Lecture 19:
Feature Detection II:
Edges in Multichannel Images and Corners

Contents

Motivation

- In the last lecture we have learned how to use first and second order derivatives for detecting edges in greyscale images.
- Can this be generalised to
  - edge detection in colour images ?
  - corner detection ?
- To this end we have to
  - remember some basic results from linear algebra,
  - generalise the gradient to a more powerful descriptor of local image structure: 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

Basics From Linear Algebra

- Let $A \in \mathbb{R}^{n \times n}$. If there exists a number λ and a vector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v} \neq \mathbf{0}$ with $A\mathbf{v} = \lambda \mathbf{v}$, then λ is called an *eigenvalue (Eigenwert)* of A with associated *eigenvector (Eigenvektor)* $\mathbf{v}$.
- If some eigenvalue λ is known, then every nontrivial solution $\mathbf{v}$ of $(A - \lambda I)\mathbf{v} = \mathbf{0}$ is an eigenvector. Eigenvectors are only defined up to a scaling factor. Often their length is normalised to 1.
- Eigenvalues can be computed as zeroes of the *characteristic polynomial* $p(\lambda) := \det(A - \lambda I)$ with unit matrix I.
  For this purpose, many numerical algorithms exist.
- If $A \in \mathbb{R}^{n \times n}$ is symmetric, then all n eigenvalues of A are real, and one can find n eigenvectors that create an orthonormal basis of $\mathbb{R}^n$.
- If a symmetric matrix has only positive (resp. nonnegative) eigenvalues, then it is called *positive definite* (resp. *positive semidefinite*).
- Let $A \in \mathbb{R}^{n \times n}$ be symmetric. Then $Q(\mathbf{x}) := \mathbf{x}^\top A \mathbf{x}$ is a *quadratic form*. Among all vectors $\mathbf{x} \in \mathbb{R}^n$ with a specified magnitude, $Q(\mathbf{x})$ is maximised (minimised) by the eigenvector with the largest (smallest) eigenvalue.

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

Edge Detection in Multichannel Images

How Can Canny's Method be Generalised to a Multichannel Image ?

1. Presmoothing $\mathbf{f} = (f_1, ..., f_m)^\top$ by componentwise convolution with a Gaussian K_σ yields $\mathbf{u} = (u_1, ..., u_m)^\top$.
2. Define edge magnitude as length of the vector $(|\nabla u_1|, ..., |\nabla u_m|)^\top$:

$$\sqrt{|\nabla u_1|^2 + ... + |\nabla u_m|^2}$$

3. Nonmaxima suppression requires the definition of a gradient direction of a multichannel image (see below).
4. Hysteresis thresholding (double threshold) as in the scalar case.

Edge Detection in Multichannel Images (2)

Edge detection in colour images. **(a) Left:** Original colour image, 400×356 pixels (from `http://focus.msn.de/`). **(b) Right:** Edge magnitude after Gaussian convolution with $\sigma = 0.5$. For better visibility a gamma correction with $\gamma = 3$ has been performed. Author: J. Weickert (2005).

Edge Detection in Multichannel Images (3)

How Can One Define a Gradient Direction of a Multichannel Image ?

- Summing up the gradients of all channels is not a good idea.

  Example:
  Two-channel image with $\nabla u_1 =: \mathbf{w}$ and $\nabla u_2 = -\mathbf{w}$.
  Adding the gradients gives $\mathbf{0}$, while there in an edge in both channels.

- Therefore, the gradient *direction* must be considered, not its *orientation*.

Edge Detection in Multichannel Images (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

Di Zenzo's Method (1986)

- Gradient direction of a multichannel image $\mathbf{u} = (u_1, ..., u_m)^\top$ is defined as the unit vector $\mathbf{n}$ that is "most parallel" to ∇u_1,...,∇u_m: It maximises

$$E(\mathbf{n}) \;:=\; \sum_{i=1}^{m} (\mathbf{n}^\top \nabla u_i)^2.$$

- Only the direction of u_i matters, not the orientation.
- Since $\mathbf{n}^\top \nabla u_i = \nabla u_i^\top \mathbf{n}$, one maximises

$$E(\mathbf{n}) \;=\; \sum_{i=1}^{m} \mathbf{n}^\top \nabla u_i \nabla u_i^\top \mathbf{n} \;=\; \mathbf{n}^\top \Big(\sum_{i=1}^{m} \nabla u_i \nabla u_i^\top \Big) \, \mathbf{n}.$$

- This is a quadratic form which the symmetric matrix

$$J \;:=\; \sum_{i=1}^{m} \nabla u_i \nabla u_i^\top \;=\; \begin{pmatrix} \sum_i u_{ix}^2 & \sum_i u_{ix} u_{iy} \\ \sum_i u_{ix} u_{iy} & \sum_i u_{iy}^2 \end{pmatrix}.$$

It is maximised if $\mathbf{n}$ is eigenvector to the largest eigenvalue of J.

Edge Detection in Multichannel Images (5)

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 Does One Compute this Eigenvector ?

- The ansatz $J\mathbf{v} = \lambda \mathbf{v}$ yields $(J - \lambda I)\,\mathbf{v} = \mathbf{0}$, where I denotes the unit matrix.

- Since an eigenvector is never $\mathbf{0}$, the matrix $(J - \lambda I)$ must be singular, i.e.

$$0 \;\overset{!}{=}\; \det(J - \lambda I) \;=\; (j_{11} - \lambda)(j_{22} - \lambda) - j_{12}^2.$$

Solving this quadratic equation in λ gives the eigenvalues

$$\lambda_1 \;=\; \frac{1}{2}\left(j_{11} + j_{22} + \sqrt{(j_{11} - j_{22})^2 + 4 j_{12}^2} \right),$$

$$\lambda_2 \;=\; \frac{1}{2}\left(j_{11} + j_{22} - \sqrt{(j_{11} - j_{22})^2 + 4 j_{12}^2} \right).$$

- We only need the larger eigenvalue λ_1.

Edge Detection in Multichannel Images (6)

- Solving $(J - \lambda_1 I)\,\mathbf{v}_1 = \mathbf{0}$ for $\mathbf{v}_1$ gives the desired eigenvector $\mathbf{v}_1$.
  This is the "gradient direction" of our multichannel image.

- After some calculations one obtains the following algorithm:

if $j_{11} = j_{22}$ and $j_{12} = 0$:

  /* isotropic situation with two equal eigenvalues */

  $\mathbf{v}_1$ can be chosen arbitrarily (but $\neq \mathbf{0}$)

else if $j_{11} > j_{22}$:

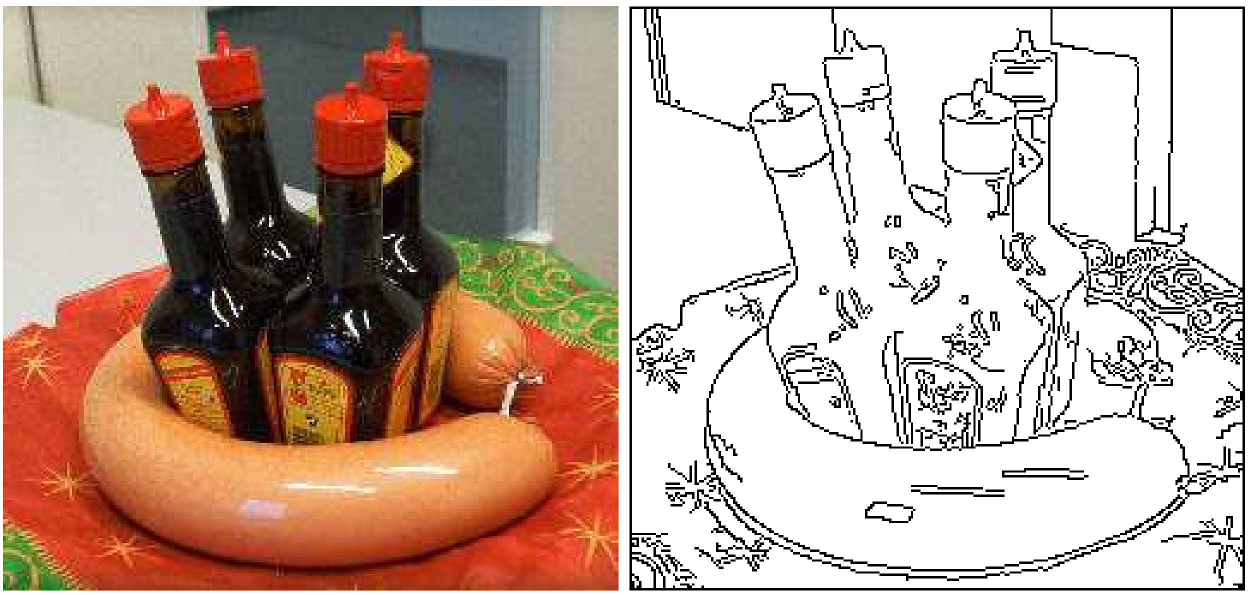
$$\mathbf{v}_1 \;\|\; \begin{pmatrix} j_{11} - j_{22} + \sqrt{(j_{11}-j_{22})^2 + 4j_{12}^2} \\ 2j_{12} \end{pmatrix}$$

else:

$$\mathbf{v}_1 \;\|\; \begin{pmatrix} 2j_{12} \\ j_{22} - j_{11} + \sqrt{(j_{11}-j_{22})^2 + 4j_{12}^2} \end{pmatrix}$$

Edge Detection in Multichannel Images (7)

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



Canny edge detection in colour images. **(a) Left:** Original colour image, 259×246 pixels, depicting a regional Christmas tradition ("saarländischer Adventskranz"). From `http://www.seelenfarben.de/`. **(b) Right:** After applying a Canny edge detector where the nonmaxima suppression takes into account the edge direction in the sense of Di Zenzo. Author: J. Weickert (2005).

The Structure Tensor

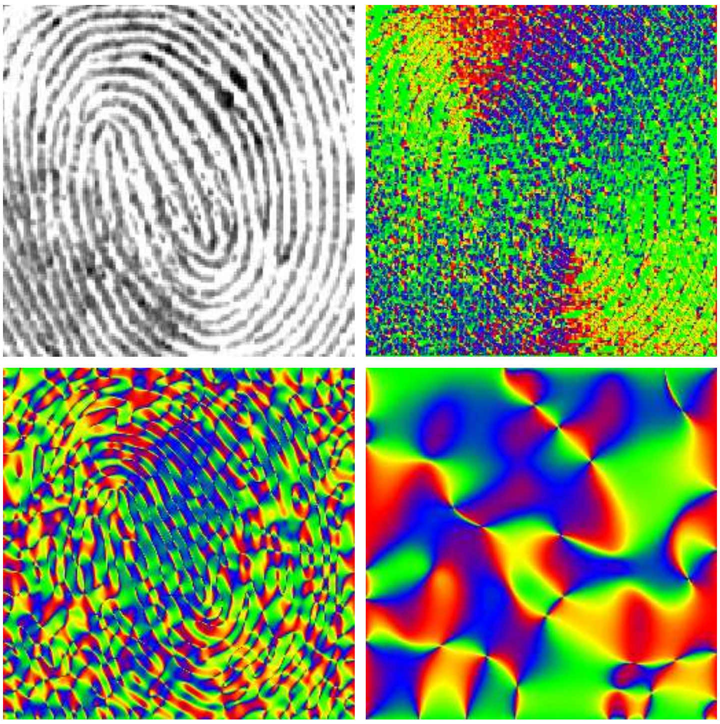
Motivation:

- So far we have analysed edges in colour images.
  This required to average directional information over channels.
- Now we would like to analyse corners in a greyscale image.
  The local structure direction around a corner changes strongly.
  This requires to integrate directional information over a neighbourhood.

Problem:

- Cancellation effects appear if gradients of opposite orientation are averaged (e.g. in flow-like structures such as fingerprints).
- Goal: Find a robust, gradient-like descriptor of the local image structure that does not suffer from this problem.

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



Cancellation effects when estimating the gradient direction. **(a) Top left:** Original image, $\Omega = (0, 256)^2$. **(b) Top right:** Colour-coded gradient direction. **(c) Bottom left:** Gradient direction after convolution with a Gaussian with standard deviation $\sigma = 4$. **(d) Bottom right:** $\sigma = 16$. Author: J. Weickert (2000).

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 (3)

Remedy:

- We want to average the structure direction within some neighbourhood $B_\rho(x,y)$ of radius ρ around a point (x,y).
- Similar strategy as for multichannel images, but with spatial averaging instead of averaging over the channels:
  desired direction given by unit vector $\mathbf{n}$ that maximises

$$
\begin{aligned}
E(\mathbf{n}) &:= \int_{B_\rho(x,y)} (\mathbf{n}^\top \nabla u)^2 \, dx'\, dy' \\
&= \int_{B_\rho(x,y)} \mathbf{n}^\top \nabla u(x',y')\, \nabla u^\top(x',y')\, \mathbf{n}\, dx'\, dy' \\
&= \mathbf{n}^\top \left(\int_{B_\rho(x,y)} \nabla u(x',y')\, \nabla u^\top(x',y')\, dx'\, dy' \right) \mathbf{n}
\end{aligned}
$$

- optimal $\mathbf{n}$: normalised eigenvector to largest eigenvalue of $\int_{B_\rho(x,y)} \nabla u \, \nabla u^\top \, dx'\, dy'$.

The Structure Tensor (4)

- Often one replaces this averaging of matrices by some weighted averaging with a Gaussian K_ρ:

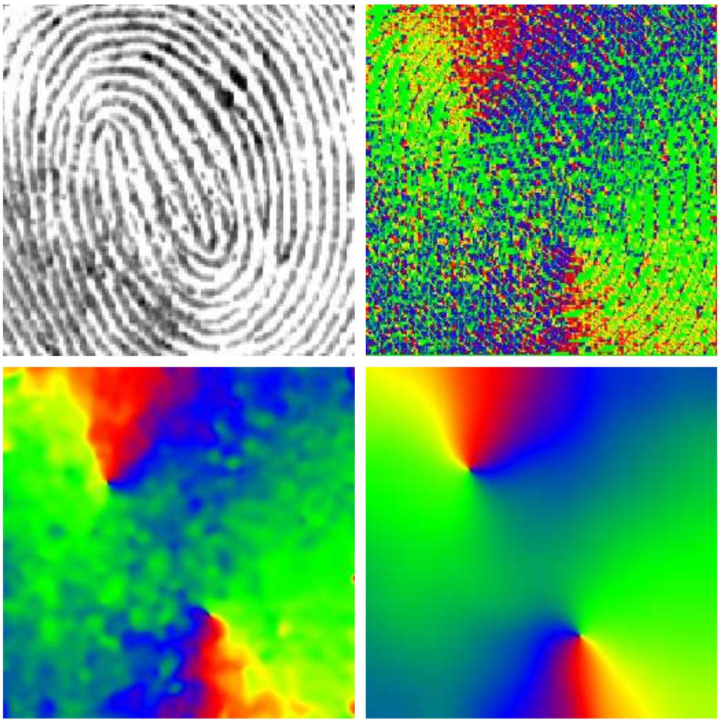
$$
J_\rho(\nabla u) \;:=\; K_\rho * (\nabla u \, \nabla u^\top) \;=\; \begin{pmatrix} K_\rho * (u_x^2) & K_\rho * (u_x u_y) \\ K_\rho * (u_x u_y) & K_\rho * (u_y^2) \end{pmatrix}
$$

- The matrix J_ρ is called *structure tensor (Strukturtensor)* (Förstner/Gülch 1987).
- J_ρ is symmetric and positive semidefinite.
  Its orthonormal eigenvectors v_1, v_2 specify the preferred structure directions within some integration scale ρ.
  The corresponding eigenvalues λ_1, λ_2 describe the average contrast along these directions.
- Let $\lambda_1 \geq \lambda_2$. 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:	$\lambda_1 \geq \lambda_2 \gg 0$
measure of anisotropy:	$(\lambda_1 - \lambda_2)^2$

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

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 does not suffer from cancellation effects. **(a) Top left:** Original image, $\Omega = (0, 256)^2$. **(b) Top right:** Direction of the eigenvector to the largest eigenvalue, with integration scale $\rho = 0$, i.e. without presmoothing. **(c) Bottom left:** $\rho = 4$. **(d) Bottom right:** $\rho = 16$. Author: J. Weickert (2000).

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 with the Structure Tensor

Why are Corners Important ?

- Corners are sparser features than edges.
  They are useful whenever point-like features are preferred over line-like features.
- They provide information about occlusions in a scene:
  The object with the corner is closer to the camera than its background.
- Corners are useful for 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.

Corner Detection with First Order Derivatives

- based on the structure tensor
- Consider a Gaussian-smoothed version $u = K_\sigma * f$ of the original image f, and compute the structure tensor

$$J_\rho(\nabla u) \;=\; K_\rho * (\nabla u \, \nabla u^\top).$$

- In corners the gradient direction changes rapidly within its neighbourhood.
- Incorporating gradient information within a neighbourhood is achieved by Gaussian convolution in the structure tensor.
- 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 $\lambda_1 \gg \lambda_2 \approx 0$).

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 Most Popular Structure Tensor Based Corner Detectors

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

- **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):**

$$\det J_\rho \;=\; j_{11} j_{22} - j_{12}^2 \overset{!}{>} T \qquad \text{and} \qquad \det J_\rho \overset{!}{=} \max .$$

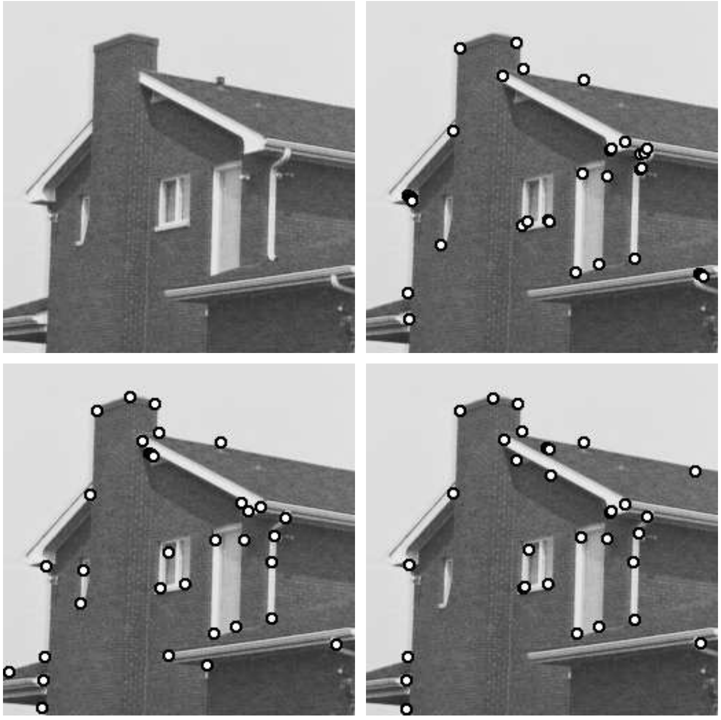
  - Since $\det J_\rho = \lambda_1 \lambda_2$, both eigenvalues must be large.
  - does not require to compute the eigenvalues

- **Förstner (1986), Harris (1988)**:

$$\operatorname{tr} J_\rho \;=\; j_{11} + j_{22} \overset{!}{>} T \qquad \text{and} \qquad \frac{\det J_\rho}{\operatorname{tr} J_\rho} \overset{!}{=} \max .$$

  - Since $\operatorname{tr} J_\rho = \lambda_1 + \lambda_2$, both eigenvalues are compared.
  - does not require to compute the 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



Comparison of structure tensor based corner detectors by choosing the 34 most significant corners for every method ($\sigma = 2$, $\rho = 4$). **(a) Top left:** Original image, 256×256 pixels. **(b) Top right:** Tomasi/Kanade ($T = 18.2$). **(c) Bottom left:** Rohr ($T = 100$) **(d) Bottom right:** Förstner ($T = 72$). Author: J. Weickert (2006).

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 with Second Order Derivatives

Basic Idea Behind Different Methods:

- Consider a Gaussian-smoothed version $u = K_\sigma * f$ of the original image f.
- 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 should have a local maximum.
- However, the image gradient $|\nabla u|$ should be sufficiently large as well (edge).
- Therefore one detects corners as locations where $\kappa |\nabla u|^\alpha$ has a local maximum and is larger than some significance threshold T.
- Depending on the nonnegative parameter α, different methods have been proposed.

Most Frequent Approaches:

- **Kitchen/Rosenfeld (1982):** $\alpha = 1.$
  often fairly good results

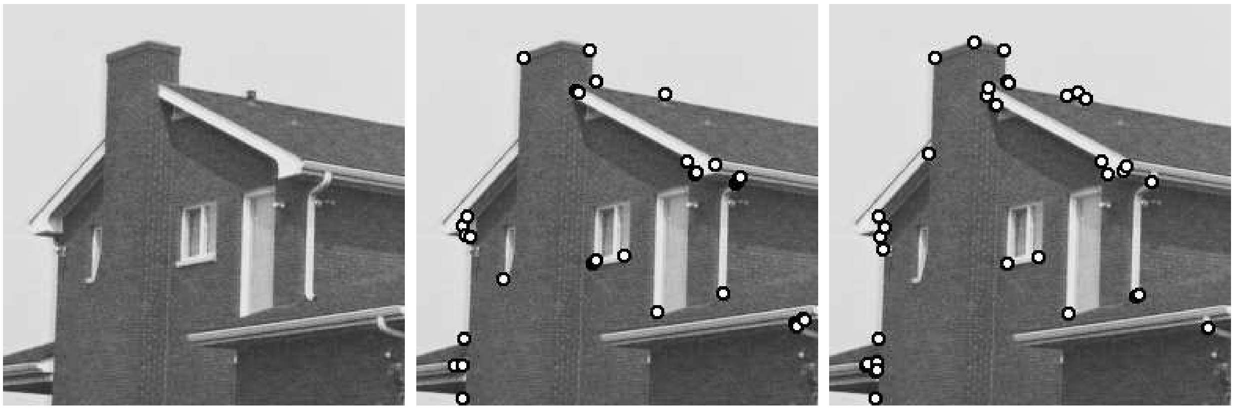
- **Blom (1992):** $\alpha = 3.$
  invariant under affine transformations $y = Ax$ with $\det A = 1$:
  result independent of the corner angle

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

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

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

MI 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



Comparison of curvature based corner detectors by choosing the 34 most significant corners for every method. **(a) Left:** Original image, 256×256 pixels. **(b) Middle:** Kitchen/Rosenfeld ($\sigma = 3$, $T = 2.17$). **(c) Right:** Blom ($\sigma = 3$, $T = 200$). Author: J. Weickert (2006).

Is it Possible to Trace Back Corners in Scale-Space for Improving Their Localisation?

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

Do First or Second Order Corner Detectors Perform Better?

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

Summary

- Basically one can generalise gradient-based edge detectors to multichannel images.
- Determining the edge direction creates an eigenvector problem.
- The structure tensor allows a robust detection of linear structures and corners.
- Eigenvectors specify the local structure directions.
  Eigenvalues give average contrast in these directions.
- Corner detection based on second order derivatives combines curvature of isolines with gradient magnitude.
- Corner detection is more difficult and less robust than edge detection, but important for many computer vision applications.

Literature

- H. Anton: *Lineare Algebra*. Spektrum, Heidelberg, 1995.
  (thick, but very well readable; also available in English)
- S. Di Zenzo: A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, Vol. 33, 116–125, 1986.
  (original paper by Di Zenzo)
- 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 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/publications.html`
  (proves that corner tracking in scale-space is problematic)