اصول پردازش تصویر

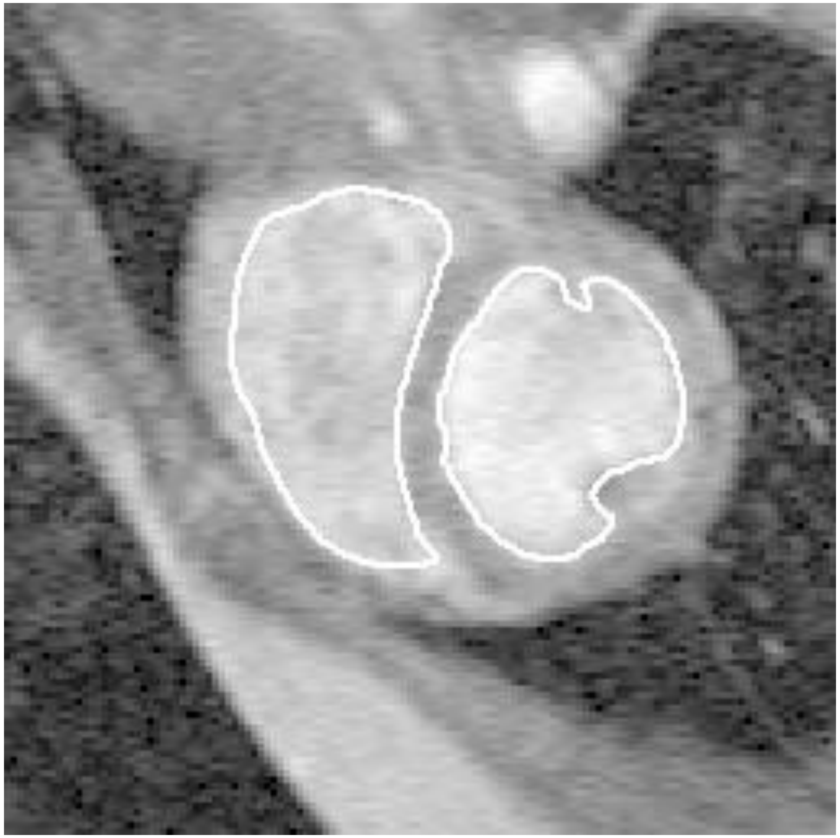
Principles of Image Processing

مصطفی کمالی تبریزی

۱۹ آبان ۱۳۹۹

جلسه پانزدهم

Deformable Contours



Deformable Contours

a.k.a. active contours, snakes

Given: initial contour (model) near desired object

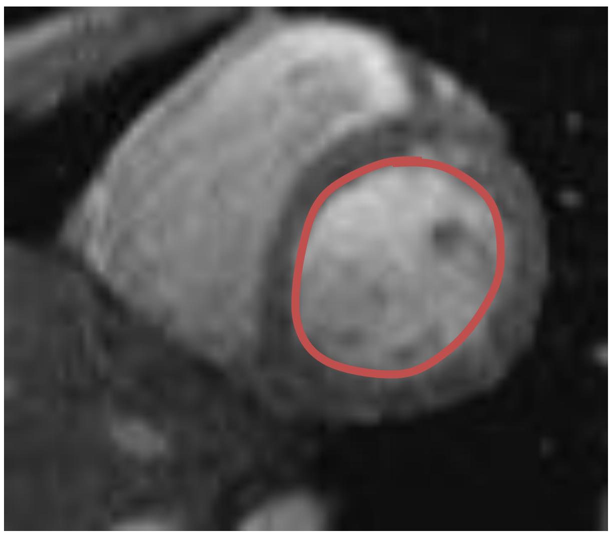
[**Snakes: Active contour models**; Kass, Witkin, & Terzopoulos, ICCV1987]

Deformable Contours

a.k.a. active contours, snakes

Given: initial contour (model) near desired object

Goal: evolve the contour to fit exact object boundary

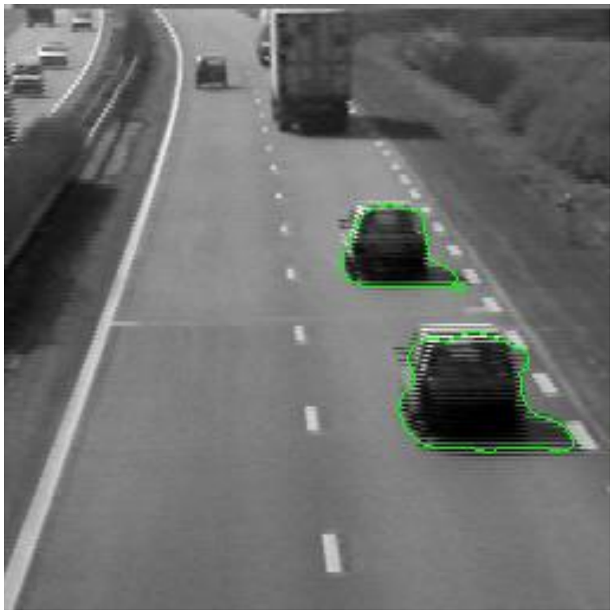
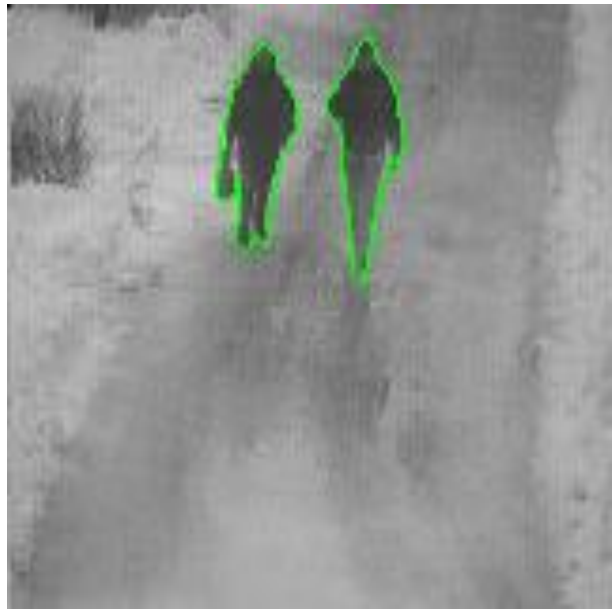
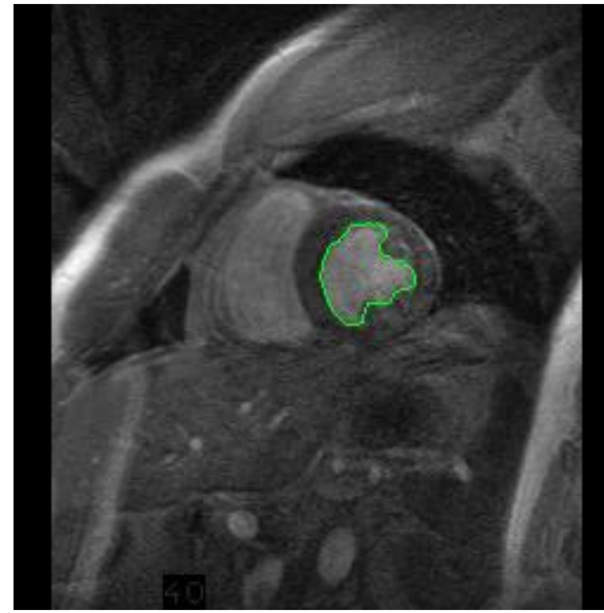


Main idea: elastic band is iteratively adjusted so as to

- be near image positions with high gradients, **and**
- satisfy shape "preferences" or contour priors

[**Snakes: Active contour models**; Kass, Witkin, & Terzopoulos, ICCV1987]

Why Deformable Shapes?

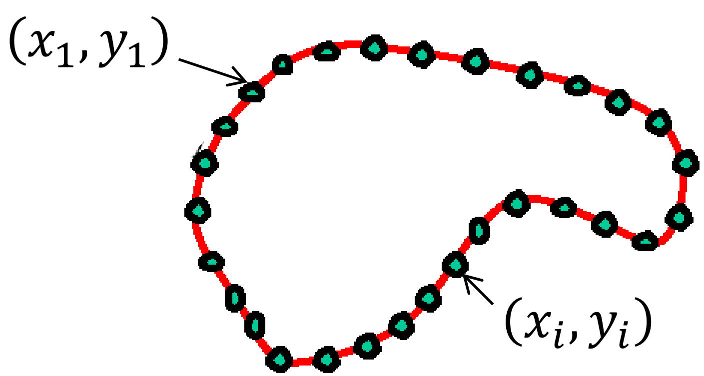




Non-rigid, deformable objects can
change their shape over time.

Slide Credit: Julien Jomier

Representation

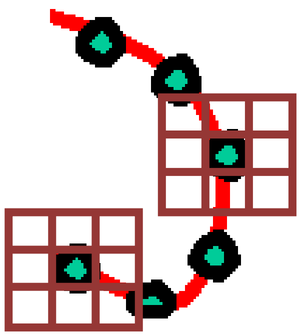
- We'll consider a discrete representation of the contour, consisting of a list of $2d$ point positions ("vertices").



$$v_i = (x_i, y_i)$$

for $i = 1, \dots, n$

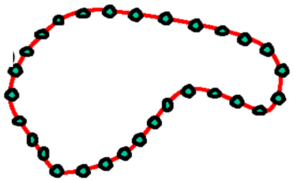
- At each iteration, we'll have the option to move each vertex to another nearby location ("state").



Energy Function

The total energy (cost) of the current snake is defined as:

$$E_{total} = E_{internal} + \gamma\, E_{external}$$

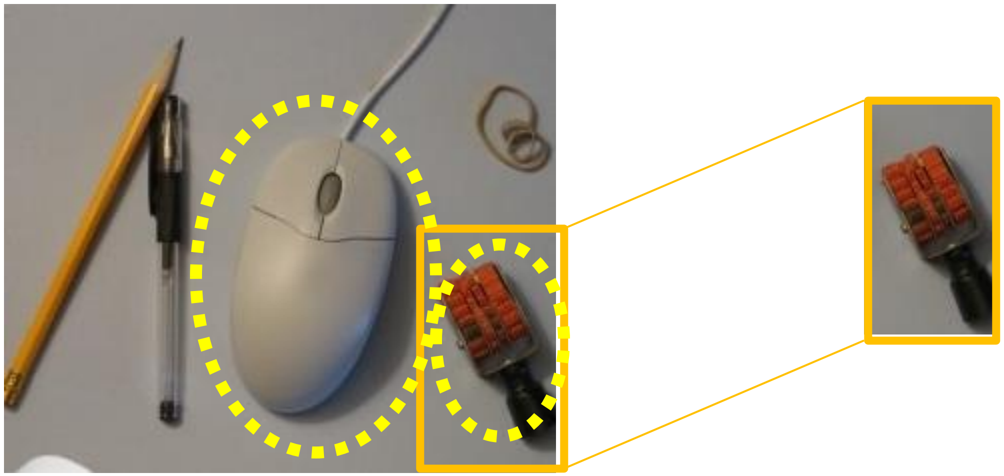


- **Internal** energy: encourage *prior* shape preferences: e.g., smoothness, elasticity, particular known shape.
- **External** energy ("image" energy): encourage contour to fit on places where image structures exist, e.g., edges.
- A good fit between the current deformable contour and the target shape in the image will yield a **low** value for this cost function.

External Energy: Intuition

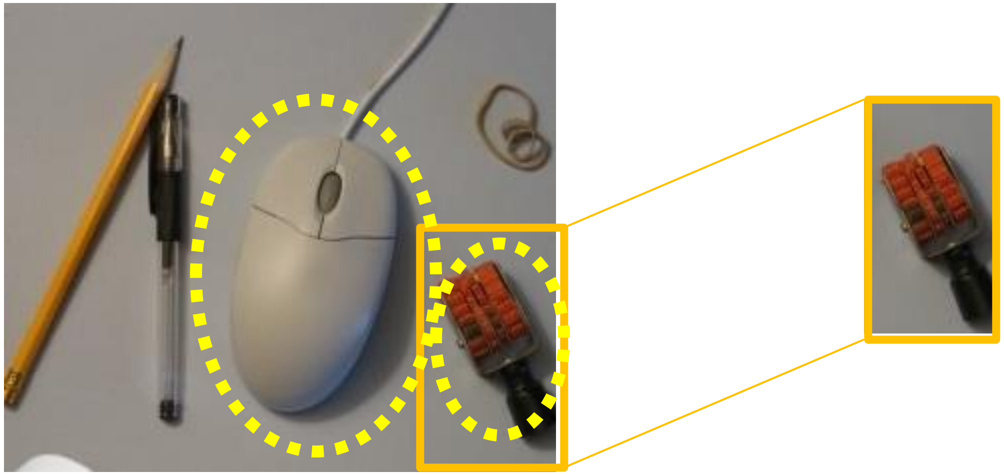
- Measure how well the curve matches the image data
- “Attract” the curve towards different image features
  - Edges, lines, texture gradient, etc.

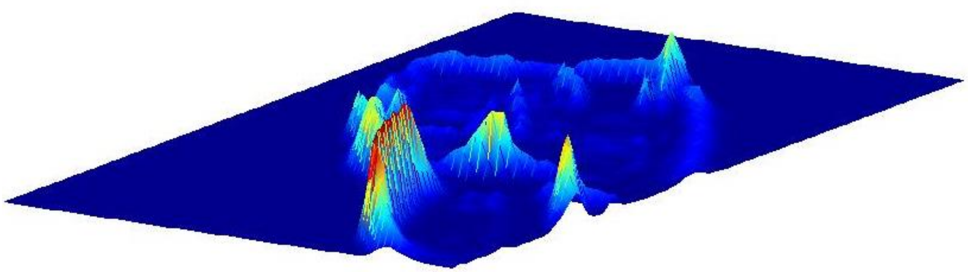
External Image Energy



Slide Credit: Kristen Grauman

External Image Energy



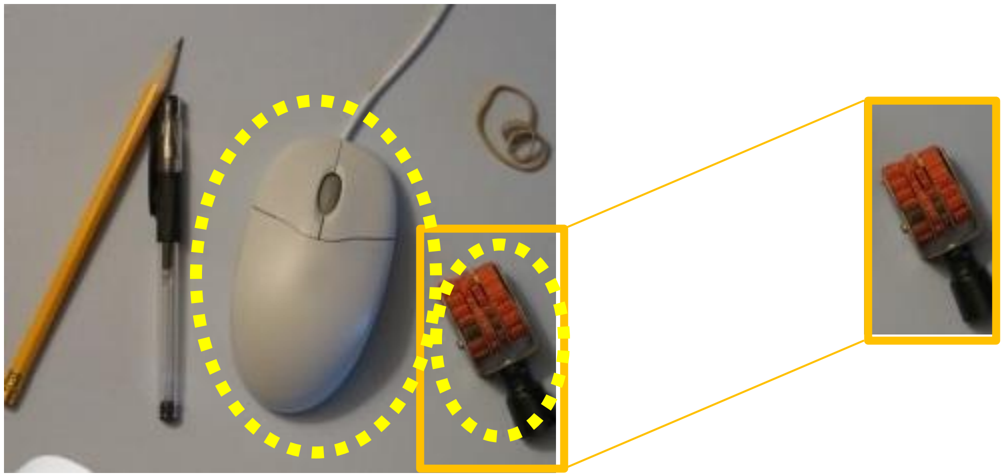


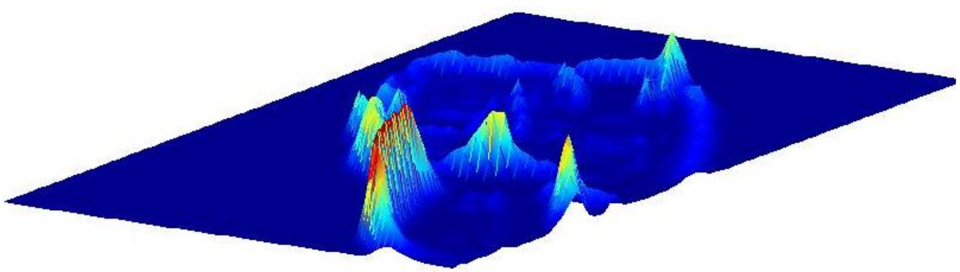
Magnitude of gradient

$$G_x(I)^2 + G_y(I)^2$$

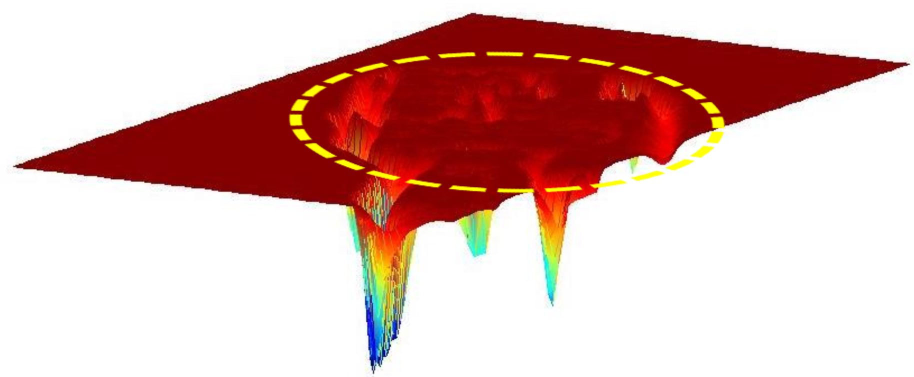
Slide Credit: Kristen Grauman

External Image Energy





Magnitude of gradient
$G_x(I)^2 + G_y(I)^2$



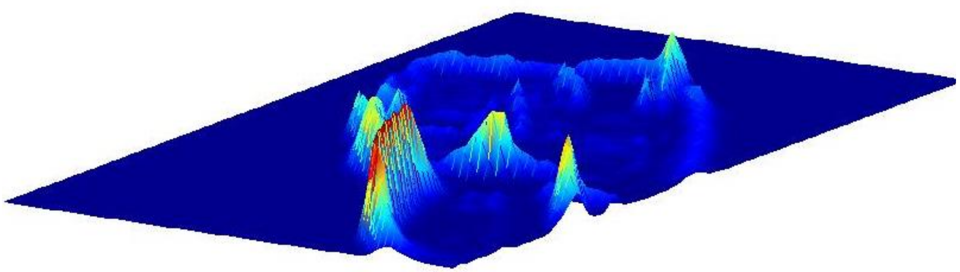
− (Magnitude of gradient)
$-(G_x(I)^2 + G_y(I)^2)$

Slide Credit: Kristen Grauman

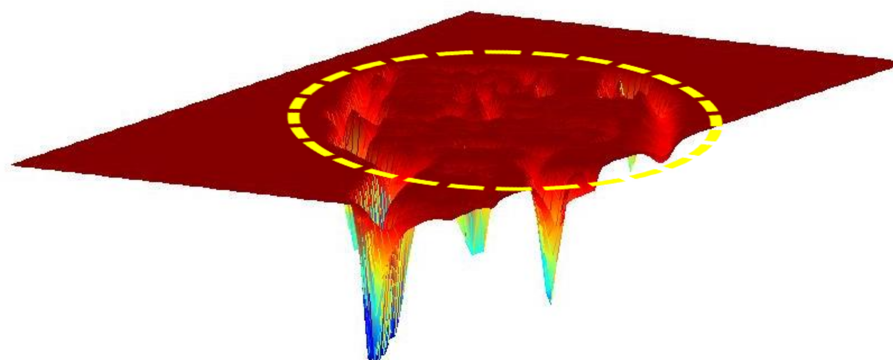
External Image Energy

How do edges affect "snap" of rubber band?

Think of external energy from image as gravitational pull towards areas of high contrast



Magnitude of gradient
$G_x(I)^2 + G_y(I)^2$

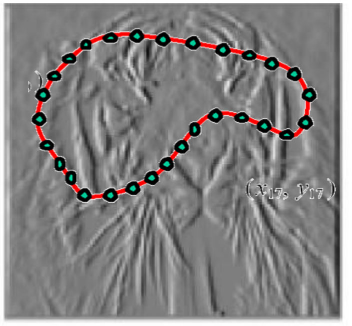


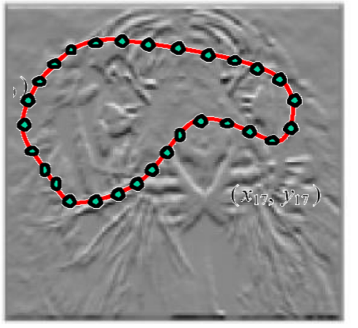
− (Magnitude of gradient)
$-(G_x(I)^2 + G_y(I)^2)$

Slide Credit: Kristen Grauman

External Image Energy

- Gradient images $G_x(x, y)$ and $G_y(x, y)$





- External energy at a point on the curve is:

$$E_{external}(v_i) = -\left(G_x(v_i)^2 + G_y(v_i)^2\right)$$

- External energy for the whole curve:

$$E_{external} = -\sum_{i=1}^{n}\left(G_x(x_i, y_i)^2 + G_y(x_i, y_i)^2\right)$$

Slide Credit: Kristen Grauman

Internal Energy: Intuition

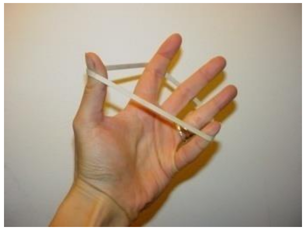
A priori, we want to favor **smooth** shapes, contours with **low curvature**, contours similar to **a known shape**, etc. to balance what is actually observed (i.e., in the gradient image).

For a *continuous* curve, a common internal energy term is the "bending energy". At some point $v(s)$ on the curve, this is:

$$E_{internal}\big(v(s)\big) = \alpha \boxed{\left|\frac{dv}{ds}\right|^2} + \beta \boxed{\left|\frac{d^2v}{ds^2}\right|^2}$$

Tension, Elasticity

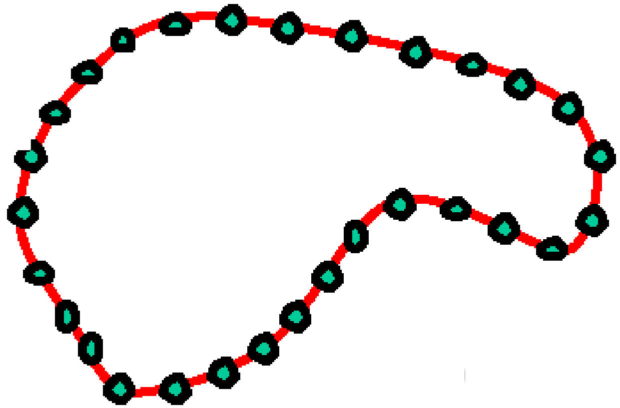
Stiffness, Curvature



Slide Credit: Kristen Grauman

Internal Energy

$v_i = (x_i, y_i) \qquad i = 1, \ldots, \mathrm{n}$



$$\frac{dv}{ds} \approx v_{i+1} - v_i$$

$$\frac{d^2v}{ds^2} \approx (v_{i+1} - v_i) - (v_i - v_{i-1}) = v_{i+1} - 2v_i + v_{i-1}$$

*Note these are derivatives relative to **position**---not spatial image gradients.*

$$E_{internal} = \alpha \sum_{i=1}^{n} \|v_{i+1} - v_i\|^2 + \beta \sum_{i=1}^{n} \|v_{i+1} - 2v_i + v_{i-1}\|^2$$

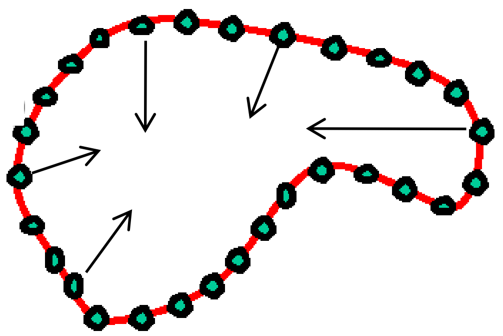
Slide Credit: Kristen Grauman

Penalizing Elasticity

- Current elastic energy definition uses a discrete estimate of the derivative:

$$E_{elastic} = \alpha \sum_{i=1}^{n} \|v_{i+1} - v_i\|^2$$

$$= \alpha \sum_{i=1}^{n} \left((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \right)$$



What is the possible problem with this definition?

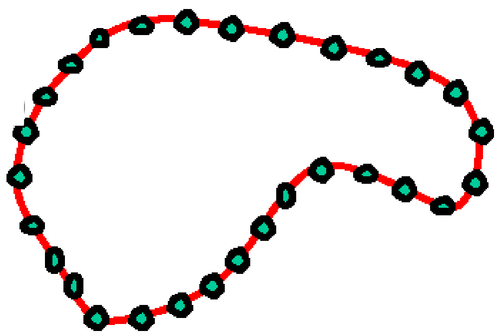
Slide Credit: Kristen Grauman

Penalizing Elasticity

- Current elastic energy definition uses a discrete estimate of the derivative:

$$E_{elastic} = \cancel{\alpha \sum_{i=1}^{n} \|v_{i+1} - v_i\|^2}$$

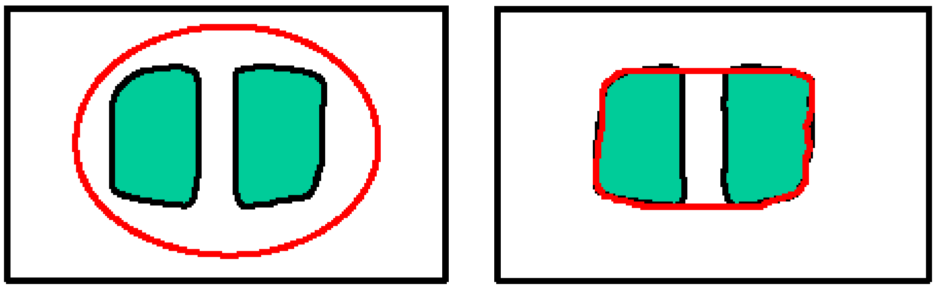
instead $$= \alpha \sum_{i=1}^{n} \left((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 - \bar{d}^{2} \right)^2$$

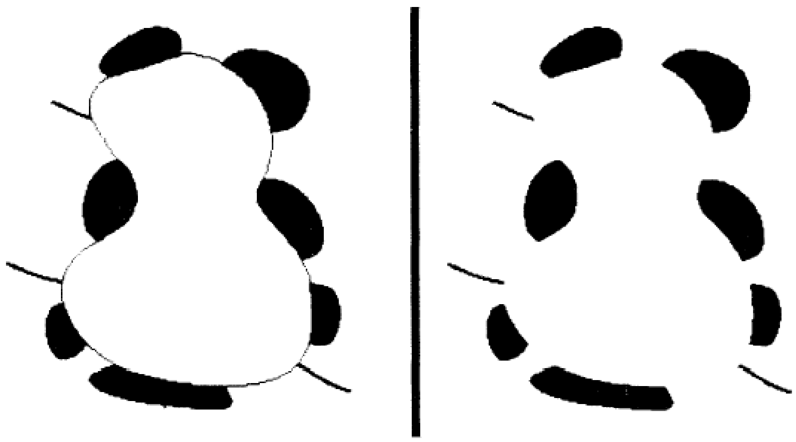


where $\bar{d}$ is the average distance between pairs of points, updated at each iteration.

Dealing with Missing Data

- The preferences for low-curvature, smoothness help deal with missing data:



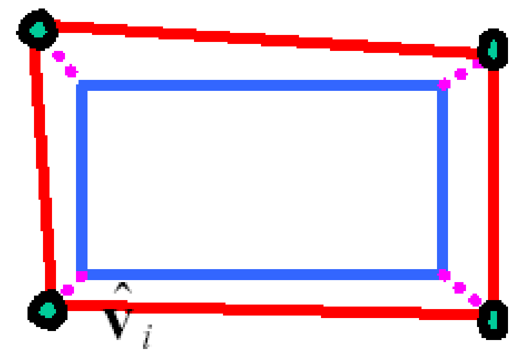


Illusory contours found!

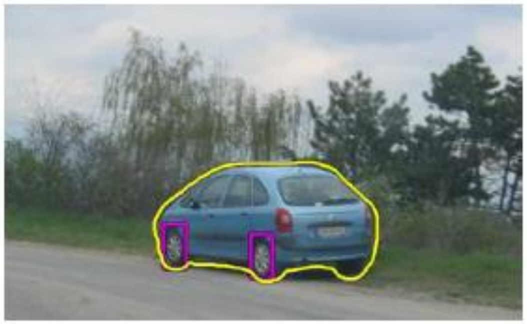
[Figure from Kass et al. 1987]

Extending the Internal Energy: Capture Shape Prior

If the object is some smooth variation of a known shape, we can use a term that will penalize deviation from that shape:



$$E_{internal} \mathrel{+}= \mu \sum_{i=1}^{n} (v_i - \hat{v}_i)^2$$



where $\{\hat{v}_i\}$ are the points of the known shape.

Slide Credit: Yuri Boykov

Total Energy: Function of the Weights

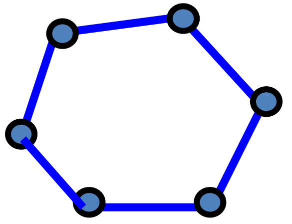
$$E_{total} = E_{internal} + \gamma E_{external}$$

$$E_{external} = -\sum_{i=1}^{n} \left(G_x(x_i, y_i)^2 + G_y(x_i, y_i)^2 \right)$$

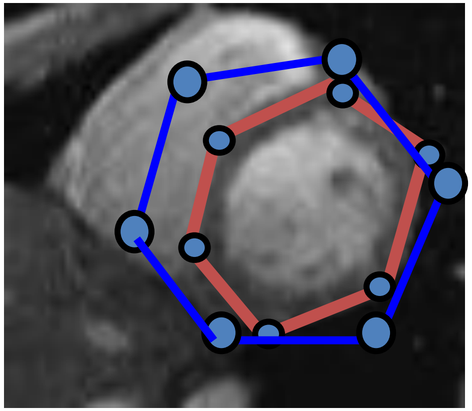
$$E_{internal} = \alpha \sum_{i=1}^{n} \left(\|v_{i+1} - v_i\|^2 - \bar{d} \right)^2 + \beta \sum_{i=1}^{n} \|v_{i+1} - 2v_i + v_{i-1}\|^2$$

Recap: Deformable Contour

- A simple elastic snake is defined by:
  - A set of n points,
  - An internal energy term (tension, bending, plus optional shape prior)
  - An external energy term (gradient-based)



- To use to segment an object:
  - Initialize in the vicinity of the object
  - Modify the points to minimize the total energy



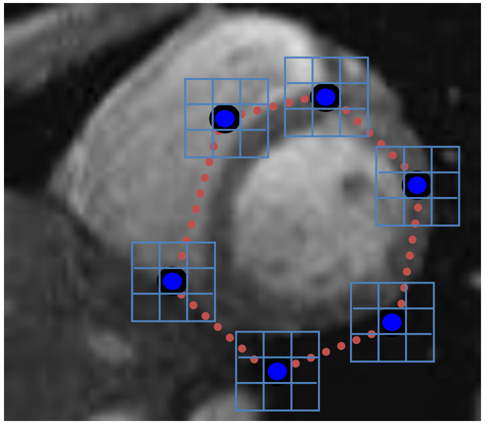
Slide: Kristen Grauman

Energy Minimization

- Several algorithms have been proposed to fit deformable contours.
- We'll look at two:
  - Greedy search
  - Dynamic programming (for 2d snakes)

Energy Minimization: Greedy

- For each point, search window around it and move to where energy function is minimal
  - Typical window size, e.g., 5 x 5 pixels

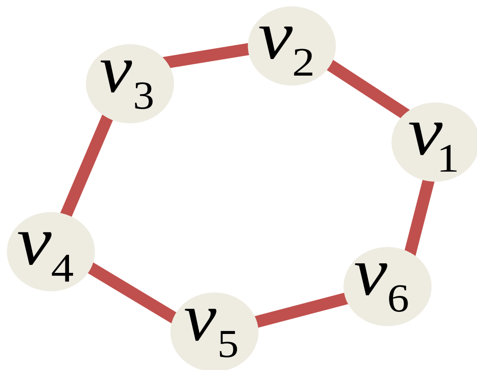


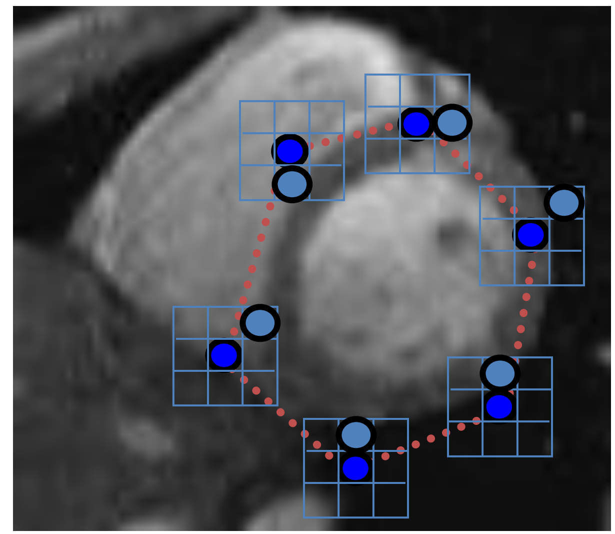
- Stop when predefined number of points have not changed in last iteration, or after max number of iterations

- Note:
  - Convergence not guaranteed
  - Need decent initialization

Slide: Kristen Grauman

Energy Minimization: Dynamic Programming





With this form of the energy function, we can minimize using dynamic programming, with the *Viterbi* algorithm.

Iterate until optimal position for each point is the center of the box, i.e., the snake is optimal in the local search space constrained by boxes.

[Amini, Weymouth, and Jain, 1990] - Slide Credit: Yuri Boykov

Energy Minimization: Dynamic Programming

Possible because snake energy can be rewritten as a sum of triple-interaction potentials:

$$E_{total}(v_1, \dots, v_n) = \sum_{i=1}^{n} E_i(v_{i-1}, v_i, v_{i+1})$$

or sum of pair-wise interaction potentials:

$$E_{total}(v_1, \dots, v_n) = \sum_{i=1}^{n} E_i(v_i, v_{i+1})$$

Snake Energy: Pair-Wise Interactions

$$E_{total}(v_1, \ldots, v_n) = -\gamma \sum_{i=1}^{n} \left(G_x(v_i\ )^2 + G_y(v_i)^2\right)\ + \alpha \sum_{i=1}^{n} \left(\|v_{i+1} - v_i\|^2 - \bar{d}\right)^2$$

$$E_{total}(v_1, \ldots, v_n) = \sum_{i=1}^{n} \left(-\gamma \|\nabla I(v_i)\|^2\ + \alpha\left(\|v_{i+1} - v_i\|^2 - \bar{d}\right)^2\right)$$

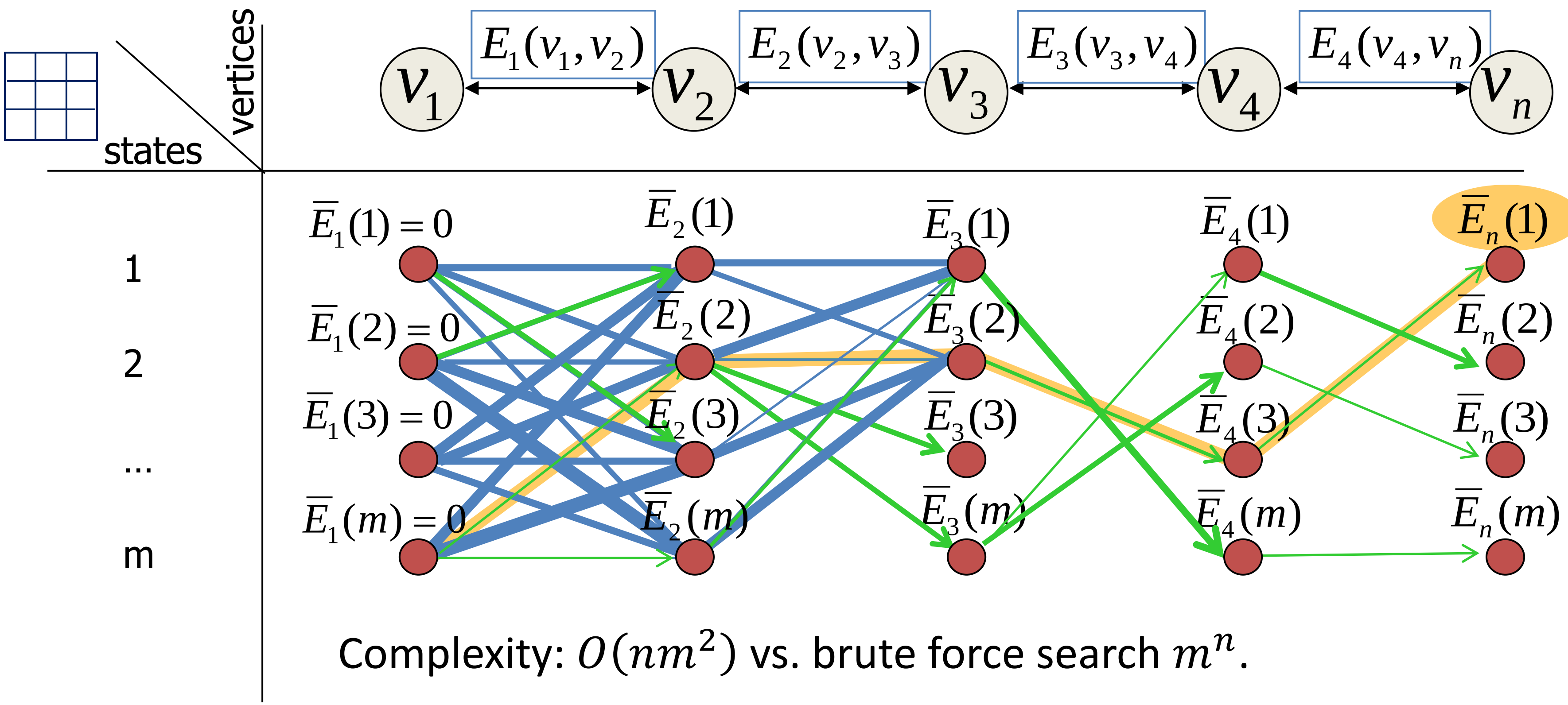
$$E_{total}(v_1, \ldots, v_n) = \sum_{i=1}^{n} E_i(v_i, v_{i+1})$$

$$E_{total}(v_1, \ldots, v_n) = E_1(v_1, v_2) + E_2(v_2, v_3) + \cdots + E_n(v_n, v_0)$$

Viterbi Algorithm

Main idea: determine optimal position (state) of predecessor, for each possible position of self. Then backtrack from best state for last vertex.

$$E_{total}(v_1, \ldots, v_n) = E_1(v_1, v_2) + E_2(v_2, v_3) + \cdots + E_n(v_n, v_0)$$

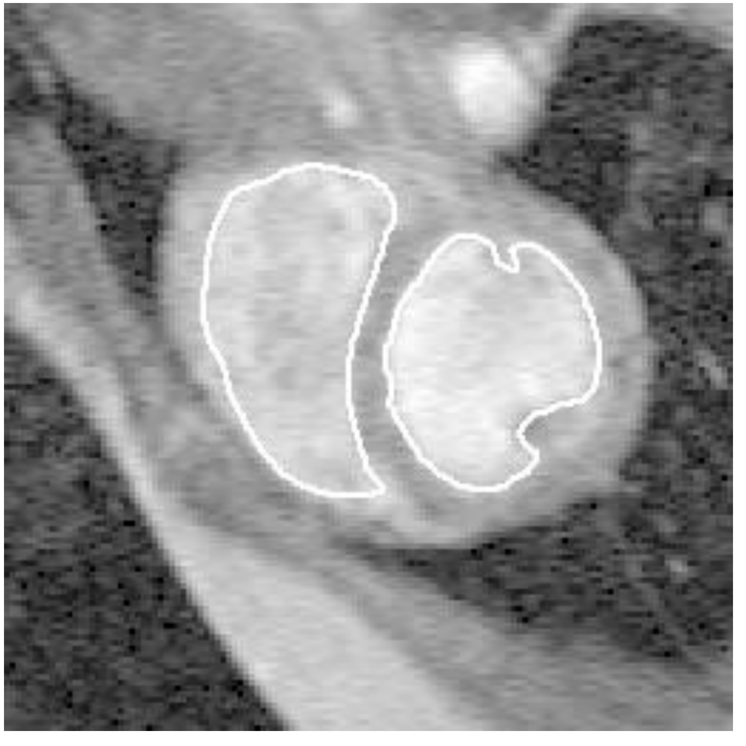


Complexity: $O(nm^2)$ vs. brute force search m^n.

Example adapted from Yuri Boykov

Tracking via Deformable Contours

1. Use final contour/model extracted at frame t as an initial solution for frame $t + 1$
2. Evolve initial contour to fit exact object boundary at frame $t + 1$
3. Repeat, initializing with most recent frame.



Tracking Heart Ventricles
(multiple frames)

Slide: Kristen Grauman

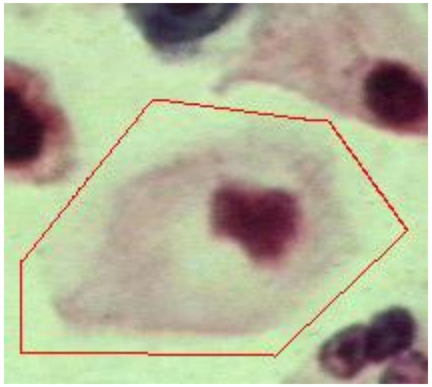
Tracking via Deformable Contours

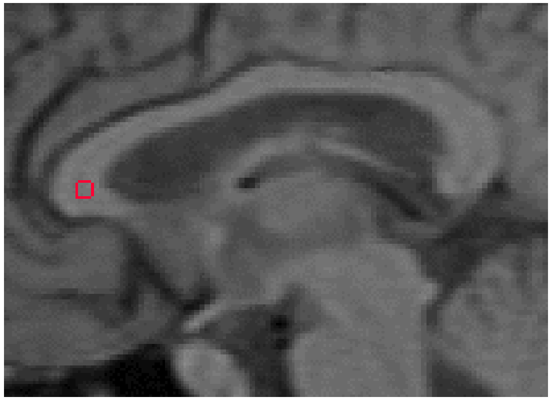
Visual Dynamics Group, Dept. Engineering Science, University of Oxford.

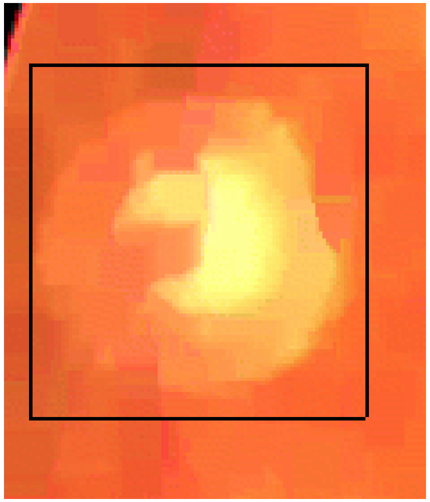
Applications: Traffic monitoring
Human-computer interaction
Animation
Surveillance
Computer assisted diagnosis in medical imaging

Slide: Kristen Grauman

Snakes







Images taken from http://www.cs.bris.ac.uk/home/xie/content.htm

Deformable Contours: Pros and Cons

<u>Pros:</u>

- Useful to track and fit non-rigid shapes
- Contour remains connected
- Possible to fill in “subjective” contours
- Flexibility in how energy function is defined, weighted.

<u>Cons</u>:

- Must have decent initialization near true boundary, may get stuck in local minimum
- Parameters of energy function must be set well based on prior information

Slide: Kristen Grauman

References

- Active Contours
  Szeliski, section 5.1

- Snakes: Active contour models
  Kass, Witkin, & Terzopoulos
  ICCV1987