Snakes (active contours), level sets
idea: segment using curves, not pixels

we want to segmentation curve that
1) conforms to image edges
2) is a smoothly varying curve vs lots of jagged nook and crannies

<u>snakes</u> are parametric curve

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} \quad s \in [0,1] \quad \text{cont} \qquad \text{eg. } \begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} r\cos(2\pi s) \\ r\sin(2\pi s) \end{bmatrix} \quad s \in [0,1]$$

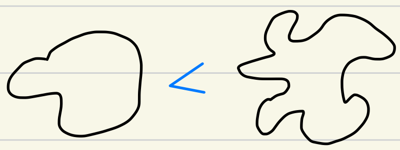
we want to define an energy function E(c) that matches our intuition about what makes a good segmentation curve will iteratively evolve to <u>reduce/minimize</u> E(c)

$E(c) = E_{internal}(c) + E_{external}(c)$

depend on the shape of curve

depends on image intensity

E_internal(c)



encapsulated by an energy like

$$E_{int}(c) = \int_0^1 \alpha \|c'(s)\|^2 + \beta \|c''(s)\|^2 \, ds$$

low c => not too "stretchy"
keeps points on the curve together as a function of s
we dont want it too non-uniform

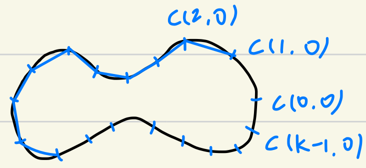
second part: low c" => not too "bendy", i.e smooth, keeps points on the curve from isolating

$$E_{ext}(c) = \int_0^1 -\|\nabla I(c(s))\|^2 \, ds = \int_0^1 -\left[\left(\frac{\partial I}{\partial x}(x(s), y(s))\right)^2 + \left(\frac{\partial I}{\partial y}(x(s), y(s))\right)^2\right] ds$$

Intuition: no edge => $\nabla I = 0 \Rightarrow F(s) = 0$

big edge => $\|\nabla I\|$ big => F(s) = big negative => low energy

how to minimize E(c)? :variation calculus
In practice for digital images, we solve the problem by creating a curve c(s, t), s, t are both discrete



curve approximation by k discrete points

$\{(x_i, y_i)\}$ then we step c(s,t-1) to c(s,t) by taking a step along gradient of E $\frac{\partial E}{\partial c}$

In practice, this results in a 2k*2k linear system solved at each iteration (bunch of finite difference
result: curve inches along until points around boundary stop changing
Lots of bookkeeping behind the scenes

problem with basic snake (E_ext):
- contour never "sees" strong edges that are far away
- image noise => many small gradient => snake gets hung up
soln to this: gradient vector flow
idea: instead of using exactly the image gradient, we create a new vector field over the image plane V(x,y) $= \begin{bmatrix} V_x(x,y) \\ V_y(x,y) \end{bmatrix}$

e(x,y) edge map of image
V is defined to minimize

$$\iint \mu\left[\left(\frac{\partial V_x}{\partial x}\right)^2 + \left(\frac{\partial V_x}{\partial y}\right)^2 + \left(\frac{\partial V_y}{\partial x}\right)^2 + \left(\frac{\partial V_y}{\partial y}\right)^2\right] + \|\nabla e\|^2 \|V - \nabla e\|^2 \, dx\,dy$$

Smoothness of V

magnitude of edge map

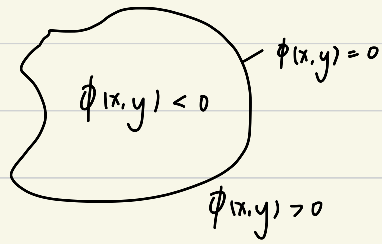
Similarity b/w V and ∇e

μ: tuning parameter, intuition: ∇e is big => gradient is large and V follow edge gradient faithfully. if ∇e is small => gradient is small, v follows along to be as smooth as possible
μ trades off how smooth vs how faithful

snakes defined as a set of point around a contour can be a pain
- keeping track of # points, point distribution
- get point to probe into concave areas
- snake can never wrap around multiple objects at once
- snake can't do holes

solution: level sets
instead of parametrize curve by set of ordered points, discretize image plane (x,y) and define a function ϕ(x,y). Evolve this entire function; pixels where ϕ(x,y)=0 implicitly define the object we care about (multiple objects, holes)

C = {(x,y)| ϕ(x,y) = 0}



$\phi(x,y) = \sqrt{x^2+y^2} - r$, $\phi(x,y) = 0 \Rightarrow \sqrt{x^2+y^2} = r$ circle of radius = r

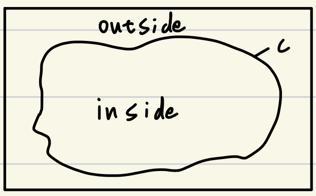
evolving level set function is mathematically better behaved

$\phi(x,y,t) = 0$

many picky details

iso-surface algorithm to extract the final segmentation

everything so far - edge based
in absence of strong edges, we can use a region based formulation



$$E(c) = \lambda_1 \int_{\text{inside}} (I(x,y) - \mu_{in})^2 \, dx\,dy + \lambda_2 \int_{\text{outside}} (I(x,y) - \mu_{out})^2 \, dx\,dy + \lambda_3 (\text{length of } c)$$

$$+ \lambda_4 (\text{area inside } c)$$

chain-vese algorithm