A fast, simple active contour algorithm for biomedical images

H. Eviatar *, R.L. Somorjai

National Research Council of Canada, Institute for Biodiagnostics, 435 Ellice Avenue, Winnipeg, MB, R3B 1Y6, Canada

Received 6 September 1995; revised 17 January 1996

Abstract

A new method for the application of active contours to biomedical images is described. The new approach, which involves extensive modification of the internal energy function and a different method of minimising the energy functional, yields rapid, excellent fits to MR images.

Keywords: Active contours; Snakes; Minimisation; Harmonic potential; Biomedical images

1. Introduction

Increasingly, clinical studies are concerned with the calculation of areas/volumes from biomedical images, such as are produced by MRI of the brain or fetal ultrasound. Typically, the operator, using a mouse, laboriously traces the edges of features in a stack of two-dimensional images, often repeating the task several times to minimise error. The resulting areas are then combined to calculate the volume. Since automating this time-consuming operation would be welcome, we have developed a quick, simple method for tracing edges in biomedical images, starting with a single, roughly drawn contour. This is an active area of research (Ranganath, 1995; Ashton et al., 1995). Our method is based on a modified form of active contours, or snakes. Snakes are mathematical constructs consisting of a series of polygon vertices strung together like beads on a wire. In combination with a registration algorithm (Alexander et al., 1995; Alexander and Somorjai, 1996), the snakes can be transferred from slice to slice to produce a stack, yielding the required volume.

Active contours were first introduced by Kass et al. (1988), as part of a larger system for describing human vision. We are only interested in those aspects which employ snakes as an interactive means of tracking edges. Their approach is founded on physics-based shape design, i.e., the use of concepts borrowed from classical mechanics to describe the behaviour of curves and shapes in two and three dimensions. The "energy" of the system formed by an image and a snake defined upon it is some objective function to be minimised; as a result of the minimising calculation, the snake is drawn to the feature of interest. Kass et al. suggested that this "energy" consist of an internal energy term \( E_{int} \), defined by characteristics of the snake, an external energy term \( E_{image} \) derived from the image, and...
possible additional constraints, $E_{\text{con}}$. Together they form a functional to be minimised:

$$E_{\text{snake}} = \int_0^1 \left[ E_{\text{int}}(x(s), y(s)) + E_{\text{image}}(x(s), y(s)) + E_{\text{con}}(x(s), y(s)) \right] ds, \quad (1)$$

where the position of the snake is represented parametrically by $x(s), y(s)$, along contour $s$, where $0 \leq s \leq 1$. The internal "energy" determines the characteristics of the snake. Ideally, one would wish the snake to be flexible, yet with a certain "coherence" which prevents it from breaking up. The image "energy" is derived from the image, and to use the snake as an edge detector, it is convenient to equate the image "energy" with the negative squared gradient of the image. The constraints can be used to force the snake in a particular direction or to prevent it from collapsing to a single point in the absence of external forces. Minimisation of the "energy" functional, Eq. (1), leads to the Euler–Lagrange equations, which are then used as dynamical equations of motion for the position of the snake vertices.

In our work, the basic idea of active contours — the minimisation of an "energy" — has been retained. However, both the internal "energy" of the snake and the solution method have been modified.

2. Classical snakes

Kass et al. (1988) use an expression for the internal energy of the snake which involves first and second derivatives of the snake vertex coordinates and two adjustable parameters, $\alpha$ and $\beta$:

$$E_{\text{int}} = \frac{(\alpha |x'|^2 + \beta |x''|^2)}{2}, \quad (2)$$

where $x'$ is the first derivative of $x$ with respect to $s$, and $x''$ is the second derivative. In this work, we will only give the expressions for the $x$-coordinate; the equations in $y$ are entirely equivalent. The derivatives may be implemented using either finite differences (Kass et al., 1988) or finite elements (Cohen and Cohen, 1990). Most snakes in the literature (Kass et al., 1988; Amini et al., 1990; Davatzikos and Prince, 1995) have been designed for interactive use. The algorithm is initialised by manually drawing a snake approximately along the edge to be fitted. Minimisation of the energy functional, Eq. (1), leads to the Euler–Lagrange equations, which can be turned into dynamical equations of motion in $x(s)$ and $y(s)$ by the addition of a $d/dt$ term (see Kass et al., 1988). Integration of these equations in time then leads, hopefully, to a stable equilibrium in which the snake is drawn to the contour and stays there.

The external energy, as defined by Kass et al., consists of constraint terms and image terms. They included repulsion terms ("volcanoes") to keep the snake from collapsing to a single point. The "volcanoes" also serve to push the snake manually out of an undesirable local minimum. Kass et al. were interested in modelling human vision; we are merely concerned with automating the volume extraction now carried out manually by clinicians. We have not found it necessary to implement explicit constraint terms, but this could easily be done should the need arise.

3. Modified snakes

The Euler–Lagrange equations are meant to be used on continuous systems. As the snakes are dis-
crete by their very nature, we have avoided employing discretised numerical derivatives in the calculation of the internal energy, thereby escaping many of the inaccuracies inherent in their use. Furthermore, the nature of our problem lends itself to a simplified approach. We have therefore replaced Eq. (2) by a simple nearest neighbour pair potential with a constant equilibrium term \( r_i \):

\[
E_{\text{int}}(i) = \kappa(i)(| x_i - x_{i+1} | - r_i)^2, \tag{3}
\]

where \( i \) indexes the \( N \) snake vertices and \( \kappa(i) \) is a nonnegative free parameter which may, if desired, be varied from vertex to vertex along the snake (see Fig. 1). \( x_i \) is the value of the \( x \) coordinate at the position of the \( i \)th vertex. An identical expression is used for the \( y \) coordinate. As \( E_{\text{int}} \) is minimised when \( r_i = | x_i - x_{i+1} | \), the equilibrium term \( r_i \) prevents the snake from collapsing to a single point, while maintaining any desired local flexibility. Eq. (3) has the form of a harmonic potential and describes a "stretching or compression energy", where
κ controls the elasticity. For large values of κ the snake is very stiff and holds together well, for small values it will fold easily into the crevices of the image, but may scatter to different edges and lose its coherence. Eq. (3) is extremely simple; it can be extended easily to embrace further neighbours and more sophisticated repulsion terms. In particular, an interaction involving the second nearest neighbours would provide an approximation to the bending energy (see Veltkamp and Wesselink, 1995). However, we have not yet found it necessary to implement it.

Although the exact form of the "energy" generated from the image is not germane to our algorithm, for this particular application we have employed a Gaussian filter to blur the image (the variance of the Gaussian, σ, is an adjustable parameter) and a Sobel operator to extract the image gradient in the x and y directions. The removal of high-frequency components in the image by the Gaussian filter allows us to calculate gradients with confidence. The quantity used for E_image is the negative squared gradient—consequently the snake is drawn to edges in the image (Fig. 4). Eq. (1) is now replaced by

$$ E_{\text{snake}} = \sum_{i=0}^{N-1} \left( E_{\text{int}}(i) + E_{\text{image}}(i) \right). $$

(4)

Rather than using dynamical equations of motion, we have chosen to minimise Eq. (4) directly, using the expression for E_int shown in Eq. (3). This procedure
has the advantage of working exceedingly well and quickly. We have experimented with global and local minimisers, and found that the best results were achieved using Powell’s method of conjugate directions, as described in (Press et al., 1992). An important advantage of Powell’s method is that it, too, does not require the use of derivatives.

4. Results

We have applied the algorithm outlined above to several images; it invariably converged rapidly. A typical run requires approximately 0.3 sec per vertex on an SGI Challenge XL, using a 256 × 256 image. Open and closed snakes could be optimised with equal facility. An implementation of the algorithm of Kass et al. was also used for comparison, but was found to be less responsive to the vague edges common in biomedical images. Figs. 2 and 3 show two typical snake configurations on an MRI image of a human brain. In the first, the operator traces a rough, closed snake around the skull; the computation moves the snake so that it closely hugs the edge. Parameter values used for this problem: \( \kappa = 800, r_i = \text{vertex separation in initial snake}/16 \). This snake has 29 vertices. In the second, an open snake is placed near the edge of the anterior cingulate: the computed snake follows the convoluted edge of the feature. Parameter values used for this problem: \( \kappa = 50, r_i = \text{vertex separation in initial snake}/16 \). This snake has 12 vertices.

Fig. 4. The gradient image used to guide the snakes.
5. Conclusions

We have shown that our active contour algorithm provides quick, accurate delineation of edges in biomedical images. Immediate extensions include more sophisticated potentials for controlling the stiffness of the snake, and also local control of the number of snake points to be calculated, based on the local curvature of the image. These extensions are under development. We are adapting the methodology described here for use in conjunction with a registration algorithm (Alexander et al., 1995; Alexander and Somorjai, 1996) for the automatic generation of volumes from a collection of two-dimensional slices.

References


