

Kapitel 4: Bildsegmentierung

Prof. Dr.-Ing. Thomas Schultz

URL: <http://cg.cs.uni-bonn.de/schultz/>

E-Mail: schultz@cs.uni-bonn.de

Büro: Friedrich-Hirzebruch-Allee 6, Raum 2.117

2./9./16. Dezember 2024

4.1 Problemstellung und Evaluierung

Zielsetzung

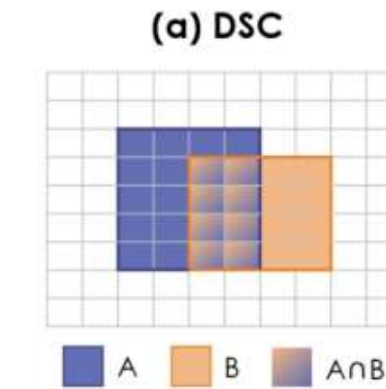
- Grundlegende Aufgabe bei der **Interpretation** von Bildern ist das Erkennen relevanter **Bildinhalte** (z.B. Organ, Tumor, Gewebetyp)
 - Diese lassen sich meist nicht aus den Intensitäten einzelner Pixel schließen, sondern erfordern die sinnvolle **Gruppierung** von Pixeln
- **Segmentierung** bezeichnet eine vollständige und überdeckungs-freie Zerlegung eines Bildes nach bestimmten Kriterien
 - z.B. Regionen- oder Kantenbasiert
 - Berücksichtigung von Vorwissen, z.B. über die gesuchte Form
- Bildet u.a. die Grundlage von **Quantifizierung** (z.B. Volumen, Form, Textur) und **Behandlungsplänen**

Verschiedene Arten der Segmentierung

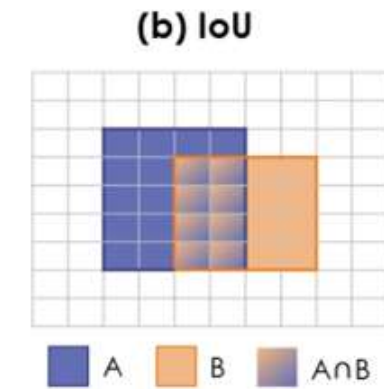
- **Semantische Segmentierung** ordnet jedem Pixel eine klare Bedeutung zu („Label“)
- **Instanz-Segmentierung** zerlegt ein Bild pixelgenau in einzelne Objekte (z.B. einzelne Zellen in einem Gewebeschnitt)
- Die Kombination beider Aufgaben (Zerlegung in Instanzen und Benennung/Labeling dieser) wird manchmal als **panoptische Segmentierung** bezeichnet

Evaluierung von Segmentierungen: Überlapp

- Den **Überlapp** zwischen
 - Segmentierung A und
 - Referenz („Ground Truth“) Bquantifiziert man häufig per
 - Dice-Score (DSC) oder
 - Verhältnis von Schnitt- und Vereinigungsmenge
 - IoU = Intersection over Union



$$\begin{aligned} \text{DSC}(A,B) &= \frac{2 \text{ (Intersection) }}{\text{Area of A} + \text{Area of B}} \\ &= \frac{2 |A \cap B|}{|A| + |B|} \end{aligned}$$



$$\begin{aligned} \text{IoU}(A,B) &= \frac{\text{Intersection}}{\text{Area of Union}} \\ &= \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \\ &= \frac{|A \cap B|}{|A \cup B|} \end{aligned}$$

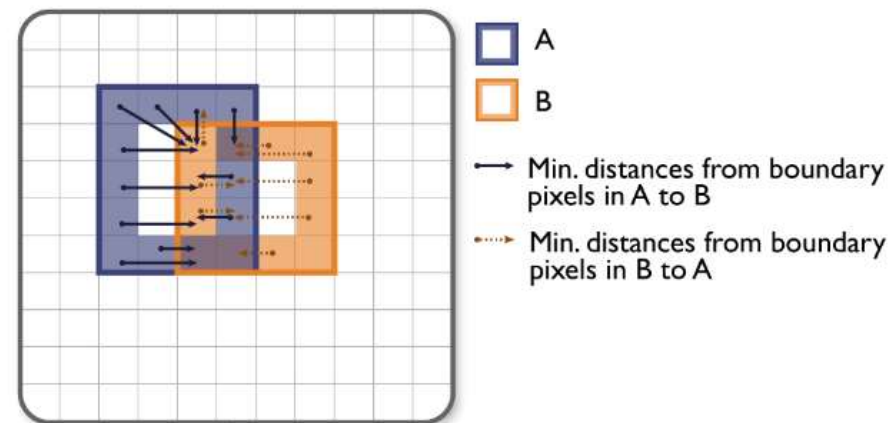
Evaluierung von Segmentierungen: Kontur-basiert

- Die maximale Abweichung der **Konturen** von

- Segmentierung **A** und
- Referenz („Ground Truth“) **B**

quantifiziert man häufig per

- Hausdorff-Distanz (HD)
- Robuster: HD mit Perzentil



(a) Hausdorff Distance (HD)

$$d(a,b) = \min_{b \in B} d(a,b)$$

$$HD(A,B) = \max \left\{ \max_{a \in A} d(a,B), \max_{b \in B} d(A,b) \right\}$$

$$= \max \left\{ \left\{ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \right\}, \left\{ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \right\} \right\}$$

\uparrow max \uparrow max

(b) Hausdorff Distance 95 percentile (HD95)

$$d_{95}(A,B) = x_{95} \left\{ \min_{b \in B} d(a,b) \right\}_{a \in A}$$

$$HD95(A,B) = \max \left\{ d_{95}(A,B), d_{95}(B,A) \right\}$$

$$= \max \left\{ \left\{ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \right\}, \left\{ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \right\} \right\}$$

\uparrow x_{95} \uparrow x_{95}

4.2 Grundlegende Verfahren

Schwellenwert-Segmentierung

- **Histogrammbasierte Segmentierung** klassifiziert die Pixel nur aufgrund ihrer individuellen Intensität
 - Anwendung von einem oder mehreren **Schwellenwerten**
 - **Täler im Histogramm** sind häufig sinnvolle Schwellenwerte

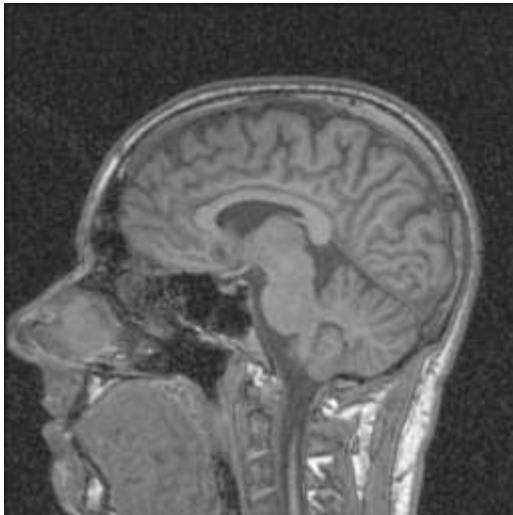
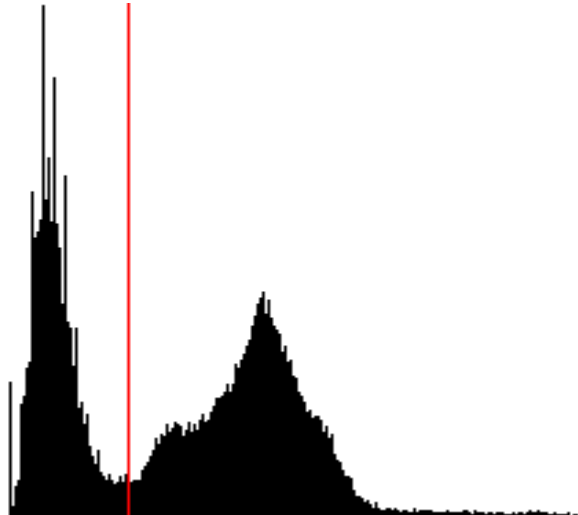


Bild $g(\mathbf{x})$



Histogramm von g mit
Schwellenwert $\theta = 75$



Binarisiertes Bild
 $g(\mathbf{x}) \geq \theta$

Otsu-Verfahren zur Schwellenwert-Bestimmung

- **Idee von Otsu:** Optimaler Schwellenwert θ sollte das Verhältnis σ_B^2/σ_W^2 der Varianz zwischen (*between*) und innerhalb (*within*) der Klassen maximieren
- **Berechnung** basiert auf Intensitäten g , Histogramm $h(g)$
 - Zahl der Pixel in den beiden Klassen:

$$N_1 = \sum_{g=0}^{\theta-1} h(g), \quad N_2 = \sum_{g=\theta}^{g_{\max}} h(g), \quad N = N_1 + N_2$$

- Mittelwerte

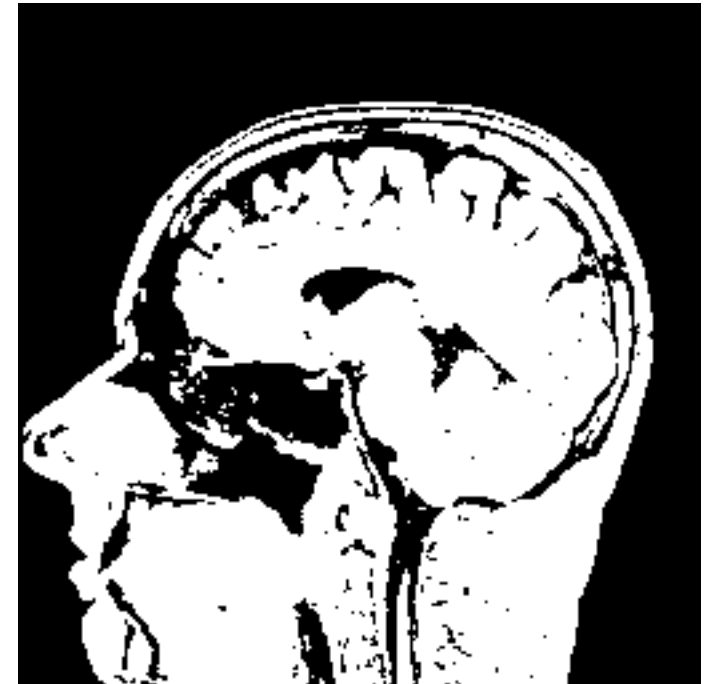
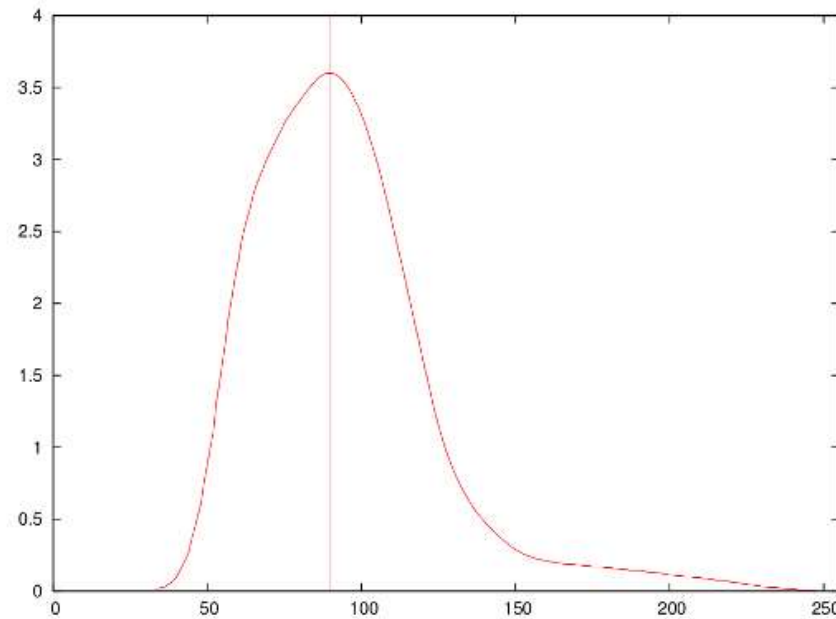
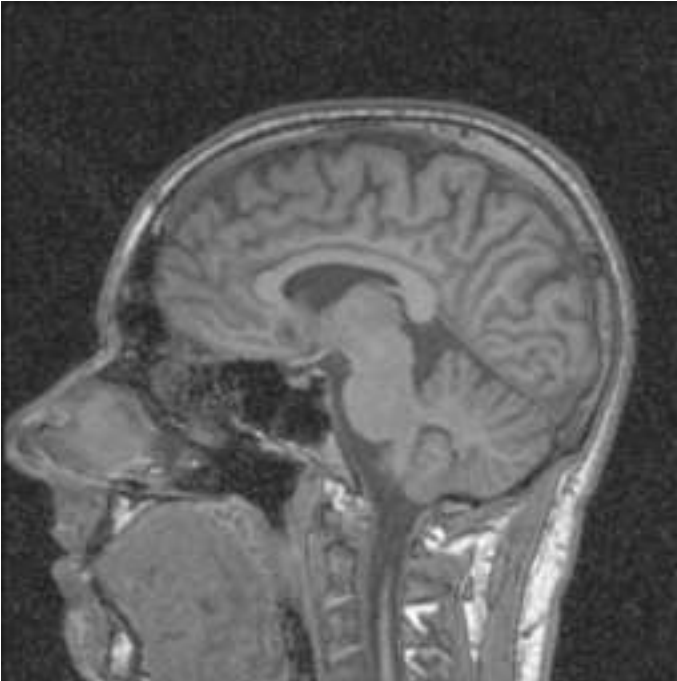
$$\mu_1 = \frac{1}{N_1} \sum_{g=0}^{\theta-1} g \cdot h(g), \quad \mu_2 = \frac{1}{N_2} \sum_{g=\theta}^{g_{\max}} g \cdot h(g), \quad \mu = \frac{N_1\mu_1 + N_2\mu_2}{N}$$

- Varianzen

$$\sigma_1^2 = \frac{1}{N_1} \sum_{g=0}^{\theta-1} (g - \mu_1)^2 h(g), \quad \sigma_2^2 = \frac{1}{N_2} \sum_{g=\theta}^{g_{\max}} (g - \mu_2)^2 h(g)$$
$$\sigma_B^2 = \frac{N_1(\mu_1 - \mu)^2 + N_2(\mu_2 - \mu)^2}{N}, \quad \sigma_W^2 = \frac{N_1\sigma_1^2 + N_2\sigma_2^2}{N}$$

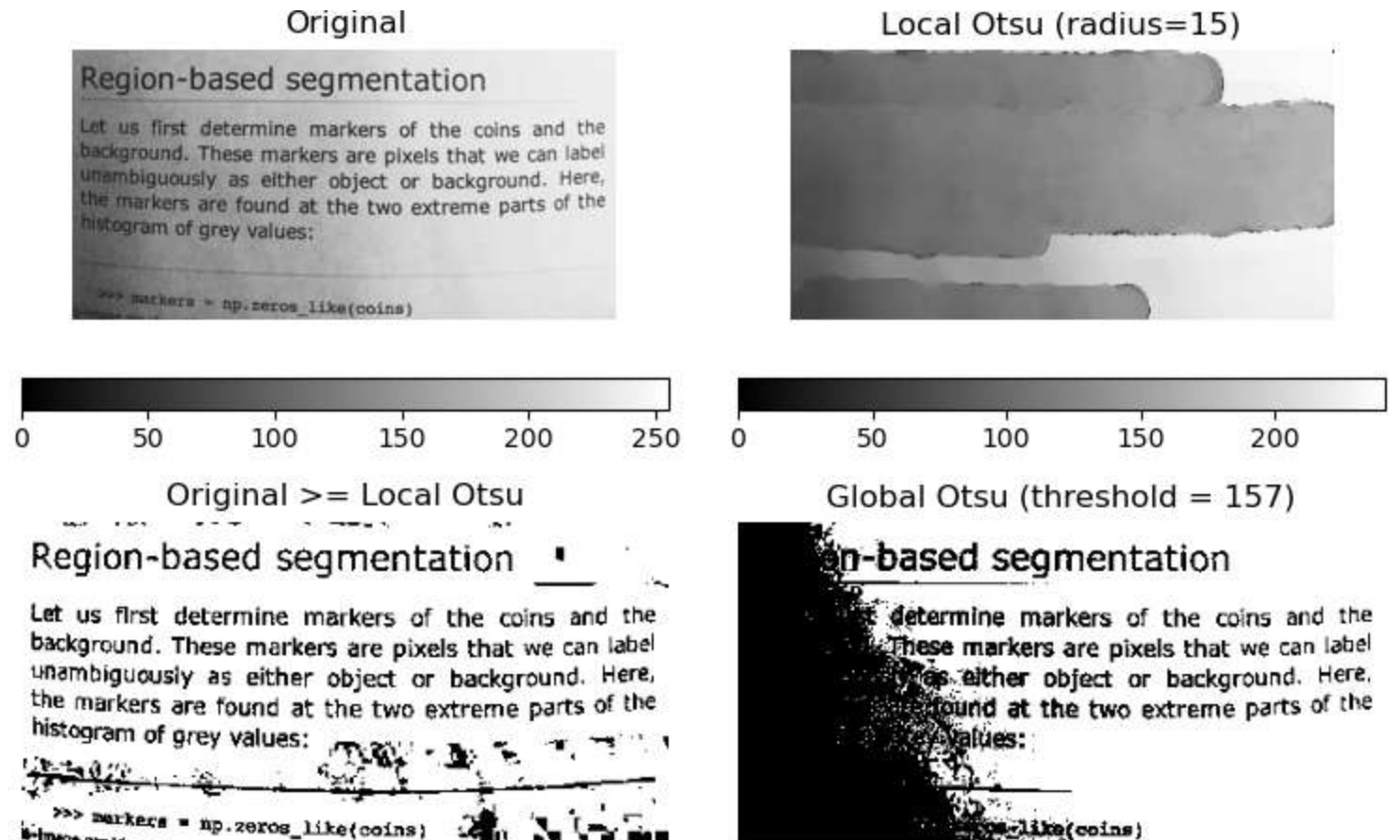
Ergebnis des Otsu-Verfahrens

- Optimierung von σ_B^2 / σ_W^2 durch Ausprobieren aller θ ergibt in unserem Beispielbild $\theta = 90$



Adaptive Schwellenwerte

- Bei ungleichmäßigen Hintergründen können **Schwellenwerte adaptiv** aufgrund des Histogramms lokaler Nachbarschaften bestimmt werden



Vor- und Nachverarbeitung

- **Probleme der histogrammbasierten Segmentierung:**

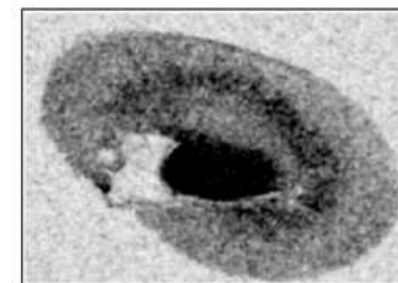
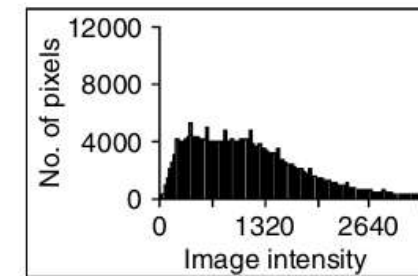
- Variationen um den Schwellenwert führen leicht zu kleinen Löchern oder Inseln
- Rauschen kann Täler im Histogramm verwischen

- Häufige Schritte zur **Vor- und Nachverarbeitung:**

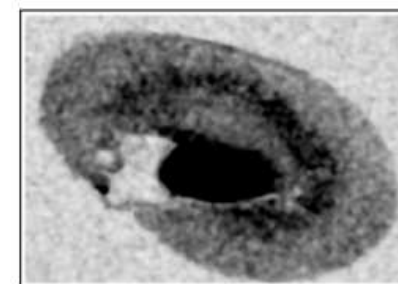
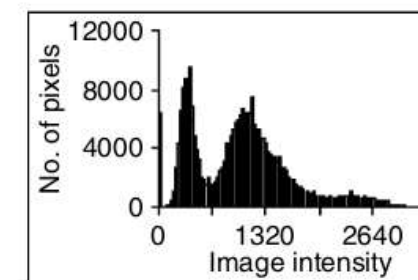
- **Glättung** des Bildes (s. Kapitel 1)
- Nachbearbeitung der Segmentierungsmaske mit **morphologischen Operationen**
- Analyse von **Zusammenhangskomponenten**



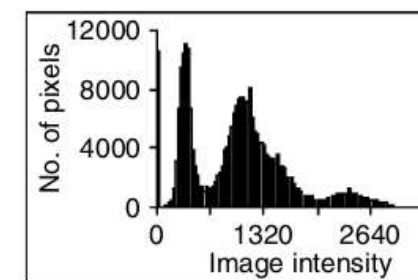
(a)



(b)



(c)

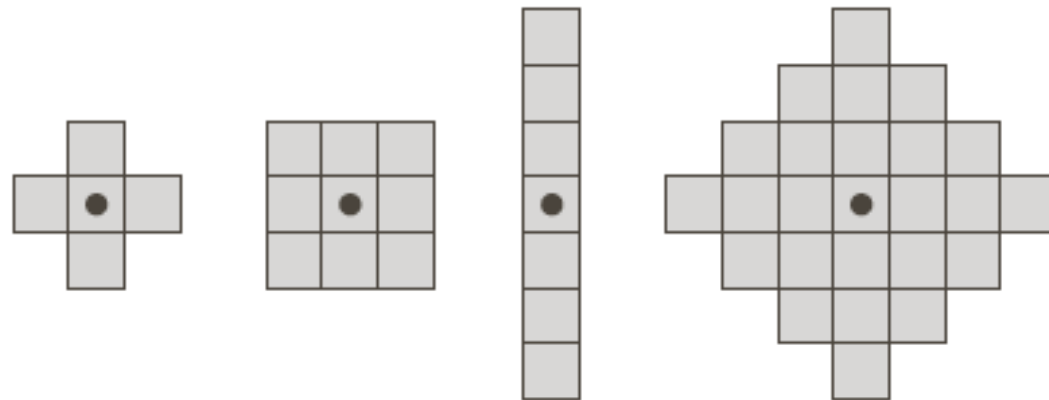


Medianfilterung: 7×7 (b) bzw. 9×9 (c) 12

Morphologische Bildverarbeitung

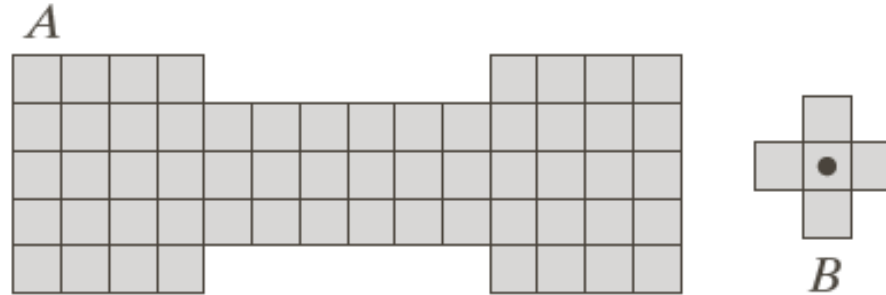
- **Morphologische Operationen**

- Können auf Graustufen- oder Binärbilder angewandt werden (0=Hintergrund, 1=Vordergrund)
- Nutzen ein **Strukturelement**, dessen Ankerpunkt – ähnlich dem Kern einer Kreuzkorrelation – auf alle Pixel verschoben wird
 - Form des Strukturelements kann an die relevanter Objekte angepasst werden
 - Grundoperationen ähneln der Median-Filterung, aber nutzen Minimum und Maximum

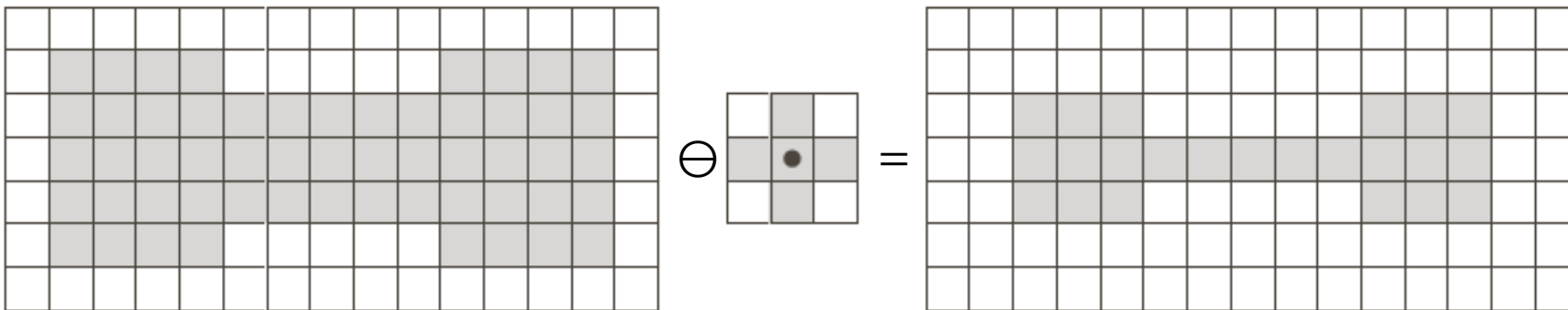


Erosion

- *Gegeben:* Binärbild A und Strukturelement B :

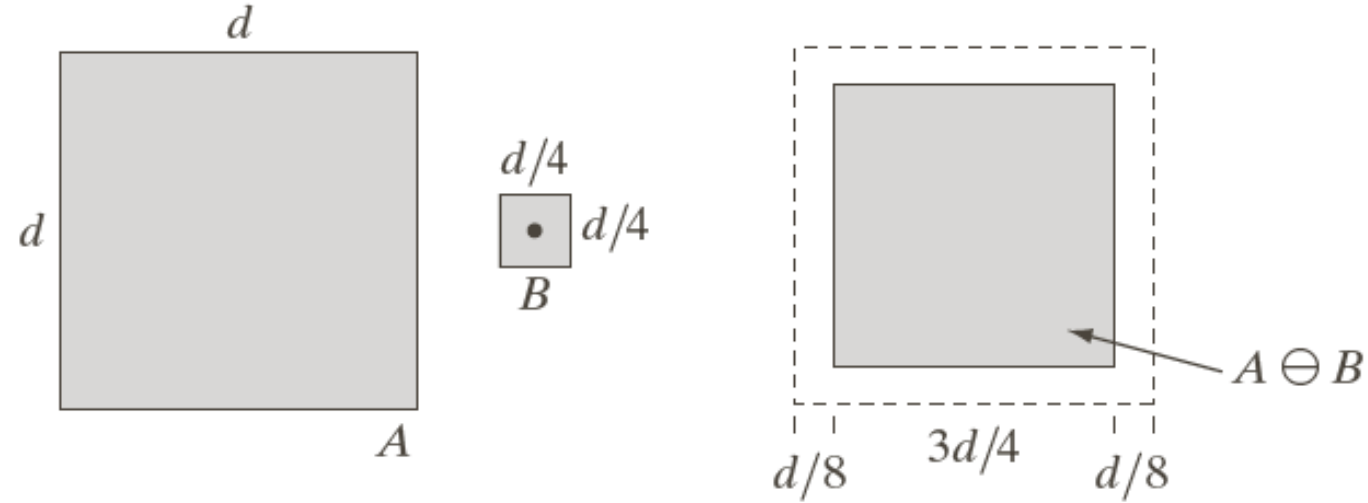


- **Erosion** \ominus entspricht Minimum-Filterung
 - Ausgabe-Pixel p ist genau dann Vordergrund, wenn das auf p zentrierte Strukturelement B vollständig im Vordergrund von A liegt
 - Am Rand wird A mit Null (Hintergrund) aufgefüllt

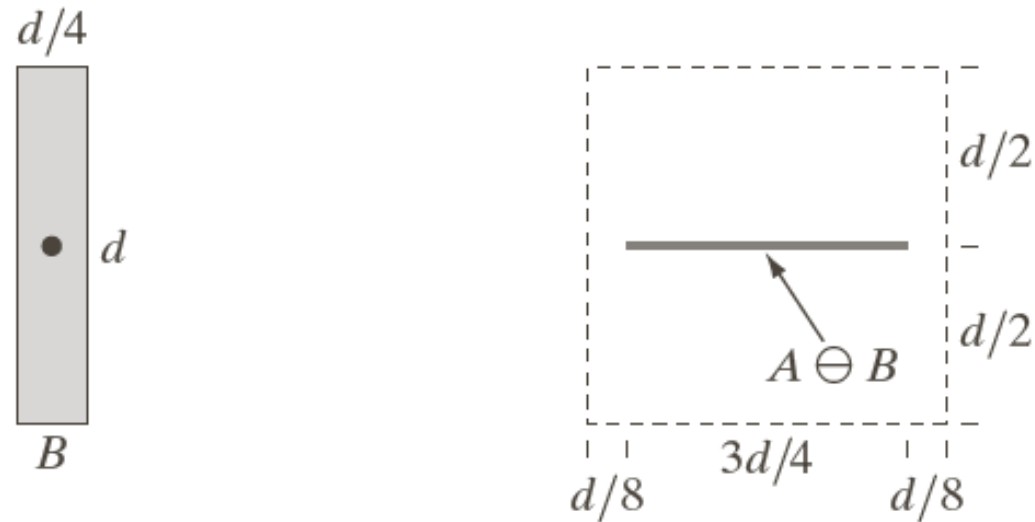


Erosion: Einfluss des Struktur-Elements

Beispiel 1

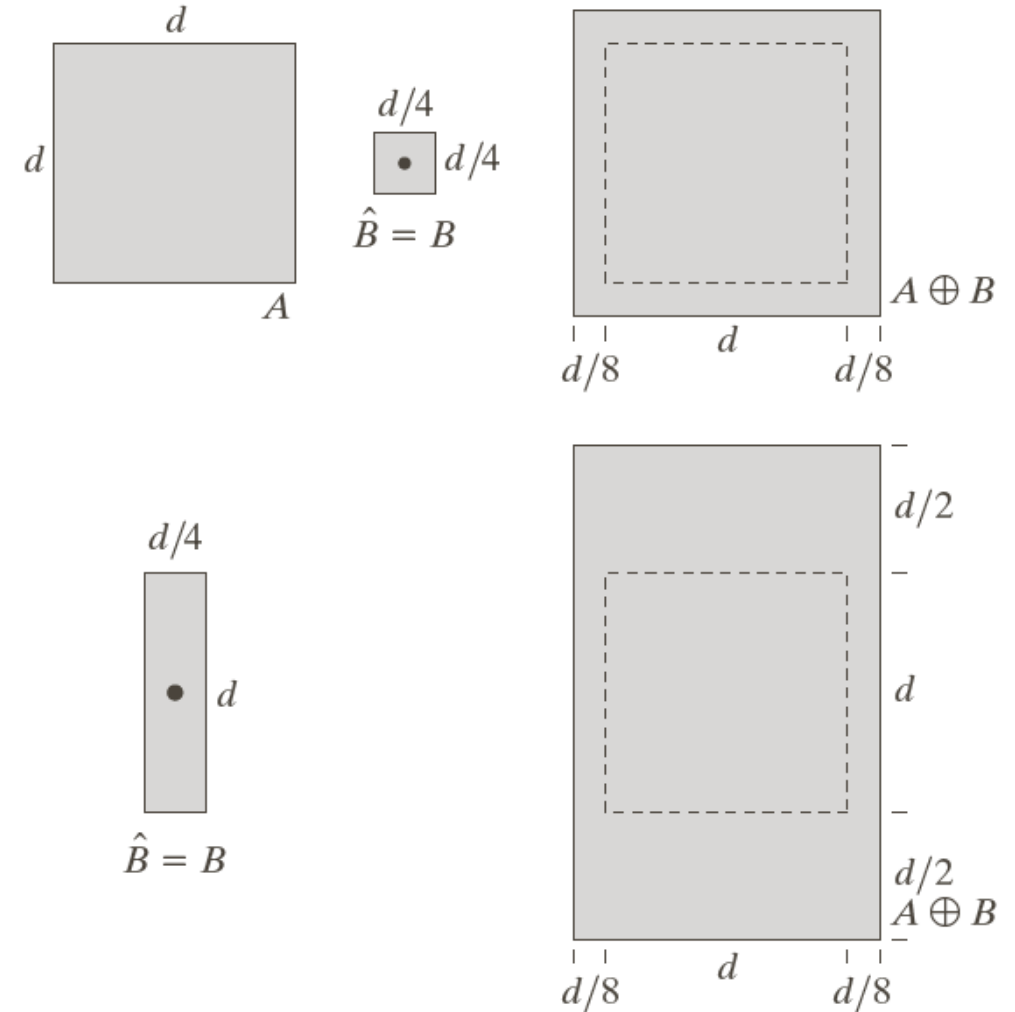


Beispiel 2



Dilatation

- **Dilatation \oplus** entspricht Maximum-Filterung
 - Ausgabe-Pixel p ist genau dann Vordergrund, wenn das auf p zentrierte Strukturelement B mit mindestens einem Vordergrund-Pixel von A überlappt



Morphologisches Öffnen und Schließen

- Definition des **morphologischen Öffnens** (Opening)

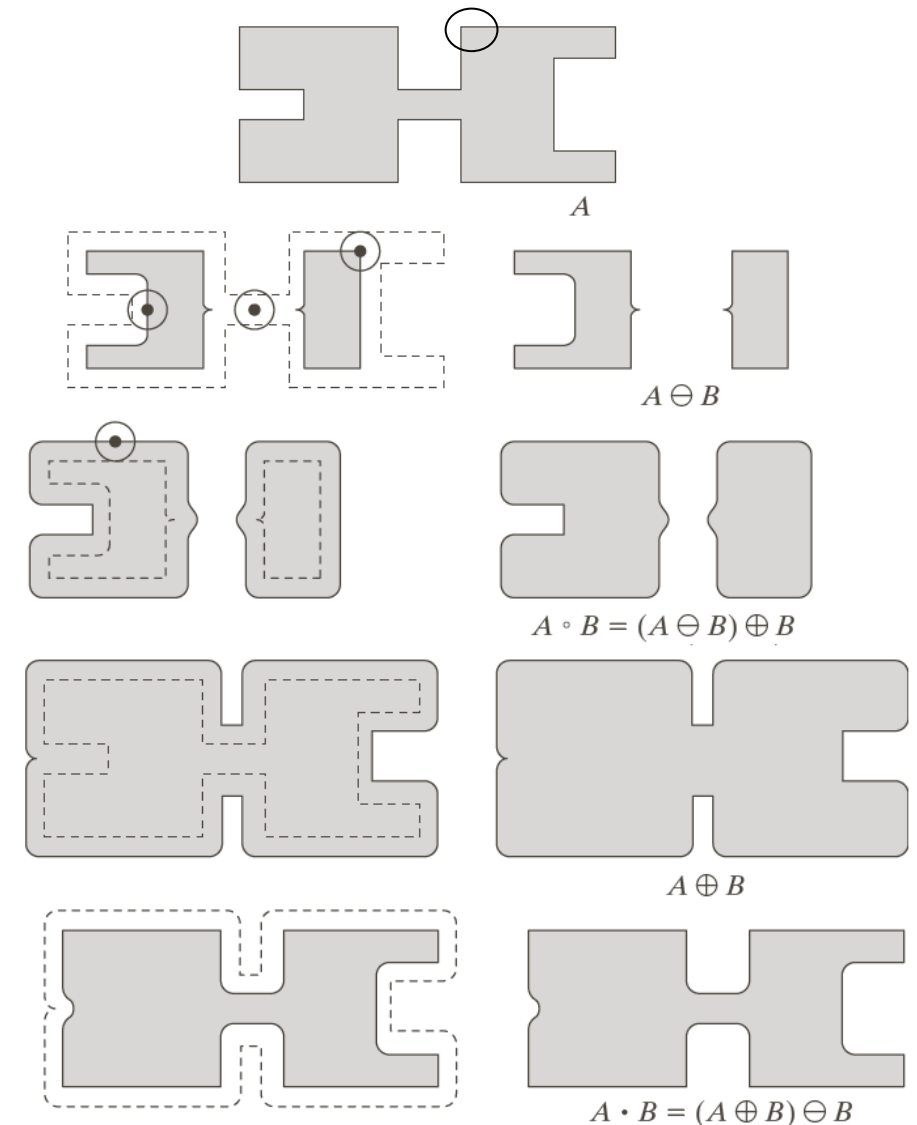
$$A \circ B = (A \ominus B) \oplus B$$

- Beseitigt schmale Vorsprünge / Verbindungen
- Glättet Objektränder

- Definition des **morphologischen Schließens** (Closing)

$$A \bullet B = (A \oplus B) \ominus B$$

- Beseitigt kleine Löcher
- Füllt schmale Lücken und enge Einbuchtungen auf



Zusammenhangskomponenten

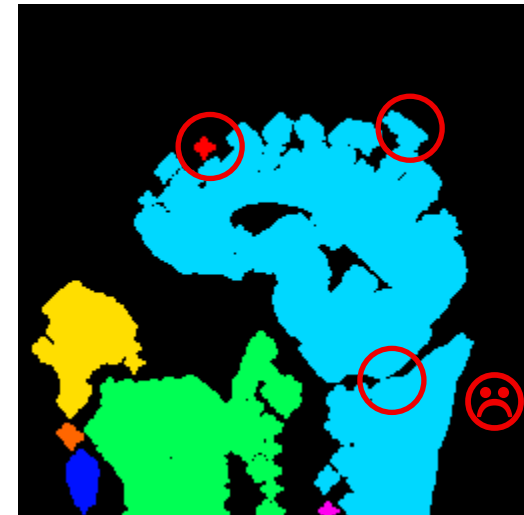
- Die Definition von **Zusammenhangskomponenten** in Binärbildern entspricht der aus der Graphentheorie
 - Vordergrund-Pixel als Knoten, Kanten verbinden benachbarte Pixel
 - Reicht in manchen Fällen aus um Objekte zu trennen
 - Leider verhindern oft wenige Pixel erwünschte Trennung / Verbindung
 - *Beispiel:* Versuch der Hirnsegmentierung als Zusammenhangskomponente nach Otsu-Schwellenwert und morphologischer Öffnung:



Binarisiertes Bild



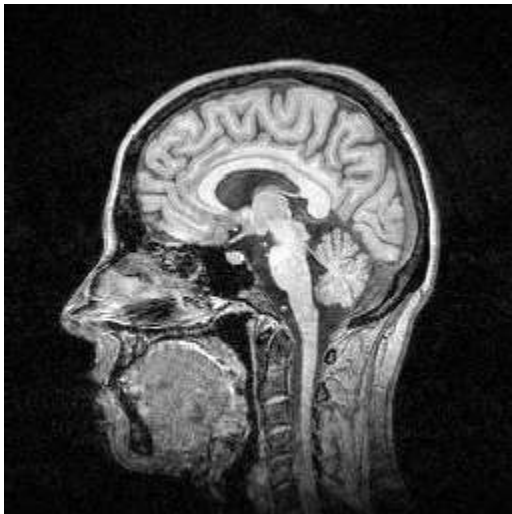
Opening mit Kreis (11x11)



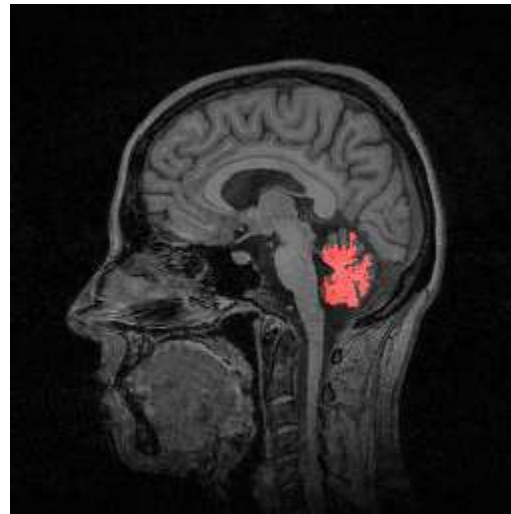
Zusammenhangskomponenten

Region Growing

- **Region Growing** fügt der Segmentierung von einem Startpunkt ausgehend so lange benachbarte Pixel hinzu, wie ein Homogenitätskriterium erfüllt ist
 - Basiert meist auf Intensitätsunterschieden (z.B. bezüglich des Startpunkts, des Nachbarpixels, des aktuellen Mittelwerts der Region)



Bild



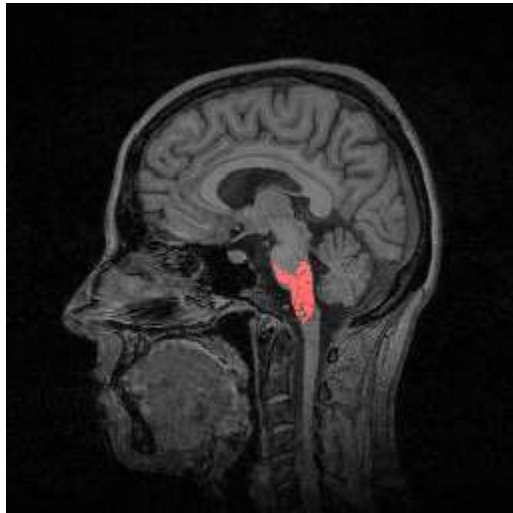
Startpunkt im Kleinhirn,
Schwelle=20



Startpunkt im Hirnstamm,
Schwelle=17

Probleme des Region Growing

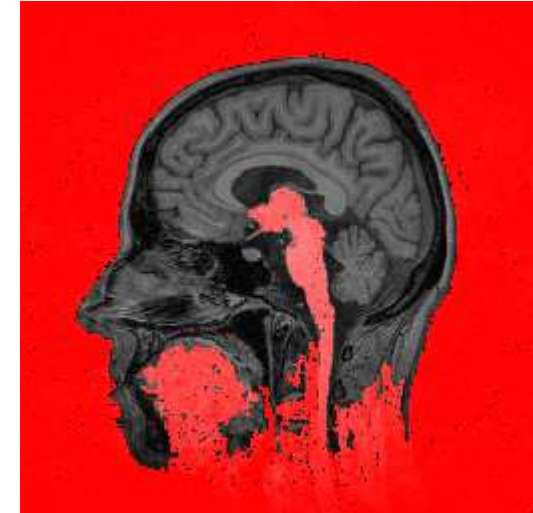
- Übliche Probleme des region growing sind
 - Plötzliches „Auslaufen“ oberhalb eines Schwellenwerts
 - Auslassen einzelner Pixel aufgrund von Bildrauschen
- Lösungsansätze ähnlich wie bei Schwellenwert-Segmentierung



Startpunkt im Hirnstamm,
Schwelle=12



Startpunkt im Hirnstamm,
Schwelle=17

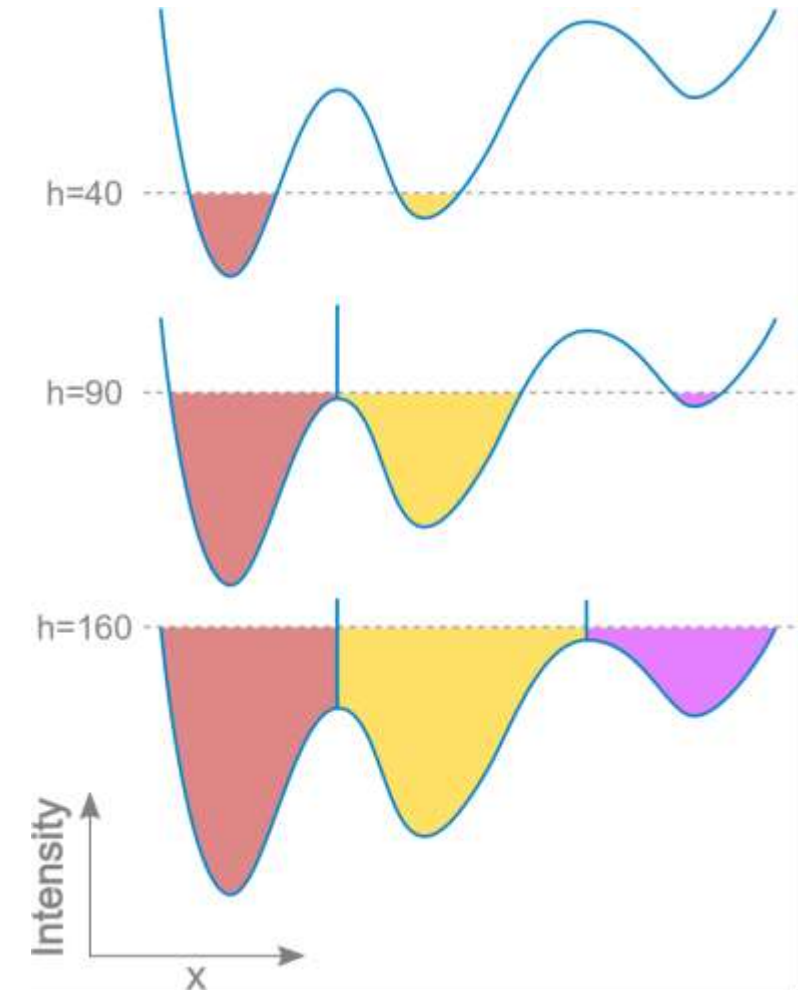


Startpunkt im Hirnstamm,
Schwelle=18

Wasserscheidentransformation

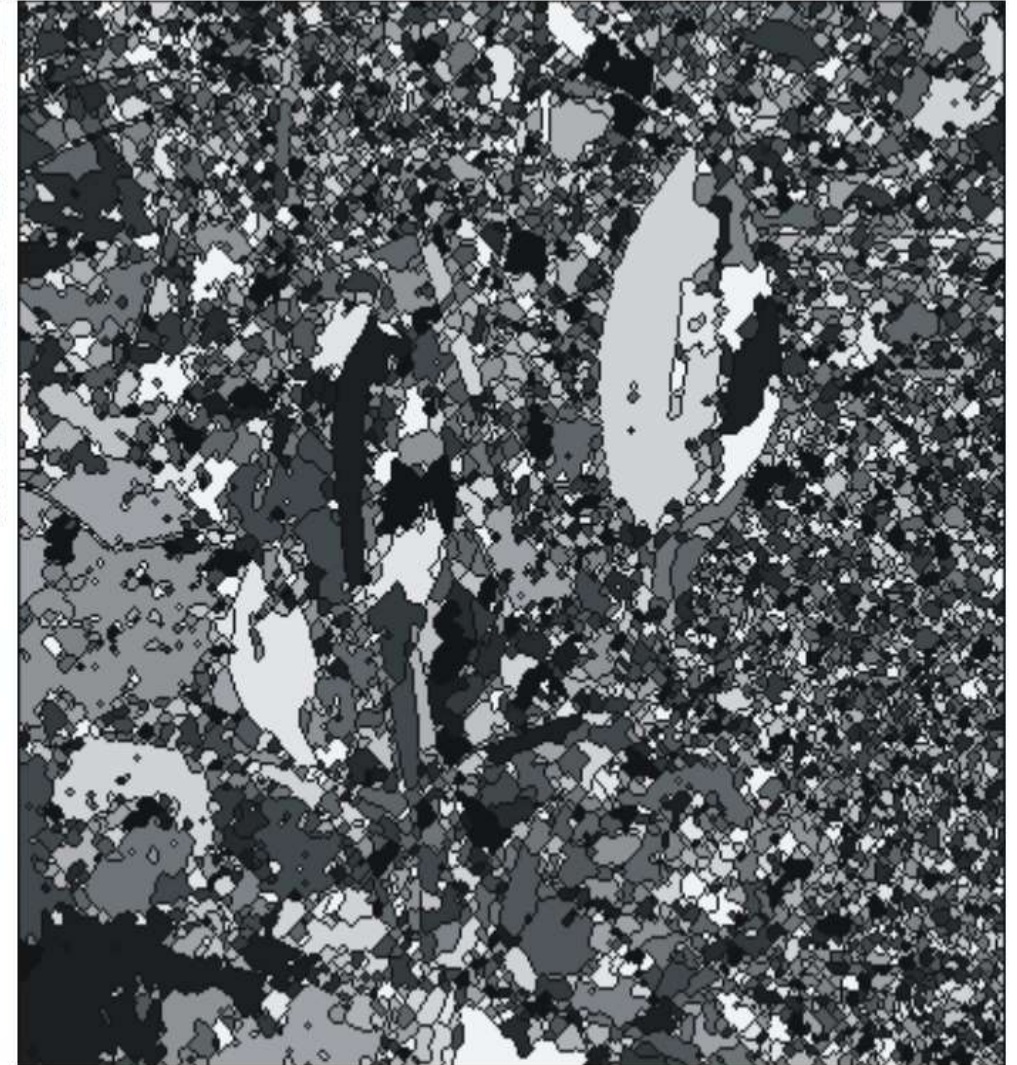
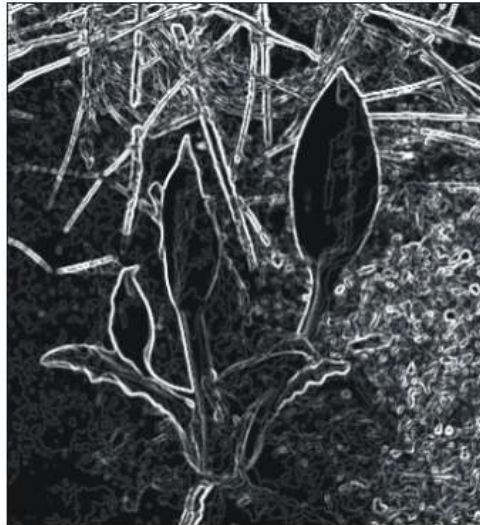
Die **Wasserscheidentransformation** (*engl.* watershed transformation) fasst alle Punkte im Bild zusammen, von denen aus ein Gradientenabstieg im selben Minimum endet

- Wenn wir Intensitäten als Höhenfeld auffassen, trennen **Wasserscheiden** im geografischen Sinne diese Gebiete
- *Häufige Vorstellung:* Steigendes Wasserniveau im Gebirge, „Dämme“ verhindern Zusammenfließen verschiedener Staubecken



Bildsegmentierung per Wasserscheidentransformation

- Wendet man die **Wasserscheiden-Transformation** auf die **Gradientenstärke** an, erhält man Wasserscheiden an Kanten
- Führt meist zu einer **Übersegmentierung** des Bildes. *Ansätze:*
 - Zusammenfassen von Regionen
 - Quellen durch Marker vorgeben

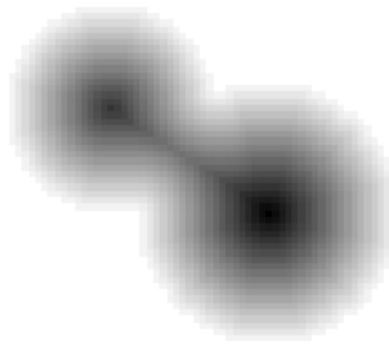


Trennung von Objekten: Distanztransformation

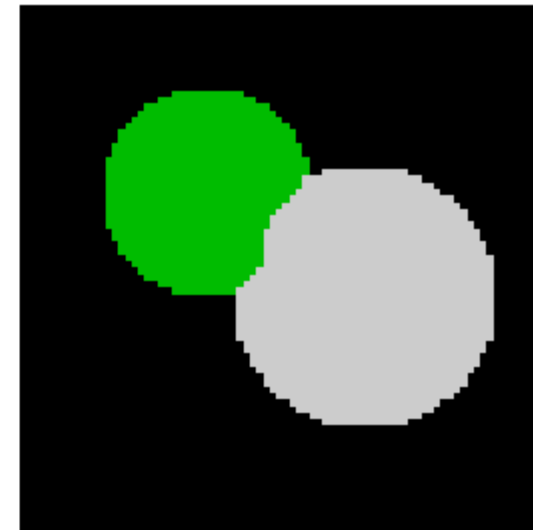
- Die **Distanztransformation** weist jedem Pixel einer binären Maske seinen Abstand vom Hintergrund zu
- Die **Wasserscheidentransformation** der Distanztransformation eignet sich dafür, sich berührende Objekte in einer Segmentierungsmaske zu trennen



Ursprüngliche Maske



Distanztransformation



Getrennte Objekte

Zusammenfassung: Einfache Segmentierungsverfahren

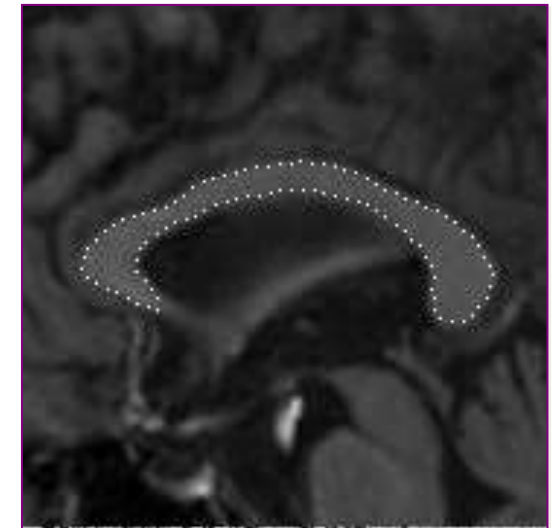
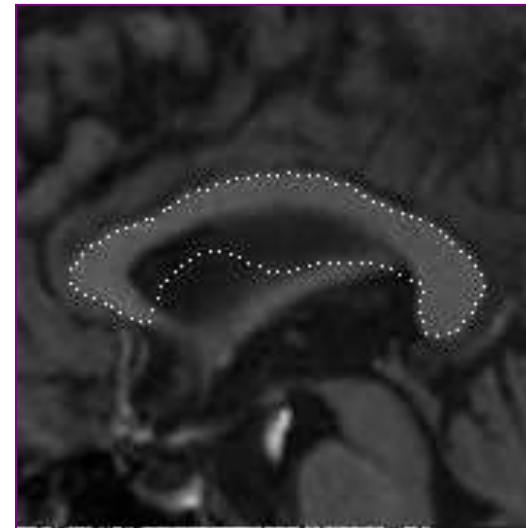
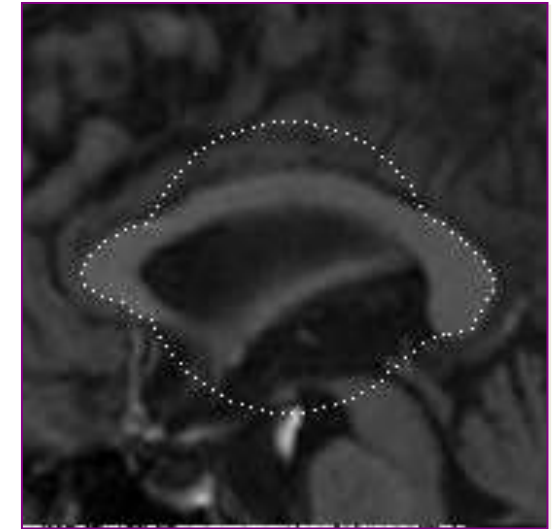
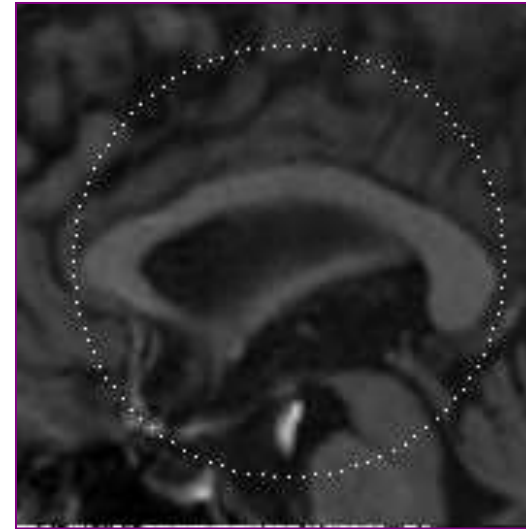
Grundlegende Segmentierungsverfahren sind:

- Segmentierung per **Schwellenwert**
 - Häufig per **Histogramm** bestimmt (*Beispiel*: Otsu)
 - Bei ungleichmäßigen Hintergründen **adaptiv** (lokale Nachbarschaft)
- **Region Growing** (flood fill)
- **Wasserscheidentransformation**
 - Anwendung meist auf **Gradientenbilder** oder **Distanztransformation**
- Vor- und Nachverarbeitung per Glättung, **morphologischen Operationen** und **Zusammenhangskomponenten**

4.3 Deformierbare Modelle

Grundidee: Segmentierung mit Aktiven Konturen

- **Aktive Konturen** werden im Bild initialisiert und verformen sich mit dem Ziel eine Energie zu minimieren, die meist aus zwei Teilen besteht:
 - **Externe Energie** verbindet die Kontur mit Bildinhalten, zieht sie z.B. in Richtung von Bildkanten
 - **Interne Energie** bewertet die Plausibilität der Kontur an sich, z.B. glatt und nicht zu lang
- *Beispiel:* Segmentierung des Balkens im Gehirn mittels einer aktiven Kontur



Bilder aus [Davatzikos et al. 1996]

Explizite Deformierbare Modelle

- **Deformierbare Modelle** in 2D lassen sich explizit als Kurven mit Parameter $s \in [0,1]$ schreiben:

$$\mathbf{v}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}$$

- **Aktive Konturen** optimieren $\mathbf{v}(s)$ im Hinblick auf eine Energie, die aus einer externen (Bildterm) und einer internen Energie (Glattheitsterm) besteht:

$$E = E_{\text{ext}} + E_{\text{int}}$$

- Betrachtet man den Prozess der iterativen Energieminimierung als Animation, kriechen die Kurven wie Schlangen über das Bild. Aktive Konturen werden daher auch als „**Snakes**“ bezeichnet.

Externe Energie

- **Externe Energie** $E_{\text{ext}} = \frac{1}{2} \int_0^1 P(\mathbf{v}(s)) ds$ zieht die Kontur in Richtung der gewünschten Bildstrukturen. *Beispiele* für Potentialfunktionen $P(\mathbf{x})$:

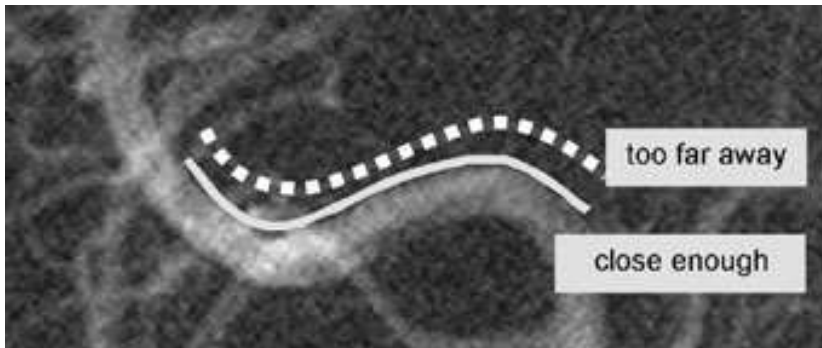
- Suche nach einer bestimmten **Bildintensität**:

$$P(\mathbf{v}(s)) = (I(\mathbf{v}(s)) - I_{\text{target}})^2$$

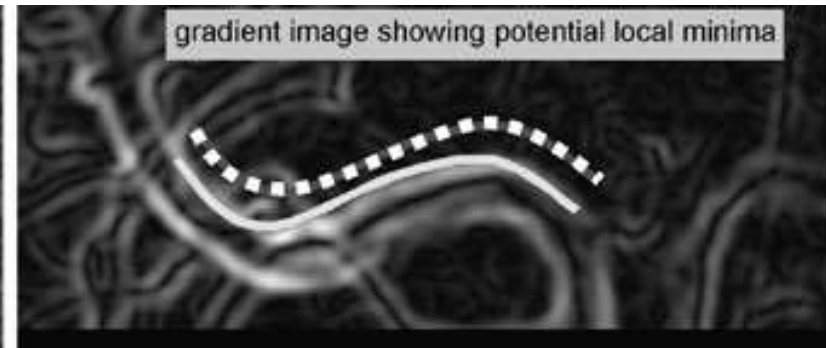
- Suche nach **Bildkanten**:

$$P(\mathbf{v}(s)) = -\|\nabla I(\mathbf{v}(s))\|^2$$

- Ableitungen erfordern in der Regel eine Glättung (s. Kapitel 1)



Bild



Bildgradient

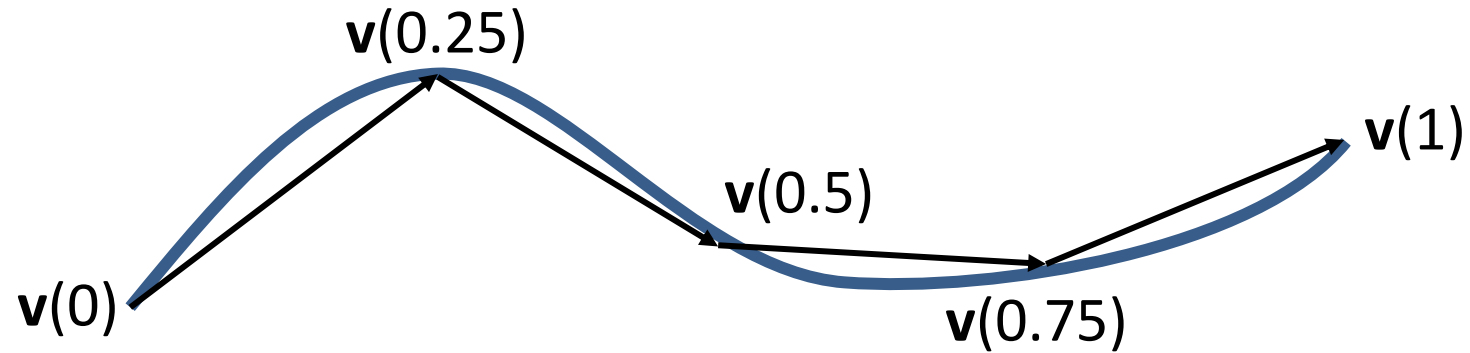
Interne Energie

- **Interne Energie** beruht bei [Kass et al. 1998] auf den ersten und zweiten Ableitungen von $\mathbf{v}(s)$:

$$E_{\text{int}} = \frac{1}{2} \int_0^1 \left[w_1(s) \left\| \frac{d\mathbf{v}(s)}{ds} \right\|^2 + w_2(s) \left\| \frac{d^2\mathbf{v}(s)}{ds^2} \right\|^2 \right] ds$$

- **Erster Term** macht die Kurve *kürzer*
 - Optimum wenn $\mathbf{v}(0) \neq \mathbf{v}(1)$ fest sind: Gerade Verbindung
 - Optimum mit Randbedingung $\mathbf{v}(0) = \mathbf{v}(1)$: Kurve schrumpft auf einen Punkt
- **Zweiter Term** macht die Kurve *glatter*
- Gewichte w_1 und w_2 bestimmen den jeweiligen Einfluss
 - $w_2(s) = 0$ ermöglicht an der Stelle s eine scharfe Ecke

Anschaung: Länge einer Parametrischen Kurve



- Approximation der Länge durch n Liniensegmente:

$$l \approx \sum_{i=0}^{n-1} \left\| \mathbf{v}\left(\frac{i+1}{n}\right) - \mathbf{v}\left(\frac{i}{n}\right) \right\| = \sum_{i=0}^{n-1} \left\| \frac{\mathbf{v}\left(\frac{i}{n} + \Delta s\right) - \mathbf{v}\left(\frac{i}{n}\right)}{\Delta s} \right\| \Delta s$$

- Im Grenzfall $n \rightarrow \infty$ erhalten wir die genaue Länge:

$$l = \int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\| ds$$

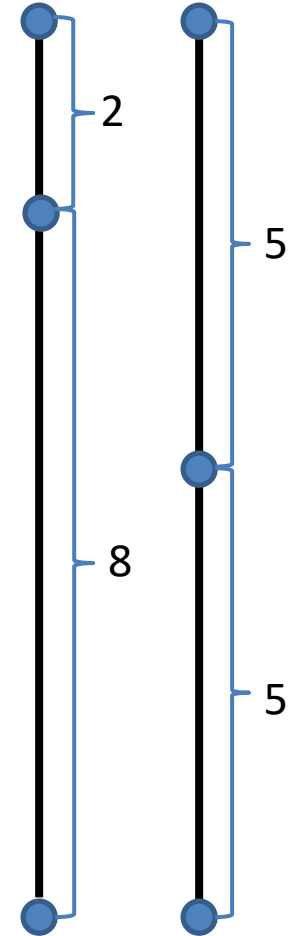
Anschaung: Was ist mit dem Quadrat?

- Statt $\int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\| ds$ minimiert die Snake $\int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\|^2 ds$
 - Das Quadrat vereinfacht spätere Rechnungen

- Aufgrund der Cauchy-Schwarz-Ungleichung gilt

$$\left(\int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\| ds \right)^2 \leq \int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\|^2 ds$$

- Die Snake minimiert eine obere Schranke der Länge
- Zusätzlich bevorzugt sie gleichmäßig verteilte Stützpunkte
- Bei perfekter Verteilung erhalten wir das Quadrat der Länge



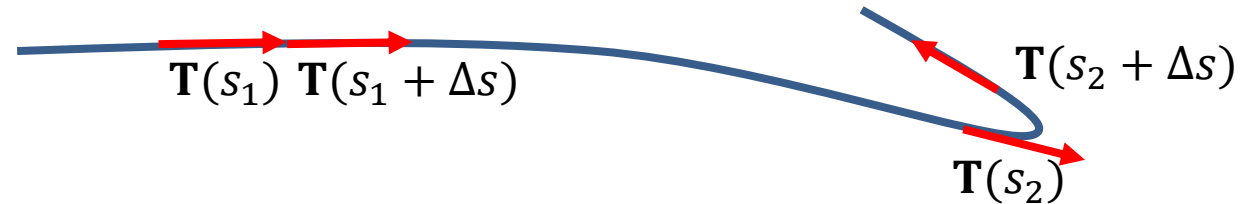
$$\begin{array}{ll} 2 + 8 = 10 & 5 + 5 = 10 \\ 2^2 + 8^2 = 68 & 5^2 + 5^2 = 50 \end{array}$$

31

Anschaung: Krümmung

- **Krümmung** κ gibt die lokale Abweichung der Kurve von einer Geraden an
 - Ihre Definition basiert auf der Ableitung des Einheitstangentenvektors

$$\mathbf{T}(s) = \frac{\frac{d\mathbf{v}(s)}{ds}}{\left\| \frac{d\mathbf{v}(s)}{ds} \right\|}$$



- Im Falle einer *Parametrisierung nach Bogenlänge* (d.h. $\left\| \frac{d\mathbf{v}(s)}{ds} \right\| = 1$) ist

$$\kappa = \left\| \frac{d^2\mathbf{v}(s)}{ds^2} \right\|$$

- Wir dürfen diese Formel verwenden, wenn wir die Stützpunkte ungefähr in festen Abständen verteilen (gleichmäßig, weder zu dicht noch zu weit)

Illustration: Interne Energie



Wenig dehnfähig (großes w_1)



Dehnfähiger, aber steif
(kleineres w_1 , großes w_2)



Dehnfähig und biegsam
(kleines w_1 , kleines w_2)

Variationsrechnung

- Differentialrechnung:

- Betrachtet **Funktionen** $f(x)$ die reelle Zahlen auf reelle Zahlen abbilden
- **Minima** sind Punkte x , für die für alle hinreichend kleinen Δx gilt:

$$f(x + \Delta x) > f(x)$$

- **Notwendige Bedingung:**

$$\frac{df(x)}{dx} = 0$$

- Variationsrechnung:

- Betrachtet **Funktionale** $F(f)$, die Funktionen auf reelle Zahlen abbilden
- **Minima** sind Funktionen $f(x)$, für die für alle hinreichend kleinen ε und jede beliebig oft differenzierbare Testfunktion $\eta(x)$ gilt:

$$F(f + \varepsilon \cdot \eta) > F(f)$$

- Notwendige Bedingung: **Euler-Lagrange-Gleichung**

Euler-Lagrange-Gleichung

- **Satz** aus der Variationsrechnung:

Eine notwendige Bedingung für die Minimierung eines Funktionals, das sich mittels einer Lagrange-Funktion L in der Form

$$F(f) = \int_a^b L(x, f(x), f'(x), f''(x)) dx$$

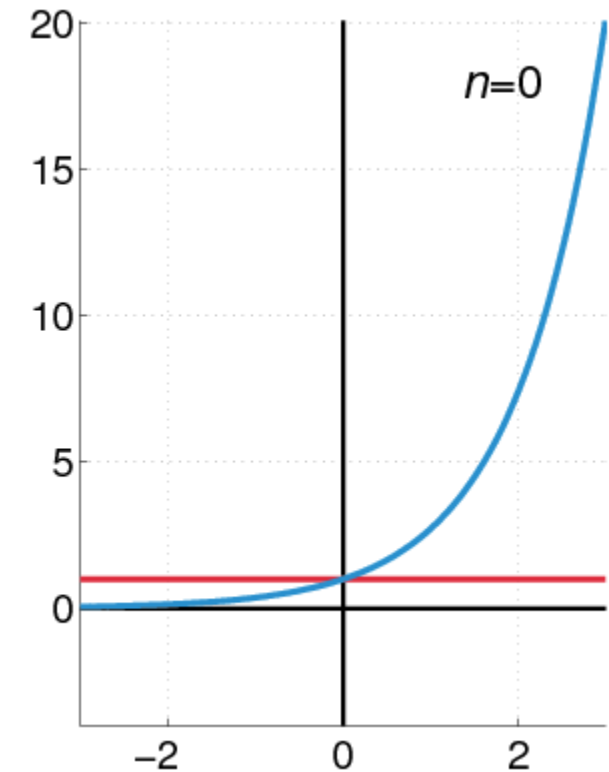
darstellen lässt, ist die Erfüllung der Euler-Lagrange-Gleichung

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x} \left(\frac{\partial L}{\partial f'} \right) + \frac{\partial^2}{\partial x^2} \left(\frac{\partial L}{\partial f''} \right) = 0$$

Kurze Erinnerung: Taylor-Entwicklung

- Die **Taylor-Entwicklung** ermöglicht die Approximation glatter Funktionen in der Umgebung einer Stelle x_0 durch Polynome

$$\begin{aligned} f(x) &\approx f(x_0) \\ &+ \frac{d}{dx} f(x_0)(x - x_0) \\ &+ \frac{1}{2} \frac{d^2}{dx^2} f(x_0)(x - x_0)^2 + \dots \\ &+ \frac{1}{n!} \frac{d^n}{dx^n} f(x_0)(x - x_0)^n \\ &+ O((x - x_0)^{n+1}) \end{aligned}$$



Bildquelle: Wikipedia

Herleitung der Euler-Lagrange-Gleichung, Teil 1

- Betrachte $F(f) = \int_a^b L(x, f(x), f'(x)) dx$
 - Zur Vereinfachung der Notation schreiben wir ab jetzt $L(x, f, f')$
- **Notwendige Bedingung** für Extrema: Für jede Testfunktion $\eta(x)$ ist

$$\lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} \int_a^b L(x, f + \epsilon\eta, f' + \epsilon\eta') - L(x, f, f') dx = 0$$

- Einsetzen der Taylor-Entwicklung von L :

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} \int_a^b L(x, f, f') + \frac{\partial L}{\partial f} \epsilon\eta + \frac{\partial L}{\partial f'} \epsilon\eta' + O(\epsilon^2) - L(x, f, f') dx &= 0 \\ \Rightarrow \int_a^b \frac{\partial L}{\partial f} \eta + \frac{\partial L}{\partial f'} \eta' dx &= 0 \end{aligned}$$

Herleitung der Euler-Lagrange-Gleichung, Teil 2

- Notwendige Bedingung (aus Teil 1): $\int_a^b \frac{\partial L}{\partial f} \eta + \frac{\partial L}{\partial f'} \eta' dx = 0$

- Partielle Integration von $\int_a^b \frac{\partial L}{\partial f'} \eta' dx$ mit $u = \frac{\partial L}{\partial f'}$ $dv = \eta'$:

$$\left[\frac{\partial L}{\partial f'} \eta \right]_a^b - \int_a^b \frac{\partial}{\partial x} \left(\frac{\partial L}{\partial f'} \right) \eta dx$$

- Fordern wir zur Einhaltung der Randbedingungen $\eta(a) = \eta(b) = 0$, wird die notwendige Bedingung zu

$$\int_a^b \frac{\partial L}{\partial f} \eta - \frac{\partial}{\partial x} \left(\frac{\partial L}{\partial f'} \right) \eta dx = \int_a^b \underbrace{\left(\frac{\partial L}{\partial f} - \frac{\partial}{\partial x} \left(\frac{\partial L}{\partial f'} \right) \right)} \eta dx = 0$$

Um für beliebige η Null zu erhalten muss dieser Teil Null sein -> Euler-Lagrange

Anwendung auf das Snake-Modell

Dritter Term der ELG analog (wiederholte partielle Integration):

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x} \left(\frac{\partial L}{\partial f'} \right) + \frac{\partial^2}{\partial x^2} \left(\frac{\partial L}{\partial f''} \right) = 0$$

Anwendung der ELG auf das Snake-Modell

$$E = \frac{1}{2} \int_0^1 P(\mathbf{v}(s)) ds + \frac{1}{2} \int_0^1 \left[w_1(s) \left\| \frac{d\mathbf{v}(s)}{ds} \right\|^2 + w_2(s) \left\| \frac{d^2\mathbf{v}(s)}{ds^2} \right\|^2 \right] ds$$

mit externem Potential P ergibt

$$\frac{1}{2} \nabla P(\mathbf{v}(s)) - \frac{d}{ds} \left(w_1(s) \frac{d\mathbf{v}(s)}{ds} \right) + \frac{d^2}{ds^2} \left(w_2(s) \frac{d^2\mathbf{v}(s)}{ds^2} \right) = \mathbf{0}$$

Da $\mathbf{v}(s)$ vektorwertig ist, ergibt sich ein **System zweier Gleichungen** (in 2D-Bildern) für die x - bzw. y -Koordinaten unserer Kurve

Energieminimierung im Snake-Modell

- Die linke Seite der Euler-Lagrange-Gleichung des Snake-Modells

$$\frac{1}{2} \nabla P(\mathbf{v}(s)) - \frac{d}{ds} \left(w_1(s) \frac{d\mathbf{v}(s)}{ds} \right) + \frac{d^2}{ds^2} \left(w_2(s) \frac{d^2\mathbf{v}(s)}{ds^2} \right) = \mathbf{0}$$

gibt eine lokale Änderung an $\mathbf{v}(s)$ an, die E so steil wie möglich ansteigen lässt.

- Wir nutzen sie zum **Gradientenabstieg**:
 - Einführung eines (künstlichen) Zeitparameters $\mathbf{v}(s, t)$
 - Ersetzen der rechten Seite durch $-\partial_t \mathbf{v}(s, t)$
 - Diskretisierung der Zeit in uniforme Schritte
 - Diskretisierung der Kurve $\mathbf{v}(s, t)$ als Polygonzug
 - Iterative Anwendung der Updates auf $\mathbf{v}(s, t)$ bis $\partial_t \mathbf{v}(s, t) \approx 0$

Interpretation als Summe von Kräften

Die Ableitung nach der Zeit in

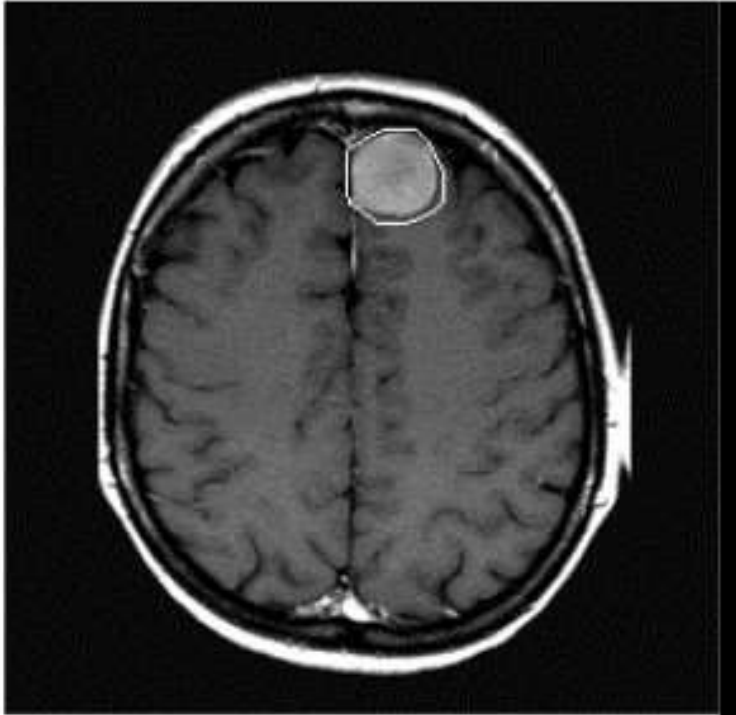
$$\frac{1}{2} \nabla P(\mathbf{v}(s, t)) - \frac{\partial}{\partial s} \left(w_1(s) \frac{\partial \mathbf{v}(s, t)}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left(w_2(s) \frac{\partial^2 \mathbf{v}(s, t)}{\partial s^2} \right) = - \frac{\partial \mathbf{v}(s, t)}{\partial t}$$

durch finite Differenzen zu ersetzen ergibt

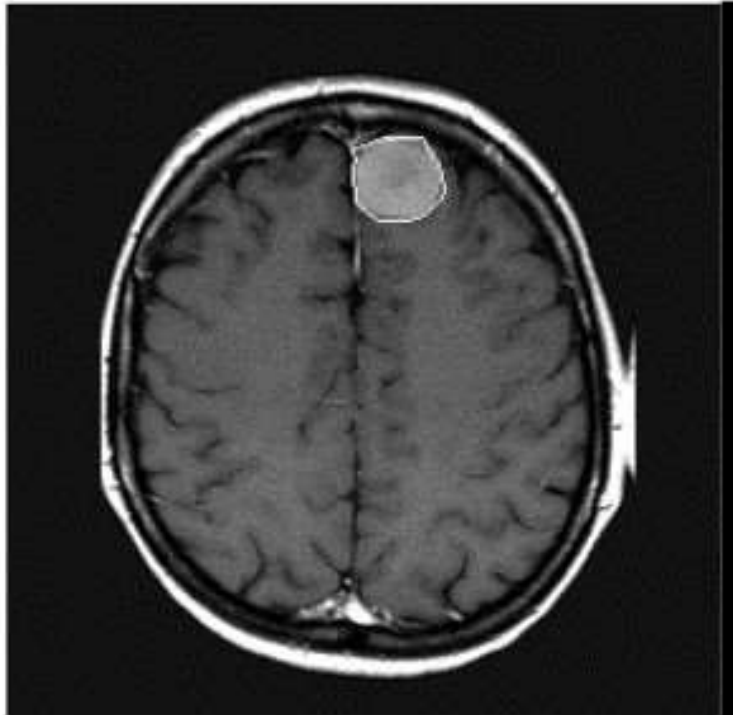
$$\begin{aligned} \text{Linke Seite} &= - \frac{\mathbf{v}(s, t + \Delta t) - \mathbf{v}(s, t)}{\Delta t} \\ \Rightarrow \mathbf{v}(s, t + \Delta t) &= \mathbf{v}(s, t) + \Delta t \times (-\text{L. S.}) \end{aligned}$$

- Aus jedem Energieterm wird eine Kraft, iterative Updates verschieben $\mathbf{v}(s, t)$ aufgrund der Summe dieser Kräfte mit Schrittweite Δt
- Mit geeigneter Schrittgröße konvergieren die Updates zu einer Lösung der Euler-Lagrange-Gleichung, am Optimum gleichen die Kräfte einander aus

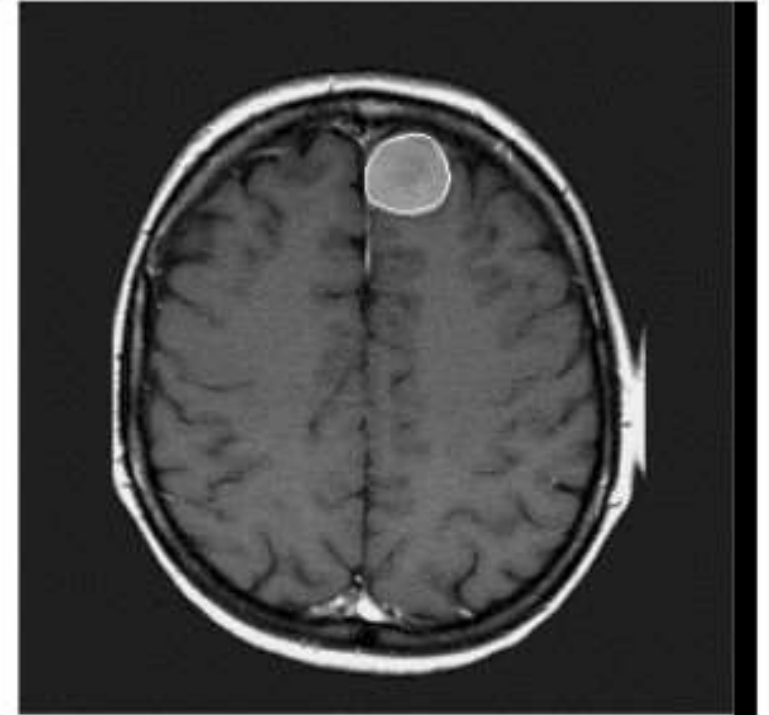
Beispiel: Tumorsegmentierung per Snake



Initialkontur



2. Iterationsschritt

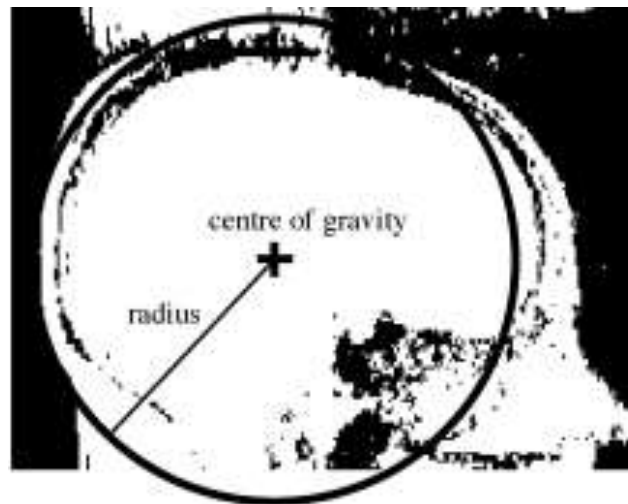
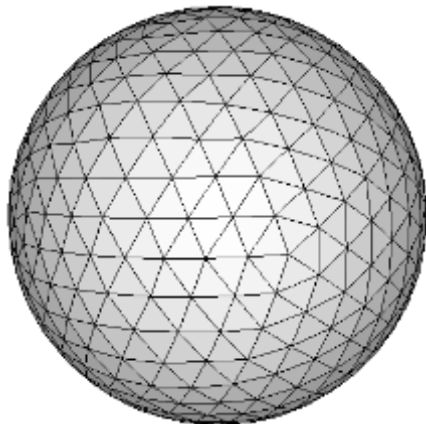


Konvergenz

Fallbeispiel: Brain Extraction Tool (BET)



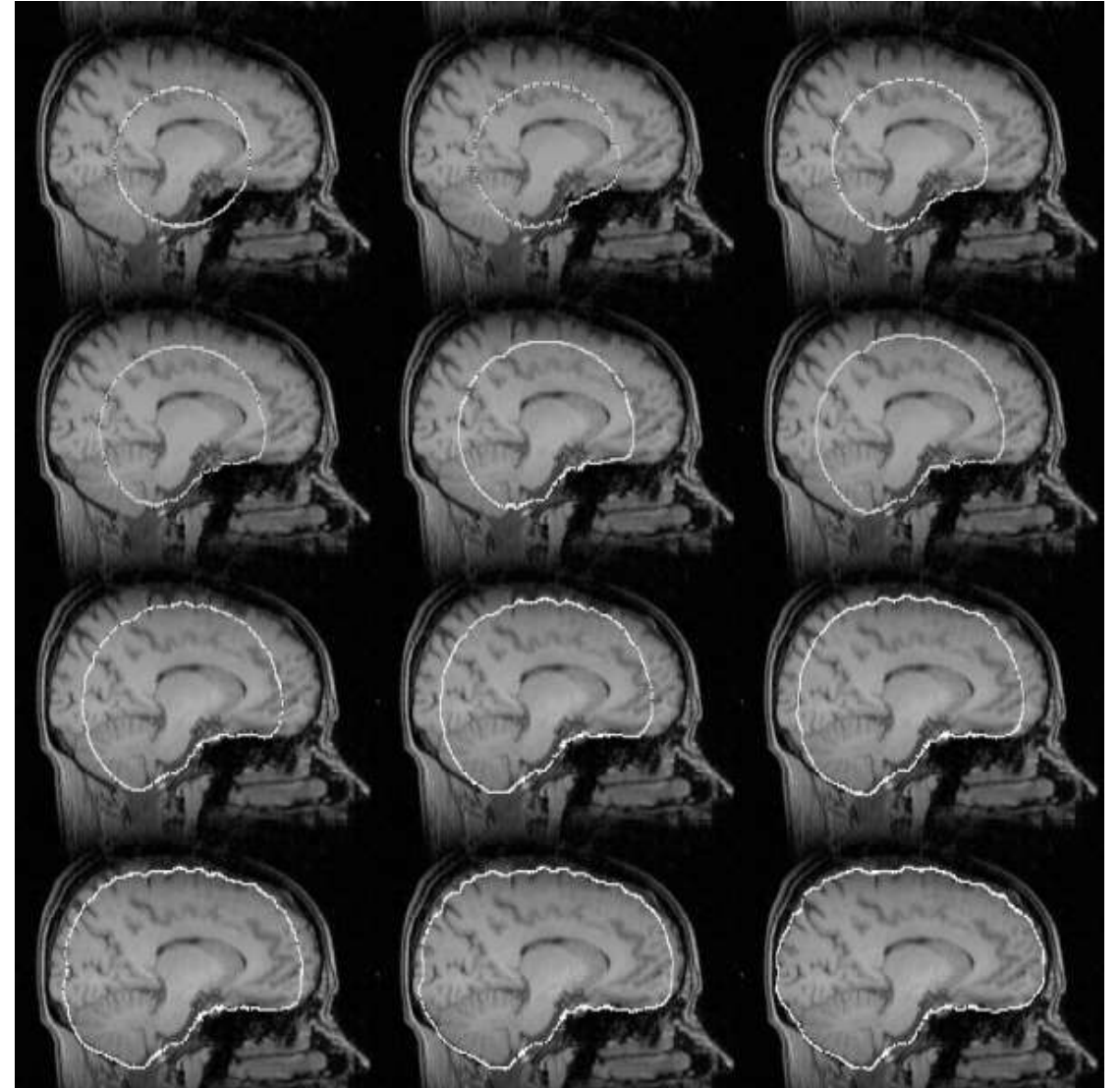
- **Brain Extraction Tool (BET)** aus FSL, <http://fsl.fmrib.ox.ac.uk/>
 - Bis heute sehr aktiv genutzt (>12000 Zitationen auf google scholar)
 - Fast vollständig automatisiert
 - Zwei Parameter für 1) Über-/Untersegmentierung 2) Intensitätsgradienten
 - Basiert auf einem deformierbaren Modell
 1. Explizite Repräsentation als Dreiecksnetz
 2. Heuristische Initialisierung: Kleine Kugel innerhalb des Gehirns



BET: Grundidee und Beispiel

Entwicklung der Fläche durch die Summe von drei ad hoc definierten Kräften:

1. Drückt die Fläche nach außen, bis die Bildintensität unter einen adaptiven Schwellenwert sinkt
2. Wirkt starken Krümmungen entgegen
3. Verteilt die Stützpunkte möglichst gleichmäßig über die Fläche



Vor- und Nachteile Aktiver Konturen

- **Vorteile:**

- *Transparent:* Energiefunktional drückt in klarer mathematischer Sprache die Ziele der Segmentierung aus
- *Relativ schnell:* Updates der Stützpunkte meist leicht auszurechnen
- Enorme Flexibilität in der Formulierung von Potentialen
 - z.B. Integration von manuellen Vorgaben, Ballon-Kräften, Vorwissen

- **Nachteile:**

- Berechnung der Euler-Lagrange-Gleichung zuweilen kompliziert
 - *Alternative:* Statt sie aus einem Energiefunktional herzuleiten werden die Kräfte manchmal einfach „ad hoc“ angegeben
- Beibehalten gleichverteilter Stützpunkte und Änderungen in der Topologie erfordern komplexen Programmcode

4.4 Level-Set-Methode

Nachteile Expliziter Repräsentationen

- Explizite Repräsentation von Kurven durch Polygonzüge hat **Vorteile**:
 - Verschieben der Stützpunkte in jedem Schritt einfach und effizient
- **Nachteile**: Benötigt nichttrivialen Code um
 - eine angemessene Zahl von Stützpunkten sicherzustellen
 - topologische Änderungen durchzuführen
 - Verbinden oder Auftrennen von Kurven
 - Erzeugen oder Verschwinden von Komponenten

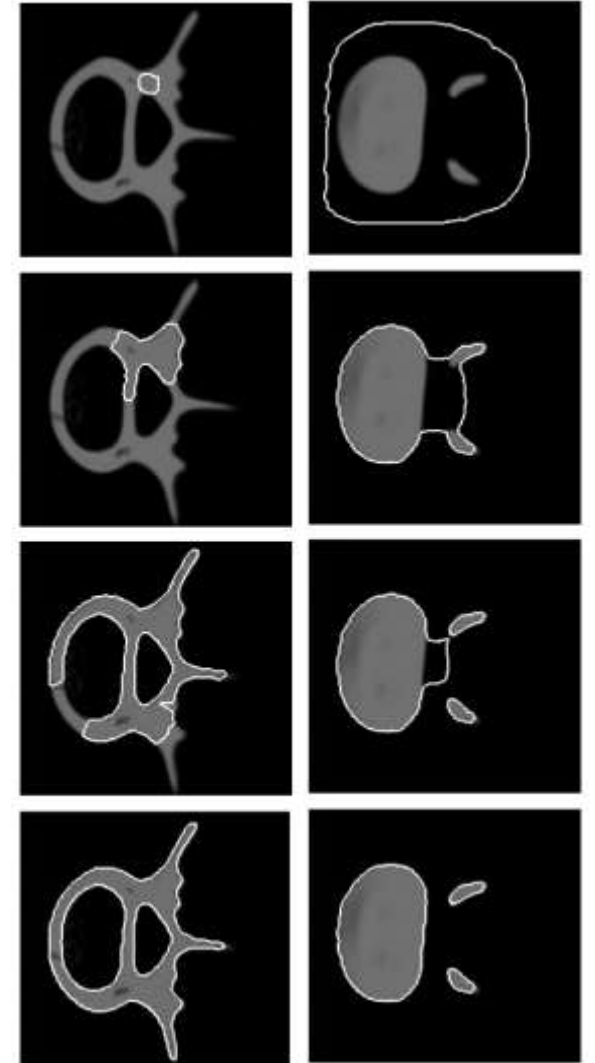


Bild: [McInerney/Terzopoulos 1999]

Alternative: Darstellung durch Level Sets

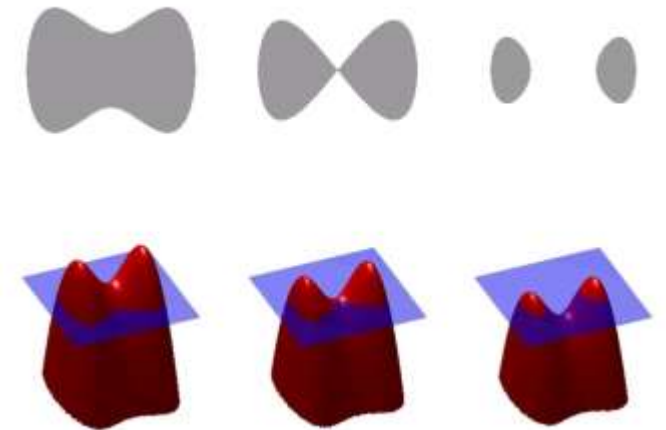
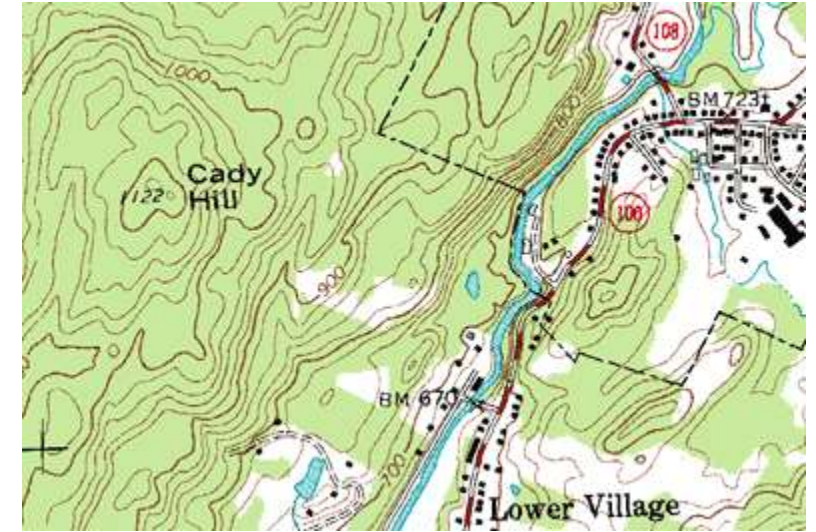
- **Implizit** können Kurven als Niveaumenge (level set) einer Höhenfunktion $\phi(x, y)$ dargestellt werden

- **Definition** einer Niveaumenge:

$$\{(x, y) \mid \phi(x, y) = c\}$$

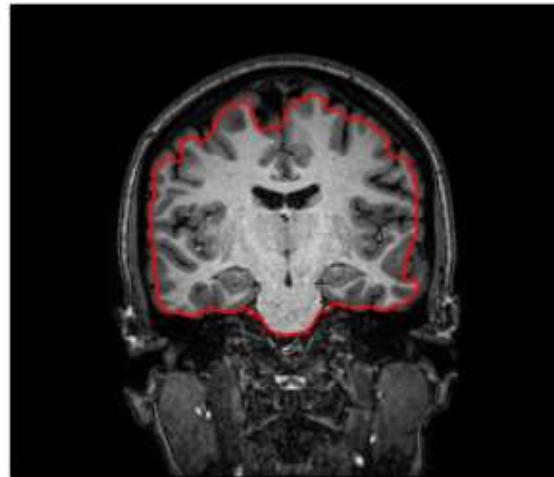
- **Vorteile:**

- Topologische Änderungen der Kurve entsprechen kontinuierlichen Änderungen von ϕ , stellen keine besonderen Schwierigkeit mehr dar
- Darstellung komplexer Formen erfordert keine Erhöhung der Zahl von Stützpunkten

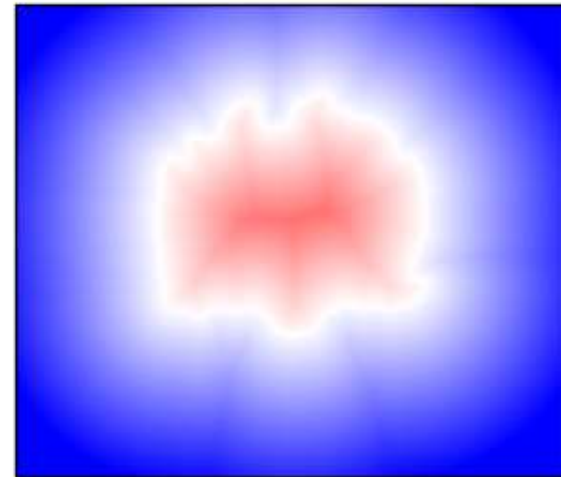


Vorzeichenbehaftete Distanztransformation

- Aus der initialen Kurve gewinnen wir ein entsprechendes Höhenfeld mit der **vorzeichenbehafteten Distanztransformation** (*engl.* signed distance transform, SDT)
 - Betrag gibt Abstand von der Kurve an
 - Vorzeichen gibt Vorder- oder Hintergrund an
 - *Konvention*: Vordergrund positiv, Hintergrund negativ



Kurve

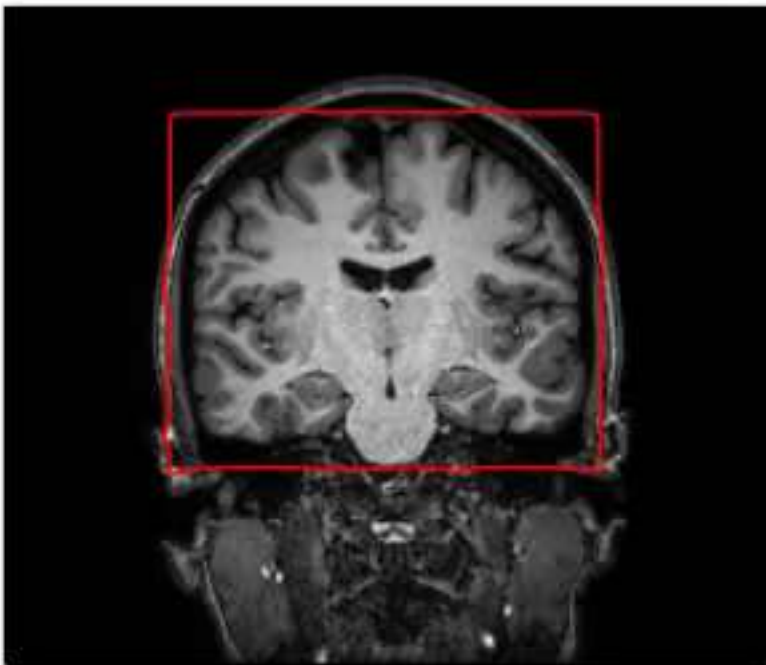


Höhenfunktion ϕ

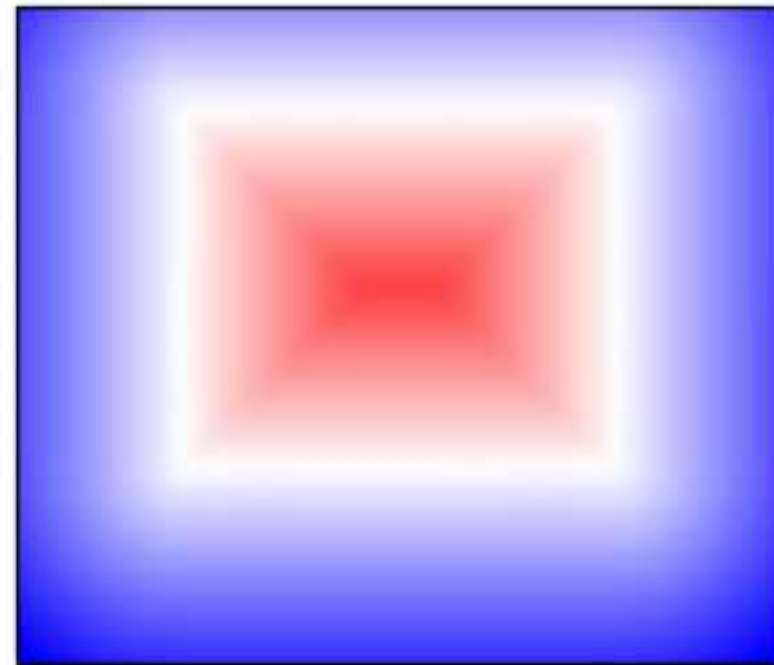


Entwicklung von Level Sets

- Durch Level Sets dargestellte Kurven werden durch iterative **Updates ihrer Höhenfunktion** deformiert
 - Genaue Formel ergibt sich häufig aus einem Energiefunktional



Entwicklung der Kontur

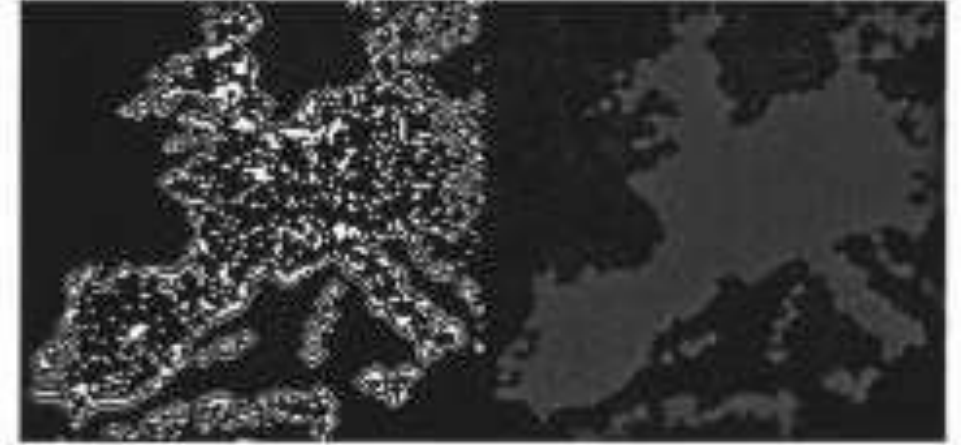


Entsprechendes Höhenfeld

Beispiel: Energie-Funktional zur Segmentierung

- Erneut formalisiert ein **Energie-Funktional** unser Segmentierungsziel
 - *Beispiel:* Segmentierung in Regionen ungefähr konstanter Intensität, die durch möglichst kurze Kurven getrennt werden
 - Vereinfachte Version des Modells von [Mumford/Shah 1989]
 - *Formalisierung:* Bestimme Partition Ω_1, Ω_2 des Definitionsbereichs eines Bildes Ω ($\Omega_2 = \Omega \setminus \Omega_1$) und Intensitäten μ_1, μ_2 , die die folgende Energie E_{MS} minimieren:

$$E_{MS}(\mu_i, \Omega_i) = \int_{\Omega_1} (I(x) - \mu_1)^2 dx + \int_{\Omega_2} (I(x) - \mu_2)^2 dx + \nu |\partial\Omega_1|$$



Update-Gleichung: Skizze der Herleitung

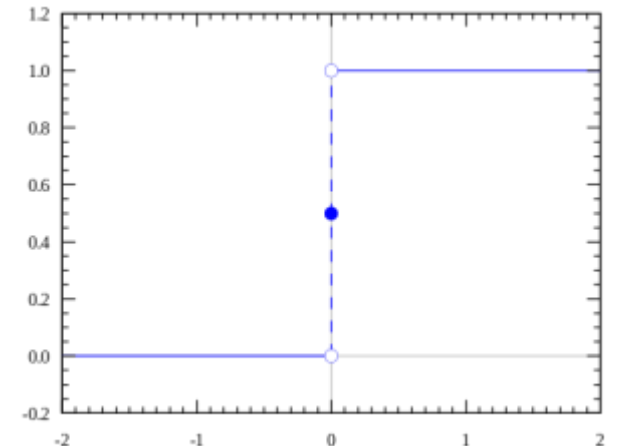
Zunächst schreiben wir

$$E_{\text{MS}}(\mu_i, \Omega_i) = \int_{\Omega_1} (I(x) - \mu_1)^2 dx + \int_{\Omega_2} (I(x) - \mu_2)^2 dx + \nu |\partial \Omega_1|$$

mittels der Heaviside (Sprung-)Funktion H als Funktional der Höhenfunktion $\phi(x, y)$

$$E_{\text{MS}}(\mu_i, \phi) = \int_{\Omega} H(\phi(x)) (I(x) - \mu_1)^2 + (1 - H(\phi(x))) (I(x) - \mu_2)^2 + \nu \|\nabla H(\phi(x))\| dx$$

- Nach jedem Update von ϕ werden μ_1 und μ_2 auf Mittelwerte von I inner- bzw. außerhalb gesetzt
- Gradientenabstieg per Euler-Lagrange-Gleichung erfordert eine Regularisierung von H .

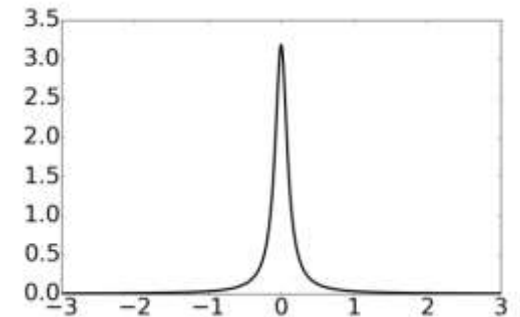


Update-Regel nach Chan und Vese

- Nach Regularisierung von H lässt sich folgende Update-Regel herleiten [Chan/Vese 2001]:

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) \left[(I - \mu_2)^2 - (I - \mu_1)^2 + \nu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right]$$

wobei $\delta_{\epsilon}(\phi) = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + \phi^2}$ mit $\epsilon > 0$

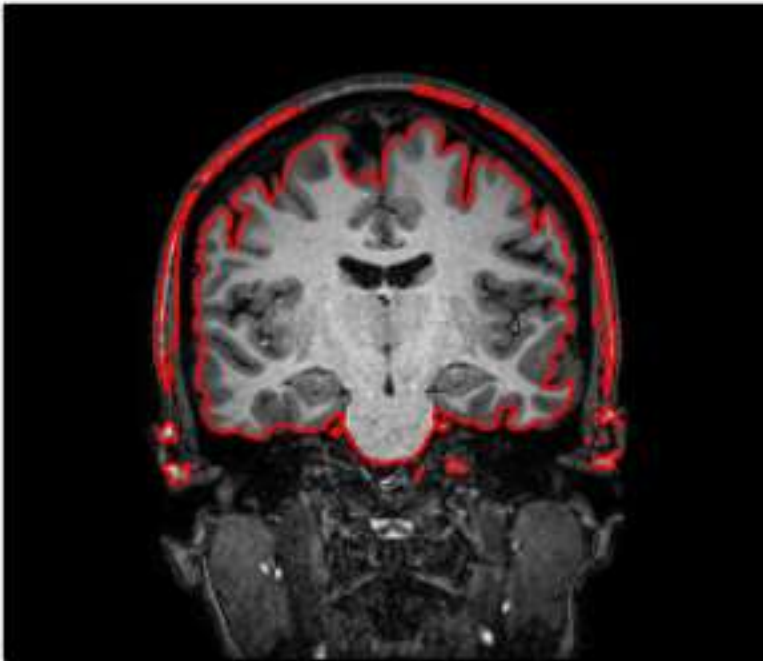


- *Hinweis*: Die genaue Herleitung des Divergenz-Terms zeigen wir nicht.
- Außerdem zu beachten:
 - Kontinuierliches „Nachführen“ von μ_1 und μ_2
 - ϕ sollte eine gültige Distanzfunktion bleiben
 - Weiterer Term in der Update-Funktion, den wir nicht weiter besprechen

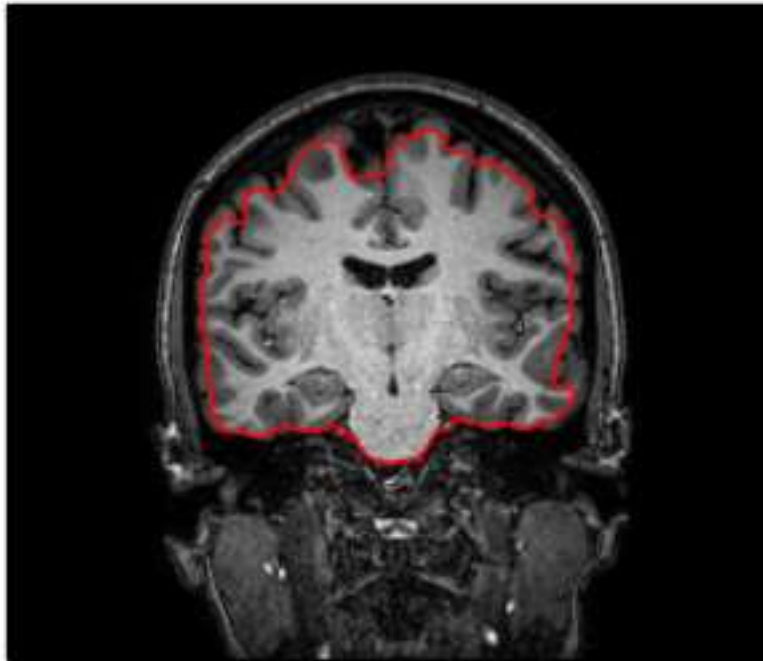
Beispiel: Effekt des Regularisierungs-Parameters

Wahl von ν :

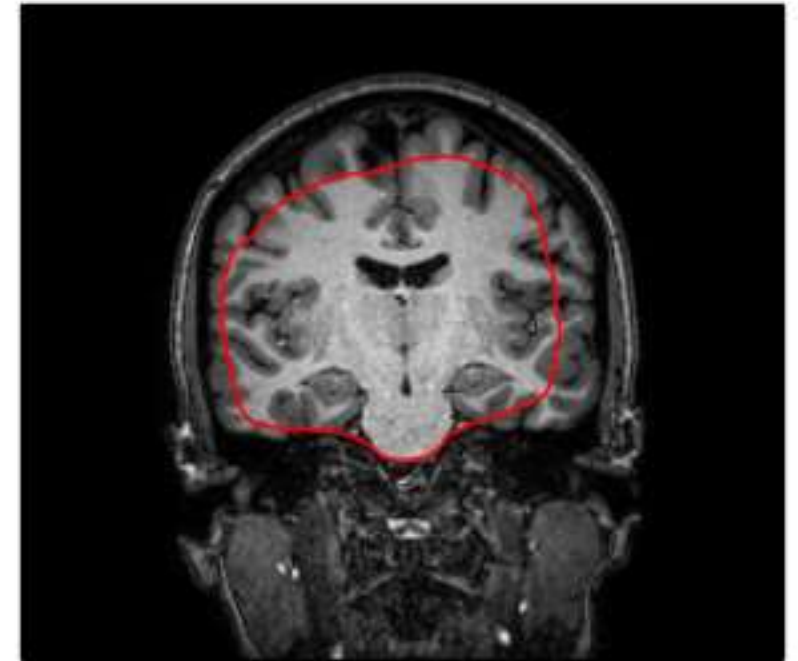
Wie stark bestrafen wir die Länge der Kontur?



$\nu = 0.1$



$\nu = 1$



$\nu = 10$

Ausblick: Varianten der Level Sets

- Formulierung verschiedener Energiefunktionale ermöglicht hohe **Flexibilität** im Hinblick auf Segmentierungsziele, z.B.
 - Kantenbasierte Modelle
 - Komplexere regionenbasierte Modelle
 - Berücksichtigung von Vorwissen und interaktivem Feedback
- Verarbeitung von **Farb-** und **3D-Bildern**
- Einteilung in **mehrere Klassen** (gekoppelte Level Sets)

Zusammenfassung: Level Sets

- **Vorteile der Level Sets**
 - **Einfach:** Topologische Änderungen verursachen keinen besonderen Implementierungsaufwand
 - **Transparent:** Im Energie-Funktional sind die Modell-Annahmen explizit ablesbar
 - **Flexibel:** Sowohl im Hinblick auf komplexe Konturen, als auch auf Umsetzbarkeit verschiedener Zielkriterien
- **Nachteile der Level Sets**
 - **Rechenaufwand**, insbesondere für große und 3D-Bilder
 - *Ansatz:* Update der Höhenfunktion nur in schmalem Band um die Kontur
 - Abhängigkeit von der **Initialisierung**

4.5 Aktive Formmodelle

Ein berühmtes Beispiel zur Motivation

- Was zeigt dieses Bild?
- Wie könnten wir es sinnvoll segmentieren?



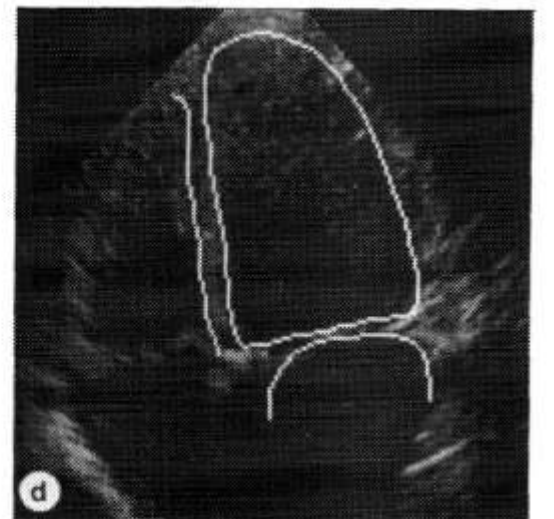
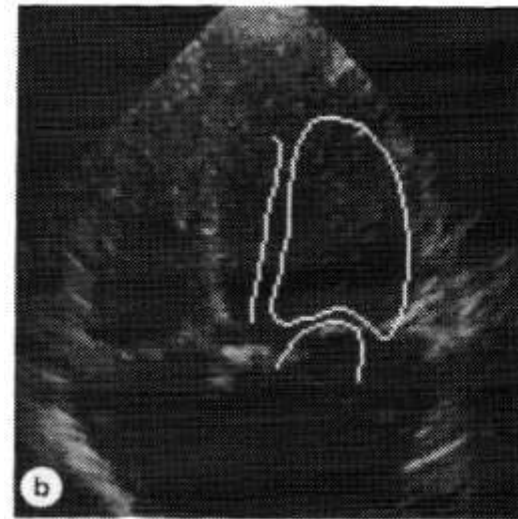
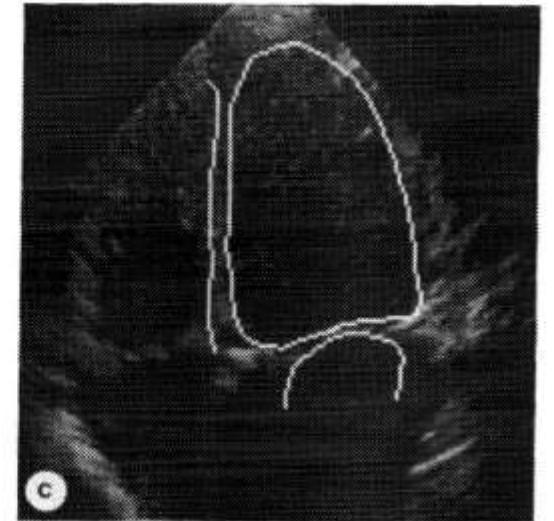
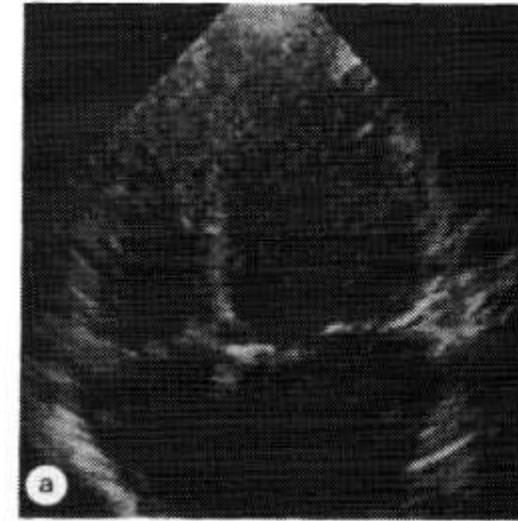
Grundidee: Modellbasierte Segmentierung

- **Modellbasierte Segmentierung** nutzt spezifisches Vorwissen über das gesuchte Objekt aus
 - **Statistische Modelle** werden aus Beispielen gelernt
- **Formmodelle** sind hilfreich, wenn das gesuchte Objekt eine charakteristische Form hat (z.B. Knochen, Organe)
 - Modellieren typische Form und mögliche Abweichungen davon
 - Getrennt betrachtet werden Größe, Position und Orientierung, die zusammen als **Pose** der Form bezeichnet werden
- Bei der **Bildsegmentierung** mittels Formmodellen werden Pose und Formparameter bestimmt, die am besten zu dem Bildinhalt passen

Beispiel: Segmentierung mit Aktiven Formmodellen

Beispiel aus Cootes et al., *Active Shape Models – Their Training and Application* (1995):

- Segmentierung einer Herzkammer in einem Echokardiogramm



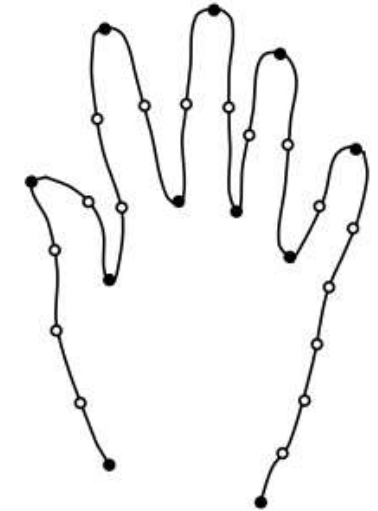
Initialisierung

Nach 200 Iterationen 60

Nach 80 Iterationen

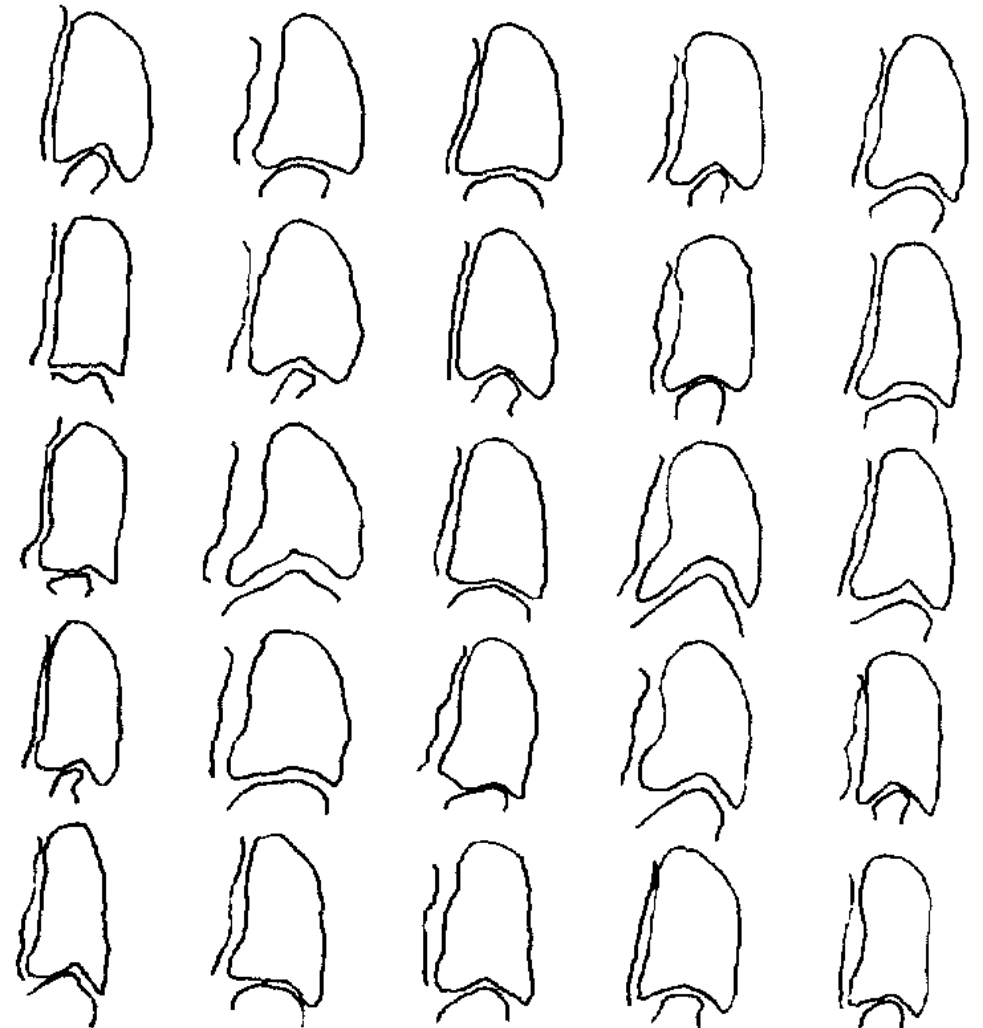
Punktverteilungsmodell

- Ein **Punktverteilungsmodell** stellt eine Form durch die Koordinaten von n Stützpunkten in einer Referenzpose sowie durch ihre Verbindungen dar
 - Die **Stützpunkte** umfassen
 - **Landmarken**, eindeutig erkennbare Punkte, die in verschiedenen Bildern in Korrespondenz gebracht werden
 - Genügend **Hilfspunkte**, um die genaue Form zwischen den Landmarken hinreichend genau linear zu approximieren
 - 2D-Formen werden durch $2n$ -dimensionale Vektoren \mathbf{x}_i beschrieben
 - Wir lernen das Formmodell aus m Beispielen, $i = 1, 2, \dots, m$
 - Mögliche Variationen der Form werden durch eine **gemeinsame Wahrscheinlichkeitsverteilung** der Punktkoordinaten erfasst



Erlernen des Formmodells aus Beispielen

- Um die typische Form und ihre möglichen Variationen zu erlernen, werden zunächst **Trainingsbilder** von Hand annotiert
 - Punkte in jedem Beispiel müssen **korrespondieren**, daher ist die Auswahl geeigneter Landmarken wichtig
 - Anhaltspunkt zur **benötigten Zahl** von Trainingsbildern: Kann jede Form sinnvoll durch ein Modell erzeugt werden, das aus den übrigen $m - 1$ trainiert wurde?



Ausrichtung zweier Formen

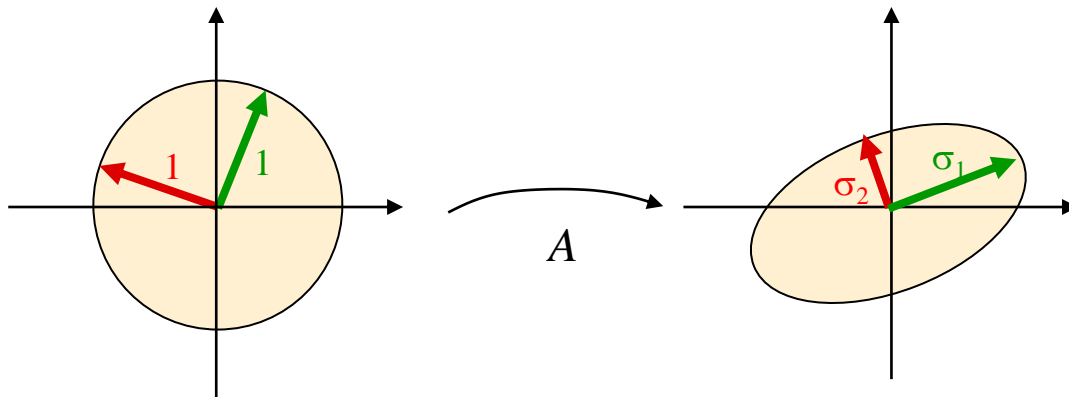
- Um die Abweichung zwischen zwei Formen \mathbf{x}_1 und \mathbf{x}_2 zu bestimmen, benötigen wir zunächst ihre **relative Pose**, d.h. Skalierung s , Verschiebung \mathbf{t} und Rotation θ , die $\|T_{s,\mathbf{t},\theta}(\mathbf{x}_1) - \mathbf{x}_2\|$ minimieren
- Die entsprechende Ausrichtung von Formen bezeichnet man als **Prokrustes-Analyse**
- Beliebte Implementierung im Kontext von Formmodellen:
 1. **Verschiebung** um die Differenz der Mittelpunkte $\mathbf{t} = \mathbf{m}_2 - \mathbf{m}_1$
 2. Skalierung auf dieselbe **Norm**, $s = \frac{\|\mathbf{z}_2\|}{\|\mathbf{z}_1\|}$
 - \mathbf{z}_1 und \mathbf{z}_2 sind \mathbf{x}_1 und \mathbf{x}_2 nach Zentrierung auf den Ursprung
 3. Rotation mittels der **Singulärwertzerlegung** (s. nächste Folie)

Singulärwertzerlegung

- **Satz:** Für jede Matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ existiert eine **Singulärwertzerlegung** (engl. Singular value decomposition, SVD)

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

- Spaltenvektoren der orthogonalen Matrizen $\mathbf{U} \in \mathbb{R}^{m \times m}$ und $\mathbf{V} \in \mathbb{R}^{n \times n}$ heißen Links- bzw. Rechts-Singulärvektoren
- positive Diagonalelemente $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ ($p = \min\{m, n\}$) der diagonalen Matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ heißen Singulärwerte



$$\mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma} \Rightarrow \mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

$$\mathbf{A} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

Rotation per Singulärwertzerlegung



Algorithmus zur Berechnung der Rotation der auf den Ursprung zentrierten Formen \mathbf{z}_1 und \mathbf{z}_2 :

1. Stelle die 2D-Formen durch $n \times 2$ Matrizen \mathbf{Z}_1 und \mathbf{Z}_2 dar
2. Berechne die 2×2 Kreuzkovarianzmatrix $\mathbf{W} = \mathbf{Z}_1^T \mathbf{Z}_2$
3. Berechne die Singulärwertzerlegung $\mathbf{W} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$
4. Die Rotation, die \mathbf{z}_1 optimal auf \mathbf{z}_2 abbildet, ergibt sich als

$$\mathbf{R} = \mathbf{V} \mathbf{U}^T$$

- Streng genommen müssen wir überprüfen, dass $\det(\mathbf{R}) = 1$.
Dies ist in der Praxis jedoch meistens der Fall.

Berechnung der Referenzform: Problemstellung

- Mittels Prokrustes-Analyse können wir nun aus allen m Formen eine mittlere **Referenzform** berechnen
- Die **Pose** der Referenzform $\bar{\mathbf{x}}$ geben wir dabei vor:
 - **Skalierung** des Formvektors $\bar{\mathbf{x}}$ auf Norm $\|\bar{\mathbf{x}}\| = 1$
 - **Position**: Zentrierung auf den Ursprung
 - **Orientierung**: Durch ein initiales Beispiel festgelegt
- Ziel ist die Berechnung einer mittleren Form $\bar{\mathbf{x}}$ und von Posen der einzelnen Formen, um die quadratische Abweichung $D = \sum_i \|T_{s_i, \mathbf{t}_i, \theta_i}(\mathbf{x}_i) - \bar{\mathbf{x}}\|^2$ zu minimieren

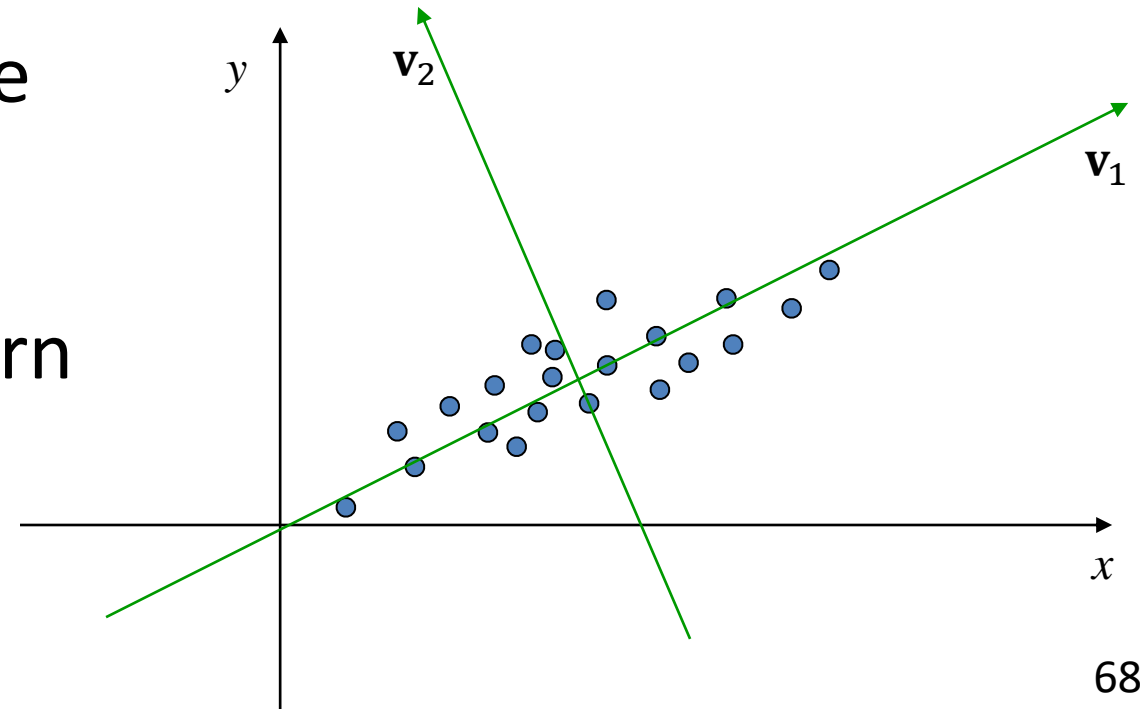
Berechnung der Referenzform: Vorgehen

Algorithmus zur iterativen Berechnung der Referenzform:

1. Zentriere jede Form um den Ursprung und normiere sie auf $\|\mathbf{x}_i\| = 1$
2. Wähle ein so normiertes Beispiel als initiale Referenz $\bar{\mathbf{x}}_0$ aus
3. Wiederhole für $j = 0, 1, 2, \dots$, bis $\|\bar{\mathbf{x}}_{j+1} - \bar{\mathbf{x}}_j\| < \epsilon$:
 - i. Richte alle \mathbf{x}_i an $\bar{\mathbf{x}}_j$ aus, nenne die Resultate \mathbf{x}'_i
 - ii. Berechne $\bar{\mathbf{x}}_{j+1} := \frac{1}{m} \sum_{i=1}^m \mathbf{x}'_i$
 - iii. Richte $\bar{\mathbf{x}}_{j+1}$ an $\bar{\mathbf{x}}_0$ aus und normiere auf $\|\bar{\mathbf{x}}_{j+1}\| = 1$

Hauptkomponenten-Analyse: Anschauung

- Die **Hauptkomponentenanalyse** (*engl.* Principal Component Analysis, PCA) bestimmt aus m gegebenen Punkten $\mathbf{x}_i \in \mathbb{R}^p$ so p orthogonale Richtungen \mathbf{v}_i , dass die Projektionen von \mathbf{x}_i auf die Unterräume $\langle \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_t \rangle$ jeweils so viel Varianz wie möglich erhalten.
- Formmodelle nutzen sie dazu, die an $\bar{\mathbf{x}}$ ausgerichteten Trainingsformen $\mathbf{x}_i \in \mathbb{R}^{2n}$ auf eine handhabbare Zahl von Formparametern zu reduzieren
 - Schränkt das Modell auf plausible Deformationen ein



Hauptkomponenten-Analyse: Algorithmus

Algorithmus zur Berechnung der Hauptkomponenten-Analyse:

1. Berechne den **Mittelwert** der Punkte, $\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i$
2. Berechne die **Kovarianzmatrix** der Punkte,

$$S = \frac{1}{m-1} \sum_{i=1}^m (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

3. Berechne die **Eigenvektoren** \mathbf{v}_i von S und die zugehörigen **Eigenwerte** λ_i . Sortiere sie absteigend ($\lambda_i \geq \lambda_{i+1}$)

Die Eigenvektoren \mathbf{v}_i sind die Hauptkomponenten, die Eigenwerte λ_i geben die Varianz in der jeweiligen Richtung an. Die totale Varianz ergibt sich als Summe aller Eigenwerte.

Dimensionsreduktion per Hauptkomponenten

- Formmodelle nutzen nur die wichtigsten Hauptkomponenten

- *Mögliches Kriterium:* Kleinste Dimension t , die mindestens z.B. $\theta = 98\%$ der totalen Varianz erhält:

$$\frac{\sum_{i=1}^t \lambda_i}{\sum_{i=1}^p \lambda_i} \geq \theta$$

- Bilde aus den entsprechenden Spaltenvektoren \mathbf{v}_i eine Matrix \mathbf{P}

- Berechnung der Formparameter $\mathbf{b} \in \mathbb{R}^t$ per Projektion
$$\mathbf{b} = \mathbf{P}^T (\mathbf{x} - \bar{\mathbf{x}})$$

- Erzeugen einer Form aus \mathbf{b} per $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$

- Koeffizienten von \mathbf{b} sollten im Rahmen der Trainingsvarianz liegen, z.B. $b_i \in [-3\sqrt{\lambda_i}, 3\sqrt{\lambda_i}]$

Eigenvalue	$\frac{\lambda_i}{\lambda_T} \times 100\%$
λ_1	37%
λ_2	17%
λ_3	13%
λ_4	7%
λ_5	6%
λ_6	4%

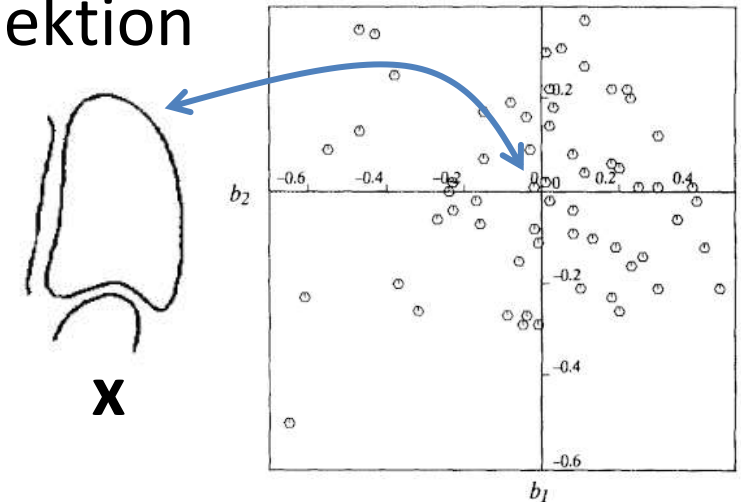
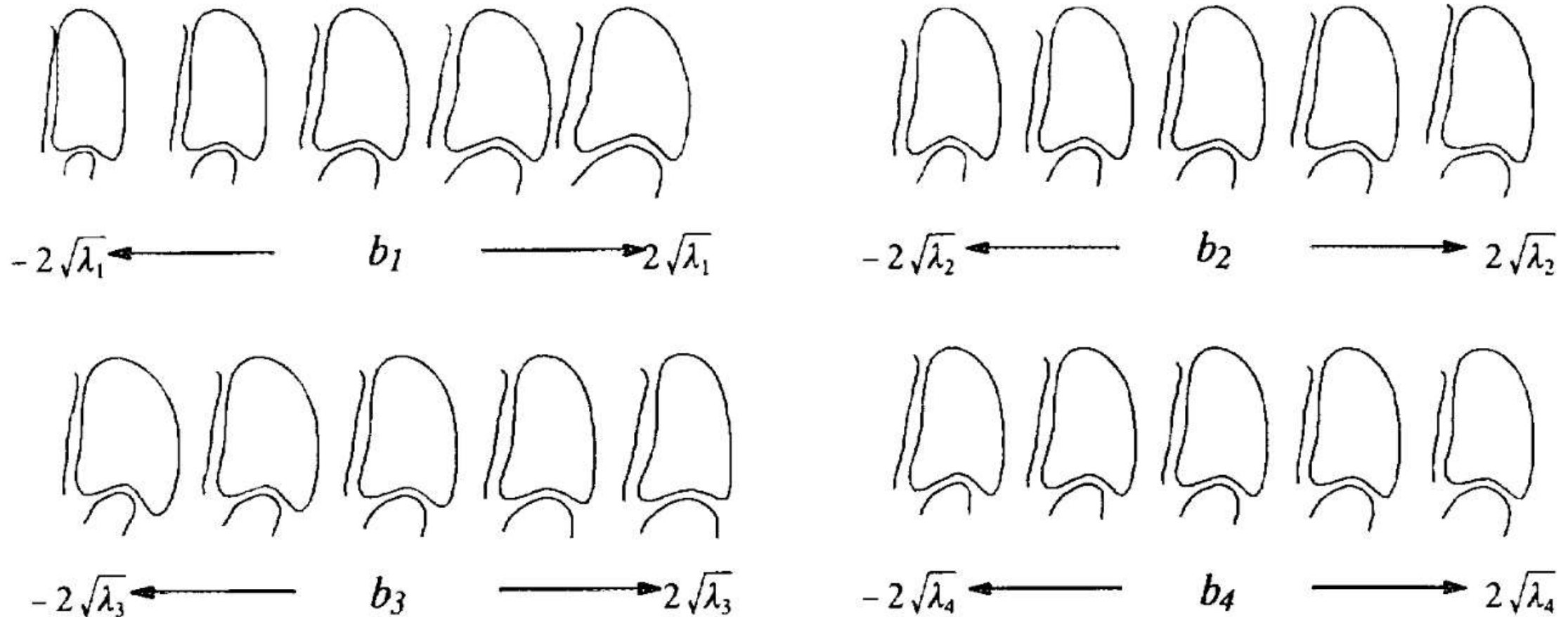


Illustration: Hauptkomponenten eines Formmodells

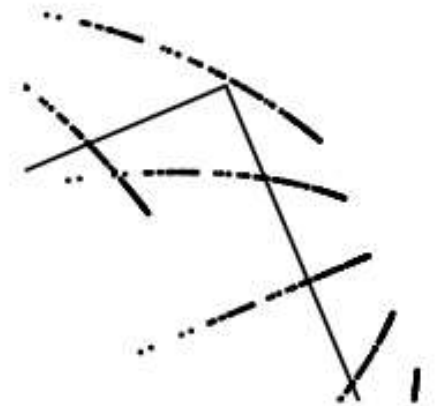
- Visualisierung der Hauptkomponenten zeigt die vom Modell zugelassenen Deformationen



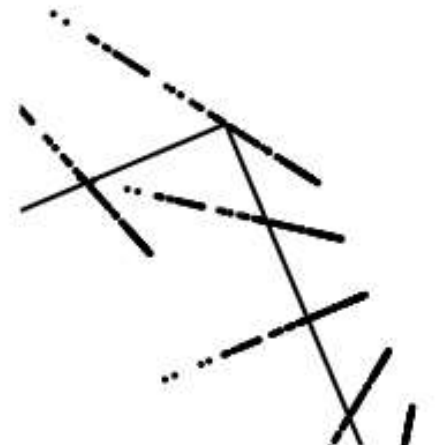
Technisches Detail: Tangentialraum von $\bar{\mathbf{x}}$



- Deformationen von $\bar{\mathbf{x}}$ in Richtung $\bar{\mathbf{x}}$ skalieren die Form. Da Skalierung Teil der Pose ist, ist dies unerwünscht.
- Zentrierte Formen \mathbf{z} werden daher vor ihrer Einbettung in den Formenraum so auf \mathbf{x}' abgebildet, dass die entsprechende Deformation zu $\bar{\mathbf{x}}$ orthogonal ist
 - Kriterium: $(\mathbf{x}' - \bar{\mathbf{x}}) \cdot \bar{\mathbf{x}} = 0$
 - Wegen $\|\bar{\mathbf{x}}\| = 1$ ergibt sich die Bedingung $\mathbf{x}' \cdot \bar{\mathbf{x}} = 1$
 - Führt zur Skalierung $\mathbf{x}' = \frac{1}{\mathbf{z} \cdot \bar{\mathbf{x}}} \mathbf{z}$
 - Wird als „Projektion auf den Tangentialraum“ bezeichnet



Punkte von $\frac{\mathbf{z}}{\|\mathbf{z}\|}$



Punkte von \mathbf{x}'

Anpassen des Modells an gegebene Punkte

Algorithmus zur iterativen Berechnung der Form- und Posenparameter aus gegebenen Punkten \mathbf{y}_B im Bildraum:

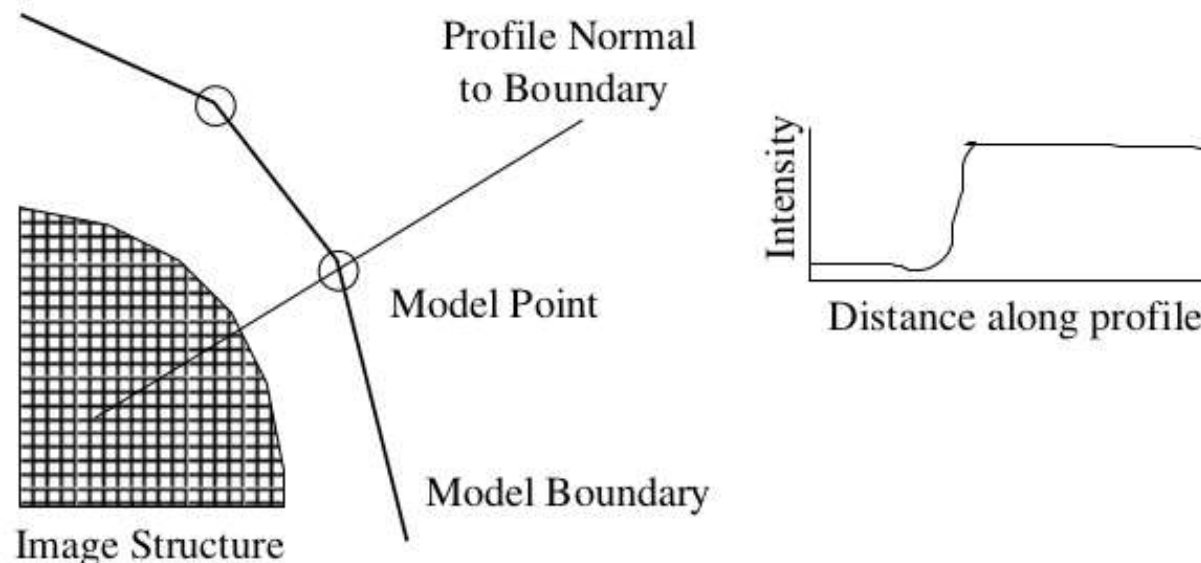
1. Initialisiere die Formparameter auf $\mathbf{b} = 0$
2. Wiederhole, solange Pose oder Form sich um mehr als ϵ ändern:
 - i. Erzeuge die aktuelle Form: $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$
 - ii. Bestimme s, \mathbf{t}, θ , die \mathbf{x} an \mathbf{y}_B ausrichten.
 - iii. Transformiere \mathbf{y}_B in den Modellraum: $\mathbf{y} = T_{s,\mathbf{t},\theta}^{-1}(\mathbf{y}_B)$
 - iv. Projiziere \mathbf{y} in den Tangentialraum von $\bar{\mathbf{x}}$: $\mathbf{y}' = \frac{1}{\mathbf{y} \cdot \bar{\mathbf{x}}} \mathbf{y}$
 - v. Berechne die zu \mathbf{y}' passenden Modellparameter: $\mathbf{b} = \mathbf{P}^T(\mathbf{y}' - \bar{\mathbf{x}})$

Bildsegmentierung mit Formmodellen

- Ähnlich wie **aktive Konturen** können sich grob initialisierte Formmodelle zur Segmentierung aktiv an Bildinhalte anpassen
- Grundsätzlich iteriert man dabei folgende Schritte, bis sich die Modell-Parameter kaum noch ändern:
 1. Finde in der Umgebung jeden Punkts der aktuellen Form eine optimale Position im Bild (Details: nächste Folien)
 2. Passe das Modell an diese neuen Punkte \mathbf{y}_B an
 3. Schränke die gefundenen Parameter auf plausible Werte ein
 - Übliche Bedingung: $|b_i| < 3\sqrt{\lambda_i}$

Suche nach stärkster Kante

- Einfache **kantenbasierte Suche** in Schritt 1:
 - Bestimme innerhalb eines Suchradius senkrecht zur aktuellen Kontur den Ort der stärksten Kante (s. Kapitel 1)
 - Nutze ggf. Vorwissen über die **Orientierung**: Ist das gesuchte Objekt heller oder dunkler als seine Umgebung?



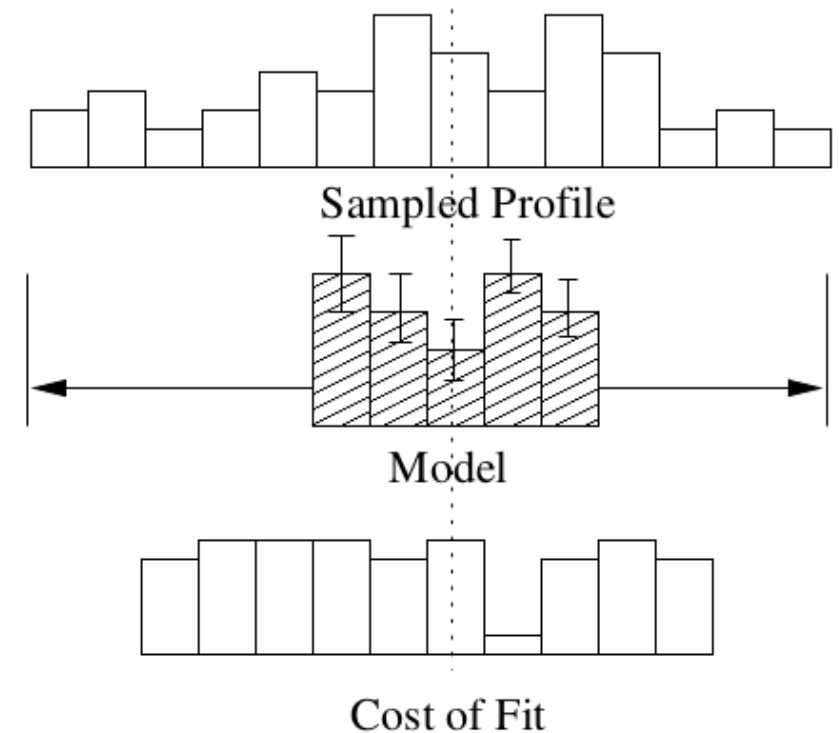
Lernen eines Kantenprofils

- *Idee*: Lerne aus den Trainingsbildern für jeden Stützpunkt nicht nur die Position, sondern auch ein Kantenprofil, um in Schritt 1 eine gezieltere Suche zu ermöglichen
 - Bestimme für k Pixel auf jeder Seite der Kontur die Intensitätsableitungen entlang der Normalen. Normiere die resultierenden Vektoren $\mathbf{g}_i \in \mathbb{R}^{2k+1}$
$$\tilde{\mathbf{g}}_i = \frac{\mathbf{g}_i}{\|\mathbf{g}_i\|_1}$$
um globale Unterschiede im Kontrast auszugleichen
 - Berechne für jeden Punkt ein mittleres Profil $\bar{\mathbf{g}}$ und eine Kovarianz $\mathbf{S}_{\mathbf{g}}$

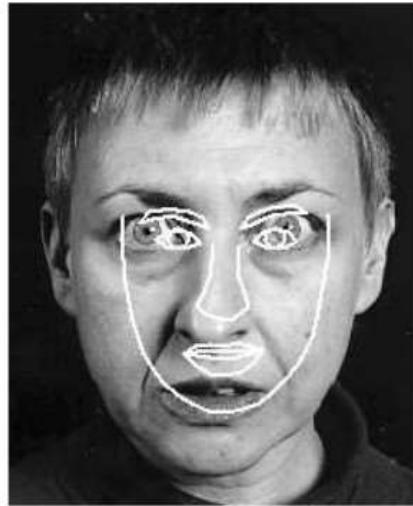
Suche nach plausiblen Kantenprofil

- Verfeinerte **Suche mittels des gelernten Kantenprofils:**
 - Berechne in Schritt 1 für jeden Kandidatenpixel p senkrecht zur Kontur das entsprechende normierte Kantenprofil $\tilde{\mathbf{g}}_p$
 - Wähle den Kandidaten mit der geringsten **Mahalanobis-Distanz** als optimale Position

$$D(\tilde{\mathbf{g}}_p) = (\tilde{\mathbf{g}}_p - \bar{\mathbf{g}})^T \mathbf{S}_{\bar{\mathbf{g}}}^{-1} (\tilde{\mathbf{g}}_p - \bar{\mathbf{g}})$$



Beispiele: Segmentierung mit Aktiven Formmodellen



Initial



After 2 iterations

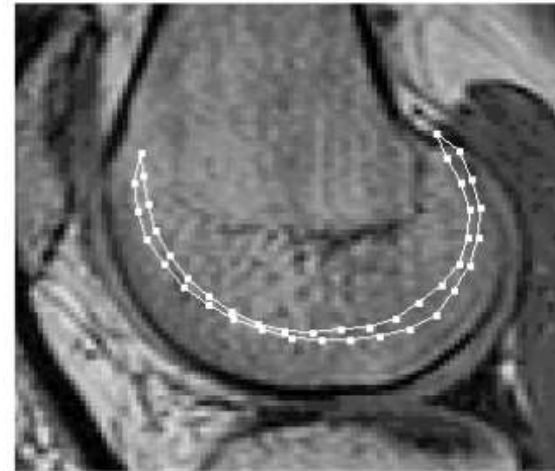


After 6 iterations

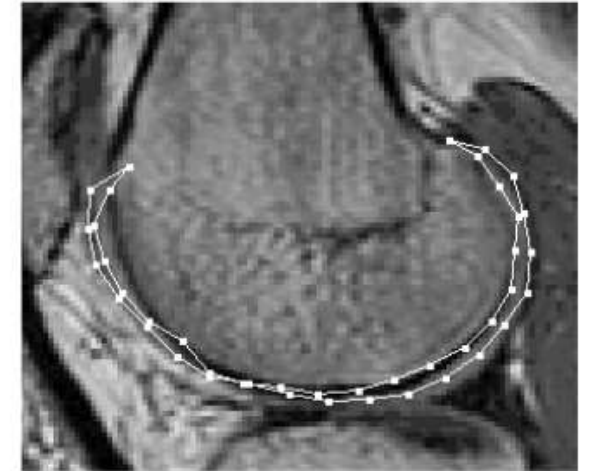


After 18 iterations

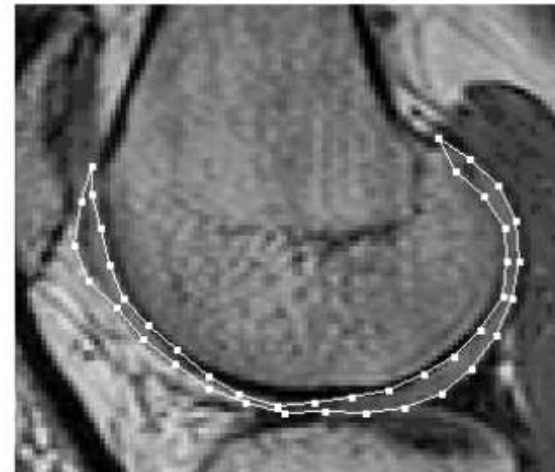
Gesichtserkennung



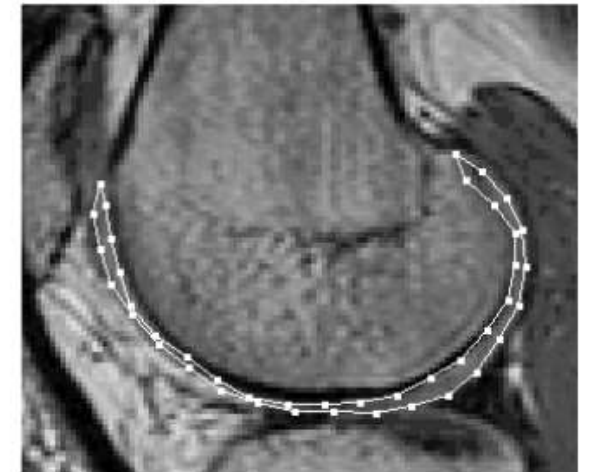
Initial



After 1 iteration



After 6 iterations



After 14 iterations

Knorpelschicht in Knie-MRT

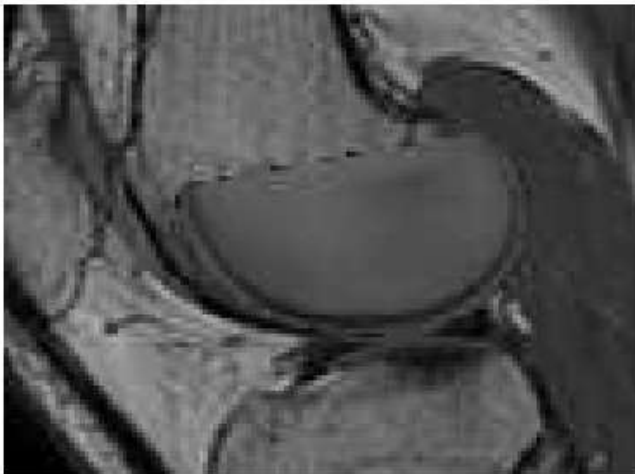
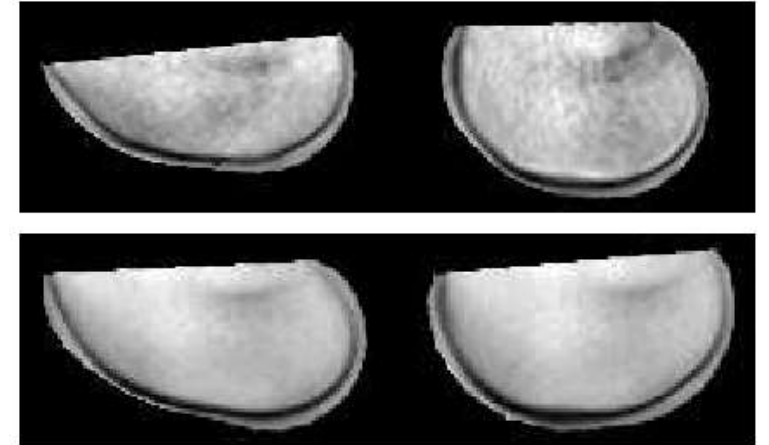
Zusammenfassung: Aktive Formmodelle

- **Aktive Formmodelle** nutzen zur Segmentierung Vorwissen über die Form der gesuchten Struktur
 - *Vorteil*: Hohe Robustheit, wenn das Modell tatsächlich passt
 - Nicht geeignet für anomales Wachstum (z.B. Tumore) oder andere Verletzungen der Modellannahmen (z.B. Knochenbrüche)
- Wesentliche **Ideen** und **Bausteine** der Formmodelle sind:
 - Erlernen eines Modells aus annotierten **Trainingsdaten**
 - **Prokrustes-Analyse** zur Bestimmung relativer Posen
 - **Hauptkomponenten-Analyse** zur Dimensionsreduktion
 - Suche nach geeigneten Modellparametern über **stärkste Kanten** oder erlernte **Kantenprofile**

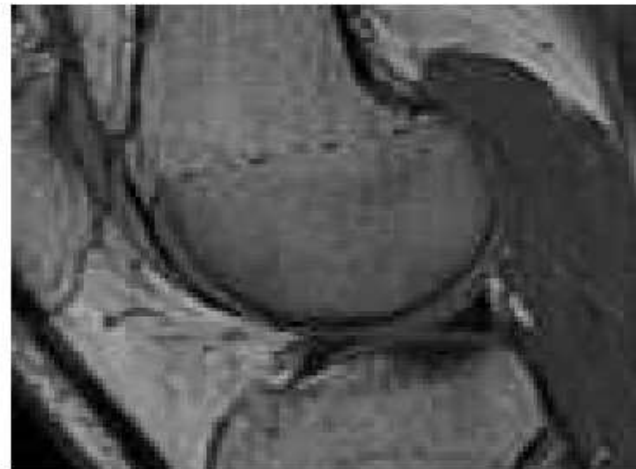
Ausblick: Aktive Erscheinungsmodelle

Aktive Erscheinungsmodelle (*engl.* Active Appearance Models) erweitern aktive Formmodelle um ein statistisches Modell des erwarteten Erscheinungsbilds (Grau- bzw. Farbwerte) in der Referenzform

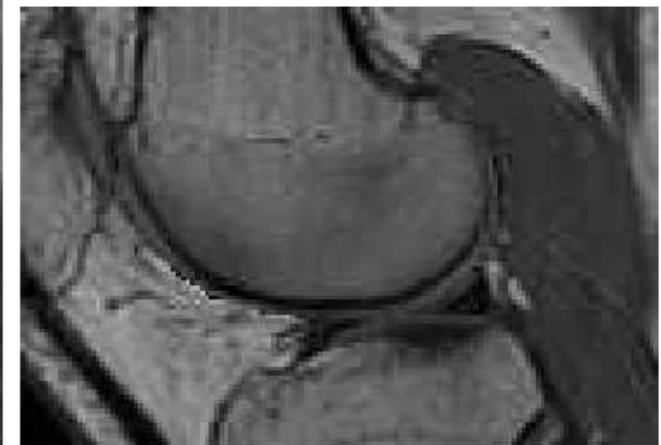
- Ansatz: Analyse durch Synthese



Initial



2 its



Converged (11 its)

Zum Nach- und Weiterlesen

- Heinz Handels: *Medizinische Bildverarbeitung*. Vieweg+Teubner, 2. Auflage, 2009
- Klaus D. Toennies: *Guide to Medical Image Analysis. Methods and Algorithms*. Springer, 2012
- I.N. Bankman: *Handbook of Medical Imaging. Processing and Analysis*. Academic Press, 2000

Zum Nach- und Weiterlesen: Level-Set-Methode

- **YouTube-Video:**

- Vorlesung „Variational Methods for Computer Vision“ von Prof. Daniel Cremers (TUM)

- **Bücher:**

- Sethian: Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999
- Osher/Fedkiw: Level Set Methods and Dynamic Implicit Surfaces, Springer 2003
- Mitiche/Ayed: Variational and Level Set Methods in Image Segmentation, Springer 2011

Zum Nach- und Weiterlesen: Aktive Formmodelle

- TF Cootes, CJ Taylor, DH Cooper, J Graham: *Active Shape Models – Their Training and Application*. Computer Vision and Image Understanding, 1995
- Tim Cootes: *An Introduction to Active Shape Models*. Kapitel 7 in „Model-Based Methods in Analysis of Biomedical Images“, Oxford University Press, 2000
- TF Cootes, GJ Edwards, CJ Taylor: *Active Appearance Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001
- Mikkel B. Stegmann: *Active Appearance Models. Theory, Extensions & Cases*. MSc-Arbeit an der Technical University of Denmark (DTU), 2000