An Overview of Snake Algorithm: Matlab Implementation

Iman Moazzen*, PhD and Mohsen Shakiba**, PhD

* University of Victoria, Canada, ** Isfahan University of Technology, Iran

Introduction:

Tracking of contours has many applications such as motion modeling, biomedical image analysis, surveillance or autonomous vehicle navigation. A special attention in the past years has been devoted to contour tracking by means of snake models. Snakes [1], or active contours, are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image.

Active Contours:

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. The procedure is as follows:

(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.

In fact, it is based on minimizing an **energy function** which contains several terms; each is corresponding to some force acting on the contour. The goal is to minimize this function with respect to the contour parameters.

A suitable energy function is the sum the following three terms:

$$E = \int (\alpha E_{cont} + \beta E_{curv} + \gamma E_{image}) \, ds \qquad (1)$$

The parameters $\alpha, \beta,$ and γ control the relative influence of the corresponding energy terms and can vary along the contour. E_{cont} and E_{curv} are called internal energy terms. E_{image} is called external energy term. It is important to normalize the contribution of each term for correct implementation

Each energy term serves a different purpose:

E_{image}: attracts contour toward the closest image edge. This can be achieved by the following function (where ∇I is the gradient of the intensity computed at each snake point):

$$E_{image} = -\|\nabla I\| \qquad (2)$$

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

E_{cont}: forces the contour to be *continuous*. It minimizes the first derivative. 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:

$$\begin{aligned} E_{cont} &= \|p_i - p_{i-1}\|^2 \\ E_{cont} &= (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \end{aligned} \tag{3}$$

However, this term tries to minimize the distance between the points, and it causes the contour to shrink. A better form for E_{cont} is as follows (where $\bar{d}$ is the average distance between the points of the snake):

$$E_{cont} = \left(\bar{d} - \|p_i - p_{i-1}\|\right)^2 \tag{4}$$

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

E_{curv}: forces the contour to be *smooth*. The purpose of this term is to enforce smoothness and avoid oscillations of the snake by penalizing high contour curvatures. It minimizes the second derivative (curvature). In the discrete case, the curvature can be approximated by the following finite difference:

$$\begin{aligned} E_{curv} &= \|p_{i-1} - 2p_i + p_{i+1}\|^2 \ or \\ E_{curv} &= (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \end{aligned} \tag{5}$$

Discrete formulation of the problem:

Let I be an image and $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N$ the initial locations of the snake (evenly spaced, chosen close to the contour of interest). Starting from $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N$ find the deformable contour $p_1, p_2, \ldots, p_N$ which fits the target contour by minimizing the energy function:

$$E = \sum_{i=1}^{N} \alpha_i E_{cont} + \beta_i\, E_{curv} + \gamma_i\, E_{image} \tag{6}$$

Algorithm:

The input is an intensity image I containing the target contour and points $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N$ defining the initial position and shape of the snake.

Step.1. For each $\bar{p}_i$, $1 \le i \le N$, search its $M \times M$ neighborhood (search is done in a square window whose length is M) to find the location that minimizes the energy function; move $\bar{p}_i$ to that location.

Step.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 $\bar{p}_j$ at which the curvature is a local maximum and exceeds a threshold.

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

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

References:

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.,* vol. 1, pp. 321–331, 1987.

[2] B.H.Lee, I.Choi, Gi.J.Jeon, "Motion-based moving object tracking using an active contour", *ICASSP* 2006.

[3] E.Trucoo, A.Verri, "Introductory Techniques for 3-D Computer Vision", Prentice Hall, 1998.