Introduction to Active Contours

Elsa Angelini, PhD

TELECOM Paris
Dpt. Image Data Signal

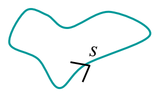
elsa.angelini@telecom-paris.fr

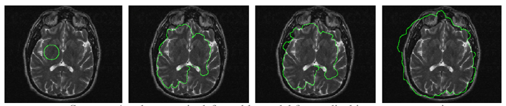
Active Contours

- Formulations:
  - Parametric
  - Geometric
  - Statistics
  - Graph-cuts
- Think about - during the lecture:
  - Implementation challenges
  - Use of image information
  - Ease of rough initialization
  - Ease of image encoding
  - Control of contour smoothness
  - Number of Hyper-parameters

Active Contours

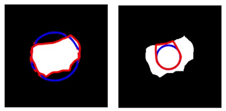
- Contour is placed in the image space and deforming towards an optimal position and shape.

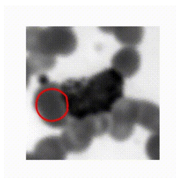




Sources: An electrostatic deformable model for medical image segmentation

- The contour moves under some forces:



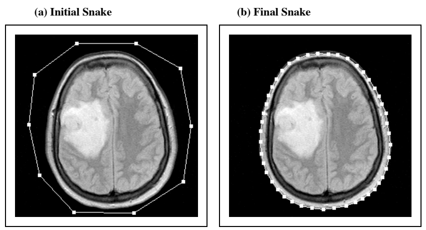


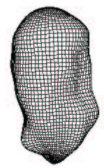
  – **Internal forces**: define intrinsic shape properties
    ⇨ preserve shape smoothness during deformation.
  – **External forces**: defined from image information.
    ⇨ attract contour towards « object borders ».

3

Active Contours

- Parametric contours.
- Geometric contours.





4

Parametric Active Contours

Formulation of the Problem

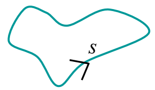
1. Energy Minimization :
   - Minimize a weighted sum of internal and external energies (Force potentials).
   - Final contour's position corresponds to an energy's minimum.

2. Dynamical Forces :
   - Equilibrium between internal and external forces at each point on the contour.

Parametric Active Contours

Definition of the Energy

- An active contour is a curve $v(s) = [x(s), y(s)]$, where $s \in [0, 1]$ is the arc length.



- $v(s)$ evolves towards a position minimizing the energy functional:

$$E_{total}(v(s)) = E_{internal}(v(s)) + E_{external}(v(s))$$

Parametric Active Contours

$E_{internal}(v(s))$:

Goal: Obtain a smooth contour

- Penalize the **size** of the object ⇨ increase the energy for high **area** and **perimeter** values.
- Penalize irregular contours ⇨ increase the energy for high contour **curvature**.
- Constrain the shape of the contour: to look like a circle, ellipse, heart shape template,...

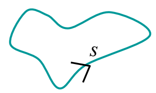
Minimization Method: via some optimization

- Finite formulation: Exhaustive search of a minimum, global or with probabilistic algorithms.
- Infinite formulation: local minimum via progressive adaptation, gradient descent or other PDE solvers.

7

Parametric Active Contours

$E_{internal}(v(s))$: **stretching & bending**



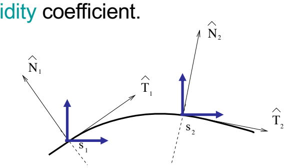
$$E_{interne} = \int_0^1 \underbrace{\alpha(s)\left(\frac{\partial v(s)}{\partial s}\right)^2}_{} + \underbrace{\beta(s)\left(\frac{\partial^2 v(s)}{\partial s^2}\right)^2}_{} ds$$

Length of the contour ⇨ tension controlled by the elasticity coefficient.

Curvature of the contour ⇨ stiffness controlled by the rigidity coefficient.

Curvature = "rate of change of **direction** of a curve with respect to the distance along the curve"

$$\kappa = \frac{d\theta}{ds}, \tan\theta = \frac{dy}{dx}, ds = \sqrt{x' + y'}$$



8

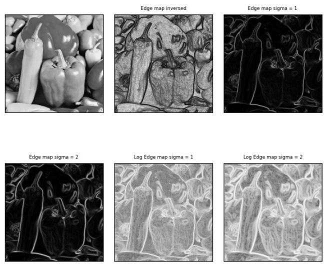
Parametric Active Contours

$E_{external}(v(s))$:

- **Standard formulation**: Integral of a force potential:

$$E_{externe} = \int_0^1 P(v(s))ds$$

- **Potential** = low values on the contours in the image (e.g.: derived from image gradient)



$$P(x,y) = -w\left|\nabla I(x,y)\right|^2$$

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

9

Parametric Active Contours

Energy Minimization

- **Goal**: find the contour *v(s)* that minimizes the global energy.
- **Framework**: Attract an initial contour towards "contours" in the image, while avoiding stretching and bending.

- **Static version**: Variational problem formulated with the **Euler-Lagrange** equation:

[Computation involves integration by parts + limit of infenitesimal steps]

Calculus of variations + gradient descent:

$$E(\mathbf{x}+\delta\mathbf{x}) \approx E(\mathbf{x}) + \frac{\partial E}{\partial \mathbf{x}} \cdot \delta\mathbf{x}$$

$$\delta\mathbf{x} \propto -\frac{\partial E}{\partial \mathbf{x}}$$

$$\Downarrow$$

$$E(\mathbf{x}+\delta\mathbf{x}) \approx E(\mathbf{x}) - \tau\left(\frac{\partial E}{\partial \mathbf{x}}\right)^2$$

➔ At *v(s)* optimal we have:

$$\frac{\partial}{\partial s}\left(\alpha\frac{\partial v(s)}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 v(s)}{\partial s^2}\right) - \nabla P(v) = 0$$

<u>**Forces**</u> applied to the contour

+

➔ Provides a local view of the solution and does not indicate "how" to move *v(s)*..

10

Parametric Active Contoursy

Energy Minimization

- **Dynamic version ➔ Dynamic contour:**
  - Build a dynamic system that we evolve towards an equilibrium state according to a Lagrangian mechanical point of view.
  - This dynamical model unifies the shape and motion descriptions, defining an active contour⇨ quantification of a shape evolution through time $v(s,t)$.

➔ Motion equation: according to the 2nd law of Newton:

$$\mu\frac{\partial^2 v}{\partial t^2}+\gamma\frac{\partial v}{\partial t}-\frac{\partial}{\partial s}\left(\alpha\frac{\partial v}{\partial s}\right)+\frac{\partial^2}{\partial s^2}\left(\beta\frac{\partial^2 v(s)}{\partial s^2}\right)+\nabla P(v)$$

➔ Equilibrium state defined as: $\frac{\partial^2 v}{\partial t^2}=\frac{\partial v}{\partial t}=0$

11

Parametric Active Contours

Energy Minimization

Methods 1 & 2: no analytical solution (due to the external energy measured at the contour location).

⇨ Need to discretize and approximate the problem:

- Finite Differences: each element of the contour is viewed as a point with individual mechanical properties.
- Finite Elements : sub-elements between nodes.

- N control points $v = (v_1, v_2, \ldots, v_N)$, distant with a spatial step h.

12

Parametric Active Contours

Energy Minimization: Numerical Schemes

- Discretization of spatial derivatives:

$$\frac{\partial^2 v_i}{\partial s^2} = \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} \qquad \frac{\partial^4 v_i}{\partial s^4} = \frac{v_{i-2} - 4v_{i-1} + 6v_i - 4v_{i+1} + v_{i+2}}{h^4}$$

- Matrix Notation (α, β cst):

$$\alpha \frac{\partial^2 v(s)}{\partial s^2} - \beta \frac{\partial^4 v(s)}{\partial s^4} - \nabla P(v) = 0$$

[Computatioal trick involved, estimating spatial derivatives at next step prior to external forces]

➔ Penta diagonal matrix

$$\Rightarrow Av = \nabla P(v)$$

$$A = \frac{1}{h^2}\begin{bmatrix} -2\alpha - 6\frac{\beta}{h^2} & \alpha + 4\frac{\beta}{h^2} & -\frac{\beta}{h^2} & 0 & \cdots & \cdots \\ \alpha + 4\frac{\beta}{h^2} & -2\alpha - 6\frac{\beta}{h^2} & \alpha + 4\frac{\beta}{h^2} & -\frac{\beta}{h^2} & 0 & \cdots \\ -\frac{\beta}{h^2} & \alpha + 4\frac{\beta}{h^2} & \ddots & \ddots & \ddots & \cdots \\ 0 & -\frac{\beta}{h^2} & \ddots & \ddots & \ddots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \cdots & \cdots \end{bmatrix}$$

[depends on h, needs regular reinitialisation]

Kass, 1987

13

Parametric Active Contours

Energy Minimization: Numerical Schemes

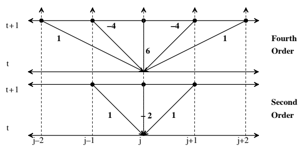
Internal forces

$$v = (x, y), \begin{cases} Ax = \nabla P_x(x, y) \\ Ay = \nabla P_y(x, y) \end{cases}$$

➔ **Used in an iterative numerical scheme:**

Inertia coefficient

$$\begin{cases} Ax^n - \nabla P_x(x^{n-1}, y^{n-1}) = -\gamma(x^n - x^{n-1}) \\ Ay^n - \nabla P_y(x^{n-1}, y^{n-1}) = -\gamma(y^n - y^{n-1}) \end{cases}$$



14

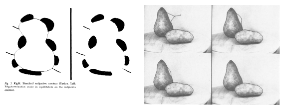
Parametric Active Contours

Energy Minimization: Hyperparameters to control during the TP

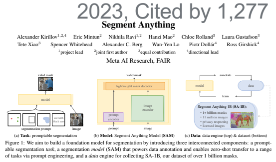
- Spatial step versus pixel size for continuity:
  - α = elasticity
  - β = rigidity
- Temporal step set to control the maximum displacement at each iteration:
  - γ = viscosity ~1/dt

Snakes: Active Contour Models 1988, Cited by 26,611

MICHAEL KASS, ANDREW WITKIN, and DEMETRI TERZOPOULOS
Schlumberger Palo Alto Research, 3340 Hillview Ave., Palo Alto, CA 94304



2023, Cited by 1,277



Parametric Active Contours

Energy Minimization: Conclusions

Advantages

- Extraction of a locally optimal position via iterative deformations of a curve.
- Suited for multiple types of contour extractions:
  - Open curve
  - Close curve (force $v_0 = v_N$)
  - Curve with fixed extremities (v_0 or v_N fixed)

- General Framework: several different types exist
- Simple an efficient 2D implementation
- Numerical stability wrt **internal** forces

Parametric Active Contours

Energy Minimization: Conclusions

Limitations

- Instability wrt **external** forces: if spatial step is too big, can miss some contours.
- Sensitive to local minima problems and to initialization.
- Challenging parameterization.
- No change in topology allowed (i.e. division/fusion of objects).
- No simultaneous deformation of multiple objects (complex extension).

$$E_{\text{mag}} = -\int_0^M |\nabla f(t)|^2 \, dt$$

$$\Downarrow$$

$$E_{\text{grad}} = -\oint_C \mathbf{k} \cdot (\nabla f(\mathbf{r}) \times d\mathbf{r}) = -\oint_C \nabla f(\mathbf{r}) \cdot \underbrace{(d\mathbf{r} \times \mathbf{k})}_{\|d\mathbf{r}\|\hat{\mathbf{n}}(\mathbf{r})}$$

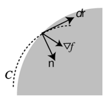


Fig. 2. Gradient and normal to the curve.

Efficient Energies and Algorithms for Parametric Snakes

Mathews Jacob, *Member, IEEE*, Thierry Blu, *Member, IEEE*, and Michael Unser, *Fellow, IEEE*

The widely-used gradient magnitude-based energy is parameter dependent; its use will negatively affect the parametrization of the curve and, consequently, its stiffness. Hence, we introduce a new edge-based energy that is independent of the parameterization. It is also more robust since it takes into account the gradient direction as well.

Parametric Active Contours

Formulation of the Problem

1. Energy Minimization :
   - Minimize the weighted sum of internal and external energies (Force potentials).
   - Final contour position corresponds to an energy's minimum.
2. Dynamical Forces :
   - Equilibrium between internal and external forces at each point on the contour.

Parametric Active Contours

Formulation with Dynamical Forces

- Dynamical problem with more general forces than potential forces
- Newton's law :

$$\mu \frac{\partial^2 C}{\partial t^2} = F_{\text{internal}}(C) + F_{\text{external}}(C) + F_{\text{viscous}}(C)$$

mass

$$-\gamma \frac{\partial C}{\partial t}$$

Parametric Active Contours

Formulation with Dynamical Forces

- Simplification: no mass

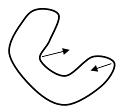
$$\gamma \frac{\partial C}{\partial t} = F_{\text{internal}}(C) + F_{\text{external}}(C)$$

idem

Superposition
of forces

Parametric Active Contours

$F_{external}$ **:GRADIENT** forces



- **Properties** of the Gradient Vectors:
  - Point towards the contours (normals).
  - Large norms near edges.
  - Norm~0 in homogeneous regions.
- **Problems**:
  - Weak attraction range (only close to edges)
  - No force in homogeneous areas (nothing moves…).
- **To solve :**
  - Initialization problems.
  - Convergence towards concave regions.

21

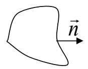
Parametric Active Contours

Example $F_{external}$: Balloon force [Cohen & Cohen] :

- Keep gradient force to attract the contours towards edges.
- Add a pressure force to constrain the model to inflate/deflate:

+/-: inflate

$$F_{\text{externe}}(C) = k\frac{\vec{\nabla}P(C)}{\left|\nabla P(C)\right|} - k_1\vec{n}(s)$$



- Computational cost: image gradient norms & normals along the contours at each node.
- Need to control the dynamical behavior of the contour far from the edges (via weight of k_1).

22

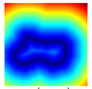
Parametric Active Contours

Example $F_{external}$: Potential Forces for Distances

- Compute the distance map *D(x, y)* (e.g. Euclidian or Chamfer) for each pixel to the closest point on the contour ⇨ field of potential forces.
- *D(x, y)* defines the potential energy… :
- … and the field of forces

$$P_{\text{distance}}\left(x,y\right)=we^{-D(x,y)^2}$$

$$F_{\text{external}}\left(C\right)=-\nabla P_{\text{distance}}\left(x,y\right)$$

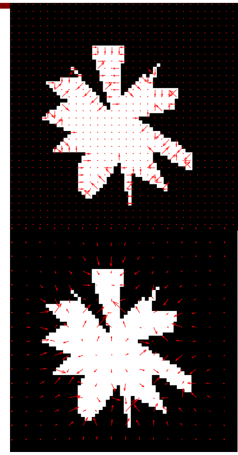


$D\left(x,y\right)$

Parametric Active Contours

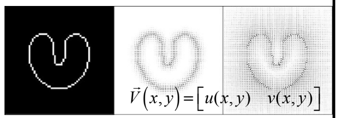
Example of $F_{external}$: Gradient vector flow GVF [Xu & Prince]

- Vector field.
- Preserve gradient properties near the edges.
- Diffuse these properties in homogeneous regions via « gradient diffusion ».



Parametric Active Contours

Example of $F_{external}$: GVF



- GVF is a vector field: $\vec{V}(x,y)=\left[u(x,y) \quad v(x,y)\right]$
- $\vec{V}(x,y)$ is defined via an energy minimization:

$$E=\int_{\Omega}\left[\begin{array}{l}\overbrace{\mu\left(u_x^2(x,y)+u_y^2(x,y)+v_x^2(x,y)+v_y^2(x,y)\right)}^{\text{Regularization = “be smooth”}}\\ \underbrace{+|\vec{\nabla}\mathrm{I}_{\text{edge}}(x,y)|^2}_{\text{Data weight}}\underbrace{|\vec{V}(x,y)-\vec{\nabla}\mathrm{I}_{\text{edge}}(x,y)|^2}_{\text{Data term = « look like the gradient »}}\end{array}\right]dxdy$$

$\mathrm{I}_{\text{edge}}(x,y)$ = Edge map of the image $I(x,y)$

Parametric Active Contours

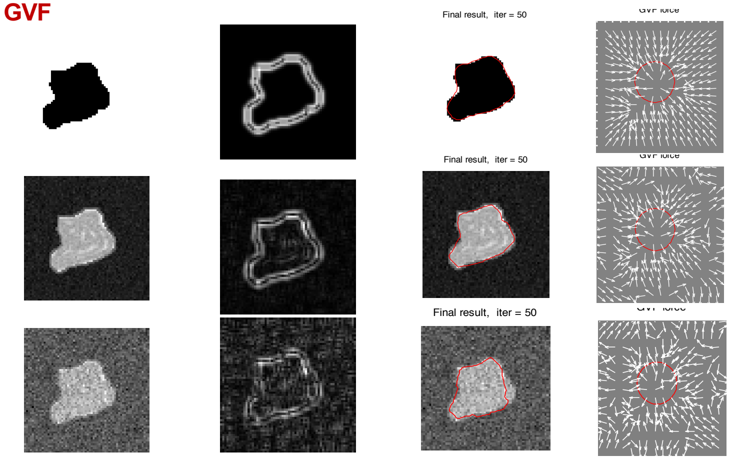
Example of $F_{externe}$: GVF

- The GVF vector field is obtained by solving the Euler equations :

$$\begin{cases}\underbrace{\mu\Delta(u(x,y))}_{\text{Laplace Eq.}}-\underbrace{\left(u(x,y)-\dfrac{\partial I_{\text{edge}}(x,y)}{\partial x}\right)}_{\text{Gradient data term}}\underbrace{\left(\dfrac{\partial I_{\text{edge}}(x,y)}{\partial x}^2+\dfrac{\partial I_{\text{edge}}(x,y)}{\partial y}^2\right)}_{\text{Weight = Edge map}}=0\\ \mu\Delta(v(x,y))-\left(v(x,y)-\dfrac{\partial I_{\text{edge}}(x,y)}{\partial y}\right)\left(\dfrac{\partial I_{\text{edge}}(x,y)}{\partial x}^2+\dfrac{\partial I_{\text{edge}}(x,y)}{\partial y}^2\right)=0\end{cases}$$

Parametric Active Contours

GVF



Parametric Active Contours

Bibliography

1. **Kass** M, Witkin A and **Terzopoulos** D. « Snakes: Active contour models », International Journal of Computer Vision, 1987.
2. **Cohen** LD and Cohen I. “Finite-elements methods for active contour models and balloons for 2-D and 3-D Images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993.
3. Xu C and **Prince** JL. “Snakes, shapes and gradient vector Flow”, IEEE Transactions on Image Processing, 1998.

Parametric Active Contours

TIP-SEEKING ACTIVE CONTOURS FOR BIOIMAGE SEGMENTATION

Virginie Uhlmann, Michael Unser**

*Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), Switzerland

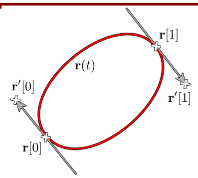


Fig. 1: Spline-snake model with tangential control. The continuous curve $\mathbf{r}(t)$ is defined by a collection of anchor points $\mathbf{r}[k]$ and their associated tangent controls $\mathbf{r}'[k]$.

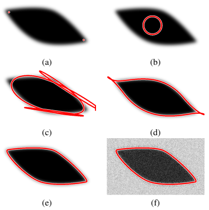
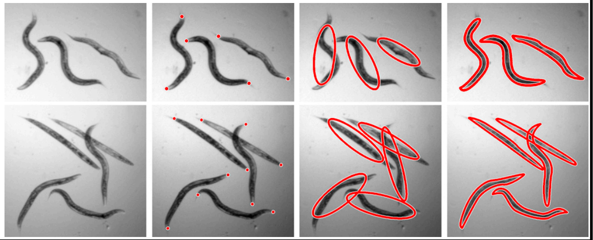


Fig. 2: Examples of synthetic image. (a) Detected features locations in red; (b) 4-points snake initialization. Segmentation results after optimizing on (c) edge-based energy only; (d) combination of edge- and constraint-based [7] energies; (e) combination of (2) and edge-based energy; (f) Results using a combination of (2) and edge-based energy on the image degraded by additive Gaussian noise.



29

Parametric Active Contours

Learning deep structured active contours end-to-end

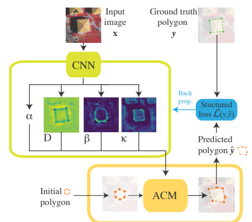
Diego Marcos, Devis Tuia, Benjamin Kellenberger
University of Wageningen, Netherlands

Lisa Zhang, Min Bai, Renjie Liao, Raquel Urtasun
University of Toronto, Canada

To this end, we present **Deep Structured Active Contours** (DSAC), a novel framework that integrates priors and constraints into the segmentation process, such as continuous boundaries, smooth edges, and sharp cor- ners. To do so, DSAC employs Active Contour Models (ACM), a family of constraint- and prior-based polygonal models.

An active contour [17] can be represented as a polygon $\mathbf{y} = (\mathbf{u}, \mathbf{v})$ with L nodes $\mathbf{y}_s = (u_s, v_s) \in \mathbb{R}^2$, with $s \in 1 \ldots L$, where each s represents one of the nodes of the discretized contour. The polygon $\mathbf{y}$ is then deformed such that the following energy function is minimized:

$$E(\mathbf{y}, \mathbf{x}) = \sum_{s=1}^{L} \left[D\big(\mathbf{x}, (\mathbf{y}_s)\big) + \alpha\big(\mathbf{x}, (\mathbf{y}_s)\big) \left| \frac{\partial \mathbf{y}}{\partial s} \right|^2 + \beta\big(\mathbf{x}, (\mathbf{y}_s)\big) \left| \frac{\partial^2 \mathbf{y}}{\partial s^2} \right|^2 \right] + \sum_{u,v \in \Omega(\mathbf{y})} \kappa\big(\mathbf{x}, (u, v)\big), \quad (1)$$



Note that i) DSAC does not depend on any particular ACM inference algorithm, and ii) the chosen ACM algorithm does not need to be differentiable.

$$\mathbf{y}^{t+1} = \mathbf{y}^t - \frac{dE_{ext}}{d\mathbf{y}^t} - (A+B)\mathbf{y}^{t+1}.$$

If we solve this expression for $\mathbf{y}^{t+1}$, we obtain:

$$\mathbf{y}^{t+1} = (I + A + B)^{-1}\Big(\mathbf{y}^t - \frac{dE_{ext}}{d\mathbf{y}^t}\Big).$$

We have implemented the described locally penalized ACM using a Tensorflow graph. The typical inference time is under 50 ms on a single CPU for the settings used in this paper.

30

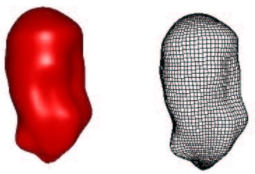
Parametric Active Contours

Introduction to the upcoming Lab-work (TP):

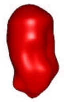
1. Check the code
2. Design your own synthetic shape model(s): can use closed and opened contours.
3. Think of how to adjust gradient map computations (eg. log transform, pre or post smoothing)
4. Think of modes to change initial shape (ellipse versus circle, add imperfactions to the circle)

Active Contours

- Parametric contours.
- Geometric contours.



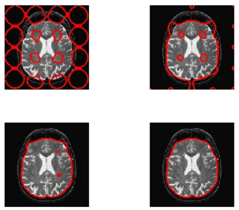
Geometric Active Contours



Introduction

- Theory of curve evolution and geometrical flows.
- The contour deforms with a **speed** made of 2 terms:
  - Regularizing term (curvature-based motion).
  - Expansion term [or contraction] to go towards image edges.
- The active contour is defined via a geometrical flow (PDE).

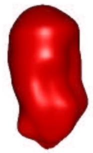
⇨ the curve evolution must stop at locations of image edges corresponding to the object to segment.



33

Geometric Active Contours

- Geometric Active contours
  - Numerical methods via level sets.
  - Geodesic.
  - Mumford-Shah.



34

Geometric Active Contours

Curve Evolution Theory:

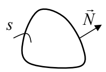
Curve evolution through geometric measures (normal vectors to the curve, curvature, ...) and independent of curve parameterization (e.g. derivatives).

- Let a curve $\vec{X}(s,t)=\left[x(s,t) \quad y(s,t)\right]$ be defined with spatial parameter *s* and temporal parameter *t*.
- The curve evolution in the normal directions is controlled by this PDE:

Propagation speed

Geometric measure on the curve

$$\frac{\partial \vec{X}(s,t)}{\partial t} = V(\kappa)\vec{N}$$



Geometric Active Contours

Curve Evolution Theory

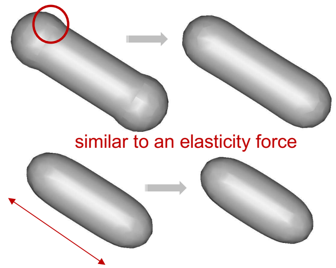
1. Constant speed:

$$\frac{\partial \vec{X}(s,t)}{\partial t} = V_0\vec{N}$$

similar to a pression force (balloon).

2. Motion under curvature:

$$\frac{\partial \vec{X}(s,t)}{\partial t} = \alpha\kappa\,\vec{N} \quad , \alpha > 0$$



Geometric Active Contours

Numerical Method ?

- **Goal:**
  - Numerical methods to compute the spatial propagation of a curve in time: add a temporal dimension.
  - Precise characterization of the geometric properties of the contour.
- **Approach**:
  - Define a spatio-temporal function Φ(x,y,t) on the image domain Ω.
  - Define the "contour" Γ we deform as the 0-level of Φ (iso-contours).
  - Extend values to define Φ over the whole image domain Ω.

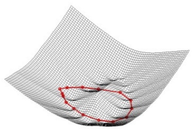
Defined on image domain Ω and time

+/- values

$$\phi : \Omega \times \mathbb{R}^{+} \rightarrow \mathbb{R}$$

Level Sets
[Osher – Sethian]

$$\begin{cases} \phi\left(\vec{\Gamma}_0, 0\right) = 0 & \text{Initial contour} \\ \phi\left(\vec{\Gamma}, t\right) = 0 & \text{Contour at time t} \end{cases}$$



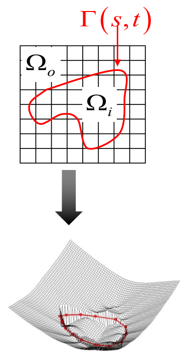
37

Geometric Active Contours

Numerical Methods with Level Sets

Definition of the level set function:

$$\begin{cases} \phi(x,y,t) > 0, (x,y) \in \Omega_i \\ \phi(x,y,t) < 0, (x,y) \in \Omega_o \\ \phi(x,y,t) = 0, (x,y) \in \Gamma(s,t) \end{cases}$$



- Γ(s,t) is defined as the 0 level of Φ(x,y,t).
- Γ(s,t) deforms with a speed v applied on each point.

⇨ How to control the level set motion?

38

Geometric Active Contours

Numerical Methods with Level Sets

Iterative Deformation Scheme:

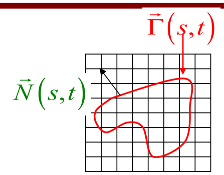
1. Define a field of speed vectors (cf. theory of curve evolution).
2. Compute initial values of the level set function, based on the initial position of the contour to evolve.
3. Adjust the function in time, so that the level 0 corresponds to a satisfactory solution of the segmentation problem.

⇨ Evolution equation for the level set function?

Geometric Active Contours

Numerical Methods with Level Sets

Evolution equation of the level set function:



$$\phi(\vec{\Gamma},t) = cste \Rightarrow \frac{d\phi(\vec{\Gamma},t)}{dt} = 0$$

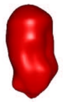
$$\Rightarrow \frac{\partial\phi(\vec{\Gamma},t)}{\partial t} + \vec{\nabla}\phi(\vec{\Gamma},t).\frac{\partial\vec{\Gamma}}{\partial t} = 0$$

Curve evolution theory: $\frac{\partial\vec{\Gamma}}{\partial t} = V(\kappa)\vec{N}$

$$\frac{\partial\phi(\vec{\Gamma},t)}{\partial t} + V(\kappa)\vec{\nabla}\phi(\vec{\Gamma},t).\vec{N} = 0$$

$$\vec{N} = \frac{\vec{\nabla}\phi(\vec{\Gamma},t)}{\left\|\vec{\nabla}\phi(\vec{\Gamma},t)\right\|}$$

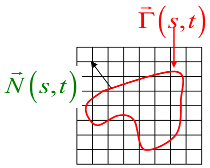
Geometric Active Contours



Numerical Methods with Level Sets

Evolution equation of the level set function:

$$\begin{cases} \dfrac{\partial \phi(\vec{\Gamma},t)}{\partial t} + V(\kappa)\left\|\vec{\nabla}\phi(\vec{\Gamma},t)\right\| = 0 \\ \phi(\vec{\Gamma}_0, 0) \quad given \end{cases}$$



With geometrical properties of the level set curve directly computed on the level set function!

$$\vec{N} = \frac{\vec{\nabla}\phi(\vec{\Gamma},t)}{\left\|\vec{\nabla}\phi(\vec{\Gamma},t)\right\|}$$

$$\kappa = \vec{\nabla}\cdot\frac{\vec{\nabla}\phi(\vec{\Gamma},t)}{\left\|\vec{\nabla}\phi(\vec{\Gamma},t)\right\|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{\left(\phi_x^2 + \phi_y^2\right)^{3/2}}$$

Geometric Active Contours

Numerical Methods with Level Sets

What type of Φ function?:

➔ most common choice is the **signed distance function**:

$$\left\|\vec{\nabla}\phi(\vec{\Gamma},t)\right\| = 1 \Rightarrow \begin{cases} \vec{N} = \vec{\nabla}\phi(\vec{\Gamma},t) \\ \kappa = \Delta\phi(\vec{\Gamma},t) \end{cases}$$

– **Watch out !**

The solution of $\dfrac{\partial \phi}{\partial t} = V(\kappa)\left\|\vec{\nabla}\phi\right\|$ is not the signed distance function of the curve solution to $\dfrac{\partial \vec{\Gamma}}{\partial t} = V(\kappa)\vec{N}$

Geometric Active Contours

Numerical Methods with Level Sets

What speed of propagation ?

$$\frac{\partial \phi(\vec{\Gamma},t)}{\partial t} + V(\kappa)\left\|\vec{\nabla}\phi(\vec{\Gamma},t)\right\| = 0$$

- **Take into account:**
  - **Image Information: zero on edges from the objects to segment.**
  - **Preserve smoothing of the contour.**

- **Particular case: Motion under curvature**
  - Each part of the model evolves in the normal direction, with a speed proportional to the curvature.

  ⇨ points can move inward or outward, depending on the curvature's sign.

$$\frac{\partial \phi(x,y,t)}{\partial t} = V(x,y)\left\|\vec{\nabla}\phi(x,y,t)\right\|$$

Arbitrary extension to whole image domain … e.g. **computing the curvature on the overall level set function**

Geometric Active Contours

Numerical Methods with Level Sets

Implementation details:

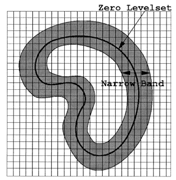
1. Narrow Band

Only evolve level sets in a **narrow band** around the level zero.

➔ Reduces computational cost.

➔ No need to compute evolution speed far from the 0-level.

➔ less constraints on Δt to maintain CFL numerical stability, which limits the maximum speed of deformation.



2. Reinitialization

Why reinitialize ?

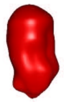
- Maintain **unique correspondence** between a desired contour and its level set function (convergence iif convergence)
- Preserve a **constant gradient norm** ⇨ numerical stability.

$$\phi_t = sign(\phi)\left(1 - \left\|\vec{\nabla}\phi\right\|\right)$$

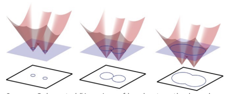
Methods:

- **Direct evaluation**:
  - Detect the 0-level and re-compute the signed distance function.
  - Problem: High computational cost!
- **Iterative update:**
  - Update toward an equilibrium state defined as "norm of gradient =1".
  - Uses opposite flows for negative and positive values.
  - Problem: the 0-level can move during reinitialization.

Geometric Active Contours



Level Sets



Source: Osher et al "A review of level-set methods and some recent applications"

Advantages:

- Change of **topology** allowed without efforts.
- Intrinsic **geometric properties** of the segmented contours are easy to compute (normals, curvatures).
- Extension to **N-D** straightforward: add new spatial variables to the evolution equation of the volume $\phi(x, y, z, \dots, t)$.
- Numerical **implementation**:
  - Discretization of $\phi(x, y, t)$ on regular grid (x, y).
  - Standard numerical schemes for the spatial derivative.

Limitations:

- Increase dimension by +1.

3 limitations related to the numerical implementation:

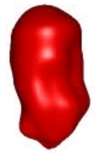
- Construction of an **initial level set function** $\phi(x, y, t = 0)$ from an initial contour (= the 0-level).
- **Evolution** equation **only** defined for the **0-level** ⇨ the speed function V is not defined in general for the other levels
  ⇨ arbitrary spatial extension.
- High risk of **numerical instabilities** of the iterative scheme to define $\phi(x,y,t+1)$

⇨ reinitialization needed every K steps.

45

Geometric Active Contours

- Geometric Active contours
  - Numerical methods via level sets.
  - Geodesic.
  - Mumford-Shah.



46

Geometric Active Contours

Geodesic Deformable Models [Caselles, Kimmel, Sapiro 1997]

Geodesic curves in a Riemannian space

Geodesic = path (locally) minimal between 2 points.

Space with metrics defined from geodesics.

- **Novel approach (equivalent)**

$$\text{Min } E(C)=\underbrace{\int_0^1 \left|C'(s)\right|^2 ds}_{\text{Minimal perimeter….}}-\underbrace{\int_0^1 g\left(\left|\nabla I\left(C(s)\right)\right|\right)ds}_{\text{At locations where the is gradient in the image.}}$$

$\Leftrightarrow$

$\text{Min } \int_0^1 g\left(\left|\nabla I\left(C(s)\right)\right|\right)\left|C'(s)\right|ds$ ➔ Geodesic computation

Geometric Active Contours

1. Geodesic Deformable Models

1st term = Euclidian Geodesic

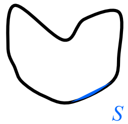
$$\text{Length } L_E(C)=\oint \left|C'(s)\right|ds=\oint dS$$

Euclidian Metric

Solution:

Motion under curvature $\frac{\partial C}{\partial t}=\kappa \vec{N}$

curvature



2nd term = Geodesic to attach on image data

$$\text{Lenght } L_R(C)=\int_0^1 g\left(\left|\nabla I\left(C(s)\right)\right|\right)\left|C'(s)\right|ds$$

$$=\int_0^{L_E(C)} g\left(\underbrace{\left|\nabla I\left(C(s)\right)\right|}_{\text{Image gradient}}\right)\left|C'(s)\right|dS$$

Solution:

$$\frac{\partial C}{\partial t}=g(I)\kappa\vec{N}-\left(\vec{\nabla}g.\vec{N}\right)\vec{N}$$

curvature

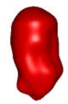
Gradients & contours align

Geometric Active Contours

- Geometric Active contours
  - Numerical methods via level sets.
  - Geodesic.
  - Mumford-Shah.

49

Geometric Active Contours



Mumford & Shah

- Variational method defining a partition I of an image I_0 into “partitions”.
- Image I_0 segmentation, defined on the domain $\boldsymbol{\Omega}$, is provided by a pair (C,I) with: C = contours in the image and I = smooth approximation of I_0.
- Energy associated with the segmentation:

$$E(C,I)=\alpha\overbrace{\int_{\Omega\setminus C}\left|\nabla I\right|^2 d\Omega}^{I\text{ is smooth}}+\beta\,\mathrm{length}(C)+\overbrace{\int_{\Omega\setminus C}\left(I-I_0\right)^2 d\Omega}^{I\text{ ressembles }I_0\text{ (piece-wise constant)}}$$

But allow some mismatches

1D, # of points on C
2D, perimeter of C
3D, surface of C

50

Geometric Active Contours

Mumford & Shah

- Conjuncture
  - There exists a minimal segmentation made of a finite set of curves C^1.
  - [Morel-Solimini 1995, Aubert-Kornprobst 2000]:
    - There exists a minimal segmentation.
    - The minimal segmentation is not unique.
    - The ensemble of solutions is a compact set.
    - Contours are rectifiable (i.e. of finite length).
    - All contours can be included in a single rectifiable curve.

51

Geometric Active Contours

Mumford & Shah

- Particular case:
  - I_0 is a cartoon-like image.
  - The smooth approximation I of I_0 is a piecewise constant image with values c_1 et c_2 which are the mean values of I_0 in the **objects** and the **background**.
  - The contour C corresponds to the **borders** of the objects.

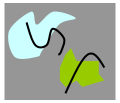
➔ Leads to:

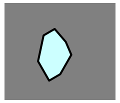
$$E(\overbrace{c_1, c_2}^{I}, C) = \upsilon Length(C) + \lambda \int_{inside(C)} |I_0 - c_1|^2 + \lambda \int_{outside(C)} |I_0 - c_2|^2$$

$$|\vec{\nabla} I| = 0$$

➔ Leads to:

c_1 = mean value of I_0 inside C
c_2 = mean value of I_0 outside C





52

Geometric Active Contours

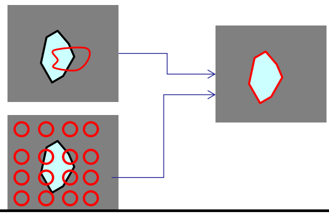
Mumford & Shah

Deformable models without edges [Chan & Vese]

$$E(C)=\underbrace{\mu L(C)+\nu A(C)}_{\text{Regularizing terms (cf. internal energy)}}+\underbrace{\lambda_1\int_{inside(C)}|I_0-c_1|^2\,d\Omega+\lambda_2\int_{outside(C)}|I_0-c_2|^2\,d\Omega}_{\text{Homogeneity constraints (cf. external energy)}}$$

$L(C)$ = length of C

$A(C)$ = area of C



53

Geometric Active Contours

Mumford & Shah

Deformable models without edges [Chan & Vese]

$$E(C)=\mu L(C)+\nu A(C)+\lambda_1\int_{inside(C)}|I_0-c_1|^2\,d\Omega+\lambda_2\int_{outside(C)}|I_0-c_2|^2\,d\Omega$$

1. Insert the n-D curve C in a level set function $(n+1)$-D ϕ:

$$C \quad = \{x\in\mathbb{R}^N / \phi(x)=0\}$$
$$C_{inside} = \{x\in\mathbb{R}^N / \phi(x)<0\}$$
$$C_{outside} = \{x\in\mathbb{R}^N / \phi(x)>0\}$$

2. Define a Heaviside function $H(\phi)$:

$$H(z)=\begin{cases}1\ if\ z\leq 0\\ 0\ if\ z\geq 0\end{cases}$$

3. Define a Dirac function $\delta(\phi)$

$$\delta(z)=\frac{dH(z)}{dz}$$

54

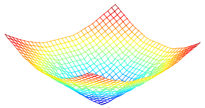
Geometric Active Contours

Mumford & Shah

Deformable models without edges [Chan & Vese]

- Level set Function: ϕ

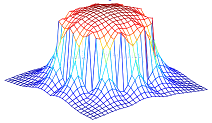
  ϕ = distance to the 0-level



$C = \{(x,y,z) / \phi(x,y,z) = 0\}$

- Heaviside Function: $H(\phi)$

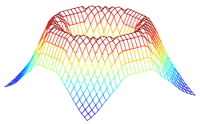
$$H(\phi) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right)$$



$$Area(C) = Area(\phi \le 0) = \int_\Omega H(\phi)\, d\Omega$$

- Dirac Function: $\delta(\phi)$

$$\delta(\phi) = \frac{1}{\pi}\left(\frac{\varepsilon}{\phi^2 + \varepsilon^2}\right)$$



$$Length(C) = Length(\phi = 0) = \int_\Omega |\nabla H(\phi)|\, d\Omega = \int_\Omega \delta(\phi)|\nabla\phi|\, d\Omega$$

Geometric Active Contours

Mumford & Shah

Deformable models without edges [Chan & Vese]

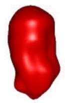
$$E_\varepsilon(\phi, c_0, c_1) = \mu \underbrace{\int_\Omega \delta_\varepsilon(\phi)|\nabla\phi|\, dx}_{\text{length}} + \upsilon \underbrace{\int_\Omega H_\varepsilon(\phi)}_{\text{area}} + \lambda_0 \underbrace{\int_\Omega |I_0 - c_0|^2 H_\varepsilon(\phi)\, dx}_{\text{homogeneity}} + \lambda_1 \underbrace{\int_\Omega |I_0 - c_1|(1 - H_\varepsilon(\phi))dx}_{\text{homogeneity}}$$

$$\begin{cases} \delta_\varepsilon(\phi)\left[\mu\, div\left(\dfrac{\nabla\phi}{|\nabla\phi|}\right) - \nu - \lambda_0 (I - c_0)^2 + \lambda_1 (I - c_1)^2\right] = 0 \\ \dfrac{\delta_\varepsilon(\phi)}{|\nabla\phi|}\dfrac{\partial\phi}{\partial n} = 0 \ on\ \partial\Omega, \end{cases}$$

Segmentation via $\inf_{\phi, c_0, c_1} E_\varepsilon(\phi, c_0, c_1)$

$$\begin{cases} c_0(\phi) = \dfrac{\int_\Omega I(x,y,z) H(\phi(x,y,z))\, dx\, dy\, dz}{\int_\Omega H(\phi(x,y,z))\, dx\, dy\, dz} \\ c_1(\phi) = \dfrac{\int_\Omega I(x,y,z)(1 - H(\phi(x,y,z)))\, dx\, dy\, dz}{\int_\Omega (1 - H(\phi(x,y,z)))\, dx\, dy\, dz} \end{cases}$$

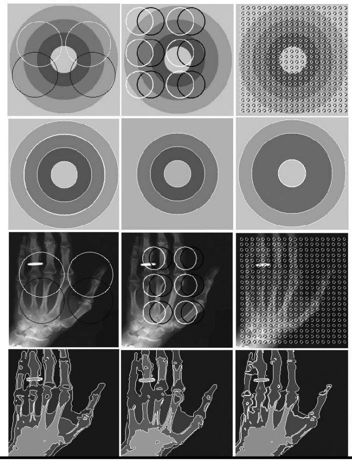
Geometric Active Contours

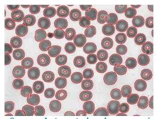


Bibliography

– D. **Mumford** and J. **Shah**, " Boundary detection by minimizing functional" International Conference on Computer Vision and Pattern Recognition, pp. 22-26, 1985.
– S. **Osher** and J. A. **Sethian**, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations" Journal of Computational Physics, vol. 79, No. 1, pp. 12-49, 1988.
– V. **Caselles**, F. Catté, B. Coll, F. Dibos, "A geometric model for edge detection", Num. Mathematik, 66, pp. 1-31, 1993.
– R. Malladi, J.A. Sethian, Baba C. Vemuri: "Shape modeling with front propagation: A level set approach". IEEE Transactions on Pattern Analysis and Machine Intelligence 17(2), pp 158-175, 1995.
– T. F. **Chan** and L. A. **Vese**, "Active contours without edges " IEEE Transactions on Image Processing, vol. 10, No. 2, pp. 266 - 277, 2001.

Source: Image segmentation and selective smoothing by using Mumford-Shah model





Source: **Automated and unsupervised detection of malaria parasites in microscopic images**

57

Active Contours summary

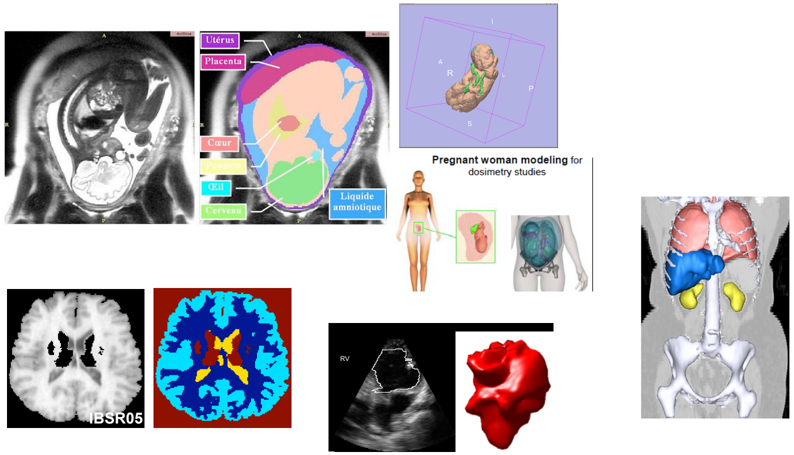
Two families of N-D Active Contours:

- Parametric :
  – Explicit representation of the contour.
  - 👍 Compact representation allowing fast implementation (enable real-time applications).
  - ✘ Changes of topology very difficult to handle (in 3D).

- Geometric:
  – Implicit representation of the contour as the level 0 of a scalar function of dimension (N-D+1).
  - 👍 Contour parameterization **after** the deformations.
  - 👍 Flexible adaptation of the contours' topology.
  - ✘ Increase dimension of space search.

58

Active Contours summary



59