CSE 473/573 Computer Vision and Image Processing (CVIP)

Ifeoma Nwogu

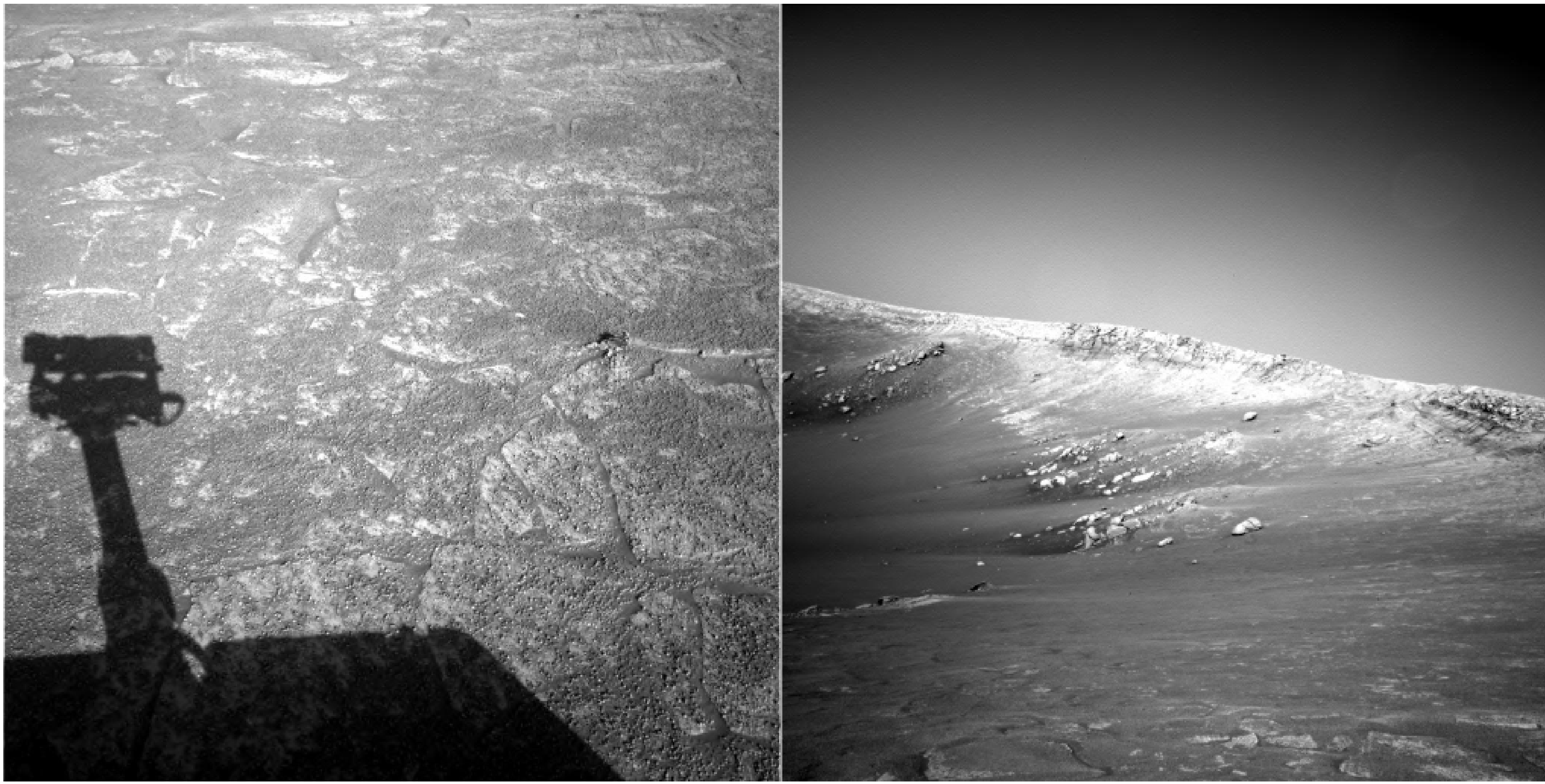
inwogu@buffalo.edu

Lecture 11 – 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

- Corners (Harris Detector)
- Blobs
- Descriptors

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

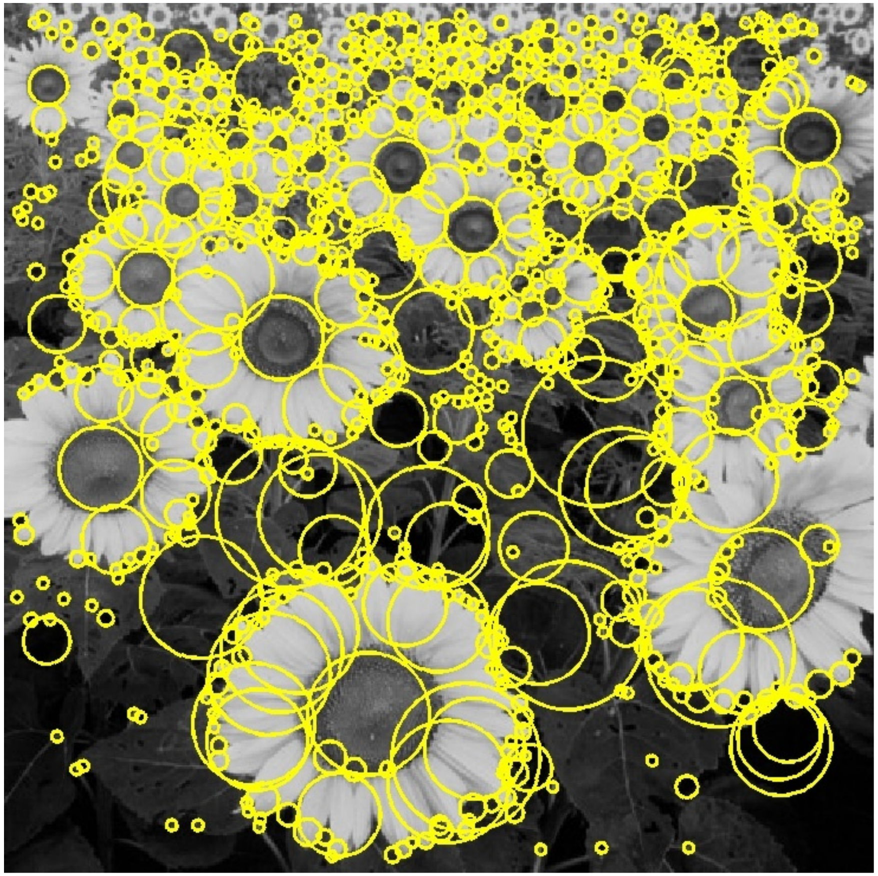
$$\mathcal{H} = \sum_{window} \{(\nabla I)(\nabla I)^T\}$$

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

Overview

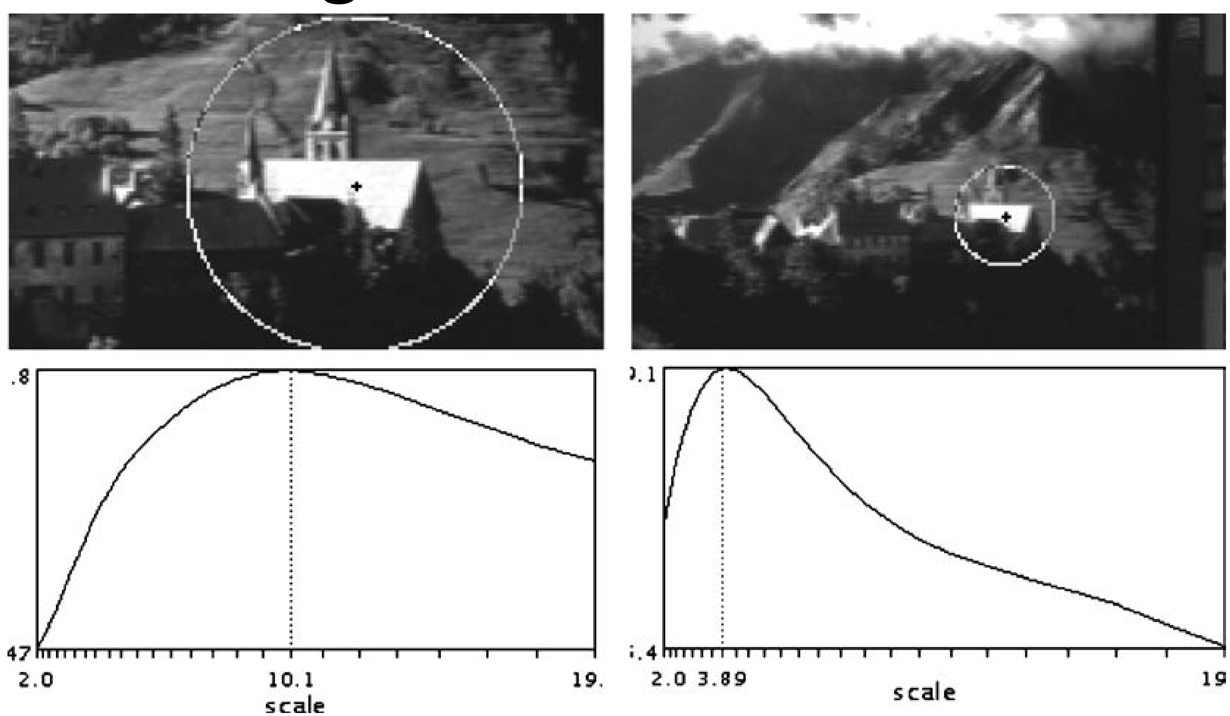
- Corners (Harris Detector)
- Blobs
- Descriptors

Blob detection with scale selection

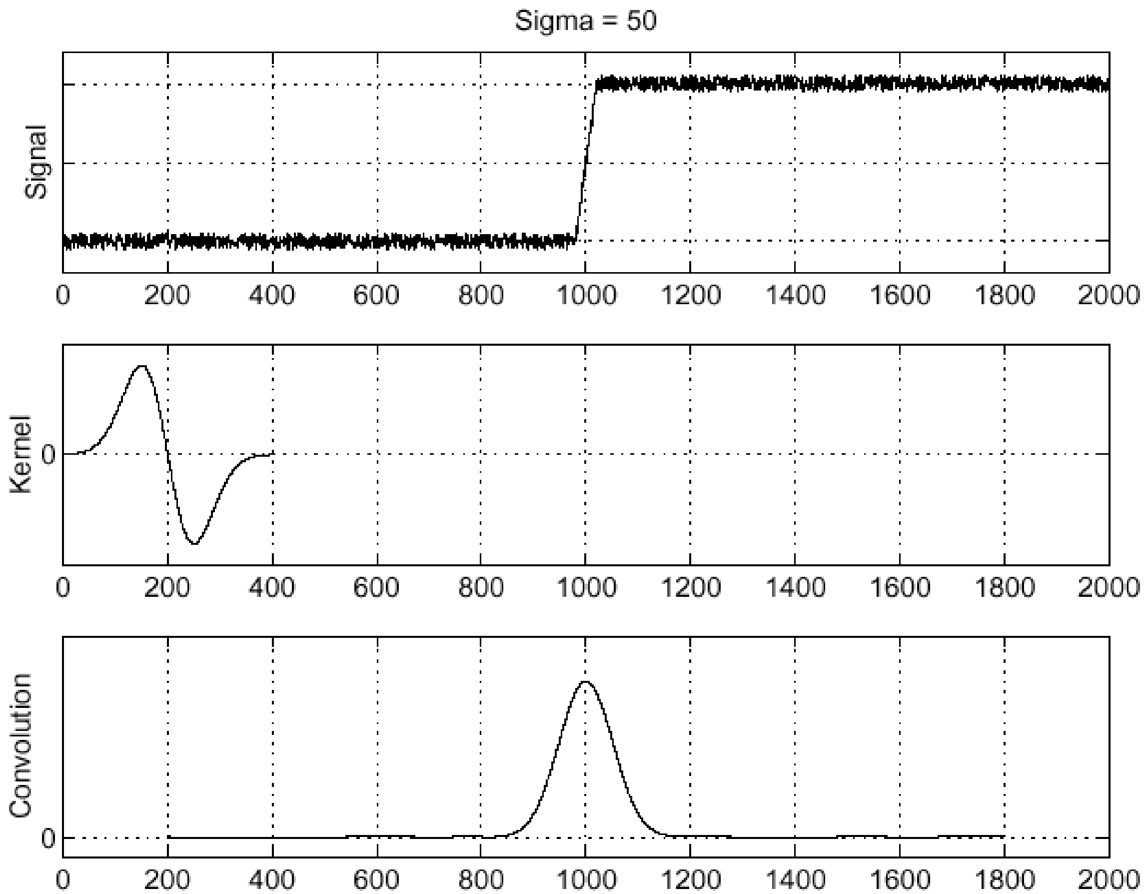


Achieving scale covariance

- 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



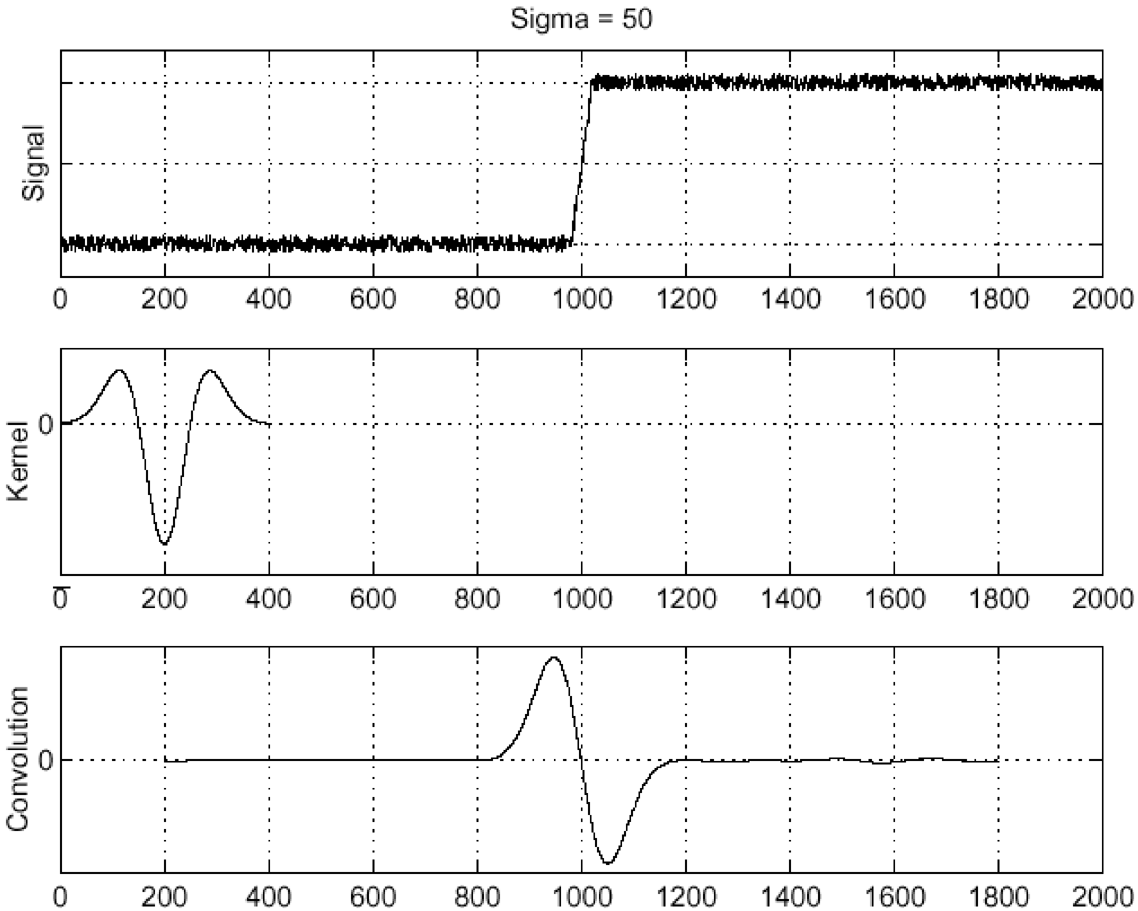
f — Edge

$\frac{d}{dx} g$ — Derivative of Gaussian

$f * \frac{d}{dx} g$ — Edge = maximum of derivative

Source: S. Seitz

Edge detection, Take 2



f — Edge

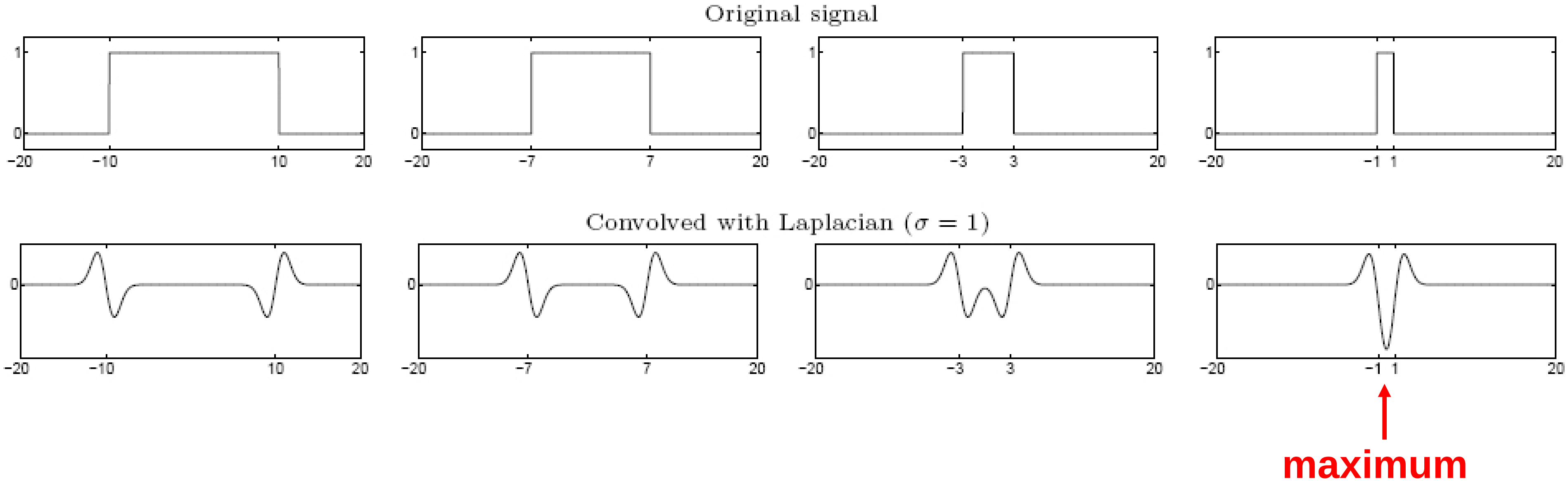
$\frac{d^2}{dx^2} g$ — Second derivative of Gaussian (Laplacian)

$f * \frac{d^2}{dx^2} g$ — Edge = zero crossing of second derivative

Source: S. Seitz

From edges to blobs

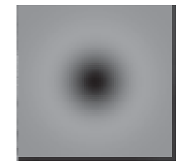
- Edge = ripple
- Blob = superposition of two ripples



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

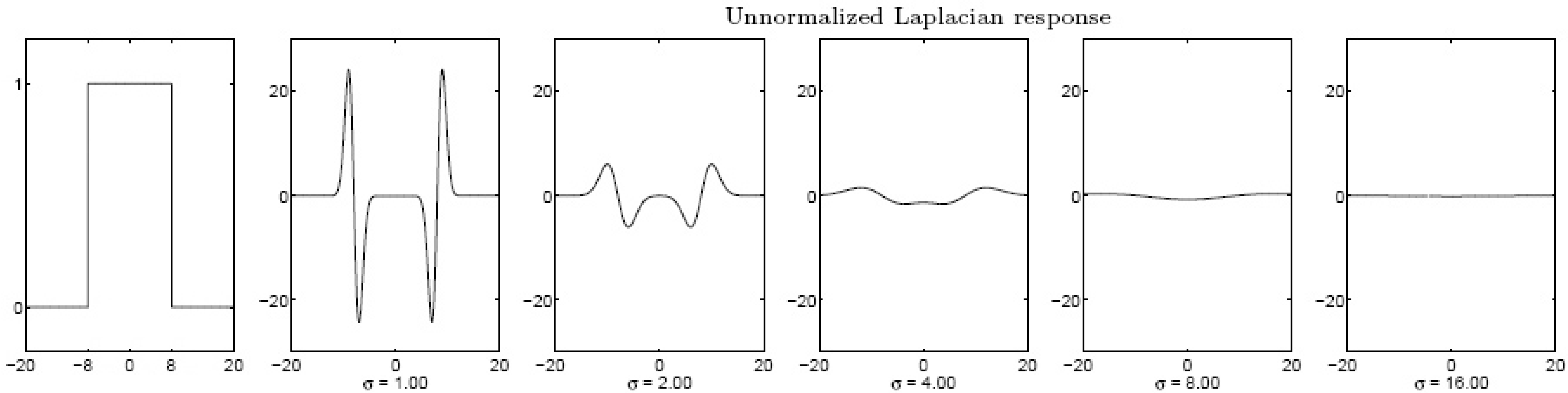
- Assume we have detected a corner
- How big is the neighborhood?
- Use Laplacian of Gaussian filter
  - Details on next slide
  - Kernel looks like fuzzy dark blob on pale light foreground
  - Scale (sigma) of Gaussian gives size of dark, light blob



- 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:



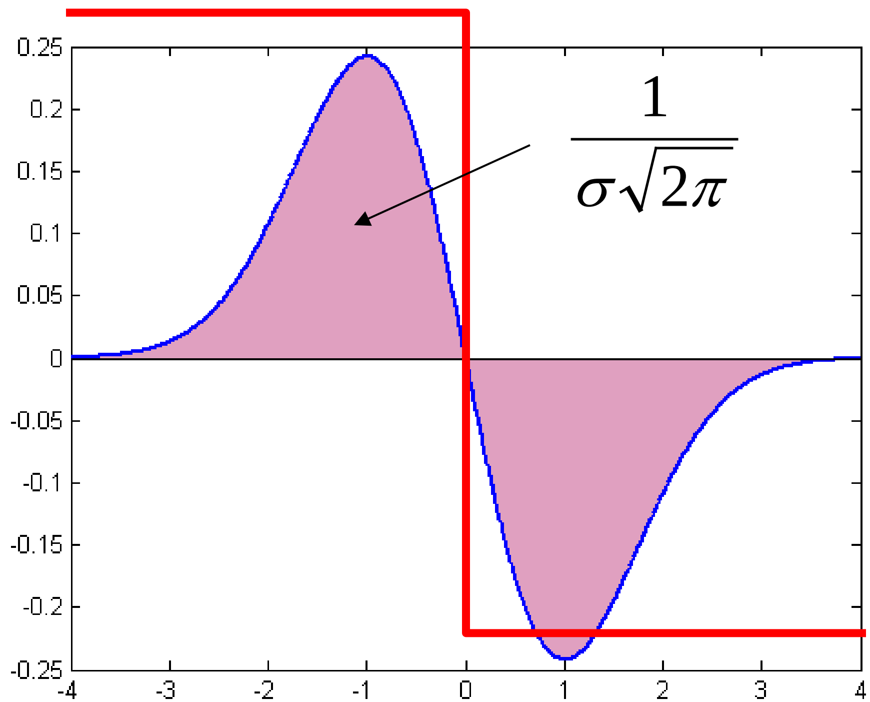
original signal (radius=8)

increasing σ →

Why does this happen?

Scale normalization

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

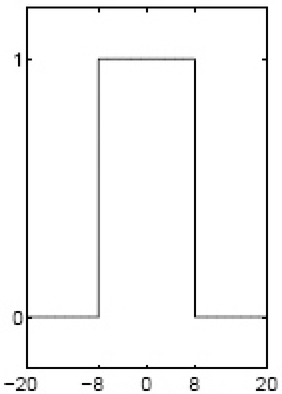


Scale normalization

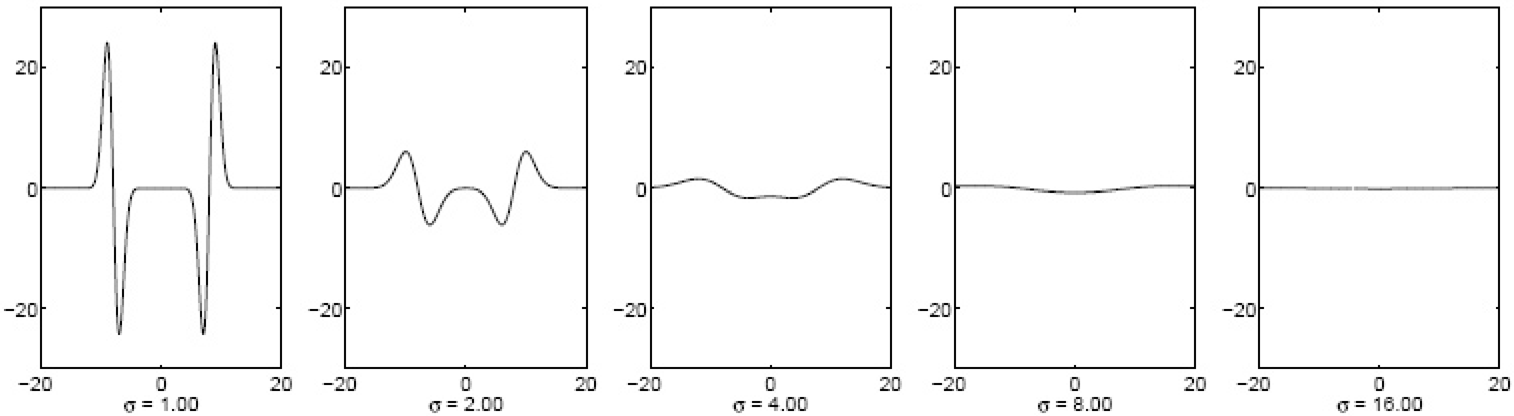
- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases
- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

Effect of scale normalization

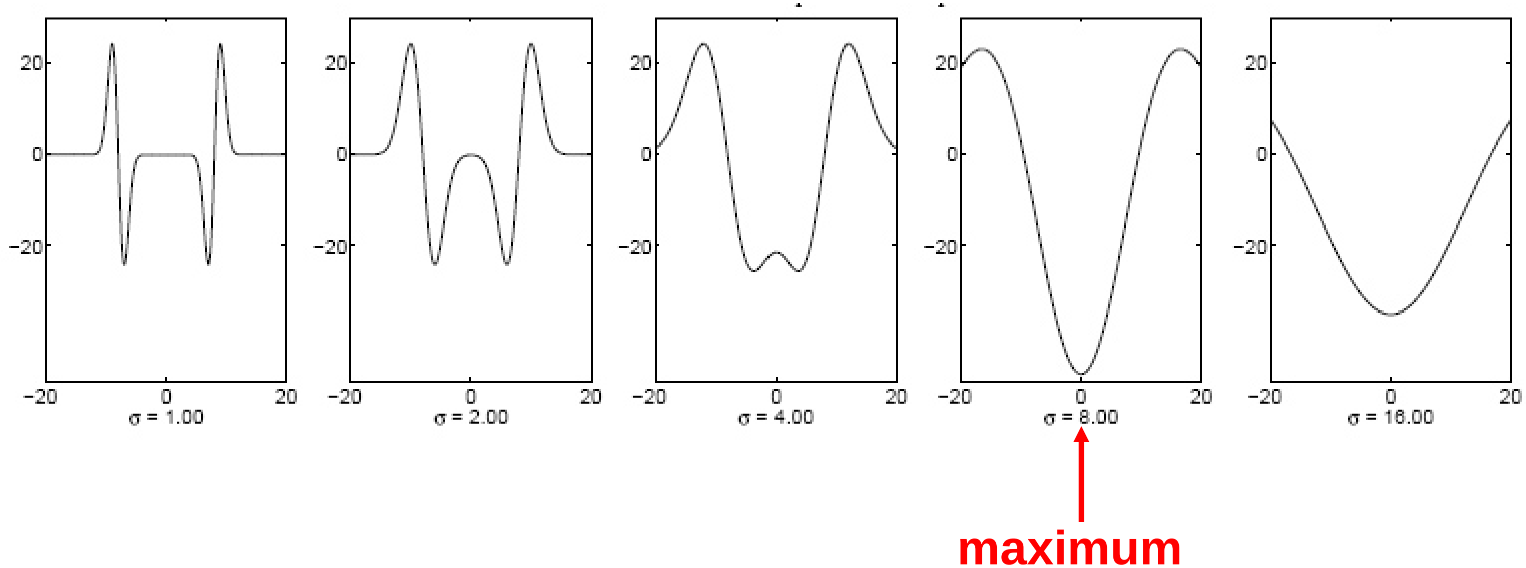
Original signal



Unnormalized Laplacian response

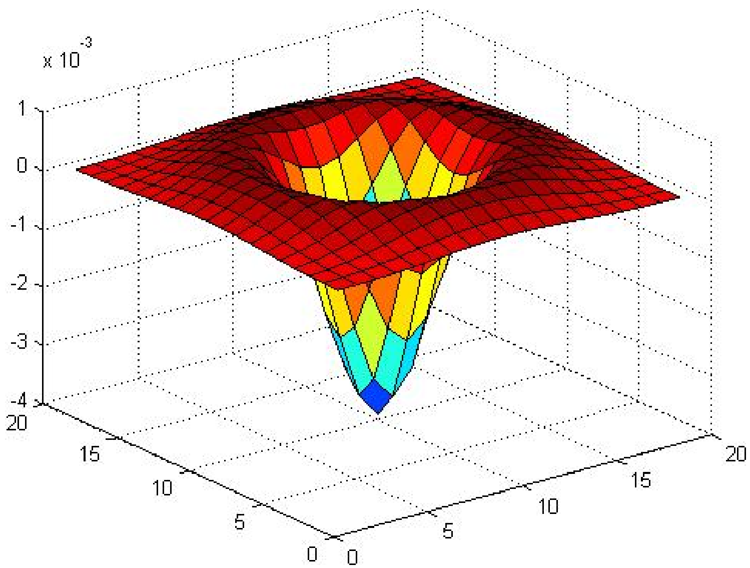


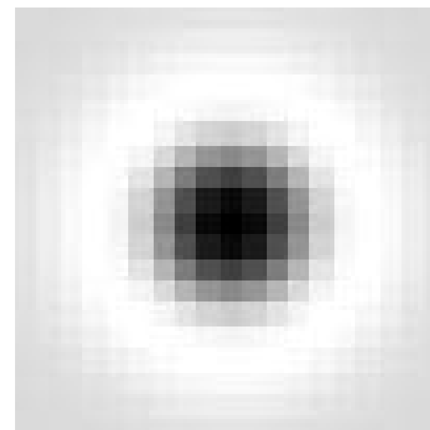
Scale-normalized Laplacian response



Blob detection in 2D

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

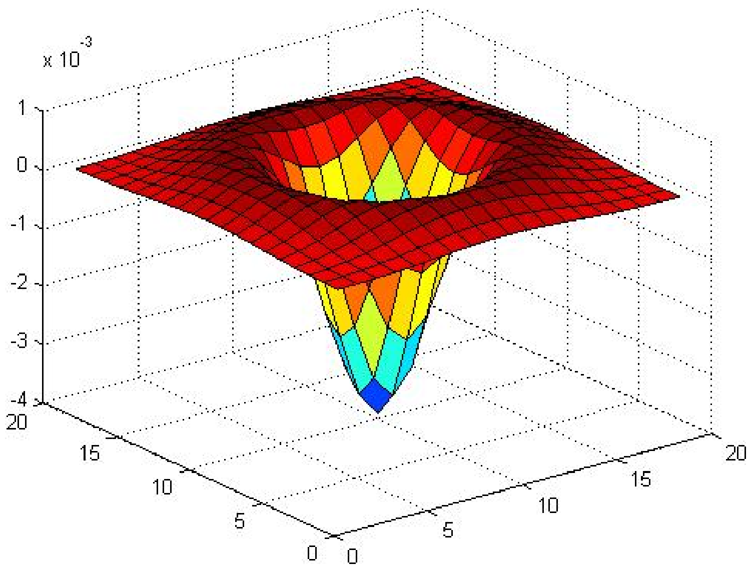


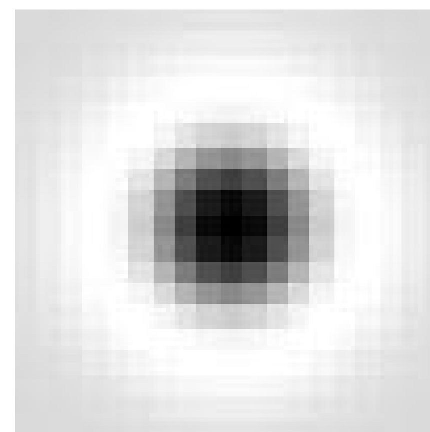


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} = \frac{1}{\pi \sigma^4}\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Blob detection in 2D

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

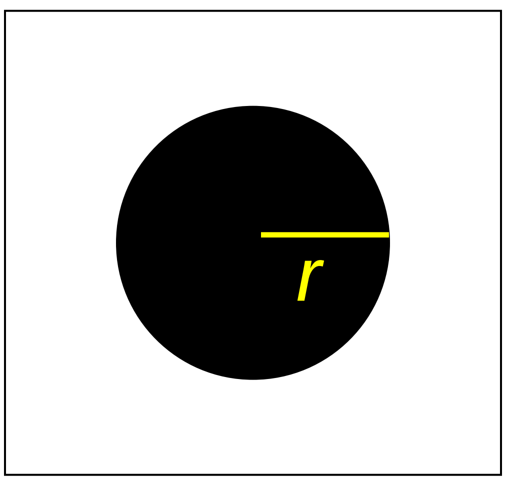




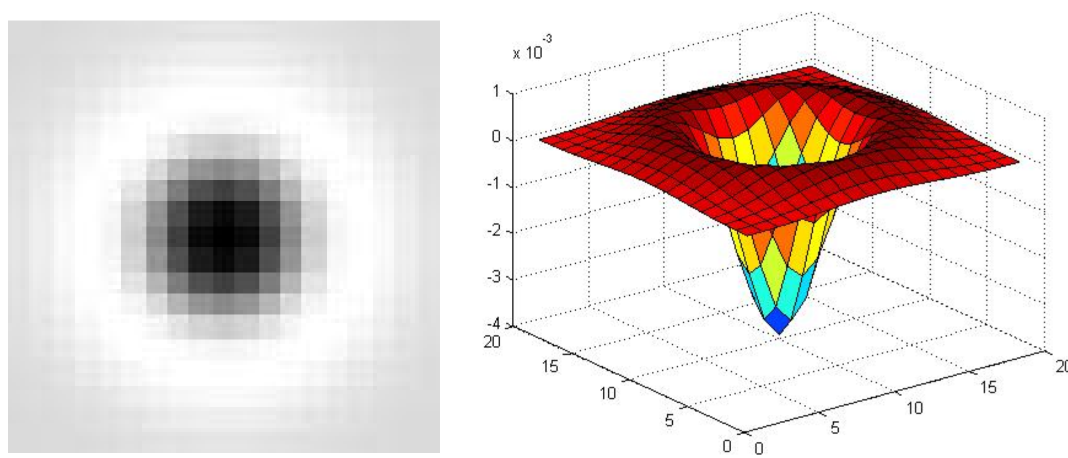
Scale-normalized: $$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

Scale selection

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



image



Laplacian

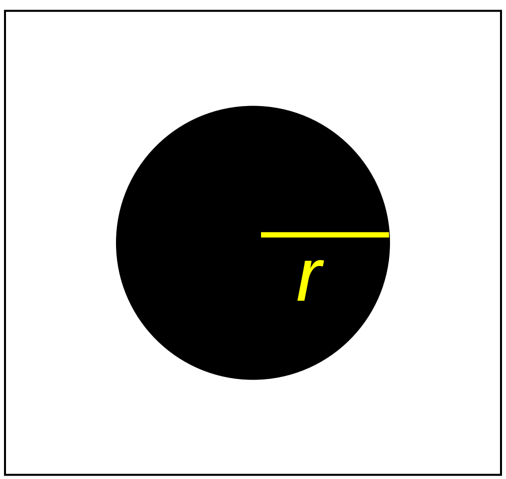
Scale selection

- 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):

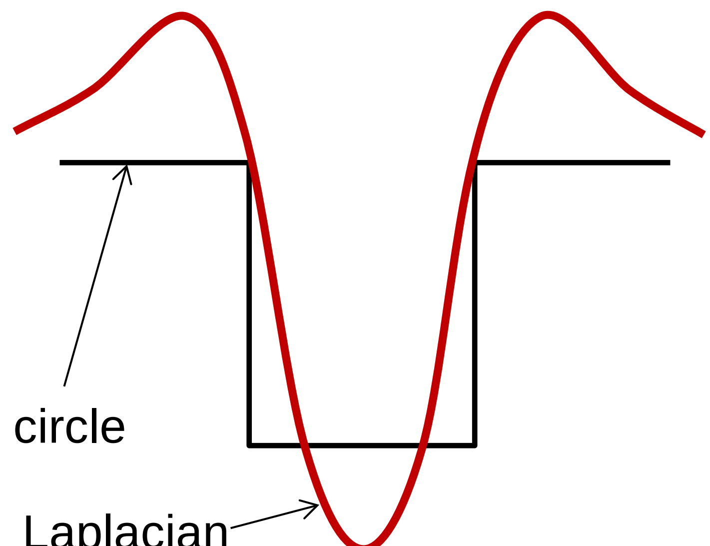
$$\left(1-\frac{x^2+y^2}{2\sigma^2}\right)=0$$

- Therefore, the maximum response occurs at

$$\sigma = r/\sqrt{2}.$$

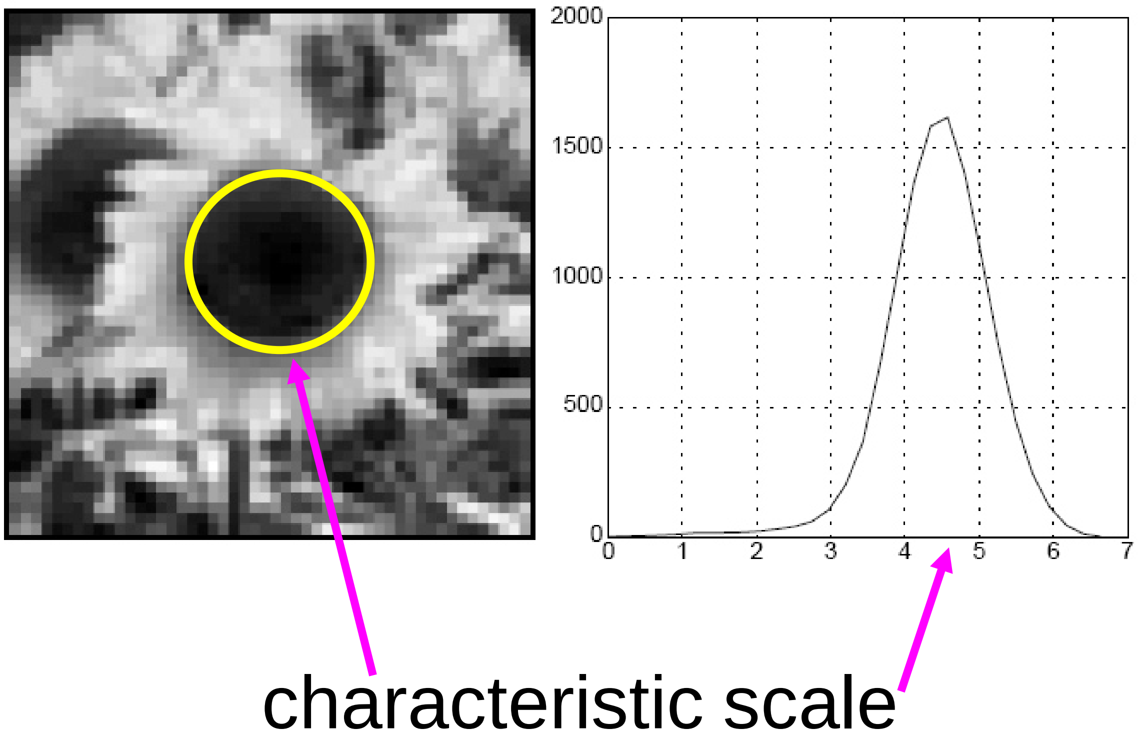


image



Characteristic scale

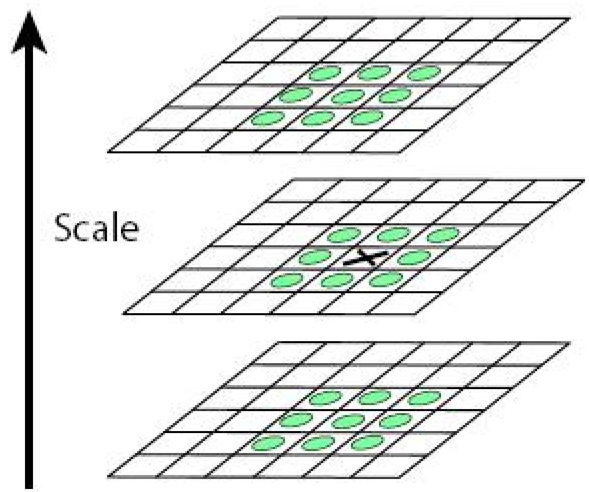
- We define the characteristic scale of a blob as the scale that produces peak of Laplacian response in the blob center



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

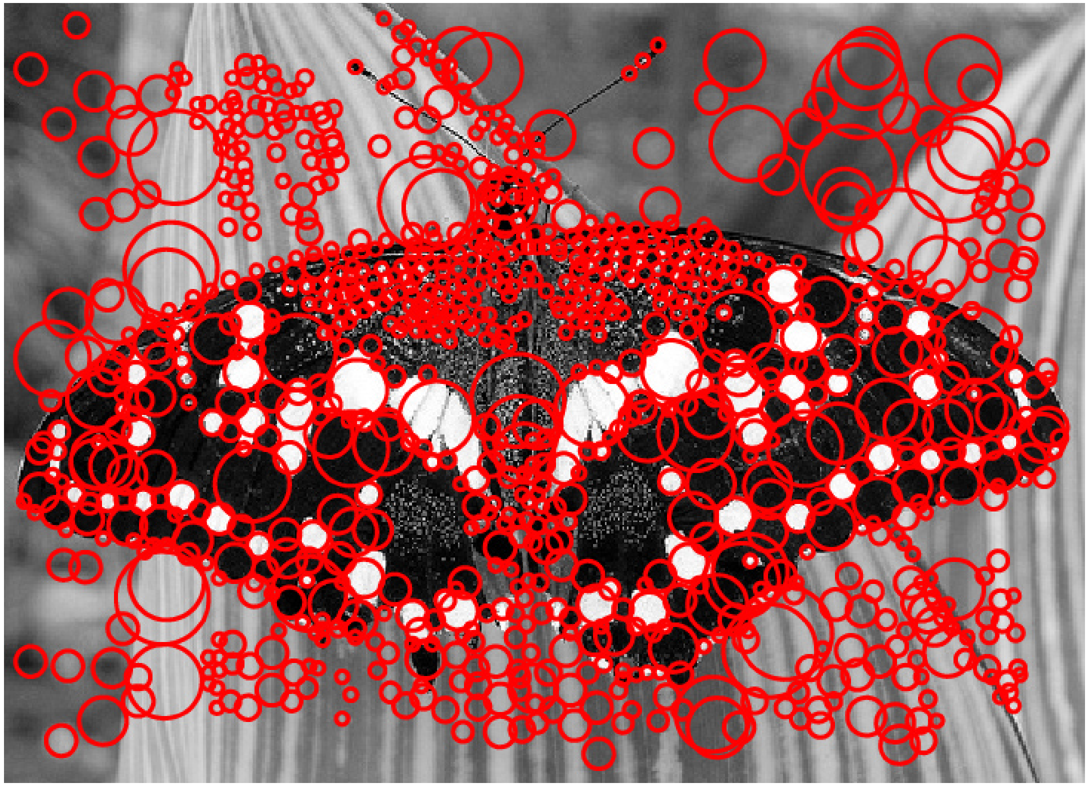


Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

Scale-space blob detector: Example



Efficient implementation

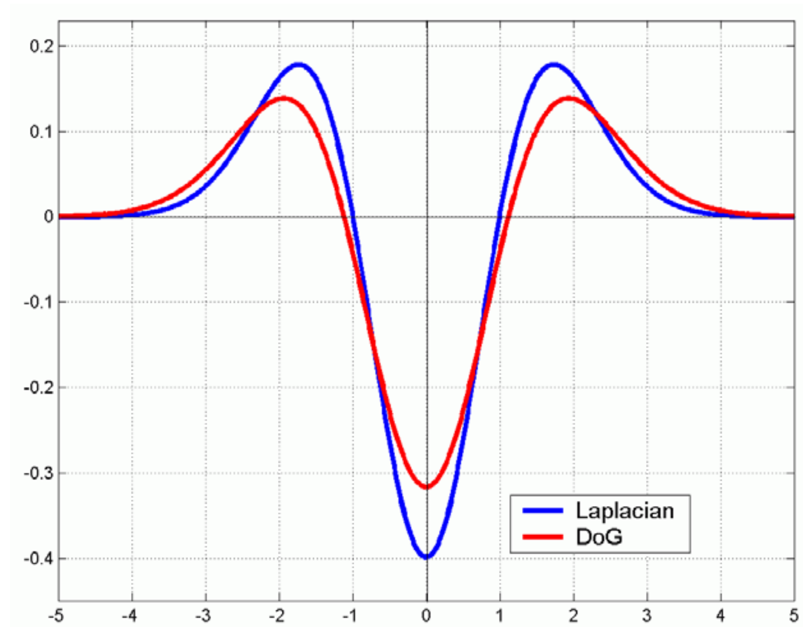
- Approximating the Laplacian with a difference of Gaussians:

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

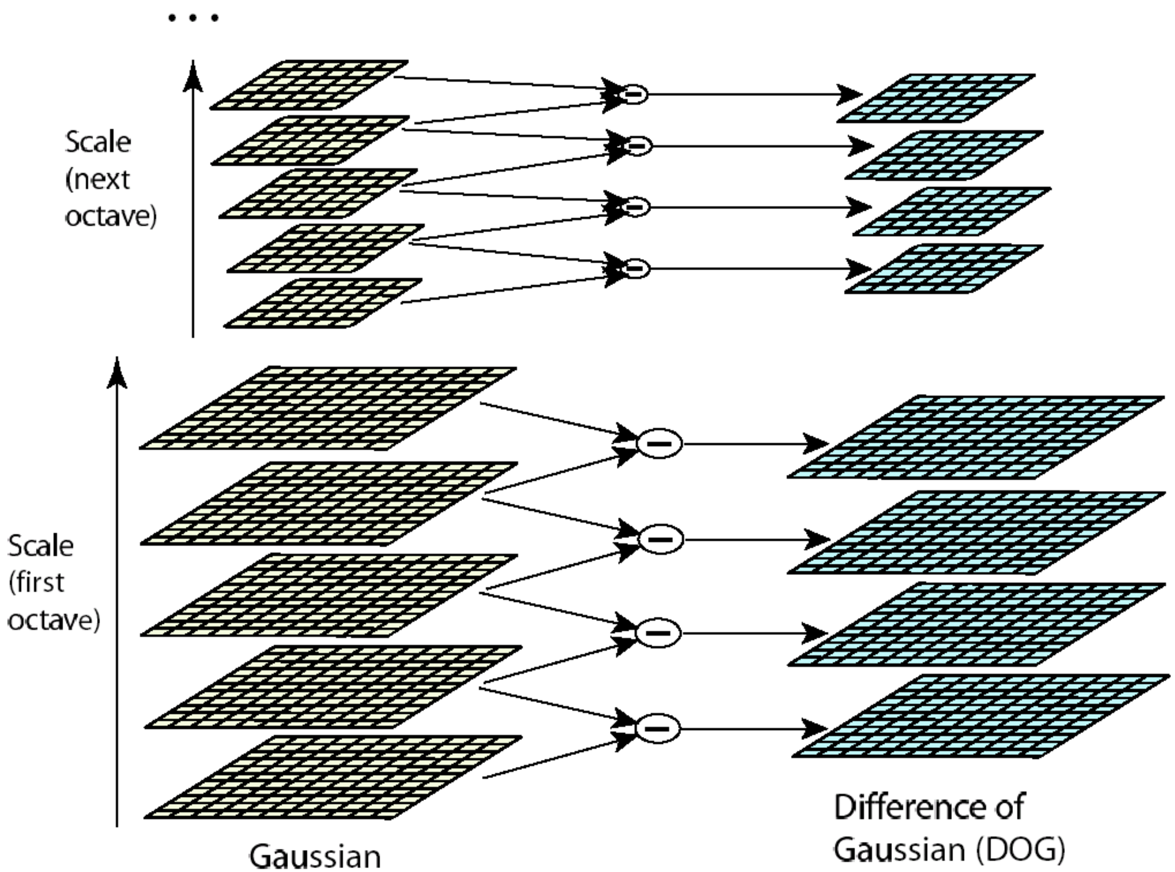
(Laplacian)

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

(Difference of Gaussians)



Efficient implementation

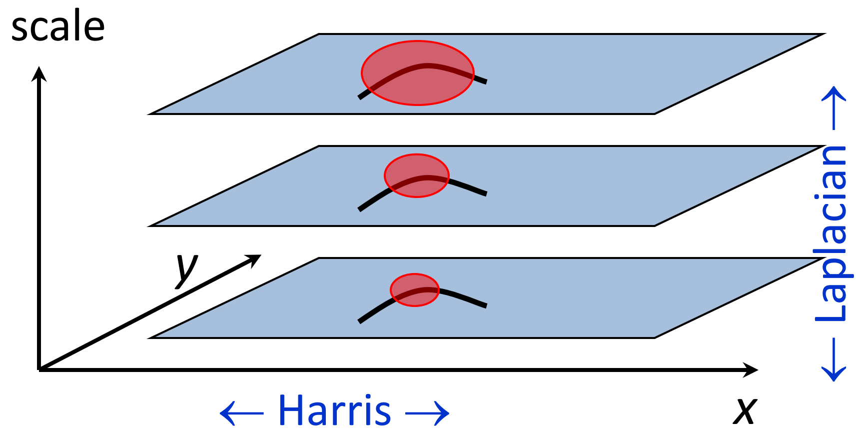


David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), pp. 91-110, 2004.

Scale Invariant Detectors

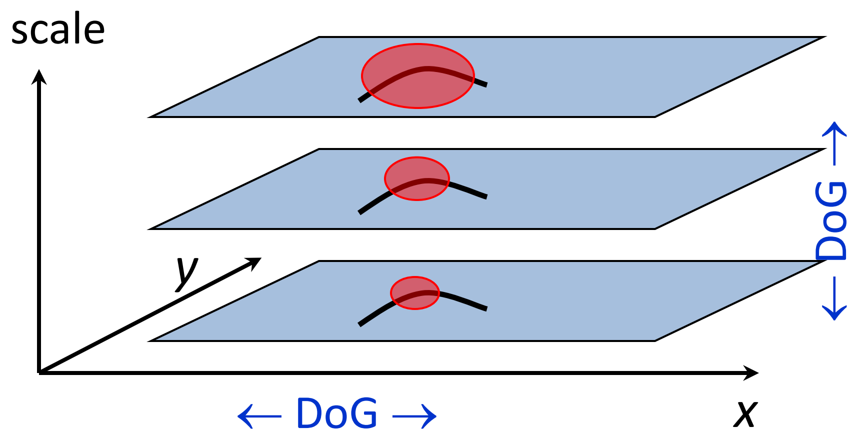
- Harris-Laplacian[1]

  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



- Difference of Gaussians
- a.k.a. SIFT (Lowe)[2]

  Find local maximum of:
  - Difference of Gaussians in space and scale

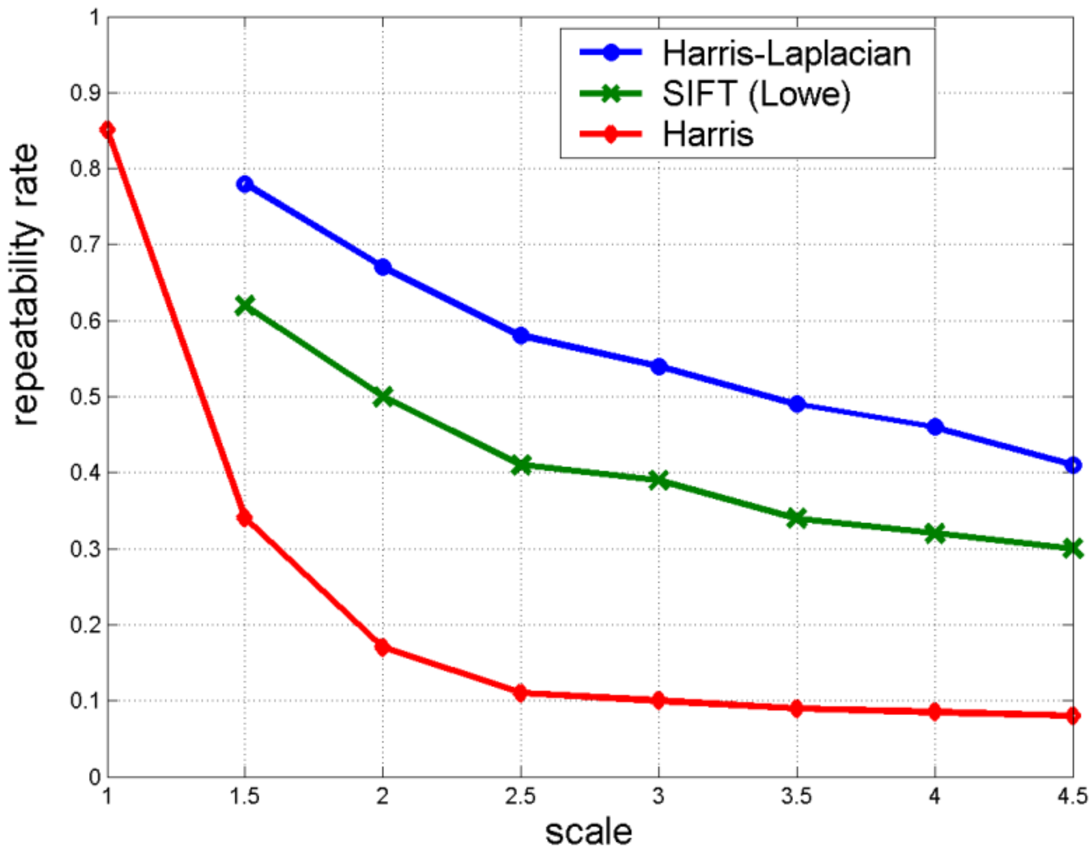


[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004

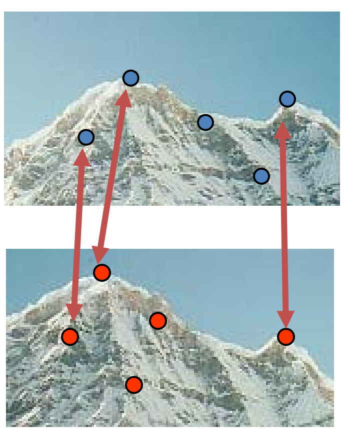
Scale Invariant Detectors

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



Repeatability rate:

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



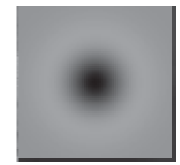
K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

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

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

Overview

- Corners (Harris Detector)
- Blobs
- Descriptors

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

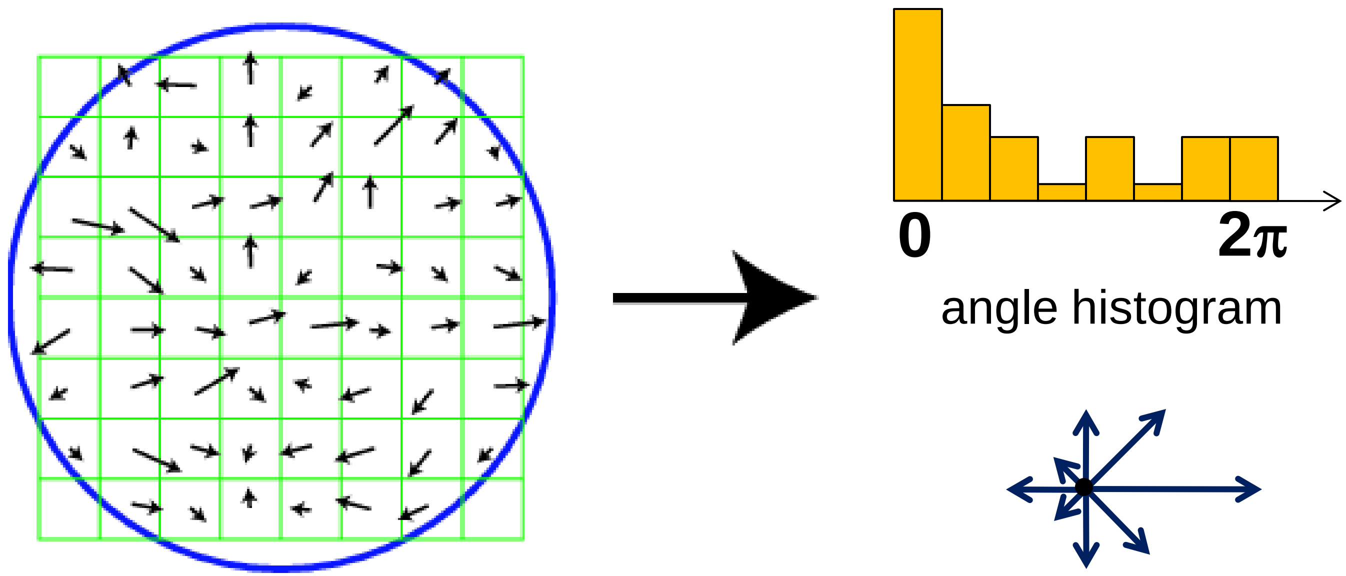


Image gradients

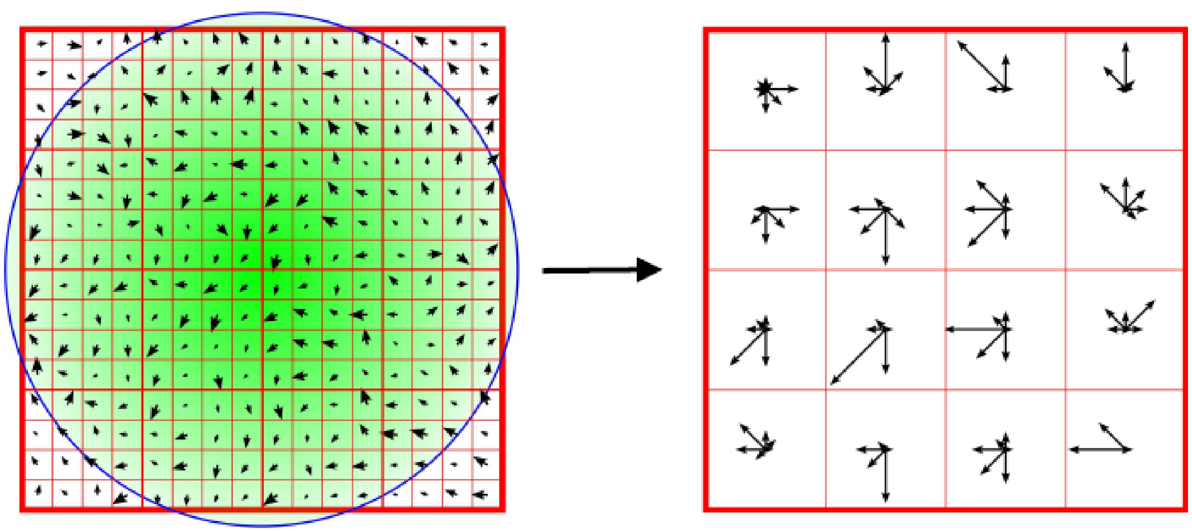
Adapted from slide by David Lowe

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

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

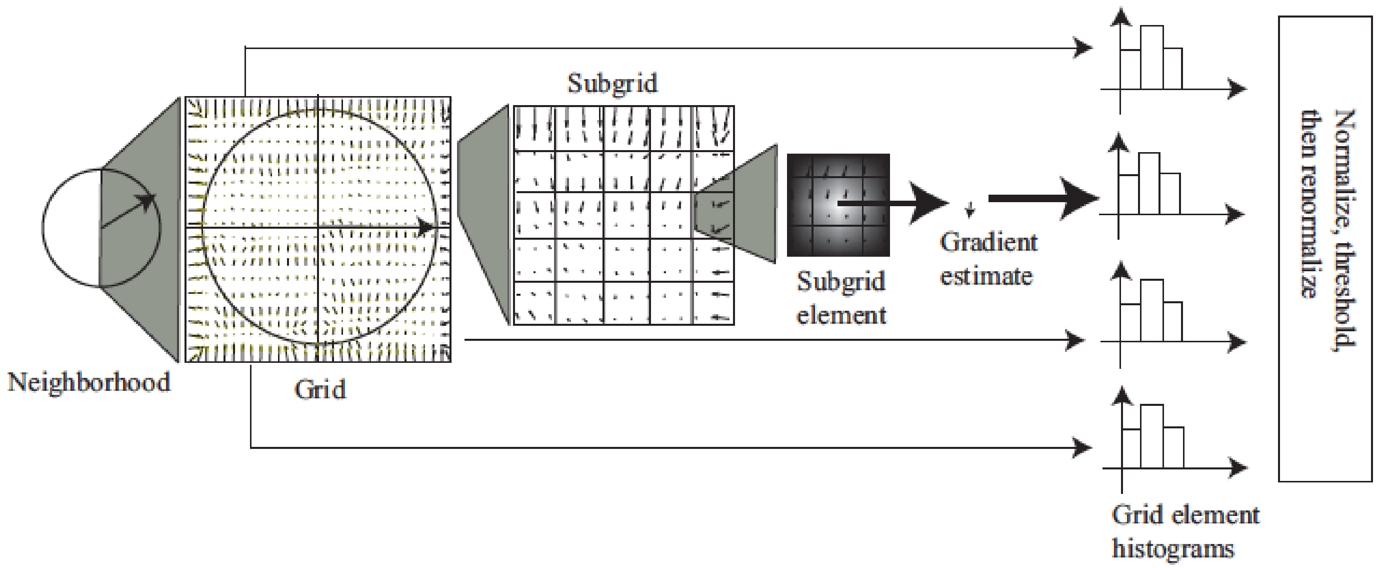


FIGURE 5.14: To construct a SIFT descriptor for a neighborhood, we place a grid over the rectified neighborhood. Each grid is divided into a subgrid, and a gradient estimate is computed at the center of each subgrid element. This gradient estimate is a weighted average of nearby gradients, with weights chosen so that gradients outside the subgrid cell contribute. The gradient estimates in each subgrid element are accumulated into an orientation histogram. Each gradient votes for its orientation, with a vote weighted by its magnitude and by its distance to the center of the neighborhood. The resulting orientation histograms are stacked to give a single feature vector. This is normalized to have unit norm; then terms in the normalized feature vector are thresholded, and the vector is normalized again.

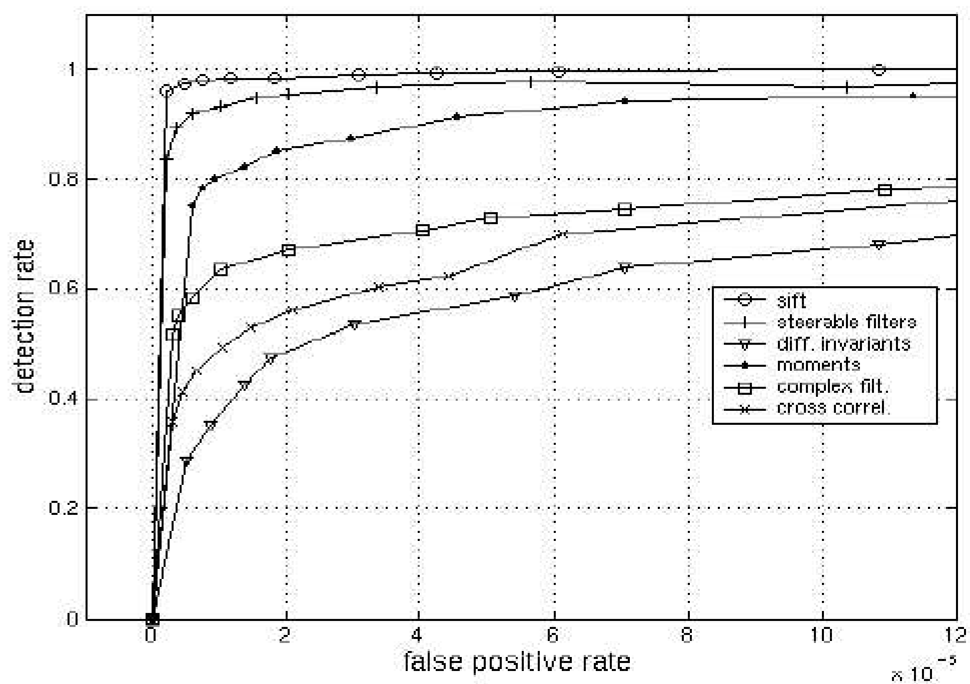
Neighborhoods and SIFT - Key Points

- 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 – Scale Invariant Feature Transform[1]

- Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45^0



[1] D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. Accepted to IJCV 2004
[2] K.Mikolajczyk, C.Schmid. “A Performance Evaluation of Local Descriptors”. CVPR 2003

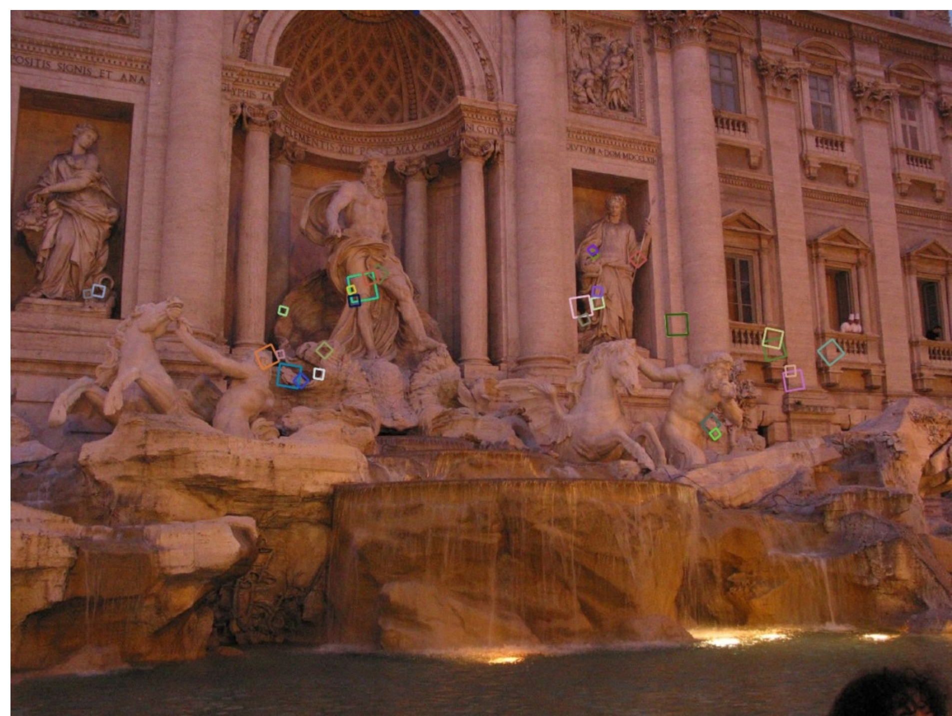
SIFT invariances

- 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

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



Summary

- We started last class with linear filters
  - including filter construction and separability, convolution methods and image blurring
- This week we discussed filter derivatives and scale space/pyramids
  - including 1st and 2nd derivatives of the Gaussian filters, the Gaussian pyramid and the Laplacian pyramid
- In this lecture we discussed how DoG filters detect edges and how post-processing works
  - specifically we focused on the Canny edge detector and its post-processing techniques

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, Szelinski 10.5 (optional)
- Readings for today:
  - Forsyth and Ponce 5; Szeliski 3.1-3.3

Questions