

# Snakes: Active Contours

Zoltan Kato

<http://www.cab.u-szeged.hu/~kato/variational/>

# Introduction

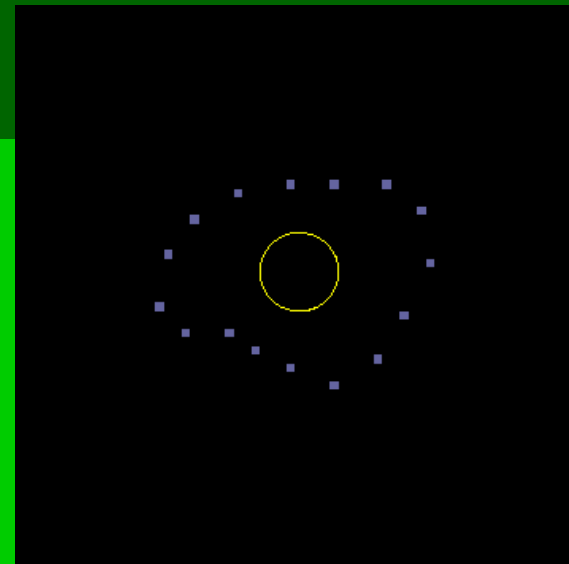
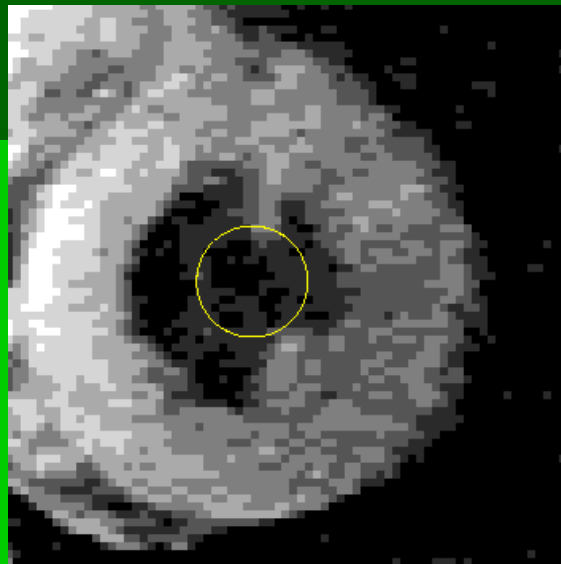
- Proposed by
  - Michael Kass
  - Andrew Witkin
    - [http://www.ri.cmu.edu/people/witkin\\_andrew.html](http://www.ri.cmu.edu/people/witkin_andrew.html)
  - **Demetri Terzopoulos**
    - <http://mrl.nyu.edu/~dt/>

**Snakes: Active Contour Models.**

*International Journal of Computer Vision,  
Vol. 1, pp 321-331, 1988.*

# What is a snake?

- An energy minimizing *spline* guided by external constraint forces and pulled by image forces toward features:
  - Edge detection
  - Subjective contours
  - Motion tracking
  - Stereo matching
  - .....



# Snake behavior

- A snake falls into the closest *local* energy minimum.
  - The local minima of the snake energy comprise the set of alternative solutions
  - A higher level knowledge is needed to choose the „*correct one*” from these solutions
    - High-level reasoning
    - User interaction
- └ These high-level methods can *interact* with the contour model by pushing it toward an appropriate local minimum

# Snake behavior

- They rely on other mechanisms to place them *near* the desired contour.
  - The existence of such an initializer is application dependent.
  - Even in the case of manual initialization, snakes are quite powerful in refining the user's input.
- Basically, snakes are trying to match a deformable model to an image by means of energy minimization.

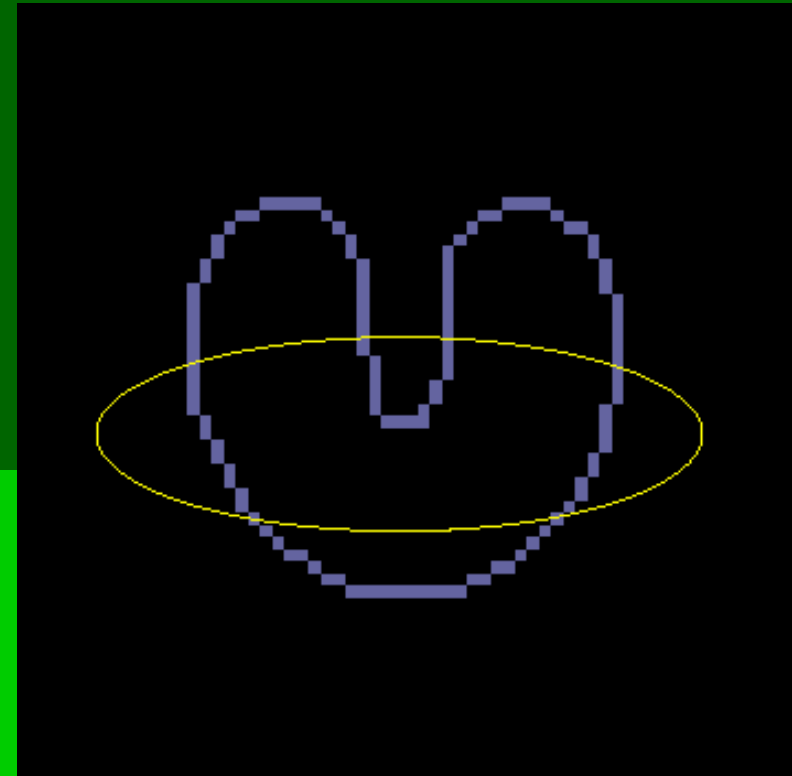


Image taken from the GVF website:  
<http://iacl.ece.jhu.edu/projects/gvf/>

# Snake energy

- Parametric representation:  $\mathbf{v}(s) = (x(s), y(s))$

$$E_{snake} = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s)) ds$$

- $E_{int}$  = internal energy due to bending. Serves to impose piecewise smoothness constraint.
- $E_{image}$  = image forces pushing the snake toward image features (edges, etc...).
- $E_{con}$  = external constraints are responsible for putting the snake near the desired local minimum. It may come from:
  - Higher level interpretation
  - User interaction, etc...

# Internal energy

- The snake is a **controlled continuity** spline

- Regularizes the problem

$$E_{\text{int}} = (\alpha(s) |v_s(s)|^2 + \beta(s) |v_{ss}(s)|^2) / 2$$

- The first order derivative  $v_s(s)$  makes the spline act like a **membrane** („elasticity”).
- The second order derivative  $v_{ss}(s)$  makes it act like a **thin-plate** („rigidity”).
- $\alpha(s)$  and  $\beta(s)$  controls the relative importance of membrane and thin-plate terms
  - Setting  $\beta(s)=0$  for a point allows the snake to become second-order discontinuous and develop a corner.

# Image forces

- Attracts the snake to features (data term)

$$E_{image} = w_{line} E_{line} + w_{edge} E_{edge} + w_{term} E_{term}$$

- **lines**: the simplest functional is the image intensity:  $E_{line} = I(x, y)$ 
  - Depending on the sign of  $w_{line}$ , the snake will be attracted to the lightest or darkest nearby contour
- **edges**: one can simply set  $E_{edge} = - |\nabla I(x, y)|^2$ 
  - attracts the snake to large intensity gradients.
- **terminations**: discussed later

# Snake convergence

- If part of a snake finds a low-energy feature  $\rightarrow$  the spline term will pull neighboring parts toward a possible continuation of the feature found.
- In fact, this places a large energy well around a good local minimum



Video taken from the website:

<http://www-2.cs.cmu.edu/afs/cs/user/aw/www/gallery.html>

# Scale space

- Minimization by scale-continuation:
  1. Spatial smooting the edge or line functional
    - $E_{edge} = -(G_\sigma * \nabla^2 I)^2$ , where  $G_\sigma$  is a Gaussian with  $\sigma$  standard deviation
    - Minima lie on zero crossings of  $G_\sigma * \nabla^2 I$  (~edges)
  2. Snake comes to equilibrium on a blurry energy
  3. Slowly reduce the blurring

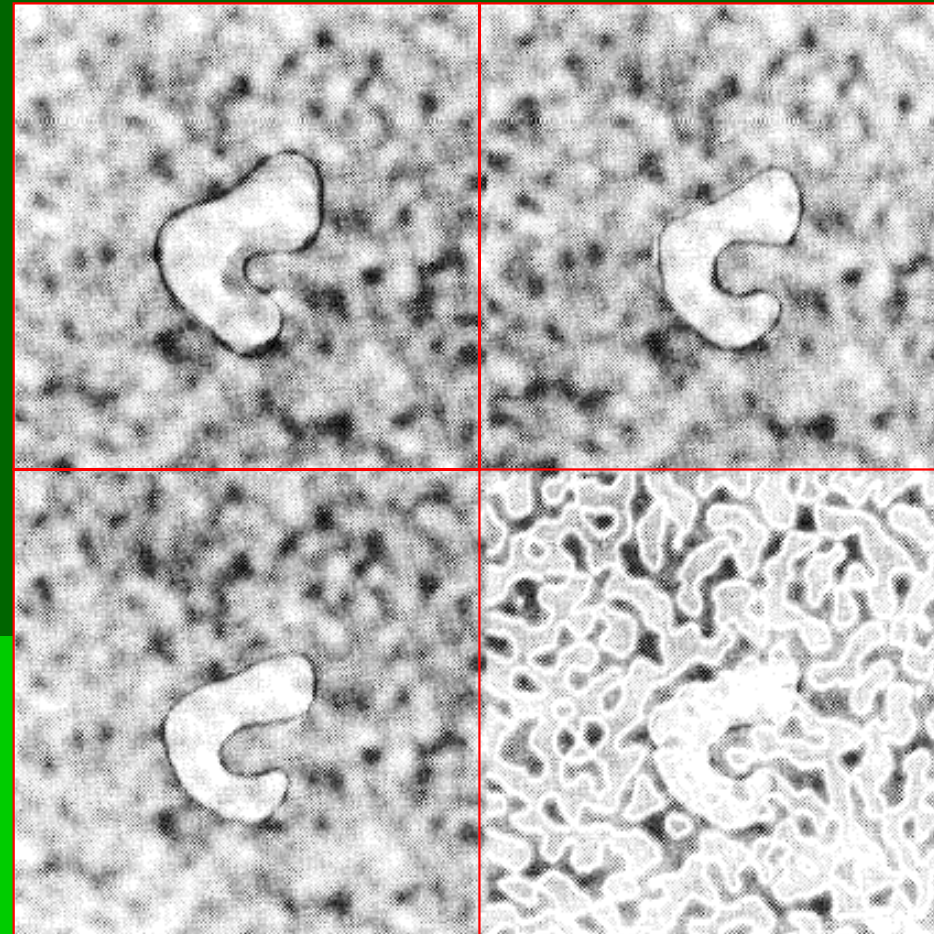


Image taken from M. Kass & A. Witkin & D. Terzopoulos: Snakes: Active Contour Models. *International Journal of Computer Vision*, Vol. 1, pp 321-331, 1988.

Zero crossings



# Termination functional

- Attracts the snake toward termination of line segments and corners.
  - Let  $C(x,y) = G_\sigma(x,y) * I(x,y)$  (smoothed image)
  - Let  $\theta = \tan^{-1}(C_y/C_x)$  the gradient angle
    - $n = (\cos \theta, \sin \theta)$  unit vector along gradient
    - $n_\perp = (-\sin \theta, \cos \theta)$  perpendicular to gradient
- ┌  $E_{term}$  is defined using curvature of level lines in

$C(x,y)$ :

$$E_{term} = \frac{\partial \theta}{\partial n_\perp} = \frac{\partial^2 C / \partial n_\perp^2}{\partial C / \partial n} = \frac{C_{yy} C_x^2 - 2C_{xy} C_x C_y + C_{xx} C_y^2}{(C_x^2 + C_y^2)^{3/2}}$$

# Subjective contour

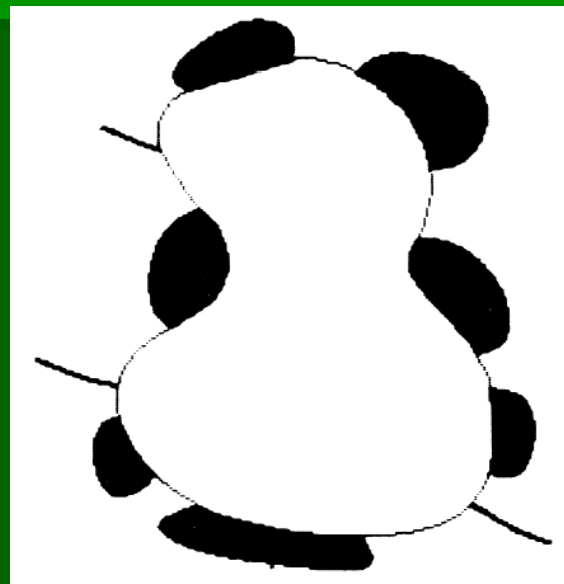
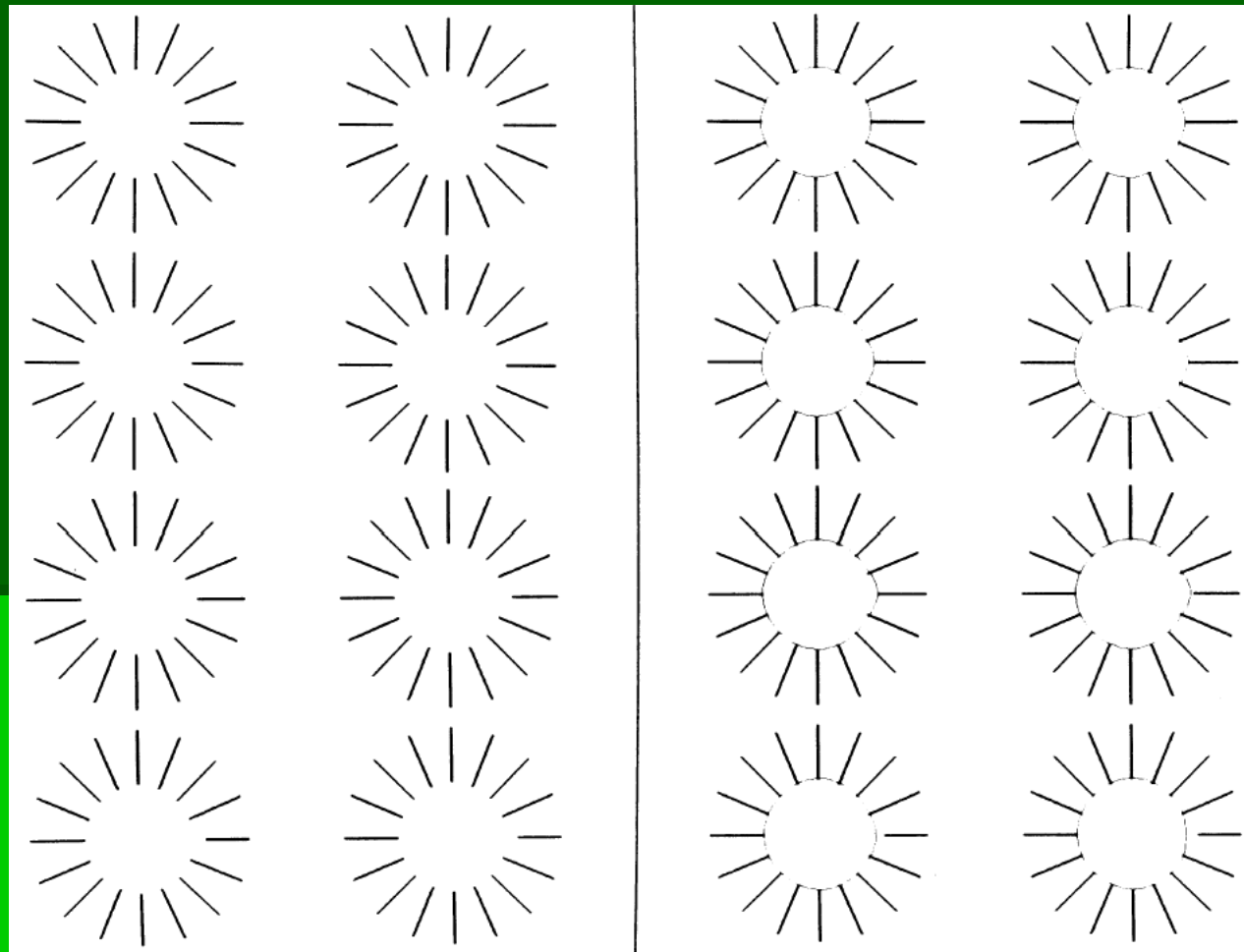


Image taken from M. Kass & A. Witkin & D. Terzopoulos: Snakes: Active Contour Models. *International Journal of Computer Vision*, Vol. 1, pp 321-331, 1988.

- Combining  $E_{edge}$  and  $E_{term}$ , we can create a snake attracted to edges and terminations
- The shape of the snake between the edges and lines in the illusion is completely determined by the spline smoothness term
- The same snake can find traditional edges in natural images

# Subjective contour: hysteresis

- Snake tracking a moving subjective contour
- The snake bends until the internal spline forces overpower image forces
- Then the snake falls off the line and returns to a smoother shape



# Motion tracking

- Once a snake finds a feature, it „locks on”
- If the feature begins to move, the snake will track the same local minimum
  - └ Fast motion could cause the snake to flip into a different minimum

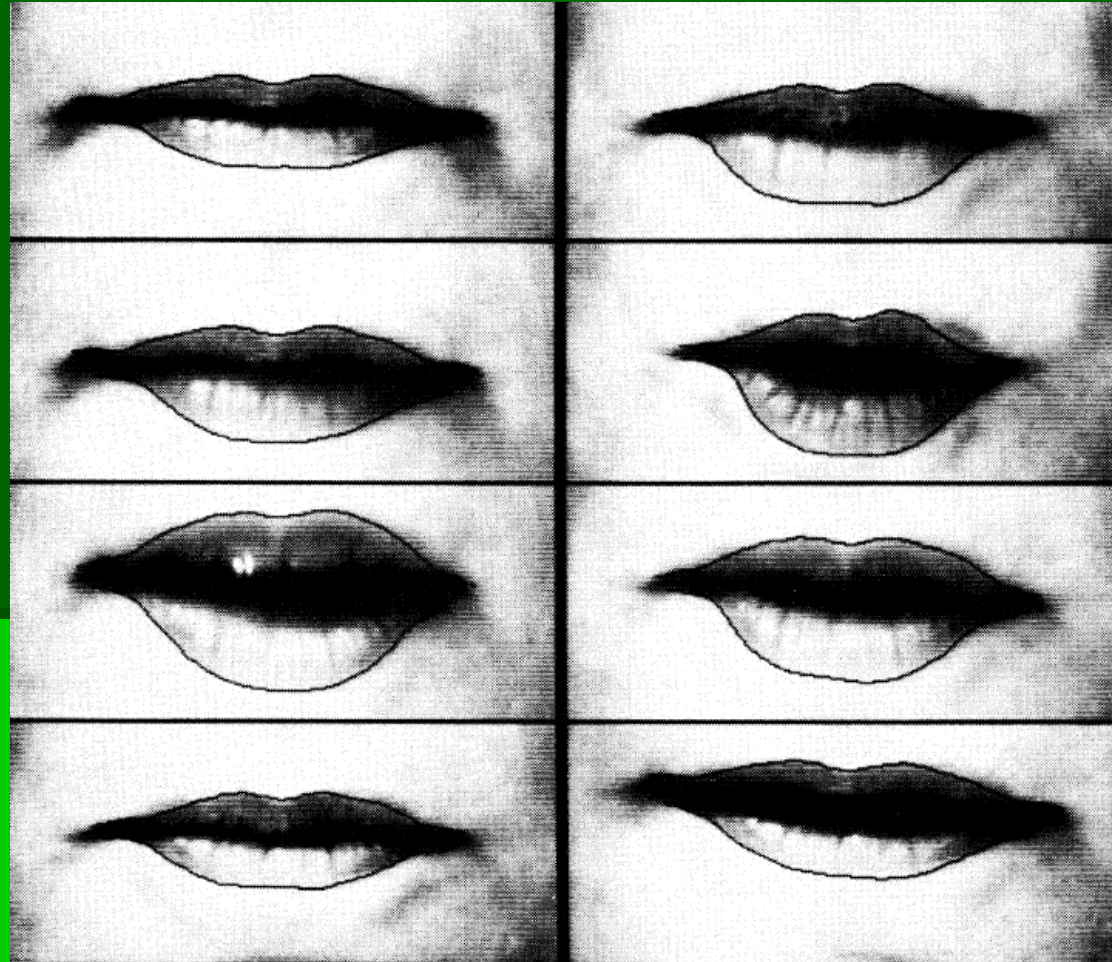


Image taken from M. Kass & A. Witkin & D. Terzopoulos: Snakes: Active Contour Models. *International Journal of Computer Vision*, Vol. 1, pp 321-331, 1988.

# Snake energy minimization

- When  $\alpha(\mathbf{s})$  and  $\beta(\mathbf{s})$  are **constant**, we get two

$$\alpha x_{ss} + \beta x_{ssss} + \frac{\partial E_{ext}}{\partial x} = 0$$

$$\alpha y_{ss} + \beta y_{ssss} + \frac{\partial E_{ext}}{\partial y} = 0$$


independent Euler-Lagrange equations.

- When  $\alpha(\mathbf{s})$  and  $\beta(\mathbf{s})$  are **not constant** then it is simpler to use a discrete formulation:

$$E_{snake} = \sum_{i=1}^n E_{int}(i) + E_{ext}(i), \quad v_i = (x_i, y_i) = (x(ih), y(ih))$$

$$E_{int}(i) = \alpha_i |v_i - v_{i-1}|^2 / 2h^2 + \beta_i |v_{i-1} - 2v_i + v_{i+1}|^2 / 2h^4$$

# Snake energy minimization

- Let  $f_x(i) = \partial E_{\text{ext}} / \partial x_i$  where derivatives are approximated by finite differences if they cannot be computed analytically.
- The corresponding Euler equations 
- In matrix form where  $A$  is a pentadiagonal banded matrix:

$$\begin{aligned} & \alpha_i(v_i - v_{i-1}) - \alpha_{i+1}(v_{i+1} - v_i) \\ & + \beta_{i-1}(v_{i-2} - 2v_{i-1} + v_i) \\ & - 2\beta_i(v_{i-1} - 2v_i + v_{i+1}) \\ & + \beta_{i+1}(v_i - 2v_{i+1} + v_{i+2}) \\ & + (f_x(i), f_y(i)) = 0 \end{aligned}$$

$$\begin{aligned} Ax + f_x(x, y) &= 0 \\ Ay + f_y(x, y) &= 0 \end{aligned}$$

# Snake energy minimization

$$Ax_t + f_x(x_{t-1}, y_{t-1}) = -\gamma(x_t - x_{t-1})$$

$$Ay_t + f_y(x_{t-1}, y_{t-1}) = -\gamma(y_t - y_{t-1})$$

$\gamma$  is the step size

- Taking into account the derivatives requires changing **A** at each iteration. Speed up:
  - We assume that **f<sub>x</sub>** and **f<sub>y</sub>** are constant during a time step → explicit Euler method w.r.t. the external forces.
  - internal forces are specified by **A** → we can evaluate the time derivative at **t** rather than **t-1**

# Snake energy minimization

- At equilibrium, the time derivative vanishes.
- The Euler equations can be solved by matrix inversion:

$$x_t = (A + \gamma I)^{-1} (\gamma x_{t-1} - f_x(x_{t-1}, y_{t-1}))$$

$$y_t = (A + \gamma I)^{-1} (\gamma y_{t-1} - f_y(x_{t-1}, y_{t-1}))$$

- The inverse can be calculated by **LU** decomposition in  **$O(n)$**  time.