Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

Recently on IPCV ...

- Image derivatives are useful for detecting features such as edges.
- Differentiation is dangerous. It can be stabilised with a lowpass filter.
- The weights of the discrete derivative approximations can be computed via a Taylor expansion with subsequent comparision of coefficients.
- The order of consistency can be increased by larger stencils.
- The continuous Fourier transform allows to analyse the frequency-dependent approximation quality.
- 2D derivative operators should have good rotation invariance.
  Example: Sobel operators.

Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

Why are Edges Important?

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

Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

A Baseline Edge Detector

- Convolve the initial image f with a Gaussian K_σ:

$$u = K_\sigma * f.$$

  This stabilises differentiation and makes it more robust under noise.

- Compute the gradient magnitude

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

  by approximating the derivatives with Sobel operators (see previous lecture).

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

A Baseline Edge Detector (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

Top left: Original image, 256×256 pixels. **Top right:** Gradient magnitude of the Gaussian-smoothed image ($\sigma = 1$). **Bottom left:** After thresholding with $T = 10$. For better visualisation, values larger or equal than T are depicted in black. **Bottom right:** Same with $T = 20$. Author: J. Weickert.

A Baseline Edge Detector (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

Advantage

- The Gaussian convolution offers reasonable robustness against noise.

Disadvantages

- two parameters: Gaussian standard deviation σ, threshold T
- Some edges may be too thick. Others may be below the threshold.

Remarks

- A suitable value for T strongly depends on the value of σ:
  Larger σ-values require smaller T-values.
- Thus, it is convenient to select T as a certain *quantile* of the histogram of $|\boldsymbol{\nabla} u|$.
- Example:
  The 0.8 quantile is the smallest number T with $|\boldsymbol{\nabla} u| \leq T$ for $80\ \%$ of all pixels.

Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

The Canny Edge Detector

- popular edge detector with sophisticated postprocessing
- better than our baseline edge detector
- proposed by John Canny in 1986
- still a prototype of a well-performing edge detector
- proceeds in three steps and requires three parameters:
  - Gaussian standard deviation σ
  - two thresholds T_1, T_2 (found via two quantiles q_1, q_2)

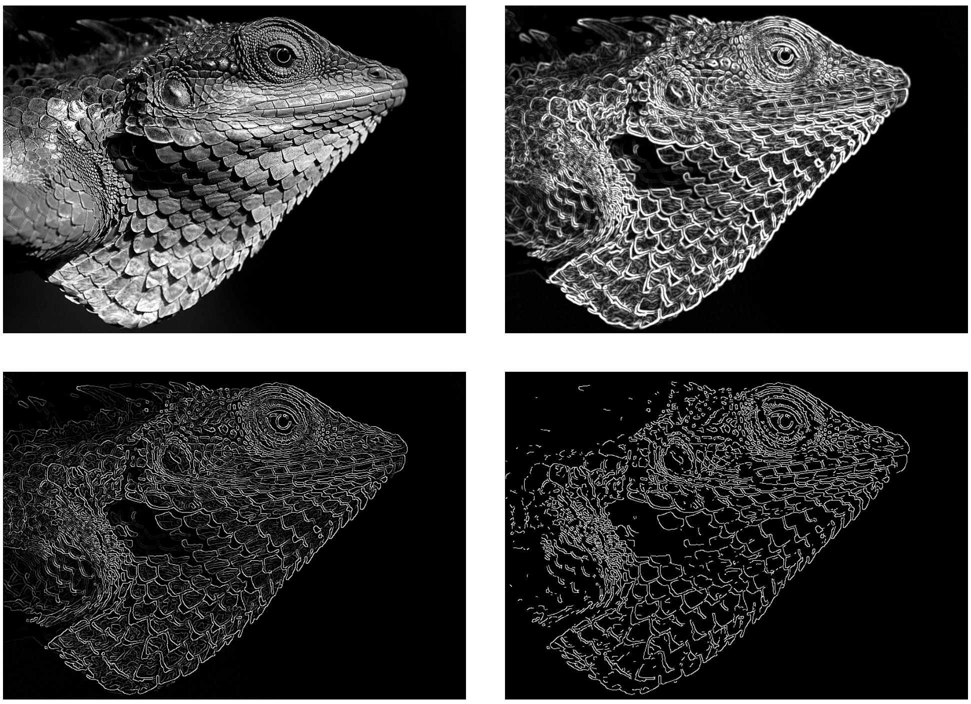
The Canny Edge Detector (2)

How Does the Canny Edge Detector Work ?

- *Gradient Approximation with Gaussian Derivatives:*
  - For Gaussian-smoothed image u, compute magnitude $|\boldsymbol{\nabla} u|$ and orientation angle $\phi = \arg(\boldsymbol{\nabla} u)$ of $\boldsymbol{\nabla} u$ (cf. also Lecture 4, polar coordinates).
  - Choose quantiles $q_1 < q_2$ for $|\boldsymbol{\nabla} u|$, find corresponding thresholds $T_1 < T_2$.
    Pixels with $T_1 \leq |\boldsymbol{\nabla} u| < T_2$ are potential edge pixels.
    Pixels with $|\boldsymbol{\nabla} u| > T_2$ are guaranteed edge pixels.
- *Nonmaxima Suppression:*
  - Goal: thinning of edges to a width of 1 pixel
  - In every pixel with nonzero gradient, 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, mark the central pixel for removal.
  - After passing through all candidates, remove marked pixels from edge map.
- *Hysteresis Thresholding (Double Thresholding):*
  - Goal: extract relevant edges with long edge contours
  - Use pixels above the upper threshold T_2 as seed points for relevant edges.
  - Add neighbours exceeding lower threshold T_1. Iterate step for added pixels.

The Canny Edge Detector (3)

greyscale image — gradient magnitude $|\boldsymbol{\nabla} u|$



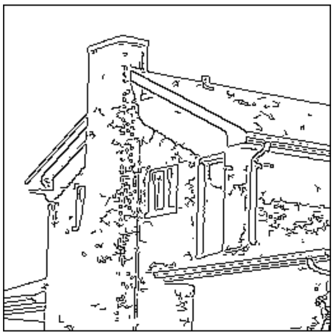
after nonmaxima suppression — after hysteresis thresholding

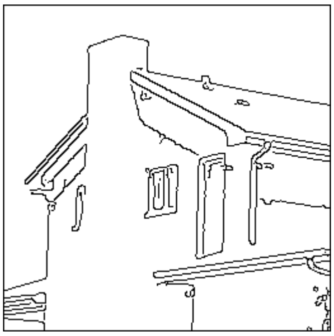
Visualisation of the three steps in the Canny edge detection algorithm.
From `https://en.wikipedia.org/wiki/Canny_edge_detector`.

The Canny Edge Detector (4)

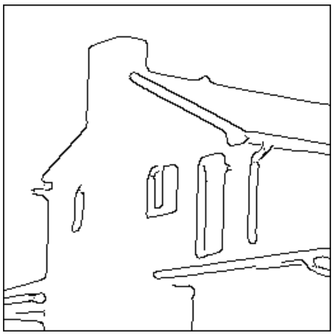
original, 256×256

$\sigma = 0.5$





$\sigma = 1$



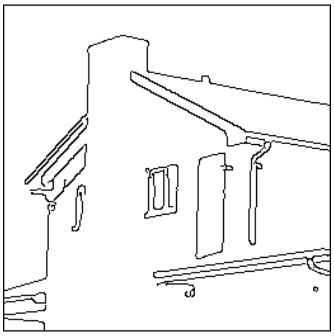
$\sigma = 2.5$

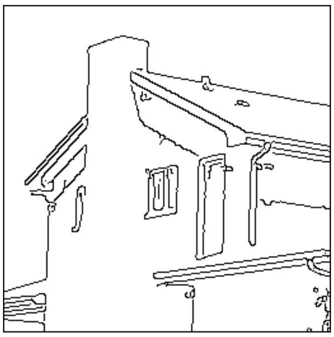
Influence of the Gaussian standard deviation σ. Larger σ-values give less details and more delocalisations. The thresholds T_1 and T_2 are set to the 0.70 resp. 0.85 quantiles of $|\boldsymbol{\nabla} u|$. Author: J. Weickert.

The Canny Edge Detector (5)

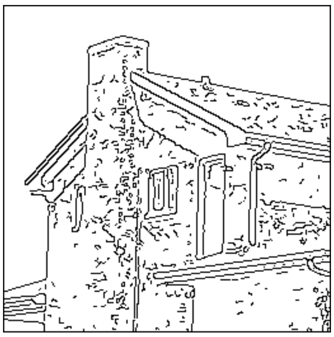
original, 256×256

0.95 quantile





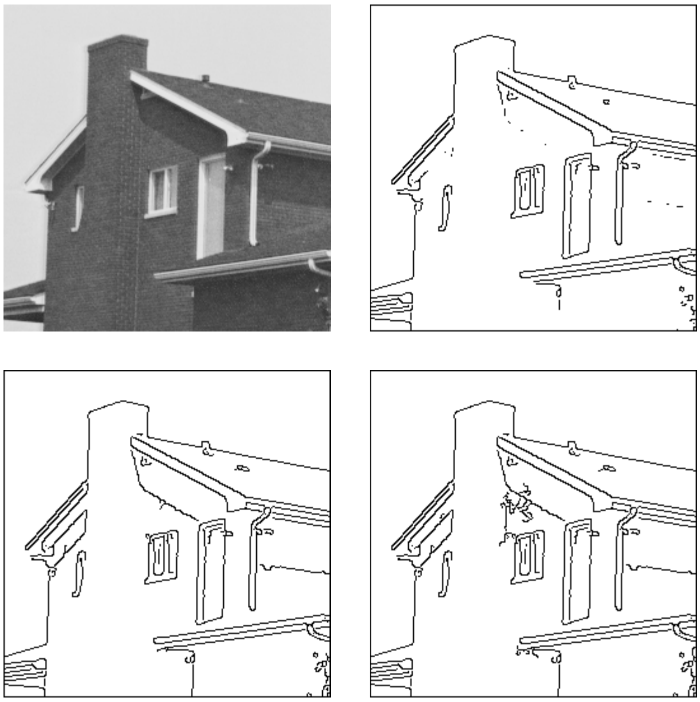
0.85 quantile



0.75 quantile

Influence of the upper threshold T_2 ($\sigma = 1$, lower threshold T_1 at 0.7 quantile). With a good value of T_2, all relevant edges are detected, and irrelevant ones are suppressed. Author: J. Weickert.

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



Influence of the lower threshold T_1 ($\sigma = 1$, upper threshold T_2 at 0.85 quantile). With a good value for T_1, edges are neither too short nor too long. Author: J. Weickert.

Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

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

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

Why are Corners Important ?

- Corners are sparser features than edges.
  They are useful whenever point-like features are preferred over line-like features.
- Corners can help solving correspondence problems in computer vision:
  - finding correspondences in stereo image pairs
  - matching medical images (so-called registration)

  Edges would be ambiguous in this context.
- However, to be able to detect corners, we first
  - have to remember some basics from linear algebra,
  - and use them to construct a good detector of the local image structure.

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

Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

Basics From Linear Algebra

- **What are eigenvalues and eigenvectors?**
  - Consider some $n \times n$ matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.
    Idea: Find characteristic directions where $\boldsymbol{A}$ acts like a simple rescaling.
  - Assume there is a number $\lambda \in \mathbb{C}$ and a vector $\boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{v} \neq \boldsymbol{0}$ with $\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$.
  - Then the scaling factor λ is called an *eigenvalue (Eigenwert)* of $\boldsymbol{A}$.
  - The characteristic direction $\boldsymbol{v}$ is its associated *eigenvector (Eigenvektor).*
    We have to exclude $\boldsymbol{v} = \boldsymbol{0}$, since $\boldsymbol{A}\,\boldsymbol{0} = \lambda\,\boldsymbol{0}$ trivially holds for any λ.

- **A simple way to obtain the sum and product of all eigenvalues.**
  - trace (sum of diagonal elements) of a matrix: sum of its eigenvalues
  - The determinant of a matrix is the product of its eigenvalues.

MIA
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

- **How can we compute eigenvalues?**
  - The basic equation $\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$ can be rewritten as $(\boldsymbol{A} - \lambda \boldsymbol{I})\,\boldsymbol{v} = \boldsymbol{0}$.
    Here $\boldsymbol{I} \in \mathbb{R}^{n \times n}$ denotes the unit matrix.
  - Eigenvectors are nontrivial solutions $(\boldsymbol{v} \neq \boldsymbol{0})$ of $(\boldsymbol{A} - \lambda \boldsymbol{I})\,\boldsymbol{v} = \boldsymbol{0}$.
    Thus, the matrix $\boldsymbol{A} - \lambda \boldsymbol{I}$ must be singular.
    Therefore, its determinant $\det(\boldsymbol{A} - \lambda \boldsymbol{I})$ must vanish.
  - This shows how we can find the eigenvalues:
    Eigenvalues are zeroes of the *characteristic polynomial*
    $p(\lambda) := \det(\boldsymbol{A} - \lambda \boldsymbol{I})$.
  - For $n = 2$, we solve a quadratic equation in λ. This case is relevant for us.
    For $n \geq 3$, usually numerical algorithms are used. Many methods exist.

- **How can we compute eigenvectors?**
  - Assume we have computed an eigenvalue λ.
  - Then we find its eigenvector $\boldsymbol{v}$ as a nontrivial solution of $(\boldsymbol{A} - \lambda \boldsymbol{I})\,\boldsymbol{v} = \boldsymbol{0}$.
  - Thus, eigenvectors are only defined up to a nonzero scaling factor.
  - Often one restricts their norm $|\boldsymbol{v}|$ to 1 (as we always do in our class).

Basics From Linear Algebra (3)

- **Is there a particularly nice scenario?**
  - Yes, if $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric.
  - Then all n eigenvalues of $\boldsymbol{A}$ are real-valued.
  - One can also find n eigenvectors that create an orthonormal basis of $\mathbb{R}^n$.
- **A frequently used definition.**
  - Let a symmetric matrix have only positive (resp. nonnegative) eigenvalues.
  - Then it is called *positive definite* (resp. *positive semidefinite*).
- **We will need the following result many times:**
  - Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be symmetric. Then $q(\boldsymbol{x}) := \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}$ is called its *quadratic form.*
  - Among all vectors $\boldsymbol{x} \in \mathbb{R}^n$ with norm 1, $q(\boldsymbol{x})$ is maximised (minimised) by the eigenvector with the largest (smallest) eigenvalue:

$$\lambda_{\text{min}} \underbrace{\boldsymbol{v}_{\text{min}}^\top \boldsymbol{v}_{\text{min}}}_{1} \le \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \le \lambda_{\text{max}} \underbrace{\boldsymbol{v}_{\text{max}}^\top \boldsymbol{v}_{\text{max}}}_{1} .$$

Outline

Lecture 13: Linear Filters III: Detection of Edges and Corners

Contents

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

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

The Structure Tensor

Motivation

- So far we have analysed edges.
  The gradient information was sufficient to give the local structure information.
- Now we would like to analyse corners.
  The maximal contrast direction changes strongly in the vicinity around a corner.
- Thus, we need a concept to measure average contrast along a specified direction.
  The averaging should be done over a user-specified scale.

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

How can We Model This?

- Let us specify a direction by a unit vector $\boldsymbol{n}$.
- We model *local contrast along $\boldsymbol{n}$* by the squared directional derivative $(\boldsymbol{n}^\top \boldsymbol{\nabla} u)^2$.
  The square guarantees that the sign of the directional derivative does not matter.
- We want a weighted average of this directional contrast in a vicinity of scale ρ.
  To this end, we convolve with a Gaussian K_ρ with standard deviation ρ.
- Thus, the *average local contrast in direction $\boldsymbol{n}$* is measured by

$$\begin{aligned} E(\boldsymbol{n}) &= K_\rho * \left((\boldsymbol{n}^\top \boldsymbol{\nabla} u)^2\right) \\ &= K_\rho * \left(\boldsymbol{n}^\top \boldsymbol{\nabla} u \, \boldsymbol{\nabla} u^\top \boldsymbol{n}\right) \\ &= \boldsymbol{n}^\top \underbrace{K_\rho * \left(\boldsymbol{\nabla} u \, \boldsymbol{\nabla} u^\top\right)}_{n \times n \text{ matrix}} \boldsymbol{n}\,. \end{aligned}$$

The Structure Tensor (3)

- This is a quadratic form with the symmetric matrix

$$J_\rho(\nabla u) \;:=\; K_\rho * (\nabla u\, \nabla u^\top) \;=\; \begin{pmatrix} K_\rho * (u_x^2) & K_\rho * (u_x u_y) \\ K_\rho * (u_x u_y) & K_\rho * (u_y^2) \end{pmatrix}.$$

- J_ρ is called *structure tensor (Strukturtensor)* (Förstner/Gülch 1987).
- The standard deviation ρ of the Gaussian K_ρ determines its locality.
  It is the scale over which directional information is integrated.
  Therefore one calls it *integration scale.*

The Structure Tensor (4)

What Does the Structure Tensor Tell Us?

- We have seen that J_ρ is symmetric.
  Thus, it has orthonormal eigenvectors v_1, v_2 and real-valued eigenvalues λ_1, λ_2.
- One can even show that J_ρ is positive semidefinite. Let w.l.o.g. $\lambda_1 \geq \lambda_2 \geq 0$.
- Our quadratic form $E(n) = n^\top J_\rho n$ models the average local contrast along n.
  Thus, eigenvectors v_1, v_2 of J_ρ are directions of largest/smallest local contrast.
- For these normalised eigenvectors we obtain

$$E(v_i) \;=\; v_i^\top J_\rho v_i \;=\; \lambda_i\, v_i^\top v_i \;=\; \lambda_i\,.$$

  Therefore, eigenvalues λ_1, λ_2 measure average local contrast along v_1, v_2.
- The eigenvalues allow to interpret the local image structure:

constant areas:	$\lambda_1 = \lambda_2 = 0$	(no large eigenvalues)
straight edges:	$\lambda_1 \gg \lambda_2 = 0$	(one large eigenvalue)
corners:	$\lambda_1 \geq \lambda_2 \gg 0$	(two large eigenvalues)

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

Lecture 13:
Linear Filters III: Detection of Edges and Corners

Contents

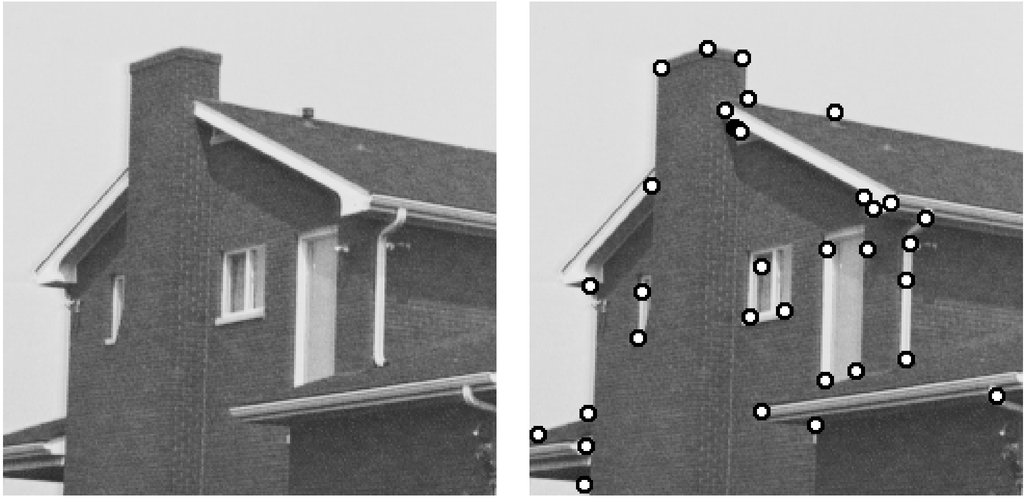
M I A

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

Corner Detection

- Consider a Gaussian-smoothed version $u = K_\sigma * f$ of the original image f.
  This makes corner detection robust w.r.t. noise/irrelevant small scale features.
- Compute the structure tensor $\boldsymbol{J}_\rho(\boldsymbol{\nabla} u) = K_\rho * (\boldsymbol{\nabla} u \, \boldsymbol{\nabla} u^\top)$.
  Convolving with K_ρ averages directional information in neighbourhood of scale ρ.
- Make sure that the integration scale ρ is larger than the noise scale σ.
- Corners are locations where the structure tensor has two large eigenvalues.
  There are different ways how to detect this.
- The *corner detector of Rohr* (1987) is particularly elegant and works well:
  - idea: measure cornerness with $\det \boldsymbol{J}_\rho = j_{1,1}\, j_{2,2} - j_{1,2}^2$
  - effective: involves both eigenvalues λ_1 and λ_2, since $\det \boldsymbol{J}_\rho = \lambda_1 \lambda_2$;
    only large when both are large (zero at edges where $\lambda_2 = 0$)
  - simple: does not even require to compute the eigenvalues explicitly
  - criterion: $\det \boldsymbol{J}_\rho$ must be locally maximal and exceed some threshold T.

Corner Detection (2)



Left: Original image, 256×256 pixels. **Right:** Corners detected with the method of Rohr, using the structure tensor with $\sigma = 2$, $\rho = 4$, and a cornerness threshold of $T = 100$. By adjusting T, one can steer the number of corners. Author: J. Weickert.

Corner Detection (3)

Extensions

- Besides corners, many other local feature descriptors have been advocated, e.g.
  - *SIFT: Scale-Invariant Feature Transform* (Lowe 2004)
  - *SURF: Speeded Up Robust Features* (Bay et al. 2008)
- They are substantially more advanced and offer higher robustness.
- These are discussed in more detail in the course Advanced Image Analysis.

MIA
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

Summary

- Gaussian-smoothed derivatives can be used for detecting edges and corners.
- Edges are locations with large gradient magnitude.
- The Canny filter is one of the best methods for edge detection.
  It uses nonmaxima suppression and hysteresis thresholding as postprocessing steps.
- The structure tensor allows a robust description of local image structure:
  - Its eigenvectors specify the local structure directions.
  - The eigenvalues give the average contrast in these directions.
- Corners are locations where both eigenvalues of the structure tensor are large.
  The determinant of the structure tensor is a simple measure for this.

References

- O. Faugeras: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
  (Chapter 4 gives a good 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)
- H. Anton, C. Rorres, A. Kaul: *Elementary Linear Algebra. Applications Version*. 12th Edition, Wiley, 2019.
  (thick, but very well readable; also available in German: H. Anton: Lineare Algebra. *Spektrum, Heidelberg, 1998.)*
- G. Strang: *Introduction to Linear Algebra*. Fifth Edition, Wellesley-Cambridge Press, 2016.
  (excellent linear algebra book; see Section 6.1 for eigenvalues and eigenvectors)
- W. Förstner, E. Gülch: A fast operator for detection and precise location of distinct points, corners and centres of circular features. *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data* (Interlaken, Switzerland, June 1987), pp. 281–305, 1987.
  (one of the early papers on the structure tensor)
- K. Rohr: Localization properties of direct corner detectors. *Journal of Mathematical Imaging and Vision*, Vol. 4, 139–150, 1994.
  (compares various corner detectors)

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

References (2)

- E. Trucco, A. Verri: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Englewood Cliffs, 1998.
  (computer vision book dealing also with corner detection)
- D. G. Lowe: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.
  (detailed description of SIFT features)
- H. Bay, A. Ess, T. Tuytelaars, L. Van Gool: SURF: Speeded up robust features. *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346–359, 2008.
  (faster alternative to SIFT)