Fast Methods to Find Optimal Shapes in Images

by
Günay Doğan

MSCD / ITL

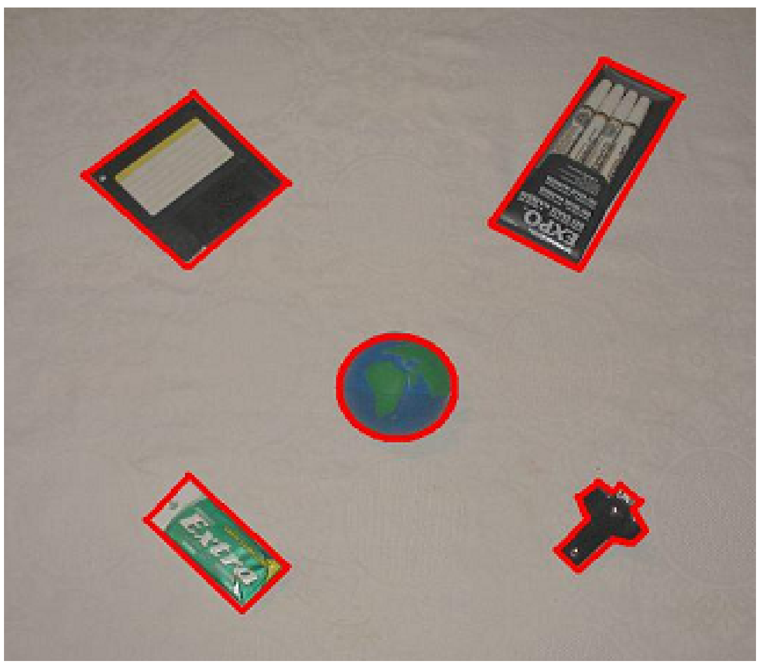
joint with
Pedro Morin, Ricardo H. Nochetto

Image Segmentation

Goal: To capture objects & boundaries in given 2d / 3d images

Image Segmentation

Goal: To capture objects & boundaries in given 2d / 3d images



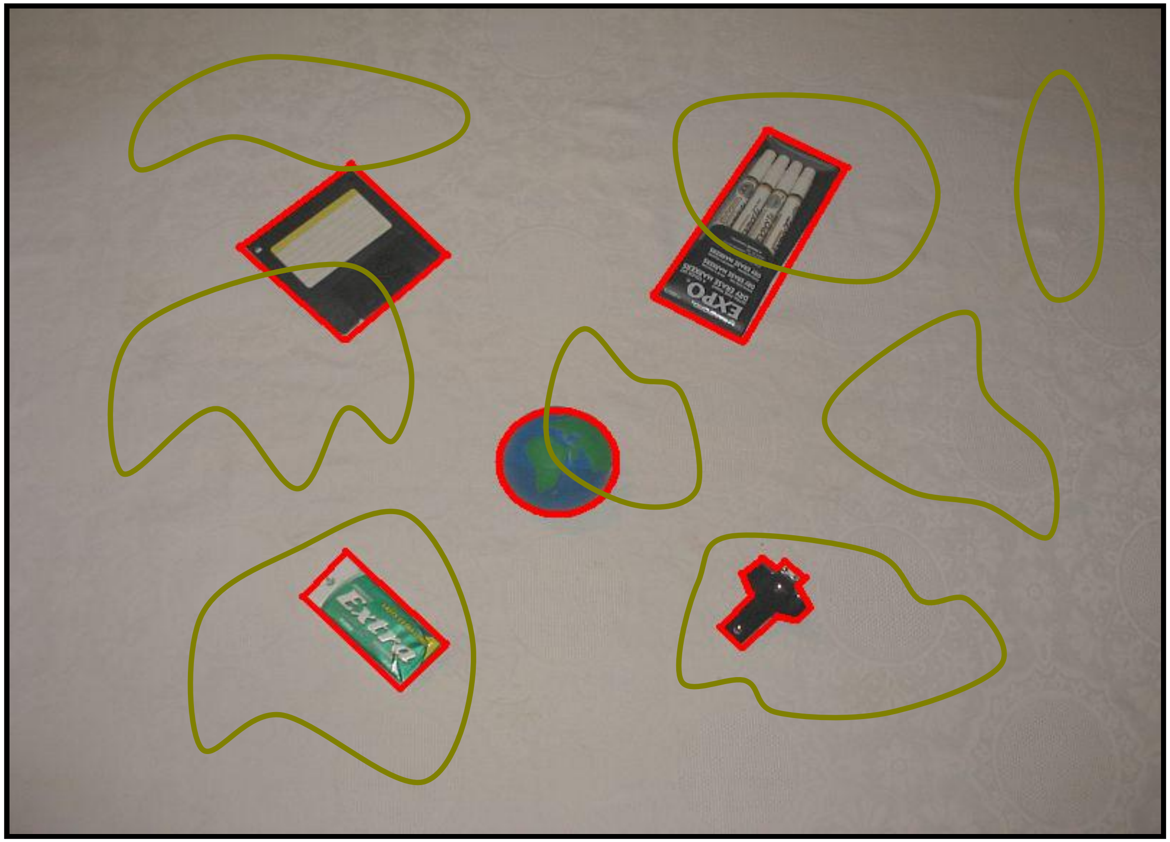
Segmentation Related

- Medical imaging
- Motion detection & tracking
- 3d scene reconstruction
- Image understanding
- Scientific visualization
- Surgery planning

… … …

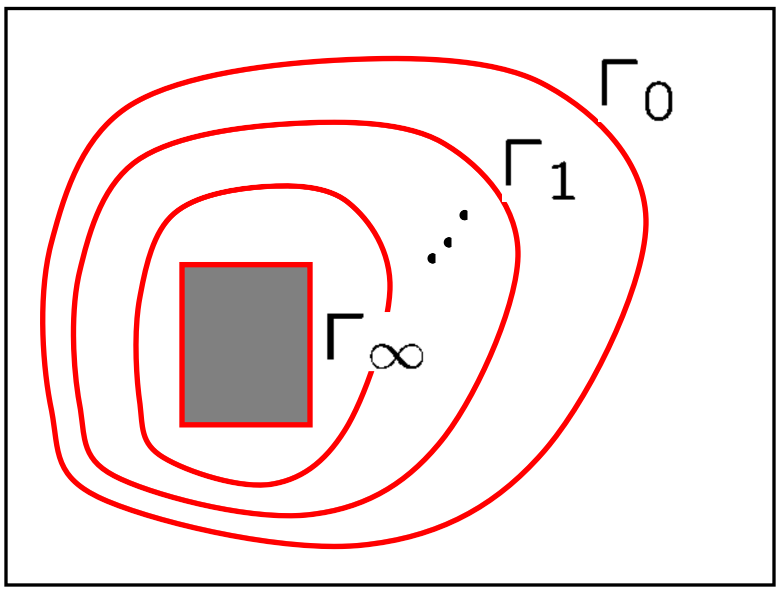
Finding the Right Curves

Q: How do we decide which are the right curves?



Shape Optimization Approach

Define shape energy $J(\Gamma)$ for given shape Γ



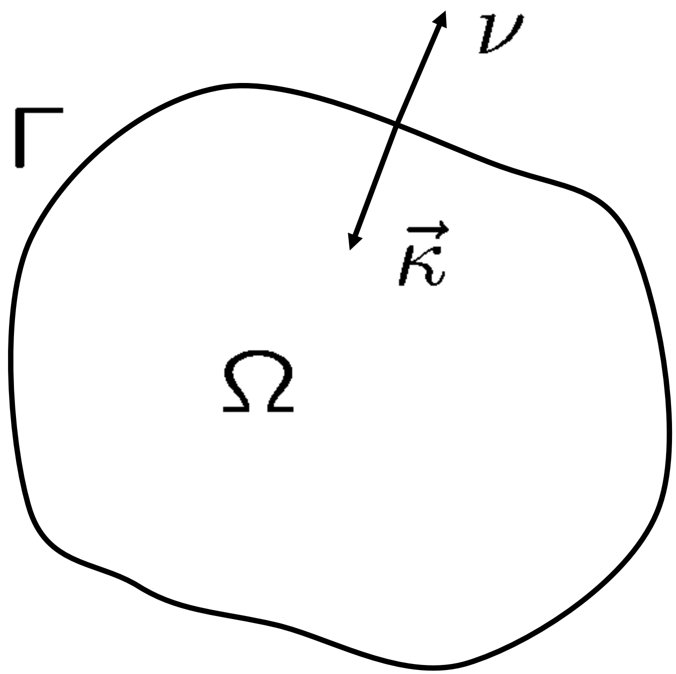
Compute a sequence

$\Gamma_0, \Gamma_1, \ldots, \Gamma_\infty$

such that

$J(\Gamma_0) \geqslant \ldots \geqslant J(\Gamma_\infty)$

Differential Geometry



Γ : boundary

Ω : domain

ν : outer unit normal

$$\kappa = \kappa_1 + \ldots + \kappa_{d-1}$$

$$\vec{\kappa} = \kappa\nu$$

Differential Geometry

For smooth $f, \vec{W}$ extended to a neighborhood of Γ

$$\partial_\nu f = \nabla f \cdot \nu$$

(normal derivative)

$$\nabla_\Gamma f = \nabla f - \nabla f \cdot \nu\,\nu$$

(tangential gradient)

(tangential divergence)

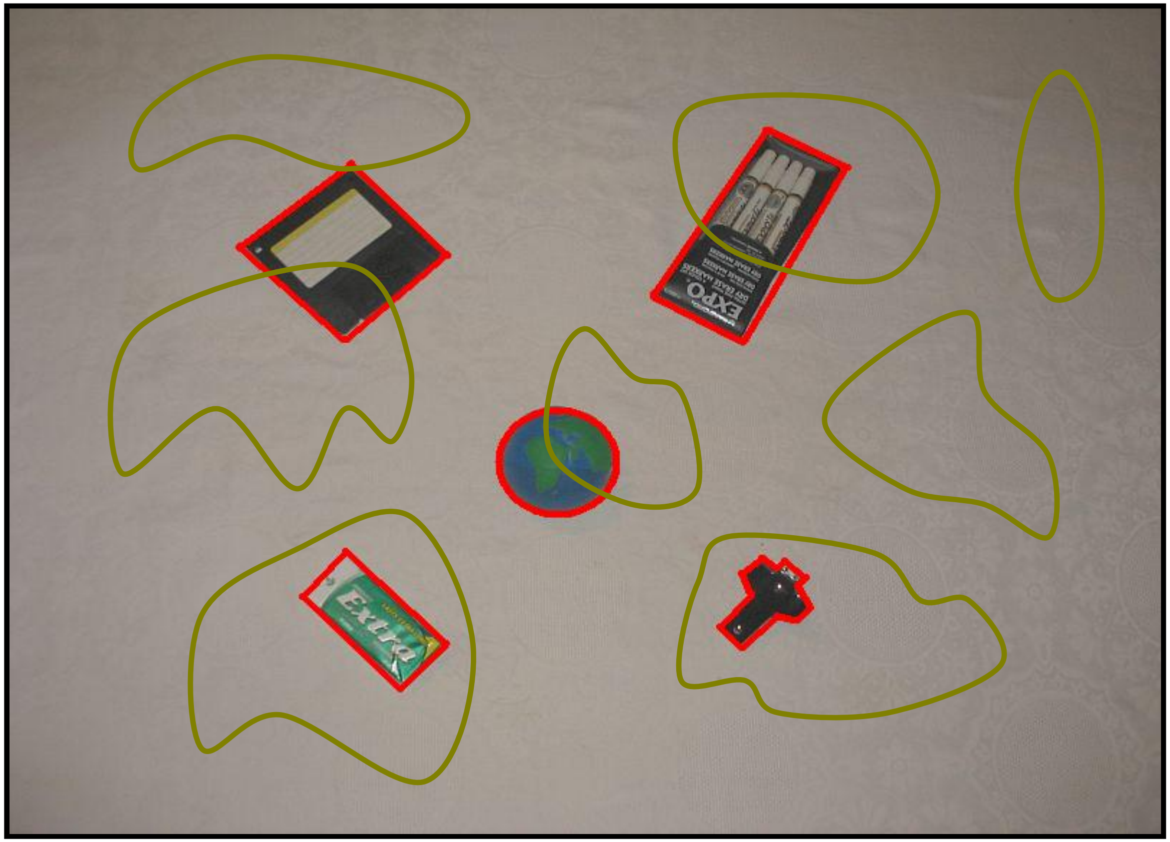
$$\mathrm{div}_\Gamma \vec{W} = \mathrm{div}\vec{W} - \nu \cdot D\vec{W} \cdot \nu$$

$$\Delta_\Gamma f = \Delta f - \nu \cdot D^2 f \cdot \nu - \partial_\nu f \kappa$$

(Laplace-Beltrami)

Finding the Right Curves

Q: What are the right energies to impose on curves?



Shape Energies for Segmentation

Kass, Witkin, Terzopoulos, 88
Fue, Leclerc, 90
Caselles, Kimmel, Sapiro, 95
Caselles, Kimmel, Sapiro, Sbert, 97
Paragios, Deriche, 00
Chan, Vese, 01
Tsai, Yezzi, Willsky, 01
Aubert, Barlaud, Faugeras, Jehan-Besson, 03
Kimmel, Bruckstein, 03
Desolneaux, Moisan, Morel, 03
Hintermueller, Ring, 03

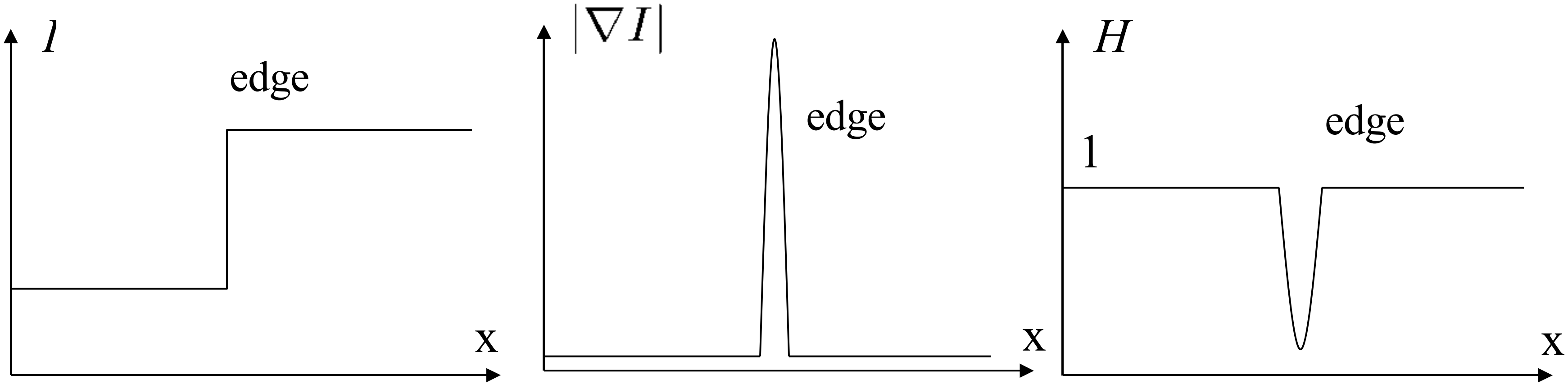
and many more ...

$$J(\Gamma) = \int_\Gamma \psi(x, \Gamma)\, dS$$

$$J(\Omega) = \int_\Omega \varphi(x, \Omega)\, dx$$

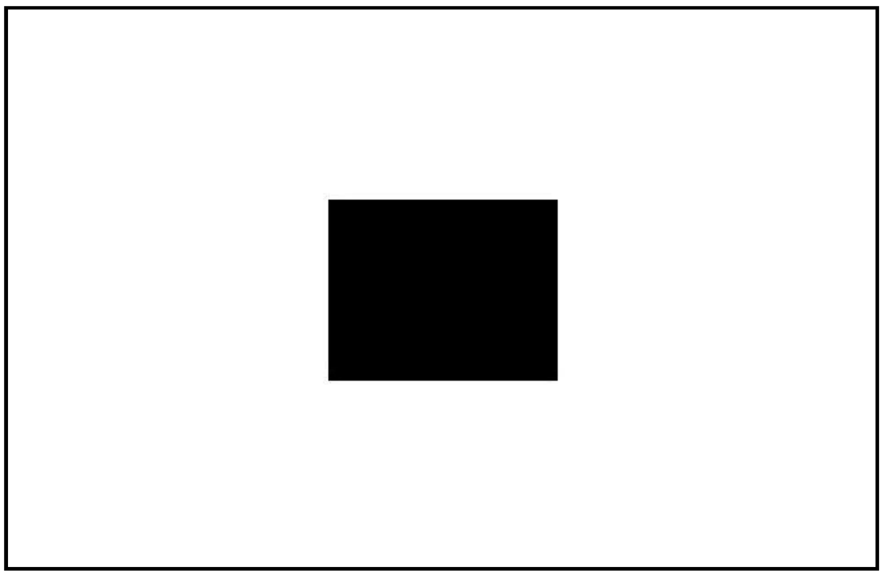
Edge Indicator Function

Edge indicator function: $$H(x) = \frac{1}{1 + \frac{|\nabla I(x)|^2}{\lambda^2}}$$

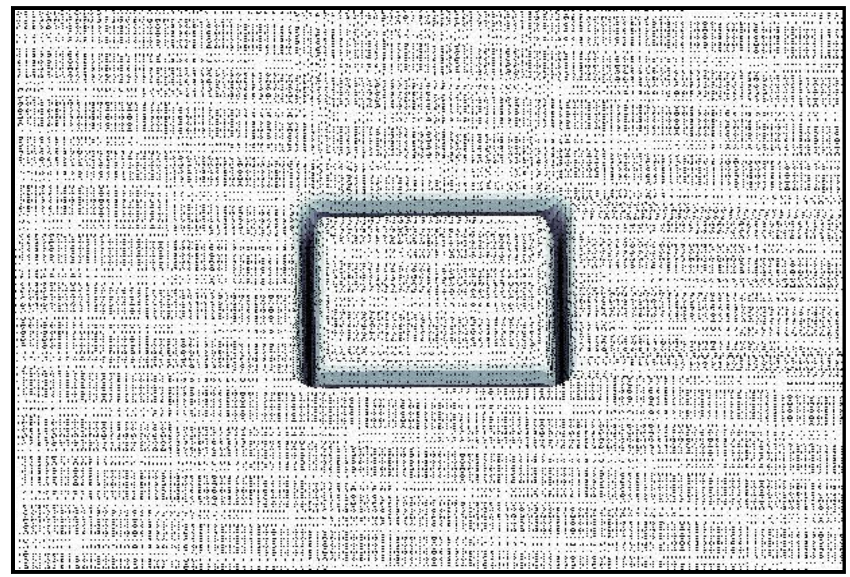


Edge Indicator Function

I(x)



H(x)



Geodesic Active Contours

Generic form of geodesic active contour / surface energy

$$J(\Gamma) = \int_\Gamma H(x)\, dS + \gamma \int_\Omega H(x) dx,$$

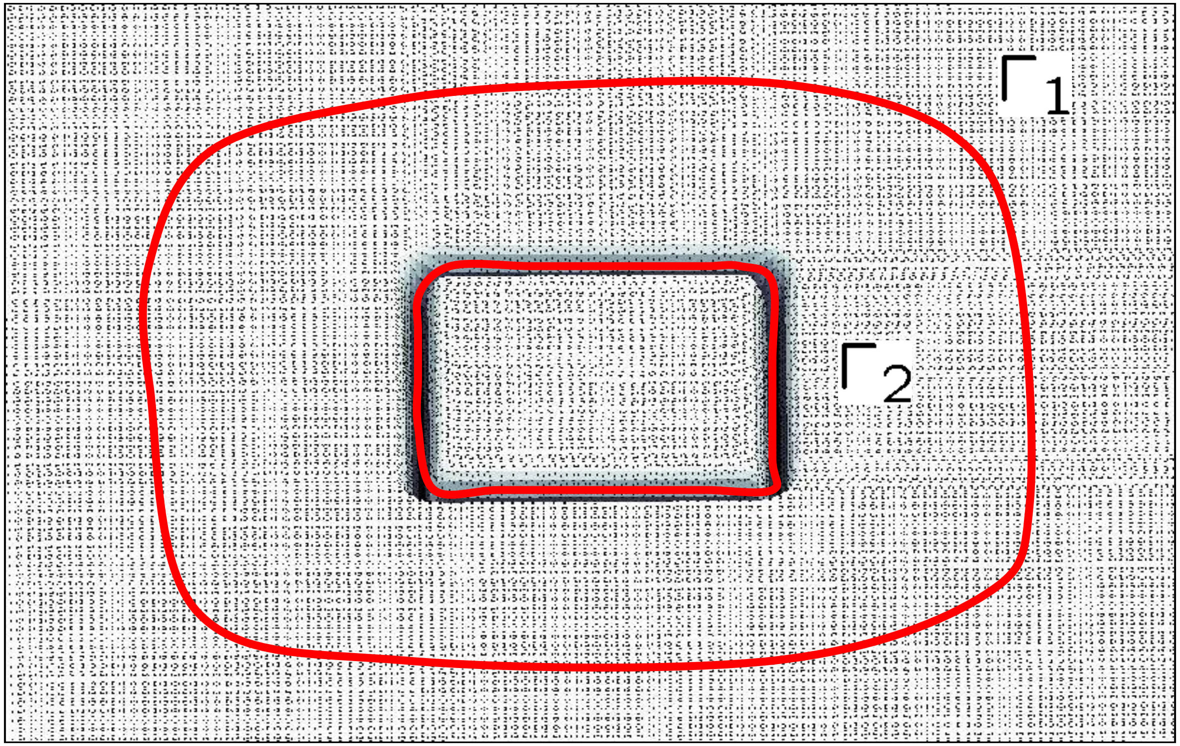
Minimize weighted area of surface Γ and enclosed volume.

2nd term helps with concavities, brings extra push.

(Caselles et al., 95; Caselles et al. 97)

Geodesic Active Contours

Computing $J(\Gamma) = \int_{\Gamma} H(x) dS$



$J(\Gamma_1)$ large

$J(\Gamma_2)$ small (local min)

The Mumford-Shah Model

Given ***I(x)***, find discontinuities ***K*** and p.w. smooth approx. ***u***

$$\min_{u,K} \left\{ \frac{1}{2} \int_{\mathcal{U}} |u - I|^2 dx + \frac{\mu}{2} \int_{\mathcal{U}-K} |\nabla u|^2 dx + \gamma \int_K dS \right\}$$

The Mumford-Shah Model

Given ***I(x)***, find discontinuities ***K*** and p.w. smooth approx. ***u***

$$\min_{u,K} \left\{ \frac{1}{2} \int_{\mathcal{U}} |u - I|^2 dx + \frac{\mu}{2} \int_{\mathcal{U}-K} |\nabla u|^2 dx + \gamma \int_K dS \right\}$$

Not practical in this form.

Two approaches:

- Ambrosio-Tortorelli approach: modified functional with diffuse discontinuities
- Chan-Vese approach: restrict ***K*** to closed curves, use curve evolution

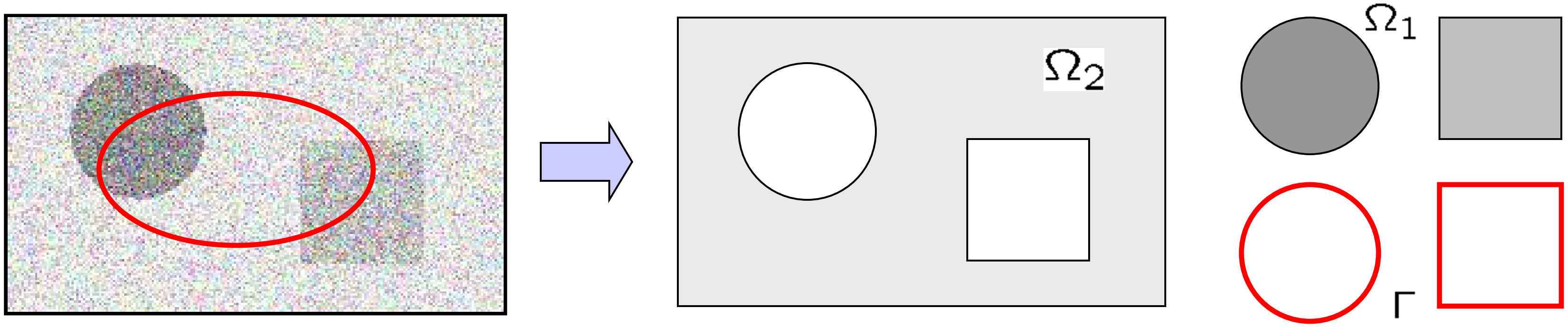
Chan-Vese Approach

$$J(\Gamma) = \sum_{i=1}^{2} \frac{1}{2} \left(\int_{\Omega_i} (u_i - I)^2 + \mu |\nabla u_i|^2 \right) dx + \gamma \int_{\Gamma} dS$$

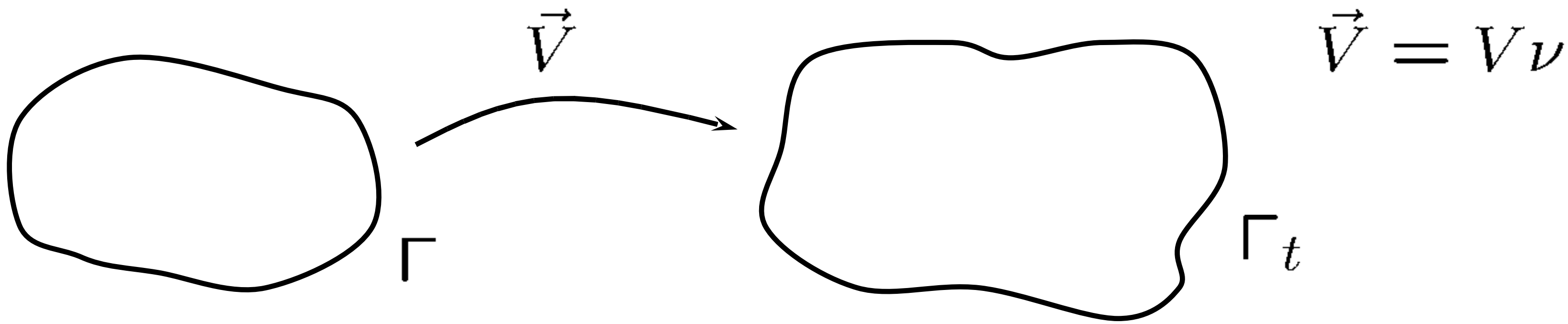
optimality cond. w.r.t u_i

$$\begin{cases} -\mu \Delta u_i + u_i = I & \text{in} \quad \Omega_i \\ \partial_{\nu_i} u_i = 0 & \text{on} \quad \partial \Omega_i \end{cases}$$

(Chan & Vese, 01; Tsai, Yezzi & Willsky, 01)



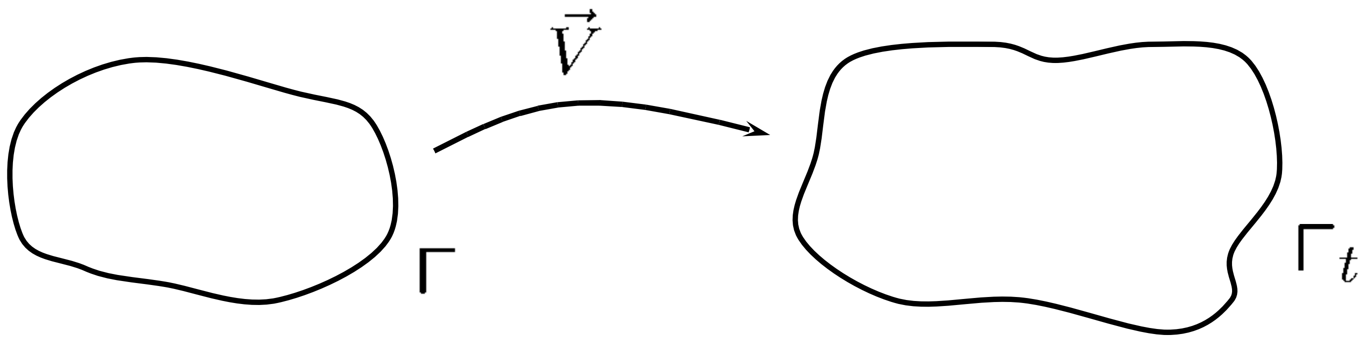
Shape Derivatives



Shape derivative of $J(\Gamma)$ in direction V:

$$dJ(\Gamma; V) = \lim_{t \to 0} \frac{J(\Gamma_t) - J(\Gamma_0)}{t}$$

Shape Derivatives



$$\vec{V} = V\nu$$

Shape derivative of ***J(Γ)*** in direction ***V***:

$$dJ(\Gamma; V) = \lim_{t \to 0} \frac{J(\Gamma_t) - J(\Gamma_0)}{t}$$

Second shape derivative w.r.t. ***V, W***:

$$d^2 J(\Gamma; V, W) = d(dJ(\Gamma; V))(\Gamma; W)$$

Shape Gradient

The structure of the shape derivative for $J(\Gamma)$

$$dJ(\Gamma;V) = \int_\Gamma G V \, dS$$

where G is the shape gradient.

For most cases

$$G = g(x,\Gamma)\kappa + f(x,\Gamma)$$

Shape Gradient

The structure of the shape derivative for $J(\Gamma)$

$$dJ(\Gamma; V) = \int_\Gamma G V \, dS$$

where G is the shape gradient.

For most cases

$$G = g(x, \Gamma)\kappa + f(x, \Gamma)$$

Choose $V = -G \;\Rightarrow\; dJ(\Gamma; V) = -\int_\Gamma G^2 dS \leqslant 0$

Geodesic Active Contours

$$J(\Gamma) = \int_\Gamma H(x)\, dS + \gamma \int_\Omega H(x) dx, \qquad \gamma > 0$$

First shape derivative

$$dJ(\Gamma; V) = \int_\Gamma (H(x)\kappa + \gamma H(x) + \partial_\nu H(x)) V\, dS$$

$$\Rightarrow G = g\kappa + f \qquad \text{with} \qquad g = H(x)$$

$$f = \gamma H(x) + \partial_\nu H(x)$$

Geodesic Active Contours

$$J(\Gamma) = \int_\Gamma H(x)\, dS + \gamma \int_\Omega H(x) dx, \qquad \gamma > 0$$

First shape derivative

$$dJ(\Gamma; V) = \int_\Gamma (H(x)\kappa + \gamma H(x) + \partial_\nu H(x)) V\, dS$$

$$\Rightarrow G = g\kappa + f \qquad \text{with} \qquad g = H(x)$$

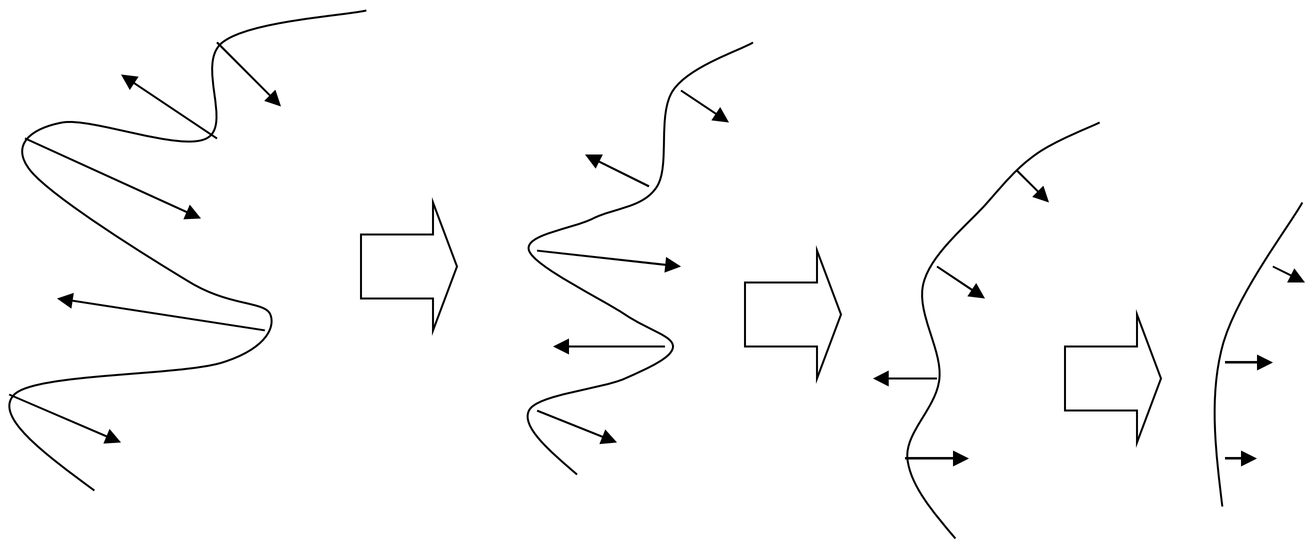
$$f = \gamma H(x) + \partial_\nu H(x)$$

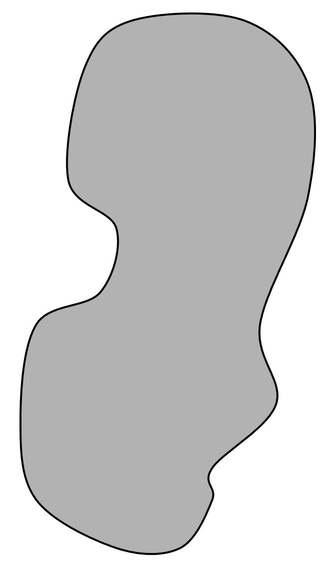
Energy-decreasing velocity

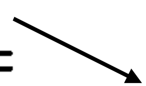
$$V = -G = -((\kappa + \gamma) H(x) + \partial_\nu H(x))$$

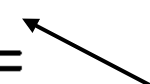
Geodesic Active Contours

Behavior of $\vec{V} = V\nu, \quad V = -H\kappa$



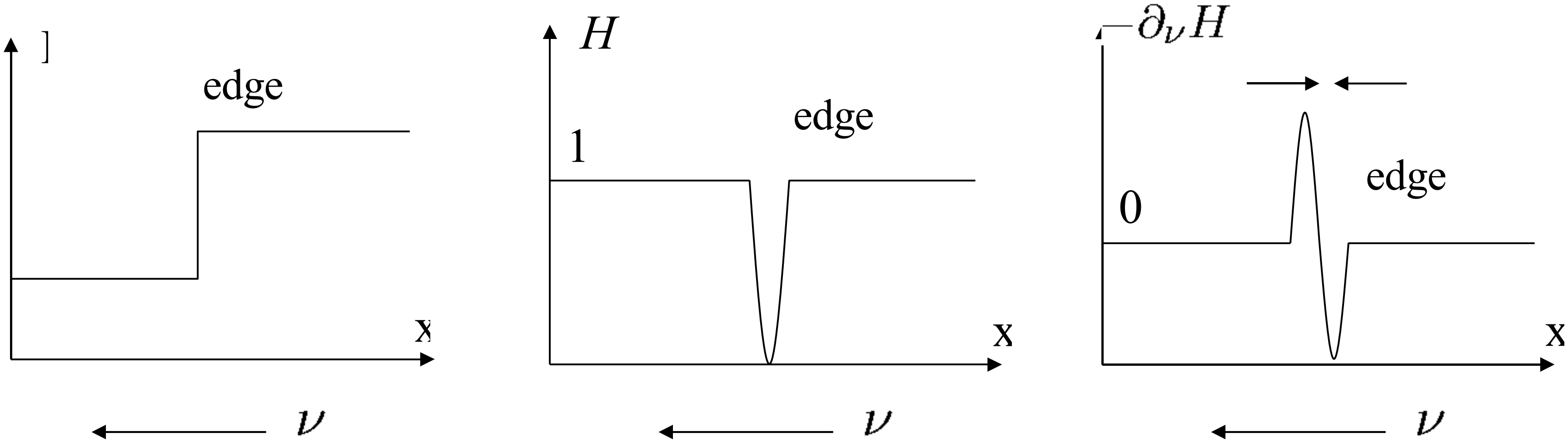


$\kappa < 0 \Rightarrow \kappa\nu =$ 

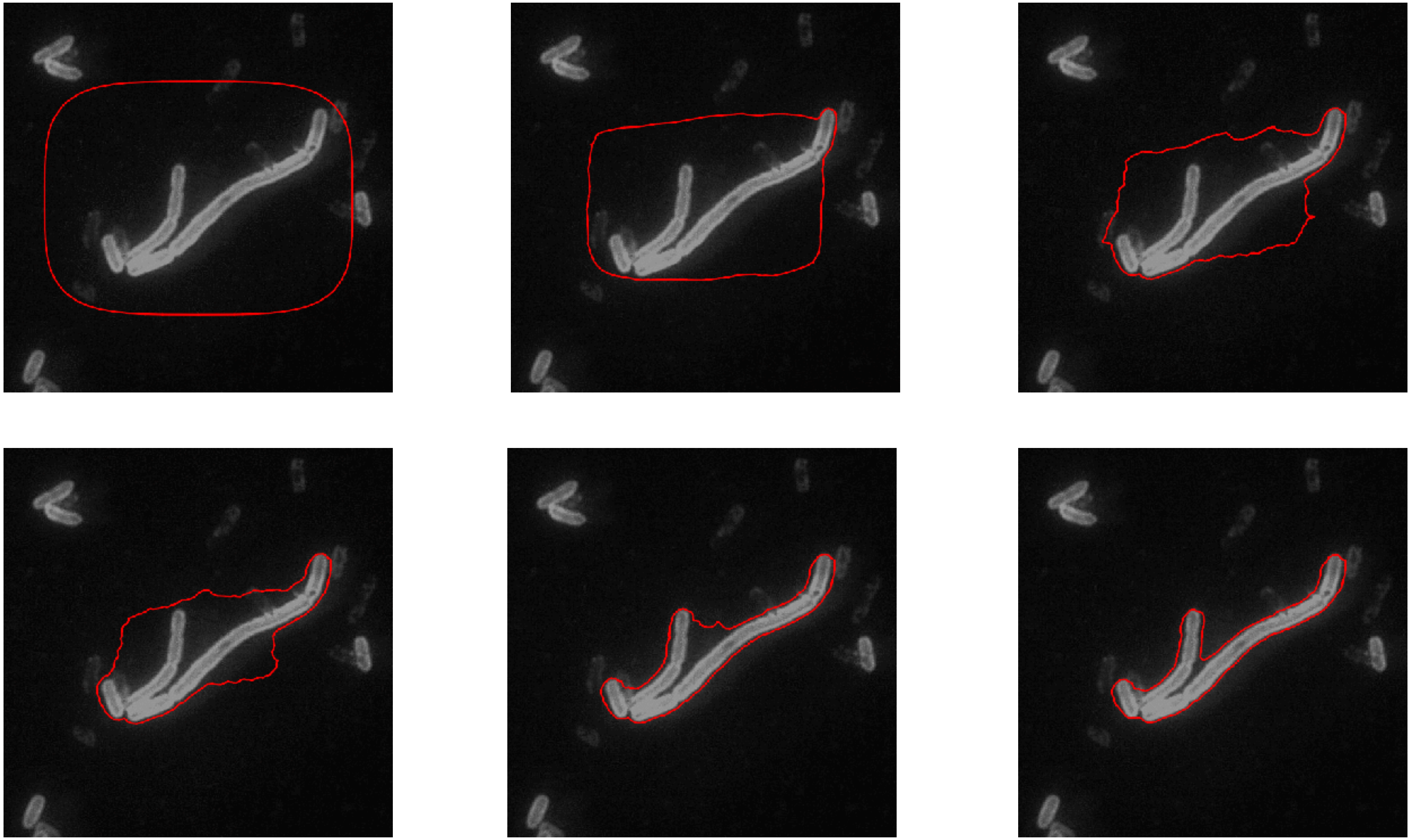
$\kappa > 0 \Rightarrow \kappa\nu =$ 

Geodesic Active Contours

Behavior of $V = -\partial_\nu H$



Example: Bacteria Image



Other Descent Directions

Take a scalar product $b(\phi, \psi)$ on $B(\Gamma)$

- continuous $b(\phi, \psi) \leq c_1 \|\phi\| \|\psi\|, \quad c_1 \geq 0$
- coercive $b(\phi, \phi) \geq c_2 \|\phi\|^2, \quad c_2 \geq 0$

And solve

$$b(V, \phi) = -dJ(\Gamma; \phi), \quad \forall \phi \in B(\Gamma)$$

The solution V satisfies

$$dJ(\Gamma; V) = -b(V, V) \leq -c_2 \|V\|^2 \leq 0$$

Other Descent Directions

Instead of the velocity eqn $V = -(g\kappa + f)$

Use the more general velocity eqn with the scalar product

$$b(V, \phi) = -\langle g\kappa + f, \phi \rangle, \quad \forall \phi \in B(\Gamma)$$

For example, use weighted $H^1(\Gamma)$ scalar product

$$b(V, \phi) = \langle \alpha \nabla_\Gamma V, \nabla_\Gamma \phi \rangle + \langle \beta V, \phi \rangle$$

The general velocity eqn is

$$\langle \alpha \nabla_\Gamma V, \nabla_\Gamma \phi \rangle + \langle \beta V, \phi \rangle = -\langle g\kappa + f, \phi \rangle$$

Geodesic Active Contours

$$J(\Gamma) = \int_\Gamma H(x)\, dS + \gamma \int_\Omega H(x) dx,$$

Second shape derivative

$$d^2 J(\Gamma; V, W) = \int_\Gamma (\alpha \nabla_\Gamma V \cdot \nabla_\Gamma W + \beta V W)\, dS$$

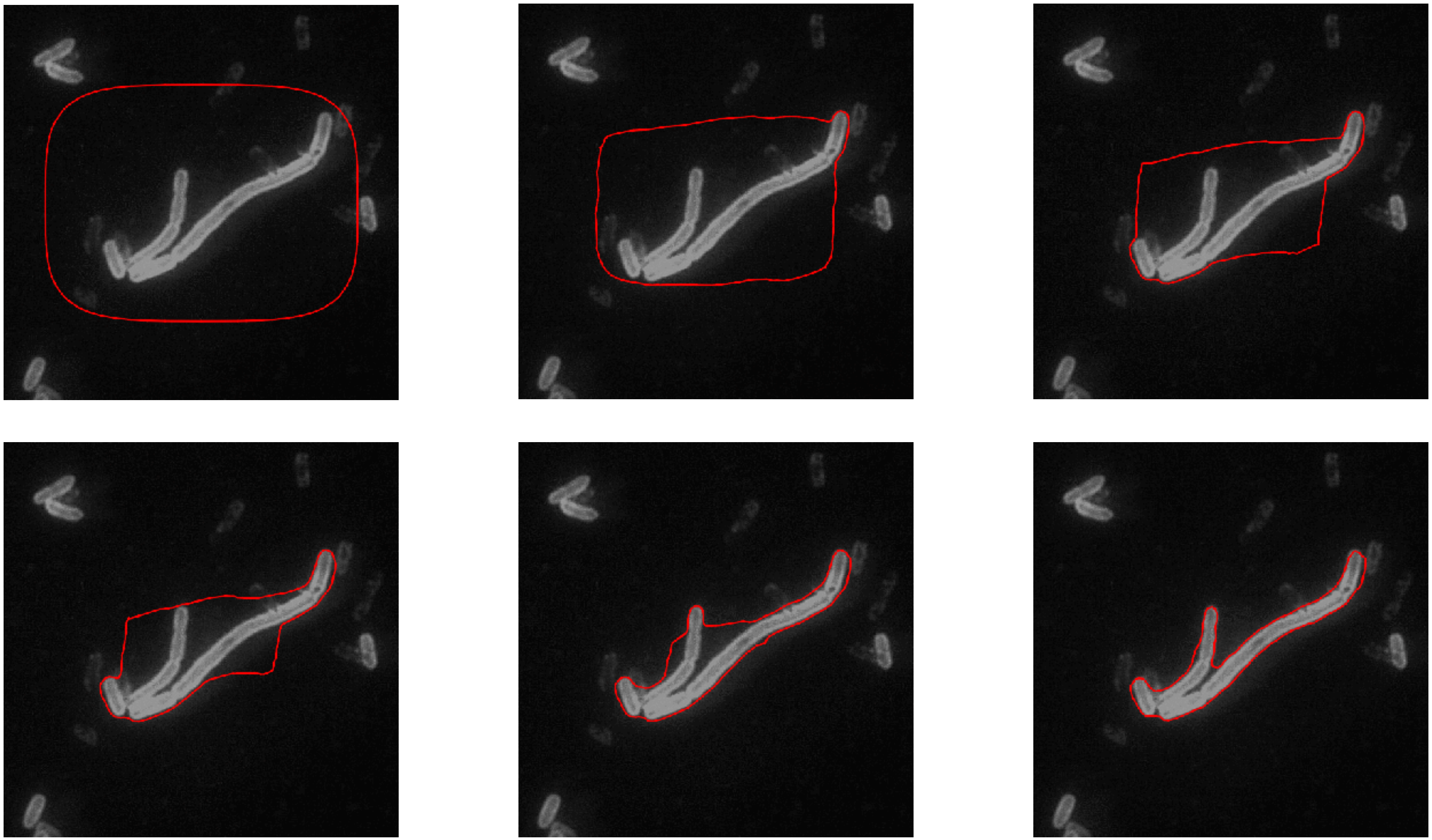
with

$$\alpha = H$$

$$\beta = \partial_{\nu\nu} H + (2\kappa + \gamma)\partial_\nu H + (\kappa^2 - \sum \kappa_i^2 + 2\gamma\kappa) H$$

(Hintermueller & Ring, 03)

Bacteria: H^1 Flow



H^1 flow 276 iters vs L^2 flow 865 iters

Finite Element Method on Surfaces

Solve $\boxed{-\alpha\Delta_\Gamma u + u = f}$ on surface Γ

Finite Element Method on Surfaces

Solve $-\alpha \Delta_\Gamma u + u = f$ on surface Γ

Weak form

$$\alpha \langle \nabla_\Gamma u, \nabla_\Gamma \phi \rangle + \langle u, \phi \rangle = \langle f, \phi \rangle, \quad \forall \phi \in H(\Gamma)$$

Finite Element Method on Surfaces

Solve $$\boxed{-\alpha \Delta_\Gamma u + u = f}$$ on surface Γ

Weak form

$$\boxed{\alpha \langle \nabla_\Gamma u, \nabla_\Gamma \phi \rangle + \langle u, \phi \rangle = \langle f, \phi \rangle, \quad \forall \phi \in H(\Gamma)}$$

Substitute $u = \sum_i u_i \phi_i$

$$\alpha \sum_i u_i \langle \nabla_\Gamma \phi_i, \nabla_\Gamma \phi_j \rangle + \sum_i u_i \langle \phi_i, \phi_j \rangle = \langle f, \phi_j \rangle$$

Finite Element Method on Surfaces

Solve $$-\alpha\Delta_\Gamma u + u = f$$ on surface Γ

Weak form

$$\alpha\langle\nabla_\Gamma u, \nabla_\Gamma \phi\rangle + \langle u, \phi\rangle = \langle f, \phi\rangle, \quad \forall\phi \in H(\Gamma)$$

Substitute $u = \sum_i u_i \phi_i$

$$\alpha\sum_i u_i\langle\nabla_\Gamma \phi_i, \nabla_\Gamma \phi_j\rangle + \sum_i u_i\langle\phi_i, \phi_j\rangle = \langle f, \phi_j\rangle$$

Linear system $$\alpha\mathbf{Au} + \mathbf{Mu} = \mathbf{f}$$

$$\mathbf{A}_{ij} = \langle\nabla_\Gamma \phi_i, \nabla_\Gamma \phi_j\rangle \qquad \mathbf{M}_{ij} = \langle\phi_i, \phi_j\rangle \quad \mathbf{f}_i = \langle f, \phi_i\rangle$$

Computing the Velocity

At each iteration solve the following to get $\vec{V}$

$$\vec{V} = V\nu$$

Obtain the new curve/surface $\vec{X}^{n+1} = \vec{X}^{n} + \tau_n \vec{V}$

Computing the Velocity

At each iteration solve the following to get $\vec{V}$

$$V = -(g\kappa + f)$$

$$\vec{V} = V\nu$$

Obtain the new curve/surface $\vec{X}^{n+1} = \vec{X}^{n} + \tau_n \vec{V}$

Computing the Velocity

At each iteration solve the following to get $\vec{V}$

$$\kappa = \vec{\kappa} \cdot \nu$$

$$V = -(g\kappa + f)$$

$$\vec{V} = V\nu$$

Obtain the new curve/surface $\vec{X}^{n+1} = \vec{X}^{n} + \tau_n \vec{V}$

Computing the Velocity

At each iteration solve the following to get $\vec{V}$

System of PDEs

$$\vec{\kappa} = -\Delta_{\Gamma}\vec{X}$$

$$\kappa = \vec{\kappa} \cdot \nu$$

$$V = -(g\kappa + f)$$

$$\vec{V} = V\nu$$

Obtain the new curve/surface $\vec{X}^{n+1} = \vec{X}^{n} + \tau_n\vec{V}$

Computing the Velocity

At each iteration solve the following to get $\vec{V}$

System of PDEs	Weak form
$\vec{\kappa} = -\Delta_\Gamma \vec{X}$	$\langle \vec{\kappa}, \vec{\phi} \rangle = \langle \nabla_\Gamma \vec{X}, \nabla_\Gamma \vec{\phi} \rangle$
$\kappa = \vec{\kappa} \cdot \nu$	$\langle \kappa, \phi \rangle = \langle \vec{\kappa} \cdot \nu, \phi \rangle$
$V = -(g\kappa + f)$	$\langle V, \phi \rangle = -\langle g\kappa + f, \phi \rangle$
$\vec{V} = V\nu$	$\langle \vec{V}, \vec{\phi} \rangle = \langle V, \nu \cdot \vec{\phi} \rangle$

Obtain the new curve/surface $\vec{X}^{n+1} = \vec{X}^n + \tau_n \vec{V}$

Linear System

At each iteration solve the following to get $\vec{V}$

Weak form	Linear system
$\langle \vec{\kappa}, \vec{\phi} \rangle = \langle \nabla_\Gamma \vec{X}, \nabla_\Gamma \vec{\phi} \rangle$	$\vec{M}\mathbf{K} = \vec{A}\mathbf{X}$
$\langle \kappa, \phi \rangle = \langle \vec{\kappa} \cdot \nu, \phi \rangle$	$M\mathbf{K} = \vec{N}^T\vec{\mathbf{K}}$
$\langle V, \phi \rangle = -\langle g\kappa + f, \phi \rangle$	$M\mathbf{V} = -M_g\mathbf{K} - \mathbf{f}$
$\langle \vec{V}, \vec{\phi} \rangle = \langle V, \nu \cdot \vec{\phi} \rangle$	$\vec{M}\vec{\mathbf{V}} = \vec{N}\mathbf{V}$

Obtain the new curve/surface $\vec{X}^{n+1} = \vec{X}^n + \tau_n \vec{V}$

Other Descent Directions

Instead of the velocity eqn $\langle V, \phi \rangle = -\langle g\kappa + f, \phi \rangle$

Use the more general velocity eqn with the bilinear form

$$b(V, \phi) = -\langle g\kappa + f, \phi \rangle$$

For example, use 2nd shape deriv. for Newton's method

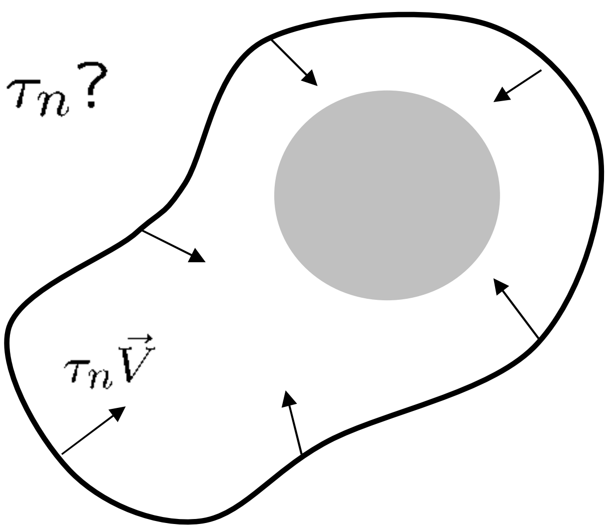
$$b(V, \phi) = \langle \alpha \nabla_\Gamma V, \nabla_\Gamma \phi \rangle + \langle \beta V, \phi \rangle$$

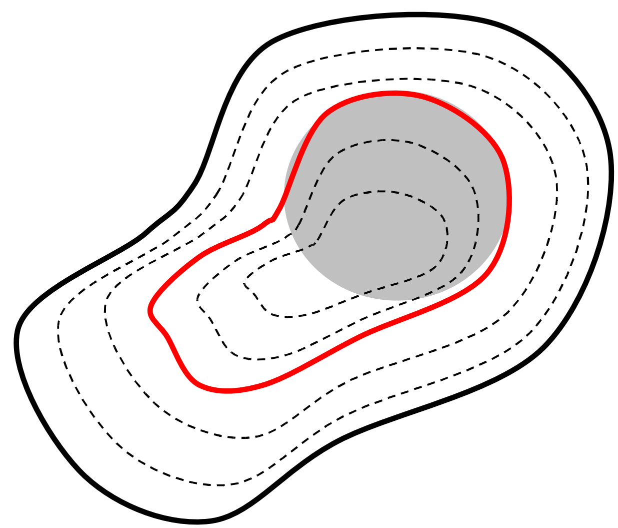
Practical Issues: Step Size

How to choose the right step τ_n in $\vec{X}^{n+1} = \vec{X}^n + \tau_n \vec{V}$

- τ_n too small $\rightarrow$ too many iterations
- τ_n too large $\rightarrow$ may miss the objects

Soln: perform **backtracking** or **line search**

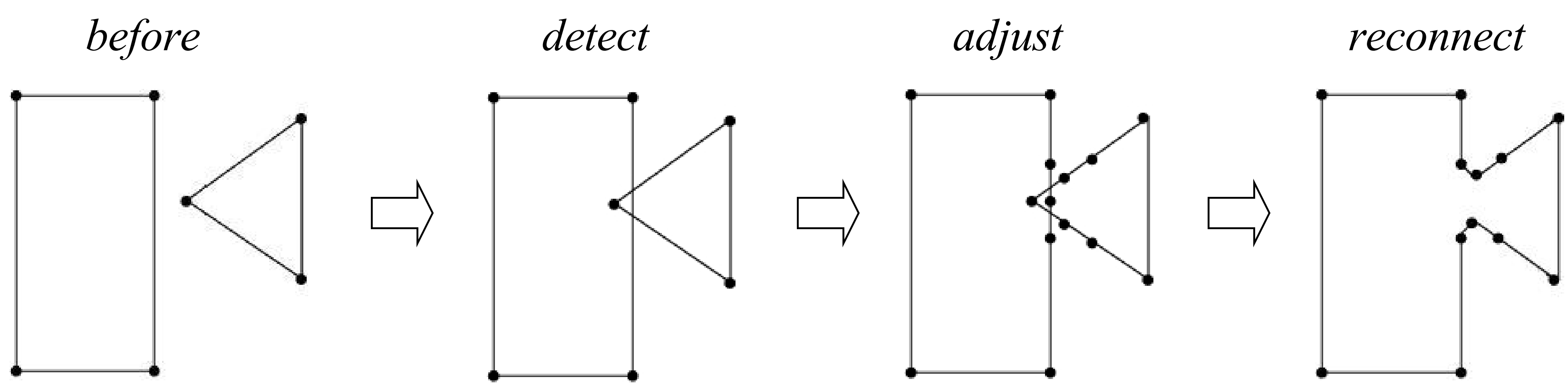




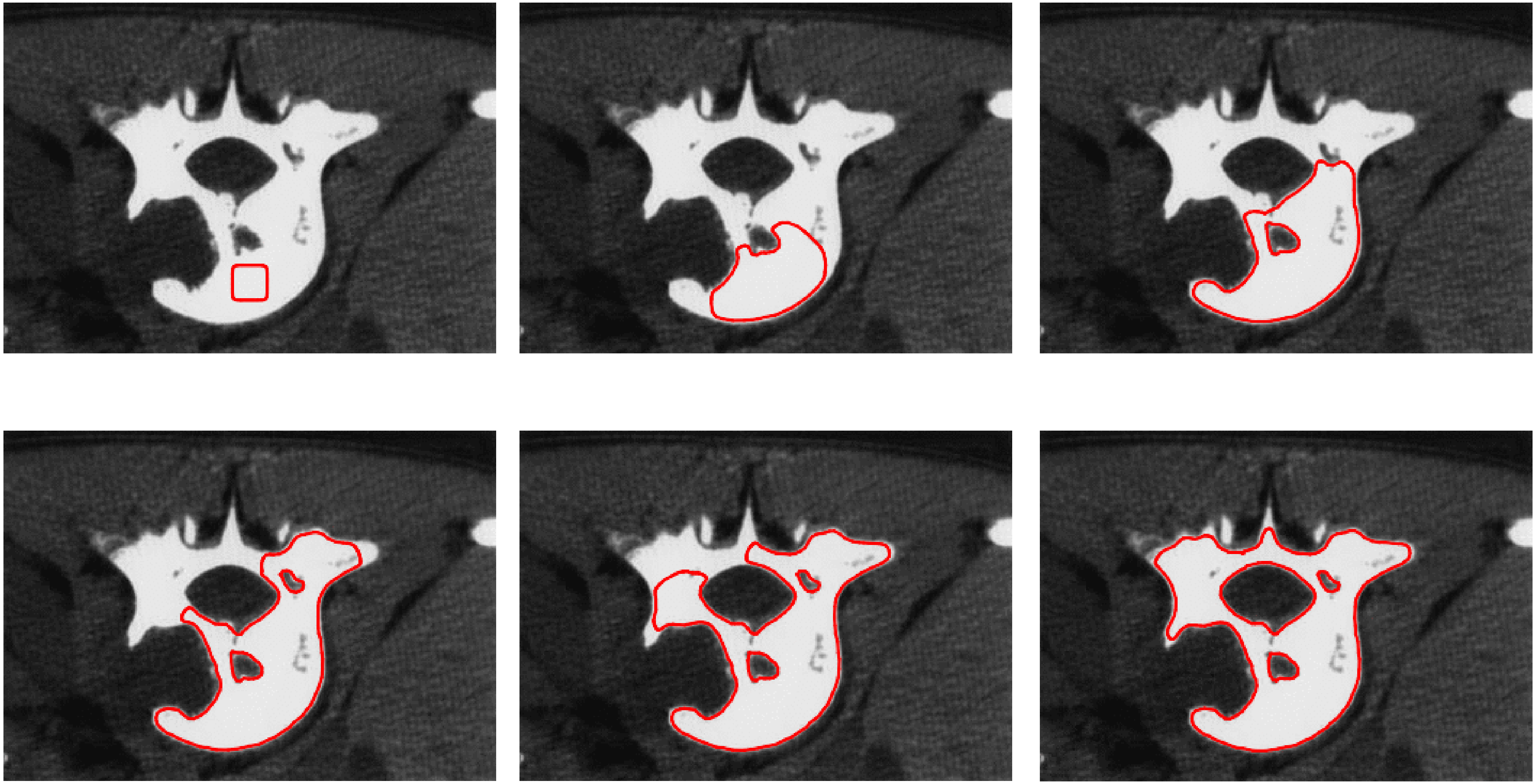
Practical Issues: Topological Changes

Four step procedure for topological changes in 2D

- detect element intersections
- adjust intersection locations
- reconnect elements
- clean up artifacts



Example: Medical Image

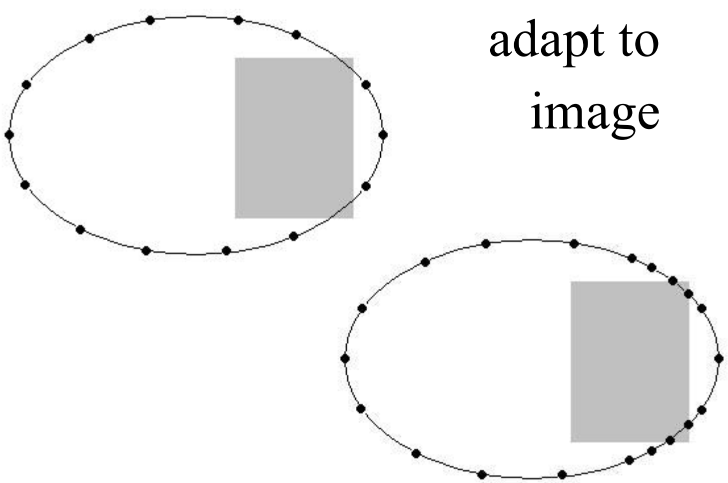


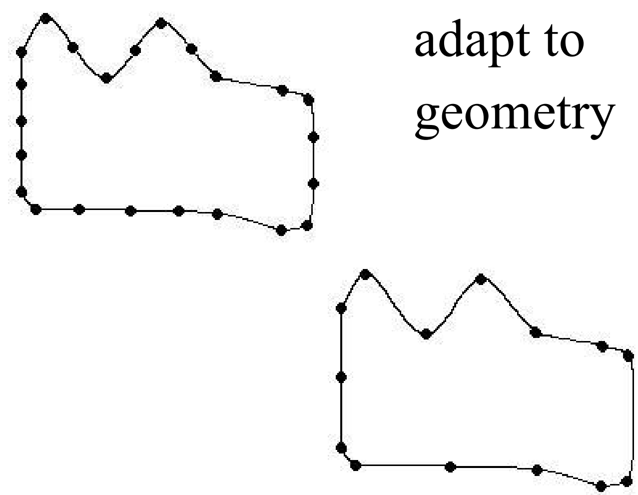
Practical Issues: Resolution

How to choose the right number of elements?

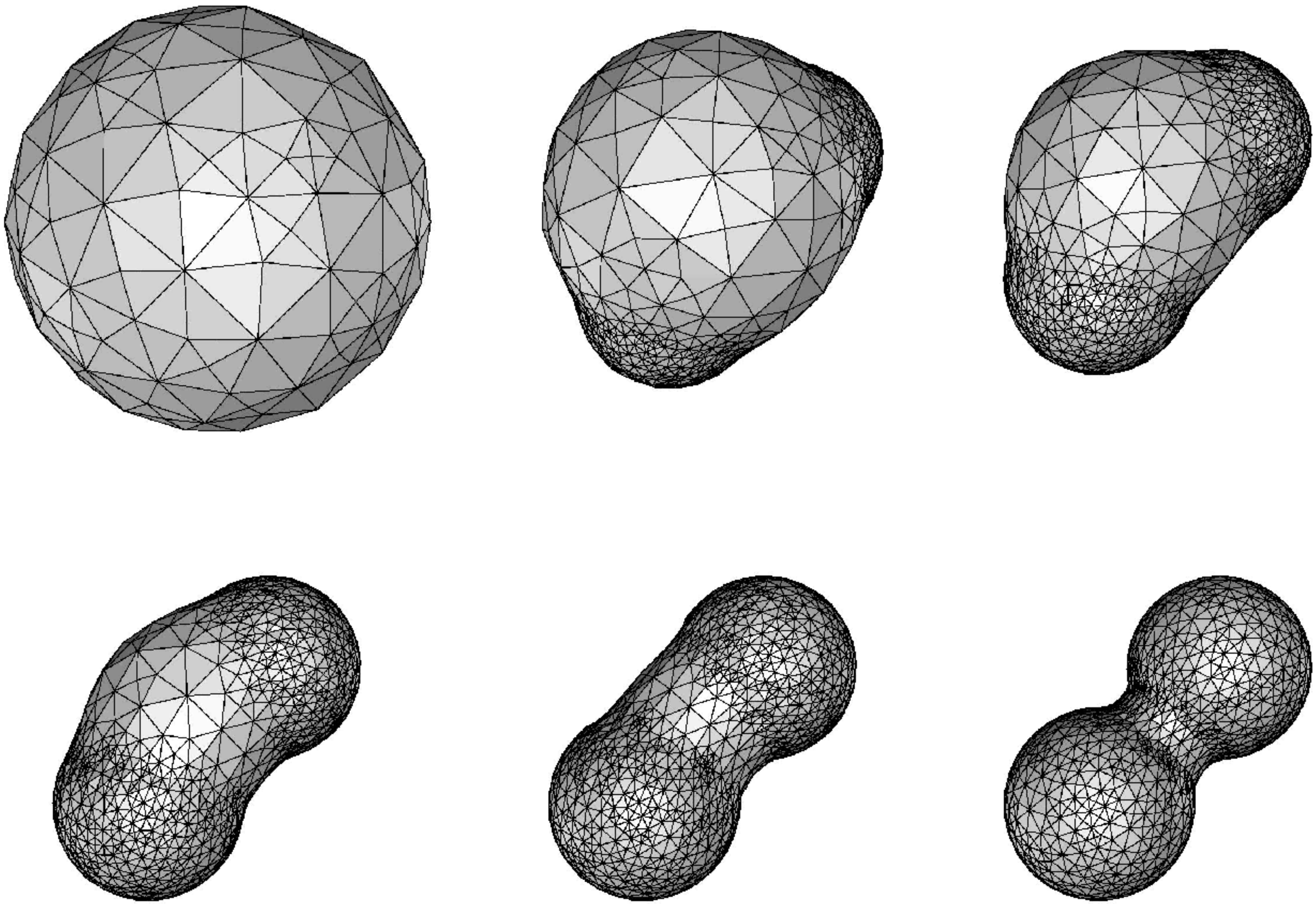
- too many elements → too many computations
- too few elements → may miss the objects

Soln: employ **space adaptivity** to adjust resolution

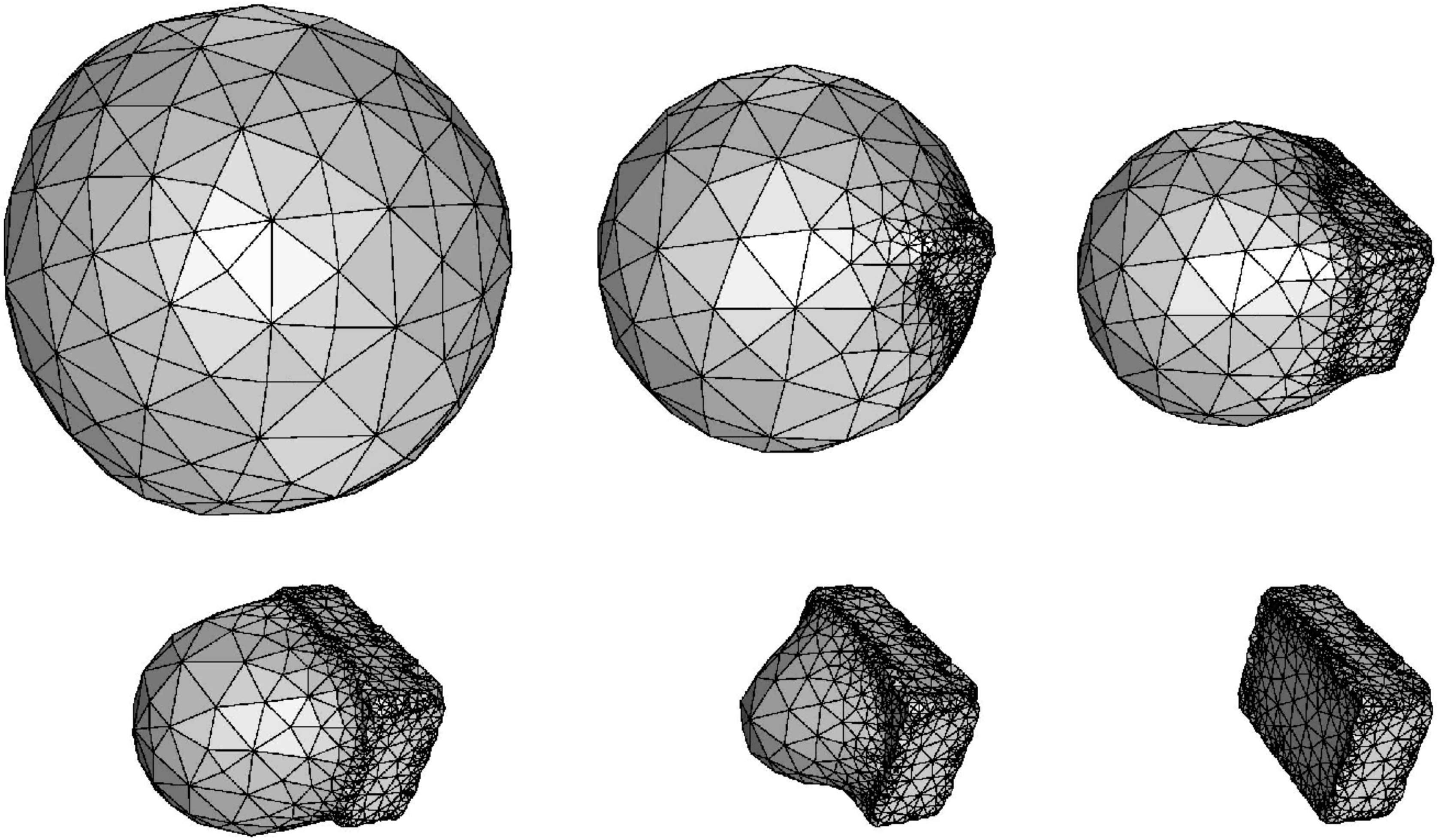




3d Example: Touching Balls



3d Example: Prism



Mumford-Shah Energy

$$J(\Gamma) = \sum_{i=1}^{2} \frac{1}{2} \left(\int_{\Omega_i} (u_i - I)^2 + \mu |\nabla u_i|^2 \right) dx + \gamma \int_{\Gamma} dS$$

subject to

$$\begin{cases} -\mu \Delta u_i + u_i & = I \quad \text{in} \quad \Omega_i \\ \partial_{\nu_i} u_i & = 0 \quad \text{on} \quad \partial\Omega_i \end{cases}$$

First shape derivative

$$dJ(\Gamma; V) = \int_{\Gamma} \left(\frac{1}{2} [\![|u - I|^2]\!] + \frac{\mu}{2} [\![|\nabla_{\Gamma} u|^2]\!] + \gamma \kappa \right) V \, dS$$

where $[\![f]\!] = f_1 - f_2$ jump of f across Γ

Mumford-Shah Energy

Second shape derivative

$$d^2 J(\Gamma; V, W) = \int_\Gamma (\alpha \nabla_\Gamma V \cdot \nabla_\Gamma W + \beta V W)\, dS + \text{other}$$

with

$$\begin{aligned} \alpha &= \gamma \\ \beta &= \frac{\mu}{2}\left(\kappa \left[\!\left[|\nabla u|^2 \right]\!\right] + \partial_\nu \left[\!\left[|\nabla u|^2 \right]\!\right]\right) \\ &\quad + \frac{1}{2}\left(\kappa \left[\!\left[|u - I|^2 \right]\!\right] + \partial_\nu \left[\!\left[|u - I|^2 \right]\!\right]\right) \end{aligned}$$

(Hintermueller & Ring, 03)

Mumford-Shah Energy

Velocity equation

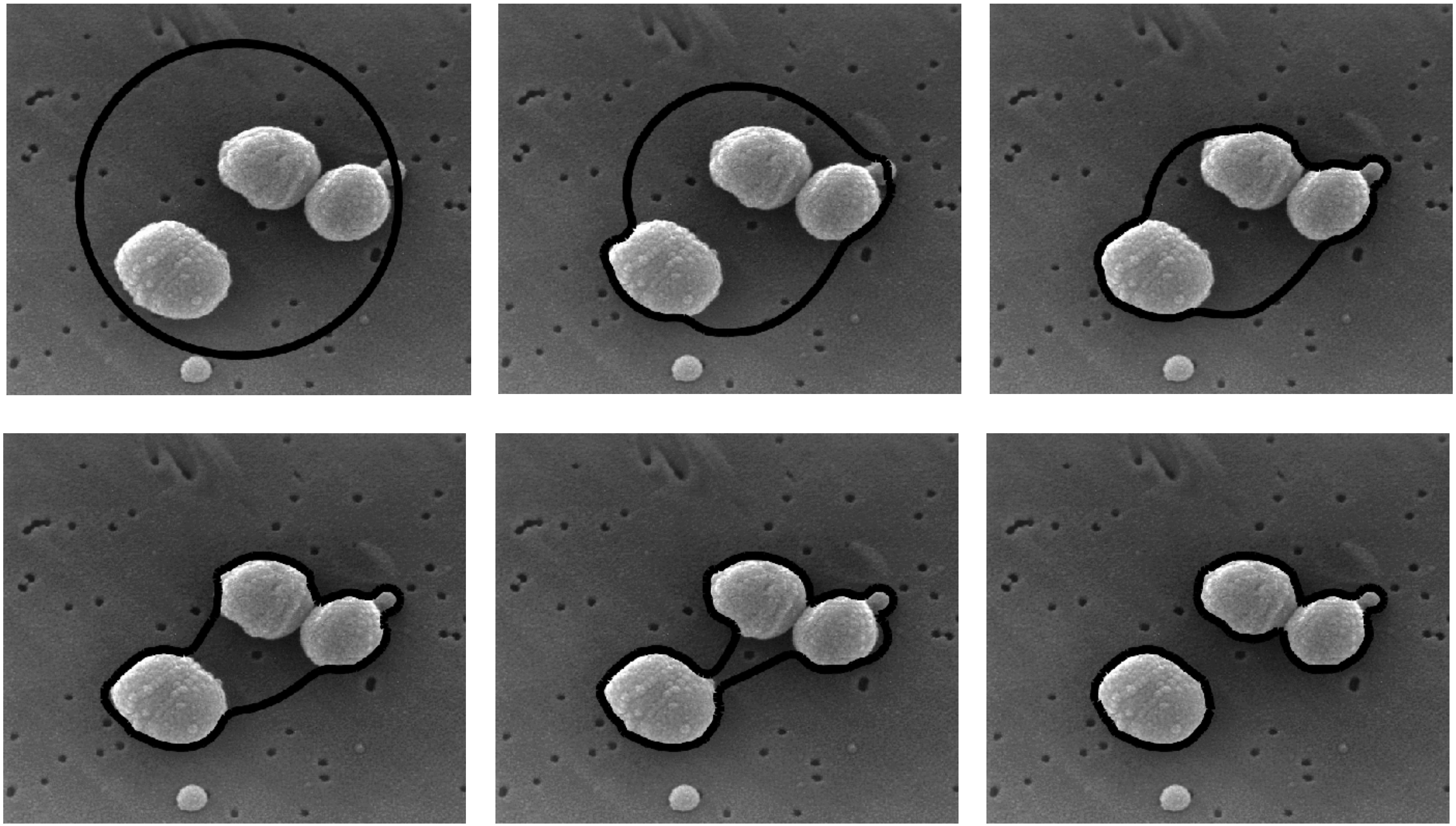
$$b(V,\phi) = -dJ(\Gamma;\phi), \qquad \forall \phi \in B(\Gamma)$$

Two choices of scalar products

- L^2 flow: $b(V,W) = \int_\Gamma VW\,dS$
- H^1 flow: $b(V,W) = \int_\Gamma (\alpha \nabla_\Gamma V \cdot \nabla_\Gamma W + \beta VW)\,dS$

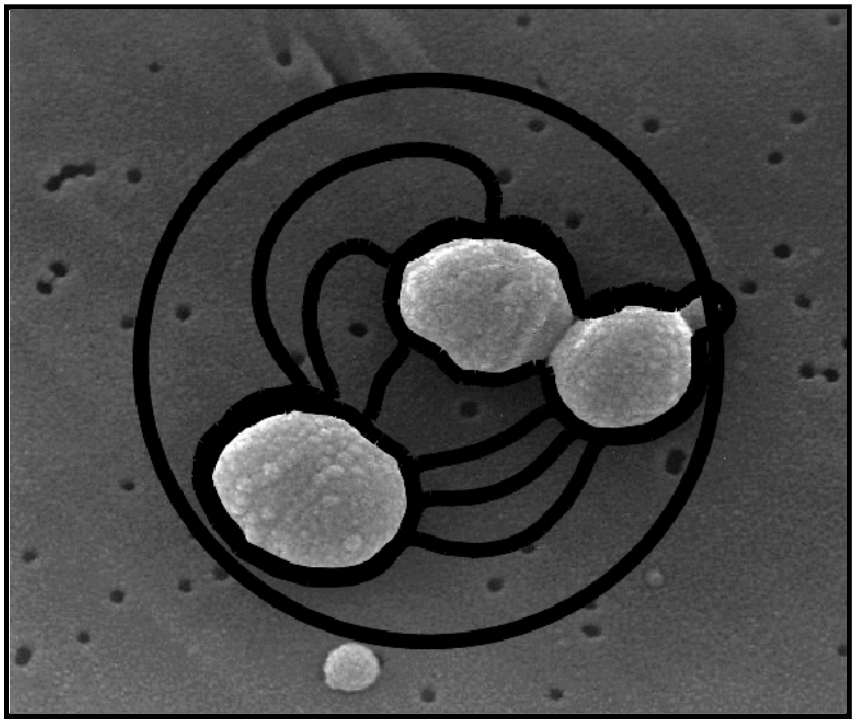
$$\alpha = \gamma,$$
$$\beta = \frac{1}{2}\left(\mu\kappa \left[\!\left[|\nabla u|^2\right]\!\right] + \kappa \left[\!\left[|u-I|^2\right]\!\right] + \partial_\nu \left[\!\left[|u-I|^2\right]\!\right]\right)_+$$

Bacteria: H^1 Flow



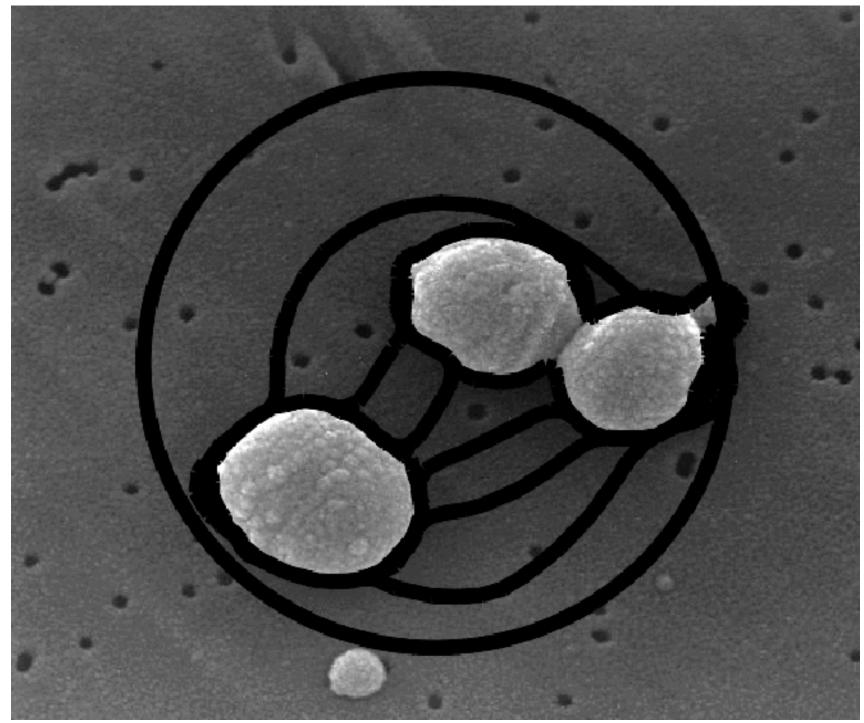
Bacteria: L^2 Flow vs H^1 Flow

L^2 flow



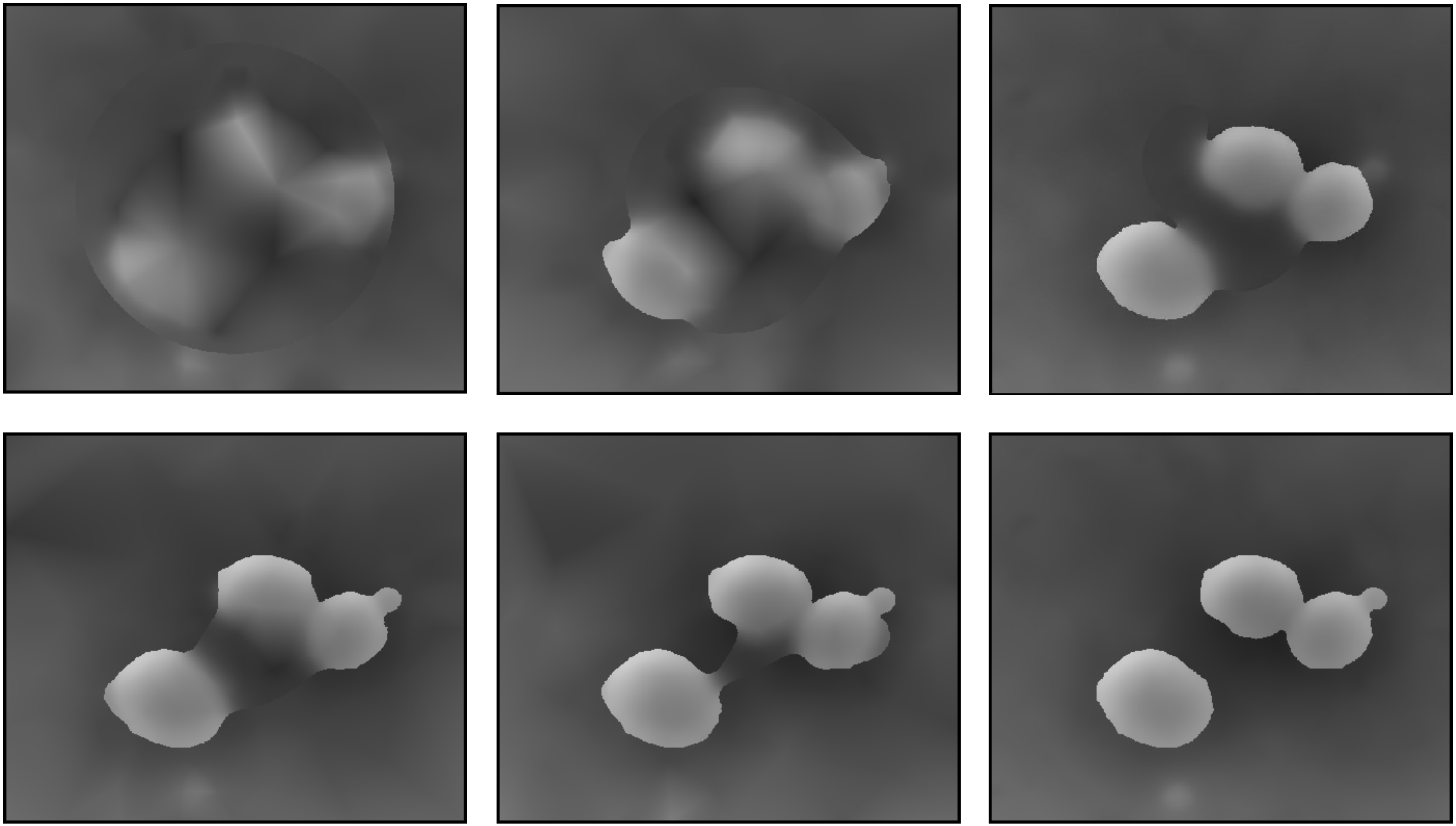
586 iters, 2m 51s

H^1 flow

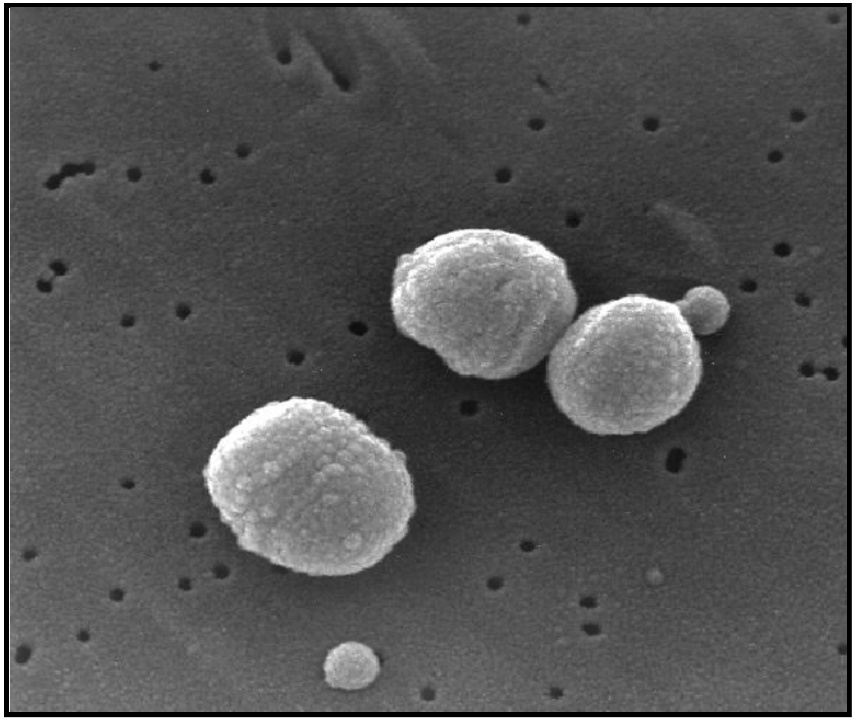


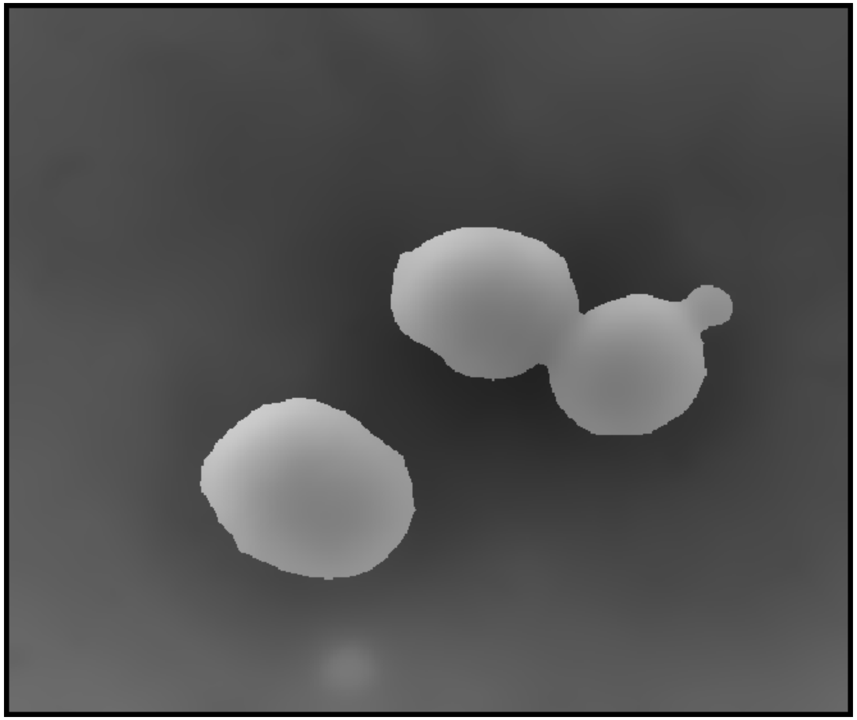
142 iters, 43s

Bacteria: Pw. Smooth Approximations

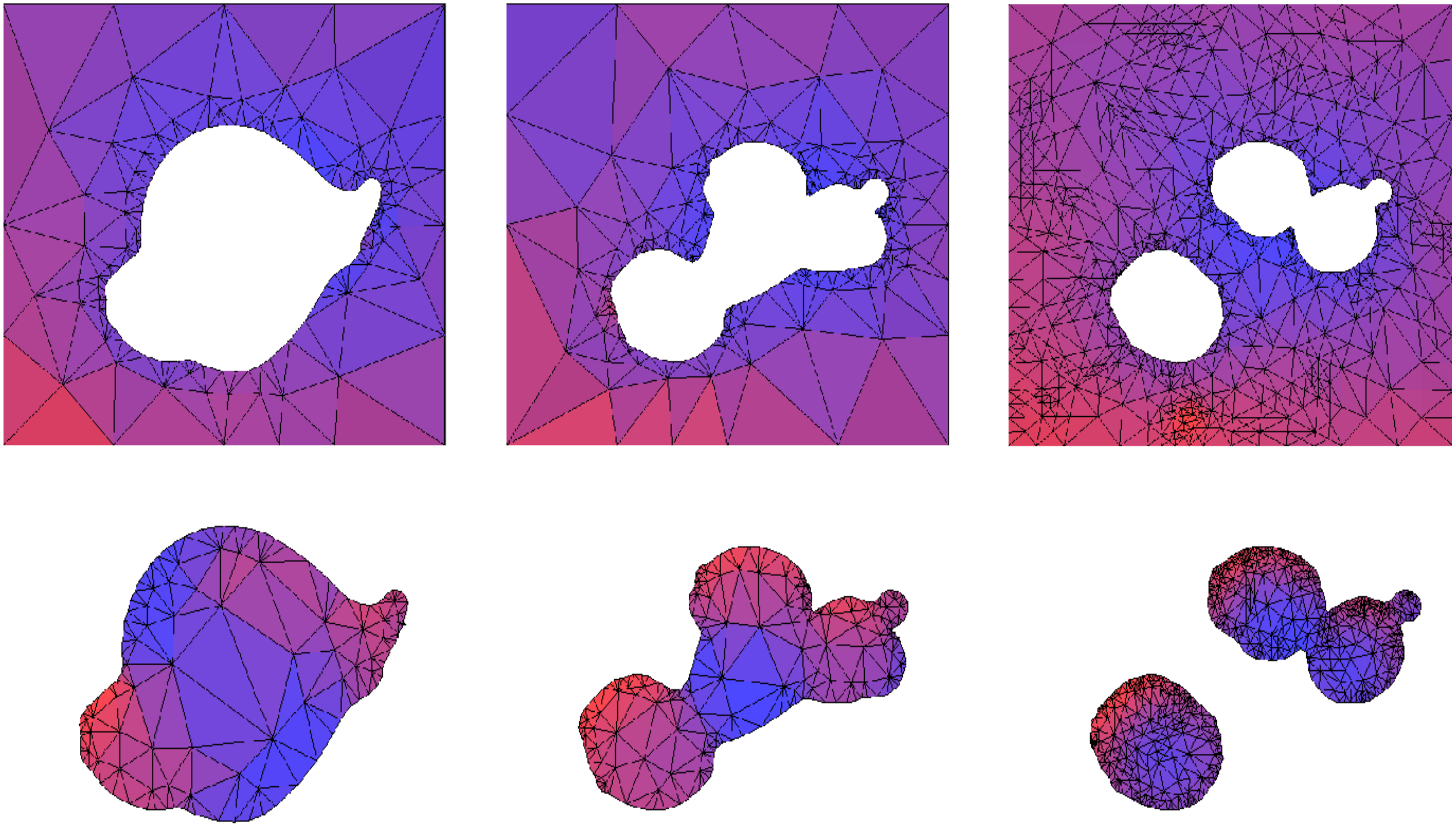


Bacteria: Pw. Smooth Approximation

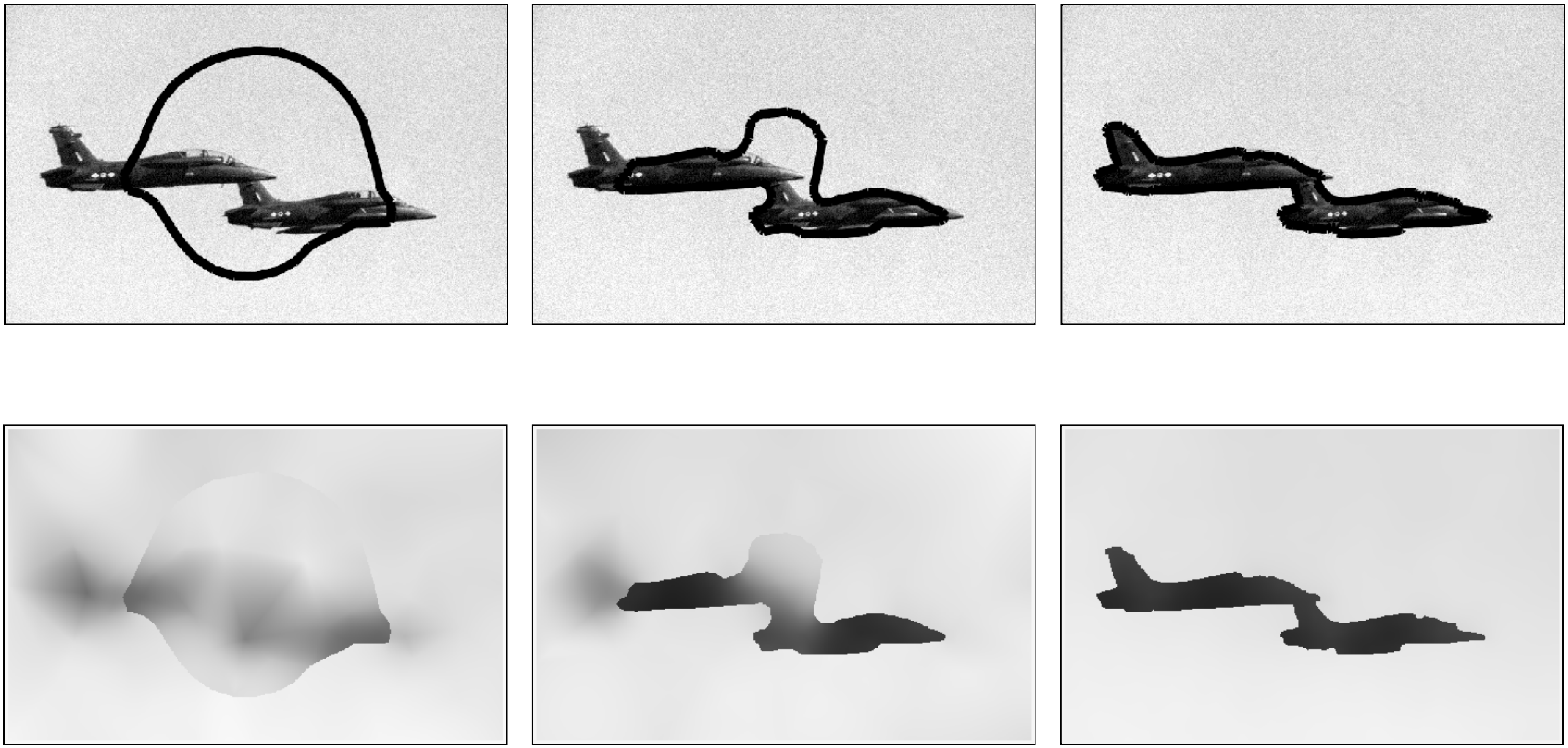




Domain Meshes

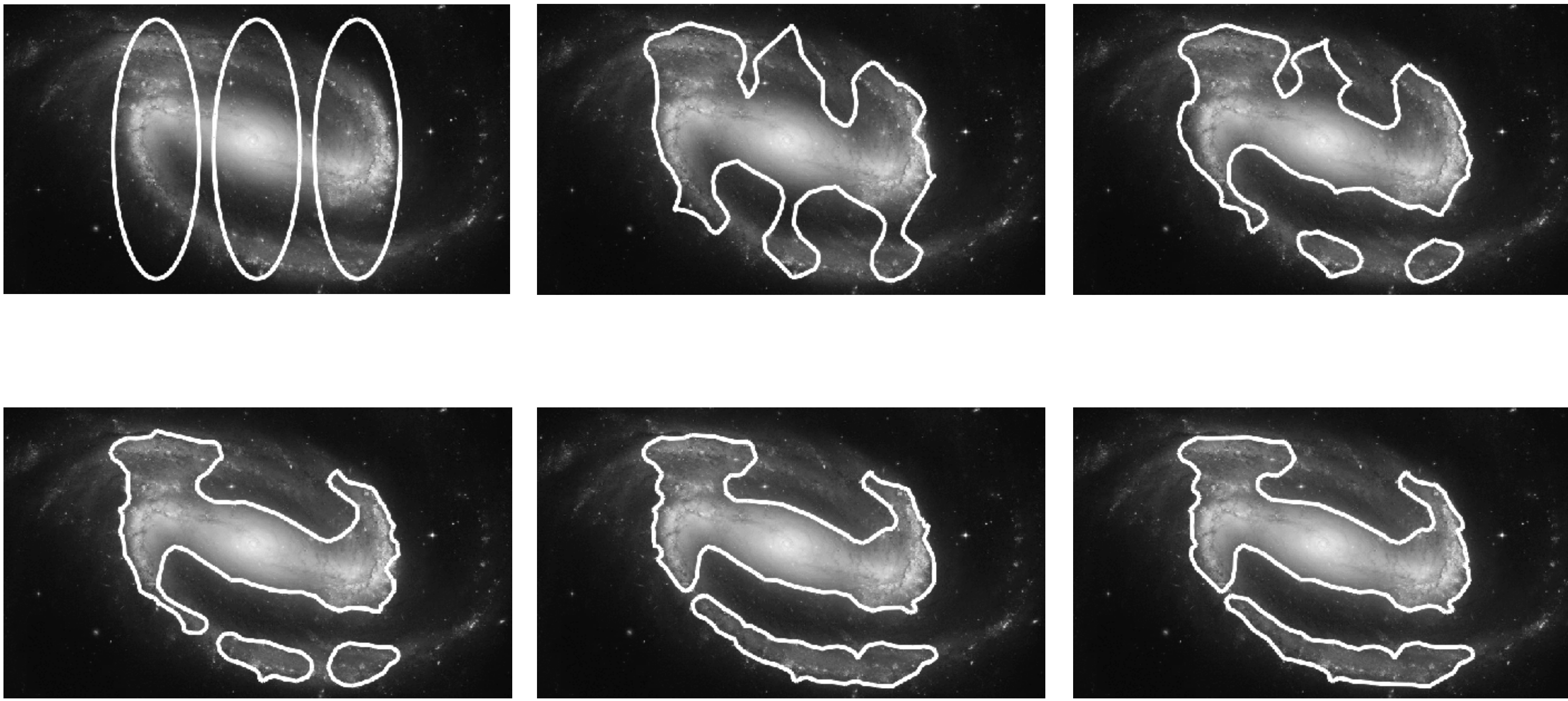


Simultaneous Segmentation & Denoising



107 iters, 47s

Galaxy: No Edges



53 iters, 20s

Summary

- Introduced shape optimization for image segmentation
- Started with shape sensitivity analysis, i.e. shape derivatives
- Implemented discrete gradient flows with finite elements
- Implemented computational enhancements for robustness
- Applications: Geodesic Active Contours,
  Mumford-Shah Model