Computergrafik 2: Filtern im Ortsraum

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz

michael.rohs@ifi.lmu.de

MHCI Lab, LMU München

Folien teilweise von Andreas Butz, sowie von Klaus D. Tönnies
(Grundlagen der Bildverarbeitung. Pearson Studium, 2005)

Themen heute

- Konvolution und Korrelation
- Lineare Filterung
- Nichtlineare Filterung
- Kantenmerkmale durch lineare Filterung
- Schärfen von Bildern
- Interpolation

Filterung im Ortsraum

- Lineare Filterung
- m×n Filtermaske
- Lokale Umgebung
- Vorgegebene Operation auf Pixeln in lokaler Umgebung
- Skalarprodukt
  f(x-1,y-1)*w(-1,-1) +
  ... + f(x,y)*w(0,0) +
  ... + f(x+1,y+1)* w(1,1)

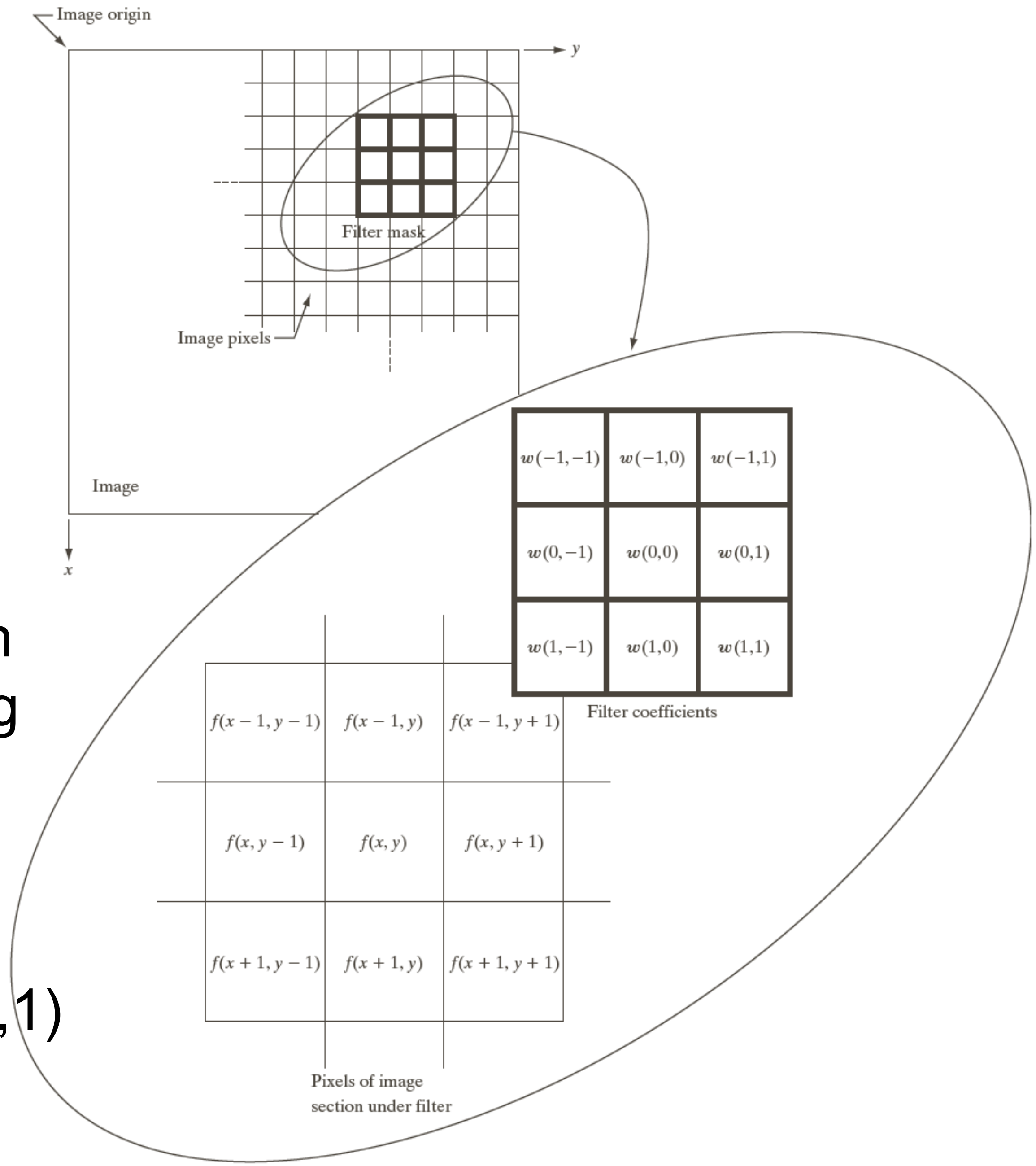


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Filtern im Ortsraum

- Filterung als gewichtetes Mittel

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

  – m×n Filtermaske mit m = 2a+1, n = 2b+1
  – M×N Bild mit M Zeilen und N Spalten

- Üblicherweise
  – Filtermaske begrenzt, Gewicht normalisiert
  – Seitenlänge des Filters ungerade

Filtergrößen

- Stärke der Glättung als Effekt der Filtergröße
  - Bild M×N = 500×500
  - quadratisches Boxfilter

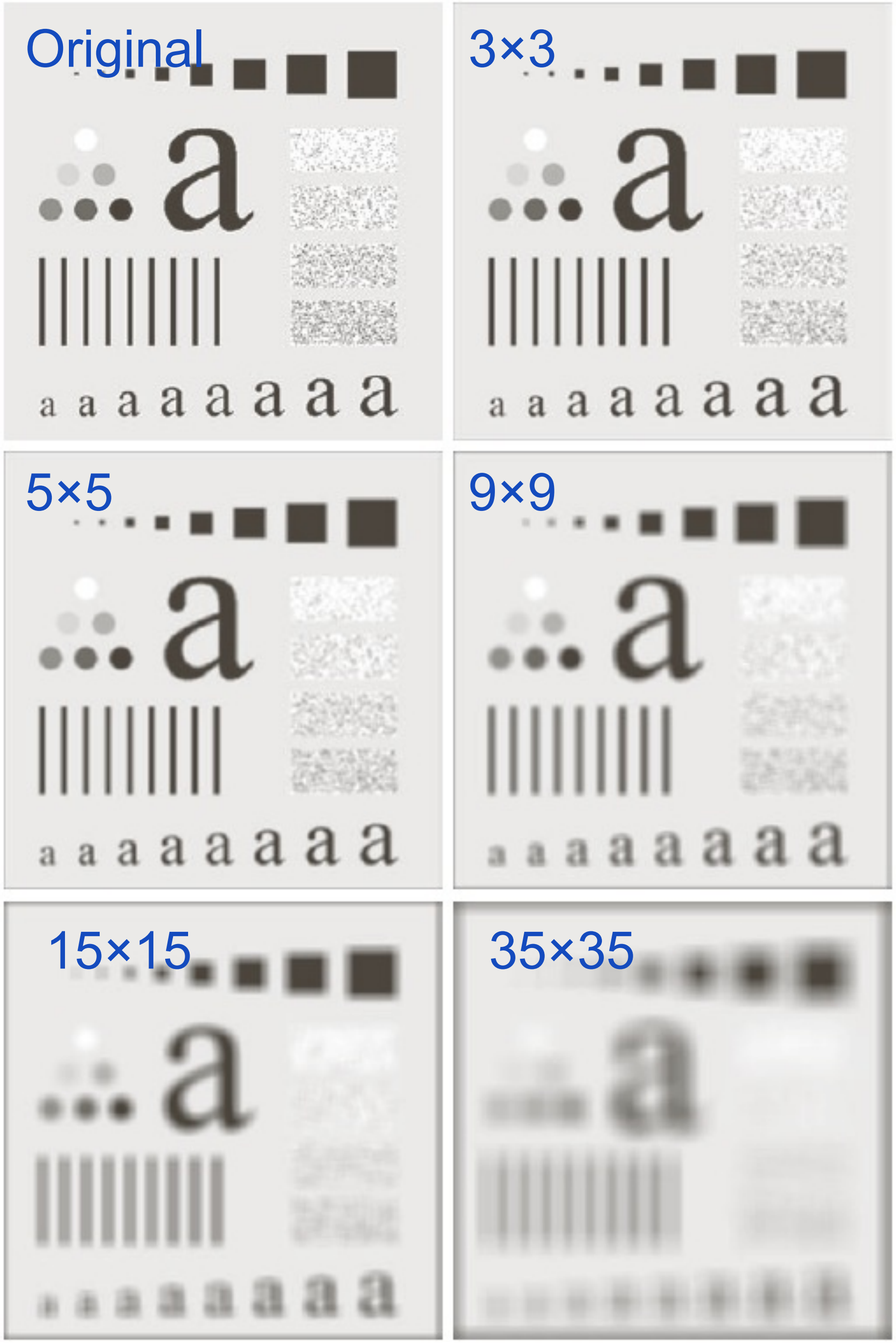
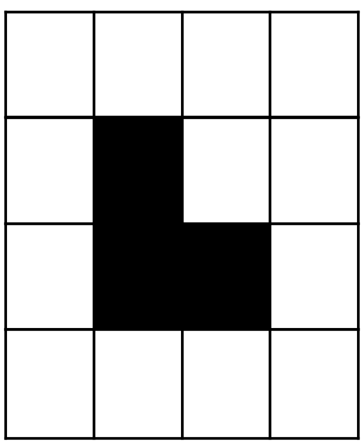


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Hintereinanderschreiben von Pixeln

- 2-dimensionales Bild (M Zeilen, N Spalten) wird zu MN Spaltenvektor



$$f = \begin{pmatrix} f(0,0) \\ f(1,0) \\ \vdots \\ f(N-1,0) \\ f(0,1) \\ \vdots \\ f(N-1,0) \\ \vdots \\ \vdots \\ f(0,M-1) \\ \vdots \\ f(N-1,M-1) \end{pmatrix}$$

Lineare Operatoren

- Definition eines linearen Operators O(.)

$$O(\alpha \vec{f} + \beta \vec{g}) = \alpha O(\vec{f}) + \beta O(\vec{g})$$

für alle Skalare α, β

- Superpositionsprinzip: komplexe Signale können in einfachere Komponenten zerlegt werden

Linearer Operator als Matrix

Linearer operator O lässt sich auch als Matrix ausdrücken:

Seien x,y die hintereinandergeschriebenen Pixel zweier Bilder, dann ist

$$\vec{y} = A\vec{x}$$

Jeder Eintrag $A_{i,j}$ gibt an, mit welchem Gewicht Pixel j aus x auf Pixel i in y abgebildet wird

Verschiebungsinvariante Operatoren

- … sind lineare Operatoren, deren Wirkung unabhängig vom Ort ist
- Beispiel: Filtermaske wirkt überall im Bild gleich
- Beispiel: gleichmäßige Unschärfe im Bild durch Bewegung der Kamera

Konvolution (Faltung)

- Seien f, g abgetastete Bilder mit unendlicher Größe, m,n, Skalare

$$(g * f)(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i,j)\, f(m-i, n-j)$$

Heißt Konvolution der Funktion f mit g

- g heißt die Konvolutionsfunktion
- Funktioniert so nur für unendlich große Bilder

Eigenschaften der Konvolution

- Linear & verschiebungsinvariant
- Kommutativ & assoziativ

$$[g_1 * g_2](m,n) = [g_2 * g_1](m,n)$$

$$g_1 * \left([g_2 * g_3](m,n)\right) = [g_1 * g_2](m,n) * g_3(m,n)$$

- D.h. wir können mehrere Konvolutionen vorab kombinieren und dann gemeinsam anwenden

Konvolution vereinfacht

Hat die Konvolutionsfunktion g nur einen begrenzten Bereich, in dem g ≠ 0, dann heißt dieser Bereich Kern von g (kernel)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	-1	0	1	0	0	0
0	0	0	-1	0	1	0	0	0
0	0	0	-1	0	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Konvolution anschaulich

1	0	-1	0	0	0	0
1	0	-1	1	0	0	0
1	0	-1	1	1	0	0
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

	1	2	0	-2	-1	
	2	2	0	-2	-2	
	3	2	0	-2	-3	
	3	2	0	-2	-3	
	2	2	0	-2	-2	

Worst case Aufwand im Ortsraum: N^4 für Kantenlänge N des Bildes

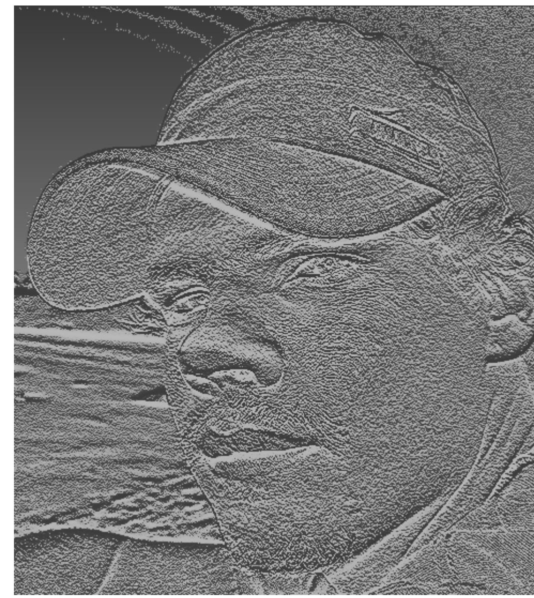
Verwendung der Konvolution

- Sobel-Operator: Kanten finden

*

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

=



Konvolution des Dirac-Impulses

Dirac-Impuls
= Einheitsimpuls
= unit impulse

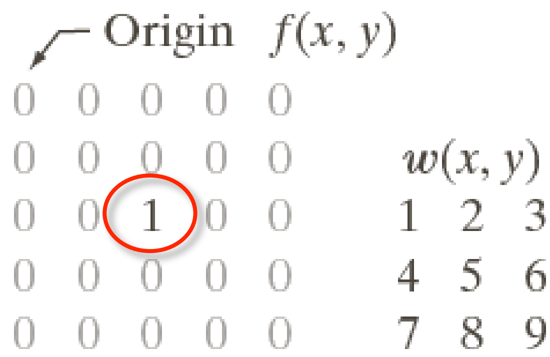


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

= Punktverteilungs-funktion
= Point Spread Function (PSF)

Konvolution des Dirac-Impulses

Dirac-Impuls
= Einheitsimpuls
= unit impulse

Origin $f(x, y)$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$w(x, y)$

1	2	3
4	5	6
7	8	9

Padded f

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Full convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Cropped convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Punktantwort
= Punktverteilungs-funktion
= Point Spread Func-tion (PSF)

Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Punktantwort (Point Spread Function, PSF)

- Faltung eines einzelnen Dirac-Impulses
  – Bild mit einem einzigen weißen Pixel
- Abgetastetes Bild =
  Folge von Dirac-Impulsen * Pixelhelligkeiten
- → Gesamtwirkung durch PSF vollständig beschrieben
- → falls PSF umkehrbar, kann Wirkung rückgängig gemacht werden
- PSF kann manchmal experimentell bestimmt werden

Punktantwort (Point Spread Function, PSF)

- Alternative Repräsentation eines linearen Operators

$$g(\alpha,\beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cdot h(x,\alpha,y,\beta)$$

- Einheitsimpuls an Pixel (a,b)

$$\delta(x-a, y-b) = \begin{cases} 1 & x = a \wedge y = b \\ 0 & sonst \end{cases}$$

- Punktantwort

$$g(\alpha,\beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(x-a, y-b) \cdot h(x,\alpha,y,\beta) = h(a,\alpha,b,\beta)$$

PSF

Dekomposition eines Bildes in Pixel

- Digitales Bild als Summe seiner Pixel

$$f(x,y) = \sum_{a=0}^{N-1} \sum_{b=0}^{M-1} f(a,b) \cdot \delta(x-a, y-b)$$

- Das Resultat einer linearen Filterung ist die Überlagerung der Punktantworten

$$g(\alpha,\beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cdot h(x,\alpha,y,\beta)$$

$$= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{a=0}^{N-1} \sum_{b=0}^{M-1} f(a,b) \cdot \delta(x-a, y-b) \cdot h(x,\alpha,y,\beta)$$

Separierbarkeit linearer Filter

- Ein zweidimensionales Filter ist separierbar, falls Punktantwort durch Hintereinanderausführung zweier eindimensionaler Impulsantworten darstellbar
- Strategie: zerlege 2D Filter in einen x- und y-Kern die hintereinander angewandt werden
- Separierbarkeit in h_x und h_y:

$$g(\alpha,\beta) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\cdot h(x,\alpha,y,\beta)$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\cdot h_x(x,\alpha)\cdot h_y(y,\beta)$$

Separierbarkeit linearer Filter

- Verarbeitung zeilenweise

$$g(\alpha,\beta) = \sum_{y=0}^{M-1} h_y(y,\beta) \sum_{x=0}^{N-1} f(x,y) \cdot h_x(x,\alpha)$$

- Verarbeitung spaltenweise

$$g(\alpha,\beta) = \sum_{x=0}^{N-1} h_x(x,\alpha) \sum_{y=0}^{M-1} f(x,y) \cdot h_y(y,\beta)$$

- Reduziert Rechenaufwand von O(NM) auf O(N+M)
- Beispiel:

$$h = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, h_x = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, h_y = \frac{1}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Korrelation

- Seien f, g abgetastete Bilder mit unendlicher Größe, m,n, Skalare

$$(g * f)(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i,j)\, f(i-m, j-n)$$

Heißt Korrelation der Funktion f mit g

- g heißt die Korrelationsfunktion
- Funktioniert so nur für unendlich große Bilder → Vereinfachung siehe Konvolution

Korrelation

Kern nicht gedreht, sonst wie bei Konvolution

Origin $f(x, y)$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$w(x, y)$

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

(a)

Padded f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(b)

Initial position for w

$$\begin{matrix} 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 8 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Full correlation result

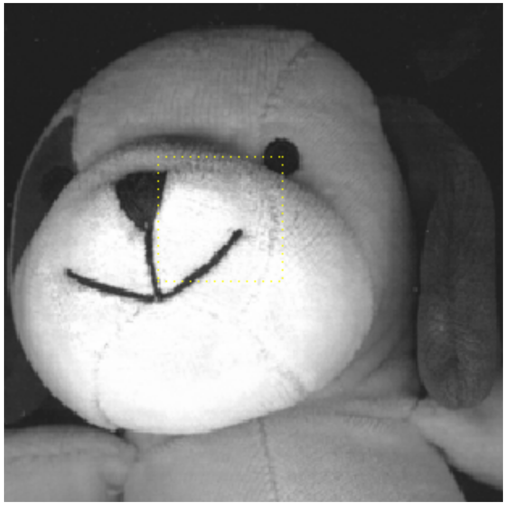
$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 8 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 5 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Cropped correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 8 & 7 & 0 \\ 0 & 6 & 5 & 4 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

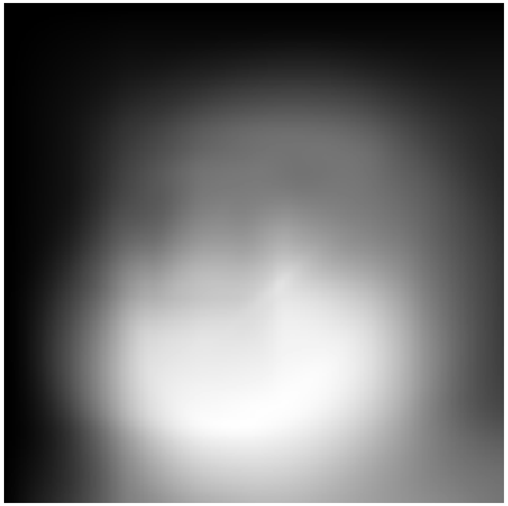
Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Verwendung der Korrelation

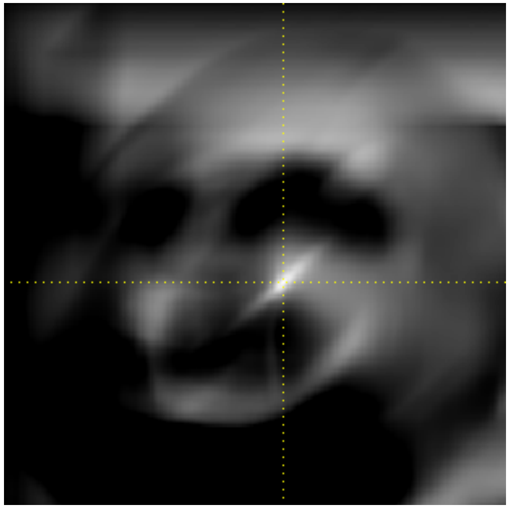


Ausgangsbild (f)

Korrelationsfunktion (g)



Ergebnis (g*f)



...normalisiert

Unterschiede Konvolution / Korrelation

- Konvolution und Korrelation sind zwei eng verwandte Filter-Operationen
- Gleich, falls der Kern von g symmetrisch unter 180° Rotation
- Beispiele:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Tiefpass

-1	-1	-1
-1	9	-1
-1	-1	-1

Hochpass

Tiefpass: Wirkung

$$* \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} =$$

Nützlich z.B. gegen Rauschen und Alias-Effekte

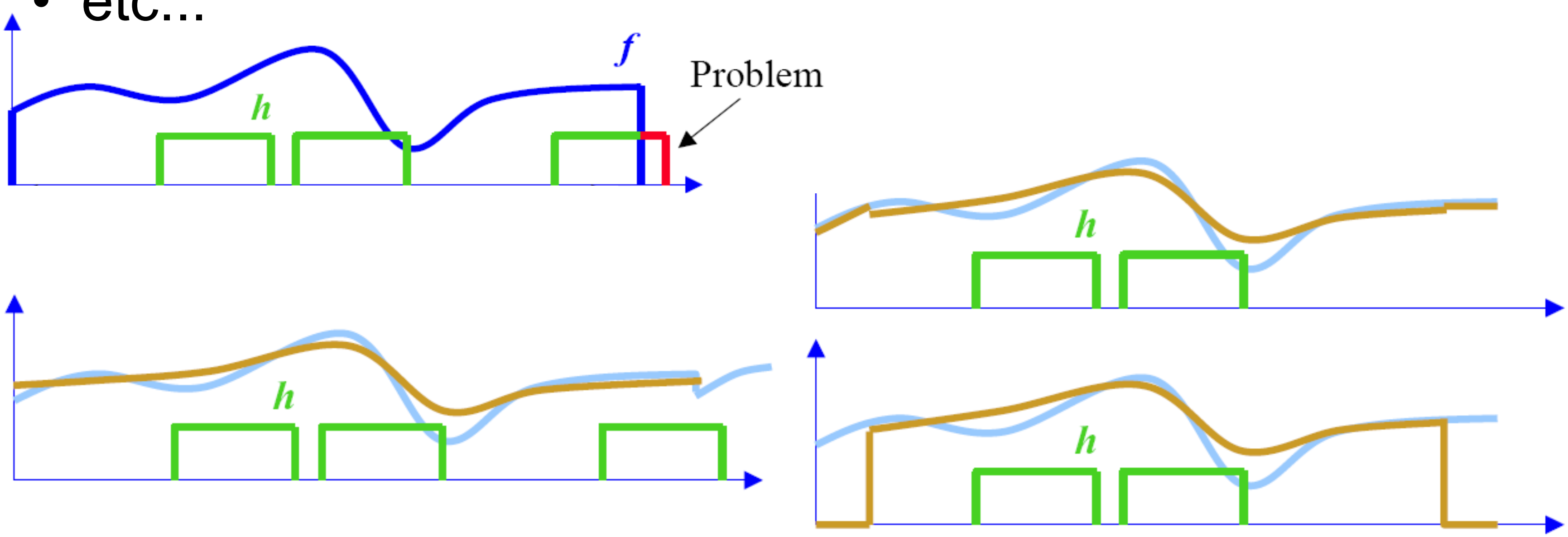
Hochpass: Wirkung

$$* \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$

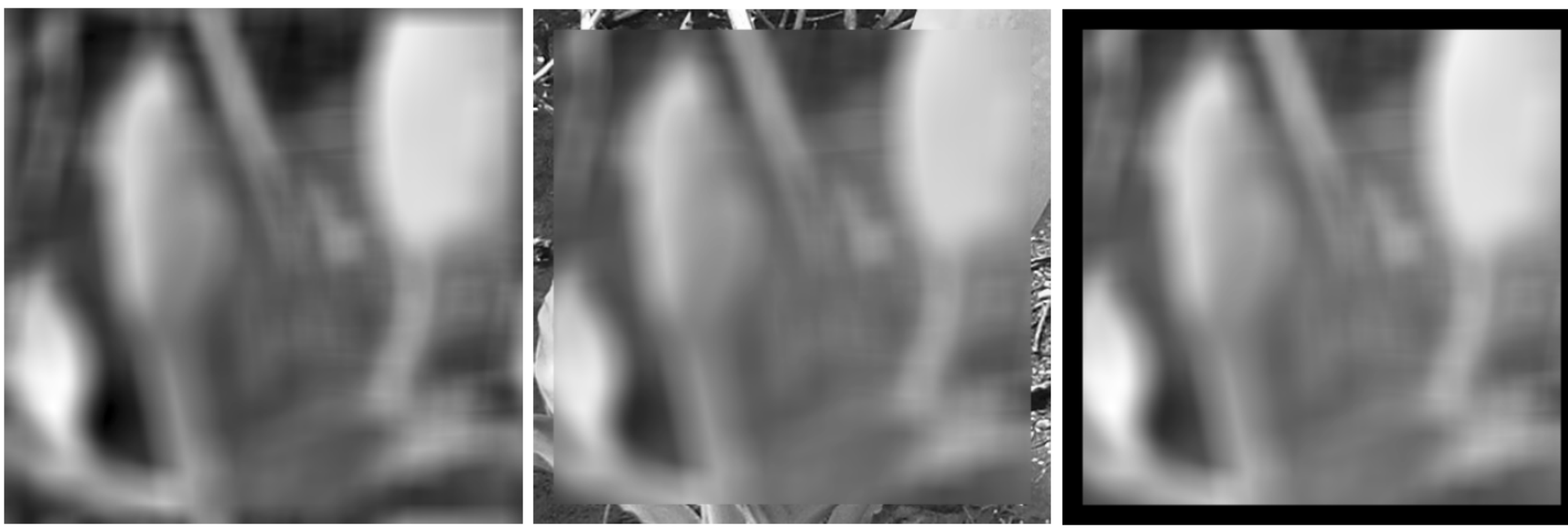
Nützlich zum Scharfzeichnen und Kanten finden

Konvolution am Bildrand

- Lösung1: periodische Fortsetzung des Bildes
- Lösung 2: Spiegeln am Rand
- Lösung 3: Rand konstant fortsetzen
- Lösung 4: Werte außerhalb des Bildes auf 0 setzen
- etc...



Konvolution am Bildrand



Flächenbasierte Bildverbesserung

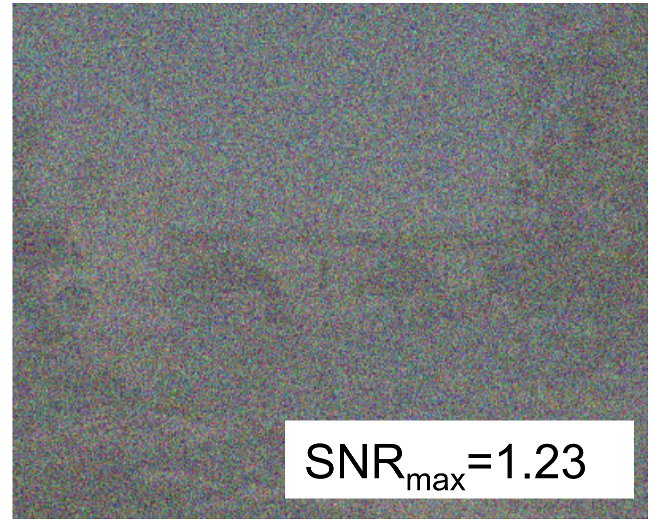
- Rauschen kann durch Integration einer Signalfolge mit (nahezu) konstantem Signal reduziert werden
- Konstante Signalfolge
  - Integration über eine zeitliche Folge
  - Integration über eine homogene Fläche
- Lineare verschiebungsinvariante Operatoren
  - Konvolutionsmethoden

Zeitliche Folge

- Annahmen
  - Aufnahme mehrerer Bilder g_i, i=1,I über einen gegebenen Zeitraum
  - Bild verändert sich über den Zeitraum nicht (keine Bewegung, keine Beleuchtungsänderung)
  - Erwartungswert E des Rauschens n ist 0
- Näherung an die unverrauschte Funktion f
  - $E\{g(m,n)\} = E\{f(m,n)\} + E\{n(m,n)\}$
    $= E\{ f(m,n) \} + 0 = f(m,n)$
  - Abschätzung von $E\{g(m,n)\}$ durch Integration über die Bilder

Beispiel

- Einzelne Aufnahme mit normalverteiltem Rauschen (SNR≈1.2)
- SNR_{max} = max. Signalamplitude / Standardabweichung des Rauschens
- Addition von 10 bzw. 50 Aufnahmen



Integration über die Fläche

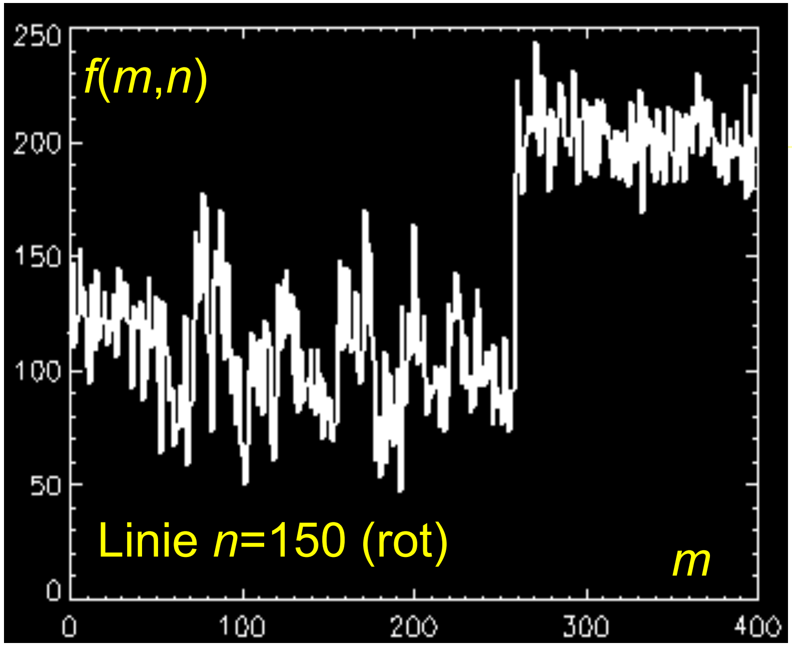
- Falls für eine Reihe von Bildpunkten $(p_0,\ldots,p_n)$ gilt, dass $f(p_i)$=const, dann kann Rauschen n mit $E\{n\}=0$ durch Addition der gemessenen Funktionswerte $g(p_i)$ reduziert werden

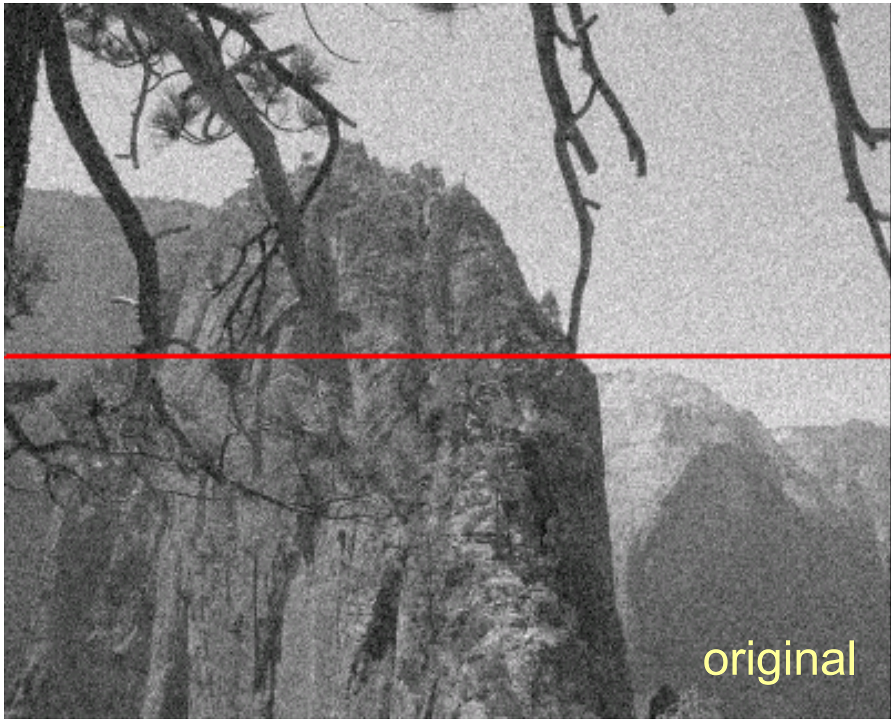
- Annahmen:
  - Bild besteht aus homogenen Bereichen
  - Benachbarte Punkte haben den gleichen Grauwert

- Rauschunterdrückung:
  - Mittelwertbildung über vorgegebene Nachbarschaft

Mittelwertbildung durch Konvolution

Konvolutionskern: Gleichmäßige Gewichtung der Pixel
in einer gegebenen Nachbarschaft

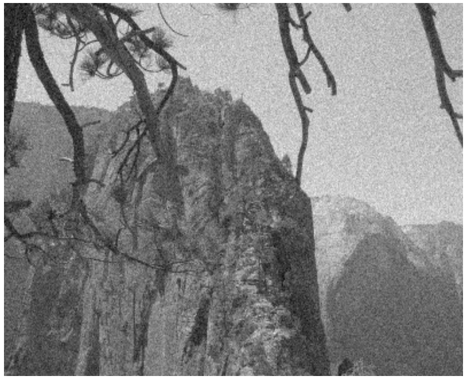


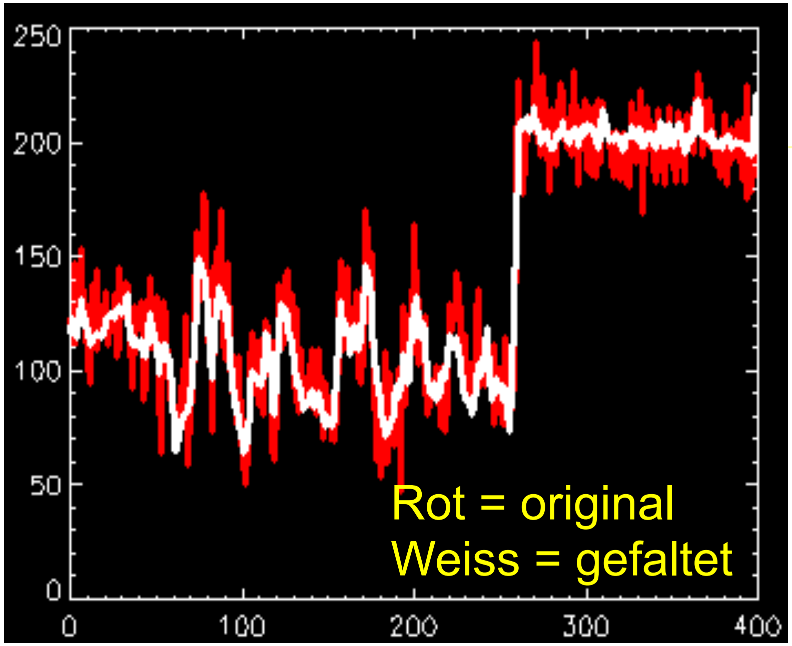


3x3 Boxcar-Filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Filterkern

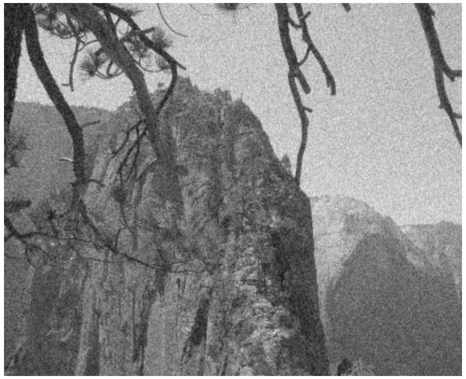


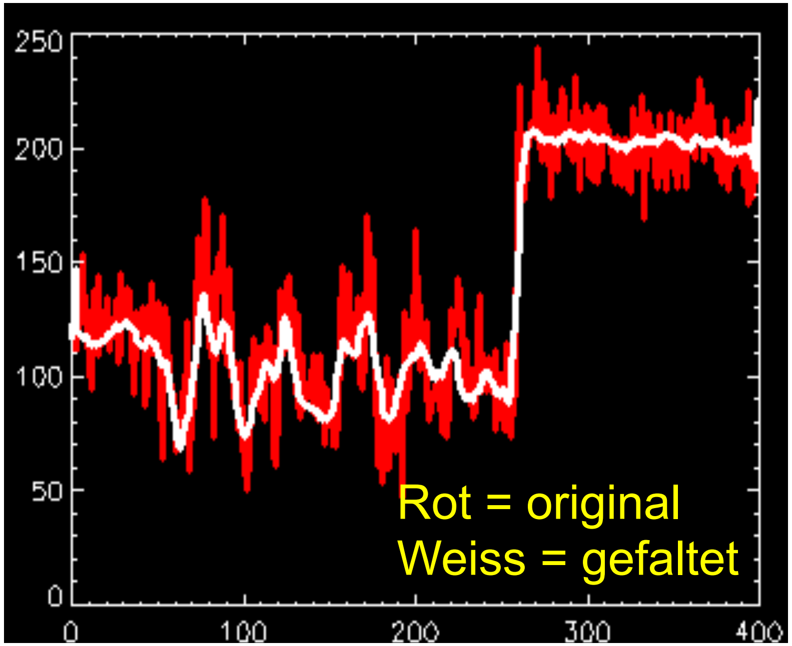


7x7 Boxcar-Filter

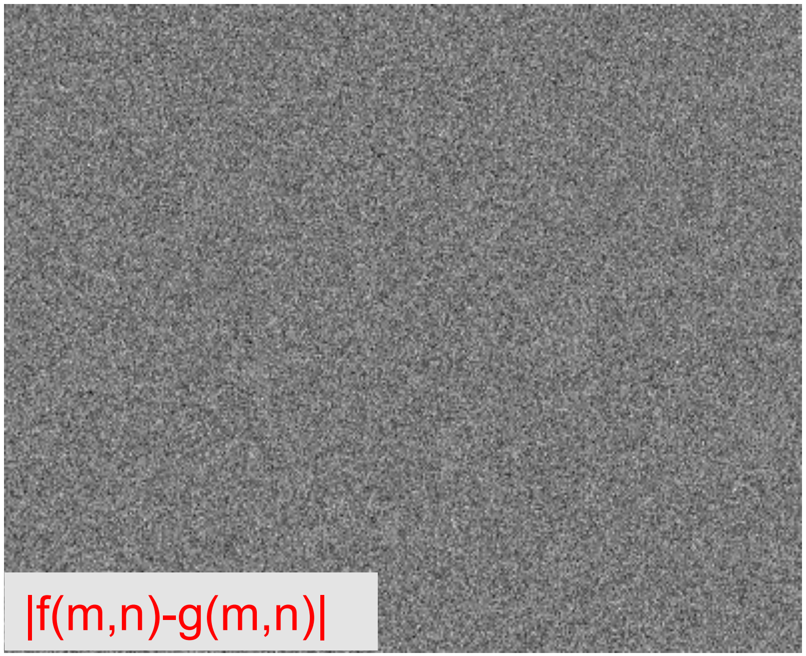
Beobachtung: Kanten werden degradiert

Grund: Annahme konstanter Funktionswerte ist nicht wahr

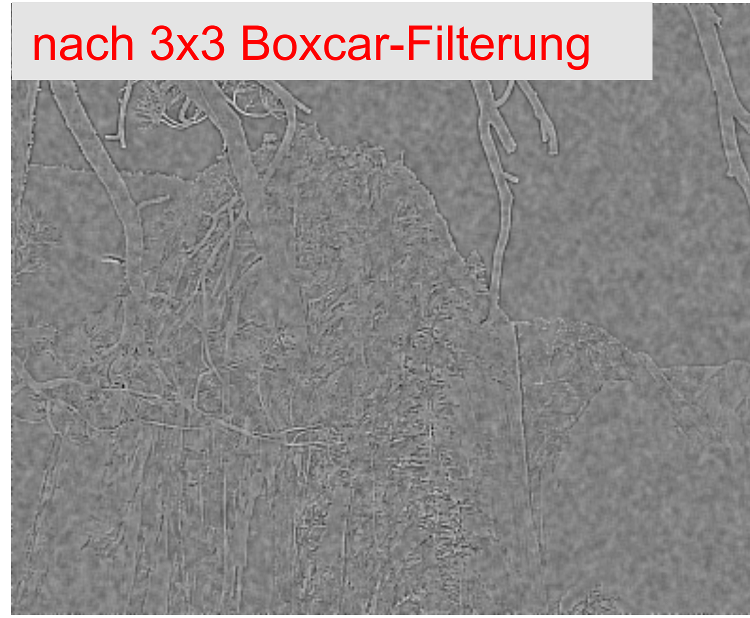




Verhalten an Kanten



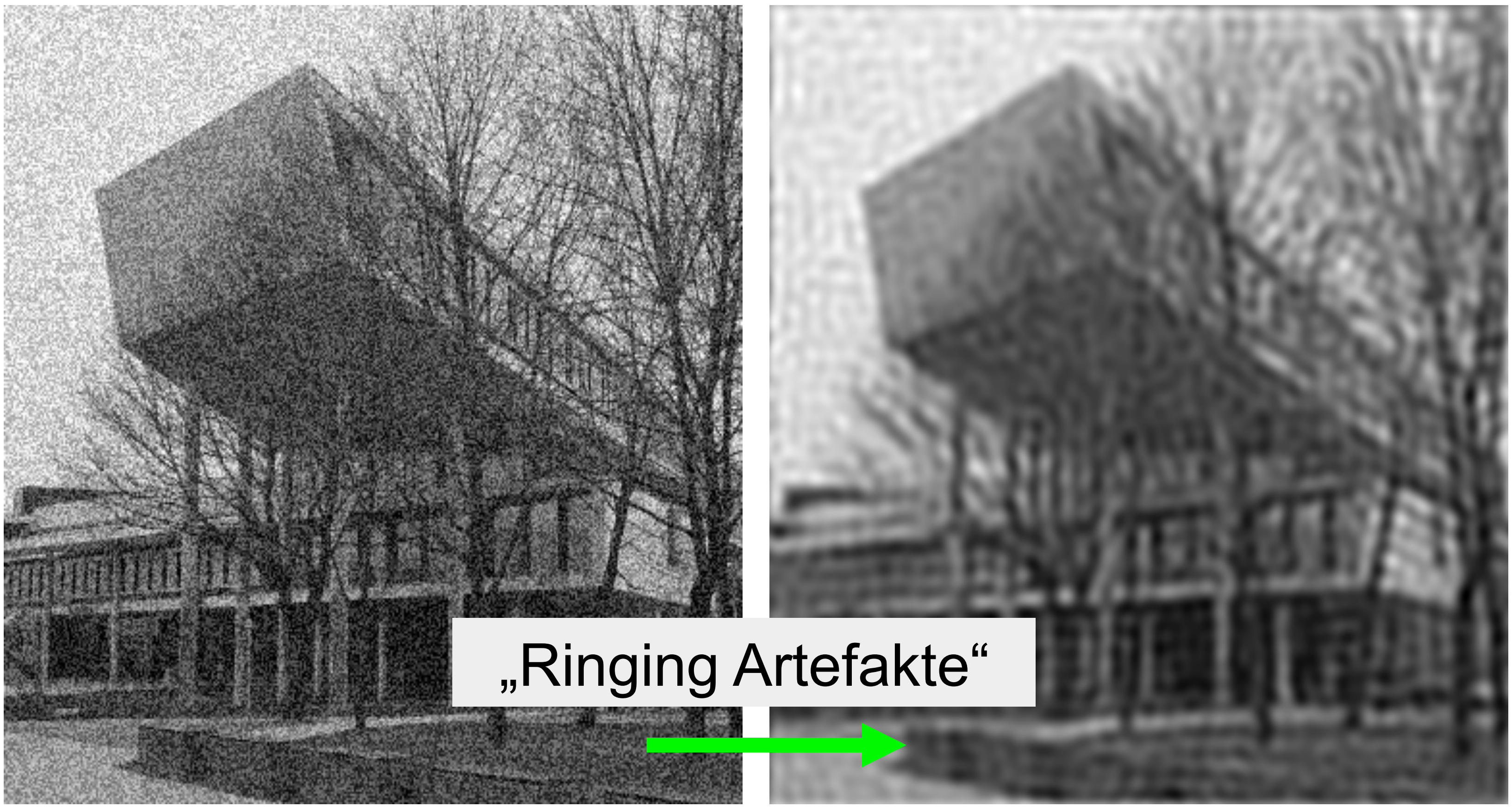
|f(m,n)-g(m,n)|



nach 3x3 Boxcar-Filterung

nach 7x7 Boxcar-Filterung

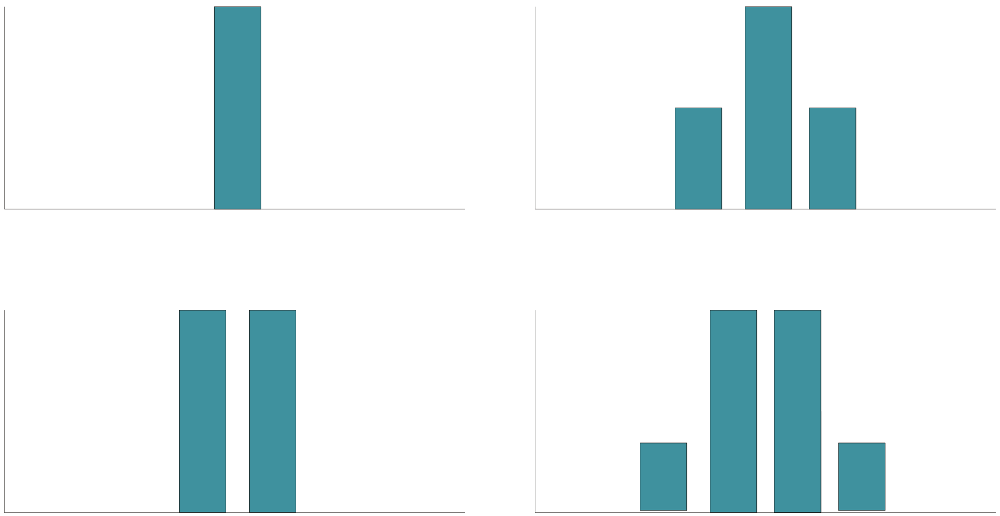
Tiefpassfilter zur Rauschunterdrückung



Binomialfilter

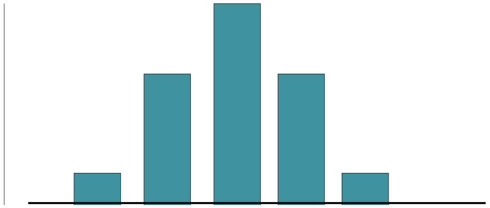
Eindimensionales Binomialfilter B^p = [1 1]*[1 1] *...*[1 1] (p-mal):

$B^0 = 1^{-1} \cdot [1]$
$B^1 = 2^{-1} \cdot [1\ 1]$
$B^2 = 4^{-1} \cdot [1\ 2\ 1]$
$B^3 = 8^{-1} \cdot [1\ 3\ 3\ 1]$
$B^4 = 16^{-1} \cdot [1\ 4\ 6\ 4\ 1]$
...



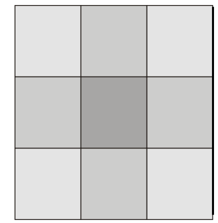
Zweidimensionales Binomialfilter $\mathbf{B}^p = B^p * (B^p)^T$:

$$\mathbf{B}^2 = 4^{-1} \cdot [1\ 2\ 1]^T \cdot 4^{-1} \cdot [1\ 2\ 1] = 16^{-1} \cdot \left\langle \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} \right\rangle$$



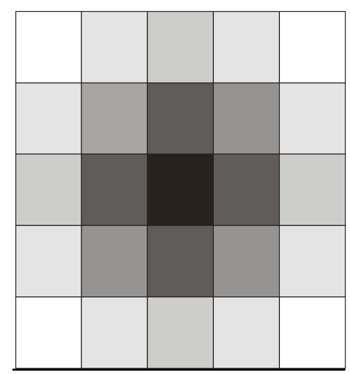
Zweidimensionale Binomialfilter

$$\mathbf{B}^2 = 1/16 \cdot [1\ 2\ 1]^T \cdot [1\ 2\ 1] = 1/16 \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

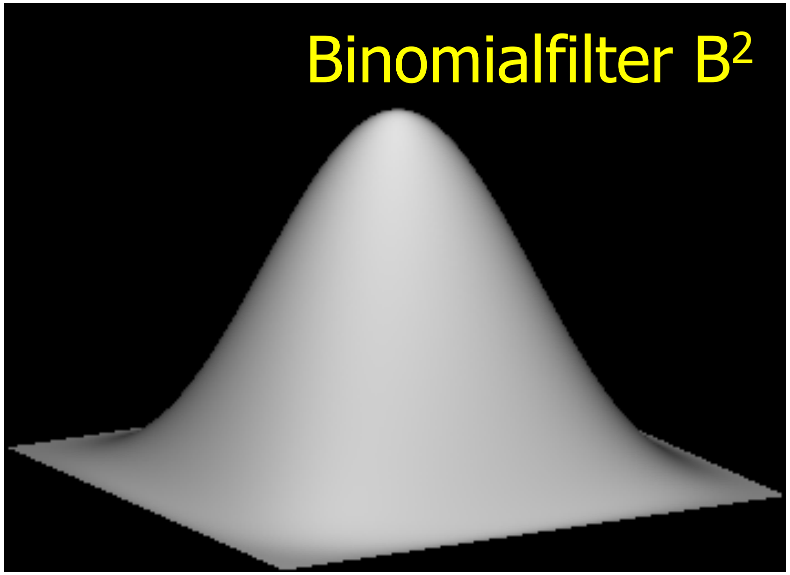


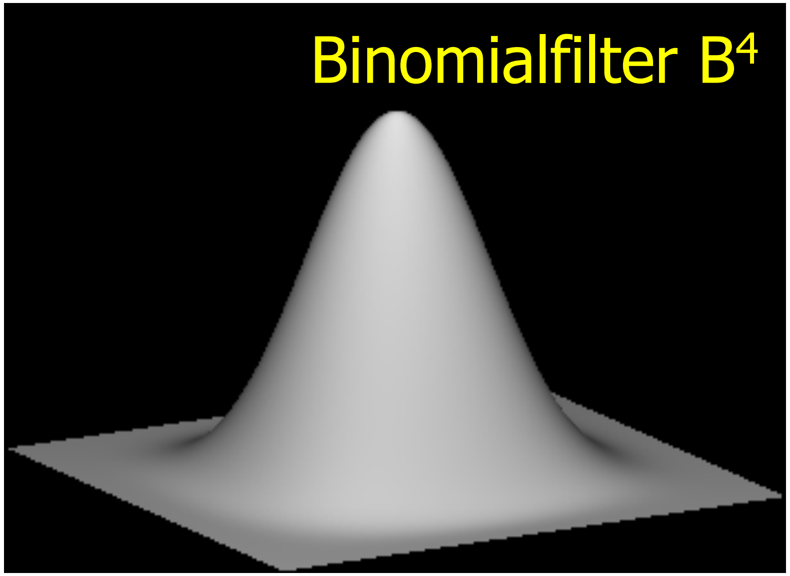
$$\mathbf{B}^3 = 1/64 \cdot [1\ 3\ 3\ 1]^T \cdot [1\ 3\ 3\ 1] = 1/64 \cdot \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{bmatrix}$$

$$\mathbf{B}^4 = 1/256 \cdot \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$



Transferfunktion des Binomialfilters





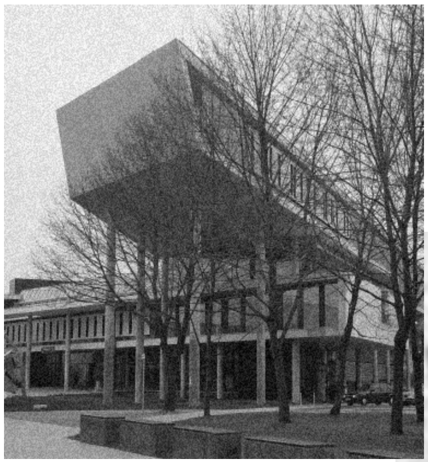
Weniger Artefakte an Kanten sind zu erwarten

Binomialfilter und Gaußfunktion

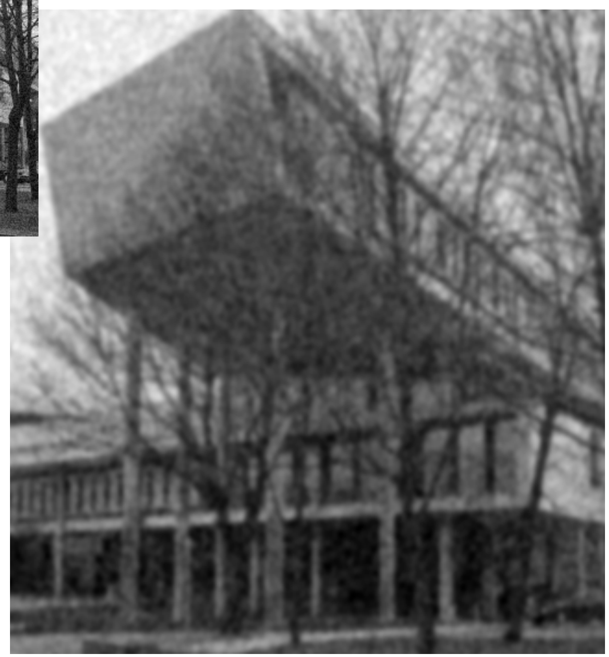
- Für immer größere Filterkerne nähert sich das Binomialfilter der Gaußschen Glockenkurve an

$$G(x,y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right]$$

Filterung mit 2D Gaußfilter



Die Gaußfunktion ist separierbar, so dass die Filterung durch zwei 1D Konvolutionen erfolgen kann

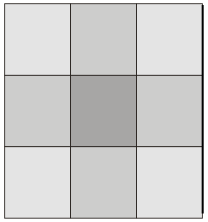


Separierbarkeit der Gaußfunktion

$$G(x,y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] = a\exp\left[-b(x^2+y^2)\right]$$

$$= a\exp\left[-bx^2\right]\exp\left[-by^2\right]$$

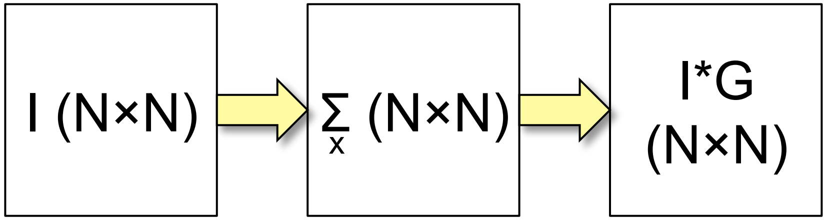
1	2	1
2	4	2
1	2	1



- Konvolution in lokaler Umgebung:

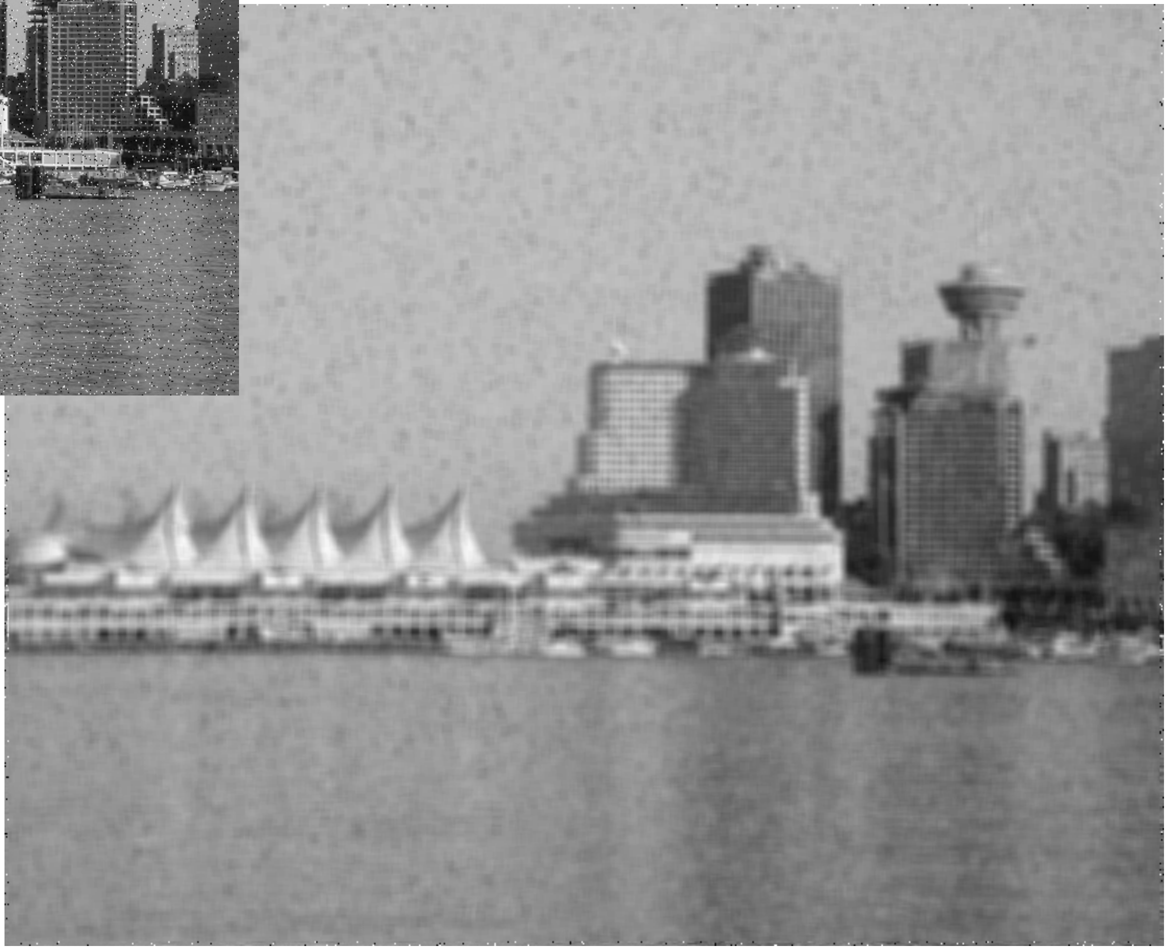
$$\sum_y \sum_x I_{xy} G(x,y) = \sum_y \sum_x I_{xy}\, a\exp\left[-bx^2\right]\exp\left[-by^2\right]$$

$$= a\sum_y \exp\left[-by^2\right] \sum_x I_{xy} \exp\left[-bx^2\right]$$



- Original: N^2K^2 Multiplikationen
- Separiert: $2N^2K$ Multiplikationen

Grenzen



Impulsrauschen (Salt & Pepper Noise) kann durch lineare Filterung nicht entfernt werden

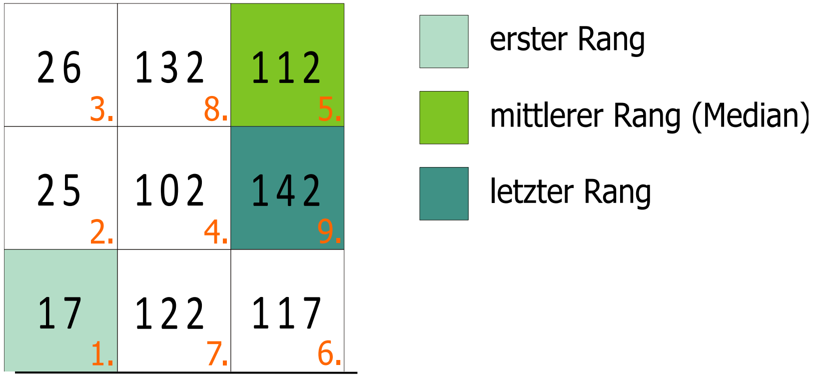
Nichtlineare Filterung

- Rauschen und Kanten haben im Frequenzbereich ähnliche Attribute
- Ist ein nichtlineares Filter denkbar, das für Rauschen und Kanteneigenschaften unterschiedlich sensitiv ist?
  - Rauschen sind räumlich gleichverteilte Grauwertvariationen
  - Grauwertvariationen an Kanten sind nicht räumlich gleichverteilt
  - Filter muss diesen Unterschied berücksichtigen

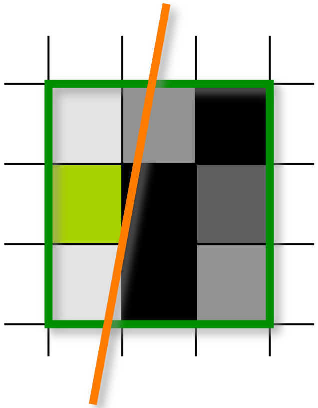
Rangordnungsfilter

- Vorgehen
  – Sortierung der Elemente in einer Filtermaske
  – Auswahl des an einer bestimmten Stelle einsortierten Werts
  – Eintragung des ausgewählten Werts in die zentrale Position
- Eigenschaften
  – Es entstehen keine neuen Werte
  – Filter ist nichtlinear, nicht kommutativ, nicht assoziativ

Gebräuchlichstes Rangordnungsfilter ist das Medianfilter



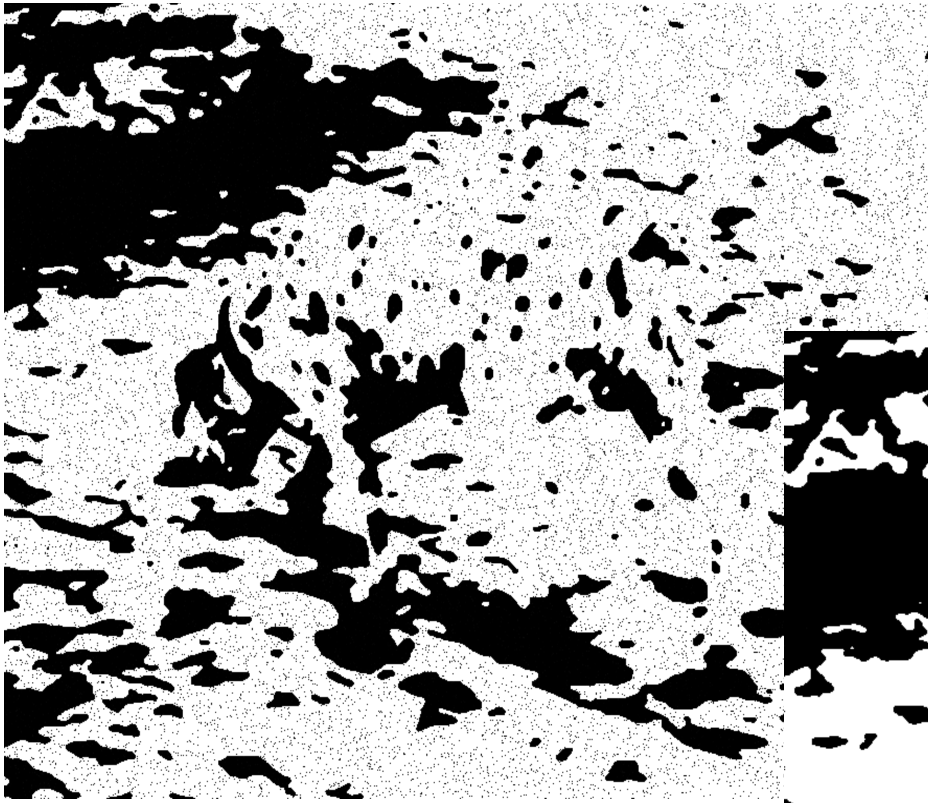
Medianfilter



- Annahmen
  1. Grauwerte auf beiden Seiten der Kante jeweils (nahezu) konstant
  2. Kantensignal größer als das Rauschsignal
  3. Kante im Filterbereich (nahezu) gerade

  k_D (k_H) = sortierte Folge der Pixelwerte der dunkleren/helleren Seite
- Kante verläuft durch Filterbereich
  - wegen (2) : alle k_D vor k_H
  - wegen (3): mehr k_D, falls Zentrum in k_D (und umgekehrt)
  - also Median von der Seite, zu der Pixel im Zentrum gehört(kantenerhaltend), Artefakte bei Ecken
- Keine Kante im Filterbereich
  - Median nähert sich dem Erwartungswert mit Anzahl der Stichproben (rauschunterdrückend)

Medianfilter

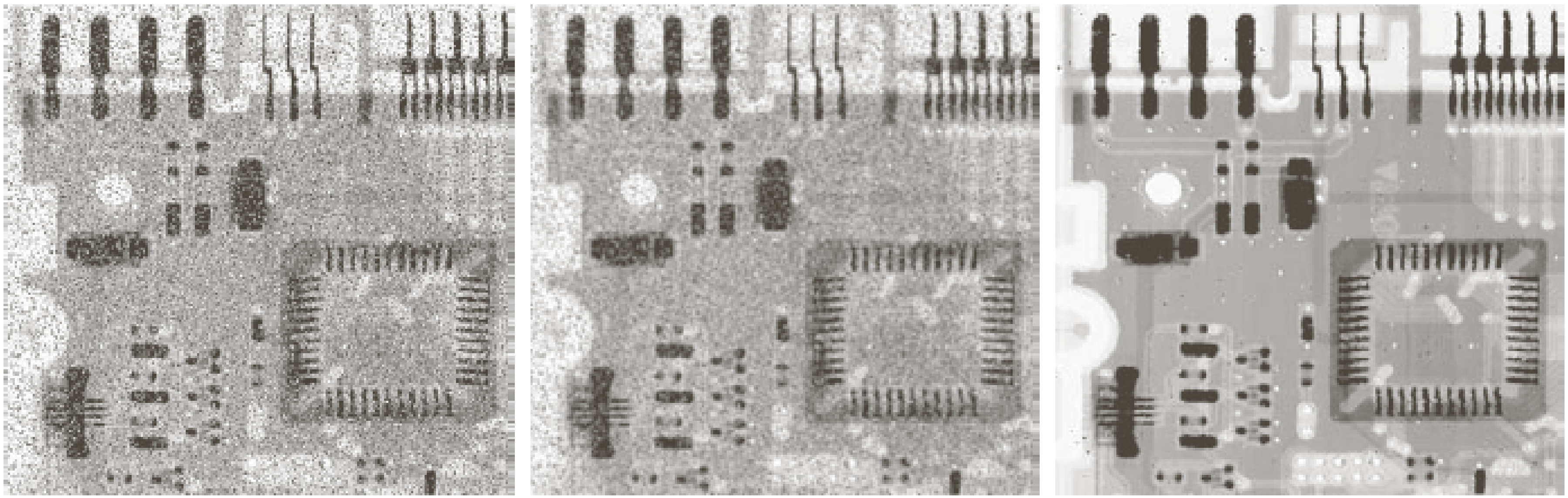


Median zur Entfernung von Pepper-Noise in einer 3x3 Umgebung

Medianfilter

- Durch Medianfilterung (rechts) kann Impulsrauschen im Gegensatz zur Mittelwertfilterung (Mitte) beseitigt werden.

Medianfilter

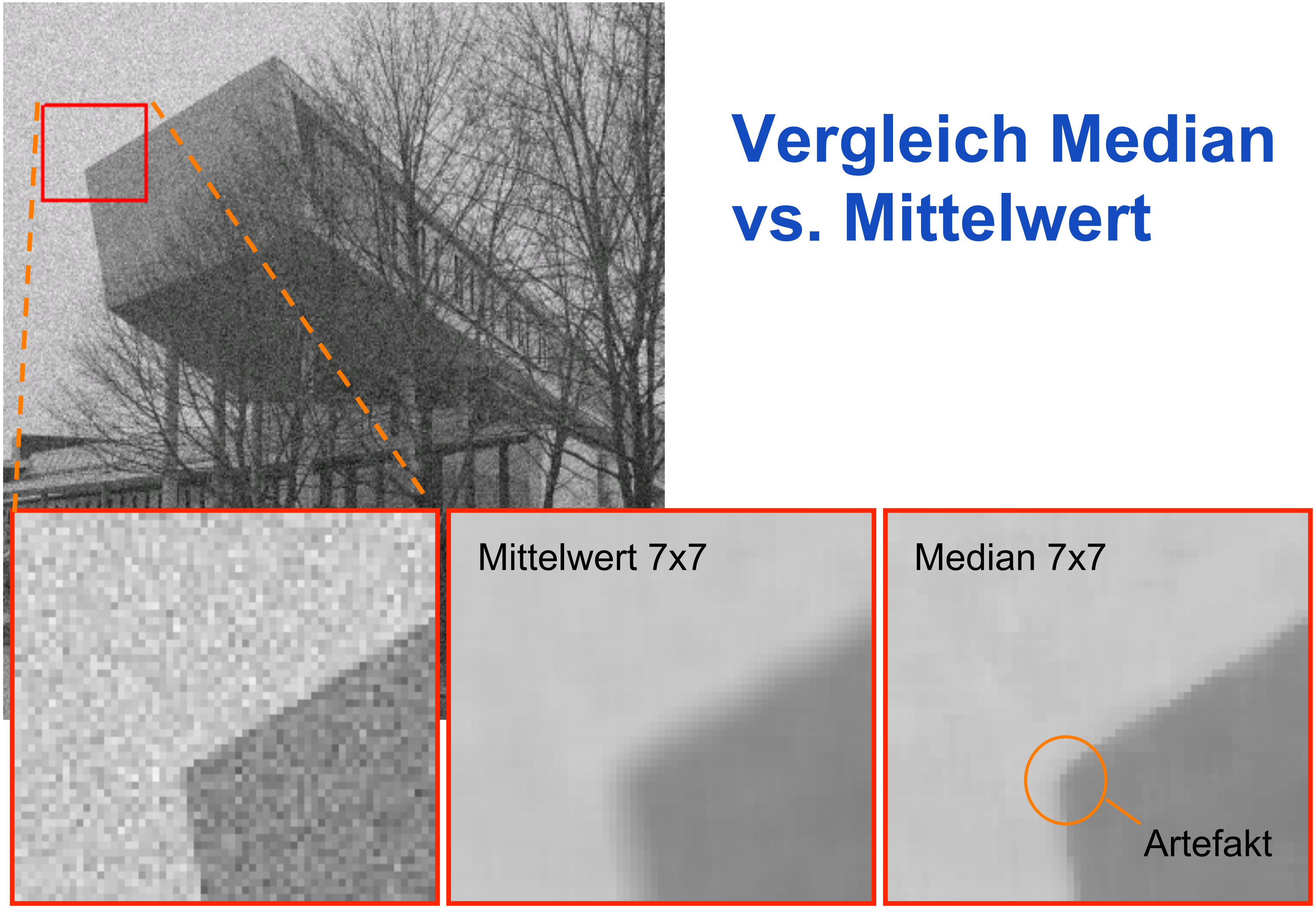


a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Vergleich Median vs. Mittelwert



Adaptives Medianfilter

- Kann mit dichterem Impulsrauschen umgehen
- Verschont Zwischenpunkte
- Parameter

  Variable (quadratische) Größe S_{xy}

  z_{min} = minimaler Grauwert in S_{xy}

  z_{max} = maximaler Grauwert in S_{xy}

  z_{med} = median Grauwert in S_{xy}

  z_{xy} = Grauwert an (x,y)

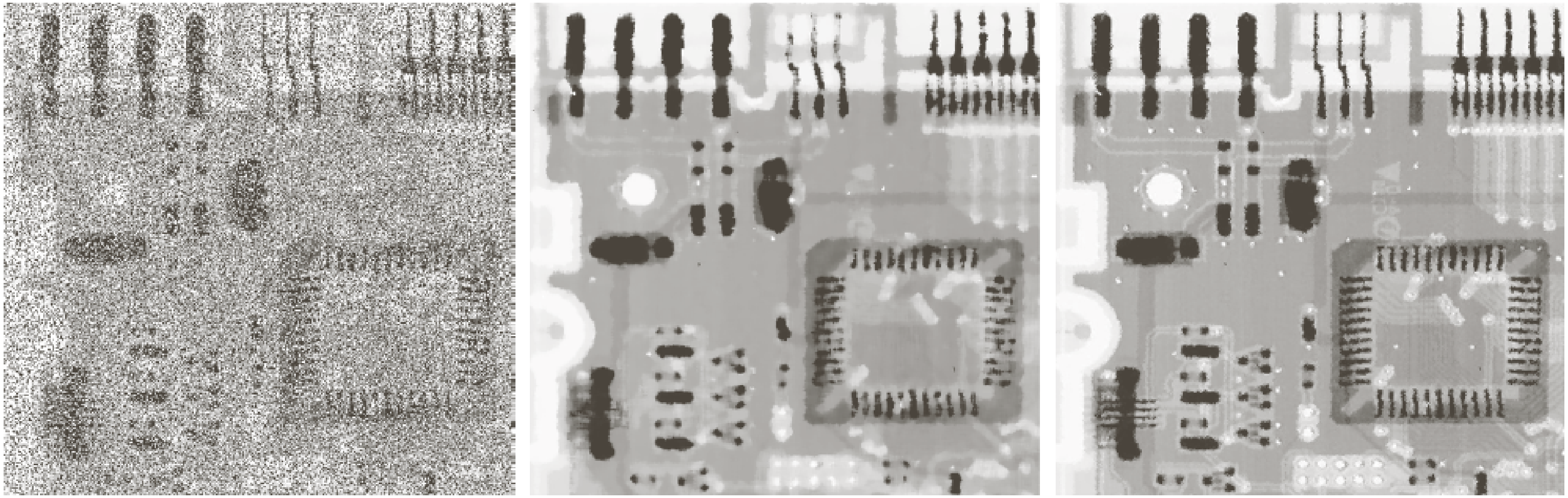
  S_{max} = Maximalgröße von S_{xy}

  S = aktuelle Größe von S_{xy}

  a = 0..1, b = 0..5

```
z_med = 0
while S ≤ S_max:
    z_min, z_med, z_max, z_xy =
            medFilter(img, x, y, S)
    if z_min + a < z_med < z_max - a:
        if z_min + b < z_xy < z_max - b:
            return z_xy
        else:
            return z_med
    S = S + 1
return z_med
```

Adaptives Medianfilter



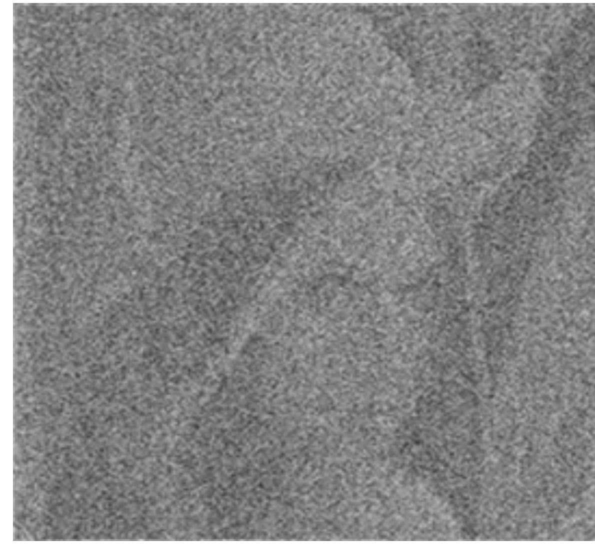
a b c

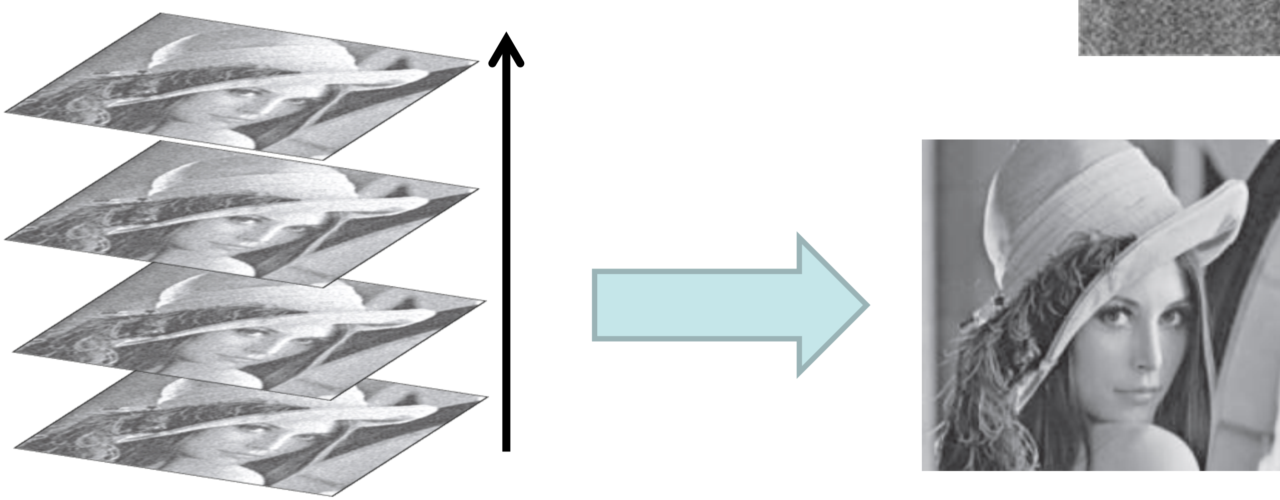
FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Nicht-lokale Mittelwertbildung

- Mehrere verrauschte Bilder einer statischen Szene
- Rauschen hat Mittelwert 0
- Mittelwert bilden

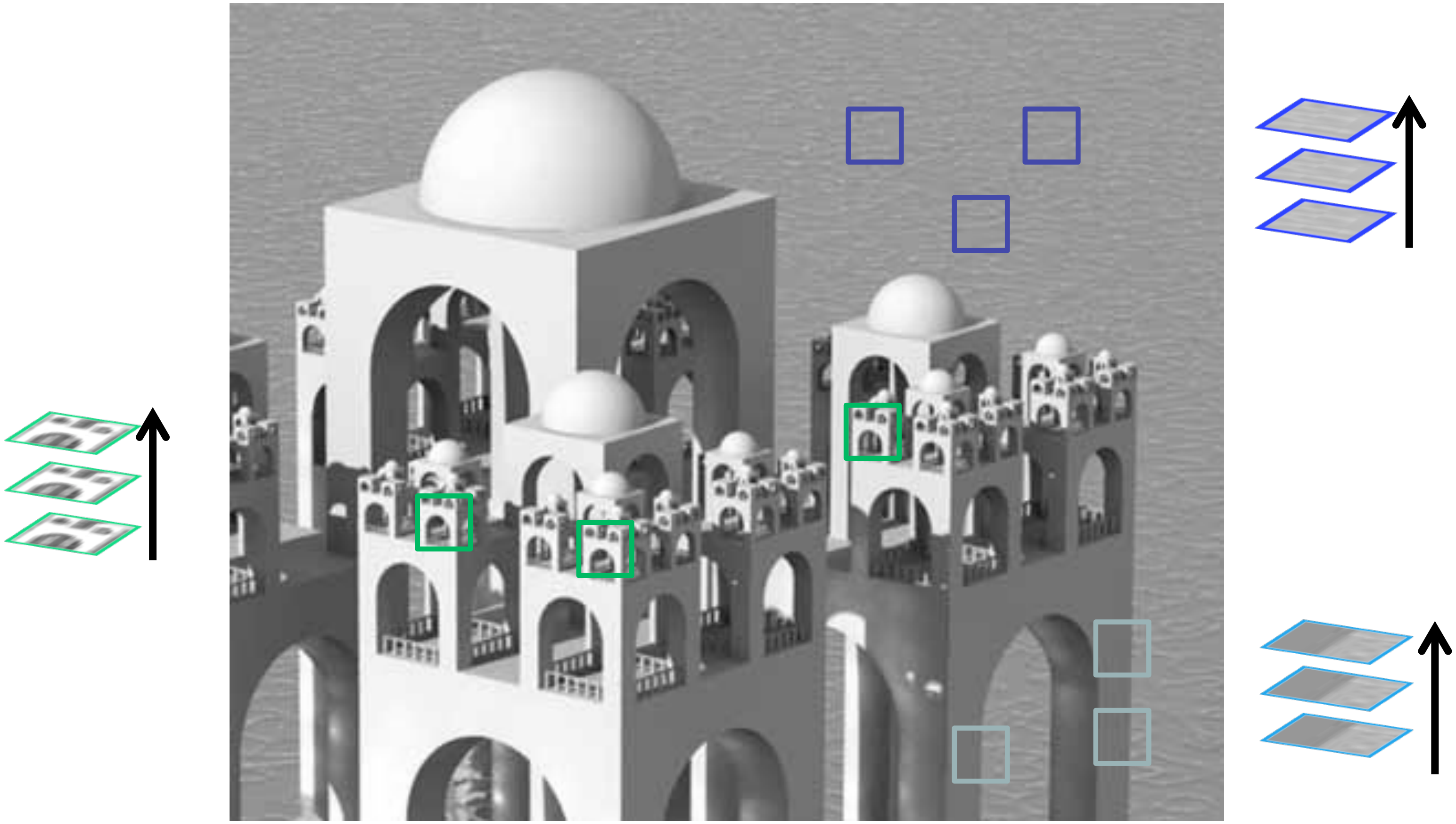




Figures from: Shahar Kovalsky, Alon Faktor: A Tour of Image Denoising, Slides

Nicht-lokale Mittelwertbildung

- Redundanz in natürlichen Bildern ausnutzen



Figures from: Shahar Kovalsky, Alon Faktor: A Tour of Image Denoising, Slides

Buades, Coll, Morel: Gewichteter Mittelwert durch Selbst-Ähnlichkeit

- Verrauschtes Bild:

$$v = \left\{ v(i) \mid i \in I \right\}, \quad v(i) = u(i) + n(i)$$

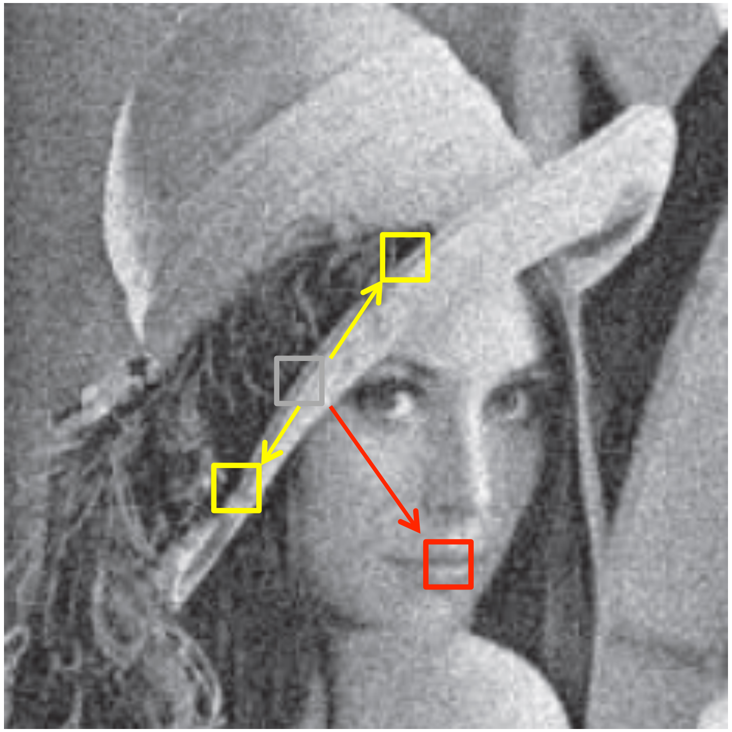
- Nicht-lokaler Mittelwert:

$$NL[v](i) = \sum_{j \in I} w(i,j) v(j)$$

- Gewichtsfunktion

$$w(i,j) = \frac{1}{Z(i)} \exp\left(- \frac{\left\| v(N_i) - v(N_j) \right\|_{2,a}^2}{h^2} \right)$$

$$Z(i) = \sum_j \exp\left(- \frac{\left\| v(N_i) - v(N_j) \right\|_{2,a}^2}{h^2} \right)$$



Figures from: Shahar Kovalsky, Alon Faktor:
A Tour of Image Denoising, Slides

KANTENMERKMALE DURCH LINEARE FILTERUNG

Erste und zweite Ableitung von Bildern

- Erste Ableitung:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- Zweite Ableitung:

$$\frac{\partial^2 f}{\partial^2 x} = (f(x+1) - f(x))$$
$$-(f(x) - f(x-1))$$
$$= f(x+1) + f(x-1) - 2f(x)$$

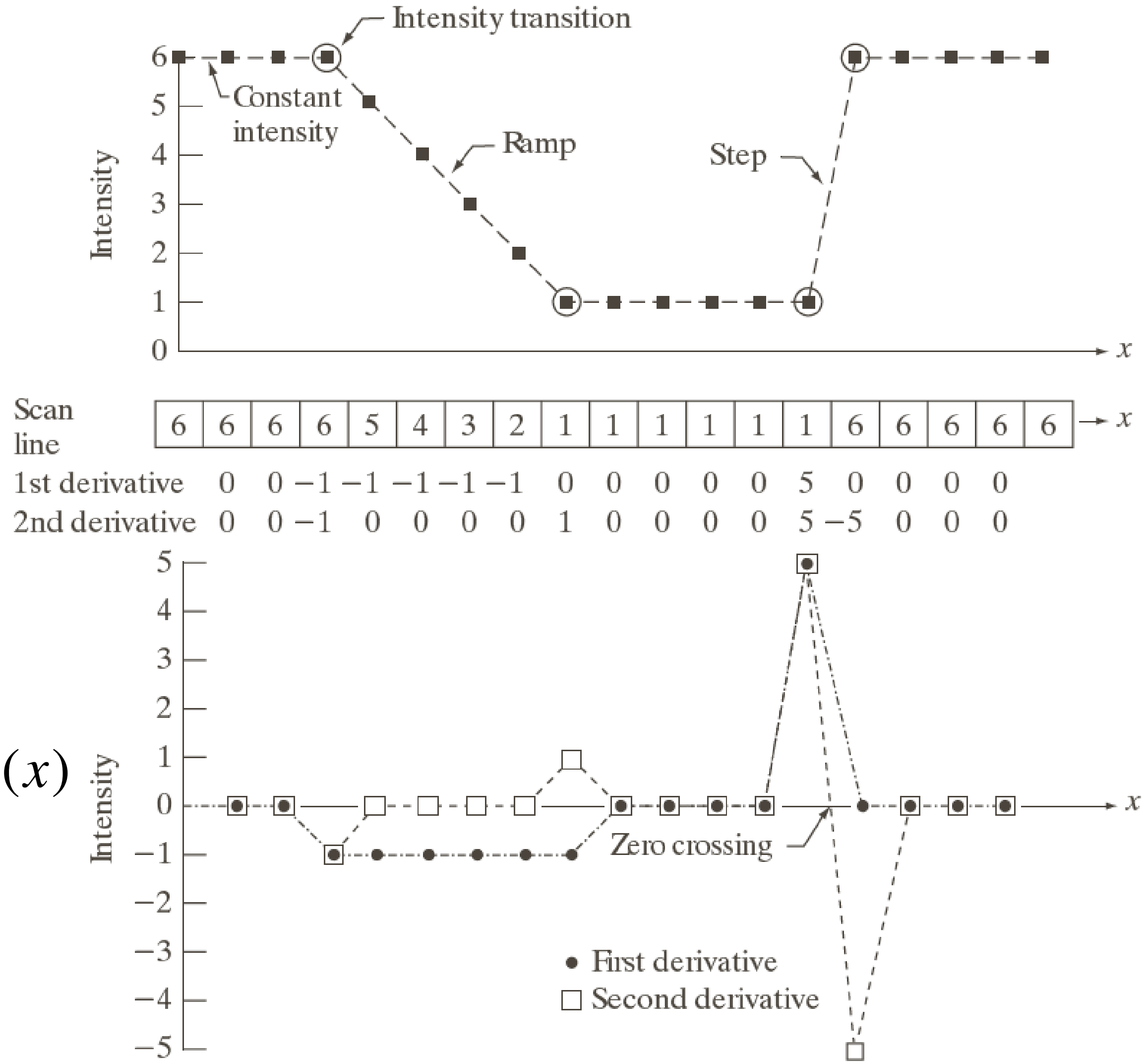


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

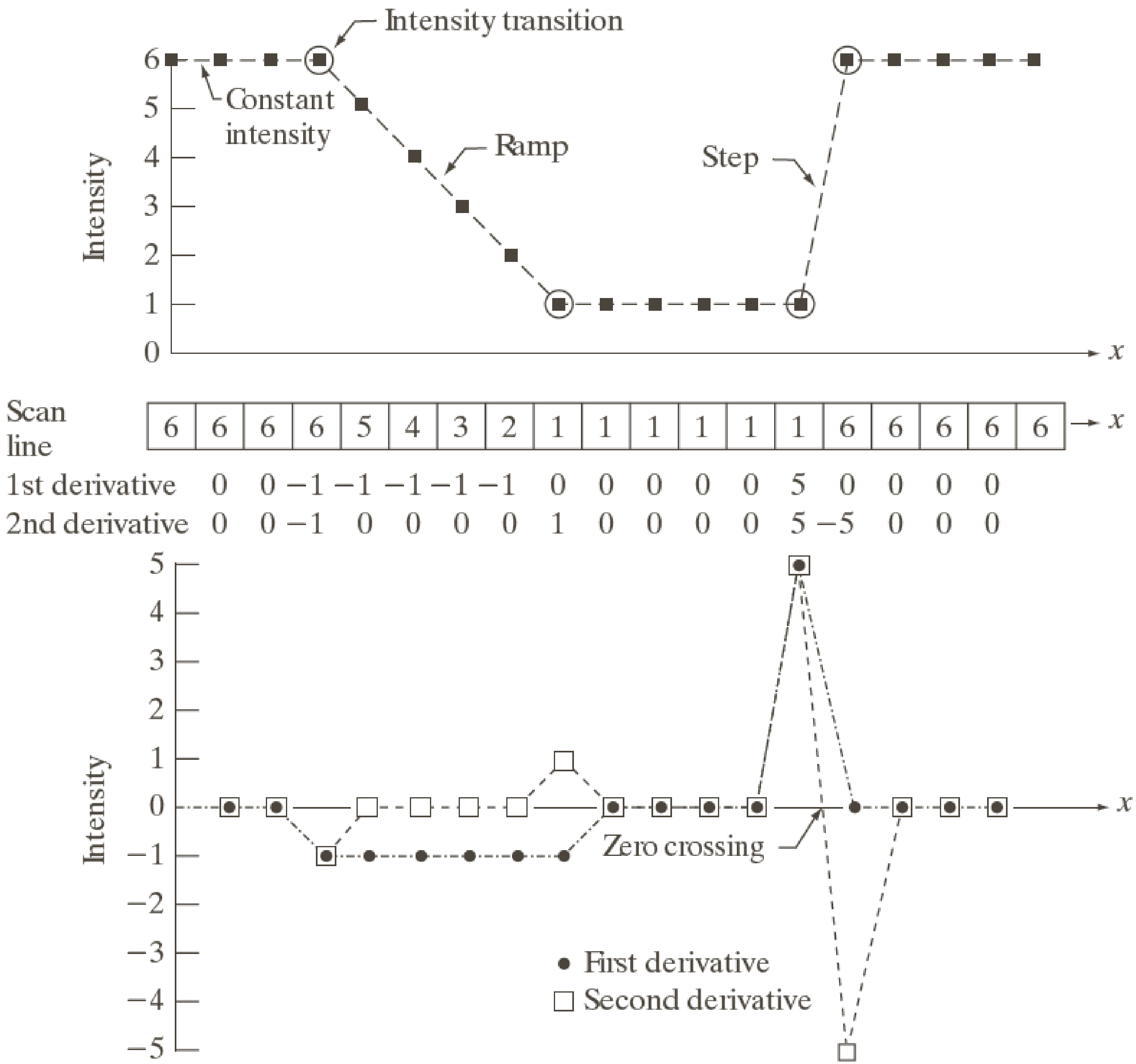


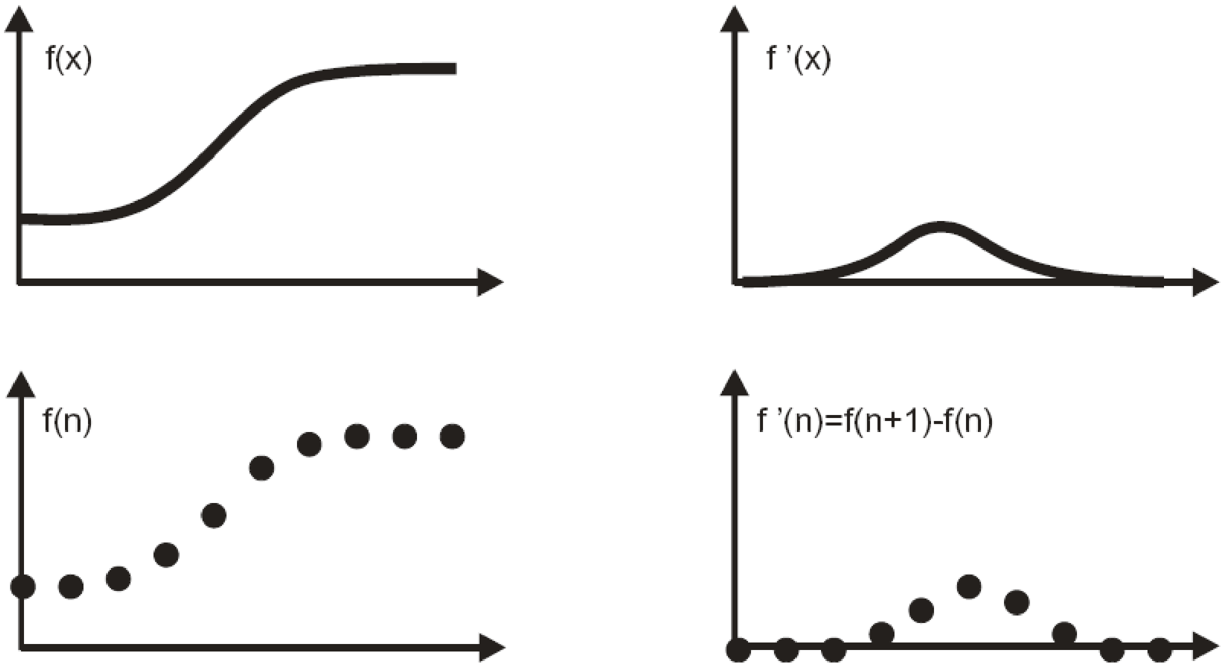
Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Kanten

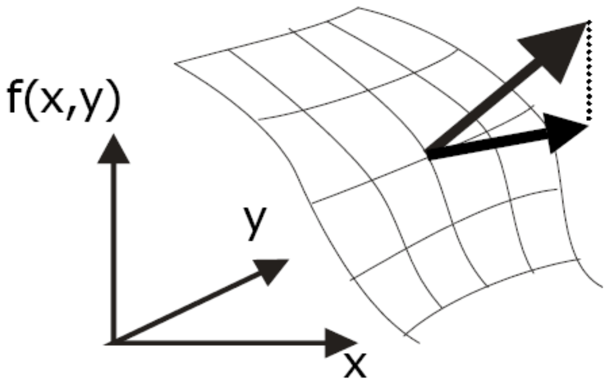
Die meisten Bilder weisen wenige Linien oder Punkte auf, aber Kanten können in fast jedem Bild gefunden werden.

1D-Kanten

- Die Stärke einer Kante hängt von der Steigung der Funktion ab: Betrag der ersten Ableitung bestimmen
- Für diskrete Funktionen: Ableitung wird durch Differenz angenähert



Kanten im 2D-Raum: Gradienten



Richtung: Richtung der größten Steigung.

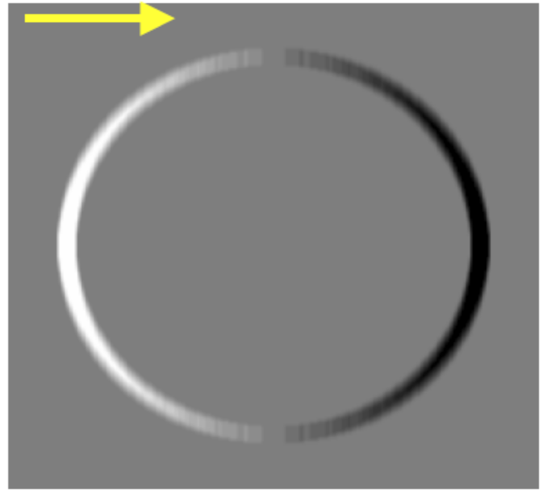
Länge: Stärke der stärksten Steigung.

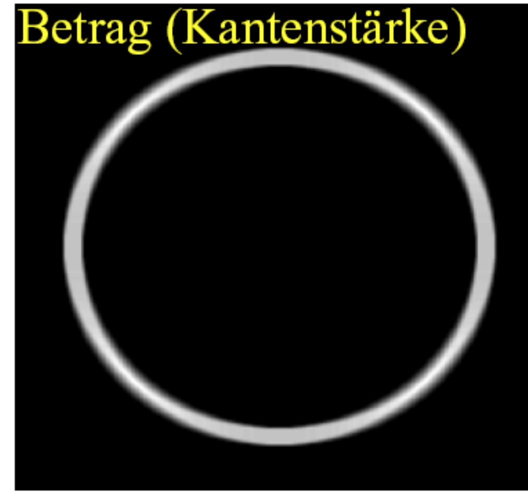
- Gradient im kontinuierlichen Raum (x,y): Vektor der partiellen Ableitungen der Bildfunktion in x- und y-Richtung: (f(x,y)) = ($\partial f/\partial x$ $\partial f/\partial y$)
- Approximation des Gradienten: Differential wird durch Differenz approximiert:

$$\vec{G}(f)(m,n) \approx [G_x(m,n) \quad G_y(m,n)] = [f(m,n)\text{-}f(m\text{-}1,n) \quad f(m,n)\text{-}f(m,n\text{-}1)]$$

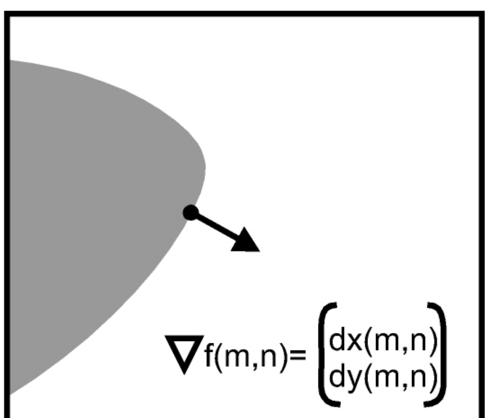
- Die Länge des Gradienten ist sein Betrag |G(f)| oder näherungsweise $|Gx|+|Gy|$.

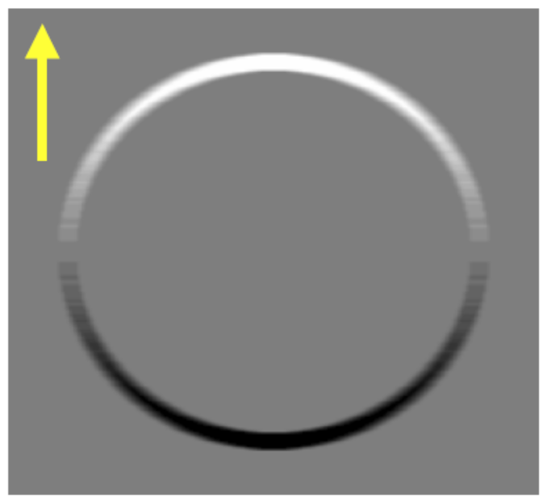
Elemente des Gradienten

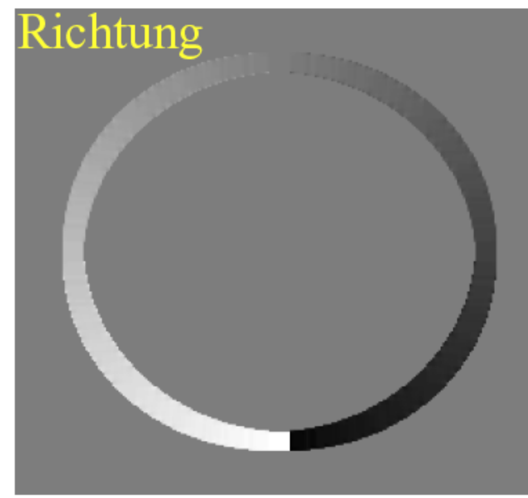




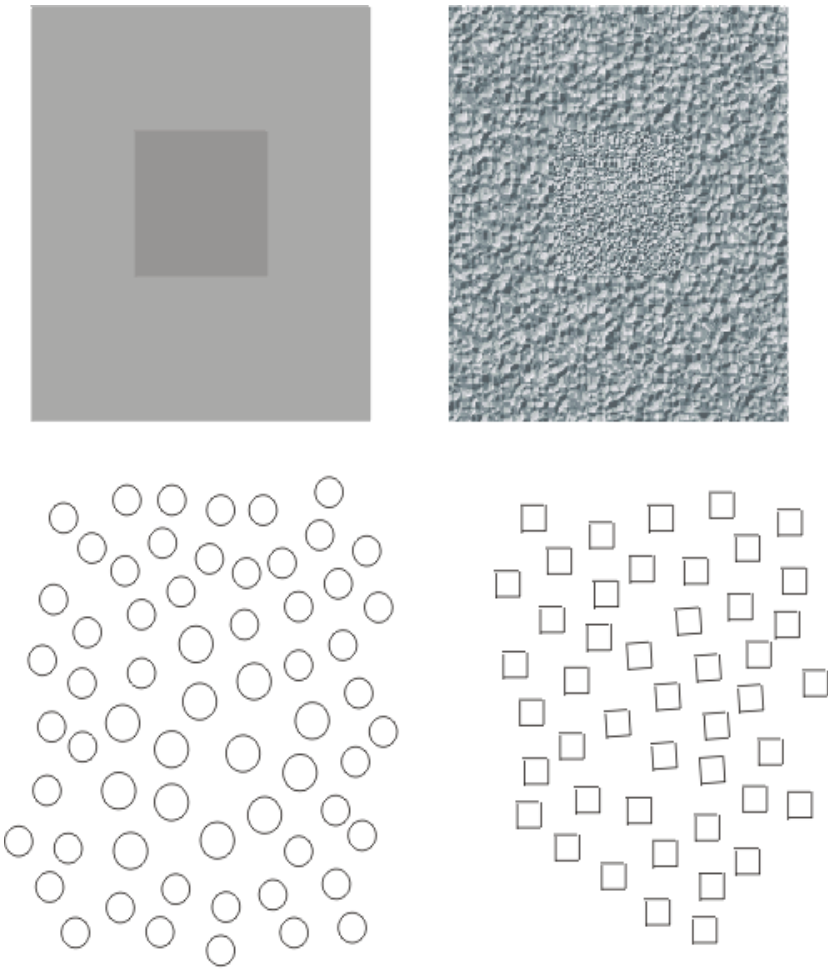
Betrag: sqrt($G_x^2 + G_y^2$)
Richtung: $\tan^{-1}(G_y / G_x)$







Was sind Kanten?



Diskontinuitäten zwischen Regionen.

Problem:

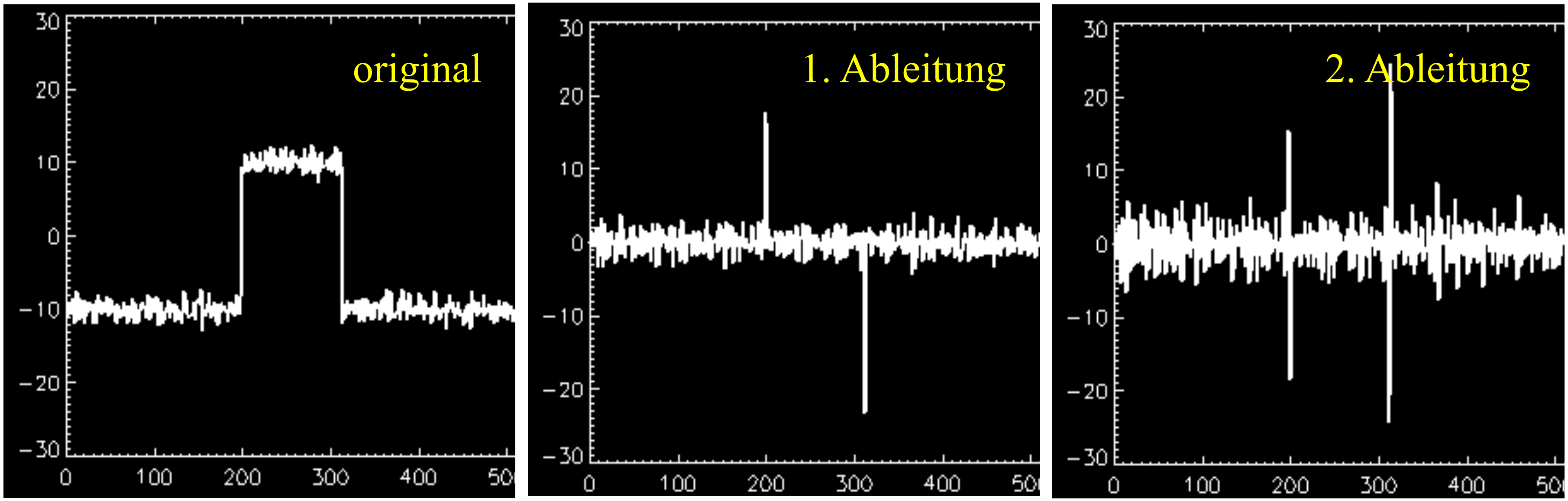
Die Region selbst und damit ihr Homogenitätskriterium muss bekannt sein, um Diskontinuität definieren zu können.

Homogenität einer Funktion:

- Grauwerte
- Textur
- Form

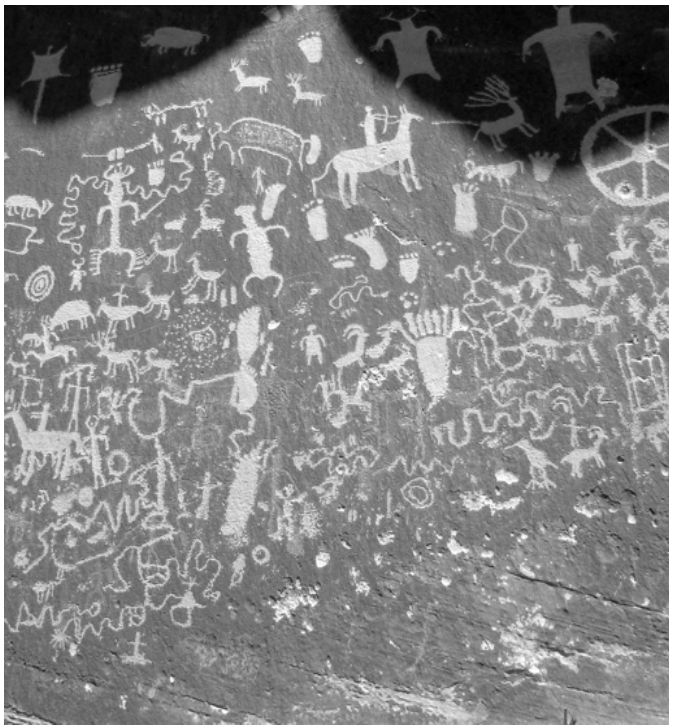
Kanten im Ortsraum

Kanten und Rauschen haben ähnliche Charakteristika im Frequenzraum → Kantendetektor verstärkt Rauschen.

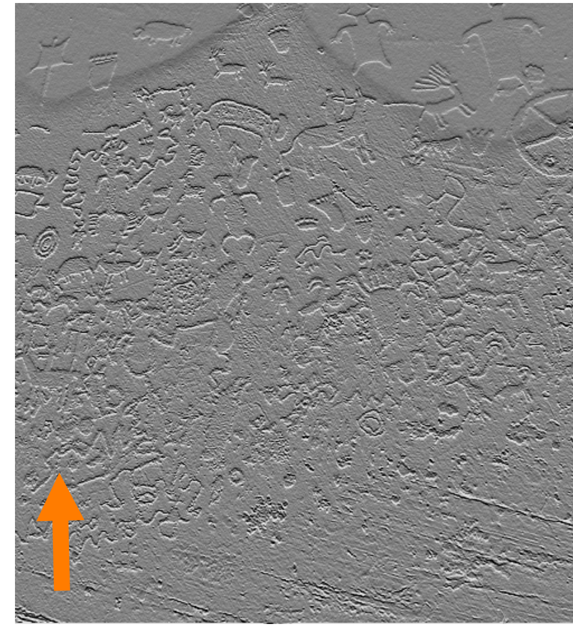


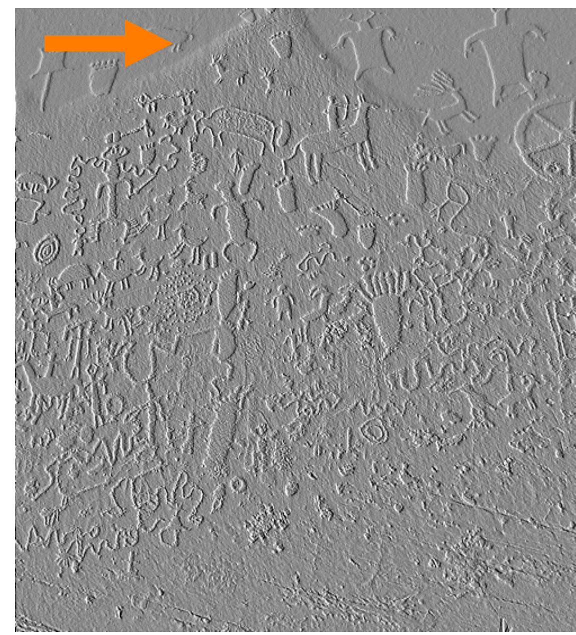
Kanten im 2-D Raum

Differenzbildung in *n*-Richtung



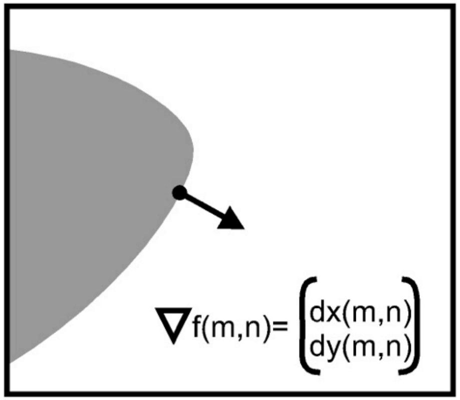
Pseudo-3D
Eindruck:





Differenzbildung in *m*-Richtung

Gradientenschätzung über Konvolution



- Gradient im *N*-dimensionalen Raum ist ein *N*-dimensionaler Vektor aus *N* partiellen Ableitungen.
- Jede partielle Ableitung kann durch eine Differenz abgeschätzt werden, die durch Konvolution berechnet werden kann.
- Beispiele:

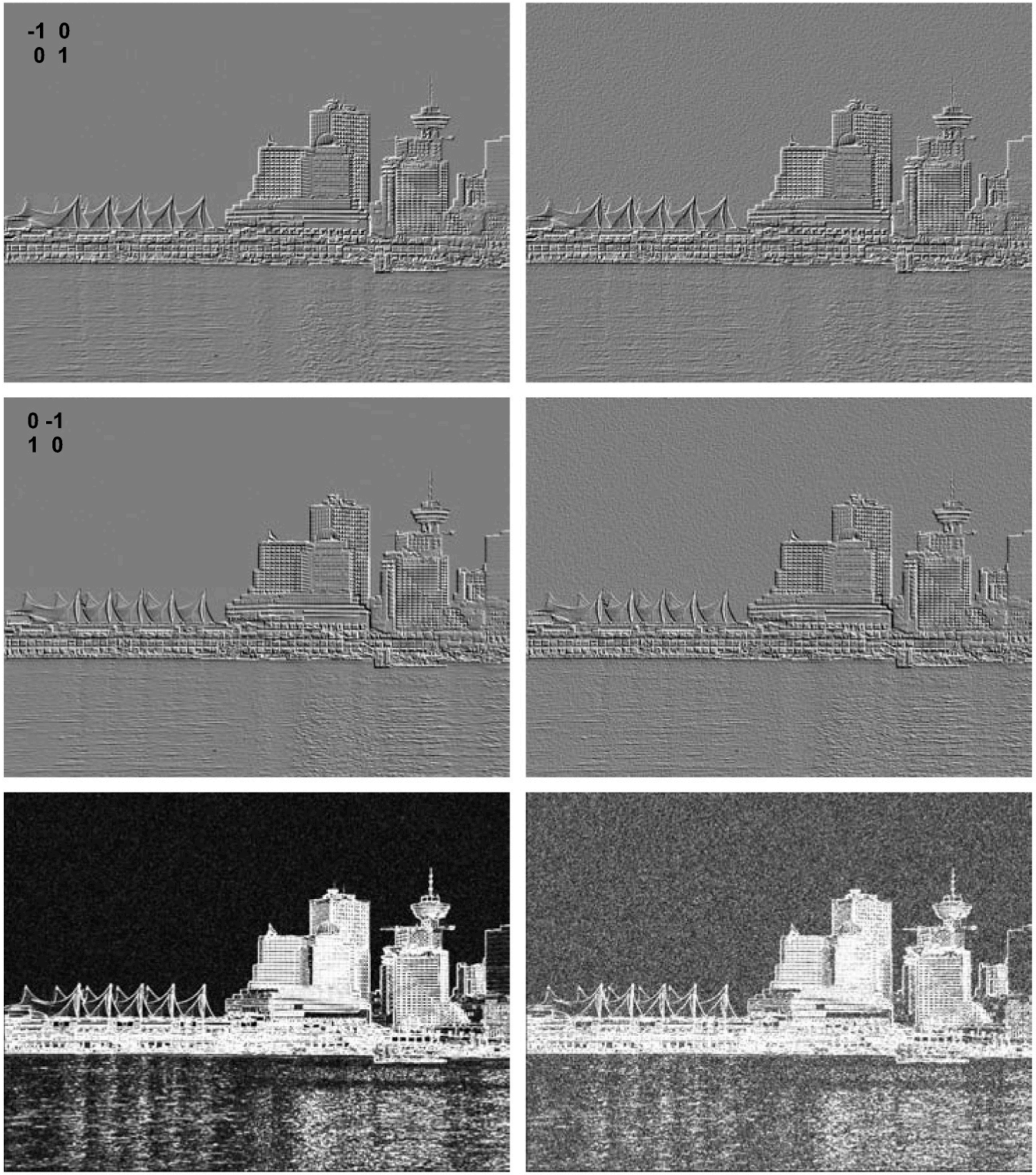
$$\vec{G}(m,n) = \begin{pmatrix} G_x(m,n) \\ G_y(m,n) \end{pmatrix}, \quad G_x(m,n) \approx [f * g_x](m,n), \quad G_y(m,n) \approx [f * g_y](m,n).$$

$$g_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad g_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \qquad g_{R1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad g_{R2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$ (Robert's Gradient)

$$g_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad g_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Robert’s Gradient

Die beiden Komponenten des Robert’s Gradient sowie der Betrag des Gradienten (unten) auf einem rauscharmen (links) und einem verrauschten (rechts) Bild



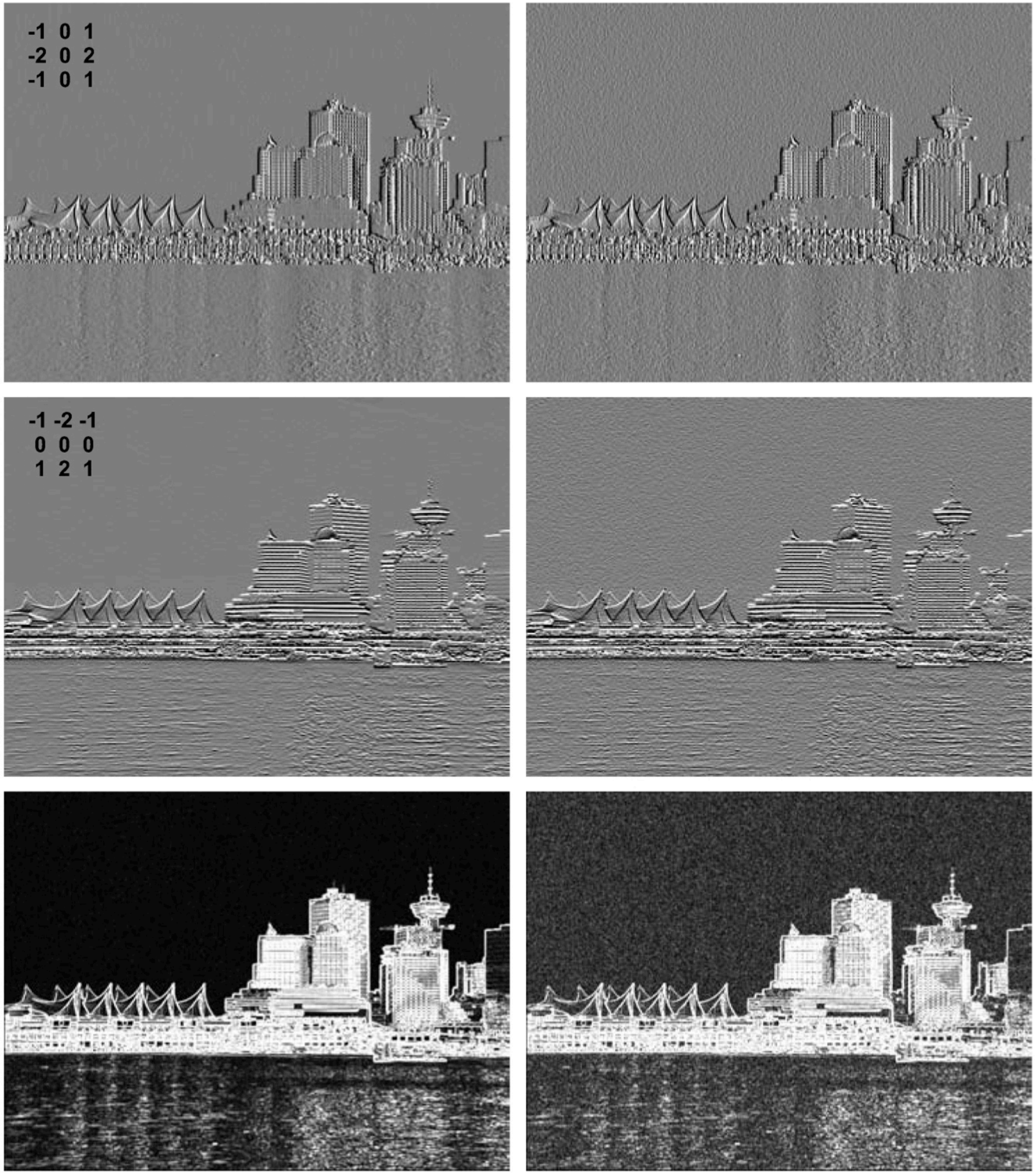
Rauschen und Kanten

Hintergrundartefakte (Rauschen) werden verstärkt.

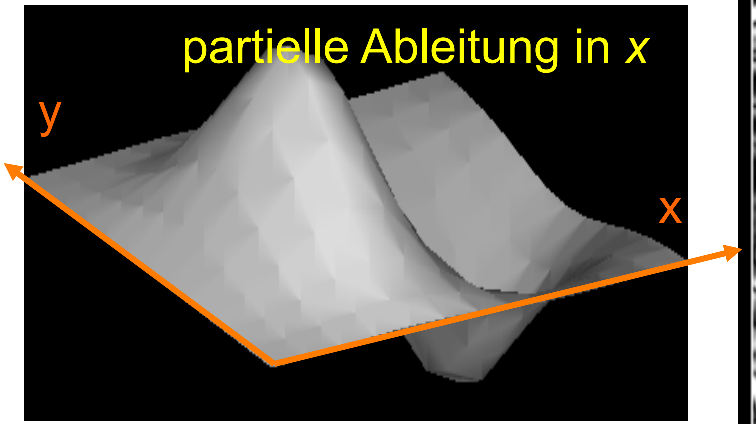
Sobel Operator

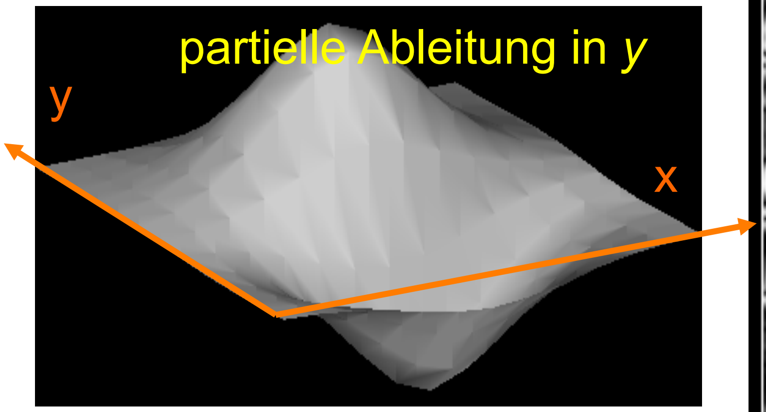
$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad \text{und} \quad \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

Faltungskern kombiniert Glättung und Differenzierung in einem Operator



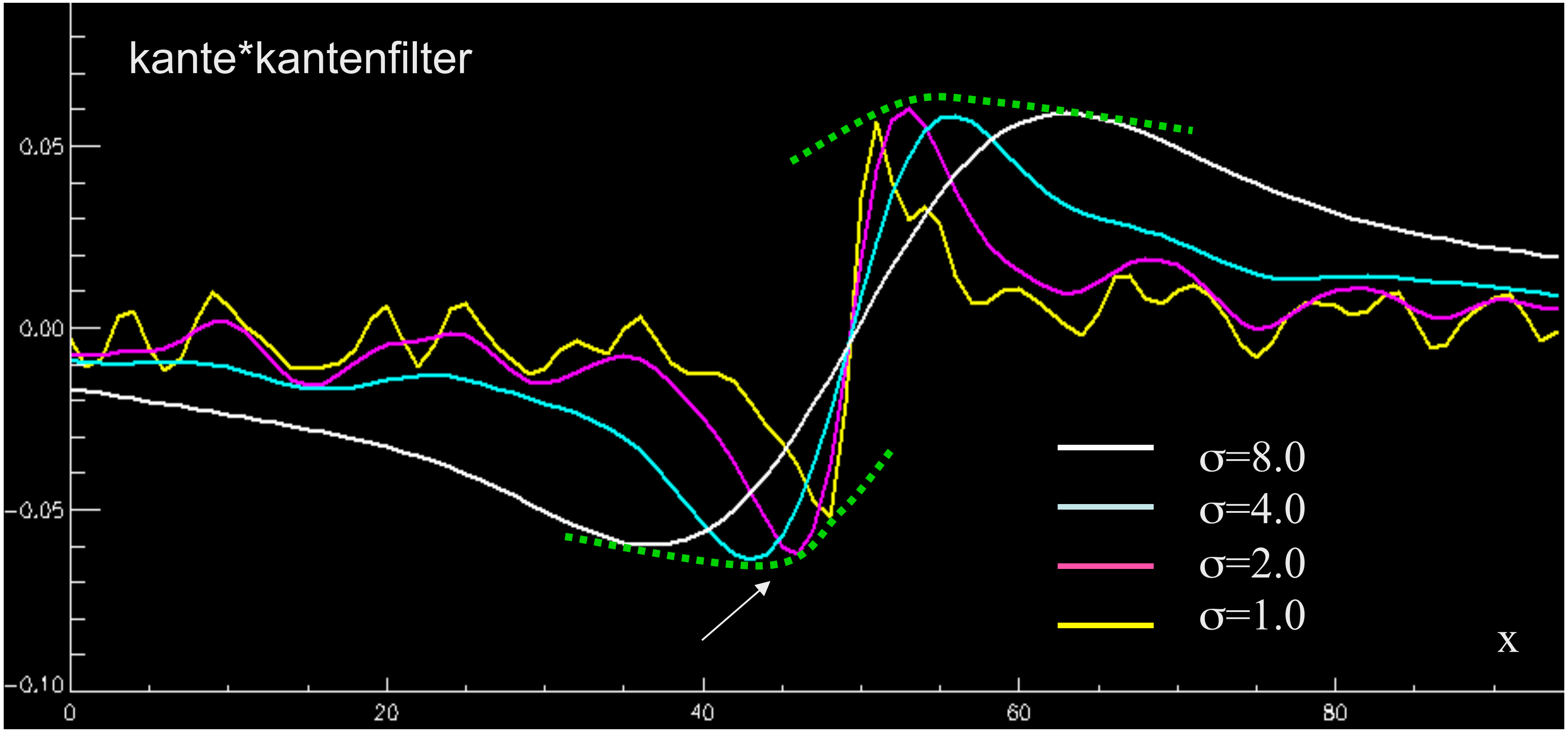
Ableitungen der Gaußfunktion





Sobel vs. Gauß

Filterresultat



Kompassfilter

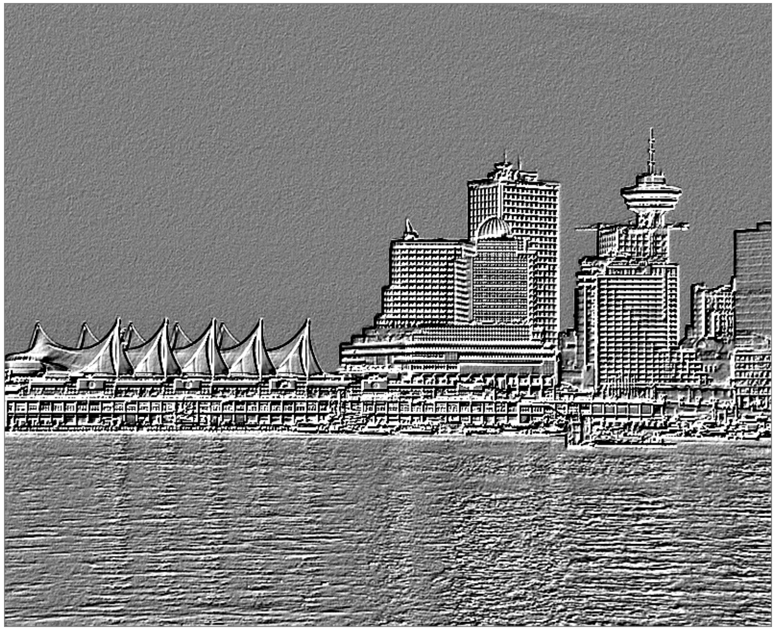
- Dienen der Hervorhebung von Kanten in einer bestimmten Richtung.
- Prewitt-Operator

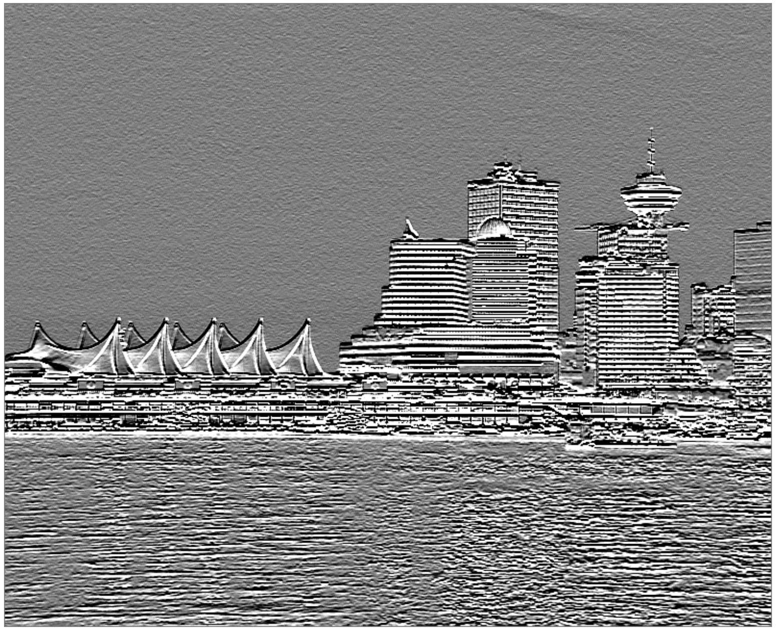
$$h_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \quad h_2 = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}, \quad h_3 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \ldots \quad h_8 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

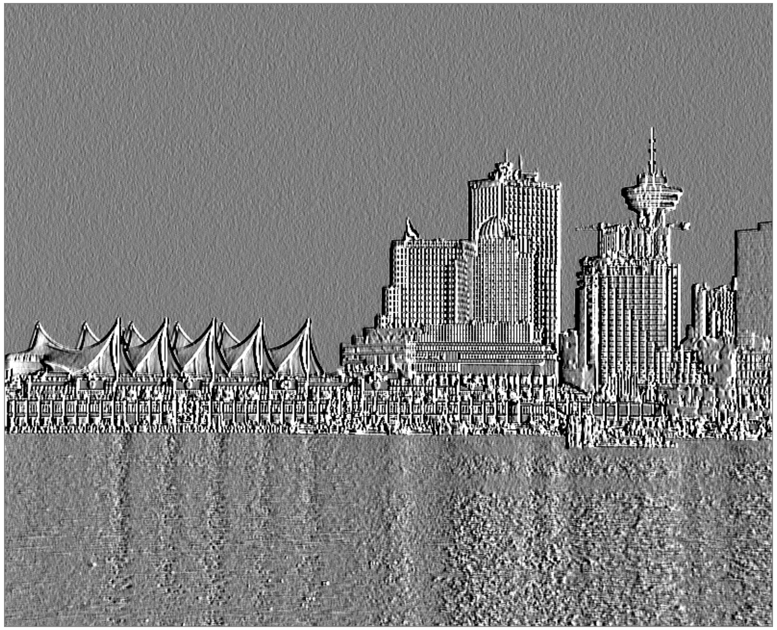
- Kirsch-Operator

$$h_1 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{pmatrix}, \quad h_2 = \begin{pmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{pmatrix}, \quad h_3 = \begin{pmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{pmatrix} \ldots \quad h_8 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{pmatrix}$$

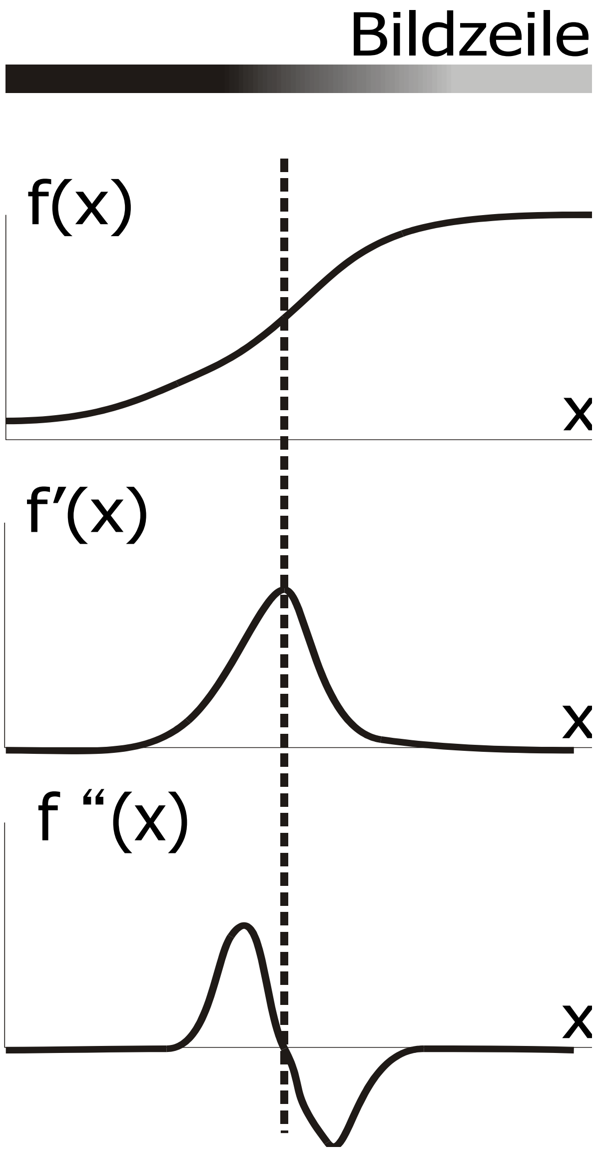
Kirschoperator







Wo ist die Regionengrenze?



Vorzeichenwechsel ist leichter zu erkennen, als ein Minimum oder Maximum.

- ➔ Gradient (Länge) als Kennzeichen für die Wichtigkeit einer Kante
- ➔ zweite Ableitung für den Ort der Kante (Nulldurchgang)

Operatoren zur Berechnung der zweiten Ableitung:

- Laplace Filter
- Marr-Hildreth Filter (LoG Filter, Mexican Hat)
- DoG (Difference of Gaussians)

Laplace Funktion

- Summe der partiellen zweiten Ableitungen: $\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$
- Nulldurchgänge der Laplacefunktion produzieren zusammenhängende Kurven entlang von Bildkanten
- Approximation durch Kombination einer doppelten Differenzbildung in *x*- und *y*-Richtung
  - z.B. Faltung eines Differenzoperators [-1 1] mit sich selbst: [-1 2 -1]
  - ergibt für [-1 1]*[-1 1] + $[-1\ 1]^T$*$[-1\ 1]^T$:

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

  - Summe **aller** partiellen Ableitungen:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} + \frac{\partial^2 f(x,y)}{\partial x \partial y} + \frac{\partial^2 f(x,y)}{\partial y \partial x} : \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Laplace-Operator

- Nulldurchgänge des Laplace-Operators. Dunkle Pixel bezeichnen negative, helle Pixel bezeichnen positive Werte.

Schärfen eines Bildes mit Lapace-Operator

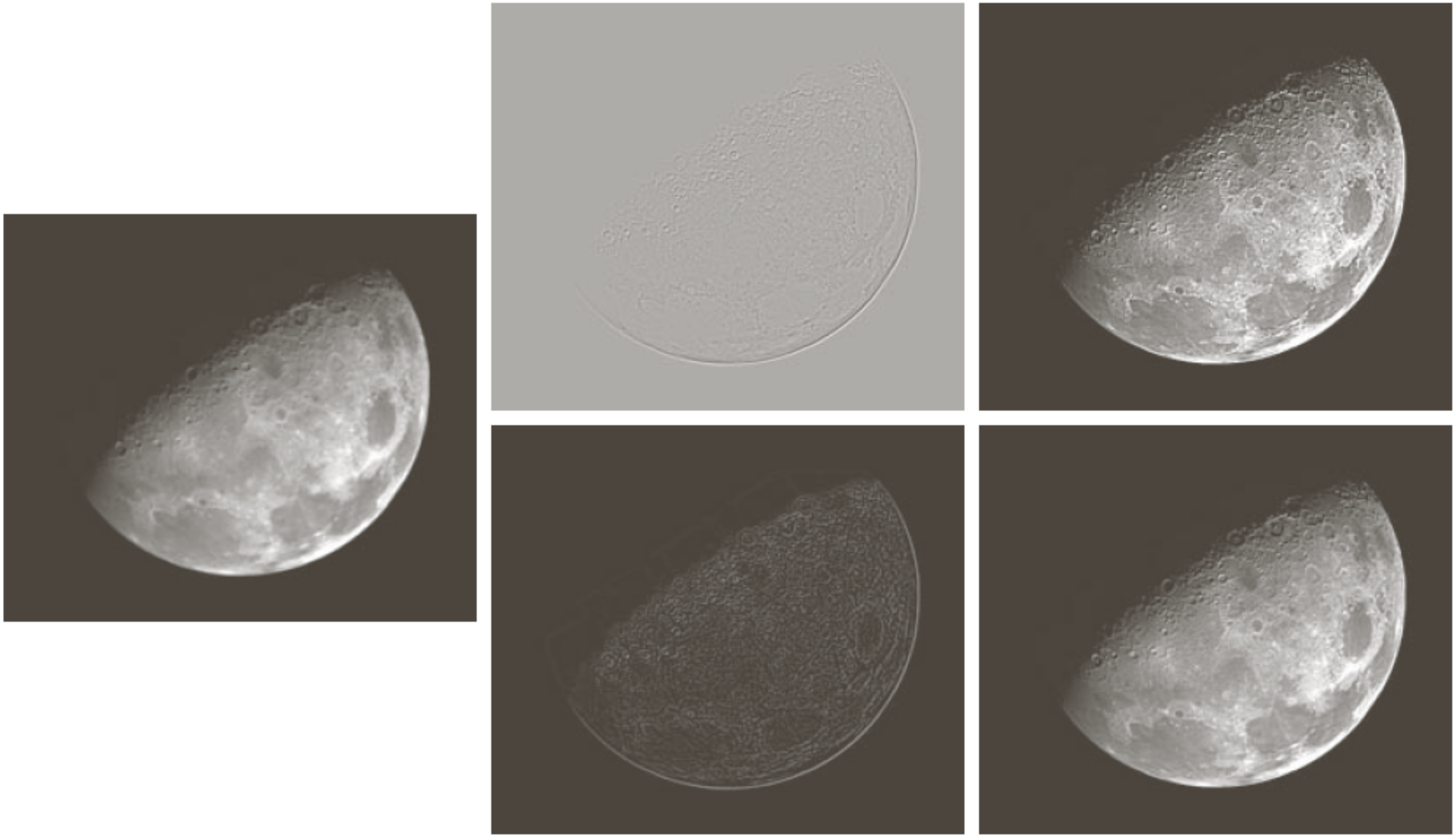


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Unscharfes Maskieren

- Originalbild unschärfer machen (Tiefpass filtern)
- Unscharfes Bild vom Original subtrahieren (ergibt die unscharfe Maske)
- Maske (nach Skalierung) zum Original addieren

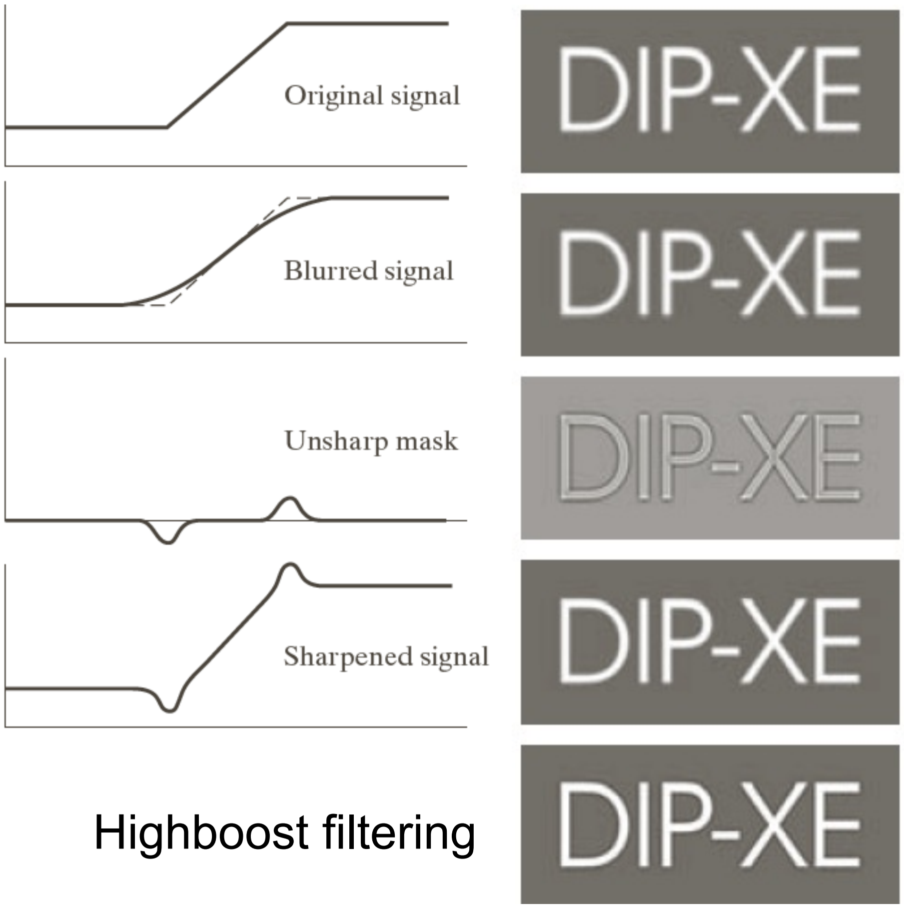
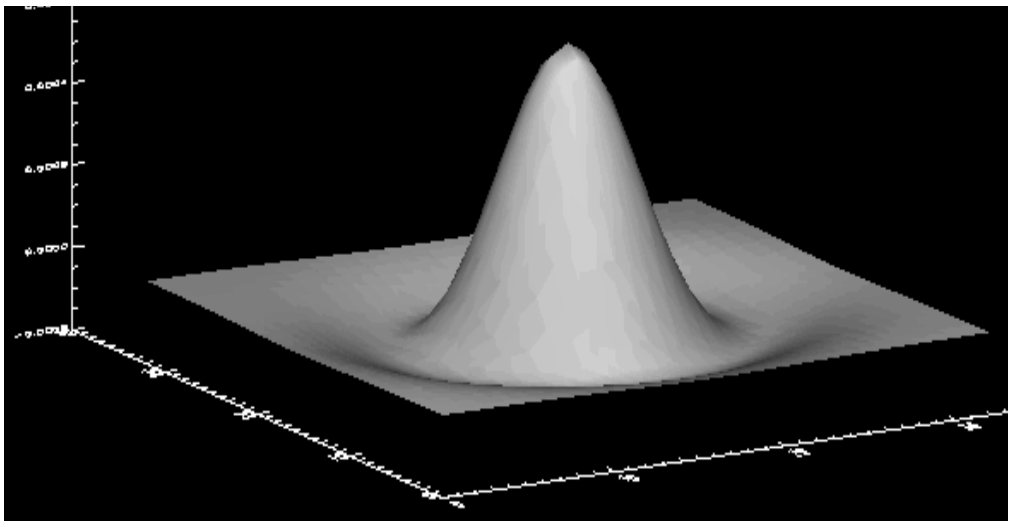
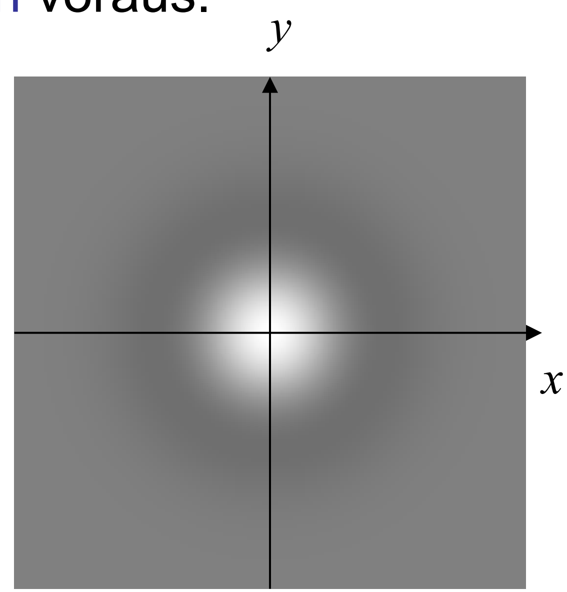


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

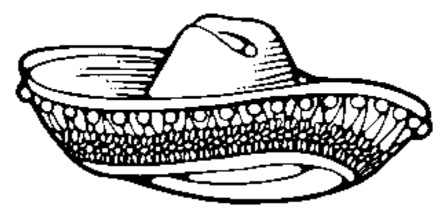
Marr-Hildreth-Filter = LoG-Filter

LoG-Filter: Laplacian-of-Gaussian, d.h. der Faltung mit dem Laplacefilter geht eine Glättung mit einer Gaußfunktion voraus.





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

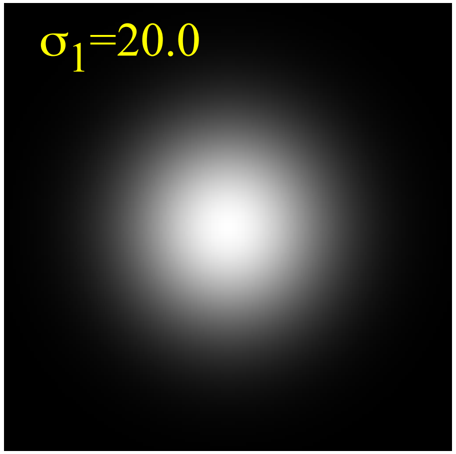


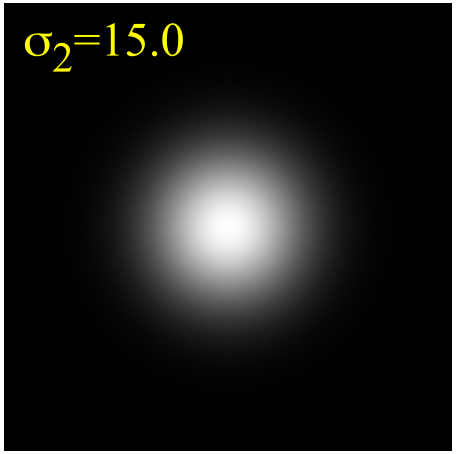
Auch genannt: „Mexican hat“ filter

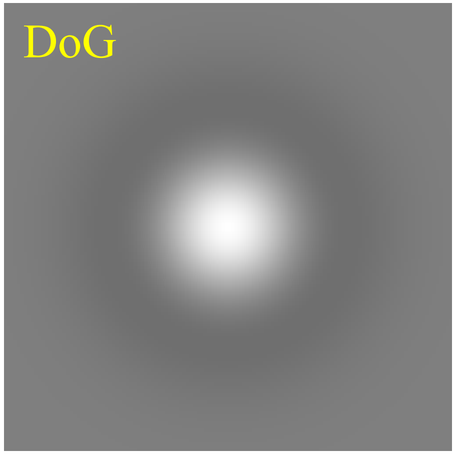
DoG (Difference of Gaussians)

Subtraktion zweier Gaußfilter mit unterschiedlicher Varianz σ^2:

$$\mathrm{DoG}(x,y,\sigma_1,\sigma_2) = [\sqrt{2\pi}\cdot\sigma_1]^{-1} \cdot \exp(-(x^2+y^2)/2\sigma_1^2) - [\sqrt{2\pi}\cdot\sigma_2]^{-1} \cdot \exp(-(x^2+y^2)/2\sigma_2^2)$$







DoG und LoG-Filter sind in ihrer Wirkungsweise vergleichbar.

TRANSFORMATION UND INTERPOLATION

Transformation und Interpolation

- Die Transformationen Translation, Rotation und Skalierung sind auf reellen Zahlen definiert:

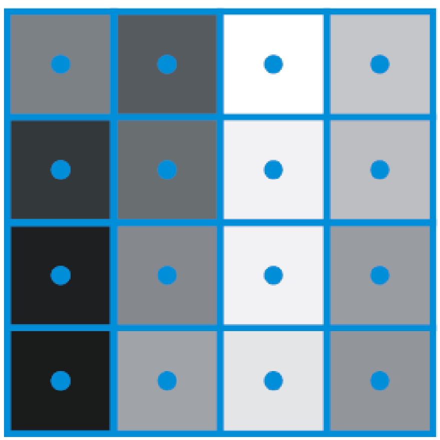
$$Rot_{\alpha}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}, \quad Tr_{dx,dy}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x+dx \\ y+dy \end{pmatrix}, \quad Sc_{s}\begin{pmatrix} x \\ y \end{pmatrix} = s\begin{pmatrix} x \\ y \end{pmatrix}$$

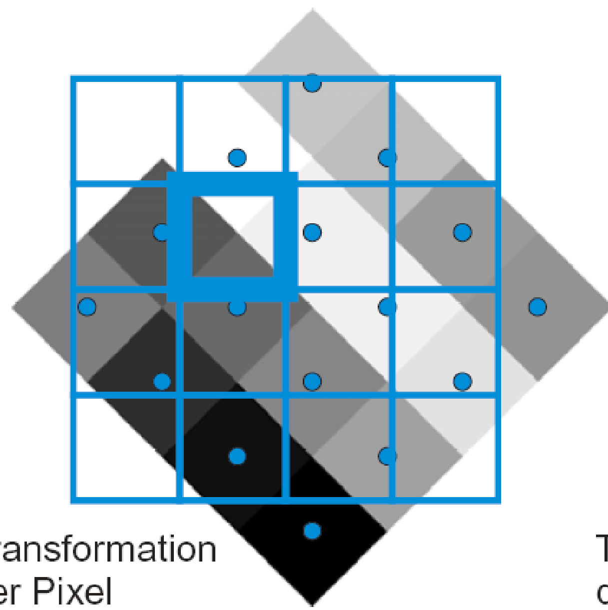
- Digitale Bilder haben einen ganzzahligen Definitionsbereich.
- Nach Transformation ist eine Interpolation notwendig.

Interpolation

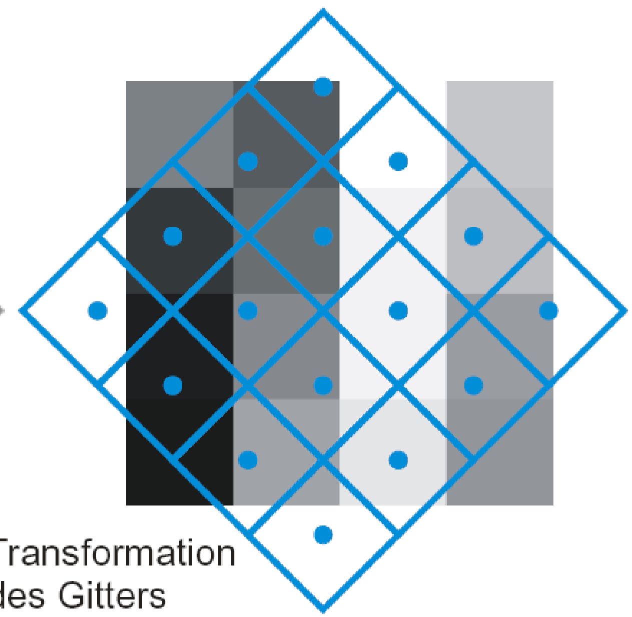
- Konstante Interpolation (Wert des nächsten Nachbarpixels)
- Lineare Interpolation
- Interpolation durch Polynome höheren Grades.
- Interpolation im Frequenzraum.

Konstante Interpolation



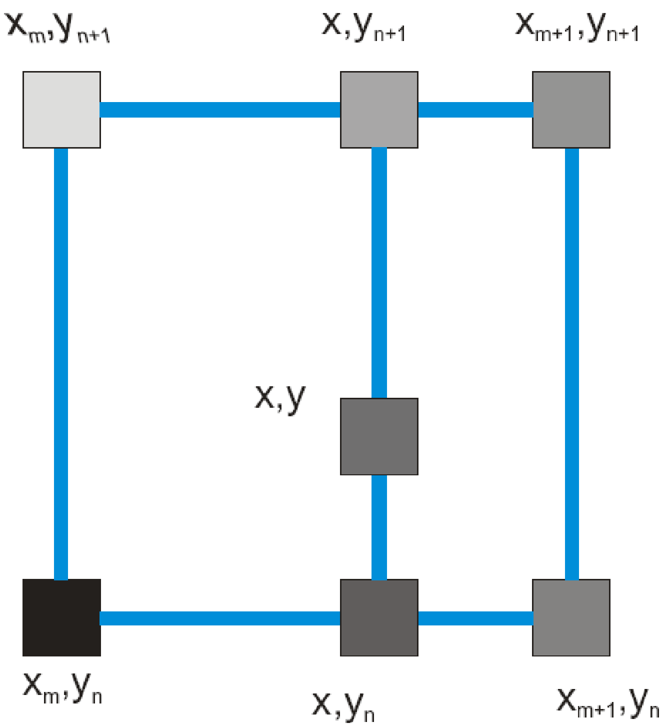


Transformation
der Pixel



Transformation
des Gitters

Bilineare Interpolation



Erster Schritt:

$$g_1(x_m,y)=\frac{y_{n+1}-y}{y_{n+1}-y_n}f(x_m,y_n)+\frac{y-y_n}{y_{n+1}-y_n}f(x_m,y_{n+1}),$$

$$g_1(x_{m+1},y)=\frac{y_{n+1}-y}{y_{n+1}-y_n}f(x_{m+1},y_n)+\frac{y-y_n}{y_{n+1}-y_n}f(x_{m+1},y_{n+1}).$$

Zweiter Schritt:

$$g(x,y)=\frac{x_{m+1}-x}{x_{m+1}-x_m}g_1(x_m,y)+\frac{x-x_m}{x_{m+1}-x_m}g_1(x_{m+1},y).$$

Polynome höheren Grades

- Interpolation der Bildfunktion durch mehr als 2 Stützpunkte
  - Polynom n-ten Grades interpoliert n+1 Punkte
- Die Bildfunktion wird besser angenähert, wenn mehr Terme der Taylor-Approximation berücksichtigt werden.
  - Ableitungen für Taylor-Reihe durch Differenzen angenähert
- Grad des Polynoms ist ein Kompromiss zwischen
  - steigender Anzahl berücksichtigter Terme der Taylor-Reihe
  - steigender Ungenauigkeit der geschätzten Ableitungen