Computer Vision & Image Processing CSE 473 / 573

Instructor - Kevin R. Keane, PhD

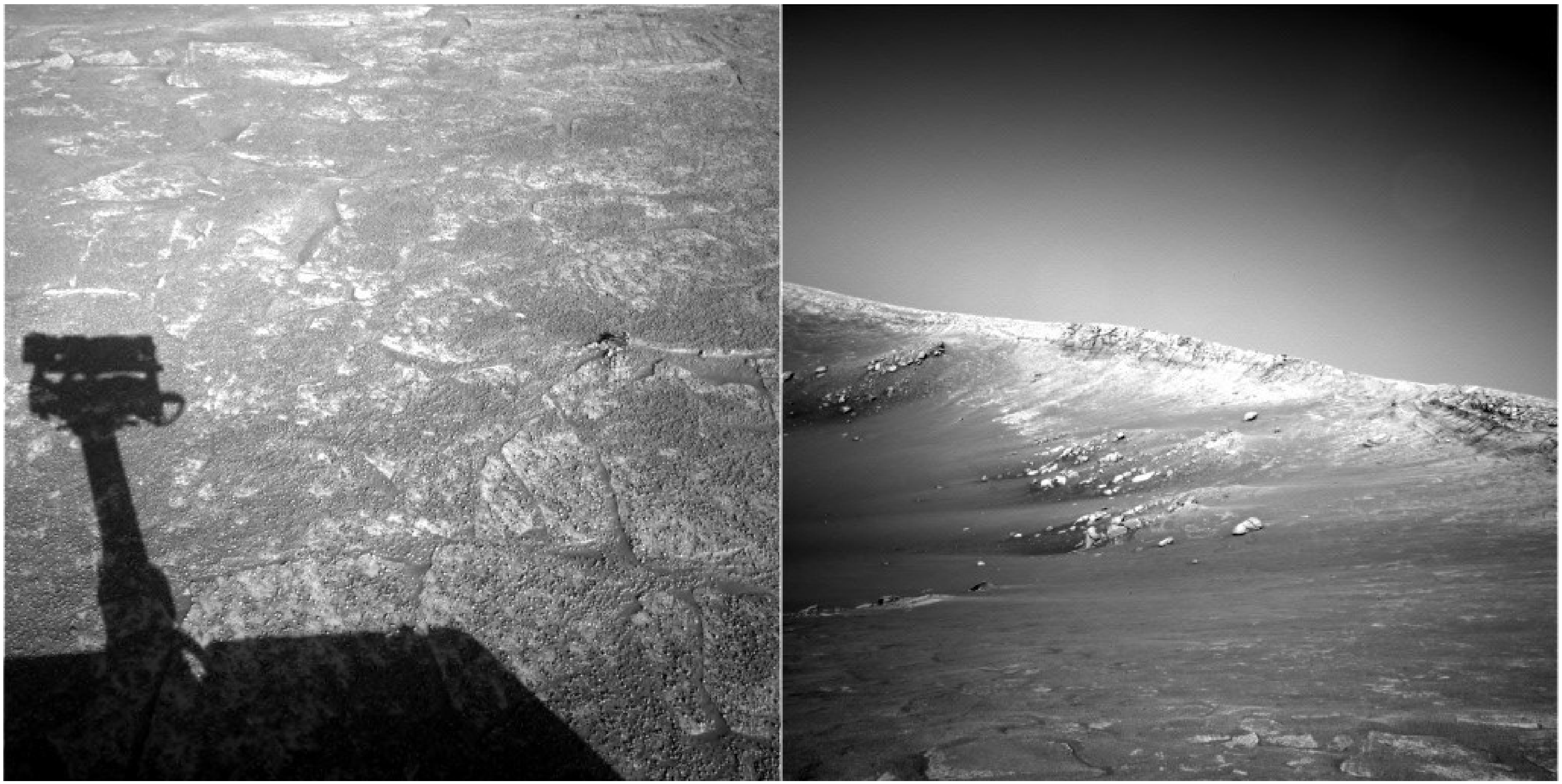
TAs - Radhakrishna Dasari, Yuhao Du, Niyazi Sorkunlu

Lecture 11
September 22, 2017
Local Features

Schedule

- Last class
  - We started local features
- Today
  - More on local features
- Readings for today: Forsyth and Ponce Chapter 5

A hard feature matching problem



NASA Mars Rover images

Overview

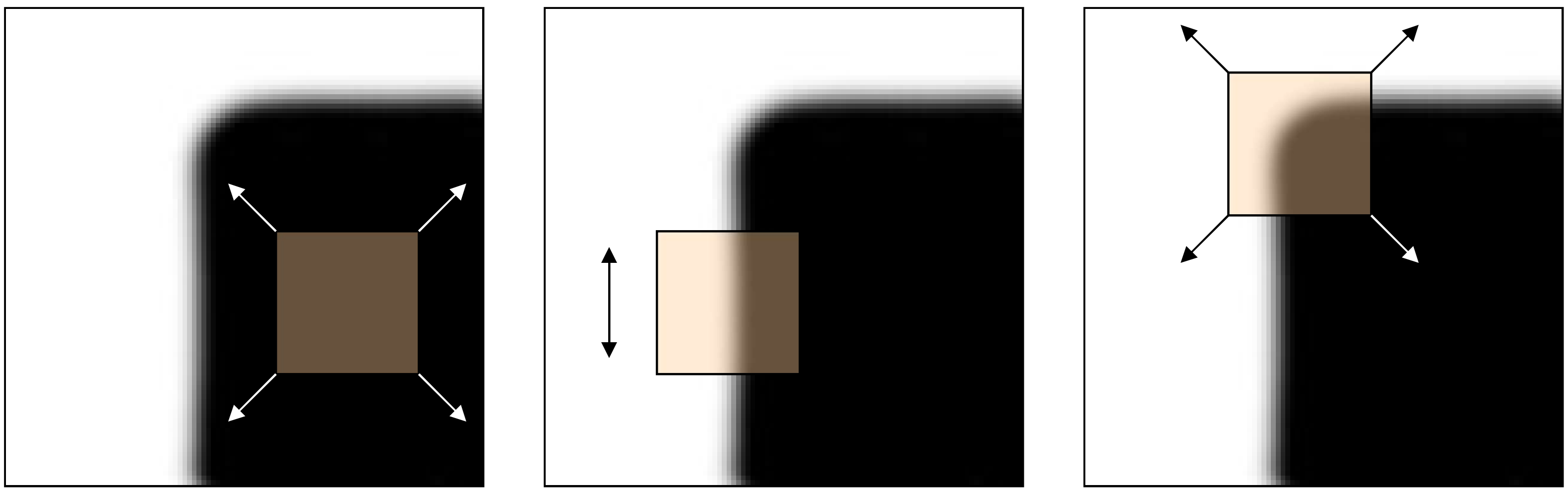
- Eigenvector / eigenvalue review
- Corners (Harris Detector)
- Blobs
- Descriptors

Eigenvectors / eigenvalues

- *Eigen*: German word for *characteristic*
- *Eigen-vector*: characteristic direction
- *Eigen-value*: characteristic magnitude
- See the Wall et.al. in our Linear Algebra folder
  – Huge M *x* N matrices
  – characteristic column response
  – characteristic row response

Corner Detection: Basic Idea

- small window, shifting in any direction should give a ***large change in intensity***
- Eigenvector: direction
- Eigenvalue: magnitude



"flat" region: *no change* in all directions

"edge": *no change* along edge direction

"corner": *big change* in all directions

Source: A. Efros

Harris corner detector summary

- Good corners
  - High contrast
  - Sharp change in edge orientation
- Image features at good corners
  - Large gradients that change direction sharply
    - Will have 2 large eigenvalues
- Compute matrix H by summing over window

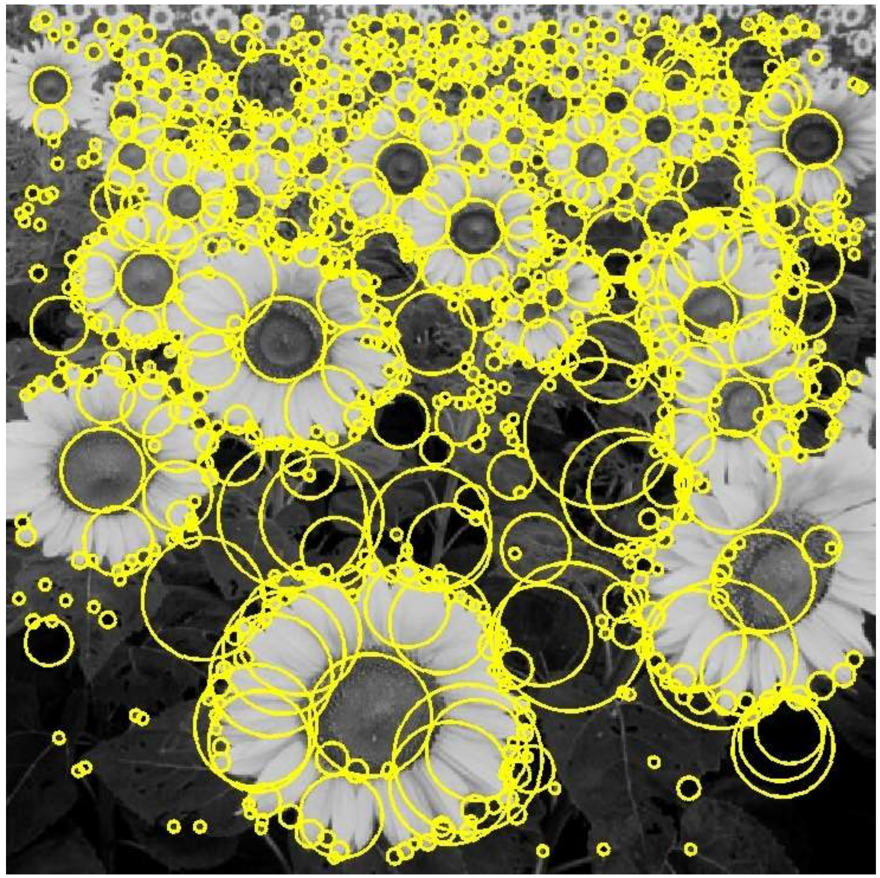
$$\mathcal{H} = \sum_{window} \{(\nabla I)(\nabla I)^T\}$$

$$\approx \sum_{window} \left\{ \begin{array}{cc} (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial x} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \\ (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial y} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \end{array} \right\}$$

Overview

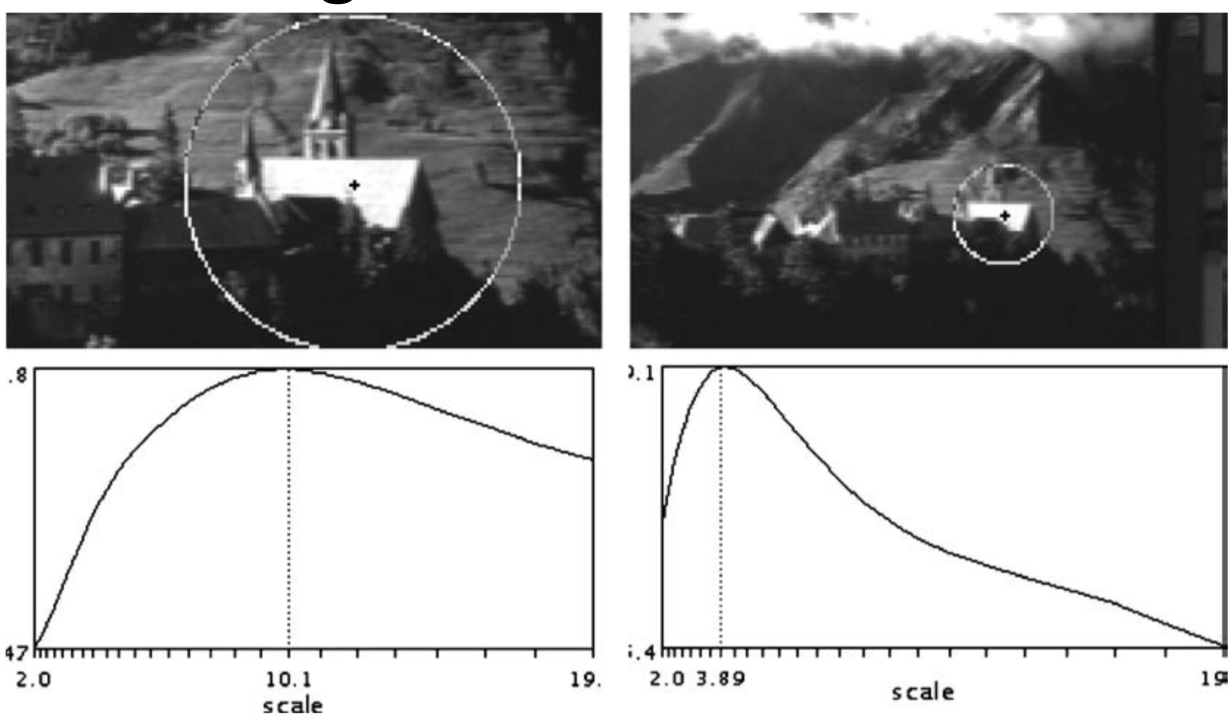
- Corners (Harris Detector)
- Blobs
- Descriptors

Blob detection with scale selection



Achieving scale covariance

- Goal: independently detect corresponding regions in scaled versions of the same image
- Need *scale selection* mechanism for finding characteristic region size that is *covariant* with the image transformation

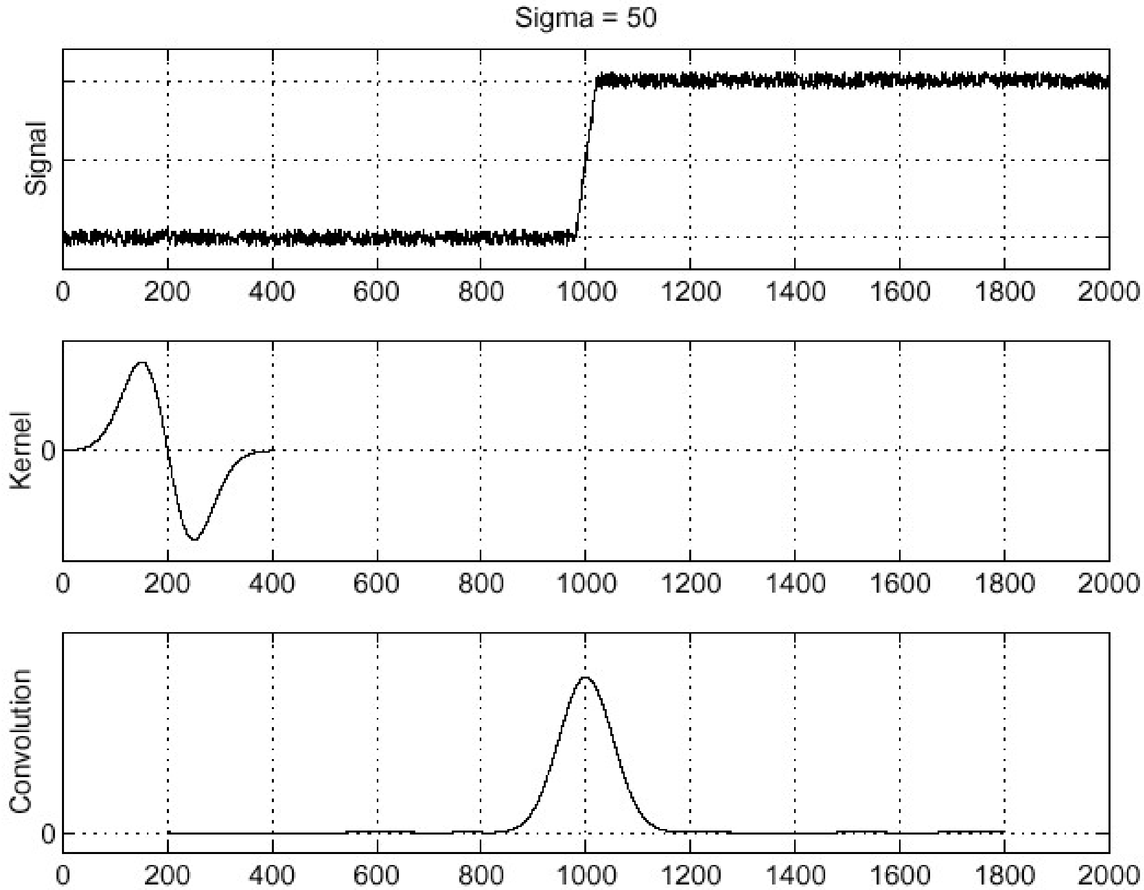


Recall: Edge detection

f

$\frac{d}{dx} g$

$f * \frac{d}{dx} g$



Edge

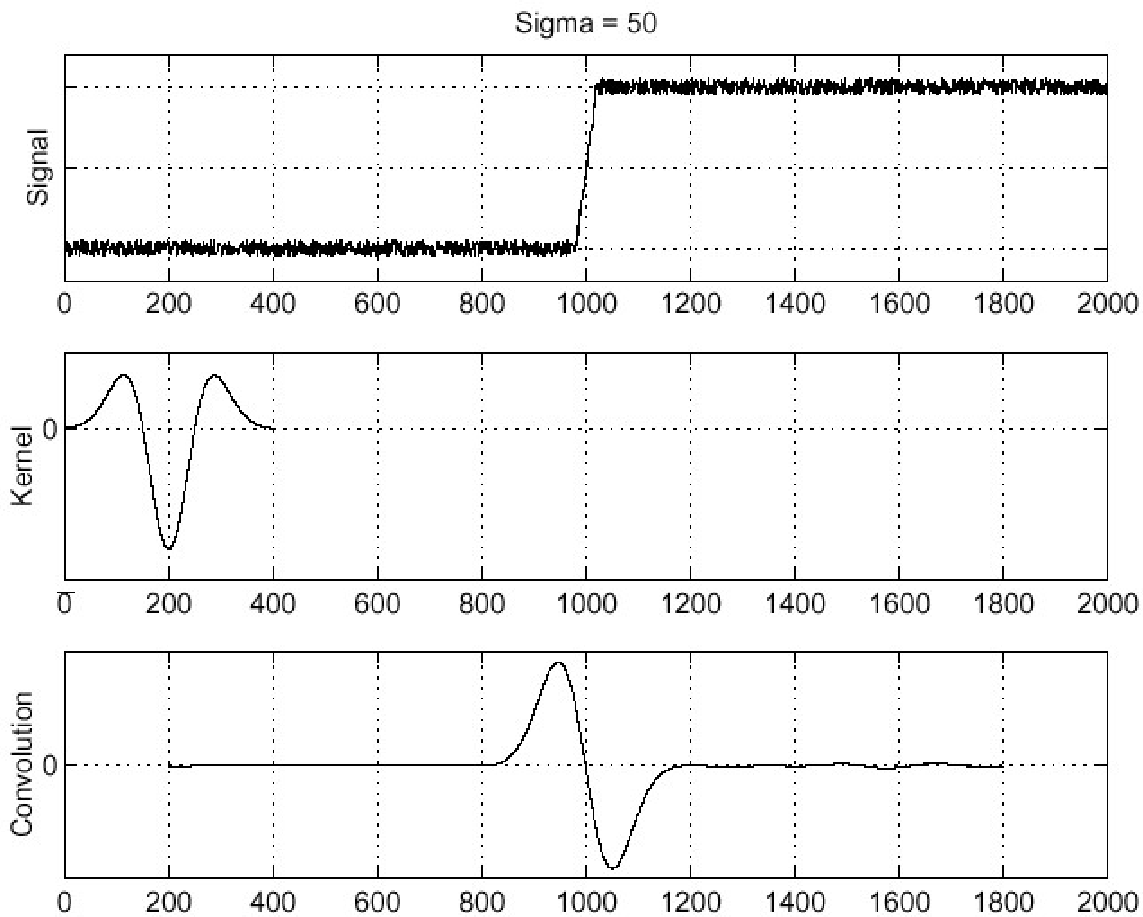
Derivative of Gaussian

Edge = maximum of derivative

Source: S. Seitz

Edge detection, Take 2

f



Edge

$$\frac{d^2}{dx^2} g$$

Second derivative of Gaussian (Laplacian)

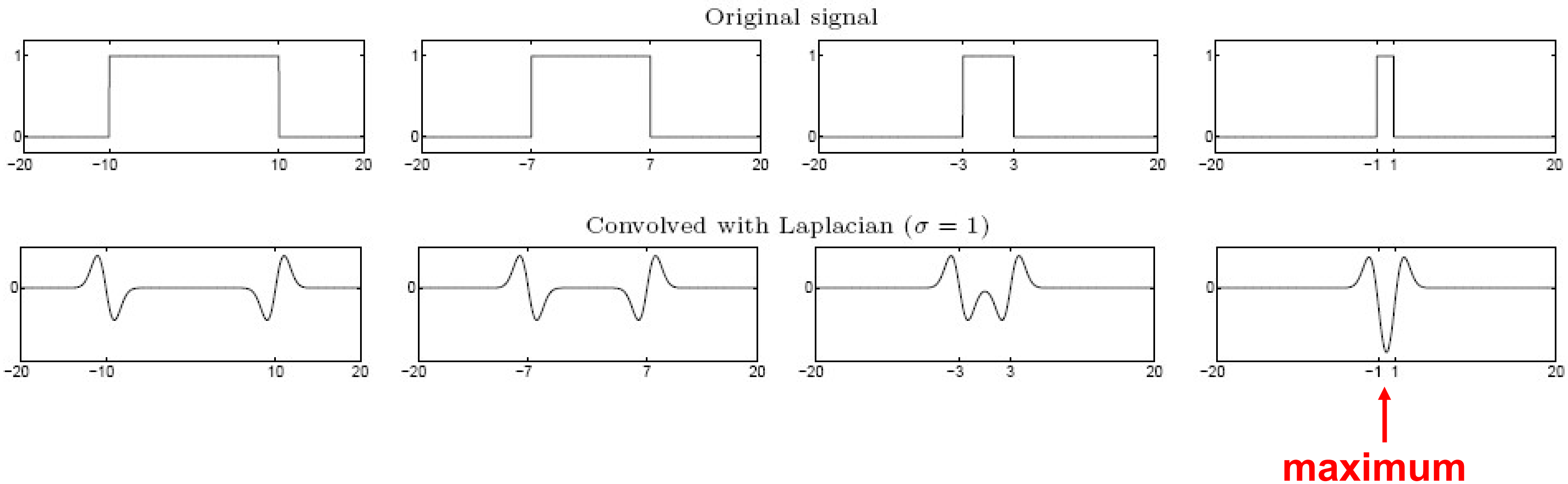
$$f * \frac{d^2}{dx^2} g$$

Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

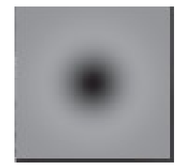
- Edge = ripple
- Blob = superposition of two ripples



Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is "matched" to the scale of the blob

Estimating scale - I

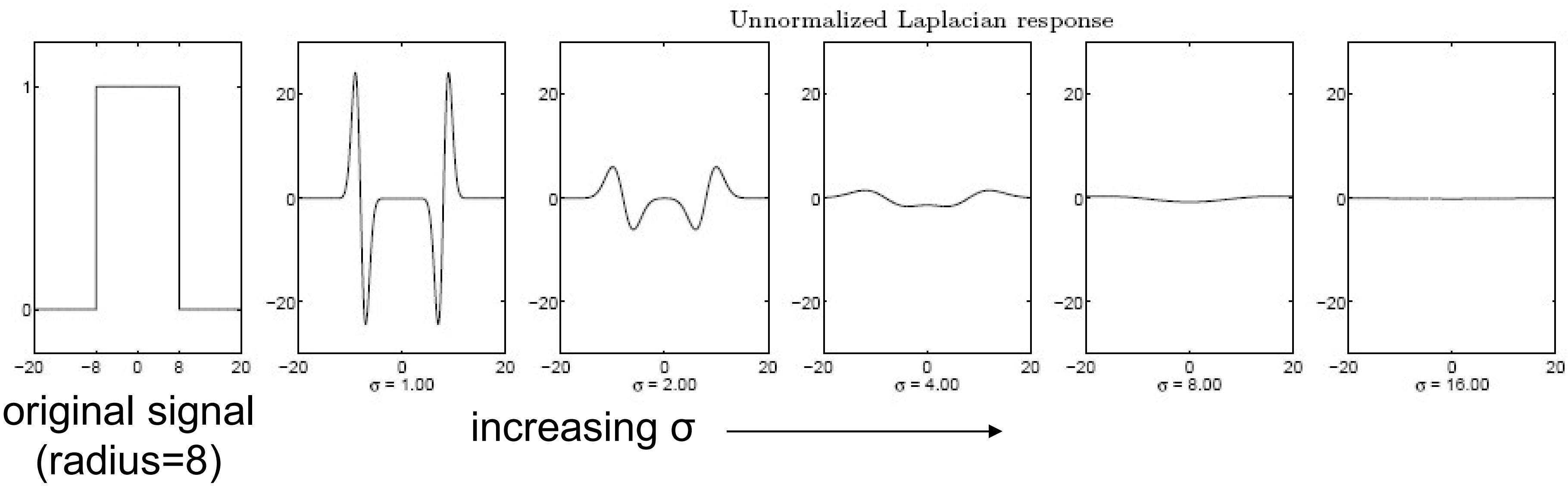
- Assume we have detected a corner
- How big is the neighborhood?
- Use Laplacian of Gaussian filter
  - Details on next slide
  - Kernel looks like fuzzy dark blob on pale light foreground
  - Scale (sigma) of Gaussian gives size of dark, light blob



- Strategy
  - Apply Laplacian of Gaussian at different scales at corner
    - response is a function of scale
  - Choose the scale that gives the largest response
    - the scale at which the neighborhood looks “most like” a fuzzy blob
  - This is covariant

Scale selection

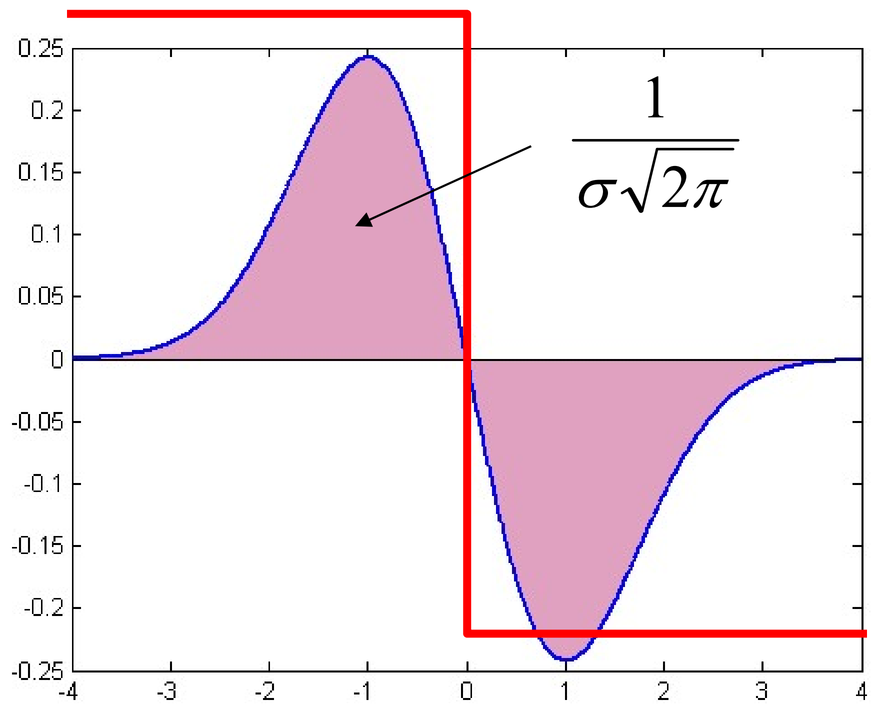
- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response
- However, Laplacian response decays as scale increases:



Why does this happen?

Scale normalization

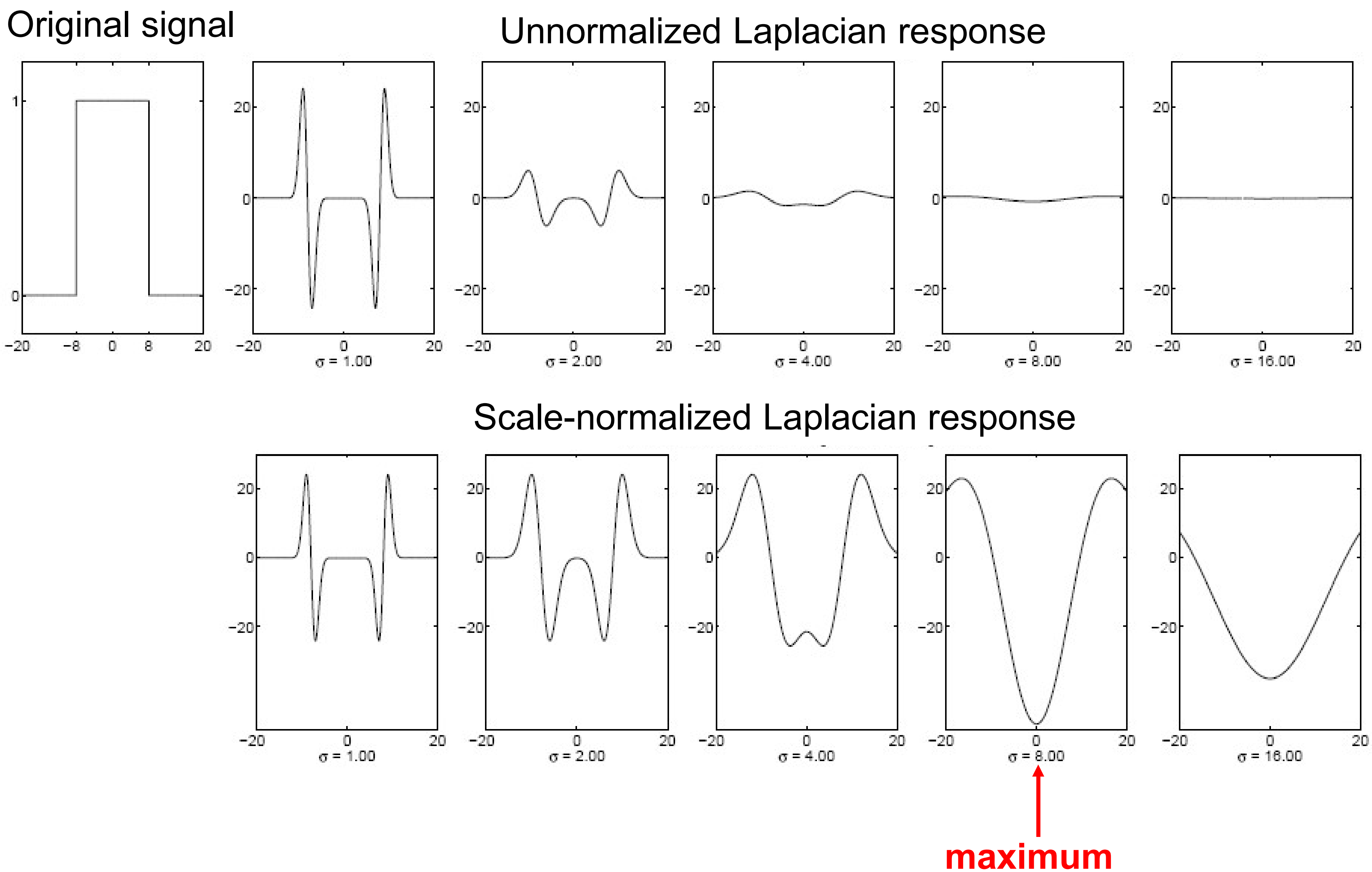
- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases



Scale normalization

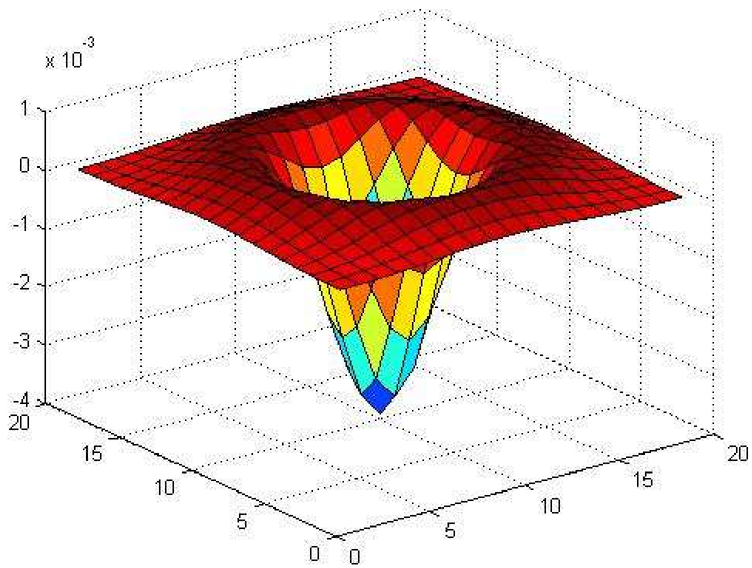
- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases
- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

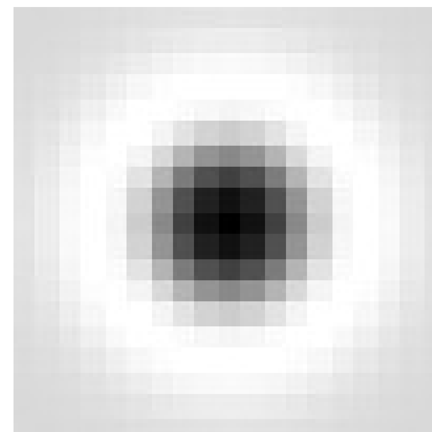
Effect of scale normalization



Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

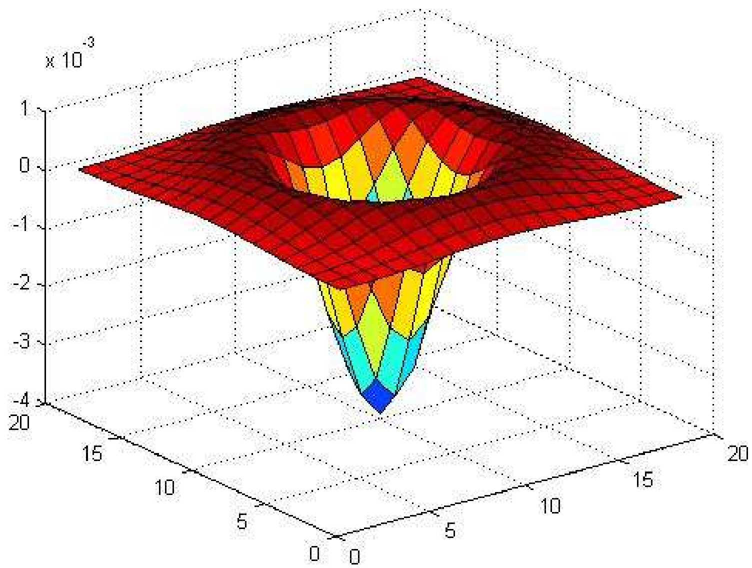


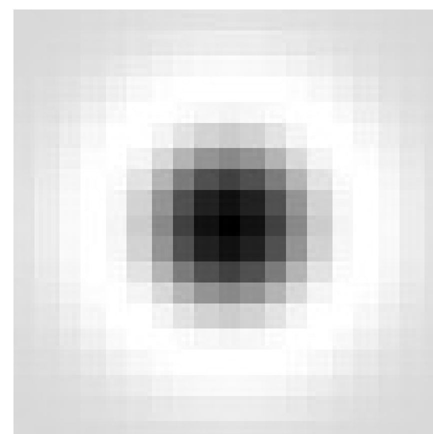


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} = \frac{1}{\pi \sigma^4}\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

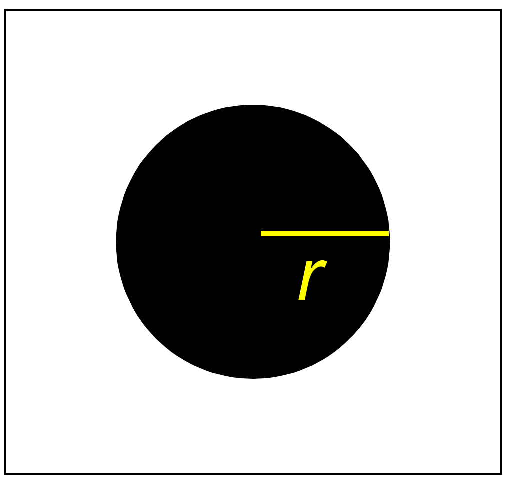




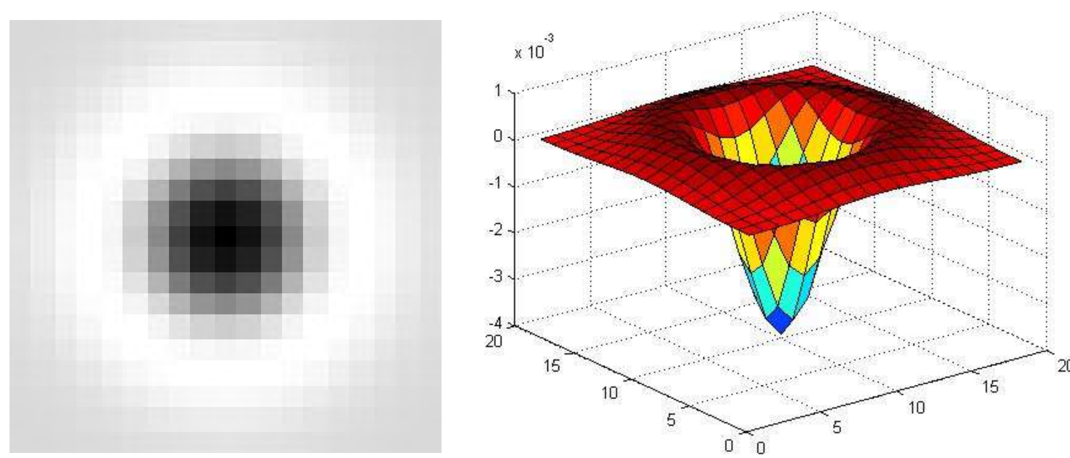
Scale-normalized: $\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$

Scale selection

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



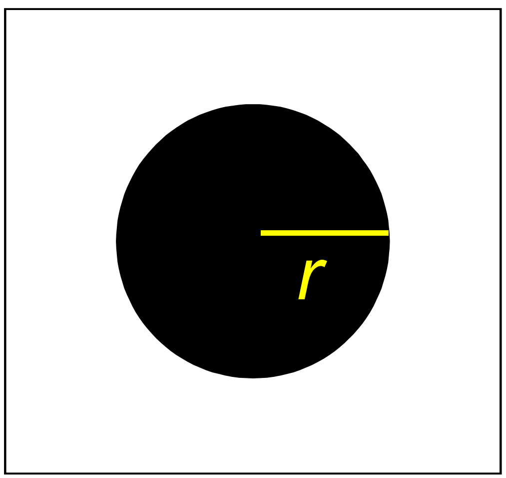
image



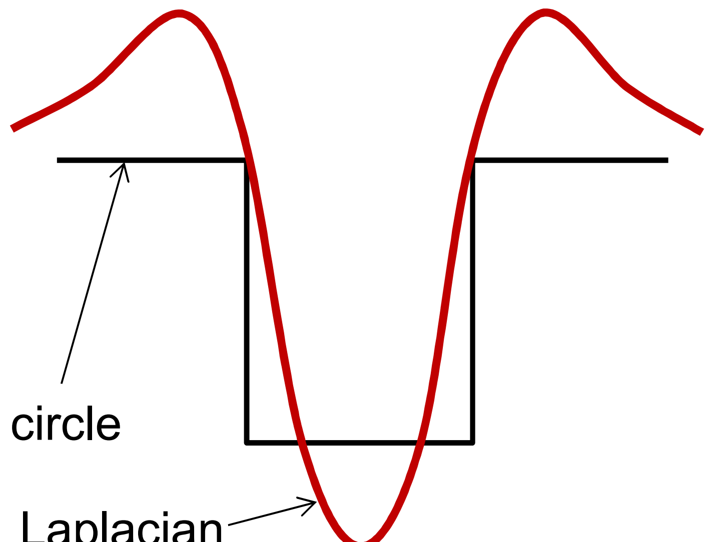
Laplacian

Scale selection

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- Zeros of Laplacian is given by (up to scale): $$\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) = 0$$
- Therefore, the maximum response occurs at $$\sigma = r / \sqrt{2}.$$

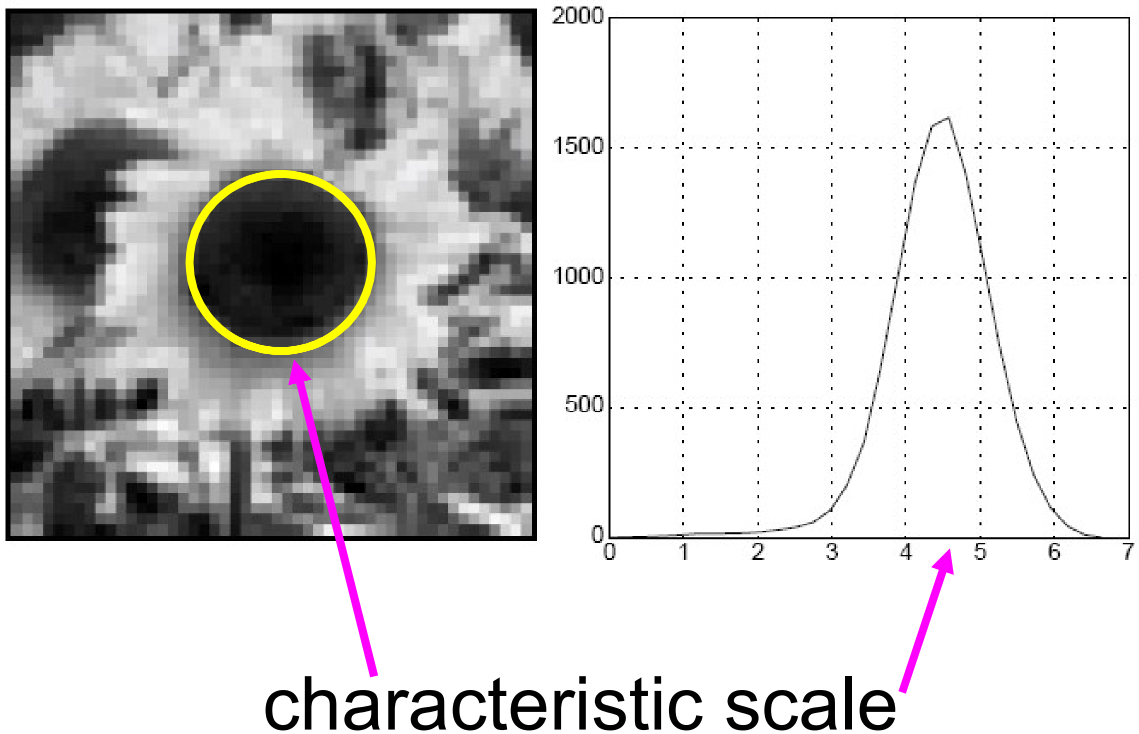


image



Characteristic scale

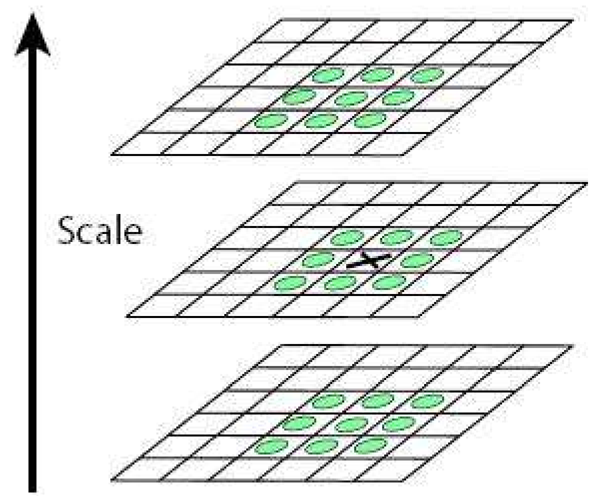
- We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center



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

Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space

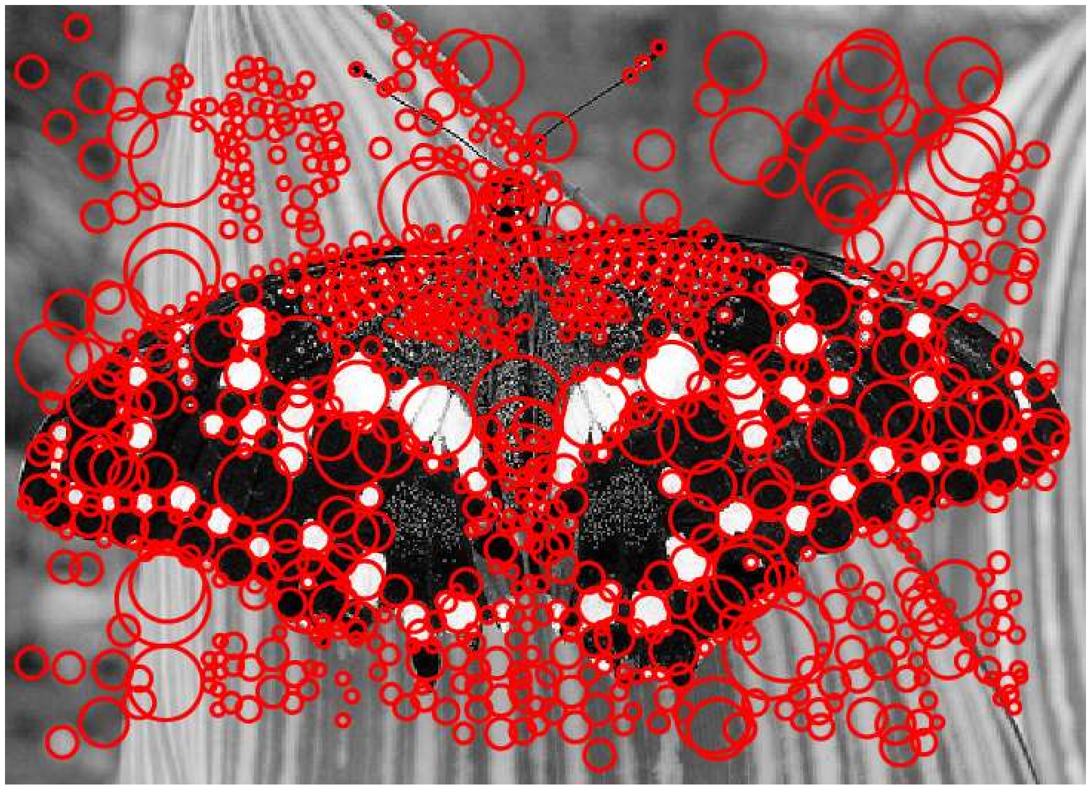


Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

Scale-space blob detector: Example



Efficient implementation

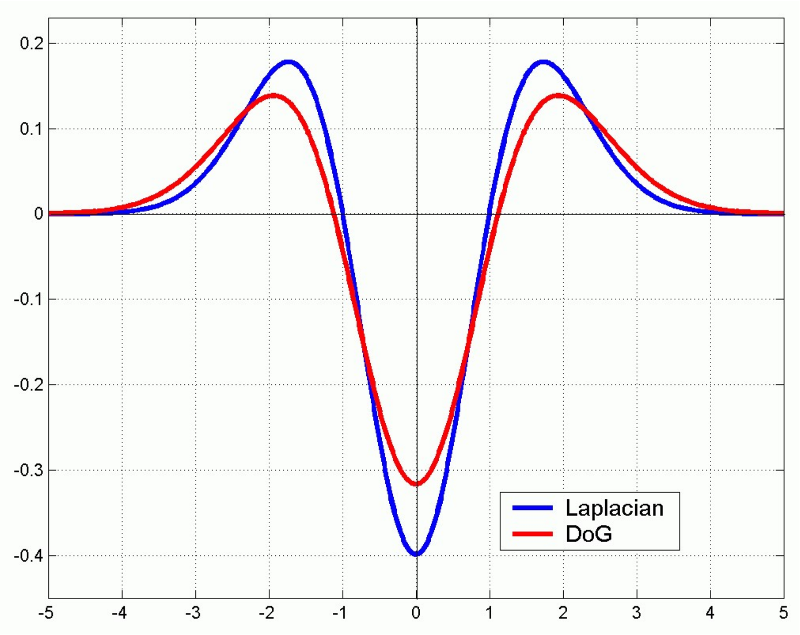
- Approximating the Laplacian with a difference of Gaussians:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

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

(Difference of Gaussians)

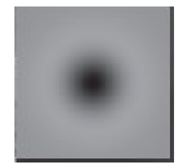


Invariance and covariance properties

- Laplacian (blob) response is *invariant* w.r.t. rotation and scaling
- Blob location is *covariant* w.r.t. rotation and scaling

Estimating scale - summary

- Assume we have detected a corner
- How big is the neighborhood?
- Use Laplacian of Gaussian filter
  - Details on next slide
  - Kernel looks like fuzzy dark blob on pale light foreground
  - Scale (sigma) of Gaussian gives size of dark, light blob



- Strategy
  - Apply Laplacian of Gaussian at different scales at corner
    - response is a function of scale
  - Choose the scale that gives the largest response
    - the scale at which the neighborhood looks “most like” a fuzzy blob
  - This is covariant

Estimating scale - summary

- Laplacian of a function

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

- Gaussian $G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-x^2 - y^2}{2\sigma^2}\right)}$

- So Laplacian of Gaussian

$$\nabla^2 G_\sigma(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) G_\sigma(x, y)$$

- Convolve with image

$$\nabla^2_\sigma \mathcal{I}(x, y) = \left(\nabla^2 G_\sigma(x, y)\right) * * \mathcal{I}(x, y)$$

Overview

- Corners (Harris Detector)
- Blobs
- Descriptors (SIFT)

Scale Invariant Feature Transform

David Lowe IJCV 2004

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

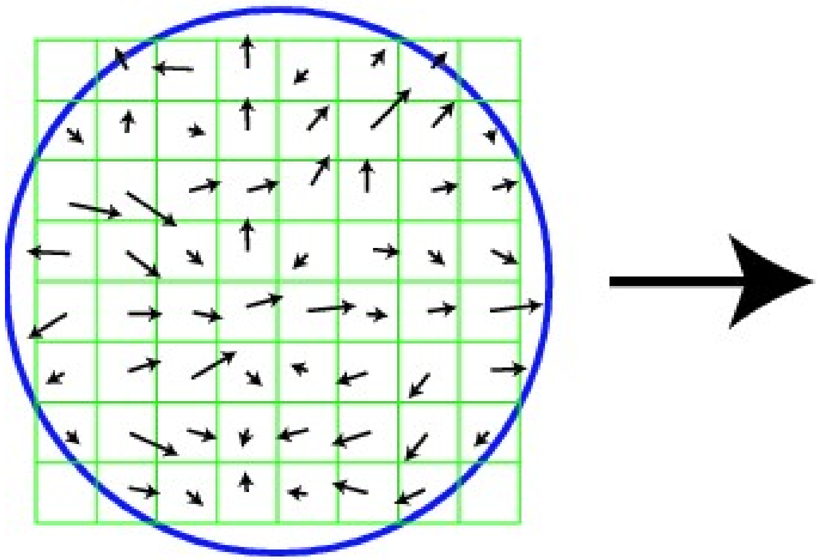
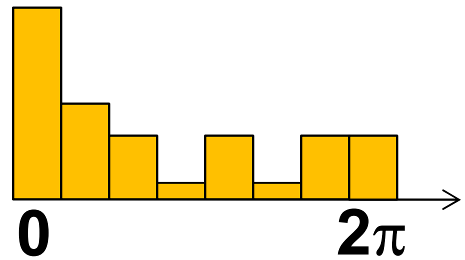
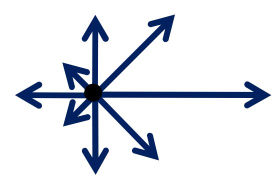


Image gradients

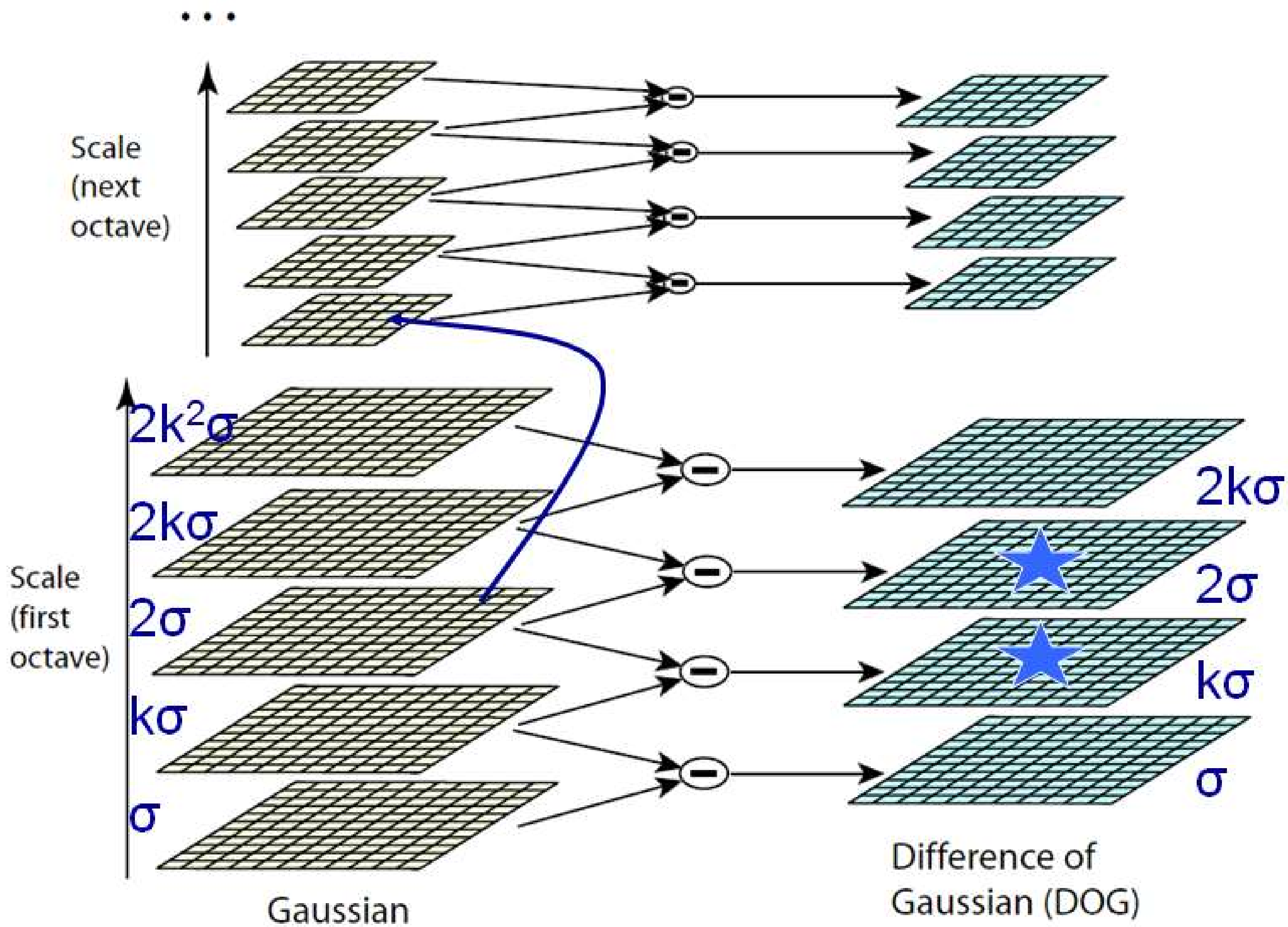


angle histogram



Adapted from slide by David Lowe

Efficient implementation



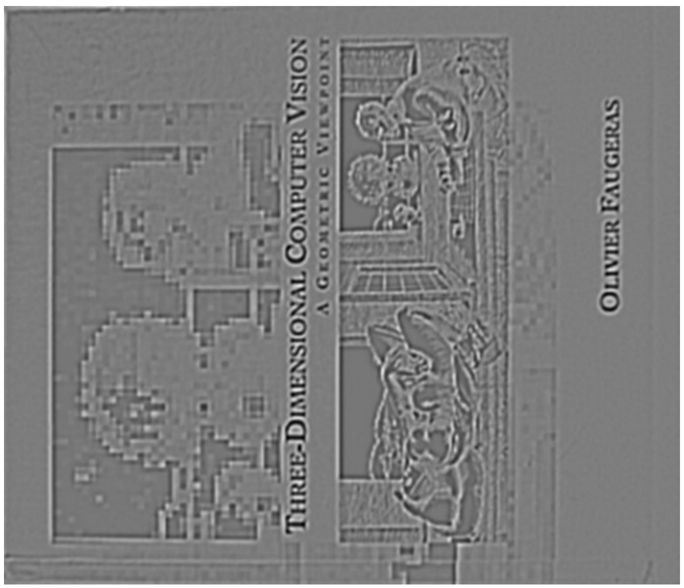
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Scale space images

first octave second octave third octave fourth octave

Difference-of-Gaussian images

...



...

first octave

...

...

second octave

...

...

third octave

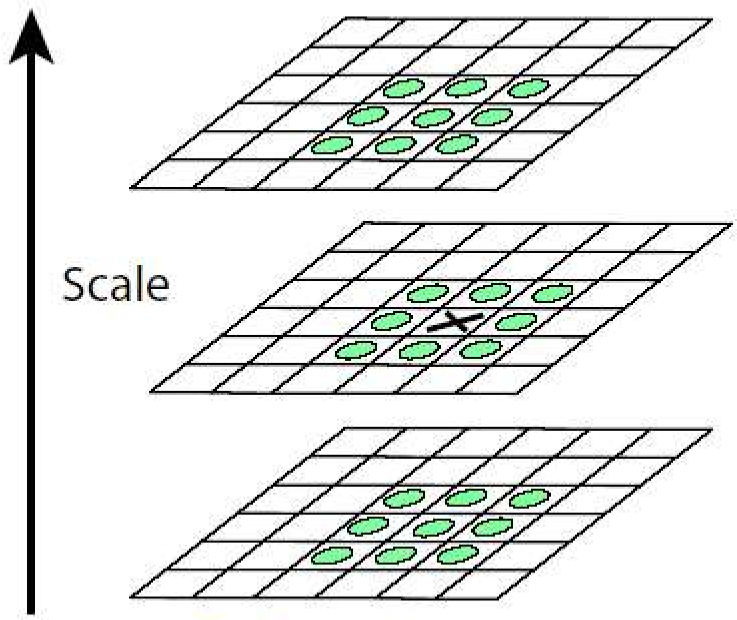
...

...

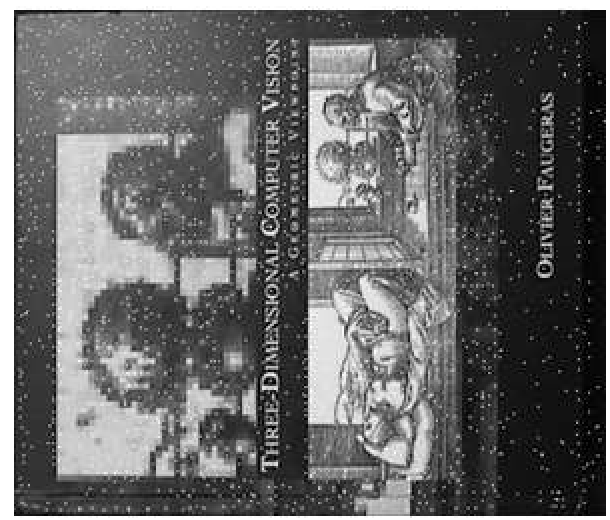
fourth octave

Finding extrema

- Sample point is selected only if it is a minimum or a maximum of these points



DoG scale space



Extrema in this image

Localization

- 3D quadratic function is fit to the local sample points
- Start with Taylor expansion with sample point as the origin $D(\mathrm{X}) = D + \frac{\partial D^T}{\partial \mathrm{X}}\mathrm{X} + \frac{1}{2}\mathrm{X}^T \frac{\partial^2 D}{\partial \mathrm{X}^2}\mathrm{X}$
  - where $\mathrm{X} = (x, y, \sigma)^T$
- Take the derivative with respect to *X*, and set it to 0, giving $0 = \frac{\partial D}{\partial X} + \frac{\partial^2 D}{\partial X^2}\hat{X}$
- $\hat{\mathrm{X}} = -\frac{\partial^2 D^{-1}}{\partial \mathrm{X}^2}\frac{\partial D}{\partial \mathrm{X}}$ is the location of the keypoint
- This is a 3x3 linear system

Localization

$$\begin{bmatrix} \frac{\partial^2 D}{\partial \sigma^2} & \frac{\partial^2 D}{\partial \sigma y} & \frac{\partial^2 D}{\partial \sigma x} \\ \frac{\partial^2 D}{\partial \sigma y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial yx} \\ \frac{\partial^2 D}{\partial \sigma x} & \frac{\partial^2 D}{\partial yx} & \frac{\partial^2 D}{\partial x^2} \end{bmatrix} \begin{bmatrix} \sigma \\ y \\ x \end{bmatrix} = - \begin{bmatrix} \frac{\partial D}{\partial \sigma} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial x} \end{bmatrix}$$

- Derivatives approximated by finite differences,
  - example:

$$\frac{\partial D}{\partial \sigma} = \frac{D_{k+1}^{i,j} - D_{k-1}^{i,j}}{2}$$

$$\frac{\partial^2 D}{\partial \sigma^2} = \frac{D_{k-1}^{i,j} - 2D_{k}^{i,j} + D_{k+1}^{i,j}}{1}$$

$$\frac{\partial^2 D}{\partial \sigma y} = \frac{(D_{k+1}^{i+1,j} - D_{k-1}^{i+1,j}) - (D_{k+1}^{i-1,j} - D_{k-1}^{i-1,j})}{4}$$

- If *X* is > 0.5 in any dimension, process repeated

Filtering

- Contrast (use prev. equation): $$D(\hat{X}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X}\hat{X}$$
  - If | *D(X)* | < 0.03, throw it out
- Edgeiness:
  - Use ratio of principal curvatures to throw out poorly defined peaks
  - Curvatures come from Hessian: $$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$
  - Ratio of *Trace(H)*^{2} and *Determinant(H)*

$$Tr(H) = D_{xx} + D_{yy}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2$$

  - If ratio > *(r+1)*^{2}*/(r)*, throw it out (SIFT uses r=10)

Orientation assignment

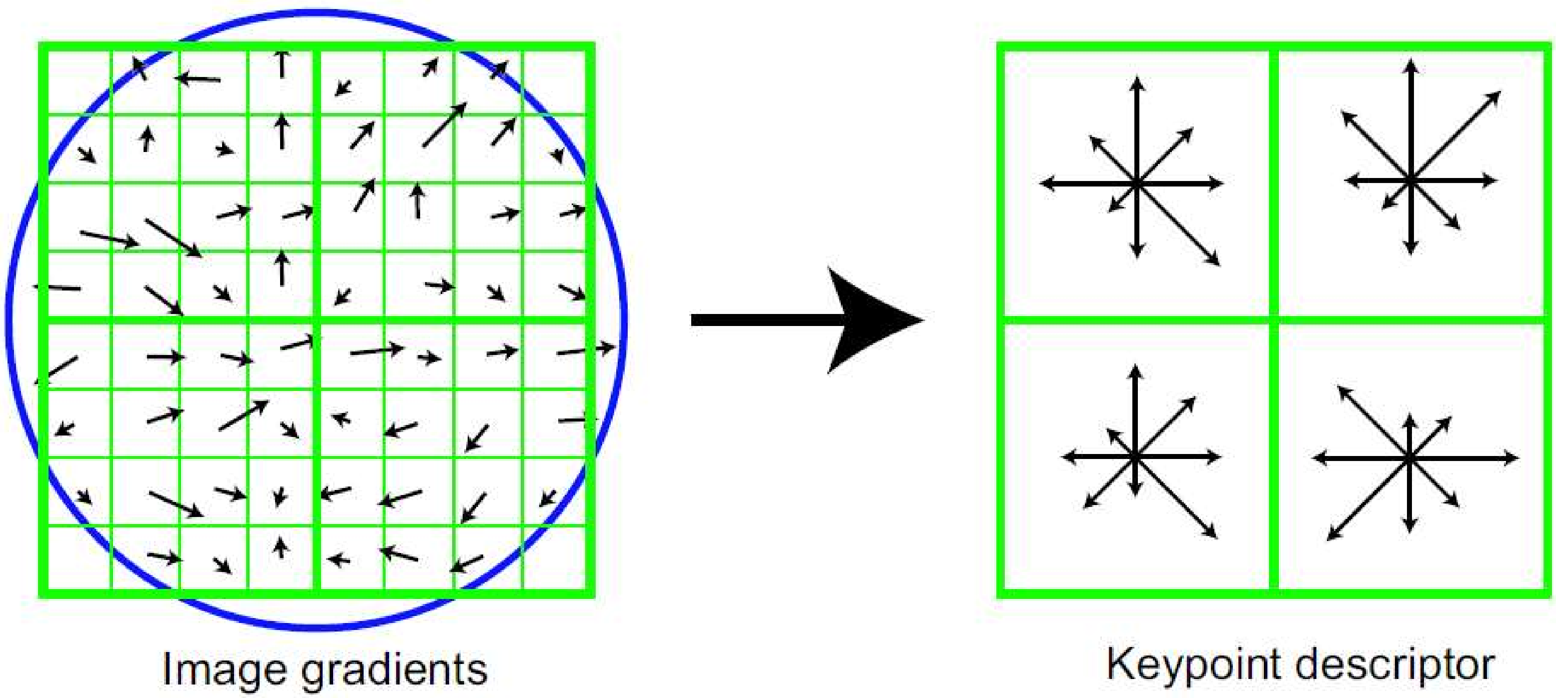
- Descriptor computed relative to keypoint's orientation achieves rotation invariance
- Precomputed along with mag. for all levels (useful in descriptor computation)

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = a\tan 2((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

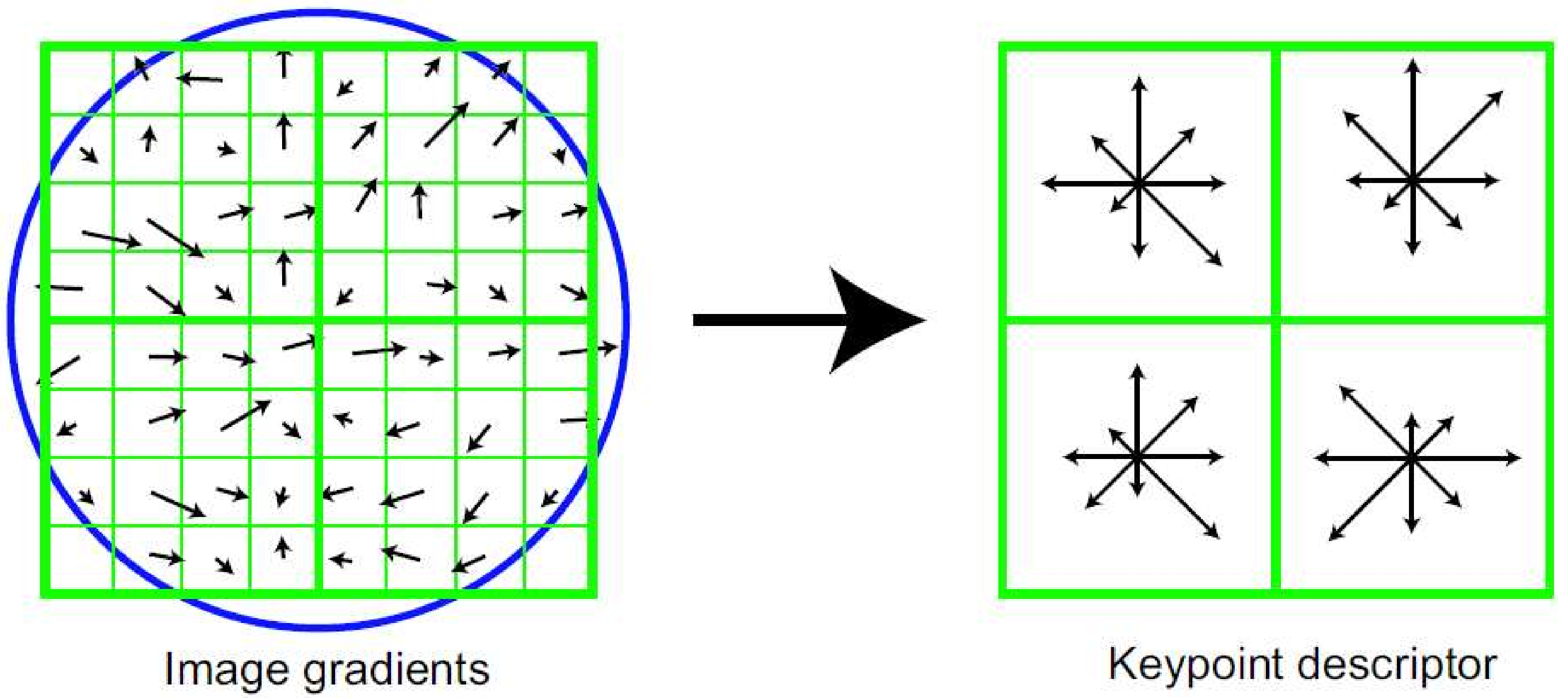
- Multiple orientations assigned to keypoints from an orientation histogram
  - Significantly improve stability of matching

Descriptor



- Descriptor has 3 dimensions *(x,y,ϑ)*
- Orientation histogram of gradient magnitudes
- Position and orientation of each gradient sample rotated relative to keypoint orientation

Descriptor



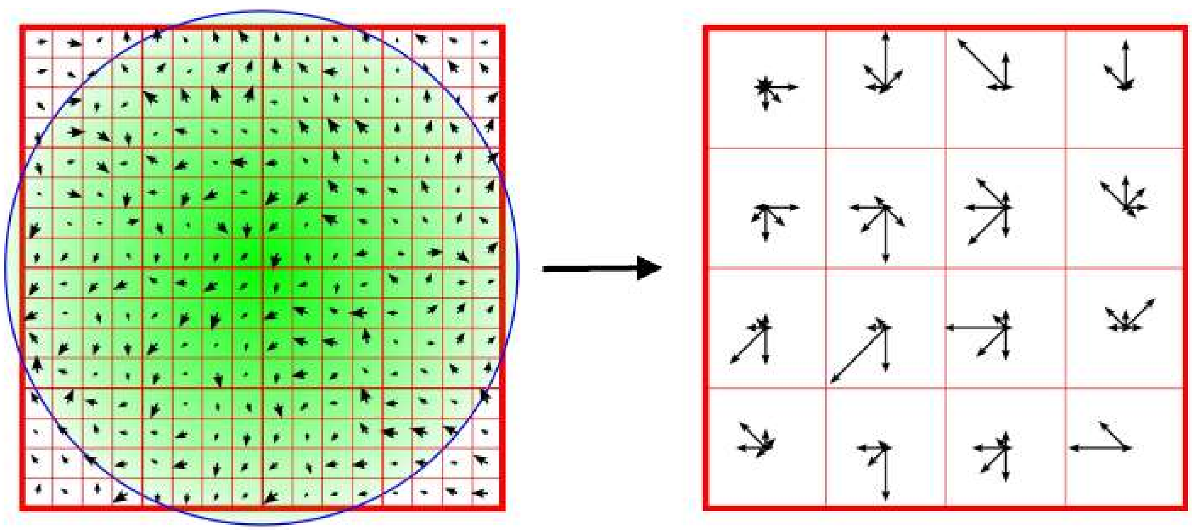
- Weight magnitude of each sample point by Gaussian weighting function
- Distribute each sample to adjacent bins by trilinear interpolation (avoids boundary effects)

Descriptor

- Best results achieved with 4x4x8 = 128 descriptor size
- Normalize to unit length
  - Reduces effect of illumination change
- Cap each element to 0.2, normalize again
  - Reduces non-linear illumination changes
  - 0.2 determined experimentally

Orientation Histogram

- 4x4 spatial bins (16 bins total)
- Gaussian center-weighting
- 8-bin orientation histogram per bin
- 8 x 16 = 128 dimensions total
- Normalized to unit norm



SIFT features

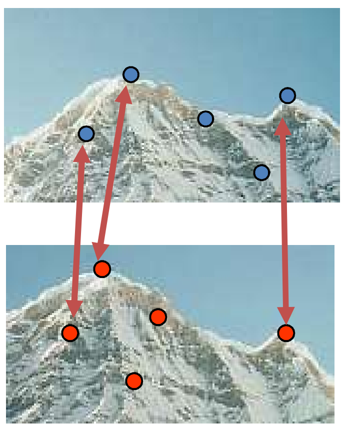
- Very strong record of effectiveness in matching applications
  - use orientations to suppress intensity change effects
  - use histograms so neighborhood need not be exactly localized
  - weight large gradients higher than small gradients
  - Weighting processes are different
  - SIFT features behave very well using nearest neighbors matching
    - i.e. the nearest neighbor to a query patch is usually a matching patch

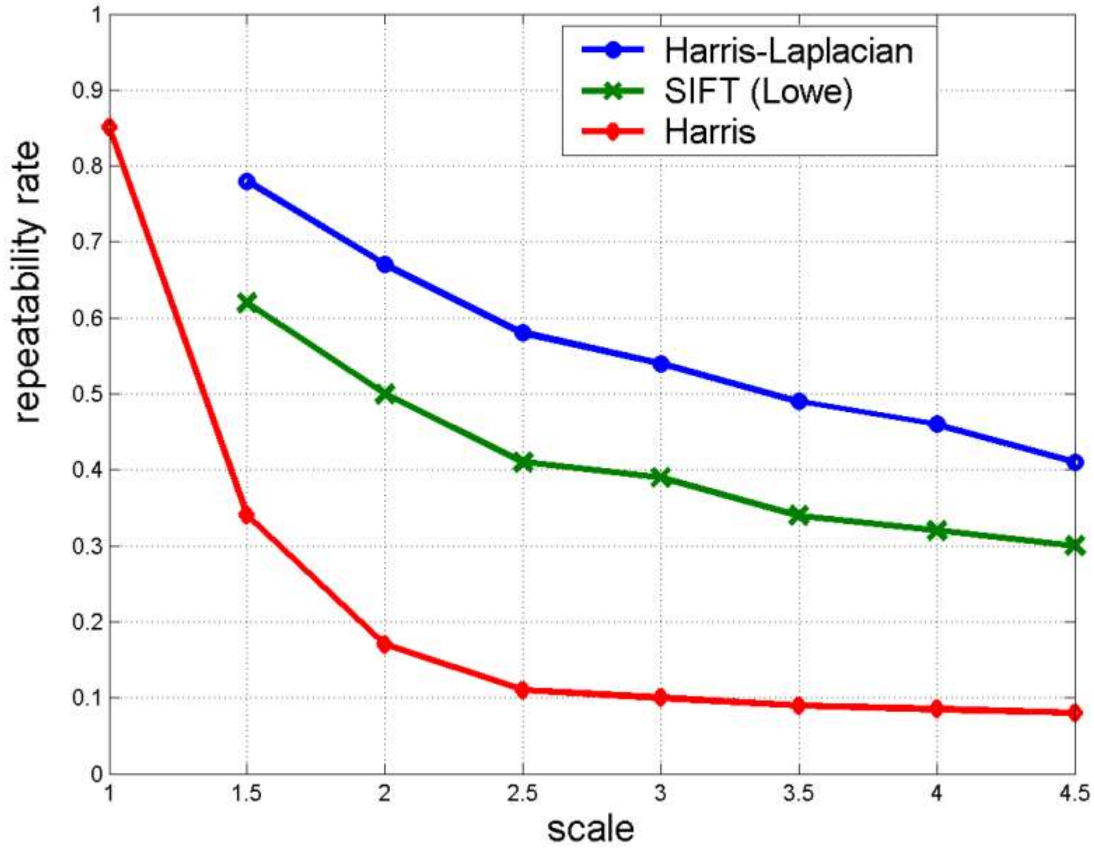
Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$$





K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

Neighborhoods and SIFT – Key Points

- Algorithms to find neighborhoods
  - Represented by location, scale and orientation
  - Neighborhood is covariant
    - If image is translated, scaled, rotated
    - Neighborhood is translated, scaled, rotated
    - Important property for matching
  - Affine covariant constructions are available
- Once found, describe with SIFT features
  - A representation of local orientation histograms, comparable to HOG
  - Normalized differently

SIFT invariances

- Spatial binning gives tolerance to small shifts in location and scale
- Explicit orientation normalization
- Photometric normalization by making all vectors unit norm
- Orientation histogram gives robustness to small local deformations

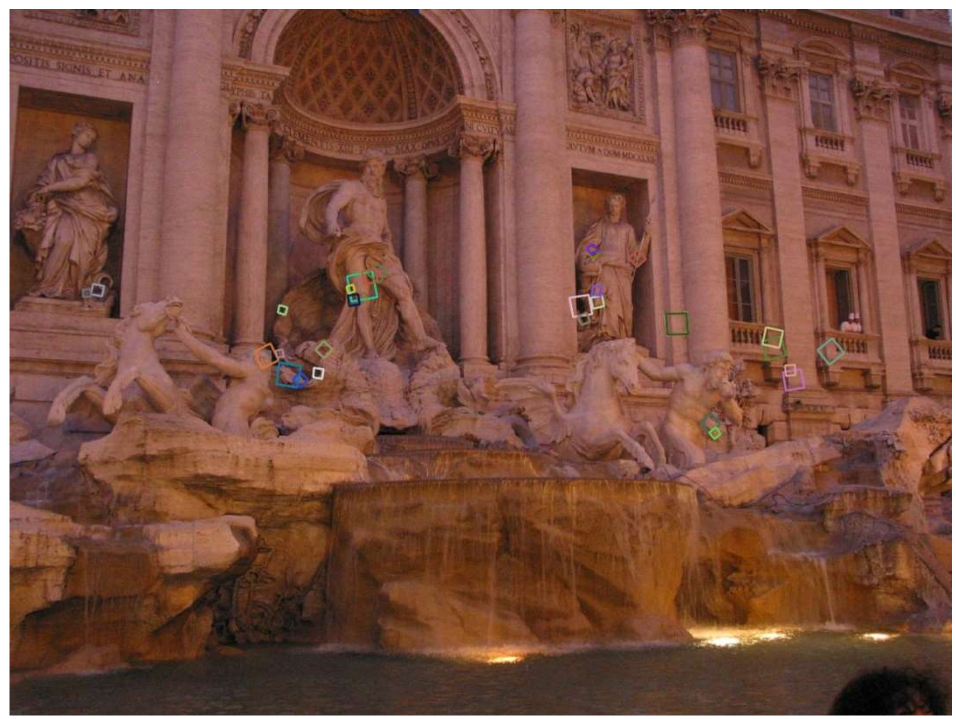
PCA-SIFT

- Different descriptor (same keypoints)
- Apply PCA to the gradient patch
- Descriptor size is 20 (instead of 128)
- More robust, faster

Summary of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT



Slide Credits

- David A. Forsyth - UIUC
- Svetlana Lazebnik – UIUC
- Rob Fergus – NYU

Next class

- Texture
- Readings for next lecture:
  – Forsyth and Ponce 6.1 – 6.4,
  – (optional) Szelinski 10.5
- Readings for today:
  – Forsyth and Ponce 5;
  – (optional) Szeliski 3.1-3.3

Questions