„Bildverarbeitung“

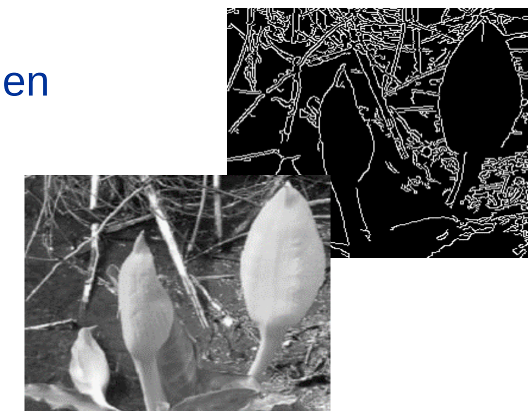
Hochschule Niederrhein

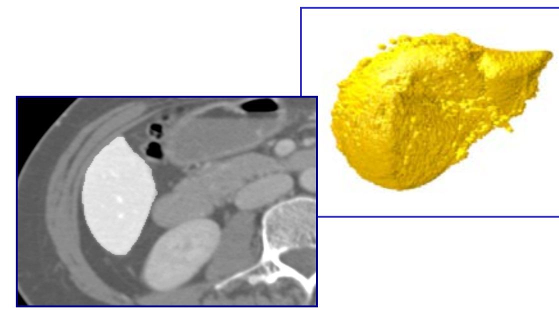
Regina Pohle-Fröhlich

Tiefpassfilterung im Ortsbereich

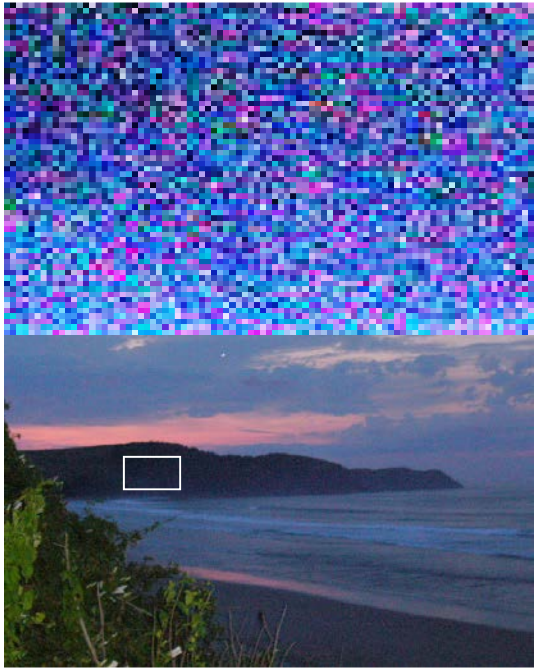
Roter Faden durch die Vorlesung

- Bildaufnahme
- Histogramme
- Grauwertmodifikation
- Glättungsfilter
- Kantenfilter
- Nichtlineare Filter
- Segmentierung
- Morphologische Operationen
- Fourier Transformation
- Anwendung der FFT
- Probeklausur

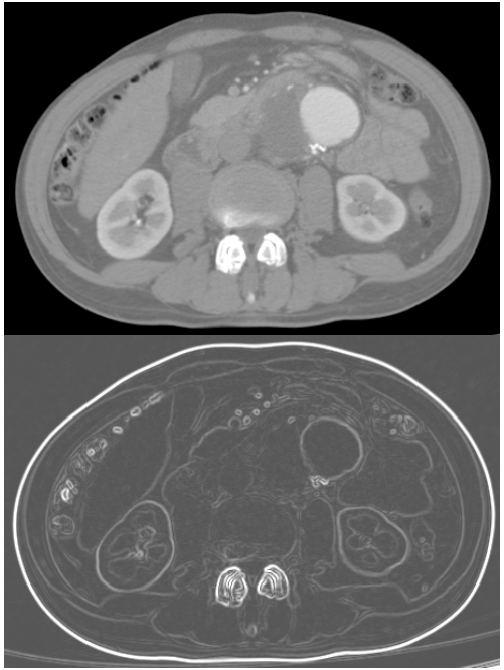




4 Filtern



Minderung von Rauschen

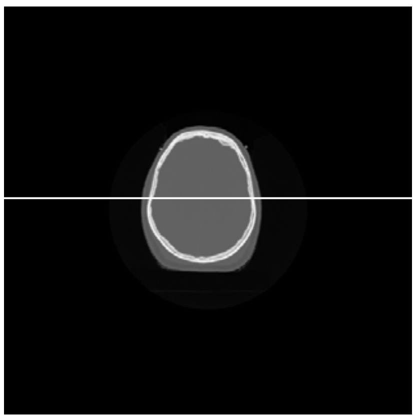


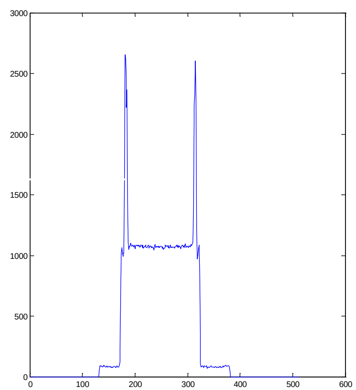
Erkennen von Struktur-
grenzen

Beseitigung von Details

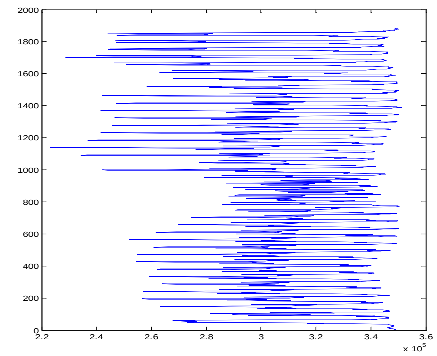
4.1 Visualisierung von Filterergebnissen

Grauwertprofil: Grauwertverlauf längs einer Linie

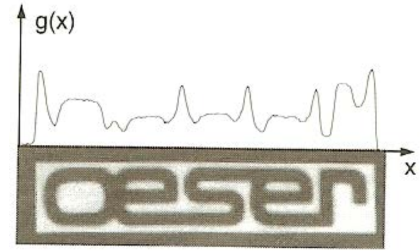




Zeilenprofil: Grauwertsumme je Zeile



Spaltenprofil: Grauwertsumme je Spalte



[1]

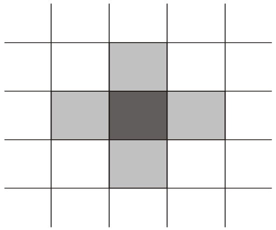
4.2 Nachbarschaftsoperationen

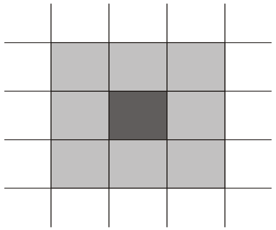
- Grauwert eines Punktes als auch bestimmte Menge seiner räumlichen bzw. zeitlichen Nachbarn dienen als Eingabe
- Aus Eingabewerten wird Ergebnis berechnet und an die Koordinate des Referenzpunktes in das Zielbild geschrieben
- Bildgeometrie ändert sich nicht, d.h. die Position der Pixel bleibt nach der Operation unverändert.

4.2 Nachbarschaftsoperationen

Quadratische (rechteckige) Gitter:

- 4-Nachbarschaft: Pixel haben eine gemeinsame Seite
- 8-Nachbarschaft: Pixel haben eine gemeinsame Seite oder einen gemeinsamen Eckpunkt

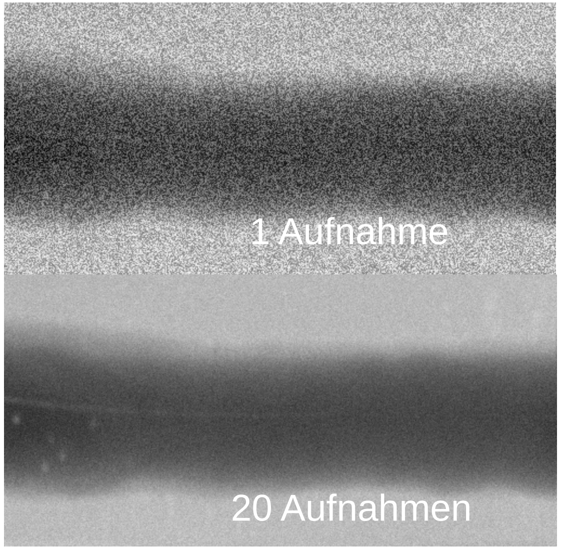




4.3 Rauschreduktion

Integration über eine 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:
  - $E\{g(x,y)\} = E\{f(x,y)\} + E\{n(x,y)\}$
    $= E\{ f(x,y) \} + 0 = f(x,y)$
  - Abschätzung von $E\{g(m,n)\}$ durch Integration über die Bilder.



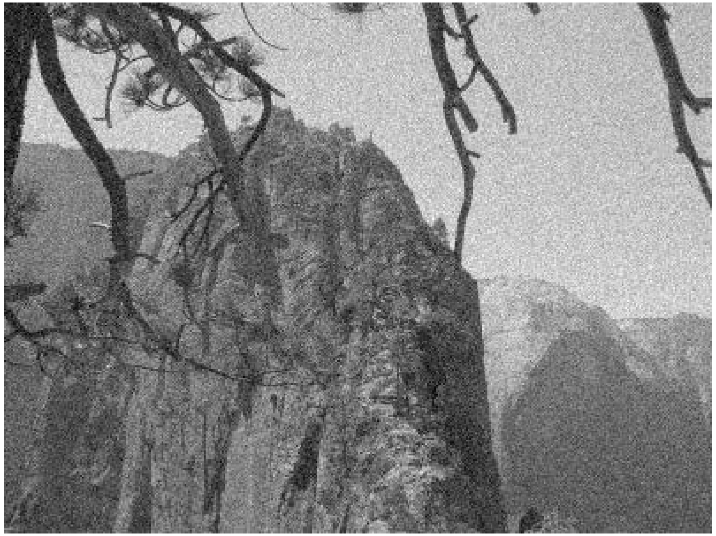
4.3 Rauschreduktion

Integration über eine homogene 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.

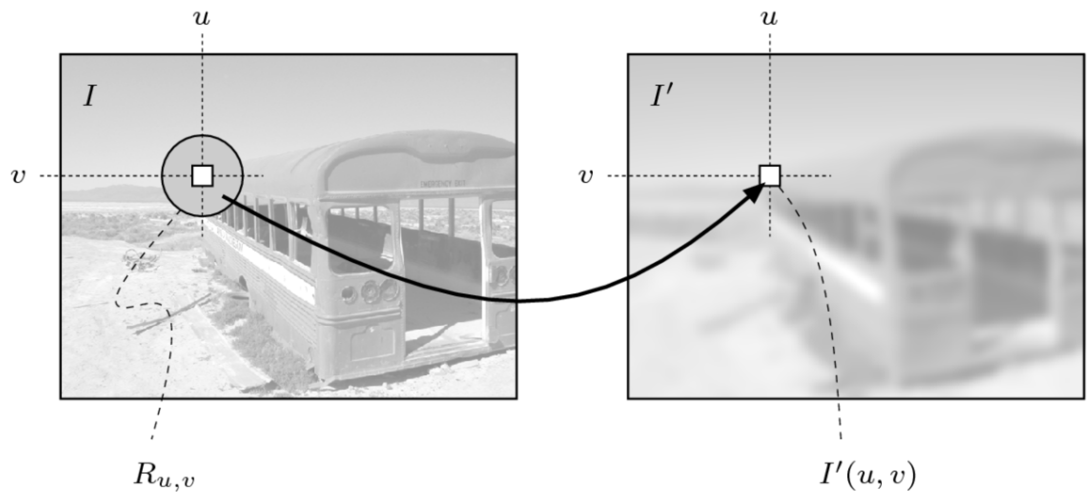
- Annahmen:
  - Bild besteht aus homogenen Bereichen.
  - Benachbarte Punkte haben den gleichen Grauwert.

- Rauschunterdrückung:
  - Mittelwertbildung über vorgegebene Nachbarschaft.



[2]

4.3 Rauschreduktion



[3]

Idee: Ersetze jeden Pixel durch den Durchschnitt seiner Nachbarschaft p1, p2, .., p9:

$$I'(u,v) \leftarrow \frac{1}{9} \sum_{i=1}^{9} p_i$$

4.3 Rauschreduktion

3x3-Boxcar-Filter in relativen Bildkoordinaten:

$$I'(u,v) \leftarrow \frac{1}{9}\left[I(u-1,v-1)+I(u,v-1)+I(u+1,v-1)+I(u-1,v)+I(u,v)+I(u+1,v)+I(u-1,v+1)+I(u,v+1)+I(u+1,v+1)\right]$$

Filtermerkmale:

- Ergebnis wird nicht aus einem einzigen Pixel berechnet, sondern aus einer Menge von Pixeln
- Koordinaten der Quellpixel habe eine feste relative Position zum Zielpixel und bilden i.A. eine zusammenhängende Region.

Parameter:

- Größe der Filterregion
- Form der Filterregion
- Gewichtung der Quellpixel (konstant oder ortsabhängig)

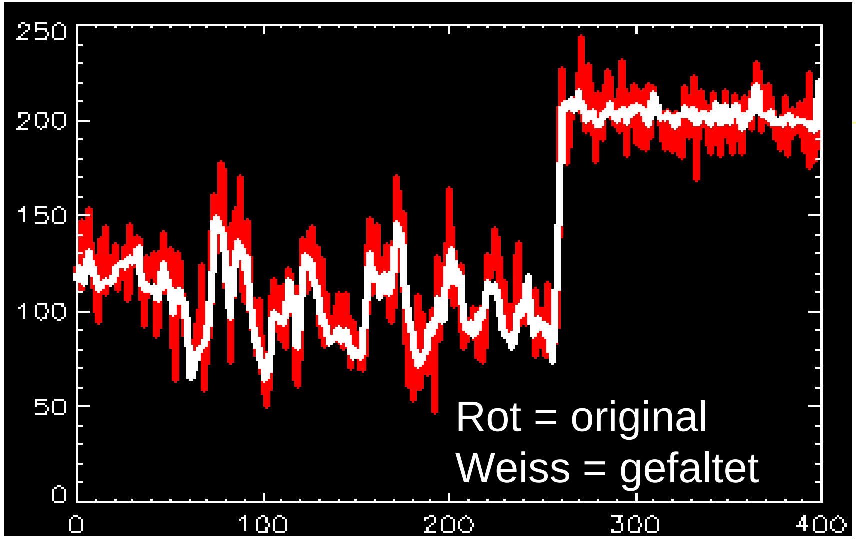
Beispiel

4.3 Rauschreduktion

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

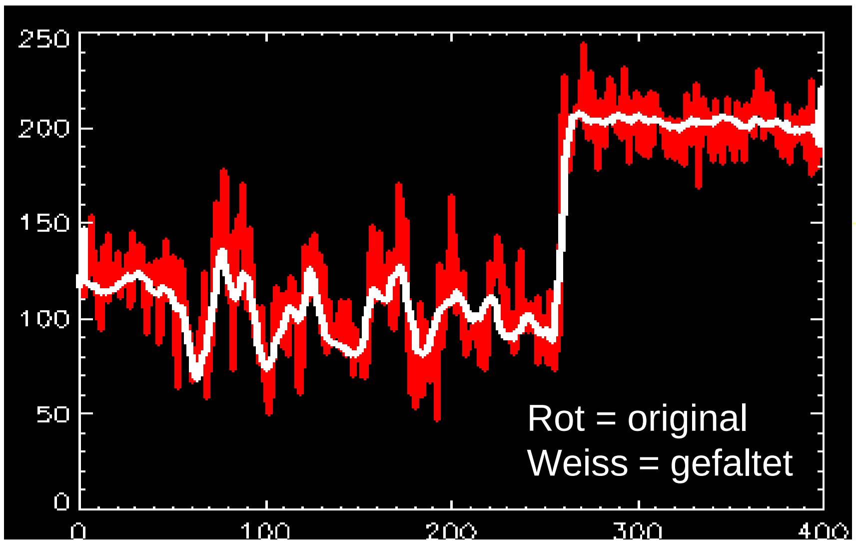
Filtermaske

3x3 Boxcar-Filter



4.3 Rauschreduktion

7x7 Boxcar-Filter

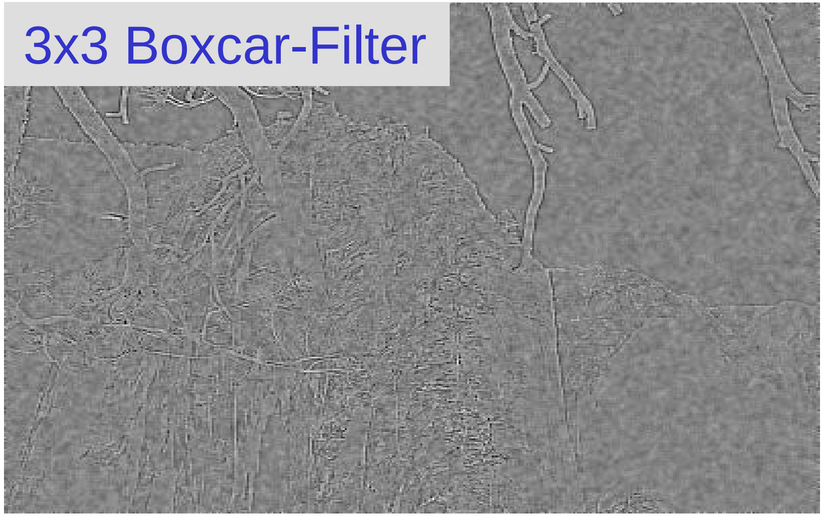


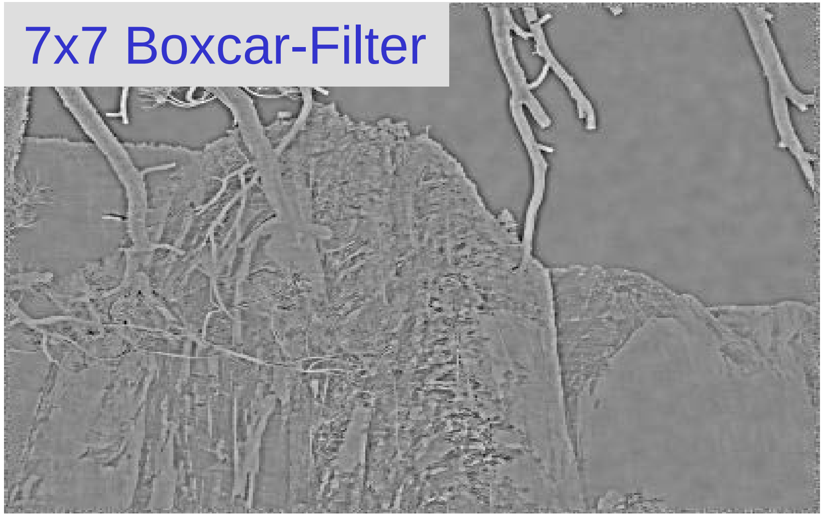
[2]

4.3 Rauschreduktion

Beobachtung: Bild wird unscharf, Kanten werden verwaschen

Grund: Annahme konstanter Funktionswerte ist nicht wahr.





$$\left| I'(u,v) - I(u,v) \right|$$

[2]

4.3 Rauschreduktion

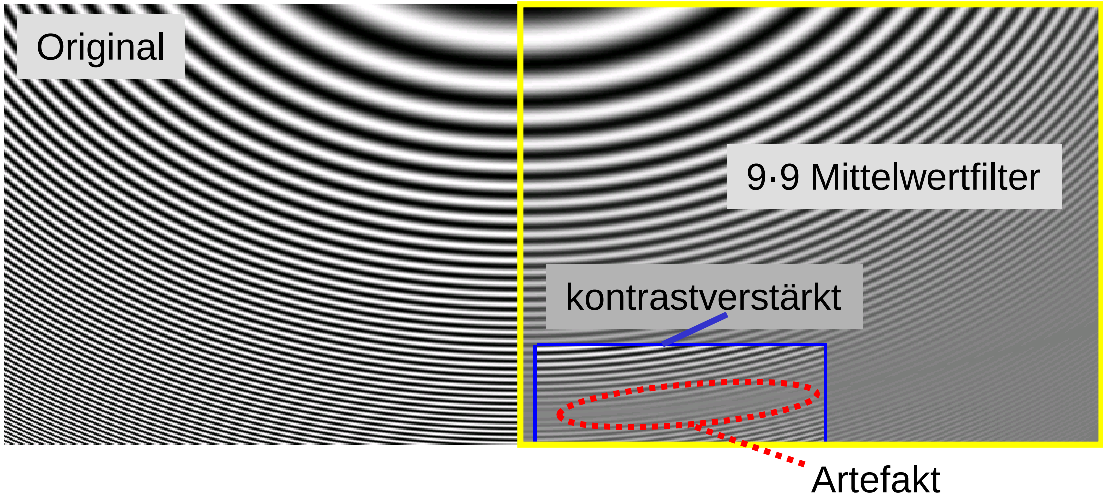
Tiefpassfilter:

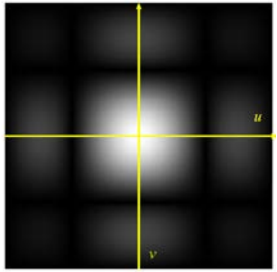
- schwächen hochfrequente Teile des Bildinhalts (Details und Rauschen)
- Visueller Eindruck des Bildes wird weicher
- Kanten werden verwischt
- in homogenen Bereichen keine Auswirkung

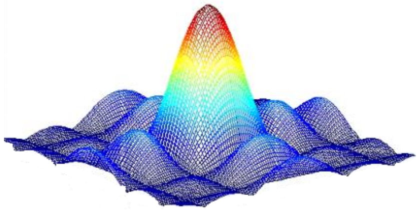
4.3 Rauschreduktion

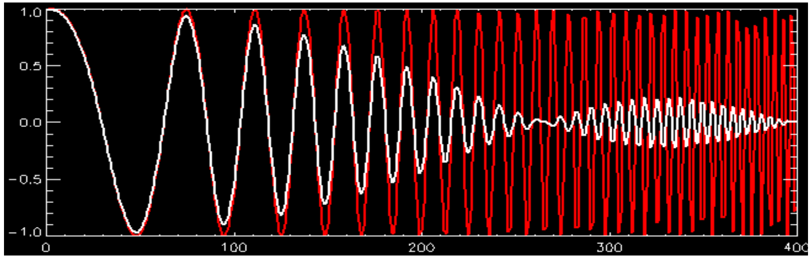
Anomalie des Mittelwertfilters

Transferfunktion: beschreibt die Abhängigkeit des Ausgangssignals von dessen Eingangssignal









[2]

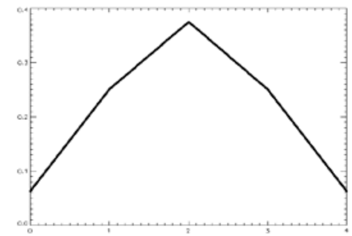
Bildzeile: rot: vor der Filterung, weiß: nach Filterung

3.3 Rauschreduktion

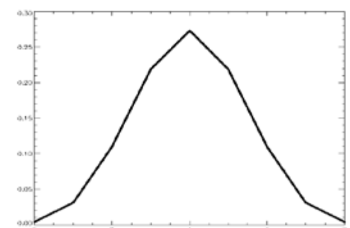
Binomialfilter

n	f		σ^2
0	1	1	0
1	1/2	1 1	1/4
2	1/4	1 2 1	1/2
3	1/8	1 3 3 1	3/4
4	1/16	1 4 6 4 1	1
5	1/32	1 5 10 10 5 1	5/4
6	1/64	1 6 15 20 15 6 1	3/2
7	1/128	1 7 21 35 35 21 7 1	7/4
8	1/256	1 8 28 56 70 56 28 8 1	2

N=4



N=8



Zweidimensionaler Binomialfilter:

$$B^n = (B^n)^T \cdot B^n$$

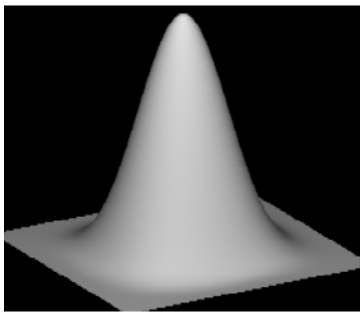
$$B^2 = \frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix}^T \cdot \frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1\end{bmatrix}$$

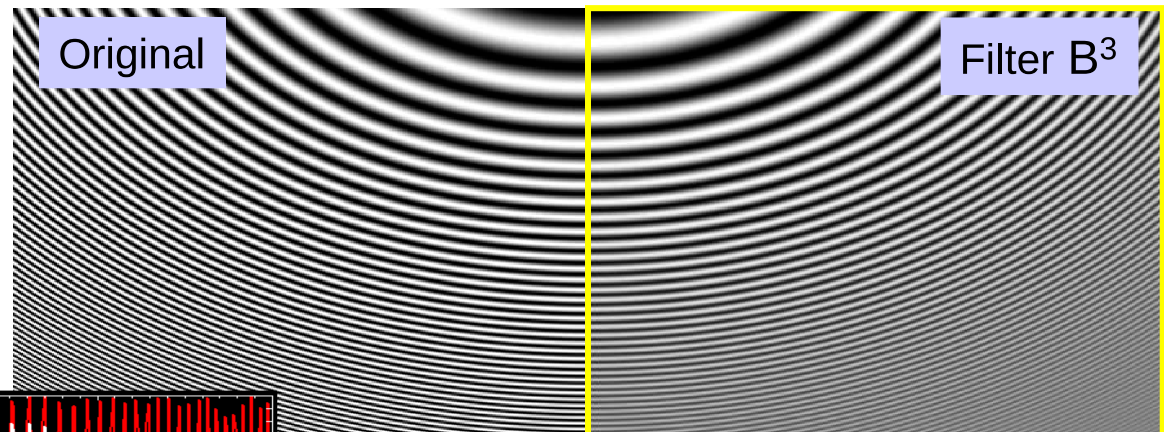
Beispiel

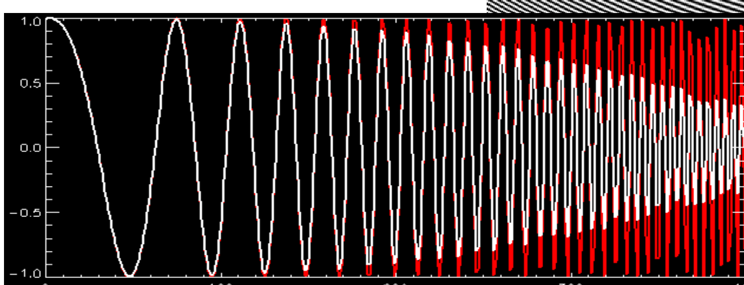
4.3 Rauschreduktion

Resultate des Binomialfilters

Transferfunktion







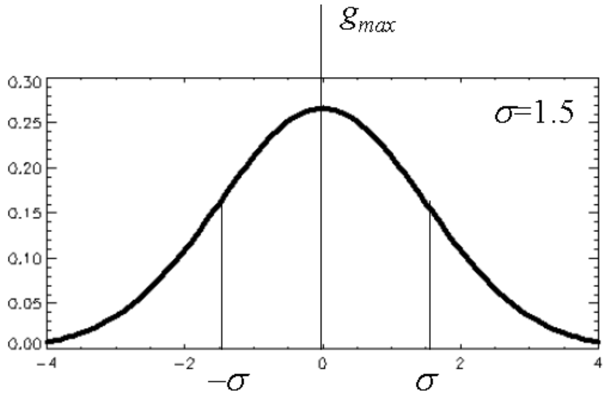
[2]

Bildzeile
rot: vor der Filterung
weiß: nach Filterung

4.3 Rauschreduktion

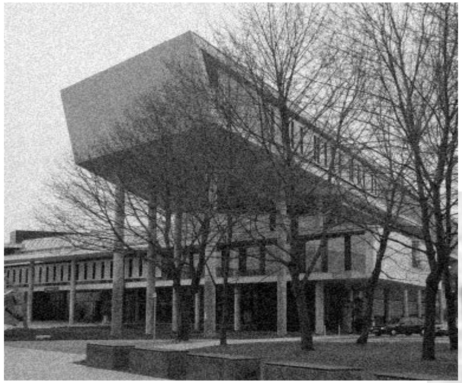
Binomialfilter und Gaußfunktion

- Für größere Filterkerne Annäherung des Binomialfilters an Gauß'sche Glockenkurve
- Funktion wird nie Null
- Filterkern endlicher Größe „schneidet" die Funktion ab
- Kerngröße ca. $2 \times 3\sigma + 1$
- ausgewählter Filterkern muss normiert werden
- Separabilität macht Filterung effizient:
  $g(x,y)=g(x)*g(y)$



4.3 Rauschreduktion

Gaußfilter



σ=1.5

σ=3.0

[2]

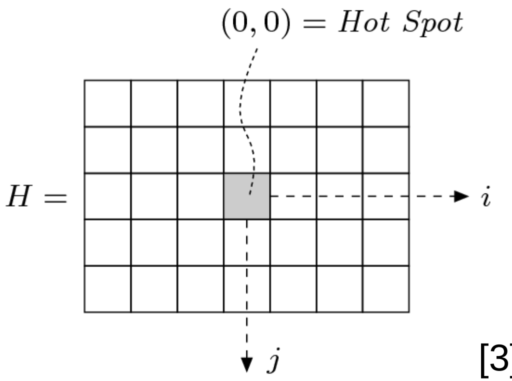
4.3 Rauschreduktion

Eigenschaften aller Rauschunterdrückungsfilter

- Die Summe aller Elemente der Maske ist immer 1 (Gesamthelligkeit bleibt gleich)
- nur positive Elemente in der Maske
- ungerade Anzahl von Werten
- Maske ist punktsymmetrisch zum Koordinatenursprung der Maske
- Bildinformation wird nicht verschoben und keine Richtung wird bevorzugt

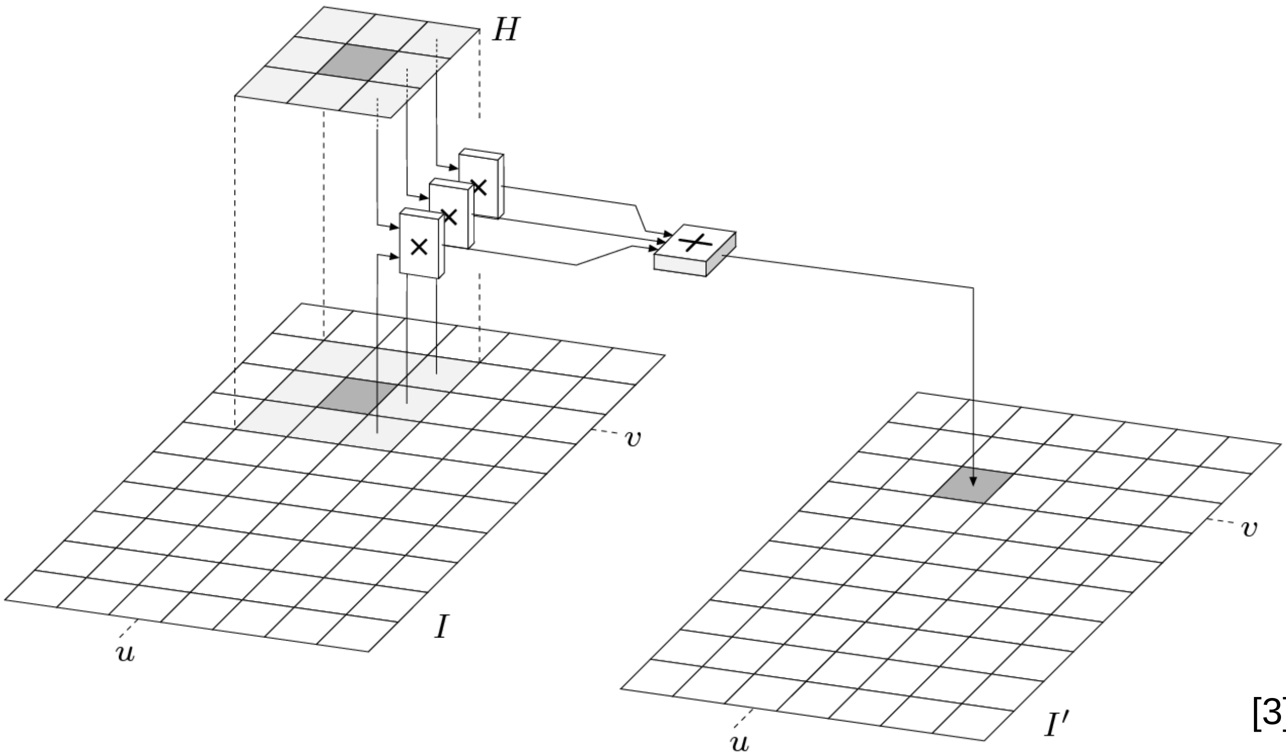
4.4 Lineare Filter

- Lineare Filter: Wert des Zielpixels wird als gewichtete Summe der Quellpixel berechnet.
- Größe und Form der Filterregion und Gewichte des Filter werden durch eine Matrix von Filterkoeffizienten spezifiziert (Filtermaske)
- Die Filtermatrix ist - wie ein Bild – eine diskrete zweidimensionale Funktion.
- Koordinaten werden meist relativ zum Zentrum angegeben (”hot spot”).



[3]

4.4 Lineare Filter



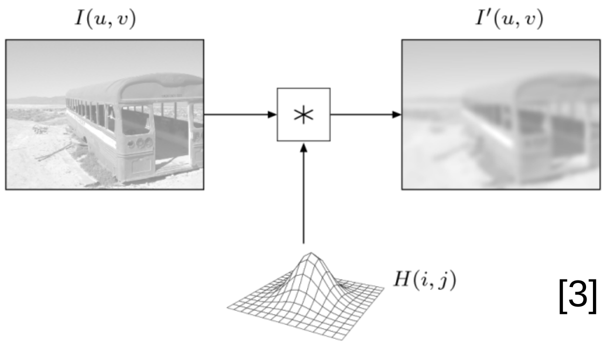
[3]

4.4 Lineare Filter

Diskrete Faltung (Konvolution)

- Funktionswert des veränderten Bildes nach der Faltung $f*g$ entsteht durch Addition der umgebenden Funktionswerte mit einer nach (m,n) verschobenen Gewichtsfunktion
- Definitionsbereich des Bildes und der Gewichtsfunktion geht von -∞ bis +∞

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



[3]

Sonderfall der Faltung

- endliche Definitionsbereiche
- Spezielle Randbehandlung erforderlich
    - Randpunkte nicht behandeln (Bild wird kleiner oder alte Werte behalten)
    - Originalbild an den Rändern spiegeln
    - Maske in den Randbereichen einschränken

4.4 Lineare Filter

Eigenschaften der Faltung

- Kommutativität: $h_1 * h_2 = h_2 * h_1$
- Assoziativität: $(h_1 * h_2) * h_3 = h_1 * (h_2 * h_3)$
- Linearität: $(a \cdot I) * H = I * (a \cdot H) = a \cdot (I * H)$

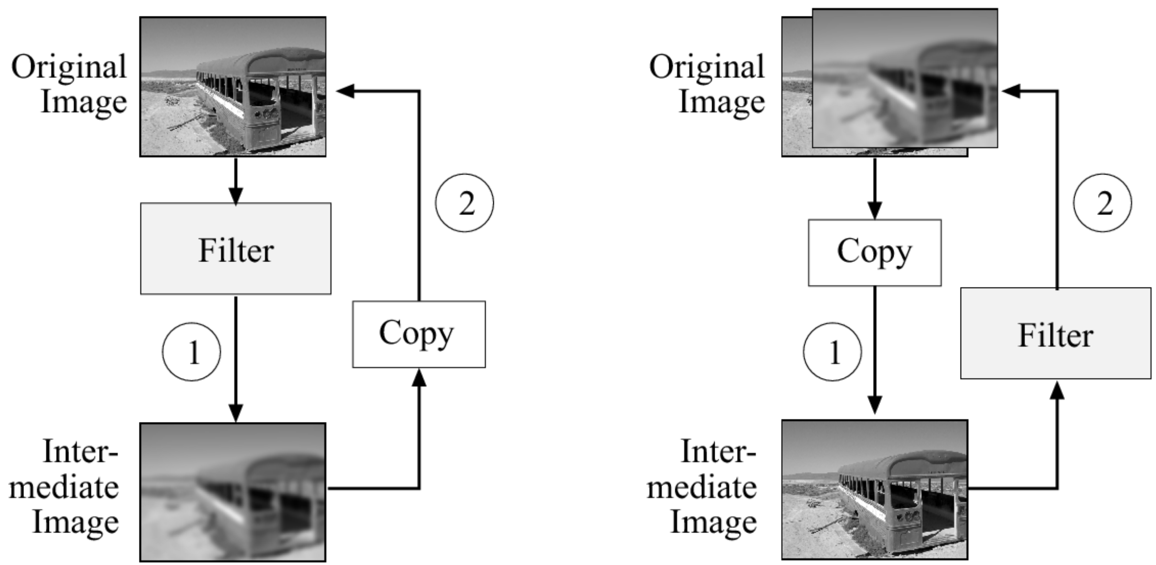
  $(I_1 + I_2) * H = I_1 * H + I_2 * H$

  aber

  $(b + I) * H \neq b + I * H$
- verschiebungsinvariant: Operatoranwort hängt nicht vom Ort, sondern nur von den Werten in der Umgebung ab

4.5 Implementierung linearer Filter

Im Gegensatz zu Punktoperationen bei Filtern keine ”in place”-Verarbeitung möglich (Quellpixel werden mehrfach benötigt)



Variante A

Variante B

[3]

4.5 Implementierung linearer Filter

Einfaches Beispiel: 3x3 Boxcar-Filter: 4 Schleifen

```
int w = orig.getWidth();
int h = orig.getHeight();
ImageProcessor copy = orig.duplicate();

for (int v=1; v<=h-2; v++) {
    for (int u=1; u<=w-2; u++) {
        //compute filter result for position (u,v)
        int sum = 0;
        for (int j=-1; j<=1; j++) {
            for (int i=-1; i<=1; i++) {
                int p = copy.getPixel(u+i,v+j);
                sum = sum + p;
            }
        }
        int q = (int) (sum / 9.0);
        orig.putPixel(u,v,q);
    }
}
```

[3]

4.5 Implementierung linearer Filter

3x3-Glättungsfilter mit Gauß-förmiger Maske

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

```
    int w = orig.getWidth();
    int h = orig.getHeight();
    // 3 × 3 filter matrix
    double[][] filter = {
        {0.075, 0.125, 0.075},
        {0.125, 0.200, 0.125},
        {0.075, 0.125, 0.075}
    };
    ImageProcessor copy = orig.duplicate();

    for (int v=1; v<=h-2; v++) {
        for (int u=1; u<=w-2; u++) {
            // compute filter result for position (u,v)
            double sum = 0;
            for (int j=-1; j<=1; j++) {
                for (int i=-1; i<=1; i++) {
                    int p = copy.getPixel(u+i,v+j);
                    // get the corresponding filter coefficient:
                    double c = filter[j+1][i+1];
                    sum = sum + c * p;
                }
            }
            int q = (int) Math.round(sum);
            orig.putPixel(u,v,q);
        }
    }
```

[3]

4.5 Implementierung linearer Filter

Oft ist es vorteilhafter, mit ganzzahligen Filterkoeffizienten zu arbeiten:

- keine Umwandlung und Speicherung des Bildes in Gleitkommaformat notwendig
- auf manchen Rechnerarchitekturen sind Ganzzahloperationen schneller
- Realisierung über einen Skalierungsfaktor

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

4.5 Implementierung linearer Filter

Separable Filter

Aus der Assoziativität ergibt sich, dass ein großer Filter H in mehrere kleine Filter H_i zerlegt werden kann:

$$I * H = I * (H_1 * H_2 * \ldots) = (\ldots((I * H_1) * H_2) * \ldots)$$

Zerlegung eines zweidimensionalen Filters in zwei eindimensionale Filter in x- und y-Richtung (x/y-Separabilität), z.B

$$H_{xy} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \textit{zerlegt in} \quad H_x = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \textit{bzw.} \quad H_y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$I' = I * H_{xy} = I * (H_x * H_y) = (I * H_x) * H_y$$

8 statt 15 Multiplikationen

Typisches Beispiel: Gauß-Filter

4.5 Implementierung linearer Filter

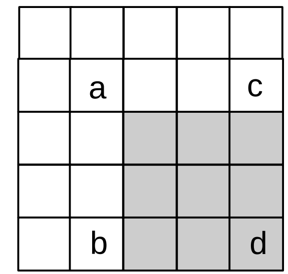
Mittelwertfilterung über Integralbild

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	..
..	..	..	..	..	..

Bild

1	2	3	4	5	..
2	4	6	8	10	..
3	6	9	12	15	..
4	8	12	16	20	..
5	10	15	20	25	..
..	..	..	..	..	..

Integralbild (Summe aller Pixel oberhalb und links vom betrachteten Pixel)

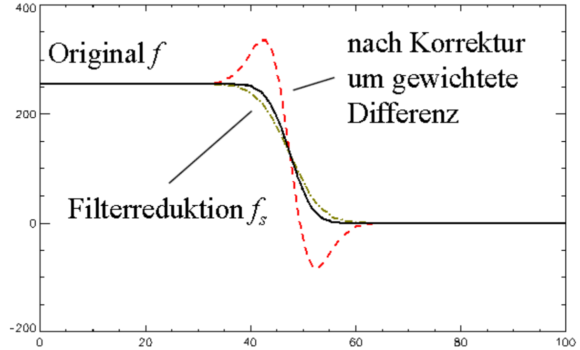


Pixelflächensumme
s=a-b-c+d

4.6 Maskierung der Unschärfe

Verbesserung des visuellen Bildeindrucks mit Maskierung der Unschärfe (unsharp masking)→ Bild wird wegen Machband-Effekt schärfer wahrgenommen

- Berechne Bild f_s durch Filterung von f mit Gaußfunktion mit Standard-abweichung σ
- Addiere mit p gewichtete Differenz $f\text{-}f_s$ auf das Originalbild $f_{USM}=f+p(f-f_s)$ (p wird oft in Prozent angegeben, d.h. 50% entspricht p=0.5)
- f_{USM} wird für f eingesetzt, falls der Unterschied zwischen f und f_s größer ist, als eine Schwelle t.



4.6 Maskierung der Unschärfe

σ (radius)

- je größer der Radius, desto breiter ist der verstärkende Rand an Kanten
- Wert ist die Standardabweichung der Gaußfunktion, d.h. die Breite ist ca. 6-mal größer

p (amount)

- je größer der Wert , desto höher ist Verstärkung an den Rändern

t (threshold)

- je höher der Wert, desto stärker muss Kante sein, damit überhaupt ein Sharpening stattfindet
- jeder Wert t>0 führt dazu, dass das Filter nicht mehr linear ist (Artefakte sind möglich!)

4.6 Maskierung der Unschärfe

4.6 Maskierung der Unschärfe

4.6 Maskierung der Unschärfe

Beispiel

4.7 Zusammenfassung

- Rauschenunterdrückung kann durch Schätzung des Erwartungswerts der Bildfunktion erreicht werden
- Schätzung des Erwartungswerts = zeitliche oder räumliche Integration
- im Ortsraum: räumliche Integration durch Faltung des Bildes mit Filtermasken
- Mehrere Möglichkeiten der Implementation

Bildquellen

[1] A. Ehrhardt: Einführung in die Digitale Bildverarbeitung, Vieweg+Teubner, 2008

[2] K. D. Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005

[3] W. Burger, M.J. Burge: Digitale Bildverarbeitung, Springer Verlag, 2005