Carnegie Mellon

Prof. Marios Savvides

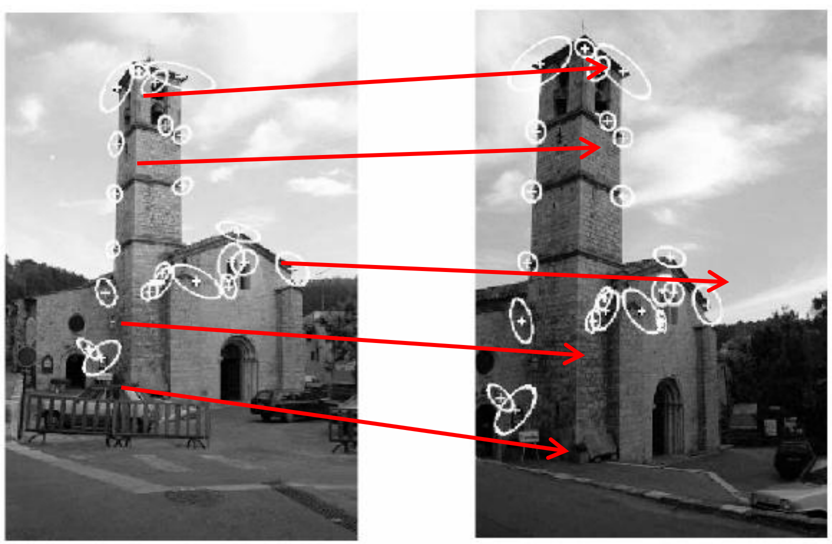
Pattern Recognition Theory

Lecture 10 : Feature Detection

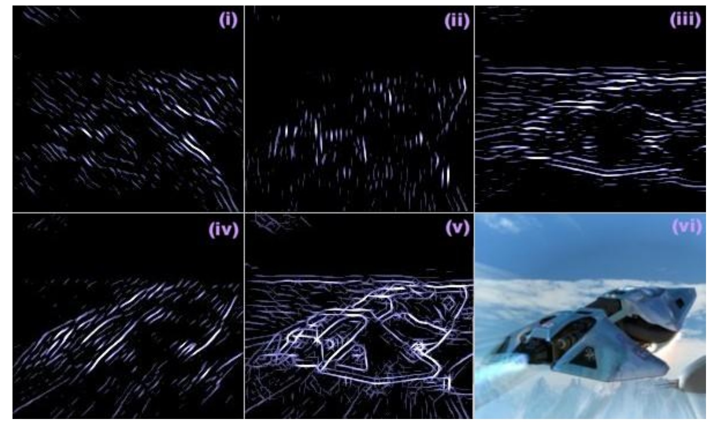
Electrical & Computer ENGINEERING

Edge & Interest Point Detection, Frequency Analysis

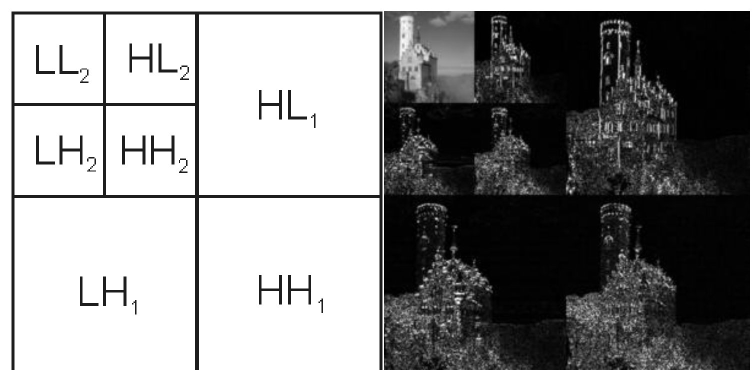
Edges



Interest points



(i) pi/4, (ii) 0, (iii) pi/2, (iv) 3*pi/4 , (v)all, (vi)input image.



Frequency analysis

Frequency Analysis

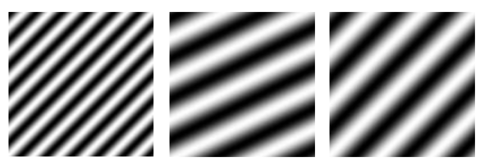
- From pixel intensities (spatial domain) to frequency and phase domain

Input image

•

=

A measure of image content at this frequency and orientation

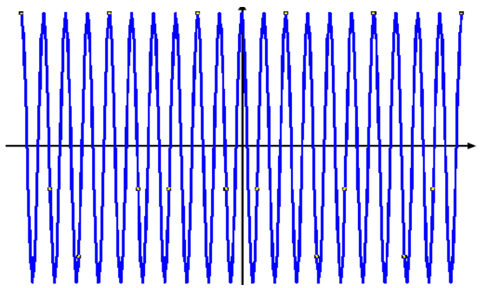


… etc

Fourier basis

Gabor Filter

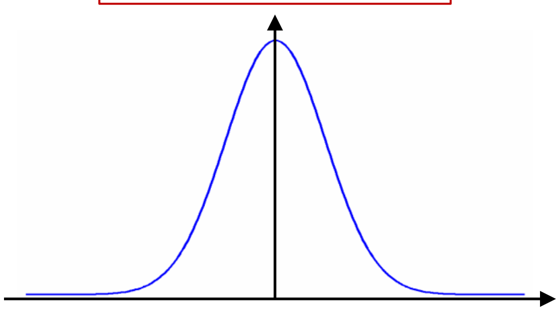
Frequency content
in local neighborhood
at every point in image
I' = I * Gabor



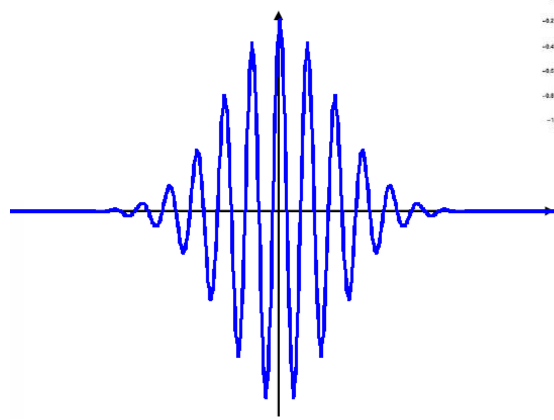
Fourier basis

×

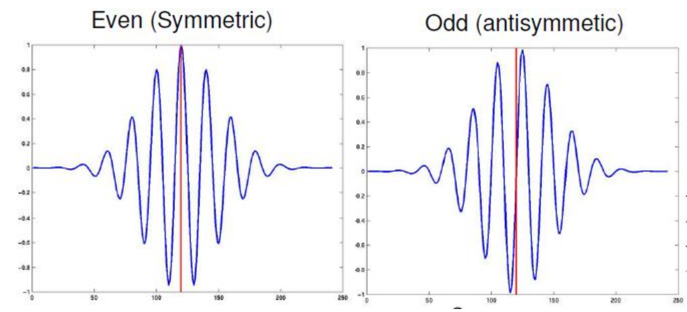
Gaussian filter



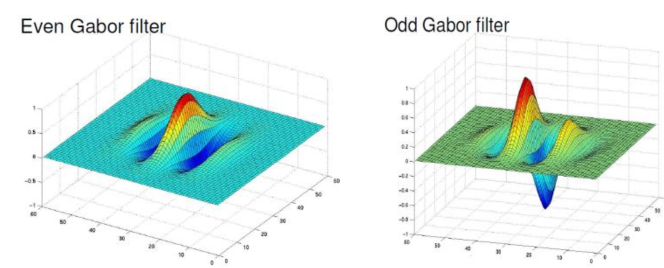
=



Gabor filter



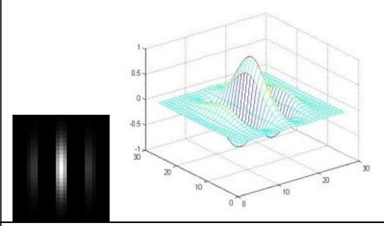
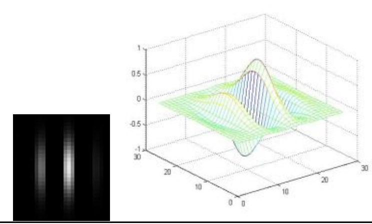
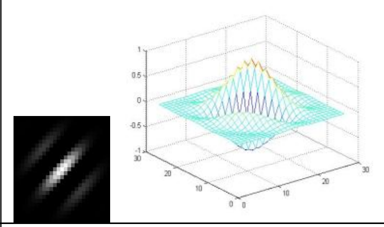
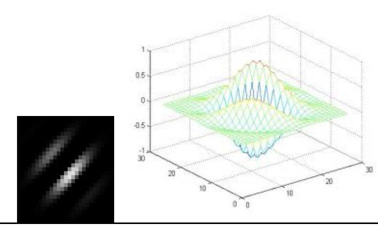
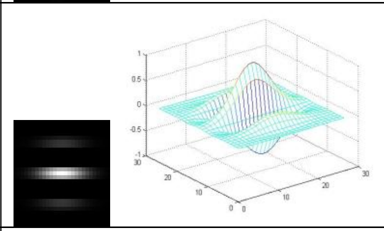
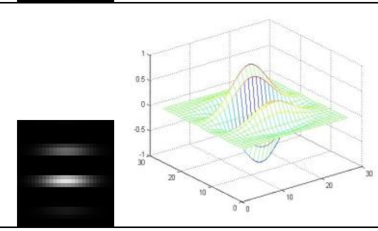
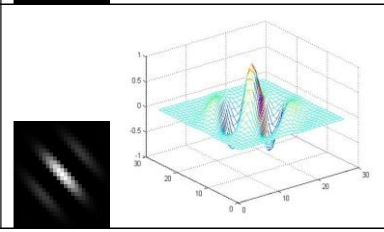
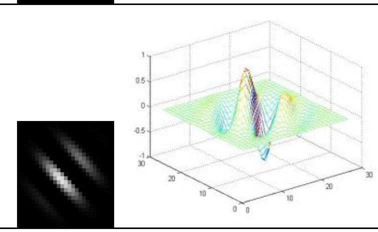
in 1D



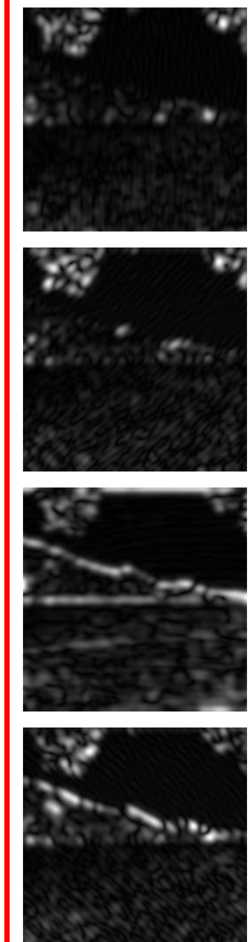
in 2D

Gabor Filter

Input image

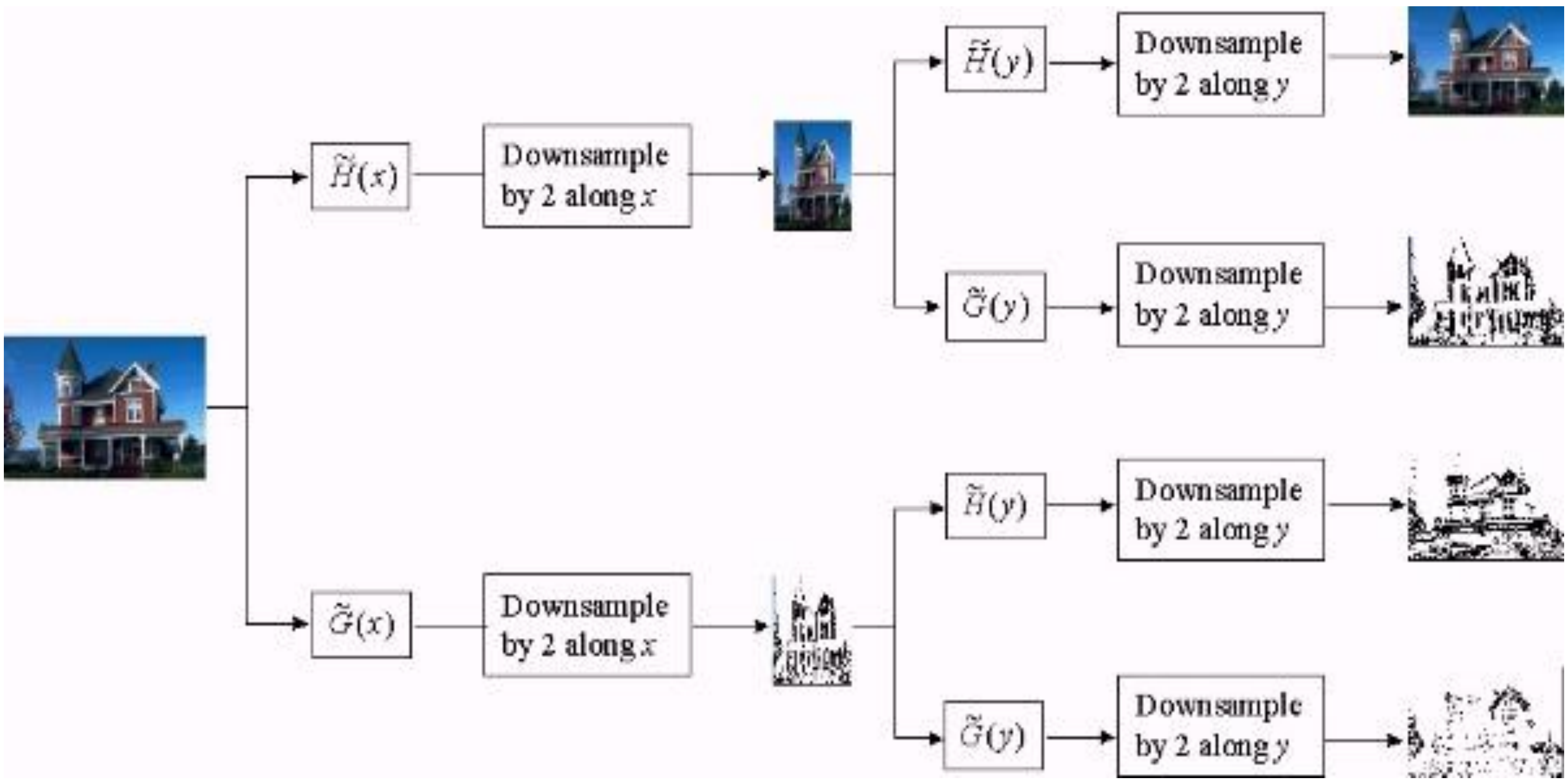
	Even Gabor Filter	Odd Gabor Filter
0 Deg.		
45 Deg.		
90 Deg.		
135 Deg.		

Gabor filter



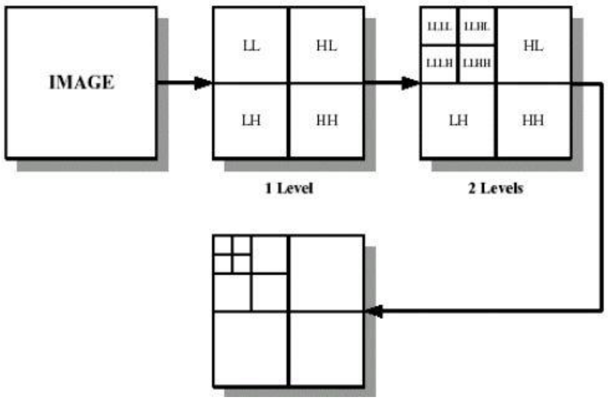
Filter response

Discrete Wavelet Transform

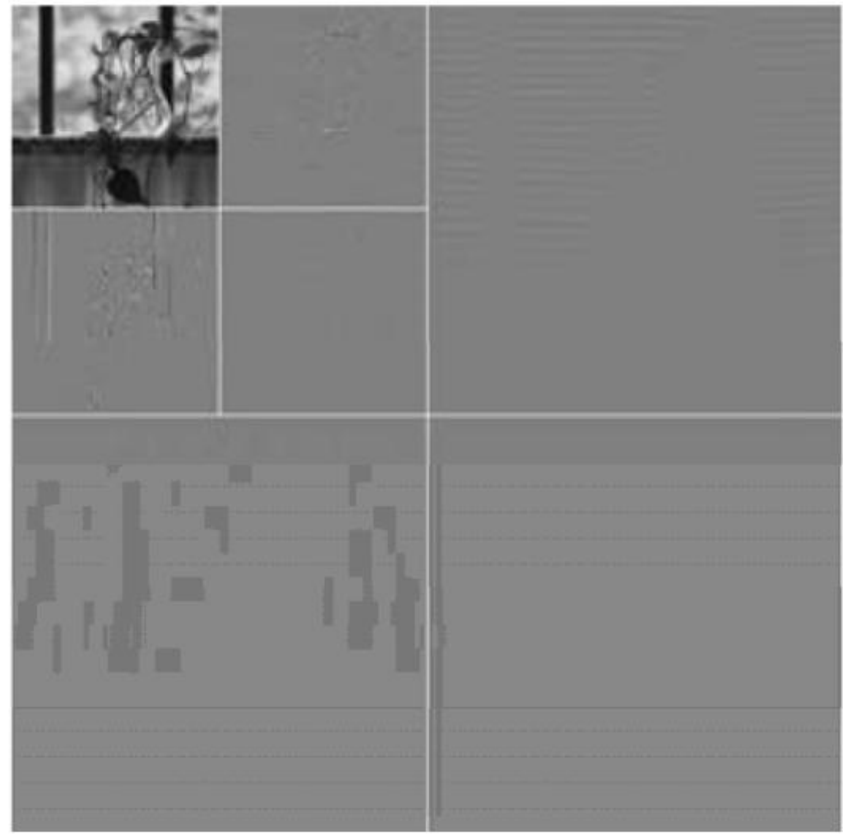


Scale and frequency are made consistent by scaling one basis filter.

Discrete Wavelet Transform



Input image

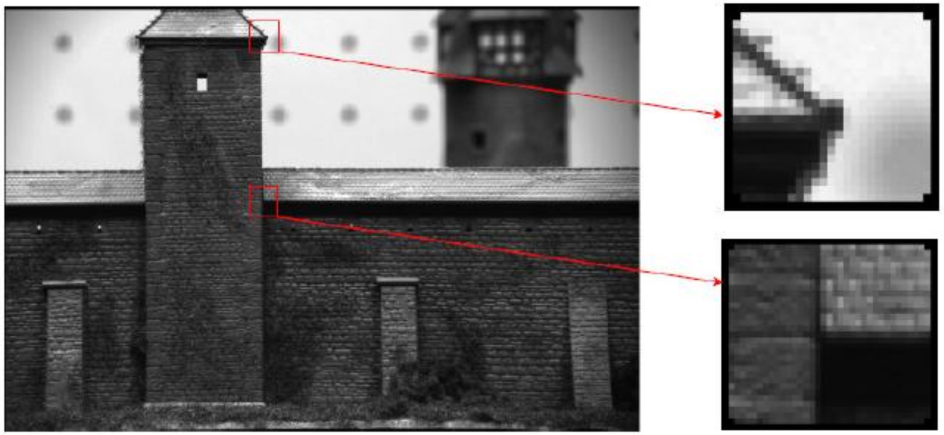


Filter response

Edge & Interest Point Detection

- What are edges?
  - Discontinuities of intensity in images

- What are interest points?
  - Junction of contours



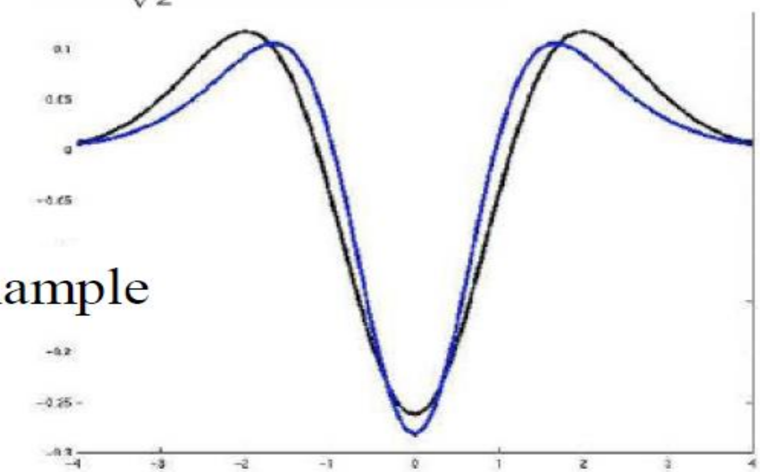
LoG and DoG Filters

- Laplacian of Gaussian (LoG) filter
  - LoG is useful for finding edges
  - LoG can be approximate by a Difference of two Gaussians (DoG) at different scales.

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

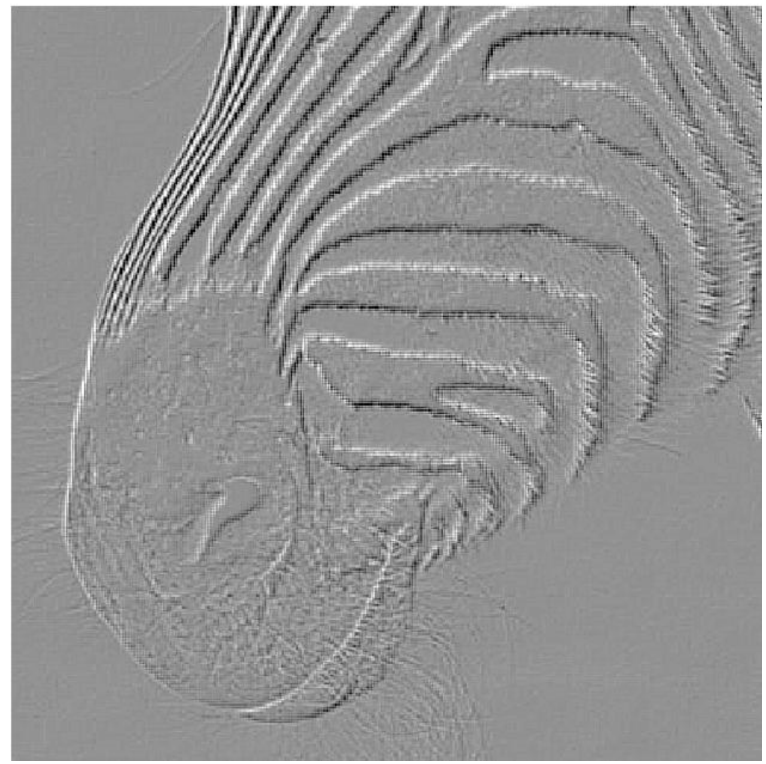
Best approximation when:
$\sigma_1 = \frac{\sigma}{\sqrt{2}},\ \sigma_2 = \sqrt{2}\sigma$

1D example

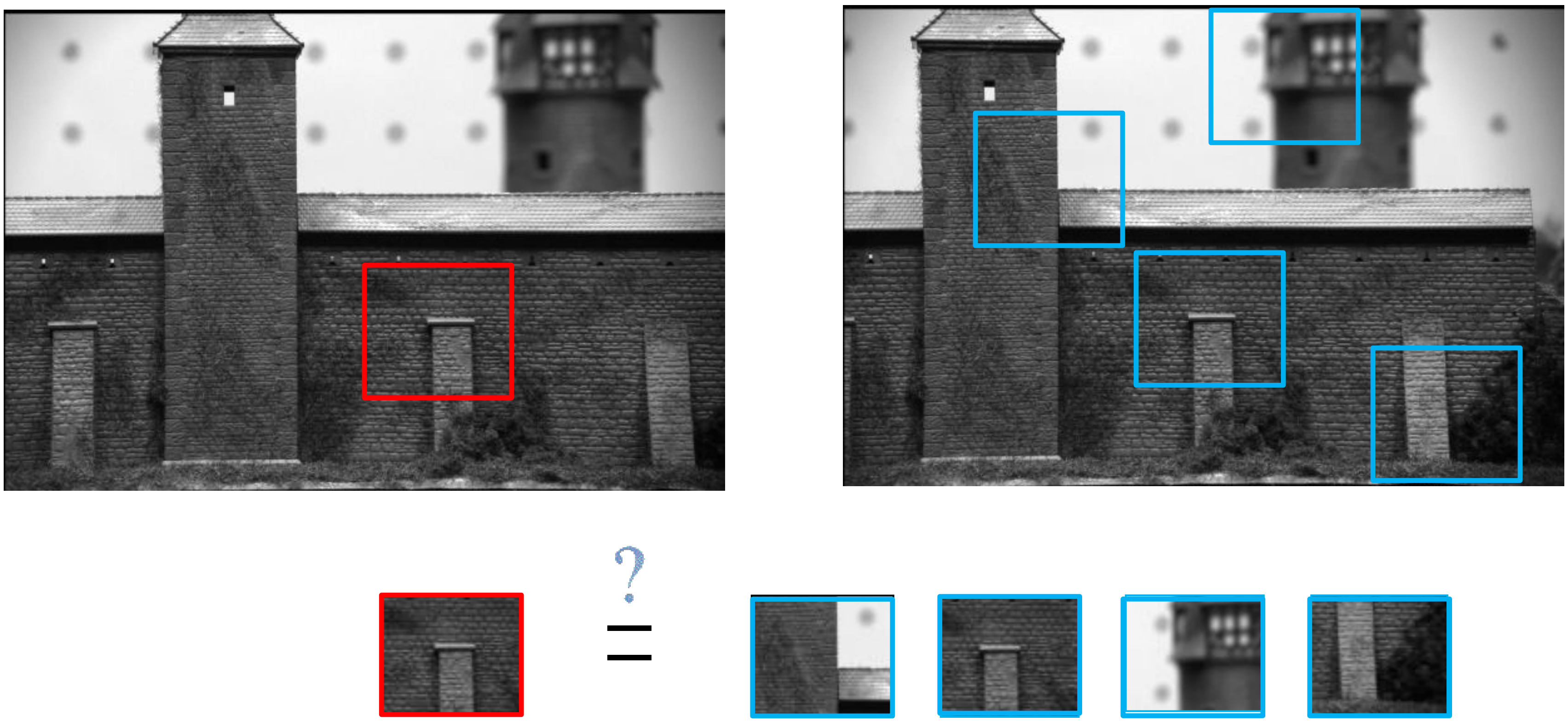


LoG and DoG Filters

Input image → $I(x,y)$ → Smooth → Laplacian → $O(x,y)$ → Filter response

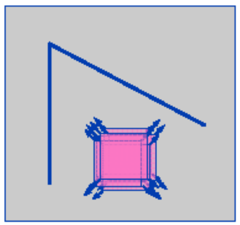


Correspondences between Images

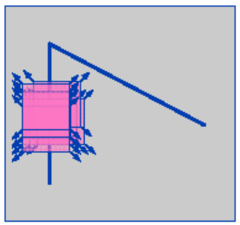


Interest points are good features to match!

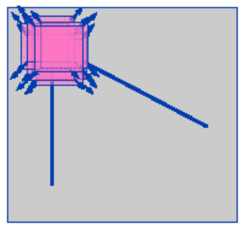
Interest points



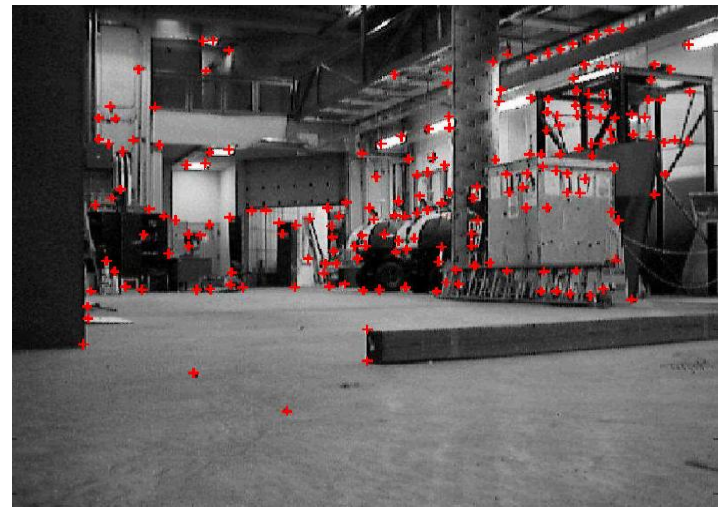
“flat” region:
no change in
all directions



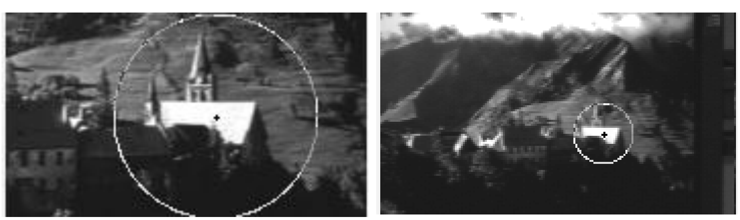
“edge”:
no change along
the edge direction

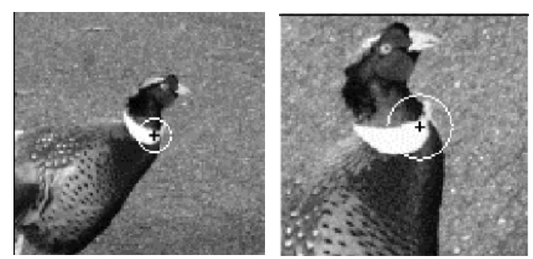


“corner”:
significant change
in all directions



But,

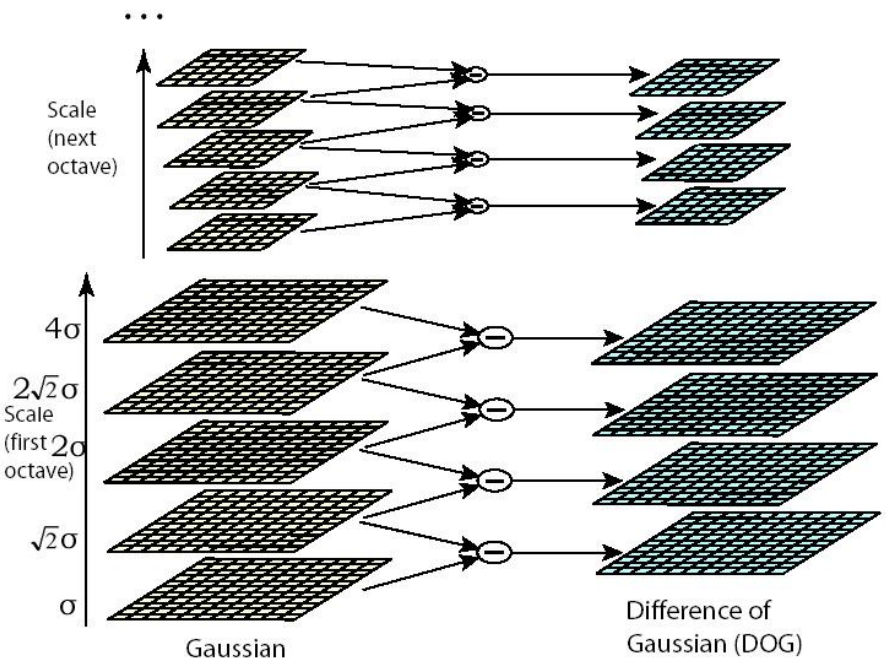


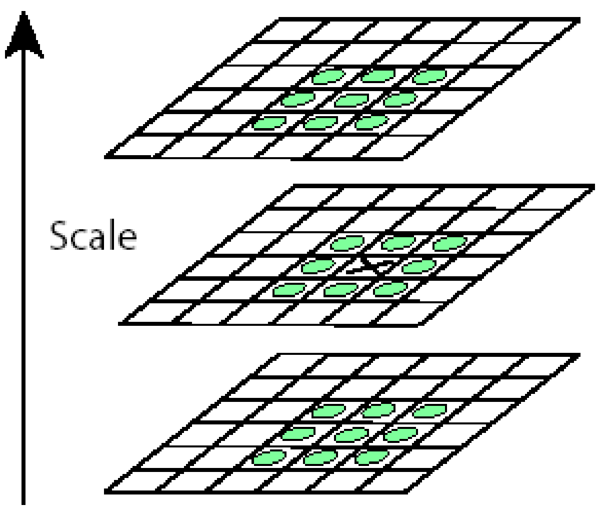


Different scale and rotated images

Scale Invariant Feature Transform(SIFT)

- We want to detect scale-invariant interest points

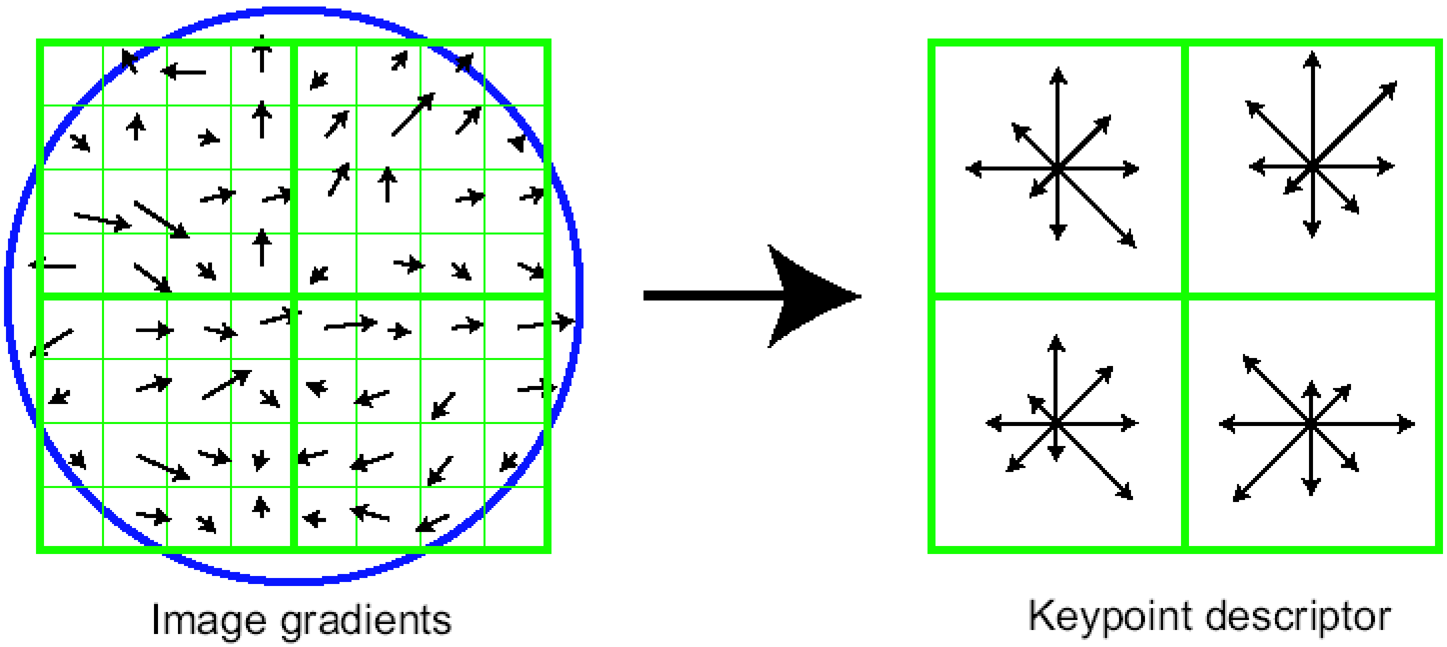




Scale-space Construction → Finding extrema

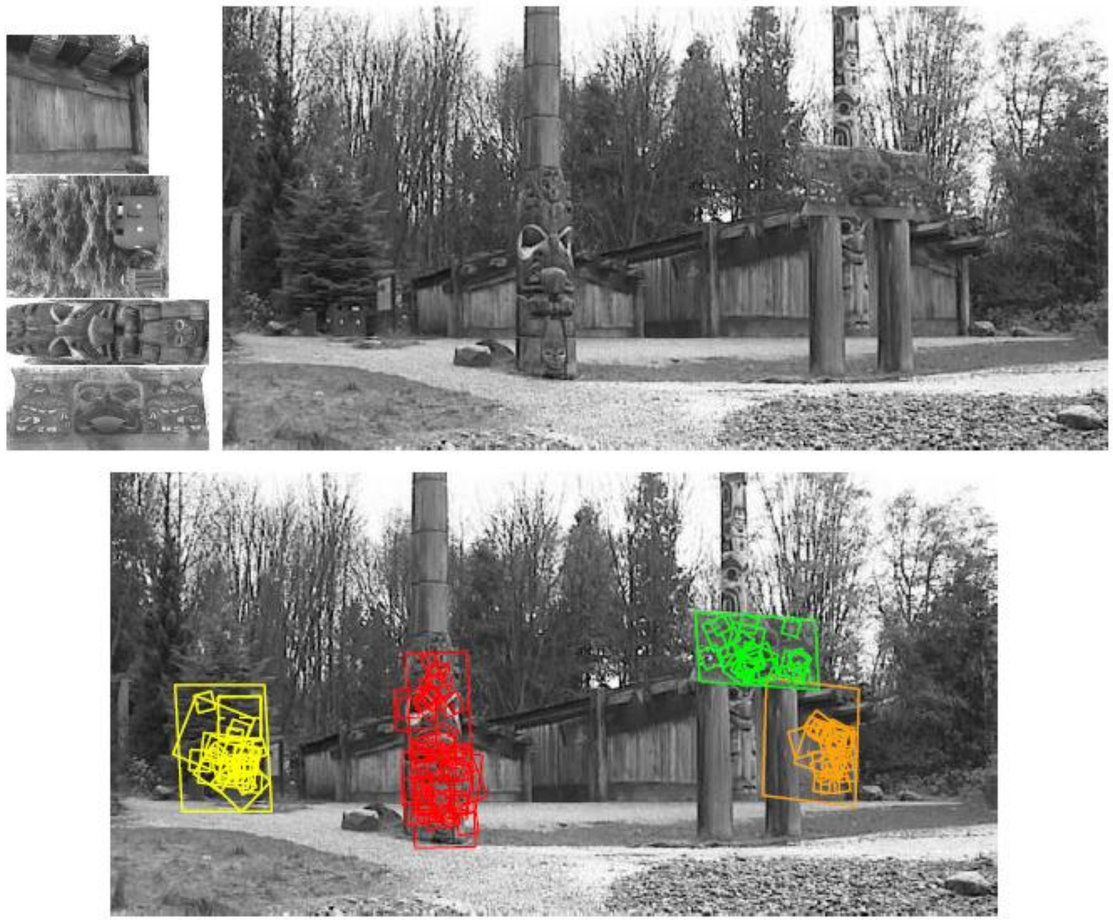
SIFT Descriptor

- The number of elements in the descriptor vector is calculated by the product of the number of histogram bins and the number of orientation directions typically 4x4x8 = 128



SIFT Results

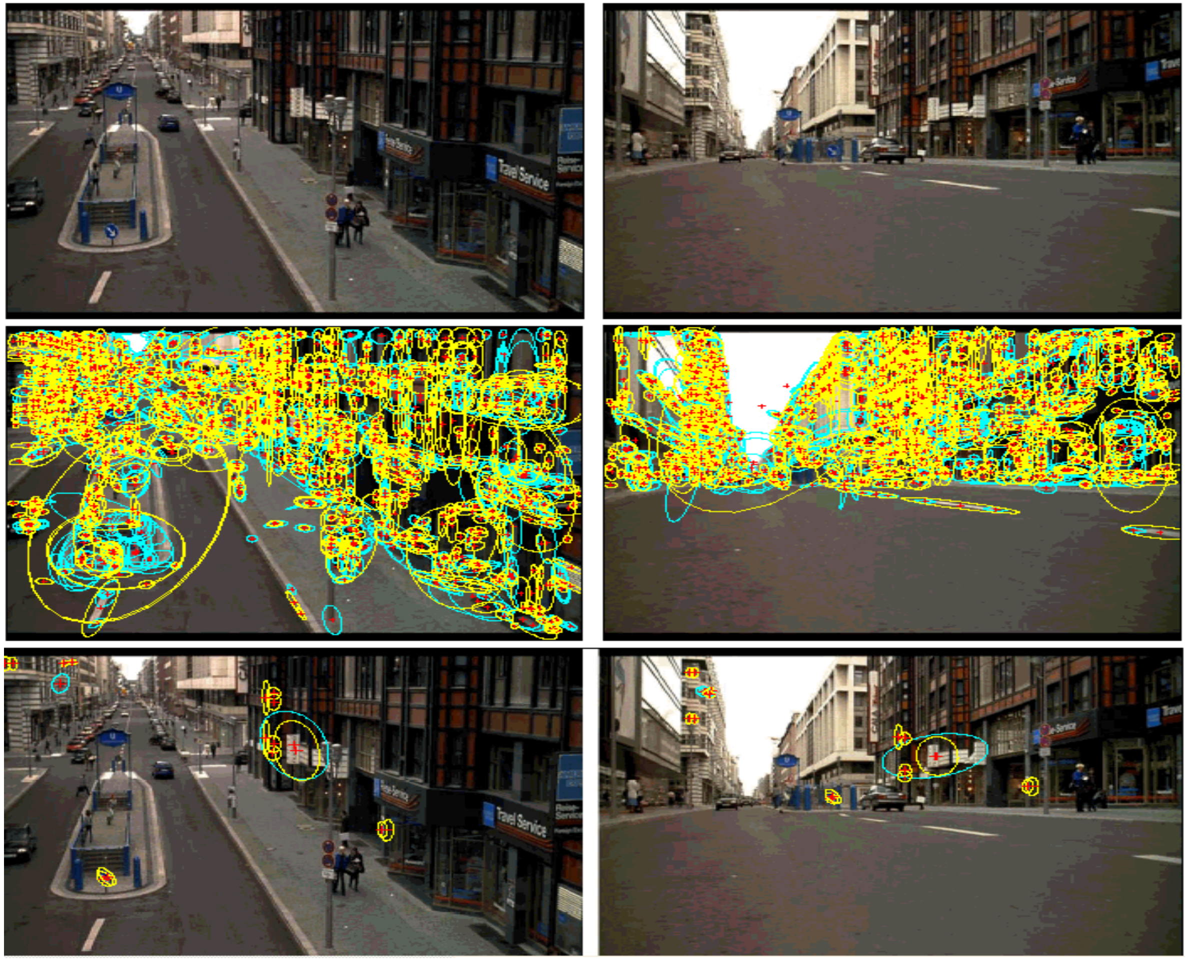
Given key images, find and trigger on them:



David G. Lowe, **Distinctive Image Features from Scale-Invariant Keypoints,** *IJCV,* 2003

SIFT Results

Find different views of same scene in video:



Josef Sivic and Andrew Zisserman, **Video Google: A Text Retrieval Approach to Object Matching in Videos,** ICCV 2003

Local Binary Patterns (LBP)

- LBP features are a powerful feature for texture classification
- The basic algorithm is as follows
  - For each pixel, compare it to each of its 8 neighbors.
  - If it is higher than a certain neighbor, assign a 1. Otherwise assign a 0.
  - Using the 8 bits computed, create a number from 0-255 representing the texture of the patch around that pixel.

Local Binary Patterns (LBP)

95	73	203
54	130	223
62	126	197

→

0	0	1
0		1
0	0	1

→ 00101001 → 41

Example of LBP feature computed on center pixel

Local Binary Patterns (LBP)

Original Image

LBP transformed image

Local Binary Patterns (LBP)

- The image is then divided into a grid of some size (e.g. a grid with non-overlapping patches of 16x16 pixels)
- In each patch, the histogram of the LBP values are computed (optionally normalize the histograms).
- All the patch histograms are concatenated into one feature vector for the image.

Histogram of Oriented Gradients (HOG)

- HOG features are computed in a similar manner to LBP features.
- The gradient at each pixel is first computed.
- This can be done by convolving the image with two filters

$$f_y = [-1\ 0\ 1] \qquad f_x = [-1\ 0\ 1]^T$$

- This gives the gradient in the y and x directions in the image.
- The magnitude and orientation of the gradient at each location can be computed

$$M = \sqrt{f_x^2 + f_y^2} \qquad O = \tan^{-1}\frac{f_y}{f_x}$$

Histogram of Oriented Gradients (HOG)

- Just like in LBP, the image is divided into patches.
- The histogram is computed by binning the orientations of the gradients however.
- One option is to just count how many times certain orientations appear in the patch or one can weight the count by the corresponding magnitude so that small gradients do not influence your histogram as much.
- The histograms are usually normalized afterwards.
- All the histograms are concatenated into one feature vector.

Histogram of Oriented Gradients (HOG)

- HOG features were used by Dalal and Triggs to create their pedestrian detector by passing the features through an SVM.
- There are a few parameters that can be manipulated such as the number of patches and the number of bins.
- When using these features, selecting the right parameters can make a huge difference.
- There are accepted parameters for certain problems lke pedestrian detection but they may not be the best for your problem.
- The only real way to find a good set of parameters is to try them and take the best.