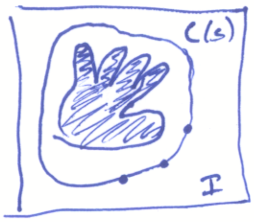
Curve evolution → Motivation

Announcements
- Mon: Paul Viola
- Tue: no talk
- Fri: Dieter Fox

Seminal work: "snakes" Kass, Witkin, Terzopoulos 88



evolve the curve to fit the contour (shrink-wrap)

Approach: define energy function (cost) of curve C

$$E(c) = \int_{s=0}^{s=1} F(c, c', c'')\, ds$$

$$F = F_{edge} + F_{smooth} + F_{inflate/deflate}$$ (lot's o' variations)

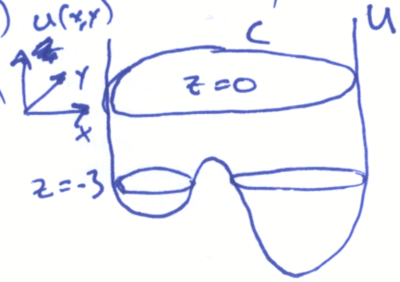
Q: how to describe the curve?
→ piecewise linear, spline, etc.) Lagrangian formulation

Probs (ask)
- self-intersections
- terms compete with each other
- fixed topology
- poor conditioning

Soln (to most of these probs)
- define curve as level set of an implicit fn on a uniform grid
  - i.e., implicit fn is an image $u(x,y)$. curve defined by points (x,y) where $u(x,y)=0$ (or $=c$)

→ evolve the values (pixels) of the grid

→ Eulerian formulation



Probs
- ~~also~~ harder to implement ~~xx~~ if you want it to run fast
- harder to control?
- integer precision (but can interpolate)

Curve evolution → key ideas

Energy of a curve: $E(C) = \int_{s=0}^{s=1} F(C, C', \dots)\,ds$

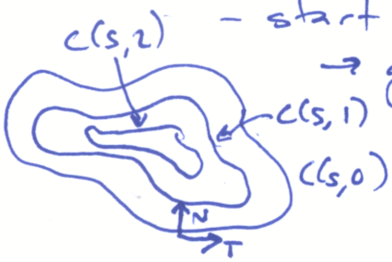
Minimize: $\underset{C}{\operatorname{argmin}}\ E(C)$

→ a "variational" prob – local opt. over space of functions

→ usually too hard to solve directly

– start with initial curve $C(s,0)$, deform over time

→ generates a sequence of curves $C(s,t)$



Curve evolution: $\frac{\partial}{\partial t} C = \cancel{\alpha T} + \beta N$

$\Rightarrow \frac{\partial}{\partial t} C = \beta N$ curve evdn eqn.

~~What to reach~~

$\boxed{\text{Key Q: how to choose } \beta \text{ to minimize } E?}$

~~want curve to converge at fixed point: $\frac{\partial C}{\partial t} = 0$~~

At fixed point (when curve converges)

$\rightarrow \frac{\partial C}{\partial t} = 0$ also $\rightarrow \frac{\partial}{\partial C} E = 0$

$\Rightarrow \frac{\partial}{\partial t} C = \underbrace{\frac{\partial}{\partial C} E}_{\beta N}$

How to compute RHS?

→ Euler Lagrange Eqns:

$$\frac{\partial E}{\partial C} = \frac{\partial F}{\partial u} - \frac{\partial}{\partial s}\frac{\partial F}{\partial u'} \quad (+\ u'', u''', \dots \text{ terms})$$

Example

$E(C) = \int_{s=0}^{s=1} |C'(s)|\,ds \Rightarrow$ minimize length of curve

$\Rightarrow \beta = K$ (curvature)

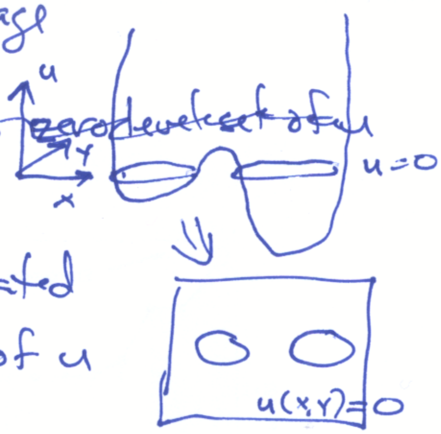
$\frac{dC}{dt} = KN$ curvature dependent flow "heat flow"

Another interesting case: β = constant

→ dilation / erosion

Level sets → key idea

Define function $u(x, y, t)$ ← time-varying image ~~such that C is zero level set of u~~
such that C is zero level set of u



Curve equation $\frac{\partial C}{\partial t} = \beta N$ can be reformulated

as $\frac{\partial u}{\partial t} = \beta \|\nabla u\|$ — gradient mag. of u

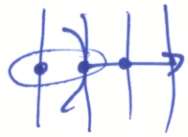
update to each "pixel" of u "level set eqn"

If you use this eqn to update u, every level set (curve) will evolve according to $\frac{\partial C}{\partial t} = \beta N$

That's it (basic idea)

Details

- discretization of level set eqn
  - → how to define derivs (central diff, left-diff, etc.)
  - → won't converge unless you're careful
    - → upwind scheme



- speedups
  - → don't update all of u, just a narrow band around C

  - → coarse-to-fine ...
- better integrators (Runge-Kutta ....)
- ... handling discontinuities, sharp corners
  - → Ross Whitaker

Snakes revisited

what is β ?

1) snake should shrink: $\beta = a$ (constant)
2) snake should glom onto edges $\beta = \underbrace{g_C}_{=0 \text{ at an edge}}$ e.g. $\frac{1}{1 + \|\nabla u\|}$
3) stay smooth: $\beta = K$

$\Rightarrow \beta = (K + a)g$

Not good enough for weak edges, or when ∇u varies over edge

→ ~~add a term to maximize change in edge strength~~

~~$\beta = (K+a)g + \nabla g \cdot \nabla u$~~

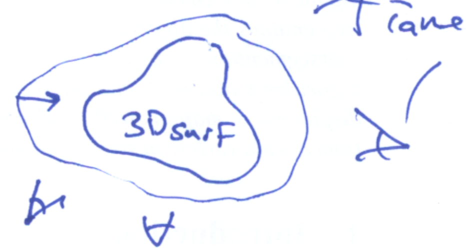
Let's go back to the energy fn we want:

$$E(C) = \int \underbrace{g_C(s)}_{\text{edge str.}} \underbrace{\frac{\partial C}{\partial s}(s)}_{\text{length}} ds$$

$$\Rightarrow \beta = (K + a)g + \underbrace{\nabla g \cdot \nabla u}_{\text{change in edge strength in direction of } N}$$

3D: multiview stereo - Faugeras & Keriven

4 cameras



Prob: given N images of a fixed object deform an initial surface in 3D to shrink-wrap onto object

Approach: define $E(S) = \int F(S, S', S'')\, d\vec{s}$

use Euler-Lagrange to compute β