Feature Detection & Invariance

<Vision System>

Department of Robot Engineering

Prof. Younggun Cho

Most of these slides were adapted from:

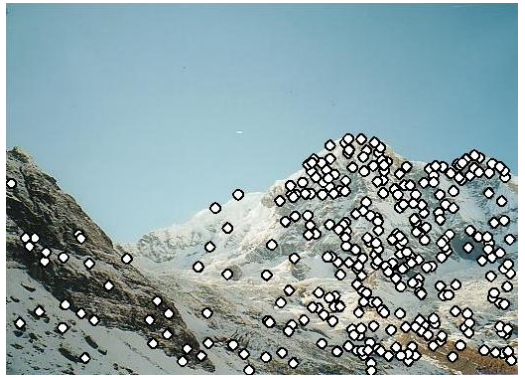
- Kris Kitani (15-463, Fall 2016).

Some slides were inspired or taken from:

- Fredo Durand (MIT).
- James Hays (Georgia Tech).
- Ionannis Gkioulekas (CMU).

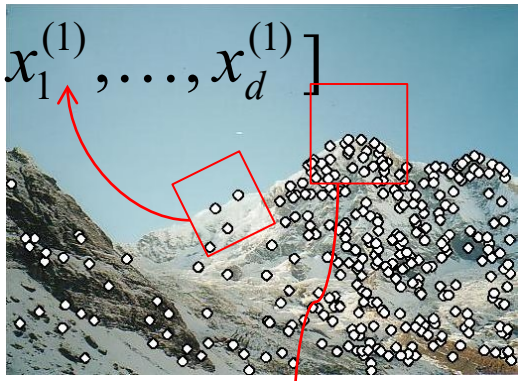
Local features: main components

1) **Detection:** Identify the interest points



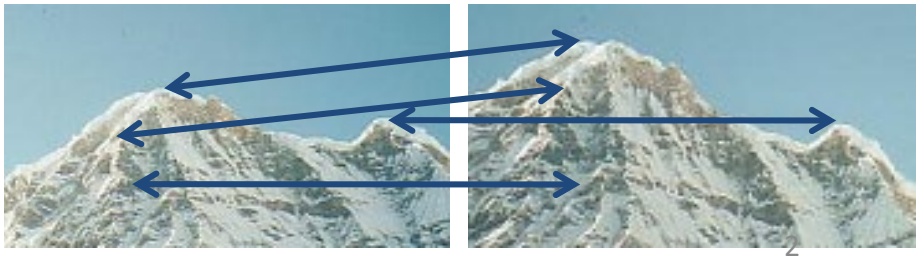
2) **Description**: Extract vector feature descriptor surrounding each interest point.

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

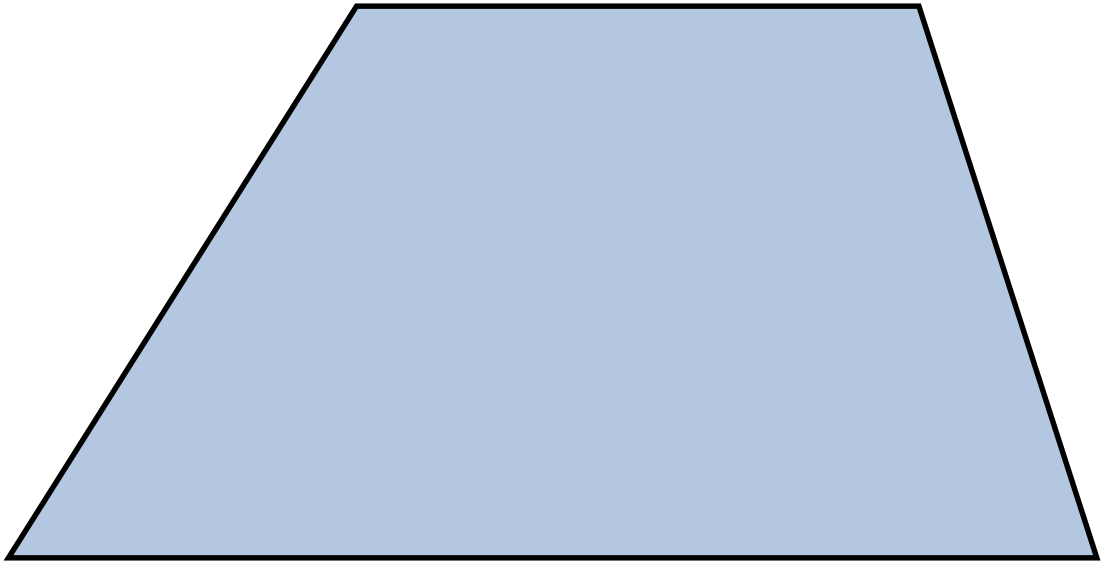


$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) **Matching:** Determine correspondence between descriptors in two views



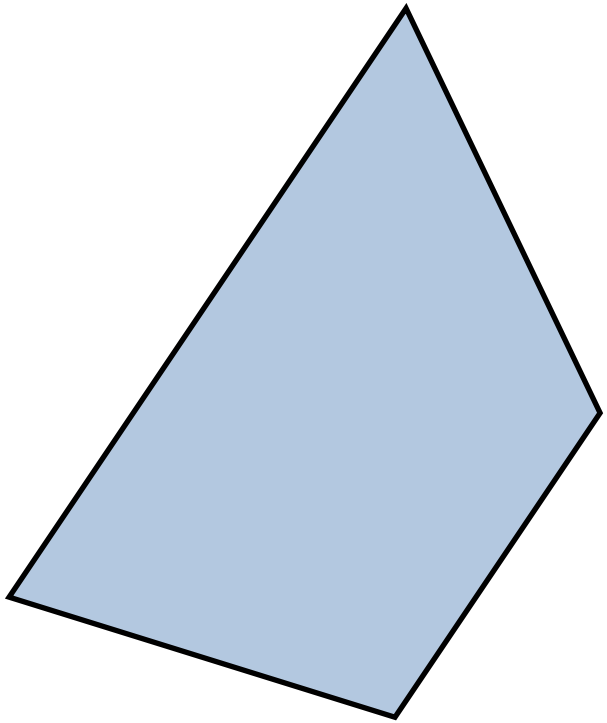
Why feature?



Pick a point in the image.
Find it again in the next image.

What type of feature would you select?

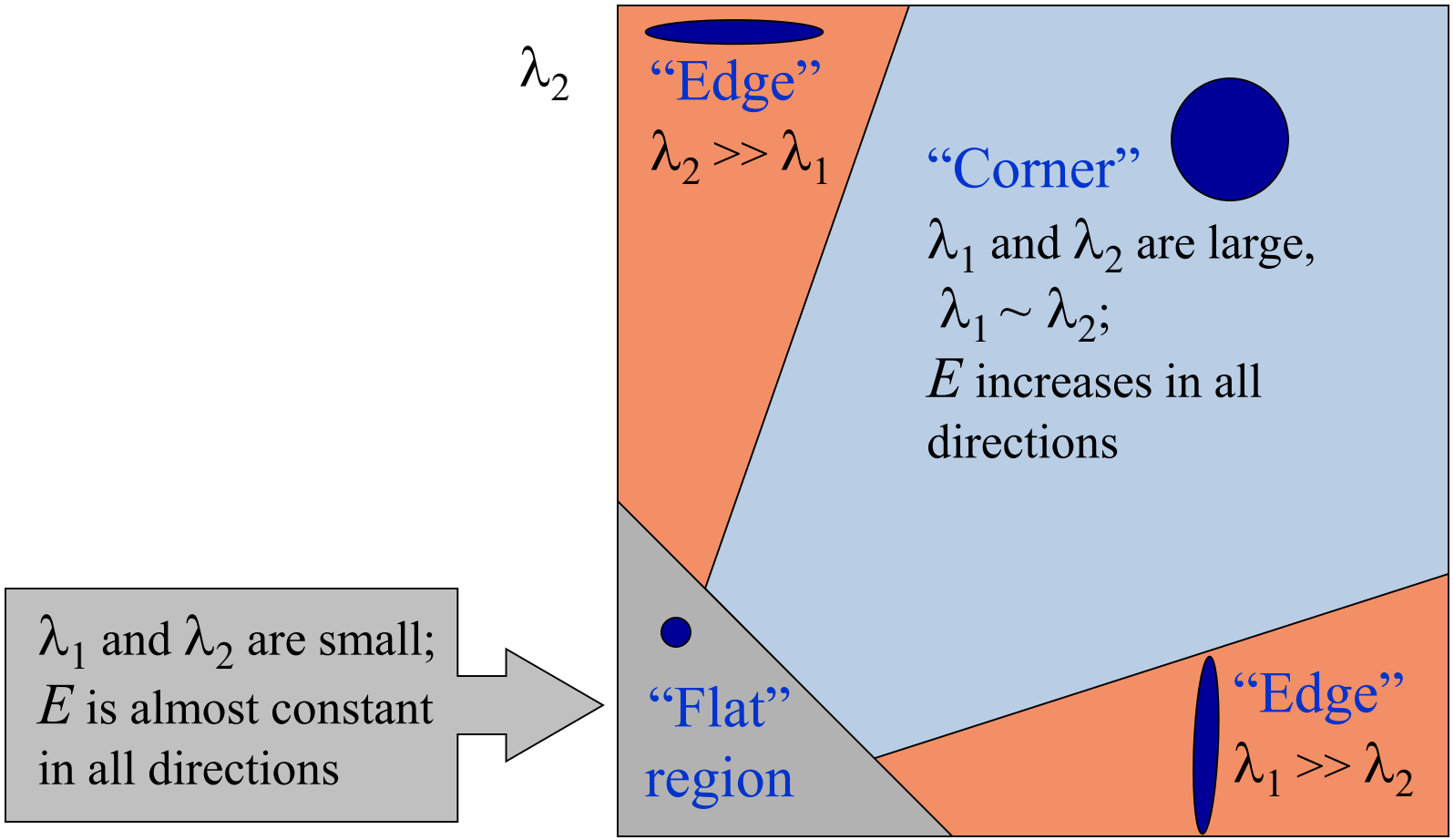
Why feature?



Pick a point in the image.
Find it again in the next image.

What type of feature would you select?

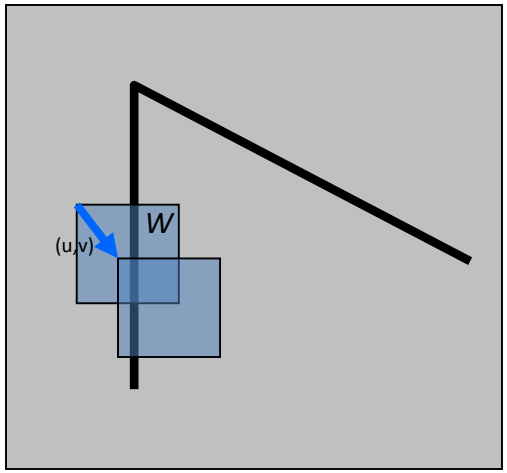
What we studied last week.



Corner detection: the math

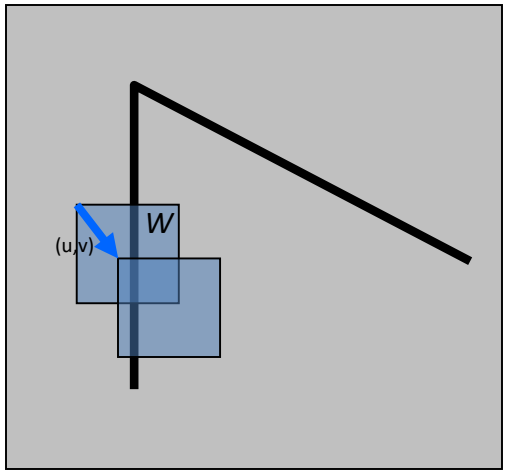
Consider shifting the window *W* by (*u,v*)

- define an SSD "error" *E(u,v)*:



$$
\begin{aligned}
E(u,v) &= \sum_{(x,y)\in W} [I(x+u,y+v) - I(x,y)]^2 \\
&\approx \sum_{(x,y)\in W} [I(x,y) + I_x u + I_y v - I(x,y)]^2 \\
&\approx \sum_{(x,y)\in W} [I_x u + I_y v]^2
\end{aligned}
$$

Corner detection: the math



Consider shifting the window *W* by (*u,v*)

- define an SSD "error" *E(u,v)*:

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

$$\approx Au^2 + 2Buv + Cv^2$$

$$A = \sum_{(x,y)\in W} I_x^2 \qquad B = \sum_{(x,y)\in W} I_x I_y \qquad C = \sum_{(x,y)\in W} I_y^2$$

- Thus, *E*(*u,v*) is locally approximated as a quadratic error function

We will need this to understand the…

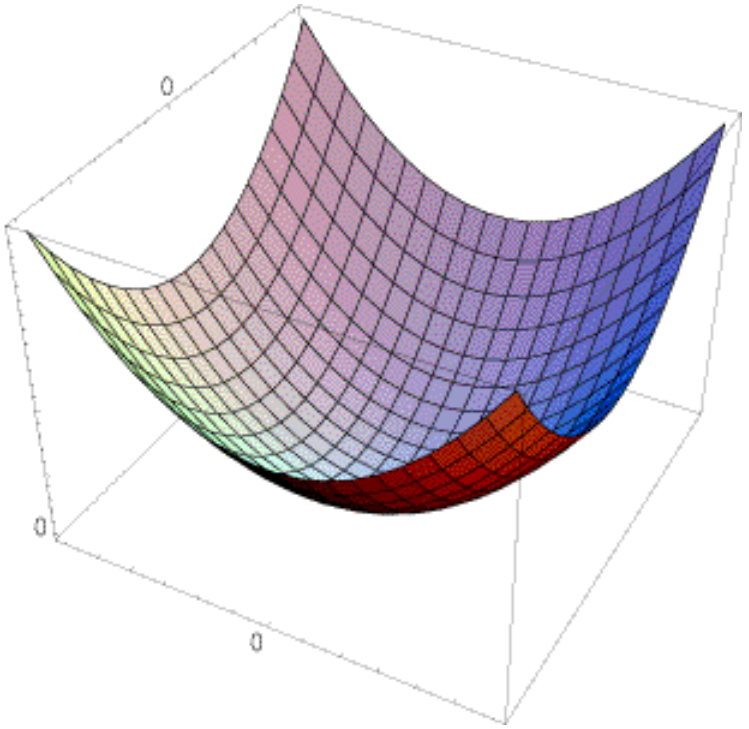
Error function for Harris Corners

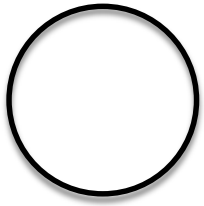
The surface $E(u,v)$ is locally approximated by a quadratic form

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

$$H = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

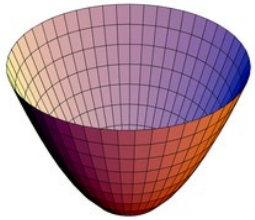
Quadratic Form





Equation of a circle

$$1 = x^2 + y^2$$



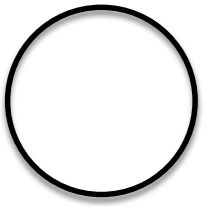
Equation of a ‘bowl’ (paraboloid)

$$f(x, y) = x^2 + y^2$$

If you slice the bowl at

$$f(x, y) = 1$$

what do you get?



$$f(x,y) = x^2 + y^2$$

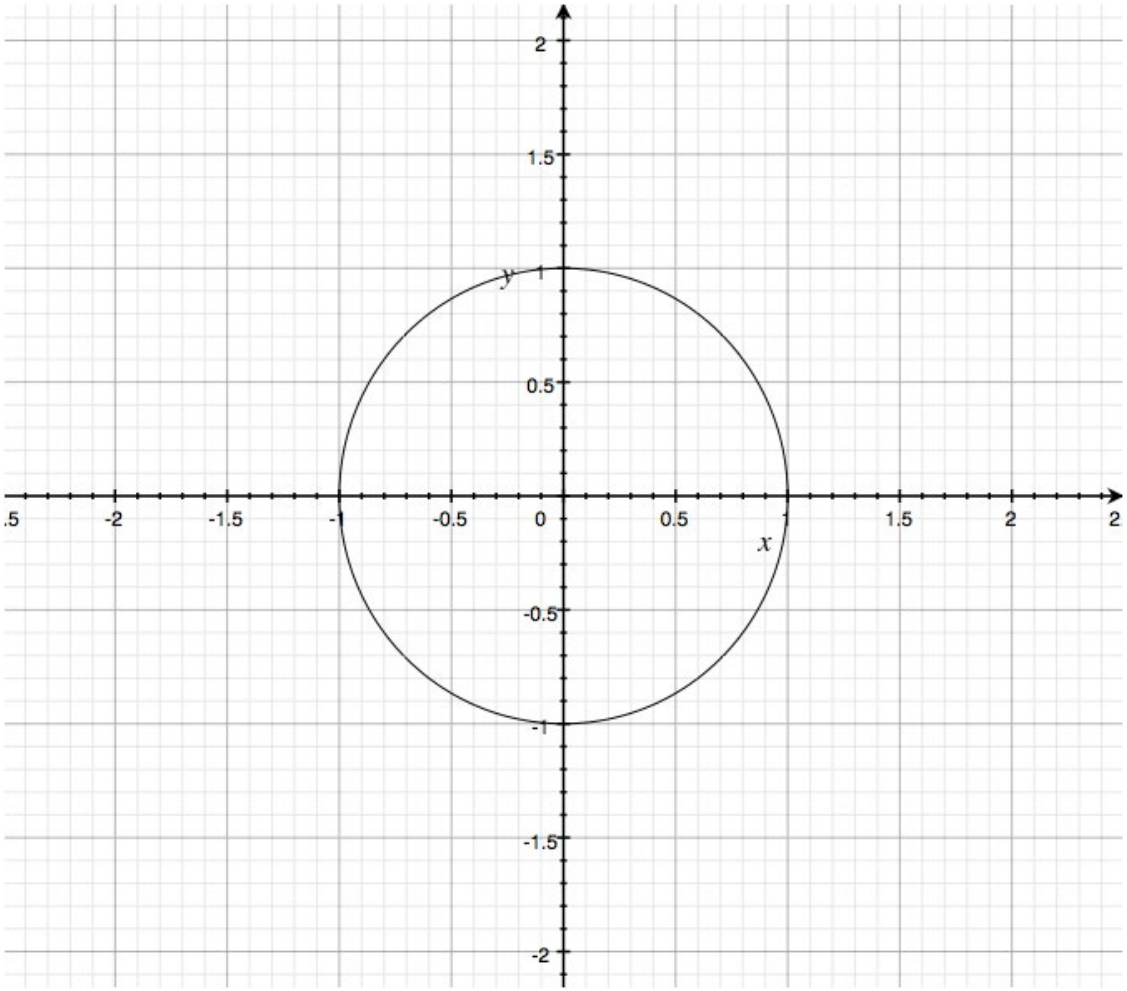
can be written in matrix form like this…

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Recap: Quadratic Form

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

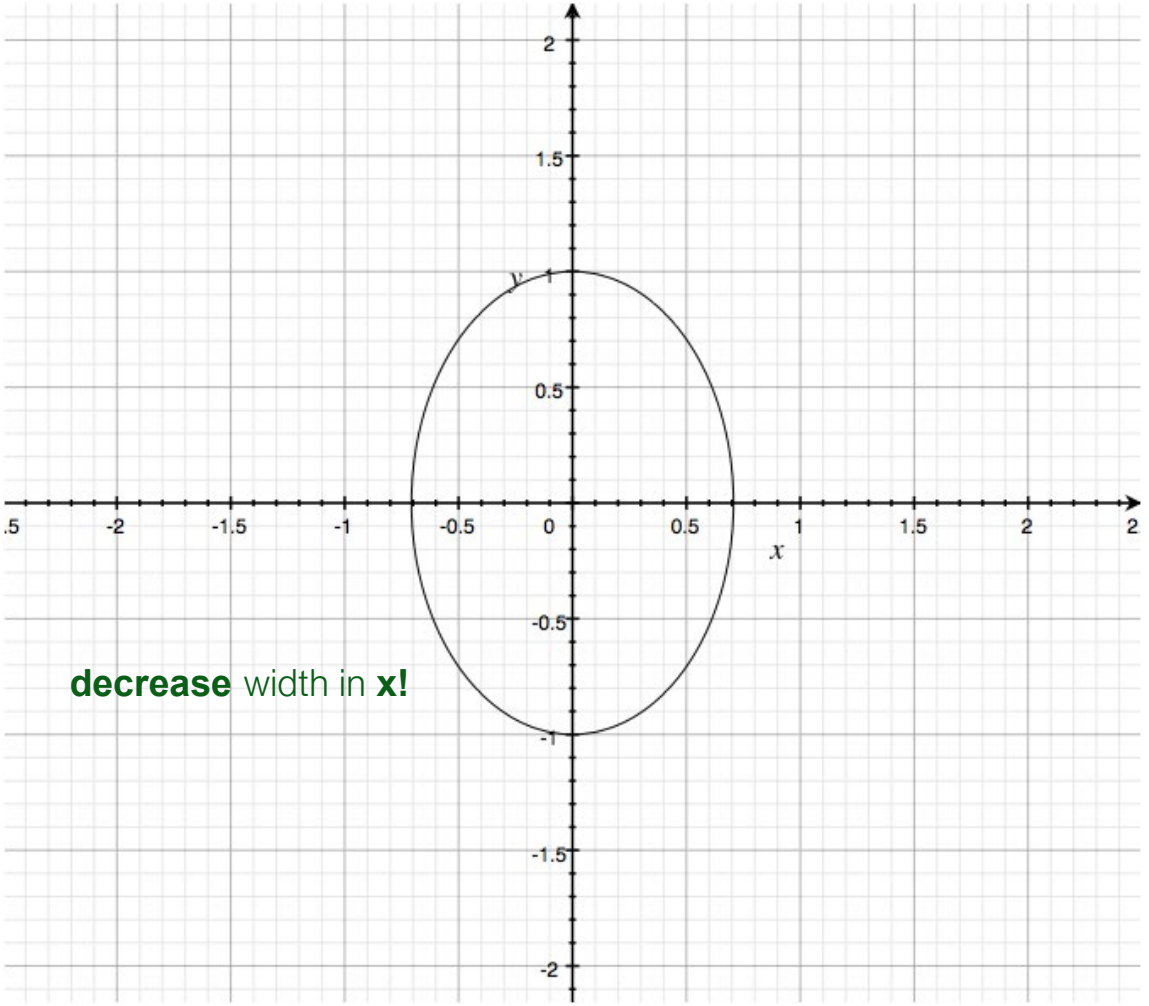
‘sliced at 1’



*What happens if you **increase** coefficient on **x**?*

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and slice at 1

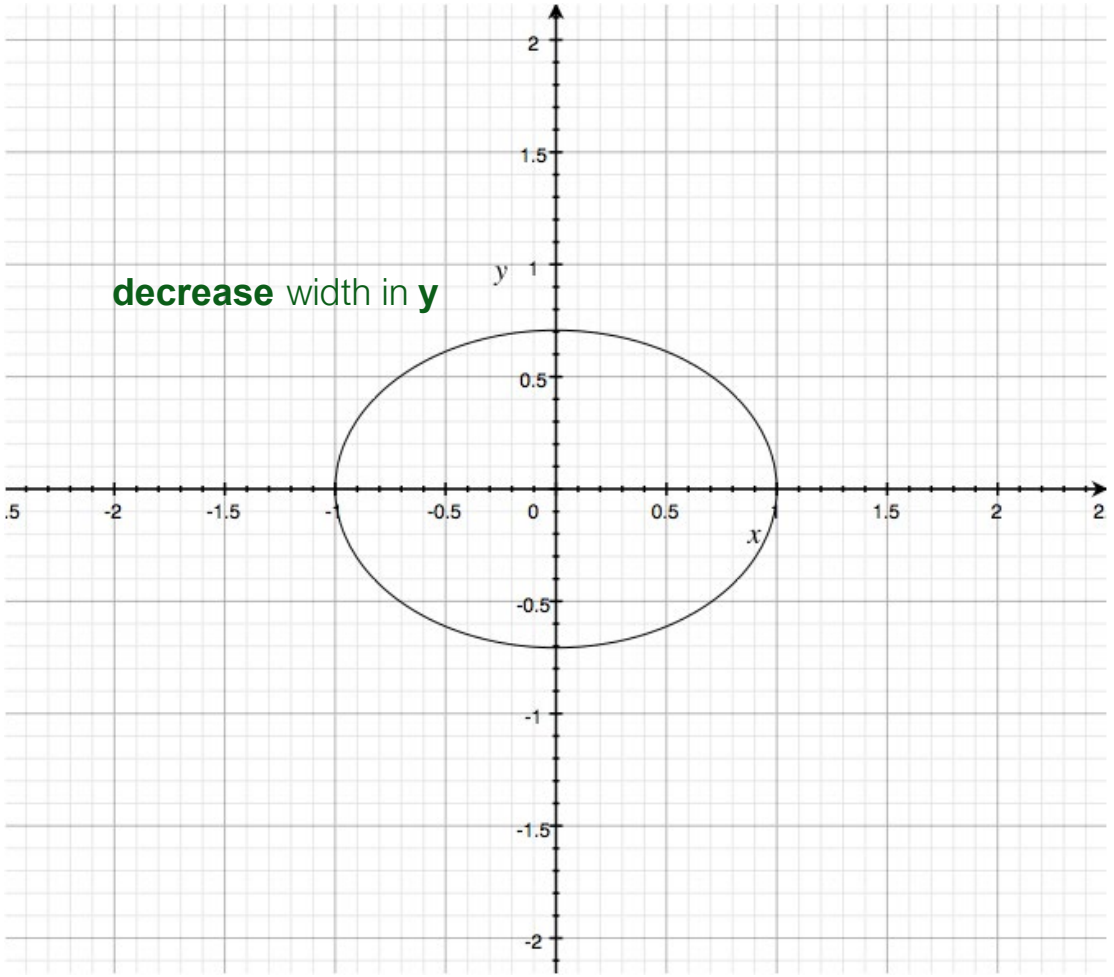


decrease width in **x!**

*What happens if you **increase** coefficient on **y**?*

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and slice at 1



$$f(x,y) = x^2 + y^2$$

can be written in matrix form like this…

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Result of Singular Value Decomposition (SVD)

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^\top$$

eigenvectors

axis of the ‘ellipse slice’

eigenvalues along diagonal

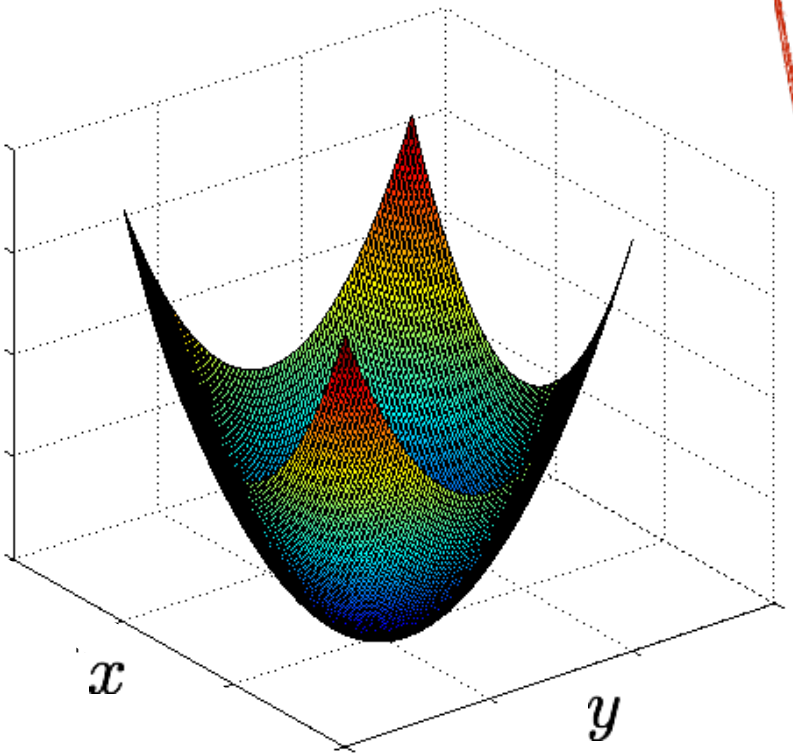
Inverse sqr of length of the quadratic along the axis

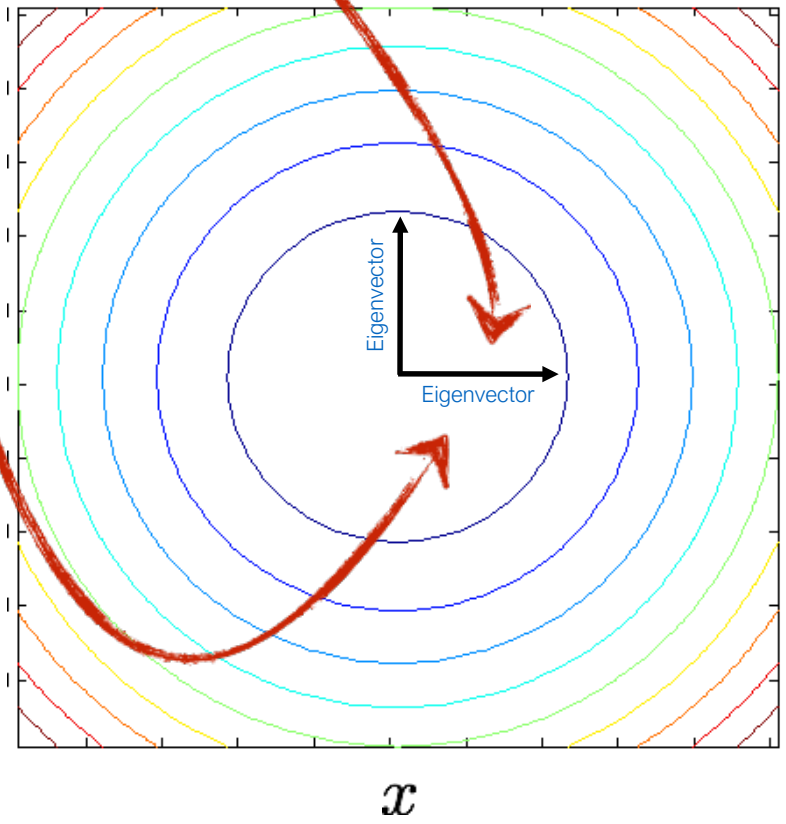
Eigenvectors Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

Eigenvectors

Inverse sqr of the size of the axis





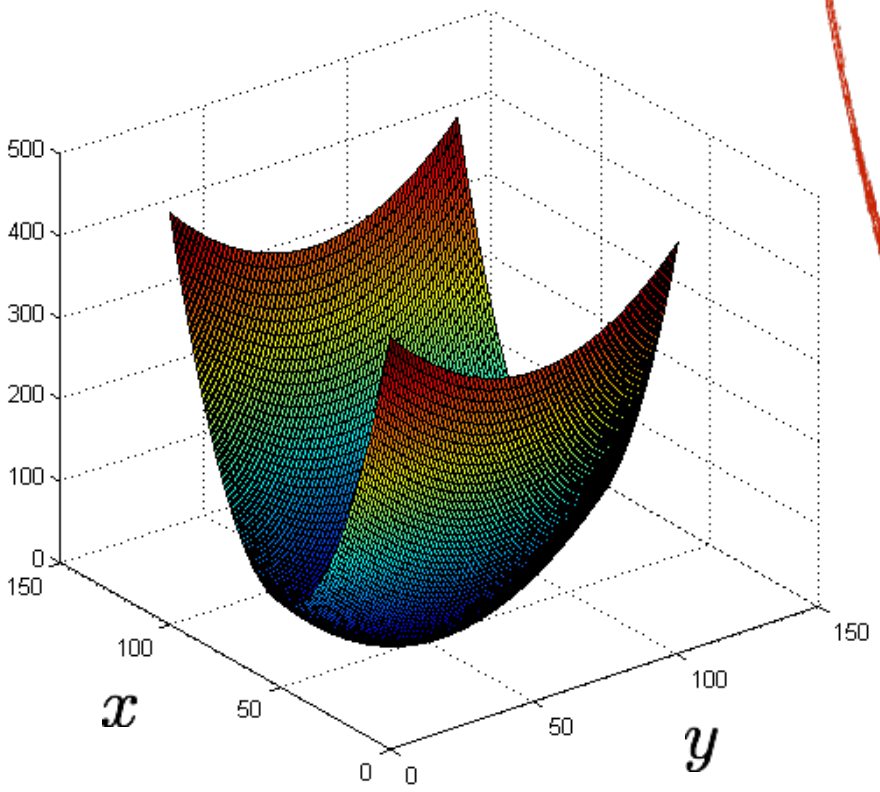
$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

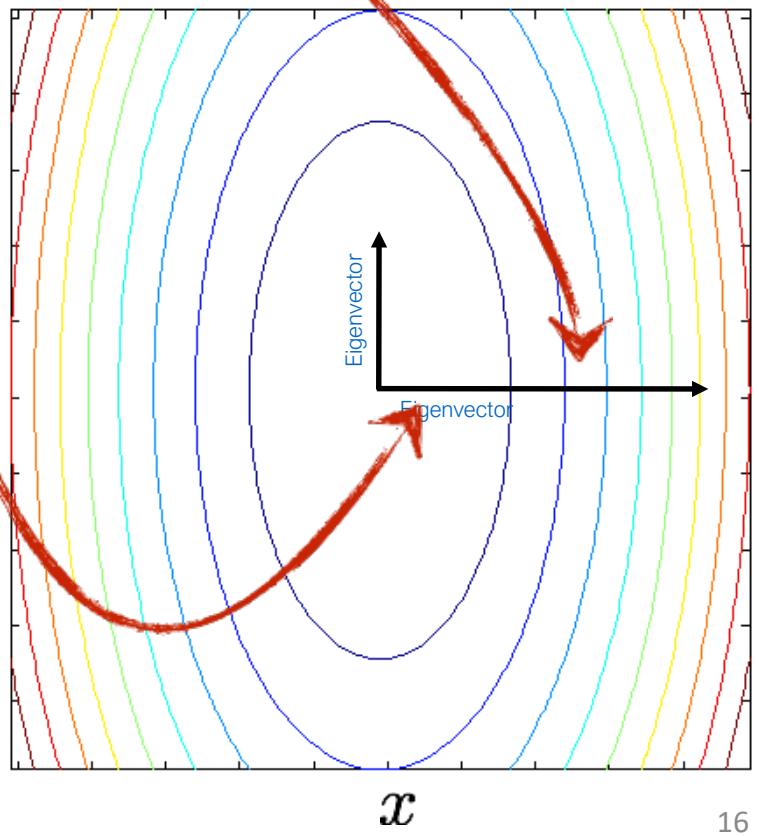
Eigenvalues

Eigenvectors

Eigenvectors

Inverse sqr of length of axis



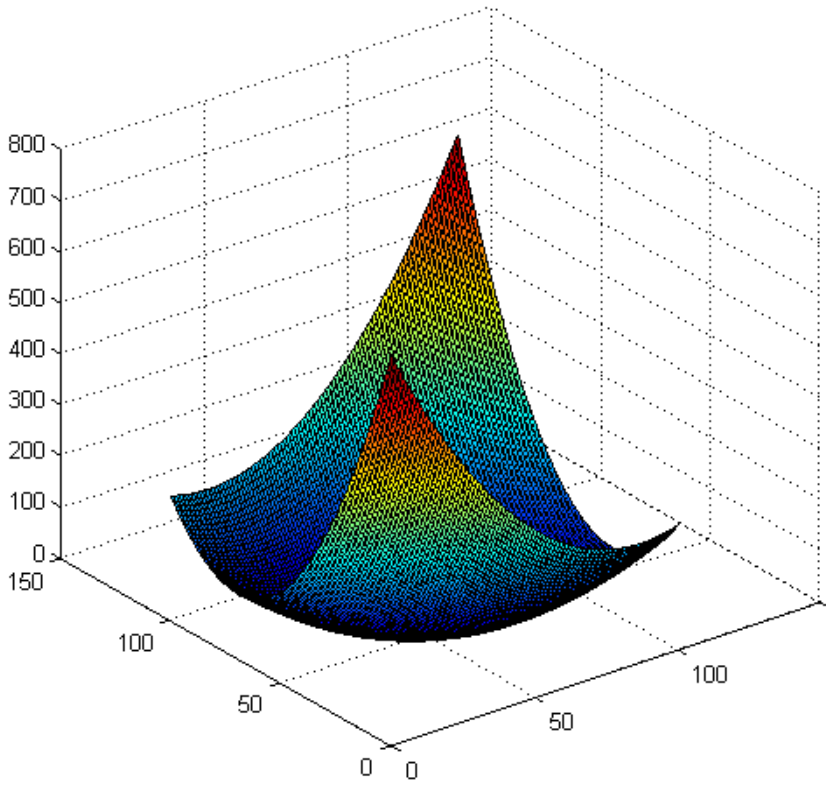


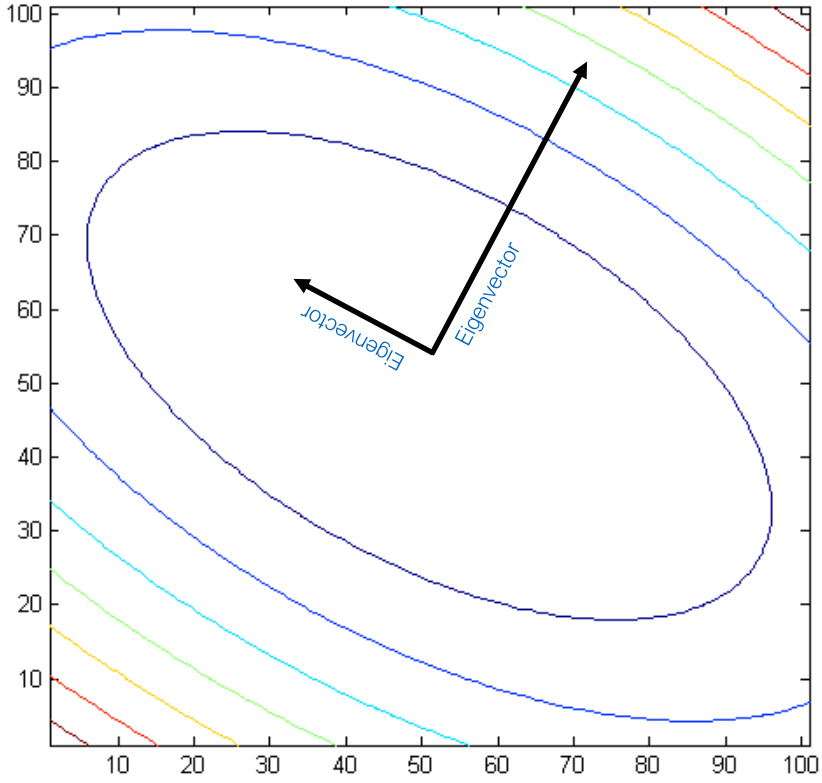
Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 3.25 & 1.30 \\ 1.30 & 1.75 \end{bmatrix} = \begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}^T$$

Eigenvectors

Eigenvectors



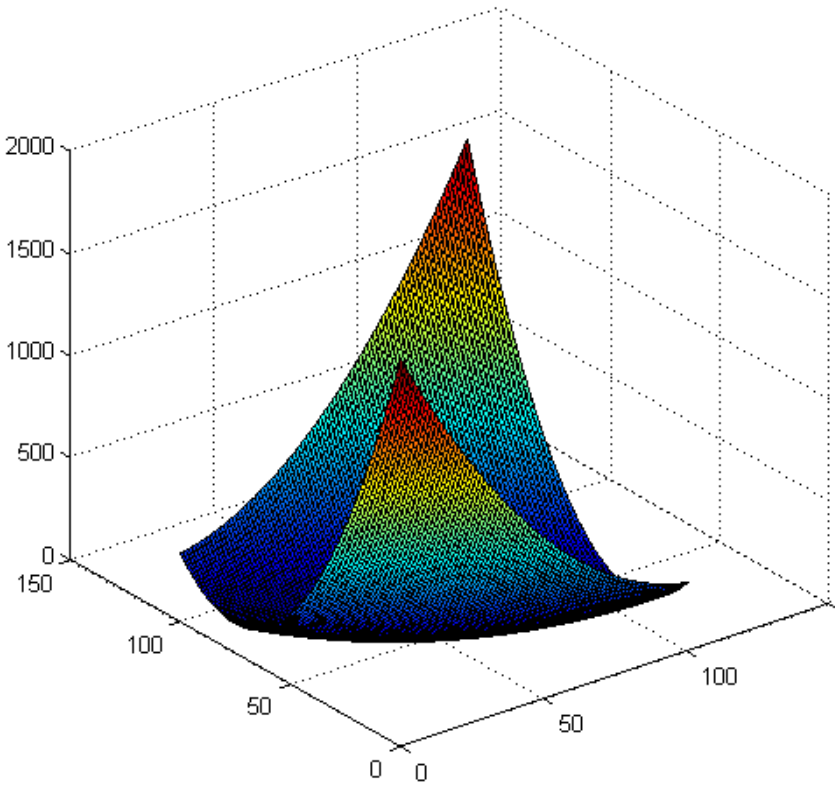


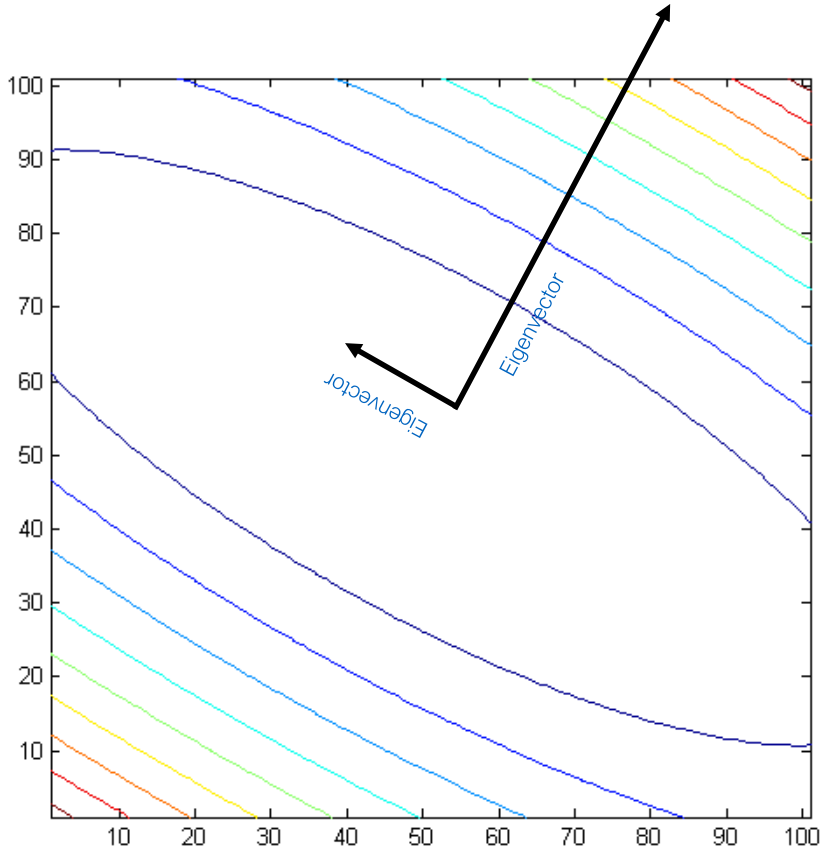
Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 7.75 & 3.90 \\ 3.90 & 3.25 \end{bmatrix} = \begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}^T$$

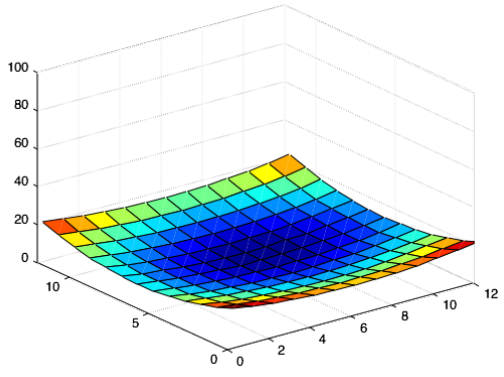
Eigenvectors

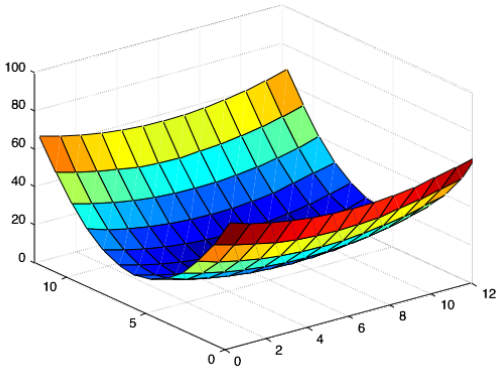
Eigenvectors

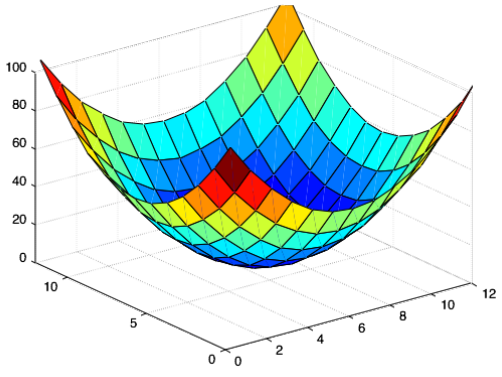




Which error surface indicates a good image feature?







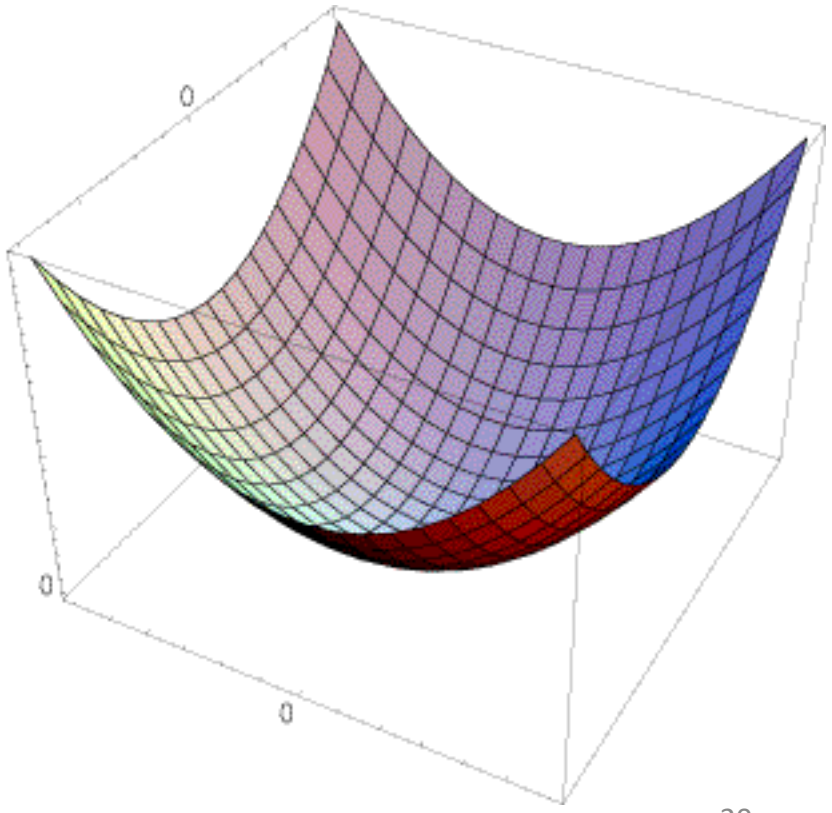
What kind of image patch do these surfaces represent?

Visualization of a quadratic

The surface $E(u,v)$ is locally approximated by a quadratic form

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

$$H = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

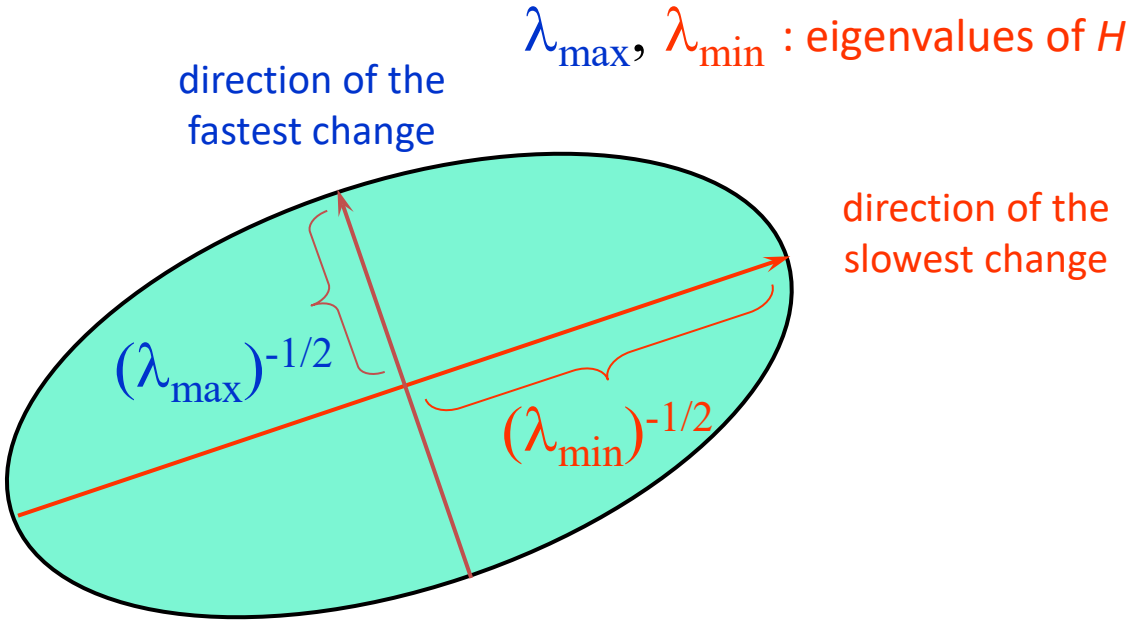


General case

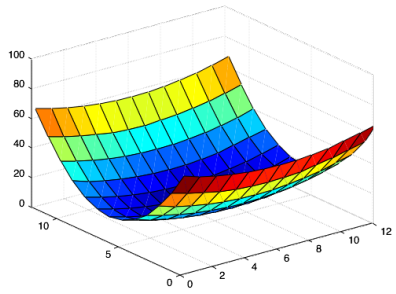
We can visualize *H* as an ellipse with axis lengths determined by the *eigenvalues* of *H* and orientation determined by the *eigenvectors* of *H*

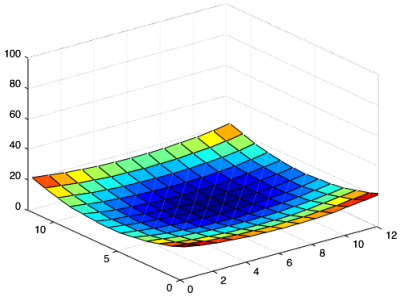
Ellipse equation:

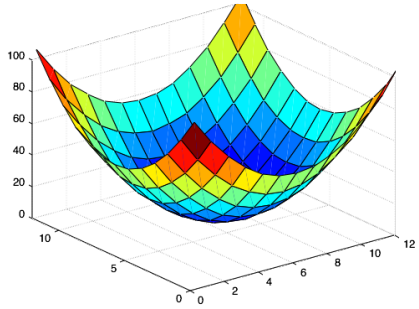
$$[u \quad v] \quad H \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

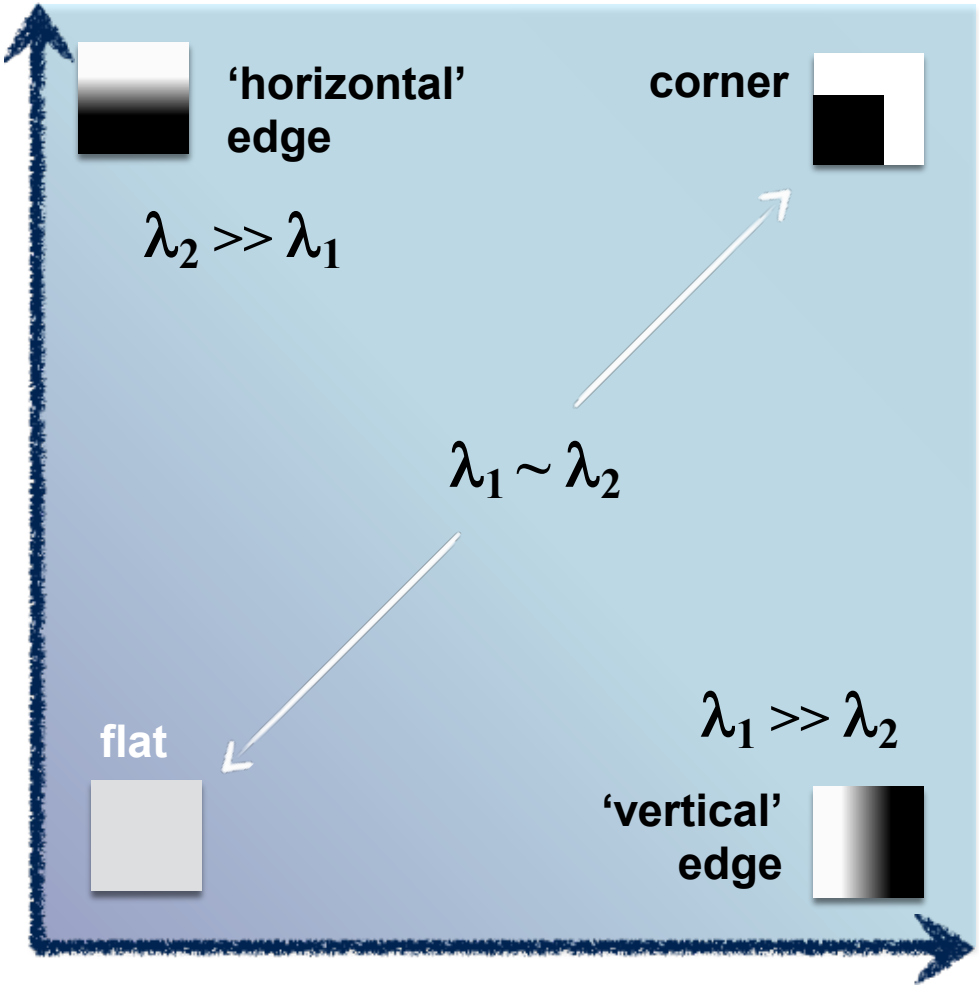


interpreting eigenvalues





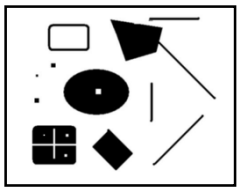




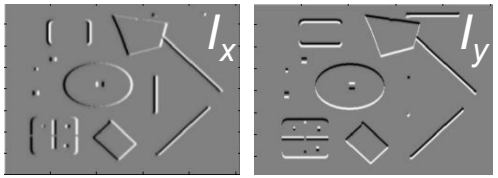
Harris Detector [Harris88]

- Second moment matrix

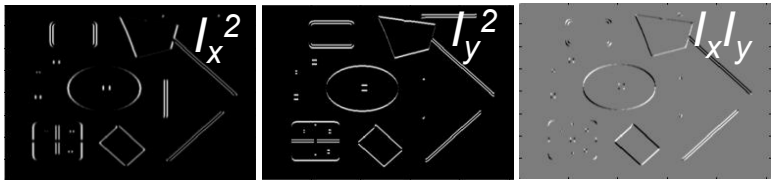
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



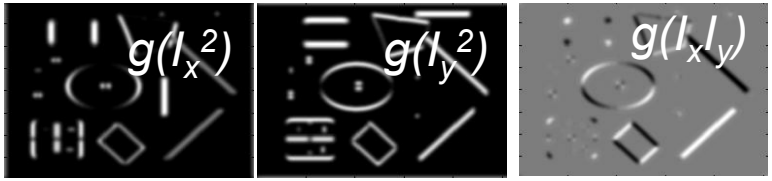
1. Image derivatives (optionally, blur first)



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$



$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$$= \frac{determinant(H)}{trace(H)}$$

$$\det M = \lambda_1 \lambda_2$$

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

4. Cornerness function – both eigenvalues are strong

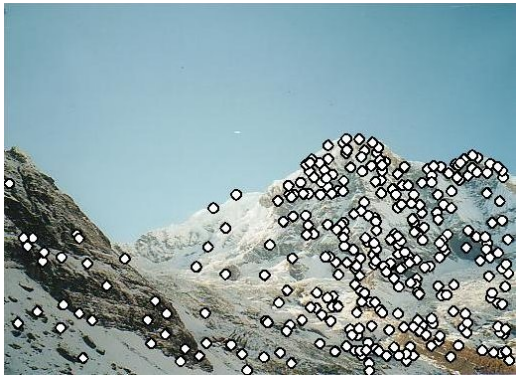
5. Non-maxima suppression

Harris features (in red)

Local features: main components

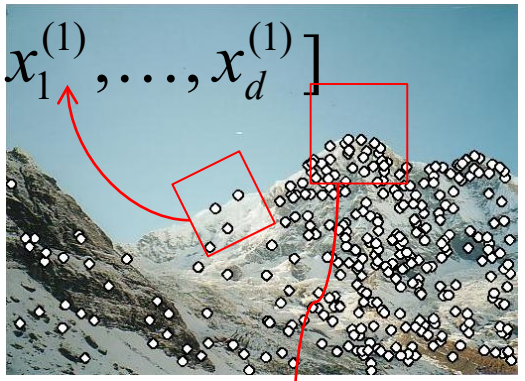
1) Detection: Identify the interest points

What is a good detector?



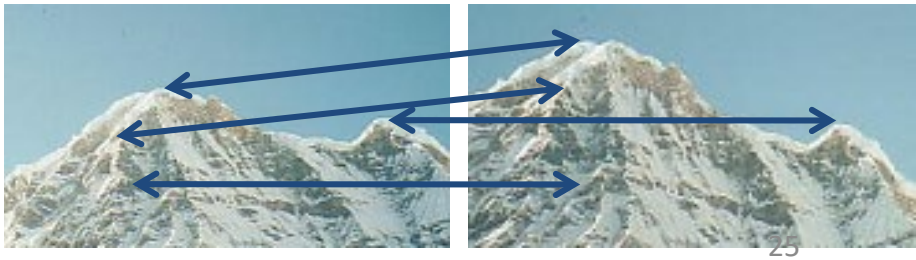
2) Description: Extract vector feature descriptor surrounding each interest point.

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



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

3) Matching: Determine correspondence between descriptors in two views



Local Features

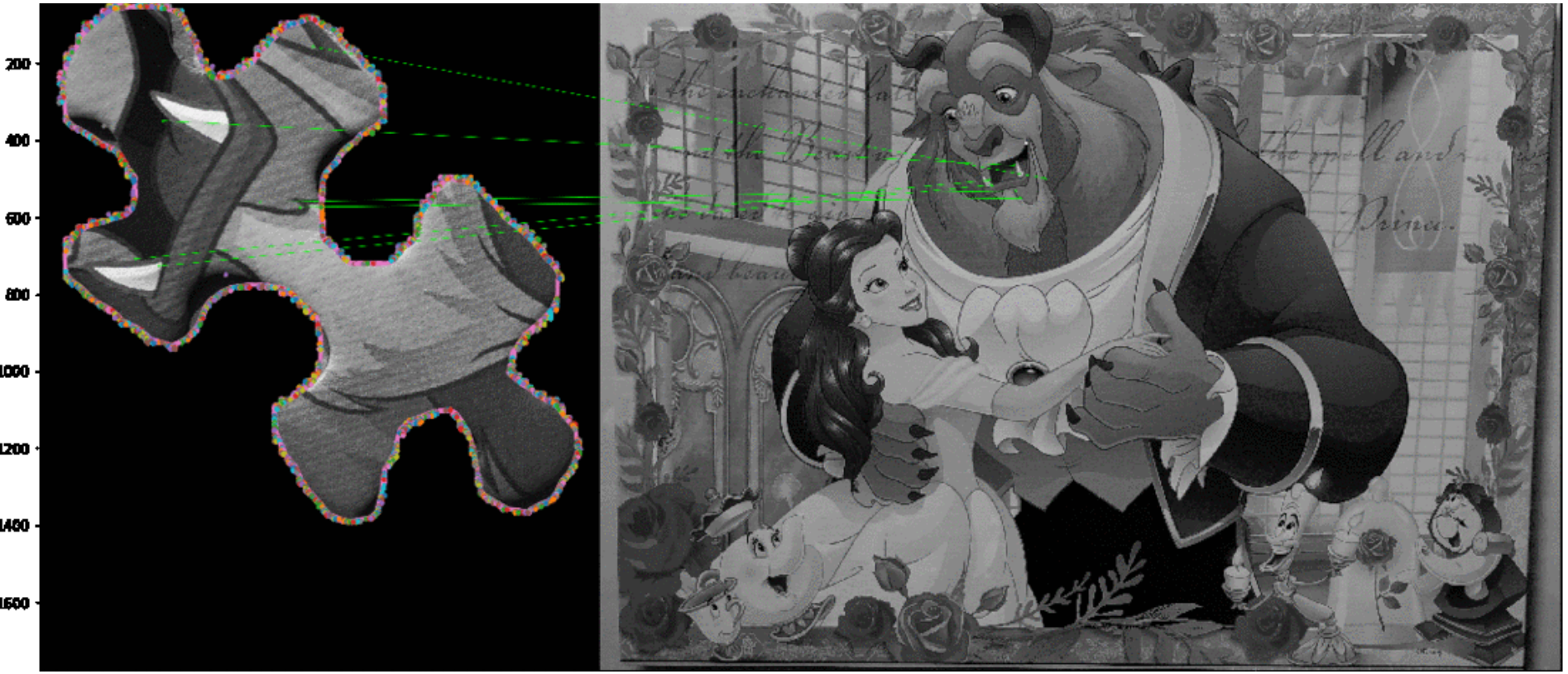
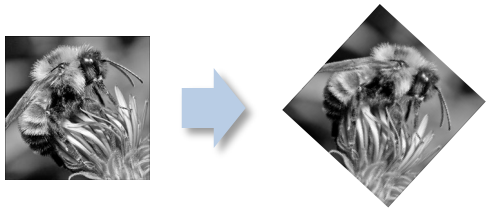


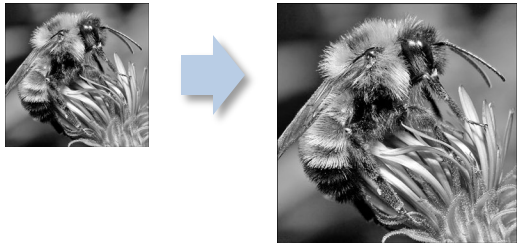
Image transformations

- Geometric

Rotation

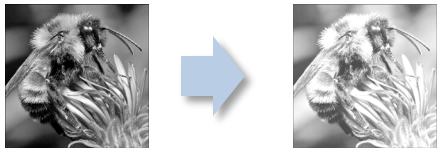


Scale



- Photometric

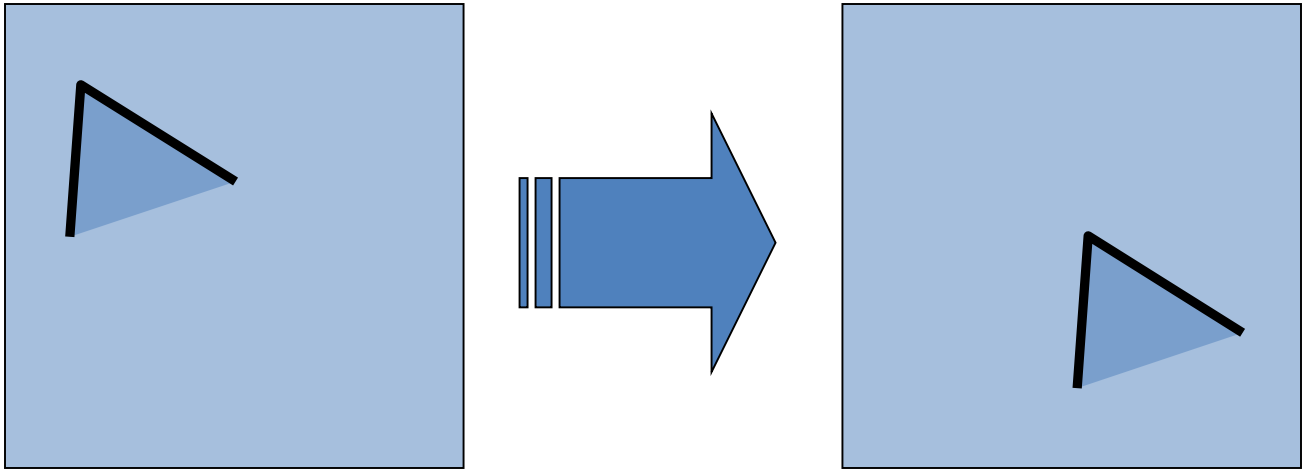
Intensity change



Invariance and equivariance

- We want corner locations to be *invariant* to photometric transformations and *equivariant* to geometric transformations
  - **Invariance:** image is transformed and corner locations do not change
  - **Equivariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations
  - (Sometimes “invariant” and “equivariant” are both referred to as “invariant”)

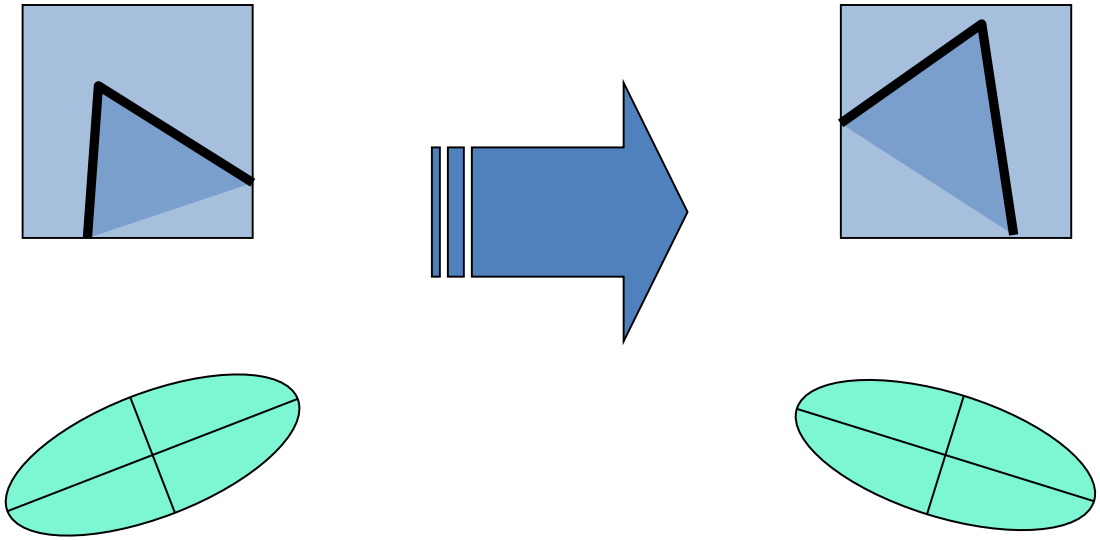
Harris detector: Invariance properties -- Image translation



- Derivatives and window function are equivariant

Corner location is equivariant w.r.t. translation

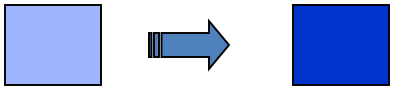
Harris detector: Invariance properties -- Image rotation



Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

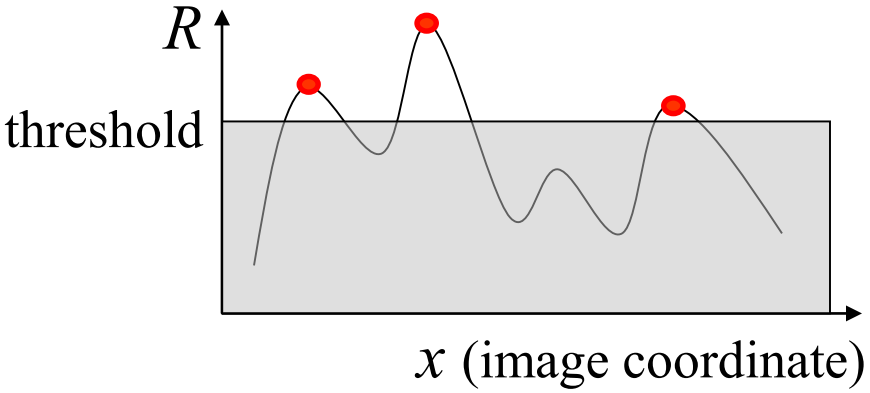
Corner location is equivariant w.r.t. image rotation

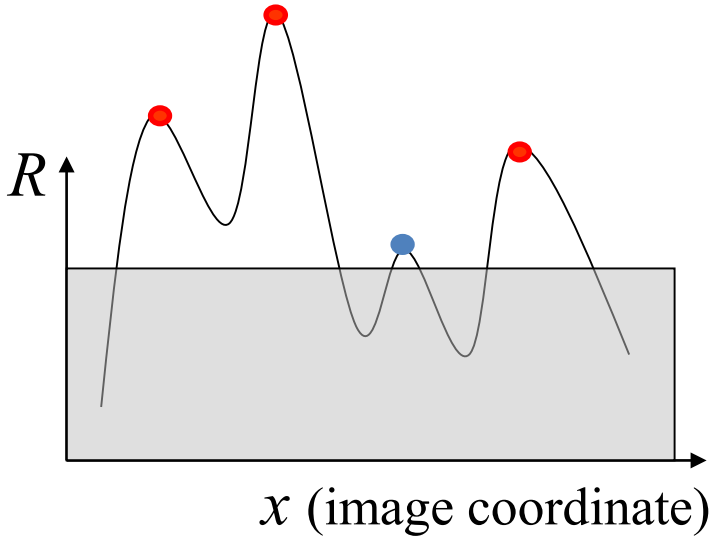
Harris detector: Invariance properties – Affine intensity change



$$I \rightarrow a\,I + b$$

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
- Intensity scaling: $I \rightarrow a\,I$

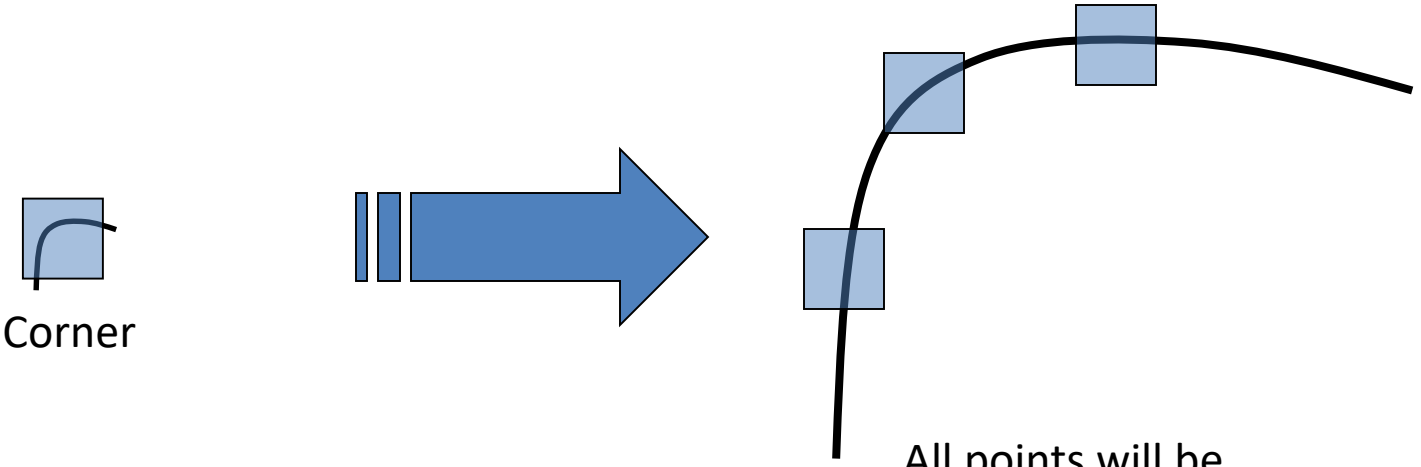




Partially invariant to affine intensity change

Harris Detector: Invariance Properties

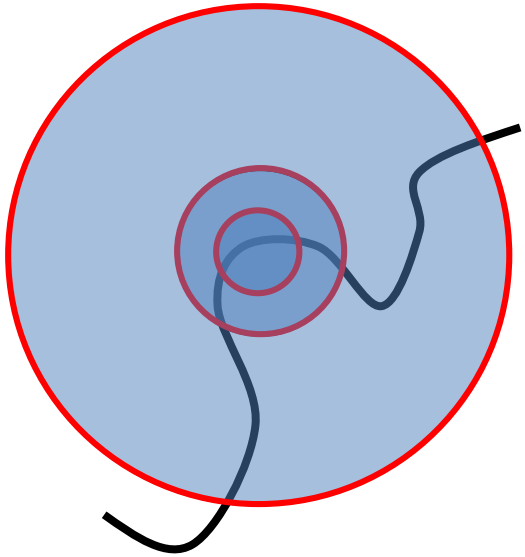
- Scaling



Neither invariant nor equivariant to scaling

Scale invariant detection

Suppose you're looking for corners

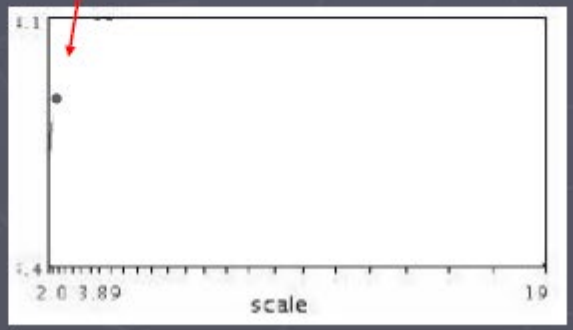


Key idea: find scale that gives local maximum of *f*

- in both position and scale
- One definition of *f*: the Harris operator

Automatic scale selection

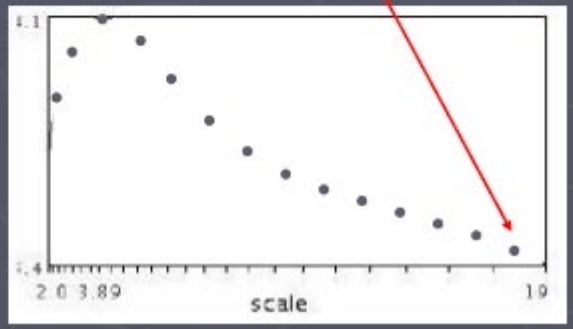
Lindeberg et al., 1996



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

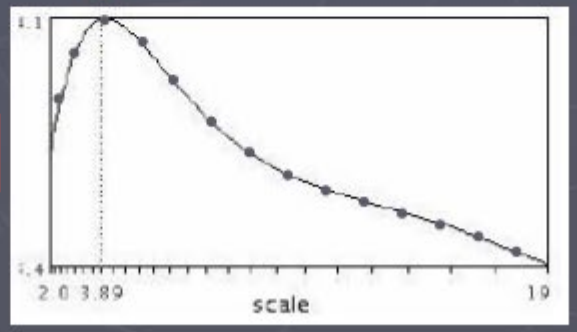
Slide from Tinne Tuytelaars

Automatic scale selection

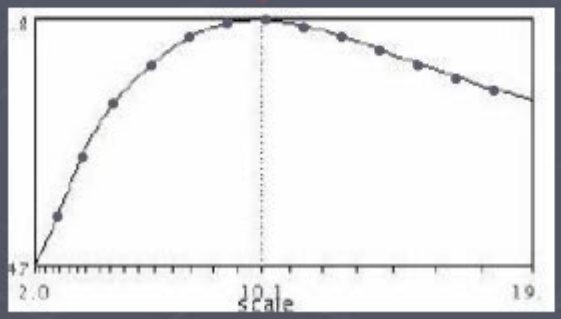


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection



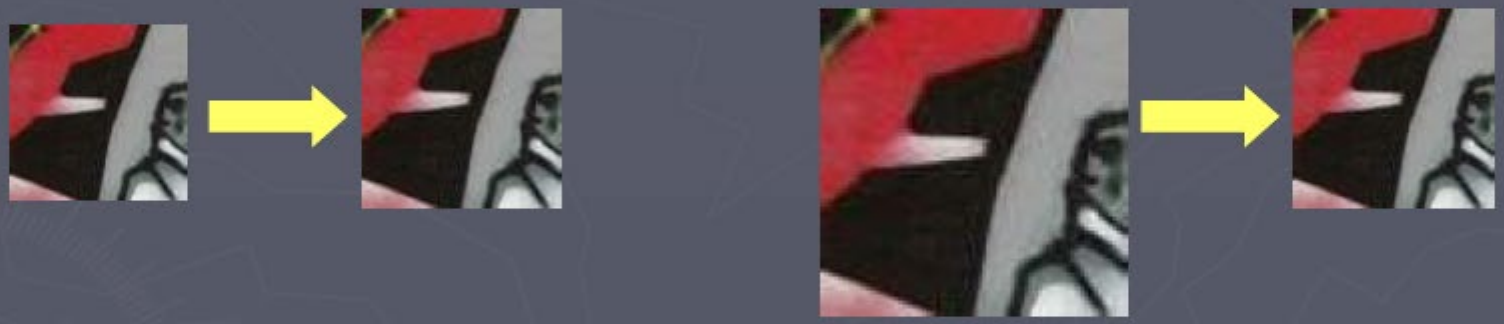
$f(I_{i_1 \ldots i_m}(x,\sigma))$



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

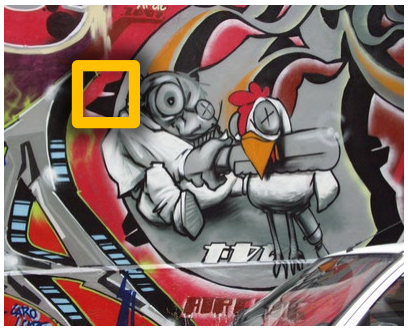
Automatic scale selection

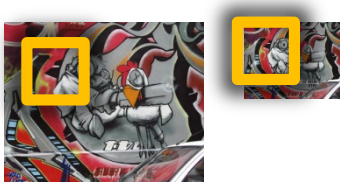
Normalize: rescale to fixed size



Implementation

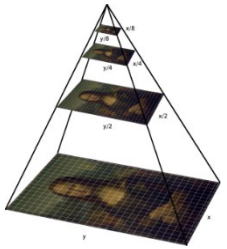
- Instead of computing f for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid



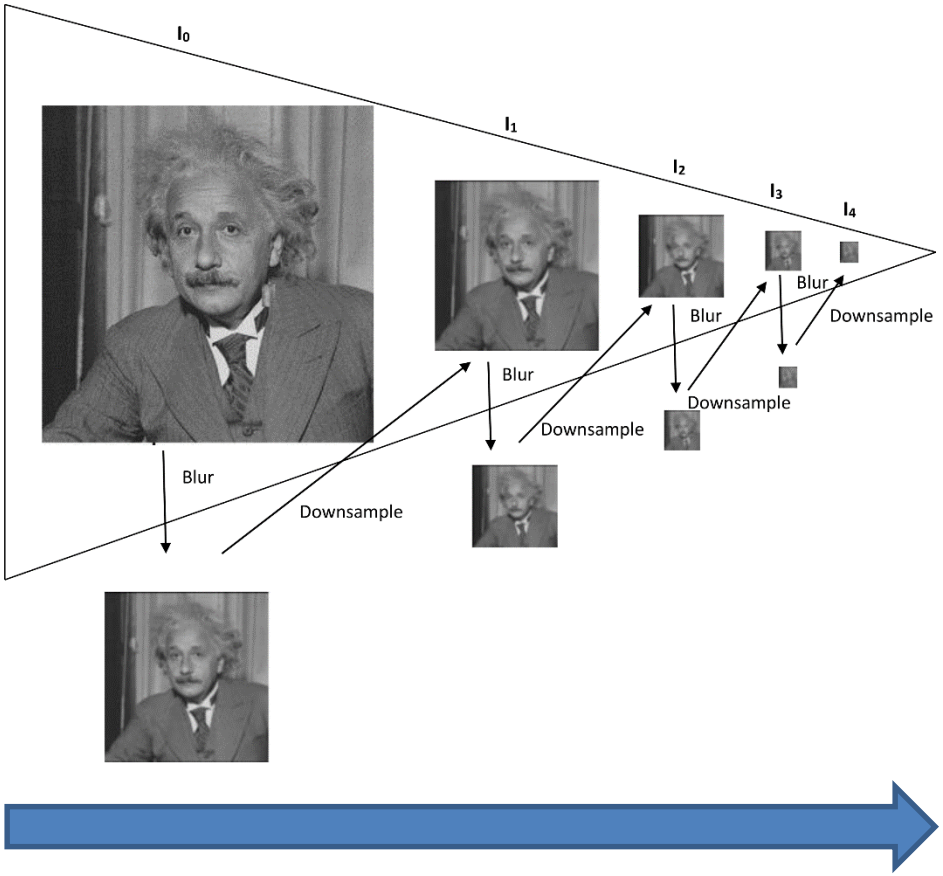


(sometimes need to create in-between levels, e.g. a ¾-size image)

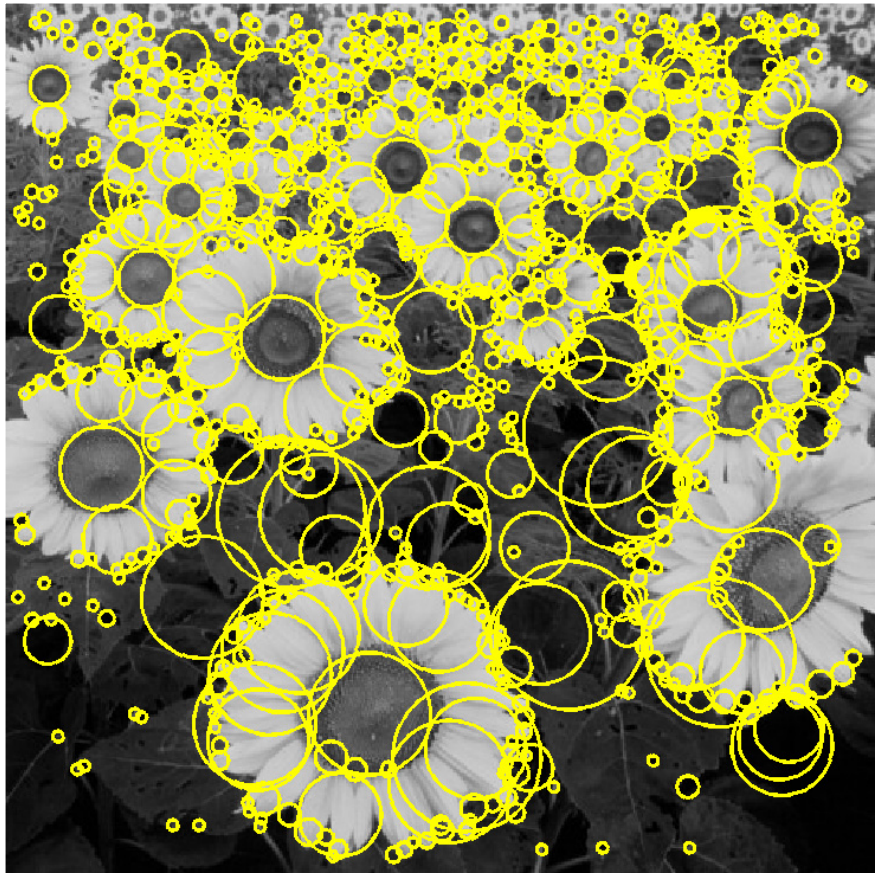
Gaussian Pyramid?



- Blur -> Subsample -> Blur -> Subsample

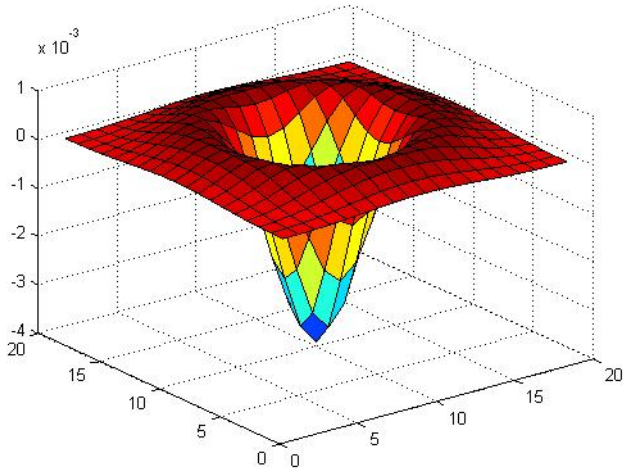


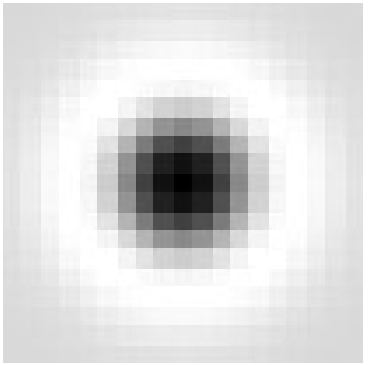
Feature extraction: Corners and blobs



Another common definition of *f*

- The *Laplacian of Gaussian* (*LoG*)



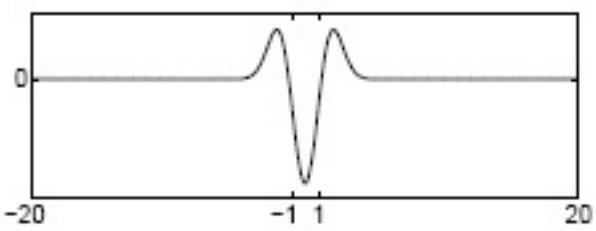


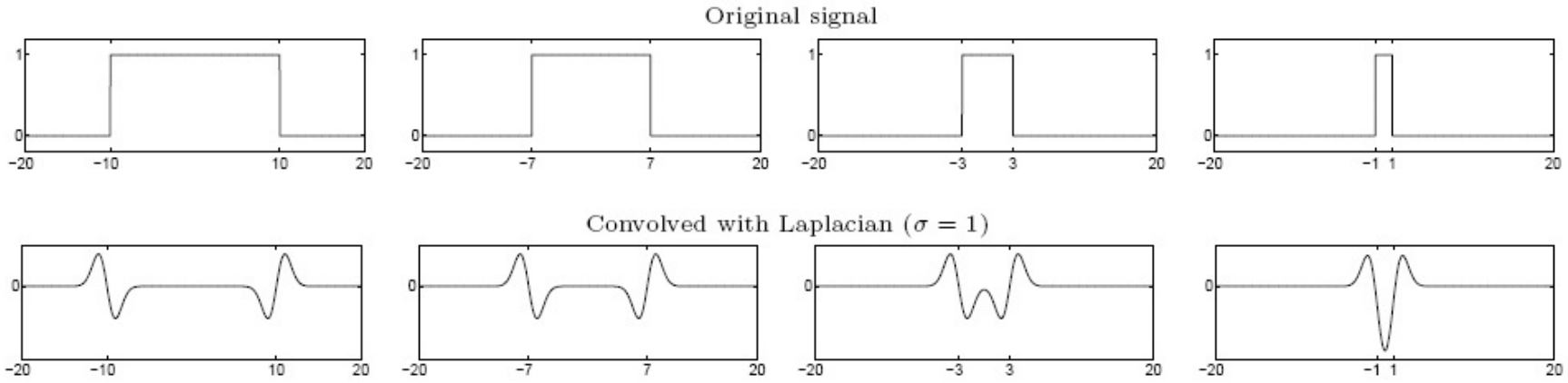
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

(very similar to a Difference of Gaussians (DoG) – i.e. a Gaussian minus a slightly smaller Gaussian)

Formally…

Laplacian filter

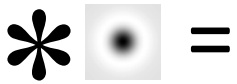


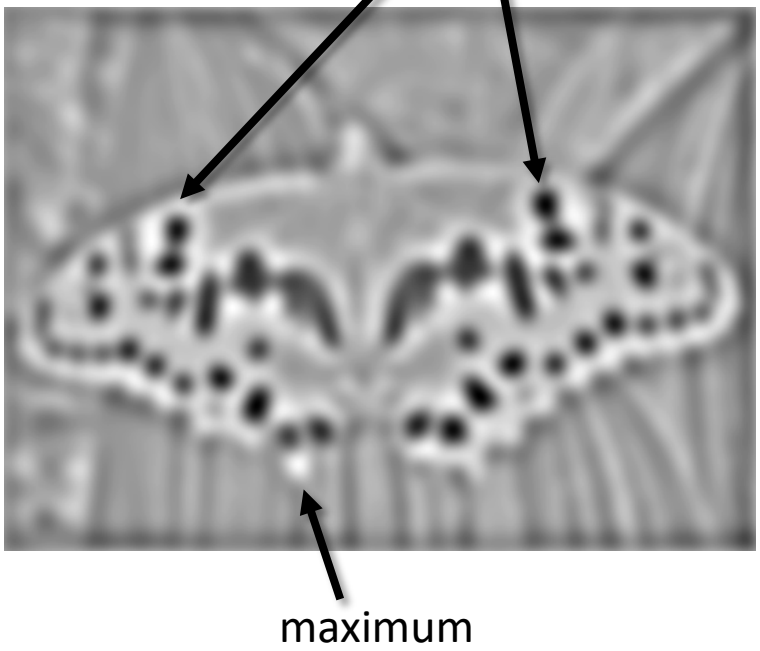


Highest response when the signal has the same **characteristic scale** as the filter

Laplacian of Gaussian

- “Blob” detector

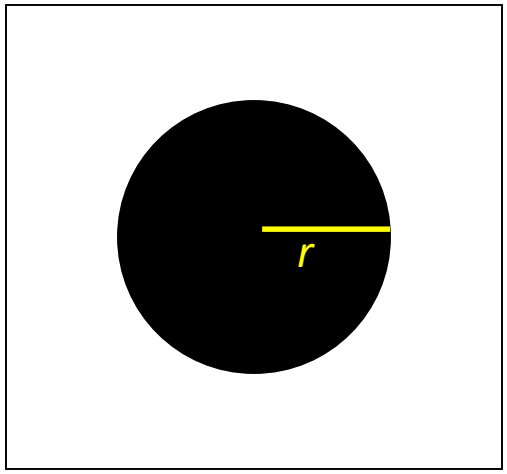




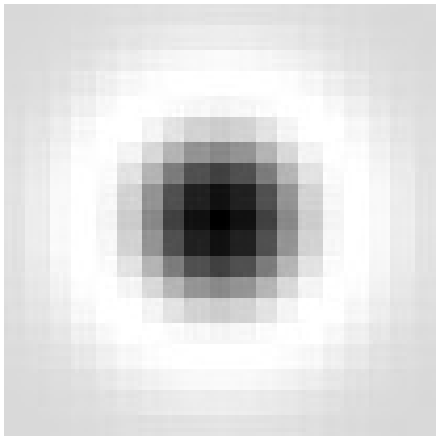
- Find maxima *and minima* of LoG operator in space and scale

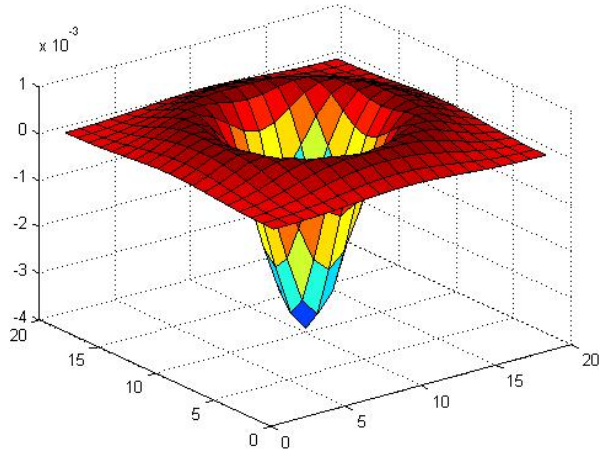
Scale selection

- At what scale does the Laplacian achieve a maximum response for a binary circle of radius r?



image

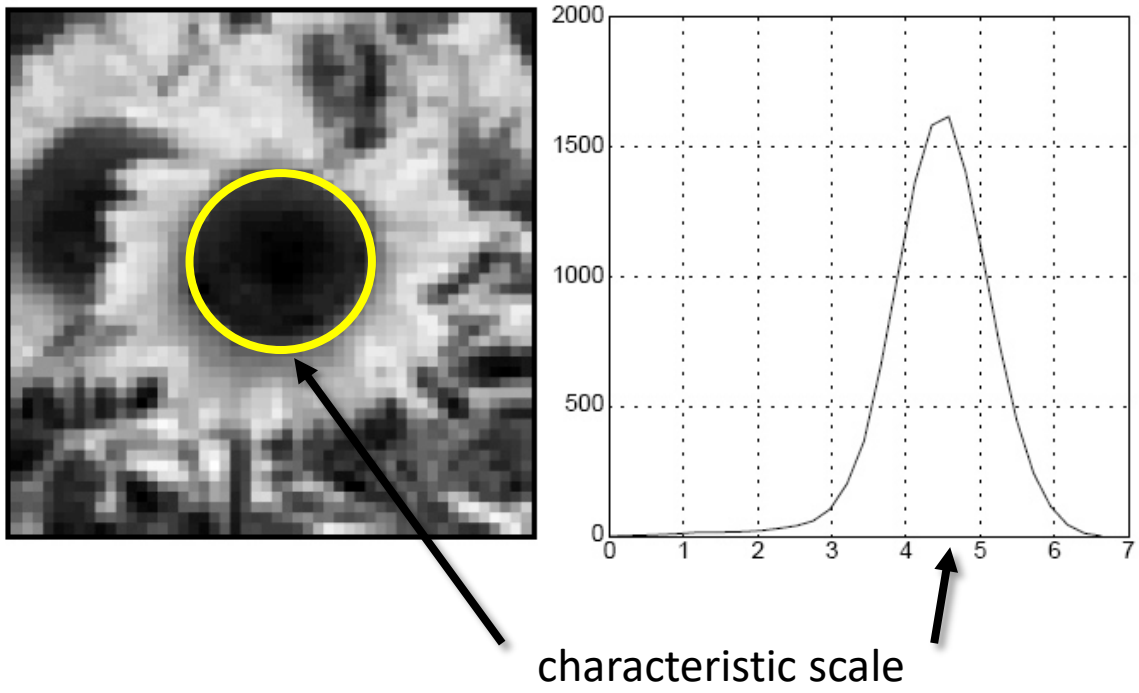




Laplacian

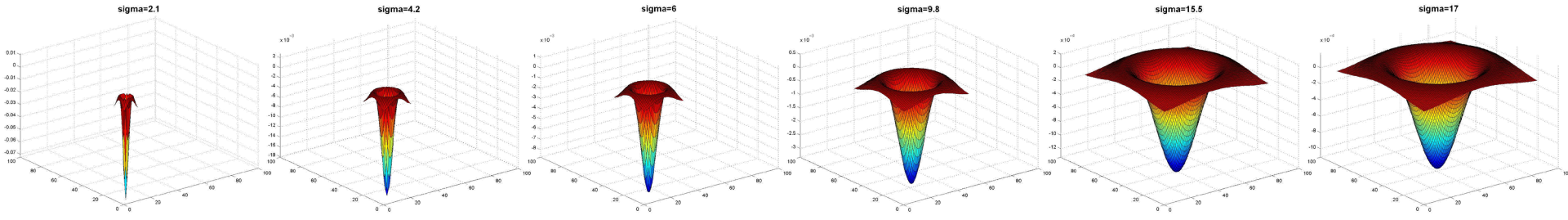
Characteristic scale

- We define the characteristic scale as the scale that produces peak of Laplacian response



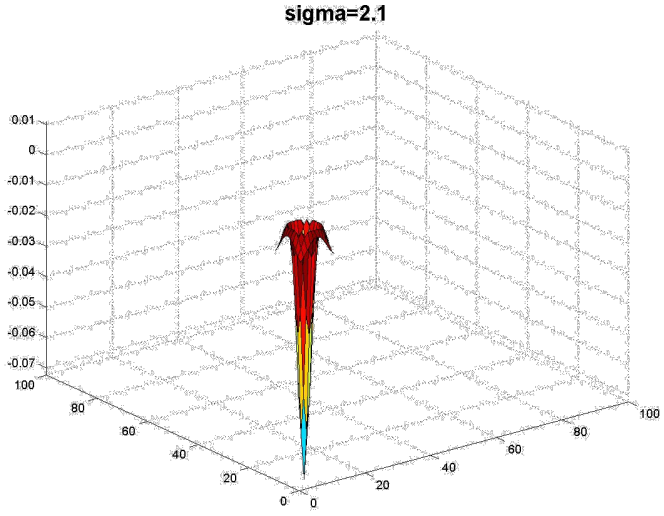
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

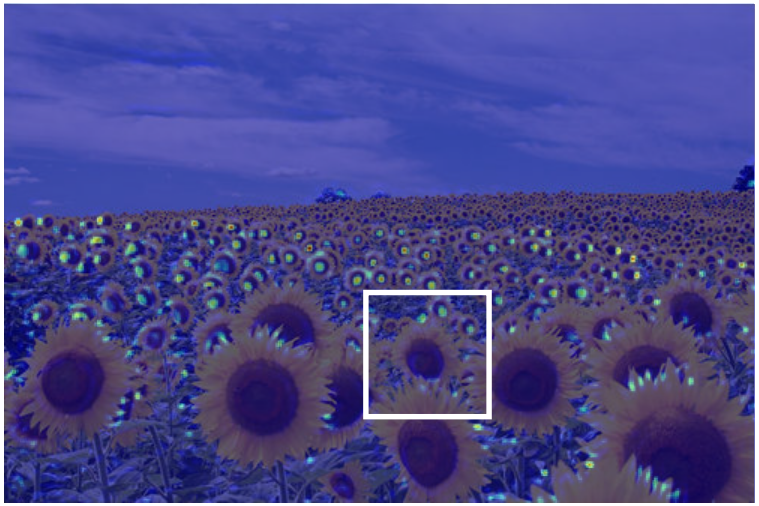
What happens if you apply different Laplacian filters?

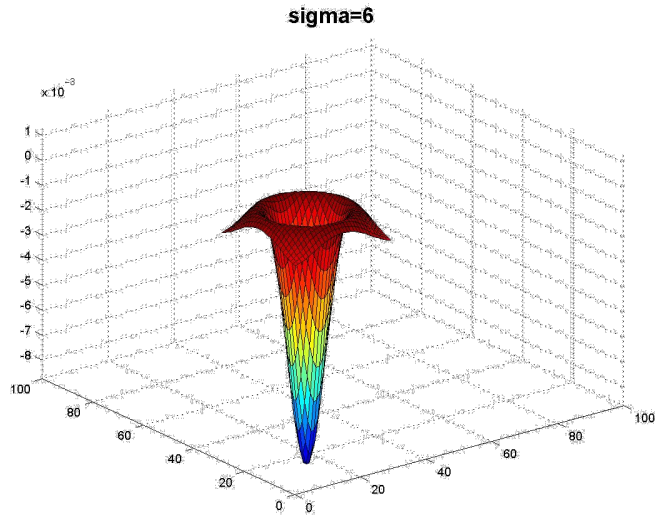


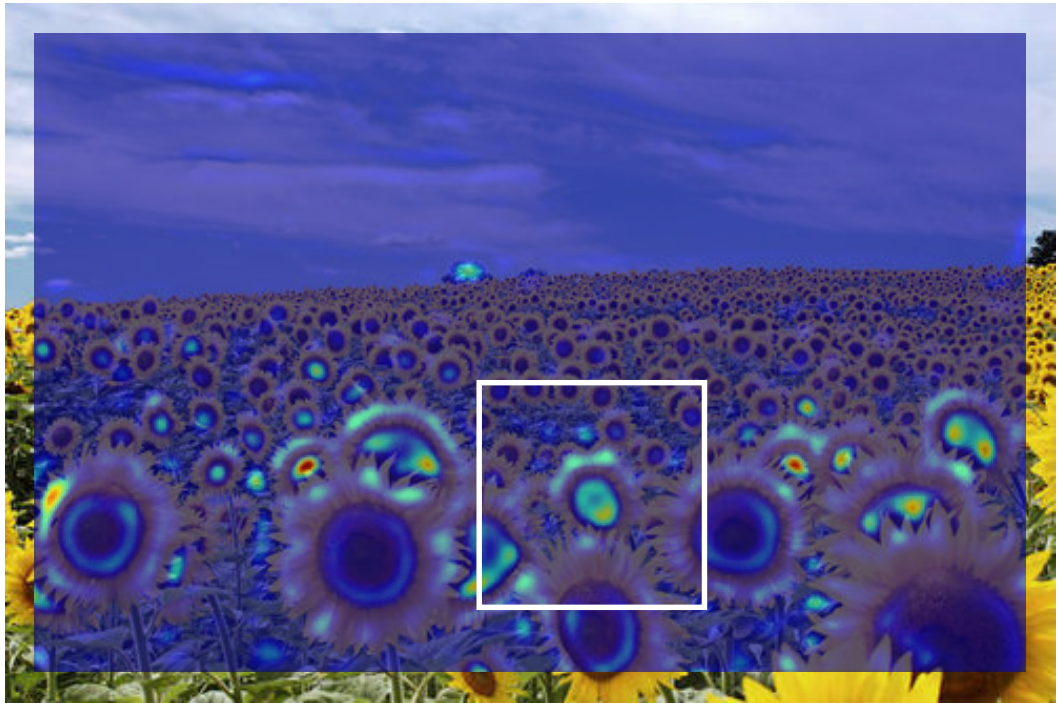
Full size

3/4 size

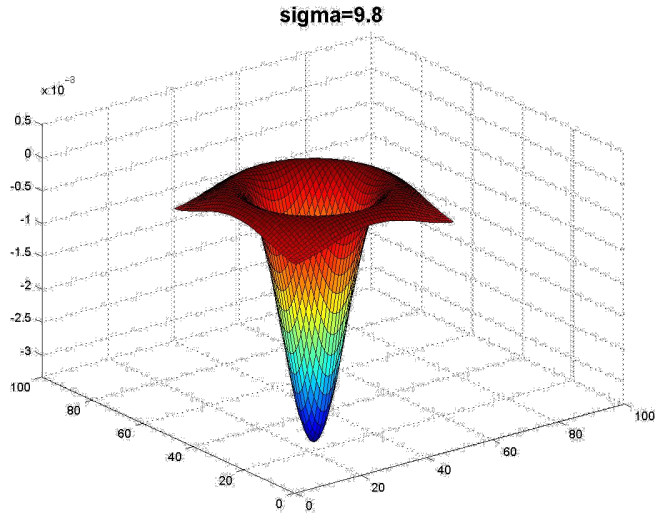

sigma=2.1
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
100
80
60
40
20
0
0
20
40
60
80
100



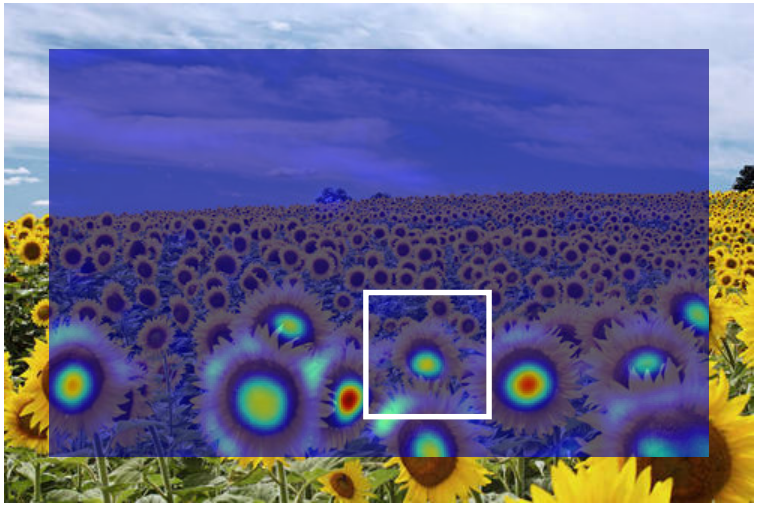

sigma=6
x10^-3
1
0
-1
-2
-3
-4
-5
-6
-7
-8
100
80
60
40
20
0
0
20
40
60
80
100

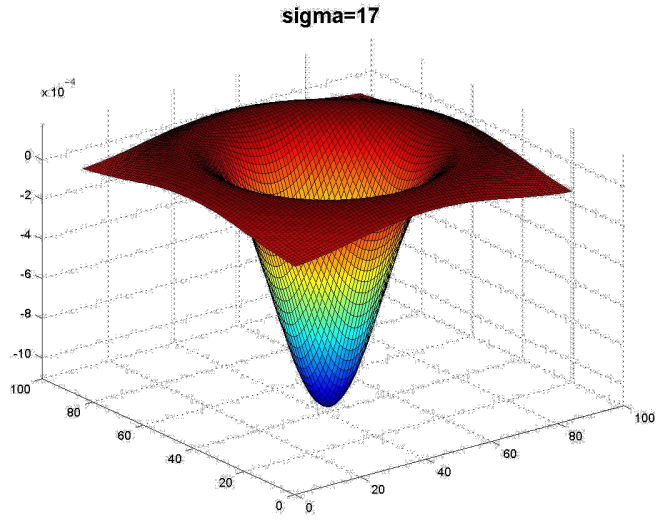


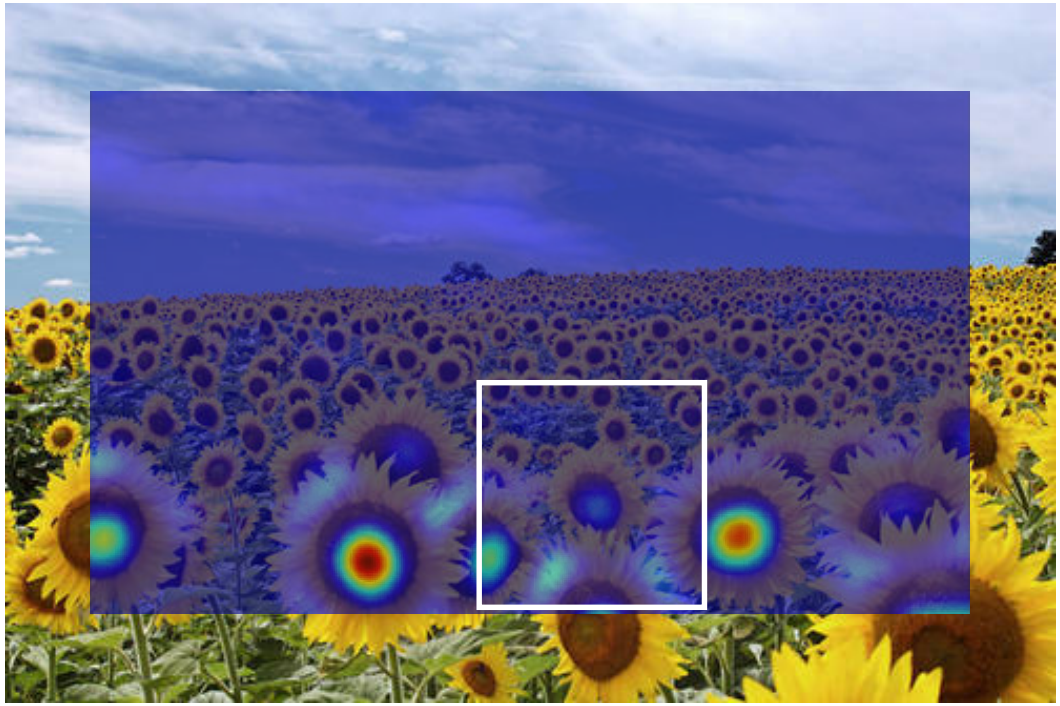

peak!

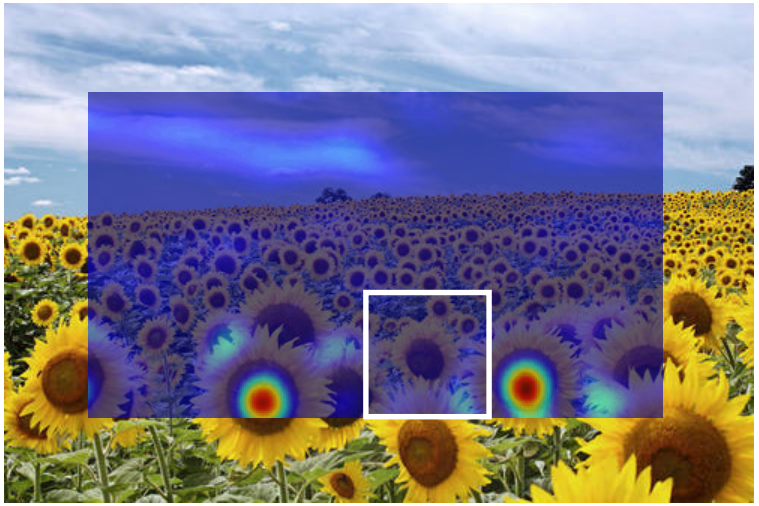

sigma=9.8
x10^-3
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
100
80
60
40
20
0
0
20
40
60
80
100


peak!




sigma=17
x10^-4
0
-2
-4
-6
-8
-10
100
80
60
40
20
0
0
20
40
60
80
100



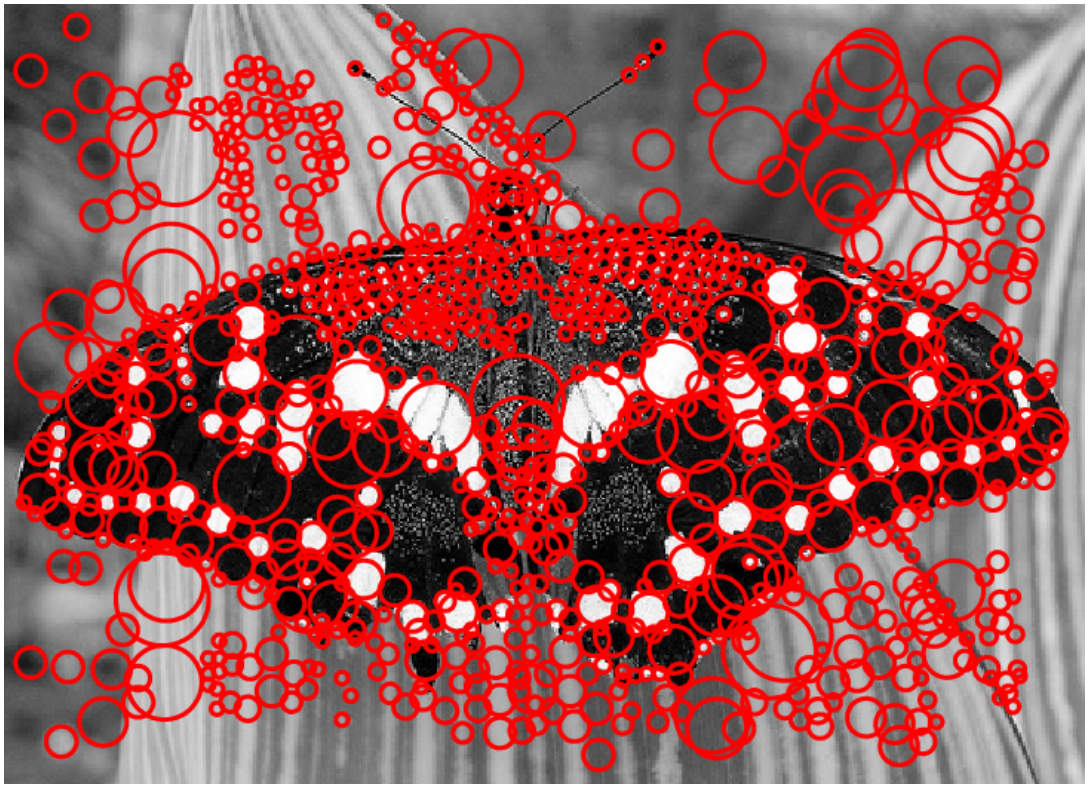


Scale-space blob detector: Example

Scale-space blob detector: Example

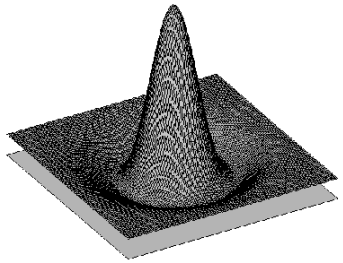
sigma = 11.9912

Scale-space blob detector: Example



Find local maxima in 3D position-scale space

Advanced



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

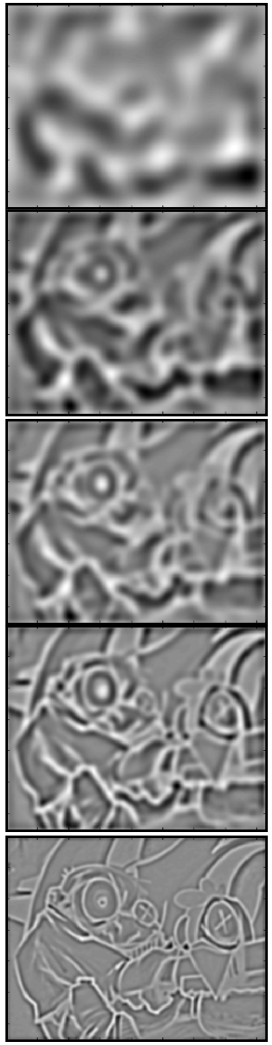
σ^5

σ^4

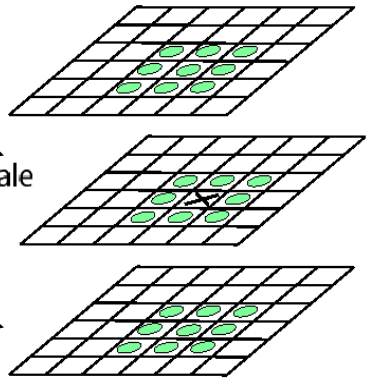
σ^3

σ^2

σ



Scale



⇒ **List of**
(x, y, s)

Scale Invariant Detection

- Functions for determining scale $f = \text{Kernel} * \text{Image}$

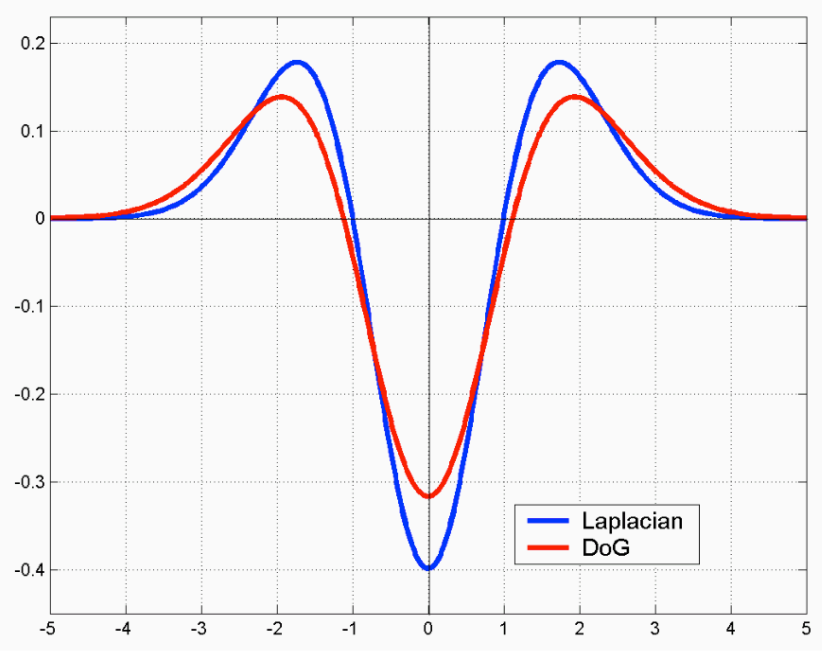
Kernels:

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



where Gaussian

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Note: The LoG and DoG operators are both rotation equivariant

DoG with Scale-space