Announcements

- Project reviews due today, but later this week is ok too
- No office hours for Kobus today
- Scott is available after class
- Class will end early today
Snakes: Active Contours

Scott Morris
Department of Computer Science
University of Arizona
Example image

Apply Canny edge detector

Snakes-Active Contours

- Seminal paper by Kass et al – ICCV 1987
- Goal: find image contours using energy minimizing “snakes”

\[ E_{\text{Snake}} = \frac{1}{0} \left\{ E_{\text{internal}}(v(s)) + E_{\text{external}}(v(s)) \right\} ds \]

- \( v(s) \) is a parametric curve, the snake (initialized somehow)
- By analogy to physical systems, define an energy functional along the snake with properties designed to extract contours.
Internal vs. External energy

- Internal energy depends on intrinsic properties of the snake such as length and curvature.
- External energy depends on external information—the image, or other user defined constraints.

\[
E_{\text{internal}} = \left[ \frac{\partial}{\partial s} v(s) \right] + \left[ \frac{\partial^2}{\partial s^2} v(s) \right]
\]

\[
E_{\text{external}} = \nabla \nabla (G \nabla I)
\]

- \(G\) are constants – tuning parameters set by the user
- \(v(s)\) is the snake (initialized somehow), \(I()\) is the image function
Internal Energy

- As defined by Kass.

- Want snake to shrink around object, so need define an “elastic” term:

\[
\left| \frac{\partial}{\partial s} \nu(s) \right|
\]

- Simply the first derivative of the curve. Implemented by finite differences:

\[
(x_i \frac{\partial x_i}{\partial i})^2 + (y_i \frac{\partial y_i}{\partial i})^2
\]

- Has the effect of minimizing distance between points, which serves to shrink the snake
Snake acting only under elastic internal energy

Internal Energy, Bending

- Would also like the snake to stay “smooth”, so we penalize high curvature by minimizing the second derivative:

\[
\left| \frac{\partial^2}{\partial s^2} \nu(s) \right|
\]

- Approximated by finite difference as:

\[
(x_{i-1} \Box 2x_i + x_{i+1}) + (y_{i-1} \Box 2y_i + y_{i+1})
\]
Snake acting under bending energy

Would like snake to attract to edges in the image in order to find contours

Kass defines external energy as:

\[ E_{external} = \nabla \nabla (G \nabla I) \]

Essentially: convolve image with gaussian and gradient approximation mask (any method will do)

Negative sign moves snake towards edges (since we are minimizing the energy along the snake)

Has the effect of locking snake on edges
Snake moving under external energy (gradient)

Solving snakes

- Solve using variational calculus to get a differential equation, then use gradient descent
- Dynamic programming
- Greedy method
Greedy Algorithm

- Makes locally optimal choices, but may not find a global minimum
- Let \((x_1, y_1), \ldots, (x_N, y_N)\) be points on snake
- One iteration of greedy algorithm:

\[
E = \text{Gradient of Gaussian blurred input image}
\]

For \(i = 1:N\)

\[
\text{min\_energy} = \text{DBL\_MAX}
\]

for each pixel in MxM neighborhood of \((x_i, y_i)\)

compute change in energy by moving \((x_i, y_i)\) here

end

move \((x_i, y_i)\) to position with minimum energy

end
Greedy Algorithm (cont)

- Keep iterating until stop criteria is met:
  - No points move
  - Set number of iterations
  - Overall energy of snake only changes slightly

Practical considerations:

- Must normalize the contribution of each energy term (so they are weighted equally)
- Fast – O(MN) complexity
Problems

- Largest – Snakes are a “power assist” method as they require initialization which is usually done manually
  - Random initialization of many snakes
  - Use conventional methods to seed snakes
  - Use snakes to further improve output of other processes
  - Object tracking—can predict next location so move snake accordingly

- Some tuning of parameters is often necessary, depending on the image/application
Extensions to snakes

- Snakes do not need to be closed loops. Fix end points, or determine energy for end points.
- Loop can be made to inflate, or mimic many physical properties (membranes, strings, plates, etc)
- Can attract/repel to image features other than edges—texture, color, etc.
- Affine snakes – can transform only by expansion, rotation and shearing. Useful for tracking objects moving relative to a camera.
- Many, many variations of energy functionals for various applications
\[ E_{\text{snake}} = \frac{1}{0} \left\{ E_{\text{internal}}(v(s)) + E_{\text{external}}(v(s)) \right\} ds \]

\[ E_{\text{gps\ int}} = \left| \nabla \left( g(s) \nabla v(s) \right) \right|^2 \]

\[ E_{\text{gps\ ext}} = \nabla I(x, y) + \nabla \left( \frac{\partial I}{\partial \nabla(s)} \right) \]

are constant tuning parameters

g(s) is the original snake – the unmodified GPS data

I(x,y) is the image at v(s)

\( \_\_\_\_ (s) \) is the direction perpendicular to the snake
Output of GPS-Snakes

\[ n = 1.5 \]
\[ n = 0.5 \]
\[ n = 10 \]