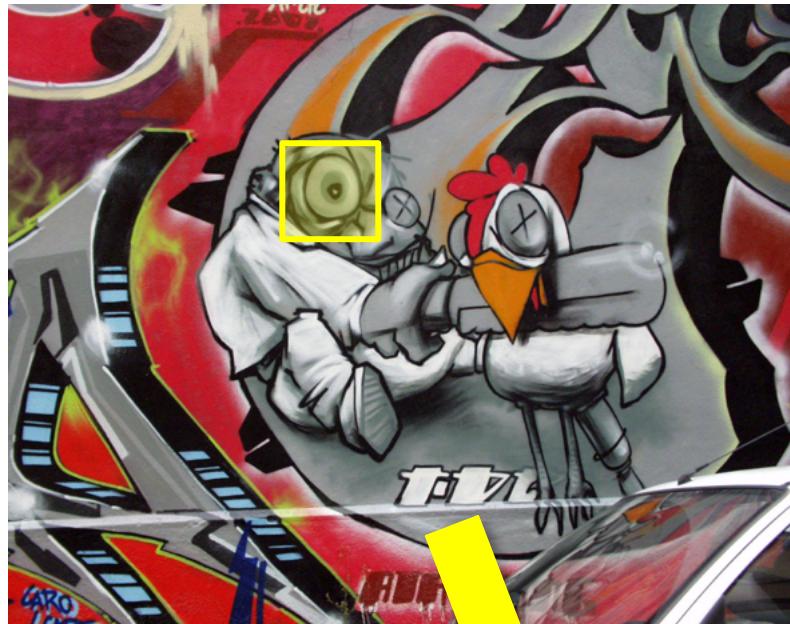
Lowe, 2004.

Results: Difference-of-Gaussian



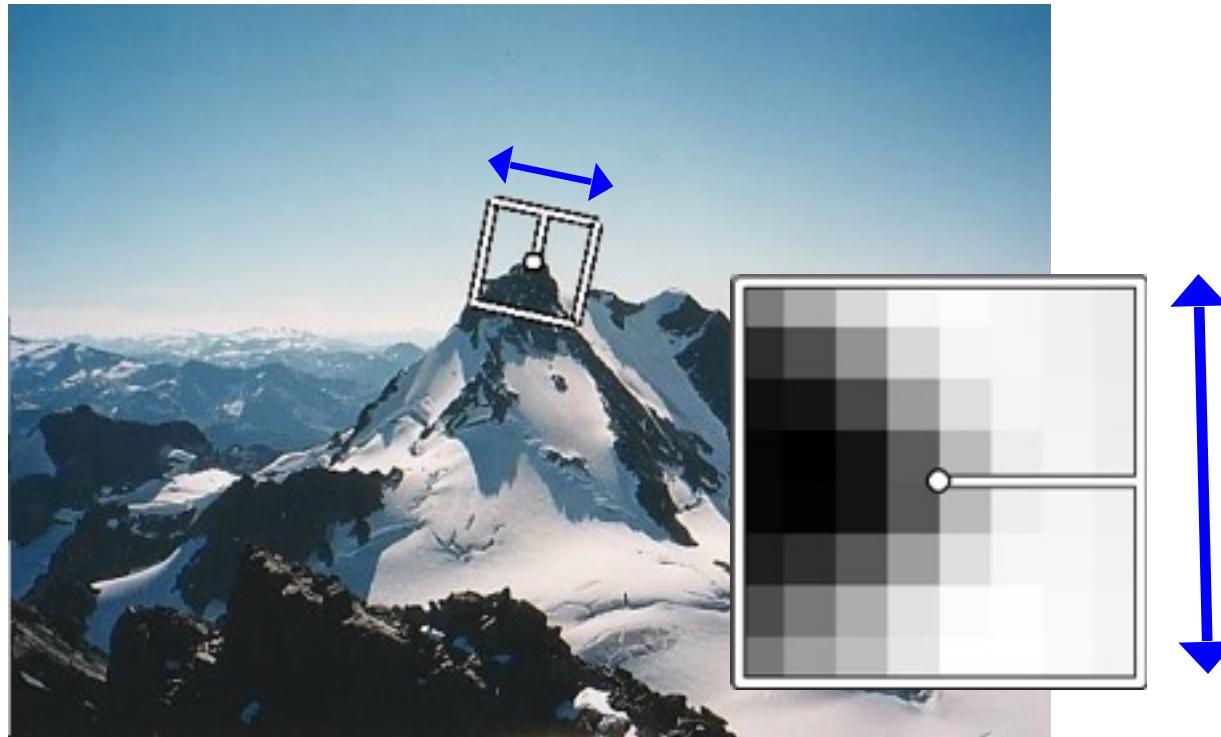
K. Grauman, B. Leibe

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.

Orientation Normalization

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

[Lowe, SIFT, 1999]

