Grundlagen Digitaler Medien

Lokale Operatoren / Filter

Dozent: Prof. Dr.-Ing. Sebastian Knorr

Email: knorr@htw-berlin.de

Lernziele

- Unterschied lokale lineare Filter und Punktoperationen
- Was ist ein lineares Filter?
- Welche Eigenschaften haben lineare Filter?
- Was sind nicht-lineare Filter?
- Wie funktioniert das gewichtete Median-Filter?
- Wie könnte man Kanten detektieren?

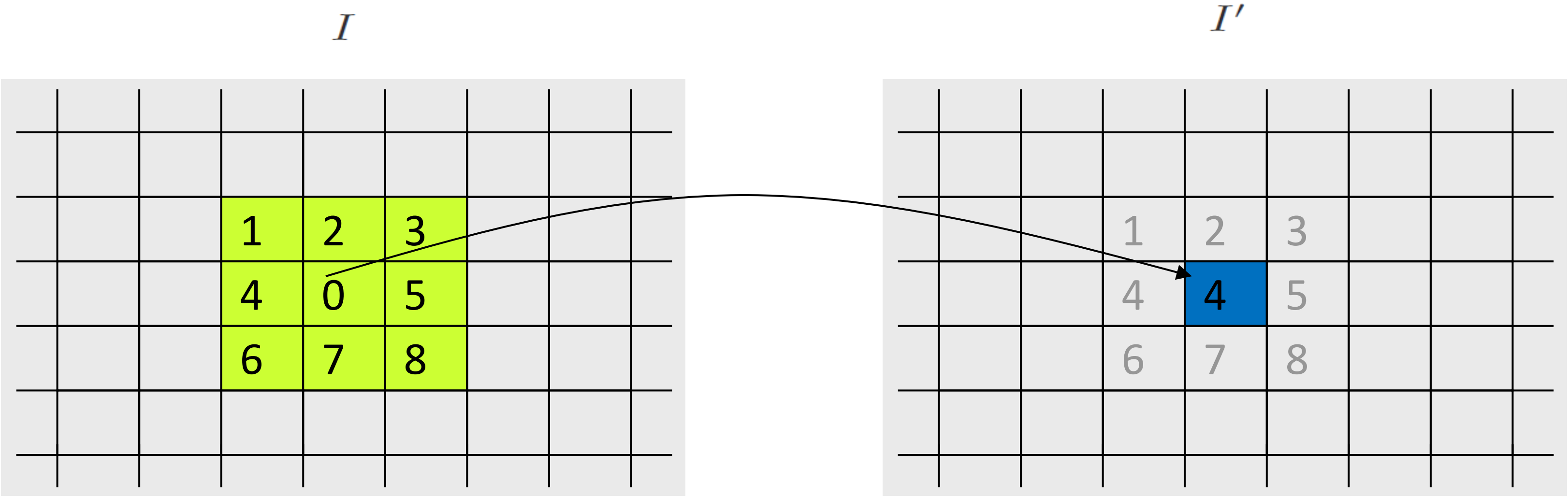
Literatur

- Relevant für dieses VL-Kapitel:
  - Kapitel 6 und 7 aus dem Buch
    W. Burger, M. J. Burge
    „Digitale Bildverarbeitung“
    Springer-Verlag, 2005

Beispiel: Glättung

- Mit Punktoperationen nicht zu erreichen
- Helligkeit, Kontrast (Intensität) bleibt insgesamt gleich
- Geometrie des Bilds bleibt unverändert

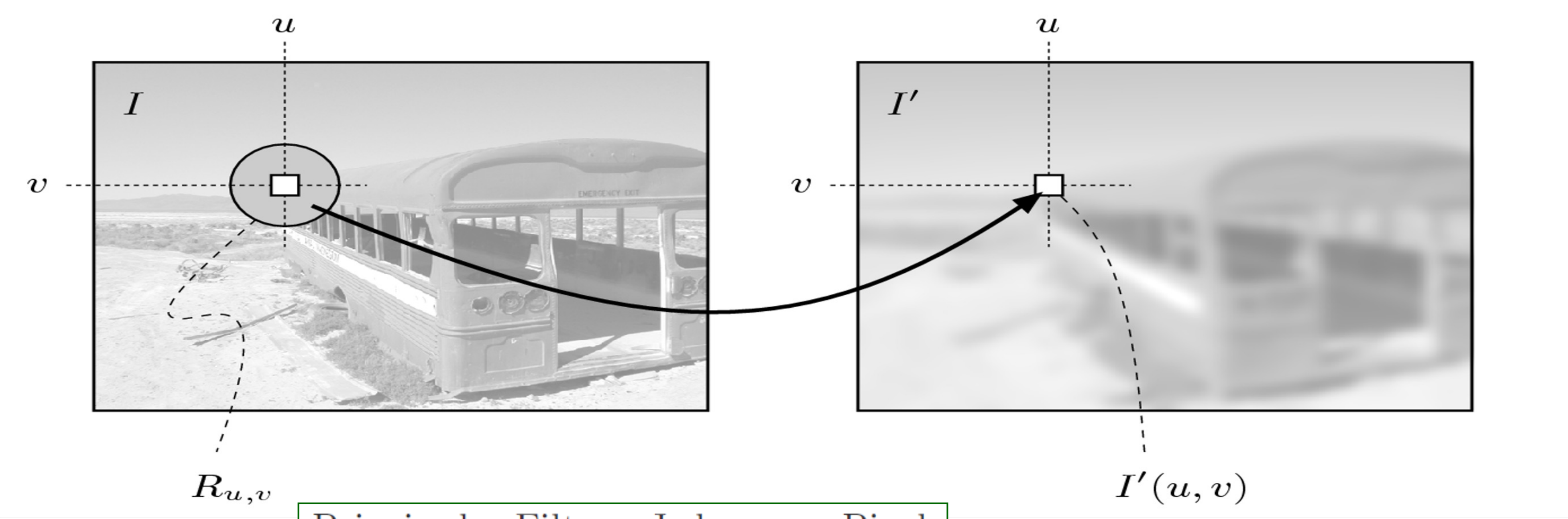
Einfache Glättung durch lokalen (arithm.) Mittelwert



Nachbarpixel $p_1, p_2, \ldots p_8$

$$\boxed{I'(u,v) \leftarrow \frac{p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8}{9}}$$

Prinzip des Filters (im Allgemeinen)



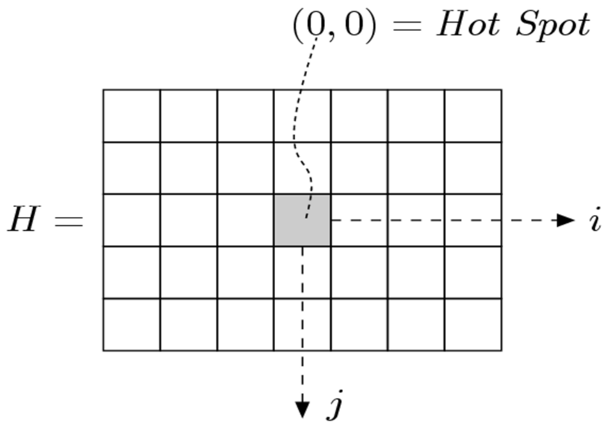
Prinzip des Filters. Jeder neue Pixelwert $I'(u,v)$ wird aus einer zugehörigen Region $R_{u,v}$ von Pixelwerten im ursprünglichen Bild I berechnet.

Filter in 2D

- Größe der Region R?
- Gleiches Gewicht für alle Pixel?
- Kann man mehr als nur glätten?
- Wie kann man Filteroperationen systematisch beschreiben?

Lineares Filter – Filtermatrix

- Neuer Pixelwert ist eine gewichtete Summe (Linearkombination) von Pixelwerten
- Gewichte = Filterkoeffizienten
- Einfache formale Eigenschaften (Orts-/Frequenzraum)



- Filtermatrix = zweidimensionale Funktion von Gewichten (Koeffizienten), bestimmt die Region *R*
- Filterfunktion ist unendlich, alle Werte außerhalb der Matrix sind null.
- „Hot Spot" = Koordinatenursprung der Filterfunktion (typ. zentriert)

Lokaler Mittelwert (*revisited*)

$$\begin{aligned} I'(u,v) \leftarrow \tfrac{1}{9}\, [\; & I(u-1,v-1) + I(u,v-1) + I(u+1,v-1) + \\ & I(u-1,v) \quad\;\; + I(u,v) \quad\;\; + I(u+1,v) \quad\;\; + \\ & I(u-1,v+1) + I(u,v+1) + I(u+1,v+1) \;] \end{aligned}$$

Zugehörige Filterfunktion (Filtermatrix):

$$H(i,j) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

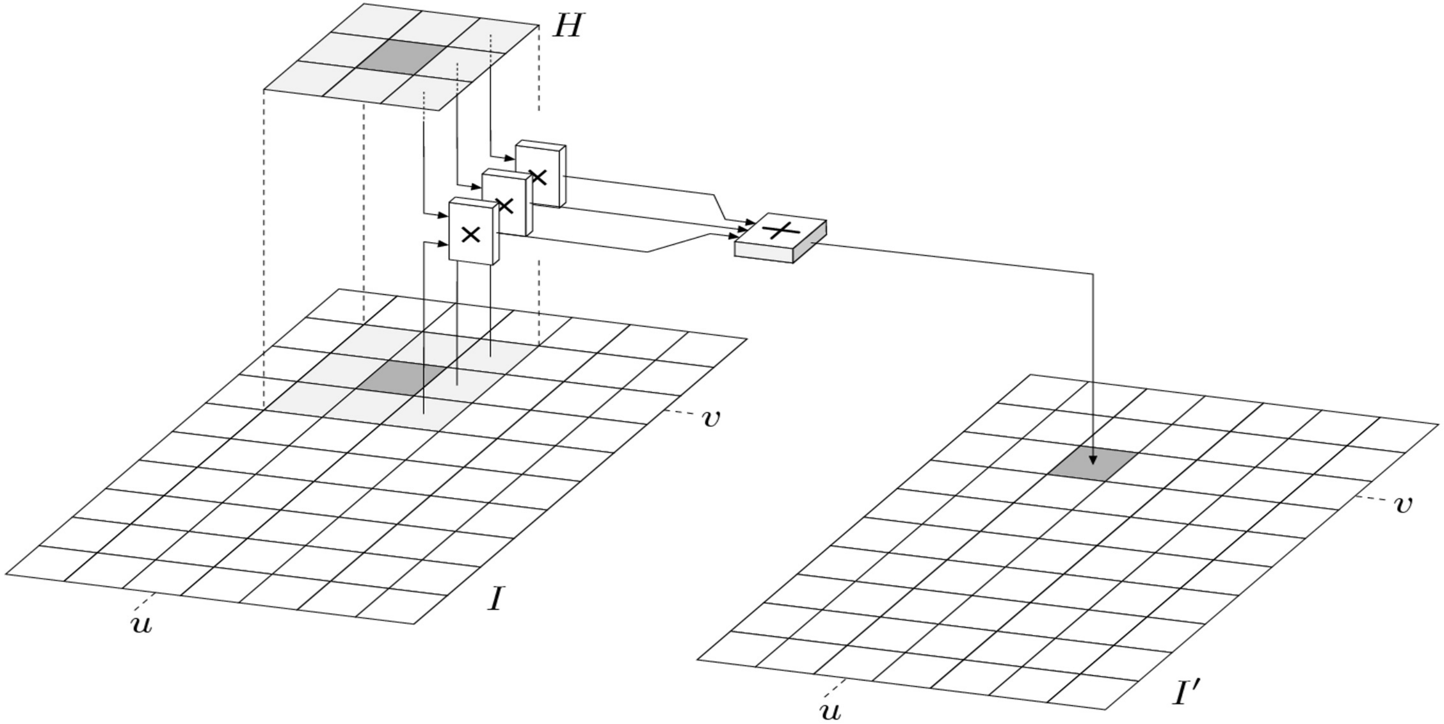
Filteroperation:

$$\boxed{I'(u,v) \leftarrow \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} I(u+i, v+j) \cdot H(i,j)}$$

für jedes Pixel (*u*,*v*)

Allgemeines, lineares 3x3-Filter

$$I'(u,v) \leftarrow \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} I(u+i, v+j) \cdot H(i,j)$$



Noch ein 3x3-Glättungsfilter (Beispiel)

Gauß-förmiges Glättungsfilter: Gewichtung der Pixel ist abhängig von der Distanz zum Hot Spot.

$$H(i,j) = \begin{bmatrix} 0.075 & 0.125 & 0.075 \\ 0.125 & \underline{0.200} & 0.125 \\ 0.075 & 0.125 & 0.075 \end{bmatrix}$$

Radiale Gaußfunktion

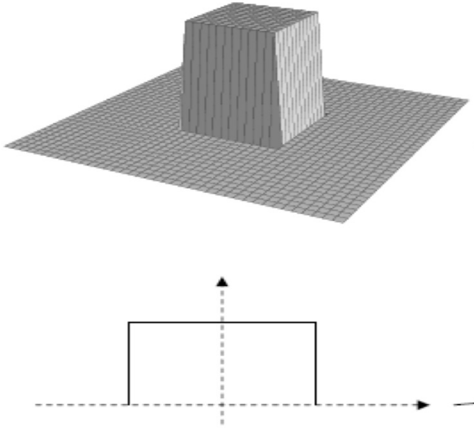
$$G_\sigma(r) = e^{-\frac{r^2}{2\sigma^2}}$$

- Filterkoeffizienten müssen nicht ganzzahlig sein!
- Bei Glättungsfiltern wird üblicherweise die Summe der Koeffizienten auf 1 normiert.
- Effiziente Berechnung mit ganzzahligen Koeffizienten und einem gemeinsamen Skalierungsfaktor, z.B:

$$H(i,j) = \begin{bmatrix} 0.075 & 0.125 & 0.075 \\ 0.125 & \underline{0.200} & 0.125 \\ 0.075 & 0.125 & 0.075 \end{bmatrix} = \frac{1}{40} \begin{bmatrix} 3 & 5 & 3 \\ 5 & \underline{8} & 5 \\ 3 & 5 & 3 \end{bmatrix}$$

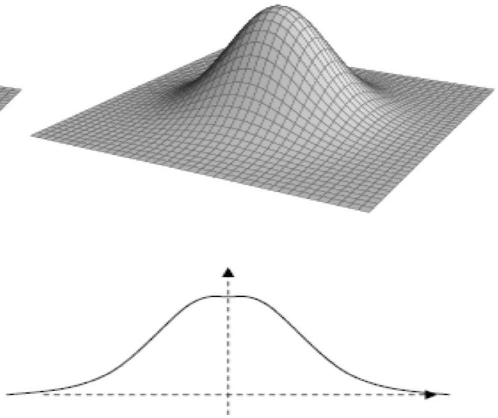
Beispiele für lineare 2D-Filter

„Box“-Filter



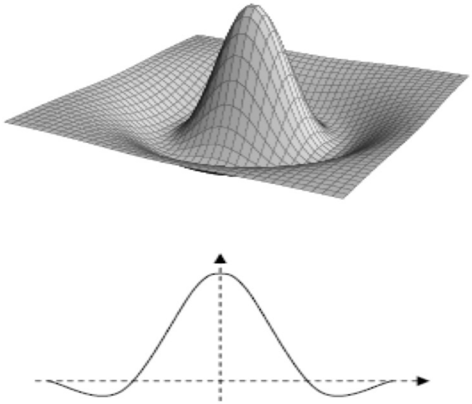
0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Gauß-Filter



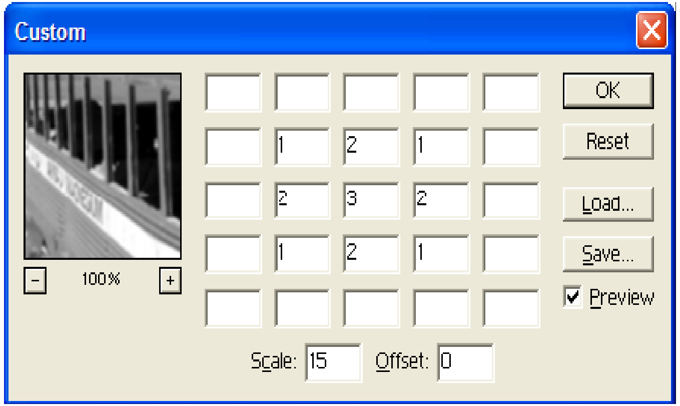
0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Laplace (Mexican Hat)



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

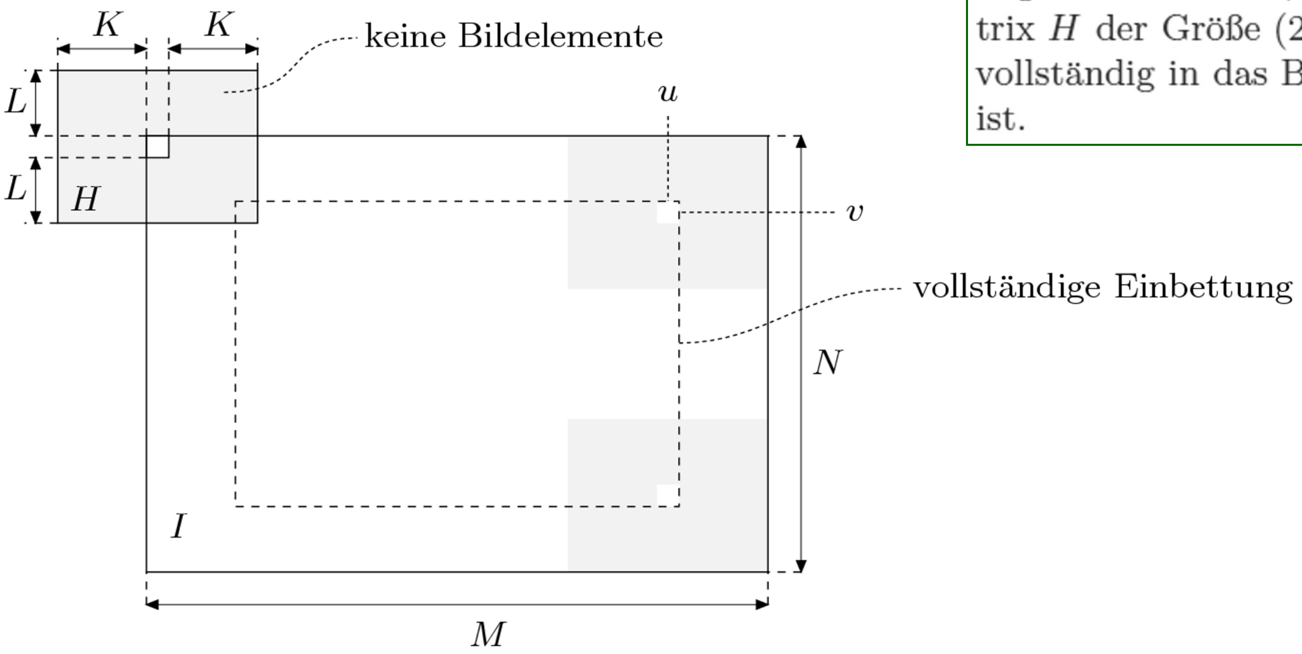
Photoshop „Custom“ Filter



- Lineares Filter mit max. Größe 5x5.
- Koeffizienten in Leerfeldern sind null.
- Ganzzahlige Koeffizienten (+/-) mit gemeinsamem Skalierungsfaktor (*Scale*)
- Zusätzlicher Offset, um negative Ergebniswerte zu vermeiden.
- Autom. Clamping (zB. auf 0…255)

$$I'(u,v) \leftarrow Offset + \frac{1}{Scale} \sum_{i=-2}^{i=2} \sum_{j=-2}^{j=2} I(u+i, v+j) \cdot H(i,j)$$

Berechnung von Filtern – Randproblem



Randproblem bei Filtern. Die Filteroperation kann problemlos nur dort angewandt werden, wo die Filtermatrix H der Größe $(2K+1) \times (2L+1)$ vollständig in das Bild eingebettet I ist.

Randproblem – Lösungsvarianten

- Randpixel im neuen Bild nicht füllen (z.B. auf schwarz setzen)
- Randpixel unverändert übernehmen (ohne Filterung).

- Bis zum Rand filtern und die Pixel außerhalb des Bildbereichs …

als konstant annehmen (z.B. 128) | durch angrenzende Bildpixel füllen | als zyklisch annehmen (in beiden Dimensionen)

(a) (b) (c)

Filterberechnung – Wertebereiche

- Welche Werte können entstehen (max/min)?

Bsp.: 8-Bit Grauwertbild (0…255), kein Clamping!

$$H(i,j) = \begin{bmatrix} -1 & -2 & 0 \\ -2 & \underline{0} & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

min = __________ ?

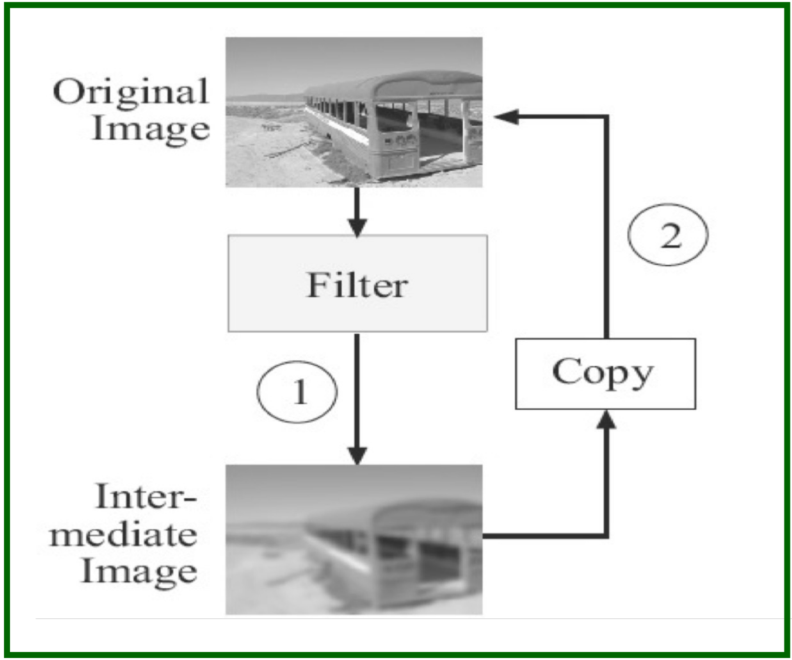
max = __________ ?

- Was tun mit zu großen Ergebniswerten (clamping?)
- Filterkoeffizienten können negativ sein!
- Was tun mit negativen Ergebnissen?
- Datenformat für Zwischenergebnisse (float, signed integer)?

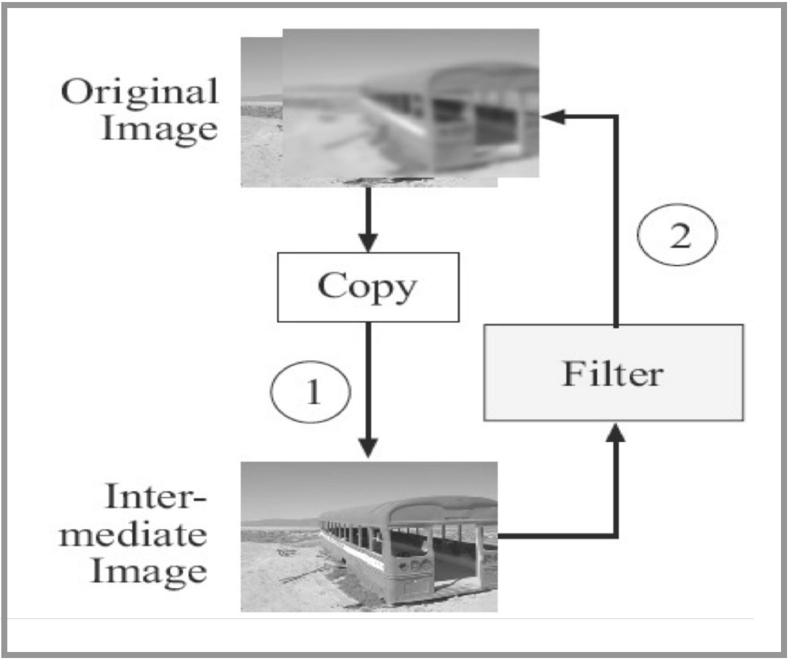
Filterberechnung in der Praxis

- „**In-Place**“ Berechnung ist **nicht** möglich!
- Zusätzliche **Kopie** des Bilds ist notwendig.

Variante A



Variante B



Lineare Filter – „Faltung“ (Convolution)

- Alle linearen Filter sind durch ihre zugehörige Filtermatrix *H* vollständig spezifiziert!
- Lineare Filter entsprechen i.A. einer linearen Faltungsoperation

$$\boxed{I'(u,v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u-i, v-j) \cdot H(i,j)}$$

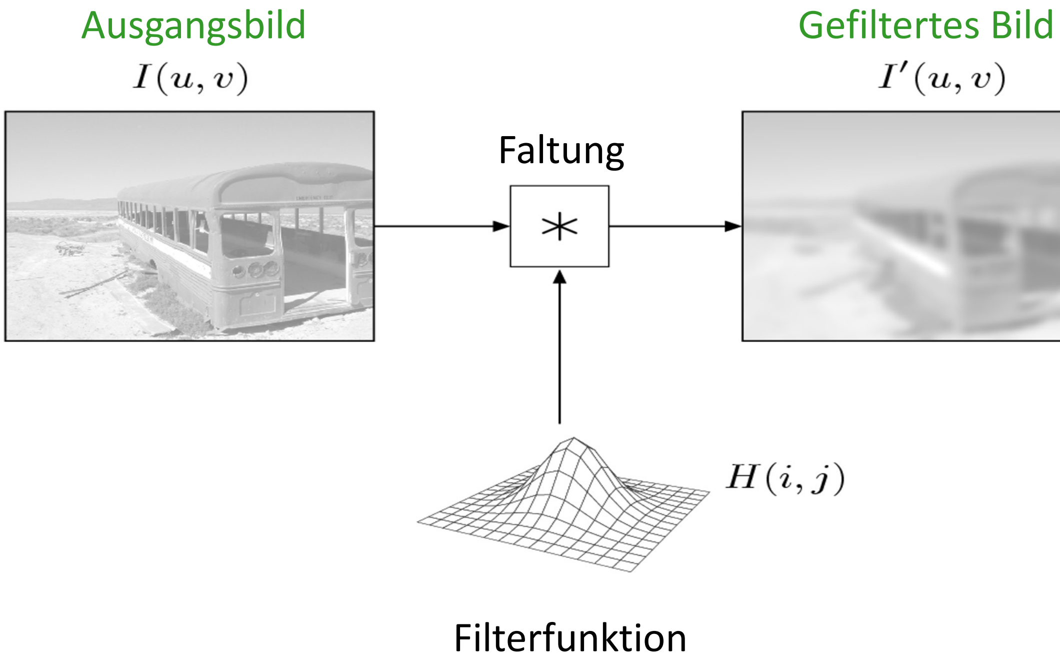
Funktion *I*

Funktion *H*

$$\boxed{I' = I * H}$$

Faltungsoperator

Faltung als „Blackbox“-Operation



Faltung – wichtige formale Eigenschaften

Die lineare Faltung ist …

Kommutativ $\boxed{I * H = H * I}$

Assoziativ $\boxed{A * (B * C) = (A * B) * C}$

Linear $\boxed{(a \cdot I) * H \quad = \quad I * (a \cdot H) \quad = \quad a \cdot (I * H)}$ (Skalar *a*)

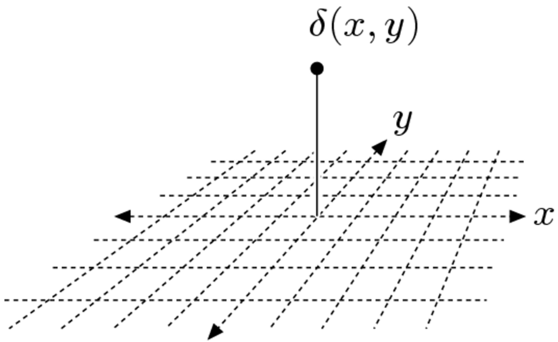
$$(I_1+I_2)*H=(I_1*H)+(I_2*H)$$

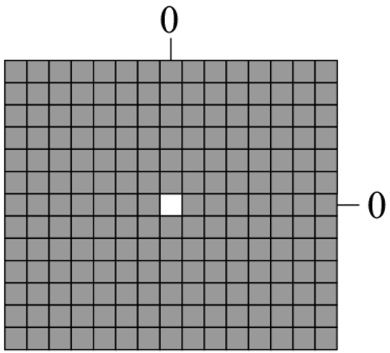
Impulsantwort – *Dirac*-Funktion

Die Dirac- oder Impulsfunktion ist das „neutrale Element“ der linearen Faltungsoperation, d.h.

$$\boxed{I * \delta = I}$$

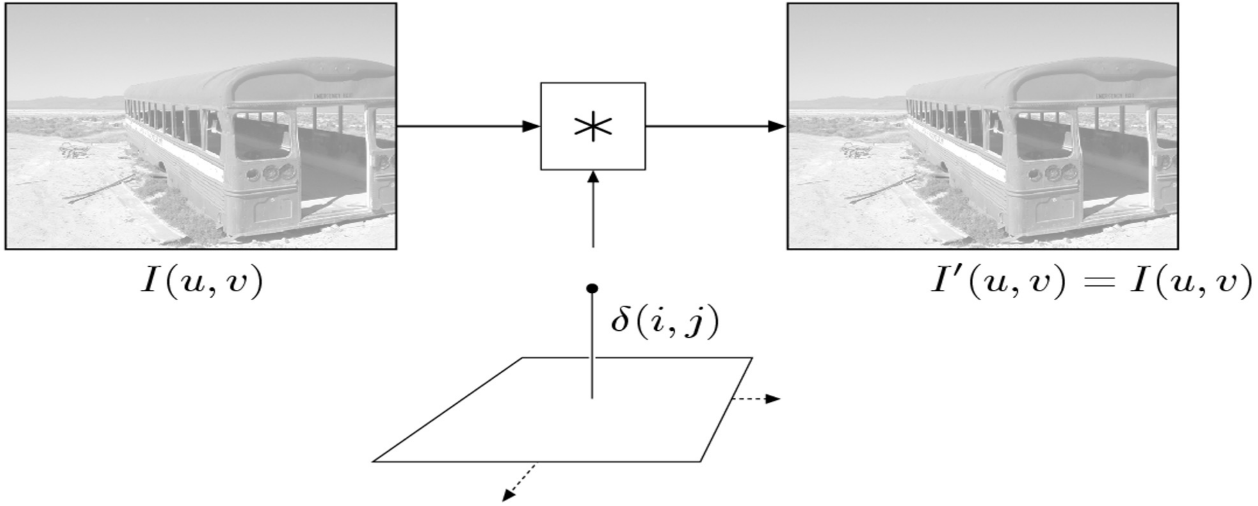
Diskrete 2D-Impulsfunktion: $$\delta(i,j) = \begin{cases} 1 & \text{für } i = j = 0 \\ 0 & \text{sonst.} \end{cases}$$





Faltung mit der Impulsfunktion

Das Ergebnis einer linearen Faltung mit der Impulsfunktion $\delta()$ ist wieder das ursprüngliche Bild $I(u,v)$.



x/y-Separierbarkeit

- Zerlegung zweidimensionaler Filter in eine Folge eindimensionaler Filter
- Effiziente Implementierung großer Filter möglich

Beispiel:

2 eindimensionale Filter: $H_x = \begin{bmatrix} 1 & 1 & \underline{1} & 1 & 1 \end{bmatrix}$ $H_y = \begin{bmatrix} 1 \\ \underline{1} \\ 1 \end{bmatrix}$

Anwendung beider Filter:

$$\boxed{I' \leftarrow (I * H_x) * H_y = I * \underbrace{(H_x * H_y)}_{H_{xy}}}$$

Ergebnis:
aus zwei kleineren Filtern kann ein größeres Filter erzeugt werden.
Welche Einsparung (an Rechenaufwand) ergibt sich?

$$H_{xy} = H_x * H_y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \underline{1} & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

x/y-Separierbarkeit – noch ein Beispiel

$$H = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{H_1} * \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{H_2} = \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_{H}$$

Die Berechnung einer Filteroperation mit H ist auf 2 Wegen möglich:

Variante 1:	$I * H$	**9** Operationen pro Bildpixel
Variante 2:	$(I * H_1) * H_2$	3+3=**6** Operationen

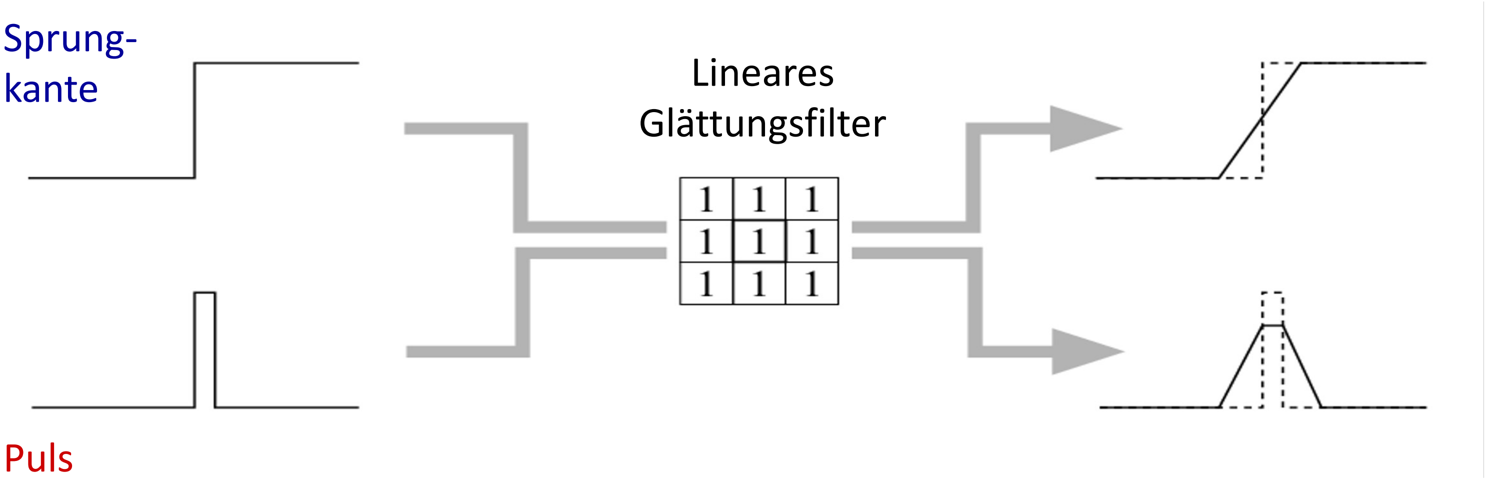
Vorteil ist bei größeren Filtern signifikant:

5x5	25	10
7x7	49	14
25x25	625	50

Lineare, homogene Filter

- Einfache Beschreibung
- Umfangreiche mathematische Theorie
- Praktischer Einsatz:
  - Glättung, Konturfilter (später)
  - Linearität durch beschränkten Wertebereich nicht gesichert
  - oft in Kombination mit nichtlinearen Operationen

Glättungsfilter: Anwendung auf Sprungkante/Puls

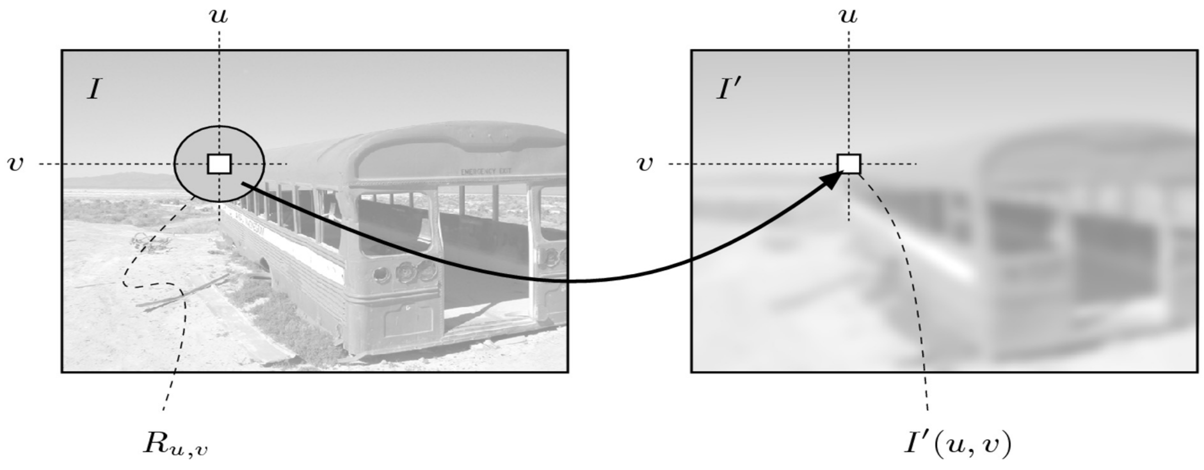


Einfache *nicht*-lineare Filter: *Minimum- /Maximum*-Filter

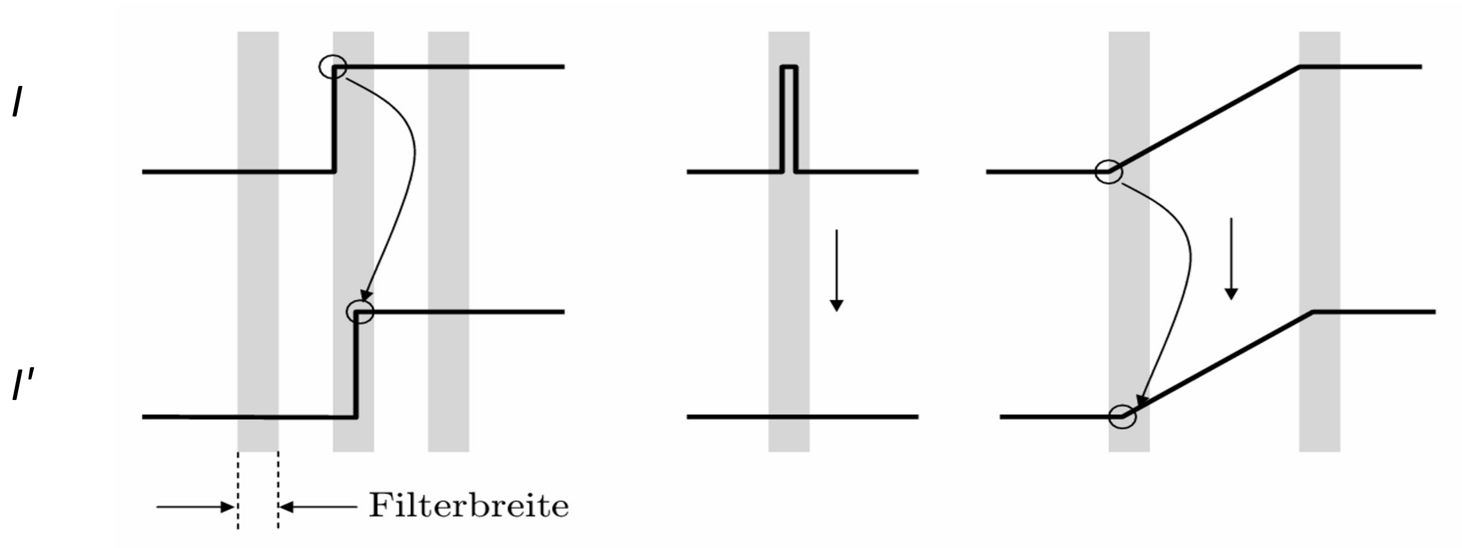
Neuer Pixelwert ist das Minimum / Maximum innerhalb der vorgegebenen Filterregion R:

$$\boxed{I'(u,v) \leftarrow \min\{I(u+i,v+j) \mid (i,j) \in R\} = \min\left(R_{u,v}\right)}$$

$$\boxed{I'(u,v) \leftarrow \max\{I(u+i,v+j) \mid (i,j) \in R\} = \max\left(R_{u,v}\right)}$$



Minimum-Filter



Minimum-Filter

Salt & Pepper Noise

Min-Filter

Maximum-Filter

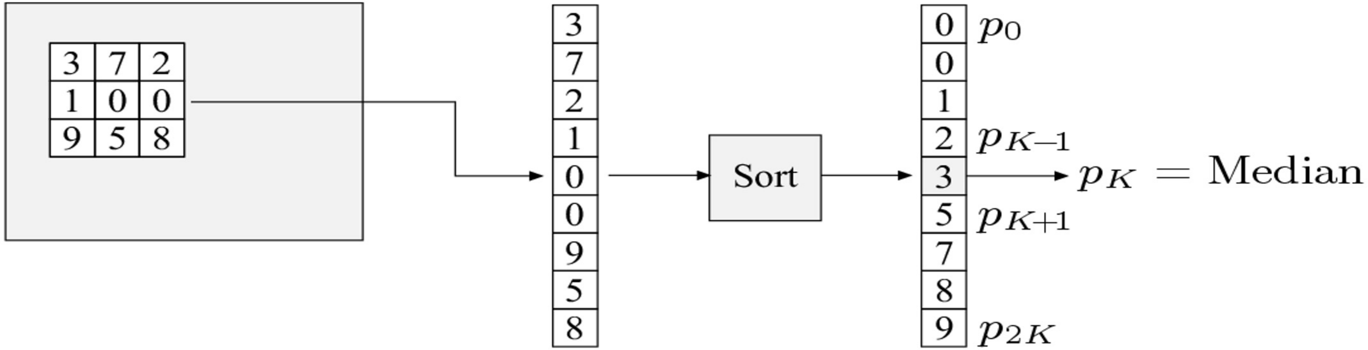
Salt & Pepper Noise

Max-Filter

Medianfilter

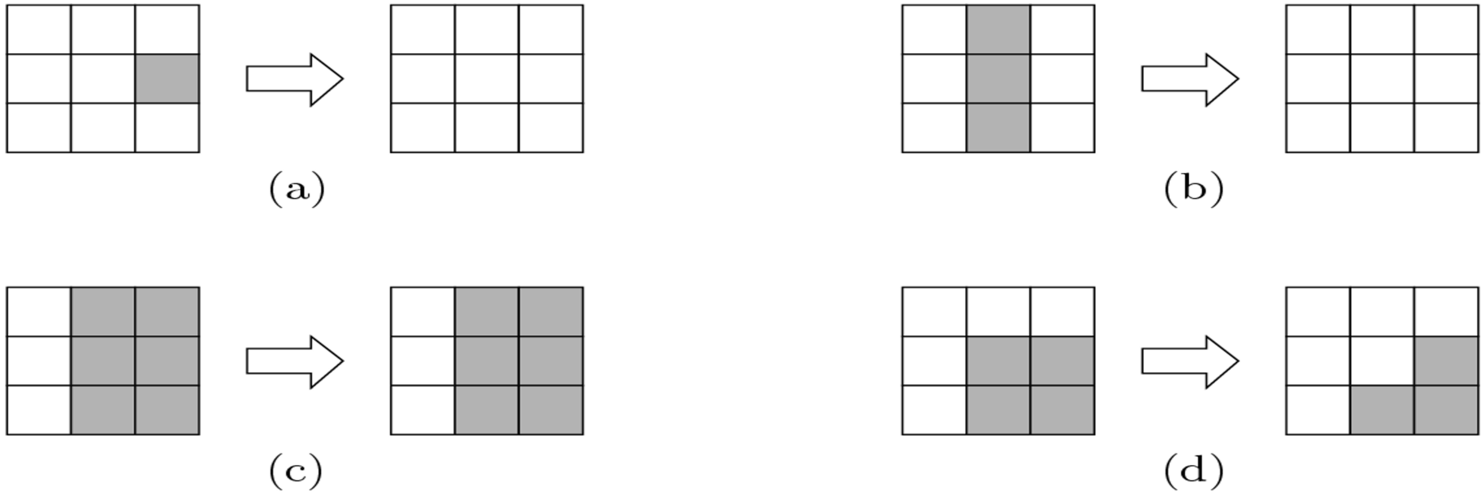
$$\boxed{I'(u,v) \leftarrow \mathrm{median}(R_{u,v})}$$

$$\mathrm{median}(p_0, p_1, \ldots, p_K, \ldots, p_{2K}) = p_K$$



Auswirkungen eines 3x3 Medianfilters

Randpixel: Wert vom direkten Nachbarn im Bild übernehmen

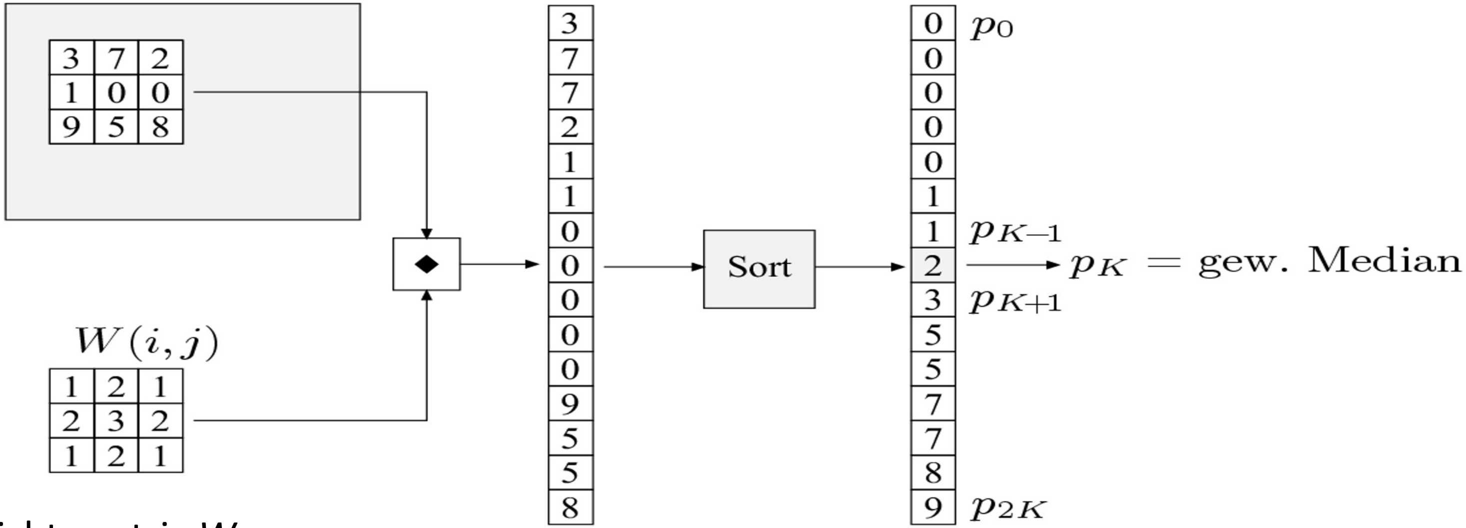


Lineares Glättungsfilter vs. Medianfilter

Lin. Glättungsfilter (3x3 Box)

3x3 Median Filter

Gewichtetes Medianfilter



Gewichtsmatrix *W*

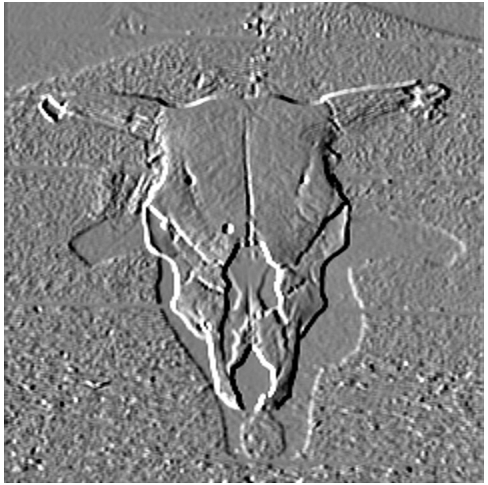
Kantenfilter

Prewitt-Filter in x-Richtung

*

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

=



Kantenfilter

Prewitt-Filter in y-Richtung

$$*$$

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

$$=$$

Bildgradienten

Kante

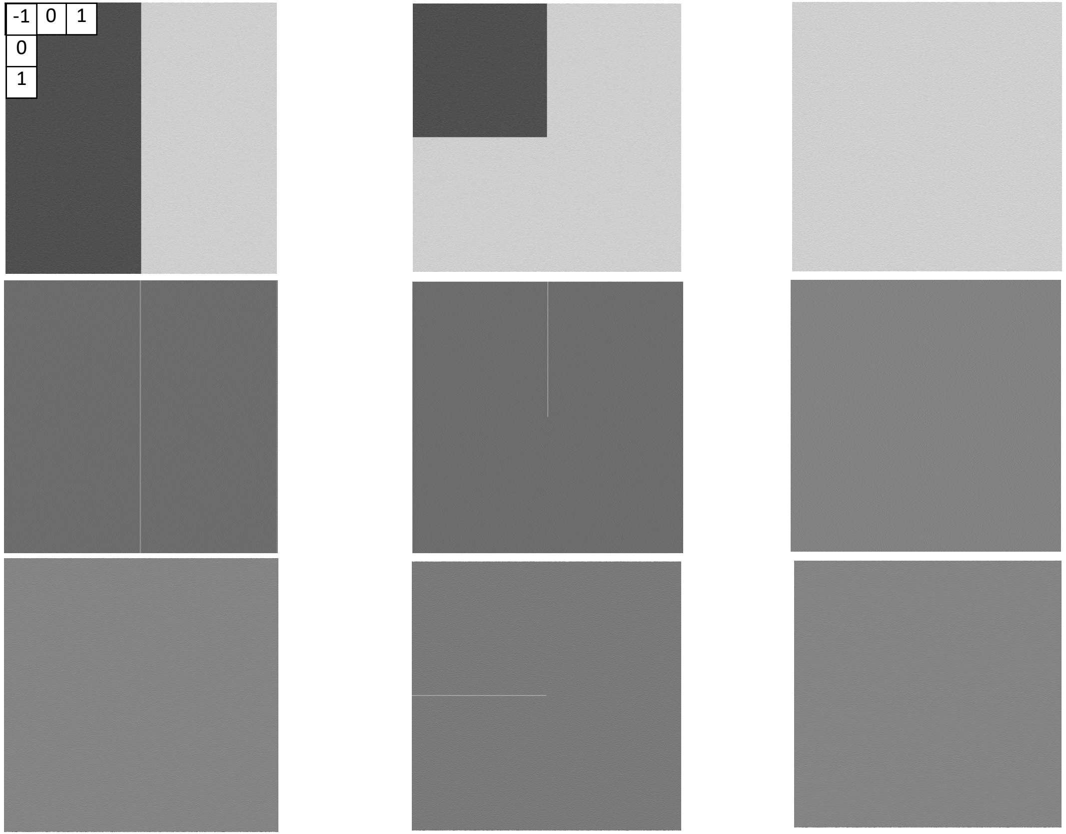
Ecke

homogene Fläche

Original Patch I(x,y)

X-Gradient $I_x(x,y)$

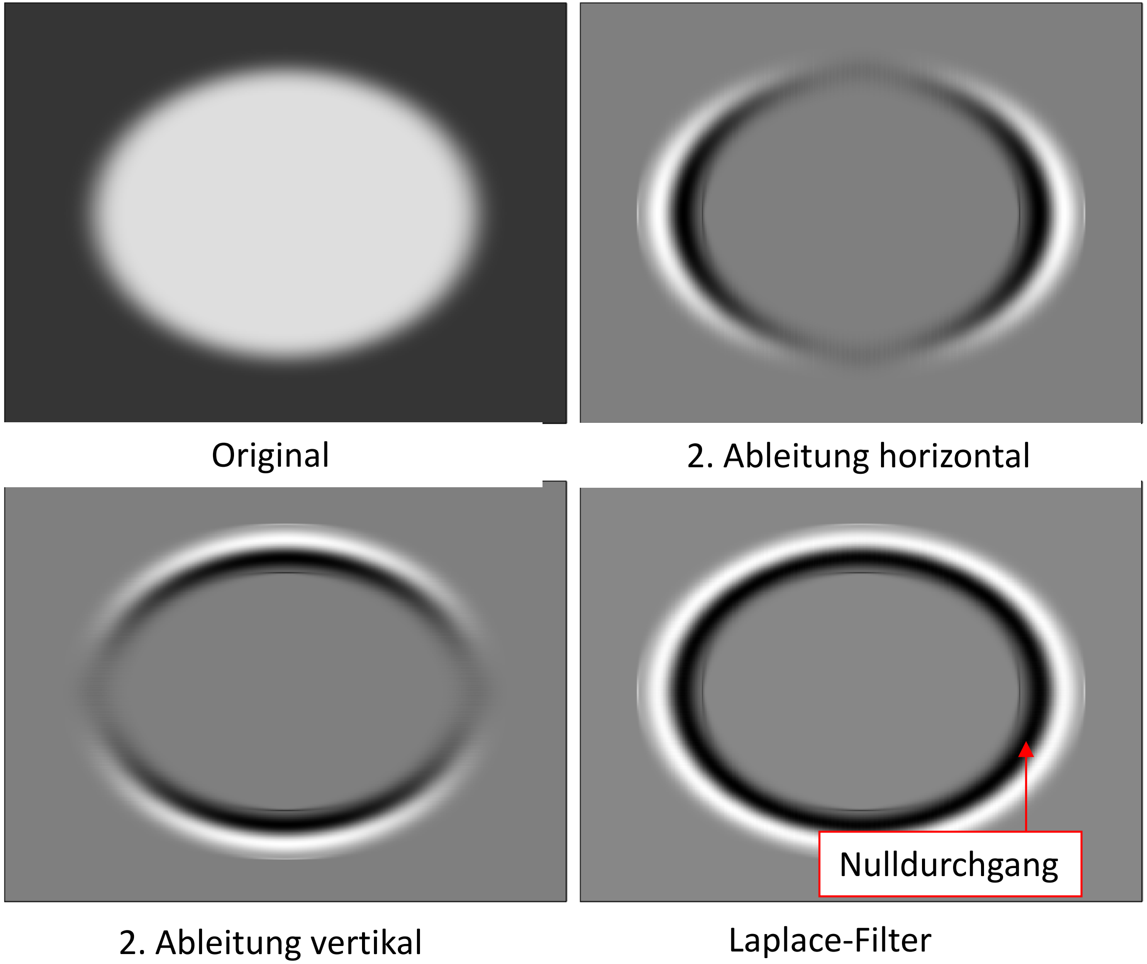
y-Gradient $I_y(x,y)$



Laplace Filter

(Schärfung)

$$H^L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -\underline{4} & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Unscharf maskieren

$-$

$=$

Unscharf maskieren

+

=

Unscharf maskieren

- Geglättete Version des Bilds wird vom Originalbild subtrahiert → „Maske“ *M*

$$\boxed{M = I - (I * \tilde{H})}$$

Glättungsfilter $H^{\sim}$: Gauß-Filter mit Radius σ

- Maske *M* wird zum Originalbild addiert, gewichtet mit einem Schärfungsfaktor *a*

$$\begin{aligned} I' &= I + a \cdot M \\ &= I + a \cdot (I - (I * \tilde{H})) \\ &= (1 + a) \cdot I - a \cdot (I * \tilde{H}) \end{aligned}$$

(*a* = Stärke der Schärfung)

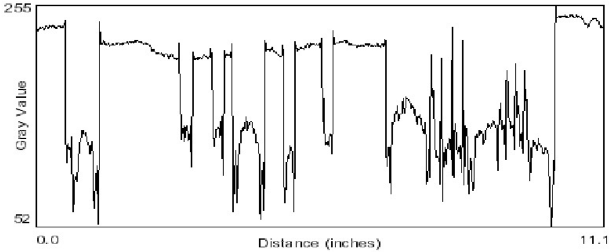
Ursprünglich in der analogen Filmtechnik entwickelt – Glättung (Unschärfe) und Subtraktion können relativ einfach auf optischem Weg realisiert werden.

Unscharf maskieren

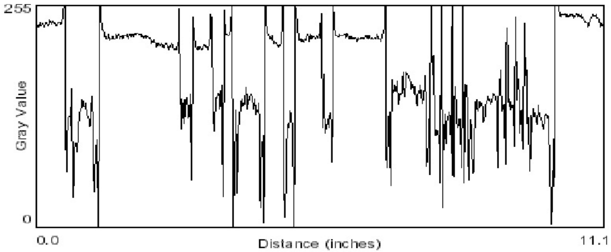
Ähnlich zu Schärfung mit Laplace-Filter

Bessere Kontrolle über Schärfungsstärke und Schärfungsbereich

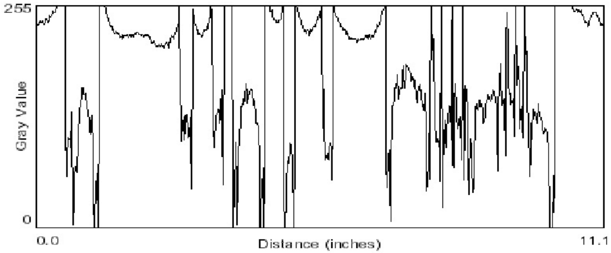
Sehr verbreitete Technik



Original



$\sigma = 2.5$

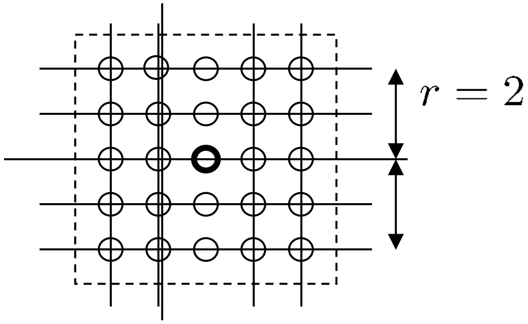


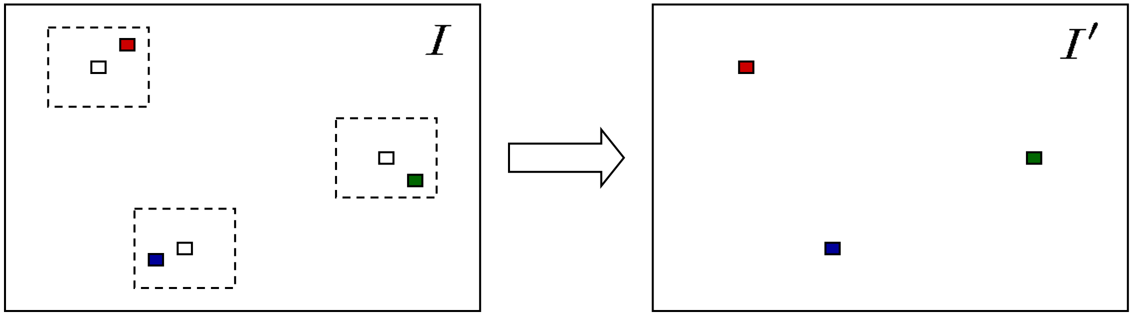
$\sigma = 10.0$

Nichthomogene Filter

Filteroperation ist nicht identisch für alle Bildkoordinaten!

Beispiel: „Jitter"-Filter – zufällige Wahl eines Quellpixels innerhalb der Filterregion



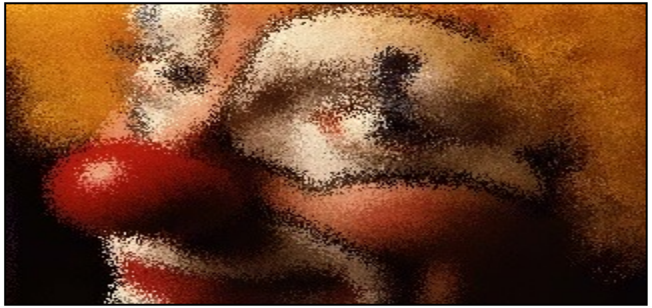


Bsp. „Jitter“-Filter

Original

r=3

[-*r*,*r*]
Uniform
Distribution



r=5

r=10

Lernziele

- Unterschied lokale lineare Filter und Punktoperationen
- Was ist ein lineares Filter?
- Welche Eigenschaften haben lineare Filter?
- Was sind nicht-lineare Filter?
- Wie funktioniert das gewichtete Median-Filter?
- Wie könnte man Kanten detektieren?

Vielen Dank!

Sebastian Knorr