

Intelligente Sehsysteme

3 Hervorhebung von Kanten

Sobel-Operator

Konvolution und Korrelation

Laplace-Operator, LoG-Operator, DoG-Operator

Henry Hölzemann

Inhalt

- Gradienten der Bildfunktion
 - Sobel-Operator
- Konvolution und Korrelation
- 2. Ableitung der Bildfunktion
 - Laplace-Operator
 - Laplacian-of-Gaussian-Operator
 - Difference-of-Gaussian-Operator
 - Positionsbestimmung von Nulldurchgängen

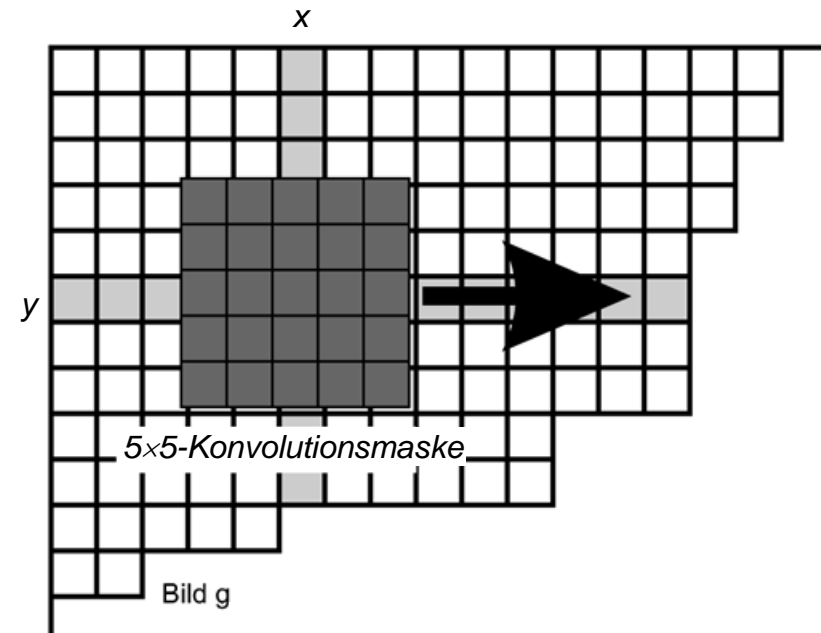
Lineare Filterung durch Konvolution (1)

Mit der Konvolution

$$(f * g)(x, y) = \sum_u \sum_v f(u, v) \cdot g(x - u, y - v) \text{ mit } u, v = -(m-1)/2, \dots, (m-1)/2$$

sind Pixel in Abhängigkeit von ihrer lokalen Pixelnachbarschaft veränderbar

- damit ist **Rauschunterdrückung** möglich (z.B. Mittelwert-, Gauß- und Binomialfilter)
- sowie die **Hervorhebung von Kanten-pixeln**

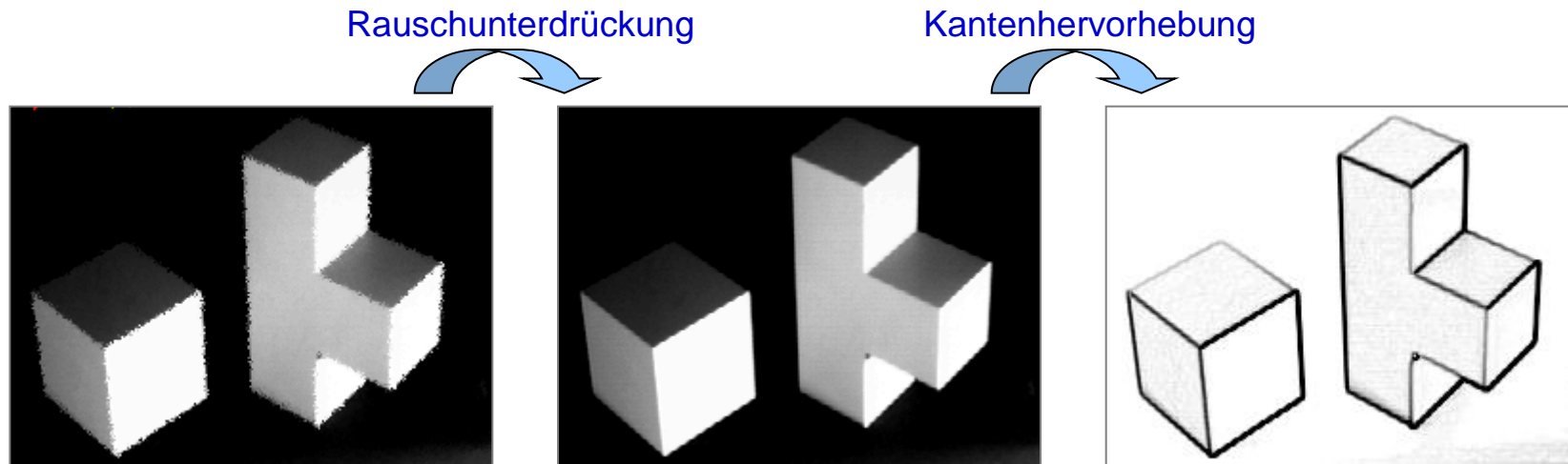


Nach Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

Lokale Nachbarschaft bei Konvolution

Beide Anwendungen der Konvolution basieren auf der Berücksichtigung der lokalen Pixelnachbarschaften:

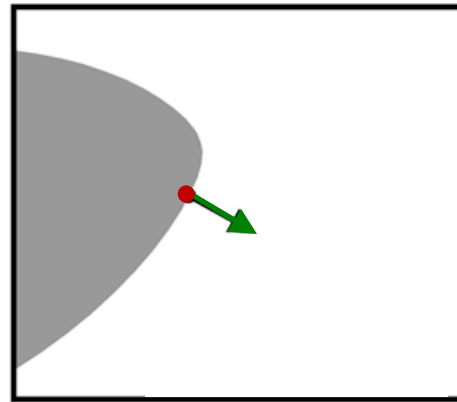
- bei **Rauschunterdrückung**, weil Rauschen nur als statistische Eigenschaft einer Menge von Pixel charakterisierbar ist
- bei **Kantenhervorhebung**, weil Kanten sich nur durch den Vergleich zwischen den Intensitäten benachbarter Pixel herausbilden



Zum Ergebnis einer Kantenhervorhebung

Die Erkennung von **Kantenpixel** kann prinzipiell resultieren in

- einen **Positionswert** des Kantenpixels (Position des Pixels)
- einen **Betragswert** des Kantenpixels (Länge des Vektors)
- einen **Orientierungswert** des Kantenpixels (Richtung des Vektors)

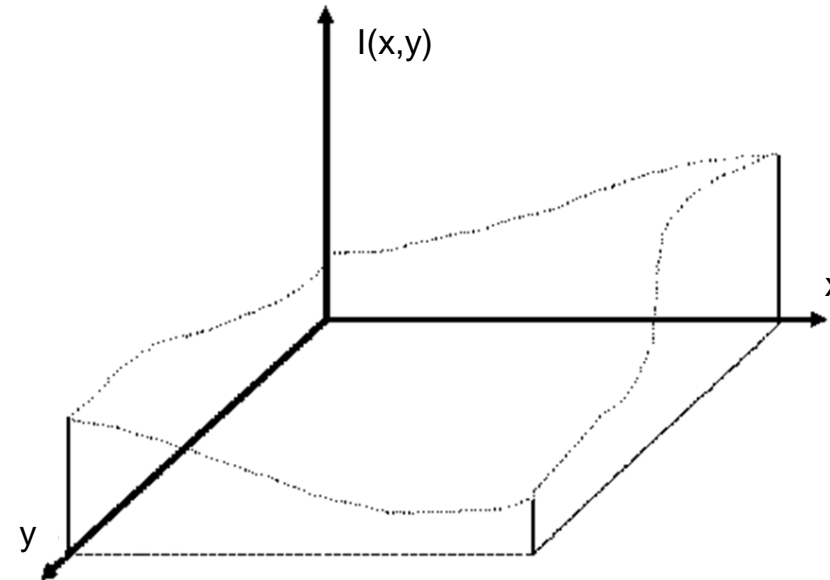


Nach Klaus Tönnies: Grundlagen der
Bildverarbeitung, Pearson Studium, 2005.

Kantenmodell bei kontinuierlicher Bildfunktion (1)

Zurück zum Modell eines Grauwertbildes mit reellwertiger Intensitätsfunktion $I(x,y)$ in reellwertigen Ortskoordinaten x und y :

$$I: \mathbb{R}_{[x_{\min}, x_{\max}]} \times \mathbb{R}_{[y_{\min}, y_{\max}]} \rightarrow \mathbb{R}_{[I_{\min}, I_{\max}]}, \\ (x,y) \rightarrow I(x,y).$$



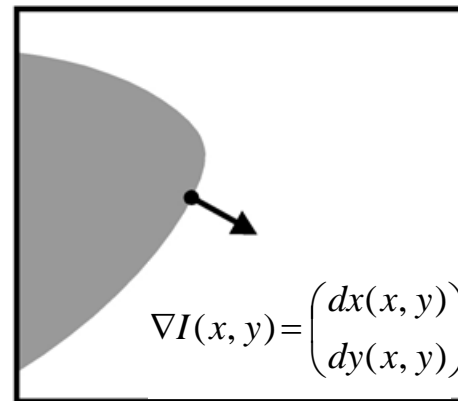
Bildquelle: Peter Haberäcker: Praxis der Digitalen Bildverarbeitung und Mustererkennung. Hanser, 1995.

- Die **Stärke einer Kante** nehmen wir als Intensitätsunterschied zwischen benachbarten Bildpunkten diesseits und jenseits der Kante wahr
- ~ bei einer kontinuierlichen Intensitätsfunktion entspricht eine Kante daher Stellen mit hohen Beträgen in der 1. Ableitung der Intensitätsfunktion

Kantenmodell bei kontinuierlicher Bildfunktion (2)

Der Gradient ist dann ein Vektor, dessen

- Länge die Kantenstärke repräsentiert und
- Richtung orthogonal zur Kantenrichtung verläuft



Nach Klaus Tönnies: Grundlagen der
Bildverarbeitung, Pearson Studium, 2005.

Kantenmodell bei diskreter Bildfunktion (1)

Bei diskreter Bildfunktion: **Approximation** der 1. Ableitung durch Ersetzen des Differentialquotienten für eine Pixelposition $p = (x,y)$ durch einen Differenzenquotienten:

$$\lim_{h \rightarrow 0} \frac{I(p+h) - I(p)}{h} \approx \frac{I(p+h) - I(p)}{h} \quad \text{für kleine } h$$

Kantenmodell bei diskreter Bildfunktion (2)

Implementierung der Approximation der 1. Ableitung der 2-dim. diskreten Bildfunktion durch Konvolution mit **Differenzen in x- und y-Richtung**:

$$\frac{\partial I(x, y)}{\partial x} \approx (-1 \quad 0 \quad 1) * I(x, y),$$

$$\frac{\partial I(x, y)}{\partial y} \approx \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} * I(x, y).$$

Die Konvolutionskerne bilden die Differenz in beiden Richtungen durch Subtraktion des Intensitätswertes vor dem aktuellen Pixel von dem Intensitätswert des nachfolgenden Pixels

Kantenmodell bei diskreter Bildfunktion (3)

Um a priori einer hohen Sensitivität gegenüber lokalem Rauschen vorzubeugen, können beide Konvolutionskerne jeweils mit einem Glättungsfilter kombiniert werden:

$$\text{Differenz in x-Richtung: } \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & +1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix},$$

$$\text{Differenz in y-Richtung: } \begin{pmatrix} +1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Der Glättungsfilter ist jeweils der unnormierte 1-dim. Binomialfilter B^1 – jeweils orthogonal orientiert zum entspr. Differenzfilter

Filter des Sobel-Operators

- Diese Kombination von zwei Faltungen zur Gradientenapproximation mit kombinierter Glättung durch das B¹-Filter heißt **Sobel-Operator**
- Jede Konvolution besteht für sich wieder aus zwei separablen Operationen:
 - Glättung orthogonal zur Ableitungsrichtung
 - Differenzbildung in Ableitungsrichtung

$$\textit{Horizontales Sobel - Filter } S_x : \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & +1 \end{pmatrix}$$

$$\textit{Vertikales Sobel - Filter } S_y : \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} = \begin{pmatrix} +1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

Resultate des Sobel-Operators

Mit der Anwendung der beiden Sobel-Filter durch Konvolution ist die Operation noch nicht abgeschlossen. Das Ergebnis sind lediglich die Approximationen der beiden gerichteten Ableitungen.

Vgl. Folie 5

Der **Gradientenbetrag** S ergibt dann nach:

$$S \approx \sqrt{S_x(x, y)^2 + S_y(x, y)^2}.$$

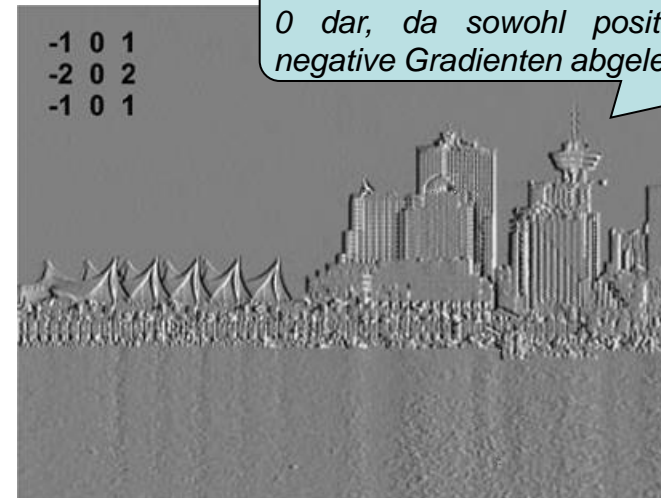
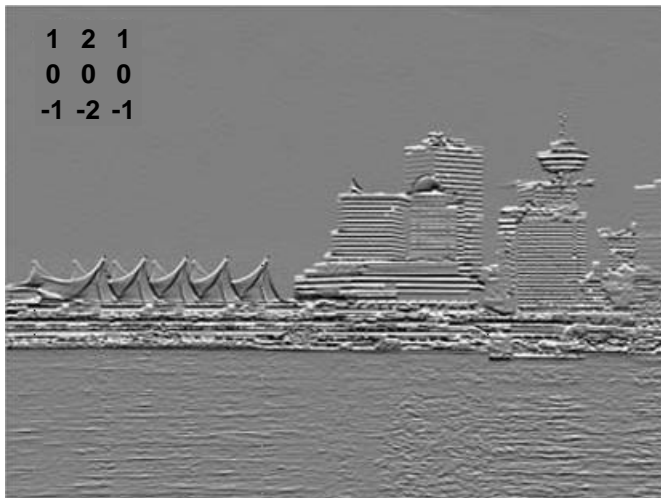
Die **Gradientenrichtung** Θ wird berechnet nach: *

$$\Theta \approx \begin{cases} \arctan(S_y(x, y) / S_x(x, y)) & \text{für } S_x(x, y) \neq 0, \\ 90^\circ & \text{für } S_x(x, y) = 0, S_y(x, y) \neq 0. \end{cases}$$

* Für $S_x(x, y) = S_y(x, y) = 0$ ist Θ undefiniert

Anwendung des Sobel-Operators (1)

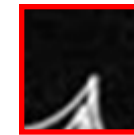
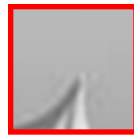
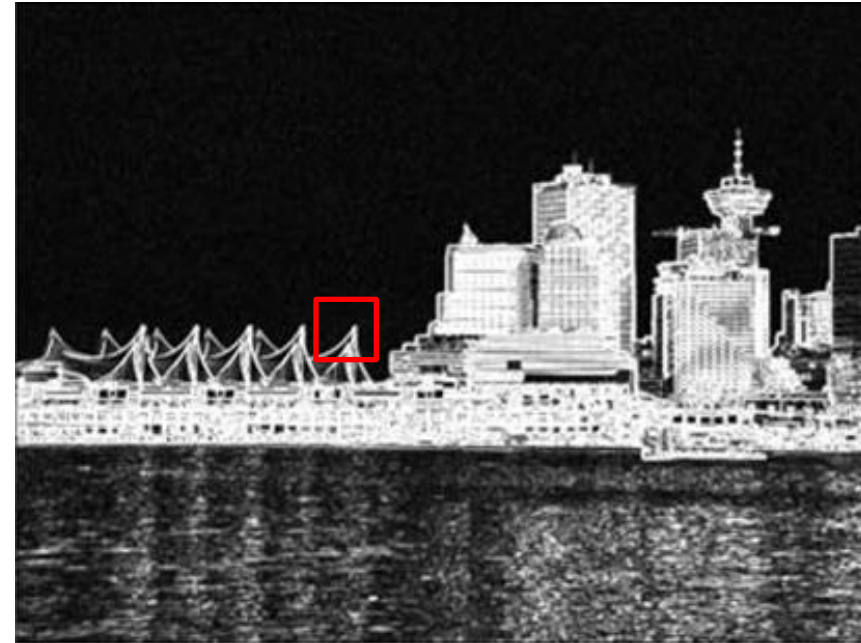
Beispiel: Das Originalbild (l.o.), die Gradientenbeträge (r.o.) sowie gerichteten Gradienten in vertikale (l.u.) und horizontale (r.u.) Richtung.



In der unteren Reihe stellt Grau den Wert 0 dar, da sowohl positive als auch negative Gradienten abgeleitet werden.

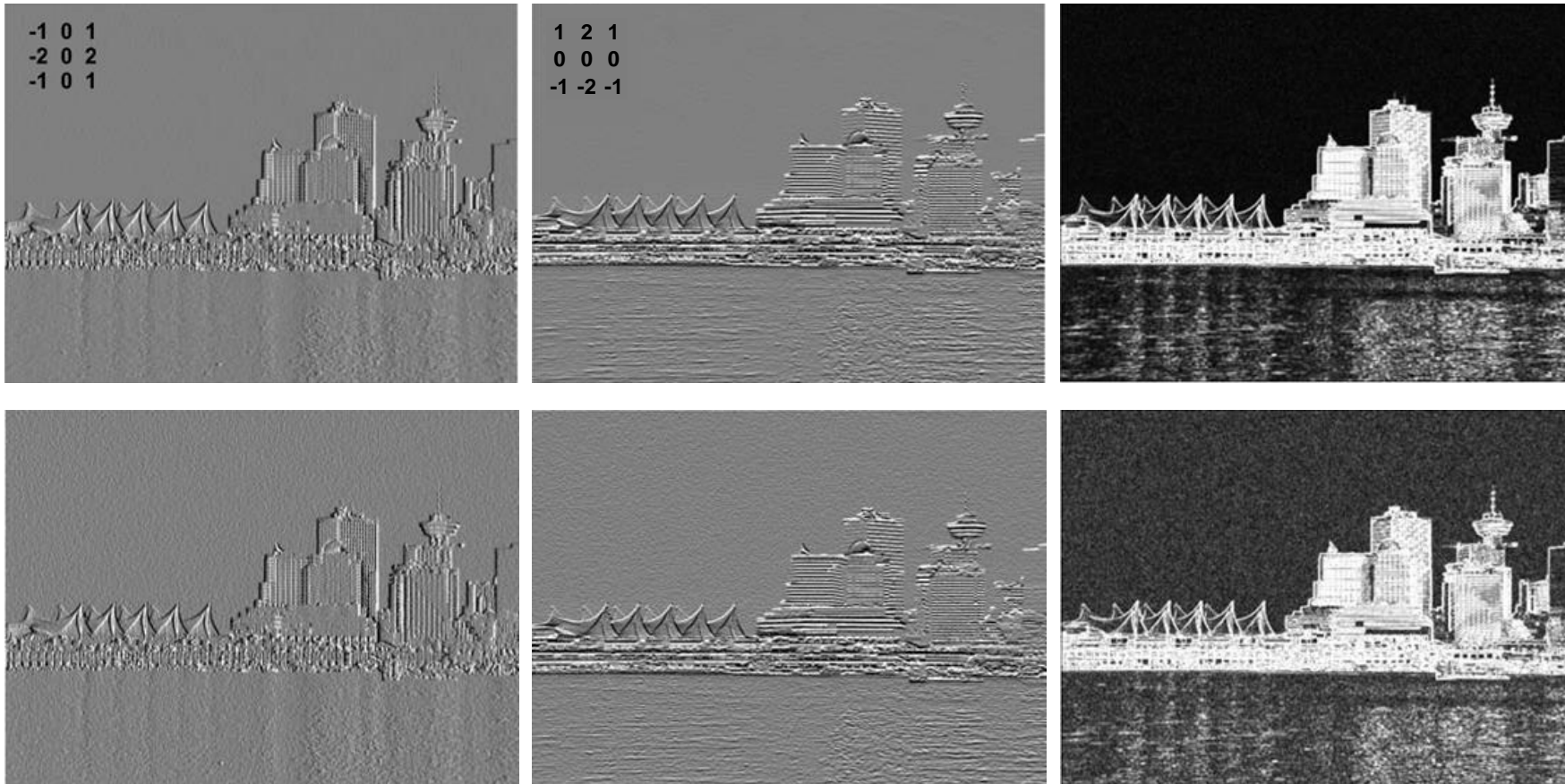
Anwendung des Sobel-Operators (2)

Beispiel: Gradientenbeträge skaliert auf [0,255]



Anwendung des Sobel-Operators (4)

Beide Komponenten des Sobel-Operators sowie der Betrag des Gradienten auf rauscharmer (oben) und verrauschter (unten) Bildversion:



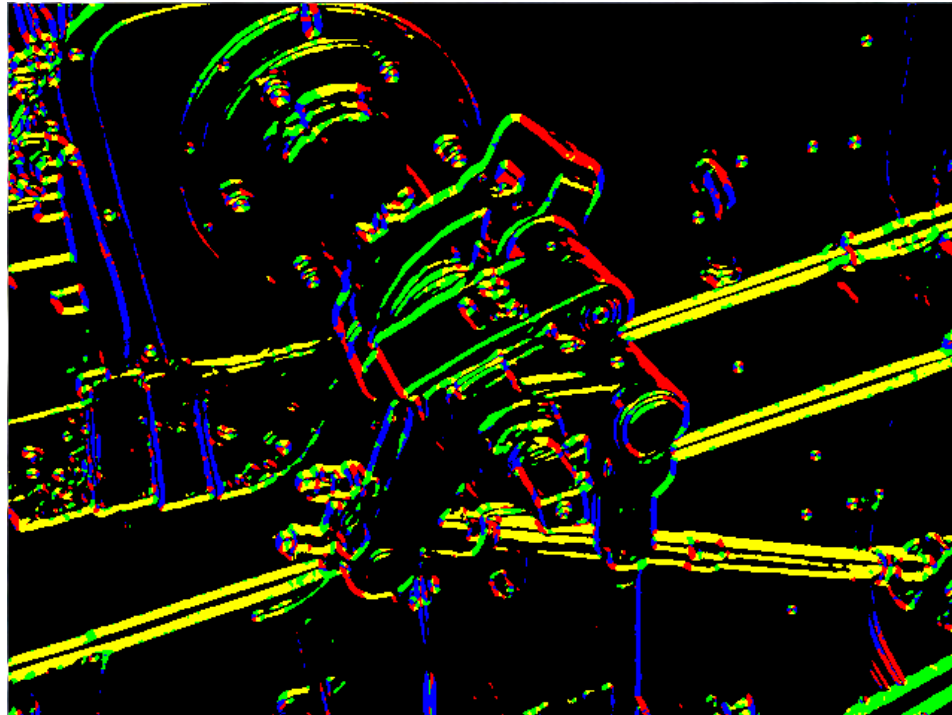
Anwendung des Sobel-Operators (5)

Für eine Visualisierung der **Gradientenrichtung** Θ bietet sich ein Binning der Winkelwerte an:

- Beim **Binning** wird der Wertebereich der Größe nach aufsteigend in Intervalle – sogenannte bins (engl. für Behälter) – eingeteilt
 - Bspl.: Farbkodierung der **Gradientenrichtung** Θ in:*
- blau = $0^\circ \pm 22,5^\circ$
 - rot = $45^\circ \pm 22,5^\circ$
 - gelb = $90^\circ \pm 22,5^\circ$
 - grün = $135^\circ \pm 22,5^\circ$

* Einige Details dazu (s. Übungsblatt):

- (1) Die Intervalle der Bins müssen natürlich disjunkt sein ($\sim <, \geq; \leq, >$)
- (2) Die Richtungen unterscheiden hier nicht Übergänge von hell nach dunkel und umgekehrt
- (3) Die Behandlung der negativen Werte von Arkustangens und der 90° -Werte sind hier nicht berücksichtigt



Bildquelle: http://en.wikipedia.org/wiki/Canny_edge_detector (24.06.2011)

Sobel-Operator und Konvolution (1)

- Die Konvolution wendet die Filtermasken generell punktgespiegelt auf die Bildmatrix an – dies ergibt sich aus den negativen Vorzeichen der Verschiebungsvektoren u und v in der Konvolutionsformel:

$$(f * g)(x, y) = \sum_u \sum_v f(u, v) \cdot g(x - u, y - v) \text{ mit } u, v = -(m-1)/2, \dots, (m-1)/2$$

- Bei punktsymmetrischen Glättungsfiltern ist dies ohne Auswirkung
- Bei orientierten Filtern wie den Sobel-Filtern sind Auswirkungen zu beachten

Sobel-Operator und Konvolution (2)

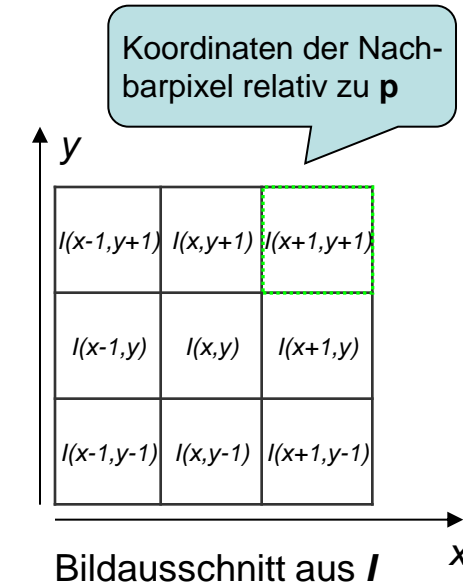
Konvolution mit den Sobel-Filtermasken auf ein Bildpixel $\mathbf{p} = (x,y)$:

-1	0	1
-2	0	2
-1	0	1

Horizontal. Sobel-Filter \mathbf{S}_x

1	2	1
0	0	0
-1	-2	-1

Vertikal. Sobel-Filter \mathbf{S}_y



$$\mathbf{S}_x * I = I(x-1, y+1) + 2 \cdot I(x-1, y) + I(x-1, y-1) - I(x+1, y+1) - 2 \cdot I(x+1, y) - I(x+1, y-1),$$

$$\mathbf{S}_y * I = I(x+1, y-1) + 2 \cdot I(x, y-1) + I(x-1, y-1) - I(x+1, y+1) - 2 \cdot I(x, y+1) - I(x-1, y+1).$$

Konvolution und Korrelation (1)

Die punktgespiegelten Anwendung der Konvolution wird oft als verwirrend betrachtet, um die Koeffizienten der Konvolutionsmaske mit den Nachbarpixeln zu assoziieren, auf die sie angewendet werden:

-1	0	1
-2	0	2
-1	0	1

Horizontal. Sobel-Filter \mathbf{S}_x

$I(x-1, y+1)$	$I(x, y+1)$	$I(x+1, y+1)$
$I(x-1, y)$	$I(x, y)$	$I(x+1, y)$
$I(x-1, y-1)$	$I(x, y-1)$	$I(x+1, y-1)$

Bildausschnitt aus I

Konvolution und Korrelation (2)

Aus diesem Grund wird die **Konvolution**

$$(f * g)(x, y) = \sum_u \sum_v f(u, v) \cdot g(x - u, y - v) \text{ mit } u, v = -(m-1)/2, \dots, (m-1)/2$$

häufig durch die **Korrelation** ersetzt bzw. implementiert:

$$(f \oplus g)(x, y) = \sum_u \sum_v f(u, v) \cdot g(x + u, y + v) \text{ mit } u, v = -(m-1)/2, \dots, (m-1)/2.$$

Die Korrelation entspricht eher dem Bild einer gleitenden Maske mit korrespondierenden Stellen von Koeffizienten der Filtermaske einerseits und zu gewichtenden Bildpixeln andererseits.

Konvolution und Korrelation (3)

Die Anwendung der beiden Sobel-Filter durch die Korrelation zeigt die anschauliche Korrespondenz zwischen den Koeffizienten der Filterkerne einerseits und den Nachbarpixeln des untersuchten Pixels andererseits:

$$(f \oplus g)(x,y) = \sum_u \sum_v f(u,v) \cdot g(x+u,y+v) \text{ mit } u,v = -(m-1)/2, \dots, (m-1)/2.$$

-1	0	1
-2	0	2
-1	0	1

Horizontal. Sobel-Filter S_x

1	2	1
0	0	0
-1	-2	-1

Vertikal. Sobel-Filter S_y

$I(x-1,y+1)$	$I(x,y+1)$	$I(x+1,y+1)$
$I(x-1,y)$	$I(x,y)$	$I(x+1,y)$
$I(x-1,y-1)$	$I(x,y-1)$	$I(x+1,y-1)$

Bildausschnitt aus I

$$S_x \oplus I = I(x+1,y-1) + 2 \cdot I(x+1,y) + I(x+1,y+1) - I(x-1,y-1) - 2 \cdot I(x-1,y) - I(x-1,y+1),$$

$$S_y \oplus I = I(x-1,y+1) + 2 \cdot I(x,y+1) + I(x+1,y+1) - I(x-1,y-1) - 2 \cdot I(x,y-1) - I(x+1,y-1).$$

Konvolution und Korrelation (4)

- Konvolution und Korrelation erzeugen bei orientierten Filtermasken unterschiedliche Ergebnisse.
- Daher ist anzugeben, mit welcher Operation eine Filtermaske angewendet wurde.
- Beispiel für horizontal. Sobel-Filter:

$$\mathbf{S}_x * I$$

$$\begin{aligned} &= (1+2+1)-(100+200+100) \\ &= 4 - 400 = -396 \end{aligned}$$

$$\mathbf{S}_x \oplus I$$

$$\begin{aligned} &= -(1+2+1)+(100+200+100) \\ &= -4 + 400 = 396 \end{aligned}$$

-1	0	1
-2	0	2
-1	0	1

Horizont. Sobel-Filter \mathbf{S}_x

1	20	100
1	20	100
1	20	100

Bildausschnitt aus I

Bemerkung: Betrachten Sie nochmals die Abbildungen zu den gerichteten Gradienten in den Folien 13 - 16. Wurde die Konvolution oder die Korrelation angewandt?

Beis

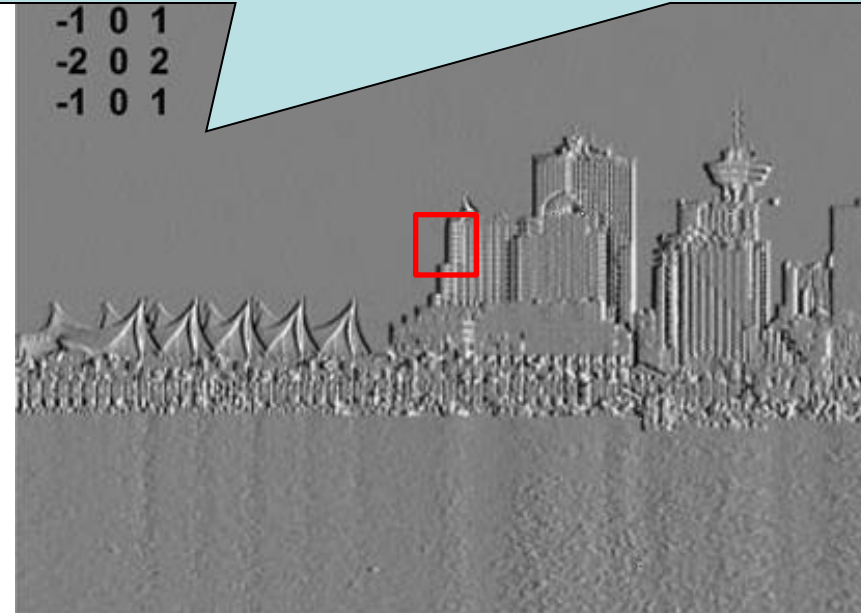
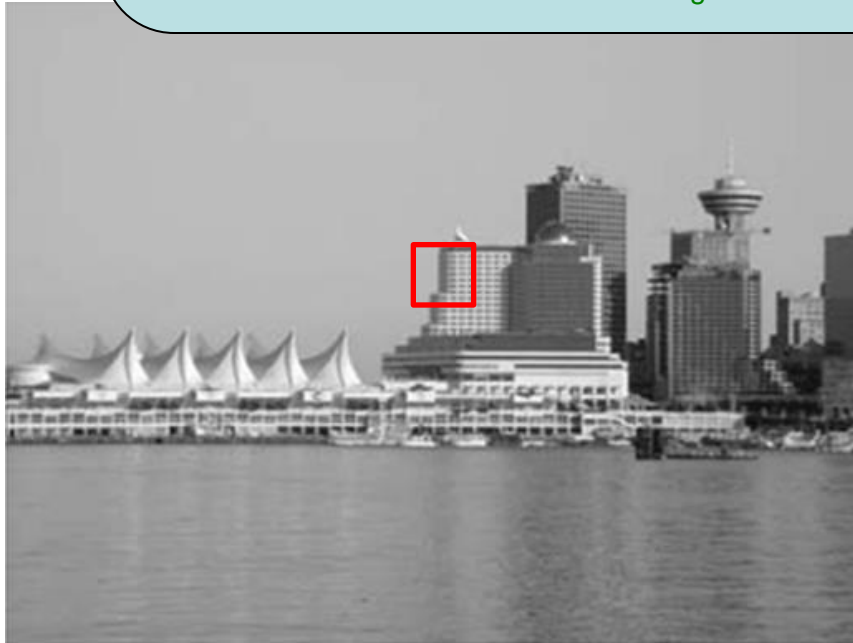
Übergang von hellem Himmel (links) zu dunkler Fassade (rechts):

1. Konvolution:

- dunkle Fassadenpixel (alle z.B. 50) werden negativ gewichtet: -200;
- helle Himmelpixel (alle z.B. 200) werden positiv gewichtet: 800;
- Gewichtete Summe: 600 \leadsto also positiver Gradient \leadsto helle Pixel im Gradientenbild
- Widerspruch zu dunklen Pixeln im Gradientenbild

2. Korrelation:

- dunkle Fassadenpixel (alle z.B. 50) werden positiv gewichtet: 200;
- helle Himmelpixel (alle z.B. 200) werden negativ gewichtet: -800;
- Gewichtete Summe: -600 \leadsto also negativer Gradient \leadsto dunkle Pixel im Gradientenbild
- Übereinstimmung mit dunklen Pixeln im Gradientenbild



\vee



Normierung

Das letzte Beispiel zeigte, dass die Ergebnisse des Sobel-Operators leicht die darstellbaren Intensitäten eines gegebenen Intensitätsspektrums (z.B. 0,...,255) unter- oder überschreiten können:

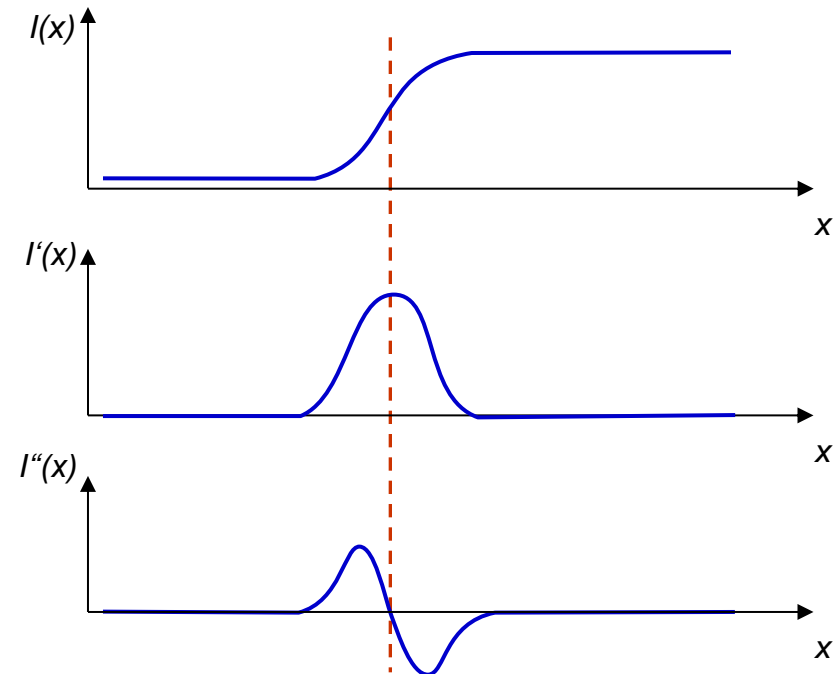
$$\begin{aligned}\mathbf{S}_x \oplus I &= 396, \mathbf{S}_y \oplus I = 0, \\ \mathbf{S} &= (S_x(x,y)^2 + S_y(x,y)^2)^{1/2} = 396.\end{aligned}$$

Allg. sind die Werte dann über *lineare Intensitätsspreizung* oder *Histogrammlinearisation* (s. 2. Vorl.) in das gegebene Intensitätsspektrum zu transformieren.

2. Ableitung der Bildfunktion

- Alle bisherigen Operatoren zur Kantenhervorhebung basieren auf der 1. Ableitung der Bildfunktion
- Auch die 2. Ableitung der Bildfunktion ist zur Kantenhervorhebung nutzbar
- Aus dem Kantenmodell für kontinuierliche Bildfunktionen ergibt sich:

- Kanten sind Orte mit maximalen Werten in der 1. Ableitung
- dort zeigt die 2. Ableitung Vorzeichenwechsel – sog. Nulldurchgänge



2. Ableitung der Bildfunktion

Analog zur 1. Ableitung besteht die 2. Ableitung der Bildfunktion mit 2-dim. Definitionsbereich aus einer Menge von partiellen Ableitungen, nämlich vier partiellen 2. Ableitungen:

$$\frac{\partial^2 I}{\partial x^2}(x, y) , \quad \frac{\partial^2 I}{\partial y^2}(x, y) , \quad \frac{\partial^2 I}{\partial x \partial y}(x, y) , \quad \frac{\partial^2 I}{\partial y \partial x}(x, y) .$$

Diskr. Approximation der 2. Ableitung

Auch die 2. Ableitung der Bildfunktion durch Differenzen approximierbar:

- $\frac{\partial^2 I}{\partial x^2}(x, y)$ ist durch zweimaliges Anwenden eines Gradientenoperators in x-Richtung approximierbar
- $\frac{\partial^2 I}{\partial x \partial y}(x, y)$ ist durch Anwenden eines Gradientenoperators in x-Richtung gefolgt von der eines Gradientenoperators in y-Richtung approximierbar

Laplace-Operator (1)

Laplace-Operator

- basiert auf der Summe der partiellen 2. Ableitungen
- wird mit ∇^2 bezeichnet
- zeigt zwei Versionen:
 - Summe über beide zweimaligen Ableitungen in x- und y-Richtung:

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

- Summe über alle vier part. Ableitungen:

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

Laplace-Operator (2)

- Die diskrete Approximation des Laplace-Operators basiert auf die bereits vorgestellten einfachen Differenzoperatoren zur partiellen Gradientenapproximation:

$$D_{x,3} = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}, \quad D_{y,3} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

- Um kompakte 3×3-Laplace-Operatoren zu erzeugen, werden aber noch kompaktere Differenzoperatoren verwendet:

$$D_{x,2} = \begin{pmatrix} -1 & 1 \end{pmatrix}, \quad D_{y,2} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Laplace-Operator (3)

- Approx. Der 2. Ableitungen entlang der x- und der y-Richtung :

$$\begin{aligned}\frac{\partial^2 I}{\partial x^2} &\approx \frac{\partial D_{x,2}}{\partial x} \\ &= \frac{\partial(I(x+1, y) - I(x, y))}{\partial x} \\ &= \frac{\partial I(x+1, y)}{\partial x} - \frac{\partial I(x, y)}{\partial x} \\ &\approx (I(x+2, y) - I(x+1, y)) - (I(x+1, y) - I(x, y)) \\ &= I(x+2, y) - 2 \cdot I(x+1, y) + I(x, y)\end{aligned}$$

- Zentrum der Approximation ist Pixel $[x+1, y]$. Substitution von x mit $x-1$ führt zur Approximation für die zweite partielle Ableitung im Pixel $[x, y]$:

$$\frac{\partial^2 I}{\partial x^2} \approx I(x+1, y) - 2 \cdot I(x, y) + I(x-1, y)$$

Laplace-Operator (4)

- Analog zur 2. Ableitung in x-Richtung

$$\frac{\partial^2 I}{\partial x^2} \approx I(x+1, y) - 2 \cdot I(x, y) + I(x-1, y)$$

ergibt sich die 2. Ableitung in y-Richtung:

$$\frac{\partial^2 I}{\partial y^2} \approx I(x, y+1) - 2 \cdot I(x, y) + I(x, y-1)$$

- Kombination beider part. Ableitungen zu einem einzigen Operator führt zum sog. L_4 -Filterkern zur Approx. des Laplace-Operators:

$$\begin{aligned} \nabla^2 I(x, y) &= \frac{\partial^2 I}{\partial x^2}(x, y) + \frac{\partial^2 I}{\partial y^2}(x, y) \\ &= I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4 \cdot I(x, y) \end{aligned}$$

Laplace-Operator (5)

Der L_4 -Laplace-Operator

$$L_4 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \xrightarrow{\text{häufige Variante}} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

basiert nur auf der Approximation der 2. Ableitungen in x- und y-Richtung nach:

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

Die Approximation der 2. Version

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

führt zum L_8 -Laplace-Operator :

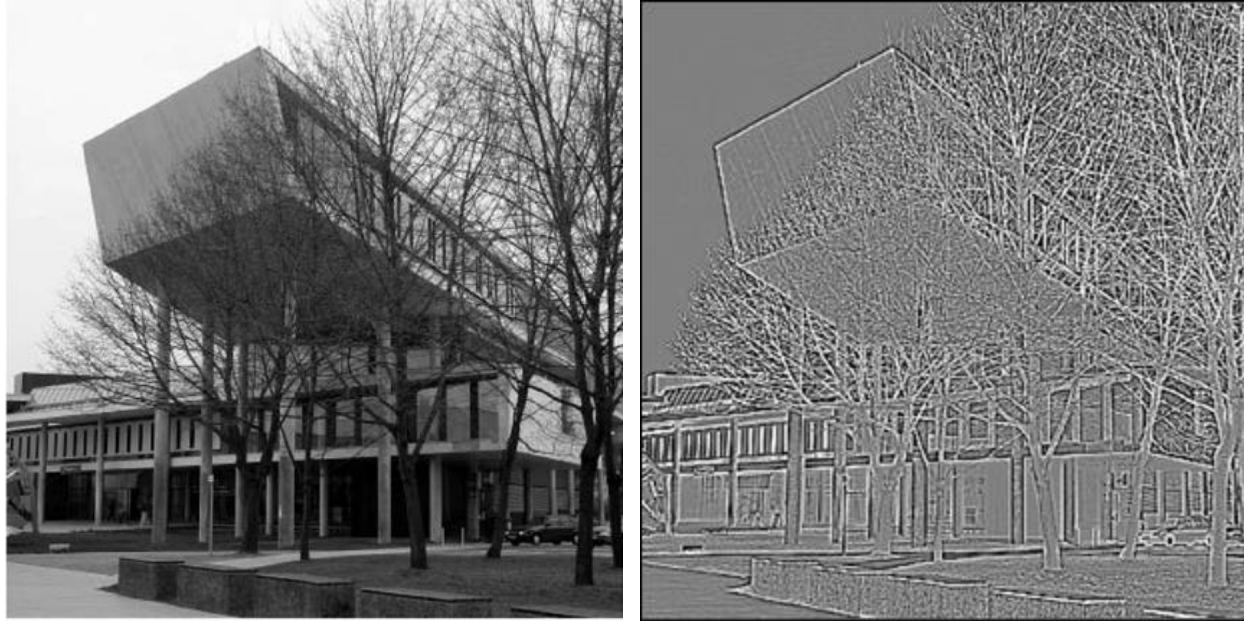
$$L_8 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \xrightarrow{\text{häufige Variante}} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Laplace-Operator (6)

- Nach Anwendung des Laplace-Operators sind die Positionen der Nulldurchgänge im Bild abzuleiten.
- Für den L_4 -Laplace-Operator:
 - das Bild wird dazu zunächst jeweils um einen Pixel nach links und um einen Pixel nach unten verschoben
 - ein Nulldurchgang liegt vor, wenn das Ergebnis des L_4 -Laplace-Operators **unterschiedliche Vorzeichen im unverschobenen Bild und einer der beiden verschobenen Versionen** zeigt
- Für den L_8 -Laplace-Operator erfolgt eine analoge Untersuchung. Allerdings ist das Bild zusätzlich auch in beide Diagonalenrichtungen zu verschieben

Laplace-Operator (7)

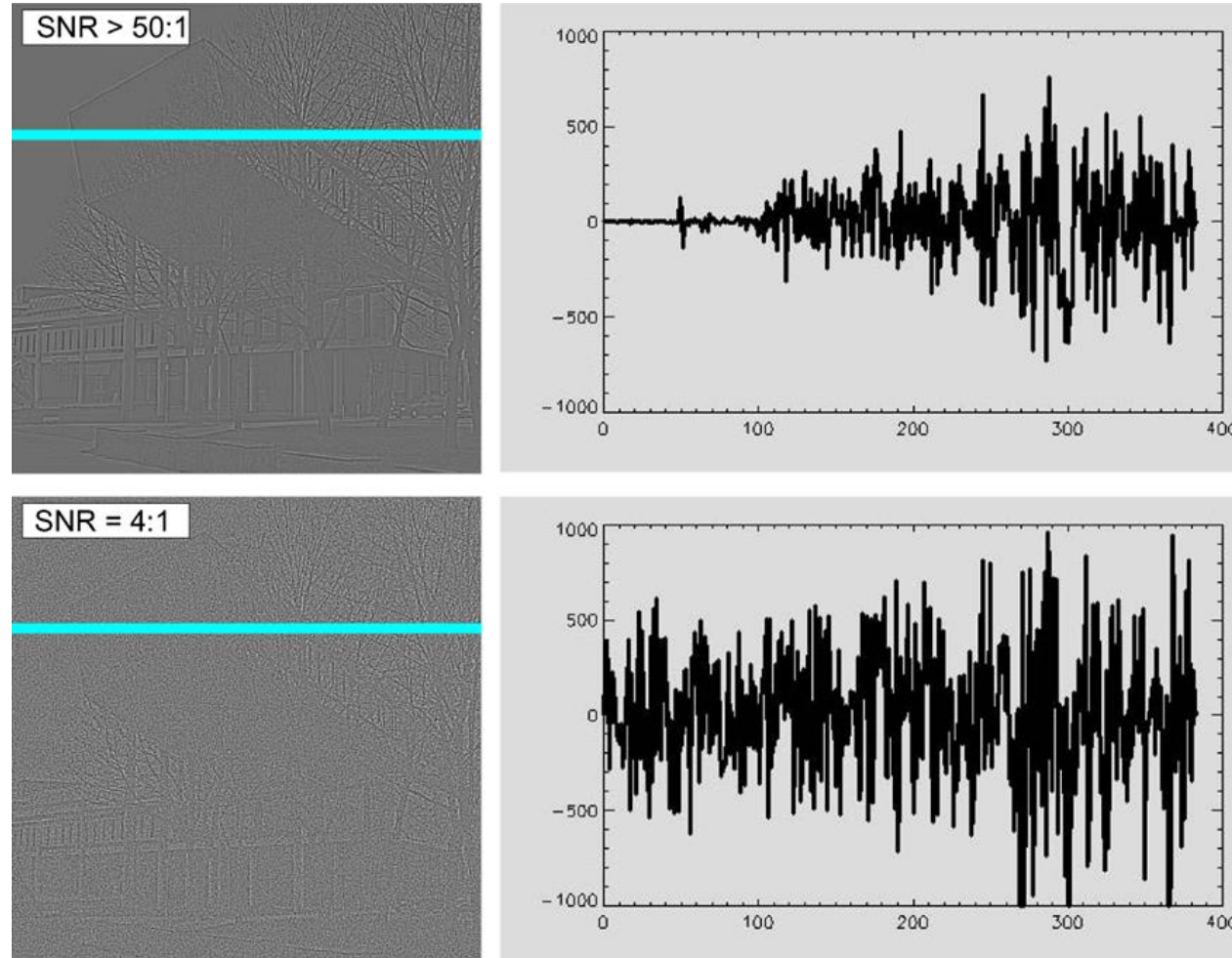
- Beispiel: gezeigt sind die Nulldurchgänge des Laplace-Operators:
 - dunkle Pixel bezeichnen negative Werte (Hell-Dunkel-Übergänge)
 - helle Pixel bezeichnen positive Werte (Dunkel-Hell-Übergänge)



Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

Laplace-Operator (8)

Der Laplace-Operator basiert auf einer zweimaligen Differentiation und ist daher sehr rauschempfindlich:



Filterantworten des Laplace-Operators in den blauen Bildzeilen.

Laplacian-of-Gaussian-Operator (1)

- Wegen der hohen Rauschempfindlichkeit des Laplace-Operators ist eine Kopplung mit einer vorherigen Glättung sinnvoll
- Der bekannteste Operator hierfür ist das **Laplacian-of-Gaussian-Filter** (kurz: **LoG-Filter**)
- Das Filter entsteht durch Anwendung des Laplace-Operators auf die Gauß-Funktion:

$$f_{G,\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

Laplacian-of-Gaussian-Operator (2)

- Die Anwendung des Laplace-Operators auf die Gauß-Funktion ergibt :

$$LoG(x, y) = \nabla^2 f_{G,\sigma}(x, y)$$

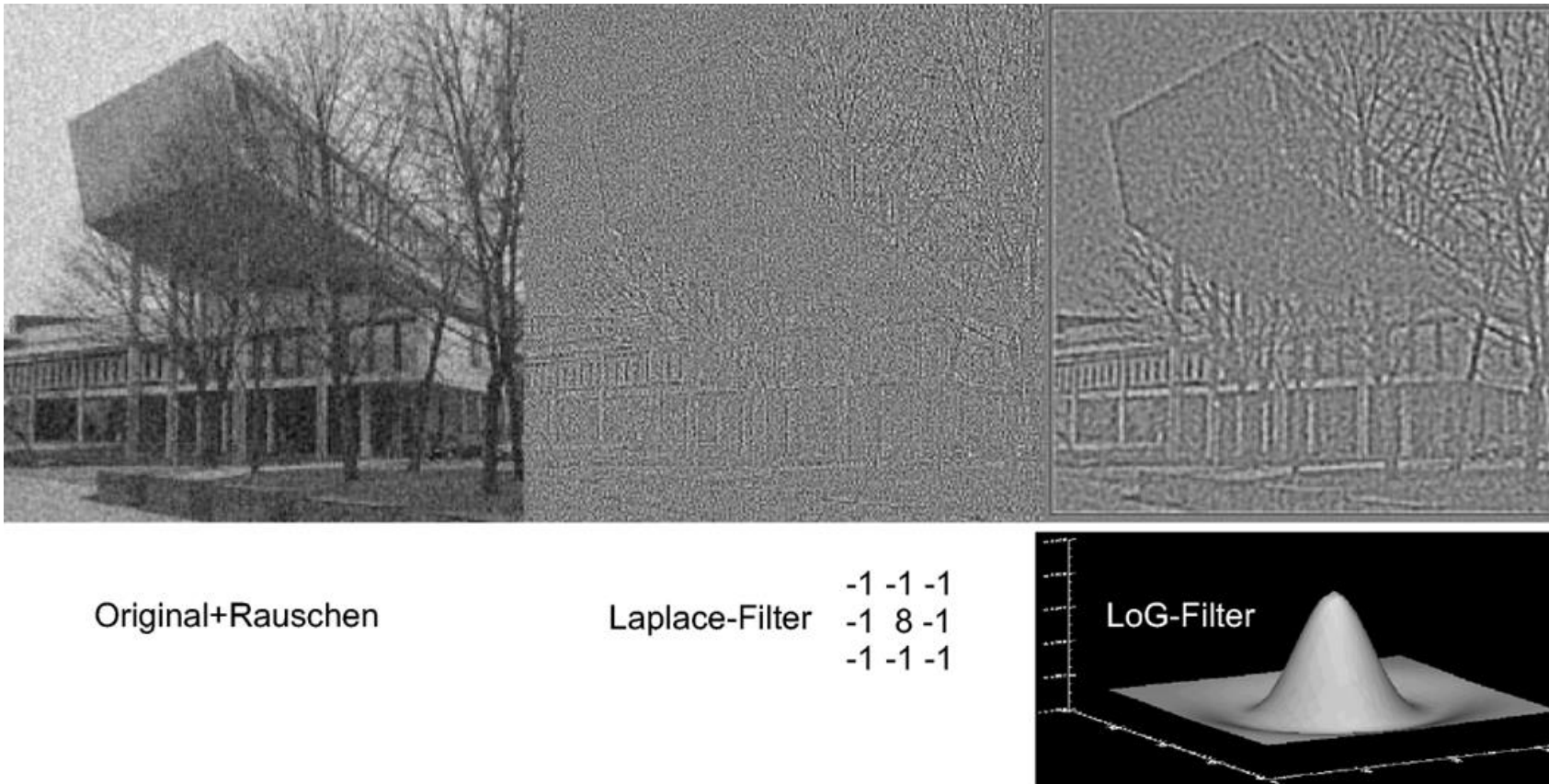
$$= \frac{\partial^2 f_{G,\sigma}}{\partial x^2}(x, y) + \frac{\partial^2 f_{G,\sigma}}{\partial y^2}(x, y)$$

$$= -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}.$$

- Wie bei der Diskretisierung der Gauß-Funktion ist auch beim LoG darauf zu achten, dass die Filtergröße k der Standardabweichung σ angemessen ist (z.B. $k = 2 \cdot \lceil 3\sigma \rceil + 1$)
- Durch Normierung ist zu sichern, dass die Summe der Filterwerte Null ergibt

Laplacian-of-Gaussian-Operator (3)

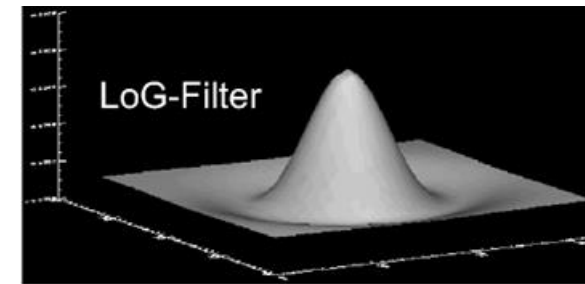
- Das LoG-Filter kann wegen der integrierten Rauschunterdrückung Nulldurchgänge von Kanten besser hervorzuheben als das Laplace-Filter:



Laplacian-of-Gaussian-Operator (4)

Das LoG-Filter ist auch bekannt als

- **Mexican-Hat-Filter** wegen seiner Form;
- **Marr-Hildreth-Filter**: der Informatiker David Marr und die Kognitionspsychologin Ellen Hildreth zeigten, dass die Signalverarbeitung in einer frühen Stufe der natürlichen visuellen Verarbeitung der LoG-Filterung ähnlich ist



Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

Laplacian-of-Gaussian-Operator (5)

- Zur Bestimmung von Nulldurchgängen werden auch oft **Folgen von LoG-Filtern mit unterschiedlichen Standardabweichungen** eingesetzt
- Orte, an denen Nulldurchgänge mit *verschiedenen* Filtergrößen erkannt werden, sind mit größerer W'keit Teil eines Kantenzuges als solche, die nur singulär mit einer bestimmten Standardabweichung erkannt werden
- Zusammen mit der integrierten Rauschunterdrückung führt diese Strategie bei verrauschten Bildvorlagen zu besseren Erkennungsraten

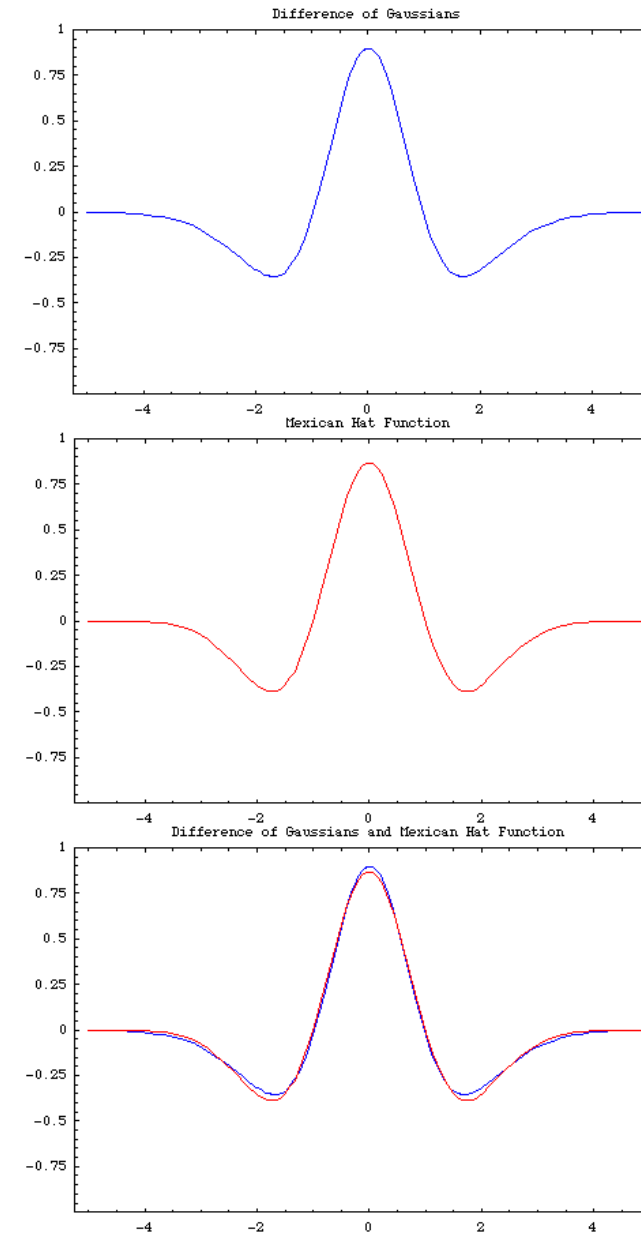
Difference-of-Gaussian-Operator (1)

- Der **Difference-of-Gaussian-Operator** (kurz: **DoG**) ist ein weiterer Operator, der die 2. Ableitung mit Rauschunterdrückung kombiniert
- Das Prinzip:
 - das Eingangsbild wird mit **zwei Gauß-Filtern mit unterschiedlichen Standardabweichungen σ_1 bzw. σ_2** geglättet
 - Die beiden **geglätteten Ergebnisbilder** werden **voneinander subtrahiert**:

$$DoG(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

Difference-of-Gaussian-Operator (2)

Die Differenz beider Gauß-Funktionen ist dem Mexican Hat und damit dem LoG-Filter ähnlich, da die Gauß-Funktion mit kleinerem σ ein höheres Maximum hat und schneller gegen Null geht als eine Gauß-Funktion mit größerem σ :



Bildquelle: http://en.wikipedia.org/wiki/Difference_of_Gaussians
(10.11.2011)

Difference-of-Gaussian-Operator (3)

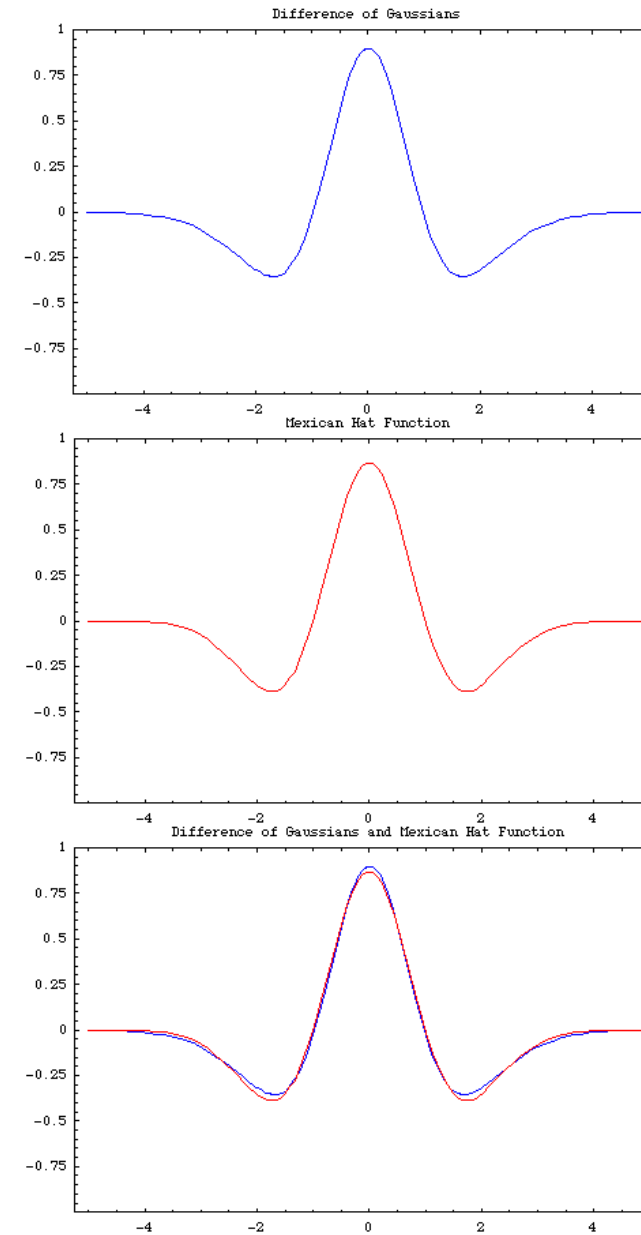
Genauer erfolgt die Approximation eines LoG-Filters mit σ durch ein DoG-Filter nach

$$DoG(x, y, \sigma) =$$

$$\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{1}{2\pi K^2\sigma^2} e^{-\frac{x^2+y^2}{2K^2\sigma^2}}$$

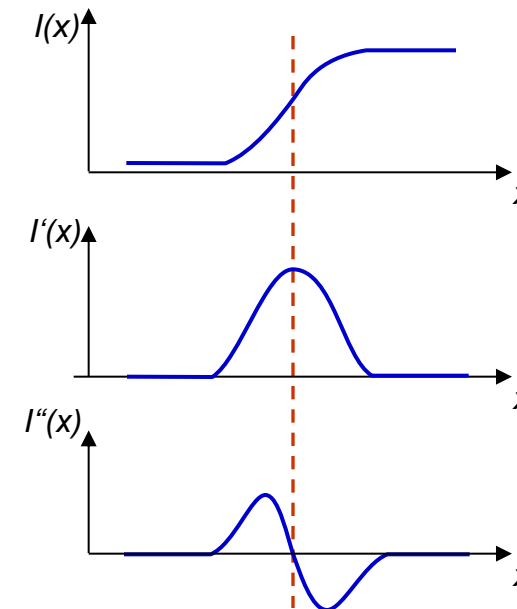
mit $K \sim 1,6$

Bildquelle: http://en.wikipedia.org/wiki/Difference_of_Gaussians
(10.11.2011)



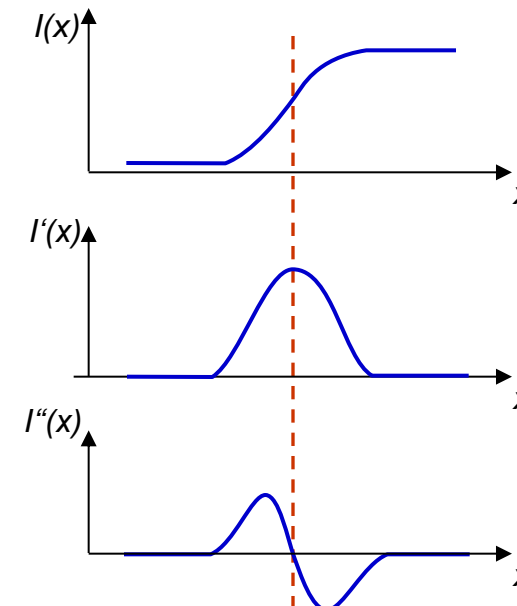
Positionsbestimmung von Nulldurchgängen (1)

- Die genaue Positionsbestimmung von Nulldurchgängen kann bei ver-
rauschten Bildern trotz integrierter Rauschunterdrückung (LoG, DoG)
schwierig sein.
- Hier kann ein **Kantenmodell, das durch Rauschen bedingte Unschärfe
berücksichtigt** und damit anschaulich „verschmierten Kanten“ beschreibt,
helfen.
- Das Modell geht davon aus, dass um einen
gefundenen Nulldurchgang bzw. Kantenort
eine lokale Umgebung in Gradientenrichtung
existiert, in der sich *keine andere Kante*
befindet.



Positionsbestimmung von Nulldurchgängen (2)

- Daher werden die **Gradientenantworten** innerhalb dieser Umgebung betrachtet **und deren Schwerpunkt als Ort des Nulldurchgangs** festgelegt.
- Innerhalb dieser Umgebung müssen die Integrale der 1. Ableitung in beiden Richtungen zu den Umgebungsgrenzen gleich sein.
- Also werden in diesem Ansatz die 1. Ableitung und die 2. Ableitung gemeinsam ausgewertet.



Positionsbestimmung von Nulldurchgängen (3)

Die subpixelgenaue Positionsbestimmung von Nulldurchgängen:

- 1) Berechnung von Gradienten *und* Nulldurchgängen
 - z.B. durch Gauß-Filterung gefolgt von *Sobel*- und *Laplace*-Filterung
- 2) An jedem Nulldurchgang
 - a) Bestimmung eines Umgebungsintervalls *in Gradientenrichtung* (Sobel-Operator) *um den Nulldurchgang* (ermittelt von Laplace-Operator)
 - b) Schwerpunktbestimmung der Gradientenbeträge im Umgebungsintervall
- 3) Der Schwerpunkt ist die gesuchte Position des Nulldurchgangs.

Fazit zur linearen Filterung (1)

- Kontrastverbesserung durch lineare Filterung erfolgt durch
 - Unterdrückung von Störsignalen,
 - Hervorhebung bzw. Extraktion von Kantensignalen.
- LEIDER führen die linearen Operatoren zur Rauschunterdrückung auch zur Kantenunterdrückung.
- DAHER muss in der Anwendung von linearen Operatoren häufig eine bzgl. des Anwendungsbereichs angemessene Balance zwischen Rauschunterdrückung und Kanten hervorhebung gewählt werden.

Fazit zur linearen Filterung (2)

- Die bei linearen Filtern notwendige Balance zwischen Rauschunterdrückung und Kantenhervorhebung ist für viele Anwendungen akzeptabel, da diese Balance hinsichtlich der Anwendung angepasst werden kann.
- Entsprechend gehören lineare Filter zum Standardmethodenarsenal des Computersehens und insbes. der *Low-Level-Vision* bzw. der *Bildverarbeitung*.

Zusammenfassung (1)

- Ein Ansatz der Kantenhervorhebung basiert auf der 1. Ableitung der Intensitätsfunktion eines Bildes. Für digitale Bilder wird die 1. Ableitung der zweidimensionalen diskreten Intensitätsfunktion durch Differenzen in x- und y-Richtung approximiert. Dies führt z.B. zum Sobel-Operator.
- Prinzipiell ist die Konvolution auch durch eine Korrelation zu implementieren. Bei asymmetrischen Filterkernen sind dabei Richtungswechsel in den Orientierungswerten zu beachten.

Zusammenfassung (2)

- Ein zweiter Ansatz der Kantenhervorhebung basiert auf der 2. Ableitung der Bildfunktion und sucht deren Nulldurchgänge. Dies führt zum Laplace-Operator.
- Wegen der Rauschempfindlichkeit des Laplace-Operators ist dessen Kopplung mit einer vorherigen Glättung sinnvoll. Dies führt zum Laplacian-of-Gaussian- sowie zum Difference-of-Gaussian-Operator.
- Für die genaue Positionsbestimmung von Nulldurchgängen werden 1. und 2. Ableitung der Bildfunktion gemeinsam analysiert.

Intelligente Sehsysteme

4 Nichtlineare Filter und Filtern von Farbbildern

Rangordnungsfiler, Medianfilter, Diffusionsfilter

RGB- und HSI-Farbmodelle,

Glättung und Kantenhervorhebung von RGB- und HSI-Farbbildern

Henry Hölzemann

- Nichtlineare Filter
 - Rangordnungsfilter
 - Medianfilter
 - Kantenerhaltung beim Medianfilter
 - Diffusionsfilter
 - isotrope homogene Diffusion
 - isotrope inhomogene Diffusion
 - anisotrope Diffusion
- Filterung von Farbbildern
 - Glättung von Farbbildern
 - Kanten hervorhebung in Farbbildern

Rückblick auf lineare Filterung

- Kontrastverbesserung durch **lineare Filterung** erfolgt durch
 - **Unterdrückung von Störsignalen**
(Mittelwertfilter, Gauß-Filter, Binomialfilter)
 - **Hervorhebung/Extraktion von Kantensignalen**
(Sobel-Operator, Laplace-Operator, LoG-Operator, DoG-Operator)
- **LEIDER** führen lineare Operatoren zur **Rauschunterdrückung** auch zur **Kantenunterdrückung**.
- **DAHER** muss i. A. eine bzgl. der Anwendung **angemessene Balance** zwischen **Rauschunterdrückung** und **Kantenhervorhebung** gewählt werden.

Nichtlineare Filterung (1)

- Die bei linearen Filtern notwendige Balance zwischen Rauschunterdrückung und Kantenhervorhebung ist für viele Anwendungen akzeptabel, da diese Balance hinsichtlich der Anwendung angepasst werden kann.
- Entsprechend gehören lineare Filter zum Standardmethodenarsenal des Computersehens und insbes. der Low-Level-Vision bzw. der Bildverarbeitung.

Nichtlineare Filterung (2)

- Für schwierige Anwendungsfälle kann aber eine effizientere Trennung zwischen Rauschunterdrückung und Kantenhervorhebung notwendig sein.
- Zwei Filterklassen können diese Trennung zwischen Rauschunterdrückung und Kantenhervorhebung effizienter gestalten als lineare Filter:
 - Rangordnungsfiler
 - Diffusionsfilter
- Beide Filterklassen gehören zur Klasse der nichtlinearen Filter.

Rangordnungsfilter (1)

Rangordnungsfilter

- werden **wie lineare Filter** auf ein Pixel p und seine Nachbarpixel angewandt – wir gehen hier zunächst von Nachbarpixeln innerhalb einer um p zentrierten quadratischen Umgebung *ungerader Größen* (3×3 , 5×5 , usw.) aus.
- berechnen im **Gegensatz zu linearen Filtern** keine gewichteten Summen über den Intensitäten innerhalb der Umgebung, sondern
 - **sortieren alle Pixel** innerhalb der Umgebung *gemäß ihrer Intensitätswerte*
 - **liefern als Ergebnis den an einem bestimmten Rang** der sortierten Reihenfolge **eingeordneten Wert**

Rangordnungsfiler (2)

- Die Anzahl der Operationen ist für lineare Filter linear in der Anzahl der Pixel der quadratischen Umgebung. Die Zeitkomplexität ist also $O(n)$ für $n = m^2$ und Größe m des Filterkerns.
- Die Zeitkomplexität für Rangordnungsfiler ist wegen der Sortierung dagegen $O(n \log n)$ für $n = m^2$ und Größe m des Filterkerns.
- Für kleine Filterkerne ist dies nicht relevant. Für sehr große Filterkerne und sehr große Bilder kann es zu spürbaren Unterschieden in den Berechnungszeiten kommen.*

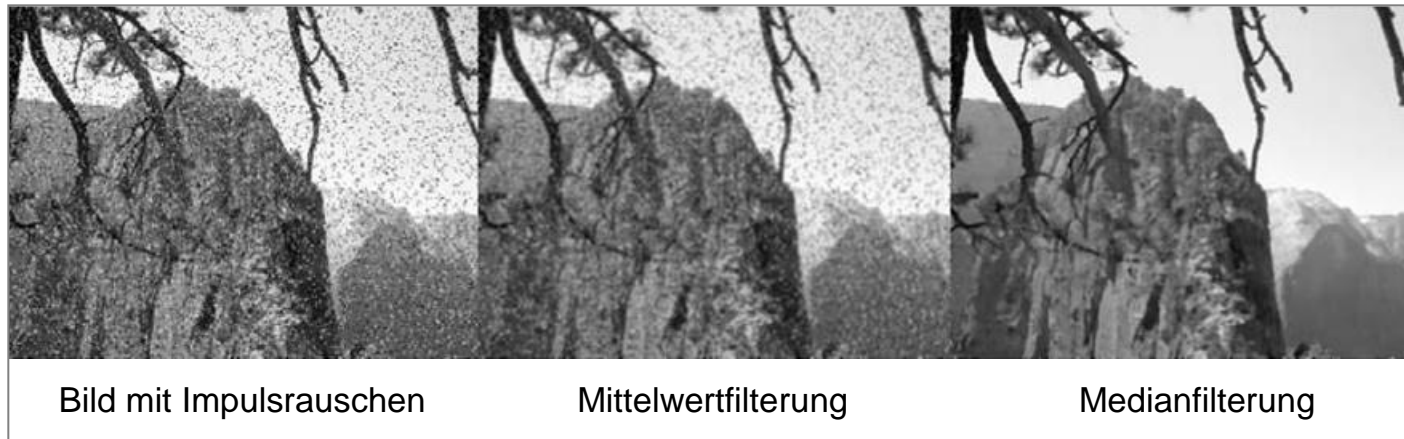
* Zumal sich für lineare Filter mit großen Filterkernen noch die FFT als effizienterer Weg anbietet. Die Konvolution eines Bildes der Größe $N \times N$ mit einem Faltungskern der Größe $m \times m$ hat die Zeitkomplexität $O(m^2 \cdot N^2)$. Über FFT ausgeführt ist sie $O(N^2 \cdot \log_2 N)$. Für $\log_2 N < m^2$ ist die Konvolution also über FFT effizienter ausführbar.

Medianfilter (1)

- Der **Median** ist der Wert in der mittleren Position einer sortierten Folge.
- Das **Medianfilter** ist ein Rangordnungsfilter, das als Ergebnis den Median liefert
- Median und Mittelwert einer sortierten Wertefolge sind i. A. ungleich, nähern sich aber für große Folgen normalverteilter Werte einander an
- Daraus folgt die Brauchbarkeit des Medianfilters für die Rauschunterdrückung
- Ist die ungestörte Bildfunktion $I_u(x,y)$ in der Filterumgebung konstant und der Erwartungswert des Rauschens Null, so entspricht der durch den Median angenäherte Erwartungswert der gesuchten ungestörten Bildfunktion.
- Ähnlich wie beim Mittelwertfilter wird die Näherung bei steigender Filtergröße des Medianfilters besser.

Medianfilter (2)

- Das Impulsrauschen (Salt-and-Pepper-Noise) ist ein Spezialfall des Rauschens
- Für das lineare Mittelwertfilter wären sehr große Bereiche nötig, um den Erwartungswert der ungestörten Bildfunktion anzunähern
- Da aber der Großteil der Intensitätswerte beim Impulsrauschen ungestört ist und damit dem Erwartungswert der ungestörten Bildfunktion entspricht, führt die Medianfilterung in homogenen Bereichen zur vollständigen Beseitigung des Impulsrauschens:



Kantenerhaltung beim Medianfilter (1)

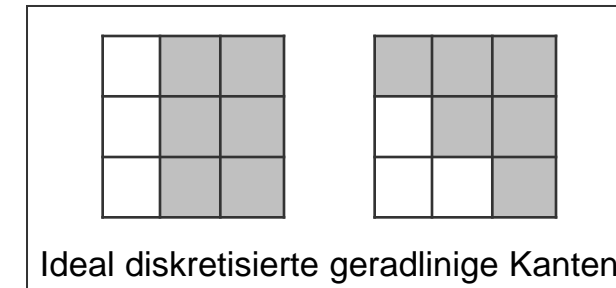
Unter den folgenden Umständen ist das Medianfilter kantenerhaltend:

- 1) die Intensitätswerte der ungestörten Bildfunktion $I_u(x,y)$
sind auf beiden Seiten der Kante jeweils *konstant*
- 2) der Signalabstand, d.h. der Werteunterschied an einer Kante,
ist größer als die Rauschamplitude
- 3) die Kante verläuft über die Fläche des Medianfilters gerade

Kantenerhaltung beim Medianfilter (2)

Diskussion:

- Das Medianfilter werde an einer Kante k angewandt
- In der sortierten Folge der Pixelintensitäten bezeichnen M_D bzw. M_H die Mengen der Pixel auf der dunkleren bzw. helleren Seite der Kante k
- Gilt Bedingung 1 (konstante Werte auf jeder Seite von k), werden alle Werte aus M_D vor allen Werten aus M_H einsortiert
- Gilt Bedingung 2 (Kantensignal stärker als Rauschen), bleibt die o.g. Sortierung der Werte aus M_D und M_H auch bei überlagertem Rauschen erhalten
- Gilt Bedingung 3 (gerade Kante), so umfasst M_D mehr Pixel als M_H , wenn das zentrale Pixel zu M_D gehört – und umgekehrt



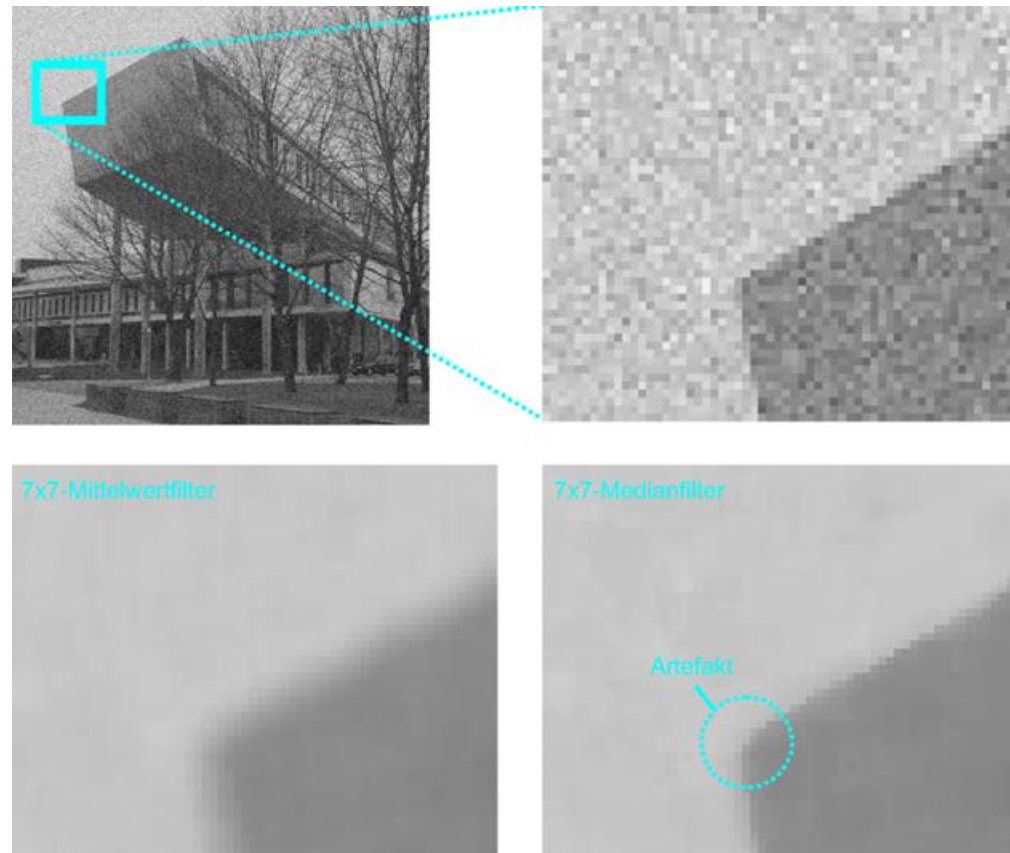
Bemerkung: das Medianfilter ist an einer Kante nicht mehr länger eine gute Näherung des Mittelwertfilters der Pixel einer homogenen Region, da er über Pixel aus zwei unterschiedl. Regionen gebildet wird. Aber genau das passt!

Kantenerhaltung beim Medianfilter (3)

- Von den drei Bedingungen ist die 3. Bedingungen für „Realweltszenen“ am schwierigsten einzuhalten:
 - Kantenzüge verlaufen selten über längere Strecken geradlinig!
- Wird das Medianfilter jedoch klein genug gewählt, so wird die 3. Bedingungen jedoch in den meisten Bildbereichen angenähert.
 - Die einzige **Ausnahme** sind dann noch **Eckpunkte**:
 - So klein der Filterkern auch gewählt wird, der Kantenverlauf ist an einer Ecke per Definition nicht geradlinig!
 - Somit verursacht das Medianfilter zunächst Artefakte an Ecken.

Kantenerhaltung beim Medianfilter (4)

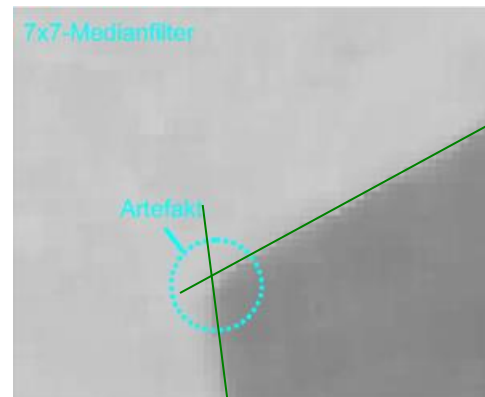
Das Medianfilter erhält Kanten nur, wenn sie innerhalb der Filterumgebung nahezu gerade verlaufen. Das ist an Ecken nicht der Fall:



Kantenerhaltung beim Medianfilter (5)

Die vom Medianfilter an Ecken erzeugten Artefakte sind ggf. (d.h. je nach Interpretation der Artefakte und nach Anwendung) aber auch kompensierbar:

- Wird das Artefakte als Positionsverschiebung der tatsächlich abgebildeten Objektecke interpretiert,
- so kann diese Eckenposition ggf. wieder durch die **Schnittbildung der extrahierten Kanten** zurück gewonnen werden:



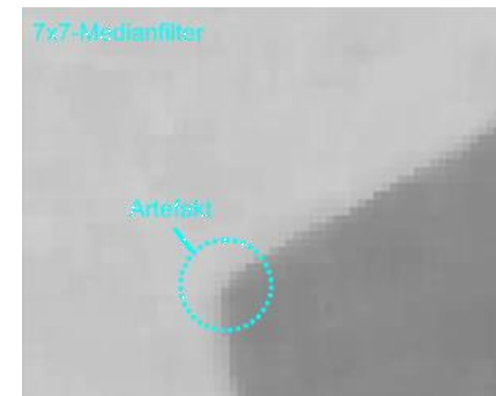
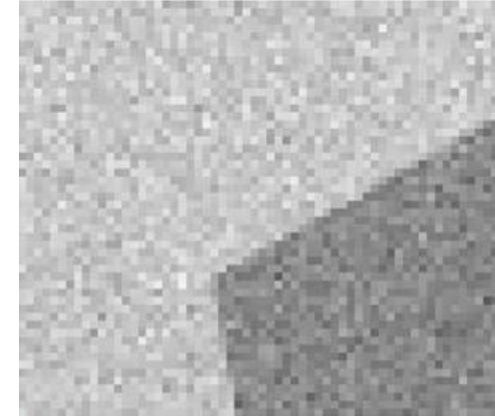
Kantenerhaltung beim Medianfilter (6)

Das Medianfilter wirkt also

- rauschunterdrückend in homogenen Bereichen und
- kantenerhaltend an Kanten.

Die Größe des Medianfilters sollte also eine Balance sein zwischen

- möglichst guter Annäherung an den Erwartungswert der ungestörten Bildfunktion $I_u(x,y)$ durch große Filter und
- einer Maximierung der Anzahl der Kantenpixel, die zu lokal geradlinig verlaufenden Kanten gehören.



Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.