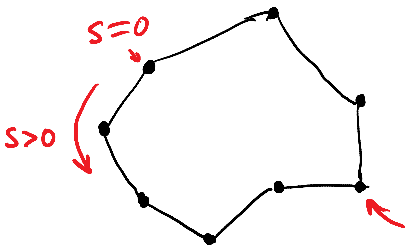
Snakes

L40

Goal: To learn about active contours, how they can be represented, and how we can control them to perform segmentation.

An active contour is a **deformable boundary** on an image designed to converge to an anatomical boundary of some sort.

This deformable boundary is often represented by a number of nodes or vertices, a form of active contour we call **snakes.**



Position along the curve is parameterized by s.

$X(s) = (x(s), y(s))$

The nodes are moved in order to minimize a cost function. The cost function usually takes the form of an energy function.

$$E(X) = \underbrace{\int E_{int}(X(s))\,ds}_{\text{Internal energy}} + \underbrace{\int E_{ext}(X(s))\,ds}_{\text{External energy}}$$

Internal energy
eg. elasticity, rigidity, etc.

External energy
eg. image intensity/gradient, user constraints etc.

Internal Energy

Often modelled using

$$E_{int}(X) = \alpha \left|\frac{\partial X}{\partial s}\right|^2 + \beta \left|\frac{\partial^2 X}{\partial s^2}\right|^2$$

elastic term discourages stretching

curvature term discourages bending

External Energy

Often modelled as a **"potential energy function"**, it takes on **smaller** values near **edges** or other objects of interest.

eg. $P(x,y) = -w \left\| \nabla (g * f) \right\|^2$ will stop or slow the contour at edges

Energy Minimization

The procedure for finding the optimal contour X to minimize the energy **"functional"** is called **variational** calculus. Just like in calculus, to find local optima of a function **f(x)** you set $\frac{df}{dx} = 0$, we set the variation of a functional to 0, $\frac{dE}{dX} = 0$

Skipping all the details, it essentially results in having to solve a PDE, in our case the Euler-Lagrange equation

$$\frac{\partial}{\partial s}\left(\alpha \frac{\partial X}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 X}{\partial s^2}\right) - \nabla P(X) = 0 \quad (1)$$

Just like in anisotropic diffusion, we introduce a time variable and use it to evolve toward a **steady-state** solution of (1)

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

Balancing-Force Formulation

Generalizing this approach to include external influences other than potential energy, we turn to a dynamic force formulation.

$$\mu \frac{\partial^2 X}{\partial t^2} = F_{damp}(X) + F_{int}(X) + F_{ext}(X) \qquad \left(\text{comes from } F=ma\right)$$

In segmentation, we often set μ to 0, and use $-\gamma \frac{\partial X}{\partial t}$ as the viscous damping force, $F_{damp}(X)$. Thus, we get

$$\gamma \frac{\partial X}{\partial t} = F_{int}(X) + F_{ext}(X)$$

<u>Internal Forces</u>

As above...

$$F_{int}(X) = \frac{\partial}{\partial s}\left(\alpha \frac{\partial X}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 X}{\partial s^2}\right)$$

In a finite-difference framework,

$$\frac{\partial}{\partial s}\left(\alpha \frac{\partial X_i}{\partial s}\right) \approx \frac{1}{h_i^2}\left[\alpha_{i+1}\left(X_{i+1}^n - X_i^n\right) - \alpha_i\left(X_i^n - X_{i-1}^n\right)\right] \qquad \left(\text{assuming } h_i = h_{i+1}\right)$$

and

$$\frac{\partial^2}{\partial s^2}\left(\beta \frac{\partial^2 X}{\partial s^2}\right) \approx \frac{1}{h_i^4}\Big[\beta_{i-1}\left(X_{i-2}^n - 2X_{i-1}^n + X_i^n\right) - 2\beta_i\left(X_{i-1}^n - 2X_i^n + X_{i+1}^n\right) + \beta_{i+1}\left(X_i^n - 2X_{i+1}^n + X_{i+2}^n\right)\Big]$$

All these can be represented by a matrix operator

$$F_{int}(X^n) \cong AX^n$$

Recall: $\gamma \frac{\partial X}{\partial t} = F_{int}(X) + F_{ext}(X)$

Thus, using finite-differencing for $\frac{\partial X}{\partial t}$, we have

$$\frac{X^n - X^{n-1}}{\Delta t} = AX^n + F_{ext}(X^{n-1})$$

Bring X^n to the left-hand side...

$$(I - \Delta t A)X^n = X^{n-1} + \Delta t F_{ext}(X^{n-1})$$

$$X^n = \underbrace{(I - \Delta t A)^{-1}}\left[X^{n-1} + \Delta t F_{ext}(X^{n-1})\right]$$

Does not change much, except that the spacing (h_i) can change.

$\Rightarrow$ efficient to compute LU factorization

$LU = I - \Delta t A$ $\quad O(m^3)$ flops

$\uparrow$ # of nodes

Then each iteration is $O(m^2)$ flops.

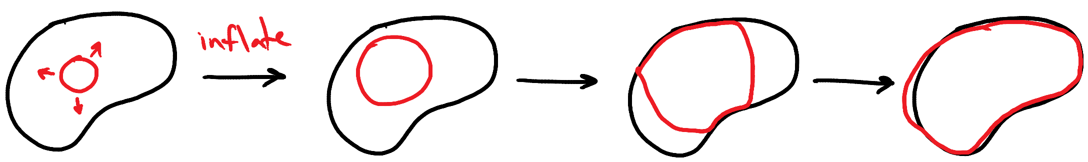
External Forces

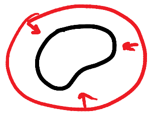
Lots of options, limited only by your imagination. Here are some common ones.

Pressure Force:

Also called **"balloon force"**, **grows** (or **shrinks**) the contour at a fixed rate. This allows one to start with a contour that isn't close to the desired solution.

eg.



Could also start with  and deflate.

Force pressure is defined as

$$F_p(x) = w\, N(x)$$

where

$N(x)$ is the outward normal vector

w is positive for inflation, negative for deflation

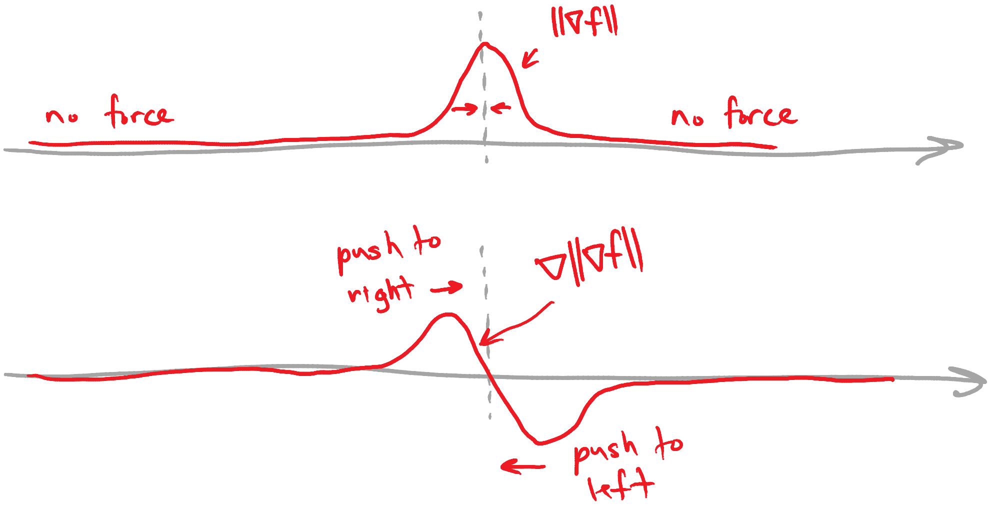
Edge-Stopping Force:

This force is designed to **stop and hold** the snake on **an edge.**

Consider an image f , and the magnitude of its gradient, $\|\nabla f\|$.

i.e.

We want to use the **gradient** of $\|\nabla f\|$ to force our snake.



Hence, we get the force

$$F(x) = \nabla \|\nabla f\| \quad \text{or} \quad \nabla \|\nabla f\|^2$$

Hence, we get the force

$$F_E(x) = \nabla\|\nabla f\| \quad \text{or} \quad \nabla\|\nabla f\|^2$$

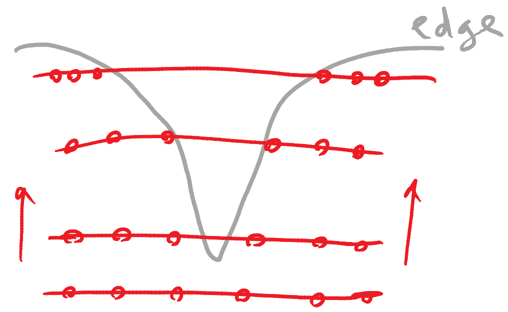
It can also be helpful to use the gradient of a blurred version of f,

$$F_E(x) = \nabla\|\nabla(G * f)\|$$

Problems with Snakes

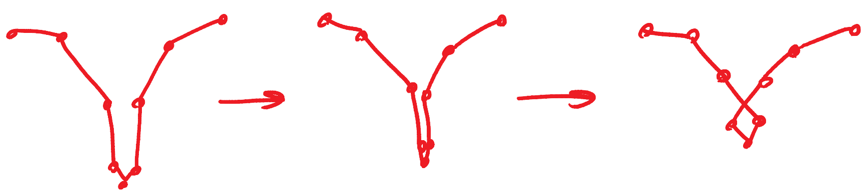
1. Node spacing: As the curve evolves, the points move around and can **spread apart**, especially if one is inflating a curve. Even if inflation is minimal, the points can still redistribute.

eg.



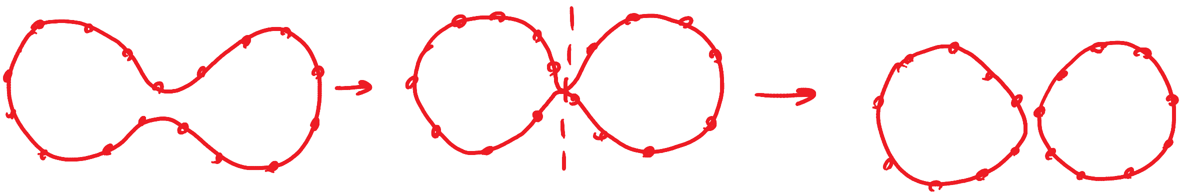
2. Toplogical Changes: It can be tricky to detect and handle situations when the curve overlaps itself.

eg.



For example, it might be appropriate to cleave a curve into two separate curves.

eg.



Fidgety to implement this break,
and then you have 2 curves & so on...