Lecture 18:
Feature Extraction I:
Edges

Contents

Introduction

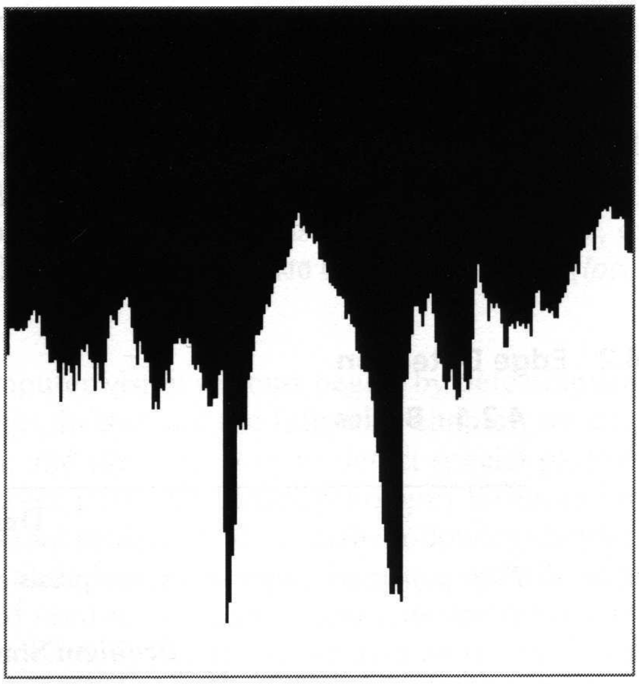
The Importance of Edges

- A strong change in the grey values within a neighbourhood indicates one of the most important image features, namely an edge.
- For the human visual system, edges
  - often provide the most relevant image information.
    This is why we can understand comics and use line drawings.
- In computer vision, edges
  - give a significantly sparser image representation than specifying the grey values of all pixels.
  - are assumed to comprise the object boundaries.
  - are a first step from a pixel-based image description *(low-level vision)* towards an automised understanding of the image content *(high-level vision)*.
- Edges can be detected with derivative operators.

Introduction (2)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

(a)

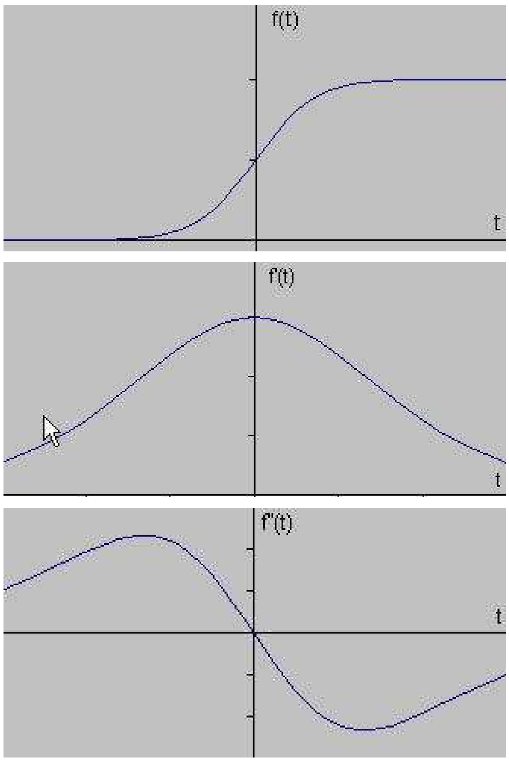


(b)

(a) Left: Image of size 325×237, from wich a 1-D signal along the horizontal scanline 56 has been extracted. **(b) Right:** Intensity profile along this scanline. The largest intensity jumps mark the boundaries of the hair region. Authors: E. Trucco and A. Verri (1998).

Introduction (3)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33



From top to bottom: A 1-D signal and its first and second derivative. Edges can be detected as locations where the magnitude of the first derivative is maximal, or as locations where the second derivative has a zero-crossing. Source: `http://www.pages.drexel.edu/~weg22/edge.html`.

Introduction (4)

Problem

- Small, but high-frequent fluctuations in the original signal can create very large perturbations in its derivatives.
- Example: The high-frequent 1-D perturbation

$$f(x) = \varepsilon \sin\left(\frac{x}{\varepsilon^2}\right)$$

becomes arbitrarily small in magnitude for $\varepsilon \to 0$. However, its derivative

$$f'(x) = \frac{1}{\varepsilon} \cos\left(\frac{x}{\varepsilon^2}\right)$$

exceeds all bounds !!!

Introduction (5)

- Alternative interpretation in the Fourier domain (Lecture 3):
Derivatives in the spatial domain become multiplications with the frequency in the Fourier domain:

$$\mathcal{F}\left[\frac{\partial^{n+m} f}{\partial x^n \partial y^m}\right](u,v) = (i2\pi u)^n \, (i2\pi v)^m \, \mathcal{F}[f](u,v).$$

Thus, high-frequent components (e.g. noise) are enormously amplified!

Remedy:

- Perform lowpass filtering before computing derivatives.
- very frequently used: Gaussian convolution (Lecture 11)

Numerical Approximation of Derivatives

Finite difference approximations of derivatives are obtained by a Taylor expansion with subsequent comparision of the coefficients.

Example

Approximate the second derivative u_i'' in pixel i with a stencil that takes into account the pixels $i-1$, i and $i+1$. The grid size is h. Compute the stencil weights.

Taylor expansion around pixel i:

$$\begin{aligned} u_{i-1} &= u_i - h u_i' + \frac{h^2}{2} u_i'' - \frac{h^3}{6} u_i''' + \frac{h^4}{24} u_i'''' - \frac{h^5}{120} u_i''''' + O(h^6) \\ u_i &= u_i \\ u_{i+1} &= u_i + h u_i' + \frac{h^2}{2} u_i'' + \frac{h^3}{6} u_i''' + \frac{h^4}{24} u_i'''' + \frac{h^5}{120} u_i''''' + O(h^6). \end{aligned}$$

Comparison the coefficients in

$$\begin{aligned} 0 \cdot u_i + 0 \cdot u_i' + 1 \cdot u_i'' &\stackrel{!}{=} \alpha_{-1} u_{i-1} + \alpha_0 u_i + \alpha_1 u_{i+1} \\ &= (\alpha_{-1} + \alpha_0 + \alpha_1) \cdot u_i \\ &+ h(-\alpha_{-1} + \alpha_1) \cdot u_i' \\ &+ \frac{h^2}{2} (\alpha_{-1} + \alpha_1) \cdot u_i'' + O(h^3) \end{aligned}$$

leads to the linear system

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2/h^2 \end{pmatrix}.$$

Its solution specifies the weights

$$\alpha_{-1} = \frac{1}{h^2}, \qquad \alpha_0 = -\frac{2}{h^2}, \qquad \alpha_1 = \frac{1}{h^2}.$$

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Plugging this into the Taylor expansion allows to quantify the error:

$$\frac{1}{h^2}\,u_{i-1} - \frac{2}{h^2}\,u_i + \frac{1}{h^2}\,u_{i+1} \;=\; u_i'' \;+\; \underbrace{\frac{h^2}{12}\,u_i'''' + O(h^4)}_{\text{error}}.$$

Due to the quadratic error term in the grid size h, this is called an approximation with *consistency order (Konsistenzordnung)* 2.
(We always assume, that the required derivatives exist and are bounded.)

Higher consistency orders give better accuracies. Discretisations must have at least the consistency order 1, in order to ensure that for $h \to 0$ the desired expression is indeed approximated. Otherwise, the method is called *inconsistent*.

Inconsistent discretisations are inacceptable!

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Derivative Filters in 1-D

The Most Important Approximations

first derivative:

$$u_i' \;=\; \frac{u_{i+1}-u_i}{h} \;+\; O(h) \qquad \text{forward difference}$$

$$u_i' \;=\; \frac{u_i-u_{i-1}}{h} \;+\; O(h) \qquad \text{backward difference}$$

$$u_i' \;=\; \frac{u_{i+1}-u_{i-1}}{2h} \;+\; O(h^2) \qquad \text{central difference}$$

second derivative:

$$u_i'' \;=\; \frac{u_{i+1}-2u_i+u_{i-1}}{h^2} \;+\; O(h^2) \qquad \text{central difference}$$

Central differences usually have a higher order of consistency:
The symmetry causes cancellation effects of the Taylor coefficients.

Derivative Filters in 1-D (2)

How to Improve the Order of Consistency

- By extending the stencil size, it is possible to increase the order of consistency (and the computational effort) to any desired number.
- Example:

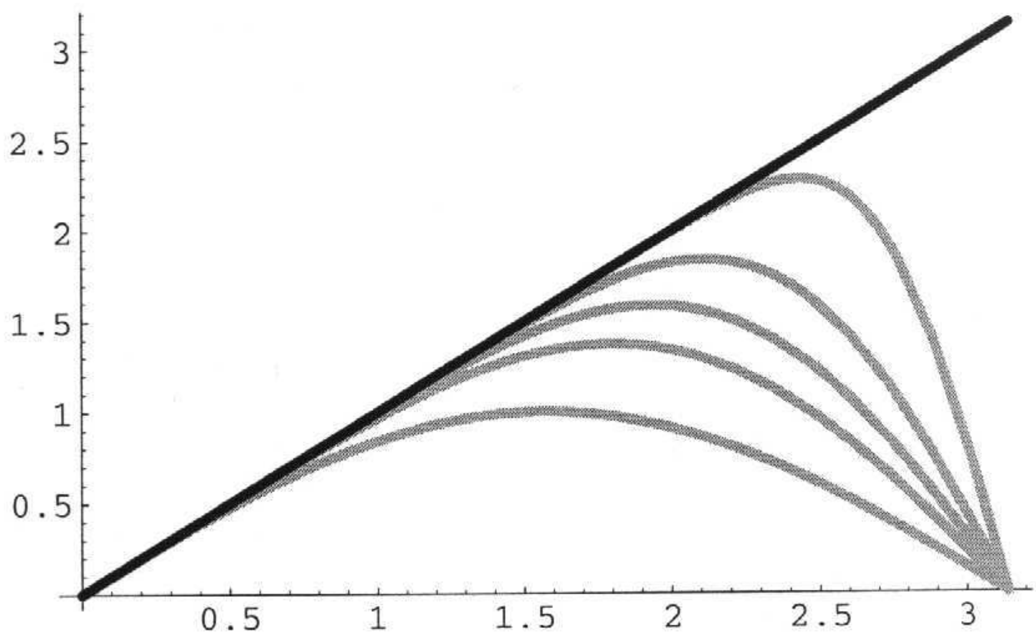
$$u_i' = \frac{u_{i+1} - u_{i-1}}{2h} + O(h^2)$$

$$u_i' = \frac{-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}}{12h} + O(h^4)$$

$$u_i' = \frac{u_{i+3} - 9u_{i+2} + 45u_{i+1} - 45u_{i-1} + 9u_{i-2} - u_{i-3}}{60h} + O(h^6)$$

- The better approximation quality is also visible in the Fourier domain.

Derivative Filters in 1-D (3)



Fourier spectrum of the ideal derivative operator (black) and its numerical approximations (grey). The numerical approximations show a lowpass effect, i.e. there is some smoothing along the direction of the derivative. Increasing mask size from 3 to 5, 7, 9 and 11 pixels one gets closer to the ideal derivative operator. Author: C. Schnörr (1999).

Derivative Filters in 2-D

First-Order Derivatives

- Basically, the 1-D masks can also be used in 2-D. Thus, for the first order derivatives, the following stencils have consistency order 2:

$$\partial_x \approx \frac{1}{2h} \boxed{\begin{array}{|c|c|c|} -1 & 0 & 1 \end{array}}\,, \qquad \partial_y \approx \frac{1}{2h} \boxed{\begin{array}{c} 1 \\ \hline 0 \\ \hline -1 \end{array}}$$

  (where the x axis goes from left to right, and the y axis from bottom to top)

- Problem:
  - The masks smooth in the direction of the derivative, but not orthogonal to it.
  - This suggests to introduce some smoothing perpendicular to the derivative direction, if one is interested in good isotropy.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

- The *Sobel operators* create such a perpendicular smoothing by convolving with the binomial kernel $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$:

$$\partial_x \approx \frac{1}{4} \boxed{\begin{array}{c} 1 \\ \hline 2 \\ \hline 1 \end{array}} * \frac{1}{2h} \boxed{\begin{array}{c|c|c} -1 & 0 & 1 \end{array}} = \frac{1}{8h} \boxed{\begin{array}{c|c|c} -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \end{array}}$$

$$\partial_y \approx \frac{1}{4} \boxed{\begin{array}{c|c|c} 1 & 2 & 1 \end{array}} * \frac{1}{2h} \boxed{\begin{array}{c} 1 \\ \hline 0 \\ \hline -1 \end{array}} = \frac{1}{8h} \boxed{\begin{array}{c|c|c} 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \end{array}}$$

- This does not deteriorate the consistency order (still 2).
- Sobel operators approximate a rotationally invariant expression such as the gradient magnitude $|\nabla u| = \sqrt{(\partial_x u)^2 + (\partial_y u)^2}$ in a better way.

Derivative Filters in 2-D (3)

Second Order Derivatives

- Standard approximation of the Laplacian $\Delta u = \partial_{xx} u + \partial_{yy} u$:

$$\Delta u_{i,j} = \frac{1}{h^2} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \; u_{i,j}$$

$$- \frac{1}{12} h^2 (\partial_{xxxx} u_{i,j} + \partial_{yyyy} u_{i,j}) \; + \; O(h^4).$$

- Problem: Although the Laplacian is rotationally invariant, the derivative expression $\partial_{xxxx} u_{i,j} + \partial_{yyyy} u_{i,j}$ in its leading error term is not.

Derivative Filters in 2-D (4)

- Better results are obtained with

$$\Delta u_{i,j} = \frac{1}{6h^2} \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array} \; u_{i,j}$$

$$- \frac{1}{12} h^2 (\partial_{xxxx} u_{i,j} + 2\,\partial_{xxyy} u_{i,j} + \partial_{yyyy} u_{i,j}) \; + \; O(h^4)$$

where the error term contains a rotationally invariant derivative operator, namely $\Delta(\Delta u)_{i,j}$.

Edge Detection with First Order Derivatives

Basic Idea:

- Convolve the initial image f with a Gaussian K_σ, in order to attenuate high frequencies:

$$u = K_\sigma * f$$

- Compute the gradient magnitude

$$|\nabla u| = \sqrt{(\partial_x u)^2 + (\partial_y u)^2}$$

  by approximating the derivatives with Sobel operators.

- Extract image edges as regions where $|\nabla u|$ exceeds a certain threshold T.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

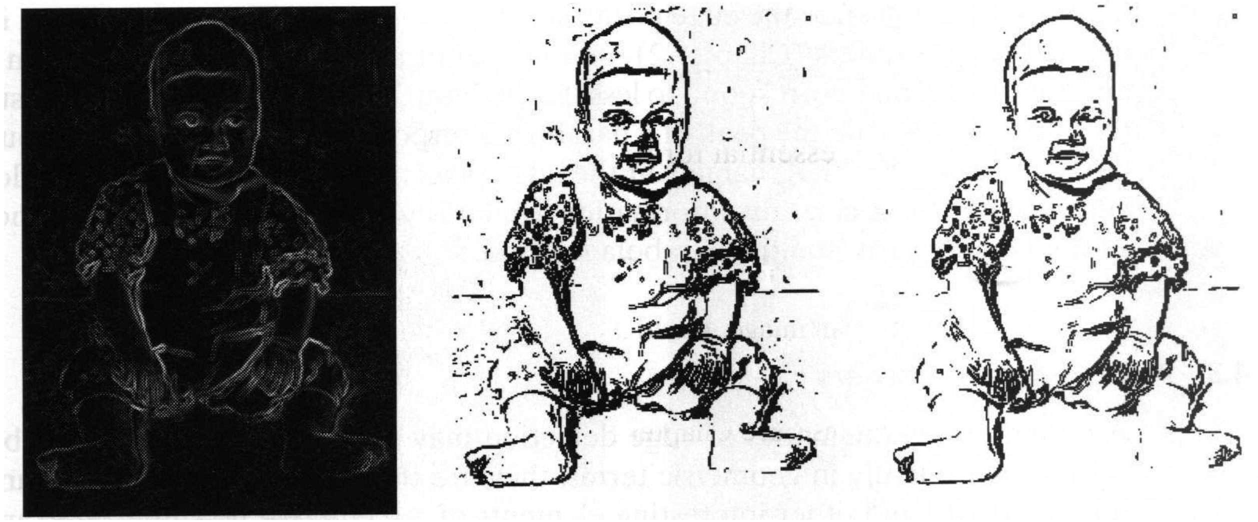
Advantage:

- The use of first order derivatives is more robust with respect to noise than computing second order derivatives.

Disadvantages:

- additional threshold parameter T (besides standard deviation σ of Gaussian)
- Some edges are too thick, others are below the threshold.
- does not create closed contours

Edge Detection with First Order Derivatives (3)



(a) Left: Gradient magnitude using Sobel operators. **(b) Middle:** Thresholding at $\sqrt{35}$ creates too thick edges. **(c) Right:** Thresholding at $\sqrt{50}$ destroys some edges, e.g. at the legs. Authors: E. Trucco, A. Verri (1998).

Edge Detection with First Order Derivatives (4)

The Canny Edge Detector

- popular edge detector with sophisticated postprocessing
- one of the edge detectors with the best performance
- proceeds in three steps and requires three parameters:
  - Gaussian standard deviation σ
  - two thresholds T_1, T_2

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

How Does the Canny Edge Detector Work ?

◆ *Gradient Approximation by Gaussian Derivatives:*

- For the Gaussian-smoothed image u, compute the edge magnitude $|\nabla u|$ and its orientation $\arctan(u_y/u_x)$.
- Identify edge candidates as locations where $|\nabla u|$ exceeds a low threshold T_1.

◆ *Nonmaxima Suppression:*

- Goal: thinning of edges to a width of 1 pixel
- In every edge candidate, consider the grid direction (out of 4 directions) that is "most orthogonal" to the edge.
- If one of the two neighbours in this direction has a larger gradient magnitude, remove the central pixel from the edge map.

◆ *Hysteresis Thresholding (Double Thresholding):*

- Goal: extract only relevant edges.
- Use points above an upper threshold T_2 as seed points for relevant edges.
- Add all neighbours that are above the lower threshold T_1.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Edge set after nonmaxima suppression with different standard deviations of the Gaussian. **From left to right:** $\sigma = 1, 2, 3$. Authors: E. Trucco, A. Verri (1998).

Edge set after nonmaxima suppression and hysteresis thresholding. **From left to right:** $\sigma = 1,\ 2,\ 3$. To improve visibility, the grey values have been inverted. Authors: E. Trucco, A. Verri (1998).

Edge Detection with Second Order Derivatives

Basic Idea:

- Perform Gaussian smoothing on the initial image: $u = K_\sigma * f$.
- Compute the Laplacian $\Delta u := \partial_{xx} u + \partial_{yy} u$
  (Laplacian-of-Gaussian (LoG), Marr–Hildreth–Operator).
- Extract edges as zero-crossings of the Laplacian.

Edge Detection with Second Order Derivatives (2)

Advantages:

- closed contours with minimal width
- no additional parameters besides the standard deviation σ of the Gaussian

Disadvantages:

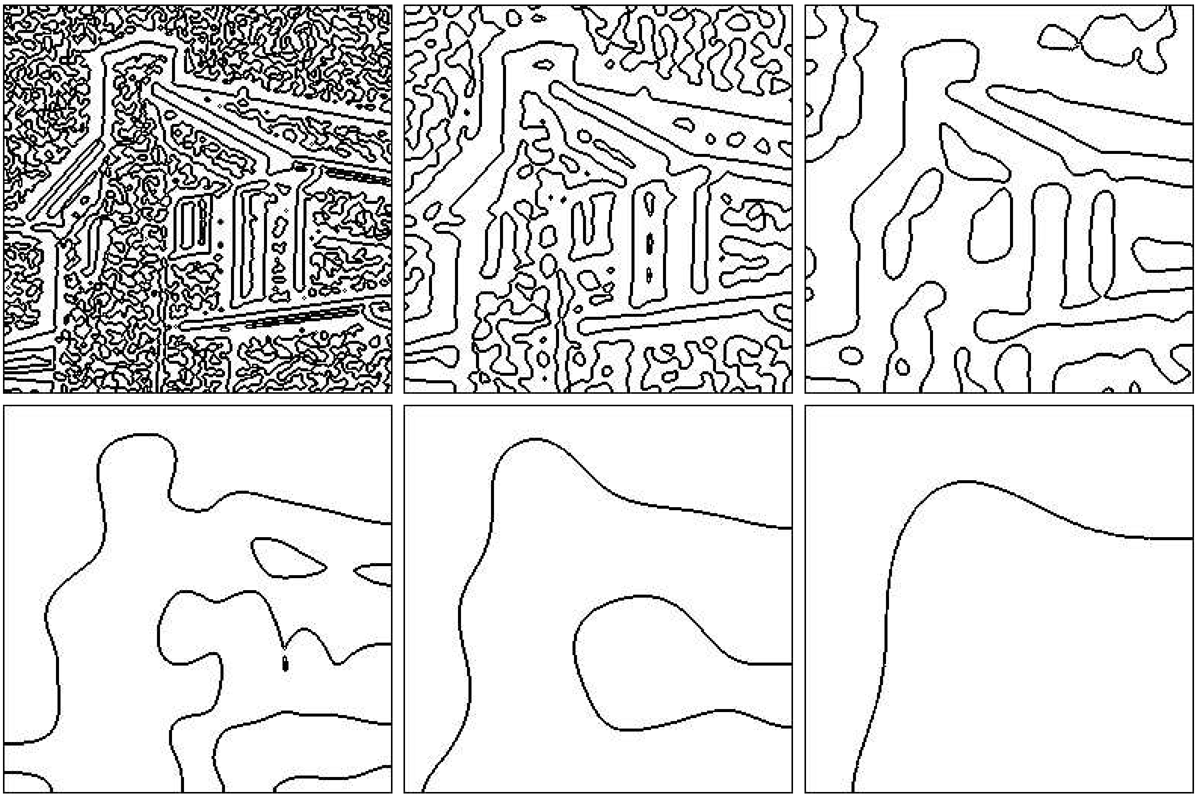
- false alarms: does not only detect maxima of the first derivative, but also minima
- Second order derivatives are more sensitive to noise than first order derivatives.
- Often strong Gaussian smoothing is required, leading to incorrect edge locations.

Edge Detection with Second Order Derivatives (3)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Gaussian smoothing of a test image (256×256 pixel). **From left to right and from top to bottom:** $\sigma = 2, 4, 8, 16, 32, 64$. Author: J. Weickert (2002).

Edge Detection with Second Order Derivatives (4)



Zero crossings of the Laplacian for the results from the previous page. Author: J. Weickert (2002).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

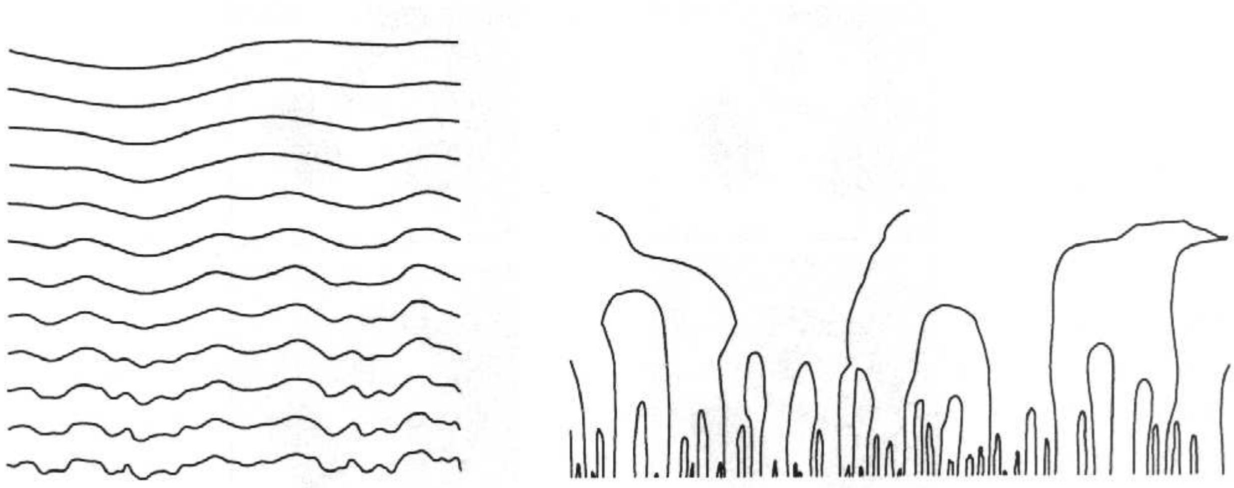
Edge Detection with Second Order Derivatives (5)

Interesting Observation: (Witkin 1983)

- Structures that can be detected at a coarse scale σ can be traced back to smaller scales in order to improve their localisation *(causality)*.
- This has led to the notion of *scale-space (Skalenraum)* in the western world: Embed an image in a continuum of more and more smoothed versions of it.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Edge Detection with Second Order Derivatives (6)



(a) Left: Evolution of a signal in Gaussian scale-space. The scale is increasing from bottom to top. **(b) Right:** Corresponding evolution of the zero-crossings of the Laplacian. The vertical axis denotes scale, the horizontal axis describes the location. Authors: T. Lindeberg, B. ter Haar Romeny (1994). Adapted from A. P. Witkin.

Summary (1)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Summary

- Derivative approximations can be used for detecting edges.
- Differentiation should be stabilised by using a lowpass filter.
- The weights of the derivative approximations can be computed via a Taylor expansion with subsequent comparision of coefficients.
- The mask size has an influence on the achievable order of consistency.
- 2-D derivative operators should have good rotation invariance.
  Example: Sobel operators.
- Important edge detector using first order derivatives: gradient magnitude
- The Canny filter uses nonmaxima suppression and hysteresis thresholding as postprocessing steps.
- Important edge detector using second-order derivatives:
  zero-crossings of the Laplacian (Marr–Hildreth operator)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Literature

- H. R. Schwarz, N. Köckler: *Numerische Mathematik.* Fünfte Auflage, Teubner, Stuttgart, 2004.
  (very recommendable numerical analysis book dealing also with finite difference approximations; also available in English)
- E. Trucco, A. Verri: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Englewood Cliffs, 1998.
  (for gradient-based edge detection, including the Canny detector)
- O. Faugeras: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
  (Chapter 4 give an excellent and detailed description of edge detection methods.)
- J. Canny: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679–698, Nov. 1986.
  (Canny's original paper)
- D. Marr, E. Hildreth: Theory of edge detection. *Proceedings of the Royal Society of London B*, Vol. 207, pp. 187–217, 1980.
  (edge detection by zero-crossings of the Laplacian)
- A. P. Witkin: Scale-space filtering. *Proc. Eighth International Joint Conference on Artificial Intelligence* (Karlsruhe, West Germany, August 1983), Vol. 2, pp. 945–951, 1983.
  (made the scale-space idea popular in the western world)

M I A

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Assignment P5 – Programming Work

Please download the required files from the webpage

`http://www.mia.uni-saarland.de/Teaching/ipcv07.shtml`

into your own directory. You can unpack them with the command `tar xvzf Ex05.tgz`.

Problem 1 (Deconvolution with Wiener Filtering) (10+3+3+4)

The object code `deconv.o` is an almost complete deblurring programme based on manipulations in the Fourier domain. It only requires to specify the deblurring function in the subroutine `deconv-filter.c`.

(a) Supplement the missing code for Wiener filtering. However, do not use C/C++ libraries such as `complex.h` that offer computations with complex data types. You can compile the programme via

```
gcc -O2 -o deconv deconv.o deconv-filter.c -lm .
```

(b) The image `bus1.pgm` has been blurred with the small Gaussian `k-gauss1.pgm`. Use the compiled programme to perform a deblurring. What is a good value for K ?

(c) Do the same experiment with the image `bus2.pgm` and its corresponding blurring kernel `k-gauss2.pgm`. What is a good value of K in this case ?

(d) Finally, consider the image `hogblur.pgm` which is a digital photo with simulated motion blur corresponding to the kernel `k-motion.pgm`. Try to compensate for the motion blur by finding a suitable value for K. Why does deblurring take so long in the case of this image?

Submission

Please remember that up to three people from the same tutorial group can work and submit their results together. For submitting the files rename the main directory `Ex05` to `Ex05_<your_name>` and use the command

```
tar czvf Ex05_<your_name>.tgz Ex05_<your_name>
```

to pack the data. The directory that you pack and submit should at least contain the following files:

- the source code for the Wiener filter in `deconv-filter.c`.
- the three deblurred images from the tasks (b)-(d)
- a text file README that
  - states the used values for the parameter K in the tasks (b)-(d),
  - answers the question w.r.t. the runtime in task (c), and
  - contains information on all people working together for this assignment.

Please make sure that only your final version of the programmes and images are included.
Submit the file via e-mail to your tutor via the address:

`ipcv-xx@mia.uni-saarland.de`

where xx is either `t1`, `t2`, `t3`, `t4`, `w1` or `w2` depending on your tutorial group.

Deadline for submission: Tuesday, January 15, 10 am (before the lecture)