



# KR II: Visuelle Verarbeitung II

(Bildverarbeitung, Objekterkennung,  
Interpretation)

Prof. Dr. Verena V. Hafner,  
Humboldt-Universität zu Berlin, Institut für Informatik,  
WS 2008/2009

Ein Großteil dieser Folien wurde von Prof.  
Dr. Hans-Dieter Burkhard und seinen  
Mitarbeitern und Studenten übernommen.

Vielen Dank!

# Bildverarbeitung

Ziele:

- effiziente Speicherung bzw. Verarbeitung
- verbesserte Darstellung  
(Kontrastveränderung, Helligkeitsveränderung,  
Elimination von Störungen, Ausgleich ... )
- Vorbereitung zur Interpretation  
Klassifikation von Bildelementen usw.

# Bildverarbeitung

In der Regel mehrere Verarbeitungsschritte, z.B.

Vorverarbeitung

- z.B. Kalibrierung, Kontrastverstärkung, ...

Komponenten bestimmen

- z.B. Segmentierung: Farbbereiche, Umrisse, ...

Objekte klassifizieren

- z.B. Ball, Spieler

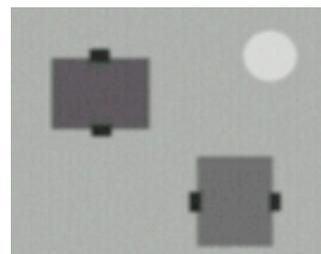
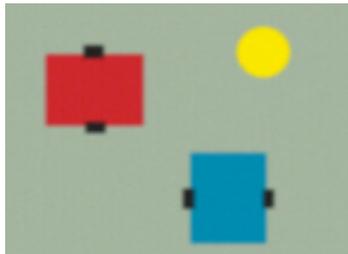
Szenenbeschreibung

- z.B. Dribbeln

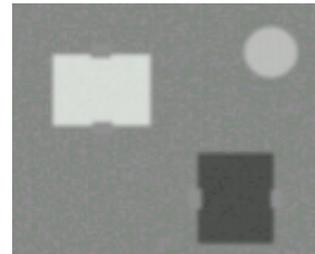
# Objekterkennung

welcher Farbkanal enthält Informationen

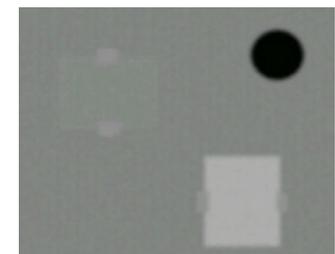
(z.B. Auswertung der Histogramme der Kanäle)



Y (hell)



V (rot)



U (blau)

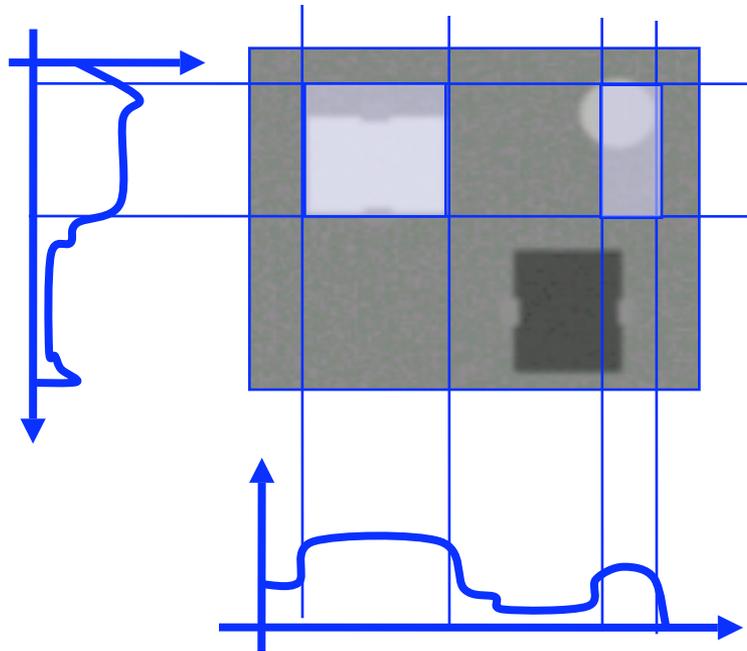
# Objekterkennung

Hypothesen für mögliche Orte des Objekts

Hypothesen für mögliche Größe des Objekts

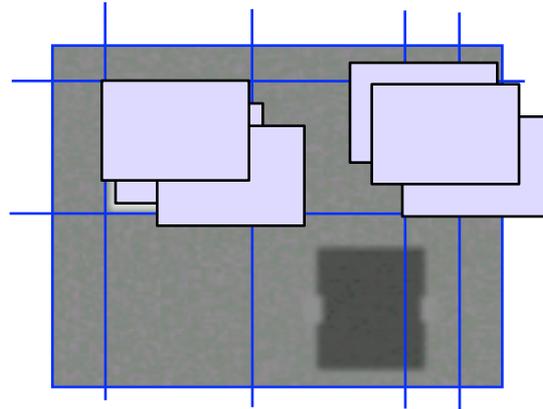
Hypothesen für Farberscheinung des Objekts

(Histogramme pro Zeilen/Spalten auswerten)



# Objekterkennung

Vergleich des gesuchten Objekts  
mit hypothetischen Positionen im Bild



Maß für Ähnlichkeit: Korrelation

Später mehr dazu

# Objekterkennung

Problematik:

Objekte erscheinen unterschiedlich:

- Skalierung
- Lage (Translation)
- Orientierung (Rotation, Blickwinkel)

# Einfluss von Skalierung

Je nach Größe erscheinen Objekte z.B. als ausgedehnte Flächen oder als Linien:  
Skalierung verändert Anwendbarkeit von Identifizierungsverfahren

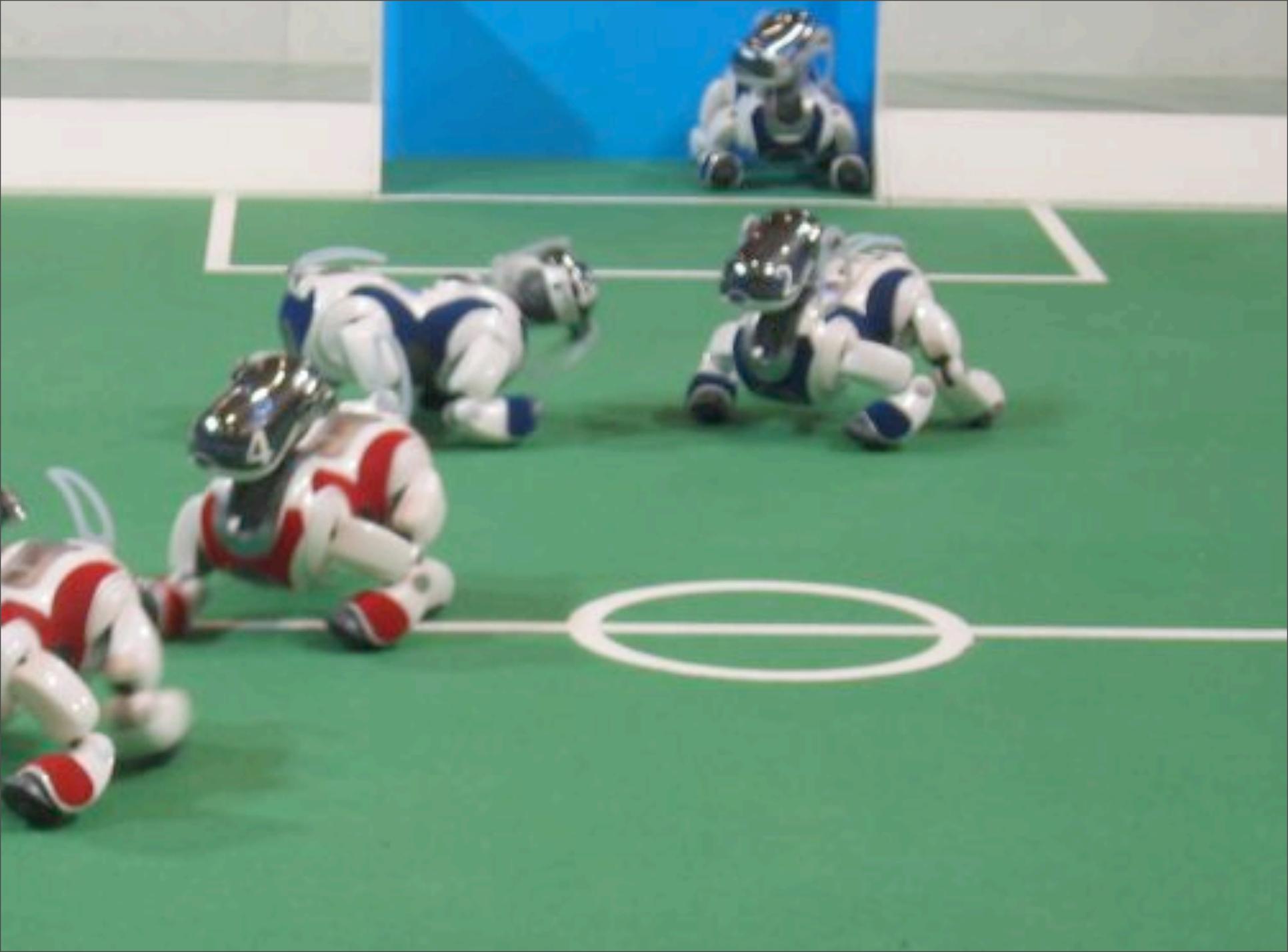


# Einfluss von Skalierung



# Einfluss von Skalierung





# Objekterkennung

Verschiedene Möglichkeiten testen bzgl.

- Skalierung **z.B. Kantenerkennung**
- Lage (Translation)
- Orientierung (Rotation, Blickwinkel)

Geringsten Abstand (Fehler) auswerten

Einschränkung plausibler Möglichkeiten

Diskretisierung der Möglichkeiten

Histogramme für Abschätzungen (z.B. bzgl. Größe)

# Objekterkennung

Andere Variante:

Invariante Merkmale verwenden

z.B.

- Größenverhältnisse
- Winkel
- Bilder unterschiedlicher Skalierung zusammenfassen zu neuem Bild

# Bildverarbeitung: Partielle Verfahren

Zeit-effiziente Verfahren:

Verzicht auf Verarbeitung des gesamten Bildes,  
Partielle Verarbeitung

Verarbeitung von reduzierten Bilder

Untersuchungen längs ausgewählter Scanlinien:

- Histogramme, Farbklassifikation, ...
- Histogramme nur für ausgewählte Farbwerte

Bei Bedarf:

Detailliertere Untersuchung in begrenzten Bereichen

# Bildverarbeitung: Transformationen

Ein Bild  $g = [ g(x,y) ]_{x,y}$

– Matrixdarstellung:

$g(x,y)$  Grauwert oder Farbinformation

wird transformiert in ein Bild  $g' = [ g'(r,s) ]_{r,s}$

– sowohl Angaben zu Pixeln als auch Bildgröße können verändert werden (meist aber Bildgröße gleich)

$$[g'(r,s)]_{r,s} = \text{Trafo}([g(x,y)]_{x,y})$$

Methoden

- Punktweise Verfahren
- Verfahren im Ortsraum
- Verfahren im Ortsfrequenzraum

# Arten von Transformationen

## Geometrische Ortsabhängigkeit

- Punktbezogen
- Lokale Nachbarschaft  $U$
- Global (z.B. Fouriertrafo)

## Inhaltliche Ortsabhängigkeit

- Homogen: unabhängig von Bildkoordinate
- Inhomogen: abhängig von Bildkoordinate
- Linearität: linear/nicht-linear

## Beispiele:

punktbezogen, homogen: Negativbild

$$g'(x, y) = 1 - g(x, y)$$

lokal, inhomogen: Kompensation von Kamera-Bewegung

# Bildverarbeitung: Punktweise Verfahren

Transformation erfolgt isoliert für jedes einzelne Pixel.

$$[ g'(x,y) ]_{x,y} = \text{Trafo}( [ g(x,y) ]_{x,y} )$$

realisiert durch punktweise Umformung

$$g'(x,y) = f( g(x,y) ) \text{ für alle Punkte } (x,y)$$

# Bildverarbeitung: Punktweise Verfahren

Lineare Skalierungsverfahren ( **Trafo** linear )

Helligkeit verändern: Additiv  $g'(x,y) := g(x,y) + c$

(ggf. Abschneiden an Grenzen)

Kontrast verändern: Multiplikativ  $g'(x,y) := c \cdot g(x,y)$

(ggf. Abschneiden an Grenzen)

Vorgegebene Mittelwerte/Varianzen erzielen

(z.B. eine mittlere Helligkeit)

Beispiel:

Maximaler Kontrast unter Ausnutzung des verfügbaren  
Bereiches (Analyse mittels Histogrammen)

# Helligkeitskorrektur

Ausgleich für

- inhomogene Beleuchtung
- Ungleichmäßige Kameraempfindlichkeit  
(Kalibrierung bzgl. Abbildungssystem, Sensoren)

$$g'(x,y) := g(x,y) : b(x,y)$$

$b(x,y)$  ist ein Referenzbild für die ungleichmäßige Helligkeit  
z.B.

- Helligkeit des Hintergrunds (ggf. Interpolation)
- Kamerareferenz-Bild

# Äquidensiten



Stellen gleicher Dichte im Bild

# Bildverarbeitung: Punktweise Verfahren

Stückweise konstante **Trafo**

Äquidensiten (vergrößerte Quantisierung), Binärbilder

Klassifikation/Segmentierung für gegebene Zerlegung  
(Klasseneinteilung)  $K_1, \dots, K_n$  des Farb-/Grauwertbereichs:

$$g'(x,y) := K_i \text{ für } g(x,y) \in K_i$$

bzw.

$$g'(x,y) := w_i \text{ für } g(x,y) \in K_i \text{ und „typisches“ } w_i \in K_i$$

## Transformation von Farbräumen

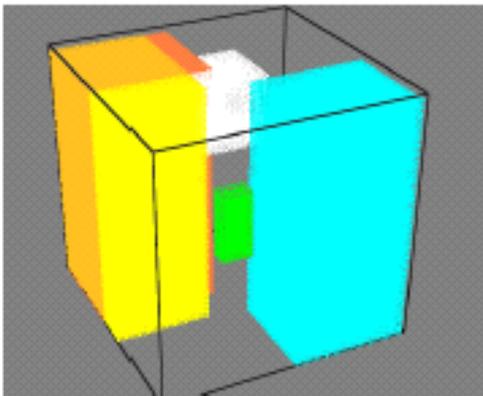
# Bildverarbeitung: Punktweise Verfahren

Beispiel:

Segmentierung (c) eines Bildes (b) für gegebene Zerlegung (a) des Farbraums

$g'(x,y) := w_i$  für  $g(x,y) \in K_i$  und „typisches“  $w_i \in K_i$

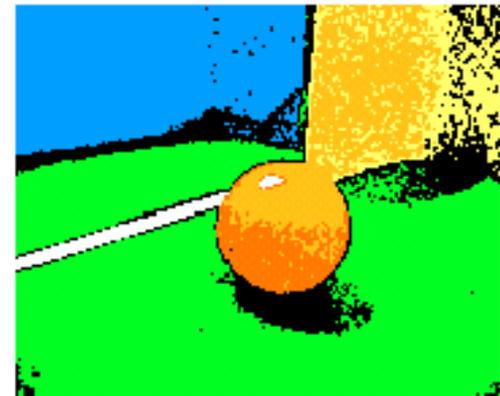
a)



b)



c)



Zerlegung z.B. gemäß Farbtabelle.

Klassen als Intervalle oder komplexere Mengen.

# Bildverarbeitung: Punktweise Verfahren

Äquivalenzen:

stückweise konstante Skalierung von Grauwertbildern

Bestimmung

geeigneter Intervalle  $K_i$  und  $w_i \in K_i$

ausgehend z.B.

von Mittelwert, Varianz, Histogramm, ...



Haberäcker S.117

# Bildverarbeitung: Punktweise Verfahren

Äquidensiten (insbesondere Binärbilder)

als Vorstufe für weitere Verarbeitung, z.B. Kantenbildung:

Äquidensiten 2. Ordnung (Kantenbild der Äquidensiten)



Haberäcker S.117

# Bildverarbeitung: Punktweise Verfahren

## Verknüpfung von Bildern

### Addition

- z.B. Überlagerung von Luftaufnahmen mit topographischer Information, politischen Grenzen usw.

### Subtraktion

- Zeitliche Differenzbilder: Erkennung von Bewegung
- Differenz verschobener Bilder: Erkennung gerichteter Kanten (parallel zur Verschiebung)

# Differenz verschobener Bilder:



Watkins S. 52ff  
Haberäcker S.62



# Bildverarbeitung: Ortsbereich

Verfahren im Ortsbereich:

Transformationen erfolgen für jedes einzelne Pixel in  
Abhängigkeit von seiner Nachbarschaft:

„**Diskrete Nachbarschaftsoperatoren**“

$[ g'(x,y) ]_{x,y} = \text{Trafo} ( [ g(x,y) ]_{x,y} )$  als

$g'(x,y) = f ( [ g(u,v) ]_{(u,v) \in U(x,y)} )$  für alle Punkte  $(x,y)$

wobei

- $U(x,y)$  = Umgebung des Punktes  $(x,y)$ , z.B. alle Nachbarn
- $[ g(u,v) ]_{(u,v) \in U(x,y)}$  Grauwerte der Umgebung
- $f$  = beliebige Funktion über Grauwerten der Umgebungen

# Bildverarbeitung: Ortsbereich

Umgebung  $U(x,y)$  z.B. als ein Fenster  
von quadratischer Form

$(x-1,y-1)$	$(x,y-1)$	$(x+1,y-1)$
$(x-1,y)$	$(x,y)$	$(x+1,y)$
$(x-1,y+1)$	$(x,y+1)$	$(x+1,y+1)$

$[ g(u,v) ]_{(u,v) \in U(x,y)}$

$g(x-1,y-1)$	$g(x,y-1)$	$g(x+1,y-1)$
$g(x-1,y)$	$g(x,y)$	$g(x+1,y)$
$g(x-1,y+1)$	$g(x,y+1)$	$g(x+1,y+1)$

# Rangordnungsoperatoren

Rangordnung:

Grauwerte der Umgebung anordnen

Rangordnungsoperator: Ein Wert wird  
entsprechend seiner Stellung in der  
Rangordnung zugewiesen

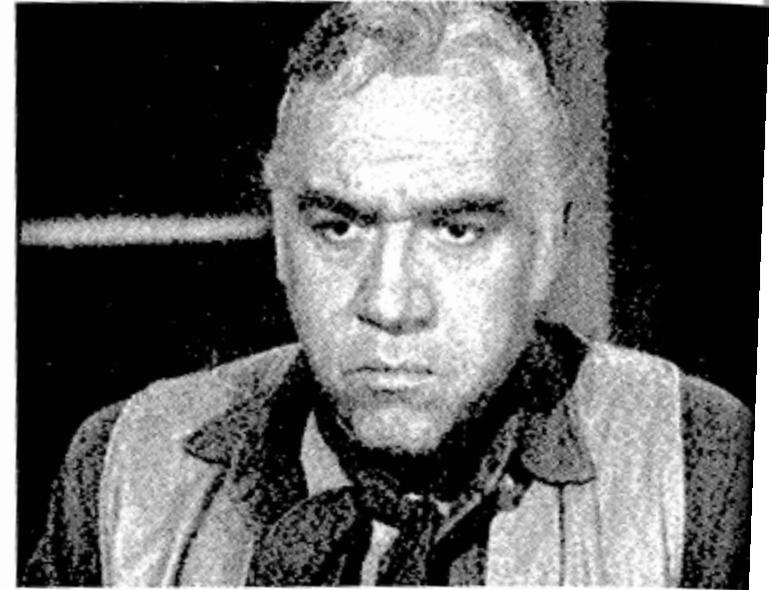
(z.B. Maximum)

i.a. nicht als Faltung (s.u.) definierbar

# Rangordnungsoperatoren

**Median-Operator:**  $g'(x,y) := \text{Median}_{(x,y) \in U} g(x,y)$

Anwendung: Elimination isolierter Störungen



Nischwitz/Haberäcker S. 458

# Rangordnungsoperatoren

## Morphologische Operatoren (Speziell für Binärbilder)

- **Dilatation:**  $g'(x,y) := \text{Max}_{(x,y) \in U} g(x,y)$

$G'$  entsteht aus  $G$  durch Ausdehnung weißer Flächen in jede Richtung um ein Pixel.

- **Erosion:**  $g'(x,y) := \text{Min}_{(x,y) \in U} g(x,y)$

$G'$  entsteht aus  $G$  durch Ausdehnung schwarzer Flächen (Verminderung weißer Flächen) um ein Pixel.

# Rangordnungsoperatoren

Anwendungen:

- Kanten-Erkennung

Symmetrische Differenz  $\text{Dilation}(G) \text{ XOR } G$

ergibt Kanten an den Übergängen Schwarz/Weiß.



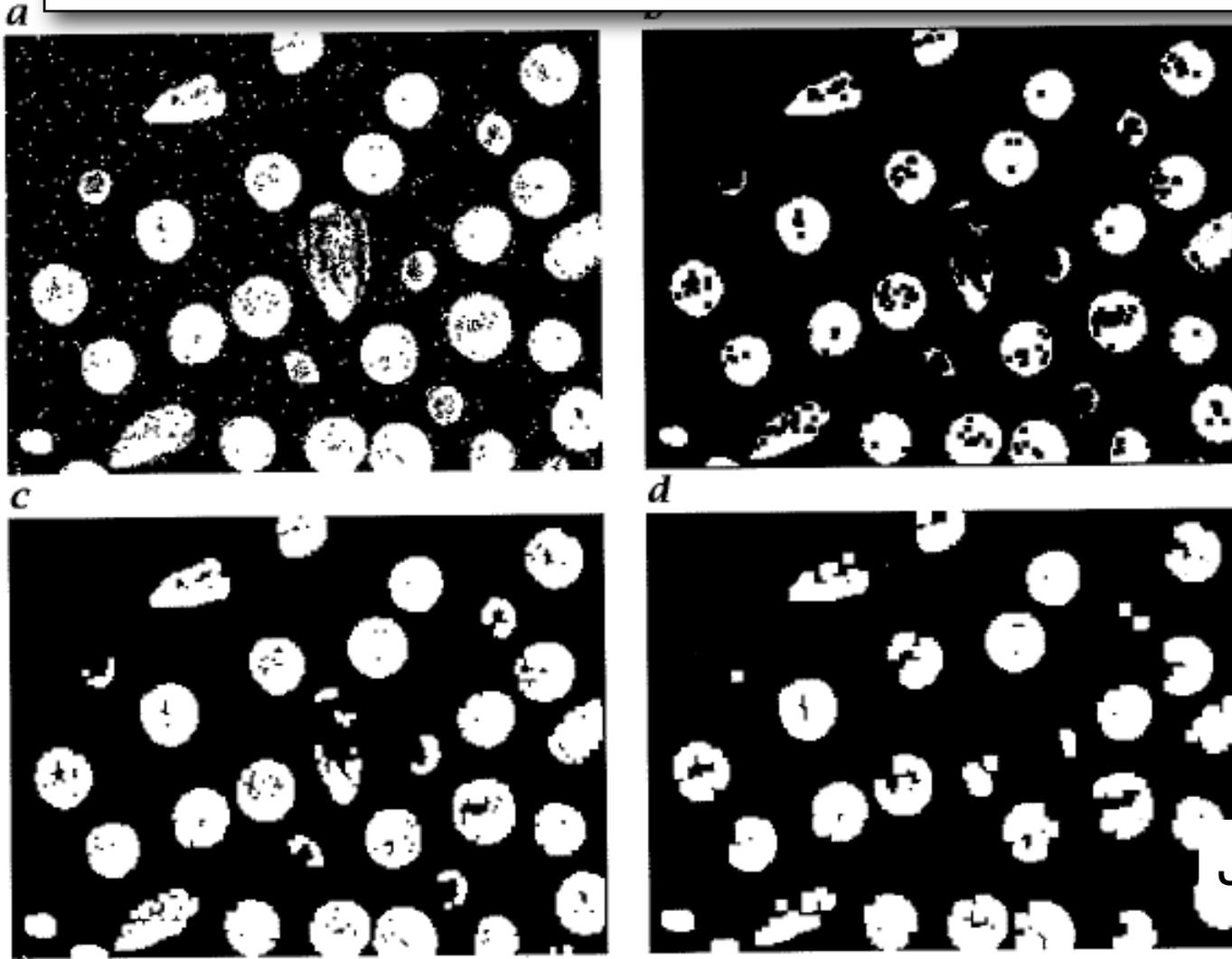
# Rangordnungsoperatoren

Anwendungen:

- Erosion und anschließende Dilation („Opening“):
  - Entfernen aller (weißen) Objekte, die das Fenster nicht ausfüllen
  - Glätten von Kanten
- Dilatation und anschließende Erosion („Closing“)
  - Entfernen aller (schwarzen) Löcher, die das Fenster nicht ausfüllen
  - Glätten von Kanten

Durch unterschiedliche Fenster können unterschiedliche Formen eliminiert werden.

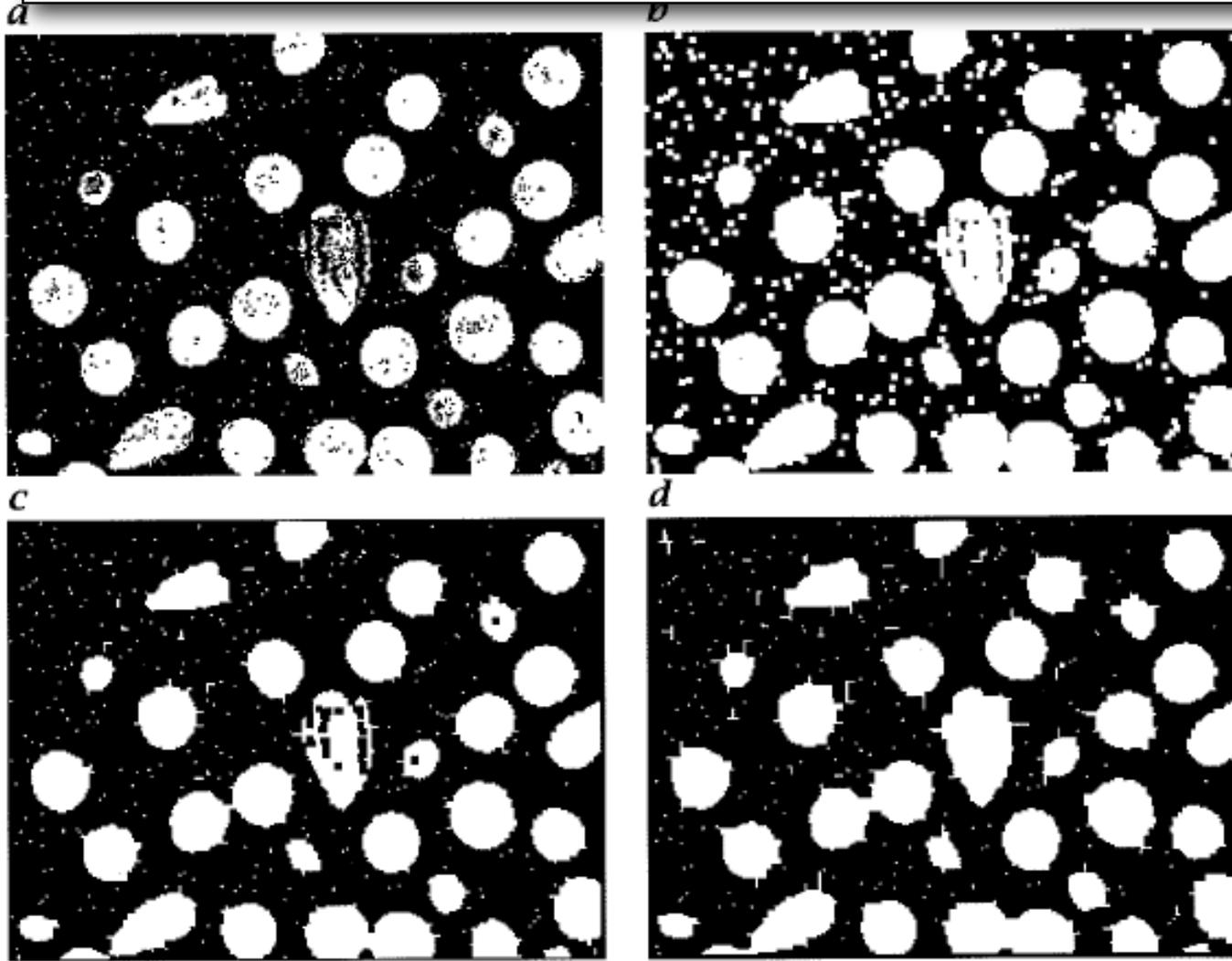
# Rangordnungsoperatoren



JÄHNE S. 513f

**Abbildung 18.2:** Erosion und Opening: **a** Originalbinärbild; **b** Erosion mit einer  $3 \times 3$ -Maske; **c** Opening mit einer  $3 \times 3$ -Maske; **d** Opening mit einer größeren Maske.

# Rangordnungsoperatoren



*Abbildung 18.3: Dilatation und Closing: a Originalbinärbild; b Dilatation mit einer  $3 \times 3$ -Maske; c Closing mit einer  $3 \times 3$ -Maske; d Closing mit einer  $5 \times 5$ -Maske.*

JÄHNE S. 513f

# Bildverarbeitung: Ortsbereich

Diskrete **ortsinvariante** Filteroperationen

Umgebungen  $U(x,y)$  („Fenster“) immer

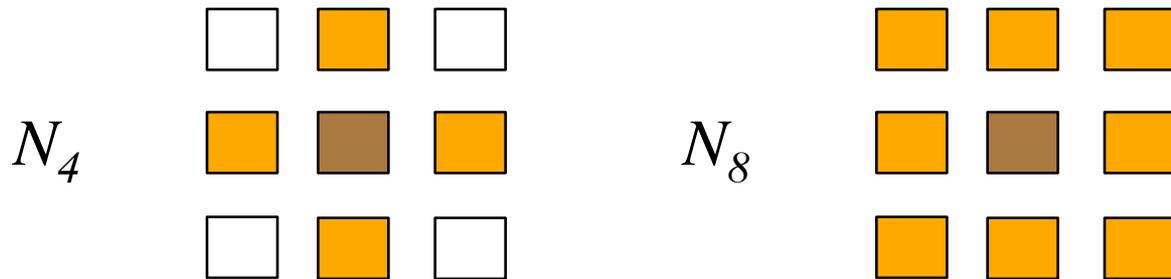
in **gleicher Form** z.B. quadratische Umgebung  $3 \times 3$

$(x-1,y-1)$	$(x,y-1)$	$(x+1,y-1)$
$(x-1,y)$	$(x,y)$	$(x+1,y)$
$(x-1,y+1)$	$(x,y+1)$	$(x+1,y+1)$

**f** unabhängig vom Ort  $(x,y)$

Am Bildrand spezielle Konventionen (z.B. Spiegelung, konstanter Wert, periodische Fortsetzung ...).

# Umgebungen (Nachbarschaften)



4-Nachbarschaft eines Punktes  $p=(r,c)$

$$\begin{aligned} N_4 &= \{(r, c - 1), (r - 1, c), (r, c + 1), (r + 1, c)\} \\ &= \{(u, v) \mid |u - r| + |v - c| = 1\} \end{aligned}$$

8-Nachbarschaft eines Punktes

$$N_8 = \{(u, v) \mid \max[|u - r|, |v - c|] = 1\}$$

# Faltungsoperationen

Beschreibung der Transformation durch

Matrix  $[ h(u,v) ]$  („Kern“, „Maske“)

- $h(u,v)$  als Wichtung für Einfluss von  $g(x+u,y+v)$  auf  $g'(x,y)$
- $h(u,v) = 0$  für Komponenten ohne Einfluss

Bezeichnung  $(u,v)$  gemäß Abständen zu  $(x,y)$  in  $U$

$h(-1,-1)$	$h(0,-1)$	$h(+1,-1)$
$h(-1,0)$	$h(0,0)$	$h(+1,0)$
$h(-1,+1)$	$h(0,+1)$	$h(+1,+1)$

# Faltungsoperationen

Beispiel für  $[ h(u,v) ]$  (Sobel-Filter)

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

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

Neuer Wert statt g:  
 $g' = -a+c-2f+2h-k+m$

# Anwendung des Filters für alle Pixel

Für  
Randpunkte  
geeignet  
ergänzen

-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	t
	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	t
	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	t
	u	v	w	x	y

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

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

$$l' = (h+2m+r)-(f+2k+p)$$

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

$$m' = (i+2n+s)-(g+2l+q)$$

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

$$n' = (j+2o+t)-(h+2m+r)$$

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

$$q' = (m+2r+w)-(k+2p+u)$$

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

$$r' = (n+2s+x)-(l+2q+v)$$

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

$$s' = (o+2t+y)-(m+2r+w)$$

# Randpixelbehandlung

Unterschiedliche Varianten:

- Unverändert
- Auf „konstant“ setzen
- Faltungskern anpassen
- Zusätzliche Nachbarn annehmen
  - Reflektierte Indizierung (Nachbarn gespiegelt)
  - Zyklische Indizierung (Fortsetzung anderer Rand)

# Faltungsoperationen

Transformation ist im Prinzip (Vorzeichen) eine **Diskrete Faltung** auf beschränktem Bereich

$$\begin{aligned}g'(x,y) &= g * h(x,y) \\ &= \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x+u,y+v) \cdot h(u,v)\end{aligned}$$

Mit Normierung, z.B.:

$$\begin{aligned}g'(x,y) &= g(x,y) * h(u,v) \\ &= (1/|U|) \cdot \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x+u,y+v) \cdot h(u,v)\end{aligned}$$

# Faltungsoperationen: Anwendungen

Glättung mit 5x5 - Maske:

$$g'(x,y) = (1/25) \cdot \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x-u,y-v)$$

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

Glättungen mit anderen Masken z.B.

0	1	0
1	1	1
0	1	0

0	1	0
1	4	1
0	1	0

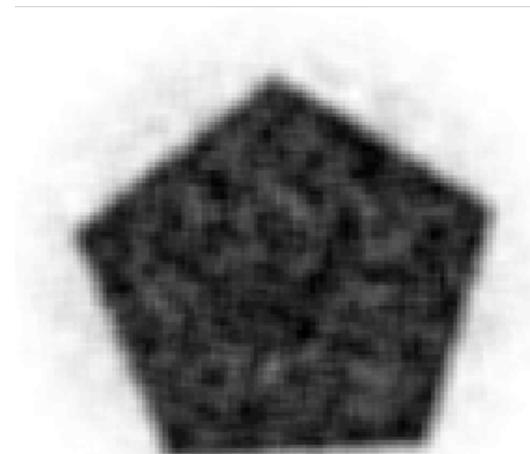
Kontrastverschärfung z.B.

-1	-1	-1
-1	1	-1
-1	-1	-1

0	-1	0
-1	4	-1
0	-1	0

# Gewichteter Mittelwertfilter, Gauß-Filter

Mittelwertfilter, z.B.

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$


Tiefpassfilter: Die Bilder werden weicher, „Glättung“  
Störungen (hohe Frequenz) werden eliminiert.

Unterschied zu Medianfilter:

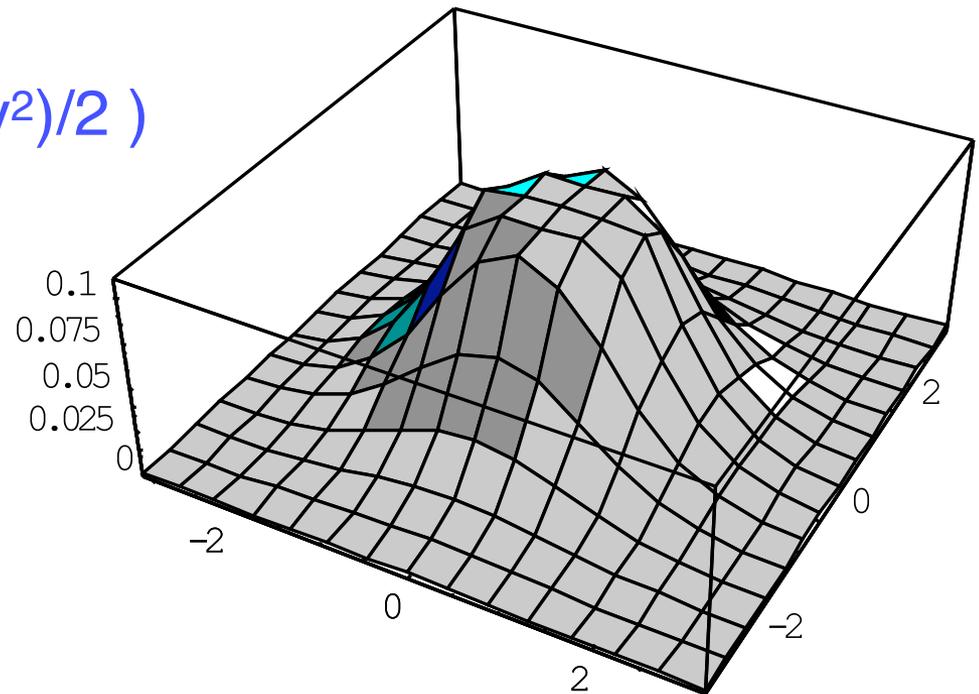
- Median unterdrückt extreme Werte,
- Mittelwert bezieht sie mit ein.

# Gewichteter Mittelwertfilter, Gauß-Filter

## Gauß-Filter

= Gewichteter Mittelwertfilter zur Approximation der Gaußfunktion

$$h(x,y) = 1/2\pi \exp( - (x^2+y^2)/2 )$$



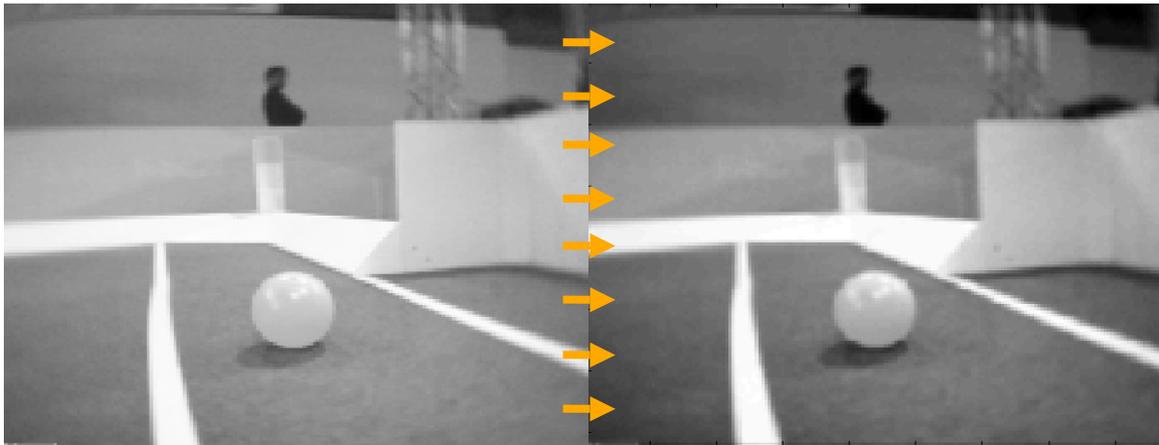
Tiefpassfilter: Die Bilder werden weicher, „Glättung“

# Gewichteter Mittelwertfilter, Gauß-Filter

$$h(x,y) = 1/2\pi \exp( - (x^2+y^2)/2 )$$

Gaußfilter mit  
Maskengröße 3x3

0.01	0.08	0.01
0.08	0.61	0.08
0.01	0.08	0.01



# Differenzenoperatoren

Richtungsabhängige Differenzenbildung:

Intensitäts-Differenzen (Kanten) in einer Richtung erkennen

$$\begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline -1 \\ \hline 1 \\ \hline \end{array}$$

Analogie zu Differentialquotient (in x-Richtung):

$$(g(x+\Delta x)-g(x)) / \Delta x = (g(x+1)-g(x)) / (x+1-x) = g(x+1)-g(x)$$

Entspricht Differenz verschobener Bilder.

# Differenzenoperatoren

Problem mit Rauschen

- Vorherige Glättung
- Erweiterte Operatoren (andere Richtung analog) z.B.

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

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

## Sobel-Operator in x- bzw. y-Richtung

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

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

# Differenzenoperatoren

„Kompass-Gradient“ mit 8 Richtungen, z.B.

Ost

-1	1	1
-1	-2	1
-1	1	1

Nord-  
Ost

1	1	1
-1	-2	1
-1	-1	1

USW.

# Sobel-Operator

x-Richtung



y-Richtung



(Haberäcker S.136)

Gradienten-Bild:  
Kombination  
der beiden  
oberen



Kanten: Mit  
Schwellwert aus  
Gradientenbild  
erzeugtes  
Binärbild

# Diskreter Laplace-Operator

Richtungsunabhängige Differenzenbildung:

- Stärke von Änderungen

in Analogie zum kontinuierlichen Laplace-Operator

$$d^2g(x,y)/dx^2 + d^2g(x,y)/dy^2$$

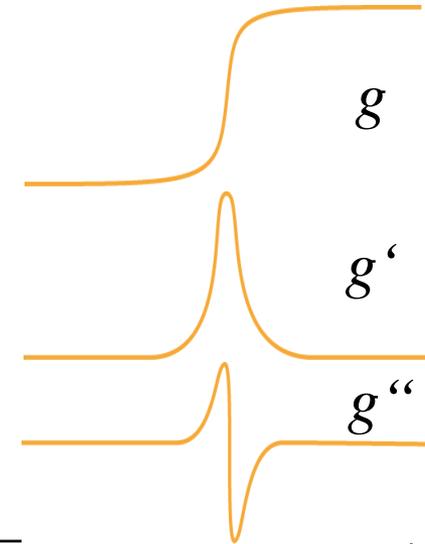
- auch: Wendepunkte des Originals als Nullstellen,  
Dabei 3. Ableitung nicht Null,  
- auf Ebenen verschwindet Laplace-Operator.

# Diskreter Laplace-Operator

– Wendepunkt

Nullstelle der 2. Ableitung

$$\nabla^2 g(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2} = 0$$



Diskrete Laplace-Operatoren z.B.

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1

# Diskreter Laplace-Operator

Approximation durch diskrete 3x3-Maske z.B.

0	1	0
1	-4	1
0	1	0

Weitere  
Varianten:

-1	-1	-1
-1	8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1



Watkins S. 52ff

# Kombination Glättung/Kantenextraktion

Problem:

- Kanten und Störungen sind beide hochfrequent
- Differenzbildung reagiert auf Störungen

Rauschen variiert sogar stärker als Kantenübergang

Lösung:

Erst Rauschunterdrückung (Tiefpass-Filter)

dann Kantenextraktion (Hochpass-Filter)

# Kombination Glättung/Kantenextraktion

Kombination in einem Operator:

LoG = Laplacian-of-Gaussian

Wegen Assoziativität:

$$\begin{aligned} & ( g(x,y) * h_{\text{Gauss}}(u,v) ) * h_{\text{Laplace}}(s,t) \\ = & g(x,y) * ( h_{\text{Gauss}}(u,v) * h_{\text{Laplace}}(s,t) ) \end{aligned}$$

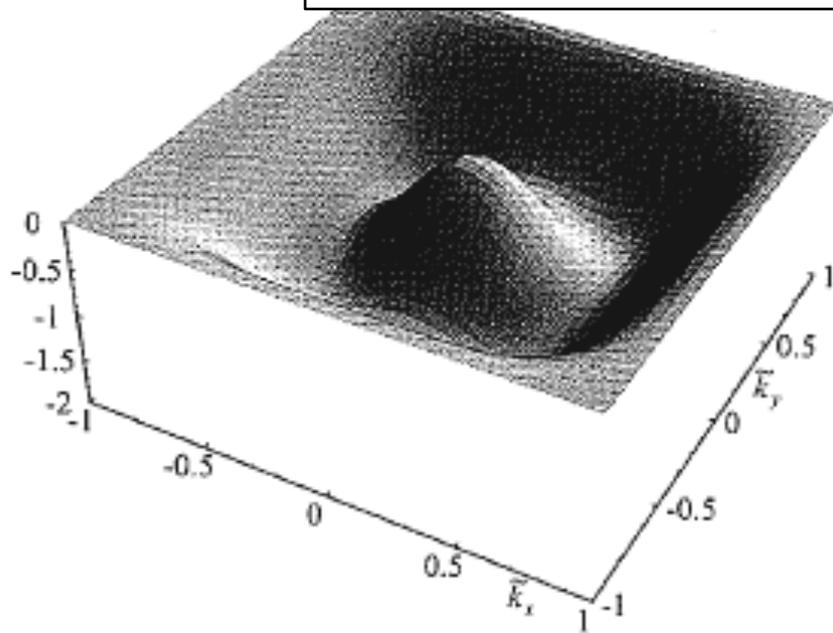
Auch als Differenz von Gauss-Operatoren approximierbar:

DoG = Difference of Gaussian

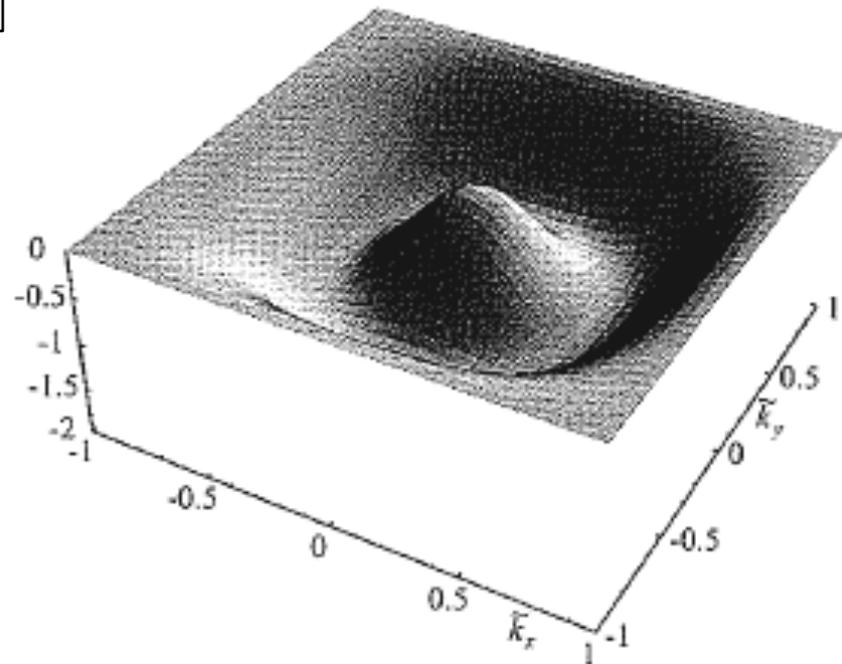
# Kombination Glättung/Kantenextraktion

„Mexican hat“  
Jähne S. 354

*a*



*b*

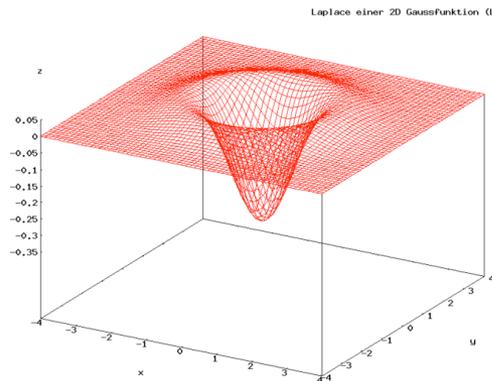


**Abbildung 12.13:** Pseudo-3D-Plot der Transferfunktionen *a* des LoG-Filters  $\mathcal{L}\mathcal{B}^2$  und *b* des DoG-Filters  $4(\mathcal{B}^4 - \mathcal{B}^2)$ .

# Laplacian-of-Gaussian (LoG)

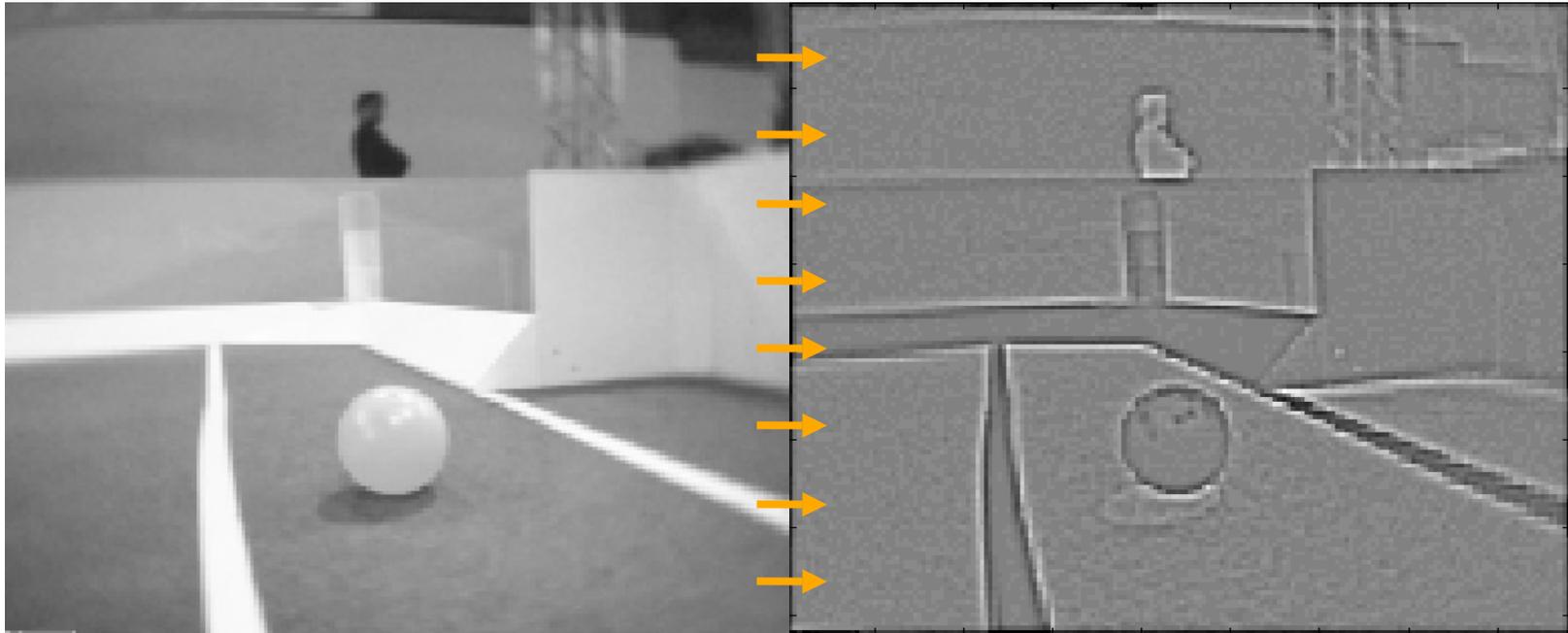
Gauß-Kernel:  $K_G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ , Laplace-Kernel:  $K_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Kombinierter Filterkern  $K$ :



$$K = K_G * K_L = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & -2 & 0 & 1 \\ 2 & -2 & -8 & -2 & 2 \\ 1 & 0 & -2 & 0 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

# *Laplacian-of-Gaussian (LoG)*



(Maskengröße 5x5)

## Filtering and Edge Detection

---

- Gaussian Smoothing
- *Removes high-frequency noise*
- *Convolution of intensity image  $I$  with  $G$ :*  $\hat{I} = G \otimes I$

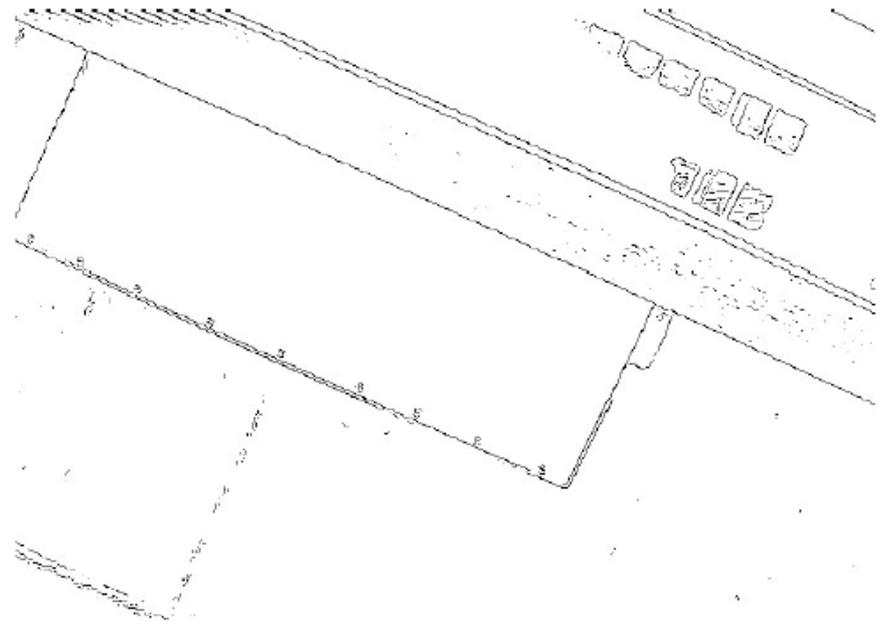
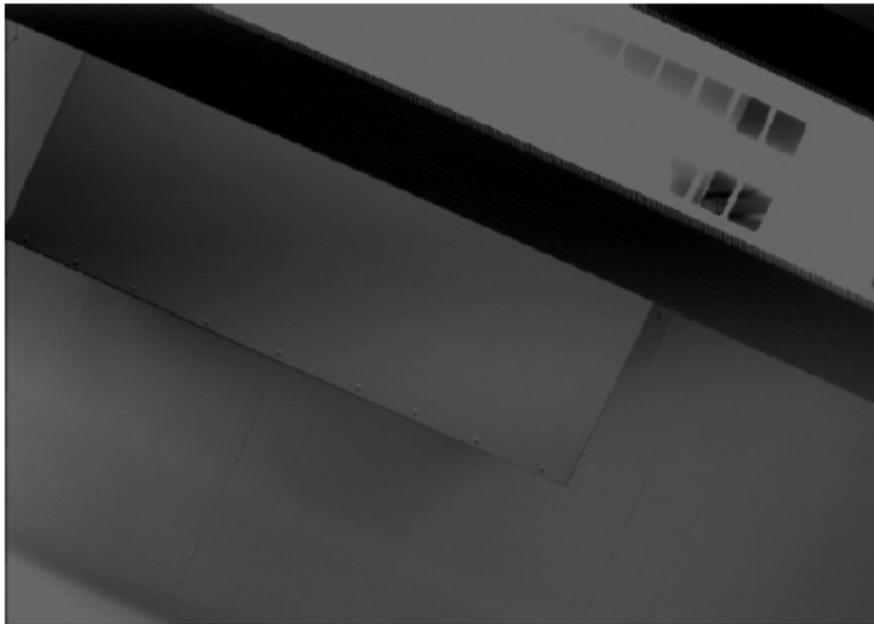
- *with:* 
$$G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Edges
- *Locations where the brightness undergoes a sharp change,*
- *Differentiate one or two times the image*
- *Look for places where the magnitude of the derivative is large.*
- *Noise, thus first filtering/smoothing required before edge detection*

# Edge Detection

---

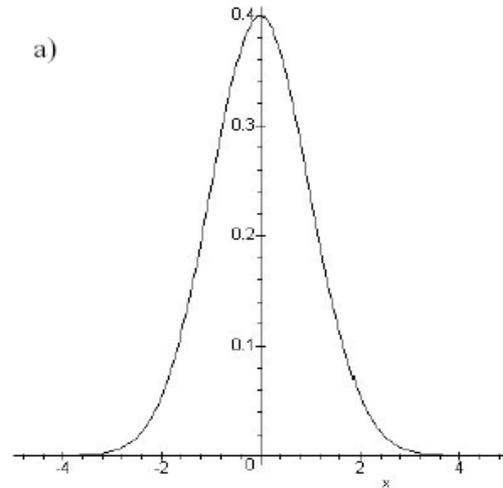
- Ultimate goal of edge detection
  - *an idealized line drawing.*
- Edge contours in the image correspond to important scene contours.



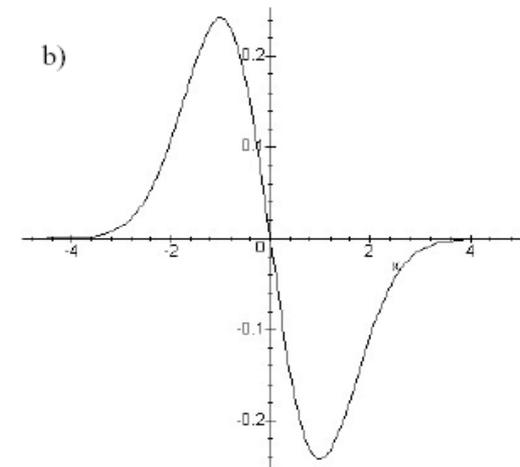
# Optimal Edge Detection: Canny

- The processing steps
  - *Convolution of image with the Gaussian function  $G$*
  - *Finding maxima in the derivative*
- Canny combines both in one operation

$$(G \otimes I)' = G' \otimes I$$



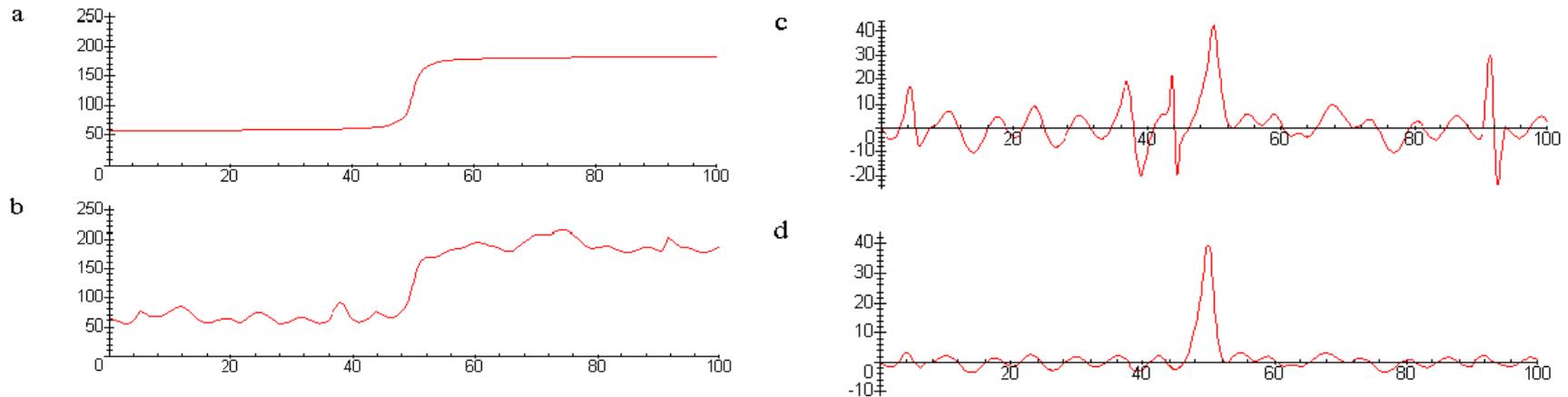
$$G_{\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$$



$$G'_{\sigma}(x) = \frac{-x}{\sqrt{2\pi\sigma^3}} e^{-\frac{x^2}{2\sigma^2}}$$

(a) A Gaussian function. (b) The first derivative of a Gaussian function.

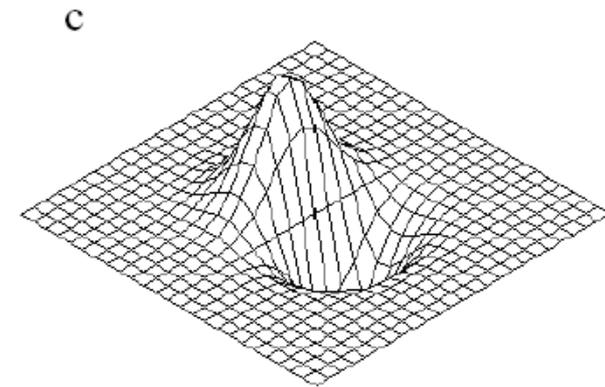
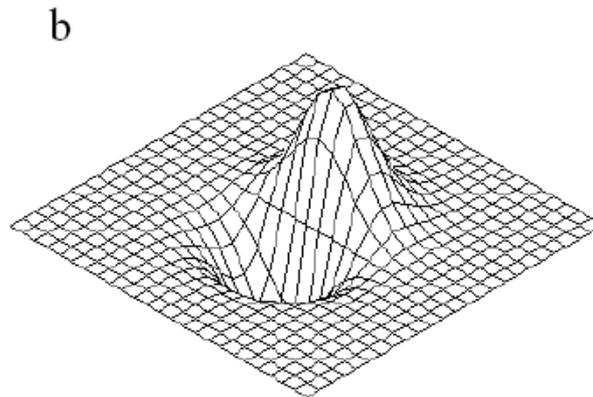
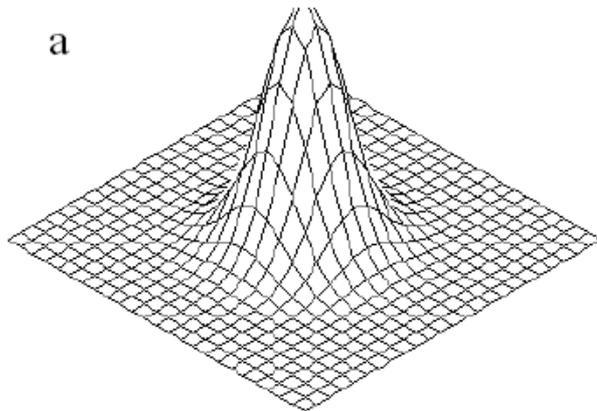
## Optimal Edge Detection: Canny 1D example



- (a) Intensity 1-D profile of an ideal step edge.
- (b) Intensity profile  $I(x)$  of a real edge.
- (c) Its derivative  $I'(x)$ .
- (d) The result of the convolution  $R(x) = G' \otimes I$ , where  $G'$  is the first derivative of a Gaussian function.

## Optimal Edge Detection: Canny

1. 1-D edge detector can be defined with the following steps:
  1. *Convolute the image I with  $G'$  to obtain R.*
  2. *Find the absolute value of R.*
  3. *Mark those peaks  $|R|$  that are above some predefined threshold T. The threshold is chosen to eliminate spurious peaks due to noise.*
2. 2D: Two dimensional Gaussian function



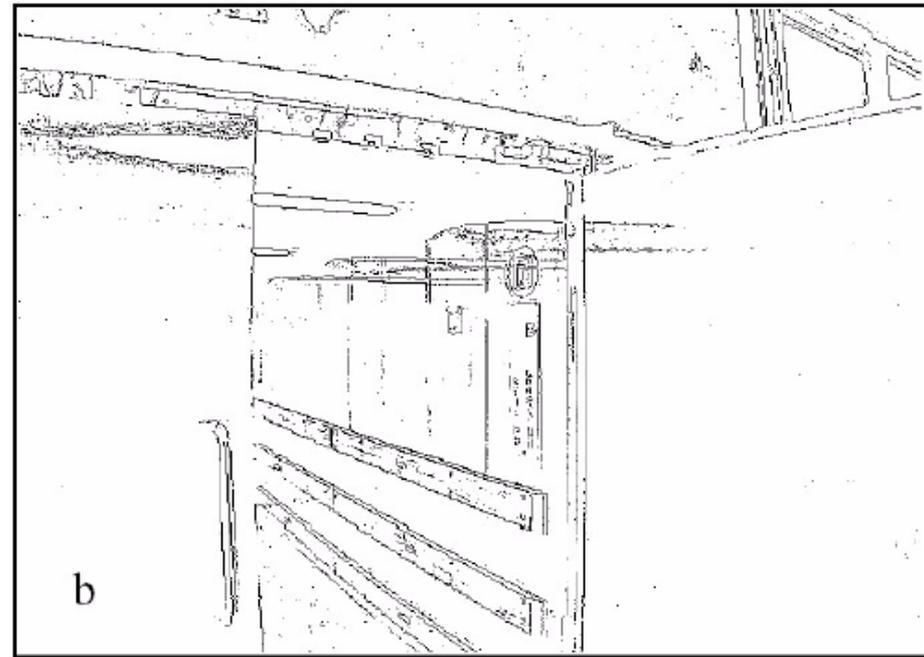
$$G_{\sigma}(x, y) = G_{\sigma}(x)G_{\sigma}(y)$$

$$f_V(x, y) = G'_{\sigma}(x)G_{\sigma}(y)$$

$$f_H(x, y) = G'_{\sigma}(y)G_{\sigma}(x)$$

## Optimal Edge Detection: **Canny** Example

- a) Example of Canny edge detection
- b) After non-maxima suppression



# Gradient Edge Detectors

---

## • Roberts

$$|G| \cong \sqrt{r_1^2 + r_2^2} ; \quad r_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} ; \quad r_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

## • Prewitt

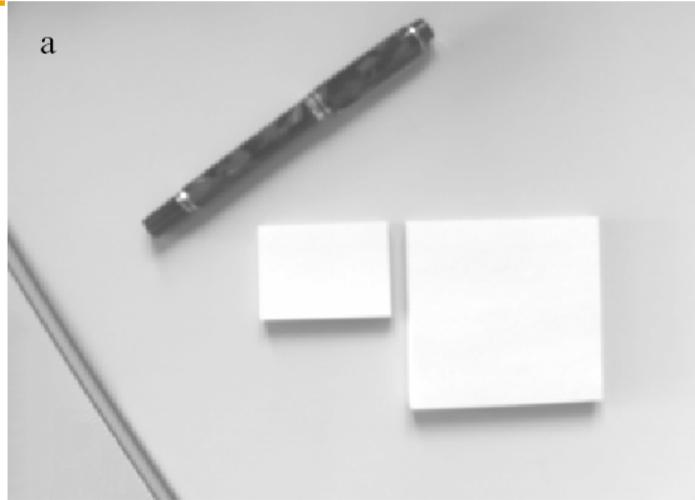
$$|G| \cong \sqrt{p_1^2 + p_2^2} ; \quad \theta \cong \text{atan}\left(\frac{p_1}{p_2}\right) ; \quad p_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} ; \quad p_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

## • Sobel

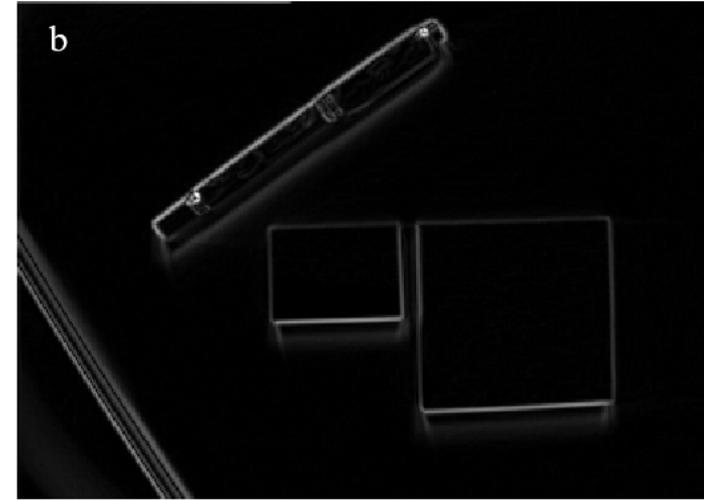
$$|G| \cong \sqrt{s_1^2 + s_2^2} ; \quad \theta \cong \text{atan}\left(\frac{s_1}{s_2}\right) ; \quad s_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} ; \quad s_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

# Example

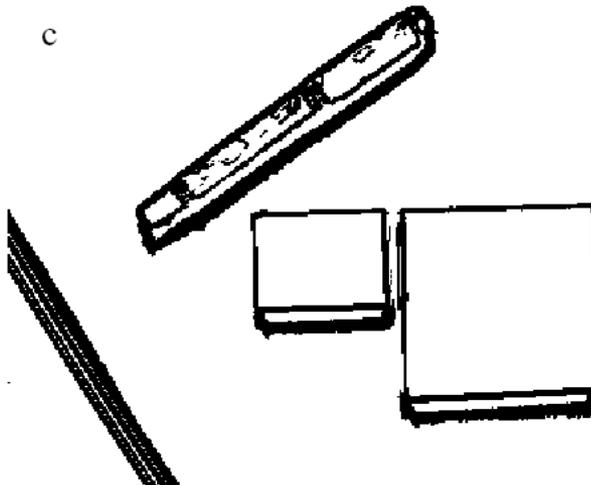
a) Raw image



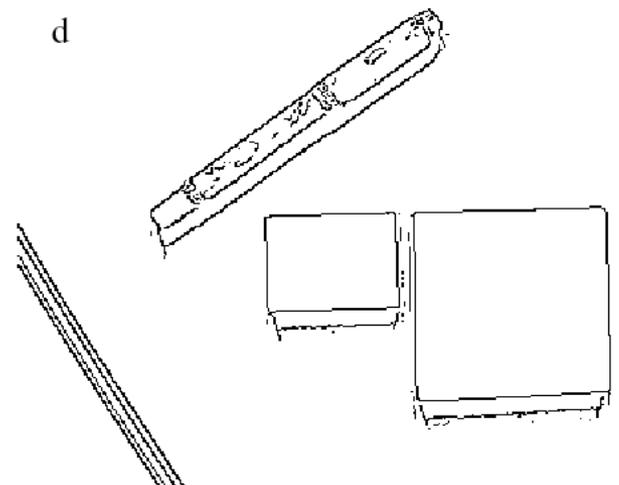
b) Filtered  
(Sobel)



c) Thresholding

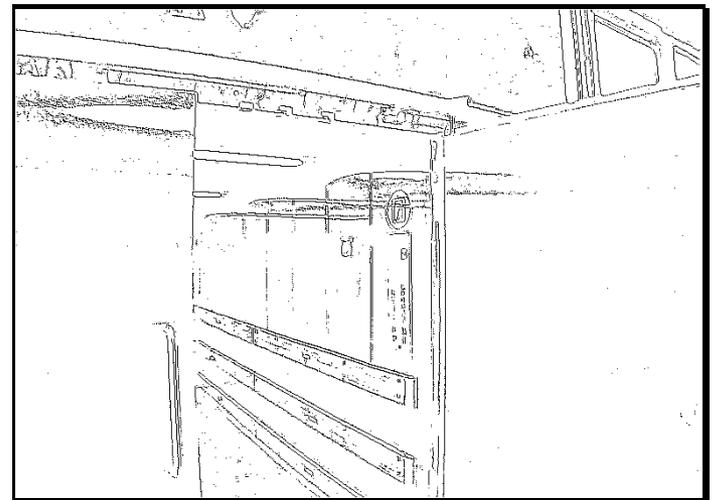


d) Nonmaxima  
suppression

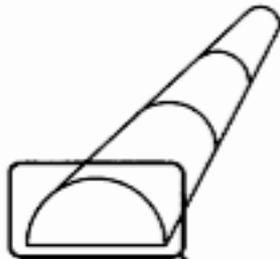


# Nonmaxima Suppression

- 1 Output of an edge detector is usually a b/w image where the pixels with gradient magnitude above a predefined threshold are black and all the others are white
- 1 Nonmaxima suppression generates contours described with only one pixel thinness

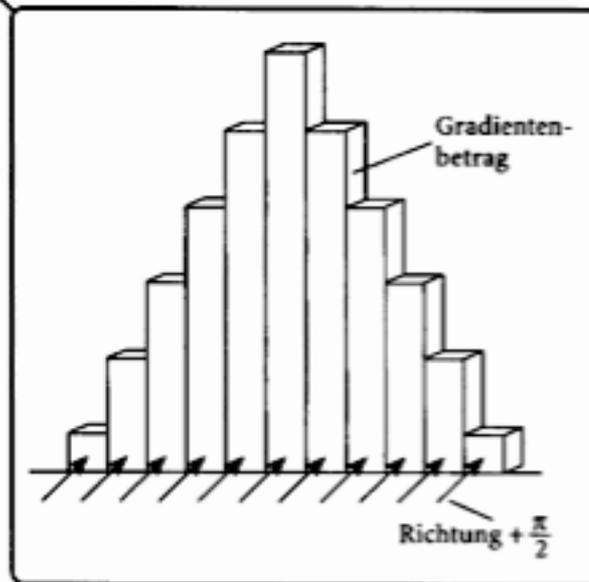


# Non-Maxima Suppression

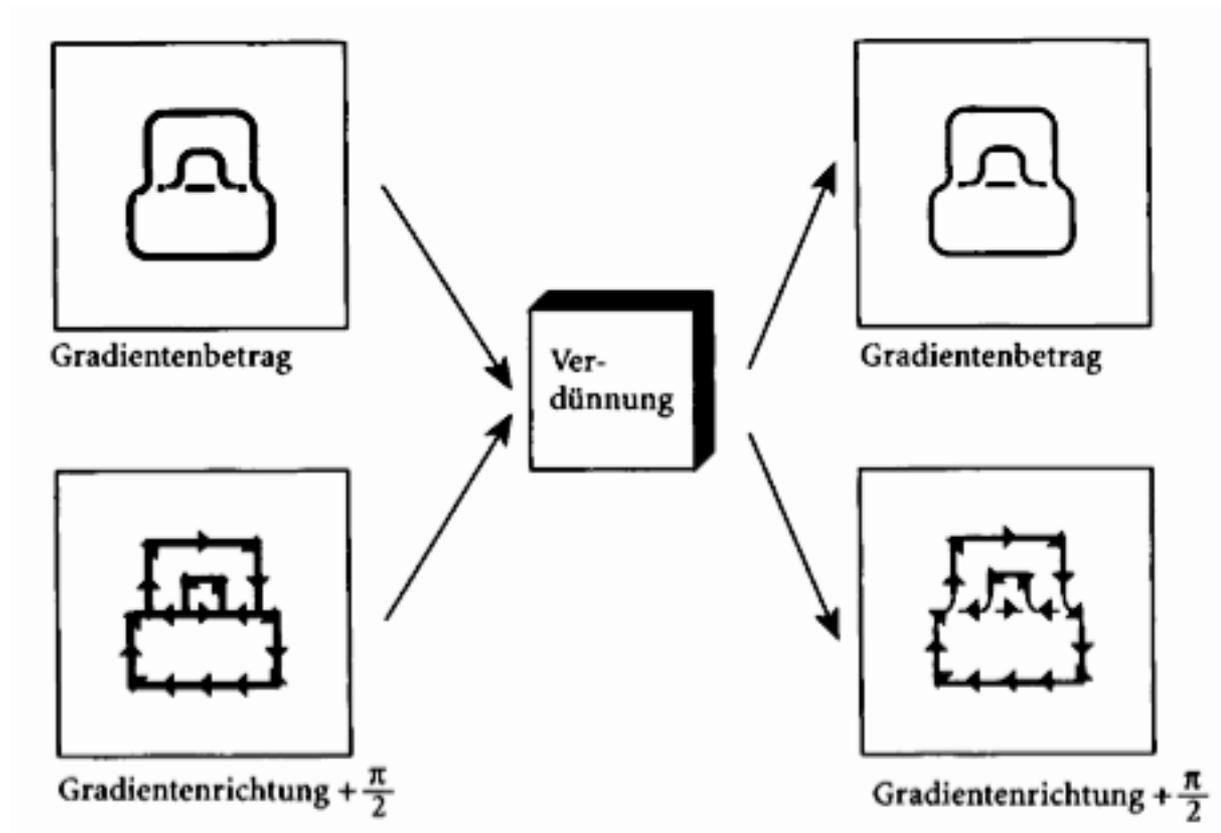


Verdünnung der Konturen:

- Beträge des Gradienten verhalten sich i.d. Regel wie langgestreckte Gebirgszüge
- Annahme: tatsächliche Objektkontur liegt beim Maximum



# Non-Maxima Suppression



# Non-Maxima Suppression

Algorithmus:

Nicht-Maxima-Unterdrückung (*Non-maxima suppression*)

Idee: lasse nur die lokalen Maxima der "Bergkette" stehen

Realisierung:

1. Einteilung der Kantenrichtungen in 8 Hauptrichtungen (entspr. 8-Nachbarschaft)
2. für jedes zur Kante gehörende Pixel betrachte die 2 Nachbarn in der Gradienten-Richtung (senkrecht zum Kantenverlauf)
3. Wenn die Kantenstärke (Gradientenbetrag) eines Nachbarn größer ist als die des aktuellen Pixels: markiere aktuelles Pixel
4. Nachdem alle Pixel bearbeitet sind: Löschen aller markierten Pixel aus dem Kantenbild

# Pyramiden

Darstellung in unterschiedlicher Skalierung

Angaben der Differenzen zwischen skalierten Bildern

Rekonstruktion mittels Differenzen und Interpolation

Anwendung für

- Merkmalsextraktion (in jeweils geeigneter Skalierung bzw. in Kombination unterschiedlich skaliertes Skalenbilder)

# Pyramiden

Problem bei Skalierung:

Geeignete Quantisierung (Abtast-Theorem)

Kombination mit einem Glättungsfilter (Tiefpass) zur Unterdrückung hoher Frequenzen  
(z.B. Beschränkung auf untere Hälfte)

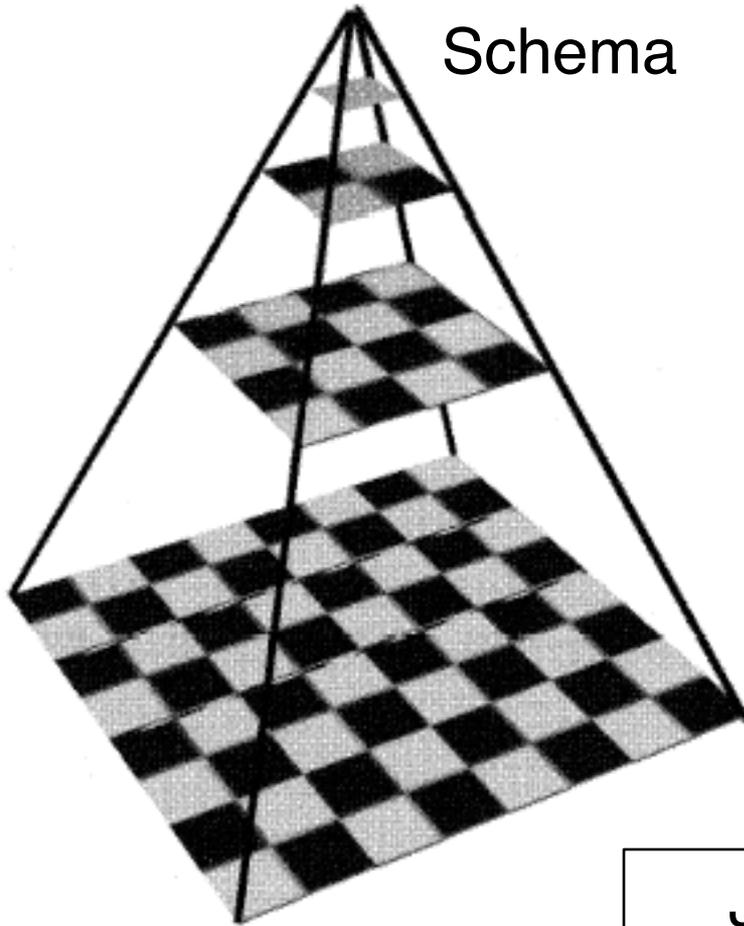
– z.B. Gauß-Filter:

Gaußpyramide: Sequenz solcher Bilder

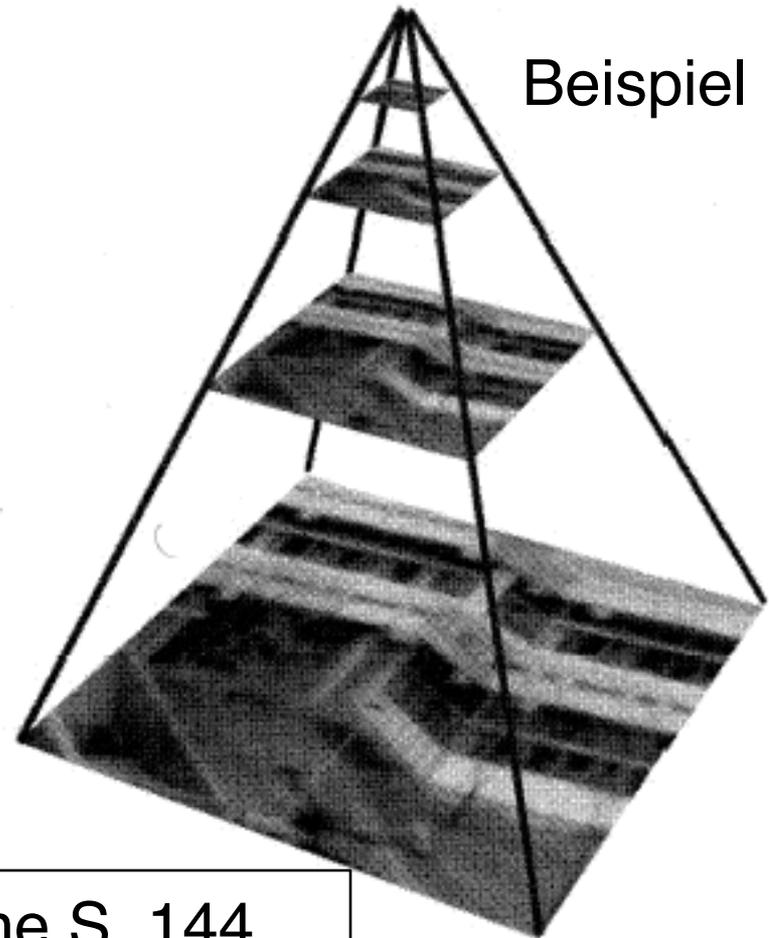
Ermöglicht Nachbarschaft für entfernte Punkte (kleine Masken)

# Gauß-Pyramiden

Schema



Beispiel



Jähne S. 144

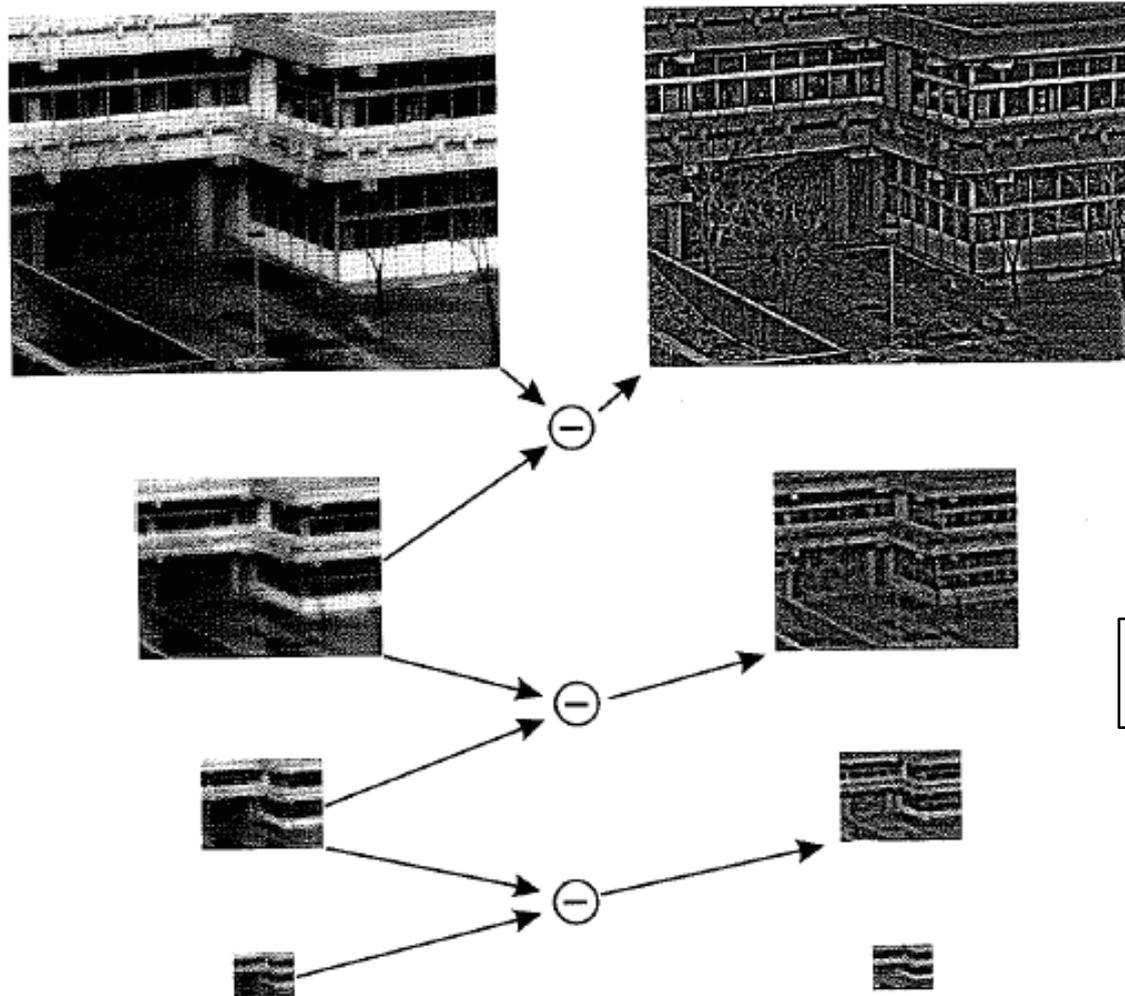
# Laplace-Pyramiden

Konstruktion als Differenz von jeweils zwei aufeinander folgenden Ebenen der Gauß-Pyramide (kleinere Ebene muss dabei expandiert werden auf Größe der anderen Ebene)

Jede Ebene enthält alle Frequenzen einer bestimmten Bandbreite unabhängig von Richtung.

Einfache Rekonstruktion des Originalbildes (früher auch für Kompression verwendet)

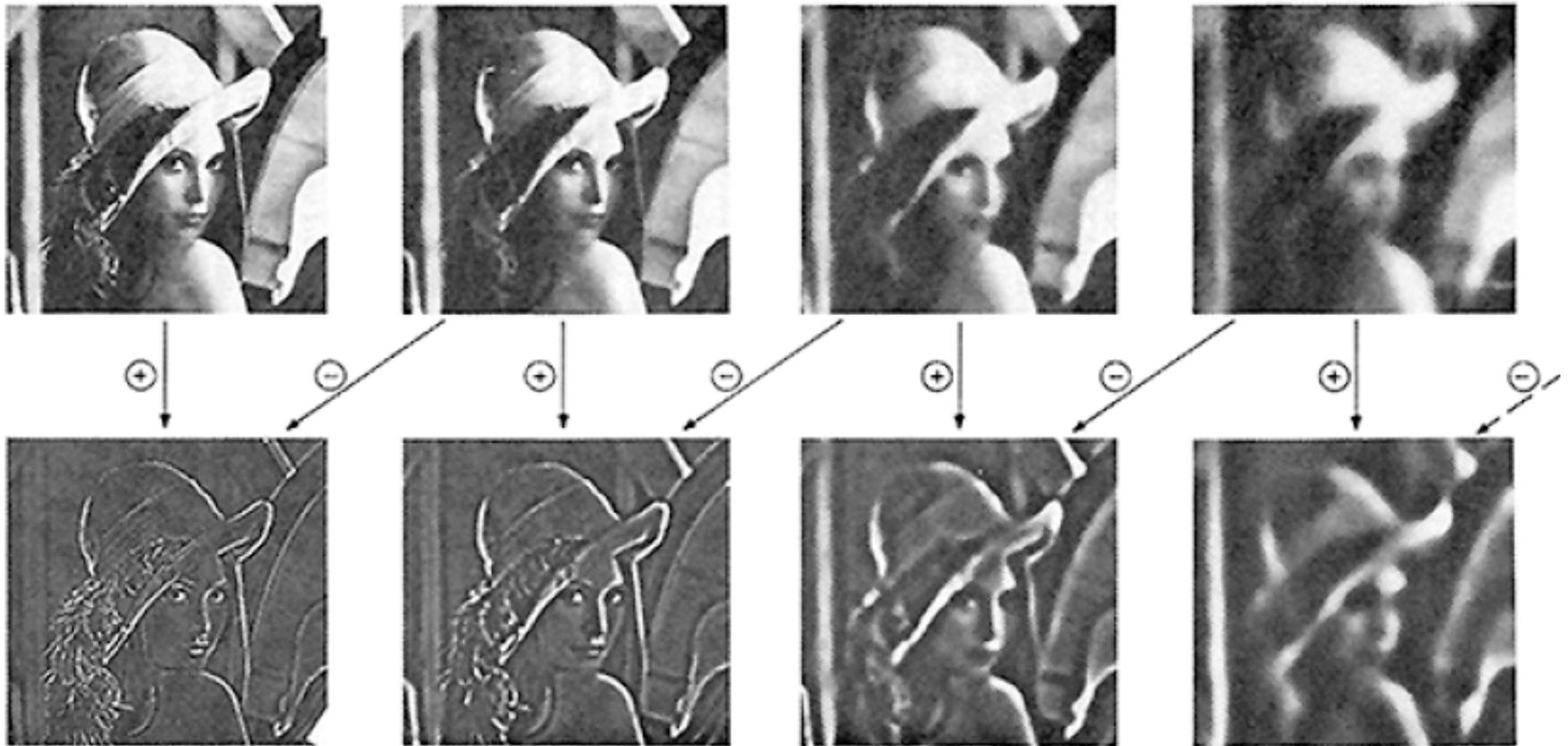
# Laplace-Pyramiden



Jähne S. 146

*Abbildung 5.6: Konstruktion der Laplacepyramide (rechts) aus einer Gaußpyramide (links) durch Berechnung der Differenzbilder zweier aufeinanderfolgender Bilder der Gaußpyramide.*

# Gauß-Laplace Pyramide



# Ortsfrequenzbereich

Übergang zu Fouriertransformierten Bildern:

Erzeugung durch überlagerte periodische Muster (Wellen),  
für diskrete (periodische) Bilder verlustlos möglich.

Hochfrequente Schwingungen:

- Störungen (global)
- Kanten (lokal)

Muster durch Spektren beschreiben

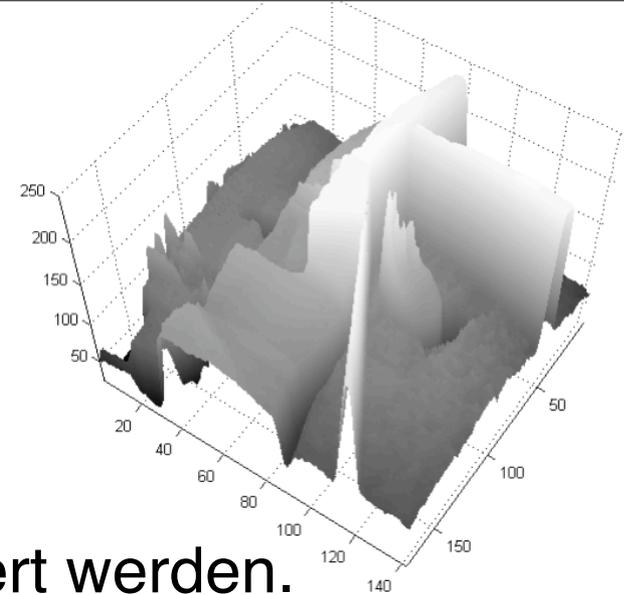
Allg.: Erzeugendensystem für Bilder, wobei ein Bild jeweils  
als Punkt in einem entsprechenden Raum erscheint

# Ortsfrequenzbereich

Grauwertgebirge:

Darstellung von  $g(x,y)$  als Funktion.

Soll durch überlagerte Wellen approximiert werden.



Für Beschreibung einer Welle sind relevant:

- Richtung
- Wellenzahl (entspricht Frequenz)
- Amplitude
- Phase

- kann durch cos/sin-Paar ersetzt werden:

$$C \cos(x-\phi) = A \cos x + B \sin x, \quad C^2 = A^2 + B^2, \quad \phi = \tan^{-1}(B/A)$$

# Diskrete Fouriertransformation

Fouriertransformation für 1 Variable:

Darstellung periodischer Funktionen  
als unendliche trigonometrische Reihen  
im Raum komplexer Zahlen

$$\begin{aligned} f(x) &= \sum c_k \cdot \exp( i 2\pi kx ) \\ &= \sum ( a_k \cdot i \sin(2\pi kx) + b_k \cdot \cos(2\pi kx) ) \end{aligned}$$

# Diskrete Fouriertransformation

Darstellung einer Funktion  $f(x)$  im Intervall  $[0, N]$   
durch Abtastung an  $N$  diskreten Punkten

$$x = 0, 1/N, 2/N, \dots, (N-1)/N$$

als endliche Summe

$$f(x) = 1/N^{1/2} \cdot \sum_{u=0, \dots, N-1} c(u) \cdot \exp(i 2\pi x u/N)$$

mit den Fourier-Koeffizienten

$$c(u) = 1/N^{1/2} \cdot \sum_{x=0, \dots, N-1} f(x) \cdot \exp(-i 2\pi u x/N)$$

$c(u)$  ist eine Funktion im Frequenzraum

$u = 0, \dots, N-1$  sind die Wellenzahlen:

Anzahl von Perioden im Intervall („Frequenz“)

# Diskrete Abtastpunkte (für 1 Variable)

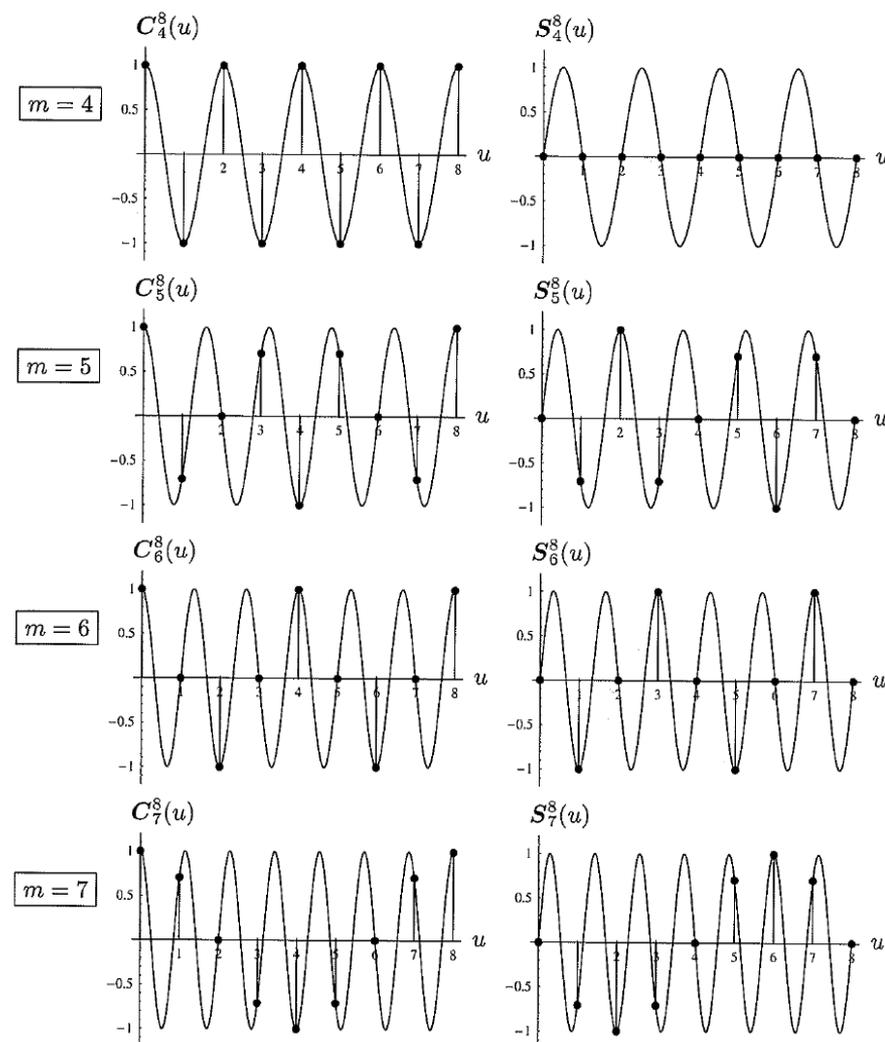
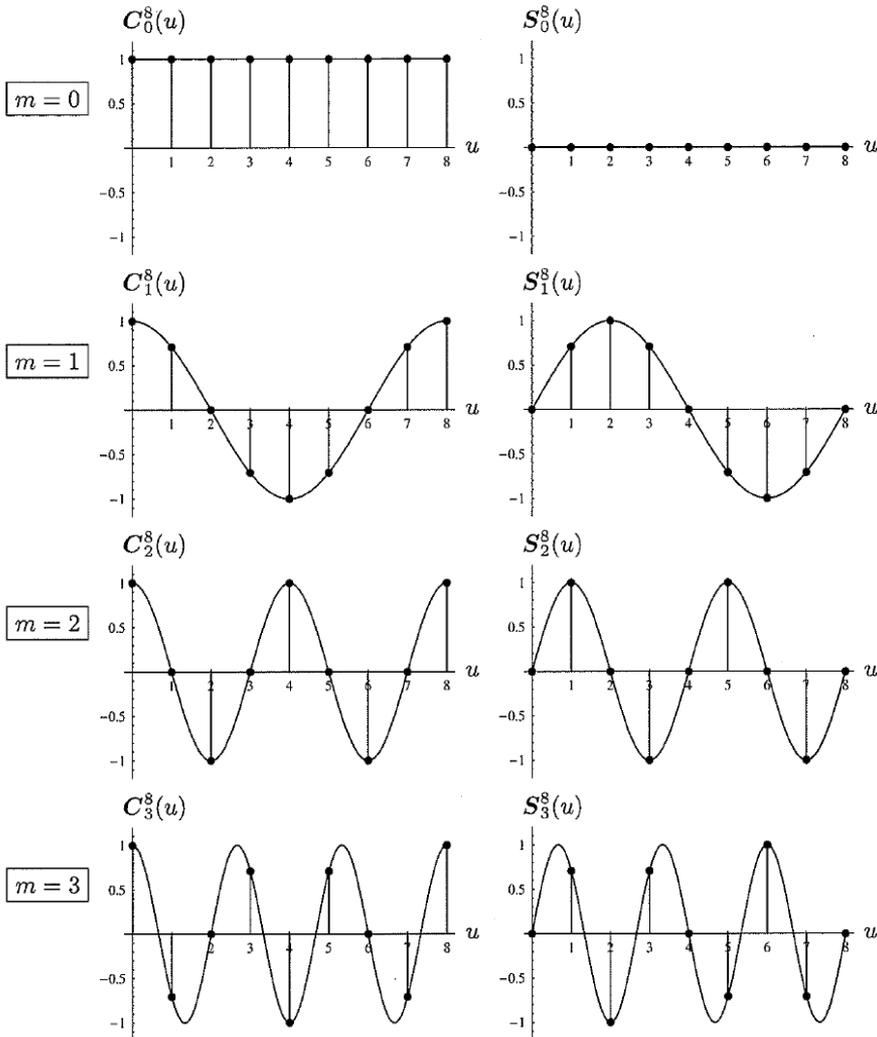
Burger/Burge S. 332f

$$C_m^8(u) = \cos\left(\frac{2\pi m u}{8}\right)$$

$$S_m^8(u) = \sin\left(\frac{2\pi m u}{8}\right)$$

$$C_m^8(u) = \cos\left(\frac{2\pi m u}{8}\right)$$

$$S_m^8(u) = \sin\left(\frac{2\pi m u}{8}\right)$$

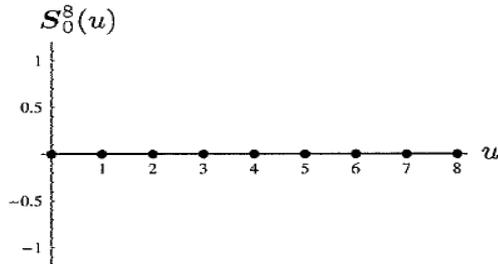
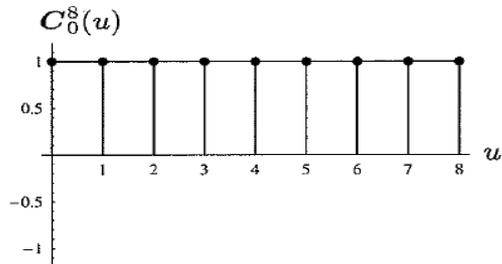


$$C_m^8(u) = \cos\left(\frac{2\pi m}{8}u\right)$$

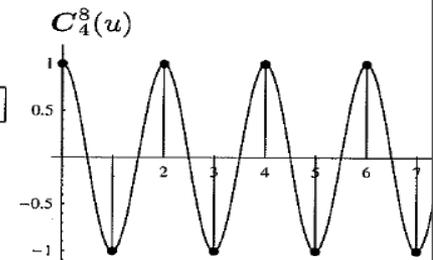
$$S_m^8(u) = \sin\left(\frac{2\pi m}{8}u\right)$$

$$C_m^8(u) = \cos\left(\frac{2\pi m}{8}u\right)$$

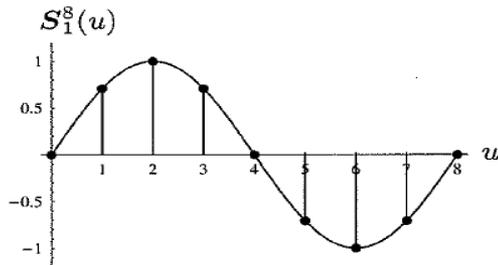
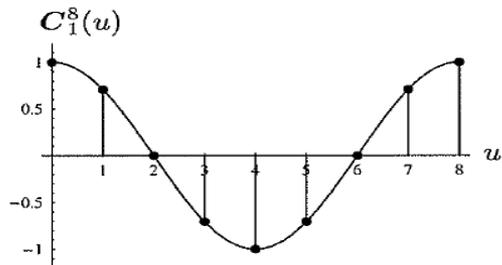
$m = 0$



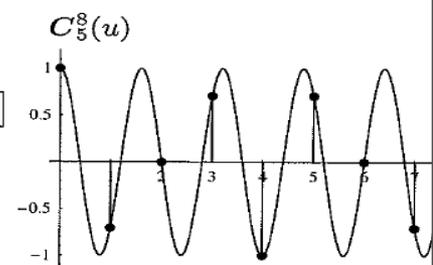
$m = 4$



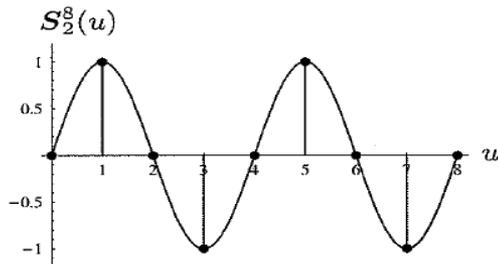
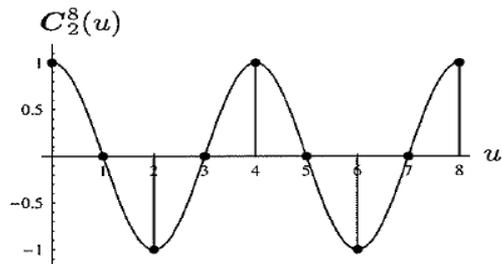
$m = 1$



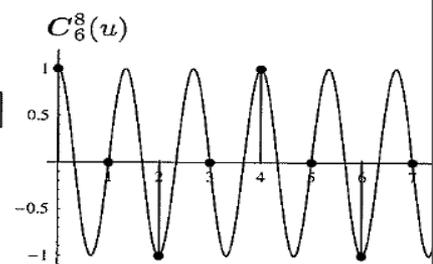
$m = 5$



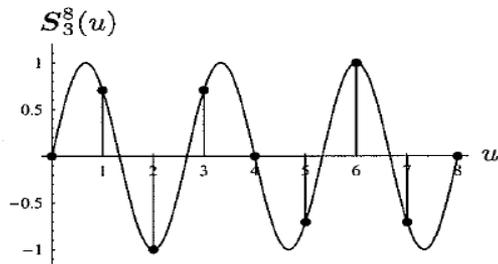
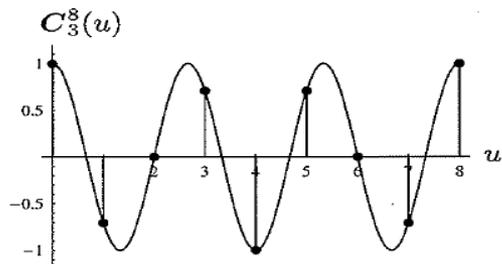
$m = 2$



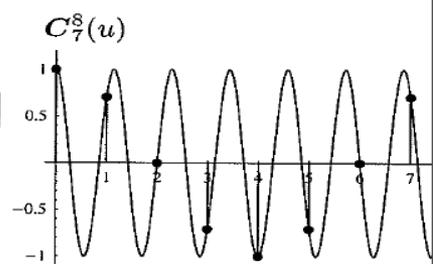
$m = 6$



$m = 3$

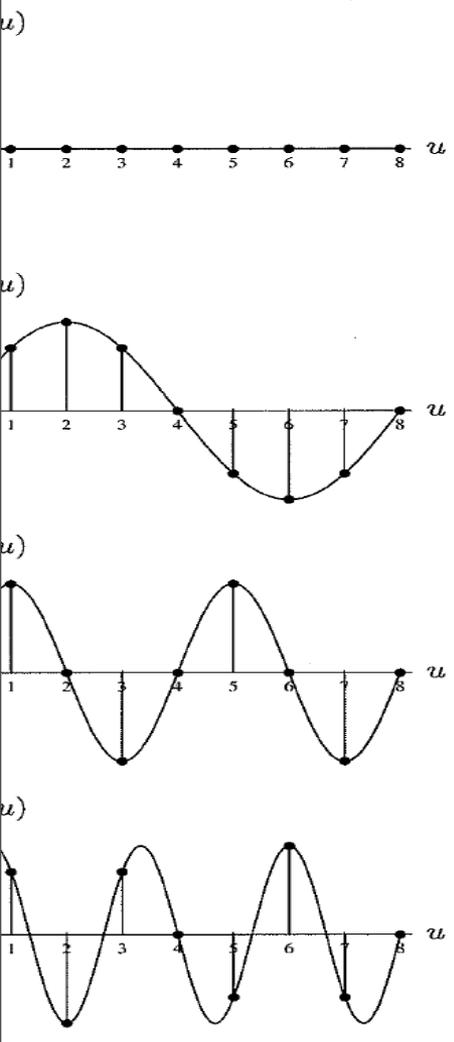


$m = 7$

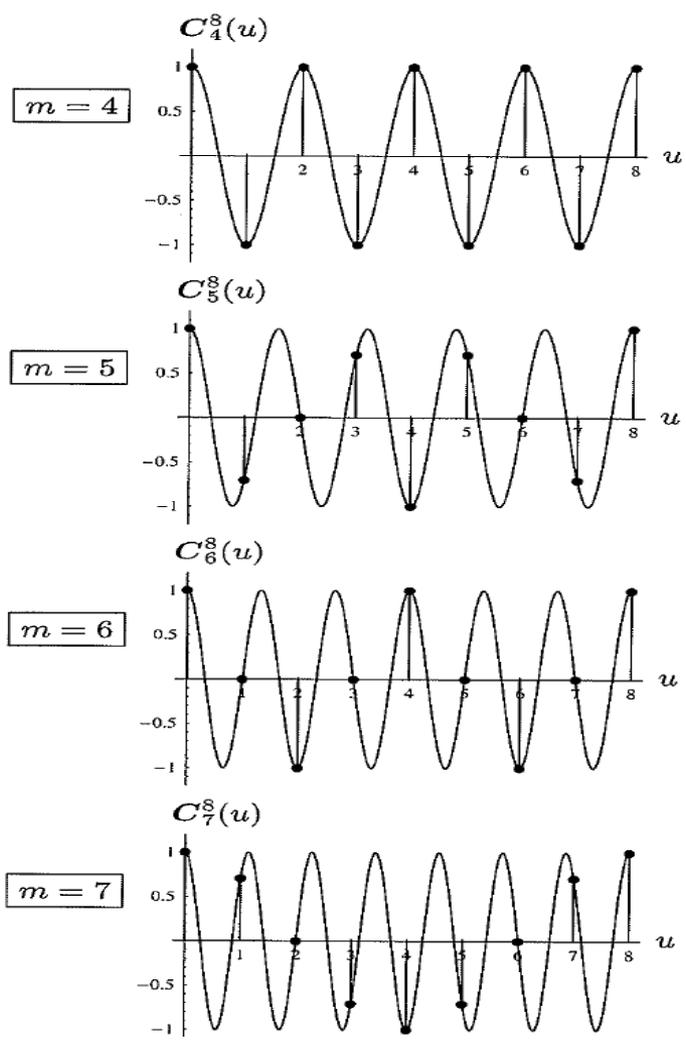


**Abb. 13.11.** Diskrete Basisfunktionen  $C_m^M(u)$  und  $S_m^M(u)$  für die Signallänge  $M = 8$  und Wellenzahlen  $m = 0 \dots 3$ . Jeder der Plots zeigt sowohl die diskreten Funktionswerte (als dunkle Punkte) wie auch die zugehörige kontinuierliche Funktion.

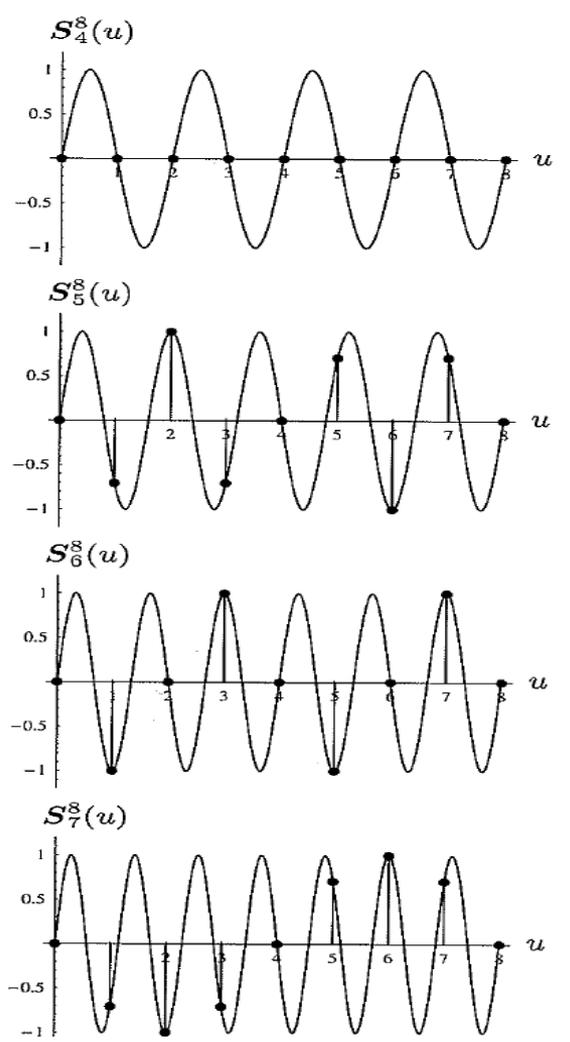
$$S_m^8(u) = \sin\left(\frac{2\pi m}{8}u\right)$$



$$C_m^8(u) = \cos\left(\frac{2\pi m}{8}u\right)$$



$$S_m^8(u) = \sin\left(\frac{2\pi m}{8}u\right)$$



**Abb. 13.12.** Diskrete Basisfunktionen (Fortsetzung). Signallänge  $M = 8$  und Wellenzahlen  $m = 4 \dots 7$ . Man beachte, dass z. B. die diskreten Funktionen für  $m = 5$  und  $m = 3$  (Abb. 13.11) identisch sind, weil  $m = 4$  die maximale Wellenzahl ist, die in einem diskreten Spektrum der Länge  $M = 8$  dargestellt werden kann.

# Diskrete Fouriertransformation

Diskrete Fouriertransformation eines Bildes

$$g(x,y) , \quad x=0,\dots,L-1 \quad y=0,\dots,M-1$$

analog durch Abtastung an den diskreten Bild-Punkten

$$(u/L, v/M) \quad \text{für} \quad u=0,\dots,L-1 , \quad v=0,\dots,M-1$$

als endliche Summe

$$g(x,y)$$

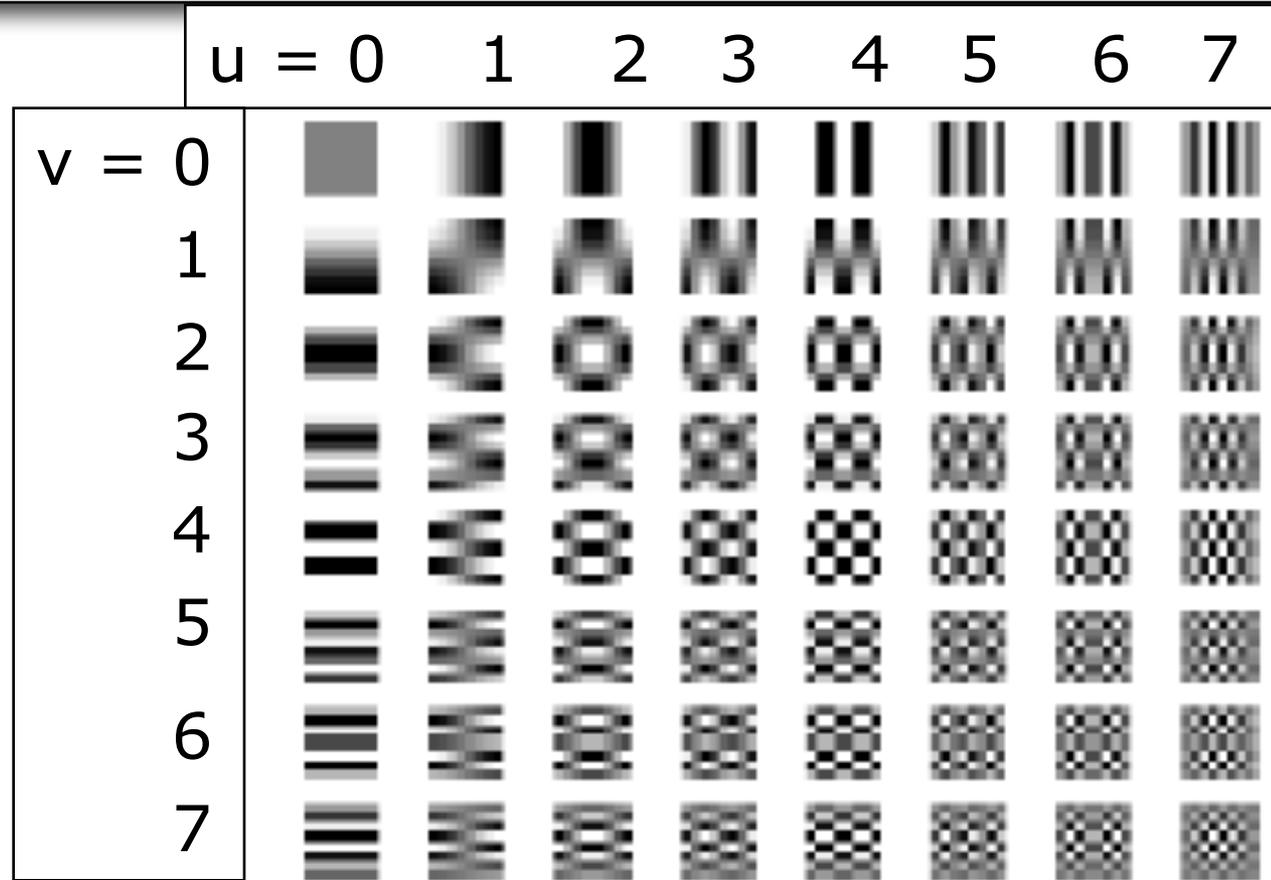
$$= (LM)^{-1/2} \sum_{u=0,\dots,L-1} \sum_{v=0,\dots,M-1} G(u,v) \cdot \exp( i 2\pi ( x u/L + y v/M ) )$$

mit den Fourier-Koeffizienten

$$G(u,v)$$

$$= (LM)^{-1/2} \cdot \sum_{x=0,\dots,L-1} \sum_{y=0,\dots,M-1} g(x,y) \cdot \exp(-i 2\pi ( u x/L + v y/M ) )$$

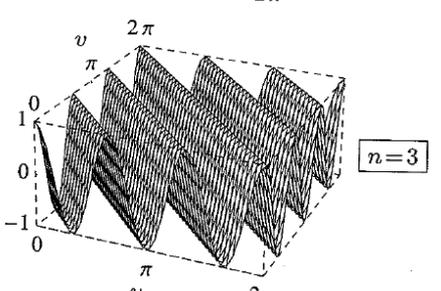
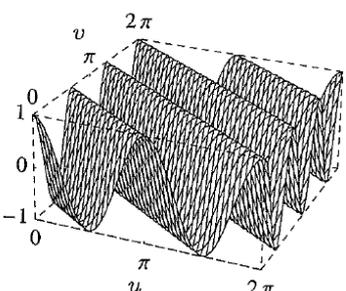
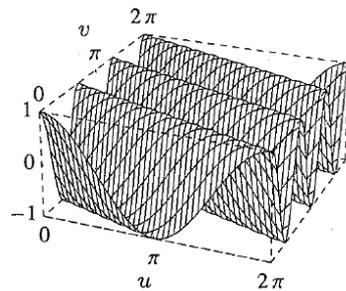
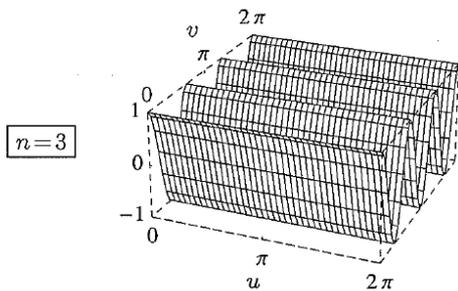
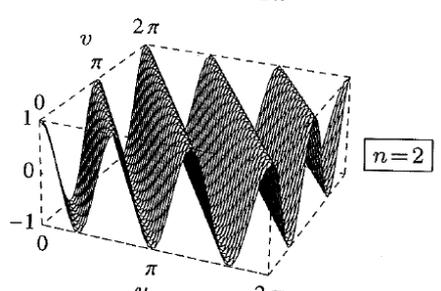
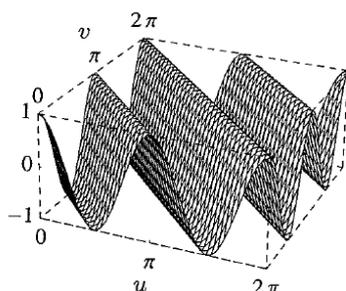
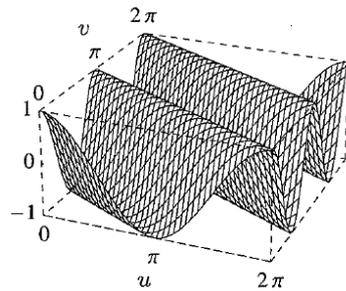
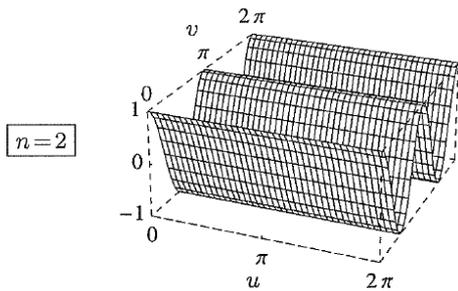
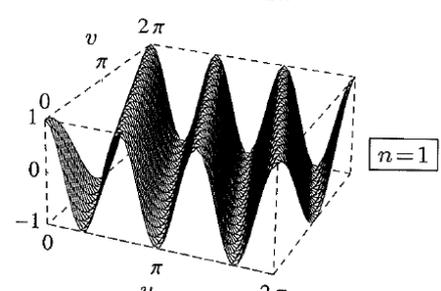
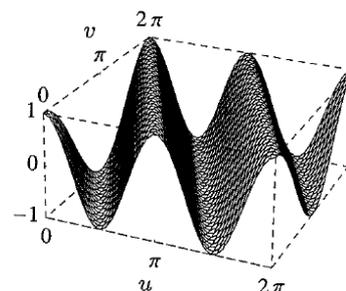
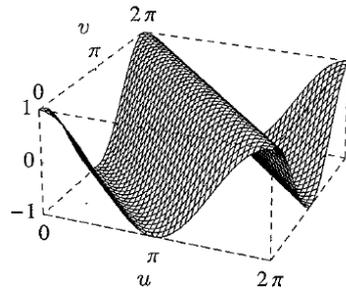
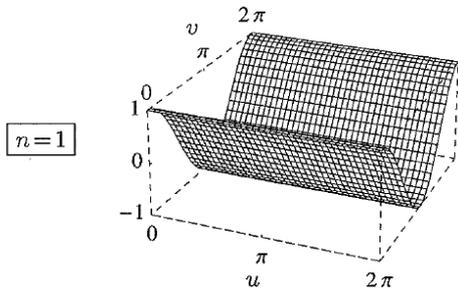
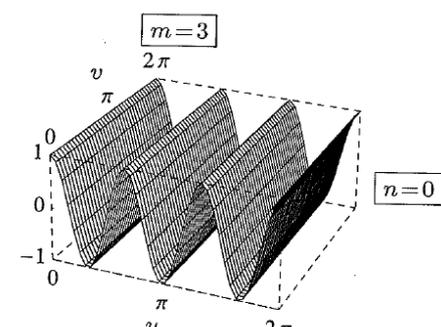
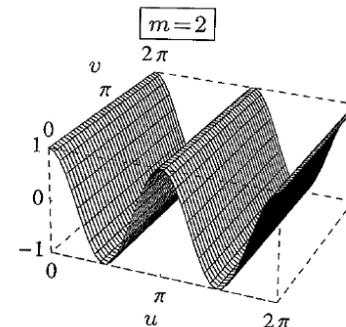
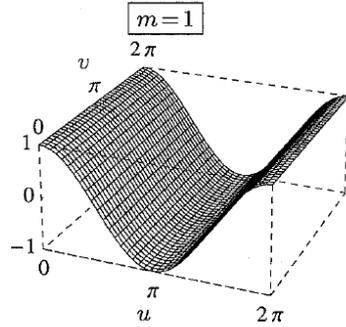
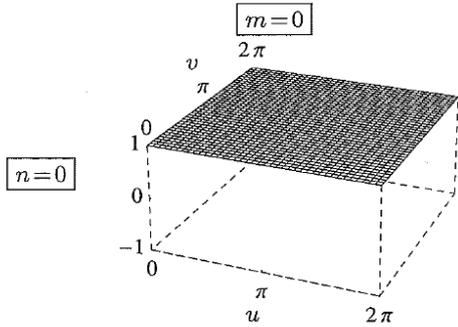
# Diskrete Fouriertransformation

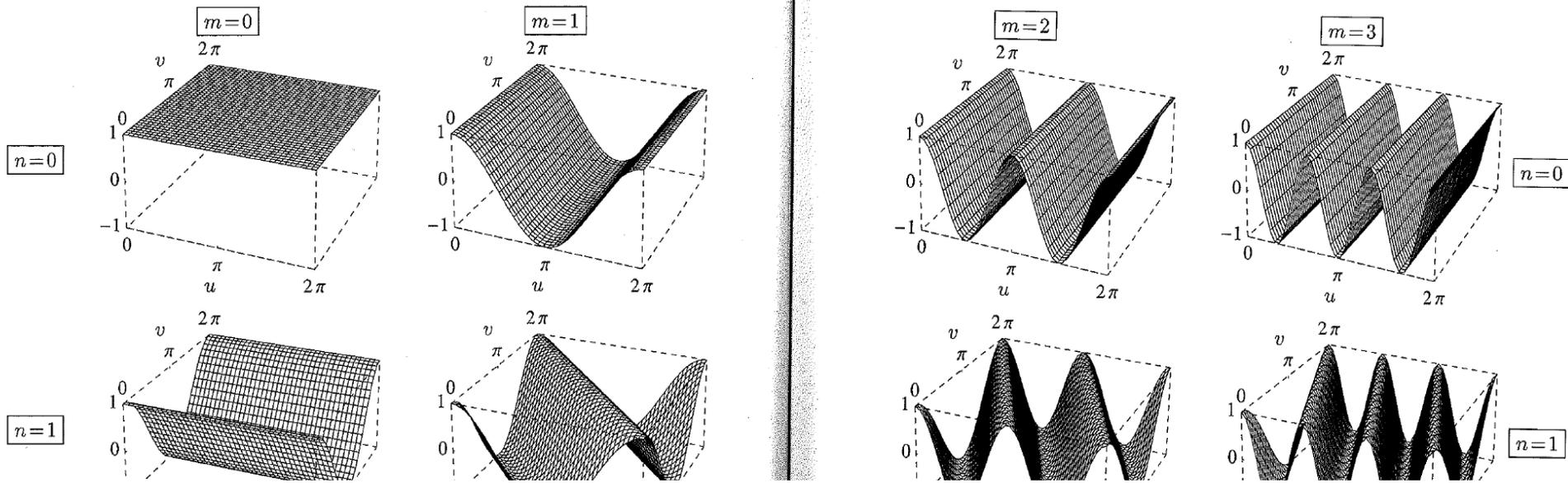


Beispiel für Basisfunktionen  
(Cosinus, Zellengröße  $N=8$ )

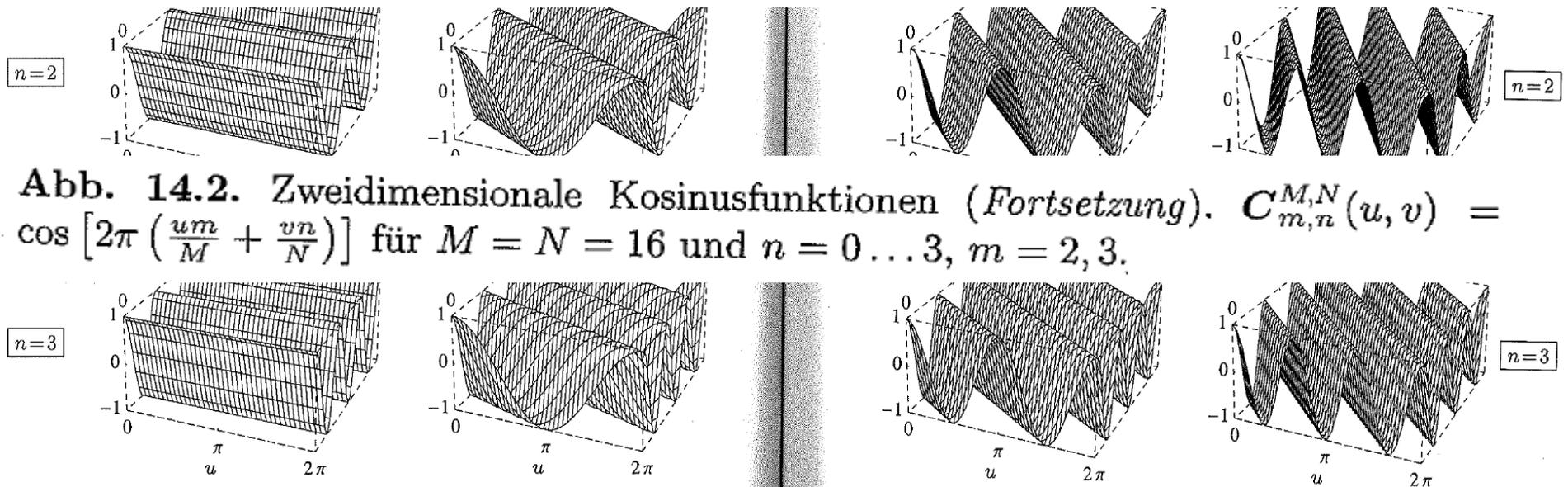
# Basisfunktionen

Burger/Burge S. 344f





**Abb. 14.1.** Zweidimensionale Kosinusfunktionen.  $C_{m,n}^{M,N}(u,v) = \cos \left[ 2\pi \left( \frac{um}{M} + \frac{vn}{N} \right) \right]$   
 für  $M = N = 16$  und  $n = 0 \dots 3$ ,  $m = 0, 1$ .



**Abb. 14.2.** Zweidimensionale Kosinusfunktionen (*Fortsetzung*).  $C_{m,n}^{M,N}(u,v) = \cos \left[ 2\pi \left( \frac{um}{M} + \frac{vn}{N} \right) \right]$   
 für  $M = N = 16$  und  $n = 0 \dots 3$ ,  $m = 2, 3$ .

# Ortsfrequenzbereich

In einem Bild darstellbare Wellen sind durch die Dimensionen  $M$  und  $N$  der Bildmatrix begrenzt.

Wellenzahlen:  $u$  und  $v$

Maximal darstellbare Frequenz (Abtast-Theorem) des Ortsraums in  $x$ -Richtung,  $y$ -Richtung bzw. diagonal:

$$L/2 \text{ bzw. } M/2 \text{ bzw. } 1/2 \sqrt{L^2+M^2}$$

# Fouriertransformiertes Bild

$G(u,v)$  für  $u=0,\dots,L-1$   $v=0,\dots,M-1$

hat wie das ursprüngliche Bild  $g(x,y)$  die Form einer Bildmatrix der Dimension  $L \times M$  im **Ortsfrequenzraum**.

$G(u,v)$

$$=(LM)^{-1/2} \cdot \sum_{x=0,\dots,L-1} \sum_{y=0,\dots,M-1} g(x,y) \cdot \exp(-i 2\pi (u x/L + v y/M) )$$

heißt **Fouriertransformierte des Bildes  $g(x,y)$**  ,

Umgekehrt ist

$g(x,y)$

$$= (LM)^{-1/2} \sum_{u=0,\dots,L-1} \sum_{v=0,\dots,M-1} G(u,v) \cdot \exp( i 2\pi ( x u/L + y v/M) )$$

die inverse **Fouriertransformierte von  $G(u,v)$**  .

# Spektrum der Fouriertransformierten

Die Fouriertransformierte ist komplex:

$$G(u,v) = G_1(u,v) + i G_2(u,v)$$

$G(u,v)$

$$= (LM)^{-1/2} \cdot \sum_{x=0,\dots,L-1} \sum_{y=0,\dots,M-1} g(x,y) \cdot \exp(-i 2\pi(xu/L + yv/M))$$

$$= (LM)^{-1/2} \cdot \sum_{x=0,\dots,L-1} \sum_{y=0,\dots,M-1} g(x,y) (-i \sin(2\pi(xu/L + yv/M)) + \cos(2\pi(xu/L + yv/M)))$$

$$= -i \cdot (LM)^{-1/2} \cdot \sum_{x=0,\dots,L-1} \sum_{y=0,\dots,M-1} g(x,y) \sin(2\pi(xu/L + yv/M)) + (LM)^{-1/2} \cdot \sum_{x=0,\dots,L-1} \sum_{y=0,\dots,M-1} g(x,y) \cos(2\pi(xu/L + yv/M))$$

# Spektrum der Fouriertransformierten

Das „**Fourier-Spektrum**“  $|G(u,v)| = \sqrt{G_1^2(u,v) + G_2^2(u,v)}$

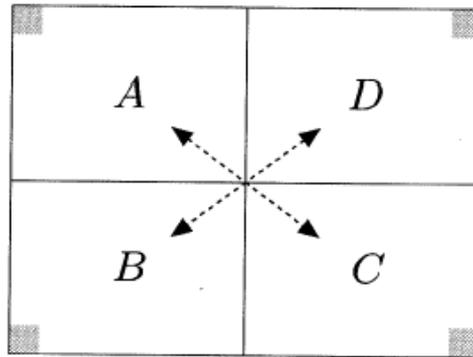
beschreibt den Anteil der Frequenzen im Bild.

Das Fourier-Spektrum ist als Grauwert-Bild darstellbar.

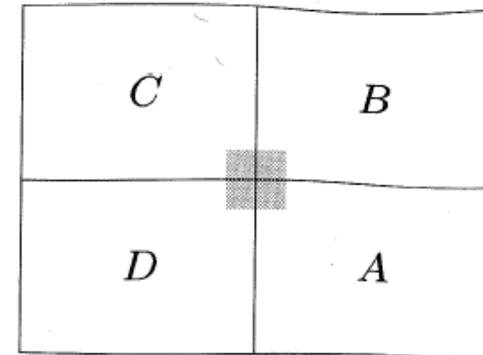
Dazu erfolgt meistens Normierung:

- zentralsymmetrische Darstellung  
(Spektrum für reelle Bilder ist symmetrisch),  
realisierbar durch Vertauschung der Quadranten.
- geringe Frequenzen (hoher Anteil) im Zentrum
- logarithmische Skalierung (  $\log(1+|G(u,v)|)$  )

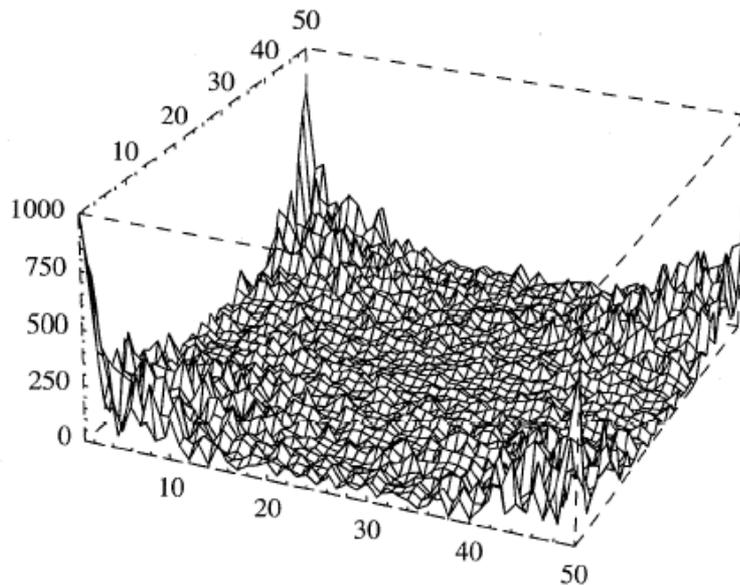
# Spektrum der Fouriertransformierten



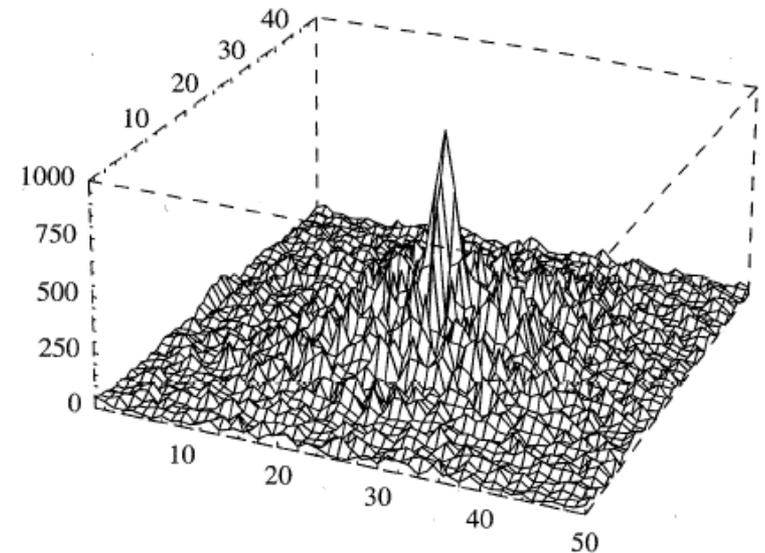
(a)



Burger/Burge S. 347



(c)



(d)

# Spektrum der Fouriertransformierten



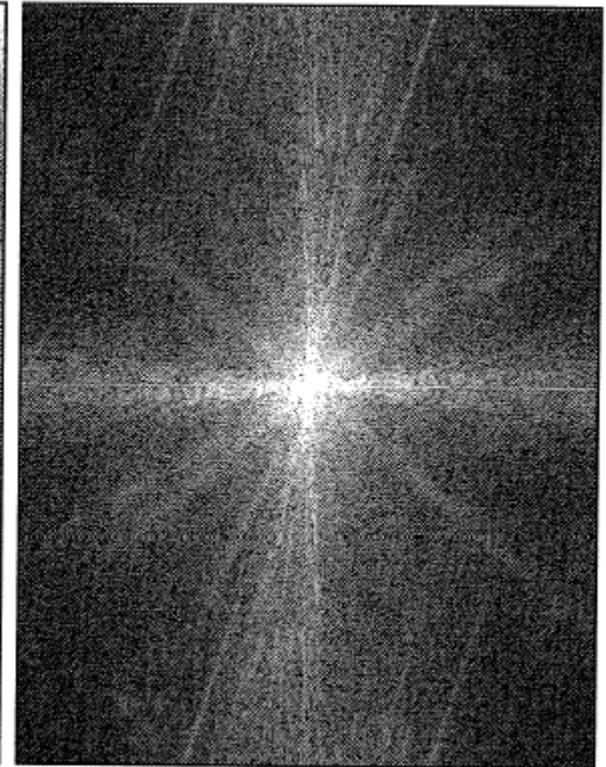
(a)

Original



(b)

$|G(u,v)|$  ohne Normierung

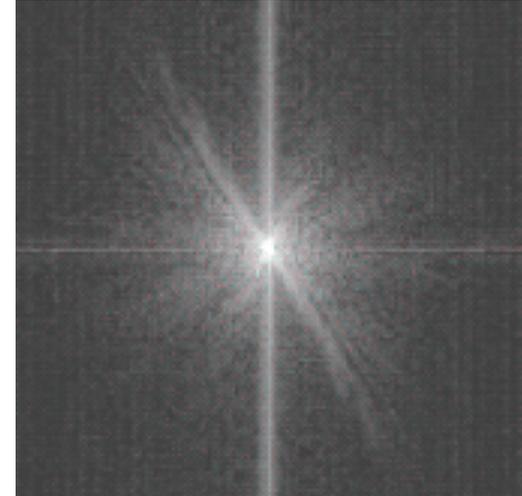
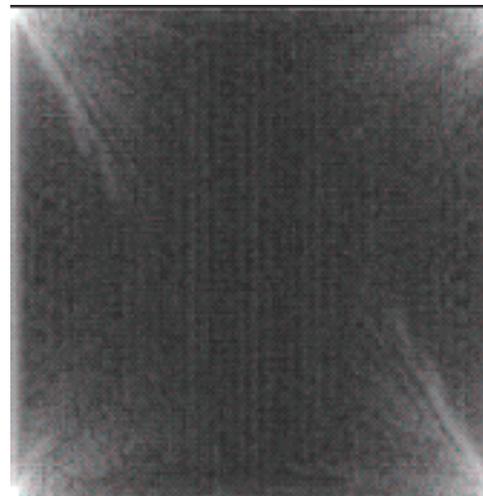


(c)

mit Normierung

Burger/Burge S. 347

# Spektrum der Fouriertransformierten



Original

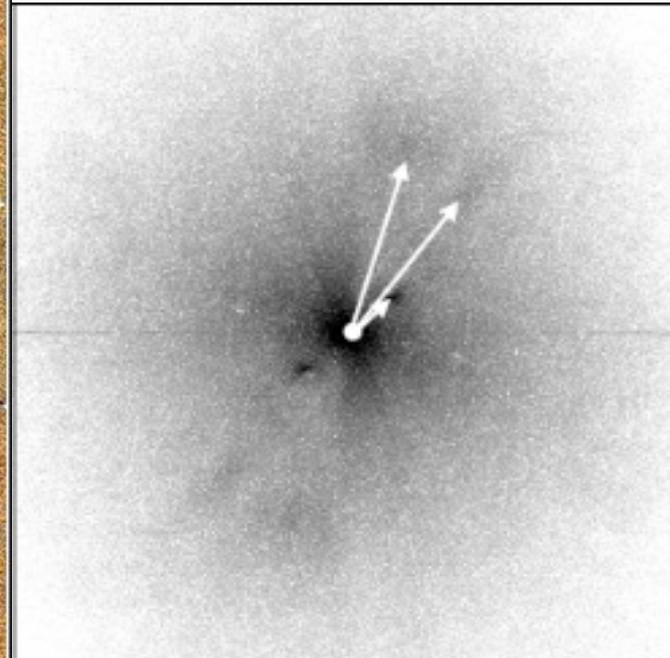
$|G(u,v)|$  ohne Normierung

mit Normierung

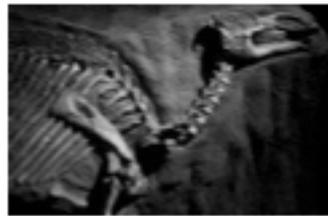
# Spektrum der Fouriertransformierten



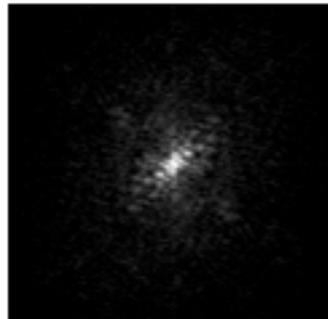
SAR-Aufnahme des indischen Ozeans mit Wasserwellen unterschiedlicher Wellenlänge



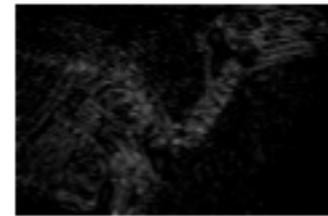
# Ausfiltern von hohen Frequenzen



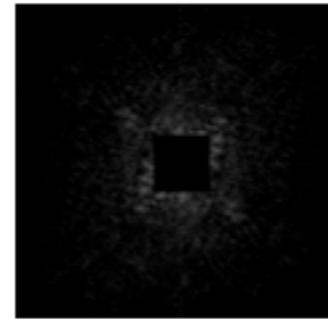
↓ DFT



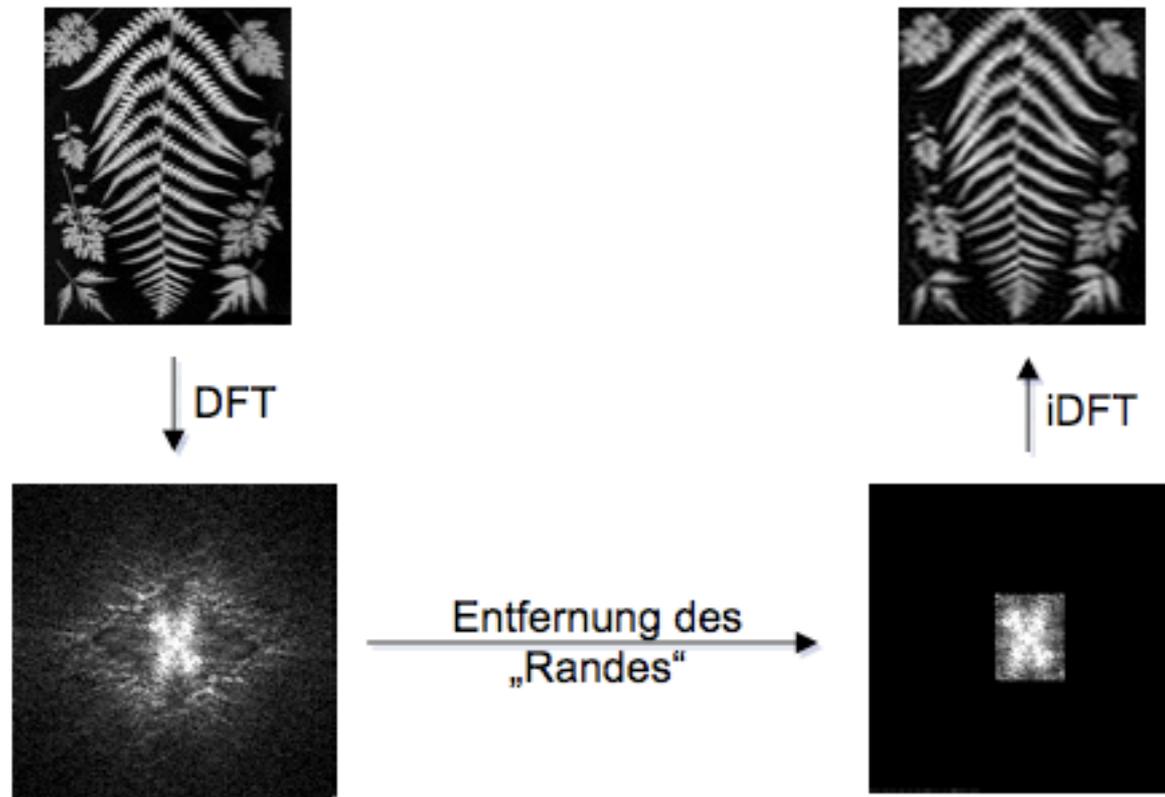
Entfernung der „Mitte“ →



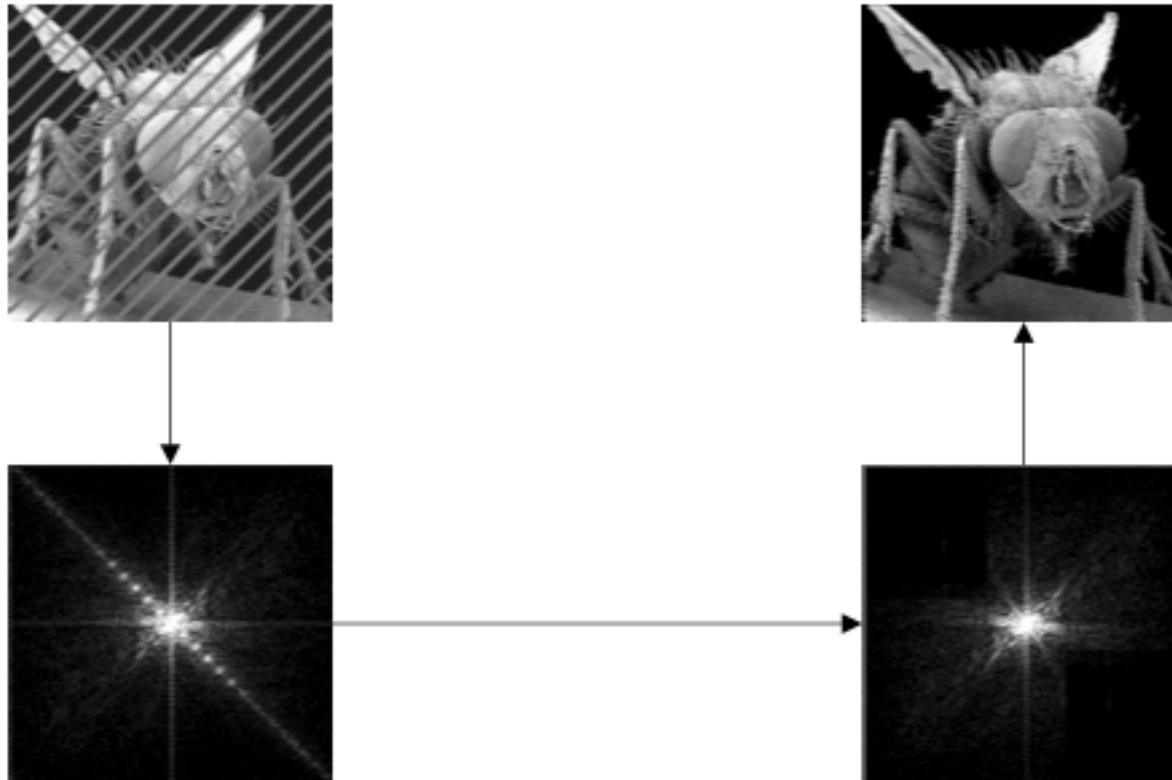
↑ iDFT



# Ausfiltern von niedrigen Frequenzen



# Filterung mit Fouriertransformation



# Operationen im Ortsfrequenzraum

Direkte Manipulation von Frequenzen.

Beispiele:

Tiefpassfilter (für Rauschunterdrückung)

- Elimination der hohen Frequenzen

Hochpassfilter (für Kantenextraktion)

- Elimination der tiefen Frequenzen

Bandpassfilter

- speziellen Frequenzbereich erhalten

# Operationen im Ortsfrequenzraum

Zusammenhang zum Ortsbereich

Faltung  $g*h(x,y) = \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x-u,y-v) \cdot h(u,v)$

im Ortsbereich = Multiplikation im Frequenzbereich

Bild:  $g(x,y) \Rightarrow G(u,v)$

Maske:  $h(x,y) \Rightarrow H(u,v)$

Faltung:  $g*h(x,y) \Rightarrow G(u,v) \cdot H(u,v)$

# Operationen im Ortsfrequenzraum

Anwendung von Filterverfahren im Frequenzraum

- Berechnung der Fouriertransformierten  $G(u,v)$
- lokale Multiplikation von  $G(u,v)$  mit einem Filter  $H(u,v)$   
$$G'(u,v) := G(u,v) \cdot H(u,v)$$
- Berechnung der Inversionen Fouriertransformierten von  $G'(u,v)$

Effiziente Verfahren:

Schnelle Fouriertransformation (Symmetrien nutzen)

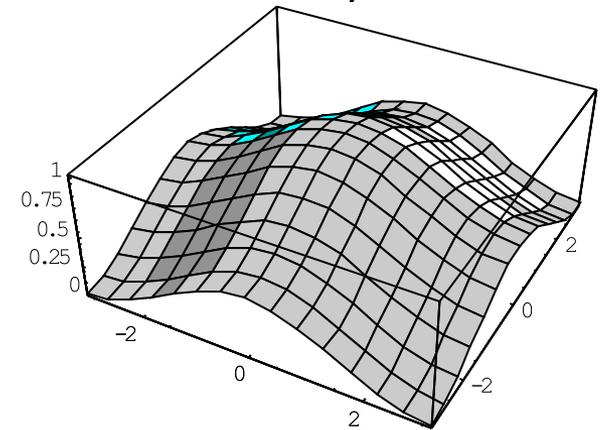
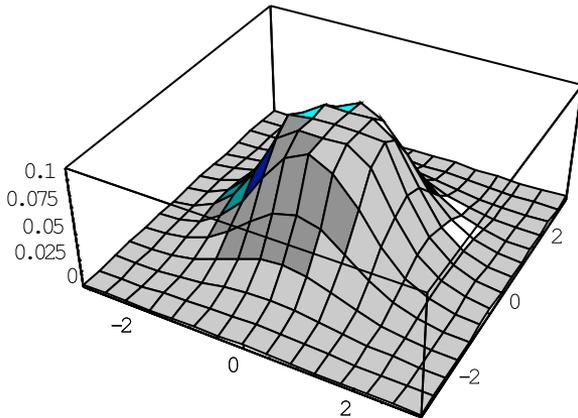
# Mittelwert vs. Gauss

Mittelwert-Filter (Resultat):

$$g'(x,y) = 1/8 ( 4g(x,y) + g(x-1,y) + g(x,y-1) + g(x,y+1) + g(x+1,y) )$$

Zugehörige Fouriertransformierte („Transferfunktion“):

$$h(u,v) = 1/8 ( 4 + 2 \cos(u) + 2 \cos(v) )$$



Gauss-Filter (kontinuierlicher Kern):

$$h(x,y) = 1/2\pi \exp( - (x^2+y^2)/2 )$$

(Transferfunktion ebenfalls eine Gauss-Funktion)

# Weitere Anwendungen

Analyse von Frequenzen

Bildvergleich durch Vergleich des Spektrums

Analyse/Vergleich von Texturen.

# Bildanalyse

Identifizierung von Komponenten (Kanten, Flächen)

Identifizierung von Objekten

Beziehungen zwischen Objekten

Interpretation der Szene

ggf. Interpretation von Bildfolgen (Abläufe)

## Schichtenmodelle

- streng sequentiell
- oder Hypothesenbasiert

(mit Backtracking, z.B. Blackboard-Architektur)

# Schichtenmodell

Vorverarbeitung (Aufbereitung, Anpassung)

- Digitalisieren
- Entzerren (Sensormodell, Kalibrieren)
- Verbessern (ohne spezielles Störmodell):  
Filteroperationen zur Glättung bzw. Verschärfung,  
evtl. Ausführung im Ortsfrequenzraum (Fouriertransf.)
- Restaurieren (mit bekanntem Störmodell)  
z.B. Bewegungsunschärfe, atmosph. Störungen
- Rekonstruktion
- Normierung (Größe, Lage, Grauwerte, Farben)

# Schichtenmodell

## Segmentierung:

Klassifizierung von Pixeln

Identifizierung von Bildprimitiven:

- Kanten aus Kantenelementen  
(„edgel“ = edge element, Pixel auf einer Kante)
- Bereiche (abgegrenzte Pixelmengen)  
bestimmt durch Grenzlinie, gemeinsame Textur  
(„texel“ = texture element, Pixelmenge mit  
spezieller Textur, d.h. spezieller Grauwertstruktur)

Zerlegung des Bildes in Bereiche

# Schichtenmodell

## Komponentenerkennung

Kombination von Bildprimitiven zu Komponenten

Verwendung von zusätzlichem Wissen

## Szeneninterpretation

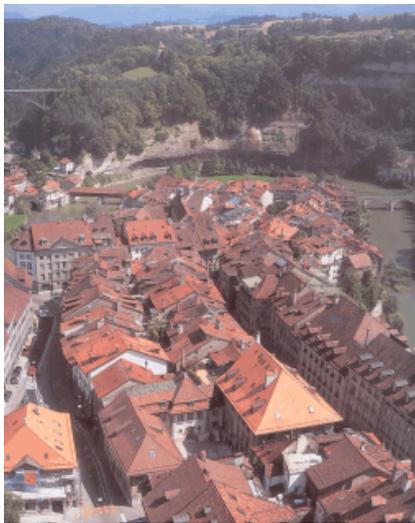
# Kanten, Linien

Grundlegend für menschliche Orientierung

Begrenzung von Körpern bzw. Flächen

Sprung der Intensität (Begrenzung)

Maximum bzw. Minimum der Intensität (Linie)



# Kanten, Linien

Kante: Übergang zwischen unterschiedlichen Bereichen

- Grauwertkante

großer Gradient in  $g(x,y)$



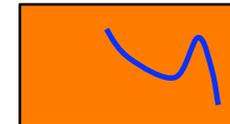
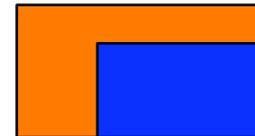
- Texturkante

durch Vorverarbeitung in Grauwertkante überführen



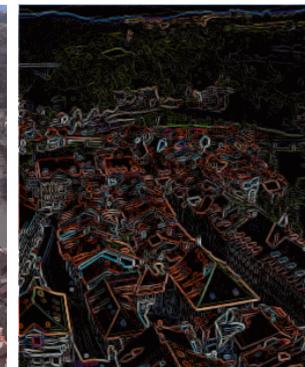
- Farbkante

Grauwertkanten in Farbkanälen



Linie = 2 naheliegende, parallele Kanten

Differenzoperatoren  
erzeugen Linien aus  
Kanten



# Kantenerkennung

Ausgangssituation: Bild  $g = [ g(x,y) ]_{x,y}$

Ergebnis: Kanten

Repräsentationsformen für Kanten:

Quantitativ, z.B.

- Mengen von Punkten:  
Koordinaten  $p=(x,y)$   
bzw. Koordinaten mit Richtung  $p=(x,y,r)$
- Kurvengleichung (Gerade, Kreis, ...)
- Gerade mit Anfangspunkt, Richtung, Länge

Qualitativ, z.B.

- „großer Kreis am linken Bildrand“

# Kantenerkennung

Mehrere Schritte, insbesondere

1) Identifizierung relevanter Bild-Punkte:

„Edgels“ = edge element: Punkt (+Richtung)

- Merkmal z.B. Grauwert-Änderung im Originalbild
- abgeleitete Merkmale nach Filterung

2) Zusammenfassung von Edgels zu Kanten

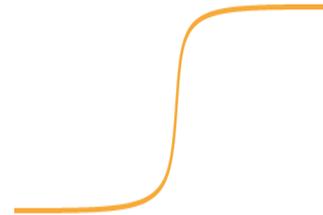
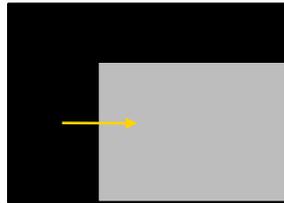
- z.B. Hough-Transformation, Regression, Suchverfahren, Korrelation mit Kanten-Maske

Vorbereitung/Zwischenschritte mit weiteren Filteroperationen  
z.B. Rauschunterdrückung, Binärbilderzeugung,  
Kantenverstärkung, Vervollständigung

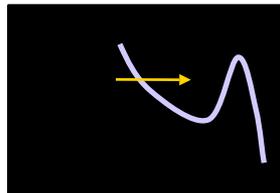
# Kantenextraktion

„Kurvendiskussion“

Kante = Sprungstelle der Grauwertverteilung



Linie = Maximum (Minimum) in der Grauwertverteilung  
bzw. zwei Sprungstellen der Grauwertverteilung



# Kantenextraktion: Edgel

- Gradient: Betrag, Richtung
  - Lokale Differenz-Operatoren
  - Sobel-Operator



## Sobel-Operator in x- bzw. y-Richtung

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

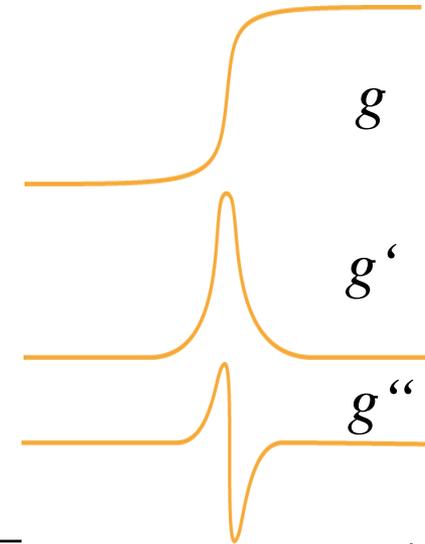
1	0	-1
2	0	-2
1	0	-1

# Kantenextraktion: Edgel

– Wendepunkt

Nullstelle der 2. Ableitung

$$\nabla^2 g(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2} = 0$$



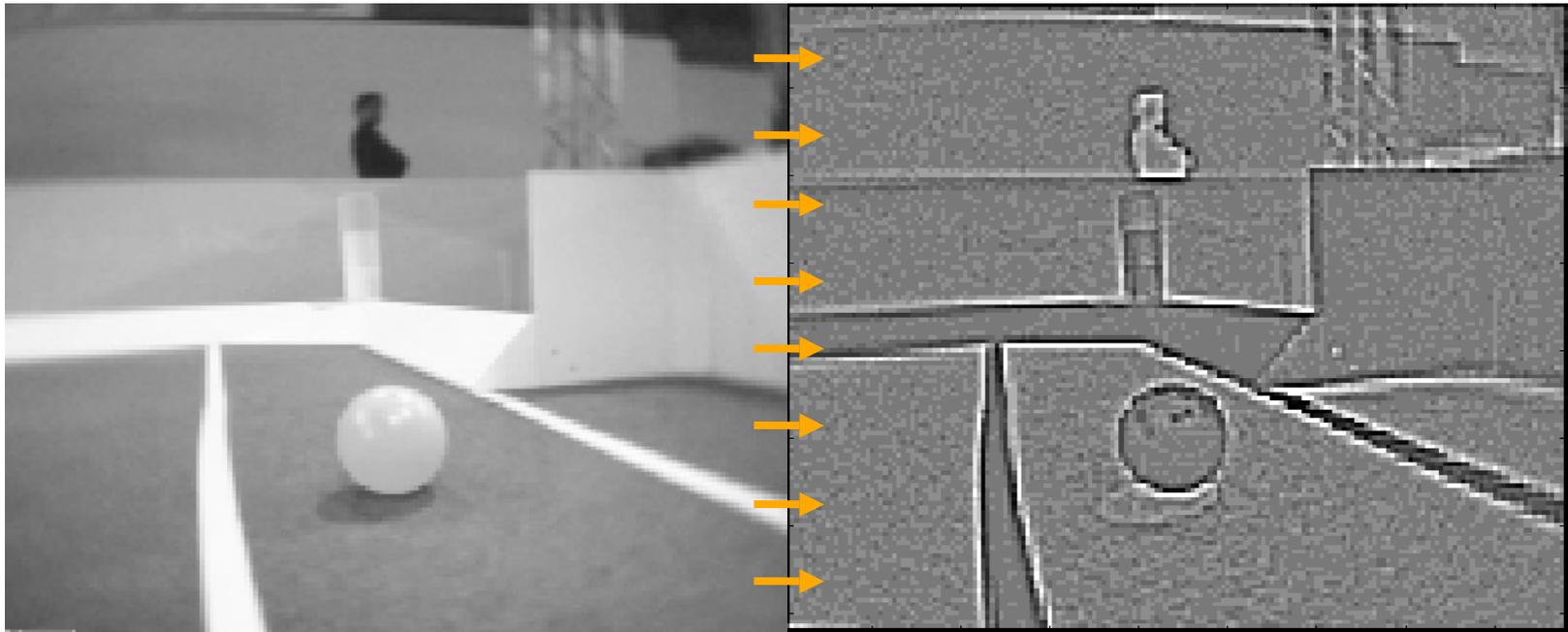
Diskrete Laplace-Operatoren z.B.

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1

# Kantendetektion mit Laplace Operator



# Kantenextraktion: Edgel

– Hochpass-Filter im Frequenzbereich

– Berechnung der Fouriertransformierten  $G(u,v)$

– lokale Multiplikation von  $G(u,v)$

mit einem Hochpass-Filter  $H(u,v)$

zur Elimination der tiefen Frequenzen:

$$G'(u,v) := G(u,v) \cdot H(u,v)$$

– Berechnung der

Inversionen Fouriertransformierten von  $G'(u,v)$

# Kantenextraktion: Edgel

- Morphologische Kantenoperatoren für Binärbild

- Symmetrische Differenz  $Dilation(G) XOR G$  ergibt Kanten an den Übergängen Schwarz/Weiß.
- Analog mit Erosion möglich, allerdings werden Linien (Breite nur 1 Pixel) eliminiert

# Kantenfindung aus Edgels

## Schwellwertbildung: Binärbild der Edgels

- Störungen beheben
- „Verdünnen“ auf Linie

## Kombinationsverfahren:

- Hough-Transformation
- Suchverfahren (z.B. Bergsteigen,  $A^*$ )
- Regression
- Template-Vergleich (Korrelation)

# Kantenfindung aus Edgels: Suchen

Suchverfahren:

Bild als Graph, Übergänge zwischen benachbarten Pixeln

Kosten:

Entfernung, Grauwertänderung, Orientierungsänderung,  
Gradient, ...

Suche nach Kosten-minimalem Weg

Suchverfahren auch im Originalbild möglich  
(ohne Bestimmung von Edgels)

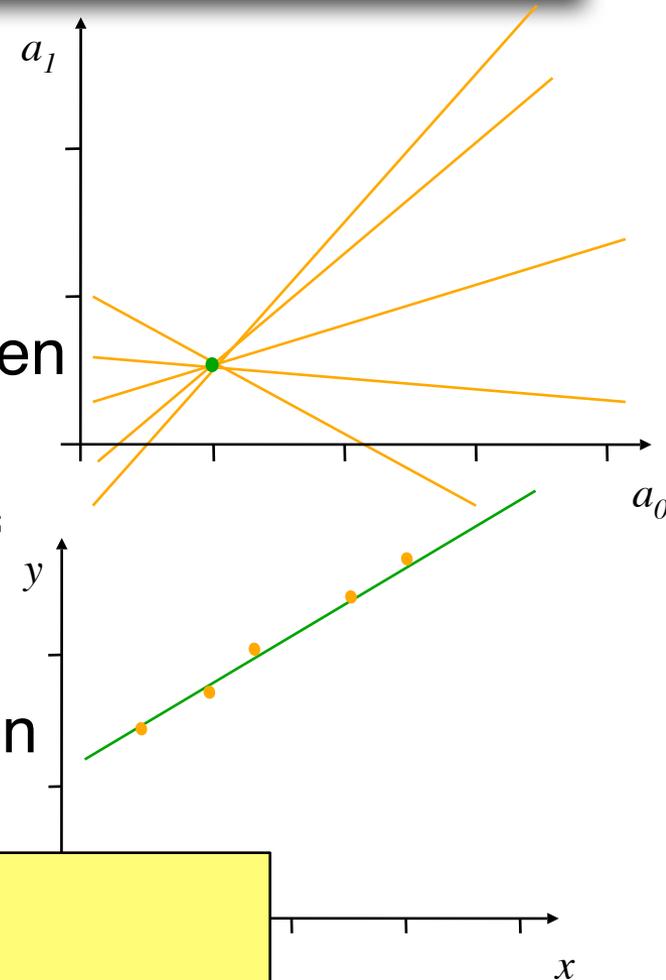
# Hough-Transformation

Ausgangspunkt: Menge von Edgels

Resultat: Menge von Geraden

Idee:

- Jedes Edgel kann auf unterschiedlichen Geraden liegen
- Voting-Verfahren: Edgel „stimmen ab“ für „ihre Geraden“
- Die Geraden mit den meisten Stimmen (Edgels) gewinnen



Gleichzeitig mehrere Kanten erkennbar.

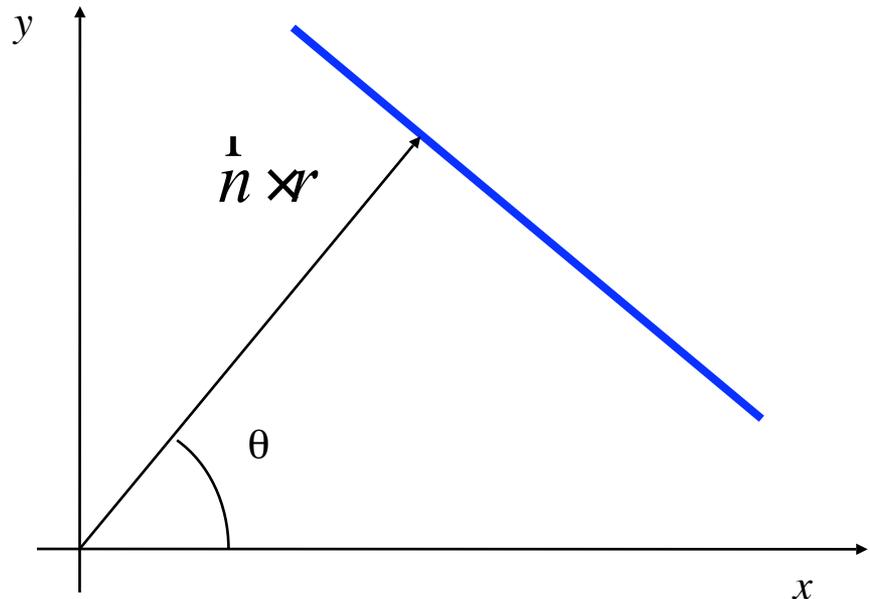
Robust bzgl. fehlender/zusätzlicher Edgels.

# Hough Transformation

Darstellung der Geraden als Hesse'sche Normalform

$$\overset{1}{n} \times \overset{1}{x} = r$$

$$x \cos \theta + y \sin \theta = r$$



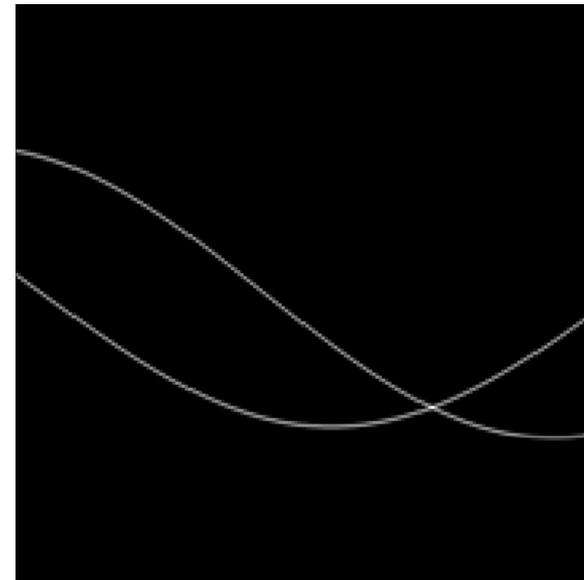
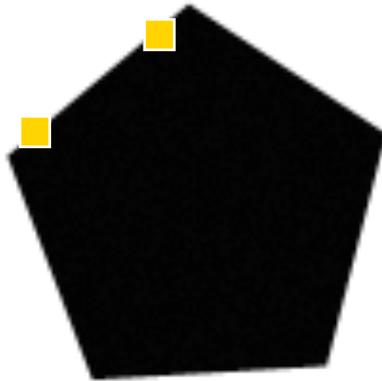
## Hough Transform: Straight Edge Extraction

---

- | All points  $p$  on a straight-line edge must satisfy  $y_p = m_1 x_p + b_1$ .
- | Each point  $(x_p, y_p)$  that is part of this line constraints the parameter  $m_1$  and  $b_1$ .
- | The Hough transform finds the line (line-parameters  $m, b$ ) that get most “votes” from the edge pixels in the image.
- | This is realized by four steps
  1. *Create a 2D array  $A [m,b]$  with axes that tessellate the values of  $m$  and  $b$ .*
  2. *Initialize the array  $A$  to zero.*
  3. *For each edge pixel  $(x_p, y_p)$  in the image, loop over all values of  $m$  and  $b$ :*  
*if  $y_p = m_1 x_p + b_1$  then  $A[m,b] += 1$*
  4. *Search cells in  $A$  with largest value. They correspond to extracted straight-line edge in the image.*

# Hough Transformation

Kurven  $\kappa_{(x,y)}$  von Punkten  $(x,y)$  auf  $r = x \cos \phi + y \sin \phi$  im  $x$ - $y$ -Raum schneiden sich im Punkt  $(r, \phi)$  im  $r$ - $\phi$ -Raum



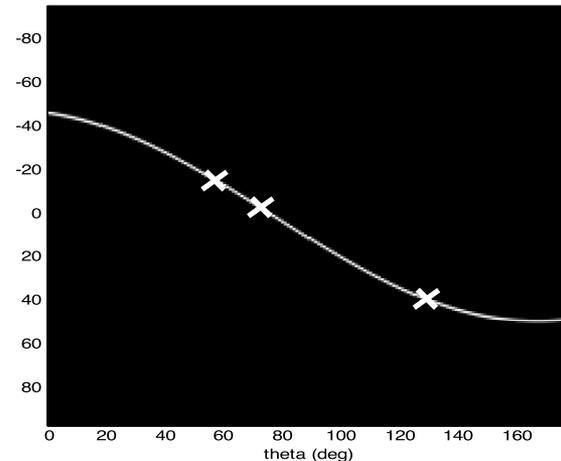
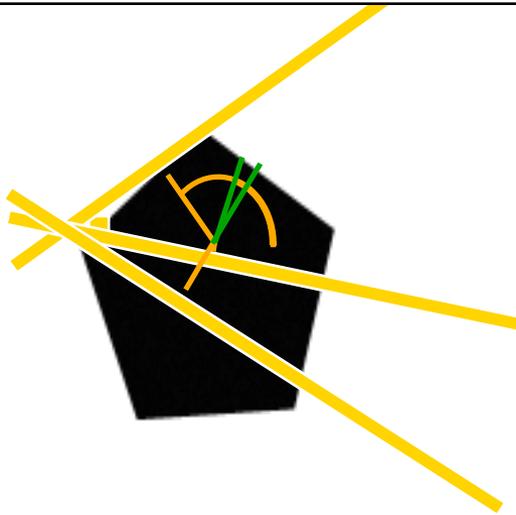
# Hough Transformation

Geraden  $r = x \cos \phi + y \sin \phi$  im  $x$ - $y$ -Raum

werden dargestellt als Punkte  $(r, \phi)$  im  $r$ - $\phi$ -Raum.

Geradenbündel durch  $(x,y)$  im  $x$ - $y$ -Raum

werden dargestellt als Kurven  $\kappa_{(x,y)}$  im  $r$ - $\phi$ -Raum



# Hough-Transformation

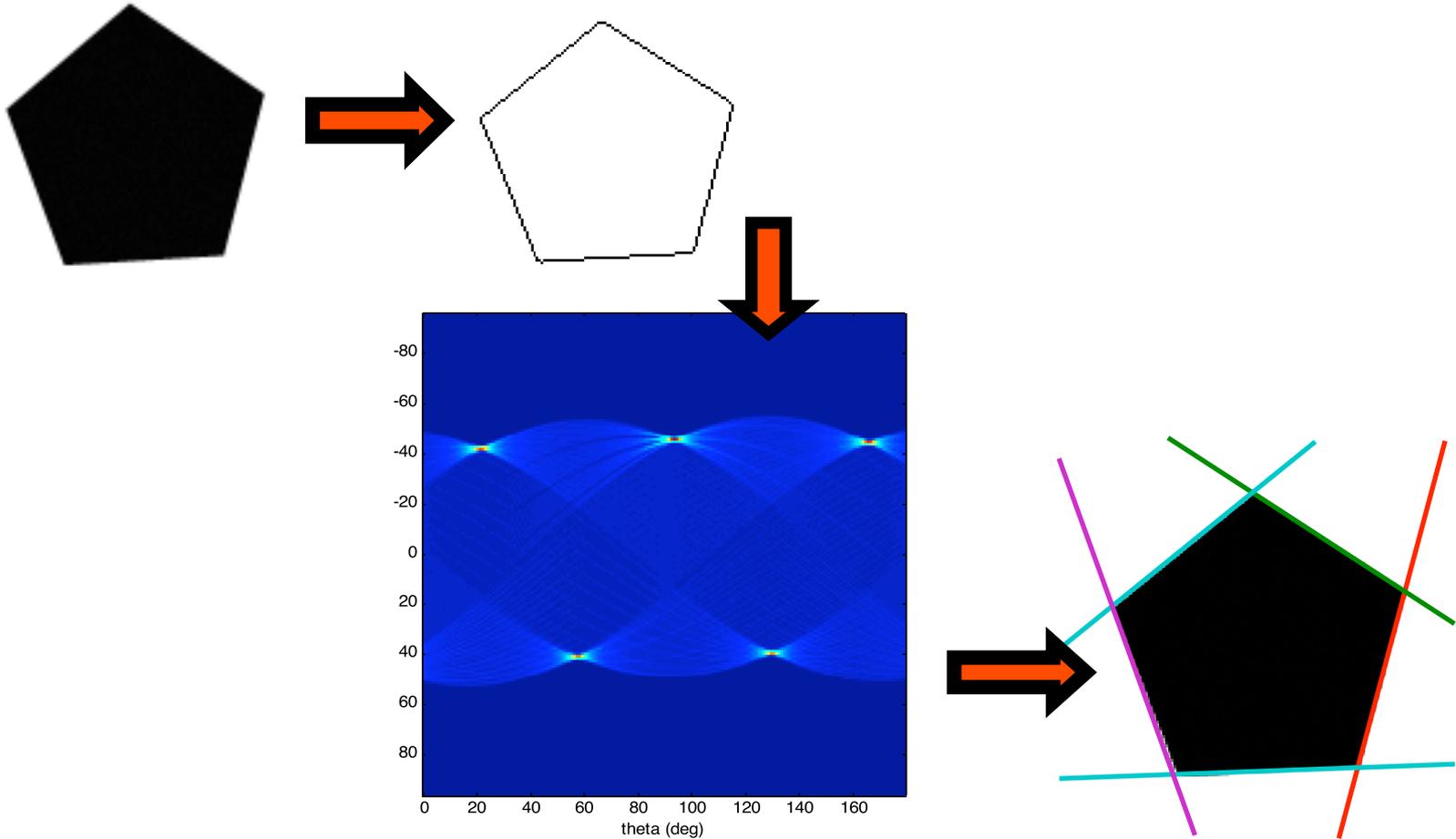
## Praktische Umsetzung:

- Akkumulator-Grid im  $r$ - $\phi$ -Raum füllen („Voting“)  
(Akkumulatorgrid beschränken entsprechend Bild)
- Für jedes Edgel entsprechende Einträge für Kandidaten (= Gerade  $(r, \phi)$ ) füllen.
- Geraden mit hinreichend vielen Stimmen auswählen

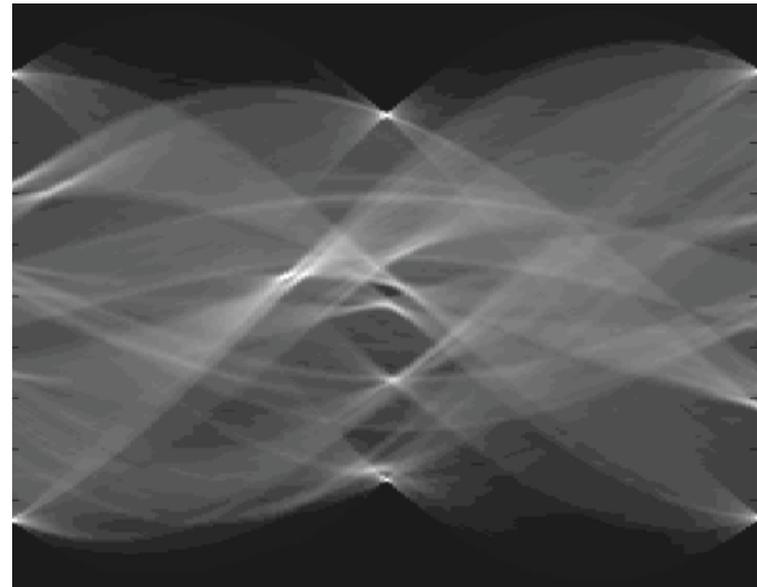
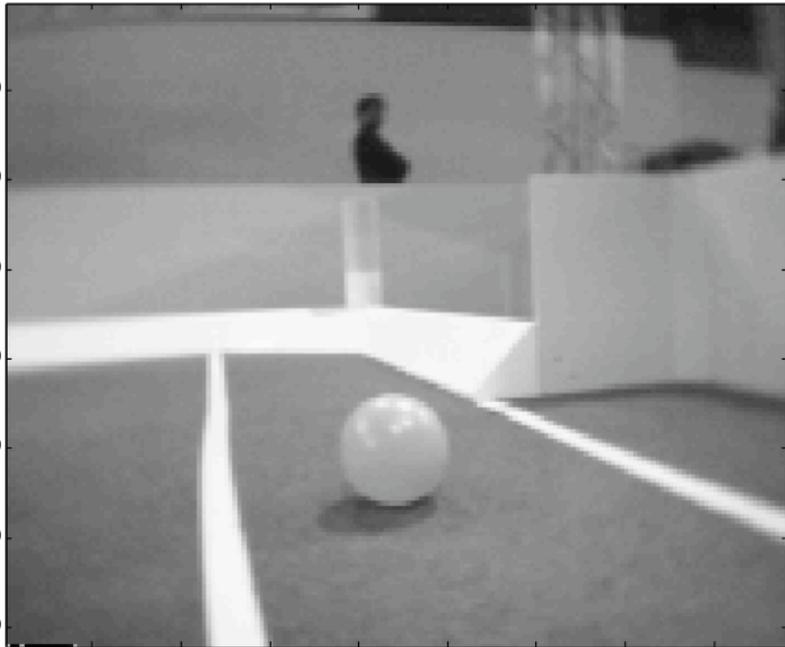
## Schnelle Hough-Transformation:

- Richtungsinformation von Edgels ausnutzen
- Nur 1 Eintrag im Parameterraum (bzw. bei grober Orientierung einige Einträge)

# Hough Transformation



# Hough Transformation



# Hough-Transformation

Erweiterbar für andere Konturen (Kreise, ...)

Kreise:

- 3-dimensionales Grid für Kreisparameter  $a$ ,  $b$ ,  $r$

$$(x-a)^2 + (y-b)^2 = r^2$$

- Rechnungen entsprechend aufwändiger

# Bildbereiche (Regionen)

Merkmale:

- Helligkeit, Farbe, Textur
- Begrenzung

Merkmale bestimmen:

- Histogramme
- Farbklassifizierung
- Texturanalyse
- Kombination von Kanälen

Erkennung

- Segmentierung anhand von Merkmalen
- Suchverfahren: Gebiete erweitern/einschränken

# Segmentierung

Erzeugung eines Binärbildes:

Klassifikation bzgl. Zugehörigkeit zu einer Region

Anschließend z.B.

- Bestimmung von Kanten
- Identifikation anhand Gestalt

# Punktbasierte Segmentierung

Gegeben:

Unterscheidungsmerkmal

als ein Grauwert-Bild bzgl. Ausprägung des Merkmals

Gesucht:

Klassifizierung anhand des Merkmals, d.h.

Zugehörigkeit zu Klassen (Äquidensiten bzw. Binärbild)

z.B. Abgrenzung von Regionen (z.B. Objekt, Hintergrund)

Benötigt Schwellwerte zur Klassifizierung:

Bestimmung mittels Histogrammen

# Punktbasierte Segmentierung

Auswirkungen unterschiedlicher Schwellwerte:

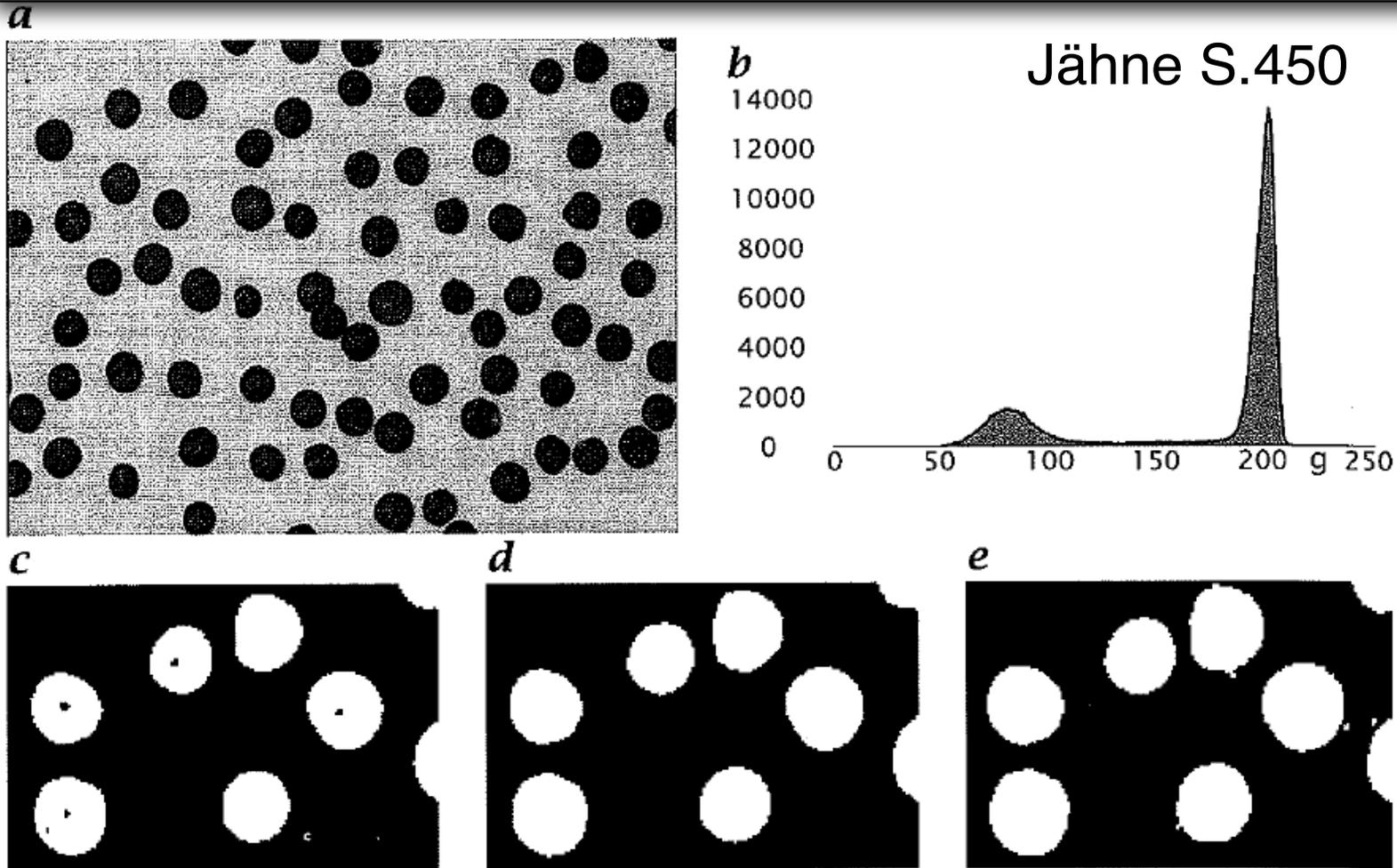
- Störungen wirken sich aus
- Unterschiedliche Größe der Regionen  
i.a. nicht vermeidbar wegen Effekten der Quantisierung

Auswirkungen bei ungleichmäßigem Hintergrund  
(unterschiedliche Beleuchtung):

- keine einheitliche Größe

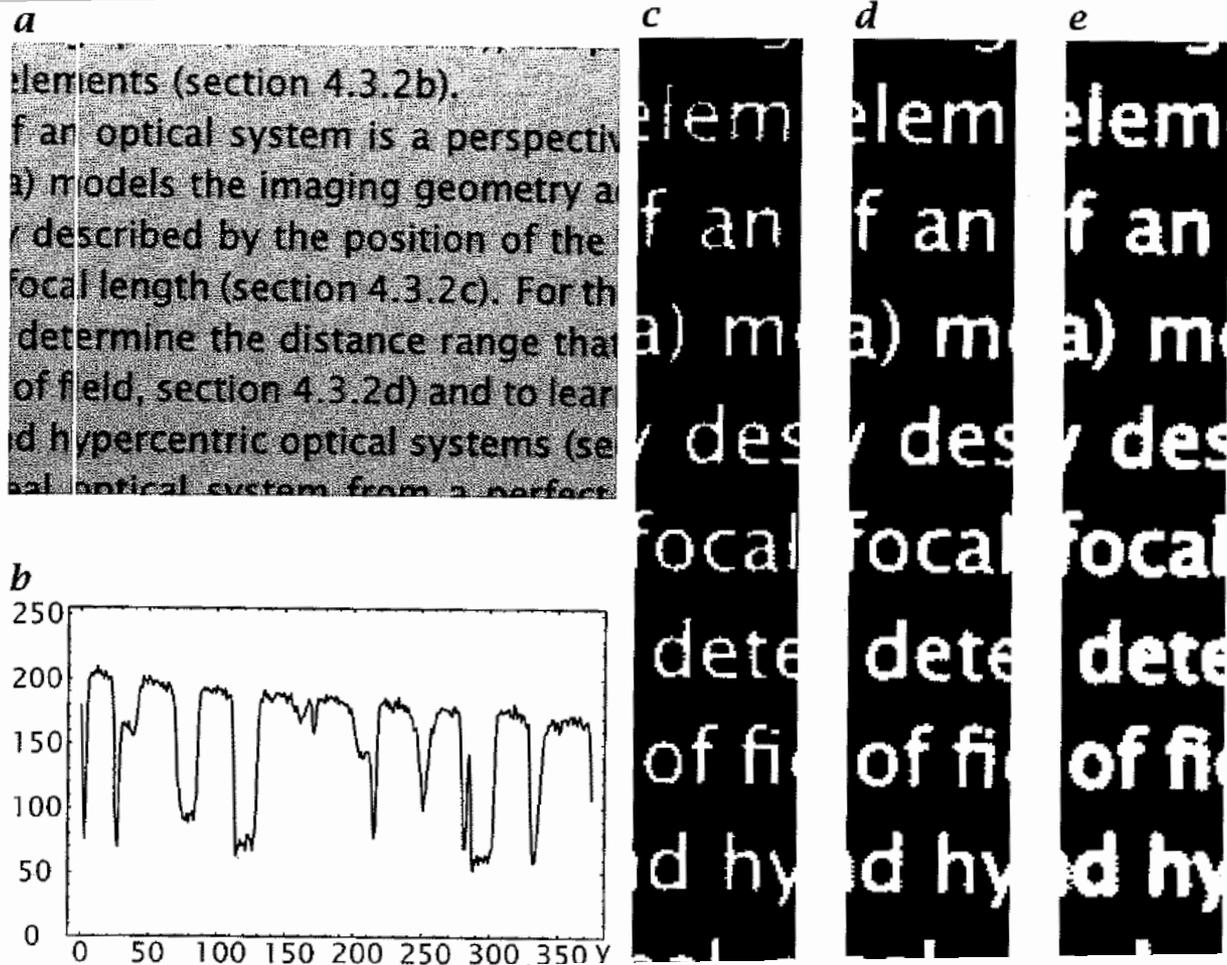
Ausgleich durch vorherige Korrektur für gleichmäßige  
Helligkeit

# Punktbasierte Segmentierung



**Abbildung 16.1:** Segmentierung mit einem globalen Schwellwert: *a* Originalbild; *b* Histogramm; *c* bis *e* oberer rechter Sektor von *a*, segmentiert mit globalen Schwellwerten von 110, 147 bzw. 185.

# Punktbasierte Segmentierung

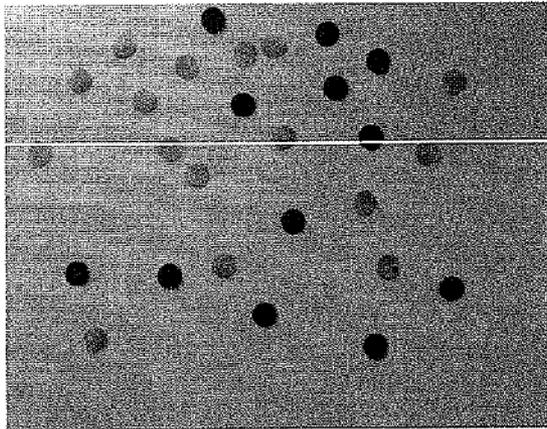


Jähne S.451

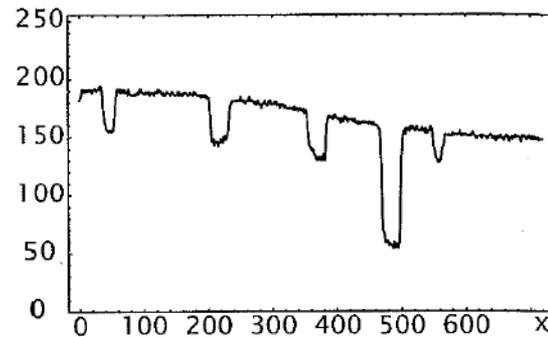
**Abbildung 16.2:** Segmentierung eines Bildes mit ungleichmäßigem Hintergrund: *a* Originalbild; *b* Profil der Spalte 55 (markiert in *a*); *c* bis *e* die ersten 64 Spalten aus *a* nach Segmentierung mit einem globalen Schwellwert von 90, 120 bzw. 150.

# Punktbasierte Segmentierung

*a*

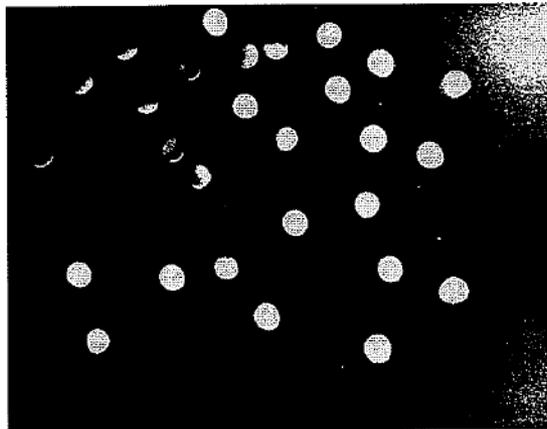


*b*

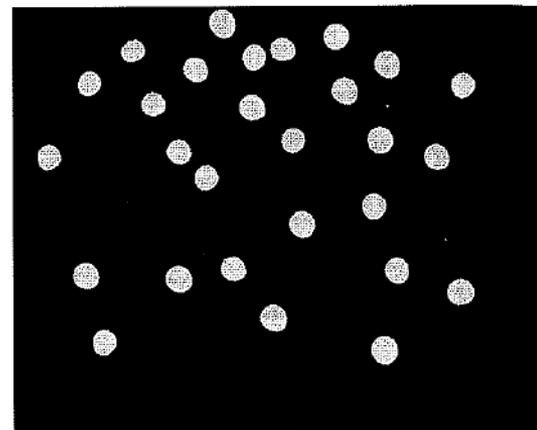


Jähne S.452

*c*



*d*



**Abbildung 16.3:** Segmentierung eines Bildes mit ungleichmäßiger Hintergrundbeleuchtung: *a* Originalbild (Histogramm dazu siehe Abb. 10.10b); *b* Profil der Zeile 186 (markiert in *a*); *c* Segmentierung des Bildes in *a* mit einem optimalen globalen Schwellwert; *d* Segmentierungsergebnis, nachdem das Bild zuerst auf den ungleichmäßigen Hintergrund korrigiert wurde (Abb. 10.10c).

# Regionen-Wachstum

Ausgehend von einem Pixel werden sukzessive „gleichartige“ Nachbarn hinzugenommen.

Implementierung als

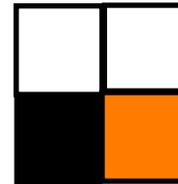
Tiefe-Zuerst oder Breite-Zuerst Suche

- Verwaltung der zu untersuchenden Nachbarn durch entsprechende OPEN-Liste

# Split-and-merge

Idee analog Quadtree-Konstruktion:

„Split“ als fortlaufende Zerlegung von jeweils quadratischen Bereichen, die nicht insgesamt zu einer Klasse gehören.



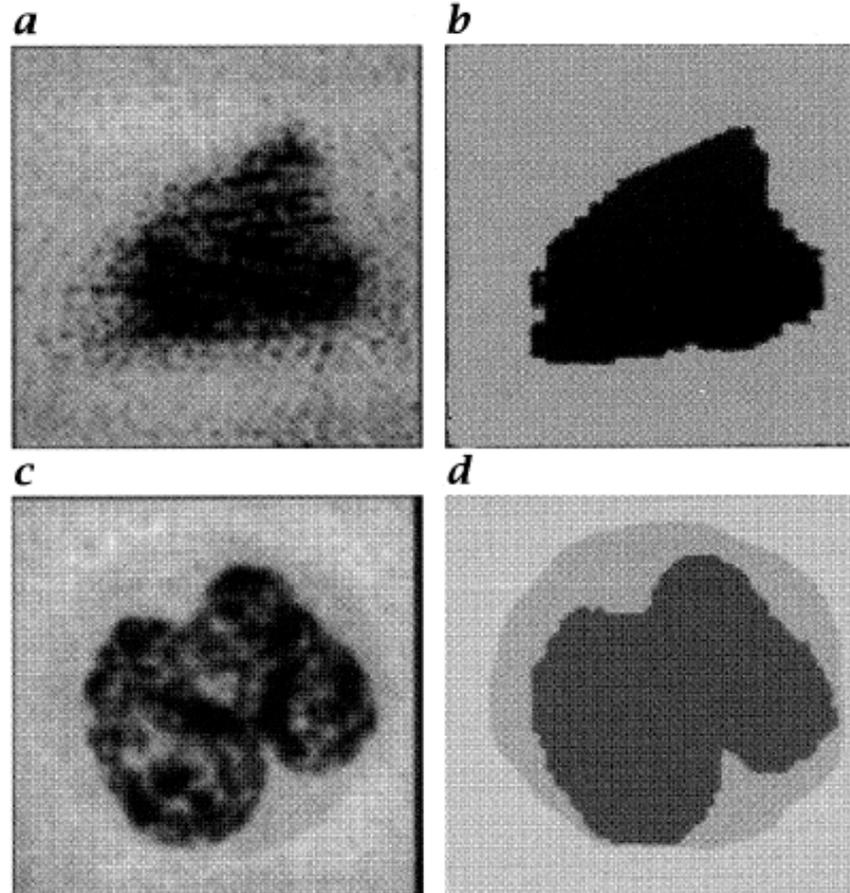
„Merge“ als jeweils anschließende Zusammenfassung gleicher Bereiche.



# Pyramid Linking

Verfahren zum Ausgleich von Störungen

JÄHNE S. 458

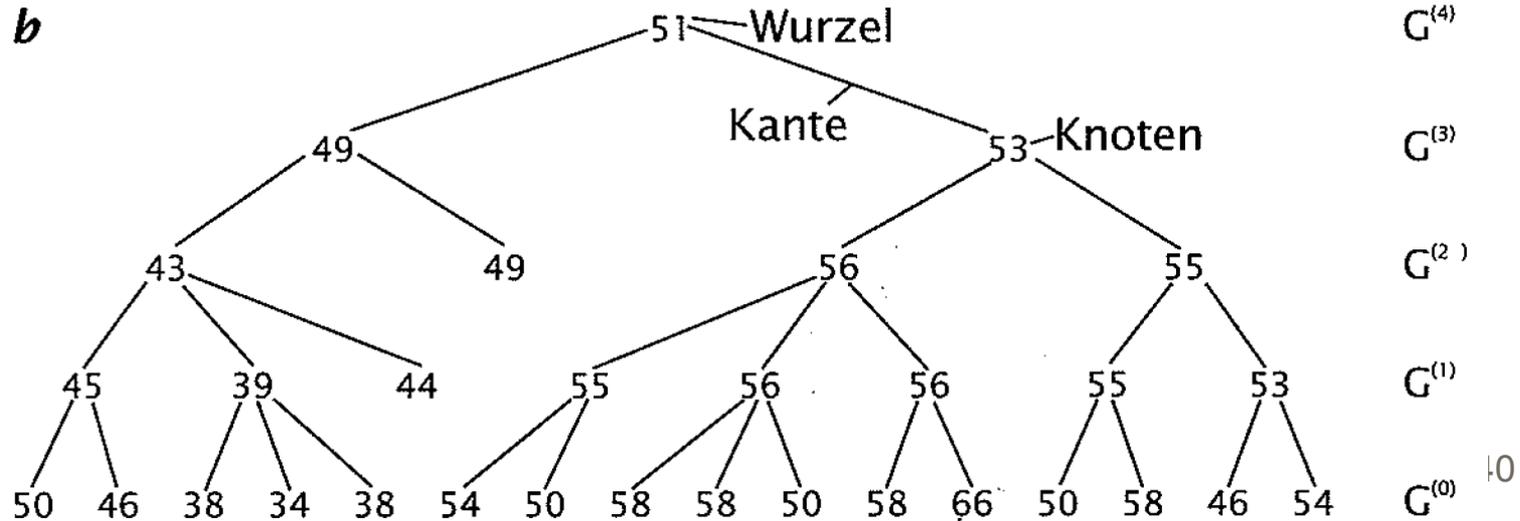
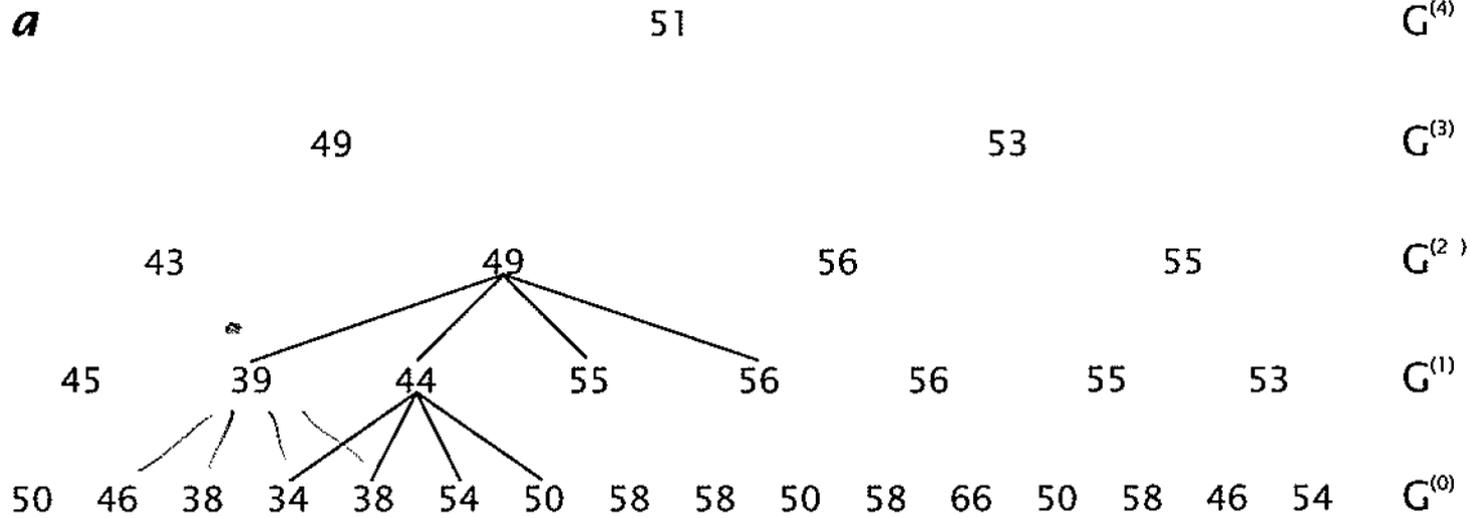


# Pyramid Linking

- a) Gauß-Pyramide: jeweils Mittelwert für 4 Pixel wird Vaterknoten in höherer Ebene.
  - b) Pyramid-Linking: Nur Kanten zu dem Vaterknoten mit nächstliegendem Grauwert.
  - c) Neuberechnung der Mittelwerte entlang der noch existierenden Kanten.
  - b) und c) wiederholen bis Stabilität erreicht.
  - d) Bei Stabilität werden Ergebnisse einer auszuwählenden Schicht nach unten propagiert.
- Unterste Ebene gibt korrigierte Grauwerte an.

# Pyramid Linking: 1D-Beispiel

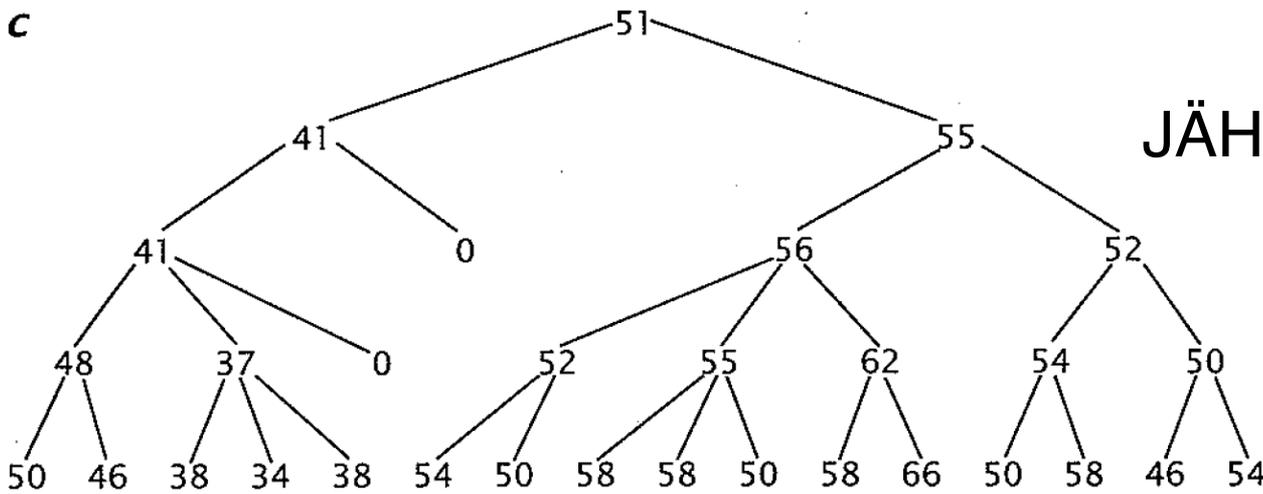
JÄHNE S. 457



c

$G^{(4)}$

JÄHNE S. 457



$G^{(2)}$

$G^{(1)}$

$G^{(0)}$



d

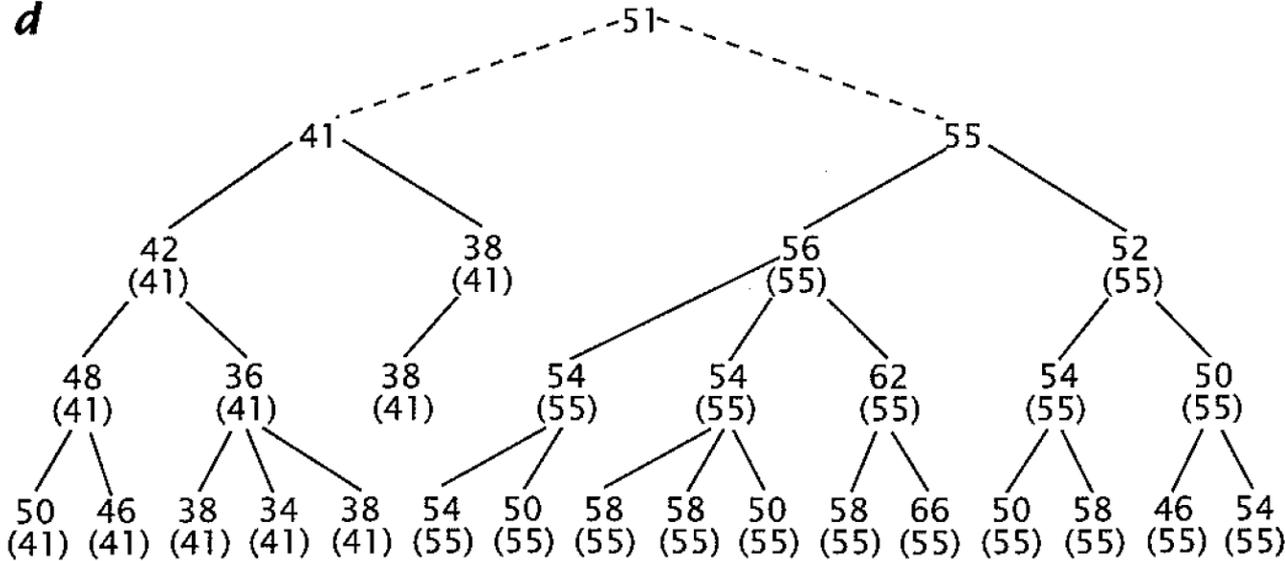
$G^{(4)}$

$G^{(3)}$

$G^{(2)}$

$G^{(1)}$

$G^{(0)}$



Ergebnis ist eine Stufenkante

# Objektklassifizierung

- Vergleichsoperationen:
  - Ähnlichkeit zwischen Bildobjekt und Referenzobjekt
    - Lokaler Vergleich (Pixel), Vergleich lokaler Merkmale (Beziehung von Pixeln: Punkte, Kanten, Farben, Texturen, ...)
    - Struktureller Vergleich (Beziehung von Merkmalen)
    - Vergleich globaler Merkmale (Histogramme usw.)
- Klassifizierung in komprimierten Darstellungen
  - RLE, Nachbarschaftsgrafan
  - Quad-Trees
  - Richtungsketten, . . .

# Lokale Ähnlichkeiten

- Differenzbild (einfaches “Fehlerbild”)
- Korrelation (Summe quadratischer Fehler minimieren)
- Filterung mit „Suchmaske“ (= Referenzobjekt)

Direkter Vergleich auf Pixelebene erfordert

- Angleichung von Position und Größe
  - Translation
  - Rotation
  - Skalierung
- Angleichung der Darstellung
  - Grauwerte/Farben
  - Ergänzung (verdeckter/unsichtbarer Teile)
  - Abstraktion (Verzicht auf individuelle Eigenschaften)
  - Rauschunterdrückung

# Ähnlichkeit struktureller Merkmale

Beziehungen zwischen Merkmalen (Graph) vergleichen  
Quantitativ oder qualitativ

- Merkmale: Bildprimitive, Komponenten:  
Kanten, Ecken, Oberflächen, komplexere Merkmale
- Struktur:  
Beziehungen von Bildprimitiven/Komponenten  
z.B. Constraint-Verfahren von Waltz anwenden

# Ähnlichkeit globaler Merkmale

Statistische Merkmale (Bilder als Zufallsvariable)

- Mittelwert (durchschnittlicher Grauwert)
- Varianz (Kontrast)
- Kovarianz bei mehrkanaligen Bildern
- Entropie

Histogramme

# Ähnlichkeit: strukturelle Merkmale

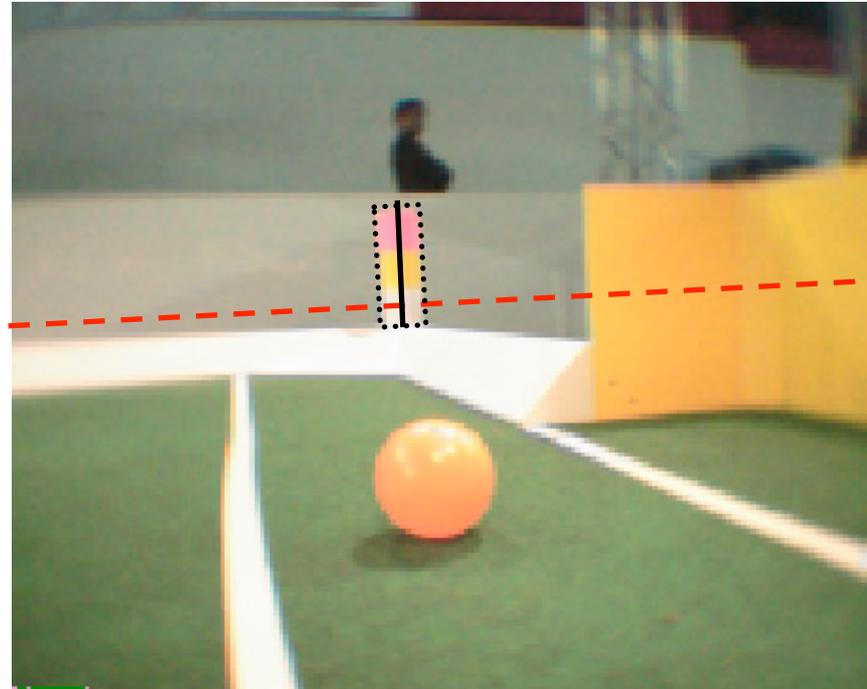
Beispiel:

Erkennung der Flaggen

Bekannte Merkmale:

Kanten: weiß-gelb  
und gelb-pink

Lage im Bild relativ  
zum Horizont



Außenkanten als Maß für die Größe einer Landmarke

# Matching: Korrelation

Abstand zwischen Template  $h(u,v)$  und Ausschnitt  $g(x+u,y+v)$  des Bildes an der Stelle  $(x,y)$  berechenbar als Summe der Quadrate (analog Euklidischer Abstand):

$d^2(x,y)$

$$\begin{aligned} &:= \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} ( g(x+u,y+v) - h(u,v) )^2 \\ &= \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} ( g(x+u,y+v)^2 + h(u,v)^2 - 2 g(x+u,y+v) \cdot h(u,v) ) \\ &= \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x+u,y+v)^2 \quad ( \Rightarrow a(x,y) ) \\ &\quad + \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} h(u,v)^2 \quad ( \Rightarrow b, \text{konstanter Wert} ) \\ &\quad - 2 \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x+u,y+v) \cdot h(u,v) \quad ( \Rightarrow -2 \text{ corr}(x,y) ) \end{aligned}$$

# Matching: Korrelation

Falls  $a(x,y)$  konstant für Bild, dann wird  $d(x,y)$  minimiert, wenn die Korrelation

$$\text{corr} = \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} g(x+u,y+v) \cdot h(u,v)$$

maximiert wird.

Falls  $a(x,y)$  nicht konstant:

Normierung bzgl. lokaler Helligkeit: z.B. Division  $( a(x,y) \cdot b )^{1/2}$

Korrelation ist ein Maß für Ähnlichkeit zwischen Bildausschnitt und Template. Aber: Wert abhängig von Gesamt-Helligkeit.

Korrelation (als gespiegelte Faltung, beachte Vorzeichen) im Frequenzraum mittels Fouriertransformierten berechenbar.

# Matching: Korrelationskoeffizient

Vergleich bezogen auf Mittelwerte

$M(g,x,y)$  (durchschnittliche Helligkeit des Ausschnitts um  $(x,y)$  )

und  $M(h)$  (durchschnittliche Helligkeit von  $h$  ) :

$$\text{COV}_{(g,x,y) h} = \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} ( g(x+u,y+v) - M(g,x,y) ) \cdot ( h(u,v) - M(h) )$$

und bei Normierung mit Varianzen

$$\sigma_{g,x,y} = \left( \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} ( g(x+u,y+v) - M(g,x,y) )^2 \right)^{1/2}$$

$$\sigma_h = \left( \sum_{u=-k\dots+k} \sum_{v=-l\dots+l} ( h(u,v) - M(h) )^2 \right)^{1/2}$$

ergibt als Helligkeitsunabhängiges Ähnlichkeitsmaß:

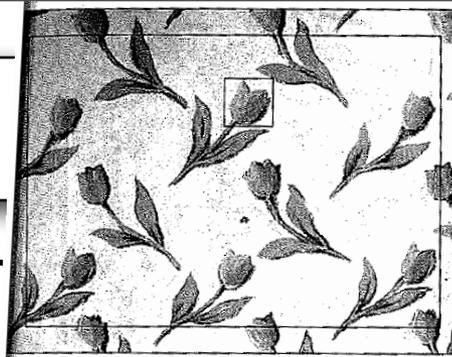
Korrelationskoeffizient  $\text{COV}_{(g,x,y) h} / (\sigma_{g,x,y} \cdot \sigma_h)$

# Matching

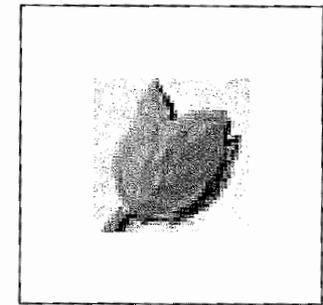
Ergebnisse unterschiedlicher  
Ähnlichkeitsmaße.

Die Helligkeit gibt den Grad  
Der Übereinstimmung an  
für die jeweilige Position  
des Templates.

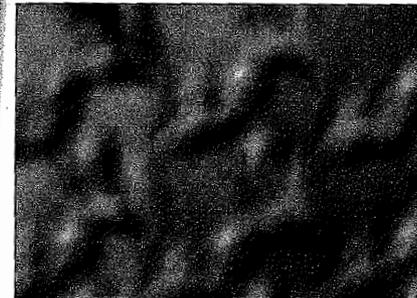
Bild Burger/Burge, S. 431



(a) Originalbild  $I$



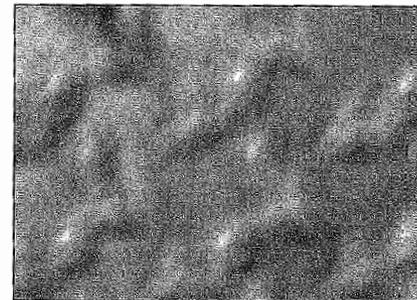
(b) Template  $R$



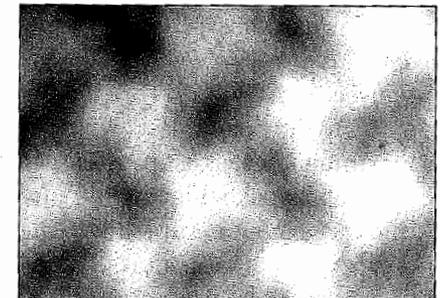
(c) Summe der Differenzbeträge



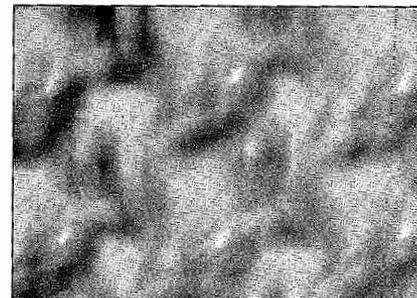
(d) Maximaler Differenzbetrag



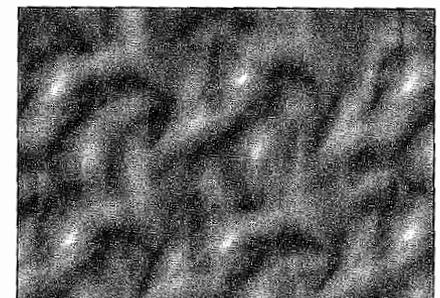
(e) Summe der quadratischen Abstände



(f) Globale Kreuzkorrelation



(g) Normalisierte Kreuzkorrelation



(h) Korrelationskoeffizient

# Rotation, Translation, Skalierung

Vergleich mit Template mit dem Bild

- an verschiedenen Stellen
- in verschiedener Orientierung
- in verschiedenen Größen

(aufwändig)



# SIFT = Scale Invariant Feature Transformation

Verfahren zur Bestimmung von Merkmalen, die unabhängig von Position, Rotation und Skalierung erkennbar sind.

Robustheit in gewissen Grenzen auch bzgl. affiner Transformation, Standpunktwechsel und Beleuchtungsunterschieden

David G. Lowe:

*Distinctive Image Features from Scale-Invariant Keypoints.*  
International Journal of Computer Vision, 2004.

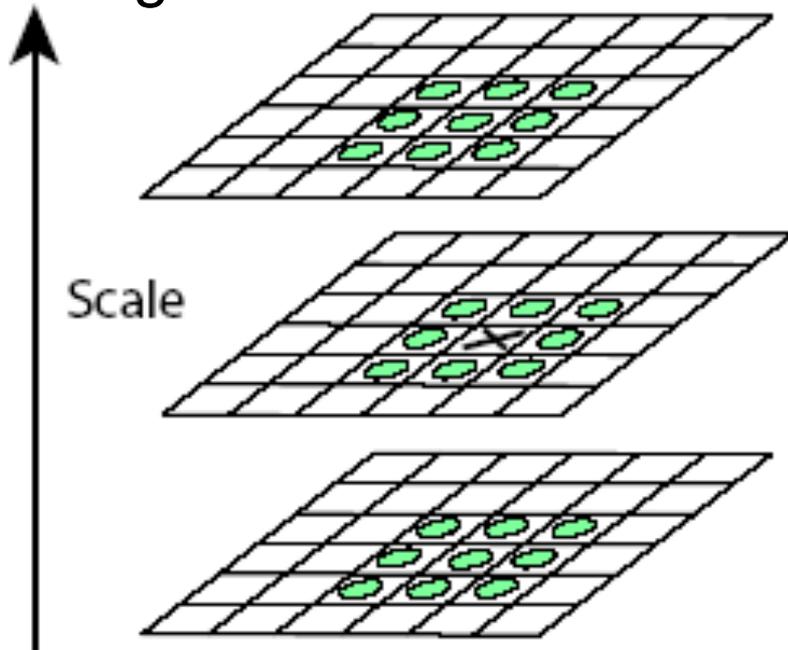
4 Schritte.

Ca. 2000 Features werden für 500x500 Pixel erzeugt.

# SIFT = Scale Invariant Feature Transformation

## 1. Kandidaten suchen:

Extremwerte in einer Laplace-Pyramide bestimmen:  
Vergleich jedes Punktes mit 26 lokalen Nachbarn in der eigenen Ebene und in den beiden Nachbar-Ebenen.



Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

# SIFT = Scale Invariant Feature Transformation

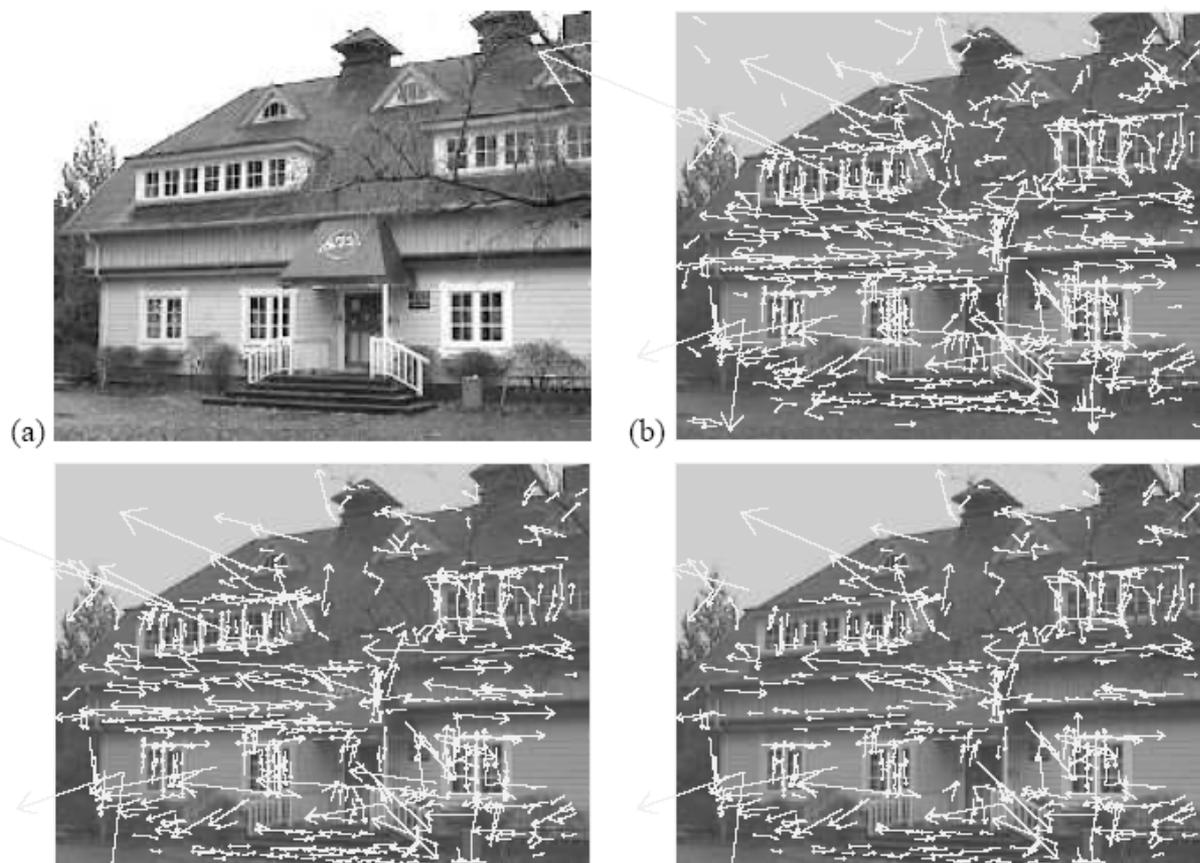
## 2. Auswahl und Lokalisierung von Keypoints:

Elimination von Kandidaten mit geringem Kontrast oder unklarer Position (auf einer Kante).

Zu jedem Keypoint gehört danach

- seine Skalen-Ebene und
- seine Position in dieser Ebene

# SIFT



This figure shows the stages of keypoint selection.

(a) The 233x189 pixel original image.

(b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location.

(c) After applying a threshold on minimum contrast, 729 keypoints remain.

(d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

# SIFT = Scale Invariant Feature Transformation

## 3. Orientierung der Keypoints bestimmen:

- Gradient (Richtung, Betrag) für alle Bildpunkte einer Nachbarschaft des Keypoints bestimmen
- Histogramme für gewichtete Richtungen (Gewicht für jeden Punkt proportional zu Betrag und Entfernung vom Keypoint)
- Orientierung entspricht Maximum im Histogramm
- Falls mehrere Kandidaten für Maximum:

Zusätzliche Keypoints (besonders gut für Matching)

Alle weiteren Eigenschaften werden relativ zu Skalen-Ebene, Position und Orientierung des Keypoints bestimmt (ergibt Invarianz bzgl. Transformationen)

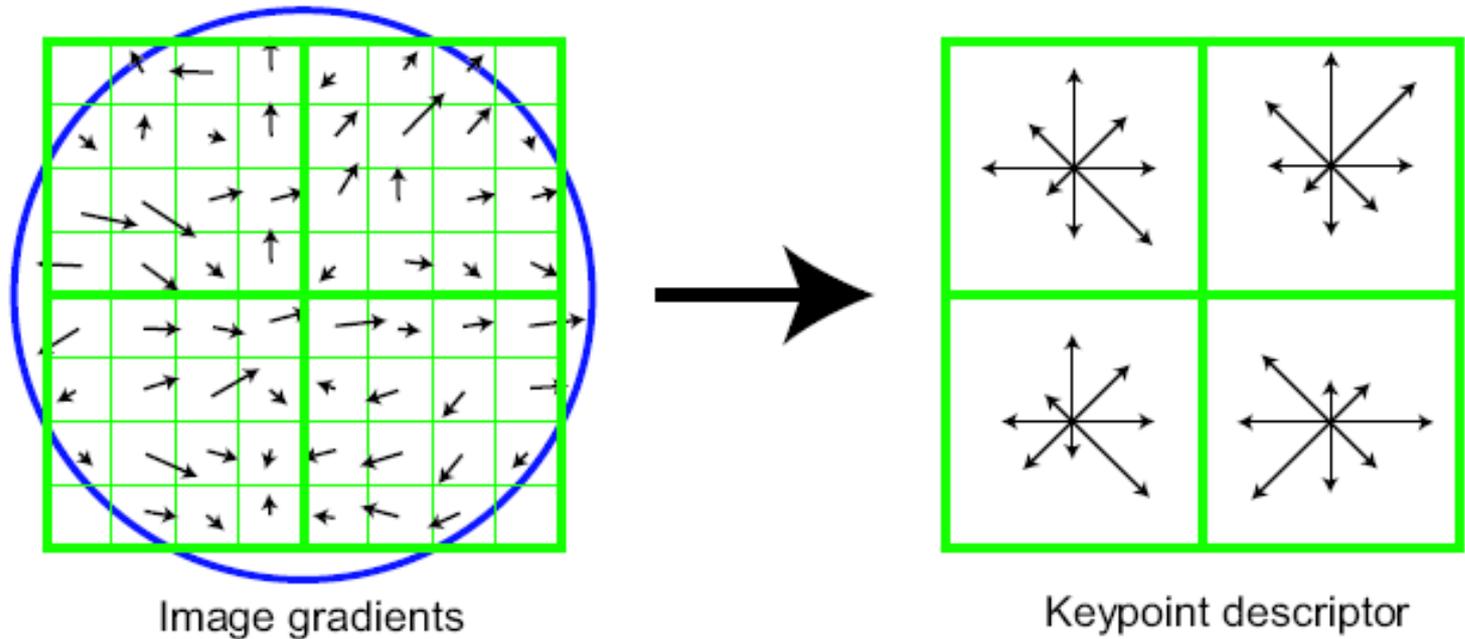
# SIFT = Scale Invariant Feature Transformation

## 4. Keypoint Descriptor:

Lokale Gradienten in einer Umgebung der Keypoints  
(relativ zu den Daten aus Schritt 2 und 3).

Normalisierung zum Ausgleich von Unterschieden bzgl.  
Kontrast und Helligkeit.

# SIFT



A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left.

These are weighted by a Gaussian window, indicated by the overlaid circle.

These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region.

This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

# SIFT = Scale Invariant Feature Transformation

Anwendung:

Vergleich von Keypoints eines Templates mit Keypoints eines Bildes

Aufwändiger Prozess des Vergleichens und Auswählens von Kandidaten für Matching von Keypoints (siehe Literatur).

# SIFT = Scale Invariant Feature Transformation



The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle (*600x480 Pixel*) . The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

# SIFT = Scale Invariant Feature Transformation



This example shows location recognition within a complex scene.

The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right.

The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.

# Bewegungen

Zeit als weitere Dimension

Bewegungen in der Szene erkennen

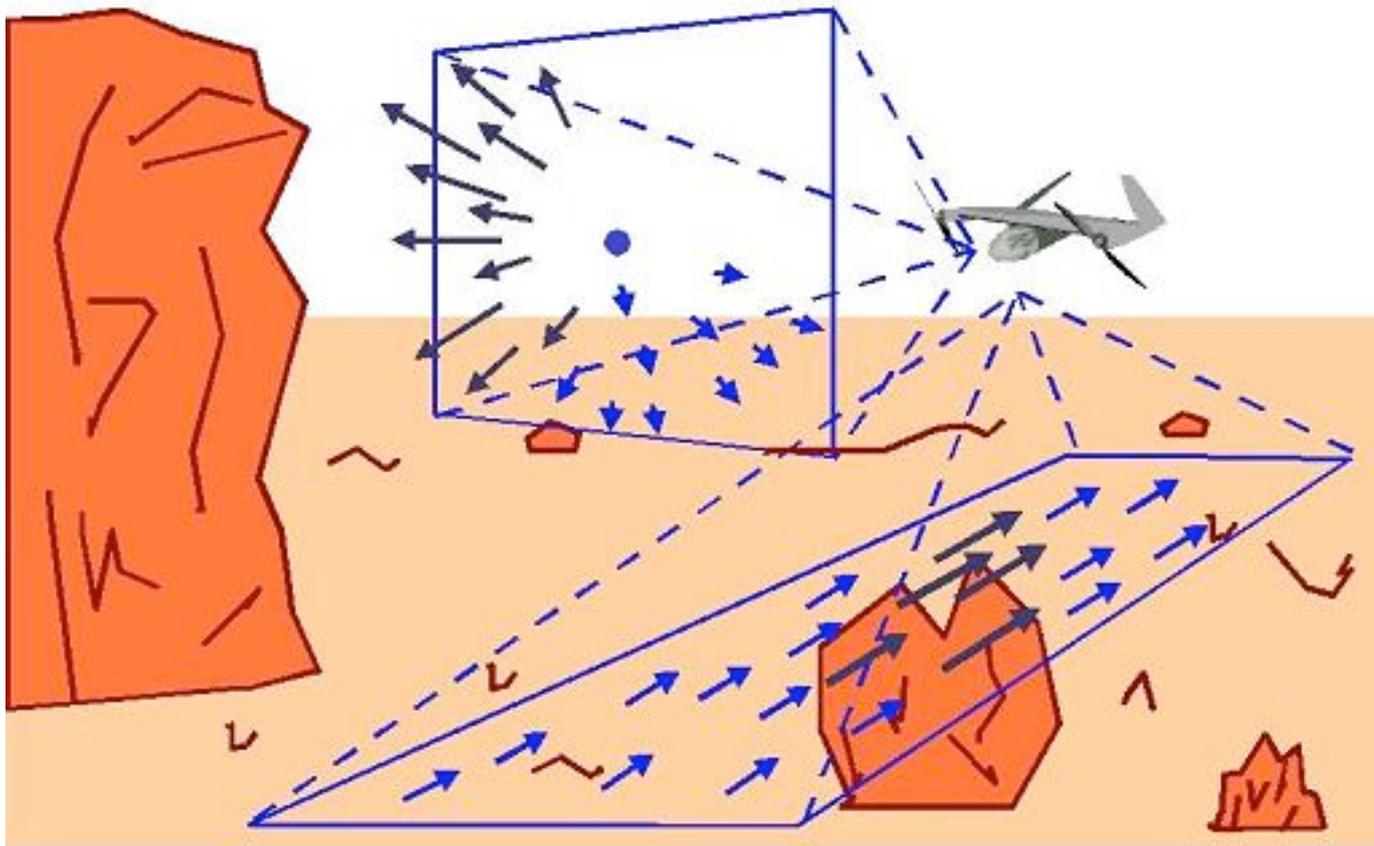
Eigene Bewegung als Veränderung

Überlagerung: Bewegung in der Szene und eigene Bewegung

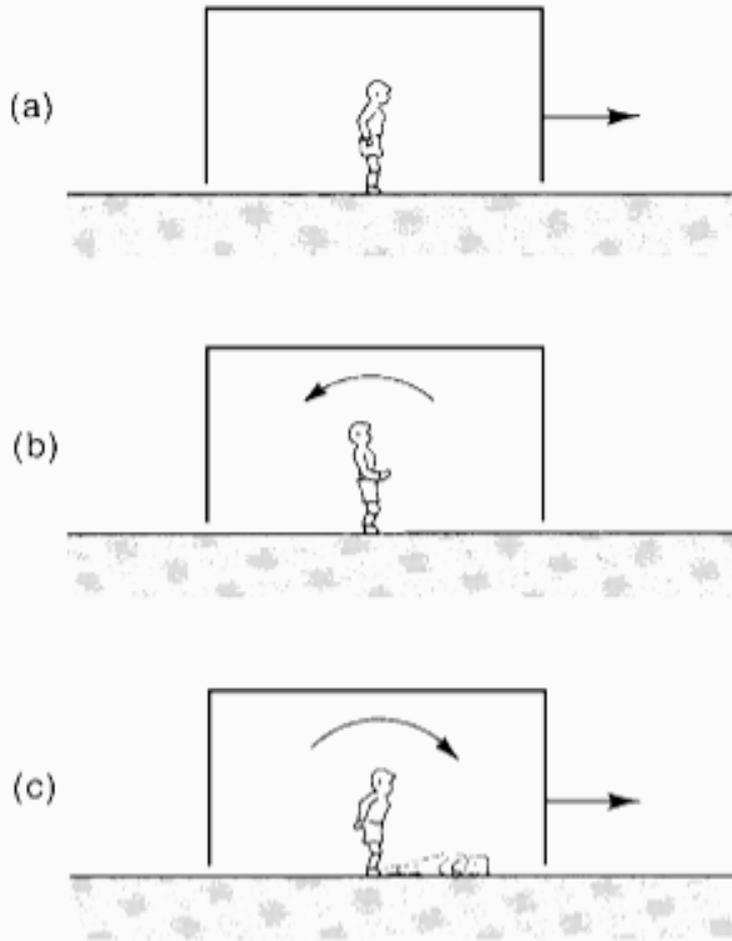
Auswirkungen:

- Bei langer Belichtung innerhalb eines Bildes
- Sonst: Unterschiede in aufeinander folgenden Bildern.

# Optischer Fluss



# Optischer Fluß und Orientierung

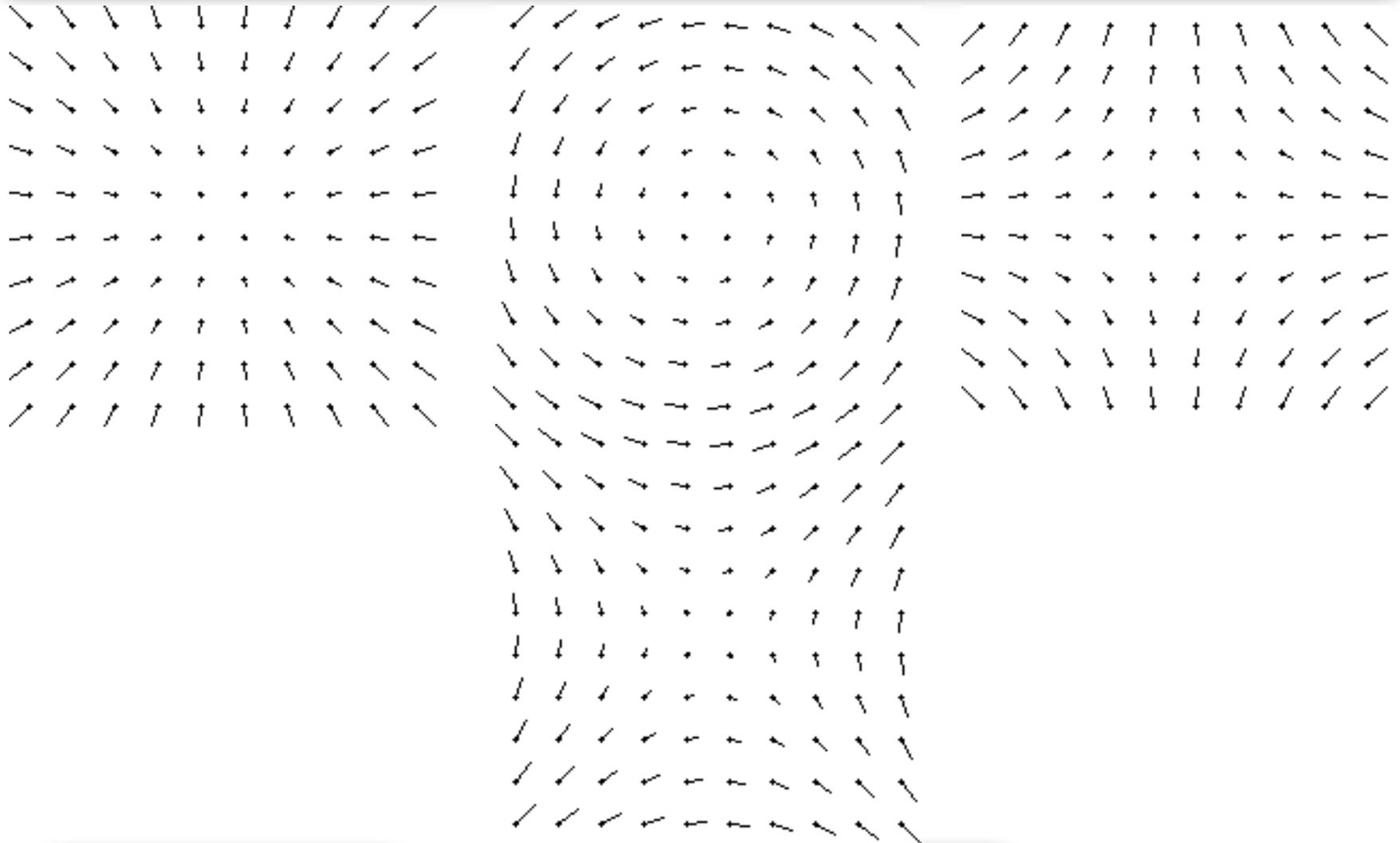


Der Schaukelraum von Lee und Aronson.

(a) Eine Vorwärtsbewegung des Raumes erzeugt dasselbe Muster des optischen Fließens, wie wenn eine Person sich nach hinten lehnt wie in (b).

Zum Ausgleich dieser Bewegungswahrnehmung lehnen sich Probanden nach vorne wie in (c) und verlieren dabei häufig das Gleichgewicht bis hin zum Umfallen.

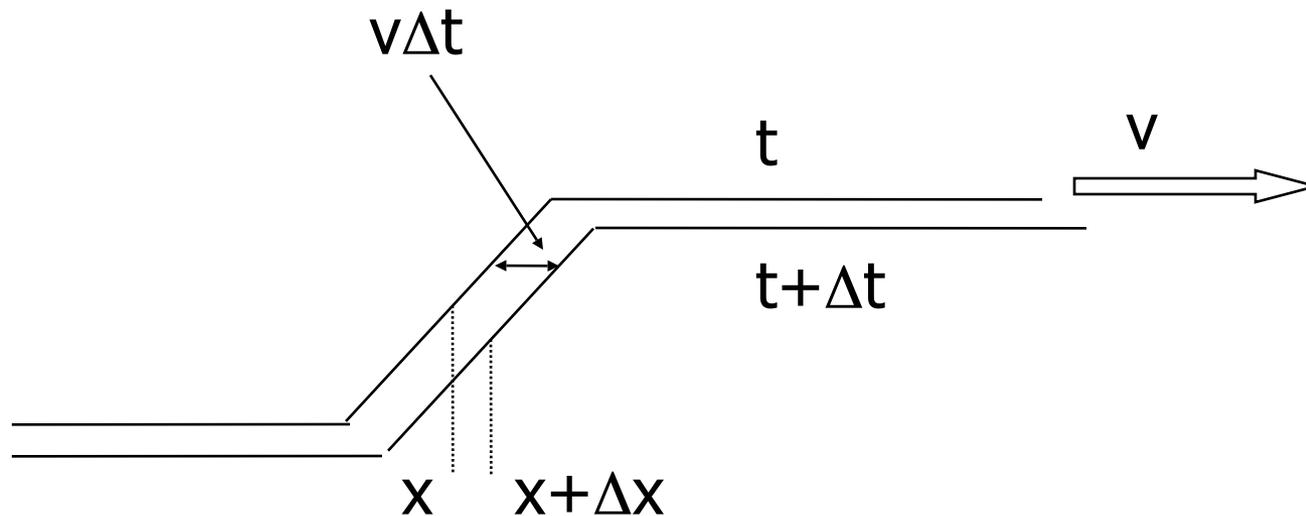
# Optischer Fluss



# Optischer Fluss

- In gradient-based methods, the greylevel profile is assumed to be locally linear which can be matched in the image pair
- Gradient-based methods assume a linear-ramped greylevel edge is displaced by  $v\Delta t$  between 2 successive frame

# Optischer Fluss



# TTC (time to contact)

## Tauchende Vögel

unterschiedliche Geschwindigkeit  
und Höhe

TTC konstant



## Was ist die TTC?

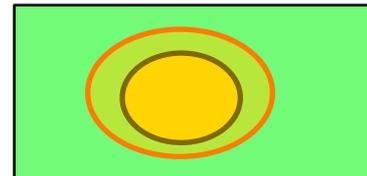
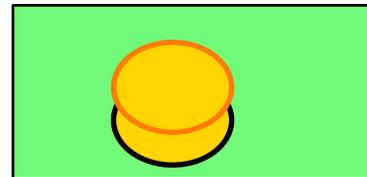
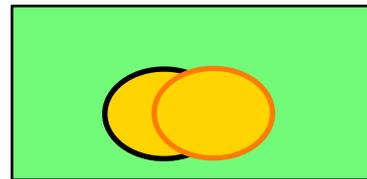
- TTC = size / change in size over time

$$\tau(x) = x / \dot{x}$$

# Bewegungen

Methoden:

- Differenzbilder
- Interpolation über mehrere Bilder
- Berechnungen im Raum-Modell



nach rechts



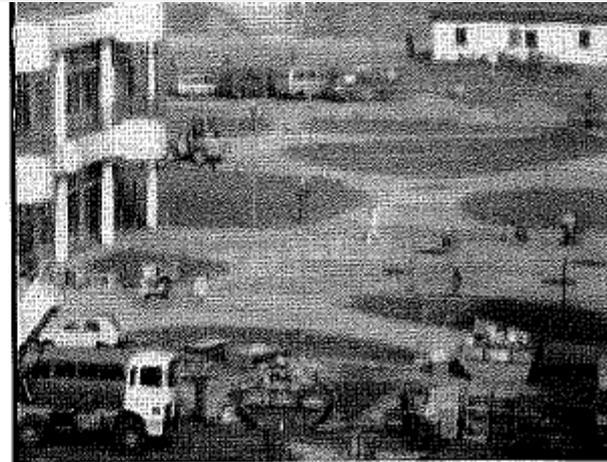
nach  
oben



nach  
vorn



# Differenzbilder



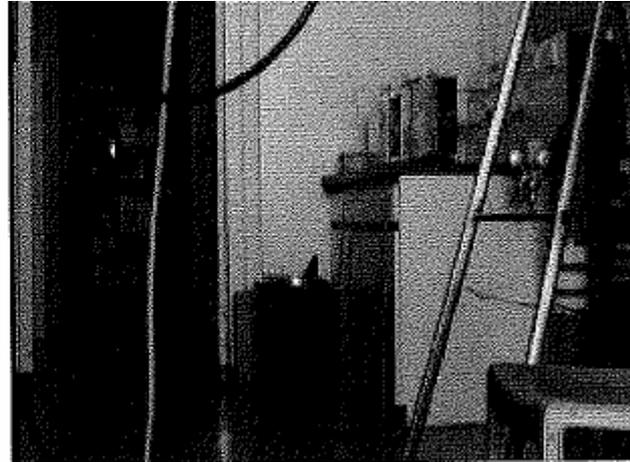
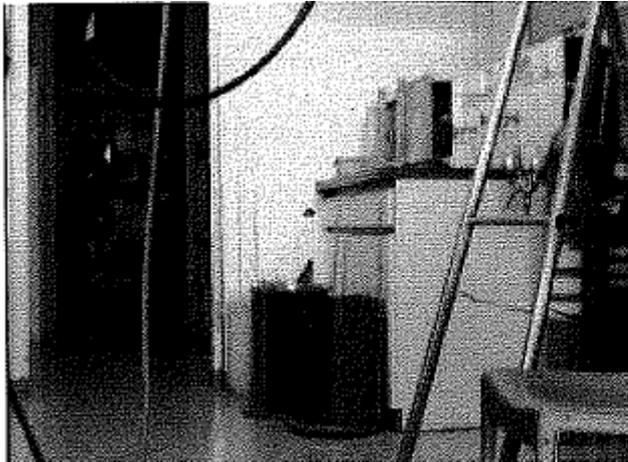
Jähne S. 396-398

# Differenzbilder



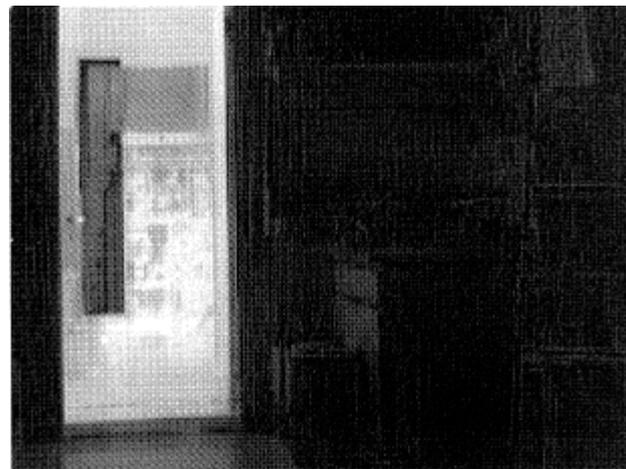
Jähne S. 396-398

# Differenzbilder



Jähne S. 396-398

# Differenzbilder

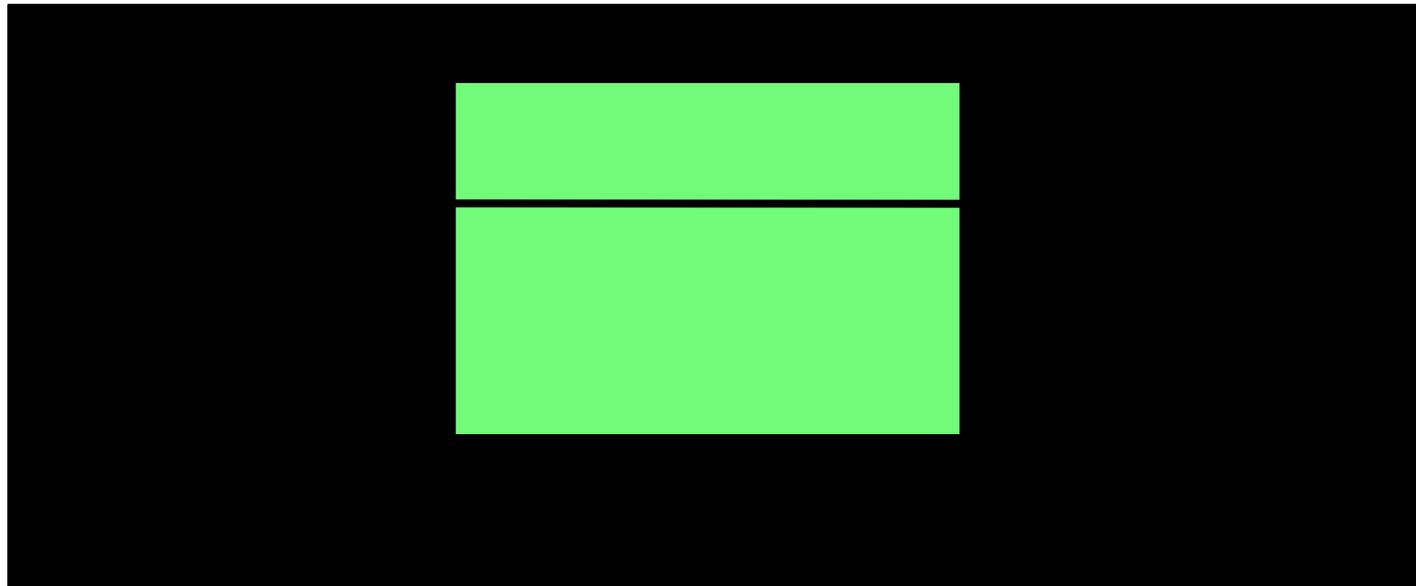


Jähne S. 396-398

# Bewegungen

Blendenproblem (Aperturproblem):

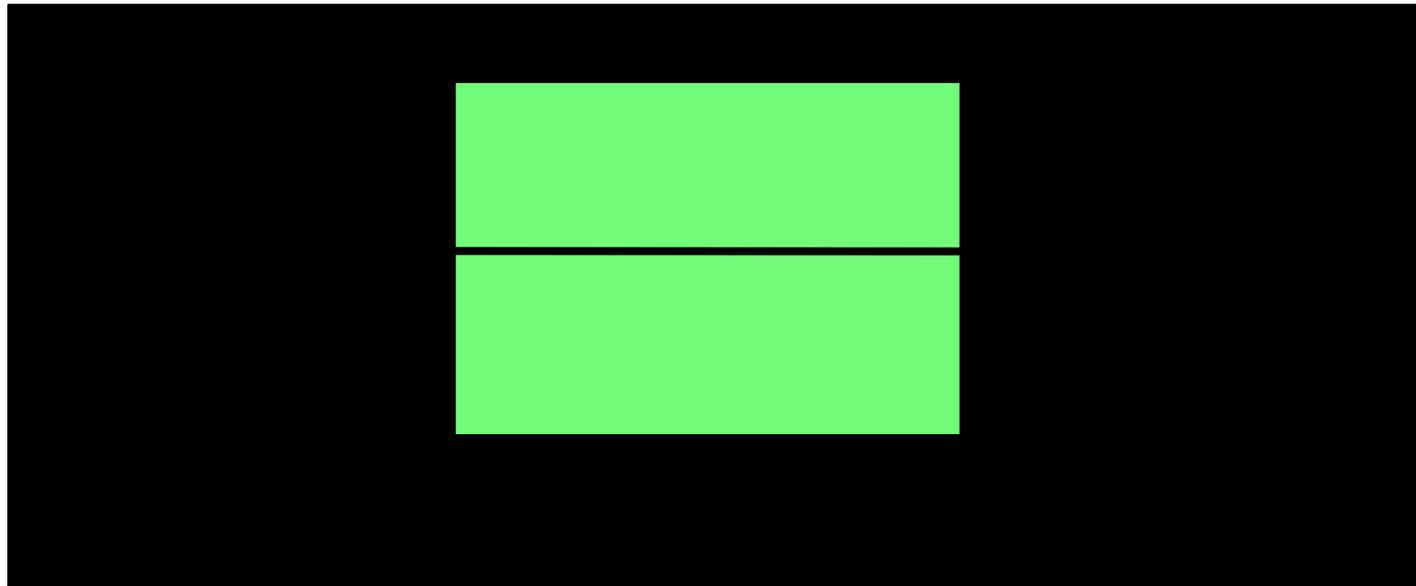
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

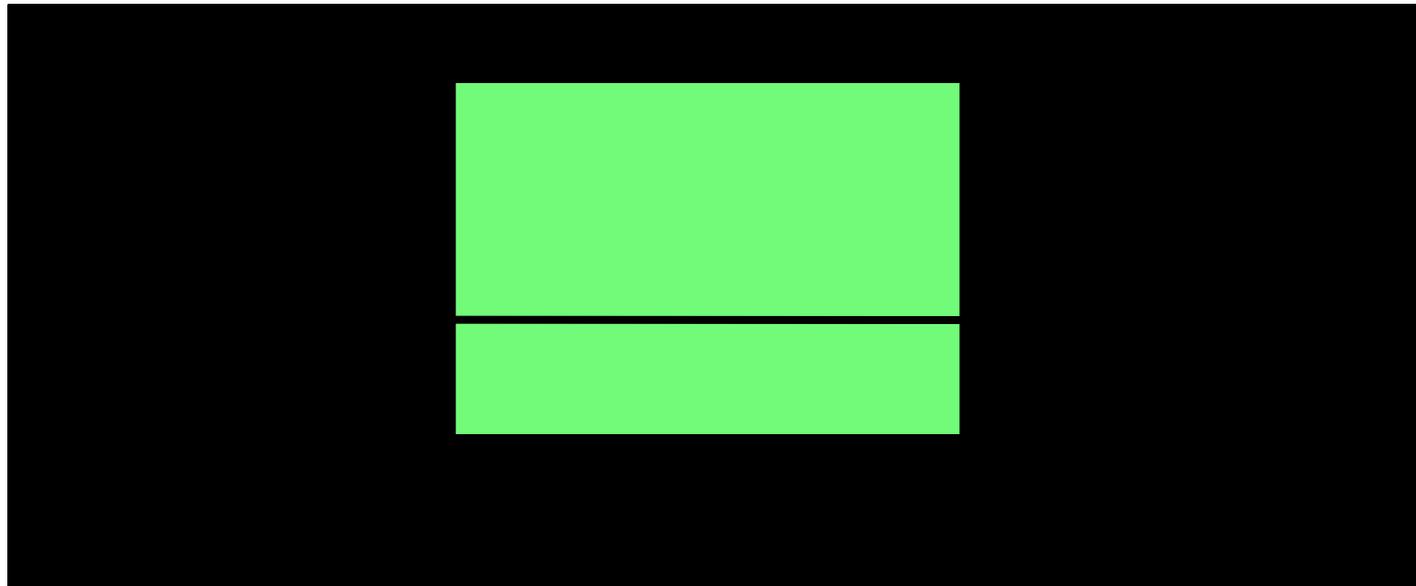
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

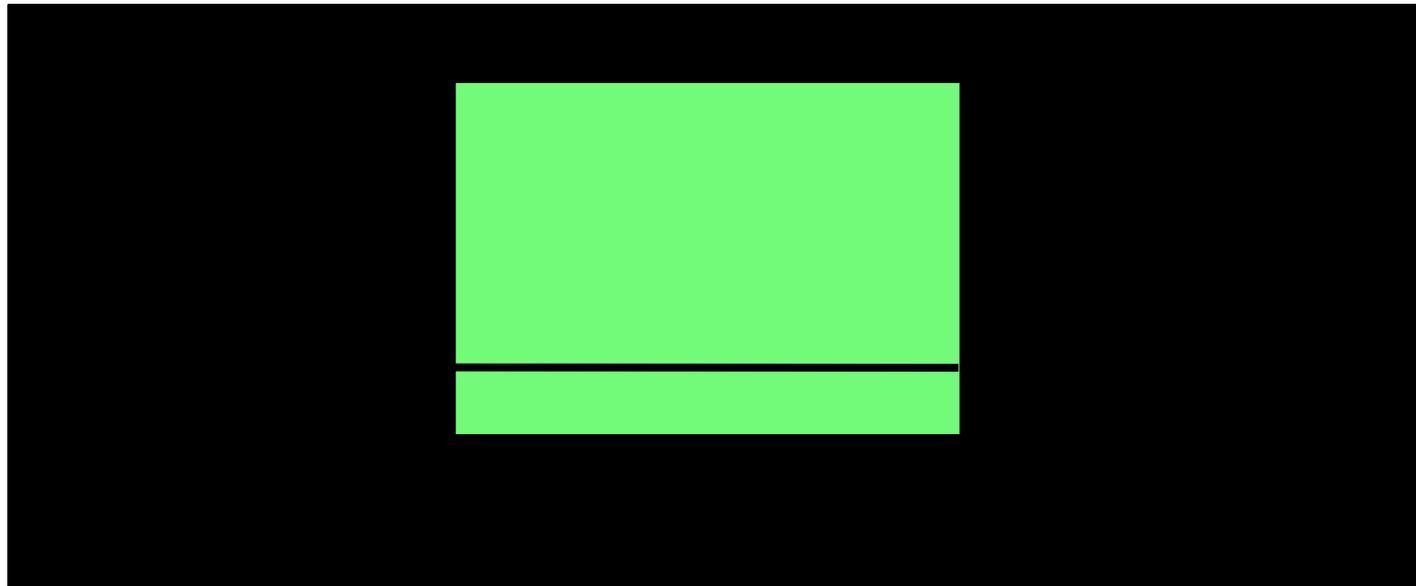
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

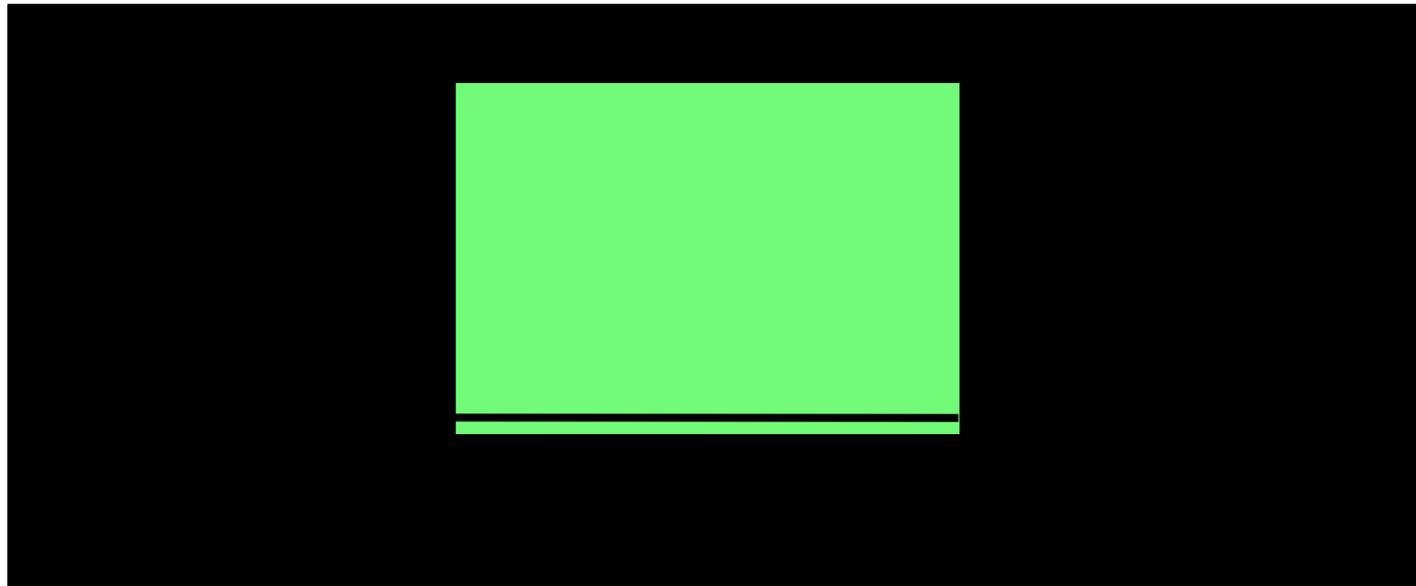
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

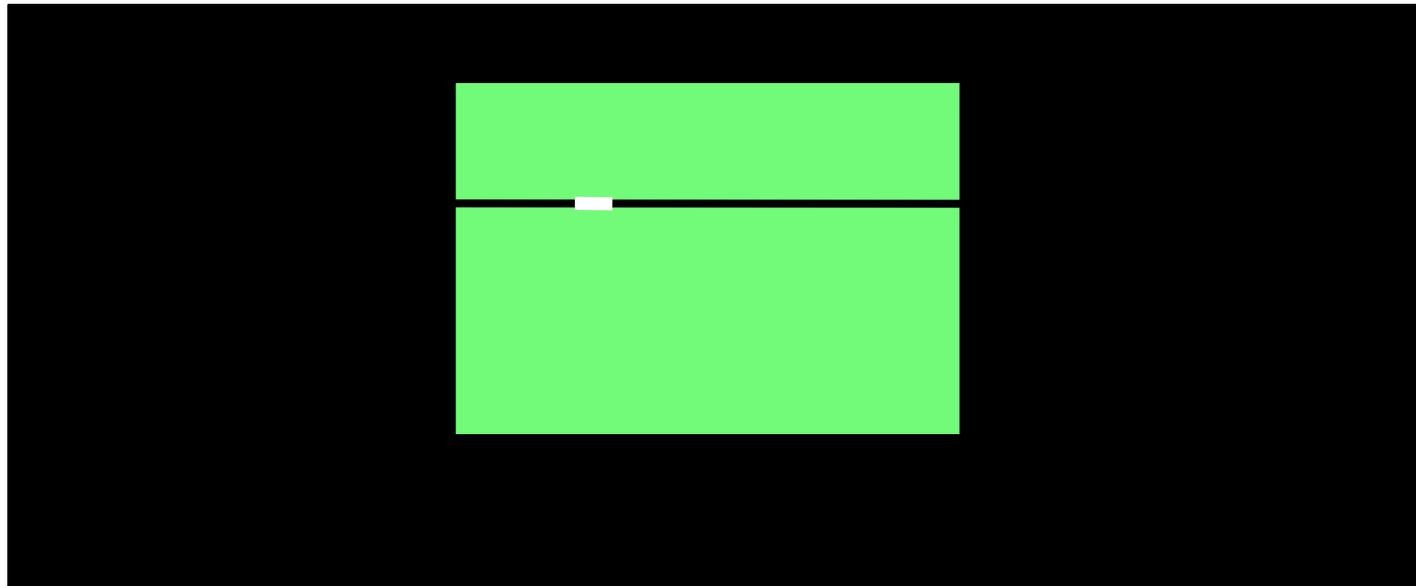
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

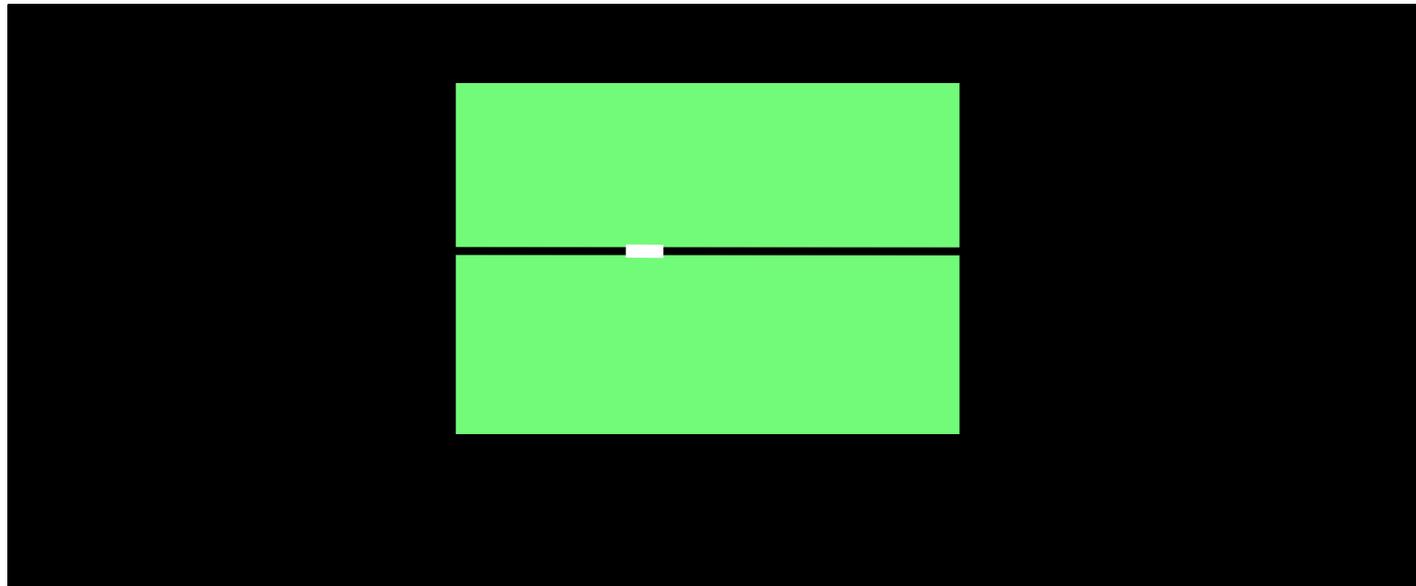
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

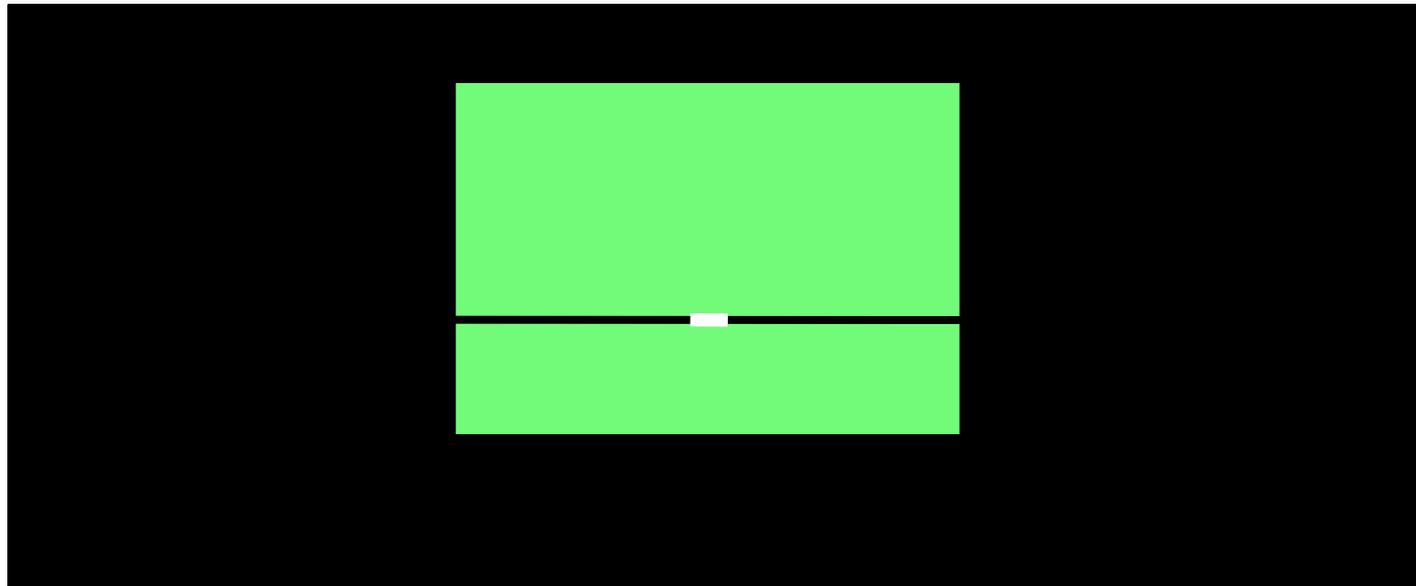
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

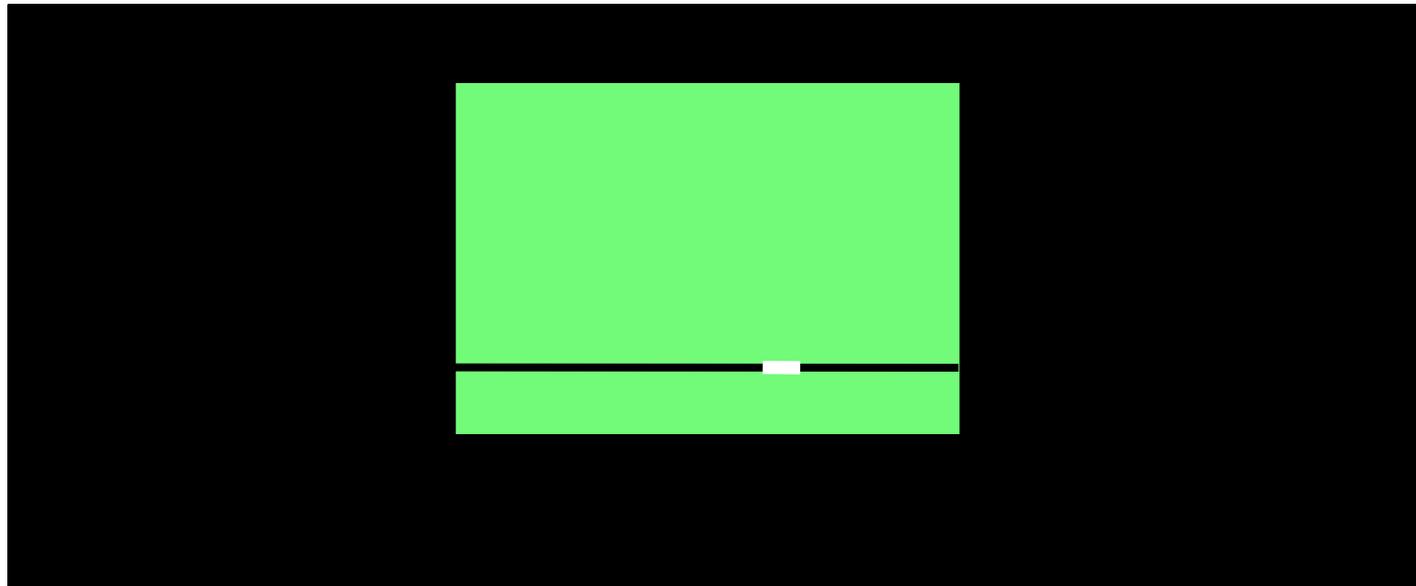
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

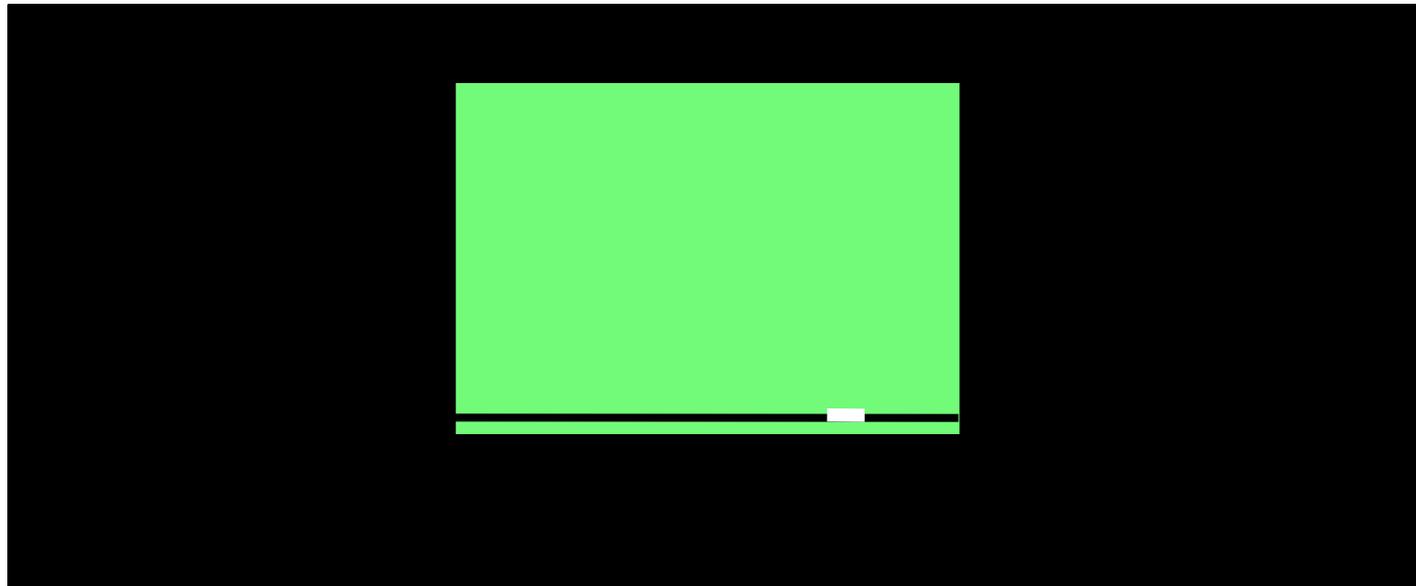
- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



# Bewegungen

Blendenproblem (Aperturproblem):

- Mehrdeutigkeiten bei Verschiebung einer Kante
- Nur senkrecht zur Kante stehende Bewegungskomponenten erkennbar



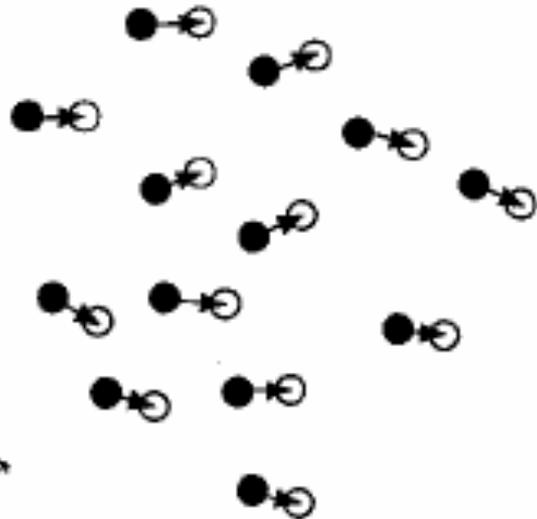
# Bewegungen

Korrespondenzproblem:

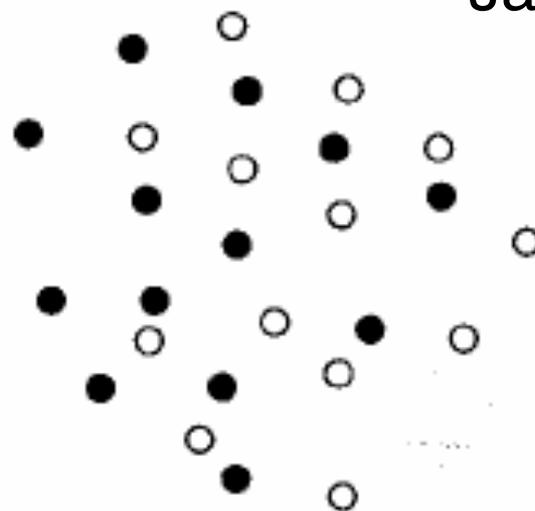
Zuordnung von Objekten in aufeinander folgenden Bildern:

Wichtig: Verhältnis der Bewegung zum Abstand von ununterscheidbaren Objekten

*a*



*b*



Jähne S. 401

# Bewegungen

## Bewegungsfeld:

- Projektion der realen Bewegung in die Bildebene gemäß Abbildung durch Kamera.

## Optischer Fluss:

- Grauwertverschiebung im Bild als Geschwindigkeitsvektor: Berechenbar aus Verschiebungsvektor zwischen zwei Bildern (Differenzbild), dividiert durch zeitlichen Abstand.

i.a. unterschiedlich (z.B. Beleuchtungsunterschiede)

## Beispiele:

- Rotierende Kugel mit gleichartiger Oberfläche
- Kugel in Ruhe bei bewegter Lichtquelle

# Bewegungen

Berechnungen im Modell des Raumes („Weltmodell“):

Kombinieren:

- Angenommene Bewegungen (Extrapolation)
- Beobachtete Bewegungen im Bild
- Kommunizierte Bewegungen anderer Beobachter

Analog zu Selbstlokalisierung (später).

# Bildfolgen verstehen

## Zusätzliche Ziele:

- Abläufe interpretieren
- Eigene Bewegung kontrollieren

## Zusätzliche Möglichkeiten:

- Bezugnahme auf weitere Bilder
- Bestätigung von Annahmen
- Erkennen von Veränderungen

# Entwurfsentscheidungen bzgl. Interpretation

## Ausgangsdaten

- Format der Bilddaten
- Kontext der Bildentstehung (Beleuchtung, Abbildungsmodell, Sensoreigenschaften, ...)

## Repräsentation von (Zwischen-)Modellen

- Prototypen von Objekten und Eigenschaften

## Repräsentation der Zielbeschreibung

- Formalismen der Wissensrepräsentation für Objekte mit Eigenschaften und Beziehungen

# Literatur

- Jähne, Bernd: Digitale Bildverarbeitung. Springer, 2002
- Nehmzow, Ulrich, Peter Haberäcker: Scientific Methods in Mobile Robotics. Springer, 2006
- Burger, Wilhelm, Mark James Burge: Digitale Bildverarbeitung. Springer, 2005
- Haberäcker, Peter: Digitale Bildverarbeitung. Hander 1985.
- Klette, Reinhard, P. Zamperoni: Handbuch der Operatoren für die Bildbearbeitung. Vieweg 1995.
- Mallot, Hans-Peter A.: Sehen und die Verarbeitung visueller Informationen. Vieweg 1998
- Watkins, Christopher, Alberto Sadun, Stephen Marenka: Modern Image Processing. Academic Press 1993.
- Borenstein, Johann (et al.): Where am I . University of