



08.11.2018

Bildverbesserung I



Bildverbesserung I

08.11.2018

Einfache Bildverbesserung mit zwei Zielen:

- Korrektur von Aufnahme Fehlern
- Hervorheben / Sichtbarmachen von schlecht sichtbaren Informationen.



Helligkeitskorrekturen (lokal)

08.11.2018

Positionsabhängige Helligkeitskorrekturen

- Wie kann man additive und multiplikative Fehler einer Aufnahmeeinrichtung korrigieren, bei der nicht alle Pixel gleich behandelt werden?

Wir bezeichnen das ideale, nicht gestörte Bild mit

$f_{ideal}(x, y)$, die Störung mit $e_{add}(x, y)$ bzw. $e_{mul}(x, y)$ und das gestörte Bild mit $f_{real}(x, y)$.



Helligkeitskorrekturen (lokal)

08.11.2018

- Dann kann man schreiben:

$$f_{real}(x, y) = f_{ideal}(x, y) + e_{add}(x, y) \quad ,$$

bzw.

$$f_{real}(x, y) = f_{ideal}(x, y) * e_{mul}(x, y)$$

oder wenn beide Arten von Fehlern gleichzeitig auftreten:

$$f_{real}(x, y) = f_{ideal}(x, y) * e_{mul}(x, y) + e_{add}(x, y)$$



Helligkeitskorrekturen (lokal)

08.11.2018

- Aufgabe: Wie werden wohl die unbekannten Funktionen $e_{mul}(x, y)$ und $e_{add}(x, y)$ bestimmt?



Helligkeitskorrekturen (lokal)

- Positionsunabhängige Grauwerttransformationen T

– ordnen global einem bestimmten Grauwert einen neuen Grauwert zu: $f' = T(f)$

- Ausgleich von Nichtlinearität des Detektors
- Invertierung: Negativ -> Positiv
- Kontrastverstärkung in bestimmten Helligkeitsbereichen (geht zwangsläufig mit Kontrastabschwächung in anderen Bereichen einher, außer wenn Bereich vorher nicht ganz ausgenutzt war.)
- Kontrastabschwächung zur Beseitigung von Störungen in bestimmten Helligkeitsbereichen.



Helligkeitskorrekturen (lokal)

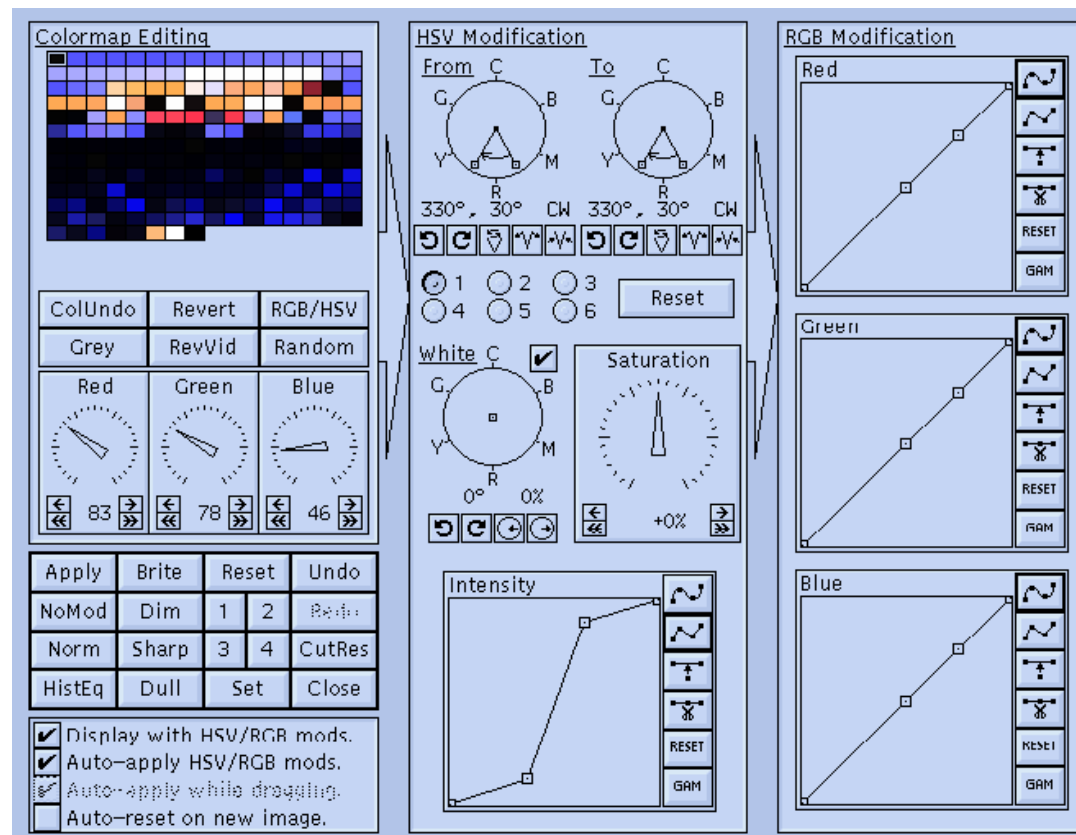
08.11.2018

- Positionsunabhängige
Grauwerttransformationen
 - Praktisch können diese Transformationen mit Look-Up Tabellen durchgeführt werden. Diese sind Speicher, die von den ursprünglichen Werten adressiert werden, und die an diesen Adressen gespeicherten Funktionswerte ausgegeben werden.

Helligkeitskorrekturen (lokal)

08.11.2018

Farbmanipulationen mit xv





Helligkeitskorrekturen (lokal)

08.11.2018

- Welche Funktion wird in die LUT eingetragen?
 - Kenntnis der Funktion z.B. durch Kenntnis der Bildschirm-Response (Gamma-Korrektur):
 - $f_{\text{dargestellt}} = f^{\gamma}$, wobei $\gamma = 2.2$.
 - Warum benutzt man solche Bildschirme?

Damit Fehler sich bei kleinen Werten weniger stark auswirken, was wegen der log. Kennlinie des Auges erwünscht ist. (Fernsehen). Das bedeutet aber z.B. bei der Mittelwertbildung von zwei Werten (z.B. für die Bildverkleinerung), daß auf Originalwerte zurückgerechnet werden muß.



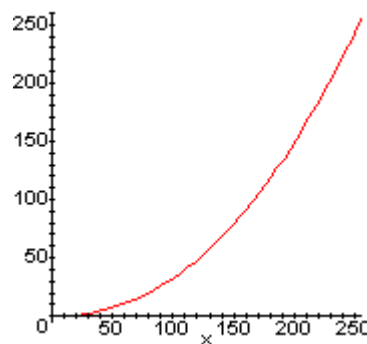
Helligkeitskorrekturen (lokal)

08.11.2018

- Wie wird die Funktion berechnet, die für die γ -Korrektur in die LUT eingetragen werden muß?

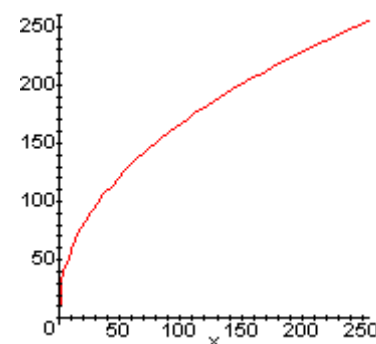
– Bildschirm-Response (Gamma-Korrektur):

$$f_{\text{dargestellt}} = f^\gamma, \text{ wird kompensiert durch } f = f_{\text{ursprünglich}}^{\frac{1}{\gamma}}$$



Bildschirm

wird
kompensiert
durch

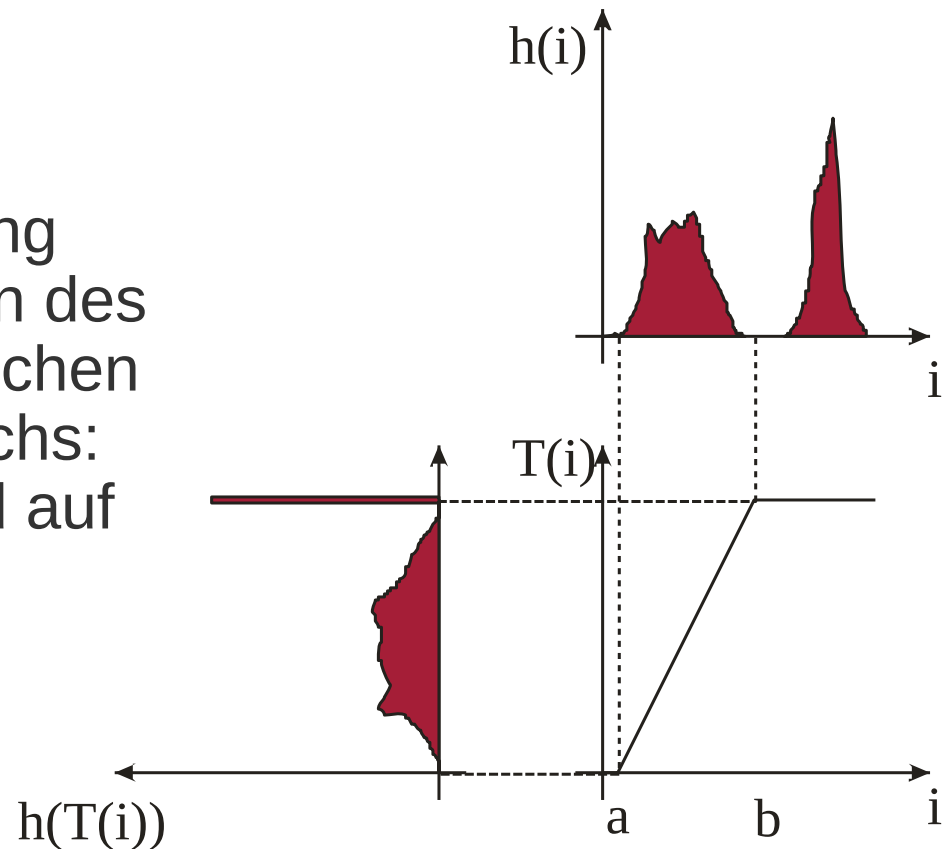


Lookuptable

Helligkeitskorrekturen (lokal)

- Weitere Möglichkeit für Manipulation mit LUT:

- Kontrasterhöhung durch Ausnützen des gesamten möglichen Helligkeitsbereichs: Intervall a - b wird auf Intervall 0 - 255 abgebildet.





Helligkeitskorrekturen (lokal)

08.11.2018

–Je steiler $T(f)$ ist (d.h., je größer $T'(f) = \frac{dT}{df}$

ist), desto desto stärker wird der Kontrast an der entsprechenden Stelle verstärkt.

–Wie wählt man a und b in der Praxis?

- Kontrasterhöhung in Helligkeitsbereichen, wo sehr viele Pixel mit ähnlichen Werten vorkommen.
- Dazu werden Histogramme eingesetzt (s. Abb. auf vorhergehender Seite).

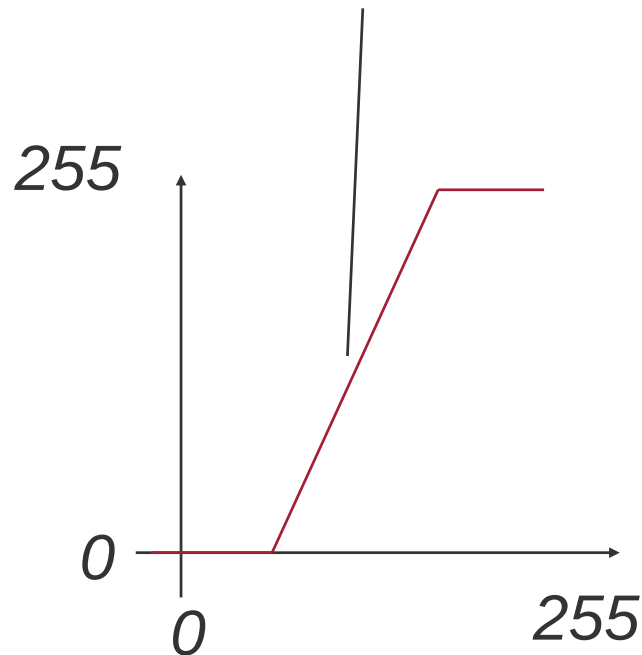


Helligkeitskorrekturen (lokal)

08.11.2018

Helligkeitstransformation durch lineare Funktion

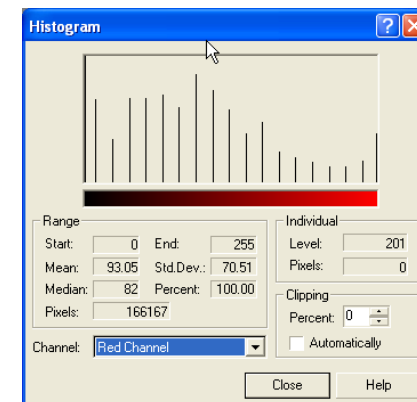
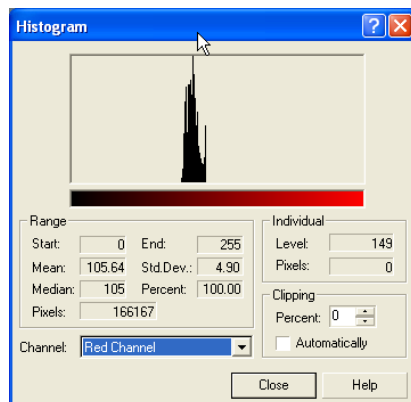
$$T(f) = af + b$$



Helligkeitskorrekturen (lokal)

08.11.2018

Helligkeitstransformation durch lineare Funktion





Histogrammeinebnung

08.11.2018

Histogrammequalization (Histogrammeinebnung)

- Ein Histogramm wird definiert durch:

$$h(i) = \sum_{m,n} \delta(f(m,n) - i)$$

- Es gibt für jede Intensität i an, wieviele Punkte mit dieser Intensität im Bild vorkommen.
- Automatische Kontrastanhebung: Der Kontrast soll umso höher sein, je mehr Pixel eine Intensität i haben.



Histogrammeinebnung

08.11.2018

- Auf diese Weise werden die resultierenden Werte in dieser Gegend „auseinandergezogen“, d.h. auf einen größeren Intensitätsbereich verteilt.
- Dazu wählen wir den Kontrast proportional zu $h(i)$

$$T' \propto h(i) \Rightarrow T \propto \int h(i) di$$



Histogrammeinebnung

08.11.2018

- $\int_0^i h(j) dj$ wird als kumulatives Histogramm bezeichnet (in der diskreten Form $\sum_{j=0}^i h(j)$) und zählt alle Pixel, die Intensitätswerte haben, die kleiner oder gleich i sind.
- Diskret funktioniert das natürlich nicht ganz so schön: Treppenstufen in T $\Rightarrow$ Lücken im resultierenden Histogramm.

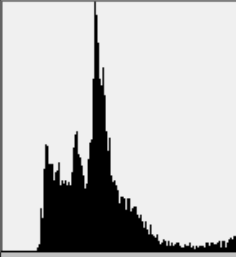
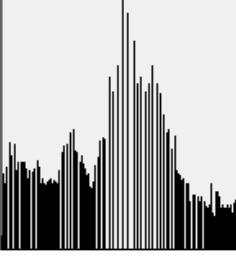
Histogrammeinebnung

08.11.2018

Computer-Graphik spielend lernen: Applet - Netscape

Histogramm-Manipulation

Bilder: Rose

Original:	Original-Ansicht	relative Häufigkeit:	Summenhäufigkeit:
			
Histogrammeinebnung:	Berechnen	relative Häufigkeit:	Summenhäufigkeit:
			

Copyright © 1996/97 University of Tübingen - [WSI/GRIS](#) Author: [Götz Rößner](#)

EXIT [Ende](#)

 [Applet](#)

[Einleitung](#)
[Bedienung](#)
[Guided Tour](#)
[Architektur](#)

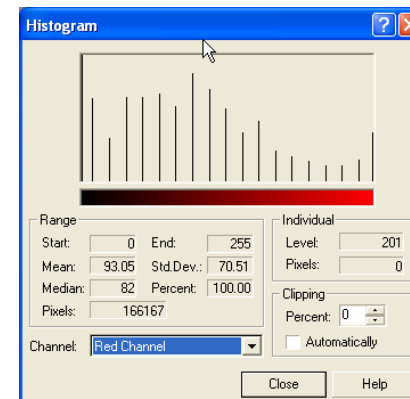
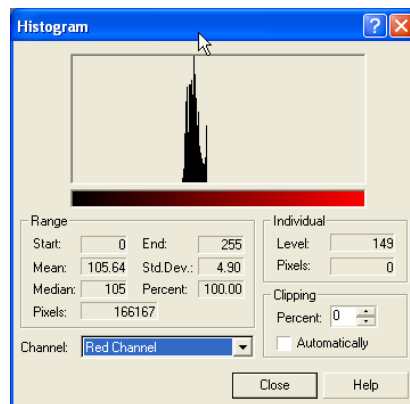
 [Kursbuch](#)

[Histogrammanipulation](#)
[Bildverbesserung](#)
[Grauwertmanipulation](#)

Helligkeitskorrekturen (lokal)

08.11.2018

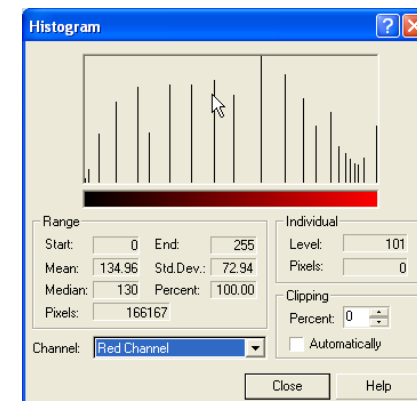
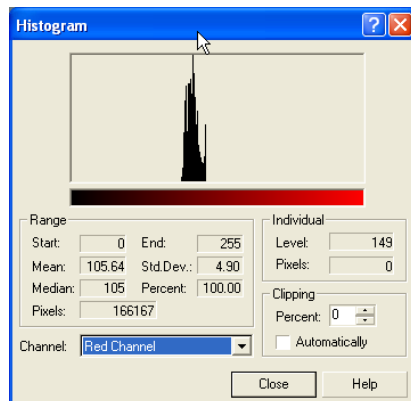
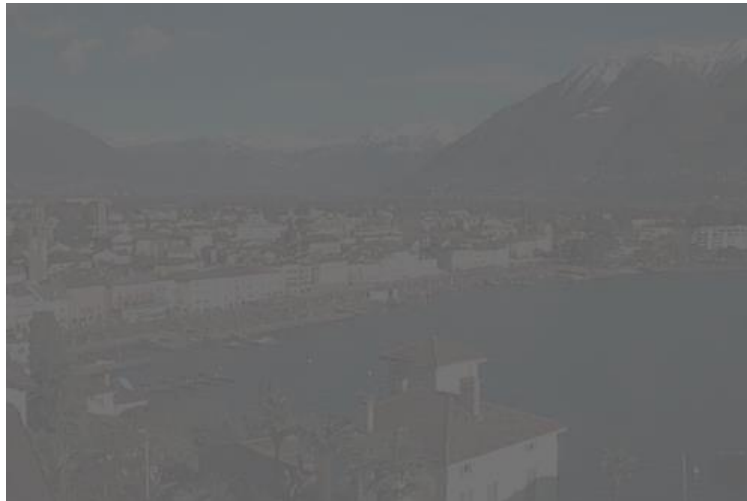
Helligkeitstransformation durch lineare Funktion



Helligkeitskorrekturen (lokal)

08.11.2018

Helligkeitstransformation durch Histogrammeinebnung





Histogrammeinebnung

08.11.2018

- Leider ist HE nicht immer unbedingt sinnvoll:
 - Großer Hintergrund bedeutet: Viele Pixel mit gleicher Helligkeit. Wird hier auseinandergezogen, so wird kleinstes Rauschen im Hintergrund sichtbar. Dafür wird zwangsläufig der Kontrast an anderer Stelle schlechter gemacht.
- Lösung: Adaptive Histogrammeinebnung: lokale Intensitätskorrektur



Histogrammeinebnung

08.11.2018

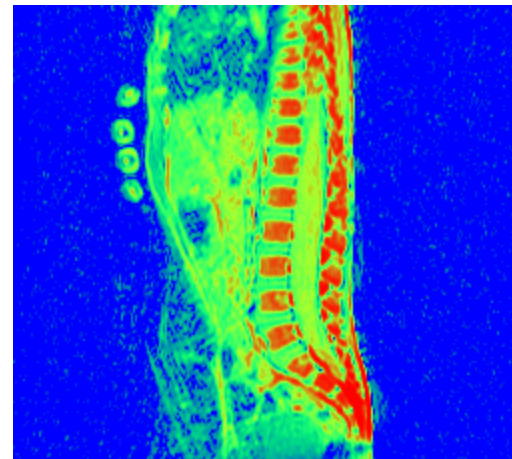
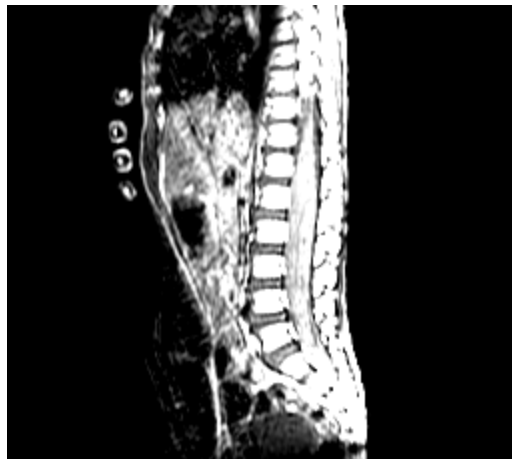
- Adaptive Histogrammeinebnung:
 - Unterteilung des Bildes in kleine Teilbilder (z.B. 16x16, 32x32)
 - Transformation mit vier Transformationsfunktionen aus den vier am nächsten gelegenen Fenstern
 - bilineare Interpolation der Ergebnisse

Helligkeitskorrekturen

08.11.2018

Eine weitere Möglichkeit, Lookuptables zu nutzen:
Farbtabelle, Pseudocolor:

- es werden drei Lookuptables für die drei Grundfarben verwendet. Beispiel:





Einfache Rauschfilterung

08.11.2018

Was kann man gegen Rauschen tun?

- Akkumulieren: Rauschen geht mit $\sqrt{\frac{1}{n}}$ gegen 0.
- Versuch, gestörte Werte durch die Umgebung zu rekonstruieren: nicht mehr lokaler Punktoperator sondern Filter.



Einfache Rauschfilterung

08.11.2018

Mittelwertbildung

- Mittelwert aus z.B. 3x3 (Boxfilter).
- Allgemein: Filterung durch Faltung mit Filterkern g (s. Bilder auf den folgenden Seiten)
- Faltung:

$$f'(x, y) = \int_{-\infty}^{\infty} f(s, t) \cdot g(x - s, y - t) ds dt$$



Exkurs: Faltung

08.11.2018

Definition der Faltung (1D):

$$g(x) \circ h(x) = \int_{-\infty}^{\infty} g(x - \alpha) h(\alpha) d\alpha$$

Applet (1D)

Lokal: Applet (1D)



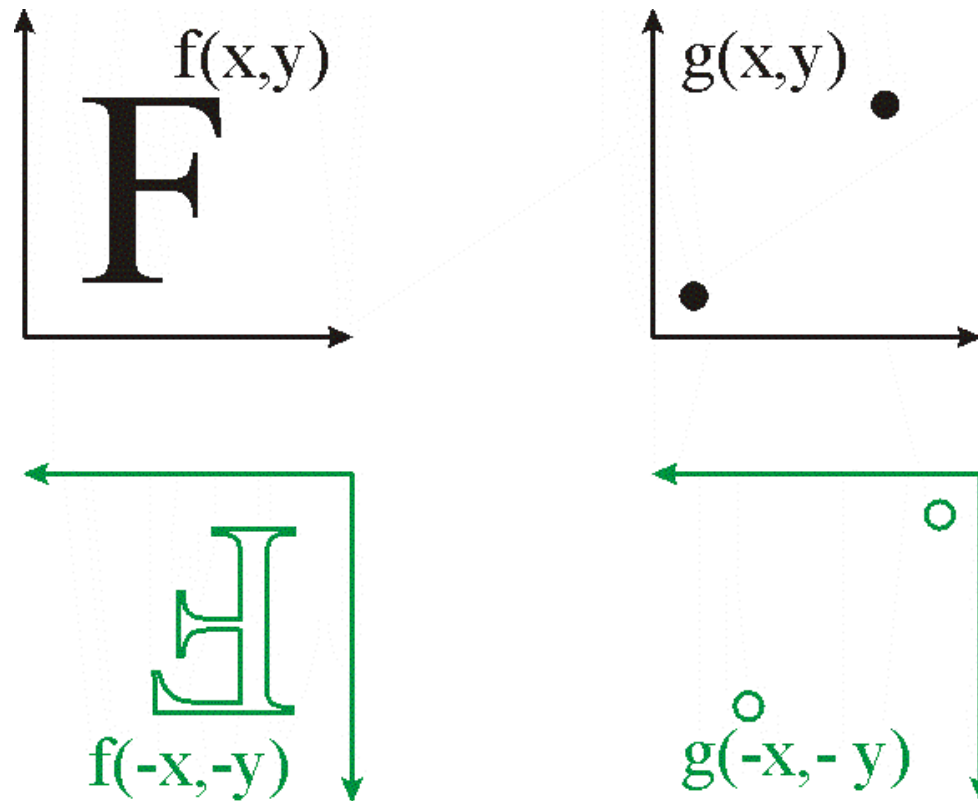
Definition der Faltung (2D):

$$\begin{aligned} f(x, y) \circ g(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - \alpha, y - \beta) g(\alpha, \beta) d\alpha d\beta \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x - \alpha, y - \beta) f(\alpha, \beta) d\alpha d\beta = g(x, y) \circ f(x, y) \end{aligned}$$

Faltung

08.11.2018

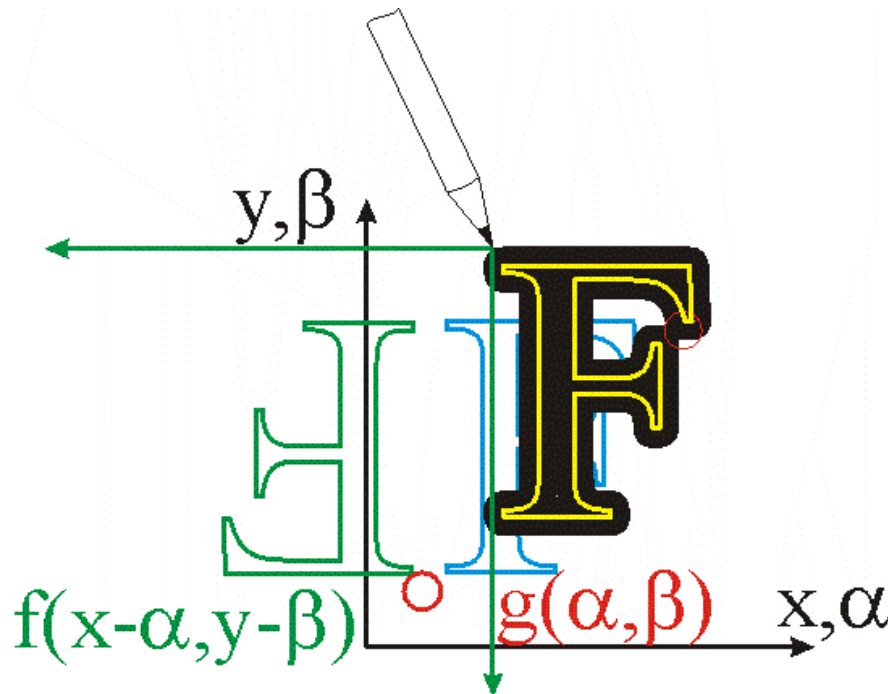
Beispiel:



Faltung

08.11.2018

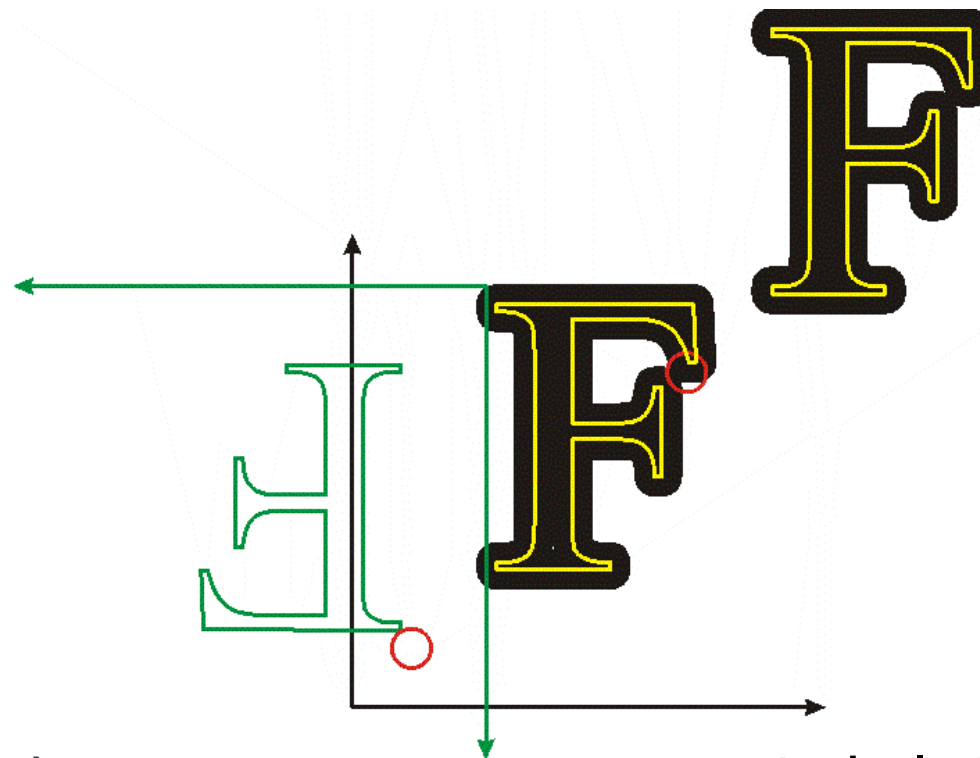
Berechnung des Integrals nach “Darüberschieben” der gespiegelten Funktion:





08.11.2018

Ergebnis:



Applet (2D)

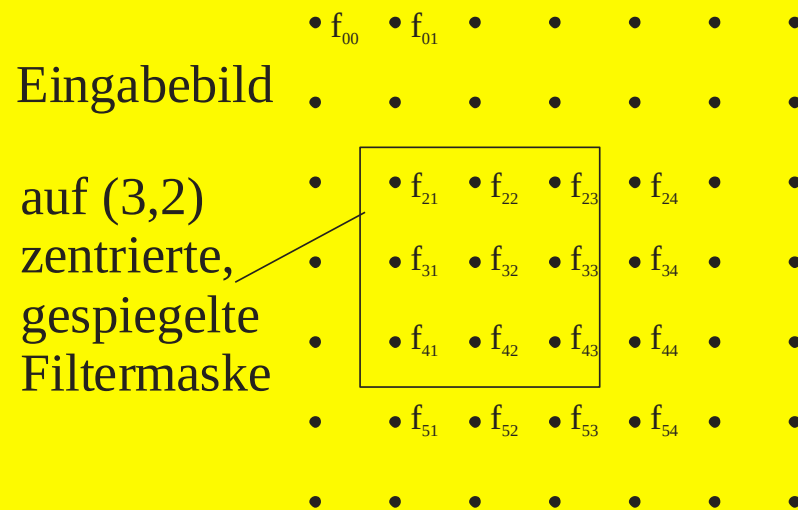
Lokal: Applet (2D)



08.11.2018

Einfache Rauschfilterung

Filterung mit Filtermaske



gespiegelte
Filtermaske

g_{22}	g_{21}	g_{20}
g_{12}	g_{11}	g_{10}
g_{02}	g_{01}	g_{00}

Ausgabe (3,2)

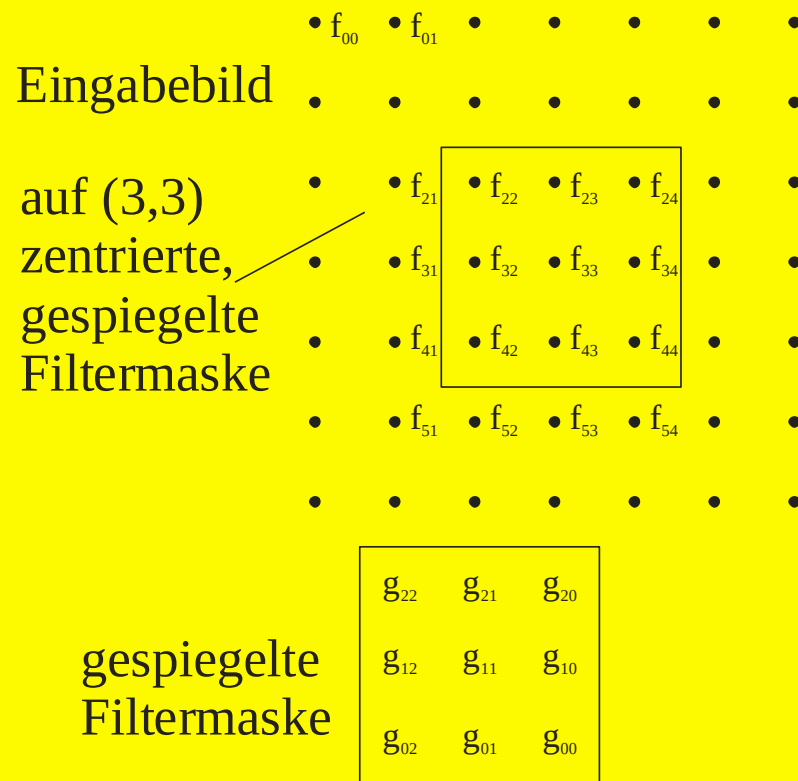
$$\begin{aligned}
 &= g_{22}f_{22} + g_{21}f_{21} + g_{20}f_{23} \\
 &+ g_{12}f_{31} + g_{11}f_{32} + g_{10}f_{33} \\
 &+ g_{02}f_{41} + g_{01}f_{42} + g_{00}f_{43}
 \end{aligned}$$



08.11.2018

Einfache Rauschfilterung

Filterung mit Filtermaske



Output (3,3)

$$= g_{22} f_{22} + g_{21} f_{23} + g_{20} f_{24} \\ + g_{12} f_{32} + g_{11} f_{33} + g_{10} f_{34} \\ + g_{02} f_{42} + g_{01} f_{43} + g_{00} f_{44}$$

Applet (Bild)

Lokal: Applet (Bild)



Einfache Rauschfilterung

08.11.2018

Boxfilter benutzt Filterkern

$$g(x, y) = \begin{cases} \text{const.} & \text{für } |x| < a, |y| < a \\ 0 & \text{sonst} \end{cases}$$

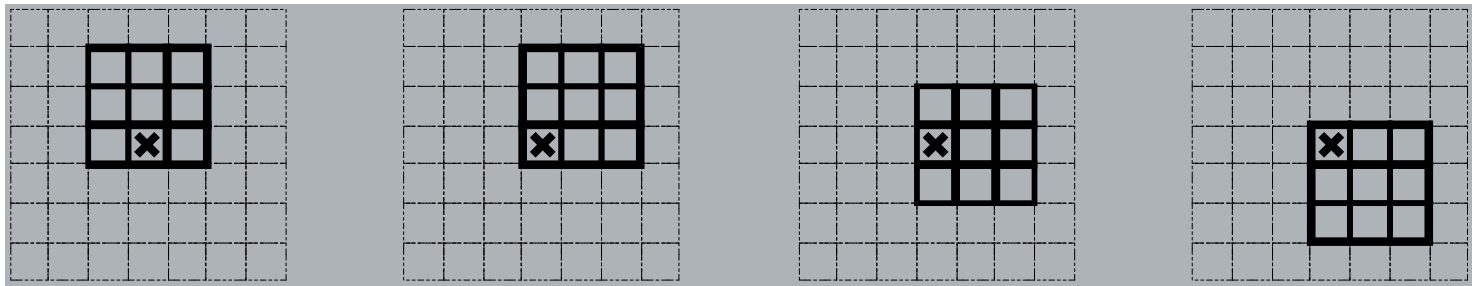
- Kanten werden verschlechtert.
- Lösungsmöglichkeiten:
 - Mittelwert nur aus ähnlichen Werten
 - Mittelwert nur, wenn dieser nicht zu sehr vom alten Wert abweicht
 - Rotating mask filtering

Einfache Rauschfilterung

08.11.2018

Rotating mask filtering

- Berechnung der Varianz für alle rotierten Masken
- Auswahl der Maske mit der kleinsten Varianz
- Berechnung des Ausgabewertes mit dieser Maske





Einfache Rauschfilterung

Rank filtering

- Werte werden nach Größe sortiert und mit einem dem Rang zugeordneten Gewichtungsfaktor multipliziert.
- Beispiel: Minimum, Maximum, Median.

12	15	87
1	2	6
194	23	136
8	3	7
65	229	46
5	9	4

Rang	Gewicht		
	Minimum	Maximum	Median
1	1	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	1
6	0	0	0
7	0	0	0
8	0	0	0
9	0	1	0



Einfache Rauschfilterung

08.11.2018

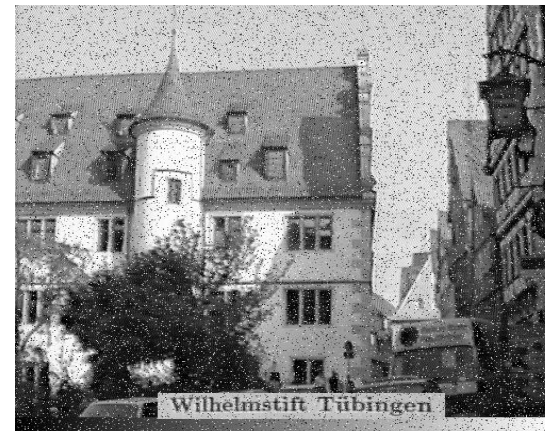
Rank filtering

- Medianfilter (Mittlerer der nach Helligkeit geordneten Werte aus Umgebung)
- Dünne Linien und Ecken werden zerstört.
- Was kann man da tun? Antwort:
 - kreuzförmige Maske für den Erhalt senkrechter und waagrechter Linien,
 - Mehrfachwertung des Zentralen Wertes.

08.11.2018



a) Original



b) Original mit Impulsrauschen



c) geglättet mit 5x5 Mittelwert



d) geglättet mit 5x5 Median