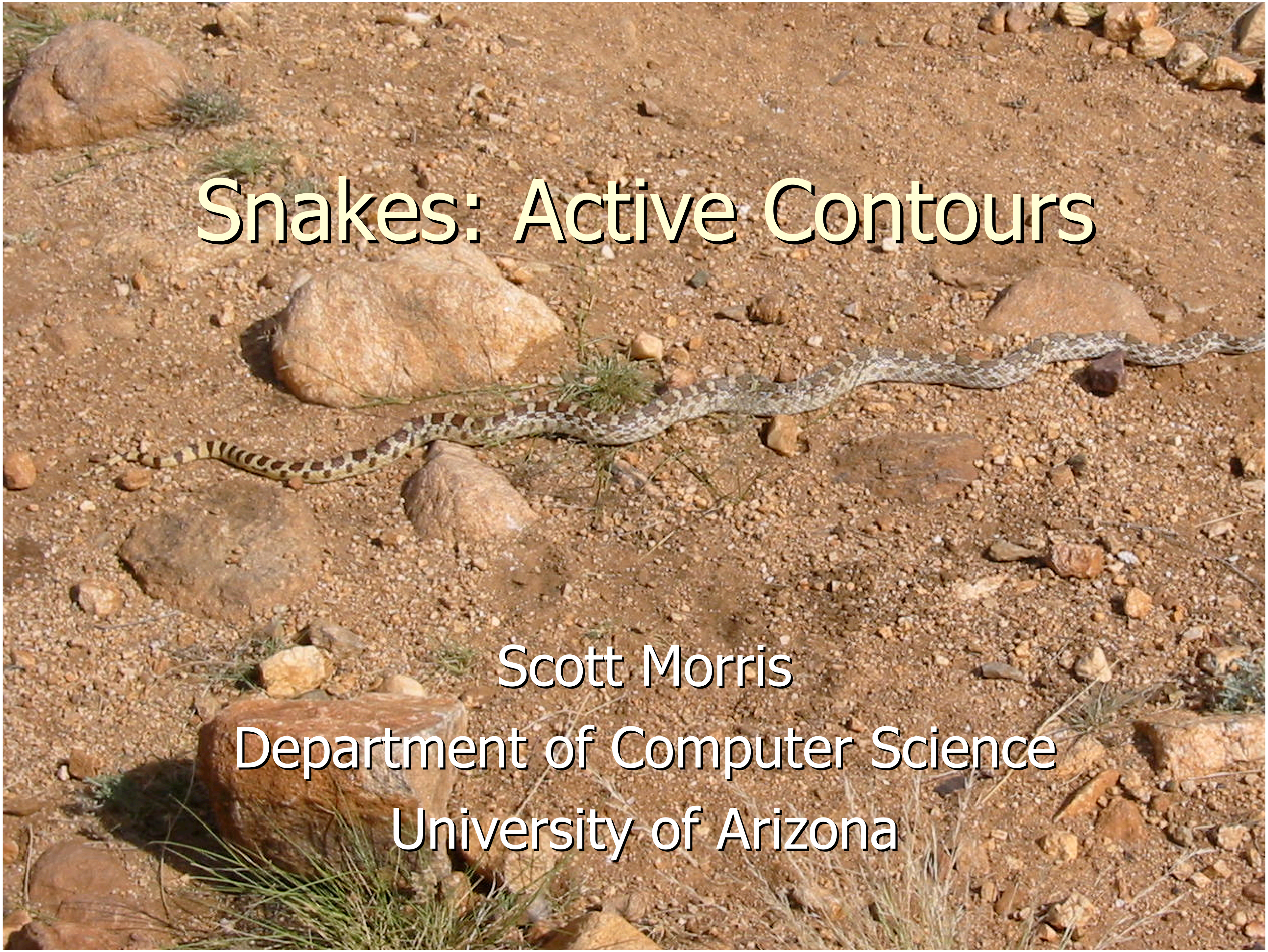
Announcements

- 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



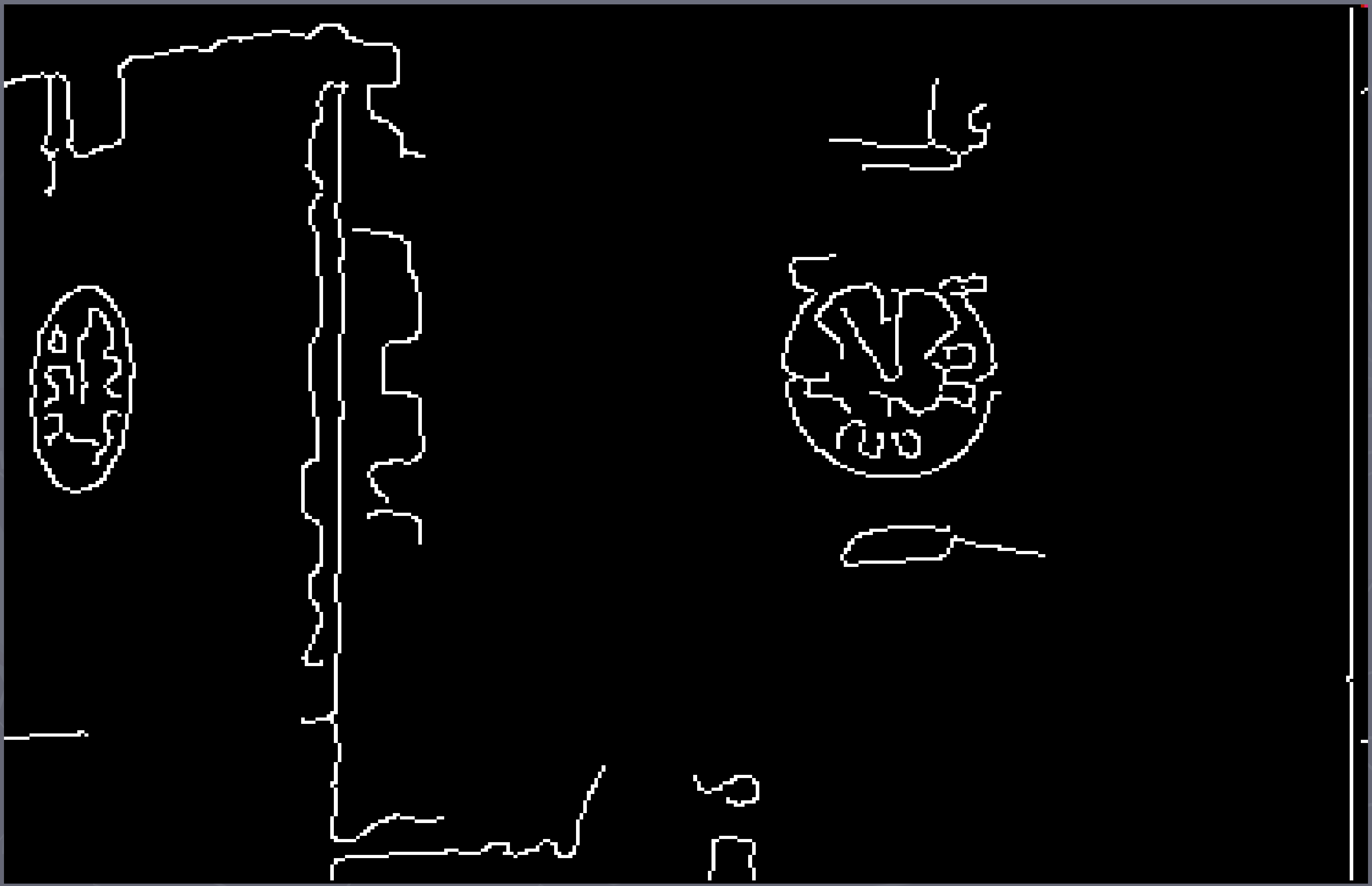
Snakes: Active Contours

Scott Morris
Department of Computer Science
University of Arizona

Example image

David Young, 1995, http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision7.html

Apply Canny edge detector



David Young, 1995, http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision7.html

Snakes-Active Contours

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

$$E_{Snake} = \int_0^1 \{E_{internal}(v(s)) + E_{external}(v(s))\} ds$$

- 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.

$$E_{\text{int}ernal} = \square\left|\frac{\partial}{\partial s}v(s)\right| + \square\left|\frac{\partial^2}{\partial s^2}v(s)\right|$$

$$E_{external} = \square\square(G_{\square} \quad \square I)$$

- _,_,_ 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:

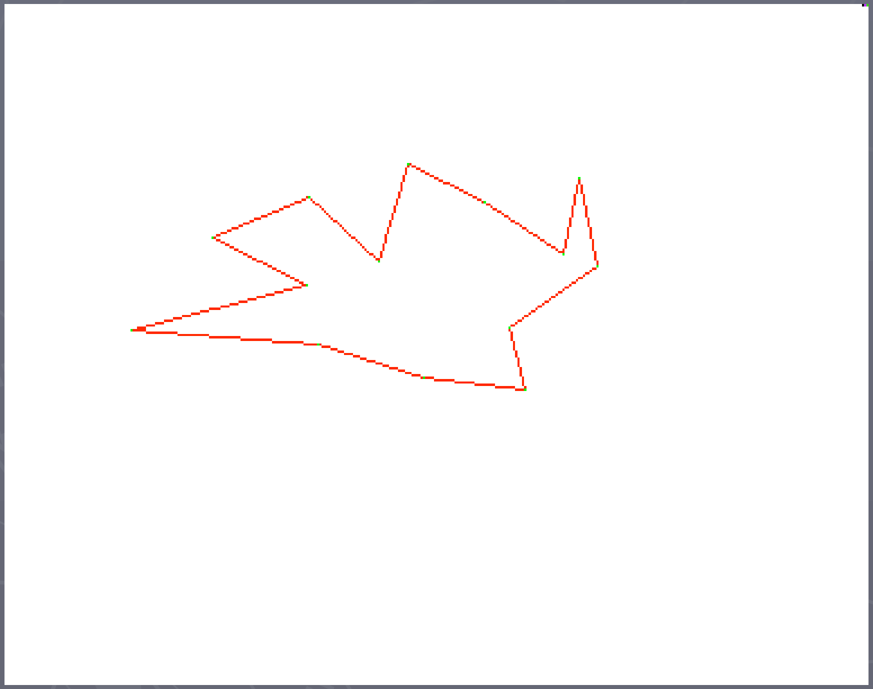
$$\alpha \left| \frac{\partial}{\partial s} v(s) \right|$$

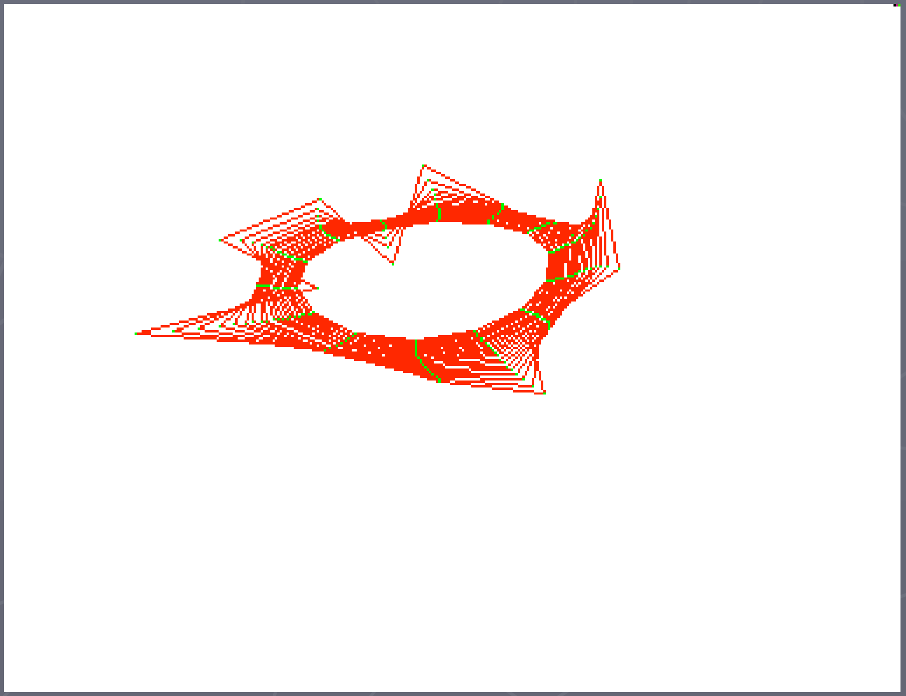
- Simply the first derivative of the curve. Implemented by finite differences:

$$(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$$

- Has the effect of minimizing distance between points, which serves to shrink the snake

Snake acting only under elastic internal energy





David Young, 1995, http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision7.html

Internal Energy, Bending

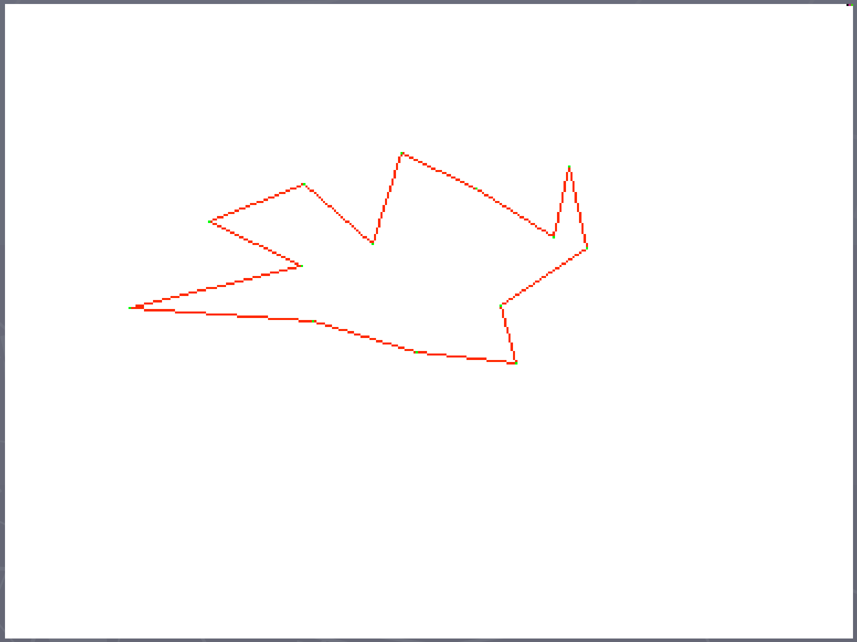
- Would also like the snake to stay "smooth", so we penalize high curvature by minimizing the second derivative:

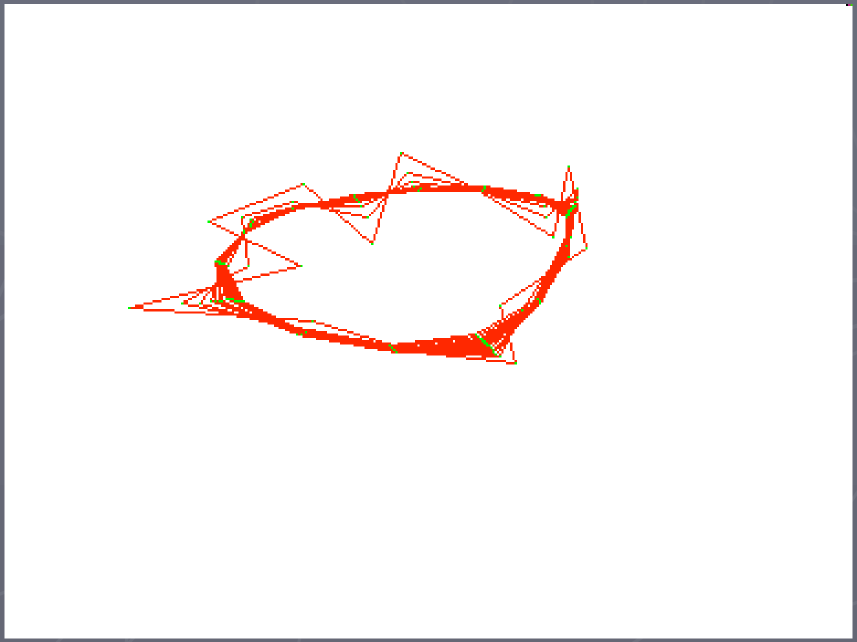
$$\square\left|\frac{\partial^2}{\partial s^2}v(s)\right|$$

- Approximated by finite difference as:

$$(x_{i-1} - 2x_i + x_{i+1}) + (y_{i-1} - 2y_i + y_{i+1})$$

Snake acting under bending energy





David Young, 1995, http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision7.html

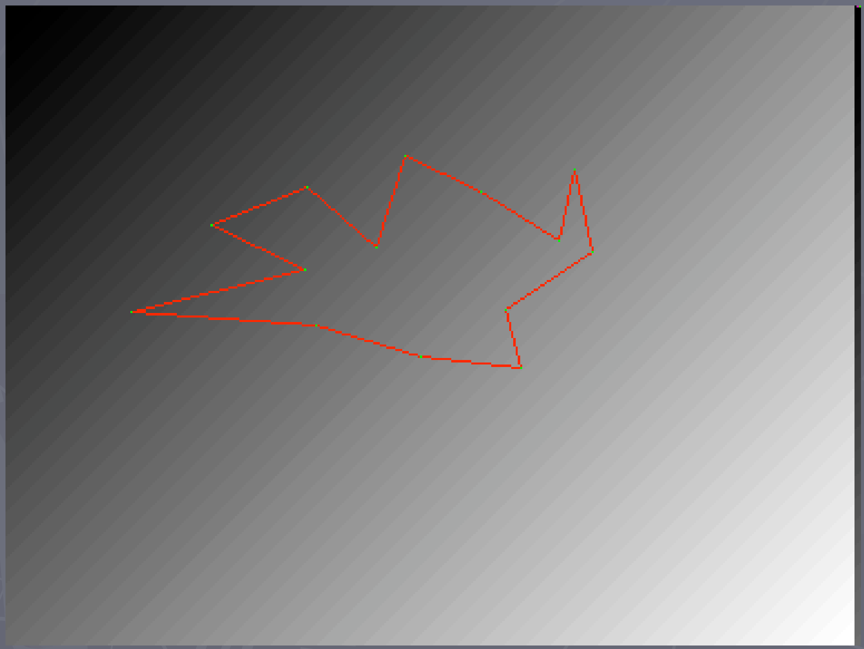
External Energy – Image

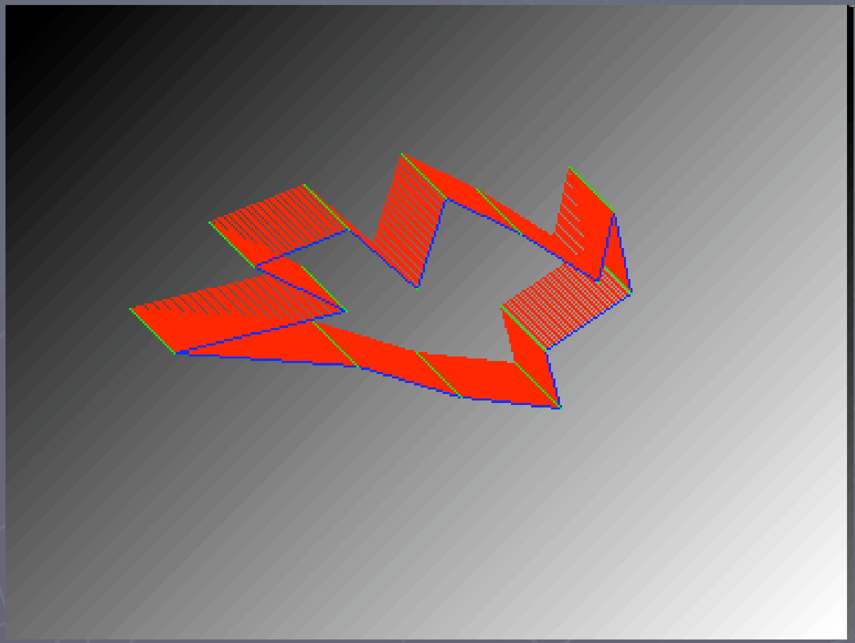
- Would like snake to attract to edges in the image in order to find contours
- Kass defines external energy as:

$$E_{external} = -\nabla(G_{\sigma} * I)$$

- 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)

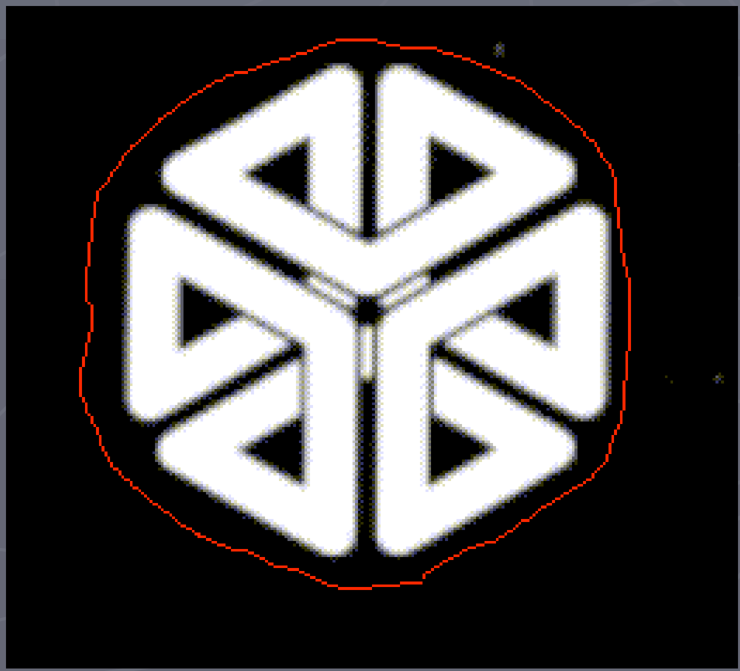




David Young, 1995, http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision7.html

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) .. (x_N ,y_N) be points on snake
- One iteration of greedy algorithm:

```
E = Gradient of Gaussian blurred input image
For i=1:N
    min_energy = DBL_MAX
    for each pixel in MxM neighborhood of (xi,yi)
        compute change in energy by moving (xi,yi) here
    end
    move (xi,yi) 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

GPS-Snakes

$$E_{snake} = \int_0^1 \{E_{\text{int}\,ernal}(v(s)) + E_{external}(v(s))\}ds$$

$$E_{gps\square\text{int}} = \square\left|g(s)\,\square\,v(s)\right|^2$$

$$E_{gps\square ext} = \square I(x,y) + \square\frac{\partial I}{\partial\square(s)}$$

_, _, _ 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

_ = 1.5

_ = 0.5

_ = 10