

## 4. 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...)

## Teilaufgaben der Segmentierung:

- Finden interessanter Stellen im Bild
  - Trennung der zu untersuchenden Objekte von den übrigen Bildstrukturen
  - Trennung von sich berührenden oder überlappenden Objekten voneinander
  - Zerlegung in Teilobjekte
- Aufteilung eines Bildes in Teile, die Objekten bzw. Hintergrund entsprechen
  - **Komplette Segmentierung:** Aufteilung in nichtüberlappende Gebiete, die Objekten entsprechen -> Wissen erforderlich
  - **Teilegmentierung:** Regionen korrespondieren nicht direkt mit Objekten -> Vorverarbeitungsschritt

## 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 kompletten 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
- Auffinden interessanter Punkte im Bild (*points of interest*)
- *kantenbasierte* Segmentierung (Kantendetektion, 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)

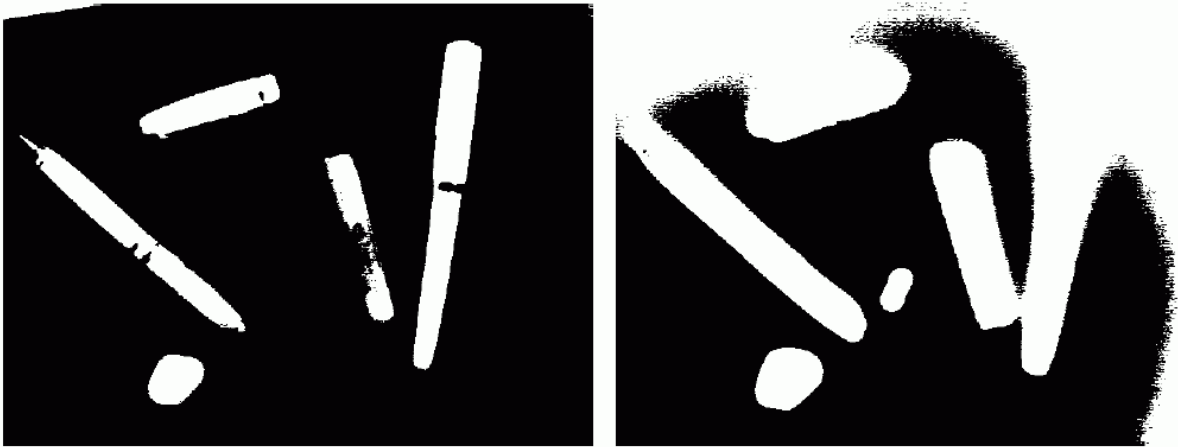
- Grauwert-**Schwellwertsetzung** ist die einfachste Art der Segmentierung
- Nicht rechenaufwendig - **schnell**
- Wird häufig verwendet für **einfache Anwendungen**
- Transformation eines Bildes  $f$  in ein segmentiertes Binärbild  $g$ :

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

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



## Probleme: Unter- und Übersegmentierung



- Einzelner, globaler Schwellwert reicht meistens nicht aus zur Segmentierung

- **Globaler** Schwellwert:

$$T = T(f)$$

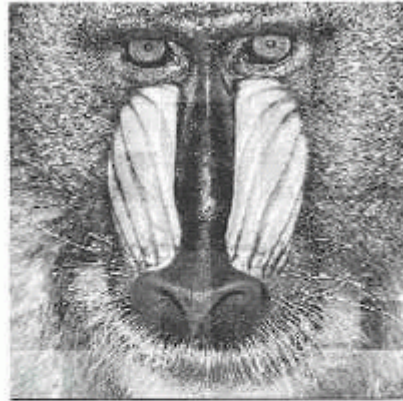
- **Lokaler** Schwellwert:

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

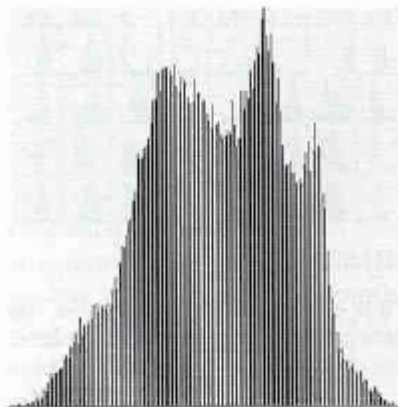
z.B. durch Aufteilung in Teilbilder



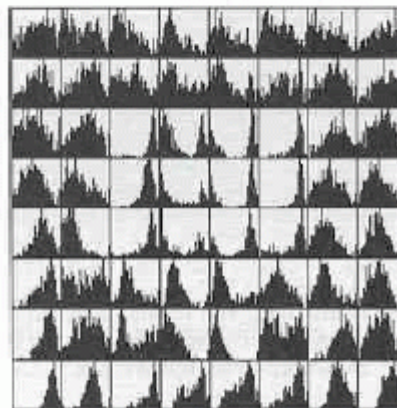
(A)



(B)



(C)

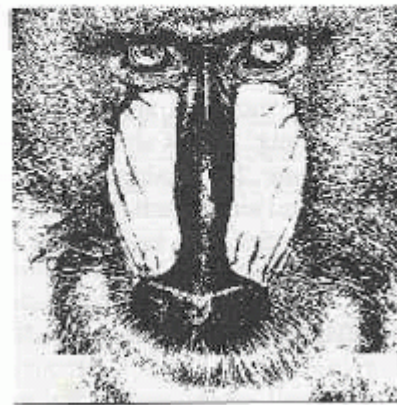


(D)



(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: Schwellenwertbestimmung

### 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):

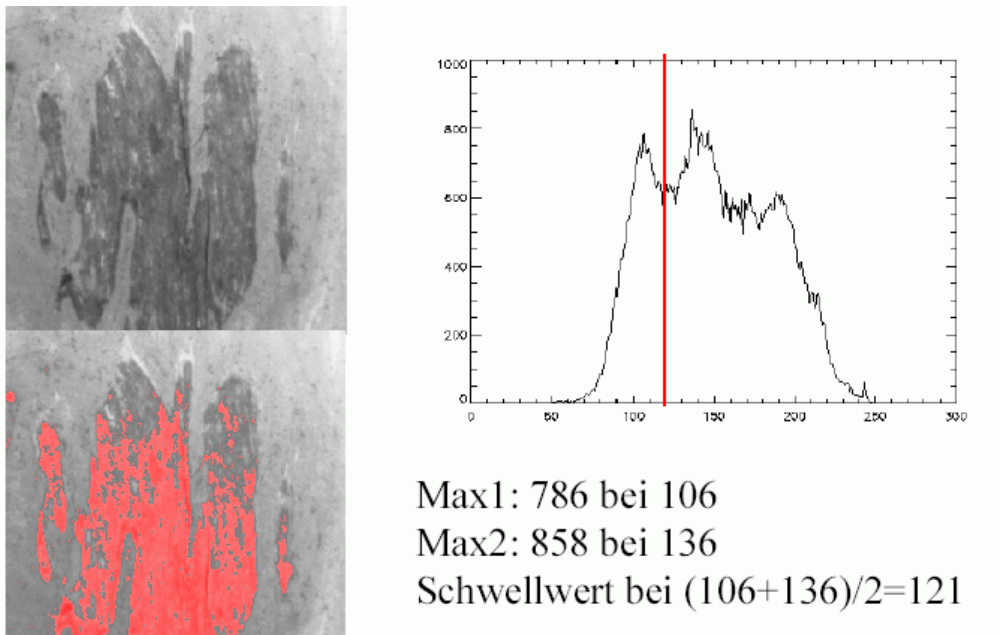
0	1	2	3	4	5	6	7	8	9	0	1	2	3	
0	1	2	3	4	5	6	7	8	9	0	6	5	3	
0	1	2	3	4	5	6	7	8	9	1	9	6	1	
0	1	2	3	4	5	6	7	8	9	7	2	3	9	
0	1	2	3	4	5	6	7	8	9	7	8	5	4	
0	1	2	3	4	5	6	7	8	9	8	4	7	2	
0	1	2	3	4	5	6	7	8	9	7	3	5	6	
0	1	2	3	4	5	6	7	8	9	3	5	7	9	
0	1	2	3	4	5	6	7	8	9	4	4	1	8	
0	1	2	3	4	5	6	7	8	9	1	5	6	9	

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	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	

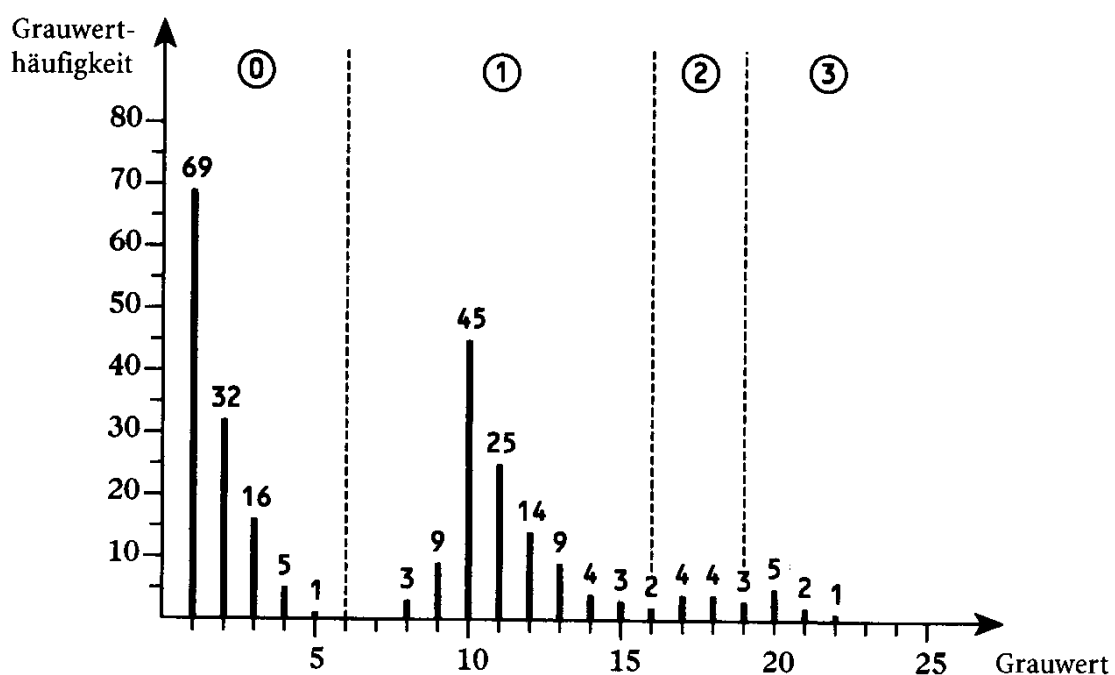


oft verwendet zur Schwellenwertbestimmung:  
Annahme eines bimodalen Histogramms



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

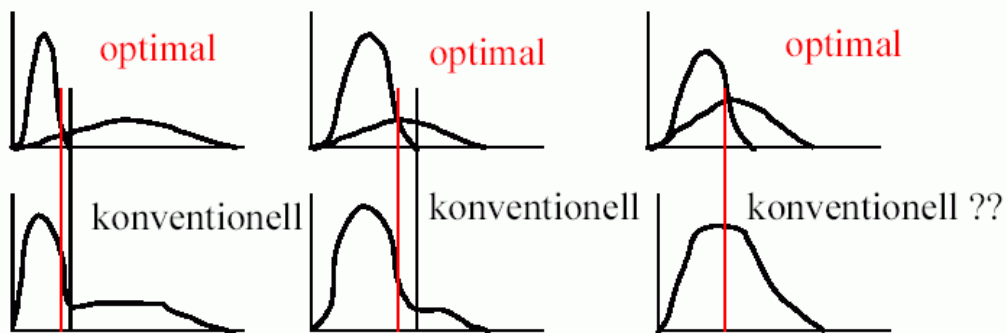
Probleme bei Verwendung der Histogrammanalyse zur  
Segmentierung:



- 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 Schwellenwertes:

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



Verfahren der iterativen optimalen Schwellenwertbestimmung:

- **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

## *Points of Interest (POI)*

= Punkte im Bild, die von besonderem Interesse sind (Ecken von Regionen, Extremstellen, Linien-Endpunkte...)

POI-Operatoren: liefern zu einem Bild eine Liste von POIs

Beispiele:

### *Moravec-Operator*

berechnet die mittleren quadratischen Gradientensummen in den 4 Hauptrichtungen für ein Bildfenster um jedes Pixel

Aus den 4 Summen wird das Minimum  $V$  gebildet:

$$V_0 = \frac{1}{p(q-1)} \sum_{i=-k}^{+k} \sum_{j=-l}^{+l} [g(i, j) - g(i, j+1)]^2$$
$$V_{90} = \frac{1}{(p-1)q} \sum_{i=-k}^{+k} \sum_{j=-l}^{+l} [g(i, j) - g(i+1, j)]^2$$
$$V_{45} = \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{+k} \sum_{j=-l}^{+l} [g(i, j) - g(i+1, j+1)]^2$$
$$V_{135} = \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{+k} \sum_{j=-l}^{+l} [g(i, j+1) - g(i+1, j)]^2$$

$$V = \min(V_0, V_{45}, V_{90}, V_{135})$$

Wenn  $V$  über einem Schwellenwert liegt, liegt ein markantes Muster vor.

Der Operator listet alle Pixel auf, für die dies der Fall ist.

## Harris-Plessey-Operator

Mit den Prewitt-Operatoren in horizontaler und vertikaler Richtung werden für jedes 3×3-Bildfenster die Intensitätsänderungen in x- und y-Richtung,  $I_x$  und  $I_y$ , berechnet. Es sei  $M$  die Matrix

$$M = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}. \text{ Dann ist die } \textit{Eckigkeit} \text{ def. als}$$

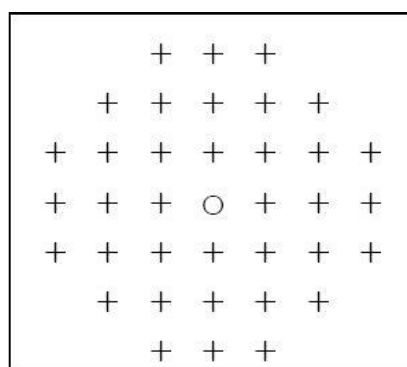
$$\textit{Corn} = \det(M) - k \cdot \text{Spur}^2(M).$$

Wieder wird dieser Wert mit einem Schwellenwert verglichen, um die POI zu selektieren.

## SUSAN-Operator

(smallest univalue segment assimilating nucleus)

Grundidee ist der Vergleich eines Kernpunktes mit allen Punkten in einer Umgebung, die als USAN (Univalue Segment Assimilating Nucleus) bezeichnet wird. Dabei wird eine etwa kreisförmige Maske (im Normalfall 37 Pixel) über das Bild geschoben und die Anzahl der Punkte aufaddiert, deren Grauwert sich nur gering von dem des Kernpunktes unterscheidet, d.h. deren Grauwertdifferenz unter einem gegebenen Schwellenwert liegt.

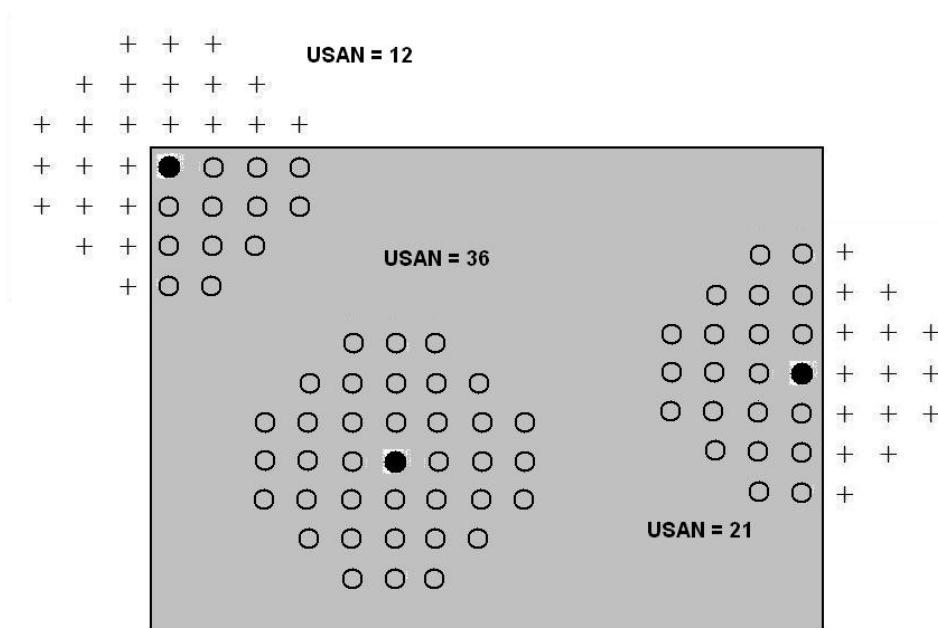


USAN-Maske mit 37 Pixeln  
(nach Hartmann et al. 2005)

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases}$$

Die Anzahl der Punkte, die in der USAN-Area gezählt wurden, gibt Aufschluss über die Lage des Kernpunktes im Bild:

1. bei maximalem Wert (36) liegt der Punkt in einer homogenen Fläche,
2. bei ca. der Hälfte der USAN (36/2) liegt der Punkt in der Nähe einer Kante,
3. bei einem sehr kleinen Wert ( $\leq 36/3$ ) handelt es sich um einen Eckpunkt.



### Werte für USAN

(aus Kozempel 2007)

Nachteil:

verrauschte Regionen können Ergebnis verfälschen.

Verbesserung: Man ermittelt den Schwerpunkt aller gezählten Punkte in der USAN. Liegt dieser nahe am Mittelpunkt, handelt es sich sehr wahrscheinlich nicht um einen echten Eckpunkt. Diese Fälle werden aussortiert.

Vorteil:

keine vorherige Filterung, keine Gradientenapproximation erforderlich.



## Kantenbasierte Segmentierung

Grundlage: Kantendetektion durch lineare Filterung und / oder morphologische Bildoperationen - siehe Kapitel 3 -

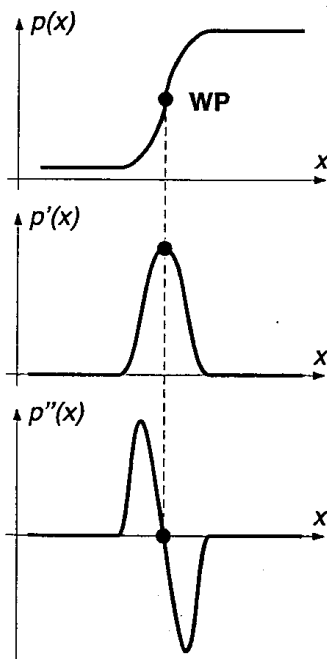
"It is hard to over-emphasize the importance of edge-detection in image understanding" (Nalwa & Binford, zit. bei Pinz 1994)

Begriffsklärung:

"Kante": Grauwertsprung im Bild

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

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 Grauwertänderungen, 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):

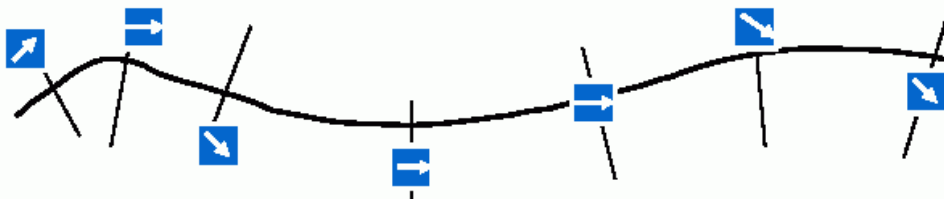
Punkt (Pixel) mit Information über Kantenrichtung und Betrag des Gradienten an diesem Punkt

*nützlich ist oft a-priori-Wissen*

Möglichkeit 1: Information über Vorliegen bestimmter Formen (Geradenstücke, Kreise...) durch Bildtransformation gewinnen (Hough-Transformation, siehe späteres Kapitel)

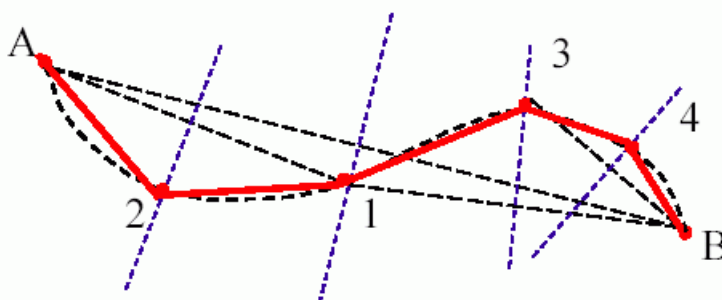
Möglichkeit 2: Randbestimmung bei bekannter Randpositionsinformation

- 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**.

## Konturfindung:

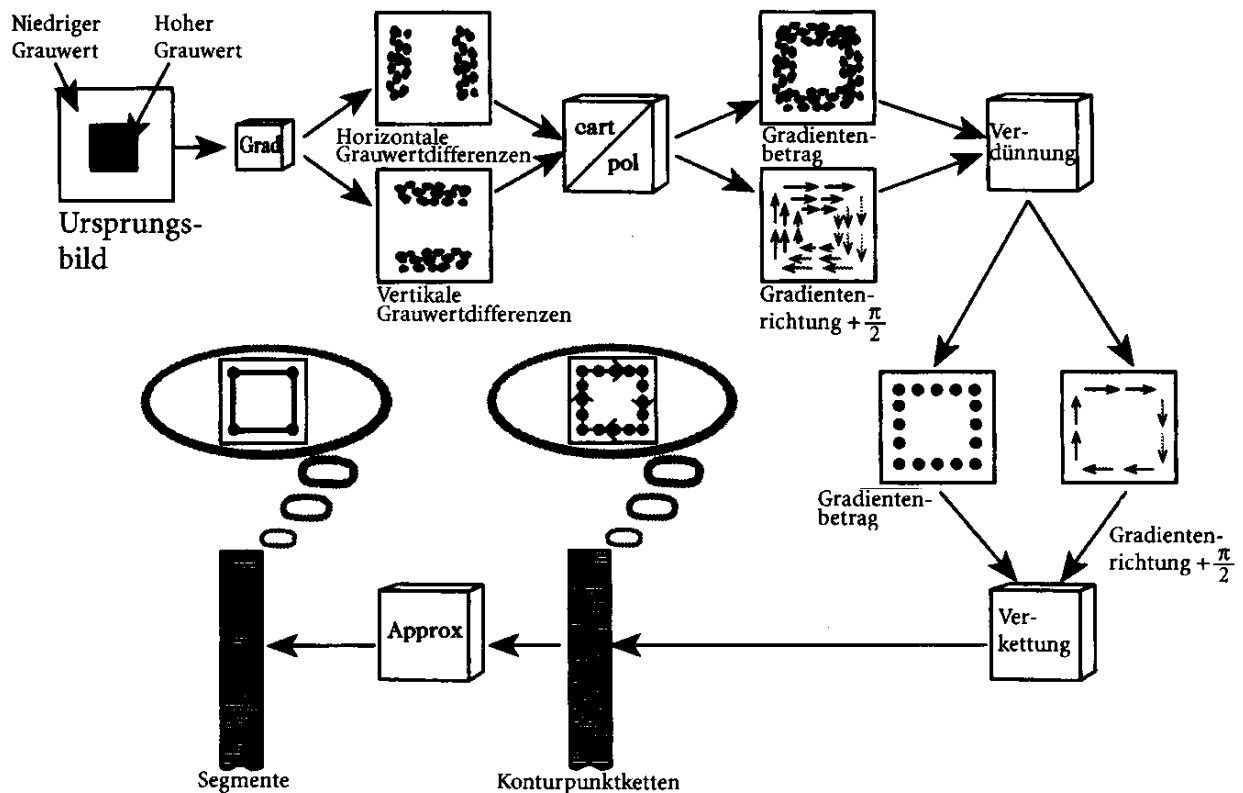
ein Objektpunkt liegt genau dann am Rand des Objektes, wenn mindestens einer seiner (Vierer- oder Achter-) Nachbarn ein Hintergrundpunkt ist

⇒ 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)

Konturfolgeverfahren für beliebige Konturformen, die durch Geradenstücke approximiert werden (zusammengefasst: "Canny-Operator"):



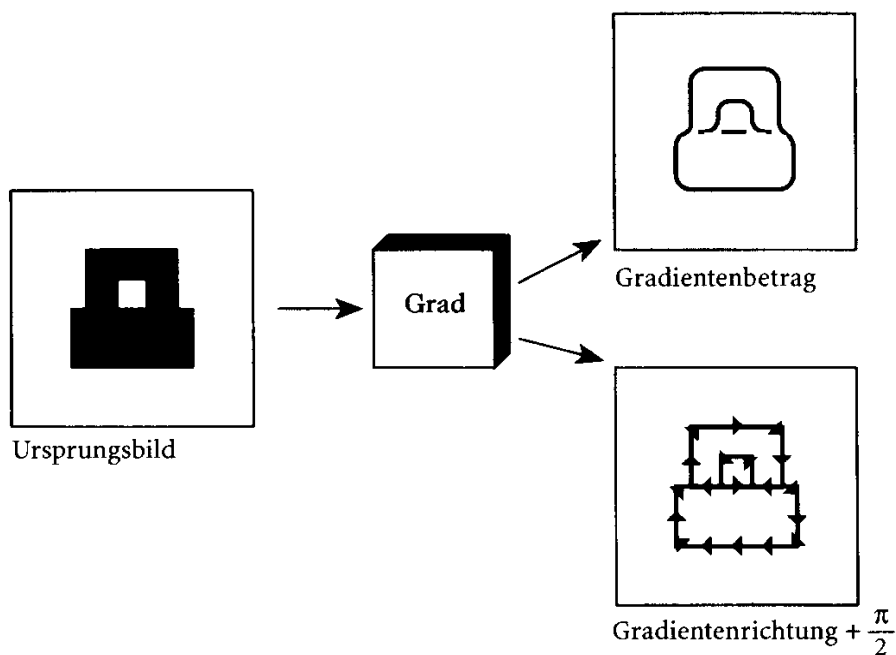
(aus Bässmann & Kreyss 1998)

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

Einzelschritte bei diesem Verfahren:

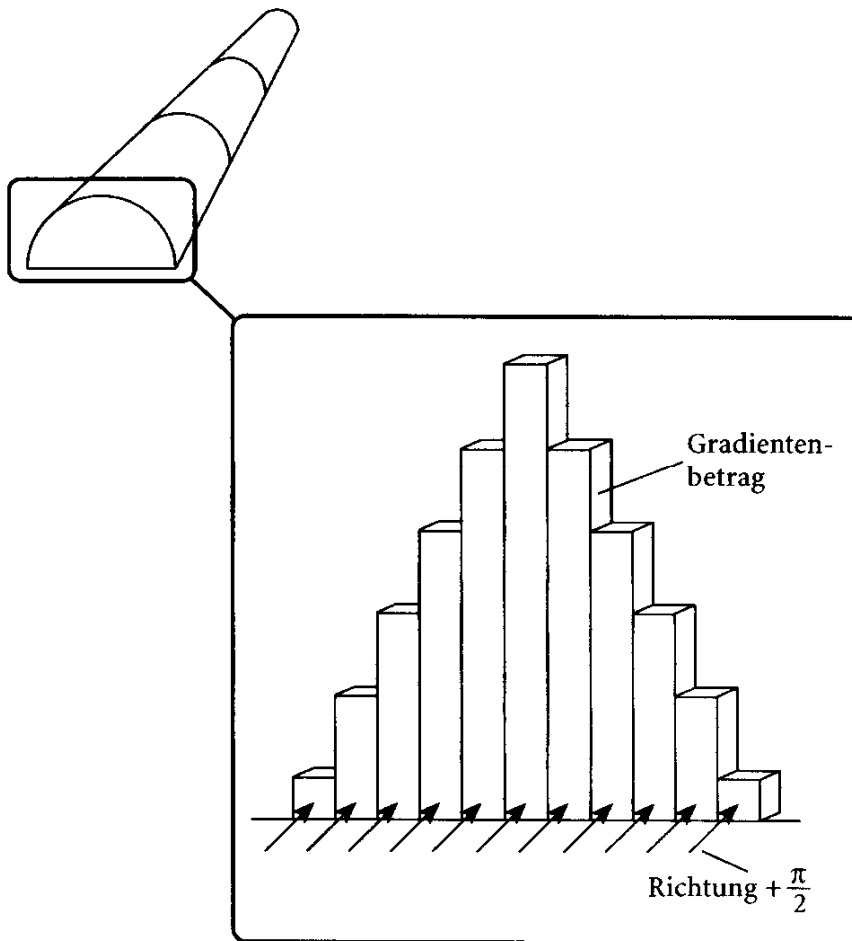
- Kantendetektion
- Verdünnung
- Verkettung
- Konturapproximation

Kantendetektion (schon behandelt):



## Verdünnung der Konturen:

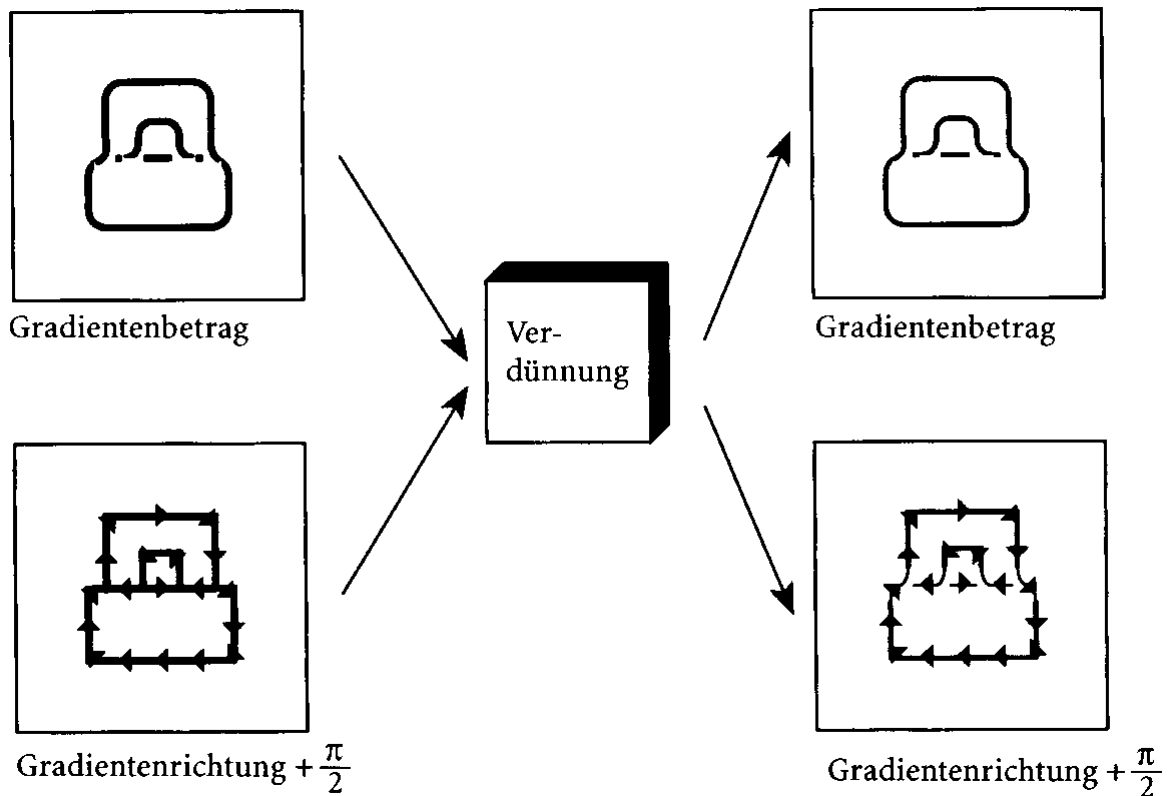
- Beträge des Gradienten verhalten sich i.d. Regel wie langgestreckte Gebirgszüge
- Annahme: tatsächliche Objektkontur liegt beim Maximum



(aus Bässmann & Kreyss 1998)

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





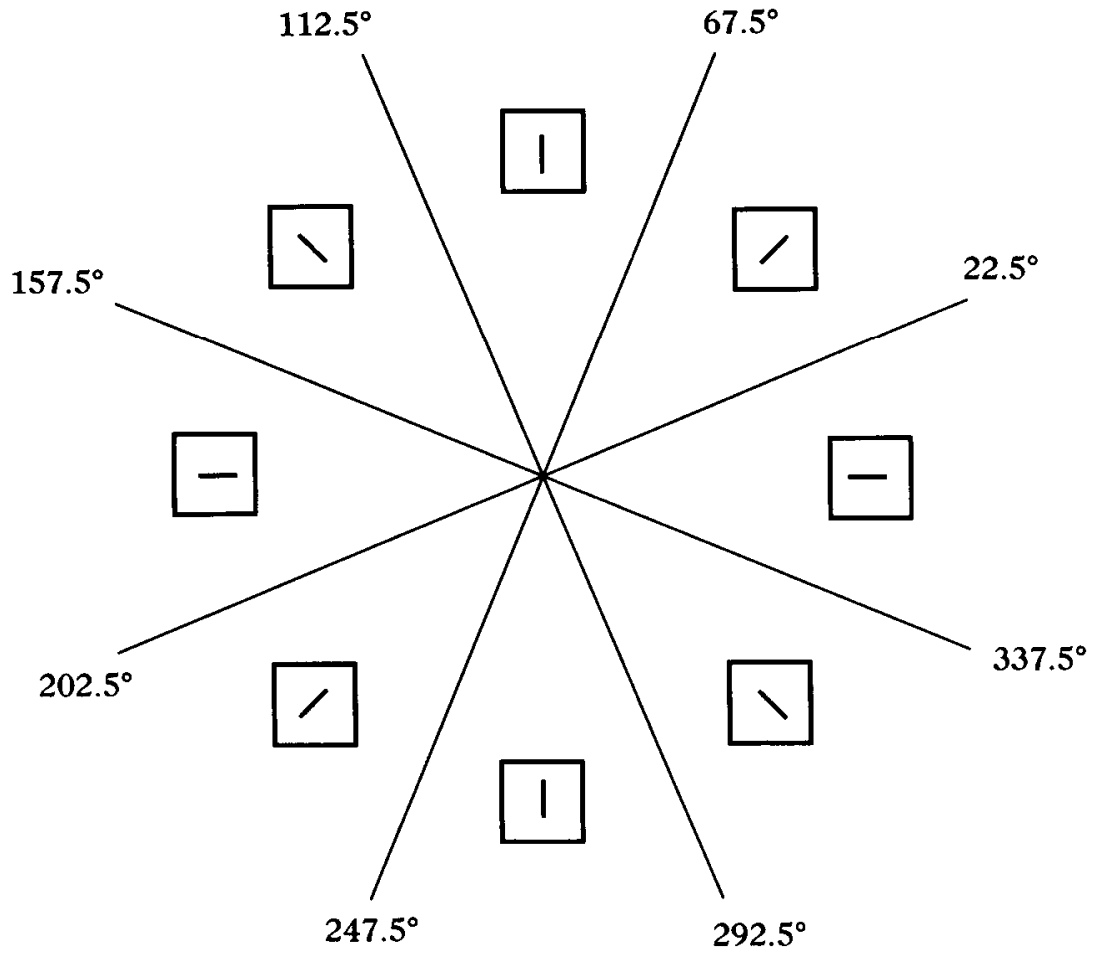
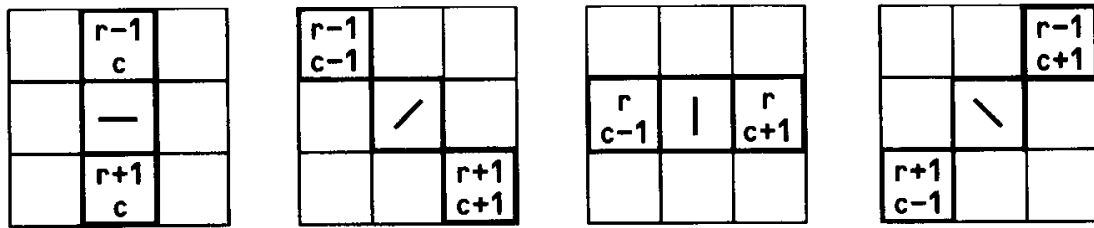
Algorithmus:  
 Nicht-Maxima-Unterdrückung  
 (*Non-maxima suppression*)

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

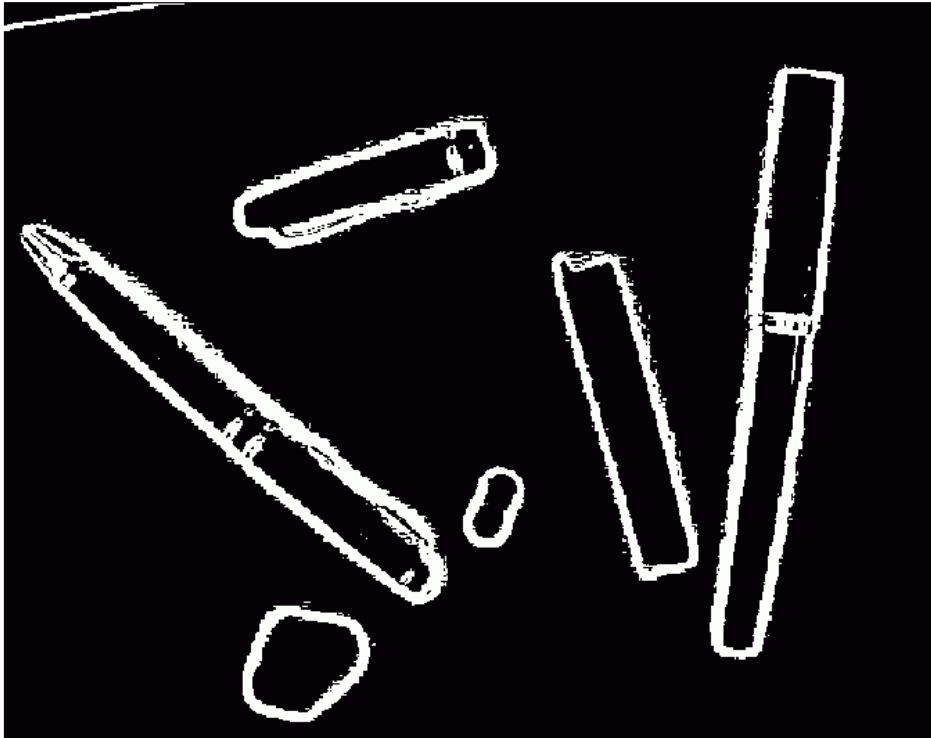
0	0	270°	270°	270°	270°	270°	0
0	0	250°	270°	270°	270°	270°	0
180°	200°	225°	250°	270°	270°	270°	0
180°	180°	200°	225°	250°	270°	270°	0
180°	180°	180°	200°	225°	250°	270°	0
180°	180°	180°	180°	200°	225°	250°	0
180°	180°	180°	180°	180°	200°	225°	0
0	0	0	0	0	0	0	0

Lokale Maxima

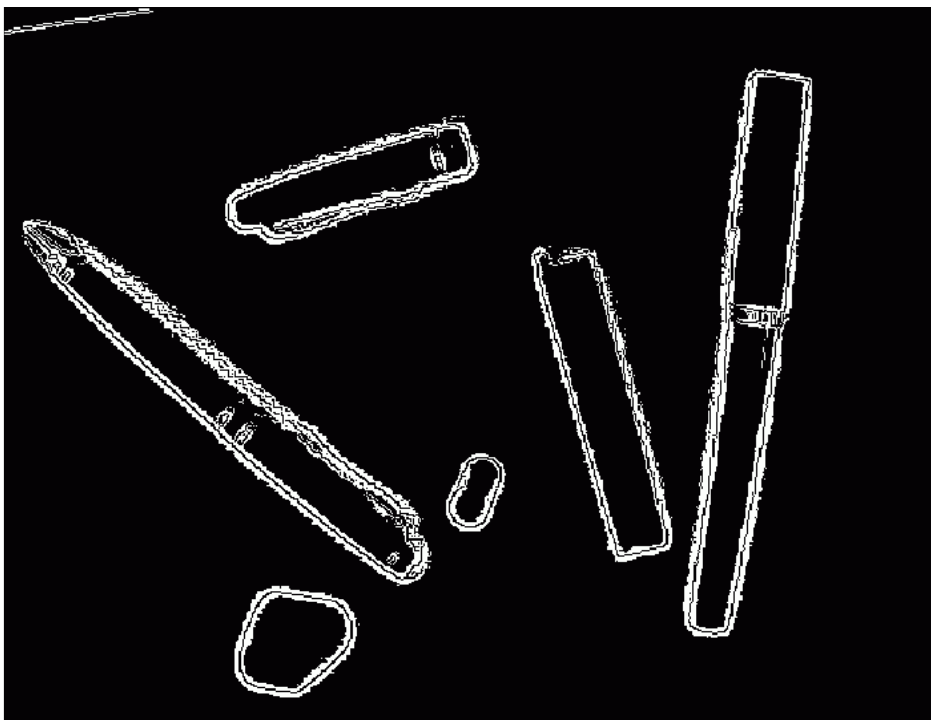
				20			
				20			
				20			
				20			
20	20	20	20	20			

(aus Bässmann & Kreys 1998)

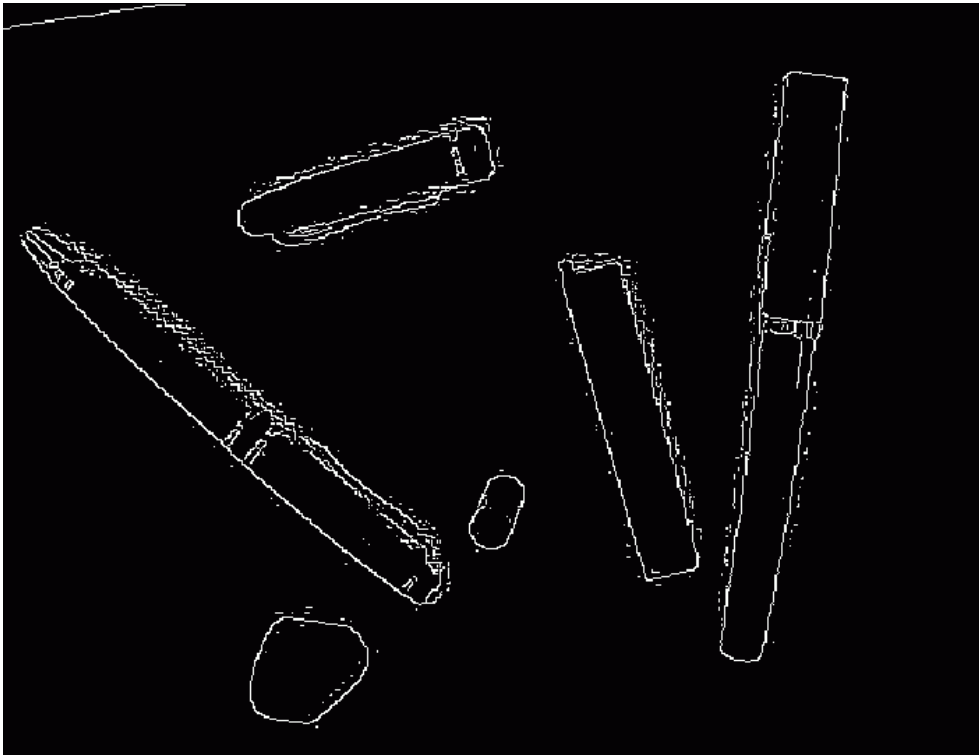
Bildbeispiel:



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.

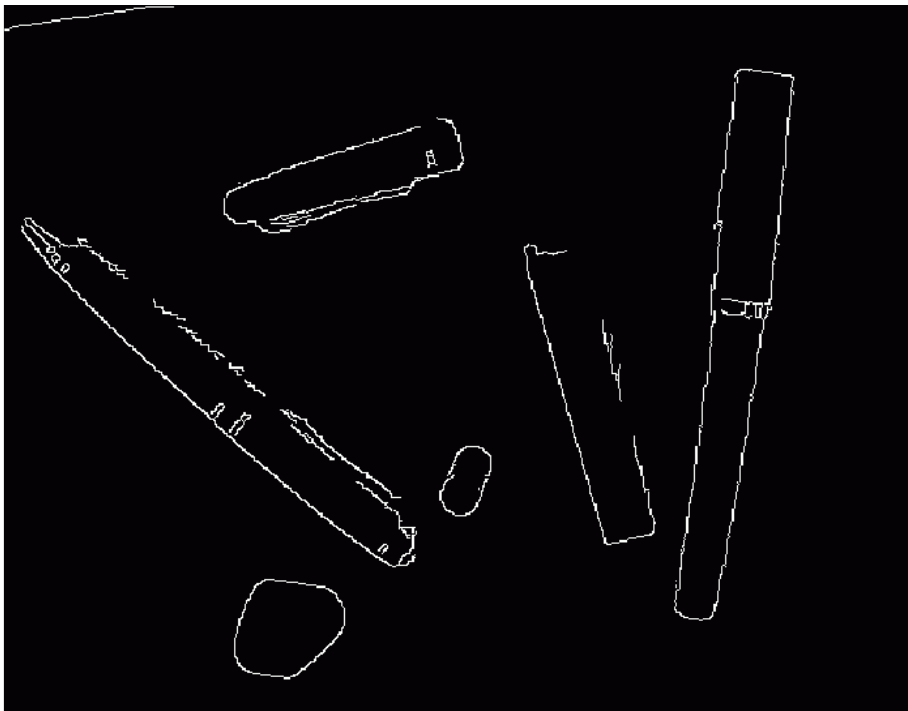
## Nachbearbeitung der verdünnten Kanten:

- evtl. Herausschneiden 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)

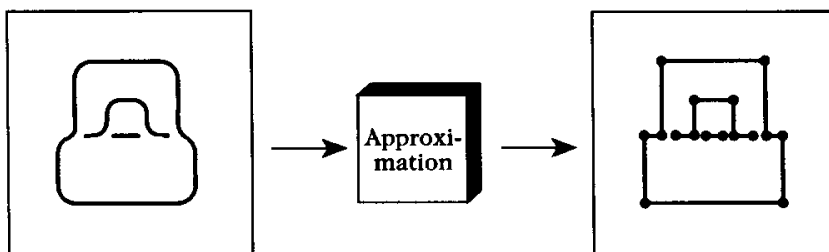
bisher nur Listen von Konturpunkten, noch keine Geradenstücke (oder andere parametrische Kurven)

nächster Schritt:

### *Konturapproximation*

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

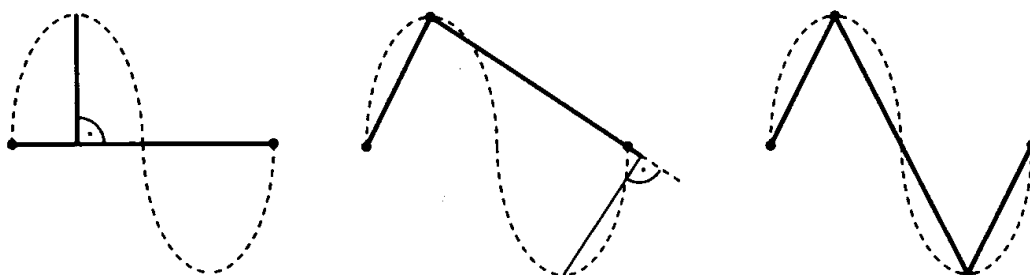
- ermöglicht Aussagen auf höherer Ebene, z.B. Parallelität von Konturen
- Eignung zur Klassifikation



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.



## alternatives Verfahren zur Konturfindung:

### "Aktive Konturen"

Active Contours: auch snakes, deformable contours, und minimum-energy contours genannt

Annahmen:

- Keine genaue Modellierung der Kontur mit ausreichender Genauigkeit möglich
- Die Abweichungen in der Form lassen sich nicht statistisch erfassen
- Die gesuchte Kontur ist glatt

Active Contours sind eine Klasse von Verfahren, die von einer initialen Kontur ausgehend die beste Kontur mittels Minimierung einer sog. Energiefunktion bestimmen. In diese Energiefunktion fließt sowohl low-level als auch high-level Wissen ein.

(Jiang 2007)



### Problemstellung:

Die parametrische Beschreibung einer Kontur:

$$v(s) = (x(s), y(s)), \quad s \in [0, 1]$$

$x(s), y(s)$ : Koordinaten entlang der Kontur. Bsp. für einen Kreis:

$$v(s) = (r \cos(2\pi s), r \sin(2\pi s)), \quad s \in [0, 1]$$

Energiefunktion einer Active Contour als Kombination interner und externer Kräfte:

$$E_{kontur} = \int_0^1 \left[ \underbrace{\alpha(s) \cdot E_1 + \beta(s) \cdot E_2}_{E_{intern}} + \underbrace{\gamma(s) \cdot E_3}_{E_{extern}} \right] ds$$

● Interne Energie:

$$E_{intern} = \alpha(s) \cdot E_1 + \beta(s) \cdot E_2, \quad E_1 = \left\| \frac{dv}{ds} \right\|^2, \quad E_2 = \left\| \frac{d^2v}{ds^2} \right\|^2$$

Großer Wert für  $\alpha(s)$ : bewirkt ein Zusammenziehen der Kurve  
Großer Wert für  $\beta(s)$ : das Streben hin zu einer kreisförmigen Gestalt  
Beide Werte müssen abhängig von der Form der zu detektierenden Kontur eingestellt werden

● Externe Energie:

$$E_{extern} = \gamma(s) \cdot E_3, \quad E_3 = -\|\nabla f(x, y)\|^2$$

Die Terme  $\alpha(s)$ ,  $\beta(s)$ , und  $\gamma(s)$  kontrollieren den relativen Einfluss der drei Faktoren

### Diskrete Problemstellung:

Input: Grauwertbild  $f(x, y)$  und Kantenstärke  $s(x, y)$   
initiale Kontur bestehend aus  $p_1, p_2, \dots, p_n$

Output: Die Position von  $p_1, p_2, \dots, p_n$  wird derart verändert, dass die Energiefunktion:

$$\sum_{i=1}^n (\alpha_i E_1 + \beta_i E_2 + \gamma_i E_3)$$

minimiert wird.

$$E_1 = (\bar{d} - \|p_i - p_{i-1}\|)^2, \quad (p_0 \equiv p_n)$$

$$\bar{d} = \frac{1}{n} \cdot \sum_{i=1}^n \|p_i - p_{i-1}\| \quad (\text{durchschnittliche Segmentlänge})$$

(Die direkte Version  $E_1 = \|p_i - p_{i-1}\|^2$  führt zu einer übermäßigen Annäherung der Punkte zu einem Cluster)

$$E_2 = \|p_{i-1} + p_{i+1} - 2p_i\|^2; \quad E_3 = -\|\nabla f(p_i)\|^2$$

## Greedy Algorithmus nach Williams/Shah: Hauptkomponenten

### • Greedy Minimierung:

Bei jedem Punkt  $p_i$ ,  $i=1, 2, \dots, n$ , wird dessen Nachbarschaft  $N(p_i)$  (typischerweise  $3 \times 3$ ,  $5 \times 5$  oder  $7 \times 7$ ) untersucht. Derjenige Punkt aus  $N(p_i)$  mit dem kleinsten Energiewert ersetzt  $p_i$ .

Achtung: Normierung der Energieterme mit einem neuen Wertebereich  $[0,1]$  ist wichtig, damit die drei Terme quantitativ überhaupt vergleichbar werden. Für alle  $p \in N(p_i)$ :

$$E_1^*(p) = \frac{E_1(p)}{\max_{q \in N(p_i)} E_1(q)} \in [0, 1]; \quad E_2^*(p) = \frac{E_2(p)}{\max_{q \in N(p_i)} E_2(q)} \in [0, 1]$$

$$\|\nabla f(p)\|^* = \frac{\|\nabla f(p)\| - m}{M - m} \in [0, 1]$$

$$M = \max_{q \in N(p_i)} \|\nabla f(q)\|, \quad m = \min_{q \in N(p_i)} \|\nabla f(q)\|$$

Die normierten Werte gehen in die Wertung ein

### • Handhabung von Ecken:

In einer zweiten Phase jeder Iteration sucht der Algorithmus nach Ecken, die sich als lokale Maxima der Krümmung  $k_i$  entlang der Kontur:

$$k_i = \|p_{i-1} + p_{i+1} - 2p_i\|$$

auszeichnen. Falls ein Punkt  $p_i$  als Ecke befunden wird, so wird  $\beta_i$  auf null gesetzt. Dadurch bekommt man die Möglichkeit, auch stückweise glatte Kurven zu erreichen.

**while** (keines der Terminierungskriterien erfüllt) **do**

1. Berechne  $\bar{d}$ .
2. Für jeden Punkt  $p_i$ ,  $i = 1, 2, \dots, n$ , suche in der Nachbarschaft  $N(p_i)$  nach demjenigen Punkt  $p$ , so dass der Beitrag von  $p$  zur gesamten Energiefunktion  $\alpha_i E_1 + \beta_i E_2 + \gamma_i E_3$  minimal ist. Dann wird  $p_i$  durch  $p$  ersetzt.
3. Für jeden Punkt  $p_i$ ,  $i = 1, 2, \dots, n$ , schätze die Krümmung  $k_i$  in  $p_i$ :

$$k_i = \|p_{i-1} + p_{i+1} - 2p_i\|$$

Suche  $p_i$  aus, die jeweils ein lokales Maximum sind (d.h.  $k_i > k_{i-1}$  und  $k_i > k_{i+1}$ ). Falls  $k_i \geq T_1$  und  $\|\nabla f(p_i)\| > T_2$ , dann setze  $\beta_i$  auf null (für die nachfolgenden Iterationen).

---

### Terminierungskriterien:

- Der überwiegende Teil der Punkte erfährt keine Verschiebung mehr
- Eine vorgegebene maximale Anzahl von Iterationen ist erreicht

### Gewichtung der Energieterme:

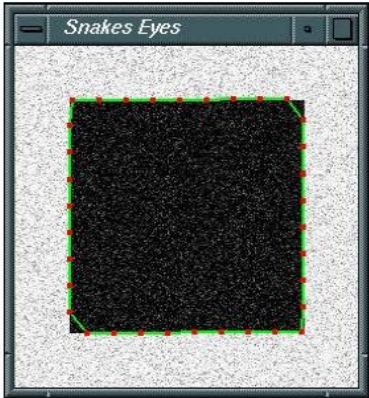
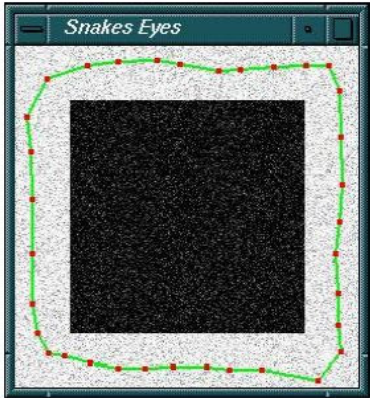
$$\alpha_i = \beta_i = 1.0, \quad \gamma_i = 1.2, \quad i = 1, 2, \dots, n$$

(andere Gewichtungen möglich)

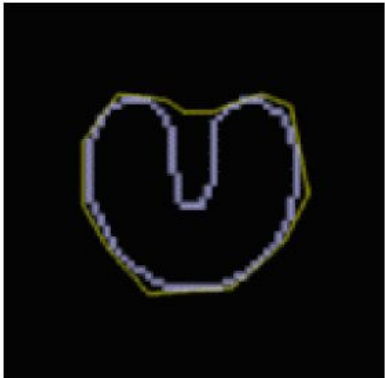
Im Laufe der Iterationen kann  $\beta_i$  für den Fall, dass  $p_i$  als Ecke befunden wird, auf null gesetzt werden



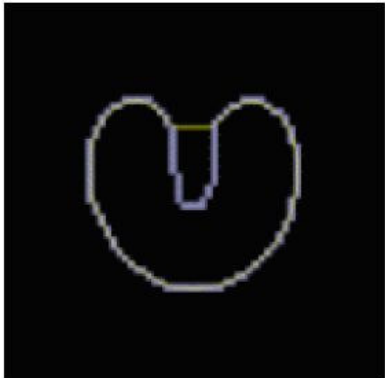
Beispiele:



Traditionelle Active Contours haben Probleme mit konkaven Konturen selbst bei guter Initialisierung



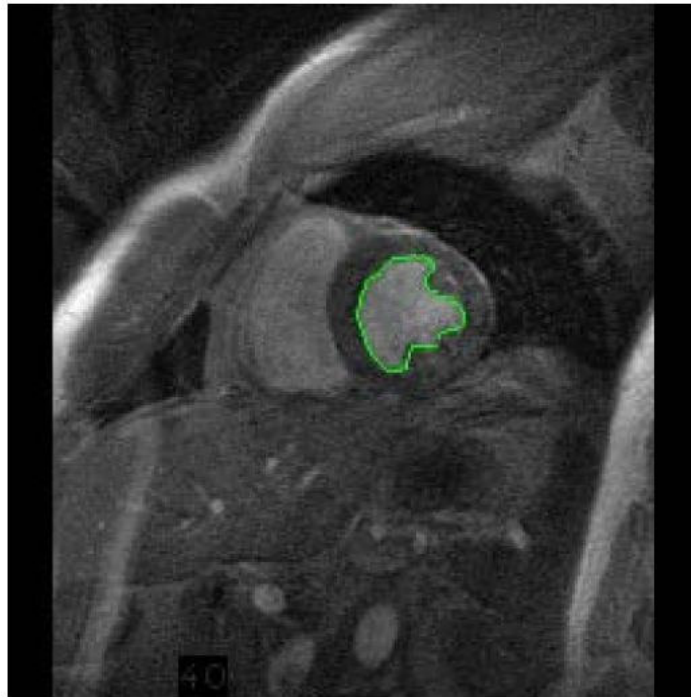
Initialisierung



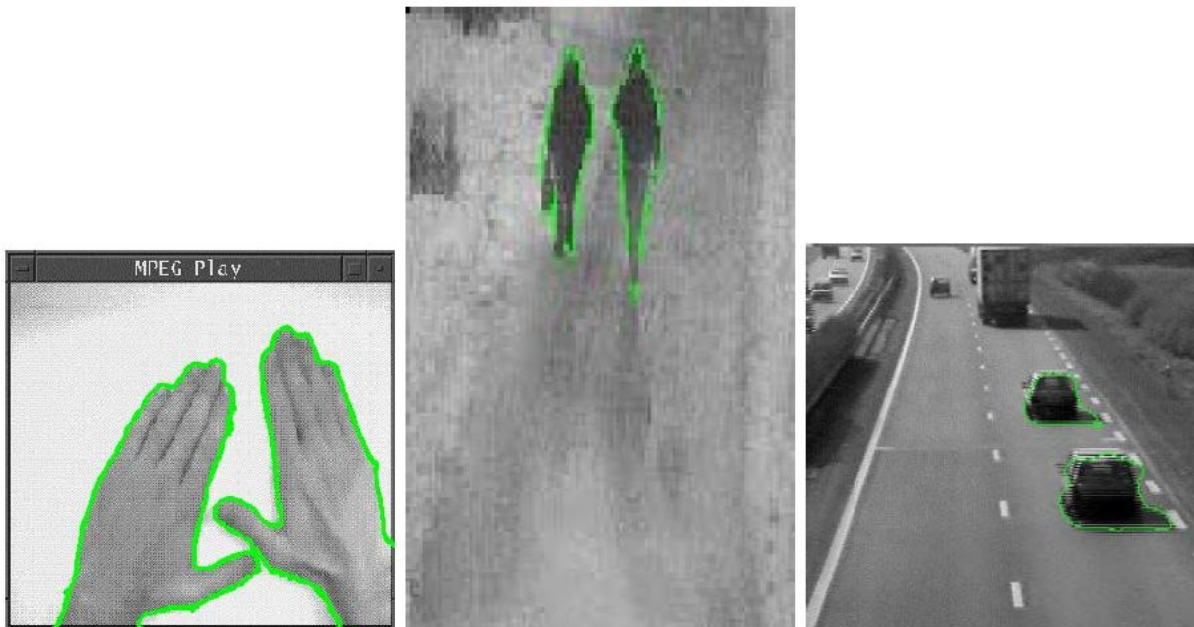
Ergebnis



**Anwendungen:** Segmentierung, insb. in der medizinischen Bildverarbeitung



**Anwendung:** Tracking



(aus Jiang 2007)

## *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:

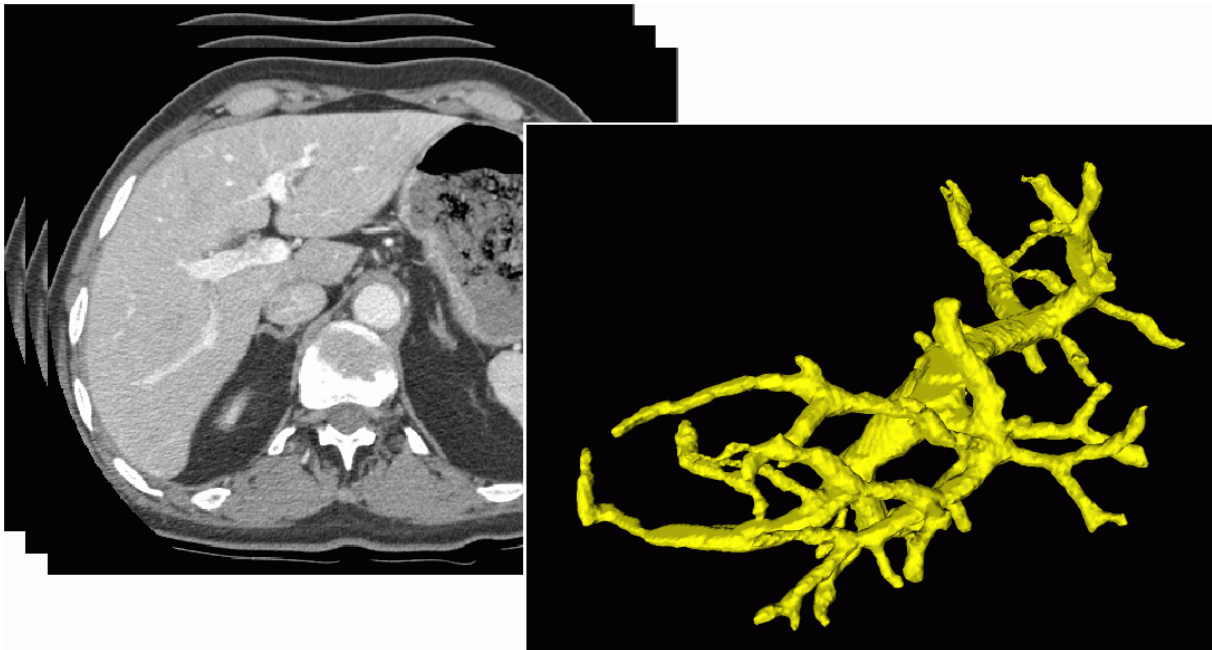
- Region Growing
- Merging
- Boundary Melting
- Splitting
- Split and Merge
- clusteranalytische Verfahren
- Matching

*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)

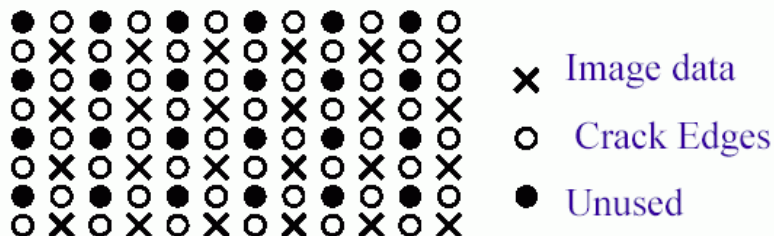


### Region Merging:

- **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  $s_{ij} < T_1$

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

$$= 1 \text{ otherwise}$$

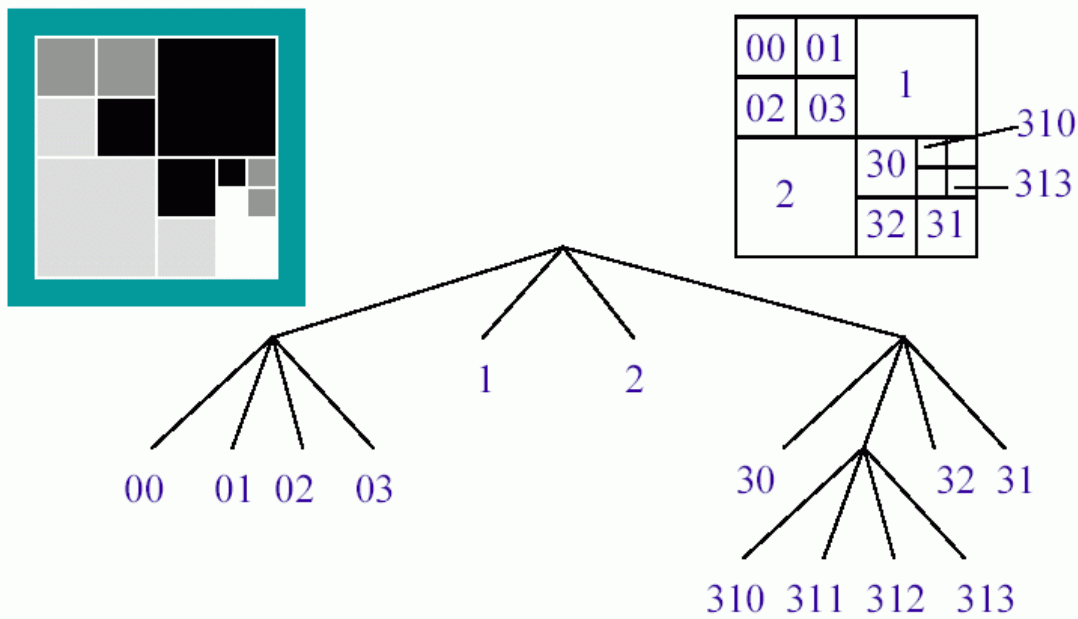
- Zwei Regionen werden verschmolzen, wenn ein **signifikanter Teil** der gemeinsamen Grenze aus **schwachen Kantenpixeln** besteht, unter Berücksichtigung der Regionengröße
- Bzw. auch unabhängig von der Regionengröße
- **Startsegmentierung** in Regionen mit konstantem Grauwert, Supergridstruktur mit 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

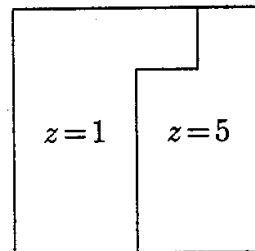
einfaches Beispiel:

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

(a)

1		1	5
1		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!