

Überblick

- Was ist ein Kantenpunkt?
- Wie können Kantenpunkte detektiert werden?
- Wie kann aus ermittelten Kantenpunkten eine Kontur bestimmt werden?
- Wie können Konturen „geschlossen“ werden?

Kantenpunkt(e)

- Grauwertkanten, Texturkanten und Farbkanten
- Grauwertänderung benachbarter Bildregionen
- Pixelorientiert
- Zweistufiges Vorgehen
 - Auffinden von Kantenpunkten
 - Verkettung von Kantenpunkten
- Auffinden durch diskrete Ableitung(en)
- Ziel: Kantenbild (in GW codiert oder auch binär)

Einleitung / Diskrete Ableitung

- 1te und 2te Ableitung
- Allg. Beschreibung mit Differenzengleichungen
- Zwei Veränderliche im 2D-Fall: x - und y - Richtung

$$E'_x(x, y) = \frac{\partial E(x, y)}{\partial x} \quad \text{und} \quad E'_y(x, y) = \frac{\partial E(x, y)}{\partial y}$$

Einleitung / Diskrete Ableitung

- 1te Ableitung
- x -Richtung:

$$\frac{\partial E(x, y)}{\partial x} = \Delta_x E(x, y) = \frac{E(x, y) - E(x - \Delta x, y)}{\Delta x}$$

- y -Richtung:

$$\frac{\partial E(x, y)}{\partial y} = \Delta_y E(x, y) = \frac{E(x, y) - E(x, y - \Delta y)}{\Delta y}$$

Einleitung / Diskrete Ableitung

$\Delta_x E(x, y)$ und $\Delta_y E(x, y)$ diskrete Ableitung

in x – und y – Richtung oder

Gradienten in x – und y – Richtung

mit $\Delta x = 1$ und $\Delta y = 1$ ist Ableitung in

in x – und y – Richtung

$$\Delta_x E(x, y) = E(x, y) - E(x - 1, y)$$

$$\Delta_y E(x, y) = E(x, y) - E(x, y - 1)$$

Einleitung / Diskrete Ableitung

Als Filtermasken C_x und C_y

$$C_x = \begin{vmatrix} -1 & 1 \end{vmatrix} \text{ und } C_y = \begin{vmatrix} -1 \\ 1 \end{vmatrix}$$

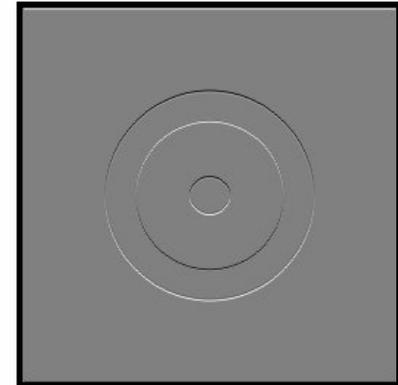
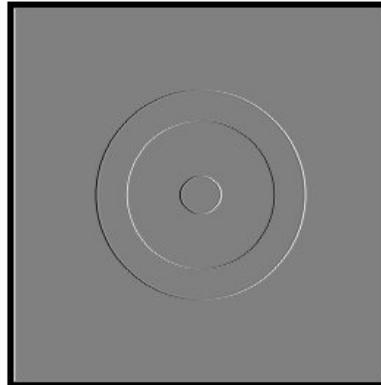
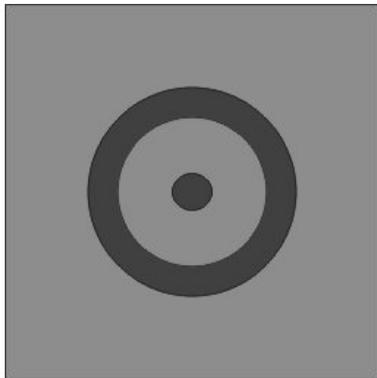
Ableitung in beliebiger Richtung Θ

mittels (horiz.) x – und (vert.) y – Ableitungen

$$\Delta_{\Theta} E(x, y) = \Delta_x E(x, y) \cos \Theta + \Delta_y E(x, y) \sin \Theta$$

Einleitung / Diskrete Ableitung

Gradientenbild: Differenzierung in zwei orthogonale Richtungen (für jeden Bildpunkt) mit C_x und C_y



Einleitung / Diskrete Ableitung

Gradientenbetrag:

$$\|\nabla\| = \sqrt{(\Delta_x E(x, y))^2 + (\Delta_y E(x, y))^2}$$

10	10	10	5
10	10	5	5
10	5	5	5
5	5	5	5

Gradientenrichtung:

$$\Theta(x, y) = \arctan\left(\frac{\Delta_y E(x, y)}{\Delta_x E(x, y)}\right)$$

[Online Berechnung arctan](#)

Detektion von Kantenpunkten

➤ Einseitiger Kantenoperator

[Klette & Zamperoni 95], S. 209

Differenz zwischen dem aktuellen GW und dem minimalem GW in einer 8er-Nachbarschaft

Ergebnis: pixelbreite Kante(npunkte)

Zwischenergebnis (wg. Binarisierungsmöglichkeit) $A_z(x,y)$

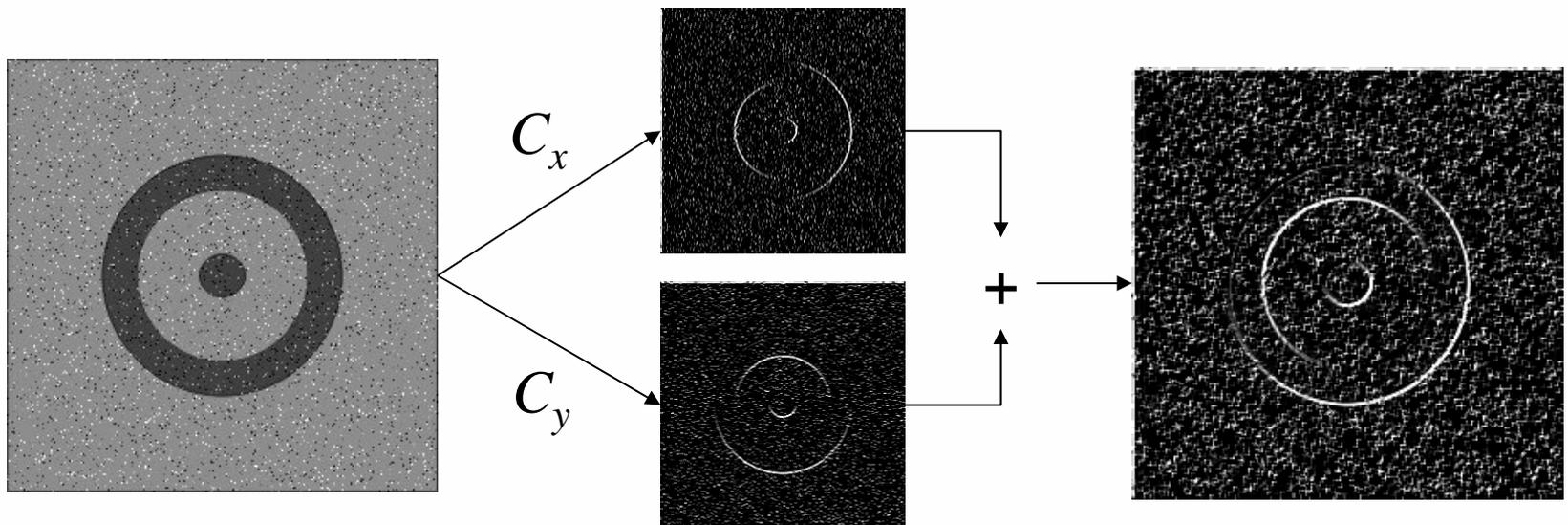
$$A_z(x, y) = E(x, y) - \min(\{(i, j) \mid (i, j) \in \Gamma\})$$

mit Γ als 8ter Nachbarschaft

$$A(x, y) = \begin{cases} A_z(x, y), & \text{wenn } \textit{binarize} = \textit{false} \\ \text{sonst,} & \begin{cases} 0, & \text{wenn } A_z(x, y) < S \\ \text{GW} - 1, & \text{wenn } A_z(x, y) \geq S \end{cases} \end{cases}$$

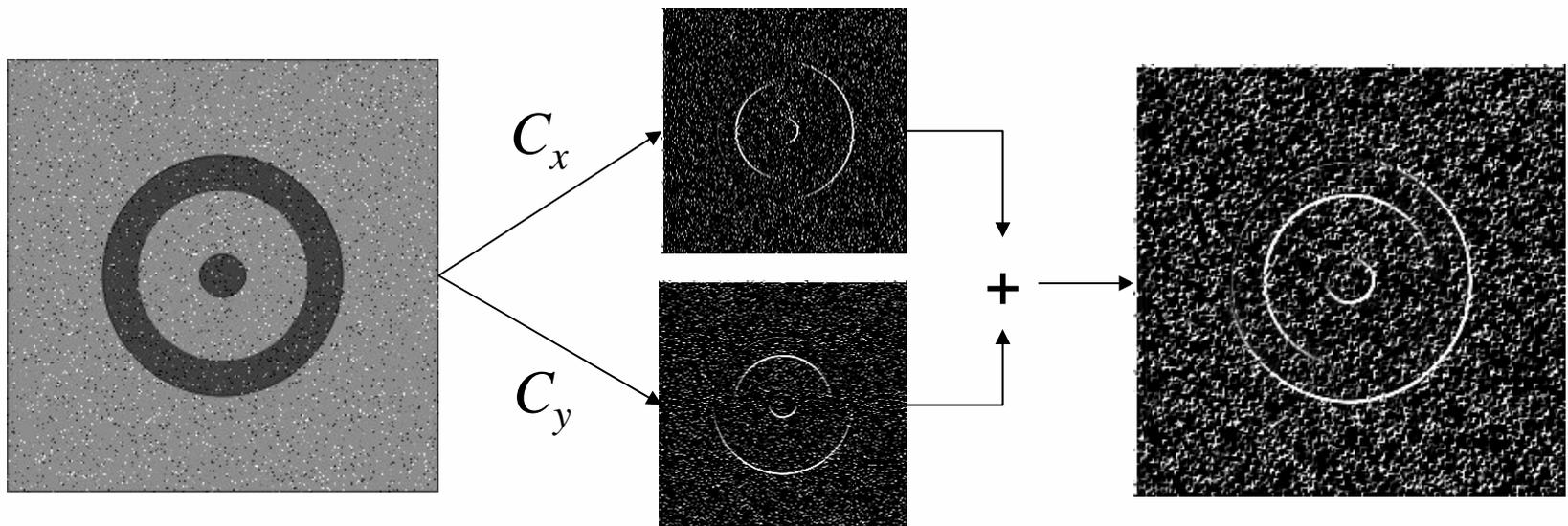
Detektion von Kantenpunkten

➤ Prewitt-Operator: $C_{Px} = \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$ bzw. $C_{Py} = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix}$



Detektion von Kantenpunkten

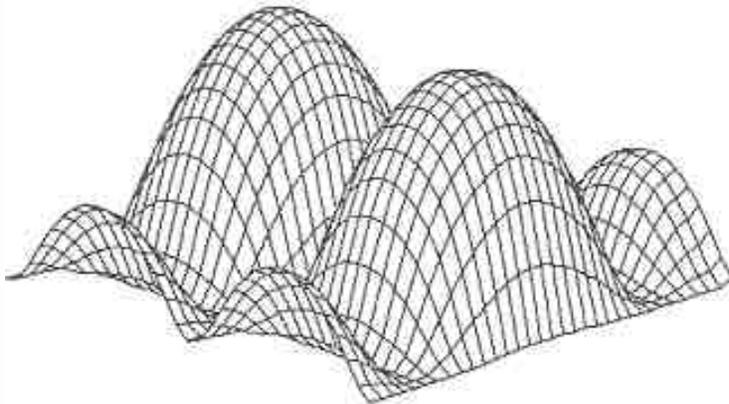
➤ Sobel-Operator: $C_{Sx} = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix}$ bzw. $C_{Sy} = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}$



Detektion von Kantenpunkten

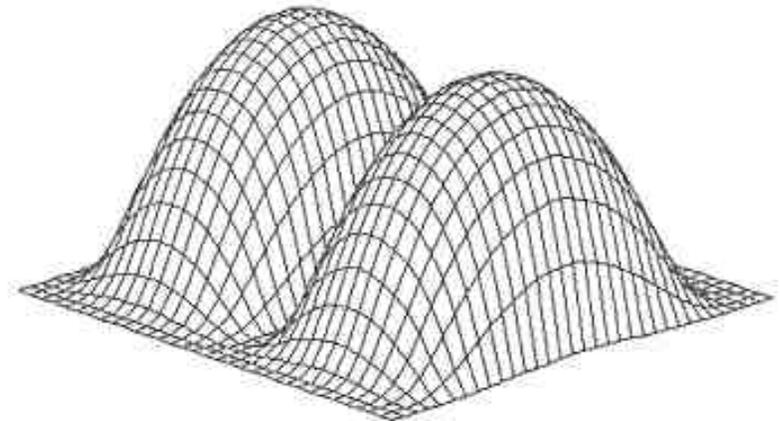
- Vergleich Powerspektrum von Prewitt & Sobel-Operator:

Powerspektrum Prewitt vertikal



(A)

Powerspektrum Sobel vertikal



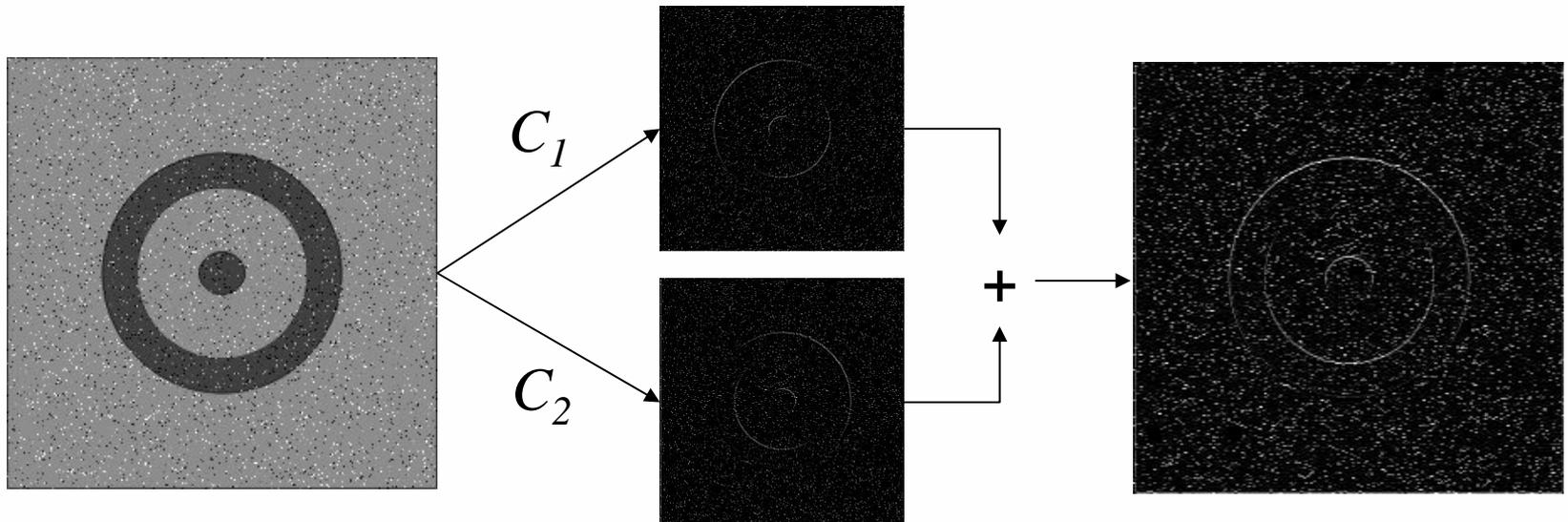
(B)

[Abmayr 94], S. 191

Detektion von Kantenpunkten

➤ Roberts-Cross:

$$C_1 = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \quad \text{bzw.} \quad C_2 = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix}$$



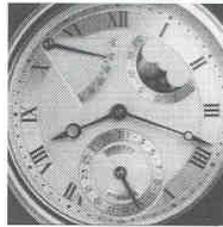
Detektion von Kantenpunkten

➤ Kompaaßgradienten-Operatoren:

$$H_0 = \begin{vmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{vmatrix} \quad H_1 = \begin{vmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{vmatrix} \quad H_2 = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{vmatrix} \quad H_3 = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{vmatrix}$$

$$H_4 = \begin{vmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{vmatrix} \quad H_5 = \begin{vmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{vmatrix} \quad H_6 = \begin{vmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad H_7 = \begin{vmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Detektion von Kantenpunkten



(A)



(B)



(C)



(D)



(E)



(F)



(G)



(H)



(I)

Detektion von Kantenpunkten

➤ Kirsch-Operator(en):

$$K_0 = \begin{vmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{vmatrix} \quad K_1 = \begin{vmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{vmatrix} \quad K_2 = \begin{vmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{vmatrix} \quad K_3 = \begin{vmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{vmatrix}$$

$$K_4 = \begin{vmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{vmatrix} \quad K_5 = \begin{vmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{vmatrix} \quad K_6 = \begin{vmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{vmatrix} \quad K_7 = \begin{vmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{vmatrix}$$

Diskrete Ableitung

- 2te Ableitung
- x -Richtung:

$$\begin{aligned}\Delta_x^2 E(x, y) &= \Delta_x E(x+1, y) - \Delta_x E(x, y) \\ &= \left((E(x+1, y) - E(x, y)) - (E(x, y) - E(x-1, y)) \right) \\ &= E(x+1, y) + E(x-1, y) - 2E(x, y)\end{aligned}$$

entsprechendes für die y -Richtung

$$\Delta_y^2 E(x, y) = E(x, y+1) + E(x, y-1) - 2E(x, y)$$

Diskrete Ableitung

Als Filtermasken C_{L_H} und C_{L_V}

$$C_{L_H} = \begin{vmatrix} 1 & -2 & 1 \end{vmatrix} \text{ und } C_{L_V} = \begin{vmatrix} 1 \\ -2 \\ 1 \end{vmatrix}$$

Summe der partiellen Ableitungen in x - und y - Richtung wird als Laplace - Operator bezeichnet

$$\nabla^2 E(x, y) \approx \frac{\partial^2 E(x, y)}{\partial x^2} + \frac{\partial^2 E(x, y)}{\partial y^2}$$

Diskrete Ableitung

$\nabla^2 E(x, y)$ ist Laplace - Operator für diskreten Fall

Realisierung:

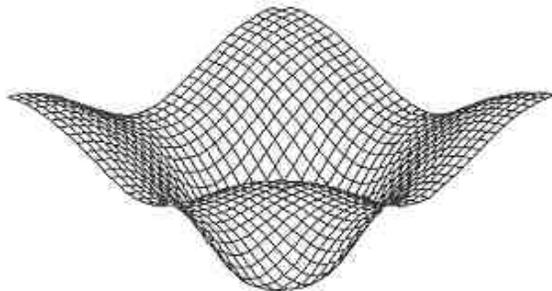
$$\begin{aligned}\nabla^2 E(x, y) &= \Delta_x^2 E(x, y) + \Delta_y^2 E(x, y) \\ &= E(x+1, y) + E(x, y+1) \\ &\quad + E(x-1, y) + E(x, y-1) - 4E(x, y)\end{aligned}$$

Diskrete Ableitung

Filtermasken für ein Laplace - Operator bzgl.
4er - und 8er - Nachbarschaft

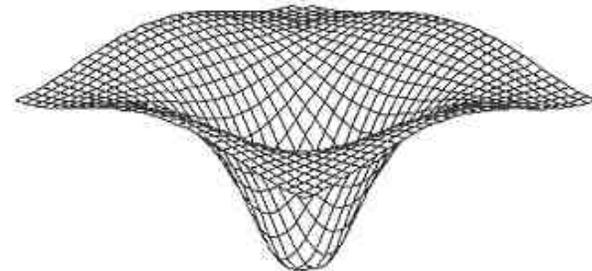
$$C_{L_1} = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

Powerspektrum Laplace1



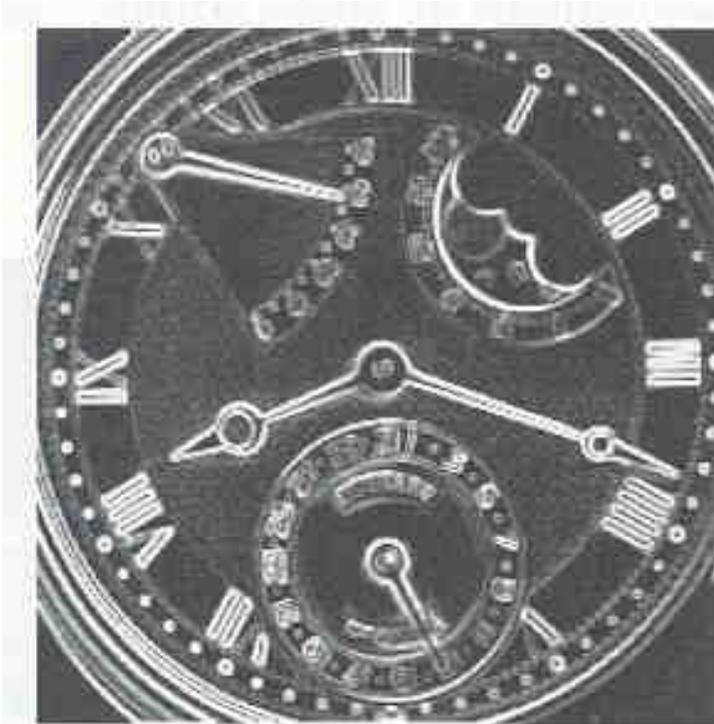
$$C_{L_2} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Powerspektrum Laplace2



Diskrete Ableitung

Max(Kirsch)



8er-Laplace



[Abmayr 94], S. 194

Diskrete Ableitung

- Diskussion: Gradienten- (G) & Laplace- (L) Filter
 - Einzelpunkte:
 - (G): verschmieren; (L): verstärken
 - Linienendpunkte:
 - (G): verbreitern der Linie; (L): verstärken der Linie, besonders der Endpunkt
 - Linien:
 - (G): verbreitern der Linie; (L): verstärken der Linie
 - Kante:
 - (G): Ergebnis ist Linie; (L): Ergebnis ist verbreiterte Linie

Diskrete Ableitung



[Jähne 89], S. 101

- Kantendetektion nach Marr & Hildreth
 - GW-Änderungen treten auf verschiedenen Skalen auf -> Operatoren unterschiedlicher Größe erforderlich
 - GW-Änderungen resultieren in einem Peak bzgl. der ersten Ableitung und in einem Nulldurchgang bzgl. der zweiten Ableitung

Diskrete Ableitung

Kantendetektion nach Marr & Hildreth

$\nabla^2 G$ (LoG) erfüllt beide Bedingungen mit

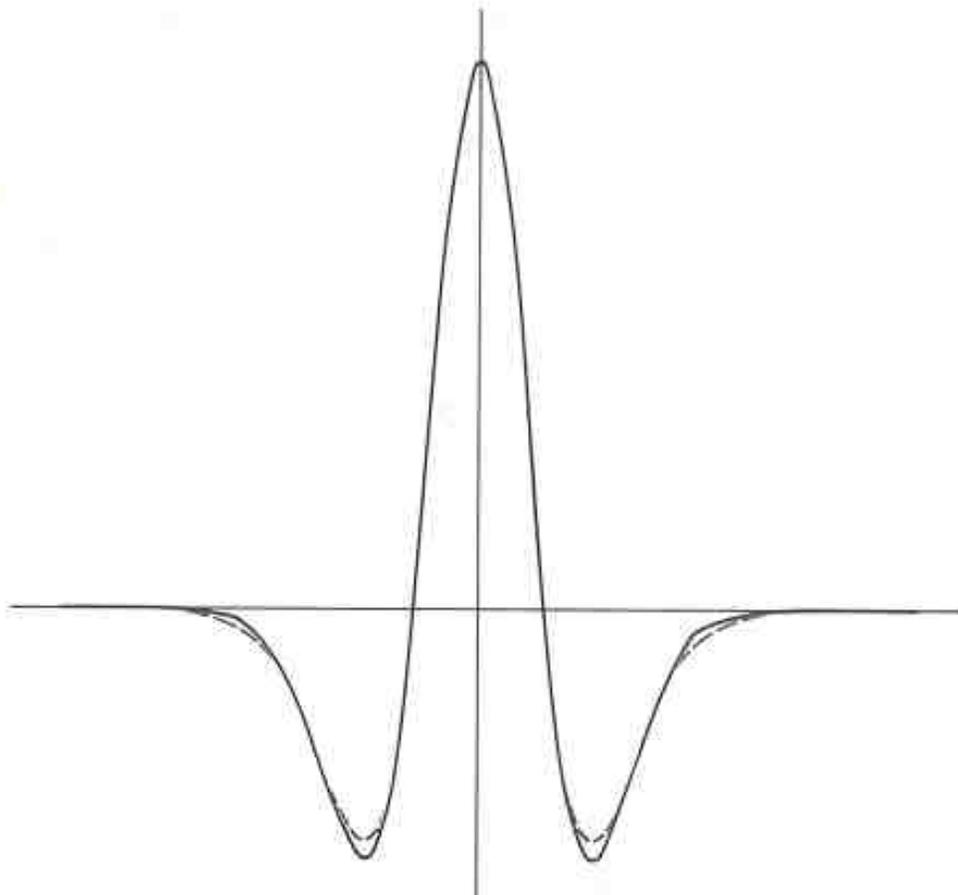
∇^2 als Laplace – Operator und

G als 2D Gauß - Verteilung

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\pi\sigma^2}}$$

$$(G(x, y) * E(x, y))^n = G(x, y)^n * E(x, y)$$

Diskrete Ableitung



LoG wird durch
einen DoG ange-
nähert

Verhältnis: 1:1.6

Gauß-Verteilung
durch Binomial-
verteilung ange-
nähert

[Marr 82], S. 63

Diskrete Ableitung

Binomial-
verteilung
aus
Pascalschen
Dreieck

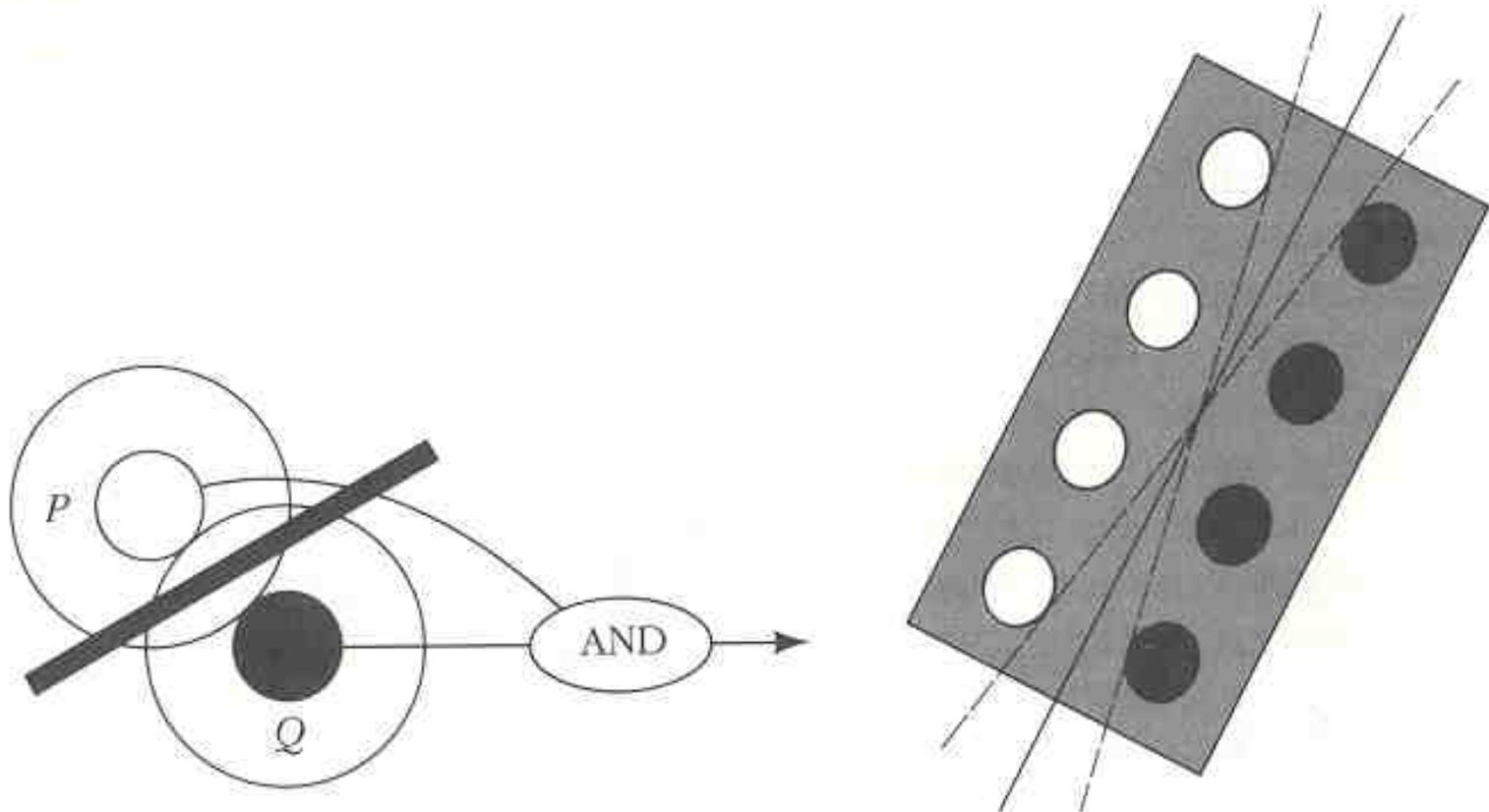
n	f		σ^2
0	1	1	0
1	$\frac{1}{2}$	1 1	$\frac{1}{4}$
2	$\frac{1}{4}$	1 2 1	$\frac{1}{2}$
3	$\frac{1}{8}$	1 3 3 1	$\frac{3}{4}$
4	$\frac{1}{16}$	1 4 6 4 1	1
5	$\frac{1}{32}$	1 5 10 10 5 1	$\frac{5}{4}$
6	$\frac{1}{64}$	1 6 15 20 15 6 1	$\frac{3}{2}$
7	$\frac{1}{128}$	1 7 21 35 35 21 7 1	$\frac{7}{4}$
8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

Diskrete Ableitung

2D Gauß-Verteilung für $n=2$

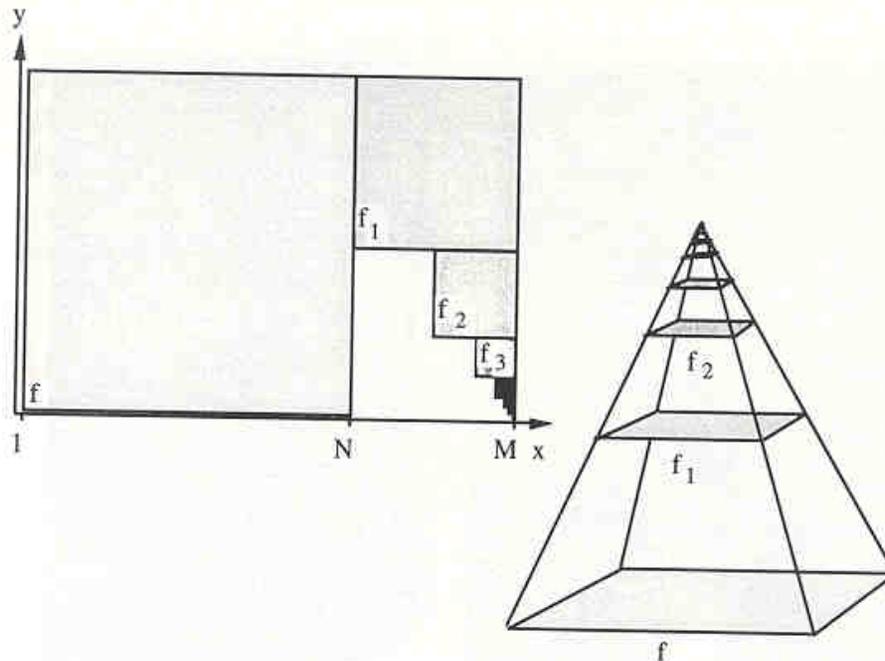
$$G = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} * \frac{1}{4} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Diskrete Ableitung



[Marr 82], S. 66

Bildpyramide



[Klette & Zamperoni 95], S. 119

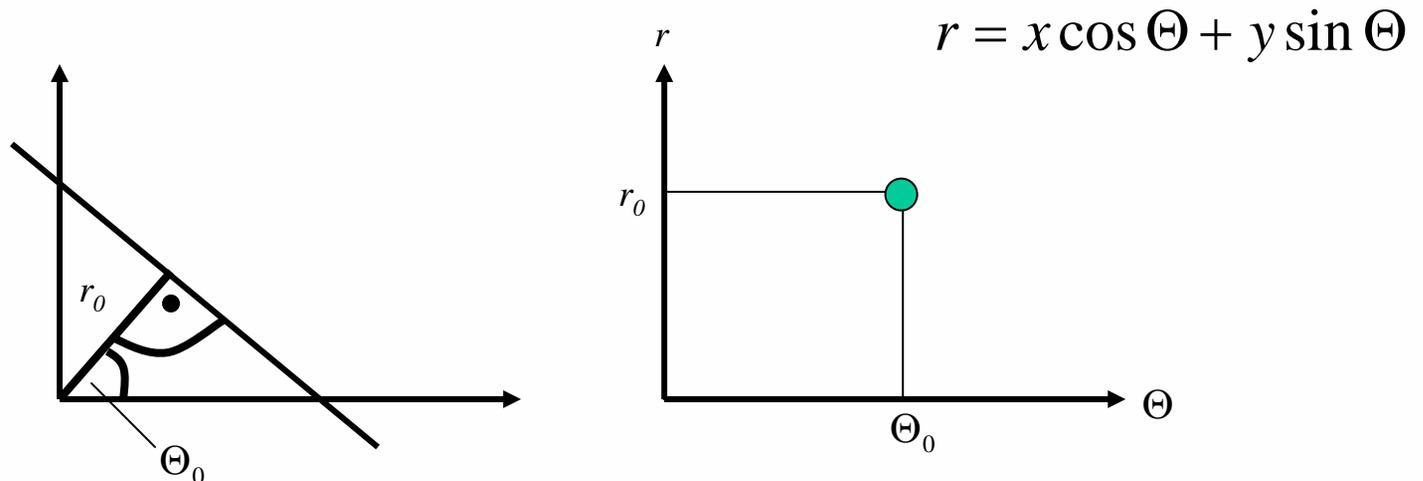
- Verschieden Auflösungen eines Bildes
- Auswahl geeigneter Auflösung (Rechenzeit vs. Detailgenauigkeit)
- Verkleinerung um jeweils die Hälfte der Länge und Breite (-> „normale“ Pyramide) oder durch Unterabtastung
- Pyramiden vs. Türme

Kontur

- Kontur & Kantenpunkt
 - Kantenpunkt: einzelnes, punktgroßes Kantenstück
 - Mögliche Kontur besteht nur aus „Lücken“, da Kantenpunkte nicht logisch zusammengehörig
 - Zusammenhängende Kantestücke (Kantenpunkte)
 - Lücken können auftreten, einzelne Kantenpunkte einzelner Kantenstücke gehören logisch zusammen
 - Geschlossene Kontur
 - Keine Lücken, keine einzelne Kantenstücke, ganze Objektkontur gehört logisch zusammen

Kontur

- Hough-Transformation
 - Anwendungsschwerpunkt: Kontursegmentierung
 - Geradendarstellung in Hessescher Normalform

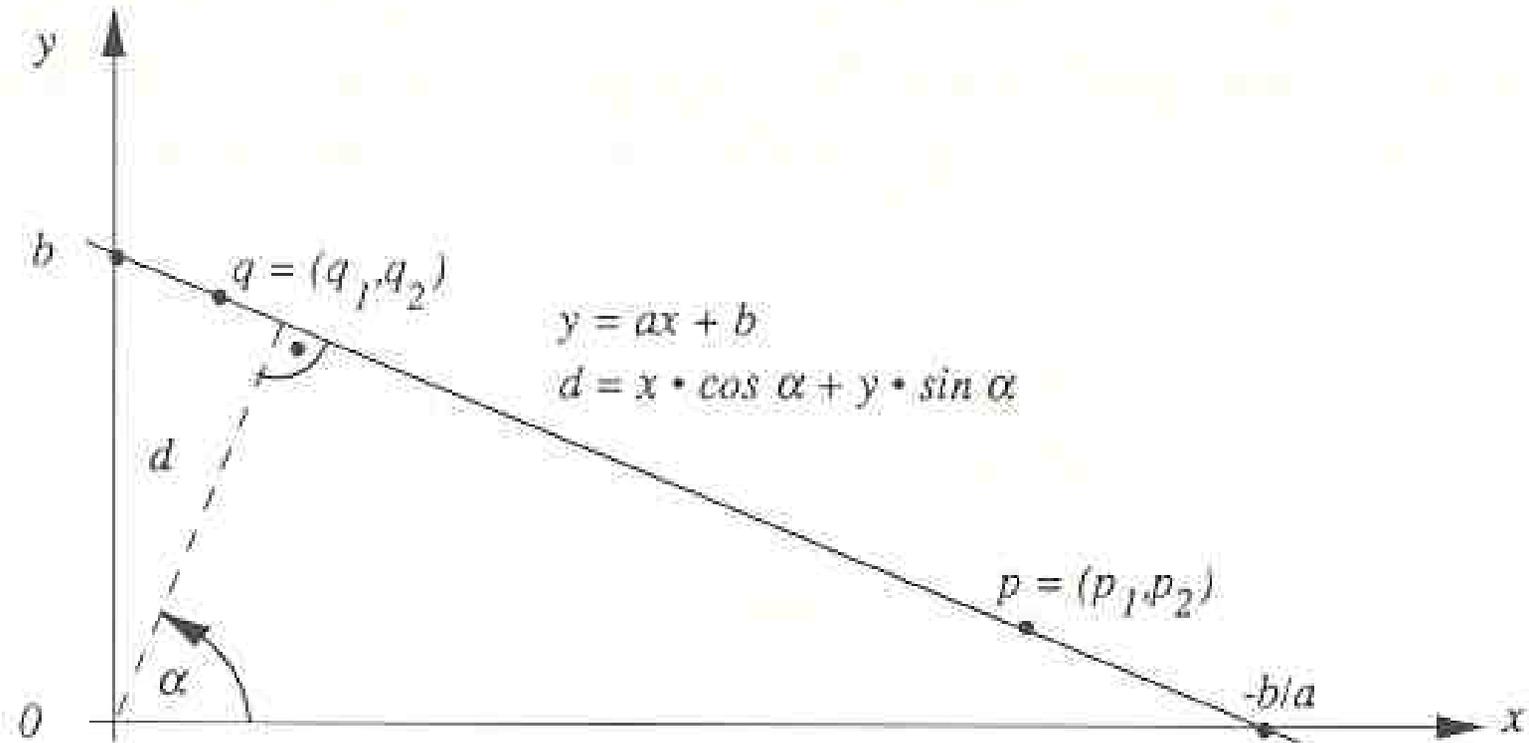


- „Akkumulator“ für Punkteinträge
- Aufdeckung von Kollinearität von Kantenpunkten (Lückenschließung)

Kontur

➤ Hough-Transformation

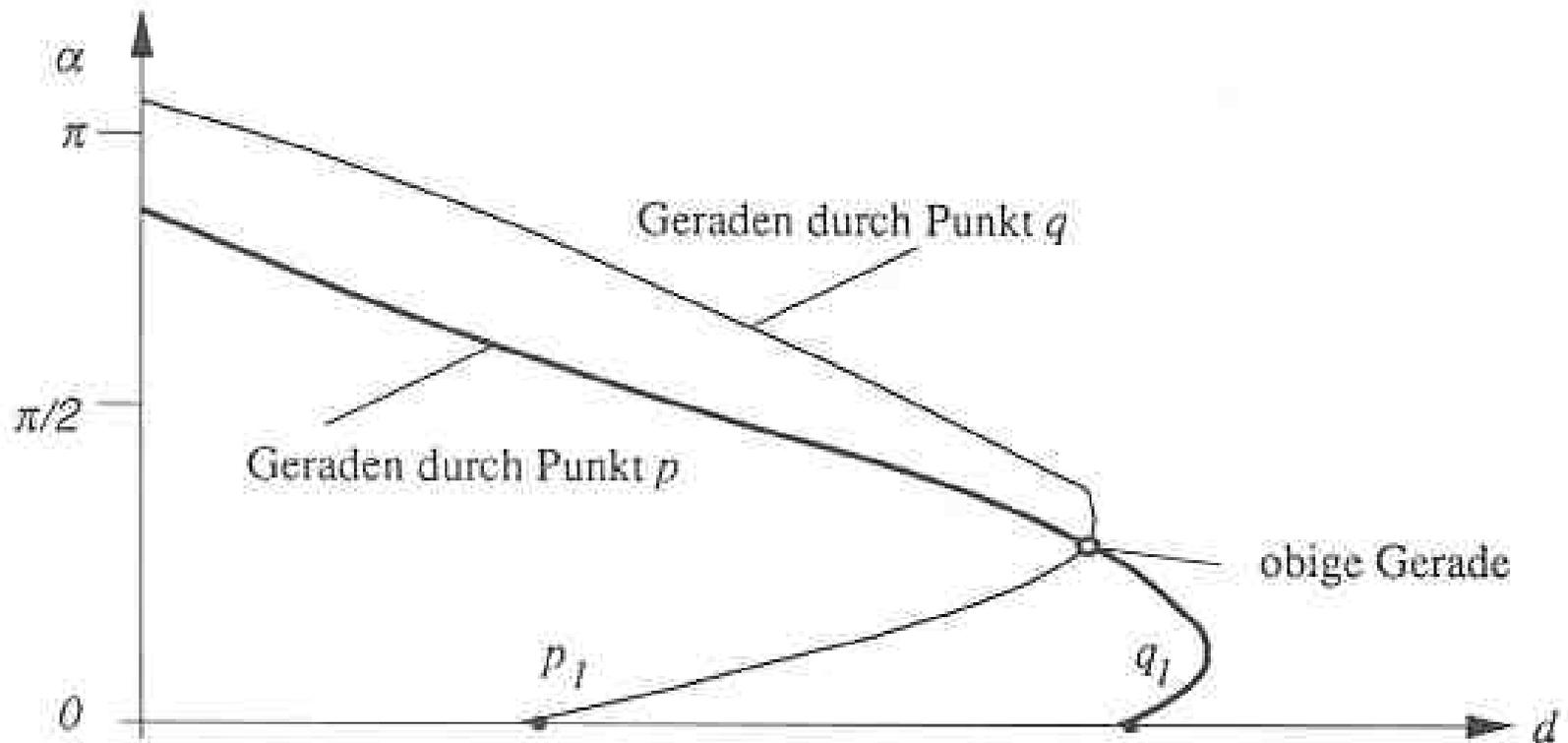
[Klette & Zamperoni 95], S. 332



Kontur

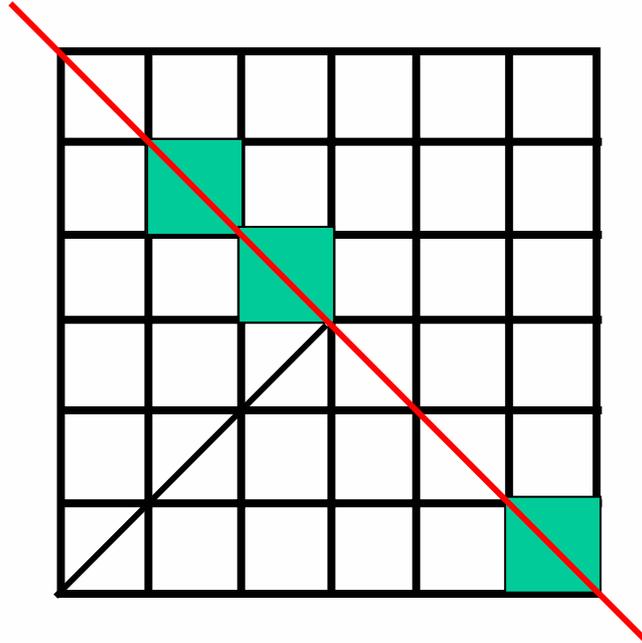
➤ Hough-Transformation

[Klette & Zamperoni 95], S. 332

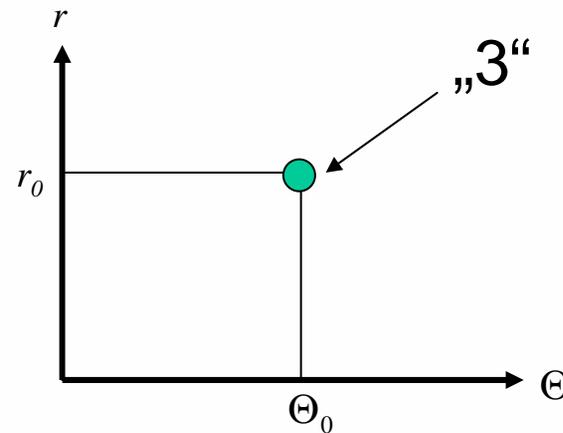


Kontur

➤ Hough-Transformation



 = markierter Kantenpunkt



Kontur

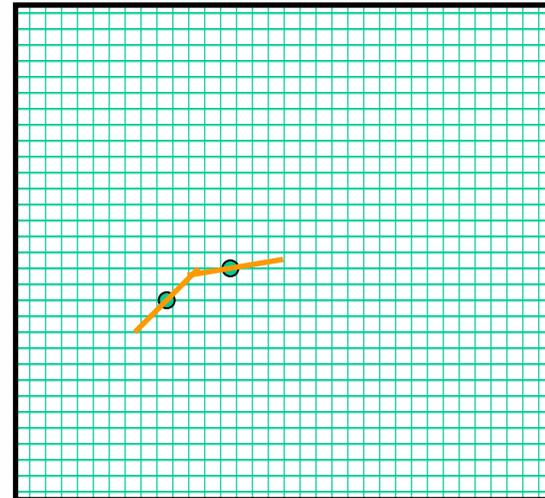
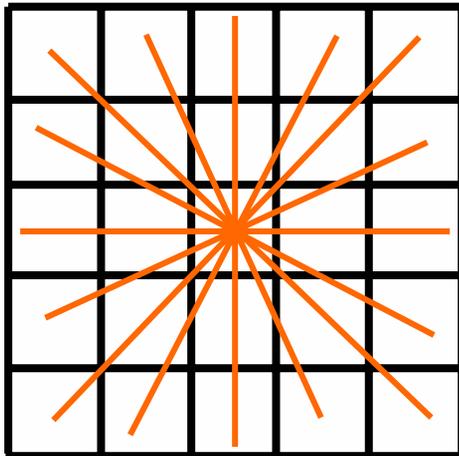
- Verkettung von Kantenpunkten
 - Lokale vs. globale Verfahren
 - Lokal:
 - rel. Einfach zu realisieren,
 - aber bei sehr komplexen Szenen unbefriedigende Ergebnisse
 - Global:
 - nutz a priori Wissen,
 - erheblich höherer Aufwand,
 - i.d.R. bessere Ergebnisse

Kontur

- Kantenstücke für n -Richtungen in lokaler Nachbarschaft
- Kettencode
- Geradenvervollständigung
- Relaxationsverfahren
- Approximation von Kurven(-stücken) durch Geraden(-stücke) (mit maximal erlaubten Fehler, Endpunkte sind Startpunkte,...)
- Approximation durch Polygone

Kontur

- Kantenstücke für n-Richtungen in lokaler Nachbarschaft
 - Gradientenrichtung bestimmt Richtung des Kantenstücks
 - schließen von Lücken



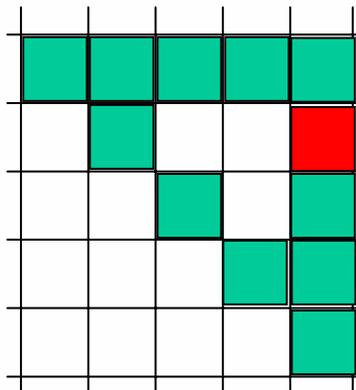
Kontur

Kontur

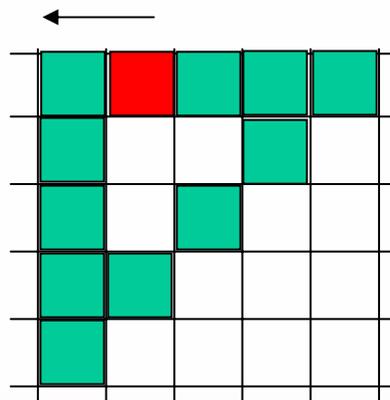
- Kettencode
- Drehung auf dem Kettencode

Kettencode für
8er Nachbarschaft

	4	3	2
	5	x/y	1
	6	7	8



Dreieck=355558888333

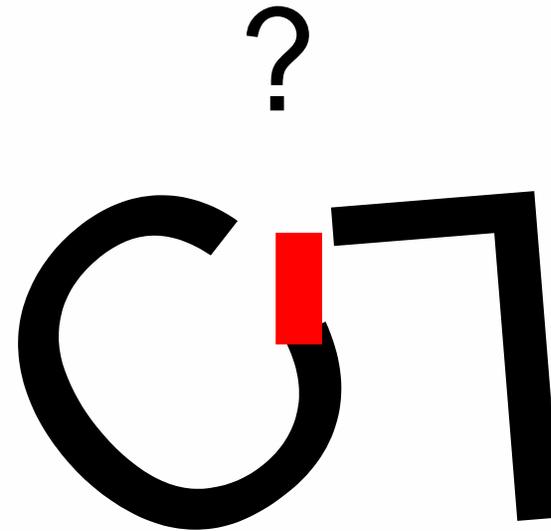
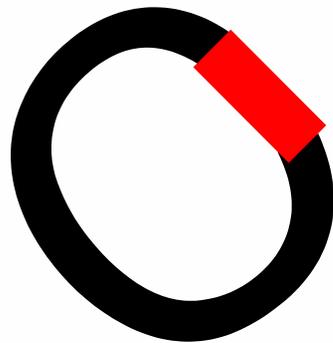


Drehung: 90° ::= +2

Dreieck=577772222555

Kontur

- „Geradenvervollständigung“

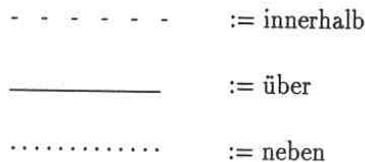
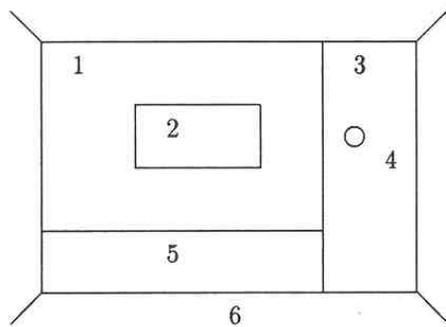


Kontur

- Relaxationsverfahren
 - Markierungsproblem ist gegeben durch
 - 1.) Menge von Objekte (a_1, \dots, a_n)
 - 2.) Menge von Markierungen $(\lambda_1, \dots, \lambda_m)$
 - 3.) Nachbarschaftsbeziehung zwischen den Objekten
 - 4.) Relation, die sog. constraints (+ Wechselwirkungsbeziehungen) zwischen Markierung für Paare (oder n -Tupel) repräsentiert
 - Ziel: konsistente Zuweisung der Markierungen zu den Objekten

Kontur

➤ Relaxationsverfahren (Beispiel/diskret)



Menge von Objekten $U = \{1,2,3,4,5,6\}$

Menge von Markierungen $L = \{W, T, B, F, K, L\}$

$W = \text{Wand}$, $T = \text{Tür}$, $B = \text{Bild}$, $F = \text{Fußboden}$,
 $K = \text{Klinke}$, $L = \text{Leiste}$

(lokaler) Kontext (3. und 4.) Graph $G = (u, v)$ mit
 $V \subseteq U \times U$ und $u \in U : n(u) = \{u' \mid (u, u') \in V\}$

Kontur

➤ Relaxationsverfahren (Beispiel)

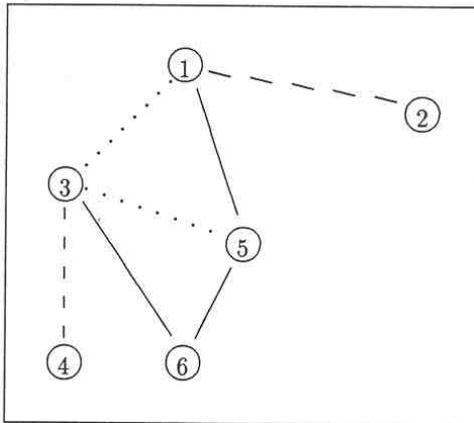


Abbildung 2.2: Ein Bildgraph entsprechend der Szene aus Abbildung 2.1.

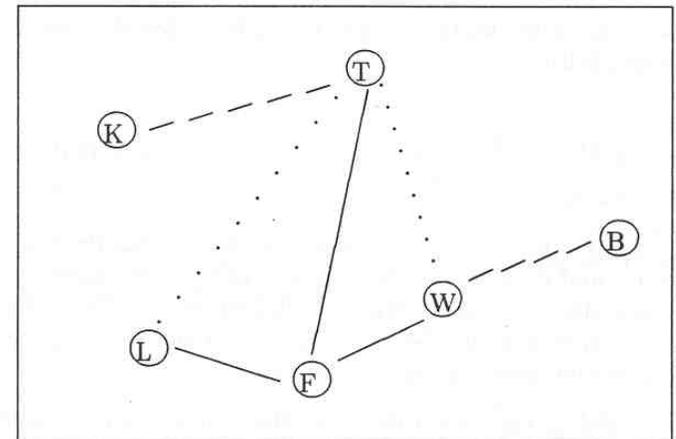


Abbildung 2.3: Ein Graph, der das Modell der Welt symbolisiert.

Kontur

- Relaxationsverfahren
 - Einschränkungen:
 - Unär (Größe der Objekte)
 - Binär (innerhalb, über und neben)
 - Kompatibilitätsfunktion

$$\Lambda : U \times U \times L \times L \rightarrow \{0,1\}$$

- Eine mögliche Belegung

$$\Lambda : u \times u' \times l \times l' \in \{0,1\}$$

Kontur

➤ Relaxationsverfahren

wobei $u = 1, u' = 5, l = W, l' = L$

d.h.: "1" über "5" und "W" über "L"

$$\Lambda(u, u', l, l') = 1,$$

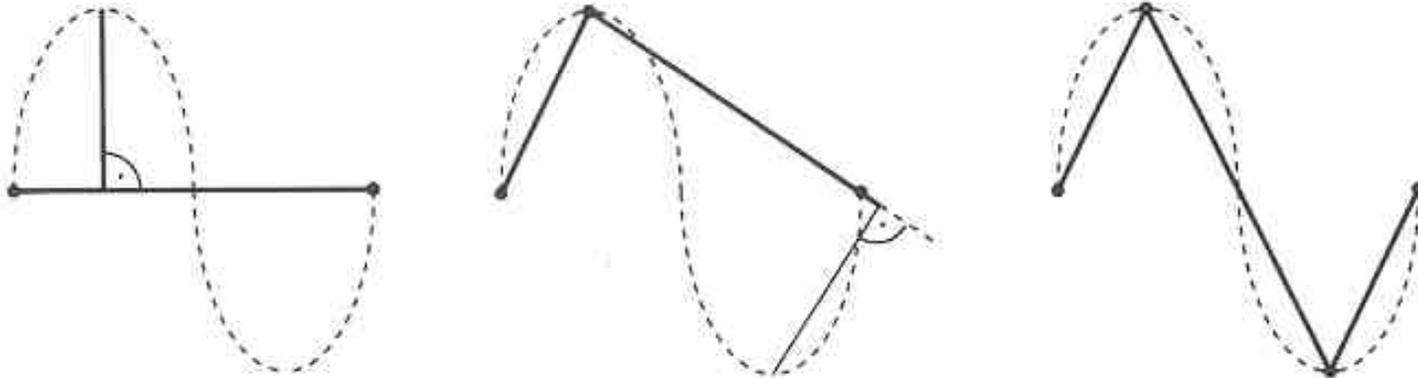
in Worten : paarweise kompatibel

update - rule

$$\lambda^i(u) = \left\{ l \in \lambda^{i-1}(u) \left[\prod_{u' \in n(u)} \sum_{l' \in \lambda^i(n')} \Lambda(u, u', l, l') \right] > 0 \right\}$$

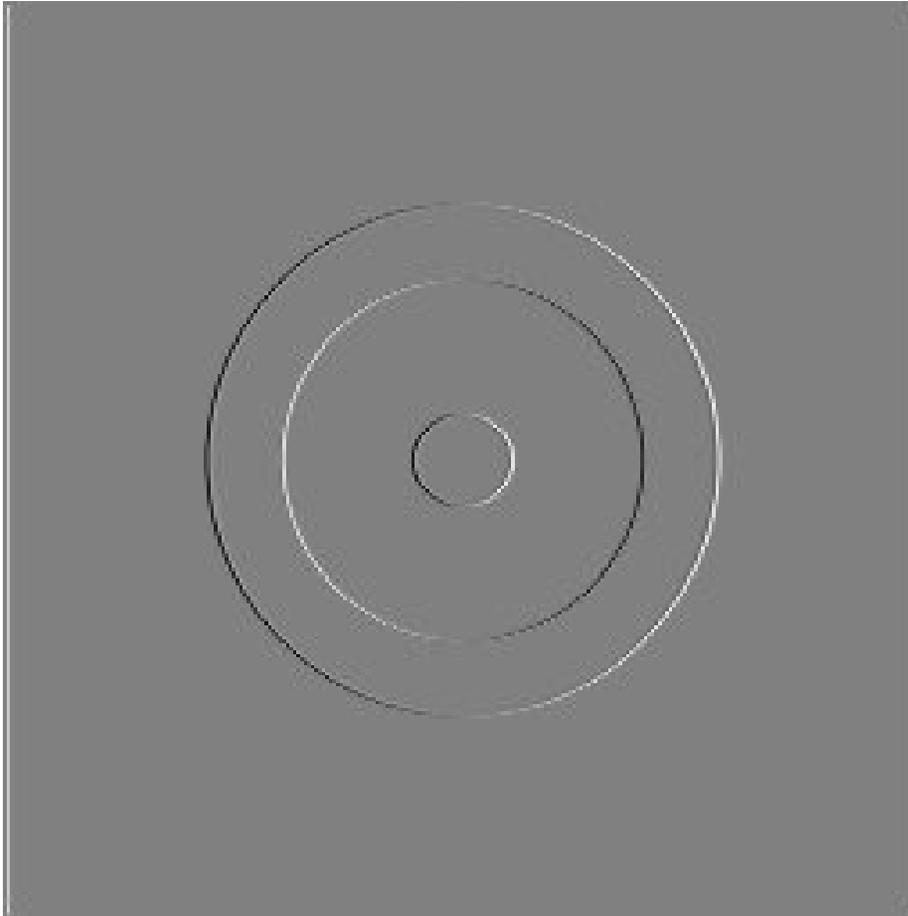
Kontur

- Approximation von Kurven(-stücken) durch Geraden(-stücke) (mit maximal erlaubten Fehler, Endpunkte sind Startpunkte,...

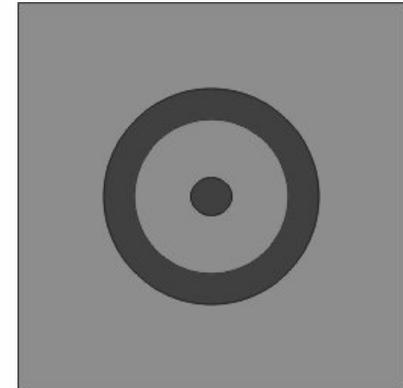


[Bässmann & Kreys 98], S. 204

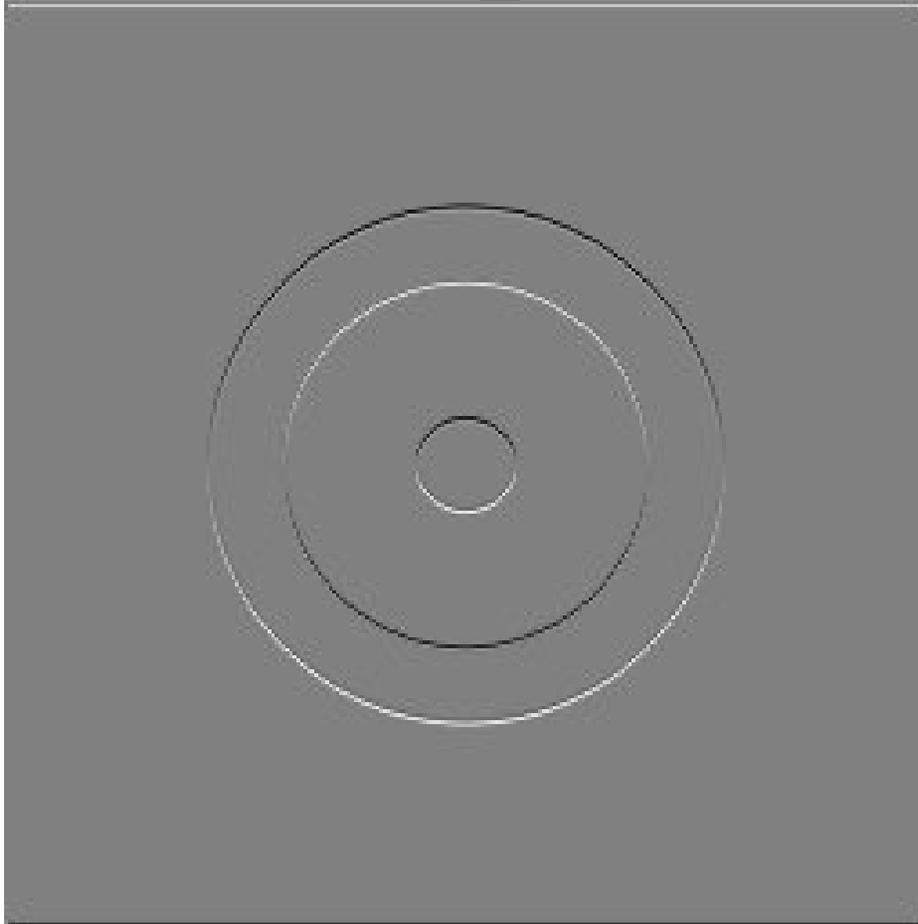
Einleitung / Diskrete Ableitung



Ableitung in
x-Richtung



Einleitung / Diskrete Ableitung



Ableitung in
y-Richtung

