

3. Bildoperationen

3.1 Übersicht zu Bildoperationen

Man unterscheidet Punktoperationen (wo für jedes Pixel unabhängig gerechnet werden kann), lokale Operationen und globale Operationen.

Es sei g_I das Input-Bild (oder eine Bildfolge), g_O das Output-Bild.

Klassen von Operationen

Punktoperation

$$g_O(i, j) = f \{ g_I(i, j, k) \mid k = 0, 1, \dots, K - 1 \}$$

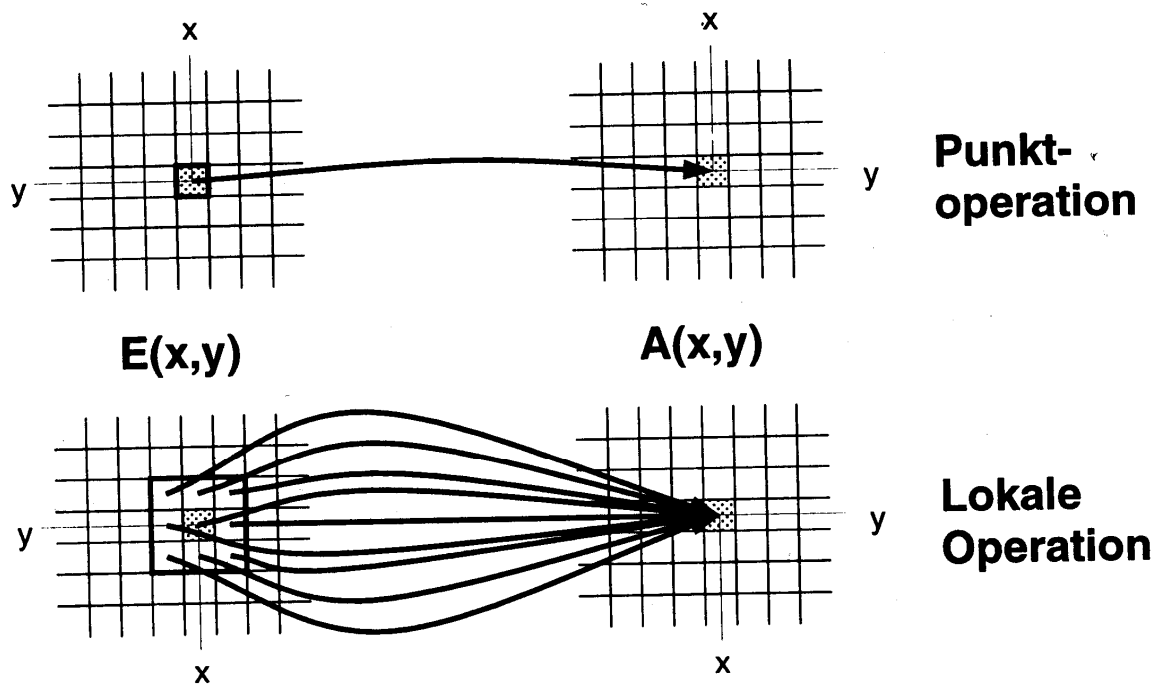
Lokale Operation

$$g_O(i, j) = f \left[\left\{ g_I(i - l, j - m, k) \mid l = -\frac{L-1}{2}, \dots, +\frac{L-1}{2}; \right. \right. \\ \left. \left. m = -\frac{M-1}{2}, \dots, +\frac{M-1}{2}; k = 0, \dots, K - 1 \right\} \right]$$

Globale Operation

$$g_O(i, j) = f \left[\left\{ g_I(i, j, k) \mid i = 0, \dots, I - 1; j = 0, \dots, J - 1; \right. \right. \\ \left. \left. k = 0, \dots, K - 1 \right\} \right]$$

(Liedtke 2002)



(aus Pinz 1994)

zusätzliche Klasse: geometrische Transformationen
(Skalierung, Drehung, Verzerrung...).

3.2 Punktoperationen

- Punktoperationen stehen auf der untersten Komplexitätsstufe der Bildverarbeitungsoperatoren
- neuer Grauwert wird ohne Berücksichtigung der Umgebung des jeweiligen Punktes berechnet

häufige Realisierung:
Lookup-Table (LUT)

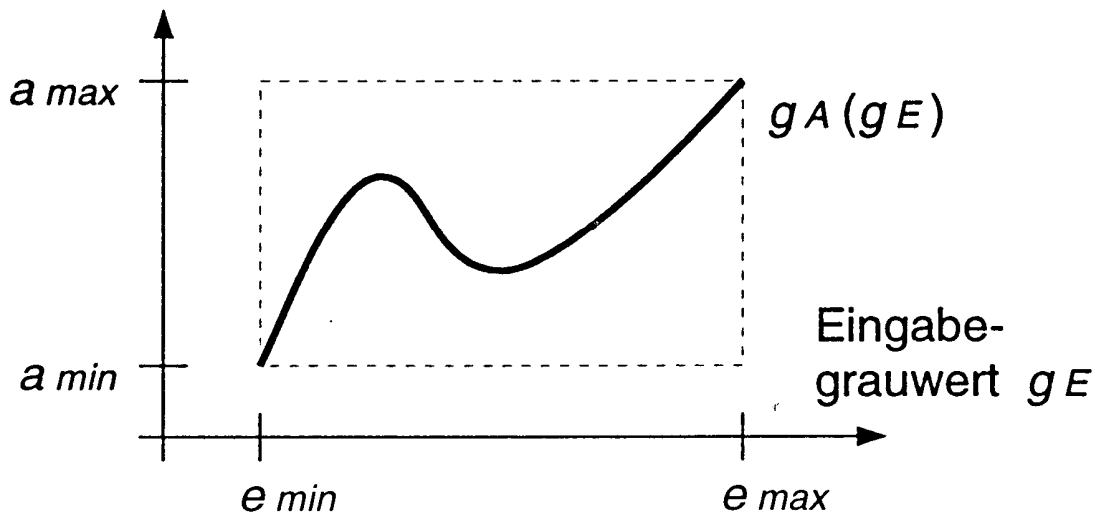
$$g_o(i, j) = \text{LUT}(g_i(i, j))$$

Speicher: unter Adresse x (Eingabe-Grauwert) ist Inhalt (transformierter Grauwert) y

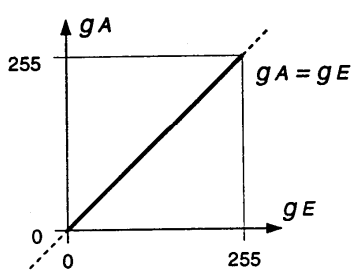
Vorteil: Berechnung erfolgt nur einmal; sehr schnell – auch Hardware-Unterstützung

Darstellung als Tabelle oder in Diagrammform

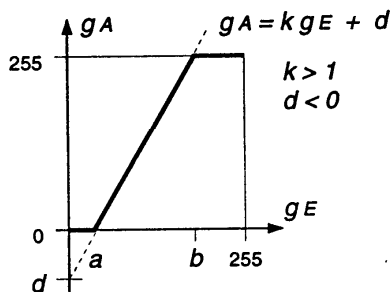
Ausgabegrauwert g_A



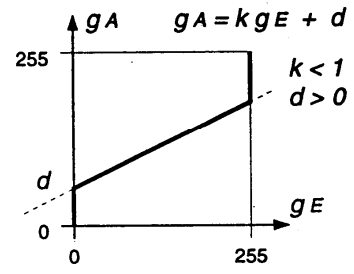
häufig verwendete Funktionen für LUTs:



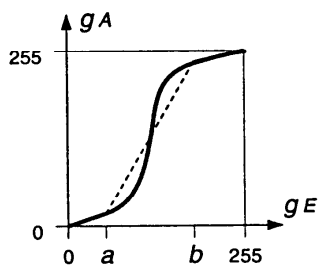
a. Identität



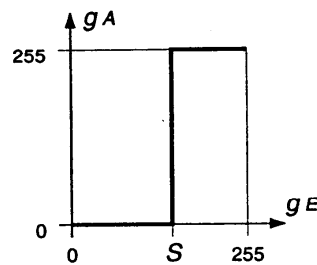
b. Kontrasterhöhung



c. Kontrastverminderung



d. Detailkontrasterhöhung



e. Schwellwertoperation

Anwendungsbeispiele:

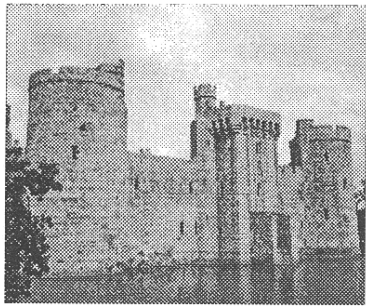
Invertierung



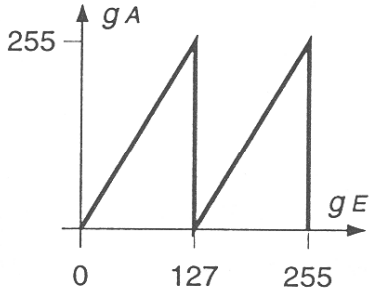
Dunkelung / Aufhellung



Schwellenwert / Spreizung



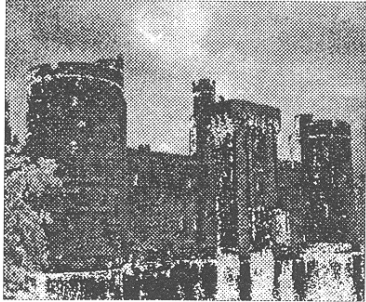
(a)



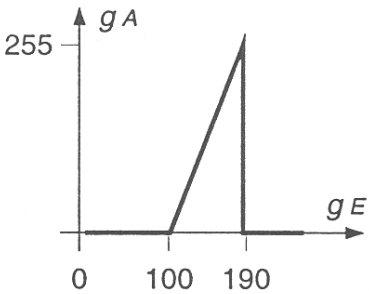
(b)

g_E	g_A
0	0
1	2
...	...
127	255
128	0
...	...
255	255

(c)



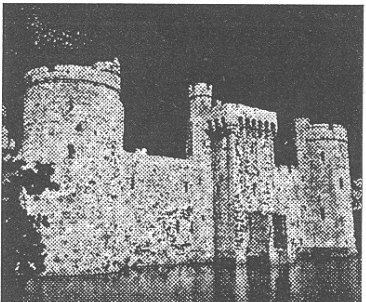
(d)



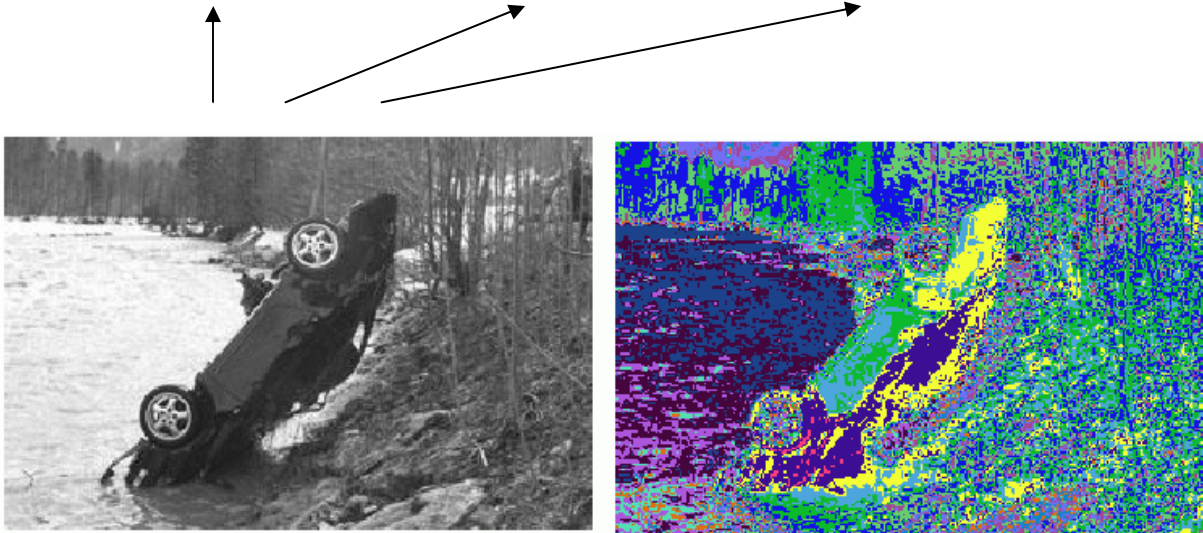
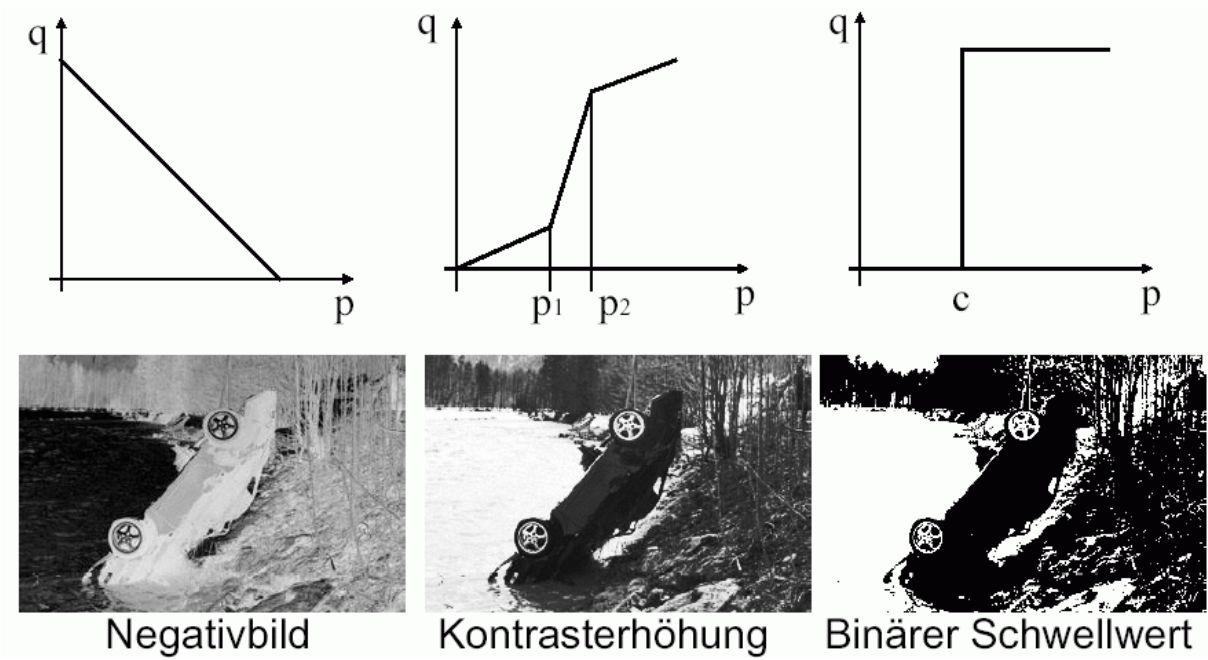
(e)

g_E	g_A
0	0
...	...
100	0
...	...
190	255
191	0
...	...
255	0

(f)



(g)



(aus Beichel 2002)

Dehnung (Spreizung) der Grauwertskala:

allg. Transformationsgleichung

$$g_o(i, j) = u \cdot g_l(i, j) + v$$

Konstante u beeinflusst den Kontrast, v die Helligkeit

Sei E das Eingabebild, A das Ausgabebild.

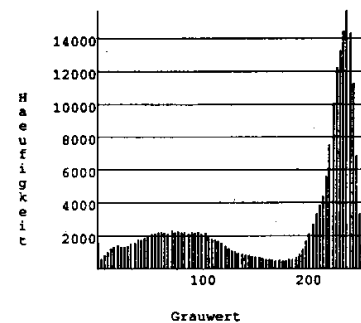
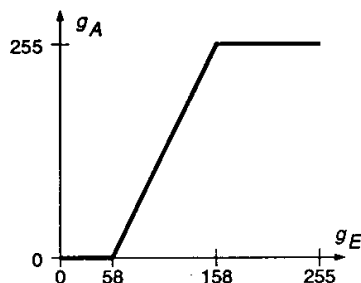
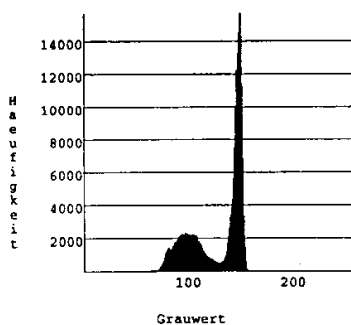
➤ Dehnung GW-Skala

➤ Diskreter Fall (mit 8Bit)

$$A(x, y) = \begin{cases} 255 & , \text{wenn } u \cdot E(x, y) + v > 255 \\ 0 & , \text{wenn } u \cdot E(x, y) + v < 0 \\ u \cdot E(x, y) + v & , \text{sonst} \end{cases}$$

➤ nicht linear

Wirkung auf das Histogramm:



➤ Lineare Dehnung GW-Skala mit Clipping

➤ Abschneiden der Extremwerte + Dehnung

➤ Man wählt Intervall, sodass

$$E_{\min} \leq E'_{\min} \leq E'_{\max} \leq E_{\max}$$

➤ Nur 1% - 5% der (Extrem-)Werte sollten jeweils in den Intervallen liegen

$$E_{\min} - E'_{\min} \quad \text{bzw.} \quad E_{\max} - E'_{\max}$$

Histogramm-Einebnung (Grauwertäqualisation)

- bewirkt Gleichverteilung der Grauwert über die (diskrete) Grauwertskala
- somit: annähernd ebenes Histogramm
- kontrastverstärkend

siehe Übung

Binarisierung

= Überführung in ein Binärbild (schwarz-weiß)

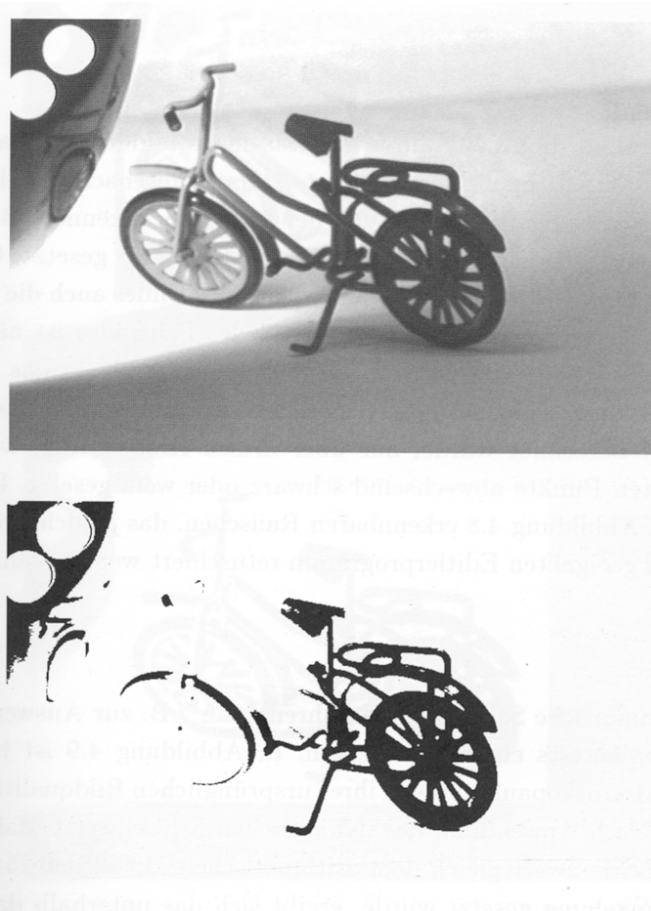
Schwellenwert-Operation

Schwellenwert "idealerweise" in einem "Tal" des Histogramms

Probleme:

- eindeutiges Optimum nicht immer vorhanden
- zweckmäßiger Schwellenwert kann mit der Position im Bild variieren

Beispiel:



(aus Klas 1996)

deshalb:

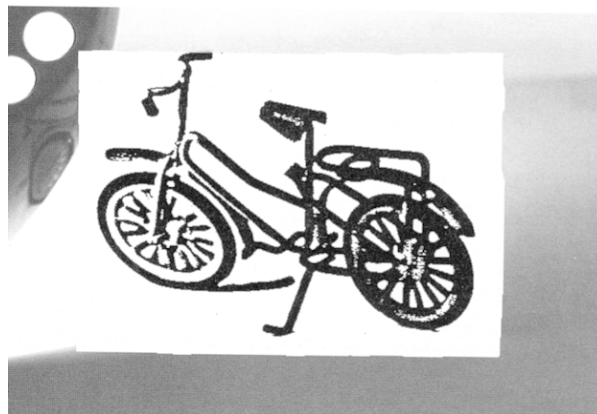
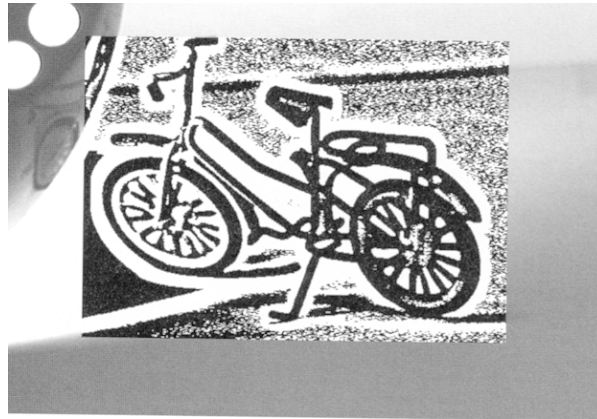
Dynamischer Schwellenwert (positionsabhängig)

Schwellenwert als Mittelwert der Helligkeit in einer Umgebung des betrachteten Pixels (z.B. n^2 Nachbarpixel)

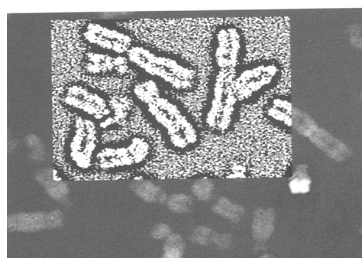
(dann keine Punktoperation mehr, sondern lokale Op.!)

Nachteil: es kann Hintergrundrauschen entstehen
Entfernen durch Glättungsverfahren oder interaktiv
(Retuschieren)

Beispiel (mit Mittelung über 50×50 Bildpunkte, oben;
unten retuschiert):



Anwendung dynamischer Schwellenwertverfahren u.a. in der
Mikroskopbild-Auswertung



ungleichmäßige Ausleuchtung oder inhomogene Sensor-Empfindlichkeit auch für die Weiterverarbeitung störend

daher:

Helligkeitskorrektur (auch: Shading-Korrektur)

einfachste Methode:

- Szene einmal mit und einmal ohne die interessanten Bildbestandteile aufnehmen (Referenzbild)
- Hintergrundeinfluss durch Bildsubtraktion beseitigen (Mikroskopbildanalyse)

wenn Helligkeit bei der Bildaufnahme künstlich geregelt werden kann (z.B. Mikroskop):

Multiplikative oder additive Shading-Korrektur durch *Verrechnen mit Weiß- und Schwarzbild*

➤ Shadingmatrix:

➤ Aufnahme eines Weißbildes $W(x,y)$, mit

➤ Vollständigem Öffnen der Aperturblende und

➤ Vollständigen „Hochregeln“ des Lichts

➤ Berechnung aus $W(x,y)$ des mittleren Weißwerts \overline{W}

➤ Sowie Aufnahme Schwarzbild $S(x,y)$, mit

➤ Geschlossener Aperturblende

➤ Ohne Licht

➤ Berechnung aus $S(x,y)$ des mittlere Schwarzwerts \overline{S}

➤ Multiplikative Shadingkorrektur

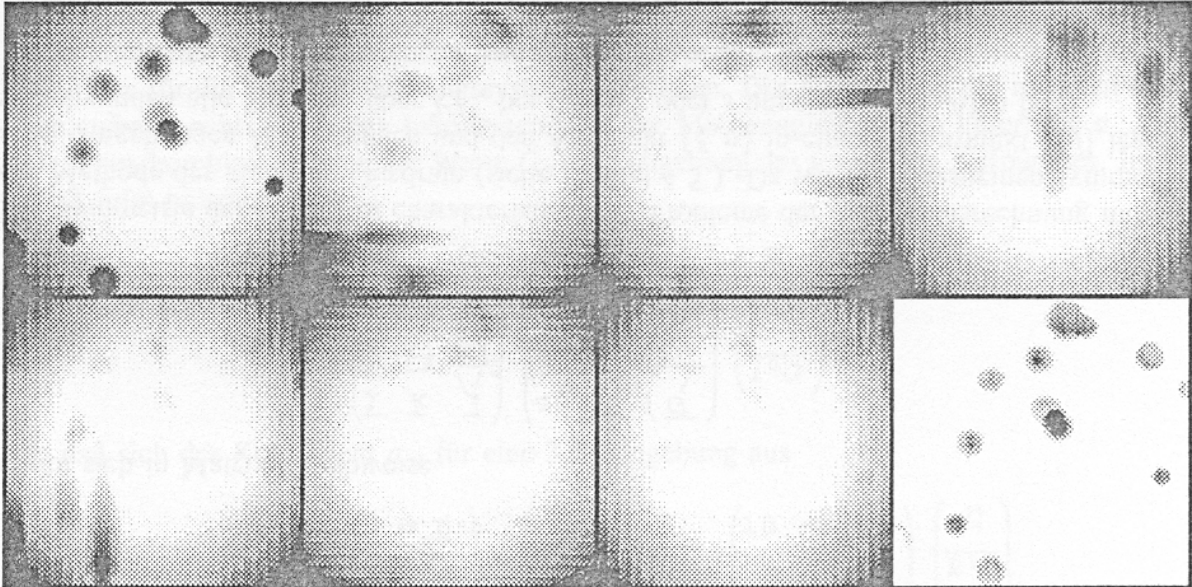
$$A_M(x, y) = E(x, y) \frac{\overline{W}}{W(x, y)} - \overline{S}$$

➤ Additive Shadingkorrektur

$$A_A(x, y) = E(x, y) - W(x, y) + \overline{W} - \overline{S}$$

wenn dies nicht möglich:
(inhomogenen) Hintergrund aus dem Bild selbst extrahieren
und vom Bild subtrahieren
(- keine Punktoperation mehr!)

- gleitendes Minimum, iteriert in verschiedene Richtungen:



(aus Voss & Süße 1991)

Punktoperationen mit mehreren Eingangsbildern

(3) Differenzbildung von Bildern

- Hintergrundsubtraktion

$$g_O(i,j) = g_I(i,j) - g_B(i,j)$$

- Änderungsdetektion

$$g_O(i,j) = | g_I(i,j,k) - g_I(i,j,k+1) |$$

(4) Summenbildung von Bildern

- Lineartransformation

$$g_O(i,j) = a \cdot g_R(i,j) + b \cdot g_G(i,j) + c \cdot b_B(i,j)$$

- Rauschminderung durch Mittelung

$$g_O(i,j) = \frac{1}{K} \sum_{k=0}^{K-1} g_I(i,j,k)$$

(5) Produktbildung von Bildern

(Maskierung, Shading-Korrektur)

$$g_O(i,j) = g_B(i,j) \cdot g_I(i,j)$$

(nach Liedtke 2002)

3.3 Geometrische Transformationen

Gegenstück zu Helligkeitstransformationen:
Helligkeiten der Pixel bleiben (wenigstens näherungs-
weise) erhalten, Ortskoordinaten ändern sich

Typische Beispiele:

Skalierung, Drehung, Spiegelung, Ver-, Entzerrung

Anwendungen:

Normierung der Bildgröße, Zoom, Angleichung zur Bild-
überlagerung, Verdrehungskorrektur beim Scannen,
Linsenkorrektur

mathematisch: Vektorfunktionen $T: (x, y) \rightarrow (x', y')$
 $x' = T_x(x, y), y' = T_y(x, y)$

T_x und T_y sind entweder bekannt oder berechenbar aus
der Lage von einigen Pixeln im Originalbild und im trans-
formierten Bild (*Passpunkte*)

wegen diskreter Natur des Digitalbildes: nach Anwendung von
 T i.allg. Interpolation erforderlich

somit 2 Schritte:

- Transformation der Pixelkoordinaten
- Grauwert-Interpolation (da transformierte Koordinaten
i.allg. nicht ganzzahlig) (*siehe später*)

eigentliche geometrische Transformation:

Vektorfunktion,

oft: *affine Abbildung* (lineare Abb. + Verschiebung)

daneben auch bilineare und polynomiale Abbildungen

Beschreibung einer affinen 2D-Transformation durch 3×3-Matrix P (homogene Koordinaten; vgl. Praktikum Computergrafik, Skript, Kap. 4):

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix}$$

Bildpunkt E mit alten Koordinaten (x, y) und neuen Koord. (x', y')

- Als homogene Koordinaten (in Spaltenvektorschreibweise)

$$E = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \text{und} \quad E' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

- Neue Koordinatenwerte durch lineare Kombinationen der alten Werte mit den Koeffizienten der Transformationsmatrix

$$\begin{aligned} x' &= P_{11} \cdot x + P_{12} \cdot y + P_{13} \\ y' &= P_{21} \cdot x + P_{22} \cdot y + P_{23} \\ 1 &= P_{31} \cdot x + P_{32} \cdot y + P_{33} \end{aligned}$$

- Transformiertes Koordinatenpaar (x', y') (in Spaltenform) durch Multiplikation eine Matrix mit einem Spaltenvektor (alte Koordinaten)

$$E' = P \cdot E$$

- Translation

- Verlegung des Ursprungs um/auf $-t_x, -t_y$
- Achsenverlauf bleibt erhalten
- Keine Skalierung

$$\begin{aligned} x' &= 1 \cdot x + 0 \cdot y - t_x \\ y' &= 0 \cdot x + 1 \cdot y - t_y \end{aligned} \quad T = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$

➤ Skalierung

➤ Achsenverlauf und Ursprung bleiben erhalten

➤ Skalierung der Achsen mit S_x bzw. S_y

$$x' = S_x \cdot x + 0 \cdot y + 0$$

$$S = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

➤ Rotation

$$y' = 0 \cdot x + S_y \cdot y + 0$$

➤ Skalierung und Ursprung bleiben erhalten

➤ Veränderung der Achsenrichtung um Winkel Φ

$$x' = \cos \Phi \cdot x + \sin \Phi \cdot y + 0$$

$$y' = -\sin \Phi \cdot x + \cos \Phi \cdot y + 0$$

$$R = \begin{pmatrix} \cos \Phi & \sin \Phi & 0 \\ -\sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Kombination mehrerer affiner Abb. durch Matrizenmultiplikation zu berechnen

Approximation beliebiger Transformationen (z.B. Verzerrungen)

Ansätze:

- Polynom-Transformation

Approximation durch polynomiale Gleichung:

$$x' = \sum_{r=0}^m \sum_{k=0}^{m-r} a_{rk} x^r y^k$$

$$y' = \sum_{r=0}^m \sum_{k=0}^{m-r} b_{rk} x^r y^k$$

- **Linear** in Bezug auf die Koeffizienten a,b
- Aus korrespondierenden **Punktpaaren** $(x,y), (x',y')$ können die **Koeffizienten** durch Lösung linearer Gleichungen bestimmt werden
- Verwendung von mehr Punkten als unbedingt notwendig führt zu robusteren Lösungen
- Für einfachere Verzerrungen typischerweise $m=2..3$
- 6 bis 10 Punktpaare zur Lösung ausreichend

- **Bilineare Transformation** $x' = a_0 + a_1x + a_2y + a_3xy$
 – 4 Punktpaare ausreichend $y' = b_0 + b_1x + b_2y + b_3xy$
- **Affine Transformation** $x' = a_0 + a_1x + a_2y$
 – 3 Punktpaare ausreichend $y' = b_0 + b_1x + b_2y$
 – Rotation, Translation, Skalierung, Scherung

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Kennzahl einer Transformation:

Der *Jacobi-Koeffizient* (Funktionaldeterminante)

- liefert Information über die Koordinatenänderung

$$J = \left| \frac{\partial(x', y')}{\partial(x, y)} \right| = \begin{vmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} \end{vmatrix}$$

- J=0: Singuläre Transformation (keine Inverse)
- J=1: Fläche invariant bzgl. Transformation

- **Bilineare Transformation**

$$J = a_1b_2 - a_2b_1 + (a_1b_3 - a_3b_1)x + (a_3b_2 - a_2b_3)y$$

- **Affine Transformation** $J = a_1b_2 - a_2b_1$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Translation: J=1

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation: J=1

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & 0 \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Skalierung: J=ab

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \tan \phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scherung: J=1

Der Interpolationsschritt

Koordinatentransformation trifft meist nicht genau das Raster;

Werte auf dem Raster werden benötigt:

Interpolation, dazu meist Verwendung der Rückwärtstransformation T^{-1} zur verwendeten geom. Transformation T

- Ursprüngliche Bildfunktion liegt nur als abgestastete Funktion vor:

$$g_s(l\Delta x, k\Delta y)$$

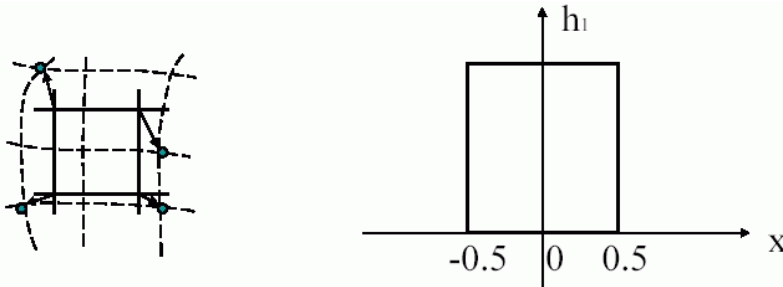
- Helligkeit kann als Faltung der ursprünglichen Bildfunktion mit einem Interpolationskern h_n betrachtet werden

$$f_n(x, y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_s(l\Delta x, k\Delta y) h_n(x - l\Delta x, y - k\Delta y)$$

h_n kann als Gewichtungsterm für g_s an verschiedenen (Nachbarschafts-) Pixeln angesehen werden.

konkrete Ansätze:

Nearest Neighbourhood Interpolation

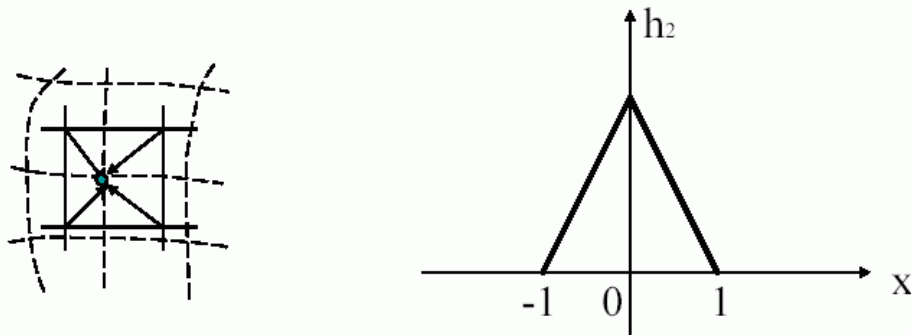


Jeder Punkt im Raster bekommt den Grauwert des nächsten Nachbarn zugewiesen.

$$f_1(x, y) = g_s[\text{round}(x), \text{round}(y)]$$

Probleme bei linienhaften Objekten: Stufen entstehen

Lineare Interpolation



Betrachtet die Helligkeit als lineare Funktion der Helligkeit der 4 umgebenden Punkte

$$f_2(x, y) = (1-a)(1-b)g_s(l, k) + a(1-b)g_s(l+1, k) + b(1-a)g_s(l, k+1) + abg_s(l+1, k+1)$$

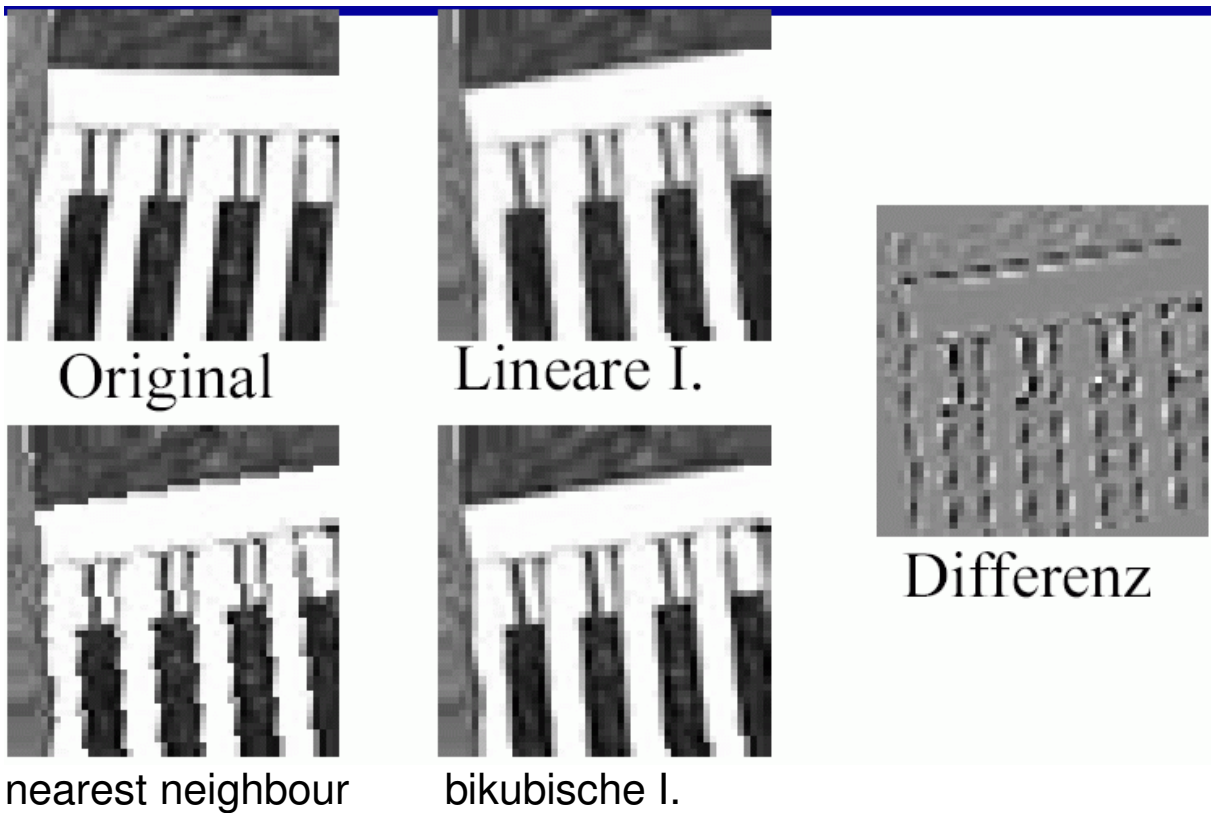
$$l = \text{round}(x), \quad a = x - l$$

$$k = \text{round}(y), \quad b = y - k$$

Bikubische Interpolation

- approximiert die Helligkeitsfunktion lokal durch eine bikubische Fläche (vgl. Skript Computergrafik)
- es werden 16 Nachbarschaftspunkte verwendet
- erhält feine Details im Bild

Beispiele:



3.4 Lokale Bildoperationen

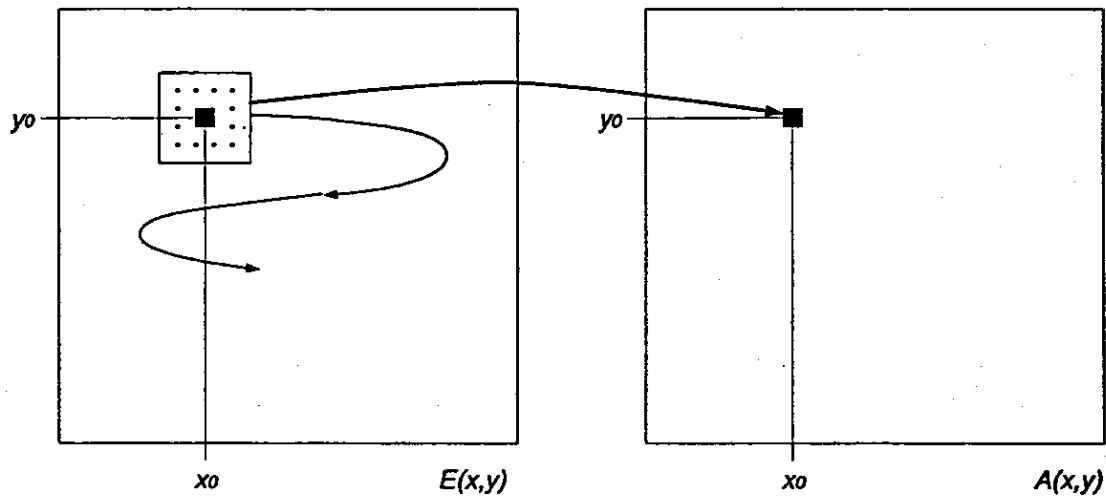
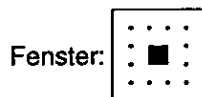
verändern Pixel in Abhängigkeit von Umgebung dieses Pixels

- Lineare Transformationen

$$f(i, j) = \sum_{(m,n) \in O} h(i-m, j-n)g(m, n)$$

entspricht diskreter Faltung mit h
 h ... Faltungskern, O ... lokale Umgebung

- Nichtlineare Transformationen
z.B. Medianfilter



Lokale Fensteroperation (E : Eingabebild, A : Ausgabebild)

Math. Grundlage für *lineare* lokale Transformationen:
Faltung zweier Matrizen K (= "Kern") und B (mehr in der Übung)

$$a_{jk} = \sum_{m=0}^{L-1} \sum_{n=0}^{R-1} k_{mn} \cdot b_{j-m, k-n}$$

in der Praxis oft auch um den Mittelpunkt der Kern-Matrix zentriert
(also $m, n = -(L-1)/2, \dots, (L-1)/2$); K manchmal anders indiziert.

kontinuierliches 1D-Analogon:

- Faltung zweier Signale (physikalisches Konzept)
- Faltungsintegral $f(x)$ definiert durch folgende Beziehung

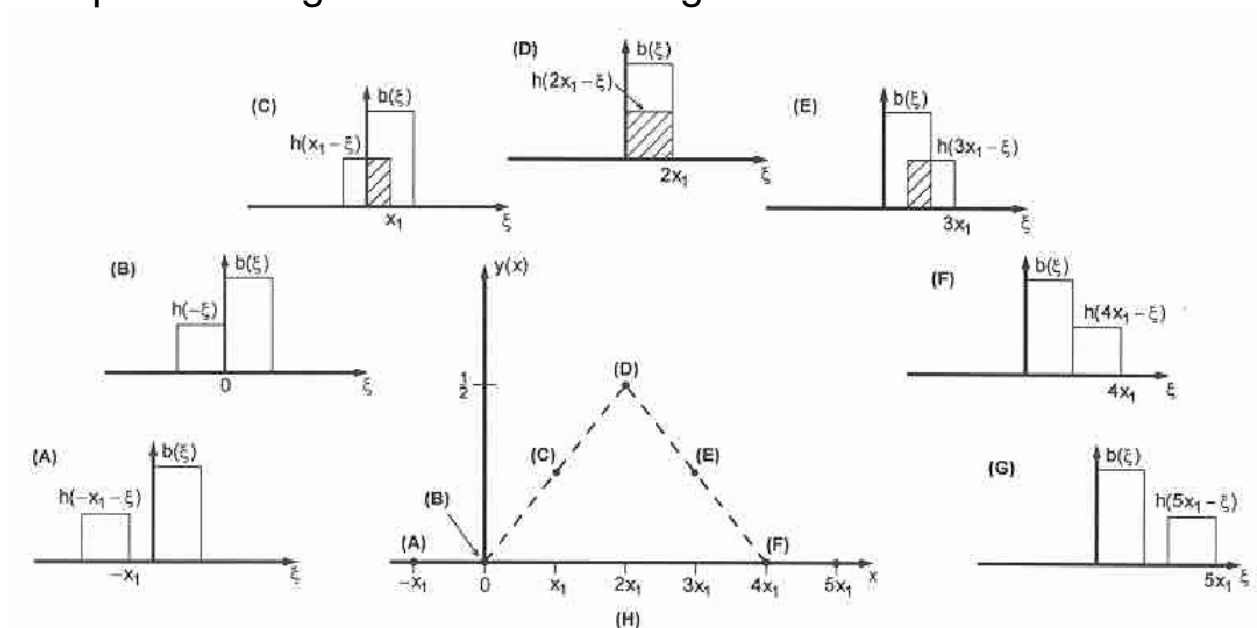
$$f(x) = \int_{\xi=-\infty}^{\infty} b(\xi)h(x-\xi)d\xi = b(x)*h(x)$$

$f(x)$ ist das Faltungsprodukt der Funktionen $b(x)$ und $h(x)$

wie ist das Integral anschaulich auszuwerten?

- Spiegelung der Funktion $h(\xi)$ an der Ordinate $\rightarrow h(-\xi)$
- Verschieben von $h(\xi)$ um $x \rightarrow h(x-\xi)$
- Multiplikation der verschobenen Funktion $h(x-\xi)$ mit $b(\xi)$
- Integration der Flächen unter dem Produkt $h(x-\xi)b(\xi)$.
Ergebnis ist der Wert des Faltungsintegrals am Ort x .

Beispiel: Faltung zweier Rechtecksignale



Effekt in 2D analog:
Werte der Bildmatrix werden (in gesetzmäßiger Weise)
"verschmiert"

Ein Beispiel:

8	5	4	6	8	10	13	11
5	5	5	6	11	14	14	14
5	3	7	5	12	15	13	12
5	5	5	8	15	17	14	13
4	5	12	18	20	16	17	16
8	14	16	20	18	20	15	21
13	22	20	18	17	18	22	27
21	21	18	20	20	25	30	32

Bild E

1	1	1
1	1	1
1	1	1

Matrix C

Zelle A_{22}

47	46	64	87	110	116		
45	49	74	103	125	126		
51	68	102	126	139	133		
74	103	132	152	152	149		
114	145	159	165	163	172		
153	169	167	176	185	210		

Bild A

lineare lokale Filterung eines Bildes:

der Grauwert eines Bildpunktes im transformierten Bild ergibt sich aus den Grauwerten des entsprechenden Punktes im Originalbild und dessen Nachbarn in einer Umgebung U durch *Linearkombination*.

Koeffizienten unabhängig von der Position:

LSI-Filter (linear shift invariant filter)

Jeder LSI-Filter lässt sich als Faltung darstellen.

Anwendungen:

A. Bildglättung (*Image Smoothing*)

- z.B. zur Rauschunterdrückung
- Beseitigen von störenden Details

- Mittelung funktioniert vor allem bei Impulsrauschen
- Große Flecken (blobs) können nur schwer entfernt werden

Eleganteste Art der Rauschunterdrückung:

Mittelung über mehrere Bilder (vom gleichen Objekt)



Verrauschtes
Bild

Mittelung über
10 verr. Bilder

$$f(i, j) = \frac{1}{n} \sum_{k=1}^n g_k(i, j)$$

wenn dies nicht möglich:
Mittelung in lokaler Umgebung

Anwendung verschiedener Faltungskerne

Beispiel: Anwendung eines 3×3-Mittelwertoperators
(Box-Filter)

Kern $h = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, anschließende Rundung auf ganzzahlige

Grauwerte

<table border="1"> <tbody> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>6</td><td>1</td><td>8</td><td>10</td><td>2</td><td>10</td></tr> <tr><td>1</td><td>3</td><td>1</td><td>1</td><td>9</td><td>10</td><td>7</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>2</td><td>8</td><td>9</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>4</td><td>1</td><td>2</td><td>9</td><td>10</td><td>2</td><td>10</td></tr> <tr><td>1</td><td>2</td><td>1</td><td>8</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> </tbody> </table>	1	1	1	1	10	10	10	10	1	1	6	1	8	10	2	10	1	3	1	1	9	10	7	10	1	1	1	2	8	9	10	10	1	1	1	1	10	10	10	10	1	4	1	2	9	10	2	10	1	2	1	8	10	10	10	10	1	1	1	1	10	10	10	10	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>2</td><td>2</td><td>4</td><td>7</td><td>8</td><td>9</td><td>0</td></tr> <tr><td>0</td><td>2</td><td>2</td><td>4</td><td>6</td><td>8</td><td>9</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>4</td><td>7</td><td>9</td><td>10</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>4</td><td>7</td><td>9</td><td>9</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>5</td><td>8</td><td>9</td><td>9</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>5</td><td>8</td><td>9</td><td>9</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	0	0	0	0	0	0	0	2	2	4	7	8	9	0	0	2	2	4	6	8	9	0	0	1	1	4	7	9	10	0	0	1	2	4	7	9	9	0	0	1	2	5	8	9	9	0	0	1	2	5	8	9	9	0	0	0	0	0	0	0	0	0
1	1	1	1	10	10	10	10																																																																																																																										
1	1	6	1	8	10	2	10																																																																																																																										
1	3	1	1	9	10	7	10																																																																																																																										
1	1	1	2	8	9	10	10																																																																																																																										
1	1	1	1	10	10	10	10																																																																																																																										
1	4	1	2	9	10	2	10																																																																																																																										
1	2	1	8	10	10	10	10																																																																																																																										
1	1	1	1	10	10	10	10																																																																																																																										
0	0	0	0	0	0	0	0																																																																																																																										
0	2	2	4	7	8	9	0																																																																																																																										
0	2	2	4	6	8	9	0																																																																																																																										
0	1	1	4	7	9	10	0																																																																																																																										
0	1	2	4	7	9	9	0																																																																																																																										
0	1	2	5	8	9	9	0																																																																																																																										
0	1	2	5	8	9	9	0																																																																																																																										
0	0	0	0	0	0	0	0																																																																																																																										
Ausgangsbild	transformiertes Bild																																																																																																																																

beachte:

- Randeffekt
- Max. Grauwert des Originals wird kaum erreicht
- kleine Störungen können "aufgebläht" werden



$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h = \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

links: Box-Filter

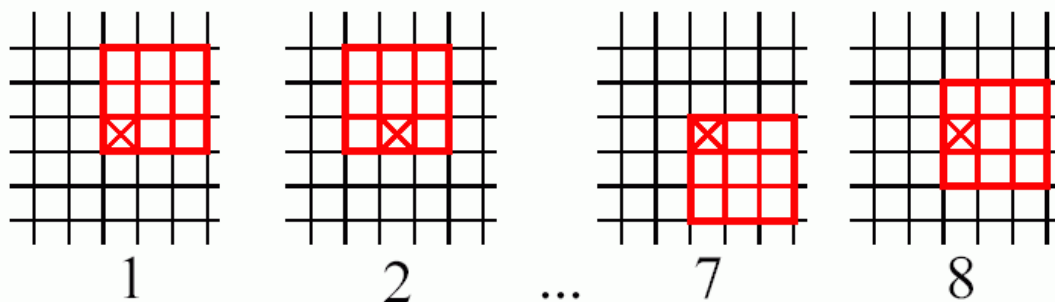
rechts: "Gauß-Filter", glatterer Verlauf \Rightarrow transformiertes Bild enthält weniger Oberschwingungen
zudem besserer Erhalt der Helligkeits-Extrema des Originals

Aufbau des $(n+1) \times (n+1)$ -Gaußfilters: $h = \frac{1}{2^{n^2}} \left[\binom{n}{j} \cdot \binom{n}{k} \right]_{j,k=0,\dots,n}$

= diskrete Approx. einer 2D-Gaußfunktion (rotierte Glockenkurve)

Mittlung mit rotierender Maske

- nichtlineares Verfahren
- vermeidet blurring an Kanten
- homogener Teil der lokalen Umgebung wird verwendet
- Helligkeitswert wird nur innerhalb dieser Region berechnet
- Verwendung der Dispersion (empirische Varianz σ^2) als Homogenitätsmaß



Algorithmus:

- 1) Betrachte jedes Pixel
- 2) Berechne Dispersion über alle Masken
- 3) Wähle die Maske mit minimaler Dispersion
- 4) Weise dem Pixel (i,j) den Mittelwert der Pixel dieser Maske zu

andere nichtlineare Mittelwertfilter:

- Verallgemeinerung der Mittelungstechniken

$$f(m,n) = u^{-1} \left\{ \frac{\sum_{i,j \in O} a(i,j) u[g(i,j)]}{\sum_{i,j \in O} a(i,j)} \right\}$$

- Gewichtungskoeffizienten $a(i,j)$ konstant: Homomorphe Filter
- Arithmetischer Mittelwert $u(g)=g$
- Harmonischer Mittelwert $u(g)=1/g$
- Geometrischer Mittelwert $u(g)=\log g$

Minimum-, Maximumfilter

Der 3×3-Min-Operator säubert die dunklen Regionen des Eingabebildes (vgl. Beisp. mit dem Mittelwertoperator), zerstört aber andererseits die hellen Regionen:

1	1	1	1	10	10	10	10
1	1	6	1	8	10	2	10
1	3	1	1	9	10	7	10
1	1	1	2	8	9	10	10
1	1	1	1	10	10	10	10
1	4	1	2	9	10	2	10
1	2	1	8	10	10	10	10
1	1	1	1	10	10	10	10

Eingabebild

0	0	0	0	0	0	0	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	1	1	1	1	8	7	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	1	1	1	1	2	2	0
0	0	0	0	0	0	0	0

Ergebnis des Min-Op.

Der Max-Operator wirkt analog asymmetrisch.

Verallgemeinerung von Min und Max:

Rangordnungsoperatoren

Idee:

- Verwendung einer beliebigen Maske ("strukturierendes Element")
- auftretende Helligkeitswerte der Eingabematrix innerhalb der Maske werden nach der Größe sortiert
- es wird der k -te Wert in der sortierten Liste (k -Quantil) als Ausgabe-Helligkeit benutzt

Spezialfälle:

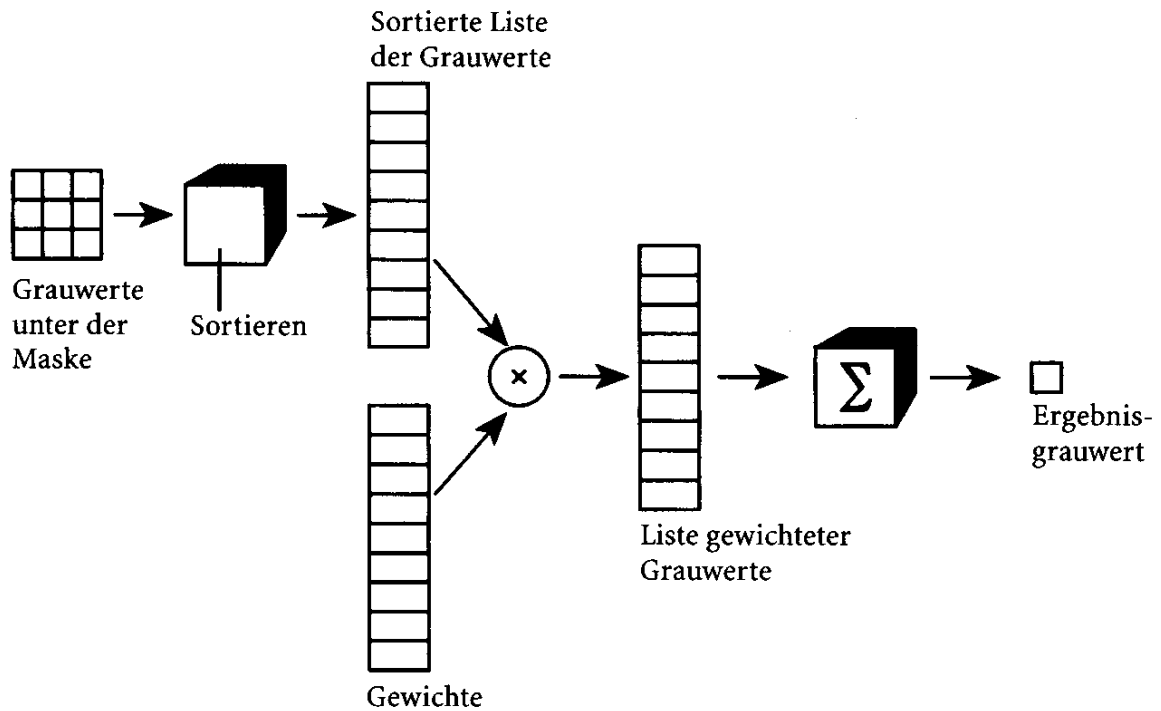
$k=1$: Min-Operator,

$k = \text{höchster Wert} = n$: Max-Operator

$k=n/2$: Median-Operator

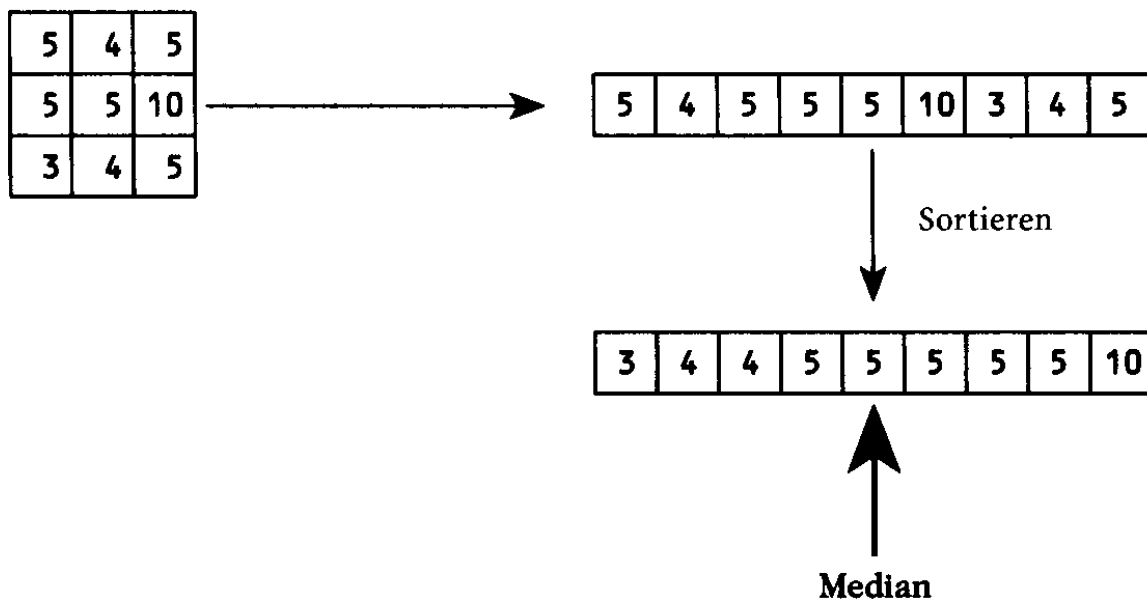
- evtl. mehrere Werte aus der sortierten Liste mit Gewichten verknüpfen

Prinzip des allgemeinen Rangordnungsoperators (mit Gewichten):



(aus Bässmann & Kreys 1998)

Spezialfall Medianoperator:



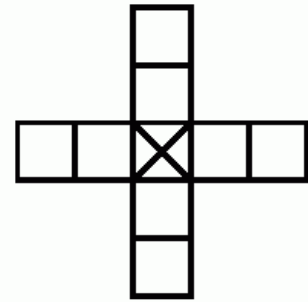
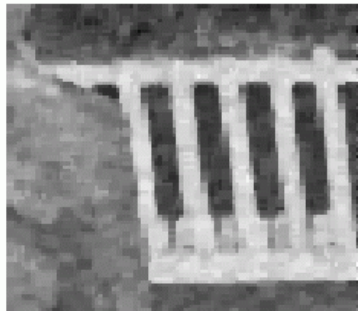
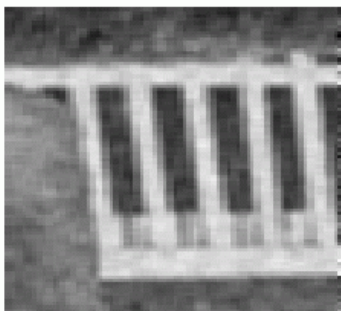
Beispiele für die Anwendung:



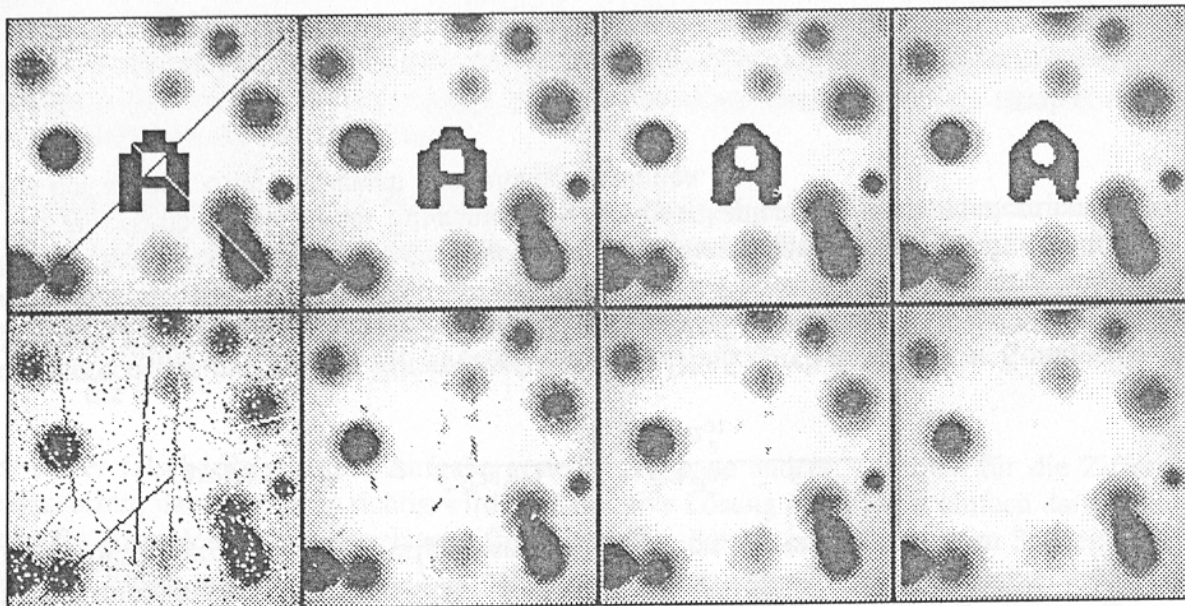
Mittelwertfilter



Medianfilter

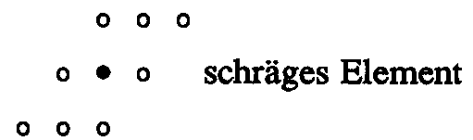
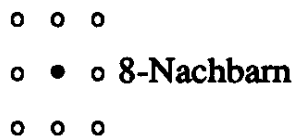


Linienerhaltende
Maske



Medianfilter mit quadratischer Maske (Seitenlängen 3; 5; 7), angewandt auf die ganz links stehenden Originalbilder (aus Voss & Süße 1991)

unterschiedliche strukturierende Elemente:

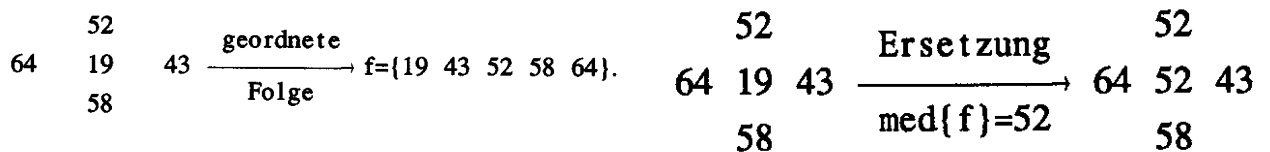


Anwendung des Median-Operators auf folgendes Grauwertbild:

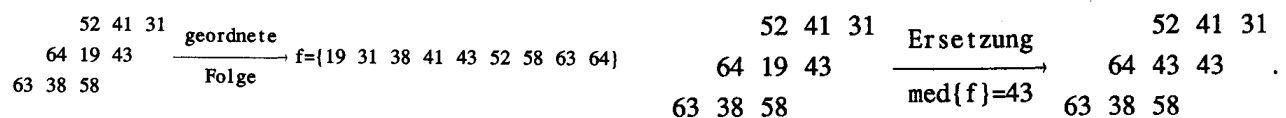
	0	1	2	3	4	...	Spalte y
0	53	68	47	36	27	...	
1	57	67	52	41	31	...	
2	61	64	19	43	35	...	
3	63	38	58	47	41	...	
...							

Zeile x

mit Elementarraute als strukt. Element:



mit schrägem Element:



(Beisp. aus Haberäcker 1991)

Vorteile des Medianfilters:

- verringert das Rauschen, ohne steile Grauwertkanten zu verschmieren
- vereinigt die "säubernden" Eigenschaften von Min- und Max-Operator
- liefert eine gewisse Kontrastverstärkung

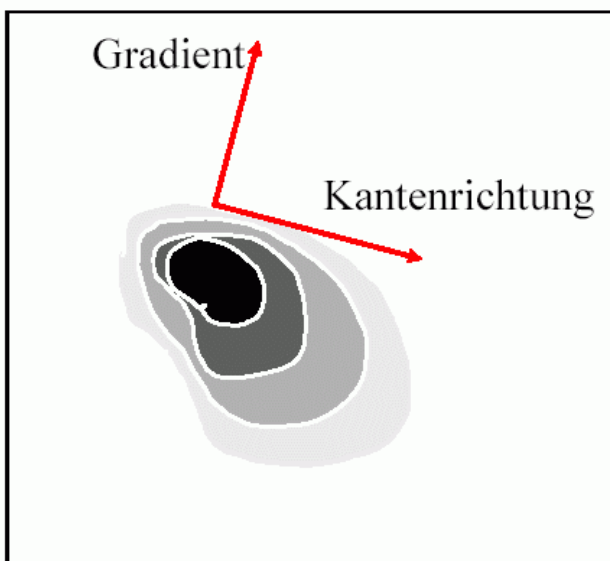
Nachteile:

- bei Verwendung größerer strukturierender Elemente werden kleine Bildbestandteile, die interessant sein könnten, eliminiert
- hoher Rechenaufwand durch Sortieren.

B. Hervorhebung von Grauwertsprüngen ("Kanten")

"Kante": Stelle mit abrupter Änderung der Helligkeit

- math. Beschreibung im kontinuierlichen Fall durch Ableitungen
- Gradient: Richtung der stärksten Änderung



$$|\text{grad } g(x, y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$
$$\psi = \arg\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right)$$

$$(\arg(x, y) = \arctan(y/x))$$

- Kanten werden zur Abgrenzung von Regionen verwendet
- sie verlaufen orthogonal zum Gradienten

Diskrete Gradientenoperatoren: Approximation der Ableitungen durch Differenzen

Roberts-Operator

- Einer der ältesten Operatoren (1965)
- Einfach zu berechnen
- 2x2 Umgebung $h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
- Faltungsmasken:
- Berechnung der Kantenstärke
 $|g(i, j) - g(i+1, j+1)| + |g(i, j+1) - g(i+1, j)|$
- NT: stark rauschabhängig



h_1



h_2



Kantenbild

Prewitt-Operator:

- Approximiert die 1. Ableitung
- Schätzung des Gradienten in 8 Richtungen (für 3x3 Masken)
- Maximumwert über alle Richtungen gibt auch die Gradientenrichtung an

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

Anwendung der Prewitt-Operatoren:



Maximum der 8 Richtungsbilder

Sobel-Operator

- einfacher Detektor für horizontale und vertikale Kanten
- Grundidee: Einfluss der Grauwerte auf die Gradientenbildung nimmt mit dem Abstand vom aktuellen Pixel ab \Rightarrow Koeffizienten werden zum Rand des Operators hin kleiner
- kontinuierliche Idealform: Halbbogen einer Sinusschwingung
- Summe der Koeff. sollte 0 sein (Mittelwert-Invarianz)

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Anwendung:



h_1



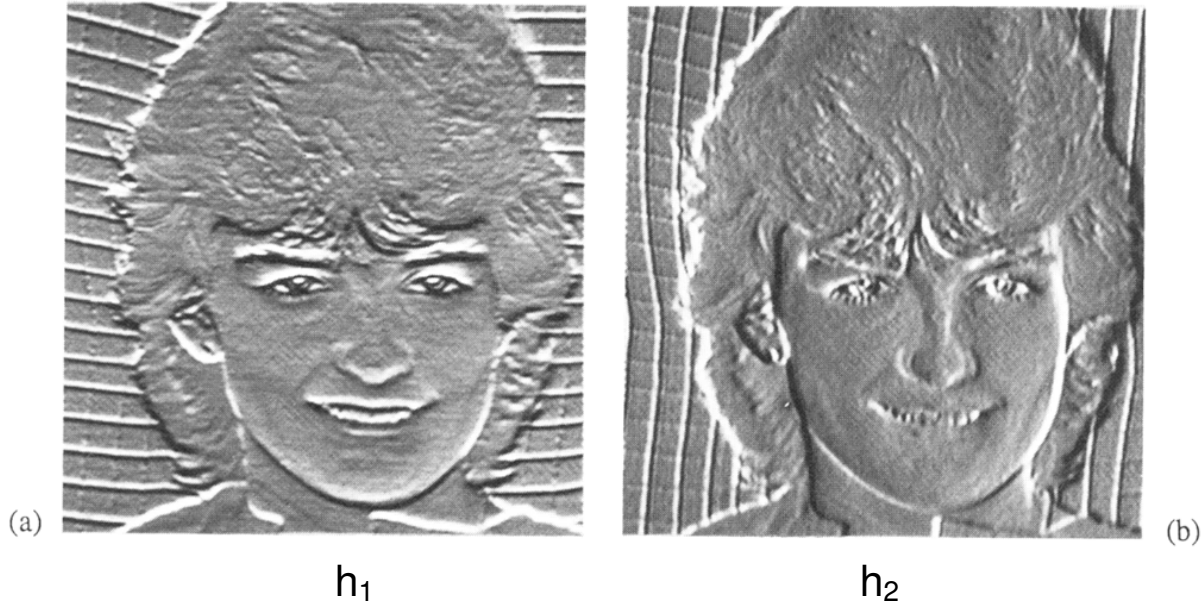
h_2



Kantenbild

$$\|(h_1(x, y), h_2(x, y))\|$$

anderes Beispiel (aus Haberäcker 1991):



Betrag des approximierten Gradienten und Binärbild nach Schwellenwertoperation:



Größere Sobel-Masken:

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

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

-10	-10	0	10	10
-17	-17	0	17	17
-20	-20	0	20	20
-17	-17	0	17	17
-10	-10	0	10	10

10	17	20	17	10
10	17	20	17	10
0	0	0	0	0
-10	-17	-20	-17	-10
-10	-17	-20	-17	-10

-14	-14	-14	-14	0	14	14	14	14
-29	-29	-29	-29	0	29	29	29	29
-44	-44	-44	-44	0	44	44	44	44
-60	-60	-60	-60	0	60	60	60	60
-71	-71	-71	-71	0	71	71	71	71
-60	-60	-60	-60	0	60	60	60	60
-44	-44	-44	-44	0	44	44	44	44
-29	-29	-29	-29	0	29	29	29	29
-14	-14	-14	-14	0	14	14	14	14

14	29	44	60	71	60	44	29	14
14	29	44	60	71	60	44	29	14
14	29	44	60	71	60	44	29	14
14	29	44	60	71	60	44	29	14
0	0	0	0	0	0	0	0	0
-14	-29	-44	-60	-71	-60	-44	-29	-14
-14	-29	-44	-60	-71	-60	-44	-29	-14
-14	-29	-44	-60	-71	-60	-44	-29	-14
-14	-29	-44	-60	-71	-60	-44	-29	-14

(aus Bässmann & Kreys 1998)

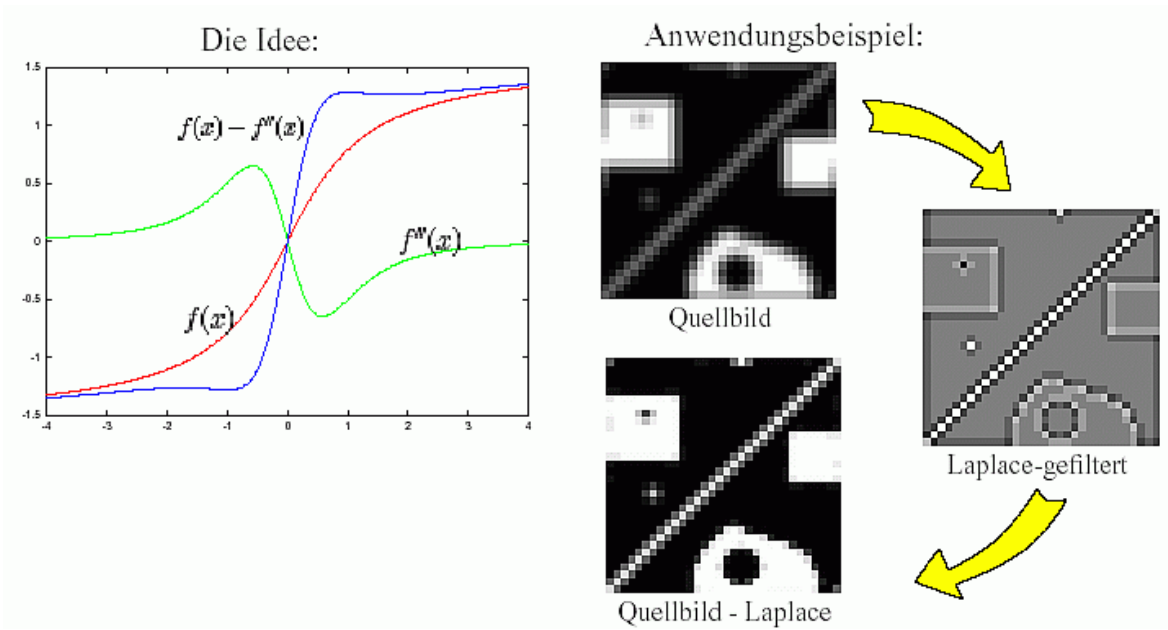
Laplace-Operator

im kontinuierlichen Fall:
Summe der zweiten partiellen Ableitungen

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

- liefert Kantenstärke ohne Richtung
- rotationsunabhängig

Schärfung eines Bildes durch Subtraktion des mit dem Laplace-Filter transformierten Bildes



Originalbild



Laplace Kantenbild



Geschärftes Bild

Diskreter Laplace-Operator

Approximation des kontinuierlichen Laplace-Op. ∇^2

meist verwendet:

$$h = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \text{ oder } h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Anwendung der Laplace-Maske $-h = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$:

<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tbody> <tr><td>1</td><td>1</td><td>3</td><td>5</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>3</td><td>5</td><td>8</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>2</td><td>7</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>2</td><td>5</td><td>9</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>3</td><td>8</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>3</td><td>8</td><td>9</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>2</td><td>6</td><td>8</td><td>10</td><td>10</td><td>10</td></tr> <tr><td>1</td><td>1</td><td>3</td><td>8</td><td>10</td><td>10</td><td>10</td><td>10</td></tr> </tbody> </table> <p>Ausgangsbild</p>	1	1	3	5	10	10	10	10	1	1	3	5	8	10	10	10	1	1	2	7	10	10	10	10	1	1	1	2	5	9	10	10	1	1	1	3	8	10	10	10	1	1	1	3	8	9	10	10	1	1	2	6	8	10	10	10	1	1	3	8	10	10	10	10	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>-2</td><td>1</td><td>-3</td><td>1</td><td>2</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>-1</td><td>-4</td><td>9</td><td>10</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>-2</td><td>-8</td><td>-9</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>-2</td><td>-2</td><td>10</td><td>4</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>-3</td><td>-6</td><td>8</td><td>-2</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>-1</td><td>-3</td><td>3</td><td>2</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table> <p>Ergebnis der Anwendung eines 3×3-Laplace-Operators</p>	0	0	0	0	0	0	0	0	0	-2	1	-3	1	2	0	0	0	-1	-4	9	10	1	0	0	0	0	-2	-8	-9	1	1	0	0	0	-2	-2	10	4	0	0	0	0	-3	-6	8	-2	1	0	0	-1	-3	3	2	3	0	0	0	0	0	0	0	0	0	0
1	1	3	5	10	10	10	10																																																																																																																										
1	1	3	5	8	10	10	10																																																																																																																										
1	1	2	7	10	10	10	10																																																																																																																										
1	1	1	2	5	9	10	10																																																																																																																										
1	1	1	3	8	10	10	10																																																																																																																										
1	1	1	3	8	9	10	10																																																																																																																										
1	1	2	6	8	10	10	10																																																																																																																										
1	1	3	8	10	10	10	10																																																																																																																										
0	0	0	0	0	0	0	0																																																																																																																										
0	-2	1	-3	1	2	0	0																																																																																																																										
0	-1	-4	9	10	1	0	0																																																																																																																										
0	0	-2	-8	-9	1	1	0																																																																																																																										
0	0	-2	-2	10	4	0	0																																																																																																																										
0	0	-3	-6	8	-2	1	0																																																																																																																										
0	-1	-3	3	2	3	0	0																																																																																																																										
0	0	0	0	0	0	0	0																																																																																																																										

(aus Bässmann & Kreys 1998)

Anwendung auf "realistische" Bilder: siehe weiter oben

größere Laplace-Masken:

$$L = \frac{1}{3} \begin{pmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{pmatrix}$$

$$L = \frac{1}{35} \begin{pmatrix} 4 & 1 & 0 & 1 & 4 \\ 1 & -2 & -3 & -2 & 1 \\ 0 & -3 & -4 & -3 & 0 \\ 1 & -2 & -3 & -2 & 1 \\ 4 & 1 & 0 & 1 & 4 \end{pmatrix}$$

$$L = \frac{1}{2345} \begin{pmatrix} & & 259 & 112 & 259 & & \\ & & 259 & -182 & -329 & -182 & 259 \\ & & 112 & -329 & -476 & -329 & 112 \\ & & 259 & -182 & -329 & -182 & 259 \\ & & 259 & 112 & 259 & & \end{pmatrix}$$

(aus Voss & Süße 1991)

zur Bestimmung der 2. Ableitung:

- Problem: stark rauschabhängig
- Abhilfe: Zuerst Glätten, danach Ableitung
- Gaußfilter hat passende Eigenschaften:

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

- σ proportional zur Größe der lokalen Umgebung

- 2. Ableitung des geglätteten Bildes mit Laplace Operator ∇^2 $\nabla^2 [G(x, y, \sigma) * f(x, y)]$

- Reihenfolge kann getauscht werden, aufgrund der Linearität: $[\nabla^2 G(x, y, \sigma)] * f(x, y)$

- Ableitung des Gaußfilters ist unabhängig vom Bildinhalt und kann vorausberechnet werden.

$$h(x, y) = c \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

"*Laplacian of Gaussian*" oder LoG-Operator h

- Operator $h(x,y)$ wird aufgrund seiner Form auch als **Mexican Hat** Operator bezeichnet
- Diskrete Approximation:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Faltungsmasken des LoG werden sehr groß für große σ (40 Pixel bei $\sigma=4$)

Approximation des LoG-Operators durch die Differenz zweier Gaußscher Mittelungskerne mit verschiedenen σ :
 "Difference of Gaussian", DoG.

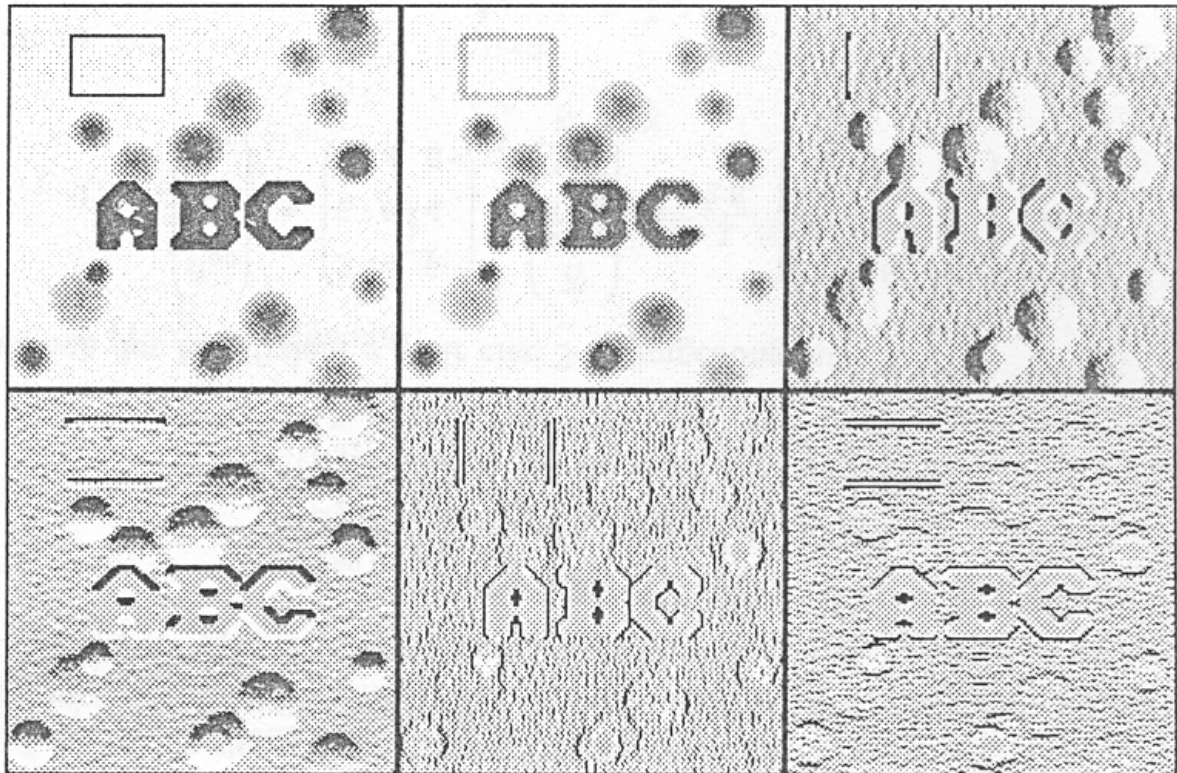
Kantenbestimmung aus 2. Ableitung: "zero crossings" (Nulldurchgänge) zu detektieren

- einfacher Schwellenwert liefert keine befriedigenden Ergebnisse
- Detektion der Nulldurchgänge mit 2×2 -Fenster: innerhalb des Fensters müssen Werte mit beiden Vorzeichen vorkommen
- Verbesserung: nur solche Pixel auswählen, wo die 1. Ableitung groß genug ist
- Menschliche Kantendetektion funktioniert vermutlich nach ähnlichen Prinzipien!

zusammenfassendes Bild zu linearen lokalen Filtern:

Original, Mittelwertoperator, 1. Ableitung (x, y),

2. Ableitung (x, y) (aus Voss & Süße 1991).



Zur Detektion und Verstärkung von Grauwertstufen sind manchmal *nichtlineare Operatoren* den linearen überlegen:

Eingabebild:

1	1	3	5	10	10	10	10
1	1	3	5	8	10	10	10
1	1	2	7	10	10	10	10
1	1	1	2	5	9	10	10
1	1	1	3	8	10	10	10
1	1	1	3	8	9	10	10
1	1	2	6	8	10	10	10
1	1	3	8	10	10	10	10

absolute Differenz zwischen 3×3-Min und Max (Hervorhebung der Grauwertübergänge zwischen dunkler und heller Region):

Min

	1	1	2	5	8	10	
	1	1	1	2	5	9	
	1	1	1	2	5	9	
	1	1	1	2	5	9	
	1	1	1	3	8	9	
	1	1	1	3	8	9	



Max

	3	7	10	10	10	10	
	3	7	10	10	10	10	
	2	7	10	10	10	10	
	1	3	8	10	10	10	
	2	6	8	10	10	10	
	3	8	10	10	10	10	

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

Verfeinerung:

"closest-of-min-and-max-Operator"

ersetzt aktuelles Pixel durch minimalen oder maximalen Grauwert aus der aktuellen Maske (hier 3×3). Ist die Differenz zum Minimum kleiner, wird das Minimum genommen, sonst das Maximum.

Ergebnis bei Anwendung auf obiges Eingabebild:

1	1	1	3	10	10	10	10
1	1	1	2	10	10	10	10
1	1	1	10	10	10	10	10
1	1	1	1	2	10	10	10
1	1	1	1	10	10	10	10
1	1	1	1	10	10	10	10
1	1	1	10	10	10	10	10
1	1	1	10	10	10	10	10

(besser als mit Laplace!)

neues Eingabebild:

1	1	1	3	10	10	10	10
1	1	1	3	3	10	10	10
1	1	3	3	3	10	10	10
1	1	3	3	3	10	10	10
1	1	3	3	3	10	10	10
1	1	1	3	3	10	10	10
1	1	1	1	3	3	10	10
1	1	1	1	1	1	3	10

Iterative Anwendung des 3×3-closest-of-min-and-max-Operators:

1	1	1	1	10	10	10	10
1	1	1	1	3	10	10	10
1	1	3	3	3	10	10	10
1	1	3	3	3	10	10	10
1	1	3	3	3	10	10	10
1	1	1	3	1	10	10	10
1	1	1	1	1	1	10	10
1	1	1	1	1	1	1	10

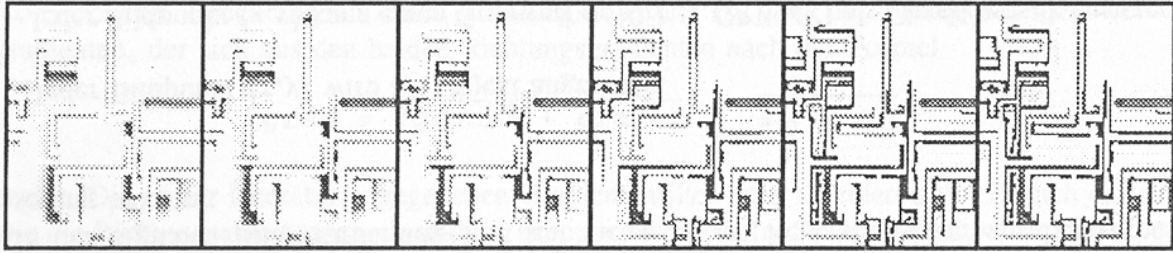
1	1	1	1	10	10	10	10
1	1	1	1	1	10	10	10
1	1	3	3	1	10	10	10
1	1	3	3	3	10	10	10
1	1	3	3	1	10	10	10
1	1	1	3	1	10	10	10
1	1	1	1	1	1	10	10
1	1	1	1	1	1	1	10

1	1	1	1	10	10	10	10
1	1	1	1	1	10	10	10
1	1	3	3	1	10	10	10
1	1	3	3	1	10	10	10
1	1	3	3	1	10	10	10
1	1	1	3	1	10	10	10
1	1	1	1	1	1	10	10
1	1	1	1	1	1	1	10

(Bsp. aus Bässmann & Kreyss 1998)

Iteration stoppen, wenn das Verfahren konvergiert:
"Relaxationsverfahren"

Anwendung eines solchen nichtlin. Relaxationsfilters auf ein reales Bild (nach 1, 4, 16, 64, 256 Schritten):



(aus Voss & Süße 1991)

C. Bildrestauration

Vorverarbeitungsschritt, um bekannte Bildstörungen zu bearbeiten

- je besser die Störung bekannt ist, desto besser kann die Restaurierung gelingen

Gründe für Bildstörungen:

- schlechte, defekte oder verschmutzte optische Linsen
- Nichtlinearität der Sensoren
- Körnigkeit des Filmmaterials
- relative Bewegung zwischen Objekt und Kamera ("Verwackeln", motion blur)
- falsche Brennweite
- atmosphärische Turbulenzen (Fernerkundung, Astronomie)
- Probleme beim Scannen
- Störungen bei der Übermittlung der Bildsignale (TV)
- Mängel am Ausgabegerät

2 Gruppen von Verfahren:

- Deterministische Methoden
Anwendung bei geringem Rauschen und bekannter Störung
- Stochastische Techniken
versuchen die beste Lösung aufgrund bestimmter stoch. Kriterien (meist Methode der kleinsten Quadrate) zu finden.
Schätzung der Störung als erster Schritt.

Schätzung der Störung mittels

- *a priori-Wissen*: Art der Störung bekannt, nur Parameter müssen geschätzt werden (z.B. lineare Bewegung zwischen Objekt und Kamera. Geschwindigkeit und Richtung gesucht.)
- *a posteriori-Wissen*: Interessante Punkte im gestörten Bild suchen (Ecken, Geradenstücke) und ursprüngliches Aussehen schätzen, oder spektrale Untersuchung homogen texturierter Bereiche.

Häufige Störquelle:

aus einem einzelnen schwarzen Punkt wird (aufgrund der Unvollkommenheit der optischen Abb. oder des Sensors) ein mehr oder weniger verschmierter grauer Fleck

- Grauwertfunktion dieses Flecks:
Punktverbreiterungsfunktion (*point spread function*, PSF).
Störung = Faltung mit der PSF.

zusätzlich oft: additives, stochastisches Rauschen.

Somit: 3 Typen von Bildrestaurationsproblemen (Wiederherstellung des Originalbildes F aus dem gestörten Bild G):

- *Entfaltung* bei $G = P * F$, d.h. man kann das Rauschen vernachlässigen, so dass nur die durch die Punktverbreiterungsfunktion P bewirkte Verschmierung rückgängig gemacht werden muss
(- Problem: Faltung ist i.allg. nicht invertierbar! siehe Übung)
- *Rauschbeseitigung* bei $G = F + R$, d.h. die Qualität der Abbildung ist so gut, dass man den Einfluss der Punktverbreiterung vernachlässigen kann
- *Allgemeine Restauration* bei $G = P * F + R$, d.h. beide Einflussfaktoren sind nicht vernachlässigbar.

Störungen, die leicht zu korrigieren sind:

Spezialfälle von Fall 1 mit einfach beschreibbarer Faltung.
Entfaltung durch $F = HG$.

- **Relative Bewegung zwischen Objekt und Kamera:**

$$H(u, v) = \frac{\sin(\pi V T u)}{\pi V u}$$

- V... konstante Geschwindigkeit in x-Rtg.
T... Zeit

- **Falscher Fokus:** $H(u, v) = \frac{J_1(ar)}{ar}$

J ... Bessel Funktion 1. Ordnung

$$r^2 = u^2 + v^2 \quad a \dots \text{Verschiebung}$$

- **Atmosphärische Turbulenzen:** Modell aus empirischen Versuchen $H(u, v) = e^{-c(u^2+v^2)^{5/6}}$

c ... Konstante abhängig von der Art der Turbulenz (aus Experimenten)

(aus Beichel 2002)

Periodisches Störsignal oder Störsignal mit anderweitig bekanntem Spektrum:

- Übergang in den Frequenzraum mittels Fourier-Transformation (globale Transformation)
- dort erscheint Störsignal in Form weniger singulärer Punkte
- Löschen dieser Punkte
- Rücktransformation

Ansatz zur allgemeinen Restauration auf der Grundlage stochastischer Prozesse:

Theorie der *Wiener-Filter*

(siehe z.B. Voss & Süße 1991, S. 102 ff.)

Elimination gestörter Bildzeilen:

- Detektion gestörter Zeilen mittels der Korrelation aufeinanderfolgender Bildzeilen
- Annahme: Korrelationskoeffizient zwischen benachbarten, ungestörten Zeilen in natürlichen Bildern groß

Vorgehensweise:

- Beginne Verarbeitung mit korrekter Bildzeile
- berechne zu je 2 aufeinanderfolgenden Zeilen k und $k+1$ den Korrelationskoeffizienten $r_{k,k+1}$:

$$r_{k,k+1} = \frac{\sum_{j=0}^{R-1} (a(k, j) - \bar{a}(k))(a(k+1, j) - \bar{a}(k+1))}{\sqrt{\sum_{j=0}^{R-1} (a(k, j) - \bar{a}(k))^2 \cdot \sum_{j=0}^{R-1} (a(k+1, j) - \bar{a}(k+1))^2}}$$

(R : Anzahl der Spalten, $a(k, j)$: Grauwert des Bildpunktes in Zeile k und Spalte j , $\bar{a}(k)$: mittlerer Grauwert der Bildzeile k .)

- liegt der Betrag $|r_{k,k+1}|$ über einem vorgegebenen Schwellenwert c , wird die Bildzeile $k+1$ als korrekt angenommen, sonst als fehlerhaft.