Medical Image Analysis
Lecture 05

Image Segmentation & Deformable Models

Martin Urschler

Definition of Image Segmentation

- Very important but very hard Computer Vision problem
- Separation of an image into (disjoint) meaningful pieces
  - Potential features: intensity, gradients, texture measures, shape information

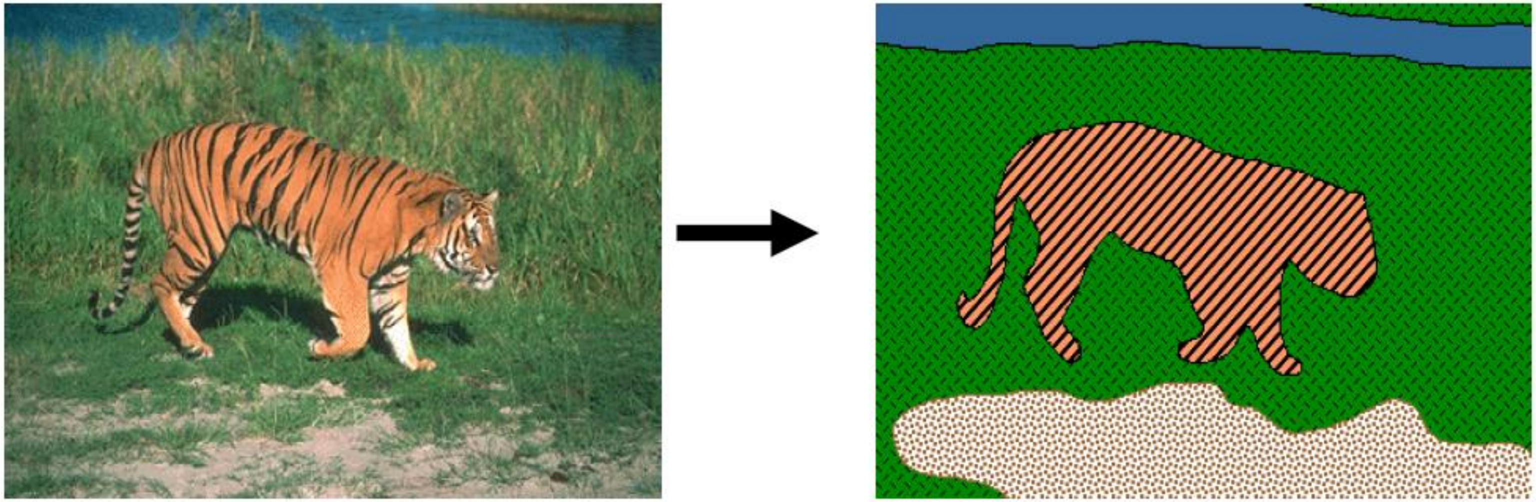
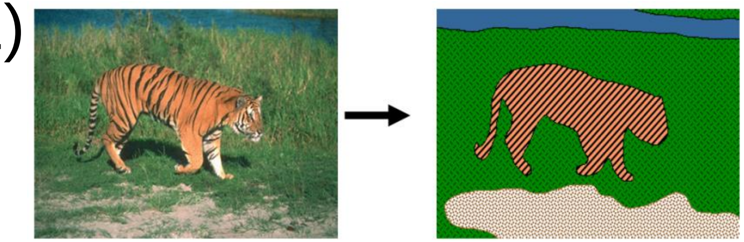
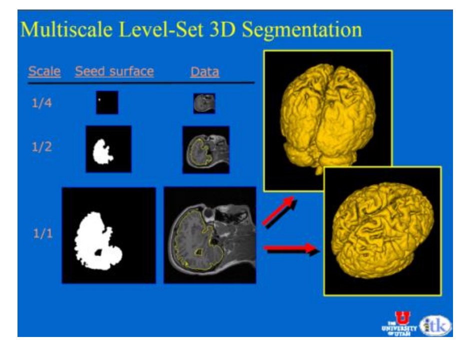
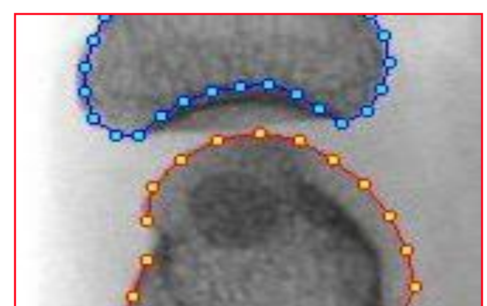


Image Segmentation Overview

- Basic Low-Level Methods (BVME)
  - Thresholding, Class Labeling, Edge-based, Region-based, Watersheds, …
- High-Level Deformable Models
  - Active Contours
  - Level Set Methods
- Shape Prior based Deformable Models
  - Shape Template Matching
  - Active Shape/Appearance Model
- Vascular Structures





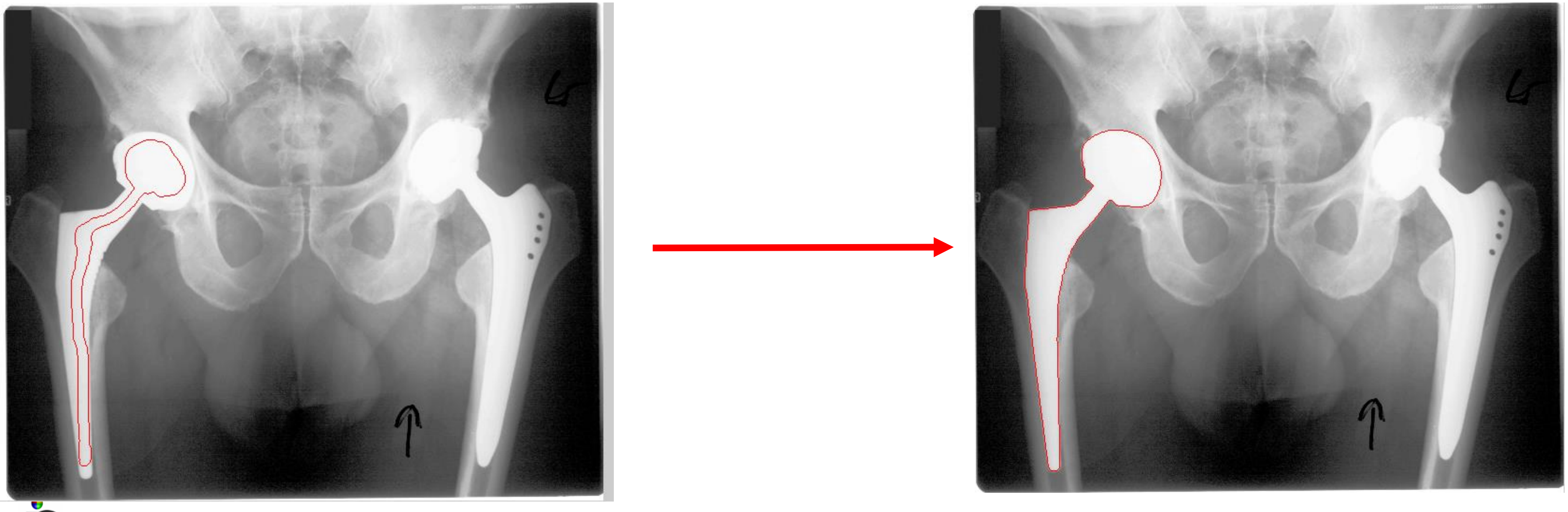


High-Level Deformable Models

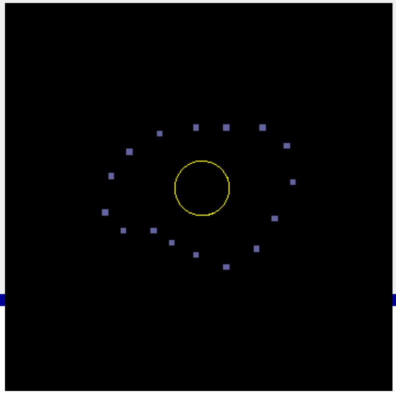
- Active Contours Method Overview
  - Snakes, Gradient Vector Flow Snakes
  - Level Sets in general and used for Geodesic Active Contours
  - Weighted Total Variation used for Geodesic Active Contours

High-Level Deformable Models

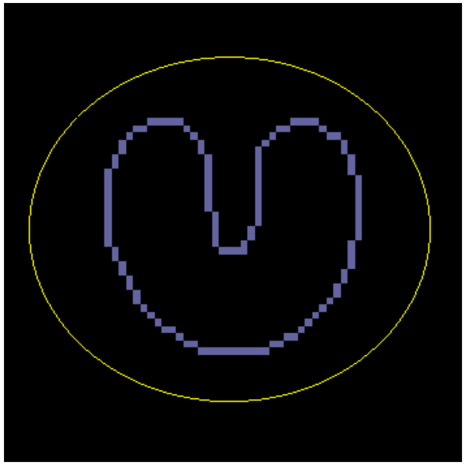
- Motivation
  - Given: image & initial contour
  - Task: compute segmentation

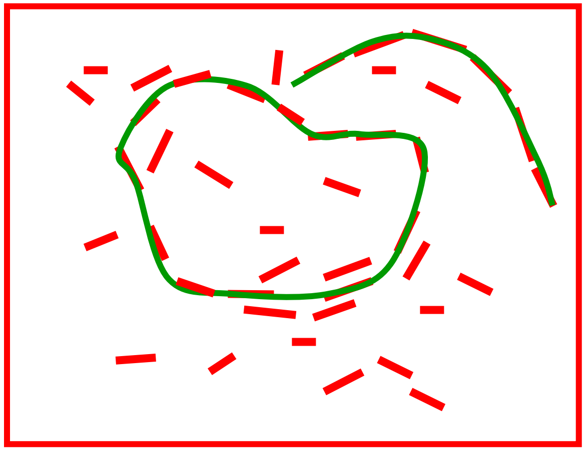


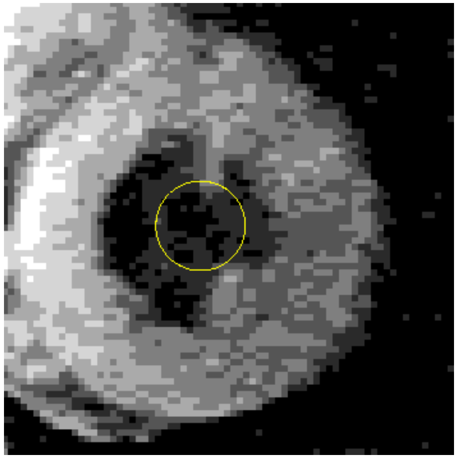
Deformable Models



- Ideas of deformable model segmentation was made popular by Kass et al. in 1988
  - The „Snakes“ model, a.k.a. the „Active Contour“ algorithm





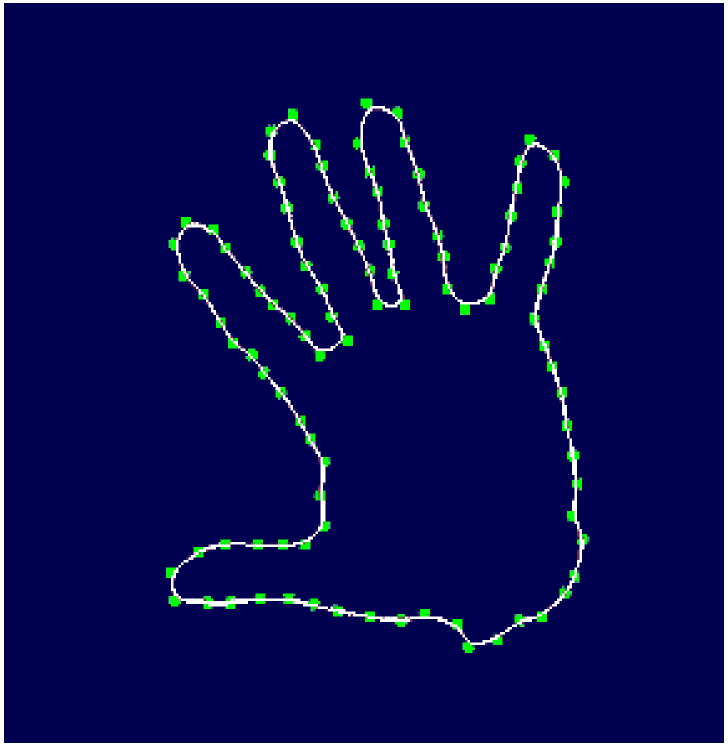


Images taken from Xu, Prince: Website on „Gradient Vector Flow Snakes“

Deformable Models - Snakes

Parameterized Curve C(s):

$$\vec{C}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} \qquad \vec{C}(s):[0,1] \rightarrow \Re^2$$



Example:

$$\vec{C}(s) = \sum_{j=1}^{n} \vec{x}_j B_j(s)$$

$$\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$$

n control points

basis functions
e.g. B-Splines

Deformable Models - Snakes

Explicitly Modeled Contour C:

$$\vec{C}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}$$

Find curve that minimizes energy functional defined by

internal forces: elasticity and bending

external forces: image information (e.g. gradient strength)

$$\min_{\vec{C}} \left\{ E(\vec{C}(s)) = \int_0^1 E_{\text{int}}(\vec{C}(s))ds + \int_0^1 E_{ext}(\vec{C}(s))ds \right\}$$

$$E_{\text{int}}(\vec{C}) = \frac{1}{2}\alpha\left|\vec{C}'(s)\right|^2 + \frac{1}{2}\beta\left|\vec{C}''(s)\right|^2$$

Penalize derivatives of C -> „short & smooth“ curve

$$E_{ext}(\vec{C}) = -\frac{1}{2}\left|\nabla I(\vec{C}(s))\right|^2$$

Penalize negative gradient magnitude -> move towards edges

Snakes – Active Contour Models

$$E(\vec{C}) = \frac{1}{2}\alpha\int_0^1 \left|\vec{C}'(s)\right|^2 ds + \frac{1}{2}\beta\int_0^1 \left|\vec{C}''(s)\right|^2 ds + \frac{1}{2}\int_0^1 \left(-\left|\nabla I(\vec{C}(s))\right|^2\right) ds$$

E_{ext}

- Minimizer can be found by Euler-Lagrange equations

∇E_{ext}

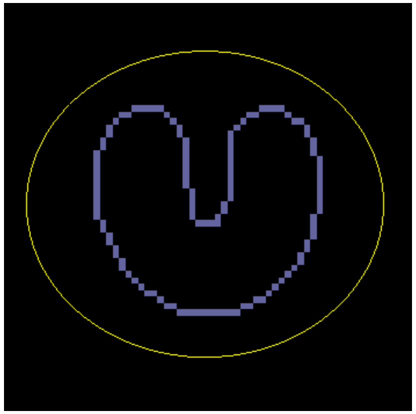
$$-\nabla\left|\nabla I(\vec{C}(s))\right|^2 - \alpha\vec{C}''(s) + \beta\vec{C}''''(s) = 0$$

- Solve by introducing artificial time dependency -> gradient descent evolution scheme

$$\vec{C}_t(s,t) = -\frac{dE(\vec{C})}{d\vec{C}} = \nabla\left|\nabla I(\vec{C}(s,t))\right|^2 + \alpha\vec{C}''(s,t) - \beta\vec{C}''''(s,t)$$

Snakes – Active Contour Models

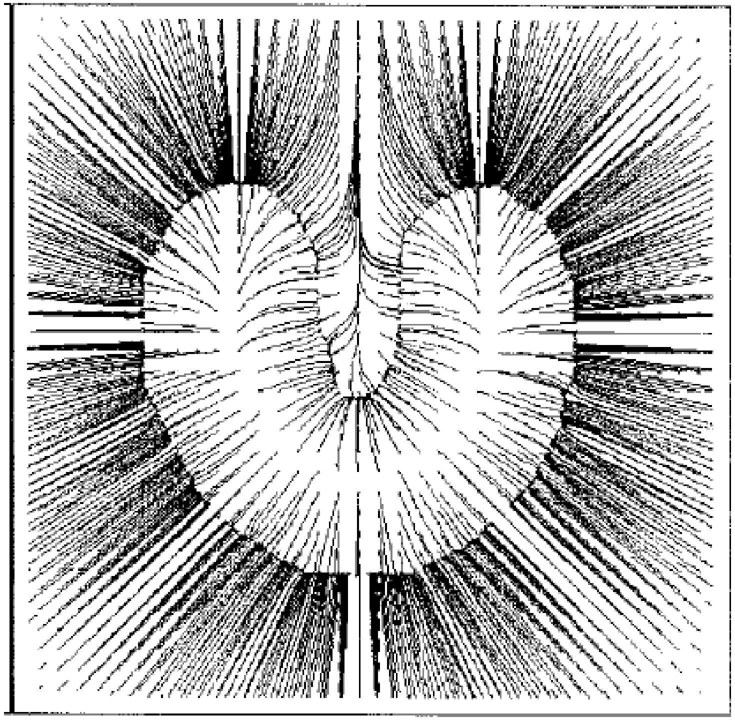
- External Energies
  - Image gradient $$E_{ext}(\vec{C}) = -\frac{1}{2}\gamma \left| \nabla I(\vec{C}(s)) \right|^2$$
  - Distance transform of e.g. Canny edges
    - Improves capture range
  - Gradient vector flow field [1]
    - Concavities, capture range



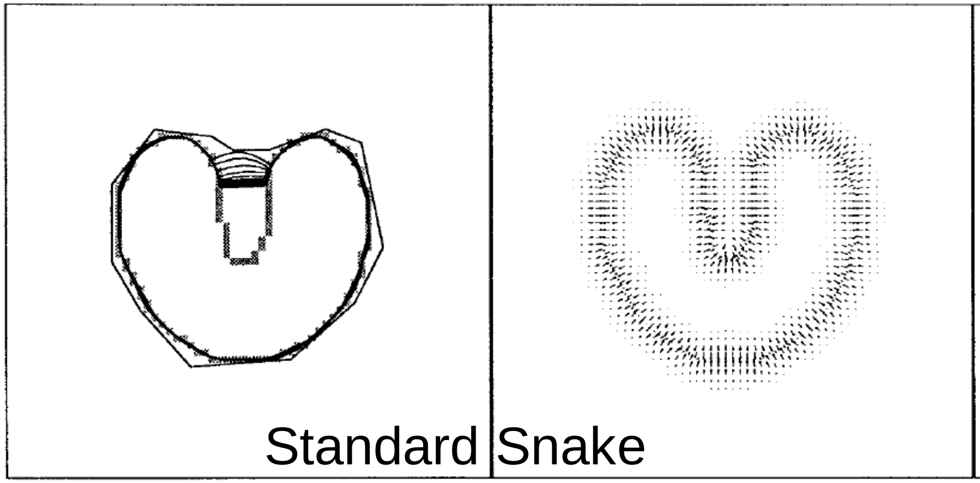
[1] Xu, Prince. Snakes, Shapes and Gradient Vector Flow. *IEEE Transactions on Image Processing*, 1998.

Gradient Vector Flow Snake

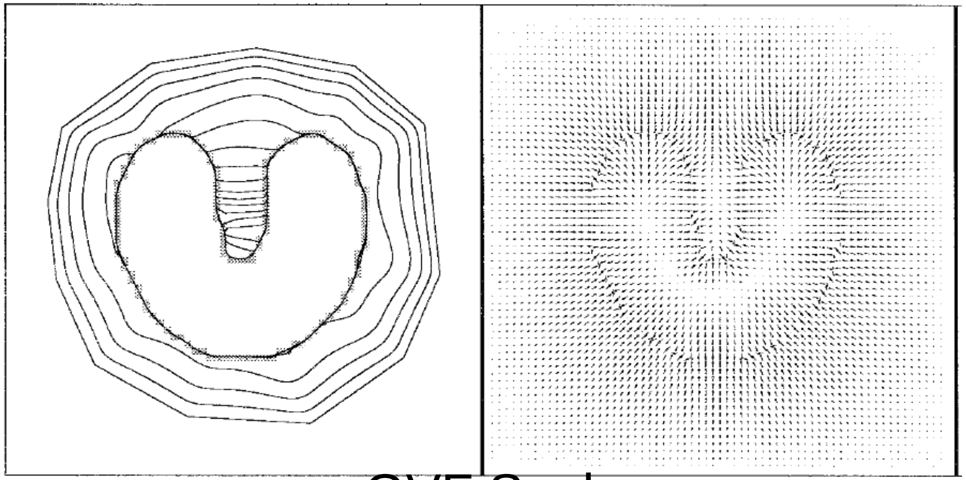
- External Forces



Streamlines

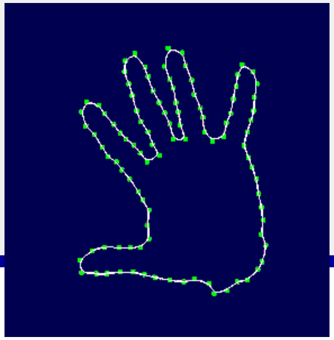


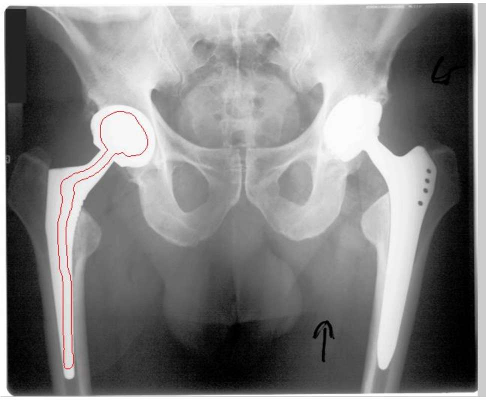
Standard Snake



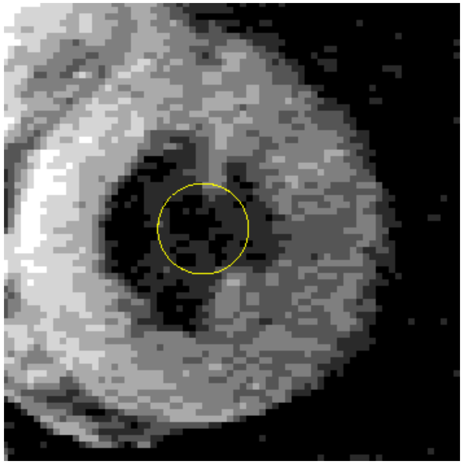
GVF Snake

Deformable Models – Properties of Snakes





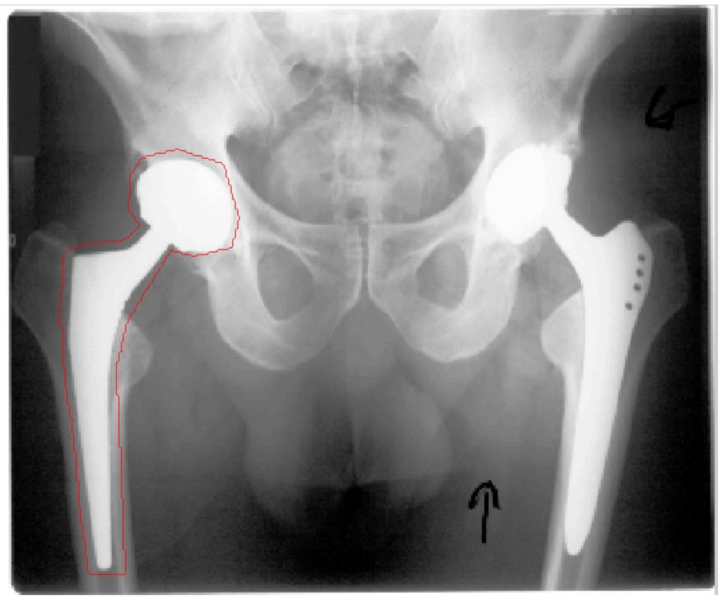
- Efficient to calculate (restricted to contour points)
- Easy incorporation of prior shape models (ASM)
- Number of control points? Reparameterization necessary when curve shrinks or expands!
- Initialization necessary and critical!
  - Optimization is not convex, so we converge to a local minimum
- How to handle topology changes?
  - What should happen if we have one initial contour and want to segment two independent structures?
- Fourth derivative in Euler Lagrange equation
- Parameter Tuning (alpha, beta)



CT of left ventricle

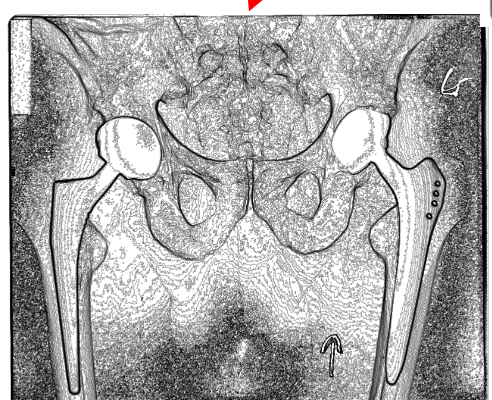
Geodesic Active Contours

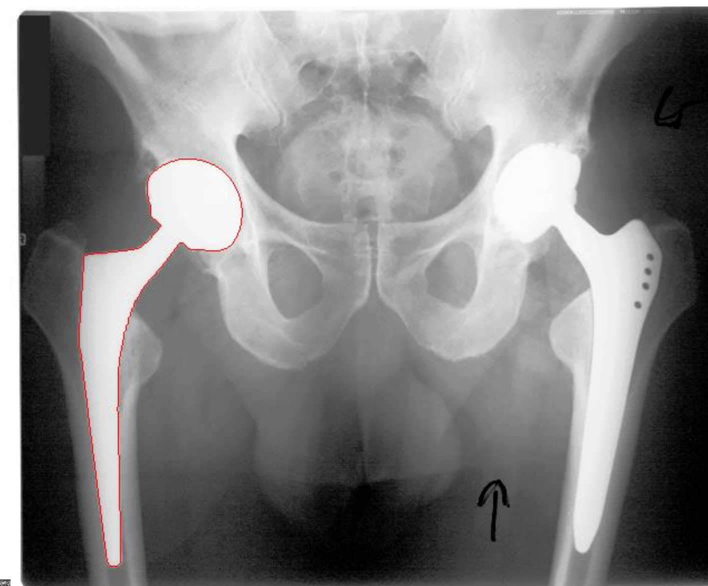
- Snake Model introduced by Caselles et al. (1997)



Initial Contour

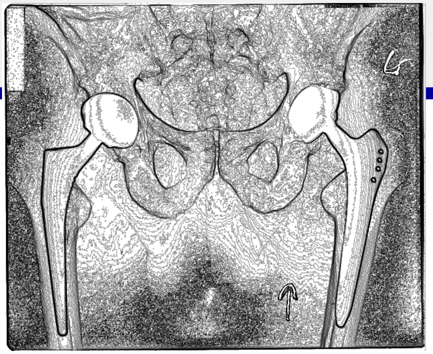
Curve Evolution Using Gradient





Final Contour

Geodesic Active Contours

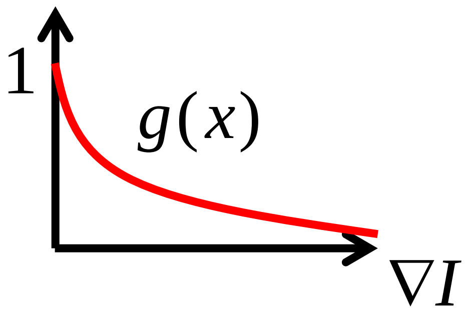


- Based on Classical Snakes Approach

$$E(C)=\alpha\int_0^1\left|\vec{C}'(s)\right|^2 ds+\cancel{\beta\int_0^1\left|\vec{C}''(s)\right|^2 ds}-\int_0^1\left|\nabla I(\vec{C}(s))\right|^2 ds$$

- Ignoring beta and generalizing the external energy to g!

$$E(C)=\alpha\int_0^1\left|\vec{C}'(s)\right|^2 ds+\int_0^1 g\left(\nabla I\left(\vec{C}(s)\right)\right)^2 ds$$



e.g. $g(x)=\dfrac{1}{1+\left|\nabla I_\sigma(x)\right|^2}$

gradient of smoothed image

g is monotonic & high for low gradients!

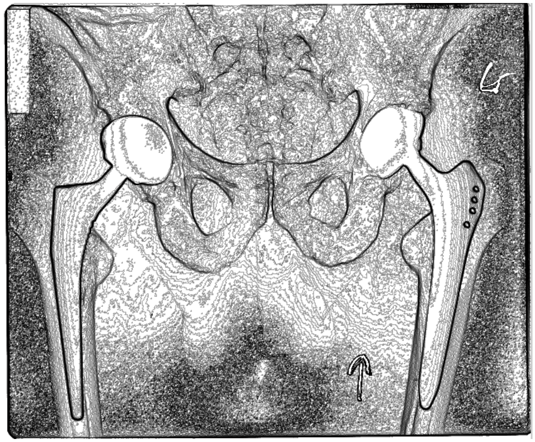
Geodesic Active Contours (GAC)

- *Caselles showed that energy minimization of E can be regarded as finding a geodesic curve in a Riemannian space using a metric derived from the image gradient.*
- Closed curves (surfaces) which evolve to minimize the weighted length (area) with weight derived from image

$$\min_{\vec{C}} \left\{ E_{GAC}(\vec{C}) = \int_0^{L(\vec{C})} g\left(\nabla I\left(\vec{C}(s')\right)\right) ds' \right\}$$

$$L(\vec{C}) = \oint ds'$$

L(C) … euclidean length of C,
ds‘ … Euclidean length element

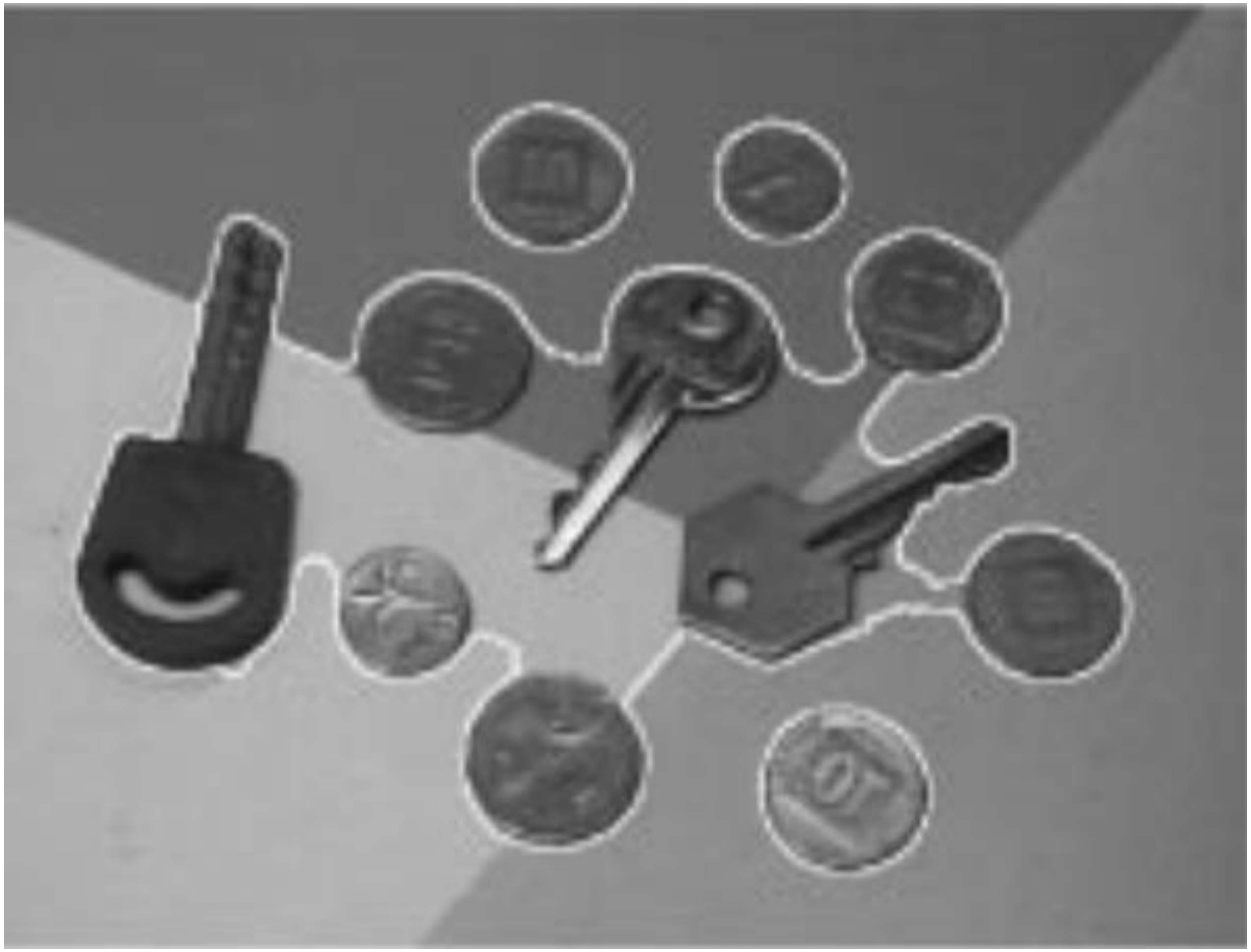


Example for g

Geodesic Active Contours (GAC)

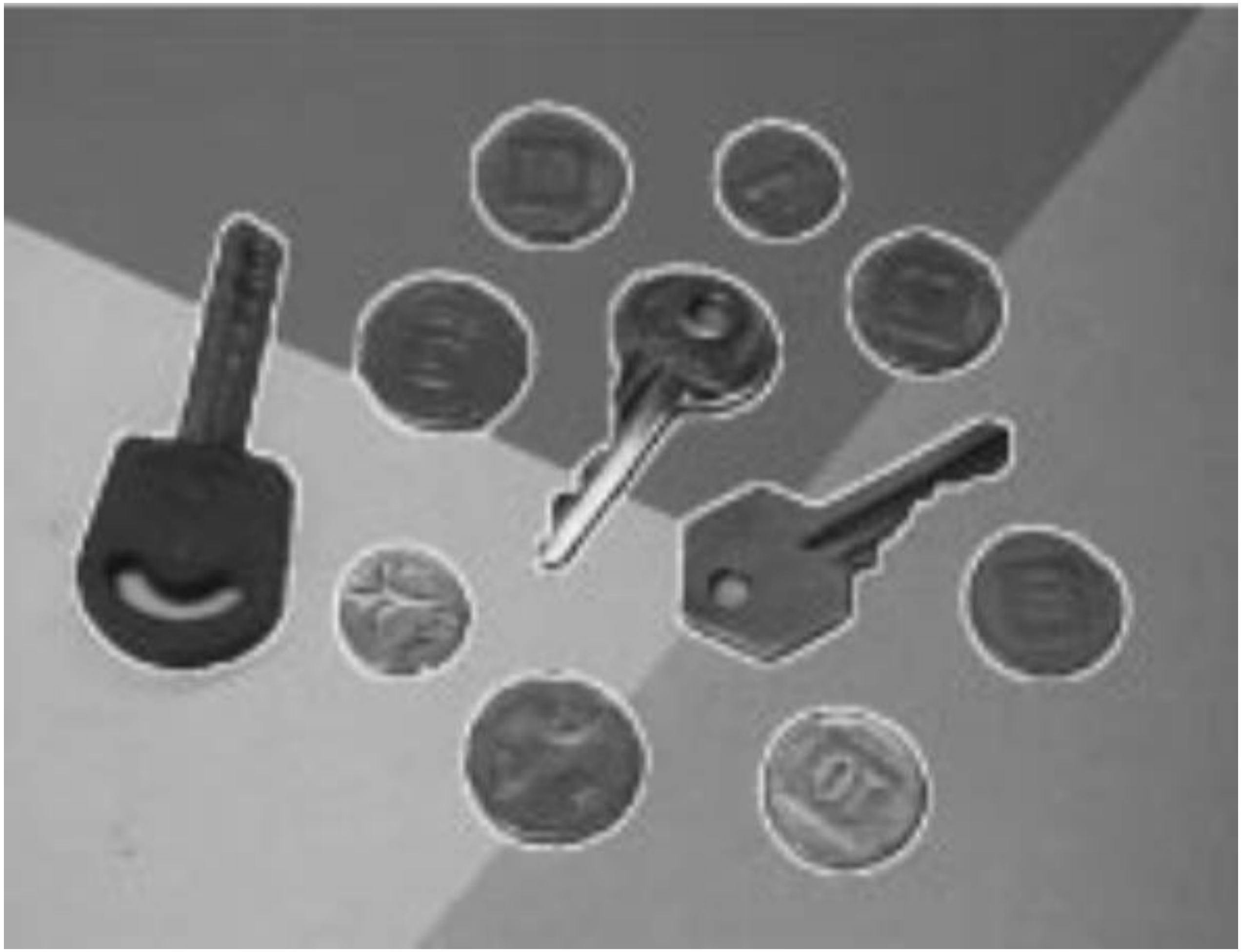
Goldenberg et al., IEEE Transactions Image Processing, 2001

Geodesic Active Contours (GAC)



Goldenberg et al., IEEE Transactions Image Processing, 2001

Geodesic Active Contours (GAC)



Goldenberg et al., IEEE Transactions Image Processing, 2001

GAC Solution – Level Sets

- May be implemented using Level Set Framework

Euler-Lagrange eq. of GAC model:

$$\frac{\partial \vec{C}}{\partial t} = \left(g\kappa - \nabla g \cdot \vec{N} \right) \vec{N} \iff \frac{\partial \phi}{\partial t} = \left| \nabla \phi \right| g\kappa + \nabla g \cdot \nabla \phi$$

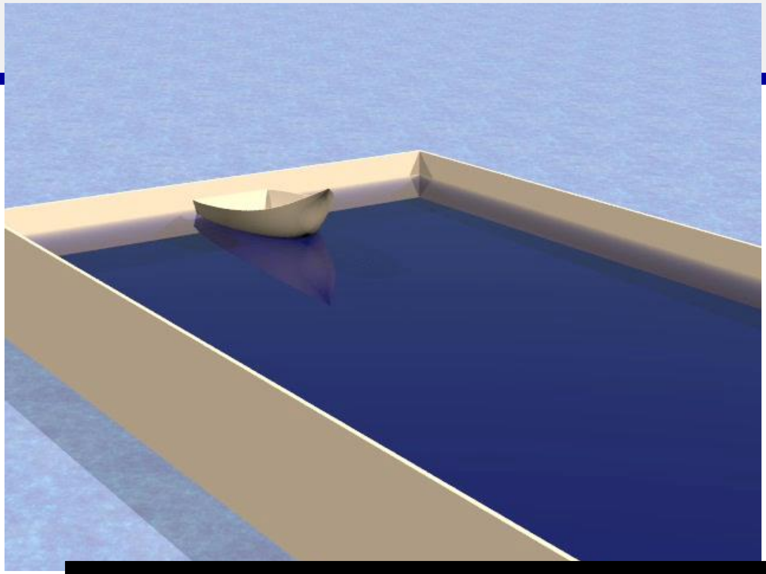
curvature normal to contour

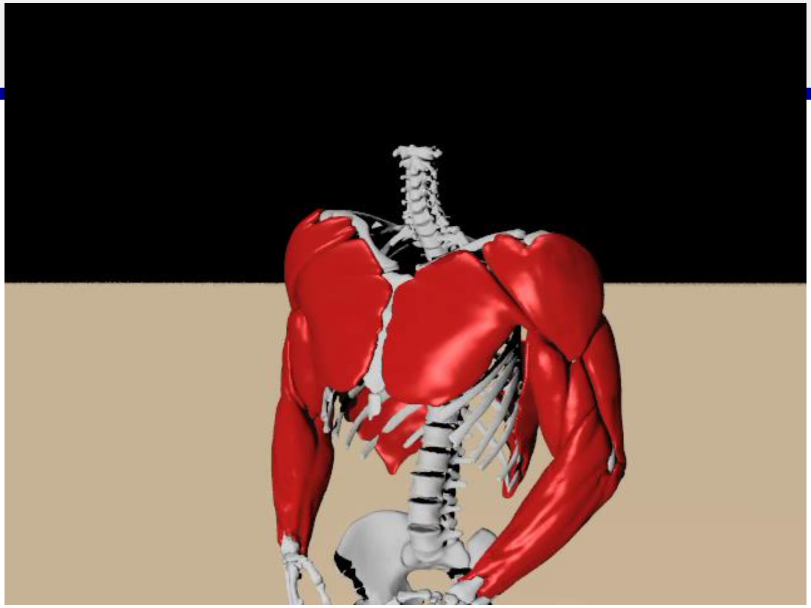
Mean Curvature Flow

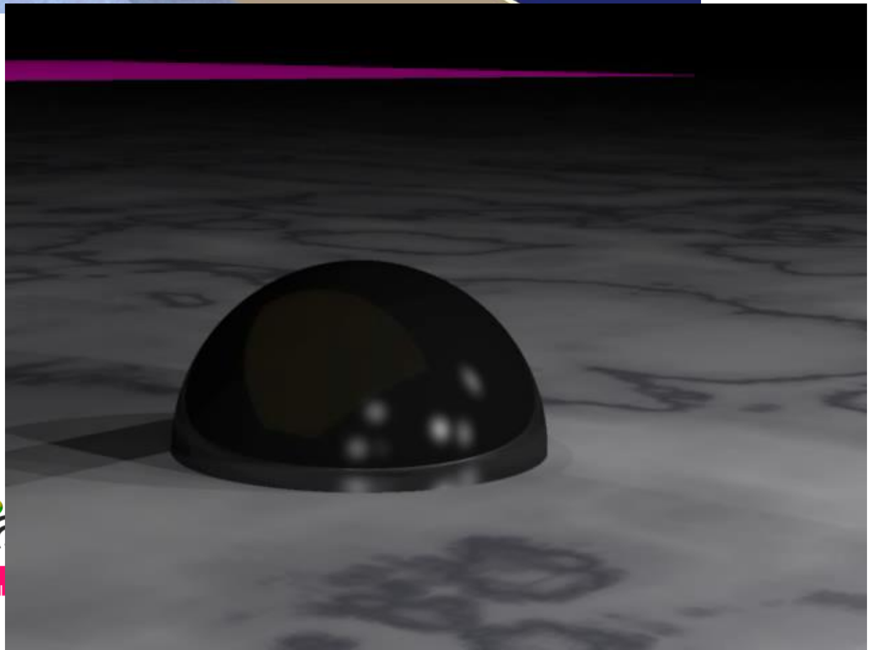
Flow according to external gradient field

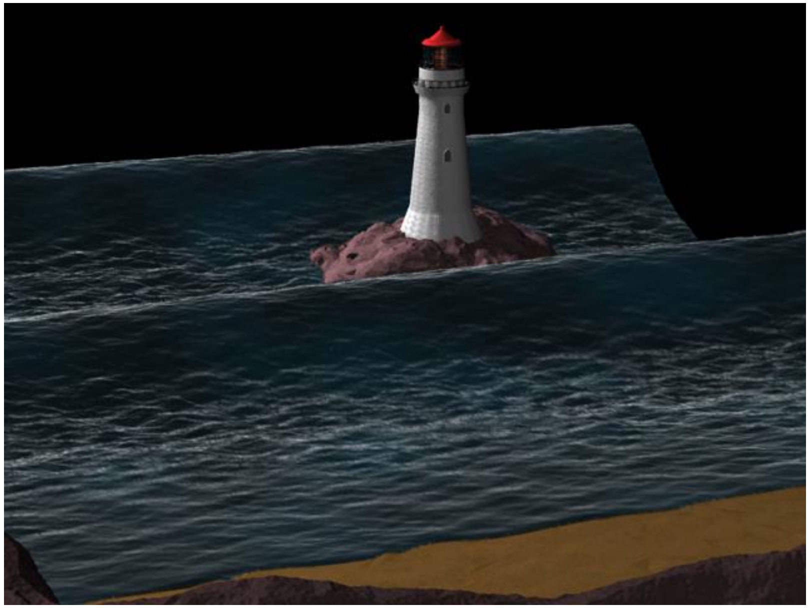
So, what is the Level Set Framework?

Level Set Framework

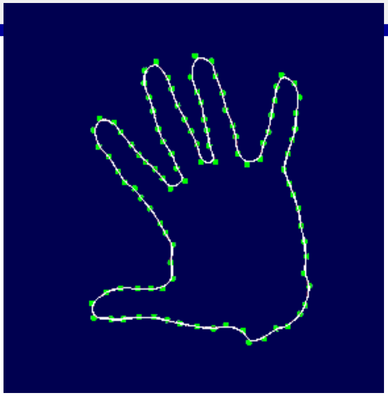








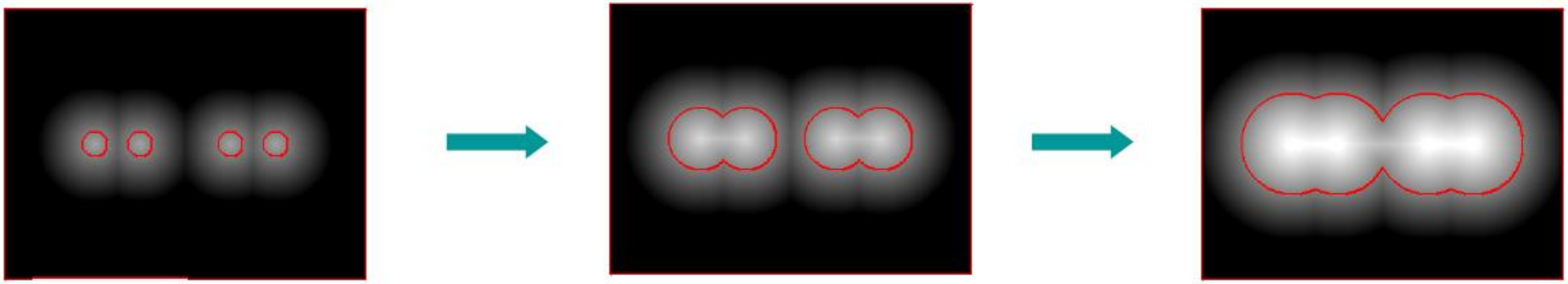
Deformable Models – Level Sets



- Explicitly defined active contours have some problems
  - Shrinking and Growing -> Reparameterization
  - Changes in Topology -> Contours (dis)appear
  - Extension from Contours to Surfaces

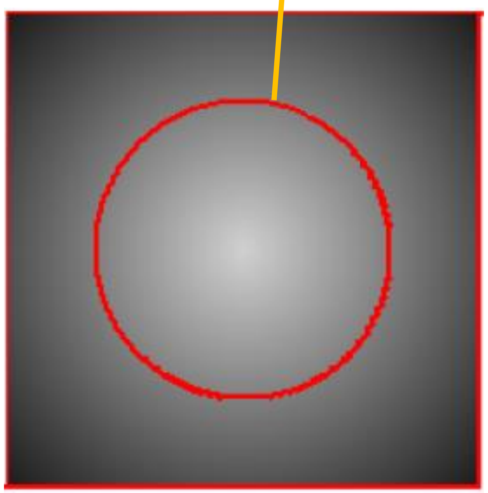
$$\vec{C}(s) = \begin{pmatrix} x(s) & y(s) \end{pmatrix}^T$$

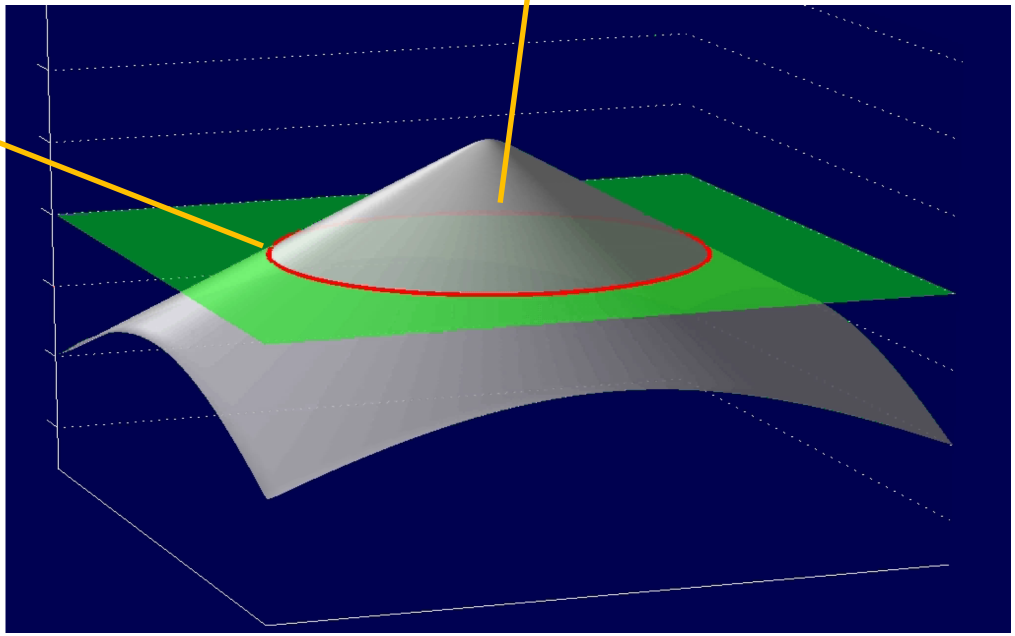
- Standard trick: Go to higher dimensional representation by embedding e.g. a 2D contour in a 3D implicit function



Deformable Models – Level Sets

- Implicit Representation of Active Contours (Osher,Sethian)
- Hypersurface C: zero level set of higher dimensional function



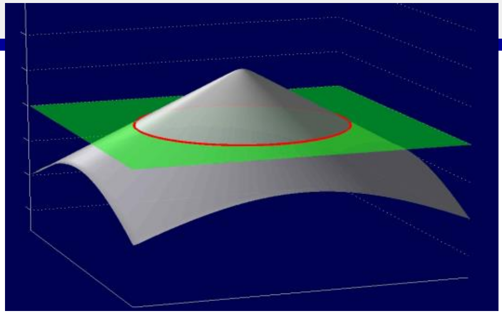


Smooth Implicit Function

$$\vec{C} = \{\vec{x} \in \Omega \mid \phi(\vec{x}) = 0\}, \phi : \Omega \subset \Re^2 \to \Re$$

Deformable Models – Level Sets

$$\vec{C} = \{\vec{x} \in \Omega \mid \phi(\vec{x}) = 0\}, \phi : \Omega \subset \Re^N \rightarrow \Re$$



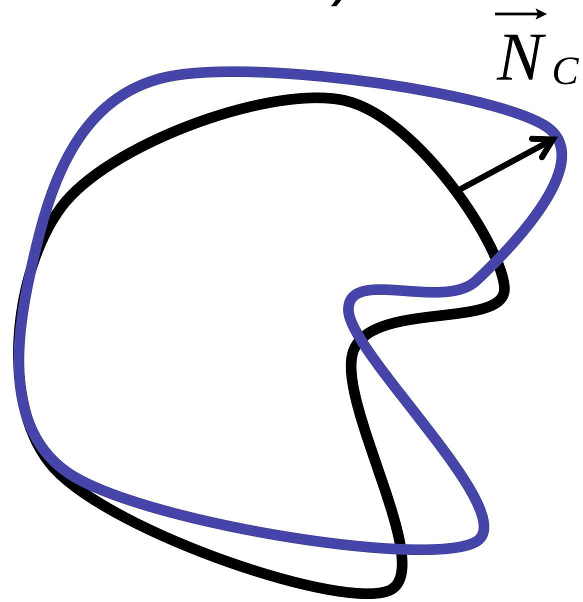
- Implicit, Analytic Representation:
  - No Reparameterization Necessary -> Always investigate Points where Implicit Function equals Zero
  - Evolving and Modifying Implicit Function leads to contour (i.e. interface) motion
    - We Model Motion using Partial Differential Equations (PDE)
  - Topology Changes for free
  - Upgrade from 2D to 3D simple

How does motion work?

- Problems & Difficulties:
  - Discretization and Numerical Solvers of PDEs

Level Set Motion

- We are interested in evolving curves (or interfaces)
- Snakes model had explicitly defined curve C(s) which evolved over time while minimizing energy E(C)



General normal motion of a hypersurface $\vec{C} \subset \Re^n$

$$\frac{d\vec{C}}{dt} = F\vec{N}_{\vec{C}} \qquad E(\vec{C}) \rightarrow \min$$

Level Set Analogon for Normal Motion:

$$\boxed{\frac{\partial \phi}{\partial t} = F\left|\nabla \phi\right|} \qquad \vec{N} = -\frac{\nabla \phi}{\left|\nabla \phi\right|}$$

N … normal direction to interface

Level Set Motion

- We represent ϕ as a signed distance function (Euclidean Distance Transformation)
- Critical question: How to choose speed function F?
  - e.g. we want to stop motion at edges
  - e.g. we want to include region properties
  - e.g. we want to minimize curvature
  - …

$$\boxed{\frac{\partial \phi}{\partial t} = F \left| \nabla \phi \right|}$$

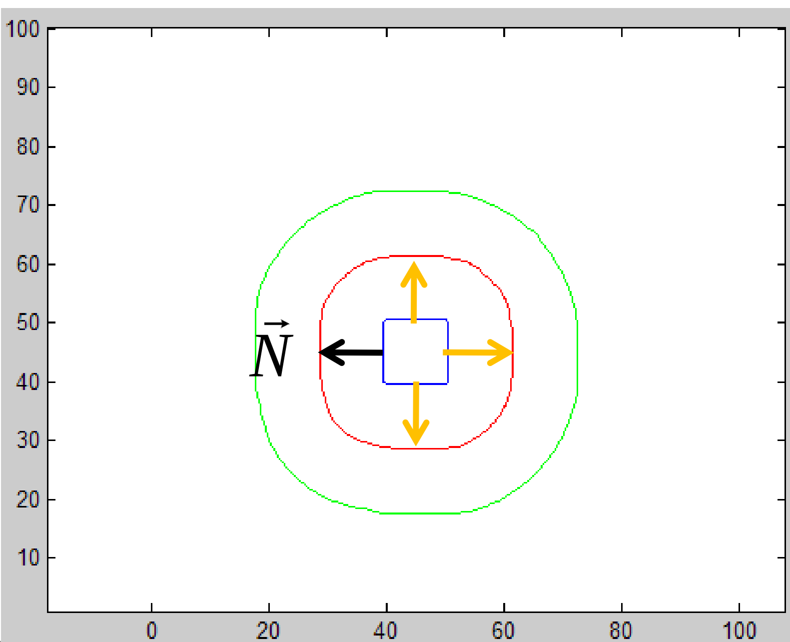
- Three types
  - Normal Flow
  - Mean Curvature Flow
  - Flow according to external velocity field

Level Set Motion - Examples

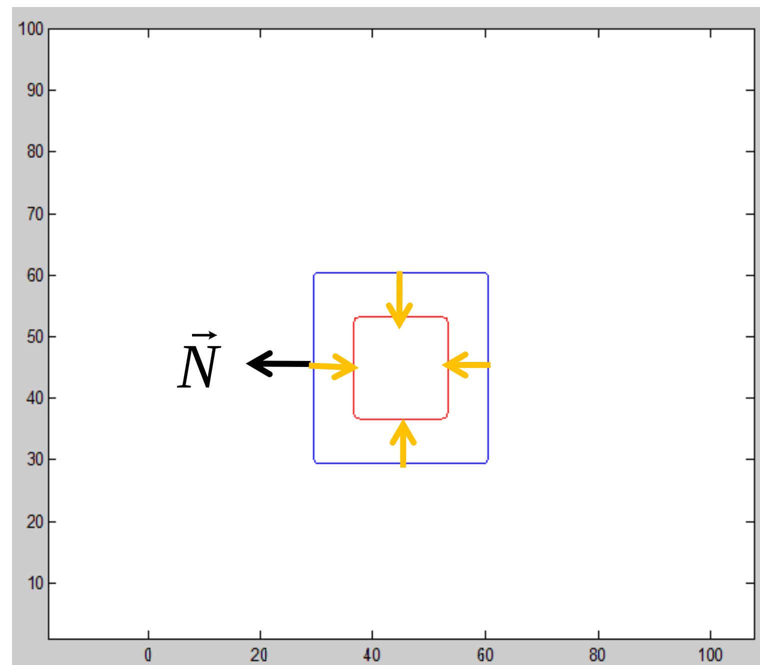
- Normal Flow

$$\frac{\partial \phi}{\partial t} + a\left|\nabla \phi\right| = 0$$

a > 0

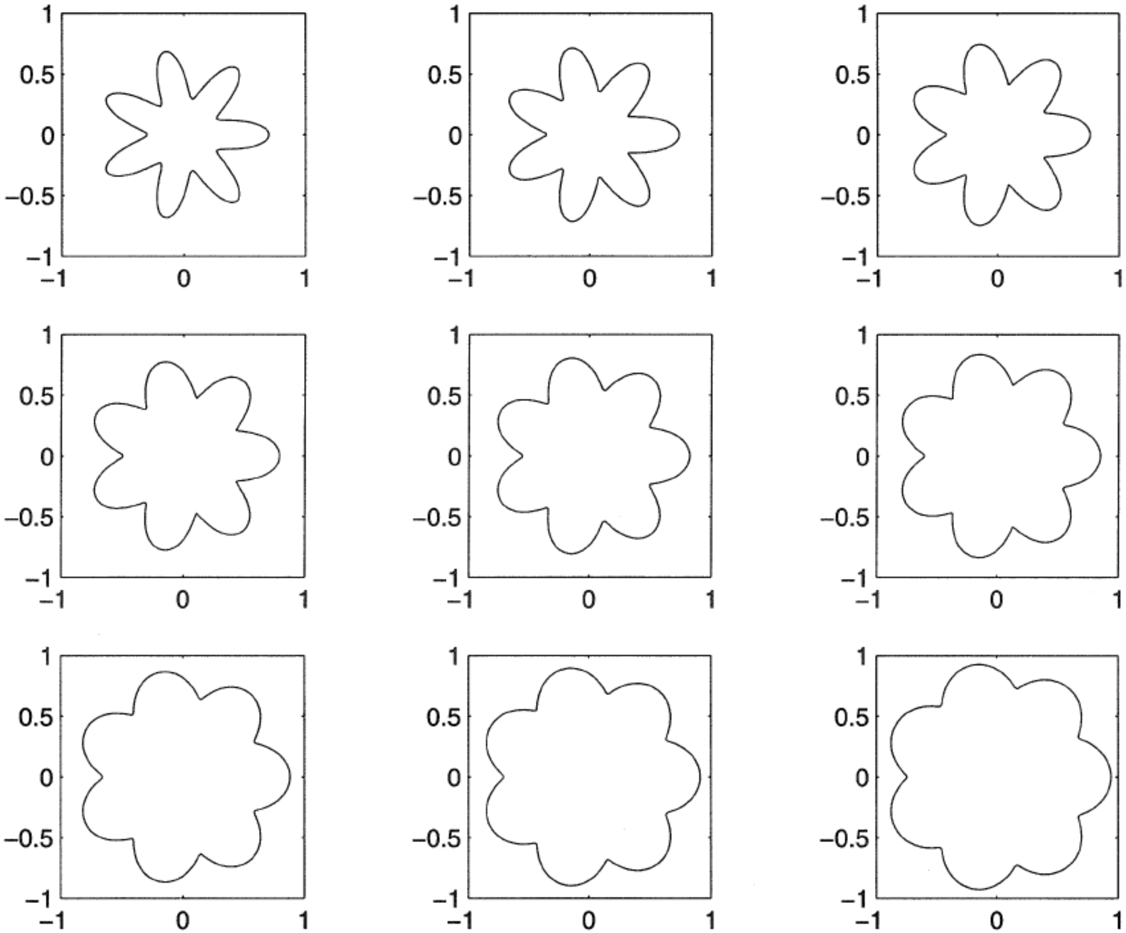


a < 0



Matlab!

Normal Flow



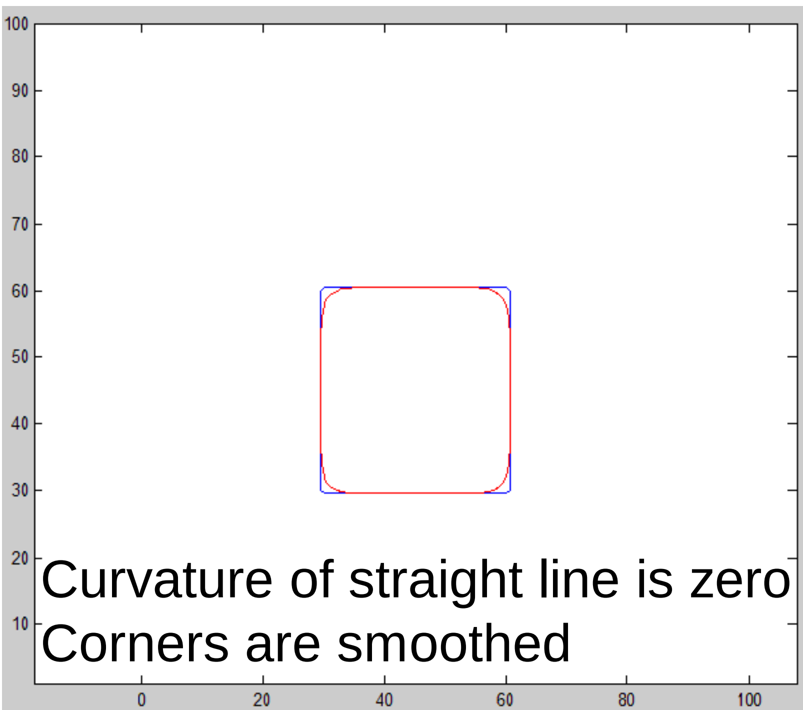
$$\frac{\partial \phi}{\partial t} + a\left|\nabla \phi\right| = 0$$

a > 0

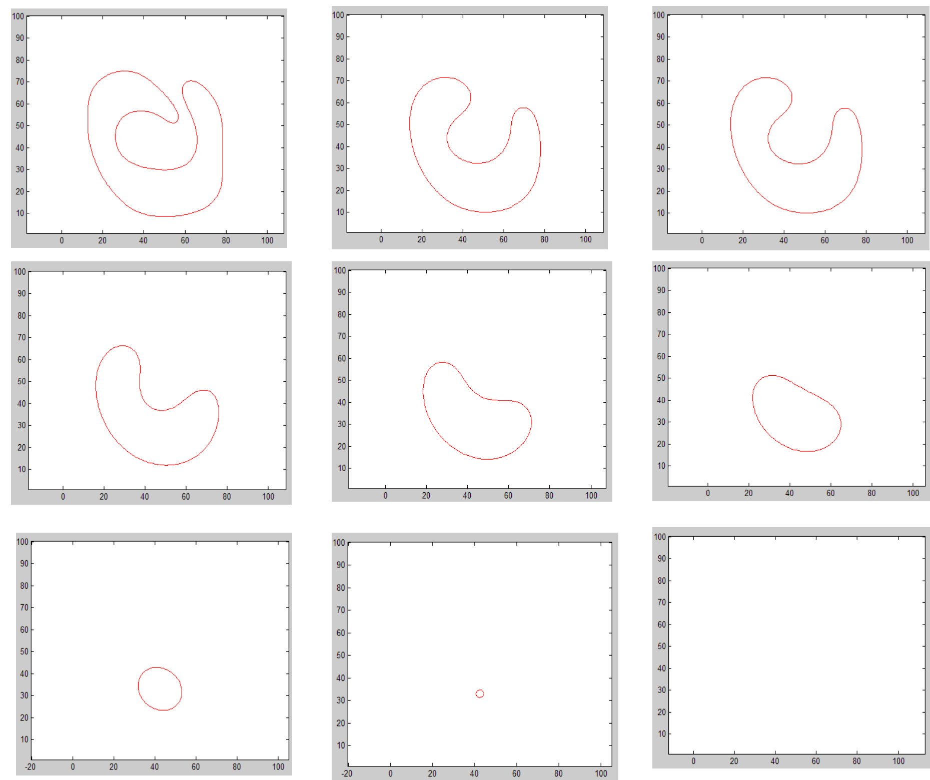
Level Set Motion Examples

- Mean Curvature Flow

$$\frac{\partial \phi}{\partial t} = \kappa \left| \nabla \phi \right|$$

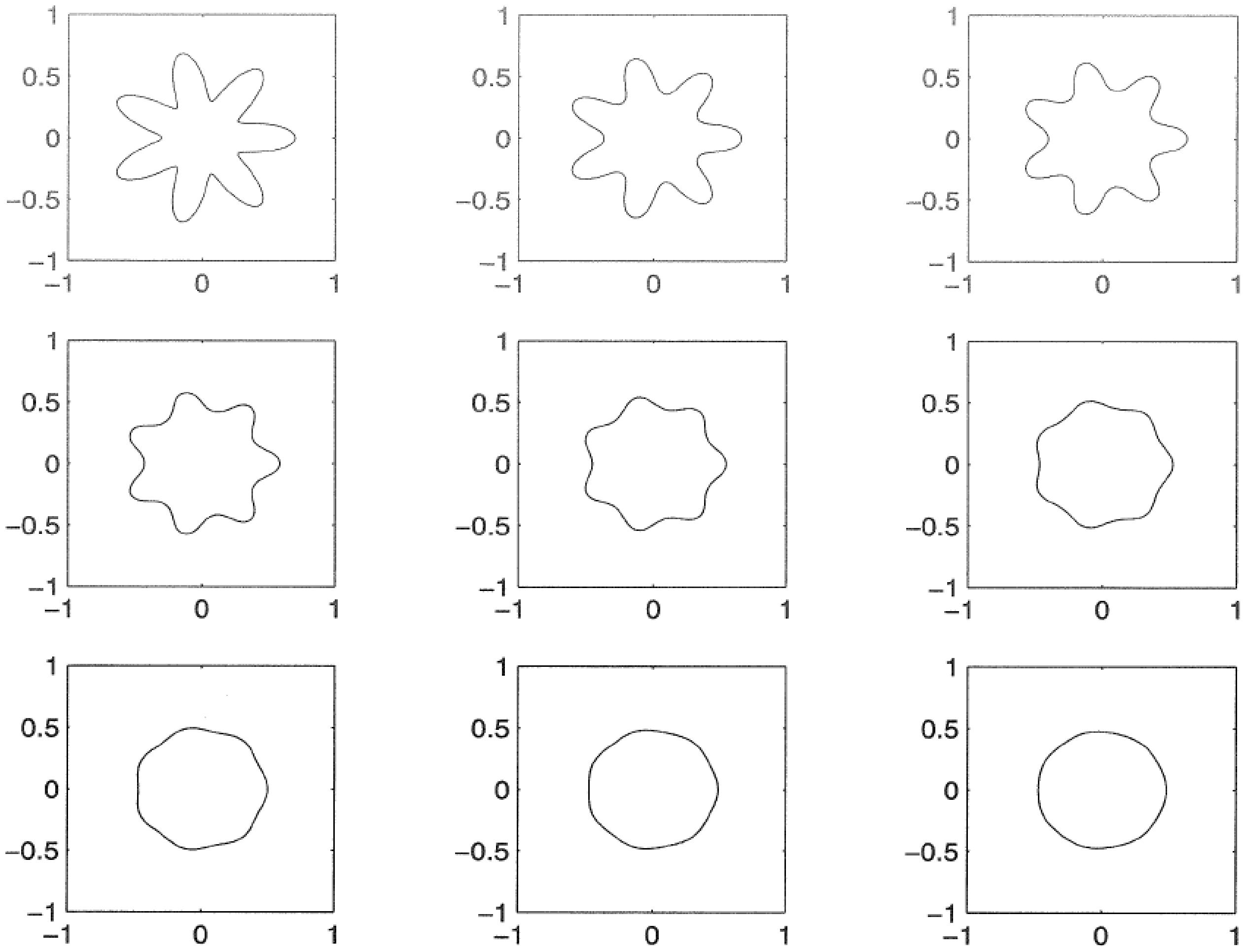


Curvature of straight line is zero
Corners are smoothed



Matlab!

Mean Curvature Flow

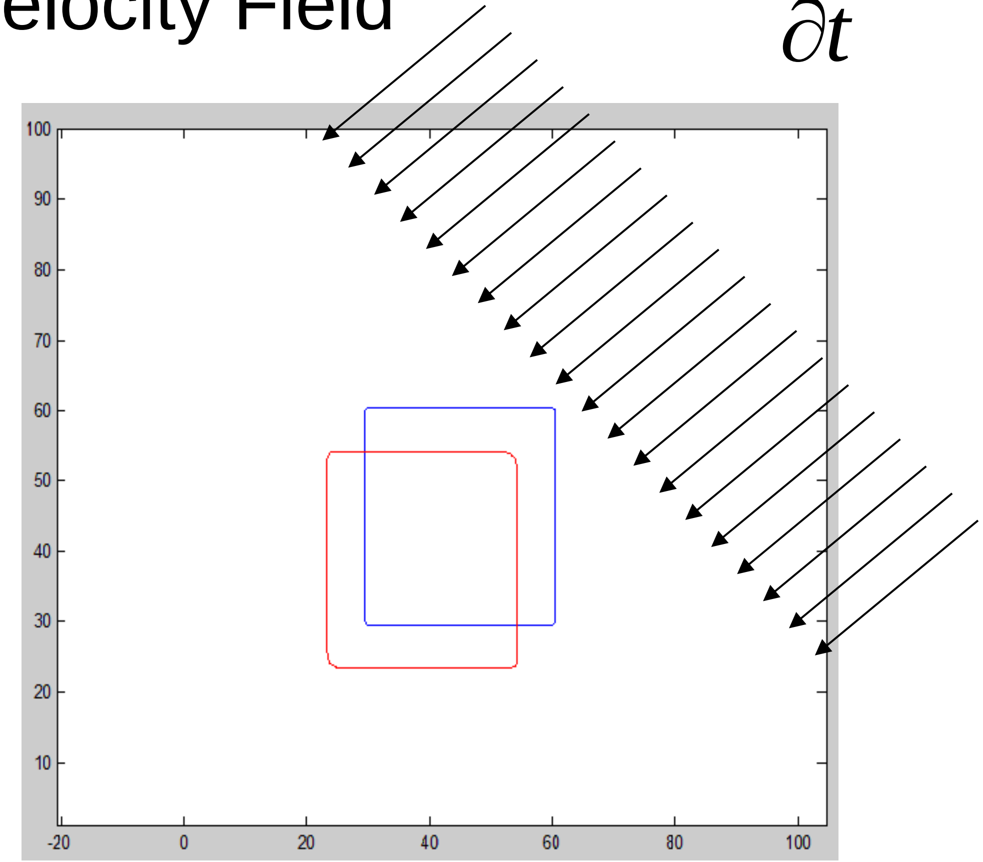


$$\frac{\partial \phi}{\partial t} = \kappa \left| \nabla \phi \right|$$

Level Set Motion Examples

$$\frac{\partial \phi}{\partial t} = -\vec{V} \cdot \nabla \phi$$

- External Velocity Field



Matlab!

Full-Grown Level Set Equation

$$\boxed{\frac{\partial \phi}{\partial t} + \vec{V} \cdot \nabla \phi + a \left| \nabla \phi \right| = b \kappa \left| \nabla \phi \right|}$$

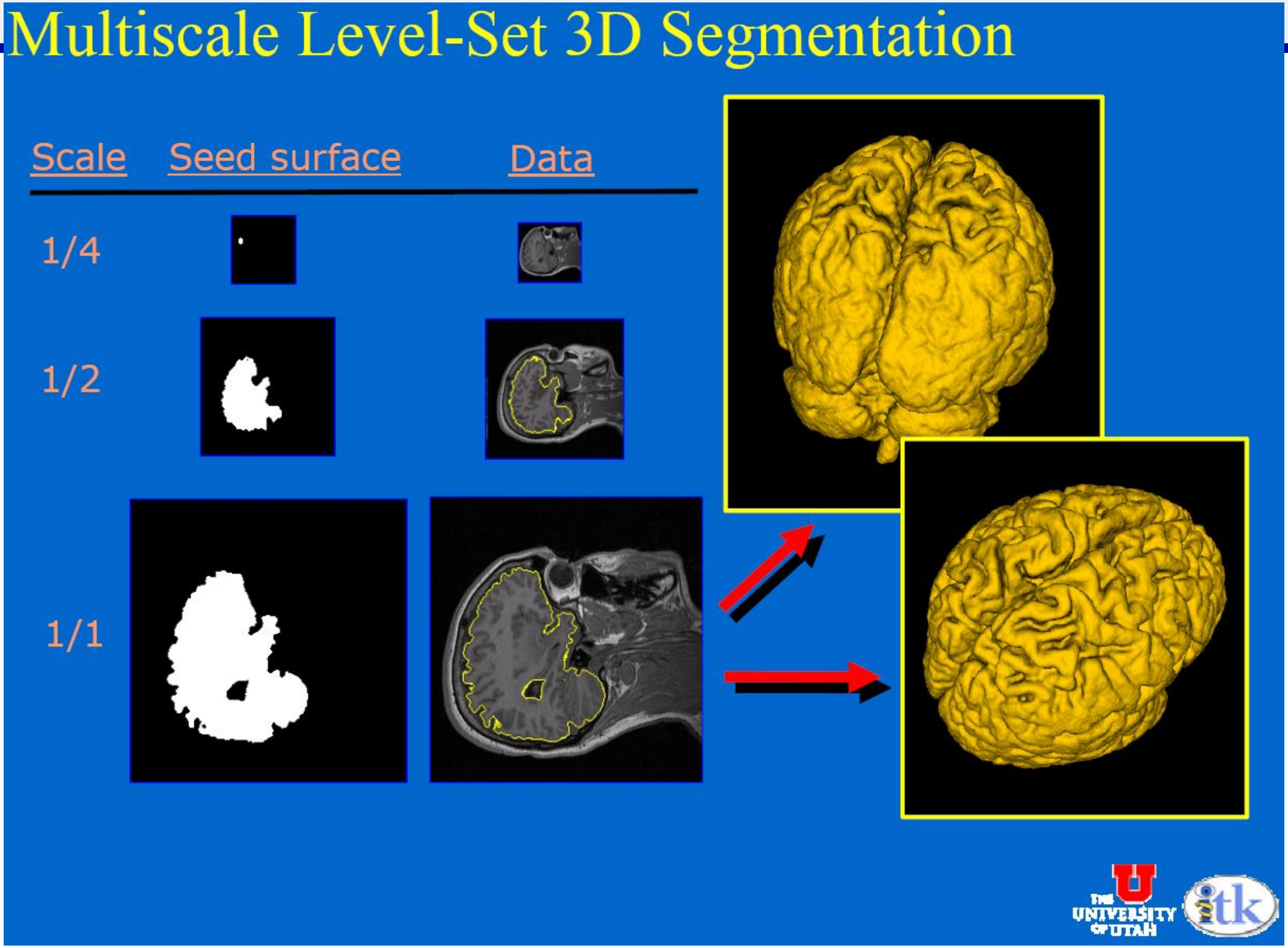
External Vector-field

Normal Flow

Mean Curvature Flow

- Numerical Implementation
  - Depending on terms, parabolic/hyperbolic PDE
  - Discretization in time and space critical!
- Matlab Toolbox for download:
  - http://barissumengen.com/level_set_methods/index.html

Level-Set Segmentation Example



Back to GAC Solution

- May be implemented using Level Sets

$$\frac{\partial C}{\partial t} = \left(g\kappa - \nabla g \cdot \vec{N}\right)\vec{N} \quad \Longleftrightarrow \quad \frac{\partial \phi}{\partial t} = \left|\nabla \phi\right| g\kappa + \nabla g \cdot \nabla \phi$$

Euler-Lagrange eq.

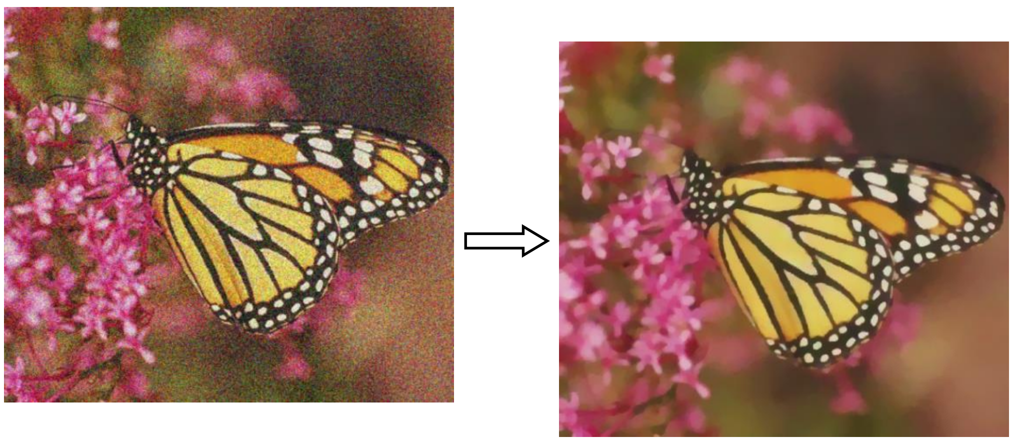
Mean Curvature Flow

Flow according to external gradient field

- Problem with Level Sets:
  - Gradient descent in level set framework usually converges to local minimum
  - It would be great if we could formulate this problem as a convex functional, i.e. we can locate a global minimum!

GAC Solution – Weighted TV

- Looking for a convex functional for GAC, finally our Total Variation framework comes back into play!



Here is a link between Denoising and Segmentation!

GAC Solution – Weighted TV

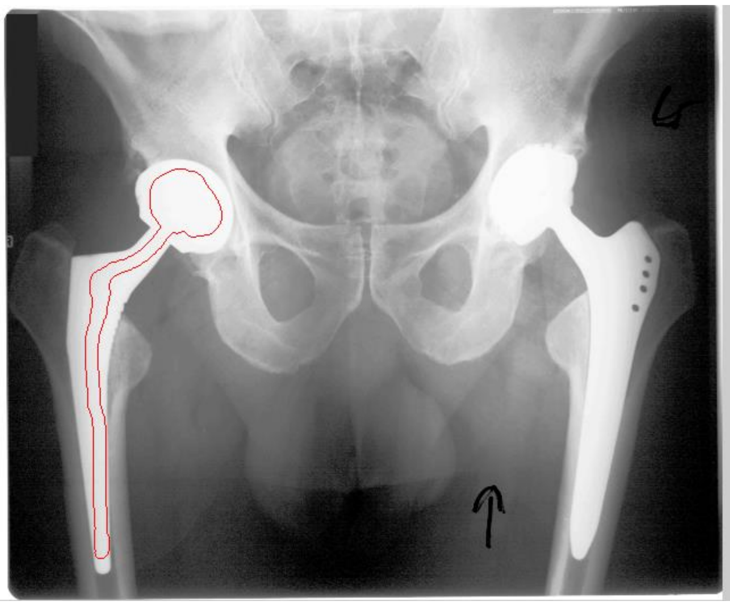
- Bresson 2007: Weighted Total Variation

$$E_{wTV} = \int_{\Omega} g(x)|\nabla u| dx \qquad TV(u) = \int_{\Omega} |\nabla u| dx = \int_{\Omega} \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2} dx$$

- In binary case $(u \in \{0,1\})$ equals the GAC energy!
- If u is allowed to vary continuously between [0,1]:
  - Energy is convex, so we can find a global optimum!
- Unfortunately: C = 0 is always the globally optimal solution -> additional constraints necessary!
- We need to threshold u for binary segmentation

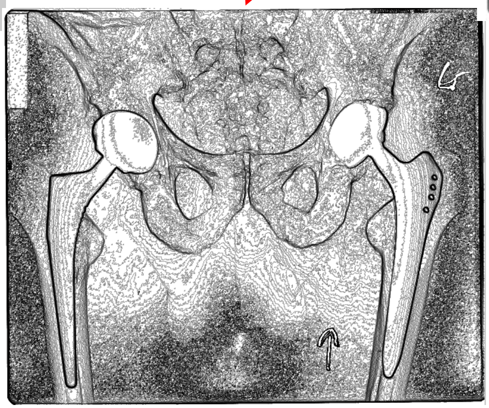
Geodesic Active Contours

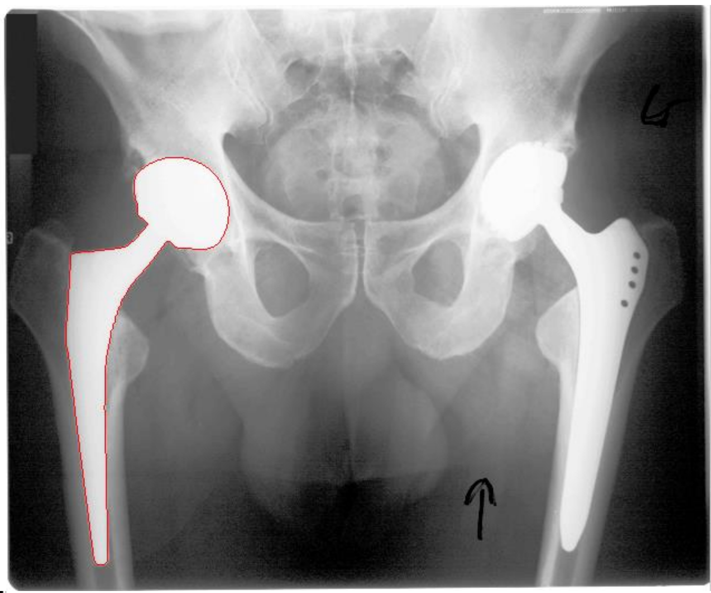
- Snake Model introduced by Caselles et al. (1997)



Initial Contour

Curve Evolution Using Gradient





Final Contour

Matlab!

GAC Solution – Weighted TV

- Specify Constraints:
- Use weighted TV with spatially varying data fidelity term:

$$\min_u \left\{ \int_\Omega g(x) |\nabla u| dx + \lambda \int_\Omega (u \cdot f) dx \right\}$$

$$f = \begin{cases} -\infty & \text{force foreground} \\ - & \text{likely foreground} \\ 0 & \text{undetermined} \\ + & \text{likely background} \\ +\infty & \text{force background} \end{cases}$$

- f user-provided or derived from prior color distribution (histogram/Gauss model)

- Minimization of this model is very similar to the minimization of TV-L2 and TV-L1 Denoising!

Solve Weighted TV Segmentation

$$\min_{u}\left\{\int_{\Omega} g(x)|\nabla u|dx + \lambda\int_{\Omega}(u \cdot f)dx\right\}$$

- Corresponding Euler-Lagrange equation:

$$-\nabla\cdot\left(g(x)\frac{\nabla u}{|\nabla u|}\right)+\lambda f=0$$

Again: Problem with Derivative!

- -> Primal-Dual Formulation:

$$\min_{u}\max_{\|\mathbf{p}\|\le g}\left\{\int_{\Omega}\mathbf{p}\cdot\nabla u dx + \lambda\int_{\Omega}(u \cdot f)dx\right\}$$

Reprojection to hyper-sphere of radius g!

Solve Weighted TV Segmentation

$$\min_u \max_{\|\mathbf{p}\|\leq g} \left\{ \int_\Omega \mathbf{p} \cdot \nabla u dx + \lambda \int_\Omega (u \cdot f) dx \right\}$$

Optimization problem in 2 variables

- Alternating optimization in u,**p**

1. $$\frac{\partial}{\partial u} \left\{ -\int_\Omega u \nabla \cdot \mathbf{p} dx + \lambda \int_\Omega (u \cdot f) dx \right\} = -\nabla \cdot \mathbf{p} + \lambda f$$

$$u^{n+1} = u^n + \tau_P (\nabla \cdot \mathbf{p} - \lambda f)$$

Additionally make sure that $u \in [0,1]$

$$\boxed{u^{n+1} = \min(1, \max(0, u^n + \tau_P (\nabla \cdot \mathbf{p} - \lambda f)))}$$

Gradient Descent

Solve Weighted TV Segmentation

2. $$\frac{\partial}{\partial \mathbf{p}}\left\{\int_{\Omega} \mathbf{p} \cdot \nabla u dx + \lambda \int_{\Omega} (u \cdot f) dx\right\} = \nabla u$$

||p||≤g

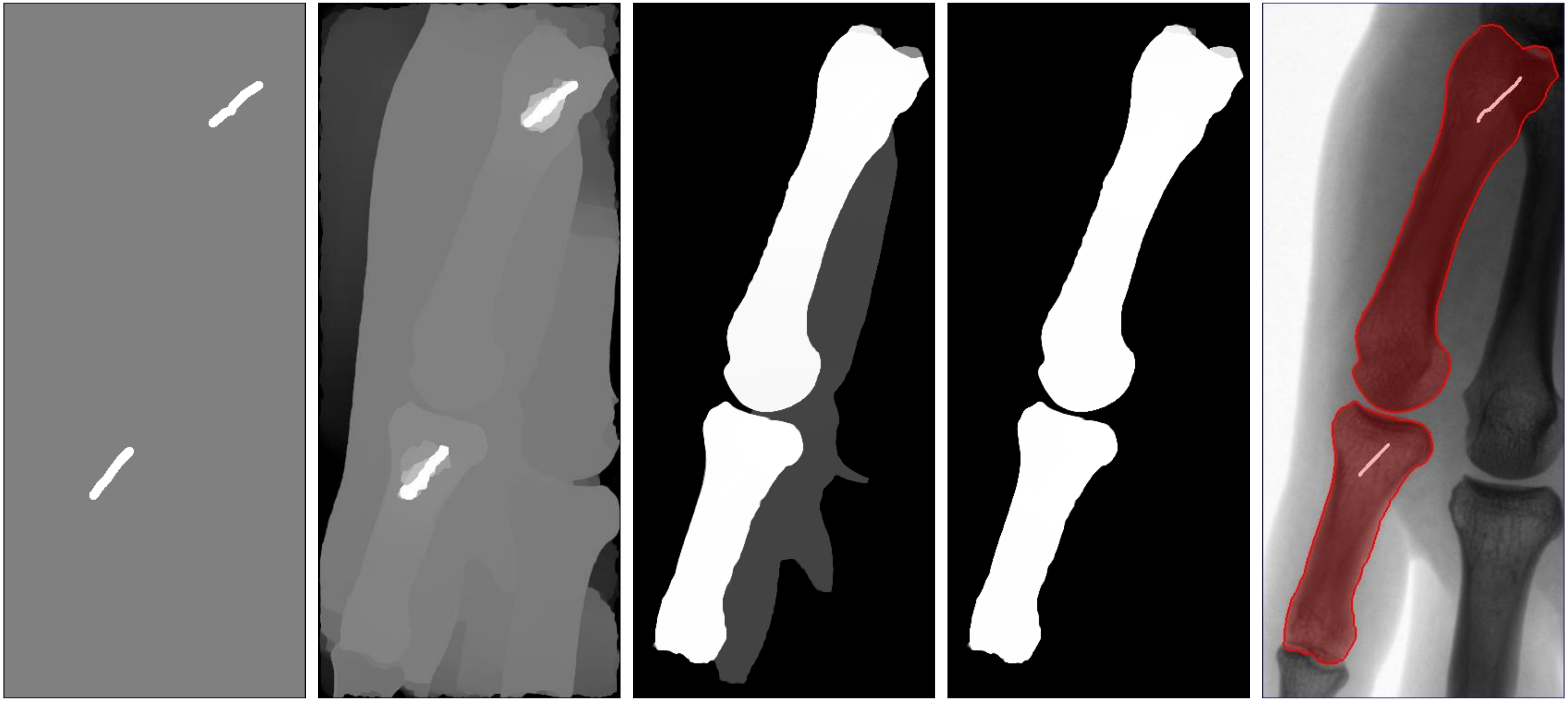
$$\tilde{\mathbf{p}}^{n+1} = \mathbf{p}^{n} + \tau_{D} \nabla u$$

$$\mathbf{p}^{n+1} = \frac{\tilde{\mathbf{p}}^{n+1}}{\max\left(1, \frac{\left\|\tilde{\mathbf{p}}^{n+1}\right\|}{g}\right)}$$

Gradient Ascent

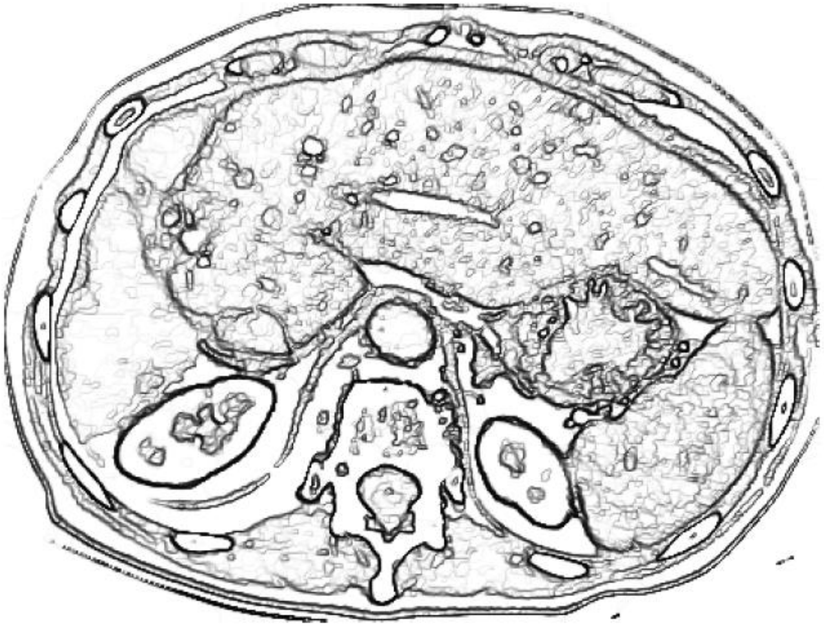
- Alternated updates over a number of iterations
- Discretization -> see ROF Primal-Dual

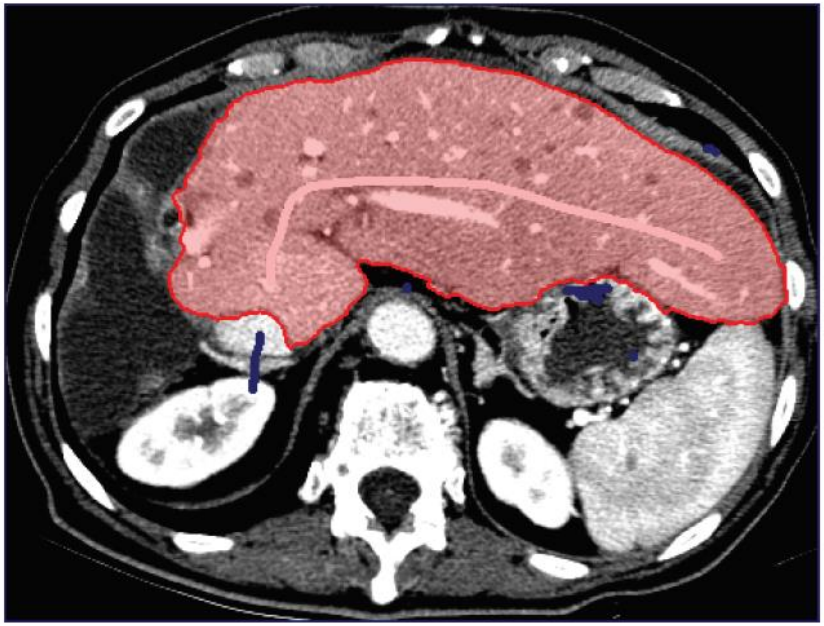
Weighted TV: Defining Constraints



iterations

Interactive Segmentation Using Weighted Total Variation





ICG Tool Available at: http://www.gpu4vision.org
Tools like Photoshop, Gimp have similar algorithms!

Interactive Segmentation Using Weighted Total Variation