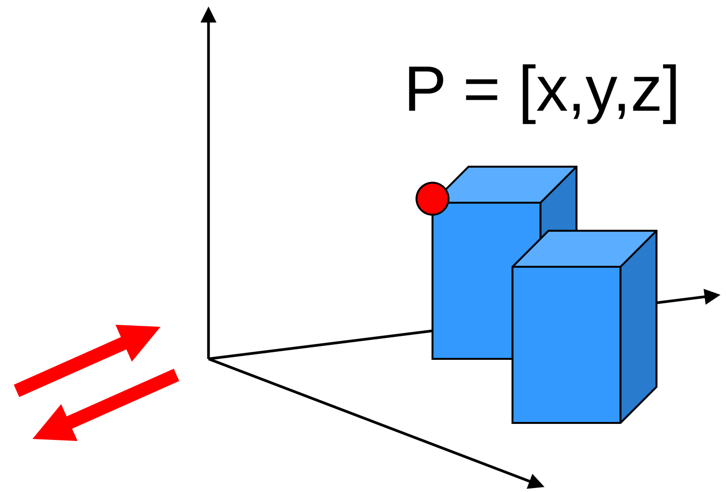
Lecture 10

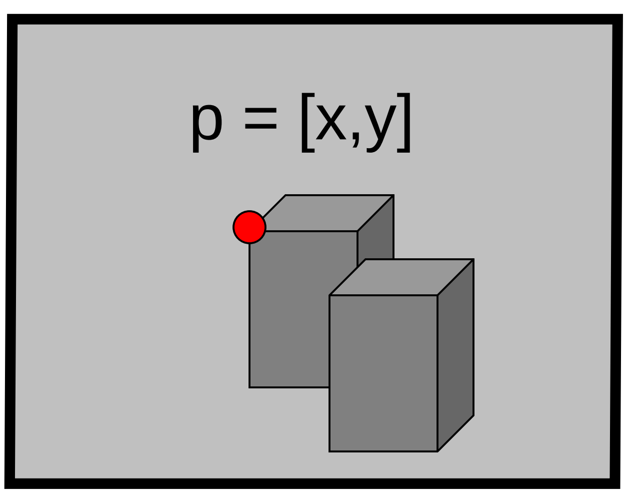
Detectors and descriptors

- Properties of detectors
  - Edge detectors
  - Harris
  - DoG
- Properties of detectors
  - SIFT
  - Shape context

From the 3D to 2D & vice versa



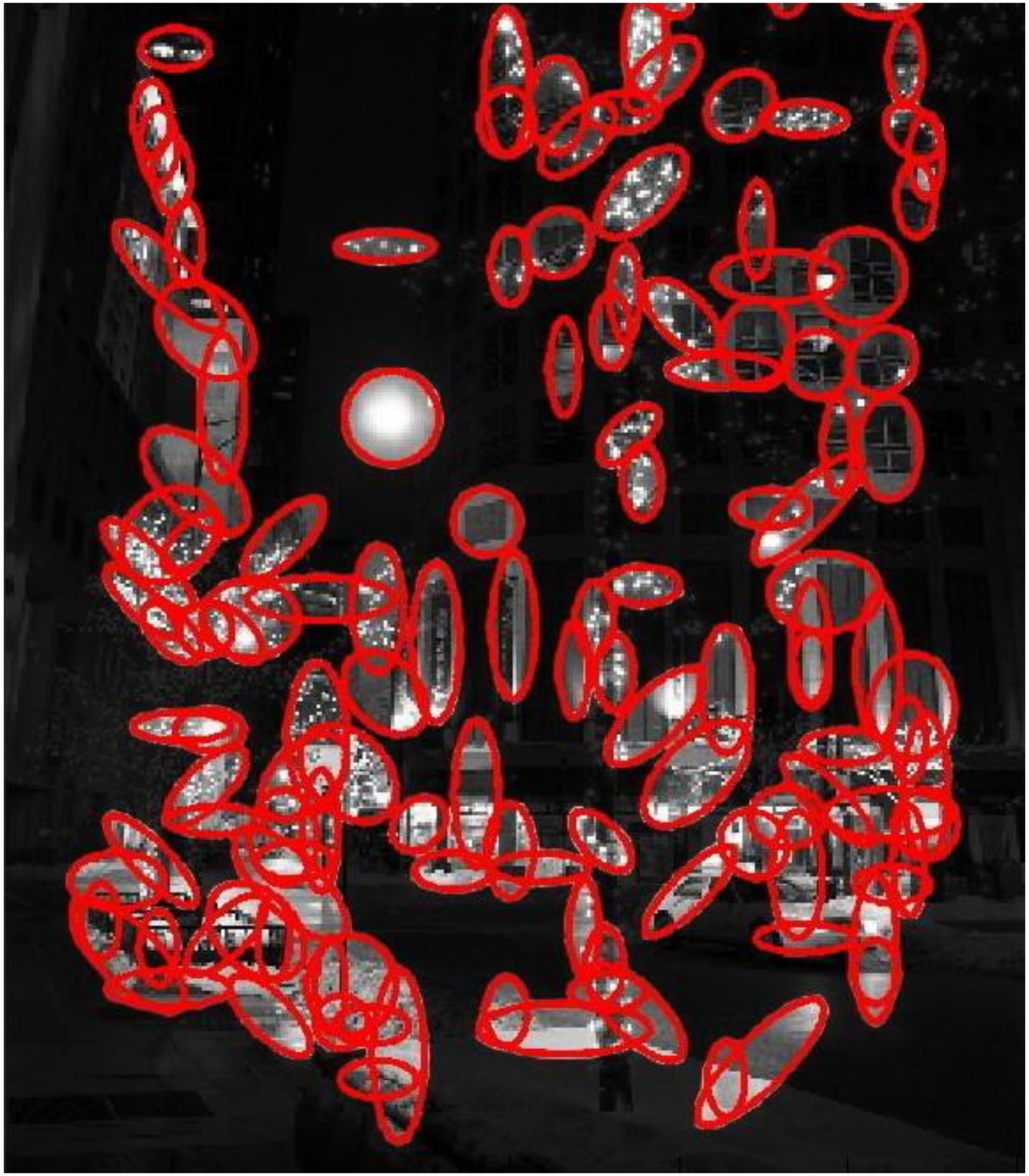
3D world



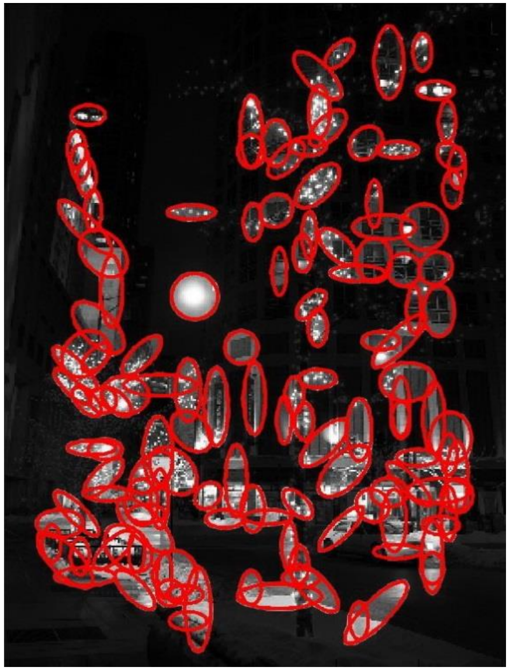
Image

- Let’s now focus on 2D

How to represent images?



The big picture…

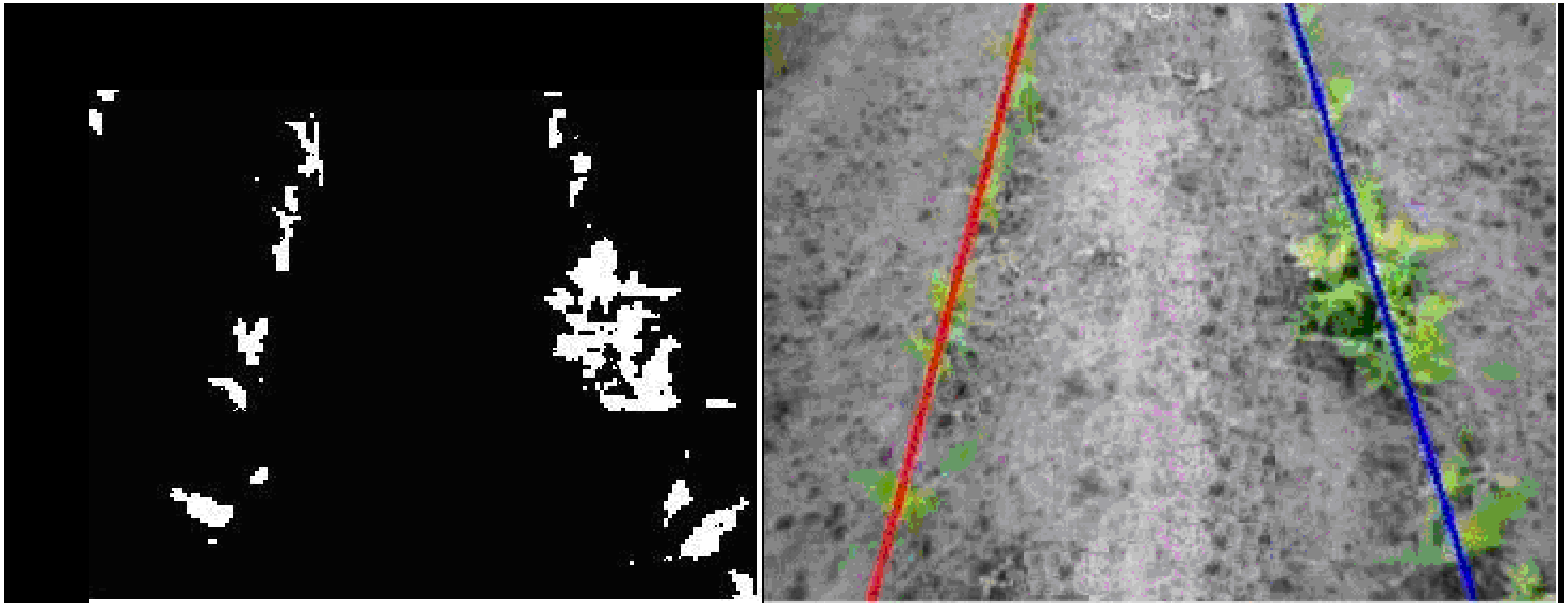


Feature Detection — e.g. DoG

Feature Description — e.g. SIFT

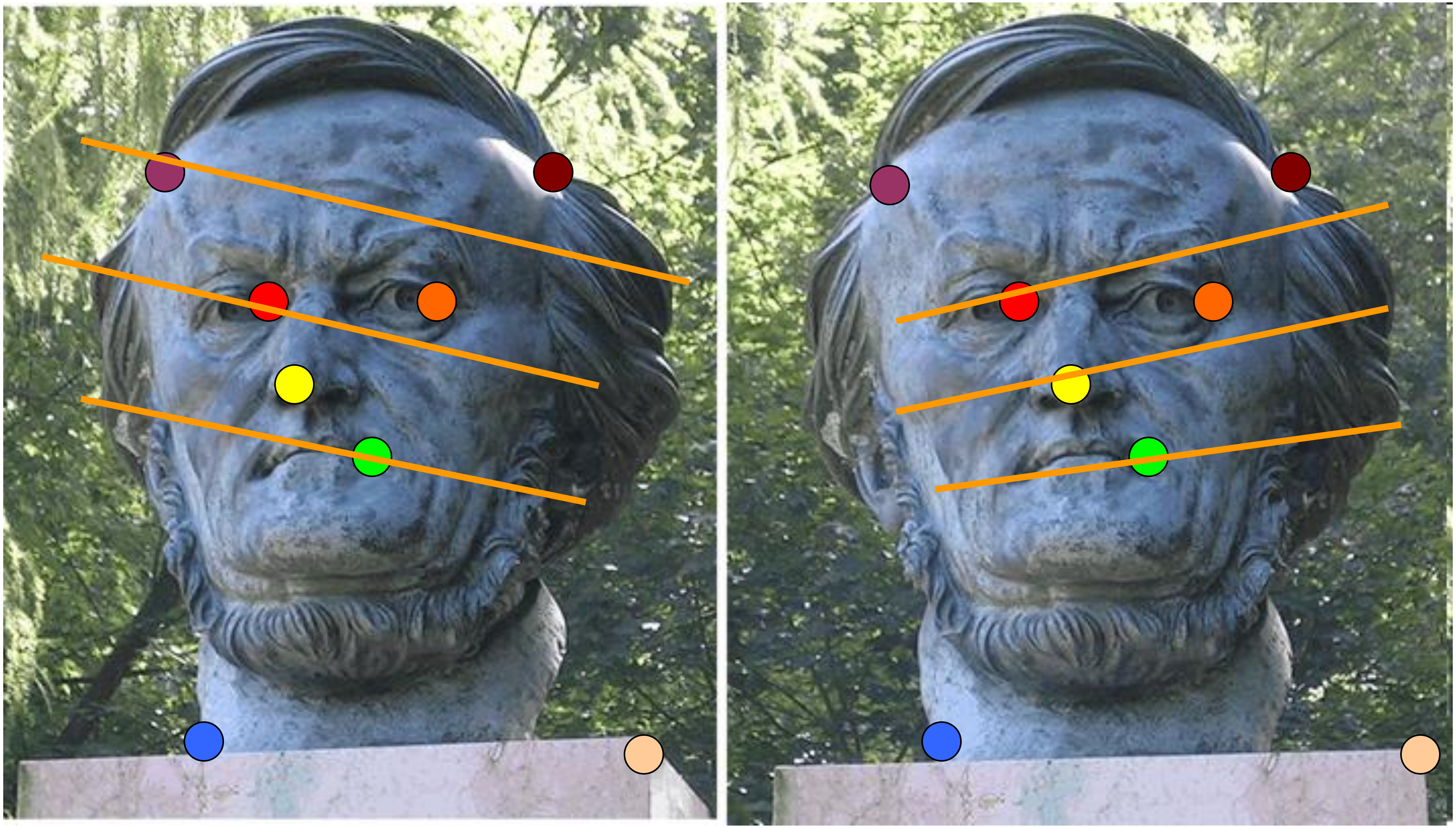
- Estimation
- Matching
- Indexing
- Detection

Estimation

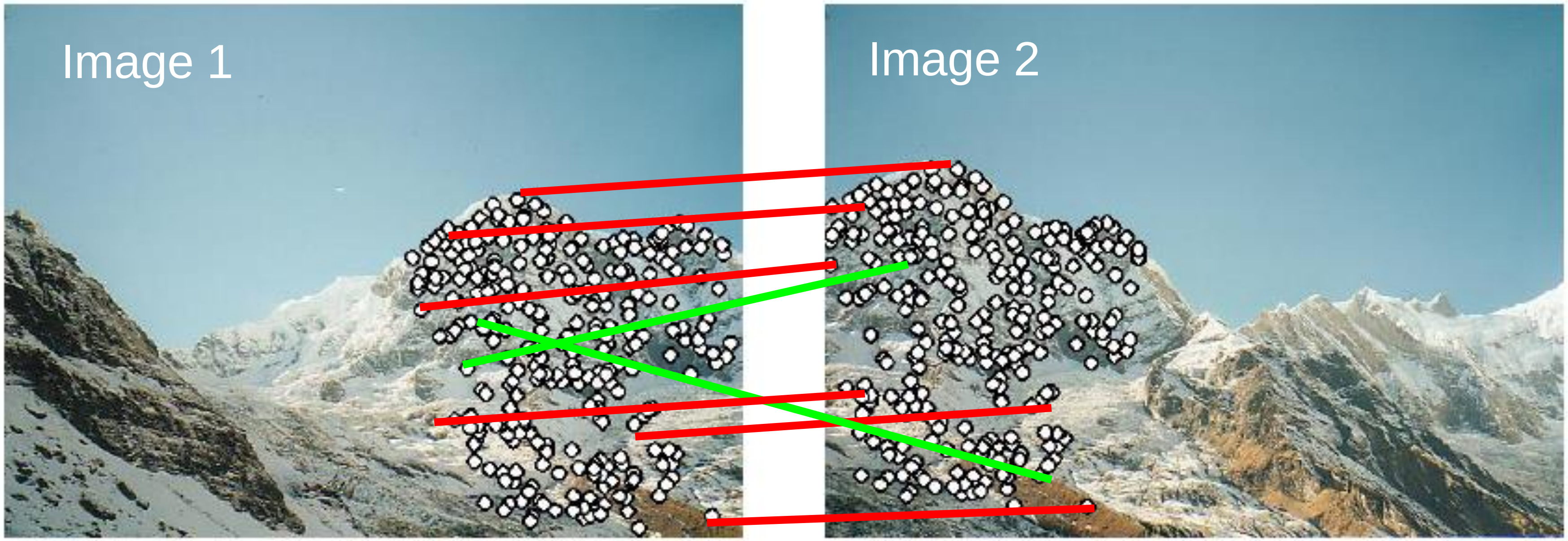


Courtesy of TKK Automation Technology Laboratory

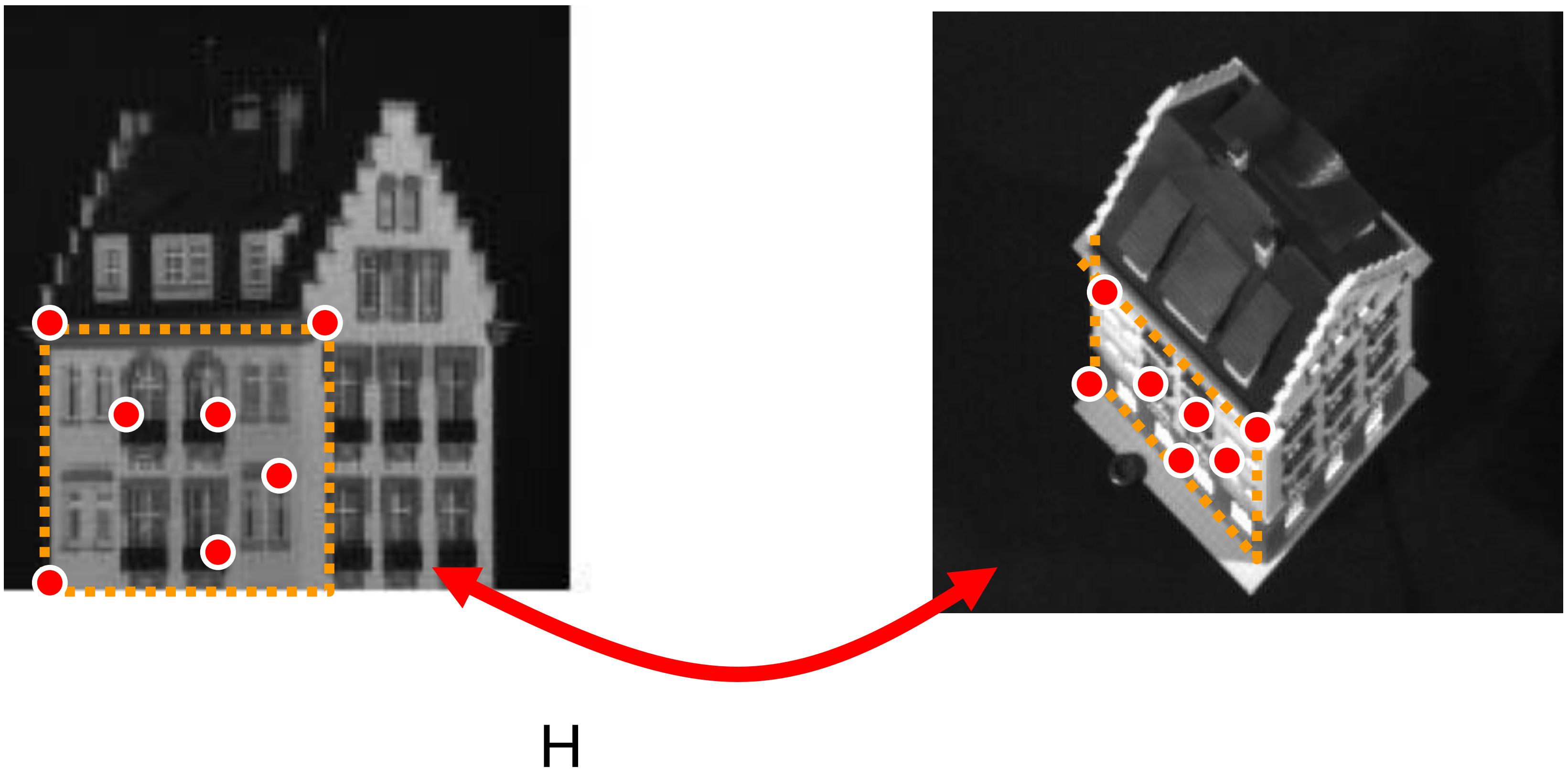
Estimation



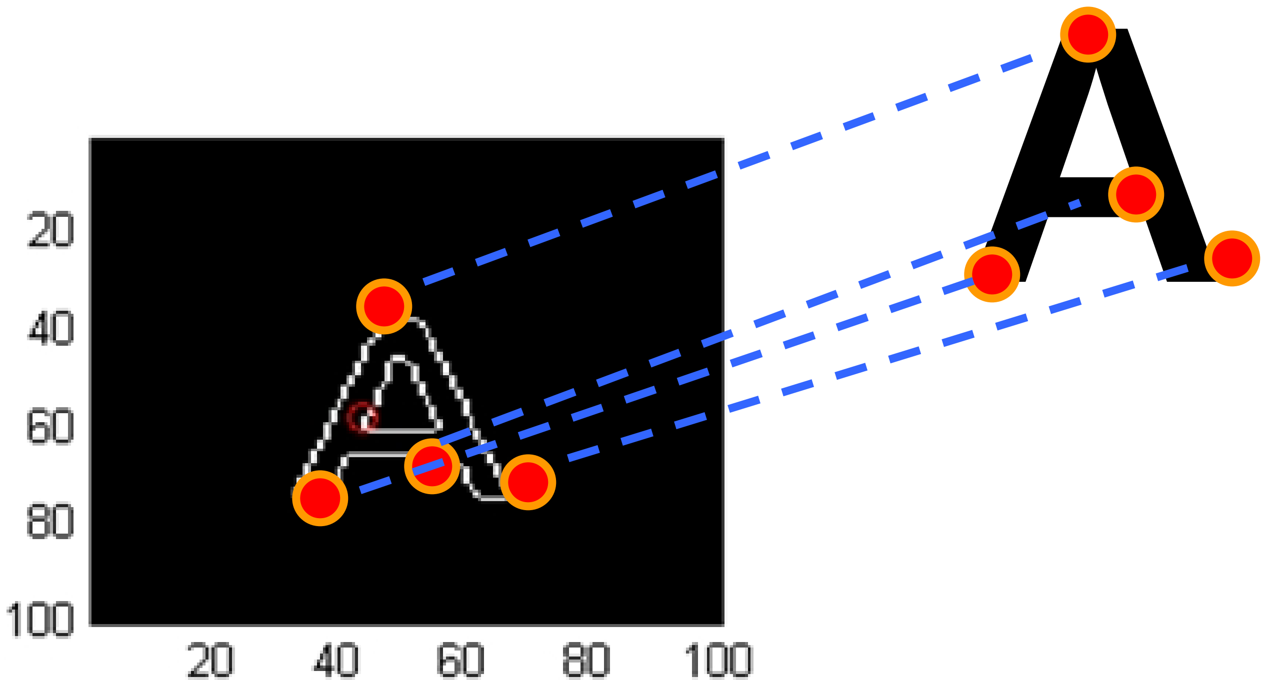
Matching



Matching



Object modeling and detection

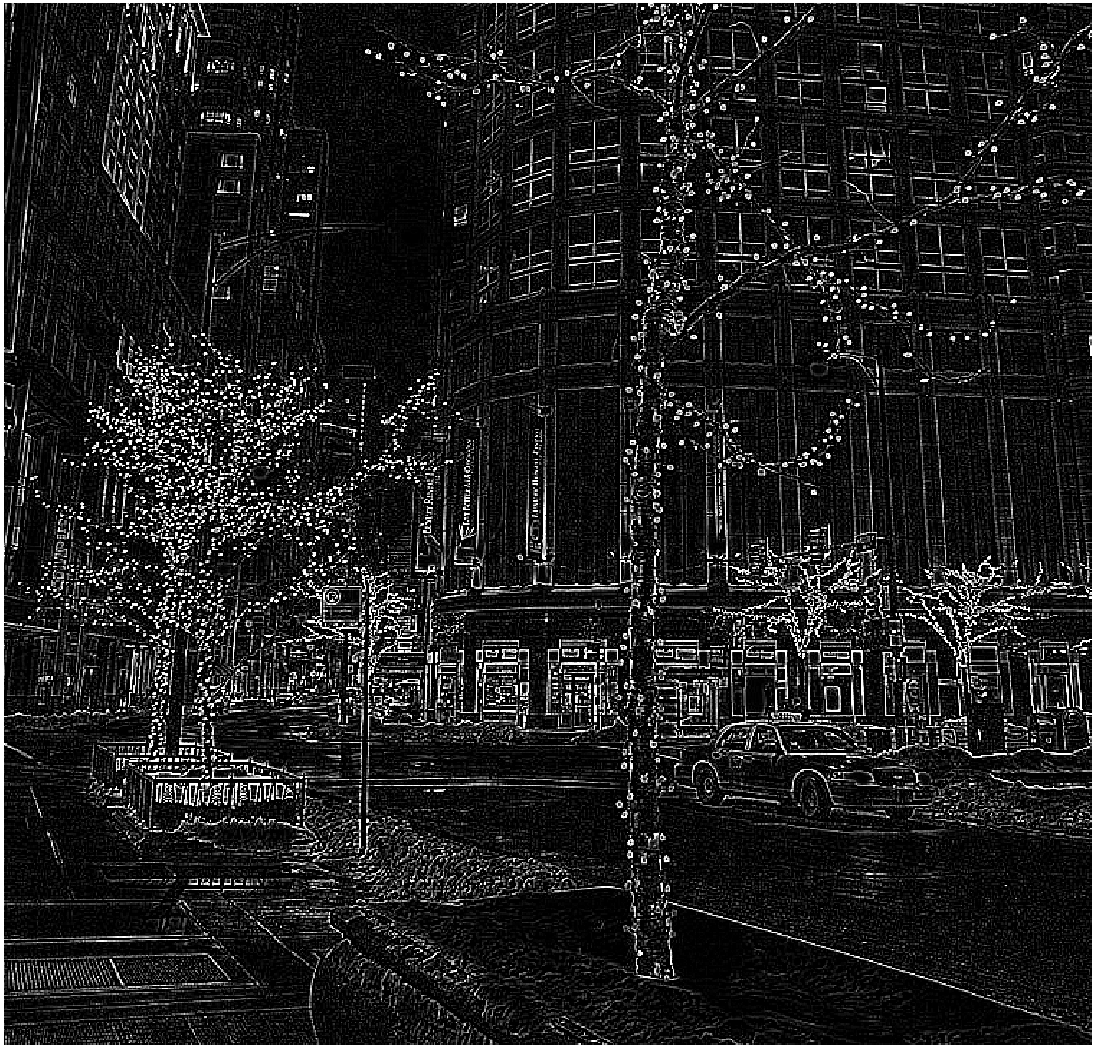


Lecture 10

Detectors and descriptors

- Properties of detectors
  - Edge detectors
  - Harris
  - DoG
- Properties of detectors
  - SIFT
  - Shape context

Edge detection



What causes an edge?

Identifies sudden changes in an image

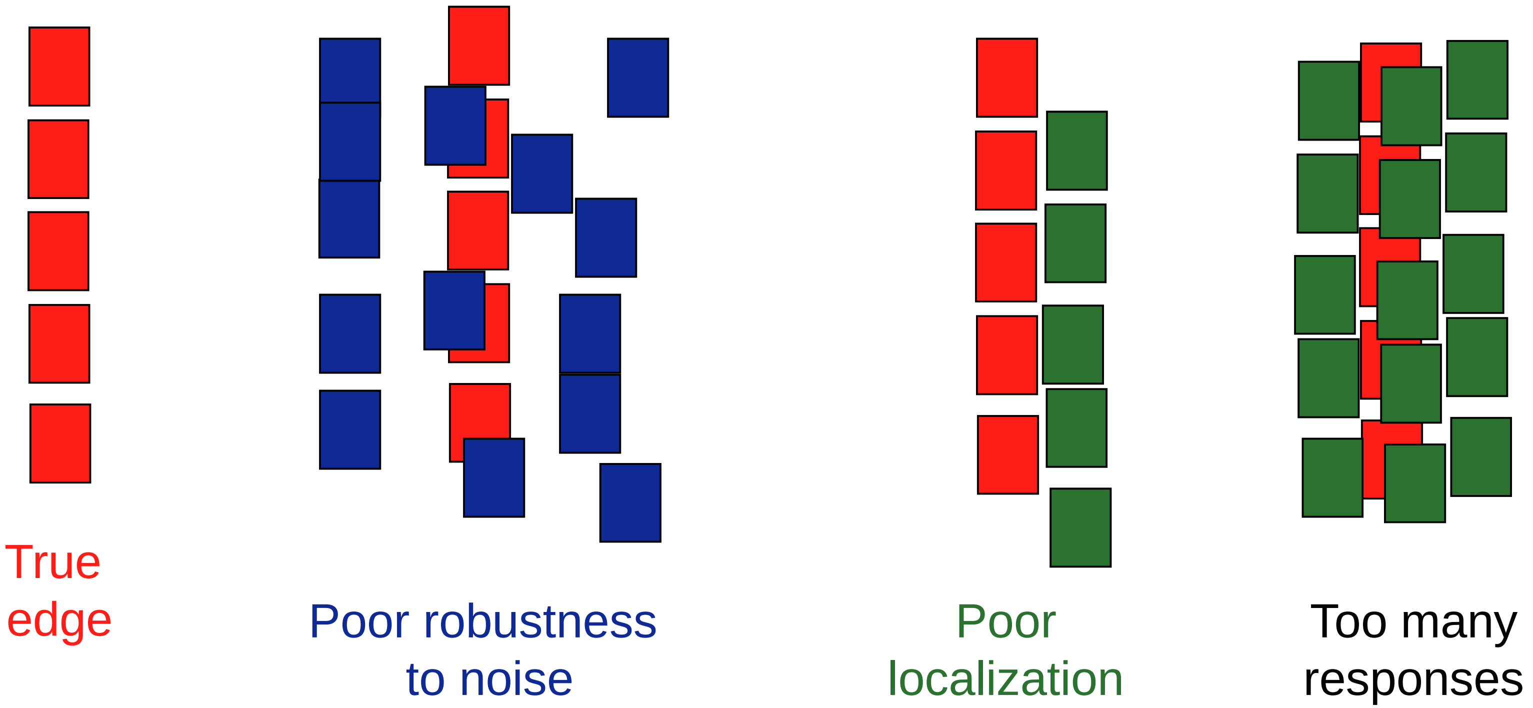
- Depth discontinuity
- Surface orientation discontinuity
- Reflectance discontinuity (i.e., change in surface material properties)
- Illumination discontinuity (e.g., highlights; shadows)

Edge Detection

- Criteria for optimal edge detection (Canny 86):
  - Good detection accuracy:
    - minimize the probability of false positives (detecting spurious edges caused by noise),
    - false negatives (missing real edges)
  - Good localization:
    - edges must be detected as close as possible to the true edges.
  - Single response constraint:
    - minimize the number of local maxima around the true edge (i.e. detector must return single point for each true edge point)

Edge Detection

- Examples:



Designing an edge detector

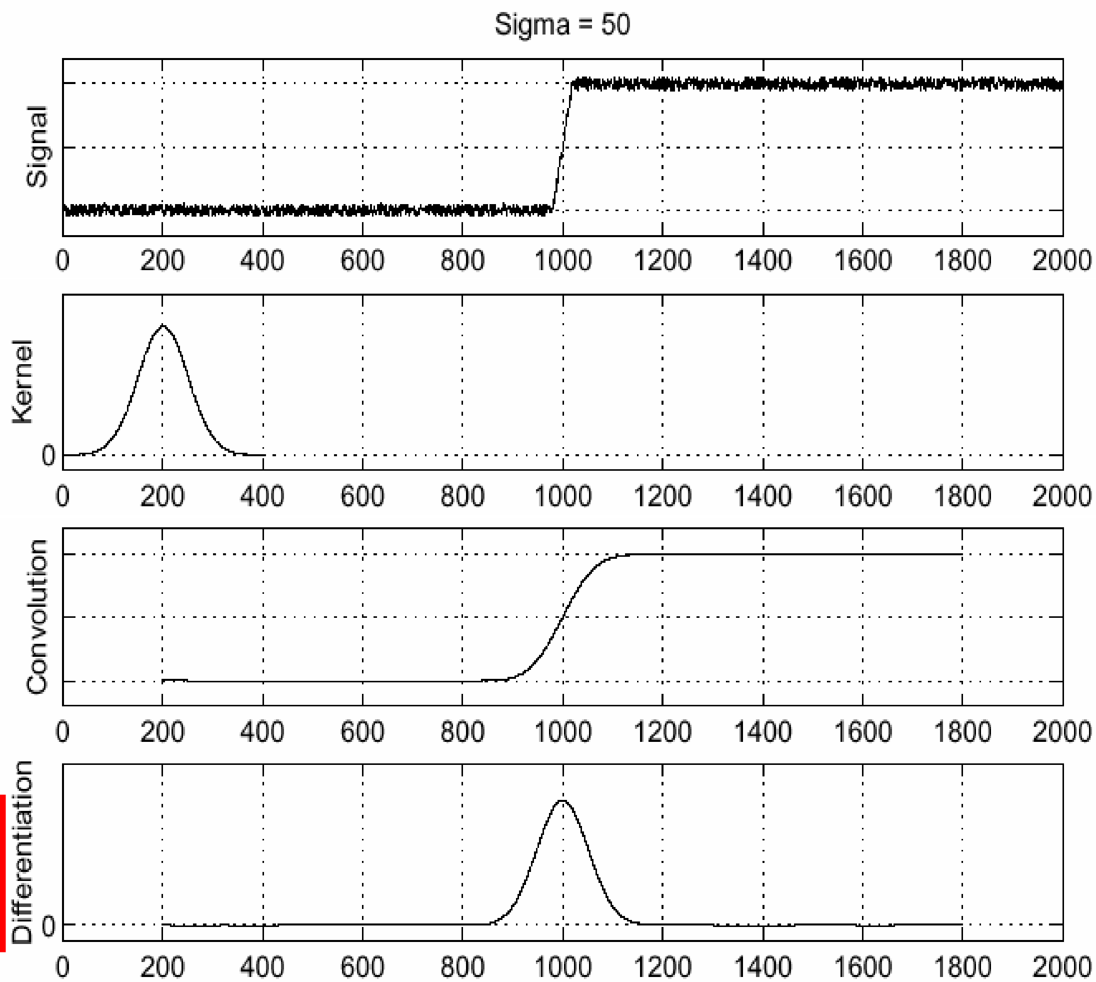
- **Two ingredients:**
- Use derivatives (in x and y direction) to define a location with high gradient .
- Need smoothing to reduce noise prior to taking derivative

Designing an edge detector

f

g

$f * g$



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

= (d g/ d x) * f = “derivative of Gaussian” filter

Source: S. Seitz

Edge detector in 2D

- Smoothing

$$I' = g(x, y) * I \qquad g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Derivative

$$S = \nabla(g * I) = (\nabla g) * I = \qquad \nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

$$= \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix} = \begin{bmatrix} S_x & S_y \end{bmatrix}$$ = gradient vector

Canny Edge Detection (Canny 86):

See CS131A for details

original

Canny with $\sigma = 1$

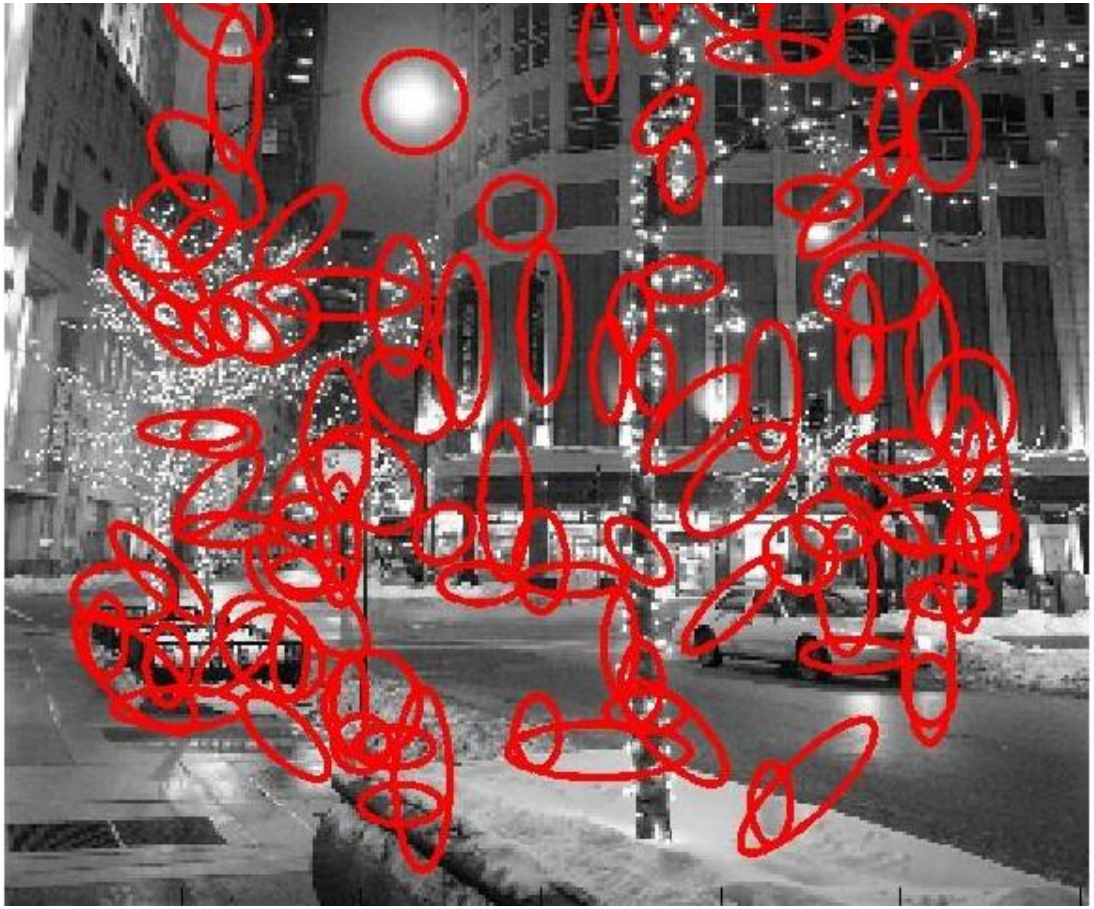
Canny with $\sigma = 2$

- The choice of σ depends on desired behavior
  - large σ detects large scale edges
  - small σ detects fine features

Other edge detectors:

- Sobel
- Canny-Deriche
- Differential

Corner/blob detectors



- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature is found at an “interesting” region of the image
- Locality
  - A feature occupies a “relatively small” area of the image;

Repeatability

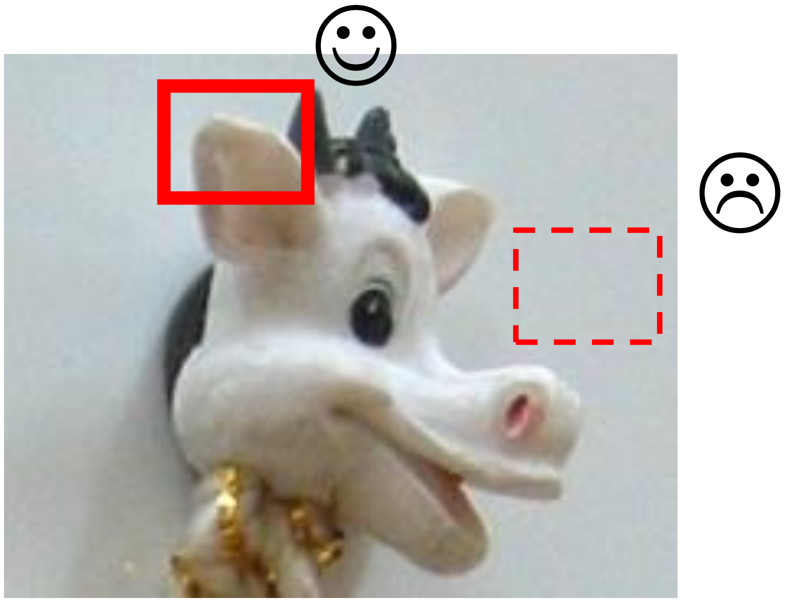
Illumination invariance

Scale invariance

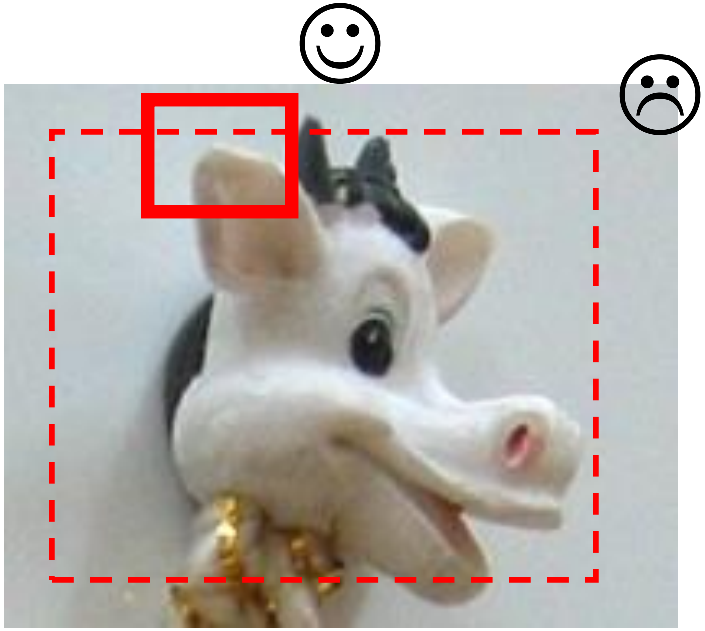
Pose invariance

- Rotation
- Affine

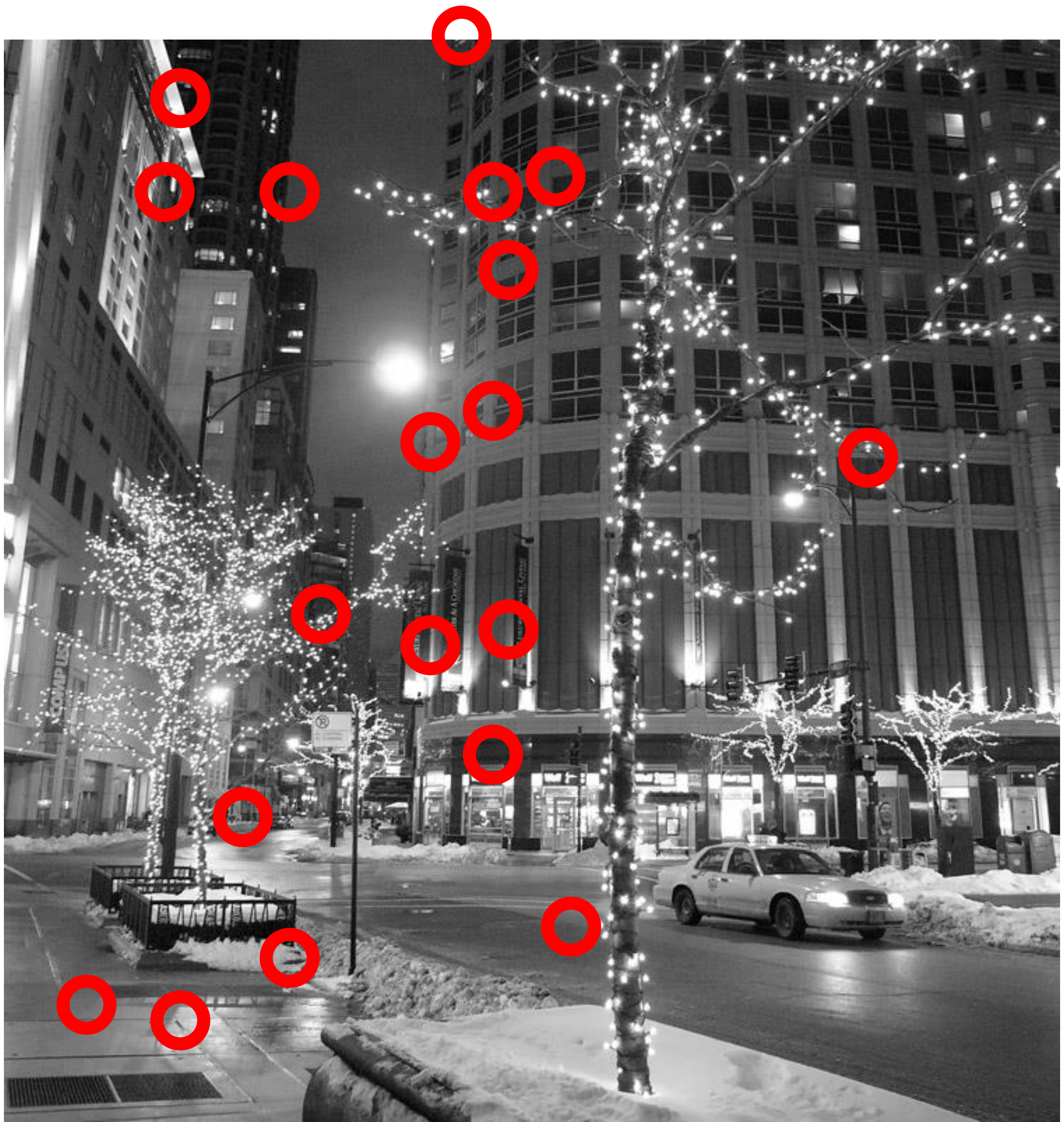
- Saliency



- Locality



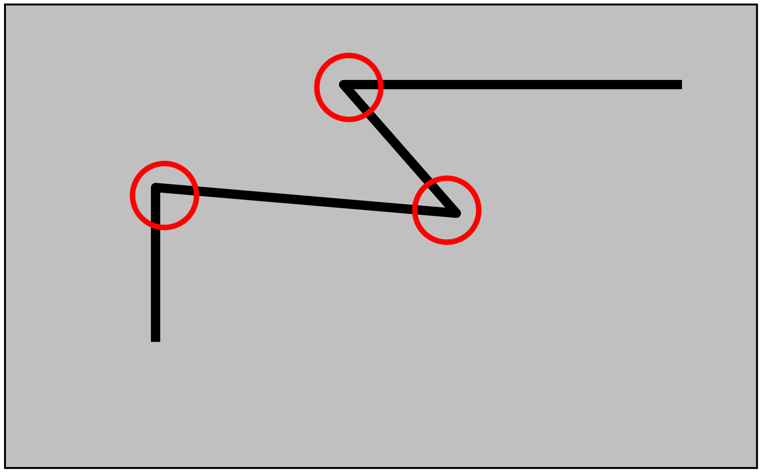
Corners detectors



Harris corner detector

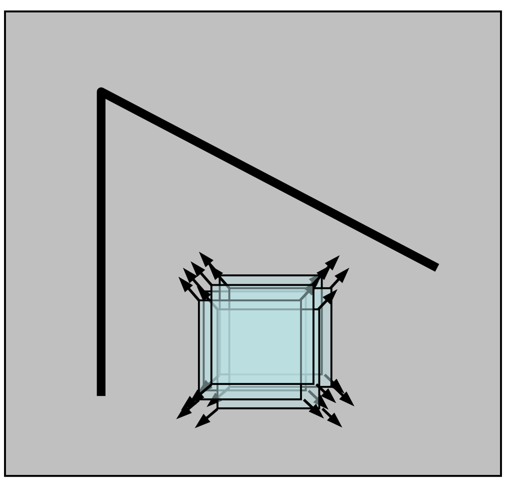
C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*: pages 147--151.

See CS131A for details

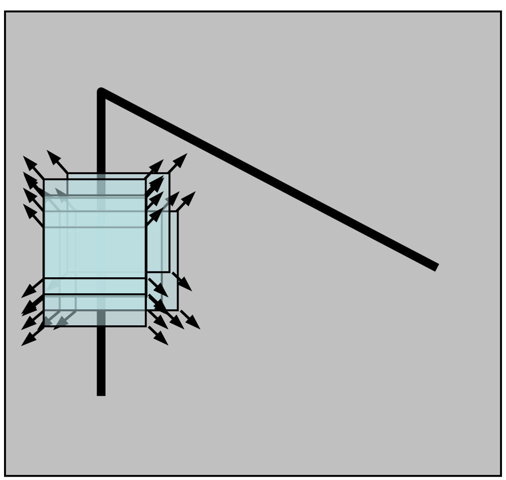


Harris Detector: Basic Idea

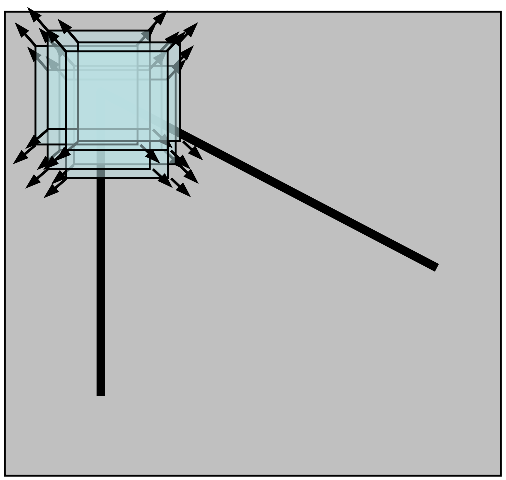
Explore intensity changes within a window
as the window changes location



"flat" region:
no change in
all directions



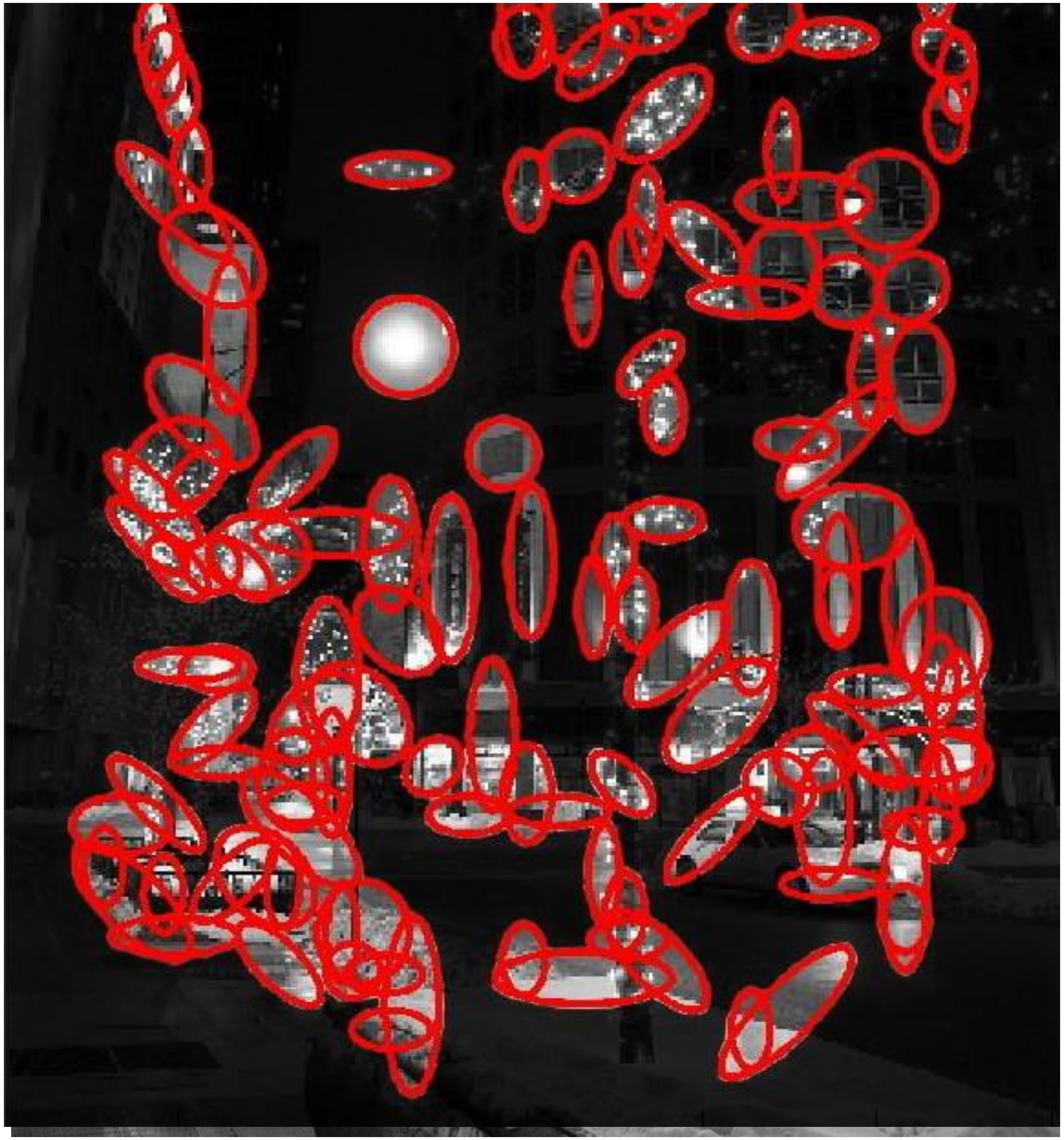
"edge":
no change
along the edge
direction



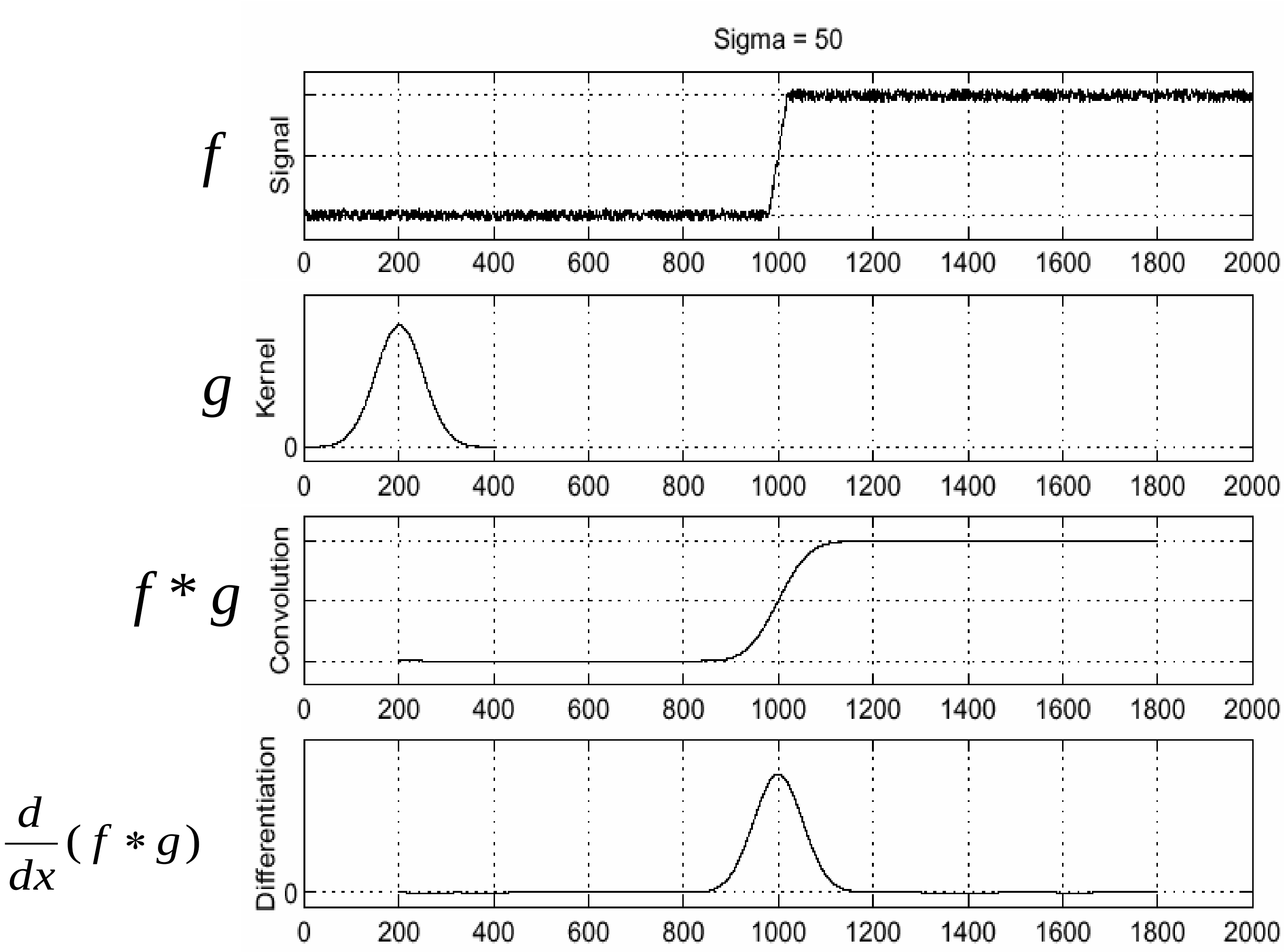
"corner":
significant
change in all
directions

Results

Blob detectors

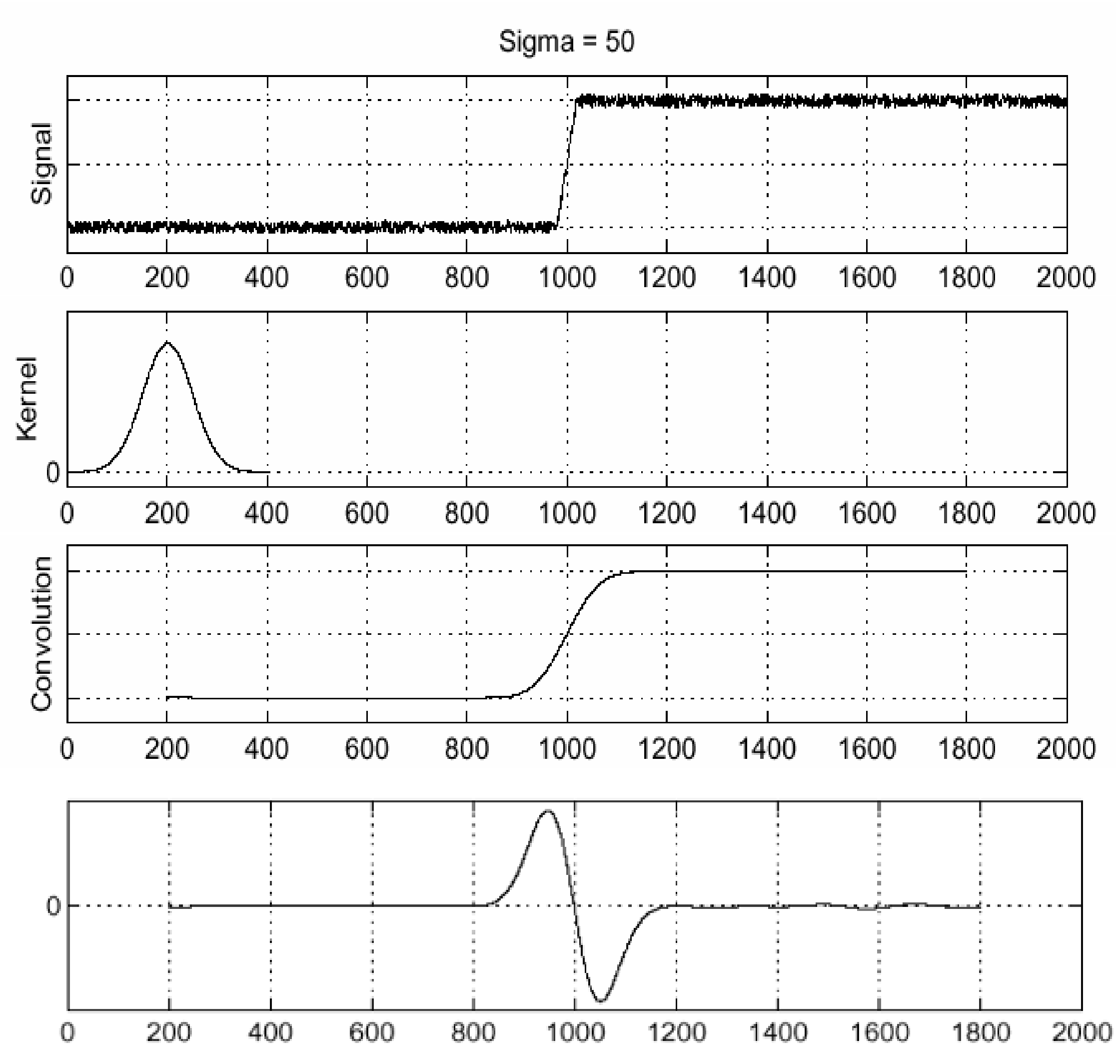


Edge detection



Source: S. Seitz

Edge detection



f

g

$f * g$

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

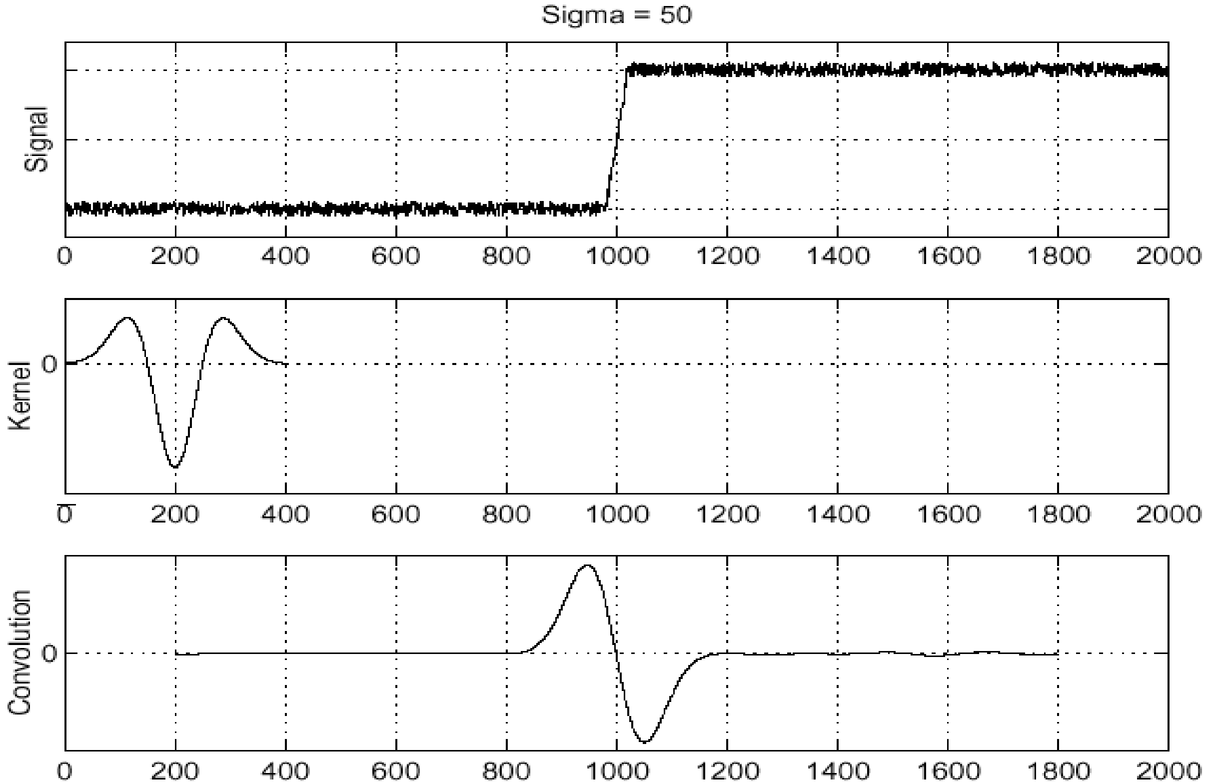
$$f * \frac{d^2}{dx^2} g$$

= "second derivative of Gaussian" filter = Laplacian of the gaussian

Edge detection as zero crossing

f

Edge



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

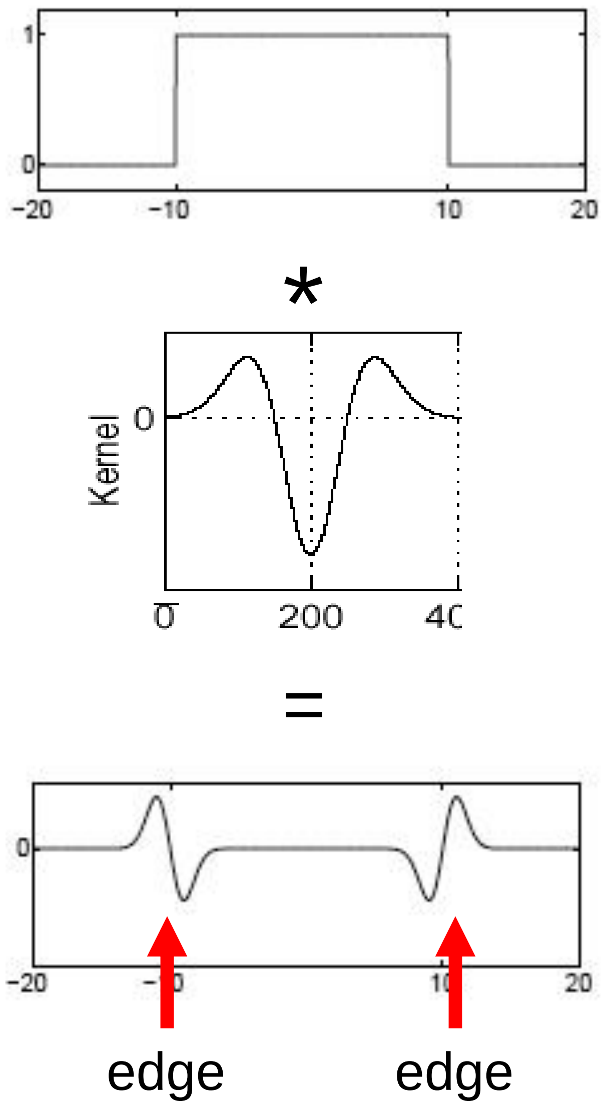
Second derivative of Gaussian (Laplacian)

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

Edge = zero crossing of second derivative

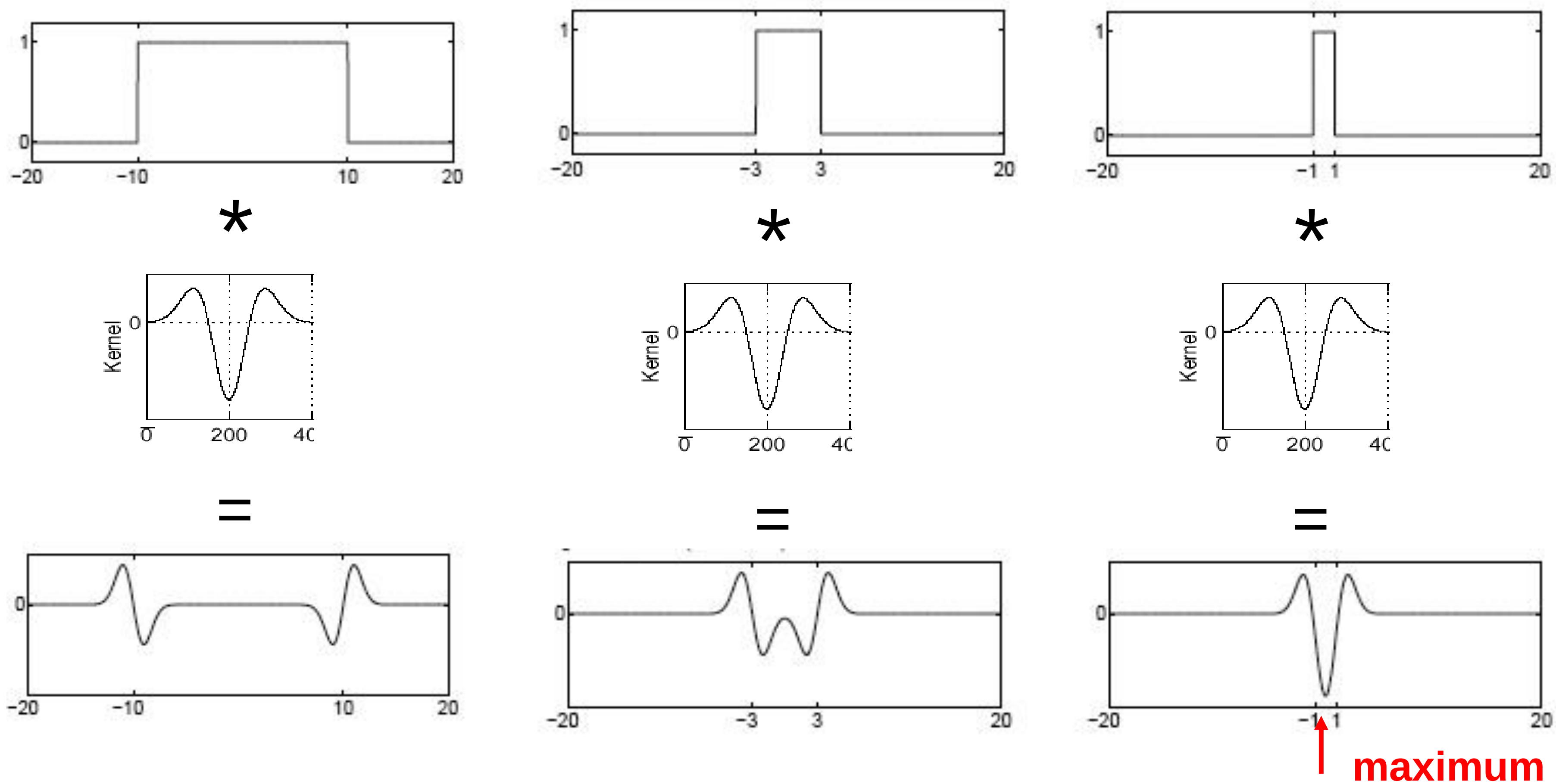
Source: S. Seitz

Edge detection as zero crossing



From edges to blobs

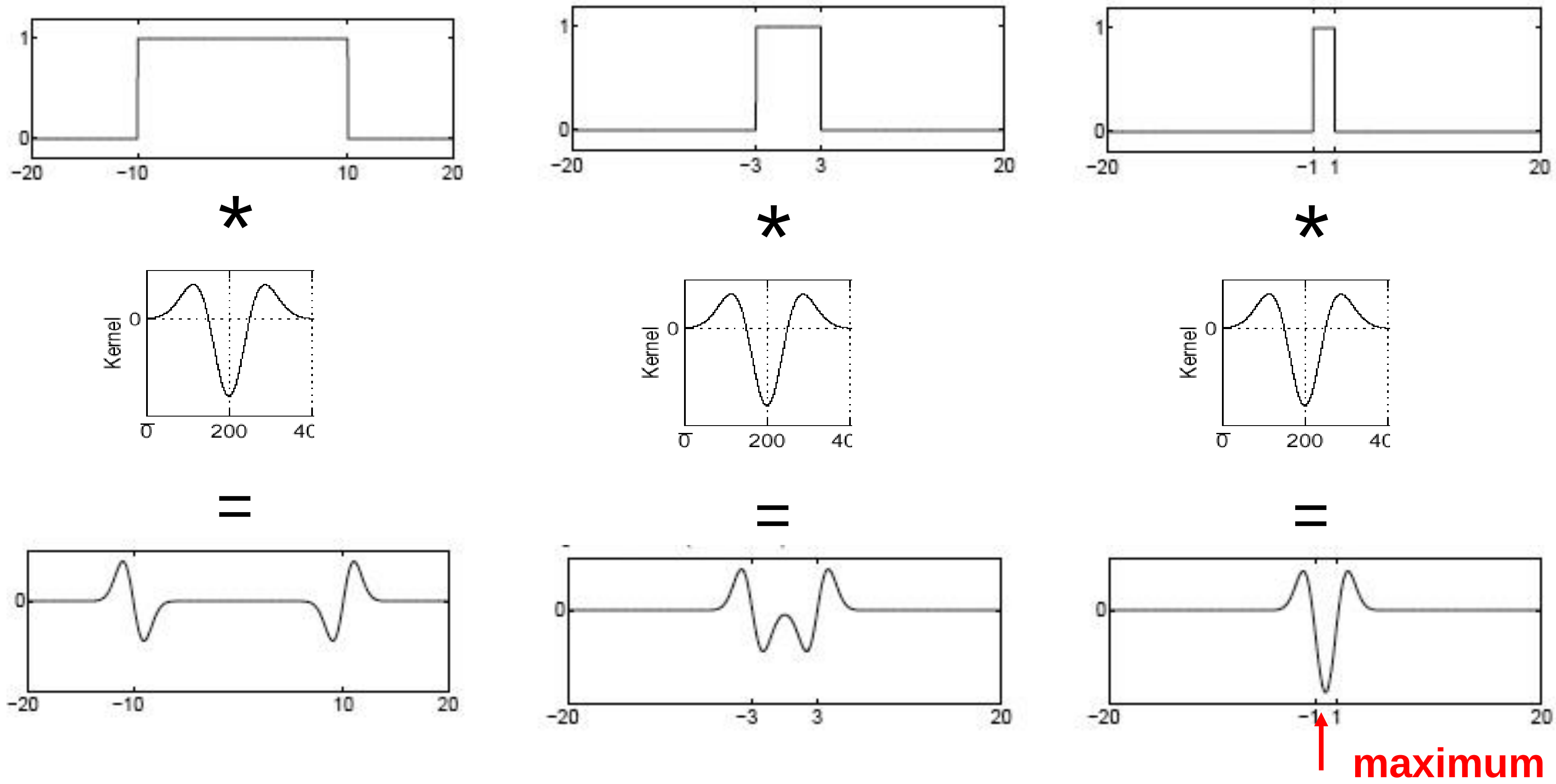
- Blob = superposition of nearby edges



Magnitude of the Laplacian response achieves a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob

From edges to blobs

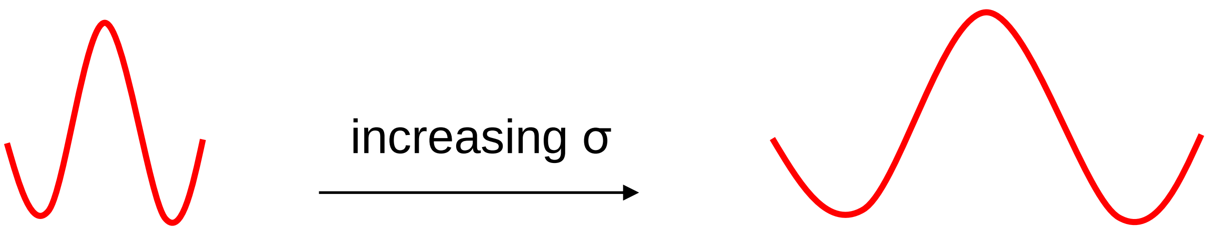
- Blob = superposition of nearby edges

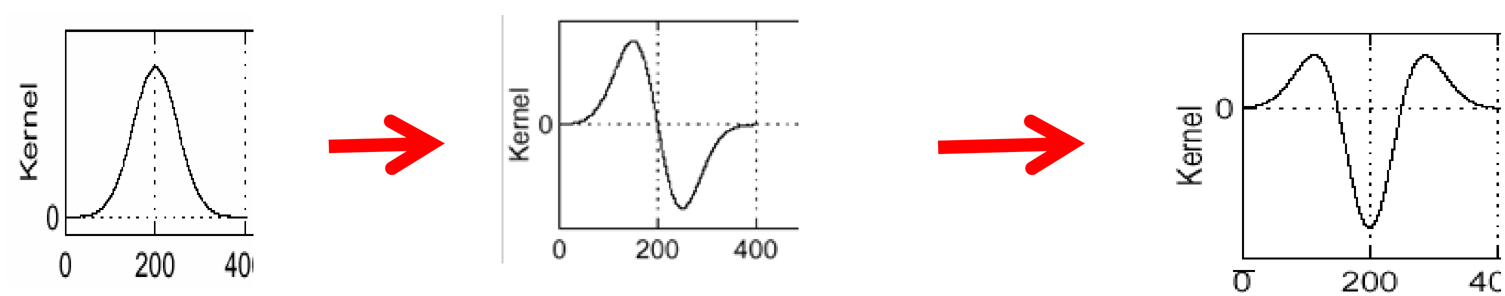


What if the blob is slightly thicker or slimmer?

Scale selection

Convolve signal with Laplacians at several sizes and looking for the maximum response





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

Scale normalization

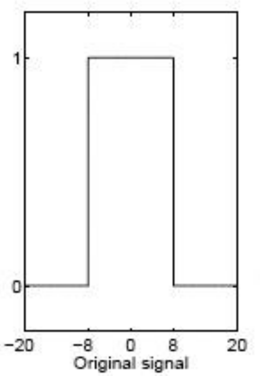
- To keep the energy of the response the same, must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

Characteristic scale

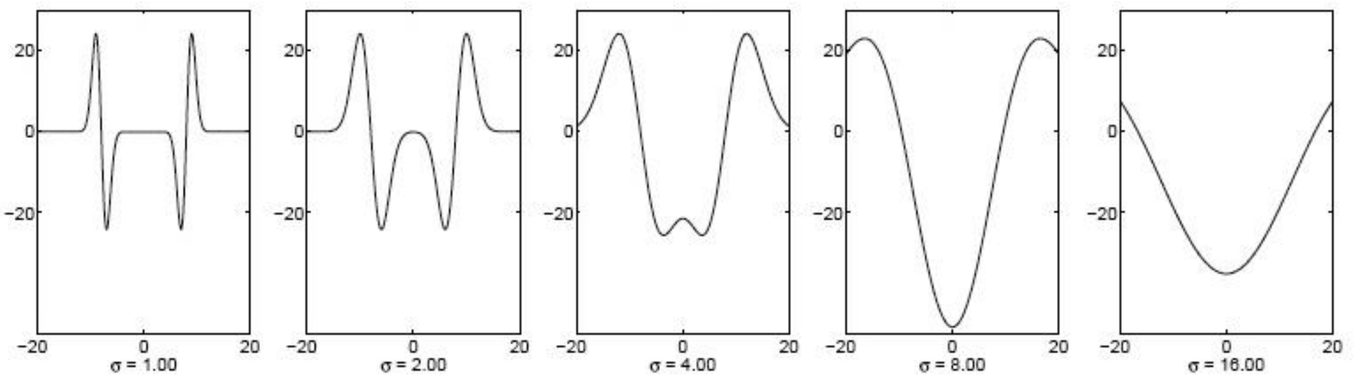
We define the characteristic scale as the scale that produces peak of Laplacian response

Scale-normalized Laplacian response

Original signal



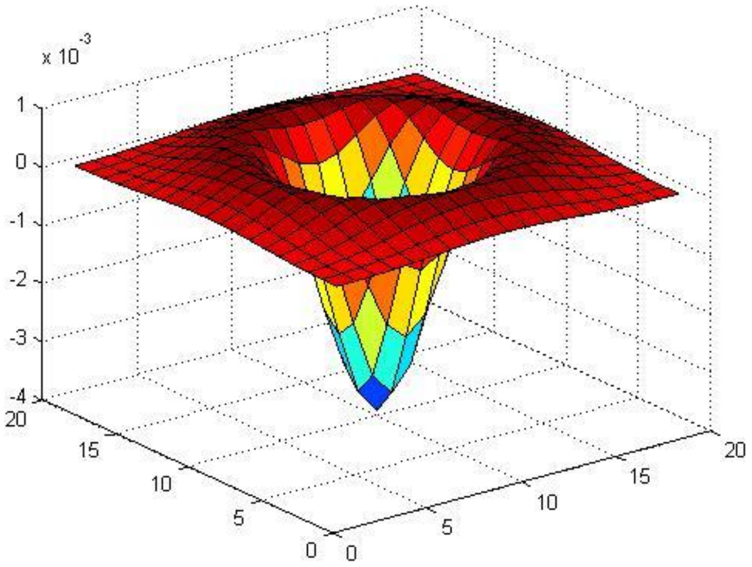
$\sigma = 1$ $\sigma = 2$ $\sigma = 4$ $\sigma = 8$ $\sigma = 16$

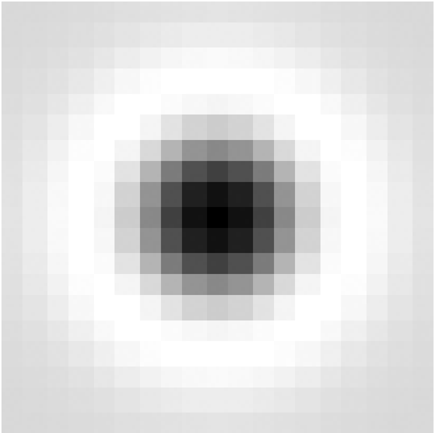


Maximum ☺

Blob detection in 2D

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

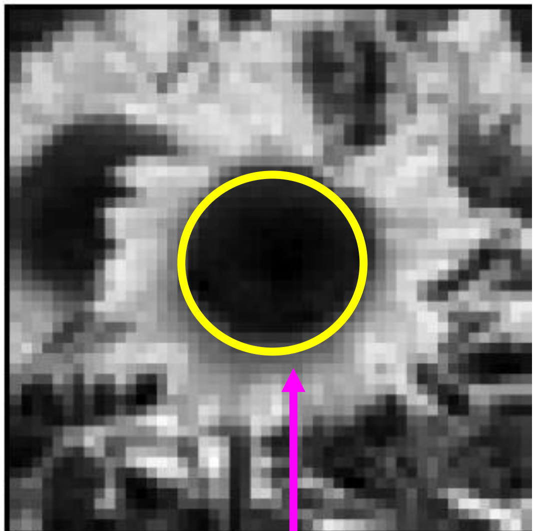


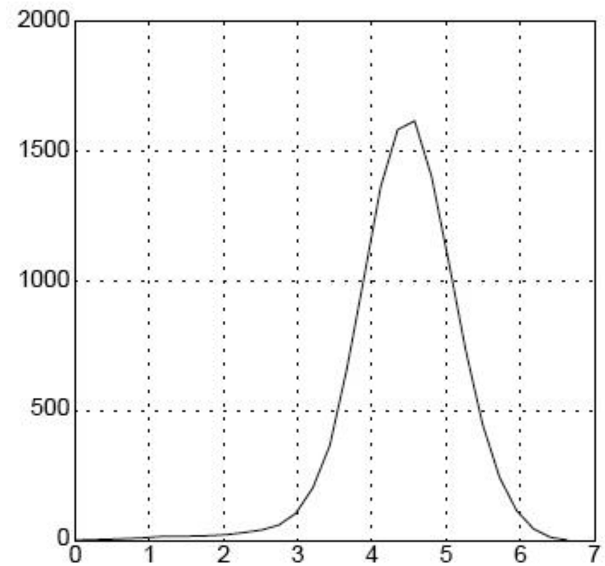


Scale-normalized: $\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$

Characteristic scale

- We define the characteristic scale as the scale that produces peak of Laplacian response





characteristic scale

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

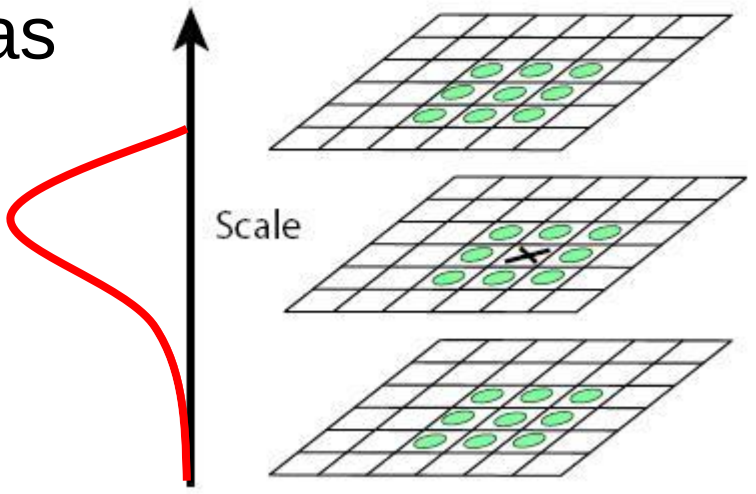
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales

2. Find maxima of squared Laplacian response in scale-space

3. This indicate if a blob has been detected

4. And what's its intrinsic scale

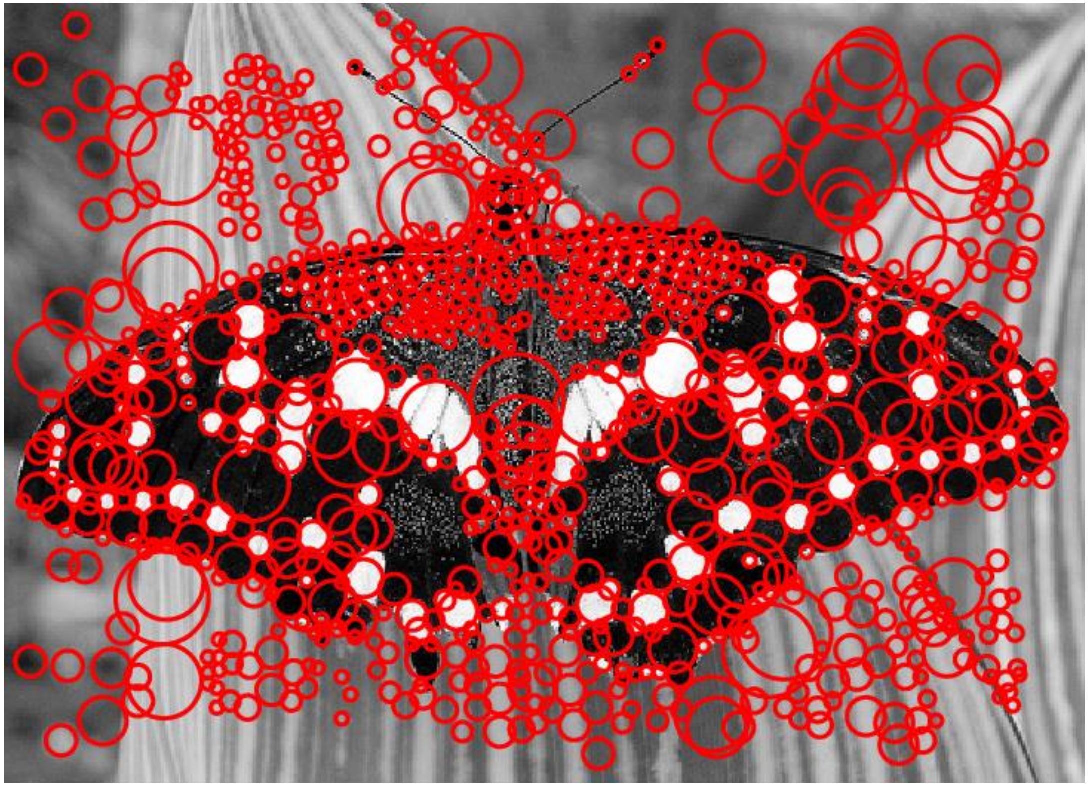


Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

Scale-space blob detector: Example



Difference of Gaussians (DoG)

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Approximating the Laplacian with a difference of Gaussians:

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

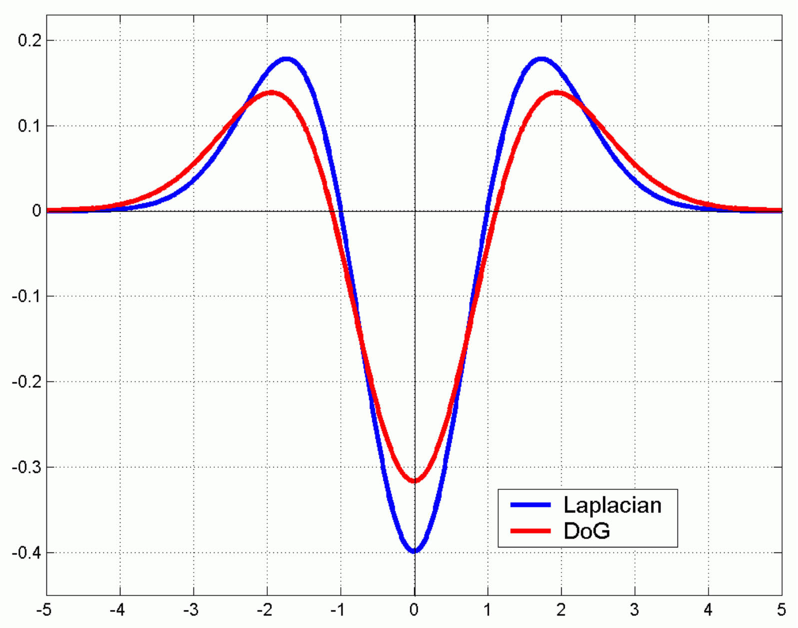
(Laplacian)

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

(Difference of Gaussians)

or

Difference of gaussian blurred images at scales k σ and σ



$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2 \boldsymbol{L}$$

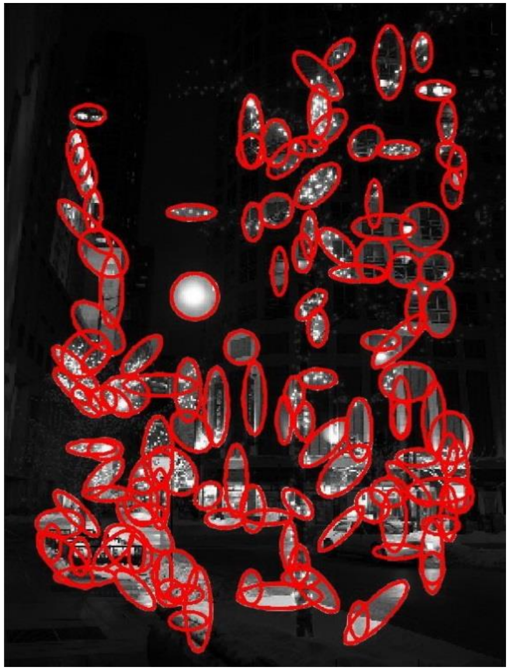
Detector	Illumination	Rotation	Scale	View point
Harris corner	Yes	Yes	No	No
Lowe '99 (DoG)	Yes	Yes	Yes	No
Mikolajczyk & Schmid '01, '02	Yes	Yes	Yes	Yes
Tuytelaars, '00	Yes	Yes	No (Yes '04)	Yes
Kadir & Brady, 01	Yes	Yes	Yes	no
Matas, '02	Yes	Yes	Yes	no

Lecture 10

Detectors and descriptors

- Properties of detectors
  - Edge detectors
  - Harris
  - DoG
- Properties of detectors
  - SIFT
  - Shape context

The big picture…



Feature Detection

e.g. DoG

Feature Description

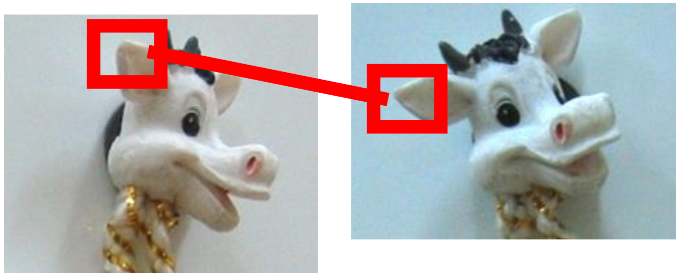
e.g. SIFT

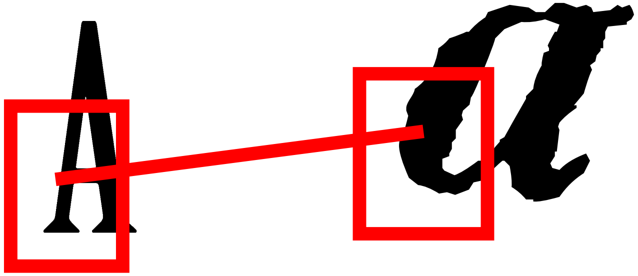
- Estimation
- Matching
- Indexing
- Detection

Properties

Depending on the application a descriptor must incorporate information that is:

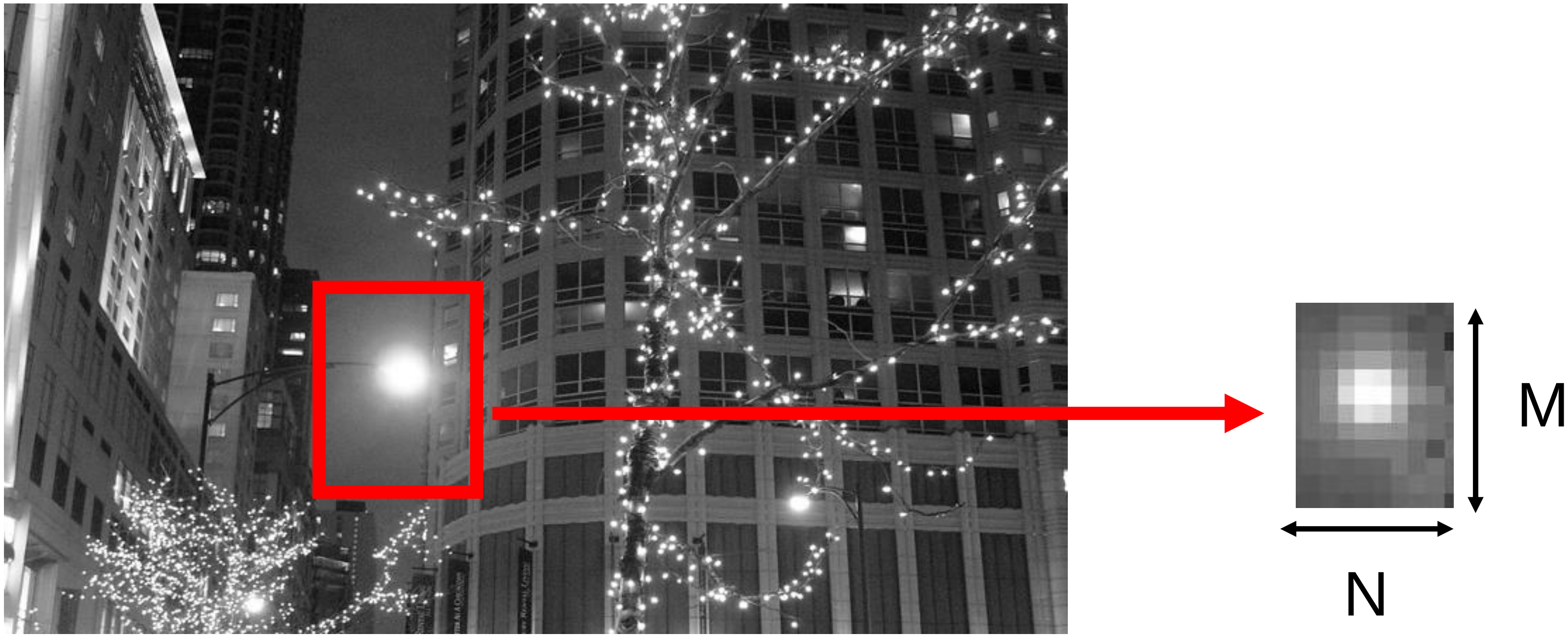
- Invariant w.r.t:
  - Illumination
  - Pose
  - Scale
  - Intraclass variability



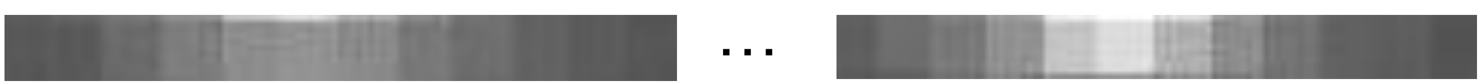


- Highly distinctive (allows a single feature to find its correct match with good probability in a large database of features)

The simplest descriptor



1 x NM vector of pixel intensities

w= []

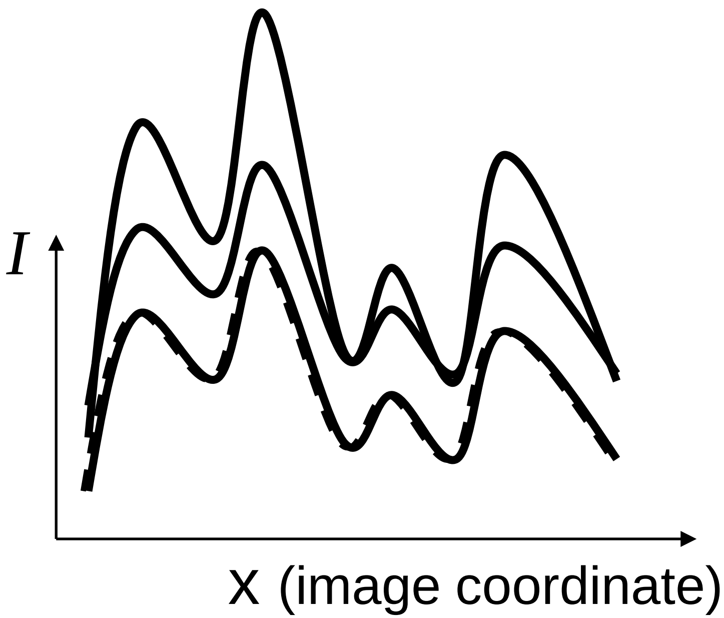
$$w_n = \frac{(w - \overline{w})}{\left\|(w - \overline{w})\right\|}$$

Makes the descriptor invariant with respect to affine transformation of the illumination condition

Illumination normalization

- A*ffine intensity* change:

$$I \rightarrow I + b$$
$$\rightarrow a\,I + b$$



- Make each patch zero mean:

$$\mu = \frac{1}{N}\sum_{x,y} I(x,y)$$
$$Z(x,y) = I(x,y) - \mu$$

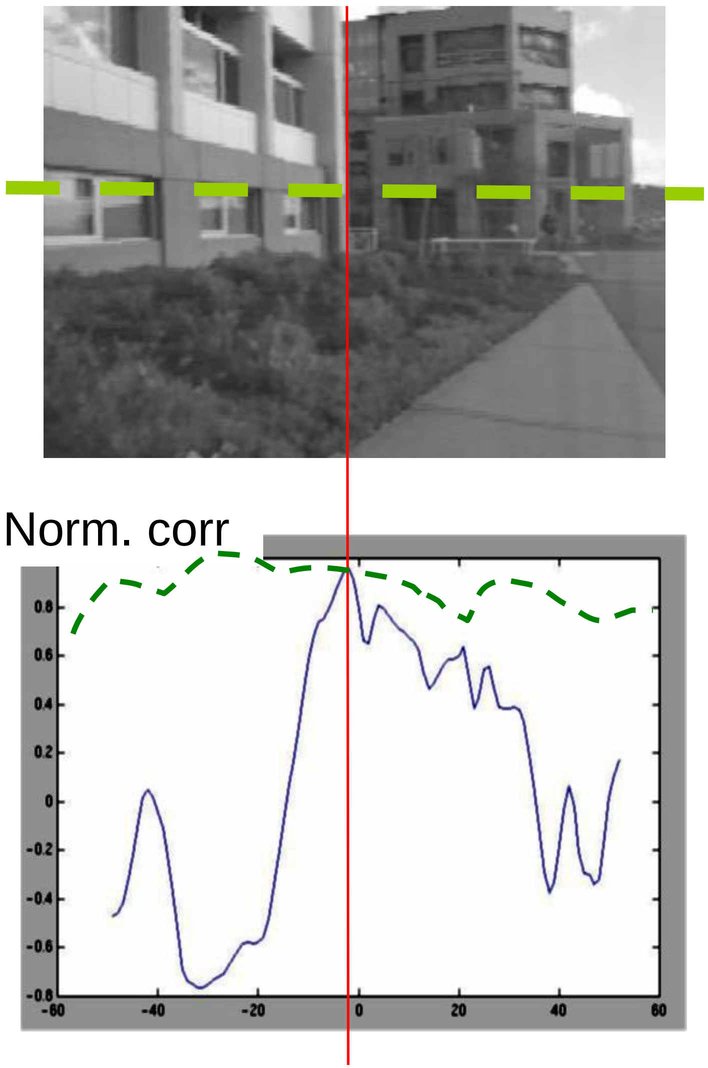
- Then make unit variance:

$$\sigma^2 = \frac{1}{N}\sum_{x,y} Z(x,y)^2$$
$$ZN(x,y) = \frac{Z(x,y)}{\sigma}$$

Why can’t we just use this?

- Sensitive to small variation of:
    - location
    - Pose
    - Scale
    - intra-class variability
- Poorly distinctive

Sensitive to pose variations

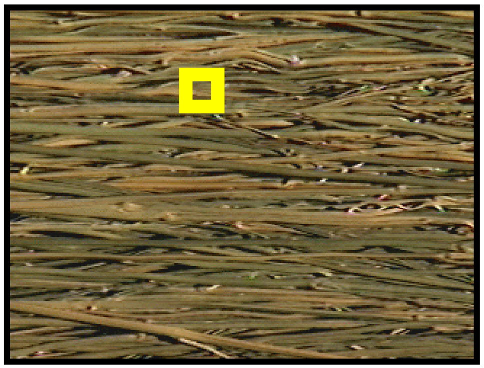


Normalized Correlation:

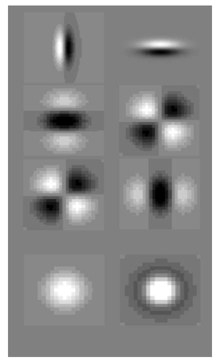
$$w_n \cdot w'_n = \frac{(w - \overline{w})(w' - \overline{w}')}{\left\|(w - \overline{w})(w' - \overline{w}')\right\|}$$

Descriptor	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor

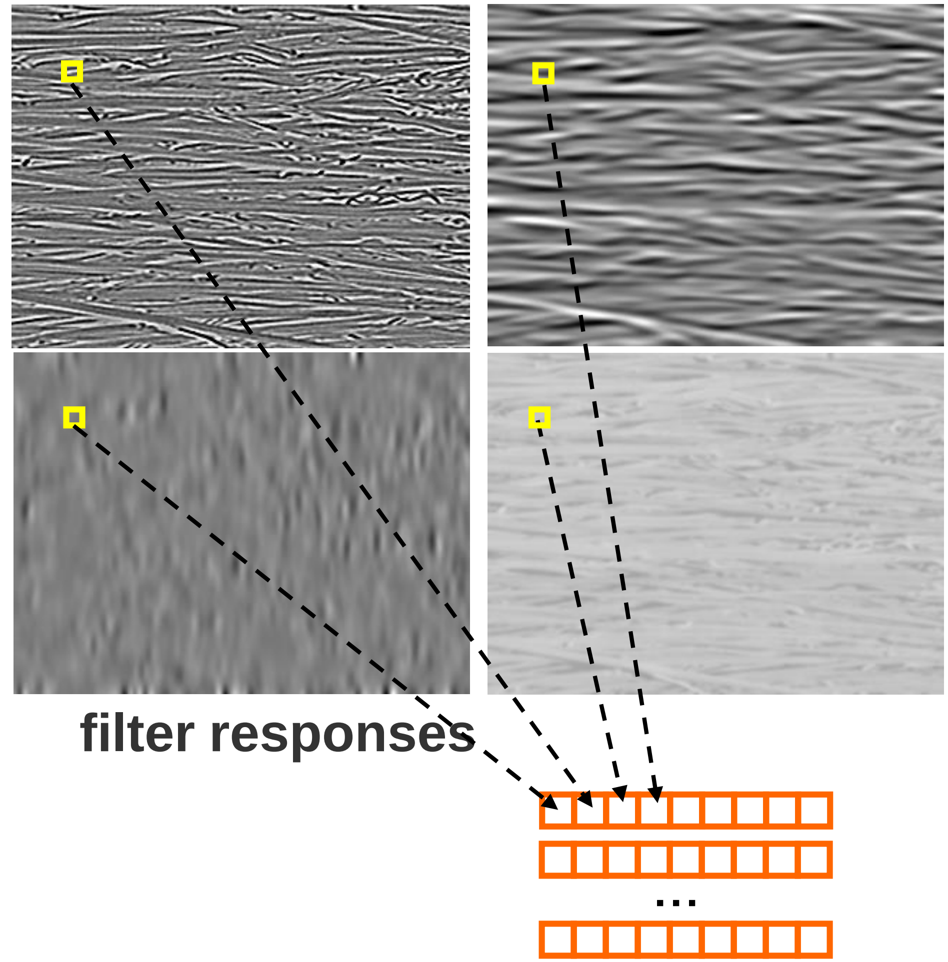
Bank of filters



image



filter bank



More robust but still quite sensitive to pose variations

Descriptor	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium

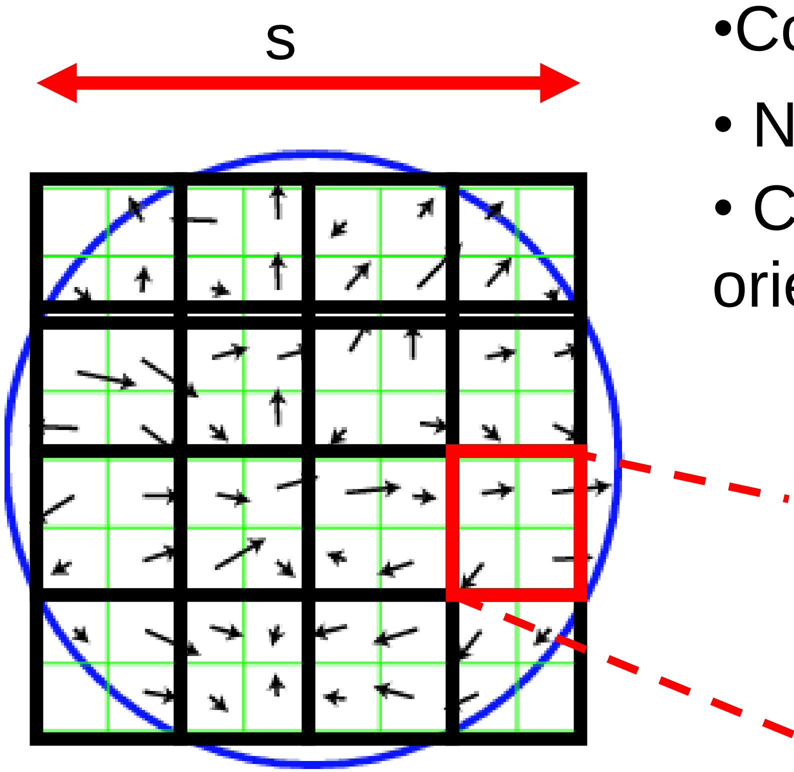
SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

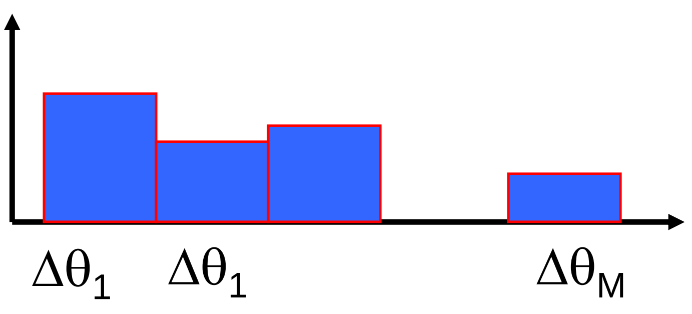
- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

SIFT descriptor

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector



- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bean



SIFT descriptor

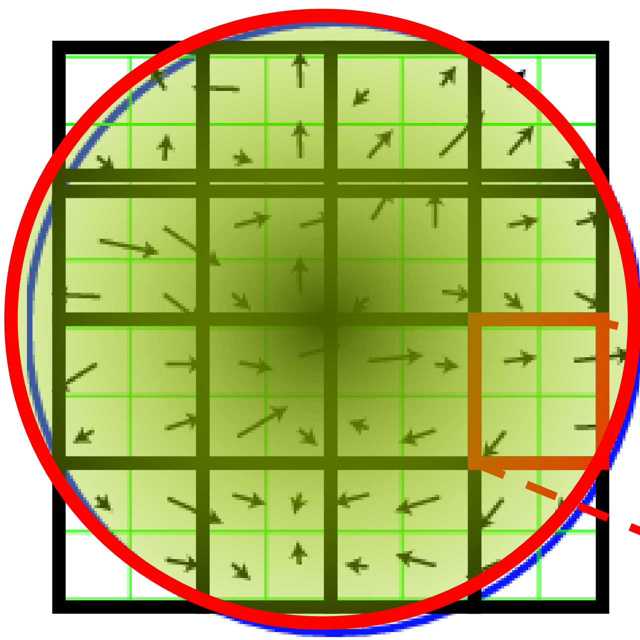
- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

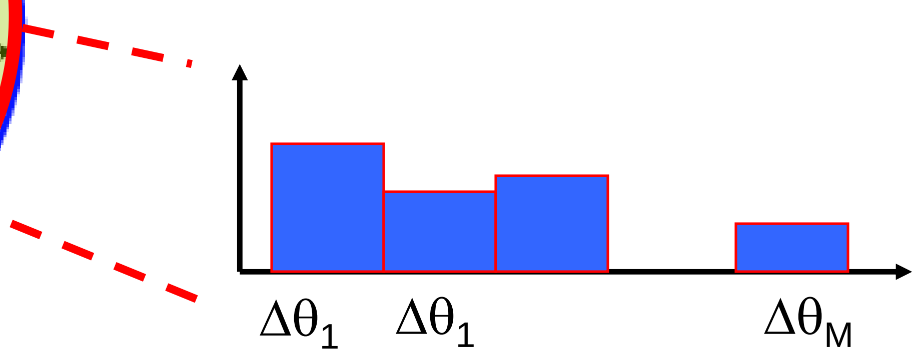
•Compute gradient at each pixel

• N x N spatial bins

• Compute an histogram of M orientations for each bean

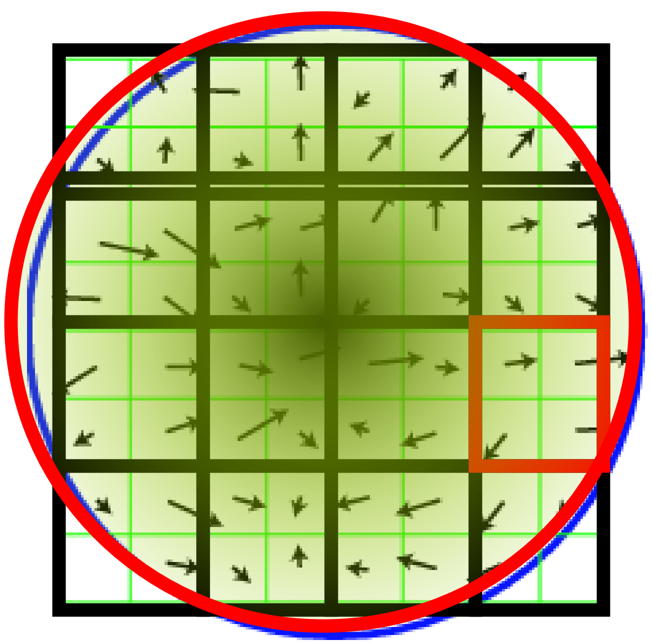
• Gaussian center-weighting





SIFT descriptor

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector



•Compute gradient at each pixel

• N x N spatial bins

• Compute an histogram of M orientations for each bean

• Gaussian center-weighting

• Normalized unit norm

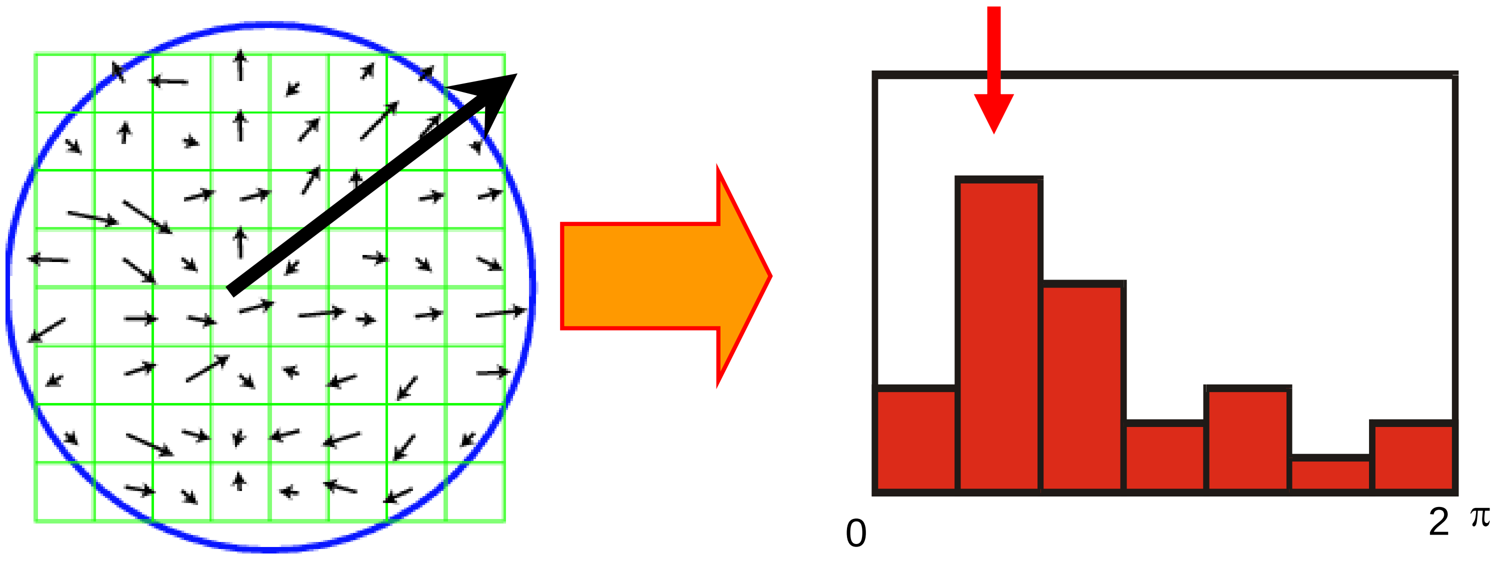
Typically M = 8; N= 4
1 x 128 descriptor

SIFT descriptor

- Robust w.r.t. small variation in:
  - Illumination (thanks to gradient & normalization)
  - Pose (small affine variation thanks to orientation histogram)
  - Scale (scale is fixed by DOG)
  - Intra-class variability (small variations thanks to histograms)

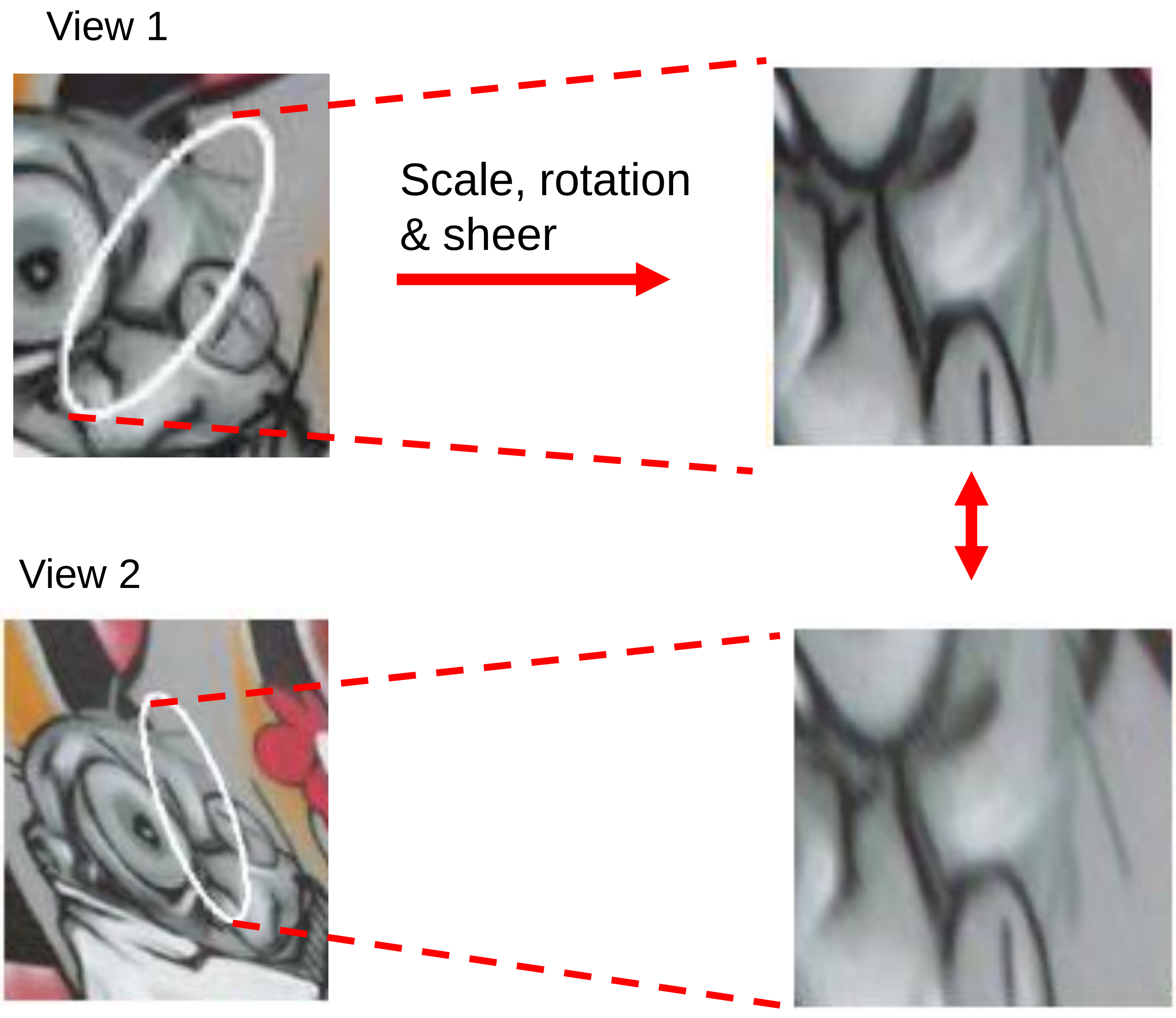
Rotational invariance

- Find dominant orientation by building a orientation histogram
- Rotate all orientations by the dominant orientation



This makes the SIFT descriptor rotational invariant

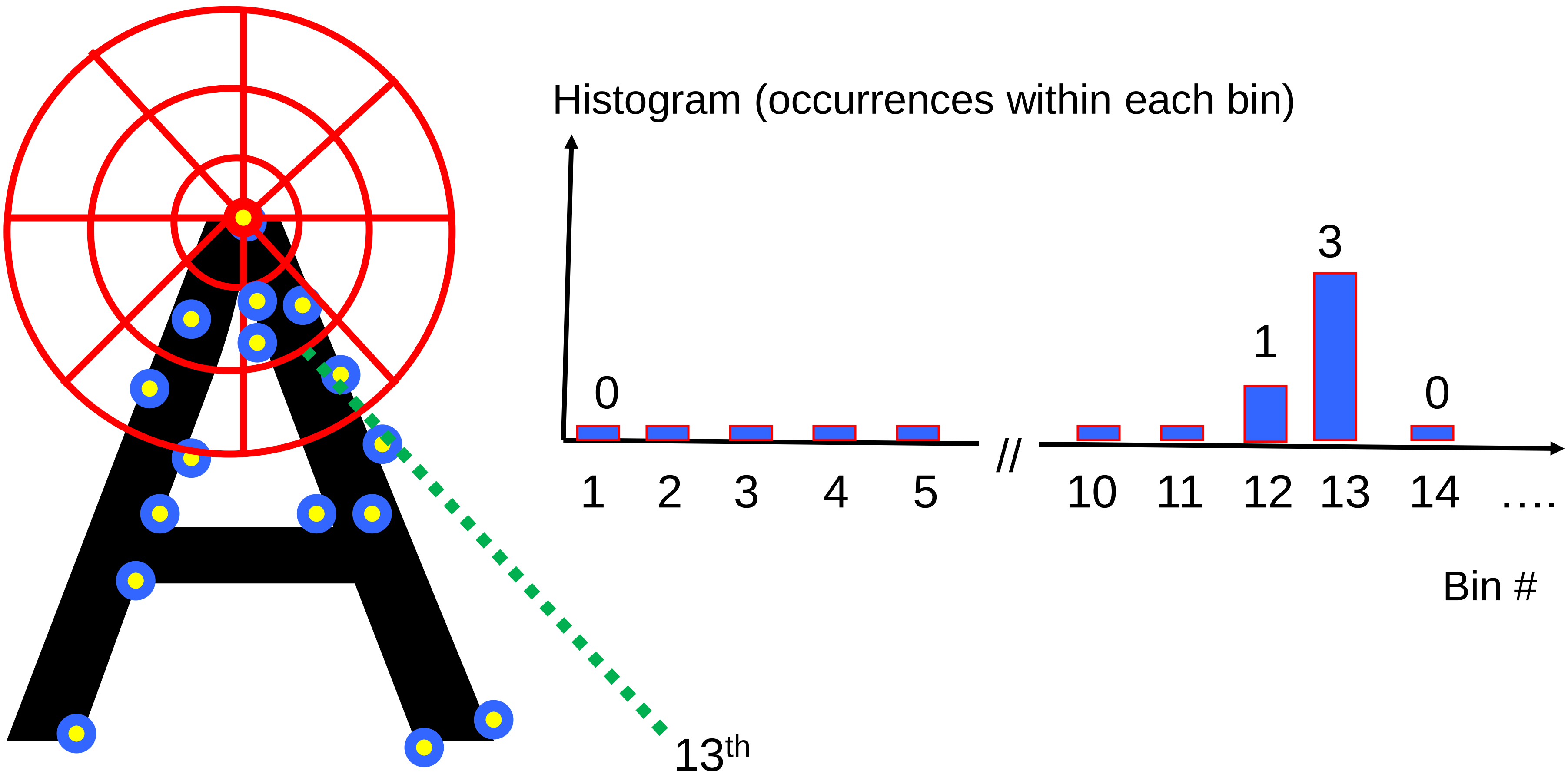
Pose normalization



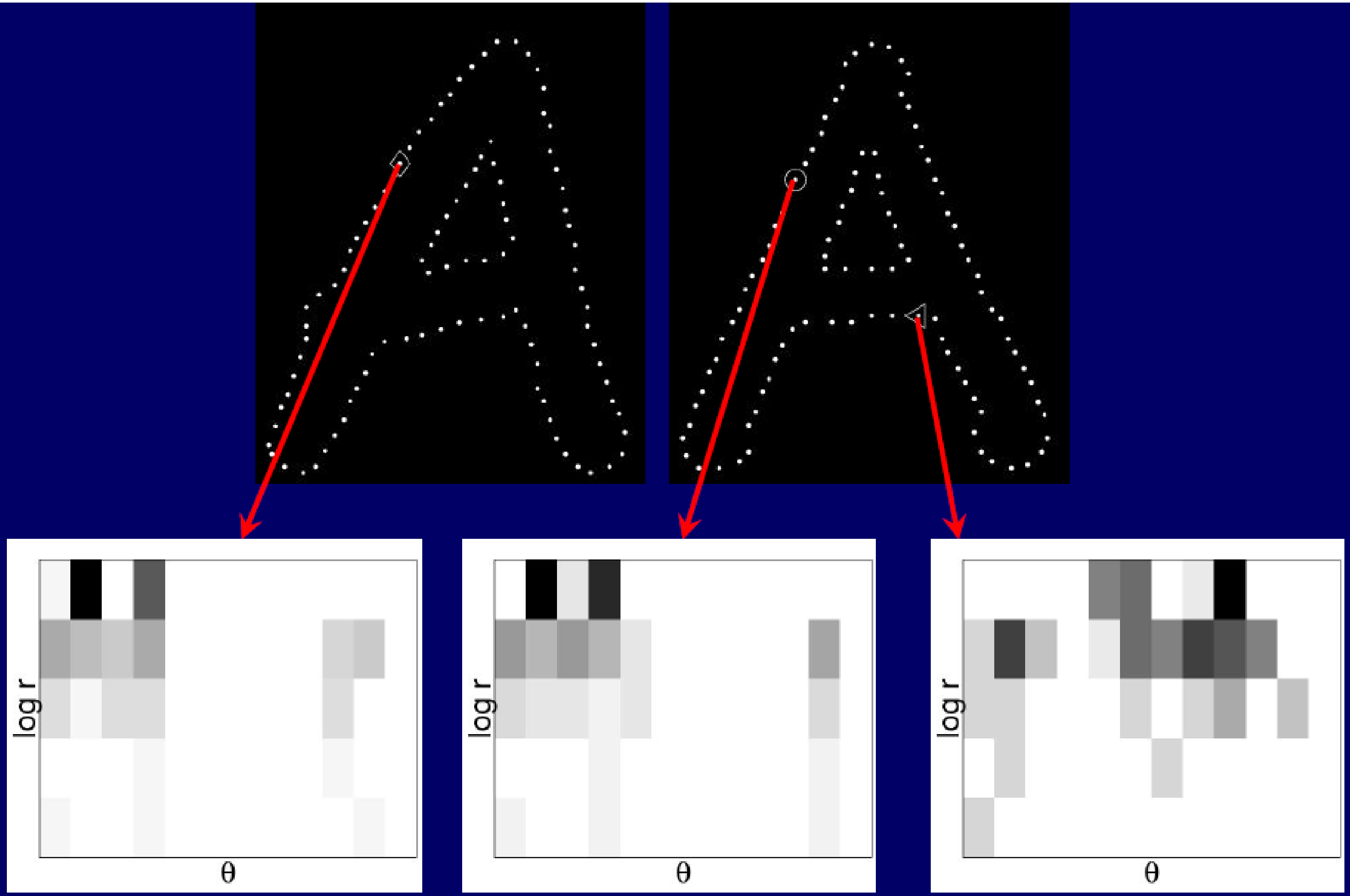
Descriptor	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium
SIFT	Good	Good	Medium

Shape context descriptor

Belongie et al. 2002



Shape context descriptor



Other detectors/descriptors

- **HOG: Histogram of oriented gradients**

  Dalal & Triggs, 2005

- **SURF: Speeded Up Robust Features**

  Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

- **FAST (corner detector)**

  Rosten. Machine Learning for High-speed Corner Detection, 2006.

- **ORB: an efficient alternative to SIFT or SURF**

  Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski: ORB: An efficient alternative to SIFT or SURF. ICCV 2011

- **Fast Retina Key- point (FREAK)**

  A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. In IEEE Conference on Computer Vision and Pattern Recognition, 2012. CVPR 2012 Open Source Award Winner.

Next lecture:

Image Classification by Deep Networks