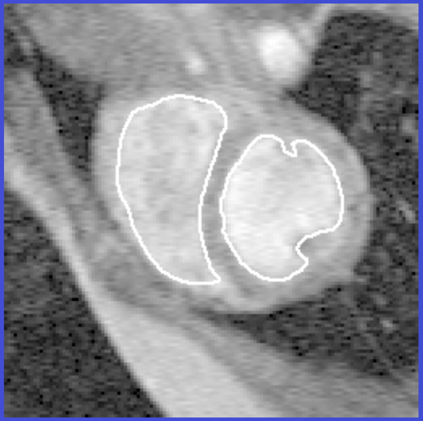
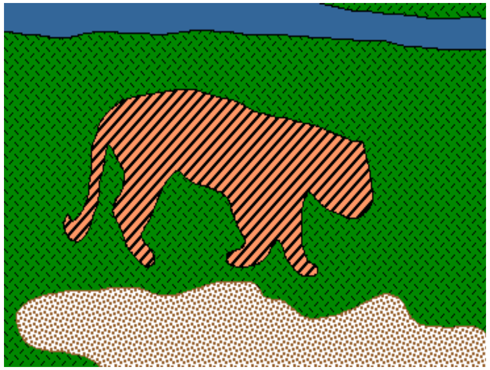


Deformable Contour / Shape Matching

Slide sources: original from Kristen Grauman;
updated by Vittorio Ferrari, Gabor Szekely, Orcun Goksel

Computer Vision

Pixel grouping problem



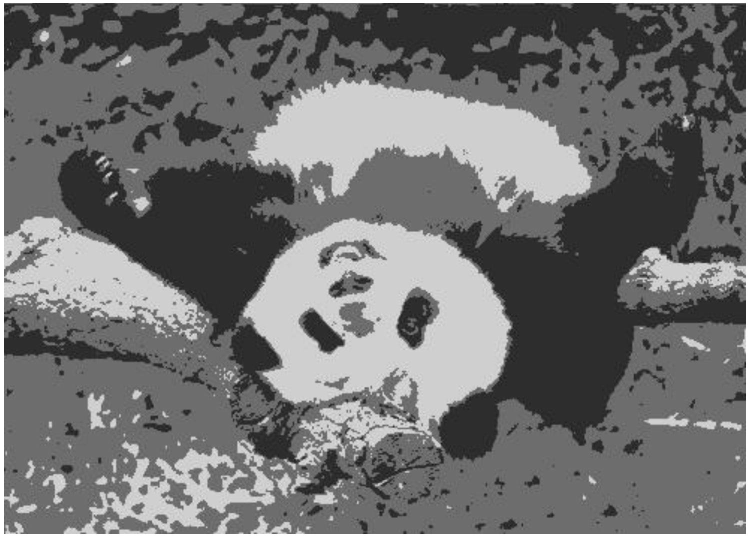
Goal: move from set of pixel values to spatial configuration of regions, objects, and shapes

<u>Similarly to:</u> region growing, watershed

<u>But differently:</u> by considering boundary, curvature, shape, etc, information

Pixels vs. regions

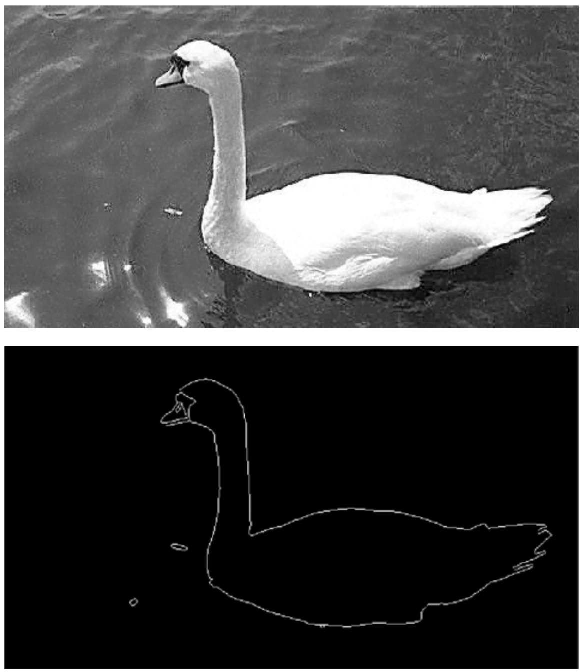
image



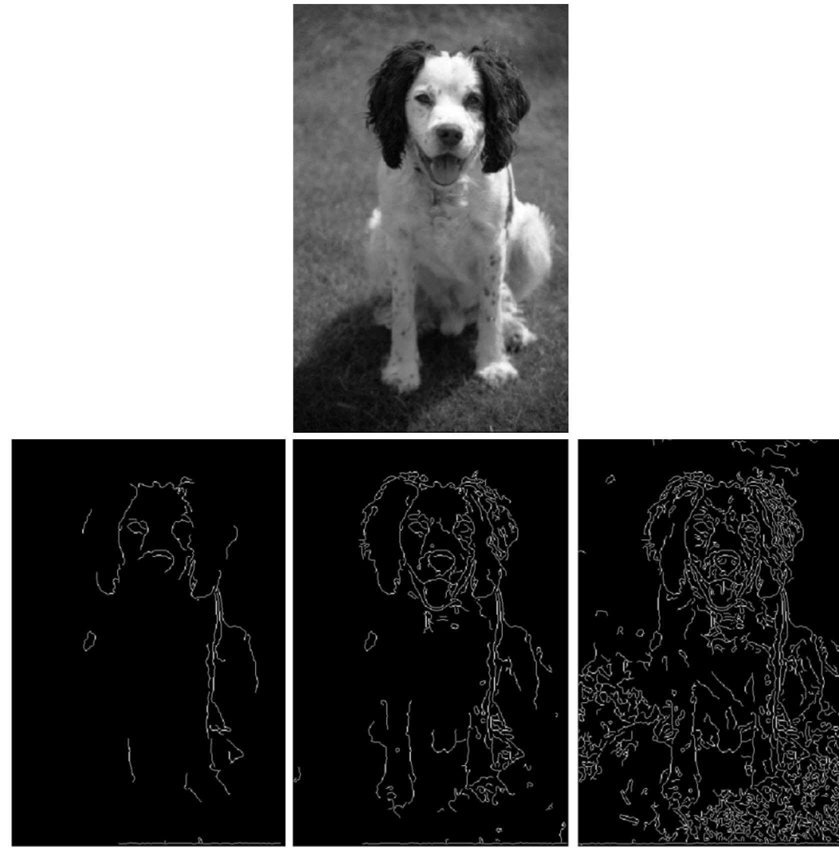
clusters of intensity

Edges vs. boundaries

Edges are useful to infer shape and occlusion



Here the raw edge output is not so bad



But, quite often the boundaries of interest are fragmented, and we have “clutter” edges

Images from D. Jacobs

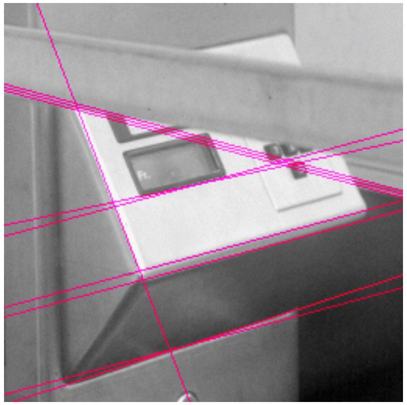
Computer Vision

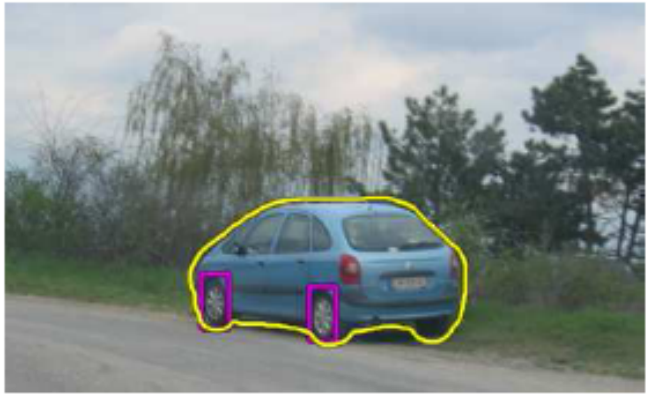
Edges vs. boundaries

Potential solution to missing edges

Given a model of target / object of interest, we can handle some missing and noisy edges using **fitting** techniques.

e.g. with voting methods like the **Hough transform**, detected points vote on possible model parameters.

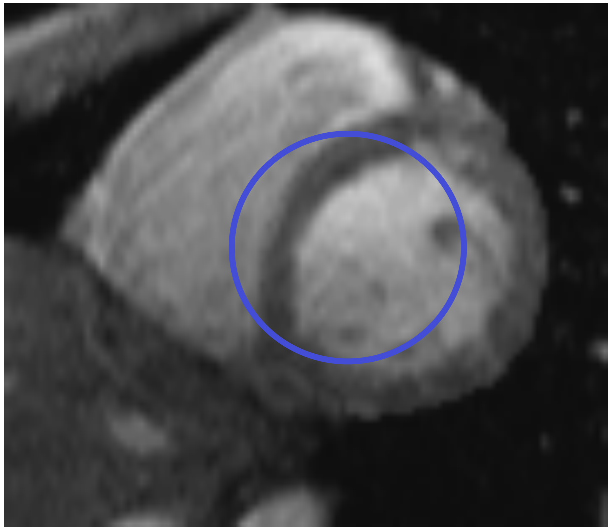




Computer Vision

Active contour models: Snakes

Given: initial contour (model) near desired object



(Single frame)

Fig: Y. Boykov

Snakes: Example

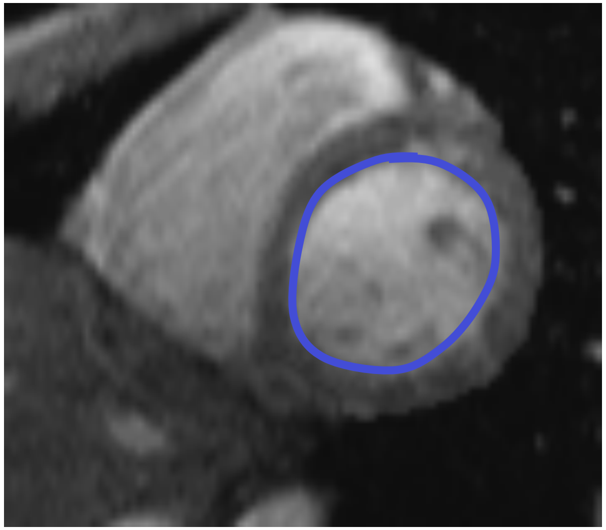
Slide sources: original from Kristen Grauman

Computer Vision

Snakes

Given: initial open or closed curve (model) near the desired structure

Goal: evolve the contour to fit exact object boundary



(Single frame)

Fig: Y. Boykov

Snakes: Example

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

Snakes

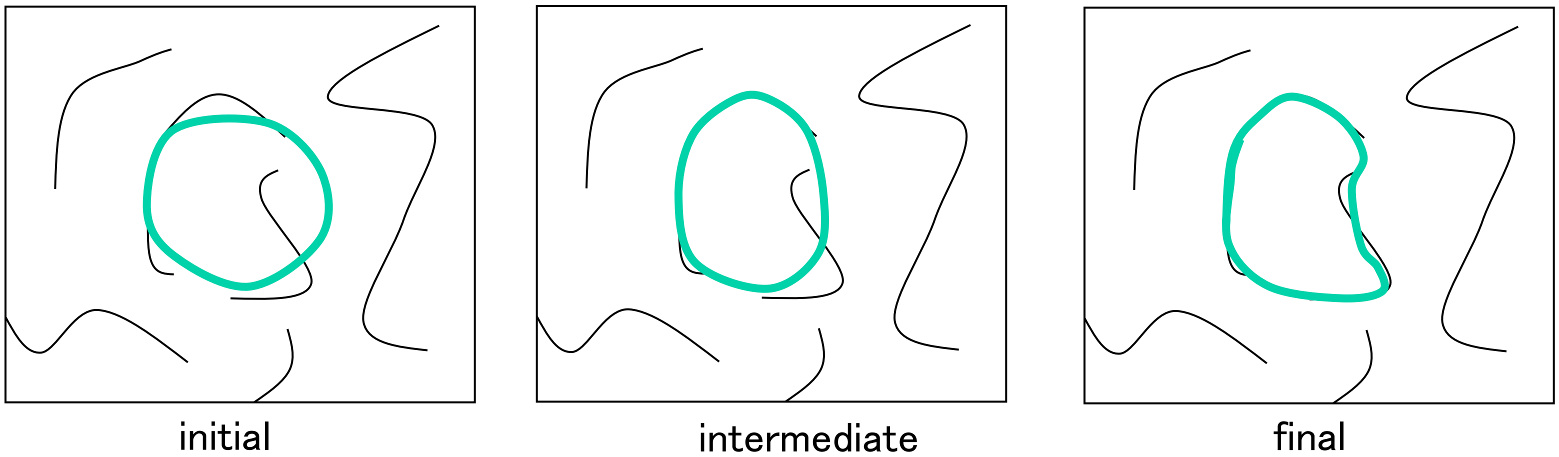


Fig: Y. Boykov

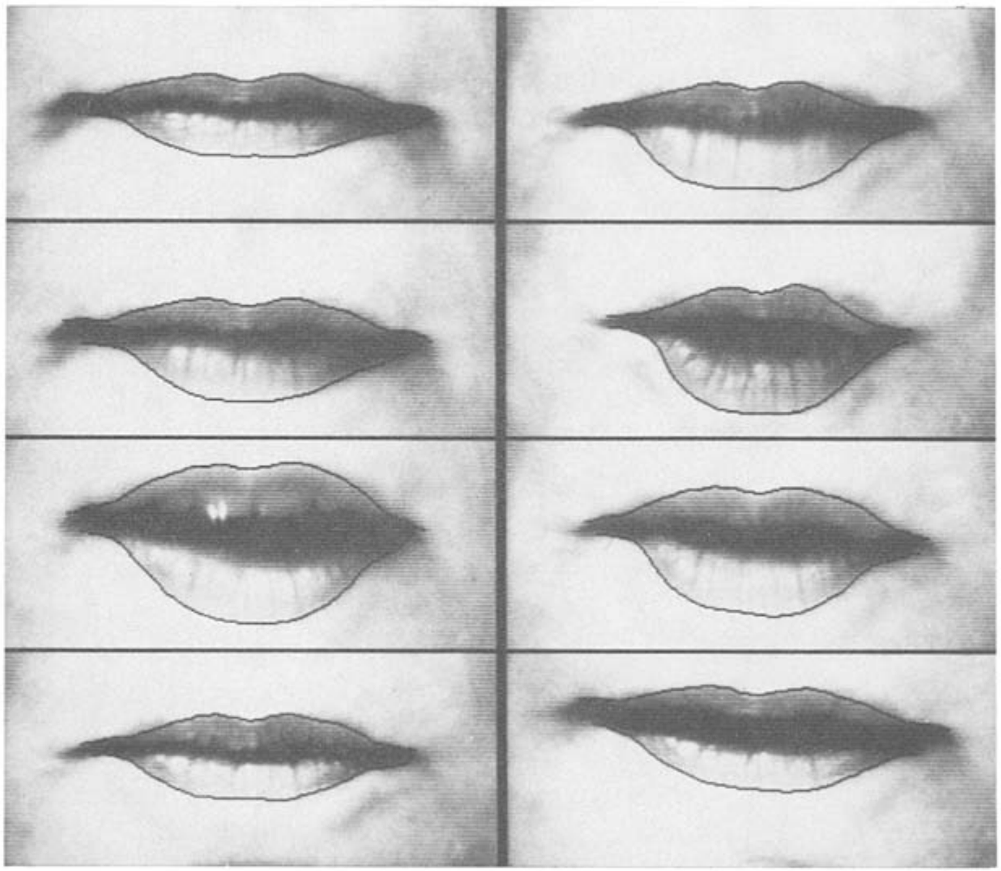
Initialize near contour of interest

Iteratively refine: elastic band is adjusted so as to

- be near image positions with high gradients, and
- satisfy shape “preferences” or contour priors
- requires initialization near object of interest
- one optimization “pass” to fit a single contour

Computer Vision

Why to fit deformable shapes?



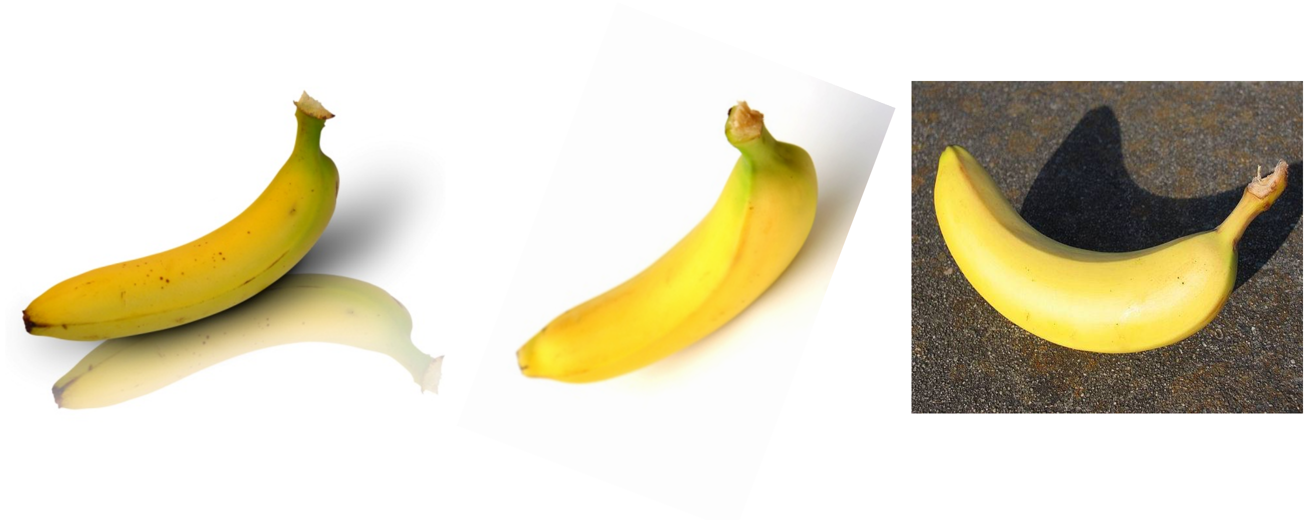
Deformable objects can change their shape over time, e.g. lips, hands.

Figure from Kass et al. 1987

Snakes: Motivation

Computer Vision

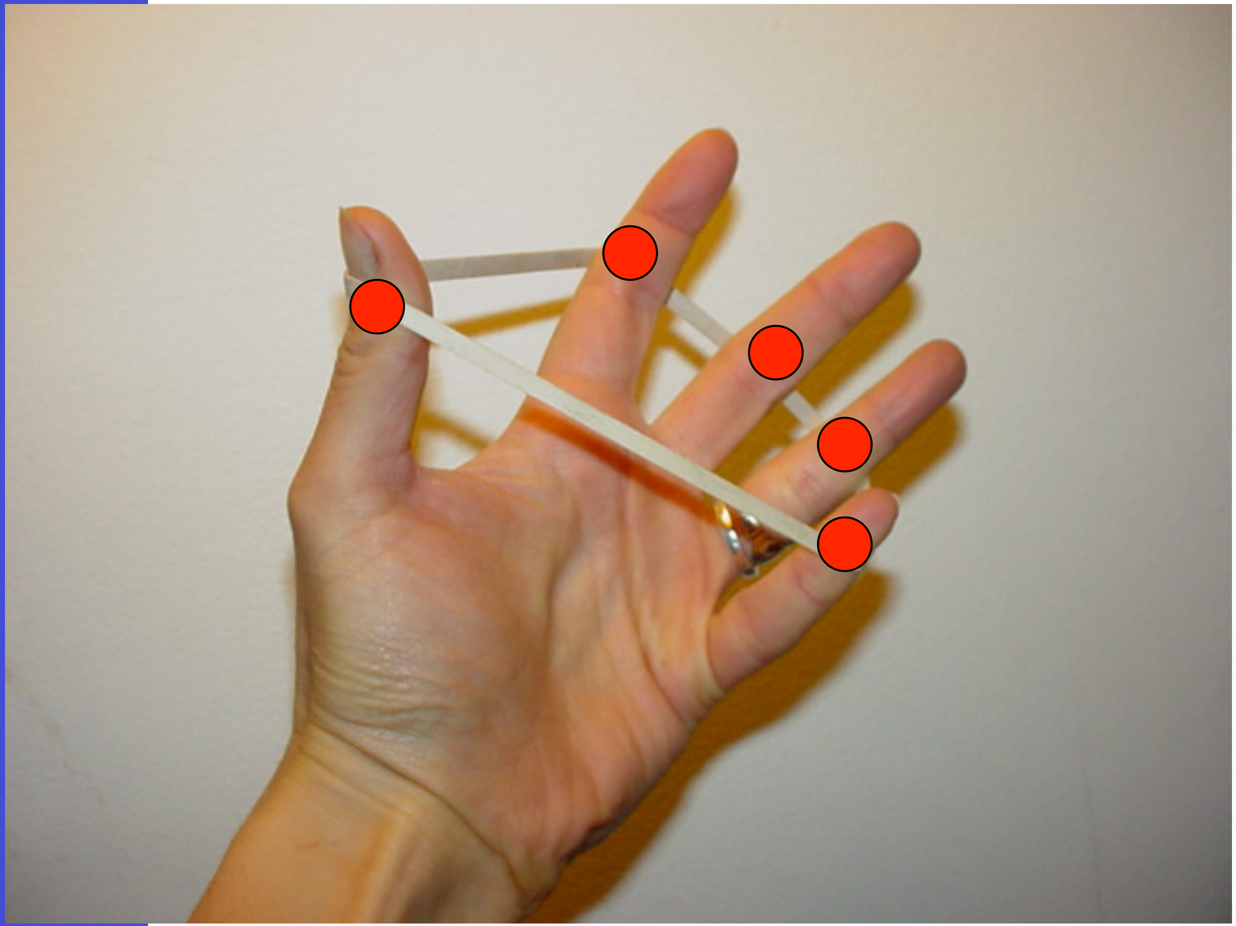
Why to fit deformable shapes?

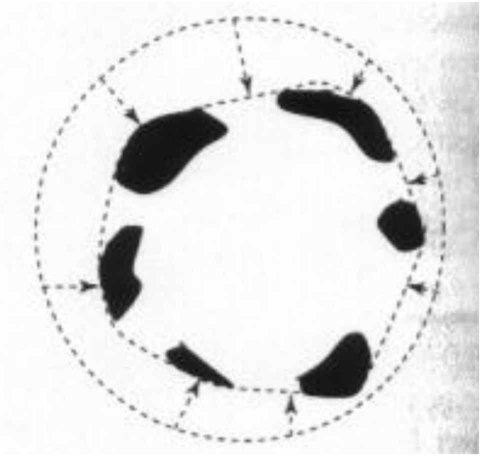


Some objects have similar basic form but some variety in the contour shape.

Snakes: Motivation

Deformable contours: intuition





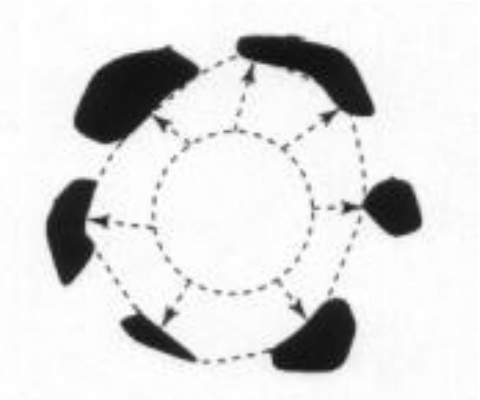


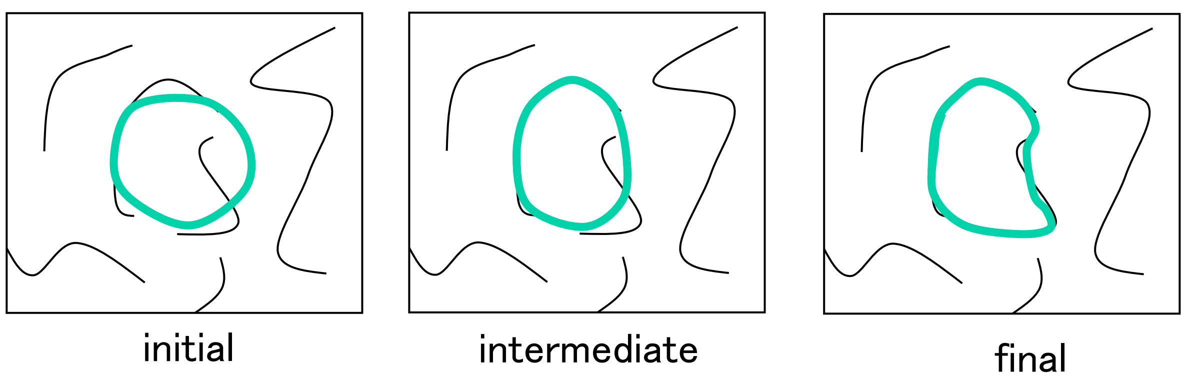
Image from http://www.healthline.com/blogs/exercise_fitness/uploaded_images/HandBand2-795868.JPG

Figures from Shapiro & Stockman

Contour adjustment

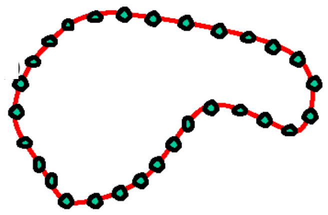
How is the current contour adjusted to find the new contour at each iteration?

- Define energy function that says how good a configuration is
- Seek next configuration that minimizes energy



Snakes energy function

The total energy of the current snake is:



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

Internal energy: encourage prior shape preferences: e.g. smoothness, elasticity, known shape prior

External energy (image energy): encourage contour to fit interesting image structures, e.g. edges

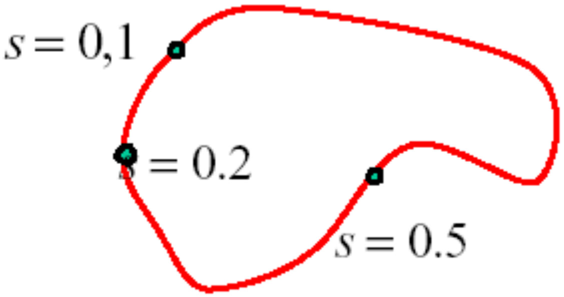
A **good** fit between the current snake and the target shape in the image will yield a **low** energy value.

Computer Vision

Parametric curve representation

Continuous case

$$v(s) = (x(s), y(s)) \qquad 0 \le s \le 1$$



Snakes: Parametrize

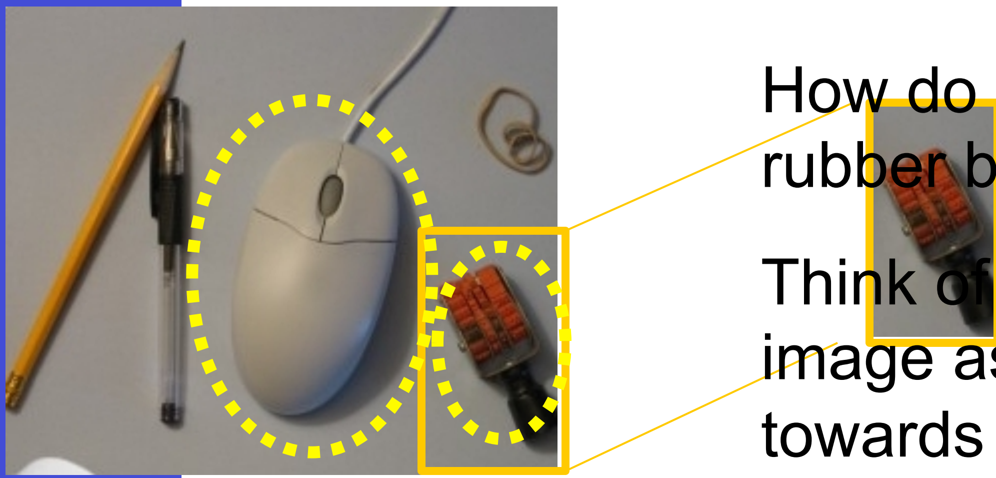
Fig from Y. Boykov

External (image) energy: intuition

- Measure how well the curve matches the image data
- Attract the curve toward interesting image features
  - Edges, lines, etc.

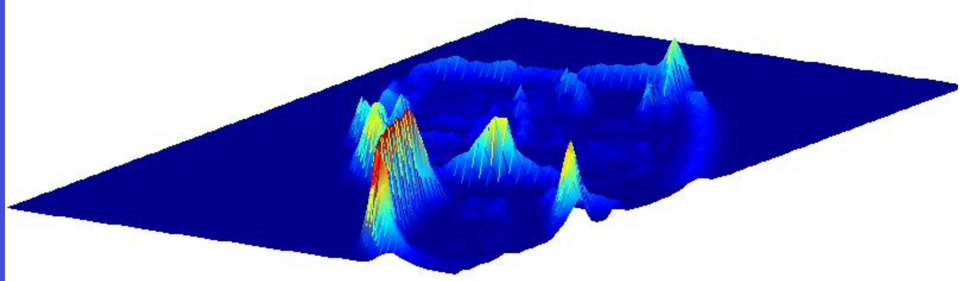
Computer Vision

External (image) energy



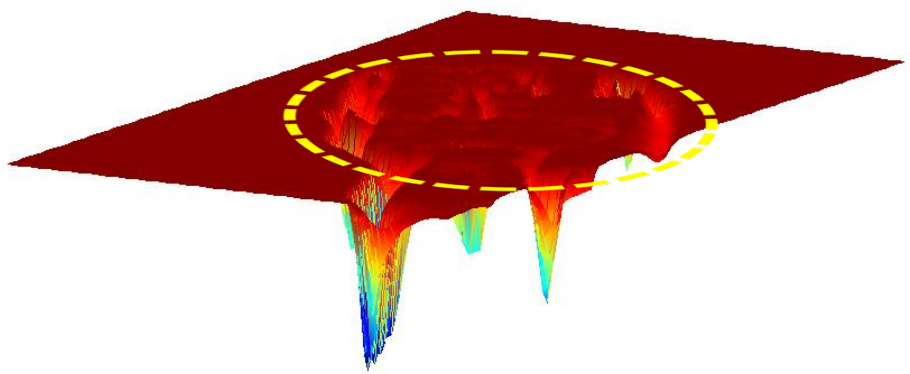
How do edges affect "snap" of rubber band?

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)

$$-\left(G_x(I)^2 + G_y(I)^2\right)$$

Snakes: Energy (image)

Computer Vision

External (image) energy

- Image *I(x,y)*
- Directional derivatives

$$G_x(x,y) \qquad G_y(x,y)$$

- External energy at a point *v(s)* on the curve is

$$E_{external}(v(s)) = -(\,|G_x(v(s))|^2 + |G_y(v(s))|^2\,)$$

- External energy for the whole curve:

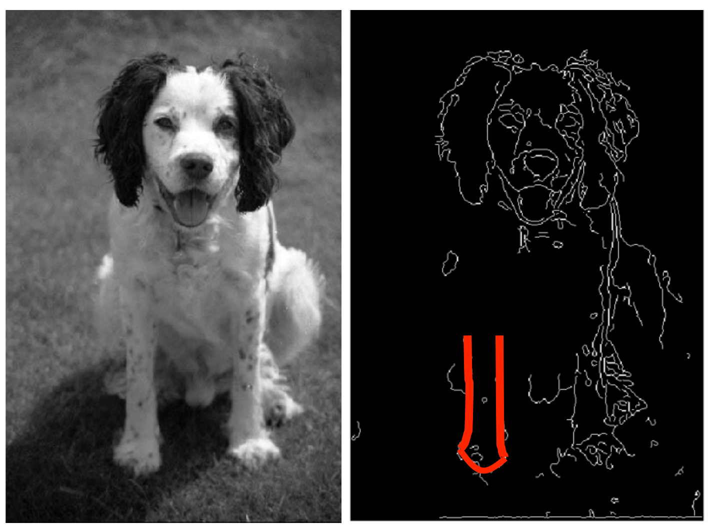
$$E_{external} = \int_0^1 E_{external}(v(s))\,ds$$

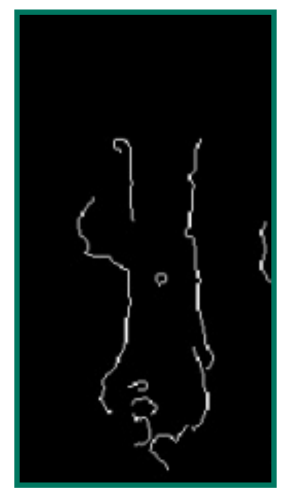
Snakes: Energy (image)

Computer Vision

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 in the gradient image.



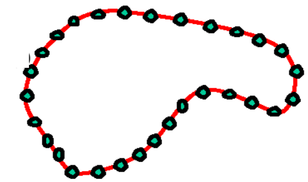


Snakes: Energy (elastic)

Computer Vision

Snakes: Energy (elastic)

Internal energy



For a continuous curve, a common internal energy term is the "deformation energy".

At some point *v(s)* on the curve, this is:

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

Elasticity, Tension
→ inhibit stretch

Stiffness, Curvature
→ inhibit bend

The more the curve stretches and bends→ the larger this energy value is.

The weights α and β dictate how much influence each component has.

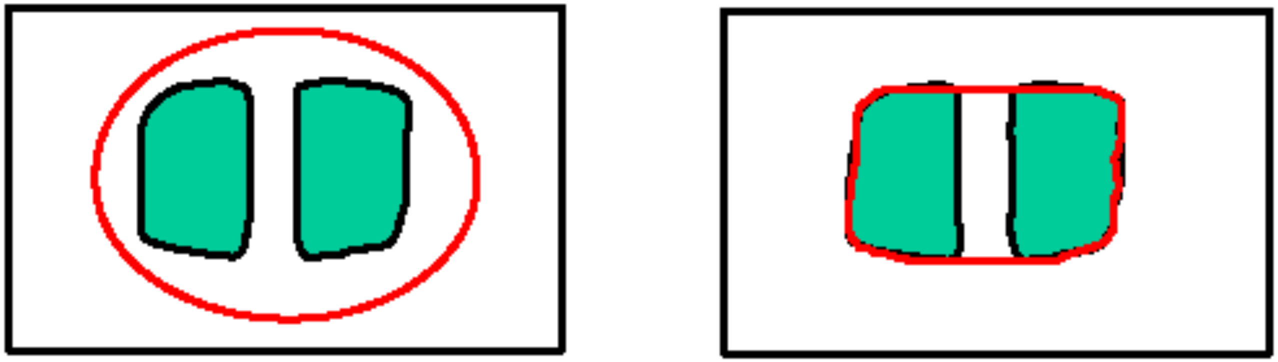
Internal energy for whole curve:

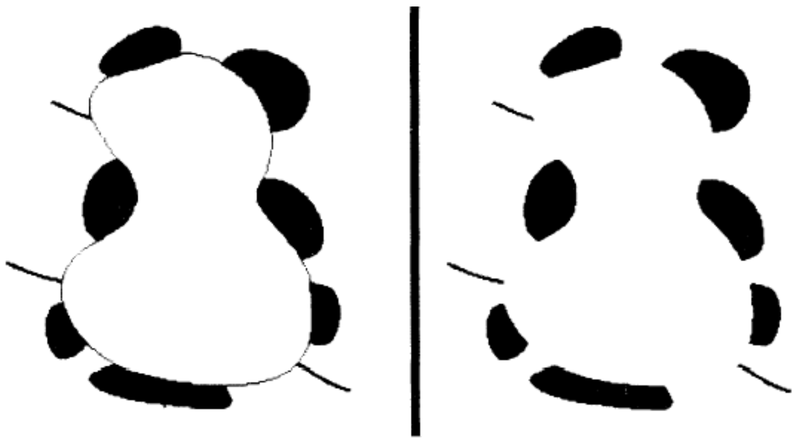
$$E_{internal} = \int_0^1 E_{internal}(v(s))\,ds$$

Computer Vision

Dealing with missing data

The smoothness constraints imposed by the internal energy helps dealing with missing data:





[Figure from Kass et al. 1987]

Snakes: Energy (elastic)

Total energy

continuous form

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

$$E_{internal} = \int_0^1 E_{\text{internal}}(\nu(s))\,ds$$

deformation energy

$$E_{external} = \int_0^1 E_{external}(\nu(s))\,ds$$

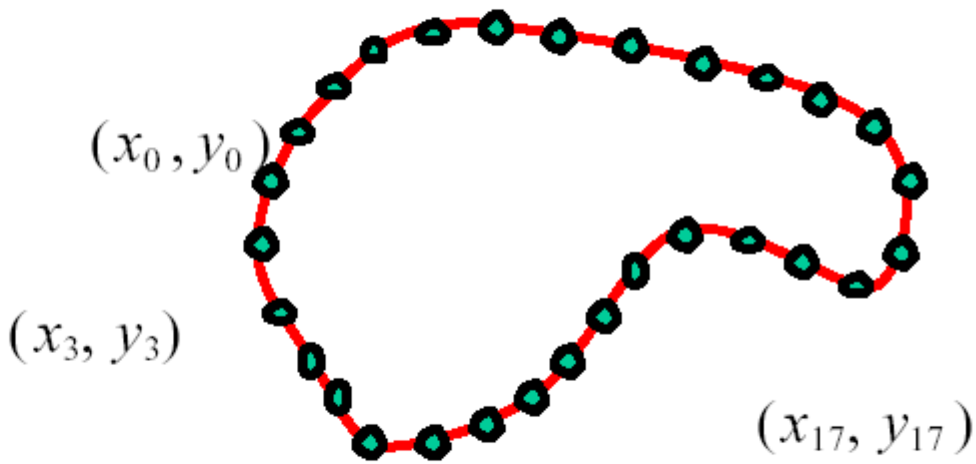
total edge strength under curve

Computer Vision

Parametric curve representation

- Curve discretization
- Represent the curve with a set of *n* points

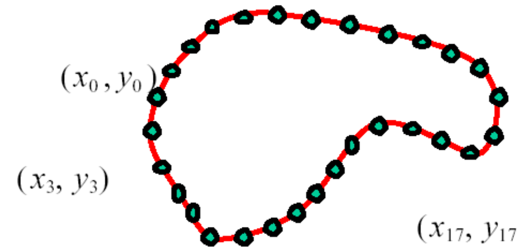
$$v_i = (x_i, y_i) \qquad i = 0 \ldots n-1$$



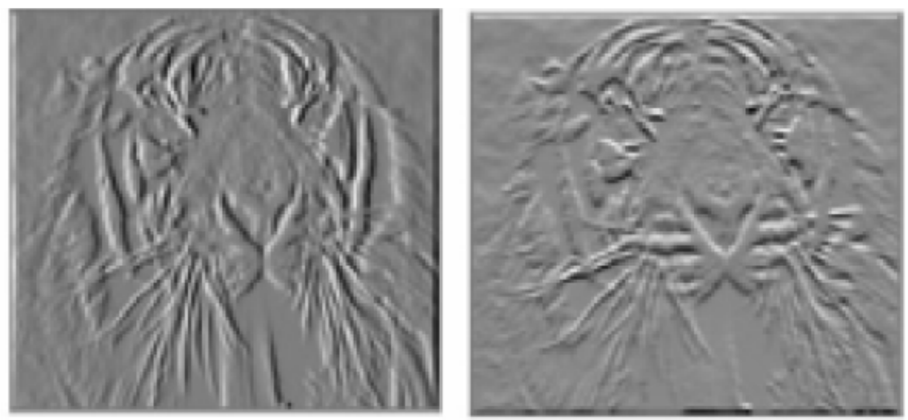
Computer Vision

Discrete energy function: external term

The curve is represented by *n* points



$$E_{external} = -\sum_{i=0}^{n-1} |G_x(x_i, y_i)|^2 + |G_y(x_i, y_i)|^2$$



$G_x(x, y)$ $G_y(x, y)$

Discrete image derivatives

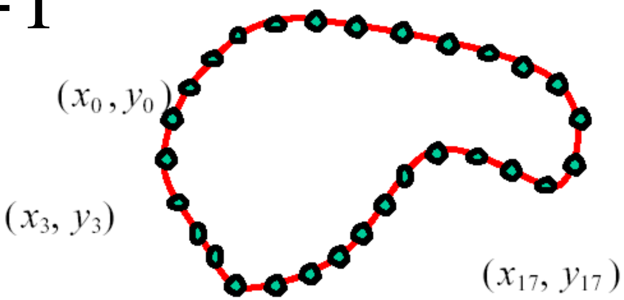
Snakes: Discretize → energy (image)

Computer Vision

Discrete energy function: internal term

The curve is represented by *n* points

$$v_i = (x_i, y_i) \qquad i = 0 \ldots n-1$$



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

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

$$E_{internal} = \sum_{i=0}^{n-1} \alpha \left\| v_{i+1} - v_i \right\|^2 + \beta \left\| v_{i+1} - 2v_i + v_{i-1} \right\|^2$$

Elasticity, Tension

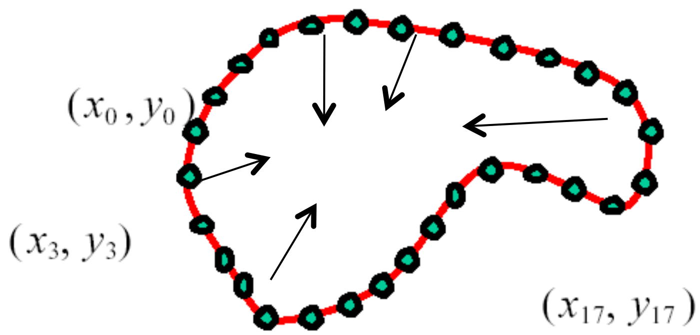
Stiffness, Curvature

Snakes: Discretize → energy (elastic)

Penalizing extension (elasticity)

$$\sum_{i=0}^{n-1} \boxed{\alpha \left\| v_{i+1} - v_i \right\|^2} + \beta \left\| v_{i+1} - 2v_i + v_{i-1} \right\|^2$$

Prefers shorter curves



Problem with this definition: encourages a closed curve to shrink to a cluster of coincident points.

Possible remedy: adjusting energy term

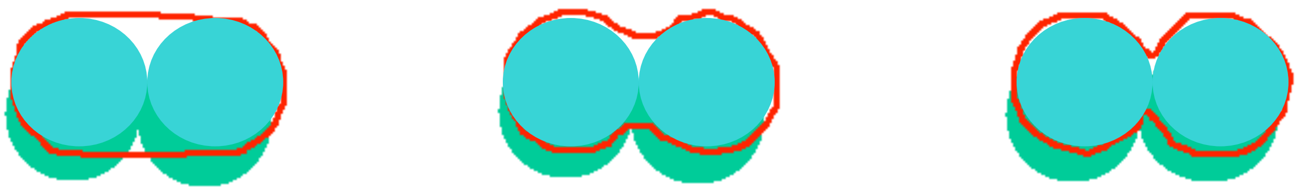
$$E_{internal} = \sum_{i=0}^{n-1} \alpha \left(\left\| v_{i+1} - v_i \right\| - 1 \right)^2 + \beta \left\| v_{i+1} - 2v_i + v_{i-1} \right\|^2$$

Encourages equal spacing but makes optimization harder

Function of the weights

$$\sum_{i=0}^{n-1} \alpha \left\| v_{i+1} - v_i \right\|^2 + \beta \left\| v_{i+1} - 2v_i + v_{i-1} \right\|^2$$

α, β weights control the penalty for deformation (α for stretching and β for bending)



large medium small

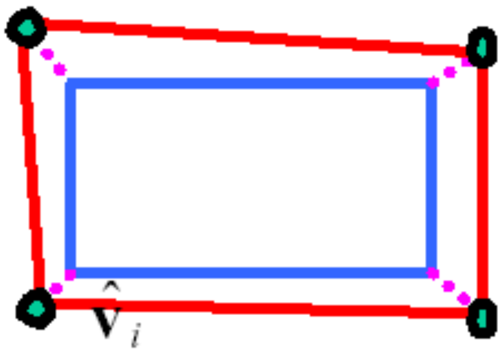
Fig from Y. Boykov

Computer Vision

Optional: specify shape prior

If object is some smooth variation on a known shape, we can add a term to penalize deviation from that shape:

$$\delta \sum_{i=0}^{n-1} (v_i - \hat{v}_i)^2$$



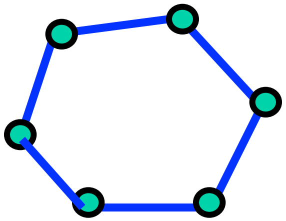
where $\{\hat{v}_i\}$ are the points of the known shape

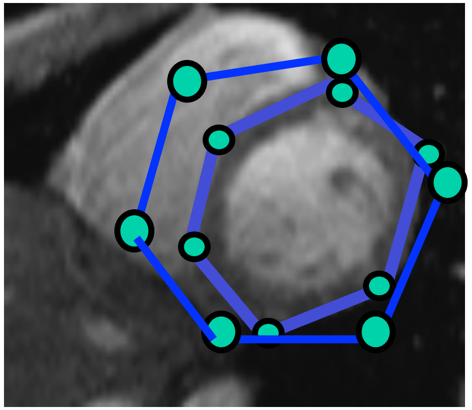
Snakes: Shape Prior

Fig from Y. Boykov

Summary

- A simple elastic snake is defined by
  - A set of *n* points,
  - An internal deformation energy term
  - An external edge-based energy term
- Use to locate the outline of an object
  - Initialize in the vicinity of the object
  - Modify the points to minimize the total energy





No generic way for the selection of the weights

Energy minimization

Several methods proposed to fit snakes, including methods based on :

- Partial Differential Equations (PDEs)
- Greedy search
- Dynamic programming (for 2d snakes)

Energy minimization through PDEs

Energy to minimize (continuous case)

$$\frac{1}{2}\int_0^1 \left(\alpha(s)\left|\left|\frac{\partial v}{\partial s}\right|\right|^2 + \beta(s)\left|\left|\frac{\partial^2 v}{\partial s^2}\right|\right|^2\right) ds - \int_0^1 P(v)ds$$

deformation energy

image energy (gradients)

Variational calculus → Euler-Lagrange differential equation

$$-\frac{\partial}{\partial s}\left(\alpha(s)\frac{\partial v}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(\beta(s)\frac{\partial^2 v}{\partial s^2}\right) = -\vec{\nabla} P(v)$$

Energy minimization through PDEs

with weights uniform over the snake,
written in 2 axes:

$$-\alpha \frac{\partial^2 x}{\partial s^2} + \beta \frac{\partial^4 x}{\partial s^4} = -\frac{\partial P}{\partial x}$$

$$-\alpha \frac{\partial^2 y}{\partial s^2} + \beta \frac{\partial^4 y}{\partial s^4} = -\frac{\partial P}{\partial y}$$

in short notation

$$-\alpha v_{ss} + \beta v_{ssss} = -P_v$$

find solution in iterative fashion
(using image gradients at position in previous iteration *t-1*)

$$-\alpha v^t_{ss} + \beta v^t_{ssss} = - P_v \Big|_{v=v^{t-1}}$$

Energy minimization through PDEs

Implementation: discretize curve into *n* points v_i

use a finite difference approximation of derivatives

$$
\begin{aligned}
v_{ss}^t(s_i) &\approx v_{i-1}^t - 2v_i^t + v_{i+1}^t \\
v_{ssss}^t(s_i) &\approx v_{i-2}^t - 4v_{i-1}^t + 6v_i^t - 4v_{i+1}^t + v_{i+2}^t
\end{aligned}
$$

Substituting in the iteration scheme → yields system of linear equations

$$-\alpha v_{ss}^t + \beta v_{ssss}^t = - P_v \Big|_{v=v^{t-1}}$$

Stiffness matrix **K**:

- penta-diagonal
- made of coeffs α,β

$$\mathbf{K} v^t = - P_v \Big|_{v=v^{t-1}}$$

Solve as a linear system (backward-substitution or by inverting **K**)

$$v^t = \mathbf{K}^{-1} P_v \Big|_{v=v^{t-1}}$$

Computer Vision

Inversion of the stiffness matrix

$$v^t = \mathbf{K}^{-1} \, P_v \Big|_{v=v^{t-1}}$$

Problem: **K** is singular

Unique solution needs boundary conditions

Closed contour: “The end is the beginning” (periodicity)
So, boundary conditions could be given implicitly:

$$v(0) = v(1) \ , \quad v'(0) = v'(1) \ , \quad v''(0) = v''(1) \ , \quad v'''(0) = v'''(1)$$

In discrete form, out-of-bounds [0,n] values should wrap around:

$$v_n \Rightarrow v_0 \ , \quad v_{n+1} \Rightarrow v_1 \ , \quad v_{-1} \Rightarrow v_{n-1} \ , \quad v_{-2} \Rightarrow v_{n-2}$$

Remark: For open contours, sufficient number of points must be fixed

Energy minimization through PDEs

- Extended model
- Idea: better physical model by taking temporal development of snake into account (it moves!)
- Add kinetic energy to total energy function

$$E_K(v) = \frac{1}{2}\int_0^1 \mu(s)\left(\frac{\partial v(s,t)}{\partial t}\right)^2 ds$$

mass of snake — temporal derivative

Total energy

$$E(v) = E_K(v) + E_I(v) + E_D(v)$$

Computer Vision

Energy minimization through PDEs

Further improvement: snake dissipates energy through friction

$$D(v_t) = \frac{1}{2}\int_0^1 \gamma(s)\, |v_t|^2 \, ds$$

damping coefficient

- *t* now has a the physical meaning of *time*, as snake moves in image
- Find where snake moves, subject to all forces, by min total energy

$$\int_0^1 (E(v) + D(v_t))\, ds$$

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

Spatio-temporal model

with weights uniform over the snake, in two axes:

$$\mu\frac{\partial^2 x}{\partial t^2}+\gamma\frac{\partial x}{\partial t}-\alpha\frac{\partial^2 x}{\partial s^2}+\beta\frac{\partial^4 x}{\partial s^4} = -\frac{\partial P}{\partial x}$$
$$\mu\frac{\partial^2 y}{\partial t^2}+\gamma\frac{\partial y}{\partial t}-\alpha\frac{\partial^2 y}{\partial s^2}+\beta\frac{\partial^4 y}{\partial s^4} = -\frac{\partial P}{\partial y}$$

discretize to points, with finite difference derivatives also for time:

$$v_t \approx v^t - v^{t-1}$$
$$v_{tt} \approx v^t - 2v^{t-1}v^{t-2}$$

Again a linear system of equations:

$$(\mu+\gamma)v_i^t-\alpha(v_{i-1}^t-2v_i^t+v_{i+1}^t)+\beta(v_{i-2}^t-4v_{i-1}^t+6v_i^t-4v_{i+1}^t+v_{i+2}^t)=-P_v\Big|_{v_i^{t-1}}+(2\mu+\gamma)v_i^{t-1}-\mu v_i^{t-2}$$

Spatiotemporal model

In matrix form:

$$[(\mu+\gamma)\mathbf{I}+\mathbf{K}]\,v^t = -\,P_v\Big|_{v^{t-1}} + (2\mu+\gamma)v^{t-1} - \mu v^{t-2}$$

Role of damping → better conditioning of (extended) **K**
(dampen sudden responses)

Role of mass → adds "memory" to the evolving curve

Solve by inverting the (extended) **K**

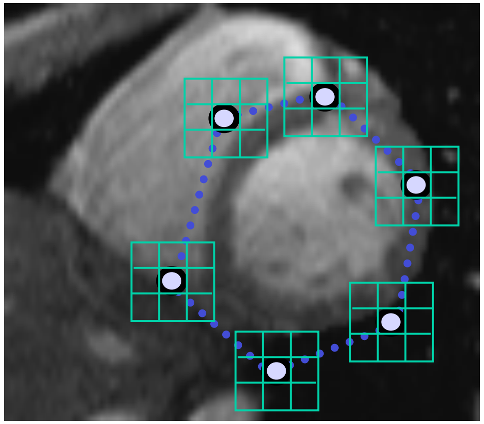
$$v^t = [(\mu+\gamma)\mathbf{I}+\mathbf{K}]^{-1}\left(-\,P_v\Big|_{v^{t-1}} + (2\mu+\gamma)v^{t-1} - \mu v^{t-2}\right)$$

Snakes: Optimization (PDEs)
> extension
solve

Computer Vision

Greedy energy minimization

- 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 the last iteration, or after max number of iterations
- Note
  - Convergence not guaranteed
  - Need decent initialization



Snakes: Optimization (greedy)

Energy minimization: dynamic programming

Often snake energy can be rewritten as a sum of
* unary potentials for individual points and
* interaction potentials between pairs of points

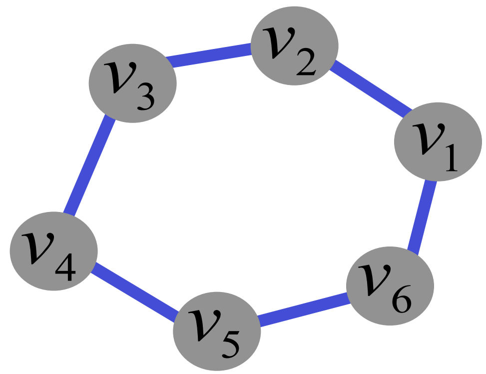
$$E_{total}(v_0,\ldots,v_{n-1}) = \sum_{i=0}^{n-1} U_i(v_i) + \sum_{i=0}^{n-2} P_i(v_i,v_{i+1})$$

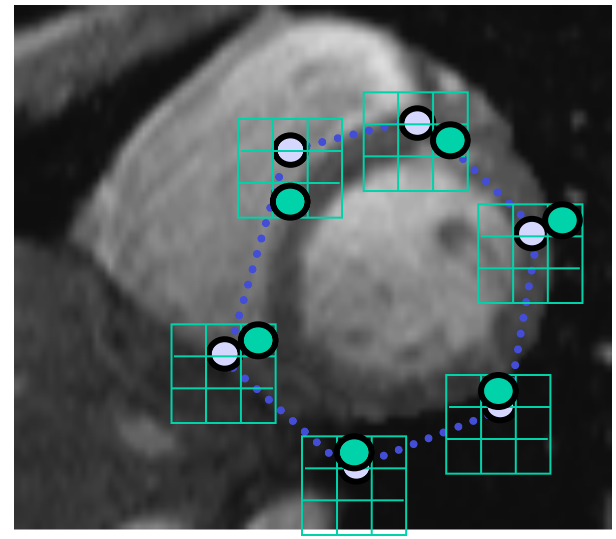
image energy (gradients)

deformation energy (good for tension, but for bending ?)

A point only affects a few terms of this sum !

Dynamic programming: discretize states





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

* Center each box to its optimal position around it, and
* Iterate until the optimal position is at the box center.

i.e., the snake is optimal in a local discrete search space defined by the boxes

Fig from Y. Boykov
[Amini, Weymouth, Jain, 1990]

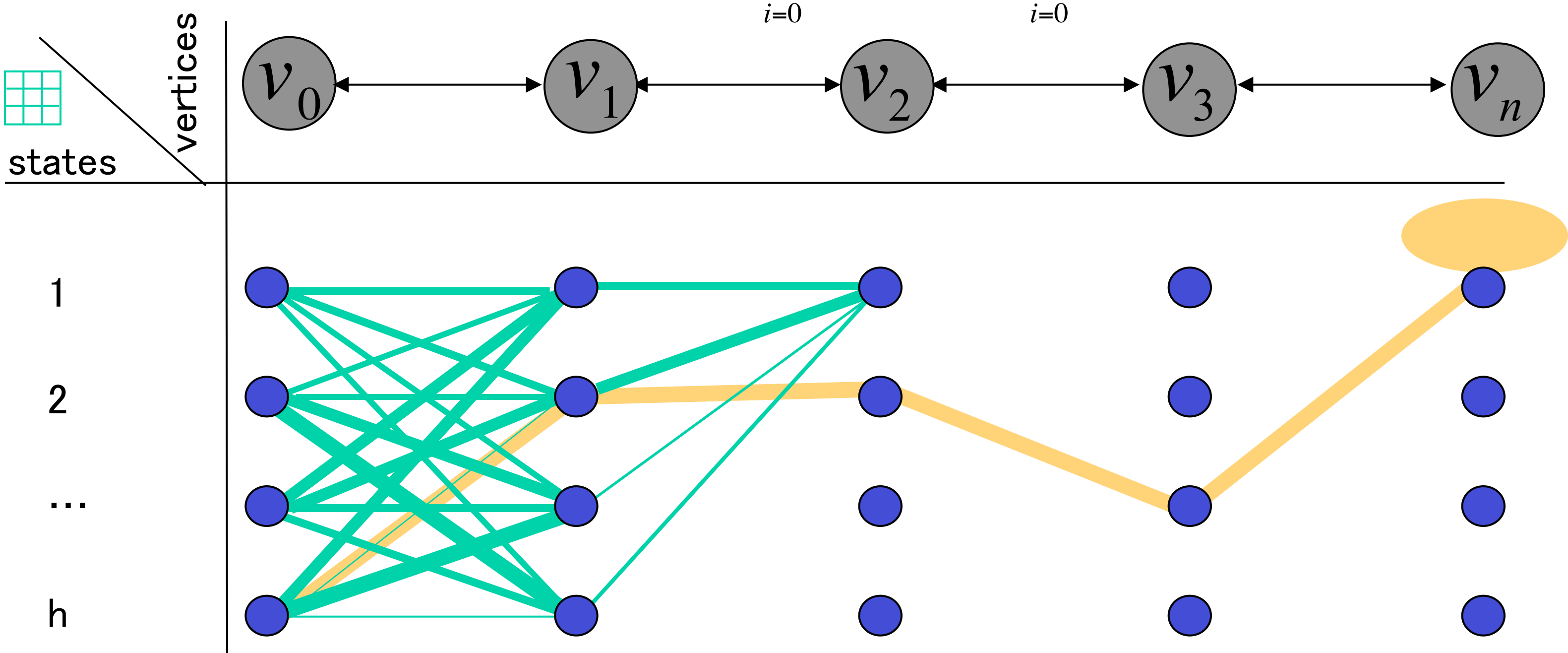
Computer Vision

Dynamic programming: optimal path

1) for each possible position (state) of a vertex, find cost of optimal path arriving there, and optimal state of predecessor.

2) backtrack from best state for last vertex.

$$E_{total}(v_0,\ldots,v_{n-1}) = \sum_{i=0}^{n-1} U_i(v_i) + \sum_{i=0}^{n-2} P_i(v_i,v_{i+1})$$



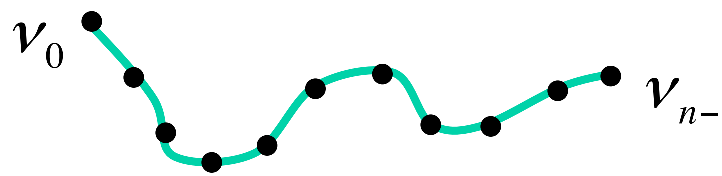
	h			
E of best path to v1 = h				
pointer to best v0				

	h			
E of best path to v2 = h				
pointer to best v1				

Snakes: Optimization (DP)
> optim path

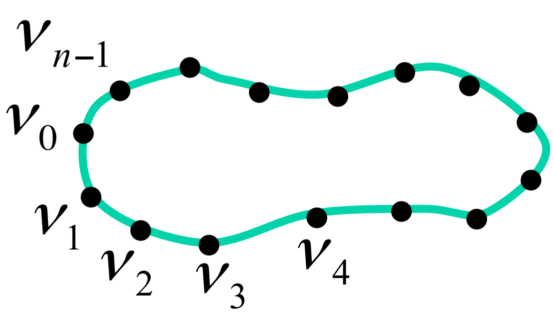
Energy minimization: dynamic programming

DP can easily be applied to optimize an open snake



Complexity: $O(nh^2)$ vs brute force search ____?

Problem: A closed snake contains a "loop" in the total energy, not solvable in one DP pass



Work arounds:

1) Fix v_1 and solve for rest.
2) Fix an intermediate node at its position found in (t-1), and solve for rest.

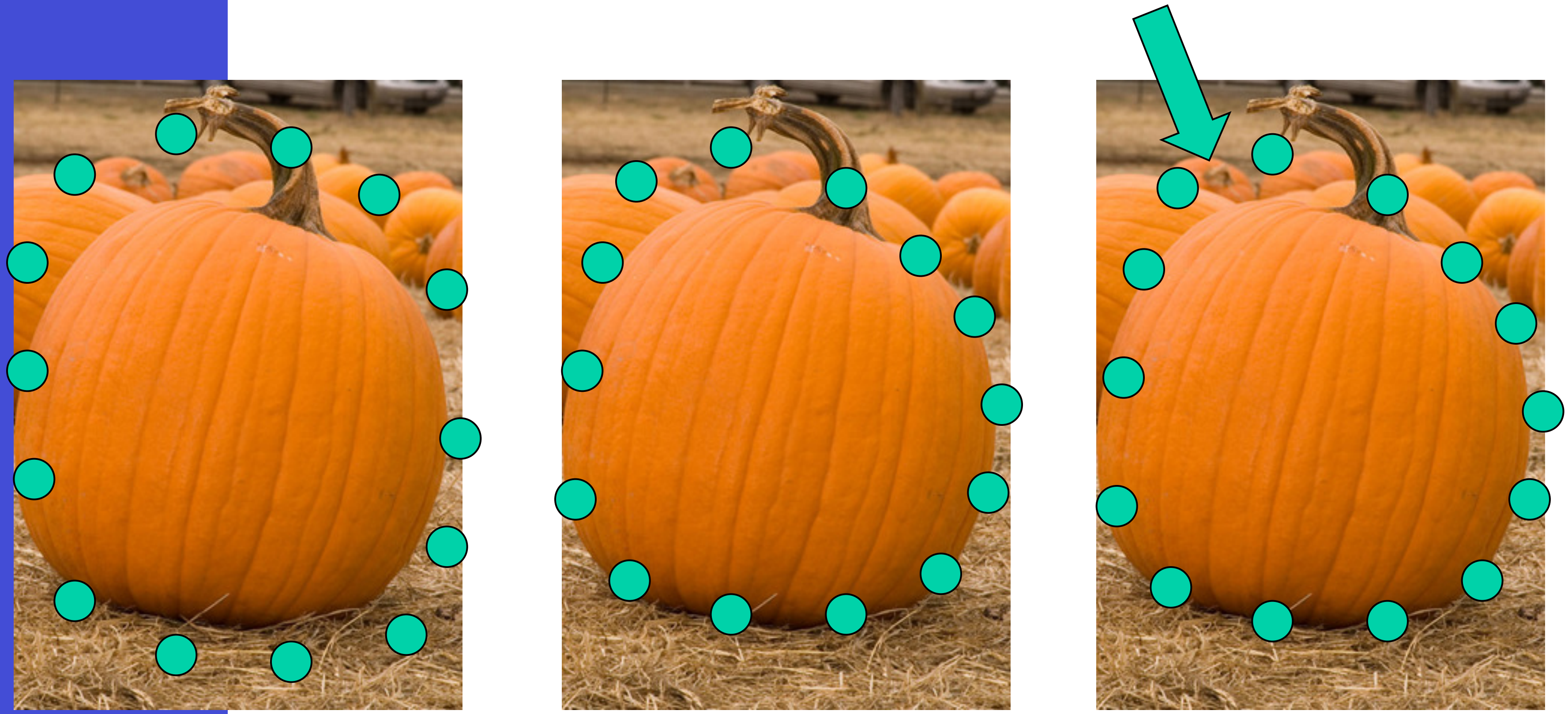
Extension with further external forces

- Pressure (Baloons)
  - Pushing the curve to extend (inside to outside)
  - Constant force magnitude
  - Shows in the direction of the curve´s normal
- Gravity

  constant force in a preferred direction

Computer Vision

Interactive forces



Snakes: Extensions (user input)

Interactive forces

- Energy function altered online based on user input; e.g. push or pull the snake points with the mouse pointer
- This could also be some heuristic force
  e.g. avoiding image borders, utilizing output of another algorithm
- Easy to modify the external energy term to include:

$$E_{push} = +\sum_{i=0}^{n-1} \frac{r^2}{|v_i - p|^2}$$

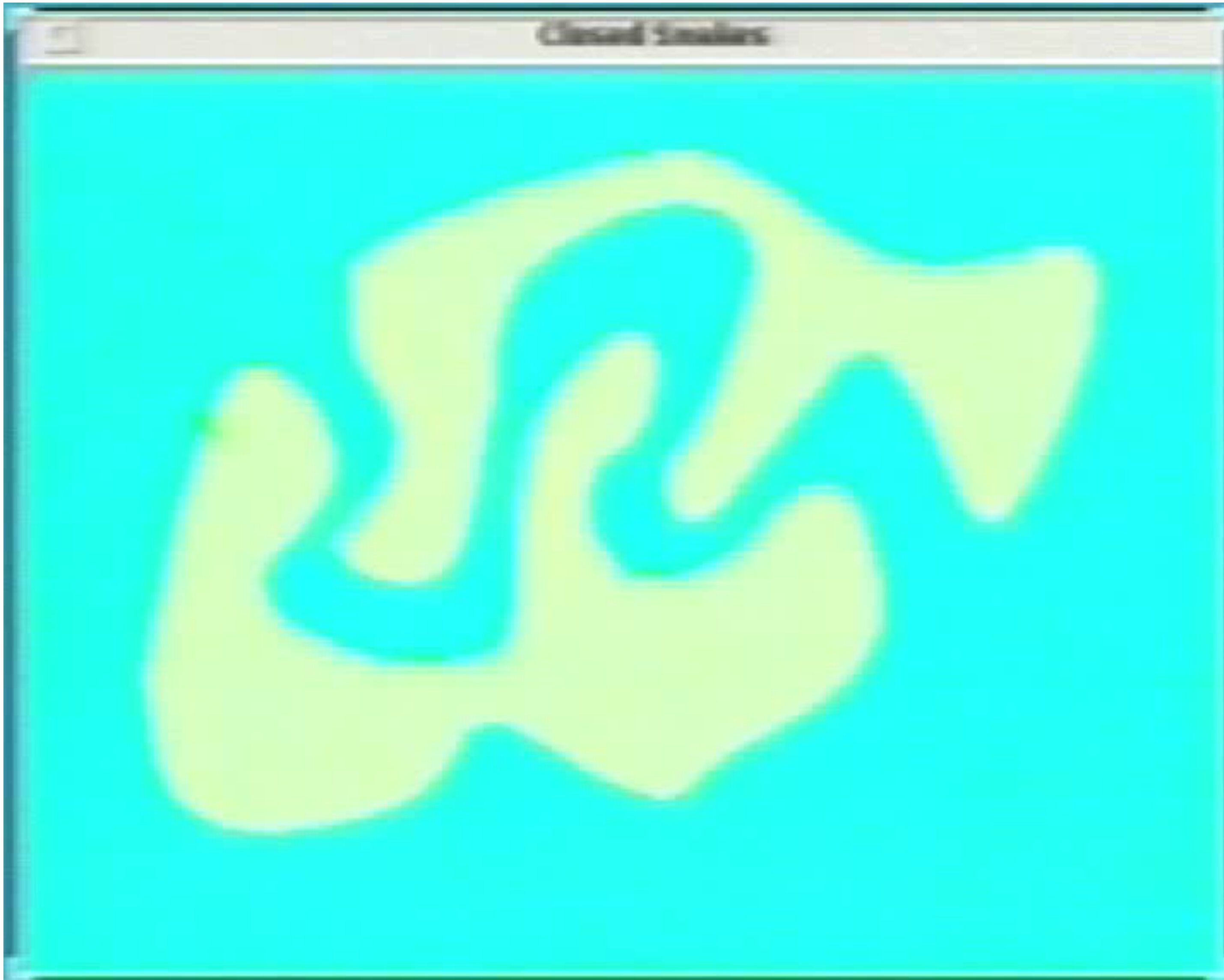
distance of snake points to pointer *p*

Attaches elastic springs to snakes points to push the points away from the pointer with the nearby points pushed hardest

- Similarly for pulling…

Computer Vision

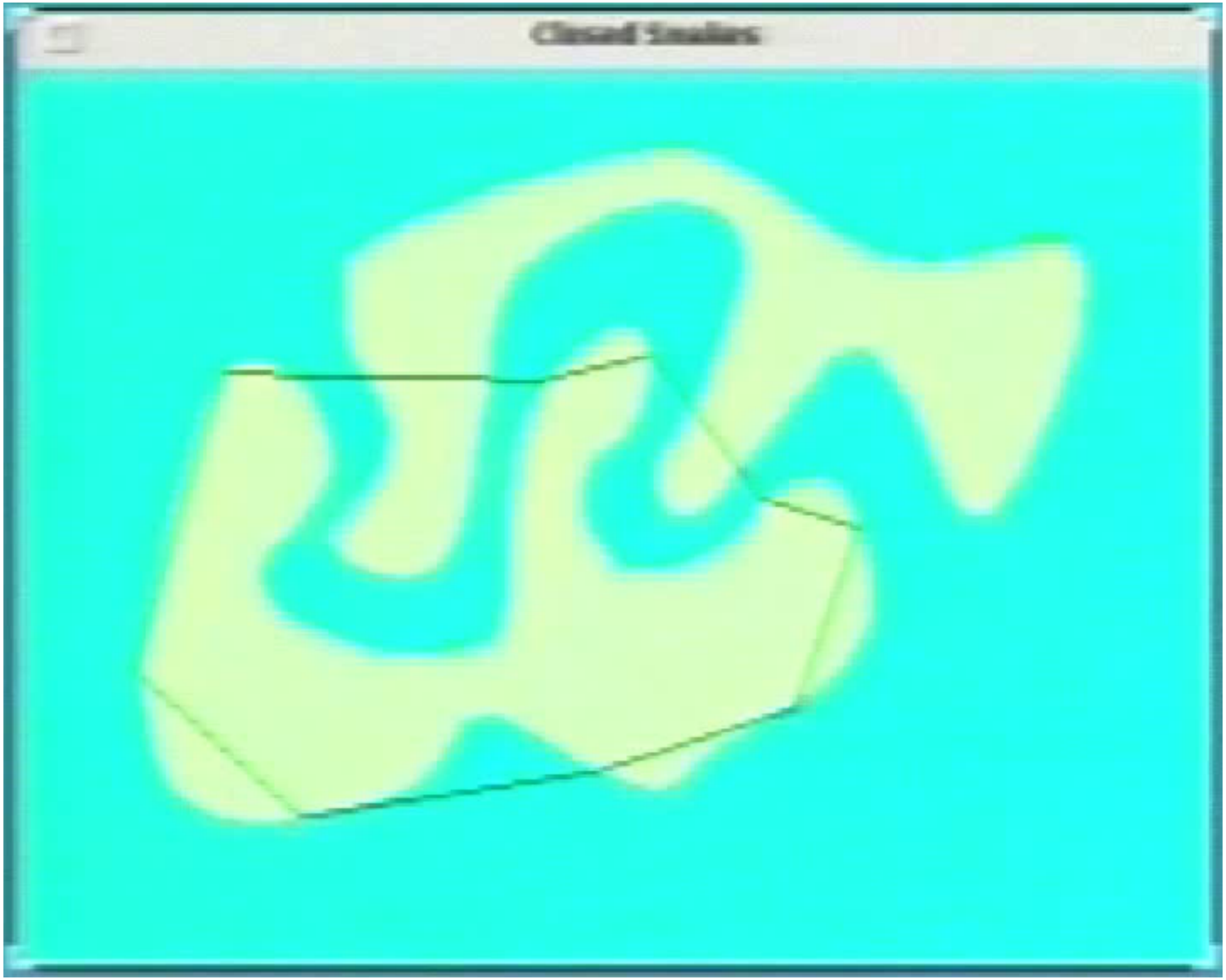
Snake example



Snakes: Interactive (example)

Computer Vision

Interactive forces example



Snakes: Interactive (example)

Tracking via deformable/contour models

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.

Computer Vision

Tracking via deformable models

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

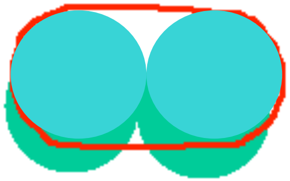
Applications: Traffic monitoring
Human-computer interaction
Animation
Surveillance
Medical imaging

Snakes: Applications

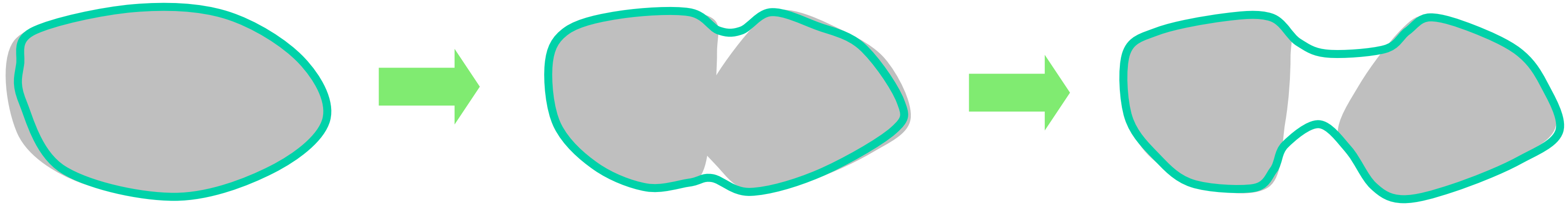
Computer Vision

Limitations

- May over-smooth the boundary



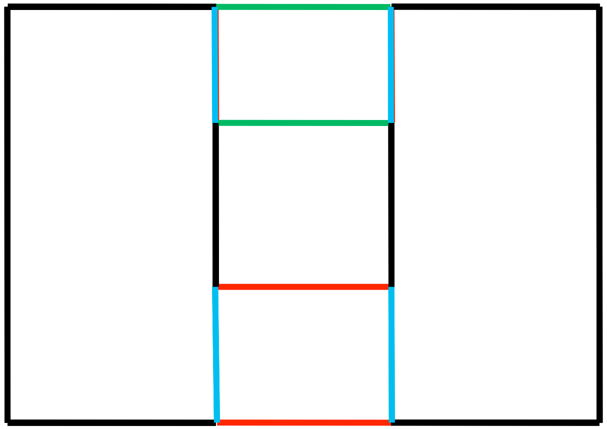
- Cannot follow topological changes of objects



Computer Vision

Limitations: Contour closing is ill-defined

Where are the gaps?



Snakes: Limitations

Computer Vision

Limitations: Only locally optimal

- Snakes only "see" nearby object boundaries
  i.e. the external energy does not consider edges in the image, unless the curve gets "very close" to it
  (determined by gradient kernel, DP search box, etc)

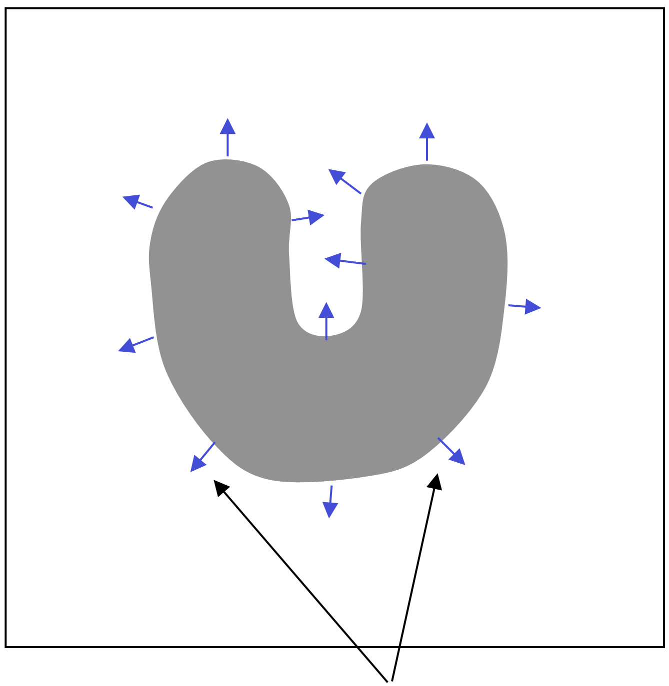
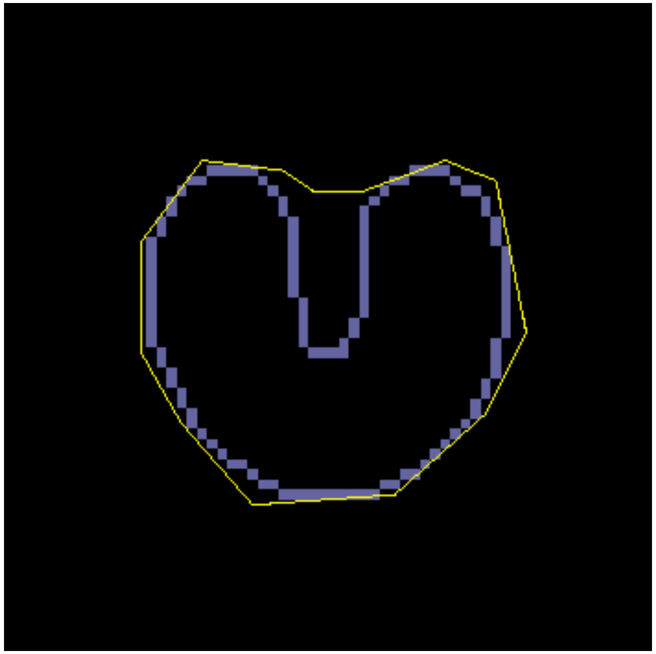


image gradients ∇I
are large only near the boundary



Snakes: Limitations

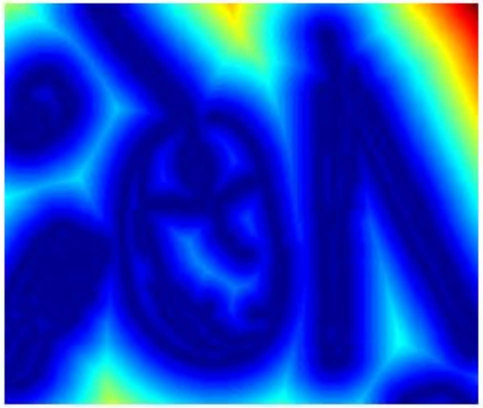
Computer Vision

Distance-based energy

- External energy can also be defined based on the **distance** from edges in the image

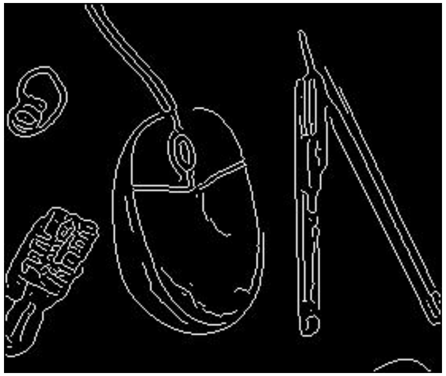
original

gradient



distance map

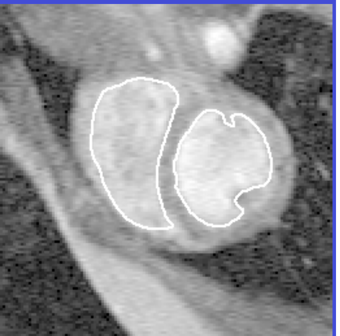
makes snake less shortsighted



edges

Value at (x,y) tells how far that position is from the nearest edge point (could also be some other binary image feature)

Snakes: Remedy to local optimality



Snakes: Pros and Cons

Pros:

- Framework to fit deformable contours via optimization
- Useful to track and fit non-rigid shapes to images
- **Contour remains connected, topology fixed**
- Possible to connect / fill in invisible contours
- **Flexibility in energy function definition** & weighting
- Enables additional energies, e.g. interactive input

Cons:

- Local optimization: may get stuck in local minimum
  Thus, needs good initialization near true boundary
- Susceptible to parameterization of energy function,
  must be set based on prior information, experience, etc

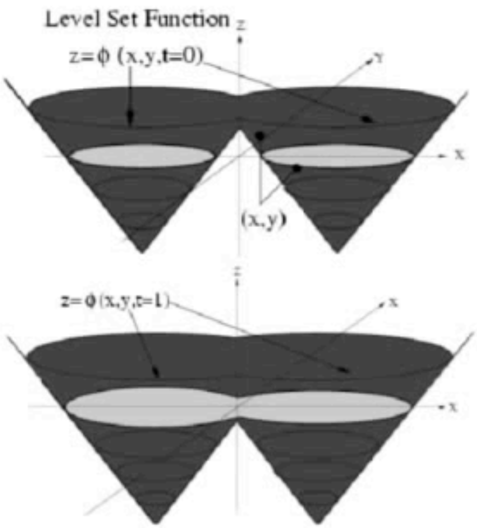
Summary: main points

- Deformable shapes and active contours are useful for
  - Segmentation: fit or "settle" to boundary in image
  - Tracking: previous frame's estimate serves to initialize the next
- Optimization for snakes: general idea of minimizing an energy function
  - Can define terms to encourage certain shapes, smoothness, low curvature, push/pulls, …
  - Can use weights to control relative influence of each component
- Edges / optima in gradients can act as "attraction" force for interactive segmentation methods.
- Distance transform definition: efficient map of distances to nearest feature of interest.

Implicit curve definitions: Level-sets

More advanced methods: Implicit Models

- Instead of representing the contour explicitly as a set of points
- Implicit models - the contour is the *level set* of a higher dimensional function



The level set function:

$$z = \Phi(x,y,t)$$

Contour at time *t:*

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

from http://www.cs.missouri.edu/~duan/tutorial-a.pdf

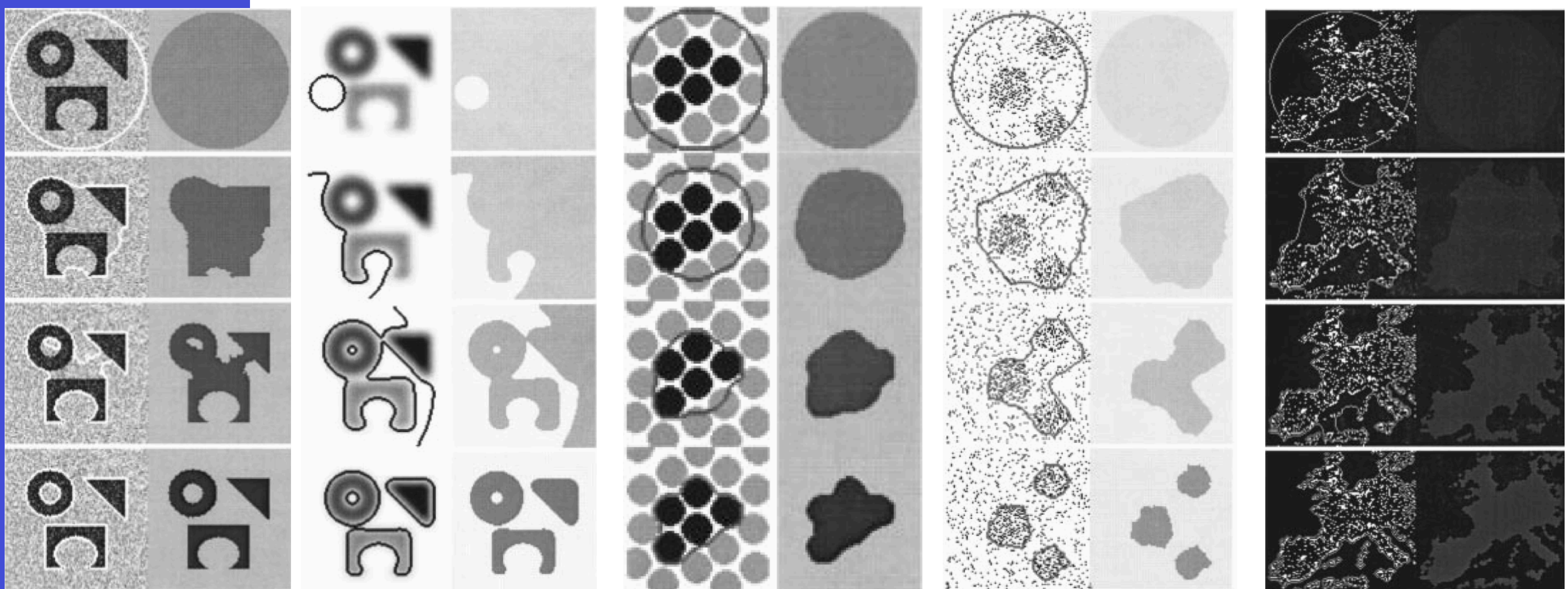
This allows the topology of the curve to change

Beyond Snakes: Implicit (level-sets)

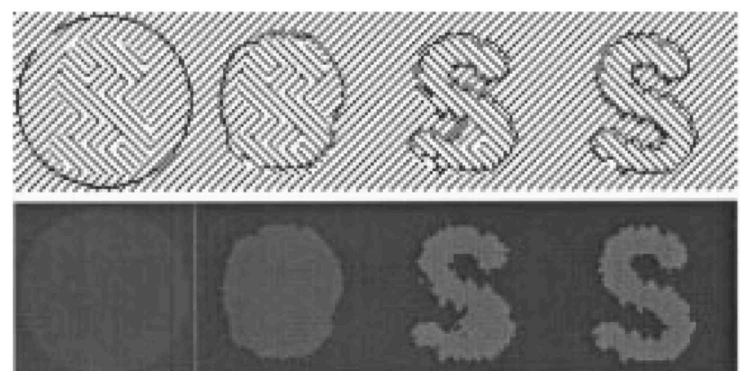
Computer Vision

"Active Contours Without Edges", Chan&Vese 2001

Adds energy term for min variance of in/out (Mumford-Shah func)



- Allows for capturing (non-edge/global) variations
- Even point-cloud distributions
- Can extend to other image features

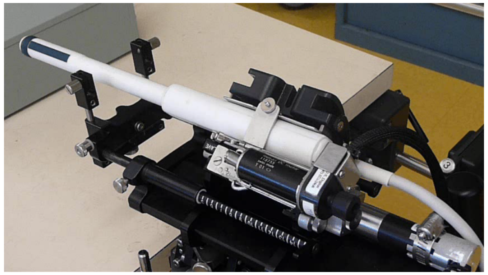


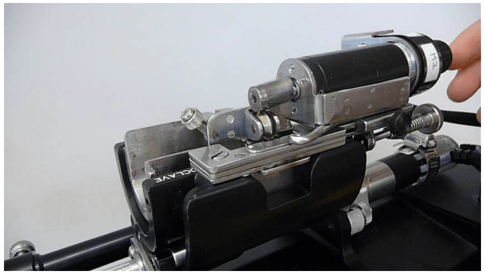
Beyond Snakes: Global energy

Computer Vision

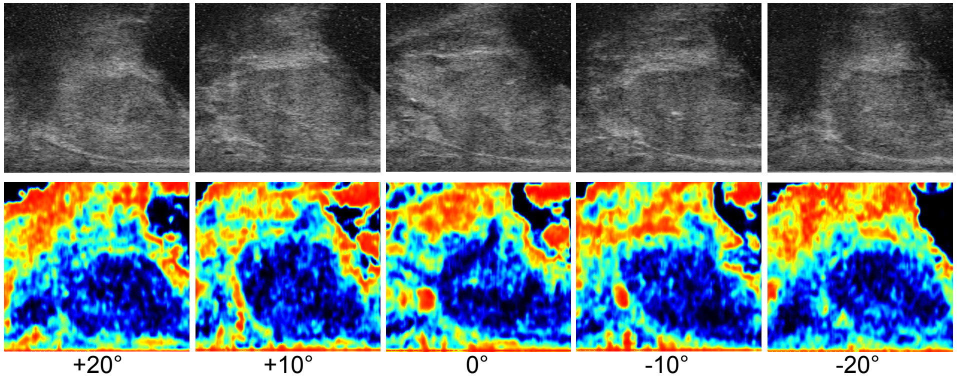
Application example: Prostate Segmentation

Prostate Ultrasound Vibroelastography

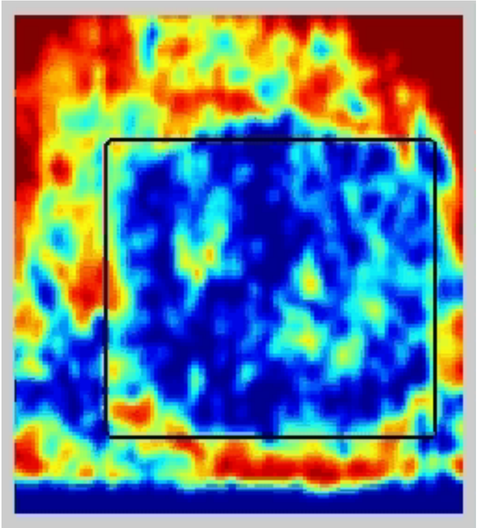


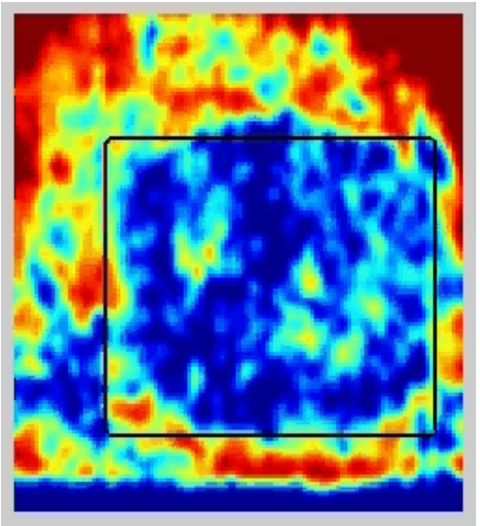


Tim Salcudean, UBC



- Elastography yields a contrast image
- 2D snakes fitted to edges
- Next slide initialized with the previous



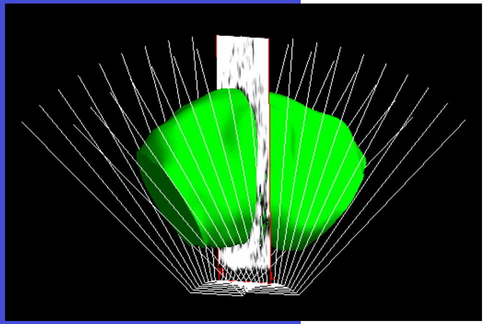


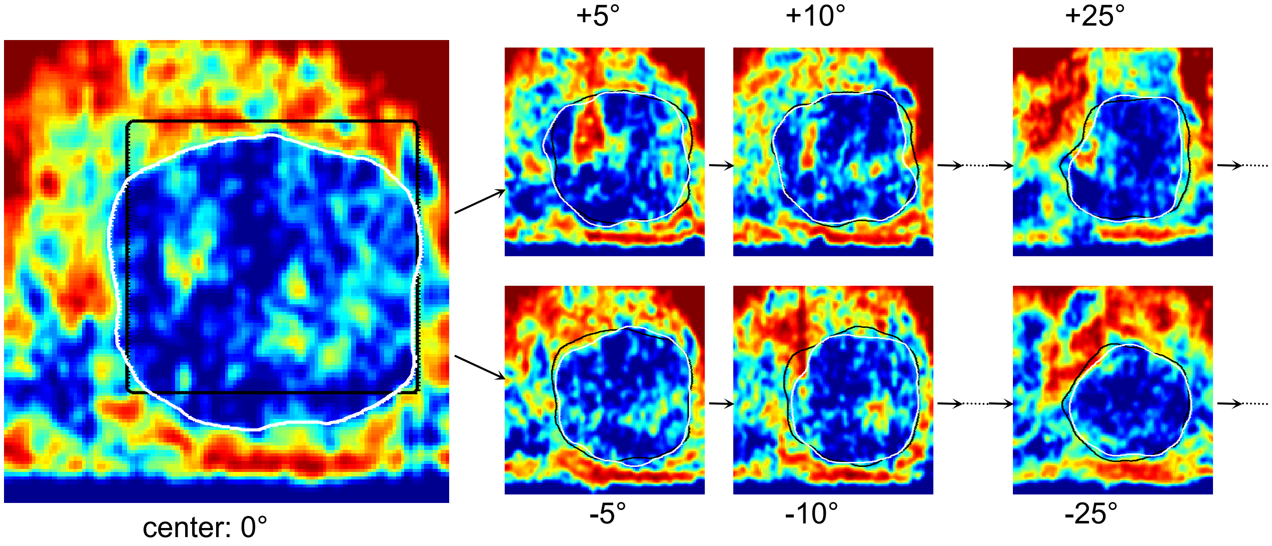
[Goksel, 2009]

Sample Application

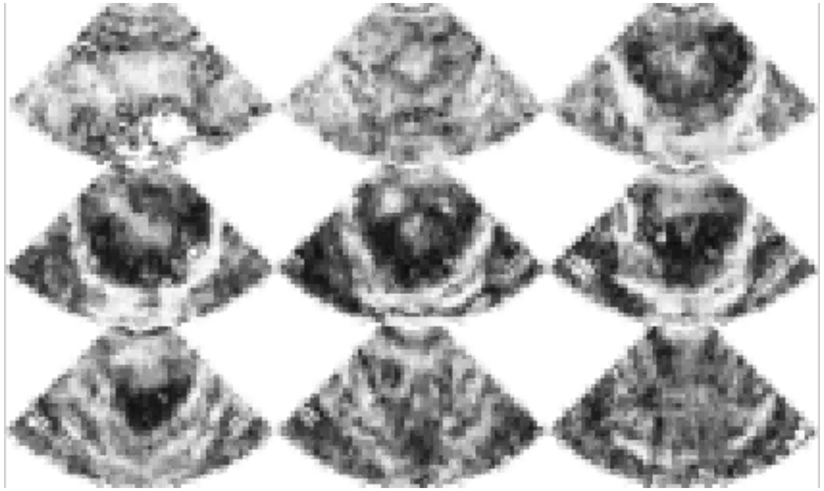
Computer Vision

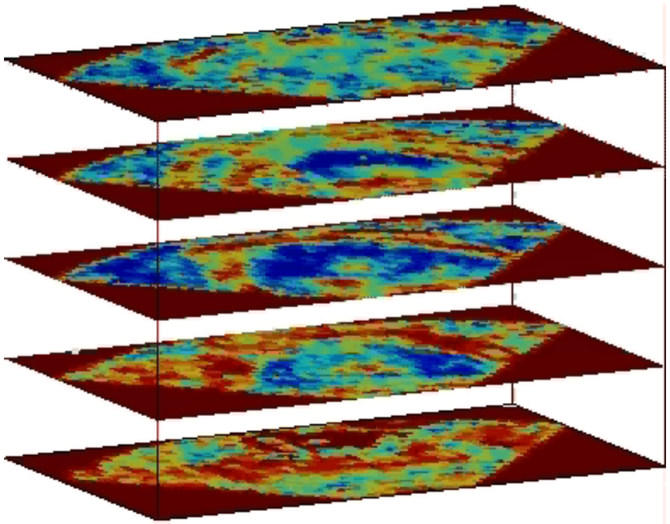
Prostate Segmentation Example in 3D





Can be also performed directly on 3D volume

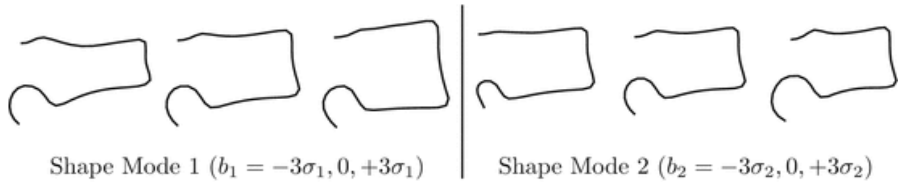




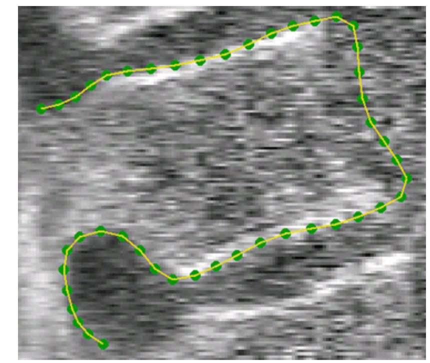
Sample Application

Active Shape and Appearance Models

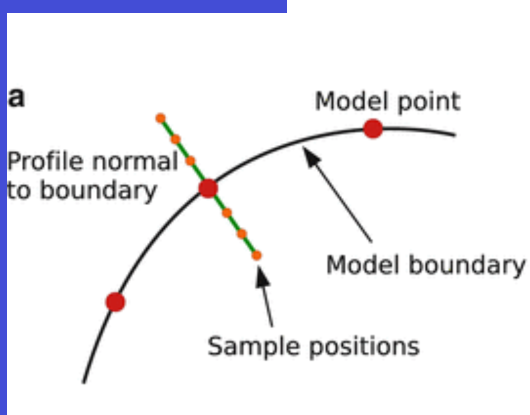
Shape model: Implicit energy via fitness to a statistical model (remember PCA)

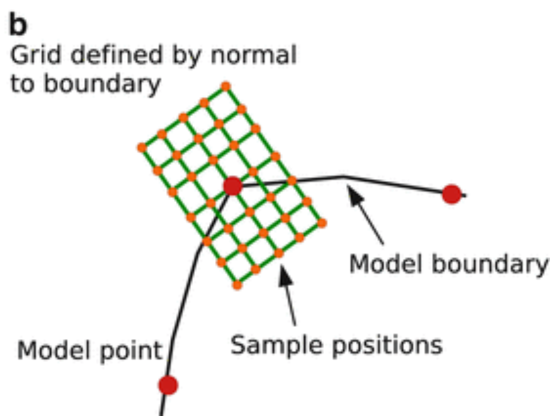


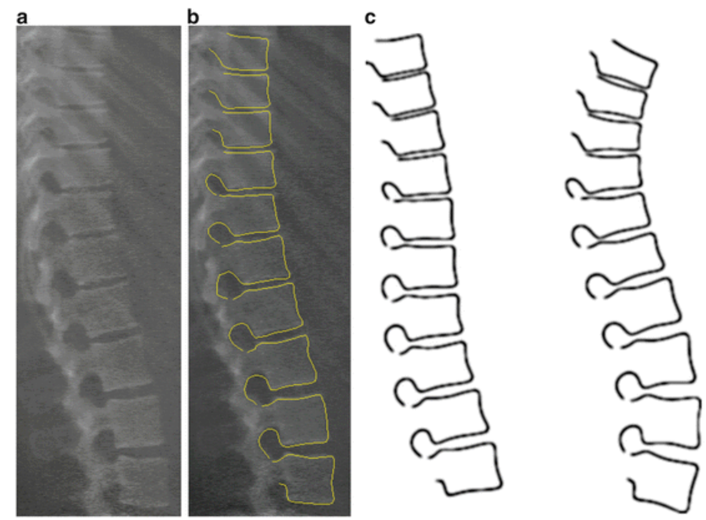
While updating the curve, project on model space to find closest shape model



Appearance model: Make a local intensity model of edge at point v_i







[Cootes et al., Handbook of Biomedical Imaging, 2015]

Example: MR bone segmentation