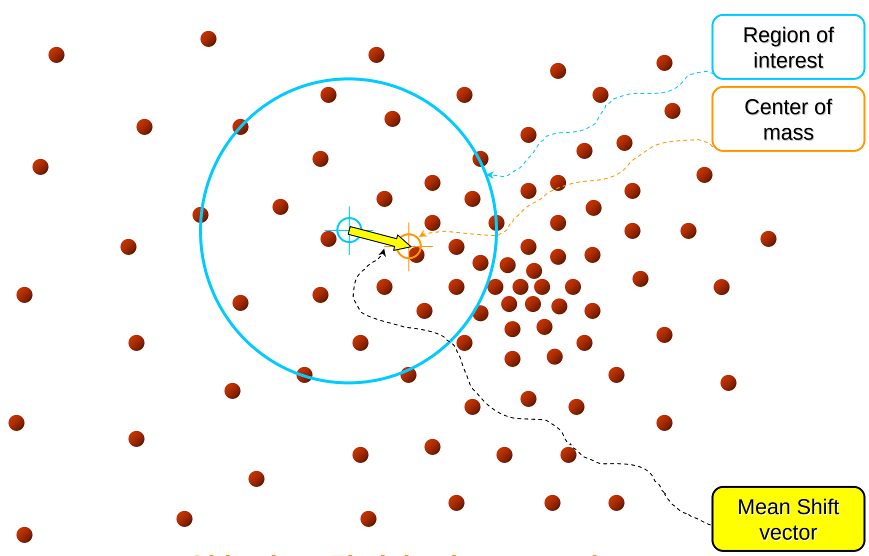
CAP6411
Computer Vision Systems
Lecture 14

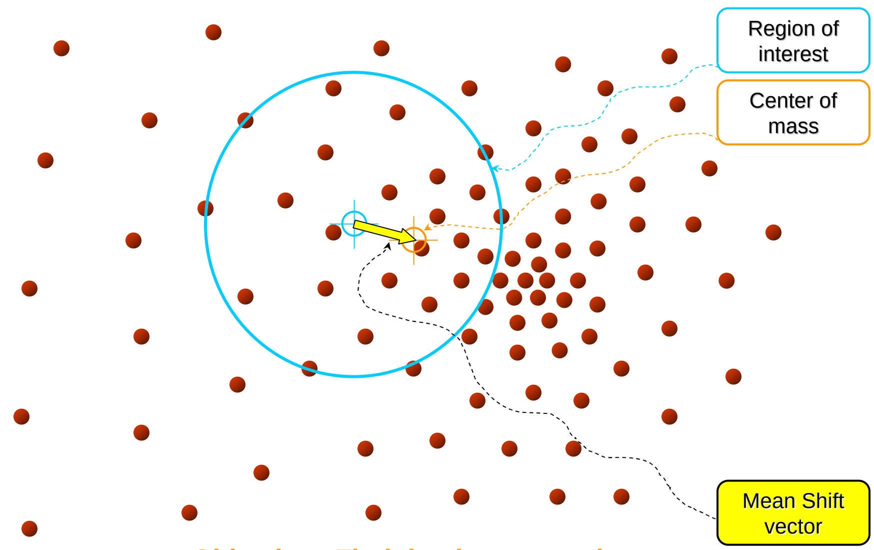
Alper Yilmaz
Office: CSB 250
Email: yilmaz@cs.ucf.edu
Web: [http://www.cs.ucf.edu/courses/**cap6411/cap6411**/spring2006](http://www.cs.ucf.edu/courses/cap6411/cap6411/spring2006)

Recap



Objective : Find the densest region

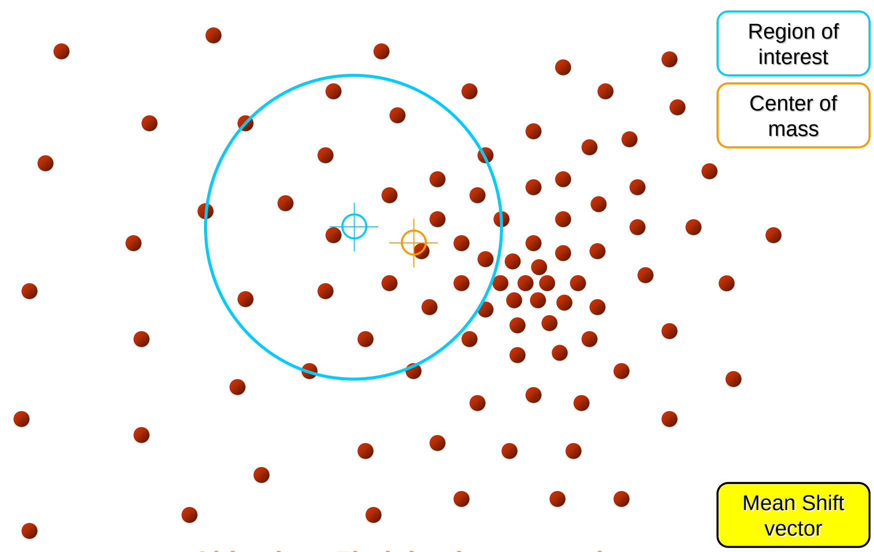
Recap



Objective : Find the densest region

Distribution of identical billiard balls

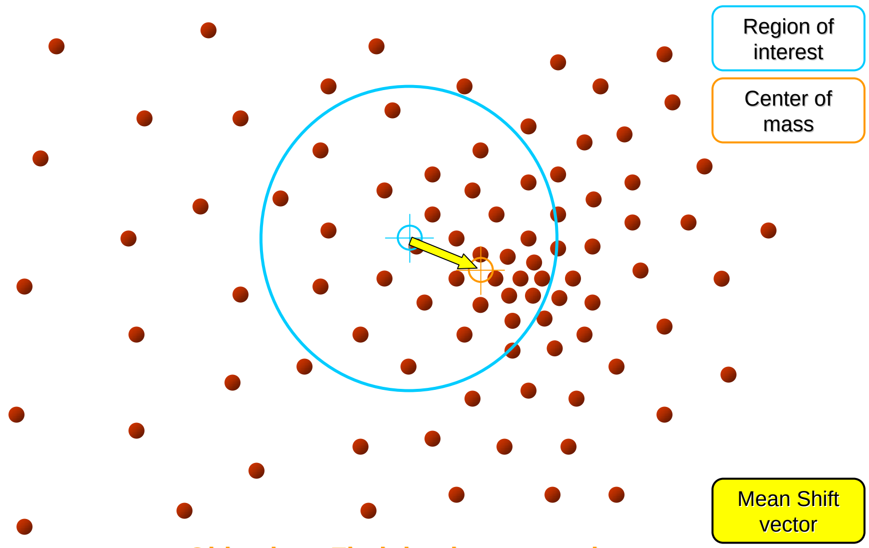
Recap



Objective : Find the densest region

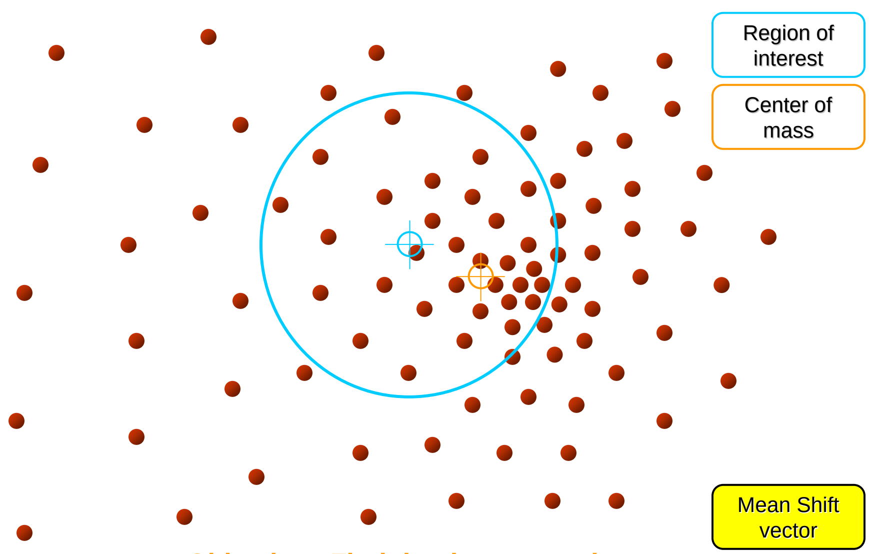
Distribution of identical billiard balls

Recap



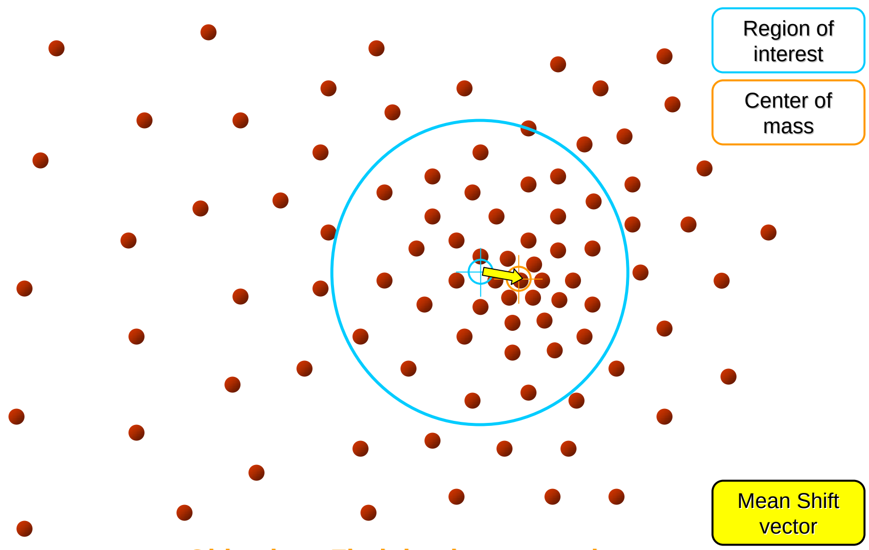
Objective : Find the densest region
Distribution of identical billiard balls

Recap



Objective : Find the densest region
Distribution of identical billiard balls

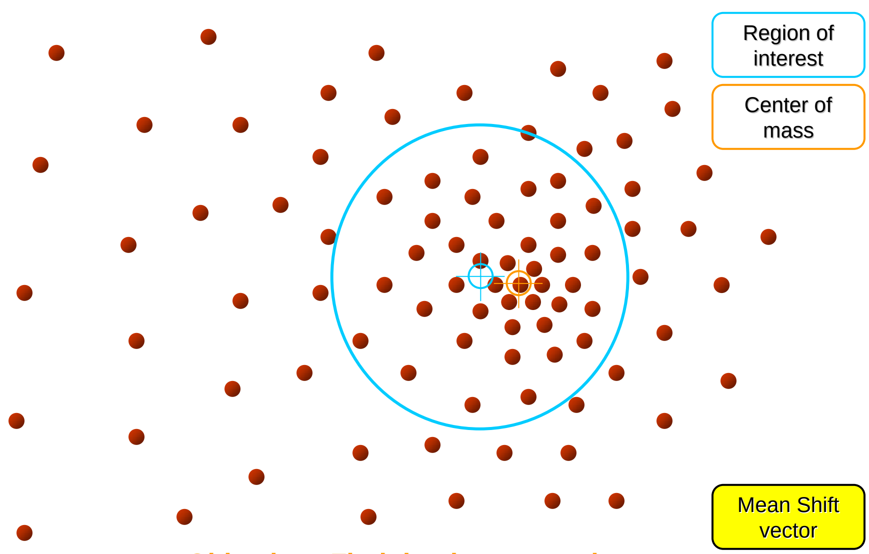
Recap



Objective : Find the densest region

Distribution of identical billiard balls

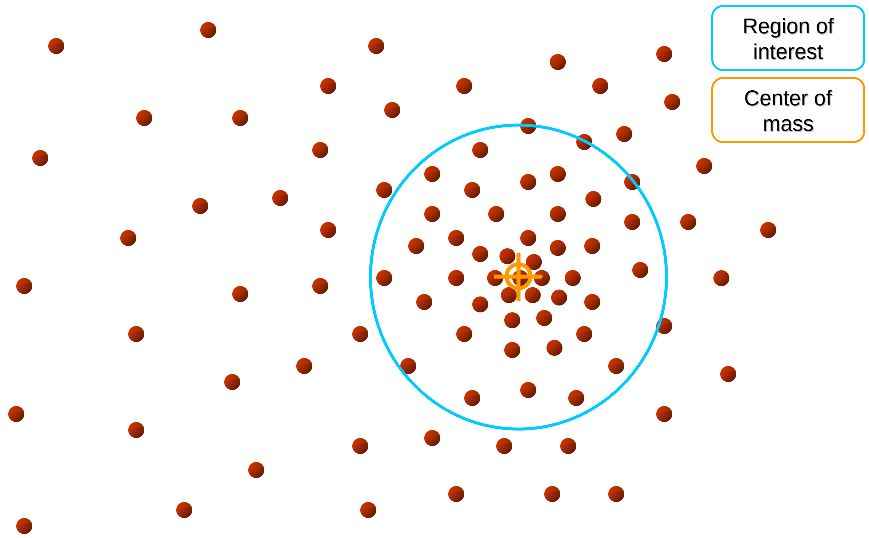
Recap



Objective : Find the densest region

Distribution of identical billiard balls

Recap



Objective : Find the densest region

Distribution of identical billiard balls

Active Contours a.k.a. Snakes

Goal

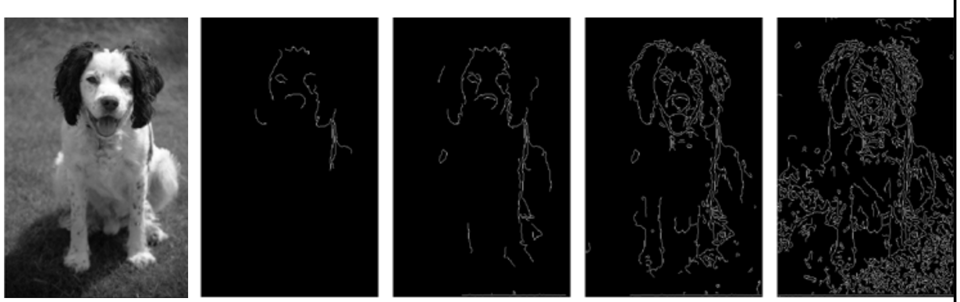
- Segmentation of images
  - Used in other contexts, i.e. registration
- Find a closed or open boundary between regions

Overview

- Minimize some energy
- Boundary based (BB) methods
- Region based (RB) methods
- Combination of BB and RB

Boundary from edge detection

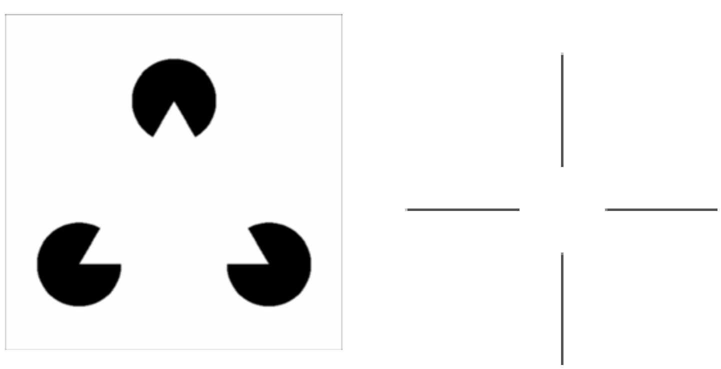
Does not always work



How to improve

- Integrate information over image plane
- Use Gestalt cues
  - Smoothness
  - Closure
- Get operator to help!

How can we integrate additional information

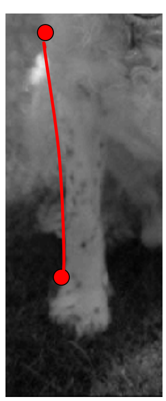


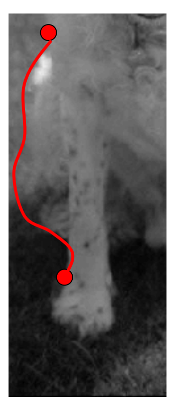
Humans integrate high level knowledge

A simpler problem

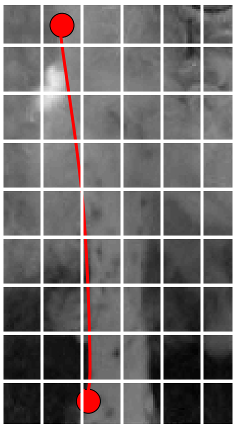
Let’s find the best path between two boundary points.

Which path is the best?





Discrete grid



- Contour should be near edge.
  - Strength of gradient.
- Contour should be smooth (good continuation).
  - Low curvature
  - Low change of direction of gradient.

Smoothness

- *Discrete Curvature:* if you go from p(j-1) to p(j) to p(j+1) how much did direction change?
  - Be careful with discrete distances.
- Change of direction of gradient from p(j-1) to p(j)

Related cost function

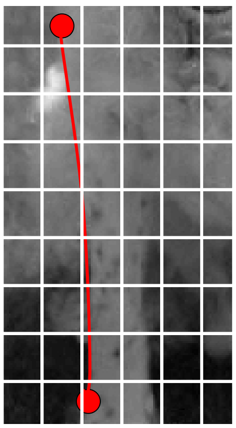
- Path: p(1), p(2), … p(n).

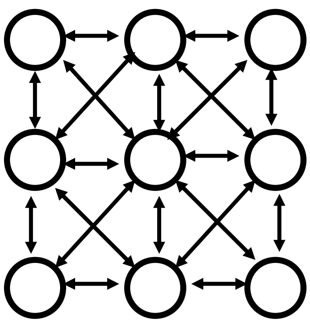
$$\sum_{j=1}^{n} d(p(j), p(j+1)) * \left[g(p(j) + \lambda f(p(j-1), p(j) \right]$$

- where
  - d(.) is distance between consecutive grid points ie, 1 or sqrt(2).
  - g(.) measures strength of gradient
  - λ is some parameter
  - f measures smoothness, curvature.

How to solve…

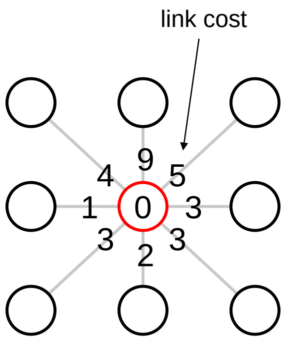
- Mapping the problem to graph





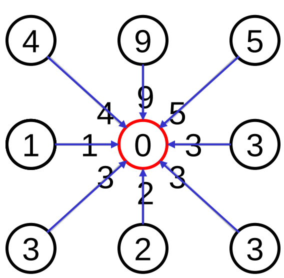
Weight represents cost of going from one pixel to another. Next term in sum.

Dijkstra's shortest path algorithm



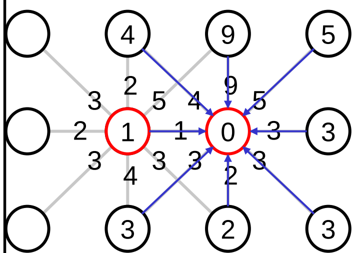
- init node costs to , set p = seed point, cost(p) = 0
- expand p as follows:
  - for each of p's neighbors q that are not expanded
    - set cost(q) = min(cost(p) + cpq, cost(q))

Dijkstra's shortest path algorithm



- init node costs to , set p = seed point, cost(p) = 0
- expand p as follows:
  - for each of p's neighbors q that are not expanded
    - set cost(q) = min(cost(p) + cpq, cost(q))
      - if q's cost changed, make q point back to p
    - put q on the ACTIVE list (if not already there)

Dijkstra’s shortest path algorithm



- init node costs to , set p = seed point, cost(p) = 0
- expand p as follows:
  - for each of p’s neighbors q that are not expanded
    - set cost(q) = min(cost(p) + cpq, cost(q))
      - if q’s cost changed, make q point back to p
    - put q on the ACTIVE list (if not already there)
- set r = node with minimum cost on the ACTIVE list
- repeat Step 2 for p = r

Intelligent scissors

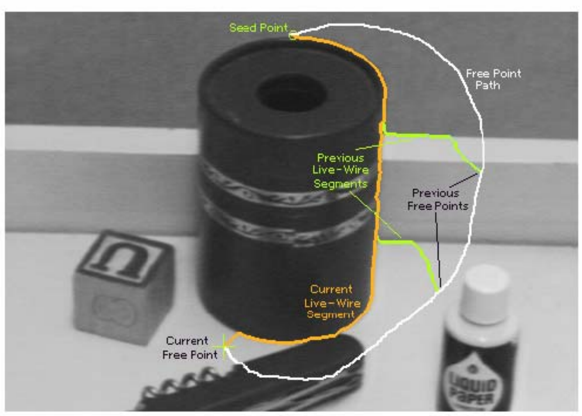
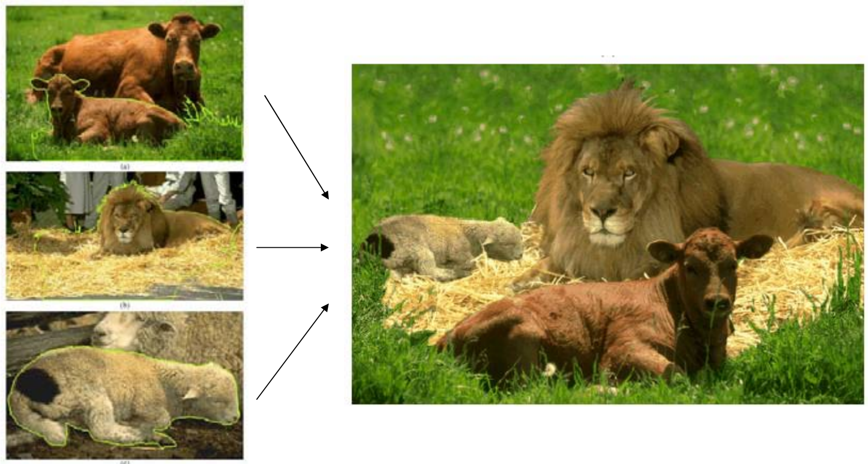


Figure 2: *Image demonstrating how the live-wire segment adapts and snaps to an object boundary as the free point moves (via cursor movement). The path of the free point is shown in white. Live-wire segments from previous free point positions (t_0, t_1, and t_2) are shown in green.*

Blending



A harder problem

- Deformable models
  - Internal forces (prior)
  - External forces (observation, image)
  - (1973) Widrow’s “rubbermask”
  - (1987) Kass, Witkin and Terzopoulos

Two types of deformable models

- Parametric: curve **x**(*s*), *s* = [0,1] is parameter, or polyline **x***i*, *i*=1,…,N, *i* is parameter
  - Snakes (Here is some confusion: Snakes are also called "nonparametric" in the sense that very little prior is included)
  - Active shape models (Sometimes called "parametric" because the shape is trained on examples using a PCA model)
- Geometric
  - Level sets

Evolution of contour

- Snake Model (1987) [Kass-Witkin-Terzopoulos]
  - Planar parameterized curve C:R→RxR
  - A cost function defined along that curve

$$E[(C)(p)] = \alpha \int_0^1 E_{int}(C(p))dp \; + \; \beta \int_0^1 E_{img}(C(p))dp \; + \; \gamma \int_0^1 E_{con}(C(p))dp$$

Image Energies

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))\,ds$$
$$= \int_0^1 E_{\text{int}}(v(s))\,ds + \int_0^1 E_{image}(v(s))\,ds + \int_0^1 E_{ext}(v(s))\,ds$$

- The internal term stands for regularity/smoothness along the curve
- The image term guides the active contour towards the desired image properties
- The external term can be used to account for user-defined constraints, or prior knowledge`
- The lowest potential of such a cost function refers to an equilibrium of these terms

The internal term

$$E_{int}(C(p)) = w_{tension}(C(p)) \left|\frac{\partial C}{\partial p}(p)\right|^2 + w_{stiffness}(C(p)) \left|\frac{\partial^2 C}{\partial p^2}(p)\right|^2$$

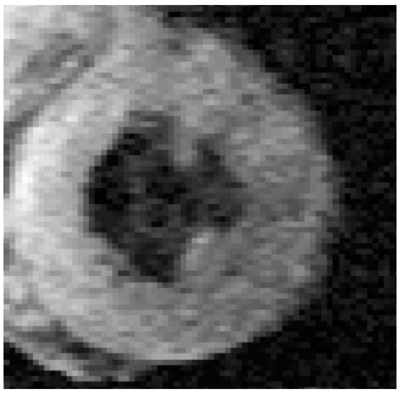
- The first order derivative makes the snake behave as a membrane
- The second order derivative makes the snake act like a thin plate

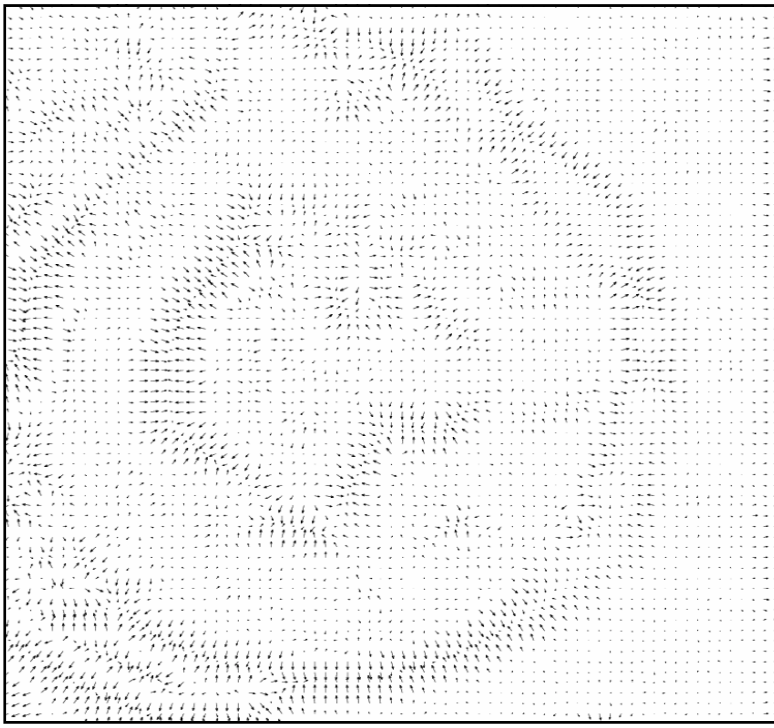
The image term

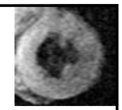
$$E_{img}(C(p)) = w_{line}E_{line}(C(p)) + w_{edge}E_{edge}(C(p)) + w_{term}E_{term}(C(p))$$

- Can guide the snake to
  - Iso-photes $E_{line}(C(p)) = I(C(p))$
  - Edges $E_{edge}(C(p)) = |\nabla I(C(p))|^2$
  - Terminations

Relation between image and energy







Evolution Forces

In Summary

- Internal forces
  - Holds curve together
  - Prevents it from bending
- External forces
  - Attract curve to edges
- Analogous to Tichonov Regularisation (see Li: Markov Random Field, page 38)
  - First two terms = prior energy,
  - Third term = likelihood energy

$$E(X)=\int_0^1[\frac{1}{2}\alpha(s)X_s^2+\frac{1}{2}b(s)X_{ss}^2+P(X(s))]ds$$

Example forces

$$P(x,y) = -w\,|\nabla[G_\sigma(x,y) * I(x,y)]|^2$$ Edges

$$P(x,y) = -w\,|G_\sigma(x,y) * I(x,y)|$$ Iso-phote

- Start with large sigma, decrease after some iterations.

Minimization
Variational Calculus

$$x, \eta : [0,1] \to \Re^2$$

$$x(0) = x_0$$

$$x(1) = x_1$$

$$\eta(0) = \eta(1)$$

$$\eta'(0) = \eta'(1)$$

$$C(x) = \int_0^1 F(x, x', x'')\,ds$$

$$C(x + \varepsilon\eta) = \int_0^1 F(x + \varepsilon\eta, x' + \varepsilon\eta', x'' + \varepsilon\eta'')\,ds$$

$$C(x+\varepsilon\eta)=\int_0^1 F(x+\varepsilon\eta, x'+\varepsilon\eta', x''+\varepsilon\eta'')ds$$

Minimization Variational Calculus

$$\frac{\partial C}{\partial \varepsilon}(x+\varepsilon\eta)=\int_0^1\left(\frac{\partial F}{\partial x}\eta+\boxed{\frac{\partial F}{\partial x'}\eta'}+\boxed{\frac{\partial F}{\partial x''}\eta''}\right)ds$$

Chain rule

$$\boxed{\int_0^1\frac{\partial F}{\partial x'}\eta'\,ds}=\frac{\partial F}{\partial x'}\eta\bigg|_0^1-\int_0^1\left(\frac{\partial}{\partial s}\frac{\partial F}{\partial x'}\right)\eta ds=-\int_0^1\left(\frac{\partial}{\partial s}\frac{\partial F}{\partial x'}\right)\eta ds$$

Integration by parts

$$\boxed{\int_0^1\frac{\partial F}{\partial x''}\eta''\,ds}=\frac{\partial F}{\partial x''}\eta'\bigg|_0^1-\int_0^1\left(\frac{\partial}{\partial s}\frac{\partial F}{\partial x''}\right)\eta'\,ds=\boxed{-\int_0^1\left(\frac{\partial}{\partial s}\frac{\partial F}{\partial x''}\right)\eta'\,ds}$$

$$\boxed{\int_0^1\left(\frac{\partial}{\partial s}\frac{\partial F}{\partial x''}\right)\eta'\,ds}=\left(\frac{\partial}{\partial s}\frac{\partial F}{\partial x''}\right)\eta\bigg|_0^1-\int_0^1\left(\frac{\partial^2}{\partial s^2}\frac{\partial F}{\partial x''}\right)\eta ds=-\int_0^1\left(\frac{\partial^2}{\partial s^2}\frac{\partial F}{\partial x''}\right)\eta ds$$

$$\frac{\partial C}{\partial \varepsilon}(x+\varepsilon\eta)=\int_0^1\left(\frac{\partial F}{\partial x}-\frac{\partial}{\partial s}\frac{\partial F}{\partial x'}+\frac{\partial^2}{\partial s^2}\frac{\partial F}{\partial x''}\right)\eta ds$$

$$\frac{\partial C}{\partial \varepsilon}(x+\varepsilon\eta)=0,\ \forall\,\eta \quad\Rightarrow\quad \frac{\partial F}{\partial x}-\frac{\partial}{\partial s}\frac{\partial F}{\partial x'}+\frac{\partial^2}{\partial s^2}\frac{\partial F}{\partial x''}=0$$

Euler

- Euler Equation from variational calculus (assume closed curve)

$$\frac{\partial}{\partial s}(\alpha X_s)-\frac{\partial^2}{\partial s^2}(\beta X_{ss})-\nabla P(X)=0$$

- F(internal) + F(external) = 0

Evolving Contour

$$\alpha\left(w_{tension}\frac{\partial^2 C}{\partial p^2}(p) - w_{stiffness}\frac{\partial^4 C}{\partial p^4}(p)\right) - \beta\nabla E_{img}(C(p)) = 0$$

- is used to update the position of an initial curve towards the desired image properties
  - Initial the curve, using a certain number of **control points** as well as a set of basic functions,
  - **Update** the positions of the control points by solving the above equation
  - **Re-parameterize** the evolving contour, and continue the process until convergence of the process…

Pros and Cons

- Pros
  - Low complexity
  - Easy to introduce prior knowledge
  - Can account for open as well as closed structures
  - A well established technique, numerous publications it works
  - User Interactivity
- Cons
  - Selection on the parameter space and the sampling rule affects the final segmentation result
  - Estimation of the internal geometric properties of the curve in particular higher order derivatives
  - Quite sensitive to the initial conditions,
  - Changes of topology

Results