# Edge detection

Václav Hlaváč

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Center for Machine Perception
http://cmp.felk.cvut.cz/~hlavac, hlavac@fel.cvut.cz

*Thanks to T. Svoboda and T. Werner for several slides.*

# Edges - what for?



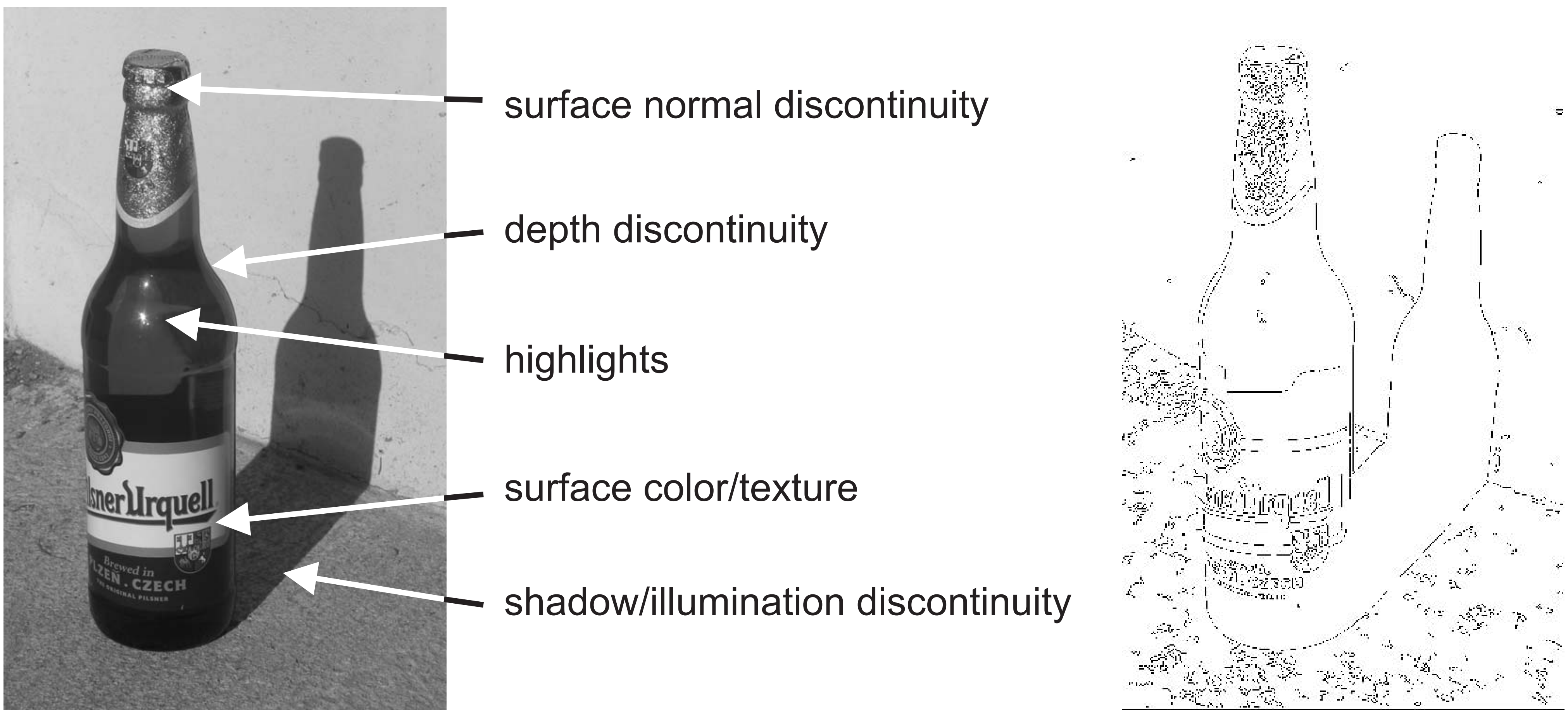
- It follows from neurophysiological and psychophysical studies that image points with high gradient attract attention.
- Such places carry more information.
- Edges often exhibit fair invariance to changes in illumination and/or viewpoint change.
- Edge detection is often a first step in computer vision algorithms: image recognition, 3D reconstruction, correspondence matching in stereo vision, tracking, etc.

# Example: Painting

Pablo Picasso, La Sieste 1919

# How do edges occur?

Edges are the result of discontinuities in the surface normal, in depth or in reflectance; they also appear due to highligts or irregularities in illumination (shadows).



# Edges and edgels



Edge

- is computed from gradient at a point. It shares its magnitude and its direction is 90 degrees away from it. It is a 2-dimensional vector.

Edgel

- is a pixel with an edge.

# Categories of Edge Detectors



Detectors based on:

1. finding maxima of first derivatives (Roberts, Prewitt, Sobel etc, Canny);
2. finding zero-crossing of second derivatives (Marr-Hildreth);
3. local approximation of image function by parametric model, e.g. a polynomial in $x, y$ (Haralick).

# Image Gradient

- Generally, the gradient of smooth function $f$ of $n$ variables is a vector of partial derivatives:

$$\nabla f(x_1, \ldots, x_n) = \left( \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

- For $n = 1$ (1D signal) it is equal to a (standard) derivative
- For $n = 2$ (2D signal) it's a 2D vector which can be described by polar coordinates (magnitude and angular direction $\psi$)

$$\|\nabla f(x, y)\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}, \quad \psi = \arctan \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right).$$

- Directional derivative of $f(x, y)$ in direction $(u, v)$ is $(u, v) \cdot \nabla f(x, y) = u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y}$.

# Discrete approximation of gradient

Can be done by either of these two ways (with similar results)

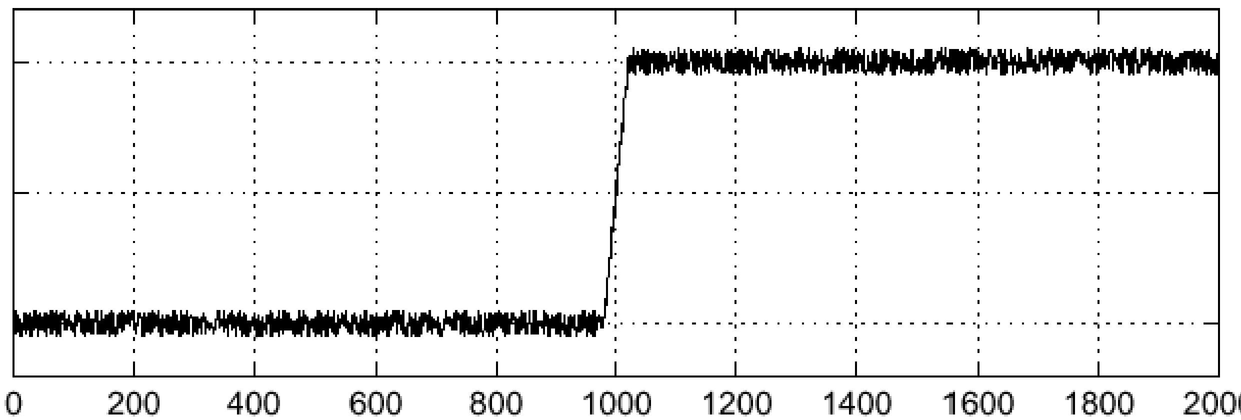
- reconstruct the continuous function from the discrete one and compute its derivative
- approximate the derivative by finite differences

Finite differences:

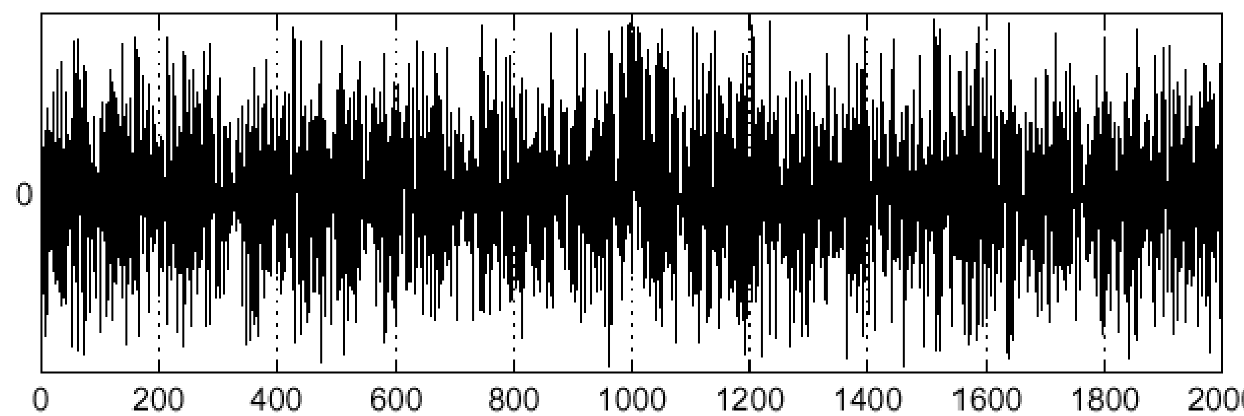
- non-symmetric: $f'(i) \approx f(i) - f(i-1)$. (left difference, does not use pixel value $f(i+1)$)
- symmetric: $f'(i) \approx \frac{1}{2}(f(i+1) - f(i-1))$ (O.K. but does not use $f(i)$)
- This can be done by convolution: $f' \approx [-1, +1] * f$,
  $f \approx [-0.5, 0, +0.5] * f$.

# Derivatives are susceptible to noise

image profile with noise

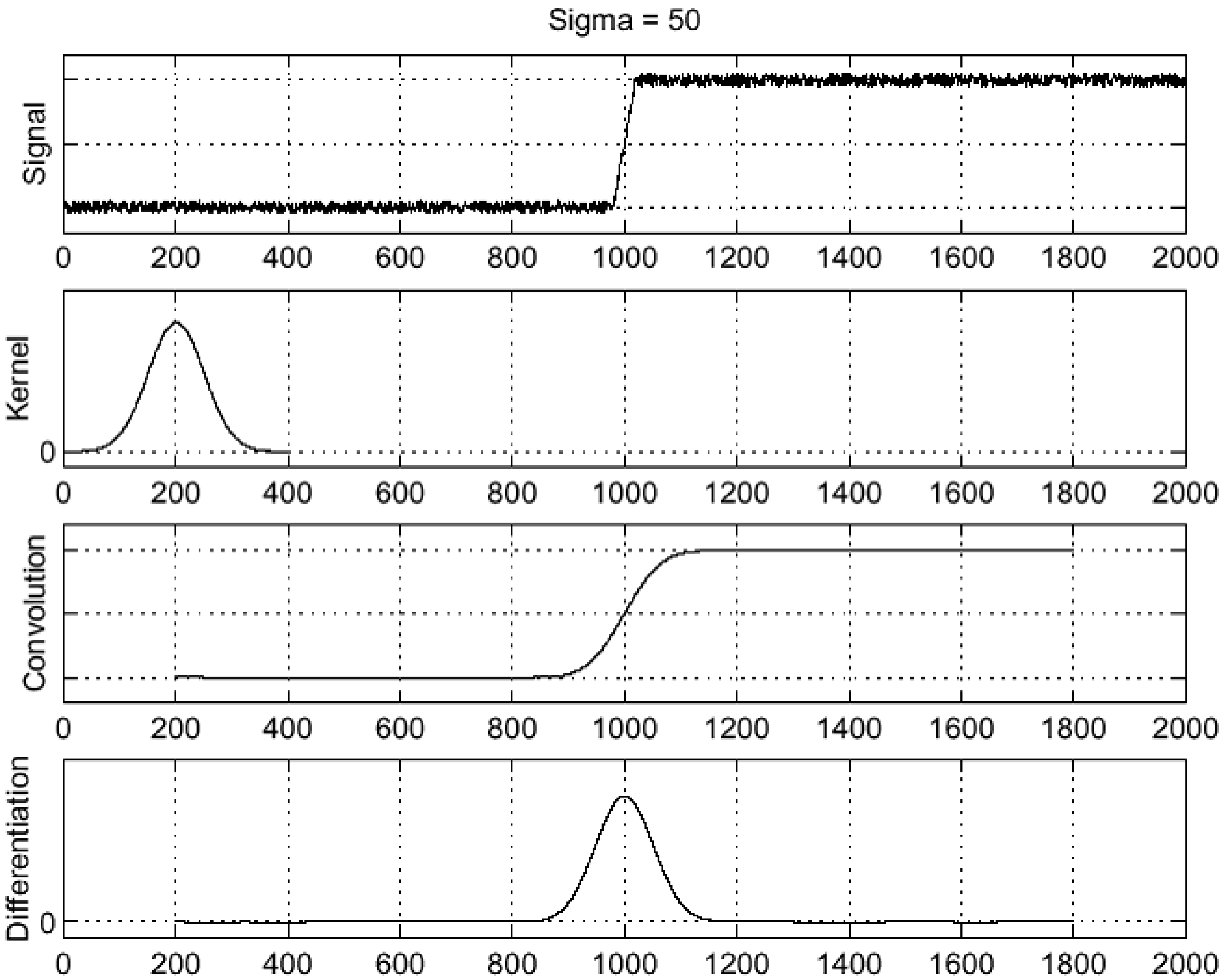



its derivative



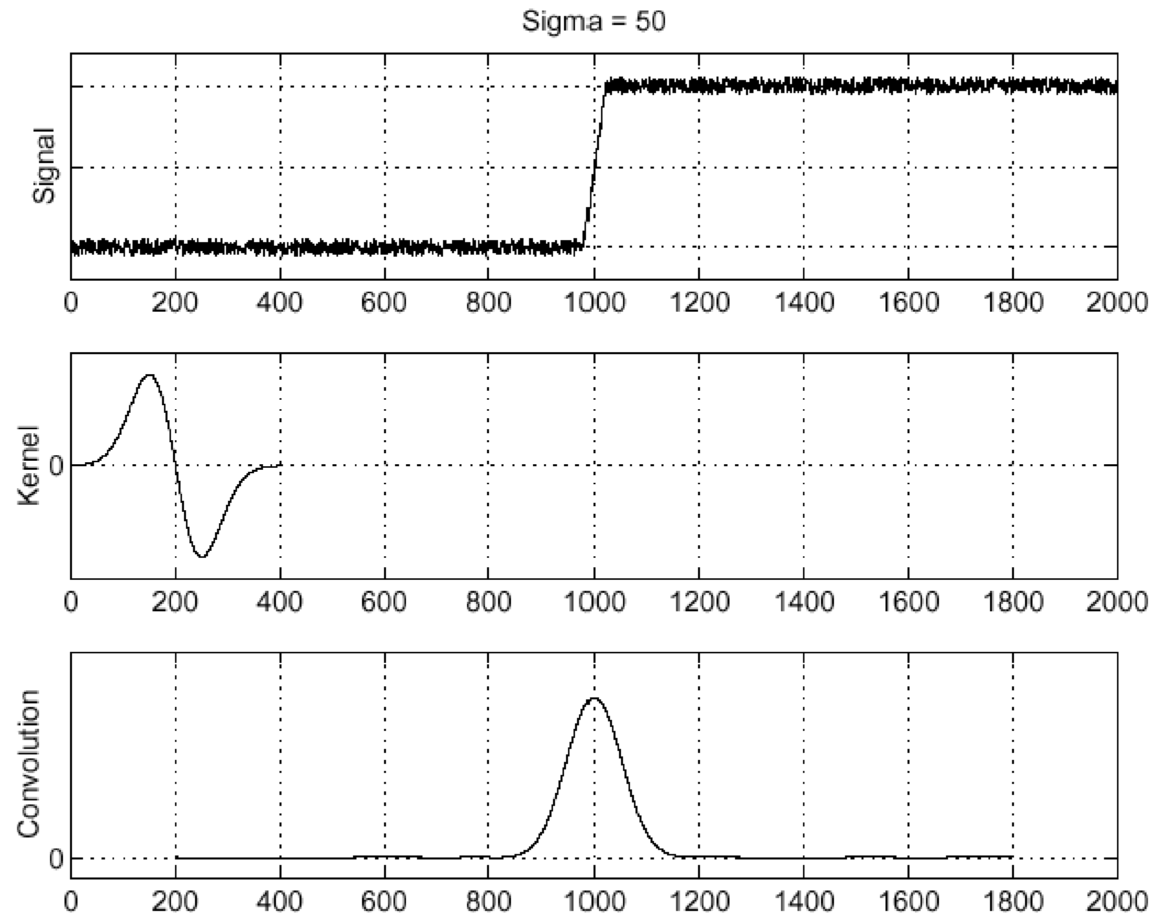

Where is the edge?

## → smooth it with a Gaussian



# Smoothing and derivative, combined



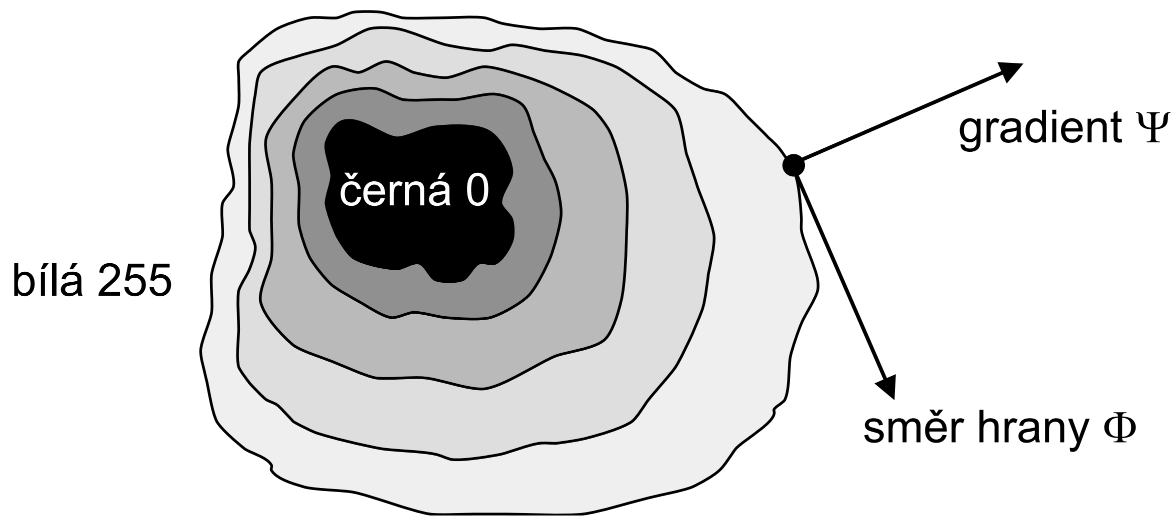



Due to commutativity of derivative and convolution, the two ops can be interchanged. Due to associativity, both operations can be combined to a single operator:

$$\frac{\mathrm{d}}{\mathrm{d}x}(h * f) = \frac{\mathrm{d}h}{\mathrm{d}x} * f \; .$$

# Edges and object boundaries

- Edges are sometimes chained in order to form object boundaries. This is why edge direction $\Phi$ is defined perpendicular to gradient direction $\Psi$.
- Provided that the object is separated from background by its image intensity, the boundaries are exactly the pixels with high magnitude of gradient.



# Convolution masks $3 \times 3$ for a derivative



- Roberts (only $2 \times 2$)
- Prewitt
- Sobel
- Robinson
- Kirsch
- Laplace (approximates trace of Hessian of image function)

## Roberts $(2 \times 2)$



Two masks:

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Magnitude of gradient computed as:

$$\| g(x,y) - g(x+1,y+1), g(x,y+1) - g(x+1,y) \| .$$

Sensitive to noise (small neighbourhood used only).

# Prewitt

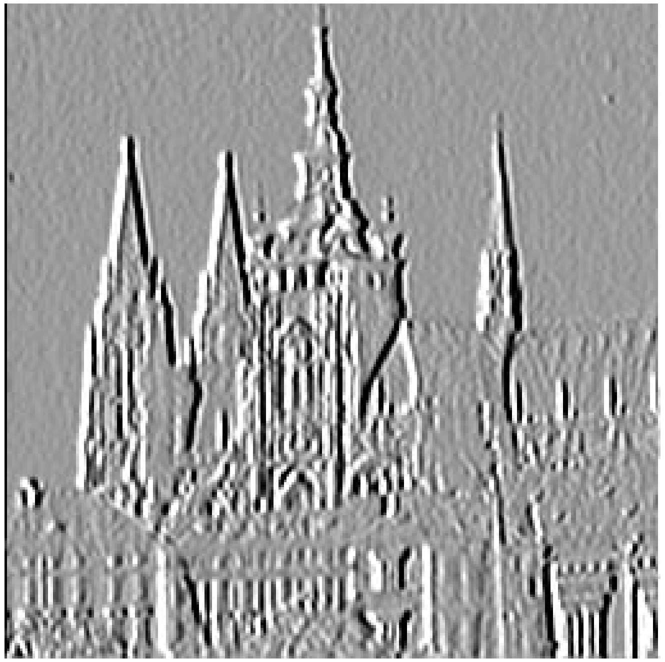
$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \qquad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix},$$

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \ldots$$

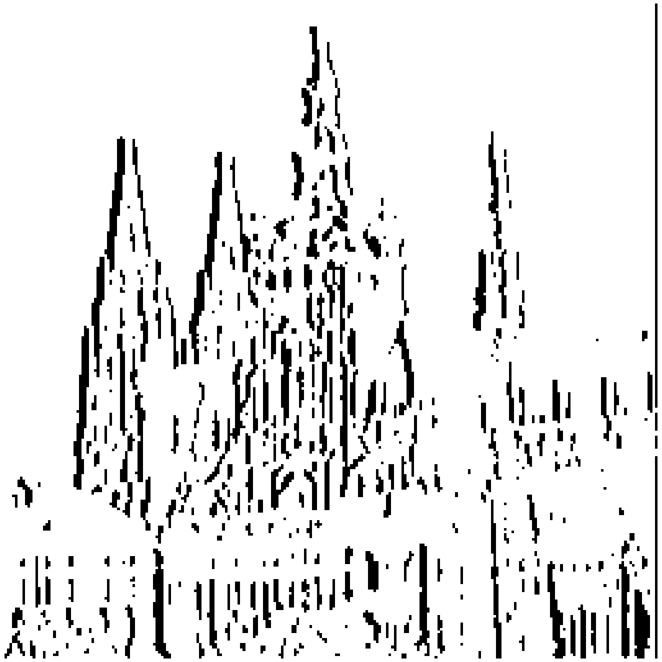
# Example, Prewitt derivative in horizontal direction



original $256 \times 256$



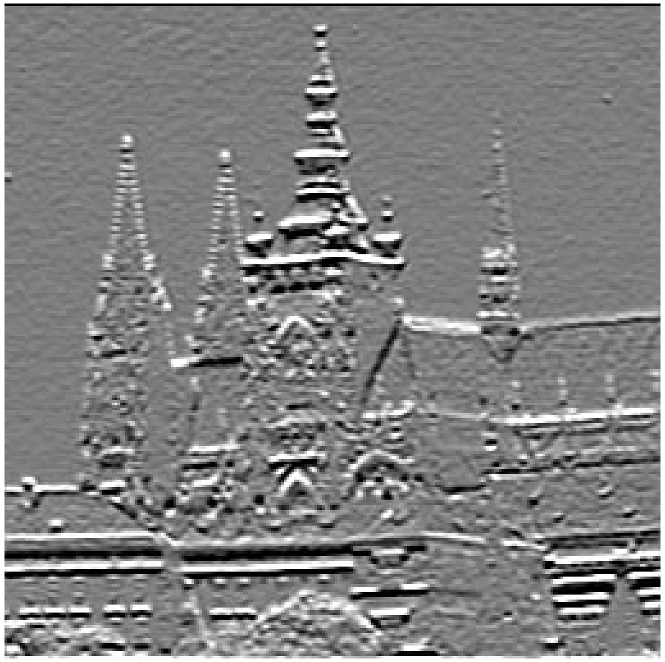
derivative



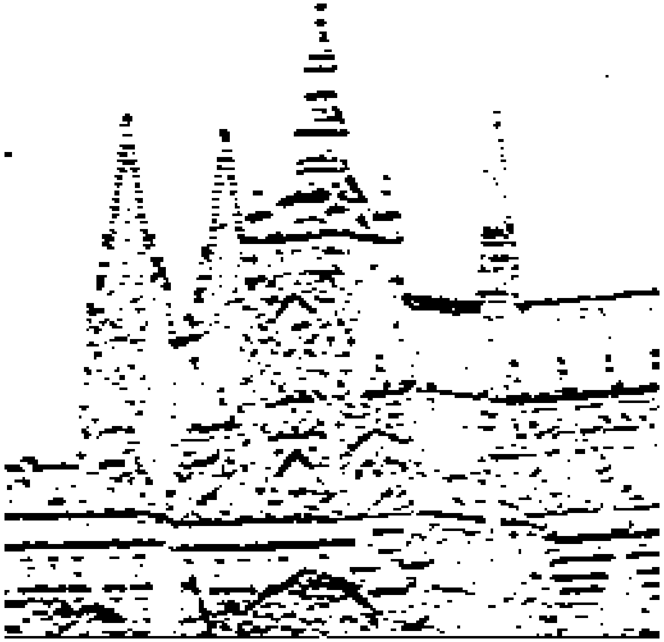
derivative, thresholded

# Example, Prewitt derivative in vertical direction

original $256 \times 256$



derivative



derivative, thresholded

# Sobel

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \qquad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix},$$

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \ldots$$

# Laplacian



Image function $f(x,y)$, its Laplacian is:

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

- $\nabla^2 f$ is a scalar, not a vector. There is no "direction" provided.
- The Laplace oprator is rotation-invariant.
- For a monotonically increasing image function $f(x,y)$, Laplacian crosses zero at a place where the gradient $\|\nabla f(x,y)\|$ is maximum.

# Discrete approximation of Laplacian

- Second finite difference is computed from first finite differences:

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2} \approx [-1, +1] * [-1, +1] = [+1, -2, +1]$$

- Laplacian is the sum of finite differences in horizontal and vertical directions:

$$\nabla^2 \approx \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Alternatives:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}, \quad \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

# Sharpening by a Laplacian



Laplacian is a high-pass filter.

original $256 \times 256$

Laplacian

Sharpening (- 0,4 $\cdot$)

# Laplacian of Gaussian (=Laplace of smoothed image)



Laplacian $\nabla^2$ is even more sensitive to noise than gradient. Thus, it is again combined with a Gaussian $G$. Again, the two operators can be combined to one $\longrightarrow$ LoG (Laplacian of Gaussian).

$$\nabla^2(G * f) = (\nabla^2 G) * f = \mathrm{LoG}(f)$$

For a given $\sigma$, (subst. $x^2 + y^2 = r^2$):

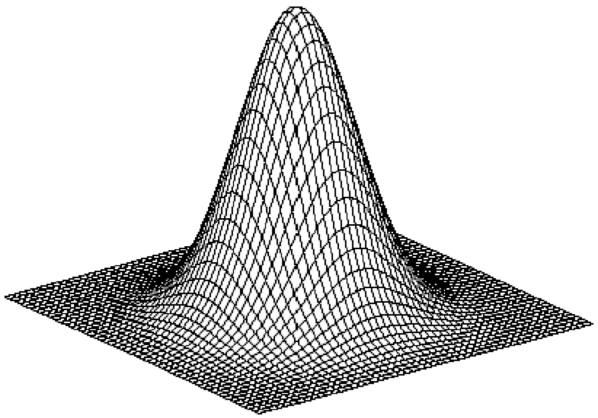
$$G(r) = e^{-\frac{r^2}{2\sigma^2}}, \quad G'(r) = -\frac{1}{\sigma^2}\, r\, e^{-\frac{r^2}{2\sigma^2}}, \quad G''(r) = \frac{1}{\sigma^2}\left(\frac{r^2}{\sigma^2} - 1\right) e^{-\frac{r^2}{2\sigma^2}}.$$

($c$ is normalization constant)

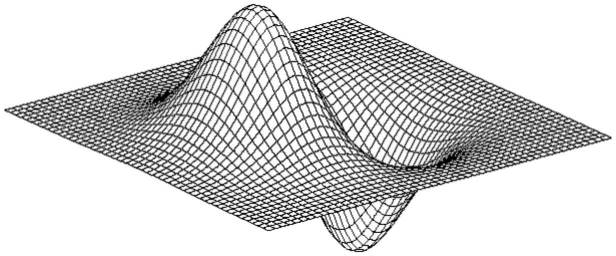
$$\nabla^2 G(x, y) = c \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

# 2D operators

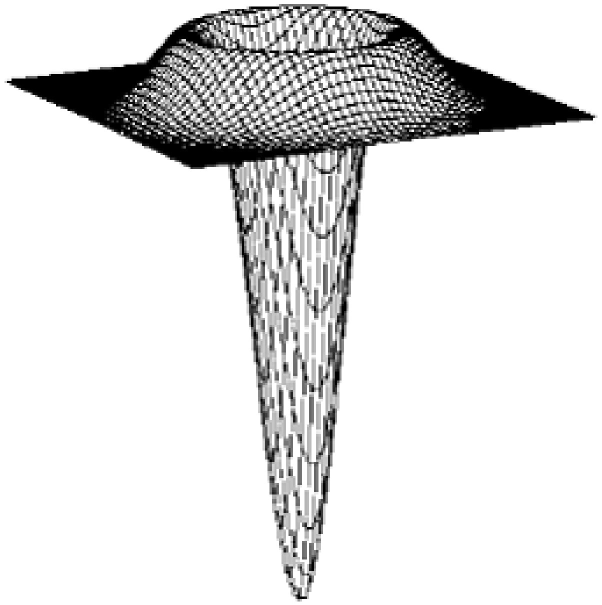




Gaussian



Derivative of Gaussian



Laplace of Gaussian (LoG)

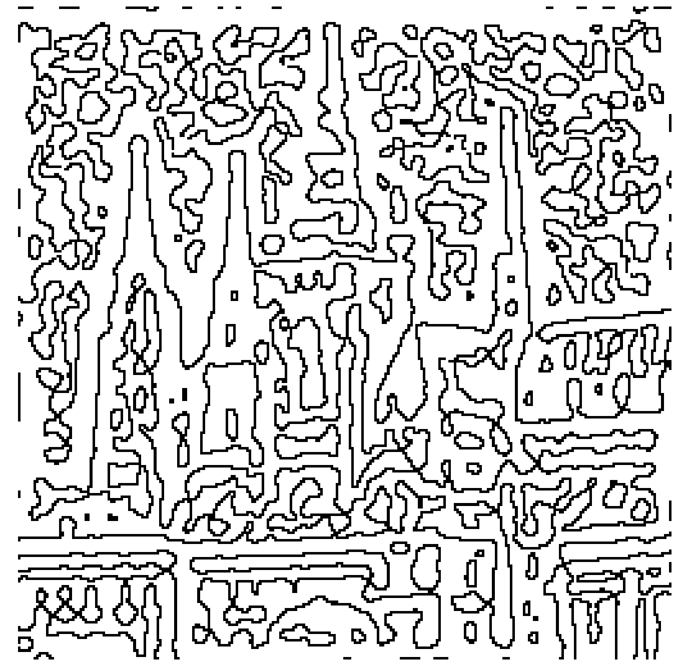
# How to evaluate zero crossings?

- E.g. in $2 \times 2$ neighbourhood; one pixel is a reference one and zero crossing is detected if sign is changed in the window.

# Zero crossings: example

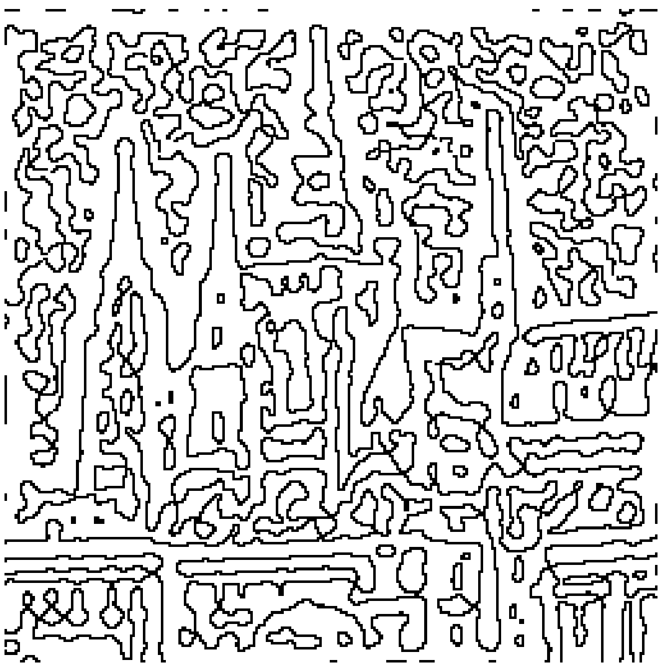
Original

DoG $\sigma_1 = 0,1$ $\sigma_2 = 0,09$

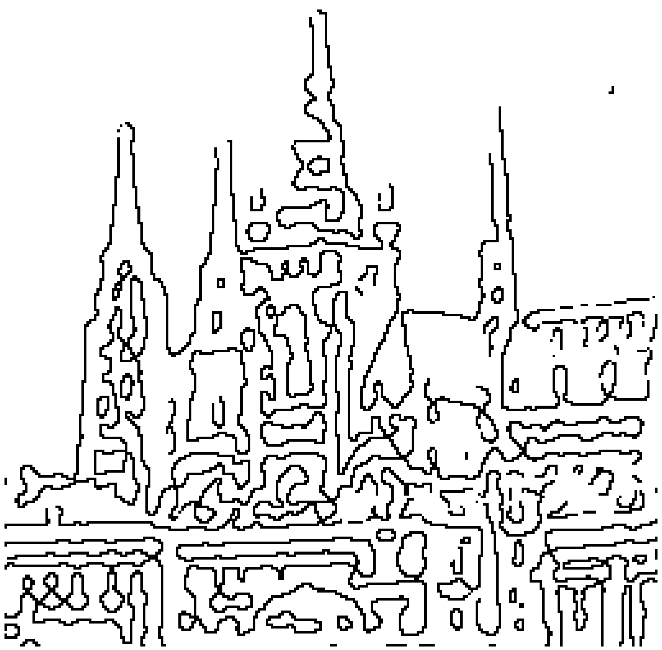


Zero crossings

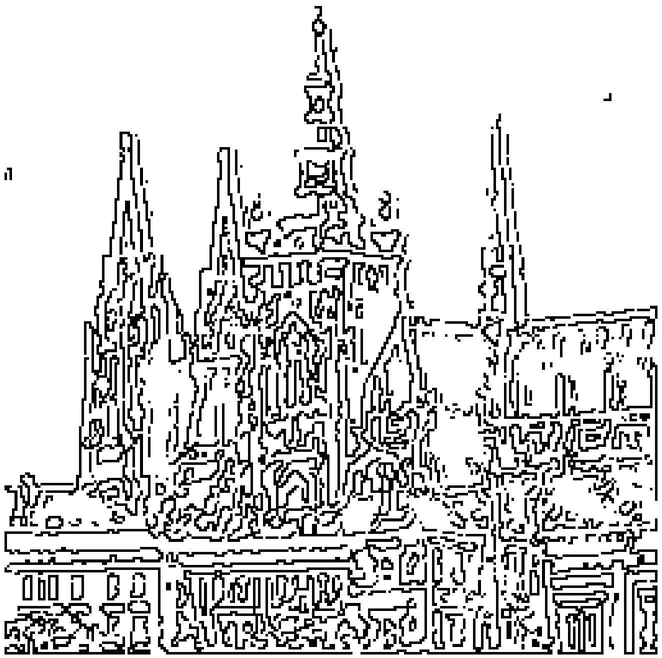
# additional edge strength thresholding



zero crossings



after weak edge removal,



LoG, $\sigma = 0,2$

# LoG & physiology

- Circular receptive fields (center-surround)

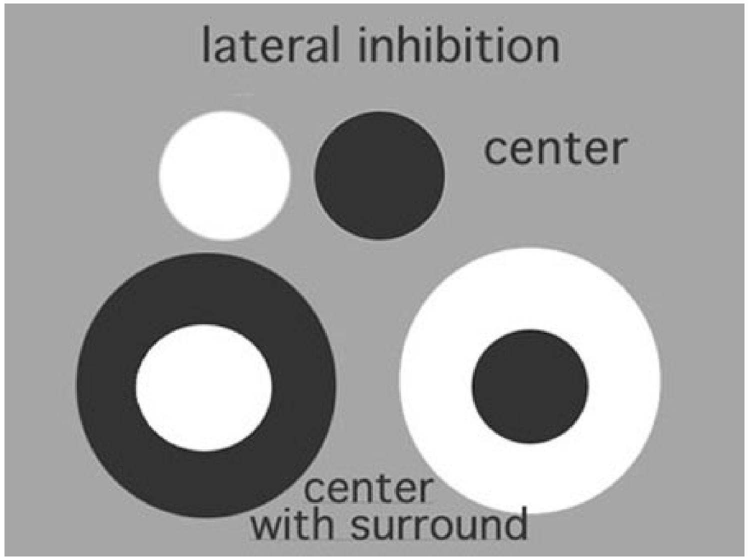



Fig. 10. Center-surround receptive fields can be ON center or OFF center with the oposite sign annular surround.

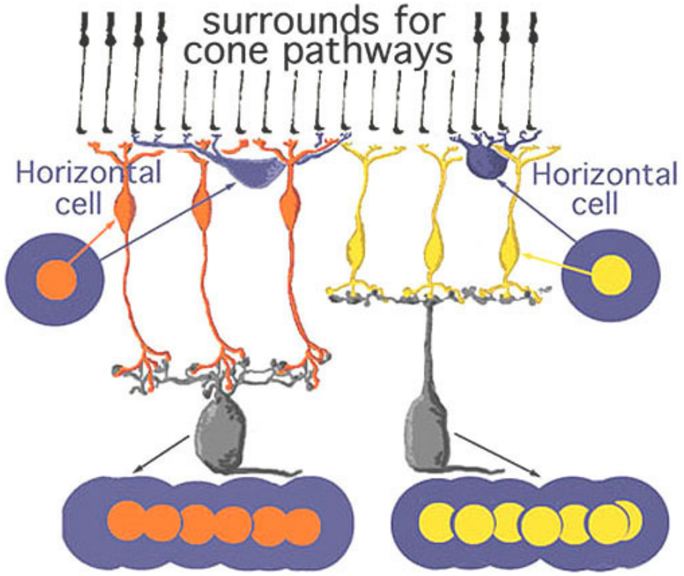



Fig. 12. Diagram of the organization of center-surround responses using horizontal cell circuitry to provide the antagonistic surround.

# Canny edge detector

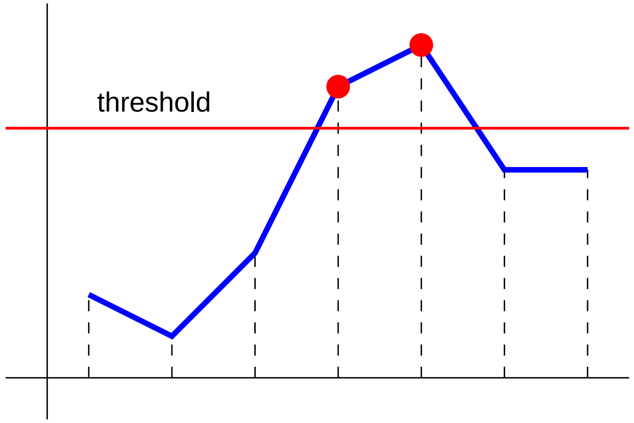
- A simple detector addressing the shortcomings of majority of (even simpler) detectors

Algorithm:

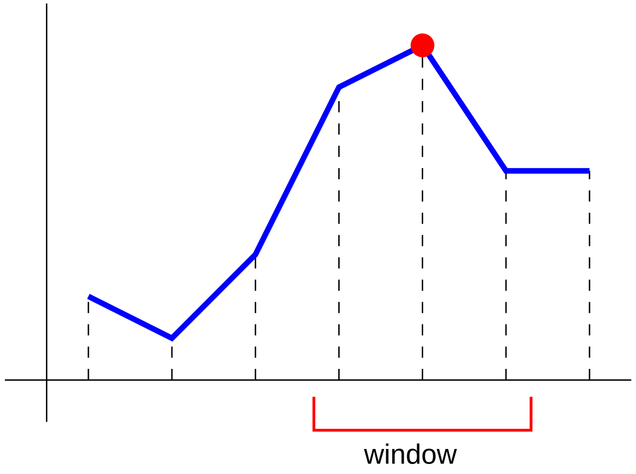
1. Compute gradient directions
2. For each pixel, compute smoothed 1D directional derivative in the gradient direction.
3. Find magnitude maxima of these derivatives
4. Get edgels by thresholding with hysteresis

# Finding maxima of derivative

- Why? Thresholding is not the way to go (leads to thick object boundaries)



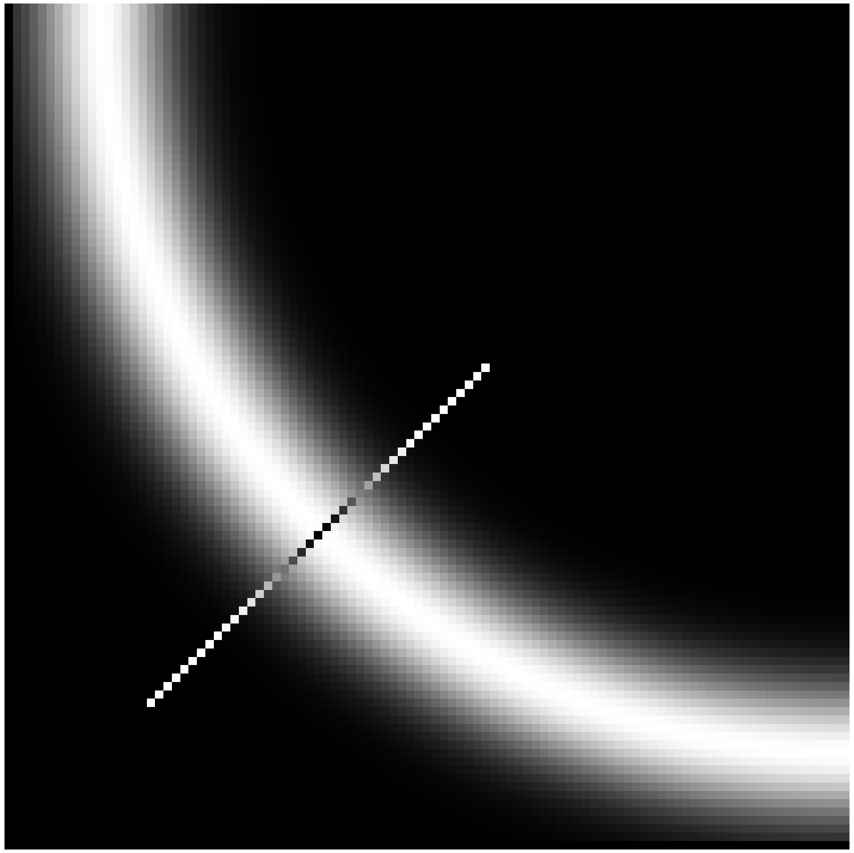

- It is better to localize the maximum of derivative (non-maxima suppression)

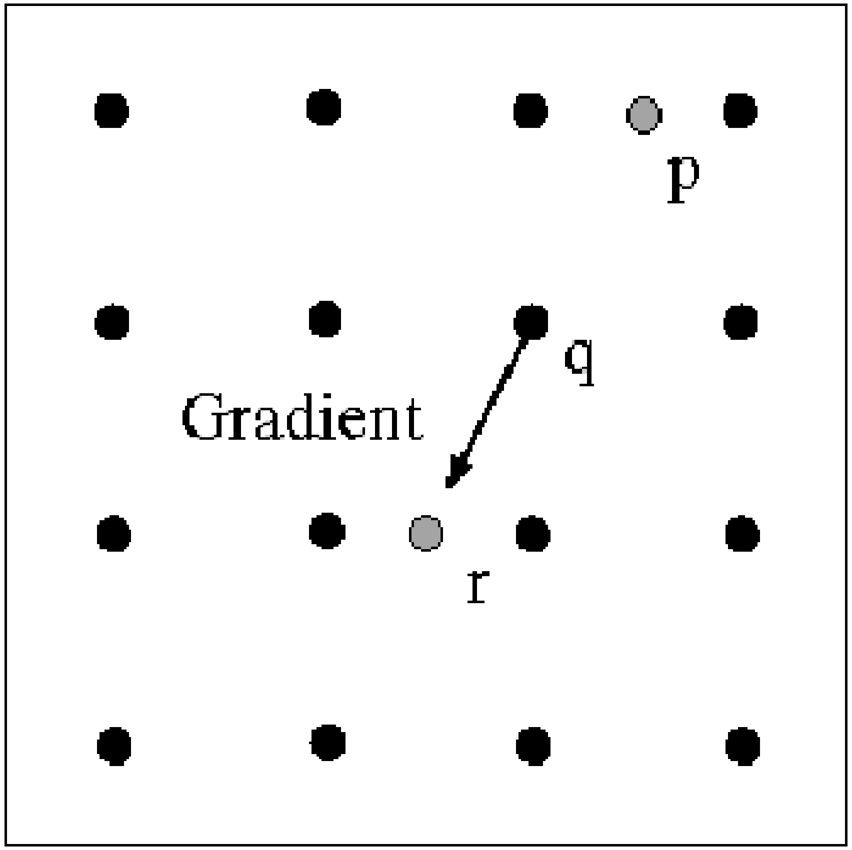



- Possible to localize the maximum with sub-pixel precision.

# .. in 2D

- look for maxima along the directio of gradient

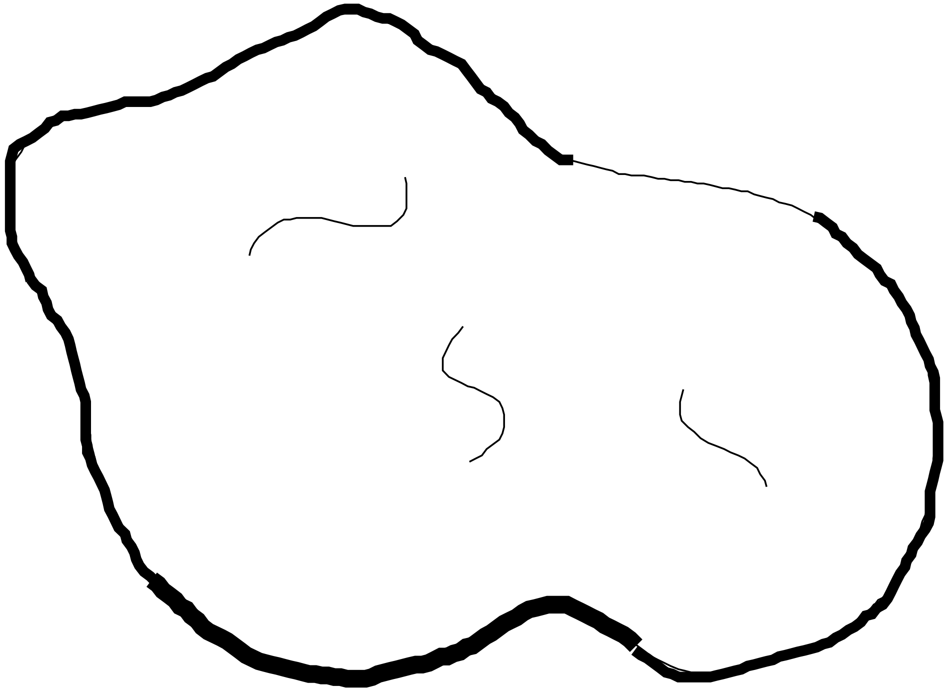




# Canny: Thresholding with hysteresis



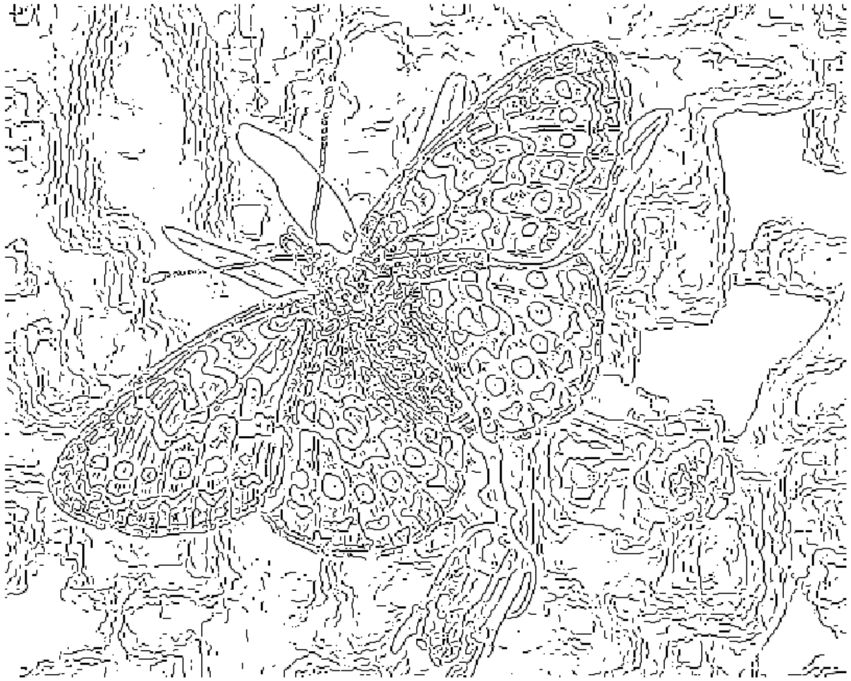
- Why? Want to supress short (usually unimportant) chains of edgels. At the same time, avoid fragmentation of long chains likely corresponding to object boundaries.
- This cannot be done by thresholding. The trick is to use two thresholds $T_1 < T_2$. Edges stronger than $T_2$ are automatically edges. Edges stronger than $T_1$ are linked if connected to stronger edges.



# Scale selection

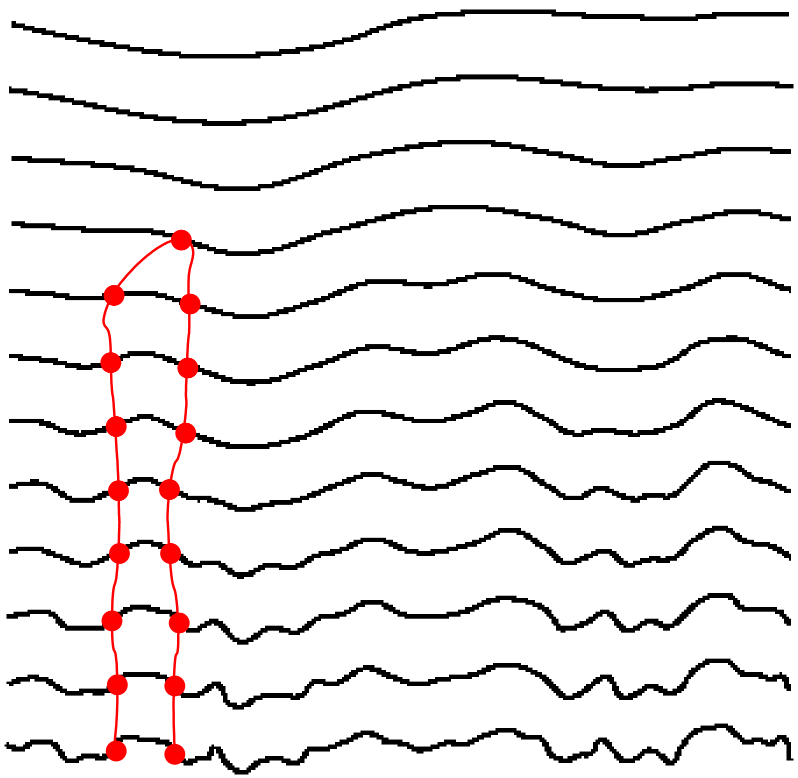
- Which $\sigma$ of the Gaussian should be selected for computing derivatives? The larger $\sigma$,
  - the more effective is the noise suppression,
  - the more weak edges disappear,
  - the worse is the edge localization.
- This problem is a general one, it is not associated just with Canny detection. It is the general problem during detection in images.

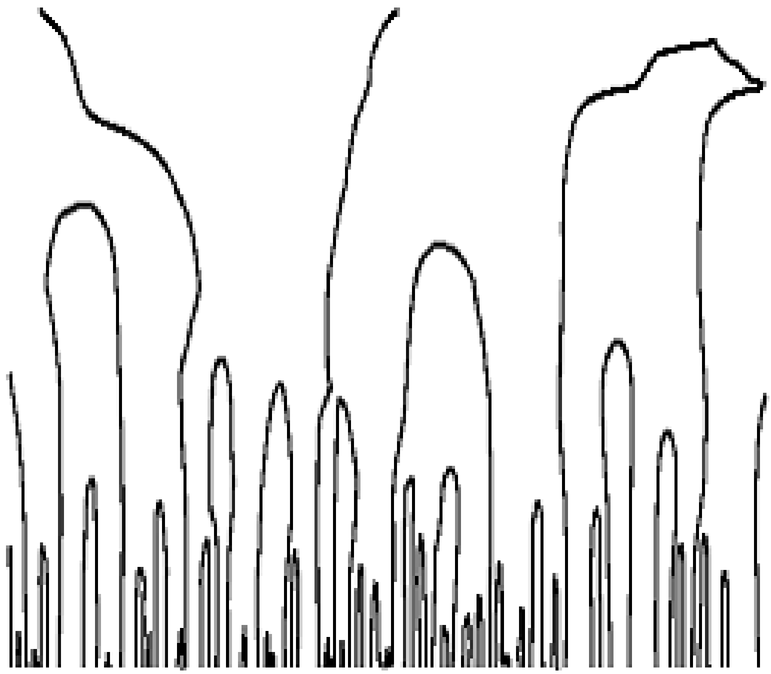
# . . . Example



# . . . Scale space

Scale space for 1D signal:





With increasing $\sigma$, an edge can annihilate, but but no new edge can possibly be created.

2D: Lindeberg (1994) Scale-Space Theory in Computer Vision, Kluwer Academic Publishers/Springer, Dordrecht, Netherlands, 1994.

# → Live Demo: Chaining edges to lines

- filtered by Laplace
- detected zero-crossings
- chained to lines