

Medical Image Analysis

CS 593 / 791

Computer Science and Electrical Engineering Dept.
West Virginia University

2nd February 2007

Outline

- 1 Basic active contour behavior
- 2 Internal forces
- 3 Image-based forces
- 4 Constraints
- 5 Numerical implementation
- 6 Weaknesses
- 7 Extensions to the original paper

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces
- 4 Constraints
- 5 Numerical implementation
- 6 Weaknesses
- 7 Extensions to the original paper

Basic active contour behavior

An **active contour** (or **snake**) is an energy minimizing parametric curve which evolves according to external constraints and influenced by image forces which pull it toward features of interest.

- The exact energy functionals involved will depend on which features are of interest.
- We will impose membrane and thin-plate spline smoothness constraints on the snakes, and also allow for user intervention.

Energy Functional

Behavior is governed by the energy functional

$$E_{snake}^* = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)) ds$$

- $v(s)$ is the parametric curve
- E_{int} represents the smoothness constraints
- E_{image} represents image data constraints
- E_{con} represents user input

Internal Energy

A combination of membrane and thin-plate spline smoothness

$$E_{int} = \frac{1}{2}(\alpha(s)\|v_s(s)\|^2 + \beta(s)\|v_{ss}(s)\|^2)$$

- This model allows the weights α, β to vary along the length of the curve.
- Setting $\beta(s) = 0$ allows a discontinuity to develop at s .

Definition

Parametric Curve : A vector valued function from some interval of the real line to Euclidean space.

$$c(p) = \begin{bmatrix} x(p) \\ y(p) \end{bmatrix}$$

where $p \in [a, b]$.

If $c(a) = c(b)$ the curve is **closed**.

We will focus on **regular** parameterizations. $(c'(p) \neq 0 \quad \forall p)$

There is generally not a **unique** parameterization of a curve.

Let $f : [c, d] \rightarrow [a, b]$

$c(f(p))$ is called a **reparameterization** of $c(p)$.

$c(f(p))$ describes the same curve as $c(p)$ for $p \in [c, d]$. The new parameterization may trace out the curve in a different direction and with different "speed".

Example:

$$c(p) = \begin{bmatrix} r \cos(p) \\ r \sin(p) \end{bmatrix}, p \in [0, 2\pi]$$

- Let $f(p) = 2\pi - p$. What is $c(f(p))$ for $p \in [0, 2\pi]$?
- Let $f(p) = 2\pi p$. What is $c(f(p))$ for $p \in [0, 1]$?
- Let $f(p) = 2\pi p^2$. What is $c(f(p))$ for $p \in [0, 1]$?

Tangent and Normal vector

Unit Tangent Vector

$$T(p) = \frac{c'(p)}{\|c'(p)\|}$$

Unit Normal Vector

$$N(p) = \frac{T'(p)}{\|T'(p)\|}$$

$$T(p) \perp N(p) \rightarrow T(p) \cdot N(p) = 0.$$

$$N(p) = \begin{bmatrix} -t_y \\ t_x \end{bmatrix} \text{ or } \begin{bmatrix} t_y \\ -t_x \end{bmatrix}$$

Example:

$$c(p) = \begin{bmatrix} r \cos(p) \\ r \sin(p) \end{bmatrix}, p \in [0, 2\pi]$$

$$c'(p) = \begin{bmatrix} -r \sin(p) \\ r \cos(p) \end{bmatrix}$$

$$\|c'(p)\| = \sqrt{r^2 \sin^2(p) + r^2 \cos^2(p)} = r$$

$$T(p) = \frac{c'(p)}{\|c'(p)\|} = \begin{bmatrix} -\sin(p) \\ \cos(p) \end{bmatrix}$$

$$N(p) = \frac{T'(p)}{\|T'(p)\|} = \begin{bmatrix} -\cos(p) \\ -\sin(p) \end{bmatrix}$$

Chord length approximation:

$$\begin{aligned} S &= \sum_i \|\Delta c_i\| \\ &= \sum_i \left\| \frac{\Delta c_i}{\Delta p} \right\| \Delta p \end{aligned}$$

$$\lim_{\Delta p \rightarrow 0} \sum_i \left\| \frac{\Delta c_i}{\Delta p} \right\| \Delta p = \int_a^b \|c'(p)\| dp$$

Arclength:

$$s = \int_a^b \|c'(p)\| dp$$

Example:

$$c(p) = \begin{bmatrix} r \cos(p) \\ r \sin(p) \end{bmatrix}, p \in [0, 2\pi]$$

$$c'(p) = \begin{bmatrix} -r \sin(p) \\ r \cos(p) \end{bmatrix}$$

$$\|c'(p)\| = \sqrt{r^2 \sin^2(p) + r^2 \cos^2(p)} = r$$

The total length of the curve is given by

$$s = \int_0^{2\pi} r dp = 2\pi r$$

Arc-length parameterization

The arc-length parameterization, $c(s)$ of the curve results in simplified computations of tangent vector, normal vector and curvature.

- $s \in [0, L]$
- $c(s)$ is "where we are on the curve after we have traveled a distance s along the curve."

For the curve $c(p)$ with $p \in [0, b]$ the arc-length function

$$s(p) = \int_0^p \|c'(p)\| dp$$

the distance on the curve between $c(0)$ and $c(p)$. If we can invert this function to find $p(s)$, then $c(p(s))$ will be an arc-length parameterization.

Example:

$$c(p) = \begin{bmatrix} r \cos(p) \\ r \sin(p) \end{bmatrix}, p \in [0, 2\pi]$$

$$\|c'(p)\| = \sqrt{r^2 \sin^2(p) + r^2 \cos^2(p)} = r$$

The arclength function is

$$s(p) = \int_0^p r dq = pr$$

Its inverse is given by

$$p(s) = \frac{s}{r}$$

$$c(p(s)) = \begin{bmatrix} r \cos\left(\frac{s}{r}\right) \\ r \sin\left(\frac{s}{r}\right) \end{bmatrix}, s \in [0, 2\pi r]$$

Tangent and Normal of $c(s)$

For arclength parameterized curve $c(s)$

- $T(s) = c'(s)$
- $c''(s) \perp c'(s)$, so
- $N(s) = c''(s)/\|c''(s)\|$
- $\|c''(s)\|$ is called the curvature

Example:

$$c(s) = \begin{bmatrix} r \cos\left(\frac{s}{r}\right) \\ r \sin\left(\frac{s}{r}\right) \end{bmatrix}, s \in [0, 2\pi r]$$

$$c'(s) = \begin{bmatrix} -\sin\left(\frac{s}{r}\right) \\ \cos\left(\frac{s}{r}\right) \end{bmatrix} = T(p(s))$$

$$c''(s) = \begin{bmatrix} -\frac{1}{r} \cos\left(\frac{s}{r}\right) \\ -\frac{1}{r} \sin\left(\frac{s}{r}\right) \end{bmatrix} = \kappa N(p(s))$$

The curvature $\kappa = \|c''(s)\| = \frac{1}{r}$.

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces**
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces
- 4 Constraints
- 5 Numerical implementation
- 6 Weaknesses
- 7 Extensions to the original paper

In general, we may want to minimize a combination of the two energies:

$$\alpha MEM(c) + \beta TPS(c) = \int_0^l \alpha \|c'(s)\|^2 + \beta \|c''(s)\|^2 ds$$

We will show that the minimization conditions are

$$-\frac{d}{ds} \alpha c'(s) + \frac{d^2}{ds^2} \beta c''(s) = 0$$

$$-\alpha c''(s) + \beta c''''(s) = 0$$

Membrane spline energy of $c(s)$

$$MEM(c) = \int_0^L \|c'(s)\|^2 ds = \int_0^L x'(s)^2 + y'(s)^2 ds$$

For an arclength parameterization, minimizing $MEM(c)$ is equivalent to minimizing the length of c .

Applying variational calculus to

$$\min_{x(s), y(s)} \int_0^L x'(s)^2 + y'(s)^2 ds$$

gives two Euler-Lagrange conditions

$$\frac{d}{ds}(2x'(s)) = 0$$

$$\frac{d}{ds}(2y'(s)) = 0$$

Membrane spline energy of $c(s)$

The Euler-Lagrange conditions can be rewritten as

$$c''(s) = 0$$

The evolution equation

$$\begin{aligned}\frac{dc}{dt} &= \frac{d^2c}{ds^2} \\ &= \kappa N\end{aligned}$$

is also known as the geometric heat equation.

Variational Calculus

The conditions for minimizing

$$\min_c \int_{\Omega} F(s, c(s), c'(s), c''(s)) ds$$

are

$$F_c - \frac{d}{ds} F_{c'(s)} + \frac{d^2}{ds^2} F_{c''(s)} = 0$$

The evolution equation that will satisfying this condition when steady-state has been reached

$$\frac{\partial c}{\partial t} = -F_c + \frac{d}{ds} F_{c'(s)} - \frac{d^2}{ds^2} F_{c''(s)}$$

Thin-plate spline energy of $c(s)$

$$TPS(c) = \int_0^L \|c''(s)\|^2 ds = \int_0^L x''(s)^2 + y''(s)^2 ds$$

For an arclength parameterization, minimizing $TPS(c)$ is equivalent to minimizing the square curvature of c .

The conditions for minimizing

$$\min_c \int_{\Omega} F(s, c(s), c'(s), c''(s)) ds$$

are

$$F_c - \frac{d}{ds} F_{c'(s)} + \frac{d^2}{ds^2} F_{c''(s)} = 0$$

The conditions for minimizing $TPS(c)$ are

$$c''''(s) = 0$$

Constant Coefficients α, β

Minimizing snake energy

$$\begin{aligned} \min_{x(s), y(s)} \int_0^1 & \frac{1}{2} \alpha (x'(s)^2 + y'(s)^2) \\ & + \frac{1}{2} \beta (x''(s)^2 + y''(s)^2) \\ & + E_{ext}(x(s), y(s)) \, ds \end{aligned}$$

The conditions for minimization are

$$\begin{aligned} -\alpha x_{ss} + \beta x_{ssss} + \frac{\partial E_{ext}}{\partial x} &= 0 \\ -\alpha y_{ss} + \beta y_{ssss} + \frac{\partial E_{ext}}{\partial y} &= 0 \end{aligned}$$

The evolution equation is

$$\frac{\partial v}{\partial t} = \alpha v_{ss} - \beta v_{ssss} - \nabla E_{ext}$$

Energy Functional

Let $E_{ext}(v(s)) = E_{image}(v(s)) + E_{con}(v(s))$.

Minimize $E_{int} + E_{ext}$ with respect to $x(s)$ and $y(s)$:

$$\begin{aligned} \min_{x(s), y(s)} \int_0^1 & \frac{1}{2} \alpha(s) (x'(s))^2 + y'(s)^2 \\ & + \frac{1}{2} \beta(s) (x''(s))^2 + y''(s)^2 \\ & + E_{ext}(x(s), y(s)) \, ds \end{aligned}$$

The conditions are

$$\begin{aligned} -\frac{\partial}{\partial s}(\alpha(s)x'(s)) + \frac{\partial^2}{\partial s^2}(\beta(s)x''(s)) + \frac{\partial E_{ext}}{\partial x} &= 0 \\ -\frac{\partial}{\partial s}(\alpha(s)y'(s)) + \frac{\partial^2}{\partial s^2}(\beta(s)y''(s)) + \frac{\partial E_{ext}}{\partial y} &= 0 \end{aligned}$$

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces**
- 4 Constraints
- 5 Numerical implementation
- 6 Weaknesses
- 7 Extensions to the original paper

- The **image** is a mapping $I : R^2 \rightarrow R$.
- The **curve** is a mapping $v : R \rightarrow R^2$.
- We may form composite functions from these two mappings

For example:

- $I(v(p))$ is the intensity of the image at the point $v(p)$ on the curve.
- $\nabla I(v(p))$ is the image gradient at the point $v(p)$ on the curve.

We will use a weighted sum of two energy terms:

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge}$$

We will ignore the termination functional.

Line Functional

$$w_{line}E_{line} = w_{line}I(v(s))$$

- if $w_{line} > 0$ the snake will be attracted to dark contours
- if $w_{line} < 0$ the snake will be attracted to light contours

Edge Functional

$$E_{edge(1)} = -\|\nabla I(v(s))\|^2$$

The snake will be attracted to large image gradients.

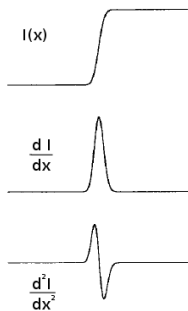
Scale Space

- We can use scale space to enlarge the convergence region of the snake.
- Recall that linear scale spaces result in blurred edges at coarse scales.
- This will propagate edge information far from the edge.

$$E_{edge(2)} = -\|\nabla(G_\sigma * I(v(s)))\|^2$$

Marr-Hildreth

- Edges : zero crossings of the Laplacian ($\nabla^2 = I_{xx} + I_{yy}$)



$$E_{edge(3)} = (G_\sigma * \nabla^2 I(v(s)))^2$$

The snake will be attracted to zero crossings of the smoothed Laplacian.

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces
- 4 Constraints**
- 5 Numerical implementation
- 6 Weaknesses
- 7 Extensions to the original paper

User Applied Constraints

Spring Energy

The user may connect virtual springs between fixed point p , and contour position v :

$$\begin{aligned} E_{spring} &= k||p - v||^2 \\ &= k((p_x - x)^2 + (p_y - y)^2) \end{aligned}$$

where k is a constant (spring stiffness)

$$\nabla E_{spring} = \begin{bmatrix} -2k(p_x - x) \\ -2k(p_y - y) \end{bmatrix} = -2k(p - v)$$

In the evolution equation

$$\frac{\partial v}{\partial t} = \alpha v_{ss} - \beta v_{ssss} - \nabla E_{ext}$$

User Applied Constraints

Repulsive Energy

Forces v away from fixed position p :

$$\begin{aligned}
 E_{repulsion} &= \frac{1}{\|p - v\|} \\
 &= \frac{1}{\sqrt{(p_x - x)^2 + (p_y - y)^2}}
 \end{aligned}$$

$$\frac{\partial E_{repulsion}}{\partial x} = (p_x - x)((p_x - x)^2 + (p_y - y)^2)^{-\frac{3}{2}}$$

$$\frac{\partial E_{repulsion}}{\partial y} = (p_y - y)((p_x - x)^2 + (p_y - y)^2)^{-\frac{3}{2}}$$

User Applied Constraints

$$\frac{\partial E_{repulsion}}{\partial x} = (p_x - x)((p_x - x)^2 + (p_y - y)^2)^{-\frac{3}{2}}$$

$$\frac{\partial E_{repulsion}}{\partial y} = (p_y - y)((p_x - x)^2 + (p_y - y)^2)^{-\frac{3}{2}}$$

$$\nabla E_{repulsion} = \frac{1}{r^2} \frac{p - v}{r}$$

where $r = \|p - v\|$.

In the evolution equation, this term pushes v in the direction $v - p$ with magnitude $\frac{1}{r^2}$.

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces
- 4 Constraints
- 5 Numerical implementation**
- 6 Weaknesses
- 7 Extensions to the original paper

Discretized Curve

Store the curve as a vector of samples of $v(s)$ at evenly spaced intervals in s .
For $v(s) = (x(s), y(s))$,

$$\mathbf{x}_i = x(ih)$$

$$\mathbf{y}_i = y(ih)$$

where h is the parameter step size.

Compute derivatives of $v(s)$ using finite difference formulas.

For General $\alpha(s)$

$$-\frac{\partial}{\partial s}(\alpha(s)x'(s)) + \frac{\partial^2}{\partial s^2}(\beta(s)x''(s)) + \frac{\partial E_{ext}}{\partial x} = 0$$

$$-\frac{\partial}{\partial s}(\alpha(s)y'(s)) + \frac{\partial^2}{\partial s^2}(\beta(s)y''(s)) + \frac{\partial E_{ext}}{\partial y} = 0$$

Witkin, Kass, Terzopoulos discretize the Euler-Lagrange equations at this point.

First, discretize (x', x'', y', y'') :

$$-\frac{\partial}{\partial s}(\alpha(s)(x_i - x_{i-1})) + \frac{\partial^2}{\partial s^2}(\beta(s)(x_{i-1} - 2x_i + x_{i+1})) + f_x(i) = 0$$

$$-\frac{\partial}{\partial s}(\alpha(s)(y_i - y_{i-1})) + \frac{\partial^2}{\partial s^2}(\beta(s)(y_{i-1} - 2y_i + y_{i+1})) + f_y(i) = 0$$

Second, discretize $(\frac{\partial}{\partial s}, \frac{\partial^2}{\partial s^2})$:

$$\begin{aligned}
 & - (\alpha_{i+1}(x_{i+1} - x_i) - \alpha_i(x_i - x_{i-1})) \\
 & + \beta_{i-1}(x_{i-2} - 2x_{i-1} + x_i) \\
 & - 2\beta_i(x_{i-1} - 2x_i + x_{i+1}) \\
 & + \beta_{i+1}(x_i - 2x_{i+1} + x_{i+2}) \\
 & + f_x(i) = 0
 \end{aligned}$$

By doing the same for y , we can write two linear systems for the snake model...

The two Euler-Lagrange equations can be written in matrix form

$$Ax + f_x(x, y) = 0$$

$$Ay + f_y(x, y) = 0$$

where A is $(n \times n)$ sparse matrix with 5 nonzero diagonals.

- A represents the smoothness of the curve
- f_x, f_y represent the external forces

The system of evolution equations is

$$\frac{\partial x}{\partial t} = -Ax - f_x(x, y)$$

$$\frac{\partial y}{\partial t} = -Ay - f_y(x, y)$$

The linearized evolution equation

The authors present a mixed explicit/implicit method:

implicit with respect to internal forces, and

explicit with respect to external forces.

Writing the finite difference in time as $\gamma(c^t - c^{t-1})$

$$Ax^t + f_x(x^{t-1}, y^{t-1}) = -\gamma(x^t - x^{t-1})$$

$$Ay^t + f_y(x^{t-1}, y^{t-1}) = -\gamma(y^t - y^{t-1})$$

These equations can be rewritten as

$$(A + \gamma I)x^t = \gamma x^{t-1} + f_x(x^{t-1}, y^{t-1})$$

$$(A + \gamma I)y^t = \gamma y^{t-1} + f_y(x^{t-1}, y^{t-1})$$

$(A + \gamma I)$ is constant over time, so this matrix may be factorized/inverted once.

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces
- 4 Constraints
- 5 Numerical implementation
- 6 Weaknesses**
- 7 Extensions to the original paper

Weaknesses

- Snakes are prone to getting stuck in local minima (they only see local image data)
- Topologically limited
- Only a 2D model
- Parameterization dependent
- Snakes may self-intersect, or become degenerate.

Outline

- 1 Basic active contour behavior
 - Geometry of curves
- 2 Internal forces
 - Membrane spline energy
 - Thin-plate spline energy
- 3 Image-based forces
- 4 Constraints
- 5 Numerical implementation
- 6 Weaknesses
- 7 Extensions to the original paper**

Inflation Force

Also called the "Balloon model"

L. D. Cohen, "On active contour models and balloons", CVGIP: Image Understanding, 1991.

$$\nabla E = f(v(s))N(v(s))$$

- The snake expands in the normal direction.
- f may be intensity-based, edge-based, or constant
- This force can push the snake past local minima of the energy functional

Adding mass to the snake

$$\mu \frac{\partial^2 v}{\partial t^2} + \gamma \frac{\partial v}{\partial t} = \alpha v_{ss} - \beta v_{ssss} - \nabla E_{ext}$$

- Introducing mass gives the model inertia.
- This model can overshoot local minima.
- Equilibrium when $\frac{\partial^2 v}{\partial t^2} = \frac{\partial v}{\partial t} = 0$

Reparameterization

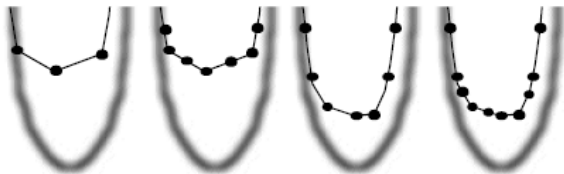
$$E_{param} = \int_{\Omega} (\|v'(s)\|^2 - c)^2 ds$$

- $v'(s) \cdot v'(s) = 1$ for an arclength parameterization.
- This energy can maintain an arclength parameterization
- Some degeneracies can be avoided

Avoided in the level-set formulation by representing the curve/surface implicitly.

Subdivision

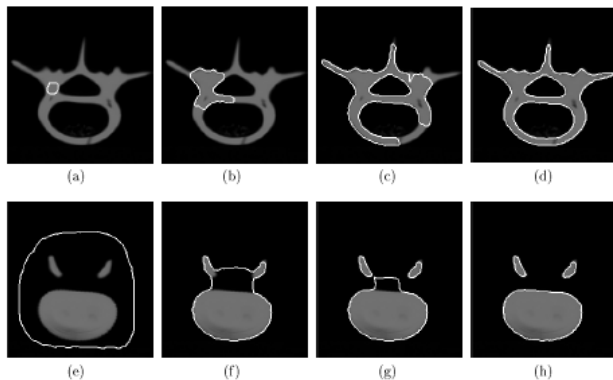
Ivins, J., Porrill, J., "Statistical snakes: Active region models.", Proc. 5th British Machine Vision Conf., 1994.
 also proposed by others.



- Add more sample points as $v(s)$ grows longer.
- Reparameterize so that high curvature regions are samples more densely.

T-snakes

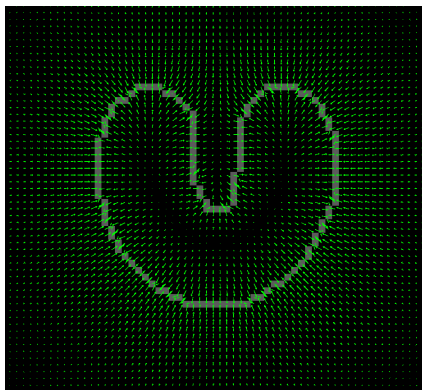
McInerney, T. and Terzopoulos, D., "T-snakes: Topology adaptive snakes", Medical Image Analysis, 2000.



The level-set formulation is also topologically adaptive.

Gradient Vector Field

- The gradient of I does not provide useful information when the image is smooth.
- We would like to know the direction to the nearest edge.



Gradient Vector Field

Xu, C. and Prince, J.L., "Gradient Vector Flow: A New External Force for Snakes", CVPR, 1997.

The gradient vector field, $\mathbf{g}(x, y) = [u(x, y), v(x, y)]$ minimizes the energy

$$E_{gvf} = \int_{\Omega} \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \|\nabla I\|^2 \|\mathbf{g} - \nabla I\|^2 ds$$

- First term: membrane spline smoothness on u, v .
- Second term: \mathbf{g} is near ∇I when $\|\nabla I\|$ is large
- When $\|\nabla I\|$ is small, the smoothness term dominates

Gradient Vector Field

- Determine the evolution equation for minimizing E_{gvf} .
- Given $I(x, y)$ compute the gvf, $\mathbf{g}(x, y)$.
- Use $\mathbf{g}(x, y)$ in the evolution equation for $v(s)$.

$$\frac{\partial v}{\partial t} = \alpha v_{ss} - \beta v_{ssss} + \mathbf{g}(v(s))$$

Next Class

Level Set Methods : implicit active contours.

Read

Malladi, R., Sethian, J., Vemuri, B., "Shape Modeling with Front Propagation : A Level Set Approach.", IEEE PAMI, 1995.