Deformable Contours

CMPE 264: Image Analysis and Computer Vision
Hai Tao
Deformable contour

- Often, we would like to fit an arbitrary curve to a set of image edge points. The approach that uses edge detection and Hough transform will not work because it is in general difficult to describe arbitrary curves using parametric forms.
- A solution to this problem is using deformable contour, or active contour, or snake.

An implementation of an active contour model
Courtesy of Clive Saha, John Hsu and Chivas Nambiar at Cornell University
Deformable contour

- An example of active contour tracking

Ultrasonic image sequences and the snake tracking results
Courtesy of the University of Maryland at Baltimore
Snake

Key ideas

- Associate an energy functional to each possible contour shape so that the energy function reaches its minimum for the desired contour.

- More specifically, the energy function is defined as

\[ E = \int_{c} \left( \alpha(s) E_{\text{cont}} + \beta(s) E_{\text{curv}} + \gamma(s) E_{\text{image}} \right) ds \]

where the contour \( c = c(s) \) is parameterized by its arc length, \( s \).

- The first two terms are defined so that smooth contours are favored. This will prevent overfitting.

- The last term favors contour passing through high image gradient (edge) areas.

- The coefficients \( \alpha(s), \beta(s), \) and \( \gamma(s) \) balance the smoothness and the fitness of the contour.
Snake – continuity term

- The continuity term $E_{cont}$ is defined as

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

When the contour is represented by a chain of points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ along the contour, the discrete form of the continuity term becomes

$$E_{cont} = \| \mathbf{p}_i - \mathbf{p}_{i-1} \|^2$$

Minimizing this energy will make the contour points clutter. A better energy function is

$$E_{cont} = (\overline{d} - \| \mathbf{p}_i - \mathbf{p}_{i-1} \|)^2$$

where $\overline{d}$ is the average distance between consecutive contour points. This energy function favors equally spaced contour points.
Snake – smoothness term

- The smoothness term $E_{curv}$ is defined as the curvature of the contour. Minimizing this term yields smooth contours.
- The discrete form of this term is

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

which is an approximation of the second order derivative (curvature) of the contour shape.
- The continuity and smoothness energy terms do not depend on the image, therefore are called the internal energy terms.
Snake – image term

- The term $E_{\text{image}}$ corresponds to the energy associated to the “external force” that attracts the contour.
- Minimizing this term will favor contours that pass through edge points.

$$E_{\text{image}} = -\|\nabla I\|^2$$

- If we only optimize the external energy term, we will obtain contours with oscillations, which can be caused by imaging noise.
- If we only optimize the internal energy term, we will obtain a smooth contour, but it has nothing to do with the image.
Problem statement

- **Assumptions**
  - $I$ is an input image
  - $\overrightarrow{p}_1, \ldots, \overrightarrow{p}_n$ are the initial positions of the chain of the contour points

- Starting from $\overrightarrow{p}_1, \ldots, \overrightarrow{p}_n$, find the deformable contour $\overrightarrow{p}_1, \ldots, \overrightarrow{p}_n$ which fits the target image contour best by minimizing the energy functional

$$E = \sum_{i=1}^{n} (\alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{image})$$
Snake – greedy algorithm

- A two-step iterative algorithm
  - Step 1: Greedy minimization
    - For each contour points, search for a new position in its neighborhood that minimizes the energy functional value
  - Step 2: Corner elimination
    - Search for corners as the curvature maxima along the contour. If such a point is \( p_j \), then \( \beta_j \) is set to zero. This makes it possible to allow snake to handle shapes with corners
Snake - Algorithm

Algorithm_Snake

- Given an input image $I$ and initial contour points $\vec{p}_1, \ldots, \vec{p}_n$, iterate the following steps until the number of contour points that are actually updated is smaller than a predefined threshold $f$.
- For each contour point, find the location that minimizes the energy function.
- For each contour point, estimate the curvature as
  \[ k = \| \vec{p}_{i-1} - 2\vec{p}_i + \vec{p}_{i+1} \|^2 \]
  and look for local maxima. Set $\beta_j = 0$ for $\vec{p}_j$ at which the curvature has a local maxima and exceeds a predefined threshold.
- Update the average distance $\bar{d}$.
Some results

Figure 5.7 (a–c): Initial position of the snake, intermediate position (6th iteration), final result (20th iteration). Parameters were $\alpha_i = \beta_i = 1$, $\gamma_i = 1.2$. SNAKE used $7 \times 7$ local neighborhoods, and stopped when only 4 or fewer points changed in an iteration.
Some results

Figure 5.8 (a): Initial position of the snake. (b): Intermediate position (84th iteration).
(c): Final result (130th iteration). Parameters were $\alpha_i = \beta_i = 1$, $\gamma_i = 1.2$. SNAKE used $7 \times 7$ local neighborhoods, and stopped when only 9 or fewer points changed in an iteration.