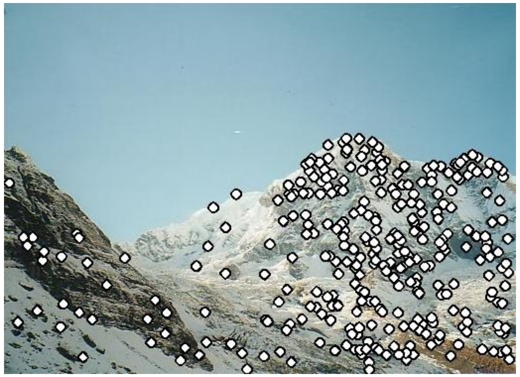
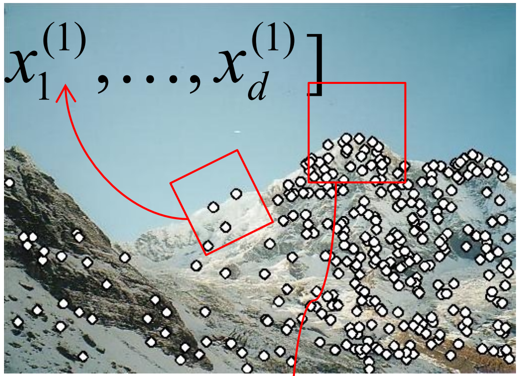
Local features: main components

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



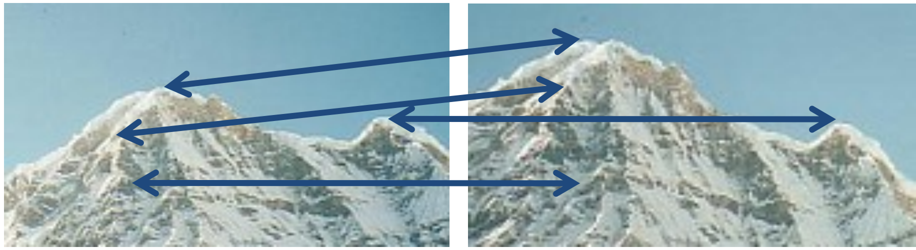
2) **Description:** Extract vector feature descriptor surrounding each interest point.

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



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

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

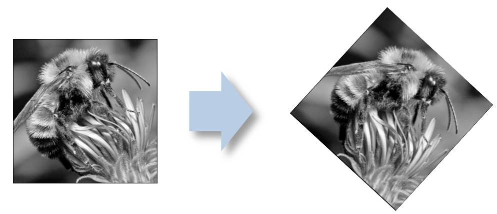


Kristen Grauman

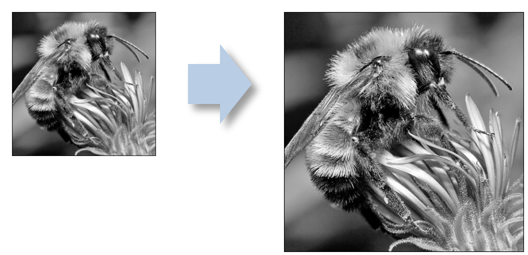
Image transformations

- Geometric

Rotation

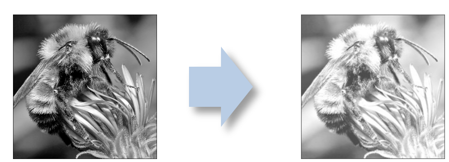


Scale

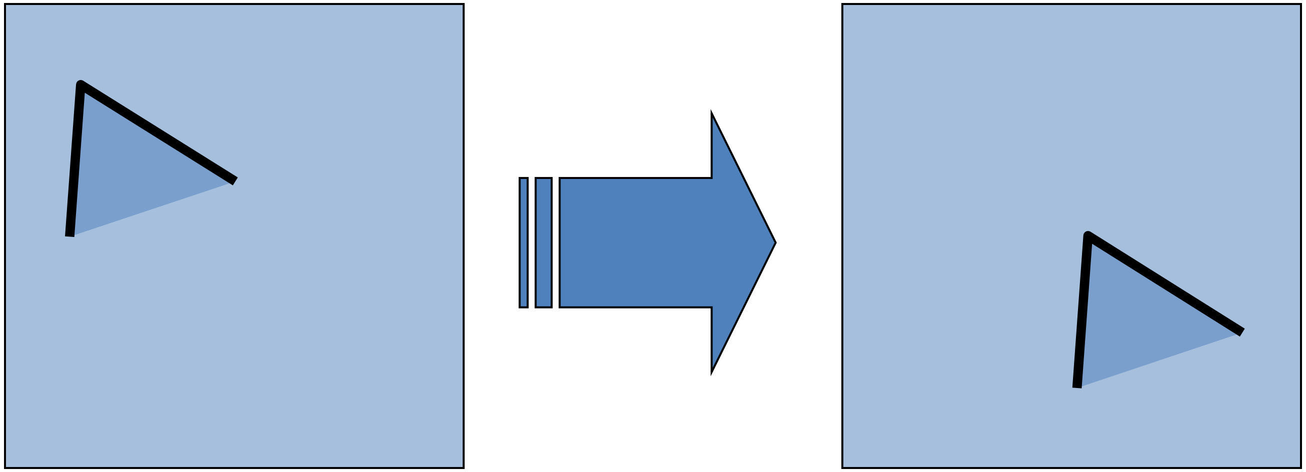


- Photometric

Intensity change

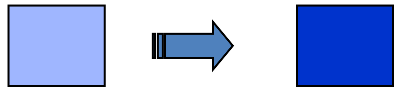


Harris detector: Invariance properties -- Image translation



- Derivatives and window function are shift-invariant

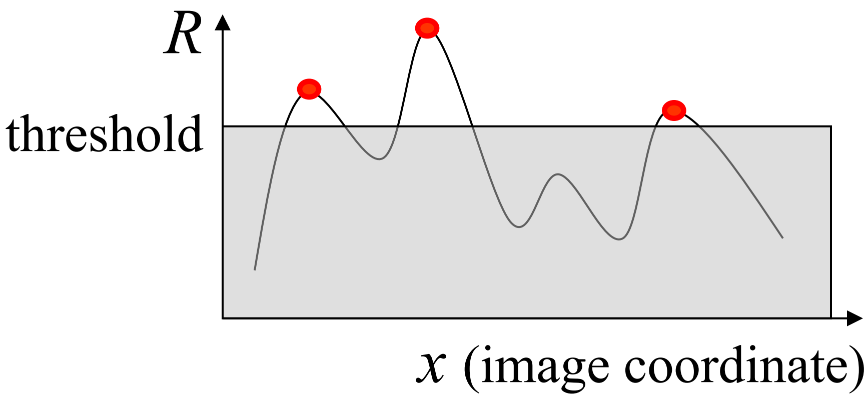
Harris detector: Invariance properties – Affine intensity change

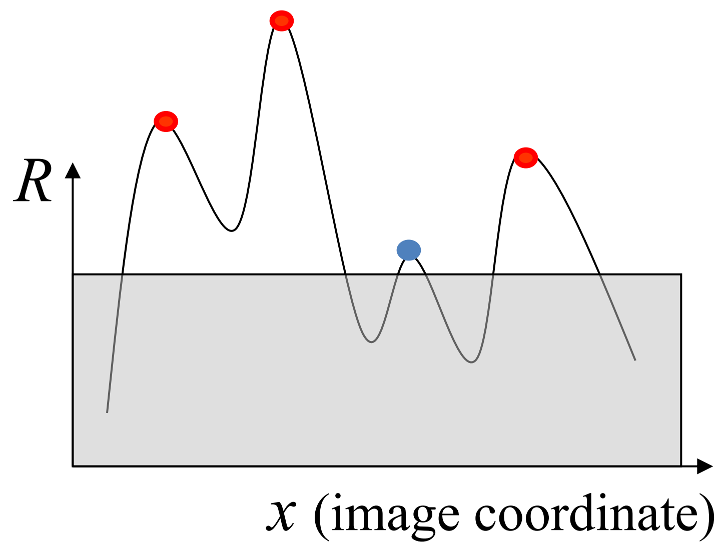


$I \rightarrow a\,I + b$

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

- Intensity scaling: $I \rightarrow a\,I$

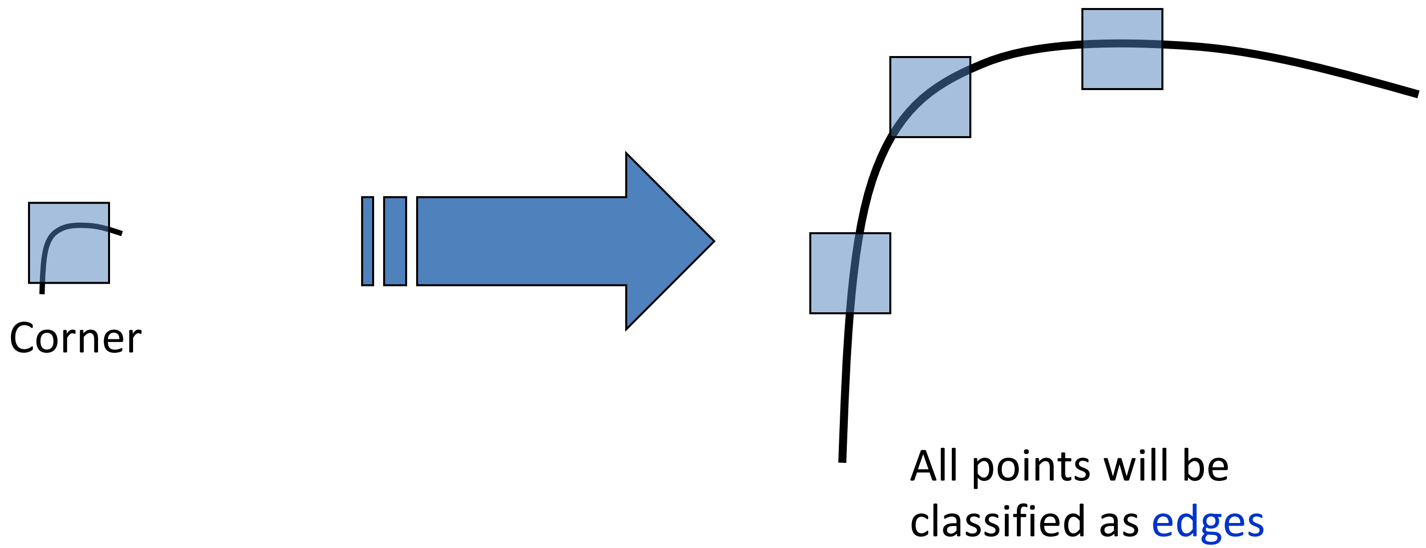




Partially invariant to affine intensity change

Harris Detector: Invariance Properties

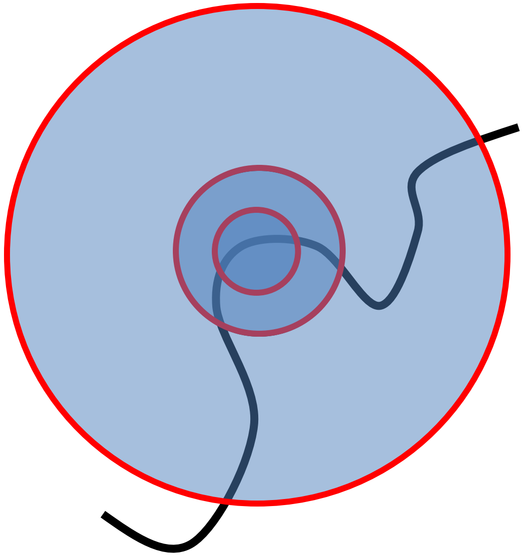
- Scaling



Not invariant to scaling

Scale invariant detection

Suppose you're looking for corners

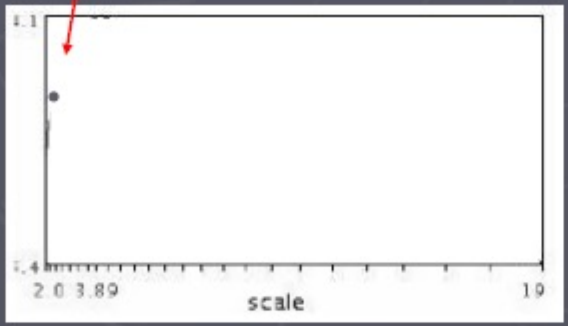


Key idea: find scale that gives local maximum of *f*

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

Automatic scale selection

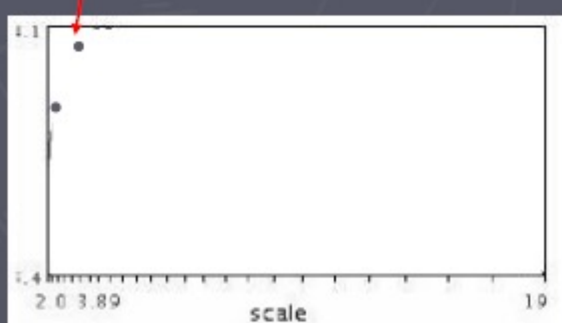
Lindeberg et al., 1996



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

Slide from Tinne Tuytelaars

Automatic scale selection



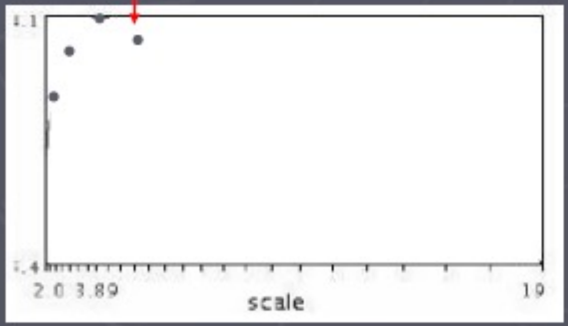
$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

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

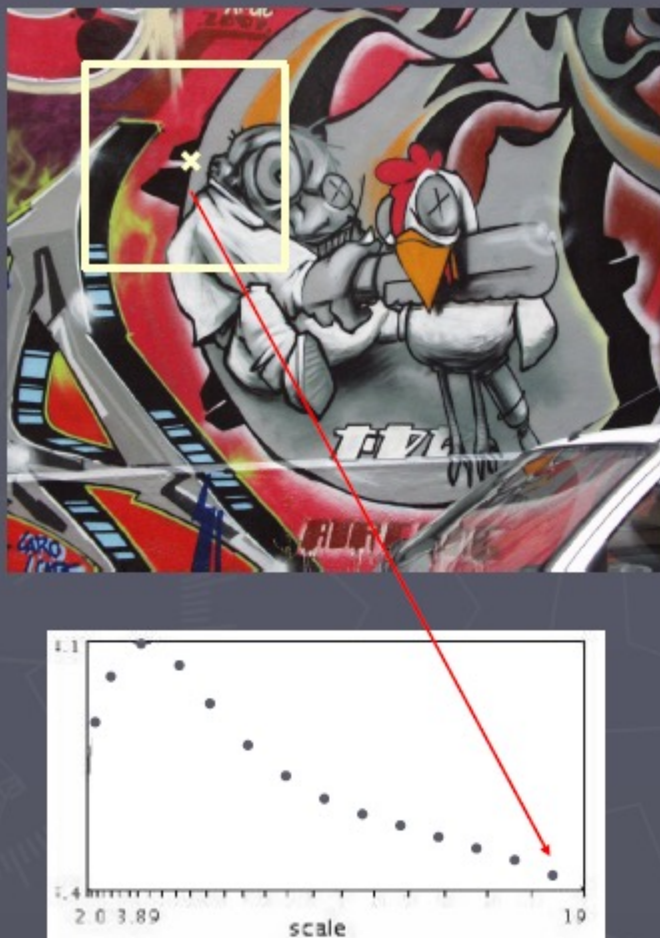
Automatic scale selection



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

Automatic scale selection

Function responses for increasing scale
Scale trace (signature)

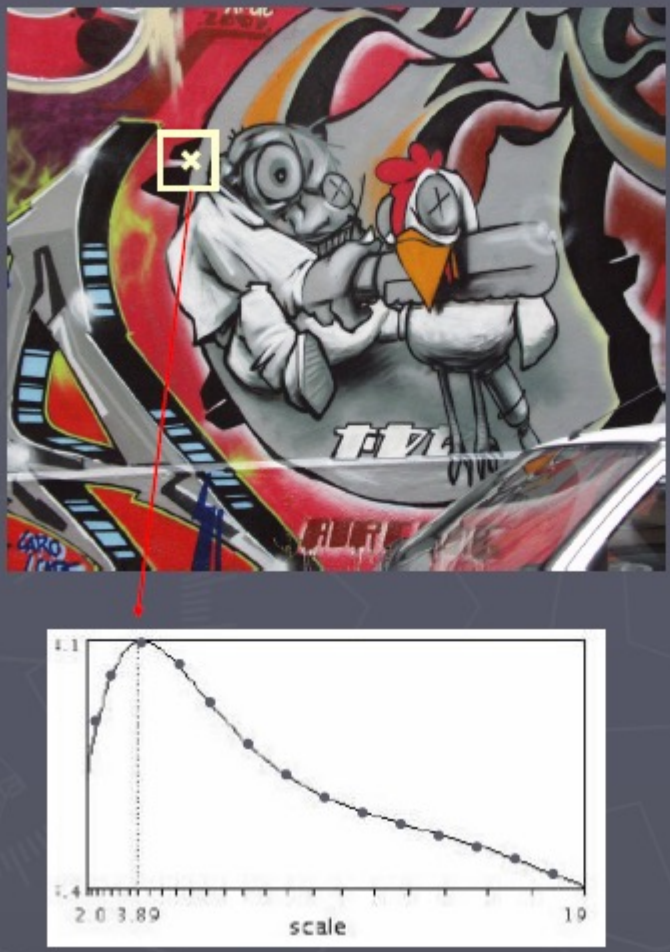


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

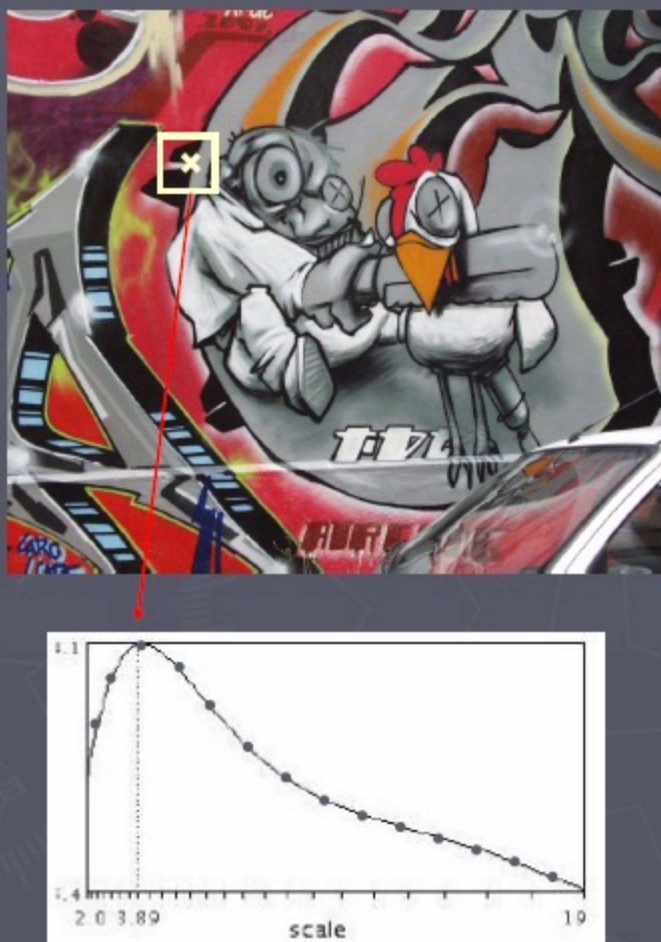
Function responses for increasing scale

Scale trace (signature)

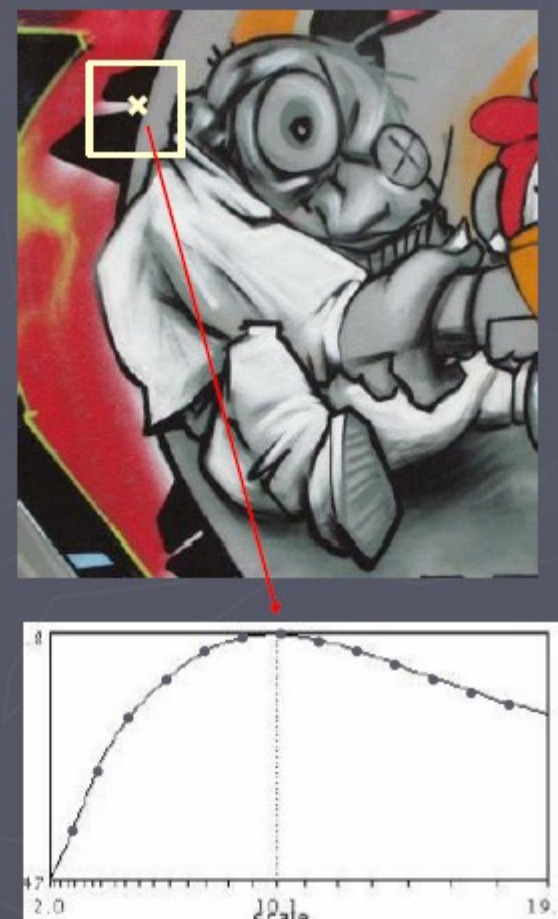


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection



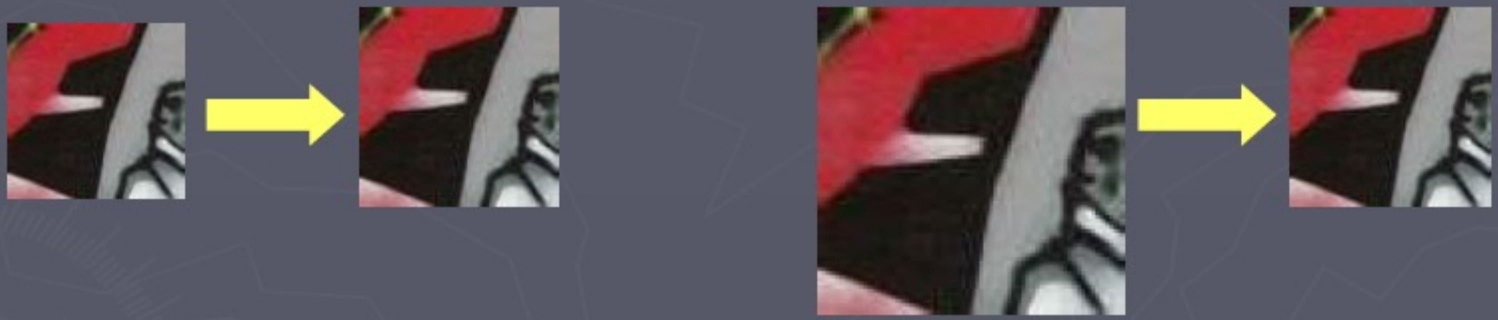
$f(I_{i_1 \dots i_m}(x,\sigma))$



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

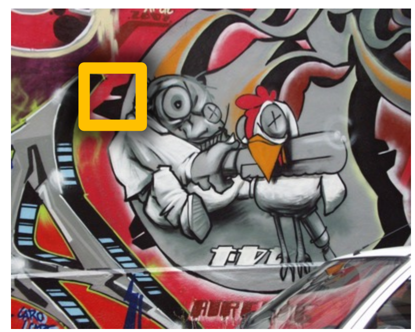
Automatic scale selection

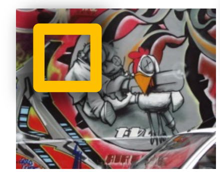
Normalize: rescale to fixed size

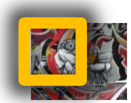


Implementation

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

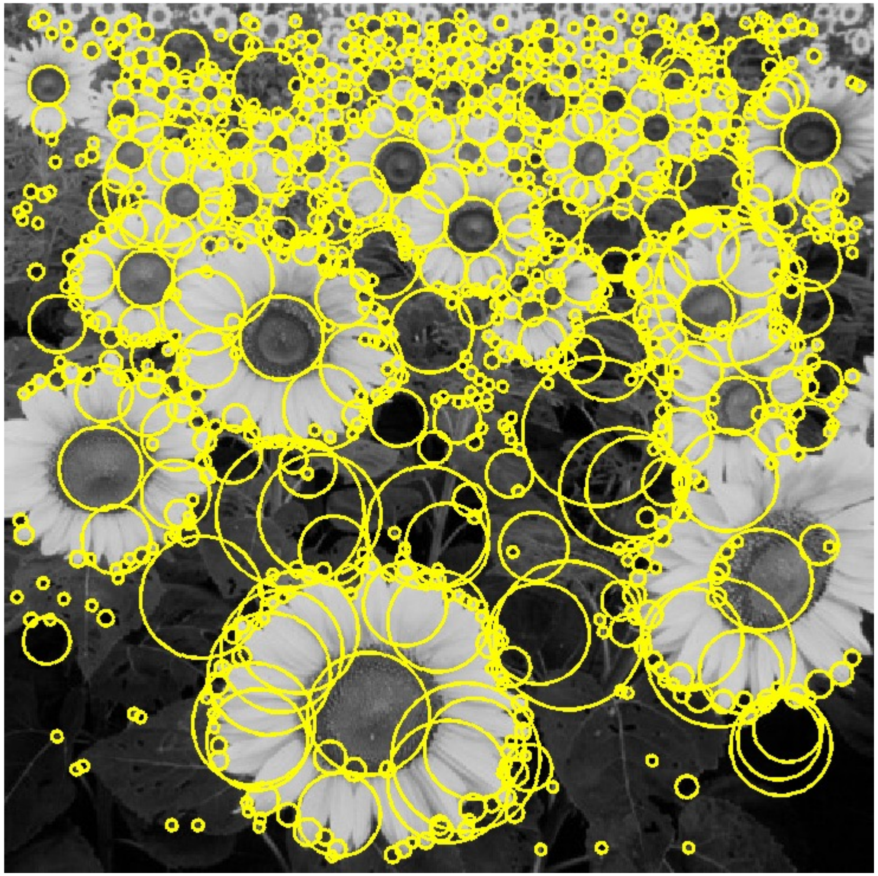






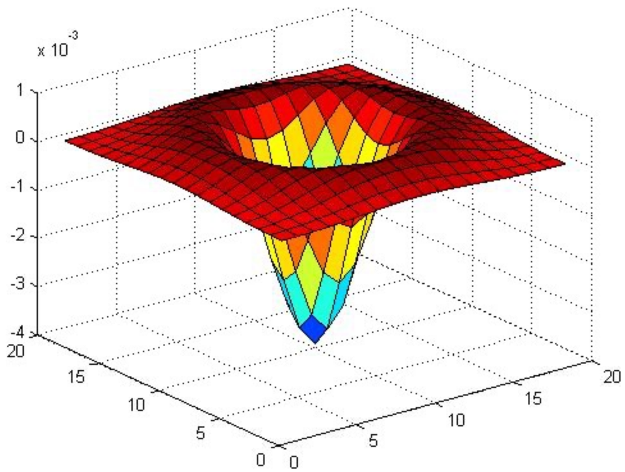
(sometimes need to create in-between levels, e.g. a ¾-size image)

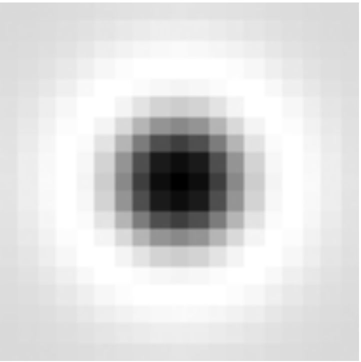
Feature extraction: Corners and blobs



Another common definition of *f*

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





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

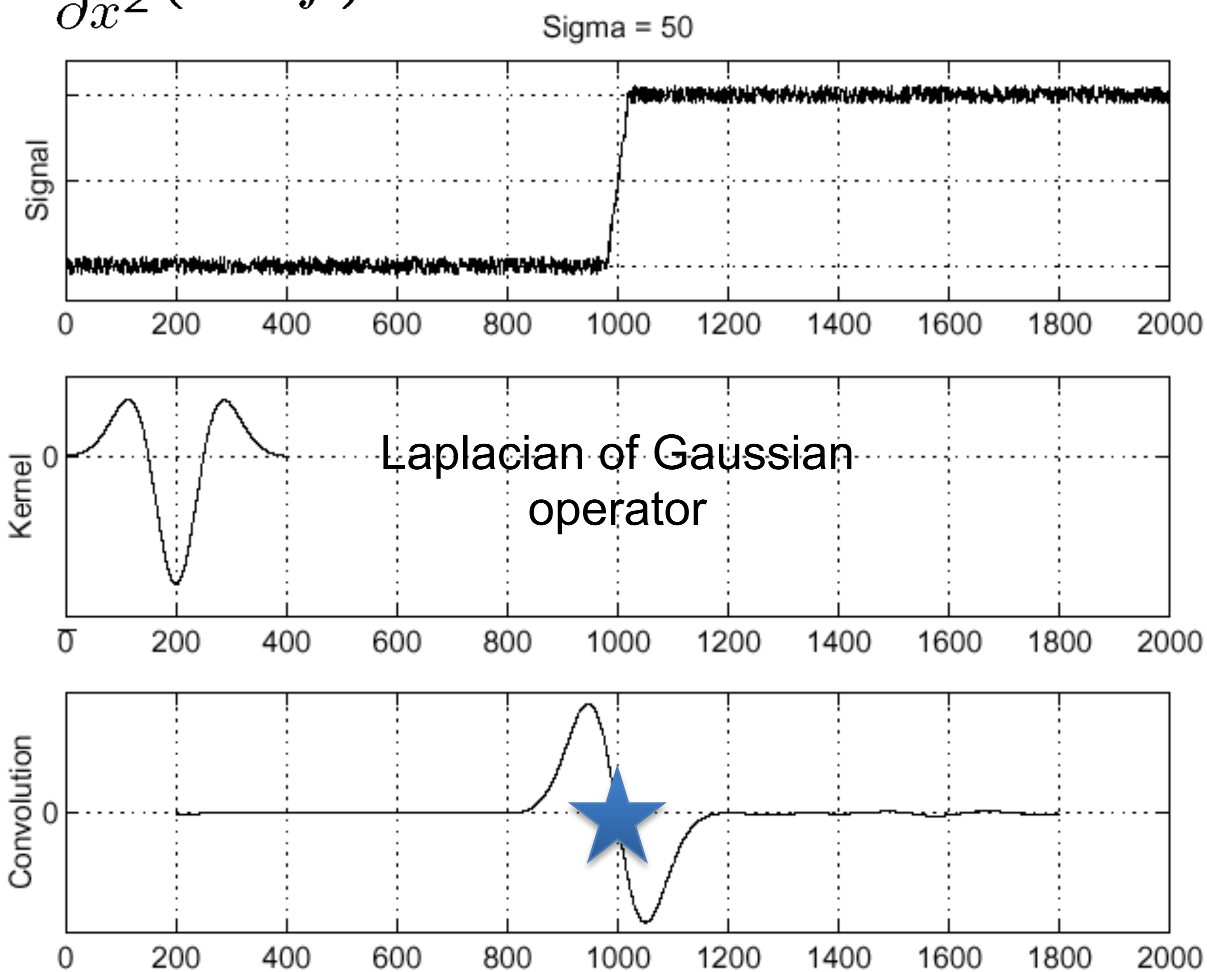
Laplacian of Gaussian (LoG)

- Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

$\frac{\partial^2}{\partial x^2}h$

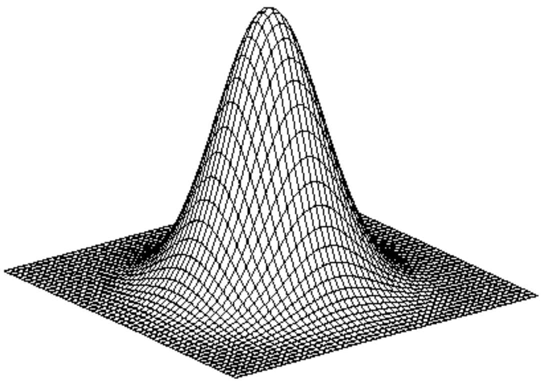
$(\frac{\partial^2}{\partial x^2}h) \star f$

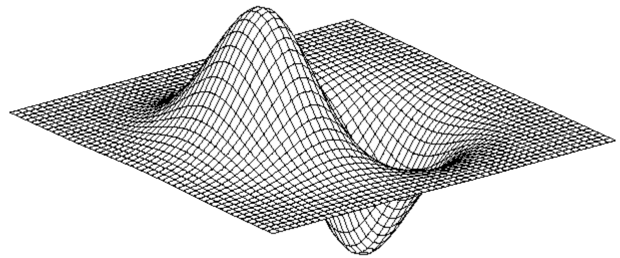


Where is the edge?

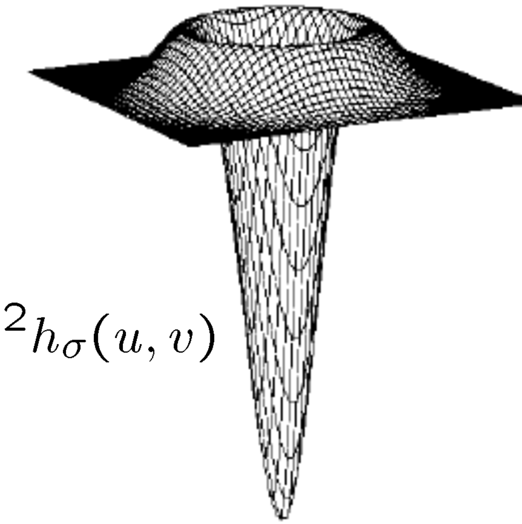
Zero-crossings of bottom graph

2D edge detection filters





Laplacian of Gaussian



Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

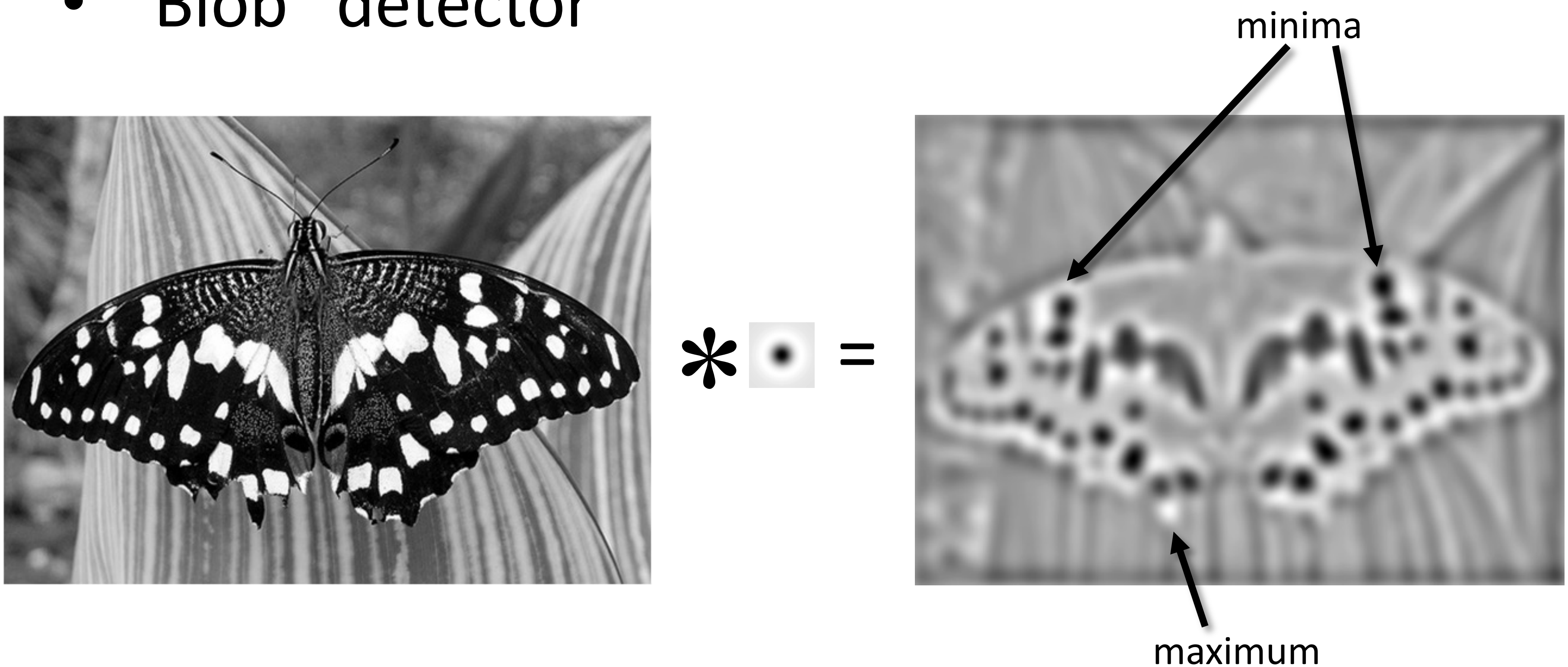
$$\nabla^2 h_\sigma(u,v)$$

∇^2 is the Laplacian operator:

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

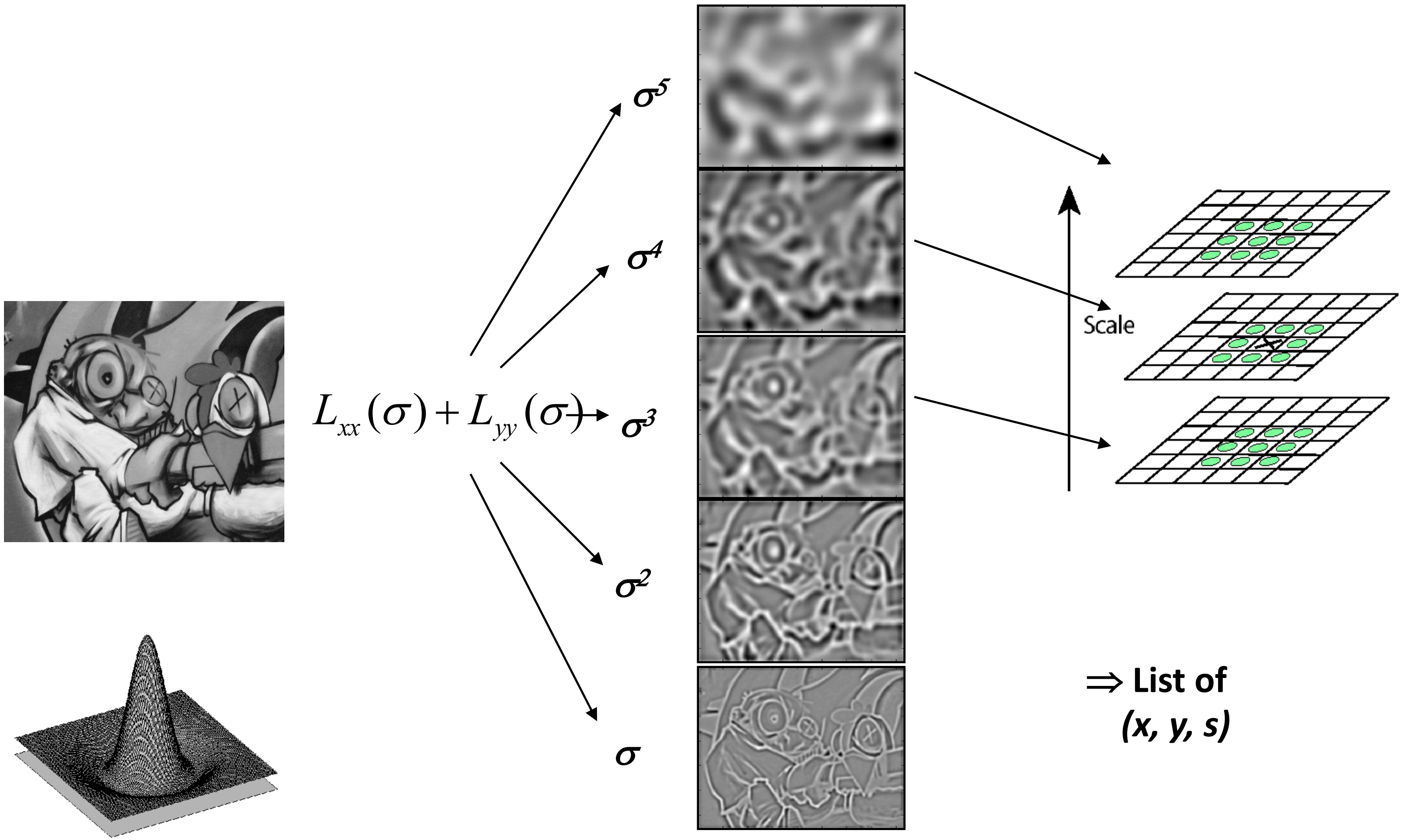
Laplacian of Gaussian

- "Blob" detector



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

Find local maxima in position-scale space



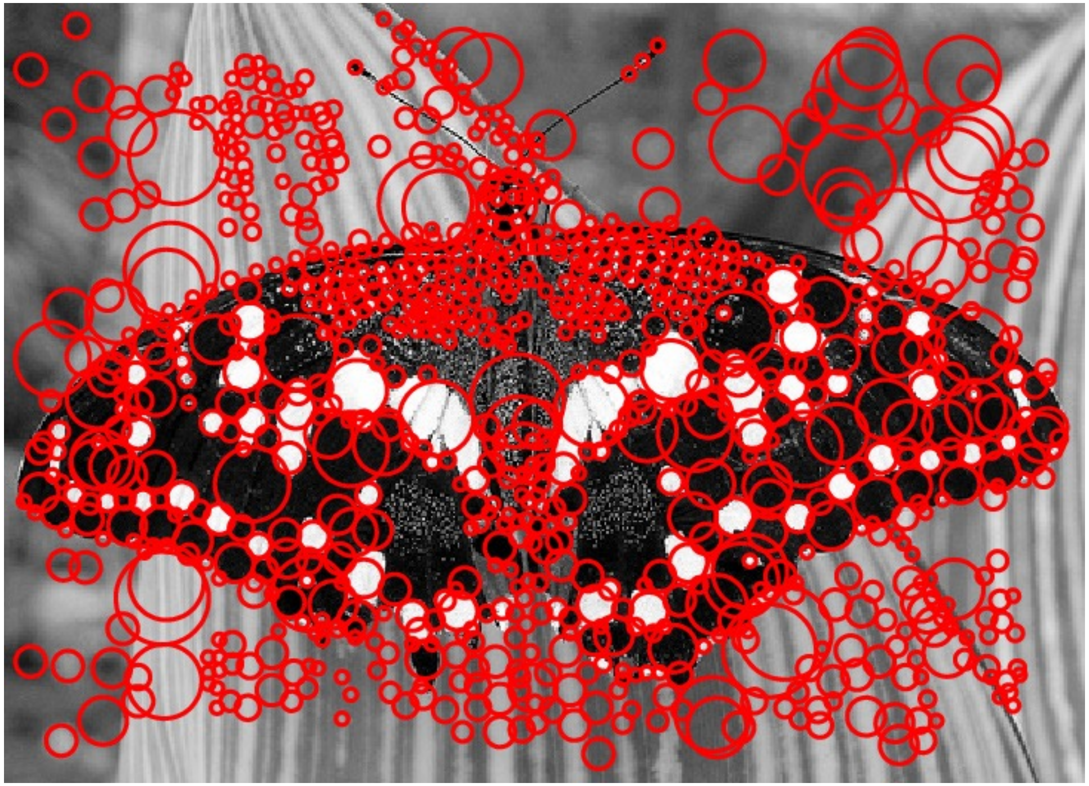
K. Grauman, B. Leibe

Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

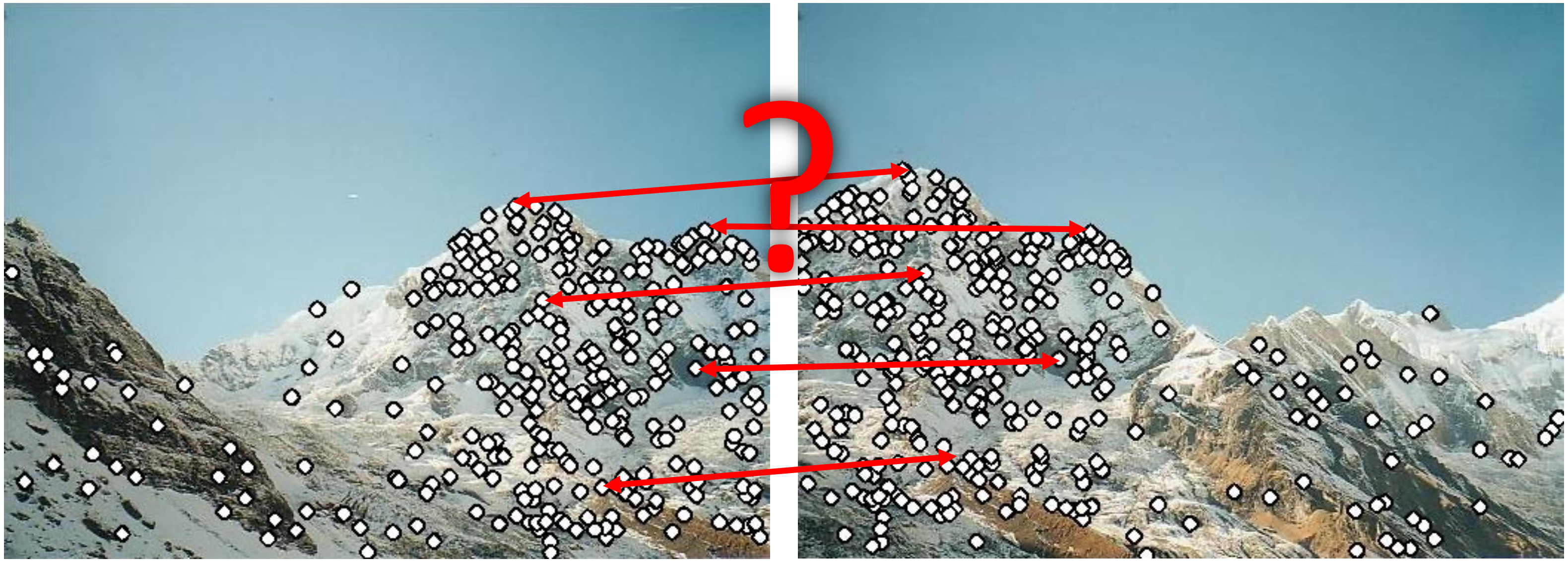
Scale-space blob detector: Example



Feature descriptors

We know how to detect good points

Next question: **How to match them?**

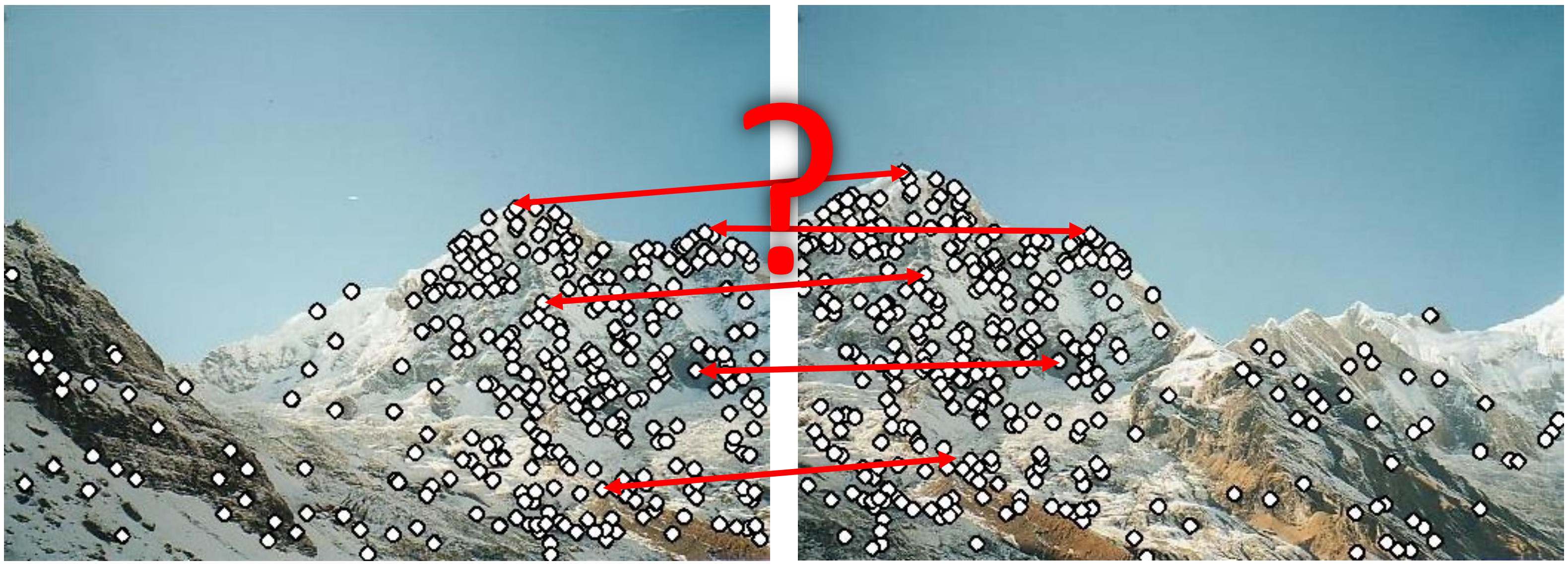


Answer: Come up with a *descriptor* for each point,
find similar descriptors between the two images

Feature descriptors

We know how to detect good points

Next question: **How to match them?**



Lots of possibilities (this was a popular research area)

- Simple option: match square windows around the point
- Popular approach: SIFT
  - David Lowe, UBC http://www.cs.ubc.ca/~lowe/keypoints/