

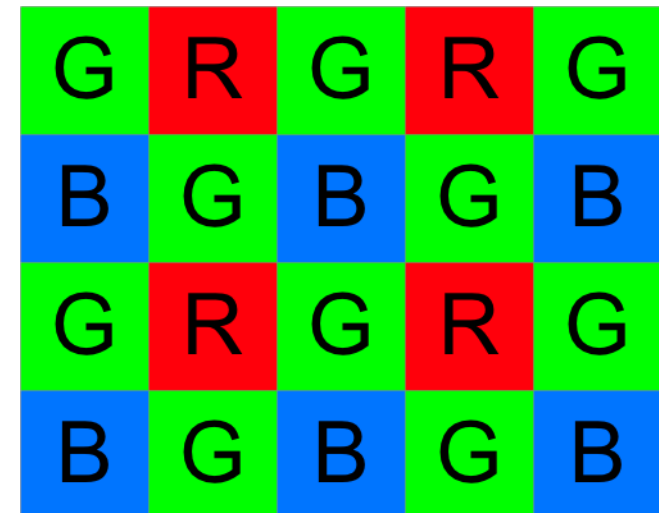
Kognitive Systeme

Bildverarbeitung II

2D-Filter in Orts- und Frequenzbereich

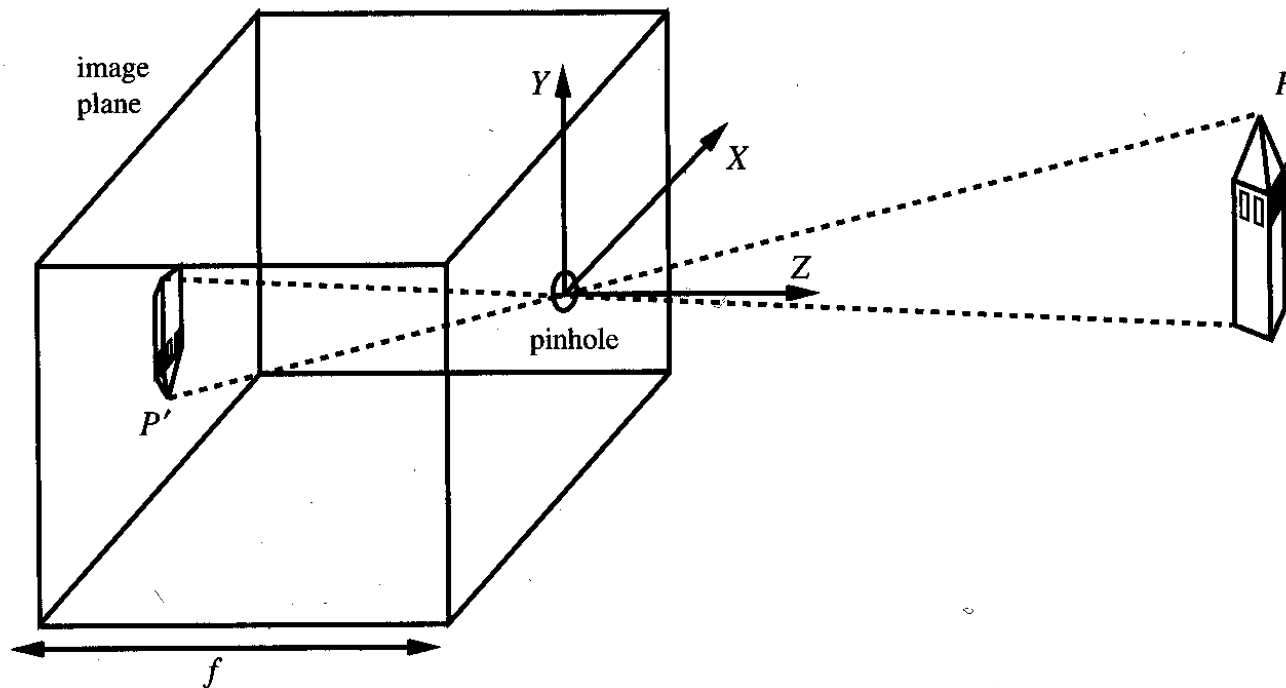
Mittwoch, 3. Mai 2017

- Sehr hochwertige Kameras, wie z.B. zum Filmen verwendet, besitzen drei Chips pro Pixel
- Die meisten Farbkameras haben einen Chip pro Pixel, der gegenüber der Farbe Rot, Grün oder Blau empfindlich ist
- Bei „Ein-Chip-Kameras“ wird meist das *Bayer-Pattern* verwendet
 - Um nach RGB24 zu konvertieren, muss interpoliert werden
 - Die Empfindlichkeit einer Ein-Chip-Kamera ist um Faktor 3 niedriger als die einer reinen Graustufen-Kamera!



Bilderzeugung: Lochkamera 1

- Einfachstes Modell auf Basis linearer Strahlenoptik :
Lochkameramodell



Interner Parameter: Brennweite f („Fokalabstand“)

Bildgenerierung: Lochkamera 2

Projektion eines Szenenpunktes $P = (X, Y, Z)$ auf einen Bildpunkt $p = (u, v, w)$:

$$\frac{-u}{f} = \frac{X}{Z}, \frac{-v}{f} = \frac{Y}{Z}, w = -f \Rightarrow X = -\frac{uZ}{f}, Y = -\frac{vZ}{f}$$

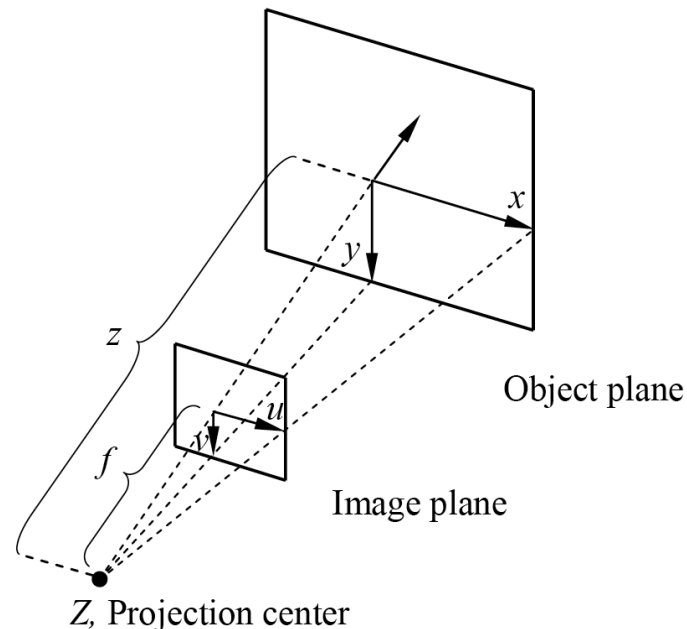
$$p = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u \\ v \\ -f \end{pmatrix} = -\frac{f}{Z} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = -\frac{f}{Z} P$$

Rückprojektion

Perspektivprojektion

Bei der Projektion geht die Z -Komponente verloren!

- Oft verwendete Variante:
Lochkameramodell *in Positivlage*:
 - Projektionszentrum C liegt hinter der Bildebene
 - Dadurch: keine Spiegelung (Minuszeichen entfallen)



Bildverarbeitung

Bildverarbeitung: Grundoperationen

- Homogene Punktoperatoren
- Histogrammauswertung
- Filterung
- Geometrische Operationen



Unterdrückung von Störungen
„Verschönern“ von Bildern
Verformen von Bildern

Homogene Punktoperatoren

- Anwendung: $\text{Img}'(u,v) = f(\text{Img}(u,v))$
- Unabhängig von der Position bzw. den Nachbarn des Pixels
- Implementierung der Funktion f oftmals als Look-Up-Table (Hardware)

- 2D Fourier-Transformation
- Bildverarbeitung im Frequenzbereich
- Bildverarbeitung im Ortsbereich durch Faltung
- Canny-Kantendetektor

Homogene Punktoperatoren

- Anwendung: $\text{Img}'(u,v) = f(\text{Img}(u,v))$
- Unabhängig von der Position bzw. den Nachbarn des Pixels
- Implementierung der Funktion f oftmals als Look-Up-Table (Hardware)

- Algorithmus für die Anwendung eines affinen Punktoperators

Algorithmus 8 AffinePunktoperation(I, a, b) $\rightarrow I'$

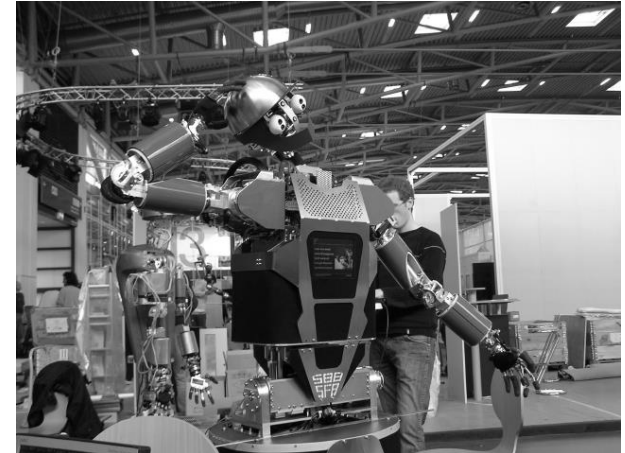
for all pixels (u, v) in I **do**
 $g := \text{round}(a \cdot I(u, v) + b)$
 $I'(u, v) := \begin{cases} 0 & \text{falls } g < 0 \\ q & \text{falls } g > q \\ g & \text{sonst} \end{cases}$
end for

Affine Punktoperatoren 3

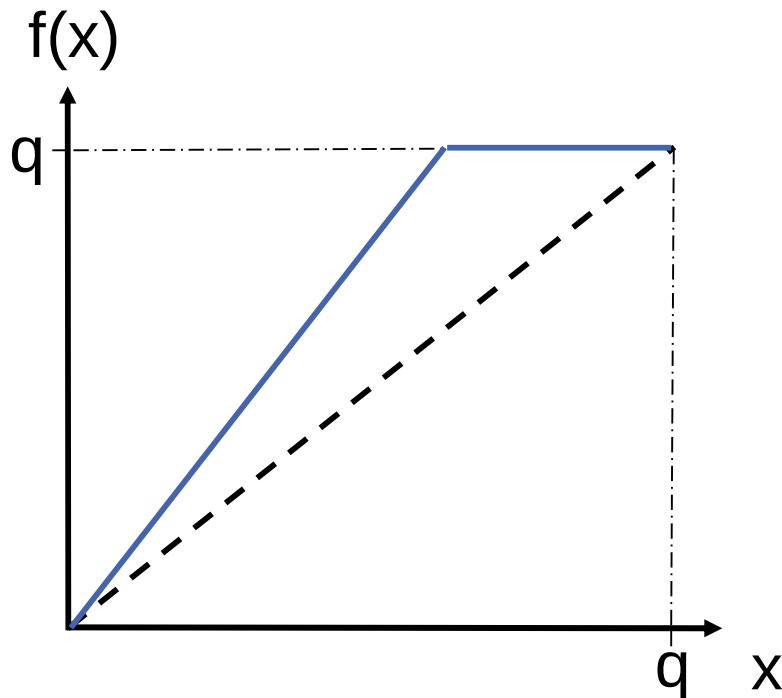
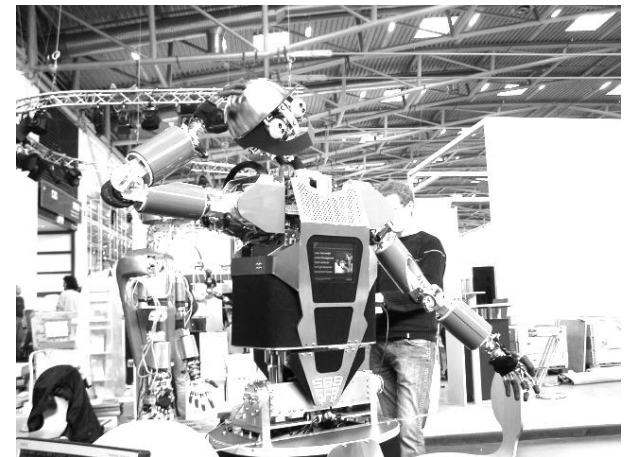
Kontrasterhöhung:
 $b = 0; a > 1$

Beispiel: $f(x) = 2x$

Original



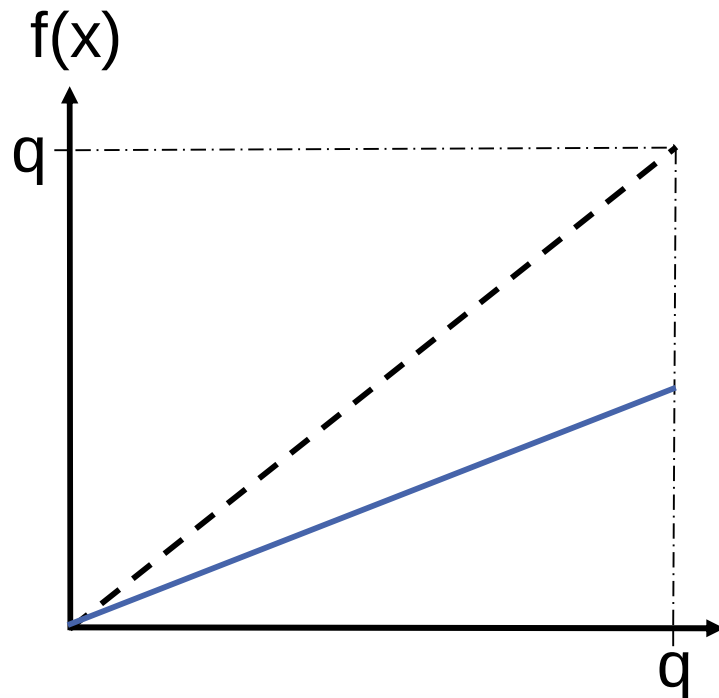
Ergebnis



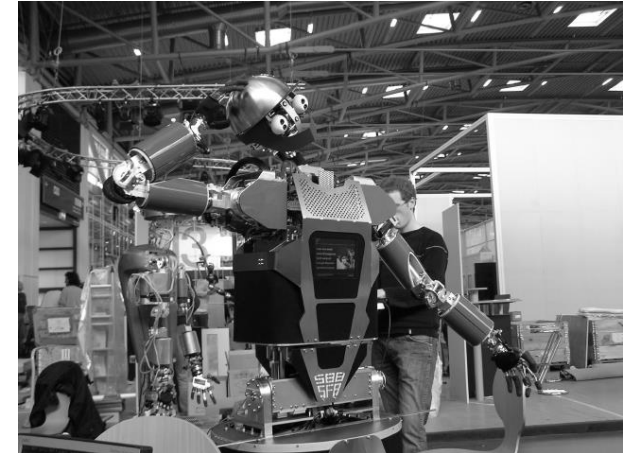
Affine Punktoperatoren 4

Kontrastverminderung:
 $b = 0; a < 1$

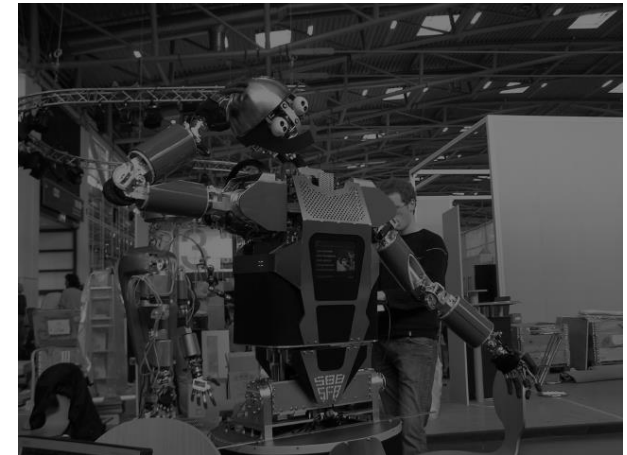
Beispiel: $f(x) = 0.5x$



Original



Ergebnis

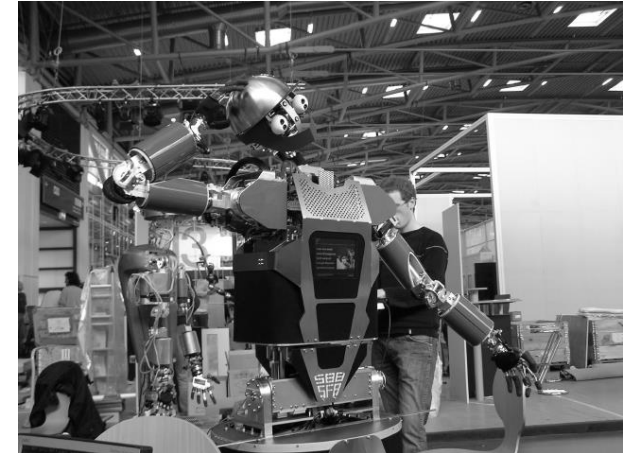


Affine Punktoperatoren 6

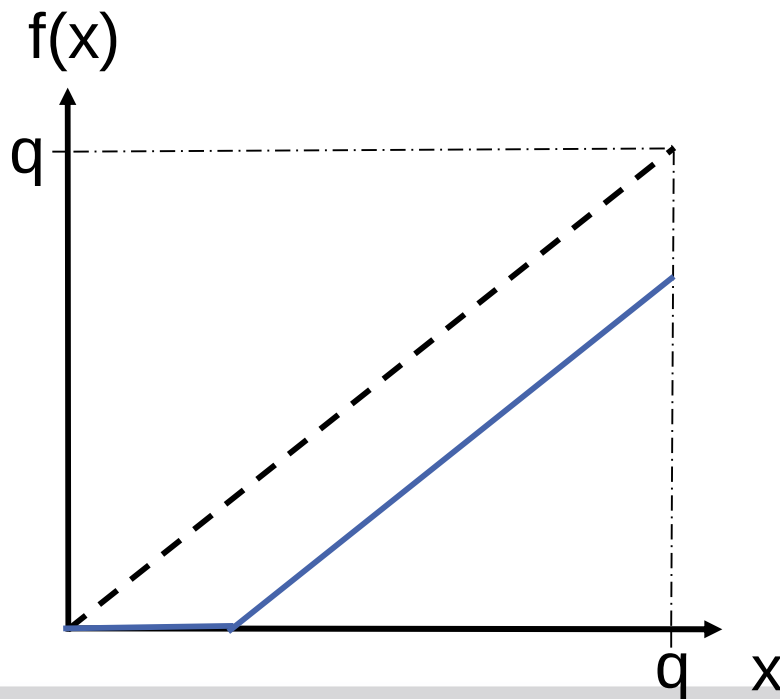
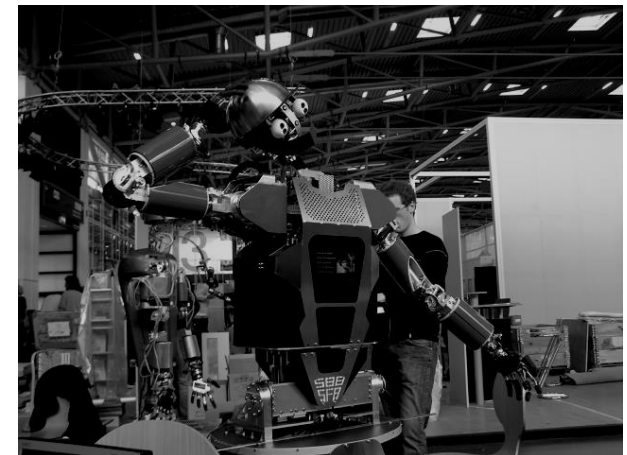
Helligkeitsverminderung:
 $b < 0; a = 1$

Beispiel: $f(x) = x - 50$

Original



Ergebnis

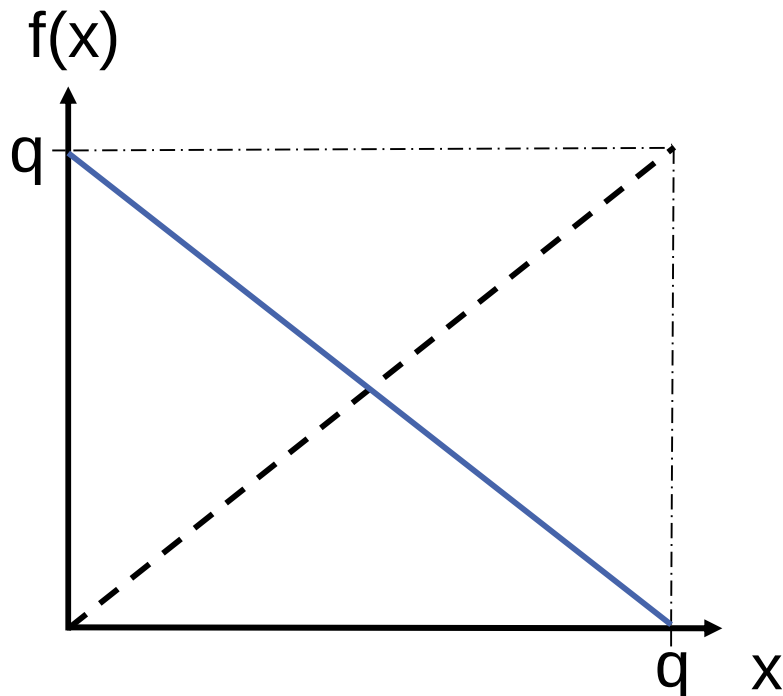


Affine Punktoperatoren 7

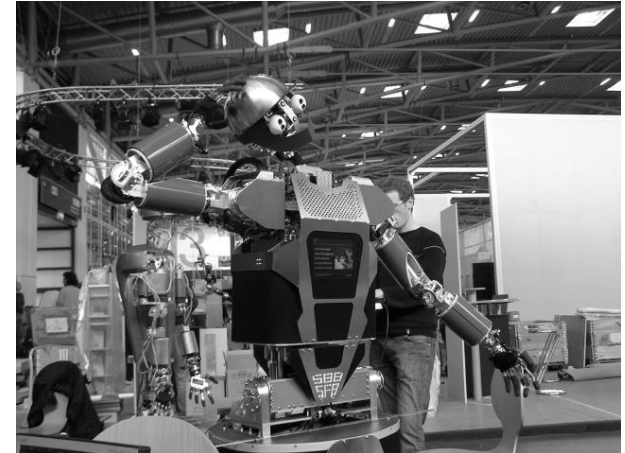
Invertierung:

$$b = q; a = -1$$

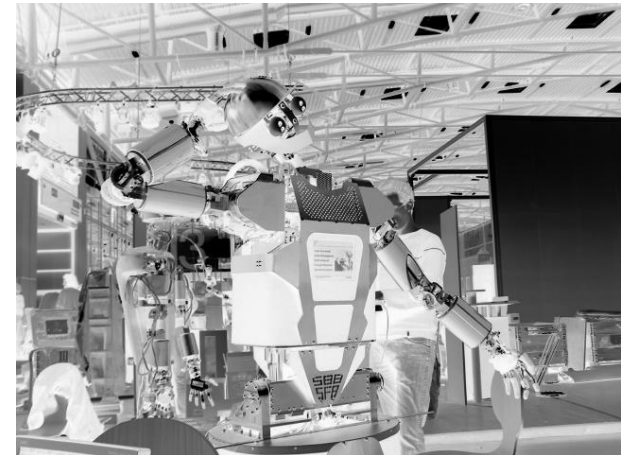
Beispiel: $f(x) = -x + q$



Original



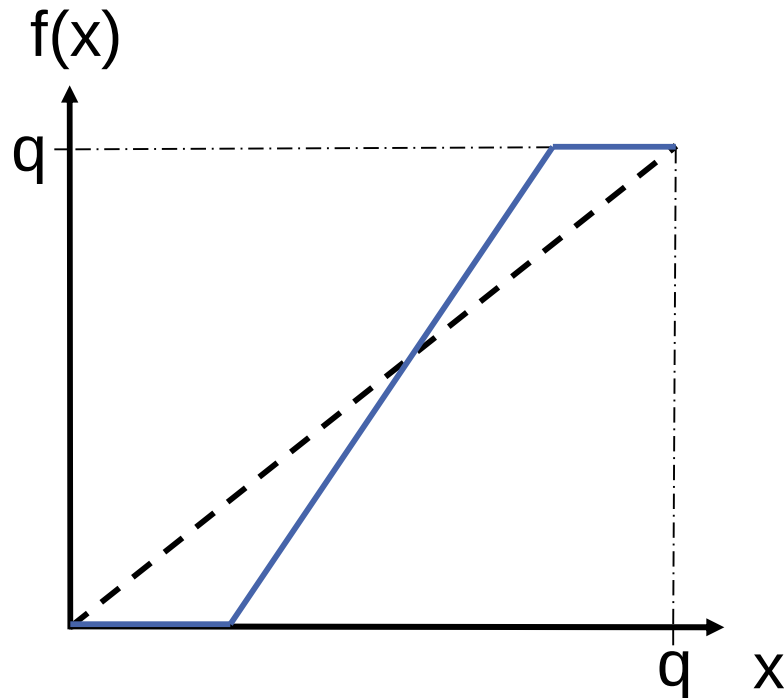
Ergebnis



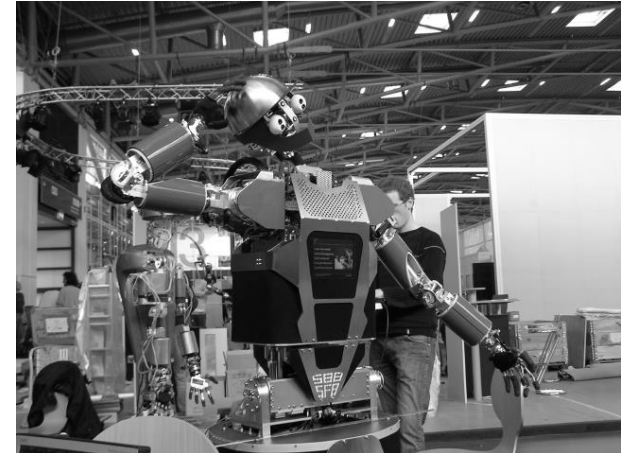
Affine Punktoperatoren 8

Kombinationen:

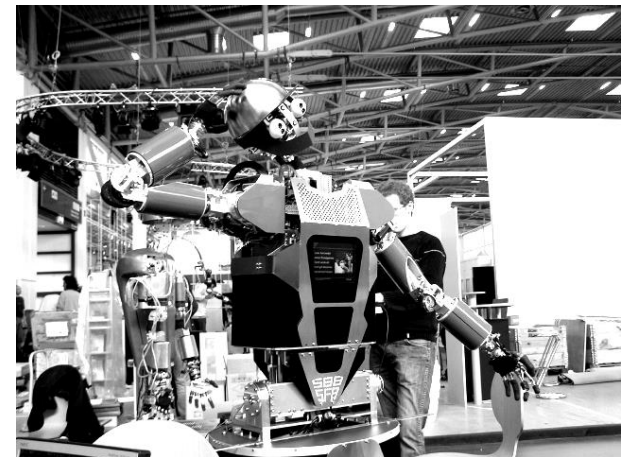
Beispiel: $f(x) = 2x - 50$



Original

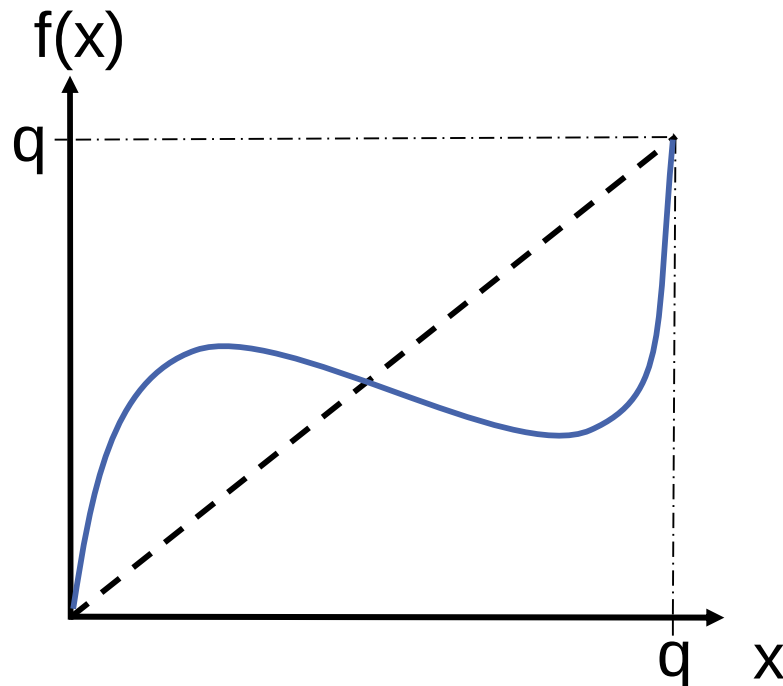


Ergebnis

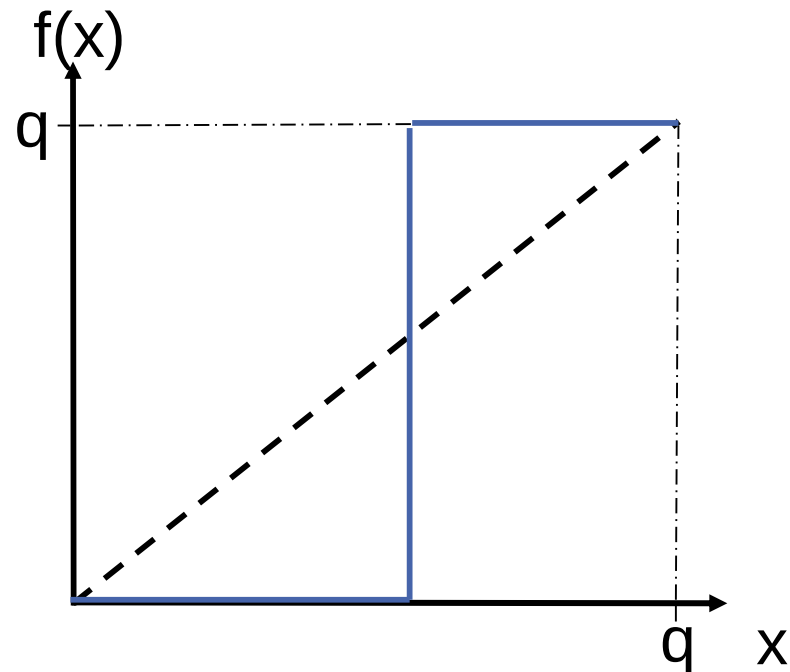


Nicht- affine Punktoperatoren

Beliebige Abbildungsfunktion $f : [0..q] \rightarrow [0..q]$



Sensor-Ausgleich



Binarisierung

Nicht-affine Punktoperatoren

Anwendung:

- Ausgleich von Sensor-Nichtlinearitäten
- Gewichtung
- Binarisierung

- Spreizung
- Histogrammdehnung
- Histogrammausgleich

- Die Spreizung berechnet in einem Bild zunächst die minimale Intensität min und die maximale Intensität max
- Anschließend wird das Intervall $[min, max]$ linear auf $[0, 255]$ abgebildet und es werden entsprechend die Intensitäten im Bild angepasst
- Die Spreizung ist eine affine Punktoperation
- Nachteil:
 - In der Praxis nicht robust
 - Nur unter optimalen Bedingungen anwendbar

Abbildungsvorschrift: $I'(u, v) = q \cdot \frac{I(u, v) - \min}{\max - \min}$

Als affine Punktoperation: $I'(u, v) = \frac{q}{\max - \min} \cdot I(u, v) - \frac{q \cdot \min}{\max - \min}$

Algorithmus 11 BerechneMinMax(I) $\rightarrow \min, \max$

```
min := q, max := 0
for all pixels (u, v) in I do
  if  $I(u, v) < \min$  then
    min :=  $I(u, v)$ 
  end if
  if  $I(u, v) > \max$  then
    max :=  $I(u, v)$ 
  end if
end for
```

Algorithmus 12 Spreize(I) $\rightarrow I'$

$\min, \max \leftarrow \text{BerechneMinMax}(I)$ {Algorithmus 11}

```
if  $\min = \max$  then
  STOP {Bild I ist homogen}
end if
```

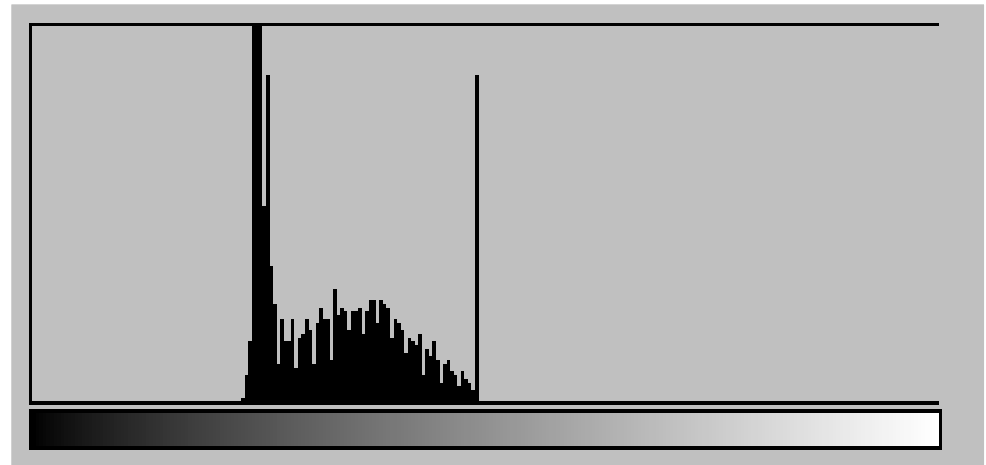
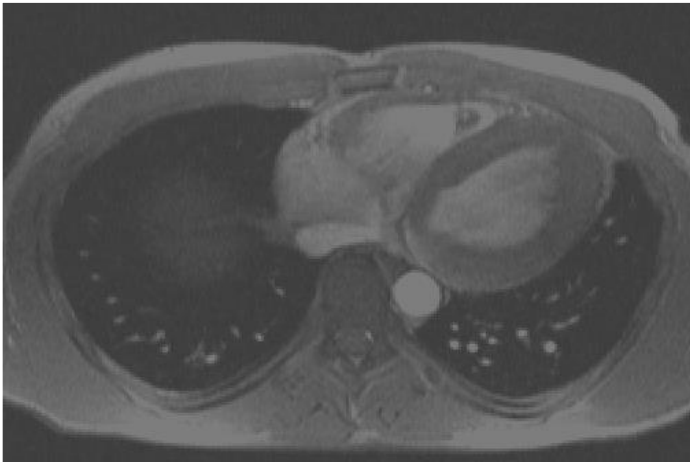
```
a :=  $\frac{q}{\max - \min}$ 
b :=  $-\frac{q \cdot \min}{\max - \min}$ 
```

$I' \leftarrow \text{AffinePunktoperation}(I, a, b)$ {Algorithmus 8}

Grauwerthistogramm

- Histogrammfunktion: gibt die Häufigkeit eines Merkmals an
Normalerweise: Grauwert

$$H(x) = \#(u, v) : I(u, v) = x, x \in [0..q-1]$$



Berechnung des Grauwertistogramms

Algorithmus 9 BerechneGrauwertHistogramm(I) $\rightarrow H$

for $x := 0$ to q do

$H(x) := 0$

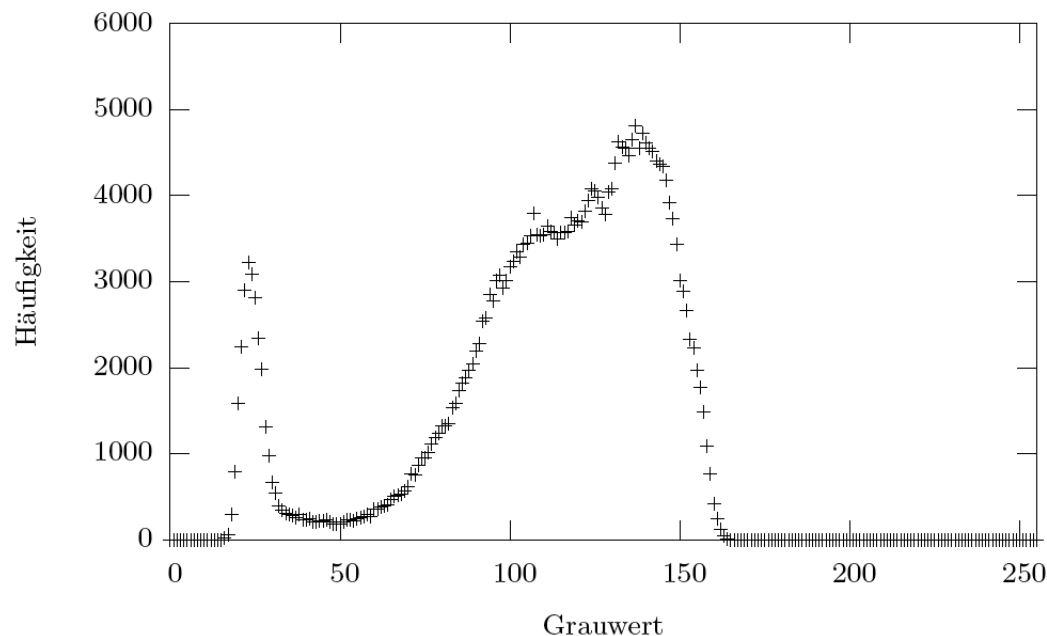
end for

for all pixels (u, v) in I do

$H(I(u, v)) := H(I(u, v)) + 1$

end for

Bei einem 8 Bit Graustufenbild
ist $q = 2^8 - 1 = 255$.



- Die Histogrammdehnung ist eine Verbesserung der Spreizung
- Anstatt der minimalen bzw. maximalen Intensität werden Quantile verwendet, um tatsächliche Maxima im Histogramm zu erkennen
- Die Histogrammdehnung ist eine affine Punktoperation
- Definition der Quantilsfunktion $H_q(p)$ für die Anwendung auf Histogrammen ($p \in [0, 1]$):

$$H_a(x) := \sum_{k=0}^x H(k)$$

$$H_q(p) := \inf \{x \in \{0, \dots, q\} : H_a(x) \geq p \cdot H_a(q)\}$$

Algorithmus 13 BerechneAkkumuliertesHistogramm(H) $\rightarrow H_a$

```
 $H_a(0) := H(0)$   
for  $x := 1$  to  $q$  do  
   $H_a(x) := H_a(x - 1) + H(x)$   
end for
```

Algorithmus 14 BerechneQuantil(H_a, p) $\rightarrow v$

```
for  $x := 0$  to  $q$  do  
  if  $H_a(x) \geq p \cdot H_a(q)$  then  
     $v := x$   
    RETURN  
  end if  
end for
```

Algorithmus 15 HistogrammDehnung($I, p_{min} = 0, 1, p_{max} = 0, 9$) $\rightarrow I'$

```
 $H \leftarrow$  BerechneGrauwertHistogramm( $I$ ) {Algorithmus 9}  
 $H_a \leftarrow$  BerechneAkkumuliertesHistogramm( $H$ ) {Algorithmus 13}  
 $min \leftarrow$  BerechneQuantil( $H_a, p_{min}$ ) {Algorithmus 14}  
 $max \leftarrow$  BerechneQuantil( $H_a, p_{max}$ ) {Algorithmus 14}
```

```
if  $min = max$  then  
  STOP {Bild  $I$  ist homogen}  
end if
```

```
 $a := \frac{q}{max-min}, b := -\frac{q \cdot min}{max-min}$   
 $I' \leftarrow$  AffinePunktoperation( $I, a, b$ ) {Algorithmus 8}
```

Algorithmus 13 BerechneAkkumuliertesHistogramm(H) $\rightarrow H_a$

```
 $H_a(0) := H(0)$   
for  $x := 1$  to  $q$  do  
   $H_a(x) := H_a(x - 1) + H(x)$   
end for
```

Algorithmus 14 BerechneQuantil(H_a, p) $\rightarrow v$

```
for  $x := 0$  to  $q$  do  
  if  $H_a(x) \geq p \cdot H_a(q)$  then  
     $v := x$   
    RETURN  
  end if  
end for
```

Algorithmus 15 HistogrammDehnung($I, p_{min} = 0, 1, p_{max} = 0, 9$) $\rightarrow I'$

```
 $H \leftarrow$  BerechneGrauwertHistogramm( $I$ ) {Algorithmus 9}  
 $H_a \leftarrow$  BerechneAkkumuliertesHistogramm( $H$ ) {Algorithmus 13}  
 $min \leftarrow$  BerechneQuantil( $H_a, p_{min}$ ) {Algorithmus 14}  
 $max \leftarrow$  BerechneQuantil( $H_a, p_{max}$ ) {Algorithmus 14}
```

```
if  $min = max$  then  
  STOP {Bild  $I$  ist homogen}  
end if
```

```
 $a := \frac{q}{max-min}, b := -\frac{q \cdot min}{max-min}$   
 $I' \leftarrow$  AffinePunktoperation( $I, a, b$ ) {Algorithmus 8}
```

- Der Histogrammausgleich erhöht den Kontrast in Bereichen des Histogramms mit vielen Treffern
- Dadurch werden stark vertretene Grauwertbereiche für das Auge sichtbarer gemacht
- Der Histogrammausgleich ist eine homogene Punktoperation, jedoch keine affine Punktoperation
- Nachteil:
Obwohl der Histogrammausgleich in der Praxis in der Regel den Kontrast erhöht, kann er auch zu einer Verminderung des Kontrasts führen (siehe Übung)

Algorithmus 16 HistogrammAusgleich(I) $\rightarrow I'$

$H \leftarrow \text{BerechneGrauwertHistogramm}(I)$ {Algorithmus 9}

$H_a \leftarrow \text{BerechneAkkumuliertesHistogramm}(H)$ {Algorithmus 13}

for $x := 0$ **to** q **do**

$H_n(x) := \text{round}(\frac{q \cdot H_a(x)}{H_a(q)})$

end for

for all pixels (u, v) in I **do**

$I'(u, v) := H_n(I(u, v))$

end for

Beispiele zur Kontrastanpassung

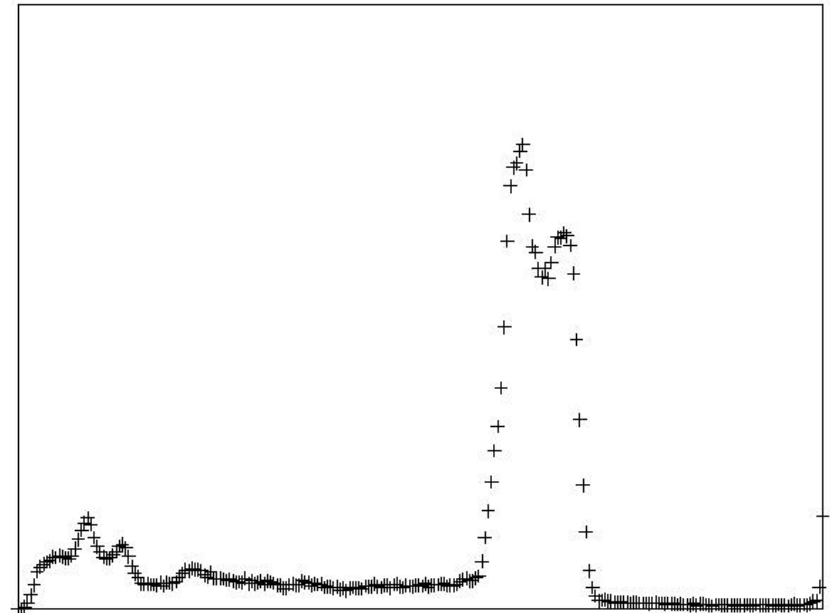
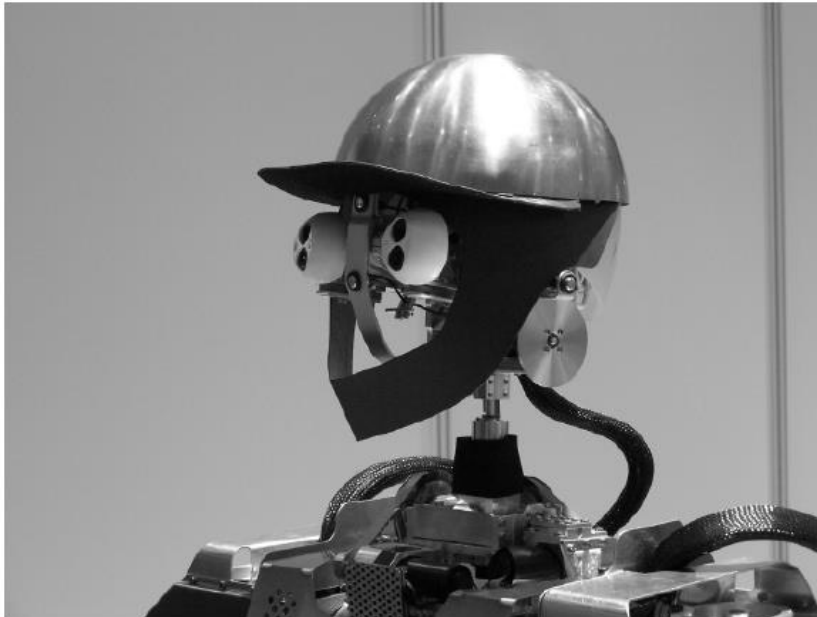
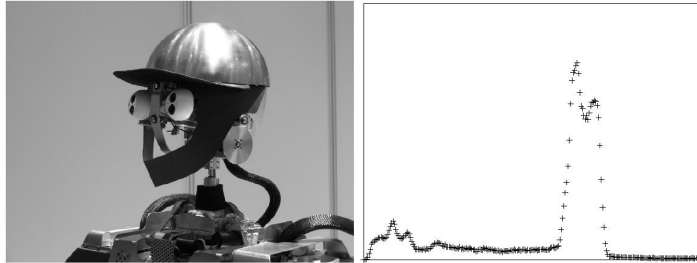
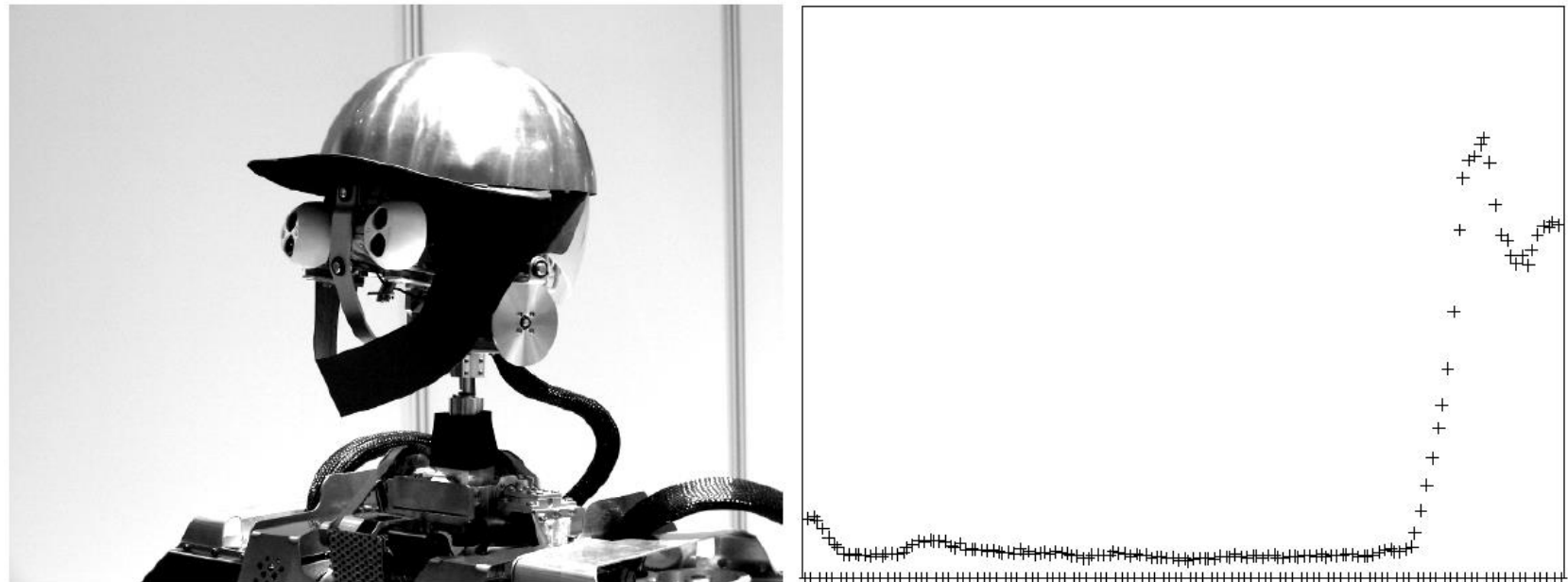


Bild des Kopfes des humanoiden Roboters ARMAR III.
Links: Eingabebild. Rechts: Grauwerthistogramm.

Beispiel Histogrammdehnung



Eingabebild mit
Grauwerthistogramm.



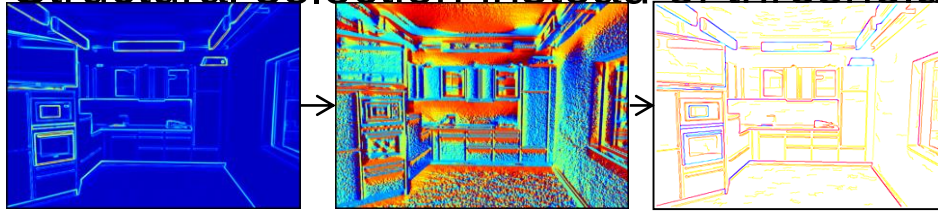
Ergebnis einer Histogrammdehnung mit $p_{\min} = 0,1$ und $p_{\max} = 0,9$.
Links: Ergebnisbild. Rechts: Grauwerthistogramm.

Visual Feature Extraction

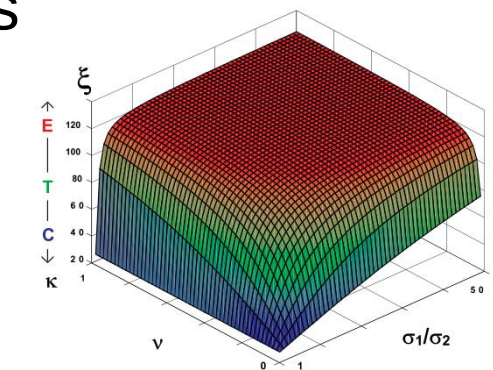
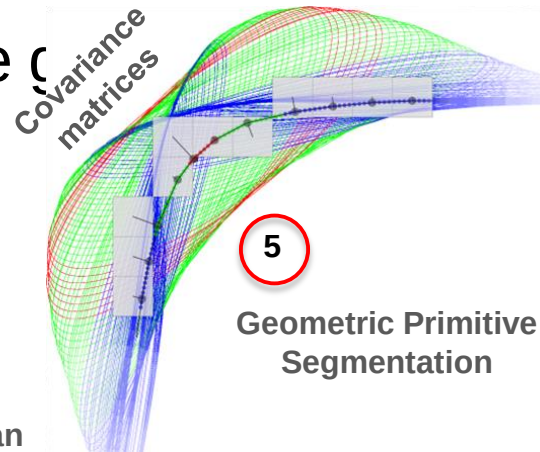
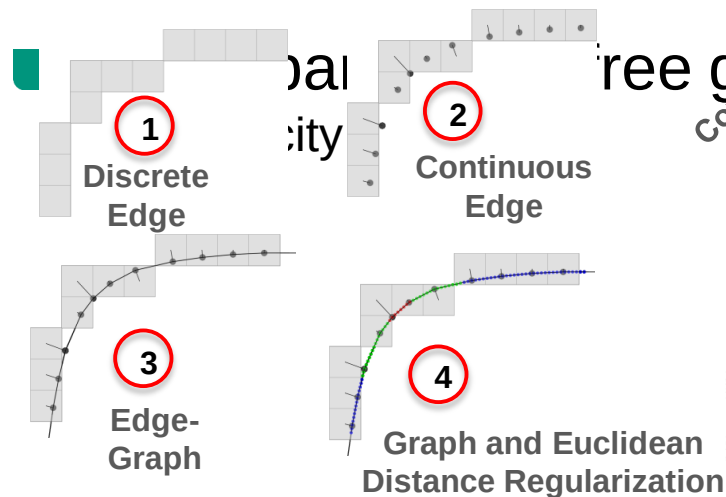
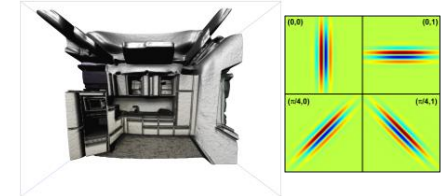
Edge extraction [Grigorescu et al. 2006]

Gabor kernels

Structural selection instead of thresholds

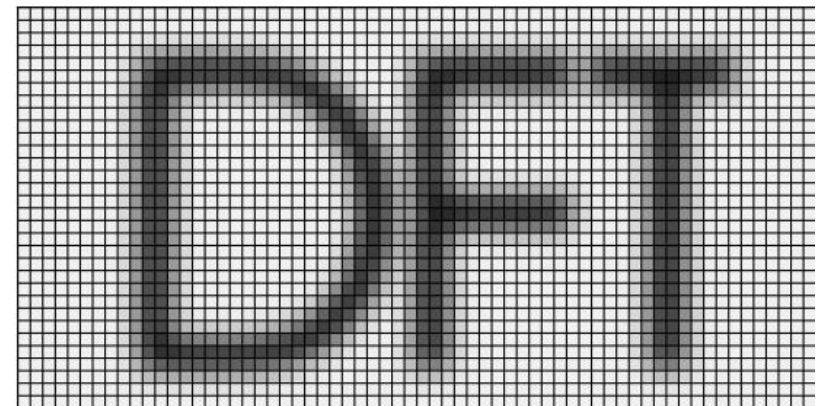
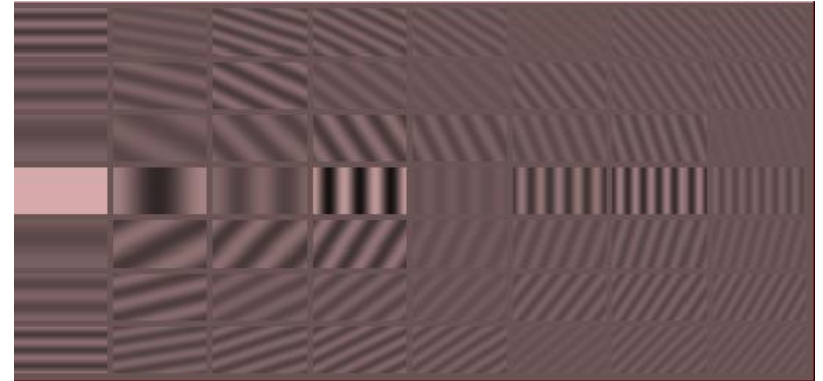


$$\Gamma(\mathbf{x}) = \sum_{p=0}^n \sum_{q=0}^1 U_{\langle p\pi/2n \rangle} \left\{ \Phi(\mathbf{x}) \otimes \Psi^{(p\pi/2n, q)} \right\}$$



Eccentricity Segmentation Model

- (Grauwert-)Bilder lassen sich signaltheoretisch als Summe verschiedenfrequenter Signale betrachten
- Niedrige Frequenzen:
Schwache
Grauwertübergänge
- Hohe Frequenzen:
Scharfe Grauwertübergänge



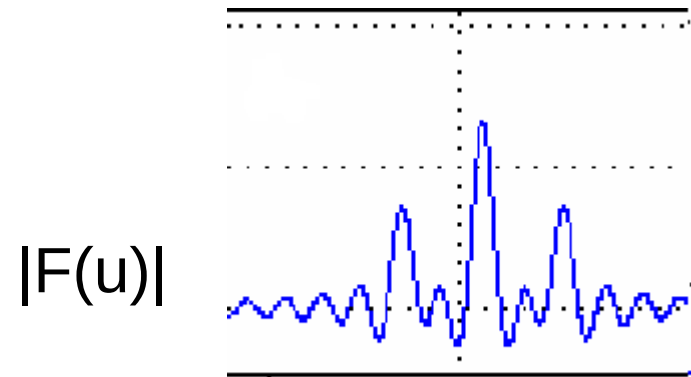
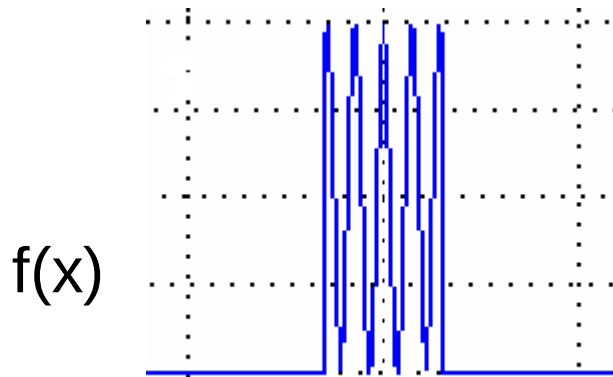
Fourier-Transformation in der Bildverarbeitung

- Gewichtete Summation aller Bildpunkte
- Zerlegung des Bildes in Sinus- und Kosinusfunktionen
- Je weiter ein Punkt im Spektrum vom Bildmittelpunkt entfernt, desto höher seine darstellende Frequenz u bzw. v
- D.h. im Bildinneren tiefe Frequenzen, in äußeren Bereichen hohe Frequenzen
- Anwendung
 - Bildanalyse (z.B. Muster-, Geschwindigkeitserkennung)
 - Bildfilterung (z.B. Tiefpass)
 - Bildkompression

- 1-dim. Fourier-Transformation kontinuierlich

$$f(x) = \int_{-\infty}^{\infty} F(u) \exp(2i\pi ux) du$$

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp(-2i\pi ux) dx$$



- Diskretisierte Verarbeitung im Rechner durch Abtastung mit Impulskamm
- Überlappung der Spektren (Aliasing) bei Nichteinhaltung des Abtasttheorems $f_A \geq 2 * f_{\max}$
- Diskretes Signal $f(x)$ -> Periodisches Signal $F(u)$
- Abgetastetes Signal $y_+ = y(nT)$

- Transformiertes Signal
$$Y_+(u) = \sum_{n=-\infty}^{\infty} y_n \exp(-i2\pi unT)$$

■ Kontinuierliches 2-dimensionales Signal

$$F(u, v) = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} f(x, y) \exp(-2i\pi(ux + vy)) dx dy$$

■ Diskretes 2-dimensionales Signal

$$F(u, v) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f[x, y] \exp(-2i\pi(ux + vy)T)$$

Fourier-Transformation in der Bildverarbeitung

$$F(u,v) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f[x,y] \exp(-2i\pi(ux + vy)T)$$

DFT bei Bild mit Bildgröße $[0 \leq x < M][0 \leq y < N]$:

$$F(u,v) = \sum_{y=0}^{N-1} \left(\sum_{x=0}^{M-1} f[x,y] \exp(-2i\pi(ux/M)) \right) * \exp(-2i\pi(vy/N))$$

DFT bei quadratischem Bild mit Bildgröße $[0 \leq x < N][0 \leq y < N]$:

$$F(u,v) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f[x,y] \exp(-2i\pi(ux + vy)/N)$$

Anschaulich: Durchführung der 1D-DFT auf jeder Zeile und Speicherung der Daten in Matrix (innere Klammer)
Durchführung der 1D-DFT auf jeder Spalte der ermittelten Matrix

- Die Variablen u und v heißen Frequenzvariablen
- $F(u, v)$ ist komplexe Funktion
- $F(u, v)$ ist darstellbar als zwei Bilder
 - in Realteil und Imaginärteil

$$F(u, v) = R(u, v) + I(u, v)$$

- Oder Betrag (auch Spektrum genannt, oft logarithmisch dargestellt) und Phase

$$F(u, v) = |F(u, v)| * \exp(i\varphi(u, v))$$

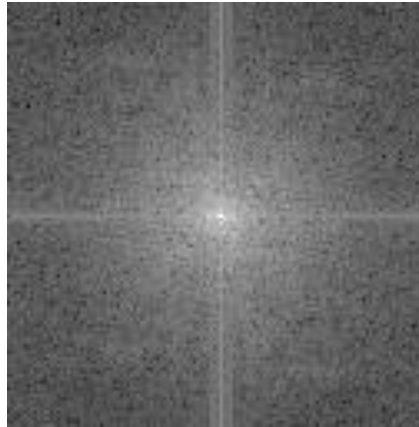
- Quadrat des Spektrum heißt spektrale Dichte

Fourier-Transformation

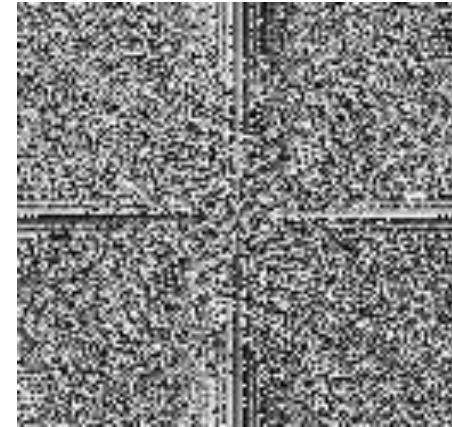
■ Beispiel



Original
 $f(x,y)$



Spektrum
 $\log(|F(u,v)|)$

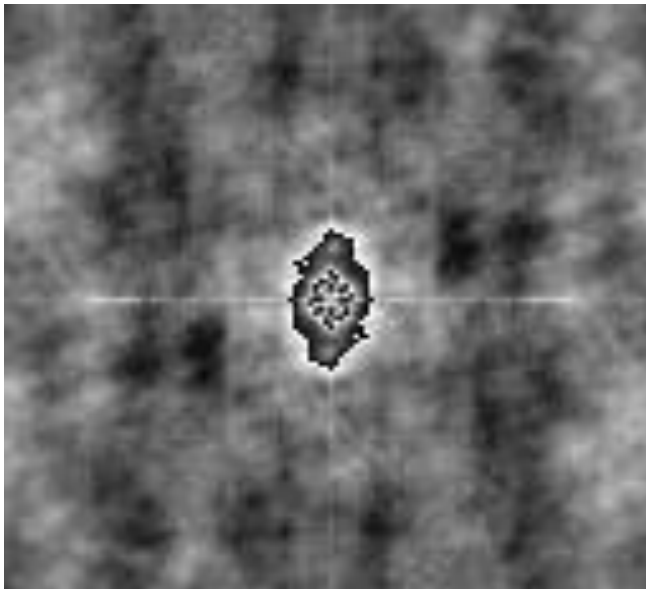


Phase
 $\text{Ang}(F(u,v))$

Fourier-Transformierte $F(u,v)$ ist eine komplexe Funktion

- Rekonstruktion ausschließlich durch

Spektrum

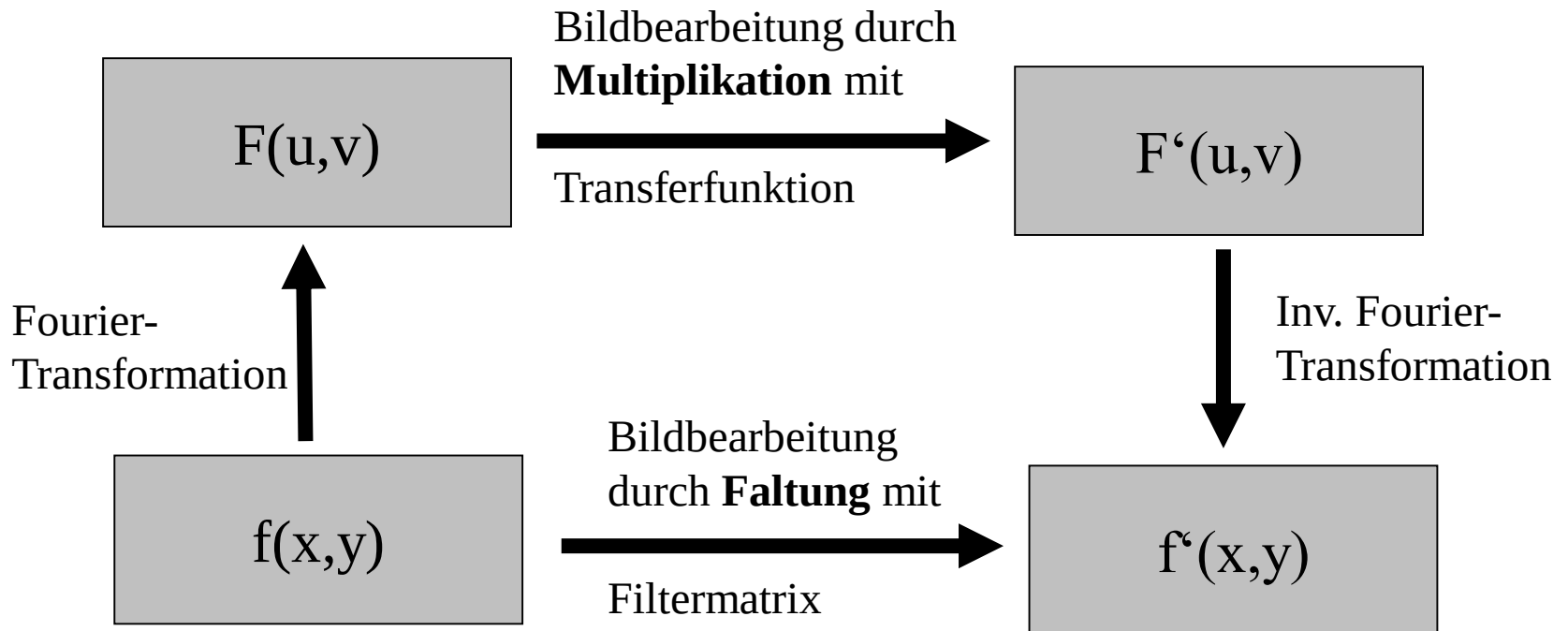


Phase

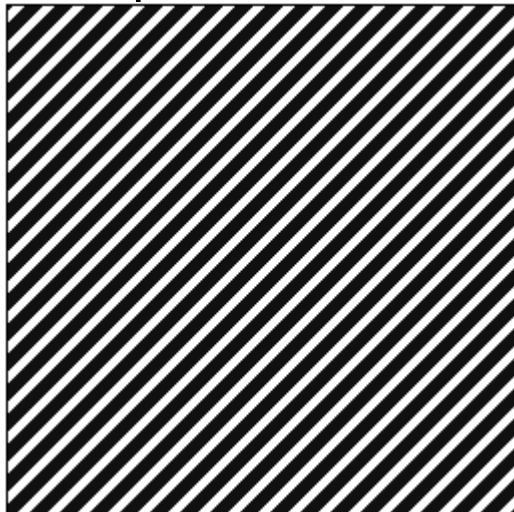


- Spektrum UND Phase sind für vollständige Rekonstruktion notwendig

- Durch Filter im Ortsbereich oder Transferfunktion im Frequenzbereich
- Ortsbereich:
 - Manipulation von Grauwerten
 - Anschaulich
 - Häufig: Inhomogene Punktoperationen, Glättung
- Frequenzbereich
 - Manipulation der Frequenzanteile
 - Keine unmittelbare bildliche Vorstellung
 - Häufig: starke Glättung, frequenzselektive Filter



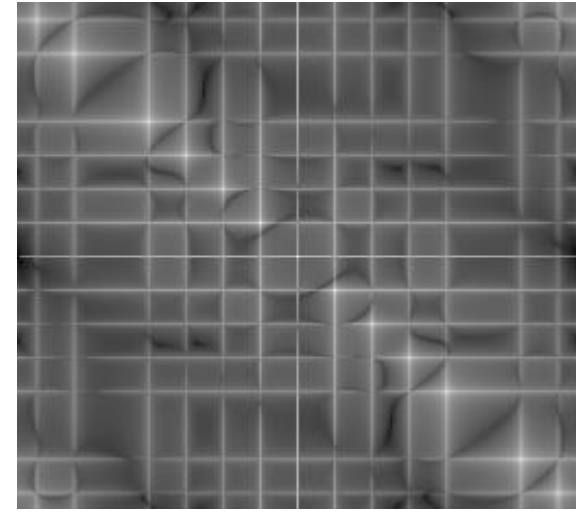
■ Beispiel



$f(x,y)$



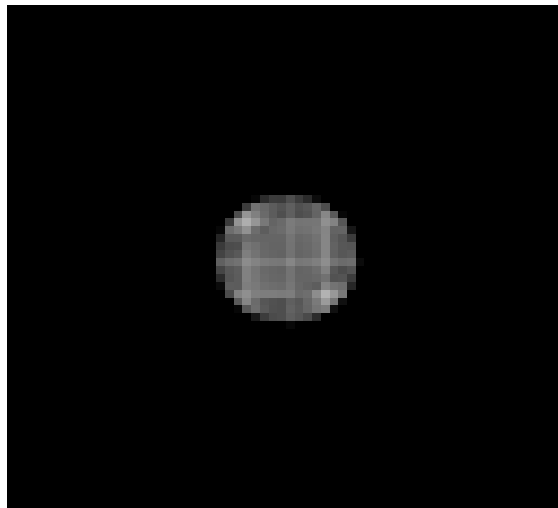
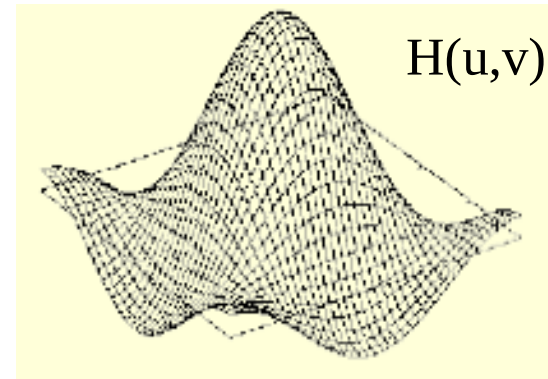
$|F(u,v)|$



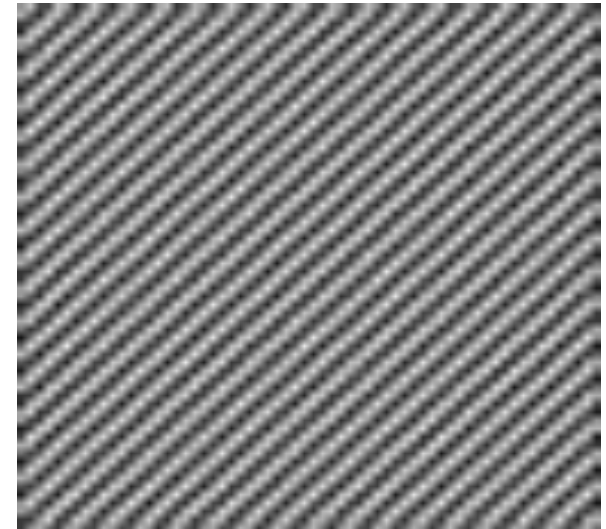
$\log(|F(u,v)|)$

Großer Spektralbereich belegt -> scharfe Übergänge

- Beispiel Tiefpassfilterung (Glättung) mit Transferfunktion $H(u,v)$: (entspricht der Fourier-Transformierten eines 3x3 Mittelwertfilters)



Spektrum



Rekonstruktion

- Filterung mit Radius $r = 32$ Pixel
→ unscharfe Übergänge

- Faltung zweier Funktionen 1D kontinuierlich

$$h(x) = f(x) * g(x) = \int_{a=-\infty}^{\infty} f(a)g(x-a)da$$

- Faltung zweier Funktionen 1D diskret

$$h[x] = f[x] * g[x] = \sum_{a=-\infty}^{\infty} f[a]g[x-a]$$

- Im Folgenden:

- f bezeichnet die Filterfunktion
- g bezeichnet die Bildfunktion

- Faltung zweier Funktionen 2D kontinuierlich

$$h(x, y) = f(x, y) * g(x, y) = \int_{a=-\infty}^{\infty} \int_{b=-\infty}^{\infty} f(a, b) g(x-a, y-b) da db$$

- Faltung zweier Funktionen 2D diskret

$$h[x, y] = f[x, y] * g[x, y] = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f[a, b] g[x-a, y-b]$$

- Bildmatrizen werden in den relevanten Randbereichen mit Nullen gefüllt
- Der neue Bildwert ist die gewichtete Summe der Pixel, die unter der gespiegelten Matrix liegen.
- Als Gewichte dienen die Matrizenwerte
- Bildfilterung ist die Faltung eines Bildes mit einer Filtermatrix bzw. Maske
- Anmerkung bzgl. Frequenzbereich:
Die Transferfunktion $F(u,v)$ ist die Fourier-Transformierte der Filterfunktion $f(x,y)$

Faltung im Ortsbereich

- Beispiel

$$f(x,y) * g(x,y) = h(x,y)$$

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

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y

	g'	h'	i'	
	l'	m'	n'	
	q'	r'	s'	

-1	0	1		
a	b	c	d	e
-2	0	2		
f	g	h	i	j
-1	0	1		
k	l	m	n	o
	p	q	r	s
	u	v	w	x
				y

$$g' = -a+c-2f+2h-k+m$$

	-1	0	1	
a	b	c	d	e
	-2	0	2	
f	g	h	i	j
	-1	0	1	
k	l	m	n	o
	p	q	r	s
	u	v	w	x
				y

$$h' = (d+2i+n)-(b+2g+l)$$

		-1	0	1
a	b	c	d	e
		-2	0	2
f	g	h	i	j
		-1	0	1
k	l	m	n	o
	p	q	r	s
	u	v	w	x
				y

$$i' = (e+2j+o)-(c+2h+m)$$

Hinweis: Die Filtermatrix wird hier der Einfachheit halber nicht gespiegelt. Dadurch ändern sich nur die Vorzeichen der Einträge der Ergebnismatrix.

- Tiefpassfilter: Glättung, Rauschelimination
 - Mittelwertfilter
 - Gauß-Filter

- Hochpassfilter: Kantendetektion
 - Prewitt
 - Sobel
 - Laplace
 - Roberts

- Kombinierte Operatoren
 - Laplacian of Gaussian

Mittelwertfilter

- Ziel: Rauschunterdrückung
- Beispiel:
Durchschnitt aus 8- Umgebung
und Punkt; Größe beliebig wählbar

$1/9 *$

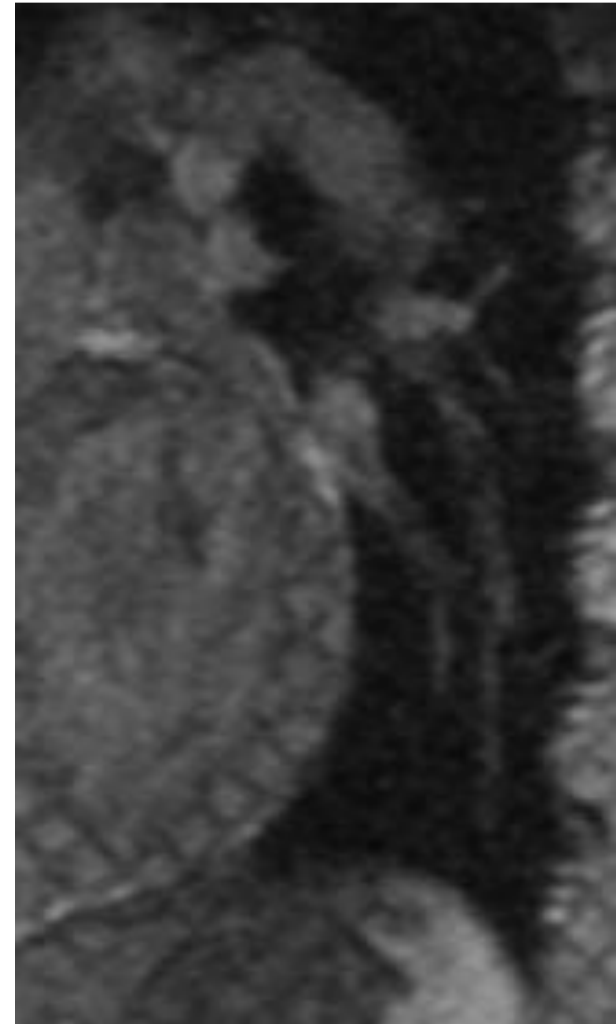
1	1	1
1	1	1
1	1	1



13	25	30	34	40	46	60	76
45	$(1/9)*50$	$(1/9)*52$	$(1/9)*55$	65	67	87	77
34	$(1/9)*45$	$(1/9)*55 \Rightarrow 52$	$(1/9)*60$	54	45	56	65
45	$(1/9)*50$	$(1/9)*52$	$(1/9)*48$	45	65	65	45
34	34	54	56	57	58	67	70
45	46	46	46	45	53	52	60
50	68	69	60	70	70	78	79
68	70	78	78	80	80	80	90

$$m'(x,y) = \sum_{j=-1}^1 \sum_{i=-1}^1 m(i,j) \cdot p(x-j, y-i)$$

Mittelwertfilter II



- Ziel: Rauschunterdrückung, Glättung
- Definiert durch zweidimensionale Gauß-Funktion (links: Ortsbereich, rechts: Frequenzbereich):

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \longleftrightarrow F(u,v) = e^{-\frac{u^2+v^2}{2}\sigma^2}$$

- Approximation von $f(x)$ (also für die Anwendung im Ortsbereich) durch einen 3×3 -Filter für $\sigma = 0,85$:

$$F_{Gau\beta} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

- Die Stärke der Glättung ist ausschließlich durch den Parameter σ bestimmt:
Je größer σ , umso stärker die Glättung.
- Die Größe $n \times n$ der Filtermaske beeinflusst die Güte der Approximation des Filters
- Im Ortsbereich kann als Faustregel für eine ausreichende Approximation verwendet werden:
$$n = \lfloor 2\sigma \rfloor \cdot 2 + 1$$
- Deshalb: Für eine starke Glättung lohnt sich die Anwendung im Frequenzbereich.

Beispiele Gauß-Filter



Originalbild

Beispiele Gauß-Filter



Ergebnis nach Anwendung des Gauß-Filters mit $\sigma = 1$

Beispiele Gauß-Filter



Ergebnis nach Anwendung des Gauß-Filters mit $\sigma^2 = 2$

Beispiele Gauß-Filter



Ergebnis nach Anwendung des Gauß-Filters mit $\sigma^2 = 4$

Beispiele Gauß-Filter



Ergebnis nach Anwendung des Gauß-Filters mit $\sigma^2 = 8$

Beispiele Gauß-Filter



Ergebnis nach Anwendung des Gauß-Filters mit $\sigma^2 = 16$

Darstellung der Filterantworten in den nachfolgenden Beispielen

Legende

Helles Grau

Filterausgabe > 0

Dunkles Grau

Filterausgabe < 0

Hintergrundfarbe

Filterausgabe $= 0$



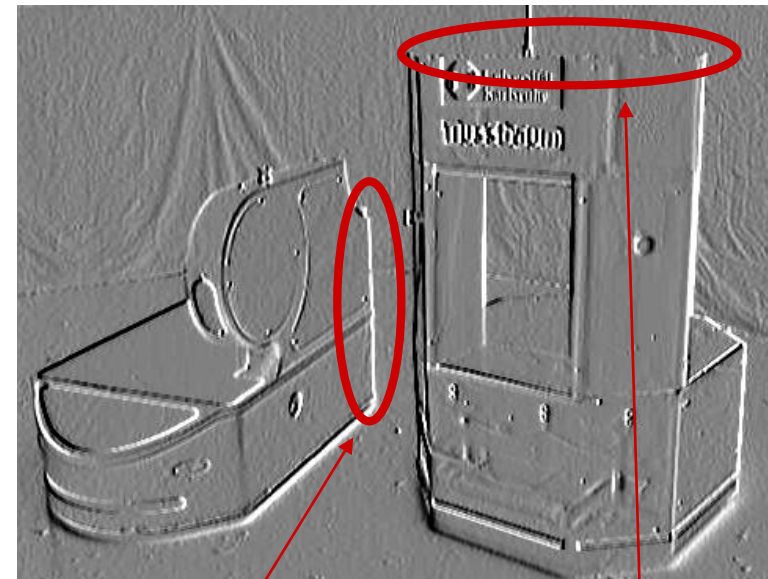
Alternativ: Betragsmäßige Darstellung (Hintergrund schwarz)

Prewitt-X Filter

$$P_x = \frac{\partial g(x, y)}{\partial x}$$

■ Approximiert durch

$$p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$



gute vertikale

schlechte horizontale

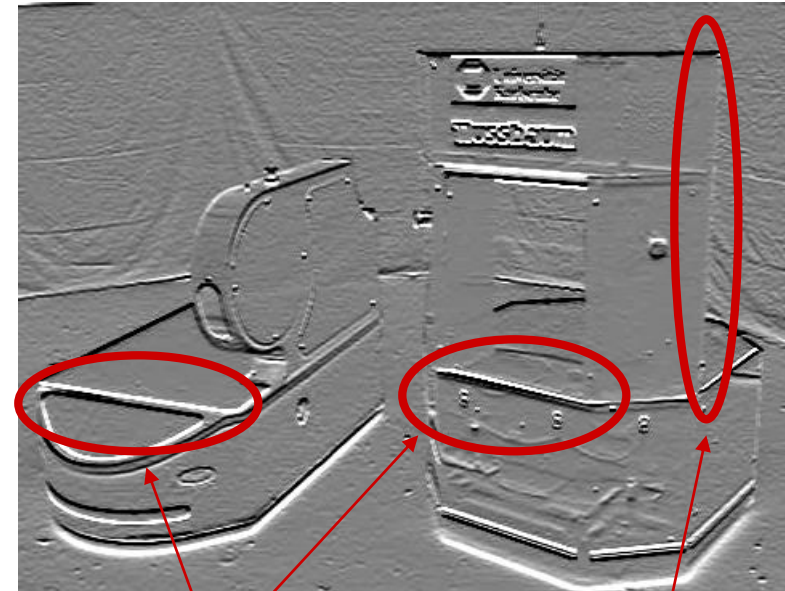
Kantendetektion

Prewitt-Y Filter

$$P_y = \frac{\partial g(x, y)}{\partial y}$$

■ Approximiert durch

$$p_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$



gute horizontale

schlechte vertikale

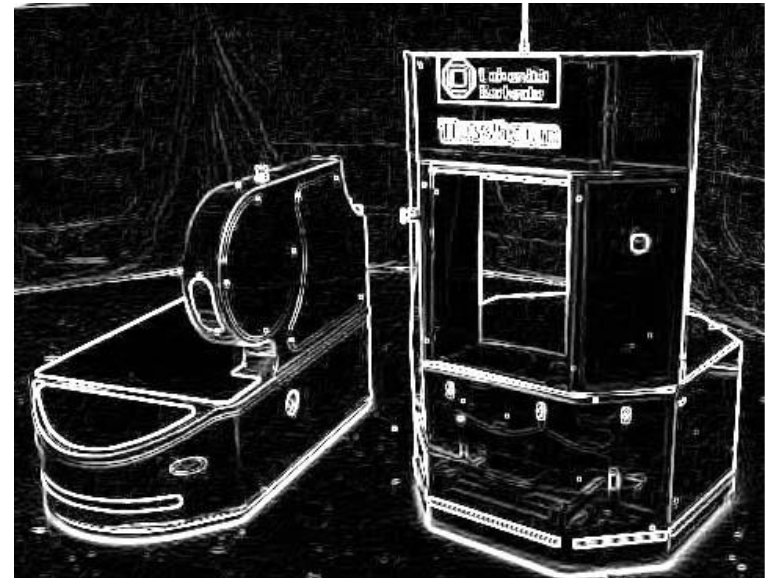
Kantendetektion

Prewitt-Operator

- Kombination der Prewitt-Filter zur Bestimmung des Grauwertgradientenbetrages M

$$M \approx \sqrt{P_x^2 + P_y^2}$$

- Danach: Schwellwertfilterung

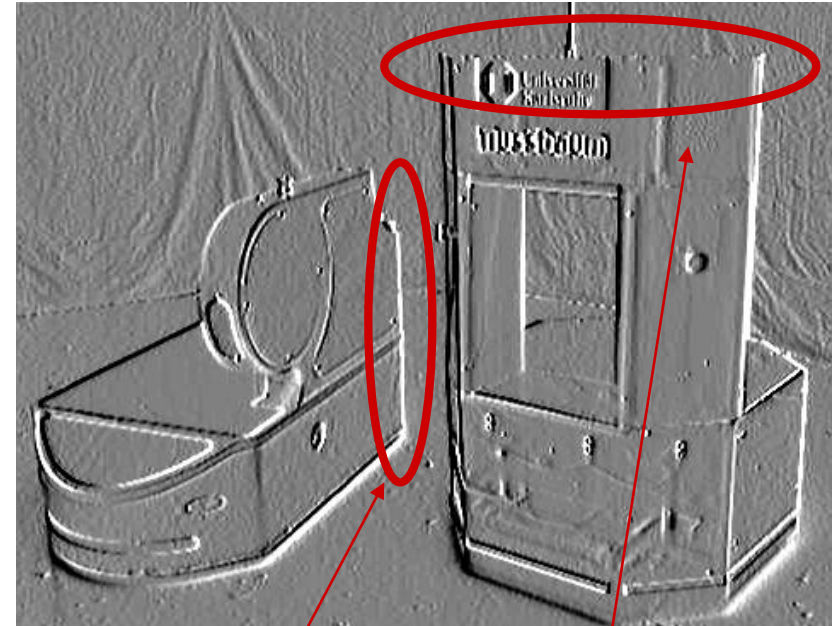


Sobel-X Filter

$$S_x = \frac{\partial g(x,y)}{\partial x}$$

■ Approximiert durch

$$s_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$



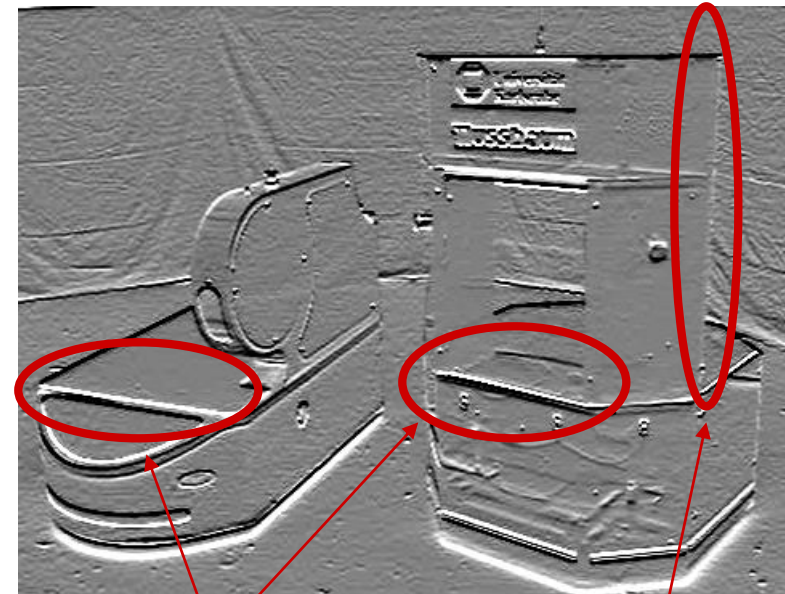
gute vertikale schlechte horizontale
Kantendetektion

Sobel-Y Filter

$$S_y = \frac{\partial g(x,y)}{\partial y}$$

■ Approximiert durch

$$s_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$



gute horizontale

schlechte vertikale

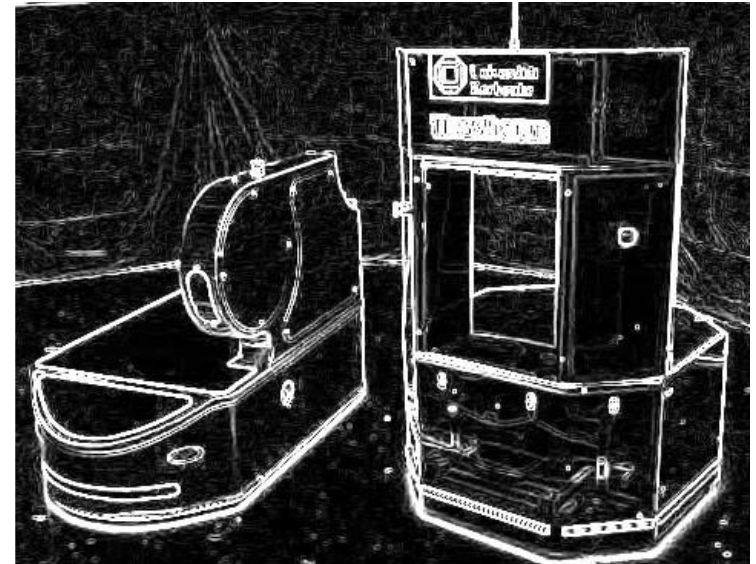
Kantendetektion

Sobel-Operator

- Kombination der Sobel-Filter zur Bestimmung des Grauwertgradienten-Betrages M

$$M \approx \sqrt{S_x^2 + S_y^2}$$

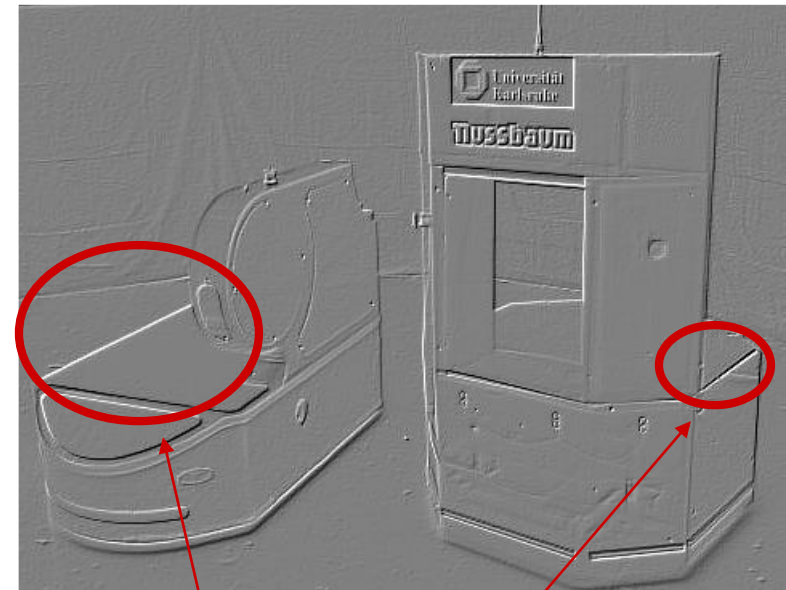
- Danach: Schwellwertfilterung



■ Roberts-Filter:

$$R(g(x,y)) = |R_x(g(x,y))| + |R_y(g(x,y))|$$

$$R_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$



Gute diagonale
Kantendetektion

Laplace-Operator:

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

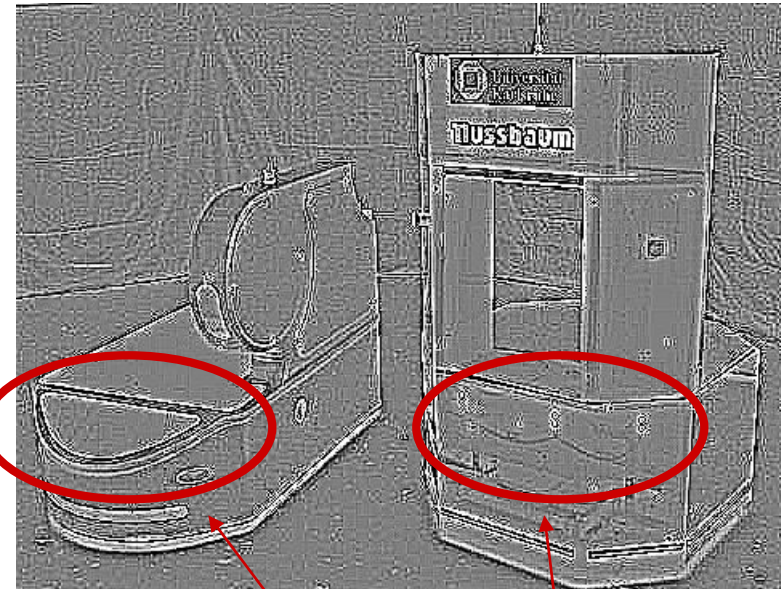
$$\nabla^2 \approx \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



Nulldurchgänge markieren Kanten

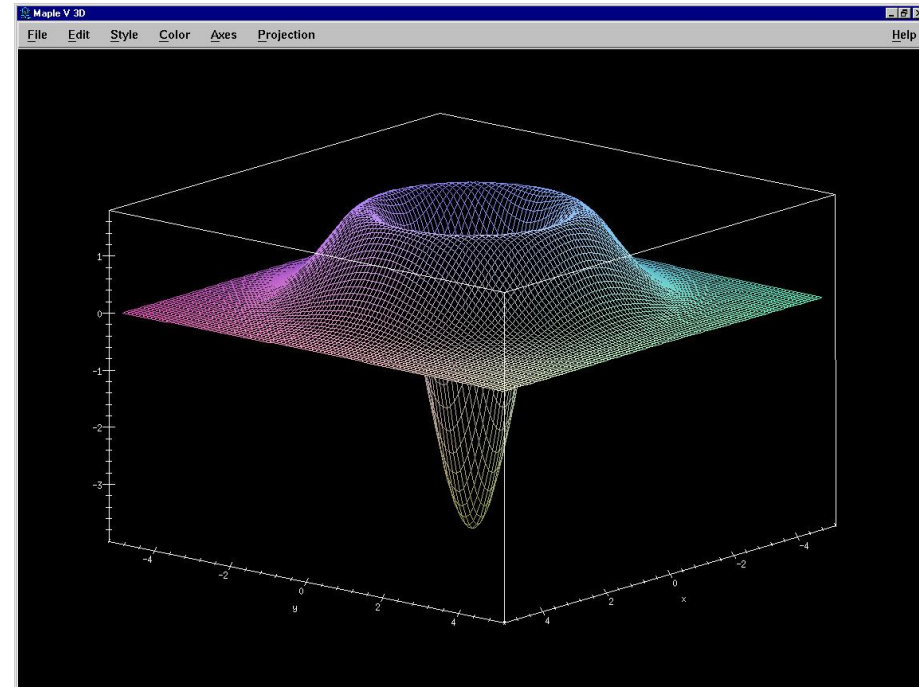
Variante des Laplace-Operators:

$$\nabla^2 \approx \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Stärkere, aber mehr Störkanten

- Der Laplace-Operator ist gegen Rauschen sehr empfindlich.
- Wesentlich bessere Ergebnisse erhält man, wenn man das Bild zunächst mit einem Gauß-Filter glättet und danach den Laplace-Operator anwendet (Laplacian of Gauß, LoG):



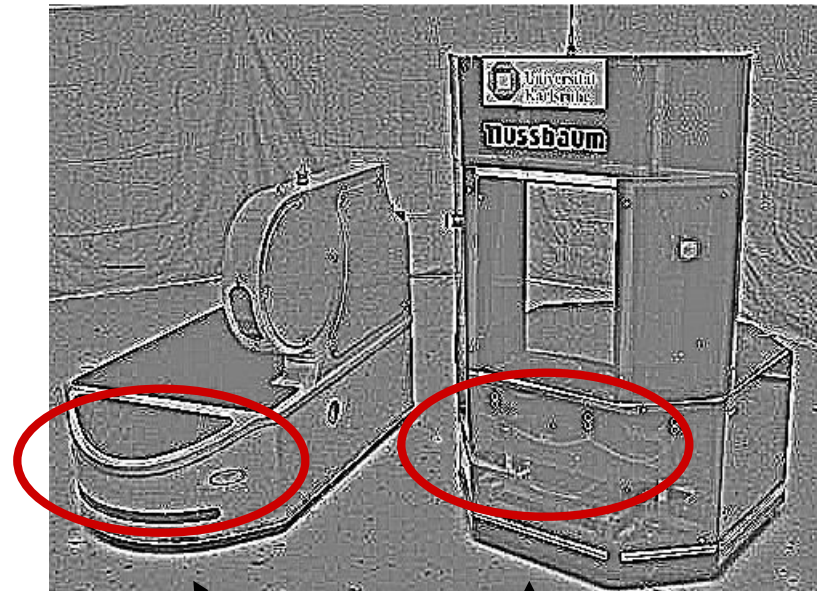
$$LoG(g(x,y)) = \nabla^2(f(x,y) * g(x,y))$$

f bezeichnet die Filterfunktion eines Gauß-Filters

Approximation

Faltung mit Matrix

$$\nabla^2 F(x,y) = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$



Stärkere Kanten,
weniger Rauschen

Canny-Kantendetektor

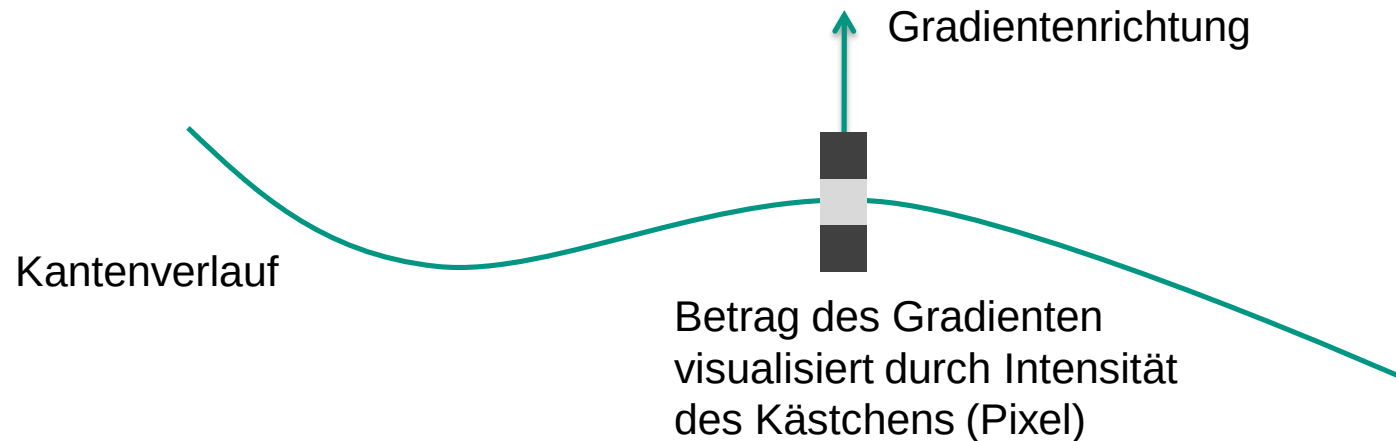
- Nach John F. Canny aus dem Jahre 1986
- Ziel war es den “optimalen” Kantendetektor zu finden:
 - Gute Detektion
 - Gute Lokalisierung
 - Minimale Antwort (“dünne Linien”)
- Canny-Kantendetektor berechnet binäre Antwort (üblicherweise 0: keine Kante, 255: Kante)
- Subpixelgenauigkeit durch Erweiterung möglich
- Algorithmus besteht aus mehreren Schritten

■ Schritte des Algorithmus

1. Rauschunterdrückung: Gauß-Filter
2. Berechnung der Gradienten in horizontaler und in vertikaler Richtung
Prewitt_x / Prewitt_y oder Sobel_x / Sobel_y
 - Berechnung der Richtung: $\varphi = \text{atan}(g_y / g_x)$
 - Einteilung der Richtung in vier Quadranten:
1 : $[-67.5^\circ, -22.5^\circ)$, 2 : $[-22.5^\circ, 22.5^\circ)$,
3 : $[22.5^\circ, 67.5^\circ)$, 4 : $[-90^\circ, -67.5^\circ)$ oder $[67.5^\circ, 90^\circ]$
3. Non-Maximum Supression
4. Hysterese-Schwellwertverfahren

■ Non-Maximum Supression

- Gradient muss lokales Maximum sein bei Betrachtung der zwei direkten Nachbarn entlang der Gradientenrichtung
- Überprüfung erfolgt gemäß dem jeweiligen Quadranten



■ Hysterese-Schwelldwertverfahren

- Verwendung von zwei Schwellwerten: low / high
- Liegt der Betrag des Gradienten für ein Pixel über dem Schwellwert high, so wird es auf jeden Fall akzeptiert
- Ausgehend von den akzeptierten Pixel werden Kanten (rekursiv) verfolgt:
 - Überprüfung der acht direkten Nachbarn
 - Betrag Gradient muss über Schwellwert low liegen
- Anmerkung: Lokale Maximalitätsbedingung aus Schritt 3 muss in jedem Fall erfüllt sein!

Canny-Kantendetektor V

■ Beispiel



Nächste Veranstaltung:

Montag, 9. Mai 2017