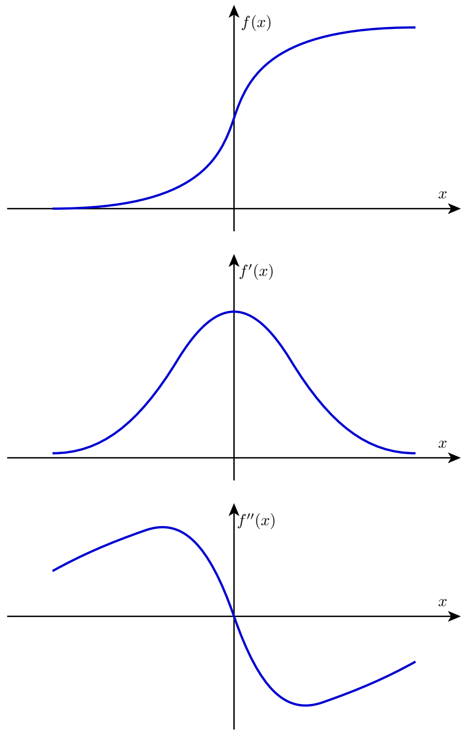
Lecture 13:
Linear Filters III:
Detection of Edges and Corners

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 comics and use line drawings.
- In computer vision, edges
  - are assumed to comprise the object boundaries.
  - belong to the most important image features.
  - 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 derivative operators.
  We can use first or second order derivatives.

Why are Edges Important? (2)



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. Author: T. Schneevoigt.

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 34 35

Edge Detection with First Order Derivatives (1)

Edge Detection with First Order Derivatives

Baseline Method

- To attenuate high frequencies, convolve the initial image f with a Gaussian K_σ:

$$u = K_\sigma * f$$

- 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.

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 34 35

Edge Detection with First Order Derivatives (2)

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.

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 34 35

Edge Detection with First Order Derivatives (3)

Advantage

- First order derivatives are more robust to noise than second order derivatives.

Disadvantages

- two parameters: Gaussian standard deviation σ, threshold T
- Some edges may be too thick, others may be below the threshold.
- In general, one cannot expect to obtain closed contours.

Remarks

- A suitable value for T strongly depends on the value of σ.
- It can be convenient to select T as a certain *quantile* of the histogram of $|\nabla u|$.
- Example:
  The 0.8 quantile is the smallest number T with $|\nabla u| \leq T$ for $80\ \%$ of all pixels.

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 34 35

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 34 35

The Canny Edge Detector

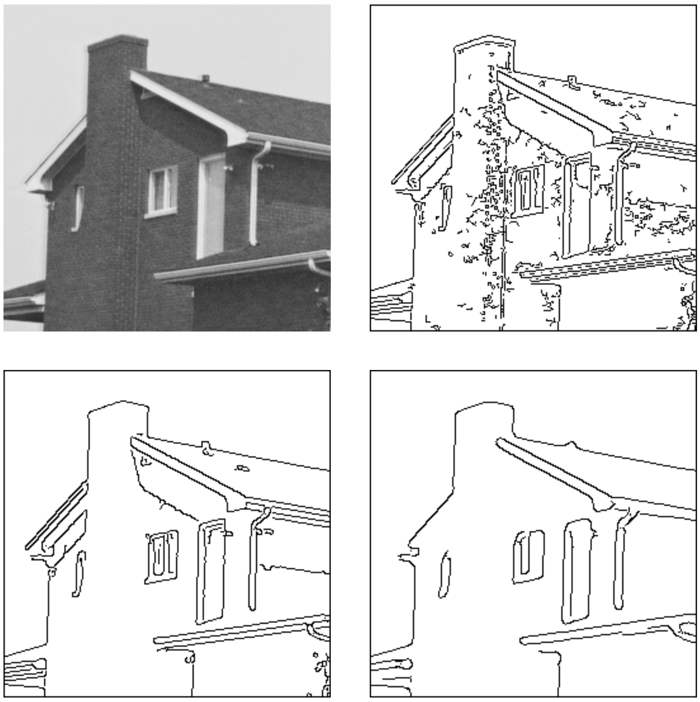
- popular edge detector with sophisticated postprocessing
- one of the edge detectors with the best performance
- proceeds in three steps and requires three parameters:
  - Gaussian standard deviation σ
  - two thresholds T_1, T_2

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 34 35

How Does the Canny Edge Detector Work ?

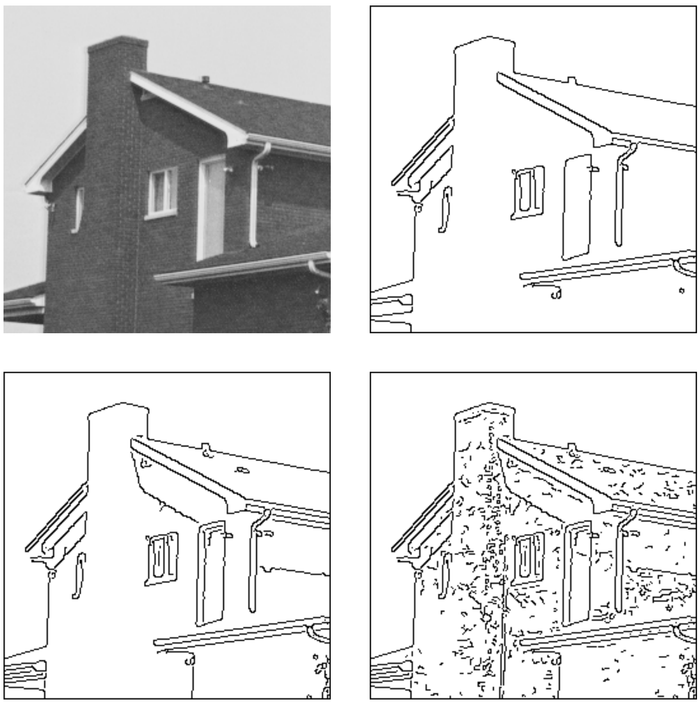
- *Gradient Approximation by Gaussian Derivatives:*
  - For the Gaussian-smoothed image u, compute the magnitude $|\boldsymbol{\nabla} u|$ and the orientation angle $\phi = \arg(\boldsymbol{\nabla} u)$ of $\boldsymbol{\nabla} u$ (cf. also Lecture 4, polar coordinates).
  - Identify edge candidates as locations where $|\boldsymbol{\nabla} u|$ 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, mark the central pixel for removal.
  - After passing through all candidates, remove marked pixels 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 (and their neighbours etc.) exceeding the lower threshold T_1.

Edge Detection with First Order Derivatives (6)



Influence of the Gaussian standard deviation σ on the Canny edge detector. The thresholds T_1 and T_2 are set to the 0.70 resp. 0.85 quantiles. Author: J. Weickert.

Edge Detection with First Order Derivatives (7)

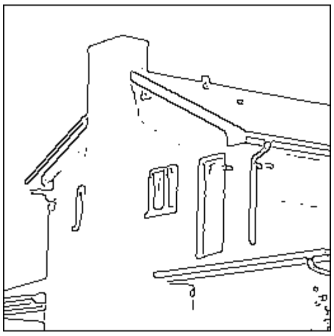


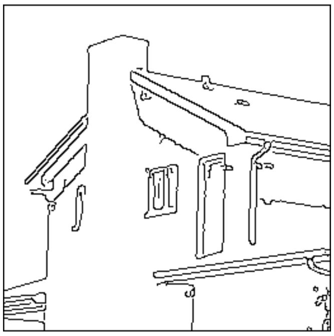
Influence of the upper threshold T_2 on the Canny edge detector ($\sigma = 1$, lower threshold T_1 at the 0.7 quantile). Author: J. Weickert.

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 26 27 28 29 30 31 32 33 34 35

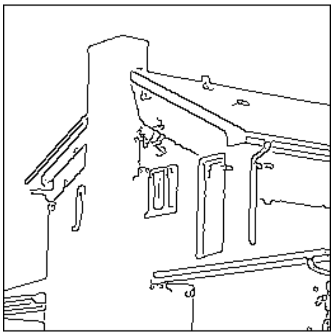
original, 256×256

0.85 quantile





0.7 quantile



0.5 quantile

Influence of the lower threshold T_1 on the Canny edge detector ($\sigma = 1$, upper threshold T_2 at the 0.85 quantile). Author: J. Weickert.

Edge Detection with Second Order Derivatives

Baseline Method

- Perform Gaussian smoothing of 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.

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 34 35

Edge Detection with Second Order Derivatives (2)

Advantages

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

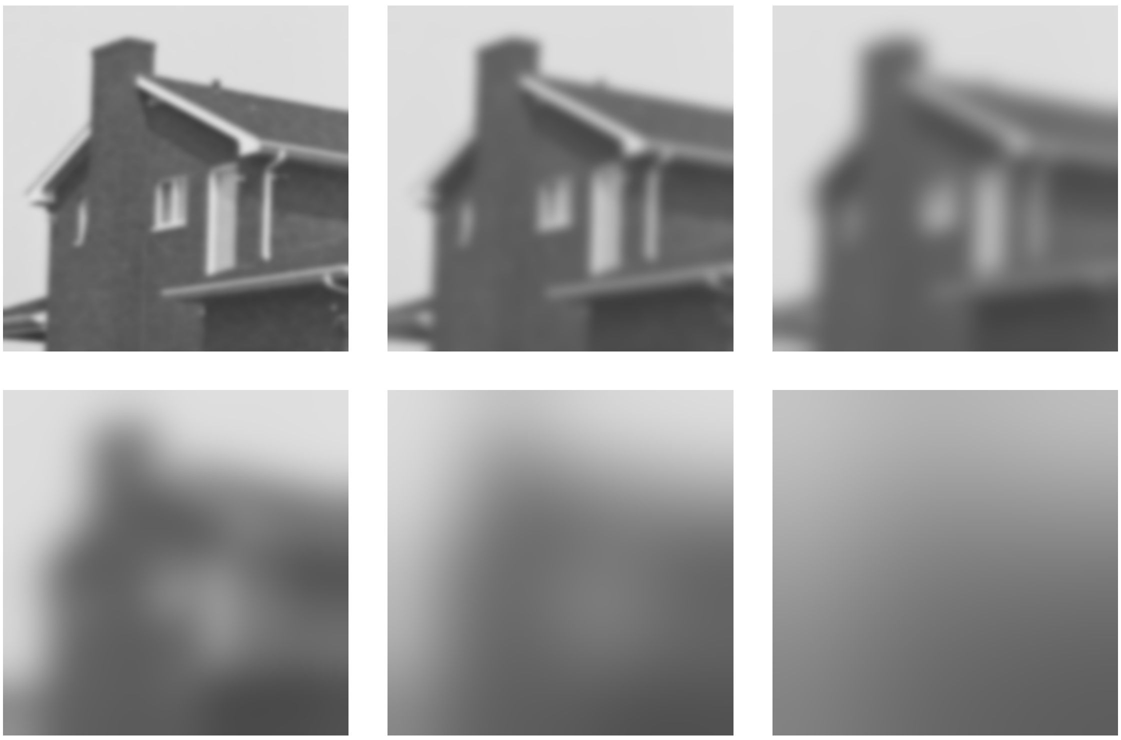
Disadvantages

- false alarms: detects not only maxima of the first derivative, but also minima
- Second order derivatives are more sensitive to noise than first order derivatives.
- Often this requires strong Gaussian smoothing, 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 34 35

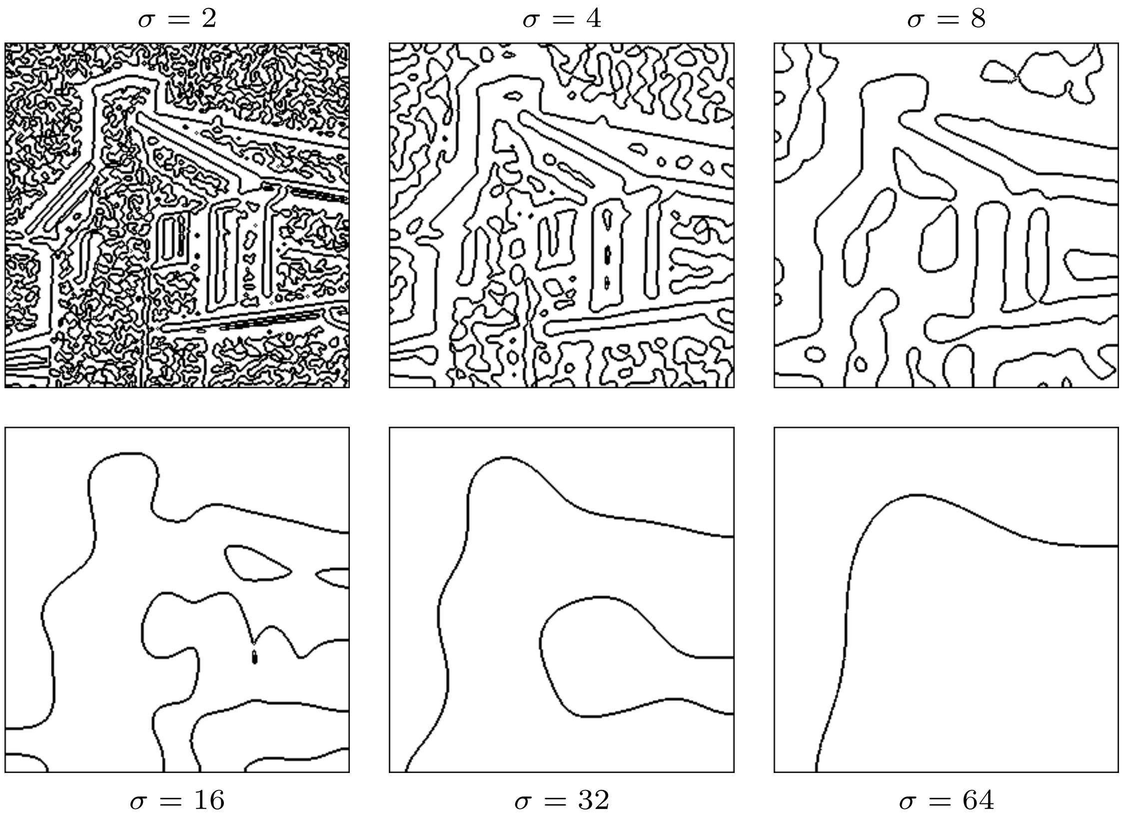
$\sigma = 2$ $\sigma = 4$ $\sigma = 8$



$\sigma = 16$ $\sigma = 32$ $\sigma = 64$

Gaussian smoothing of a test image (256×256 pixel). Author: J. Weickert.

Edge Detection with Second Order Derivatives (4)



Zero crossings of the Laplacian for the results from the previous page. 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 26 27 28 29 30 31 32 33 34 35

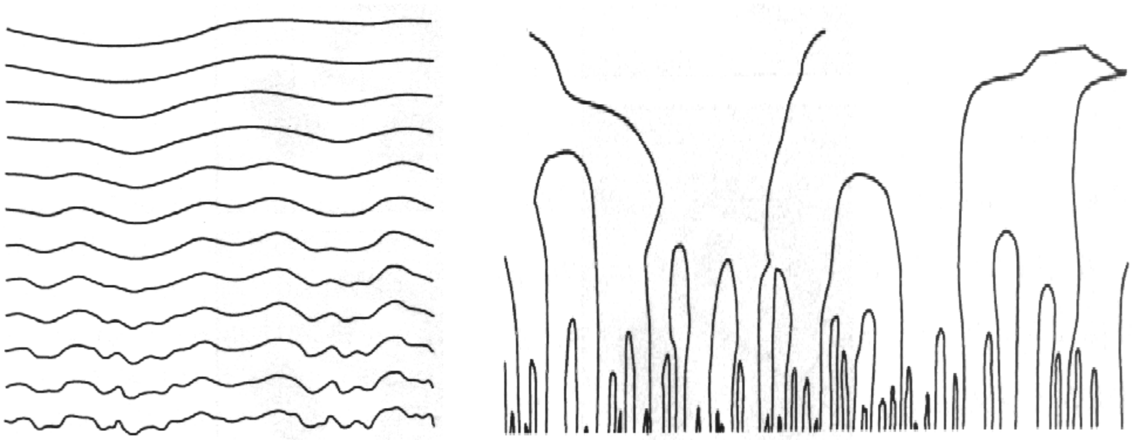
Edge Detection with Second Order Derivatives (5)

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: Embedding of an image in a continuum of more and more smoothed versions of it.
- oldest example for a scale-space: *Gaussian scale-space* $\{K_\sigma f \,|\, \sigma \geq 0\}$

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 34 35

Edge Detection with Second Order Derivatives (6)



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

Why are Corners Important ?

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.
- Similar to edge detection, corner detection can use either first or second order derivatives.
- However, to this end we first
  - have to remember some basics from linear algebra,
  - and use them to construct a good detector of the local image structure.

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 34 35

Basics From Linear Algebra

- **What are eigenvalues and eigenvectors?**
  Consider some $n \times n$ matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.
  Assume there exists 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 λ is called an *eigenvalue (Eigenwert)* of $\boldsymbol{A}$.
  The vector $\boldsymbol{v}$ is the associated *eigenvector (Eigenvektor)*.

- **How are eigenvectors computed?**
  Assume we know some eigenvalue λ of $\boldsymbol{A}$. Let $\boldsymbol{I} \in \mathbb{R}^{n \times n}$ denote the unit matrix.
  Obviously $\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$ can be rewritten as $(\boldsymbol{A} - \lambda \boldsymbol{I})\, \boldsymbol{v} = \boldsymbol{0}$.
  Thus, every nontrivial solution $\boldsymbol{v}$ of $(\boldsymbol{A} - \lambda \boldsymbol{I})\, \boldsymbol{v} = \boldsymbol{0}$ is an eigenvector.
  We see that eigenvectors are only defined up to a nonzero scaling factor.
  Often their length is normalised to 1 (as always in our class).

- **How are eigenvalues computed?**
  For nontrivial solutions of $(\boldsymbol{A} - \lambda \boldsymbol{I})\, \boldsymbol{v} = \boldsymbol{0}$, the matrix $\boldsymbol{A} - \lambda \boldsymbol{I}$ must be singular.
  This means that its determinant $\det(\boldsymbol{A} - \lambda \boldsymbol{I})$ must vanish.
  Thus, eigenvalues are zeroes of the *characteristic polynomial* $p(\lambda) := \det(\boldsymbol{A} - \lambda \boldsymbol{I})$.
  Usually they are computed numerically if $n \geq 3$. Many algorithms exist.

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 34 35

- **Are there particularly nice scenarios?**
  If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric, then all n eigenvalues of $\boldsymbol{A}$ are real.
  Moreover, one can find n eigenvectors that create an orthonormal basis of $\mathbb{R}^n$.

- **Here is a useful definition.**
  If a symmetric matrix has 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:

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

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 local structure direction around a corner changes strongly.
- We want to find the directions of largest / smallest greyvalue changes within a user-specified neighbourhood window.
- This requires to integrate directional information over a neighbourhood.

How can We Model This?

- We want to average directional information within some disk-shaped neighbourhood $B_\rho(x, y)$ of radius ρ around a point (x, y).
- We look for a direction given by a unit vector $\boldsymbol{n}$ that is "most parallel" or "most orthogonal" to $\boldsymbol{\nabla} u$ within $B_\rho(x, y)$.
- Hence, $\boldsymbol{n}$ should maximise or minimise the *average local contrast* defined as

$$\begin{aligned}
E(\boldsymbol{n}) \;&:=\; \int_{B_\rho(x,y)} (\boldsymbol{n}^\top \boldsymbol{\nabla} u)^2 \, dx' \, dy' \\
&= \int_{B_\rho(x,y)} \boldsymbol{n}^\top \boldsymbol{\nabla} u(x', y') \, \boldsymbol{\nabla} u^\top(x', y') \, \boldsymbol{n} \, dx' \, dy' \\
&= \boldsymbol{n}^\top \underbrace{\int_{B_\rho(x,y)} \boldsymbol{\nabla} u(x', y') \, \boldsymbol{\nabla} u^\top(x', y') \, dx' \, dy'}_{\text{symmetric } n \times n \text{ matrix}} \; \boldsymbol{n}
\end{aligned}$$

- This is a quadratic form with the matrix $\mathbf{M}_\rho(\boldsymbol{\nabla} u) := \displaystyle\int_{B_\rho(x,y)} \boldsymbol{\nabla} u \, \boldsymbol{\nabla} u^\top \, dx' \, dy'$.

The Structure Tensor (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 34 35

- optimal direction $\boldsymbol{n}$:
  normalised eigenvector to largest / smallest eigenvalue of $\mathbf{M}_\rho(\boldsymbol{\nabla} u)$
- Note that
  $$\mathbf{M}_\rho(\boldsymbol{\nabla} u) \;=\; \int\limits_{B_\rho(x,y)} \boldsymbol{\nabla} u\, \boldsymbol{\nabla} u^\top \, dx'\, dy'$$
  is just a componentwise convolution of the matrix-valued function $\boldsymbol{\nabla} u(x,y)\, \boldsymbol{\nabla} u^\top(x,y)$ with a pillbox-like kernel
  $$b_\rho(x,y) \;:=\; \begin{cases} 1 & \text{for } x^2+y^2 \le \rho^2, \\ 0 & \text{else.} \end{cases}$$
- For smoothness reasons, one should replace the pillbox convolution $\mathbf{M}_\rho(\boldsymbol{\nabla} u) = b_\rho * (\boldsymbol{\nabla} u\, \boldsymbol{\nabla} u^\top)$ by a convolution with a Gaussian K_ρ:
  $$\boldsymbol{J}_\rho(\boldsymbol{\nabla} u) \;:=\; K_\rho * (\boldsymbol{\nabla} u\, \boldsymbol{\nabla} u^\top) \;=\; \begin{pmatrix} K_\rho * (u_x^2) & K_\rho * (u_x u_y) \\ K_\rho * (u_x u_y) & K_\rho * (u_y^2) \end{pmatrix}.$$
- This matrix $\boldsymbol{J}_\rho$ is called *structure tensor (Strukturtensor)* (Förstner/Gülch 1987).
- The standard deviation ρ of the Gaussian K_ρ determines its locality.

The Structure Tensor (4)

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 34 35

What Does the Structure Tensor Tell Us?

- Obviously $\boldsymbol{J}_\rho$ is symmetric.
  Thus, it has orthonormal eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and real-valued eigenvalues λ_1, λ_2.
- One can even show that $\boldsymbol{J}_\rho$ is positive semidefinite. Let w.l.o.g. $\lambda_1 \ge \lambda_2 \ge 0$.
- Our considerations before imply that the quadratic form
  $$F(\boldsymbol{n}) \;:=\; \boldsymbol{n}^\top \boldsymbol{J}_\rho\, \boldsymbol{n}$$
  measures the Gaussian-weighted average local contrast in the direction $\boldsymbol{n}$.
  Hence, the eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ are the directions of largest/smallest local contrast.
- For these normalised eigenvectors we obtain
  $$F(\boldsymbol{v}_i) \;=\; \boldsymbol{v}_i^\top \boldsymbol{J}_\rho\, \boldsymbol{v}_i \;=\; \lambda_i\, \boldsymbol{v}_i^\top \boldsymbol{v}_i \;=\; \lambda_i \qquad (i \in \{1,2\}).$$
  Thus, the eigenvalues λ_1, λ_2 measure the average local contrast along $\boldsymbol{v}_1$, $\boldsymbol{v}_2$.
- The eigenvalues allow a useful analysis of the local image structure:

constant areas:	$\lambda_1 \;=\; \lambda_2 \;=\; 0$
straight edges:	$\lambda_1 \;\gg\; \lambda_2 \;=\; 0$
corners:	$\lambda_1 \;\ge\; \lambda_2 \;\gg\; 0$
measure of anisotropy:	$(\lambda_1 - \lambda_2)^2$

How Do We Compute these Eigenvalues?

- The ansatz $\boldsymbol{J}_\rho \boldsymbol{v} = \lambda \boldsymbol{v}$ yields $(\boldsymbol{J}_\rho - \lambda \boldsymbol{I})\,\boldsymbol{v} = \boldsymbol{0}$, where $\boldsymbol{I}$ denotes the unit matrix.
- Since an eigenvector is never $\boldsymbol{0}$, the matrix $\boldsymbol{J}_\rho - \lambda \boldsymbol{I}$ must be singular, i.e.

$$0 \stackrel{!}{=} \det(\boldsymbol{J}_\rho - \lambda \boldsymbol{I}) \;=\; (j_{1,1}-\lambda)\,(j_{2,2}-\lambda) - j_{1,2}^2\,.$$

Solving this quadratic equation in λ gives the eigenvalues

$$\lambda_1 \;=\; \frac{1}{2}\left(j_{1,1} + j_{2,2} + \sqrt{(j_{1,1}-j_{2,2})^2 + 4j_{1,2}^2}\right),$$
$$\lambda_2 \;=\; \frac{1}{2}\left(j_{1,1} + j_{2,2} - \sqrt{(j_{1,1}-j_{2,2})^2 + 4j_{1,2}^2}\right).$$

Corner Detection with the Structure Tensor

- based on the first derivative information in terms of the structure tensor
- Consider a Gaussian-smoothed version $u = K_\sigma * f$ of the original image f. Compute its structure tensor

$$\boldsymbol{J}_\rho(\boldsymbol{\nabla} u) \;=\; K_\rho * (\boldsymbol{\nabla} u\, \boldsymbol{\nabla} u^\top).$$

- Incorporating gradient information within a neighbourhood of scale ρ is achieved by convolution with a Gaussian K_ρ.
- The integration scale ρ should be larger than the noise scale σ.
- In corners, the structure tensor has two large eigenvalues: $\lambda_1 \geq \lambda_2 \gg 0$.
- Different strategies have been proposed in the literature in order to distiguish between corners (where $\lambda_1 \geq \lambda_2 \gg 0$) and edges (where $\lambda_1 \gg \lambda_2 \approx 0$).

The Most Popular Corner Detectors with the Structure Tensor

Let $\lambda_1 \geq \lambda_2$, let $\max$ denote a local maximum, and T some threshold.

- **Tomasi/Kanade (1991):**

$$\lambda_2 \overset{!}{>} T \qquad \text{and} \qquad \lambda_2 \overset{!}{=} \max .$$

  - looks simple, but requires to compute the smaller eigenvalue λ_2.

- **Rohr (1987):**

$$\det \boldsymbol{J}_\rho \;=\; j_{1,1}\, j_{2,2} \;-\; j_{1,2}^2 \;\overset{!}{>}\; T \qquad \text{and} \qquad \det \boldsymbol{J}_\rho \;\overset{!}{=}\; \max .$$

  - Since $\det \boldsymbol{J}_\rho = \lambda_1 \lambda_2$, both eigenvalues must be large.
  - does not require to compute the eigenvalues explicitly

- **Förstner (1986), Harris (1988)**:

$$\operatorname{tr} \boldsymbol{J}_\rho \;=\; j_{1,1} \;+\; j_{2,2} \;\overset{!}{>}\; T \qquad \text{and} \qquad \frac{\det \boldsymbol{J}_\rho}{\operatorname{tr} \boldsymbol{J}_\rho} \;\overset{!}{=}\; \max .$$

  - Since the trace satisfies $\operatorname{tr} \boldsymbol{J}_\rho = \lambda_1 + \lambda_2$, both eigenvalues are compared.
  - does not require to compute the eigenvalues explicitly

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 34 35

original, 256×256 — Tomasi/Kanade, $T = 18.2$

Rohr, $T = 100$ — Förstner, $T = 72$

Comparison of structure tensor based corner detectors by choosing the 34 most significant corners for every method ($\sigma = 2$, $\rho = 4$). 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 26 27 28 29 30 31 32 33 34 35

Corner Detection with Second Order Derivatives

Basic Idea Behind Different Methods:

- Consider a Gaussian-smoothed version $u = K_\sigma * f$ of the original image f.
- The *curvature*

$$\kappa \;=\; \frac{u_x^2 u_{yy} - 2 u_x u_y u_{xy} + u_y^2 u_{xx}}{(u_x^2 + u_y^2)^{3/2}}$$

  of the isolines (lines with equal grey values) should have a local maximum.
- However, the image gradient $|\boldsymbol{\nabla} u|$ should be sufficiently large as well (edge).
- Therefore, one detects corners as locations where $\kappa |\boldsymbol{\nabla} u|^\alpha$ has a local maximum and is larger than some significance threshold T.
- Depending on the nonnegative parameter α, different methods have been proposed.

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 34 35

Most Frequent Approaches:

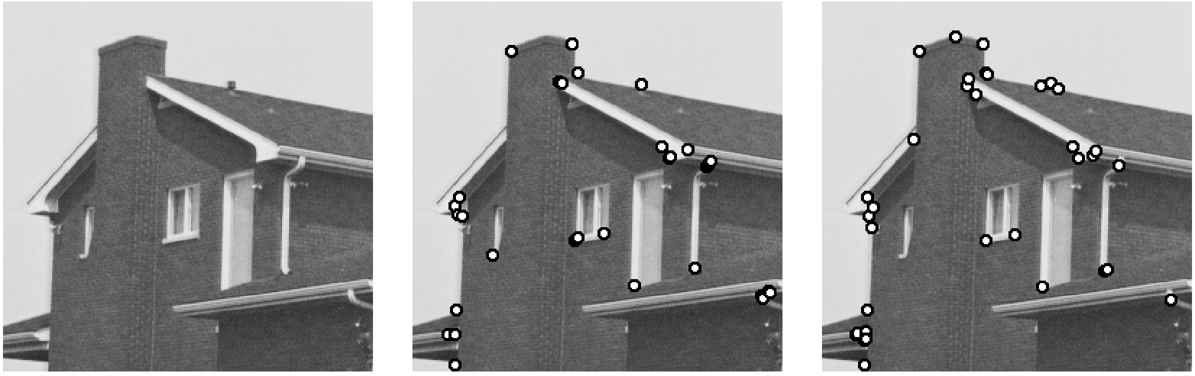
- **Kitchen/Rosenfeld (1982):**
  - chooses $\alpha = 1$
  - often gives fairly good results
- **Blom (1992):**
  - chooses $\alpha = 3$
  - invariant under affine transformations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ with $\det \boldsymbol{A} = 1$: result independent of the corner angle

How is the Mixed Derivative $\partial_{xy} u$ Discretised ?

For a quadratic grid with pixel size h, the simplest approximation is given by

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

Corner Detection with Second Order Derivatives (3)



original, 256×256 | Kitchen/Rosenfeld, $T = 2.17$ | Blom, $T = 200$

Comparison of curvature based corner detectors by choosing the 34 most significant corners for every method ($\sigma = 3$). Author: J. Weickert.

Corner Detection with Second Order Derivatives (4)

Can One Trace Back Corners in Scale-Space to Improve Their Localisation?

- Not always. One can show that Gaussian smoothing may even create corners.
- This seems to occur less frequently for $\alpha = 1$ than for $\alpha = 3$.

Do First or Second Order Corner Detectors Perform Better?

- Often corner detectors based on first order derivatives (structure tensor) perform slightly better.

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.

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 34 35

Summary

- Gaussian-smoothed derivatives can be used for detecting edges and corners.
- Important edge detector using first order derivatives: gradient magnitude
- The Canny filter uses nonmaxima suppression and hysteresis thresholding as postprocessing steps. It is one of the best methods for edge detection.
- Important edge detector using second-order derivatives:
zero-crossings of the Laplacian (Marr–Hildreth operator)
- The structure tensor allows a robust description of local image structure:
  - Its eigenvectors specify the local structure directions.
  - The eigenvalues give average contrast in these directions.
- Corner detection with first derivatives uses the eigenvalues of the structure tensor.
- Corner detection with second order derivatives combines curvature of isolines with gradient magnitude.
- Corner detection is more difficult and less robust than edge detection, but important in computer vision applications.

References

- O. Faugeras: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
(Chapter 4 gives 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)
- H. Anton: *Lineare Algebra*. Spektrum, Heidelberg, 1995.
(thick, but very well readable; also available in English)
- 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)

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 34 35

References (2)

- K. Rohr: Localization properties of direct corner detectors. *Journal of Mathematical Imaging and Vision*, Vol. 4, 139–150, 1994.
  (compares corner detectors based on first and second order derivatives)
- E. Trucco, A. Verri: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Englewood Cliffs, 1998.
  (computer vision book dealing also with corner detection)
- J. Sporring, M. Nielsen, O. F. Olsen, J. Weickert: Smoothing images creates corners. *Image and Vision Computing*, Vol. 18, No. 3, pp. 261–266, Febr. 2000.
  (`http://www.mia.uni-saarland.de/weickert/Papers/diku9801.ps.gz`)
  (proves that corner tracking in scale-space is problematic)
- D. G. Lowe: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.
  (`http://www.cs.ubc.ca/∼lowe/papers/ijcv04.pdf`)
  (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.
  (`http://www.vision.ee.ethz.ch/∼surf/papers.html`)
  (faster alternative to SIFT)