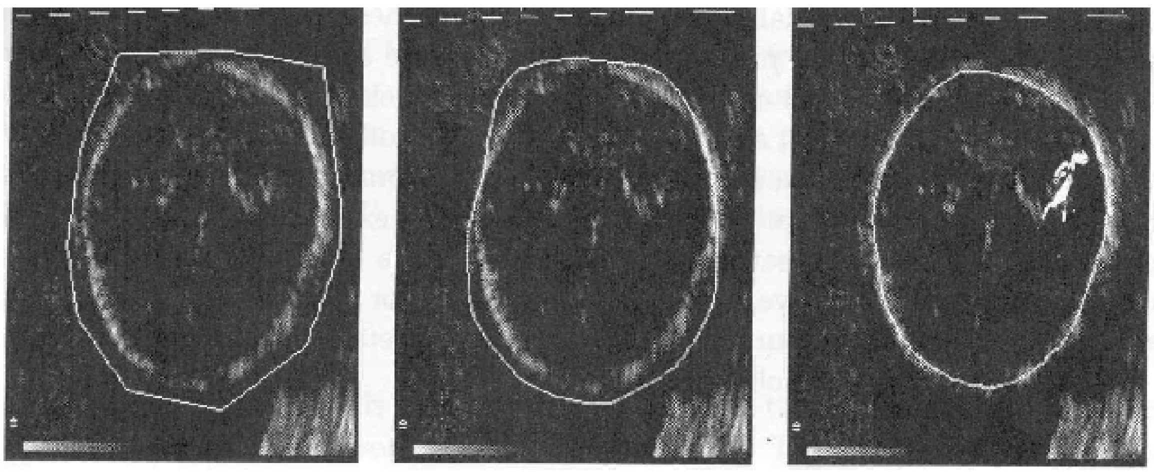
Deformable/Active Contours (or Snakes)

(Trucco, Chapt 4)

- The goal is to find a contour that best approximates the perimeter of an object.

- It is helpful to visualize it as a rubber band of arbitrary shape that is capable of deforming during time, trying to get as close as possible to the target contour.

- It is applied to the gradient magnitude of the image, not to the edge points (e.g., like the Hough transform).



• Procedure

- Snakes do not solve the entire problem of finding contours in images.

- They depend on other mechanisms such as interaction with a user or with some other higher-level computer vision mechanism:

(1) First, the snake is placed near the image contour of interest.

(2) During an iterative process, the snake is attracted towards the target contour by various forces that control the *shape* and *location* of the snake within the image.

• Approach

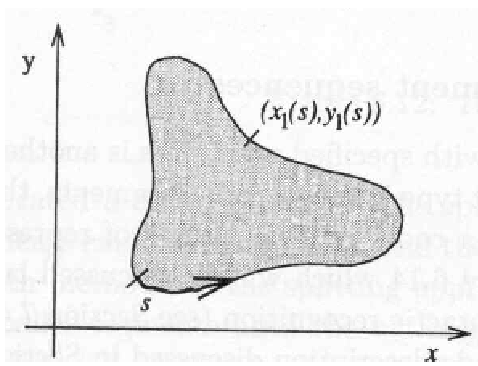
- It is based on constructing an *energy functional* which measures the appropriateness of the contour.

- Good solutions correspond to *minima* of the functional.

- The goal is to minimize this functional with respect to the contour parameters.

• Contour parameterization

- The snake is a contour represented parametrically as $c(s) = (x(s), y(s))$ where $x(s)$ and $y(s)$ are the coordinates along the contour and $s \in [0,1]$



• The energy functional

- The energy functional used is a sum of several terms, each corresponding to some force acting on the contour.

- A suitable energy functional is the sum the following three terms:

$$E = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{image})ds$$

- The parameters α, β, and γ control the relative influence of the corresponding energy terms and can vary along c.

• Interpretation of the functional's terms

- Each energy term serves a different purpose:

$\underline{E_{image}}$: it attracts the contour toward the closest image edge.

$\underline{E_{cont}}$: it forces the contour to be *continuous*.

$\underline{E_{curv}}$: it forces the contour to be *smooth*.

- E_{cont} and E_{curv} are called internal energy terms.

- E_{image} is called external energy term.

• The continuity term

- Minimize the first derivative:

$$E_{cont} = \| \frac{dc}{ds} \|^2$$

- In the discrete case, the contour is approximated by N points $p_1, p_2, \ldots, p_N$ and the first derivative is approximated by a finite difference:

$$E_{cont} = \|p_i - p_{i-1}\|^2 \text{ or}$$

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

- This term tries to minimize the distance between the points, however, it has the effect of causing the contour to shrink.

- A better form for E_{cont} is the following:

$$E_{cont} = (\bar{d} - \|p_i - p_{i-1}\|)^2$$

where $\bar{d}$ is the average distance between the points of the snake.

- The new E_{cont} attempts to keep the points at equal distances (i.e, spread them equally along the snake).

• The smoothness term

- The purpose of this term is to enforce smoothness and avoid oscillations of the snake by penalizing high contour curvatures.

- Minimize the second derivative (curvature):

$$E_{curv} = \| \frac{d^2 c}{ds^2} \|^2$$

- In the discrete case, the curvature can be approximated by the following finite difference:

$$E_{curv} = \| p_{i-1} - 2p_i + p_{i+1} \|^2 \text{ or}$$

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

• The edge attraction term

- The purpose of this term is to attract the contour toward the target contour.

- This can be achieved by the following function:

$$E_{image} = - \|\nabla I\|$$

where ∇I is the gradient of the intensity computed at each snake point.

- Note that E_{image} becomes very small when the snake points get close to an edge.

• Discrete formulation of the problem

Assumptions

Let I be an image and $\bar{p}_1$, ..., $\bar{p}_N$ the initial locations of the snake (evenly spaced, chosen close to the contour of interest).

Problem Statement

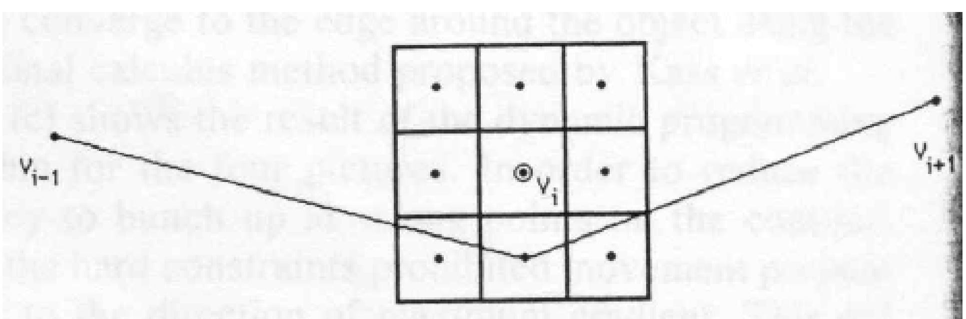
Starting from $\bar{p}_1$, ..., $\bar{p}_N$, find the deformable contour p_1, ..., p_N which fits the target contour by minimizing the energy functional:

$$\sum_{i=1}^{N}(\alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{image})$$

• A greedy algorithm

- A greedy algorithm makes *locally optimal choices*, hoping that the final solution will be *globally optimum*.

Step1 (greedy minimization): each point of the snake is moved within a small neighborhood (e.g., MxM) to the point which minimizes the energy functional



Step 2 (corner elimination): search for corners (curvature extrema) along the contour; if a corner is found at point p_j, set β_j to zero.

Algorithm

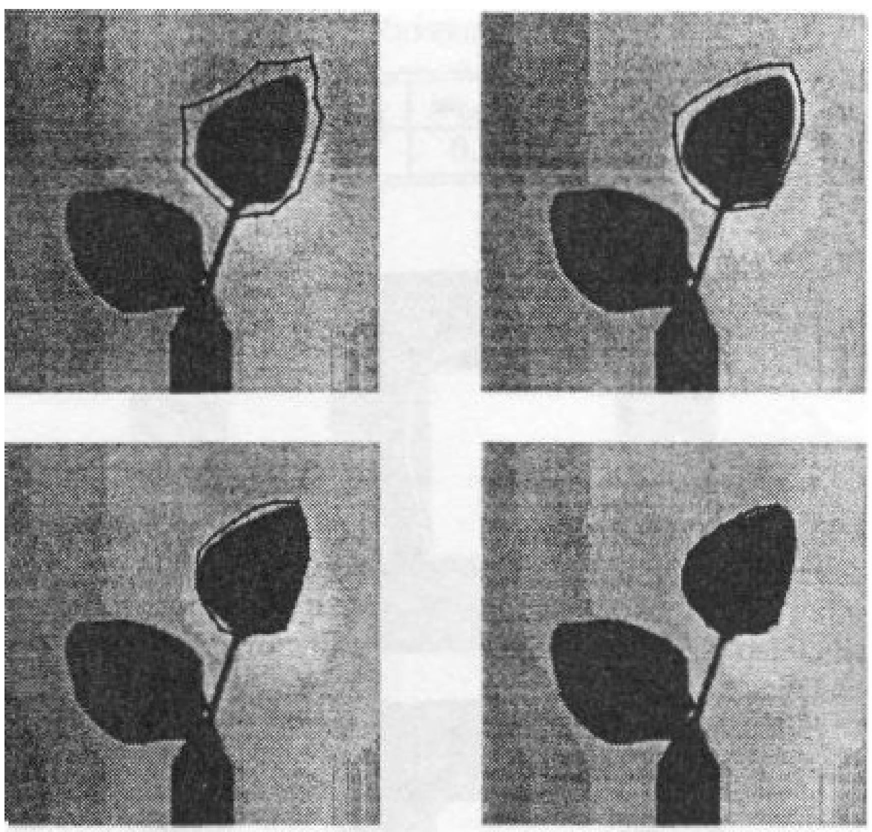
The input is an intensity image I containing the target contour and points $p_1,...,p_N$, defining the initial position and shape of the snake.

1. For each p_i, $i = 1, \ldots, N$, search its MxM neighborhood to find the location that minimizes the energy functional; move p_i to that location.

2. Estimate the curvature of the snake at each point and look for local maxima (i.e., corners); Set β_j to zero for each p_j at which the curvature is a local maximun and exceeds a threshold.

3. Update the value of $\bar{d}$.

Repeat steps 1-3 until only a very small fraction of snake points move in an iteration.



• Implementation details

- It is important to normalize the contribution or each term for correct implementation:

(1) For E_{cont} and E_{curv}, it is sufficient to divide by the largest value in the neighborhood in which the point can move.

(2) normalize $\overline{||\nabla I||}$ as $\frac{||\nabla I|| - min}{max - min}$ where min and max are the minimum and maximum gardient values in the neighborhood.

• Comments

- This approach is simple and has low computational requirements ($O(MN)$).

- It does not guarantee convergence to the global minimum of the functional.

- Works very well as far as the initial snake is not too far from the desired solution.