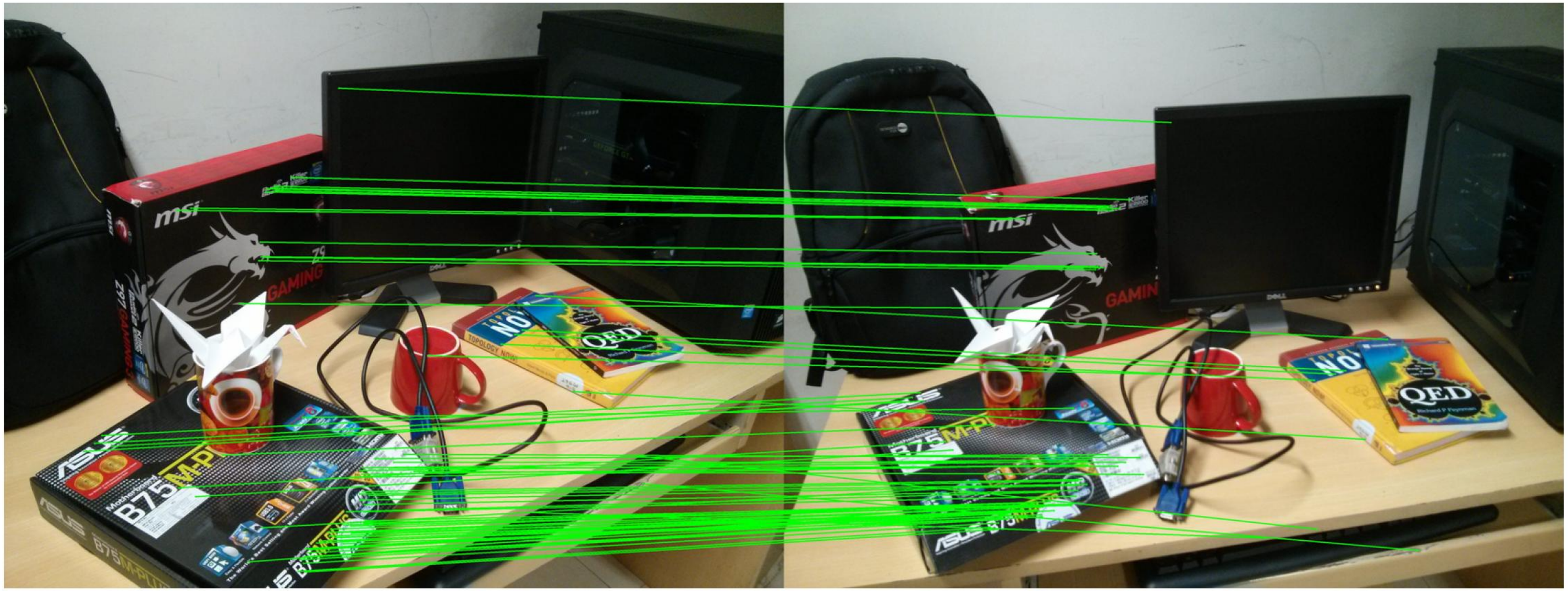
Scale Invariance and SIFT

CS 4391 Introduction Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Feature Detection and Matching

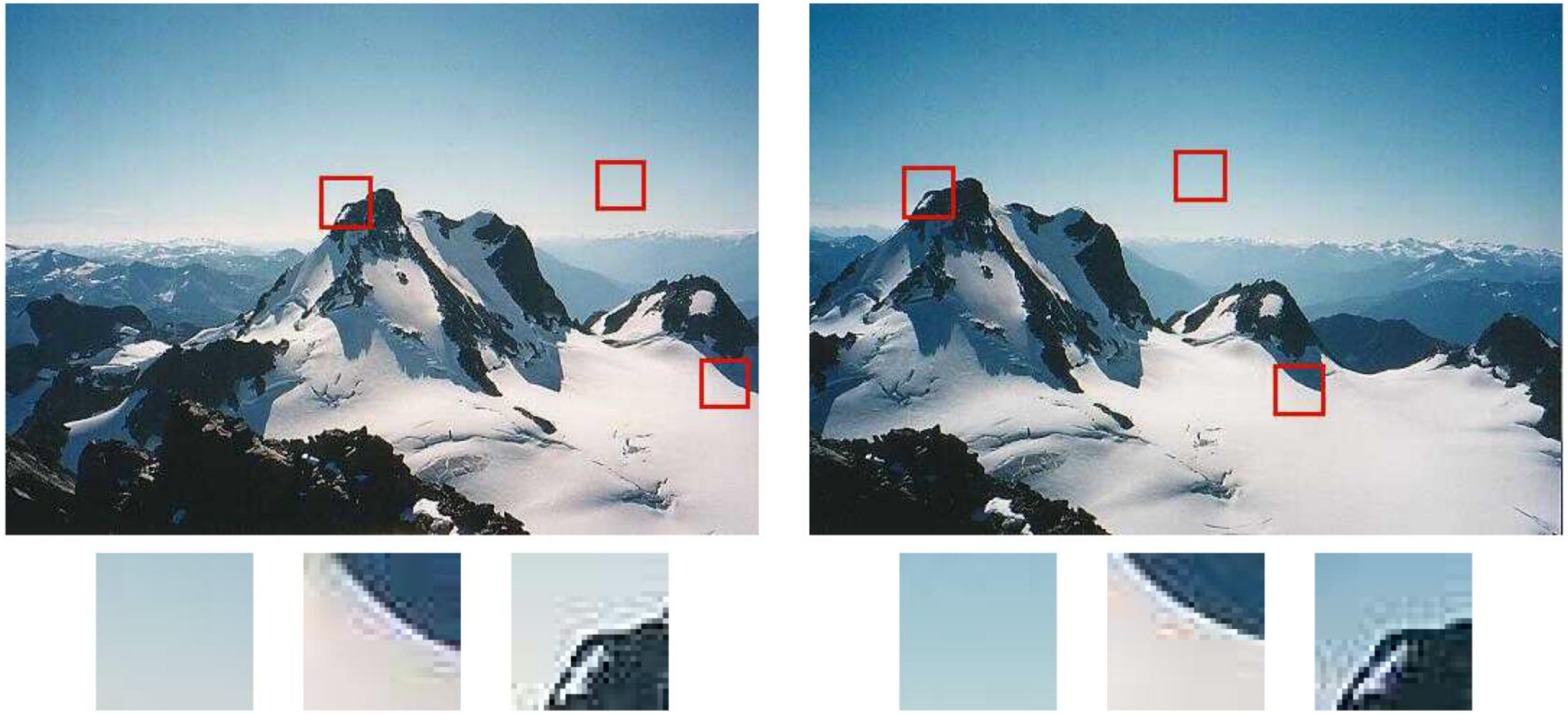


Geometry-aware Feature Matching for Structure from Motion Applications. Shah et al, WACV'15

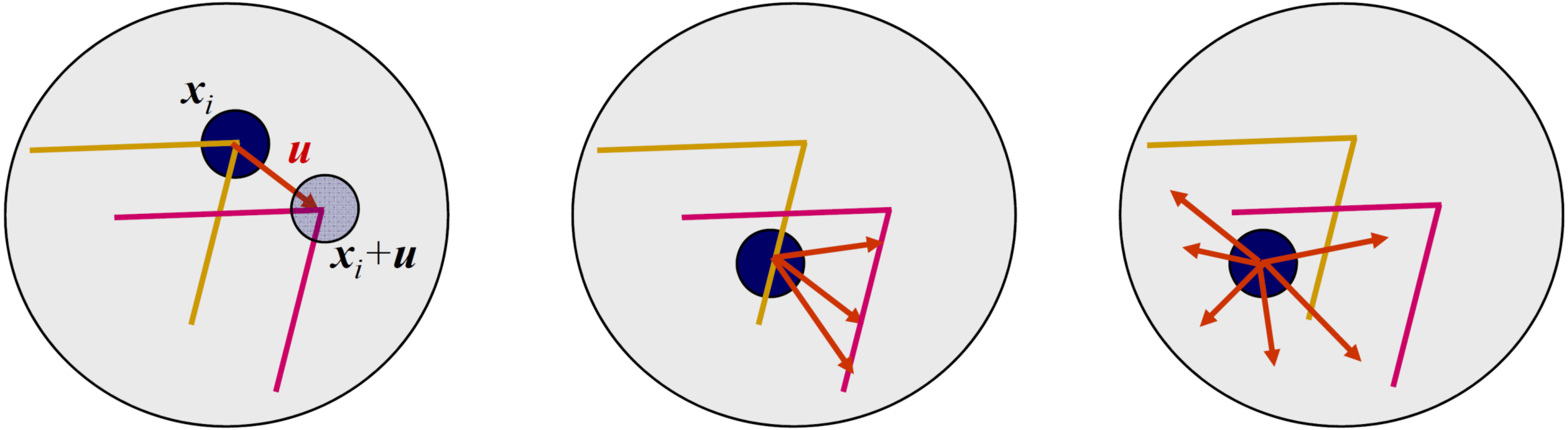
Applications: stereo matching, image stitching, 3D reconstruction, camera pose estimation, object recognition

Feature Detectors

- How to find image locations that can be reliably matched with images?



Feature Detectors



(a) Corner

(b) Edge

(c) Textureless region

Harris Corner Detector

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

1. Compute x and y derivatives of image

Sobel filter

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$I_x = G_\sigma^x * I \qquad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x^2} = I_x \cdot I_x \qquad I_{y^2} = I_y \cdot I_y \qquad I_{xy} = I_x \cdot I_y$$

3. Compute the sums of products of derivatives at each pixel

Gaussian window

$$S_{x^2} = G_{\sigma'} * I_{x^2} \qquad S_{y^2} = G_{\sigma'} * I_{y^2} \qquad S_{xy} = G_{\sigma'} * I_{xy}$$

Harris Corner Detector

3. Determine the matrix at every pixel

$$M(x, y) = \begin{bmatrix} S_{x^2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{y^2}(x, y) \end{bmatrix}$$

4. Compute the response of the detector at each pixel

$$R = \det M - k(\mathrm{trace} M)^2$$

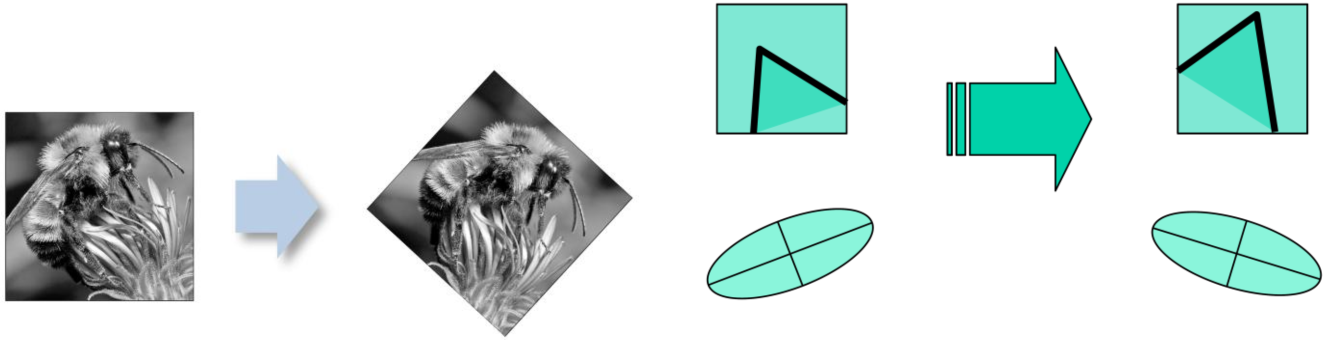
5. Threshold on R and perform non-maximum suppression

Invariance

- Can the same feature point be detected after some transformation?
  - Translation invariance

    Are Harris corners translation invariance?
  - 2D rotation invariance

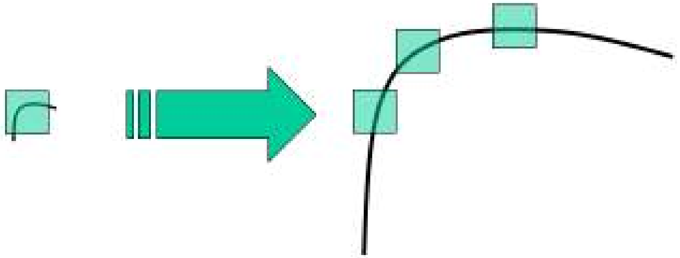
    Are Harris corners rotation invariance?

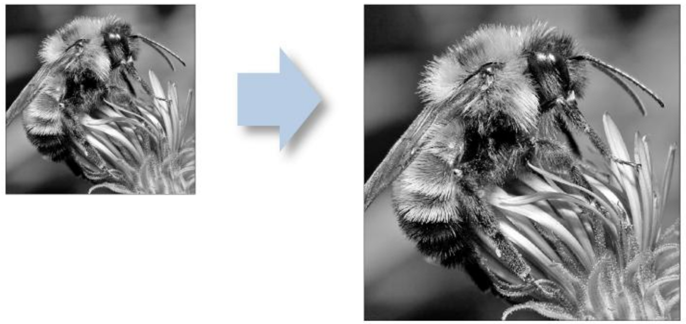


  - Scale invariance

    Are Harris corners scale invariance?

No





Scale Invariance

- Solution 1: detection features in all scales, matching features in corresponding scale (for small scale change)

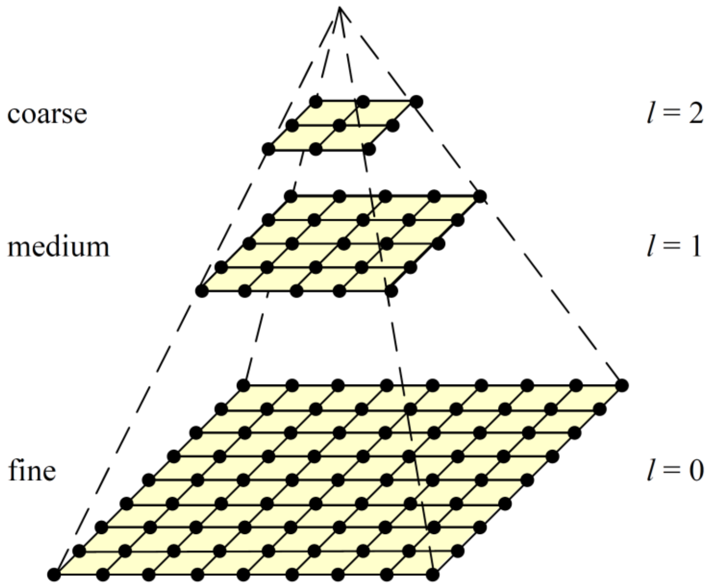
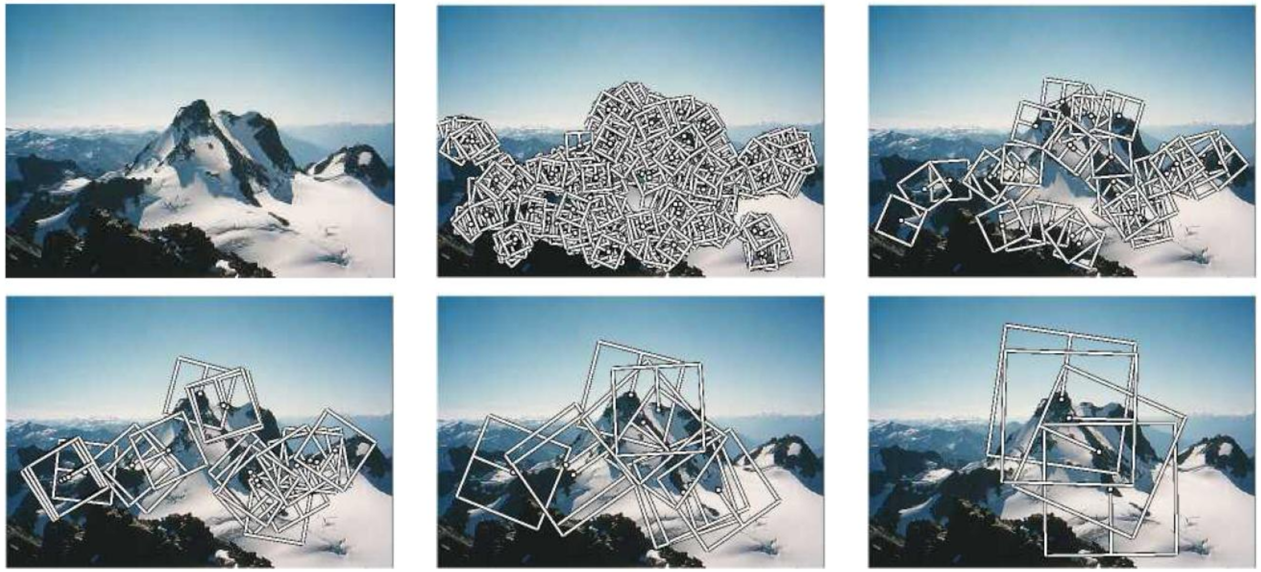


Image pyramid



Multi-scale oriented patches (MOPS) extracted at five pyramid levels (Brown, Szeliski, and Winder 2005)

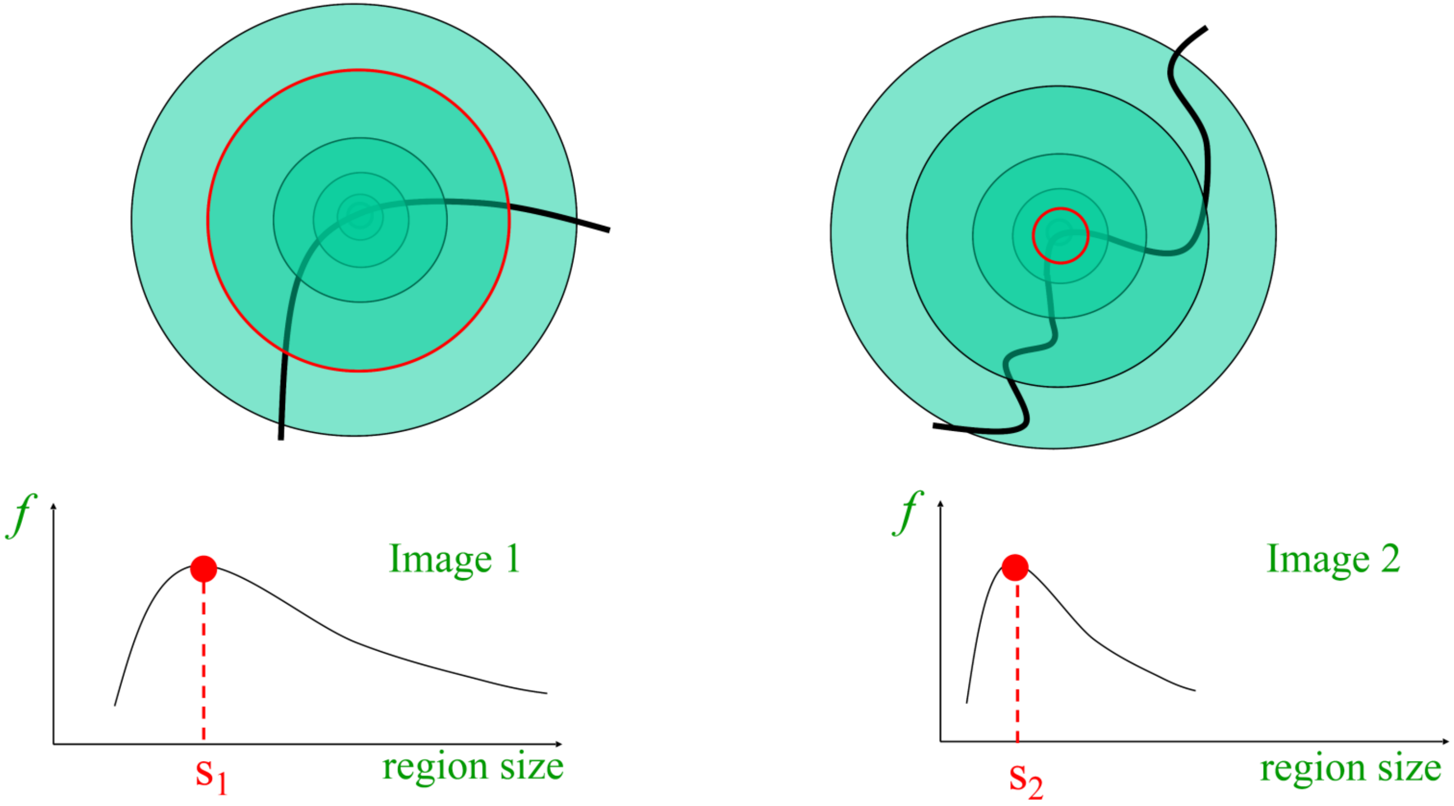
Scale Invariance

- Solution 2: detect features that are stable in both location and scale

Consider Harris corner detector

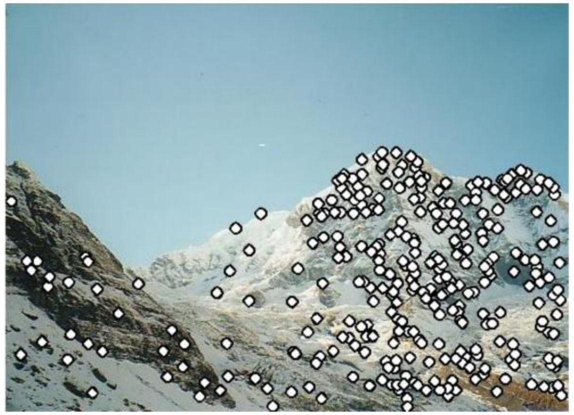
Intuition: Find local maxima in both position and scale

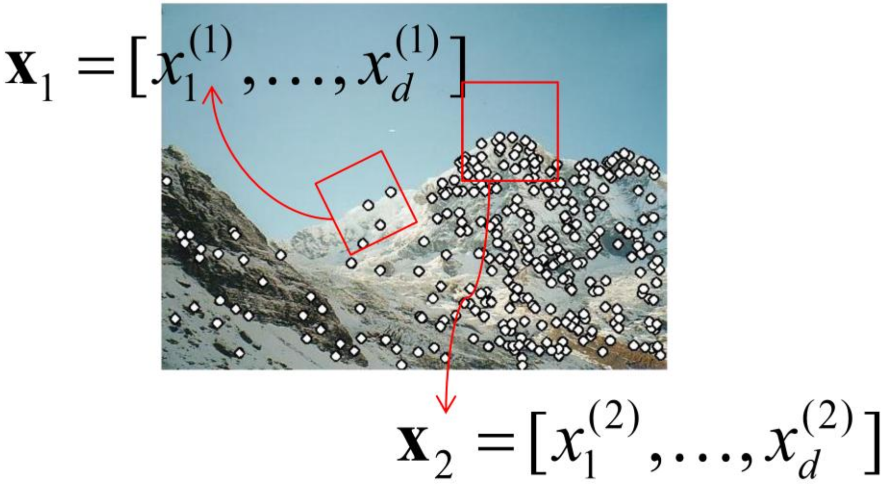
What filter can we use for scale selection?



Scale Invariance Feature Transform (SIFT)

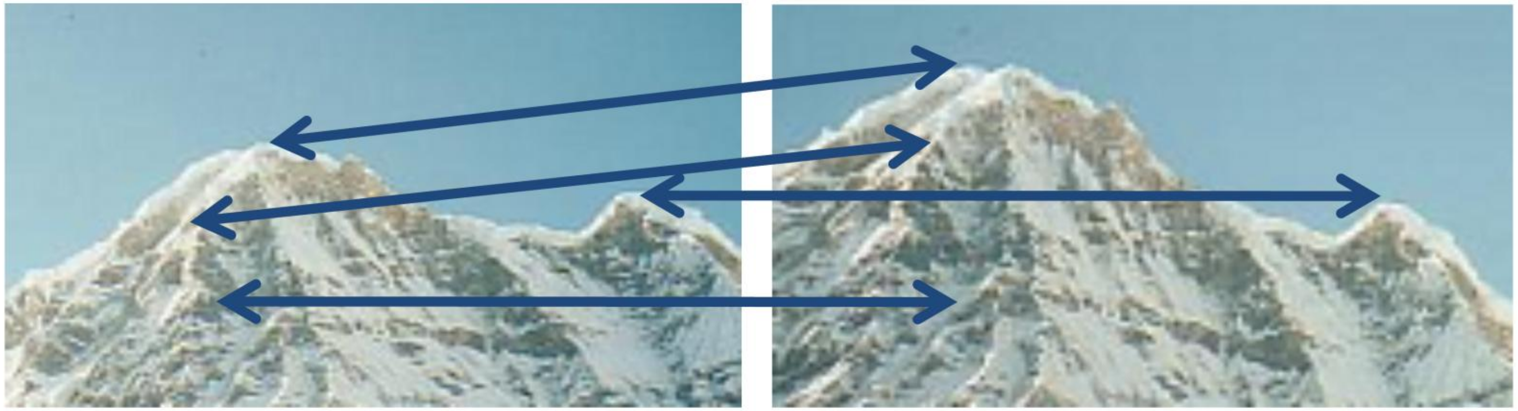
- Keypoint detection





- Compute descriptors

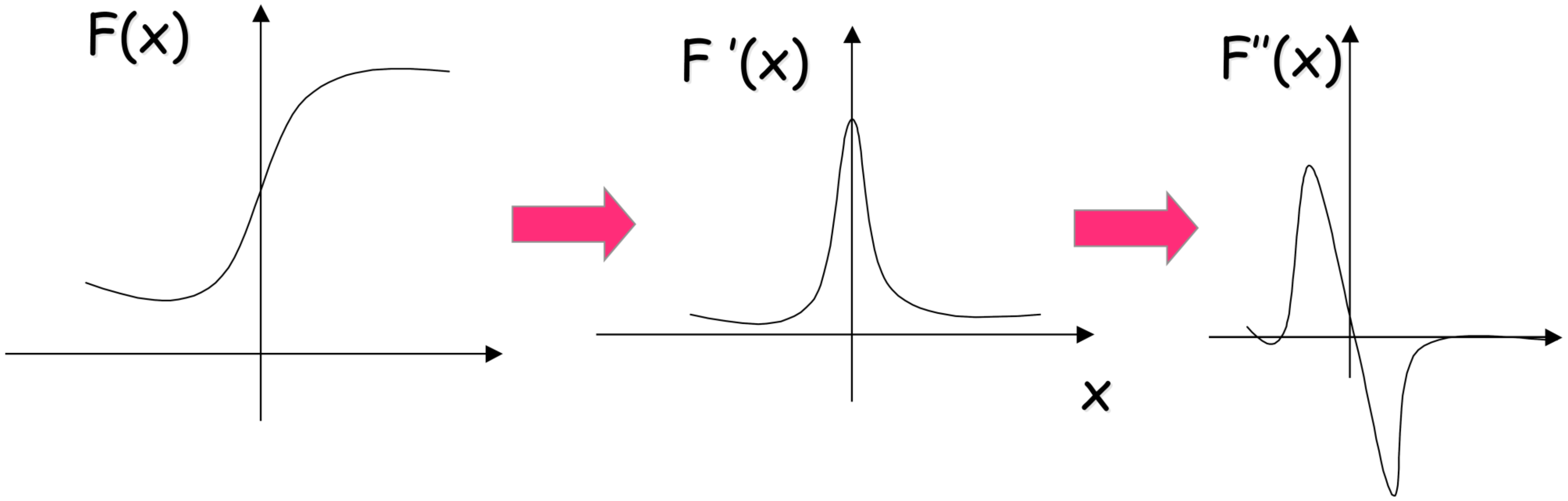
- Matching descriptors



David Lowe, Distinctive Image Features from Scale-Invariant Keypoints. IJCV, 2004

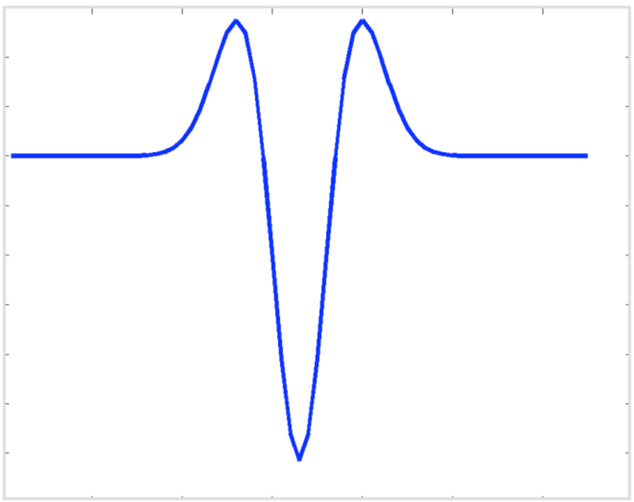
Recall: Second Derivative Filters

- Peaks or valleys of the first-derivative of the input signal, correspond to "zero-crossings" of the second-derivative of the input signal



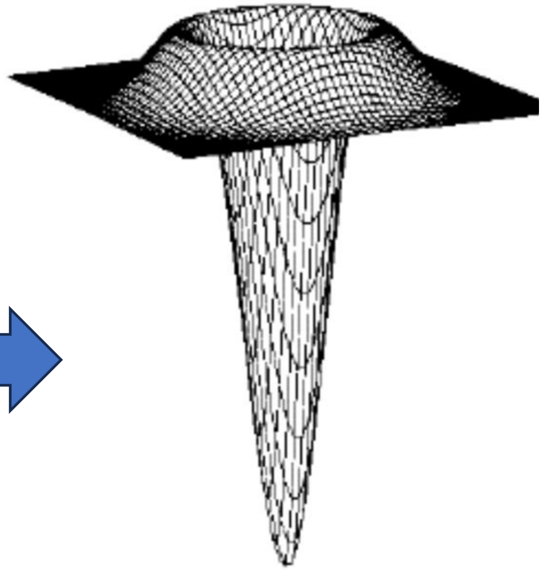
Recall: Second Derivate of Gaussian

$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right)e^{-\frac{x^2}{2\sigma^2}}$$



2D

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

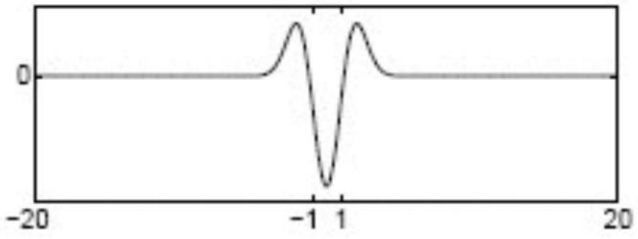


Laplacian of Gaussian

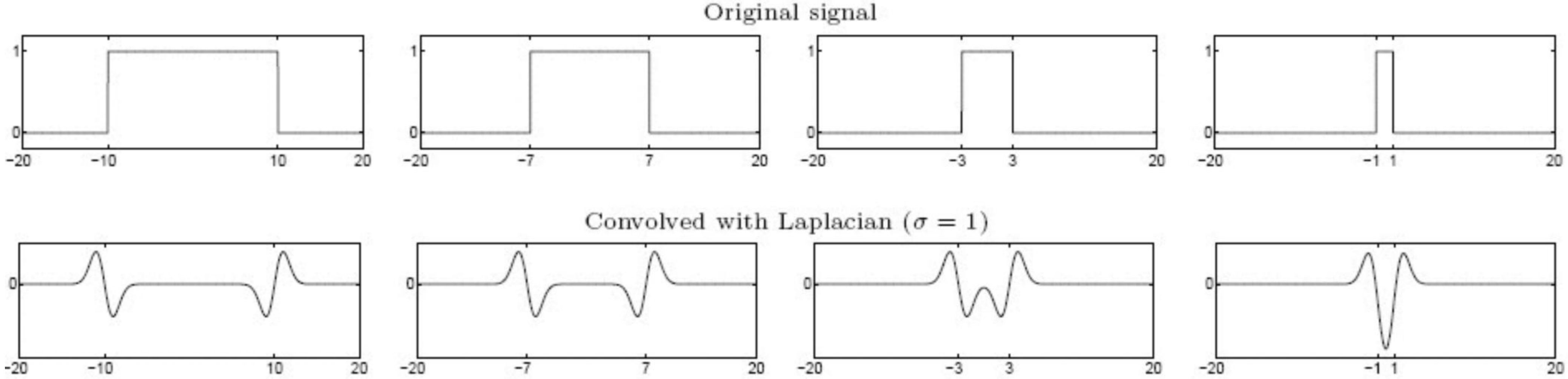
Mexican Hat Function

Laplacian of Gaussian for Scale Selection

Laplacian filter

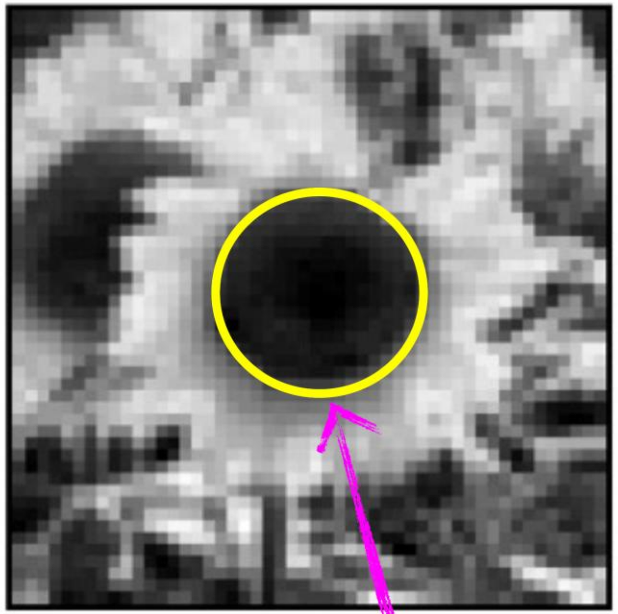


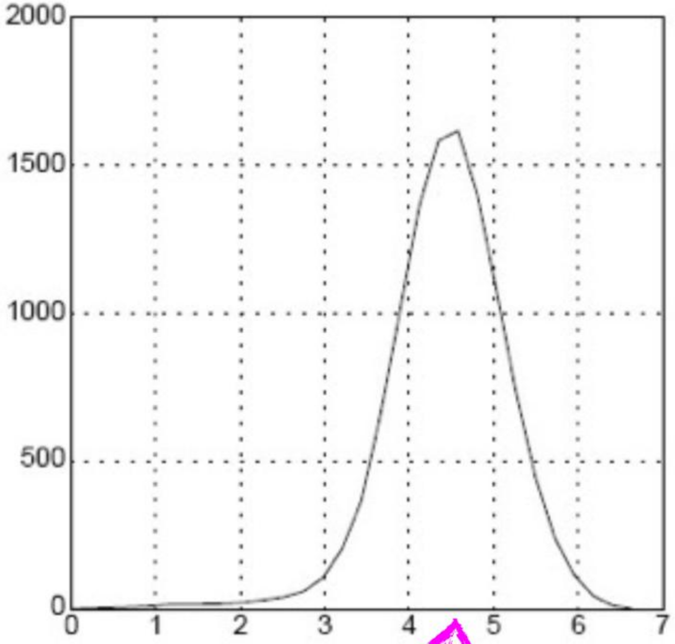
$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right)e^{-\frac{x^2}{2\sigma^2}}$$



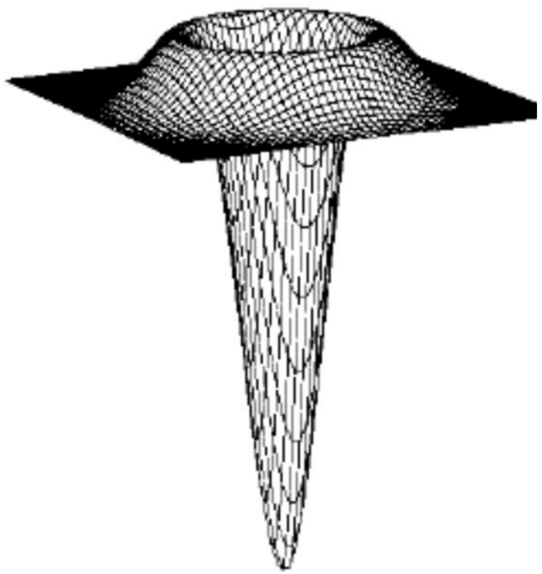
Highest response when the signal has the same **characteristic scale** as the filter

Laplacian of Gaussian for Scale Selection





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

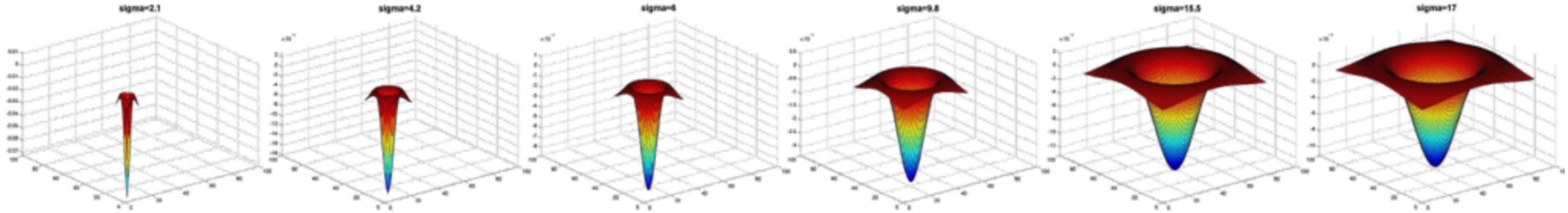


Laplacian of Gaussian

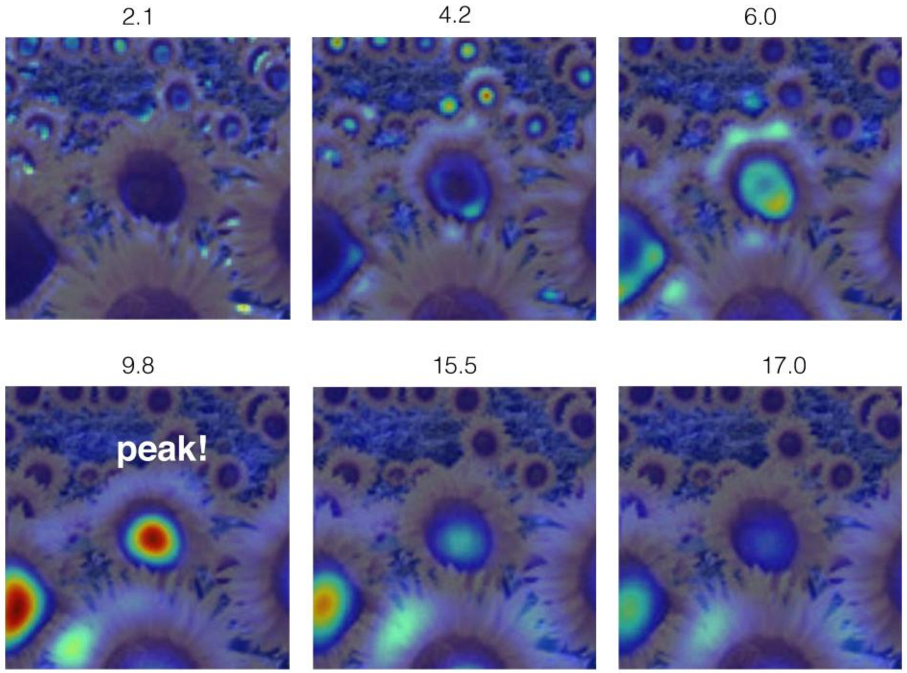
characteristic scale

Search over different scales σ

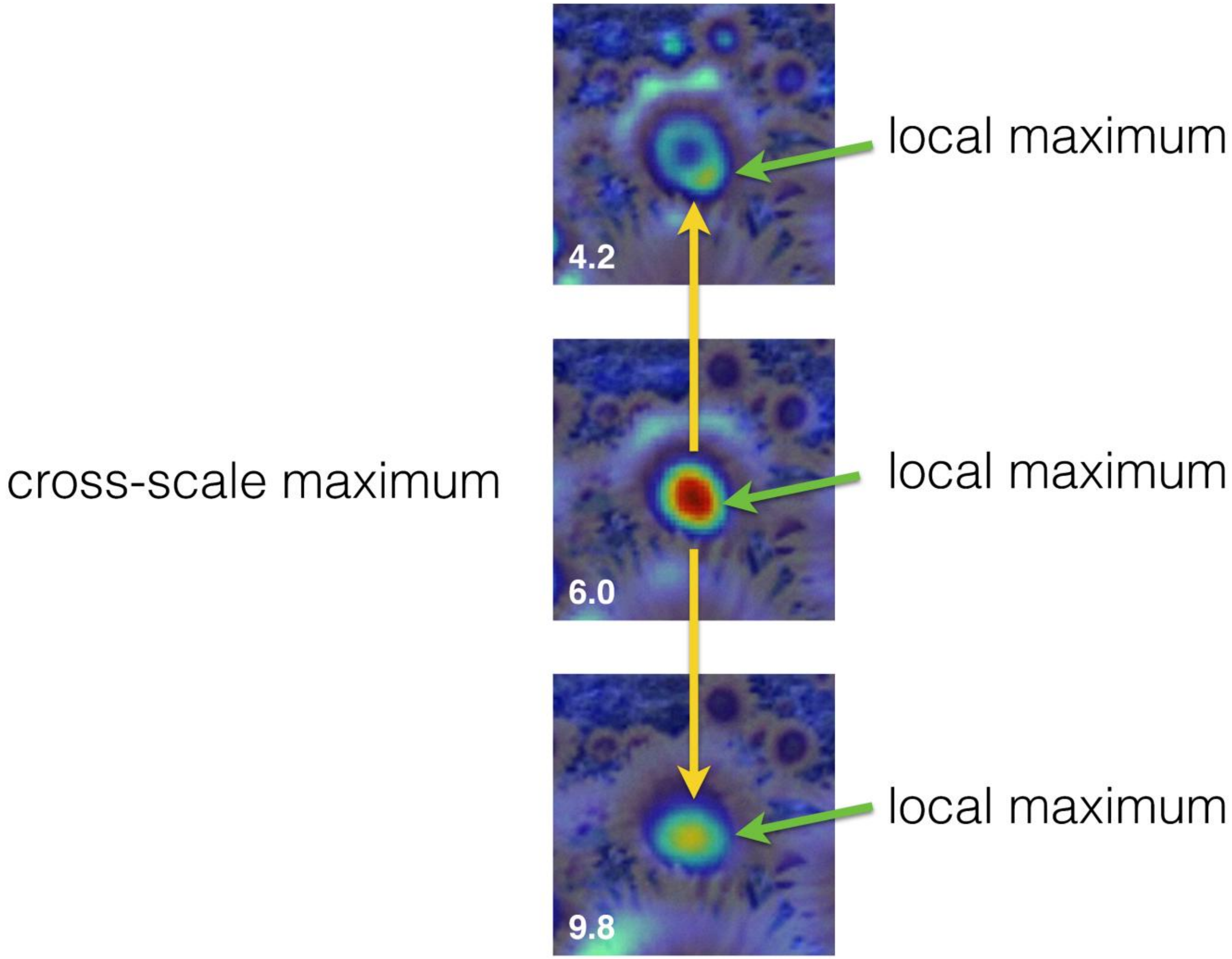
Laplacian of Gaussian for Scale Selection



Multi-scale
2D Blob detection



Laplacian of Gaussian for Scale Selection



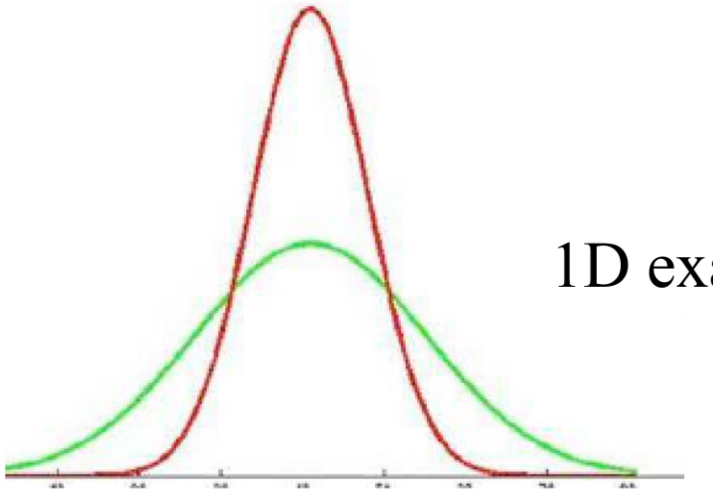
Approximating LoG with DoG

- LoG can be approximate by a Difference of two Gaussians (DoG) at different scales

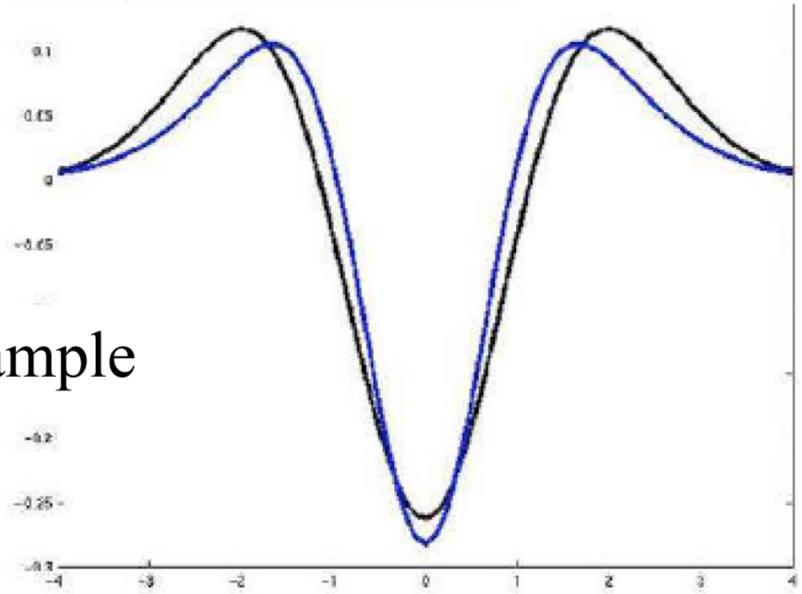
$$\nabla^2 G_\sigma \approx G_{\sigma_1} - G_{\sigma_2}$$

Best approximation when:

$\sigma_1 = \frac{\sigma}{\sqrt{2}}, \ \sigma_2 = \sqrt{2}\sigma$



1D example



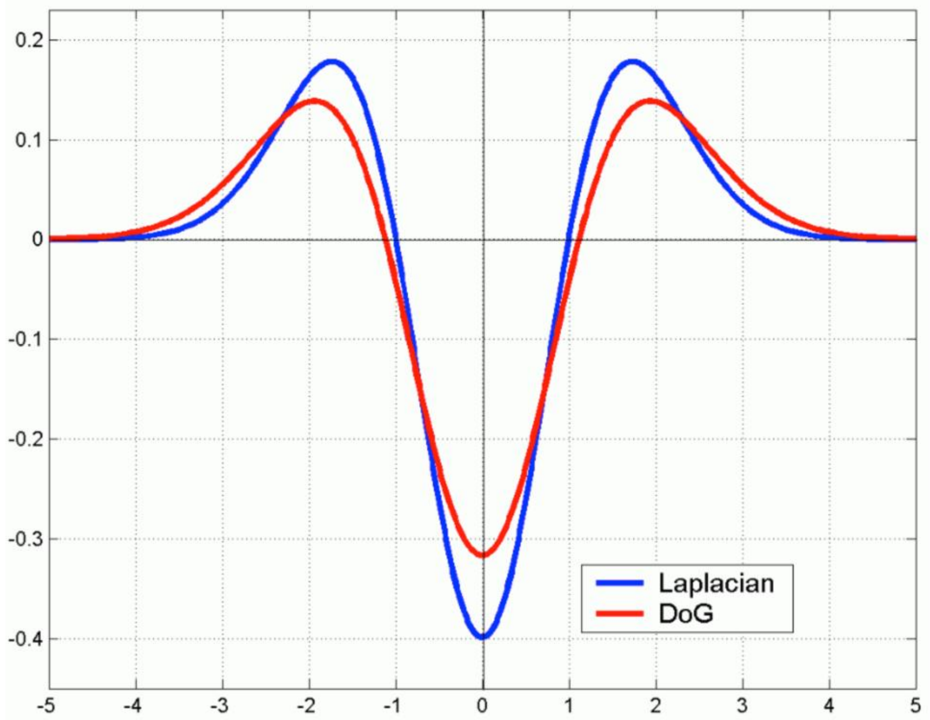
SIFT: Scale-space Extrema Detection

- Difference of Gaussian (DoG)

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

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

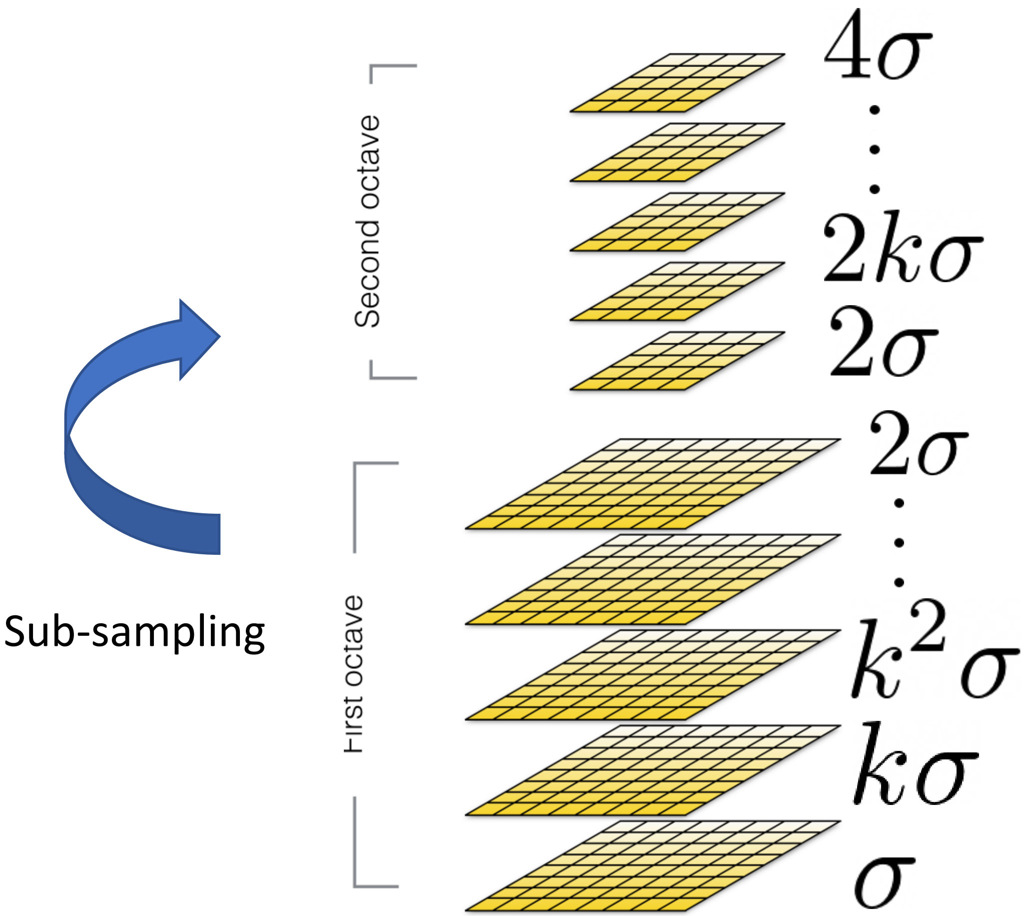


Approximate of Laplacian of Gaussian (efficient to compute)

k is a constant

SIFT: Scale-space Extrema Detection

- Gaussian pyramid



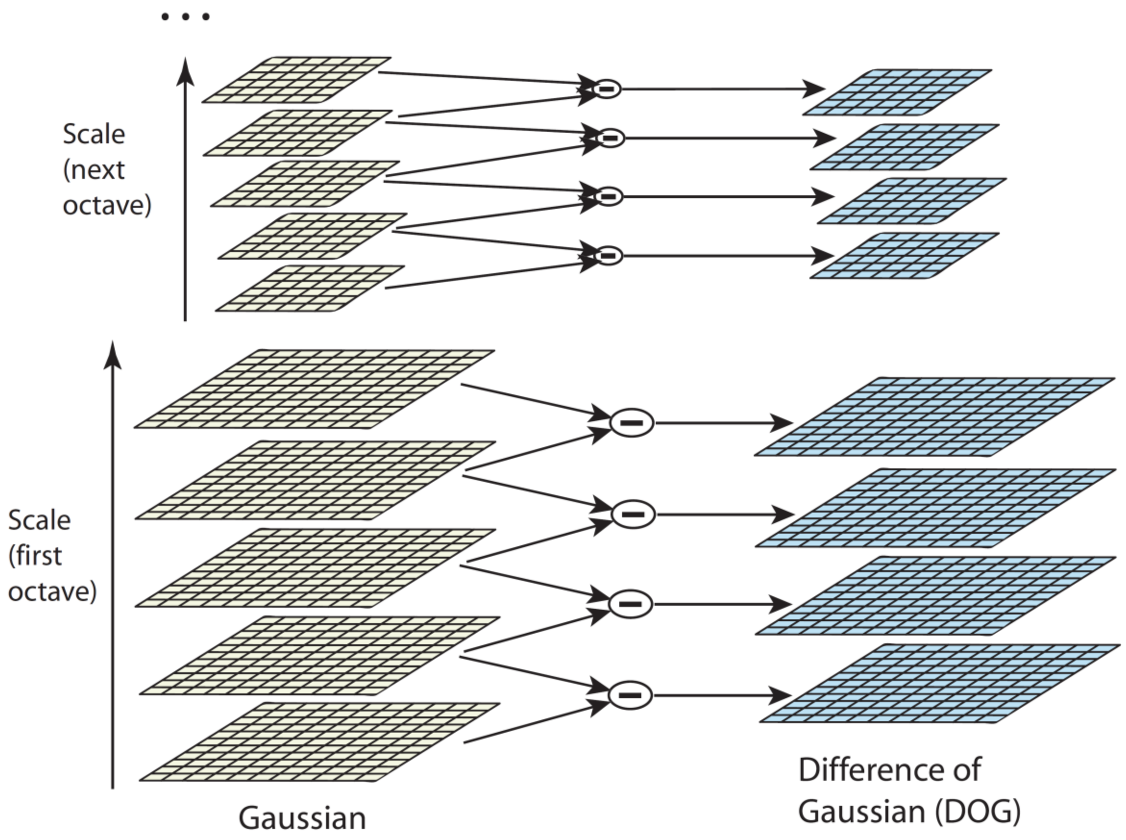
$$k = 2^{\frac{1}{s}}$$

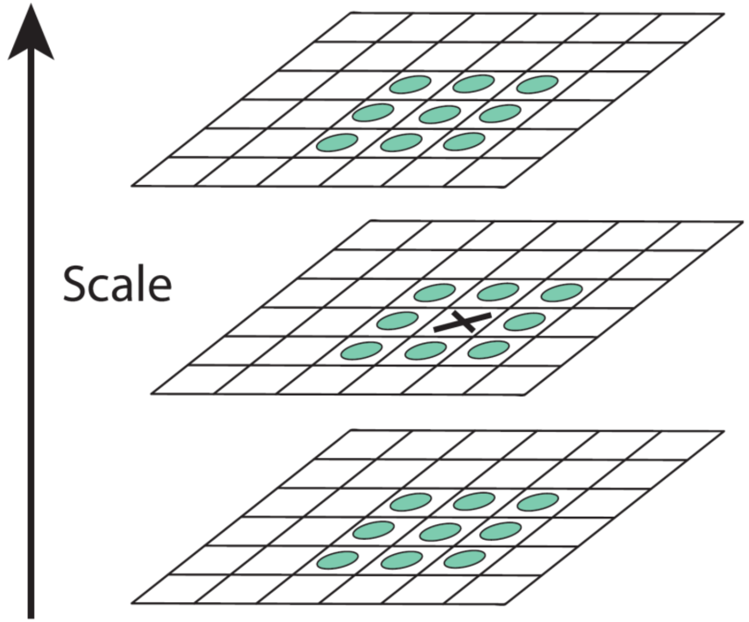
- Gaussian filters

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

- Sub-sampling by a factor of 2
  - Multiple the Gaussian kernel deviation by 2

SIFT: Scale-space Extrema Detection





Maxima and minima of DoG images

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

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

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

Further Reading

- Section 7.1, Computer Vision, Richard Szeliski

- David Lowe, Distinctive Image Features from Scale-Invariant Keypoints. IJCV, 2004 https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

- ORB: An efficient alternative to SIFT or SURF. Rublee et al., ICCV, 2011