Computergrafik 2: Übung 4

Konvolution, Unscharfmaskierung, Medianfilter

Quiz

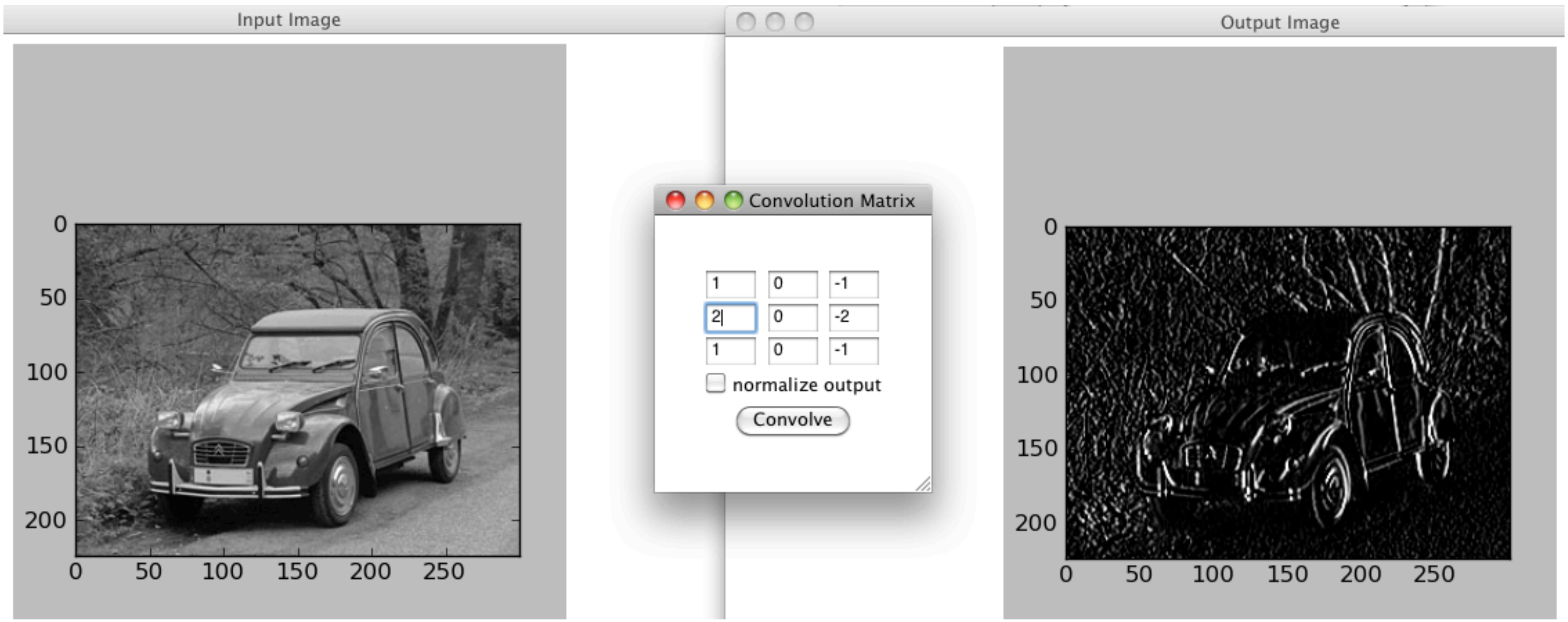
- Unterschied Konvolution / Korrelation?
- Zusammenhang Punktantwort / Konvolution?
- Probleme am Rand?
- Vorteil Gaußfilter gegenüber Boxcarfilter?
- Idee der Non-Local Means?

Besprechung Übung 3

- Probleme?

convolve.py

- Experimentieren mit verschiedenen Konvolutionskernen



Filterung im Ortsraum

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

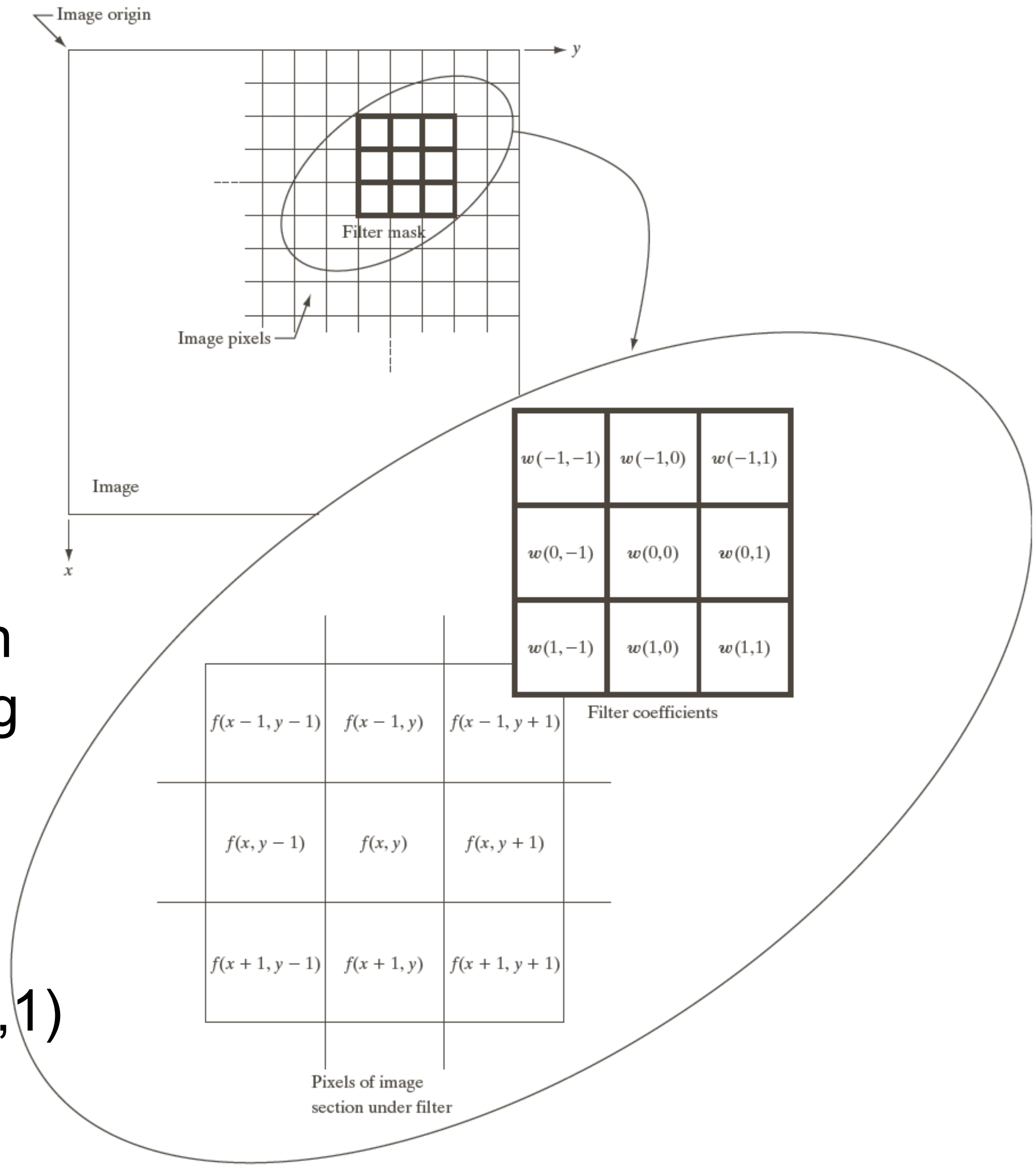


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Filtern im Ortsraum

- Filterung als gewichtetes Mittel

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

  - m×n Filtermaske mit m = 2a+1, n = 2b+1
  - M×N Bild mit M Zeilen und N Spalten
- Üblicherweise
  - Filtermaske begrenzt, Gewicht normalisiert (sum=1)
  - Seitenlänge des Filters ungerade

Berechnung Konvolutionskern Gauß?

- 2D-Gaußfunktion (Mittelpunkt (0,0), Standardabw. σ)

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

- Vereinfachung

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

- an verschiedenen Stellen auswerten (z.B. -1, 0, 1 für x,y)
- σ wie gewünscht einstellen
- normalisieren (so dass Summe 1)

Separierbarkeit linearer Filter

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

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

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

Separierbarkeit linearer Filter

- Verarbeitung zeilenweise

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

- Verarbeitung spaltenweise

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

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

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

Separierbarkeit der Gaußfunktion

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

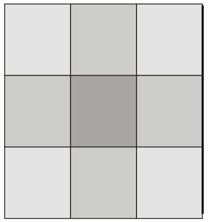
$$= a\exp\left[-bx^2\right]\exp\left[-by^2\right]$$

G(0,0)

G(1,-1)

1	2	1
2	4	2
1	2	1

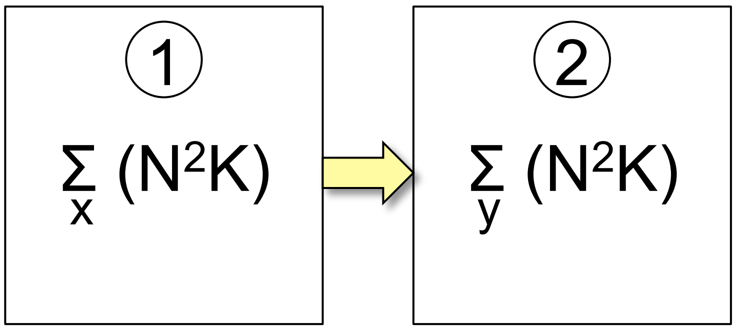
K



- Konvolution in lokaler Umgebung:

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

$$= a\underbrace{\sum_y \exp\left[-by^2\right]}_{2}\underbrace{\sum_x I_{xy}\exp\left[-bx^2\right]}_{1}$$



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

Unscharfes Maskieren

- Originalbild unschärfer machen (Tiefpass filtern)
- Unscharfes Bild vom Original subtrahieren (ergibt die unscharfe Maske)
- Maske (nach Skalierung) zum Original addieren
- k > 1: Highboost filtering

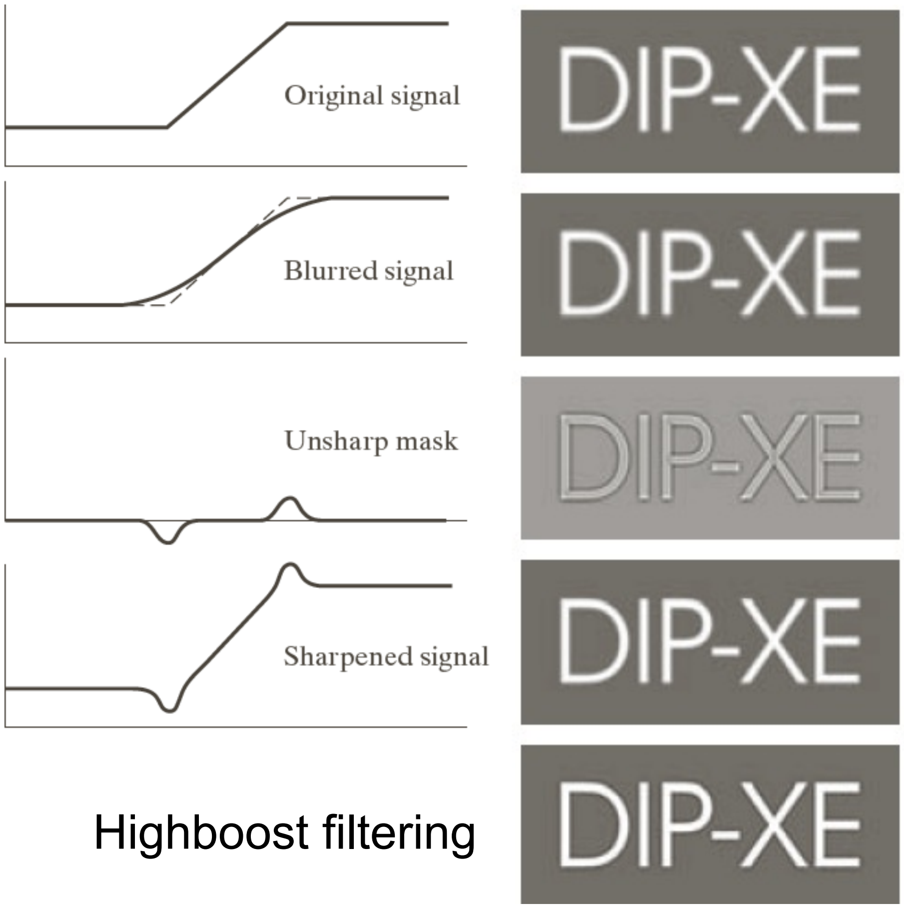
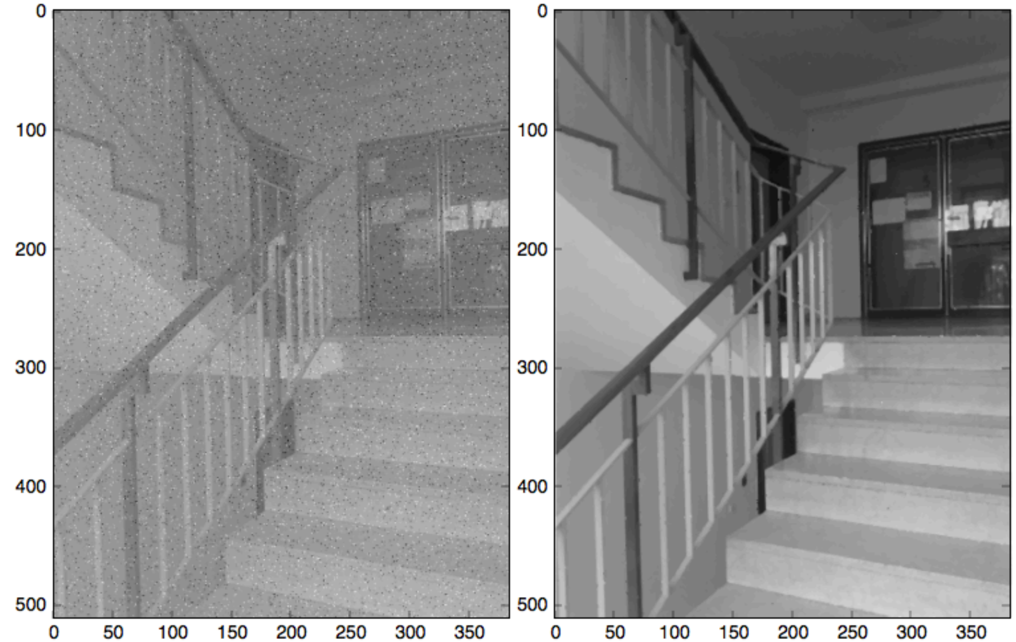


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Medianfilter

- Vorgehen
  - Sortierung der Elemente in einer Filtermaske (z.B. 3x3)
  - Auswahl des in der Mitte einsortierten Werts (Median)
  - Eintragung in die zentrale Position



26 3.	132 8.	112 5.
25 2.	102 4.	142 9.
17 1.	122 7.	117 6.

erster Rang

mittlerer Rang (Median)

letzter Rang

Bild laden und speichern

- Bild laden und speichern

```
img = plt.imread('lena-gray-256.png')

# Bild manipulieren...

plt.gray()
plt.imsave('/Users/.../myImage.png', img)
```