

3. Fortsetzung zu Kap. 4: Bildoperationen / Globale Operationen

Die Hough-Transformation

(sprich: "haff")

P.V.C. Hough, Patentschrift von 1962

Ziel: Erkennen von Geraden

– allgemeiner: von einfachen parametrischen Kurven (Kreise, Ellipsen...)

Anwendung:

Detektion kollinearere Punkte

Konturbeschreibung

⇒ auch oft zur *Bildsegmentierung* gerechnet

Idee:

- Transformation in einen Raum, wo Geraden (bzw. andere Kurven) als Punkte dargestellt werden ⇒ kollineare Geradenstücke fallen alle auf einen Punkt
- Zählung in einer Akkumulator-Matrix
- Maxima in dieser Matrix entsprechen den Geraden im Originalbild

Geradengleichung:

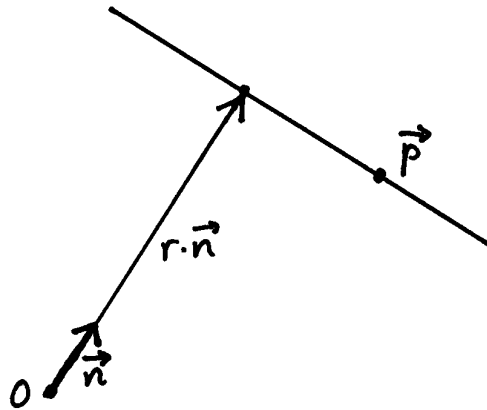
$$y = mx + b$$

- Parameter m , b (Steigung, y -Achsen-Abschnitt) wenig geeignet, da m auch ∞ werden kann

bessere Geradendarstellung: *Hessesche Normalform* (HNF)

Herleitung:

Gerade kann charakterisiert werden durch Normaleneinheitsvektor \vec{n} (steht senkrecht zur Geraden und zeigt weg von 0) und ihren Abstand r vom Nullpunkt



$\Rightarrow r \cdot \vec{n}$ liegt auf der Geraden

\Rightarrow für jeden beliebigen Punkt \vec{p} auf der Geraden gilt:

$\vec{p} - r \cdot \vec{n}$ hat die Richtung der Geraden

$$\Leftrightarrow (\vec{p} - r \cdot \vec{n}) \perp \vec{n}$$

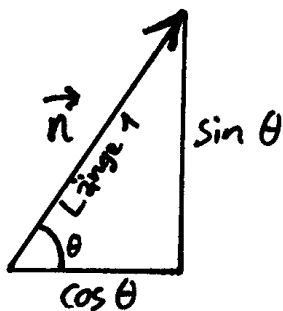
$$\Leftrightarrow (\vec{p} - r \cdot \vec{n}) \cdot \vec{n} = 0$$

$$\Leftrightarrow \vec{p} \cdot \vec{n} - r \cdot 1 = 0$$

$$\Leftrightarrow \vec{p} \cdot \vec{n} = r \quad (\text{HNF})$$

(der dicke Punkt bezeichnet das Skalarprodukt).

Statt der Angabe beider Komponenten von \vec{n} genügt der Winkel θ von \vec{n} zur x-Achse:



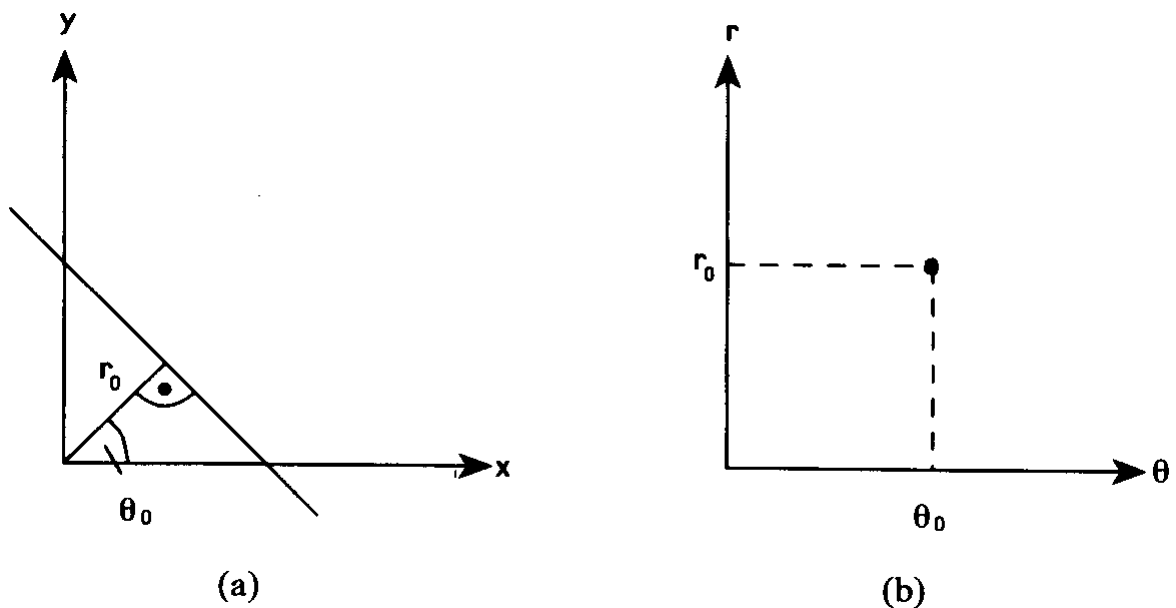
$$\vec{n} = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$$

Somit (mit $\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}$):

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

\Rightarrow Codierung der Geraden durch (r, θ) .

Somit kann jede Gerade durch einen Punkt im (r, θ) -Raum dargestellt werden:



(θ im Intervall $[0; \pi)$, r darf negativ werden.)

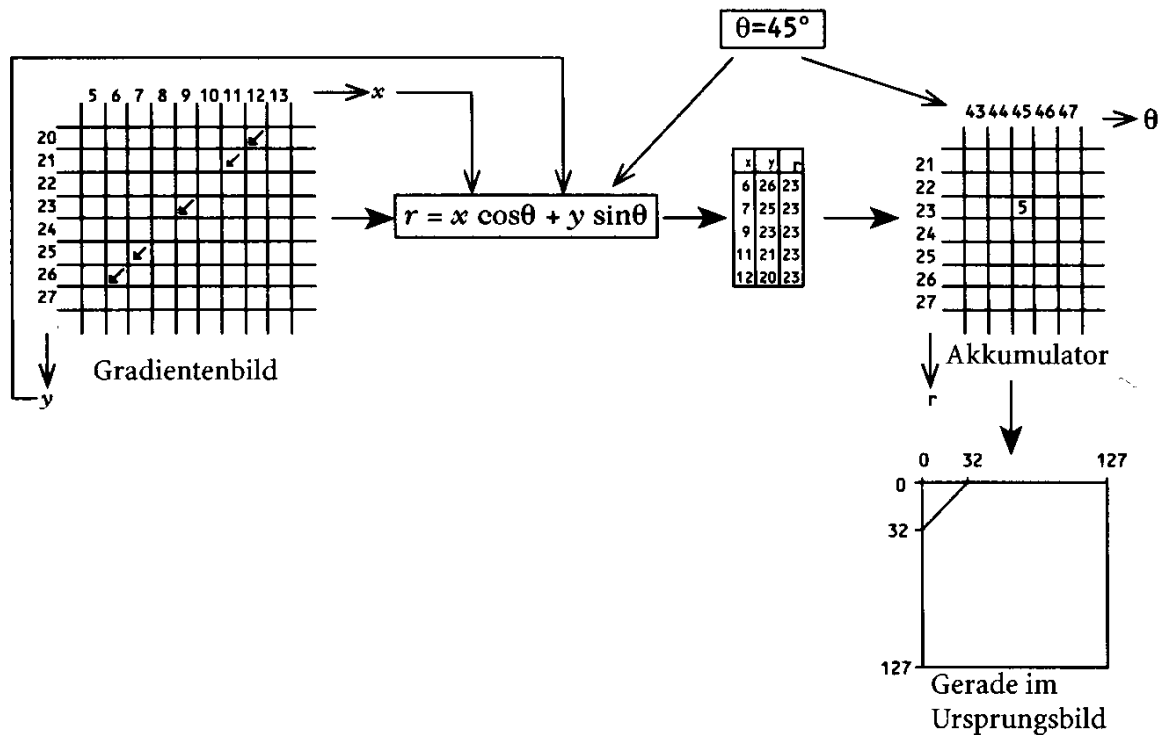
Hough-Transformation: Gerade-zu-Punkt-Transformation

- dient der Aufbereitung der Daten, um sie in einem nachfolgenden Schritt einfacher auswerten zu können

Vorangehende Schritte sind i.allg.:

- Konturextraktion durch Gradientenoperation (z.B. Sobel, Laplace)
- Verdünnung des Kantenbildes (dazu später mehr)
- Richtung orthogonal zum Gradienten stellt Kontur-Richtung dar

Beispiel: 5 Kontur-Vektoren, die auf einer Geraden liegen



Winkelinformation θ aus der Richtung des Gradienten

r aus x , y und θ

⇒ hier für alle 5 Punkte dasselbe r

⇒ Wert 5 in der Akkumulator-Tabelle

(Akkumulator diskretisiert wie ein digitales Bild)

Ergebnis der Hough-Transformation: gefüllter Akkumulator

nächste Schritte:

evtl. "Säubern" des Akkumulators (Beseitigung von Störeinflüssen: etwa Zusammenfassen von Clustern)

Analyse des Akkumulators (Maxima, Überschreitung von Schwellenwerten)

Rücktransformation der gefundenen "kritischen Punkte"

(Maxima) ins Originalbild: dort *Geraden*

Problem: man will meist Geradenstücke haben (Information über Anfangs- und Endpunkte), keine kompletten Geraden

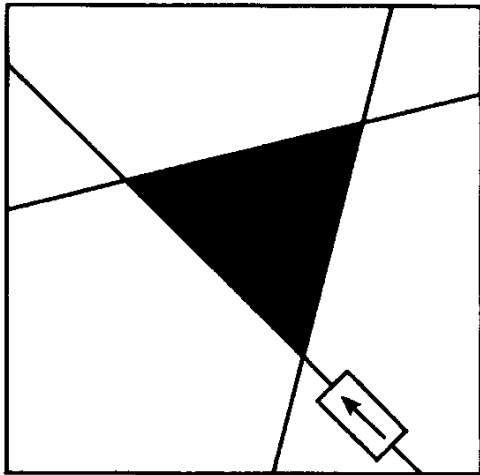
Verwendung der kompletten Geraden als "Wegweiser" durch das Originalbild:

z.B. dort morphologischen Filter entlanglaufen lassen

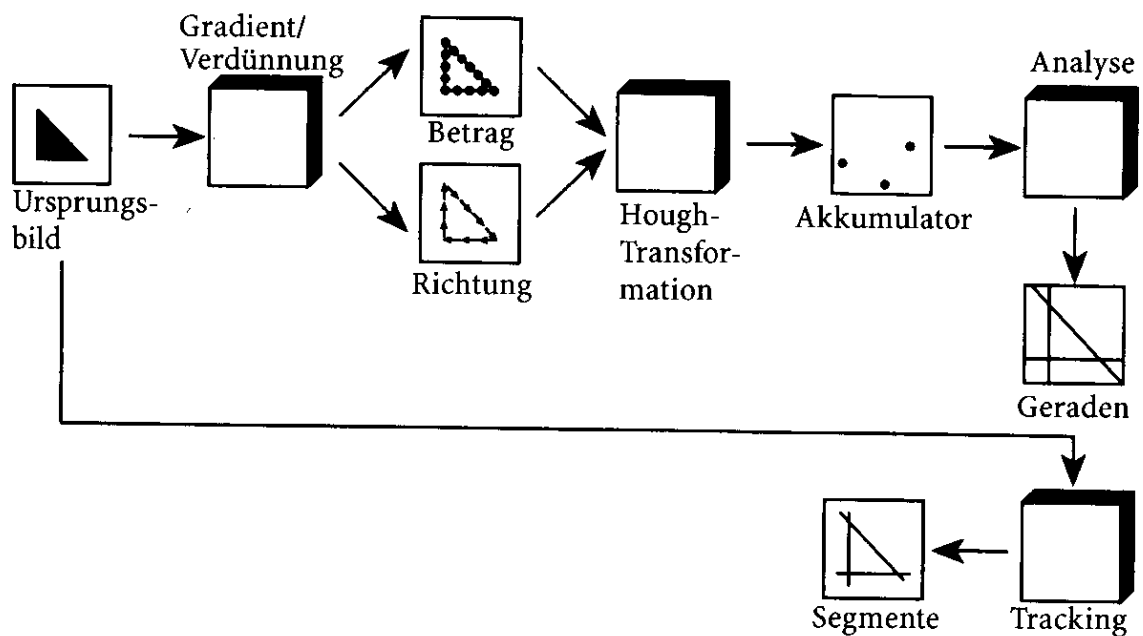
Vergleich der Pixel rechts und links der Geraden

wenn vorgegebene Grauwertdifferenz überschritten wird:

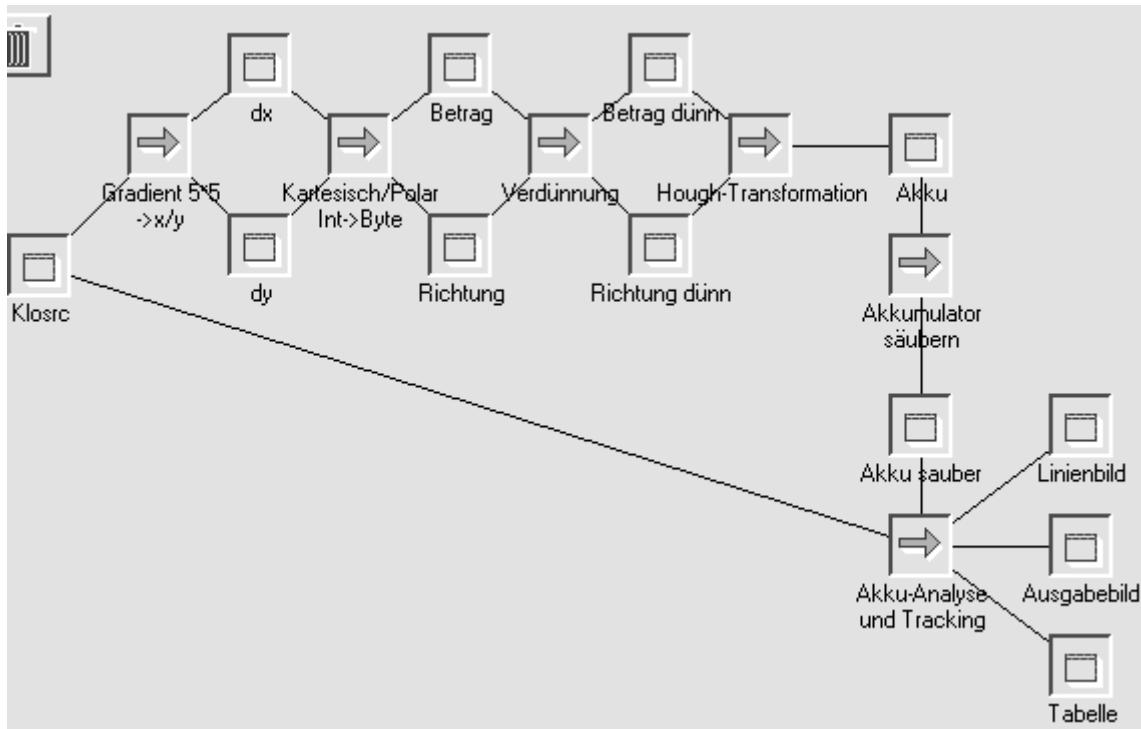
Filtermaske befindet sich auf einer Objektkontur



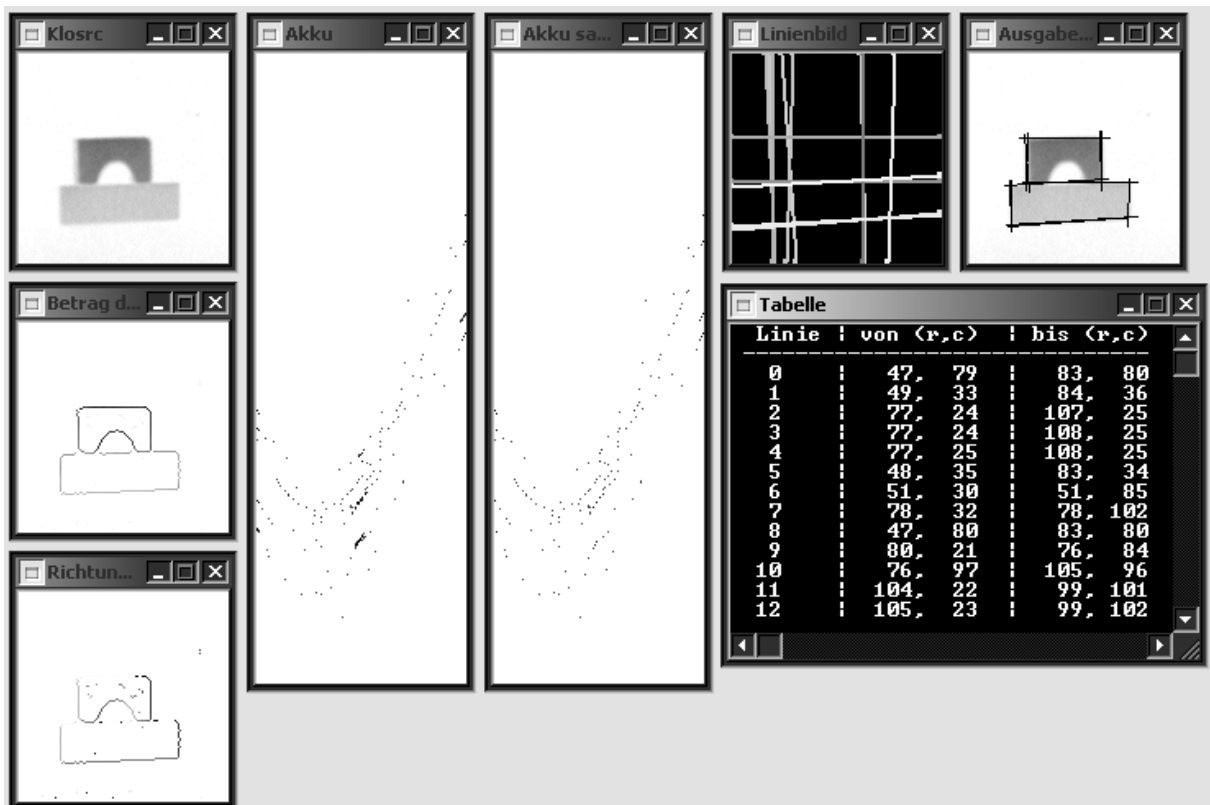
Kompletter Arbeitsgang (aus Bässmann & Kreys 1998):



Beispiel mit AdOculos-Software (Setup HOUGH.SET):



Ergebnisse (invertiert):



Original
Kantenbild
Richtungsbild

Akkumulator

Akkumulator
(geclustert)

Geraden

Geradensegmente
im Originalbild
tabellarische Darstellung d. Seg.

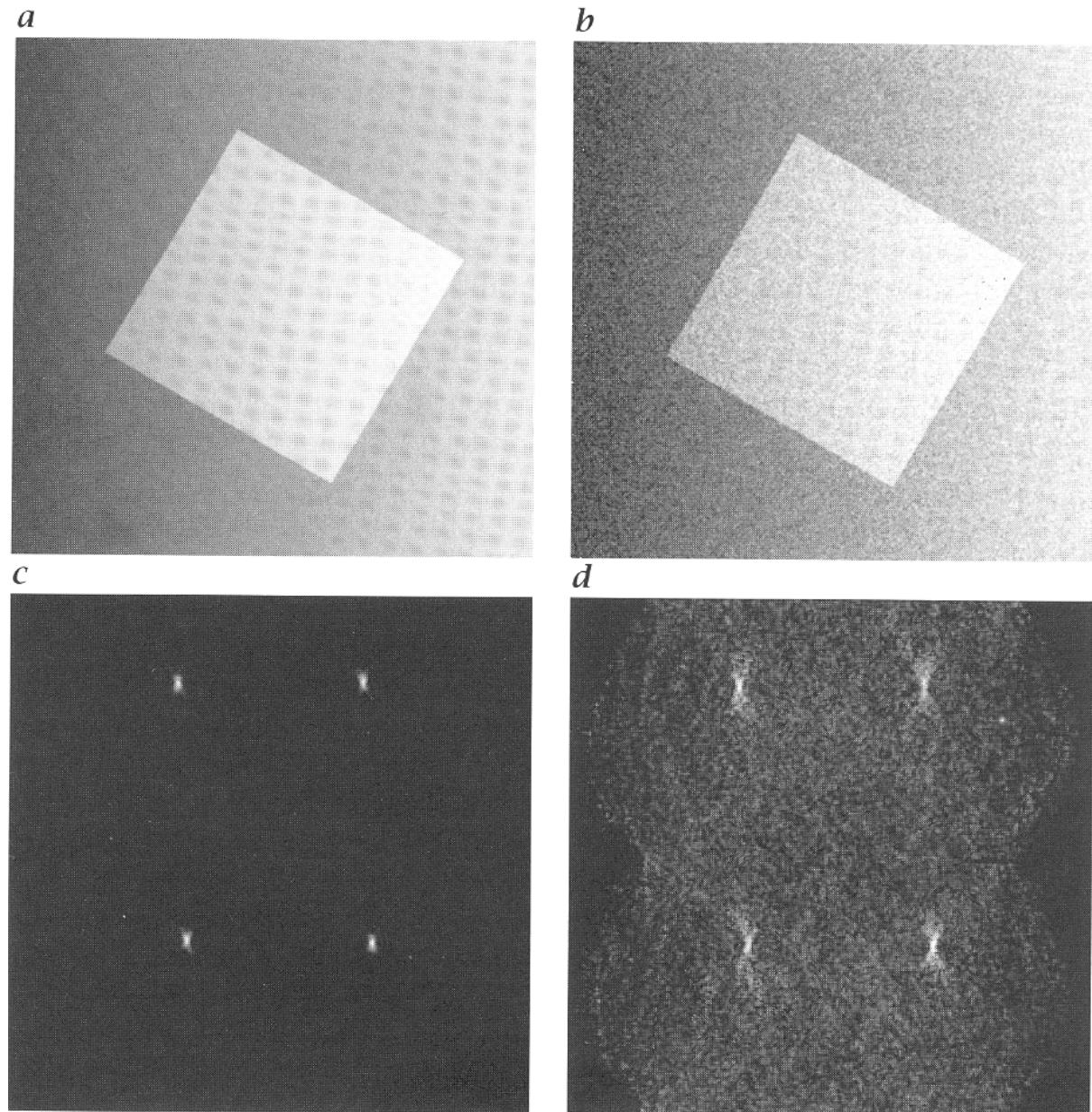
Linie	von (r,c)	bis (r,c)
0	47, 79	83, 80
1	49, 33	84, 36
2	77, 24	107, 25
3	77, 24	108, 25
4	77, 25	108, 25
5	48, 35	83, 34
6	51, 30	51, 85
7	78, 32	78, 102
8	47, 80	83, 80
9	80, 21	76, 84
10	76, 97	105, 96
11	104, 22	99, 101
12	105, 23	99, 102

- Vorteil: Geraden werden auch erkannt, wenn sie verrauscht sind oder Unterbrechungen aufweisen.

anderes Beispiel (aus Jähne 2002):

a, b: ungleichmäßig beleuchtete, verrauschte Quadrate

c, d: entsprechende Hough-Modellräume ((r, θ) -Ebene) zu a und b



Beachte: parallele Geraden stehen in derselben Spalte des Akkumulators \Rightarrow Verfahren auch zum Erkennen von Parallelität verwendbar.

Probleme:

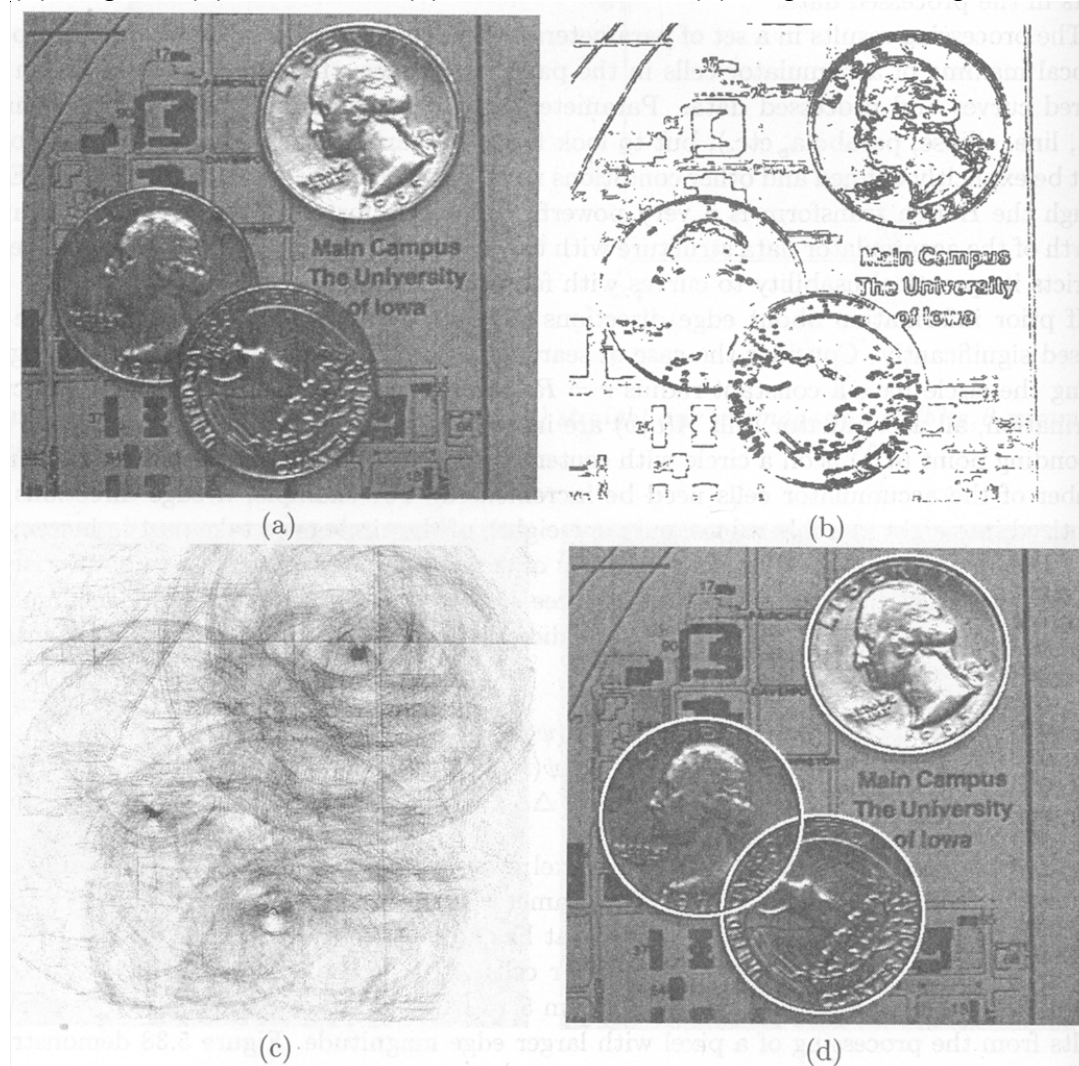
- in dieser Form nur zur Detektion von geraden Konturen geeignet (– aber Verallgemeinerung möglich!)
- Clustering im Akkumulator kann die "falschen" Geraden entfernen
- großer Speicherbedarf des Akkumulators
- Tracking der Geraden ist rechenaufwändig

verschiedene Verbesserungen des Verfahrens wurden vorgeschlagen

Verallgemeinerung:

z.B. auf Kreise (aus Sonka et al. 1999):

(a) Original, (b) Kantenbild, (c) Parameterraum, (d) Orig. m. überlagerten Kreisen



verallgemeinerte Hough-Transformation in ähnlicher Weise möglich für Ellipsen, Parabeln...

weitere Verallgemeinerung:

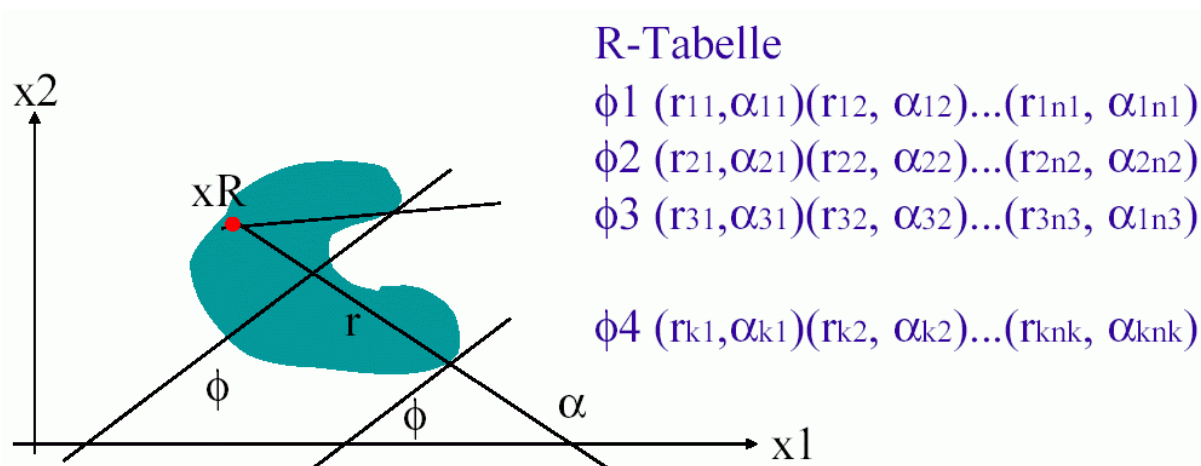
für beliebige (nicht-parametrische), aber fest vorgegebene Form, die im Bild aufgefunden werden soll

⇒

generalized Hough transform nach Ballard (s. Sonka et al. 1999):

- in der zu suchenden Form (Kontur) wird ein Referenzpunkt R festgelegt
- in einem preprocessing-Schritt wird eine R-table (Referenz-tabelle) angelegt: für jeden von R ausgehenden Strahl wird der Schnittpunkt mit der Konturlinie der Form bestimmt
- im Schnittpunkt wird die Kontur-Richtung und der Abstand von R bestimmt und in die Tabelle eingetragen
- die Tabelle wird nach den Kontur-Richtungen (Winkeln) sortiert
- für jeden Kontur-Punkt (mit Richtungsinformation) *im Bild* wird nun anhand der Tabelle ein Kandidaten-Punkt für R (unter der Hypothese, dass diese Kontur an dieser Stelle durch die gesuchte Form erzeugt wurde) errechnet (evtl. auch mehrere Kandidaten-Punkte)
- die Koordinatenpaare der Kandidaten-Punkte bilden den Hough space (Parameterraum; Akkumulator)
- bei Punkten mit hohem Akkumulator-Eintrag ist tatsächlich die gesuchte Form lokalisiert
- wenn Rotationen und Skalierungen der gesuchten Form zugelassen sind, erhöht sich die Dimension des Hough space von 2 auf 4

zum preprocessing der Konturpunkte (Erstellung der R-table):



Beachte:

Nach der Hough-Transformation (einfach oder verallgemeinert) ist die Aufgabe der automatischen Detektion vorgegebener Formen noch nicht gelöst.

- Detektion dunkler Stellen im Akkumulator-Raum zu leisten
- evtl. Zusatzinformation aus dem Bild erforderlich (z.B. Endpunkte von Geradenstücken)

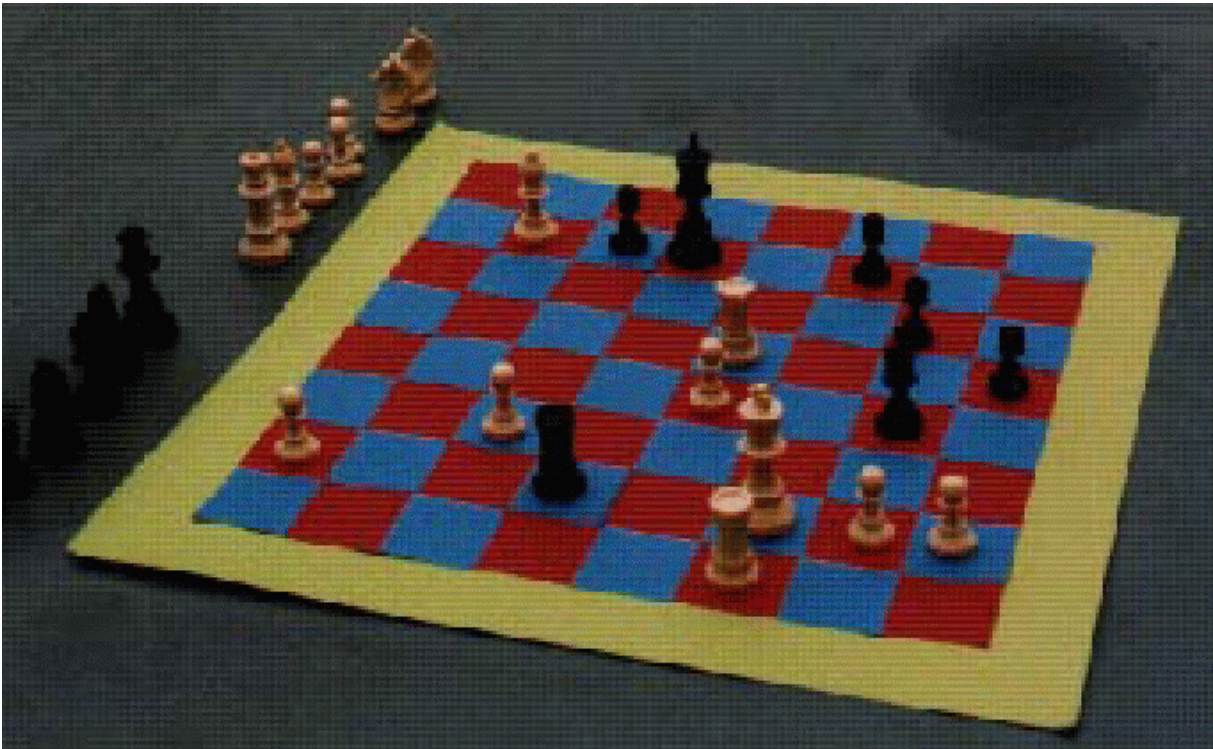
je komplizierter die Form, desto mehr Parameter werden benötigt \Rightarrow Akkumulator-Raum wird hochdimensional \Rightarrow Verfahren wird sehr speicher- und rechenintensiv

Einsatz der Hough-Transformation:

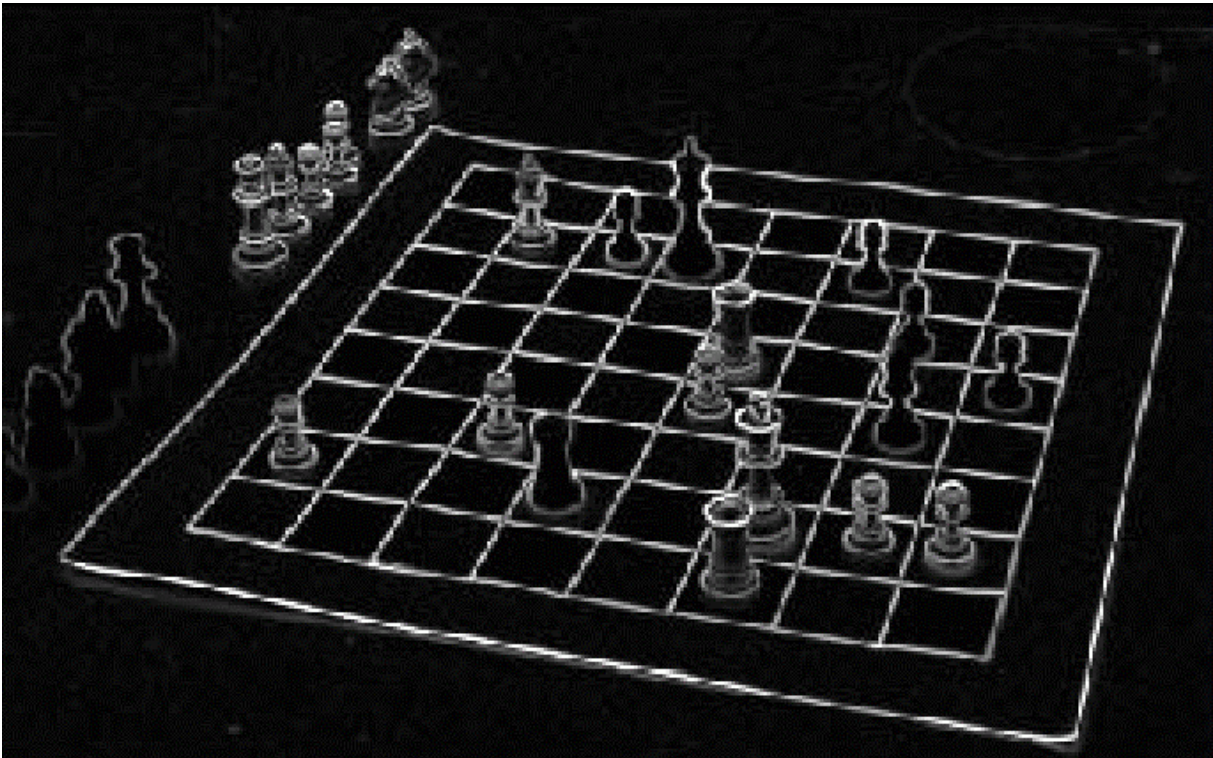
besonders in industriellen Anwendungen

- einfache Formen werden gesucht
- viele Konturen sind Geradenstücke oder Kreisbögen
- Parameter können oft eingeschränkt werden

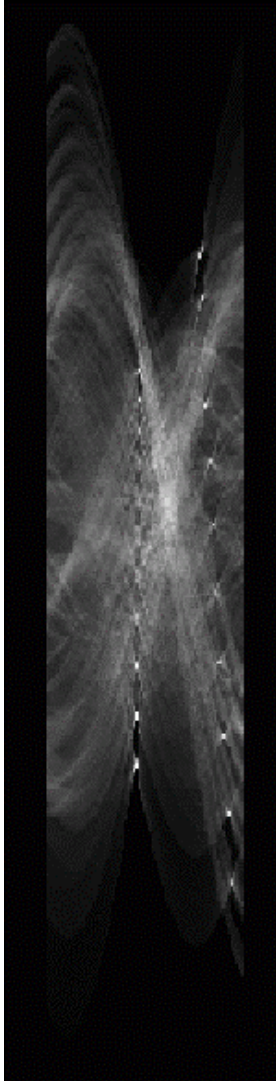
Beispiel Schachbrett (aus Beichel 2002):



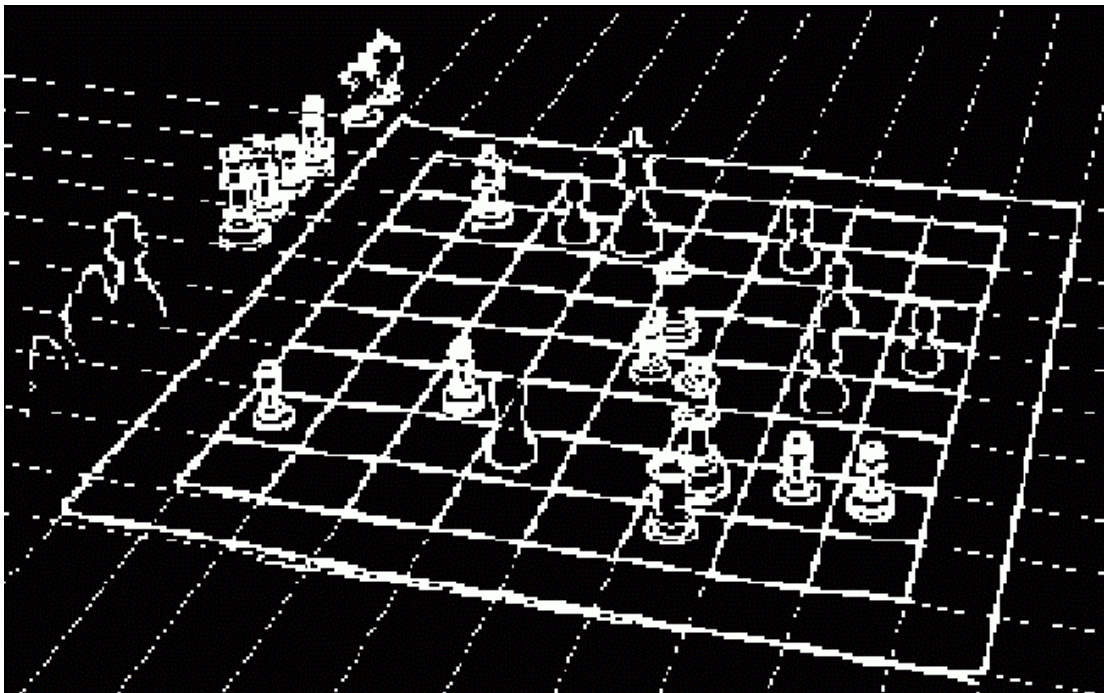
Original-Aufnahme



Kantenbild

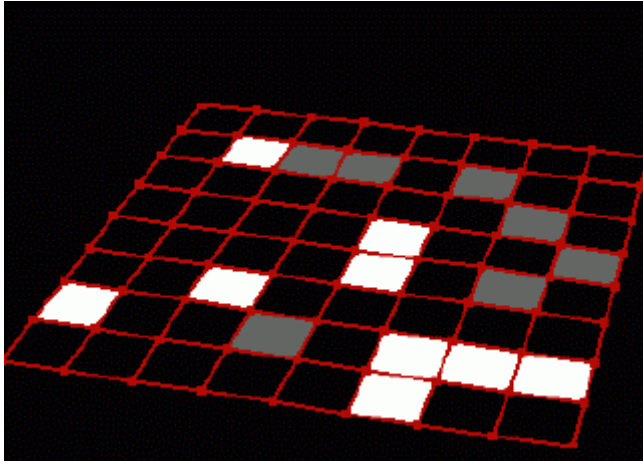


Akkumulator (Hough-Space)



Ergebnis der Hough-Transformation (Geraden)

weiterverwendbar für formale Verarbeitung:



5. Segmentierung

Erinnerung an die "Bildanalyse- (Bildverarbeitungs-) Pipeline":

BV-System

- Bilderfassung
 - Kamera, Satelliten, NMR/CT,...
- Vorverarbeitung
 - Rauschunterdrückung, Kontrastverstärkung,...
- Segmentierung
 - Konturen, Flächen,...
- Merkmalsextraktion
 - Kettencode, Umfang, Schwerpunkt, Farbe,...
- Klassifikation
 - Objekterkennung, Einteilung in Klassen, ...

Segmentierung:

Vorgang, das Bild in *sinnvolle Bildteile* aufzuteilen (es zu *segmentieren*)

"Fernziel": Objekterkennung; Trennung von Objekt und Hintergrund

Annahme: Objektoberflächen erscheinen in der Szene weitgehend *homogen* und entsprechen daher homogenen Flächen im Bild

⇒ Versuche, Bild in *homogene Bereiche* aufzuteilen

unterschiedliche Homogenitätskriterien
(Grauwert, Farbe, Textur...)

- Aufteilung eines Bildes in Teile, die Objekten bzw. Hintergrund entsprechen
- **Komplette Segmentierung:** Aufteilung in nichtüberlappende Gebiete, die Objekten entsprechen -> Wissen erforderlich
- **Teilsegmentierung:** Regionen korrespondieren nicht direkt mit Objekten -> Vorverarbeitungsschritt

Teilaufgaben der Segmentierung:

- Trennung der zu untersuchenden Objekte von den übrigen Bildstrukturen
- Zerlegung des Bildes in zu interessierende Strukturen (Objekte) und nicht zu interessierende Strukturen (Hintergrund)
- Trennung von sich berührenden oder überlappenden Objekten voneinander
- Zerlegung in Teilobjekte

Mathematische Definition der *kompletten Segmentierung*:

Partition der Menge P aller Bildpunkte in Regionen (Teilmengen) P_i ($i = 1, \dots, n$), so dass gilt:

1. $\bigcup_{i=1}^n P_i = P$
2. P_i ($i = 1, \dots, n$) ist zusammenhängend
3. für $i \neq j$ gilt $P_i \cap P_j = \emptyset$
4. $H(P_i) = \text{TRUE}$ für alle $i = 1, \dots, n$
5. $H(P_i \cup P_j) = \text{FALSE}$ für $i \neq j$, wenn P_i und P_j benachbart sind. (Maximalität der Segmentierung)

Dabei ist H ein (fest gewähltes) Homogenitätskriterium.

Ergebnis der Segmentierung:

Regionenbild

- jedes Pixel bekommt die entsprechende Regionen- bzw. Segment-Nummer als Label zugeordnet

oft ist komplette Segmentierung nicht zu erreichen oder unpraktikabel

⇒ man schwächt ab: Forderungen 1 und 3 werden fallengelassen

⇒ Begriffe "token", "tokenset"

Segmentierungsverfahren:

- *punktbasierte* (pixelbasierte) Verfahren: Klassifikation auf Ebene einzelner Pixel (Forderung 2, Zusammenhang, wird vernachlässigt) – eigentlich noch keine echte Segmentierung
- *kantenbasierte* Segmentierung (Kantendetektion mit Diskontinuitätskriterium oder mit Hough-Transf., darauf aufbauend Kontur-Rekonstruktion, Abgrenzung von Regionen durch ihre Konturen)
- *regionenbasierte* (bereichsbasierte) Verfahren: Partitionierung der Pixel unter Einbeziehung der Nachbarschaft
- hybride Verfahren

Punktbasierte Segmentierung

einfachstes Verfahren:

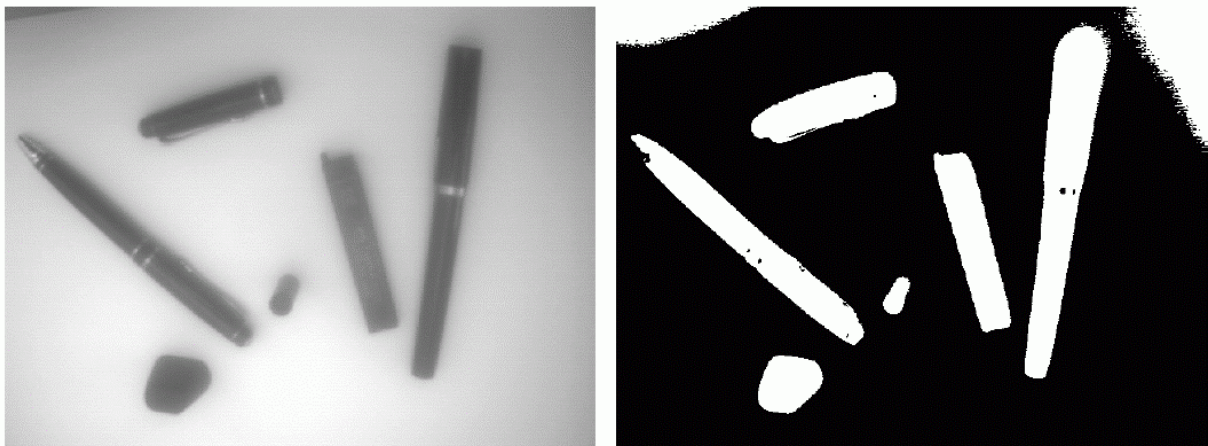
Thresholding (Schwellenwert-Setzung)

Binarisierung des Bildes mit Hilfe eines Grauwert- oder Farbintensitäts-Schwellenwertes (vgl. Vorverarbeitung, Punktoperationen)

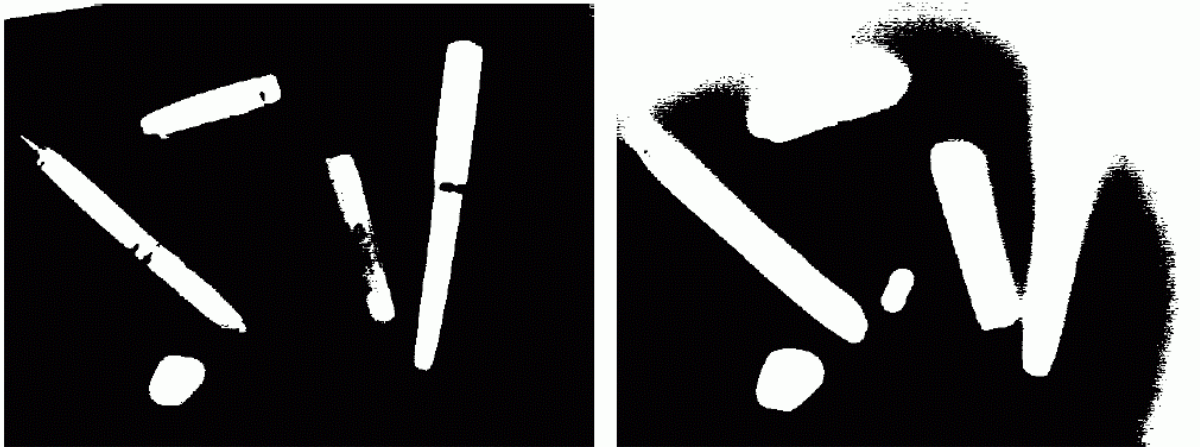
- Grauwert-**Schwellwertsetzung** ist die einfachste Art der Segmentierung
- Nicht rechenaufwendig - **schnell**
- Wird häufig verwendet für **einfache Anwendungen**
- Transformation eines Bildes f in ein segmentiertes Binärbild g :

$$g(i, j) = \begin{cases} 1 & \text{for } f(i, j) \geq T \\ 0 & \text{for } f(i, j) < T \end{cases}$$

funktioniert nur, wenn sich vereinzelt, weitgehend homogene Objekte von einem homogenen Hintergrund abzeichnen



Probleme: Unter- und Übersegmentierung



- Einzelner, globaler Schwellwert reicht meistens nicht aus zur Segmentierung

- **Globaler** Schwellwert:

$$T = T(f)$$

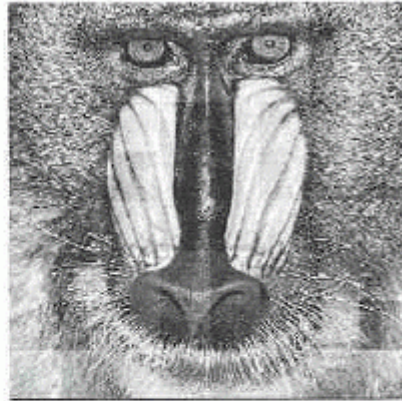
- **Lokaler** Schwellwert:

$$T = T(f, f_c)$$

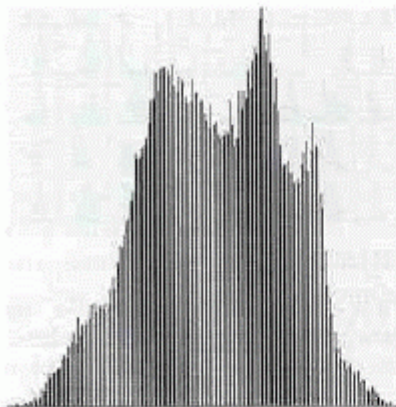
z.B. durch Aufteilung in Teilbilder



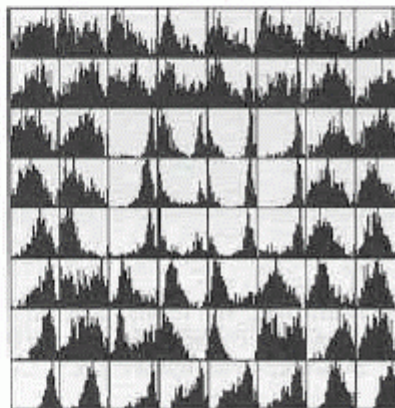
(A)



(B)



(C)

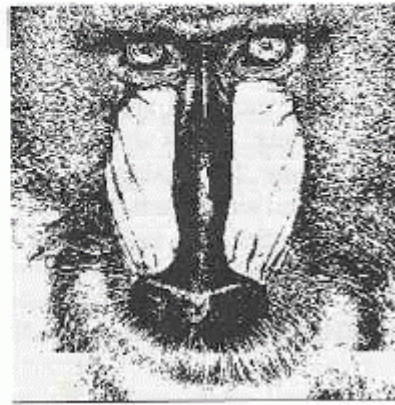


(D)



(E)

globaler Schwellwert



(F)

lokal (Teilbilder)

unten jeweils das gewonnene Binärbild (aus Hermes 2001)

Modifikationen:

- Multiple Schwellwerte:**

$$g(i, j) = \begin{cases} 1 & \text{for } f(i, j) \in D_1 \\ 2 & \text{for } f(i, j) \in D_2 \\ 3 & \text{for } f(i, j) \in D_3 \\ n & \text{for } f(i, j) \in D_n \\ 0 & \text{otherwise} \end{cases}$$
- Semi-Thresholding**

$$g(i, j) = \begin{cases} f(i, j) & \text{for } f(i, j) \geq T \\ 0 & \text{for } f(i, j) < T \end{cases}$$

Problem: Schwellwertbestimmung

Einbeziehung von Vorwissen:

- Eigenschaft** nach der Segmentierung **bekannt:** diese muß durch den Schwellwert eingehalten werden.
 Bsp: Gedruckter Text nimmt 1/p Fläche ein (**p-tile thresholding**)
- => Schwellwert entsprechend wählen (mit Hilfe des Histogramms)

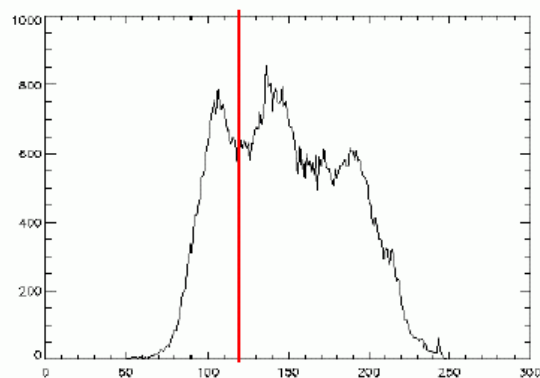
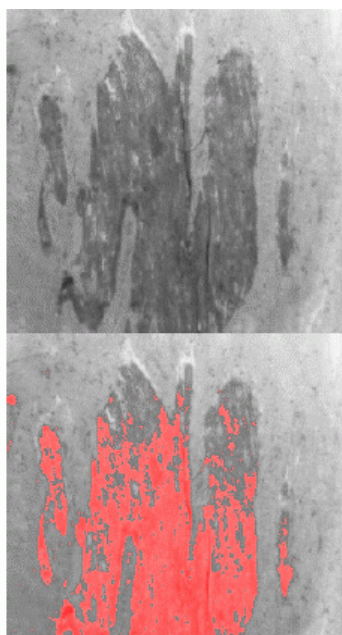
Beispiel (aus Beichel 2002):

0	1	2	3	4	5	6	7	8	9	0	1	2	3	
0	1	2	3	4	5	6	7	8	9	0	6	5	3	
0	1	2	3	4	5	6	7	8	9	1	9	6	1	
0	1	2	3	4	5	6	7	8	9	7	2	3	9	
0	1	2	3	4	5	6	7	8	9	7	8	5	4	
0	1	2	3	4	5	6	7	8	9	8	4	7	2	
0	1	2	3	4	5	6	7	8	9	7	3	5	6	
0	1	2	3	4	5	6	7	8	9	3	5	7	9	
0	1	2	3	4	5	6	7	8	9	4	4	1	8	
0	1	2	3	4	5	6	7	8	9	1	5	6	9	

10 % - Quantil - Segmentierung:

0	1	2	3	4	5	6	7	8	9	0	1	2	3
0	1	2	3	4	5	6	7	8	9	0	6	5	3
0	1	2	3	4	5	6	7	8	9	1	9	6	1
0	1	2	3	4	5	6	7	8	9	7	2	3	9
0	1	2	3	4	5	6	7	8	9	7	8	5	4
0	1	2	3	4	5	6	7	8	9	8	4	7	2
0	1	2	3	4	5	6	7	8	9	7	3	5	6
0	1	2	2	4	5	6	7	8	9	3	5	7	9
0	1	2	3	4	5	6	4	8	9	4	4	1	8
0	1	2	3	4	5	6	4	8	9	1	3	6	9

oft verwendet zur Schwellwertbestimmung:
Annahme eines bimodalen Histogramms



Max1: 786 bei 106

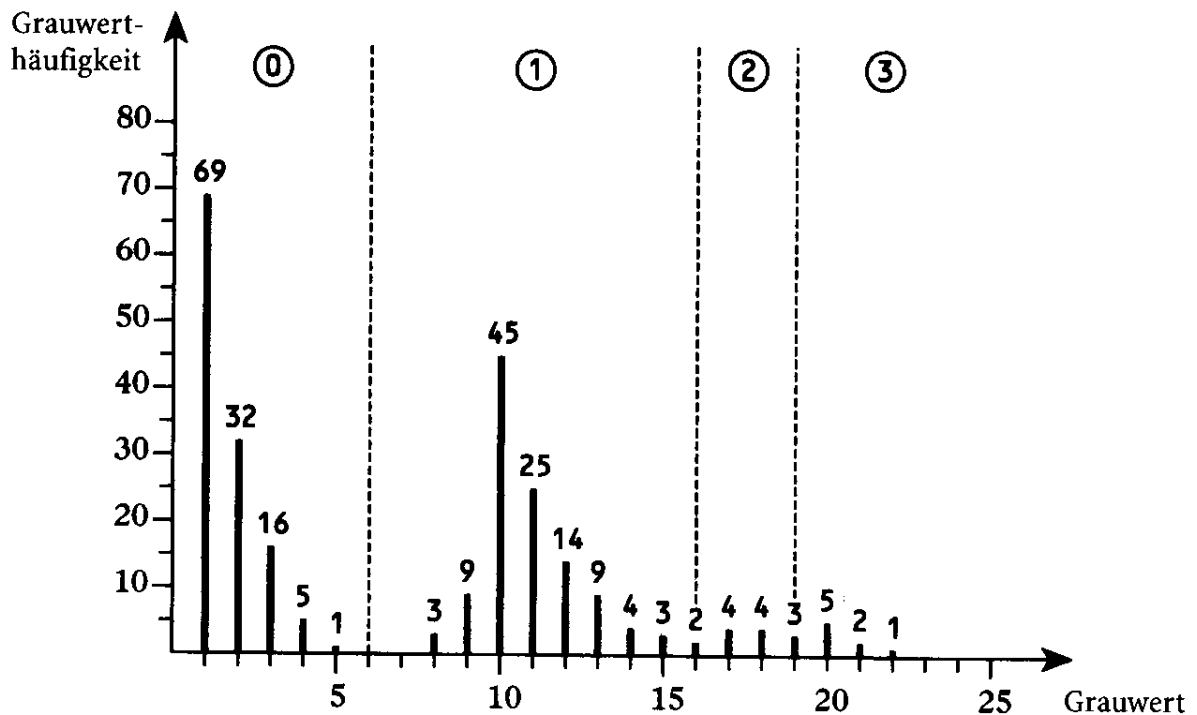
Max2: 858 bei 136

Schwellwert bei $(106+136)/2=121$

Verallgemeinerung:

multimodales Histogramm \Rightarrow mehrere Schwellenwerte, mehr als 2 Labels im Bild

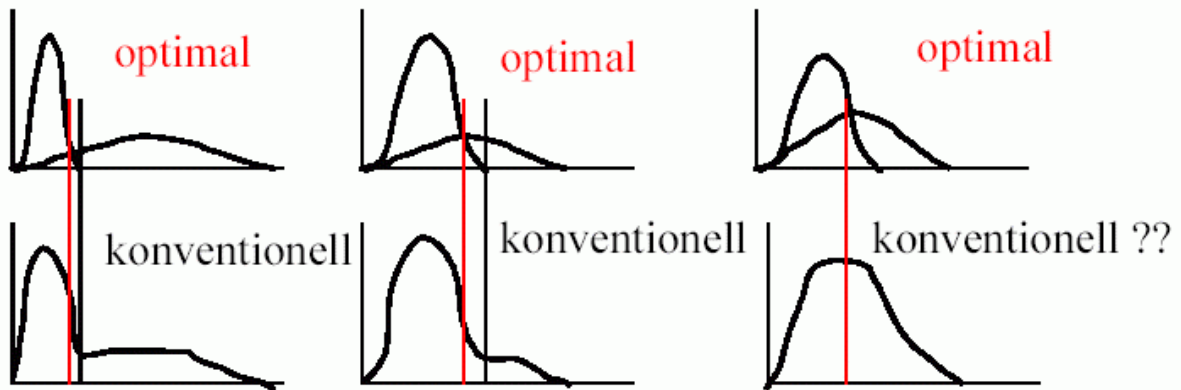
Probleme bei Verwendung der Histogrammanalyse zur Segmentierung:



- schwach ausgeprägte Maxima können trotzdem für die Histogrammstruktur bedeutsam sein (z.B. oben bei 20)
 - Abhilfe: nichtlineare Reskalierung des Histogramms (log; sqrt...)
- nicht-signifikante lokale Minima führen zu überflüssigen, störenden Schwellenwerten (z.B. oben zwischen Bereich 2 und 3)
 - Abhilfe: Glättung des Histogramms

Optimierung des Schwellwertes:

- **Approximation von Verteilungen** ans Histogramm - Parameterschätzung
- Berechnung der exakten Schnittpunkte



Verfahren der iterativen optimalen Schwellwertbestimmung:

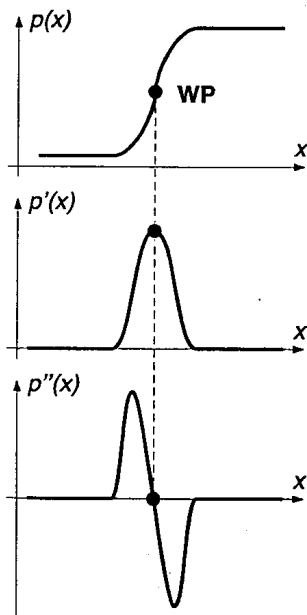
- **Initialisierung:** 4 Eckpunkte sind Hintergrund, Rest ist Objekt
- **Iteration:** mittleren Grauwert über Objekte und Hintergrund berechnen
Schwellwert in der Mitte setzen und neu segmentieren
- **solange bis** sich der Schwellwert nicht mehr ändert
- 4-10 Iterationen meistens ausreichend
- Funktioniert auch wenn verschiedene Objekte vor einheitlichem Hintergrund vorkommen

Kantenbasierte Segmentierung

Grundlage: Kantendetektion durch lineare Filterung und / oder morphologische Bildoperationen
- siehe Kapitel 4 -

"It is hard to over-emphasize the importance of edge-detection in image understanding" (Nalwa & Binford, zit. bei Pinz 1994)

Grauwertsprung:



Detektion durch:

- erste Ableitung (diskret: $(-1; 1)$ -Maske; Roberts; Prewitt; Sobel...)
- zweite Ableitung (Laplace)
- morphologische Operatoren ...

aber das Ergebnis der Kantendetektion kann in der Regel nicht direkt zur Segmentierung verwendet werden

- Linien zu dick
- Lücken in der Kontur
- Rauschen
- Information immer noch ausschließlich auf Pixelebene repräsentiert

⇒ Nachbearbeitung notwendig:

Kantenpixel zu Ketten verbinden, die Rändern von Regionen entsprechen

Probleme:

- Kanten im Bild, die keiner Regionengrenze entsprechen;
- Regionengrenzen, an denen keine Kantenpixel auftreten

Schwellenwertbildung im Kantenbild (= im Ergebnisbild von Sobel etc.):

- im Kantenbild meist kaum Nullen
- kleine Werte entsprechen nicht-signifikanten Grauwert-änderungen, sind uninteressant für Segmentierung
- Schwellenwertsetzung entfernt diese unwichtigen Kanten
- globaler Schwellenwert nicht immer ausreichend: *p*-tile thresholding; lokale Schwellenwertbildung

oft ist Binarisierung des Kantenbildes gar nicht sinnvoll: Betrag und Richtung des Gradienten können wertvolle Informationen für die Segmentierung liefern!

Begriff "edge!" (edge element, Kantenelement):

Punkt (Pixel) mit Information über Kantenrichtung und Betrag des Gradienten an diesem Punkt

nützlich ist oft a-priori-Wissen

Beispiel 1: Hough-Transformation (bekannte Form wird gesucht: Geradenstück, Kreis...), siehe Kap. 4

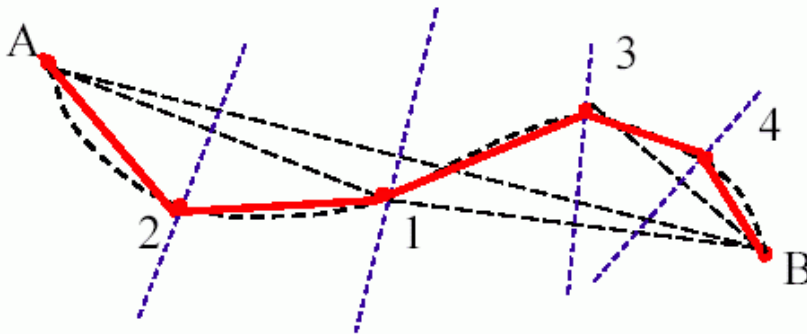
Beispiel 2: Randbestimmung bei bekannter Randpositionsinformation

- Information über Position oder Form des Randes ausnutzen
- Bekannt z.B. aus niedrigerer Auflösung



- z.B. Suche nach signifikanten Kantenpixeln quer zur vermuteten Grenze, Approximation der Kurve durch diese Punkte

- **Iterative Unterteilung** der Grenze und Suche nach Kantenpixel quer zur Verbindungslinie



- Funktioniert bei **geringem Rauschen** und **relativ geraden Kanten**.

Begriffsklärung:

"Kante": Grauwertsprung im Bild

"Kontur": ein am Rand eines Objektes liegender geschlossener Weg

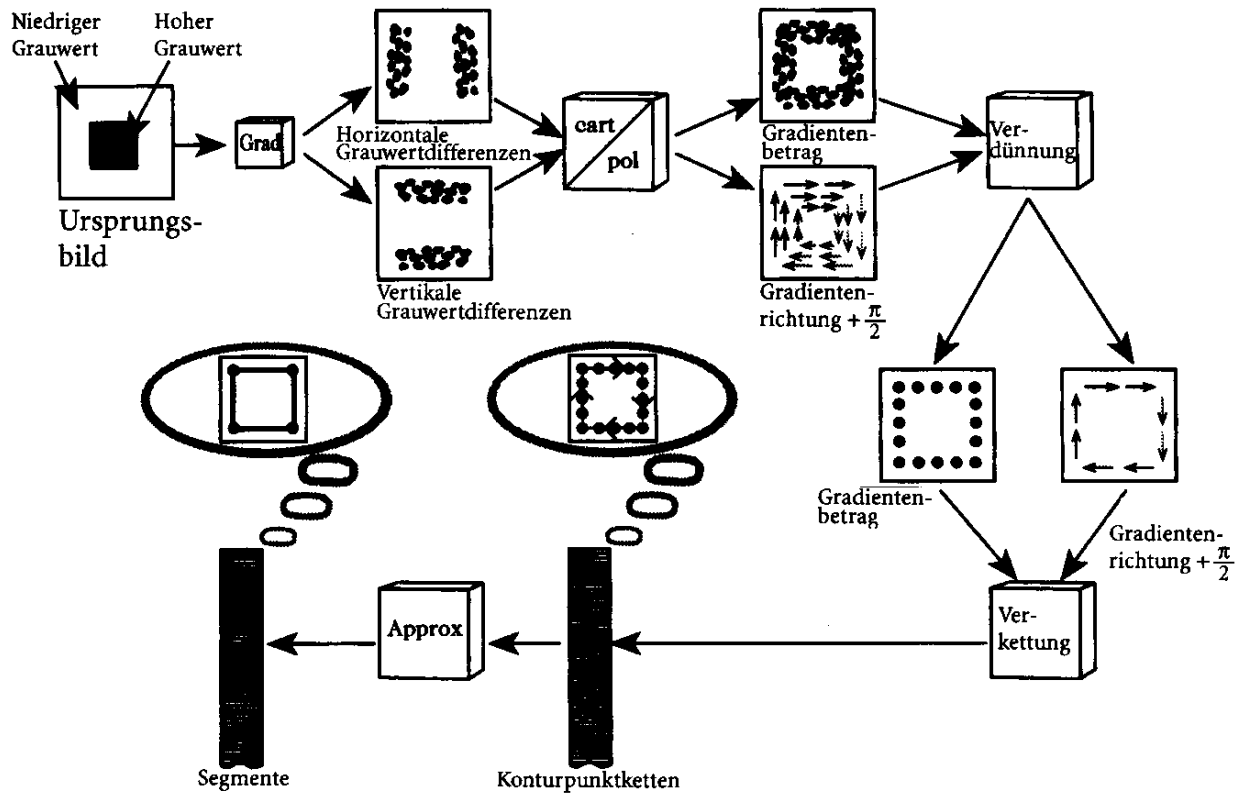
ein Objektpunkt liegt genau dann am Rand des Objektes, wenn mindestens einer seiner (Vierer- oder Achter-) Nachbarn ein Hintergrundpunkt ist

⇒ Konturfindung impliziert Segmentierung

Oft genügt es, die Kontur eines Objektes zu kennen (wenn man sich für Größe, Form und Lage interessiert, aber nicht für die Innenstruktur des Objektes)

⇒ gesucht: Verarbeitungskette vom Kantenbild zu einer Kontur-Repräsentation (Konturfolgeverfahren)

Beispiel eines Konturfolgeverfahrens für beliebige Konturformen, die durch Geradenstücke approximiert werden:



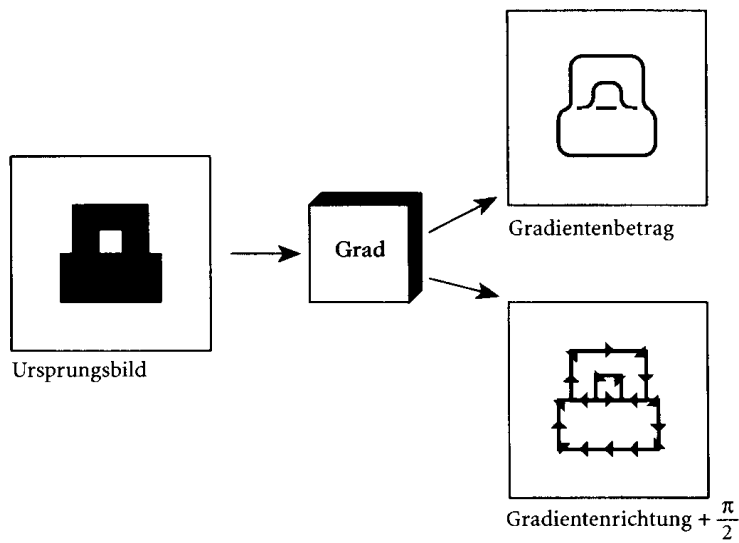
(aus Bässmann & Kreys 1998)

Ergebnis: Liste von Geradenstücken (z.B. gegeben durch Anfangs- und Endpunktkoordinaten), d.h. eine *strukturelle* Beschreibung der Regionen

Einzelschritte bei diesem Verfahren:

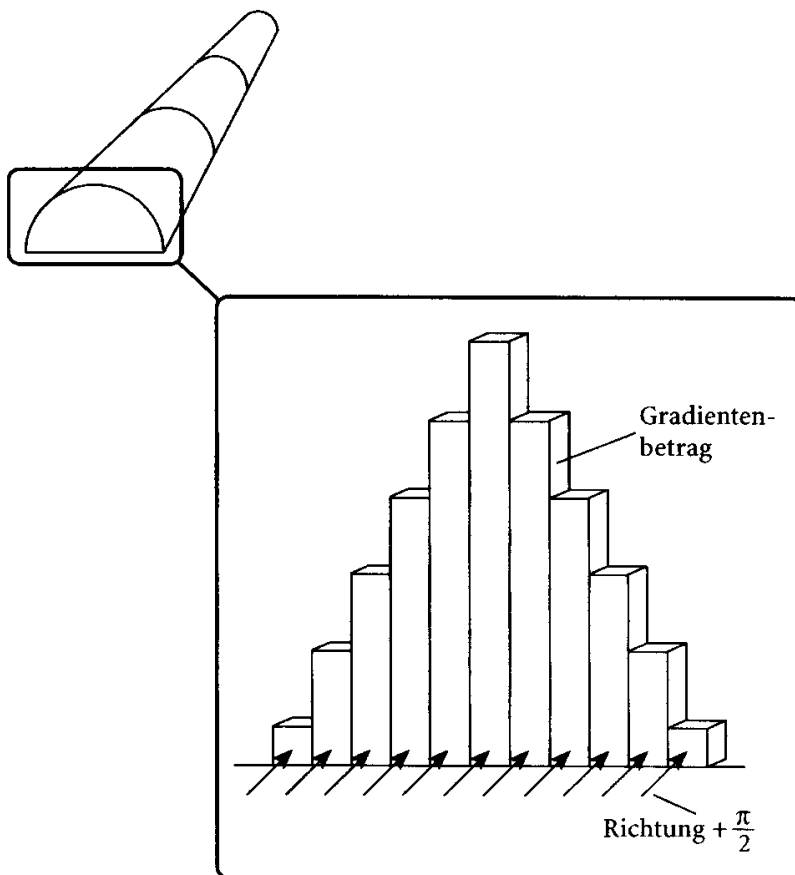
- Kantendetektion
- Verdünnung
- Verkettung
- Konturapproximation

Kantendetektion (schon behandelt):

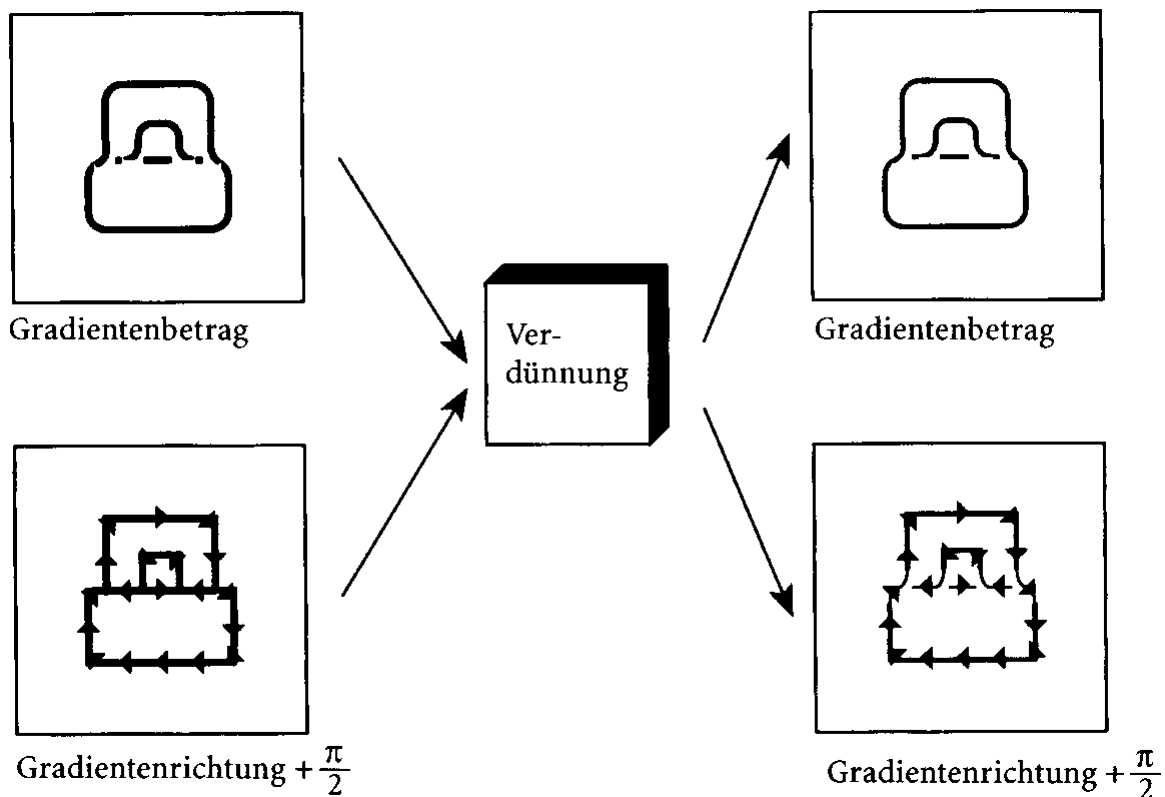


Verdünnung der Konturen:

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



Ziel der Verdünnung: Verbesserung der "verschmierten" Kontur, so dass lediglich 1 Pixel dünne Linien verbleiben



Algorithmus:

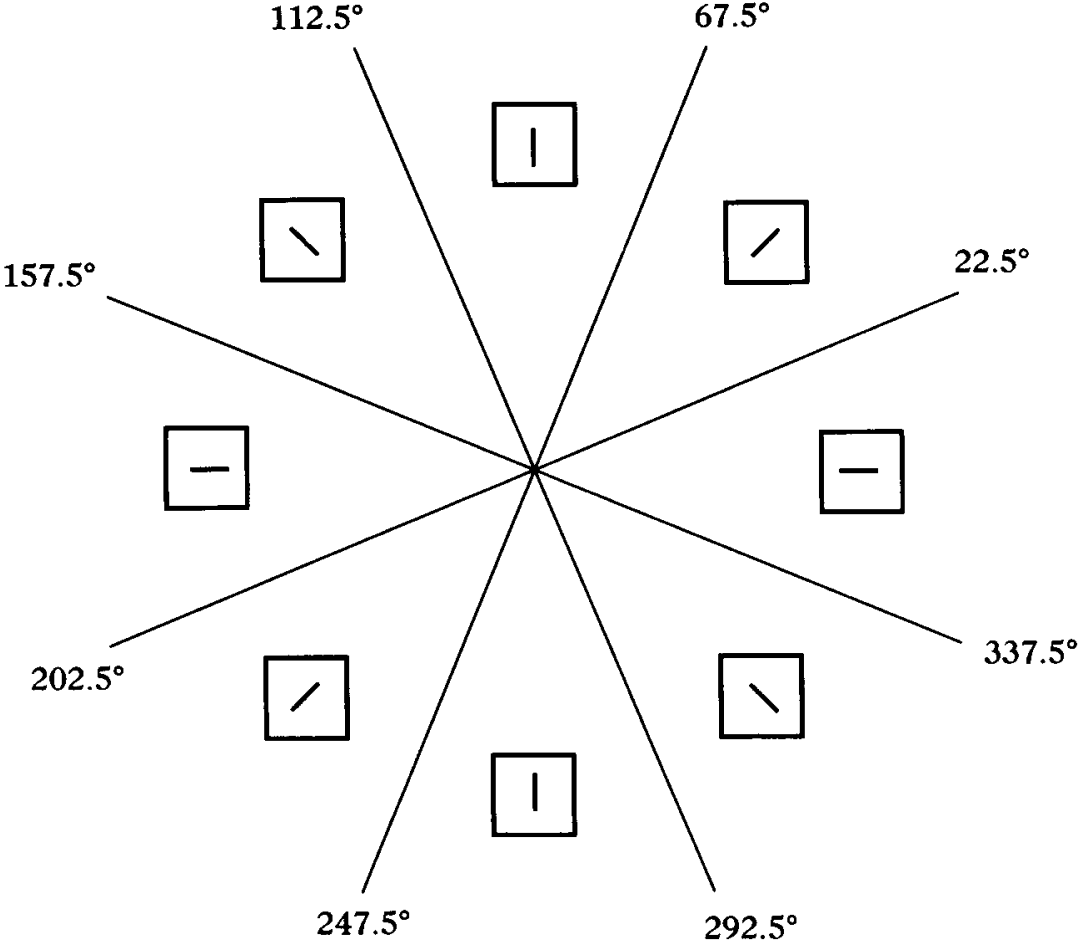
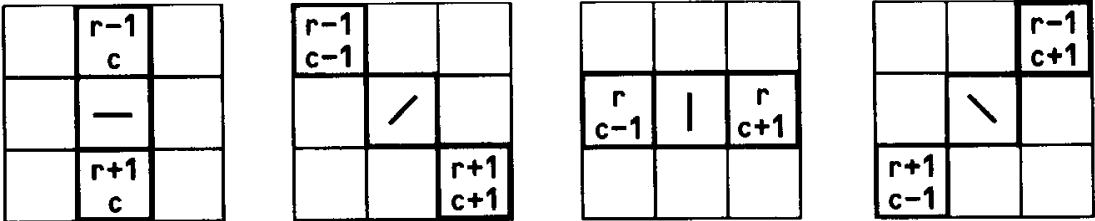
Nicht-Maxima-Unterdrückung (*Non-maxima suppression*)

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

Einteilung der Richtungen in Schritt 1:



Beispiele

einfaches Zahlenbeispiel:

Betrag

0	0	5	15	20	15	5	0
0	0	5	15	20	15	5	0
5	5	5	15	20	15	5	0
15	15	15	15	20	15	5	0
20	20	20	20	20	15	5	0
15	15	15	15	15	15	5	0
5	5	5	5	5	5	5	0
0	0	0	0	0	0	0	0

Nachbarn

0	0	270°	270°	270°	270°	270°	0
0	0	250°	270°	270°	270°	270°	0
180°	200°	225°	250°	270°	270°	270°	0
180°	180°	200°	225°	250°	270°	270°	0
180°	180°	180°	200°	225°	250°	270°	0
180°	180°	180°	180°	200°	225°	250°	0
180°	180°	180°	180°	180°	200°	225°	0
0	0	0	0	0	0	0	0

Richtung

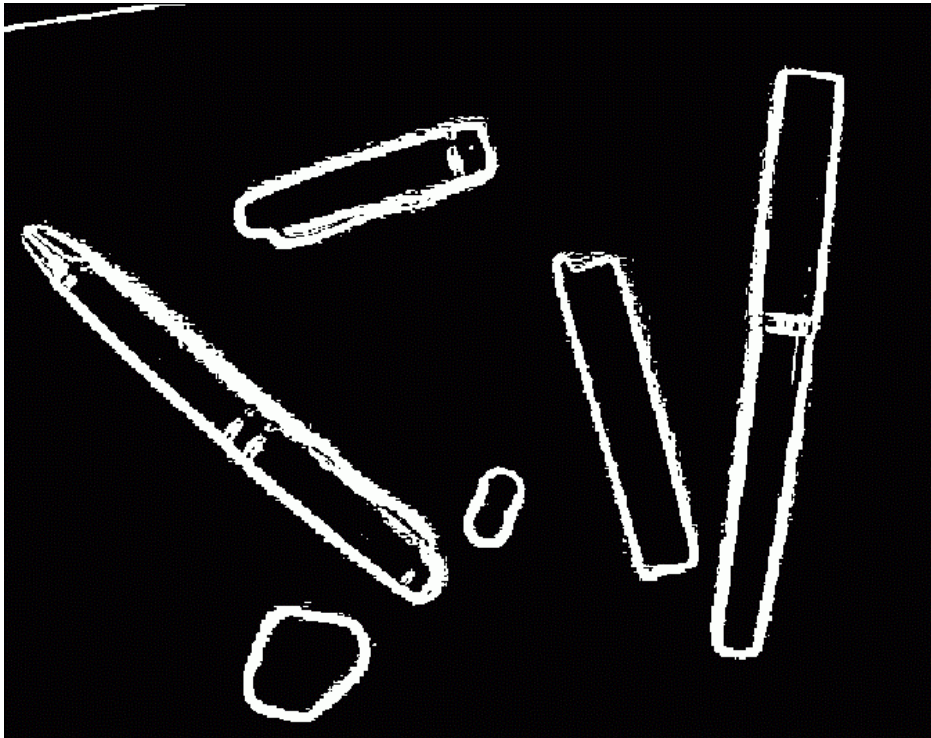
0	0	270°	270°	270°	270°	270°	0
0	0	250°	270°	270°	270°	270°	0
180°	200°	225°	250°	270°	270°	270°	0
180°	180°	200°	225°	250°	270°	270°	0
180°	180°	180°	200°	225°	250°	270°	0
180°	180°	180°	180°	200°	225°	250°	0
180°	180°	180°	180°	180°	200°	225°	0
0	0	0	0	0	0	0	0

Lokale Maxima

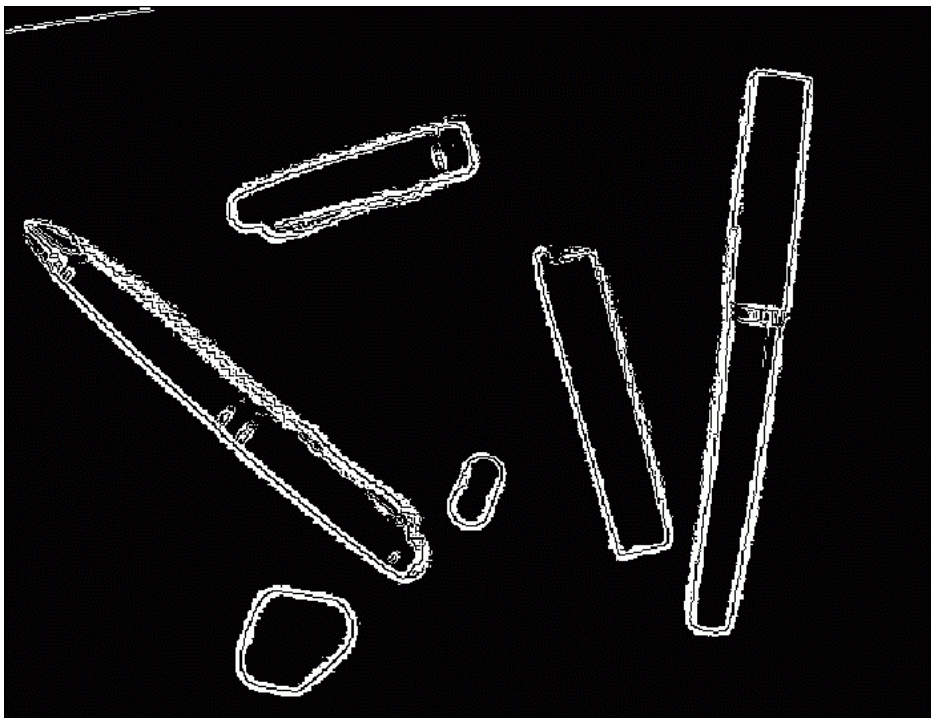
				20			
				20			
				20			
				20			
20	20	20	20	20			

(aus Bässmann & Kreyss 1998)

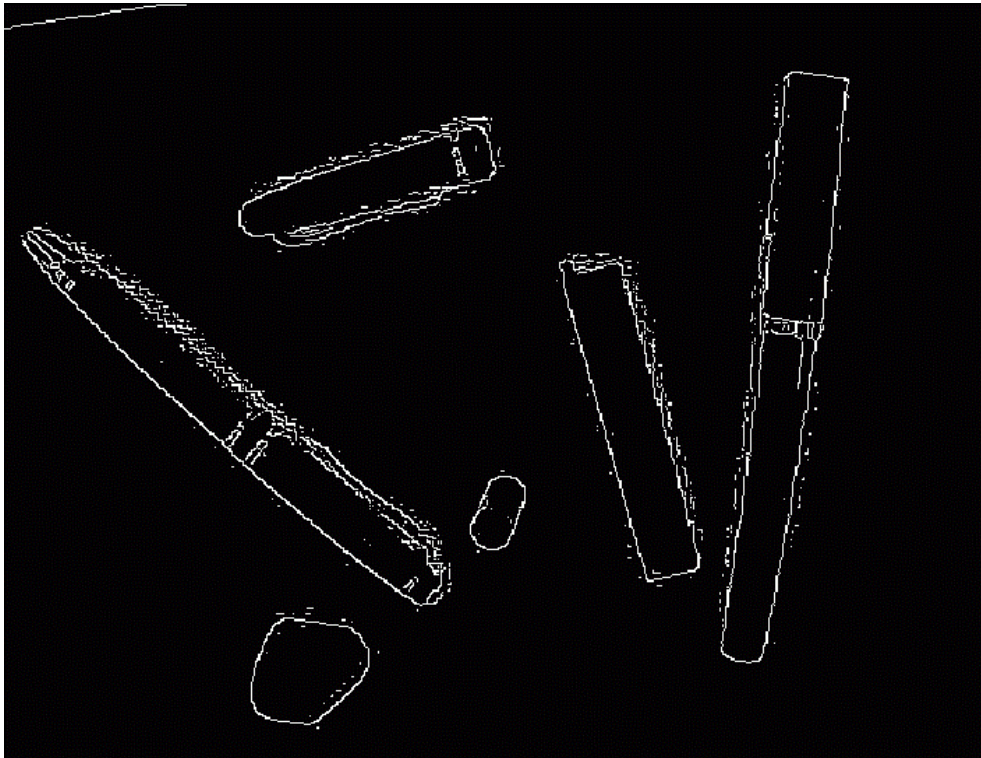
Bildbeispiel:



Kantenbild (Maximum über 8 Richtungen des Sobel-Operators, Schwellenwert 35) (aus Beichel 2002)



Markierte Pixel zum Löschen



Ergebnis

Verbesserung des Verfahrens:

Non-maxima-Unterdrückung sollte nicht nur auf dem Vergleich benachbarter Gradientenbeträge, sondern auch auf dem Vergleich der *Gradientenrichtungen* beruhen (Verwendung von *edgels*):

Innerhalb der verschmierten Kontur sind benachbarte Gradientenrichtungen als ähnlich anzunehmen \Rightarrow wenn solche Ähnlichkeit nicht vorliegt, liegt mit hoher Wahrscheinlichkeit kein Konturpunkt vor.

Ähnlichkeit der Richtung sollte Variation bis zu $\pm 30^\circ$ erlauben.

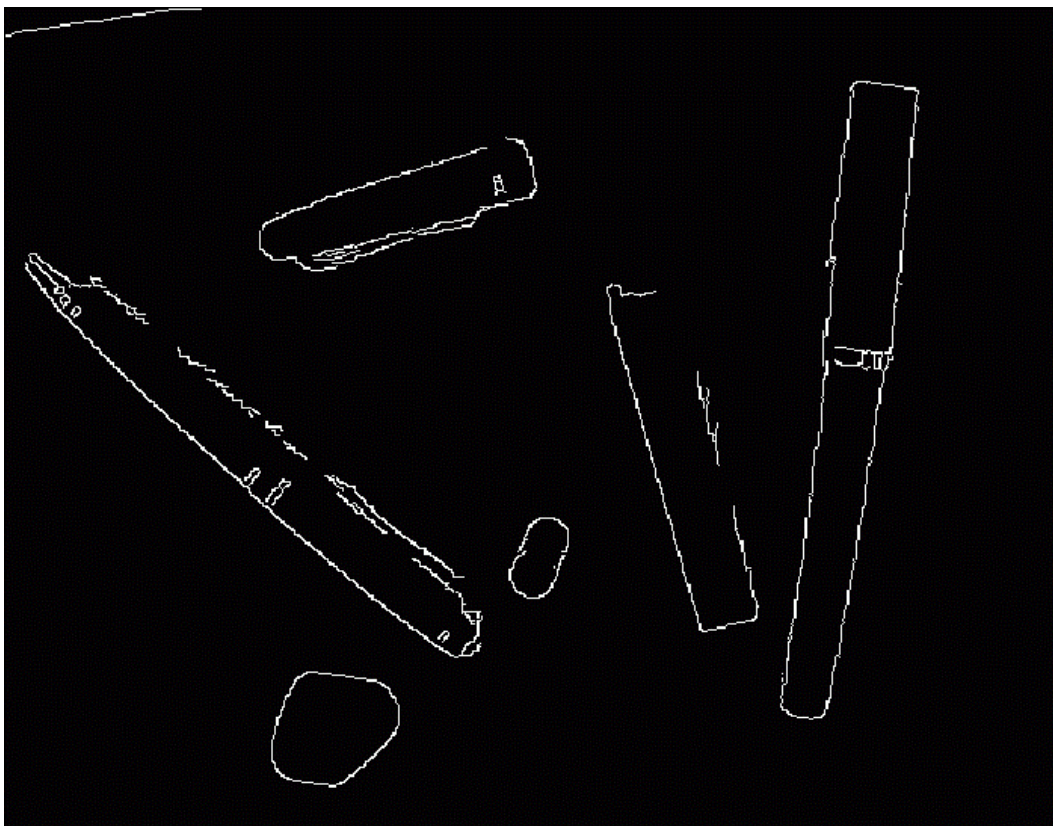
Nachbearbeitung der verdünnten Kanten:

- evtl. Herausslöchen von Kantenstücken, die eine gewisse Minimallänge nicht erreichen
- evtl. Übergang von Vierer- in Achternachbarschaft (Eliminieren von Eck-Pixeln durch geeignete Filtermasken, s. Bässmann & Kreyss S. 198 ff.).
- Hysteresisfilterung zur Elimination einzelner, immer noch vorhandener Rauschpixel

Hysteresisfilterung:

gegeben Schwellenwerte t_0 , t_1

1. Markiere Kanten mit Stärke größer als t_1 als korrekt
2. Untersuche alle Pixel mit Stärke zwischen t_0 und t_1
3. Markiere jedes solche Pixel, das benachbart zu einem bereits markierten Pixel ist
4. Zurück zu 2, bis sich nichts mehr ändert
5. Alle nicht markierten Kantenpixel werden gelöscht.



Ergebnis der Anwendung der Hysteresisfilterung auf obiges Beispiel (aus Beichel 2002)

Bisherige Methoden liefern immer noch Rauschpixel und fehlende Kantenstücke

⇒ weitere Untersuchungen innerh. der lokalen Nachbarschaft notwendig

- Ansatz: iterative Evaluierung von Kantenstärken, bis der Kanten-Zusammenhang geklärt ist
- Erhöhung oder Verminderung der Konfidenz aufgrund lokaler Nachbarschaften

Verfahren:

Edge Context Evaluation (ein Relaxationsverfahren)

siehe Beichel 2002, Sonka et al. 1999

Grundidee:

- schwache Kante zwischen 2 starken Kanten: gehört wahrscheinlich zur Kontur



- Einzelne (auch starke) Kante: wahrscheinlich kein Teil einer Regionengrenze



nächster Schritt nach der Verdünnung der Kanten:

Konturpunktverkettung

Hinzufügen einer symbolischen Information aufgrund des Zusammenhangs (Label für jede Zusammenhangskomponente)

- Ziel: Pixel als zusammengehörig markieren
- verschiedene Objekte sollen verschieden markierte Konturen haben

Verfahren:

Border tracing (Randverfolgung)

iterierte Suche nach dem nächsten Konturpunkt, beginnend mit dem "östlichen" Nachbarn des aktuellen Pixels und die Nachbarn entgegen dem Uhrzeigersinn durchlaufend

resultierende Datenstruktur: Liste!

Beispiel zur Durchführung der Konturpunktverkettung:

					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	1	1	

Eingabebild
(verdünnte Kanten)

					a		
	b	b	b			a	
				b			a
	a	a			b		a
a					b		a
a		b	b	b			a
a							a
	a	a	a	a	a	a	

2 detektierte Ketten
(Labels a und b)

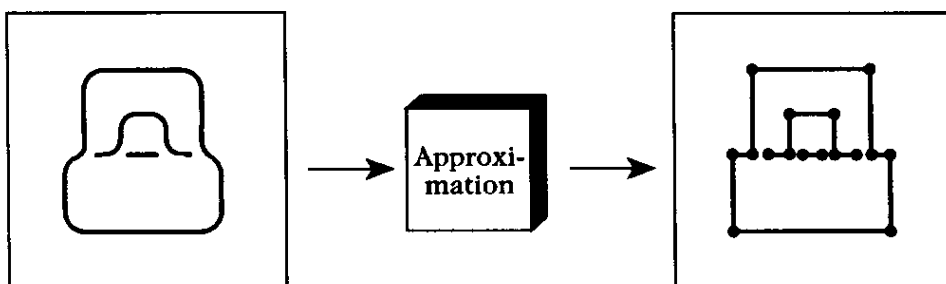
bisher nur Listen von Konturpunkten, noch keine Geradenstücke (oder andere parametrische Kurven)

nächster Schritt:

Konturapproximation

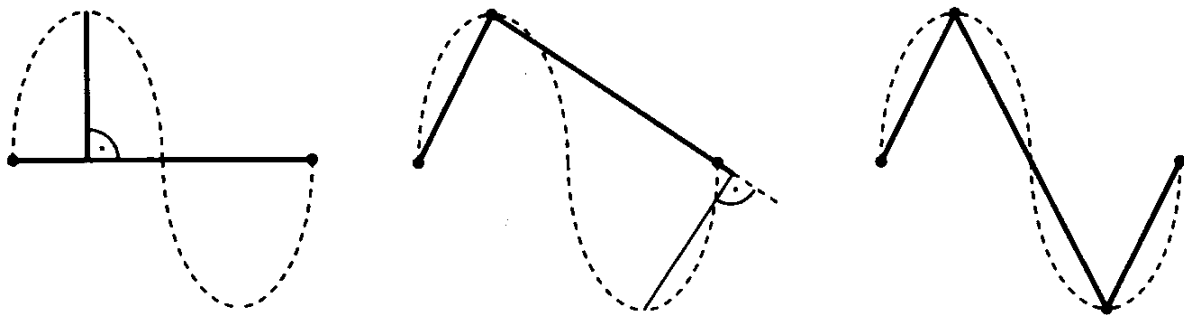
gemeint ist: Approximation durch Modell-Kurven (oft Geraden)

- ermöglicht Aussagen auf höherer Ebene, z.B. Parallelität von Konturen
- Eignung zur Klassifikation



einfaches Approximationsverfahren
(erzeugt Polylinien = Ketten von Geradenstücken):

1. Annäherung der Konturpunktkette durch einziges Geradenstück
2. Wenn der maximale senkrechte Abstand zwischen Polylinie und Konturpunktkette einen Schwellenwert überschreitet, spalte man die Polylinie an der Stelle des maximalen Fehlers
3. Wiederhole 2., bis der erlaubte Fehler nicht mehr überschritten wird.



Regionenbasierte Segmentierung

Grundprinzip:

Aufteilung des Bildes in Zonen maximaler Homogenität

- besser als kantenbasierte Segmentierung bei verrauschten und unscharfen Bildern
- Homogenität bezogen auf: Grauwert, Farbe, Textur, Form oder andere Modell-Merkmale

Verfahren:

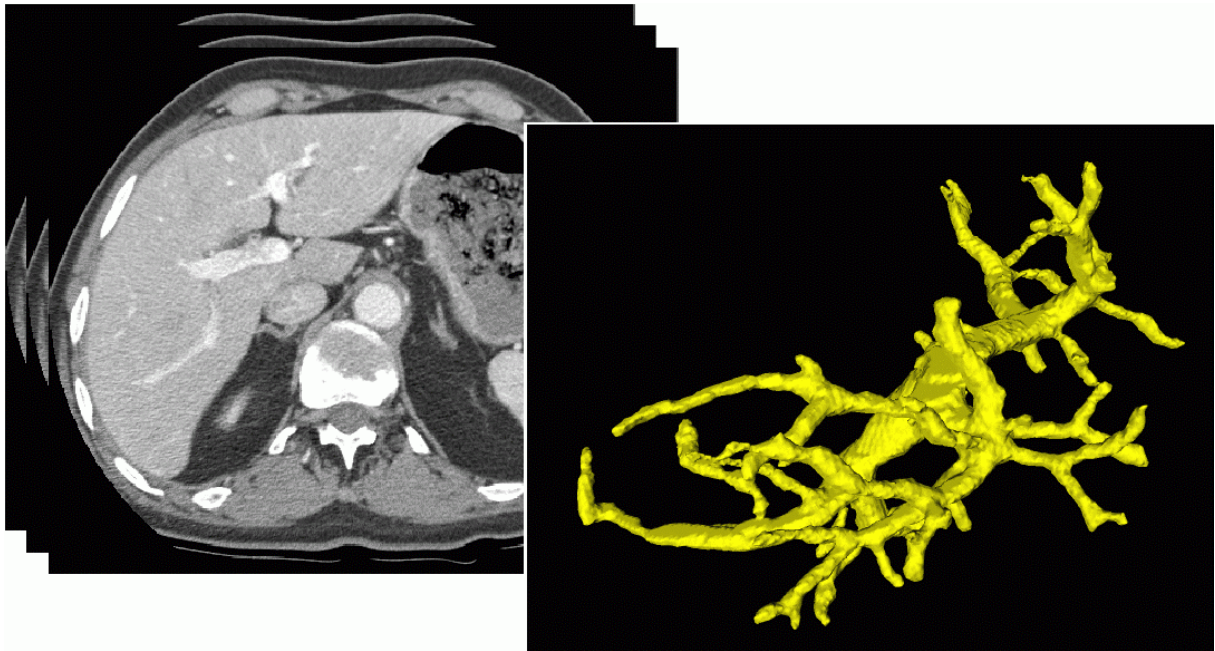
- Region Growing
- Merging
- Boundary Melting
- Splitting
- Split and Merge
- clusteranalytische Verfahren
- Matching

Region Growing (Regionenexpansion):

- Bottom-up Algorithmus
- **Start:** Seed Pixeln (Startregionen), die zu verschiedenen Regionen gehören (kann auch nur ein Seed Pixel sein)
- **Hinzufügen** von benachbarten Pixel, die ein Homogenitätskriterium erfüllen und nicht zu einer anderen Region gehören
- **Abbruch**, wenn keine Änderungen mehr auftreten.

- Unterschiedliche **Startpunkte** können unterschiedliche Ergebnisse liefern
- Wahl des **Homogenitätskriteriums** ist entscheidend für gute Ergebnisse (leakage)
- Keine vollständige Aufteilung des Bildes in Regionen möglich

Beispiel:
Region Growing in 3D (aus Beichel 2002)

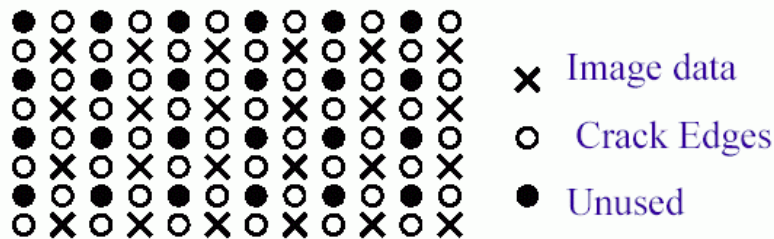


Region Merging:

- **Start:** Jedes Pixel ist eine eigene Region
- **Verbinde** benachbarte Regionen, die dasselbe Homogenitätskriterium erfüllen
- **Bis** keine Änderungen mehr auftreten.
- Auch anderes Anfangskriterium möglich
- Resultat kann abhängen von der gewählten Suchrichtung (wenn Homogenitätskriterien laufend angepaßt werden)

Boundary Melting:

- Berücksichtigung von **Crack Edges**
- **Supergrid** Datenstruktur:



- **Kantensignifikanz:** $v_{ij} = 0$ if $s_{ij} < T_1$

$$s_{ij} = |f(x_i) - f(x_j)|$$

$$= 1 \text{ otherwise}$$

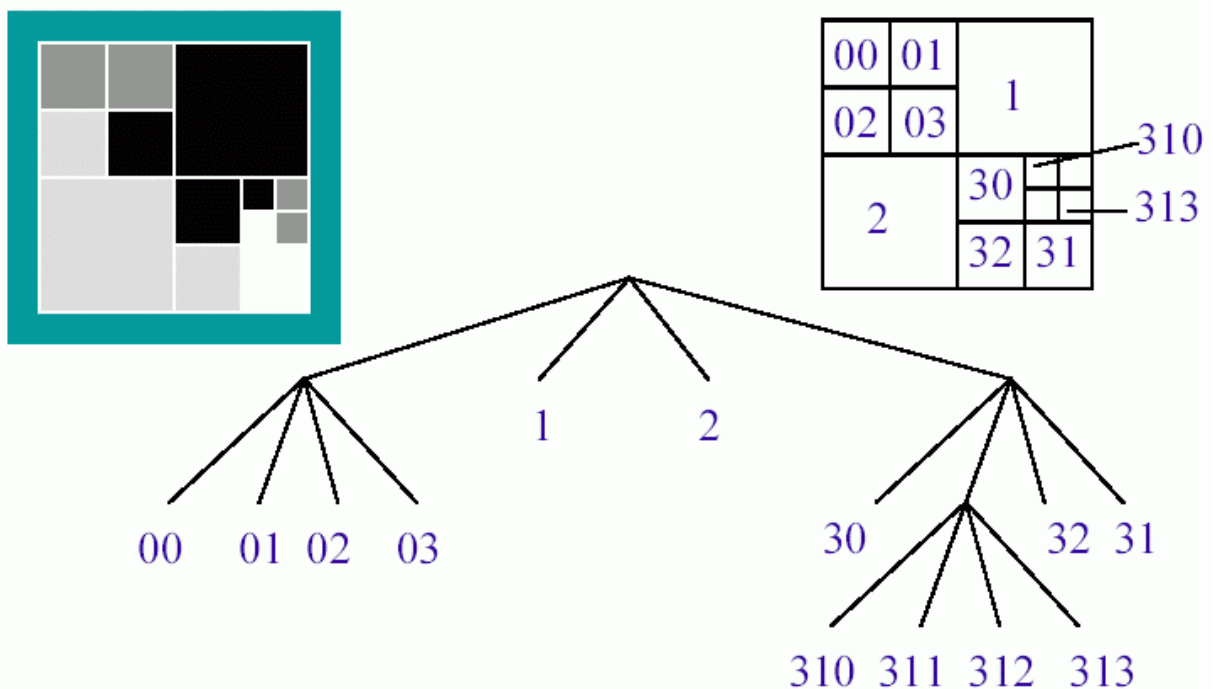
- Zwei Regionen werden verschmolzen, wenn ein **signifikanter Teil** der gemeinsamen Grenze aus **schwachen Kantenpixeln** besteht, unter Berücksichtigung der Regionengröße
- Bzw. auch unabhängig von der Regionengröße
- **Startsegmentierung** in Regionen mit konstantem Grauwert, Supergridstruktur mit Crack Edge Information füllen
- **Schwache Kanten** aus Supergridstruktur **entfernen** ($< T_1$)
- Rekursiv Grenzen zwischen benachbarten Regionen entfernen: $\frac{W}{\min(l_i, l_j)} \geq T_2$
- Rekursiv weitere Grenzen entfernen:

$$\frac{W}{l} \geq T_3 \quad \text{oder:} \quad W \geq T_3$$

Region Splitting:

- Umgekehrt zu Merging, geht vom ganzen Bild als eine Region aus,
- unterteilt so lange, bis Homogenitätskriterium erfüllt ist.
- Liefert andere Ergebnisse als Merging!

Beispiel:



Split and Merge:

- **Initialisierung:** Startregion, Pyramidenstruktur, Homogenitätskriterium
- Teile jede **inhomogene Region R** in der Pyramidenstruktur **in 4 Teilregionen**
- **Verbinde benachbarte Regionen** (auch aus unterschiedlichen Ebenen), die gemeinsam das Homogenitätskriterium erfüllen

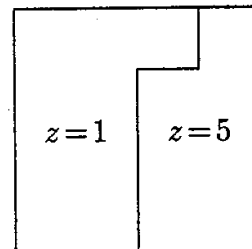
einfaches Beispiel:

1	1	1	5
1	1	5	5
1	1	5	5
1	1	5	5

(a)

1	1	5
	5	5
1	5	

(b)



(c)

(a) Ausgangsbild, (b) Quadtree-Aufteilung (*split*),
(c) Endergebnis (*merge*).

Clusteranalytische Verfahren:

- jeder Bildpunkt wird mit einem Merkmalsvektor versehen
- und zwar so, dass die zu einer Fläche gehörenden Merkmalsvektoren nah beieinander liegen
- Auffinden einer Fläche = Auffinden einer Anhäufung (Cluster) von Merkmalsvektoren im Merkmalsraum
- "Fehler" eines (beliebigen) Clusters von Merkmalsvektoren: (euklidischer) Abstand vom Schwerpunkt
- "Fehler" einer Clusterung aller Merkmalsvektoren = Summe der Fehler-Werte der einzelnen Cluster
- Klasse von Verfahren: ausgehend von einer initialen Clusterung wird der Fehler sukzessive minimiert (unterschiedliche Strategien im Detail)

Nachverarbeitung für Region Growing und andere regionenbasierte Methoden:

- Durch **nichtoptimale Parameter** entweder
 - zu viele Regionen (under-growing)
 - zu wenige Regionen (over-growing)
- Mögliche **Nachverarbeitungsschritte**:
 - Verwendung von Kanteninformation
 - Verringerung der Anzahl kleiner Regionen

Regionenbasierte Verfahren benötigen außerdem oft eine *Zusammenhangsanalyse* (vgl. Übung, Aufgabe U3 c):

Bereiche mit gleichem Label, aber ohne Zusammenhang werden unterschiedlichen Marken (Regionen) zugeordnet

"Komponentenmarkierung", "blob coloring"

detailliertes Verfahren bei Bässmann & Kreyss 1998, S. 171 ff.

Beispiel für die Komponentenmarkierung:

5	5	0	0	0	0	10	10
5	5	0	0	0	0	10	10
10	10	10	10	5	5	10	10
10	10	10	10	5	5	10	10
10	10	5	5	5	5	10	10
10	10	5	5	5	5	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10

Ursprungsbild

Schwelle bei
Grauwert 3 und 7

1	1	0	0	0	0	2	2
1	1	0	0	0	0	2	2
2	2	2	2	1	1	2	2
2	2	2	2	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2

Labelbild

Zusammenhangsanalyse

a	a	-	-	-	-	b	b
a	a	-	-	-	-	b	b
b	b	b	b	c	c	b	b
b	b	b	b	c	c	b	b
b	b	c	c	c	c	b	b
b	b	c	c	c	c	b	b
b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b

Markenbild

(Label 0 soll hier dem Hintergrund entsprechen und wird keiner markierten Region zugeordnet)

Problemfälle bei einigen Kontursegmentierungsverfahren:

Löcher in Objekten

Objekte in Löchern von Objekten...

⇒ rekursive Algorithmen; Aufbau einer Hierarchie

Matching

- Finden **bekannter Objekte** im Bild
- Anwendungen:
 - Stereoauswertung
 - **Objektverfolgung** in Bildsequenzen
- Meistens werden **Grauwerte verglichen**
- Können aber auch aus dem Bild berechnete **Features oder Deskriptoren** höherer Ordnung sein.

Matching-Algorithmus:

- Prüfe das Matching-Kriterium für jede Position und Rotation des gesuchten Musters im Bild (Berechnung eines Qualitätsmaßes für das Matching)
- Lokale Maxima des Qualitätsmaßes, die einen festgelegten Schwellenwert überschreiten, beschreiben die Positionen des Musters im Bild

Matching-Qualitätsmaß:

- exakte Kopie des gesuchten Objektes kaum vorhanden (Rauschen, geometrische Unterschiede...)
- Korrelation ist ein übliches Qualitätsmaß (verschiedene Varianten der Korrelation)

Matching kann auch im Frequenzraum ausgeführt werden

Matching = zeitaufwändige Art der Segmentierung!