CURSO DE POSTGRADO

Procesamiento de Imágenes y Bioseñales I & II

Image Segmentation II

Advanced Approaches

Jorge Jara W. [1,2]

[1] Department of Computer Science DCC, U. of Chile

[2] SCIAN-Lab, BNI

Outline

1. Segmentation I

    - Digital image processing
    - Segmentation basics

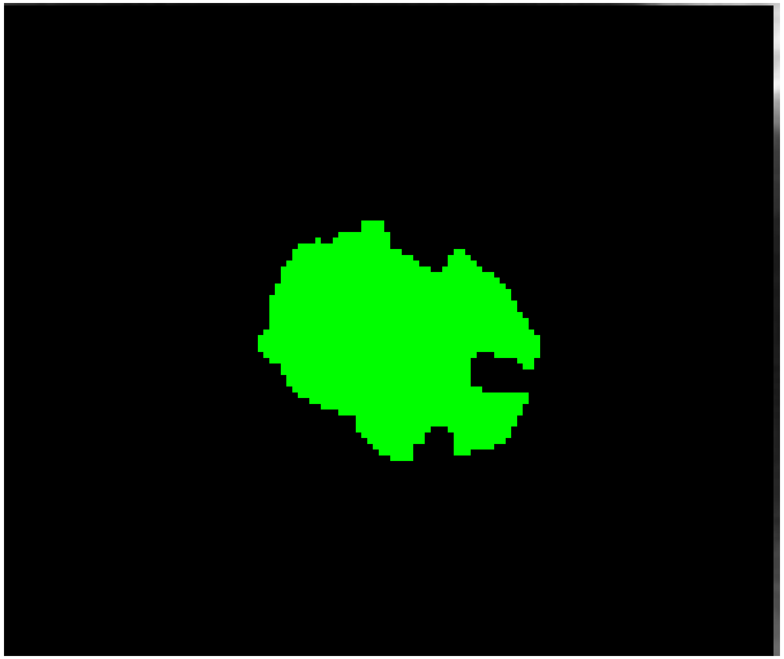
2. Segmentation II
    - Advanced techniques

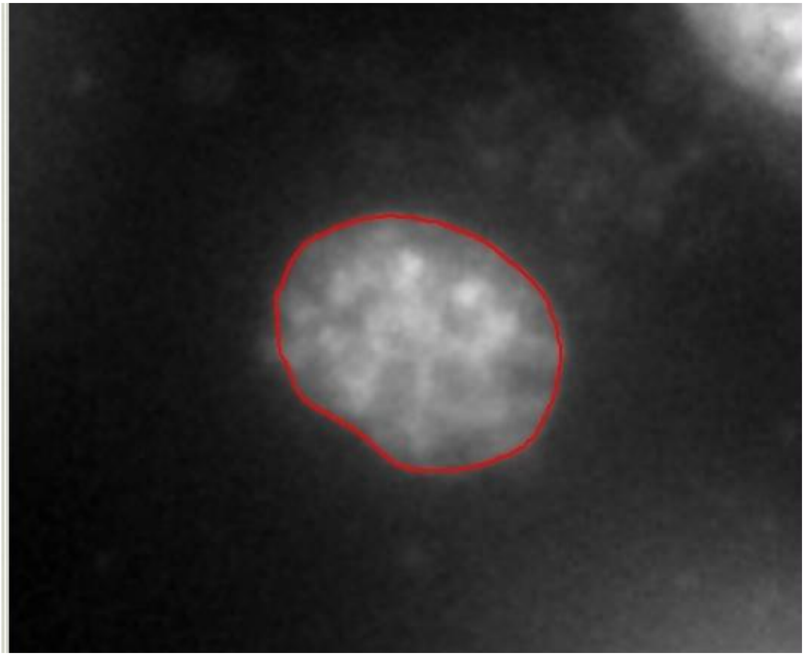
Laboratory for SCIAN Scientific Image Analysis

Segmentation – basics

Biomedical Neuroscience Institute

- “Some” times more information is needed in order to achieve a good segmentation





- Designed to optimize a cost or fitness function
  - Interest features detection (e.g. edges, points)
    - SIFT
  - Template matching
    - Hough transform
  - Pixel clustering
    - Statistics, probabilities
    - Graph cuts
    - Machine learning (support vector machine, neural networks…)
  - Differential equations, calculus of variations
    - Contour properties
    - Region properties

This can be a VERY long list…

Segmentation - Advanced

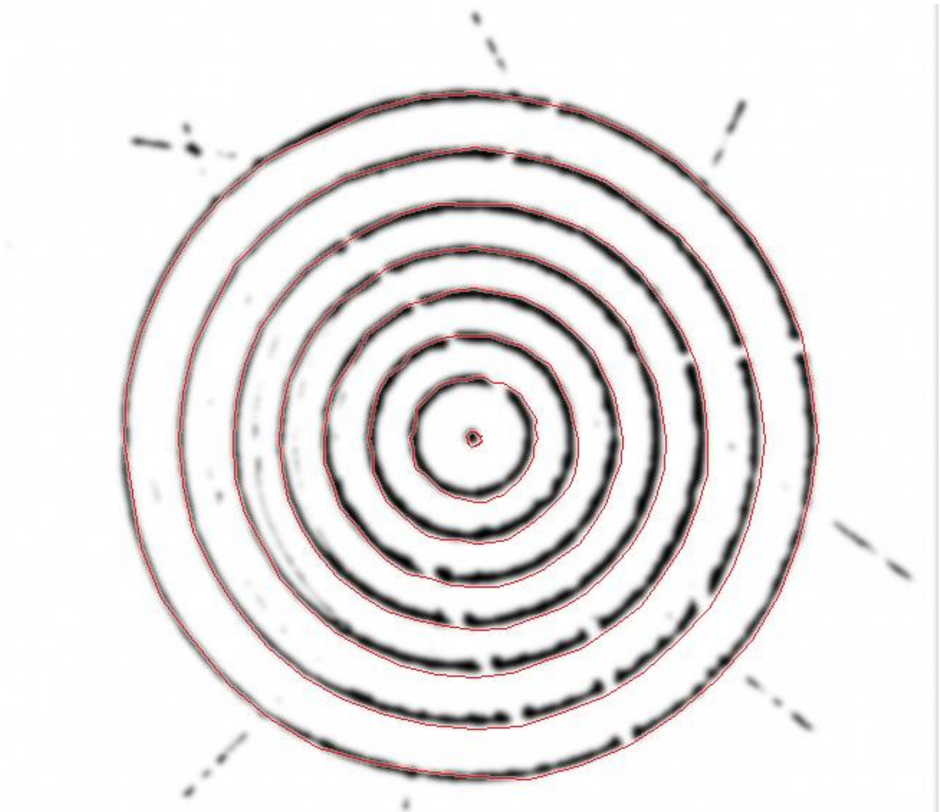
- Template matching
  - "Classic model" Hough transform
  - Applies to circles, line segments and a variety of shapes

Hough P (1959)

If we can detect edges we can approximate a circle (center, radius)

n connected edges and ***m*** (=***n*** or <> ***n***) circles

A test is performed to determine the circles with "best fit"



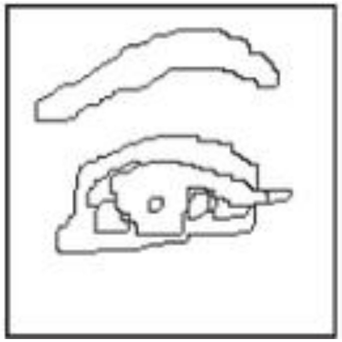
Aguilar P, Hitschfeld N (DCC, 2010)

Segmentation - Advanced

- Variational methods
  - Based on energy minimization, defining integral models
  - Idea: to include desirable features on segmented images (like homogeneus regions, short or smooth ROI boundaries)
  - Optimum solutions found by partial differential equations
  - Examples: Mumford-Shah, Ambrosio-Tortorelli, Chan-Vese (details in the book from Aubert & Kornprobst 2006)

image I

main discontinuties in I



ROI boundaries B

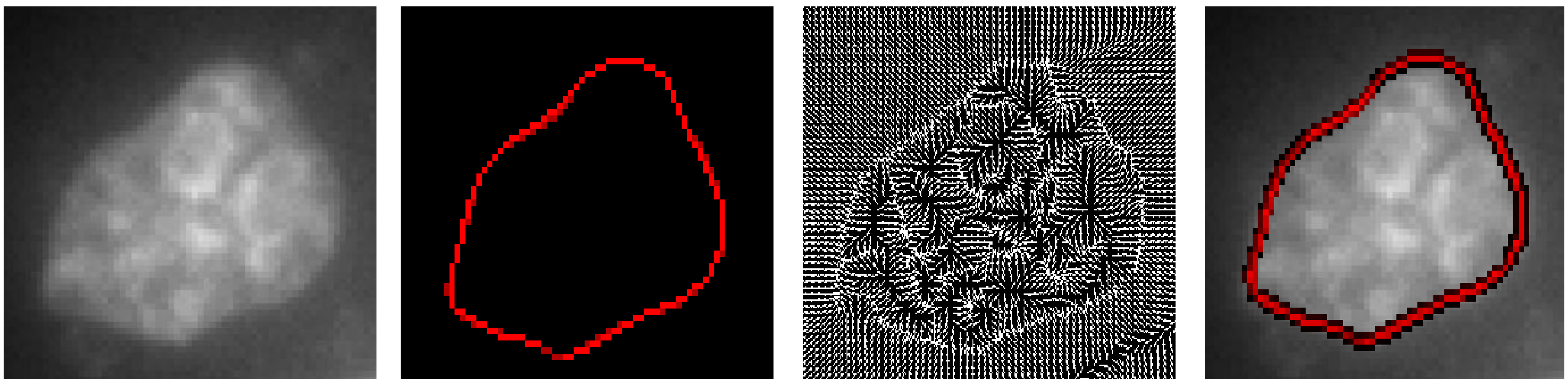
piecewise smooth image J

$$E[J,B] = C\int d\vec{x}\,(I(\vec{x}) - J(\vec{x}))^2 + A\int_{D/B} \vec{\nabla} J(\vec{x}) \cdot \vec{\nabla} J(\vec{x})\,d\vec{x} + B\int_B ds$$

The Mumford & Shah functional (1989)

Segmentation - Advanced

- Active contour models
  - Optimization of different properties



input
image
+initial guess

contour C(s)
- elasticity (contraction)
- rigidity (bending, cornering)

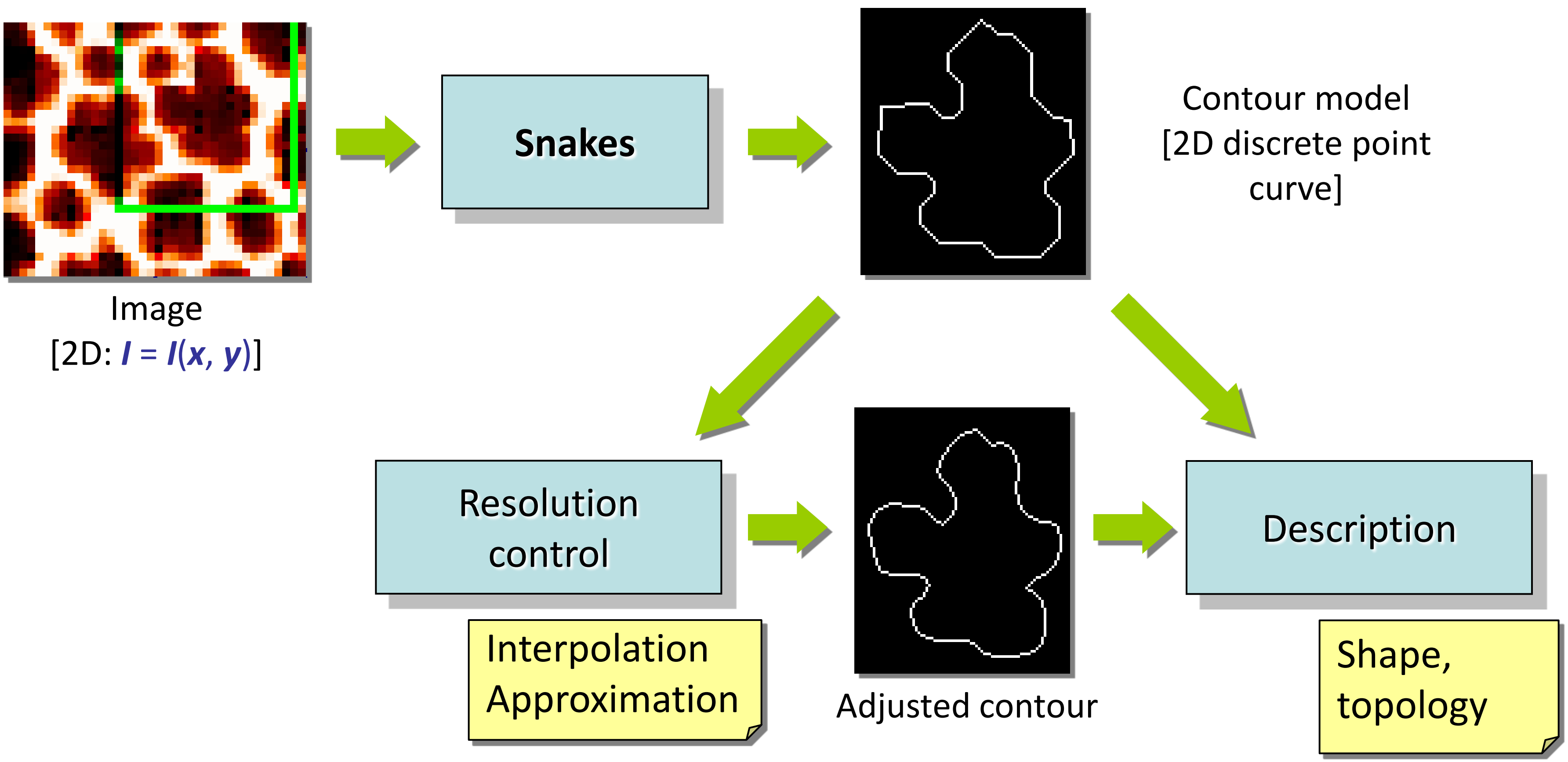
force field
- repulsion
- attraction

output: force balance
minimal energy

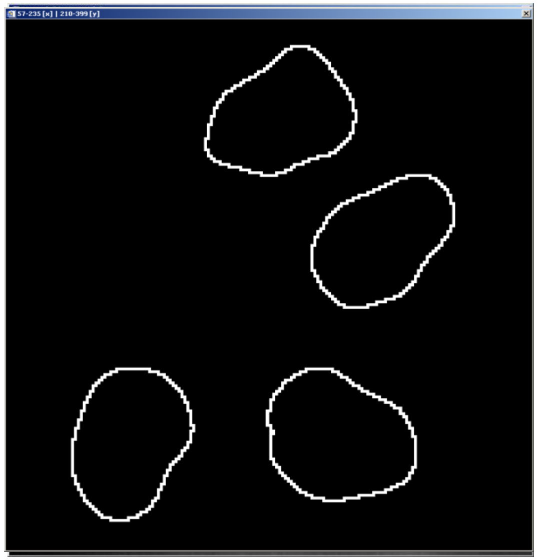
First active contours approach:
Kass, Witkin & Terzopoulos (1988)
“Snakes”

Segmentation - Advanced

- 2D active contours or *snakes*



1 contour function → 1 ROI (this is called *parametric approach*)



– 2D parametric curve

$C = C(s) = [x(s), y(s)]$

$s \in [0, 1]$ (arbitrary length)

– 2D discretization

$C = \{[x_i , y_i]; i = 0..n\}$

Segmentation - Advanced

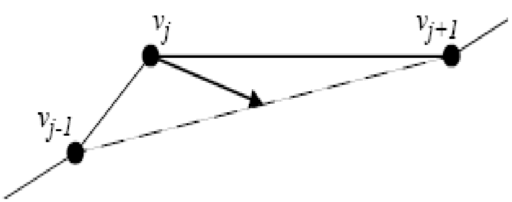
- Snakes: optimization derived from a variational approach
  - Minimization of an integral functional
    "a snake minimizes its energy"

$$E = \int_0^1 \frac{1}{2}\left[\alpha\left|\frac{\partial C(s)}{\partial s}\right|^2 + \beta\left|\frac{\partial^2 C(s)}{\partial s^2}\right|^2\right] + E_{ext}[C(s)]ds$$

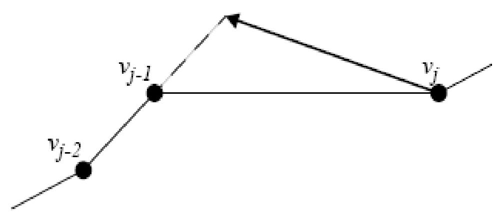
Internal energy, contour dependant

External energy, image dependant

Elasticity term (coefficient α)

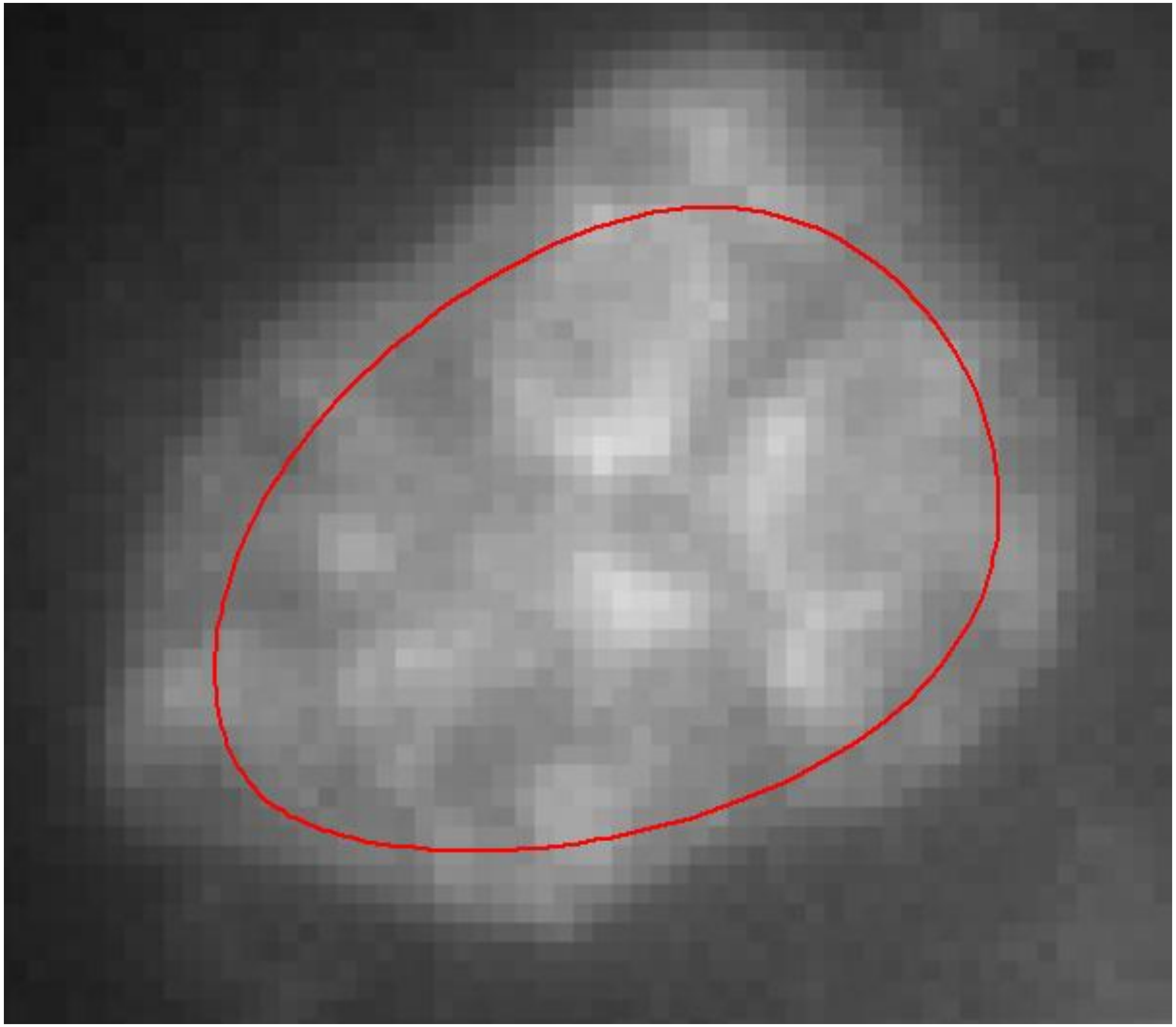


Rigidity term (coefficient β)

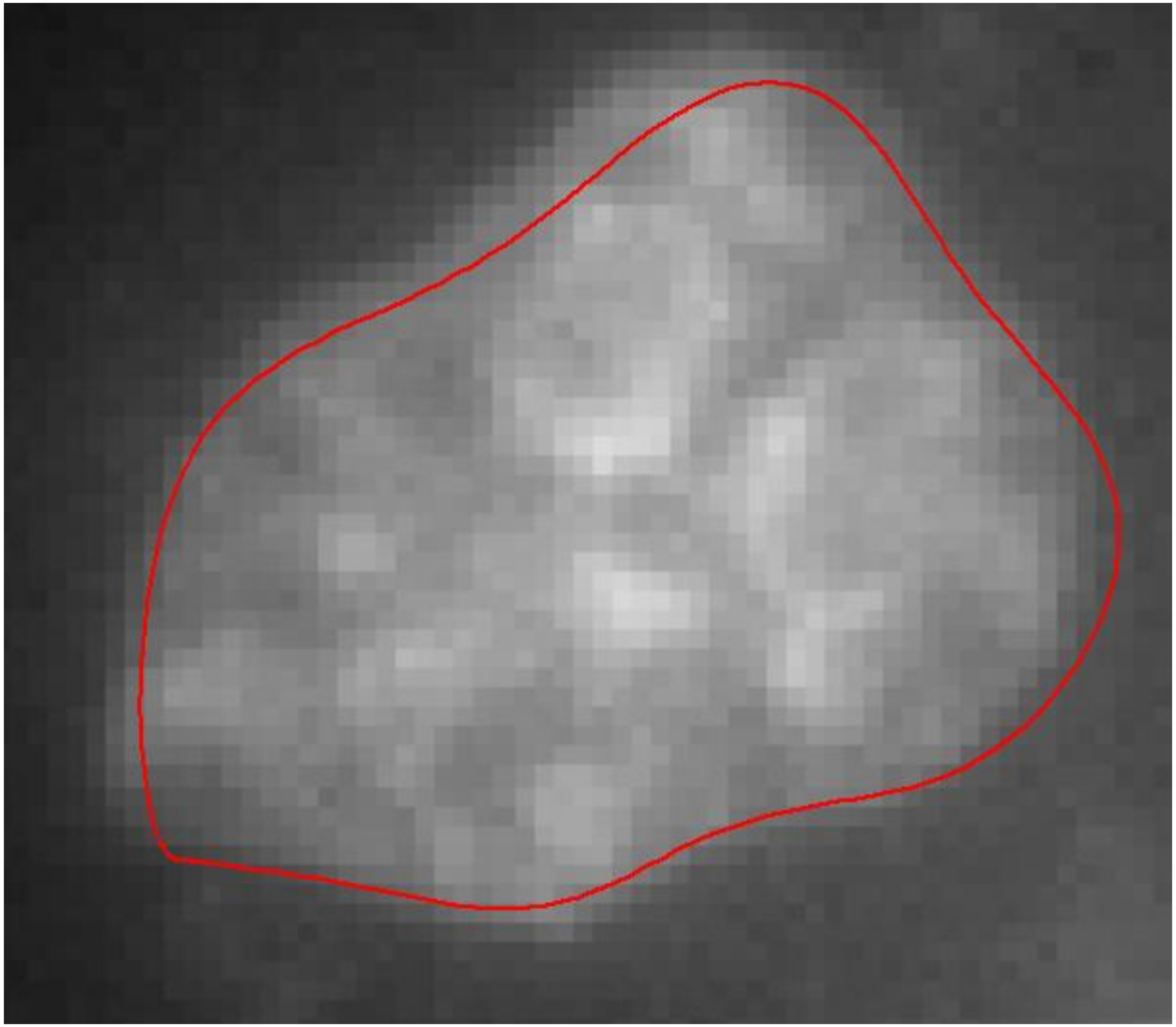


Kass et al (1988) Snakes: active contour models
Int. J. of Computer Vision 1(4): 321-331

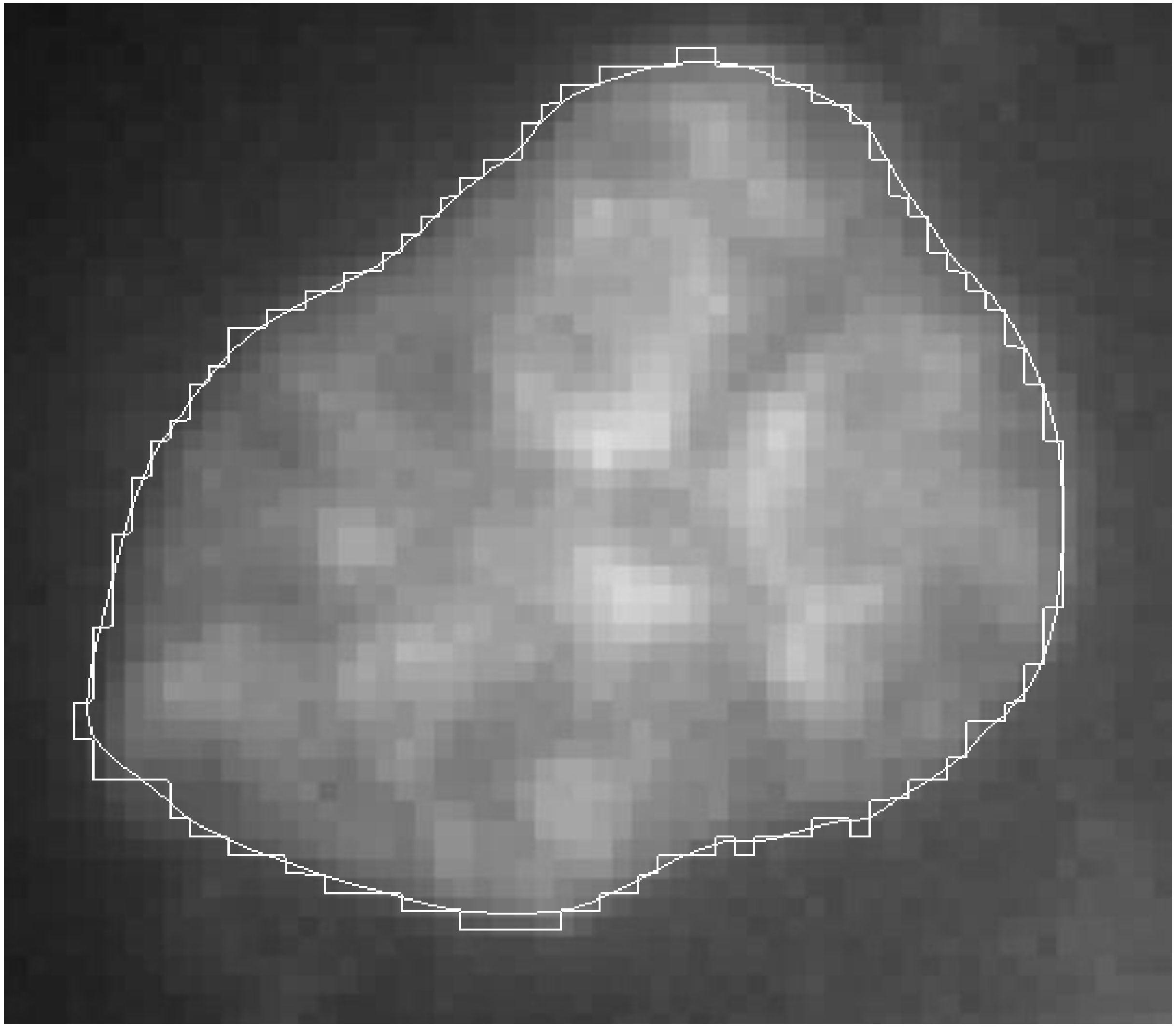
Elasticity - **α**



Rigidity - β

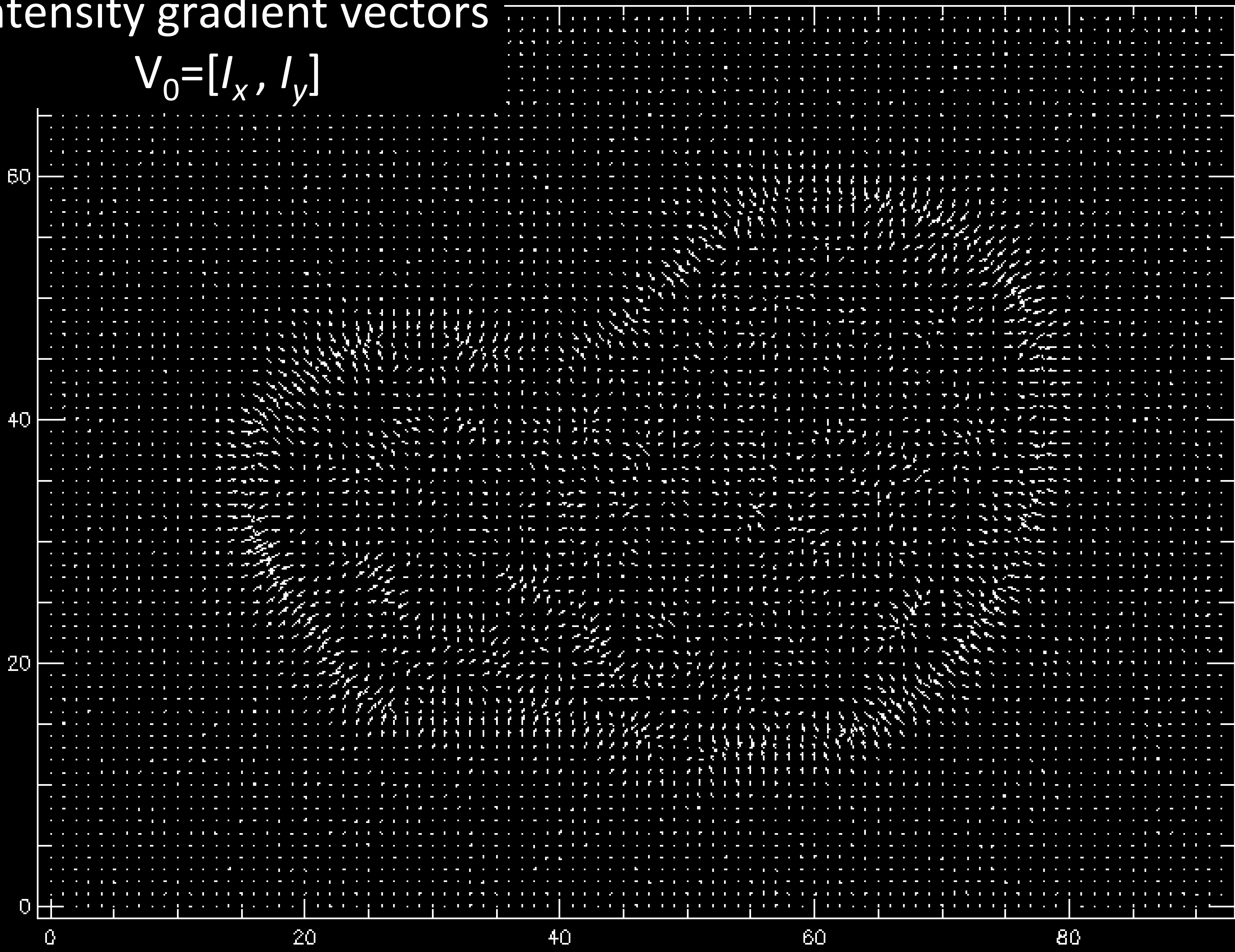


Resolution (f)



Intensity gradient vectors

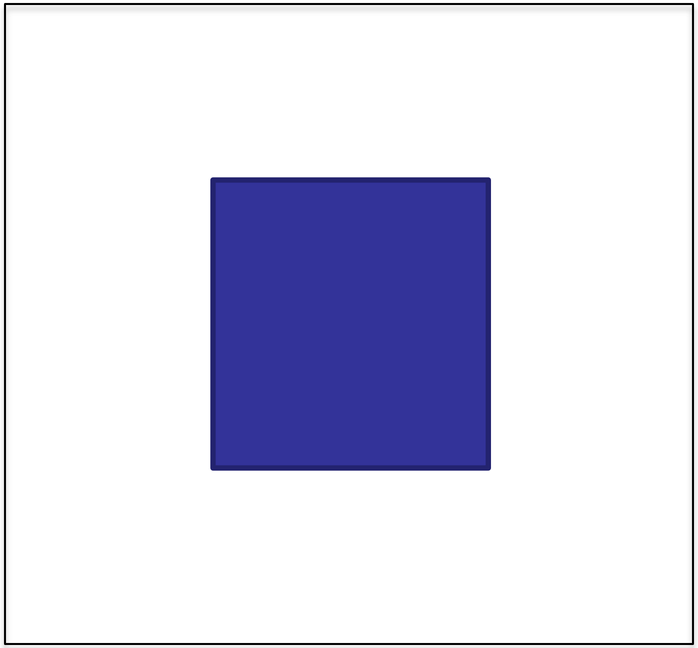
$V_0=[I_x , I_y]$

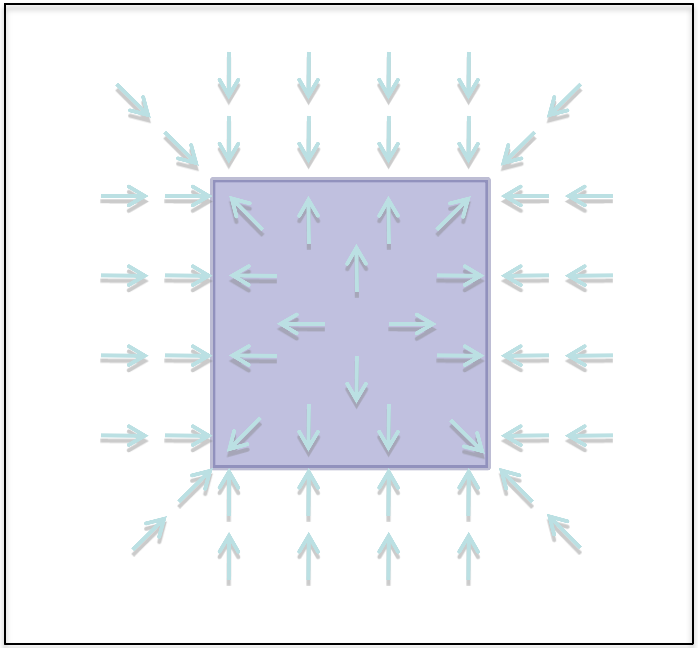


Laboratory for SCAN Scientific Image Analysis

BIOMEDICAL NEUROSCIENCE INSTITUTE

Attraction fields can be constructed from the intensity gradients





$$\underbrace{I(x,y)}$$

Image intensity

Edgemap image

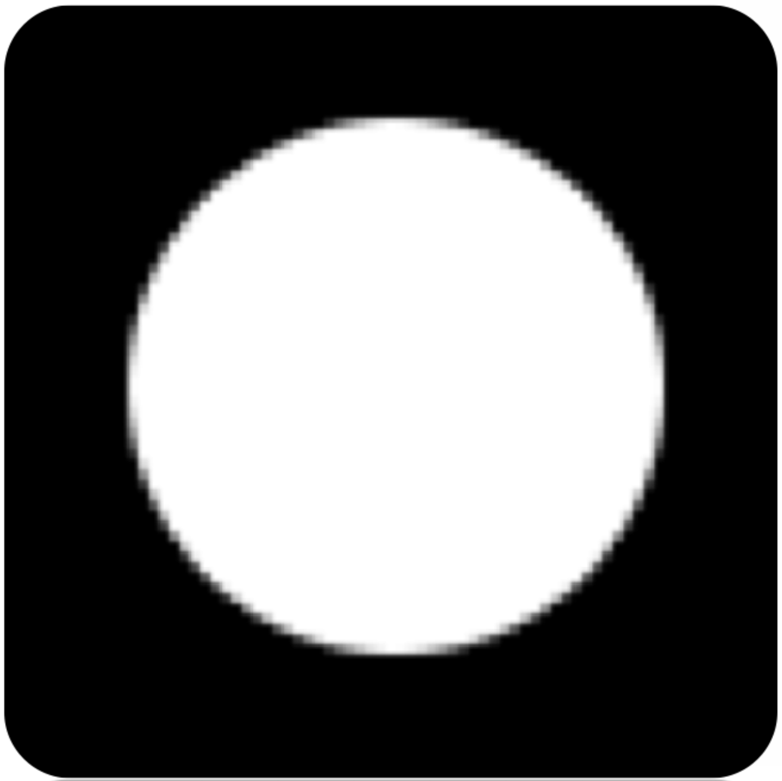
$$\nabla\left|\nabla I(x,y)\right|^2$$

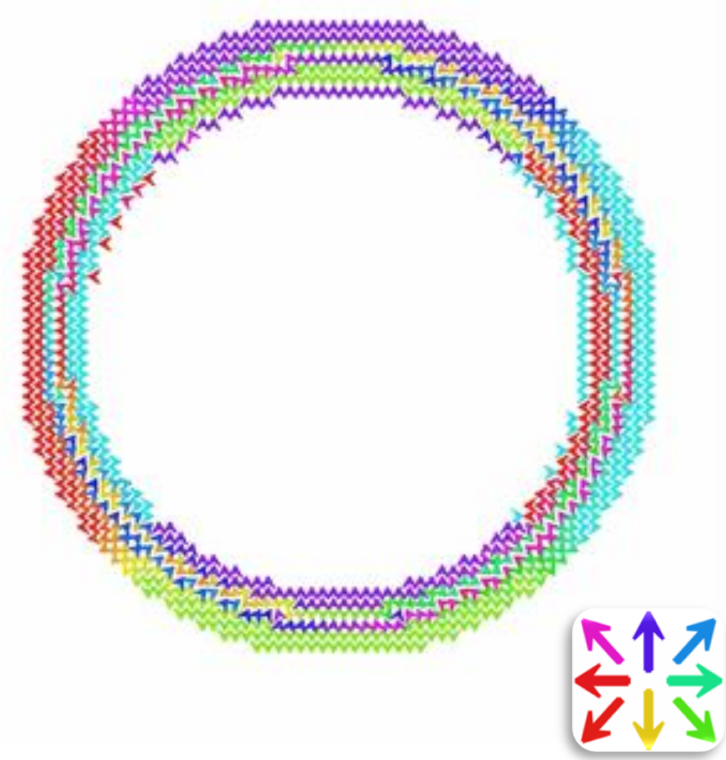
Or a gaussian-smoothed version…

$$\nabla\left|G_\sigma * \nabla I(x,y)\right|^2$$

Image force fields (external forces in active contour models)

- Gradient vector flow (GVF)
- Generalized GVF (GGVF)





Xu & Prince (1998) Active contours and gradient vector flow http://iacl.ece.jhu.edu/projects/gvf

Xu & Prince (1998) Generalized gradient vector flow external forces for active contours

GVF image force field

Defined by an energy functional to be minimized

Let

$$V(x,y) = [u,v] \qquad f = |G_\sigma * \nabla I|$$

$$u = u(x,y)$$

$$v = v(x,y)$$

$$\min_{(u,v)\in C^2(\Omega;\mathbb{R}^2)} \int_\Omega \underbrace{g(|\nabla f|)\left(|\nabla u|^2 + |\nabla v|^2\right)}_{\text{Diffusion (global smoothness) term}} + \underbrace{h(|\nabla f|)\left(|u - f_x|^2 + |v - f_y|^2\right)}_{\text{Edge similarity term}} \, dx$$

Diffusion (global smoothness) term — Edge similarity term

$$\begin{aligned} g\Delta u - h(u - f_x) &= \frac{\partial u}{\partial t} \\ g\Delta v - h(v - f_y) &= \frac{\partial v}{\partial t} \end{aligned} \;\longrightarrow\; \begin{aligned} g\Delta u - h(u - f_x) &= 0 \\ g\Delta v - h(v - f_y) &= 0 \end{aligned}$$

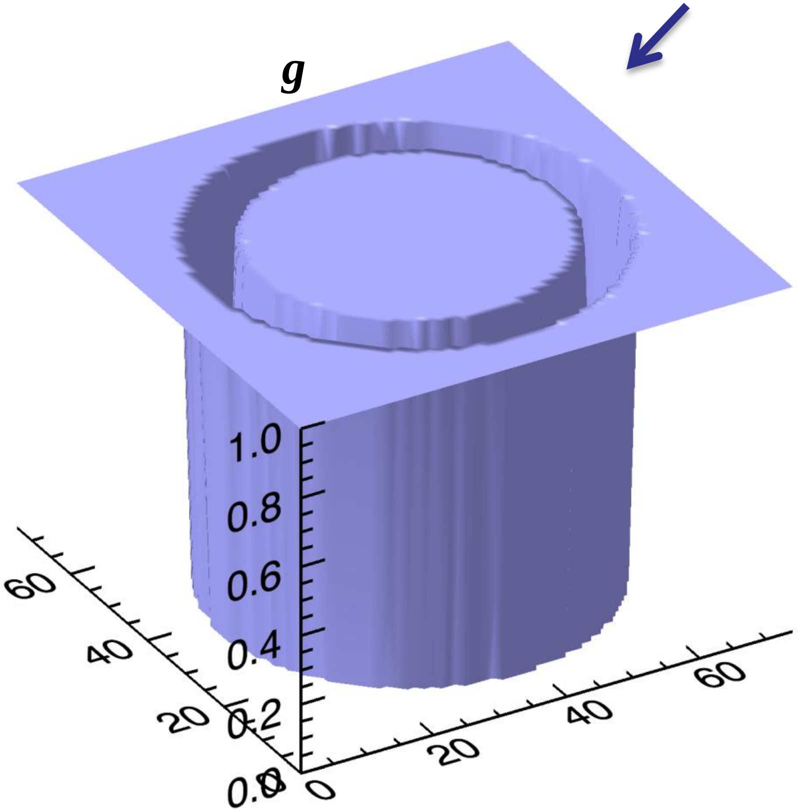
$$\downarrow$$

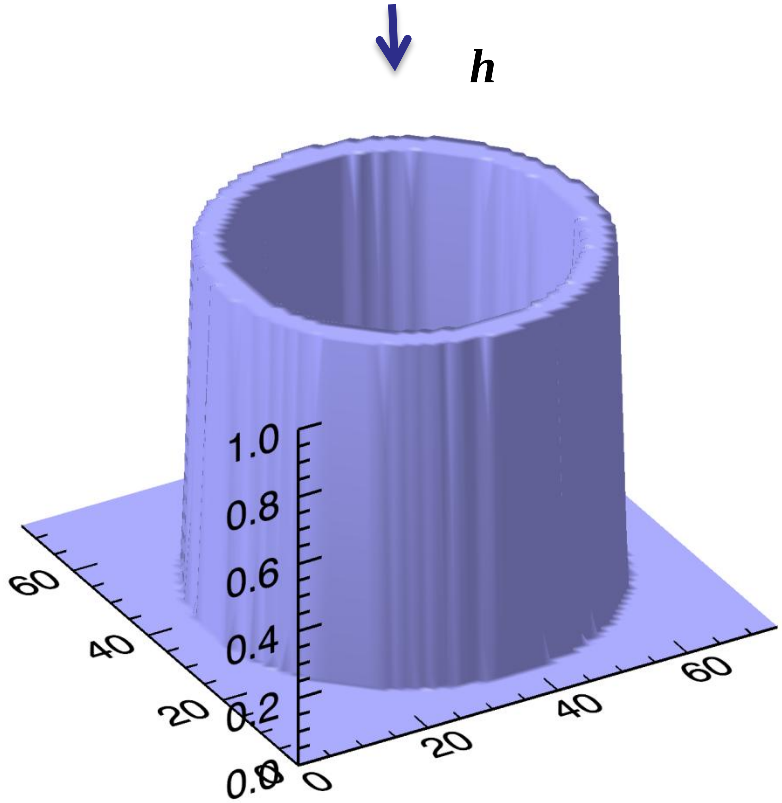
$$\begin{aligned} \nabla g \cdot \nabla u + g\Delta u - h(u - f_x) &= 0 \\ \nabla g \cdot \nabla v + g\Delta v - h(v - f_y) &= 0 \end{aligned}$$

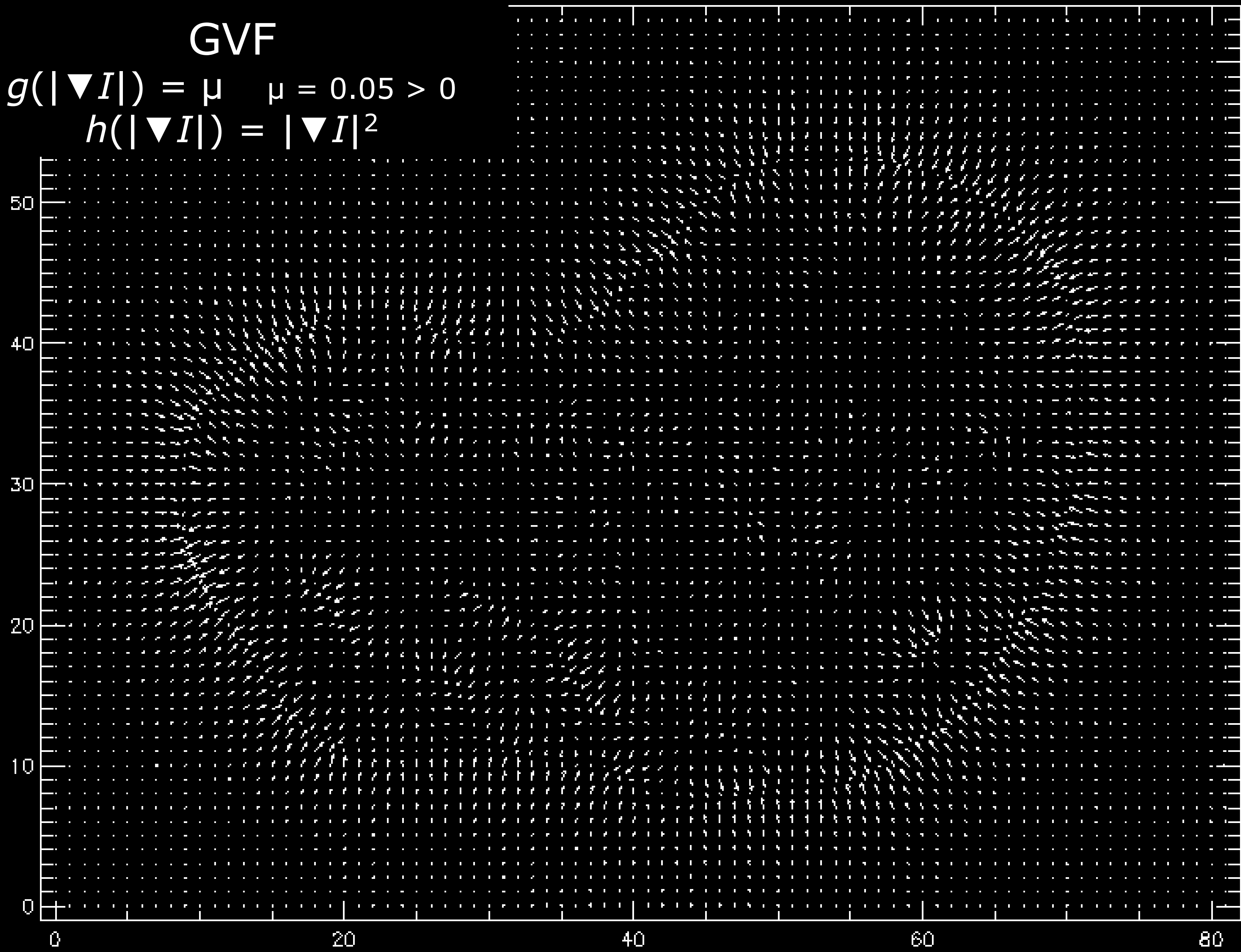
The GVF (GGVF) energy minimization

$$f = |G_\sigma * \nabla I|$$

$$\min_{(u,v)\in C^2(\Omega;\mathbb{R}^2)} \int_\Omega \underbrace{g(|\nabla f|)\left(|\nabla u|^2 + |\nabla v|^2\right)}_{\text{Diffusion term}} + \underbrace{h(|\nabla f|)\left(|u - f_x|^2 + |v - f_y|^2\right)}_{\text{Edge term}}\, dx$$





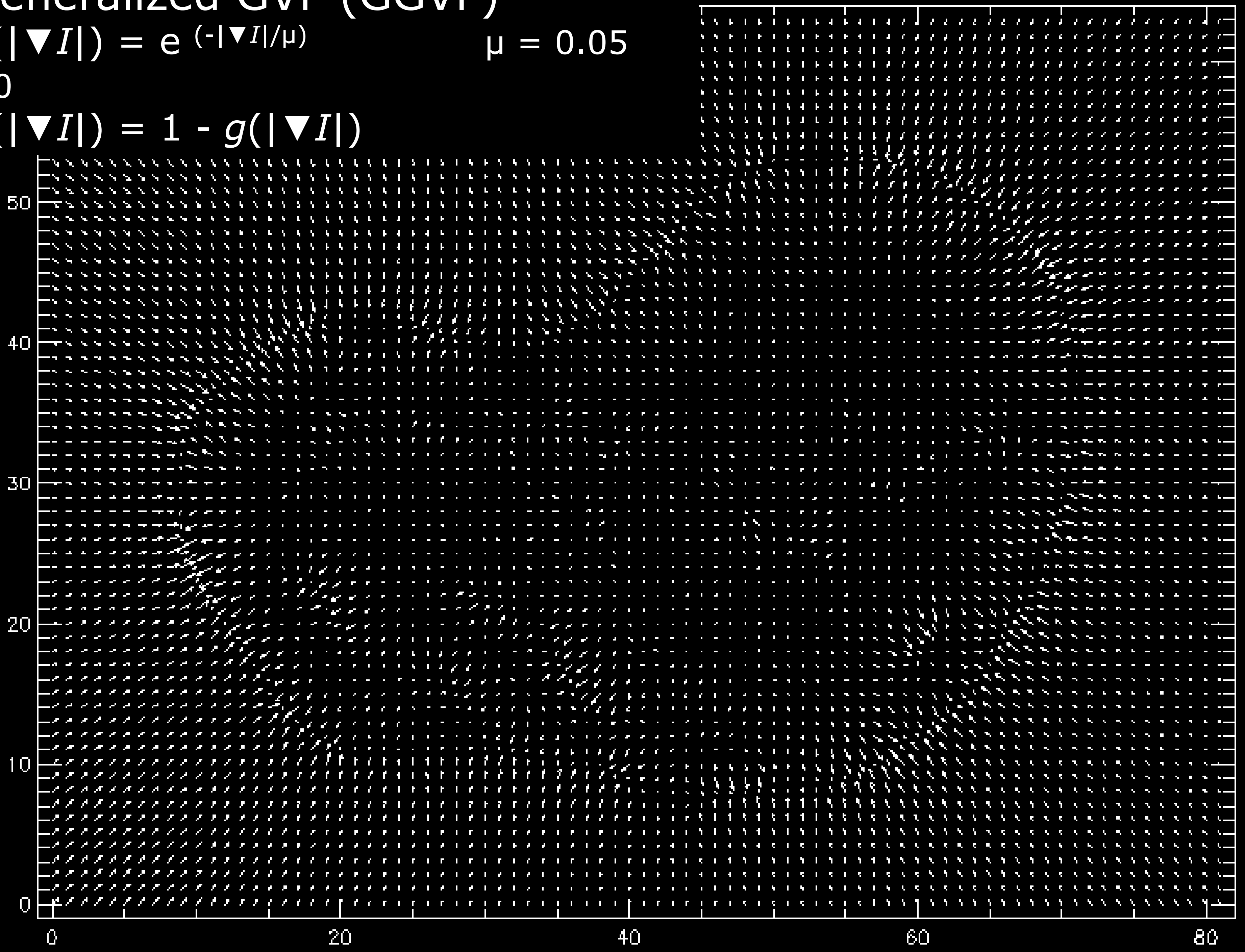

GVF
g(|∇I|) = μ   μ = 0.05 > 0
h(|∇I|) = |∇I|²
0
10
20
30
40
50
0
20
40
60
80

Generalized GVF (GGVF)

$g(|\nabla I|) = e^{(-|\nabla I|/\mu)}$ $\mu = 0.05$

>0

$h(|\nabla I|) = 1 - g(|\nabla I|)$



Snakes – formulation

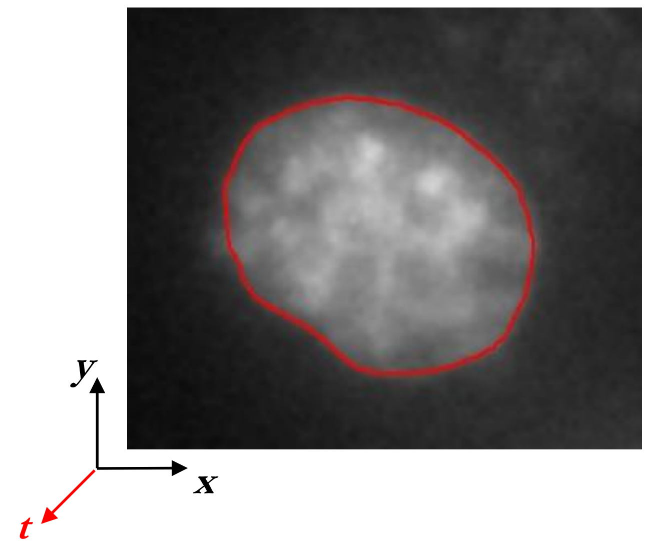
– Energy minimization condition: Euler-Lagrange equation

$$\alpha \frac{\partial^2 C(s)}{\partial s^2} - \beta \frac{\partial^4 C(s)}{\partial s^4} - \nabla E_{ext}[C(s)] = 0$$

Internal **forces** External **force**

Snakes – numerical solution

- Dynamic formulation
  - *C*(*s*) … *C*(*s*, *t*)
  - Time equilibrium leads to solution



$$\frac{\partial C(s,t)}{\partial t} = 0$$

$$\frac{\partial C(s,t)}{\partial t} = \alpha \frac{\partial^2 C(s,t)}{\partial s^2} - \beta \frac{\partial^4 C(s,t)}{\partial s^4} - \nabla E_{ext}[C(s,t)]$$

Time evolution

Internal forces

External force

Snakes – numerical solution

- Parameters
  - α (elasticity), β (rigidity)

$$\alpha \frac{\partial^2 C}{\partial s^2} - \beta \frac{\partial^4 C}{\partial s^4} - \nabla E_{ext}[C] = 0$$

$$A_{n*n} C - \nabla E_{ext}[C] = 0$$

A is a 5-diagonal matrix, suitable for numeric solution

  - κ (added as the external force weight), γ (contour viscosity)

$$\frac{\partial C(s,t)}{\partial t} = 0$$

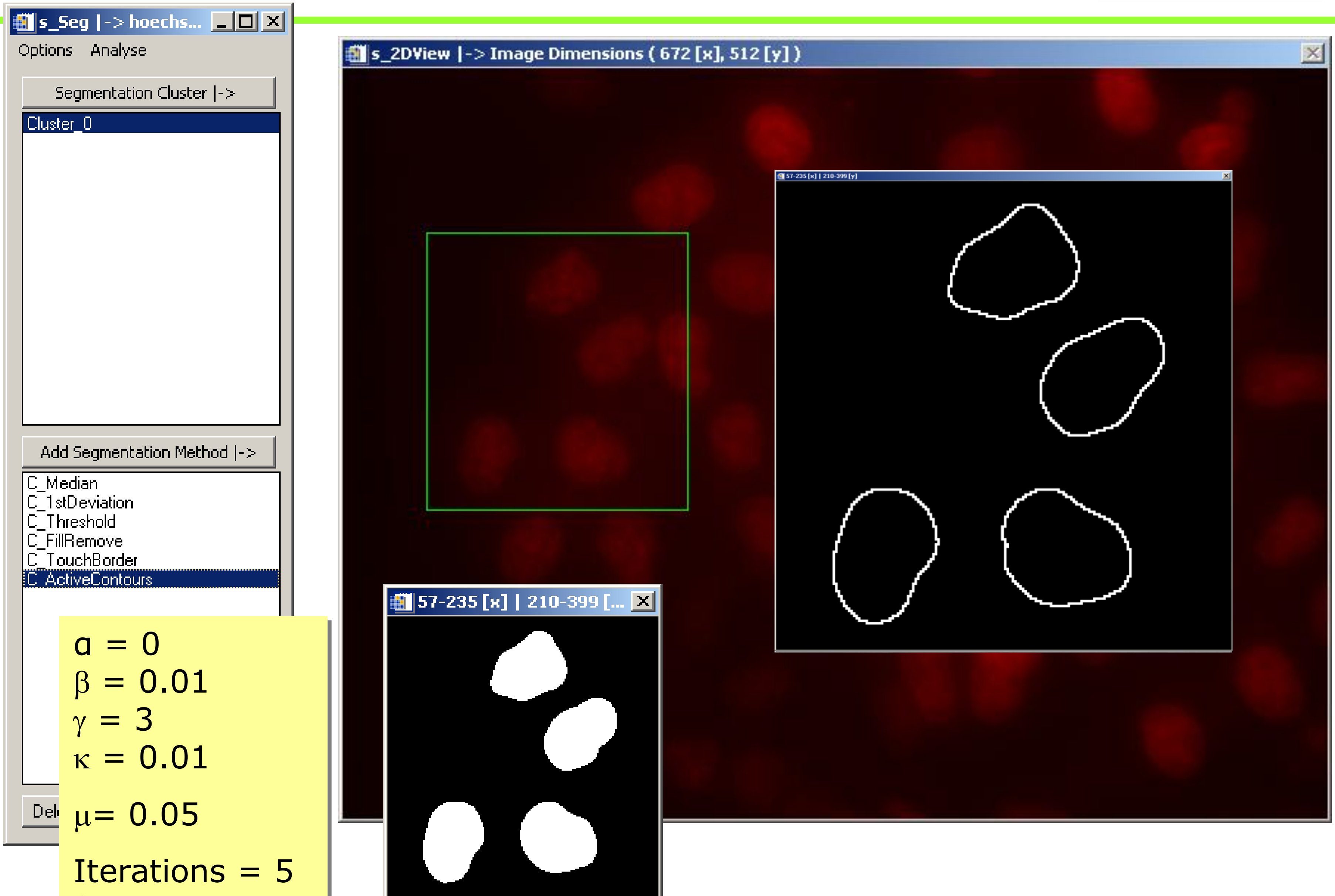
$$AC_t - \kappa \nabla E_{ext}(C_{t-1}) = -\gamma (C_t - C_{t-1})$$

Iterative scheme, solved computationally

$$C_t = (A + \gamma I)^{-1} - (C_{t-1} + \kappa \nabla E_{ext}(C_{t-1}))$$

In order to computationallly solve this system:
matrix inversion, array data structures, adaptive algorithms, …

Segmentation - Advanced

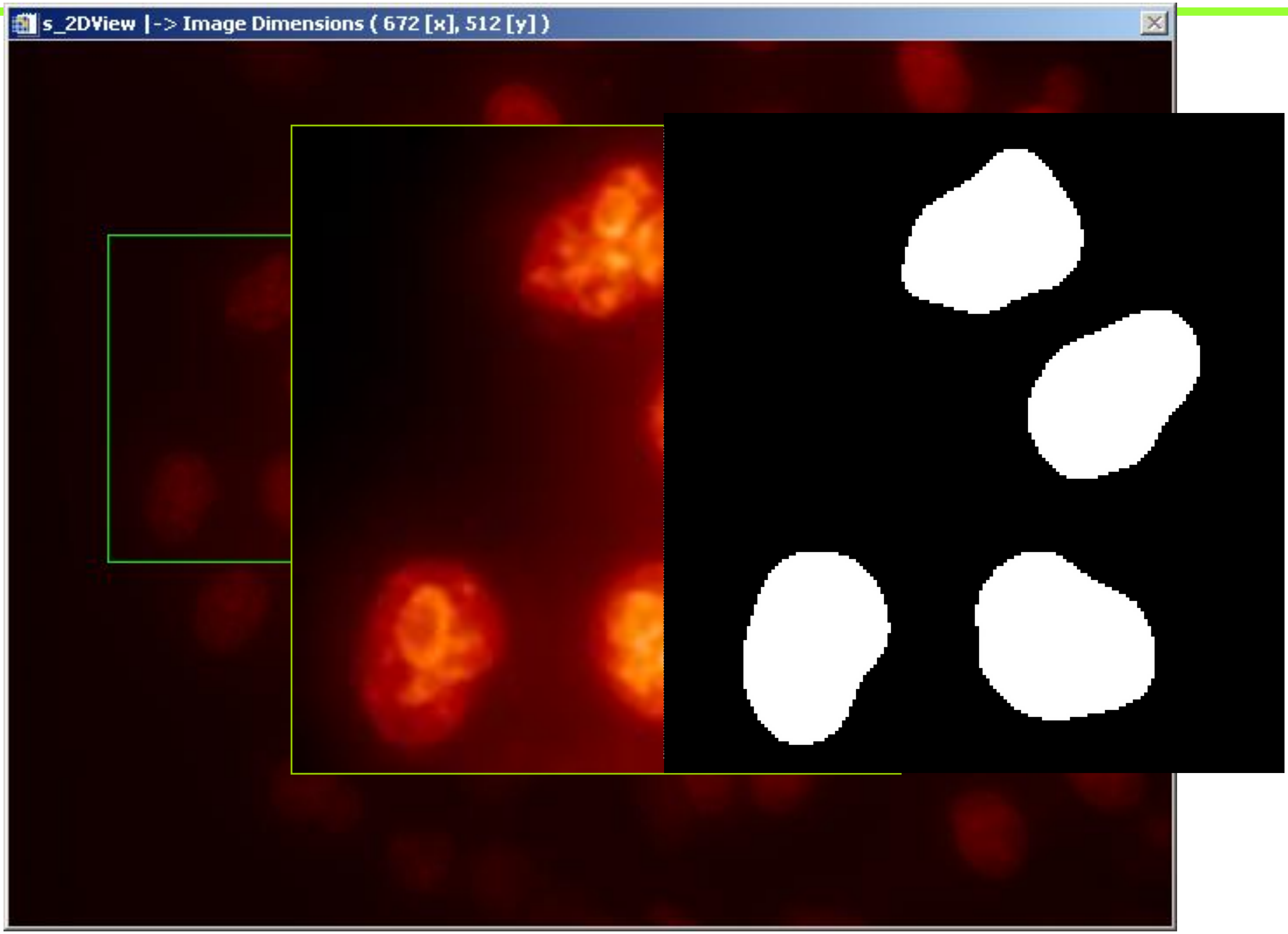


$\alpha = 0$
$\beta = 0.01$
$\gamma = 3$
$\kappa = 0.01$

$\mu = 0.05$

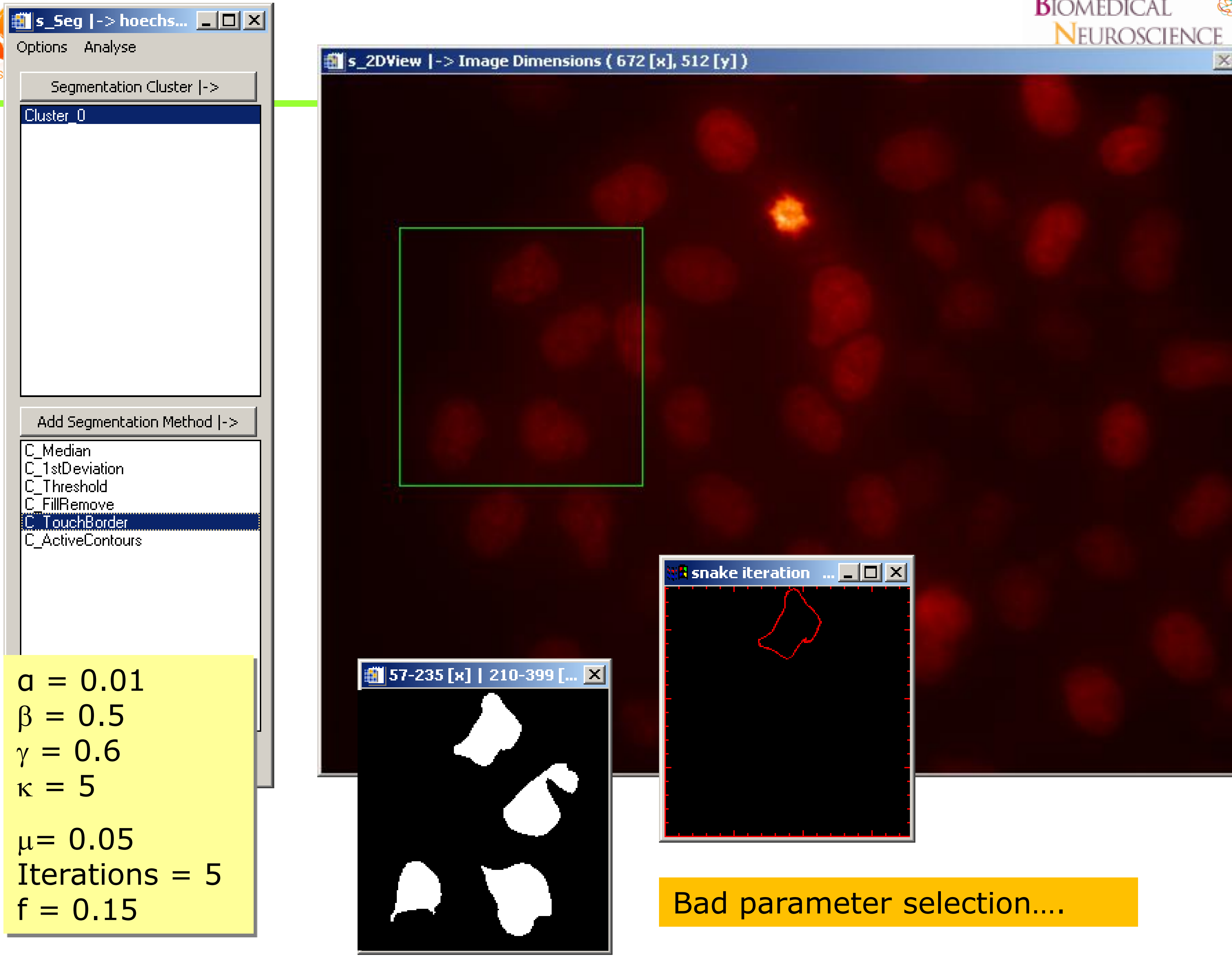
Iterations = 5
f = 0.15

Segmentation - Advanced



Segmentation - Advanced

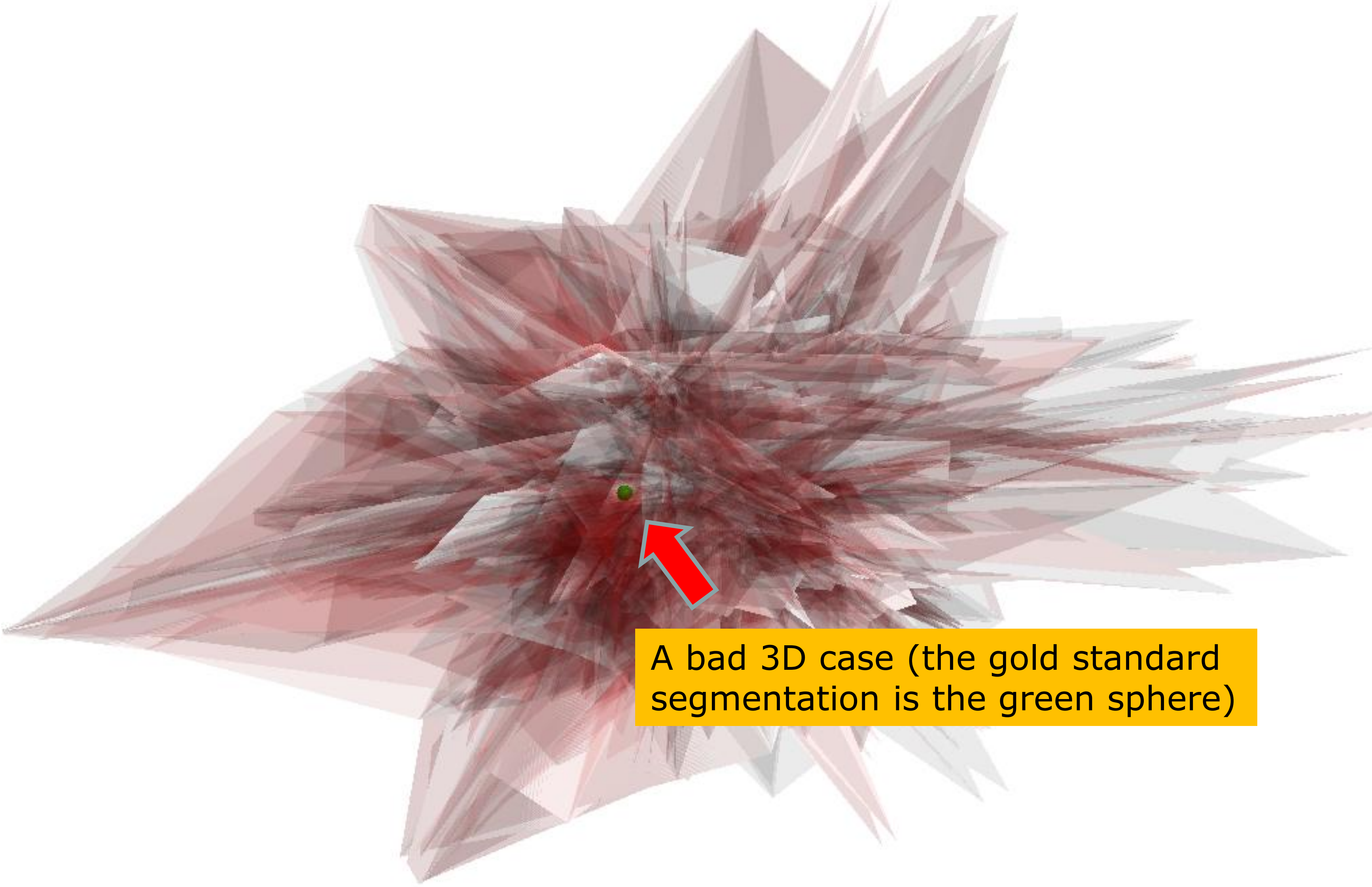
- Some problems with the snakes method
  - The quantity of contours must not change, given the 1-1 correspondence of the model
    - Dependecy on the initial guess
  - Local optima for the functional minimization
    - A bad initialization of the snakes can lead to unstability and undesired results
    - Control of the snake parameters is crucial



α = 0.01
β = 0.5
γ = 0.6
κ = 5

μ= 0.05
Iterations = 5
f = 0.15

Bad parameter selection….

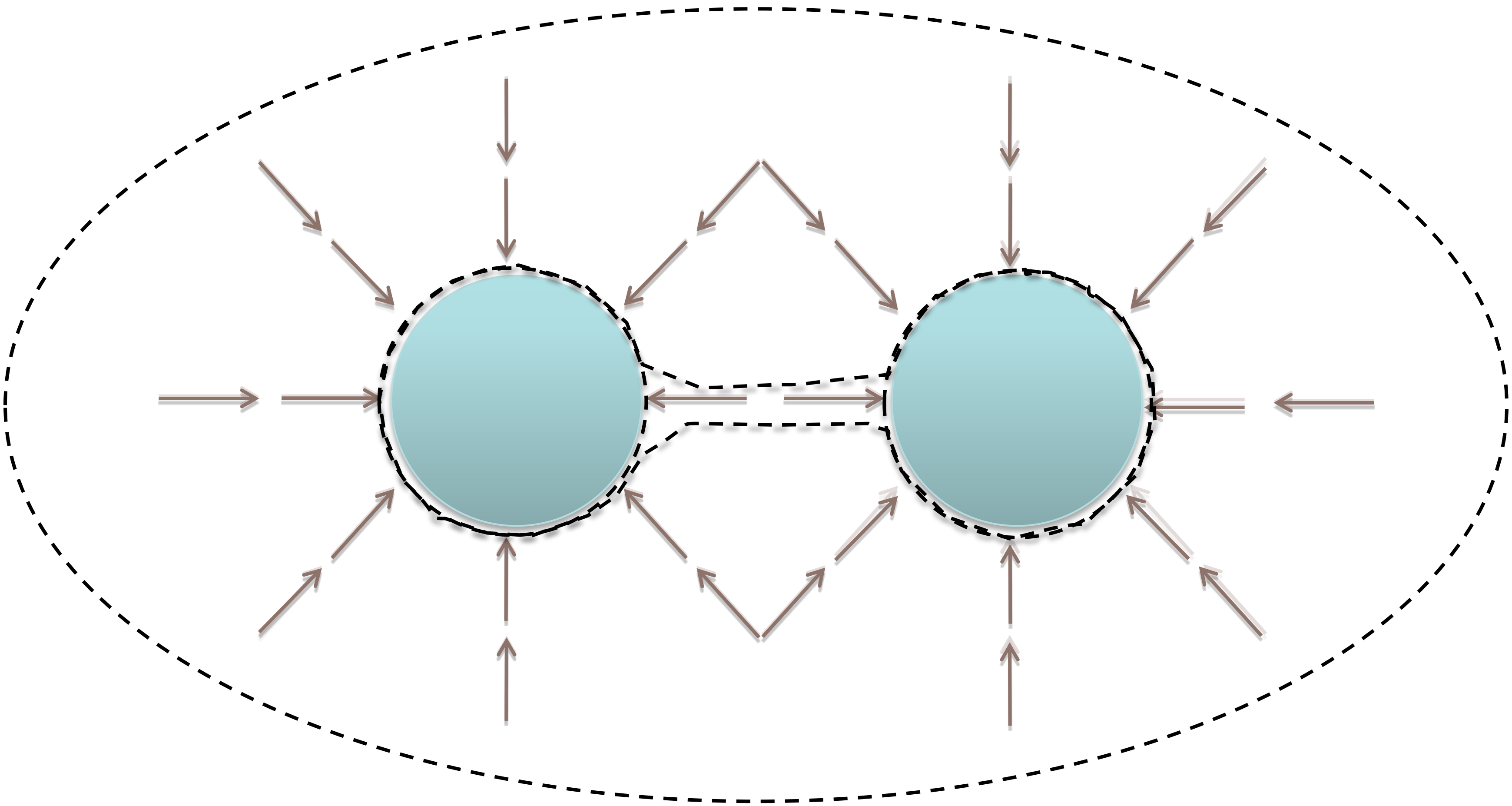

A bad 3D case (the gold standard segmentation is the green sphere)

Segmentation - Advanced

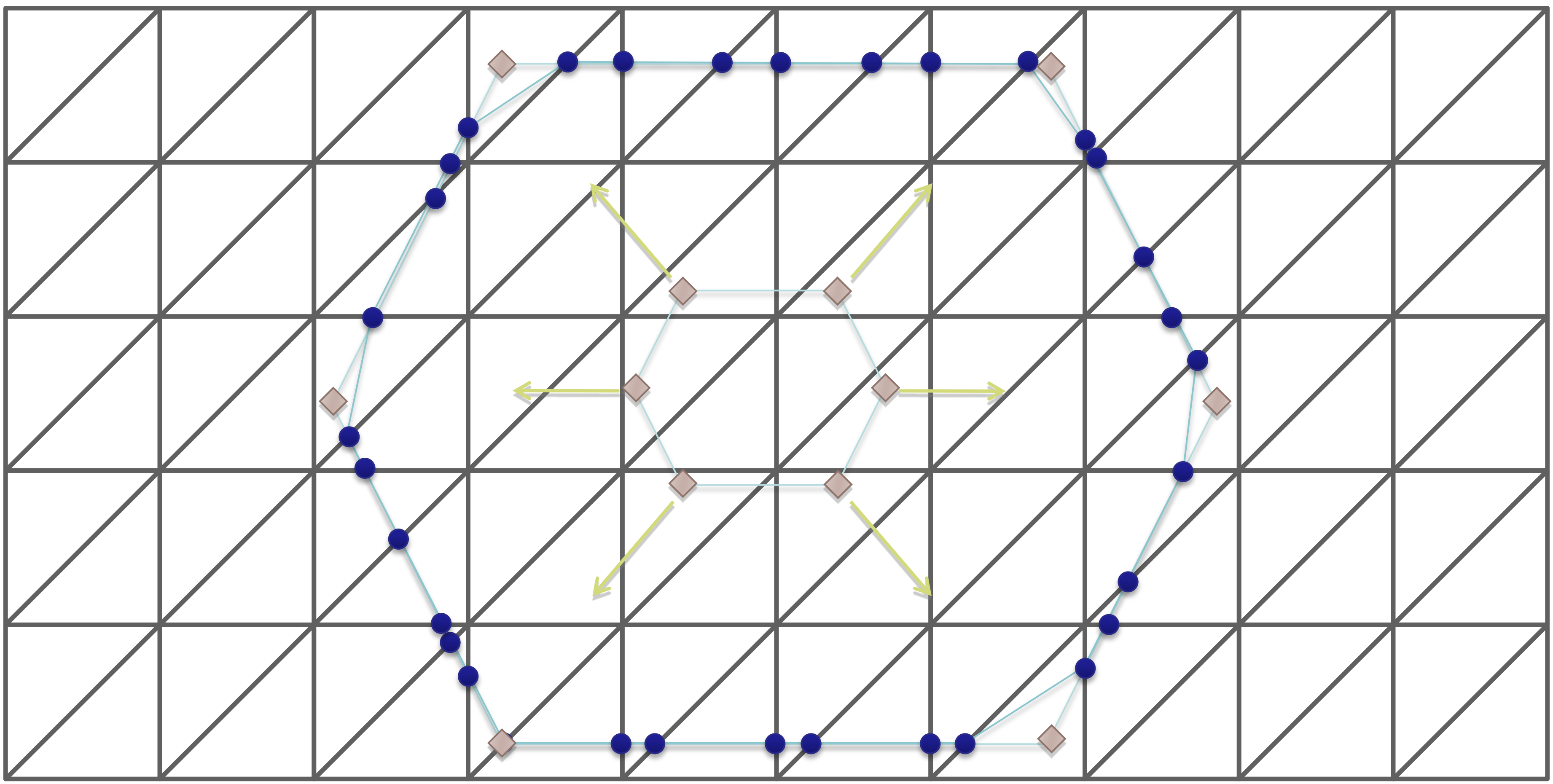
- Alternatives…
  - Arbitrary contour initialization… automation
  - Inference (machine learning approaches)

- Topology adaptive snakes (or surfaces) McInerney & Terzopoulos, 1995 (2D), 2002 (3D)

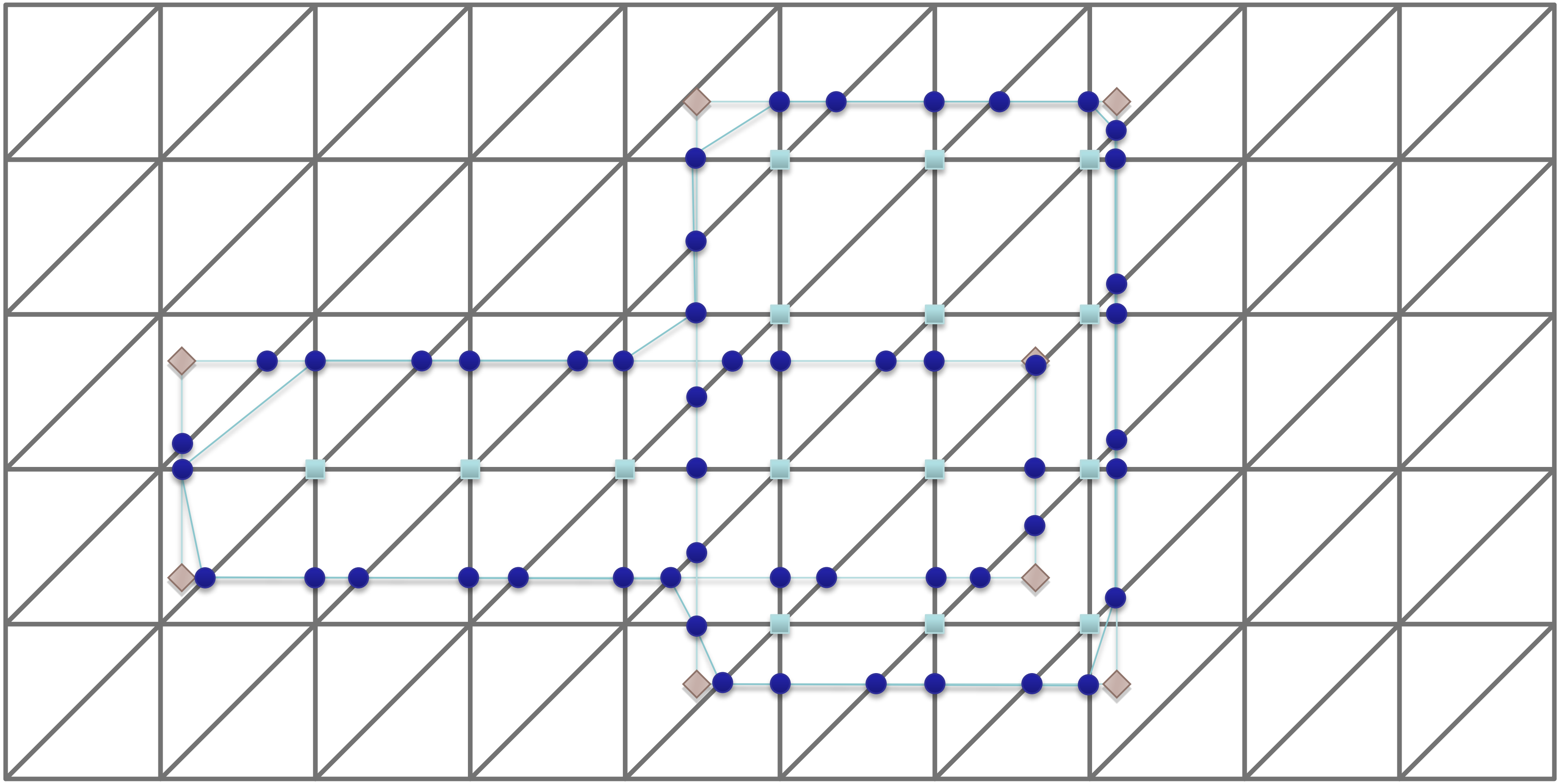
Topological changes handling: splitting, merging, collapse



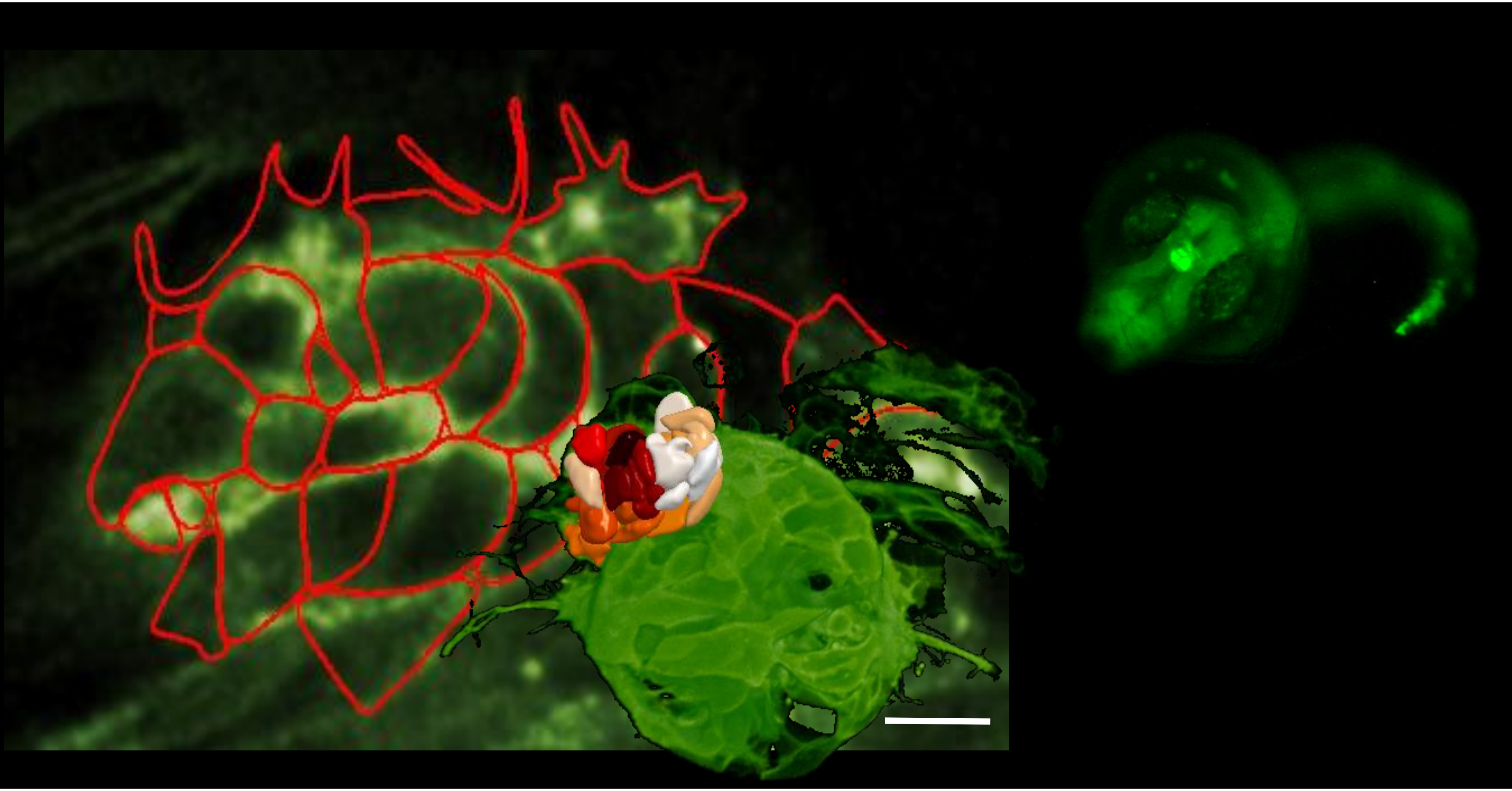
Between deformations, the contours are re-parametrized by using a grid to detect topology changes



This is an algorithmic approach, “additional” to the mathematical optimization model. The computation becomes more intensive

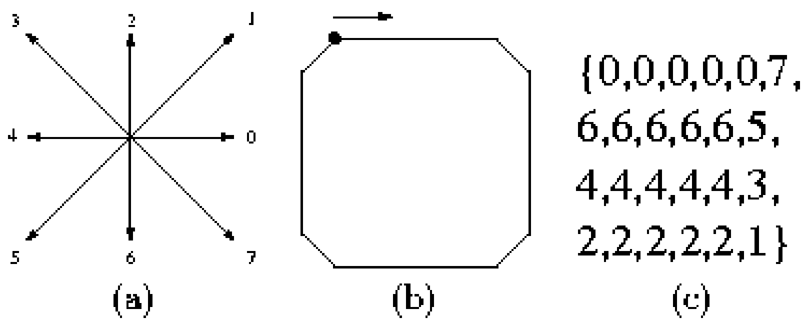


Hardest cases: initialization by manual ROI sketching

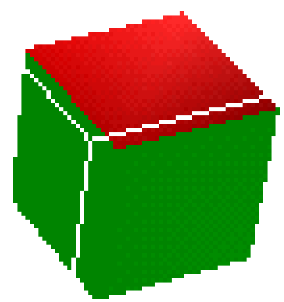


Boundary model construction

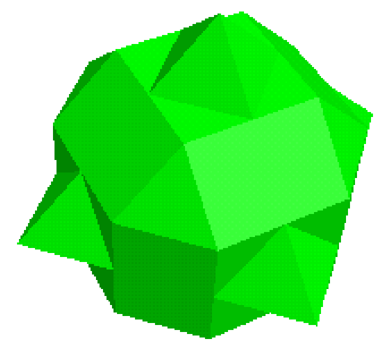
- 2D Freeman chain code
- 3D mesh models (voxel based): *marching cubes* (surface meshe<u>s</u>), *tethraedra* (volume meshes)



2D polygon chain code



Voxel ("3D pixel") model

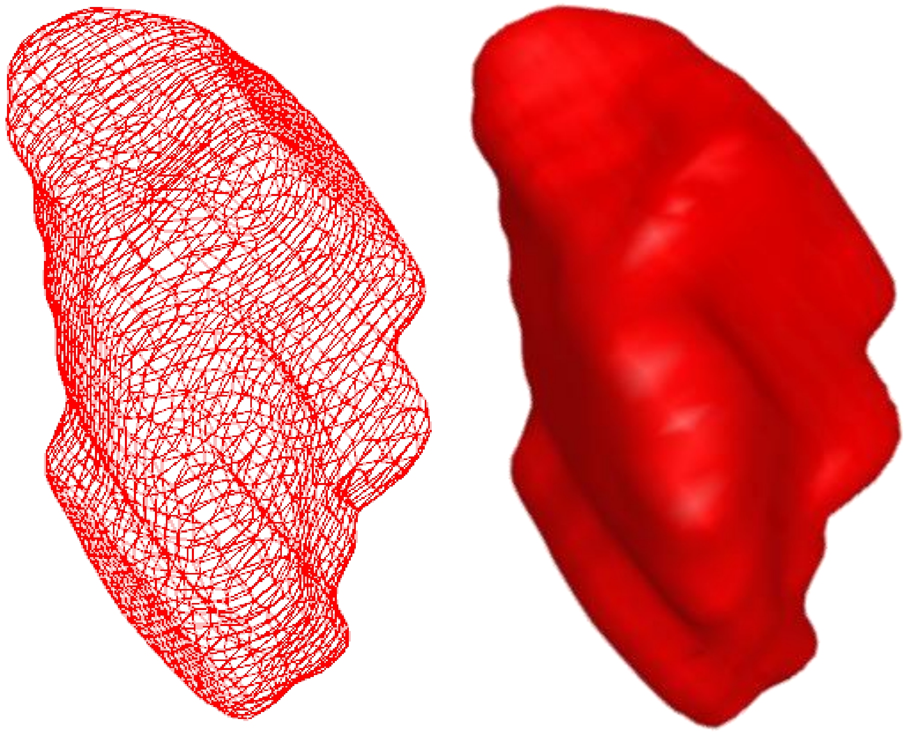


3D polygon mesh

Segmentation - Advanced

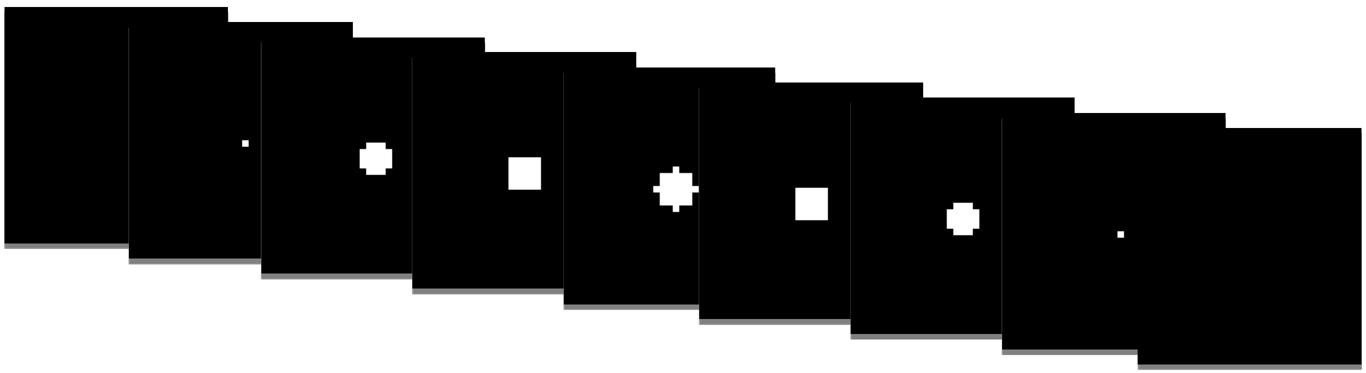
- A typical 3D surface mesh model is formed by:
  - Nodes or vertices
  - Polygons

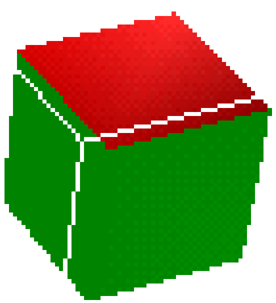
- Other models
  - Polynomial surfaces (splines, Bézier, NURBS, …)
  - Primitives composition



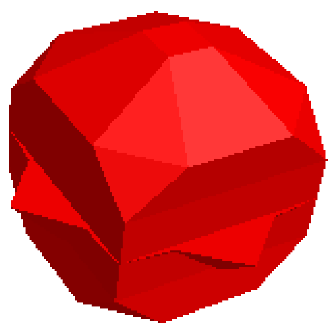
Segmentation - Advanced

- Surface mesh model construction
  - Many approaches for construction
  - Many approaches for modeling

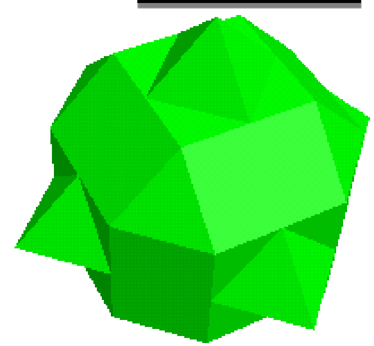




Voxel (“3D pixel”) model

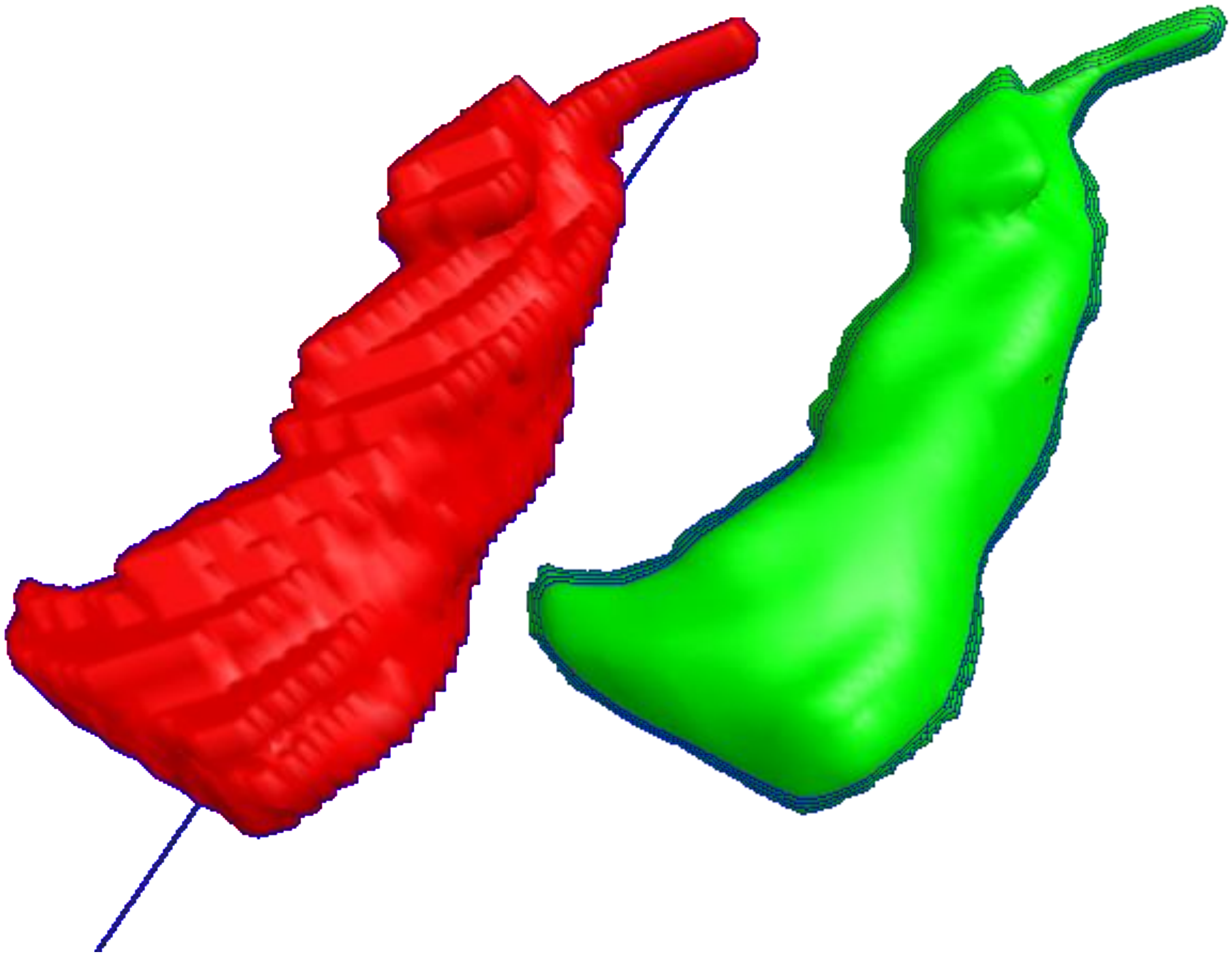


Triangle mesh



Triangular and quarilateral mesh

- Sample 3D ROI



References

- David Marr. Vision
  MIT press, 1982

- John Russ. The image processing handbook, 4th ed.
  CRC Press, 2002

- Nixon, Aguado. Feature extraction & image processing, 1st|2nd|3rd ed.
  Academic Press, 2002|2008|2012

- Aubert & Kornprobst. Mathematical Problems In Image Processing
  Springer, 2006

- From SCIAN-Lab (see publications section on the website)
  - Jara, 2006. 2D/3D active contours
  - Olmos, 2009. 2D topology adaptive active contours (t-snakes)

Laboratory for SCIAN Scientific Image Analysis

References

BIOMEDICAL NEUROSCIENCE INSTITUTE

Some free and open source software tools

- Java based (Java runtime requried)
  - ImageJ (http://rsbweb.nih.gov/ij/, public domain)
  - Fiji (http://fiji.sc; GPL license)
  - Icy (http://icy.bioimageanalysis.org; GPLv3 license)

- Others
  - CellProfiler (http://cellprofiler.org; GPL, BSD licenses)
  - Slicer (www.slicer.org; BSD license)
  - IPOL (Image Processing Online): open access electronic journal with peer reviewed articles + code (in C language) + examples (www.ipol.im; BSD / GPL / LGPL licenses or similar)