Snakes, Active Contour Models

Submitted by

Richa Kumar (2501329)
Harini Veeraraghavan (2532478)

Acknowledgements

We would like to thank Dr. Nicolaos Papanikolopolous for his guidance and help during this project. He was a great source of encouragement. The project was interesting and challenging and has helped us gain some basic concepts in the field of active contours. This will help us immensely with our future research.

Table of Contents

Summary **4**

Introduction **5**

Energy Functionals *5*
Internal Energy 5
External Energy 5
Image Energy 6

Energy Minimization **7**

Semi- Implicit Scheme *7*

Energy Maximization **9**

Implementation **11**

Iteration1 : Attempt to implement Kass's snake *11*

Iteration 2 : Addition of Termination Energy *11*

Iteration 3 : Using two snakes for object detection *12*

Iteration 4 : Using image properties to guide the two snakes *13*

Iteration 5 : Increasing/decreasing number of Control points and redistribution of points *13*

Simulation of Object Motion **14**

Constraints **15**

References **16**

Summary

Snakes are energy minimizing curves that help in detecting objects of interest. A snake is initialized automatically or manually by a set of control points. Associated with a snake is an energy function. This energy function is used to move the snake in the image. A gradient descent approach is used to move the snake towards the object of interest as in Kass et al's implementation. A gradient ascent technique is used in combination with the gradient descent method to track the object accurately. The snake moves towards the object's boundary from outside by minimizing this energy using a gradient descent approach and moves out towards the boundary by maximizing its energy from inside. Once the snake has detected an object, the object can be tracked for small displacements.

Introduction

A snake is defined by a set of control points, which represent an open or closed deformable contour. Snakes are used to detect certain image features. In order to do this the control points need to be recalculated with each iteration so that the contour moves closer to the object of interest. To find the new position of the control points, we use the energy of the snake, which is a function of the properties of the image as well as some internal and external forces. For each control point, the energy is recalculated for all points in its neighborhood and the point that minimizes this energy function is selected as the new control point.

This implementation is based on Kass et al's implementation where an active contour can only find the local energy minimum in an image. The control points should therefore be initialized around the feature that needs to be detected.

Energy Functionals

A snake can be represented by a set of control points u, where

$$u(s, t) = (x(s,t), y(s,t))$$

where s varies between 0 – N-1 where N is the number of control points and t is the time step or iteration.

The energy of a control point comprises of three main components:

- Internal Energy : This contributes towards the tension and the stiffness of the contour.
- External Energy : These are high level constraints which interact with the snake to guide it towards features of interest
- Image Energy : This component of the energy depends on properties of the image like intensity values, gradient magnitudes and directions, edges etc. and can be used to guide the snake towards such features.

Thus the total energy of the snake can be represented as:

$$E_{snake} = \int_0^{N-1} E_{intern}(u)\,ds + \int_0^{N-1} E_{extern}(u)\,ds + \int_0^{N-1} E_{image}(u)\,ds \qquad \text{----- (1)}$$

where 0 to N-1 represents the number of snake elements.

Internal Energy

The internal energy of a snake element as in [1] is defined as a function of tension and stiffness

$$E_{intern} = \underbrace{\alpha(s)\left|\frac{\partial u}{\partial s}\right|^2}_{\text{(tension)}} + \underbrace{\beta(s)\left|\frac{\partial^2 u}{\partial s^2}\right|^2}_{\text{(stiffness)}} \qquad \text{----- (2)}$$

The first order term makes the snake contract guiding it towards the object by introducing tension while the second order term allows it to resist bending and retain the shape of a smoothed curve by producing stiffness. Alpha value controls the extent of collapsibility in the absence of other constraints. A very large value of alpha allows the snake to collapse to a point. The beta term controls the curvature of the snake. A very small value of beta allows the snake to develop corners.

External Energy

External energy can be introduced by controlling the attraction or repulsion energy to drive the snake towards a feature of interest.

The attractive energy term between a snake element and a point i in the image [1] is given as

$$E_{extern}(x) \quad = \quad k|i-x|^2 \qquad \text{---- (3)}$$

Similarly, repulsive force for the snake element [1] is calculated as,

$$E_{extern}(x) \quad = \quad \frac{k}{|i-x|^2} \qquad \text{---- (4)}$$

Image Energy

The image energy of a snake is comprised of three different potential energy functions, which attract snake towards edges, lines and terminations. The total energy is represented as a weighted sum of these energies.

$$\text{Image energy} \quad = \quad w_{line}E_{line} \quad + \quad w_{edge}E_{edge} \quad + \quad w_{term}E_{term} \qquad \text{---- (5)}$$

Line energy is calculated as a function of the intensity of the image as a result of which snake will be attracted towards light or dark regions of the image.

Edge functional is calculated using the gradient of the image as

$$\boldsymbol{E_{edge}} \quad = \quad -\int_0^1 \left|\frac{\partial I}{\partial x}\right|^2 ds \qquad \text{---- (6)}$$

Line termination energy provides a means for attracting snakes towards corners and terminations.

$$\boldsymbol{E_{term}} \quad = \int_0^1 \frac{C_{yy}C_x{}^2 + C_{xx}C_y{}^2 - 2C_{xy}C_xC_y}{\left(C_x^2 + C_y^2\right)^{3/2}} ds \qquad \text{---- (7)}$$

Energy Minimization

The nearest local energy minimum can be found using an iterative gradient descent approach. The elements of the snake can be moved using either an Euler Time stepping scheme or the semi-implicit scheme of [1].

The change in energy of E(u) can be expressed as

$$\delta E = \int_0^{N-1} \frac{\delta E}{\delta u} \delta u ds \quad \text{---- (8)}$$

When the energy is minimum δE/δu = 0.
For energy to decrease with each change in δu,

$$\delta E < 0$$

δu can be calculated as

$$\delta u = \delta t\, f(u) \qquad \text{where } f(u) \equiv \frac{-\delta E}{\delta u} \quad \text{---- (9)}$$

The snake energy functional is given as

$$E = \underbrace{\int_0^{N-1} P(u)\, ds}_{\text{potential energy}} + \underbrace{\frac{\alpha}{2} \int_0^{N-1} \left| \frac{\partial u}{\partial s} \right|^2 ds}_{\text{tension}} + \underbrace{\frac{\beta}{2} \int_0^{N-1} \left| \frac{\partial^2 u}{\partial s^2} \right|^2 ds}_{\text{stiffness}} \quad \text{---- (10)}$$

A small change in *u* produces a corresponding change in the energy E.
Using the Euler Lagrange equations as in [2] we arrive at the equation,

$$\frac{\delta E}{\delta u} = \frac{\partial P}{\partial u} - \alpha \frac{\partial^2 u}{\partial s^2} + \beta \frac{\partial^4 u}{\partial s^4} = 0 \quad \text{---- (11)}$$

This equation is satisfied by all extrema: minima, maxima and points of inflection. An iterative gradient descent approach has to be used since *u* is not known before $\partial P / \partial u$ can be found.

Semi- Implicit Scheme

The equations of motion for minimizing the basic snake energy functional by iterative gradient descent are

$$\frac{\partial u}{\partial t} = -\frac{\delta E}{\delta u} = -\frac{\partial P}{\partial u} + \alpha \frac{\partial^2 u}{\partial s^2} - \beta \frac{\partial^4 u}{\partial s^4} \quad \text{---- (12)}$$

The approximation of the above equation using the finite difference formulae is,

$$\frac{u_s^{t+\delta t} - u_s^t}{\delta t} = \alpha\left(u_{s+1}^{t+\delta t} + u_{s-1}^{t+\delta t} - 2u_s^{t+\delta t}\right) - \beta\left(u_{s+2}^{t+\delta t} - 4u_{s+1}^{t+\delta t} + 6u_s^{t+\delta t} - 4u_{s-1}^{t+\delta t} + u_{s-2}^{t+\delta t}\right) - \frac{\partial P}{\partial u_s^t} \quad \text{---- (13)}$$

The above equation can be expressed as,

$$\tilde{u}_s^{t+\delta t} = u_s^t - \partial t \frac{\partial P}{\partial u_s^t} \quad \text{---- (14)}$$

where,

$$\tilde{u}_s^{t+\delta t} = \left(pu_{s+2}^{t+\delta t} + qu_{s+1}^{t+\delta t} + ru_s^{t+\delta t} + qu_{s-1}^{t+\delta t} + pu_{s-2}^{t+\delta t}\right), \qquad \text{---- (15)}$$

where,

$$p \equiv b \qquad a \equiv \alpha \frac{\delta t}{\delta s^2}$$

$$q \equiv -a - 4b$$

$$r \equiv 1 + 2a + 6b \qquad b \equiv \beta \frac{\delta t}{\delta s^4}$$

A pentadiagonal matrix M, whose diagonal element is the sequence, can represent the above relationship : p q r q p.

The above can be written as

$$Mx^{t+\delta t} = \tilde{x}^{t+\delta t} = x^t - \delta t \frac{\partial P}{\partial x^t} \qquad \text{---- (16)}$$

$$My^{t+\delta t} = \tilde{y}^{t+\delta t} = y^t - \delta t \frac{\partial P}{\partial y^t} \qquad \text{---- (17)}$$

the final solution is given by,

$$x^{t+\delta t} = M^{-1}\left(x^t - \delta t \frac{\partial P}{\partial x^t}\right) \qquad \text{---- (18)}$$

$$y^{t+\delta t} = M^{-1}\left(y^t - \delta t \frac{\partial P}{\partial y^t}\right) \qquad \text{---- (19)}$$

Energy Maximization

In the previous section it has been outlined how the snake minimizes it's energy to find the object of interest. This is useful when the snake is finding the object from the outside. However during the tracking process the object may move in such a way that the snake may be inside the object and needs to find it from the inside. In this case energy minimization will not work. We therefore need energy maximization for this purpose.

The relation used for energy minimization was :

$$\frac{\partial u}{\partial t} = -\frac{\delta E}{\delta u} = -\frac{\partial P}{\partial u} + \alpha\frac{\partial^2 u}{\partial s^2} - \beta\frac{\partial^4 u}{\partial s^4} \qquad \text{---- (20)}$$

For maximization we need to invert the term for potential energy and the term which collapses it to a point. Instead the alpha should be negative so as to create a balloon force which pushes the snake outwards, instead of inwards.

Therefore we use the following relation for energy maximization :

$$\frac{\partial u}{\partial t} = \frac{\partial P}{\partial u} - \alpha\frac{\partial^2 u}{\partial s^2} - \beta\frac{\partial^4 u}{\partial s^4} \qquad \text{---- (21)}$$

This gives us the following set of equations for calculating the new position of the control points

$$\frac{u_s^{t+\delta t} - u_s^t}{\delta t} = -\alpha\left(u_{s+1}^{t+\delta t} + u_{s-1}^{t+\delta t} - 2u_s^{t+\delta t}\right) - \beta\left(u_{s+2}^{t+\delta t} - 4u_{s+1}^{t+\delta t} + 6u_s^{t+\delta t} - 4u_{s-1}^{t+\delta t} + u_{s-2}^{t+\delta t}\right) + \frac{\partial P}{\partial u_s^t} \qquad \text{---- (22)}$$

The above equation can be expressed as,

$$\tilde{u}_s^{t+\delta t} = u_s^t + \partial t\frac{\partial P}{\partial u_s^t} \qquad \text{---- (23)}$$

where,

$$\tilde{u}_s^{t+\delta t} = \left(pu_{s+2}^{t+\delta t} + qu_{s+1}^{t+\delta t} + ru_s^{t+\delta t} + qu_{s-1}^{t+\delta t} + pu_{s-2}^{t+\delta t}\right), \qquad \text{---- (24)}$$

where,

$$p \equiv b \qquad a \equiv \alpha\frac{\delta t}{\delta s^2}$$
$$q \equiv +a - 4b$$
$$r \equiv 1 - 2a + 6b \qquad b \equiv \beta\frac{\delta t}{\delta s^4}$$

A pentadiagonal matrix M, whose diagonal element is the sequence, can represent the above relationship : p q r q p.

The above can be written as

$$Mx^{t+\delta t} = \tilde{x}^{t+\delta t} = x^t + \delta t\frac{\partial P}{\partial x^t} \qquad \text{---- (25)}$$

$$My^{t+\delta t} = \tilde{y}^{t+\delta t} = y^t + \delta t \frac{\partial P}{\partial y^t} \quad \text{---- (26)}$$

the final solution is given by,

$$x^{t+\delta t} = M^{-1}\left(x^t + \delta t \frac{\partial P}{\partial x^t} \right) \quad \text{---- (27)}$$

$$y^{t+\delta t} = M^{-1}\left(y^t + \delta t \frac{\partial P}{\partial y^t} \right) \quad \text{---- (28)}$$

Implementation

Iteration1 : Attempt to implement Kass's snake

The first iteration was a strict implementation of Kass's snake. The functionals used have been mentioned earlier in the report. In this first attempt , termination energy as well as external forces were excluded. This means the energy functionals were as follows :

Internal Energy = Tension + Stiffness
Image Energy (Potential Energy) = Line Energy + Edge Energy

Observations :
As a number of parameters are involved in tuning the snake (parameters that define the weights of the various energies) it was found that the snake behaved differently with changes in parameters however it was never able to tightly lock on to objects. It was observed in some cases that when the snake found one point on the object, it stopped there instead of bringing the remaining points close to the object as well.

Given below is a screen dump of one such trial run of our implementation of kass's snake.

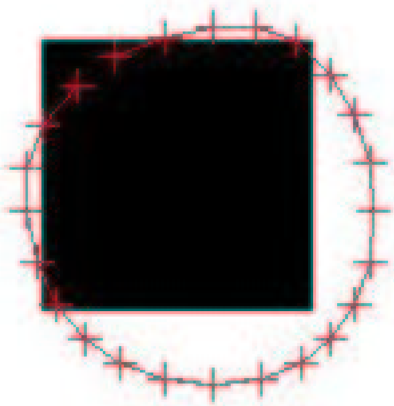

*figure 1 : Image dump after **55** iterations of the snake.*

The snake does deform according to the image properties however it does not closely lock onto the object as desired.

Iteration 2 : Addition of Termination Energy

From the observations of Iteration 1 it seemed that the behavior of the snake could be attributed to the omission of the Termination Energy which attracts the snake towards the line and edge terminations. So the changes made in this iteration were to include the Termination Energy in the calculation of the Potential Energy. The Potential Energy for each control point was now calculated as :

Potential Energy = Line Energy + Edge Energy + Termination Energy

Observations :
It was found that the way the snake converged (in terms of it's shape) was quite different. It tended to enter into the object and collapsed. This may be because the termination energies were extremely small in magnitude and did not seem to help.

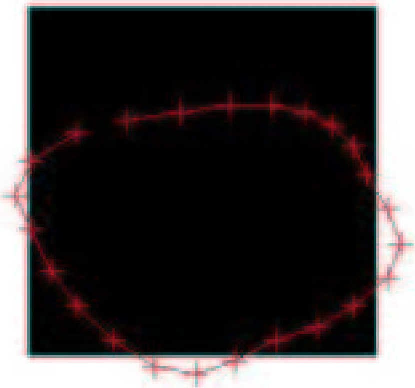

figure 2 : Snake implementation using termination energy
*(after **65** iterations)*

Iteration 3 : Using two snakes for object detection

Since the earlier implementations were not producing the expected results, a different strategy was employed for the implementation of the snake. The stiffness and tension properties were defined as described in Kass's implementation. However two snakes were used for finding the object. The real snake was the one that locked onto the object whereas the other snake (which was called the pseudo snake) guided the real snake till it fully locked onto the object. The strategy was as follows : Initially both the snakes started off together, guided only by stiffness and tension (having nothing to do with the image properties). However everytime the new control points were calculated the point was checked to see if it was a point of interest (point on object to be detected). If it was, that control point of the real snake was locked not allowing it to move any further into the object, while the pseudo snake continued to move on. So at this point the two snakes separated. The purpose was to lock points that had found the object while still moving other control points that had not yet found the object , towards the object.

Observations :
One major drawback of this implementation is that it precludes any tracking mechanism that may be required of the snake. The snake may be able to closely lock onto the object but it will not move with it since it does not use image properties for moving, it only uses image properties for the stopping condition.

Given below is a screen dump of the results

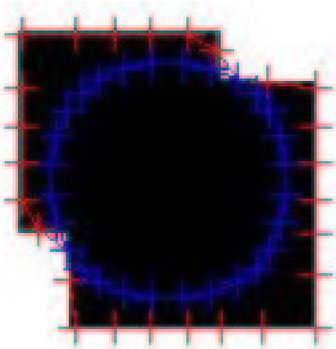

figure 3 : real snake at the boundary and pseudo snake within the image
*(**55** iterations)*

Iteration 4 : Using image properties to guide the two snakes

To overcome the drawback of the previous iteration, image properties were used to guide the two snakes. The "real" snake did not lock onto the object as tightly as before however it will probably be able to track the object since the image properties are being used to calculate new positions of the control points.

Given below is a screen dump of the implementation :

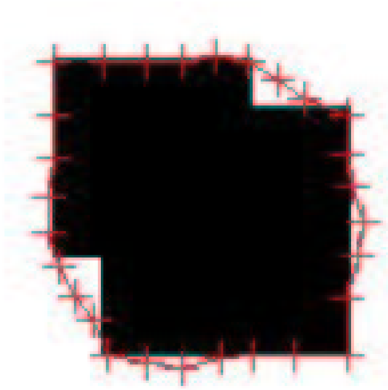

figure 4 : two snakes implementation using image properties
*(**40** iterations)*

Iteration 5 : Increasing/decreasing number of Control points and redistribution of points

Due to the fact that two snakes are used in order to find and track the object, the control points do not remain equally distributed along the boundary of the object. This affects the tracking process. To rectify this situation the points are redistributed explicitly along the boundary of the object everytime the object is detected to have moved.
This is done as follows :
The average distance between the control points is calculated. Lines are fitted into the various control points. All points on a line except for the end points are discarded and new points are added on the line (if required) while keeping them equally spaced.

Observations :
The redistribution of points improved the tracking process. It also takes care of adding new points if required and removing unwanted control points. Due to this the tracking becomes more accurate.

Given below are some screen dumps of how the object is tracked as it moves :

Simulation of Object Motion

To test whether the snake could track an object, motion of an object was simulated in the following way :

1. The snake was allowed to lock onto the object within an image.
2. Once the snake locked onto the object , the image was changed with the object in a new position (as if it had moved)
3. The snake algorithm was run again with the new image and the last set of control points from the locked object.

Provided below are screen dumps showing how this was done.

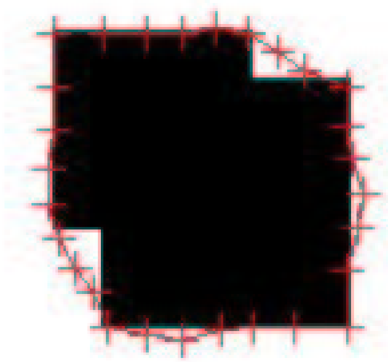

figure 5 : step 1, snake locks onto object

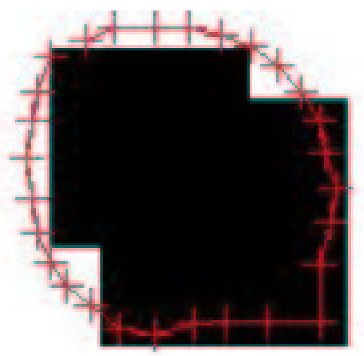

figure 6 : step 2, object moved

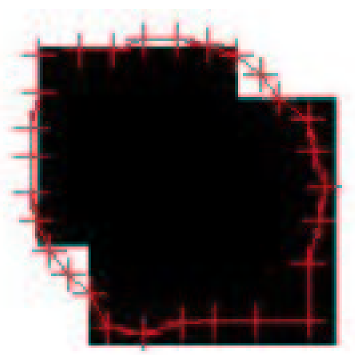

figure 7 : step 3, snake is able to lock onto the object when it has to move from light to dark, but the control points that are already within the object cannot expand outwards to find the object boundary

Constraints

The following are some of the constraints of implementation discussed in this report :

1. Images must be binary
2. Images must be noise free (this requires that the image be pre-processed)
3. Objects can be tracked for small displacements (however this may not be a very unreasonable constraint as motion of objects between consecutive frames is generally small)

References

Ivins, J; Porrill, J, 1993. (Revised June 1995; March 2000.)
Everything You Always Wanted To Know About Snakes (But Were Afraid To Ask).

Ivins, J.P. : *Statistical Snakes: Active Region Models*

M. Kass, A. Witkin, and D. Terzopoulos, *"Snakes: Active contour models."* International Journal of Computer Vision. v. 1, n. 4, pp. 321-331, 1987.

Young David, March 1995 : *Sussex Computer Vision: Teach Vision7* ,
URL: www.cogs.susx.ac.uk/users/davidy/techvision/vision7.html