Multi-modal Registration of Visual Data

Massimiliano Corsini

Visual Computing Lab, ISTI - CNR - Italy

Overview

- Introduction and Background
- Features Detection and Description (2D case)
- Features Detection and Description (3D case)
- Image-geometry registration
- Recent Advances and Applications

Overview

- Introduction and Background
- **Features Detection and Description (2D case)**
  - Image Filtering and Image Derivatives
  - Edge extraction (Hough Transform, RANSAC)
  - Corners - Harris Corner Detector
  - Scale Invariant Detection (LoG, DoG)
  - Scale Invariant Feature Transform (SIFT)
  - SURF
  - Affine-invariant Regions (IBR, MSER)
  - Histogram of Oriented Gradients (HOG)
- Features Detection and Description (3D case)
- Image-geometry registration
- Recent Advances and Applications

Features Detection and Description (2D case)

Local and Global Features

- Features can be *global* or *local*.
- We concentrate on local features, which are more robust w.r.t:
  - ***Occlusions***
  - ***Variations***
- Local features can be widely applied.

Main Motivations

- Image Registration
- 3D Reconstruction
- Visual Tracking
- Object Recognition
- Etc..

Automatic Panorama

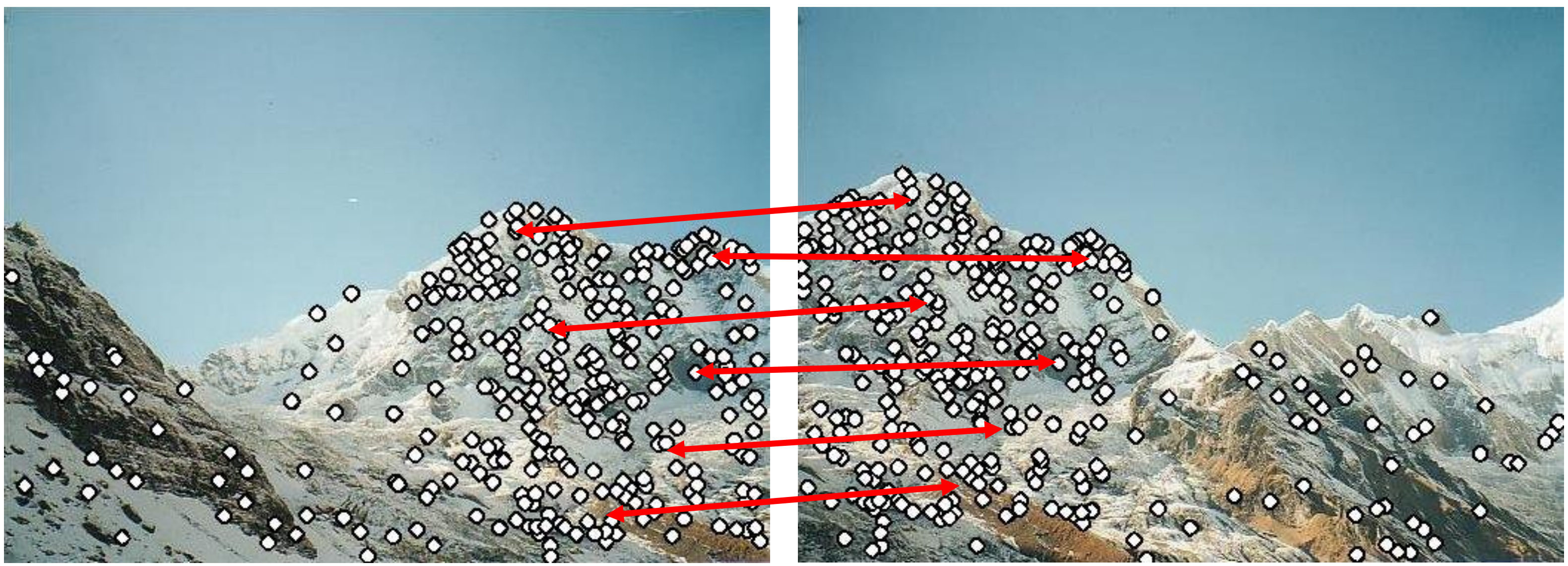
Musée du Louvre - Paris by David Engle (from GigaPan.com)

Panorama Creation

- We have two images – how do we combine them?

Panorama Creation

- Motivation: panorama stitching
  - We have two images – how do we combine them?



Step 1: extract features
Step 2: match features

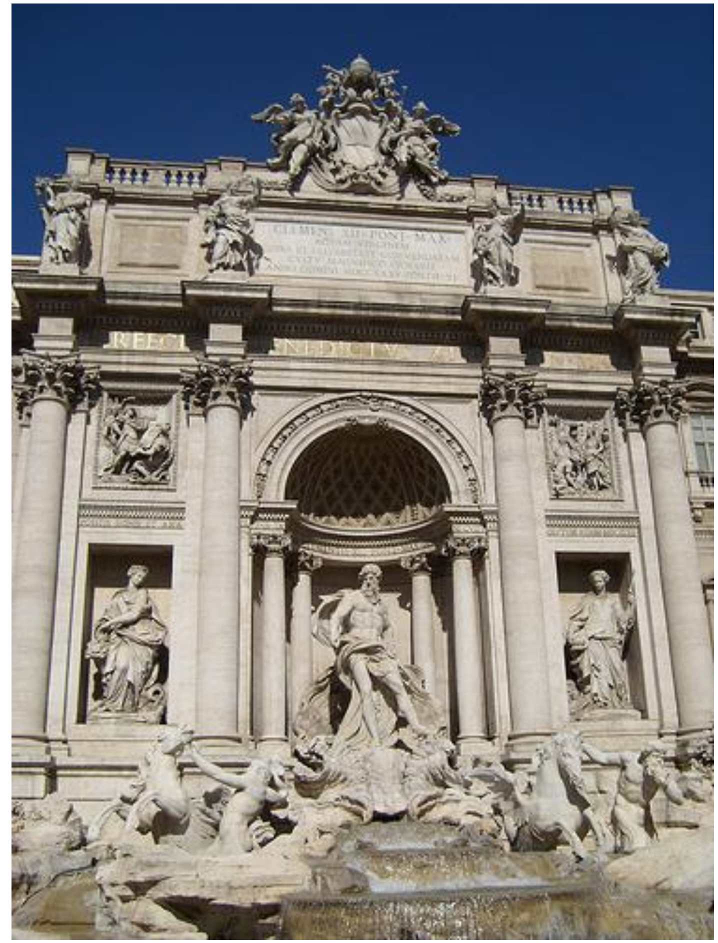
Panorama Creation

- We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
Step 3: align images

Image matching

by Diva Sian



by swashford

Harder case

by [Diva Sian]

by [scgbt]

Visual Tracking - SLAM

- Let’s me show you a video..

Features Properties

- Robustness
  - Invariant to translation, rotation, scale
  - Robust to *affine* geometric transformation
  - Robust to photometric variations
- Distinctiveness (“interesting” structure)
- Locality (local features are usually robust to occlusions and clutter)
- Repeatability
- Accuracy
- Quantity
- Computational Efficiency

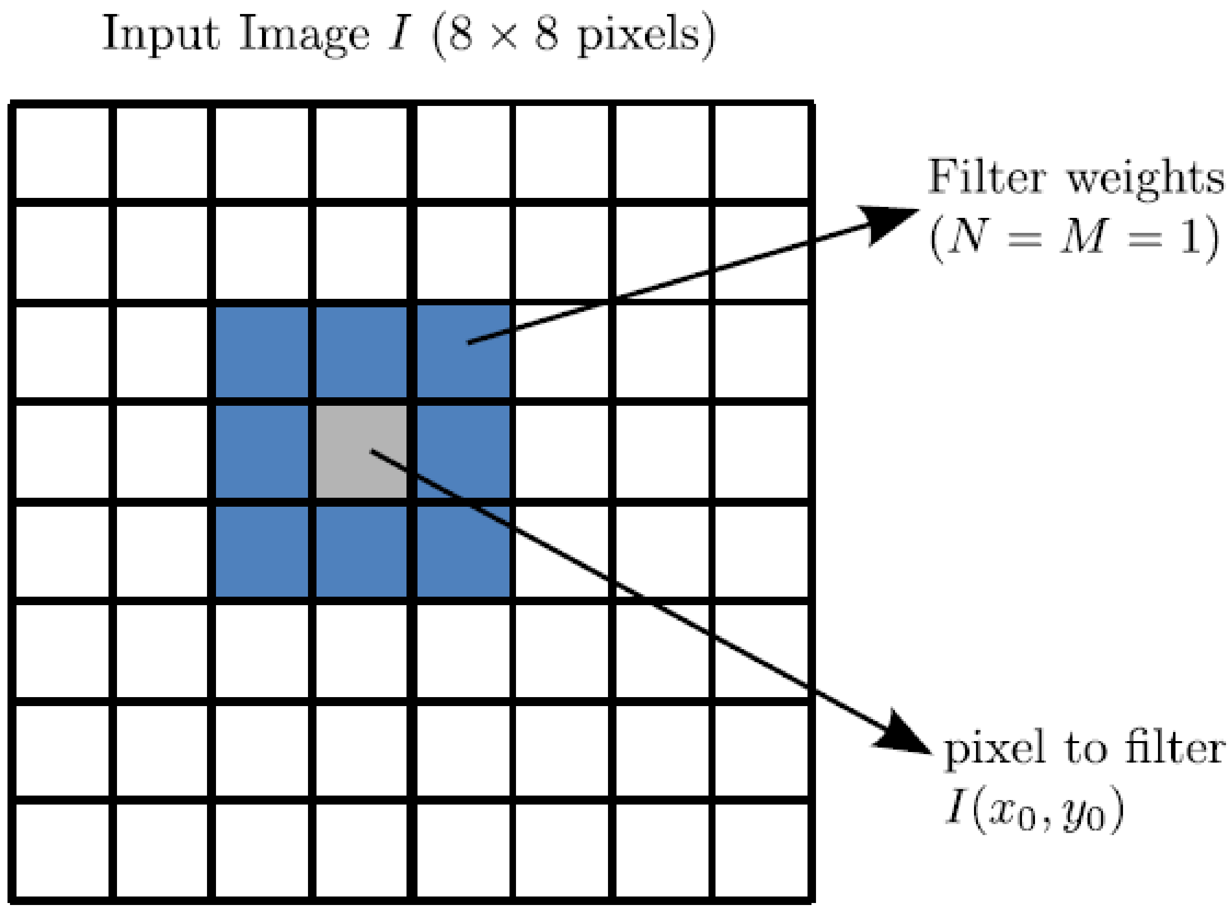
Features Detection and Description

- Image Features:
  - Corners
  - Edges
  - Blobs
  - Keypoints
- The features detected can be described using a *feature descriptor.*

Image Processing

- Many feature detectors involve the computation of first or second order derivatives.
- Often the image is filtered before to calculate its derivatives.
- Image derivatives are approximated using forward/backward of central differences.

Image Filtering



A generic filter of 3 x 3 kernel size.

Image Filtering

$$I'(x_0, y_0) = \frac{1}{T} \sum_{x=x_0-N}^{x_0+N} \sum_{y=y_0-M}^{y_0+M} W(x+N-x_0, y+M-y_0) I(x,y)$$

Pixel Filtered

Normalization factor

Filter kernel (weights)

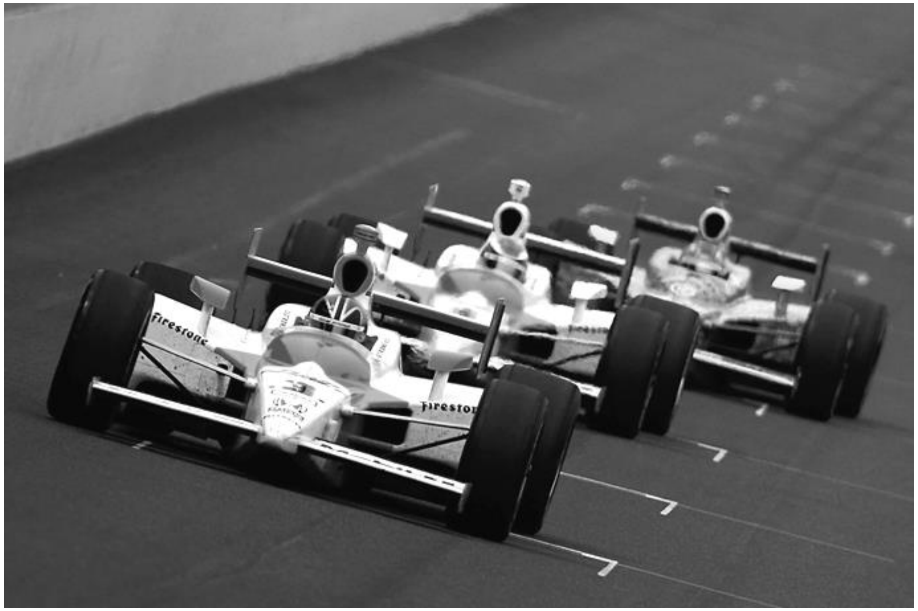
Blurring – Box Filter

- Simpler form of blurring → averaging the pixel values on the support of the filter.
- Constant weighting function:

$$W(i,j) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Example of 5 x 5 kernel (T = 25)

Box Filter



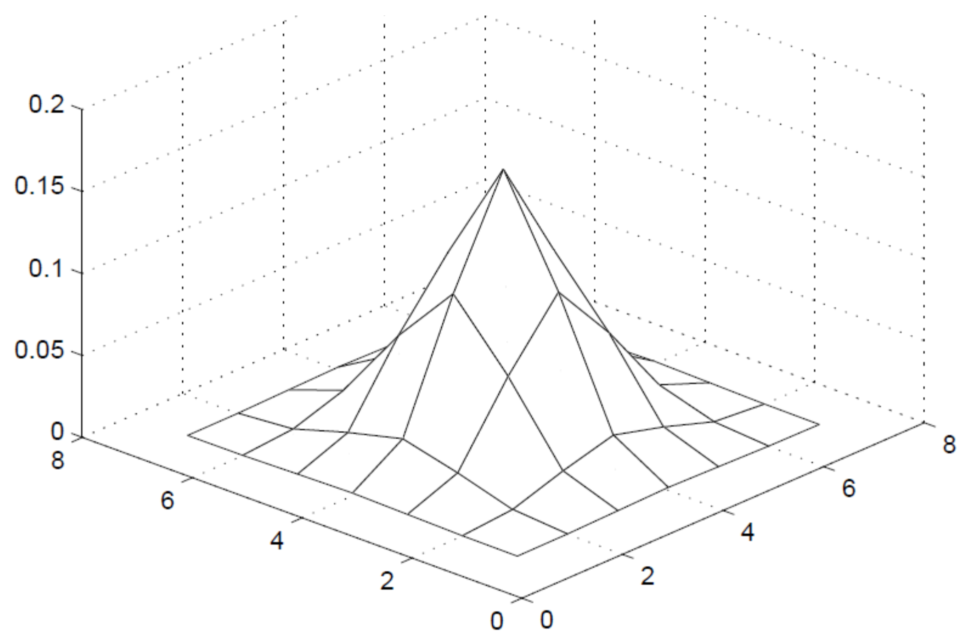
Original Image

Image Filtered (9x9 box filter)

Blurring – Gaussian Filter

- More the pixels are far from the central one and less they influence the average.
- 2D Gaussian: $$g(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Weights of a 7x7 Gaussian filter



Gaussian Filter

Original Image

Image Filtered
(9x9 Gaussian filter)

Image Derivatives

- Numerical approximations of derivatives:

Forward Differences: $$\Delta_x(x_i) = f(x_{i+1}) - f(x_i)$$

Backward Differences: $$\Delta_x(x_i) = f(x_i) - f(x_{i-1})$$

Central Differences: $$\Delta_x(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2}$$

Image Derivatives

- First-order derivatives using the central differences:

Horizontal derivative

$$\frac{\partial I}{\partial x} = I_x = I(x+1, y) - I(x-1, y)$$

$$\frac{\partial I}{\partial y} = I_y = I(x, y-1) - I(x, y+1)$$

Vertical derivative

Image Derivatives

- Matrix form of the first-order derivatives:

$$W_{\Delta_x} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad W_{\Delta_y} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Image Derivatives (I_x)

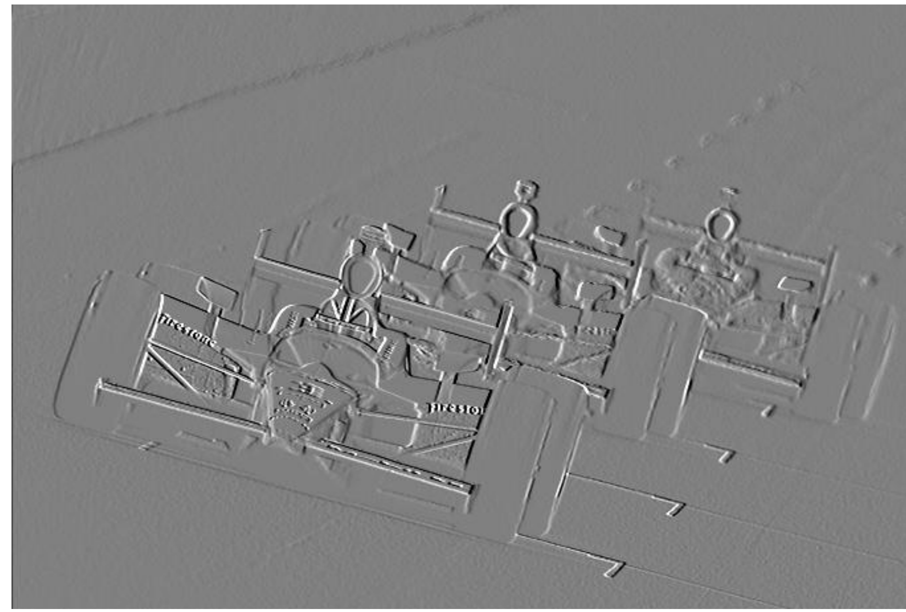


Image Derivatives (I_y)

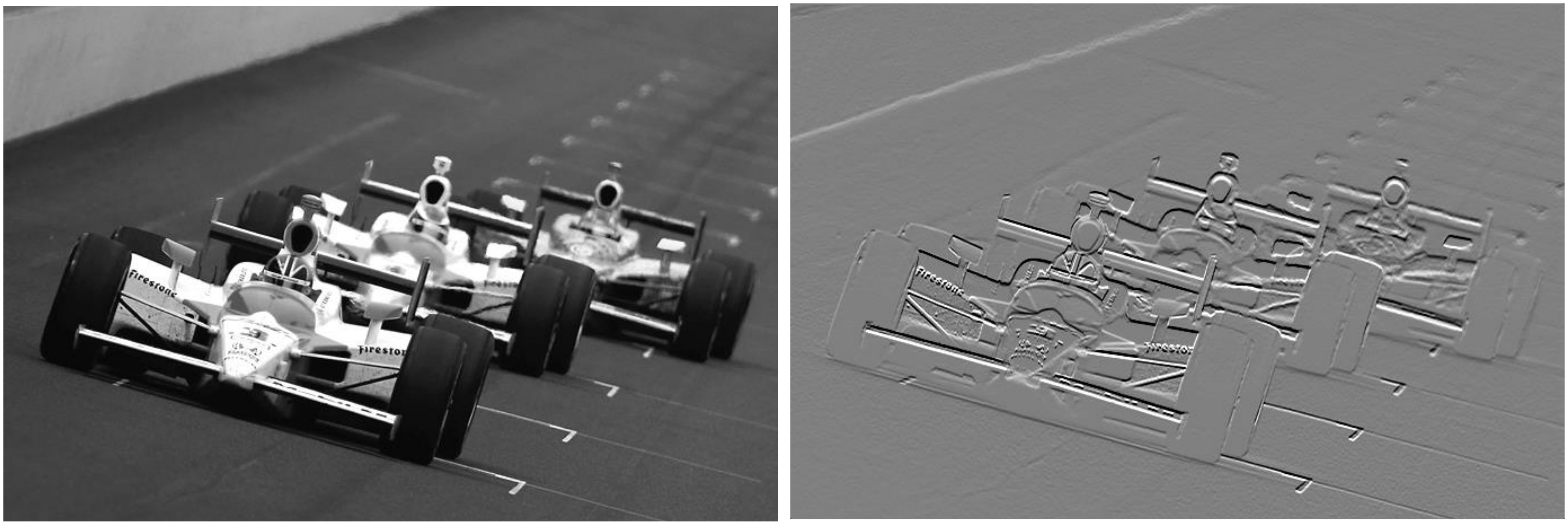


Image Derivatives

- More accurate numerical approximations:

Prewitt operator:

$$W_{\Delta_x} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad W_{\Delta_y} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Sobel operator:

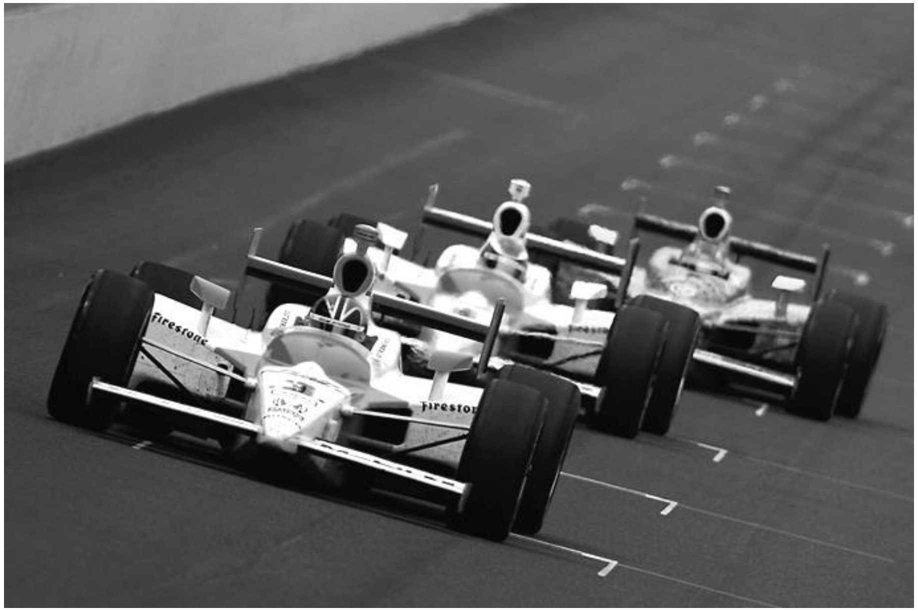
$$W_{\Delta_x} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad W_{\Delta_y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Edge Detection with the Sobel Operator

- Apply the Sobel operator and obtain I_x and I_y
- Gradient magnitude: $|G| = \sqrt{I_x^2 + I_y^2}$

  (edge strength)
- Gradient direction: $G_\theta = \operatorname{atan2}(I_y, I_x)$

Edge Detection with the Sobel Operator



Edge extraction

- We know for each pixel the edge strength (and the direction).
- Some parts of the image lines are missing / some parts are noisy.
- Improve the detection → fit segments/lines
  - RANSAC
  - Hough transform

RANSAC

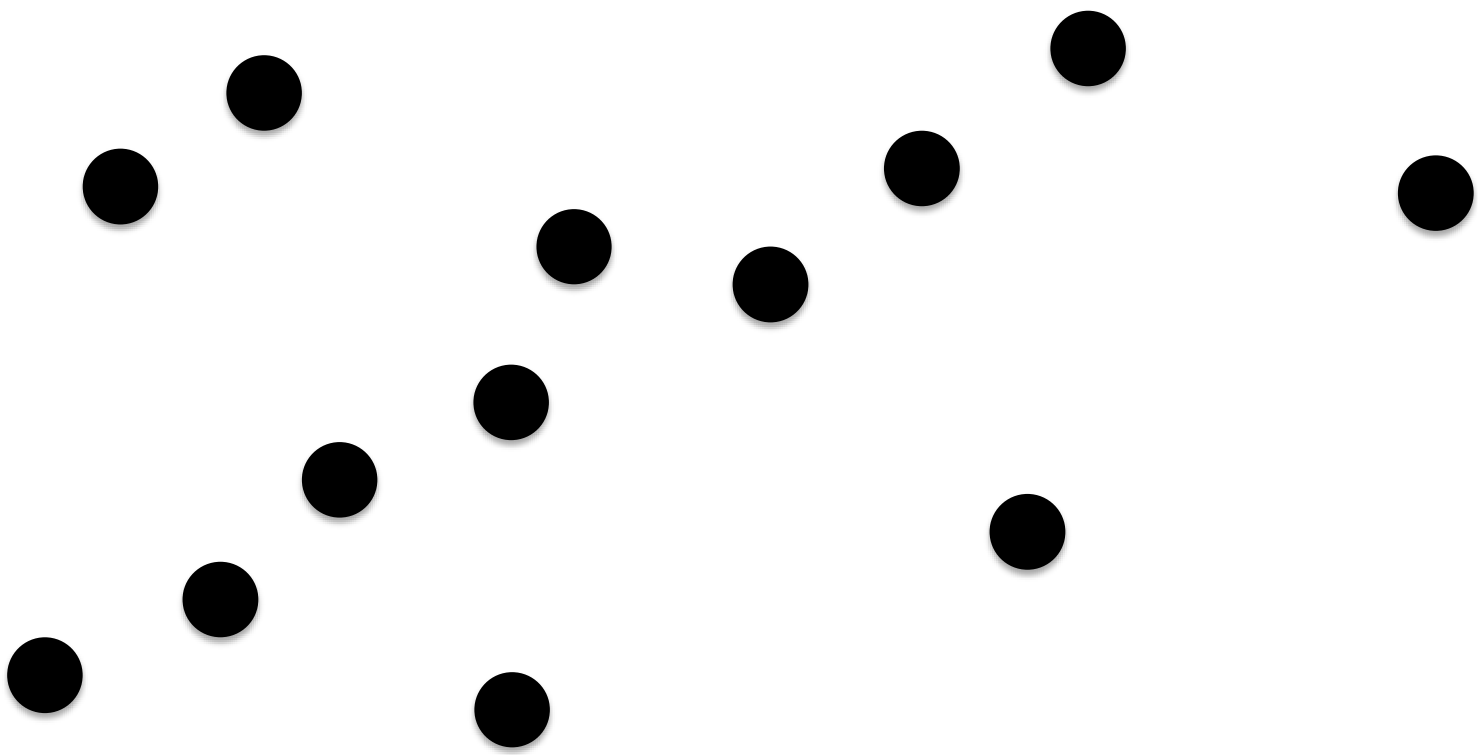
- ***Random Samples Consensus (RANSAC):*** an iterative general method to estimate parameters of a mathematical model starting from data containing (many) outliers.
- Not only for edges → widely applied in Computer Vision (!) → simple and very robust to the presence of outliers.

Martin A. Fischler and Robert C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", Comm. of the ACM 24 (6): pp. 381–395, 1981.

RANSAC algorithm

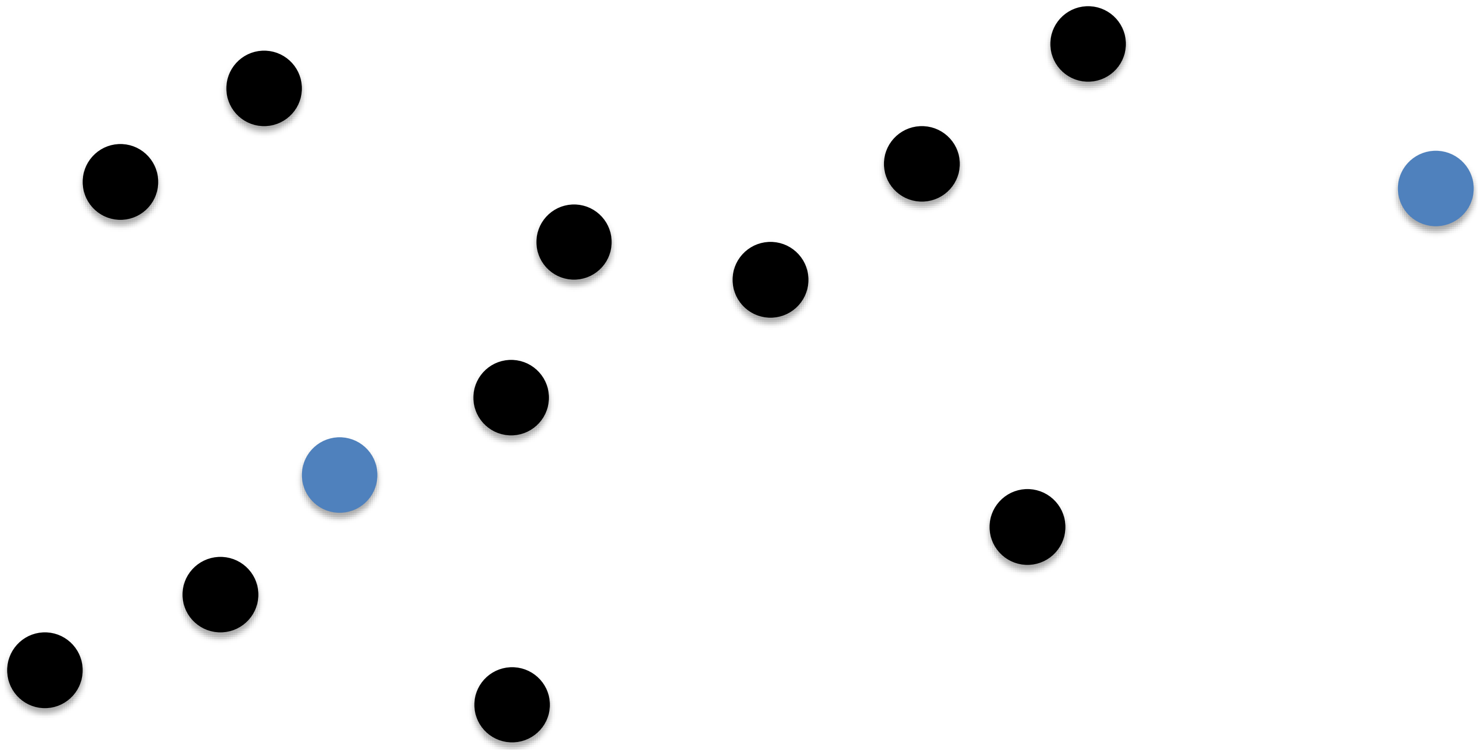
- Get randomly a minimal set of data that solves the model we want to estimate
- Check the number of inliers (evaluate the consensus)
- Iterate
  - Until a maximum number of iterations is reached
  - Until a certain stop condition is reached
- Get the solution with the maximum consensus

RANSAC for edges



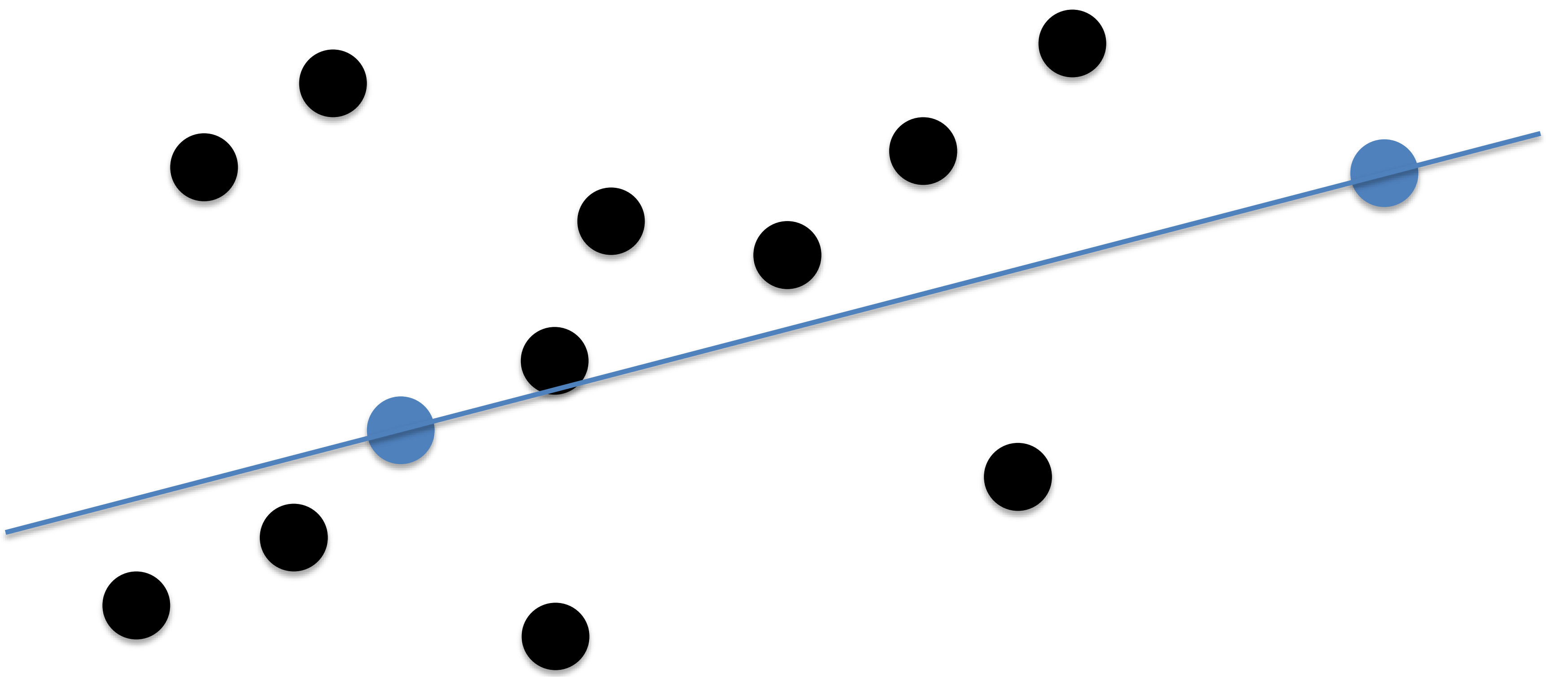
RANSAC for edges

Select two points



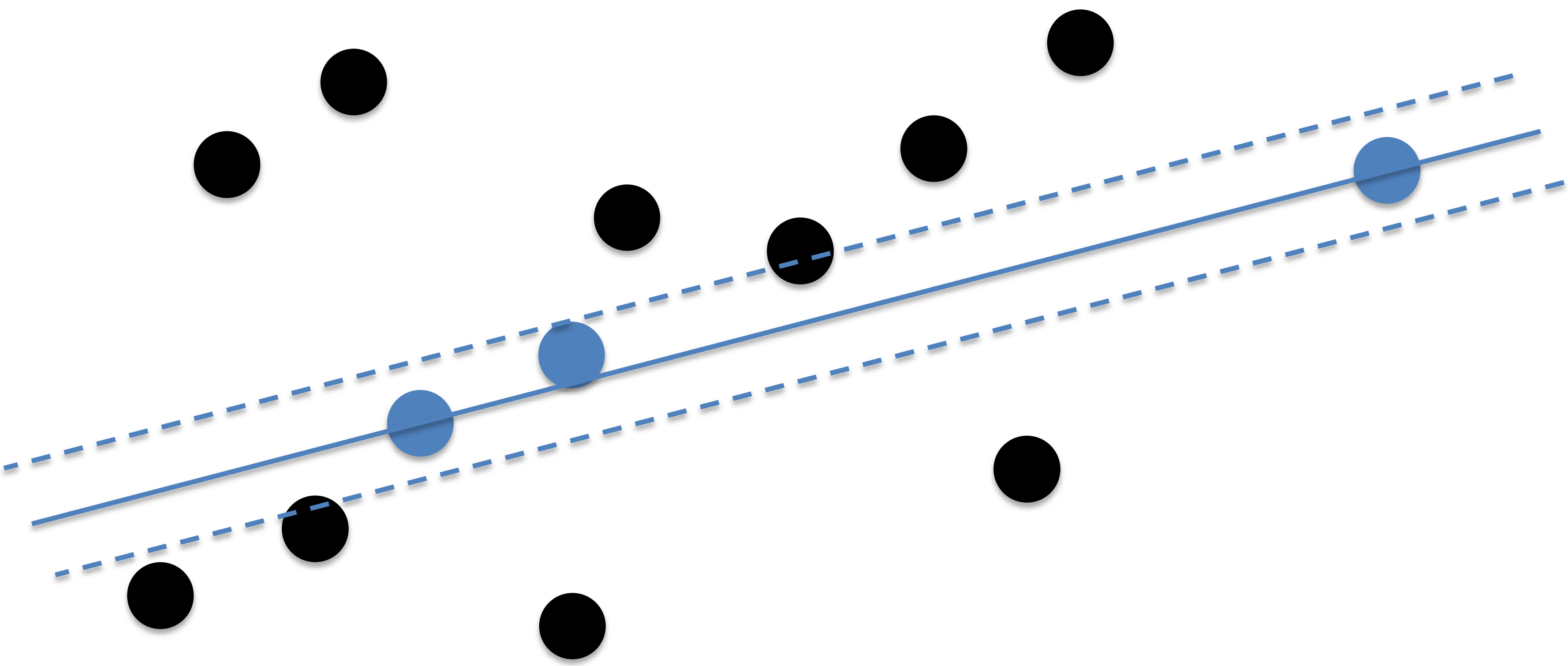
RANSAC for edges

Fit the line



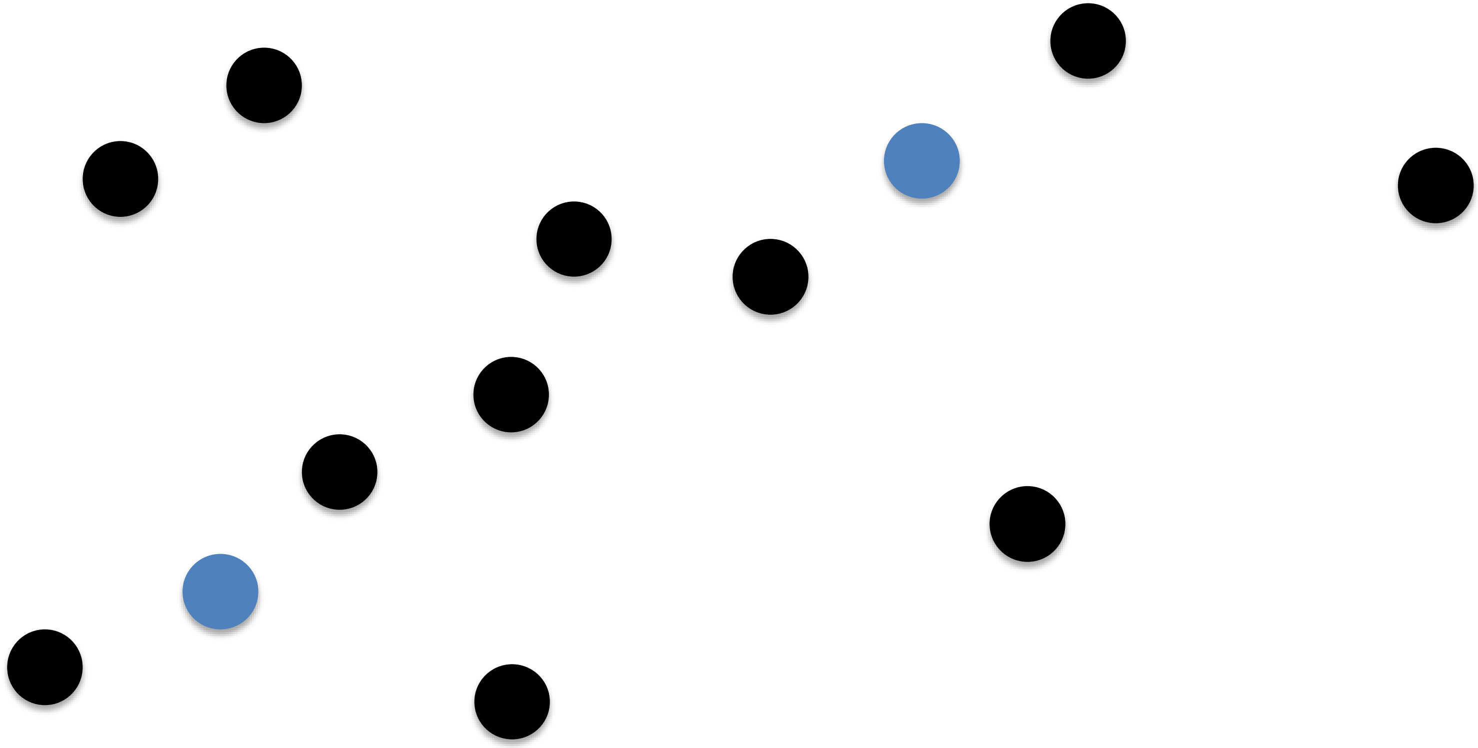
RANSAC for edges

Evaluate consensus (3 inliers)



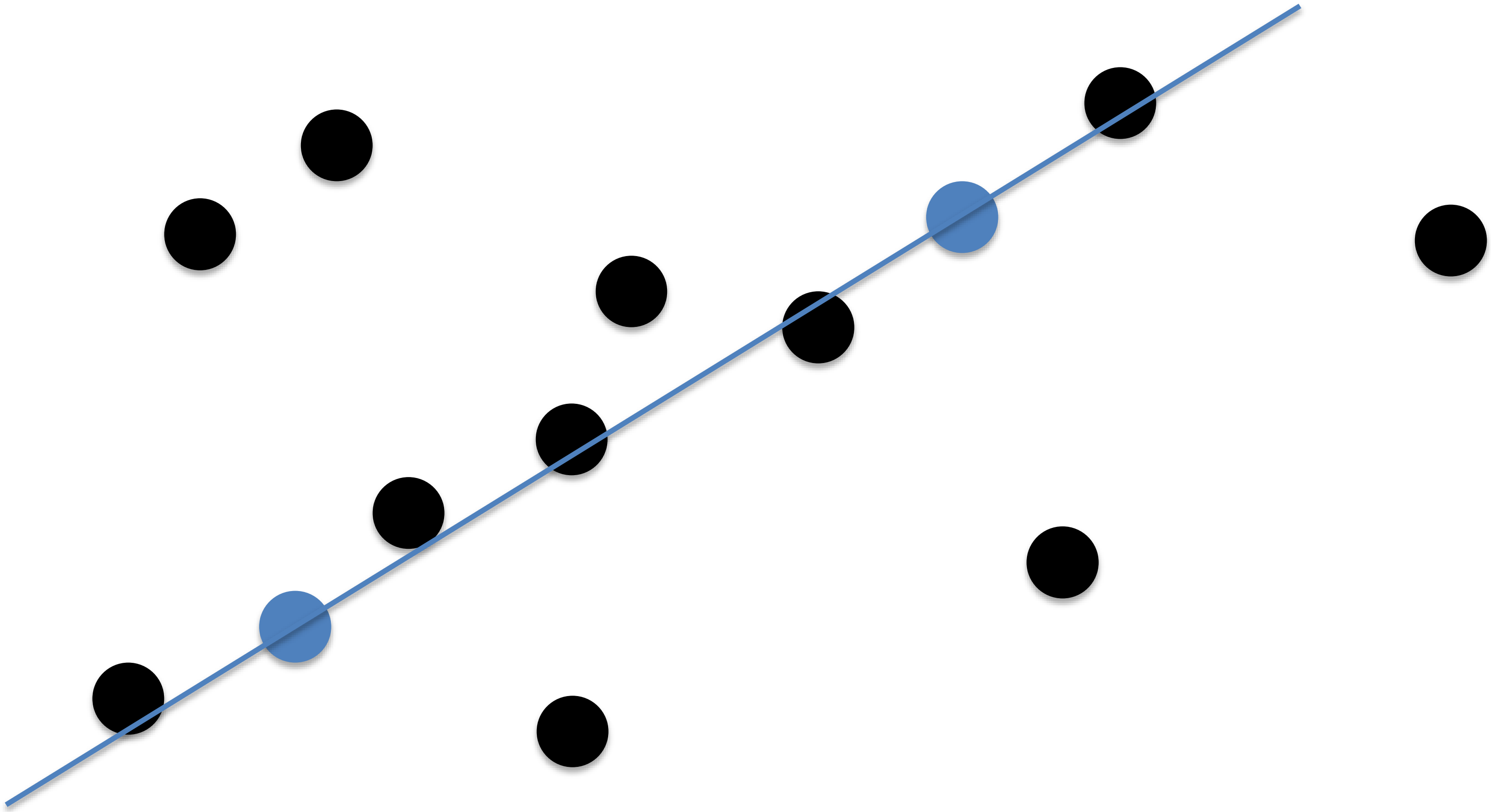
RANSAC for edges

Get other two points



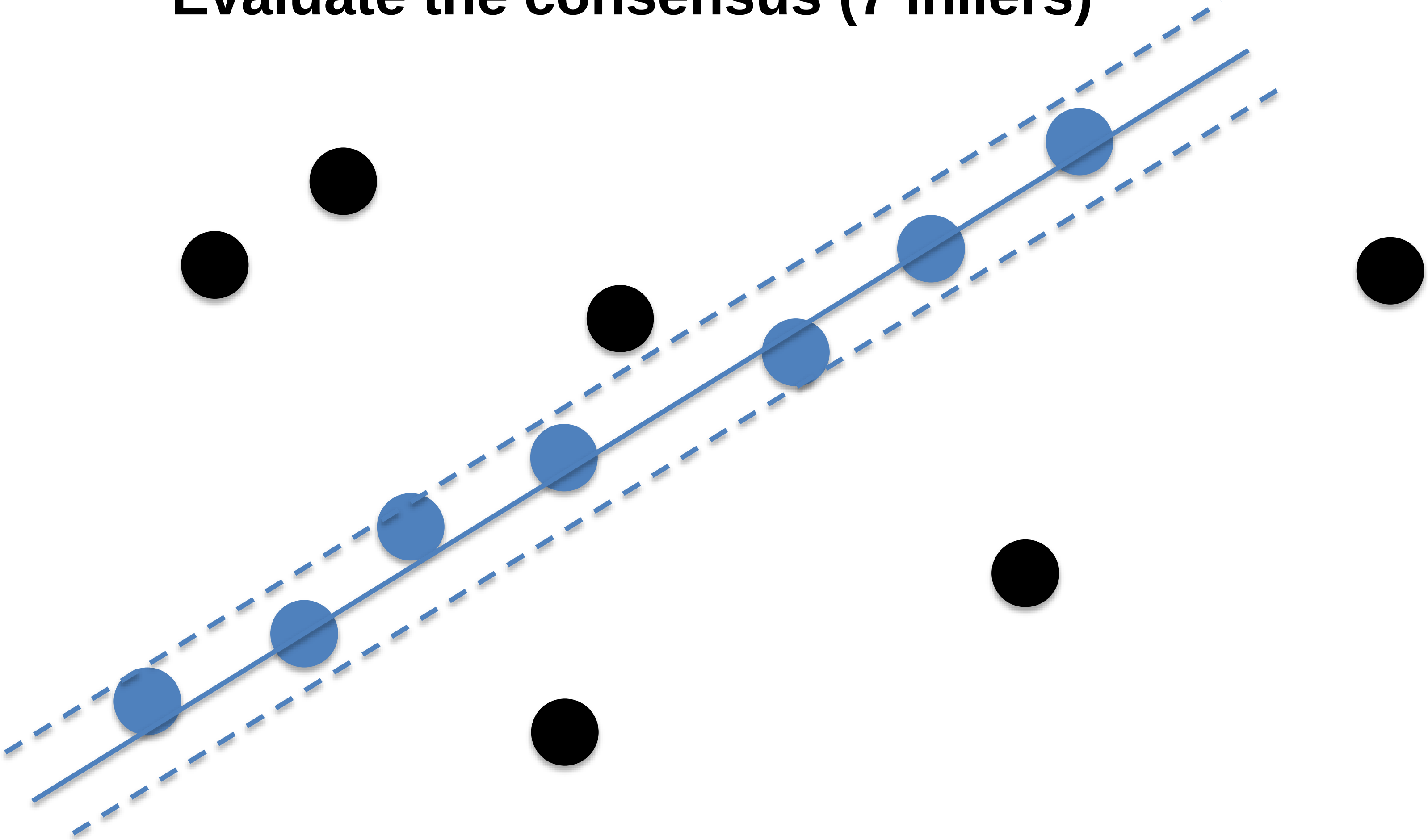
RANSAC for edges

Fit the line



RANSAC for edges

Evaluate the consensus (7 inliers)

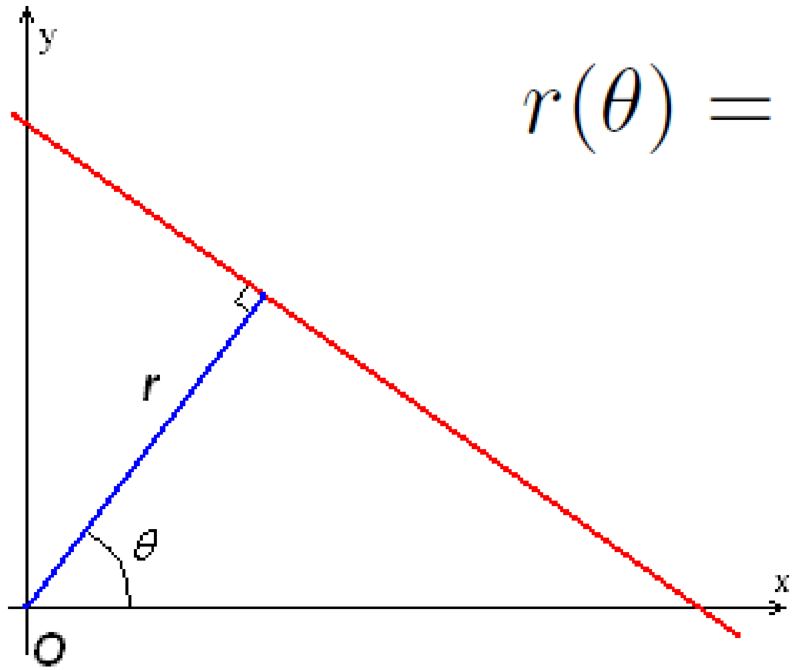


Hough Transform

- The *Hough transform* is a technique to detect features of a particular shape within an image.
- It requires that *the desired features be specified in some parametric form*.
- The *classical* Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc.*
- A *generalized* Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible (i.e. work using templates).

Hough Transform

- The idea is to pass from the image space to a parameter space; the *Hough space*.
- Lines are parameterized as:

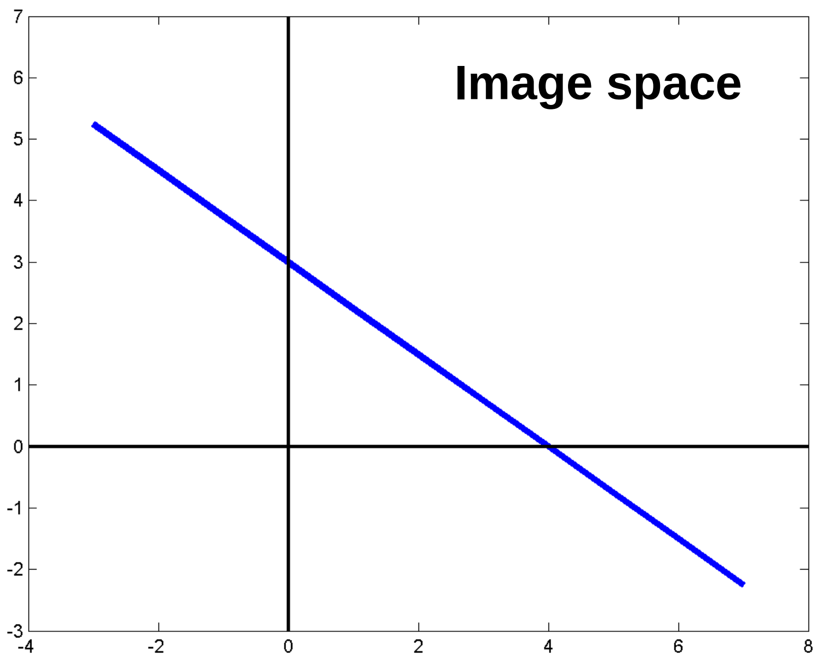


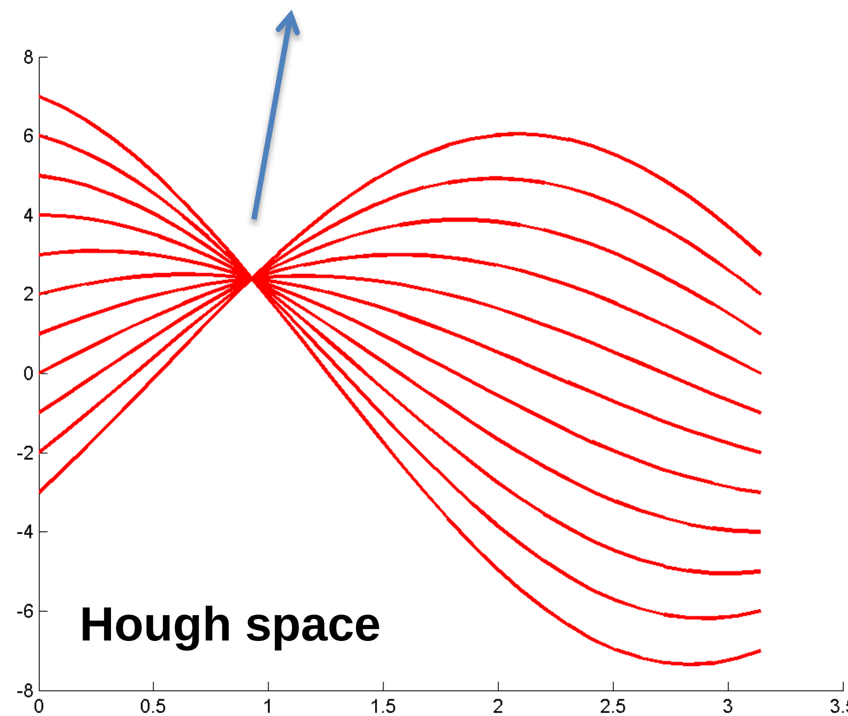
$$r(\theta) = x\cos\theta + y\sin\theta$$

Hough Transform

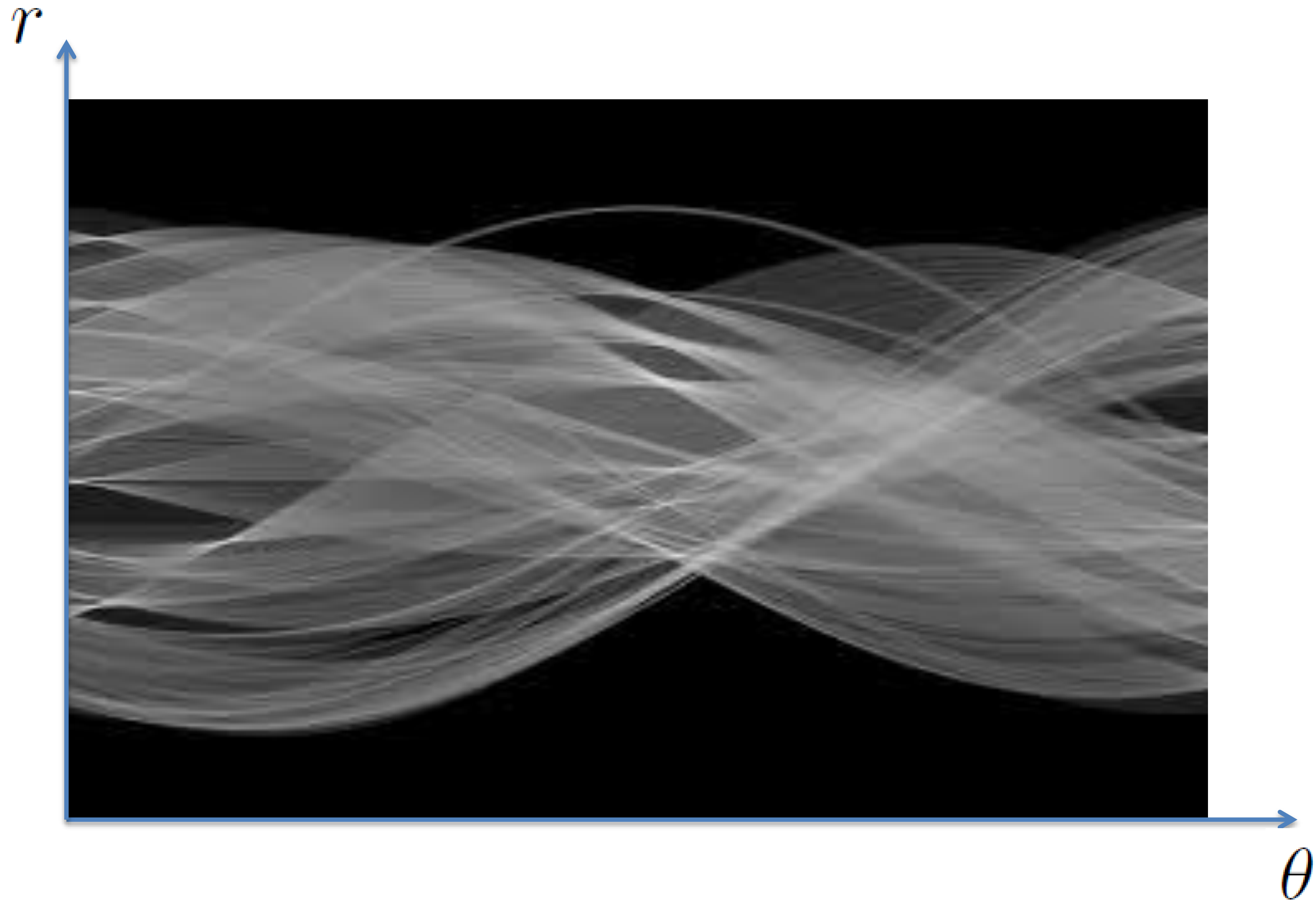
- Points *(x,y)* in image space corresponds to sinusoids in the Hough space.
- Points *(r,θ)* in Hough space corresponds to lines in image space.

Co-liner points intersect at an unique point.

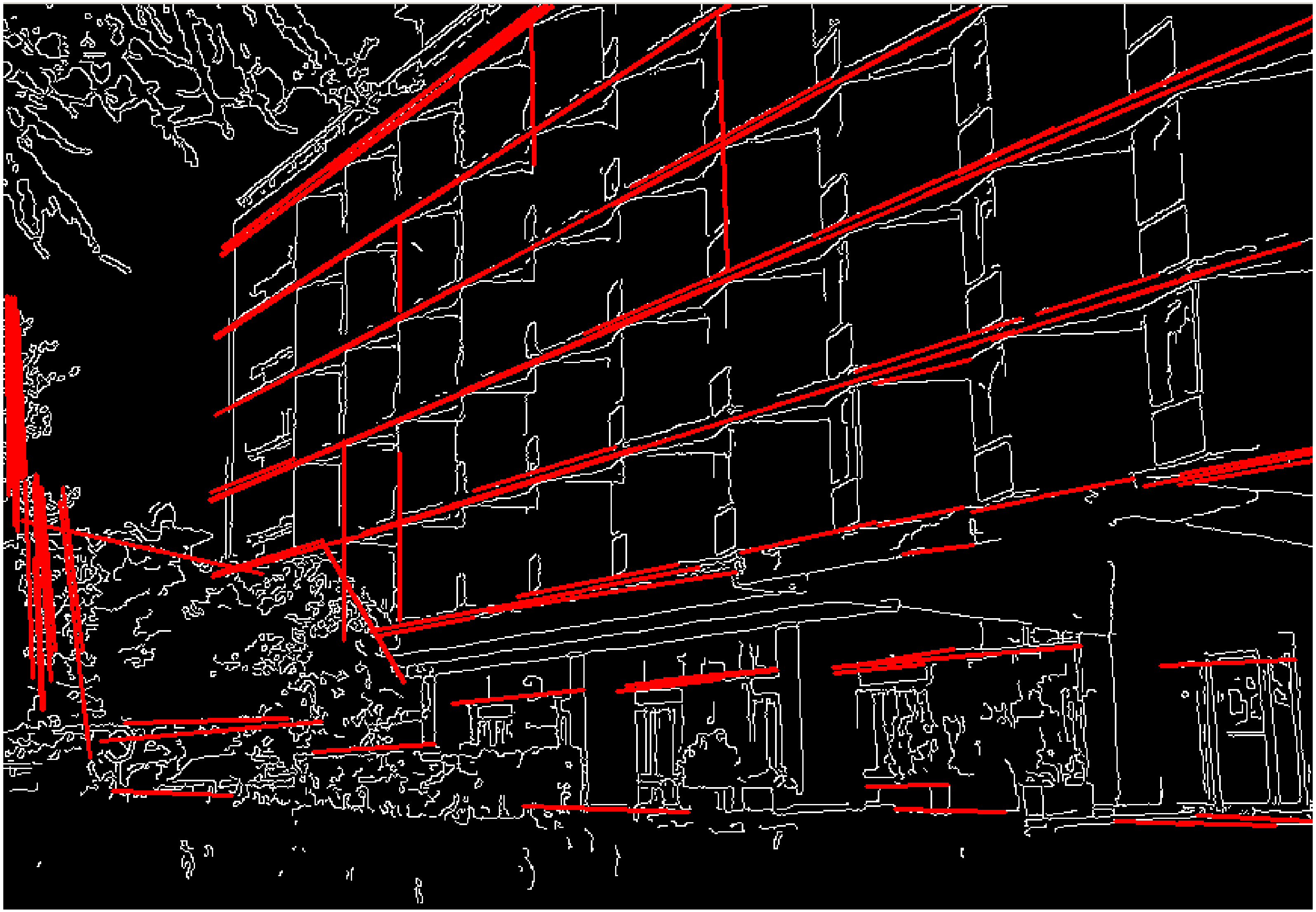




Hough Transform



Hough Transform

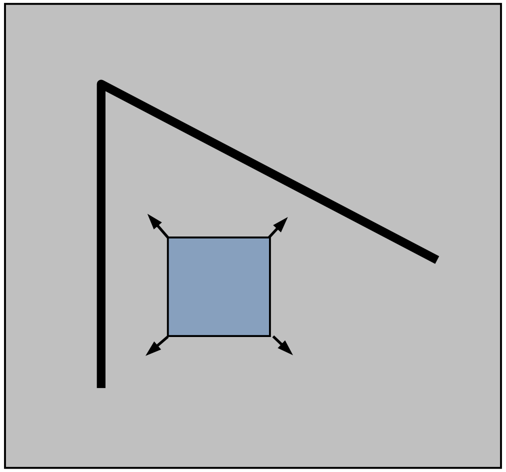


Hough Transform

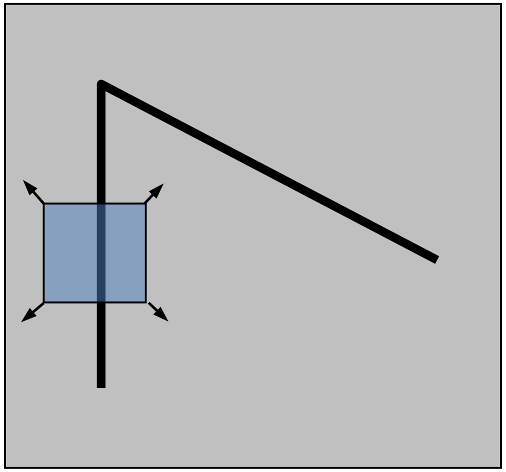
- In a nutshell: a voting scheme in a parameter space.
- Improvements:
  - Direction information (gradient)
  - Smoothing

Corners

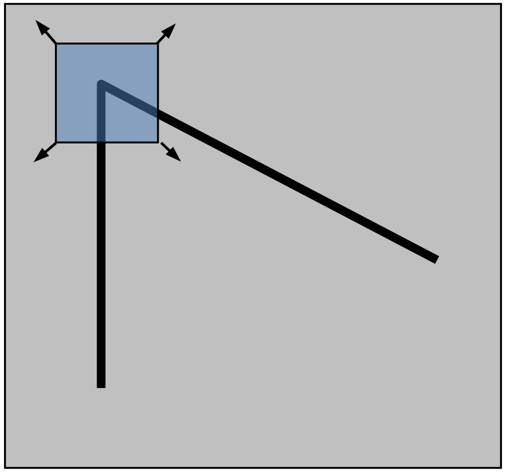
What is a “corner” ?



“flat” region:
no change in all
directions



“edge”:
no change along the
edge direction



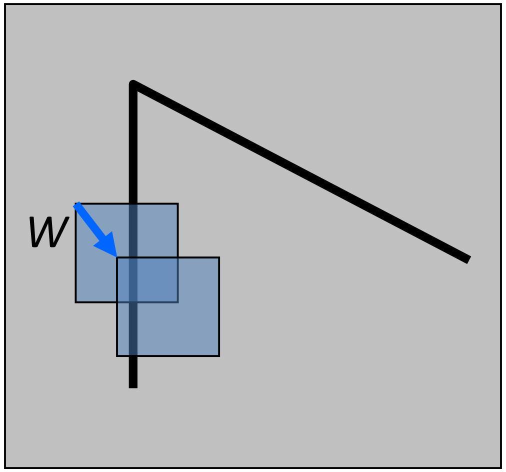
“corner”:
significant change in
all directions

Credit: S. Seitz, D. Frolova, D. Simakov

Harris Corner Detection

Consider shifting the window *W* by (*u*,*v*)

- how do the pixels in *W* change?
- compare each pixel before and after by summing up the squared differences (SSD)



$$E(u,v) \quad = \quad \sum_{(x,y)\in W} \left(I(x+u,y+v) - I(x,y)\right)^2$$

Small motion assumption

- Taylor expansion:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

- First-order approximation is good for small motion:

$$I(x+u, y+v) \approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x, y) + [I_x \;\; I_y] \left[\begin{array}{c} u \\ v \end{array} \right]$$

Harris Corner Detection

$$
\begin{aligned}
E(u,v) &= \sum_{(x,y)\in W} \left(I(x+u,y+v) - I(x,y)\right)^2 \\
&\approx \sum_{(x,y)\in W} \left(I(x,y) + I_x(x,y)u + I_y(x,y)v - I(x,y)\right)^2 \\
&\approx \sum_{(x,y)\in W} \left(I_x(x,y)u + I_y(x,y)v\right)^2 \\
&\approx \sum_{(x,y)\in W} \left(I_x^2u^2 + 2I_xI_yuv + I_y^2v^2\right)
\end{aligned}
$$

Harris Corner Detection

$$E(u,v) = \sum_{(x,y)\in W} \left(I(x+u,y+v) - I(x,y)\right)^2$$

$$\approx \sum_{(x,y)\in W} \left(I_x^2 u^2 + 2I_x I_y uv + I_y^2 v^2\right)$$

$$\approx Au^2 + 2Buv + Cv^2$$

$$A = \sum_{(x,y)\in W} I_x^2 \qquad B = \sum_{(x,y)\in W} I_x I_y \qquad C = \sum_{(x,y)\in W} I_y^2$$

Second Moment Matrix (*M*)

The surface *E*(*u*,*v*) is locally approximated by a quadratic form.

$$E(u,v) \approx Au^2 + 2Buv + Cv^2$$

$$\approx \begin{bmatrix} u & v \end{bmatrix} \underbrace{\begin{bmatrix} A & B \\ B & C \end{bmatrix}}_{\boldsymbol{M}} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = \sum_{(x,y)\in W} I_x^2 \qquad B = \sum_{(x,y)\in W} I_x I_y \qquad C = \sum_{(x,y)\in W} I_y^2$$

Harris Corner Detection

E(u,v) can be rewritten as:

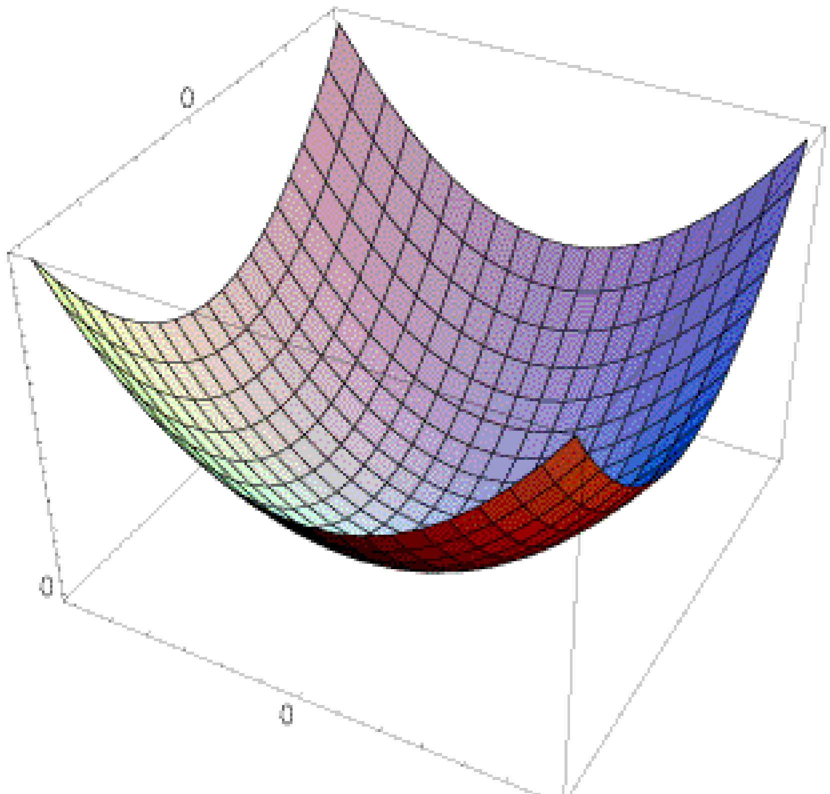
$$E(u,v) = \sum_{(x,y)\in W} [u \ v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Second Moment Matrix (*M*)

The surface $E(u,v)$ is locally approximated by a quadratic form.

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



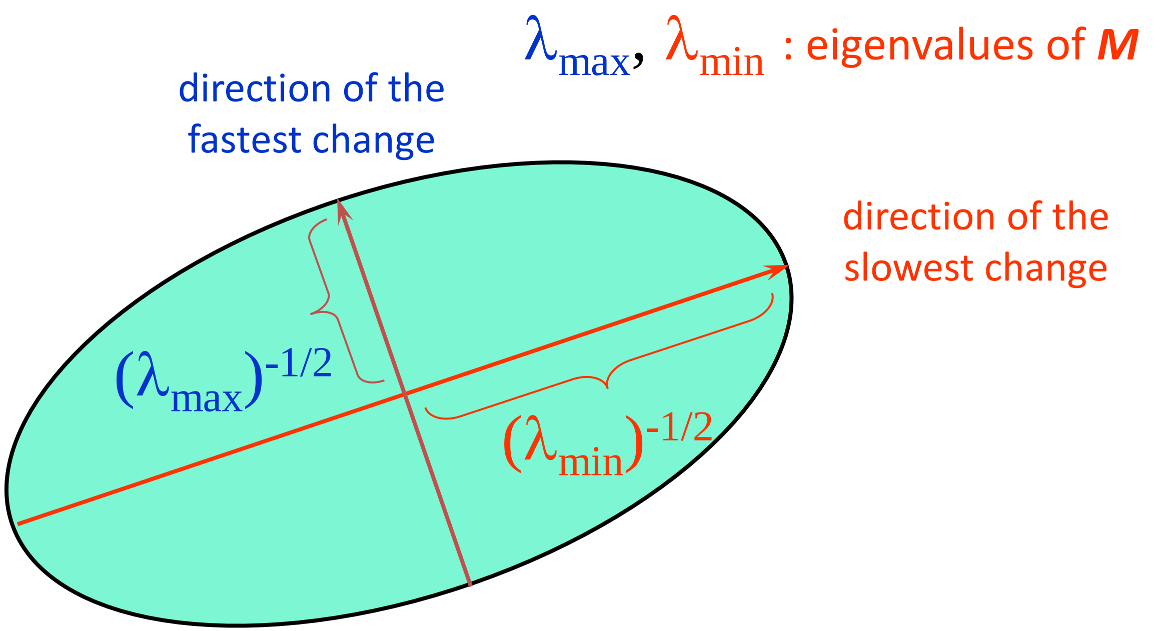
Second Moment Matrix (***M***)

The shape of ***M*** tells us something about the *distribution of gradients* around a pixel.

The *eigenvectors* of ***M*** identify the directions of fastest and slowest change.

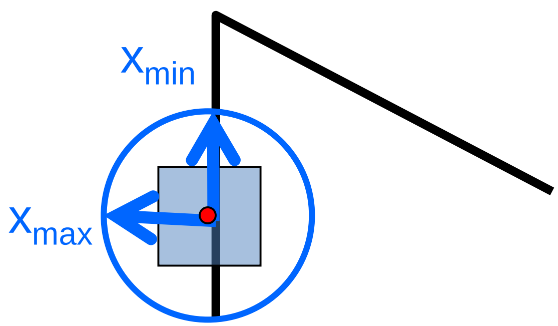
Ellipse equation:

$$[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$



Second Moment Matrix (*M*)

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} \underbrace{\begin{bmatrix} A & B \\ B & C \end{bmatrix}}_{\boldsymbol{M}} \begin{bmatrix} u \\ v \end{bmatrix}$$



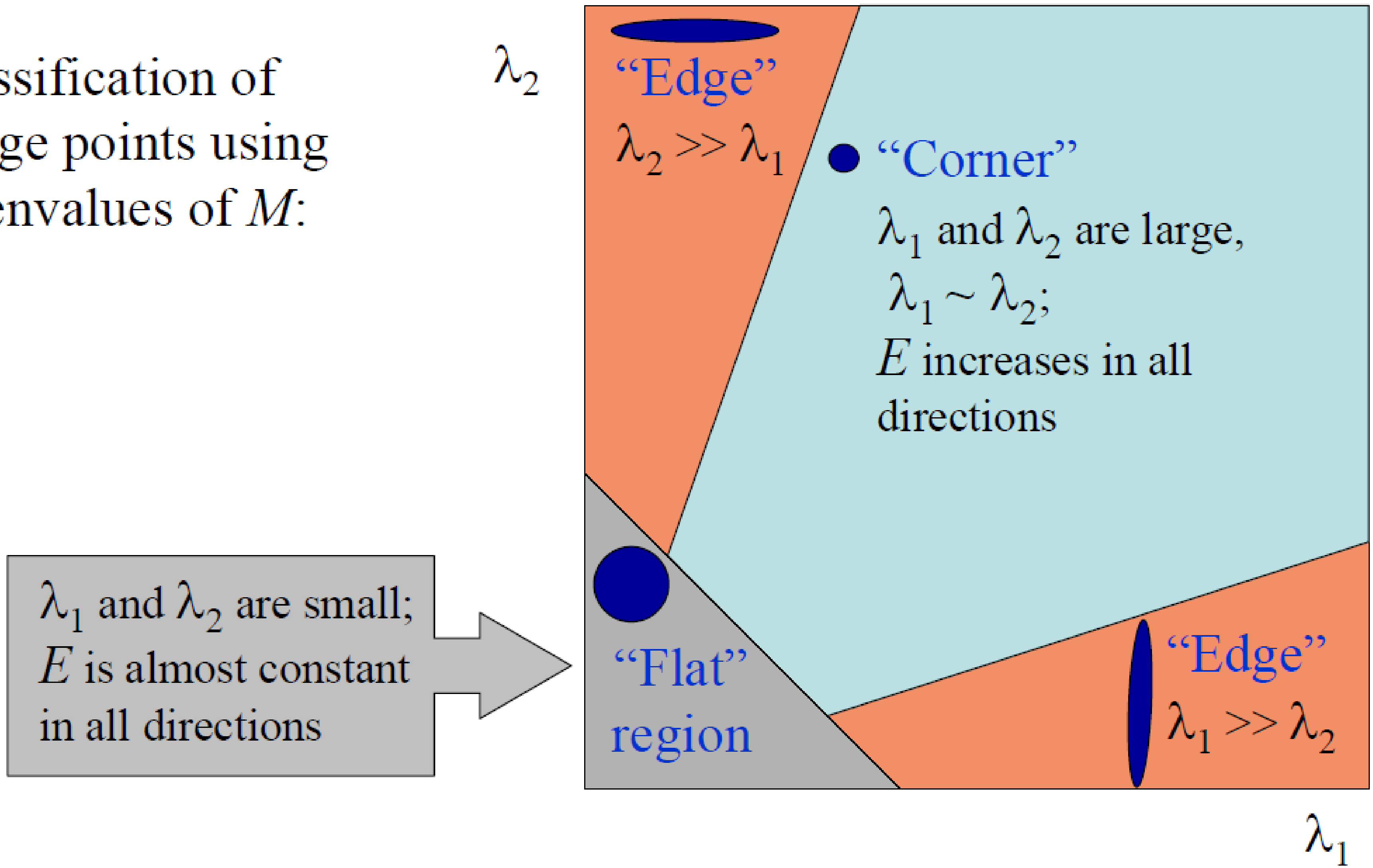
$$\begin{aligned} M x_{\max} &= \lambda_{\max} x_{\max} \\ M x_{\min} &= \lambda_{\min} x_{\min} \end{aligned}$$

Eigenvalues and eigenvectors of ***M***

- Define shift directions with the smallest and largest change in error
- x_{max} = direction of largest increase in *E*
- λ_{max} = amount of increase in direction x_{max}
- x_{min} = direction of smallest increase in *E*
- λ_{min} = amount of increase in direction x_{min}

Corners/Edges Classification

Classification of image points using eigenvalues of M:



Cornerness Measure

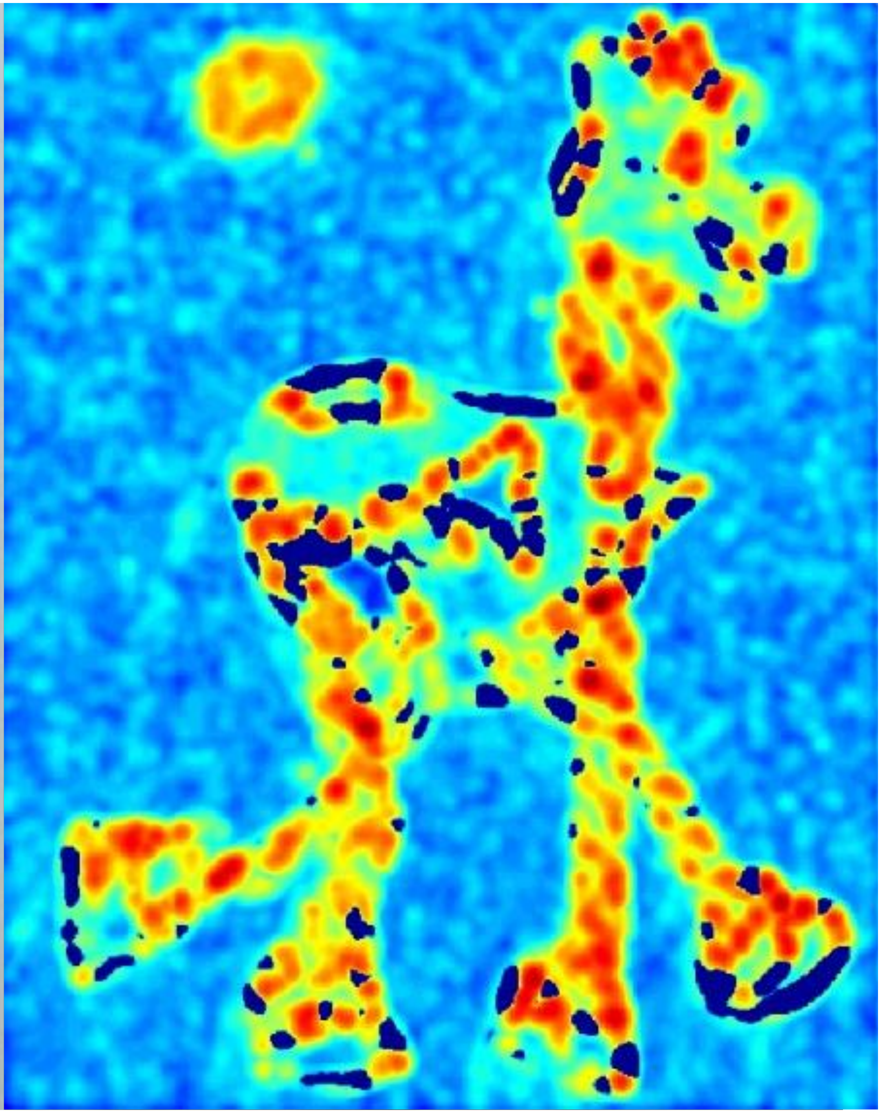
$$\boxed{R = \det(M) - k\operatorname{Tr}(M)^2}$$

$$\det(M) = \lambda_1 \lambda_2$$

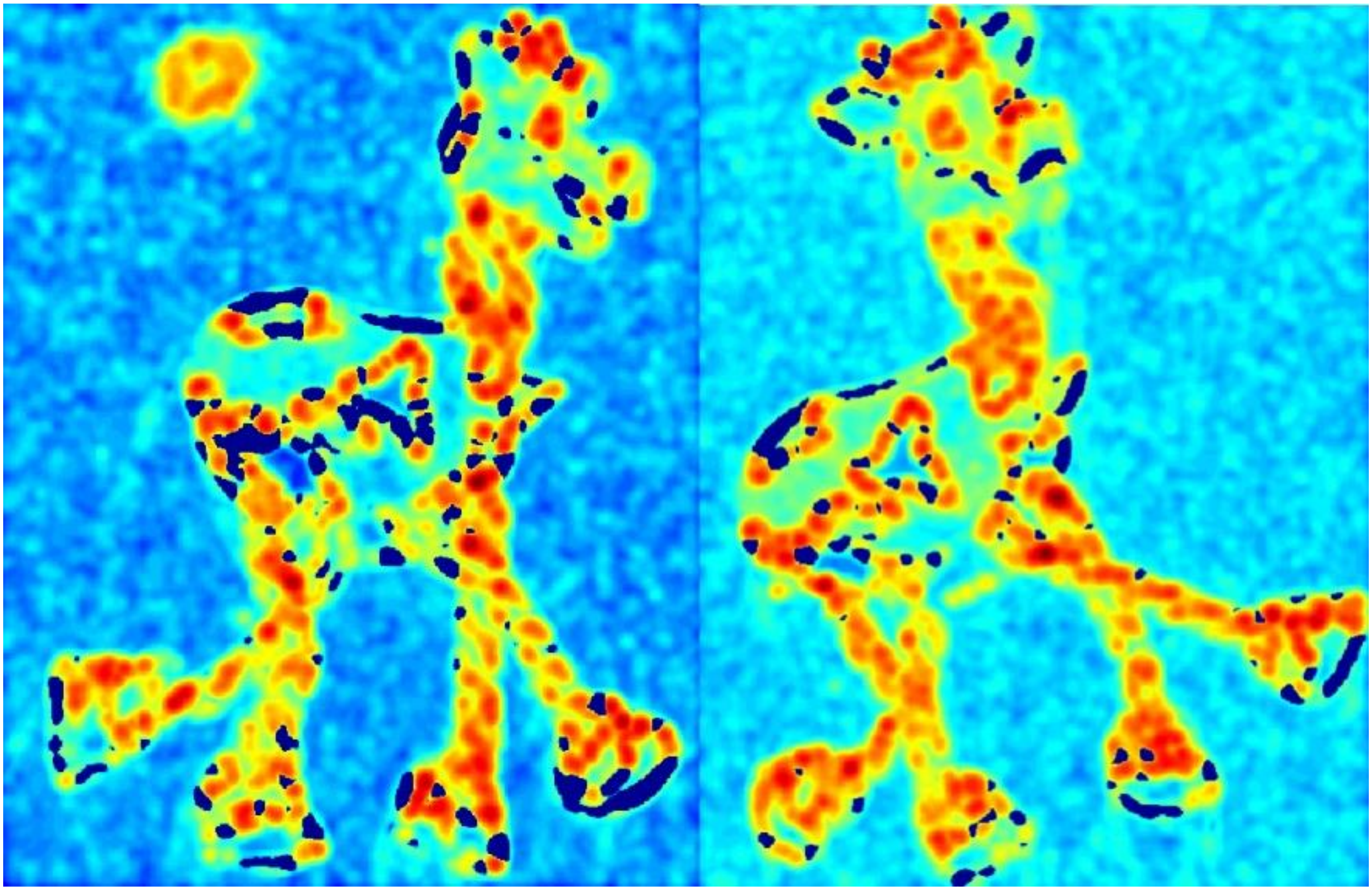
$$\operatorname{Tr}(M) = \lambda_1 + \lambda_2$$

Harris Detector – example

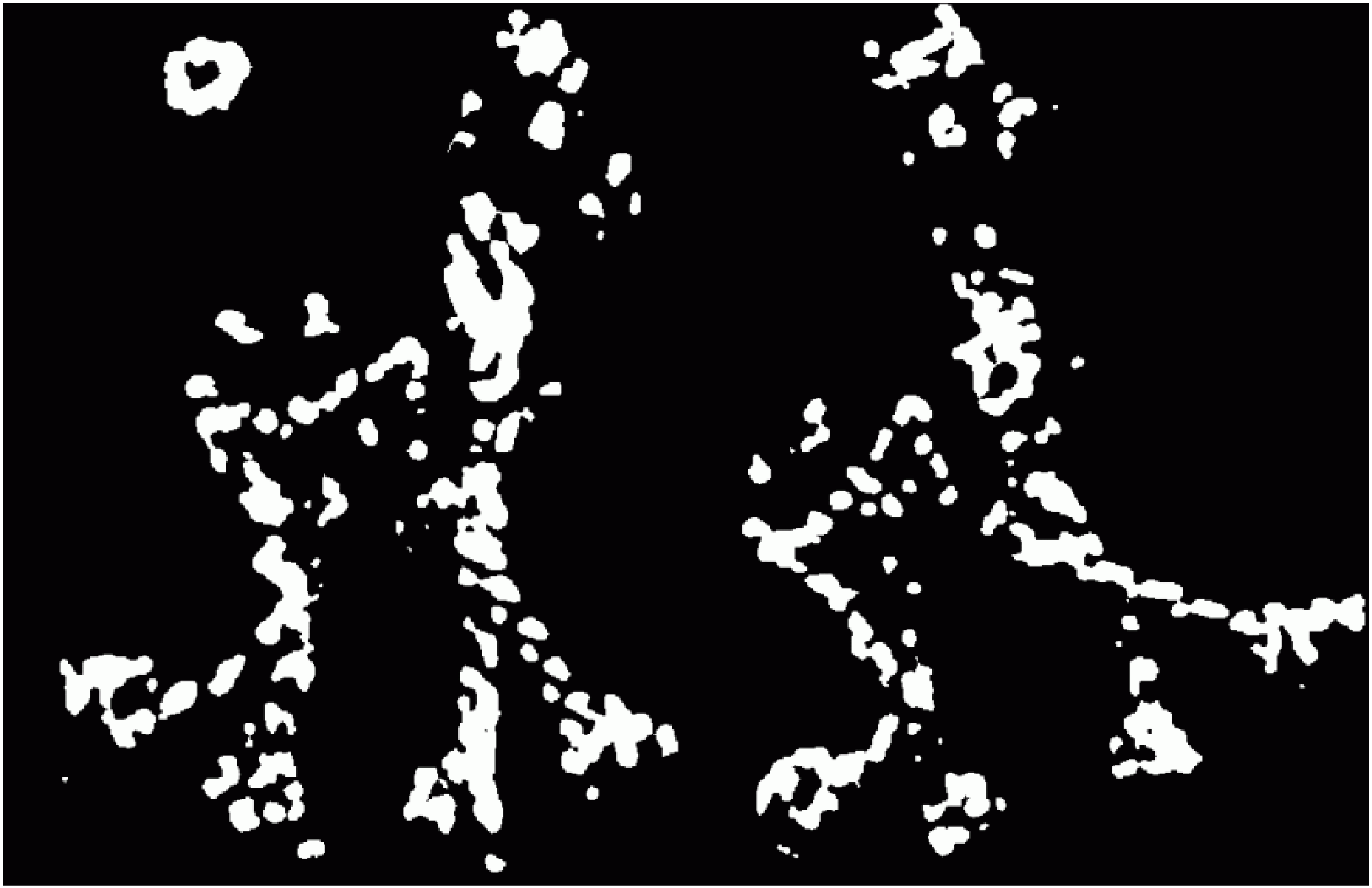
Harris Detector – example



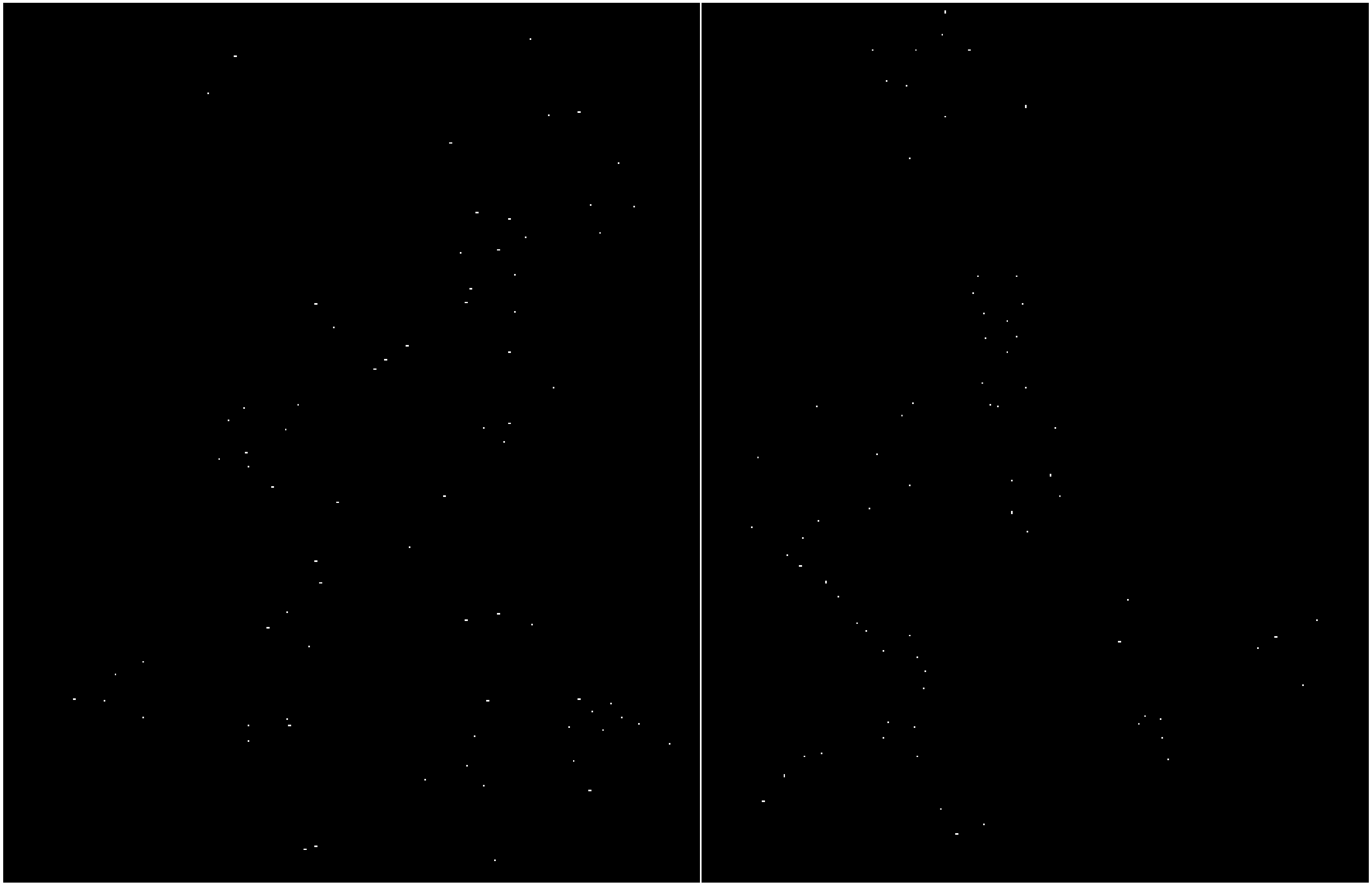
Harris Detector – example



Harris Detector – threshold



Harris Detector – local maxima



Harris Detector – corners in red

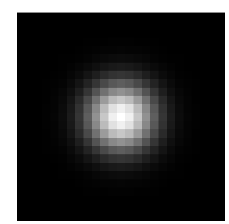
Weighting the derivatives

- In practice, using a simple window *W* does not work well (noisy response):

$$M = \sum_{(x,y \in W)} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- It is better to *weight* each derivative value based on its distance from the center pixel:

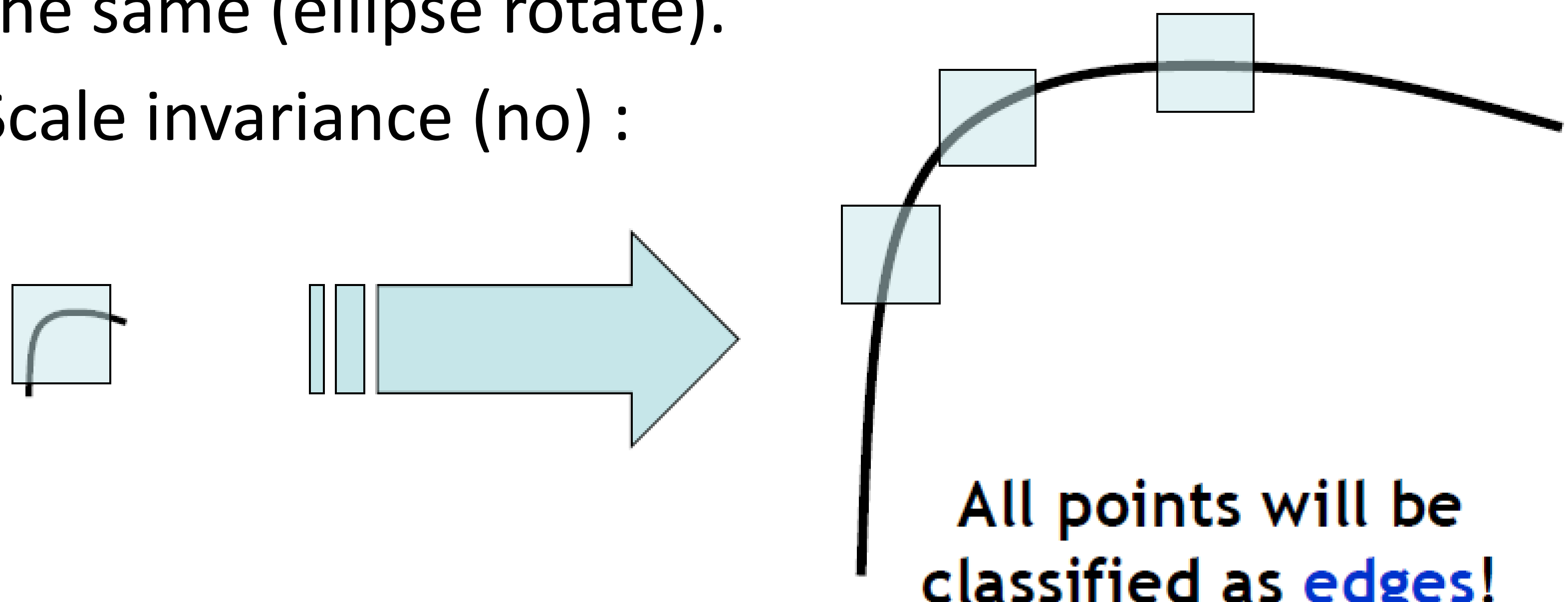
$$M = \sum_{(x,y \in W)} w_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



$w_{x,y}$

Harris Corner Detection -Robustness

- Partially invariant to intensity changes *(I'(x,y) = a I(x,y) + b)*
- Translation invariance (yes) → corner location is covariant.
- Rotational invariance (yes) → eigenvalues remains the same (ellipse rotate).
- Scale invariance (no) :

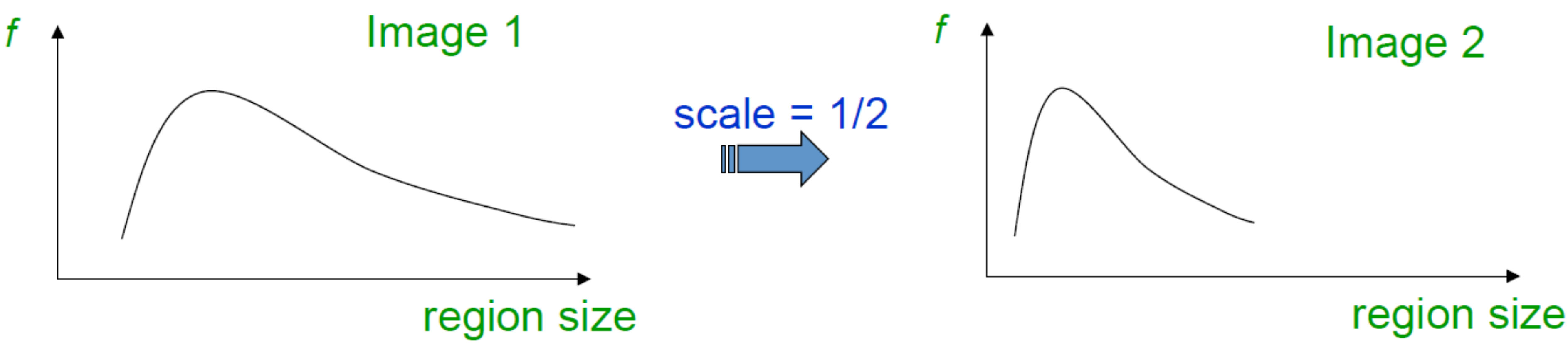


Scale Invariant Detection

- The same feature in different images can have different size.
- ***Goal:*** find a function of the region size which responds equally for corresponding regions of different size.

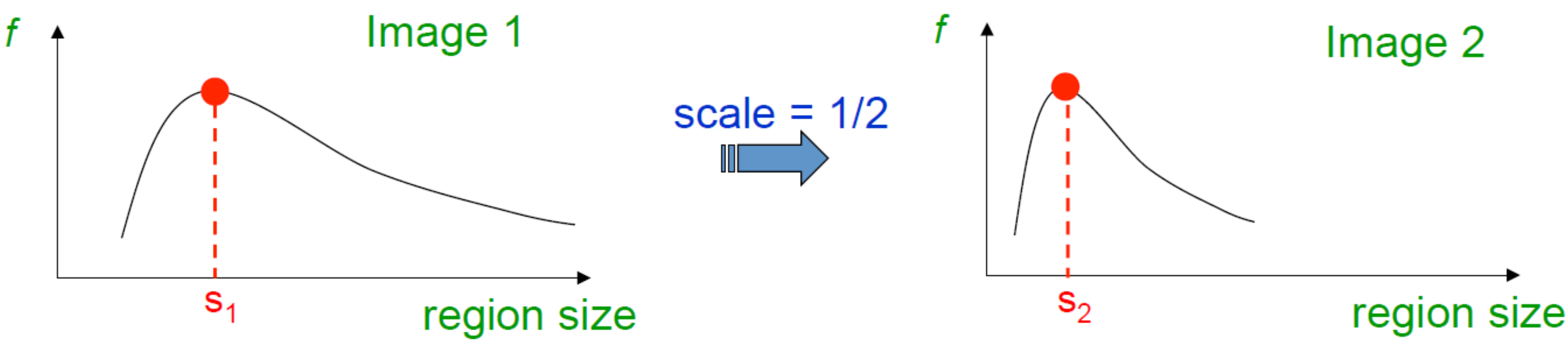
Scale Invariant Detection

- For a point in one image, we can consider it as a function of the region size (e.g. radius of a circle):



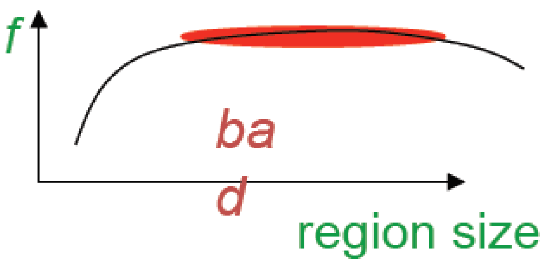
Scale Invariant Detection

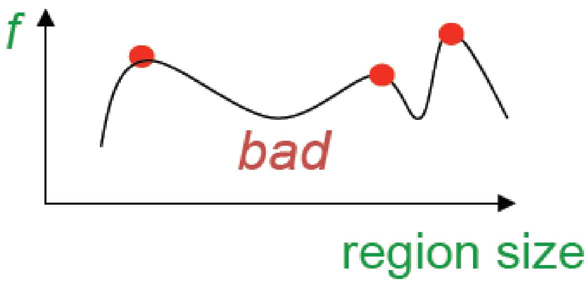
- It is common to consider the local maximum of such function → corresponding region size should be covariant with image scale.



Scale Invariant Detection

- A good function for scale detection should have a peak easily identifiable:





Scale Invariant Detection

- Common ***characteristic scale functions*** can be defined by composing the derivatives of the images convolved with Gaussians of different size.

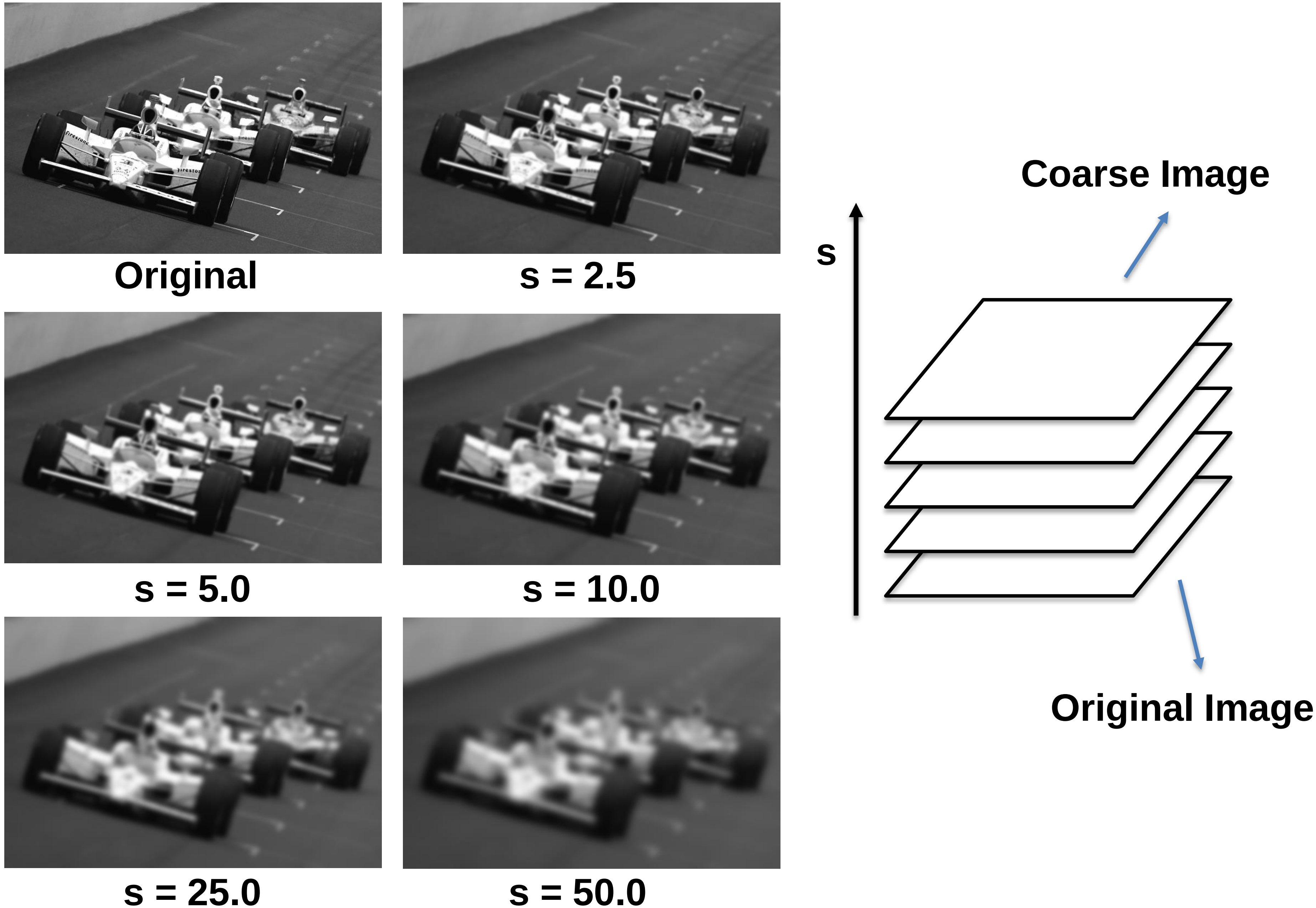
Scale Space Image Representation

- A family of images convolved with a Gaussian kernel of different size.

$$L(x, y; s) = G(x, y; s) * I(x, y)$$

$$G(x, y; s) = \frac{1}{2\pi s} e^{-(x^2+y^2)/2s}$$

(note that $s = \sigma^2$)



Original
s = 2.5
s = 5.0
s = 10.0
s = 25.0
s = 50.0
s
Coarse Image
Original Image

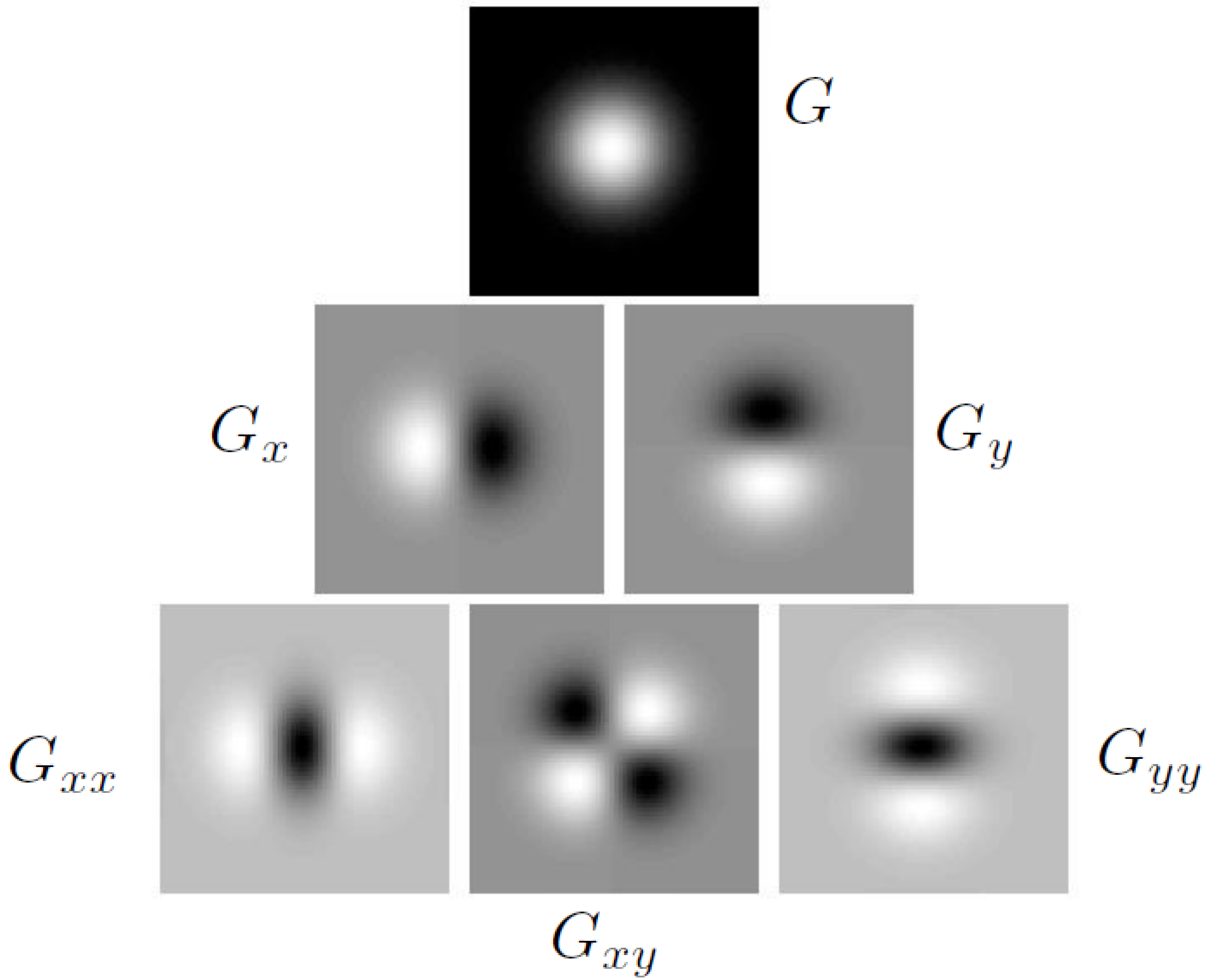
Why Gaussian Kernels ?

- Gaussian kernels satisfies *scale-space axioms*.
- Derivatives of the scale space can be easily obtained through convolution with Gaussian derivatives:

$$\partial_{x^n y^m} L(x, y; s) = (\partial_{x^n y^m} G(x, y; s)) * I(x, y)$$

$$L_{x^n y^m}(x, y; s)$$

Scale Space Image Representation



Laplacian of Gaussian (LoG)

- The *Laplacian* of an image *I(x,y)* can be used to detect edges (zero-crossing).

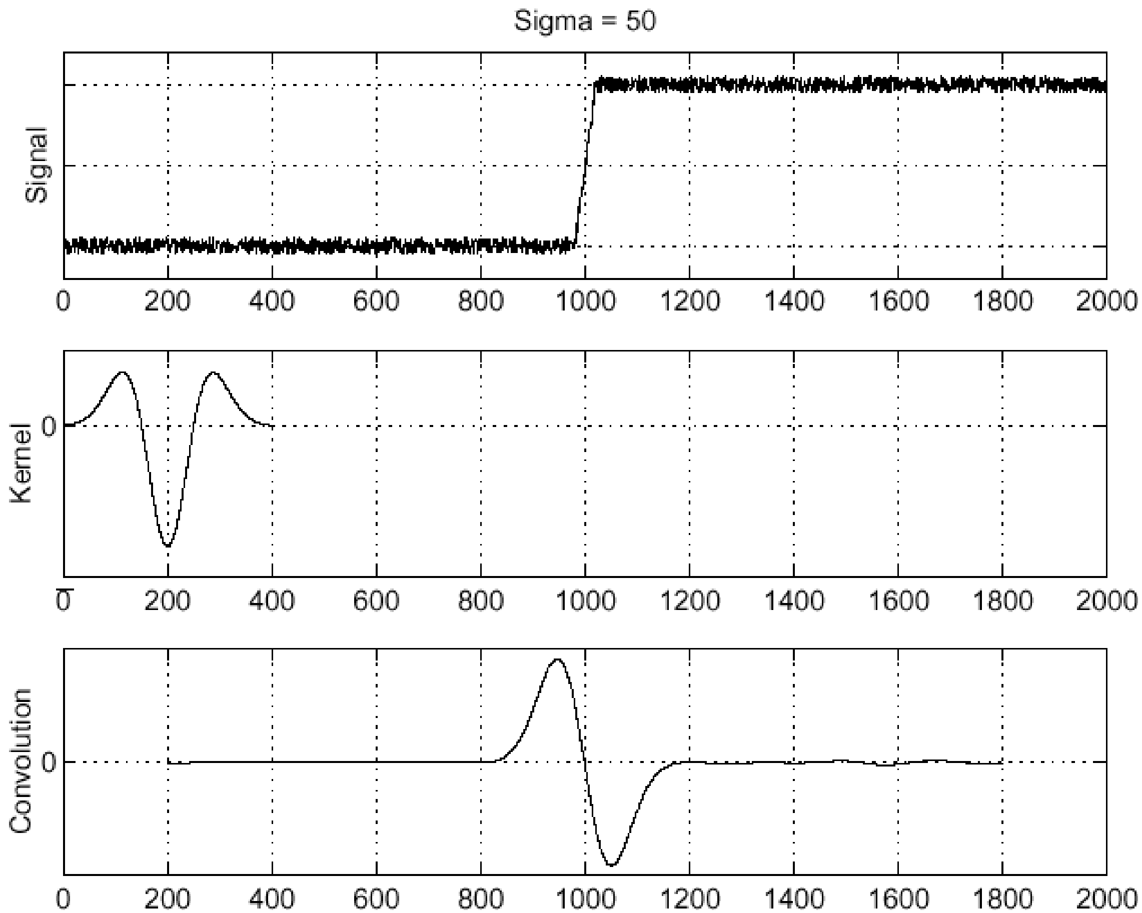
$$\nabla^2 I(x,y) = \frac{\partial^2 I}{\partial_x^2} + \frac{\partial^2 I}{\partial_y^2}$$

- In a scale-space representation this can be calculated as

$$\nabla^2 L = L_{xx} + L_{yy}$$

Laplacian of Gaussian

Laplacian of Gaussian (LoG)



f — Edge

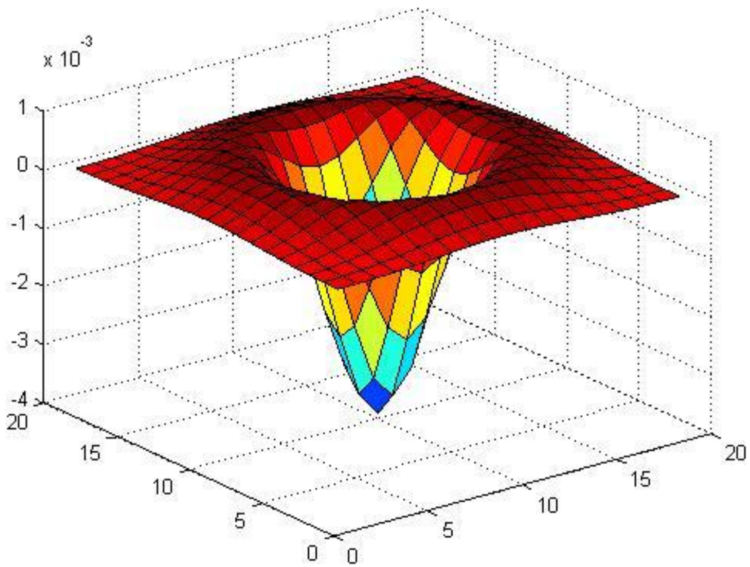
$\frac{d^2}{dx^2} g$ — Second derivative of Gaussian (Laplacian)

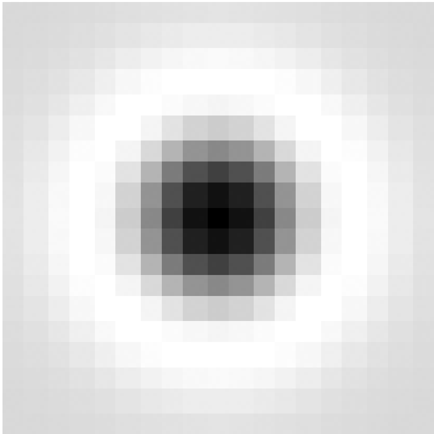
$f * \frac{d^2}{dx^2} g$ — Edge → zero crossing of second derivative

Source: S. Seitz

Laplacian of Gaussian (LoG)

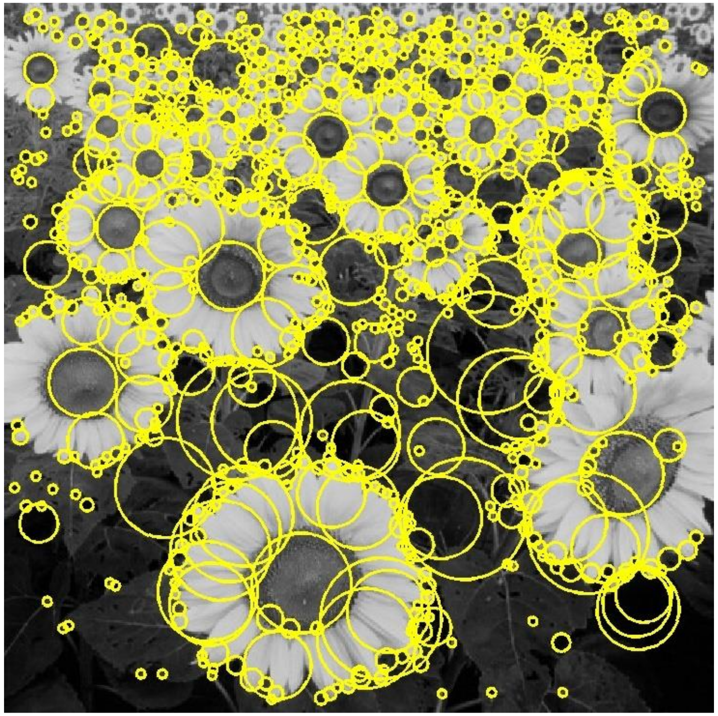
The LoG can be used also as a ***blob detector***.

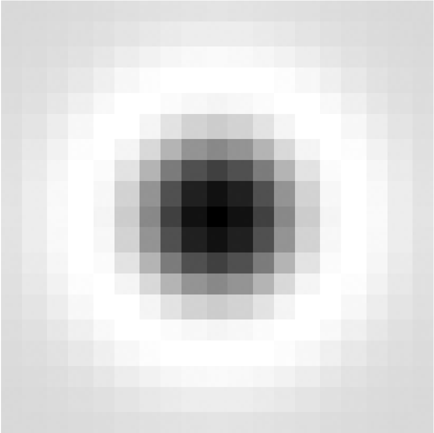




Blob Detection at Multiple Scale

- The local extrema of the LoG response in the scale-space representation can be used to detected “blob” region at different scale.





Automatic Scale Selection

- A scale-space representation can be made invariant to scales, by performing automatic local scale selection by using *γ-normalized derivatives**:

$$\partial_\xi = s^{\gamma/2}\partial_x \ , \ \partial_\eta = s^{\gamma/2}\partial_y$$

$$\partial_{\xi^n \eta^m} L(x, y; s) = s^{(m+n)\gamma/2} L_{x^n y^m}(x, y; s)$$

T. Lindeberg, "Feature Detection with Automatic Scale Selection",
Int. Journal of Computer Vision, **Vol. 30(2), pp. 79-116, 1998.**

Automatic Scale Selection

- The scale at which a *scale-normalized differential entity* assumes a local extremum over scales is proportional to the size of the corresponding image structure in the image domain.

$$\nabla^2 L = L_{xx} + L_{yy}$$

LoG

$$\det H = L_{xx} L_{yy} - L_{xy}^2$$

Determinant of the *Hessian matrix*

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$$

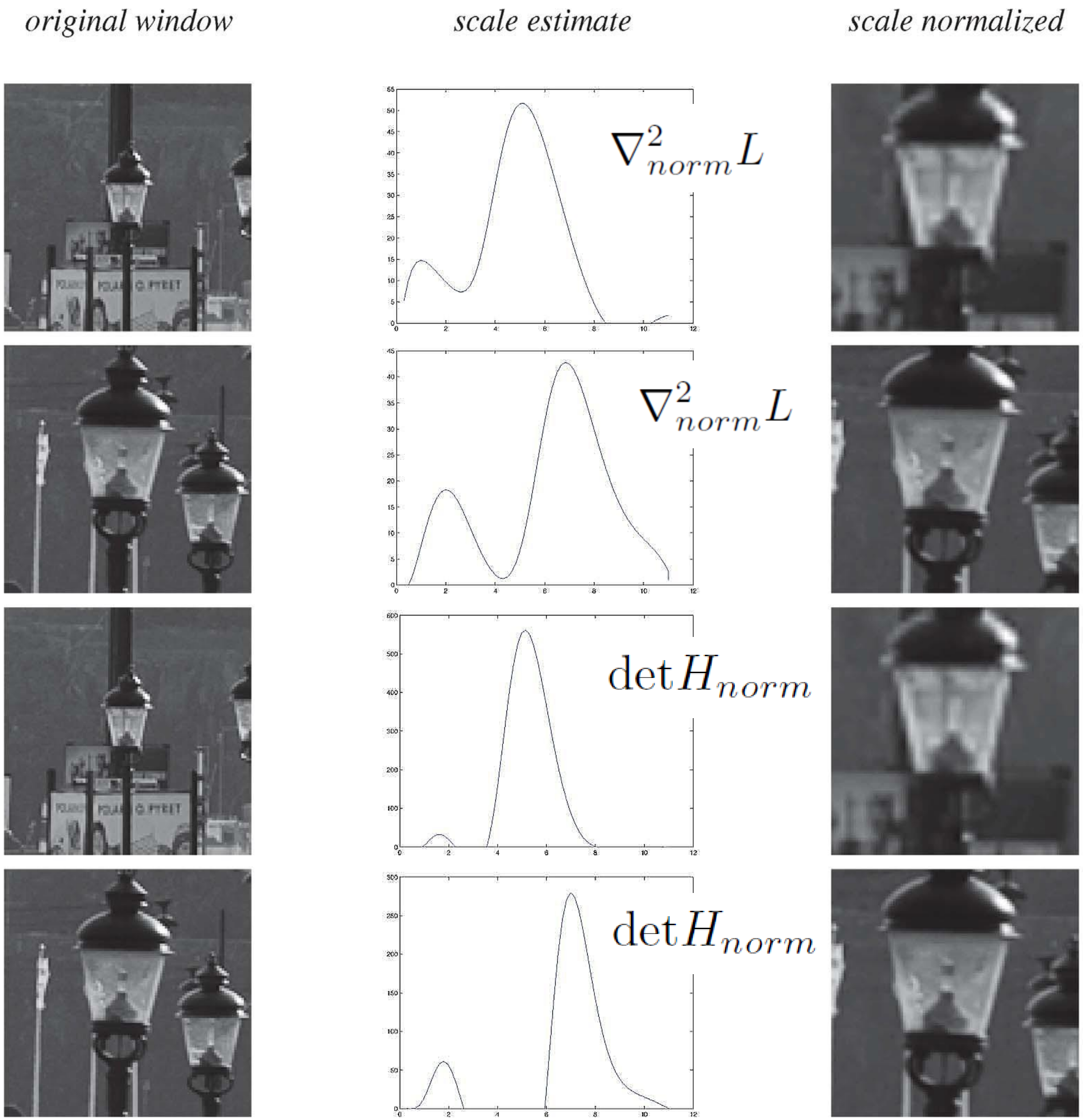
Automatic Scale Selection

Scale-normalized version

$$\nabla^2 L \rightarrow \nabla^2_{norm} L = s\left(L_{xx} + L_{yy}\right)$$

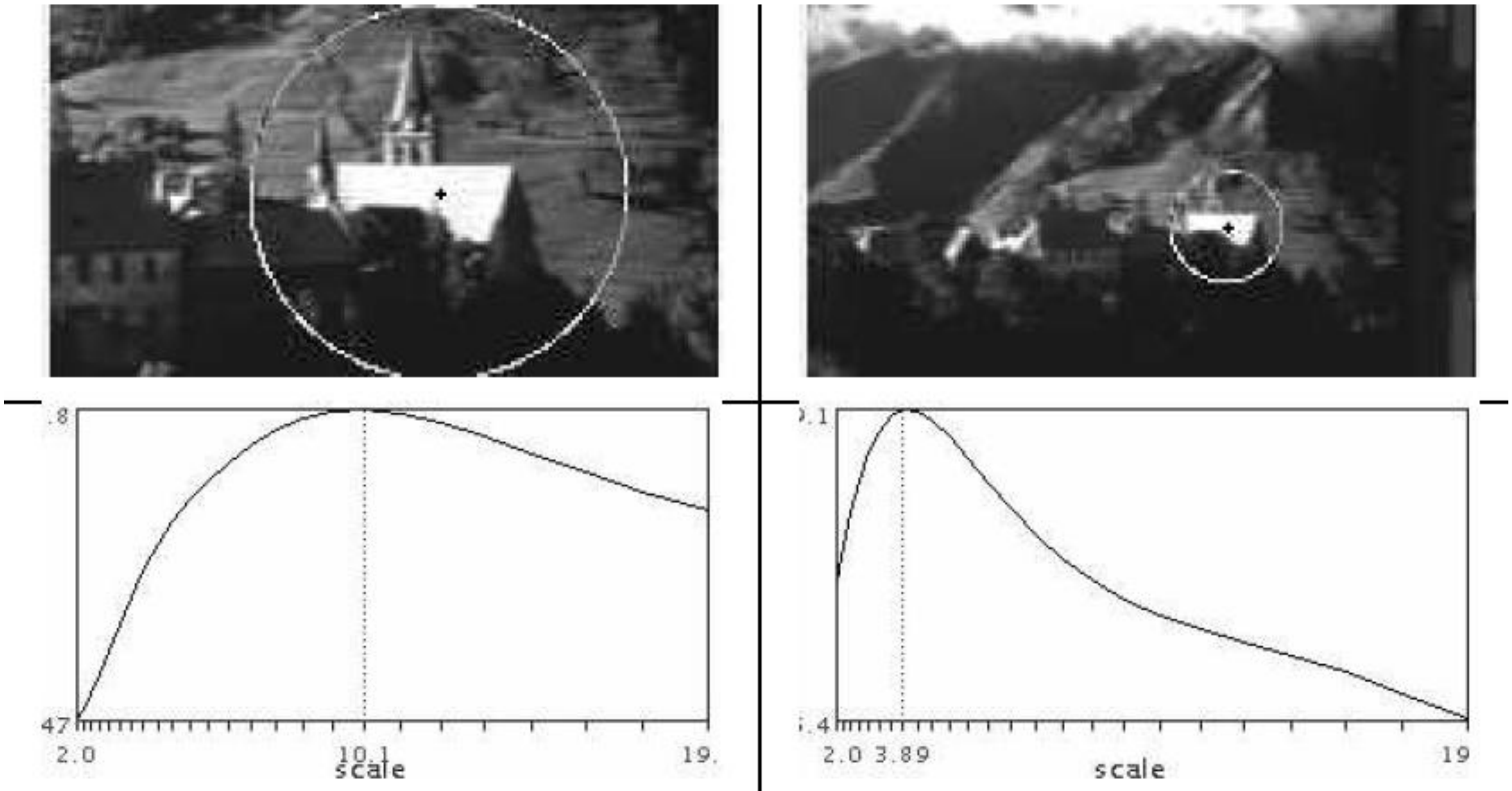
$$\det H \rightarrow \det H_{norm} = s^2 (L_{xx} L_{yy} - L_{xy}^2)$$

(these are an example of the characteristic scale function mentioned before)



Lindeberg T (2013) "Invariance of visual operations at the level of receptive Fields" ***PLoS ONE 8(7): e66990. doi:10.1371/journal.pone.0066990.***

Scale Invariant Harris Detector



K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", *Proc. of ICCV 2001*, 2001.

Scale Inv. Harris Detector (Harris-Laplace)

- Scale-adapted Harris Corner detector:

$$\mathbf{C}(x, y, s, \bar{s}) = s^2 G(x, y, \bar{s}) * \begin{bmatrix} L_x^2(x, y, s) & L_x L_y(x, y, s) \\ L_x L_y(x, y, s) & L_y^2(x, y, s) \end{bmatrix}$$

$$\boxed{R = \det(\mathbf{C}) - k\mathrm{Tr}(\mathbf{C})^2}$$

- Local extrema of LoG is used for scale selection.

K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", *Proc. of ICCV 2001*, 2001.

Harris-Laplace Detector

K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", ***Proc. of ICCV 2001*****, 2001.**

Scale Invariant Features Transform (SIFT*)

- Presented by Lowe* in 2004 (patented).
- Rotation and Scale invariant
- Robust to
  – Viewpoint change
  – Illumination change
- It is one of the most used detector/descriptor.

David G. Lowe, "Distinctive image features from scale-invariant keypoints",** ***Int. J. of Computer Vision, 60(2), pp. 91-110**, 2004.**

SIFT – Stages of Computation

- Find Scale-Space Extrema
- Keypoint Localization & Filtering
  - Improve keypoints and throw out bad ones

Detector

- Orientation Assignment
  - Remove effects of rotation and scale
- Create descriptor
  - Using histograms of orientations

Descriptor

Detection of Scale Space Extrema

- Stable keypoints are localized using local extrema of *Difference of Gaussian (DoG)* function convolved with the image *I*.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

DoG and LoG

- *D(x,y,σ)* is efficient to compute and it is an approximation of the normalized Laplacian of Gaussian:

From the heat diffusion equation: $$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$$

We can write:

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

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

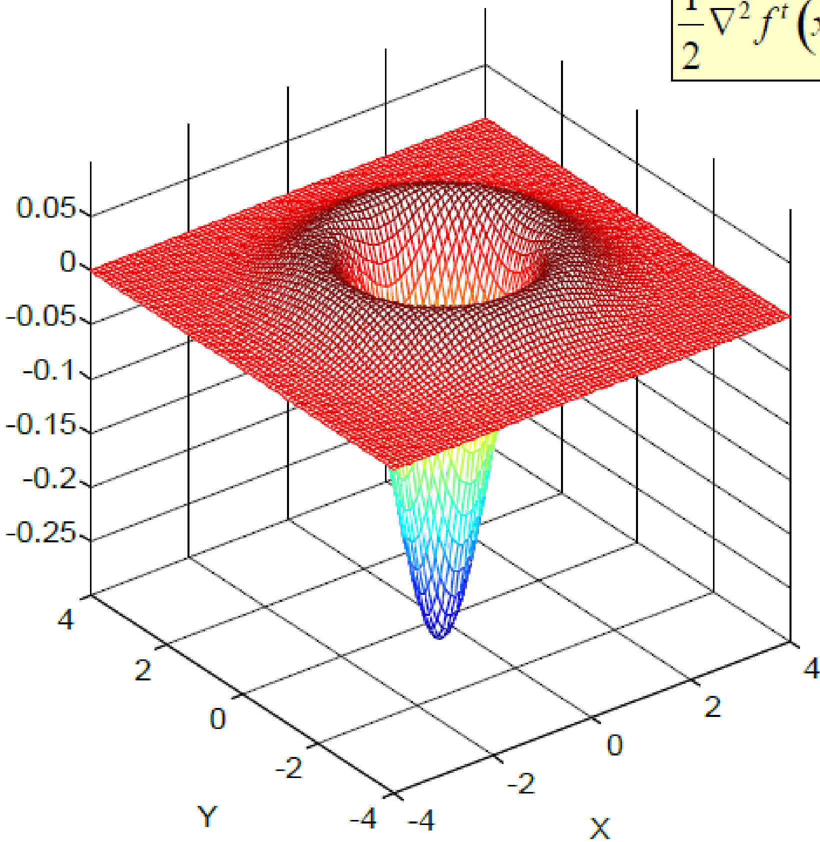
DoG and LoG

- Remember that:

$$\nabla^2_{norm} L = s \nabla^2 L \qquad\qquad (s = \sigma^2)$$

Laplacian of Gaussian

$t = \sigma^2 = 1$

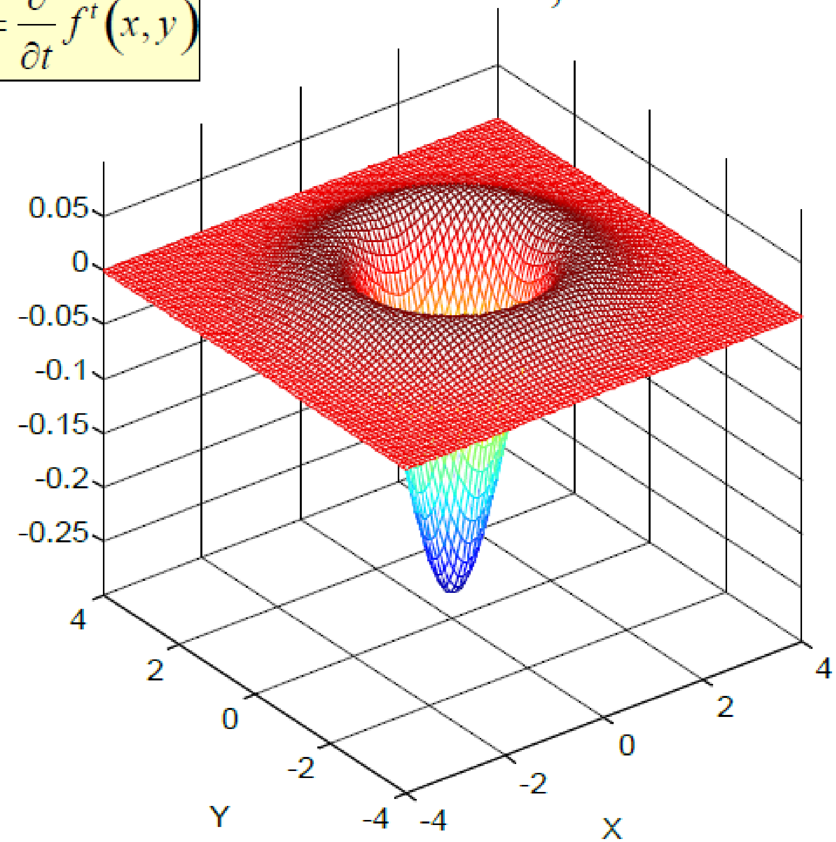


$$\frac{1}{2}\nabla^2 f^t(x,y) = \frac{\partial}{\partial t} f^t(x,y)$$

$$LoG(x,y) = -\frac{1}{\pi t^2}\left(1 - \frac{x^2+y^2}{2t}\right)e^{-\frac{x^2+y^2}{2t}}$$

Difference of Gaussians

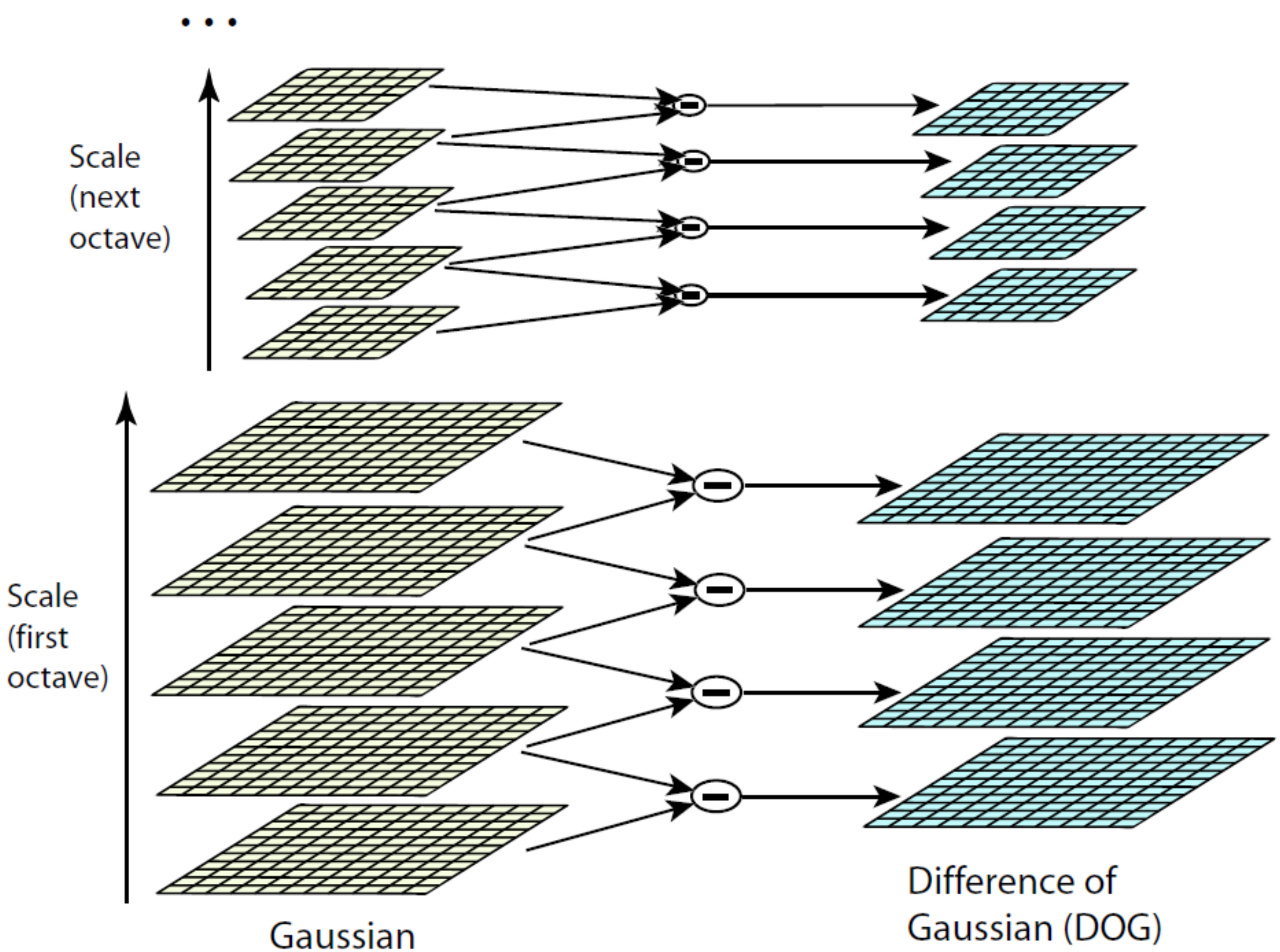
$t = \sigma^2 = 1,\ k = 1.1$



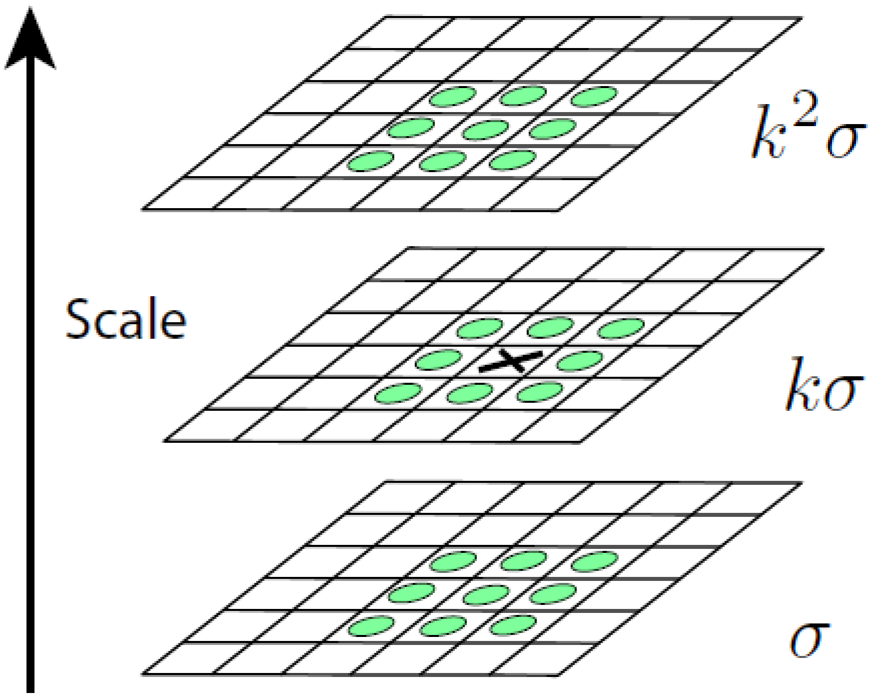
$$DoG(x,y) = \frac{1}{(k-1)t}\left(g^{k^2 t}(x,y) - g^t(x,y)\right)$$

From Digital Image Processing course by Bernd Girod, Stanford University. 2013.

Scale Space Extrema Detection



Scale Space Extrema Detection



The minimum/maximum is selected considering a 3 x 3 x 3 neighborhood.

Accurate Keypoints Detection

- The detection accuracy can be improved (sub-pixel, sub-space accuracy) by finding the extrema of a local 3D quadratic function which fit locally the scale-space representation (Brown and Lowe 2002):

From Taylor expansion:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}} \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

Extrema at :

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$

Keypoints Filtering

- Keypoints are filtered in two ways:
  - Low contrast (unstable extrema)
  - Keypoints that belong to an edge
- Low-contrast check:
  - a keypoint is removed if $D(x^*) < 0.03$

(image values are normalized in [0,1])

Keypoints Filtering

- A keypoint on an edge “can move” along the edge → not stable.
- Local Hessian matrix (***H***) provides information about curvature of the image.
- In analogy with the Harris Corner Detector, it is possible to use ***Det(H)*** and ***Tr(H)*** to write a condition of rejection without computing explicitly the eigenvalues of ***H***

Keypoints Filtering

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

Edge Check:

$$\boxed{\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}}$$

r **is the ratio between the directions of principal curvature**

832 keypoints

729 keypoints

536 keypoints

SIFT – Orientation Assignment

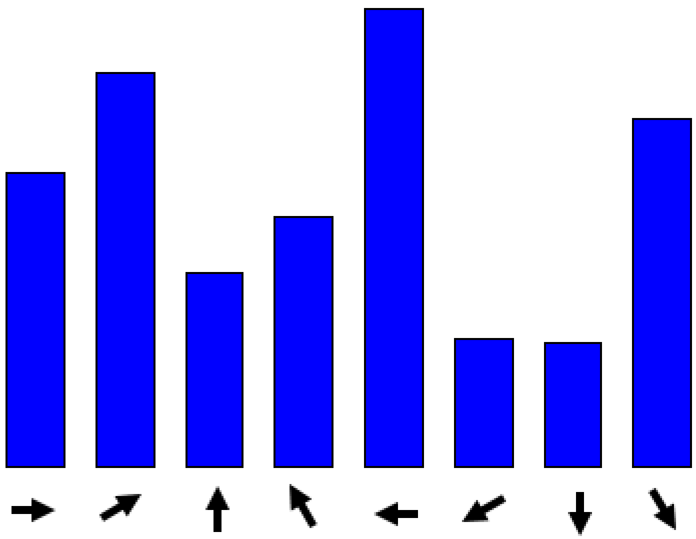
- To get orientation invariance a reference orientation is assigned to each keypoint.
- First, compute orientation at the selected scale:

$$
\begin{aligned}
m(x,y) &= \sqrt{L_x^2 + L_y^2} \\
\theta(x,y) &= \tan^{-1}(L_y / L_x)
\end{aligned}
$$

(note the use of ***L(.)*** and not ***D(.)***)

SIFT – Orientation Assignment

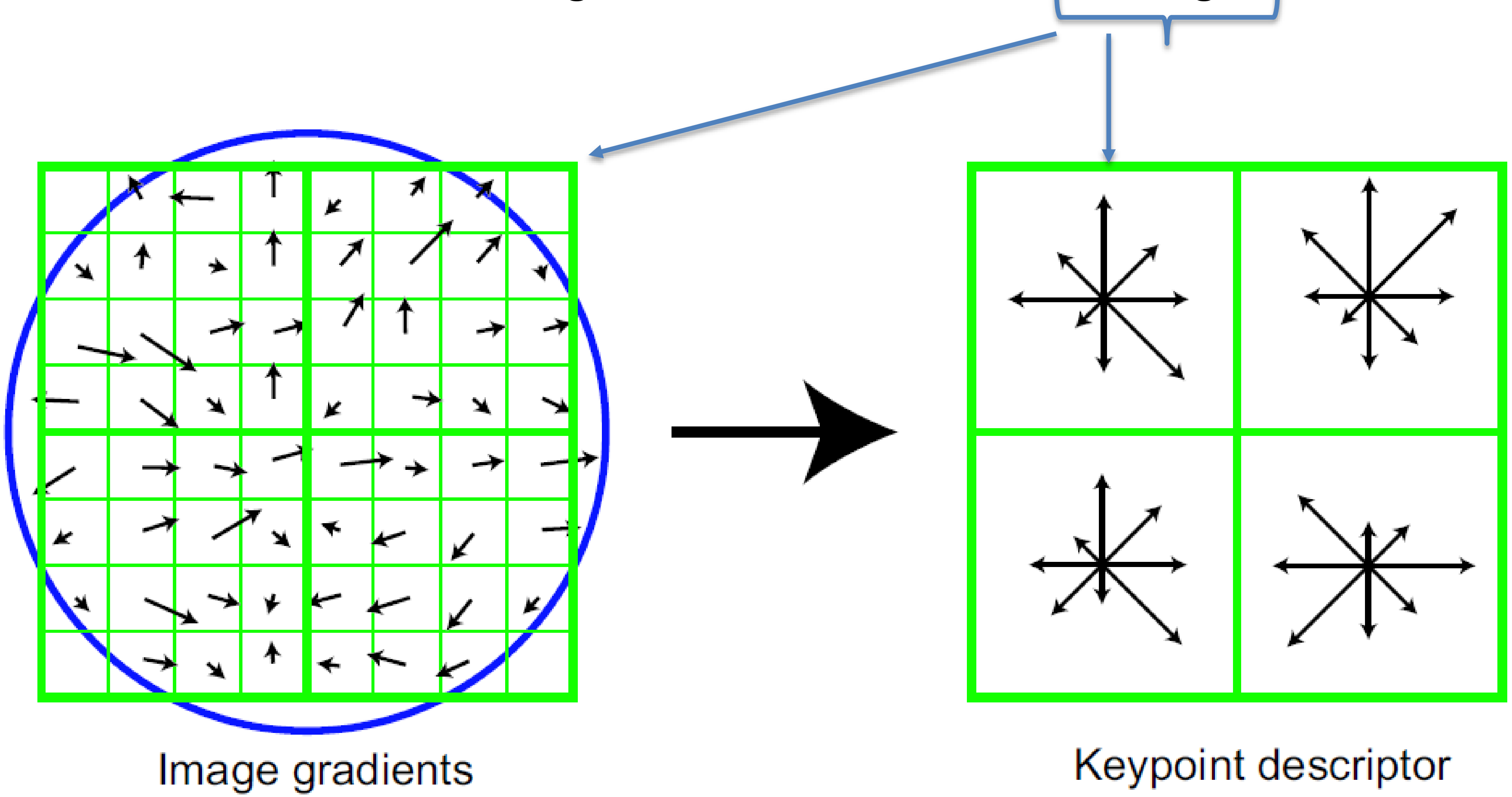
- An histogram of orientations is accumulated considering the surrounding pixels (*σ=1.5*).
- The orientation is given by the highest peak.



- A new keypoint (with orientation) is assigned to any peak within the 80% of the highest peak.

SIFT Descriptor

Orientation histogram accumulated over a 4 x 4 region



2x2 histograms from an 8x8 region

SIFT Descriptor

- SIFT descriptor is composed by 4 x 4 histograms (with 8 bins) accumulated over 4 x 4 regions of 4 x 4 size.
- SIFT has 128 components (4x4x8=128)

SIFT Performance

- Robust to lighting changes (gradient-based)
- Robust to image noise
- Robust to viewpoint changes
  - about 80% repeatability at 35 degree viewpoint change (rotation in depth)
- Very effective in object recognition

SURF

- First presented at ECCV'06 (Bay et al. 2006*)
- The aim is to build a *robust* (like SIFT) features detector and descriptor but *fast* to compute.
- Exploit integral image and box filters approximation of Gaussian derivatives.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *ECCV 2006.

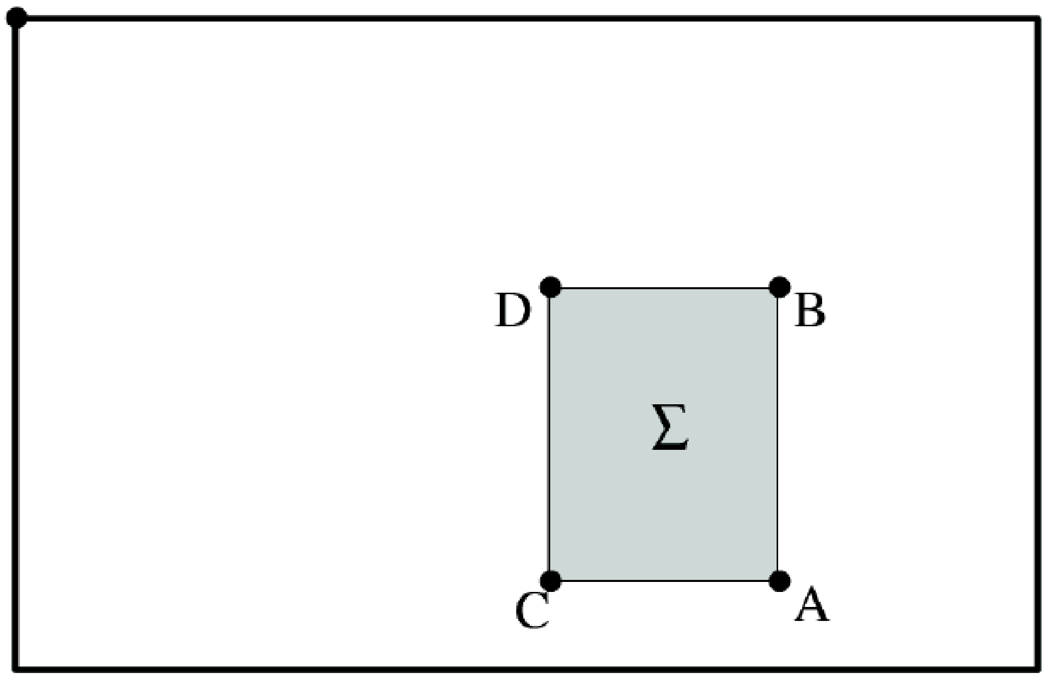
SURF – Integral Image

- The *integral image* I_{Σ} is an image such that the pixel *(x,y)* contains the sum of all the pixels within the rectangular region defined by the origin *(0,0)* of the image and the point *(x,y)*:

$$I_{\Sigma}(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x,y)$$

SURF – Integral Image

- It is easy to use I_{Σ} to calculate the sum of the pixels in any region of the image:

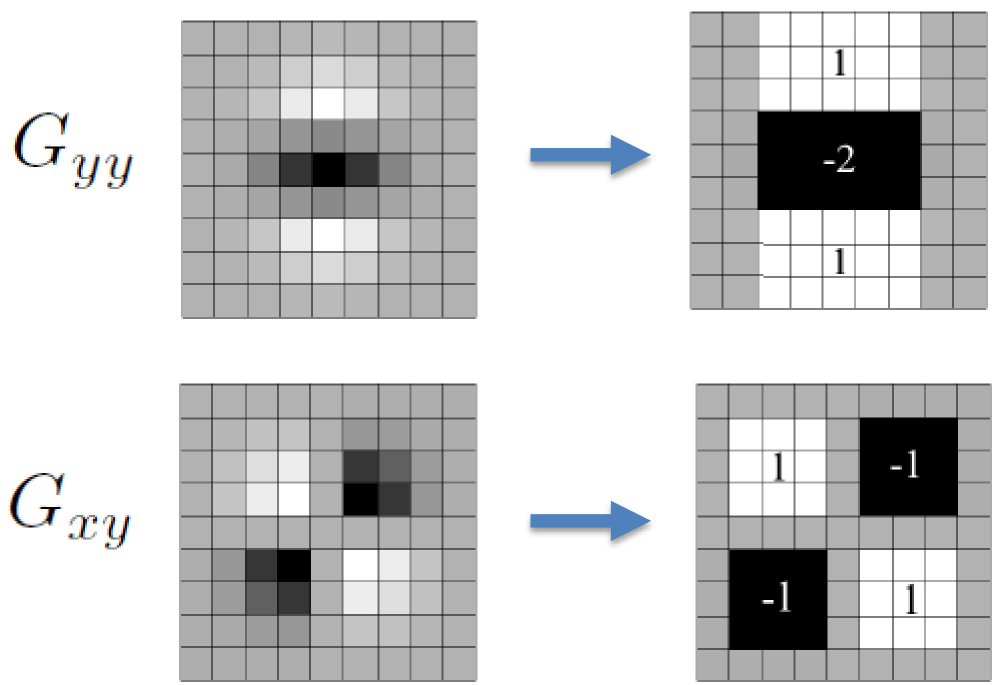


$$\boxed{\Sigma = A - B - C + D}$$

*Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359, 2008.

Gaussian Derivatives Approximation

- *Box filters* can be used to approximate the computation of Gaussian derivatives:



Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359, 2008*.

Keypoints Detection

- Interest point localization is based on the Hessian matrix:

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$$

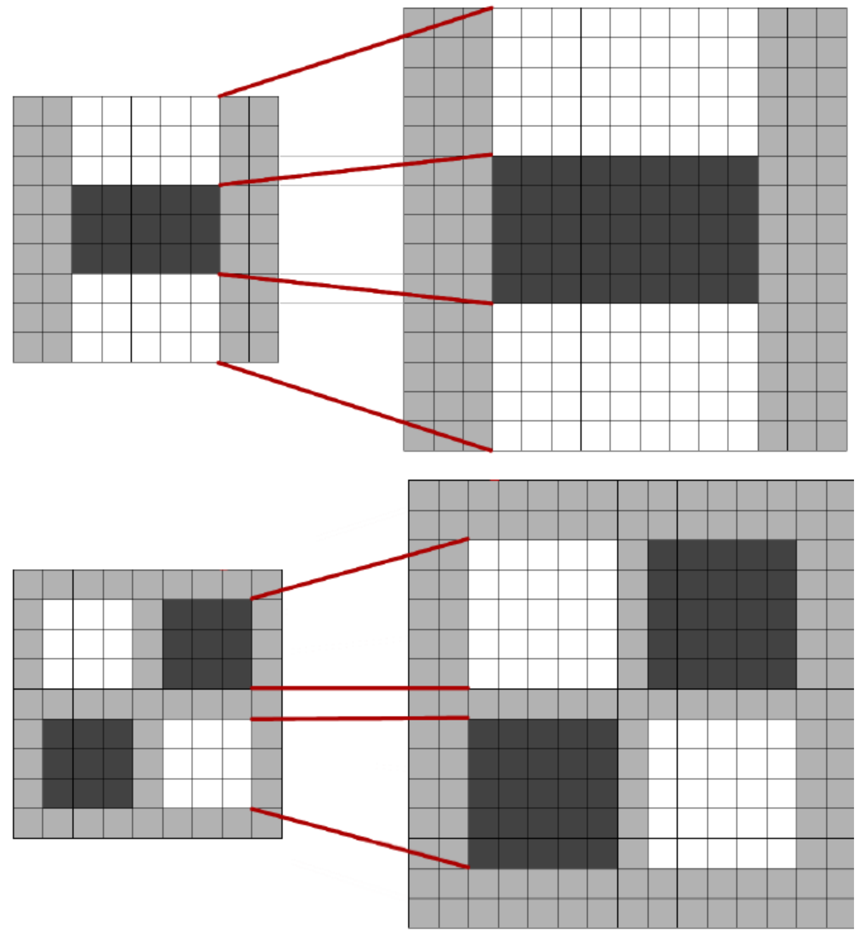
- The maximum response of the determinant of the Hessian matrix is searched:

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2$$

→ ***D***$_{xx}$ is the approximation of ***L***$_{xx}$ computed using the box filter, ***D***$_{yy}$ is the approximation of ***L***$_{yy}$, and so on..

Scale Space Detection

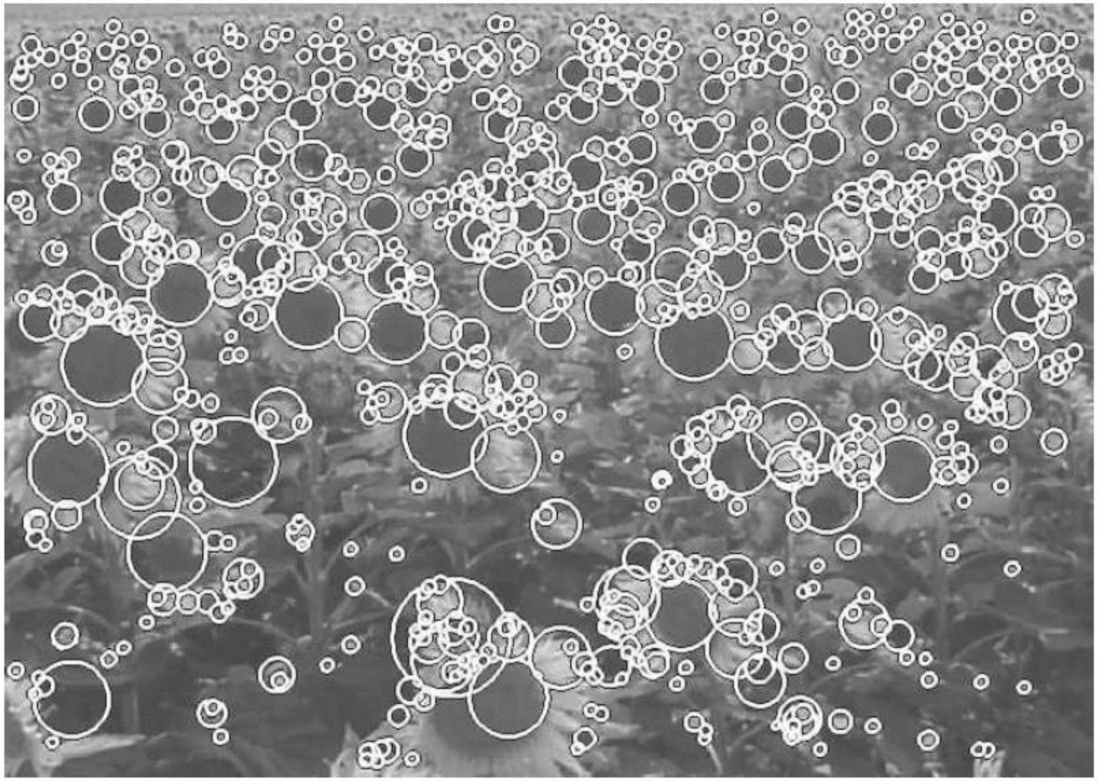
- The scale space representation is build by increase the filter size instead of reducing the image size.



*Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359, 2008.

Scale Space Detection

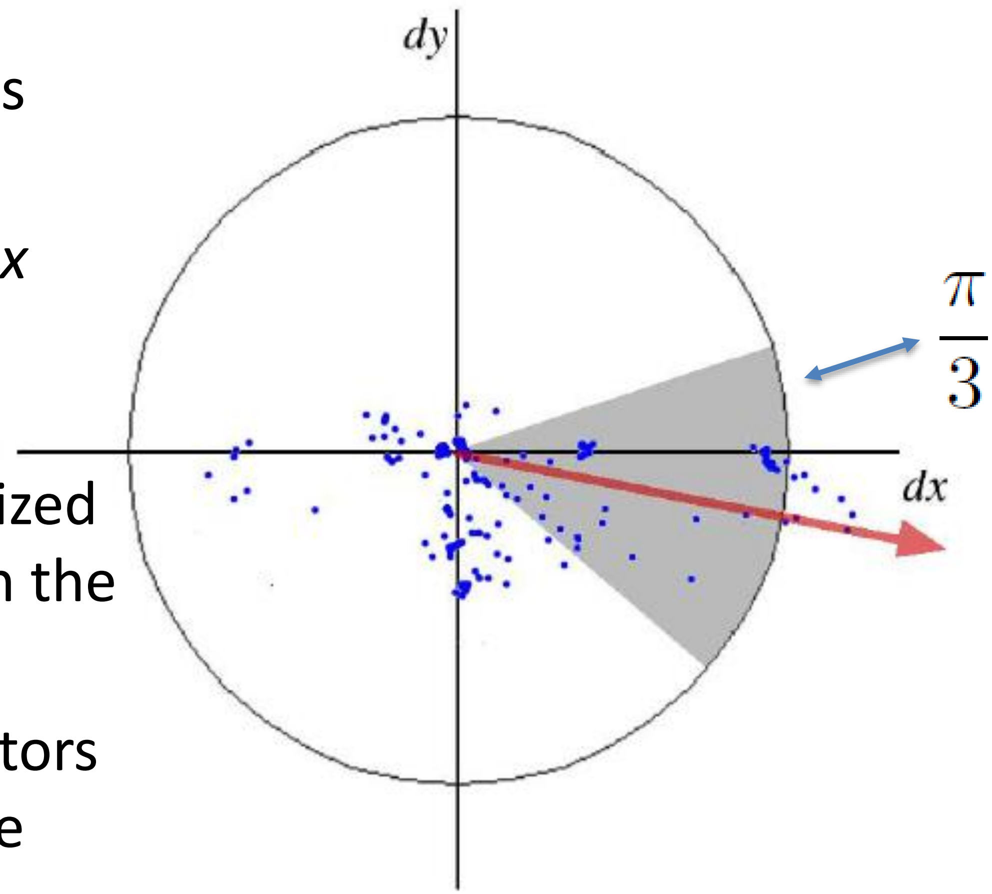
- Maxima in a 3 x 3 x 3 neighborhood.
- Localization improvement by the same approach in SIFT (Brown and Lowe 2002).



Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359, *2008.

SURF – Orientation Assignment

- A neighborhood of size *6s* is considered.
- At a sampling step of *s* the *x* and *y* wavelet response is calculated.
- These responses are organized as points as in the figure on the right (*dx/dy space*).
- The longest sum of the vectors in a windows is taken as the *dominant orientation*.



***Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359, 2008.**

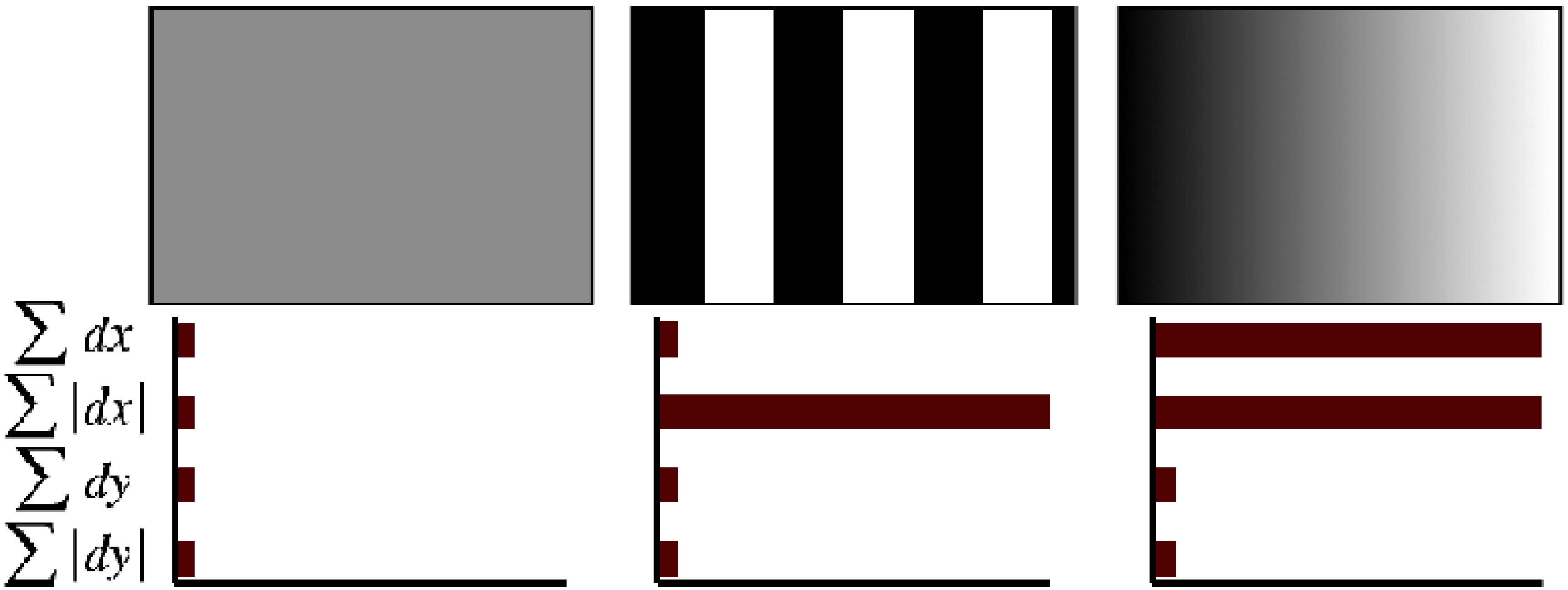
SURF Descriptor

- An oriented window *W* of size *20s* is constructed.
- *W* is subdivided in 4 x 4 sub-regions.
- For each sub-region, the Haar wavelet responses at 5 x 5 regularly spaced sample points are computed.
- Each sub-region is associated to the following descriptor vector:

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right)$$

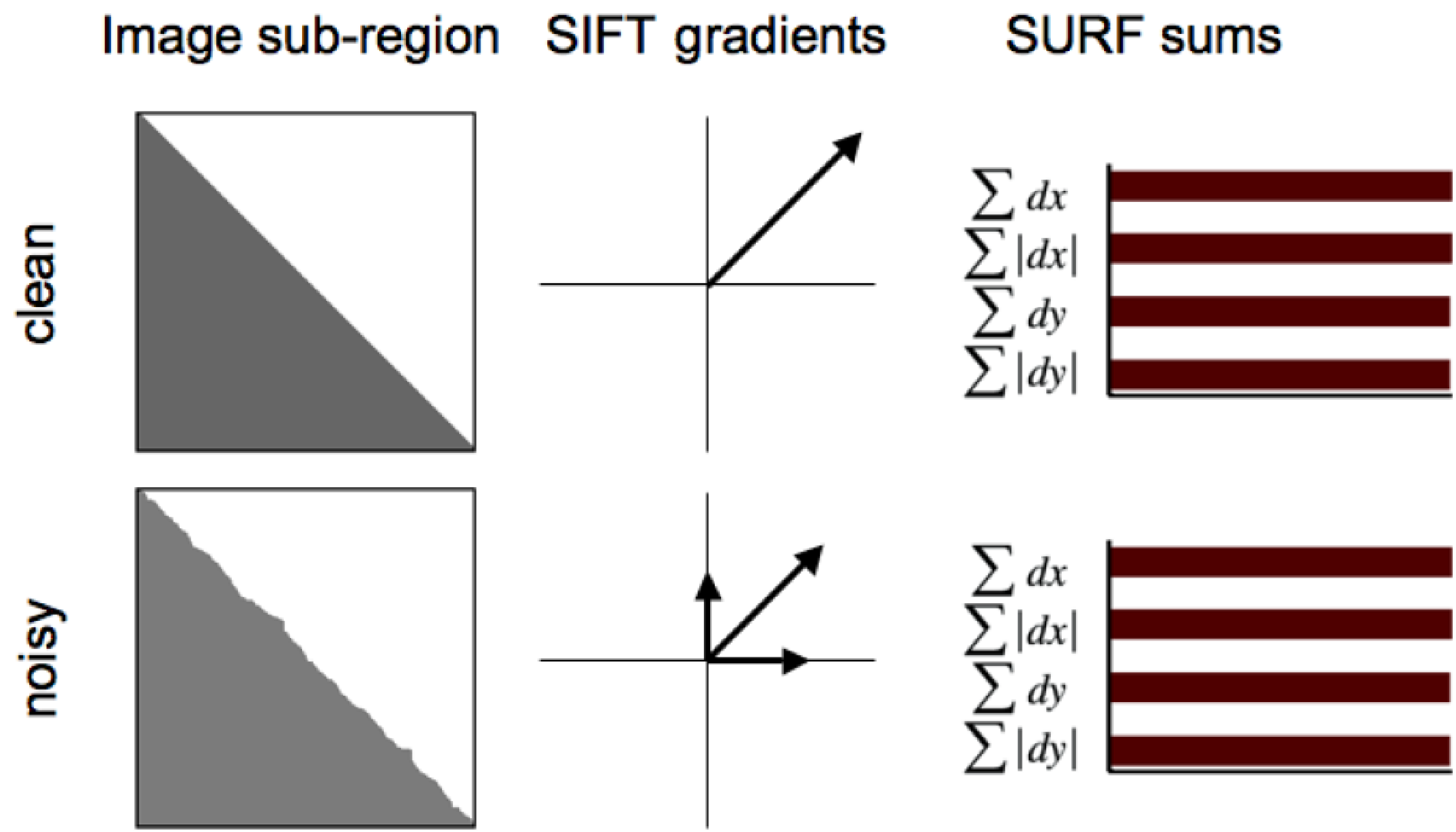
- ***SURF-64***: 4 x 4 x 4 = 64 components.
- SURF-128: the vector v is splitted in positive and negative d_x and d_y.

SURF Descriptor



*Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359, 2008.

SURF Descriptor vs SIFT Descriptor



Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speed Up Robust Features", *CVIU,* Vol. 110, No. 3, pp. 346–359*, 2008.

SURF vs SIFT

- SURF is faster than SIFT (about *3x* w.r.t the original implementation)
- Performance are similar, in general
  - SURF is more precise
  - SURF is less robust in viewpoint change and illumination change

Affine Invariant Regions

- The idea is to find image regions that have properties such that they are robust against affine transformations.
- We take a look at:
  - IBR (Intensity-Based Regions) by Tuytelaars et al. (presented at BMVC’00)
  - MSER by Matas et al. (presented at BMVC’02)

Intensity-Based Regions (IBR)

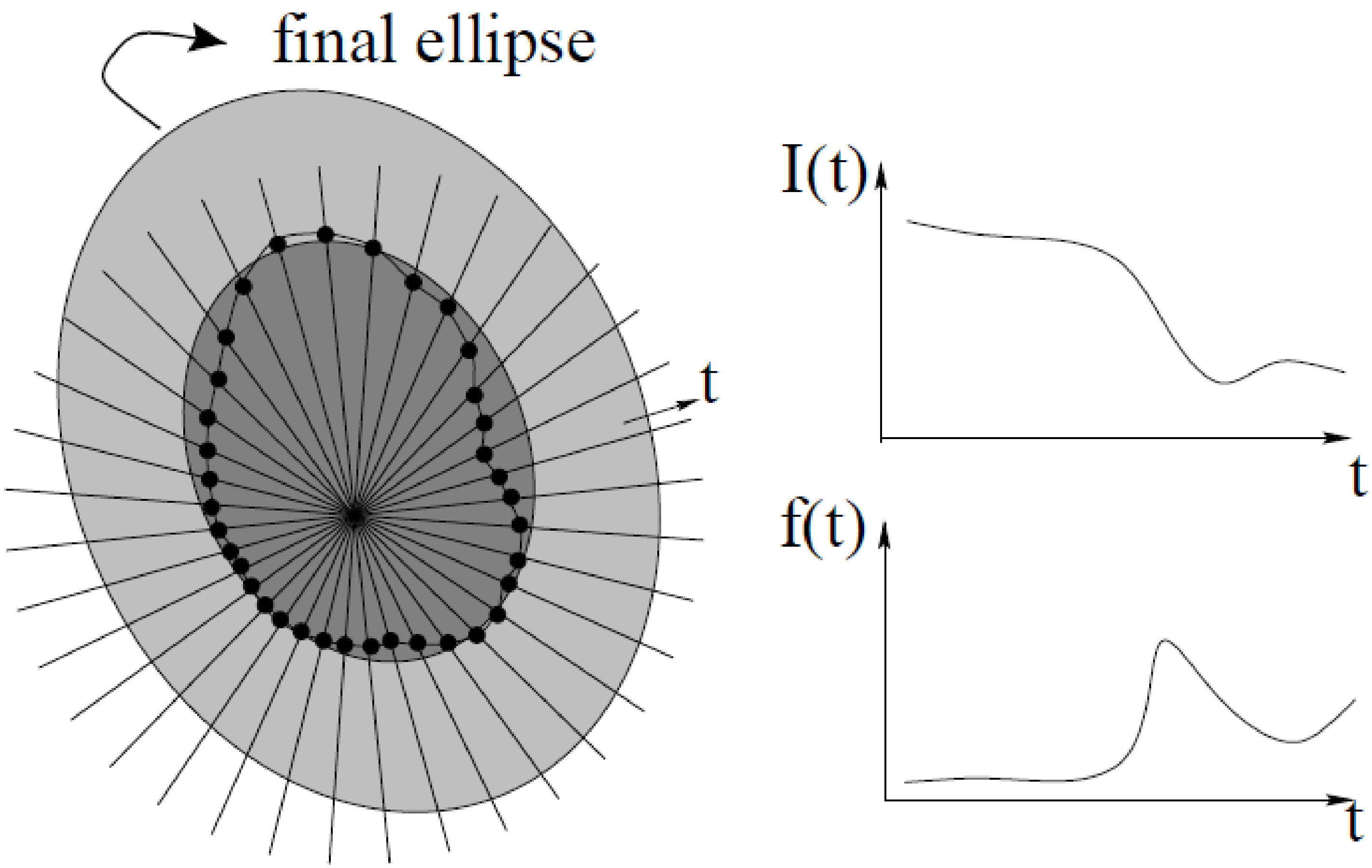
- A set of local maxima is found (non-maxima suppression is used).
- For each local maxima a set of rays is shot.
- Along each ray the extremum of:

$$f(t) = \frac{|I(t) - I_0|}{\max\left(\frac{\int_0^t |I(t)-I_0|dt}{t}, d\right)}$$

is found.

- These extremum points are connected to form an affine invariant region.

Intensity-Based Regions (IBR)



Tinne Tuytelaars and Luc Van Gool, "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions", *BMVC 2000*.

Intensity-Based Regions (IBR)

- Each region is described using the "Generalized Color Moments":

$$M_{pq}^{abc} = \iint_{\Omega} x^p y^q [R(x,y)]^a [G(x,y)]^b [B(x,y)]^c \, dx dy$$

- These moments characterize the shape, the intensity and the color distribution of the region in a robust and uniform way.

MSER - definitions

- **Region** Q is a contiguous subset of D, i.e. for each $p, q \in Q$ there is a sequence $p, a_1, a_2, \dots, a_n q$ and $pAa_1, a_i A a_{i+1}, a_n A q$.
- **(Outer) Region Boundary** $\partial Q = \{q \in D \backslash Q : \exists p \in Q : qAp\}$, i.e. the boundary ∂Q of Q is the set of pixels being adjacent to at least one pixel of Q but not belonging to Q.

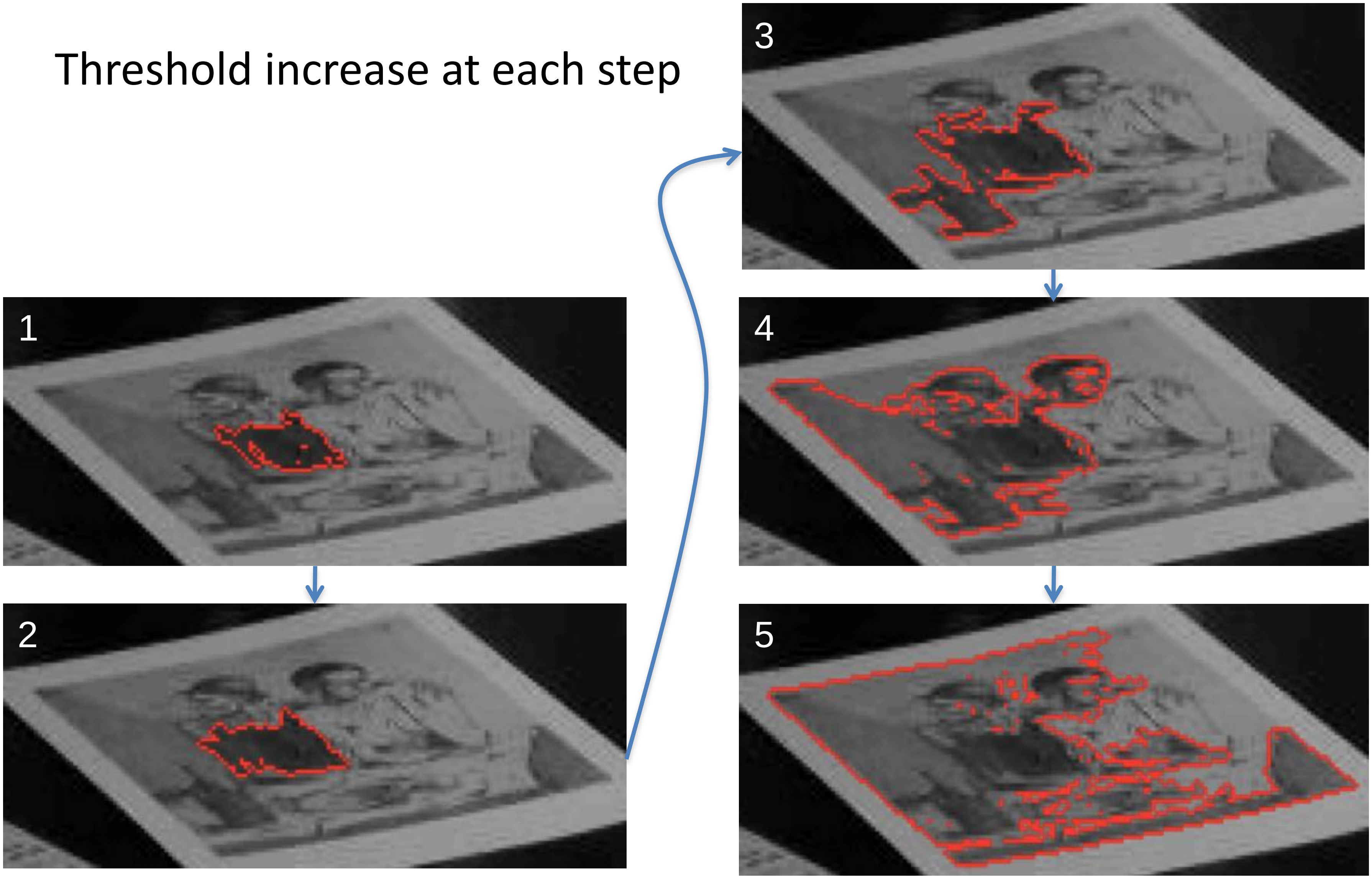
J. Matas, O. Chum, M.Urban, T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", ***BMVC 2002*****.**

MSER - definitions

- **Extremal Region** $Q \subset D$ is a region such that for all $p \in Q,\ q \in \partial Q$: $I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region).
- **Maximally Stable Extremal Region (MSER).** Let $Q_1, \ldots, Q_{i-1}, Q_i, \ldots$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$. Extremal region Qi^* is maximally stable iff $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}|/|Q_i|$ has a local minimum at i^* ($|.|$ denotes cardinality). $\Delta \in S$ is a parameter of the method.

MSER “visualized”

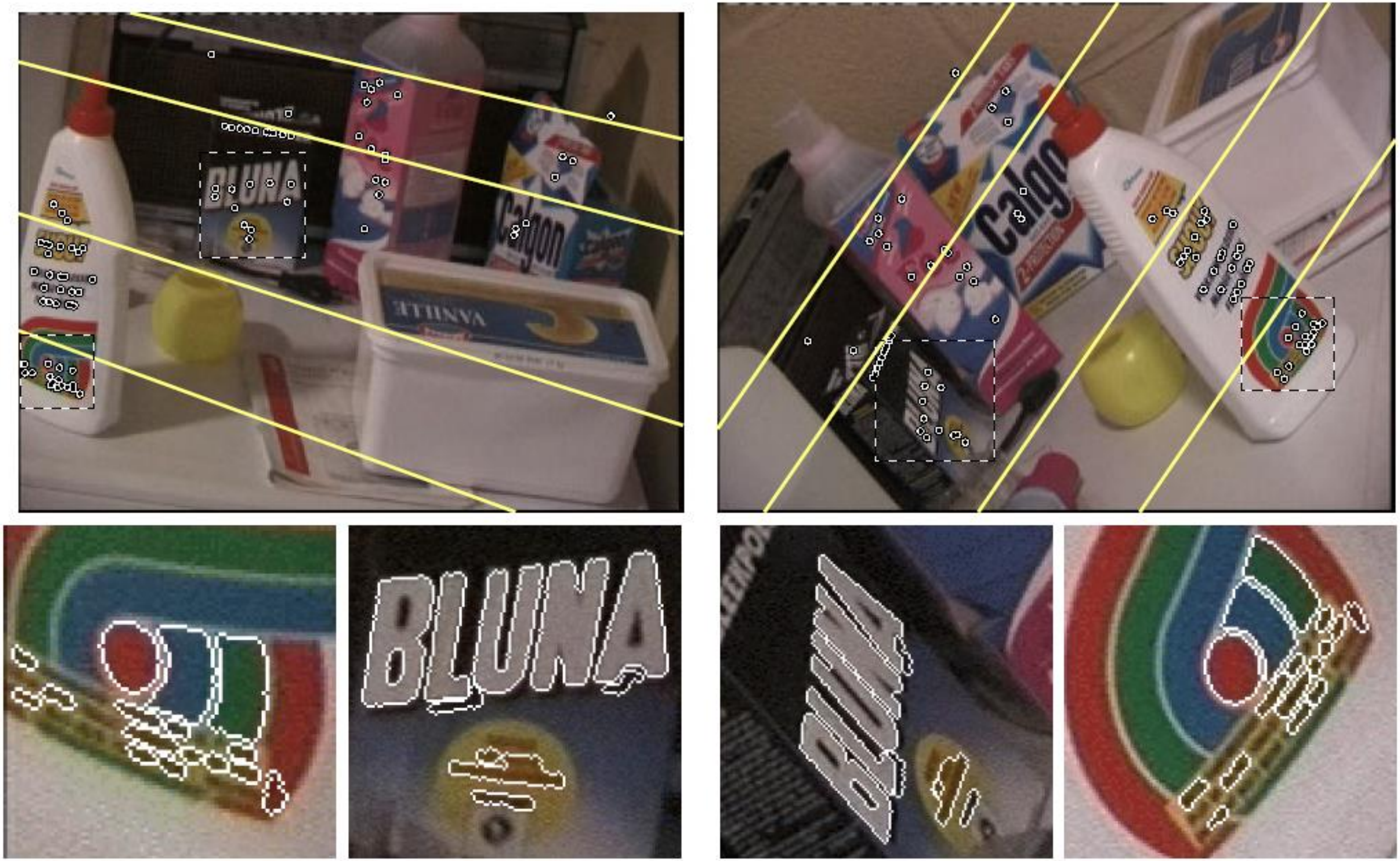
Threshold increase at each step



MSER

J. Matas, O. Chum, M.Urban, T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", ***BMVC 2002*****.**

MSER



J. Matas, O. Chum, M.Urban, T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", *BMVC 2002*.

Histogram of Oriented Gradients (HOG)

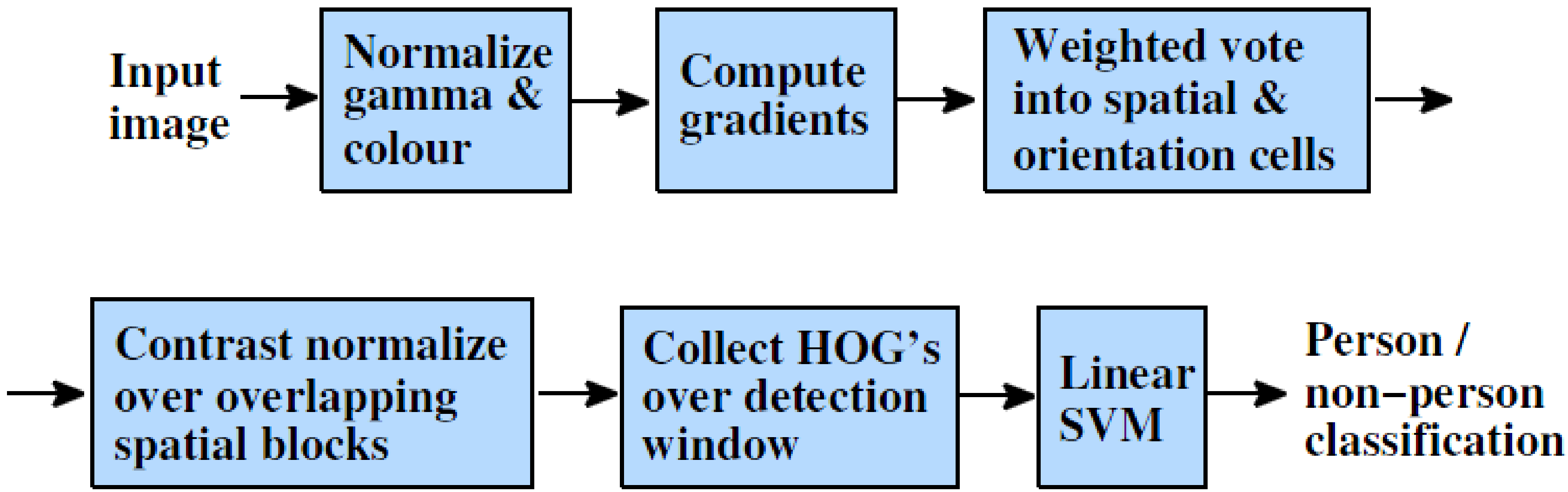
- First proposed by Dalal and Triggs* in 2005 for human detection.
- Now used in thousands of Computer Vision works.
- Based on the orientation of the image gradient → ***dense descriptor (!)***

N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *CVPR 2005*.

Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG)

HOG - Pipeline

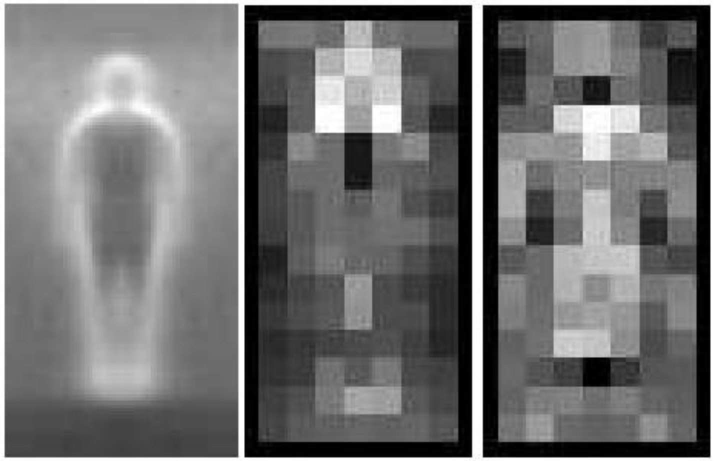


N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", ***CVPR 2005*****.**

HOG – Implementation Details

- Image gradient → simple 1-D mask works best!! (*[-1 0 1] , [1 0 -1]*^{T})
- Spatial binning → 16 x 16 pixel blocks of four 8 x 8 pixel cells.
- Orientation binning → 9 orientations (in the range 0-180).
- Block normalization → clipped L2-norm.
- The descriptor is all the components of the normalized cell responses for all the blocks in the detection window (blocks are overlapped).

HOG for Person Detection



Average gradient image

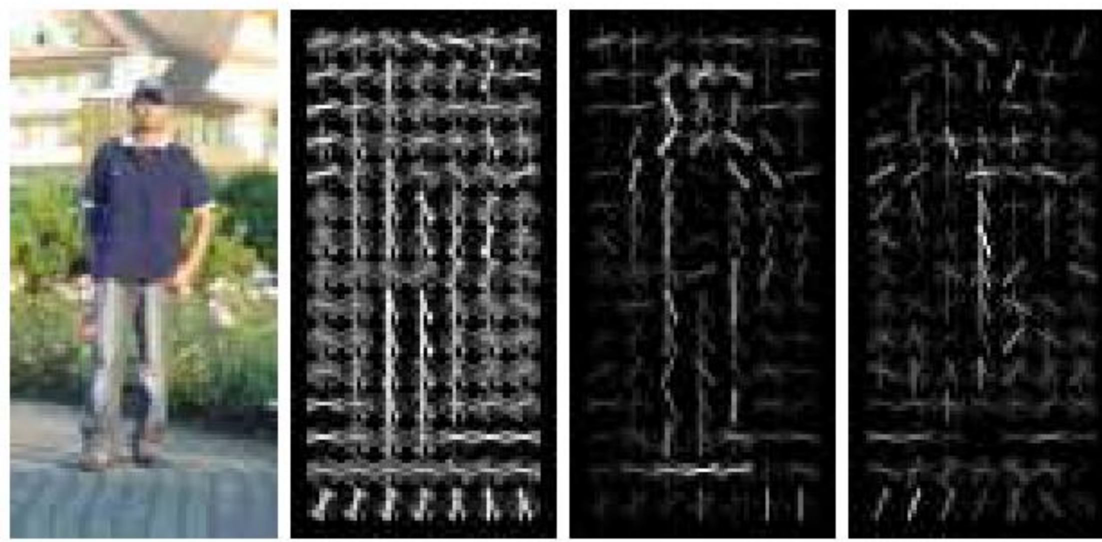
Max positive weights

Min positive weights

R-HOG weighted (negative weights)

R-HOG weighted (positive weights)

R-HOG



Person detection with HOG & linear SVM

N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", ***CVPR 2005*****.**

Recap

- Many types of feature detectors and descriptors have been developed during the last 10 years.
- Local features together with different descriptors are used in many applications.
- Scale invariance is very important (!)
- SIFT-related ideas and HOG are used in thousands of papers.

Questions ?