Introduction to Computer Vision

3. Feature Extraction and Classification

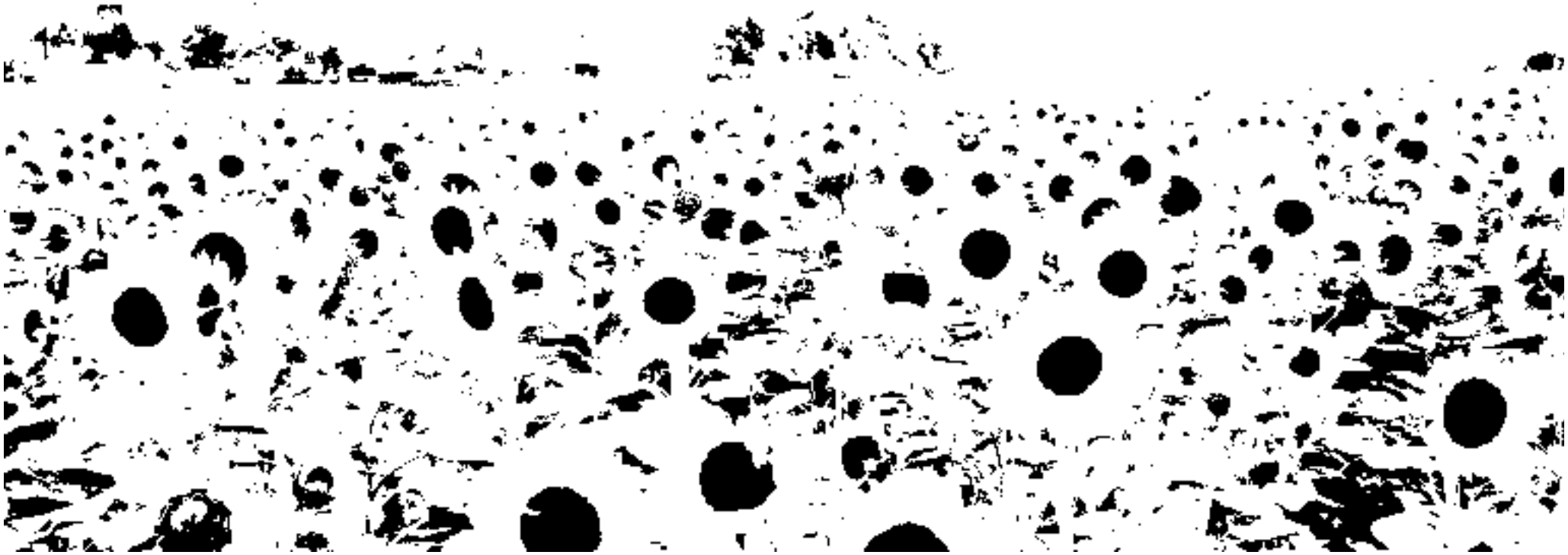
UCLA – CS 188 – Fall 2019

Fabien Scalzo, Ph.D.

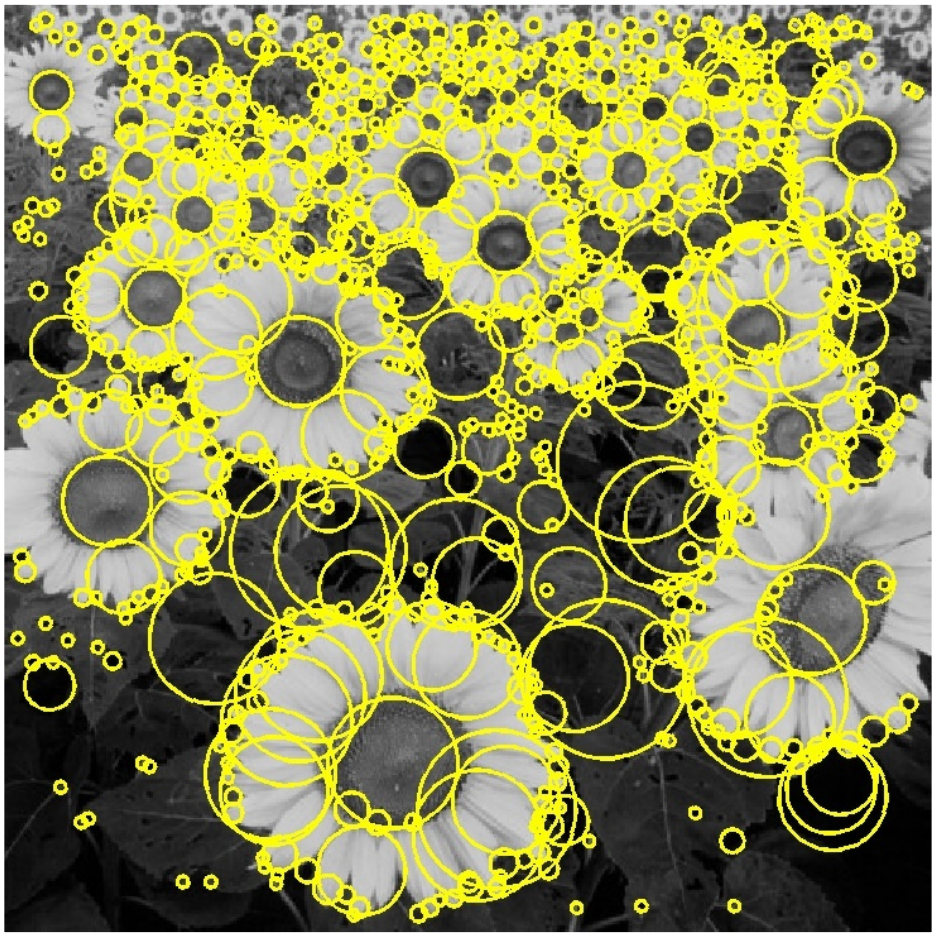
Part I: Feature Extraction
Part II: Clustering and Classification

Week 1			26-Sep	**Introduction**
Week 2	1-Oct	**Basic Image Processing**	3-Oct	**Feature Extraction and Classification**
Week 3	8-Oct	**Feature Tracking/Optical Flow**	10-Oct	**2D Image transformations, RANSAC**
Week 4	15-Oct	**SVD, 2D camera model, projective plane**	17-Oct	**Euclidean geometry, rigid body motion**
Week 5	22-Oct	**Epipolar Geometry**	24-Oct	**3D Cameras and processing**
Week 6	29-Oct	**Midterm**	31-Oct	**Statistical decision theory/Pattern Recognition**
Week 7	5-Nov	**Deep Learning**	7-Nov	**Deep Learning for Image Classification**
Week 8	12-Nov	**Object Detection**	14-Nov	**Generative models**
Week 9	19-Nov	**Medical Imaging**	21-Nov	**Autonomous Navigation**
Week 10	26-Nov	**Guest Lecture (Nikhil Naik)**	28-Nov	
Week 11	3-Dec	**Recursive 3D reconstruction and pose estimation**	5-Dec	**Recap**
Week 12	10-Dec		12-Dec	**Final**

Which features best describe this image?



Blob detection



Automatic Feature Extraction from an image

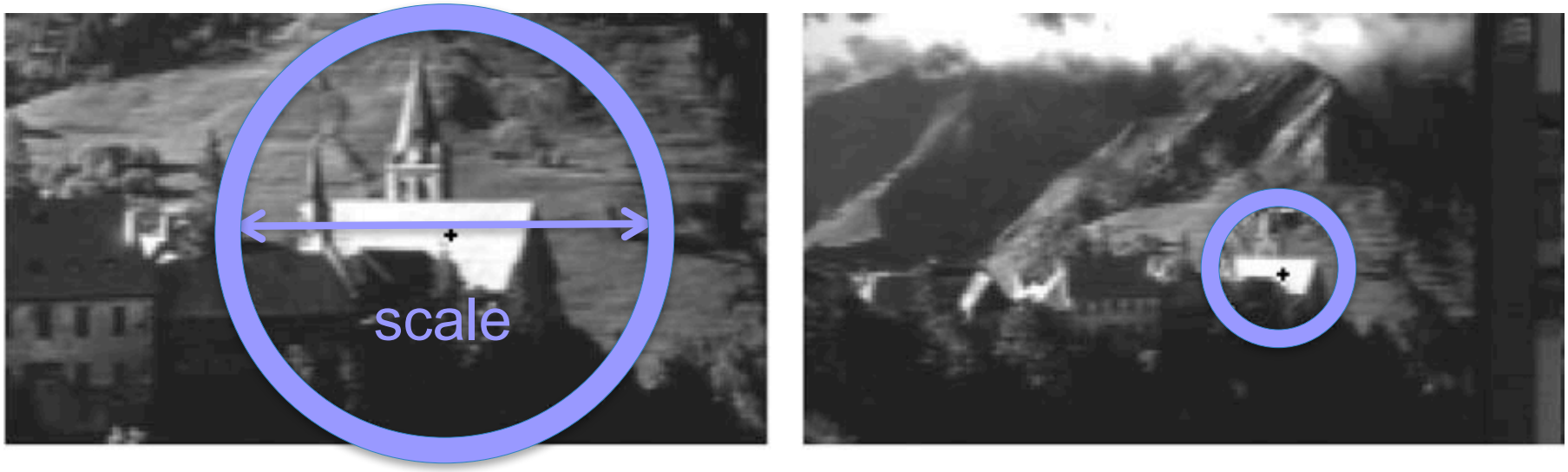
- Detection: Where is it?
- Description: What does it look like?

should be robust to common variations

- ✓ Intensity
- ✓ Scaling
- ✓ Viewpoint

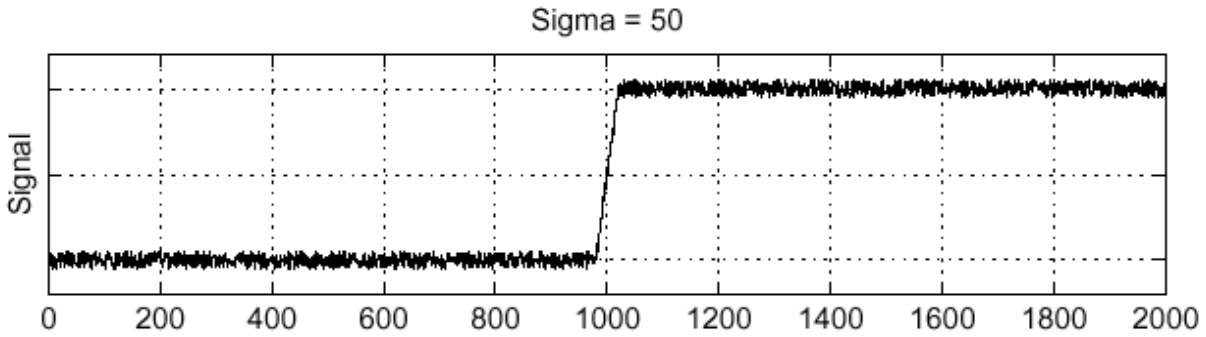
Blob detection and Scale invariance

- Goal: independently detect corresponding regions in scaled versions of the same image
- Characteristic region size that is ***covariant*** with the image. Content description is ***invariant*** to transformation.



Edge detection, version 1

f

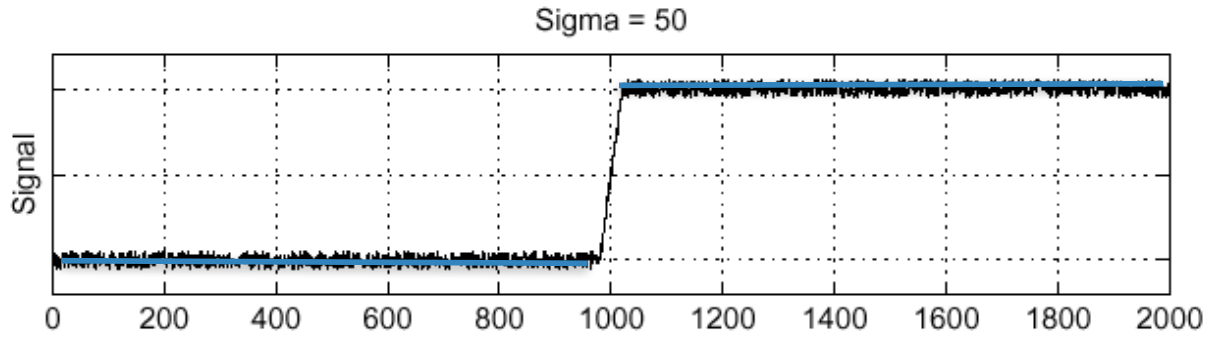


Edge

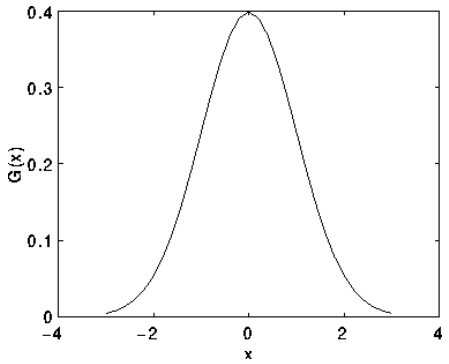
1	0	-1

Edge detection, version 1

f



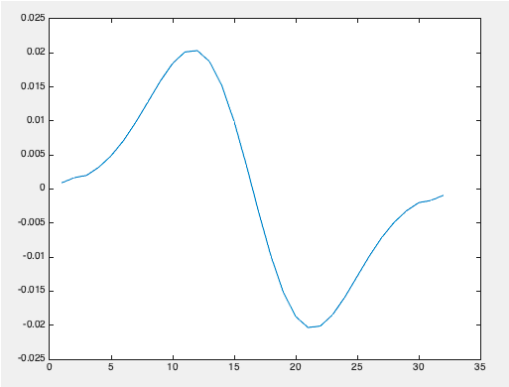
Edge



Gaussian filter

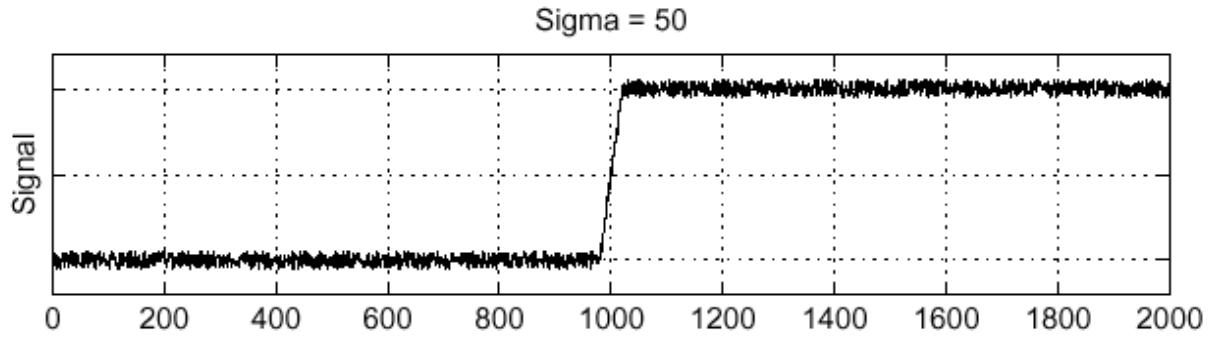
*

1	0	-1



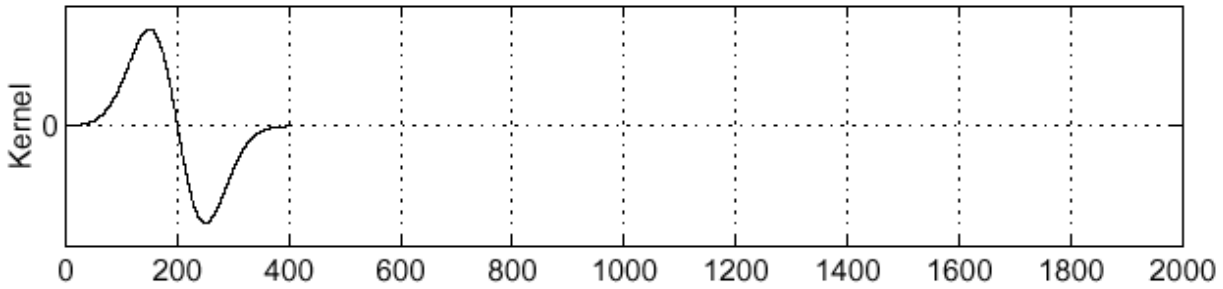
Edge detection, version 1

f



Edge

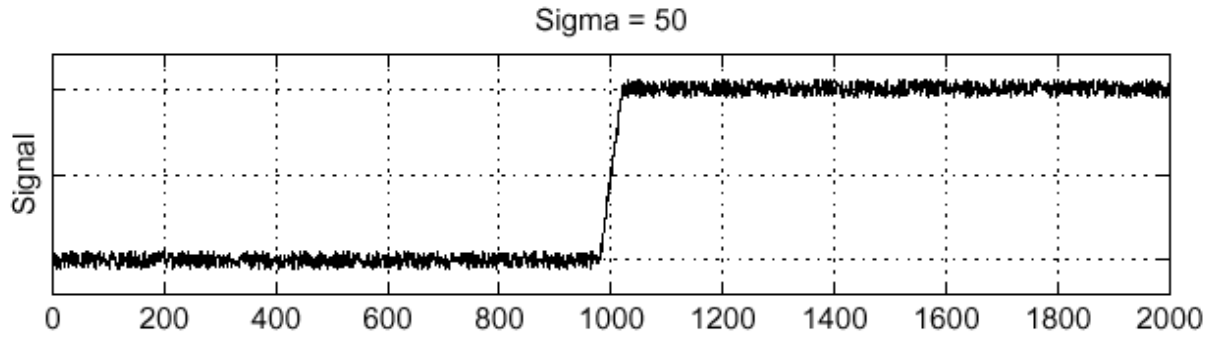
$\frac{d}{dx} g$



Derivative
of Gaussian

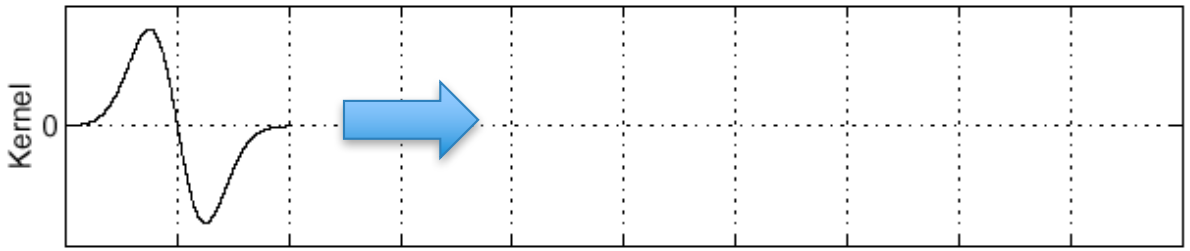
Edge detection, version 1

f

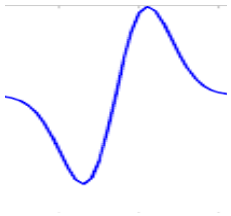


Edge

$\frac{d}{dx} g$

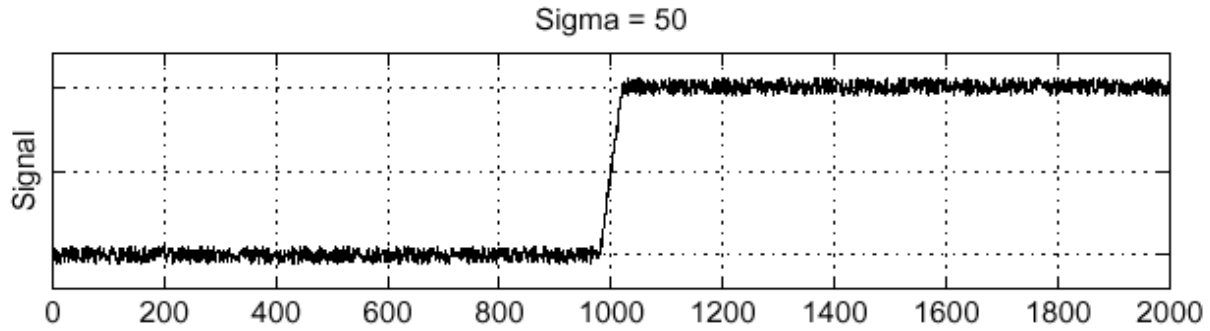


Derivative of Gaussian



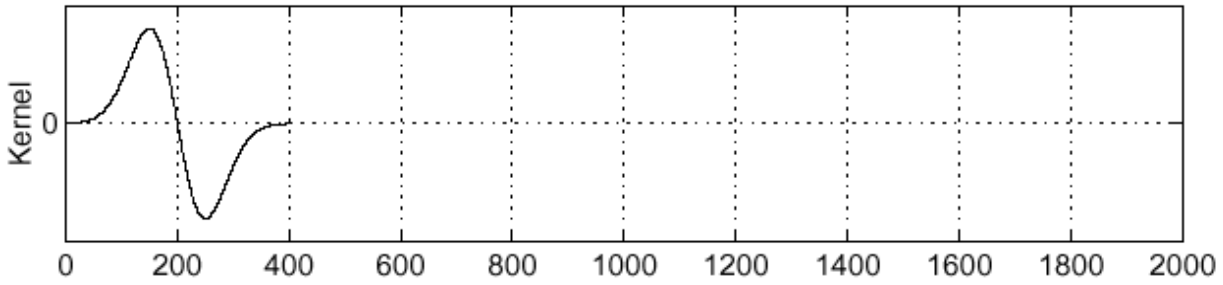
Edge detection, version 1

f



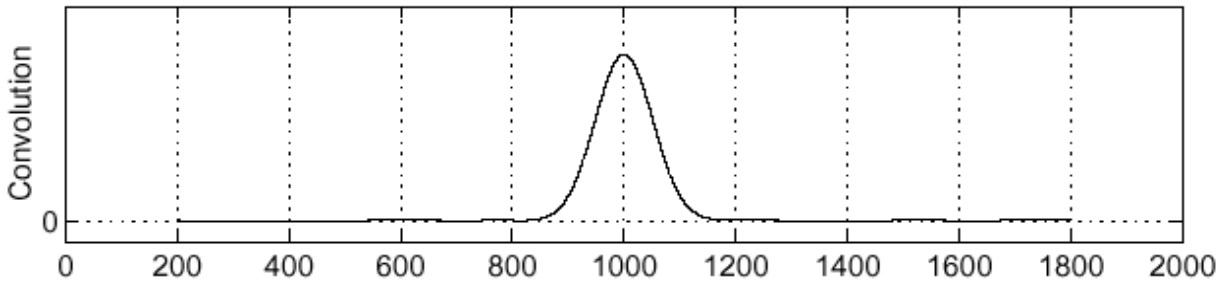
Edge

$\frac{d}{dx} g$



Derivative of Gaussian

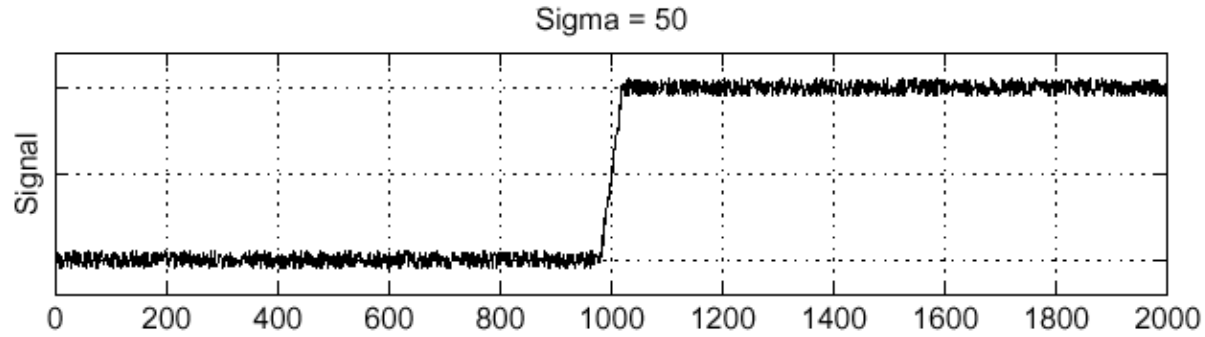
$f * \frac{d}{dx} g$



Edge = extrema of derivative

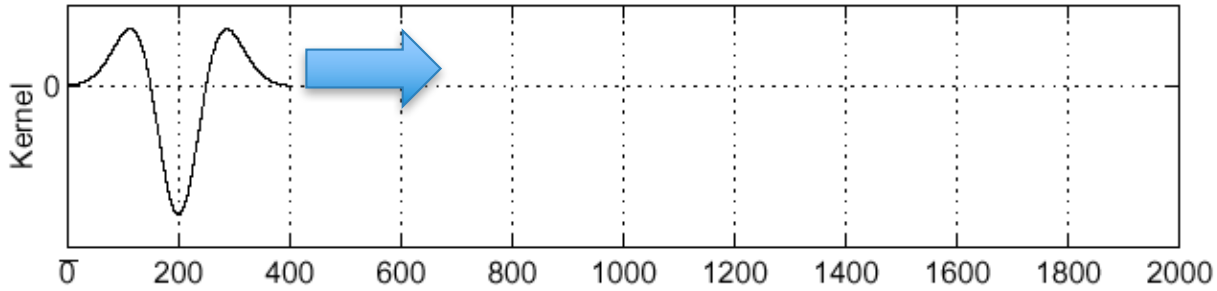
Edge detection, version 2

f



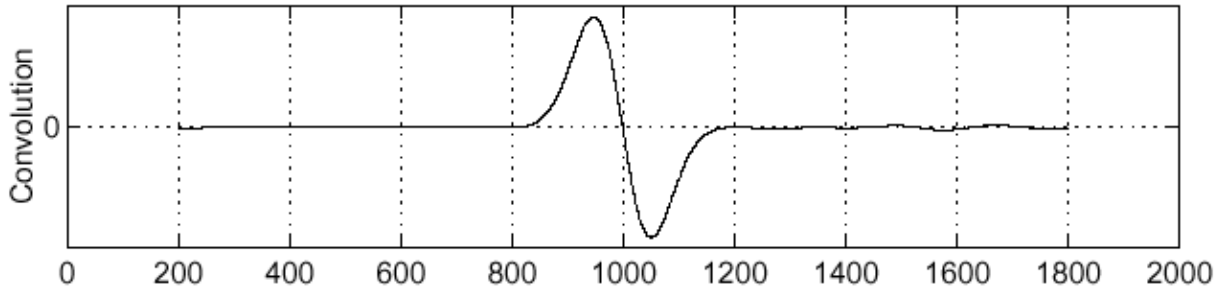
Edge

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

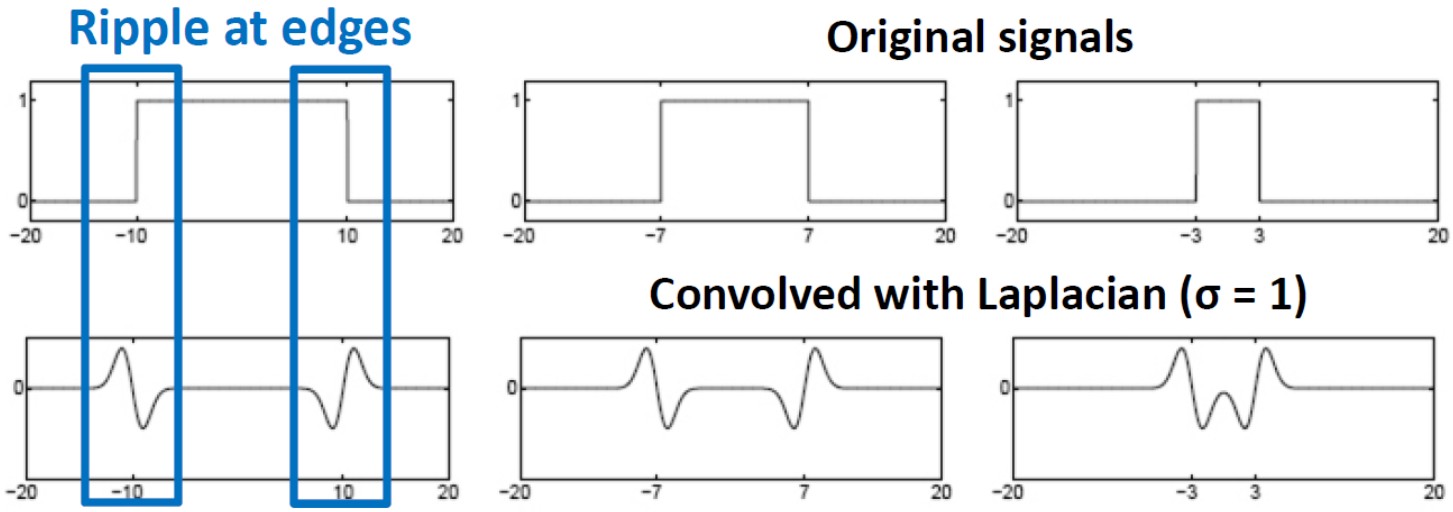


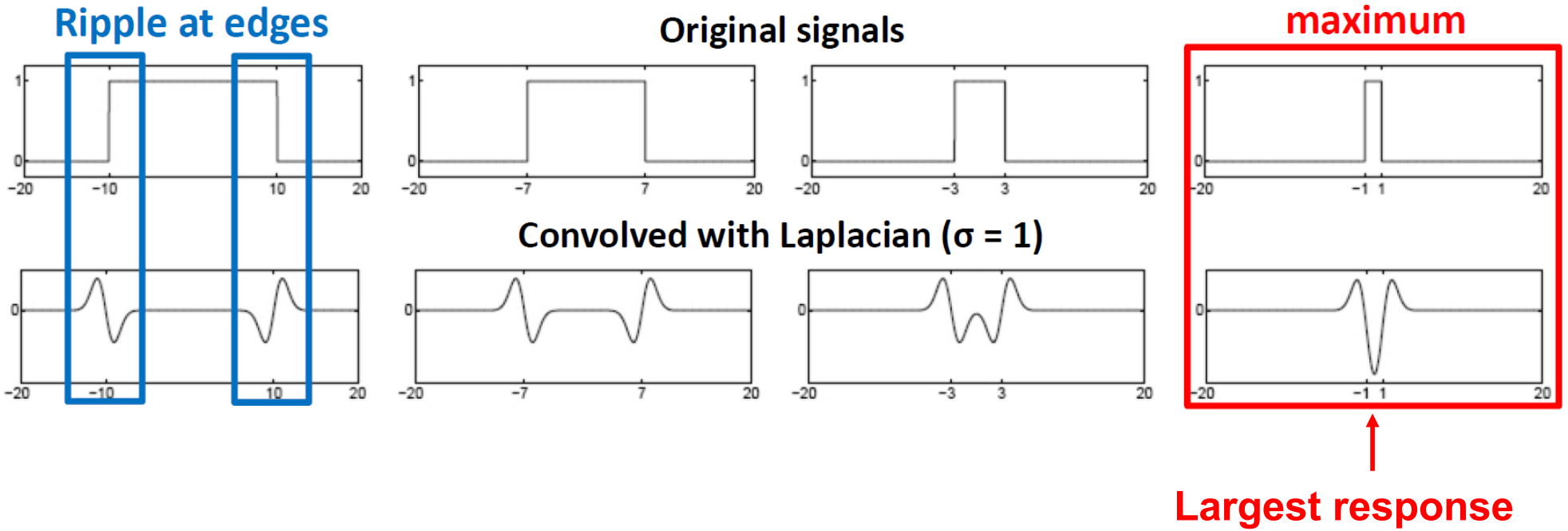
Second derivative of Gaussian (Laplacian)

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



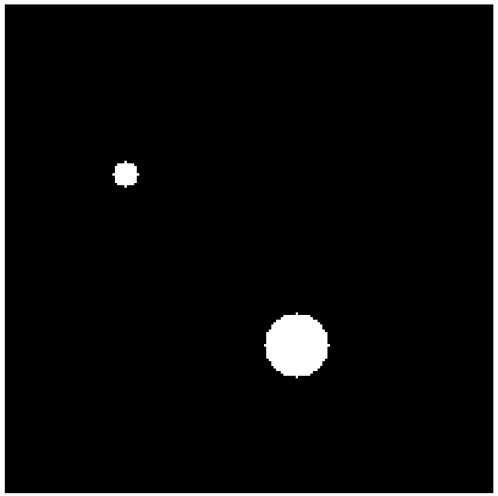
Edge = zero crossing of second derivative

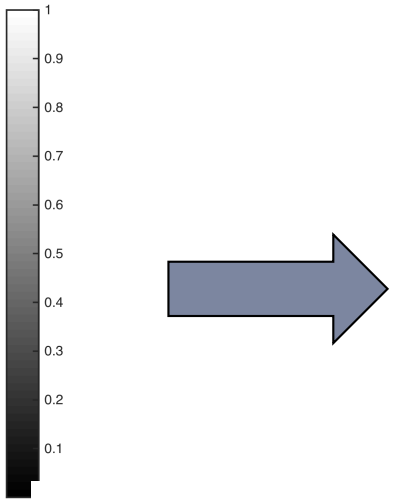

Ripple at edges
Original signals
Convolved with Laplacian (σ = 1)
1
0
−20
−10
10
20
−7
7
−3
3

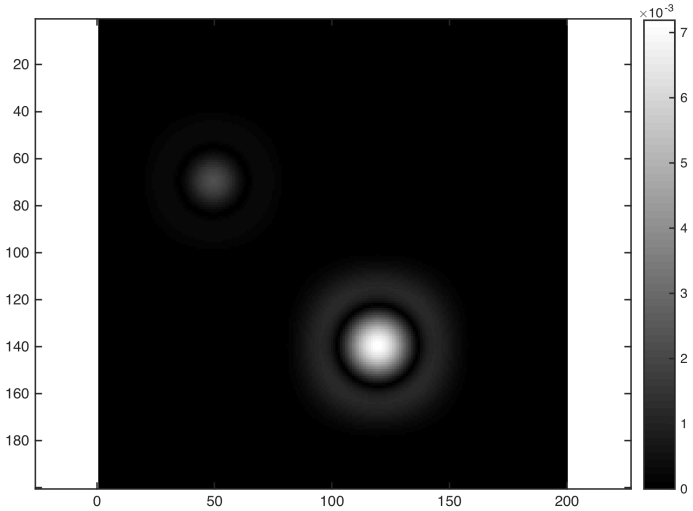


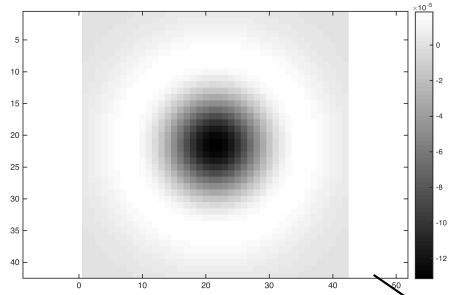
The largest response will be at the center of the blob;
provided the scale of the Laplacian is “matched” to the scale of the blob

Blob Detector

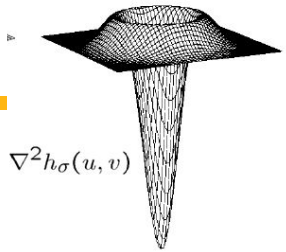






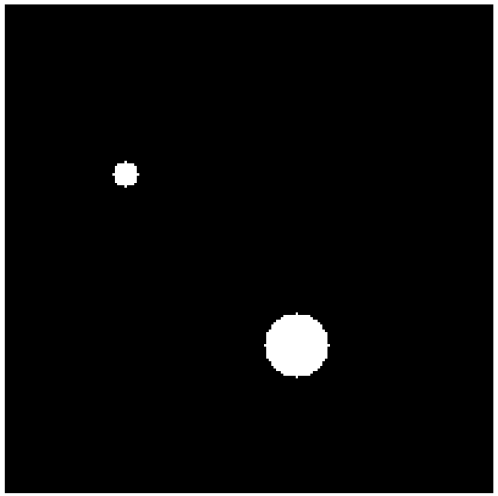


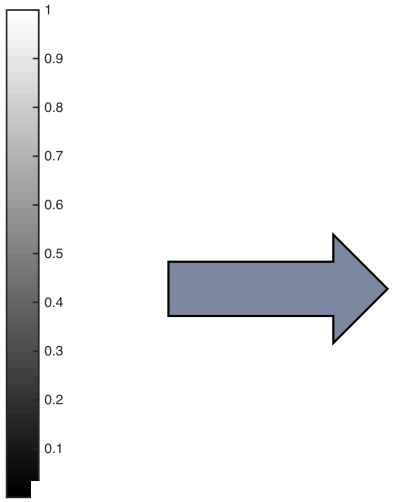
Laplacian of Gaussian

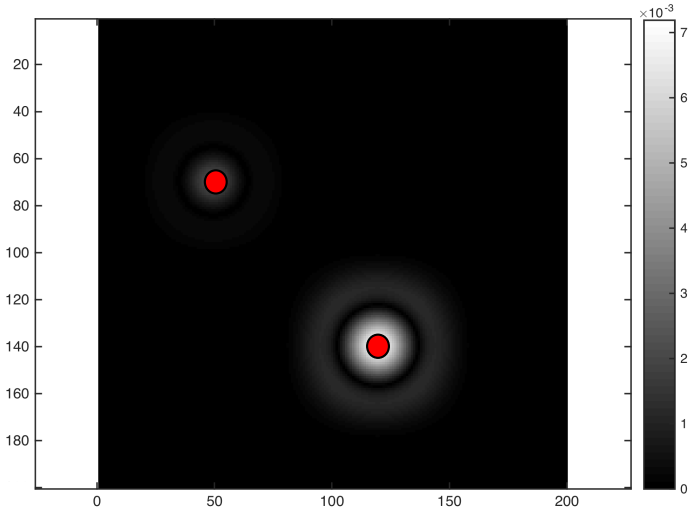


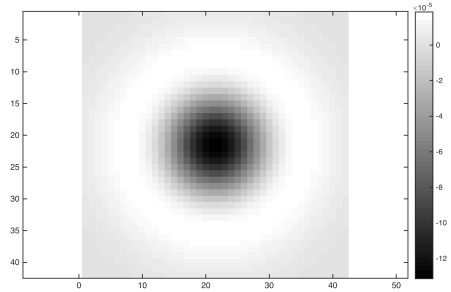
$\nabla^2 h_\sigma(u, v)$

Blob Detector





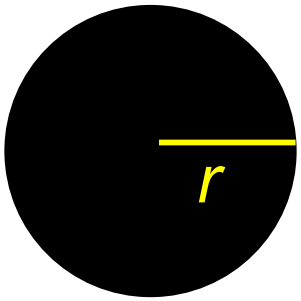




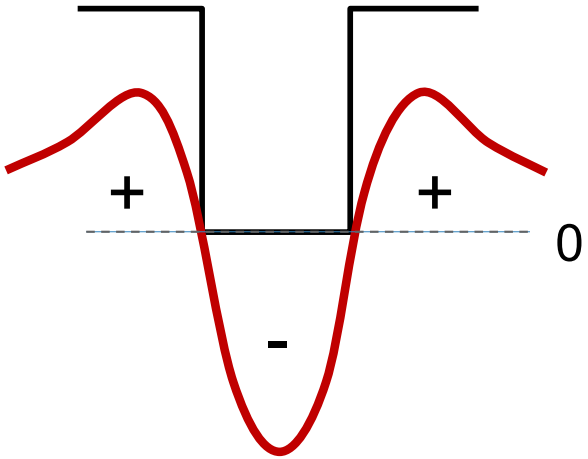
Local maxima

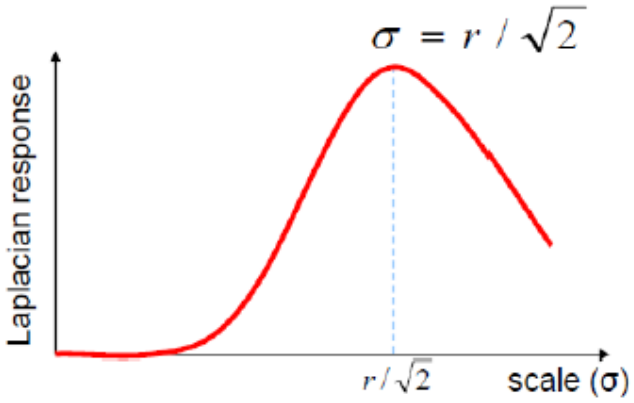
Scale selection

- To get maximum response, the zeros of the Laplacian have to be aligned with the circle.
- The Laplacian achieves a maximum response at a scale $\sigma = r / \sqrt{2}.$



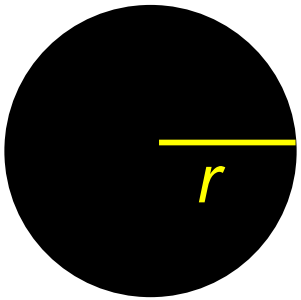
image



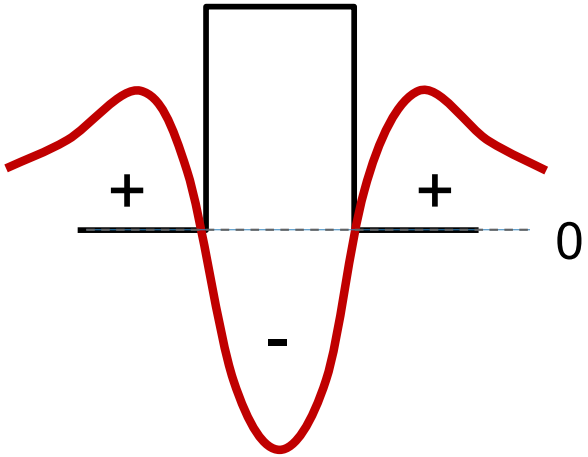


Scale selection

- To get maximum response, the zeros of the Laplacian have to be aligned with the circle.
- The Laplacian achieves a maximum response at a scale $\sigma = r / \sqrt{2}.$

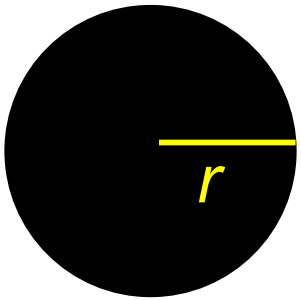


image

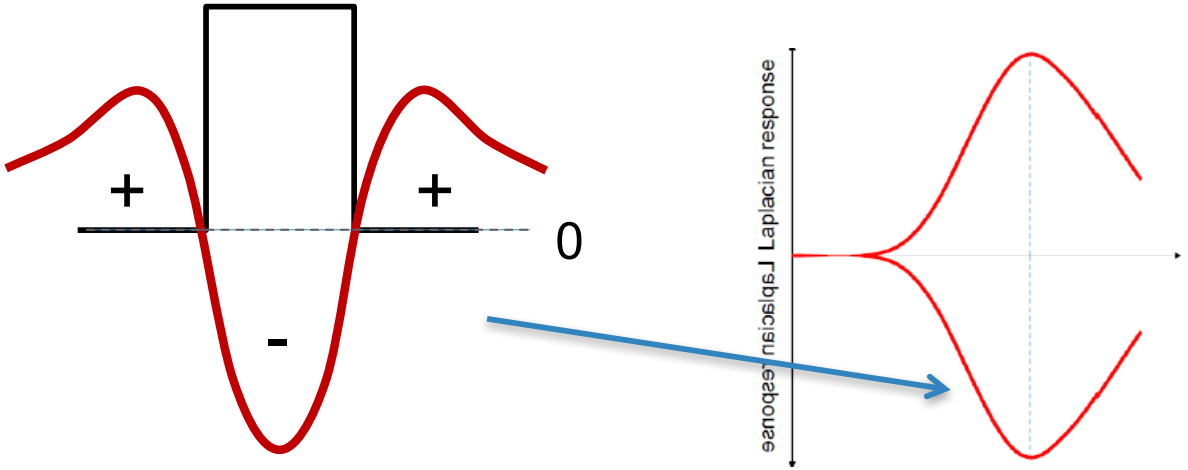


Scale selection

- To get maximum response, the zeros of the Laplacian have to be aligned with the circle.
- The Laplacian achieves a maximum response at a scale $\sigma = r / \sqrt{2}.$



image

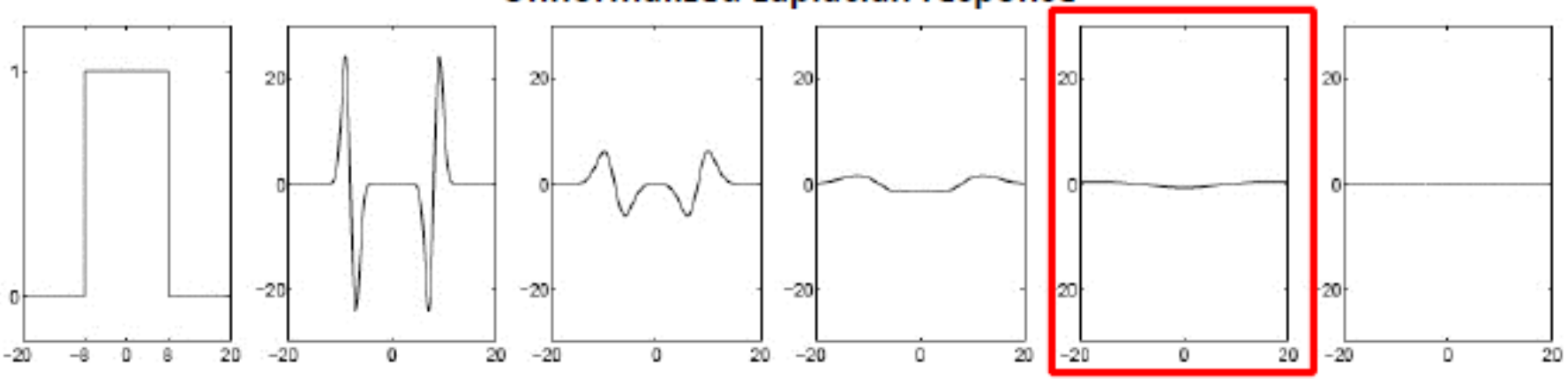


Effect of scale normalization

- However, the magnitude of the Laplacians depends on scale.

increasing σ ⟶

Unnormalized Laplacian response



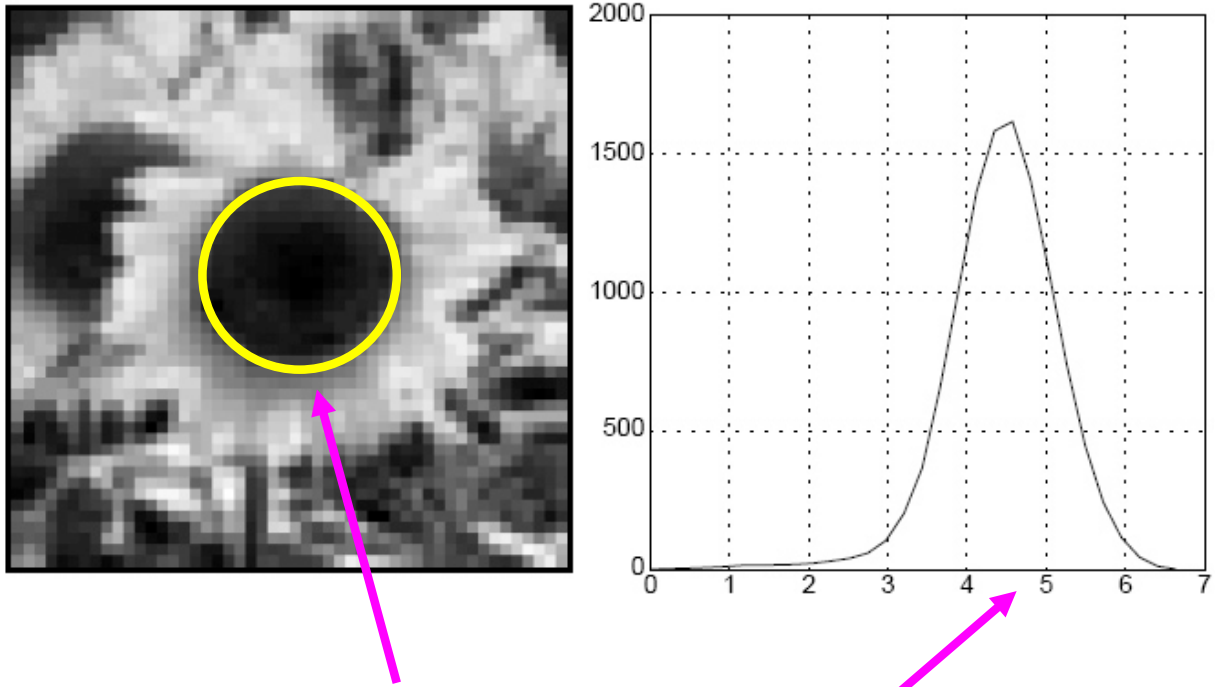
original signal
(radius=8)

increasing σ ⟶

optimal scale

Characteristic scale

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



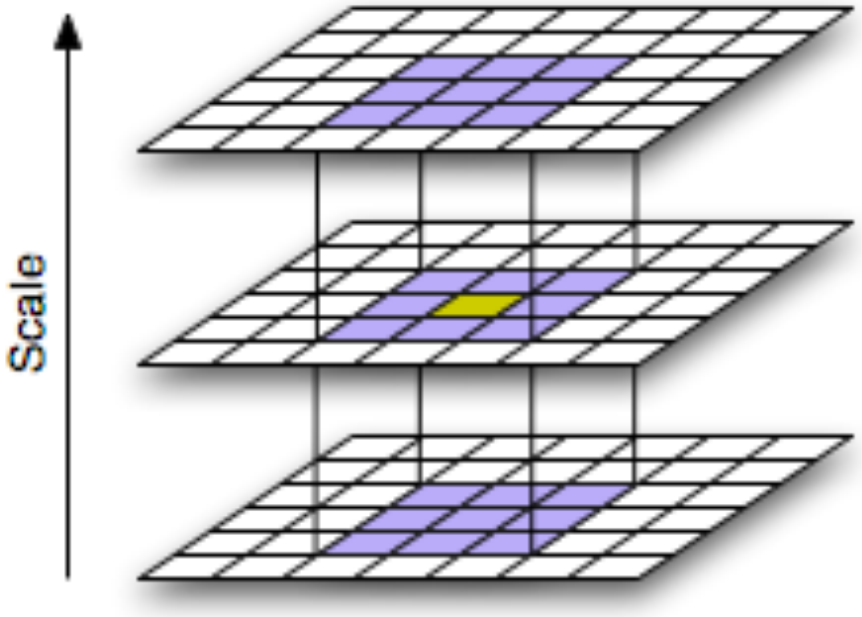
Scale-space blob detector: Example

Scale-space blob detector: Example

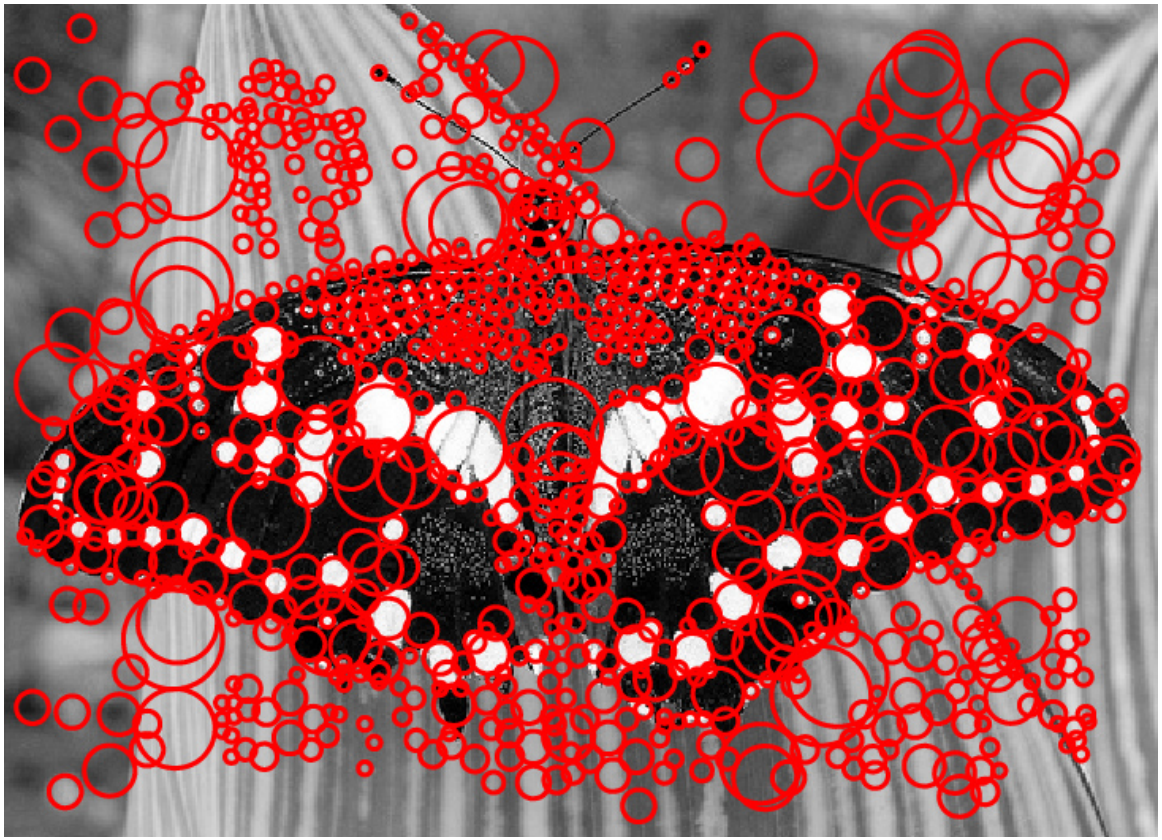
sigma = 11.9912

Scale-space blob detector

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

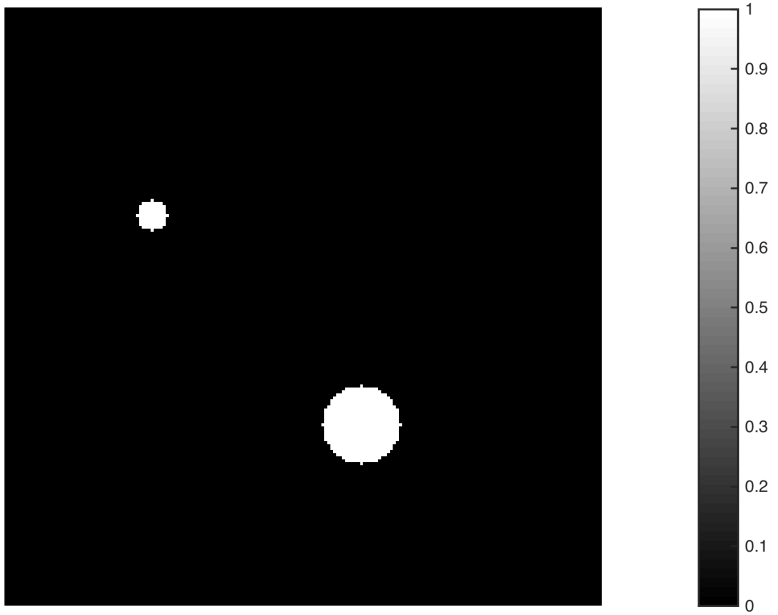


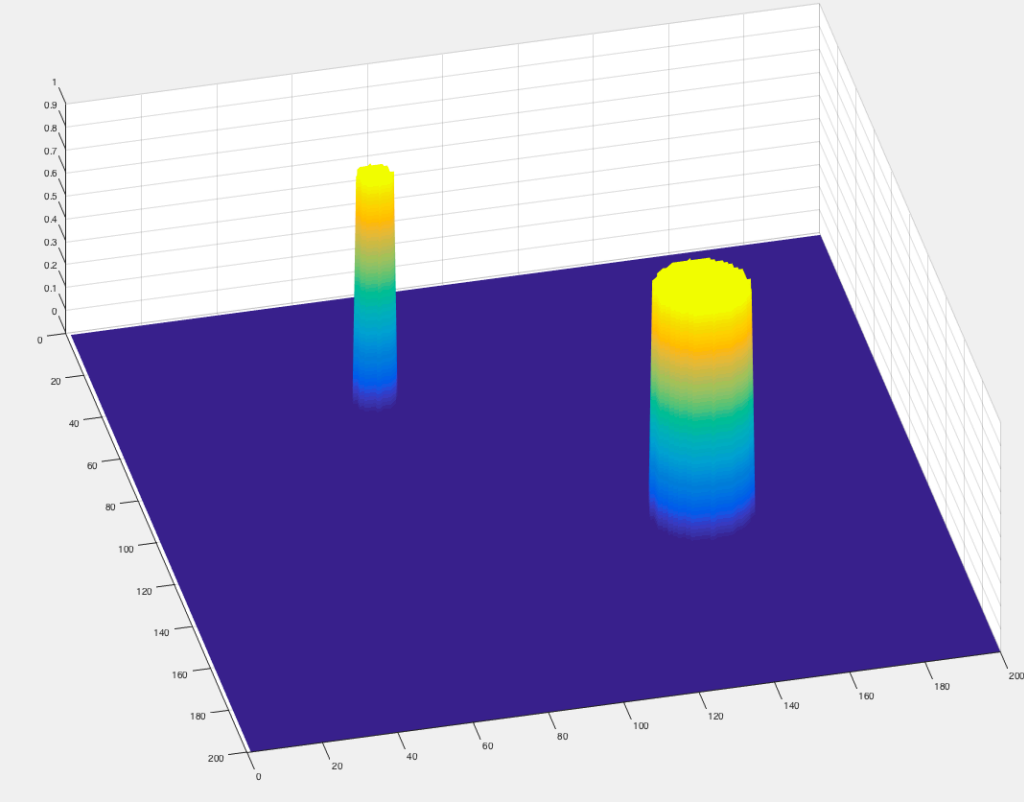
Scale-space blob detector: Example



Spatial + Scale Selection

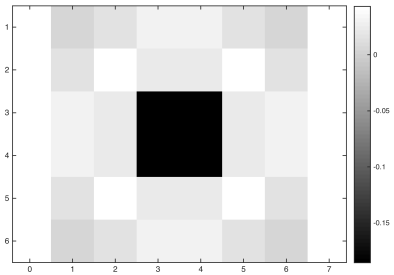
- Purpose: to identify the location and scale of objects

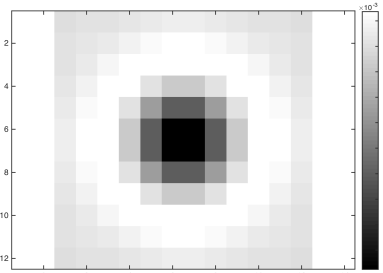


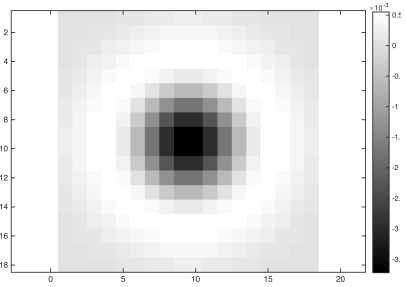


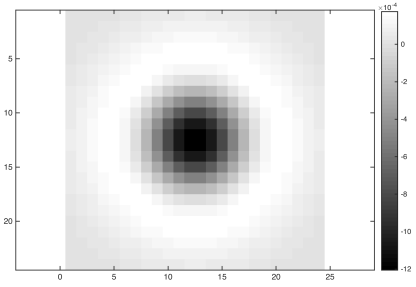
2D Laplacian filter

Filter size = 6 * σ

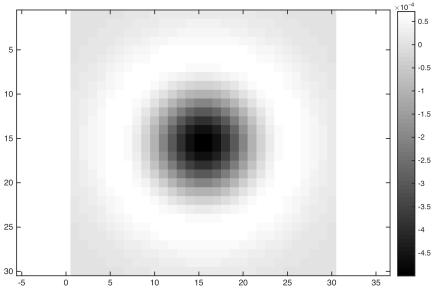


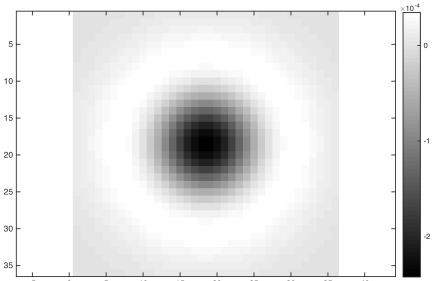


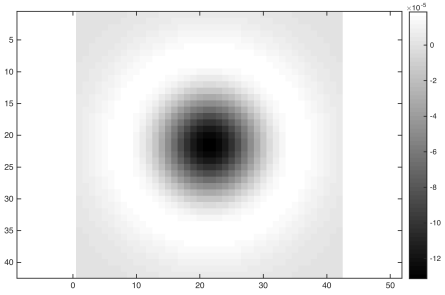


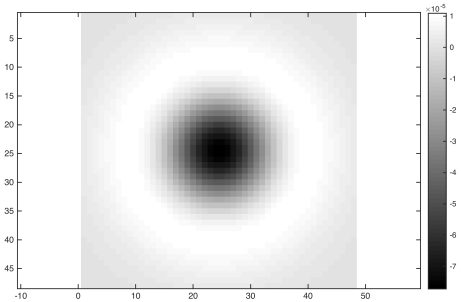


σ = 1 2 3 4

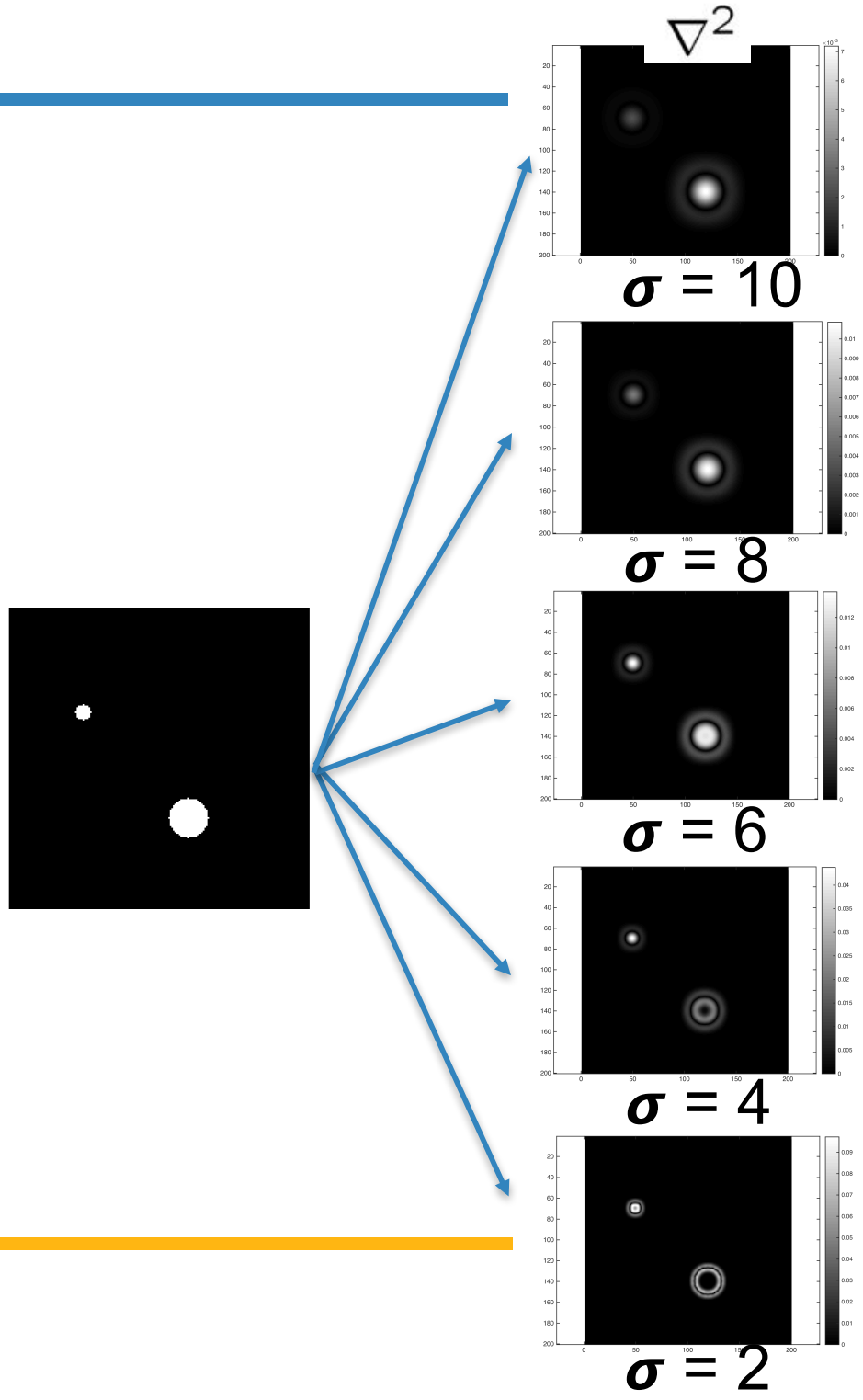


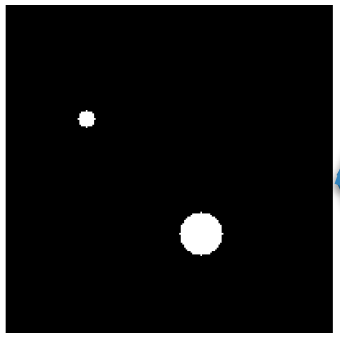




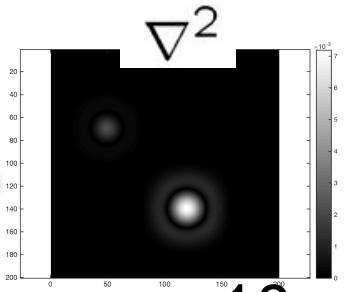


5 6 7 8

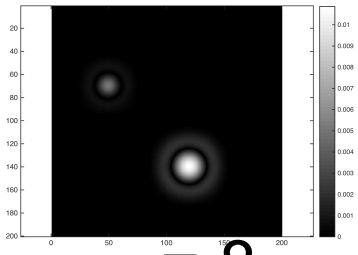

∇^2
$\sigma = 10$
$\sigma = 8$
$\sigma = 6$
$\sigma = 4$
$\sigma = 2$



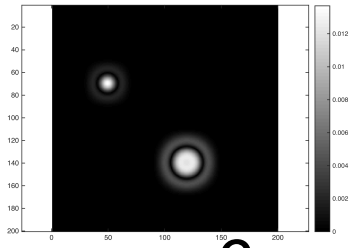
∇^2



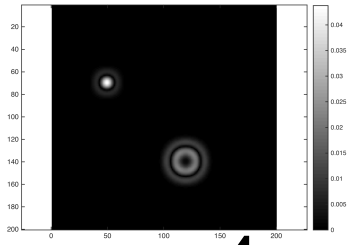
$\sigma = 10$



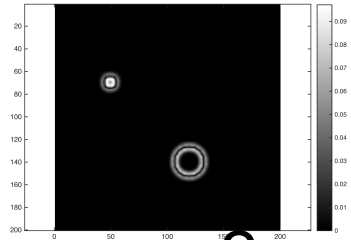
$\sigma = 8$



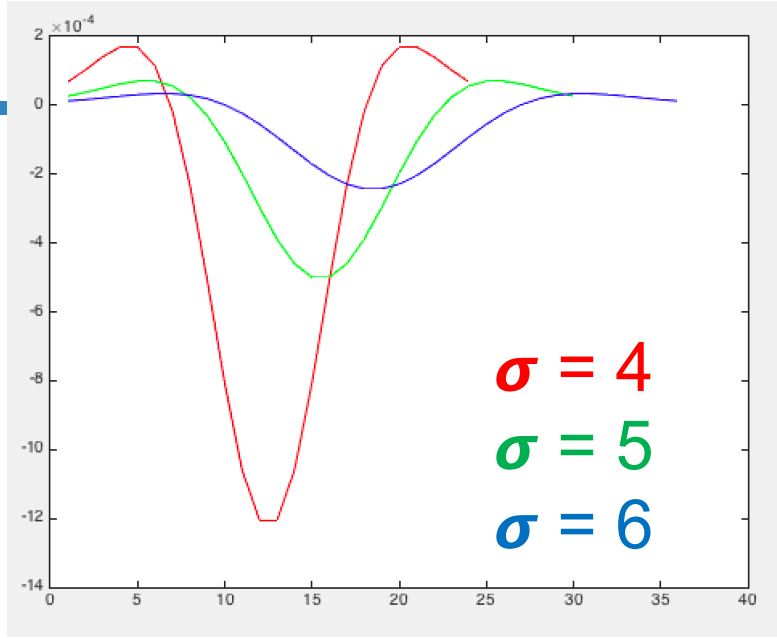
$\sigma = 6$



$\sigma = 4$



$\sigma = 2$

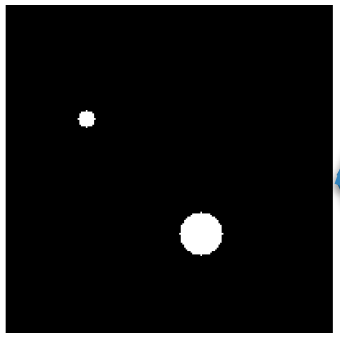


Range is different

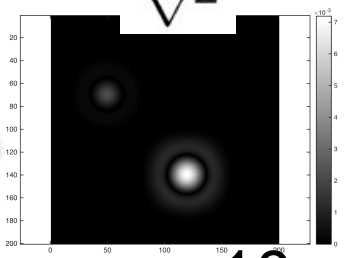
(not directly comparable)

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

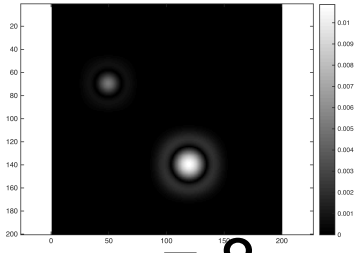
We adjust for the scale



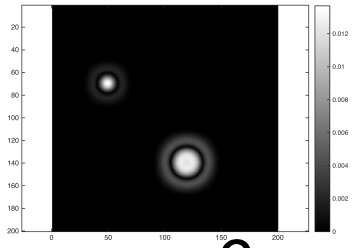
∇^2



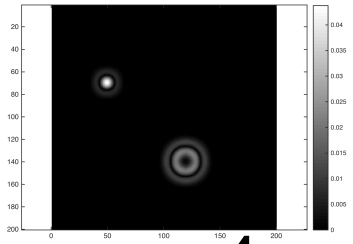
$\sigma = 10$



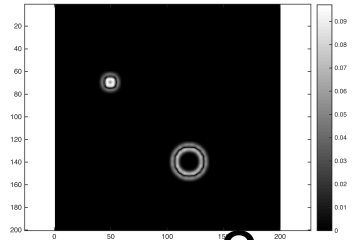
$\sigma = 8$



$\sigma = 6$

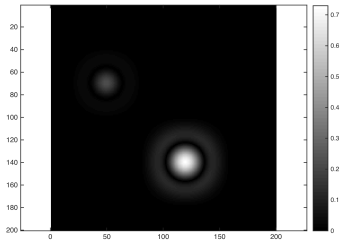


$\sigma = 4$

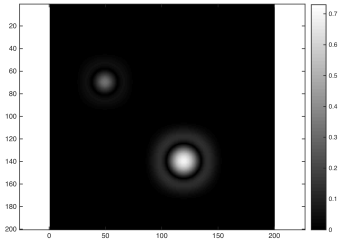


$\sigma = 2$

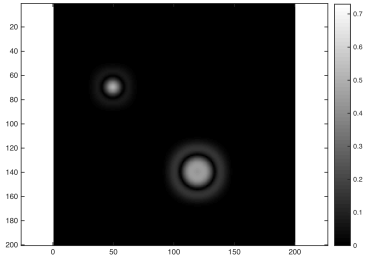
$* \sigma^2$



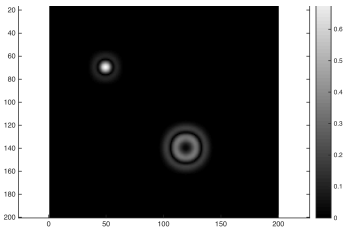
$* \sigma^2$



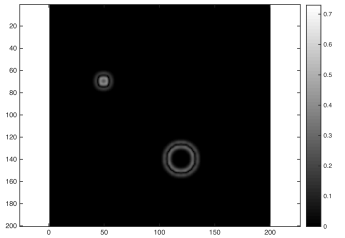
$* \sigma^2$



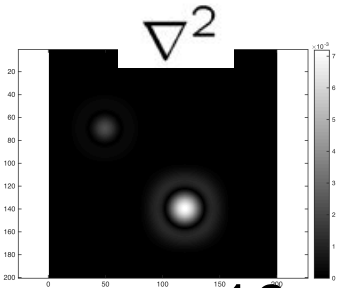
$* \sigma^2$



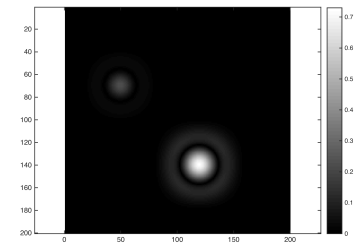
$* \sigma^2$

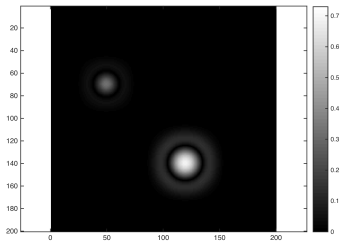


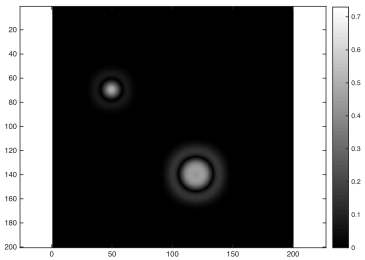
∇^2_{norm}

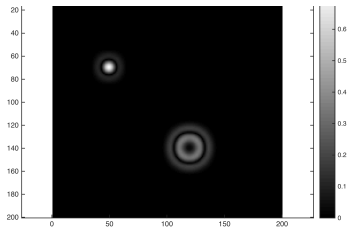

∇^2
σ = 10
* σ^2
∇^2_{norm}
σ = 8
σ = 6
σ = 4
σ = 2
Threshold
> 0.4

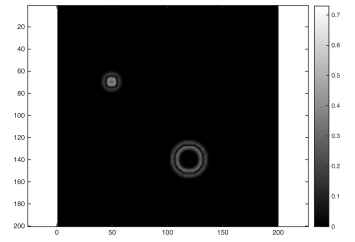
∇^2_{norm}





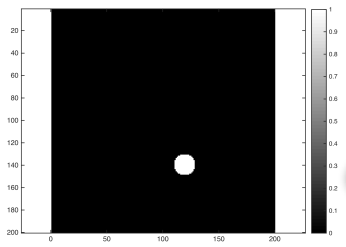


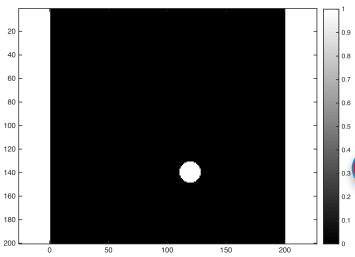


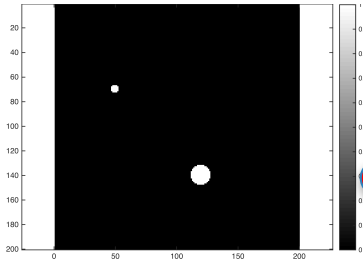


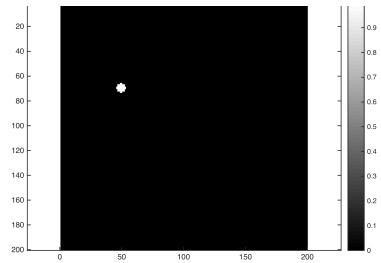
Threshold

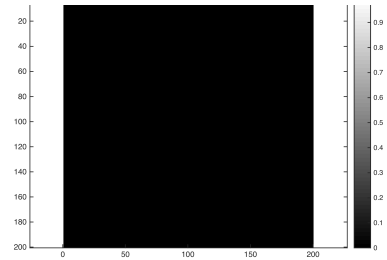
> 0.4











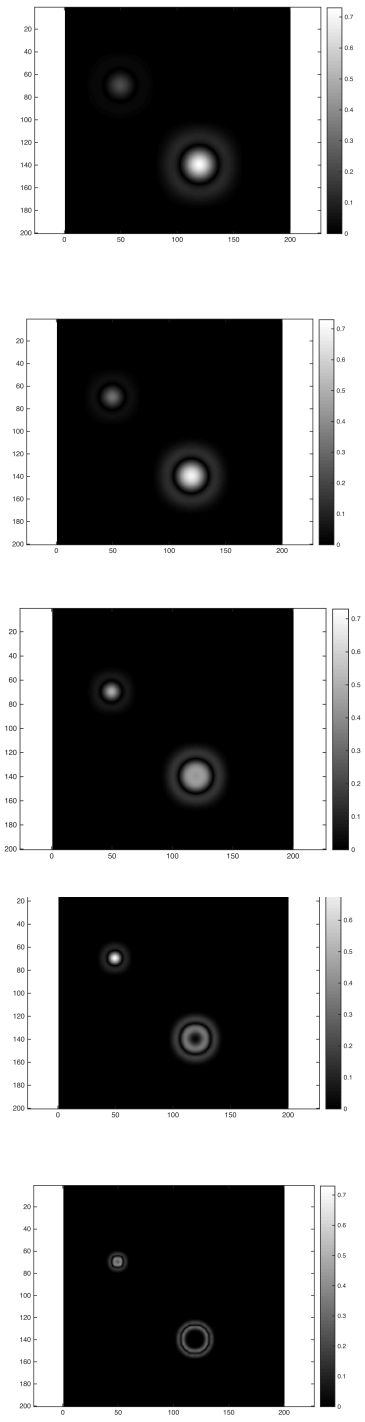
Local maxima #1

scale = 8

radius = 8 * sqrt(2)

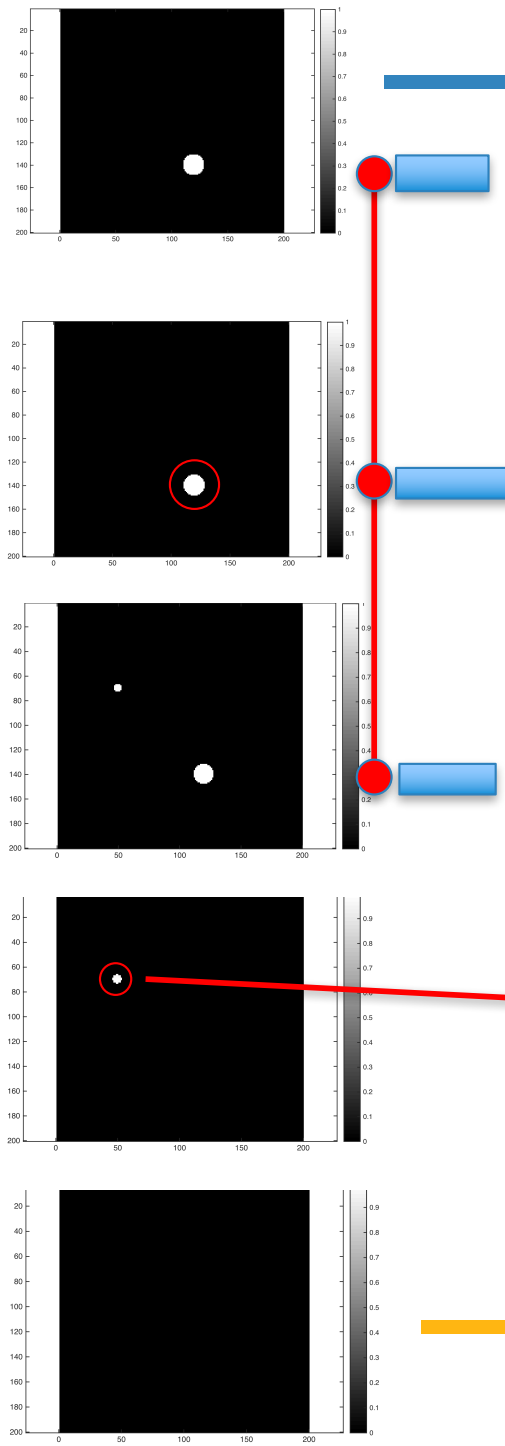
= 11.3 pixels

∇^2_{norm}



Threshold

> 0.4



Local maxima #1

scale = 8

radius = 8 * sqrt(2)

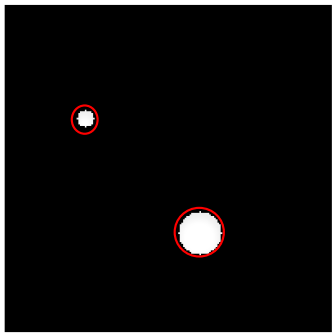
= 11.3 pixels

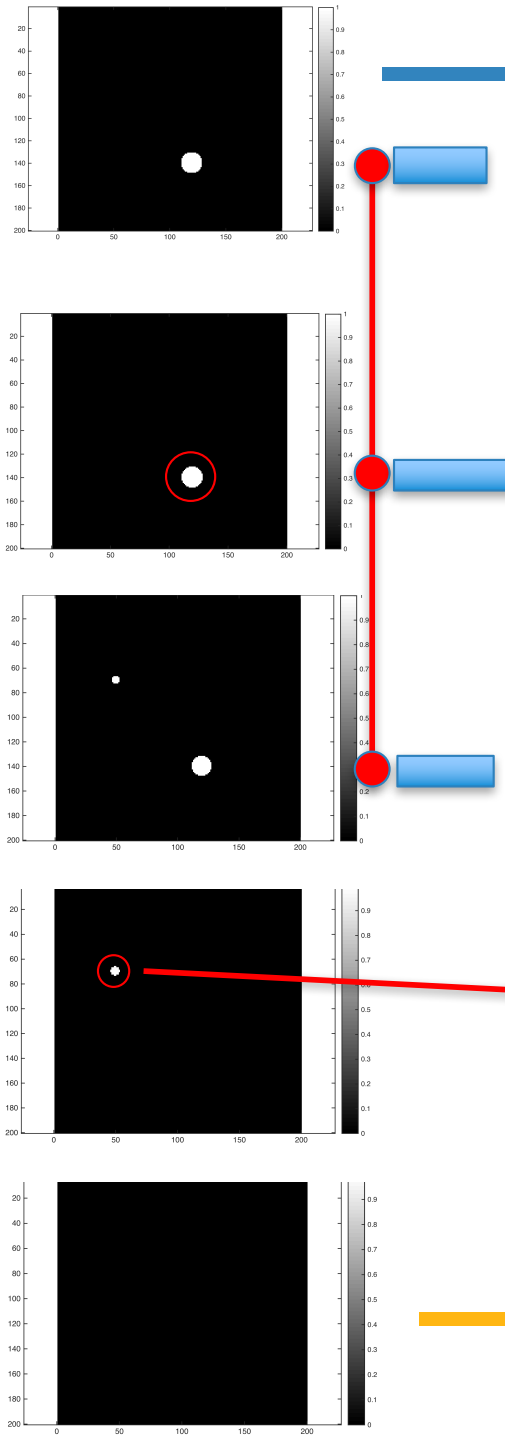
Local maxima #2

scale = 4

radius = 4 * sqrt(2)

= 5.6 pixels





Local maxima #1

scale = 8

radius = 8 * sqrt(2)

= 11.3 pixels

Local maxima #2

scale = 4

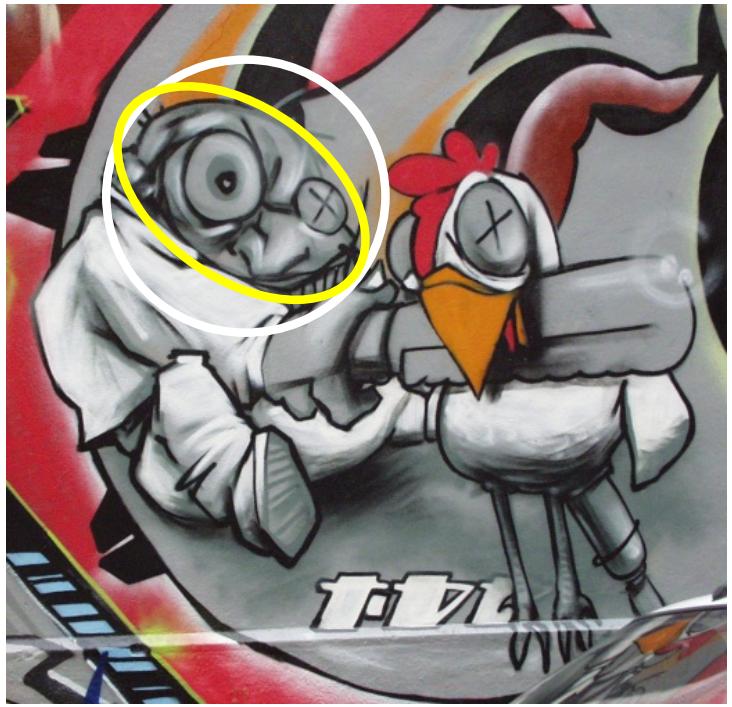
radius = 4 * sqrt(2)

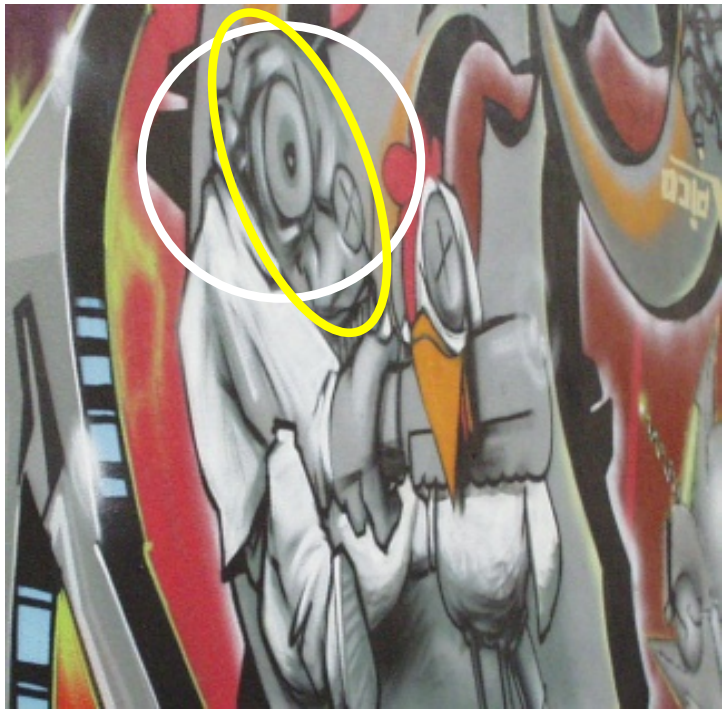
= 5.6 pixels

Not always circular

Affine covariance

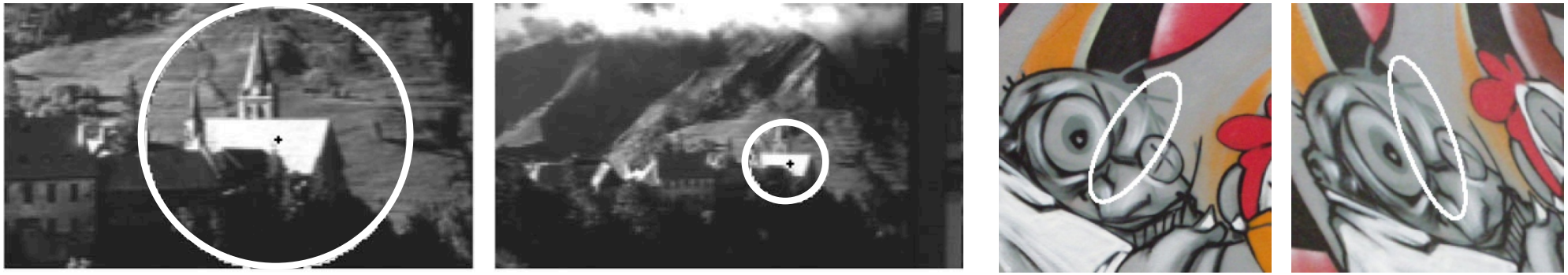
- Affine transformation approximates viewpoint changes (for planar objects)





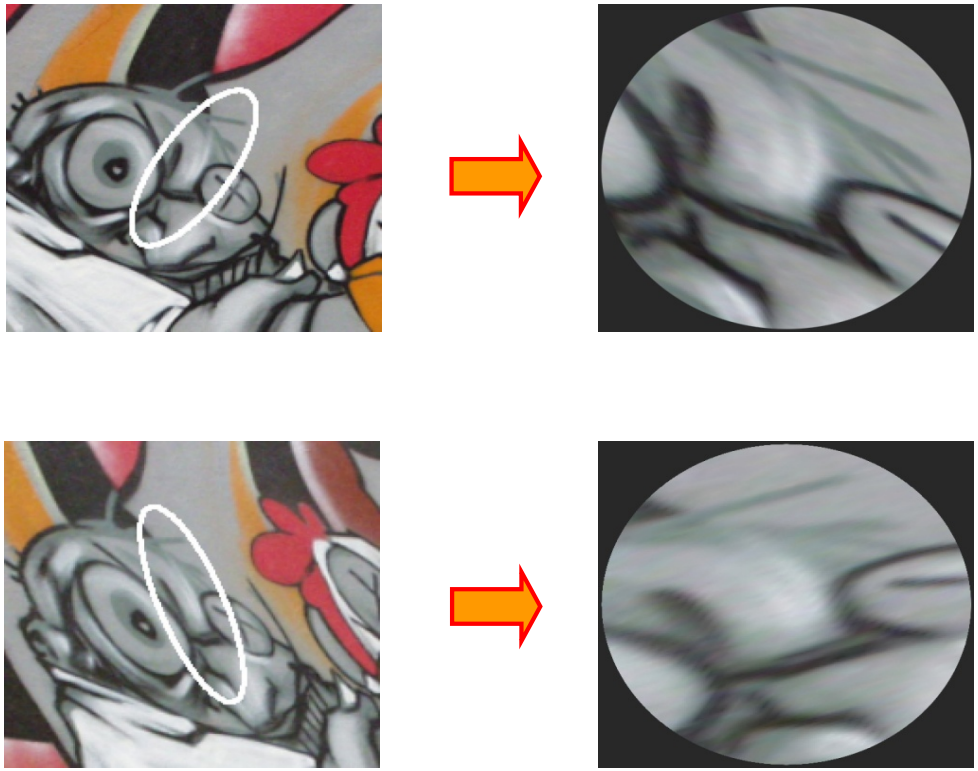
From covariant detection to invariant description

- How to compare the appearance of these image regions?
  - *Normalization*: transform these regions into same-size circles



Affine normalization

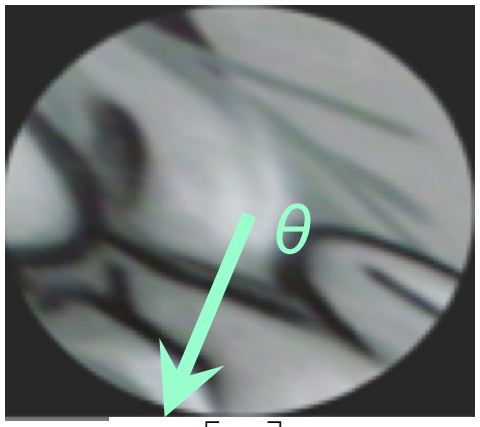
Problem: There is no unique transformation from an ellipse to a unit circle. (rotated or flipped versions remain a circle)

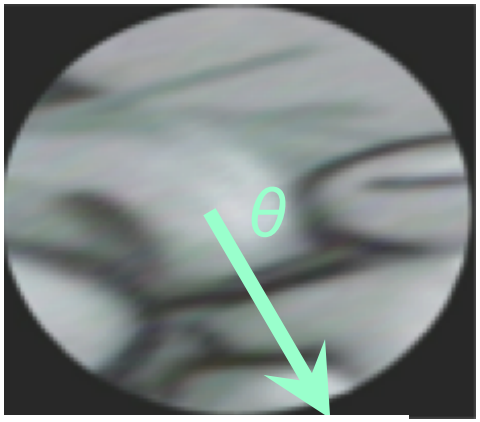


- 5-min Break

Eliminating rotation ambiguity

Assign unique orientation to image patch using gradient orientation

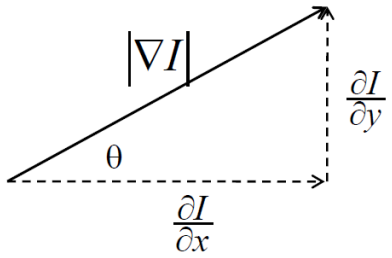




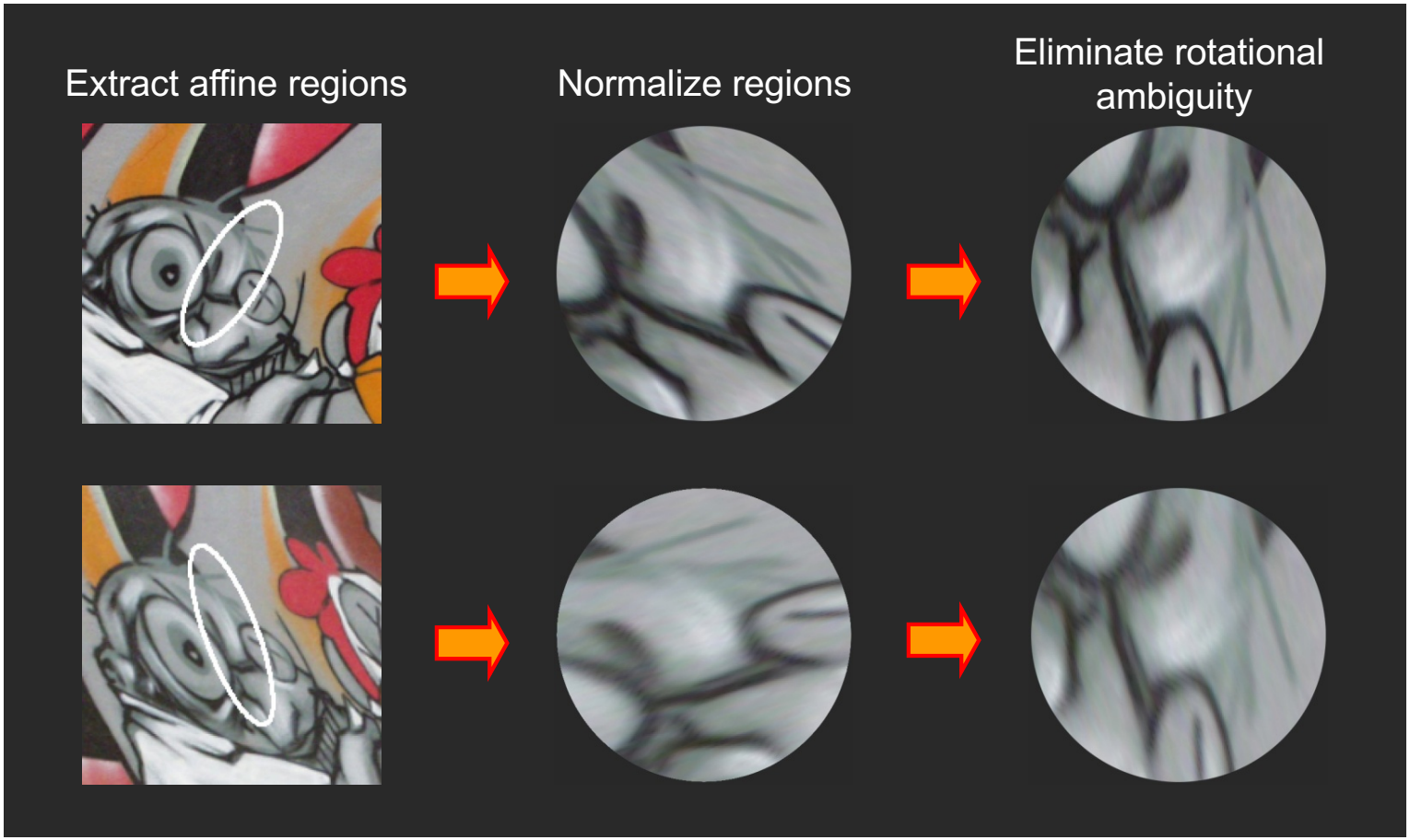
$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \approx \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$$

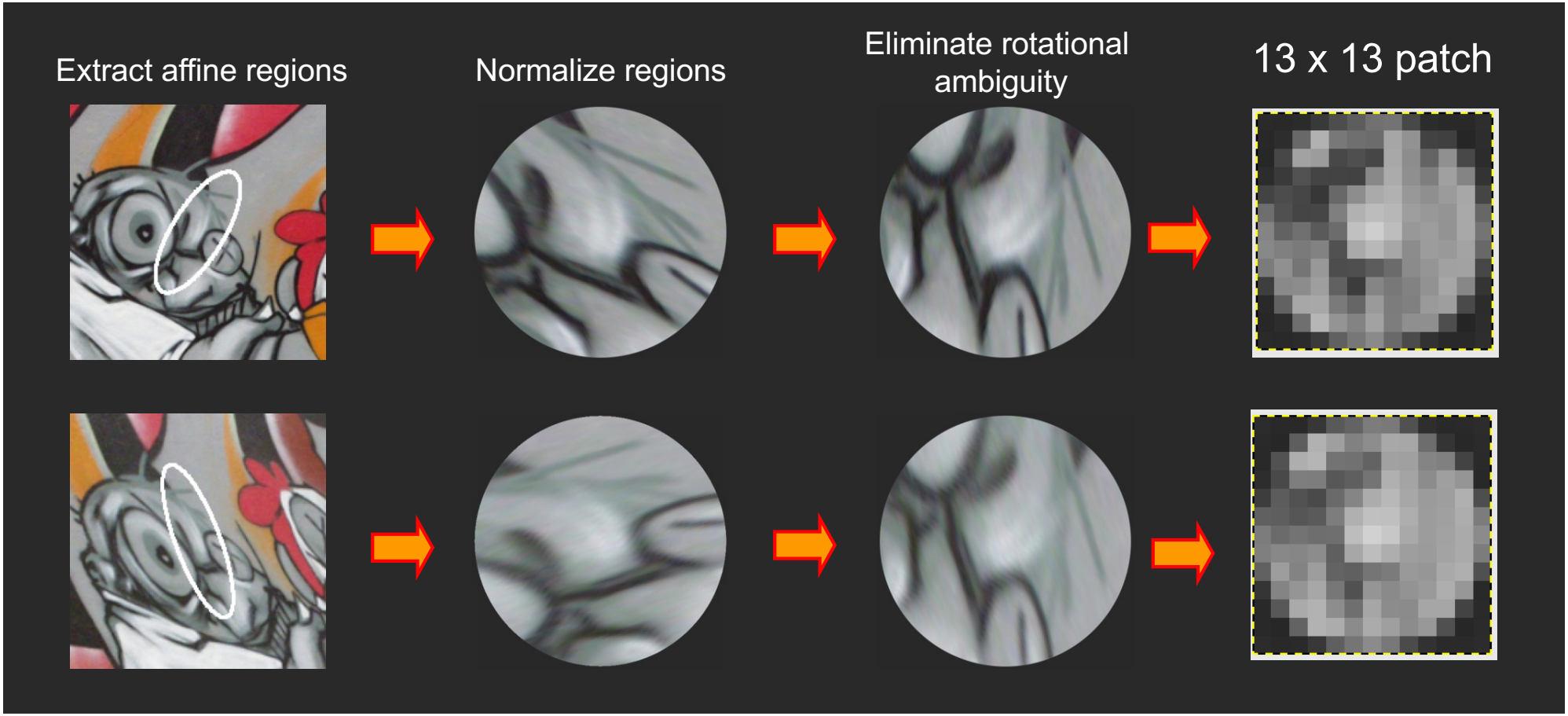
$$\theta = \tan^{-1} \frac{\left(\frac{\partial I}{\partial y}\right)}{\left(\frac{\partial I}{\partial x}\right)}$$



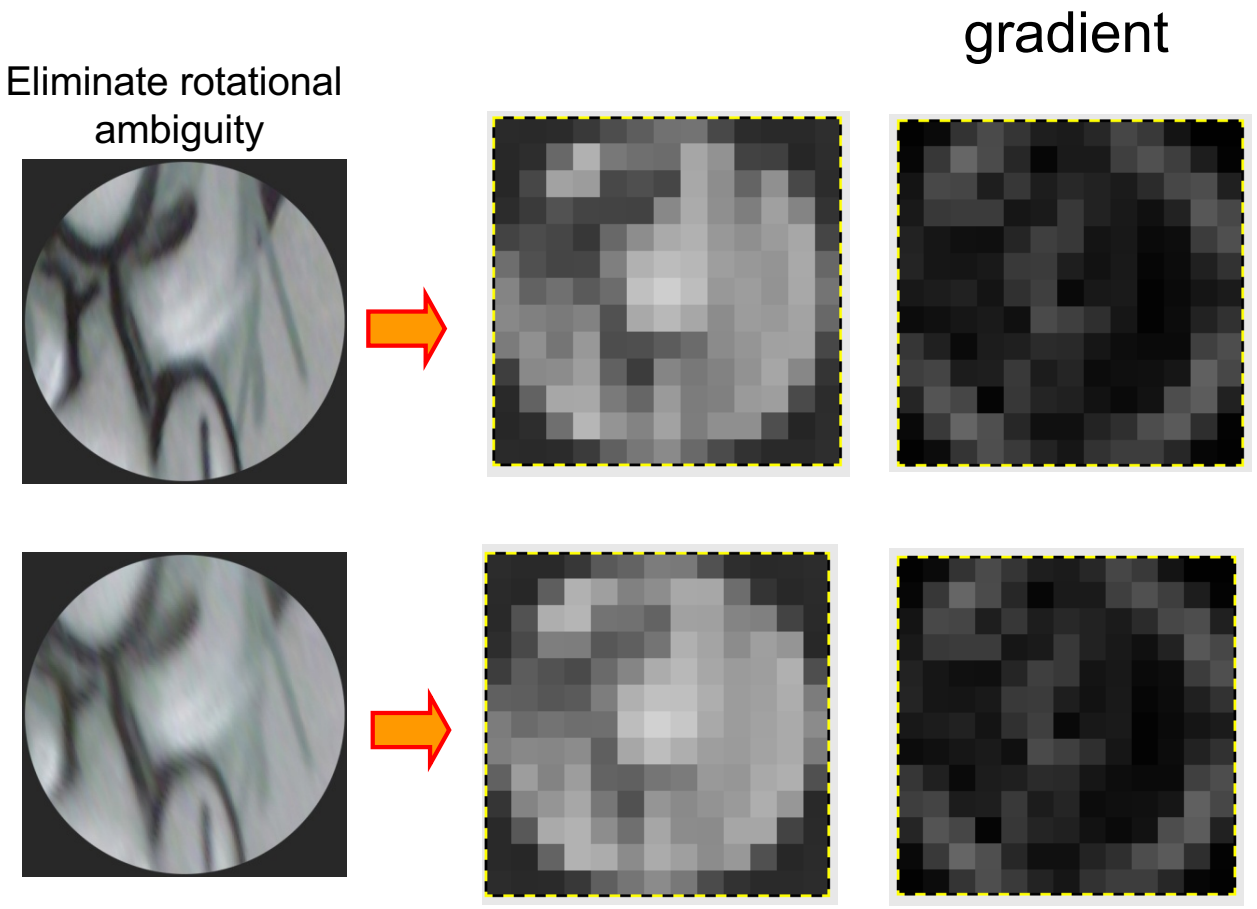
From covariant regions to invariant features



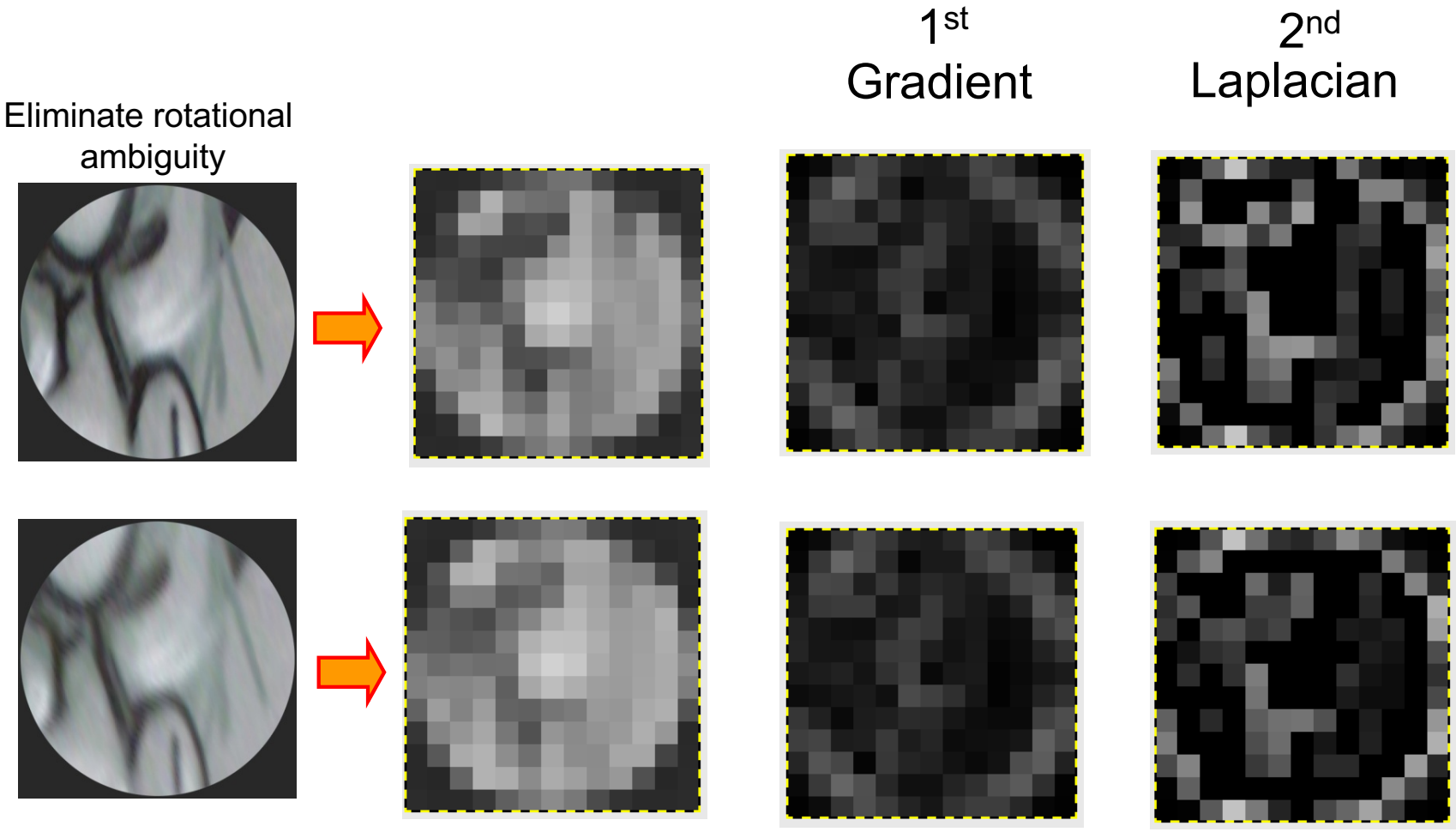
Local Descriptor <option 1>: Raw pixels



Local Descriptor <option 2>: Derivatives

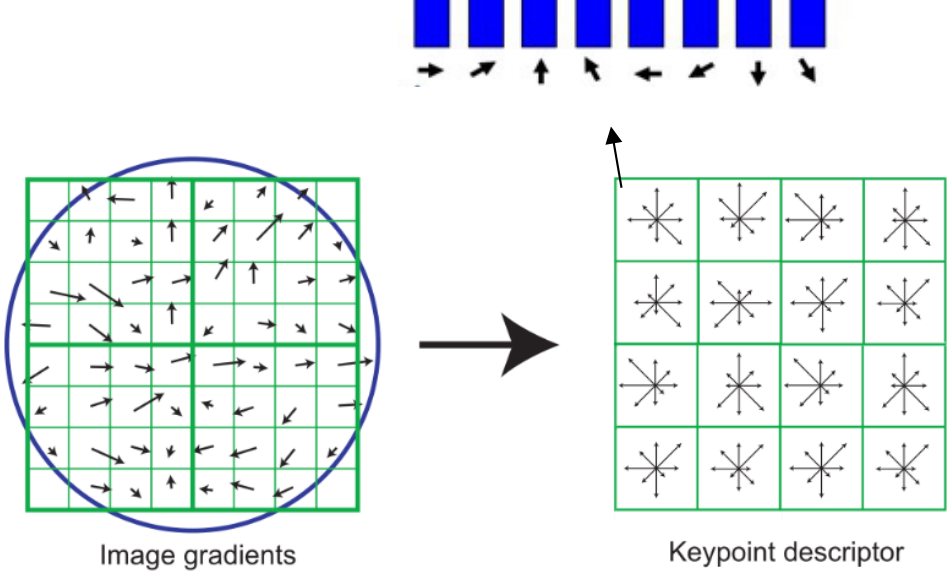


Local Descriptor <option 2>: Derivatives



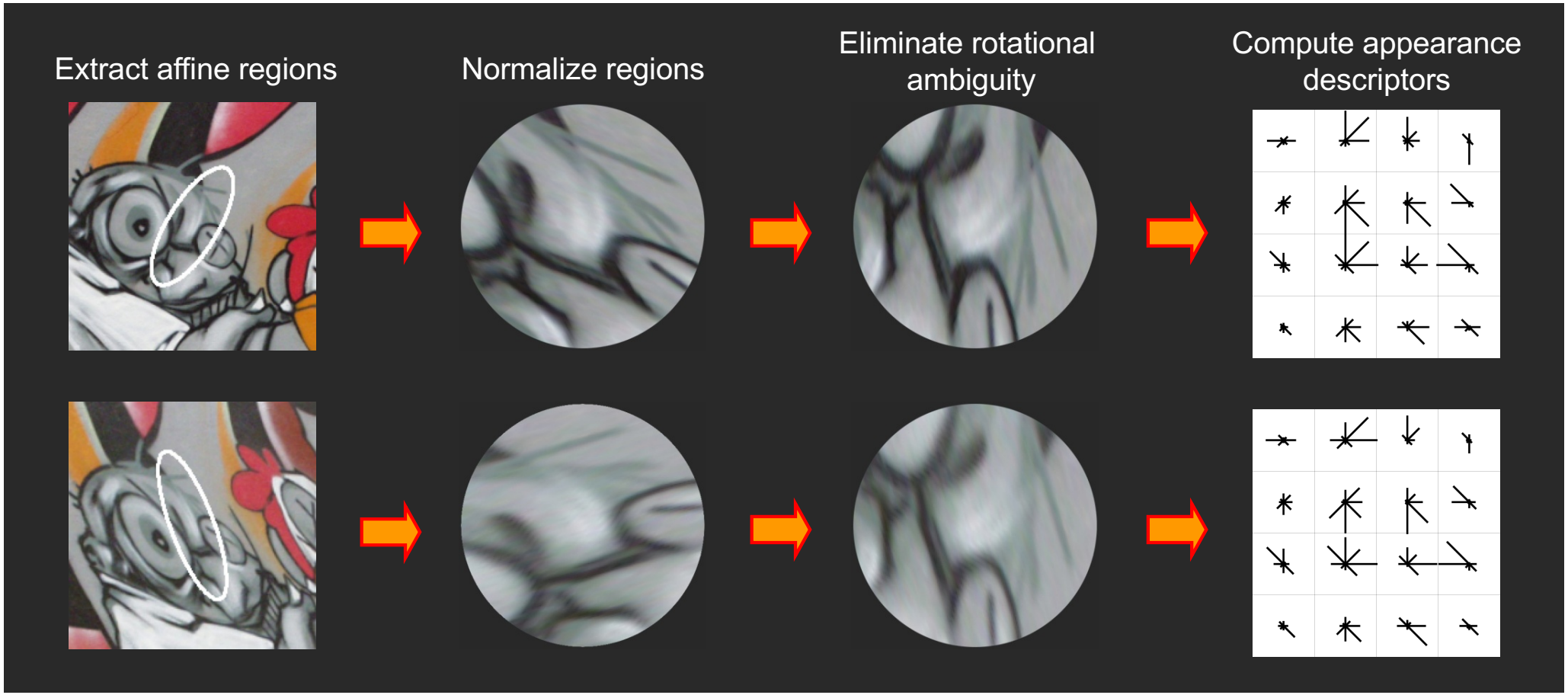
Descriptors: <option 3>: Local Distributions of Derivatives

- SIFT feature descriptor
  - 128 dimension: 4x4 quadrant, 8 bin histogram

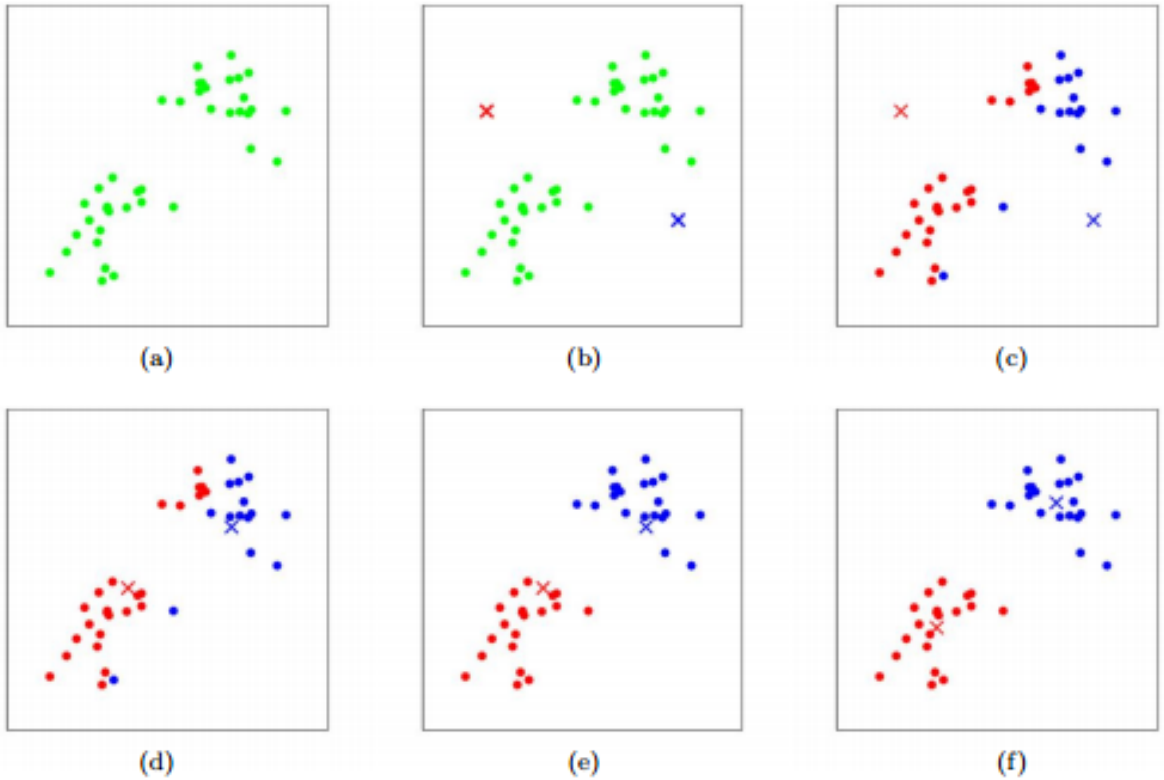


From covariant regions to invariant features

SIFT (Lowe ’04)



K-means clustering



K-means algorithm.

Training examples are shown as dots, and cluster centroids are shown as crosses. (a) Original dataset. (b) Random initial cluster centroids. (c-f) Illustration of running two iterations of k-means. In each iteration, we assign each training example to the closest cluster centroid; then we move each cluster centroid to the mean of the points assigned to it.

K-means clustering

In the clustering problem, we are given a training set $x^{(1)}, \ldots, x^{(m)}$, and want to group the data into a few cohesive "clusters." Here, we are given feature vectors for each data point $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ (making this an unsupervised learning problem). Our goal is to predict k centroids **and** a label $c^{(i)}$ for each datapoint. The k-means clustering algorithm is as follows:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set

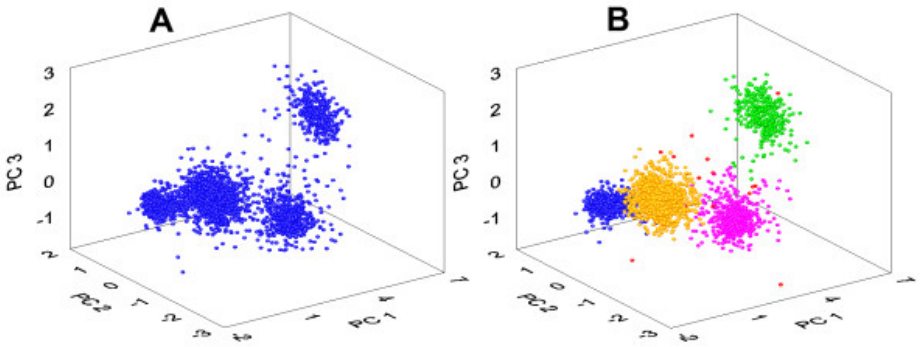
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

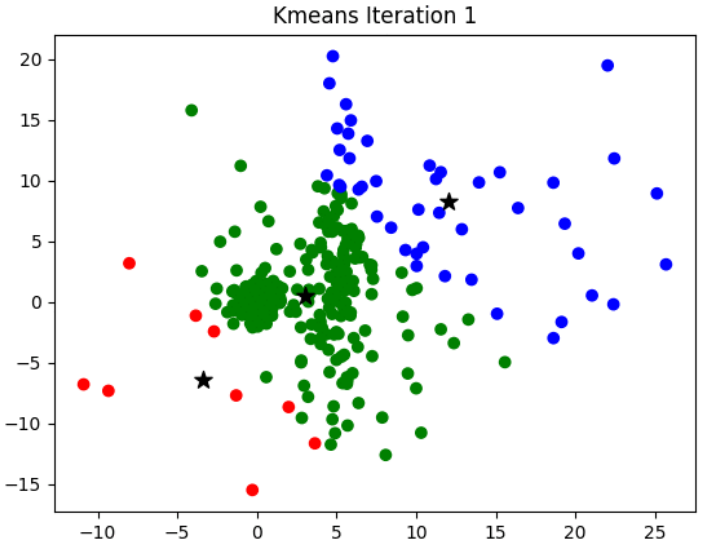
   For each j, set

   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

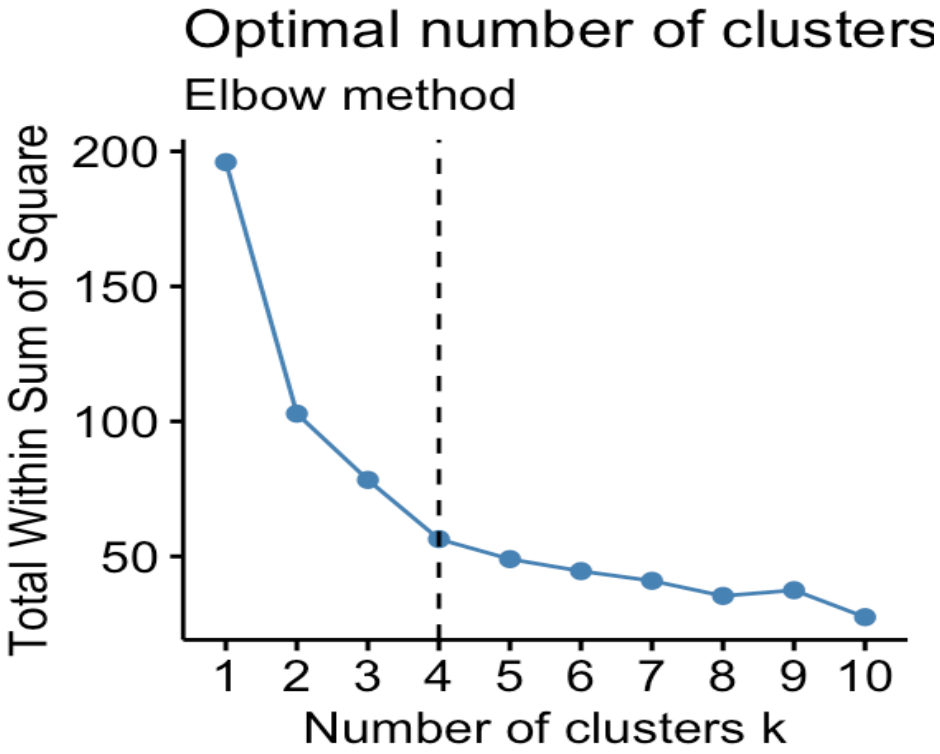
   }

K-means clustering

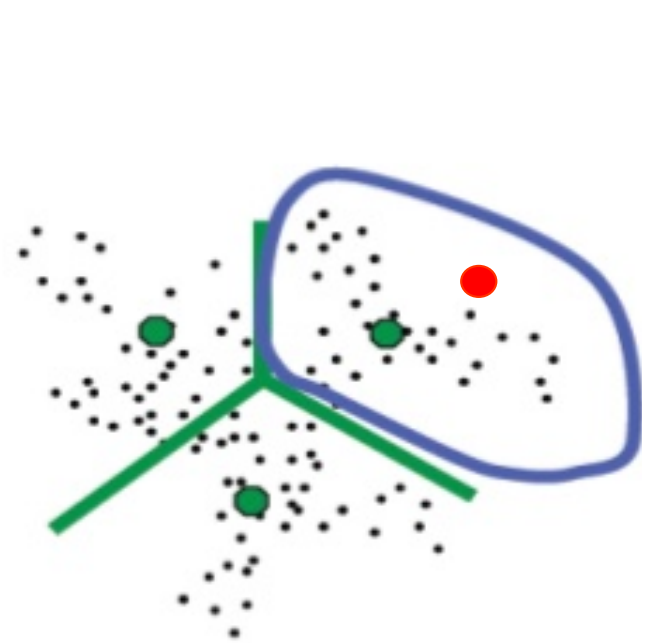


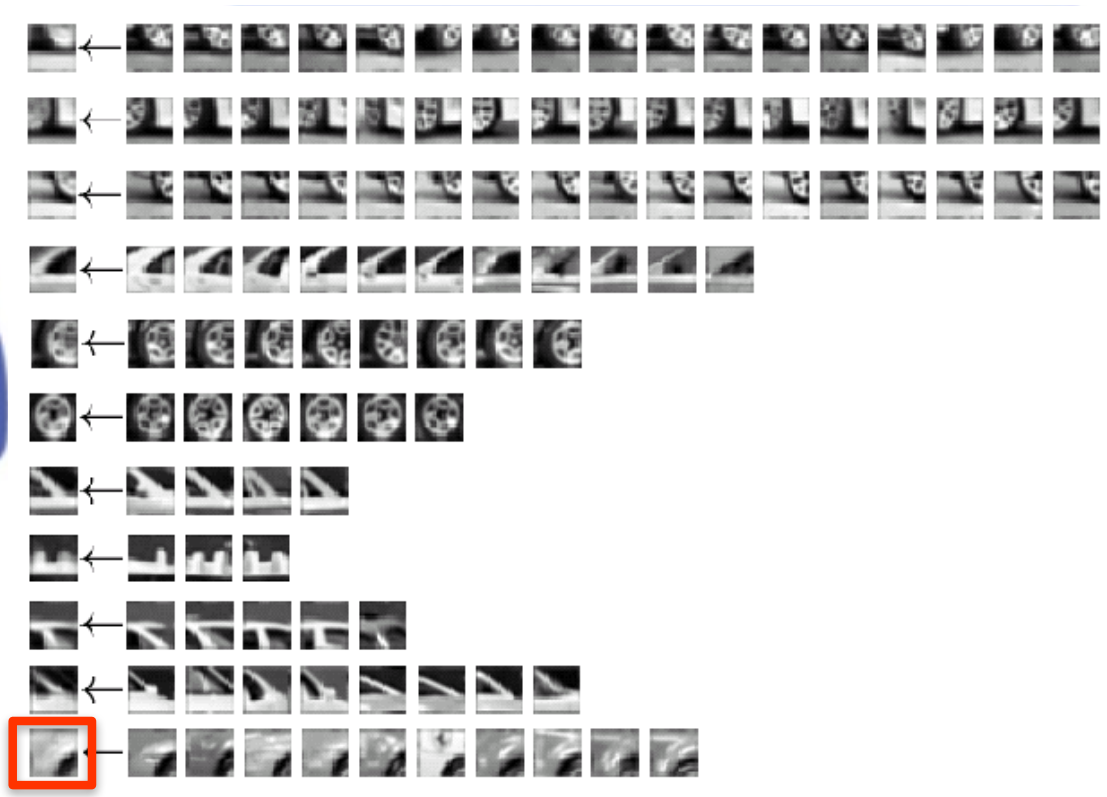


How many clusters?

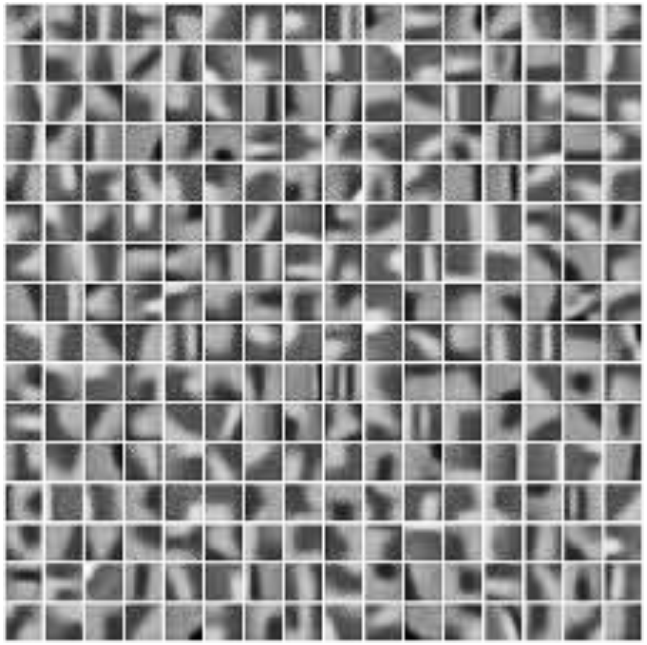


Visual words per object category





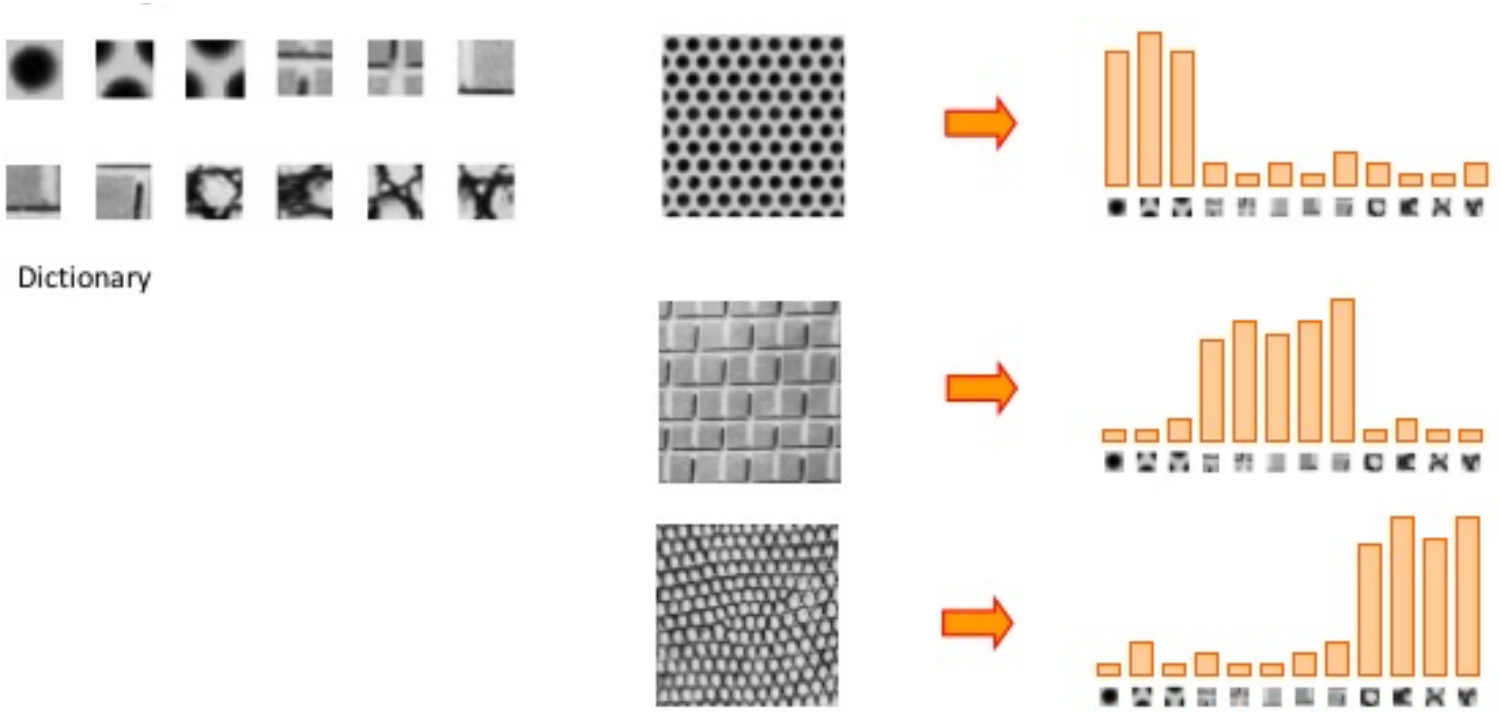
Example of learned visual dictionary



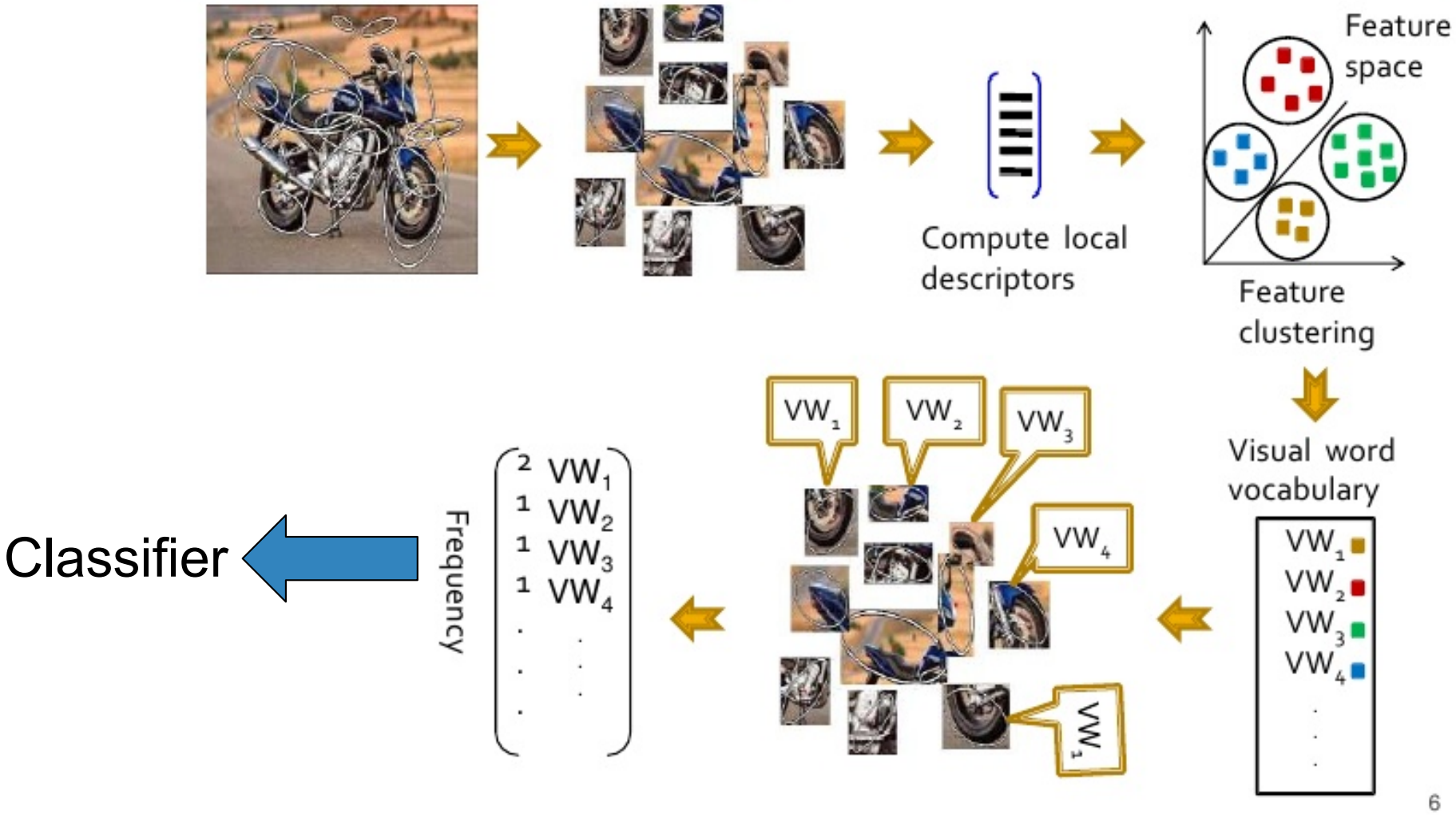
“Bag of words” classification

Object → **Bag of ‘words’**

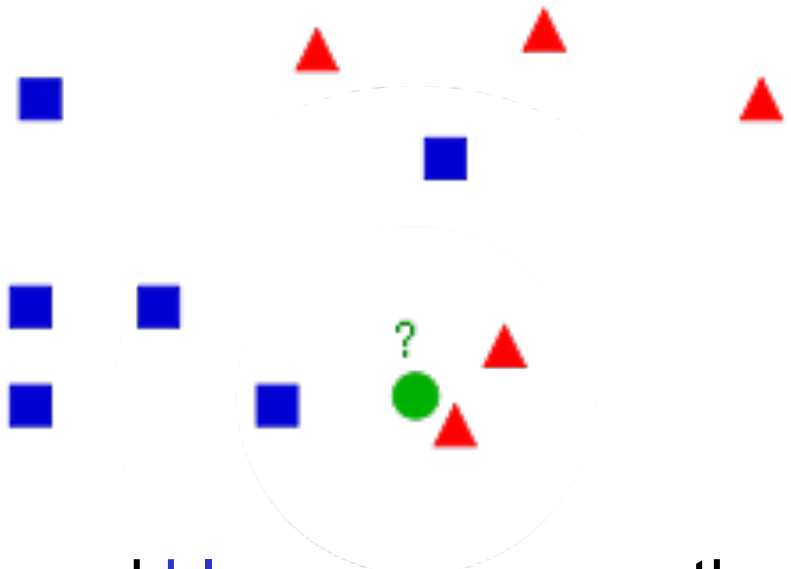
Bag of words



“Bag of words” classification



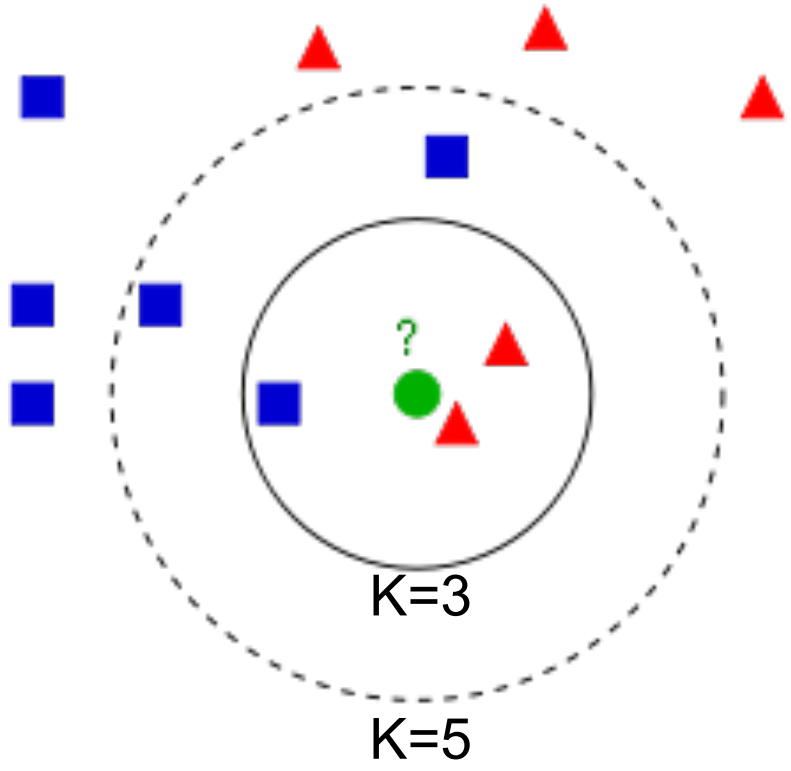
K-nearest Neighbor classifier



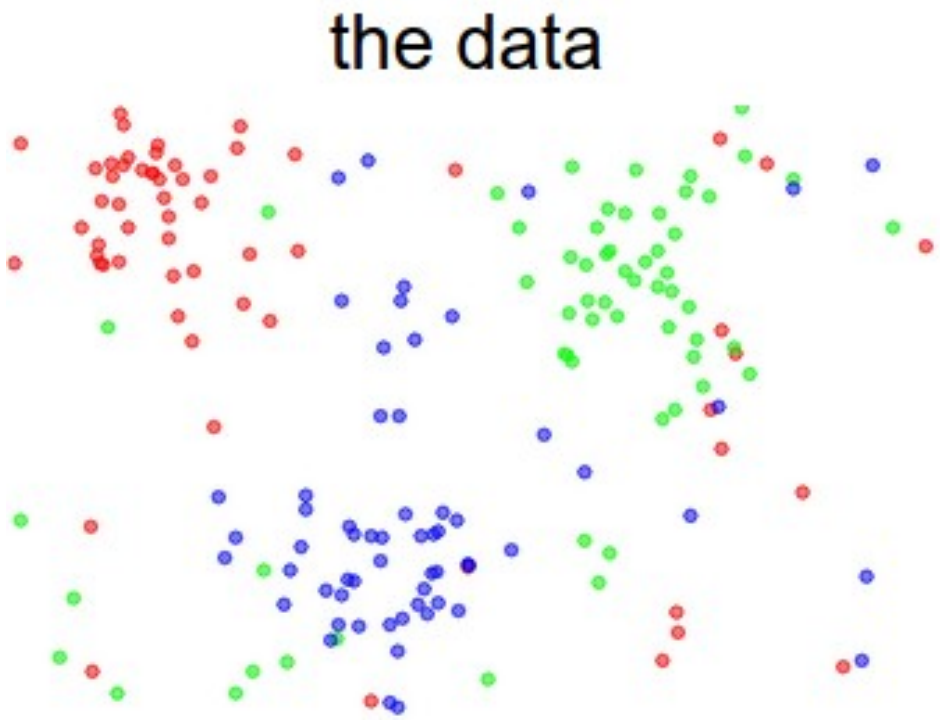
Red triangles and blue squares are the points in the training dataset.

To which class should ● (test point) be assigned to?

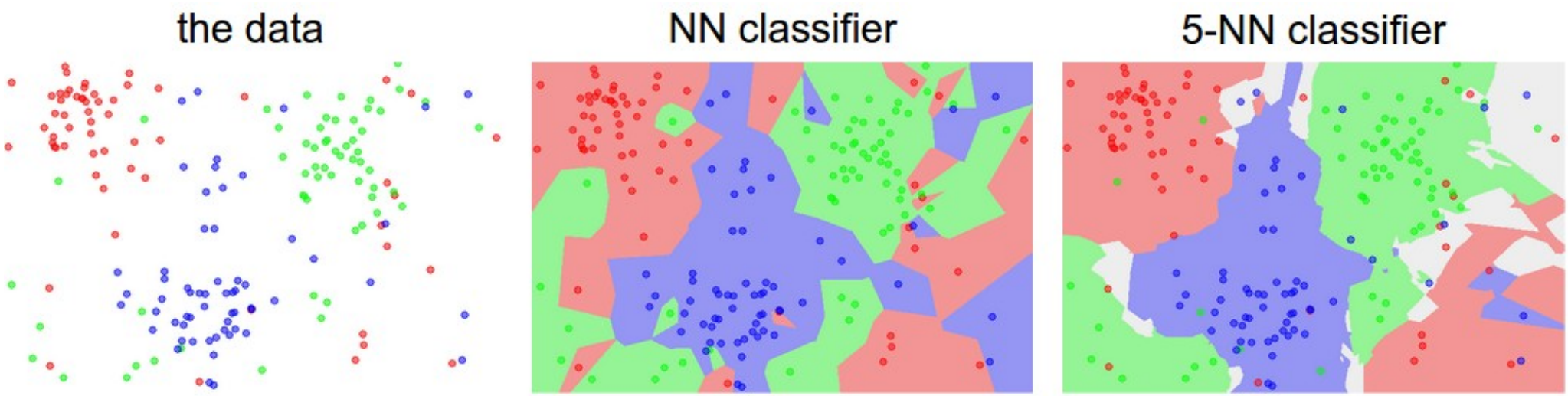
K-nearest Neighbor classifier



K-nearest Neighbor classifier

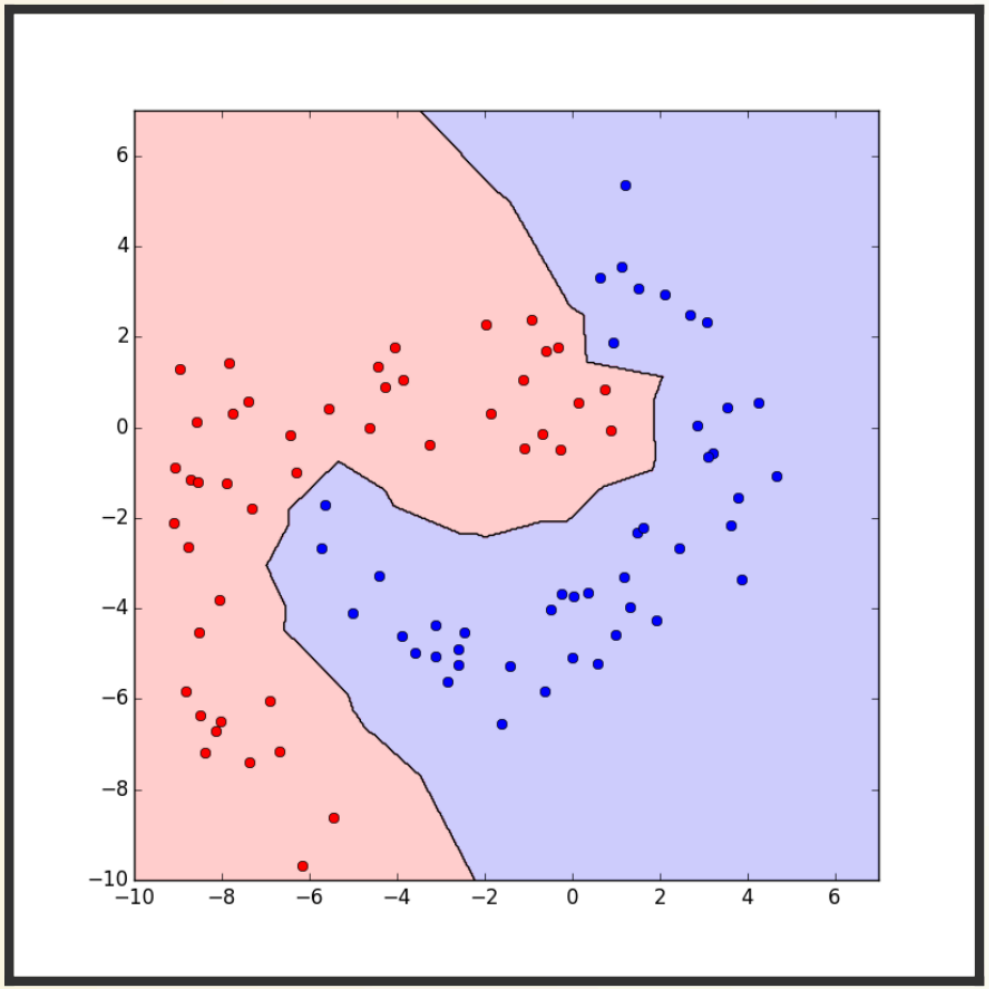


K-nearest Neighbor classifier



K-nearest Neighbor classifier

K=1



K=25